OnApp Cloud 6.5 Edge 5 CDN Guide
# Table of Contents

## 1 Get Started

1.1 OnApp Dashboard ........................................................................................................ 9
1.2 How your CDN Account Works .................................................................................. 10
1.3 Hardware & Marketplace Requirements ..................................................................... 11
1.3.1 Hardware Requirements ....................................................................................... 11
1.3.2 CDN Marketplace Requirements .......................................................................... 12
1.3.3 Recommended Network Configuration ................................................................ 13
1.4 OnApp CDN Terms of Use ......................................................................................... 13
1.5 Types of CDN Deployment ......................................................................................... 13
1.5.1 CDN with Marketplace Resources Only ................................................................. 13
1.5.2 Own Edge Servers with Marketplace Resources ................................................... 14
1.6 Set up CDNs Using Marketplace Servers Only ......................................................... 14
1.7 Set up CDNs Combining Local & Marketplace Resources ........................................ 15
1.8 CDN Activation & Configuration Process ..................................................................... 15
1.8.1 Enable CDN for Your Cloud ................................................................................ 15
1.8.2 Set Customer and Admin Permissions for CDN ..................................................... 16
1.8.3 Launch CDN Setup Wizard .................................................................................. 17
1.8.4 Create and Subscribe to CDN Edge Servers ......................................................... 20
1.8.5 Create CDN Storage Servers .............................................................................. 22
1.8.6 Create CDN Edge Groups & Assign Edge Servers to Them .................................. 24
1.8.7 Assign CDN Resources to Bucket and Set Prices ................................................ 24
1.9 Next Steps .................................................................................................................... 26

## 1.10 What’s New ............................................................................................................... 27
1.10.1 OnApp 5.9 ........................................................................................................... 27
1.10.2 OnApp 5.8 ........................................................................................................... 27
1.10.3 OnApp 5.7 ........................................................................................................... 27

## 2 Administration Guide

2.1 CDN Setup Wizard ...................................................................................................... 28
2.1.1 Step 1 Permissions .............................................................................................. 29
2.1.2 Step 2 Edge Groups ............................................................................................ 30
2.1.3 Step 3 Billing ....................................................................................................... 31
2.2 CDN Edge Servers ..................................................................................................... 31
2.2.1 Create CDN Edge Server .................................................................................... 32
2.2.2 View CDN Edge Server Details .......................................................................... 32
2.2.3 Manage CDN Edge Server ................................................................................ 34
2.2.4 Set VIP Status for Edge Server ........................................................................... 35
2.2.5 Delete CDN Edge Server .................................................................................... 36
2.2.6 CDN Edge Server Network Interface Usage ...................................................... 36
2.2.7 Slicing Cache Files into Smaller Files ................................................................. 37
2.3 CDN Storage Servers ............................................................................................... 37
2.3.1 View CDN Storage Server Details ....................................................................... 37
2.3.2 Create CDN Storage Server ............................................................................... 38
2.3.3 Edit CDN Storage Server ................................................................................... 40
2.3.4 Set VIP Status for Storage Server ....................................................................... 40
2.3.5 Delete CDN Storage Server ............................................................................... 40
2.4 CDN Resources ......................................................................................................... 40
2.4.1 HTTP CDN Resource ........................................................................................ 41
2.4.2 Video On Demand CDN Resource ...................................................................... 42
2.4.3 Live Streaming CDN Resource ........................................................................... 42
2.4.4 View CDN Resources ........................................................................................ 42
2.4.5 View CDN Resource Details ............................................................................... 43
2.4.6 Create HTTP CDN Resource ............................................................................. 48
2.4.7 Create Video On Demand CDN Resource ......................................................... 56
2.4.8 Create Live Streaming CDN Resource ............................................................... 60
2.4.9 Edit CDN Resource ............................................................................................ 64
2.4.10 Raw Logs ............................................................................................................ 72
2.4.11 Prefetch Content .................................................................................................. 74
2.4.12 Purge Content .................................................................................................... 75
2.4.13 CDN Billing Statistics ........................................................................................ 76
2.4.14 Wildcard Invalidation Rules ................................................................................ 76
2.4.15 HTTP Caching Rules ......................................................................................... 78
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.2 Step 2 Edge Groups</td>
<td>146</td>
</tr>
<tr>
<td>3.1.3 Step 3 Billing</td>
<td>147</td>
</tr>
<tr>
<td>3.2 Edge Servers</td>
<td>147</td>
</tr>
<tr>
<td>3.2.1 Create Edge Server</td>
<td>149</td>
</tr>
<tr>
<td>3.2.2 View Edge Server Details</td>
<td>151</td>
</tr>
<tr>
<td>3.2.3 Edit Edge Server</td>
<td>152</td>
</tr>
<tr>
<td>3.2.4 Delete Edge Server</td>
<td>152</td>
</tr>
<tr>
<td>3.2.5 Edge Server Network Interface Usage</td>
<td>153</td>
</tr>
<tr>
<td>3.3 Storage Servers</td>
<td>153</td>
</tr>
<tr>
<td>3.3.1 View Storage Server Details</td>
<td>153</td>
</tr>
<tr>
<td>3.3.2 Create Storage Server</td>
<td>154</td>
</tr>
<tr>
<td>3.3.3 Edit Storage Server</td>
<td>155</td>
</tr>
<tr>
<td>3.3.4 Delete Storage Server</td>
<td>155</td>
</tr>
<tr>
<td>3.4 Resources</td>
<td>156</td>
</tr>
<tr>
<td>3.4.1 View Resources</td>
<td>156</td>
</tr>
<tr>
<td>3.4.2 View Resource Details</td>
<td>156</td>
</tr>
<tr>
<td>3.4.3 Create HTTP Resource</td>
<td>161</td>
</tr>
<tr>
<td>3.4.4 Create Video On Demand Resource</td>
<td>170</td>
</tr>
<tr>
<td>3.4.5 Create Live Streaming Resource</td>
<td>174</td>
</tr>
<tr>
<td>3.4.6 Edit CDN Resources</td>
<td>178</td>
</tr>
<tr>
<td>3.4.7 CDN Raw Logs</td>
<td>178</td>
</tr>
<tr>
<td>3.4.8 View Advanced Reporting</td>
<td>180</td>
</tr>
<tr>
<td>3.4.9 Prefetch CDN Resources Content</td>
<td>180</td>
</tr>
<tr>
<td>3.4.10 Purge CDN Resources Content</td>
<td>181</td>
</tr>
<tr>
<td>3.4.11 Billing Statistics</td>
<td>182</td>
</tr>
<tr>
<td>3.4.12 Token Authentication</td>
<td>182</td>
</tr>
<tr>
<td>3.4.13 Edge Server IP Ranges</td>
<td>187</td>
</tr>
<tr>
<td>3.5 Edge Groups</td>
<td>188</td>
</tr>
<tr>
<td>3.5.1 View Edge Group Details</td>
<td>188</td>
</tr>
<tr>
<td>3.5.2 Create Edge Group</td>
<td>189</td>
</tr>
<tr>
<td>3.5.3 Edit/Delete Edge Group</td>
<td>189</td>
</tr>
<tr>
<td>3.5.4 Assign/Remove Edge Group Locations</td>
<td>189</td>
</tr>
<tr>
<td>3.6 Upload Instructions</td>
<td>190</td>
</tr>
<tr>
<td>3.6.1 HTTP Push CDN Resource</td>
<td>190</td>
</tr>
<tr>
<td>3.6.2 VOD Pull CDN Resource</td>
<td>190</td>
</tr>
<tr>
<td>3.6.3 VOD Push CDN Resource</td>
<td>191</td>
</tr>
<tr>
<td>3.6.4 Live Streaming CDN Resource</td>
<td>192</td>
</tr>
<tr>
<td>3.7 SSL Certificates</td>
<td>196</td>
</tr>
<tr>
<td>3.7.1 View Custom SNI SSL Certificates</td>
<td>198</td>
</tr>
<tr>
<td>3.7.2 Add Custom SNI SSL Certificates</td>
<td>198</td>
</tr>
<tr>
<td>3.7.3 Edit Custom SNI SSL Certificates</td>
<td>199</td>
</tr>
<tr>
<td>3.7.4 Delete Custom SNI SSL Certificates</td>
<td>199</td>
</tr>
<tr>
<td>3.8 Reporting</td>
<td>200</td>
</tr>
<tr>
<td>3.8.1 Overview Report</td>
<td>200</td>
</tr>
<tr>
<td>3.8.2 Cache Statistics Report</td>
<td>201</td>
</tr>
<tr>
<td>3.8.3 Top Files Report</td>
<td>202</td>
</tr>
<tr>
<td>3.8.4 Top Referrers Report</td>
<td>202</td>
</tr>
<tr>
<td>3.8.5 Status Codes Report</td>
<td>203</td>
</tr>
<tr>
<td>3.8.6 Concurrent Statistics</td>
<td>204</td>
</tr>
<tr>
<td>3.8.7 Stream Bandwidth Report</td>
<td>205</td>
</tr>
<tr>
<td>3.8.8 Visitors Report</td>
<td>206</td>
</tr>
<tr>
<td>3.8.9 Usage Statistics</td>
<td>206</td>
</tr>
<tr>
<td>3.9 CDN Reports</td>
<td>207</td>
</tr>
<tr>
<td>3.10 DNS</td>
<td>208</td>
</tr>
<tr>
<td>3.10.1 DNS Setup</td>
<td>208</td>
</tr>
<tr>
<td>3.10.2 DNS Zones</td>
<td>209</td>
</tr>
<tr>
<td>4 API Guide</td>
<td>217</td>
</tr>
<tr>
<td>4.1 API Authentication</td>
<td>217</td>
</tr>
<tr>
<td>4.2 HTTP Methods</td>
<td>217</td>
</tr>
<tr>
<td>4.3 HTTP Response Codes</td>
<td>217</td>
</tr>
<tr>
<td>4.4 Formatting and Naming Conventions</td>
<td>218</td>
</tr>
<tr>
<td>4.5 FAQ</td>
<td>219</td>
</tr>
<tr>
<td>4.6 CDN API</td>
<td>219</td>
</tr>
<tr>
<td>4.7 Buckets API</td>
<td>220</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>4.10.1</td>
<td>Get List of Buckets</td>
</tr>
<tr>
<td>4.10.2</td>
<td>Get Bucket Details</td>
</tr>
<tr>
<td>4.10.3</td>
<td>Add Bucket</td>
</tr>
<tr>
<td>4.10.4</td>
<td>Edit Bucket</td>
</tr>
<tr>
<td>4.10.5</td>
<td>Clone Bucket</td>
</tr>
<tr>
<td>4.10.6</td>
<td>Delete Bucket</td>
</tr>
<tr>
<td>4.10.7</td>
<td>Access Control</td>
</tr>
<tr>
<td>4.10.8</td>
<td>Rate Card</td>
</tr>
<tr>
<td>4.10.9</td>
<td>CDN Access Control API</td>
</tr>
<tr>
<td>4.10.10</td>
<td>Get List of Accelerators</td>
</tr>
<tr>
<td>4.10.11</td>
<td>Get Accelerator Details</td>
</tr>
<tr>
<td>4.10.12</td>
<td>Add Accelerator</td>
</tr>
<tr>
<td>4.10.13</td>
<td>Edit Accelerator</td>
</tr>
<tr>
<td>4.10.14</td>
<td>Add/Edit Admin/User Note for CDN Accelerator</td>
</tr>
<tr>
<td>4.10.15</td>
<td>Rebuild Network for Accelerator</td>
</tr>
<tr>
<td>4.10.16</td>
<td>Start up Accelerator</td>
</tr>
<tr>
<td>4.10.17</td>
<td>Shut down Accelerator</td>
</tr>
<tr>
<td>4.10.18</td>
<td>Suspend Accelerator</td>
</tr>
<tr>
<td>4.10.19</td>
<td>Reboot Accelerator</td>
</tr>
<tr>
<td>4.10.20</td>
<td>Rebuild Accelerator</td>
</tr>
<tr>
<td>4.10.21</td>
<td>Migrate Accelerator</td>
</tr>
<tr>
<td>4.10.22</td>
<td>Delete Accelerator</td>
</tr>
<tr>
<td>4.10.23</td>
<td>Unlock Accelerator</td>
</tr>
<tr>
<td>4.10.24</td>
<td>Segregate Accelerator</td>
</tr>
<tr>
<td>4.10.25</td>
<td>Change Accelerator Owner</td>
</tr>
<tr>
<td>4.10.26</td>
<td>Accelerator Network Interfaces</td>
</tr>
<tr>
<td>4.10.27</td>
<td>Accelerator IP Address Joins</td>
</tr>
<tr>
<td>4.10.28</td>
<td>View Accelerator Disks</td>
</tr>
<tr>
<td>4.10.29</td>
<td>Rebuild Network for Accelerator</td>
</tr>
<tr>
<td>4.10.30</td>
<td>Get Accelerator CPU Usage Statistics</td>
</tr>
<tr>
<td>4.10.31</td>
<td>CDN Edge Groups API</td>
</tr>
<tr>
<td>4.10.32</td>
<td>Get List of CDN Edge Groups</td>
</tr>
<tr>
<td>4.10.33</td>
<td>Get List of Available CDN Edge Groups</td>
</tr>
<tr>
<td>4.10.34</td>
<td>Get CDN Edge Group Details</td>
</tr>
<tr>
<td>4.10.35</td>
<td>Add CDN Edge Group</td>
</tr>
<tr>
<td>4.10.36</td>
<td>Edit Edge Group</td>
</tr>
<tr>
<td>4.10.37</td>
<td>Delete Edge Group</td>
</tr>
<tr>
<td>4.10.38</td>
<td>Export Edge Group</td>
</tr>
<tr>
<td>4.10.39</td>
<td>Assign Location to CDN Edge Group</td>
</tr>
<tr>
<td>4.10.40</td>
<td>Unassign Location From CDN Edge Group</td>
</tr>
<tr>
<td>4.10.41</td>
<td>Modify CDN Edge Group</td>
</tr>
<tr>
<td>4.10.42</td>
<td>Search CDN Edge Groups</td>
</tr>
<tr>
<td>4.10.43</td>
<td>CDN Edge Servers API</td>
</tr>
<tr>
<td>4.10.44</td>
<td>Get List of CDN Edge Servers</td>
</tr>
<tr>
<td>4.10.45</td>
<td>Get CDN Edge Server Details</td>
</tr>
<tr>
<td>4.10.46</td>
<td>Add CDN Edge Server</td>
</tr>
<tr>
<td>4.10.47</td>
<td>Edit CDN Edge Servers</td>
</tr>
<tr>
<td>4.10.48</td>
<td>Reboot CDN Edge Server</td>
</tr>
<tr>
<td>4.10.49</td>
<td>Reboot CDN Edge Server in Recovery</td>
</tr>
<tr>
<td>4.10.50</td>
<td>Start up CDN Edge Server</td>
</tr>
<tr>
<td>4.10.51</td>
<td>Shut down CDN Edge Server</td>
</tr>
<tr>
<td>4.10.52</td>
<td>Stop CDN Edge Server</td>
</tr>
<tr>
<td>4.10.53</td>
<td>Rebuild CDN Edge Server</td>
</tr>
<tr>
<td>4.10.54</td>
<td>Suspend CDN Edge Server</td>
</tr>
<tr>
<td>4.10.55</td>
<td>Rerun CDN Edge Server Creation Scripts</td>
</tr>
<tr>
<td>4.10.56</td>
<td>Unlock CDN Edge Server</td>
</tr>
<tr>
<td>4.10.57</td>
<td>Delete CDN Edge Servers</td>
</tr>
<tr>
<td>4.10.58</td>
<td>Migrate CDN Edge Server</td>
</tr>
<tr>
<td>4.10.59</td>
<td>Segregate CDN Edge Server</td>
</tr>
<tr>
<td>4.10.60</td>
<td>Change CDN Edge Server Owner</td>
</tr>
<tr>
<td>4.10.61</td>
<td>Set VIP Status for CDN Edge Server</td>
</tr>
<tr>
<td>4.10.62</td>
<td>Add/Edit Admin/User Note for CDN Edge Server</td>
</tr>
<tr>
<td>4.10.63</td>
<td>CDN Edge Server Disks</td>
</tr>
<tr>
<td>4.10.64</td>
<td>CDN Edge Server Network Interfaces</td>
</tr>
<tr>
<td>4.10.65</td>
<td>CDN Edge Server IP Address Joins</td>
</tr>
<tr>
<td>4.10.66</td>
<td>Rebuild Network for CDN Edge Server</td>
</tr>
<tr>
<td>4.10.67</td>
<td>Get CDN Edge Server Billing Statistics</td>
</tr>
<tr>
<td>4.10.68</td>
<td>Get CDN Edge Server CPU Usage Statistics</td>
</tr>
<tr>
<td>4.10.69</td>
<td>Search CDN Edge Server by Label</td>
</tr>
</tbody>
</table>
4.11 CDN HTTP Caching Rules API ............................................................... 290
4.11.1 Get List of HTTP Caching Rules ......................................................... 291
4.11.2 Add HTTP Caching Rule ................................................................. 292
4.11.3 Edit HTTP Caching Rule ................................................................. 293
4.11.4 Delete HTTP Caching Rule .............................................................. 294
4.12 CDN Locations API ............................................................................. 294
4.12.1 Get List of CDN Locations for Location Group .................................... 294
4.12.2 Get List of CDN Locations ............................................................... 295
4.12.3 Detach CDN Location from Location Group ....................................... 296
4.12.4 Attach CDN Location to Location Group .......................................... 297
4.13 CDN Reporting API ............................................................................. 297
4.13.1 Get Top Files Report ........................................................................ 298
4.13.2 Purge CDN Resource in Top Files Report ......................................... 299
4.13.3 Get Top Referrers Report ................................................................. 299
4.13.4 Get CDN Overview Report ............................................................... 300
4.13.5 Get Cache Statistics Report ............................................................. 302
4.13.6 Get Status Codes Report ................................................................. 303
4.13.7 Get Stream Bandwidth Report .......................................................... 305
4.13.8 View CDN Resource Concurrent Statistics ....................................... 307
4.13.9 Get Visitors Report .......................................................................... 308
4.13.10 Get Admin Report ......................................................................... 309
4.14 CDN Resources API ............................................................................ 313
4.14.1 Get List of CDN Resources ............................................................... 313
4.14.2 Get CDN Resource Basic Details ...................................................... 315
4.14.3 Get CDN Resource Advanced Details .............................................. 317
4.14.4 Add HTTP CDN Resource ............................................................... 319
4.14.5 Add Video on Demand CDN Resource ............................................ 320
4.14.6 Add Live Streaming CDN Resource .................................................. 322
4.14.7 Add HTTP Pull CDN Resource with Advanced Settings ................. 323
4.14.8 Add HTTP Push CDN Resource with Advanced Settings ................ 328
4.14.9 Add VoD Push CDN Resource With Advanced Settings .................. 331
4.14.10 Add VoD Pull CDN Resource With Advanced Settings ................. 333
4.14.11 Add Live Streaming CDN Resource with Advanced Settings .......... 335
4.14.12 Edit Resource ................................................................................. 337
4.14.13 Edit HTTP Pull CDN Resource with Advanced Settings ................. 338
4.14.15 Edit VoD Push CDN Resource with Advanced Settings ................. 346
4.14.16 Edit VoD Pull CDN Resource with Advanced Settings ................... 347
4.14.18 Delete CDN Resource ................................................................. 351
4.14.19 Change CDN Resource FTP Password .......................................... 352
4.14.20 Prefetch CDN Resource Content .................................................... 352
4.14.21 Purge CDN Resource Content ....................................................... 353
4.14.22 View CDN Resource Bandwidth Statistics ..................................... 355
4.14.27 Search CDN Resource ................................................................. 364
4.14.28 Suspend CDN Resource ............................................................... 365
4.14.29 Resume CDN Resource ............................................................... 366
4.14.30 View CDN Resource Advanced Reporting ..................................... 366
4.14.31 Get List of Available Storage Locations ......................................... 368
4.14.32 Get List of Edge Server IP Ranges .................................................. 369
4.15 CDN SSL Certificates API ................................................................. 369
4.15.1 Get List of Custom SNI SSL Certificates ......................................... 370
4.15.2 Get Custom SNI SSL Certificate Details ......................................... 371
4.15.3 Add Custom SNI SSL Certificate ...................................................... 373
4.15.4 Edit SNI SSL Certificate ................................................................. 375
4.15.5 Delete SNI SSL Certificate .............................................................. 377
4.15.6 Add Custom SNI SSL Certificate to CDN Resource ...................... 378
4.15.7 Search Custom SNI SSL Certificates by Name ............................... 378
4.16 CDN Storage Servers API ................................................................. 379
4.16.1 Get List of CDN Storage Servers ..................................................... 379
4.16.2 Get List of HTTP Storage Servers ................................................... 382
4.16.3 Get List of Streaming Storage Servers ............................................ 383
4.16.4 Get CDN Storage Server Details ......................................................... 383
4.16.5 Add CDN Storage Server ............................................................... 385
4.16.6 Edit Storage Servers ...................................................................... 386
4.16.7 Reboot CDN Storage Server .......................................................... 387
4.16.8 Start up CDN Storage Server ......................................................... 387
4.16.9 Shut down CDN Storage Server ..................................................... 388
4.16.10 Stop CDN Storage Server ......................................................... 388
4.16.11 Rebuild CDN Storage Server ...................................................... 388
4.16.12 Suspend CDN Storage Server ...................................................... 389
4.16.13 Unlock CDN Storage Server ....................................................... 389
4.16.14 Set VIP Status for CDN Storage Server .......................................... 390
4.16.15 Delete CDN Storage Servers ....................................................... 390
4.16.16 Migrate CDN Storage Server ....................................................... 391
4.16.17 Segregate CDN Storage Server ................................................... 391
4.16.18 Resize CDN Storage Server ........................................................ 392
4.16.19 Change CDN Storage Server Owner ............................................ 392
4.16.20 Rebuild Network for CDN Storage Server ...................................... 392
4.16.21 CDN Storage Server Disks .......................................................... 393
4.16.22 CDN Storage Server Network Interfaces ....................................... 393
4.16.23 CDN Storage Server IP Address Joins .......................................... 397
4.16.24 Get CDN Storage Server CPU Usage Statistics .............................. 399
4.16.25 Get CDN Storage Server Hourly Statistics ................................... 400
4.16.26 Get CDN Storage Server Billing Statistics .................................... 401
4.16.27 Search CDN Storage Server by Label ......................................... 406
4.16.28 Get List of CDN Storage Server Backups ..................................... 406
4.16.29 Add/Edit Admin/User Note for CDN Storage Server ........................ 408
4.17 CDN Usage Statistics API ............................................................... 409
4.18 DNS Setup API .................................................................................. 410
4.18.1 Get DNS Domain Details ............................................................... 410
4.18.2 Set Up DNS Domain ..................................................................... 410
4.18.3 Edit Domain .................................................................................. 411
4.18.4 Get List of Glue Records ................................................................. 411
4.19 DNS Zones API .................................................................................. 412
4.19.1 Get List of Own DNS Zones .......................................................... 412
4.19.2 Get Domain Zone Details ............................................................... 413
4.19.3 Get List of Users DNS Zones ......................................................... 413
4.19.4 Add DNS Zone ............................................................................. 414
4.19.5 Delete Zone ................................................................................... 415
4.19.6 Get List of Name Servers .............................................................. 415
4.19.7 Get List of DNS Zone Records ....................................................... 416
4.19.8 Get DNS Record Details .............................................................. 420
4.19.9 Add DNS Record .......................................................................... 421
4.19.10 Edit DNS Records ...................................................................... 422
4.19.11 Delete DNS Record ..................................................................... 424
4.19.12 Search for DNS Zone ................................................................. 424
4.20 Wildcard Invalidation Rules API ....................................................... 425
4.20.1 Get List of Wildcard Invalidation Rules ........................................... 425
4.20.2 Add Wildcard Invalidation Rules .................................................... 426
4.20.3 Invalidate Files Again ................................................................. 426
4.20.4 Delete Wildcard Invalidation Rules ................................................ 427
4.21 Web Application Firewall (WAF) API ............................................... 428
4.21.1 Check Status of WAF Protection .................................................. 428
4.21.2 Enable/Disable WAF Protection .................................................... 429
4.22 Get CDN API Status ........................................................................ 430
OnApp CDN is a unique approach to CDN for hosting providers. It's a federated CDN platform that uses spare capacity in OnApp Clouds to provide a global network of low-cost, high-performance CDN PoPs (Point of Presence).

There are three main elements to OnApp CDN:

- **OnApp CDN Stack** – edge server software that installs in your cloud, in much the same way as a virtual server. This lets you create your own CDN Point of Presence to cache and distribute web content to end users.

- **OnApp CDNaaS** – this is an Anycast DNS redirection service that directs content requests from end users to the most appropriate PoP. The service is hosted by OnApp at datacenters around the world.

- **OnApp CDN Federation** – a marketplace where you can buy CDN bandwidth from other hosts using OnApp CDN. This lets you build a global CDN service without having to build global CDN infrastructure. You can also submit your own edge servers to the marketplace and (if your servers are accepted) sell bandwidth to other hosts.
1 Get Started

OnApp CDN is a unique approach to CDN for hosting providers. It is a federated CDN platform that uses spare capacity in OnApp Clouds to provide a global network of low-cost and high-performance CDN PoPs (Point of Presence).

There are three main elements to OnApp CDN:

• **OnApp CDN Stack** – edge server software that is installed in your cloud in much the same way as a virtual server. This enables you to create your own CDN Point of Presence to cache and distribute web content to end users.

• **OnApp CDNaaS** – this is an Anycast DNS redirection service that directs content requests from end users to the most appropriate PoP. The service is hosted by OnApp at data centers around the world.

• **OnApp CDN Federation** – a marketplace where you can buy CDN bandwidth from other hosts using OnApp CDN. This allows you to build a global CDN service without having to build global CDN infrastructure. You can also submit your own edge servers to the marketplace and (if your servers are accepted) sell bandwidth to other hosts.

This guide contains information on how to setup CDN in OnApp. For detailed information, refer to one of the following sections:

• [OnApp Dashboard](#)
• [How your CDN Account Works](#)
• [Hardware & Marketplace Requirements](#)
• [OnApp CDN Terms of Use](#)
• [Types of CDN Deployment](#)
• [Set up CDNs Using Marketplace Servers Only](#)
• [Set up CDNs Combining Local & Marketplace Resources](#)
• [CDN Activation & Configuration Process](#)
• [Next Steps](#)

For information on OnApp installation and upgrade procedures, refer to the [Installation](#) or the [Upgrade](#) guide.
1.1 OnApp Dashboard

OnApp CDN is managed in two places:

- **The OnApp Control Panel** is where you deploy edge servers, manage users and set pricing for your edge server resources.
- **The OnApp Dashboard** is where you manage your CDN account and software licenses, buy bandwidth on the CDN marketplace, set prices for bandwidth you sell on the marketplace, and access reporting and other CDN tools.

You can access the Dashboard at [http://dashboard.onapp.com](http://dashboard.onapp.com). You'll need a username and password to log in.
If you don't have login credentials or have problems accessing Dashboard, contact support@onapp.com.

1.2 How your CDN Account Works

OnApp acts as a clearing house for CDN transactions, handling all payments and charges for the CDN.

Since June 20, 2019, all new CDN customers will be subjected to the new pricing plan. For the details, refer to [https://onapp.com/pricing-cdn/](https://onapp.com/pricing-cdn/)

Your CDN account is managed through the OnApp Dashboard. This is how it works:

- **You deposit credit in your CDN account.**
  You can add credit and view your account balance at any time through the OnApp Dashboard: [http://dashboard.onapp.com](http://dashboard.onapp.com).

- **OnApp automatically credits your account with revenue you earn from the marketplace.**
  If you have edge servers on the CDN marketplace, any revenue you make from them, by selling CDN bandwidth, is automatically added to your account.
OnApp automatically deducts CDN usage charges from your account, including:

- **The cost of any bandwidth you buy on the marketplace:**
  The amount is set by the owners of the marketplace servers you subscribe to. $x.xx per Gigabyte
- **The fee for using the core OnApp CDN service (OnApp CDNaaS):**
  This fee applies to all traffic routed to your end users through OnApp CDNaaS, whether it's from your own edge servers or marketplace servers. $5 per Terabyte of traffic routed via CDNaaS
- **The fee for selling bandwidth on the OnApp CDN marketplace:**
  This fee is only levied on the value of bandwidth you sell on the OnApp CDN marketplace. 10% of the marketplace value of any bandwidth you sell

If your CDN revenue is greater than your usage costs, OnApp pays you the difference.

If the value of bandwidth you sell is greater than the cost of the bandwidth you buy + OnApp CDN usage charges, we'll send you a check and deduct that amount from your CDN account balance. The calculation and any payments due are processed monthly. For details on payment withdrawal, contact us at finance@onapp.com.

### 1.3 Hardware & Marketplace Requirements

This page contains the hardware requirements, CDN marketplace requirements, and the recommended network configuration for OnApp CDN.

#### 1.3.1 Hardware Requirements

These are the minimum recommended specs for OnApp CDN controller, edge, and storage servers:

<table>
<thead>
<tr>
<th>OnApp Controller Server</th>
<th>Compute Resource Servers</th>
<th>Backup Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Dual or Quad Core 2 Ghz+ CPU</td>
<td>• 8 GB RAM&lt;br&gt;- Quad Core 2 Ghz+&lt;br&gt;- 200 GB HD (SSD recommended) and no-raid setup&lt;br&gt;- 3x Gig NIC (4 recommended)&lt;br&gt;- CentOS 7 (x64)</td>
<td>• 4GB RAM (8 GB+ recommended)&lt;br&gt;- Dual or Quad Core 2 Ghz+&lt;br&gt;- 2 TB Storage mounted locally&lt;br&gt;- 2x Gig NIC&lt;br&gt;- CentOS 7 (x64)</td>
</tr>
<tr>
<td>• 8 GB+ RAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 100 GB RAID 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• 2x Gig network interface cards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• CentOS 7 (x64)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Also below you can find hardware requirements for:

<table>
<thead>
<tr>
<th>Accelerator</th>
<th>Edge and Storage Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Minimum requirements: 4 cores, 4 GB of RAM, and 100 GB of storage&lt;br&gt;• Recommended parameters: 8 cores, 16 GB of RAM, and 1 TB of storage&lt;br&gt;• SSD is recommended to avoid slowing down access</td>
<td>• Minimum requirements: 4 cores, 4 GB RAM, and 100 GB of disk space (SSD recommended)&lt;br&gt;• Recommended parameters: 8 cores, 16 GB RAM, and 1 TB of disk space (SSD recommended)</td>
</tr>
</tbody>
</table>
1.3.2 CDN Marketplace Requirements

Edge servers submitted to the marketplace are assessed on a case-by-case basis before they are accepted. This helps us to ensure effective CDN performance for hosting providers and end users.

CDN storage server is not available in the marketplace (only HTTP and streaming edge servers are allowed).

The assessment process includes:

**Benchmarking**

We benchmark your PoP via SSH. It must meet these minimum requirements:

- Max marketplace price of $0.05/GB (EU/US) or $0.20/GB (rest of world)
- Two or more dedicated edge servers with the same location, but on different compute resources (for redundancy)

Minimum edge server hardware/network/storage specs:

<table>
<thead>
<tr>
<th></th>
<th>HTTP</th>
<th>Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>Quad Core</td>
<td>Quad Core</td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>8 GB</td>
<td>8 GB</td>
</tr>
<tr>
<td><strong>Sequential Read</strong></td>
<td>45 MB/s</td>
<td>45 MB/s</td>
</tr>
<tr>
<td><strong>Port Speed</strong></td>
<td>N.America/EU</td>
<td>Other</td>
</tr>
<tr>
<td></td>
<td>1 Gbps</td>
<td>100 Mbps</td>
</tr>
<tr>
<td><strong>RAID</strong></td>
<td>None / RAID-0</td>
<td>None / RAID-0</td>
</tr>
<tr>
<td><strong>IPs</strong></td>
<td>3 Public IPs</td>
<td>1 Public IP</td>
</tr>
<tr>
<td><strong>Disk Space</strong></td>
<td>400 GB*</td>
<td>1 TB**</td>
</tr>
<tr>
<td><strong>IOPS</strong></td>
<td>10,000*</td>
<td>130**</td>
</tr>
</tbody>
</table>

* this configuration provides you with a small disk and good IOPS performance
** this configuration provides you with a large disk and lower IOPS performance

**Location/Existing Coverage Assessment**

Next, we consider your PoP's physical location. Our aim is to ensure broad CDN coverage while preventing the saturation of edge servers in a given location.

- A PoP in a new location is likely to be accepted as long as it passes the benchmark process.
- In a location with many existing PoPs, your server may not be accepted unless it has unusually good performance characteristics.

**Benchmark + Location = Decision!**

If your PoP is accepted, it is available immediately on the marketplace.

- If your PoP is not accepted for performance reasons, we'll raise a ticket for resolution with you, and keep it in view for reassessment in the future.
- If location issues prevented your PoP from being accepted, we'll keep it in view for reassessment in the future.
1.3.3 Recommended Network Configuration

To use OnApp CDN you must agree and abide by the OnApp CDN Acceptable Use policy, and OnApp CDN General Use Terms and Conditions policy, which you can find at Legal Documentation.

1.4 OnApp CDN Terms of Use

There are two main ways to get up and running with OnApp CDN. You can build a CDN using your own physical hardware, and add locations from our global CDN marketplace; or you can build a CDN entirely from marketplace resources.

If you are an existing OnApp Cloud user and have a Control Panel server installed, you do not have to perform any server setup procedures. Just proceed with CDN Activation & Configuration Process steps.

1.5 Types of CDN Deployment

1.5.1 CDN with Marketplace Resources Only

This is the simplest way to configure OnApp CDN, since it involves minimal hardware. In essence you’re building a virtual CDN. You don’t host your own physical edge servers: instead, you subscribe to locations on the OnApp CDN marketplace, purchase CDN bandwidth from other providers, and resell it to your customers. Obviously, without your own edge servers you won’t be able to provide local edge resources to your customers, or sell edge server resources on the marketplace.

Basic hardware required:
1.5.2 Own Edge Servers with Marketplace Resources

To provision your own local edge server resources you will need to create a small OnApp Cloud. Existing OnApp Cloud users can deploy edge servers in their existing cloud.

The edge server is a virtual appliance that is deployed and managed in much the same way as a virtual machine. They are hosted on compute resource(s) and will need their own primary storage volume(s).

**Basic hardware required:**

<table>
<thead>
<tr>
<th>1x Control Panel server</th>
<th>This hosts the OnApp Control Panel, where you manage your local edge servers, marketplace resources, users, billing and so on.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x compute resource server</td>
<td>This hosts the edge server virtual appliance and provides its physical CPU and storage resources. Any edge servers you deploy in your cloud can be used by any of your Control Panel Servers.</td>
</tr>
</tbody>
</table>

At this point it might be a good idea to consider any future cloud plans, as deploying your CDN is basically the same as deploying a small cloud. Check out the [OnApp Cloud Preparation Guide](#) for more information.

1.6 Set up CDNs Using Marketplace Servers Only

This is the simplest way to deploy OnApp CDN, since it only needs an OnApp Control Panel Server. You don't host your own edge servers: you build a virtual CDN entirely from marketplace resources.

To set up CDN using marketplace servers only:

2. **Activate and configure your CDN**

   - You don't have to subscribe to your own edge PoPs. You will be automatically subscribed to any PoPs you own.
   - If you are an existing OnApp Cloud user and have a Control Panel server installed, you do not have to perform any server setup procedures. Just proceed with [CDN Activation & Configuration Process](#) steps.
1.7 Set up CDNs Combining Local & Marketplace Resources

To create a CDN with local edge servers as well as marketplace resources, you need an OnApp Controller Server and at least one compute resource to host the edge server virtual appliance included with OnApp CDN.

Follow the steps in our Installation Workflow to set up CDNs Combining Local & Marketplace Resources.

1.8 CDN Activation & Configuration Process

Activating and setting up your CDN is a fairly straightforward process. The steps are the following:

1. Enable CDN for Your Cloud
2. Set customer and admin permissions for CDN
3. Run CDN Setup Wizard
4. Create or Subscribe to CDN Edge Servers
5. Create your own CDN storage servers
6. Create CDN Edge Groups and assign edges to them
7. Assign CDN Edge Groups to a bucket, and set prices for CDN bandwidth

For more information on each step, refer to the linked sections.

1.8.1 Enable CDN for Your Cloud

If you want to enable CDN for your cloud, you should contact your account manager. This person will create a CDN subscription for you and link the subscription in the Dashboard.

The CDN marketplace allows you to subscribe to CDN locations provided by other hosts. It also enables you to sell CDN bandwidth from your own edge servers to other hosts. Please note that all servers submitted to the marketplace, in order to sell bandwidth, are assessed before they are accepted. Criteria include geographic location, bandwidth and server specs.

You are limited to one license per city - if you have two clouds very close to the same city, please choose a different city which is nearby.

Setting prices for edge server resources you sell to end users (rather than other marketplace members) is handled through the OnApp control panel, via CDN Edge Groups and Billing Groups.

Once the CDN setup process is completed, you will see new CDN tools in the main left-hand navigation of your OnApp Control Panel.

You should have at least one configured Location Group to be able to create different CDN servers (edge servers, storage servers, accelerators etc.).
1.8.2 Set Customer and Admin Permissions for CDN

You must set correct CDN permissions for customers and administrators in your OnApp Control Panel, otherwise, users won't be able to purchase bandwidth or manage CDN resources.

**Customer/End User Roles**

1. Go to your Control Panel > **Admin > Roles** menu.
2. To edit a customer/end user role, click the **Actions** button next to the role you want to change, then click **Edit**.
3. Select **CDN Resources** in the permissions list and enable the following permissions:
   - Create a new CDN resource (cdn_resources.create)
   - Destroy own CDN resources (cdn_resources.delete.own)
   - See own CDN resources (cdn_resources.read.own)
   - Update own CDN resources (cdn_resources.update.own)
4. Click the **Save** button.

**Administrator Roles**
1. Go to your Control Panel > Admin > Roles menu.

2. To edit an administrator role, click the Actions button next to the role you want to change, then click Edit.

3. Enable the following permissions:
   - CDN Resources
     Any action on CDN resources
   - Edge Groups
     Any action on Edge Groups
   - Edge Servers
     Any action on Edge Servers

4. Click the Save button.

### 1.8.3 Launch CDN Setup Wizard

To launch the CDN setup wizard, users should have the following permissions enabled:

- Update any Role
- See all Roles
- Create a new edge group

The CDN setup cycle consists of three steps that involve configuring:

1. Permissions
2. Edge Groups
3. Billing

You can rerun the wizard after the initial configuration as many times as you need.

To start the CDN wizard:

1. Go to the CDN > Edge Servers menu in your Control Panel.
2. Click the CDN Setup Wizard button.
3. Complete the steps in the wizard as described below.
1.8.3.1 Step 1 Permissions

- **Client role** - select a user role to enable the CDN-related permissions listed in the *Enables Permissions* section. When you complete the wizard, these CDN permissions will be automatically applied to the selected role. You can also enable CDN-related permissions for additional user roles via the Roles menu.

- **Admin role** - select the Administrator role to enable the CDN-related permissions listed in the *Enables Permissions* section. For the Administrator role, any action on CDN resources is enabled. When you complete the wizard, these CDN permissions will be automatically applied to the Administrator role.

- Do not assign the Administrator permissions to a client-shared role.
- You may skip the **Step 1 Permissions** if you have already set CDN-related permissions for the user and administrator roles.
- Users will not be able to purchase and manage CDN resources unless CDN permissions are enabled for their role.

Click the **Next** button to proceed to the next step.

1.8.3.2 Step 2 Edge Groups
- **Label** - enter a name for your edge group. For example, you can create the edge group called "North America" and add to it your North American PoPs.

- **Assigned Locations** - the list of locations assigned to the edge group.

- **Available Locations** - the list of available locations that you can assign to the edge group. To add a location, click the '+' button next to the location you want to assign to the group.

Edge groups can include your own servers and servers from the CDN Marketplace. You can assign additional edge groups later in the **Bucket**.

Click the **Next** button to proceed to the next step.

1.8.3.3 Step 3 Billing

- **Bucket** - assign the CDN edge group to a bucket.

- **Price per GB** - set a price per GB for traffic your users push through all servers in this edge group.

You can assign more edge groups to a bucket and set different prices for them, using the **Admin > Buckets** menu. Any user assigned to the destination bucket will be able to create a CDN service that includes the edge group’s locations.

When you are finished configuring billing for CDN, click the **Create Edge Group** button. On the page that loads, you will see a summary on the created edge group. You can click the **CDN Dashboard** button to access your Dashboard.

After CDN is set up, synchronization between CDN and OnApp is run every 20 minutes by default (value can be changed at the **Edit Infrastructure Configuration** page). If synchronization fails because of CDN Sync Runner issues, you will receive the notification. To solve this issue check CDN Sync Runner status via **Sysadmin**. The CDN section in
Sysadmin enables you to check the status of CDN API and see the last time CDN statistics were gathered.

1.8.4 Create and Subscribe to CDN Edge Servers

Your CDN needs edge servers to cache and deliver content. You can set up your own edge servers in your cloud, subscribe to edge servers on the CDN marketplace, or do both.

- **Add CDN Edge Server to Your Cloud**
- **Subscribe to CDN Marketplace Locations**

1.8.4.1 Add CDN Edge Server to Your Cloud

To add a CDN edge server to the cloud:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. On the screen that appears, click the Create Edge Server button or press the "+" button.
3. Fill in the edge server creation form step by step:

   **Step 1 of 4**
   The Cloud Locations step applies to those users who have compute zones assigned to
location groups in their billing plan. Indicate your edge server’s cloud location: country and city. Click Next.

**Step 2 of 4**

- Give your edge server a label. The label can consist of:
  
  Lower- & upper-case letters [A-Za-z]
  Digits [0-9]
  Dash [-]
  Underscore [ _ ]
  Space character [ ]
  At sign [@]
  Brackets [ (){} ]
  Slashes [/]
  Caret [ ^ ]
  Dollar sign [$]
  Asterisk [*]
  Comma [,]
  Dot [.]

- Select an edge server type: HTTP or streaming.

  **PLEASE NOTE:** When the first streaming edge server is created, a Wowza key will be assigned to the operator. The operator will be then billed monthly depending on the number of instances of that particular license key being used in that month.

- Choose a compute zone to build this server on.

- Choose a specific compute resources to build this server on.

- Move the **Add to Marketplace** slider to the right to submit this server to the OnApp CDN marketplace. If so, the minimum required disk size is 1 TB.

- Click Next.

Any servers you submit will be assessed before they are accepted into the marketplace. Criteria include geographic location, bandwidth and server specs.

**Step 3 of 4**

- Set the resources needed for this edge server: RAM, CPU cores and CPU priority.

- Choose a data store zone for this edge server’s primary disk.

- Set the primary disk size.

  The disk size is calculated in the following way: 10 GB for OS, the rest of total disk space is estimated 80% per Pull population and 20% per Push population.
Choose a network zone and set the port speed for this edge server, or make it unlimited. It is not possible to set port speed value for edge servers based on smart compute resources.

Step 4

- Move the **Build Edge Server Automatically** slider to the right if you want the system to automatically build the edge server. Otherwise, you will have to build your server manually after it is created.

4. Click **Create Edge Server** button.

- Open all available ports to the CDN edge server if you are using firewall rules, since edge servers do not support the firewall configuration.
- Edge servers are managed in much the same way as a virtual machine. The exception is that autoscaling, backups, NAT and firewall rules are not available for edge servers. For a full description of edge server management tools, see the CDN sections of the OnApp Cloud Administration Guide.

1.8.4.2  **Subscribe to CDN Marketplace Locations**

To subscribe to a CDN marketplace location:

2. Click the **CDN** link in the main Dashboard navigation.
3. Select the **Marketplace** tab on the main CDN dashboard screen, then choose **Subscribe POPs**.
4. Use **Locations**, **Provider** and **Price** tabs above the map to search available edge servers by location/provider/price.
5. In each case, the providers are shown in a list along with their details and price per GB. Choose one or more providers by checking the **Purchase** box next to their entry in the list.
6. Click the **OK** button at the bottom of the screen to confirm.

**1.8.5 Create CDN Storage Servers**

To create a new storage server:
1. Go to your Control Panel > CDN > Storage Servers menu.
2. Click the Create Storage Server button at the bottom of the screen.
3. Fill in the storage server creation form step by step:

**Step 1 of 4. Cloud Locations**
The Cloud Locations step applies to those users who have compute zones assigned to location groups in their billing plan. Indicate your storage server's cloud location: country and city. Click Next.

**Step 2 of 4. Properties**
Specify the storage server details:

- Specify the server's label in a human-recognizable format.
- Select the storage server type: HTTP or Streaming.

A third party application - Wowza will be installed automatically when installing a streaming storage server and additional charges will apply. Please, contact your account manager for details.

- Specify the compute resource and compute zone.
- Click Next.

**Step 3 of 4. Resources**

- Set the resources needed for this storage server: RAM, CPU cores and CPU priority.

The minimum memory capacity is 8 GB.

- Choose a data store zone for this storage server’s primary disk.
- Set the primary disk size (Storage server HDD). The minimum required disk size is 30 GB.
- Choose a network zone from the drop-down box.
- If the option is available, you can also assign an IP address for the VS from the drop-down menu. Indicate compute resource and network to have the list of available IPs. Tick the Show Only My IP Addresses check box to view only own IP addresses in the IP addresses drop box.
- Set the port speed in Mbps or tick it as unlimited. It is not possible to set port speed value for storage servers based on smart compute resource.
- Click Next.

**Step 4. Confirmation**
On the screen that appears, tick the **Build Edge Server automatically** box to build the storage server automatically, otherwise you will have to build your storage server manually after it is created.

- Click the **Create Storage Server** button to start the creation process.

### 1.8.6 Create CDN Edge Groups & Assign Edge Servers to Them

Once you have created your own CDN edge servers, and/or subscribed to CDN marketplace locations, you can use your OnApp Control Panel to bundle them into CDN Edge Groups. CDN Edge Groups normally include edge servers in a specific region – North America, for example. They can contain a mix of your own servers and marketplace servers, as required.

To create an edge group and assign edge servers to it:

1. Go to your Control Panel > **CDN > Edge Groups** menu.
2. Click the **Create Edge Group** button.
3. On the screen that appears, give your new group a label and click the **Create Edge Group** button.
4. You will be redirected to a screen where you can assign edge servers ('locations') to the group.
5. On the screen that appears, you'll see the list of locations divided into two sections:
   - **Assigned Locations** (your own edge servers and marketplace locations that are part of this group)
   - **Available Locations** (edge servers and marketplace locations not yet assigned to this group)

Using the + and – icons to add/remove locations to your CDN Edge Group.

For more information on how to manage CDN edge groups, refer to the **CDN Edge Groups** section.

### 1.8.7 Assign CDN Resources to Bucket and Set Prices

The final step is to assign your CDN edge group to a bucket and set prices for bandwidth sold to customers from this edge group.

For more information on how to manage buckets, refer to the **Buckets** section of the Administration Guide.
1.8.7.1 Configure Access Control

The access to CDN edge groups is configured in the Access Control section of the bucket. To give access to an edge group for users to whom the bucket is assigned:

1. Go to the Admin > Buckets menu on your Control Panel.
2. Click a label of a destination bucket and open the Other tab from the Access Control section.
3. In the Access Control, you can manage the following CDN-related resources:

- **Limits for Edge Groups**
  Click the Add New Edge Group (+) button in the Limits for Edge Groups box. In the box that appears, select an edge group that you want to assign to the bucket and click the Submit button. Select the Duplicate to rate card checkbox before clicking Submit to add the edge group to Rate Card of the bucket with the default price of 0. When the edge group is added to the Access Control section, users to whom the bucket is assigned will be able to use this edge group. Go to the Rate Card tab to set a price for the bandwidth available to users assigned to this bucket.

  If no edge groups are added to Access Control, users under the bucket will have access to none of the edge groups available on the system.

- **Limits for CDN Bandwidth**
  Click the Add Limits for CDN Bandwidth (+) button in the Limits for CDN Bandwidth box. In the box that appears, enter the maximum CDN bandwidth limit in GB per month and click the Submit button. The maximum CDN bandwidth limit is allocated per month for each user assigned to the bucket. If the user exceeds the maximum bandwidth limit allocated per month, the user account is auto-suspended and all CDN resources become unavailable. The user account will be automatically unsuspended the next calendar month.

  You can unsuspend the account of the user who exceeded the maximum bandwidth limit by increasing this limit in the bucket. Note that the increased limit will be applicable for all users assigned to the bucket.

  If the limit for CDN bandwidth is not set but edge groups are added in Access Control, the bandwidth will be unlimited for each user under this bucket.

  The price per GB of bandwidth is set for each edge group separately in the Rate Card > Pricing for Edge Groups section.

  If you want to disable prices completely you can do so by enabling the Disable billing slider at the Edit System Configuration page.
When the billing is disabled, the Rate Cards are removed from existing buckets. Note that if billing is enabled again, the prices won't be recalculated again. Instead, the price calculation will start with the next hour.

1.8.7.2 Configure Rate Card

The prices for CDN edge groups are configured in the Rate Card section of the bucket. To set a price for a CDN edge group:

1. Go to the Admin > Buckets menu on your Control Panel.
2. Click a label of a destination bucket and open the Other tab from the Rate Card section.
3. Click the Add New Edge Group (+) button in the Pricing for Edge Groups box.

If you have duplicated the edge group to Rate Card while adding it to Access Control, the edge group will already be present here but with the default price that is 0.

4. In the box that appears, select a target edge group, set the price per GB of bandwidth and click the Submit button.

If you only add a price for the edge group in Rate Card but not add the edge group to Access Control, users under the bucket will not have access to the edge group. To give users under the bucket access to the edge group, you need to add the group to Access Control.

When removing an edge group from Rate Card, note that the prices will be set to zero for all CDN resources using this edge group.

1.9 Next Steps

Once you have completed the CDN activation and set-up process, you are ready to start selling CDN services to your customers. Enabling your customers to distribute content via CDN is also handled through your OnApp Control Panel, via the CDN > Resources menu. The CDN > Resources menu appears once you have set up at least one CDN Edge Group, containing at least one edge server or marketplace location, and assigned this edge group to the user's bucket. The CDN resource is basically a customer web server that will use CDN to cache and distribute content.
For details on how to manage CDN resources, refer to the Administration Guide.

1.10 What's New

1.10.1 OnApp 5.9

- Added a possibility to delete a CDN edge group locally in case its remote ID is empty.

1.10.2 OnApp 5.8

- Added a possibility to limit CDN bandwidth allocated within edge groups that are added to the bucket. The maximum CDN bandwidth limit is allocated per month for each user assigned to the bucket.

- Added a page that provides a list of IP ranges available for edge servers to which you are subscribed on CDN Marketplace. You can use IP ranges to filter incoming traffic on your origin website by means of firewall rules.

1.10.3 OnApp 5.7

- Web Application Firewall (WAF) secures your CDN resources from malicious threats and attacks. WAF monitors, filters or blocks the traffic to and from a Web application. It can prevent such attacks as Cross-site scripting (XSS), SQL injection, session hijacking and buffer overflows through customizable web security rules.

- Wildcard invalidation rules allow you to remove certain folders or files from the cache. You can launch the invalidation process multiple times for a rule you specify. Wildcard invalidation is available for HTTP Pull and HTTP Push resources and you can set up to five wildcard invalidation rules for a resource.
2 Administration Guide

OnApp CDN is a software product that works with OnApp Cloud. If you enable CDN for your end users, they can get access to a network of global edge servers (yours, and/or those provided by other cloud hosts) and use this network to deliver web content more quickly and reliably to visitors all over the world. Your CDN is managed alongside your cloud via the OnApp Control Panel. For more information, see OnApp CDN for Service Providers.

OnApp provides the following types:

- HTTP Push
- HTTP Pull
- Live Streaming
- On-Demand Streaming

HTTP Push
The HTTP Push method is similar to secondary server: the user uploads content to CDN and links to it, so the content is physically stored at CDN storage servers.

HTTP Pull
With the HTTP Pull CDN, the website owner stores content on their server and rewrites URLs to include a subdomain. Then, when the specific content is requested, it is pulled to the CDN network from the host and delivered to the closest point to the consumer who requested that content (the content is cached on edge server).

Streaming
There are two supported streaming types in the OnApp CDN – live streaming and on-demand streaming. To utilize CDN streaming service, you have to deploy CDN streaming Edge Server.

Live Streaming
Live Streaming CDN allows delivering the content to end users in a live broadcast mode. When using a live streaming CDN service, end users receive media the same time like a traditional broadcasting.

Video On-Demand Streaming
Video On-Demand Streaming CDN allows delivering video on request and makes it repeatedly accessed. Consumers can control content and are able to fast forward or rewind it the same as live streaming.

Only mp4 and flv files are currently supported by VoD streaming.

Setting up OnApp CDN includes:

- Enabling CDN for your cloud in the OnApp customer dashboard (contact OnApp Support if you don’t have dashboard access)
- Running CDN Setup Wizard in your OnApp Control Panel
- Setting up storage servers
- Setting up edge servers
- Adding CDN edge groups and assigning them to buckets
- Creating CDN resources (when you create a CDN resource, CDN is enabled automatically in the OnApp)
- Assigning users to a bucket and setting their permissions
For more information on how to set up OnApp CDN, refer to the [Get Started](#) section of this guide.

When creating a user account, you need to grant the user all necessary permissions for managing CDN resources and assign the user to the bucket.

**Storage servers** store web content to be distributed over the Content Delivery Network. The content is then cached by edge servers and delivered to consumers.

**Edge servers** cache web content and deliver it to website visitors. They are deployed to compute resources and managed the same way as virtual servers. You can use edge servers to sell CDN bandwidth to your end users, and/or submit the edge server to the OnApp CDN marketplace and sell your bandwidth to other hosting providers. You can create as many edge servers as you need and place them to different compute resources in different geographical locations, and easily broaden your CDN by combining your own edge servers with other locations on the CDN marketplace. You can also build a CDN solely with marketplace resources. For details, refer to the [Edge servers](#) chapter.

**Edge groups** are groups of edge servers that can be your own and/or those you subscribe to from the CDN marketplace. They are usually grouped by location, so they represent a pool of servers for a given geographical area.

Edge groups are assigned [Buckets](#) to set the prices for the bandwidth that your end users consume. You can assign several groups to one bucket at a time, and establish different geographical zones with different pricing. The bandwidth pricing of the bucket is the price for CDN bandwidth sold to your end users.

**CDN resources** are specific servers with content that an end user wants to distribute via CDN. CDN resources are assigned to edge groups, which determine the list of servers taking part in distributing/caching of their data.

### 2.1 CDN Setup Wizard

To launch the CDN setup wizard, users should have the following permissions enabled:

- *Update any Role*
- *See all Roles*
- *Create a new edge group*

The CDN setup cycle consists of three steps that involve configuring:

1. **Permissions**
2. **Edge Groups**
3. **Billing**

You can rerun the wizard after the initial configuration as many times as you need.

To start the CDN wizard:

1. Go to the CDN > Edge Servers menu in your Control Panel.
2. Click the CDN Setup Wizard button.
3. Complete the steps in the wizard as described below.

On this page:

- **Step 1 Permissions**
- **Step 2 Edge Groups**
- **Step 3 Billing**

### 2.1.1 Step 1 Permissions

- **Client role** - select a user role to enable the CDN-related permissions listed in the *Enables Permissions* section. When you complete the wizard, these CDN permissions will be automatically applied to the selected role. You can also enable CDN-related permissions for additional user roles via the *Roles* menu.

- **Admin role** - select the Administrator role to enable the CDN-related permissions listed in the *Enables Permissions* section. For the Administrator role, any action on CDN resources is enabled. When you complete the wizard, these CDN permissions will be automatically applied to the Administrator role.

  - Do not assign the Administrator permissions to a client-shared role.
  - You may skip the **Step 1 Permissions** if you have already set CDN-related permissions for the user and administrator roles.
  - Users will not be able to purchase and manage CDN resources unless CDN permissions are enabled for their role.

Click the **Next** button to proceed to the next step.

### 2.1.2 Step 2 Edge Groups

- **Label** - enter a name for your edge group. For example, you can create the edge group called "North America" and add to it your North American PoPs.
- **Assigned Locations** - the list of locations assigned to the edge group.
- **Available Locations** - the list of available locations that you can assign to the edge group. To add a location, click the '+' button next to the location you want to assign to the group.
Edge groups can include your own servers and servers from the CDN Marketplace. You can assign additional edge groups later in the Bucket.

Click the Next button to proceed to the next step.

### 2.1.3 Step 3 Billing

- **Bucket** - assign the CDN edge group to a bucket.
- **Price per GB** - set a price per GB for traffic your users push through all servers in this edge group.

You can assign more edge groups to a bucket and set different prices for them, using the Admin > Buckets menu. Any user assigned to the destination bucket will be able to create a CDN service that includes the edge group’s locations.

You cannot add two edge groups with the same location to one bucket.

When you are finished configuring billing for CDN, click the Create Edge Group button. On the page that loads, you will see a summary on the created edge group. You can click the CDN Dashboard button to access your Dashboard.

After CDN is set up, synchronization between CDN and OnApp is run every 20 minutes by default (value can be changed at the Edit Infrastructure Configuration page). If synchronization fails because of CDN Sync Runner issues, you will receive the notification. To solve this issue check CDN Sync Runner status via Sysadmin. The CDN section in Sysadmin enables you to check the status of CDN API and see the last time CDN statistics were gathered.

### 2.2 CDN Edge Servers

Web content is cached in the network of edge servers on the CDN, distributed across different geographic locations. Currently, there are two types of edge servers in OnApp: HTTP and Streaming.

HTTP edge servers support both Push and Pull population methods in 80/20 ratio (80% HTTP Pull and 20% HTTP Push). When the edge server is created, its storage limit for HTTP Pull and HTTP Push is automatically assigned by the system.

Streaming edge server type allows sending a stream to one of the publishing points, or pick up the stream externally and deliver it to the end users. Take note that Operator has to deploy CDN streaming Edge Server to utilize CDN streaming service.

Streaming edge server support only streaming services. The following protocols are supported:
Streaming service includes the following advanced features:

- **Hotlinking protection** - protect your media from being hotlinked (linked to website without your permission)
- **Geo Blocking** - restrict access to your media so that it is accessible only for certain countries/regions
- **RMTPE (secure Wowza)** - streaming encryption

You do not have to add the Wowza license key manually to enable streaming edge servers. A third party application - Wowza will be installed automatically when installing an edge server and you will simply be charged for it. Please, contact your account manager for details.

Content is delivered to end users from the server which is closest to the user or has the best availability. If you have CDN enabled for your cloud, you can use the control panel to set up your own edge servers and manage them in the same way you manage virtual servers. You can submit your edge server to the locations in the marketplace to sell bandwidth across it.

To be able to sell CDN bandwidth through our online marketplace, you must first submit your edge server for assessment.

You can use smart Compute resources for CDN edge server deployment.

CDN edge servers do not support firewall and NAT (network address translation).

### 2.2.1 Create CDN Edge Server

To add new CDN edge server:

1. Go to your Control Panel → **CDN** → **Edge Servers** menu.
2. On the screen that appears, click the **Create Edge Server** button or press the "+" button.
3. Fill in the edge server creation form step by step:

On this page:
OnApp Cloud 6.5 Edge 5 CDN Guide

- **Step 1 of 4. Locations**
- **Step 2 of 4. Properties**
- **Step 3 of 4. Resources**
- **Step 4. Confirmation**

2.2.1.1  Step 1 of 4. Locations

The Cloud Locations step applies to those users who have **compute zones** assigned to **location groups** in their company billing plan. If the user's billing plan has several compute zones, some of which are assigned to location groups, whereas others are not - the cloud locations screen will not be available in the wizard. Also if there is only one location this step will be skipped. In this case the wizard will start with the Properties step. Indicate your edge server's cloud location:

- **Country** - choose the country, where the cloud is located, from the drop-down menu.
- **City** - specify the city, where the cloud is located, from the drop-down menu.

Click **Next** to proceed to the following step of the wizard to specify the edge server properties.

2.2.1.2  Step 2 of 4. Properties

- Give your edge server a label. The label can consist of:
  - Lower- & upper-case letters [A-Za-z]
  - Digits [0-9]
  - Dash [-]
  - Underscore [ _]
  - Space character [ ]
  - At sign [@]
  - Brackets [ (){}]
  - Slashes [/]
  - Caret [ ^]
  - Dollar sign [$]
  - Asterisk [*]
  - Commas [,]
  - Dot [.]

- Select an edge server type: HTTP or Streaming

  A third party application - **Wowza** will be installed automatically when installing a streaming edge server and additional charges will apply. Please, contact your account manager for details.

- Move the **Add to Marketplace** slider to the right to submit this server to the **OnApp CDN marketplace**. If so, the minimum required disk size is 1 TB.

- Click **Next**.

Any servers you submit will be assessed before they are accepted into the marketplace. Criteria include geographic location, bandwidth and server specs.

If your cloud has sufficient resources, but the **Next** button is dimmed during server creation, the reason might be a browser issue. For more information, refer to our **Knowledge base article**.
2.2.1.3  Step 3 of 4. Resources

- Select the compute zone and the specific compute resource to build the edge server on.
- Set the resources needed for this edge server: RAM, CPU cores and CPU priority.
- Choose a data store zone and data store for this edge server's primary disk.
- Set the primary disk size. The disk size is calculated in the following way: 10 GB for OS, the rest of total disk space is estimated 80% per Pull population and 20% per Push population.

The disk size should not exceed 2 TB when a new disk is added. You can later resize the disk if you need it to be larger than 2 TB.

- Choose a network zone and set the port speed for this edge server, or make it unlimited. It is not possible to set port speed value for edge servers based on smart compute resources.
- Choose the network from which the VS should get the IP address

2.2.1.4  Step 4. Confirmation

1. Move the Build Edge Server Automatically slider to the right if you want the system to automatically build the edge server. Otherwise, you will have to build your server manually after it is created.

2. Click Create Edge Server.

2.2.2  View CDN Edge Server Details

Edge servers are virtual Compute resources that are much like other virtual servers in your cloud. You can perform the same basic actions on them as for VSs.

To view all edge servers in the cloud:

1. Go to your Control Panel > CDN > Edge Servers menu to see an overview of all edge servers in your cloud: their label, IP addresses, power status (with on/off buttons), allocated disk size, RAM and backups.

2. To reboot, start up or shut down a CDN edge server, click the Actions button next to the required edge server, then select the relevant action.

3. To narrow the list of edge servers by type, click the relevant tab at the top of the list.

If you are viewing the edge servers list on a narrow screen, you can customize the way the table is displayed by clicking the actions icon at the top of the table. In the drop-down list that appears, check the columns you want to be displayed and click Apply. The narrower your screen is, the more unchecked columns will be hidden from the table. If your screen is too narrow to fit all the columns you have checked, a scrollbar will appear at the bottom of the edge servers list. You can always alter your column selection later. Note that by default the VIP column is not visible in the table on narrow screens.

Column selection is currently set for one browser. If you have checked some columns in one browser and open the list in some other browser, the column selection will be the default one for that other browser.

To view a particular edge server's details:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. Click the label of the edge server required.
3. On the screen that appears, use the top navigation tabs to manage your edge server.

Edge servers are managed with the same tools you use to manage VSs in your cloud. The table below links to the relevant sections of this guide's Virtual Servers chapter. The exception is that autoscaling, backups and firewall rules are not available for edge servers.

<table>
<thead>
<tr>
<th>Overview</th>
<th>The Overview tab shows CDN edge server information and gives access to the most frequently-used management tools.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>CDN edge server's details page</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>CDN edge server's CPU usage statistics</td>
</tr>
<tr>
<td>Billing Statistics</td>
<td>CDN edge server's billing statistics information</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Networking</th>
<th>The Networking tab gives access to the edge server's Network interfaces and IP addresses.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Interfaces</td>
<td>CDN edge server's network configuration</td>
</tr>
<tr>
<td>IP Addresses</td>
<td>CDN edge server's IP addresses</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
<th>The Storage tab lets you manage your edge server's disks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disks</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Console</th>
<th>The integrated VNC console tab gives access to the users' edge servers through the Control Panel UI. This option is available only for edge servers running on KVM compute resources.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re-connect</td>
<td>For the HTML5 console, click the Re-connect button if the connection is lost. The re-connection to the console runs as follows:</td>
</tr>
<tr>
<td></td>
<td>• If the console runs as expected, clicking the Re-connect button causes disconnection and the console is re-connected automatically after 1.5 seconds.</td>
</tr>
<tr>
<td></td>
<td>• If the console gets stuck, clicking the Re-connect button runs your request once again and re-connects the console without reloading.</td>
</tr>
<tr>
<td></td>
<td>• If the console gets disconnected with a status code and an error message, the console is re-connected automatically after 1.5 seconds.</td>
</tr>
</tbody>
</table>

4. To expand the Tools menu, click the Tools button on the edge server's screen. The list of available options depends on the edge server's status. For options description, refer to relevant sections of the Virtual servers chapter.

2.2.3 Manage CDN Edge Server

2.2.3.1 Edit a CDN Edge Server

Editing a CDN edge server means adjusting the resources allocated to it, changing its label and basic properties.
To edit a CDN edge server:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. Click the label of an edge server.
3. On the next screen, click the Tools button, then click the Edit Edge Server link:
   - Change the edge server label.
   - Edit CPU core/priority and RAM values.
   - Move the Add to Marketplace slider to the right to add the edge server to the marketplace. (This option is only available for edge servers that were not added to the marketplace earlier.)
4. Click Save Edge server.

2.2.3.2 Pause/Resume a CDN Edge Server

To pause/Resume my CDN Edge Server:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. Under the label of needed edge server label, click the Pause or Resume button.
   
   This will take 15 minutes to apply. Once the server is paused, it will stop serving CDN content, regardless of the content from your own CDN resources or marketplace subscribers.

2.2.4 Set VIP Status for Edge Server

If a Compute resource fails or reboots, the system migrates edge servers to another Compute resource, one server at a time. The order servers are migrated in is random. However, you can give an edge server “VIP” status, and this will give that server priority in the migration queue.

To set or remove VIP status for an edge server:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. Use the VIP button next to a required edge server to change its VIP status.

2.2.5 Delete CDN Edge Server

To delete a CDN edge server:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. Click the label of an edge server.
3. When the page loads, click the Tools button, then click Delete Edge Server.
4. You will be asked for confirmation before the edge server is deleted.

2.2.6 CDN Edge Server Network Interface Usage

Network Interface usage page displays bandwidth used in two charts: statistics for 24 hours and hourly statistics for the period up to three months.

To view the edge server’s network interface usage statistics:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. Click the label of the Edge Server required.
3. On the screen that appears, click Networking tab > Network Interfaces.
4. Click the Interface Usage icon next to the network interface needed.
5. On the screen that appears, set the start time and end time and click Apply.

2.2.7 Slicing Cache Files into Smaller Files

A CDN edge server stores and caches an entire file upon user’s request. For large files, e.g., videos larger than 1 GB, the CDN edge server may be overloaded even if the user spent watching only the first few minutes of the video.

The slice module allows to split a file into smaller files and cache the specified duration on the edge server. It reduces the disk space waste and improves the edge server’s stability and performance.

The slice operations are carried out on the backend and can only be verified using a raw log:

- A 1-GB test file prepared and configured to be split every 2 MB
- The $body_bytes_sent variable displays 2,000 bytes in the raw log below:

```
00:00:00 +0000 123456789 1.2.3.5 GET test-site.com /testfile.txt 200 1086
HTTP 0.000 0.002 110972 234567891 345678912 test-site.com 1.2.3.4
Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36 https://www.test-site.com/test-page - 2000 456789123 TLSv1.3 - - - - - - - - V8
```

Currently, it is a beta feature. To enable it in your CDN resources, contact OnApp Support.

2.3 CDN Storage Servers

CDN storage servers are used for storing the content which should be distributed over CDN. When the content is requested on the CDN, it is served by the edge server nearest to the customer’s geographical location.

There are two types of CDN storage servers in OnApp: HTTP and Streaming.

You can use smart Compute resources for CDN storage server deployment.

2.3.1 View CDN Storage Server Details

To view the list of all CDN storage servers in the cloud:

1. Go to your Control Panel > CDN > Storage Servers menu to see an overview of all storage servers in your cloud: their label, IP addresses, power status (with on/off buttons), allocated disk size, RAM and backups.
2. To reboot, startup/shutdown a storage server, click the Actions button next to the required storage server, then select the relevant action.
3. To narrow the list of storage servers by type, click the relevant tab at the top of the list.

To view a particular edge server's details:

1. Go to your Control Panel > CDN > Storage Servers menu.
2. Click the label of the storage server required.
3. On the screen that appears, use the top navigation tabs to manage your storage server.

Storage servers are managed with the same tools you use to manage VSs in your cloud. The table below links to the relevant sections of this guide's **Virtual Servers** chapter. The exception is that autoscaling, backups and firewall rules are not available for storage servers.

<table>
<thead>
<tr>
<th>Overview</th>
<th>The Overview tab shows storage server's information and gives access to the most frequently-used management tools.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Storage server's details page</td>
</tr>
<tr>
<td>CPU Usage</td>
<td>Storage server's CPU usage statistics</td>
</tr>
<tr>
<td>Billing Statistics</td>
<td>Storage server's billing statistics information</td>
</tr>
<tr>
<td>Networking</td>
<td>The Networking tab gives access to the storage server's Network interfaces and IP addresses.</td>
</tr>
<tr>
<td>Network Interfaces</td>
<td>Storage server's network configuration</td>
</tr>
<tr>
<td>IP Addresses</td>
<td>Storage server's IP addresses</td>
</tr>
<tr>
<td>Storage</td>
<td>Disks The Storage tab lets you manage your edge server's disks.</td>
</tr>
</tbody>
</table>

4. To expand the **Tools** menu, click the **Tools** button on the storage server's screen. The list of available options depends on the edge server's status). For options description, refer to **VS properties** section.

### 2.3.2 Create CDN Storage Server

To create new storage server:

1. Go to your Control Panel > **CDN > Storage Servers** menu.
2. Click the **Create Storage Server** button at the bottom of the screen.
3. Fill in the storage server creation form step by step:

   **Step 1 of 3. Properties**

   Specify the storage server details:

   - Specify the server's label in a human-recognizable format.
   - Select the storage server type: HTTP or Streaming.

   A third party application - **Wowza** will be installed automatically when installing a streaming storage server and additional charges will apply. Please, contact your account manager for details.

   - Location - choose the location group to assign this storage server to.
CDN Location - choose CDN location group to assign this edge server to.

Specify the Compute resource and Compute zone.

You can use **smart Compute resources** for CDN storage server deployment.

Click **Next**.

If your cloud has sufficient resources, but the **Next** button is dimmed during server creation, the reason might be a browser issue. For more information, refer to our **Knowledge base article**.

**Step 2 of 3. Resources**

Set the resources needed for this storage server: RAM, CPU cores and CPU priority.

The minimum RAM capacity is 2 GB.

Choose a data store zone for this storage server’s primary disk.

Set the primary disk size (Storage server HDD). The minimum required disk size is 20 GB.

Choose a network zone from the drop-down box.

Choose the network from which the VS should get the IP address.

If the option is available, you can also assign an IP address for the VS from the drop-down menu. Indicate Compute resource and network to have the list of available IPs. Tick the **Show Only My IP Addresses** check box to view only own IP addresses in the IP addresses dropbox.

Set the port speed in Mbps or tick it as unlimited. It is not possible to set port speed value for storage servers based on smart Compute resources.

Click **Next**.

**Step 3. Confirmation**

On the screen that appears, move the **Build Edge Server Automatically** slider to the right if you want to build the storage server automatically, otherwise, you will have to build your storage server manually after it is created.

Click the **Create Storage Server** button to start the creation process.
2.3.3 Edit CDN Storage Server

1. Go to your Control Panel > CDN > Storage Servers menu.
2. Click the label of a required storage server.
   - On the next screen, click the Tools button, then click the Edit CDN Edge Server link under the Storage Server Options.
   - Change the storage server label.
   - Edit CPU core/priority and RAM values.
3. Click Save.

2.3.4 Set VIP Status for Storage Server

If a Compute resource fails or reboots, the system migrates storage servers to another Compute resource, one server at a time. The order servers are migrated in is random. However, you can give a storage server "VIP" status, and this will give that server priority in the migration queue.

To set or remove VIP status for a storage server:

1. Go to your Control Panel > CDN > Storage Servers menu.
2. Use the VIP button next to a required storage server to change its VIP status.

2.3.5 Delete CDN Storage Server

To delete a storage server:

1. Go to your Control Panel > CDN > Storage Servers menu.
2. On the screen that appears, you'll see the list of all storage servers in the cloud. Click the HTTP/Streaming tabs to view storage servers by type.
3. Click the Actions button next to the storage server you want to remove, then click Delete.

2.4 CDN Resources

A CDN resource is a host (e.g. a specific webserver), the content of which you are going to distribute over the network of edge servers.

There are three types of CDN resources in the OnApp Control Panel:
- **HTTP** CDN resource type supports both Push and Pull population.
- **VoD** CDN resource type (Pull and Push types) allows using on-demand video streaming service - uploading video and streaming to the end-users.
- **Live Streaming** CDN resource type allows broadcasting content using CDN.

Only servers added to the edge groups assigned to the resource will distribute/cache the host's content.

To activate the CDN Resources menu, at least one CDN Edge Group with at least one edge server or marketplace location must be available.
Apart from the CDN Resources permissions enabled, the following requirements must be met for the publisher to be able to create respective resources. If the requirements are not met, the publisher will not be able to create a particular resource type:

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>HTTP Pull</th>
<th>HTTP Push</th>
<th>VoD Pull</th>
<th>VoD Push</th>
<th>Live Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements based on user Edge group</td>
<td>User bucket must include an edge group with at least one location that supports HTTP.</td>
<td>User bucket must include an edge group with at least one location that supports HTTP Push, and an HTTP storage server.</td>
<td>User bucket must include an edge group with at least one location that supports VoD Push.</td>
<td>User bucket must include an edge group with at least one location that supports VoD Push, and streaming storage server.</td>
<td>User bucket must include an edge group with at least one location that supports live streaming.</td>
</tr>
</tbody>
</table>

### 2.4.1 HTTP CDN Resource

There are two types of HTTP CDN resources:

- **HTTP Pull** is used for the delivery of small static content, such as images, CSS, and pdf files, via CDN edge servers. When a user requests content, it's cached on edge servers.
- **HTTP Push** is used for the delivery of large files (>3 MB), such as game patch, software installer, etc., via CDN edge servers. HTTP Push allows users to upload content to the CDN storage server without the origin.

**The detailed workflow for HTTP PULL CDN Resource**

1. The end users create a HTTP PULL CDN resource from the control panel.
2. The end users specify an origin (website) where the files of the website have to be stored.
3. A visitor requests content from the website.
4. The OnApp core CDN logic determines which CDN edge server (owned or subscribed) is appropriate for content delivery.
5. The CDN edge server fetches files from the specified origin (website).
6. The visitor loads the content at the fastest speed.

Site owners may set the cache expiry on the origin or define it in the advanced settings of the relevant CDN Resource. To offer HTTP Pull service to your clients, you need HTTP edge servers. You can build it [on your own](#), or subscribe for [HTTP CDN locations](#) in OnApp Federation (via the dashboard).

**The detailed workflow for HTTP PUSH CDN Resource**

1. The owner of the content uploads the files to the HTTP CDN storage server.
2. A visitor requests content from a website.
3. The OnApp core CDN logic determines which edge server is appropriate for content delivery.
4. The edge server fetches the files from the CDN storage server and cache them.
5. The visitor downloads the files at the fastest speed from the CDN edge server.
2.4.2 Video On Demand CDN Resource

A VoD CDN Resource is using for the delivery of static video files within the CDN via the following protocols: RTMP (Flash Player), RTSP (Android OS), MPEG-TS, Microsoft Silverlight (Microsoft Smooth Streaming), HDS (HTTP Dynamic Streaming), Apple iOS HLS.

There are 2 types of VoD CDN resources: VoD PUSH and VoD PULL.

The detailed workflow for VoD PUSH CDN Resource
1. The content owner should first upload the video files to the CDN storage server.
2. The CDN storage server distributes the files to the streaming edge servers automatically.
3. A visitor requests to view the videos.
4. The OnApp core CDN logic determines which streaming edge server is appropriate for content delivery.
5. A visitor loads the video at the fastest speed.

The detailed workflow for VoD PULL CDN Resource
1. The end users create a VoD PULL CDN resource from the control panel.
2. The end users specify an origin where the video files have to be stored.
3. A visitor requests to view the videos.
4. The OnApp core CDN logic determines which streaming edge server is appropriate for content delivery.
5. The streaming edge server fetches video files from the specified origin.
6. A visitor loads the video at the fastest speed.

2.4.3 Live Streaming CDN Resource

A Live Streaming CDN Resource delivers live streaming within the CDN via the following protocols: RTMP, RTSP (Android OS), MPEG-TS, Microsoft Silverlight (Microsoft smooth streaming), HDS (HTTP Dynamic Streaming), Apple HLS.

The detailed workflow for Live Streaming CDN Resource
1. The content owner publishes the live event through a video camera, live encoder, etc., using a publishing point server within the CDN (internal publishing point), or using the existing external publishing point for the live streaming source (external publishing point).
2. The live streaming is served through the streaming edge servers within the CDN.
3. A visitor requests to view the live event.
4. The OnApp core CDN logic determines which streaming edge server is appropriate for content delivery.
5. Visitor loads the video at the fastest speed.

2.4.4 View CDN Resources

Go to your Control Panel > CDN > Resources menu.
You will see the list of all CDN resources with the following information:
- **CDN Hostname** – hostname of the CDN resource.
- **Origin Sites** – path of the content that will be served from the CDN.
- **Type** - resource type: HTTP Push, HTTP Pull, VoD or Live streaming.
- **Cost** – cost of the resource.
To view HTTP, VoD or live streaming resources only, click the required tab.

You can edit/delete a resource using the relevant icons next to each resource in the list, and add a resource with the **CDN Resource Wizard** button.

### 2.4.5 View CDN Resource Details

View basic and advanced details of a CDN Resource.

**To view basic settings:**

1. Go to your Control Panel > **CDN > Resources** menu.
2. Click the required CDN Hostname.
3. On the page that appears, click the **Basic Settings** tab. The screen provides you with the following information (depending on the resource):

   **CDN Resource Details**
   - **Owner**
   - **CDN hostname**
   - **Resource type**
   - **Let’s Encrypt SSL** - whether Let’s Encrypt SSL is enabled for the resource or not

   To see more details on the Let’s Encrypt SSL status, select the **Click to view more** option.

   - **SSL On** - whether SSL is enabled for the resource or not
   - **SNI SSL Certificate** - custom SNI SSL certificate associated with the resource
   - **CDN reference** – the ID of the resource in database
   - **Resource Status** – shows the resource status. Click **Suspend** to terminate.

**Origins**

- The IP of the content that will be served from the CDN.

**DNS Settings**

- A CNAME for the CDN Hostname which can then be used to view the contents. Use this for the origin settings.

**Edge Groups**

- Shows to which Edge groups the resource is assigned.

**Last 24 hours cost**

- Cost of the resource for the last 24 hours.

**To view advanced details:**

1. Go to your Control Panel > **CDN > Resources** menu.
2. Click a CDN Hostname.
3. On the screen that appears, click the **Advanced Details** tab.
4. This screen provides you with the detailed information on the CDN Resources. See the following sections for details.

2.4.5.1 View HTTP CDN Resource Details
To view instructions and basic settings of an HTTP CDN resource:
1. Go to your Control Panel > CDN > Resources menu.
2. Click the required CDN Hostname.
3. On the page that appears, click the Basic Settings tab. The screen provides you with the following information:

**CDN Resource details**
- **Owner**
- **CDN hostname**
- **Resource type** - Push or Pull

- **Let's Encrypt SSL** - whether Let's Encrypt SSL is enabled for the resource or not

To see more details on the Let's Encrypt SSL status, select the **Click to view more** option.

- **SSL on** - whether SSL is enabled for the resource or not
- **SNI SSL Certificate** - custom SNI SSL certificate associated with the resource
- **CDN reference** – the ID of the resource in database
- **Resource status** – shows the resource status.

