

# **OnApp Cloud 6.4 Edge 1 Installation Guide**

## Table of Contents

4.1	Prepare Servers Configuration.....	7
4.2	Install Control Panel Server .....	7
4.3	Install Compute Resources .....	7
4.4	Install Data Stores .....	7
4.5	Install Backup Server .....	8
4.6	Configure Cloud .....	8
6.1	Install CloudBoot Compute Resources .....	23
6.1.1	Configure CloudBoot Settings in BIOS.....	26
6.1.2	Configure InfiniBand .....	27
6.2	Install Static Compute Resources .....	28
6.3	Install Static KVM Compute Resources with Integrated Storage .....	34
7.1	Install LVM Data Store .....	37
7.2	Install Integrated Storage Data Store.....	39
7.3	Install SolidFire Data Store .....	39
8.1	Install Static Backup Server .....	41
8.2	Install CloudBoot Backup Server .....	45
8.3	Install Static Backup Servers with Integrated Storage.....	48
9.1	Configure Control Panel Settings.....	50
9.2	Configure Compute Resources.....	50
9.3	Configure Data Stores.....	50
9.4	Configure Networks.....	51
9.5	Configure Backup Servers .....	51
9.6	Configure Relations Between Entities.....	51
9.7	Configure Templates .....	52
9.8	Configure ISOs.....	52
9.9	Configure OVAs .....	52
9.10	Configure Buckets .....	53
11.1	24/7 Support .....	55
11.2	What Does OnApp Support in my Cloud? .....	55
11.3	Professional Services .....	55
11.4	OnApp Community .....	55
11.5	Knowledge Base.....	56
11.6	Documentation.....	56
12.1	Install CloudBoot Compute Resources .....	57
12.1.1	Configure CloudBoot Settings in BIOS.....	60
12.2	Install Static Compute Resources .....	61
13.1	Control Panel Configuration File Custom Values .....	69
13.2	The Full List of OnApp Hypervisor Custom Values .....	72
13.3	The Full List of Installer Options for Static Hypervisors .....	73
13.4	The Full List of Configuration Options for Static Hypervisors.....	74
13.5	Backup Server Default Settings Configuration .....	75
13.6	The Full List of Installer Options for Static Backup Server.....	75
13.7	The Full List of Configuration Options for Static Backup Server.....	76
13.8	The Full List of Installer Options for Static Backup Server Complete Installation.....	76
13.9	The Full List of Control Panel Installer Options .....	77
14.1	OnApp Database Server Installation .....	80
14.2	RabbitMQ Server Installation.....	83
14.3	Redis Server Installation.....	85
14.4	Zabbix Server Installation .....	86
14.5	Zabbix Server Uninstall.....	87

This document describes how to install the 6.4 version of the OnApp Cloud. Please read each section carefully, as it is vital to the smooth operation of OnApp Cloud.

# 1 Preparation

---

1. Read the [Technical Details](#)
2. Read the [Preparation Guide](#)

## 2 Installation

---

1. [Install Control Panel Server](#)
2. [Install Compute Resources](#)
3. [Install Data Stores](#)
4. [Install Backup Server](#)

## 3 Post Install Configuration

---

1. [Configure Cloud](#)

## 4 Installation Workflow

The following workflow includes steps that you run to set up your OnApp installation. This is a standard workflow for OnApp Cloud. You can also integrate OnApp Cloud with VMware solutions, such as [VMware Cloud Director](#) or [vCenter](#).

**See also:**

[Get Started Guide](#)

[Upgrade Guide](#)



### 4.1 Prepare Servers Configuration

Before you install OnApp, make sure your network, storage, and servers configuration meet the requirements.



### 4.2 Install Control Panel Server

The Control Panel server hosts the OnApp user interface and manages all the processes controlled by OnApp. You install the Control Panel server from a ready-made installer package provided by OnApp.



### 4.3 Install Compute Resources

Compute resources provide CPU, RAM, and storage resources for applications and virtual servers that you or your users run in the cloud. OnApp enables you to install and use Static and CloudBoot compute resources.



### 4.4 Install Data Stores

Data stores provide storage for templates, backups, ISOs, and virtual server disks. You can set up a separate server with NFS or SSH connection, use any block-based storage, or set up OnApp Integrated Storage.



## 4.5 Install Backup Server

Backup servers are used to store backups and templates of virtual servers running in the cloud to prevent data loss in the event of failure. You can install Static or CloudBoot backup servers.



## 4.6 Configure Cloud

After you finish setting up the servers, you can log in to your OnApp Control Panel user interface and configure relations between cloud entities.

## 5 Install Control Panel Server

### Before You Begin

- Review [Get Started Guide](#) to ensure that your environment is ready for the installation.
- Use a corresponding [option](#) of the Control Panel installer in case MySQL is already installed and configured.
- The installer output is redirected to `./onapp-cp-install.log`
- All the installer critical errors are written to `/var/log/messages`
- If you want to reinstall an existing Control Panel, see [Control Panel Migration Guide](#) for instructions.
- If you plan to deploy [Accelerator](#), refer to the [RabbitMQ Configuration for Accelerator](#) document for more details.
- We provide a pre-built [OVA template](#) to install your OnApp CP on another virtualization environment (like VMware, Hyper-V, AWS, etc) if you prefer. It is required to have 4 virtual CPU(s) processor, 16384MB of memory, VMXNET 3 network adapter and 250GB hard disk drive (thin-provisioned). Please contact your service provider to learn how to upload this template and create server instances from it.
- If you need to install other components, for example, OnApp Database Server, RabbitMQ Server, or Redis Server, see [OnApp Installation Components](#) for instructions. You can also [migrate an OnApp database](#) from MySQL to MariaDB, Percona Server, or Percona Cluster.
- MariaDB new version 10.4 has been using to serve OnApp's database since version 6.3 Edge 1. It is default for CentOS 7.x fresh installs, and an extra `--mariadb` option is required by the CP installer on CentOS 6.x

### See also:

[Get Started Guide](#)

[Upgrade Guide](#)

To install the Control Panel server, run the following procedure:

1. Update your server:

```
# yum update
```

2. Download the OnApp YUM repository file:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo-6.4.noarch.rpm
```

3. Install the OnApp Control Panel installer package:

```
# yum install onapp-cp-install
```

4. **(Optional)** You can set custom configuration options for Control Panel. It is important to set custom values before the installer script runs.

### The full list of custom configuration options for Control Panel.

Edit the `/onapp/onapp-cp.conf` file to set custom values for Control Panel:

#Template server URL

```
TEMPLATE_SERVER_URL='http://templates-manager.onapp.com'
```

# IPs (separated with coma) list for the SNMP to trap. This is the list of Control Panel IP addresses on which the traps sent from the compute resources are processed.

```
SNMP_TRAP_IPS=""
```

# OnApp Control Panel custom version

```
ONAPP_VERSION=""
```

# OnApp MySQL/MariaDB connection data (database.yml)

```
ONAPP_CONN_WAIT_TIMEOUT=15
ONAPP_CONN_POOL=30
ONAPP_CONN_RECONNECT='true'
ONAPP_CONN_ENCODING='utf8'
```

# MySQL/MariaDB server configuration data (in case of local server)

```
MYSQL_WAIT_TIMEOUT=604800
MYSQL_MAX_CONNECTIONS=500
MYSQL_LIMITNOFILE=8192
```

# [Use MariaDB instead of MySQL as OnApp database server](#) (Deprecated parameter. If you set any values for this parameter, they will not take effect)

```
WITH_MARIADB=0
```

# Configure the database server relative amount of available RAM

```
TUNE_DB_SERVER=1
```

# The number of C data structures that can be allocated before triggering the garbage collector. It defaults to 8 million. Only change this value if you understand what it does.

```
RUBY_GC_MALLOC_LIMIT=16000000
```

# sysctl.conf net.core.somaxconn value

```
NET_CORE_SOMAXCONN=2048
```

# The root of OnApp database dump directory (on the Control Panel box)

```
ONAPP_DB_DUMP_ROOT=""
```

# Remote server's (to store database dumps) IP, user, path, openssh connection options and number of dumps to keep

```
DB_DUMP_SERVER=""
DB_DUMP_USER="root"
DB_DUMP_SERVER_ROOT="/onapp/backups"
DB_DUMP_SERVER_SSH_OPT="-o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o PasswordAuthentication=no"
KEEP_DUMPS=168
DB_DUMP_CRON='40 * * * *'
```

# [Enable monit - tool for managing and monitoring Unix systems](#)

```
ENABLE_MONIT=1
```

# DEPRECATED: If enabled (the 1 value is set) - install (if local box) and configures RabbitMQ Server (messaging system) for the vCloud support. (Deprecated parameter. If you set any values for this parameter, they will not take effect)

```
ENABLE_RABBITMQ=1
```

# Rotate transactions' log files created more than TRANS\_LOGS\_ROTATE\_TIME day(s) ago

```
TRANS_LOGS_ROTATE_TIME=30
```

# Maximum allowed for uploading file size in bytes, from 0 (meaning unlimited) to 2147483647 (2GB). Default is 0.