**Origins (HTTP Pull only)**
- **Path of the content that will be served from the CDN.**

**DNS Settings**
- **Add a CNAME for the CDN Hostname which can then be used to view the contents.**

**Edge Groups**
- **Shows to which Edge groups the resource is assigned.**

**Last 24 hours cost**
- **Cost of the resource for the last 24 hours.**

2.4.5.1.1 Advanced details
Click the **Advanced Details** tab to view advanced details of an HTTP CDN resource. The screen that appears will provide you with the following information:

- **Publisher name** - name of the user who created the CDN Resource record
- **Country Access Policy** - access policy to the CDN resource’s content for specified countries: either NONE (disabled), ALLOWED BY DEFAULT or DISABLED BY DEFAULT
- **Hotlink Policy** - whether hotlink policy is enabled or not
- **IP Access Policy** - access policy from a range of IP addresses: either NONE (disabled), ALLOWED BY DEFAULT or DISABLED BY DEFAULT
- **Password On** - whether the password is enabled or not
- **MP4 Pseudo Streaming** - whether the MP4 pseudo streaming is enabled or not
• **FLV Pseudo Streaming** - whether the FLV pseudo streaming is enabled or not
• **URL Signing Enabled** - whether access requires URL signing or not
• **CORS Headers Enabled** - whether the cross-origin resource sharing (CORS) is enabled or not

**Nginx Settings:**
- **Limit rate** - sets the speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
- **Limit rate after** - sets the amount after which the speed of a response to a client will be limited in MB. Maximum limit rate after value - 2147483647 KB
- **Proxy cache key** - key for caching

• **Origin Policy** - the type of connection chosen (HTTP or HTTPS)
• **Cache Expiry** - cache expiry time in minutes
• **Ignore Set-Cookie** - whether content caching with SetCookie response headers is enabled or not
• **Secure Wowza** - whether secure Wowza token is enabled or not
• **Token for Edge/Flash player** - whether token for Edge/Flash player is enabled or not
• **Token Authentication Enabled** - whether token authentication is enabled or not
• **Block search engine crawlers** - whether search engine crawlers are blocked from indexing the CDN content or not (for HTTP Pull CDN resources only)

2.4.5.1.2 **Upload instructions (HTTP Push only)**
Click the **Instructions** tab to view the instructions for uploading files and embedding video from HTTP Push CDN resources.

2.4.5.1.3 **Prefetch/Purge CDN content (HTTP Pull only)**
To **prefetch** or **purge** the resource content of the HTTP Pull CDN resource, click the required tab respectively.

2.4.5.1.4 **View CDN resource billing statistics**
To view the resource billing statistics, click the **Billing Statistics** tab.

**Advanced Reporting**
View the analysis of your resources in reports.

**HTTP Caching Rules**
Customize the CDN edge server behavior by setting the rules. More info at **HTTP Caching Rules**.

2.4.5.2 **View VoD CDN Resource Details**
To view details of a video on demand CDN resource:
1. Go to your Control Panel > **CDN > Resources** menu.
2. Click the CDN hostname of a required VOD CDN resource.
3. On the page that appears you will see basic resource settings:

**CDN Resource details**
- **Owner**
- **CDN hostname**
- **Resource type** - VOD Push or VOD Pull
OnApp Cloud 6.5 Edge 5 CDN Guide

- SSL on - whether SSL is enabled for the resource or not
- CDN reference – the ID of the resource in database
- Resource status – shows the resource status.

Origins (VOD Pull only)

- Path of the content that will be served from the CDN.

Edge Groups

- Shows to which Edge groups the resource is assigned.

Last 24 hours cost

- Cost of the resource for the last 24 hours.

2.4.5.2.1 Advanced details
Click the Advanced details tab to view advanced details of a VOD CDN resource. The screen that appears will provide you with the following information:

- Publisher name - name of the user who created the CDN Resource record
- Country Access Policy - access policy to the CDN resource’s content for specified countries: either NONE (disabled) or BLOCK BY DEFAULT
- Hotlink Policy - whether hotlink policy is enabled or not
- Secure Wowza - whether secure Wowza token is enabled or not
- Token for Edge/Flash player - whether token for Edge/Flash player is enabled or not
- Token Authentication Enabled - whether token authentication is enabled or not

2.4.5.2.2 Upload instructions
Click the Instructions tab to view the instructions for uploading files and embedding video from Video On Demand CDN resources.

2.4.5.2.3 View CDN resource billing statistics
To view the resource billing statistics, click the Billing Statistics tab.

2.4.5.3 View Live Streaming CDN Resource Details
To view details of a live streaming CDN resource:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the CDN Hostname of a required live streaming CDN resource.
3. On the page that appears you will see basic resource settings:

On this page:

- CDN Resource details
- Advanced Settings
- Upload instructions
- View CDN resource billing statistics
2.4.5.3.1 CDN Resource details
- **Owner**
- **CDN hostname**
- **Resource type** - Live Streaming
- **CDN Reference** the ID of the resource in database
- **Publishing Point** - the publishing point type: external or internal
- **Main Internal Publishing Location** - main internal publishing point URL address (in case of internal publishing point type)
- **Failover Internal Publishing Location** - internal publishing point failover URL (in case of internal publishing point type)
- **Main External Publishing Location** - main external publishing point URL address (in case of external publishing point type)
- **Failover External Publishing Location** - external publishing point failover URL (in case of external publishing point type)
- **Resource status** – shows the resource status

2.4.5.3.1.1 Edge Groups
- Shows to which Edge groups the resource is assigned.

2.4.5.3.1.2 Last 24 hours cost
- Cost of the resource for the last 24 hours.

2.4.5.3.2 Advanced Settings
Click the **Advanced details** tab to view advanced details of a Live Streaming CDN resource. The screen that appears will provide you with the following information:

1. Go to your Control Panel > **CDN** > **Resources** menu.
2. Click a CDN Hostname.
3. On the screen that appears, click the **Advanced Details** tab.
4. This screen provides you with the following information:
   - **Publisher name** - name of the user who created the CDN Resource record
   - **Country Access Policy** - access policy to the CDN resource's content for specified countries: either NONE (disabled) or BLOCK BY DEFAULT
   - **Hotlink Policy** - whether hotlink policy is enabled or not
   - **Secure Wowza** - whether secure Wowza token is enabled or not
   - **Token for Edge/Flash player** - whether token for Edge/Flash player is enabled or not
   - **Token Authentication Enabled** - whether token authentication is enabled or not

2.4.5.3.3 Upload instructions
Click the **Instructions** tab to view the instructions for uploading files and embedding video from Live Streaming CDN resources.

2.4.5.3.4 View CDN resource billing statistics
To view the resource billing statistics, click the **Billing Statistics** tab.
2.4.6 Create HTTP CDN Resource

To add an HTTP CDN resource:

1. Go to your Control Panel > CDN > Resources menu. The page that loads shows the list of CDN resources.

2. To create a new CDN resource, click the "+" button in the top right corner or the CDN Resource Wizard button.

3. Follow the steps of the CDN resource creation wizard:

   On this page:

   - **Type Select**
   - **Properties**
   - **Edge Locations**
   - **Advanced Settings**

2.4.6.1 Type Select

Click HTTP to select the required resource type, and then click Next to proceed.

2.4.6.2 Properties

- **CDN hostname** – the hostname from which you will serve static content. E.g. if your site (origin) is onapp.com, and you want to serve static content from the CDN and make it available at static.onapp.com, then static.onapp.com would be the CDN hostname.

- **Enable SSL** - move the slider to the right to enable the secure socket protocol for your CDN resource.
  - **Let's Encrypt** - select this option if you want to use a Let's Encrypt SSL certificate for the resource

   The Let's Encrypt SSL certificate is automatically generated for the following types of hostname:

   - CDN hostname (e.g. cdn.abc.com)
   - Alias (CNAME) for CDN hostname (e.g. 123.r.worldcdn.net)
   - Operator Basehostname (e.g. 123.r.worldcdn.net)
   - Secondary CDN hostname (e.g. cdn1.abc.com)

   All the hostnames are bundled into one Let's Encrypt SSL certificate. If the secondary hostname cannot be validated, the system generates the LE certificate based on the CDN hostname, CNAME, and Operator Basehostname. The unverified hostname is revalidated by the system every 15 minutes.

- **Shared SSL** - choose this option if you want to apply a shared SSL certificate for the resource
If the SSL protocol is enabled, you can only have fourth-level domain names. If you select the Shared SSL certificate, the 'r.worldssl.net' ending will be automatically added to the CDN hostname. Be aware that if CDN hostname ends with 'r.worldssl.net', it cannot be digit-only (for example 123456.r.worldssl.net is not applicable).

A CDN resource can only be linked to one SSL certificate - Let's Encrypt, shared, or custom SNI.

- **Custom SNI SSL** - choose this option if you want to apply a custom SNI SSL certificate for the resource and choose the required certificate from the drop-down list.
- **Content origin** – specify the content origin type (PULL or PUSH):
  - For the PULL type, you can use a custom origin port. Specify a port number using the colon character ("\:"") in the Origins field. If you do not indicate the origin port, then the system will put it by default depending on origin policy:
    - 80 if origin policy is HTTP
    - 443 if origin policy is HTTPS
    - None if origin policy is AUTO, that is when the origin port is custom

The valid port values include 80, 443, and the range from 1024 to 65535. Values other than mentioned above will be forbidden.

In case of using multiple origins, the same port number should be specified for all origins using a colon character ("\:""). Erase the port number from the origin resource field to reset the custom origin port.

Please note that you can specify one or more origins in the Origins field:
- When a single origin is used, specify the value based on a hostname or IP.
- When more than one origin is used, only IP addresses are allowed.

For IP-based origins, ensure your origin is configured to accept HTTP requests with a CDN resource primary hostname as a Host header from CDN edge servers.

When creating a CDN resource, specify domain.com or subdomain.domain.com in the Origins field. domain.com/folder is not allowed in the Origins field. If
you want CDN to cache the content only in `domain.com/folder`, move the content from `domain.com/folder` to the subdomain `folder.domain.com`, which should be used as an origin.

- For the PUSH type:
  - Storage server location - choose the storage server location from the drop-down list
  - FTP password - specify the FTP password. It can consist of 6-32 alphanumeric characters
  - FTP password confirmation - confirm the password

2.4.6.3 Edge Locations
Tick the checkbox next to the edge group(s) that will share the new resource. Available groups depend on the assigned bucket's edge groups limit. The map displays own, subscribed and available CDN resources:

Map legend:

At this point, you can create the CDN resource or proceed to the Advanced Settings step which is optional in the wizard.

2.4.6.4 Advanced Settings

**Origin Policy**
The origin policy, which is available to a CDN resource, allows a CDN edge server to fetch content from the origin by using different HTTP or HTTPS protocol.

Select the type of the connection from the drop-down list:

- **HTTP**—connection between an edge server and an origin where HTTP is always used. It is a default value
- **HTTPS**—connection between an edge server and an origin where HTTPS is always used
- **AUTO**—connection between an edge server and an origin based on a visitor's request (HTTP or HTTPS)

**Country Access**
Configure a rule to enable/disable access to the CDN resource’s content for specified countries.
- **Access Policy** – select Disabled to switch off the rule, otherwise, choose between Allow by default/Block by default.
- **Except for Countries** – select countries to which the access policy won’t be applied. To select more than one country, hold Ctrl during selection.

**Hotlink Policy**
- **Hotlink Policy** – select Disabled to switch off a hotlink policy, otherwise, choose between Allow by default/Block by default.
- **Except for domains** – specify domains to which the hotlink policy won’t be applied.

1. Please note that a wildcard is not supported. Instead, use a URL.

**IP Access**
Configure a rule to enable/disable access to the CDN resource’s content for a range of IP addresses, including both IPv4 and IPv6.
- **Access Policy** – select Disabled to switch off the rule, otherwise, choose between Allow by default/Block by default.
- **Except for IP Addresses** – fill in IP address(es) to which the access policy won’t be applied.

**Secondary CDN Hostnames**
Submit secondary hostnames apart from the default one for HTTP based CDN sites. With these configured, users will be able to access the CDN site using secondary CDN hostname(s). You can add up to 7 secondary CDN hostnames to your CDN resource.

To be able to use a secondary hostname for a CDN resource with SSL enabled, you need an SSL certificate for your custom hostname. For more details about the purchase of an SSL certificate, contact OnApp Support. Also, set CNAME for both CDN hostname and secondary CDN hostnames.

If you create a CDN resource with the following settings:
- **CDN hostname**—cdn.example.com
- **Origin**—example.com
- **Secondary hostname**—cdn1.example.com, cdn2.example.com
and visitors visit cdn.example.com, cdn1.example.com, cdn2.example.com, all the three URLs will be displayed as example.com.

URL Signing
Protect your files from unauthorized access with a key. A signed URL looks like http://example.com/filename?hash=DMF1ucDxtqgxwYQ==.

- Enable URL Signing – move the slider to the right to enable it.
- URL Signing Key – fill in the key which will be used for URL signing. The secret key is similar to a password and can contain a minimum of 6 to a maximum of 32 characters. Spaces are not allowed.

A signed URL can be in one of the following formats:

- Query string format: http://example.com/filename?secure=DMF1ucDxtqgxwYQ&expires=1546300800&ip=1.2.3.4
- Path format: http://example.com/secure=DMF1ucDxtqgxwYQ&expires=1546300800&ip=1.2.3.4/filename

A secure token consists of the following parameters:

- Expires—the expiration time of a URL or the time when an URL becomes invalid. The time is passed in the URL itself in a Unix timestamp format and takes part in hash generation. It is an optional parameter
- Path—a file path or file directory

For HLS, put a path instead of an M3U8 file, so that all the chunks of the HLS are authenticated as well.

- Key—a URL signing key
- IP—an IP that provides access. It is optional and only one IP allowed when generating the hash key each time

Here is the format of a secure token:

<expires><path><key><ip>

Here is an example of the PHP script used to generate the hash key:
*/
* Create hash link CDN resource
* @param string $cdnResourceUrl
* The CDN resource URL, eg cdn.yourdomain.com
* @param string $filePath
* File path of the CDN resource
* @param string $secretKey
* The secret key that is obtained from CDN resource property
* @param int $expiryTimestamp [optional]
* UNIX timestamp format, specify how long the hash link is accessible
to the public
* By default will be accessible forever.
* @return string URL with generated hash link
* URL with designated format to access the resource
* Example:
* Generate hash link for resource  www.example.com/images/photo.png
* for next 3 days, assume today is Sun, 01 Apr 2012.
* <?php
* $hashLink = generateHashLink('www.example.com',
* '/images/photo.png', 'l33tf0olol', 1333497600);
* print $hashLink;
* ?>
* http://www.example.com/images/photo.png?secure=kaGd_cu6Iy4LDgfX3jy5Rw==,13
33497600
* /
* function generateHashLink($cdnResourceUrl, $filePath, $secretKey,
$expiryTimestamp = NULL){
    // NOTE [yasir 20110331] + and ?  are some of represented chars of
    // based64 encoding (8 bits)
    // + is 62 and / is 63 . and These char should be replaced by
    other predefined chars.
    $searchChars = array('+','/');
    $replaceChars = array('-','_');
    if($filePath[0] != '/'){
        $filePath = '/{$filePath}';
    }
    if($pos = strpos($filePath, '?')){
        $filePath = substr($filePath, 0, $pos);
    }
    $hashStr = $filePath.$secretKey;
    if($expiryTimestamp){
        $hashStr = $expiryTimestamp.$hashStr;
        $expiryTimestamp = "/{$expiryTimestamp}";
    }
    return "http://{$cdnResourceUrl}{$filePath}?secure=".
    str_replace($searchChars, $replaceChars,
    base64_encode(md5($hashStr, TRUE))).
    $expiryTimestamp;
}
Ruby:
ruby UrlSigning.rb -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

Python:
python UrlSigning.py -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

PHP:
php UrlSigning.php -f path -s https -r example.com -p images/photo.png -k abc123s -e 1546300800 -i 1.2.3.4

Java:
javac UrlSigning.java
java UrlSigning -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

Available options:
- f: format, path or querystring, default = querystring
- s: scheme for resource URL, http or https, default = http
- r: resource hostname (compulsory)
- p: file path of the resource, default = /
- k: URL signing key (compulsory)
- e: expiration of the URL (optional)
- i: IP that allow to access (optional)

Cache Expiry
- Cache expiry – set the cache expiry time in minutes (min=1, max=35000000).

Password
- Enable Password – move the slider to the right to restrict access to the resource (cdn hostname).
- Unauthorized HTML – fill in the text which will be displayed for unauthorized login.
- Username – choose a username.
- Password – select password for the user.

To remove a user, clear both fields.

Pseudo Streaming
- Enable MP4 pseudo streaming – move the slider to the right to enable the pseudo streaming support for MP4 file type.
- Enable FLV pseudo streaming – move the slider to the right to enable pseudo streaming for FVL file type, respectively.

With pseudo streaming enabled, your viewers can seek around a video even if it has not finished downloading. A Flash player and a prepared video are required for pseudo-streaming.
CORS Header

- *Enable CORS headers* - move the slider to the right to enable cross-origin resource sharing (CORS) by adding HTTP header with Access-Control-Allow-Origin: *

Ignore Set-Cookie

- *Ignore Set-Cookie* - move the slider to the right to enable caching content with Set-Cookie response headers.

Nginx Settings

- *Limit rate* - set speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
- *Limit rate after* - the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value -2147483647 KB
- *Proxy cache key* - key for caching. Select one of four supported types from the drop-down list:
  - $host$request_uri
  - $host$url
  - $proxy_host$request_uri
  - $proxy_host$url

Due to Nginx limitations, file extensions are not supported for gzip. It supports only specific MIME types (the Content-Type header returned from an origin). You can change the MIME types in the origin using the parameters from the whitelist.

**Click here to view the whitelist**

text/compressible

text/html

text/xml

text/javascript

text/css

application/x-javascript

text/xml

application/xml

application/xml+rss

application/json

application/javascript

image/svg+xml

image/x-icon

image/vnd.microsoft.icon

application/x-font-ttf

application/vnd.ms-fontobject

application/x-font-opentype

application/x-font-truetype

application/x-font-ttf

font/eot
font/opentype
font/otf

The MIME types included in the whitelist are gzipped on demand on CDN edge servers. Any MIME types that are not listed but can be further compressed can be added to the whitelist.

The MIME types in the whitelist can be requested either in a gzip or non-gzip file format based on the passed headers. The reason why it is based on the passed headers is serving compressed content without the client requesting it and risks of breaking client/browser that do not handle Content-Encoding: gzip properly.

Search Engine Crawlers

- **Block search engine crawlers** - move the slider to the right to block web crawling bots from indexing the CDN content (for HTTP Pull CDN resources only).

HTTP Live Streaming (HLS) Optimization

- **Enable HLS Optimization** - move the slider to enable/disable HLS optimization. This option is available only for HTTP Pull CDN resources.

- **Enforce Cache Expiry** - tick this checkbox to create an HTTP rule that will enforce cache expiry. After you enable HLS optimization during HTTP Pull resource creation or editing, this checkbox will not be displayed if you edit the resource as long as the **Enable HLS Optimization** option is switched on. If you disable the option and then enable it again, this checkbox will be displayed. If an enforce cache expiry rule has already been set for the resource, a new rule will not be created after you check this box and save changes. You can manually add or delete the HTTP rule that will enforce cache expiry on the [HTTP Caching Rules](#) page of your CDN resource.

To ensure Samsung Smart TV’s compatibility with Microsoft Smooth Streaming through the CDN, the Suppress CDN Headers feature must be enabled. Contact [OnApp Support](#) to have this feature available in Control Panel.

4. Click **Create CDN Resource**.

### 2.4.7 Create Video On Demand CDN Resource

#### 2.4.7.1 Requirements to Provide VoD Services

To provide VoD CDN service to your clients, you need the following:

- Streaming edge servers (**created on your own in Control Panel**) or subscription for the streaming location from CDN federation. Assign these streaming location into the edge group of the billing plan used by your client. It is applicable to both VoD Pull and VoD Push CDN resources.

- The streaming CDN storage server, which is applicable only to VoD Push CDN resources. It is for your clients to be able to upload their video files.
Both streaming edge servers and storage servers require a Wowza charge of $50/month/instance.

On this page:

- **Requirements to Provide VoD Services**
- **Create Video On Demand CDN Resource**
  - Type Select
  - Properties
  - Edge Locations
  - Advanced Settings

### 2.4.7.2 Create Video On Demand CDN Resource

To add a VoD CDN resource:

1. Go to your Control Panel > **CDN > Resources** menu.
2. Click the "+" button in the top right corner or the **CDN Resource Wizard** button.
3. Follow the steps of the CDN resource creation wizard:

#### 2.4.7.2.1 Type Select

Click **VOD** to select the required resource type, and then click **Next** to proceed.

#### 2.4.7.2.2 Properties

- **CDN hostname** – specify the name which will serve as a label only

  For VoD CDN resources, a CNAME is not required to be set for a CDN hostname, as the CDN hostname is not used. For example, if the CDN hostname is **cdn.domain.com**, the video files are always served from **video.domain.com**.
  
  A CDN hostname is used for labelling purposes in Control Panel. If the CDN hostname of a VoD resource has a configured CNAME, e.g., **xxx.r.worldcdn.net** (where xxx is a resource ID), a message with a Wowza version will appear by default when visitors access the CDN hostname.

  - **Content origin** – specify the content origin type PULL or PUSH
    
    If you have selected the PULL type, specify the origin.
    
    If you have selected the PUSH type:
    
    - **Storage server location** - choose the storage server location from the drop-down list
    - **FTP password** - specify the FTP password. It can consist of 6-32 alphanumeric characters
    - **FTP password confirmation** - confirm the password

#### 2.4.7.2.3 Edge Locations

Tick the checkbox next to the edge group(s) which will share the resource added. Available edge groups depend on the assigned bucket limits.

The map displays own, subscribed and available CDN resources:
2.4.7.2.4 Advanced Settings

Country Access

Configure a rule to enable/disable access to the CDN resource’s content for specified countries.

- **Access Policy** – select the Disabled option to switch off a rule or Block by default. If the access policy is set to block by default, fill in the Except for Countries field to specify countries to which the access policy won’t be applied. To select more than one country, hold Ctrl during selection.

Hotlink policy

- **Hotlink policy** - select the Disabled option to switch off hotlink policy security; otherwise choose Block by default. If the hotlink policy is set to block by default, fill in the Except for domains field to specify the domains to which the hotlink policy won’t be applied.

Cache expiry

- **Cache expiry** – set the cache expiry time in minutes (min=1, max=35000000).

For a VoD Push CDN resource, video files are cached on the streaming edge servers for 24 hours. For a VoD Pull CDN resource, video file caching honors the expiry host header from the origin. If the origin sends a max-age=1000, it means that the video files will be stored for 1,000 seconds.

Ignore Set-Cookie

- **Ignore Set-Cookie** - tick this checkbox to enable caching content with Set-Cookie response headers.

Search Engine Crawlers
• **Block search engine crawlers** - move the slider to the right to block web crawling bots from indexing the CDN content (for HTTP Pull CDN resources only).

**Secure Wowza**

- **Enable secure Wowza** – tick the checkbox to protect your stream with Wowza secure token.
- **Token for Edge/Flash player** – specify the token authentication code. The authentication code will be then saved on the edge server. The token at the player side and the token at the edge server must match in order to make the video available.

VOD setup requires the origin to send the `Accept-Ranges: Bytes` header in order for Wowza to make range requests to the Nginx cache. Amazon S3 does not send the header by default. For this, add the `Accept-Ranges: Bytes` header to the files stored in Amazon S3.

**Token Authentication**

- **Enable Token Authentication** - tick the checkbox to enable token authentication.
- **Token Auth Primary Key** - specify the secret key to be used with the scripts which generate token.
- **Token Auth Backup Key** - input the backup key which can be used if you want to change the primary key. To make sure the link generated with existing token won't be broken, the suggested procedure is the following:
  - Copy the existing primary key and paste to the backup key.
  - Fill in a new primary key.
- **Protected Path** - specify the protected path. By default it is "/". Only the Path inserted is allowed to be streamed.

Now proceed with running the .NET or JAVA scripts to complete the procedure.

4. Click **Create CDN Resource**.

Only mp4 and flv files are currently supported.

For a VoD CDN resource, use unicast to stream a video file (not live streaming with a camera).

**Click here to see more details**

- The client's player first saves part of the video (not the whole file) and then plays it.
- The part of the video is distributed via unicast connection. If the video bitrate is 300 Kbps and there are 100 people watching the video, it requires 30,000 Kbps.
2.4.8 Create Live Streaming CDN Resource

2.4.8.1 Requirements to Provide Live Streaming Services
To provide Live Streaming CDN service to your clients, you would require at least one of the following:

- Streaming edge servers (created on your own in control panel). These streaming edge servers and storage servers would require Wowza charges, which will be forwarded to your account at $50/month/instance.

  Subscription for the streaming location from CDN federation. Assign these streaming location into the edge group of the billing plan, used by your client. It is applicable to both internal and external live streaming publishing points.

On this page:

- Requirements to Provide Live Streaming Services
- Create Live Streaming CDN Resource
  - Type Select
  - Properties
  - Edge Locations
  - Advanced Settings

2.4.8.2 Create Live Streaming CDN Resource
To add a live streaming CDN resource:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the "+" button in the top right corner or the CDN Resource Wizard button.
3. Follow the steps of the CDN resource creation wizard:

2.4.8.2.1 Type Select
Click LIVE STREAMING to select the required resource type, and then click Next to proceed.

2.4.8.2.2 Properties
- CDN hostname – specify the name which will serve as a label only

For Live Streaming CDN resources, a CNAME is not required to be set for a CDN hostname, as the CDN hostname is not used. For example, if the CDN hostname is cdn.domain.com, the video files are always served from video.domain.com. A CDN hostname is used for labelling purposes in Control Panel. If the CDN hostname of a Live Streaming resource has a configured CNAME, e.g., xxx.r.worldcdn.net (where xxx is a resource ID), a message with a Wowza version will appear by default when visitors access the CDN hostname.
HTTPS support for a live streaming resource is limited to a playback via the video.worldcdn.net CDN hostname. Currently, HTTPS support is unavailable for white label CDN hostnames.

- **Publishing point** – specify the resource’s publishing point settings - Internal or External. The internal publishing point is on one of your storage servers. It is configured at the next step - Edge locations. The external publishing point is simply an URL. Specify its settings here:

- **External publishing location** - specify your publishing point's URL as an RTMP protocol. For example, rtmp://domain.com/xxx

An external publishing point must start with an rtmp, e.g., rtmp://xxxxxx.xxxx.com/live. It is possible to exclude the stream name in the external publishing point URL. For example, if the URL of an RTMP stream is rtmp://mystreamingurl.com/live/stream1, insert rtmp://mystreamingurl.com/live/ for the external publishing point URL. Once it is restreamed across the CDN, it can be viewed in RTMP, HDS, HLS, Microsoft Smooth Streaming, and RTSP protocols. The connection flow would be as follows:

**Viewer’s PC > Closest Streaming Edge > External Publishing Point.**

For more details on limitations, click here
- Wowza supports only restreaming of h.264+AAC to HLS/HDS.
- For HLS/HDS restreaming, only Wowza RTMP origin is supported.

- **Failover external publishing location** - specify the failover URL

Please note that username and password used to connect to a live stream publishing point cannot be changed.

2.4.8.2.3 Edge Locations
Tick the checkbox next to the edge group(s) which will share the resource added. Available edge groups depend on the assigned bucket limits.
The map displays own, subscribed and available CDN resources:

Map legend:
In case of choosing Internal Publishing Point in the previous step, specify its settings here:

- **Internal publishing location** - select location of your edge servers from the drop-down list

**Click here to explore how the publishing point is selected**

For instance, you created a CDN resource: `cdn.livestream.com` with the internal publishing point Location A. Assuming location A has three streaming edge servers X, Y, and Z, our system will choose (randomly) a specific streaming edge server among the three as the publishing point. If X is chosen, whenever you connect to the live streaming resource FMS URL, it always reaches the edge server X. If server X is down or removed, CDN resource `cdn.livestream.com` will be down, as the encoder can no longer connect to the FMS URL, because the FMS URL will be only resolved to edge server X. After that, you have to select a different publishing point location from the UI. Our system will not automatically use Y and Z servers from the same location as the publishing points.

- **Failover internal publishing location** - specify the failover edge server

2.4.8.2.4 Advanced Settings

**Country Access**

Configure a rule to enable/disable access to the CDN resource’s content for specified countries.

- **Access Policy** – select the Disabled option to switch off a rule or Block by default. If the access policy is set to block by default, fill in the Except for Countries field to specify countries to which the access policy won’t be applied. To select more than one country, hold Ctrl during selection.

**Hotlink policy**

- **Hotlink policy** - select the Disabled option to switch off hotlink policy security; otherwise choose Block by default. If the hotlink policy is set to block by default, fill in the Except for domains field to specify the domains to which the hotlink policy won’t be applied.

The Hotlink policy feature is applicable to RTMP and RTSP only. For HLS streaming, it is not supported.

If you need to enable the hotlink policy for HLS streaming, it is recommended to create an HTTP Pull resource.

**Secure Wowza**

- **Enable secure Wowza** – tick the checkbox to protect your stream with Wowza secure token.

- **Token for Edge/Flash player** – specify the token authentication code. The authentication code will be then saved on the edge server. The token at the player side and the token at the edge server must match in order to make the video available.

**Token Authentication**

- **Enable Token Authentication** - tick the checkbox to enable token authentication.
- **Token Auth Primary Key** - specify the secret key to be used with the scripts which generate token.
- **Token Auth Backup Key** - input the backup key which can be used if you want to change the primary key. To make sure the link generated with existing token won't be broken, the suggested procedure is the following:
  - Copy the existing primary key and paste to backup key.
  - Fill in a new primary key.
- **Protected Path** - specify the protected path. By default it is "/". Only the Path inserted is allowed to be streamed.

Now proceed with [running the .NET or JAVA scripts](#) to complete the procedure.

With an HTTP Pull CDN resource, you can restream an HLS stream. To do it, you need to [create an HTTP PULL resource](#).

When creating the HTTP Pull resource, in the Origins field, specify the HLS URL (without a path and a filename), and move the **Enable HLS Optimization** slider to the right.

**Click here to see the possibilities this feature provides**

- Cache in the edge server RAM rather than in a hard disk
- An automatic cache expiry rule with 10 seconds for video content
- The resource will have a 30-second DNS TTL

4. Click the **Create CDN Resource** button.

For a Live Streaming CDN resource, use multicast.

**Click here to see more details**

- The stream is never saved in the client's hard drive.
- The stream is distributed via multicast. If the video bitrate is 300 Kbps and there are 100 people watching the video, it requires only 300 Kbps for the camera to distribute it to CDN PoPs.

If a CDN live streaming resource is used as an origin, it may cause a poor cache hit ratio on HLS.

To improve the cache hit ratio, refer to the instructions on [how to configure Wowza as an HTTP caching origin](#).

**Live Streaming URL**

**To get the CDN Live Streaming URL**

You can use the following links:
For RTMP player (JW, FLOW)
** http://video.worldcdn.net/[resource id]/_definst_/streamname.smil
or
rtmp://[resource id].r.worldcdn.net/[resource id]/_definst_/streamname

For Apple devices
** http://video.worldcdn.net/[resource id]/_definst_/streamname.m3u8
or
http://[resource id].r.worldcdn.net/[resource id]/_definst_/streamname/playlist.m3u8

For Android devices
rtsp://[resource id].r.worldcdn.net/[resource id]/_definst_/streamname

For Adobe HTTP Dynamic Streaming
** http://video.worldcdn.net/[resource id]/_definst_/streamname.f4m
or
http://[resource id].r.worldcdn.net/[resource id]/_definst_/streamname/manifest.f4m

Where:
- The [resource id] is the numeric number of the "CDN Reference" which shows on the details page.
- "Streamname" should be replace with the stream name you used in Adobe Media Encoder.

This URL method performs a 302 Url Redirection, which may not be compatible with newer web players such as JWplayer and Flowplayer.

2.4.9 Edit CDN Resource

Note that a CDN resource cannot be re-assigned to another owner after creation. Instead, you need to delete the resource and then re-create it under the new owner account.

To edit a CDN Resource:
1. Go to your Control Panel > CDN > Resources menu.
2. Click the Actions button next to a resource and choose Edit.
3. On the screen that appears, you can edit the following CDN resource parameters:

2.4.9.1 CDN Resource Properties
- **Enable SSL** - move the slider to the right to enable the secure socket protocol for your CDN resource.
  - Let's Encrypt - select this option if you want to use a Let's Encrypt SSL certificate for the resource

The Let's Encrypt SSL certificate is automatically generated for the following types of hostname:
- CDN hostname (e.g. cdn.abc.com)
- Alias (CNAME) for CDN hostname (e.g. `123.r.worldcdn.net`)
- Operator Basehostname (e.g. `123.r.worldcdn.net`)
- Secondary CDN hostname (e.g. `cdn1.abc.com`)

All the hostnames are bundled into one Let's Encrypt SSL certificate. If the secondary hostname cannot be validated, the system generates the LE certificate based on the CDN hostname, CNAME, and Operator Basehostname. The unverified hostname is revalidated by the system every 15 minutes.

- **Shared SSL** - choose this option if you want to apply a shared SSL certificate for the resource

  If the SSL protocol is enabled, you can only have fourth-level domain names.

  If you select the Shared SSL certificate, the `r.worldssl.net` ending will be automatically added to the CDN hostname. Be aware that if CDN hostname ends with `r.worldssl.net`, it can not be digit-only (for example `123456.r.worldssl.net` is not applicable).

  A CDN resource can only be linked to one SSL certificate - Let's Encrypt, shared, or custom SNI.

- **Custom SNI SSL** - choose this option if you want to apply a custom SNI SSL certificate for the resource and choose the required certificate from the drop-down menu

  - **CDN hostname** – the hostname from which you will serve static content.
  - **Origins** - specify the content origin type (PULL or PUSH):

    - For the PULL type, you can use a custom origin port. Specify a port number using the colon character (";") in the Origins field. If you do not indicate the origin port, then the system will put it by default depending on origin policy:
      - 80 if origin policy is HTTP
      - 443 if origin policy is HTTPS
      - None if origin policy is AUTO, that is when the origin port is custom

    The valid port values include 80, 443, and the range from 1024 to 65535. Values other than mentioned above will be forbidden.

In case of using multiple origins, the same port number should be specified for all origins using a colon character (";:"). Erase the port number from the origin resource field to reset the custom origin port.
Please note that you can specify one or more origins in the *Origins* field:

- When a single origin is used, specify the value based on a hostname or IP.
- When more than one origin is used, only IP addresses are allowed.

For IP-based origins, ensure your origin is configured to accept HTTP requests with a CDN resource primary hostname as a Host header from CDN edge servers.

When creating a CDN resource, specify *domain.com* or *subdomain.domain.com* in the *Origins* field. *domain.com/folder* is not allowed in the *Origins* field. If you want CDN to cache the content only in *domain.com/folder*, move the content from *domain.com/folder* to the subdomain *folder.domain.com*, which should be used as an origin.

- For the PUSH type:
  - *Storage server location* - choose the storage server location from the drop-down menu.
  - *FTP password* - specify the FTP password. It can consist of 6-32 alphanumeric characters.
  - *FTP password confirmation* - confirm the password.

**Edge Groups**

Tick the box next to the edge group(s) that will share the new resource. Available groups depend on the assigned bucket's edge groups limit. 

*Advanced settings* - move the slider to the right to enable advanced settings for CDN resource.

At this point, you can create the CDN resource or proceed to the Advanced Settings step which is optional.

**Country Access**

Configure a rule to enable/disable access to the CDN resource’s content for specified countries.

- *Access Policy* – select Disabled to switch off the rule, otherwise, choose between Allow by default/Block by default.
• *Except for Countries* – select countries to which the access policy won’t be applied. To select more than one country, hold Ctrl during selection.

**Hotlink Policy**

• *Hotlink Policy* – select Disabled to switch off a hotlink policy, otherwise, choose between Allow by default/Block by default.

• *Except for domains* – specify domains to which the hotlink policy won’t be applied.

> Please note that a wildcard is not supported. Instead, use a URL.

2.4.9.1.1 Origin Policy

The origin policy, which is available to a CDN resource, allows a CDN edge server to fetch content from the origin by using different HTTP or HTTPS protocol.

Select the type of the connection from the dropdown menu:

• *HTTP*—connection between an edge server and an origin where HTTP is always used. It is a default value

• *HTTPS*—connection between an edge server and an origin where HTTPS is always used

• *AUTO*—connection between an edge server and an origin based on a visitor’s request (HTTP or HTTPS)

**IP Access**

Configure a rule to enable/disable access to the CDN resource’s content for a range of IP addresses.

• *Access Policy* – select Disabled to switch off the rule, otherwise, choose between Allow by default/Block by default.

• *Except for IP Addresses* – fill in IP address(es) to which the access policy won’t be applied.

**Secondary CDN Hostnames**

Submit secondary hostnames apart from the default one for HTTP based CDN sites. With these configured, users will be able to access the CDN site using secondary CDN hostname(s). You can add up to 7 secondary CDN hostnames to your CDN resource.

> To be able to use a secondary hostname for a CDN resource with SSL enabled, you need an SSL certificate for your custom hostname. For more details about the purchase of an SSL certificate, contact [OnApp Support](#). Also, set CNAME for both CDN hostname and secondary CDN hostnames.

If you create a CDN resource with the following settings:

• *CDN hostname*—cdn.example.com
• **Origin**—`example.com`

• **Secondary hostname**—`cdn1.example.com`, `cdn2.example.com`

and visitors visit `cdn.example.com`, `cdn1.example.com`, `cdn2.example.com`, all the three URLs will be displayed as `example.com`.

### URL Signing

Protect your files from unauthorized access with a key. A signed URL looks like `http://example.com/filename?hash=DMF1ucDxtqgxwYQ==`.

- **Enable URL Signing** – move the slider to the right to enable it.

- **URL Signing Key** – fill in the key which will be used for URL signing. The secret key is similar to a password and can contain a minimum of 6 to a maximum of 32 characters. Spaces are not allowed.

A signed URL can be in one of the following formats:

- **Query string format**: `http://example.com/filename?secure=DMF1ucDxtqgxwYQ&expires=1546300800&ip=1.2.3.4`

- **Path format**: `http://example.com/secure=DMF1ucDxtqgxwYQ&expires=1546300800&ip=1.2.3.4/filename`

A secure token consists of the following parameters:

- **Expires**—the expiration time of a URL or the time when an URL becomes invalid. The time is passed in the URL itself in a Unix timestamp format and takes part in hash generation. It is an optional parameter

- **Path**—a file path or file directory

For HLS, put a path instead of an M3U8 file, so that all the chunks of the HLS are authenticated as well.

- **Key**—a URL signing key

- **IP**—an IP that provides access. It is optional and only one IP allowed when generating the hash key each time

Here is the format of a secure token:

```xml
<expires><path><key><ip>
```

Here is an example of the PHP script used to generate the hash key:
/**
 * Create hash link CDN resource
 *
 * @param string $cdnResourceUrl
 * The CDN resource URL, eg cdn.yourdomain.com
 * @param string $filePath
 * File path of the CDN resource
 * @param string $secretKey
 * The secret key that is obtained from CDN resource property
 * @param int $expiryTimestamp [optional]
 * UNIX timestamp format, specify how long the hash link is accessible
to the public
 * By default will be accessible forever.
 * @return string URL with generated hash link
 * URL with designated format to access the resource
 *
 * Example:
 * Generate hash link for resource www.example.com/images/photo.png
 * for next 3 days, assume today is Sun, 01 Apr 2012.
 *
 * <?php
 * $hashLink = generateHashLink('www.example.com',
 * '/images/photo.png', 'l33tf0olol', 1333497600);
 * print $hashLink;
 * ?>
 * http://www.example.com/images/photo.png?secure=kaGd_cu6Iy4LDgfX3jy5Rw==,1333497600
 */

function generateHashLink($cdnResourceUrl, $filePath, $secretKey,
$expiryTimestamp = NULL){

    // NOTE [yasir 20110331] + and ? are some of represented chars of
    // based64 encoding (8 bits)
    // + is 62 and / is 63 . and These char should be replaced by
    // other predefined chars.
    $searchChars = array('+','/');
    $replaceChars = array('-','_');

    if($filePath[0] != '/'){
        $filePath = '/{$filePath}';
    }

    if($pos = strpos($filePath, '?')){
        $filePath = substr($filePath, 0, $pos);
    }

    $hashStr = $filePath.$secretKey;

    if($expiryTimestamp){
        $hashStr = $expiryTimestamp.$hashStr;
        $expiryTimestamp = "",{$expiryTimestamp}"
    }

    return "http://{$cdnResourceUrl}{$filePath}?secure=".
        str_replace($searchChars, $replaceChars,
            base64_encode(md5($hashStr, TRUE))).
        $expiryTimestamp;
    }

To generate the hash key, download the Ruby, Python, PHP, or Java script.
Ruby:
ruby UrlSigning.rb -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

Python:
python UrlSigning.py -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

PHP:
php UrlSigning.php -f path -s https -r example.com -p images/photo.png -k abc123s -e 1546300800 -i 1.2.3.4

Java:
javac UrlSigning.java
java UrlSigning -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

Available options:
- **f**: format, path or querystring, **default** = querystring
- **s**: scheme for resource URL, http or https, **default** = http
- **r**: resource hostname (compulsory)
- **p**: file path of the resource, **default** = /
- **k**: URL signing key (compulsory)
- **e**: expiration of the URL (optional)
- **i**: IP that allow to access (optional)

**Cache Expiry**
- *Cache expiry* – set the cache expiry time in minutes (min=1, max=35000000).

**Password**
- *Enable Password* – move the slider to the right to restrict access to the resource (cdn hostname).
- *Unauthorized HTML* – fill in the text which will be displayed for unauthorized login.
- *Username* – choose a username.
- *Password* – select password for the user.

To remove a user, clear both fields.

**Pseudo Streaming**
- *Enable MP4 pseudo streaming* – move the slider to the right to enable the pseudo streaming support for MP4 file type.
- *Enable FLV pseudo streaming* – move the slider to the right to enable pseudo streaming for FVL file type, respectively.

With pseudo streaming enabled, your viewers can seek around a video even if it has not finished downloading. A Flash player and a prepared video are required for pseudo-streaming.
CORS Header
- **Enable CORS headers** - move the slider to the right to enable cross-origin resource sharing (CORS) by adding HTTP header with Access-Control-Allow-Origin: *

Ignore Set-Cookie
- **Ignore Set-Cookie** - move the slider to the right to enable caching content with Set-Cookie response headers.

Nginx Settings
- **Limit rate** - set speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
- **Limit rate after** - the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value -2147483647 KB
- **Proxy cache key** - key for caching. Select one of four supported types from the drop-down list:
  - $host$request_uri
  - $host$uri
  - $proxy_host$request_uri
  - $proxy_host$uri

Search Engine Crawlers
- **Block search engine crawlers** - move the slider to the right to block web crawling bots from indexing the CDN content (for HTTP Pull CDN resources only).

HTTP Live Streaming (HLS) Optimization
- **Enable HLS Optimization** - move the slider to enable/disable HLS optimization. This option is available only for HTTP Pull CDN resources.
- **Enforce Cache Expiry** - tick this checkbox to create an HTTP rule that will enforce cache expiry. After you enable HLS optimization during HTTP Pull resource creation or editing, this check box will not be displayed if you edit the resource as long as the Enable HLS Optimization option is switched on. If you disable the option and then enable it again, this checkbox will be displayed. If an enforce cache expiry rule has already been set for the resource, a new rule will not be created after you check this box and save changes. You can manually add or delete the HTTP rule that will enforce cache expiry on the **HTTP Caching Rules** page of your CDN resource.

To ensure Samsung Smart TV’s compatibility with Microsoft Smooth Streaming through the CDN, the Suppress CDN Headers feature must be enabled.
Contact [OnApp Support](#) to have this feature available in Control Panel.
To be able to remove CDN headers from an HTTP Pull or HTTP Push resource (e.g., X-Storage, X-Edge-IP, and X-Edge-Location), contact OnApp Support to enable this feature.

4. Click **Save** to save the changes.

### 2.4.10 Raw Logs

The raw logs functionality allows you to send logs associated with your CDN resources to your distant server in real time. The raw log allows customers to understand, analyze, and debug files delivered via OnApp CDN, or can be served as audit trailed. Once the user creates CDN resource(s), the raw logs are enabled for this account and the user can configure and receive raw logs. Logs are sent for all the CDN Resources associated with the user. There are three different types of delivery: Syslog, SFTP and FTP. The frequency of uploading the log to client destination is every 10 minutes for SFTP and FTP protocols. For the Syslog protocol, logs are uploaded instantaneously.

If the user does not have any CDN resources, they cannot configure raw logs.

To view and set the raw log configuration:

1. Go to your Control Panel > **CDN** > **Resources** menu.
2. Click the **Raw Log** tab.
3. The page that loads shows the current raw log configuration. On this page you can also set the raw log configuration:
   - For the **FTP/SFTP** delivery protocol:
     - **Hostname** - fill in the hostname of the server to which the log will be delivered
     - **Ftp username** - specify the user name of the FTP/SFTP client on the server to which the log will be delivered
     - **Ftp password** - fill in the password of the FTP/SFTP client on the server to which the log will be delivered
   - For the **Syslog** delivery protocol:
     - **Hostname** - fill in the hostname of the server to which the log will be delivered
     - **Syslog protocol** - select the protocol that will be used for sending the log: TCP or UDP
     - **Syslog port** - specify the port number of the syslog server to which the log will be delivered
   - Choose **Disabled** to disable raw logs.
4. Click **Save** to save the configuration.

To edit, set new configuration parameters and click **Save**.

To disable, select **Disabled** from the raw log configuration delivery protocol drop-down list.
The list of raw log parameters includes the following:

- **$upstream_response_time** - keeps time spent on receiving the response from the upstream server; the time is kept in seconds with millisecond resolution. Times of several responses are separated by commas and colons like addresses in the **$upstream_addr** variable.
- **$upstream_http_x_cache** - a value of the X-Cache HTTP response header field from the upstream server.
- **$request_time** - the time elapsed between the first bytes that have been read from the client and the writing log after the last bytes have been sent to the client.
- **$tcpinfo_rtt** - the round trip time (ping) estimate of the TCP socket in microseconds.
- **$http_referer** - the HTTP request header field that identifies the address of the webpage (i.e., the URI or IRI) linked to the resource requested.
- **$for_operator** - a buyer (if any) subscribed to the edge server.

If the SFTP or FTP protocol is applied, raw logs are delivered as an archive. If the Syslog delivery protocol is selected, user will receive the text of the logs.

There are different log formats for the Syslog and FTP/SFTP delivery protocols.

An example of the HTTP log format (V8) for the Syslog delivery protocol:

```
$time_local<TAB>$edge_server_id<TAB>$remote_addr<TAB>$request_method<TAB>$http_host<TAB>$request_uri<TAB>$status<TAB>$bytes_sent<TAB>$upstream_http_x_cache<TAB>$upstream_response_time<TAB>$request_time<TAB>$tcpinfo_rtt<TAB>$for_operator<TAB>$resource_id<TAB>$server_name<TAB>$server_addr<TAB>$http_user_agent<TAB>$http_referer<TAB>$http_range<TAB>$body_bytes_sent<TAB>$publisher_id<TAB>$tls_version<TAB>$placeholder1<TAB>$placeholder2<TAB>$placeholder3<TAB>$placeholder4<TAB>$placeholder5<TAB>$placeholder6<TAB>$placeholder7<TAB>$V8
```

An example of the Stream log format for the Syslog delivery protocol:

```
date<TAB>time<TAB>edge-id<TAB>type<TAB>x-app<TAB>x-category<TAB>x-event<TAB>x-severity<TAB>x-source<TAB>x-clientid<TAB>x-ip<TAB>x-category<TAB>x-event<TAB>x-duration<TAB>x-streamid<TAB>x-streampos<TAB>x-streamsize<TAB>x-streambytes<TAB>x-comment<TAB>x-sname-query
```
An example of the HTTP log format for FTP/SFTP delivery protocols:

```
$remote_addr - - [${time_local}] "${request_method} ${request_uri} HTTP/1.1"
$http_status $bytes_sent "${http_referer}" "${http_user_agent}"
```

An example of the HTTP log format for FTP/SFTP delivery protocols:

```
1.2.3.4 - - [02/Jan/2020:07:59:42 +0000] "GET /test-file.txt HTTP/1.1" 200
1000 "https://www.test-site.com/test-page" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.79 Safari/537.36"
```

An example of the Stream log format for FTP/SFTP delivery protocols:

```
x-app<TAB>x-severity<TAB>x-category<TAB>x-event<TAB>date<TAB>time<TAB>x-client-id<TAB>x-ip<TAB>x-port<TAB>x-bytes<TAB>sc-bytes<TAB>x-duration<TAB>x-sname<TAB>x-stream-id<TAB>x-spos<TAB>x-stream-bytes<TAB>cs-stream-bytes<TAB>x-file-size<TAB>x-file-length<TAB>x-ctxt<TAB>x-comment<TAB>x-sname-query
```

### 2.4.11 Prefetch Content

This tool allows prepopulating content of an HTTP Pull or HTTP Push CDN resource to the CDN. Recommended only for files which are especially large.

To prefetch content:

1. Go to your Control Panel > Resources > CDN resource’s label.
2. On the page that appears, click the Prefetch tab.
3. In the Paths to Prefetch box, specify the destination path(s). You can enter only one path per line.
4. Click Prefetch.

Please note that you can prefetch only the content of HTTP Pull and Push CDN resources.

Also, on the Prefetch tab, you can view the statuses of your transactions in the table with the following details:

- **Job ID** - the ID of transaction
- **Url** - the Url path to prefetch
- **Status** - the status of transaction ( % of progress, completed or failed)
- **Time Initiated** - the time when transaction was started
- **Time Completed** - the time when transaction was completed
2.4.12 Purge Content

This tool allows instant removal of HTTP Pull and HTTP Push cache content in the CDN, if newly updated content has not been properly replicated.

To purge a single file:
1. Go to your Control Panel > CDN > Resources > CDN resource's label.
2. On the page that appears, click the Purge tab.
3. In the Paths to Purge box, specify the destination path(s). You can enter only one path per line.

The maximum capacity of the Paths to Purge box is 1,000 lines.

4. Click Purge.

To purge all content:
1. Go to your Control Panel > CDN > Resources > CDN resource's label.
2. On the page that appears, click the Purge tab.
3. Click Purge All Contents of This Site to purge all content.

Please note that you can purge only the content of HTTP Pull and HTTP Push CDN resources.

Also, on the Purge tab, you can view the statuses of your transactions in the table with the following details:

- **Job ID** - the ID of transaction
- **Url** - the Url path to purge
- **Status** - the status of transaction (queued, % of progress, completed or failed)
- **Time Initiated** - the time when transaction was started
- **Time Completed** - the time when transaction was completed

Be aware that statuses for PurgeAll transactions are not tracked.

Because objects are not stored in the cache directory structure, it is not possible to purge URLs with a wildcard or a directory as in the example below:

```
/path/to/file/*.jpg
/path/to/directory/
```
It is recommended to add a version ID to your URL query string. Apart from invalidating edge server cache, it also invalidates browser cache.

2.4.13 CDN Billing Statistics

OnApp has a record of all the charges applied to your CDN resources. You can view the resource statistics under the statistics available or those for a shorter period by setting a Start and End time.

To view billing statistics for a CDN resource:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the label of the resource you're interested in and then click the Billing Statistics tab.
3. Set Start and End time.
4. Move the Show in my Timezone slider to the right to show billing statistics according to your profile's timezone settings.
5. Press the Apply button.
6. On the screen that appears, you will see the following billing statistics details:
   - Date – particular date and time for the generated statistics
   - Edge Group - the edge group to which the CDN resource belongs to.
   - Traffic - resource traffic in MB.
   - Cost – the total due for the CDN resource at the point of time specified in the Date column.

Scroll down to see Total Amount (the total due for the whole billing statistics period).

2.4.14 Wildcard Invalidation Rules

Wildcard invalidation is an effective tool that allows you to remove certain folders or all the CSS files from the cache. You can purge all files in a folder, all files and folders that start with certain characters and all files of a certain format (e.g. PNG) in a folder. You can launch the invalidation process multiple times for a rule you specify. Wildcard invalidation is available only for HTTP Pull and HTTP Push resources and you can set up to five wildcard invalidation rules for a resource.

Users need to have either 'Update any CDN resource' or 'Update own CDN resources' permission to access the Wildcard Invalidations page.

On this page:

- View Wildcard Invalidations
- Create Wildcard Invalidation Rules
- Invalidate Files Again
- Delete Wildcard Invalidation Rules
2.4.14.1 View Wildcard Invalidations

To view wildcard invalidation rules:
1. Go to the Control Panel > CDN > Resources menu.
2. Click the required resource label.
3. On the page that appears, click the Wildcard Invalidations tab.
4. You will see the list of all Wildcard invalidation rules with the following information:
   - Path - the URL path to the cached files that are to be invalidated
   - Last Invalidation - the time when the last invalidation occurred

2.4.14.2 Create Wildcard Invalidation Rules

To add a wildcard invalidation rule:
1. Go to the Control Panel > CDN > Resources menu.
2. Click the required resource label.
3. On the page that appears, click the Wildcard Ininvalidations tab.
4. Click the "+" button.
5. In the input field, specify the path to the cached files that are to be invalidated. You may indicate only one path per line. You can set up to five invalidation rules per resource.
   - if you set, for example, /abc/* - all files in the /abc/ folder will be invalidated
   - if you set, for example, /abc*/ - all files and folders the label of which start with abc will be invalidated
   - if you set, for example, /img/*.png - all PNG files in the img folder will be invalidated
6. Click the + button to add the rule. Once you add the rule, the files for which you have indicated the path in this rule, will be invalidated.

2.4.14.3 Invalidate Files Again

You can launch the invalidation process multiple times for a rule you have added. To invalidate again the files for which rules have been set:
1. Go to the Control Panel > CDN > Resources menu.
2. Click the required resource link.
3. On the page that appears, click the Wildcard Invalidations tab.
4. Click the button. Once you click this button, the files for which you have indicated the path in this rule, will be invalidated.

2.4.14.4 Delete Wildcard Invalidation Rules

To delete wildcard invalidation rules:
1. Go to the Control Panel > CDN > Resources menu.
2. Click the required resource link.
3. On the page that appears, click the Wildcard Invalidations tab.
4. Click the Delete button.

2.4.15 HTTP Caching Rules

The HTTP Rules engine allows users to customize the CDN edge server behavior, e.g. how the CDN will manage cache and redirection. This creates a lot of flexibility and adaptability for different platform websites to integrate with CDN without much hassle. HTTP rules are set in OnApp Control Panel and are specific to each CDN resource.

A CDN resource can be configured with up to 100 rules. This is collectively called a ruleset. A rule consists of conditions and actions. A condition consists of a subject, which determines the value to select, and a predicate, which specifies what to compare the subject against. Conditions are bonded by the connectives “AND” or “OR”. When all the conditions are met, the CDN edge server will perform the actions associated with the rule. The CDN edge server processes these rules from top to bottom order. It ends processing on the first match.

2.4.15.1 Limitations
- You can create up to 100 rules per resource
- You can set up to 100 actions per rule
- Values can be up to 1000 characters long
- Rule processing ends after the first match
- You can set the rules for HTTP Pull and HTTP Push resources

For an HTTP Pull caching mechanism, a file larger than 2 GB is served through edge servers but not cached (not even the first 2 GB). If your files are larger than 200 MB, it is recommended to do advance prefetching for smoother delivery.

2.4.15.2 Set HTTP Rules

To create a rule, you have to specify the required conditions and the action which should be performed if the condition is met.

To set HTTP rules:
1. Log in to your OnApp Control Panel.
2. Click the Resources under the CDN menu.
3. On the page that appears, click the required resource label.
4. On the following page, click the HTTP Caching Rules tab.
5. Click Create New Rules button.
6. Give the Name to your rule.
7. Set the Conditions: specify the Connective/Subject/Predicate/Value from the drop-down menu.
8. To specify additional conditions, click the plus button next to the Conditions header.
9. Choose the appropriate action to take place from the Actions drop-down.
10. Click the plus button next to the Actions header to add some more actions which should take place when the conditions are met.
11. Click **Save**.

Refer to the following sections for details on subjects, predicates, values and actions.

### 2.4.15.3 Edit HTTP Rules
To edit an already created HTTP rule:
1. Log in to your OnApp Control Panel.
2. Click the **Resources** under the CDN menu.
3. On the page that appears, click the label of the required resource.
4. On the following page, click the **HTTP Caching Rules** tab.
5. Click the **Actions** icon and choose **Edit** next to a required rule.
6. Make all necessary changes.
7. Click **Save**.

### 2.4.15.4 Delete HTTP Rule
To delete an HTTP rule:
1. Log in to your OnApp Control Panel.
2. Click the **Resources** under the CDN menu.
3. On the page that appears, click the label of the required resource.
4. On the following page, click the **HTTP Caching Rules** tab.
5. Click the **Actions** icon and choose **Delete** next to a required rule.

### 2.4.15.5 The List of Subjects

<table>
<thead>
<tr>
<th>Subject</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Matches all remaining requests. When used, it should be the last rule in the ruleset.</td>
</tr>
</tbody>
</table>
| URL | Selects the URL part of the request. It excludes the query string.  
**Example:**  
Client requests: http://cdn.example.com/image.jpg  
Selected value: “/image.jpg” |
| IP | Selects the IP address of the client. If the clients use a proxy server, the IP of their proxy server which made the request to the edge server will be selected.  
**Example**  
“192.0.2.43” |
| Cookie | Selects the value of a specific cookie sent by the client.  
**Example**  
Cookie chosen: “logged_in”  
Client request header: “Cookie: session_id=abcdef; logged_in=1; cart_id=defabc”  
Selected value: “1” |
<p>| Country | Selects the client’s two-letter country code. If the client’s country cannot be derived from their IP, the value “” is selected. |</p>
<table>
<thead>
<tr>
<th>Subject</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong></td>
<td>Client's IP: 193.113.9.162</td>
</tr>
<tr>
<td></td>
<td>Selected value: “GB”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Param</strong></th>
<th>Selects the value of a specific query string parameter. If there are multiple identical keys, the last value is selected.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong></td>
<td>Parameter chosen: “page”</td>
</tr>
<tr>
<td></td>
<td>Client requests: <a href="http://cdn.example.com/index.php?page=about&amp;id=53">http://cdn.example.com/index.php?page=about&amp;id=53</a></td>
</tr>
<tr>
<td></td>
<td>Selected value: “about”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Extension</strong></th>
<th>Selects the file extension of the request. If the request filename does not contain a dot, then the value “” is selected.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong></td>
<td>Client requests: <a href="http://cdn.example.com/image.jpg">http://cdn.example.com/image.jpg</a></td>
</tr>
<tr>
<td></td>
<td>Selected value: “jpg”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Header</strong></th>
<th>Selects the value of a specific client request header. If the request header does not exist, then the value “” is selected.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong></td>
<td>Header chosen: “User-Agent”</td>
</tr>
<tr>
<td></td>
<td>Client sends header: “User-Agent: Mozilla/5.0 (Windows NT 6.3) Firefox/30.0”</td>
</tr>
<tr>
<td></td>
<td>Selected value: “Mozilla/5.0 (Windows NT 6.3) Firefox/30.0”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Scheme</strong></th>
<th>Selects the scheme part of the request. It can be either http or https.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong></td>
<td>Client requests: <a href="http://cdn.example.com/image.jpg">http://cdn.example.com/image.jpg</a></td>
</tr>
<tr>
<td></td>
<td>Selected value: “http”</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>Client requests: <a href="https://secure.example.com/image.jpg">https://secure.example.com/image.jpg</a></td>
</tr>
<tr>
<td></td>
<td>Selected value: “https”</td>
</tr>
</tbody>
</table>

### 2.4.15.6 The List of Predicates
Note that all predicates are case-insensitive.

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Compares the subject to an exact value.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>URL “/index.php”</td>
</tr>
<tr>
<td></td>
<td>Equals “/index.php”</td>
</tr>
<tr>
<td></td>
<td>Result: TRUE</td>
</tr>
<tr>
<td></td>
<td>URL “/ExampleFile.txt”</td>
</tr>
</tbody>
</table>
## Predicate Details

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>“/examplefile.txt”</td>
</tr>
<tr>
<td>Result</td>
<td>TRUE</td>
</tr>
<tr>
<td>URL</td>
<td>“/image.jpg”</td>
</tr>
<tr>
<td>Equals</td>
<td>“/index.php”</td>
</tr>
<tr>
<td>Result</td>
<td>FALSE</td>
</tr>
<tr>
<td>Starts with</td>
<td>Compares whether the subject starts with a value.</td>
</tr>
<tr>
<td>Example</td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td>“192.0.2.54”</td>
</tr>
<tr>
<td>Starts With</td>
<td>“192.0.2.”</td>
</tr>
<tr>
<td>Result</td>
<td>TRUE</td>
</tr>
<tr>
<td>URL</td>
<td>“/images/files.jpg”</td>
</tr>
<tr>
<td>Starts With</td>
<td>“/images/”</td>
</tr>
<tr>
<td>Result</td>
<td>TRUE</td>
</tr>
<tr>
<td>IP</td>
<td>“192.5.54.3”</td>
</tr>
<tr>
<td>Starts With</td>
<td>“192.0.2.”</td>
</tr>
<tr>
<td>Result</td>
<td>FALSE</td>
</tr>
<tr>
<td>Ends with</td>
<td>Compares whether the subject ends with a value.</td>
</tr>
<tr>
<td>Example</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>“/images/files.jpg”</td>
</tr>
<tr>
<td>Ends With</td>
<td>“.jpg”</td>
</tr>
<tr>
<td>Result</td>
<td>TRUE</td>
</tr>
<tr>
<td>URL</td>
<td>“/videos/video.mp4”</td>
</tr>
<tr>
<td>Ends With</td>
<td>“.jpg”</td>
</tr>
<tr>
<td>Result</td>
<td>FALSE</td>
</tr>
<tr>
<td>In List</td>
<td>Compares the subject to the list of values. Each value is separated by a single space.</td>
</tr>
<tr>
<td>Example</td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>“GB”</td>
</tr>
<tr>
<td>In List</td>
<td>“GB ES FR DE”</td>
</tr>
<tr>
<td>Result</td>
<td>TRUE</td>
</tr>
<tr>
<td>Country</td>
<td>“US”</td>
</tr>
<tr>
<td>In List</td>
<td>“GB ES FR DE”</td>
</tr>
<tr>
<td>Result</td>
<td>FALSE</td>
</tr>
<tr>
<td>Matches wildcard</td>
<td>Compares whether the subject matches a wildcard value. The wildcard character “<em>” matches any 0 or more characters. Multiple “</em>”s can be specified.</td>
</tr>
<tr>
<td>Example</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>“/images/photos/photo.jpg”</td>
</tr>
<tr>
<td>Matches Wildcard</td>
<td>“/images/<em>.</em>”</td>
</tr>
<tr>
<td>Predicate</td>
<td>Details</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Result TRUE</td>
</tr>
<tr>
<td></td>
<td>Url “/images/videos/video.mp4”</td>
</tr>
<tr>
<td></td>
<td>Matches Wildcard “/images/*.jpg”</td>
</tr>
<tr>
<td></td>
<td>Result FALSE</td>
</tr>
<tr>
<td></td>
<td>Url “/archives/2014/news/index.html”</td>
</tr>
<tr>
<td></td>
<td>Matches Wildcard “/*2014/news/**”</td>
</tr>
<tr>
<td></td>
<td>Result TRUE</td>
</tr>
<tr>
<td>Does not equal</td>
<td>Opposite of the Equals value</td>
</tr>
<tr>
<td>Does not start with</td>
<td>Opposite of the Starts with value</td>
</tr>
<tr>
<td>Does not end with</td>
<td>Opposite of the Ends with value</td>
</tr>
<tr>
<td>Is not in list</td>
<td>Opposite of the In list value</td>
</tr>
<tr>
<td>Does not match wildcard</td>
<td>Opposite of the Matches wildcard value</td>
</tr>
</tbody>
</table>

### 2.4.15.7 The List of Country Codes

Here is the list of all the country codes which you can set as a subject Country in your HTTP rules:

- AD  Andorra
- AE  United Arab Emirates
- AF  Afghanistan
- AG  Antigua and Barbuda
- AI  Anguilla
- AL  Albania
- AM  Armenia
- AO  Angola
- AP  Asia/Pacific Region
- AQ  Antarctica
- AR  Argentina
- AS  American Samoa
- AT  Austria
- AU  Australia
- AW  Aruba
- AX  Aland Islands
- AZ  Azerbaijan
- BA  Bosnia and Herzegovina
- BB  Barbados
<table>
<thead>
<tr>
<th>Code</th>
<th>Country Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD</td>
<td>Bangladesh</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
</tr>
<tr>
<td>BH</td>
<td>Bahrain</td>
</tr>
<tr>
<td>BI</td>
<td>Burundi</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
</tr>
<tr>
<td>BL</td>
<td>Saint Barthelemy</td>
</tr>
<tr>
<td>BM</td>
<td>Bermuda</td>
</tr>
<tr>
<td>BN</td>
<td>Brunei Darussalam</td>
</tr>
<tr>
<td>BO</td>
<td>Bolivia</td>
</tr>
<tr>
<td>BQ</td>
<td>Bonaire, Saint Eustatius and Saba</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
</tr>
<tr>
<td>BS</td>
<td>Bahamas</td>
</tr>
<tr>
<td>BT</td>
<td>Bhutan</td>
</tr>
<tr>
<td>BV</td>
<td>Bouvet Island</td>
</tr>
<tr>
<td>BW</td>
<td>Botswana</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
</tr>
<tr>
<td>BZ</td>
<td>Belize</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
</tr>
<tr>
<td>CC</td>
<td>Cocos (Keeling) Islands</td>
</tr>
<tr>
<td>CD</td>
<td>Congo, The Democratic Republic of the</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
</tr>
<tr>
<td>CI</td>
<td>Cote d'Ivoire</td>
</tr>
<tr>
<td>CK</td>
<td>Cook Islands</td>
</tr>
<tr>
<td>CL</td>
<td>Chile</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
</tr>
<tr>
<td>CO</td>
<td>Colombia</td>
</tr>
<tr>
<td>CR</td>
<td>Costa Rica</td>
</tr>
<tr>
<td>CU</td>
<td>Cuba</td>
</tr>
<tr>
<td>CV</td>
<td>Cape Verde</td>
</tr>
<tr>
<td>CW</td>
<td>Curacao</td>
</tr>
<tr>
<td>CX</td>
<td>Christmas Island</td>
</tr>
<tr>
<td>CY</td>
<td>Cyprus</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
</tr>
<tr>
<td>DJ</td>
<td>Djibouti</td>
</tr>
<tr>
<td>Code</td>
<td>Country</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
</tr>
<tr>
<td>DM</td>
<td>Dominica</td>
</tr>
<tr>
<td>DO</td>
<td>Dominican Republic</td>
</tr>
<tr>
<td>DZ</td>
<td>Algeria</td>
</tr>
<tr>
<td>EC</td>
<td>Ecuador</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
</tr>
<tr>
<td>EG</td>
<td>Egypt</td>
</tr>
<tr>
<td>EH</td>
<td>Western Sahara</td>
</tr>
<tr>
<td>ER</td>
<td>Eritrea</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
</tr>
<tr>
<td>ET</td>
<td>Ethiopia</td>
</tr>
<tr>
<td>EU</td>
<td>Europe</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
</tr>
<tr>
<td>FJ</td>
<td>Fiji</td>
</tr>
<tr>
<td>FK</td>
<td>Falkland Islands (Malvinas)</td>
</tr>
<tr>
<td>FM</td>
<td>Micronesia, Federated States of</td>
</tr>
<tr>
<td>FO</td>
<td>Faroe Islands</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
</tr>
<tr>
<td>GB</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>GD</td>
<td>Grenada</td>
</tr>
<tr>
<td>GE</td>
<td>Georgia</td>
</tr>
<tr>
<td>GF</td>
<td>French Guiana</td>
</tr>
<tr>
<td>GG</td>
<td>Guernsey</td>
</tr>
<tr>
<td>GH</td>
<td>Ghana</td>
</tr>
<tr>
<td>GI</td>
<td>Gibraltar</td>
</tr>
<tr>
<td>GL</td>
<td>Greenland</td>
</tr>
<tr>
<td>GM</td>
<td>Gambia</td>
</tr>
<tr>
<td>GN</td>
<td>Guinea</td>
</tr>
<tr>
<td>GP</td>
<td>Guadeloupe</td>
</tr>
<tr>
<td>GQ</td>
<td>Equatorial Guinea</td>
</tr>
<tr>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>GS</td>
<td>South Georgia and the South Sandwich Islands</td>
</tr>
<tr>
<td>GT</td>
<td>Guatemala</td>
</tr>
<tr>
<td>GU</td>
<td>Guam</td>
</tr>
<tr>
<td>GW</td>
<td>Guinea-Bissau</td>
</tr>
<tr>
<td>GY</td>
<td>Guyana</td>
</tr>
<tr>
<td>HK</td>
<td>Hong Kong</td>
</tr>
<tr>
<td>HM</td>
<td>Heard Island and McDonald Islands</td>
</tr>
<tr>
<td>HN</td>
<td>Honduras</td>
</tr>
</tbody>
</table>
HR  Croatia
HT  Haiti
HU  Hungary
ID  Indonesia
IE  Ireland
IL  Israel
IM  Isle of Man
IN  India
IO  British Indian Ocean Territory
IQ  Iraq
IR  Iran, Islamic Republic of
IS  Iceland
IT  Italy
JE  Jersey
JM  Jamaica
JO  Jordan
JP  Japan
KE  Kenya
KG  Kyrgyzstan
KH  Cambodia
KI  Kiribati
KM  Comoros
KN  Saint Kitts and Nevis
KP  Korea, Democratic People's Republic of
KR  Korea, Republic of
KW  Kuwait
KY  Cayman Islands
KZ  Kazakhstan
LA  Lao People's Democratic Republic
LB  Lebanon
LC  Saint Lucia
LI  Liechtenstein
LK  Sri Lanka
LR  Liberia
LS  Lesotho
LT  Lithuania
LU  Luxembourg
LV  Latvia
LY  Libyan Arab Jamahiriya
MA  Morocco
MC  Monaco
MD  Moldova, Republic of
ME  Montenegro
MF  Saint Martin
MG  Madagascar
MH  Marshall Islands
MK  Macedonia
ML  Mali
MM  Myanmar
MN  Mongolia
MO  Macao
MP  Northern Mariana Islands
MQ  Martinique
MR  Mauritania
MS  Montserrat
MT  Malta
MU  Mauritius
MV  Maldives
MW  Malawi
MX  Mexico
MY  Malaysia
MZ  Mozambique
NA  Namibia
NC  New Caledonia
NE  Niger
NF  Norfolk Island
NG  Nigeria
NI  Nicaragua
NL  Netherlands
NO  Norway
NP  Nepal
NR  Nauru
NU  Niue
NZ  New Zealand
OM  Oman
PA  Panama
PE  Peru
PF  French Polynesia
PG  Papua New Guinea
PH  Philippines
<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>Pakistan</td>
</tr>
<tr>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>PM</td>
<td>Saint Pierre and Miquelon</td>
</tr>
<tr>
<td>PN</td>
<td>Pitcairn</td>
</tr>
<tr>
<td>PR</td>
<td>Puerto Rico</td>
</tr>
<tr>
<td>PS</td>
<td>Palestinian Territory</td>
</tr>
<tr>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>PW</td>
<td>Palau</td>
</tr>
<tr>
<td>PY</td>
<td>Paraguay</td>
</tr>
<tr>
<td>QA</td>
<td>Qatar</td>
</tr>
<tr>
<td>RE</td>
<td>Reunion</td>
</tr>
<tr>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>RS</td>
<td>Serbia</td>
</tr>
<tr>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>RW</td>
<td>Rwanda</td>
</tr>
<tr>
<td>SA</td>
<td>Saudi Arabia</td>
</tr>
<tr>
<td>SB</td>
<td>Solomon Islands</td>
</tr>
<tr>
<td>SC</td>
<td>Seychelles</td>
</tr>
<tr>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>SG</td>
<td>Singapore</td>
</tr>
<tr>
<td>SH</td>
<td>Saint Helena</td>
</tr>
<tr>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td>SJ</td>
<td>Svalbard and Jan Mayen</td>
</tr>
<tr>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>SL</td>
<td>Sierra Leone</td>
</tr>
<tr>
<td>SM</td>
<td>San Marino</td>
</tr>
<tr>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>SO</td>
<td>Somalia</td>
</tr>
<tr>
<td>SR</td>
<td>Suriname</td>
</tr>
<tr>
<td>SS</td>
<td>South Sudan</td>
</tr>
<tr>
<td>ST</td>
<td>Sao Tome and Principe</td>
</tr>
<tr>
<td>SV</td>
<td>El Salvador</td>
</tr>
<tr>
<td>SX</td>
<td>Sint Maarten</td>
</tr>
<tr>
<td>SY</td>
<td>Syrian Arab Republic</td>
</tr>
<tr>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>TC</td>
<td>Turks and Caicos Islands</td>
</tr>
<tr>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>TF</td>
<td>French Southern Territories</td>
</tr>
<tr>
<td>TG</td>
<td>Togo</td>
</tr>
</tbody>
</table>
TH  Thailand
TJ  Tajikistan
TK  Tokelau
TL  Timor-Leste
TM  Turkmenistan
TN  Tunisia
TO  Tonga
TR  Turkey
TT  Trinidad and Tobago
TV  Tuvalu
TW  Taiwan
TZ  Tanzania, United Republic of
UA  Ukraine
UG  Uganda
UM  United States Minor Outlying Islands
US  United States
UY  Uruguay
UZ  Uzbekistan
VA  Holy See (Vatican City State)
VC  Saint Vincent and the Grenadines
VE  Venezuela
VG  Virgin Islands, British
VI  Virgin Islands, U.S.
VN  Vietnam
VU  Vanuatu
WF  Wallis and Futuna
WS  Samoa
YE  Yemen
YT  Mayotte
ZA  South Africa
ZM  Zambia
ZW  Zimbabwe

2.4.15.8  The List of Actions
Here is the list of all the actions for HTTP Rules:

<table>
<thead>
<tr>
<th>Action</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force Edge To Never Cache</td>
<td>Forces the CDN edge server to never cache the request. However, if the request is already cached (for example, if it was cached prior to setting up this rule), it will not be forced out of cache.</td>
</tr>
<tr>
<td>Action</td>
<td>Details</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Force Client To Never Cache</td>
<td>Forces the client to never cache the request. This is achieved by removing all Cache-Control and Expires response headers, sending instead “Cache-Control: no-cache”.</td>
</tr>
<tr>
<td>Force Edge To Cache</td>
<td>Forces the CDN edge server to cache the request for a specified duration. This overrides any Cache-Control or Expires headers from the origin, even if they specify “private” or “no-cache”.</td>
</tr>
<tr>
<td></td>
<td>The value must be 1 second or longer.</td>
</tr>
<tr>
<td>Force Client To Cache</td>
<td>Forces the client to cache the request for a specified duration. This is achieved by removing all Cache-Control and Expires response headers, sending instead “Cache-Control: max-age=...”.</td>
</tr>
<tr>
<td></td>
<td>The value must be 1 second or longer.</td>
</tr>
<tr>
<td>Override Default Cache Validity</td>
<td>Changes the CDN edge server’s default validity period for origin responses that do not explicitly specify Cache-Control or Expires. Responses with Cache-Control or Expires headers are still honored.</td>
</tr>
<tr>
<td></td>
<td>The value must be 1 second or longer.</td>
</tr>
<tr>
<td>Forbid Client</td>
<td>Returns a simple 403 Forbidden response to the client.</td>
</tr>
<tr>
<td>Redirect Client from HTTP to HTTPS</td>
<td>Returns a 301 Redirect response from http address to https address instead of loading from cache or upstream.</td>
</tr>
<tr>
<td>Redirect Client</td>
<td>Returns a 302 Redirect response to the client, to the specified URL. The URL must be specified in full, starting with http:// or https://</td>
</tr>
<tr>
<td>Set Request Header</td>
<td>Overrides a request header to the origin. A header name and header value must be provided.</td>
</tr>
<tr>
<td>Set Response Header</td>
<td>Overrides a response header to the client. A header name and header value must be provided.</td>
</tr>
<tr>
<td>Set Client IP In Request Header</td>
<td>Sets the client’s IP address in a request header to the origin. A header name must be provided.</td>
</tr>
<tr>
<td>Prepend Origin Directory</td>
<td>Prepends a directory to the URL when the edge server requests it from the origin. Examples: \ Prepend origin directory “images” \ Client requests to edge: <a href="http://cdn.example.com/photo.jpg">http://cdn.example.com/photo.jpg</a> \ Edge requests to origin: <a href="http://cdn.example.com/images/photo.jpg">http://cdn.example.com/images/photo.jpg</a></td>
</tr>
<tr>
<td></td>
<td>Prepend origin directory “/some/sub%20directory/” \ Client requests to edge: <a href="http://cdn.example.com/some/file.txt">http://cdn.example.com/some/file.txt</a> \ Edge requests to origin: <a href="http://cdn.example.com/some/sub%20directory/some/file.txt">http://cdn.example.com/some/sub%20directory/some/file.txt</a></td>
</tr>
<tr>
<td></td>
<td>A value must be provided (it cannot be empty).</td>
</tr>
</tbody>
</table>
Leading and trailing slashes from the directory are automatically stripped.

Special characters, such as spaces, must be percent-encoded.

Set Custom Origin

Overrides the origin that the edge server connects to.

Example:

Resource has origin “3.3.3.3”.

A rule is added such that if a URL starts with “/images/”, it sets a custom origin to “5.5.5.5”.

A value must be provided (it cannot be empty) and it must be a valid hostname or IP.

A custom origin port cannot be specified or overridden.

“Set Request Header” may be used in conjunction with this to set a correct Host header.

Passthrough HTTP Host Header to Origin

Goes back to the origin when the condition = true, during the first time, and the subsequent request (being cached in edge) will not go back to the origin again although the condition = true.

CDN edge server will only go back to the origin when the file does not exist (cache) in the edge server (for ALL scenario), and the feature ”passthrough http host header to origin” happens during the path from cdn edge --> origin.

This feature is not about ”making all requests goes back to origin”, but ”sending the host header info to origin”.

2.4.16 Web Application Firewall (WAF)

Web Application Firewall (WAF) monitors, filters or blocks the traffic to and from a web application. A WAF inspects every HTML, HTTPS, SOAP and XML-RPC data packet. This firewall enables preventing attacks such as cross-site scripting (XSS), SQL injection, session hijacking, and buffer overflows through customizable web security rules.

You need to have the Update any CDN resource or Update own CDN resources permission enabled to access the WAF menu of a CDN resource.

See also:

View CDN Resources
Create HTTP CDN Resource
Raw Logs
Token Authentication
2.4.16.1 Enable/Disable WAF Rules on CDN Resources

If you want to enable/disable WAF protection on your CDN resource, follow the procedure below:

1. Go to Control panel's **CDN Resources** menu.
2. Click the required resource label.
3. On the page that appears, click the **WAF** tab.
4. Move the **WAF** slider to enable/disable the protection. The menu allows you to turn on/off the following OWASP rules:

   - **Drupal Exclusion Rules Request** - these rules disable CRS on a set of well-known parameter fields that are often the source of false positives or false alarms of the CRS. This includes the session cookie, the password fields and article/node bodies.
   - **Wordpress Exclusion Rules Request** - these exclusions remedy false positives in a default WordPress install.
   - **IP Reputation Request** - these rules deal with detecting traffic from IPs that have previously been involved in malicious activity, either on our local site or globally.
   - **Method Enforcement Request** - this rule detects HTTP Request Method Anomalies.
   - **DDoS Protection Request** - these rules will attempt to detect some level 7 DoS (Denial of Service) attacks against your server.
   - **Scanner Detection Request** - these rules are concentrated around detecting security tools and scanners.
   - **Protocol Enforcement Request** - these rules center around detecting requests that either violate HTTP or represent a request that no modern browser would generate, for instance missing a user-agent.
   - **Protocol Attack Request** - these rules focus on specific attacks against the HTTP protocol itself such as HTTP Request Smuggling and Response Splitting.
   - **Application Attack LFI Request** - these rules attempt to detect when a user is trying to include a file that would be local to the webserver that they should not have access to. Exploiting this type of attack can lead to the web application or server being compromised.
   - **Application Attack RFI Request** - these rules attempt to detect when a user is trying to include a remote resource into the web application that will be executed. Exploiting this type of attack can lead to the web application or server being compromised.
   - **Application Attack RCE Request** - this rule detects Unix command injections and protects against Remote Code Execution
   - **Application Attack PHP Request** - this rule detects PHP open tags "<?" and "<?php". Also detects "[/php]", "[php]" and "[php]" tags used by some applications to indicate PHP dynamic content.
   - **Application Attack XSS Request** - these rules are intended to prevent all cross-site scripting (XSS) attacks in your CDN resources and the CDN Accelerator.
• **Application Attack Sql Request** - these rules protect against common initial SQL injection attacks where attackers insert quote characters to the existing normal payload to see how the database responds.

• **Application Attack Session Fixation Request** - these rules focus around providing protection against Session Fixation attacks.

• **Blocking Evaluation Request** - these rules provide the anomaly based blocking for a given request.

• **Data Leakages Response** - these rules provide protection against data leakages that may occur generally.

• **Data Leakages SQL Response** - these rules provide protection against data leakages that may occur from backend SQL servers. Often these are indicative of SQL injection issues being present.

• **Data Leakages Java Response** - these rules provide protection against data leakages that may occur because of Java.