```
MAX_UPLOAD_SIZE=0
```

# Timeout before ping Redis Server to check if it is started. Default is 10 sec.

```
REDIS_PING_TIMEOUT=10
```

# OnApp Control Panel SSL certificates (please do not change if you aren't familiar with SSL certificates)

# \* The data below to generate self-signed PEM-encoded X.509 certificate

```
SSL_CERT_COUNTRY_NAME="UK"
SSL_CERT_ORGANIZATION_NAME='OnApp Limited'
SSL_CERT_ORGANIZATION_ALUNITNAME='OnApp Cloud'
SSL_CERT_COMMON_NAME="`hostname --fqdn 2>/dev/null`"
```

# SSLCertificateFile, SSLCertificateKeyFile Apache directives' values

# ssl\_certificate, ssl\_certificate\_key Nginx directives' values

```
SSLCERTIFICATEFILE="/etc/pki/tls/certs/ca.crt"
SSLCERTIFICATECSRFILE="/etc/pki/tls/private/ca.csr"
SSLCERTIFICATEKEYFILE="/etc/pki/tls/private/ca.key"
```

# \* PEM-encoded CA Certificate (if custom one exists)

# SSLCACertificateFile, SSLCertificateChainFile Apache directives' values

# ssl\_client\_certificate Nginx directives' values

```
SSLCACERTIFICATEFILE=""
SSLCERTIFICATECHAINFILE=""
```

# SSLCipherSuite, SSLProtocol Apache directives' values

# ssl\_ciphers, ssl\_protocols Nginx directives' values

```
SSLCIPHERSUITE=""
SSLPROTOCOL=""
```

```
# vi /onapp/onapp-cp.conf
```

**For successful installation you need to accept the EULA.**

5. Run the Control Panel installer:

a. For CentOS 6:

```
# /onapp/onapp-cp-install/onapp-cp-install.sh -i SNMP_TRAP_IPS --mariadb
```

b. For CentOS 7:

```
# /onapp/onapp-cp-install/onapp-cp-install.sh -i SNMP_TRAP_IPS
```

## The full list of installer options for Control Panel.

```
# /onapp/onapp-cp-install/onapp-cp-install.sh -h
Usage: /onapp/onapp-cp-install/onapp-cp-install.sh [-c
CONFIG_FILE] [--mariadb | --mariadb-custom | --community | --
percona | --percona-cluster] [-m MYSQL_HOST] [--mysql-
port=MYSQL_PORT] [--mysql-sock[=MYSQL_SOCKET] [-p MYSQL_PASSWD]
[-d MYSQL_DB] [-u MYSQL_USER] [-U ADMIN_LOGIN] [-P
ADMIN_PASSWD] [-F ADMIN_FIRSTNAME] [-L ADMIN_LASTNAME] [-E
ADMIN_EMAIL] [-v ONAPP_VERSION] [-i SNMP_TRAP_IPS] [--redis-
host=REDIS_HOST] [--redis-bind[=REDIS_BIND] [--redis-
passwd[=REDIS_PASSWD] [--redis-port=REDIS_PORT] [--redis-
sock[=REDIS_SOCKET] [--rbthost RBT_HOST] [--vcdlogin VCD_LOGIN]
[--vcdpasswd VCD_PASSWD] [--vcdvhost VCD_VHOST] [--rbtlogin
RBT_LOGIN] [--rbtpasswd RBT_PASSWD] [-a] [-y] [-D] [-t] [--
noservices] [--ha-install] [--rake=RAKE_TASKS] [--quick|--
quick-update[=SERVICE] [--accept-eula] [-w] [-h]
```

Database server options:

Default database SQL server is MySQL Server.

Please use one of the following option to install LOCALLY:

```
--mariadb : MariaDB Server
--mariadb-custom : MariaDB Server (custom for CentOS
7.x only)
--community : MySQL
Community Server
--percona : Percona Server
--percona-cluster : Percona
Cluster
```