• **Data Leakages PHP Response** - these rules provide protection against data leakages that may occur because of PHP.

• **Data Leakages IIS Response** - these rules provide protection against data leakages that may occur because of Microsoft IIS.

• **Blocking Evaluation Response** - these rules provide the anomaly based blocking for a given response.

• **Correlation Response** - these rules facilitate the gathering of data about successful and unsuccessful attacks on the server.

Note that you can enable a maximum of 10 WAF rules per CDN resource.

5. Click the **Update** button to save the changes.

### 2.4.17 CDN Token Authentication

Token authentication helps to protect CDN streams from being snitched. Similar to HTTP URL signing, this feature allows customers to enter a secret key during setting up a CDN resource. Then, customer can use secret key, along with expiry date and allowed/blocked referrer site to generate the token from a script.

To enable token authentication:

1. Enter your secret key and secure path using OnApp UI.
2. Download .NET or Java token generator.
3. Generate the token with the secret key, allow referrer, deny referrer, expiry date, and allow IP.

For more information, refer to the readme file on the token generator.

Missing can be applied to the allow referrer if the HTTP referrer is missing from the browser to the video player. In some cases, the referrer header the video player sent to the edge server is the JS script domain instead of the website domain.
4. Append the token with your stream URL, eg ?token=
   110ea31ac69c09a2db0bddd74238843631cdab498ff7e6e75c6d99cc4d05426ab679a57015d
   4e48438c97b921652daec62de3829f8ff437e27449cfd9fc65e91a57015d

5. Embed with your website.

2.4.17.1 Set up Token Authentication in UI
You can set up token authentication for Video on demand and Live streaming CDN resources.
To do so:

2. Now proceed with running the .NET or JAVA scripts.

2.4.17.2 Run Token Generator
Download the script from the following locations:
- .NET script: https://bitbucket.org/onappcore/cdn-wowza-token-tool/src/f06c7cc4842a9854ba6759a7ef18191cc2dd60e7/dotnet/?at=master
- Java script: https://bitbucket.org/onappcore/cdn-wowza-token-tool/src/f06c7cc4842a9854ba6759a7ef18191cc2dd60e7/java/?at=master

Refer to the following sections on instructions for running the scripts.

2.4.17.2.1 Generate Token Using .NET

Prerequisites:
- .NET Framework 4.5
- BouncyCastle C# Crypto library 1.7
  (http://www.bouncycastle.org/csharp/)

On this page:
- **Build**
- **Usage**
  - Security Parameters
  - Allow blank/missing referrer
  - Generate Token
  - Decrypt token

2.4.17.2.1.1 Build
To build a generator:
1. Go to /cdn-wowza-token-tool/dotnet/src location
2. Run xbuild
Upon success of the build, you will find the .exe (TokenAuthGenerator.exe) file at the 'TokenAuthGenerator/bin/Debug' folder.

2.4.17.2.1.2 Usage
TokenAuthGenerator.exe (encrypt | decrypt) (<primary_key> | <backup_key>) "<security_parameters>"

2.4.17.2.1.2.1 Security Parameters
expire
- Number of seconds since Unix time (Epoch time)
- UTC based
- Must not be earlier than current time
ref_allow
- Referrer domain (e.g. google.com) or path (e.g. google.com/video/)
- Allow multiple referrers separated by comma (,) without space(s)
- Wildcard (*) allowed only at the beginning of a referrer, e.g. *.DOMAIN
- Do not append space at the start & end of a referrer
- Domain must fulfill RFC 3490
- Path must fulfill RFC 2396
- Should not include port (e.g. google.com:3000/video)
- Should not include protocol(e.g. http) portion
ref_deny
- The same rules as for ref_allow

If both ref_allow & ref_deny are specified, ref_allow will be taking precedence over ref_deny

2.4.17.2.1.2.2 Allow blank/missing referrer
Both "ref_allow" & "ref_deny" could be configured to allow/deny blank or missing referrer during TokenAuth validation.

The following configuration allows blank or missing referrer:
ref_allow=allow.com,
ref_allow=allow.com,MISSING
ref_deny=deny.com

The following configuration deny blank or missing referrer:
ref_allow=allow.com
ref_deny=deny.com,
ref_deny=deny.com, MISSING

Normally ref_allow & ref_deny should not be used together, but if this happened ref_allow will take precedence over ref_deny.
2.4.17.2.1.2.3 Generate Token
To generate token, run the following:
TokenAuthGenerator.exe encrypt samplekey
"expire=1598832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com"

Sample Output:

```
token=110ea31ac69c09a2db0bddd74238843631cdab498ff7e6e75cbd99cc4d05426ab679a57015d4e48438c97b921652daec62de3829f8ff437e27449cfdfc2f1e5d9fc47f14e91a51e8a7
```

After generating a token, append the result to the playback URL.

2.4.17.2.1.2.4 Decrypt token
To decrypt a token, run the following:
TokenAuthGenerator.exe decrypt samplekey
110ea31ac69c09a2db0bddd74238843631cdab498ff7e6e75cbd99cc4d05426ab679a57015d4e48438c97b921652daec62de3829f8ff437e27449cfdfc2f1e5d9fc47f14e91a51e8a7

Output example:

```
security parameters=expire=1598832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com
```

2.4.17.2 Generate Token Using Java

Prerequisites:
- Java 6 or 7
- Maven 2 or 3

On this page:

- **Build**
- **Usage**
  - Security parameters
  - Allow blank/missing referrer
  - Generate token
  - Decrypt token
2.4.17.2.2.1 Build
To build a generator:
1. Go to /cdn-wowza-token-tool/java/ location.
2. Run the following:
   
mvn clean install

Upon success of the build, you will find the jar (token-auth-generator.jar) file at the 'target' folder.

2.4.17.2.2.2 Usage

decrypt

\[\text{java -jar token-auth-generator-1.2.jar decrypt (<primary_key> | <backup_key>) "<security_parameters>"}\]

Security parameters

2.4.17.2.2.2.1 Security parameters

expire
- Number of seconds since Unix time (Epoch time)
- UTC based
- Must not be earlier than current time

ref_allow
- Referrer domain (e.g. google.com) or path (e.g. google.com/video)
- Allowed multiple referrers separated by comma (,) without space(s)
- Wildcard (*) allowed only at the beginning of a referrer, e.g. *.DOMAIN
- Do not append space at the start & end of a referrer
- Domain must fulfill RFC 3490
- Path must fulfill RFC 2396
- Should not include port (e.g. google.com:3000/video)
- Should not include protocol (e.g. http)

ref_deny
- Same rules as in ref_allow

If both ref_allow & ref_deny are specified, ref_allow will be taking precedence over ref_deny

2.4.17.2.2.2.2 Allow blank/missing referrer

Both "ref_allow" & "ref_deny" could be configured to allow/deny blank or missing referrer during TokenAuth validation. The following configuration allow blank or missing referrer:

ref_allow=allow.com
ref_allow=allow.com,MISSING
ref_deny=deny.com

The following configuration deny blank or missing referrer:

ref_allow=allow.com
ref_deny=deny.com
ref_deny=deny.com,MISSING
Normally ref_allow & ref_deny are not to be used together, but if this happened ref_allow will take precedence over ref_deny.

2.4.17.2.2.3 Generate token
To generate token, run the following:

```
java -jar token-auth-generator-1.2.jar encrypt samplekey
"expire=1598832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com"
```

Sample Output:
```
token=110ea31ac69c09a2db0b2d74238843631cdab498ff7e6e75cbed99cc4d05426a679a
57015d4e48438c97b921652daec62de3829f8ff437e27449cfdfc2f1e5d9fc47f14e91a51e
a7codecode
```

After generating a token, append the result to the playback URL.

2.4.17.2.2.4 Decrypt token
To decrypt token, run the following:

```
java -jar token-auth-generator-1.2.jar decrypt samplekey
110ea31ac69c09a2db0b2d74238843631cdab498ff7e6e75cbed99cc4d05426a679a57
015d4e48438c97b921652daec62de3829f8ff437e27449cfdfc2f1e5d9fc47f14e91a5
1ea7
```

Sample Output:
```
security
parameters=expire=1598832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com
```

2.4.18 CDN Edge Server IP Ranges
OnApp CDN provides a list of IP ranges available for edge servers to which you are subscribed on CDN Marketplace. You can use IP ranges to filter incoming traffic on your origin website by means of firewall rules. Whitelisting edge server IPs enables a firewall to accept incoming traffic from IP addresses within the range and block malicious traffic. In this document, you can find information on where to locate edge server IP ranges at your Control Panel.

You need to have the See own CDN resources permission enabled to access the IP Ranges menu of a CDN resource.
2.4.18.1 View IP Ranges
To view the list of IP ranges allocated to edge servers:
1. Go to the CDN > Resources menu at your Control Panel.
2. Click the IP Ranges tab.
3. On the page that appears, you will see the list of available IP ranges. You can copy the IP range and set up firewall rules on your origin website to accept incoming traffic from IP addresses within this range.

- Since the range of IP addresses may be changed by the edge server provider, it is recommended to use API to retrieve the latest list of IP ranges for your firewall rules. You can also run a cron to schedule jobs for checking changes in the IP range. For more information on API, refer to the Get Edge Server IP Ranges document.
- At Control Panel, the list of IP ranges is automatically updated to include the relevant IPs.

2.4.19 Verify CNAME Status
CNAME (Alias)—alias domain records to your domain.
To check the status of CNAME in Unix systems, use the Linux command:

```
dig +trace cdn.yourdomain.com
```

Here is the example below:
```
```

To check the status of CNAME in Windows systems, press the Windows logo key and type cmd in the search box. In a Command Prompt window, type the following command:

```
nslookup cdn.yourdomain.com
```

Here is the example below:
2.4.20 Dynamic Speed Limiting

The Dynamic Speed Limiting feature allows customers to restrict the maximum speed based on the specified parameter in a URL.

The maximum speed allowed to stream the following video is up to 1000 B/s:

http://cdn.example.com/testvideo.mp4?speed=1000

Please note that the Dynamic Speed Limiting feature is not available in the UI. To enable it, contact OnApp Support.

This feature also works with the existing URL Signing feature that requires the speed parameter to match with the hash key.
If a visitor changes the speed value manually on their own, it will not be possible to view the following video:

http://cdn.example.com/testvideo.mp4?speed=1000&secure=Sas2Xasd

To generate a hash key, refer to the Python script as in the example below:
import hashlib
import string
import base64
import time

# Fill in speed parameter and secret key
secret_key = ''
speed = '1000'
url = '/index.html'

# Expiry time
expires_y = 2013
expires_m = 9
expires_d = 29
expires_H = 0
expires_M = 0
expires_S = 0

expires = str(int(time.mktime(time.strptime("%d-%m-%d %H:%M:%S", "%Y-%m-%d %H:%M:%S")))))

def get_hashkey(url, secret_key, expires='', speed=''):  
    keystr = "%s%s%s%s" % (expires, url, secret_key, speed)
    hash_key = hashlib.md5()
    hash_key.update(keystr)
    base64_hash_key = base64.encodestring(hash_key.digest()).rstrip('
').replace('+', '-').replace('/', '_')
    return base64_hash_key

# Print the results:
print "Details: '%s' '%s' '%s' '%s'
print "No Expires : %s" % (get_hashkey(url, secret_key))
print "With Expires : %s" % (get_hashkey(url, secret_key, expires))
print "No Expires & Speed : %s" % (get_hashkey(url, secret_key, speed=speed))
print "With Expires & Speed : %s" % (get_hashkey(url, secret_key, expires, speed=speed))

2.5 CDN Edge Groups

CDN edge groups are groups of edge servers – your own, and those you subscribe to from the CDN marketplace. They are usually grouped by location, so they represent a pool of servers for a given geographical area. Once you have created an edge group containing edge servers in specific locations, you can then assign the group (or groups) to a specific CDN resource. The CDN edge groups menu enables you to see available edge server locations and form them into CDN Edge groups.

You need to associate CDN Edge groups with buckets to make them available for users.
2.5.1 View CDN Edge Group Details

To see details of a CDN Edge Group:

1. Go to your Control Panel > CDN > Edge Groups menu.
2. Click the label of the edge group you want to see.
3. On the screen that appears you will see the list of assigned locations and available locations with the following information:
   - **ID** – the ID of a location
   - **City** – the city the edge server is in.
   - **Operator** – name of the edge server owner.
   - **Type** - HTTP or streaming
   - **Source** – either Marketplace (locations added from the CDN marketplace) or your Own Edge servers (servers added by you).
   - **Status** - whether edge server is active or not.
   - **Price** – price per GB transferred.

2.5.2 Create CDN Edge Group

There are two ways of creating a CDN edge group:

1. Using a CDN setup wizard
2. Creating the edge group under the Edge Groups menu at OnApp Cloud Control Panel

To create a new CDN Edge Group using the Edge Groups menu:

1. Go to your Control Panel > CDN > Edge Groups menu.
2. On the screen that appears, you will see existing groups with the number of assigned locations and associated buckets.
3. Click the Create Edge Group button.
4. On the screen that appears, give your new group a label and click the Create Edge Group button.
5. You will be redirected to the screen where you can assign locations to the group. Do note that the location has an indication of whether it supports HTTP, Streaming, or both.

For details on the CDN setup wizard, refer to the CDN wizard section.

2.5.3 Edit CDN Edge Group

To edit the name of a CDN Edge Group:

1. Go to your Control Panel > CDN > Edge Groups menu.
2. On the screen that appears you will see the list of existing groups.
3. Click the Actions button next to appropriate CDN Edge Group, then click Edit.

2.5.4 Delete CDN Edge Group

To delete a CDN Edge Group:

1. Go to your Control Panel > CDN > Edge Groups menu.
2. On the screen that appears you will see the list of existing groups.
3. To delete the group, click the **Actions** icon, then click **Delete**.

Be careful when deleting an edge group which is associated with [CDN resources](#).

### 2.5.5 Assign/Remove CDN Edge Group Locations

1. Go to your Control Panel > **CDN** > **Edge Groups** menu.
2. Click the label of the **CDN Edge Group** you want to configure.
3. On the screen that appears you may assign or remove locations by clicking the **Actions** button next to the required location.

### 2.6 CDN Upload and Video Embed Instructions

Here is the list of instructions for uploading files and embedding video to CDN resources. Follow the step-by-step instructions below to upload files or embed video to the required CDN resource type:

- **Http Push CDN Resources**
- **VOD Pull CDN Resource**
- **VOD Push CDN Resource**
- **Live Streaming CDN Resource**

#### 2.6.1 HTTP Push CDN Resource

To upload files to the HTTP Push CDN resource:

1. Connect to the FTP origin using an FTP client. For example, a browser plug-in like FireFTP or FTP software like FileZilla.
2. Please wait up to 10 minutes until the FTP server configures with the HTTP resource.
3. Specify the following FTP details:
   - Hostname: 6789.origin.customercdn.com
   - Username: 6789
   - Password: The FTP password set at CDN resource creation.

   This is an instruction template. Replace “6789” with the resource id, and “customercdn.com” with the operator's domain.

4. Upload your files.
2.6.2 VOD Pull CDN Resource

To upload files to the HTTP Push CDN resource, enter the following script into your web page:

```html
<html>
<head>
  <script src="http://video.worldcdn-beta.net/player.js" type="text/javascript"></script>
</head>
<body>
  <div id="my-video-player"/>
  <script type="text/javascript">
    CDNPlayer("my-video-player", 1234, "1234/mystream", {width:640, height:360});
  </script>
</body>
</html>
```

- This is an instruction template. Replace “1234” with the resource id, “customercdn.com” with the operator’s domain, and “mystream.mp4” with the filename or stream name.
- resource_id/ must prefix the path with <resource_id>, it is ONLY applicable to VOD PULL playback.
- This example provides default values for width and height. You can change them to your own values.

Our easy video embed script automatically detects the browser type (Desktop or Mobile device) and loads the appropriate player. Currently, this is either Flow Player or the browser’s native HTML5 player. The streaming protocol is also set appropriately.

2.6.3 VOD Push CDN Resource

To upload files to the VOD Push CDN resource:

1. Connect to the FTP origin using an FTP client. For example, a browser plug-in like FireFTP, or FTP software like FileZilla.
2. Please allow up to 10 minutes for the FTP server to be configured with the VOD resource.
3. Specify the FTP details:
   - Hostname: 6789.origin.customercdn.com
   - Username: 6789
   - Password: the password set at creation
4. Upload your files.

5. Enter the following script into your web page:

```html
<html>
<head>
  <script src="http://video.worldcdn-beta.net/player.js" type="text/javascript"></script>
</head>
<body>
  <div id="my-video-player"/>
  <script type="text/javascript">
    CDNPlayer("my-video-player", 1234, "1234/mystream", {width:640, height:360});
  </script>
</body>
</html>
```

- This is an instruction template. Replace “1234” with the resource id, “customercdn.com” with the operator’s domain, and “mystream.mp4” with the filename or stream name.
- This example provides default values for width and height. You can change them to your own values.

Our easy video embed script automatically detects the browser type (Desktop or Mobile device) and loads the appropriate player. Currently, this is either Flow Player or the browser’s native HTML5 player. The streaming protocol is also set appropriately.

2.6.4 Live Streaming CDN Resource

1. Before you start, make sure your publishing point settings meet the following requirements (to be able to retrieve with the Silverlight Player):

   - h.264 Baseline 3
   - AAC or MP3-stereo-44100Hz audio
- 2 seconds key frame frequency
- lower bitrate

OnApp CDN does not autotranscode a stream into multiple bitrate or adaptive bitrate. It is recommended to publish multiple streams with a different bitrate from the publishing point and allow end users to choose the stream with the bitrate they need.

2. Install and configure the Adobe Live media encoder:
   a. Install Adobe Live Encoder.
   b. Once the Adobe Live Encoder is installed, run the application and move on to the next step.
   c. Complete the form:
      - FMS URL: rtmp://1234.publishstream.customercdn.com/P1234
      - Backup URL: rtmp://backup.1234.publishstream.customercdn.com/P1234
      - Stream: your stream name

      This is an instruction template. Replace “1234” with the resource id, and “customercdn.com” with the operator’s domain.

   d. Press Connect.
   e. In the password pop up window, enter “P1234” as the username and the resource secret key for the password.
   f. Press Start to start publishing the live stream.
Enter the following script into your web page to embed video to the Live Streaming CDN resource:

```html
<html>
<head>
<script src="http://video.worldcdn-beta.net/player.js" type="text/javascript"></script>
</head>
<body>
<div id="my-video-player"/>
<script type="text/javascript">
  CDNPlayer("my-video-player", 1234, "mystream", {width:640, height:360} );
</script>
</body>
</html>
```

This example provides default values for width and height. You can change them to your own values.

Our easy video embed script automatically detects the browser type (Desktop or Mobile device) and loads the appropriate player. Currently, this is either Flow Player or the browser's native HTML5 player. The streaming protocol is also set appropriately.

To stream a video using RTMP, enable secure Wowza and fill in the secure token after the creation of a live streaming resource. Example of the Flow Player embed code:
<h1>FLOW SMIL</h1>

Click here to see what should be changed


- token: escape('apppp&&') - change apppp&& to the secure token that has been set up.

To publish a stream via FFmpeg, use the following commands:

- `ffmpeg -re -i dummy.mp4 -f flv rtmp://username:password@fms_URL:80/P"CDN_Reference"/mystream`

**Example:**

- `ffmpeg -re -i dummy.mp4 -f flv rtmp://P404130149:mtXsvYDwQQ@404130149.publishstream.worldcdn.net:80/P404130149/streamname`

If an external publishing point is used, the customer should have the RTMP URL as in the example below:

- `rtmp//xxxxxx.xxxx.com/live/streamname1`
Click here to see the URL requirements

- URL must start with an rtmp, e.g., rtmp://xxxxxx.xxxx.com/live.
- URL must include the name of a stream specified in the ffmpeg command, which is provided in the command line in the CentOS system, during the streaming to the external publishing point.

This URL (from the external publishing point) should be playable. Then try to restream it using the CDN network.

Create a live streaming resource and insert rtmp://xxxxxx.xxxx.com/live/ in the External publishing location box.

As a result, your CDN URL should be as in the example below:

http://video.yourdomina.net/414746811/_definst_/streamname1.m3u8

3. Manual Instructions

We support a variety of methods to get the CDN URL to use in your player.

SMIL

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.smil

The SMIL playlist provides an RTMP URL and should be used with Flash-based players only. Longtail Player and Flow Player are compatible with SMIL redirection.

Apple HTTP Live Streaming

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.m3u8

This returns a 302 redirect to a Apple HLS manifest and should be used with Apple HLS-compatible players only.

Adobe HTTP Dynamic Streaming

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.f4m

This returns an Adobe HDS manifest and should be used with Adobe HDS-compatible players only.

Microsoft Smooth Streaming (Silverlight)

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.ism

This returns a 302 redirect to Smooth Streaming manifest and should be used with Smooth Streaming-compatible players only.

Javascript JSONP
http://video.cdn.qaonapp.net/726128906/_definst_/mystream.jsonp?callback=MyCallBack
This returns a JSONP document, embeddable using `<script>`

An example of a callback with a successful result:

```javascript
MyCallBack({
  "rtmp": "rtmp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream",
  "rtmpe": "rtmpe://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream",
  "apple": "http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/playlist.m3u8",
  "adobe": "http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/manifest.f4m",
  "rtsp": "rtsp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream",
  "silverlight": "http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/Manifest"
});
```

An example with an error:

```javascript
MyCallBack({
  "error": "File not found"
});
```

Javascript JSON

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.json
This returns a JSON document. Cross-origin resource sharing is enabled to allow XMLHttpRequest from any domains.

An example of a callback with a successful result:

```javascript
{
  "rtmp": "rtmp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream",
  "rtmpe": "rtmpe://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream",
  "apple": "http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/playlist.m3u8",
  "adobe": "http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/manifest.f4m",
  "rtsp": "rtsp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream",
  "silverlight": "http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/Manifest"
}
```

An example of a callback with a JSON document with an error thrown:

```javascript
{
  "error": "File not found"
}
```
2.7 CDN SSL Certificates

OnApp customers can import their own SSL certificates with the Subject Name Indication (SNI) extension.

SNI lets the client specify the hostname it is trying to reach at the start of the handshaking process. SNI is supported by most modern browsers, and provides an efficient way to deliver content over HTTPS using your own domain and SSL certificate. Custom SNI SSL relies on the SNI extension of the Transport Layer Security protocol, which allows multiple domains to serve SSL traffic over the same IP address by including the hostname viewers are trying to connect to.

Previously, OnApp applied SAN SSL certificate from a certificate authority to which additional certified domains can be added. This allowed you to host several domains on one IP by sharing the same certificate, and add all domains to this IP. However, the number of domains per SAN certificate is limited. Moreover, the certificate's size increases as more domains are added. This causes additional bandwidth to be used for the SSL handshake.

Currently, OnApp applies the CloudSSL+SNI solution. Users can import custom SNI SSL certificates into the system or request SSL to be enabled for their CDN resource. One SSL certificate can be associated with several CDN resources, but a resource can only be linked to one SSL certificate. Removal of a CDN resource does not affect the status of the SSL certificate associated with the CDN resource. However, some of the older browsers do not support SNI. In this case, users, who prefer browsers that do not support SNI, can purchase an SSL certificate and the SAN solution will be applied. On questions about the SSL certificate purchase, please contact OnApp support.

For the list of browsers that do not support SNI, kindly refer to the [Server Name Indication](#) article.

OnApp currently supports the following types of certificates:

- **domain-validated (DV) certificate** ([example.com](#))
  - single certificate
  - wildcard certificate (*.[example.com](#))
  - SAN certificate (any domains)
- **organization validation (OV) certificates**
  - single certificate
  - wildcard certificate (*.[example.com](#))
  - SAN certificate (any domains)
- **extended validation (EV) certificates**
  - single certificate
  - wildcard certificate (*.[example.com](#))
  - SAN certificate (any domains)
- **high-assurance certificates**

- This feature is available for HTTP Pull and HTTP Push resources only.
2.7.1 View Custom SNI SSL Certificate

To view the list of available SSL certificates:

1. Go to your Control Panel > CDN > SSL Certificates menu.
2. On the page that appears, all available custom SNI SSL certificates with their details are listed:
   - ID - the ID of the custom SNI SSL certificate
   - Name - the name of the certificate. Click the name to view the properties of the certificates and associated CDN resources
   - Actions - click the Actions button to edit or delete the certificate

2.7.2 Create Custom SNI SSL Certificate

To import an SSL certificate:

1. Go to your Control Panel > CDN > SSL Certificates menu.
2. On the page that appears, all available custom SNI SSL certificates are listed. Click the Import SSL Certificate button.

To add custom SNI SSL certificates, the user needs to have the CDN SSL Certificates permissions enabled.

3. On the following page, fill in the required information:
   - Name - specify a name for the certificate. This parameter is optional
• **Ssl certificate key** - fill in the certificate key, it must be in PEM-format

• **Private key** - fill in the SSL key provided by your SSL provider

Make sure that the *Ssl certificate key* and *Private key* parameters are filled out in the same way they are generated - with the line breaks.

4. Click the **Create SSL Certificate** button to import the certificate.

After you add a custom SNI SSL certificate to the cloud, you can associate it with a CDN resource.

To do this:

1. Go to your Control Panel > **CDN > Resources** menu.

2. On the page that appears, click the **Actions** button next to the CDN resource which you want to associate with the custom SNI SSL certificate, and select **Edit**.

3. On the following page, you can associate the custom SNI SSL certificate with the CDN resource (see the **Create HTTP CDN Resource** section for details).

4. Click **Save**.

When a custom SNI SSL certificate is associated with a CDN resource, the certificate applies only to the edge servers subscribed to that resource.

**2.7.3 Edit Custom SNI SSL Certificate**

You can edit your custom SNI SSL certificates, by following this procedure:

1. Go to your Control Panel > **CDN > Resources** menu.

2. On the page that appears, click the **Actions** button next to the CDN resource which you want to associate with the custom SNI SSL certificate, and select **Edit**. Alternatively, click the name of the certificate and click the **Edit** button on the page that loads.

3. On the following page, edit the details of the certificate:

   o **Name** - specify a name for the certificate
   
   o **Ssl certificate key** - fill in the certificate key, it must be in PEM-format
   
   o **Private key** - fill in the SSL key provided by your SSL provider. Private key will not be displayed for security reasons

Make sure that the *Ssl certificate key* and *Private key* parameters are filled in the same way they are generated - with the line breaks.

4. Click **Save**.

**2.7.4 Delete Custom SNI SSL Certificate**

To delete a custom SNI SSL Certificate:

1. Go to your Control Panel > **CDN > SSL Certificates** menu.
2. On the page that appears, click the **Actions** button next to the required certificate and select **Delete**.

Please note that the removal of an SSL certificate is not allowed if it is associated with a CDN resource.

### 2.8 CDN Accelerator

You can now access this page at [Accelerators](#).

#### 2.8.1 View Accelerators

You can now access this page at [View Accelerators](#).

#### 2.8.2 Create Accelerator

You can now access this page at [Create Accelerator](#).

#### 2.8.3 Manage Accelerators

You can now access this page at [Manage Accelerators](#).

#### 2.8.4 Accelerator Disks

You can now access this page at [Accelerator Disks](#).

#### 2.8.5 Accelerator Networks

You can now access this page at [Accelerator Networks](#).
2.8.6 Accelerator IP Addresses
You can now access this page at Accelerator IP Addresses.

2.8.7 Accelerator Statistics
You can now access this page at Accelerator Statistics.

2.8.8 Accelerator Console Access
You can now access this page at Accelerator Console Access.

2.8.9 RabbitMQ Configuration for Accelerator
Below you can find instructions on how to configure RabbitMQ for CDN Accelerator.
Compute Resources and Control Panel must use the same rabbitmq-server. For instructions on how to install RabbitMQ server, refer to the RabbitMQ Server Installation document.

- Starting from 6.2 Edge 2, Accelerator is renamed into Edge Accelerator.
- If you have the latest OnApp update installed, there is no need to upgrade and reboot compute resources. In this case:
  - configure users for RabbitMQ server (where RabbitMQ server is installed),
  - configure /home/mq/onapp/messaging/credentials.yml file (for Cloudboot compute resources this file should be configured after every reboot)
  - run the service onapp-messaging start command.

2.8.9.1 Configuration for Accelerator
Perform the following steps for your Cloudboot compute resources if you plan to deploy Accelerator. These steps are to be performed on each of the compute resources.

1. Run the following command on the CP server:
   - For all compute resources:
     ```
rake hypervisor:messaging:configure
     ```
   - For certain compute resources only:
To perform the configuration for a number of compute resources, separate their IPs with a space.

2. The command above should be run after every reboot. However, you can avoid the necessity to run the command repeatedly after every reboot by coping the following information (using your parameters) from /home/mq/onapp/messaging/credentials.yml to the custom config:

```bash
rake hypervisor:messaging:configure['11.0.50.111 11.0.50.112']
```

To make the configuration for the Accelerator manually, perform the following steps:

1. Copy file:

   ```bash
cp /home/mq/onapp/messaging/credentials_{example,}.yml
   ```

2. Open vi `/home/mq/onapp/messaging/credentials.yml` and check the following details:

   ```yaml
   ---
   host: 10.0.50.4 # RABBITMQ SERVER IP/FQDN
   port: 5672     # RABBITMQ CONNECTION PORT (default: 5672)
   vhost: '/'
   user: accelerator-example # RABBITMQ USER NAME
   password: 'e{y31?s8l' # RABBITMQ ACCESS PASSWORD
   queue: 'hv-10.0.50.102' # hv-[IP Address of Compute Resource]
   exchange:
     name: 'acceleration'
     type: 'direct'
     durable: True
   > /home/mq/onapp/messaging/credentials.yml
   chown -R mq:mq /home/mq
   service onapp-messaging restart
   ```

3. Change owner:

   ```bash
   chown -R mq:mq /home/mq
   ```

4. Run the following:

   ```bash
   service onapp-messaging start
   ```
Note that steps 1-4 of the above instruction should be done after every reboot of CloudBoot compute resource. You can run the following commands (using your parameters) to the custom config instead:

```bash
cp /home/mq/onapp/messaging/credentials{,_example}.yml
echo "---
host: 10.0.50.4 # RABBITMQ SERVER IP/FQDN
port: 5672 # RABBITMQ CONNECTION PORT (default: 5672)
vhost: '/'
user: accelerator-example # RABBITMQ USER NAME
password: 'e{y31?s8l' # RABBITMQ ACCESS PASSWORD
queue: 'hv-10.0.50.102' # hv-[IP Address of Compute Resource]
exchange:
  name: 'acceleration'
  type: 'direct'
  durable: True" > /home/mq/onapp/messaging/credentials.yml
chown -R mq:mq /home/mq
service onapp-messaging restart
```

2.9 CDN Reporting

CDN reporting functionality allows you to conduct the in-depth analysis of your own CDN resources by viewing different reports. At the moment the following reports are available:

- CDN Top Files Report
- CDN Top Referrers Report
- CDN Overview Report
- CDN Cache Statistics Report
- CDN Status Codes Report
- CDN Stream Bandwidth Report
- CDN Visitors Report
- CDN Admin Report

You can apply filters for every report (by time period, by CDN resource). Also export to csv format is available.

- Ensure that the CDN reports permissions are on before managing CDN reports. For more information refer to the Permissions section.
- Ensure that you have Aflexi ID before managing CDN reporting statistics.
- Cloud Administrator cannot view the CDN Reports on CDN Resource Usage of one's end users.
2.9.1 CDN Overview Report

To view the Overview report:

1. Go to your Control Panel > Metrics > CDN > Overview tab. You can filter the statistics by frequency and by date - select frequency (one minute, one hour or one day) and the time period from the drop-down menu and click the Apply button.

2. You will get the statistics divided into several sections:

   Line chart
   The line chart shows the cached and uncached bandwidth statistics (in MB) for the selected period.

   Top 5 CDN resources
   You can view top five CDN resources together with their details:
   - **Bandwidth** - the amount of transmitted bandwidth for the selected period
   - **Cache Hit** - the amount of successful file requests for the selected period
   - **Miss** - the amount of failed file requests for the selected period

   Top 5 HTTP errors
   This section shows top five CDN resources with the biggest amount of http errors (4xx&5xx)

   Top 5 CDN locations
   You can view top five CDN locations with the biggest amount of bandwidth.

   Visitor Statistic
   In this section you can find a diagram, which shows visitor statistics by region.

   To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

2.9.2 CDN Cache Statistics Report

To view the Cache Statistics report:

1. Go to your Control Panel > Metrics > Cache Statistics menu.

2. You can filter the statistics by:
   - **Type** - select the statistics type (GB, Hit/Miss, Speed) from the drop-down list
   - **CDN resource** - choose the CDN resource for which you want to view the statistics
   - **From** - select period start date
To - select period end date

3. Click the Apply button. You will get the statistics chart and table with locations

Depending on the selected statistics type, the chart and the table will show the following:

- If bandwidth type (GB) is selected
  The chart shows the cached and uncached bandwidth statistics (in Mb) for the selected period.
  Below you can find the list of corresponding CDN locations with their number of requests and amount of bandwidth.

- If Hit/Miss type is selected
  The chart shows the amount of hit and miss requests for the selected period.
  Below you can find the list of corresponding CDN locations with their total number of requests and amount of hit and miss requests.

- If speed type is selected
  The chart shows the cached and uncached bandwidth speed statistics (in Mbit/s) for the selected period.
  Below you can find the list of corresponding CDN locations with their speed amount.

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

To export the statistics in csv format, click the Export to CSV button.

The \$upstream_http_x_cache status can be either MISS, BYPASS, EXPIRED, UPDATING, or HIT.

Click here to see the meaning of the abovementioned statuses

- MISS - the content is not found in cache. It gets the content from the origin, cache it, and returns the content.
- BYPASS - no caching. It gets the content straight from the origin.
- EXPIRED - cache time limit reach. No content change in the origin.
- UPDATING - cache time limit reach. Cache gets new content change in the origin.
- HIT - the content is found in cache. It returns the content from the cache.

2.9.3 CDN Top Files Report

To view the Top files report:

1. Go to your Control Panel > Metrics > CDN menu.
2. Click the Top Files section. You will get the list of top 50 files (by default for the last week) with the following details:
   - Resource - the name of the CDN resource
• **File URL** - the URL of the resource file
• **Request** - the total amount of file requests for the selected period
• **Hit** - the amount of successful file requests for the selected period
• **Miss** - the amount of failed file requests for the selected period
• **Bandwidth** - the amount of transmitted bandwidth for the selected period
• **Actions** - if available, you can click the **Purge** button to remove cache content. This action is available only for accelerated CDN resources.

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

You can filter the statistics by date or by CDN resource - select the time period or the CDN resource from the drop-down menu and click the **Apply** button.

To export the statistics in **csv** format, click the **Export to CSV** button.

### 2.9.4 CDN Top Referrers Report

This report is available only for resource owner.

To view the Top referrers report:

1. Go to your Control Panel > **Metrics** > **CDN** menu.
2. Click the **Top Referrers** section. You will get the list of top 50 referrers (by default for the last week) with the following details:

   - **Resource** - the name of the CDN resource
   - **Referrer** - the name of the referrer
   - **Hit** - the amount of references for the selected period

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

You can filter the statistics by date or by CDN resource - select the time period or the CDN resource from the drop-down menu and click the **Apply** button.

To export the statistics in **csv** format, click the **Export to CSV** button.

### 2.9.5 CDN Status Codes Report

To view the Status Codes report:

1. Go to your Control Panel > **Metrics** > **CDN** menu.
2. Click the **Status Codes** section. You can filter the statistics by:

   - **Frequency** - select the frequency type (one minute, one hour or one day) from the drop-down menu
   - **CDN resource** - choose the CDN resource for which you want to view the statistics
   - **Time period** - select period start and end date

3. Click the **Apply** button. You will get the statistics chart and two tables:

   - The chart shows the number of requests with different error codes for the selected period.
• You can view the list of error codes together with their amount of requests in Status Codes table.
• Http error report table shows the list of CDN resources together with their amount of error requests.

OnApp applies the global setting for HTTP error code caching as follows:
• proxy_cache_valid 404 1m
• proxy_cache_valid 200 301 1d

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.
To export the statistics in csv format, click the Export to CSV button.

If your CDN hostname returns the following status codes:
• 2XX (200,201,202, etc.) means that the HTTP request is successful.
• 3XX (301, 302, etc.) means that there is a redirection layer being set up on the origin or webserver. To avoid any redirection conflict when using CDN, you can use an IP origin instead of a hostname origin. For example, instead of using www.mywebsite.com as an origin, use the IP address of www.mywebsite.com as a CDN resource origin.
• 4XX (401,402, etc.) means that the CDN PoP (edge servers) has an issue.

Click here to see more details
Check the following:
  o The status of your CDN resource is Active. If it is Preparing, this status code is correct because the configuration of the CDN resource has not been populated to the CDN edge server.
  o If you set any country access in the Access Policy field when creating or editing for your CDN resource, this status code is correct provided that you are from the prohibited countries.
  o With the Content Compare Tool, check which edge server is throwing a 4XX error. If it happened only on a few CDN edge servers but not all, contact OnApp Support and remove these locations from our CDN edge group temporary to keep users from receiving a 4XX error.
• 5XX (501, 502, etc.) means that CDN edge servers have an issue with fetching content from the CDN resource origin.

Click here to see more details
Check the following:
  o The Origin HTTP is accessible.
OnApp Cloud 6.5 Edge 5 CDN Guide

- The firewall has not blocked the edge server.
- DNS for the origin hostname is working.

2.9.6 CDN Stream Bandwidth Report

Be aware that the bandwidth statistics report shows information on Stream type CDN resources only.

To view the bandwidth statistics report:

1. Go to your Control Panel > Metrics > Stream Bandwidth menu.

2. Select the type of the filter – either GB or MBit/s. In MBPS mode you can get statistics for the last 10 days only. The older statistics is removed. There are no limitations for GB mode.

3. Specify the period in the From and To fields.

4. Select a resource and/or location for which the statistics will be generated.

5. Click the Apply button.

The points in the graph are displayed according to a particular frequency, which depends on the time period specified with the From and To parameters:

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Frequency in seconds</th>
<th>Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 30 days</td>
<td>86400</td>
<td>1 day</td>
</tr>
<tr>
<td>31 - 93 days</td>
<td>604800</td>
<td>1 week</td>
</tr>
<tr>
<td>equal or more than 93 days</td>
<td>1209600</td>
<td>2 weeks</td>
</tr>
</tbody>
</table>

Statistics available in the frequency higher than selected will be accumulated to a single point of such frequency. For example, the statistics was requested for the period of 31-93 days, so the frequency of points in the graph is 7 days. If the statistics was generated few times during those 7 days (day1+day2+day\text{n}), it will be added up and displayed as a single point, with a time stamp marked as the first day of such 7 days.

The statistics are displayed in two graphs: Cached and Non Cached.
To zoom into a time period, click and drag in a chart. Click the Reset Zoom button to zoom out again.

### 2.9.7 CDN Concurrent Statistics

The CDN streaming statistics screen shows the concurrent viewers report for your CDN streaming sites.

To view the statistics/graphs for your CDN streaming sites:

1. Go to your Control Panel > Metrics > Concurrent Statistics menu.
2. Specify the period in the From and To fields.
3. Select the type of the filter – either by resources or by locations.
4. Click the Apply button.

To zoom into a time period, drag the chart by holding down the left mouse button and moving the mouse. Click the Reset Zoom button to zoom out again.

The concurrent number showed is the average number of a five-minute interval. It is recommended to scale down the graph by limiting the time zone to a smaller time frame to get an accurate result. If the selected time frame is too large, the graph aggregates the data.

### 2.9.8 CDN Visitors Report

To view the Visitors report:

1. Go to your Control Panel > Metrics > CDN menu.
2. Click the Visitors section. You can filter the statistics by:
   - CDN resource - choose CDN resource, for which you want to view the statistics
   - Time period - select period start and end date
3. Click the Apply button. You will get the Top 5 Countries chart (by default for the last week) with the visitors’ amount statistics. Also you will get the Visitor Countries table (sorted by the highest request) with the following details:
   - Visitor Country - the code of the visitor country
   - AVG Latency - the average latency for the selected period
   - AVG Transfer Rate - the average transfer rate for the selected period
   - Requests - the amount of successful file requests for the selected period
   - Bandwidth - the total amount of transmitted bandwidth for the selected period

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.
To export the statistics in csv format, click the **Export to CSV** button.

### 2.9.9 CDN Admin Report

Ensure that the *See Admin report* permission is on before managing the Admin report. For more information refer to the **Permissions** section.

To view the Admin report:
1. Go to your Control Panel > **Metrics** > **CDN** menu.
2. Click the **Admin** section. You can filter the statistics by:
   - *Frequency* - select the frequency type (one minute, one hour or one day) from the drop-down menu
   - *Time period* - select period start and end date
3. Click the **Apply** button. You will get the statistics divided into several sections:

   **Line Chart (Bandwidth statistics)**
   The line chart shows the total amount of cached and uncached bandwidth statistics (in Gb) for the selected period.

   **Top 5 CDN resources**
   You can view top five CDN resources together with their details:
   - *CDN Resource* - the name of CDN resource
   - *Past Hour* - the amount of bandwidth for the past hour
   - *This Hour* - the amount of bandwidth for this hour
   - *Bandwidth* - the amount of transmitted bandwidth for the selected period

   Click the table's label to view it separately.

   **Top 5 CDN locations**
   You can view top five CDN locations with the biggest amount of bandwidth. Click the table's label to view it separately.

   **Bandwidth, sold in Federation, is not included in this statistics table.**

   **Top 5 Http errors**
   This section shows top five CDN resources with the biggest amount of http errors (4xx&5xx). Click the table's label to view it separately.
To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

2.9.10 CDN Usage Statistics

The Metrics > CDN > CDN Usage page lists the summary of CDN Resources used by CDN with their details:

- **Owner** – the owner's user name. Click the owner's name for details.
- **Edge Group** – the edge group to which the CDN resource belongs to.
- **Location** – CDN edge server's location.
- **Data cached** – cached CDN traffic in a `number_to_human_size` format. (See the table below)
- **Data non cached** – non cached CDN traffic in a `number_to_human_size` format. (See the table below)

The table of formatting the bytes in `number` into a more understandable representation:

<table>
<thead>
<tr>
<th><code>number_to_human_size</code></th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>number_to_human_size(123)</code></td>
<td>123 Bytes</td>
</tr>
<tr>
<td><code>number_to_human_size(1234)</code></td>
<td>1.21 KB</td>
</tr>
<tr>
<td><code>number_to_human_size(12345)</code></td>
<td>12.1 KB</td>
</tr>
<tr>
<td><code>number_to_human_size(123456)</code></td>
<td>1.18 MB</td>
</tr>
<tr>
<td><code>number_to_human_size(1234567)</code></td>
<td>1.15 GB</td>
</tr>
<tr>
<td><code>number_to_human_size(1234567890)</code></td>
<td>1.12 TB</td>
</tr>
<tr>
<td><code>number_to_human_size(1234567890123)</code></td>
<td>1.15 GB</td>
</tr>
<tr>
<td><code>number_to_human_size(483989, :precision =&gt; 2)</code></td>
<td>470 KB</td>
</tr>
<tr>
<td><code>number_to_human_size(1234567, :precision =&gt; 2, :separator =&gt; ',')</code></td>
<td>1,2 MB</td>
</tr>
</tbody>
</table>

By default, statistics are generated for the last 72 hours. To specify another period, set the Start and End time and click the Apply button. Tick the Show in my Timezone box to show CDN usage statistics according to your profile's time zone settings.

Deleted CDN resources/locations will be marked as unavailable after the upgrade to newer version of the OnApp cloud.

2.10 Assign Zones to CDN Location Groups

To properly configure the Location groups in your cloud, assign the CDN locations and Compute resource, Data store, Network and Backup server zones to them.

To assign zones to an CDN location group:

1. Log in to your OnApp Control Panel.
2. Go to your Admin > Settings > Location Groups menu.
3. Click the Country or City of the Location Group in question.
4. The page that loads is organized into the list of CDN Locations and Compute resource/Data store/Network/Backup server zones. Click the "+" button next to a required zone.
5. In the window that pops up, choose a particular location or zone and click **Attach**. The CDN locations available for sign up are those configured in OnApp Dashboard, while zones are taken from OnApp CP.

6. Repeat the procedure for other zones/locations.

You can also assign a Location Group to a particular CDN Edge Server zone by following the [Create CDN Edge Server](#) guide.

For more information on **Location Groups**, refer to the linked section.

### 2.11 CDN Debugger

OnApp CDN Debugger provides you with the ability of initial investigation in case of any problem with your CDN performance. It contains the following tools:

#### 1. DNS Debugger - explains the redirection logic of the CDN.

DNS-Debugger allows you to check which CDN PoP would the visitor request get redirected to. Go to: [https://debug.onappcdn.com](https://debug.onappcdn.com) and select the "DNS-Debugger". You will be required to provide the following details:

- a. Resource ID, or Resource name - mandatory field
- b. IP - optional field

You can obtain the Resource ID using the following ways:

- a. In the CNAME of your CDN resource, the Resource ID is the numeric sequence that precedes your CDN resource CNAME.
- b. In CP, the Resource ID is the value of "CDN reference" which can be found at the "CDN Resource Details" page.

IP (an optional field) is used to simulate the visitor's request on a CDN resource. If no IP is provided, the current connection IP will be automatically selected. You may also insert a different IP in order to simulate the DNS-Debugger result from a different region/country.

The result will display:

- a. The place you (acting as a visitor) have come from, with the information about the country, coordinate, and closest airport.
- b. The selected CDN locations you were get redirected to based on your location in part 1. All possible locations will be listed, and the selected location's edge IP(s) will be highlighted in green color.
- c. The health status of each Edge location.

OnApp CDN redirection decision is made based on the ping result from the CDN PoPs to the DNS IP of the visitor.

If a server has multiple IPs, the IP shown in the DNS debugger may be different from the server’s "main IP" shown in other debugging tools, because the customers are distributed across all POP's IPs.

#### 2. Ping Test - provides the ping result from an IP/hostname to OnApp CDN POPs.
3. **Traceroute Test** - provides the traceroute result from an IP/hostname to OnApp CDN PoPs.
   The Traceroute test gives you the traceroute result of your CDN resource to a specific OnApp CDN PoP.
   
   Go to [https://debug.onappcdn.com](https://debug.onappcdn.com) and select the Traceroute Test, insert IP/hostname, select the CDN PoP that you would like to test, and click "Run Test".

4. **Content Compare Tool** - allows you to compare the file on origin with the edge servers.
   Content Compare allows you to check if the HTTP request of a CDN resource is successful on all the possible CDN PoPs where it can be accessed. Go to [http://debug.onappcdn.com/contentcompare](http://debug.onappcdn.com/contentcompare), insert your CDN hostname together with a file.
   
   The result will display if the HTTP request is successful on all CDN PoPs (showing code 200). If it is not successful, for example, showing error code (400) on specific PoPs, please, stop using that particular PoP to avoid users getting an error page and inform CDN support.

5. **Munin Graphs** - provides an overview of the edge servers' status.
   Munin, the monitoring tool surveys all your computers and presents all the information in graphs via the web interface. Using Munin you can easily monitor the performance of your computers, networks, SANs, applications, weather measurements, and so on. It determines "what's different today" when a performance problem crops up and helps to check the capacity-wise on any resources.
   
   OnApp CDN provides Munin graphs for all edge servers to help operators monitor their edge servers’ status. The Munin graphs are accessible through the [CDN Debug](https://debug.onappcdn.com) page. These are the examples of good and bad graphs:

<table>
<thead>
<tr>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU Usage</strong>&lt;br&gt;Y-Axis represents the CPU usage percentage.&lt;br&gt;Avoid having high iowait and high steal. Also, spot the unusual trend like the system CPU usage and user CPU usage growing rapidly.</td>
<td><img src="image" alt="Example Good CPU Usage" /></td>
</tr>
<tr>
<td>This shows CPU has low iowait and low steal.</td>
<td>This shows CPU has high iowait and some steal.</td>
</tr>
<tr>
<td><strong>Possible Actions</strong>: Upgrade storage to a high-performance disk (e.g., SSD) and/or upgrade CPU.</td>
<td></td>
</tr>
<tr>
<td><strong>Disk usage in percent</strong>&lt;br&gt;Y-Axis represents disk usage percentage.&lt;br&gt;Ignore /dev, /run, /run/lock, /run/shm and /boot partitions.</td>
<td><img src="image" alt="Example Good Disk Usage" /></td>
</tr>
</tbody>
</table>
Cache partitions are OK to fill up until 90% of the disk space.

<table>
<thead>
<tr>
<th>Possible Actions:</th>
<th>There are a lot of free space on /, /mnt/nginx/bay-* and /var/cache/nginx-hls partitions.</th>
<th>There is not enough free space on / partition. Generally, all partitions should not grow beyond 90% of disk space.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utilization per device</strong></td>
<td><strong>Possible Actions:</strong> Require investigation.</td>
<td><strong>Possible Actions:</strong> Upgrade storage to a high-performance disk (eg. SSD) and/or add more disks.</td>
</tr>
<tr>
<td><strong>Y-Axis represents the disk percentage busy.</strong></td>
<td>This shows minimal disk utilization.</td>
<td>This shows high disk utilization, reaching 90% disk utilization.</td>
</tr>
<tr>
<td><strong>The disk utilization should be below 80% on average.</strong></td>
<td><strong>Possible Actions:</strong> Upgrade storage to a high-performance disk (eg. SSD) and/or add more disks.</td>
<td><strong>Possible Actions:</strong> Require further investigation</td>
</tr>
<tr>
<td><strong>Disk IOs per device</strong></td>
<td><strong>Possible Actions:</strong> Require further investigation</td>
<td><strong>Possible Actions:</strong> Upgrade storage to a high-performance disk (eg. SSD) and/or add more disks.</td>
</tr>
<tr>
<td><strong>Y-Axis represents IO operations per second.</strong></td>
<td>This shows sustained reading and writing operations in the edge which indicates the edge is working well.</td>
<td>This shows reading and writing are almost zero (middle line) which indicates the edge is idle and not processing any request. The spike showed when the edge is resumed to ACTIVE.</td>
</tr>
<tr>
<td><strong>The positive value shows writing operations. The negative value shows reading operations. Zero value indicates no IO operations on the server.</strong></td>
<td><strong>Possible Actions:</strong> Upgrade storage to a high-performance disk (eg. SSD) and/or add more disks.</td>
<td><strong>Possible Actions:</strong> Require further investigation</td>
</tr>
<tr>
<td><strong>Disk latency per device</strong></td>
<td>This shows the disk latency is on average below 10 milliseconds. This is reasonable disk latency for an edge with SSD storage. For edge with HDD storage, disk latency below 50 milliseconds is acceptable.</td>
<td>This shows the disk latency is high (more than 10 milliseconds).</td>
</tr>
<tr>
<td><strong>Y-Axis represents Average IO Wait in seconds. The shorter disk latency means better performance.</strong></td>
<td><strong>Possible Actions:</strong> Upgrade storage to a high-performance disk (eg. SSD) and/or add more disks.</td>
<td><strong>Possible Actions:</strong> Require further investigation</td>
</tr>
<tr>
<td><strong>Load average</strong></td>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Y-Axis represents CPU Load. Higher numbers may represent a problem or an overloaded machine.</td>
<td>The load is below 6 on average. Generally, the load of 6 to 8 is normal for average specification edge servers with 4 to 8 CPU cores.</td>
<td>The load is high which is above 10 on average. However, for a server with high specification, it is OK to have a higher CPU Load. <strong>Possible Actions:</strong> Upgrade CPU to be able to handle large loads.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Memory usage</strong></th>
<th><img src="image" alt="Graph" /></th>
<th><img src="image" alt="Graph" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-Axis represents memory usage in bytes.</td>
<td>This shows steady memory usage.</td>
<td>Memory usage grows quickly beyond its capacity and has small unused space. <strong>Possible Actions:</strong> Require further investigation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Nginx</strong></th>
<th><img src="image" alt="Graph" /></th>
<th><img src="image" alt="Graph" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-Axis represents Connections or Requests. Higher the value indicates more connections and requests it handles.</td>
<td>This shows that the edge handles a lot of user requests.</td>
<td>This shows that the edge handles a little number of user requests. <strong>Possible Actions:</strong> Ensure the edge has a good specification so that DNS will redirect more requests to the edge.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Ping &amp; Packet Loss</strong></th>
<th><img src="image" alt="Graph" /></th>
<th><img src="image" alt="Graph" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-Axis represents both Packet Loss and Ping. A positive value represents ping time in millisecond. The negative value represents the percentage of packet loss.</td>
<td>This shows a consistent connection between the edge and OnApp CDN Monitoring servers with no negative value.</td>
<td>This shows packet loss (negative value) and unreachable from OnApp CDN Monitoring servers.</td>
</tr>
</tbody>
</table>
### Throughput per device

Y-Axis represents bytes of reading and writing per second.

- A positive value indicates data writing. A negative value indicates the data reading. Zero value indicates no data operations on the storage devices.

<table>
<thead>
<tr>
<th>Possible Actions: Ensure the Internet connection to the server is stable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>This shows that the storage devices are actively handling user requests.</td>
</tr>
<tr>
<td>This shows that the storage devices are near zero which indicates the edge might be idle from handling user requests.</td>
</tr>
</tbody>
</table>

### Uptime

Y-Axis represents uptime in days.

<table>
<thead>
<tr>
<th>Possible Actions: Ensure the server has lesser downtime and network connection is good.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The server uptime is increasing over time indicates no downtime on the server.</td>
</tr>
<tr>
<td>The sudden drop in server uptime indicates the server experienced downtime recently.</td>
</tr>
</tbody>
</table>

External links:
http://munin-monitoring.org/
https://en.wikipedia.org/wiki/Munin_(software)

6. **Edge Monitoring** - allows you to track your edge servers' health status.

Go to [https://debug.onappcdn.com/edgemon](https://debug.onappcdn.com/edgemon). This monitoring panel allows you to check the health of the CDN edge servers. It also shows the graph of the edge server components, like CPU, RAM, and Disk usage.

- `check_fs_error` - This is to check the filesystem of the edge server by writing to it
- `check_heartbeat` - This is to check if the edge responds to the monitoring
- `check_http` - This is to check the edge server port 80 for respond
- `check_load` - This is to check the CPU load
- `check_munin` - This is to check if OnApp Munin monitoring can retrieve data from this edge
- `check_network` - This is to check the outgoing speed of the server
- `check_ping` - This is to check if the edge server is pingable from inside and outside and packet loss
- `check_puppet_compile` - This is to check if the edge server has the latest OnApp CDN configuration

You can also subscribe for alerts on your edge server status at [https://debug.onappcdn.com/edgealert](https://debug.onappcdn.com/edgealert)
7. Ping IP Submission - allows you to submit DNS IPs for the Ping engine to collect the ping data and to receive the results in the redirection algorithm.

8. Edge Servers IP Ranges - allows you to view all the current IP ranges for the subscribed marketplace and edge servers of your firewall system(s).

9. Hardware Inventory - provides you with a list of your servers and POPs.

OnApp CDN Debugger requires the same login credentials as the dashboard user account does.
If you experience issues using the CDN debugger, it means that the CDN Debug access option may not be enabled in your dashboard user account. Edit the user account permission in the dashboard or contact OnApp Support.

### 2.12 CDN Restrictions Sets

Restrictions sets can limit the following CDN resources:

Some resources can be limited both by bucket and user group. If two restrictions are selected for one parameter, the reseller's access to this resource will be defined by both these limitations at the same time.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Restriction Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDN resources</td>
<td>by user group</td>
<td>The reseller can manage only those CDN resources, which are used by customers, who are members of the user group to which this reseller is assigned.</td>
</tr>
<tr>
<td>CDN resources</td>
<td>by bucket</td>
<td>The reseller can manage CDN resources assigned to Edge groups which are added to reseller bucket.</td>
</tr>
<tr>
<td>CDN usage statistics</td>
<td>by user group</td>
<td>The reseller can monitor only CDN usage statistics of those users, who are members of the user group to which this reseller is assigned.</td>
</tr>
</tbody>
</table>

### 2.13 CDN Permissions

The list below includes the permissions related to CDN:

- CDN Locations
- CDN Reports
- CDN Resources
- CDN SSL Certificates
- CDN Usage Statistics
- DNS Zones
- DNS Zone's Records
• **Edge groups**
• **Edge servers**
• **HTTP Caching Rules**
• **Storage Servers**
• **Default Permissions for Admin Role**
• **Default Permissions for User Role**

### 2.13.1 CDN Locations

OnApp administrators can control users' ability to manage CDN locations through the Control Panel's Roles menu. You can set the following CDN locations permissions for user roles:

- **Any action on CDN Locations** - the user can take any action on CDN locations
- **See all CDN Locations** - the user can see details of all CDN locations
- **Update any CDN Locations** - the user can edit any CDN locations

### 2.13.2 CDN Reports

OnApp administrators can control users' ability to manage CDN reports through the Control Panel's Roles menu. You can set the following CDN reports permissions for user roles:

- **Any action on Reports** - the user can take any action on CDN reports
- **See Admin Report** - the user can see Admin Report
- **See Stream Bandwidth Report** - the user can see Stream Bandwidth Report
- **See Overview Report** - the user can see Overview Report
- **See Cache Statistics Report** - the user can view Cache Statistics Report
- **See Top Files Report** - the user can view Top Files Report
- **See Top Referrers Report** - the user can view Top Referrers Report
- **See Status Codes Report** - the user can view Status Codes Report
- **See Visitors Report** - the user can see Visitors Report

For details, refer to the [CDN Reporting](#) section.

### 2.13.3 CDN Resources

OnApp administrators can control users' ability to manage CDN resources through the Control Panel's Roles menu. You can set the following CDN resources permissions for user roles:

- **Any action on CDN resources** - the user can take any action on CDN resources
- **Create a new CDN resource** - the user can create a new CDN resource
- **Destroy any CDN resource** - the user can delete a CDN resource
- **Destroy own CDN resources** - the user can only delete their own CDN resources
- **See all CDN resources** - the user can see all CDN resources
- **See own CDN resources** - the user can only see their own CDN resources
- **Update any CDN resource** - the user can edit any CDN resource
• Update own CDN resources - the user can only edit their own CDN resources
For details, refer to the CDN Resources section.

2.13.4 CDN SSL Certificates
• Any action on CDN SSL Certificates - the user can take any action on CDN SSL certificates
• Create a new CDN SSL Certificate - the user can create a new CDN SSL certificate
• Destroy any CDN SSL Certificate - the user can delete any CDN SSL certificate
• Destroy own CDN SSL Certificate - the user can only delete their own CDN SSL certificates
• See all CDN SSL Certificates - the user can see all CDN SSL certificates. If this permission is disabled, the user cannot create SSL certificates.
• See own CDN SSL Certificates - the user can only see their own CDN SSL certificates
• Update any CDN SSL Certificate - the user can edit any CDN SSL certificate
• Update own CDN SSL Certificate - the user can only edit their own CDN SSL certificates
For details, refer to the CDN SSL Certificates section.

2.13.5 CDN Usage Statistics
OnApp administrators can control users' ability to manage CDN usage statistics through the Control Panel's Roles menu. You can set the following CDN usage statistics permissions for user roles:
• See details of CDN usage statistics - the user can see CDN usage statistics details
• User can see CDN usage statistics - the user can see CDN usage statistics
• User can see own CDN usage statistics - the user can only see own CDN usage statistics
For details, refer to the CDN Usage Statistics section.

2.13.6 DNS Zones
OnApp administrators can control users' ability to manage DNS zones through the Control Panel's Roles menu. You can set the following DNS zone permissions for user roles:
• Any action on DNS zone - the user can take any action on DNS zone
• Create a new DNS zone - the user can create a new DNS zone
• Destroy any DNS zone - the user can delete a DNS zone
• Destroy own DNS zone - the user can only delete their own DNS zones
• See all DNS zones - the user can see all DNS zones
• See own DNS zones - the user can only see their own DNS zones
• DNS Setup - the user can set up DNS
For details, refer to the DNS section.

2.13.7 DNS Zone's Records
OnApp administrators can control users' ability to manage DNS zone's records through the Control Panel's Roles menu. You can set the following DNS zone's record permissions for user roles:
• Any action on DNS Zone's Records - the user can take any action on DNS zone's records
• Create a new DNS Zone’s Records - the user can create a new DNS zone's record
• Delete any DNS Zone’s Record - the user can delete any DNS zone's record
• Delete own DNS Zone’s Record - the user can delete own DNS zone's records
• See all DNS Zone’s Records - the user can see all DNS zone's records
• See own DNS Zone’s Records - the user can see own DNS zone's records
• Update any DNS Zone’s Record - the user can update any DNS zone's records
• Update own DNS Zone’s Records - the user can update own DNS zone's records

For details, refer to the DNS section.

2.13.8 Edge groups

OnApp administrators can control users’ ability to manage edge groups through the Control Panel's Roles menu. You can set the following edge groups permissions for user roles:

• Any action on edge groups - the user can take any action on edge groups
• Create a new edge group - the user can create a new edge group
• Destroy any edge group - the user can delete any edge group
• See all edge groups - the user can see all edge groups
• See list of available Edge Group Locations - allows users to see the list of all available locations which can be assigned to the edge group
• Read price for all Edge Group Locations - with this permission users will see the price for using the location. Without this permission, users won’t see the price column at all neither for assigned location nor for available
• Update any edge group - the user can edit any edge group

For details, refer to the Create CDN Edge Group section.

2.13.9 Edge servers

OnApp administrators can control users’ ability to manage edge servers through the Control Panel's Roles menu. You can set the following edge server permissions for user roles:

• Any action on Edge Server - the user can take any actions on edge servers
• Change an owner of any Edge Server - the user can change the owner of any edge server
• Create a new Edge Server - the user can create a new edge server
• Destroy any Edge Server - the user can destroy any edge server
• Destroy own Edge Servers - the user can destroy own edge servers
• Migrate any Edge Server - the user can migrate any edge server
• Migrate own Edge Servers - the user can migrate own edge servers
• Any power action on Edge Servers - the user can take any power-related action on edge server
• Any power action on own Edge Servers - the user can take any power-related action on own edge servers
• See all Edge Servers - the user can see all edge servers
• See own Edge Servers - the user can see own edge servers
• Read VIP status - the user can read VIP status of edge servers
- **Rebuild Network on any Edge Server** - the user can rebuild network on any edge server
- **Rebuild Network on own Edge Servers** - the user can only rebuild network on own edge servers
- **Set VIP status** - the user can set/delete VIP status for edge servers
- **Change Suspended status for Edge Server** - the user can change Suspended status for any edge server
- **Unlock any Edge Server** - the user can unlock any edge server
- **Update any Edge Server** - the user can update any edge server
- **Update own Edge Servers** - the user can update own edge servers

For details, refer to the [CDN Edge Servers](#) section.

### 2.13.10 HTTP Caching Rules

OnApp Administrators can control user's ability to manage HTTP Caching rules for CDN. You can set the following permission:

- **See http caching rules for cdn resources** - the user can set HTTP caching rules for the resources.

For details, refer to the [HTTP Caching Rules](#) section.

### 2.13.11 Storage Servers

- **Any action on Storage Server** - the user can take any actions on storage servers
- **Change an owner of any Storage Server** - the user can change the owner of any storage server
- **Create a new Storage Server** - the user can create a new storage server
- **Destroy any Storage Server** - the user can delete any storage server
- **Destroy own Storage Servers** - the user can delete own storage servers
- **Migrate any Storage Server** - the user can migrate any storage server
- **Migrate own Storage Servers** - the user can migrate own storage servers
- **Any power action on Storage Servers** - the user can migrate own storage servers
- **Any power action on own Storage Servers** - the user can take any power-related action on own storage servers
- **See all Storage Servers** - the user can see all storage servers
- **See own Storage Servers** - the user can see own storage servers
- **Read VIP status** - the user can read VIP status of storage servers
- **Rebuild Network on any Storage Server** - the user can rebuild network on any storage server
- **Rebuild Network on own Storage Servers** - the user can only rebuild network on own storage servers
- **Set VIP status** - the user can set/delete VIP status for storage servers
- **Change Suspended status for Storage Server** - the user can change Suspended status for any storage server
- **Unlock any Storage Server** - the user can unlock any storage server
- **Update any Storage Server** - the user can update any storage server
• **Update own Storage Servers** - the user can update own storage servers

  For details, refer to the [CDN Storage Servers](#) section.

### 2.13.12 Default Permissions for Admin Role

The list below includes the set of default CDN permissions for the Admin role.

#### CDN Locations
- **Any action on CDN Locations** - the user can take any action on CDN locations

#### CDN Reports
- **Any action on CDN reports** - the user can take any action on CDN reports

#### CDN Resources
- **Any action on CDN resources** - the user can take any action on CDN resources

#### CDN SSL Certificates
- **Any action on CDN SSL Certificates** - the user can take any action on CDN SSL certificates

#### CDN Usage Statistics
- **See details of CDN usage statistics** - the user can see CDN usage statistics details

#### DNS Zones
- **Any action on DNS zone** - the user can take any action on DNS zone

#### DNS Zone’s Records
- **Any action on DNS Zone’s Records** - the user can take any action on DNS zone’s records

### 2.13.13 Default Permissions for User Role

The list below includes the set of default CDN permissions for the User role.

#### CDN reports
- **Any action on Reports** - the user can take any action on CDN reports
- **See Overview Report** - the user can see Overview Report
- **See Cache Statistics Report** - the user can view Cache Statistics Report
- **See Top Files Report** - the user can view Top Files Report
- **See Top Referrers Report** - the user can view Top Referrers Report
- **See Status Codes Report** - the user can view Status Codes Report
- **See Visitors Report** - the user can see Visitors Report
- **See Stream Bandwidth Report** - the user can see Stream Bandwidth Report

#### CDN Resources
- **Create a new CDN resource** - the user can create a new CDN resource
- **Destroy own CDN resources** - the user can only delete their own CDN resources
- **See own CDN resources** - the user can only see their own CDN resources
- **Update own CDN resources** - the user can only edit their own CDN resources

#### CDN SSL Certificates
- **Create a new CDN SSL Certificate** - the user can create new CDN SSL certificates
• **Destroy own CDN SSL Certificate** - the user can only delete their own CDN SSL certificates
• **See own CDN SSL Certificates** - the user can only see their own CDN SSL certificates
• **Update own CDN SSL Certificates** - the user can only edit their own CDN SSL certificates

**CDN Usage Statistics**
• **User can see own CDN usage statistics** - the user can only see own CDN usage statistics

**DNS Zones**
• **Create a new DNS zone** - the user can create a new DNS zone
• **Destroy own DNS zone** - the user can only delete their own DNS zones
• **See own DNS zones** - the user can only see their own DNS zones

**DNS Zone’s Records**
• **Create a new DNS Zone’s Records** - the user can create a new DNS zone's record
• **Delete own DNS Zone’s Records** - the user can delete own DNS zone's records
• **See own DNS Zone’s Records** - the user can see own DNS zone's records
• **Update own DNS Zone’s Records** - the user can update own DNS zone's records

### 2.14 Manage DNS

The full version of OnApp Cloud (with CDN enabled) now gives you access to our free Anycast DNS service. Instead of managing your own DNS servers you can use our fully redundant global DNS, hosted at multiple datacenters around the world, and manage it through your OnApp Control Panel.

You can use our DNS service with domains registered anywhere on a third party domain registrar. The Control Panel lets you set up hostnames, manage DNS records, aliases, Mail Exchange, TXT and SRV records.

Using DNS has two main steps: setting up DNS hostnames and managing DNS zones.

#### 2.14.1 Set Up DNS

DNS setup allows you to create a DNS hostname. After you create a DNS hostname, you get access to creating and managing DNS zones. To set up a new DNS, make sure that:

- CDN is enabled
- You have dns_zone or dns_zone.setup permission to access this page.

You can create only one DNS domain. Once created, a DNS domain can’t be deleted, only updated.

If an administrator uses the same license for two different Control Panels, they can use the same DNS domain for CP1 and CP2. To do this, administrator should set the same DNS domain settings for both Control Panels.

To add a DNS domain:

1. Go to your Control Panel > **Admin** > **Settings** menu.
2. Click the **DNS Setup** button.
3. On the screen that appears, type your fully qualified domain name. Mind that you won't be able to use a domain name that is already registered with OnApp DNS.

The domains of the following kind are forbidden:

- google
- microsoft
- domain.com
- onapp.com
- facebook.com
- gmail.com
- googlemail.com
- yahoo

4. Click the Save button.

After the DNS domain is added, the DNS service will be available to users. If you have added a domain, but still face some issues or the "Unable to get DNS Zone Setup: CDN service is temporarily unavailable" error occurs, contact support.

Make sure your domain name registrar has designated your domain to the appropriate glue records displayed in the infobox at Control Panel > Admin > Settings > DNS Setup.

2.14.1.1 Edit DNS Domains
To edit your DNS domain:

1. Go to your Control Panel > Admin > Settings menu, and click the DNS Setup icon.
2. Type a new fully qualified domain name to replace your existing domain
3. Click the Save button to save changes.

If DNS domain is updated, all NS records for all DNS zones under this user will be updated.

2.14.2 Manage DNS Zones
OnApp DNS Zone feature allows you to manage your and your clients' domain DNS. Each time DNS zone, record or setup settings are refreshed, the DNS configuration is immediately updated on the DNS vendor server.

2.14.2.1 Create DNS Zones
To add a new DNS zone:

1. Go to your Control Panel > Cloud > DNS menu.
2. Click the Create DNS Zone button.
3. Fill in your domain name. At the domain registrar, point your domain to the following name servers:
   - ns1.yourdomain.com
   - ns2.yourdomain.com
   - ns3.yourdomain.com
   - ns4.yourdomain.com
Where yourdomain.com is your fully qualified domain name, which you have specified at DNS setup.

Starting with OnApp 5.3, you can create rDNS zone. Reverse DNS resolution (rDNS) is the determination of a domain name associated with an IP address via querying DNS. Most of mail servers make rDNS lookup before accepting messages originating from mail server as one of the anti spam email technique.

To create rDNS zone, at the domain registrar, point your domain to the following name servers:

0.0.0.0.0.0.0.8.b.d.0.1.0.0.2 ip6.arpa
0-25.228.169.69.in-addr.arpa
64-30.228.169.69.in-addr.arpa
228.169.69.in-addr.arpa

4. Move the Auto Populate With Existing DNS record slider to the right if you want to automatically import your existing DNS settings, or skip this step to start from scratch. Note that this option may not import all existing settings, so you should check your new record for any missing entries.

5. Click the Submit button.

6. On the page that appears, click the Add icons next to the DNS records you want to add. You can add and manage the following DNS records:

   - **SOA** (Start of Authority) – change the start of authority time to live value (TTL). To change a SOA TTL, click the TTL cell next to the SOA record and type a new value.

   - **NS** (Name Server) – change the TTL of the existing name servers or add new name servers. To change a name server's TTL, click the TTL cell next to the name server record and type a new value.

   - It is not allowed to add an NS record for the main DNS zone ('@' in a "Name" field). An NS record is used to point a subdomain to a set of your nameservers (subdomain delegation).

   - **A** (Host) — point your domain name to a static IP address. To create a new A record, enter the following parameters into the cells:

     - **Name** – enter a host name or use the "@" sign to represent your current host.
     - **IP** – enter the IP address to which the user would be sent for this host name.
     - **TTL** – set the TTL value for this record.

     A record example: ftp 192.168.0.1 86400
     Where: ftp is the host; 643763287490 - IP , 86400 is TTL value.
     So your ftp.yourdomain.com will resolve to 2a00:1450:400b:c00::68 IP address and the TTL value = 86400 seconds.

   - **AAAA** (Host)

     To create a new AAAA record, fill in the following cells:

     - **Name** – enter a host name or use the "@" sign to represent your current host.
     - **IP** – enter the IP address to which the user would be sent for this host name.
     - **TTL** – set the TTL value for this record.

     AAAA record example:
     ftp 2a00:1450:400b:c00::68 86400
Where: ftp is the host, 2a00:1450:400b:c00 – IPv6 address, 86400 is the TTL value. So your ftp.yourdomain.com will resolve to 2a00:1450:400b:c00 IPv6 address.

- **CNAME (Alias)** – alias domain records to your domain.

It is possible to use underscore character in the CNAME records.

To add a CNAME record, fill in the following cells:
- **Name** – enter an alias you want to assign to your domain.
- **Hostname** – enter the host name or use the "@" sign to represent your current host.
- **TTL** – set the TTL value.

CNAME record example:
```
www example.com 86400
```
Where: www is an alias, example.com is a valid domain name, 86400 is TTL value.

To add the aliases, make sure an A record is added to this domain.

- **PTR (Pointer)** - create PTR records for rDNS zone.
To add a PTR record, fill in the following cells:
- **Name** – enter the IP:
  - for IPv4
    One of the IP from the range, such as 1 or 2
  - for IPv6
    One of the IP from range, such as 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2
- **Hostname** – enter the host name or use the "@" sign to represent your current host.
- **TTL** – set the TTL value.

MX record example:
```
10 mail example.com 86400
```
Where: 10 is priority, mail is the hostname, example.com is a domain, 86400 is TTL.

- **TXT** – add additional information about the DNS zone.
- **Name** – enter the valid host name
- **Txt** – any free text you want within a TXT record. Maximum 1300 characters.
**TTL** – TTL value.

**TXT** record example:
@ v=spf1 a mx ptr ip4:192.168.1.1 ~all 86400
Where: @ is the host name, v=spf1 a mx ptr ip4:192.168.1.1 ~all is a Txt value, 86400 is TTL.

- **SRV** (Service) – specify services that you have on your domain.
  To add a SRV record, enter the following cells parameters:
  - **Name** – enter the domain name.
  - **Priority** – set the host priority. Lower value means more preferred.
  - **Weight** – the approximate weight for relative records with the same priority.
  - **Port** – the port on which the service can be found.
  - **Hostname** – type the host for which this record is valid.
  - **TTL** – set the time to live value.

  SRV record example:
  xmpp._tcp 0 1 5222 jabber.example.com 86400
  Where: xmpp._tcp is a host, 0 is priority, 1 is weight, 5222 is port, jabber.example.com is a name, 86400 is TTL.

### 2.14.2.1.1 Wildcards
It is possible to use wildcards in all DNS records. The table below explains the rules of wildcard use.

<table>
<thead>
<tr>
<th>DNS record type</th>
<th>Allowed</th>
<th>Disallowed</th>
</tr>
</thead>
</table>
| A               | *.example.com  
|                 | **.example.com  
|                 | *abc.example.com  
| ** Note: multiple "** will be changed to single "". E.g. ****.example.com will be changed to *.example.com  
| ** Can only be prefixed for domain.  | abc.*example.com  
|                 | *.*.example.com  
|                 | sub.*.example.com  | abc.*.*.example.com  |
|AAAA             | Same to A record  | Same to A record  |
|MX               | Same to A record  | Same to A record  |
|CNAME            | Same to A record  | Same to A record  |
| ** Note: CNAME wildcard record can not coexist with A record.  |
|TXT              | *.example.com  
|                 | *.*.example.com  
|                 | sub.*.example.com  
|                 | sub.*.*.example.com  
|                 | *sub.*.example.com  |
| ** Note: Wildcards are valid in any position, as long as the domain remains DNS zone’s subdomain.  |
It is not possible to use wildcards for NS, SOA and SRV records because of PowerDNS limitations.

2.14.2.1.2 Underscore characters
It is possible to use wildcards in all DNS records. The table below explains the rules of underscore use.

<table>
<thead>
<tr>
<th>DNS Record type</th>
<th>Allowed</th>
<th>Disallowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>_abc.example.com</td>
<td>abc_.example.com</td>
</tr>
<tr>
<td></td>
<td>__abc.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>abc_abc</em>.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Multiple '_' e.g. ‘_____’ will not be changed to a single underscore, unless stated.</td>
<td></td>
</tr>
<tr>
<td>AAAA</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td>MX</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td>CNAME</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td></td>
<td>Note: CNAME record with underscore can not coexist with A record.</td>
<td></td>
</tr>
<tr>
<td>NS</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td>SRV</td>
<td>_xmpp._tcp.example.com</td>
<td>All except examples in the Allowed column.</td>
</tr>
<tr>
<td></td>
<td>__xmpp.__tcp.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>_xmpp.<em>tcp</em>.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>_xmpp._tcp__abc.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Multiple '_' e.g. ‘_____’ will be changed to a single underscore character.</td>
<td></td>
</tr>
<tr>
<td>TXT</td>
<td>All except examples in the Disallowed column. Note: underscores are valid in any position, as long as the domain remains DNS zone's subdomain.</td>
<td><em>example.com abc</em>.example.com example.com example_.com</td>
</tr>
</tbody>
</table>

2.14.2.2 Edit DNS Zones
When you edit a DNS Zone, you manage the records assigned to this zone. The changes are instantly updated on DNS.

To edit a DNS zone:

1. Go to your Control Panel > Cloud > DNS menu. On the screen that appears, the list of DNS zones will be displayed.
2. Click the domain name of the zone you want to edit.
3. On the screen that appears, edit the DNS records as required:
   - SOA (Start of Authority) – change the start of authority TTL.
2.14.2.2 Delete DNS Zones
To delete a domain zone:

1. Go to your Control Panel > Cloud > DNS menu.
2. Click the Actions button next to the domain zone you want to delete, then click Delete.
3. Click OK to confirm the deletion.

2.14.2.3 User DNS Zone
User DNS zones tab allows you to manage your clients' DNS zones. Use the Actions button next to the required user DNS zone to delete it. Click the domain name to edit a user's DNS zone. The changes will be instantly updated on our DNS.

2.14.2.4 Set End-User Access to DNS Services
To set end-users' access to DNS service:

1. Go to your Control Panel > Admin > Roles menu.
2. Click the Actions button next to the required user, then click Edit.
3. On the screen that follows, choose the DNS Zone group in the Groups field.
4. The following list of DNS permissions will appear:

   o Any action on DNS zone
   o Create a new DNS zone
   o Destroy any DNS zone
   o Destroy own DNS zone
   o See all DNS zones
   o See own DNS zones
   o Create a new DNS record
   o Destroy any DNS record
- Any action on DNS record
- See all DNS records
- See own DNS records
- Update any DNS records
- Update own DNS records
- DNS Setup

5. Tick the required boxes.
6. Click the **Save** button.

2.14.2.6 View User DNS Zones
To view existing clients’ DNS zones:
1. Go to your Control Panel > **Cloud** > **DNS** menu.
2. Click the **User DNS Zones** tab. On the screen that appears, you’ll see a list of all clients’ DNS zones.

To search for a particular zone, type the text you want to find in the search box and click the **Search** button.

2.14.2.7 Edit User DNS Zones
To edit existing clients’ DNS zones:
1. Go to your Control Panel > **Cloud** > **DNS** menu.
2. Click the **User DNS Zones** tab. On the screen that appears, you’ll see a list of all clients’ DNS zones.
3. Click the domain name of the zone you wish to edit. On the screen that appears, edit the zone’s details and click the **Save** button.

2.14.2.8 Delete User DNS Zones
To delete existing clients’ DNS zones:
1. Go to your Control Panel > **Cloud** > **DNS** menu.
2. Click the **User DNS Zones** tab. On the screen that appears, you’ll see a list of all clients’ DNS zones. Click the **Actions** button next to the DNS zone you want to delete, then click **Delete**. You’ll be asked to confirm deletion.
3 User Guide

OnApp CDN is a software product that works with OnApp Cloud v2.3 and later. If you enable CDN for your end users, they can get access to a network of global edge servers (yours, and/or those provided by other cloud hosts) and use this network to deliver web content more quickly and reliably to visitors all over the world. Your CDN is managed alongside your cloud via the OnApp Control Panel. For more information, see OnApp CDN for Service Providers.

OnApp provides the following types:

- HTTP Push
- HTTP Pull
- Live Streaming
- On-Demand Streaming

**HTTP Push**

The HTTP Push method is similar to secondary server: the user uploads content to CDN and links to it, so the content is physically stored at CDN storage servers.

**HTTP Pull**

With the HTTP Pull CDN, the website owner stores content on their server and rewrites URLs to include a subdomain. Then, when the specific content is requested, it is pulled to the CDN network from the host and delivered to the closest point to the consumer who requested that content (the content is cached on edge server).

**Streaming**

There are two supported streaming types in the OnApp CDN – live streaming and on-demand streaming. To utilize CDN streaming service, you have to deploy CDN streaming Edge Server.

**Live Streaming**

Live Streaming CDN allows delivering the content to end users in a live broadcast mode. When using a live streaming CDN service, end users receive media the same time like a traditional broadcasting.

**Video On-Demand Streaming**

Video On-Demand Streaming CDN allows delivering video on request and makes it repeatedly accessed. Consumers can control content and are able to fast forward or rewind it the same as live streaming.

To utilize CDN streaming service, you have to deploy CDN streaming Edge Server.

Setting up OnApp CDN includes:

- Enabling CDN for your cloud in the OnApp customer dashboard (contact OnApp Support if you don’t have dashboard access)
- Running CDN Setup Wizard in your OnApp Control Panel
- Setting up storage servers
- Setting up edge servers
- Adding CDN edge groups and assigning them to buckets
- Creating CDN resources (when you create a CDN resource, CDN is enabled automatically in the OnApp)
- Assigning users to a bucket and setting their permissions

**Storage servers** store web content to be distributed over the Content Delivery Network. The content is then cached by edge servers and delivered to consumers.

**Edge servers**, cache web content and deliver it to website visitors. They are deployed to compute resources and managed the same way as virtual servers. You can use edge servers to sell CDN bandwidth to your end users, and/or submit the edge server to the OnApp CDN marketplace and sell your bandwidth to other hosting providers. You can create as many edge servers as you need and place them to different compute resources in different geographical locations, and easily broaden your CDN by combining your own edge servers with other locations on the CDN marketplace. You can also build a CDN solely with marketplace resources. For details, refer to the **Edge servers** chapter.

**Edge groups** are groups of edge servers that can be your own and/or those you subscribe to from the CDN marketplace. They are usually grouped by location, so they represent a pool of servers for a given geographical area.

Edge groups are assigned **Buckets** to set the prices for the bandwidth that your end users consume. You can assign several groups to one bucket at a time, and establish different geographical zones with different pricing. The bandwidth pricing of the bucket is the price for CDN bandwidth sold to your end users.

**CDN resources** are specific servers with content that an end user wants to distribute via CDN. CDN resources are assigned to edge groups, which determine the list of servers taking part in distributing/caching of their data.

---

Starting with OnApp Cloud v3.0, CDN is enabled automatically after adding the first DNS record or CDN resource.

### 3.1 Setup Wizard

To launch the CDN setup wizard, users should have the following permissions enabled:

- **Update any Role**
- **See all Roles**
- **Create a new edge group**

The CDN setup cycle consists of three steps that involve configuring:

1. **Permissions**
2. **Edge Groups**
3. **Billing**

You can rerun the wizard after the initial configuration as many times as you need.

To start the CDN wizard:

1. Go to the **CDN > Edge Servers** menu in your Control Panel.
2. Click the **CDN Setup Wizard** button.
3. Complete the steps in the wizard as described below.
3.1.1 Step 1 Permissions

- **Client role** - select a user role to enable the CDN-related permissions listed in the *Enables Permissions* section. When you complete the wizard, these CDN permissions will be automatically applied to the selected role. You can also enable CDN-related permissions for additional user roles via the *Roles* menu.

- **Admin role** - select the Administrator role to enable the CDN-related permissions listed in the *Enables Permissions* section. For the Administrator role, any action on CDN resources is enabled. When you complete the wizard, these CDN permissions will be automatically applied to the Administrator role.

  - Do not assign the Administrator permissions to a client-shared role.
  - You may skip the **Step 1 Permissions** if you have already set CDN-related permissions for the user and administrator roles.
  - Users will not be able to purchase and manage CDN resources unless CDN permissions are enabled for their role.

Click the **Next** button to proceed to the next step.

3.1.2 Step 2 Edge Groups

- **Label** - enter a name for your edge group. For example, you can create the edge group called "North America" and add to it your North American PoPs.

- **Assigned Locations** - the list of locations assigned to the edge group.

- **Available Locations** - the list of available locations that you can assign to the edge group. To add a location, click the '+-' button next to the location you want to assign to the group.
Edge groups can include your own servers and servers from the CDN Marketplace. You can assign additional edge groups later in the Bucket.

Click the Next button to proceed to the next step.

### 3.1.3 Step 3 Billing

- **Bucket** - assign the CDN edge group to a bucket.
- **Price per GB** - set a price per GB for traffic your users push through all servers in this edge group.
- You can assign more edge groups to a bucket and set different prices for them, using the **Buckets** menu. Any user assigned to the destination bucket will be able to create a CDN service that includes the edge group’s locations.
- You cannot add two edge groups with the same location to one bucket.

When you are finished configuring billing for CDN, click the **Create Edge Group** button. On the page that loads, you will see a summary on the created edge group. You can click the **CDN Dashboard** button to access your Dashboard.

After CDN is set up, synchronization between CDN and OnApp is run every 20 minutes. If synchronization fails because of CDN Sync Runner issues, you will receive the notification. To solve this issue check CDN Sync Runner status via **Sysadmin**. The CDN section in Sysadmin enables you to check the status of CDN API and see the last time CDN statistics were gathered.

### 3.2 Edge Servers

Web content is cached in the network of edge servers on the CDN, distributed across different geographic locations. Currently there are two types of edge servers in OnApp: HTTP and Streaming.
Starting with OnApp 5.4 version, edge servers functionality is applicable for users with vCloud Director integration. For more information on VMware Cloud Director Edge Servers, refer to the linked guide.

HTTP edge servers support both Push and Pull population methods in 80/20 ratio (80% HTTP Pull and 20% HTTP Push). When the edge server is created, its storage limit for HTTP Pull and HTTP Push is automatically assigned by the system.

Streaming edge server type allows to send a stream to one of the publishing points, or pick up the stream externally and deliver it to the end users.

Take note that Operator has to deploy CDN streaming Edge Server to utilize CDN streaming service.

Streaming edge server support only streaming services. The following protocols are supported:

- HTTP
- RTMP/RTMPE/RTMPT/
- HDS
- RTSP/RTP
- iPhone
- SilverLight
- MPEG-TS

Streaming service includes the following advanced features:

- Hotlinking protection - protect your media from being hotlinked (linked to website without your permission)
- Geo Blocking - restrict access to your media so that it is accessible only for certain countries/regions
- RMTPE (secure Wowza) – streaming encryption.

You do not have to add the Wowza license key manually to enable streaming edge servers. A third party application - Wowza will be installed automatically when installing an edge server and you will simply be charged for it. Please, contact your account manager for details.

Content is delivered to end users from the server which is closest to the user or has the best availability. If you have CDN enabled for your cloud, you can use the control panel to set up your own edge servers and manage them in the same way you manage virtual servers. You can submit your edge server to the locations in the marketplace to sell bandwidth across it. To be able to sell CDN bandwidth through our online marketplace, you must first submit your edge server for assessment.

PLEASE NOTE: CDN edge servers do not support firewall and NAT (network address translation).
You can use smart compute resources for CDN edge server creation.

3.2.1 Create Edge Server

Starting with OnApp 5.4 version, edge servers functionality is applicable for users with vCloud Director integration. For more information on VMware Cloud Director Edge Servers, refer to the linked guide.

To add new CDN edge server:
1. Go to your Control Panel > CDN > Edge Servers menu.
2. On the screen that appears, click the Create Edge Server button or press the "+" button.
3. Fill in the edge server creation form step by step:

   On this page:
   - Step 1 of 4. Locations
   - Step 2 of 4. Properties
   - Step 3 of 4. Resources
   - Step 4. Confirmation

3.2.1.1 Step 1 of 4. Locations
The Cloud Locations step applies to those users who have compute zones assigned to location groups in their bucket.

If the user's bucket has several compute zones, some of which are assigned to location groups, whereas others are not - the cloud locations screen will not be available in the wizard. Also if there is only one location this step will be skipped. In this case the wizard will start with the Properties step.

Indicate your edge server's cloud location:
- **Country** - choose the country, where the cloud is located, from the drop-down menu.
- **City** - specify the city, where the cloud is located, from the drop-down menu.

Click Next to proceed to the following step of the wizard to specify the edge server properties.

3.2.1.2 Step 2 of 4. Properties
- Give your edge server a label. The label can consist of:
  - Lower- & upper-case letters [A-Za-z]
  - Digits [0-9]
  - Dash [- ]
  - Underscore [ _ ]
  - Space character [ ]
  - At sign [@]
  - Brackets [ (){}]
  - Slashes [/]
  - Caret [ ^ ]
  - Dollar sign [$]
• Select an edge server type: HTTP or streaming

A third party application - Wowza will be installed automatically when installing a streaming edge server and additional charges will apply. Please, contact your account manager for details.

• Move the Add to Marketplace slider to the right to submit this server to the OnApp CDN marketplace. If so, the minimum required disk size is 1 TB.

• Click Next.

Any servers you submit will be assessed before they are accepted into the marketplace. Criteria include geographic location, bandwidth and server specs.

If your cloud has sufficient resources, but the Next button is dimmed during server creation, the reason might be a browser issue. For more information, refer to our Knowledge base article.

3.2.1.3 Step 3 of 4. Resources

• Select the compute zone and the specific compute resource to build the edge server on.
• Set the resources needed for this edge server: RAM, CPU cores and CPU priority.
• Choose a data store zone and data store for this edge server's primary disk.
• Set the primary disk size. The disk size is calculated in the following way: 10 GB for OS, the rest of total disk space is estimated 80% per Pull population and 20% per Push population.

The disk size should not exceed 2 TB when a new disk is added. You can later resize the disk if you need it to be larger than 2 TB.

• Choose a network zone and set the port speed for this edge server, or make it unlimited. It is not possible to set port speed value for edge servers based on smart Compute resources.
• Choose the network from which the VS should get the IP address.

3.2.1.4 Step 4. Confirmation

1. 
   • Move the Build Edge Server Automatically slider to the right if you want the system to automatically build the edge server. Otherwise, you will have to build your server manually after it is created.

2. Click Create Edge Server.
3.2.2 View Edge Server Details

Edge servers are virtual compute resources that are much like other virtual servers in your cloud. You can perform the same basic actions on them as for VSs.

To view all edge servers in the cloud:
1. Go to your Control Panel > CDN > Edge Servers menu to see an overview of all edge servers in your cloud: their label, IP addresses, power status (with on/off buttons), allocated disk size, RAM and backups.
2. To reboot, start up or shut down a CDN edge server, click the Actions button next to the required edge server, then select the relevant action.
3. To narrow the list of edge servers by type, click the relevant tab at the top of the list.

To view a particular edge server’s details:
1. Go to your Control Panel > CDN > Edge Servers menu.
2. Click the label of the edge server required.
3. On the screen that appears, use the top navigation tabs to manage your edge server.

---

**Overview**
The Overview tab shows CDN edge server information and gives access to the most frequently-used management tools.

**Properties**
CDN edge server’s details page

**CPU Usage**
CDN edge server’s CPU usage statistics

**Billing Statistics**
CDN edge server’s billing statistics information

**Networking**
The Networking tab gives access to the edge server’s network interfaces and IP addresses.

**Network Interfaces**
CDN edge server’s network configuration

**IP Addresses**
CDN edge server’s IP addresses

**Storage**
The Storage tab lets you manage your edge server’s disks.

**Disks**
The integrated VNC console tab gives access to the users’ edge servers through the Control Panel UI. This option is available only for edge servers running on KVM compute resources.

**Console**

For the HTML5 console, click the Re-connect button if the connection is lost. The re-connection to the console runs as follows:...
4. To expand the Tools menu, click the Tools button on the edge server's screen. The list of available options depends on the edge server's status. For options description, refer to relevant sections of the Virtual Servers chapter.

3.2.3 Edit Edge Server

Editing a CDN edge server means adjusting the resources allocated to it, changing its label and basic properties.

To edit a CDN edge server:

1. Go to your Control Panel > CDN > Edge Server menu.
2. Click the label of an edge server.
3. On the next screen, click the Tools button, then click the Edit Edge Server link:
   - Change the edge server label.
   - Edit CPU core/priority and RAM values.

   Note that the server will be rebooted if you edit resources allocated.

   - Move the Add to Marketplace slider to the right to add the edge server to the marketplace. (This option is only available for edge servers that were not added to the marketplace earlier.)

4. Click Save Edge server.

3.2.4 Delete Edge Server

To delete a CDN edge server:

1. Go to your Control Panel > CDN > Edge Servers menu.
2. Click the label of an edge server.
3. When the page loads, click the Tools button, then click Delete Edge Server.
4. You will be asked for confirmation before the edge server is deleted.
3.2.5 Edge Server Network Interface Usage

Network Interface usage page displays bandwidth used in two charts: statistics for 24 hours and hourly statistics for the period up to three months.

To view the edge server's network interface usage statistics:
1. Go to your Control Panel > CDN > Edge Servers menu.
2. Click the label of the Edge Server required.
3. On the screen that appears, click Networking tab > Network Interfaces.
4. Click the Interface Usage icon next to the network interface needed.
5. On the screen that appears, set the start time and end time and click Apply.

3.3 Storage Servers

CDN storage servers are used for storing the content to be distributed over CDN. When the content is requested on the CDN, it is served by the edge server nearest to the customer’s geographical location.

There are two types of CDN storage servers in OnApp: HTTP and Streaming.

You can use smart compute resources for CDN storage server creation.

3.3.1 View Storage Server Details

To view the list of all CDN storage servers in the cloud:
1. Go to your Control Panel > CDN > Storage Servers menu to see an overview of all storage servers in your cloud: their label, IP addresses, power status (with on/off buttons), allocated disk size, RAM and backups.
2. To reboot, startup/shutdown a storage server, click the Actions button next to the required storage server, then select the relevant action.
3. To narrow the list of storage servers by type, click the relevant tab at the top of the list.

To view a particular edge server's details:
1. Go to your Control Panel > CDN > Storage Servers menu.
2. Click the label of the storage server required.
3. On the screen that appears, use the top navigation tabs to manage your storage server.

Storage servers are managed with the same tools you use to manage VSSs in your cloud. The table below links to the relevant sections of this guide's Virtual Servers chapter. The exception is that autoscaling, backups and firewall rules are not available for storage servers.

| Overview | The Overview tab shows CDN edge server information and gives access to the most frequently-used management tools. |
### Properties
- CDN edge server’s details page
- CDN edge server's CPU usage statistics
- CDN edge server's billing statistics information

### Networking
- The Networking tab gives access to the edge server's Network interfaces and IP addresses.
- CDN edge server's network configuration
- CDN edge server’s IP addresses

### Storage
- Disks
- The Storage tab lets you manage your edge server's disks.

4. To expand the Tools menu, click the Tools button on the storage server's screen. The list of available options depends on the edge server's status). For options description, refer to the Virtual Servers section.

#### 3.3.2 Create Storage Server

To create a new storage server:

1. Go to your Control Panel > CDN > Storage Servers menu.
2. Click the Create Storage Server button at the bottom of the screen.
3. Fill in the storage server creation form step by step:

**Step 1 of 4. Cloud Locations**

The Cloud Locations step applies to those users who have compute zones assigned to location groups in their bucket.

If the user's bucket has several compute zones, some of which are assigned to location groups, whereas others are not - the cloud locations screen will not be available in the wizard. In this case the wizard will start with the Properties step.

Indicate your application server's cloud location:
- **Country** - choose the country, where the cloud is located, from the drop-down menu.
- **City** - specify the city, where the cloud is located, from the drop-down menu.

**Step 2 of 4. Properties**

Specify the storage server details:
- Specify the server’s label in a human-recognizable format.
- Select the storage server type: HTTP or Streaming.

> A third party application - Wowza will be installed automatically when installing a streaming storage server and additional charges will apply. Please, contact your account manager for details.

- **Location** - choose the location group to assign this storage server to.
- Specify the compute resource and compute zone.
• Click Next.

Step 3 of 4. Resources
• Set the resources needed for this storage server: RAM, CPU cores and CPU priority. The minimum RAM capacity is 2 GB.
• Choose a data store zone for this storage server’s primary disk
• Set the primary disk size (Storage server HDD). The minimum required disk size is 20 GB.
• Choose a network zone from the drop-down box.
• Choose the network from which the VS should get the IP address.
• If the option is available, you can also assign an IP address for the VS from the drop-down menu. Indicate compute resource and network to have the list of available IPs. Tick the Show Only My IP Addresses checkbox to view only own IP addresses in the IP addresses drop-down box.
• Set the port speed in Mbps or tick it as unlimited.
• Click Next.

Step 4. Confirmation
• On the screen that appears, tick the Build Edge Server automatically box to build the storage server automatically, otherwise you will have to build your storage server manually after it is created.
• Click the Create Storage Server button to start the creation process.

3.3.3 Edit Storage Server
1. Go to your Control Panel > CDN > Storage Servers menu.
2. Click the label of a required storage server.
   o On the next screen, click the Tools button, then click the Edit Storage Server link under the Storage Server Options.
   o Change the storage server label.
   o Edit CPU core/priority and RAM values.
3. Click Save.

3.3.4 Delete Storage Server
To delete a storage server:
1. Go to your Control Panel > CDN > Storage Servers menu.
2. On the screen that appears, you’ll see the list of all storage servers in the cloud. Click the HTTP/Streaming tabs to view storage servers by type.
3. Click the Actions button next to the storage server you want to remove, then click Delete.
3.4 Resources

A CDN resource is a host (e.g. a specific web server), the content of which you are going to distribute over the network of edge servers.

There are three types of CDN resources in the OnApp Control Panel:

- **HTTP** CDN resource type supports both Push and Pull population.
- **VoD** CDN resource type (Pull and Push types) allows to use on-demand video streaming service - uploading video and streaming to the end users.
- **Live Streaming** CDN resource type allows to broadcast content using CDN.

Only servers added to the edge groups assigned to the resource will distribute/cache the host's content.

To activate the CDN Resources menu, at least one CDN Edge Group with at least one edge server or marketplace location must be available.

Apart from the CDN Resources permissions enabled, the following requirements must be met for the publisher to be able to create respective resources. If the requirements are not met, the publisher will not be able to create the particular resource type:

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>HTTP Pull</th>
<th>HTTP Push</th>
<th>VoD Pull</th>
<th>VoD Push</th>
<th>Live Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements based on user Edge group</td>
<td>User bucket must include an edge group with at least one location that supports HTTP.</td>
<td>User bucket must include an edge group with at least one location that supports HTTP, and an HTTP storage server.</td>
<td>User bucket must include an edge group with at least one location that supports VoD Push.</td>
<td>User bucket must include an edge group with at least one location that supports VoD Push, and a streaming storage server.</td>
<td>User bucket must include an edge group with at least one location that supports live streaming.</td>
</tr>
</tbody>
</table>

3.4.1 View Resources

Go to your Control Panel > CDN > Resources menu.
You will see the list of all CDN resources with the following information:

- **CDN Hostname** – hostname of the CDN resource.
- **Origin Sites** – path of the content that will be served from the CDN.
- **Type** - resource type: HTTP Push, HTTP Pull, VoD or Live streaming.
- **Cost** – cost of the resource.

To view HTTP, VoD or live streaming resources only, click the required tab.

You can edit/delete a resource using the relevant icons next to each resource in the list, and add a resource with the CDN Resource Wizard button.

3.4.2 View Resource Details

To view basic settings:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the required CDN Hostname.

3. On the page that appears, click Basic settings tab. The screen provides you with the following information:

**CDN Resource details**
- **Owner**
- **CDN hostname**
- **Resource type**
- **SSL on** - whether SSL is enabled for the resource or not
- **Let's Encrypt SSL** - whether Let's Encrypt SSL is enabled for the resource or not

To see more details on the Let's Encrypt SSL status, select the Click to view more option.
- **SNI SSL Certificate** - custom SNI SSL certificate associated with the resource
- **CDN reference** – the ID of the resource in database
- **Resource status** – shows the resource status. Click **Suspend** to terminate.

**Origins**
- The IP of the content that will be served from the CDN.

**DNS settings**
- A CNAME for the CDN Hostname which can then be used to view the contents. Use this for the origin settings.

**Edge Groups**
- Shows to which Edge groups the resource is assigned.

**Last 24 hours cost**
- Cost of the resource for the last 24 hours.

To view advanced details:
1. Go to your Control Panel > **CDN > Resources** menu.
2. Click a CDN Hostname.
3. On the screen that appears, click the **Advanced Details** tab.

For details on the Advanced Settings for each resource, refer to the following sections.

3.4.2.1 View HTTP Resource Details
To view instructions and basic settings of an HTTP CDN resource:
1. Go to your Control Panel > **CDN > Resources** menu.
2. Click the required CDN Hostname.
3. On the page that appears, click **Basic Settings** tab. The screen provides you with the following information:

**CDN Resource details**
- **Owner**
• CDN hostname
• Resource type - Push or Pull

• Let's Encrypt SSL - whether Let's Encrypt SSL is enabled for the resource or not

To see more details on the Let's Encrypt SSL status, select the Click to view more option.

• SSL on - whether SSL is enabled for the resource or not
• SNI SSL Certificate - custom SNI SSL certificate associated with the resource
• CDN reference – the ID of the resource in database
• Resource status – shows the resource status.

Origins (HTTP Pull only)

• Path of the content that will be served from the CDN.

DNS Settings
• Add a CNAME for the CDN Hostname which can then be used to view the contents.

Uploaded files (HTTP Push only)

A list of uploaded files. Click the file name to preview the video file.

Edge Groups
• Shows to which Edge groups the resource is assigned.

Last 24 hours cost
• Cost of the resource for the last 24 hours.

3.4.2.1.1 Advanced details
Click the Advanced details tab to view advanced details of an HTTP CDN resource. The screen that appears will provide you with the following information:

• Publisher name - name of the user who created the CDN Resource record
• IP Access Policy - access policy from a range of IP addresses: either NONE (disabled), ALLOWED BY DEFAULT or DISABLED BY DEFAULT
• Country Access Policy - access policy to the CDN resource's content for specified countries: either NONE (disabled), ALLOWED BY DEFAULT or DISABLED BY DEFAULT
• Url Signing - whether access requires URL signing or not
• Hotlink Policy - whether hotlink policy is enabled or not
• Password On - whether the password is enabled or not
• Cache Expiry - cache expiry time in minutes
• MP4 Pseudo Streaming - whether the MP4 pseudo streaming is enabled or not
• FLV Pseudo Streaming - whether the FLV pseudo streaming is enabled or not
• Ignore Set-Cookie - whether content caching with SetCookie response headers is enabled or not
• Origin Policy
• **Nginx Settings:**
  
  - **Limit rate** - sets the speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
  - **Limit rate after** - sets the amount after which the speed of a response to a client will be limited in MB. Maximum limit rate after value - 2147483647 KB
  - **Proxy cache key** - key for caching

• **Block search engine crawlers** - whether search engine crawlers are blocked from indexing the CDN content or not (for HTTP Pull CDN resources only)

3.4.2.1.2 Upload instructions (HTTP Push only)
Click the **Instructions** tab to view the instructions for uploading files and embedding video from HTTP Push CDN resources.

3.4.2.1.3 Prefetch/Purge CDN content (HTTP Pull only)
To **prefetch** or **purge** the resource content of the HTTP Pull CDN resource, click the required tab respectively.

3.4.2.1.4 View CDN resource billing statistics
To view the resource billing statistics, click the **Billing Statistics** tab.

3.4.2.2 View VoD Resource Details
To view details of a video on demand CDN resource:

1. Go to your Control Panel > **CDN > Resources** menu.
2. Click the CDN hostname of a required VoD CDN resource.
3. On the page that appears you will see basic resource settings:

  **CDN Resource details**
  
  - **Owner**
  - **CDN hostname**
  - **Resource type** - VOD Push or VOD Pull
  - **CDN reference** – the ID of the resource in database
  - **Resource status** – shows the resource status.

**Origins (VOD Pull only)**

  - Path of the content that will be served from the CDN.

**DNS settings**

  - CNAME for the CDN Hostname which is used to view the contents.

**Uploaded files (VOD Push only)**

  - A list of uploaded files. Click the file name to preview the video file.

**Edge Groups**

  - Shows to which Edge groups the resource is assigned.
3.4.2.2.1 Advanced details
Click the Advanced details tab to view advanced details of a VOD CDN resource. The screen that appears will provide you with the following information:

- **Publisher name** - name of the user who created the CDN Resource record
- **Country Access Policy** - access policy to the CDN resource’s content for specified countries: either NONE (disabled) or BLOCK BY DEFAULT
- **Hotlink Policy** - whether hotlink policy is enabled or not
- **Secure Wowza** - whether secure Wowza token is enabled or not
- **Token for Edge/Flash player** - whether token for Edge/Flash player is enabled or not
- **Token Authentication Enabled** - whether token authentication is enabled or not

3.4.2.2.2 Upload instructions
Click the Instructions tab to view the instructions for uploading files and embedding video from Video On Demand CDN resources.

3.4.2.2.3 View CDN resource billing statistics
To view the resource billing statistics, click the Billing Statistics tab.

3.4.2.3 View Live Streaming Resource Details
To view details of a live streaming CDN resource:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the CDN Hostname of a required live streaming CDN resource.
3. On the page that appears you will see basic resource settings:

**CDN Resource details**
- **Owner**
- **CDN hostname**
- **Resource type** - Live Streaming
- **CDN Reference** the ID of the resource in database
- **Publishing Point** - the publishing point type: external or internal
- **Main Internal Publishing Location** - main internal publishing point URL address
- **Failover Internal Publishing Location** - internal publishing point failover URL
- **Resource status** – shows the resource status

**DNS settings**
- **CNAME** for the CDN Hostname which is used to view the contents.

**Edge Groups**
- Shows to which Edge groups the resource is assigned.

**Last 24 hours cost**
- Cost of the resource for the last 24 hours.
3.4.2.3.1 Advanced Settings
Click the Advanced details tab to view advanced details of a Live Streaming CDN resource. The screen that appears will provide you with the following information:

1. Go to your Control Panel > CDN > Resources menu.
2. Click a CDN Hostname.
3. On the screen that appears, click the Advanced Details tab.
4. This screen provides you with the following information:
   - Publisher name - name of the user who created the CDN Resource record
   - Country Access Policy - access policy to the CDN resource's content for specified countries: either NONE (disabled) or BLOCK BY DEFAULT
   - Hotlink Policy - whether hotlink policy is enabled or not
   - Secure Wowza - whether secure Wowza token is enabled or not
   - Token for Edge/Flash player - whether token for Edge/Flash player is enabled or not
   - Token Authentication Enabled - whether token authentication is enabled or not

3.4.2.3.2 Upload instructions
Click the Instructions tab to view the instructions for uploading files and embedding video from Live Streaming CDN resources.

3.4.2.3.3 View CDN resource billing statistics
To view the resource billing statistics, click the Billing Statistics tab.

3.4.3 Create HTTP Resource
To add an HTTP CDN resource:

1. Go to your Control Panel > CDN > Resources menu. The page that loads shows the list of CDN resources.
2. To create a new CDN resource, click the "+" button in the top right corner or the CDN Resource Wizard button.
3. Follow the steps of the CDN resource creation wizard:

On this page:

- Type Select
- Properties
- Edge Locations
- Advanced Settings

3.4.3.1 Type Select
Click HTTP to select the required resource type, and then click Next to proceed.

3.4.3.2 Properties
- CDN hostname – the hostname from which you will serve static content. E.g. if your site (origin) is onapp.com, and you want to serve static content from the CDN and make it available at static.onapp.com, then static.onapp.com would be the CDN hostname.
• **Enable SSL** - move the slider to the right to enable the secure socket protocol for your CDN resource.

If the SSL protocol is enabled, you can only have fourth-level domain names. If the CDN hostname ends with `.r.worldssl.net`, SSL will be enabled automatically.

A CDN resource can only be linked to one SSL certificate - Let's Encrypt SSL, shared, or custom SNI.

  o **Let's Encrypt** - select this option if you want to use a Let's Encrypt SSL certificate for the resource

The Let's Encrypt SSL certificate is automatically generated for the following types of hostname:

- CDN hostname (e.g. `cdn.abc.com`)
- Alias (CNAME) for CDN hostname (e.g. `123.r.worldcdn.net`)
- Operator Basehostname (e.g. `123.r.worldcdn.net`)
- Secondary CDN hostname (e.g. `cdn1.abc.com`)

All the hostnames are bundled into one Let's Encrypt SSL certificate. If the secondary hostname cannot be validated, the system generates the LE certificate based on the CDN hostname, CNAME, and Operator Basehostname. The unverified hostname is revalidated by the system every 15 minutes.

  o **Shared SSL** - choose this option if you want to apply a shared SSL certificate for the resource

  o **Custom SNI SSL** - choose this option if you want to apply a custom SNI SSL certificate for the resource and choose the required certificate from the drop-down list

• **Content origin** – specify the content origin type (PULL or PUSH):

  o For the PULL type, you can use a custom origin port. Specify a port number using the colon character (`:`) in the Origins field. If you do not indicate the origin port, then the system will put it by default depending on origin policy:

    - 80 if origin policy is HTTP
    - 443 if origin policy is HTTPS
    - None if origin policy is AUTO, that is when the origin port is custom

The valid port values include 80, 443, and the range from 1024 to 65535. Values other than mentioned above will be forbidden.
In case of using multiple origins, the same port number should be specified for all origins using a colon character (":"). Erase the port number from the origin resource field to reset the custom origin port.

Please note that you can specify one or more origins in the Origins field:

- When a single origin is used, specify the value based on a hostname or IP.
- When more than one origin is used, only IP addresses are allowed.

For IP-based origins, ensure your origin is configured to accept HTTP requests with a CDN resource primary hostname as a Host header from CDN edge servers.

When creating a CDN resource, specify domain.com or subdomain.domain.com in the Origins field. domain.com/folder is not allowed in the Origins field. If you want CDN to cache the content only in domain.com/folder, move the content from domain.com/folder to the subdomain folder.domain.com, which should be used as an origin.

- For the PUSH type:
  - Storage server location - choose the storage server location from the drop-down list
  - FTP password - specify the FTP password. It can consist of 6-32 alphanumeric characters
  - FTP password confirmation - confirm the password

3.4.3.3 Edge Locations
Tick the checkbox next to the group(s) that will share the new resource. Available groups depend on the assigned billing plan limits.
The map displays own, subscribed and available CDN resources:
At this point, you can create the CDN resource or proceed to the Advanced Settings step which is optional in the wizard.

3.4.3.4 Advanced Settings

**Origin Policy**
The origin policy, which is available to a CDN resource, allows a CDN edge server to fetch content from the origin by using different HTTP or HTTPS protocol. Select the type of the connection from the drop-down list:

- **HTTP**—connection between an edge server and an origin where HTTP is always used. It is a default value
- **HTTPS**—connection between an edge server and an origin where HTTPS is always used
- **AUTO**—connection between an edge server and an origin based on a visitor’s request (HTTP or HTTPS)

**Country Access**
Configure a rule to enable/disable access to the CDN resource’s content for specified countries.

- **Access Policy** – select Disabled to switch off the rule, otherwise, choose between Allow by default/Block by default.
- **Except for Countries** – select countries to which the access policy won’t be applied. To select more than one country, hold Ctrl during selection.

**Hotlink Policy**
• **Hotlink Policy** – select Disabled to switch off a hotlink policy, otherwise, choose between Allow by default/Block by default.

• **Except for domains** – specify domains to which the hotlink policy won’t be applied.

1. Please note that a wildcard is not supported. Instead, use a URL.

**IP Access**

Configure a rule to enable/disable access to the CDN resource’s content for a range of IP addresses.

• **Access Policy** – select Disabled to switch off the rule, otherwise, choose between Allow by default/Block by default.

• **Except for IP Addresses** – fill in IP address(es) to which the access policy won’t be applied.

**Secondary CDN Hostnames**

Submit secondary hostnames apart from the default one for HTTP based CDN sites. With these configured, users will be able to access the CDN site using secondary CDN hostname(s). You can add up to 7 secondary CDN hostnames to your CDN resource.

To be able to use a secondary hostname for a CDN resource with SSL enabled, you need an SSL certificate for your custom hostname. For more details about the purchase of an SSL certificate, contact OnApp Support. Also, set CNAME for both CDN hostname and secondary CDN hostnames.

If you create a CDN resource with the following settings:

- **CDN hostname**—cdn.example.com
- **Origin**—example.com
- **Secondary hostname**—cdn1.example.com, cdn2.example.com

and visitors visit cdn.example.com, cdn1.example.com, cdn2.example.com, all the three URLs will be displayed as example.com.

**URL Signing**

Protect your files from unauthorized access with a key. A signed URL looks like

```
http://example.com/filename?hash=DMF1ucDxtqgxwYQ==
```

• **Enable URL Signing** – move the slider to the right to enable it.

• **URL Signing Key** – fill in the key which will be used for URL signing. The secret key is similar to a password and can contain a minimum of 6 to a maximum of 32 characters. Spaces are not allowed.

A signed URL can be in one of the following formats:
• Query string format:
  http://example.com/filename?secure=DMF1ucDxtqgxwYQ&expires=1546300800&ip=1.2.3.4
• Path format:
  http://example.com/secure=DMF1ucDxtqgxwYQ&expires=1546300800&ip=1.2.3.4/filename

A secure token consists of the following parameters:

- **Expires**—the expiration time of a URL or the time when an URL becomes invalid. The time is passed in the URL itself in a Unix timestamp format and takes part in hash generation. It is an optional parameter
- **Path**—a file path or file directory

  For HLS, put a path instead of an M3U8 file, so that all the chunks of the HLS are authenticated as well.

- **Key**—a URL signing key
- **IP**—an IP that provides access. It is optional and only one IP allowed when generating the hash key each time

Here is the format of a secure token:

```
<expires><path><key><ip>
```

Here is an example of the PHP script used to generate the hash key:
Create hash link CDN resource

@param string $cdnResourceUrl
The CDN resource URL, eg cdn.yourdomain.com

@param string $filePath
File path of the CDN resource

@param string $secretKey
The secret key that is obtained from CDN resource property

@param int $expiryTimestamp [optional]
UNIX timestamp format, specify how long the hash link is accessible to the public
By default will be accessible forever.

@return string URL with generated hash link
URL with designated format to access the resource

Example:
Generate hash link for resource www.example.com/images/photo.png for next 3 days, assume today is Sun, 01 Apr 2012.

```php
<?php
$hashLink = generateHashLink('www.example.com', '/images/photo.png', 'l33tf0olol', 1333497600);
print $hashLink;
?>
```

http://www.example.com/images/photo.png?secure=kaGd_cu6Iy4LDgfX3jy5Rw==,1333497600

```
function generateHashLink($cdnResourceUrl, $filePath, $secretKey, $expiryTimestamp = NULL)
{
    // NOTE [yasir 20110331] + and ? are some of represented chars of based64 encoding (8 bits)
    // + is 62 and / is 63 . These char should be replaced by other predefined chars.
    $searchChars = array('+','/');
    $replaceChars = array('-','_');
    if($filePath[0] != '/'){ $filePath = '/{$filePath}'; }
    if($pos = strpos($filePath, '?')){ $filePath = substr($filePath, 0, $pos); }
    $hashStr = $filePath.$secretKey;
    if($expiryTimestamp){
        $hashStr = $expiryTimestamp.$hashStr;
        $expiryTimestamp = "",{$expiryTimestamp};
    }
    return "http://{$cdnResourceUrl}{$filePath}?secure=".
        str_replace($searchChars, $replaceChars, base64_encode(md5($hashStr, TRUE))).
        $expiryTimestamp;
}
```

To generate the hash key, download the Ruby, Python, PHP, or Java script.
Ruby:
ruby UrlSigning.rb -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

Python:
python UrlSigning.py -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

PHP:
php UrlSigning.php -f path -s https -r example.com -p images/photo.png -k abc123s -e 1546300800 -i 1.2.3.4

Java:
javac UrlSigning.java
java UrlSigning -f path -s https -r example.com -p images/photo.png -k abc123 -e 1546300800 -i 1.2.3.4

Available options:
-f: format, path or querystring, default = querystring
-s: scheme for resource URL, http or https, default = http
-r: resource hostname (compulsory)
-p: file path of the resource, default = /
-k: URL signing key (compulsory)
-e: expiration of the URL (optional)
-i: IP that allow to access (optional)

Cache expiry

- Cache expiry – set the cache expiry time in minutes (min=1, max=35000000).

Password

- Enable Password – move the slider to the right to restrict access to the resource (cdn hostname).
- Unauthorized HTML – fill in the text which will be displayed for unauthorized login.
- Username – choose a username.
- Password – select password for the user.

To remove a user, clear both fields.

Pseudo Streaming

- Enable MP4 pseudo streaming – move the slider to the right to enable the pseudo streaming support for MP4 file type.
- Enable FLV pseudo streaming – move the slider to the right to enable pseudo streaming for FLV file type, respectively.

With pseudo streaming enabled, your viewers can seek around a video even if it has not finished downloading. A Flash player and a prepared video are required for pseudo-streaming.
Ignore Set-Cookie

- **Ignore Set-Cookie** - move the slider to the right to enable caching content with Set-Cookie response headers.

Nginx Settings

- **Limit rate** - set speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
- **Limit rate after** - the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value -2147483647 KB
- **Proxy cache key** - key for caching. Select one of four supported types from the drop-down list:
  - $host$request_uri
  - $host$uri
  - $proxy_host$request_uri
  - $proxy_host$uri

Due to Nginx limitations, file extensions are not supported for gzip. It supports only specific MIME types (the Content-Type header returned from an origin). You can change the MIME types in the origin using the parameters from the whitelist.

**Click here to view the whitelist**

- text/compressible
- text/html
- text/xml
- text/javascript
- text/css
- application/x-javascript
- text/xml
- application/xml
- application/xml+rss
- application/json
- application/javascript
- image/svg+xml
- image/x-icon
- image/vnd.microsoft.icon
- application/x-font-ttf
- application/vnd.ms-fontobject
- application/x-font-opentype
- application/x-font-truetype
- application/x-font-ttf
- font/eot
- font/opentype
- font/otf
The MIME types included in the whitelist are gzipped on demand on CDN edge servers. Any MIME types that are not listed but can be further compressed can be added to the whitelist. The MIME types in the whitelist can be requested either in a gzip or non-gzip file format based on the passed headers. The reason why it is based on the passed headers is serving compressed content without the client requesting it and risks of breaking client/browser that do not handle Content-Encoding: gzip properly.

**Search Engine Crawlers**
- *Block search engine crawlers* - move the slider to the right to block web crawling bots from indexing the CDN content (for HTTP Pull CDN resources only).

To ensure Samsung Smart TV's compatibility with Microsoft Smooth Streaming through the CDN, the Suppress CDN Headers feature must be enabled. Contact [OnApp Support](#) to have this feature available in Control Panel.

1. 4. Click **Create CDN Resource**.

### 3.4.4 Create Video On Demand Resource

#### 3.4.4.1 Requirements to Provide VoD Services

To provide VoD CDN service to your clients, you need the following:

- Streaming edge servers ([created on your own in Control Panel](#)) or subscription for the streaming location from CDN federation. Assign these streaming location into the edge group of the billing plan used by your client. It is applicable to both VoD Pull and VoD Push CDN resources.
- The streaming CDN storage server, which is applicable only to VoD Push CDN resources. It is for your clients to be able to upload their video files.

Both streaming edge servers and storage servers require a Wowza charge of $50/month/instance.

On this page:

- Requirements to Provide VoD Services
- Create Video On Demand CDN Resource
  - Type Select
  - Properties
  - Edge Locations
  - Advanced Settings

#### 3.4.4.2 Create Video On Demand CDN Resource

To add a VoD CDN resource:
1. Go to your Control Panel > CDN > Resources menu.

2. Click the "+" button in the top right corner or the CDN Resource Wizard button.

3. Follow the steps of the CDN resource creation wizard:

3.4.4.2.1 Type Select
Click VOD to select the required resource type, and then click Next to proceed.

3.4.4.2.2 Properties
• CDN hostname – specify the name which will serve as a label only

For VoD CDN resources, a CNAME is not required to be set for a CDN hostname, as the CDN hostname is not used. For example, if the CDN hostname is cdn.domain.com, the video files are always served from video.domain.com. A CDN hostname is used for labelling purposes in Control Panel. If the CDN hostname of a VoD resource has a configured CNAME, e.g., xxx.r.worldcdn.net (where xxx is a resource ID), a message with a Wowza version will appear by default when visitors access the CDN hostname.

• Content origin – specify the content origin type, PULL or PUSH

If you have selected the PULL type, specify the origin. This is where your VOD video files are located at. For example if your video files are hosted at http://example.com/video.mp4, then please fill in "example.com" in the origin field.

If you have selected the PUSH type:

• Storage server location - choose the storage server location from the drop-down list. Creating VOD-PUSH resource would require you to upload the Video files to "CDN Storage Server" via FTP, so you will need to create a CDN storage server beforehand. Our system would then push the files from CDN storage servers to the Wowza edge servers

• FTP password - specify the FTP password. It can consist of 6-32 alphanumeric characters

• FTP password confirmation - confirm the password

3.4.4.2.3 Edge Locations
Tick the checkbox next to the group(s) which will share the resource added. Available groups depend on the assigned billing plan limits.
The map displays own, subscribed and available CDN resources:
3.4.4.2.4 Advanced Settings

Country Access

Configure a rule to enable/disable access to the CDN resource’s content for specified countries.

- **Access Policy** – select the Disabled option to switch off a rule or Block by default. If the access policy is set to block by default, fill in the Except for Countries field to specify countries to which the access policy won’t be applied. To select more than one country, hold Ctrl during selection.

Hotlink policy

- **Hotlink policy** - select the Disabled option to switch off hotlink policy security; otherwise choose Block by default. If the hotlink policy is set to block by default, fill in the Except for domains field to specify the domains to which the hotlink policy won’t be applied.

Cache expiry

- **Cache expiry** – set the cache expiry time in minutes (min=1, max=35000000).

For a VoD Push CDN resource, video files are cached on the streaming edge servers for 24 hours. For a VoD Pull CDN resource, video file caching honors the expiry host header from the origin. If the origin sends a **max-age=1000**, it means that the video files will be stored for 1,000 seconds.

Ignore Set-Cookie

- **Ignore Set-Cookie** - tick this checkbox to enable caching content with Set-Cookie response headers.
Search Engine Crawlers

- **Block search engine crawlers** - move the slider to the right to block web crawling bots from indexing the CDN content (for HTTP Pull CDN resources only).

Secure Wowza

- **Enable secure Wowza** – tick the checkbox to protect your stream with Wowza secure token.
- **Token for Edge/Flash player** – specify the token authentication code. The authentication code will be then saved on the edge server. The token at the player side and the token at the edge server must match in order to make the video available.

VOD setup requires the origin to send the Accept-Ranges: Bytes header in order for Wowza to make range requests to the Nginx cache. Amazon S3 does not send the header by default. For this, add the Accept-Ranges: Bytes header to the files stored in Amazon S3.

Token Authentication

- **Enable Token Authentication** - tick the checkbox to enable token authentication.
- **Token Auth Primary Key** - specify the secret key to be used with the scripts which generate token.
- **Token Auth Backup Key** - input the backup key which can be used if you want to change the primary key. To make sure the link generated with existing token won’t be broken, the suggested procedure is the following:
  - Copy the existing primary key and paste to backup key.
  - Fill in a new primary key.
- **Protected Path** - specify the protected path. By default it is "/". Only the Path inserted is allowed to be streamed.

Now proceed with running the .NET or JAVA scripts to complete the procedure.

4. Click **Create CDN Resource**.

Only mp4 and flv files are currently supported.

For a VoD CDN resource, use unicast to stream a video file (not live streaming with a camera).

**Click here to see more details**

- The client's player first saves part of the video (not the whole file) and then plays it.
- The part of the video is sent via unicast connection. If the video bitrate is 300 Kbps and there are 100 people watching the video, it requires 30,000 Kbps.
3.4.5 Create Live Streaming Resource

3.4.5.1 Requirements to Provide Live Streaming Services
To provide Live Streaming CDN service to your clients, you would require at least one of the following:

- Streaming edge servers (created on your own in control panel). These streaming edge servers and storage servers would require Wowza charges, which will be forwarded to your account at $50/month/instance.
- Subscription for the streaming location from CDN federation. Assign these streaming location into the edge group of the billing plan, used by your client. It is applicable to both internal and external live streaming publishing points.

On this page:

- Requirements to Provide Live Streaming Services
- Create Live Streaming CDN Resource
  - Type Select
  - Properties
  - Edge Locations
  - Advanced Settings

3.4.5.2 Create Live Streaming CDN Resource
To add a live streaming CDN resource:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the "+" button in the top right corner or the CDN Resource Wizard button.
3. Follow the steps of the CDN resource creation wizard:

3.4.5.2.1 Type Select
Click LIVE STREAMING to select the required resource type, and then click Next to proceed.

3.4.5.2.2 Properties
- CDN hostname – specify the name which will serve as a label only

For Live Streaming CDN resources, a CNAME is not required to be set for a CDN hostname, as the CDN hostname is not used. For example, if the CDN hostname is cdn.domain.com, the video files are always served from video.domain.com. A CDN hostname is used for labelling purposes in Control Panel. If the CDN hostname of a Live Streaming resource has a configured CNAME, e.g., xxx.r.worldcdn.net (where xxx is a resource ID), a message with a Wowza version will appear by default when visitors access the CDN hostname.
HTTPS support for a live streaming resource is limited to a playback via the video.worldcdn.net CDN hostname. Currently, HTTPS support is unavailable for white label CDN hostnames.

- **Publishing point** – specify the resource’s publishing point settings - Internal or External. The internal publishing point is one of your storage servers. It is configured at the next step - Edge locations. The external publishing point is simply an URL. Specify its settings here:

  - **External publishing location** - specify your publishing point's URL as an RTMP protocol. For example, rtmp://domain.com/xxx

  An external publishing point must start with an rtmp, e.g., rtmp://xxxxx.xxxx.com/live. It is possible to exclude the stream name in the external publishing point URL. For example, if the URL of an RTMP stream is rtmp://mystreamingurl.com/live/stream1, insert rtmp://mystreamingurl.com/live/ for the external publishing point URL. Once it is restreamed across the CDN, it can be viewed in RTMP, HDS, HLS, Microsoft Smooth Streaming, and RTSP protocols. The connection flow would be as follows: Viewer’s PC > Closest Streaming Edge > External Publishing Point.

  For more details on limitations, click here
  - Wowza supports only restreaming of h.264+AAC to HLS/HDS.
  - For HLS/HDS restreaming, only Wowza RTMP origin is supported.

- **Failover external publishing location** - specify the failover URL

  Please note that username and password used to connect to a live stream publishing point cannot be changed.

3.4.5.2.3 Edge Locations
Tick the checkbox next to the group(s) which will share the resource added. Available groups depend on the assigned billing plan limits.
The map displays own, subscribed and available CDN resources:
In case of choosing Internal Publishing Point in previous step, specify its settings here:

- **Internal publishing location** - select any of your edge servers from the drop-down list
- **Failover internal publishing location** - specify the failover edge server

### 3.4.5.2.4 Advanced Settings

#### Country Access

Configure a rule to enable/disable access to the CDN resource’s content for specified countries.

- **Access Policy** – select the **Disabled** option to switch off a rule or Block by default. If the access policy is set to block by default, fill in the **Except for Countries** field to specify countries to which the access policy won’t be applied. To select more than one country, hold Ctrl during selection.

#### Hotlink policy

- **Hotlink policy** - select the **Disabled** option to switch off hotlink policy security: otherwise choose Block by default. If the hotlink policy is set to block by default, fill in the **Except for domains** field to specify the domains to which the hotlink policy won’t be applied.

> The Hotlink policy feature is applicable to RTMP and RTSP only. For HLS streaming, it is not supported. If you need to enable the hotlink policy for HLS streaming, it is recommended to create an HTTP Pull resource.

#### Secure Wowza

- **Enable secure Wowza** – tick the checkbox to protect your stream with Wowza secure token.
• **Token for Edge/Flash player** – specify the token authentication code. The authentication code will be then saved on the edge server. The token at the player side and the token at the edge server must match in order to make the video available.

**Token Authentication**

• **Enable Token Authentication** - tick the checkbox to enable token authentication.

• **Token Auth Primary Key** - specify the secret key to be used with the scripts which generate token.

• **Token Auth Backup Key** - input the backup key which can be used if you want to change the primary key. To make sure the link generated with existing token won’t be broken, the suggested procedure is the following:
  
  o Copy the existing primary key and paste to backup key.
  
  o Fill in a new primary key.

• **Protected Path** - specify the protected path. By default it is "/". Only the Path inserted is allowed to be streamed.

Now proceed with running the .NET or .JAVA scripts to complete the procedure.

With an HTTP Pull CDN resource, you can restream an HLS stream. To do it, you need to create an HTTP PULL resource.

When creating the HTTP Pull resource, in the **Origins** field, specify the HLS URL (without a path and a filename), and move the **Enable HLS Optimization** slider to the right.

**Click here to see the possibilities this feature provides**

• Cache in the edge server RAM rather than in a hard disk

• An automatic cache expiry rule with 10 seconds for video content

• The resource will have a 30-second DNS TTL

4. Click the **Create CDN Resource** button.

For a Live Streaming CDN resource, use multicast.

**Click here to see more details**

• The stream is never saved in the client’s hard drive.

• The stream is distributed via multicast. If the video bitrate is 300 Kbps and there are 100 people watching the video, it requires only 300 Kbps for the camera to distribute it to CDN PoPs.

If a CDN live streaming resource is used as an origin, it may cause a poor cache hit ratio on HLS.

To improve the cache hit ratio, refer to the instructions on how to configure Wowza as an HTTP caching origin.
3.4.6 Edit CDN Resources

To edit a CDN Resource:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the Actions button next to the CDN resource’s label that you want to edit and then select Edit.
3. On the page that appears, you can edit all CDN resource parameters (see the Create HTTP CDN Resource, Create Video On Demand Resource, Create Live Streaming Resource sections for details).
4. Click Save to save the changes.

3.4.7 CDN Raw Logs

The raw logs functionality allows you to send logs associated with your CDN resources to your distant server in real time. The raw log allows customers to understand, analyze, and debug files delivered via OnApp CDN, or can be served as audit trailed. Once the user creates CDN resource(s), the raw logs are enabled for this account and the user can configure and receive raw logs. Logs are sent for all the CDN Resources associated with the user. There are three different types of delivery: Syslog, SFTP and FTP. The frequency of uploading the log to client destination is every 10 minutes for SFTP and FTP protocols. For the Syslog protocol, logs are uploaded instantaneously.

If the user does not have any CDN resources, they cannot configure raw logs.

To view and set the raw log configuration:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the Raw Log tab.
3. The page that loads shows the current raw log configuration. On this page you can also set the raw log configuration:
   - For the FTP/SFTP delivery protocol:
     - Hostname - fill in the hostname of the server to which the log will be delivered
     - Ftp username - specify the user name of the FTP/SFTP client on the server to which the log will be delivered
     - Ftp password - fill in the password of the FTP/SFTP client on the server to which the log will be delivered
   - For the Syslog delivery protocol:
     - Hostname - fill in the hostname of the server to which the log will be delivered
     - Syslog protocol - select the protocol that will be used for sending the log: TCP or UDP
     - Syslog port - specify the port number of the syslog server to which the log will be delivered
   - Choose Disabled to disable raw logs.
4. Click Save to save the configuration.
To edit, set new configuration parameters and click **Save**.

To disable, select **Disabled** from the raw log configuration delivery protocol drop-down list.

The list of raw log parameters includes the following:

- **$upstream_response_time** - keeps time spent on receiving the response from the upstream server; the time is kept in seconds with millisecond resolution. Times of several responses are separated by commas and colons like addresses in the **$upstream_addr** variable.

- **$upstream_http_x_cache** - a value of the X-Cache HTTP response header field from the upstream server.

- **$request_time** - the time elapsed between the first bytes that have been read from the client and the writing log after the last bytes have been sent to the client.

- **$tcpinfo_rtt** - the round trip time (**ping**) estimate of the TCP socket in microseconds.

- **$http_referer** - the HTTP request header field that identifies the address of the webpage (i.e., the URI or IRI) linked to the resource requested.

- **$for_operator** - a buyer (if any) subscribed to the edge server.

If the SFTP or FTP protocol is applied, raw logs are delivered as an archive. If the Syslog delivery protocol is selected, user will receive the text of the logs.

There are different log formats for the Syslog and FTP/SFTP delivery protocols.

An example of the HTTP log format (V8) for the Syslog delivery protocol:

```
$time_local<TAB>$edge_server_id<TAB>$remote_addr<TAB>$request_method<TAB>$http_host<TAB>$request_uri<TAB>$status<TAB>$bytes_sent<TAB>$upstream_http_x_cache<TAB>$upstream_response_time<TAB>$request_time<TAB>$tcpinfo_rtt<TAB>$for_operator<TAB>$resource_id<TAB>$server_name<TAB>$server_addr<TAB>$http_user_agent<TAB>$http_referer<TAB>$http_range<TAB>$body_bytes_sent<TAB>$publisher_id<TAB>$tls_version<TAB>$placeholder1<TAB>$placeholder2<TAB>$placeholder3<TAB>$placeholder4<TAB>$placeholder5<TAB>$placeholder6<TAB>$placeholder7<TAB>$placeholder8<TAB>V8
```

An example of the Steam log format for the Syslog delivery protocol:
An example of the HTTP log format for FTP/SFTP delivery protocols:

```
$date\TAB\time\TAB\edge-id\TAB\type\TAB\x-app\TAB\x-severity\TAB\x-category\TAB\x-event\TAB\date\TAB\c-client-id\TAB\c-ip\TAB\c-port\TAB\cs-bytes\TAB\sc-bytes\TAB\x-duration\TAB\x-sname\TAB\x-stream-id\TAB\x-spos\TAB\sc-stream-bytes\TAB\cs-stream-bytes\TAB\x-file-size\TAB\x-file-length\TAB\x-ctx\TAB\x-comment\TAB\x-sname-query
```

An example of the HTTP log format for FTP/SFTP delivery protocols:

```
$remote_addr - - [$time_local] "$request_method $request_uri HTTP/1.1"
$http_status $bytes_sent "$http_referer" "$http_user_agent"
```

An example of the Stream log format for FTP/SFTP delivery protocols:

```
x-app\TAB\x-severity\TAB\x-category\TAB\x-event\TAB\date\TAB\c-client-id\TAB\c-ip\TAB\c-port\TAB\cs-bytes\TAB\sc-bytes\TAB\x-duration\TAB\x-sname\TAB\x-stream-id\TAB\x-spos\TAB\sc-stream-bytes\TAB\cs-stream-bytes\TAB\x-file-size\TAB\x-file-length\TAB\x-ctx\TAB\x-comment\TAB\x-sname-query
```

3.4.8 View Advanced Reporting
To view advanced reporting on bandwidth statistics for HTTP type CDN resources:
1. Go to your Control Panel > CDN > Resources menu
2. Click the Actions button next to a required CDN Hostname and choose Advanced Reporting.
3. On the page that appears, the .json report will show the total/cached/non-cached/hits/misses statistics. The default period is the last week.

3.4.9 Prefetch CDN Resources Content
This tool allows prepopulating content of an HTTP Pull or HTTP Push CDN resource to the CDN. Recommended only for files which are especially large.

To prefetch content:
1. Go to your Control Panel > Resources > CDN resource’s label.
2. On the page that appears, click the Prefetch tab.
3. In the Paths to Prefetch box, specify the destination path(s). You can enter only one path per line.
4. Click Prefetch.
Please note that you can prefetch only the content of HTTP Pull and Push CDN resources.

Also, on the **Prefetch** tab, you can view the statuses of your transactions in the table with the following details:

- **Job ID** - the ID of transaction
- **Url** - the Url path to prefetch
- **Status** - the status of transaction ( % of progress, completed or failed)
- **Time Initiated** - the time when transaction was started
- **Time Completed** - the time when transaction was completed

### 3.4.10 Purge CDN Resources Content

This tool allows instant removal of HTTP Pull and HTTP Push cache content in the CDN, if newly updated content has not been properly replicated.

To purge a single file:

1. Go to your Control Panel > **CDN** > **Resources** > CDN resource's label.
2. On the page that appears, click the **Purge** tab.
3. In the **Paths to Purge** box, specify the destination path(s). You can enter only one path per line.
   
   The maximum capacity of the **Paths to Purge** box is 1,000 lines.

4. Click **Purge**.

To purge all content:

1. Go to your Control Panel > **CDN** > **Resources** > CDN resource's label.
2. On the page that appears, click the **Purge** tab.
3. Click **Purge All Contents of This Site** to purge all content.

Please note that you can purge only the content of HTTP Pull and HTTP Push CDN resources.

Also, on the **Purge** tab, you can view the statuses of your transactions in the table with the following details:

- **Job ID** - the ID of transaction
- **Url** - the Url path to purge
- **Status** - the status of transaction ( queued, % of progress, completed or failed)
- **Time Initiated** - the time when transaction was started
- **Time Completed** - the time when transaction was completed
Be aware that statuses for PurgeAll transactions are not tracked.

Because objects are not stored in the cache directory structure, it is not possible to purge URLs with a wildcard or a directory as in the example below:

```
/path/to/file/*.jpg
/path/to/directory/
```

It is recommended to add a version ID to your URL query string. Apart from invalidating edge server cache, it also invalidates browser cache.

### 3.4.11 Billing Statistics

OnApp has a record of all the charges applied to your CDN resources. You can view the resource statistics under the statistics available, or those for a shorter period by setting a Start and End time.

To view billing statistics for a CDN resource:

1. Go to your Control Panel > CDN > Resources menu.
2. Click the label of the resource you’re interested in and then click the Billing Statistics tab.
3. Set Start and End time.
4. Move the Show in my Timezone slider to the right to show bandwidth statistics according to your profile's timezone settings.
5. Press the Apply button.
6. On the screen that appears, you will see the following billing statistics details:
   - **Date** – particular date and time for the generated statistics
   - **Edge Group** - the edge group to which the CDN resource belongs to.
   - **Traffic** - resource traffic in MB.
   - **Cost** – the total due for the CDN resource at the point of time specified in the Date column.

Scroll down to see Total Amount (the total due for the whole billing statistics period).

### 3.4.12 Token Authentication

Token authentication helps to protect CDN streams from being snitched. Similar to HTTP URL signing, this feature allows customers to enter a secret key during setting up a CDN resource. Then, customers can use the secret key, along with expiry date and allowed/blocked referrer site to generate the token from a script.

To enable token authentication:

1. Enter your secret key and secure path using OnApp UI.
2. Download .NET or Java token generator.
3. Generate the token with the secret key, allow referrer, deny referrer, expiry date, and allow IP.

For more information, refer to the readme file on the token generator. Missing can be applied to the allow referrer if the HTTP referrer is missing from the browser to the video player. In some cases, the referrer header the video player sent to the edge server is the JS script domain instead of the website domain.

4. Append the token with your stream URL, eg ?token=

   110ea31ac69c09a2db0bdd74238843631cdab498ff7e6e75cbd99cc4d05426ab679a57015d
   4e48436c97b921652daec62de3829f8f437e27449cfdfc2f1e5d9fc47f14e91a51ea7

5. Embed with your website.

3.4.12.1 Set up TokenAuthentication via UI
You can set up token authentication for Video on demand and Live streaming CDN resources.

To do so:
2. Now proceed with running the .NET or .JAVA scripts.

3.4.12.2 Execute Token Generator
Download the script from the following locations:

- .NET script: https://bitbucket.org/onappcore/cdn-wowza-token-tool/src/f06c7cc4842a9854ba6759a7ef18191cc2dd60e7/dotnet/?at=master
- Java script: https://bitbucket.org/onappcore/cdn-wowza-token-tool/src/f06c7cc4842a9854ba6759a7ef18191cc2dd60e7/java/?at=master

Refer to the following sections on instructions for running the scripts.

3.4.12.2.1 Generate Token with .NET

Prerequisites:
- .NET Framework 4.5
- BouncyCastle C# Crypto library 1.7 (http://www.bouncycastle.org/csharp/)

On this page:

- Build
- Usage
  - Security Parameters
  - Allow blank/missing referrer
  - Generate Token
  - Decrypt token
3.4.12.2.1.1 Build
To build a generator:
1. Go to /cdn-wowza-token-tool/dotnet/src location
2. Run xbuild
Upon success of the build, you will find the .exe (TokenAuthGenerator.exe) file at the 'TokenAuthGenerator/bin/Debug' folder.

3.4.12.2.1.2 Usage
TokenAuthGenerator.exe (encrypt | decrypt) (primary_key | backup_key) "security_parameters"

3.4.12.2.1.2.1 Security Parameters
expire
- Number of seconds since Unix time (Epoch time)
- UTC based
- Must not be earlier than current time
ref_allow
- Referrer domain (e.g. google.com) or path (e.g. google.com/video/)
- Allow multiple referrers separated by comma (,) without space(s)
- Wildcard (*) allowed only at the beginning of a referrer, e.g. *.DOMAIN
- Do not append space at the start & end of a referrer
- Domain must fulfill RFC 3490
- Path must fulfill RFC 2396
- Should not include port (e.g. google.com:3000/video)
- Should not include protocol (e.g. http) portion
ref_deny
- The same rules as for ref_allow

If both ref_allow & ref_deny are specified, ref_allow will be taking precedence over ref_deny

3.4.12.2.1.2.2 Allow blank/missing referrer
Both "ref_allow" & "ref_deny" could be configured to allow/deny blank or missing referrer during TokenAuth validation.
The following configuration allows blank or missing referrer:
ref_allow=allow.com,
ref_allow=allow.com,MISSING
ref_deny=deny.com

The following configuration deny blank or missing referrer:
ref_allow=allow.com
ref_deny=deny.com,
ref_deny=deny.com, MISSING
Normally ref_allow & ref_deny should not be used together, but if this happened ref_allow will take precedence over ref_deny.

3.4.12.1.2.3 Generate Token
To generate token, run the following:
```
TokenAuthGenerator.exe encrypt samplekey
"expire=1598832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com"
```
Sample Output:

```
token=110ea31ac69c09a2db0bd74238843631cdab498ff7e6e75cb99cc4d05426ab679a
      57015d4e48438c97b921652daec62de3829f8ff437e27449cfdfcc2f1e5d9fc47f14e91a51e
      a7
```

After generating a token, append the result to the playback URL.

3.4.12.1.2.4 Decrypt token
To decrypt a token, run the following:
```
TokenAuthGenerator.exe decrypt samplekey
110ea31ac69c09a2db0bd74238843631cdab498ff7e6e75cb99cc4d05426ab679a57
015d4e48438c97b921652daec62de3829f8ff437e27449cfdfcc2f1e5d9fc47f14e91a51e
```
Output example:

```
security
parameters=expire=1598832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com
```

3.4.12.2 Generate Token with Java

**Prerequisites:**
- Java 6 or 7
- Maven 2 or 3

**On this page:**
- **Build**
- **Usage**
  - Security parameters
  - Allow blank/missing referrer
  - Generate token
3.4.12.2.2.1 Build
To build a generator:
1. Go to /cdn-wowza-token-tool/java/ location.
2. Run the following:
   mvn clean install
Upon success of the build, you will find the jar (token-auth-generator.jar) file at the 'target' folder.

3.4.12.2.2 Usage
java -jar token-auth-generator-1.2.jar (encrypt | decrypt)
( <primary_key> | <backup_key> ) "<security_parameters>"

3.4.12.2.2.2 Security parameters
expire
• Number of seconds since Unix time (Epoch time)
• UTC based
• Must not be earlier than current time

ref_allow
• Referrer domain(e.g. google.com) or path(e.g. google.com/video/)
• Allowed multiple referrers separated by comma (,) without space(s)
• Wildcard (*) allowed only at the beginning of a referrer, e.g. *.DOMAIN
• Do not append space at the start & end of a referrer
• Domain must fulfill RFC 3490
• Path must fulfill RFC 2396
• Should not include port (e.g. google.com:3000/video)
• Should not include protocol (e.g. http)

ref_deny
• Same rules as in ref_allow

If both ref_allow & ref_deny are specified, ref_allow will be taking precedence over ref_deny

3.4.12.2.2.2.2 Allow blank/missing referrer
Both "ref_allow" & "ref_deny" could be configured to allow/deny blank or missing referrer during TokenAuth validation. The following configuration allow blank or missing referrer:
ref_allow=allow.com,
ref_allow=allow.com,MISSING
ref_deny=deny.com

The following configuration deny blank or missing referrer:
ref_allow=allow.com
normally ref_allow & ref_deny are not to be used together, but if this happened ref_allow will take precedence over ref_deny.

3.4.12.2.2.2.3 Generate token
To generate token, run the following:

```
java -jar token-auth-generator-1.2.jar encrypt samplekey
"expire=159832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com"
```

Sample Output:
```
token=110ea31ac69c09a2db0bddd7423843631cdab498ff7e6e75cbd99cc4d05426ab679a57015d4e48438c97b921652daec62de3829f8ff437e27449cfdfc2f1e5d9fc47f14e91a51ea7codecode
```

After generating a token, append the result to the playback URL.

3.4.12.2.2.2.4 Decrypt token
To decrypt token, run the following:

```
java -jar token-auth-generator-1.2.jar decrypt samplekey
110ea31ac69c09a2db0bddd7423843631cdab498ff7e6e75cbd99cc4d05426ab679a57015d4e48438c97b921652daec62de3829f8ff437e27449cfdfc2f1e5d9fc47f14e91a51ea7
```

Sample Output:
```
security
parameters=expire=159832000&ref_allow=*.TrustedDomain.com&ref_deny=Denied.com
```

3.4.13 Edge Server IP Ranges
OnApp CDN allows you to access a list of IP ranges allocated to edge servers by their providers. You can use IP ranges to filter incoming traffic on your origin website by means of firewall rules. Whitelisting edge server IPs enables a firewall to accept incoming traffic from IP addresses within the range and block malicious traffic. In this document, you can find information on where to locate edge server IP ranges at your Control Panel.

You need to have the See own CDN resources permission enabled to access the IP Ranges menu of a CDN resource.
3.4.13.1 View IP Ranges

To view the list of IP ranges allocated to edge servers:

1. Go to the CDN > Resources menu at your Control Panel.
2. Click the IP Ranges tab.
3. On the page that appears, you will see the list of available IP ranges. You can copy the IP range and set up firewall rules on your origin website to accept incoming traffic from IP addresses within this range.

- Since the range of IP addresses may be changed by the edge server provider, it is recommended to use API to retrieve the latest list of IP ranges for your firewall rules. You can also run a cron to schedule jobs for checking changes in the IP range. For more information on API, refer to the Get Edge Server IP Ranges document.
- At Control Panel, the list of IP ranges is automatically updated to include the relevant IPs.

3.5 Edge Groups

CDN edge groups are groups of edge servers – your own, and those you subscribe to from the CDN marketplace. They are usually grouped by location, so they represent a pool of servers for a given geographical area. Once you have created an edge group containing edge servers in specific locations, you can then assign the group (or groups) to a specific CDN resource. The CDN edge groups menu enables you to see available edge server locations and form them into CDN Edge groups.

You need to associate CDN Edge groups with buckets to make them available for users.

3.5.1 View Edge Group Details

To see details of a CDN Edge Group:

1. Go to your Control Panel > CDN > Edge Groups menu.
2. Click the label of the edge group you want to see.
3. On the screen that appears you will see the list of assigned locations and available locations with the following information:
   - ID – the ID of a location
o City – the city the edge server is in.
  o Operator – name of the edge server owner.
  o Type - HTTP or streaming
  o Source – either Marketplace (locations added from the CDN marketplace) or your Own Edge servers (servers added by you).
  o Status - whether edge server is active or not.
  o Price – price per GB transferred.

3.5.2 Create Edge Group

There are two ways of creating a CDN edge group:

1. Using a CDN setup wizard

2. Creating the edge group under the Users and Groups menu

To create a new CDN Edge Group using the Users and Groups menu:

1. Go to your Control Panel > CDN > Edge Groups menu.

2. On the screen that appears, you will see existing groups with the number of assigned locations and associated buckets.

3. Click the Create Edge Group button.

4. On the screen that appears, give your new group a label and click the Create Edge Group button.

5. You will be redirected to the screen where you can assign locations to the group.

For details on CDN setup wizard, refer to CDN wizard section.

3.5.3 Edit/Delete Edge Group

To edit the name of a CDN Edge Group, or delete a group:

1. Go to your Control Panel > CDN > Edge Groups menu.

2. On the screen that appears you will see the list of existing groups:
   • To edit the group's label, click the Actions button, then click Edit.
   • To delete the group, click the Actions icon, then click Delete.

Be careful when deleting an edge group which is associated with CDN resources.

3.5.4 Assign/Remove Edge Group Locations

1. Go to your Control Panel > CDN > Edge Groups menu.

2. Click the label of the CDN Edge Group you want to configure.

3. On the screen that appears you may assign or remove locations by clicking the Actions button next to the required location.
3.6 Upload Instructions

Here is the list of instructions for uploading files and embedding video to CDN resources. Follow the step-by-step instructions below to upload files or embed video to the required CDN resource type.

- HTTP Push CDN Resources
- VOD Pull CDN Resource
- VOD Push CDN Resource
- Live Streaming CDN Resource

3.6.1 HTTP Push CDN Resource

To upload files to the HTTP Push CDN resource:

1. Connect to the FTP origin using an FTP client. For example, a browser plug-in like FireFTP or FTP software like FileZilla.
2. Please wait up to 10 minutes until the FTP server configures with the HTTP resource.
3. Specify the following FTP details:
   - Hostname: 6789.origin.customercdn.com
   - Username: 6789
   - Password: The FTP password set at CDN resource creation.

   This is an instruction template. Replace “6789” with the resource id, and “customercdn.com” with the operator’s domain.
4. Upload your files.

3.6.2 VOD Pull CDN Resource

To upload files to the HTTP Push CDN resource, enter the following script into your web page:

```html
<html>
<head>
  <script src="http://video.worldcdn-beta.net/player.js" type="text/javascript"></script>
</head>
<body>
  <div id="my-video-player"/>
  <script type="text/javascript">
    CDNPlayer("my-video-player", 1234, "1234/mystream", {width:640, height:360});
  </script>
</body>
</html>
```
• This is an instruction template. Replace “1234” with the resource id, “customercdn.com” with the operator’s domain, and “mystream.mp4” with the filename or stream name.

• resource_id/ must prefix the path with <resource_id>, it is ONLY applicable to VOD PULL playback.

• This example provides default values for width and height. You can change them to your own values.

Our easy video embed script automatically detects the browser type (Desktop or Mobile device) and loads the appropriate player. Currently, this is either Flow Player or the browser’s native HTML5 player. The streaming protocol is also set appropriately.

3.6.3 VOD Push CDN Resource

To upload files to the VOD Push CDN resource:

1. Connect to the FTP origin using an FTP client. For example, a browser plug-in like FireFTP, or FTP software like FileZilla.

2. Please allow up to 10 minutes for the FTP server to be configured with the VOD resource.

3. Specify the FTP details:

   o Hostname: 6789.origin.customercdn.com
   o Username: 6789
   o Password: the password set at creation

   This is an instruction template. Replace “6789” with the resource id, and “customercdn.com” with the operator’s domain.

4. Upload your files.

5. Enter the following script into your web page:
OnApp Cloud 6.5 Edge 5 CDN Guide

3.6.4 Live Streaming CDN Resource

1. Before you start, make sure your publishing point settings meet the following requirements (to be able to retrieve with the Silverlight Player):
   - h.264 Baseline 3
   - AAC or MP3-stereo-44100Hz audio
   - 2 seconds key frame frequency
   - lower bitrate

   OnApp CDN does not autotranscode a stream into multiple bitrate or adaptive bitrate. It is recommended to publish multiple streams with a different bitrate from the publishing point and allow end users to choose the stream with the bitrate they need.

2. Install and configure the Adobe Live media encoder:
a. Install Adobe Live Encoder.

b. Once the Adobe Live Encoder is installed, run the application and move on to the next step.

c. Complete the form:
   - FMS URL: rtmp://1234.publishstream.customercdn.com/P1234
   - Backup URL: rtmp://backup.1234.publishstream.customercdn.com/P1234
   - Stream: your stream name

   This is an instruction template. Replace “1234” with the resource id, and “customercdn.com” with the operator’s domain.

d. Press Connect.

e. In the password pop up window, enter “P1234” as the username and the resource secret key for the password.

f. Press Start to start publishing the live stream.

Enter the following script into your web page to embed video to the Live Streaming CDN resource:

```html
<html>
  <head>
    <script src="http://video.worldcdn-beta.net/player.js" type="text/javascript"></script>
  </head>
  <body>
    <div id="my-video-player"/
      <script type="text/javascript">
        CDNPlayer("my-video-player", 1234, "mystream", {width:640, height:360 });
      </script>
    </body>
</html>
```

This example provides default values for width and height. You can change them to your own values.

Our easy video embed script automatically detects the browser type (Desktop or Mobile device) and loads the appropriate player. Currently, this is either Flow Player or the browser’s native HTML5 player. The streaming protocol is also set appropriately.
To stream a video using RTMP, **enable secure Wowza** and **fill in the secure token** after the creation of a live streaming resource.

**Example of the Flow Player embed code:**

```html
<h1>FLOW SMIL</h1>
<a href="http://video.onappcdn.net/12946278/livestream.smil" style="display:block;width:520px;height:330px" id="player">
</a>
<script type="text/javascript" src="flowplayer/flowplayer-3.2.10.min.js"></script>
<script>
flowplayer("player", "flowplayer/flowplayer-3.2.11.swf",

{ plugins: {
smil: { url: "flowplayer/flowplayer.smil-3.2.8.swf" },
rtmp: { url: "flowplayer/flowplayer.rtmp-3.2.10.swf" },
secure: { url: "flowplayer/flowplayer.securestreaming-3.2.8.swf",
token: escape('apppp&&')} },
clip: {
provider: 'rtmp',
autoPlay: false,
urlResolvers: ['smil'],
connectionProvider: 'secure'
}
});
</script>
```

**Click here to see what should be changed**

- `<href="http://video.onappcdn.net/12946278/livestream.smil" > to

- `token: escape('apppp&&')` - change `apppp&&` to the secure token that has been set up.

To publish a stream via FFmpeg, use the following commands:

```
ffmpeg -re -i dummy.mp4 -f flv rtmp://username:password@fms_URL:80/P"CDN_Reference"/mystream
```

**Example:**

```
ffmpeg -re -i dummy.mp4 -f flv rtmp://P404130149:mtXsvYDwQQ@404130149.publishstream.worldcdn.net:80/P404130149/streamname
```
If an external publishing point is used, the customer should have the RTMP URL as in the example below:
rtmp://xxxxxx.xxxx.com/live/streamname1

Click here to see the URL requirements

- URL must start with an rtmp, e.g., rtmp://xxxxxx.xxxx.com/live.
- URL must include the name of a stream specified in the ffmpeg command, which is provided in the command line in the CentOS system, during the streaming to the external publishing point.

This URL (from the external publishing point) should be playable. Then try to restream it using the CDN network.

Create a live streaming resource and insert rtmp://xxxxxx.xxxx.com/live/ in the External publishing location box.

As a result, your CDN URL should be as in the example below:
http://video.yourdomina.net/414746811/_definst_/streamname1.m3u8

3. Manual Instructions

We support a variety of methods to get the CDN URL to use in your player.

SMIL

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.smil
The SMIL playlist provides an RTMP URL and should be used with Flash-based players only. Longtail Player and Flow Player are compatible with SMIL redirection.

Apple HTTP Live Streaming

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.m3u8
This returns a 302 redirect to a Apple HLS manifest and should be used with Apple HLS-compatible players only.

Adobe HTTP Dynamic Streaming

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.f4m
This returns an Adobe HDS manifest and should be used with Adobe HDS-compatible players only.

Microsoft Smooth Streaming (Silverlight)

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.ism
This returns a 302 redirect to Smooth Streaming manifest and should be used with Smooth Streaming-compatible players only.

Javascript JSONP

http://video.cdn.qaonapp.net/726128906/_definst_/mystreamjsonp?callback=MyCallBack
This returns a JSONP document, embeddable using <script>

An example of a callback with a successful result:
MyCallBack({ “rtmp”: “rtmp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream”, “rtmpe”: “rtmp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream”, “apple”: “http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/playlist.m3u8”, “adobe”: “http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/manifest.f4m”, “rtsp”: “rtsp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream”, “silverlight”: “http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/Manifest” });

An example with an error:

MyCallBack({ “error”: “File not found” });

Javascript JSON

http://video.cdn.qaonapp.net/726128906/_definst_/mystream.json

This returns a JSON document. Cross-origin resource sharing is enabled to allow XMLHttpRequest from any domains.

An example of a callback with a successful result:

{ “rtmp”: “rtmp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream”, “rtmpe”: “rtmp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream”, “apple”: “http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/playlist.m3u8”, “adobe”: “http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/manifest.f4m”, “rtsp”: “rtsp://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream”, “silverlight”: “http://609821627.e.726128906.r.cdn.qaonapp.net/726128906/_definst_/mystream/Manifest” };

An example of a callback with a JSON document with an error thrown:

{ “error”: “File not found” }

3.7 SSL Certificates

OnApp customers can import their own SSL certificates with the Subject Name Indication (SNI) extension.

SNI lets the client specify the hostname it is trying to reach at the start of the handshaking process. SNI is supported by most modern browsers, and provides an efficient way to deliver content over HTTPS using your own domain and SSL certificate. Custom SNI
SSL relies on the SNI extension of the Transport Layer Security protocol, which allows multiple domains to serve SSL traffic over the same IP address by including the hostname viewers are trying to connect to.

Previously, OnApp applied SAN SSL certificate from a certificate authority to which additional certified domains can be added. This allowed you to host several domains on one IP by sharing the same certificate, and add all domains to this IP. However, the number of domains per SAN certificate is limited. Moreover, the certificate’s size increases as more domains are added. This causes additional bandwidth to be used for the SSL handshake.

Currently, OnApp applies the CloudSSL+SNI solution. Users can import custom SNI SSL certificates into the system or request SSL to be enabled for their CDN resource. One SSL certificate can be associated with several CDN resources, but a resource can only be linked to one SSL certificate. Removal of a CDN resource does not affect the status of the SSL certificate associated with the CDN resource. However, some of the older browsers do not support SNI. In this case, users, who prefer browsers that do not support SNI, can purchase an SSL certificate and the SAN solution will be applied. On questions about the SSL certificate purchase, please contact OnApp support.

For the list of browsers that do not support SNI, kindly refer to the Server Name Indication article.

OnApp currently supports the following types of certificates:

- domain-validated (DV) certificate (example.com)
  - single certificate
  - wildcard certificate (*.example.com)
  - SAN certificate (any domains)

- organization validation (OV) certificates
  - single certificate
  - wildcard certificate (*.example.com)
  - SAN certificate (any domains)

- extended validation (EV) certificates
  - single certificate
  - wildcard certificate (*.example.com)
  - SAN certificate (any domains)

- high-assurance certificates

- This feature is available for HTTP Pull and HTTP Push resources only.
- To add custom SNI SSL certificates, the user needs to have the CDN SSL Certificates permissions enabled.
- Custom SNI SSL certificates can be used for secondary hostnames.
- A custom SNI SSL certificate can only be associated with a CDN resource if the certificate and the resource have the same owner. The drop-down list of SSL certificates in the CDN resource creation wizard shows only the certificates of the user who will be the resource owner.
When a custom SNI SSL certificate is associated with a CDN resource, the certificate applies only to the edge servers subscribed to that resource.

3.7.1 View Custom SNI SSL Certificates
To view the list of available SSL certificates:
1. Go to your Control Panel > CDN > SSL Certificates menu.
2. On the page that appears, all available custom SNI SSL certificates with their details are listed:
   - ID - the ID of the custom SNI SSL certificate
   - Name - the name of the certificate. Click the name to view the properties of the certificates and associated CDN resources
   - Actions - click the Actions button to edit or delete the certificate

3.7.2 Add Custom SNI SSL Certificates
To import an SSL certificate:
1. Go to your Control Panel > CDN > SSL Certificates menu.
2. On the page that appears, all available custom SNI SSL certificates are listed. Click the Import SSL Certificate button.

   To add custom SNI SSL certificates, the user needs to have the CDN SSL Certificates permissions enabled.

3. On the following page, fill in the required information:
   - Name - specify a name for the certificate. This parameter is optional
   - Ssl certificate key - fill in the certificate key, it must be in PEM-format
   - Private key - fill in the SSL key provided by your SSL provider

   Make sure that the Ssl certificate key and Private key parameters are filled out in the same way they are generated - with the line breaks.
4. Click the Create SSL Certificate button to import the certificate.
After you add a custom SNI SSL certificate to the cloud, you can associate it with a CDN resource.
To do this:

1. Go to your Control Panel > CDN > Resources menu.
2. On the page that appears, click the Actions button next to the CDN resource which you want to associate with the custom SNI SSL certificate, and select Edit.
3. On the following page, you can associate the custom SNI SSL certificate with the CDN resource (see the Create HTTP Resource section for details).
4. Click Save.

When a custom SNI SSL certificate is associated with a CDN resource, the certificate applies only to the edge servers subscribed to that resource.

3.7.3 Edit Custom SNI SSL Certificates

You can edit your custom SNI SSL certificates, by following this procedure:

1. Go to your Control Panel > CDN > Resources menu.
2. On the page that appears, click the Actions button next to the CDN resource which you want to associate with the custom SNI SSL certificate, and select Edit. Alternatively, click the name of the certificate and click the Edit button on the page that loads.
3. On the following page, edit the details of the certificate:
   - Name - specify a name for the certificate
   - Ssl certificate key - fill in the certificate key, it must be in PEM-format
   - Private key - fill in the SSL key provided by your SSL provider. Private key will not be displayed for security reasons

Make sure that the Ssl certificate key and Private key parameters are filled in the same way they are generated - with the line breaks.
4. Click Save.

3.7.4 Delete Custom SNI SSL Certificates

To delete a custom SNI SSL Certificate:

1. Go to your Control Panel > CDN > SSL Certificates menu.
2. On the page that appears, click the Actions button next to the required certificate and select Delete.

Please note that the removal of an SSL certificate is not allowed if it is associated with a CDN resource.
3.8 Reporting

CDN reporting functionality allows you to conduct the in-depth analysis of your own CDN resources by viewing different reports. At the moment the following reports are available:

- Top Files Report
- Top Referrers Report
- Overview Report
- Cache Statistics Report
- Status Codes Report
- Stream Bandwidth Report
- Visitors Report

You can apply filters for every report (by time period, by CDN resource). Also export to csv format is available.

- Ensure that the CDN reports permissions are on before managing CDN reports. For more information refer to the Default Permissions for User Role section of this guide.
- Ensure that you have an Aflexi ID before managing CDN reporting statistics.

3.8.1 Overview Report

To view the Overview report:

1. Go to your Control Panel > Metrics > CDN menu.
2. Click the Overview section. You can filter the statistics by frequency and by date - select frequency (one minute, one hour or one day) and the time period from the drop-down menu and click the Apply button.
3. You will get the statistics divided into several sections:
   - **Line chart**
     The line chart shows the cached and uncached bandwidth statistics (in MB) for the selected period.
   - **Top 5 CDN resources**
     You can view top five CDN resources together with their details:
     - **Bandwidth** - the amount of transmitted bandwidth for the selected period
     - **Cache Hit** - the amount of successful file requests for the selected period
     - **Miss** - the amount of failed file requests for the selected period
   - **Top 5 Http errors**
     This section shows top five CDN resources with the biggest amount of http errors (4xx&5xx)
   - **Top 5 CDN locations**
     You can view top five CDN locations with the biggest amount of bandwidth.
Visitor Statistic
In this section you can find a diagram, which shows visitor statistics by region.

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

3.8.2 Cache Statistics Report

To view the Cache Statistics report:
1. Go to your Control Panel > Metrics > Cache Statistics menu.
2. You can filter the statistics by:
   - **Type** - select the statistics type (GB, Hit/Miss, Speed) from the drop-down list
   - **CDN resource** - choose the CDN resource for which you want to view the statistics
   - **From** - select period start date
   - **To** - select period end date
3. Click the **Apply** button. You will get the statistics chart and table with locations

   Depending on the selected statistics type, the chart and the table will show the following:

   - **If bandwidth type (GB) is selected**
     The chart shows the cached and uncached bandwidth statistics (in Mb) for the selected period.
     Below you can find the list of corresponding CDN locations with their number of requests and amount of bandwidth.

   - **If Hit/Miss type is selected**
     The chart shows the amount of hit and miss requests for the selected period.
     Below you can find the list of corresponding CDN locations with their total number of requests and amount of hit and miss requests.

   - **If speed type is selected**
     The chart shows the cached and uncached bandwidth speed statistics (in Mbit/s) for the selected period.
     Below you can find the list of corresponding CDN locations with their speed amount.
To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

To export the statistics in *csv* format, click the **Export to CSV** button.

The `$upstream_http_x_cache` status can be either **MISS**, **BYPASS**, **EXPIRED**, **UPDATING**, or **HIT**.

**Click here to see the meaning of the abovementioned statuses**
- **MISS** - the content is not found in cache. It gets the content from the origin, cache it, and returns the content.
- **BYPASS** - no caching. It gets the content straight from the origin.
- **EXPIRED** - cache time limit reach. No content change in the origin.
- **UPDATING** - cache time limit reach. Cache gets new content change in the origin.
- **HIT** - the content is found in cache. It returns the content from the cache.

### 3.8.3 Top Files Report

To view the Top files report:
1. Go to your Control Panel > **Metrics** > **CDN** menu.
2. Click the **Top Files** section. You will get the list of top 50 files (by default for the last week) with the following details:
   - **Resource** - the name of the CDN resource
   - **File URL** - the URL of the resource file
   - **Request** - the total amount of file requests for the selected period
   - **Hit** - the amount of successful file requests for the selected period
   - **Miss** - the amount of failed file requests for the selected period
   - **Bandwidth** - the amount of transmitted bandwidth for the selected period
   - **Actions** - if available, you can click the **Purge** button to remove cache content. This action is available only for accelerated CDN resources

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

You can filter the statistics by date or by CDN resource - select the time period or the CDN resource from the drop-down menu and click the **Apply** button.

To export the statistics in *csv* format, click the **Export to CSV** button.

### 3.8.4 Top Referrers Report

This report is available only for resource owner.
To view the Top referrers report:
1. Go to your Control Panel > Metrics > CDN menu.
2. Click the Top Referrers section. You will get the list of top 50 referrers (by default for the last week) with the following details:
   - Resource - the name of the CDN resource
   - Referrer - the clickable referrer link
   - Hit - the amount of references for the selected period
To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.
You can filter the statistics by date or by CDN resource - select the time period or the CDN resource from the drop-down menu and click the Apply button.
To export the statistics in csv format, click the Export to CSV button.

3.8.5 Status Codes Report
To view the Status Codes report:
1. Go to your Control Panel > Metrics > CDN menu.
2. Click the Status Codes section. You can filter the statistics by:
   - Frequency - select the frequency type (one minute, one hour or one day) from the drop-down menu
   - CDN resource - choose the CDN resource for which you want to view the statistics
   - Time period - select period start and end date
3. Click the Apply button. You will get the statistics chart and two tables:
   - The chart shows the number of requests with different error codes for the selected period.
   - You can view the list of error codes together with their amount of requests in Status Codes table.
   - Http error report table shows the list of CDN resources together with their amount of error requests.

OnApp applies the global setting for HTTP error code caching as follows:
- proxy_cache_valid 404 1m
- proxy_cache_valid 200 301 1d

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.
To export the statistics in csv format, click the Export to CSV button.

If your CDN hostname returns the following status codes:
- **2XX** (200, 201, 202, etc.) means that the HTTP request is successful.
- **3XX** (301, 302, etc.) means that there is a redirection layer being set up on the origin or webserver. To avoid any redirection conflict when using CDN, you can use an IP origin instead of a hostname origin. For
example, instead of using www.mywebsite.com as an origin, use the IP address of www.mywebsite.com as a CDN resource origin.

- **4XX (401, 402, etc.)** means that the CDN PoP (edge servers) has an issue.

  **Click here to see more details**

  Check the following:
  
  o The status of your CDN resource is Active. If it is Preparing, this status code is correct because the configuration of the CDN resource has not been populated to the CDN edge server.
  
  o If you set any country access in the **Access Policy** field when creating or editing for your CDN resource, this status code is correct provided that you are from the prohibited countries.
  
  o With the Content Compare Tool, check which edge server is throwing a 4XX error. If it happened only on a few CDN edge servers but not all, contact **OnApp Support** and remove these locations from our CDN edge group temporary to keep users from receiving a 4XX error.

- **5XX (501, 502, etc.)** means that CDN edge servers have an issue with fetching content from the CDN resource origin.

  **Click here to see more details**

  Check the following:
  
  o The Origin HTTP is accessible.
  
  o The firewall has not blocked the edge server.
  
  o DNS for the origin hostname is working.

### 3.8.6 Concurrent Statistics

The CDN streaming statistics screen shows the concurrent viewers report for your CDN streaming sites.

To view the statistics/graphs for your CDN streaming sites:

1. Go to your Control Panel > **Metrics** > **Concurrent Statistics** menu.
2. Specify the period in the **From** and **To** fields.
3. Select the type of the filter – either by resources or by locations.

4. Click the Apply button.

To zoom into a time period, drag the chart by holding down the left mouse button and moving the mouse. Click the Reset Zoom button to zoom out again.

The concurrent number showed is the average number of a five-minute interval. It is recommended to scale down the graph by limiting the time zone to a smaller time frame to get an accurate result. If the selected time frame is too large, the graph aggregates the data.

3.8.7 Stream Bandwidth Report

Be aware that the bandwidth statistics report shows information on Stream type CDN resources only.

To view the bandwidth statistics report:

1. Go to your Control Panel > Metrics > Stream Bandwidth menu.

2. Select the type of the filter – either GB or MBit/s. In MBPS mode you can get statistics for the last 10 days only. The older statistics is removed. There are no limitations for GB mode.

3. Specify the period in the From and To fields.

4. Select a resource and/or location for which the statistics will be generated.

5. Click the Apply button.

The points in the graph are displayed according to a particular frequency, which depends on the time period specified with the From and To parameters:

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Frequency in seconds</th>
<th>Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 30 days</td>
<td>86400</td>
<td>1 day</td>
</tr>
<tr>
<td>31 - 93 days</td>
<td>604800</td>
<td>1 week</td>
</tr>
<tr>
<td>equal or more than 93 days</td>
<td>1209600</td>
<td>2 weeks</td>
</tr>
</tbody>
</table>

Statistics available in the frequency higher than selected will be accumulated to a single point of such frequency. For example, the statistics was requested for the period of 31-93 days, so the frequency of points in the graph is 7 days. If the statistics was
generated few times during those 7 days (day1+day2+dayₙ), it will be added up and displayed as a single point, with a time stamp marked as the first day of such 7 days.

The statistics are displayed in two graphs: Cached and Non Cached.

To zoom into a time period, click and drag in a chart. Click the Reset Zoom button to zoom out again.

3.8.8 Visitors Report

To view the Visitors report:

1. Go to your Control Panel > Metrics > CDN menu.

2. Click the Visitors section. You can filter the statistics by:
   - CDN resource - choose CDN resource, for which you want to view the statistics
   - Time period - select period start and end date

3. Click the Apply button. You will get the Top 5 Countries chart (by default for the last week) with the visitors amount statistics. Also you will get the Visitor Countries table (sorted by the highest request) with the following details:
   - Visitor Country - the code of the visitor country
   - AVG Latency - the average latency for the selected period
   - AVG Transfer Rate - the average transfer rate for the selected period
   - Requests - the amount of successful file requests for the selected period
   - Bandwidth - the total amount of transmitted bandwidth for the selected period

To sort information by column in ascending or descending order, mouse over the particular column header and click a triangle icon.

To export the statistics in csv format, click the Export to CSV button.

3.8.9 Usage Statistics

The Metrics > CDN > CDN Usage page lists the summary of CDN Resources used by CDN with their details:

- **Owner** – the owner’s user name. Click the owner’s name for details.
- **Edge Group** – the edge group to which the CDN resource belongs to.
- **Location** – CDN edge server’s location.
- **Data cached** – cached CDN traffic in a number_to_human_size format. (See the table below)
- **Data non cached** – non cached CDN traffic in a number_to_human_size format. (See the table below)

The table of formatting the bytes in number into a more understandable representation:
number_to_human_size(123) => 123 Bytes
number_to_human_size(1234) => 1.21 KB
number_to_human_size(12345) => 12.1 KB
number_to_human_size(1234567) => 1.18 MB
number_to_human_size(1234567890) => 1.15 GB
number_to_human_size(123456790123) => 1.12 TB
number_to_human_size(1234567, :precision => 2) => 1.2 MB
number_to_human_size(483989, :precision => 2) => 470 KB
number_to_human_size(1234567, :precision => 2, :separator => ',') => 1,2 MB

By default, statistics are generated for the last 72 hours. To specify another period, set the **Start** and **End** time and click the **Apply** button. Tick the **Show in my Timezone** box to show CDN usage statistics according to your profile’s time zone settings.

### 3.9 CDN Reports

- **Any action on Reports** - the user can take any action on CDN reports
- **See Overview Report** - the user can see Overview Report
- **See Cache Statistics Report** - the user can view Cache Statistics Report
- **See Top Files Report** - the user can view Top Files Report
- **See Top Referrers Report** - the user can view Top Referrers Report
- **See Status Codes Report** - the user can view Status Codes Report
- **See Stream Bandwidth Report** - the user can see Stream Bandwidth Report
- **See Visitors Report** - the user can see Visitors Report

For details, refer to the **CDN Reporting** section.

### CDN Resources

- **Create a new CDN resource** - the user can create a new CDN resource
- **Destroy own CDN resources** - the user can only delete their own CDN resources
- **See own CDN resources** - the user can only see their own CDN resources
- **Update own CDN resources** - the user can only edit their own CDN resources

For details, refer to the **CDN Resources** section.

### CDN SSL Certificates

- **Create a new CDN SSL Certificate** - the user can create new CDN SSL certificates
- **Destroy own CDN SSL Certificate** - the user can only delete their own CDN SSL certificates
- **See own CDN SSL Certificates** - the user can only see their own CDN SSL certificates
- **Update own CDN SSL Certificates** - the user can only edit their own CDN SSL certificates

For details, refer to the **SSL Certificates** section.

### CDN Usage Statistics

- **User can see own CDN usage statistics** - the user can only see own CDN usage statistics

For details, refer to the **Usage Statistics** section.
DNS Zones

- *Create a new DNS zone* - the user can create a new DNS zone
- *Destroy own DNS zone* - the user can only delete their own DNS zones
- *See own DNS zones* - the user can only see their own DNS zones
- *Create a new DNS record* - the user can create a new DNS record
- *See own DNS records* - the user can see all DNS records
- *Update own DNS records* - the user can update own DNS records

For details, refer to the DNS section.

### 3.10 DNS

The full version of OnApp Cloud (with CDN enabled) now gives you access to our free Anycast DNS service. Instead of managing your own DNS servers you can use our fully redundant global DNS, hosted at multiple datacenters around the world, and manage it through your OnApp Control Panel.

You can use our DNS service with domains registered anywhere on a third party domain registrar. The Control Panel lets you set up hostnames, manage DNS records, aliases, Mail Exchange, TXT and SRV records.

Using DNS has two main steps: setting up DNS hostnames, and DNS Zones.

#### 3.10.1 DNS Setup

DNS setup allows you to create a DNS hostname. After you create a DNS hostname, you get access to creating and managing DNS zones. To set up a new DNS, make sure that:

- CDN is enabled
- You have dns_zone or dns_zone.setup permission to access this page.

You can create only one DNS domain. Once created, a DNS domain can't be deleted, only updated.

If an administrator uses the same license for two different Control Panels, they can use the same DNS domain for CP1 and CP2. To do this, the administrator should set the same DNS domain settings for both Control Panels.

To add a DNS domain:

1. Go to your Control Panel's **Settings** menu.
2. Click the **DNS Setup** button.
3. On the screen that appears, type your fully qualified domain name. Mind that you won't be able to use a domain name that is already registered with OnApp DNS.

   The domains of the following kind are forbidden:
   
   - google
   - microsoft
   - domain.com
   - onapp.com
4. Click the Save button.

After the DNS domain is added, the DNS service will be available to users. If you have added a domain, but still face some issues or the "Unable to get DNS Zone Setup: CDN service is temporarily unavailable" error occurs, contact support.

Make sure your domain name registrar has designated your domain to the appropriate glue records displayed in the infobox at Control Panel > Settings > DNS Setup.

3.10.1.1 Edit DNS Domain
To edit your DNS domain:

1. Go to your Control Panel > Admin > Settings menu, and click the DNS Setup icon.
2. Type a new fully qualified domain name to replace your existing domain
3. Click the Save button to save changes.

If DNS domain is updated, all NS records for all DNS zones under this user will be updated.

3.10.2 DNS Zones
OnApp DNS Zone feature allows you to manage your and your clients’ domain DNS. Each time DNS zone, record, or setup settings are refreshed, the DNS configuration is immediately updated on the DNS vendor server.

3.10.2.1 Limits
For DNS zones, there is a limit of 10,000 per OnApp client.

As for records in the DNS zones, we do not impose a limit on them as long as the records are unique.

3.10.2.2 Dynamic DNS
Using dynamic DNS Services (DDNS) to modify zone records is not always possible. DNS records are created on your OnApp Cloud server and can be edited later. However, if your router does not support dynamic DNS Services updates, it won't be possible to edit zone records via the standard DDNS client's interface. In this case, use API calls to edit zone records.

3.10.2.3 Check DNS Usage Statistics
At this time, it is not possible to check the usage statistics of your domains using the OnApp anycast DNS.

3.10.2.4 Reserved DNS Zone

Click to see the domains that are reserved and cannot be added as a DNS zone in OnApp anycast DNS:

- amazon.com
- domain.com
- ebay.com
- facebook.com
We currently have 13 anycast DNS locations.

Click here to see the list
1. Stockholm, Sweden
2. Göteborg, Sweden
3. Amsterdam, Netherlands
4. Hong Kong
5. London, England
6. Singapore
7. Bucharest, Romania
8. Frankfurt, Germany
10. Los Angeles, United States
11. Washington, United States
12. Miami, United States
13. San Jose, United States

To add a new DNS zone:
1. Go to your Control Panel > Cloud > DNS menu.
2. Click the Create DNS Zone button.
3. Fill in your domain name. At the domain registrar, point your domain to the following name servers:
   - ns1.yourdomain.com
   - ns2.yourdomain.com
   - ns3.yourdomain.com
   - ns4.yourdomain.com
Where yourdomain.com is your fully qualified domain name, which you have specified at DNS setup. You cannot use Unicode characters for a domain name.

Starting with OnApp 5.3, you can create rDNS zone. Reverse DNS resolution (rDNS) is the determination of a domain name associated with an IP address via querying DNS. Most of mail servers make rDNS lookup before accepting messages originating from mail server as one of the anti spam email technique.

To create rDNS zone, at the domain registrar, point your domain to the following name servers:

- 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa
- 0-25.228.169.69.in-addr.arpa
- 64-30.228.169.69.in-addr.arpa
- 228.169.69.in-addr.arpa

4. Tick the Auto Populate With Existing DNS Record box to automatically import your existing DNS settings, or skip this step to start from scratch. Note that this option may not import all existing settings, so you should check your new record for any missing entries.

5. Click the Submit button.

6. On the page that appears, click the Add icons next to the DNS records you want to add. You can add and manage the following DNS records:

   - **SOA (Start of Authority)** – change the start of authority time to live value (TTL). To change a SOA TTL, click the TTL cell next to the SOA record and type a new value.

   - **NS (Name Server)** – change the TTL of the existing name servers or add new name servers. To change a name server's TTL, click the TTL cell next to the name server record and type a new value.

     It is not allowed to add an NS record for the main DNS zone (‘@’ in a "Name" field). An NS record is used to point a subdomain to a set of your nameservers (subdomain delegation).

   - **A (Host)** – point your domain name to a static IP address. To create a new A record, enter the following parameters into the cells:
     
     * Host – enter a host name or use the "@" sign to represent your current host.
     * Point to – enter the IP address to which the user would be sent for this host name.
     * TTL – set the time to live value for this record.

     A record example: ftp 192.168.0.1 86400
     Where: ftp is the host; 192.168.0.1 - IP , 86400 is TTL value.
     So your ftp.yourdomain.com will resolve to 192.168.0.1 IP address and the TTL value = 86400 seconds.

   - **AAAA (Host)**
     To create a new AAAA record, fill in the following cells:
     
     * Host – enter a host name or use the "@" sign to represent your current host.
     * Point to – enter the IP address to which the user would be sent for this host name.
     * TTL – set the TTL value for this record.

     AAAA record example: ftp 2a00:1450:400b:c00::68 86400
     Where: ftp is the host, 2a00:1450:400b:c00 – IPv6 address, 86400 is the TTL value.
     So your ftp.yourdomain.com will resolve to 2a00:1450:400b:c00 IPv6 address.
- **CNAME** (Alias) – alias domain records to your domain.

  It is possible to use underscore character in the CNAME records.

  To add a CNAME record, fill in the following cells:
  
  * Host – enter the host name or use the "@" sign to represent your current host.
  * Point to – enter an alias you want to assign to your domain.
  * TTL – set the TTL value.

  CNAME record example:
  
  `www.example.com 86400`

  Where: www is an alias, example.com is a valid domain name, 86400 is TTL value.

  To add the aliases, make sure an A record is added to this domain.

- **PTR** (Pointer) - create PTR records for rDNS zone.

  To add a PTR record, fill in the following cells:
  
  * Name – enter the IP:
    
    - for IPv4
    
    One of the IP from the range, such as 1 or 2
    
    - for IPv6
    
    One of the IP from range, such as `0.0.0.0.0.0.8.b.d.0.1.0.0.2`
  
  * Hostname – enter the host name or use the "@" sign to represent your current host.
  
  * TTL – set the TTL value.

  PTR record example:
  
  `0.0.0.1 @example.com 86400`

  Where: 0.0.0.1 is an IP, example.com is a valid domain name, 86400 is TTL value.

- **MX** (Mail Exchange) - identify the mail server for your domain name.

  To add a MX record, fill in the following cells:
  
  * Priority – set the MX priority to specify the routing order (lower value means higher priority).

  * Host: enter the hostname to which the emails should go

  * Goes to – enter the valid domain name.

  * TTL – set the TTL value.

  MX record example:
  
  `10 mail example.com 86400`

  Where: 10 is priority, mail is the host, example.com is a domain, 86400 is TTL.

- **TXT** – add additional information about the DNS zone.

  * Host – enter the valid host name
  
  * Value – any free text you want within a TXT record.

  * TTL – TTL value.

  TXT record example:
  
  `@ v=spf1 a mx ptr ip4:192.168.1.1 ~all 86400`
Where: @ is the host name, v=spf1 a mx ptr ip4:192.168.1.1 ~all is value, 86400 is TTL.

- **SRV** (Service) – specify services that you have on your domain.

To add a SRV record, enter the following cells parameters:

- **Host** – type the host for which this record is valid.
- **Priority** – set the host priority. Lower value means more preferred.
- **Weight** – the approximate weight for relative records with the same priority.
- **Port** – the port on which the service can be found.
- **Points to** – enter the domain name.
- **TTL** – set the time to live value.

SRV record example:

xmpp._tcp 0 1 5222 jabber.example.com 86400

Where: xmpp._tcp is a host, 0 is priority, 1 is weight, 5222 is port, jabber.example.com is a points to value, 86400 is TTL.

### 3.10.2.6.1 Wildcards

In the OnApp version 3.0 it is possible to use wildcards in all DNS records. The table below explains the rules of wildcard use.

<table>
<thead>
<tr>
<th>DNS record type</th>
<th>Allowed</th>
<th>Disallowed</th>
</tr>
</thead>
</table>
| A               | *.example.com  
|                 | **.example.com  
|                 | *.abc.example.com | abc.*example.com  
|                 | Note: multiple ‘*’ will be changed to single ‘*’. E.g. ****.example.com will be changed to *.example.com  
|                 | Can only be prefixed for domain. | |
| AAAA            | Same to A record | Same to A record |
| MX              | Same to A record | Same to A record |
| CNAME           | Same to A record  
|                 | Note: CNAME wildcard record cannot coexist with A record. | Same to A record |
| TXT             | *.example.com  
|                 | **.example.com  
|                 | sub.*.example.com  
|                 | sub.*.*.example.com  
|                 | *sub*.*.example.com | |
|                 | Note: Wildcards are valid in any position, as long as the domain remains DNS zone's subdomain. | |

It is not possible to use wildcards for NS, SOA and SRV records because of PowerDNS limitations.
3.10.2.6.2 Underscore characters
In the OnApp version 3.0 it is possible to use wildcards in all DNS records. The table below explains the rules of underscore use.

<table>
<thead>
<tr>
<th>DNS Record type</th>
<th>Allowed</th>
<th>Disallowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>_abc.example.com</td>
<td>abc._example.com</td>
</tr>
<tr>
<td></td>
<td>__abc.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>abc_abc</em>.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Multiple '<em>' e.g. '</em>___' will not be changed to a single underscore, unless stated.</td>
<td></td>
</tr>
<tr>
<td>AAAA</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td>MX</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td>CNAME</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td></td>
<td>Note: CNAME record with underscore cannot coexist with A record.</td>
<td></td>
</tr>
<tr>
<td>NS</td>
<td>Same to A record</td>
<td>Same to A record</td>
</tr>
<tr>
<td>SRV</td>
<td>_xmpp._tcp.example.com</td>
<td>All except examples in the Allowed column.</td>
</tr>
<tr>
<td></td>
<td>__xmpp.__tcp.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>_xmpp.<em>tcp</em>.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>_xmpp.<em>tcp</em>.abc.example.com</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Multiple '_' e.g. '_____‘ will be changed to a single underscore character.</td>
<td></td>
</tr>
<tr>
<td>TXT</td>
<td>All except examples in the Disallowed column.</td>
<td>_example.com</td>
</tr>
<tr>
<td></td>
<td>Note: underscores are valid in any position, as long as the domain remains DNS zone's subdomain.</td>
<td>abc._example.com</td>
</tr>
<tr>
<td></td>
<td></td>
<td>example.com_</td>
</tr>
<tr>
<td></td>
<td></td>
<td>example_.com</td>
</tr>
</tbody>
</table>

3.10.2.7 Edit DNS Zone
When you edit a DNS Zone, you manage the records assigned to this zone. The changes are instantly updated on DNS.

To edit a DNS zone:
1. Go to your Control Panel > Cloud > DNS menu. On the screen that appears, the list of DNS zones will be displayed.
2. Click the Actions button next to the domain zone you want to change, then choose Edit.
3. On the screen that appears, edit the DNS records as required:
   - SOA (Start of Authority) – change the start of authority TTL.
   - NS (Name Server) – change the TTL of the existing name servers or add a new name server.

It is not allowed to add an NS record for the main DNS zone (‘@’ in a "Name" field). An NS record is used to point a
subdomain to a set of your nameservers (subdomain delegation).

- **A(Host)** – change the A host record properties:
  - In the Host text box, type the name for a host.
  - In the Point to text box, type the IP address for the new host.
  - Set the TTL value.

- **AAAA (Host)** – change the AAAA record properties as described for the A record.

- **CNAME (Alias)** – canonical name properties.

- **PTR (Pointer)** – change the PTR record properties.

- **MX (Mail Exchange)** – change the mail server properties for your domain name.

- **SRV (Service)** – specify services that you have on your domain.

### 3.10.2.8 Delete DNS Zone

**To delete a domain zone:**

1. Go to your Control Panel > **Cloud** > **DNS** menu.
2. Click the **Actions** button next to the domain zone you want to delete, then click **Delete**.
3. Click **OK** to confirm the deletion.

### 3.10.2.9 User DNS Zones

User DNS zones tab allows you to manage your clients’ DNS zones. Use the **Actions** button next to the required user DNS zone to edit/delete it. The changes will be instantly updated on our DNS.

### 3.10.2.10 Set End-User Access to DNS Service

**To set end-users’ access to DNS service:**

1. Go to your Control Panel > **Admin** > **Roles** menu.
2. Click the **Actions** button next to the required user, then click **Edit**.
3. On the screen that follows, choose the **DNS Zone** group in the **Groups** field.
4. The following list of DNS permissions will appear:
   - **Any action on DNS zone**
   - **Create a new DNS zone**
   - **Destroy any DNS zone**
   - **Destroy own DNS zone**
   - **See all DNS zones**
   - **See own DNS zones**
   - **Any action on DNS record**
   - **Create a new DNS record**
   - **Destroy any DNS record**
   - **See all DNS records**
   - **Update any DNS record**
   - **DNS Setup**
5. Tick the required boxes.
6. Click the **Save** button.

3.10.2.11 View/Edit/Delete User DNS Zones

To view, edit and delete existing clients' DNS zones:

1. Go to your Control Panel > **Cloud** > **DNS** menu.

2. Click the **User DNS Roles** tab. On the screen that appears, you'll see a list of all clients' DNS zones.

3. To edit a clients' DNS zone, click the **Actions** button next to it, then click **Edit**. On the screen that appears, edit its details and click the **Save** button.

4. To delete a clients' DNS zone, click the **Actions** button next to the DNS zone you want to delete, then click **Delete**. You'll be asked to confirm deletion.
4 API Guide

The API enables cloud integration with third party applications. This guide is a complete reference for all API calls and includes detailed API information, code and output examples. The version of the guide corresponds to the latest OnApp API version. For comprehensive instructions on previous versions, refer to corresponding guides at docs.onapp.com.

- The OnApp API is RESTful
- All function calls respond to XML and JSON exchange formats
- All function calls need authorization and authentication (Basic HTTP or API key)
- The OnApp API is backward compatible within one major version. However, a new major version might include changes that are not backward compatible with the previous one.

4.1 API Authentication

To authenticate using HTTP Basic, just use your username/password combination. Curl example:

```
curl -u user:userpass
```

To authenticate using API key, put your account email as a login and the key to the server as a password.

4.2 HTTP Methods

The API uses the following HTTP methods:

- **GET** - used for retrieving information from a particular URI
- **POST** - used for creating new object and adding new transactions into the queue
- **PUT** - used for altering object properties

**NOTE:** updated_at value is changed in PUT requests even if the request fails.

- **DELETE** - used for object deletion

4.3 HTTP Response Codes

The API returns appropriate HTTP status codes for every request:

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 OK</td>
<td>The request completed successfully</td>
</tr>
<tr>
<td>204 No content</td>
<td>The request completed successfully. The 204 status is returned on DELETE and PUT requests</td>
</tr>
<tr>
<td>201 Scheduled</td>
<td>The request has been accepted and scheduled for processing</td>
</tr>
</tbody>
</table>
4.4 Formatting and Naming Conventions

The table below represents all the existing formatting and naming conventions used in this guide:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>user:userpass</td>
<td>stands for username:password combination</td>
<td>Admin:123456</td>
</tr>
<tr>
<td>onapp.test</td>
<td>stands for address, where your Control Panel is located</td>
<td>Example.com</td>
</tr>
<tr>
<td>:id</td>
<td>stands for the resource ID. Sometimes also: :resource_id</td>
<td>23</td>
</tr>
<tr>
<td>* (asterisk)</td>
<td>marks the required parameters</td>
<td>label *</td>
</tr>
<tr>
<td>preformatted</td>
<td>indicates request examples in XML or JSON</td>
<td>GET /roles.xml</td>
</tr>
<tr>
<td><strong>Code block</strong></td>
<td>indicates console requests and response examples.</td>
<td></td>
</tr>
<tr>
<td>info</td>
<td>An info message emphasizes or explains the information within the chapter.</td>
<td>Clicking the OFF button performs graceful shutdown and then powers off the VS.</td>
</tr>
<tr>
<td>note</td>
<td>A note message contains information essential for the task completion.</td>
<td>The maximum length of a Mount Point is 256 characters.</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>warning</td>
<td>A warning message informs you of something you should not do or be cautious.</td>
<td>You won't be able to restore a VS after deleting it.</td>
</tr>
<tr>
<td>🧠</td>
<td>The element showing new parameters added in the latest release of API.</td>
<td>limit_type – hourly or monthly limit type set for the resource</td>
</tr>
</tbody>
</table>

### 4.5 FAQ

**Q: Is it possible to enable API access via https?**

A: We can enable https for your cloud, which can be used for both WebUI access and API access. Or you can do so yourself: the Apache config file is located at: /etc/httpd/conf.d/onapp.conf

**Q: Can you create a VS on behalf of another user?**

A: No. It is possible to switch VS owners, however. Refer to Change VS owner section for details.

**Q: How are passwords stored – in plain text?**

A: No, passwords are not stored in plain text. Except for a login and password combination, you can use email + API key combination to authorize a user via the API. API keys can be generated and changed easily on a user's profile page (as well as through the API). For security reasons we recommend users authenticate through the API key, not the login and password.

**Q: Which parameters are required, and which are optional?**

A: Required parameters are marked in this guide with an asterisk *.

### 4.6 CDN API

The OnApp CDN API guide describes API requests you can use to manage your CDN related resources.

Refer to one of the following sections for details:

- [CDN Accelerator](#)
- [CDN Edge Groups](#)
- [CDN Edge Servers](#)
- [CDN HTTP Caching Rules API](#)
- [CDN Reporting](#)
- [CDN Resources](#)
- [CDN SSL Certificates](#)
- [CDN Storage Servers](#)
• CDN Usage Statistics
• CDN Locations API
• DNS Zones
• DNS Setup

For information on how to manage CDN related resources on OnApp UI refer to the Administration or the User guide.

4.7 Buckets API

Buckets define the resource allocation and prices for resources in the cloud. They are made up of two parts:

• Access Control which defines the resources the user under the bucket has access to
• Rate Card which includes the prices for resource usage

This section contains the API requests you can use to manage buckets for your CDN resources.

4.7.1 Get List of Buckets

To get the list of buckets, use the following request:

GET /billing/buckets.xml
GET /billing/buckets.json

XML Request Example


JSON Request Example


XML Output Example
<buckets type="array">
  <bucket>
    <id type="integer">3</id>
    <label>ut</label>
    <created_at type="dateTime">2017-06-26T08:48:05+00:00</created_at>
    <updated_at type="dateTime">2017-06-26T08:48:05+00:00</updated_at>
    <currency_code>USD</currency_code>
    <show_price nil="true"/>
    <monthly_price type="decimal">1.0</monthly_price>
    <allows_mak type="boolean">true</allows_mak>
    <allows_kms type="boolean">true</allows_kms>
    <allows_own type="boolean">true</allows_own>
    <associated_with_users type="integer">1</associated_with_users>
  </bucket>
</buckets>

Where:
id - the bucket ID
label - the bucket name
created_at - the date in the [YYYY][MM][DD][hh][mm][ss]Z format
updated_at - the date when the bucket was updated in the [YYYY][MM][DD][hh][mm][ss]Z format
currency_code - the currency in which the users are charged
show_price - true, if users can see the prices set up for them, otherwise false
monthly_price - the monthly fee for bucket usage
allows_mak - true, if the MAK licensing is allowed, otherwise false
allows_kms - true, if the KMS licensing is allowed for this bucket, otherwise false
allows_own - true, if adding own licenses is allowed for this bucket, otherwise false
associated_with_users - the number of users with which this bucket is associated

4.7.2 Get Bucket Details
To get bucket details, use the following request:

GET /billing/buckets/id.xml
GET /billing/buckets/id.json

XML Request Example


JSON Request Example
-H 'Accept: application/json' -H 'Content-type: application/json'

XML Output Example

```xml
<buckets type="array">
  <bucket>
    <id type="integer">21</id>
    <label>minima</label>
    <created_at type="dateTime">2017-06-26T08:48:09+00:00</created_at>
    <updated_at type="dateTime">2017-06-26T08:48:09+00:00</updated_at>
    <currency_code>USD</currency_code>
    <show_price nil="true"/>
    <monthly_price type="decimal">1.0</monthly_price>
    <allows_mak type="boolean">true</allows_mak>
    <allows_kms type="boolean">true</allows_kms>
    <allows_own type="boolean">true</allows_own>
    <type>Billing::Buckets::Plan</type>
    <associated_with_users type="integer">1</associated_with_users>
  </bucket>
</buckets>
```

Where:

- **id** - the bucket ID
- **label** - the bucket name
- **created_at** - the date in the [YYYY][MM][DD][hh][mm][ss]Z format
- **updated_at** - the date when the bucket was updated in the [YYYY][MM][DD][hh][mm][ss]Z format
- **currency_code** - the currency in which the users are charged
- **show_price** - true, if users can see the prices set up for them, otherwise false
- **monthly_price** - the monthly fee for bucket usage
- **allows_mak** - true, if the MAK licensing is allowed, otherwise, false
- **allows_kms** - true, if the KMS licensing is allowed for this bucket, otherwise, false
- **allows_own** - true, if adding own licenses is allowed for this bucket, otherwise, false
- **type** - the type of bucket
- **associated_with_users** - the number of users with which this bucket is associated

### 4.7.3 Add Bucket

To create a new bucket, use the following request:

```bash
POST /billing/buckets.xml
POST /billing/buckets.json
```

XML Request Example
OnApp Cloud 6.5 Edge 5 CDN Guide

**JSON Request Example**

```bash
```

Where:
- **label** - the bucket name
- **currency_code** - the currency that users will be charged in within this bucket (USD by default)
- **monthly_price** - set the monthly fee for bucket usage
- **allows_kms** - true, if the KMS licensing is allowed for this bucket, otherwise, false
- **allows_mak** - true, if the MAK licensing is allowed, otherwise, false
- **allows_own** - true, if adding own licenses is allowed for this bucket, otherwise, false

**4.7.4 Edit Bucket**

To edit a bucket, use the following request:

PUT /billing/buckets/14.xml

PUT /billing/buckets/13.json

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

Where:
- **label** - the bucket name
- **monthly_price** - set the monthly fee for bucket usage
4.7.5 Clone Bucket
To clone a bucket with its prices and added resources, use the following request:

POST /billing/buckets/:bucket_id/clone.xml
POST /billing/buckets/:bucket_id/clone.json

XML Request Example

curl -i -X POST -u 'user:userpass' --url
- H 'Content-Type: application/xml'

JSON Request Example

curl -i -X POST -u 'user:userpass' --url
- H 'Content-Type: application/json'

Where you indicate in the URL the ID of the cloned bucket.

4.7.6 Delete Bucket
To delete a bucket, use the following request:

DELETE /billing/buckets/:id.xml
DELETE /billing/buckets/:id.json

XML Request Example

curl -i -X DELETE http://onapp.test/billing/buckets/9.xml
-H 'Accept: application/xml'
- H 'Content-Type: application/xml'
- u user:userpass

JSON Request Example

curl -i -X DELETE http://onapp.test/billing/buckets/9.json
-H 'Accept: application/json'
- H 'Content-Type: application/json'
- u user:userpass

4.7.7 Access Control
The Access Control is the part of the bucket which defines the resources to which a user under the bucket has access. In the Access Control, you define the access to CDN resources.

4.7.7.1 Get List of Access Controls for Other Server Type
To get the list of access controls, use the following request:

GET /billing/buckets/:bucket_id/access_controls.xml
GET /billing/buckets/:bucket_id/access_controls.json

XML Request Example
4.7.7.2 Add Access Control for Other Server Type

To create a new access control, use the following request:

```
POST /billing/buckets/:bucket_id/access_controls.xml
```

**Where:**

- **bucket_id** - the ID of the bucket with which this access control is associated
- **server_type** - the server type this access control is applicable to, in this case, **other**
- **target_id** - the ID of the resource for which configuration is set
- **type** - the type of the resource for which configuration is set. For CDN resources, it can be one of the following values:
  - **edge_groups_resource**
  - **cdn_bandwidth_resource**
- **timing_strategy** - the type of billing for each resource: **hourly** or **monthly** (on peak usage)
- **target_name** - the label of the resource added to the access control
- **limits** - the array of limits for the resource. For CDN resources, it includes the **cdn_bandwidth_resource** limit allocated in GB per month.
POST /billing/buckets/:bucket_id/access_controls.json

XML Request Example

```
curl -i -X POST http://onapp.test/billing/buckets/331/access_controls.xml
-u user:userpass -H 'Accept: application/xml' -H 'Content-type: application/xml' -d
'<?xml version="1.0" encoding="UTF-8"?>
<access_control><bucket_id>331</bucket_id><server_type>other</server_type>
<target_id>2</target_id><create_rate_card>false</create_rate_card><type>edge_groups_resource</type><target_name>EdgeGroup1</target_name></access_control>'
```

JSON Request Example

```
curl -i -X POST http://onapp.test/billing/buckets/331/access_controls.json
-u user:userpass -H 'Accept: application/json' -H 'Content-type: application/json' -d
'{"access_control": {"bucket_id": 331, "server_type": "other", "target_id": 2, "create_rate_card": "false", "type": "edge_groups_resource", "target_name": "EdgeGroup1"}}'
```

Where:

- **bucket_id**: the ID of the bucket with which this access control is associated
- **server_type**: the server type this access control is applicable to, in this case, other
- **target_id**: the ID of the resource added to the access control
- **create_rate_card**: set true if you want to add this resource not only to the access control, but to the rate card as well, otherwise, set false
- **type**: the type of the resource for which configuration is set. For CDN resources, it can be one of the following values:
  - **edge_groups_resource**
  - **cdn_bandwidth_resource**
- **target_name**: the name of the resource that is added to the access control. For CDN resources, it can be a label of an edge group.
- **limits**: the array of limits for the resource. For CDN resources, it includes the cdn_bandwidth_resource limit allocated in GB per month.

If no CDN edge groups are added to Access Control, users under the bucket will have access to none of the edge groups available on the system.

4.7.7.3 Edit Access Control for Other Server Type

To edit access control, use the following request:

PUT /billing/buckets/:bucket_id/access_controls.xml
PUT /billing/buckets/:bucket_id/access_controls.json

XML Request Example
"<access_control><bucket_id>331</bucket_id><server_type>other</server_type><target_id>2</target_id><create_rate_card>false</create_rate_card><type>edge_groups_resource</type><target_name>EdgeGroup1</target_name></access_control>"

JSON Request Example

{
"access_control": {
"bucket_id": 331,
"server_type": "other",
"target_id": 2,
"type": "edge_groups_resource",
"target_name": "EdgeGroup1"
}
}"

Where:

- **bucket_id** - the ID of the bucket with which this access control is associated
- **server_type** - the server type this access control is applicable to, in this case, **other**
- **target_id** - the ID of the resource added to the access control
- **type** - the type of the resource for which configuration is set. For CDN resources, it can be one of the following values:
  - **edge_groups_resource**
  - **cdn_bandwidth_resource**
- **target_name** - the name of the resource that is added to the access control. For CDN resources, it can be a label of an edge group.
- **limits** - the array of limits for the resource. For CDN resources, it includes the **cdn_bandwidth_resource** limit allocated in GB per month.

4.7.7.4 Delete Resource from Access Control for Other Server Type

To delete a resource from access controls, use the following request:

DELETE /billing/buckets/:bucket_id/access_controls.xml
DELETE /billing/buckets/:bucket_id/access_controls.json

XML Request Example

```
curl -i -X DELETE
"<access_control><type>edge_groups_resource</type><bucket_id>331</bucket_id><server_type>other</server_type><target_id>2</target_id></access_control>"
```

JSON Request Example
curl -i -X DELETE
http://onapp.test/billing/buckets/331/access_controls/delete.json -u
user:userpass -H 'Accept: application/json' -H 'Content-type:
application/json' -d '{"access_control": 
  "type": "edge_groups_resource",
  "bucket_id": 331,
  "server_type": "other",
  "target_id": 2}'}

Where:

- **type** - the type of the resource to be deleted from the access control. For CDN resources, it can be one of the following values:
  - `edge_groups_resource`
  - `cdn_bandwidth_resource`

- **bucket_id** - the ID of the bucket with which this access control is associated

- **server_type** - the server type this access control is applicable to, in this case, `other`

- **target_id** - the ID of the resource to be deleted

### 4.7.8 Rate Card

Rate Cards are the part of buckets that contain the free limits for resources and prices for resource usage. This section contains the API requests which you can use to manage Rate Cards for CDN resources.

#### 4.7.8.1 Get List of Rate Cards for Other Server Type

To get the list of rate cards, use the following request:

GET /billing/buckets/:bucket_id/rate_cards.xml

GET /billing/buckets/:bucket_id/rate_cards.json

**XML Request Example**

curl -i -X GET http://onapp.test/billing/buckets/5263/rate_cards.xml -u
user:userpass -H 'Accept: application/xml' -H 'Content-type:
application/xml'

**JSON Request Example**

curl -i -X GET http://onapp.test/billing/buckets/5263/rate_cards.json -u
user:userpass -H 'Accept: application/json' -H 'Content-type:
application/json'

**XML Output Example**
<rate_cards type="array">
  <rate_card>
    <bucket_id type="integer">5263</bucket_id>
    <server_type>other</server_type>
    <target_id type="integer">2</target_id>
    <type>edge_groups_resource</type>
    <timing_strategy>Hourly</timing_strategy>
    <target_name>qaOHegF</target_name>
    <prices>
      <price type="decimal">0.0</price>
    </prices>
  </rate_card>
  ...
  <rate_card>
    ...
  </rate_card>
</rate_cards>

Where:

bucket_id - the ID of the bucket with which this rate card is associated

server_type - the server type this rate card is applicable to, in this case, other

target_id - the ID of the resource for which the prices are set

type - the type of the resource for which configuration is set. For CDN, the value can be edge_groups_resource.

timing_strategy - the type of billing for each resource: hourly or monthly

target_name - the name of the resource that was added to the bucket. For example, this can be a label of an edge group.

price - the price for the resource

4.7.8.2 Add Rate Cards for Other Server Type

To add rate cards, use the following request:

POST /billing/buckets/:bucket_id/rate_cards.xml

POST /billing/buckets/:bucket_id/rate_cards.json

XML Request Example


JSON Request Example

4.7.8.3 Edit Rate Cards for Other Server Type

To edit rate cards, use the following request:

PUT /billing/buckets/:bucket_id/rate_cards.xml
PUT /billing/buckets/:bucket_id/rate_cards.json

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

Where:

- `bucket_id` - the ID of the bucket with which this rate card is associated
- `server_type` - the server type this rate card is applicable to, in this case, *other*
- `target_id` - the ID of the resource for which the prices are set
- `type` - the type of the resource that is added to the rate card. For CDN, it can be *edge_groups_resource*.
- `prices` - the array of prices for the resources. For CDN, it includes the price per GB of bandwidth in the edge group.

4.7.8.4 Delete Resource from Rate Card for Other Server Type

To delete resources from rate card, use the following request:

DELETE /billing/buckets/:bucket_id/rate_cards.xml
DELETE /billing/buckets/:bucket_id/rate_cards.json

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```
OnApp Cloud 6.5 Edge 5 CDN Guide

4.8 CDN Accelerator API

This chapter provides requests for accelerators.

4.8.1 Get List of Accelerators

To view all accelerators in the cloud with their details, use the following request:

GET /accelerators.xml
GET /accelerators.json

XML Request Example

curl -i -X GET -u user:userpass http://onapp.test/accelerators.xml

JSON Request Example

curl -i -X GET -u user:userpass http://onapp.test/accelerators.json

XML Output Example

When removing an edge group from Rate Card, note that the prices will be set to zero for all CDN resources using this edge group.
Explanation of the data returned:

admin_note – an optional reminder for this accelerator created by an administrator
allowed_hot_migrate – true if hot migration is allowed; otherwise false
allowed_swap – true if swap is allowed; otherwise false
booted – true if the accelerator is booted; otherwise false
built - true if the accelerator is built; otherwise false
cores_per_socket - the number of cores per socket for accelerator
cpu_shares – the CPU priority percentage
cpu_sockets - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
cpu_threads - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user bucket
cpus – number of CPU cores allocated to this accelerator
created_at – the date when the accelerator was created in the [YYYY][MM][DD]T[hh][mm][ss]Z format
customer_network_id - ID of a customer network
deleted_at - time when the accelerator was deleted
enable_autoscale – false; not available for accelerators
enable_monitis - deprecated attribute; will be removed in upcoming release
firewall_notrack - true if the NOTRACK rule is set in iptables
hot_add_cpu - false; not available for accelerators
hot_add_memory - false; not available for accelerators
hypervisor_id – the ID of the compute resource, on which the accelerator is deployed
id – the accelerator ID in OnApp CP database
identifier – the accelerator identifier

initial_root_password – the accelerator root password

initial_root_password_encrypted – true, if the accelerator root password is encrypted, otherwise false

instance_package_id - false; not available for accelerators

iso_id - false; not available for accelerators

label – an arbitrary name of the accelerator

local_remote_access_ip_address - the IP address used for console access

local_remote_access_port – the port ID used for console access

locked – true if locked; otherwise false

memory – the amount of RAM resources allocated to this accelerator

min_disk_size – minimum disk space required by the template

note - an optional reminder for this accelerator made by a user account

operating_system – type of operating system

operating_system_distro – the distribution of the operating system

preferred_hvs - the array of preferable compute resources based on compute zone that meet some accelerator configuration settings

recovery_mode – true if the accelerator is booted in the recovery mode; otherwise false

remote_access_password – the password for remote access

service_password - service account password

state – deprecated attribute; will be removed in upcoming release

strict_virtual_machine_id - the ID of a virtual server (or edge server) that will never reside on the same compute resource with this accelerator

suspended – true if suspended; otherwise false

template_id – the ID of the template, on which the accelerator is based

template_label – label of the template on which the accelerator is based

time_zone - the time zone set for the accelerator

updated_at – the date when the accelerator was updated in the [YYYY][MM][DD][hh][mm][ss]Z format

user_id – the ID of the user, who is the accelerator owner

vip – true if the accelerator has VIP status for migration; otherwise false

xen_id – the accelerator ID set by the virtualization engine

ip_addresses – an array of assigned IP addresses with their details assigned to this accelerator:

- address – IP address
- broadcast – a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.
- created_at – time when the IP address was created in the [YYYY][MM][DD][hh][mm][ss]Z format
- customer_network_id - customer network ID
- disallowed_primary – true if not allowed to be used as primary (for baremetal server), otherwise false
- gateway - gateway address
OnApp Cloud 6.5 Edge 5 CDN Guide

- `hypervisor_id` - the ID of a compute resource the IP address is associated with
- `id` - the ID of the IP address
- `ip_address_pool_id` - ID of the IP address pool the IP address is associated with
- `network_address` - IP address of the network
- `network_id` - the ID of the network
- `pxe` - true, if this compute resource address can be used for cloud booting a compute resource
- `updated at` - time when the IP address was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format
- `user_id` - the ID of the user this IP address is assigned to
- `free` - true if free, otherwise false
- `netmask` - netmask for the IP address

`monthly_bandwidth_used` - accelerator monthly bandwidth in KB
`total_disk_size` - total disk space in GB of primary and swap disks
`price_per_hour` - accelerator's price per hour
`price_per_hour_powered_off` - price per hour when accelerator is powered off
`support_incremental_backups` - 1, if accelerator supports incremental backups, and 0 if it does not
`cpu_priority` - this is a new parameter reserved for further use; currently will have the same value as `cpu_shares`
`edge_status` - the CDN server status
`cdn_reference` - the identifier in database

### 4.8.2 Get Accelerator Details
To view the accelerator details, use the following request:

GET /accelerators/:id.xml
GET /accelerators/:id.json

**XML Request Example**

```bash
curl -i -X GET -u user:userpass http://onapp.test/accelerators/:id.xml
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass http://onapp.test/accelerators/:id.json
```

**XML Output Example**
Where:

- **admin_note** – an optional reminder for this accelerator created by an administrator
- **allowed_hot_migrate** – true if hot migration is allowed; otherwise false
- **allowed_swap** – true if swap is allowed; otherwise false
- **booted** – true if the accelerator is booted; otherwise false
- **built** - true if the accelerator is built; otherwise false
- **cores_per_socket** - the number of cores per socket for accelerator
- **cpu_shares** – the CPU priority percentage
- **cpu_sockets** - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
- **cpu_threads** - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
- **cpu_units** - the amount of CPU units per core if the CPU priority is replaced with CPU units in user bucket
- **cpus** – number of CPU cores allocated to this accelerator
- **created_at** – the date when the accelerator was created in the [YYYY][MM][DD][T][hh][mm][ss]Z format
- **customer_network_id** - ID of a customer network
- **deleted_at** - time when the accelerator was deleted
- **enable_autoscale** – false; not available for accelerators
- **enable_monitis** - deprecated attribute; will be removed in upcoming release
- **firewall_notrack** - true if the NOTRACK rule is set in iptables
- **hot_add_cpu** - false; not available for accelerators
- **hot_add_memory** - false; not available for accelerators
- **hypervisor_id** – the ID of the compute resource, on which the accelerator is deployed
- **id** – the accelerator ID in OnApp CP database
- **identifier** – the accelerator identifier
- **initial_root_password** – the accelerator root password
- **initial_root_password_encrypted** – true, if the accelerator root password is encrypted, otherwise false
instance package_id - false; not available for accelerators
iso_id - false; not available for accelerators
label – an arbitrary name of the accelerator
local_remote_access_ip_address - the IP address used for console access
local_remote_access_port – the port ID used for console access
locked – true if locked; otherwise false
memory – the amount of RAM resources allocated to this accelerator
min_disk_size – minimum disk space required by the template
note - an optional reminder for this accelerator made by a user account
operating_system – type of operating system
operating_system_distro – the distribution of the operating system
preferred_hvs - the array of preferable compute resources based on compute zone that meet some accelerator configuration settings
recovery_mode – true if the accelerator is booted in the recovery mode; otherwise false
remote_access_password – the password for remote access
service_password - service account password
state – deprecated attribute; will be removed in upcoming release
strict_virtual_machine_id - the ID of a virtual server (or edge server) that will never reside on the same compute resource with this accelerator
suspended – true if suspended; otherwise false
template_id – the ID of the template, on which the accelerator is based
template_label – label of the template on which the accelerator is based
updated_at – the date when the accelerator was updated in the [YYYY][MM][DD][hh][mm][ss]Z format
user_id – the ID of the user, who is the accelerator owner
vip – true if the accelerator has VIP status for migration; otherwise false
xen_id – the accelerator ID set by the virtualization engine
ip_addresses – an array of assigned IP addresses with their details assigned to this accelerator:
  - address – IP address
  - broadcast – a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.
  - created_at – time when the IP address was created in the [YYYY][MM][DD][hh][mm][ss]Z format
  - customer_network_id - customer network ID
  - disallowed_primary – true if not allowed to be used as primary (for baremetal server), otherwise false
  - gateway - gateway address
  - hypervisor_id - the ID of a compute resource the IP address is associated with
  - id – the ID of the IP address
  - ip_address_pool_id - ID of the IP address pool the IP address is associated with
  - network_address - IP address of the network
• network_id - the ID of the network
• pxe - true, if this compute resource address can be used for cloudbooting a compute resource
• updated at - time when the IP address was updated in the [YYYY][MM][DD][hh][mm][ss]Z format
• user_id - the ID of the user this IP address is assigned to
• free – true if free, otherwise false
• netmask — netmask for the IP address
• monthly_bandwidth_used - accelerator monthly bandwidth in KB
• total_disk_size – total disk space in GB of primary and swap disks
• price_per_hour - accelerator's price per hour
• price_per_hour_powered_off - price per hour when accelerator is powered off
• support_incremental_backups - 1, if accelerator supports incremental backups, and 0 if it does not
• cpu_priority - this is a new parameter reserved for further use; currently will have the same value as cpu_shares
• edge_status - the CDN server status
• cdn_reference - the identifier in database

4.8.3 Add Accelerator
To create an accelerator, use the following API call:

POST / accelerators.xml
POST / accelerators.json

Below you can find requirements for Accelerator creation:
• Minimum: 4 cores, 4GB RAM and 100GB disks
• Recommended: 8 cores, 16 GB RAM and 1TB disks
• SSD recommended to avoid slowing down access

XML Request example
curl -i -X POST -d
'"<accelerator><template_id>37</template_id><label>Accelerator</label><hostname>Accelerator</hostname><hypervisor_group_id>4</hypervisor_group_id><hypervisor_id>2</hypervisor_id><memory>4096</memory><cpus>4</cpus><cpu_shares>1</cpu_shares><data_store_group_primary_id>12</data_store_group_swap_id>8504</data_store_group_swap_id><swap_disk_size>100</swap_disk_size><primary_network_group_id>null</primary_network_group_id><required_ip_address_assignment>null</required_ip_address_assignment><rate_limit>0</rate_limit><required_virtual_machine_build></required_virtual_machine_build><initial_root_password><primary_disk_min_iops>100</primary_disk_min_iops><swap_disk_min_iops>100</swap_disk_min_iops><enable_autoscale>false</enable_autoscale><selected_ip_address>null</selected_ip_address></accelerator>
" -u user:userpass http://onapp.test/accelerators.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'

JSON Request example

curl -i -X POST -d '{"accelerator": {"template_id": 37, "label": "Accelerator", "hostname": "Accelerator", "hypervisor_group_id": 4, "hypervisor_id": 2, "memory": 4096, "cpus": 4, "cpu_shares": 1, "data_store_group_primary_id": 12, "data_store_group_swap_id": 8504, "swap_disk_size": 100, "primary_network_group_id": null, "required_ip_address_assignment": 1, "rate_limit": 0, "required_virtual_machine_build": 1, "initial_root_password": "", "primary_disk_min_iops": 100, "swap_disk_min_iops": 100, "enable_autoscale": false}, "selected_ip_address": null}' -u user:userpass
http://onapp.test/accelerators.json
-H 'Accept: application/json'
-H 'Content-type: application/json'

Where:

- **template_id** - the ID of the template the accelerator is based on
- **label** - a unique name of your accelerator. The label can consist of letters [A-Za-z], digits [0-9], dash [-], lower dash [ _ ], space character [ ], at sign [@], round brackets [()], slashes [/], comma [,] and dot [.]. You can use both lower- and uppercase letters. The label should begin with an alphanumeric character or lower dash [ _ ]
- **hostname** - specify the accelerator hostname
- **hypervisor_group_id** - indicate the compute zone ID
- **hypervisor_id** - indicate the ID of the compute resource, on which the accelerator will be deployed
- **memory** - the amount of RAM, which you want to allocate to this accelerator
- **cpus** - the amount of CPU cores allocated to this accelerator
- **cpu_shares** - the percentage of allocated CPU priority resource
- **data_store_group_primary_id** – specify the ID of a data store zone, where you want to locate the disk of your accelerator. If not specified – the system will select the data store zone with higher available capacity
- **primary_disk_size** - the size in GB of the primary disk
- **data_store_group_swap_id** - set the ID of the data store zone to which this swap disk is allocated
**swap_disk_size** - set swap space. There is no swap disk for Windows-based accelerators

**primary_network_group_id** – indicate the network zone ID

**required_ip_address_assignment** - set "1" if you want IP address to be assigned automatically after creation. Otherwise set "0"

**rate_limit** - the port speed

**required_virtual_machine_build** – set "1" to build the accelerator automatically after creation. Otherwise set "0"

**initial_root_password** - the new root password for a container server. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [ - ] and lower dash [ _ ]. You can use both lowercase and uppercase letters.

**primary_disk_min_ops** - minimum number of IO operations per second for primary disk (this is a SolidFire related parameter)

**swap_disk_min_ops** - minimum number of IO operations per second for swap disk (this is a SolidFire related parameter)

**enable_autoscale** - true if autoscaling is allowed for this accelerator

**selected_ip_address** - an IP address to assign to this container server; if the parameter required_ip_address_assignment was set "1" but this parameter selected_ip_address is empty - the first available IP address will be assigned to container server automatically

---

**Page history:**

v. 6.0

- Added the following parameters:
  - **template_id**
  - **hostname**
  - **data_store_group_swap_id**
  - **swap_disk_size**
  - **initial_root_password**
  - **primary_disk_min_ops**
  - **swap_disk_min_ops**
  - **enable_autoscale**

- Removed the **network_id** parameter

v. 5.4:

- added **selected_ip_address** parameter
- added **network_id** parameter

---

**4.8.4 Add/Edit Admin/User Note for CDN Accelerator**

To edit/make an admin note, use the following request:

PUT /accelerators/:accelerator_id.xml
PUT /accelerators/:accelerator_id.json

XML Request Example

```bash
curl -i -X PUT -u user:userpass http://onapp.test/accelerators/9.xml -d
  '<accelerator><admin_note>agfagwe tiuuytjghyuytu</admin_note></accelerator>'
  -H 'Accept:application/xml' -H 'Content-type:application/xml'
```

JSON Request Example

```bash
curl -i -X PUT -u user:userpass http://onapp.test/accelerators/9.json -d
  '{"accelerator":{"admin_note":"kjfjhhjtrtjt"}}'
  -H 'Accept:application/json' -H 'Content-type:application/json'
```

Where:

- **admin_note** – enter the text of your note.

Add/Edit User Note XML Request Example

```bash
curl -i -X PUT -u user:userpass http://onapp.test/accelerators/9.xml -d
  '<accelerator><note>changed</note></accelerator>'
  -H 'Accept:application/xml' -H 'Content-type:application/xml'
```

Add/Edit User Note JSON Request Example

```bash
curl -i -X PUT -u user:userpass http://onapp.test/accelerators/9.json -d
  '{"accelerator":{"note":"kjfjhhjtrtjt"}}'
  -H 'Accept:application/json' -H 'Content-type:application/json'
```

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no accelerator with a requested ID, or URL is incorrect.

### 4.8.5 Edit Accelerator

To change the accelerator label or resource allocation, use the following request:

PUT /accelerators/:id.xml

XML Request Example

```bash
curl -i -X PUT -d
  '<accelerator><label>test</label><cpus>1</cpus><cpu_shares>10</cpu_shares>
  <memory>512</memory></accelerator>'
  -u onapp.test
  http://onapp.test/accelerators/9.xml
  -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request Example
curl -i -X PUT -d

Where:

* **label** – a unique name of your accelerator. The label can consist of letters [A-Za-z], digits [0-9], dash [-], lower dash [-], space character [ ], at sign [@], round brackets [()], slashes [/], comma [.] and dot [.]. You can use both lower- and uppercase letters. The label should begin with an alphanumeric character or lower dash [-].

* **cpus** - the amount of CPU cores allocated to this accelerator

* **cpu_shares** - the percentage of allocated CPU priority resource

* **memory** - the amount of RAM, which you want to allocate to this accelerator

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no accelerator with a requested ID, or URL is incorrect.

### 4.8.6 Reboot Accelerator

To reboot the accelerator, use the following request:

POST /accelerators/:accelerator_id/reboot.xml
POST /accelerators/:accelerator_id/reboot.json

**XML Request Example**

```bash
curl -i -X POST -u user:userpass
```

**JSON Request Example**

```bash
curl -i -X POST -u user:userpass
```

### 4.8.7 Start up Accelerator

To start up an accelerator, use the following request:

POST /accelerators/:accelerator_id/startup.xml
POST /accelerators/:accelerator_id/startup.json

**XML Request Example**

```bash
curl -i -X POST -u user:userpass
```

**JSON Request Example**

```bash
curl -i -X POST -u user:userpass
```
4.8.8 Shut down Accelerator
To terminate an edge server gracefully, use the following request:

POST /accelerators/:accelerator_id/shutdown.xml
POST /accelerators/:accelerator_id/shutdown.json

XML Request Example

```bash
curl -i -X POST -u user:userpass
http://onapp.test/accelerators/9/shutdown.xml -H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

JSON Request Example

```bash
curl -i -X POST -u user:userpass
-H 'Content-type: application/json'
```

4.8.9 Suspend Accelerator
To suspend an accelerator, use the following request:

POST /accelerators/:accelerator_id/suspend.xml
POST /accelerators/:accelerator_id/suspend.json

XML Request Example

```bash
curl -i -X POST -u user:userpass
http://onapp.test/accelerators/9/suspend.xml -H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

JSON Request Example

```bash
curl -i -X POST -u user:userpass
http://onapp.test/accelerators/9/suspend.json -H 'Accept: application/json'
-H 'Content-type: application/json'
```

To unsuspend the accelerator, run the request again.

4.8.10 Rebuild Accelerator
To rebuild (or build manually) the accelerator, use the following request:

POST /accelerators/:accelerator_id/build.xml
POST /accelerators/:accelerator_id/build.json

XML Request Example

```bash
curl -i -X POST -u user:userpass
http://onapp.test/accelerators/9/build.xml -H 'Accept: application/xml'
-H 'Content-type: application/xml'
```
4.8.11 Migrate Accelerator

To migrate an accelerator to another compute resource, use the following request:

POST /accelerators/:accelerator_id/migrate.xml
POST /accelerators/:accelerator_id/migrate.json

Currently, accelerators support only cold migration.

Where:

destination * - the ID of a target compute resource, to which you migrate the accelerator
cold_migrate_on_rollback - set 1 if you wish to switch to a cold migration if hot migration fails, otherwise set 0.
If there are accelerated virtual servers in the cloud, these VSs will be still billed for acceleration even if you delete the accelerator.

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no accelerator with a requested ID, or URL is incorrect.

4.8.13 Unlock Accelerator
To unlock the accelerator, use the following request:

POST /accelerators/:accelerator_id/unlock.xml
POST /accelerators/:accelerator_id/unlock.json

XML Request Example

curl -i -X POST -u user:userpass

JSON Request Example


4.8.14 Segregate Accelerator
To segregate an accelerator (that is, instruct it never to reside on the same compute resource with another accelerator), use the following request:

POST /accelerators/:accelerator_id/strict_vm.xml
POST /accelerators/:accelerator_id/strict_vm.json

XML Request Example

curl -i -X POST -u user:userpass
http://onapp.test/accelerators/21/strict_vm.xml

JSON Request Example

curl -i -X POST -u user:userpass
JSON Request Example

curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d "{{"virtual_machine":{"strict_virtual_machine_id":"gv03xz1x31t53h"}}}" --url http://onapp.test/accelerators/21/strict_vm.json

Where:

strict_virtual_machine_id * - the ID of accelerator you wish to segregate from the given accelerator

4.8.15 Change Accelerator Owner

To reassign an accelerator to another user, use the following request:

POST /accelerators/:accelerator_id/change_owner.xml
POST /accelerators/:accelerator_id/change_owner.json

XML Request Example


Where:

user_id * – input ID of a new owner

4.8.16 Accelerator Network Interfaces

Here is the list of API requests for managing accelerators' network interfaces. Accelerators' network interfaces have the same attributes as network interfaces of virtual servers.

To get the list of network interfaces allocated to this particular accelerator, use the following request:

GET /accelerators/:accelerator_id/network_interfaces.xml
GET /accelerators/:accelerator_id/network_interfaces.json

To get a particular network interface details, use the following request:

GET /accelerators/:accelerator_id/network_interfaces/:id.xml
GET /accelerators/:accelerator_id/network_interfaces/:id.json

To edit network interface details, use the following request:

PUT /accelerators/:accelerator_id/network_interfaces/:id.xml
PUT /accelerators/:accelerator_id/network_interfaces/:id.json

To add a new network interface, use the following request:

POST /accelerators/:accelerator_id/network_interfaces.xml
POST /accelerators/:accelerator_id/network_interfaces.json
To delete a network interface from the accelerator, use the following request:

DELETE /accelerators/:accelerator_id/network_interfaces/:id.xml
DELETE /accelerators/:accelerator_id/network_interfaces/:id.json

**XML Output Example**

```xml
<network_interface>
  <connected nil="true"/>
  <created_at type="datetime">2015-09-25T14:33:13+03:00</created_at>
  <default_firewall_rule>ACCEPT</default_firewall_rule>
  <id type="integer">372</id>
  <identifier>yekx0libarssan</identifier>
  <label>eth0</label>
  <mac_address>00:16:3e:81:42:83</mac_address>
  <network_join_id type="integer">5</network_join_id>
  <primary type="boolean">true</primary>
  <rate_limit type="integer">1</rate_limit>
  <updated_at type="datetime">2015-09-25T14:33:13+03:00</updated_at>
  <usage nil="true"/>
  <usage_last_reset_at nil="true"/>
  <usage_month_rolled_at nil="true"/>
  <virtual_machine_id type="integer">359</virtual_machine_id>
</network_interface>
```

Where:

- **label** - network interface name
- **created_at** - the timestamp in the database when this network interface was created
- **default_firewall_rule** - set default firewall rule for the particular network interface – either DROP or ACCEPT
- **updated_at** - the timestamp in the database when this network interface was updated
- **primary** - True if this network interface is primary, otherwise false
- **id** - the ID of this network interface
- **mac_address** – network interface mac address
- **rate_limit** - port speed in Mbps
- **identifier** - the identifier in the database of this network interface
- **network_join_id** - the ID of the network join to which this network interface belongs
- **virtual_machine_id** - the ID of an accelerator to which this network interface is attached
- **connected** - not relevant to accelerators
- **usage** - not relevant to accelerators
- **usage_last_reset_at** - not relevant to accelerators
- **usage_month_rolled_at** - not relevant to accelerators

### 4.8.17 Accelerator IP Address Joins

An IP address allocated to an accelerator is an IP address join. Use the following requests to view, assign and delete IP address joins of your accelerators.

To get the list of IP address assignments for a particular accelerator, use the following request:

GET /accelerators/:accelerator_id/ip_addresses.xml
GET /accelerators/:accelerator_id/ip_addresses.json

**XML Request Example**
To assign an IP Address to an accelerator, use the following request:

**XML Request Example**

```
```

**JSON Request Example**

```
```
<ip_address_join>
  <created_at type="datetime">2013-10-31T13:04:05+03:00</created_at>
  <id type="integer">173</id>
  <ip_address_id type="integer">7</ip_address_id>
  <network_interface_id type="integer">131</network_interface_id>
  <updated_at type="datetime">2013-10-31T13:04:05+03:00</updated_at>

  <ip_address>
    <address>1.1.3</address>
    <broadcast>1.1.1.255</broadcast>
    <created_at type="datetime">2013-08-07T13:29:09+03:00</created_at>
    <customer_network_id nil="true"/>
    <disallowed_primary type="boolean">false</disallowed_primary>
    <gateway>1.1.1</gateway>
    <hypervisor_id nil="true"/>
    <id type="integer">7</id>
    <ip_address_pool_id nil="true"/>
    <network_address>1.1.1.0</network_address>
    <network_id type="integer">1</network_id>
    <pxe type="boolean">false</pxe>
    <updated_at type="datetime">2013-08-07T13:29:09+03:00</updated_at>
    <user_id nil="true"/>
    <free type="boolean">false</free>
    <netmask>255.255.255.0</netmask>
  </ip_address>
</ip_address_join>

Where:

created_at - the date when the record was created in DB
id - the IP address join ID
ip_address_id - the IP address ID
network_interface_id - the network interface ID
updated_at - the date when the record was updated in DB
ip_address - the array of IP address details
address - the IP address
broadcast - a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.
customer_network_id - the ID of the customer network
disallowed_primary - true if this address is not set as primary (for VS build), otherwise false
gateway - gateway address
hypervisor_id - the ID of the compute resource
ip_address_pool_id - the ID of the IP address pool to which this join belongs
network_address - the address of a VLAN network address that will be associated with this IP address pool
network_id - the ID of the network
pxe - true, if this address can be used for cloudbooting a compute resource
free - true if free, otherwise false
netmask - netmask for the IP address

To delete an IP address assignment from a particular accelerator, use the following request:
DELETE /accelerators/:accelerator_id/ip_addresses/:id.xml
DELETE /accelerators/:accelerator_id/ip_addresses/:id.json

XML Request Example


JSON Request Example


Where:
data_store_id - the ID of the data store, which is attached to the compute resource
hypervisor_id - reserved parameter
id - the join ID
target_join_id - the ID of the join target; in this case it is the compute resource ID
target_join_type - type of join target; in this case it is compute resource

4.8.18 View Accelerator Disks

To view the accelerator disks, use the following request:
GET /accelerators/:accelerator_id/disks.xml
GET /accelerators/:accelerator_id/disks.json

XML Request Example


JSON Request Example


XML Output Example
<disks type="array">
  <disk>
    <mounted>true</mounted>
    <built type="boolean">true</built>
    <burst_bw type="integer">1000</burst_bw>
    <created_at type="datetime">2015-09-25T14:33:13+03:00</created_at>
    <data_store_id type="integer">4</data_store_id>
    <disk_size type="integer">20</disk_size>
    <disk_vm_number type="integer">1</disk_vm_number>
    <file_system type="symbol">ext3</file_system>
    <id type="integer">460</id>
    <identifier>pd60674pgnqfx4</identifier>
    <iqn nil="true"/>
    <is_swap type="boolean">false</is_swap>
    <label>Disk#460</label>
    <locked type="boolean">false</locked>
    <max_bw type="integer">1000</max_bw>
    <mount_point nil="true"/>
    <primary type="boolean">true</primary>
    <updated_at type="datetime">2015-09-27T19:26:54+03:00</updated_at>
    <virtual_machine_id type="integer">359</virtual_machine_id>
    <volume_id nil="true"/>
    <has_autobackups type="boolean">false</has_autobackups>
  </disk>
</disks>

Where:

- **add_to_freebsd_fstab** - true, if this disk is added to the FreeBSD fstab, otherwise false
- **add_to_linux_fstab** - true, if this disk is added to Linux fstab, otherwise false
- **mounted** - set 'true' to mount the disk inside OS automatically, otherwise set 'false'

You can use a single mounted parameter, to substitute the two add_to_linux_fstab and add_to_freebsd_fstab parameters.

- **built** - true if the disk is built, otherwise false
- **created_at** - the date when the disk was created in the [YYYY][MM][DD][T][hh][mm][ss]Z format
- **updated_at** - the date when the disk was updated in the [YYYY][MM][DD][T][hh][mm][ss]Z format
- **data_store_id** - the ID of the data store this disk is located
- **disk_size** - disk size in GB
- **disk_vm_number** - the number of virtual servers using this disk
- **file_system** - disk filesystem (ext3 or ext4)
- **id** - the disk ID
- **identifier** - disk identifier
- **is_swap** - true if this is a swap disk, otherwise false
- **label** - disk's label
- **locked** - true if the disk is locked, otherwise false
- **mount_point** - disk mount point.
- **primary** - true if the disk is primary. Otherwise false.
- **virtual_machine_id** - the ID of the accelerator using this disk
volume_id - data store ID

has_autobackups - true if the disk has automatic backups set up, otherwise false

SolidFire - related parameters (irrelevant for accelerators)

  iqn
  burst_bw
  max_bw

4.8.19 Rebuild Network for Accelerator

It is required to rebuild network after any changes on IP address joins or network interfaces. To rebuild network, use the following request:

POST /accelerators/:accelerator_id/rebuild_network.xml
POST /accelerators/:accelerator_id/rebuild_network.json

XML Request Example

```bash
curl -X POST -u user:userpass --url
  'http://onapp.test/accelerators/21/rebuild_network.xml?force=1&shutdown_type=hard&required_startup=1'
  -H 'Accept: application/xml'
  -H 'Content-type: application/xml'
```

JSON Request Example

```bash
curl -X POST -u user:userpass --url
  'http://onapp.test/accelerators/21/rebuild_network.json?force=1&shutdown_type=hard&required_startup=1'
  -H 'Accept: application/json'
  -H 'Content-type: application/json'
```

Where:

  accelerator_id - ID of the accelerator

  shutdown_type - type of the accelerator shutdown: hard, graceful or soft

  required_startup - set 1 to start up the accelerator automatically after build, otherwise set 0

4.8.20 Get Accelerator CPU Usage Statistics

To view CPU usage statistics of an accelerator, use the following request:

GET /accelerators/:accelerator_id/cpu_usage.xml
GET /accelerators/:accelerator_id/cpu_usage.json

Define a shorter period by setting Start and End time in the API call:

GET /accelerators/:accelerator_id/vm_stats.xml?period[startdate]=YYYY-MM-DD+hh:mm:ss&period[enddate]=YYYY-MM-
XML Request Example

```
curl -i GET -u user:userpass --url http://onapp.test/accelerators/21/cpu_usage.xml
```

XML Request Example

```
curl -i GET -u user:userpass --url http://onapp.test/accelerators/21/cpu_usage.json
```

Where you have to specify the accelerator ID.

### 4.9 CDN Edge Groups API

CDN edge groups are groups of edge servers – your own, and those you subscribe to from the CDN marketplace. They are usually grouped by location, so they represent a pool of servers for a given geographical area. Once you have created an edge group containing edge servers in specific locations, you can then assign the group (or groups) to a specific CDN resource. You need to associate CDN Edge groups with buckets to make them available for users.

PLEASE NOTE: Starting from the OnApp Cloud v3.0, CDN is enabled automatically after adding the first DNS record or CDN resource.

### 4.9.1 Get List of CDN Edge Groups

To view CDN edge groups available in the cloud:

```
GET /edge_groups.xml
GET /edge_groups.json
```

XML Request Example

```
```

JSON Request Example
4.9.2 Get List of Available CDN Edge Groups

To view the list of all the edge groups and their locations, which are available to create CDN resources on, use the following request:

GET /cdn_resources/available_edge_groups.xml
GET /cdn_resources/available_edge_groups.json

The list of available edge groups is defined by the bucket to which a user is assigned.

XML Request Example

```
```

JSON Request Example

```
```

XML Output Example

```xml
<edge_groups type="array">
  <edge_group>
    <label>tredty</label>
    <created_at type="datetime">2011-10-11T12:58:40Z</created_at>
    <updated_at type="datetime">2011-10-11T12:58:40Z</updated_at>
    <id type="integer">1</id>
  </edge_group>
...
</edge_groups>
```
<edge_groups type="array">
  <edge_group>
    <edge_group_locations type="array">
      <edge_group_location>
        <city>dallas</city>
        <price type="decimal">0.7</price>
        <created_at type="datetime">2012-03-01T11:16:10+02:00</created_at>
        <country>US</country>
        <aflexi_location_id type="integer">147</aflexi_location_id>
        <updated_at type="datetime">2012-03-01T11:16:10+02:00</updated_at>
        <id type="integer">10</id>
        <operator>WK</operator>
      </edge_group_location>
    </edge_group_locations>
    <id type="integer">35</id>
  </edge_group>
</edge_groups>

Where:

edge_groups – the array of edge groups with their locations available for a user to create a CDN resource on.

edge_group – the particular edge group details:

label – the edge group label

edge_group_locations – the array of locations assigned to this group:

edge_group_location – the list of details for a particular edge group

city – the city where the edge server is located

price - price per GB of sold excess bandwidth

created_at – the date when the record was created in DB

country -- country codes, related to country_access_policy in ISO 3166-1 alpha-2 format

aflexi_location_id – the ID of this location in Aflexi database

updated_at – the date when the record was updated in DB

id – the location ID

operator – the location operator

edge_group_id – the ID of the edge group to which this location is assigned

4.9.3 Get CDN Edge Group Details

To view the edge group details, use the following request:

GET /edge_groups/:id.xml
GET /edge_groups/:id.json

XML Request Example

```
curl -i -X GET -u user:userpass http://onapp.test/edge_groups/:edge_group_id.xml?available_locations=true
-H 'Accept: application/xml' -H 'Content-type: application/xml'
```
**JSON Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/edge_groups/:edge_group_id.json -d
'{"available_locations":true}' -H 'Accept: application/json' -H 'Content-
type:application/json'
```

Where:

*available_locations* - set true to view the list of locations available to this edge group; set *false* to view only the list of assigned locations.

**XML Output Example**

```xml
<edge_group>
  <created_at type="datetime">2012-04-18T11:33:01+00:00</created_at>
  <id type="integer">152</id>
  <label>CDN Edge group name</label>
  <updated_at type="datetime">2012-04-18T11:33:01+00:00</updated_at>
  <assigned_locations type="array">
    <location>
      <description/>
      <id type="integer">146</id>
      <price type="float">0.3</price>
      <region>DC</region>
      <city>washington</city>
      <latitude type="float">38.895</latitude>
      <country>US</country>
      <deleted type="boolean">false</deleted>
      <longitude type="float">-77.0367</longitude>
    </location>
    ...
  </assigned_locations>
  <available_locations type="array">
    <location>
      <description>abc</description>
      <id type="integer">2</id>
      <price type="float">10.0</price>
      <region>T2</region>
      <city>bangor</city>
      <latitude type="float">54.65</latitude>
      <country>GB</country>
      <deleted type="boolean">false</deleted>
      <longitude type="float">-5.7</longitude>
    </location>
    ...
  </available_locations>
</edge_group>
```

Where:

*available_locations* – an array of all available locations

*assigned_locations* – an array of locations, which are assigned to the group

*city* – city where the edge server is located

*region* – region where the edge server is located

*price* – price per GB of sold excess bandwidth
4.9.4 Add CDN Edge Group

To create an edge group, use the following request:

POST /edge_groups.xml
POST /edge_groups.json

XML Request Example

```
curl -i -X POST -u user:userpass http://onapp.test/edge_groups.xml -d
'<?xml version="1.0" encoding="UTF-8"?>
<edge_group>
  <label>az_3</label>
</edge_group>'
-H 'Accept:application/xml'
-H 'Content-type:application/xml'
```

JSON Request Example

```
curl -i -X POST -u user:userpass http://onapp.test/edge_groups.json -d
'{"edge_group":{"label":"az_4"}}'
-H 'Accept:application/json'
-H 'Content-type:application/json'
```

Where:

- **label** - the name of new group

4.9.5 Edit Edge Group

You can edit the **label** of the edge group:

PUT /edge_groups/:id.xml
PUT /edge_groups/:id.json

XML Request Example

```
curl -i -X PUT -u user:userpass http://onapp.test/edge_groups/12.xml -d
'<?xml version="1.0" encoding="UTF-8"?>
<edge_group>
  <label>az_5</label>
</edge_group>'
-H 'Accept:application/xml'
-H 'Content-type:application/xml'
```

JSON Request Example

```
curl -i -X PUT -u user:userpass http://onapp.test/edge_groups/12.json -d
'{"edge_group":{"label":"az_6"}}'
-H 'Accept:application/json'
-H 'Content-type:application/json'
```
Returns HTTP 204 response on successful processing, and HTTP 404 when there is no edge group with a requested ID, or URL is incorrect.

4.9.6 Delete Edge Group
To delete the edge group, use the following request:

DELETE /edge_groups/:id.xml
DELETE /edge_groups/:id.json

XML Request Example

```
curl -i -X DELETE -u user:userpass http://onapp.test/edge_groups/12.xml -H 'Accept:application/xml' -H 'Content-type:application/xml'
```

JSON Request Example

```
curl -i -X DELETE -u user:userpass http://onapp.test/edge_groups/12.json -H 'Accept:application/json' -H 'Content-type:application/json'
```

Where you have to specify ID of a CDN group you want to delete in the URL.

Be careful when deleting an edge group which is associated with CDN resources.

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no edge group with a requested ID, or URL is incorrect.

4.9.7 Assign Location to CDN Edge Group
Check the ID of the required location and assign it to the group with the following API call:

POST /edge_groups/:edge_group_id/assign.xml
POST /edge_groups/:edge_group_id/assign.json

XML Request Example

```
curl -i -X POST -u user:userpass http://onapp.test/edge_groups/12/assign.xml -d '<location>175</location>' -H 'Accept:application/xml' -H 'Content-type:application/xml'
```

JSON Request Example

```
curl -i -X POST -u user:userpass http://onapp.test/edge_groups/12/assign.json -d '{"location":"175"}' -H 'Accept:application/json' -H 'Content-type:application/json'
```

Where:

*location* - input the ID of the required location
4.9.8 Unassign Location From CDN Edge Group

To remove a location from the group, use the following request:

POST /edge_groups/:edge_group_id/unassign.xml
POST /edge_groups/:edge_group_id/unassign.json

XML Request Example

```bash
curl -i -X POST -u user:userpass http://onapp.test/edge_groups/1/unassign.xml -d 'location=175' -H 'Accept:application/xml' -H 'Content-Type:application/xml'
```

JSON Request Example

```bash
curl -i -X POST -u user:userpass http://onapp.test/edge_groups/1/unassign.json -d '{location:175}' -H 'Accept:application/json' -H 'Content-Type:application/json'
```

Where you have to specify ID of a location you want to unassign.

You can retrieve the list of location IDs with the Get CDN Edge Group Details API call.

4.9.9 Modify CDN Edge Group

To modify CDN edge group, use the following request:

POST /edge_groups/:edge_group_id/modify.xml
POST /edge_groups/:edge_group_id/modify.json

XML Request Example

```bash
curl -i -X POST -u user:userpass http://onapp.test/edge_groups/1/modify.xml -d '<locations type="array"><location>123</location><location>123</location><location>123</location></locations>' -H 'Accept:application/xml' -H 'Content-Type:application/xml'
```

JSON Request Example
Where you have to specify IDs of locations you want to see eventually in the required CDN resource (at least one ID is required).

You can retrieve the list of location IDs with the Get CDN Edge Group Details API call.

### 4.9.10 Search CDN Edge Groups

To search for a specific CDN Edge Group, use the following request:

**XML Request Example**

```sh
```

**JSON Request Example**

```sh
```

The request will search for the Edge Group with the `test` label.

**XML Output Example**
<edge_groups type="array">
  <edge_group>
    <created_at type="datetime">2013-08-12T11:39:09+03:00</created_at>
    <id type="integer">228</id>
    <label>PItest</label>
    <updated_at type="datetime">2013-08-12T11:39:09+03:00</updated_at>
    <cdn_reference type="integer">426776953</cdn_reference>
  </edge_group>
  <edge_group>
    <created_at type="datetime">2013-08-19T14:32:54+03:00</created_at>
    <id type="integer">232</id>
    <label>TestPI</label>
    <updated_at type="datetime">2013-08-19T14:32:54+03:00</updated_at>
    <cdn_reference type="integer">668633450</cdn_reference>
  </edge_group>
</edge_groups>

Where:
- **created_at** – the date when the edge group was created
- **id** – the resource ID in the database
- **label** – the edge group label
- **updated_at** – the date when the edge group was updated
- **cdn_reference** – the identifier in database

### 4.10 CDN Edge Servers API

CDN edge servers are the virtual server which form a Content Delivery Network. In this network the web content is cached and delivered to end users from the server which is closest to the user or has the best availability.

#### 4.10.1 Get List of CDN Edge Servers

To view all edge servers in the cloud with their details, use the following request:

GET /edge_servers.xml
GET /edge_servers.json

**XML Request Example**

```
curl -i -X GET -u user:pass http://onapp.test/edge_servers.xml
```

**JSON Request Example**

```
curl -i -X GET -u user:pass http://onapp.test/edge_servers.json
```

To get the list of HTTP edge servers

**XML Request Example**
To get the list of streaming edge servers:

XML Request Example

curl -i -X GET -u user:userpass
http://onapp.test/edge_servers.xml?type=streaming

JSON Request Example

curl -i -X GET -u user:userpass
http://onapp.test/edge_servers.json?type=streaming
<edge_servers type="array">
<edge_server>
<add_to_marketplace type="boolean">true</add_to_marketplace>
<admin_note nil="true"/>
<allowed_hot_migrate type="boolean">false</allowed_hot_migrate>
<allowed_swap type="boolean">true</allowed_swap>
<booted type="boolean">false</booted>
<built type="boolean">true</built>
<cpuShares type="integer">7</cpuShares>
<cpu_sockets nil="true"/>
<cpu_threads nil="true"/>
<cpu_units nil="true"/>
<cpu_units type="integer">140</cpu_units>
<created_at type="datetime">2015-02-10T14:19:45+02:00</created_at>
<customer_network_id nil="true"/>
<deleted_at nil="true"/>
<edge_server_type>streaming</edge_server_type>
<enable_autoscale nil="true"/>
<enable_monitis type="boolean">false</enable_monitis>
<firewall_notrack type="boolean">true</firewall_notrack>
<hypervisor_id type="integer">25</hypervisor_id>
<identifier>kcs0o46otoxbr0</identifier>
<initial_root_password>3yV4Orl1B1Le</initial_root_password>
<initial_root_password_encrypted type="boolean">false</initial_root_password_encrypted>
<label>qawsedrf</label>
<local_remote_access_ip_address>109.123.91.36</local_remote_access_ip_address>
<locked type="boolean">false</locked>
<memory type="integer">2054</memory>
<min_disk_size type="integer">20</min_disk_size>
<note nil="true"/>
<operating_system>linux</operating_system>
<operating_system_distro>ubuntu</operating_system_distro>
<preferred_hvs type="array"/>
<recovery_mode type="boolean">false</recovery_mode>
<remote_access_password>GPRdQyq28jVR</remote_access_password>
<service_password nil="true"/>
<state>delivered</state>
<storage_server_type nil="true"/>
<suspended type="boolean">false</suspended>
<template_id type="integer">11</template_id>
<template_label>debian-6.0-x64-1.14-xen.kvm.virtio.tar.gz</template_label>
<updated_at type="datetime">2015-03-05T10:18:53+02:00</updated_at>
<user_id type="integer">4</user_id>
<vip nil="true"/>
<xen_id type="true"/>
<ip_addresses type="array">
<ip_address>
<address>109.123.91.154</address>
<broadcast>109.123.91.191</broadcast>
<created_at type="datetime">2014-01-14T14:19:52+02:00</created_at>
<customer_network_id nil="true"/>
<disallowed_primary type="boolean">false</disallowed_primary>
<gateway>109.123.91.129</gateway>
<hypervisor_id nil="true"/>
<id type="integer">25</id>
<ip_address_pool_id nil="true"/>
<network_address>109.123.91.128</network_address>
<network_id type="integer">1</network_id>
<pxe type="boolean">false</pxe>
<updated_at type="datetime">2014-11-06T17:10:35+02:00</updated_at>
</ip_address>
</ip_addresses>
</edge_server>
</edge_servers>
Where:

- **add_to_marketplace** – true if this edge server is added to the marketplace; otherwise false
- **admin_note** – an optional reminder for this VS created by an administrator
- **allow_resize_without_reboot** – true if adjusting resource allocation without reboot is possible; otherwise false
- **allowed_hot_migrate** – true if hot migration is allowed; otherwise false
- **allowed_swap** – true if swap is allowed; otherwise false
- **booted** – true if the server is booted; otherwise false
- **cpu_shares** – the CPU priority percentage
- **cpu_sockets** - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
- **cpu_threads** - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
- **cpu_units** - the amount of CPU units per core if the CPU priority is replaced with CPU units in user bucket.
- **cpus** – number of CPU cores allocated to this edge server
- **created_at** – the date when the CDN edge server was created in the [YYYY][MM][DD][hh][mm][ss]Z format
- **customer_network_id** - ID of a customer network
- **deleted_at** - time when the VS was deleted
- **edge_server_type** - true if this is the edge server
- **enable_autoscale** – false; not available for edge servers
- **enable_monitis** - deprecated attribute; will be removed in upcoming release
- **firewall_notrack** - true if the NOTRACK rule is set in iptables
- **hypervisor_id** – the ID of the compute resource, on which the server is deployed
- **id** – the edge server ID in OnApp CP database
- **identifier** – the edge server identifier
- **initial_root_password** – the server root password
- **initial_root_password_encrypted** – true, if the server root password is encrypted, otherwise false
- **label** – an arbitrary name of the edge server
local_remote_access_port – the port ID used for console access

locked – true if locked; otherwise false

memory – the amount of RAM resources allocated to this edge server

min_disk_size – minimum disk space required by the template

note - an optional reminder for this VS made by a user account

operating_system – type of operating system

operating_system_distro – the distribution of the operating system

preferred_hvs - the array of preferable compute resources based on compute zone that meet some VS configuration settings

recovery_mode – true if the server is booted in the recovery mode; otherwise false

remote_access_password – the password for remote access

service_password - service account password

state – deprecated attribute; will be removed in upcoming release

storage_server_type - true if this is a storage server

strict_virtual_machine_id - the ID of a virtual server (or edge server) that will never reside on the same compute resource with this server

suspended – true if suspended; otherwise false

template_id – the ID of the template, on which the edge server is based

template_label – label of the template on which the server is based; currently – OnApp CDN compute resource

updated_at – the date when the CDN edge server was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id – the ID of the user, who is the server owner

vip – true if the server has VIP status for migration; otherwise false

xen_id – the edge server ID set by the virtualization engine

ip_addresses – an array of assigned IP addresses with their details assigned to this edge server:

- address – baremetal server IP
- broadcast – a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.
- created_at – time when the IP address was created in the [YYYY][MM][DD]T[hh][mm][ss]Z format
- customer_network_id - customer network ID
- disallowed_primary – true if not allowed to be used as primary (for baremetal server), otherwise false
- gateway - gateway address
- hypervisor_id - the ID of a compute resource the IP address is associated with
- id – the ID of the IP address
- ip_address_pool_id - ID of the IP address pool the IP address is associated with
- pxe - true, if this compute resource address can be used for cloudbooting a compute resource
- updated at - time when the IP address was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format
• **user_id** - the ID of the user this IP address is assigned to
• **free** - true if free, otherwise false
• **netmask** — netmask for the IP address

`monthly_bandwidth_used` - VS monthly bandwidth in KB

`total_disk_size` – total disk space in GB of primary and swap disks

`price_per_hour` - server's price per hour

`price_per_hour_powered_off` - price per hour when server is powered off

`support_incremental_backups` - 1, if virtual server supports incremental backups, and 0 if it does not

`cpu_priority` - this is a new parameter reserved for further use; currently will have the same value as `cpu_shares`

`edge_status` - the edge server status

`cdn_reference` - the identifier in database

### 4.10.2 Get CDN Edge Server Details

To view the edge server details, use the following request:

GET `/edge_servers/:id.xml`
GET `/edge_servers/:id.json`

**XML Request Example**

```bash
curl -i -X GET -u user:userpass http://onapp.test/edge_servers/3781.xml
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass http://onapp.test/edge_servers/3781.json
```

To get the list of HTTP edge servers, use the following request:

**XML Request Example**

```bash
curl -i -X GET -u user:userpass http://onapp.test/edge_servers.xml?type=http
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass http://onapp.test/edge_servers.json?type=http
```

To get the list of streaming edge servers, use the following request:

**XML Request Example**

```bash

```
curl -i -X GET -u user:userpass
http://onapp.test/edge_servers.xml?type=streaming

**JSON Request Example**

curl -i -X GET -u user:userpass
http://onapp.test/edge_servers.json?type=streaming

**XML Output Example**
```xml
<edge_servers type="array">
<edge_server>
<add_to_marketplace type="boolean">true</add_to_marketplace>
<admin_note nil="true"/>
<allowed_hot_migrate type="boolean">false</allowed_hot_migrate>
<allowed_swap type="boolean">true</allowed_swap>
<booted type="boolean">false</booted>
<built type="boolean">true</built>
<cpu_shares type="integer">7</cpu_shares>
<cpu_sockets nil="true"/>
<cpu_threads nil="true"/>
<cpu_units type="integer">140</cpu_units>
<cpus type="integer">1</cpus>
<created_at type="datetime">2015-02-10T14:19:45+02:00</created_at>
<customer_network_id nil="true"/>
<deleted_at nil="true"/>
<edge_server_type>streaming</edge_server_type>
<enable_autoscale nil="true"/>
<enable_monitis type="boolean">false</enable_monitis>
<firewall_notrack type="boolean">true</firewall_notrack>
<hypervisor_id type="integer">25</hypervisor_id>
<id type="integer">3781</id>
<identifier>kcs0o46otoxbr0</identifier>
<initial_root_password>3yV4Orl1B1Le</initial_root_password>
<initial_root_password_encrypted type="boolean">false</initial_root_password_encrypted>
<label>qawsedrf</label>
<local_remote_access_ip_address>109.123.91.36</local_remote_access_ip_address>
<locked type="boolean">false</locked>
<memory type="integer">2054</memory>
<min_disk_size type="integer">20</min_disk_size>
<note nil="true"/>
<operating_system>linux</operating_system>
<operating_system_distro>ubuntu</operating_system_distro>
<preferred_hvs type="array"/>
<recovery_mode type="boolean">false</recovery_mode>
<remote_access_password>GPRdQyq28jVR</remote_access_password>
<service_password nil="true"/>
<state>delivered</state>
<storage_server_type nil="true"/>
<strict_virtual_machine_id nil="true"/>
<suspended type="boolean">false</suspended>
<template_id type="integer">11</template_id>
<template_label>debian-6.0-x64-1.14-xen.kvm.virtio.tar.gz</template_label>
<updated_at type="datetime">2015-03-05T10:18:53+02:00</updated_at>
<user_id type="integer">4</user_id>
<vip nil="true"/>
<xen_id nil="true"/>
<ip_addresses type="array">
<ip_address>
<address>109.123.91.154</address>
<broadcast>109.123.91.151</broadcast>
<created_at type="datetime">2014-01-14T14:19:52+02:00</created_at>
<customer_network_id nil="true"/>
<disallowed_primary type="boolean">false</disallowed_primary>
<gateway>109.123.91.129</gateway>
<hypervisor_id nil="true"/>
<id type="integer">25</id>
<ip_address_pool_id nil="true"/>
<network_address>109.123.91.129</network_address>
<network_id type="integer">1</network_id>
<pxe type="boolean">false</pxe>
<updated_at type="datetime">2014-11-06T17:10:35+02:00</updated_at>
</ip_address>
</ip_addresses>
</edge_server>
</edge_servers>
```
<user_id nil="true"/>
<free type="boolean">false</free>
<netmask>255.255.255.192</netmask>
</ip_address>
</ip_addresses>
<monthly_bandwidth_used>0</monthly_bandwidth_used>
<price_per_hour type="float">20540.0</price_per_hour>
<price_per_hour_powered_off type="float">0.0</price_per_hour_powered_off>
<support_incremental_backups type="boolean">true</support_incremental_backups>
<cpu_priority type="integer">7</cpu_priority>
<edge_status>Active</edge_status>
<cdn_reference type="integer">276964394</cdn_reference>
</edge_server>

Where:

add_to_marketplace – true if this edge server is added to the marketplace; otherwise false
admin_note – an optional reminder for this VS created by an administrator
allow_resize_without_reboot – true if adjusting resource allocation without reboot is possible; otherwise false
allowed_hot_migrate – true if hot migration is allowed; otherwise false
allowed_swap – true if swap is allowed; otherwise false
booted – true if the server is booted; otherwise false
cpu_shares – the CPU priority percentage
cpu_sockets - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
cpu_threads - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted
cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user bucket.
cpus – number of CPU cores allocated to this edge server
created_at – the date when the CDN edge server was created in the [YYYY][MM][DD][hh][mm][ss]Z format
customer_network_id - ID of a customer network
deleted_at - time when the VS was deleted
domain
edge_server_type - true if this is the edge server
enable_autoscale – false; not available for edge servers
enable_monitis - deprecated attribute; will be removed in upcoming release
firewall_notrack - true if the NOTRACK rule is set in iptables
hypervisor_id – the ID of the compute resource, on which the server is deployed
id – the edge server ID in OnApp CP database
identifier – the edge server identifier
initial_root_password – the server root password
initial_root_password_encrypted – true, if the server root password is encrypted, otherwise false
label – an arbitrary name of the edge server
local_remote_access_port – the port ID used for console access
locked – true if locked; otherwise false
memory – the amount of RAM resources allocated to this edge server
min_disk_size – minimum disk space required by the template
note – an optional reminder for this VS made by a user account
operating_system – type of operating system
operating_system_distro – the distribution of the operating system
preferred_hvs - the array of preferable compute resources based on compute zone that meet some VS configuration settings
recovery_mode – true if the server is booted in the recovery mode; otherwise false
remote_access_password – the password for remote access
service_password - service account password
state – deprecated attribute; will be removed in upcoming release
storage_server_type - true if this is a storage server
strict_virtual_machine_id - the ID of a virtual server (or edge server) that will never reside on the same compute resource with this server
suspended – true if suspended; otherwise false
template_id – the ID of the template, on which the edge server is based
template_label – label of the template on which the server is based; currently – OnApp CDN compute resource
updated_at – the date when the CDN edge server was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format
user_id – the ID of the user, who is the server owner
vip – true if the server has VIP status for migration; otherwise false
xen_id – the edge server ID set by the virtualization engine
ip_addresses – an array of assigned IP addresses with their details assigned to this edge server:
  • address – baremetal server IP
  • broadcast – a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.
  • created_at – time when the IP address was created in the [YYYY][MM][DD]T[hh][mm][ss]Z format
  • customer_network_id - customer network ID
  • disallowed_primary – true if not allowed to be used as primary (for baremetal server), otherwise false
  • gateway - gateway address
  • hypervisor_id - the ID of a compute resource the IP address is associated with
  • id – the ID of the IP address
  • ip_address_pool_id - ID of the IP address pool the IP address is associated with
  • pxe - true, if this compute resource address can be used for cloudbooting a compute resource
- **updated at** - time when the IP address was updated in the [YYYY][MM][DD][hh][mm][ss]Z format
- **user_id** - the ID of the user this IP address is assigned to
- **free** - true if free, otherwise false
- **netmask** — netmask for the IP address

- **monthly_bandwidth_used** - VS monthly bandwidth in KB
- **total_disk_size** — total disk space in GB of primary and swap disks
- **price_per_hour** - server's price per hour
- **price_per_hour_powered_off** - price per hour when server is powered off
- **support_incremental_backups** - 1, if virtual server supports incremental backups, and 0 if it does not
- **cpu_priority** - this is a new parameter reserved for further use; currently will have the same value as cpu_shares
- **edge_status** - the edge server status
- **cdn_reference** - the identifier in database

### 4.10.3 Add CDN Edge Server

To create an edge server, use the following request:

**POST /edge_servers.xml**

**POST /edge_servers.json**

**XML Request Example**

```bash
curl -i -X POST -d '
"<edge_server><label>az_CDN_test</label><add_to_marketplace>true</add_to_marketplace><cpus>1</cpus><data_store_group_primary_id>2</data_store_group_primary_id><primary_network_group_id>3</primary_network_group_id><cpu_shares>1</cpu_shares><memory>2048</memory><required_virtual_machine_build>1</required_virtual_machine_build><hypervisor_group_id>1</hypervisor_group_id><hypervisor_id>1</hypervisor_id><required_ip_address_assignment>1</required_ip_address_assignment><primary_disk_size>20</primary_disk_size><rate_limit>0</rate_limit><user_group_id>2500</user_group_id><vdc_id>192</vdc_id><data_store_id>236</data_store_id><network_id>653</network_id><cdn_location_id>5</cdn_location_id><edge_server_type>http</edge_server_type><location_id>2</location_id></edge_server>' -u user:userpass
```

**JSON Request Example**

```bash
curl -i -X POST -d '{
"edge_server": {"label": "az_CDN_test", "add_to_marketplace": "true", "cpus": "1", "data_store_group_primary_id": "2", "primary_network_group_id": "3", "cpu_shares": "1" , "memory": "2048", "required_virtual_machine_build": "1", "hypervisor_group_id": "1", "hypervisor_id": "1", "required_ip_address_assignment": "1", "primary_disk_size": "20", "rate_limit": "0", "user_group_id": "2500", "vdc_id": "192", "data_store_id": "236", "network_id": "653", "cdn_location_id": "5", "edge_server_type": "http", "location_id": "2" }}' -u user:userpass http://onapp.test/edge_servers.json -H 'Accept: application/json' -H 'Content-type: application/json'
```
Where:

*label* – a unique name of your CDN edge server. The label can consist of letters [A-Za-z], digits [0-9], dash [-], lower dash [ _ ], space character [ ], at sign [@], round brackets [()], slashes [/], comma [,] and dot [.] You can use both lower- and uppercase letters. The label should begin with an alphanumeric character or lower dash [ _ ]

*hypervisor_id* - indicate the ID of the compute resource, on which the server will be deployed

*hypervisor_group_id* - indicate the compute zone ID

*cpus* - the amount of CPU cores allocated to this edge server

*cpu_shares* - the percentage of allocated CPU priority resource

*memory* - the amount of RAM, which you want to allocate to this edge server

*primary_disk_size* - the size in GB of the primary disk

*data_store_group_primary_id* – specify the ID of a data store zone, where you want to locate the disk of your server. If not specified – the system will select the data store zone with higher available capacity

*primary_network_group_id* – indicate the network zone ID

*required_virtual_machine_build* – set "1" to build the server automatically after creation. Otherwise set "0"

*required_ip_address_assignment* - set "1" if you want IP address to be assigned automatically after creation. Otherwise set "0"

*network_id* - the ID of the network to which the edge server will be connected

*selected_ip_address* - an IP address to assign to this container server; if the parameter *required_ip_address_assignment* was set "1" but this parameter *selected_ip_address* is empty - the first available IP address will be assigned to container server automatically

*add_to_marketplace* – set "true", if the edge server is added to marketplace; otherwise set "false". The default value is "false".

*cdn_location_id* - the ID of the CDN location. Use the following API call to find the ID [Get List of CDN Locations for Location Group](https://example.com). The parameter is optional. If not set, the edge server will be assigned to the first CDN Location in its Location Group.

*edge_server_type* - set http or streaming server type

*location_id* - the ID of the location group

ATTENTION! Creating a Streaming Edge or Storage server will result in an additional monthly charge. You will be charged 50$ per month for deploying this Streaming server once it is provisioned.

Parameters, specific to vCloud edge servers:

*user_group_id* - the ID of the organization to which the edge server will be associated

*vdc_id* - the ID of the resource pool to which the edge server will be connected

*data_store_id* - the ID of the data store, to which the edge server will be assigned

*network_id* - the ID of the network to which the edge server will be connected
v.6.0
- added the `location_id` parameter
- replaced `cdn_location` with `cdn_location_id` parameter

v.5.4
- added the following parameters:
  - for vCloud edge server creation:
    - `user_group_id`
    - `vdc_id`
    - `data_store_id`
    - `network_id`
  - `selected_ip_address`
  - `network_id`

v.3.3.1
- added the following parameters:
  - `cdn_location`
  - `edge_server_type`

4.10.4 Edit CDN Edge Servers
To change the server label and resource allocation:

PUT /edge_servers/:id.xml
PUT /edge_servers/:id.json

XML Request Example

```
curl -i -X PUT -d 
'\<edge_server\><label>az_CDN_test_1</label><add_to_marketplace>true</add_to_marketplace><cpus></cpus><cpu_shares>10</cpu_shares><memory>512</memory>\</edge_server>' -u onapp.test http://onapp.test/edge_servers/12.xml -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request Example

```
curl -i -X PUT -d 
'\"edge_server\":\"label\":\"az_CDN_test_3\",\"add_to_marketplace\":\"true\",\"cpus\":\"1\",\"cpu_shares\":\"20\",\"memory\":\"512\"\")\' -u onapp.test http://onapp.test/edge_servers/12.json -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:

`label` – a unique name of your CDN edge server. The label can consist of letters [A-Za-z], digits [0-9], dash [-], lower dash [_], space character [ ], at sign [@], round brackets [], slashes [ ], comma [,] and dot [.]. You can use both lower- and uppercase letters. The label should begin with an alphanumeric character or lower dash [_].

`add_to_marketplace` – set "true", if the edge server is added to marketplace; otherwise set "false". The default value is "false"

`cpus` - the amount of CPU cores allocated to this edge server
**cpu_shares** - the percentage of allocated CPU priority resource

**memory** - the amount of RAM, which you want to allocate to this edge server

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no edge server with a requested ID, or URL is incorrect.

### 4.10.5 Reboot CDN Edge Server

To reboot the edge server, use the following request:

POST /edge_servers/:edge_server_id/reboot.xml

POST /edge_servers/:edge_server_id/reboot.json

**XML Request Example**

```sh
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/reboot.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**JSON Request Example**

```sh
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/reboot.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```

### 4.10.6 Reboot CDN Edge Server in Recovery

To reboot the edge server in recovery mode with a temporary login ("root") and password ("recovery"), use the following request:

POST /edge_servers/:edge_server_id/reboot.xml

POST /edge_servers/:edge_server_id/reboot.json

**XML Request Example**

```sh
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/reboot.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```  

**JSON Request Example**

```sh
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/reboot.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```

### 4.10.7 Start up CDN Edge Server

POST /edge_servers/:edge_server_id/startup.xml

POST /edge_servers/:edge_server_id/startup.json

**XML Request Example**
**4.10.8 Shut down CDN Edge Server**

To terminate the edge server gracefully, use the following request:

```bash
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/shutdown.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**XML Request Example**

```bash
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/shutdown.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**JSON Request Example**

```bash
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/shutdown.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```

**4.10.9 Stop CDN Edge Server**

To terminate the edge server forcefully, use the following request:

```bash
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/stop.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**XML Request Example**

```bash
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/stop.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**JSON Request Example**

```bash
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/stop.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```
4.10.10  Rebuild CDN Edge Server

To rebuild (or build manually) the edge server, use the following request:

POST /edge_servers/:edge_server_id/build.xml
POST /edge_servers/:edge_server_id/build.json

**XML Request Example**

curl -i -X POST -u user:userpass  
http://onapp.test/edge_servers/12/build.xml  
d'<edge_server><template_id>398</template_id><required_startup>1</required_startup></edge_server>'  
-H 'Accept: application/xml'  
-H 'Content-type: application/xml'

**JSON Request Example**

curl -i -X POST -u user:userpass  
http://onapp.test/edge_servers/12/build.json  
-d '{"edge_server":{"template_id":"398","required_startup":"1"}}'  
-H 'Accept: application/json'  
-H 'Content-type: application/json'

Where:
- **template_id** * - the ID of the template on which this server will be based
- **required_startup** – set "1" to start up the server automatically after build, otherwise set "0"

4.10.11  Suspend CDN Edge Server

To suspend a CDN edge server, use the following request:

POST /edge_servers/:edge_server_id/suspend.xml
POST /edge_servers/:edge_server_id/suspend.json

**XML Request Example**

curl -i -X POST -u user:userpass  
http://onapp.test/edge_servers/12/suspend.xml  
-H 'Accept: application/xml'  
-H 'Content-type: application/xml'

**JSON Request Example**

curl -i -X POST -u user:userpass  
http://onapp.test/edge_servers/12/suspend.json  
-H 'Accept: application/json'  
-H 'Content-type: application/json'

To unsuspend the server, run the request again.

4.10.12  Rerun CDN Edge Server Creation Scripts

When an edge server is built, the system will run the scripts for creation of an edge server. To rerun the CDN edge server creation scripts, use the following request:

POST /edge_servers/:edge_server_id/rerun_creation_scripts.xml
POST /edge_servers/:edge_server_id/rerun_creation_scripts.json
XML Request Example

```
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/rerun_creation_scripts.xml -H
'Accept:application/xml' -H 'Content-type:application/xml'
```

JSON Request Example

```
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/rerun_creation_scripts.json -H
'Accept:application/json' -H 'Content-type:application/json'
```

4.10.13 Unlock CDN Edge Server

To unlock the edge server, use the following request:

POST /edge_servers/:edge_server_id/unlock.xml
POST /edge_servers/:edge_server_id/unlock.json

XML Request Example

```
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/unlock.xml -H
'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request Example

```
curl -i -X POST -u user:userpass
http://onapp.test/edge_servers/12/unlock.json -H
'Accept: application/json' -H 'Content-type: application/json'
```

4.10.14 Delete CDN Edge Servers

To delete a CDN edge server, use the following request:

DELETE /edge_servers/:id.xml
DELETE /edge_servers/:id.json

XML Request Example

```
curl -i -X DELETE -u user:userpass http://onapp.test/edge_servers/12.xml -H
'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request Example

```
curl -i -X DELETE -u user:userpass http://onapp.test/edge_servers/12.json
-H 'Accept: application/json' -H 'Content-type: application/json'
```

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no edge server with a requested ID, or URL is incorrect.
4.10.15  Migrate CDN Edge Server

To migrate an edge server to another compute resource, use the following request:

**POST /edge_servers/:edge_server_id/migrate.xml**
**POST /edge_servers/:edge_server_id/migrate.json**

**XML Request Example**

```bash
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d "<edge_server><destination>1</destination><cold_migrate_on_rollback>1</cold_migrate_on_rollback></edge_server>" --url http://onapp.test/edge_servers/12/migrate.xml
```

**JSON Request Example**

```bash
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"edge_server":{"destination":"1","cold_migrate_on_rollback":"1"}}' --url http://onapp.test/edge_servers/12/migrate.json
```

Where:

- **destination** - the ID of a target compute resource, to which you migrate the edge server
- **cold_migrate_on_rollback** - set 1 if you wish to switch to a cold migration if hot migration fails, otherwise set 0.

4.10.16  Segregate CDN Edge Server

To segregate an edge server (that is, instruct it never to reside on the same compute resource as another VS or edge server), use the following request:

**POST /edge_servers/:edge_server_id/strict_vm.xml**
**POST /edge_servers/:edge_server_id/strict_vm.json**

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"virtual_machine":{"strict_virtual_machine_id":"gv03xz1x31t53h"}}' --url http://onapp.test/edge_servers/12/strict_vm.json
```

Where:

- **strict_virtual_machine_id** - the ID of virtual server you wish to segregate from the given edge server
4.10.17  Change CDN Edge Server Owner

To reassign an edge server to another user, use the following request:

POST /edge_servers/:edge_server_id/change_owner.xml
POST /edge_servers/:edge_server_id/change_owner.json

XML Request Example

```bash
```

JSON Request Example

```bash
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"user_id":1}'' --url http://onapp.test/edge_servers/12/change_owner.json
```

Where:
user_id* – input ID of a new server owner

4.10.18  Set VIP Status for CDN Edge Server

To give your edge server a migration priority, set the VIP status for it with the following request:

POST /edge_servers/:edge_server_id/set_vip.xml
POST /edge_servers/:edge_server_id/set_vip.json

XML Request Example

```bash
```

JSON Request Example

```bash
```

Where:
vip* - whether VIP status is enabled for the server or not. Set this parameter to 'true' to enable and to 'false' to disable the VIP status.

4.10.19  Add/Edit Admin/User Note for CDN Edge Server

To edit/make an admin note, use the following request:

PUT /edge_servers/:edge_server_id.xml
PUT /edge_servers/:edge_server_id.json

XML Request Example
To edit/make a user note, use the following request:

**XML Request Example**

curl -i -X PUT -u user:userpass http://onapp.test/edge_servers/12.xml -d
'<?xml version="1.0" encoding="utf-8"?>
<edge_server>
<admin_note>agfa tjuuytjgh yuytu</admin_note>
</edge_server>' -H 'Accept:application/xml' -H 'Content-type:application/xml'

**JSON Request Example**

curl -i -X PUT -u user:userpass http://onapp.test/edge_servers/12.json -d
'{"edge_server":{"admin_note":"kjfjhjtrtjt"}}' -H
'Accept:application/json' -H 'Content-type:application/json'

Where:

*admin_note* – enter the text of your note.

To view the edge server disks, use the following request:

**XML Request Example**

curl -i -X GET -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url
http://onapp.test/edge_servers/12/disks.xml

**JSON Request Example**

curl -i -X GET -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url
http://onapp.test/edge_servers/12/disks.json

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no edge server with a requested ID, or URL is incorrect.

### 4.10.20 CDN Edge Server Disks

Since CDN edge servers are virtual servers in their essence, you may perform all the same actions with edge servers’ disks as with VS disks, except POST and DELETE. The only difference would be in the routes for the following requests:

To view the edge server disks, use the following request:

GET /edge_servers/:edge_server_id/disks.xml
GET /edge_servers/:edge_server_id/disks.json

**XML Request Example**

curl -i -X GET -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url
http://onapp.test/edge_servers/12/disks.xml

**JSON Request Example**

curl -i -X GET -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url
http://onapp.test/edge_servers/12/disks.json

Parameters description and output example.

For other possible requests, refer to corresponding sections of Disks chapter.

4.10.21 CDN Edge Server Network Interfaces

Here is the list of API calls for managing CDN edge servers' network interfaces. Edge servers' network interfaces have the same attributes as network interfaces of virtual servers.

To get the list of network interfaces allocated to this particular edge server, use the following request:

GET /edge_servers/:edge_server_id/network_interfaces.xml
GET /edge_servers/:edge_server_id/network_interfaces.json

To get a particular network interface details, use the following request:

GET /edge_servers/:edge_server_id/network_interfaces/:id.xml
GET /edge_servers/:edge_server_id/network_interfaces/:id.json

To edit network interface details, use the following request:

PUT /edge_servers/:edge_server_id/network_interfaces/:id.xml
PUT /edge_servers/:edge_server_id/network_interfaces/:id.json

To add a new network interface, use the following request:

POST /edge_servers/:edge_server_id/network_interfaces.xml
POST /edge_servers/:edge_server_id/network_interfaces.json

To delete a network interface from the edge server, use the following request:

DELETE /edge_servers/:edge_server_id/network_interfaces/:id.xml
DELETE /edge_servers/:edge_server_id/network_interfaces/:id.json

XML Output Example

```xml
<network_interface>
  <label>eth0</label>
  <usage nil="true"></usage>
  <created_at type="datetime">2011-03-18T17:45:07+07:00</created_at>
  <updated_at type="datetime">2011-04-08T18:57:20+07:00</updated_at>
  <primary type="boolean">true</primary>
  <usage_month_rolled_at nil="true"></usage_month_rolled_at>
  <id type="integer">502</id>
  <mac_address>00:16:3e:50:35:52</mac_address>
  <usage_last_reset_at nil="true"></usage_last_reset_at>
  <default_firewall_rule>DROP</default_firewall_rule>
  <rate_limit type="integer">0</rate_limit>
  <virtual_machine_id type="integer">518</virtual_machine_id>
  <network_join_id type="integer">4</network_join_id>
  <identifier>pdfjrtpkday9e1</identifier>
</network_interface>
```

Where:

- **label** - network interface name
- **created_at** - the timestamp in the database when this network interface was created
4.10.22 CDN Edge Server IP Address Joins

An IP address allocated to an edge server is an IP address join. Use the following methods to view, assign and delete IP address joins of your CDN edge servers.

To get the list of IP address assignments for a particular edge server, use the following request:

GET /edge_servers/:edge_server_id/ip_addresses.xml
GET /edge_servers/:edge_server_id/ip_addresses.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

To assign an IP Address to an edge server, use the following request:

POST /edge_servers/:edge_server_id/ip_addresses.xml
POST /edge_servers/:edge_server_id/ip_addresses.json

**XML Request Example**

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d '<ip_address_join><ip_address_id>7</ip_address_id><network_interface_id>131</network_interface_id></ip_address_join>' --url http://onapp.test/edge_servers/12/ip_addresses.xml
```

**JSON Request Example**

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"ip_address_join":{"ip_address_id":7,"network_interface_id":131}}' --url http://onapp.test/edge_servers/12/ip_addresses.json
```
XML Output Example

```xml
<ip_address_join>
  <created_at type="datetime">2013-10-31T13:04:05+03:00</created_at>
  <id type="integer">173</id>

  <ip_address_id type="integer">7</ip_address_id>
  <network_interface_id type="integer">131</network_interface_id>
  <updated_at type="datetime">2013-10-31T13:04:05+03:00</updated_at>

  <ip_address>
    <address>1.1.1.3</address>
    <broadcast>1.1.1.255</broadcast>
    <created_at type="datetime">2013-08-07T13:29:09+03:00</created_at>
    <customer_network_id nil="true"/>

    <disallowed_primary type="boolean">false</disallowed_primary>
    <gateway>1.1.1.1</gateway>
    <hypervisor_id nil="true"/>
    <id type="integer">7</id>

    <ip_address_pool_id nil="true"/>
    <network_address>1.1.1.0</network_address>
    <network_id type="integer">1</network_id>
    <pxe type="boolean">false</pxe>

    <updated_at type="datetime">2013-08-07T13:29:09+03:00</updated_at>
    <user_id nil="true"/>
    <free type="boolean">false</free>
    <netmask>255.255.255.0</netmask>
  </ip_address>
</ip_address_join>
```

Where:
- `created_at` - the date when the record was created in DB
- `id` - the IP address join ID
- `ip_address_id` - the IP address ID
- `network_interface_id` - the network interface ID
- `updated_at` - the date when the record was updated in DB
- `ip_address` - the array of IP address details
- `address` - the IP address
- `broadcast` - a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.
- `customer_network_id` - the ID of the customer network
- `disallowed_primary` - true if this address is not set as primary (for VS build), otherwise false
- `gateway` - gateway address
- `hypervisor_id` - the ID of the compute resource
- `ip_address_pool_id` - the ID of the IP address poll to which this join belongs
- `network_address` - the address of a VLAN network address that will be associated with this IP address pool
- `network_id` - the ID of the network
- `pxe` - true, if this address can be used for cloudbooting a compute resource
free - true if free, otherwise false

netmask — netmask for the IP address

To delete an IP address assignment from a particular edge server, use the following request:

DELETEnedge_servers/:edge_server_id/ip_addresses/:id.xml
DELETEnedge_servers/:edge_server_id/ip_addresses/:id.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where:

data_store_id - the ID of the data store, which is attached to the compute resource

hypervisor_id - reserved parameter

id - the join ID

target_join_id - the ID of the join target; in this case it is the compute resource ID

target_join_type - type of join target; in this case it is compute resource

### 4.10.23 Rebuild Network for CDN Edge Server

It is required to rebuild network after any changes on IP address joins or network interfaces. To rebuild network, use the following request:

POST /edge_servers/:edge_server_id/rebuild_network.xml
POST /edge_servers/:edge_server_id/rebuild_network.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where:

storage_server_id - ID of the edge server
4.10.24 Get CDN Edge Server Billing Statistics

To view the billing statistics for a particular edge server, use the following request:

GET /edge_servers/:edge_server_id/vm_stats.xml
GET /edge_servers/:edge_server_id/vm_stats.json

Define a shorter period by setting Start and End time in the API call:


XML Output Example
<vm_stats type="array">
  <vm_hourly_stat>
    <created_at type="datetime">2011-11-01T00:00:13Z</created_at>
    <updated_at type="datetime">2011-11-01T00:00:13Z</updated_at>
    <stat_time type="datetime">2011-11-01T00:00:02Z</stat_time>
    <total_cost type="float">0.0</total_cost>
    <id type="integer">9582</id>
    <vm_billing_stat_id type="integer">7795</vm_billing_stat_id>
    <user_id type="integer">1</user_id>
    <billing_stats>
      <virtual_machines type="array">
        <virtual_machine>
          <label>QAVP XEN serveridze</label>
          <costs type="array">
            <cost>
              <value type="integer">100</value>
              <resource_name>cpu_shares</resource_name>
              <cost type="float">0.0</cost>
            </cost>
          </costs>
          <id type="integer">237</id>
        </virtual_machine>
        <virtual_machines type="array">
          <network_interfaces type="array">
            <network_interface>
              <label>eth0</label>
              <costs type="array">
                <cost>
                  <value type="integer">1</value>
                  <resource_name>ip_addresses</resource_name>
                  <cost type="float">0.0</cost>
                </cost>
              </costs>
              <id type="integer">254</id>
            </network_interface>
            <disks type="array">
              <disk>
                <label>#499</label>
                <costs type="array">
                  <cost>
                    <value type="integer">20</value>
                    <resource_name>disk_size</resource_name>
                    <cost type="float">0.0</cost>
                  </cost>
                </costs>
                <id type="integer">499</id>
              </disk>
            </disks>
          </network_interfaces>
          <edge_servers type="array">
            <edge_server>
              <label>QAVP XEN serveridze</label>
              <costs type="array">
                <cost>
                  <value type="integer">14</value>
                  <resource_name>template</resource_name>
                  <cost type="float">0.0</cost>
                </cost>
              </costs>
            </edge_server>
          </edge_servers>
        </virtual_machines>
      </virtual_machines>
    </billing_stats>
  </vm_hourly_stat>
</vm_stats>

Where:

created_at – the timestamp in DB when this record was created
updated_at – the date when these statistics were updated
cost – the total amount of money owed by this particular edge server for the resources spent at stat_time
stat_time – the particular hour for which these statistics were generated
id – the ID of these statistics
user_id - the ID of edge server owner
currency_code - currency in which this virtual machine is charged within the bucket
billing_stats - an array of billing details for the resources used by this edge server
virtual_machine - an array of edge server billing details:
label – name of the edge server
costs- an array of edge server resources with their total prices for the period specified in the stat-time parameter, where:
  - resource_name - the resource in question. This can be cpu_shares, cpus, memory, cpu_usage and template
  - value - the amount of resources allocated to this edge server. Here are the units of measurement for each type of resource_name:
    - cpu_shares - CPU priority percentage
    - cpus - number of CPU cores
    - memory - amount of RAM in Mb
    - cpu_usage - CPU time in seconds
  - cost - the total due for this resource
  - id - Virtual machine ID

network_interfaces - an array of network interfaces used by this edge server with their billing statistics:
  - label - network interface name used in OnApp
  - id - network interface ID
  - costs- an array of network interface related resources with their total prices for the period specified in the stat-time parameter, where:
  - resource_name - the resource in question. This can be ip_addresses, rate, data_received and data_sent
- **value**: the amount of resources used by this network interface. Here are the units of measurement for each type of `resource_name`:
  - `ip_addresses`: number of IPs
  - `rate`: the port speed in Mb per second
  - `data_received`: amount of received data in Kb
  - `data_sent`: amount of sent data in Kb

- **cost**: the total due for the resource

- **disks**: an array of disks used by this edge server with their billing details:
  - `label`: disk name used in UI
  - `id`: disk ID used in database
  - `costs`: an array of disk related resources with their total prices for the period specified in the `stat_time` parameter, where:
    - `resource_name`: the resource in question. This can be `disk_size`, `data_read`, `data_written`, `reads_completed` and `writes_completed`
    - `value`: the amount of resources used. Here are the units of measurement for each type of `resource_name`:
      - `disk_size`: size in GB
      - `data_read`: read data in Kb
      - `data_written`: amount of written data in Kb
      - `reads`: number read operations
      - `writes`: number of write operations
    - `cost`: the total due for the resource

- **edge_server**: an array of edge server with its billing details:
  - `label`: edge server name used in UI
  - `id`: server ID used in database
  - `costs`: an array of related resources with their total prices for the period specified in the `stat_time` parameter, where:
    - `resource_name`: the resource in question. In this case - `template`
    - `value`: here, the template ID in the database
    - `cost`: the total due for the resource

- **total_cost**: the total amount of money owed for the edge server specified by `id` parameter for a particular hour specified by `stat_time` parameter (`total_cost = vm_resources_cost + usage_cost`)
  - **vm_resources_cost**: the amount of money due for the edge server resources for the particular hour specified by `stat_time` parameter (memory, disks, templates)
  - **usage_cost**: the total due for edge server usage for this particular hour specified by `stat_time` parameter (data sent/received, bandwidth, CPU usage).

### 4.10.25 Get CDN Edge Server CPU Usage Statistics

To view CPU usage statistics of a CDN edge server, use the following request:

GET /edge_servers/:edge_server_id/cpu_usage.xml
GET /edge_servers/:edge_server_id/cpu_usage.json
Define a shorter period by setting Start and End time in the API call:

XML Request Example

```
curl -i -u user:userpass --url http://onapp.test/edge_servers/12/cpu_usage.xml
```

XML Request Example

```
curl -i -u user:userpass --url http://onapp.test/edge_servers/12/cpu_usage.json
```

Where you have to specify the edge server ID.

**4.10.26 Search CDN Edge Server by Label**

To search an edge server by label, use the following request:

GET /edge_servers.xml?q=label
GET /edge_servers.json?q=label

XML Request Example

```
```

JSON Request Example

```
```

Where you have to specify the label of a CDN edge server you are searching for.

**4.11 CDN HTTP Caching Rules API**

This chapter includes API request for managing HTTP caching rules. The HTTP Rules engine allows users to customize the CDN edge server behavior, e.g. how the CDN will manage cache and redirection.
A CDN resource can be configured with up to 100 rules. This is collectively called a ruleset. A rule consists of conditions and actions. A condition consists of a subject, which determines the value to select, and a predicate, which specifies what to compare the subject against. Conditions are bonded by the connectives “AND” or “OR”. When all the conditions are met, the CDN edge server will perform the actions associated with the rule.

4.11.1 Get List of HTTP Caching Rules

To view the list of HTTP caching rules for a CDN resource, use the following request:

GET /cdn_resources/:cdn_resource_id/http_caching_rules.xml
GET /cdn_resources/:cdn_resource_id/http_caching_rules.json

XML Request Example

```
```

JSON Request Example

```
```

XML Output Example

```
<objects type="array">
 <object>
   <name>2</name>
   <conditions type="array">
     <condition>
       <subject>url</subject>
       <predicate>ends with</predicate>
       <value>gif</value>
     </condition>
   </conditions>
   <actions type="array">
     <action>
       <act>forbid client</act>
     </action>
   </actions>
 </object>
<object>...
</objects>
```

Where:

- **name** - the name of the rule
- **conditions** - the array of parameters of the conditions associated with the rule
- **subject** - the subject of the condition. For the list of subjects you can set for a rule refer to The List of Subjects.
- **predicate** - the predicate of the condition. For the list of predicates you can set for a rule refer to The List of Predicates.
- **value** - the value against which the subject is compared.
4.11.2 Add HTTP Caching Rule

To add an HTTP caching rule, use the following request:

POST /cdn_resources/:cdn_resource_id/http_caching_rules.xml
POST /cdn_resources/:cdn_resource_id/http_caching_rules.json

XML Request Example

curl -i -X POST -u user:userpass -H 'Accept: application/xml' -H 'Content-type: application/xml' --url 'http://onapp.test/cdn_resources/12/http_caching_rules.xml' -d '<rule><name>some_name</name><conditions><0><connective>if</connective><subject>url</subject><predicate>default</predicate><value></value></0><1><connective>and</connective><subject>cookie</subject><cookie></cookie><predicate>default</predicate><value></value></1><2><connective>and</connective><subject>param</subject><param></param><predicate>default</predicate><value></value></2><3><connective>and</connective><subject>header</subject><header></header><predicate>default</predicate><value></value></3></conditions><actions><0><act>force edge to cache</act><seconds></seconds></0><1><act>redirect client</act><url></url></1><2><act>set response header</act><header></header><value></value></2><3><act>set custom origin</act><value></value></3></actions></rule>'

JSON Request Example

curl -i -X POST -u user:userpass -H 'Accept: application/json' -H 'Content-type: application/json' --url 'http://onapp.test/cdn_resources/12/http_caching_rules.json' -d '{"rule": {"name":"some_name", "conditions": {"0": {"connective": "if", "subject": "url", "predicate": "default", "value": ""}, "1": {"connective": "and", "subject": "cookie", "cookie": "", "predicate": "default", "value": ""}, "2": {"connective": "and", "subject": "param", "param": "", "predicate": "default", "value": ""}, "3": {"connective": "and", "subject": "header", "header": "", "predicate": "default", "value": ""}}, "actions": {"0": {"act": "force edge to cache", "seconds": ""}, "1": {"act": "redirect client", "url": ""}, "2": {"act": "set response header", "header": "", "value": ""}, "3": {"act": "set custom origin", "value": ""}}}'

Where:

- name - name of the rule
- conditions - the array of parameters of the conditions associated with the rule
- connective - the connective by which the conditions are bonded, either 'and' or 'or'.
- subject - the subject of the condition. The subject should be written using small letters only and with spaces between the words. For the list of subjects you can set for a rule refer to The List of Subjects.
- predicate - the predicate of the condition. The predicate should be written using small letters only and with spaces between the words. For the list of predicates you can set for a rule refer to The List of Predicates.
- value - the value against which the subject is compared.
header - the subject that selects the value of a specific client request header. If the request header does not exist, then the value "" is selected.

act - the action associated with the rule. The action should be written using small letters only and with spaces between the words. For the list of actions you can set for a rule refer to The List of Actions.

url - the subject that selects the URL part of the request. It excludes the query string.

cookie - the subject that selects the value of a specific cookie sent by the client.

param - the subject that selects the value of a specific query string parameter. If there are multiple identical keys, the last value is selected.

seconds - the time in seconds set for the action. For more information, refer to The List of Actions.

4.11.3 Edit HTTP Caching Rule

Use the following request to edit an HTTP caching rule:

PUT /cdn_resources/:cdn_resource_id/http_caching_rules.xml

PUT /cdn_resources/:cdn_resource_id/http_caching_rules.json

**XML Request Example**

```
  <id>1</id><rule><name>name</name><conditions><0><connective>if</connective><subject>url</subject><predicate>default</predicate><value></value></0><1><connective>and</connective><subject>cookie</subject><cookie></cookie><predicate>default</predicate><value></value></1><2><connective>and</connective><subject>param</subject><param></param><predicate>default</predicate><value></value></2><3><connective>and</connective><subject>header</subject><header></header><predicate>default</predicate><value></value></3></conditions><actions><0><act>force edge to cache</act><seconds></seconds></0><1><act>redirect client</act><url></url></1><2><act>set response header</act><header></header><value></value></2><3><act>set custom origin</act><value></value></3></actions></rule>
```

**JSON Request Example**

```
  {"id":1,"rule": {"name": "name", "conditions": {"0": {"connective": "if", "subject": "url", "predicate": "default", "value": ","}, "1": {"connective": "and", "subject": "cookie", "cookie": ",", "predicate": "default", "value": ","}, "2": {"connective": "and", "subject": "param", "param": ",", "predicate": "default", "value": ","}, "3": {"connective": "and", "subject": "header", "header": ",", "predicate": "default", "value": ","}, "actions": [{"act": "force edge to cache", "seconds": ","}, {"act": "redirect client", "url": ","}, {"act": "set response header", "header": ",", "value": ","}, {"act": "set custom origin", "value": ","}]}}
```

Where:

id - the ID of the HTTP caching rule you want to edit

name - name of the rule
**conditions** - the array of parameters of the conditions associated with the rule

**connective** - the connective by which the conditions are bonded, either 'and' or 'or'.

**subject** - the subject of the condition. The subject should be written using small letters only and with spaces between the words. For the list of subjects you can set for a rule refer to [The List of Subjects](#).

**predicate** - the predicate of the condition. The predicate should be written using small letters only and with spaces between the words. For the list of predicates you can set for a rule refer to [The List of Predicates](#).

**value** - the value against which the subject is compared.

**cookie** - the subject that selects the value of a specific cookie sent by the client.

**param** - the subject that selects the value of a specific query string parameter. If there are multiple identical keys, the last value is selected.

**header** - the subject that selects the value of a specific client request header. If the request header does not exist, then the value "" is selected.

**act** - the action associated with the rule. The action should be written using small letters only and with spaces between the words. For the list of actions you can set for a rule refer to [The List of Actions](#).

**seconds** - the time in seconds set for the action. For more information, refer to [The List of Actions](#).

**url** - the subject that selects the URL part of the request. It excludes the query string.

### 4.11.4 Delete HTTP Caching Rule

To delete an HTTP caching rule, use the following request:

```
DELETE /cdn_resources/:cdn_resource_id/http_caching_rules/:rule_id.xml
```

```
DELETE /cdn_resources/:cdn_resource_id/http_caching_rules/:rule_id.json
```

**XML Request Example**

```
```

**JSON Request Example**

```
```

### 4.12 CDN Locations API

This section contains the API requests you can use to manage your CDN locations.

#### 4.12.1 Get List of CDN Locations for Location Group

To get an array of CDN locations set up within your cloud for a particular Location group, use the following request:
GET /settings/location_groups/:location_group_id/cdn_locations.xml
GET /settings/location_groups/:location_group_id/cdn_locations.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

**XML Output Example**

```
<cdn_locations type="array">
<cdn_location>
  <city>Mexicali</city>
  <country>Mexico</country>
  <created_at type="datetime">2014-09-22T16:30:36+03:00</created_at>
  <id type="integer">1</id>
  <location_group_id type="integer">1</location_group_id>
  <updated_at type="datetime">2014-09-23T16:46:35+03:00</updated_at>
  <cdn_reference type="integer">331</cdn_reference>
</cdn_location>
</cdn_locations>
```

**Where:**

- **city** - the CDN location city
- **country** - the CDN location country
- **created_at** - the date in the [YYYY][MM][DD][T][hh][mm][ss]Z format
- **id** - the CDN location ID
- **updated_at** - the date when the location group was updated in the [YYYY][MM][DD][T][hh][mm][ss]Z format
- **cdn_reference** - the location ID in OnApp Dashboard

### 4.12.2 Get List of CDN Locations

To get an array of CDN locations set up within your cloud, use the following request:

GET /settings/cdn_locations.xml
GET /settings/cdn_locations.json

**XML Request Example**

JSON Request Example


XML Output Example

```xml
<cdn_locations type="array">
  <cdn_location>
    <city>Mexicali</city>
    <country>Mexico</country>
    <created_at type="datetime">2014-09-22T16:30:36+03:00</created_at>
    <id type="integer">1</id>
    <location_group_id type="integer">1</location_group_id>
    <updated_at type="datetime">2014-09-23T16:46:35+03:00</updated_at>
    <cdn_reference type="integer">331</cdn_reference>
  </cdn_location>
</cdn_locations>
```

Where:

city - the CDN location city

country - the CDN location country

created_at - the date in the [YYYY][MM][DD][hh][mm][ss]Z format

id - the CDN location ID

updated_at - the date when the location group was updated in the [YYYY][MM][DD][hh][mm][ss]Z format

cdn_reference - the location ID in OnApp Dashboard

location_group_id - the ID of the location group to which this CDN location is attached

4.12.3 Detach CDN Location from Location Group

To detach a CDN location from a location group, use the following request:

POST /settings/location_groups/:location_group_id/cdn_locations/:cdn_location_id/detach_resource.xml

POST /settings/location_groups/:location_group_id/cdn_locations/:cdn_location_id/detach_resource.json

XML Request Example
4.12.4 Attach CDN Location to Location Group

To attach a CDN location to a location group, use the following request:

**XML Request Example**

```bash
```

This request attaches a particular CDN location (*attach_resource_id*) to a specific location group (:location_group_id)

4.13 CDN Reporting API

CDN reporting functionality allows you to conduct the in-depth analysis of your own CDN resources by viewing different reports.
4.13.1 Get Top Files Report

To get the Top 50 files report, use the following request:

GET /cdn/reports/top_files.xml
GET /cdn/reports/top_files.json

XML Request Example

curl -l -X GET -u user:userpass --url http://onapp.test/cdn/reports/top_files.xml -d '⟨top_files⟩⟨start_date⟩2016-11-09⟩⟨end_date⟩2016-11-10⟩⟨entity_id⟩945986057⟩⟨/top_files⟩' -H 'Accept: application/xml' -H 'Content-type: application/xml'

JSON Request Example


Where:

Define a time period by setting the following parameters:

- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

Get statistics for a particular CDN resource by setting the following parameters:

- **entity_id** - the ID of the CDN resource

XML Output Example

```xml
<top_files>
  <top_fifty_files_table type="array">
    <top_fifty_files_table>
      <resourceId type="integer">422564898</resourceId>
      <fileUrl>/test.gif</fileUrl>
      <request type="float">113.0</request>
      <hit type="float">110.0</hit>
      <miss type="float">3.0</miss>
      <bandwidth type="float">3781649.0</bandwidth>
    </top_fifty_files_table>
  </top_fifty_files_table>
</top_files>
```

Where:

- **resourceId** - the ID of the CDN resource
- **fileUrl** - the URL of the resource file
- **request** - the total amount of file requests for the selected period
- **hit** - the amount of successful file requests for the selected period
- **miss** - the amount of failed file requests for the selected period
- **bandwidth** - the amount of transmitted bandwidth for the selected period
4.13.2 Purge CDN Resource in Top Files Report

To remove content from cache, use the following request:

POST /cdn/reports/top_files/purge.xml
POST /cdn/reports/top_files/purge.json

**XML Request Example**

```bash
curl -i -X POST -u user:userpass
http://onapp.test/cdn/reports/top_files/purge.xml -d
'<root><remote_id>945986057</remote_id><url>/</url></root>' -H 'Accept:application/xml' -H 'Content-type:application/xml'
```

**JSON Request Example**

```bash
curl -i -X POST -u user:userpass
```

Where:
- **remote_id** – the ID of the CDN resource
- **url** – the URL of the resource file

4.13.3 Get Top Referrers Report

This report is available only for resource owner.

To get the Top 50 referrers report, use the following request:

GET /cdn/reports/top_referrers.xml
GET /cdn/reports/top_referrers.json

**XML Request Example**

```bash
curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/top_referrers.xml -d
'<?xml version="1.0" encoding="UTF-8"?>
<top_referrers><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date><entity_id>945986057</entity_id></top_referrers>' -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/top_referrers.json -d '{"top_referrers": {"start_date": "2016-11-09", "end_date": "2016-11-10", "entity_id": "945986057"}}' -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:
Define a time period by setting the following parameters:

- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

Get statistics for a particular CDN resource by setting the following parameters:

- **entity_id** - the ID of the CDN resource

**XML Output Example**

```xml
<top_referrers>
  <top_fifty_referrers_table type="array">
    <top_fifty_referrers_table>
      <resourceId type="integer">422564898</resourceId>
      <referrer>/test.gif</referrer>
      <hit type="float">110.0</hit>
    </top_fifty_referrers_table>
  </top_fifty_referrers_table>
</top_referrers>
```

Where:

- **resourceId** - the ID of the CDN resource
- **referrer** - the referrer link
- **hit** - the amount of references for the selected period

### 4.13.4 Get CDN Overview Report

To get the CDN Overview report, use the following request:

**GET /cdn/reports/overview.xml**

**GET /cdn/reports/overview.json**

**XML Request Example**

```
curl -i -X GET -u user:userpass --url http://onapp.test/cdn/reports/overview.xml -d 'subscriptionDate=2016-11-09
start_date=2016-11-09
end_date=2016-11-10' -H 'Accept: application/xml' -H 'Content-Type: application/xml'
```

**JSON Request Example**

```
curl -i -X GET -u user:userpass --url http://onapp.test/cdn/reports/overview.json -d '{"overview":{"frequency":"2","start_date": "2016-11-09", "end_date": "2016-11-10"}}' -H 'Accept: application/json' -H 'Content-Type: application/json'
```

Where:

Define statistics filters using the following parameters:

- **frequency** - set statistics frequency using the following values: 0- one minute, 1- one hour or 2- one day.
- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

**XML Output Example**

```xml
<overview>
  <cache_statistic_line_chart type="array">
    <cache_statistic_line_chart>
      <time>2016-12-09</time>
      <cached type="float">533803623.0</cached>
      <uncached type="float">112393003.0</uncached>
      <hit type="float">0.0</hit>
      <miss type="float">0.0</miss>
      <gb type="float">0.0</gb>
    </cache_statistic_line_chart>
  </cache_statistic_line_chart>
  <overview_top_five_http_error_codes_table type="array">
    <resourceId type="integer">741579723</resourceId>
    <errorRequest type="float">614.0</errorRequest>
  </overview_top_five_http_error_codes_table>
  <overview_top_five_resources_table type="array">
    <resourceId type="integer">741579723</resourceId>
    <bandwidth type="float">866516552.0</bandwidth>
    <cacheHit type="float">744.0</cacheHit>
    <miss type="float">508.0</miss>
  </overview_top_five_resources_table>
  <overview_top_five_visitor_locations_pie_chart type="array">
    <country>GB</country>
    <request type="float">1493.0</request>
  </overview_top_five_visitor_locations_pie_chart>
</overview>
```

**Where:**

- **cache_statistic_line_chart** - an array of cache statistics chart details:
  - **time** - the selected time period
  - **cached** - the amount of cached bandwidth
  - **uncached** - the amount of uncached bandwidth
  - **hit** - the amount of successful file requests for the selected period
  - **miss** - the amount of failed file requests for the selected period
  - **gb** - the amount of transmitted bandwidth for the selected period

- **overview_top_five_http_error_codes_table** - an array of top five error codes table details:
  - **resourceId** - the ID of CDN resource
  - **errorRequest** - the amount of error requests

- **overview_top_five_resources_table** - an array of top five CDN resources table details:
  - **resourceId** - the ID of CDN resource
  - **bandwidth** - the amount of transmitted bandwidth
  - **cacheHit** - the amount of successful file requests
  - **miss** - the amount of failed file requests

- **overview_top_five_visitor_locations_pie_chart** - an array of top five visitor locations pie chart details:
  - **country** - the country, where visitors are located
request - the amount of requests from the corresponding country

4.13.5 Get Cache Statistics Report

To get the Cache Statistics report, use the following request:

GET /cdn/reports/cache_statistics.xml
GET /cdn/reports/cache_statistics.json

XML Request Example

```
curl -i -X GET -u user:userpass --url http://onapp.test/cdn/reports/cache_statistics.xml -d '
  <cache_statistics><frequency>2</frequency><filter_type>1</filter_type><entity_id><entity_id></entity_id><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date></cache_statistics>' -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request Example

```
curl -i -X GET -u user:userpass --url http://onapp.test/cdn/reports/cache_statistics.json -d '
  {"cache_statistics":
    {"frequency":"2","filter_type":"1","entity_id":"","start_date": "2016-11-09", "end_date": "2016-11-10"}}' -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:

Define statistics filters using the following parameters:

- **frequency** - set statistics frequency using the following values: 0 - one minute, 1 - one hour or 2 - one day
- **filter_type** - select the statistics type using the following values: 0 - GB, 1 - hit/miss or 2 - speed
- **entity_id** - set CDN resource ID, for which you want to view the statistics, or leave this parameter blank to choose all CDN resources
- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

XML Output Example
<cache_statistics>

<cache_statistic_line_chart type="array">
<cache_statistic_line_chart>
<time>2016-12-09</time>
<cached type="float">533803623.0</cached>
<uncached type="float">11239003.0</uncached>
<hit type="float">0.0</hit>
<miss type="float">0.0</miss>
<gb type="float">0.0</gb>
</cache_statistic_line_chart>
<cache_statistic_table type="array">
<cache_statistic_table>
<locationId type="integer">547</locationId>
<request type="float">1761.0</request>
<gb type="float">736003742.0</gb>
<hit type="float">0.0</hit>
<miss type="float">0.0</miss>
<speed type="float">0.0</speed>
</cache_statistic_table>
</cache_statistic_table>
</cache_statistics>

Where:

cache_statistic_line_chart - an array of cache statistics chart details:
  time - the selected time period
  cached - the amount of cached bandwidth
  uncached - the amount of uncached bandwidth
  hit - the amount of successful file requests for the selected period
  miss - the amount of failed file requests for the selected period
  gb - the amount of transmitted bandwidth in bytes for the selected period

cache_statistic_table - an array of cache statistic table details:
  locationId - the ID of CDN location
  request - the amount of requests at the corresponding CDN location
  gb - the amount of transmitted bandwidth in bytes
  hit - the amount of successful file requests
  miss - the amount of failed file requests
  speed - the cache statistics speed amount

4.13.6 Get Status Codes Report

To get the Status Codes report, use the following request:

1. To get the breakdown of one or all publisher’s resource and total error code, use:
   GET /cdn/reports/status_codes.xml
   GET /cdn/reports/status_codes.json

2. To get the breakdown of location with total error code for selected resource, use:
   GET /cdn/reports/status_codes/error_requests.xml
   GET /cdn/reports/status_codes/error_requests.json

3. To get the breakdown of error code in graph and table for selected location, use:
   GET /cdn/reports/status_codes/location_error_requests.xml
GET /cdn/reports/status_codes/location_error_requests.json

XML Request Example

curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/status_codes.xml -d
  '<status_codes><frequency>2</frequency><entity_id></entity_id><start_date>
  2016-11-09</start_date><end_date>2016-11-10</end_date></status_codes>' -H
  'Accept: application/xml' -H 'Content-type: application/xml'

JSON Request Example

curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/status_codes.json -d '{"status_codes":
  {"frequency":"2","entity_id":"","start_date": "2016-11-09", "end_date":
  "2016-11-10"}}' -H 'Accept: application/json' -H 'Content-type:
application/json'

Where:
Define statistics filters using the following parameters:

- **frequency** - set statistics frequency using the following values: 0 - one minute, 1- one hour
  or 2- one day
- **entity_id** - set CDN resource ID, for which you want to view the statistics, or leave this
  parameter blank to choose all CDN resources
- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

XML Output Example
<status_codes>
  <status_code_line_chart type="array">
    <status_code_line_chart>
      <time>2016-12-09</time>
      <statusCode>200</statusCode>
      <request type="float">183.0</request>
    </status_code_line_chart>
    <status_code_line_chart>
      <time>2016-12-09</time>
      <statusCode>403</statusCode>
      <request type="float">28.0</request>
    </status_code_line_chart>
    <status_code_table type="array">
      <status_code_table>
        <statusCode>200</statusCode>
        <request type="float">2962.0</request>
      </status_code_table>
      <status_code_table>
        <statusCode>403</statusCode>
        <request type="float">1439.0</request>
      </status_code_table>
    </status_code_table>
  </status_code_line_chart>
  <http_error_code_table type="array">
    <http_error_code_table>
      <resourceId type="integer">741579723</resourceId>
      <errorRequest type="float">1289.0</errorRequest>
    </http_error_code_table>
  </http_error_code_table>
</status_codes>

Where:

status_code_line_chart - an array of status code line chart details:
  time - the selected time period
  request - the amount of requests with the corresponding error
  statusCode - the status code of an error

status_code_table - an array of status codes table details:
  request - the amount of requests with the corresponding error
  statusCode - the status code of an error

http_error_code_table - an array of http error code table details:
  resourceId - the ID of CDN resource
  errorRequest - the amount of error requests

4.13.7 Get Stream Bandwidth Report

To get bandwidth statistics report, use the following request:

GET /cdn/reports/bandwidth_statistics.xml
GET /cdn/reports/bandwidth_statistics.json

Be aware, that the bandwidth statistics report shows information on
Stream type CDN resources only.

XML Request Example
curl -i -X GET -u user:userpass 
"http://onapp.test/cdn/reports/bandwidth_statistics.xml" -d 
'\t<bandwidth_statistics><type>MBPS</type><start>2016-12-12T19:52</start><end>2016-12-28T19:52</end><group_by>location</group_by><locations 
type="array"><location>547</location></locations><resources type = 
"array"><resource>898241191</resource><resources></resources></bandwidth_statistics>" 
-H 'Accept: application/xml' -H 'Content-type: application/xml'

**JSON Request Example**

curl -i -X GET -u user:userpass -H 'Accept: application/json' -H 'Content-type: application/json' 
"http://onapp.test/cdn/reports/bandwidth_statistics.json" -d 
'{ "bandwidth_statistics": {"start": "2016-12-12T19:52","end": "2016-12-28T19:52","locations": [547],"resources": [898241191], 
"group_by": "location", "type": "MBPS"}}' -H 'Accept: application/json' -H 'Content-type: application/json'

**Where:**

- **start** – the start date to generate statistics in the YYYY-MM-DD+hh:mm:ss format
- **end** – the end date to generate statistics in the YYYY-MM-DD+hh:mm:ss format
- **resources** – the identifier of the resource in Aflexi database. To get the identifier, check with cdn_reference parameter in the GET /cdn_resources/:id.{format} request
- **locations** – the ID of the location
- **type** – the statistics type (MBPS or GB). In MBPS mode you can get statistics for the last 10 days only. The older statistics is removed. There are no restrictions for GB mode.

**group_by** - to get the bandwidth statistics breaking down per location or per resource via API, use the group_by parameter with two possible values: location and resource. In case group_by = location bandwidth stats is breaking down per location, and if group_by = resource is breaking down per resource. This is the optional parameter.

**XML Output Example if GB statistics type is selected**

```xml
<stats type="array">
  <stat>
    <date type="dateTime">2016-12-24T02:00:00+02:00</date>
    <locations type="array">
      <location>547</location>
      <cached type="float">0.0</cached>
      <non_cached type="float">0.10429243099999999</non_cached>
    </locations>
  </stat>
  ...
</stats>
```

**Where:**

- **date** – the point of time for which the statistics is generated
- **location** - an array of locations with the following details per location:
**cached** – the amount of data cached

**nonCached** – the amount of content which is not cached

### XML Output Example if MBPS statistics type is selected

```xml
<stats type="array">
  <stat>
    <date type="dateTime">2016-12-23T02:00:00+02:00</date>
    <speed type="float">0.202613</speed>
  </stat>
  ...
</stats>
```

Where:
- **date** – the point of time for which the statistics is generated
- **speed** - a bandwidth statistics speed (in Mbits/s)

### 4.13.8 View CDN Resource Concurrent Statistics

To view CDN resource concurrent statistics, use the following request:

**GET /cdn/reports/concurrent_statistics.xml**

**GET /cdn/reports/concurrent_statistics.json**

#### XML Request Example

```bash
```

#### JSON Request Example

```bash
```

Where:
- **start** – the start date to generate statistics in the YYYY-MM-DD format
- **end** – the end date to generate statistics in the YYYY-MM-DD format
- **resources** – the identifier of the resource in Aflexi database. To get the identifier, check with cdn_reference parameter in the GET /cdn_resources/:id.{format} request
- **locations** – the ID of the location
4.13.9 Get Visitors Report

To get the Visitors report, use the following request:

GET /cdn/reports/visitors.xml
GET /cdn/reports/visitors.json

XML Request Example

```bash
curl -i -X GET -u user:userpass --url http://onapp.test/cdn/reports/visitors.xml
-d '<<visitors><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date><entity_id>945986057</entity_id></visitors>' -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request Example

```bash
curl -i -X GET -u user:userpass --url http://onapp.test/cdn/reports/visitors.json
-d '
{"visitors":
{"start_date": "2016-11-09", "end_date": "2016-11-10", "entity_id": 
"945986057"}}
'} -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:

Define a time period by setting the following parameters:

- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

Get statistics for a particular CDN resource by setting the following parameters:

- **entity_id** - the ID of the CDN resource

XML Output Example

```xml
<visitors>
<top_five_visitor_countries_line_chart type="array">
<top_five_visitor_countries_line_chart>
<time>2016-12-09</time>
<country><!--</country-->
<request type="float">44.0</request>
</top_five_visitor_countries_line_chart>
<visitor_country_table type="array">
<visitor_country_table>
<country><!--</country-->
<avgLatency type="float">1.5404897959183672</avgLatency>
<avgTransferRate type="float">1422596.5896</avgTransferRate>
<request type="float">4313.0</request>
</visitor_country_table>
<country><!--</country-->
<bandwidth type="float">190627943.0</bandwidth>
</visitor_country_table>
</visitors>
```

Where:

- **top_five_visitor_countries_line_chart** - an array of top five visitor countries chart details:
  - **time** - the selected time period
  - **country** - the country, where visitors are located
  - **request** - the total amount of requests for the selected period
**visitor_country_table** - an array of visitor country table details:

- **country** - the country, where visitors are located
- **avgLatency** - the average latency for the selected period
- **avgTransferRate** - the average transfer rate for the selected period
- **request** - the amount of successful file requests for the selected period
- **bandwidth** - the amount of transmitted bandwidth for the selected period

### 4.13.10 Get Admin Report

You can get separate reports for every table of the admin report:

- **top 50 CDN resources**
- **locations**
- **top 50 HTTP errors** (for a specific CDN resource and/or specific location)

To get the full Admin report, use the following request:

GET /cdn/reports/admin.xml
GET /cdn/reports/admin.json

**XML Request Example**
```
curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin.xml -d
  '<admin><frequency>2</frequency><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date></admin>'
  -H 'Accept: application/xml'
  -H 'Content-type: application/xml'
```

**JSON Request Example**
```
curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin.json -d '{"admin":
  {"frequency":"2","start_date": "2016-11-09", "end_date": "2016-11-10"}'}
  -H 'Accept: application/json'
  -H 'Content-type: application/json'
```

Where:

Define statistics filters using the following parameters:

- **frequency** - set statistics frequency using the following values: 0- one minute, 1- one hour or 2- one day.
- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

**XML Output Example**
```
curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin.xml -d
  '<admin><frequency>2</frequency><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date></admin>'
  -H 'Accept: application/xml'
  -H 'Content-type: application/xml'
```
Where:

cache_statistic_admin_line_chart - an array of cache statistics chart details:
  - time - the selected time period
  - cached - the amount of cached bandwidth
  - uncached - the amount of uncached bandwidth
  - hit - the amount of successful file requests for the selected period
  - miss - the amount of failed file requests for the selected period
  - gb - the amount of transmitted bandwidth for the selected period

top_five_resources_admin_table - an array of top five CDN resources table details:
  - resourceId - the ID of CDN resource
  - pastHour - the amount of bandwidth for the past hour
  - currentHour - the amount of bandwidth for this hour
  - bandwidth - the amount of transmitted bandwidth

top_five_locations_admin - an array of top five locations table details:
  - locationId - the ID of the location
  - operatorId - the ID of the operator
  - bandwidth - the amount of transmitted bandwidth

top_five_http_error_codes_admin_table - an array of top five error codes table details:
  - resourceId - the ID of CDN resource
  - errorRequest - the amount of error requests
4.13.10.1 Top 50 CDN resources
To get the top 50 CDN resources report, use the following request:

GET /cdn/reports/admin/top_50_cdn_resources.xml
GET /cdn/reports/admin/top_50_cdn_resources.json

**XML Request Example**

```
curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin/top_50_cdn_resources.xml -d
'admin><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date></admin>'
-H 'Accept: application/xml' -H 'Content-type: application/xml'
```

**JSON Request Example**

```
curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin/top_50_cdn_resources.json -d
'{"admin": {"start_date": "2016-11-09", "end_date": "2016-11-10"}}'
-H 'Accept: application/json' -H 'Content-type: application/json'
```

**Where:**

Define statistics filters using the following parameters:

- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

**XML Output Example**

```xml
<top_50_cdn_resources>
  <top_fifty_resources_admin_table type="array">
    <top_fifty_resources_admin_table>
      <resourceId type="integer">798150172</resourceId>
      <pastHour type="float">0.0</pastHour>
      <currentHour type="float">0.0</currentHour>
      <bandwidth type="float">27423031805.0</bandwidth>
    </top_fifty_resources_admin_table>
  </top_fifty_resources_admin_table>
</top_50_cdn_resources>
```

**Where:**

- **resourceId** - the ID of CDN resource
- **pastHour** - the amount of bandwidth for the past hour
- **currentHour** - the amount of bandwidth for this hour
- **bandwidth** - the amount of transmitted bandwidth

4.13.10.2 Locations
To get the locations report, use the following request:

GET /cdn/reports/admin/locations.xml
GET /cdn/reports/admin/locations.json
XML Request Example

curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin/locations.xml -d
'<!--admin><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date></admin-->' -H 'Accept: application/XML' -H 'Content-type: application/xml'

JSON Request Example

curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin/locations.json -d '{"admin":
{"start_date": "2016-11-09", "end_date": "2016-11-10"}}' -H 'Accept: application/json' -H 'Content-type: application/json'

Where:

Define statistics filters using the following parameters:

- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period

XML Output Example

```xml
<locations>
  <location_admin_table type="array">
    <location_admin_table>
      <locationId type="integer">189649337</locationId>
      <operatorId type="integer">0</operatorId>
      <bandwidth type="float">27423031805.0</bandwidth>
    </location_admin_table>
  </location_admin_table>
</locations>
```

Where:

- **locationId** - the ID of the location
- **operatorId** - the ID of the operator
- **bandwidth** - the amount of transmitted bandwidth

4.13.10.3 Top 50 HTTP errors

To get the top 50 HTTP errors report, use the following request:

- GET /cdn/reports/admin/top_50_http_errors.xml
- GET /cdn/reports/admin/top_50_http_errors.json

XML Request Example

curl -i -X GET -u user:userpass --url
http://onapp.test/cdn/reports/admin/top_50_http_errors.xml -d
'<!--admin><start_date>2016-11-09</start_date><end_date>2016-11-10</end_date></admin-->' -H 'Accept: application/XML' -H 'Content-type: application/xml'
JSON Request Example

```
```

Where:
Define statistics filters using the following parameters:
- **start_date** - the start date of the specific time period
- **end_date** - the end date of the specific time period
- **entity_id** - set CDN resource ID
- **location_id** - the ID of the location

XML Output Example

```
<top_50_http_errors>
  <top_fifty_http_error_codes_admin_table type="array">
    <top_fifty_http_error_codes_admin_table>
      <resourceId type="integer">798150172</resourceId>
      <errorRequest type="float">5232828.0</errorRequest>
    </top_fifty_http_error_codes_admin_table>
  </top_fifty_http_error_codes_admin_table>
</top_50_http_errors>
```

Where:
- **resourceId** - the ID of CDN resource
- **errorRequest** - the amount of error requests

### 4.14 CDN Resources API

A CDN resource is a host (e.g. a specific web server), the content of which you are going to distribute over the network of edge servers. The list of servers taking part in distributing/caching of data is limited to the locations added to those edge groups assigned to the resource.

#### 4.14.1 Get List of CDN Resources

To see all CDN resources in the cloud, use the following request:

**GET /cdn_resources.xml**
**GET /cdn_resources.json**

XML Request Example

```
curl -i -X GET -u user:userpass http://onapp.test/cdn_resources.xml
```

JSON Request Example

```
curl -i -X GET -u user:userpass http://onapp.test/cdn_resources.json
```
To view the list of HTTP resources, use the following request:

**XML Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/cdn_resources.xml?type=http
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/cdn_resources.json?type=http
```

To view the list of VoD resources, use the following request:

**XML Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/cdn_resources.xml?type=vod
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/cdn_resources.json?type=vod
```

To view the list of live streaming resources, use the following request:

**XML Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/cdn_resources.xml?type=live_streaming
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass
```
<cdn_resources type="array">  
<cdn_resource>  
<cdn_hostname>cdn.lexample.com</cdn_hostname>  
<cdn_ssl_certificate_id nil="true"/>  
<cname>990113320.r.worldcdn-beta.net</cname>  
<created_at type="datetime">2012-05-10T14:19:02+00:00</created_at>  
<id type="integer">2</id>  
<resource_type>HTTP_PULL</resource_type>  
<updated_at type="datetime">2012-05-10T14:19:02+00:00</updated_at>  
<user_id type="integer">1</user_id>  
<last_24h_cost type="float">0.0</last_24h_cost>  
<cname>990113320.r.worldcdn-beta.net</cname>  
<origins type="array">  
<origin>  
</origin>  
</origins>  
</cdn_resource>  
</cdn_resources>

Where:

cdn_hostname - the hostname which will serve static content  
cdn_ssl_certificate_id - the ID of the custom SNI SSL certificate added to the resource  
cname - CNAME record  
created_at - the date when the CDN resource was created in the [YYYY][MM][DD][hh][mm][ss]Z format  
id - the resource ID in the database  
resource_type - CDN resource type  
updated_at - the date when the CDN resource was updated in the [YYYY][MM][DD][hh][mm][ss]Z format  
user_id - the ID of the user, who owns the resource  
last_24h_cost - the amount due for the last 24 hours  
cdn_reference - ID of the SSL certificate on the remote Aflexi server  
origins - an array of CDN origins with the following parameters:  
origin - the path from which the CDN requests the content  
key - access key, if any  
value - the path to the content

4.14.2 Get CDN Resource Basic Details

To view details of the particular CDN resource, use the following request:

GET /cdn_resources/:id.xml  
GET /cdn_resources/:id.json

XML Request Example

JSON Request Example


XML Output Example

<cdn_resource>
<cdn_hostname>cdn.test.com</cdn.hostname>
<created_at type="datetime">2012-05-14T10:19:37+00:00</created_at>
<id type="integer">3</id>
<resource_type>HTTP_PULL</resource_type>
<updated_at type="datetime">2012-05-14T10:19:37+00:00</updated_at>
<user_id type="integer">1</user_id>
<last_24h_cost type="float">0.0</last_24h_cost>
<edge_groups type="array">
<edge_group>
<created_at type="datetime">2012-04-18T11:58:05+00:00</created_at>
<id type="integer">158</id>
<label>dfgfg</label>
<updated_at type="datetime">2012-04-18T11:58:05+00:00</updated_at>
</edge_group>
</edge_groups>
<status>ACTIVE</status>
<secondary_hostnames type="array"/>
<ssl_on type="boolean">false</ssl_on>
<ssl nil="true"/>
<origins type="array">
<origin>
<key/>
<value>test.com</value>
</origin>
</origins>
</cdn_resource>

Where:

cdn_hostname - the hostname which will serve static content
created_at – the date when the resource was created
id – the resource ID in the database
resource_type – HTTP PULL or PUSH
updated_at – the date when the resource was updated
user_id – the ID of the user, who owns the resource
last_24h_cost – the amount of money owed for the resource for the last 24 hours.
edge_groups – the array of edge groups assigned to this resource, where:
created_at – the date when the edge group was created
label – the label of the particular edge group assigned
id – the edge group id
updated_at – the date when the edge group was updated
status – the resource status (can be Preparing, Active, Suspended)
secondary hostname – secondary CDN hostname
ssl_on - whether SSL is enabled for the resource or not
ssl - custom SNI SSL certificate if it is added to the resource
origins – the path from which the CDN requests the content
key – access key, if any
value – the path to the content

4.14.3 Get CDN Resource Advanced Details

To view advanced details of the CDN resource, use the following request:

GET /cdn_resources/:id/advanced.xml
GET /cdn_resources/:id/advanced.json

XML Request Example

```
```

JSON Request Example

```
```

XML Output Example
```xml
<cdn_resource>
  <secondary_hostnames type="array">
    <secondary_hostname>test.com</secondary_hostname>
  </secondary_hostnames>
  <ip_access_policy>ALLOW_BY_DEFAULT</ip_access_policy>
  <country_access_policy>BLOCK_BY_DEFAULT</country_access_policy>
  <countries type="array">
    <country>AL</country>
    <country>AR</country>
    <country>GT</country>
    <country>HR</country>
  </countries>
  <url_signing_on type="boolean">true</url_signing_on>
  <domain>www.example.com</domain>
  <password_on type="boolean">true</password_on>
  <passwords>
    <username1>password1</username1>
    <username2>password2</username2>
  </passwords>
  <password_unauthorized_html>password unauthorized</password_unauthorized_html>
  <flv_pseudo_on type="boolean">true</flv_pseudo_on>
  <mp4_pseudo_on type="boolean">true</mp4_pseudo_on>
  <limit_rate type="integer">80</limit_rate>
  <limit_rate_after type="integer">13</limit_rate_after>
</cdn_resource>
```

Where:

- **secondary hostname** – secondary CDN hostname
- **ip_access_policy** – displays access policy from a range of IP addresses; either NONE (disabled), ALLOW_BY_DEFAULT or BLOCKED_BY_DEFAULT
- **country_access_policy** – displays access policy to the CDN resource’s content for specified countries; either NONE (disabled), ALLOW_BY_DEFAULT or BLOCKED_BY_DEFAULT
- **url_signing_on** – true if the access requires URL signing; otherwise false
- **url_signing_key** – the key for URL signing; a signed URL looks like: 
  `_http://example.com/filename?hash=url-signing-key=_`
- **hotlink_policy** – displays the hotlink policy; either NONE (disabled), ALLOW_BY_DEFAULT or BLOCKED_BY_DEFAULT
- **domains** - domains related to hotlink_policy
- **password_on** – true, if the access to the resource is restricted; otherwise false
- **cors_on** - true, if CORS Headers are enabled for this resource; otherwise, false
- **passwords** – an array of username and password for restricted access in the following format:
  `<username>password</username>`
- **mp4_pseudo_on** - 1 if MP4 pseudo streaming is enabled, otherwise set 0
- **flv_pseudo_on** - 1 if FLV pseudo streaming is enabled, otherwise set 0
- **limit_rate** - speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
- **limit_rate_after** - the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value -2147483647 KB
4.14.4 Add HTTP CDN Resource

To create an HTTP resource, use the following request:

POST /cdn_resources.xml
POST /cdn_resources.json

Add HTTP PULL XML Request Example

```bash
curl -i -X POST -d
  '<cdn_resource><cdn_hostname>cdn.test.co</cdn_hostname><edge_group_ids type="array"><edge_group_id type="integer">1</edge_group_id></edge_group_ids><resource_type>HTTP_PULL</resource_type><origin>test.origin.com</origin><letsencrypt_ssl_on type="boolean">true</letsencrypt_ssl_on></cdn_resource>' -u user:userpass http://onapp.test/cdn_resources.xml -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

Add HTTP PULL JSON Request Example

```bash
curl -i -u user:userpass -X POST http://onapp.test/cdn_resources.json -H 'Accept: application/json' -H 'Content-type: application/json' -d '{"cdn_resource":{"cdn_hostname":"cdn.test.co","resource_type":"HTTP_PULL","cdn_ssl_certificate_id":"ssl_cert_id","edge_group_ids":[1],"origin":"test.origin.com"}}'
```

Add HTTP PUSH XML Request Example

```bash
curl -i -X POST -d
  '<cdn_resource><cdn_hostname>cdn.test.co</cdn_hostname><cdn_ssl_certificate_id="ssl_cert_id"/><edge_group_ids type="array"><edge_group_id type="integer">2</edge_group_id></edge_group_ids><resource_type>HTTP_PUSH</resource_type><ftp_password>j3x8svyba2gmrgYOZZr01YToQ</ftp_password></cdn_resource>' -u user:userpass http://onapp.test/cdn_resources.xml -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

Add HTTP PUSH JSON Request Example

```bash
curl -i -u user:userpass -X POST http://onapp.test/cdn_resources.json -H 'Accept: application/json' -H 'Content-type: application/json' -d '{"cdn_resource":{"cdn_hostname":"cdn.test.co","resource_type":"HTTP_PUSH","edge_group_ids":[7],"ftp_password":"j3x8svyba2gmrgYOZZr01YToQ","letsencrypt_ssl_on":"true"}}'
```

Where:

- `origin` - the path from which the CDN requests the content. You can specify up to 3 origins. You can specify custom origin port (for HTTP pull resource only). To use the custom port for resource's origin, specify a port number using a colon (":"). For example, `<origin>1.2.3.4:80</origin>`.
When you are specifying only one origin, it can be either a CDN hostname or an IP address. In case you are specifying more than one origin, they can only be IP addresses.

To send two or more origins in the API request, use array:

**XML Request Example**

```xml
<origins type="array">
  <origin>111.111.11.111</origin>
  <origin>111.111.11.111</origin>
</origins>
```

**JSON Request Example**

```json
{"origins": ["111.111.11.111", "222.222.22.222"]}
```

cdn_hostname * - indicate the hostname which will serve static content. Specify the following fourth-level domain name for this parameter to create a resource with SSL enabled: "example.r.worldssl.net", where replace the example with the desired name.

resource_type * - HTTP_PULL or HTTP_PUSH

cdn_ssl_certificate_id - the ID of the custom SNI SSL certificate you want to add to the resource. You should only specify the IDs of those certificates that were added by the user with whom the new resource will be associated.

edge_group_ids * - indicate the ID(s) of required CDN edge groups

ftp_password * - specify the FTP password if you add an HTTP PUSH CDN resource type. It can consist of 6-32 alphanumeric characters.

letsencrypt_ssl_on - set to true to enable Let's Encrypt SSL for the resource

**Response**

You will get a response consisting of two parts - the header with HTTP status code and the response body including the parameters. At this stage, some of the parameters can be empty. This is expected behavior, because the full process of creation takes some time, and remote service cannot fill in the parameters at this time. For the complete list of parameters use GET request.

**Page History**

v. 5.6
- added letsencrypt_ssl_on parameter that enables Let's Encrypt SSL for a CDN resource

v. 4.0:
- origin parameter allows adding port to HTTP pull resource.
- added cdn_ssl_certificate_id parameter that enables a user to associate a custom SNI SSL certificate with a CDN resource

### 4.14.5 Add Video on Demand CDN Resource

To create an HTTP resource, use the following request:
POST /cdn_resources.xml
POST /cdn_resources.json

Add VoD PULL XML Request Example

curl -i -X POST -d
'<?xml version="1.0"?>
<cdn_resource><cdn_hostname>az.test.api</cdn_hostname><edge_group_ids type="array"><edge_group_id type="integer">7</edge_group_id></edge_group_ids><resource_type>STREAM_VOD_PULL</resource_type><origin>test.org</origin></cdn_resource>' -u user:userpass http://onapp.test/cdn_resources.xml -H 'Accept: application/xml' -H 'Content-type: application/xml'

Add VoD PULL JSON Request Example


Add VoD PUSH XML Request Example

curl -i -X POST -d
'<?xml version="1.0"?>
<cdn_resource><cdn_hostname>cdn.test.co</cdn_hostname><edge_group_ids type="array"><edge_group_id type="integer">7</edge_group_id></edge_group_ids><resource_type>STREAM_VOD_PUSH</resource_type><ftp_password>j3x8svqybA2gmrgYOZSr01YTQc</ftp_password></cdn_resource>' -u user:userpass http://onapp.test/cdn_resources.xml -H 'Accept: application/xml' -H 'Content-type: application/xml'

Add VoD PUSH JSON Request Example

curl -i -X POST http://onapp.test/cdn_resources.json -H 'Accept: application/json' -H 'Content-type: application/json' -d '{"cdn_resource":{"cdn_hostname":"cdn.test.co","resource_type":"STREAM_VOD_PUSH","ftp_password":"j3x8svqybA2gmrgYOZSr01YTQc"}}'

Where:
cdn_hostname * - specify the name which will serve as a label only
resource_type* - specify the resource type - STREAM_VOD_PULL or STREAM_VOD_PUSH
origin* - the path from which the CDN requests the content (for VoD Pull request). The VoD Pull resource can have only one origin.
edge_group_ids * - indicate the ID(s) of required CDN edge groups
ftp_password * - specify the FTP password for VoD PUSH type. It should consist of 6-32 alphanumeric symbols.

Response
You will get a response consisting of two parts - the header with HTTP status code and the response body including the parameters. At this stage some of the parameters can be empty. This is expected behavior because the full process of creation takes some time, and remote
service cannot fill in the parameters at this time. For the complete list of parameters use GET request.

4.14.6 Add Live Streaming CDN Resource

To create a live streaming CDN resource, use the following request:

POST /cdn_resources.xml
POST /cdn_resources.json

XML Request Example (with external publishing point)

```bash
curl -i -X POST -d
'<?xml version="1.0" encoding="UTF-8" ?>
<cdn_resource><cdn_hostname>onapp.stream.resource</cdn_hostname><edge_group_ids type="array"><edge_group_id type="integer">1</edge_group_id></edge_group_ids><resource_type>STREAM_LIVE</resource_type><publishing_point>external</publishing_point><publishing_location>www.google.com</publishing_location><failover_publishing_location>rtmp://test.com/test</failover_publishing_location></cdn_resource>'
-u user:userpass http://onapp.test/cdn_resources.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

JSON Request Example (with external publishing point)

```bash
curl -i -u user:userpassword -X POST http://onapp.test/cdn_resources.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
-d '{"cdn_resource":{"cdn_hostname":"onapp.stream.resource","resource_type":"STREAM_LIVE","edge_group_ids":[7],"publishing_point":"external","publishing_location":"rtmp://test.me/test","failover_publishing_location":"rtmp://test.com/test"}}'
```

XML Request Example (with internal publishing point)

```bash
curl -i -X POST -d
'<?xml version="1.0" encoding="UTF-8" ?>
<cdn_resource><cdn_hostname>onapp.stream.resource</cdn_hostname><edge_group_ids type="array"><edge_group_id type="integer">1</edge_group_id></edge_group_ids><resource_type>STREAM_LIVE</resource_type><publishing_point>internal</publishing_point><publishing_location>532</publishing_location><failover_publishing_location>128</failover_publishing_location></cdn_resource>'
-u user:userpass http://onapp.test/cdn_resources.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

JSON Request example (with internal publishing point)

```bash
curl -i -u user:userpass -X POST http://test/cdn_resources.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
-d '{"cdn_resource":{"cdn_hostname":"onapp.stream.resource","resource_type":"STREAM_LIVE","edge_group_ids":[7],"publishing_point":"internal","publishing_location":"532","failover_publishing_location":"128"}}'
```

Where:
**cdn_hostname** - specify the name which will serve as a label only

**resource_type** - STREAM_LIVE

**publishing_point** - the publishing point type: external or internal

**publishing_location** - specify the URL address for external publishing point. Set the ID of a location that will serve as a publishing point for internal type.

**failover_publishing_location** - publishing point failover URL for external publishing point. Specify the ID of a location that will serve as a failover publishing point for internal type.

**edge_group_ids** - indicate the ID(s) of required CDN edge groups

**Response**

You will get a response consisting of two parts - the header with HTTP status code (see Introduction for details) and the response body including the parameters. At this stage some of the parameters can be empty. This is expected behavior, because the full process of creation takes some time, and remote service cannot fill in the parameters at this time. For the complete list of parameters use GET request.

### 4.14.7 Add HTTP Pull CDN Resource with Advanced Settings

To create an HTTP resource with advanced settings, use the following request:

POST /cdn_resources.xml
POST /cdn_resources.json

**XML Request example**

  <cdn_resource>
    <resource_type>HTTP_PULL</resource_type>
    <cdn_hostname>HTTPPULCDNresource.com</cdn_hostname>
    <origin>109.123.105.178</origin>
    <cdn_ssl_certificate_id>ssl_sert_id</cdn_ssl_certificate_id>
    <edge_group_ids type="array"><edge_group_id>225</edge_group_id></edge_group_ids>
    <secondary_hostnames type="array"><secondary_hostname>test110.com</secondary_hostname><secondary_hostname>test220.com</secondary_hostname></secondary_hostnames>
    <ip_access_policy>BLOCK_BY_DEFAULT</ip_access_policy>
    <ip_addresses>111.111.111.111,222.222.222.222</ip_addresses>
    <country_access_policy>ALLOW_BY_DEFAULT</country_access_policy>
    <countries type="array"><country>AL</country><country>GT</country></countries>
    <hotlink_policy>ALLOW_BY_DEFAULT</hotlink_policy>
    <url_signing_on>1</url_signing_on>
    <url_signing_key>12345qwertyuy</url_signing_key>
    <cache_expiry>45</cache_expiry>
    <password_on>1</password_on>
    <form_pass type="array"><string>user123new</string><string>user234new</string></form_pass>
    <password_unauthorized_html>YOU ARE NOT AUTHORIZED</password_unauthorized_html>
    <flv_pseudo_on>1</flv_pseudo_on>
    <mp4_pseudo_on>0</mp4_pseudo_on>
    <cors_on>1</cors_on>
    <limit_rate>150</limit_rate>
    <limit_rate_after>1</limit_rate_after>
    <proxy_cache_key>$host$uri</proxy_cache_key>
    <proxy_read_time_out>60</proxy_read_time_out>
    <proxy_connect_time_out>20</proxy_connect_time_out>
    <origin_policy>HTTP</origin_policy>
  </cdn_resource>

**JSON Request example**

Where:

**origin** - the path from which the CDN requests the content. When using the hostnames according to RFC 1035, the origin may consist of letters [A-Z a-z] (case insensitive manner), digits [0-9], and dash [-]. The limit for hostname is 255 chars. You can specify up to 3 origins.

When you are specifying only one origin, it can be either a CDN hostname or an IP address. In case you are specifying more than one origin, they can only be IP addresses.

To send two or more origins in the API request, use array. For example:

```xml
<origins type="array">
  <origin>111.111.111.111</origin>
  <origin>111.111.111.111</origin>
</origins>
```

**cdn_hostname** - indicate the hostname which will serve static content. Specify the following fourth-level domain name for this parameter to create a resource with SSL enabled: "example.r.worldssl.net", where replace the example with the desired name.

**ssl_on** - set to 'true' to enable SSL, otherwise set to 'false'. See the note below for more information.

**SSL**

If the CDN hostname ends with '.r.worldssl.net', SSL will be enabled automatically.

To disable, remove the '.r.worldssl.net' ending and send the "ssl_on":false parameter. To enable, add the '.r.worldssl.net' ending to the cdn_hostname and send the "ssl_on":true parameter.

Be aware that if CDN hostname ends with 'r.worldssl.net', it can not be digit-only (for example 123456.r.worldssl.net is not applicable).
resource_type* - HTTP_PULL

cdn_ssl_certificate_id - the ID of the custom SNI SSL certificate you want to add to the resource. You should only specify the IDs of those certificates that were added by the user with whom the new resource will be associated.

letsencrypt_ssl_on - set to true to enable Let's Encrypt SSL for the resource

default_edge_group_ids* - indicate the ID(s) of required CDN edge groups

secondary_hostnames - an array of secondary CDN hostnames. You can add up to 7 secondary CDN hostnames. For example:

```xml
<secondary_hostnames type="array">
  <secondary_hostname>test100.com</secondary_hostname>
  <secondary_hostname>test200.com</secondary_hostname>
</secondary_hostnames>
```

To be able to use a secondary hostname for the CDN resource with SSL enabled, you require an SSL certificate for your custom hostname. For help with questions about the SSL certificate purchase, please contact OnApp support.

ip_access_policy - configure a rule to control access to the CDN resource's content for a range of IP addresses:

- ALLOW_BY_DEFAULT - allow IP access policy by default, except for IP addresses specified in the ip_addresses parameter
- BLOCK_BY_DEFAULT - block IP access policy by default, except for IP addresses specified in the ip_addresses parameter
- NONE - switch off the IP access policy

ip_addresses - IP address(es) related to ip_access_policy parameter; The comma-separated list of IP addresses or IP ranges allowed/blocked by default. Use the following format "10.10.10.10, 20.20.20.0/24"

hotlink_policy - configure hotlink policy properties to protect your content from unauthorized hotlinking:

- ALLOW_BY_DEFAULT - allow hotlink policy by default, except for domains specified in the domains parameter
- BLOCK_BY_DEFAULT - block hotlink_policy by default, except for domains specified in the domains parameter
- NONE - switch off the rule

domains - domains related to hotlink_policy

hls_on - set to 1 to enable HTTP Live Streaming (HLS) Optimization

hls_force_cache - set to 1 to create an HTTP rule that will enforce cache expiry. This option is only available when the hls_on parameter is set to 1

country_access_policy - configure a rule to control access to the CDN resource's content for specified countries:

- ALLOW_BY_DEFAULT - allow country access policy by default, except for countries specified in the countries parameter
• BLOCK_BY_DEFAULT - block country access policy by default, except for countries specified in the countries parameter

• NONE - switch off the country access policy
countries - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format
cache_expiry - set the cache expiry time in minutes (min=1, max=35000000)
url_signing_on - set 1 to enable and protect your files from unauthorized access with a key
url_signing_key - input the key for URL signing. Input letters and digits (6-32 symbols).
password_on - set 1 to enable and to restrict access to the resource (cdn_hostname), otherwise set 0
form_pass - an array with usernames and passwords to access the resource
pass - the user password.
user - the user login, which may consist of letters [A-Z a-z] (case insensitive manner), digits [0-9], underscore [ _ ], and dash [ - ]. The first symbol should be alphabetic. The username cannot be duplicated.
password_unauthorized_html - text, which will be displayed in case of fail of authentication. Max 1000 chars.
mp4_pseudo_on - set 1 to enable MP4 pseudo streaming, otherwise set 0
flv_pseudo_on - set 1 to enable FLV pseudo streaming, otherwise set 0

Nginx Settings
• limit_rate - sets the speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
• limit_rate_after - sets the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value - 2147483647 KB
• proxy_cache_key - specify the key for caching. This parameter defines what information is included in the cache key. You can set the following options:
  o $host$request_uri
  o $host$uri
  o $proxy_host$request_uri
  o $proxy_host$uri

http_bot_blocked - set 1 to block web crawling bots from indexing the CDN content (for HTTP Pull CDN resources only)
origin_policy - set the parameter to choose the type of connection. Possible values are: HTTP, HTTPS, AUTO.

Response
You will get a response consisting of two parts - the header with HTTP status code and the response body including the parameters. At this stage, some of the parameters can be empty. This is expected behavior, because the full process of creation takes some time, and remote service cannot fill in the parameters at this time. For the complete list of parameters use GET request.
XML Response Example

```xml
<?xml version="1.0" encoding="UTF-8"?><cdn_resource>
  <cdn_hostname>PI-HTTPPULCDNresource.com</cdn_hostname>
  <created_at type="datetime">2013-07-31T12:19:55+03:00</created_at>
  <id type="integer">10743</id>
  <resource_type>HTTP_PULL</resource_type>
  <updated_at type="datetime">2013-07-31T12:19:55+03:00</updated_at>
  <user_id type="integer">20</user_id>
  <last_24h_cost type="float">0.0</last_24h_cost>
  <cdn_reference type="integer">431059243</cdn_reference>
  <secondary_hostnames type="array">
    <secondary_hostname>test100.com</secondary_hostname>
    <secondary_hostname>test200.com</secondary_hostname>
  </secondary_hostnames>
  <origins type="array">
    <origin>1.1.1.1/</origin>
  </origins>
  <ssl_on nil="true"/>
</cdn_resource>
```

4.14.7.1  Page history

v.6.1
- Added the cors_on parameter

v.5.6
- Added letsencrypt_ssl_on parameter that enables Let's Encrypt SSL for a CDN resource
- Removed proxy_read_time_out and proxy_connect_time_out parameters from CDN advanced settings

v.5.1
- added the hls_force_cache parameter
- added the hls_on parameter

v.4.0
- Added cdn_ssl_certificate_id parameter that enables a user to associate a custom SNI SSL certificate with a CDN resource
v. 3.3.1
Added the following parameter:
- `origin_policy`

4.14.8 Add HTTP Push CDN Resource with Advanced Settings
To create an HTTP resource with advanced settings, use the following request:

**POST /cdn_resources.xml**
**POST /cdn_resources.json**

**XML Request Example**
```
curl -i -X POST -u user:userpass http://onapp.test/cdn_resources.xml -H 'Accept:application/xml' -H 'Content-type: application/xml' -d '<cdn_resource><resource_type>HTTP_PUSH</resource_type><cdn_hostname>xmlcdn.apitest.com</cdn_hostname><cdn_ssl_certificate_id>ssl_cert_id</cdn_ssl_certificate_id><edge_group_ids type="array"><edge_group_id type="integer">225</edge_group_id></edge_group_ids><storage_server_location>816382921</storage_server_location><ftp_password>qwerty123</ftp_password><secondary_hostnames type="array"><secondary_hostname>test12.com</secondary_hostname><secondary_hostname>test234.com</secondary_hostname></secondary_hostnames><ip_access_policy>ALLOW_BY_DEFAULT</ip_access_policy><ip_addresses>111.111.111.111,222.222.22.222</ip_addresses><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries>AO</countries><hotlink_policy>ALLOW_BY_DEFAULT</hotlink_policy><domains><domain>example.com</domain></domains><storage_server_location>816382921</storage_server_location><ftp_password>qwerty123</ftp_password><secondary_hostnames type="array"><secondary_hostname>test12.com</secondary_hostname><secondary_hostname>test234.com</secondary_hostname></secondary_hostnames><ip_access_policy>ALLOW_BY_DEFAULT</ip_access_policy><ip_addresses>111.111.111.111,222.222.22.222</ip_addresses><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries>AO</countries><hotlink_policy>ALLOW_BY_DEFAULT</hotlink_policy><domains><domain>example.com</domain></domains>...```

**JSON Request Example**
```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url http://onapp.test/cdn_resources.json -d '{"cdn_resource":{"resource_type":"HTTP_PUSH","cdn_hostname":"jsoncdn.apitest.com","cdn_ssl_certificate_id":"ssl_cert_id","edge_group_ids": ["225"],"storage_server_location":816382921,"ftp_password":"qwerty123","secondary_hostnames": ["test12.com","test234.com"],"ip_access_policy":"ALLOW_BY_DEFAULT","ip_addresses":111.111.111.111,222.222.22.222,123.123.123.123,"country_access_policy":"ALLOW_BY_DEFAULT","countries": ["AL","AR","GT","HR"],"hotlink_policy":"ALLOW_BY_DEFAULT","domains": "mnw.net","url_signing_on":1,"url_signing_key": "12345qwertuyio","password_on":1,"form_pass": {"user": ["user190","user278"],"pass": ["pass123","pass2348"]},"password_unauthorized_html": "You are blocked!"},"mp4_pseudo_on":1,"mp4_pseudo_on":1,"flv_pseudo_on":1,"country_access_policy": "ALLOW_BY_DEFAULT","countries": ["AO"],"hotlink_policy": "ALLOW_BY_DEFAULT","domains": ["mnw.net","url_signing_on":1,"url_signing_key": "12345qwertuyio","password_on":1,"form_pass": {"user": ["user190","user278"],"pass": ["pass123","pass2348"]},"password_unauthorized_html": "You are blocked!"}'}
```

Where:
**cdn_hostname** - indicate the hostname which will serve static content. Specify the following fourth-level domain name for this parameter to create a resource with SSL enabled: "example.r.worldssl.net", where replace the example with the desired name.

**ssl_on** - set to 'true' to enable SSL, otherwise set to 'false'. See the note below for more information.

**SSL**

If the CDN hostname ends with '.r.worldssl.net', SSL will be enabled automatically.

To disable, remove the '.r.worldssl.net' ending and send the "ssl_on":false parameter. To enable, add the '.r.worldssl.net' ending to the cdn_hostname and send the "ssl_on":true parameter.

Be aware that if CDN hostname ends with 'r.worldssl.net', it can not be digit-only (for example 123456.r.worldssl.net is not applicable).

**resource_type** - HTTP_PUSH

**cdn_ssl_certificate_id** - the ID of the custom SNI SSL certificate you want to add to the resource. You should only specify the IDs of those certificates that were added by the user with whom the new resource will be associated.

**letsencrypt_ssl_on** - set to true to enable Let's Encrypt SSL for the resource

**edge_group_ids** - indicate the ID(s) of required CDN edge groups

**storage_server_location** - the ID of the storage server location which should be assigned to this resource. To get the ID of the required storage server location, use the request described at the Get List of Available Storage Locations section. If no location set, the first active storage server is chosen automatically.

**ftp_password** - indicate the FTP server password. It should consist of 6-32 alphanumeric symbols.

**secondary_hostnames** - an array of secondary CDN hostnames. You can add up to 7 secondary CDN hostnames.

```xml
<secondary_hostnames type="array">
    <secondary_hostname>test100.com</secondary_hostname>
    <secondary_hostname>test200.com</secondary_hostname>
</secondary_hostnames>
```

To be able to use a secondary hostname for the CDN resource with SSL enabled, you require an SSL certificate for your custom hostname. For help with questions about the SSL certificate purchase, please contact OnApp support.

**ip_access_policy** - configure a rule to control access to the CDN resource's content for a range of IP addresses:

- ALLOW_BY_DEFAULT - allow IP access policy by default, except for IP addresses specified in the *ip_addresses* parameter
• BLOCK_BY_DEFAULT - block IP access policy by default, except for IP addresses specified in the ip_addresses parameter

• NONE - switch off the IP access policy

ip_addresses - IP address(es) related to ip_access_policy parameter; The comma-separated list of IP addresses or IP ranges allowed/blocked by default. Use the following format "10.10.10.10, 20.20.20.0/24"

hotlink_policy - configure hotlink policy properties to protect your content from unauthorized hotlinking:

• ALLOW_BY_DEFAULT - allow hotlink policy by default, except for domains specified in the domains parameter

• BLOCK_BY_DEFAULT - block hotlink_policy by default, except for domains specified in the domains parameter

• NONE - switch off the rule

domains - domains related to hotlink_policy

country_access_policy - configure a rule to control access to the CDN resource's content for specified countries:

• ALLOW_BY_DEFAULT - allow country access policy by default, except for countries specified in the countries parameter

• BLOCK_BY_DEFAULT - block country access policy by default, except for countries specified in the countries parameter

• NONE - switch off the country access policy

countries - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format.

url_signing_on - set 1 to enable and protect your files from unauthorized access with a key

url_signing_key - input the key for URL signing. Input letters and digits (6-32 symbols).

form_pass - an array with usernames and passwords to access the resource:

user - the user login, which may consist of letters [A-Z a-z] (case insensitive manner), digits [0-9], underscore [ _ ], and dash [-]. The first symbol should be alphabetic. The username cannot be duplicated.

password_unauthorized_html - text, which will be displayed in case of fail of authentication. Max 1000 chars.

mp4_pseudo_on - set 1 to enable MP4 pseudo streaming, otherwise set 0

flv_pseudo_on - set 1 to enable FLV pseudo streaming, otherwise set 0

☑️ cors_on - set 1 to enable cross-origin resource sharing (CORS), otherwise set 0

Nginx Settings

• limit_rate - sets speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s

• limit_rate_after - sets the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value -2147483647 KB

Page History

v.6.1

• added the cors_on parameter

v. 5.6
OnApp Cloud 6.5 Edge 5 CDN Guide

- **added** `letsencrypt_ssl_on` parameter that enables Let's Encrypt SSL for a CDN resource v. 4.0
- **updated** `cdn_hostname` parameter with ability to enable SSL
- **added** `cdn_ssl_certificate_id` parameter that enables a user to associate a custom SNI SSL certificate with a CDN resource v. 3.1.1
- **added** `storage_server_location` parameter

### 4.14.9 Add VoD Push CDN Resource With Advanced Settings

To create new video on demand resource with advanced settings, use the following request:

**POST /cdn_resources.xml**

**XML Request Example**

```plaintext
'<!--xml_root-->
<cdn_resource>
  <resource_type>STREAM_VOD_PUSH</resource_type>
  <cdn_hostname>apitest.com</cdn_hostname>
  <ftp_password>testpassword</ftp_password>
  <edge_group_ids>
    <edge_group_id>1</edge_group_id>
    <edge_group_id>2</edge_group_id>
  </edge_group_ids>
  <hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy>
  <domains>goo.coabuse.ua</domains>
  <country_access_policy>ALLOW_BY_DEFAULT</country_access_policy>
  <countries type="array">
    <country>AL</country>
    <country>GT</country>
  </countries>
  <secure_wowza_on>1</secure_wowza_on>
  <secure_wowza_token>test123456</secure_wowza_token>
  <storage_server_location>4637643278</storage_server_location>
  <token_auth_on>1</token_auth_on>
  <token_auth_security_paths type="array">
    <token_auth_secure_path>/Video1</token_auth_secure_path>
    <token_auth_secure_path>/Video2</token_auth_secure_path>
  </token_auth_security_paths>
  <token_auth_backup_key>fgff45788787878</token_auth_backup_key>
</cdn_resource>
'
```

**JSON Request Example**

```plaintext
'{"cdn_resource":"resource_type":"STREAM_VOD_PUSH","ftp_password":"testpassword","cdn_hostname":"apitest.com","edge_group_ids": ["1", "2"],"hotlink_policy":"BLOCK_BY_DEFAULT","domains": ["test.com"],"country_access_policy": "ALLOW_BY_DEFAULT","countries": ["AL", "GT"],"storage_server_location": "4545566" ,"secure_wowza_on": "1", "secure_wowza_token": "test123456","token_auth_on": "1","token_auth_primary_key": "zsfdasfga","token_auth_security_paths": [ "/video1", "/video2" ],"token_auth_backup_key": "fgff45788787878"}'
```

**Where:**

- `cdn_hostname`*: specify the name which will serve as a label only
resource_type* - STREAM_VOD_PUSH

ftp_password* - indicate the FTP server password. It should consist of 6-32 alphanumeric symbols.

domain_group_ids* - indicate the ID(s) of required CDN edge groups

advanced_settings* - set 1 to enable advanced settings

hotlink_policy - configure hotlink policy properties to protect your content from unauthorized hotlinking:

- BLOCK_BY_DEFAULT - block hotlink_policy by default, except for domains specified in the domains parameter
- NONE - switch off the rule

domains - domains related to hotlink_policy

country_access_policy - configure a rule to control access to the CDN resource’s content for specified countries:

- BLOCK_BY_DEFAULT - block country access policy by default, except for countries specified in the countries parameter
- NONE - switch off the country access policy

countries - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format.

secure_wowza_on - set 1 to enable secure Wowza streaming encryption, otherwise set 0

secure_wowza_token - specify the Wowza token

token_auth_on - indicate whether Token Authentication is enabled for this resource. It’s only supported for VoD PUSH, VoD PULL, and Live Streaming resource.

token_auth_primary_key - set primary key to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters. This must not be blank if token_auth_on is enabled.

token_auth_backup_key - set backup key (optional) to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt (if primary key failed) the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters.

token_auth_secure_paths - set secure paths that mark streaming to these paths requires a valid token to play a stream. Include only the relative path that appears after the content access point. If undefined, forward slash (/) will be set, means the resource is secured at root level and all streaming request to the resource will be granted only if the provided token is valid. Example - ["/video1", "/video2"]

storage_server_location - the ID of the storage server location which should be assigned to this resource. To get the ID of the required storage server location, use the request described at the Get List of Available Storage Locations section. If no location set, the first active storage server is chosen automatically.

Response

You will get a response consisting of two parts - the header with HTTP status code and the response body including the parameters. At this stage some of the parameters can be empty. This is expected behavior, because the full process of creation takes some time, and remote service cannot fill in the parameters at this time. For the complete list of parameters use GET request.

Page History
OnApp Cloud 6.5 Edge 5 CDN Guide

v.3.3.1

- added the following parameters:
  - `token_auth_on`
  - `token_auth_primary_key`
  - `token_auth_backup_key`
  - `token_auth_secure_paths`
  - `storage_server_location`

4.14.10 Add VoD Pull CDN Resource With Advanced Settings

To create a new video on demand resource with advanced settings, use the following request:

POST /cdn_resources.xml
POST /cdn_resources.json

XML Request Example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d 
  http://onapp.test/cdn_resources.xml -d 
  '<cdn_resource><resource_type>STREAM_VOD_PULL</resource_type><cdn_hostname>apitest.com.ua</cdn_hostname>
  <origin>111.111.11.111</origin><edge_group_ids type="array"><edge_group_id>1</edge_group_id><edge_group_id>2</edge_group_id></edge_group_ids><hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy><domains>test.com</domains><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries type="array"><country>AL</country><country>GT</country></countries><secure_wowza_on>1</secure_wowza_on><secure_wowza_token>test123456</secure_wowza_token><token_auth_on>1</token_auth_on><token_auth_primary_key>zsfdfsag</token_auth_primary_key><token_auth_secure_paths type="array"><token_auth_secure_path>/Video1</token_auth_secure_path><token_auth_secure_path>/Video2</token_auth_secure_path></token_auth_secure_paths><token_auth_backup_key>fgff45788787878</token_auth_backup_key><token_auth_secure_paths type="array"><token_auth_secure_path>/Video1</token_auth_secure_path><token_auth_secure_path>/Video2</token_auth_secure_path></token_auth_secure_paths></cdn_resource>'
```

JSON Request Example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d 
  http://onapp.test/cdn_resources.json -d 
  '{"cdn_resource":{"resource_type":"STREAM_VOD_PULL","origin":"111.111.11.111","cdn_hostname":"apitest.com","edge_group_ids":["1","2"],"hotlink_policy":"BLOCK_BY_DEFAULT","domains":"test.com","country_access_policy":"ALLOW_BY_DEFAULT","countries":{"AL","GT"}, "secure_wowza_token":test123456,"token_auth_on":1,"token_auth_primary_key":zsfdfsag,"token_auth_secure_paths": ["/video1", 
  "/video2"],"token_auth_backup_key":fgff45788787878}'}
```

Where:

- `cdn_hostname`* - specify the name which will serve as a label only
  - `resource_type`* - STREAM_VOD_PULL
**origin** - the path from which the CDN requests the content. When using the hostnames according to RFC 1035, the origin may consist of letters [A-Z a-z] (case-insensitive manner), digits [0-9], and dash [-]. The limit for a hostname is 255 chars.

The VoD Pull resource can have only one origin. You cannot use the **origins** attribute while adding a new VoD Pull resource, otherwise, you will get an error message in the API response.

**edge_group_ids** - indicate the ID(s) of required CDN edge groups

**advanced_settings** - set 1 to enable advanced settings

**hotlink_policy** - configure hotlink policy properties to protect your content from unauthorized hotlinking:
- **BLOCK_BY_DEFAULT** - block hotlink_policy by default, except for domains specified in the **domains** parameter
- **NONE** - switch off the rule

**domains** - domains related to hotlink_policy

**country_access_policy** - configure a rule to control access to the CDN resource's content for specified countries:
- **BLOCK_BY_DEFAULT** - block country access policy by default, except for countries specified in the **countries** parameter
- **NONE** - switch off the country access policy

**countries** - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format.

**secure_wowza_on** - set 1 to enable secure Wowza streaming encryption, otherwise set 0

**secure_wowza_token** - specify the Wowza token

**token_auth_on** - indicate whether Token Authentication is enabled for this resource. It's only supported for VoD PUSH, VoD PULL, and Live Streaming resource.

**token_auth_primary_key** - set primary key to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters. This must not be blank if **token_auth_on** is enabled.

**token_auth_backup_key** - set backup key (optional) to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt (if primary key failed) the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters.

**token_auth_secure_paths** - set secure paths that marks streaming to these paths requires a valid token to play a stream. Include only the relative path that appears after the content access point. If undefined, forward slash (/) will be set, means the resource is secured at root level and all streaming request to the resource will be granted only if the provided token is valid. Example - ["/video1", "/video2"]

**Response**

You will get a response consisting of two parts - the header with HTTP status code (see **Introduction** for details) and the response body including the parameters. At this stage, some of the parameters can be empty. This is expected behavior because the full process of creation takes some time, and remote service cannot fill in the parameters at this time. For the complete list of parameters use GET request.
3.3.1 added the following parameters:
- token_auth_on
- token_auth_primary_key
- token_auth_backup_key
- token_auth_secure_paths
- storage_server_location

4.14.11 Add Live Streaming CDN Resource with Advanced Settings

To add live streaming CDN resource with advanced settings, use the following request:

POST /cdn_resources.xml
POST /cdn_resources.json

**XML Request Example (with internal publishing point)**

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-Type: application/xml' -u user:userpass --url http://onapp.test/cdn_resources.xml -d '"<cdn_resource><cdn_hostname>testLV.internal</cdn_hostname><resource_type>STREAM_LIVE</resource_type><publishing_point>internal</publishing_point><publishing_location>532</publishing_location><failover_publishing_location>336</failover_publishing_location><hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy><domains>www.google.com</domains><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries>AO</countries><secure_wowza_on>1</secure_wowza_on><secure_wowza_token>4t534564tyrt</secure_wowza_token><edge_group_ids>224</edge_group_ids><token_auth_on>1</token_auth_on><token_auth_primary_key>zsfdfasga</token_auth_primary_key><token_auth_backup_key>fgff45788787878</token_auth_backup_key><token_auth_secure_paths type="array">/video1</token_auth_secure_path>/video2</token_auth_secure_path></cdn_resource>"'
```

**JSON Request Example (with internal publishing point)**

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-Type: application/json' -u user:userpass --url http://onapp.test/cdn_resources.json -d '{"cdn_resource":{"cdn_hostname":"testLSJ.internal", "resource_type":"STREAM_LIVE", "publishing_point":"internal","publishing_location":"532","failover_publishing_location":"336", "hotlink_policy":"BLOCK_BY_DEFAULT", "domains":"www.google.com", "country_access_policy":"ALLOW_BY_DEFAULT", "countries":"["AO","BH"], "secure_wowza_on":1", "secure_wowza_token":4t534564tyrt", "edge_group_ids":224,"token_auth_on":1,"token_auth_primary_key":"zsfdfasga","token_auth_secure_paths": ["/video1", "/video2"],"token_auth_backup_key":"fgff45788787878"}}'
```

**XML Request Example (with external publishing point)**

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-Type: application/xml' -u user:userpass --url http://onapp.test/cdn_resources.xml -d '"<cdn_resource><cdn_hostname>testLV.external</cdn_hostname><resource_type>STREAM_LIVE</resource_type><publishing_point>external</publishing_point><publishing_location>532</publishing_location><failover_publishing_location>336</failover_publishing_location><hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy><domains>www.google.com</domains><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries>AO</countries><secure_wowza_on>1</secure_wowza_on><secure_wowza_token>4t534564tyrt</secure_wowza_token><edge_group_ids>224</edge_group_ids><token_auth_on>1</token_auth_on><token_auth_primary_key>zsfdfasga</token_auth_primary_key><token_auth_backup_key>fgff45788787878</token_auth_backup_key><token_auth_secure_paths type="array">/video1</token_auth_secure_path>/video2</token_auth_secure_path></cdn_resource>"'
```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d
http://onapp.test/cdn_resources.xml -d
'<cdn_resource><cdn_hostname>testLV1.external</cdn_hostname><resource_type>STREAM_LIVE</resource_type><publishing_point>external</publishing_point><publishing_location>http://www.google.com</publishing_location><failover_publishing_location>rtmp://test.com/test</failover_publishing_location><hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy><domains>www.google.com</domains><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries>AO</countries><countries>BH</countries><secure_wowza_on>1</secure_wowza_on><secure_wowza_token>4t534564tyrt</secure_wowza_token><edge_group_ids>224</edge_group_ids></cdn_resource>'

JSON Request Example (with external publishing point)

curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d
http://onapp.test/cdn_resources.json -d
'{"cdn_resource":{"cdn_hostname":"testLSJ.external","resource_type":"STREAM_LIVE","publishing_point":"external","publishing_location":"http://www.google.com","failover_publishing_location":"rtmp://test.com/test","hotlink_policy":"BLOCK_BY_DEFAULT","domains":"www.google.com","country_access_policy":"ALLOW_BY_DEFAULT","countries":["AO", "BH"],"secure_wowza_on":1,"secure_wowza_token":4t534564tyrt","edge_group_ids":224}}'

Where:
- **cdn_hostname** - specify the name which will serve as a label only
- **resource_type** - STREAM_LIVE
- **publishing_point** - the publishing point type: external or internal
- **publishing_location** - specify the URL address for external publishing point. Set the ID of a location that will serve as a publishing point for internal type. For external type, the field can't be blank, must begin with 'rtmp', contain maximum 255 characters and comply with RFC2396.
- **failover_publishing_location** - publishing point failover URL for external publishing point. Specify the ID of a location that will serve as a failover publishing point for internal type. For external type, the field can't be blank, must begin with 'rtmp', contain maximum 255 characters, and comply with RFC2396. The failover_publishing_location can't be the same as publishing_location parameter.
- **advanced_settings** - set 1 to enable advanced settings
- **hotlink_policy** - configure hotlink policy properties to protect your content from unauthorized hotlinking:
  - **BLOCK_BY_DEFAULT** - block hotlink_policy by default, except for domains specified in the domains parameter
  - **NONE** - switch off the rule
- **domains** - domains related to hotlink_policy
- **country_access_policy** - configure a rule to control access to the CDN resource's content for specified countries:
  - **BLOCK_BY_DEFAULT** - block country access policy by default, except for countries specified in the countries parameter
  - **NONE** - switch off the country access policy
countries - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format.

secure_wowza_on - set 1 to enable secure Wowza streaming encryption, otherwise set 0

secure_wowza_token - specify the Wowza token

token_auth_on - indicate whether Token Authentication is enabled for this resource. It's only supported for VoD PUSH, VoD PULL, and Live Streaming resource.

token_auth_primary_key - set primary key to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters. This must not be blank if token_auth_on is enabled.

token_auth_backup_key - set backup key (optional) to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt (if primary key failed) the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters.

token_auth_secure_paths - set secure paths that marks streaming to these paths requires a valid token to play a stream. Include only the relative path that appears after the content access point. If undefined, forward slash (/) will be set, means the resource is secured at root level and all streaming request to the resource will be granted only if the provided token is valid. Example - ["/video1", "/video2"]

Response

You will get a response consisting of two parts - the header with HTTP status code and the response body including the parameters. At this stage some of the parameters can be empty. This is expected behavior, because the full process of creation takes some time, and remote service cannot fill in the parameters at this time. For the complete list of parameters use GET request.

Page History

v.3.3.1

- added the following parameters:
  - token_auth_on
  - token_auth_primary_key
  - token_auth_backup_key
  - token_auth_secure_paths

4.14.12 Edit Resource

To edit details of the CDN resource, use the following API call:

PUT /cdn_resources/:id.xml
PUT /cdn_resources/:id.json

XML Request example

curl -i -X PUT -u user:userpass -H 'Accept: application/xml' -H 'Content-type: application/xml' --url 'http://onapp.test/cdn_resources/:id.xml' -d '<cdn_resource><edge_group_ids type="array"><edge_group_id type="Integer">1</edge_group_id></edge_group_ids><origin>origin4.com</origin><cdn_hostname>az.test.api</cdn_hostname><cdn_ssl_certificate_id>8</cdn_ssl_certificate_id></cdn_resource>'
**JSON Request example**

```bash
```

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no CDN resource with a requested ID, or URL is incorrect.

**Where you can edit all strings:**

*origin* - the path from which the CDN requests the content. You can specify up to 3 origins. You can specify custom origin port (for HTTP Pull resource only). To use the custom port for resource’s origin, specify a port number using a colon (":"). For example, `<origin>1.2.3.4:80</origin>`.

When you are specifying only one origin, it can be either a CDN hostname or an IP address. In case you are specifying more than one origin, they can only be IP addresses. The VoD Pull CDN resource can have only one origin.

To send two or more origins in the API request, use array. For example:

```xml
<origins type="array">
  <origin>111.111.11.111</origin>
  <origin>111.111.11.111</origin>
</origins>
```

*cdn_hostname* - indicate the hostname which will serve static content

*edge_group_ids* - indicate the ID(s) of required CDN edge groups

### 4.14.13 Edit HTTP Pull CDN Resource with Advanced Settings

To edit HTTP Pull CDN resource:

- PUT `/cdn_resources/:cdn_resource_id.xml`
- PUT `/cdn_resources/:cdn_resource_id.json`

**XML Request example**
curl -i -X PUT -u user:pass --url http://onapp.test/cdn_resources/12710.xml -H 'Accept: application/xml' -H 'Content-type: application/xml' -d "<cdn_resource><cdn_hostname>testnewnewpull.qwe</cdn_hostname><origin>111.111.111.111</origin><cdn_ssl_certificate_id>ssl_sert_id</cdn_ssl_certificate_id><edge_group_ids type="array"><edge_group_id>225</edge_group_id><edge_group_ids type="array"></edge_group_ids><secondary_hostnames type="array"><secondary_hostname>test100.com</secondary_hostname><secondary_hostname>test200.com</secondary_hostname><secondary_hostnames type="array"></secondary_hostnames><ip_access_policy>BLOCK_BY_DEFAULT</ip_access_policy><ip_addresses>111.111.111.22,122.222.222.22</ip_addresses><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries type="array"><country>AL</country><country>GT</country><countries type="array"></country><hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy><domains>abuse.co.ua abuse.org</domains><url_signing_on>1</url_signing_on><url_signing_key>newurlkey</url_signing_key><cache_expiry>45</cache_expiry><password_on>1</password_on><form_pass><user type="array"><string>user123new</string><string>user234new</string></user><pass type="array"><string>passw123new</string><string>passw234new</string></pass></form_pass><password_unauthorized_html>YOU ARE NOT AUTHORIZED</password_unauthorized_html><flv_pseudo_on>1</flv_pseudo_on><mp4_pseudo_on>1</mp4_pseudo_on><cors_on>1</cors_on><limit_rate>100</limit_rate><limit_rate_after>150</limit_rate_after><proxy_cache_key>$host$uri</proxy_cache_key><proxy_read_time_out>60</proxy_read_time_out><proxy_connect_time_out>60</proxy_connect_time_out><http_bot_blocked>1</http_bot_blocked><origin_policy>HTTP</origin_policy></cdn_resource>"

JSON Request example

curl -i -X PUT -H 'Accept: application/json' -H 'Content-type: application/json' -u user:pass --url http://onapp.test/cdn_resources/12711.json -d '{"cdn_resource":{"origin":"test123test.com","cdn_hostname":"jssscdn.apitestpull.com","cdn_ssl_certificate_id":"ssl_cert_id","edge_group_ids":['225'],"secondary_hostnames":['nameltest.com','name2test.com'],'ip_access_policy':'ALLOW_BY_DEFAULT','ip_addresses':'10.10.5.6,125.125.125.125','country_access_policy':'BLOCK_BY_DEFAULT','countries':['AL','AR','GT','HR'],'hotlink_policy':'ALLOW_BY_DEFAULT','domains':'mnw.netggl.com','url_signing_on':1,'url_signing_key':"123456789321","cache_expiry":45,"password_on":1,"form_pass":{"user":['user190','user278']},"password_unauthorized_html":"You are blocked!"},"mp4_pseudo_on":1,"flv_pseudo_on":1,"cors_on":1,"ignore_set_cookie_on":1,"limit_rate":150,"limit_rate_after":1,"proxy_cache_key":$host$uri,"proxy_read_time_out":60,"proxy_connect_time_out":60,"http_bot_blocked":1,"origin_policy":"HTTP"}

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no CDN resource with a requested ID, or URL is incorrect.

Where:

origin* - the path from which the CDN requests the content. When using the hostnames according to RFC 1035, the origin may consist of letters [A-Z a-z] (case insensitive manner), digits [0-9], and dash [-]. The limit for hostname is 255 characters. You can specify up to 3 origins.

cdn_hostname* - indicate the hostname which will serve static content
ssl_on - set to 'true' to enable SSL, otherwise set to 'false'. See the note below for more information.

**SSL**

If the CDN hostname ends with '.r.worldssl.net', SSL will be enabled automatically.

To disable, remove the '.r.worldssl.net' ending and send the "ssl_on":false parameter. To enable, add the '.r.worldssl.net' ending to the cdn_hostname and send the "ssl_on":true parameter.

Be aware that if CDN hostname ends with '.r.worldssl.net', it can not be digit-only (for example 123456.r.worldssl.net is not applicable).

**cdn_ssl_certificate_id** - the ID of the custom SNI SSL certificate you want to add to the resource. You should only specify the IDs of those certificates that were added by the user with whom the new resource will be associated.

**letsencrypt_ssl_on** - set to true to enable Let's Encrypt SSL for the resource, otherwise set to false.

**edge_group_ids** - indicate the ID(s) of required CDN edge groups

**secondary_hostnames** - an array of secondary CDN hostnames. You can add up to 7 secondary CDN hostnames.

```xml
<secondary_hostnames type="array">
  <secondary_hostname>test100.com</secondary_hostname>
  <secondary_hostname>test200.com</secondary_hostname>
</secondary_hostnames>
```

To be able to use a secondary hostname for the CDN resource with SSL enabled, you require an SSL certificate for your custom hostname. For help with questions about the SSL certificate purchase, please contact OnApp support.

**advanced_settings** - set 1 to enable advanced settings:

**ip_access_policy** - configure a rule to control access to the CDN resource's content for a range of IP addresses:

- ALLOW_BY_DEFAULT - allow IP access policy by default, except for IP addresses specified in the ip_addresses parameter
- BLOCK_BY_DEFAULT - block IP access policy by default, except for IP addresses specified in the ip_addresses parameter
- NONE - switch off the IP access policy

**ip_addresses** - IP address(es) related to ip_access_policy parameter; The comma-separated list of IP addresses or IP ranges allowed/blocke by default. Use the following format "10.10.10.10, 20.20.20.0/24"
hotlink_policy - configure hotlink policy properties to protect your content from unauthorized hotlinking:

- ALLOW_BY_DEFAULT - allow hotlink policy by default, except for domains specified in the domains parameter
- BLOCK_BY_DEFAULT - block hotlink policy by default, except for domains specified in the domains parameter
- NONE - switch off the rule

domains - domains related to hotlink_policy

hls_on - set to 1 to enable HTTP Live Streaming (HLS) Optimization

hls_force_cache - set to 1 to create an HTTP rule that will enforce cache expiry. This option is only available when the hls_on parameter is set to 1

country_access_policy - configure a rule to control access to the CDN resource's content for specified countries:

- ALLOW_BY_DEFAULT - allow country access policy by default, except for countries specified in the countries parameter
- BLOCK_BY_DEFAULT - block country access policy by default, except for countries specified in the countries parameter
- NONE - switch off the country access policy

countries - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format.

cache_expiry - set the cache expiry time in minutes (min=1, max=71582788)

url_signing_on - set 1 to enable and protect your files from unauthorized access with a key

url_signing_key - input the key for URL signing. Input letters and digits (6-32 symbols).

password_on - set 1 to enable and to restrict access to the resource (cdn_hostname), otherwise set 0

form_pass - an array with usernames and passwords to access the resource

pass - the user password.

user - the user login, which may consist of letters [A-Z a-z] (case insensitive manner), digits [0-9], underscore [ _ ], dash [ - ]. The first symbol should be alphabetic. The username cannot be duplicated.

password_unauthorized_html - text, which will be displayed in case of fail of authentication. Max 1000 chars.

mp4_pseudo_on - set 1 to enable MP4 pseudo streaming, otherwise set 0

flv_pseudo_on - set 1 to enable FLV pseudo streaming, otherwise set 0

cors_on - set 1 to enable cross-origin resource sharing (CORS), otherwise set 0

ignore_set_cookie_on - set 1 to enable caching content with Set-Cookie response headers, otherwise set 0 to ignore content caching

Nginx Settings

- limit_rate - sets speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
- limit_rate_after - sets the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value -2147483647 KB
- proxy_cache_key - specify the cache key. You can set the following options:
  - $host$request_uri
  - $host$uri
OnApp Cloud 6.5 Edge 5 CDN Guide

- $proxy_host$request_uri
- $proxy_host$uri

http_bot_blocked - set 1 to block Google web crawling bot from indexing the CDN content (for HTTP Pull CDN resources only)

origin_policy - set the parameter to choose the type of connection. Possible values are: HTTP, HTTPS, AUTO.

4.14.13.1 Page history
v.6.1
- Added the cors_on parameter

v 5.6
- Added letsencrypt_ssl_on parameter that enables Let's Encrypt SSL for a CDN resource
- Removed proxy_read_time_out and proxy_connect_time_out from CDN advanced settings

v 5.1
- added the hls_force_cache parameter
- added the hls_on parameter

v. 4.0
- Added cdn_ssl_certificate_id parameter that enables a user to associate a custom SNI SSL certificate with a CDN resource

v. 3.3.1
Added the following parameter:
- origin_policy


To edit HTTP Push CDN resource:

PUT /cdn_resources/:cdn_resource_id.xml
PUT /cdn_resources/:cdn_resource_id.json

XML Request example
JSON Request example

```
curl -i -X PUT -u user:userpass http://onapp.test/cdn_resources/12713.json -H 'Accept: application/json' -H 'Content-type: application/json' -d '{"cdn_resource":{"cdn_hostname":"Ycdn.apitestruhu.com","cdn_ssl_certificate_id":"ssl_cert_id","edge_group_ids":['"225"'],"ftp_password":"password123","secondary_hostnames":['"name1test12.co","name2test34.co"]","ip_access_policy":"ALLOW_BY_DEFAULT","ip_addresses":"10.10.5.6,125.125.125.125","country_access_policy":"BLOCK_BY_DEFAULT","countries":['"AL","AR","GT","HR"],"hotlink_policy":"ALLOW_BY_DEFAULT","domains":['"mnw.netggl12.com"'],"url_signing_on":1,"url_signing_key":newtest123456789321,"password_on":1,"form_pass":{"user":['"user190","user278"'],"pass":['"pass123","pass2348"']}},"password_unauthorized_html":"<b>You are blocked!</b>","mp4_pseudo_on":1,"flv_pseudo_on":1,"cors_on":1,"limit_rate":140,"limit_rate_after":11}}'
```

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no CDN resource with a requested ID, or URL is incorrect.

Where:

* origin* - the path from which the CDN requests the content

* cdn_hostname* - indicate the hostname which will serve static content

* ssl_on* - set to 'true' to enable SSL, otherwise set to 'false'. See the note below for more information.

**SSL**
If the CDN hostname ends with '.r.worldssl.net', SSL will be enabled automatically.

To disable, remove the '.r.worldssl.net' ending and send the "ssl_on":false parameter. To enable, add the '.r.worldssl.net' ending to the cdn_hostname and send the "ssl_on":true parameter.

Be aware that if CDN hostname ends with '.r.worldssl.net', it can not be digit-only (for example 123456.r.worldssl.net is not applicable).

cdn_ssl_certificate_id - the ID of the custom SNI SSL certificate you want to add to the resource. You should only specify the IDs of those certificates that were added by the user with whom the new resource will be associated.

letsencrypt_ssl_on - set to true to enable Let's Encrypt SSL for the resource, otherwise set to false.

danger group ids* - indicate the ID(s) of required CDN edge groups
tp_password* - indicate the FTP server password. It should consist of 6-32 alphanumeric symbols.

secondary_hostnames - an array of secondary CDN hostnames. You can add up to 7 secondary CDN hostnames.

    <secondary_hostnames type="array">
        <secondary_hostname>test100.com</secondary_hostname>
        <secondary_hostname>test200.com</secondary_hostname>
    </secondary_hostnames>

To be able to use a secondary hostname for the CDN resource with SSL enabled, you require an SSL certificate for your custom hostname. For help with questions about the SSL certificate purchase, please contact OnApp support.

advanced_settings* - set 1 to enable advanced settings:

    ip_access_policy - configure a rule to control access to the CDN resource's content for a range of IP addresses:
        - ALLOW_BY_DEFAULT - allow IP access policy by default, except for IP addresses specified in the ip_addresses parameter
        - BLOCK_BY_DEFAULT - block IP access policy by default, except for IP addresses specified in the ip_addresses parameter
        - NONE - switch off the IP access policy

    ip_addresses - IP address(es) related to ip_access_policy parameter; the comma-separated list of IP addresses or IP ranges allowed/blocked by default. Use the following format "10.10.10.10, 20.20.20.0/24"

hotlink_policy - configure hotlink policy properties to protect your content from unauthorized hotlinking:
- ALLOW_BY_DEFAULT - allow hotlink policy by default, except for domains specified in the `domains` parameter
- BLOCK_BY_DEFAULT - block hotlink policy by default, except for domains specified in the `domains` parameter
- NONE - switch off the rule

`domains` - domains related to hotlink policy

`country_access_policy` - configure a rule to control access to the CDN resource's content for specified countries:
- ALLOW_BY_DEFAULT - allow country access policy by default, except for countries specified in the `countries` parameter
- BLOCK_BY_DEFAULT - block country access policy by default, except for countries specified in the `countries` parameter
- NONE - switch off the country access policy

`countries` - country codes, related to `country_access_policy` in ISO 3166-1 alpha-2 format.

`cache_expiry` - set the cache expiry time in minutes

`url_signing_on` - set 1 to enable and protect your files from unauthorized access with a key

`url_signing_key` - input the key for URL signing. Input letters and digits (6-32 symbols).

`password_on` - set 1, if the access to the resource is restricted; otherwise 0

`form_pass` - an array with usernames and passwords to access the resource

`pass` - the user password.

`user` - the user login, which may consist of letters [A-Z a-z] (case insensitive manner), digits [0-9], underscore [ _ ], dash [-]. The first symbol should be alphabetic. The username cannot be duplicated.

`password_unauthorized_html` – the message that is displayed when there is unauthorized access. Max 1000 chars.

`mp4_pseudo_on` - set 1 to enable MP4 pseudo streaming, otherwise set 0

`flv_pseudo_on` - set 1 to enable FLV pseudo streaming, otherwise set 0

`cors_on` - set 1 to enable cross-origin resource sharing (CORS), otherwise set 0

`ignore_set_cookie_on` - set 1 to enable caching content with Set-Cookie response headers, otherwise set 0

**Nginx Settings**

- `limit_rate` - sets speed limit of a response to a client (per request) in KB/s. Maximum limit rate value - 2147483647 KB/s
- `limit_rate_after` - sets the amount after which the speed of a response to a client will be limited in KB. Maximum limit rate after value - 2147483647 KB

---


v.6.1

- Added the `cors_on` parameter

v. 5.6

- Added `letsencrypt_ssl_on` parameter that enables Let's Encrypt SSL for a CDN resource

v. 4.0

- Updated `cdn_hostname` parameter with ability to enable or disable SSL
• Added `cdn_ssl_certificate_id` parameter that enables a user to associate a custom SNI SSL certificate with a CDN resource.

### 4.14.15 Edit VoD Push CDN Resource with Advanced Settings

To create new video on demand resource with advanced settings, use the following request:

**PUT /cdn_resources/:cdn_resource_id.xml**
**PUT /cdn_resources/:cdn_resource_id.json**

**XML Request Example**

```
curl -i -X PUT -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url http://onapp.test/cdn_resources/12.xml -d '<cdn_resource><cdn_hostname>apitest.com</cdn_hostname><ftp_password>testpassword</ftp_password><edge_group_ids type="array"><edge_group_id>466</edge_group_id><edge_group_id>1</edge_group_id></edge_group_ids><hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy><domains>goo.coabuse.ua</domains><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries type="array"><country>AL</country><country>GT</country></countries><secure_wowza_on>1</secure_wowza_on><secure_wowza_token>test123456</secure_wowza_token><token_auth_on>1</token_auth_on><token_auth_primary_key>zsfdfasga</token_auth_primary_key><token_auth_backup_key>fgff4578787878</token_auth_backup_key><token_auth_secure_paths type="array"><token_auth_secure_path>/Video1</token_auth_secure_path><token_auth_secure_path>/Video2</token_auth_secure_path></token_auth_secure_paths></cdn_resource>
```

**JSON Request Example**

```
```

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no CDN resource with a requested ID, or URL is incorrect.

**Where:**

- `cdn_hostname`* - specify the name which will serve as a label only
- `ftp_password`* - indicate the FTP server password. It should consist of 6-32 alphanumeric symbols.
- `edge_group_ids`* - indicate the ID(s) of required CDN edge groups
- `advanced_settings`* - set 1 to enable advanced settings:
- `hotlink_policy` - configure hotlink policy properties to protect your content from unauthorized hotlinking:
• BLOCK_BY_DEFAULT - block hotlink_policy by default, except for domains specified in the domains parameter
• NONE - switch off the rule
domains - domains related to hotlink policy
country_access_policy - configure a rule to control access to the CDN resource's content for specified countries:
  • BLOCK_BY_DEFAULT - block country access policy by default, except for countries specified in the countries parameter
  • NONE - switch off the country access policy
countries - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format.
secure_wowza_on - set 1 to enable secure Wowza streaming encryption, otherwise set 0
secure_wowza_token - specify the Wowza token
token_auth_on - indicate whether Token Authentication is enabled for this resource. It's only supported for VoD PUSH, VoD PULL, and Live Streaming resource.
token_auth_primary_key - set primary key to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters. This must not be blank if token_auth_on is enabled.
token_auth_backup_key - set backup key (optional) to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt (if primary key failed) the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters.
token_auth_secure_paths - set secure paths that marks streaming to these paths requires a valid token to play a stream. Include only the relative path that appears after the content access point. If undefined, forward slash (/) will be set, means the resource is secured at root level and all streaming request to the resource will be granted only if the provided token is valid. Example - ["/video1", "/video2"]

Page History
v.3.3.1
• added the following parameters:
  o token_auth_on
  o token_auth_primary_key
  o token_auth_backup_key
  o token_auth_secure_paths
  o storage_server_location

4.14.16 Edit VoD Pull CDN Resource with Advanced Settings
To create new video on demand resource with advanced settings, use the following request:
PUT /cdn_resources/:cdn_resource_id.xml
PUT /cdn_resources/:cdn_resource_id.json
XML Request Example
curl -i -X PUT -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url
http://onapp.test/cdn_resources/12.xml -d
'"cdn_resource"<cdn_hostname>apitest.com</cdn_hostname><origin>111.111.11.11</origin><edge_group_ids
type="array"><edge_group_id>466</edge_group_id><edge_group_id>1</edge_group_id></edge_group_ids><hotlink_policy>BLOCK_BY_DEFAULT</hotlink_policy><domains>goo.coabuse.ua</domains><country_access_policy>ALLOW_BY_DEFAULT</country_access_policy><countries
type="array"><country>AL</country><country>GT</country></countries><secure_wowza_on></secure_wowza_on><secure_wowza_token>test123456</secure_wowza_token><token_auth_on>1</token_auth_on><token_auth_token>zsfdfasga</token_auth_token><token_auth_backup_key>fgff45788787878</token_auth_backup_key><token_auth_secure_path>/video1</token_auth_secure_path><token_auth_secure_path>/video2</token_auth_secure_path>
</cdn_resource>"

JSON Request Example

curl -i -X PUT -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url
http://onapp.test/cdn_resources/12.json -d
'{"cdn_resource":{"origin":"test.com","cdn_hostname":"ohcdn235.apitest.com","edge_group_ids":["1","2"],"hotlink_policy":"BLOCK_BY_DEFAULT","domains":"test.com","country_access_policy":"ALLOW_BY_DEFAULT","countries":["AL","GT"],"secure_wowza_on":1,"secure_wowza_token":test123456,"token_auth_on":1,"token_auth_token":zsfdfasga,"token_auth_backup_key":fgff45788787878,"token_auth_secure_paths": ["/video1","/video2"],"token_auth_secure_path":/video1","token_auth_secure_path":/video2"}

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no CDN resource with a requested ID, or URL is incorrect.

Where:

- **cdn_hostname** - specify the name which will serve as a label only
- **origin** - the path from which the CDN requests the content. When using the hostnames according to RFC 1035, the origin may consist of letters [A-Z a-z] (case insensitive manner), digits [0-9], and dash [-]. The limit for hostname is 255 chars.

The VoD Pull resource can have only one origin. You cannot use the **origins** attribute while editing the VoD Pull resource, otherwise, you will get an error message in the API response.

- **edge_group_ids** - indicate the ID(s) of required CDN edge groups
- **advanced_settings** - set 1 to enable advanced settings:
- **hotlink_policy** - configure hotlink policy properties to protect your content from unauthorized hotlinking:
  - BLOCK_BY_DEFAULT - block hotlink_policy by default, except for domains specified in the **domains** parameter
- NONE - switch off the rule

**domains** - domains related to hotlink policy

**country_access_policy** - configure a rule to control access to the CDN resource's content for specified countries:

- **BLOCK_BY_DEFAULT** - block country access policy by default, except for countries specified in the **countries** parameter
- **NONE** - switch off the country access policy

**countries** - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format

**secure_wowza_on** - set 1 to enable secure Wowza streaming encryption, otherwise set 0

**secure_wowza_token** - specify the Wowza token

**token_auth_on** - indicate whether Token Authentication is enabled for this resource. It's only supported for VoD PUSH, VoD PULL, and Live Streaming resource.

**token_auth_primary_key** - set primary key to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters. This must not be blank if token_auth_on is enabled.

**token_auth_backup_key** - set backup key (optional) to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt (if primary key failed) the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters.

**token_auth_secure_paths** - set secure paths that marks streaming to these paths requires a valid token to play a stream. Include only the relative path that appears after the content access point. If undefined, forward slash (/) will be set, means the resource is secured at root level and all streaming request to the resource will be granted only if the provided token is valid. Example - ["/video1","/video2"]

**Page History**

v.3.3.1

- added the following parameters:
  - **token_auth_on**
  - **token_auth_primary_key**
  - **token_auth_backup_key**
  - **token_auth_secure_paths**
  - **storage_server_location**

**4.14.17 Edit Live Streaming CDN Resource with Advanced Settings**

To edit live streaming CDN resource with advanced settings, use the following request:

PUT /cdn_resources/:id.xml
PUT /cdn_resources/:id.json

**XML Request Example**

JSON Request Example


Returns HTTP 204 response on successful processing, and HTTP 404 when there is no CDN resource with a requested ID, or URL is incorrect.

Where:

- **cdn_hostname** - specify the name which will serve as a label only
- **edge_group_ids** - indicate the ID(s) of required CDN edge groups
- **advanced_settings** - set 1 to enable advanced settings:
- **hotlink_policy** - configure hotlink policy properties to protect your content from unauthorized hotlinking:
  - **BLOCK_BY_DEFAULT** - block hotlink_policy by default, except for domains specified in the **domains** parameter
  - **NONE** - switch off the rule
- **domains** - domains related to hotlink policy
- **country_access_policy** - configure a rule to control access to the CDN resource's content for specified countries:
  - **BLOCK_BY_DEFAULT** - block country access policy by default, except for countries specified in the **countries** parameter
  - **NONE** - switch off the country access policy
countries - country codes, related to country_access_policy in ISO 3166-1 alpha-2 format.

secure_wowza_on - set 1 to enable secure Wowza streaming encryption, otherwise set 0

secure_wowza_token - specify the Wowza token

publishing_point* - the publishing point type: external or internal. For a resource with the external publishing point, this parameter is automatically completed if not specified in the API request.

publishing_location* - specify the URL address for external publishing point. Set the ID of a location that will serve as a publishing point for internal type. For external type, the field can't be blank, must begin with 'rtmp', contain maximum 255 characters and comply with RFC2396.

failover_publishing_location* - publishing point failover URL for external publishing point. Specify the ID of a location that will serve as a failover publishing point for internal type. For external type, the field can't be blank, must begin with 'rtmp', contain maximum 255 characters, and comply with RFC2396. The failover_publishing_location can't be the same as publishing_location parameter.

token_auth_on - indicate whether Token Authentication is enabled for this resource. It's only supported for VoD PUSH, VoD PULL, and Live Streaming resource.

token_auth_primary_key - set primary key to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters. This must not be blank if token_auth_on is enabled.

token_auth_backup_key - set backup key (optional) to encrypt Token Authentication security parameters to generate token for streaming playback. This same key is used to decrypt (if primary key failed) the token at Wowza Server. The key must be alphanumeric with length 6 to 32 characters.

token_auth_secure_paths - set secure paths that marks streaming to these paths requires a valid token to play a stream. Include only the relative path that appears after the content access point. If undefined, forward slash (/) will be set, means the resource is secured at root level and all streaming request to the resource will be granted only if the provided token is valid. Example - ["/video1", "/video2"]

Page History
v.3.3.1
- added the following parameters:
  - token_auth_on
  - token_auth_primary_key
  - token_auth_backup_key
  - token_auth_secure_paths

4.14.18 Delete CDN Resource

To delete a CDN resource, use the following request:

DELETE /cdn_resources/:id.xml
DELETE /cdn_resources/:id.json

XML Request Example

```
curl -i -X DELETE -u user:userpass http://onapp.test /cdn_resources/5.xml
```

JSON Request Example
Where you have to specify ID of a CDN resource you want to delete.
Returns HTTP 204 response on successful processing, and HTTP 404 when there is no CDN resource with a requested ID, or URL is incorrect.

4.14.19 Change CDN Resource FTP Password
To change FTP password of HTTP Push and VoD PUSH resources, use the following request:

PUT /cdn_resources/:cdn_resource_id.xml
PUT /cdn_resources/:cdn_resource_id.json

XML Request Example

```bash
```

JSON Request Example

```bash
```

Where:

- `cdn_resource_id` - CDN resource ID
- `ftp_password` - required FTP password

4.14.20 Prefetch CDN Resource Content
To pre-populate HTTP PULL and PUSH content to the CDN, use the following request:

POST /cdn_resources/:id/prefetch.xml
POST /cdn_resources/:id/prefetch.json

PLEASE NOTE: You can only prefetch content of HTTP CDN resources.

You can use prefetch CDN resource content API requests with entry slashes:

```
<prefetch_paths>/home/123.jpeg</prefetch_paths>
```

as well as without them:
4.14.21 Purge CDN Resource Content

To remove content from HTTP Pull and PUSH cache, use the following request:

POST /cdn_resources/:id/purge.xml
POST /cdn_resources/:id/purge.json
PLEASE NOTE: You can only purge content of HTTP CDN resources.

```
<purge_paths>/home/123.jpeg</purge_paths>
```

You can use purge CDN resource content API requests with entry slashes:
as well as without them:

```
<purge_paths>home/123.jpeg</purge_paths>
```

**XML Request Example**

```
curl -i -X POST -u user:userpass
http://onapp.test/cdn_resources/:cdn_resource_id/purge.xml -d
'"purge_paths":/home/123.jpeg"' -H 'Accept:application/xml' -H 'Content-type:application/xml'
```

**JSON Request Example**

```
curl -i -X POST -u user:userpass
http://onapp.test/cdn_resources/:cdn_resource_id/purge.json -d
'fabric"purge_paths":/home/123.jpeg"' -H 'Accept:application/json' -H 'Content-type:application/json'
```

Where:

- **purge_path** – path to the content you want to remove

To purge an array of paths, use the following request:

**XML Request Example**

```
curl -i -X POST -u user:userpass
http://onapp.test/cdn_resources/12/purge.xml -d
```

**JSON Request Example**

```
curl -i -X POST -u user:userpass
http://onapp.test/cdn_resources/12/purge.json -d
```
Where you have to specify an array of paths to the content you want to remove.

To purge all content, use the following request:

**XML Request Example**

```
curl -i -u user:userpass -X POST
http://onapp.test/cdn_resources/12/purge_all.xml
-H 'Accept: application/xml' -H 'Content-type: application/xml'
```

**JSON Request Example**

```
curl -i -u user:userpass -X POST
http://onapp.test/cdn_resources/12/purge_all.json
-H 'Accept:application/json' -H 'Content-type:application/json'
```

Where you have to specify a CDN resource ID.

If you are using version 3.0 or earlier, use the following request:

**XML Request Example**

```
curl -i -u user:userpass -X POST
http://onapp.test/cdn_resources/12/purge.xml
-d '<purge_paths>/ui3/test1\r\n/ui3/test2\r\n/ui3/test3\r\n/ui3/test4</purge_paths>'
-H 'Accept:application/xml' -H 'Content-type:application/xml'
```

**JSON Request Example**

```
curl -i -X POST -u user:userpass
http://onapp.test/cdn_resources/12/purge.json
-d '{"purge_paths": ["/ui3/test1\r\n/ui3/test2\r\n/ui3/test3\r\n/ui3/test4"]}'
-H 'Accept:application/json' -H 'Content-type:application/json'
```

### 4.14.22 View CDN Resource Bandwidth Statistics

The following requests are deprecated in OnApp 5.4-5.5 and are completely removed from OnApp 5.6.

To get bandwidth statistics for the resources or a particular resource, use the following request:
Bandwidth statistics is returned in gigabytes in the SI format (1 GB = 1000 MB).

You can also define a shorter period, specify a particular resource or location, set the type and specify how to sort the result returned:

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

Where:
- `start` – the start date to generate statistics in the YYYY-MM-DD+hh:mm:ss format
- `end` – the end date to generate statistics in the YYYY-MM-DD+hh:mm:ss format
- `resources` – the identifier of the resource in Aflexi database. To get the identifier, check with `cdn_reference` parameter in the GET /cdn_resources/:id.{format} request
- `locations` – the ID of the location
- `type` – the statistics type (MBPS or GB). In MBPS mode you can get statistics for the last 10 days only. The older statistics is removed. There are no restrictions for GB mode.
- `group_by` - to get the bandwidth statistics breaking down per location or per resource via API, use the `group_by` parameter with two possible values: `location` and `resource`. In case `group_by = location` bandwidth stats is breaking down per location, and if `group_by = resource` is breaking down per resource. This is the optional parameter.

**XML Output Example with resources grouped by resource**

```xml
<bandwidth><type>GB</type><group_by>location</group_by><start>2014-11-25T18:25</start><end>2014-11-27T11:43</end><locations type="array"><location>532</location><location>331</locations><resources type="array"><resource>977655738</resource><resource>389478438</resource><resource>725618714</resource></resources></bandwidth>
```
XML Output Example with resources without the `group_by` parameter

```xml
<stats type="array">
  <stat>
    <date type="datetime">2014-11-25T18:25:00+02:00</date>
    <resources type="array">
      <resource>
        <977655738>
          <cached type="float">2.0390744709999997</cached>
          <non_cached type="float">0.2265379899999998</non_cached>
        </977655738>
      </resource>
      <resource>
        <725618714>
          <cached type="float">1.187458449</cached>
          <non_cached type="float">0.13193981700000001</non_cached>
        </725618714>
      </resource>
      <resource>
        <389478438>
          <cached type="float">0.425808007</cached>
          <non_cached type="float">0.047311991</non_cached>
        </389478438>
      </resource>
    </resources>
  </stat>
  <stat>
    <date type="datetime">2014-11-26T18:25:00+02:00</date>
    <resources type="array">
      <resource>
        <977655738>
          <cached type="float">2.8387202499999997</cached>
          <non_cached type="float">0.0</non_cached>
        </977655738>
      </resource>
      <resource>
        <389478438>
          <cached type="float">0.425808007</cached>
          <non_cached type="float">0.047311991</non_cached>
        </389478438>
      </resource>
    </resources>
  </stat>
</stats>
```

Where:

- `non_cached` – the amount of content which is not cached
- `cached` – the amount of data cached
**date** – the point of time for which the statistics is generated

The frequency of the points of time for which the statistics is generated depends on the period of time of requested statistics:

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Frequency in seconds</th>
<th>Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 30 days</td>
<td>86400</td>
<td>1 day</td>
</tr>
<tr>
<td>31 - 93 days</td>
<td>604800</td>
<td>1 week</td>
</tr>
<tr>
<td>equal or more than 93 days</td>
<td>1209600</td>
<td>2 weeks</td>
</tr>
</tbody>
</table>

Statistics available in the frequency higher than selected will be accumulated to a single point of applicable frequency. E.g. The statistics was requested for the period of 31-93 days, so the frequency of points in the graph is 7 days. If the statistics was generated few times during those 7 days (day1+day2+day_n) it will be added up and displayed as a single point, with a time stamp marked as the first day of such 7 days.

### 4.14.23 View CDN Resource Billing Statistics

To view billing statistics for a resource, use the following request:

GET /cdn_resources/:id/billing.xml
GET /cdn_resources/:id/billing.json

You can also define a shorter period by setting Start and End date (set use_local_time to 1 to use local time):


**XML Output Example**
HTTP/1.1 200 OK
Date: Thu, 01 Nov 2012 08:46:57 GMT
Server: Apache/2.2.3 (CentOS)
X-Powered-By: Phusion Passenger (mod_rails/mod_rack) 3.0.9
X-UA-Compatible: IE=Edge,chrome=1
ETag: "5639dcb4af97cc6b5614f0e7102414a6"
Cache-Control: must-revalidate, private, max-age=0
X-Request-Id: b4a84f305441fa8e791a2b597ed0c8e
X-Runtime: 0.064716
X-Rack-Cache: miss
Set-Cookie: _session_id=e01cb0aac4fc66b5dcb5bb29496d12e; path=/; HttpOnly
Status: 200
Connection: close
Transfer-Encoding: chunked
Content-Type: application/xml; charset=utf-8

<?xml version="1.0" encoding="UTF-8"?>
<user_hourly_stats type="array">
  <user_hourly_stat>
    <cost type="float">1.51980002556229e-05</cost>
    <edge_group_id type="integer">78</edge_group_id>
    <edge_group_label>OH_203</edge_group_label>
    <stat_time type="datetime">2012-10-10T14:00:00+03:00</stat_time>
    <value type="decimal">0.0</value>
  </user_hourly_stat>
</user_hourly_stats>

Where:

cost - the total due for this resource
edge_group_id – the ID of the edge group
edge_group_label – the label of the edge group
stat_time - time when the statistics was gathered
value - traffic value


To view the raw log configuration, use the following request:
GET /cdn_resources/raw_log.xml
GET /cdn_resources/raw_log.json

XML Request Example

```
```

JSON Request Example

```
```

XML Output Example

Disabled:
Where:

*protocol* - delivery protocol

For the *FTP/SFTP* delivery protocol:

```xml
<raw_log>
  <protocol>ftp</protocol>
  <uri>rawlog.com</uri>
  <user>username</user>
  <pass>password</pass>
</raw_log>
```

Where:

*protocol* - delivery protocol

*uri* - the hostname of the server to which the log will be delivered

*user* - the user name of the FTP/SFTP client on the server to which the log will be delivered

*pass* - the password of the FTP/SFTP client on the server to which the log will be delivered

For the *Syslog* delivery protocol:

```xml
<raw_log>
  <protocol>syslog</protocol>
  <uri>rawlog.com</uri>
  <syslog_protocol>tcp</syslog_protocol>
  <port>80</port>
</raw_log>
```

Where:

*protocol* - delivery protocol

*uri* - the hostname of the server to which the log will be delivered

*syslog_protocol* - the protocol that will be used for sending the log: TCP or UDP

*port* - the port number of the syslog the server to which the log will be delivered

### 4.14.25 Edit CDN Resource Raw Log Configuration

To edit the raw log configuration, use the following request:

```
POST /cdn_resouces/raw_log.json
```

To disable raw logs:

XML Request Example

**JSON Request Example**

```
```

Where:
- **protocol** - delivery protocol

To configure the FTP/SFTP delivery protocol, use the following request:

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where:
- **protocol** - delivery protocol
- **uri** - the hostname of the server to which the log will be delivered
- **user** - the user name of the FTP/SFTP client on the server to which the log will be delivered
- **pass** - the password of the FTP/SFTP client on the server to which the log will be delivered

To configure the Syslog delivery protocol, use the following request:

**XML Request Example**

```
```

**JSON Request Example**

```
```
JSON Request Example

```
```

Where:

- `protocol` - delivery protocol
- `uri` - the hostname of the server to which the log will be delivered
- `syslog_protocol` - the protocol that will be used for sending the log: TCP or UDP
- `port` - the port number of the syslog server to which the log will be delivered

### 4.14.26 Get Instruction for Live Streaming CDN Internal Resource

To get instruction for live streaming CDN internal resource, use the following request:

GET `/cdn_resources/:cdn_resource_id/instructions.xml`
GET `/cdn_resources/:cdn_resource_id/instructions.json`

**XML Request Example**

```
curl -i -X GET -u user:userpass
http://onapp.test/cdn_resources/11639/instructions.xml
```

**JSON Request Example**

```
curl -i -X GET -u user:userpass
http://onapp.test/cdn_resources/11639/instructions.json
```

**XML Output Example**
Where:

*cdn_hostname* - indicate the hostname which will serve the content

cname - a CNAME for the CDN Hostname which can then be used to view the contents.

*resource_type* - the Stream Live resource

*id* - the ID of the resource

*user_id* - the user ID

*last_24h_cost* - cost of the resource for the last 24 hours.

*cdn_reference* - the ID in OnApp Dashboard

*publishing_point* - the publishing point type: internal

*instructions* - the array of parameters for embedding video and enabling live streaming for Live Streaming CDN resources.

*credentials* - the array with user credentials:

  *username* - the user log in

  *password* - the user password

  *stream* - the stream name
urls - the array with URLs
fms - the FMS URL
backup - the Backup URL
smil - The SMIL playlist provides an RTMP URL and should be used with Flash-based players only.
apple_http_live_streaming - This returns a 302 redirect to an Apple HLS manifest and should be used with Apple HLS-compatible players only.
adobe_http_dynamic_streaming - This returns an Adobe HDS manifest and should be used with Adobe HDS-compatible players only.
microsoft_smooth_streaming - This returns a 302 redirect to Smooth Streaming manifest and should be used with Smooth Streaming-compatible players only.

4.14.27 Search CDN Resource
To search for a specific CDN resource, use the following request:
GET /cdn_resources.xml?q=search_param
GET /cdn_resources.json?q=search_param
The request will search the CDN hostname, origin (both the hostname and an IP address, including redundant origins). For Live Stream CDNs, the request searches the pattern inside Publishing Location fields (main and failover) and Aflexi ID.

XML Request Example
```
```

JSON Request Example
```
```
The example request will search all the CDN resources where the 111.111.111.1 IP is used.

XML output example:
<cdn_resources type="array">
  <cdn_resource>
      <cdn_hostname>oldtestnewnewpull.qwe</cdn_hostname>
      <cname>438335686.r.worldcdn-beta.net</cname>
      <created_at type="datetime">2013-07-29T13:58:55+03:00</created_at>
      <id type="integer">10375</id>
      <resource_type>HTTP_PULL</resource_type>
      <updated_at type="datetime">2013-08-20T17:18:21+03:00</updated_at>
      <user_id type="integer">20</user_id>
      <last_24h_cost type="float">0.0</last_24h_cost>
      <cdn_reference type="integer">438335686</cdn_reference>
      <origins type="array">
        <origin>111.111.11.111</origin>
      </origins>
    </cdn_resource>
</cdn_resources>

Where:

cdn_hostname - the hostname which will serve static content

cname - CNAME record

created_at - the date when the resource was created

id – the resource ID in the database

resource_type – HTTP PULL or PUSH

updated_at – the date when the resource was updated

user_id – the ID of the user, who owns the resource

last_24h_cost – the amount of money owed for the resource for the last 24 hours.

cdn_reference - the identifier in database

origins – the path from which the CDN requests the content

4.14.28 Suspend CDN Resource

To suspend a specific CDN resource, use the following request:

PUT /cdn_resources/:resource_id/suspend.xml
PUT /cdn_resources/:resource_id/suspend.json

XML Request Example

curl -i -X PUT -u user:userpass --url
http://onapp.test/cdn_resources/34/suspend.xml
-H 'Accept: application/xml' -H 'Content-type: application/xml'

JSON Request Example

curl -i -X PUT -u user:userpass --url
http://onapp.test/cdn_resources/34/suspend.json
-H 'Accept: application/json' -H 'Content-type: application/json'

Where:

resource_id - the ID of the CDN resource which you wish to suspend.
You will get a 204 status response on success, and 404 if there is no such CDN Resource with a requested ID or you entered incorrect ID.

### 4.14.29 Resume CDN Resource

To resume a specific suspended CDN resource, use the following request:

```
PUT /cdn_resources/:resource_id/resume.xml
PUT /cdn_resources/:resource_id/resume.json
```

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

Where:

resource_id - the ID of the CDN resource which you wish to resume.

You will get a 204 status response on success, and 404 if there is no such CDN Resource with a requested ID or you entered incorrect ID.

### 4.14.30 View CDN Resource Advanced Reporting

The following requests are deprecated in OnApp 5.4-5.5 and are completely removed from OnApp 5.6.

To view CDN advanced reporting for HTTP PULL and HTTP PUSH resources, use the following request:

```
GET /cdn_resources/:resource_id/advanced_reporting.xml
GET /cdn_resources/:resource_id/advanced_reporting.json
```

**Get Advanced Bandwidth Reporting (Including Cache Utilization) XML Request Example**

```bash
```

**Get Advanced Bandwidth Reporting (Including Cache Utilization) JSON Request Example**

Where:
report - the array with the time points for generating statistics
start - the start date of the period for which the statistics should be generated
end - the end date of the period for which the statistics should be generated
locations - the location ID for which the statistics should be generated

The default request returns Bandwidth and Caching report for the last five days covering all Locations.

Default XML Request

```bash
```

Get Advanced Status Codes Reporting XML API Request Example
(for HTTP Pull request only)

```bash
```

Get Advanced Status Codes Reporting JSON API Request Example
(for HTTP Pull request only)

```bash
```

Where:
report - the array with the time points for generating statistics
stats_type* - required parameter, in this case status_codes
start - the start date of the period for which the statistics should be generated
end - the end date of the period for which the statistics should be generated
locations - the location ID for which the statistics should be generated

4.14.31 Get List of Available Storage Locations

To create a CDN PUSH resource, it is necessary to specify a storage location. To view the list of available storage locations, run the following request:

GET /cdn_resources/available_storage_server_locations.xml
GET /cdn_resources/available_storage_server_locations.json

XML Request Example

```
```

JSON Request Example

```
```

Where

type - the optional parameter which describes the Storage Server’s type - streaming or http;
only_active - the optional parameter which describes the Storage Server’s status (only_active = true if only all active SSs or only_active = false if all the available SSs).

On success HTTP 200 status is returned.

XML Output Example

```
<storage_server_locations type="array">
  <storage_server_location>
    <city>Kolin</city>
    <country>CZ</country>
    <id type="integer">816382921</id>
    <storage_server>
      <id type="integer">670906719</id>
      <status>ACTIVE</status>
      <http_push_on type="boolean">false</http_push_on>
      <vod_push_on type="boolean">true</vod_push_on>
    </storage_server>
  </storage_server_location>
</storage_server_locations>
```

Where:

city - the city where the storage server is located
country - the country where the storage server is located
id - the ID of the location
storage_server - the array of the storage server details
id - the ID of the storage server
status - the status of the storage server
http_push_on
vod_push_on

4.14.32 Get List of Edge Server IP Ranges
To get the list of edge server IP ranges, use the following request:
GET /cdn_resources/edge_ips.xml
GET /cdn_resources/edge_ips.json

XML Request Example

```bash
```

JSON Request Example

```bash
```

XML Output Example

```xml
<edge_ips type="array">
  <edge_ip>66.167.227.0/24</edge_ip>
  <edge_ip>51.9.120.0/24</edge_ip>
</edge_ips>
```

Where:
edge_ips – the array of edge server IP ranges

4.15 CDN SSL Certificates API

OnApp customers can import their own SSL certificates with the Subject Name Indication (SNI) extension. One SSL certificate can be associated with several CDN resources, but a resource can only be linked to one SSL certificate.

However, some of the older browsers do not support SNI. In this case, users who prefer browsers that do not support SNI can purchase an SSL certificate and the SAN solution will be applied. On questions about the SSL certificate purchase, please contact OnApp support.

For the list of browsers that do not support SNI, kindly refer to the Server Name Indication article.

OnApp currently supports the following types of certificates:

- domain-validated (DV) certificate (example.com)
  - single certificate
  - wildcard certificate (*.example.com)
- SAN certificate (any domains)
- organization validation (OV) certificates
  - single certificate
  - wildcard certificate (*.example.com)
  - SAN certificate (any domains)
- extended validation (EV) certificates
  - single certificate
  - wildcard certificate (*.example.com)
  - SAN certificate (any domains)
- high-assurance certificates

- This feature is available for HTTP Pull and HTTP Push resources only.
- To add custom SNI SSL certificates, the user needs to have CDN resources in the cloud and CDN SSL Certificates permissions.
- Custom SNI SSL certificates can be used for secondary hostnames.
- A custom SNI SSL certificate can only be associated with a CDN resource if the certificate and the resource have the same owner. When you link a resource to a certificate you should only specify the IDs of those certificates that were added by the user with whom the new resource will be associated. If you indicate some other certificate's ID an error will occur.
- When a custom SNI SSL certificate is associated with a CDN resource, the certificate applies only to the edge servers subscribed to that resource.

### 4.15.1 Get List of Custom SNI SSL Certificates

To view the list of available custom SNI SSL certificates, use the following request:

GET /cdn_ssl_certificates.xml
GET /cdn_ssl_certificates.json

**XML Request Example**

```
```

**JSON Request Example**

```
```
**XML Output Example**

```xml
<cdn_ssl_certificates type="array">
  <cdn_ssl_certificate>
    <created_at type="datetime">2015-03-25T14:26:00+00:00</created_at>
    <id type="integer">1</id>
    <name>cert3</name>
    <updated_at type="datetime">2015-03-25T14:26:00</updated_at>
    <user_id type="integer">3</user_id>
    <cdn_reference type="integer">61654962</cdn_reference>
  </cdn_ssl_certificate>
  ...<cdn_ssl_certificate>
</cdn_ssl_certificates>
```

Where:
- `created_at` - the time in the `[YYYY][MM][DD][hh][mm][ss]Z` format
- `id` - ID of the certificate
- `name` - name of the certificate
- `updated_at` - the time in the `[YYYY][MM][DD][hh][mm][ss]Z` format
- `user_id` - the ID of the user who added the certificate
- `cdn_reference` - the ID of the certificate on the remote Aflexi server

### 4.15.2 Get Custom SNI SSL Certificate Details

To view the details of a custom SNI SSL certificate, use the following request:

**GET** /cdn_ssl_certificates/:id.xml

**GET** /cdn_ssl_certificates/:id.json

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

Where:
- `id` - ID of the certificate

**XML Output Example**
<cdn_ssl_certificate>
  <created_at type="datetime">2015-04-08T03:52:00-10:00</created_at>
  <id type="integer">35</id>
  <name></name>
  <updated_at type="datetime">2015-04-08T03:52:00-10:00</updated_at>
  <user_id type="integer">30</user_id>
  <cdn_resources type="array">
    <cdn_resource>
      <cdn_hostname>res1.test.com</cdn_hostname>
      <cdn_ssl_certificate_id type="integer">35</cdn_ssl_certificate_id>
      <cname>535478274.r.worldcdn-beta.net</cname>
      <created_at type="datetime">2014-11-25T23:54:10:00</created_at>
      <id type="integer">7401</id>
      <resource_type>HTTP_PULL</resource_type>
      <updated_at type="datetime">2015-04-09T03:29:10:00</updated_at>
      <user_id type="integer">30</user_id>
      <last_24h_cost type="float">0.0</last_24h_cost>
      <cdn_reference type="integer">535478274</cdn_reference>
      <origins type="array">
        <origin>1.1.1.1</origin>
      </origins>
    </cdn_resource>
  </cdn_resources>
  <cdn_reference type="integer">729656986</cdn_reference>
</cdn_ssl_certificate>

Where:

*created_at* - the time in the [YYYY][MM][DD][T][hh][mm][ss]Z format

*id* - ID of the certificate

*name* - name of the certificate

*updated_at* - the time in the [YYYY][MM][DD][T][hh][mm][ss]Z format

*user_id* - ID of the user who added the certificate

*cdn_resources* - array of parameters related to the CDN resources associated with the certificate:

*cdn_hostname* - the hostname that will serve static content

*cdn_ssl_certificate_id* - the ID of the custom SNI SSL certificate associated with the resource

*cname* - CNAME record

*created_at* - the date when the CDN resource was created in the [YYYY][MM][DD][T][hh][mm][ss]Z format

*id* - the resource ID in the database

*resource_type* - CDN resource type

*updated_at* - the date when the CDN resource was updated in the [YYYY][MM][DD][T][hh][mm][ss]Z format

*user_id* - the ID of the user, who owns the resource

*last_24h_cost* - the amount due for the last 24 hours

*cdn_reference* - ID of the SSL certificate on the remote Aflexi server

*origin* - the path from which the CDN requests the content

*cdn_reference* - ID of the certificate on the remote Aflexi server
4.15.3 Add Custom SNI SSL Certificate

To add a custom SNI SSL Certificate, use the following request:

POST /cdn_ssl_certificates.xml

To add custom SNI SSL certificates, the user needs to have CDN SSL Certificates in their OnApp account.

XML Request Example

curl -i -X POST -u user:userpass -H 'Accept: application/xml' -H 'Content-Type: application/xml' -d "<?xml version="1.0" encoding="UTF-8"?>
<xmlRequest>
<!-- Add Custom SNI SSL Certificate -->
<addSNISSLCertificate>
  <name>cert1</name>
  <cert>
    <![CDATA[http://onapp.test/cdn_ssl_certificates.xml]]>
    <!-- Add the certificate's URL here -->
  </cert>
</addSNISSLCertificate>
</xmlRequest>"
JSON Request Example

```json
"cdn_ssl_certificate": {
"name": "cert1",
"cert": "http://onapp.test/cdn_ssl_certificates.json"
}
```

OnApp Cloud 6.5 Edge 5 CDN Guide
4.15.4 Edit SNI SSL Certificate

To update a custom SNI SSL certificate, use the following request:

```
PUT /cdn_ssl_certificates/:id.xml
```

**XML Request Example**

```
```

```xml
<cdn_ssl_certificate>
  <name>cert1</name>
  <cert>
    -----BEGIN CERTIFICATE-----
    MIIFKzCCBBOgAwIBAgIQMvEFlcrw7X8kOaJ/Sy1eYjANBgkqhkiG9w0BAQUFADB/
    -----END CERTIFICATE-----
    
    -----BEGIN RSA PRIVATE KEY-----
    ----
    
</cdn_ssl_certificate>
```

**Where:**

- **name** - the name of the certificate
- **cert** - the certificate key
- **key** - the SSL key issued by your SSL provider

- Every line must be separated by \n.
- Make sure there are no spaces between lines of **cert** and **key**. Except delimiters:
  ```
  -----BEGIN CERTIFICATE-----
  -----END CERTIFICATE-----
  -----BEGIN RSA PRIVATE KEY-----
  -----END RSA PRIVATE KEY-----
  ```

4.15.4 Edit SNI SSL Certificate

To update a custom SNI SSL certificate, use the following request:

```
PUT /cdn_ssl_certificates/:id.xml
```

**XML Request Example**

```
```

```xml
<cdn_ssl_certificate>
  <name>cert1</name>
  <cert>
    -----BEGIN CERTIFICATE-----
    MIIFKzCCBBOgAwIBAgIQMvEFlcrw7X8kOaJ/Sy1eYjANBgkqhkiG9w0BAQUFADB/
    -----END CERTIFICATE-----
    
    -----BEGIN RSA PRIVATE KEY-----
    ----
    
</cdn_ssl_certificate>
```

**Where:**

- **name** - the name of the certificate
- **cert** - the certificate key
- **key** - the SSL key issued by your SSL provider

- Every line must be separated by \n.
- Make sure there are no spaces between lines of **cert** and **key**. Except delimiters:
  ```
  -----BEGIN CERTIFICATE-----
  -----END CERTIFICATE-----
  -----BEGIN RSA PRIVATE KEY-----
  -----END RSA PRIVATE KEY-----
  ```
Where:

`id` - ID of the certificate

`name` - the name of the certificate

`cert` - the certificate key

`key` - the SSL key issued by your SSL provider

- Every line should be separated by \n.
- Make sure that `cert` and `key` parameters do not contain spaces between lines. Except delimiters:

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
```

### 4.15.5 Delete SNI SSL Certificate

To delete a custom SNI SSL certificate, use the following request:

DELETE /cdn_ssl_certificates/:id.xml

DELETE /cdn_ssl_certificates/:id.json
XML Request Example

```
```

JSON Request Example

```
```

Where:

- id - the ID of the certificate

4.15.6 Add Custom SNI SSL Certificate to CDN Resource

To add a custom SNI SSL certificate to a CDN HTTP Pull or Push resource, use the following request:

PUT /cdn_resources/:id.xml
PUT /cdn_resources/:id.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where:

- `cdn_ssl_certificate_id` - the ID of the custom SNI SSL certificate you want to add to the resource

4.15.7 Search Custom SNI SSL Certificates by Name

To search a custom SNI SSL certificate by name, use the following request:

GET /cdn_ssl_certificates.xml
GET /cdn_ssl_certificates.json

**XML Request Example**

```
```

---

OnApp Cloud 6.5 Edge 5 CDN Guide
JSON Request Example

```
```

Where:

- `q` - search pattern that can be the full or incomplete certificate's name. In the output, you will get all results that match the search pattern.

4.16 CDN Storage Servers API

A CDN storage server is a server used for storing the content to be distributed over CDN. When the content is requested on the CDN, it is served by the edge server nearest to the customer’s geographical location. There are two types of CDN storage servers in OnApp: HTTP and Streaming.

4.16.1 Get List of CDN Storage Servers

To get the list of all CDN storage servers in the cloud, use the following request:

GET /storage_servers.xml
GET /storage_servers.json

**XML Request Example**

```
curl -i -X GET -u user:userpass http://onapp.test/storage_servers.xml
```

**JSON Request Example**

```
curl -i -X GET -u user:userpass http://onapp.test/storage_servers.json
```

**XML Output Example**
<storage_servers type="array">
<storage_server>
<admin_note nil="true"/>
<allow_resize_without_reboot type="boolean">true</allow_resize_without_reboot>
<allow_hot_migrate type="boolean">true</allow_hot_migrate>
<allow_swap type="boolean">true</allow_swap>
<booted type="boolean">true</booted>
<built type="boolean">true</built>
<cpu_shares type="integer">1</cpu_shares>
<cpus type="integer">1</cpus>
<created_at type="datetime">2012-08-07T05:41:02-10:00</created_at>
<customer_network_id nil="true"/>
<edge_server_type nil="true"/>
<enable_autoscale nil="true"/>
<enable_monitis type="boolean">false</enable_monitis>
<firewall_notrack type="boolean">false</firewall_notrack>
<hypervisor_id type="integer">12</hypervisor_id>
<id type="integer">4891</id>
<identifier>dgsfyze4lu9m6q</identifier>
<initial_root_password>1is9qsjtsjmf</initial_root_password>
<initial_root_password_encrypted type="boolean">false</initial_root_password_encrypted>
<label>storage3</label>
<local_remote_access_port type="integer">5903</local_remote_access_port>
<locked type="boolean">true</locked>
<memory type="integer">512</memory>
<min_disk_size type="integer">5</min_disk_size>
<note nil="true"/>
<operating_system>linux</operating_system>
<operating_system_distro>ubuntu</operating_system_distro>
<recovery_mode type="boolean">false</recovery_mode>
<remote_access_password nil="true"/>
<state>new</state>
<storage_server_type>streaming</storage_server_type>
<strict_virtual_machine_id nil="true"/>
<suspended type="boolean">false</suspended>
<template_id type="integer">8</template_id>
<template_label>OnApp CDN hypervisor</template_label>
<updated_at type="datetime">2012-08-07T06:08:07-10:00</updated_at>
<user_id type="integer">1</user_id>
<vip nil="true"/>
<xen_id type="integer">69</xen_id>
<ip_addresses type="array">
<ip_address>
<address>109.123.105.189</address>
<broadcast>109.123.105.191</broadcast>
<created_at type="datetime">2012-07-13T00:58:21-10:00</created_at>
<customer_network_id nil="true"/>
<disallowed_primary type="boolean">false</disallowed_primary>
<gateway>109.123.105.177</gateway>
<id type="integer">12</id>
<ip_address_pool_id nil="true"/>
<network_address>109.123.105.176</network_address>
<netmask>255.255.255.240</netmask>
</ip_address>
</ip_addresses>
<total_disk_size type="integer">5</total_disk_size>
<cpu_priority type="integer">1</cpu_priority>
<edge_status>Active</edge_status>
</storage_server>
Where:

- **add_to_marketplace** – true if this storage server is added to the marketplace; otherwise false
- **admin_note** – an optional reminder for this server created by an administrator
- **allow_resize_without_reboot** – true if adjusting resource allocation without reboot is possible; otherwise false
- **allowed_hot_migrate** – true if hot migration is allowed; otherwise false
- **allowed_swap** – true if swap is allowed; otherwise false
- **booted** – true if the server is booted; otherwise false
- **built** – true if the server is already built; otherwise false
- **cpu_shares** – the CPU priority percentage
- **cpus** – number of CPU cores allocated to this storage server
- **created_at** – the date when the CDN storage server was created in the [YYYY][MM][DD][hh][mm][ss]Z format
- **edge_server_type** - type of the CDN storage server
- **enable_autoscale** – false; not available for edge servers
- **enable_monitis** - deprecated attribute; will be removed in upcoming release
- **hypervisor_id** – the ID of the compute resource, on which the server is deployed
- **id** – the edge server ID in OnApp CP database
- **identifier** – the edge server identifier
- **initial_root_password** – the server root password
- **initial_root_password_encrypted** – true, if the server root password is encrypted, otherwise false
- **label** – an arbitrary name of the edge server
- **local_remote_access_port** – the port ID used for console access
- **locked** – true if locked; otherwise false
- **memory** – the amount of RAM resources allocated to this edge server
- **min_disk_size** – minimum disk space required by the template
- **note** - an optional reminder for this VS made by a user account
- **operating_system** – type of operating system
- **operating_system_distro** – the distribution of the operating system
- **recovery_mode** – true if the server is booted in the recovery mode; otherwise false
- **remote_access_password** – the password for remote access
- **state** – deprecated attribute; will be removed in upcoming release
- **strict_virtual_machine_id** - the ID of a virtual server (or edge server) that will never reside on the same compute resource with this server
- **suspended** – true if suspended; otherwise false
- **template_id** – the ID of the template, on which the edge server is based
- **template_label** – label of the template on which the server is based; currently – OnApp CDN compute resource
- **updated_at** – the date when the CDN edge server was updated in the [YYYY][MM][DD][hh][mm][ss]Z format
user_id – the ID of the user, who is the server owner

vip – true if the server has VIP status for migration; otherwise false

xen_id – the edge server ID set by the virtualization engine

ip_addresses – an array of assigned IP addresses with the following parameters:

- address – IP address
- broadcast – broadcast address
- created_at — the date in the [YYYY][MM][DD][hh][mm][ss]Z format
- customer_network_id -
- disallowed_primary – true if not allowed to be used as primary (for VS build), otherwise false
- gateway – gateway address
- id – the ID of the IP address
- ip_address_pool_id - ID of the IP address pool
- network_address – the address of the network
- network_id – the ID of the network
- updated_at – the date when the Network was updated in the [YYYY][MM][DD][hh][mm][ss]Z format
- free – true if free, otherwise false
- netmask – netmask for the IP address

total_disk_size – total disk space in GB of primary and swap disks

edge_status - status of the storage server

cpu_priority - this is a new parameter reserved for further use; currently will have the same value as cpu_shares

4.16.2 Get List of HTTP Storage Servers

To get the list of HTTP storage servers, use the following request:

GET /storage_servers.xml?type=http
GET /storage_servers.json?type=http

XML Request Example

curl -i -X GET -u user:userpass
http://onapp.test/storage_servers.xml?type=http

JSON Request Example

curl -i -X GET -u user:userpass
http://onapp.test/storage_servers.json?type=http

For output example and parameters description, refer to Get List of CDN Storage Servers section.
4.16.3 Get List of Streaming Storage Servers

To get the list of streaming edge servers, use the following request:

GET /storage_servers.xml?type=streaming
GET /storage_servers.json?type=streaming

**XML Request Example**

```
curl -i -X GET -u user:userpass
http://onapp.test/storage_servers.xml?type=streaming
```

**JSON Request Example**

```
curl -i -X GET -u user:userpass
http://onapp.test/storage_servers.json?type=streaming
```

For output example and parameters description, refer to [Get List of CDN Storage Servers](#) section.

4.16.4 Get CDN Storage Server Details

To view details of a particular storage server, use the following request:

GET /storage_servers/:id.xml
GET /storage_servers/:id.json

**XML Request Example**

```
curl -i GET -u user:userpass --url
http://onapp.test/storage_servers/12.xml
```

**JSON Request Example**

```
curl -i GET -u user:userpass --url
http://onapp.test/storage_servers/12.json
```

Where:

id - ID of the CDN storage server

**XML Output Example**
<storage_server>
<add_to_marketplace type="boolean">false</add_to_marketplace>
<admin_note nil="true"/>
<allow_resize_without_reboot type="boolean">true</allow_resize_without_reboot>
<allowed_hot_migrate type="boolean">true</allowed_hot_migrate>
<allowed_swap type="boolean">true</allowed_swap>
<booted type="boolean">true</booted>
<built type="boolean">true</built>
<cpu_shares type="integer">1</cpu_shares>
<cpus type="integer">1</cpus>
<created_at type="datetime">2012-08-07T05:41:02-10:00</created_at>
<customer_network_id nil="true"/>
<edge_server_type nil="true"/>
<enable_autoscale nil="true"/>
<enable_monitis type="boolean">false</enable_monitis>
<firewall_notrack type="boolean">false</firewall_notrack>
<hypervisor_id type="integer">12</hypervisor_id>
<id type="integer">4891</id>
<identifier>dgsfyze4lu9m6q</identifier>
<initial_root_password>1is9qsjtsjmf</initial_root_password>
<initial_root_password_encrypted type="boolean">false</initial_root_password_encrypted>
<label>storage3</label>
<local_remote_access_port type="integer">5903</local_remote_access_port>
<locked type="boolean">true</locked>
<min_disk_size type="integer">5</min_disk_size>
<note nil="true"/>
<operating_system>linux</operating_system>
<operating_system_distro>ubuntu</operating_system_distro>
<recovery_mode type="boolean">false</recovery_mode>
<remote_access_password nil="true"/>
<state>new</state>
<storage_server_type>streaming</storage_server_type>
<suspended type="boolean">false</suspended>
<template_id type="integer">8</template_id>
<template_label>OnApp CDN hypervisor</template_label>
<updated_at type="datetime">2012-08-08T23:37:20-10:00</updated_at>
<user_id type="integer">1</user_id>
<vip nil="true"/>
<xen_id type="integer">69</xen_id>
<ip_addresses type="array">
<ip_address>
<address>109.123.105.189</address>
<broadcast>109.123.105.191</broadcast>
<created_at type="datetime">2012-04-03T06:08:07-10:00</created_at>
<customer_network_id nil="true"/>
<disallowed_primary type="boolean">false</disallowed_primary>
<gateway>109.123.105.177</gateway>
<hypervisor_id nil="true"/>
<id type="integer">12</id>
<ip_address_pool_id nil="true"/>
<network_address type="integer">109.123.105.176</network_address>
<network_id type="integer">1</network_id>
<pxe type="boolean">false</pxe>
<updated_at type="datetime">2012-07-13T00:58:21-10:00</updated_at>
<user_id nil="true"/>
<free type="boolean">false</free>
<netmask>255.255.255.240</netmask>
</ip_address>
</ip_addresses>
<total_disk_size type="integer">5</total_disk_size>
<edge_status>Active</edge_status>
</storage_server>
For parameters description, refer to Get List of CDN Storage Servers section.

### 4.16.5 Add CDN Storage Server

To create a new CDN storage server, use the following request:

**POST /storage_servers.xml**

**POST /storage_servers.json**

**XML Request Example**

```curl
curl -i -X POST -d
'<![CDATA[
<storage_server>
  <label>az_CDN_test</label>
  <cpus>1</cpus>
  <data_store_group_primary_id>2</data_store_group_primary_id>
  <primary_network_group_id>3</primary_network_group_id>
  <cpu_shares>1</cpu_shares>
  <memory>512</memory>
  <required_virtual_machine_build>1</required_virtual_machine_build>
  <hypervisor_group_id>1</hypervisor_group_id>
  <hypervisor_id>1</hypervisor_id>
  <required_ip_address_assignment>1</required_ip_address_assignment>
  <primary_disk_size>5</primary_disk_size>
  <rate_limit>0</rate_limit>
  <cdn_location_id>5</cdn_location_id>
  <storage_server_type>http</storage_server_type>
  <location_id>2</location_id>
</storage_server>]]>
' -u user:userpass
http://onapp.test/storage_servers.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**JSON Request Example**

```curl
curl -i -X POST -d '
"storage_server":{
  "label":"az_CDN_test",
  "cpus":"1",
  "data_store_group_primary_id":"2",
  "primary_network_group_id":"3",
  "cpu_shares":"1",
  "memory":"512",
  "required_virtual_machine_build":"1",
  "hypervisor_group_id":"1",
  "hypervisor_id":"1",
  "required_ip_address_assignment":"1",
  "primary_disk_size":"5",
  "rate_limit":"0",
  "cdn_location_id":"5",
  "storage_server_type":"http",
  "location_id":"2"}
' -u user:userpass
http://onapp.test/storage_servers.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```

**Where:**  
- **label** – a unique name of your CDN storage server. The label can consist of letters [A-Za-z], digits [0-9], dash [-], lower dash [ _ ], space character [ ], at sign [@], round brackets [[]], slashes [ / ], comma [ , ] and dot [.]. You can use both lower- and uppercase letters. The label should begin with an alphanumeric character or lower dash [ _ ]
- **hypervisor_id** - indicate the ID of the compute resource, on which the server will be deployed
- **hypervisor_group_id** - indicate the compute zone ID
- **cpus** * - the amount of CPU cores allocated to this storage server
- **cpu_shares** * - the percentage of allocated CPU priority resource
- **memory** * - the amount of RAM, which you want to allocate to this storage server
- **primary_disk_size** * - the size in GB of the primary disk
- **data_store_group_primary_id** – specify the ID of a data store zone, where you want to locate the disk of your server. If not specified – the system will select the data store zone with higher available capacity
- **primary_network_group_id** – indicate the network zone ID
- **network_id** - the ID of the network to which the edge server will be connected
**required_virtual_machine_build** – set "1" to build the server automatically after creation. Otherwise set "0"

**required_ip_address_assignment** - set "1" if you want IP address to be assigned automatically after creation. Otherwise set "0"

**selected_ip_address** - an IP address to assign to this container server; if the parameter **required_ip_address_assignment** was set "1" but this parameter **selected_ip_address** is empty - the first available IP address will be assigned to container server automatically

**cdn_location_id** - the ID of the CDN location. Use the following API call to find the ID [Get List of CDN Locations for Location Group](#). The parameter is optional. If not set, the edge server will be assigned to the first CDN Location in its Location Group.

**storage_server_type** - set http or streaming server type.

**location_id** - the ID of the location group

---

**ATTENTION!** Creating a Streaming Edge or Storage server will result in an additional monthly charge. You will be charged 50$ per month for deploying this Streaming server once it is provisioned.

---

### Page History

**v.6.0**
- added the **location_id** parameter
- replaced **cdn_location** with **cdn_location_id** parameter

**v.5.4**
- added the following parameters:
  - **selected_ip_address**
  - **network_id**

**v. 3.3.1**
- added the following parameters:
  - **cdn_location**
  - **storage_server_type**

### 4.16.6 Edit Storage Servers

To edit a CDN storage server, use the following request:

PUT /storage_servers/:id.xml
PUT /storage_servers/:id.json

**XML Request Example**

```
curl -l -X PUT -d
  '<storage_server><label>az_CDN_test_1</label><add_to_marketplace>true</add_to_marketplace><cpus>1</cpus><cpu_shares>10</cpu_shares><memory>512</memory><storage_server>' -u onapp.test
  'Content-type: application/xml'
```
**4.16.7 Reboot CDN Storage Server**

To reboot the storage server, use the following request:

**POST** /storage_servers/:storage_server_id/reboot.xml

**POST** /storage_servers/:storage_server_id/reboot.json

**XML Request Example**

```bash
curl -i -X POST -u user:userpass --url http://onapp.test/storage_servers/12/reboot.xml
```

**JSON Request Example**

```bash
curl -i -X POST -u user:userpass --url http://onapp.test/storage_servers/12/reboot.json
```

**4.16.8 Start up CDN Storage Server**

To start up a CDN storage server, use the following request:

**POST** /storage_servers/:storage_server_id/startup.xml

**POST** /storage_servers/:storage_server_id/startup.json

**XML Request Example**

```bash
curl -i -X POST -u user:userpass --url http://onapp.test/storage_servers/12/startup.xml
```

**JSON Request Example**
4.16.9 Shut down CDN Storage Server

To shut down a CDN storage server, use the following request:

POST /storage_servers/:storage_server_id/shutdown.xml
POST /storage_servers/:storage_server_id/shutdown.json

**XML Request Example**

```
curl -i -X POST -u user:userpass --url
http://onapp.test/storage_servers/12/shutdown.xml
```

**JSON Request Example**

```
curl -i -X POST -u user:userpass --url
http://onapp.test/storage_servers/12/shutdown.json
```

4.16.10 Stop CDN Storage Server

To terminate the storage server forcefully, use the following request:

POST /storage_servers/:storage_server_id/stop.xml
POST /storage_servers/:storage_server_id/stop.json

**XML Request Example**

```
curl -i -X POST -u user:userpass --url
http://onapp.test/storage_servers/12/stop.xml
```

**JSON Request Example**

```
curl -i -X POST -u user:userpass --url
http://onapp.test/storage_servers/12/stop.json
```

4.16.11 Rebuild CDN Storage Server

To rebuild (or build manually) the CDN storage server, use the following request:

POST /storage_servers/:storage_server_id/build.xml
POST /storage_servers/:storage_server_id/build.json

**XML Request Example**
JSON Request Example

curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"storage_server":{"template_id":"8","required_startup":"1"}}' --url http://onapp.test/storage_servers/12/build.json

Where:

*template_id* - the ID of the template on which this server will be based

*required_startup* – set "1" to start up the server automatically after build, otherwise set "0"

4.16.12 Suspend CDN Storage Server

To suspend a CDN storage server, use the following request:

POST /storage_servers/:storage_server_id/suspend.xml
POST /storage_servers/:storage_server_id/suspend.json

XML Request Example

curl -i -X POST -u user:userpass --url http://onapp.test/storage_servers/12/suspend.xml

JSON Request Example

curl -i -X POST -u user:userpass --url http://onapp.test/storage_servers/12/suspend.json

To unsuspend a storage server, run the request again.

4.16.13 Unlock CDN Storage Server

To unlock CDN storage server, use the following request:

POST /storage_servers/:storage_server_id/unlock.xml
POST /storage_servers/:storage_server_id/unlock.json

XML Request Example

curl -i -X POST -u user:userpass --url http://onapp.test/storage_servers/12/unlock.xml

JSON Request Example
4.16.14 Set VIP Status for CDN Storage Server

To give your storage server a migration priority, set the VIP status for it with the following request:

POST /storage_servers/:storage_server_id/set_vip.xml
POST /storage_servers/:storage_server_id/set_vip.json

XML Request Example


JSON Request Example


Where:

vip - whether VIP status is enabled for the server or not. Set this parameter to 'true' to enable and to 'false' to disable the VIP status.

4.16.15 Delete CDN Storage Servers

To delete a CDN storage server, use the following request:

DELETE /storage_servers/:id.xml
DELETE /storage_servers/:id.json

XML Request Example

curl -i -X DELETE -u user:userpass

JSON Request Example

curl -i -X DELETE -u user:userpass

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no storage server with a requested ID, or URL is incorrect.
### 4.16.16 Migrate CDN Storage Server

To migrate a storage server, use the following request:

- POST `/storage_servers/:storage_server_id/migrate.xml`
- POST `/storage_servers/:storage_server_id/migrate.json`

**XML Request Example**

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d
  '<storage_server><destination>13</destination><cold_migrate_on_rollback>1</cold_migrate_on_rollback><storage_server>'
  --url http://onapp.test/storage_servers/12/migrate.xml
```

**JSON Request Example**

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d
  '{"storage_server":{"destination":"18","cold_migrate_on_rollback":"1"}}'
  --url http://onapp.test/storage_servers/12/migrate.json
```

**Where:**
- `destination`* - the ID of a target compute resource where you migrate a storage server
- `cold_migrate_on_rollback` - set to 1 if you wish to switch to a cold migration if hot migration fails, otherwise set 0

### 4.16.17 Segregate CDN Storage Server

To segregate an edge server (that is, instruct it never to reside on the same compute resource as another VS or edge server), use the following request:

- POST `/storage_servers/:storage_server_id/strict_vm.xml`
- POST `/storage_servers/:storage_server_id/strict_vm.json`

**XML Request Example**

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d
  '<virtual_machine><strict_virtual_machine_id>l615jn0d9tlv0s</strict_storage_server_id></virtual_machine>'
  --url http://onapp.test/storage_servers/12/strict_vm.xml
```

**JSON Request Example**

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d
  '{"virtual_machine":{"strict_virtual_machine_id":"l615jn0d9tlv0s"}}'
  --url http://onapp.test/storage_servers/12/strict_vm.json
```

**Where:**
- `strict_virtual_machine_id`* - the ID of the server you wish to segregate
4.16.18  Resize CDN Storage Server

To resize a CDN storage server, use the following request:

POST /storage_servers/:storage_server_id/resize.xml
POST /storage_servers/:storage_server_id/resize.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where:

- **memory** - the amount of RAM, which you want to allocate to this storage server
- **cpus** - the amount of CPU cores allocated to this storage server

4.16.19  Change CDN Storage Server Owner

To change the owner of the CDN storage server, use the following request:

POST /storage_servers/:storage_server_id/change_owner.xml
POST /storage_servers/:storage_server_id/change_owner.json

**XML Request Example**

```
```

**JSON Request Example**

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d {'user_id":"1"}' --url http://onapp.test/storage_servers/12/change_owner.json
```

4.16.20  Rebuild Network for CDN Storage Server

It is required to rebuild network after any changes on IP address joins or network interfaces. To rebuild a network, use the following request:

POST /storage_servers/:storage_server_id/rebuild_network.xml
POST /storage_servers/:storage_server_id/rebuild_network.json

**XML Request Example**

```
4.16.21 CDN Storage Server Disks

You can perform all the same actions with storage servers' disks as with VS disks, except POST and DELETE. The only difference would be in the routes for the following requests:

To view the storage server disks, use the following request:

GET /storage_servers/:storage_server_id/disks.xml
GET /storage_servers/:storage_server_id/disks.json

Parameters description and output example.
For other possible requests, refer to corresponding sections of Disks chapter.

4.16.22 CDN Storage Server Network Interfaces

This section contains the API requests you can use to manage storage servers' network interfaces:

- Get List of CDN Storage Server Network Interfaces
- Get CDN Storage Server Network Interface Details
- Add CDN Storage Server Network Interface
- Edit CDN Storage Server Network Interface
- Delete CDN Storage Server Network Interface

4.16.22.1 Get List of CDN Storage Server Network Interfaces

To get the list of network interfaces allocated to the storage server, use the following request:

GET /storage_servers/:storage_server_id/network_interfaces.xml
GET /storage_servers/:storage_server_id/network_interfaces.json

XML Request Example

JSON Request Example


XML Output Example

```
<network_interfaces type="array">
  <network_interface>
    <label>eth0</label>
    <usage nil="true"></usage>
    <created_at type="datetime">2011-03-18T17:45:07+07:00</created_at>
    <updated_at type="datetime">2011-04-08T18:57:20+07:00</updated_at>
    <primary type="boolean">true</primary>
    <usage_month_rolled_at nil="true"></usage_month_rolled_at>
    <id type="integer">502</id>
    <mac_address>00:16:3e:50:35:52</mac_address>
    <usage_last_reset_at nil="true"></usage_last_reset_at>
    <default_firewall_rule>DROP</default_firewall_rule>
    <rate_limit type="integer">0</rate_limit>
    <storage_server_id type="integer">518</storage_server_id>
    <network_join_id type="integer">4</network_join_id>
    <identifier>pdfjrpdkday9e1</identifier>
  </network_interface>
  ...
</network_interfaces>
```

Where:

- **label** - network interface name
- **created_at** - the timestamp in the database when this network interface was created
- **updated_at** - the timestamp in the database when this network interface was updated
- **primary** - *true* if this network interface is primary, otherwise *false*
- **id** - the ID of this network interface
- **mac_address** - network interface MAC address
- **rate_limit** - port speed in Mbps
- **identifier** - the identifier in the database of this network interface
- **network_join_id** - the ID of the network join to which this network interface belongs
- **storage_server_id** - the ID of a storage server to which this network interface is attached

4.16.22.2 Get CDN Storage Server Network Interface Details

To get a particular network interface details, use the following request:

GET /storage_servers/:storage_server_id/network_interfaces/:id.xml
GET /storage_servers/:storage_server_id/network_interfaces/:id.json
XML Request Example

```bash
```

JSON Request Example

```bash
```

XML Output Example

```xml
<network_interface>
    <label>eth0</label>
    <usage nil="true"></usage>
    <created_at type="datetime">2011-03-18T17:45:07+07:00</created_at>
    <updated_at type="datetime">2011-04-08T18:57:20+07:00</updated_at>
    <primary type="boolean">true</primary>
    <usage_month_rolled_at nil="true"></usage_month_rolled_at>
    <id type="integer">502</id>
    <mac_address>00:16:3e:50:35:52</mac_address>
    <usage_last_reset_at nil="true"></usage_last_reset_at>
    <default_firewall_rule>DROP</default_firewall_rule>
    <rate_limit type="integer">0</rate_limit>
    <storage_server_id type="integer">13</storage_server_id>
    <network_join_id type="integer">4</network_join_id>
    <identifier>pdfjrtpkday9e1</identifier>
</network_interface>
```

Where:
- `label` - network interface name
- `created_at` - the timestamp in the database when this network interface was created
- `updated_at` - the timestamp in the database when this network interface was updated
- `primary` - `true` if this network interface is primary, otherwise `false`
- `id` - the ID of this network interface
- `mac_address` – network interface MAC address
- `rate_limit` - port speed in Mbps
- `identifier` - the identifier in the database of this network interface
- `network_join_id` - the ID of the network join to which this network interface belongs
- `storage_server_id` - the ID of a storage server to which this network interface is attached

4.16.22.3 Add CDN Storage Server Network Interface

To add a new network interface, use the following request:

```bash
POST /storage_servers/:storage_server_id/network_interfaces.xml
POST /storage_servers/:storage_server_id/network_interfaces.json
```

XML Request Example
4.16.22.4 Edit CDN Storage Server Network Interface

To edit a network interface, use the following request:

**PUT /storage_servers/:storage_server_id/network_interfaces/:id.xml**

**PUT /storage_servers/:storage_server_id/network_interfaces/:id.json**

**XML Request Example**

```
```

**JSON Request Example**

```
curl -i -X PUT http://onapp.test/storage_servers/9/network_interfaces/12.json -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"network_interface":{"label":"eth0(test" rate_limit":"32","primary":false}}
```

Where:
- **label** - the label of a network interface
- **rate_limit** - the port speed of a network interface

After adding the network interface, the virtual server should be power cycled for the change to take effect.

4.16.22.5 Edit CDN Storage Server Network Interface

To edit a network interface, use the following request:

**PUT /storage_servers/:storage_server_id/network_interfaces/:id.xml**

**PUT /storage_servers/:storage_server_id/network_interfaces/:id.json**

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where:
- **label** - give the label of a network interface you wish to attach
- **rate_limit** - set the port speed of a network interface you wish to attach
- **network_join_id** - set the ID of a physical network used to attach this network interface
- **primary** - set 1 if the interface is primary. Otherwise false.
4.16.22.5 Delete CDN Storage Server Network Interface

To delete a network interface from the storage server, use the following request:

DELETE /storage_servers/:storage_server_id/network_interfaces/:id.xml
DELETE /storage_servers/:storage_server_id/network_interfaces/:id.json

**XML Request Example**

```bash
curl -i -X DELETE -u user:userpass
http://onapp.test/storage_servers/9/networkInterfaces/12.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**JSON Request Example**

```bash
curl -i -X DELETE -u user:userpass
http://onapp.test/storage_servers/9/networkInterfaces/12.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```

4.16.23 CDN Storage Server IP Address Joins

An IP address allocated to a storage server is an IP address join. Use the following methods to manage IP address joins of your CDN storage servers.

To get the list of IP address assignments for a particular storage server, use the following request:

GET /storage_servers/:storage_server_id/ip_addresses.xml
GET /storage_servers/:storage_server_id/ip_addresses.json

**XML Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/storage_servers/1/ip_addresses.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**JSON Request Example**

```bash
curl -i -X GET -u user:userpass
http://onapp.test/storage_servers/1/ip_addresses.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```

To assign an IP Address to a storage server, use the following request:

POST /storage_servers/:storage_server_id/ip_addresses.xml
POST /storage_servers/:storage_server_id/ip_addresses.json

**XML Request Example**

```bash
curl -i -X POST -u user:userpass
http://onapp.test/storage_servers/1/ip_addresses.xml
-H 'Accept: application/xml'
-H 'Content-type: application/xml'
```

**JSON Request Example**

```bash
curl -i -X POST -u user:userpass
http://onapp.test/storage_servers/1/ip_addresses.json
-H 'Accept: application/json'
-H 'Content-type: application/json'
```
```bash
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d '
  <ip_address_join>  
  <ip_address_id>7</ip_address_id>  
  <network_interface_id>113</network_interface_id>  
  </ip_address_join>' --url http://onapp.test/storage_servers/1/ip_addresses.xml
```

**JSON Request Example**

```bash
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{
  "ip_address_join": {
    "ip_address_id": "7",
    "network_interface_id": "113"
  }
}' --url http://onapp.test/storage_servers/1/ip_addresses.json
```

**XML Output Example**

```xml
<ip_address_join>
  <created_at type="datetime">2013-10-31T13:41:31+03:00</created_at>
  <id type="integer">174</id>
  <ip_address_id type="integer">7</ip_address_id>
  <network_interface_id type="integer">113</network_interface_id>
  <updated_at type="datetime">2013-10-31T13:41:31+03:00</updated_at>
  <ip_address>
    <address>1.1.3</address>
    <broadcast>1.1.1.255</broadcast>
    <created_at type="datetime">2013-08-07T13:29:09+03:00</created_at>
    <customer_network_id nil="true"/>
    <disallowed_primary type="boolean">false</disallowed_primary>
    <gateway>1.1.1.1</gateway>
    <hypervisor_id nil="true"/>
    <id type="integer">7</id>
    <ip_address_pool_id nil="true"/>
    <network_address>1.1.0</network_address>
    <network_id type="integer">1</network_id>
    <pxe type="boolean">false</pxe>
    <updated_at type="datetime">2013-08-07T13:29:09+03:00</updated_at>
    <user_id nil="true"/>
    <free type="boolean">false</free>
    <netmask>255.255.255.0</netmask>
  </ip_address>
</ip_address_join>
```

**Where:**

- `created_at` - the date when the record was created in DB
- `id` - the IP address join ID
- `ip_address_id` - the IP address ID
- `network_interface_id` - the network interface ID
- `updated_at` - the date when the record was updated in DB
- `ip_address` - the array of IP address details
address - the IP address

broadcast - a logical address at which all devices connected to a multiple-access communications network are enabled to receive datagrams.

customer_network_id - the ID of the customer network
disallowed_primary - true if this address is not set as primary (for VS build), otherwise false
gateway - gateway address

hypervisor_id - the ID of the compute resource

ip_address_pool_id - the ID of the IP address pool to which this join belongs

network_address - the address of a VLAN network address that will be associated with this IP address pool

network_id - the ID of the network

pxe - true, if this address can be used for cloud booting a compute resource

free - true if free, otherwise false

netmask — netmask for the IP address

To delete an IP address assignment from a particular storage server, use the following request:

DELETE storage_servers/:storage_server_id/ip_addresses/:id.xml
DELETE storage_servers/:storage_server_id/ip_addresses/:id.json

XML Request Example

```
```

JSON Request Example

```
```

4.16.24 Get CDN Storage Server CPU Usage Statistics

To view CPU usage statistics of a CDN storage server, use the following request:

GET /storage_servers/:storage_server_id/cpu_usage.xml
GET /storage_servers/:storage_server_id/cpu_usage.json

Define a shorter period by setting Start and End time in the API call:

**XML Request Example**

```
curl -i GET -u user:userpass --url http://onapp.test/storage_servers/12/cpu_usage.xml
```

**JSON Request Example**

```
curl -i GET -u user:userpass --url http://onapp.test/storage_servers/12/cpu_usage.json
```

Where you have to specify the storage server ID.

**4.16.25  Get CDN Storage Server Hourly Statistics**

You can view the hourly statistics for a particular storage server using the following request:

```
GET /storage_servers/:storage_server_id/vm_stats/:hourly_statistics_id.xml
GET /storage_servers/:storage_server_id/vm_stats/:hourly_statistics_id.json
```

Define a shorter period by setting Start and End time in the API call:

```
```

**XML Request example**

```
```

**XML Output Example**
<vm_stats type="array">
  <vm_hourly_stat>
    <created_at type="datetime">2013-05-01T00:00:25Z</created_at>
    <currency_code>USD</currency_code>
    <id type="integer">13868</id>
    <stat_time type="datetime">2013-05-01T00:00:00Z</stat_time>
    <user_id type="integer">1</user_id>
    <virtual_machine_id type="integer">156</virtual_machine_id>
    <vm_billing_stat_id type="integer">7780</vm_billing_stat_id>
    <total_cost type="float">0.0</total_cost>
    <vm_resources_cost type="float">0.0</vm_resources_cost>
    <usage_cost type="float">0.0</usage_cost>
  </vm_hourly_stat>
</vm_stats>

Where:
created_at – the timestamp in DB when this record was created
updated_at – the time stamp in DB when this record was updated
currency_code - currency in which this storage server is charged within the bucket
id – the hourly statistics ID. Use this ID for retrieving the billing statistics.
stat_time – the particular hour for which these statistics were generated
user_id - the ID of storage server owner
virtual_machine_id - ID of a storage server
vm_billing_stat_id - billing statistics ID.

total_cost – the total amount of money owed for the storage server specified by id parameter for a particular hour specified by stat_time parameter (total_cost = vm_resources_cost + usage_cost)
vm_resources_cost – the amount of money due for the storage server resources for the particular hour specified by stat_time parameter (memory, disks, templates)
usage_cost – the total due for storage server usage for this particular hour specified by stat_time parameter (data sent/received, bandwidth, CPU usage)

4.16.26  Get CDN Storage Server Billing Statistics
You can view the billing statistics for a particular storage server using the following request:
GET /storage_servers/:storage_server_id/vm_stats.xml
GET /storage_servers/:storage_server_id/vm_stats.json

Define a shorter period by setting Start and End time in the API call:

MM-DD+hh:mm:ss&period[enddate]=YYYY-MM-
DD+hh:mm:ss&period[use_local_time]=1

GET
/stage_rage_servers/:storage_server_id/vm_stats.json?period[startdate]=YYYY-
MM-DD+hh:mm:ss&period[enddate]=YYYY-MM-
DD+hh:mm:ss&period[use_local_time]=1

XML Output Example
<vm_stats>
  <created_at type="datetime">2013-05-02T06:00:27Z</created_at>
  <currency_code>USD</currency_code>
  <id type="integer">15490</id>
  <stat_time type="datetime">2013-05-02T06:00:00Z</stat_time>
  <updated_at type="datetime">2013-05-02T06:00:27Z</updated_at>
  <user_id type="integer">307</user_id>
  <virtual_machine_id type="integer">1214</virtual_machine_id>
  <vm_billing_stat_id type="integer">8089</vm_billing_stat_id>
  <billing_stats><disks type="array">
    <disk>
      <id type="integer">2430</id>
      <costs type="array">
        <cost>
          <value type="integer">100</value>
          <cost type="float">0.0</cost>
          <resource_name>disk_min_iops</resource_name>
        </cost>
        <cost>
          <value type="integer">5</value>
          <cost type="float">0.0</cost>
          <resource_name>disk_size</resource_name>
        </cost>
        <cost><value type="integer">0</value>
          <cost type="float">0.0</cost>
          <resource_name>data_read</resource_name>
        </cost>
        <cost>
          <value type="integer">0</value>
          <cost type="float">0.0</cost>
          <resource_name>data_written</resource_name>
        </cost>
        <cost>
          <value type="integer">0</value>
          <cost type="float">0.0</cost>
          <resource_name>reads_completed</resource_name>
        </cost>
        <cost>
          <value type="integer">0</value>
          <cost type="float">0.0</cost>
          <resource_name>writes_completed</resource_name>
        </cost>
      </costs>
      <label nil="true"/>
    </disk>
    <disk>
      <id type="integer">2431</id>
      <costs type="array">
        <cost>
          <value type="integer">100</value>
          <cost type="float">0.0</cost>
          <resource_name>disk_min_iops</resource_name>
        </cost>
        <cost>
          <value type="integer">1</value>
          <cost type="float">0.0</cost>
          <resource_name>disk_size</resource_name>
        </cost>
        <cost>
          <value type="integer">0</value>
          <cost type="float">0.0</cost>
          <resource_name>data_read</resource_name>
        </cost>
        <cost>
          <value type="integer">0</value>
          <cost type="float">0.0</cost>
          <resource_name>data_written</resource_name>
        </cost>
      </costs>
    </disk>
  </billing_stats>
</vm_stats>
<cost type="integer">0</cost>
<resource_name>reads_completed</resource_name>
</cost>

<cost type="integer">0</cost>
<resource_name>writes_completed</resource_name>
</cost>

<cost type="integer">0</cost>
<resource_name>template</resource_name>
</cost>

<cost type="integer">0</cost>
<resource_name>cpu_usage</resource_name>
</cost>

<total_cost type="float">0.0</total_cost>
<vm_resources_cost type="float">0.0</vm_resources_cost>
<usage_cost type="float">0.0</usage_cost>
Where:

- `created_at` – the timestamp in DB when this record was created
- `updated_at` – the time stamp in DB when this record was updated
- `currency_code` - currency in which this storage server is charged within the bucket
- `id` – the ID of the storage server hourly statistics
- `stat_time` – the particular hour for which these statistics were generated
- `user_id` - the ID of storage server owner
- `virtual_machine_id` - ID of a storage server
- `virtual_machine_billing_statistics_id` -ID of a storage server billing statistics
- `billing_stats` - an array of billing details for the resources used by this storage server:
  - **disks** - an array of disks used by this storage server with their billing details:
    - `label` - disk name used in UI
    - `id` - disk ID used in database
    - `costs` - an array of disk related resources with their total prices for the period specified in the stat-time parameter, where:
      - `resource_name` - the resource in question. This can be disk_size, data_read, data_written, reads_completed and writes_completed
      - `value` - the amount of resources used (GBs of disk size, Kbs of data read/written, the number of reads/writes)
      - `cost` - the total due for the resource
  - **network_interfaces** - an array of network interfaces used by this storage server with their billing statistics:
    - `label` - network interface name used in OnApp
    - `id` - network interface ID
    - `costs` - an array of network interface related resources with their total prices for the period specified in the stat-time parameter, where:
      - `resource_name` - the resource in question. This can be ip_addresses, rate, data_received and data_sent
      - `value` - the amount of resources used by this network interface (the number of IPs, the port speed in Mb per second, the data sent and received in GB)
      - `cost` - the total due for the resource
  - **virtual_machines** - an array of storage server billing details:
    - `label` - storage server name
    - `costs` - An array of storage server resources with their total prices for the period specified in the stat-time parameter, where:
      - `resource_name` - the resource in question. This can be cpu_shares, cpus, memory, cpu_usage and template
      - `value` - the amount of resources allocated to this storage server. For the templates resource, this parameter means a template ID in database.
      - `cost` - the total due for this resource
    - `id` - storage server ID
  - **total_cost** – the total amount of money owed for the storage server specified by id parameter for a particular hour specified by stat_time parameter (total_cost = vm_resources_cost + usage_cost)
- `vm_resources_cost` – the amount of money due for the storage server resources for the particular hour specified by `stat_time` parameter (memory, disks, templates)
- `usage_cost` – the total due for storage server usage for this particular hour specified by `stat_time` parameter (data sent/received, bandwidth, CPU usage)

### 4.16.27 Search CDN Storage Server by Label

To search a storage server by label, use the following request:

GET /storage_servers.xml?q=label
GET /storage_servers.json?q=label

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

Where you have to specify the label of a CDN storage server you are searching for.

### 4.16.28 Get List of CDN Storage Server Backups

CDN storage server backups are managed exactly the same as virtual server backups. See the [Backups/Snapshots](#) section of the API guide for details.

GET /storage_servers/:storage_server_id/backups.xml
GET /storage_servers/:storage_server_id/backups.json

An array of backups is returned. If there are no backups, an empty array is returned.

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

**XML Output Example**
Where:

- **allowed_resize_without_reboot** – true if resizing CPU & RAM is allowed without restarting the storage server backed up
- **allowed_hot_migrate** – true if hot migration is allowed for the storage server backed up
- **allowed_swap** – true if swap disk is allowed for storage server backed up, otherwise false
- **backup_server_id** – the ID of the backup server on which the backup is stored
- **backup_size** – the disk space taken by this backup in kB
- **backup_type** – disk backup
- **built** – true if the storage server backed up has been built
- **built_at** – the date when the disk backup was built
- **created_at** – the date when the record in the database was created
- **updated_at** – the date when this record in database was updated
- **disk_id** – the id of a disk backed up
- **id** – the ID of this backup
- **identifier** - disk identifier
- **image_type** - backup type (currently only tar is available)
- **locked** – true if the storage server backed up has been locked
- **marked_for_delete** – the backup is marked for deletion (for auto-backups)
- **min_disk_size** – the minimum disk size
- **operating_system_distro** – the OS distribution of the storage server backed up
operating_system – the OS of the storage server backed up

template_id – the ID of a template from which the storage server backed up was built

user_id - the ID of a user the storage server belongs to

volume_id - data store ID

SolidFire - related parameters:

iqn - volume ISCSI qualified name

### 4.16.29 Add/Edit Admin/User Note for CDN Storage Server

To edit/make an admin note, use the following request:

PUT /storage_servers/:storage_server_id.xml

PUT /storage_servers/:storage_server_id.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where:

**admin_note** – enter the text of your note.

To edit/make a user note, use the following request:

**XML Request Example**

```
```

**JSON Request Example**

```
```

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no edge server with a requested ID, or URL is incorrect.
4.17 CDN Usage Statistics API

CDN usage statistics shows detailed information on all the resources used by CDN. To view the CDN usage statistics, use the following request:

GET /cdn_usage_statistics.xml
GET /cdn_usage_statistics.json

XML Request Example


JSON Request Example


XML Output Example

<cdn_stats>
  <cdn_stat>
    <user_id>1</user_id>
    <edge_group_id>6</edge_group_id>
    <cached>20411832</cached>
    <non_cached>1074815565</non_cached>
    <location_id>532</location_id>
  </cdn_stat>
</cdn_stats>

Where:

user_id - the ID of the OnApp user
edge_group_id - the ID of the edge group in use
cached - the amount of cached data in bytes
non_cached - the amount of non-cached data in bytes
location_id - the ID of the location assigned to this particular edge group (edge_group_id) in OnApp.

You can view the list of locations via the Get CDN Edge Group Details API request, where available_locations parameter is the array of locations available for this edge group.
4.18 DNS Setup API

The DNS setup chapter provides OnApp customers with information how to create a DNS hostname. After you create a DNS hostname, you get access to creating and managing DNS zones.

4.18.1 Get DNS Domain Details

To view the DNS domain details, use the following request:

GET /settings/dns_setup.xml
GET /settings/dns_setup.json

XML Request Example

```
```

JSON Request Example

```
```

XML Output Example

```
<dns_setup>
  <domain>mydns.com</domain>
</dns_setup>
```

Where:

domain – DNS domain name.

4.18.2 Set Up DNS Domain

To set up DNS domain, use the following request:

POST /settings/dns_setup.xml
POST /settings/dns_setup.json

XML Request Example

```
```

JSON Request Example
Where:
The only required parameter is *domain* – DNS domain name.

**XML Output Example**

```xml
<dns_setup>
  <domain>mydns.com</domain>
</dns_setup>
```

**4.18.3 Edit Domain**

To edit DNS domain, use the following request:

PUT /settings/dns_setup.xml
PUT /settings/dns_setup.json

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```

Where:
The only required parameter is *domain* – DNS domain name.

You will get a 204 status response on success and 404 if the domain name is invalid or you entered incorrect URL.

**4.18.4 Get List of Glue Records**

To get the list of glue records, use the following request:

GET /settings/dns_setup/glue_records.xml
GET /settings/dns_setup/glue_records.json

**XML Request Example**

```bash
```

**JSON Request Example**

```bash
```
XML Output Example

```xml
<hash>
  <ns1>109.23.125.206</ns1>
  <ns2>109.23.125.206</ns2>
  <ns3>109.23.125.206</ns3>
  <ns4>109.23.125.206</ns4>
</hash>
```

Where:

ns1, ns2, ns3, ns4 – DNS domain glue records.

4.19 DNS Zones API

OnApp DNS Zone feature allows you to manage your and your clients’ domain DNS. Each time DNS zone, record or setup settings are refreshed, the DNS configuration is immediately updated on the DNS vendor server.

4.19.1 Get List of Own DNS Zones

To get the list of your own DNS zones, use the following request:

GET /dns_zones.xml
GET /dns_zones.json

XML Request Example

```bash
```

JSON Request Example

```bash
```

XML Output Example

```xml
<dns_zones type="array">
  <dns_zone>
    <name>dns_example</name>
    <created_at type="datetime">2011-12-19T12:51:02Z</created_at>
    <updated_at type="datetime">2011-12-19T12:51:02Z</updated_at>
    <id type="integer">5</id>
    <user_id type="integer">1</user_id>
  </dns_zone>
</dns_zones>
```
Where:

name – DNS zone name

created_at – the date when the DNS zone was created in the [YYYY][MM][DD]T[hh][mm][ss]Z format

updated_at – the date when the DNS zone was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

id – DNS zone ID

current_user – the ID of a user who has created a DNS zone

4.19.2 Get Domain Zone Details

To get the details for a particular domain zone, use the following request:

GET /dns_zones/:id.xml

GET /dns_zones/:id.json

XML Request Example

```
```

JSON Request Example

```
```

XML Output Example

```
<dns_zone>
  <name>abc3.com</name>
  <created_at type="datetime">2011-12-19T12:51:02Z</created_at>
  <updated_at type="datetime">2011-12-19T12:51:02Z</updated_at>
  <id type="integer">5</id>
  <user_id type="integer">1</user_id>
</dns_zone>
```

Where:

name – DNS zone name

created_at – the date when the DNS zone was created in the [YYYY][MM][DD]T[hh][mm][ss]Z format

updated_at – the date when the DNS zone was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

id – DNS zone ID

user_id – the ID of a user who has created a DNS zone

4.19.3 Get List of Users DNS Zones

To get the list of DNS zones created by users, use the following request:
GET /dns_zones/user.xml
GET /dns_zones/user.json

XML Request Example


JSON Request Example


XML Output Example

<dns_zones type="array">
  <dns_zone>
    <name>yoyohow.com</name>
    <created_at type="datetime">2011-12-30T11:56:55Z</created_at>
    <updated_at type="datetime">2011-12-30T11:56:55Z</updated_at>
    <id type="integer">13</id>
    <user_id type="integer">1</user_id>
  </dns_zone>
</dns_zones>

Where:

name – DNS zone name

created_at – the date in the [YYYY][MM][DD][hh][mm][ss]Z format

updated_at – the date when the DNS zone was updated in the [YYYY][MM][DD][hh][mm][ss]Z format

id – DNS zone ID

user_id – the ID of a user who has created a DNS zone

4.19.4 Add DNS Zone

To create a new DNS zone, use the following request:

POST /dns_zones.xml
POST /dns_zones.json

XML Request Example


JSON Request Example
curl -i -X POST http://onapp.test/dns_zones.json -d '{"dns_zone":{"name":"domain.com", "auto_populate":"1"}}' -u user:userpass
-H 'Accept: application/json' -H 'Content-type: application/json'

Where:

name* – name of a DNS zone you create

auto_populate – autopopulate option lets you automatically import your existing DNS settings.
To autopopulate your DNS settings, set the autopopulate value 1, otherwise set 0.
You will get a 200 status response on success, and 404 if there is no DNS zone with a requested ID or you entered incorrect URL.

4.19.5 Delete Zone
To delete a DNS zone, use the following request:
DELETE /dns_zones/:id.xml
DELETE /dns_zones/:id.json

XML Request Example

curl -i -X DELETE --url http://onapp.test/dns_zones/12.xml -u user:userpass

JSON Request Example

curl -i -X DELETE --url http://onapp.test/dns_zones/12.json -u user:userpass

You will get a 204 status response on success, and 404 if there is no DNS zone with a requested ID or you entered incorrect URL.

4.19.6 Get List of Name Servers
To get the list of name servers, use the following request:
GET /dns_zones/name_servers.xml
GET /dns_zones/name_servers.json

XML Request Example


JSON Request Example

XML Output Example

```xml
<strings type="array">
  <string>ns1.ay.my</string>
  <string>ns2.ay.my</string>
  <string>ns3.ay.my</string>
  <string>ns4.ay.my</string>
</strings>
```

4.19.7 Get List of DNS Zone Records

To view DNS zone records, use the following request:

GET /dns_zones/:dns_zone_id/records.xml
GET /dns_zones/:dns_zone_id/records.json

XML Request Example

```
```

JSON Request Example

```
```

XML Output Example
<dns_zone>
<name>example.com</name>
<created_at type="datetime">2012-01-19T16:53:47Z</created_at>
<updated_at type="datetime">2012-01-19T16:53:47Z</updated_at>
<id type="integer">322</id>
<user_id type="integer">1</user_id>
<records>
  <MX type="array">
    <dns_record>
      <name>@</name>
      <ttl type="integer">3600</ttl>
      <priority type="integer">10</priority>
      <id type="integer">3540</id>
      <type>MX</type>
      <hostname>mx1.me.com.akadns.net</hostname>
    </dns_record>
  </MX>
  <SRV type="array">
    <dns_record>
      <name>_xmpp._tcp</name>
      <ttl type="integer">86400</ttl>
      <port type="integer">5222</port>
      <weight type="integer">1</weight>
      <priority type="integer">0</priority>
      <id type="integer">4533</id>
      <type>SRV</type>
      <hostname>jabber.example.com</hostname>
    </dns_record>
  </SRV>
  <A type="array">
    <dns_record>
      <name>@</name>
      <ttl type="integer">20</ttl>
      <id type="integer">3547</id>
      <type>A</type>
      <ip>17.172.192.8</ip>
    </dns_record>
  </A>
  <CNAME type="array">
    <dns_record>
      <name>www</name>
      <ttl type="integer">3600</ttl>
      <id type="integer">3551</id>
      <type>CNAME</type>
      <hostname>www.me.com.edgekey.net</hostname>
    </dns_record>
  </CNAME>
  <PTR type="array">
    <dns_record>
      <name>0.0.0.0.0.0.0.8.b.d.0.1.0.0.2</name>
      <hostname>test.com</hostname>
      <ttl type="integer">2</ttl>
      <id type="integer">1255774</id>
      <type>PTR</type>
    </dns_record>
  </PTR>
  <AAAA type="array">
    <dns_record>
      <name>sdfgeg</name>
      <ttl type="integer">456</ttl>
      <id type="integer">4052</id>
      <type>AAAA</type>
    </dns_record>
  </AAAA>
  <TXT type="array">
    <dns_record>
  </TXT>
</records>
</dns_zone>
Where:

name – DNS zone name.

created_at – the date when the DNS zone was created in the [YYYY][MM][DD][hh][mm][ss]Z format

updated_at - the date when the DNS zone was updated in the [YYYY][MM][DD][hh][mm][ss]Z format

id – DNS zone ID

user_id – the ID of a user who has created a DNS zone

The array of DNS records sorted by type with their details:

MX – the array of mail exchange records with the following parameters:

name – DNS domain set for the record

ttl – time to live value

id – DNS zone ID

type – the type of the record. For this array, it is MX

priority - the mail server preference

hostname – DNS hostname

SRV – the array of service records with the following parameters:

name – DNS domain set for the record

ttl – time to live value
port – the port on this target host of this service.
weight – the proportion of traffic the server pointed to will handle.
priority – the priority of the target host
id – DNS zone ID
hostname – DNS hostname

A – the array of A host records with the following parameters:
name – DNS domain set for the record
ttl – time to live value
id – DNS zone ID
type – the type of the record. For this array, it is A
ip – domain IP

CNAME – the array of CNAME records with the following parameters:
name – DNS domain set for the record
ttl – time to live value
id – DNS zone ID
type – the type of the record. For this array, it is CNAME
hostname – DNS hostname

PTR – the array of PTR records with the following parameters:
name – DNS domain set for the record
hostname – DNS hostname
ttl – time to live value
id – DNS zone ID
type – the type of the record. For this array, it is PTR

AAAA – the array of AAAA record with the following parameters:
name – DNS domain set for the record
ttl – time to live value
id – DNS zone ID
type – the type of the record. For this array, it is AAAA
ip – domain IP

TXT – the array of TXT record with the following parameters:
name – DNS domain set for the record
ttl – time to live value
id – DNS zone ID
txt – TXT value
type - the type of the record. For this array, it is TXT

NS – the array of name server records with the following parameters:
name – DNS domain set for the record
ttl – time to live value
id – DNS zone ID
type – the type of the record. For this array, it is NS
hostname – DNS hostname
SOA – the array of start of authority record with the following parameters:
name – DNS domain set for the record
serial – DNS zone serial number
primaryNs – primary name server
retry - the amount of time your secondary name servers will wait to contact the primary name server again if the last attempt failed
ttl – time to live value
id – DNS zone ID
refresh – the number of seconds between update requests
type – DNS record name. For this array, it is SOA
minimum – value of negative caching (in seconds)
expire - the number of seconds a server will wait before considering the data invalid if it cannot reach the primary name server
hostmaster – a hostmaster e-mail address

Page History
v.5.3
- added PTR array of records with their parameters

4.19.8 Get DNS Record Details
To get the DNS record details, use the following request:
GET /dns_zones/:dns_zone_id/records/:record_id.xml
GET /dns_zones/:dns_zone_id/records/:record_id.json

XML Request Example
```
```

JSON Request Example
```
```

XML Output Example
<dns_record>
  <name>sub2</name>
  <ttl type="integer">121</ttl>
  <id type="integer">2689</id>
  <type>A</type>
  <ip>127.0.0.0</ip>
</dns_record>

Where:

- **name** – DNS zone name
- **ttl** – the time to live value
- **type** – the record type. This can be one of the following records: NS, A, AAAA, CNAME, MX, TXT or SRV
- **id** – DNS zone ID
- **ip** – domain IP

### 4.19.9 Add DNS Record

To create a DNS record, use the following request:

- POST /dns_zones/:dns_zone_id/records.xml
- POST /dns_zones/:dns_zone_id/records.json

#### XML Request Example

```bash
```

#### JSON Request Example

```bash
curl -i -X POST http://onapp.test/dns_zones/12/records.json -d '{"dns_record":{"name":"TEST_NAME","ttl":"111","type":"A","ip":"127.1.1.1"}}' -u user:userpass -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:

- **name** - DNS record name
- **ttl** - the time to live value
- **type** - the record type
- **ip** - host IP (for A and AAAA records)

#### XML Output Example

```xml
<dns_record>
  <name>sub2</name>
  <ttl type="integer">121</ttl>
  <id type="integer">2689</id>
  <type>A</type>
  <ip>127.0.0.0</ip>
</dns_record>
```
<dns_record>
  <name>test23</name>
  <ttl>111</ttl>
  <id type="integer">2696</id>
  <type>A</type>
  <ip>127.1.1.1</ip>
</dns_record>

Starting from the 3.0.6 version of the OnApp Cloud it is possible to use underscore character in the CNAME records.

4.19.10 Edit DNS Records

To edit a DNS record, use the following request:

PUT /dns_zones/:dns_zone_id/records/:record_id.xml
PUT /dns_zones/:dns_zone_id/records/:record_id.json

XML Request Example


JSON Request Example


Where:

MX record:
- name* – DNS domain set for the record
- ttl* – time to live value
- priority* – the mail server preference
- hostname* – DNS hostname

SRV record:
- name* – DNS domain set for the record
- ttl* – time to live value
- port* – the port on this target host of this service.
- weight* – the proportion of traffic the server pointed to will handle.
- priority* – the priority of the target host
- hostname* – DNS hostname
A record:
- *name* – DNS domain set for the record
- *ttl* – time to live value
- *ip* – domain IP

CNAME record
- *name* – DNS domain set for the record
- *ttl* – time to live value
- *hostname* – DNS hostname

PTR record
- *name* – DNS domain set for the record
- *ttl* – time to live value
- *hostname* – DNS hostname

AAAA record
- *name* – DNS domain set for the record
- *ttl* – time to live value
- *ip* – domain IP

TXT record
- *name* – DNS domain set for the record
- *ttl* – time to live value
- *txt* – TXT value

NS record
- *name* – DNS domain set for the record
- *ttl* – time to live value
- *hostname* – DNS hostname

SOA record
- *name* – DNS domain set for the record
- *ttl* – time to live value

**XML Output Example**

```xml
<dns_record>
  <name>@</name>
  <ttl type="integer">1001</ttl>
  <id type="integer">2680</id>
  <type>NS</type>
  <hostname>ns1.worldcdn-beta-operator.doubleukay.com</hostname>
</dns_record>
```

You will get a 204 status response on success, and 404 if there is no DNS zone with a requested ID or you entered incorrect URL.

**Page History**
v.5.3
• added PTR array of records with their parameters

4.19.11 Delete DNS Record
To delete a record, use the following request:
DELETE /dns_zones/:dns_zone_id/records/:record_id.xml
DELETE /dns_zones/:dns_zone_id/records/:record_id.json

XML Request Example

```bash
*curl -i -X DELETE --url http://onapp.test/dns_zones/12/records/24.xml
-u user:userpass
```

JSON Request Example

```bash
curl -X DELETE --url http://onapp.test/dns_zones/12/records/24.json
-u user:userpass
```

You will get a 204 status response on success, and 404 if there is no DNS record with a requested ID or you entered incorrect URL.

4.19.12 Search for DNS Zone
To search for a particular DNS zone, use one of the following requests:

• To search for a DNS zone in the list of DNS Zones in the cloud, use the following request:

XML Request Example

```bash
-d '<q>237</q>'
```

JSON Request Example

```bash
-d '{"q":237}"
```

• To search for a DNS zone in the list of user DNS Zones, use the following request:

XML Request Example

```bash
-d '<q>vv</q>'
```

JSON Request Example
4.20 Wildcard Invalidation Rules API

This chapter provides requests you can use to create and manage wildcard invalidation rules. Wildcard invalidation rules are available only for HTTP Pull and HTTP Push resources.

4.20.1 Get List of Wildcard Invalidation Rules

To view all wildcard invalidation rules for a CDN resource, use the following request:

**XML Request Example**
```bash
```

**JSON Request Example**
```bash
```

**XML Output Example:**
```xml
<dns_zone>
  <id>22</id>
  <name>237.150.25.in-addr.arpa</name>
  <user_id>1318</user_id>
  <created_at>2017-01-11T07:27:23.000+00:00</created_at>
  <updated_at>2017-01-11T07:27:23.000+00:00</updated_at>
  <cdn_reference>463782090</cdn_reference>
</dns_zone>
```

Where:
- id – DNS zone ID
- name – DNS zone name
- user_id – the ID of a user who has created a DNS zone
- created_at – the date in the [YYYY][MM][DD][hh][mm][ss]Z format
- updated_at – the date in the [YYYY][MM][DD][hh][mm][ss]Z format
4.20.2 Add Wildcard Invalidation Rules

You can set up to 5 invalidation rules. To add a wildcard invalidation rule, use the following request:

**POST /cdn_resources/:id/invalidations.xml**

**POST /cdn_resources/:id/invalidations.json**

**XML Request Example**

```
```

**JSON Request Example**

```
```

**Where:**

- **wildcard_invalidation_rule** - the path to the file you want to add to the list of invalidation rules
  
  - if you set, for example, `/abc/*` - all files and folders the label of which start with `abc` will be invalidated
  
  - if you set, for example, `/img/*.png` - all PNG files in the `img` folder will be invalidated

4.20.3 Invalidate Files Again

To invalidate files again by applying the previously configured invalidation rule, use the following request:

**PUT /cdn_resources/:id/invalidations.xml**
**PUT /cdn_resources/:id/invalidations.json**

**XML Request Example**

```
```

**JSON Request Example**

```
```

**XML Output Example**

```
<invalidations>
  <wildcard_invalidation_rule_id>434231429</wildcard_invalidation_rule_id>
</invalidations>
```

Where:

- `wildcard_invalidation_rule_id` - the ID of the rule you want to invalidate again

### 4.20.4 Delete Wildcard Invalidation Rules

To delete wildcard invalidation rules, use the following request:

DETECT /cdn_resources/:id/invalidations/:id.xml
DETECT /cdn_resources/:id/invalidations/:id.json

**XML Request Example**

```
```

**JSON Request Example**

```
```

Where you have to specify the ID of a rule you want to delete in the URL.
4.21 Web Application Firewall (WAF) API

This chapter provides requests you can use to enable and disable WAF protection and WAF rules.

You need to have the Edit CDN resource permission enabled to access the WAF menu of a CDN resource.

4.21.1 Check Status of WAF Protection

To check the status of WAF protection, use the following request:

GET /cdn_resources/:id/waf.xml
GET /cdn_resources/:id/waf.json

XML Request Example


JSON Request Example


XML Output Example

<resource>
  <waf_on type="boolean">true</waf_on>
  <waf_ruleset type="array">
    <waf_ruleset>
      <id type="integer">1</id>
      <name>DRUPAL-EXCLUSION-RULES-REQUEST</name>
    </waf>
  </waf_ruleset>
</resource>

Where:

waf_on - status of the WAF protection. True if WAF protection is on; otherwise false
id - the ID of the rule, see the list below for details.
name - the name of the rule you want to check, see the list below for details.

The list below indicates the names of available rules with their corresponding IDs:

- 2 - WORDPRESS-EXCLUSION-RULES-REQUEST
- 3 - IP-REPUTATION-REQUEST
- 4 - METHOD-ENFORCEMENT-REQUEST
- 5 - DDOS-PROTECTION-REQUEST
• 6 - SCANNER-DETECTION-REQUEST
• 7 - PROTOCOL-ENFORCEMENT-REQUEST
• 8 - PROTOCOL-ATTACK-REQUEST
• 9 - APPLICATION-ATTACK-LFI-REQUEST
• 10 - APPLICATION-ATTACK-RFI-REQUEST
• 11 - APPLICATION-ATTACK-RCE-REQUEST
• 12 - APPLICATION-ATTACK-PHP-REQUEST
• 13 - APPLICATION-ATTACK-XSS-REQUEST
• 14 - APPLICATION-ATTACK-SQLI-REQUEST
• 15 - APPLICATION-ATTACK-SESSION-FIXATION-REQUEST
• 16 - BLOCKING-EVALUATION-REQUEST
• 17 - DATA-LEAKAGES-RESPONSE
• 18 - DATA-LEAKAGES-SQL-RESPONSE
• 19 - DATA-LEAKAGES-JAVA-RESPONSE
• 20 - DATA-LEAKAGES-PHP-RESPONSE
• 21 - DATA-LEAKAGES-IIS-RESPONSE
• 22 - BLOCKING-EVALUATION-RESPONSE
• 23 - CORRELATION-RESPONSE

4.21.2 Enable/Disable WAF Protection

To enable or disable WAF protection, use the following request:

PUT /cdn_resources/:id/waf.xml
PUT /cdn_resources/:id/waf.json

XML Request Example

```
```

JSON Request Example

```
curl -i -X PUT -H "Accept: application/json" -H 'Content-type: application/json' -u user:userpass --url http://onapp.test/cdn_resources/12/waf.json -d '{"cdn_resource":{"waf_on":"true", "waf_ruleset_blacklists":[1,2,3]}}'
```

Where:

- `waf_on` - status of the WAF protection. True if WAF protection is on; otherwise false
- `waf_ruleset_blacklists` - number of the rule you want to blacklist. Leave empty [] if you want to unblacklist all rules
4.22 Get CDN API Status

To get the CDN API status, use the following request:

GET /sysadmin_tools/cdn/api_status.xml
GET /sysadmin_tools/cdn/api_status.json

XML Request Example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url 
http://onapp.test/sysadmin_tools/cdn/api_status.xml
```

JSON Request Example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url
http://onapp.test/sysadmin_tools/cdn/api_status.json
```

XML Output Example

```
<status>Errors found</status>
```

Where:

*status* - CDN API status