```
-m MYSQL_HOST : MySQL host.
Default is 'localhost'
--mysql-
port=MYSQL_PORT : TCP port where MySQL Server serves
connections.
```

```
Default values is 3306 for the local installation
--mysql-sock[=MYSQL_SOCKET] : Unix
socket on which MySQL Server serves connections.
```

```
Default values is /var/lib/mysql/mysql.sock. Used if local
server only
```

```
The
socket is unset if the option's argument isn't specified.
-p MYSQL_PASSWD : MySQL password.
Random is generated if is not set or specified.
```

```
-d MYSQL_DB : OnApp MySQL
database name. Default is 'onapp'
```

```
-u MYSQL_USER : MySQL user. Default is 'root'
Redis
Server options:
```

```
--redis-host=REDIS_HOST : IP
address/FQDN where Redis Server runs. It is used by Control
```

Panel to connect to Redis Server.

The Redis Server will be installed and configured on the current box **if** localhost/127.0.0.1 or box's **public** IP address (listed in SNMP\_TRAP\_IPS) is specified.

Default value is 127.0.0.1.

If local Redis, it will serve as well on the unix socket '**PORT**' (**if** --redis-sock without argument isn't specified)

--redis-bind[=REDIS\_BIND] : The IP address **for** Redis Server to serve connections (to listen) option isn't mandatory.

--redis-port=REDIS\_PORT : Redis Server listen port.

Defaults are:

0 - **if** local server  
6379 - **if** remote server

--redis-passwd[=REDIS\_PASSWD] : Redis Server password to authenticate.

Random password is generated **if** the option's **argument** isn't specified.

By default no password is used **for** local Redis.

--redis-sock[=REDIS SOCK] : Path to the Redis Server's socket. Used **if** local server only. Default is /var/run/redis/redis.sock

The socket is unset **if** the option's **argument** isn't specified.

Options to manage OnApp Control Panel administrator account:

Please note, that these options are **for** NEW INSTALL only and not **for** upgrade

-P ADMIN\_PASSWD : CP administrator password  
-F ADMIN\_FIRSTNAME : CP administrator first name  
-L ADMIN\_LASTNAME : CP administrator last name  
-E ADMIN\_EMAIL : CP administrator e-mail

RabbitMQ Server and vCloud options:

--rbthost RBT\_HOST : IP address/FQDN where RabbitMQ Server runs.

The RabbitMQ will be installed and configured on the current box **if** localhost/127.0.0.1 or box's **public** IP address (enlisted

in SNMP\_TRAP\_IPS) Default  
 values is 127.0.0.1.

VCD\_\* : Options are usefull if vCloud/RabbitMQ are already installed and configured.

--vcdlogin VCD\_LOGIN : RabbitMQ/vCloud user. Default value is 'rbtvcd'.

--vcdpasswd VCD\_PASSWD : RabbitMQ/vCloud user password. The random password is generated if isn't specified.

--vcdvhost VCD\_VHOST : RabbitMQ/vCloud vhost. Default value is '/'

RBT\_\* : Options are used to configure RabbitMQ manager account. If local RabbitMQ server.

--rbtlogin RBT\_LOGIN : RabbitMQ manager login. The default value is 'rbtmgr'.

--rbtpasswd RBT\_PASSWD : RabbitMQ manager password. The random password is generated if isn't specified.

General options:

--ha-install : Proceed with Control Panel and Hight Availability components installation  
 RHEL/CentOS 7.x is supported only!

--rake RAKE\_TASKS : List of OnApp Control Panel rake tasks (separated with space) to run at the very end of install or upgrade

-v ONAPP\_VERSION : Install custom OnApp CP version

-i SNMP\_TRAP\_IPS : IP addresses separated with coma for snmp to trap

-y : Update OS packages (except of OnApp provided) on the box with 'yum update'.

-a : Do not be interactive. Process with automatic installation.

Please note, this will continue OnApp Control Panel install/upgrade even there is transaction currently running.

-t : Add to the database and download Base Templates. For new installs only.

--noservices : Do not start OnApp services: monit, onapp and httpd

Please note, crond and all OnApp's cron tasks remain running. They could be disabled by stopping crond service manually for your own risk.

-D : Do not make database dump,

and make sure it is disabled in the cron and not running at the moment

```

-w : Do not disable iptables
service.
Is applicable on fresh
installs only.

--quick|--quick-update[=SERVICE] :
Proceed with quick update procedure.
This will
skip update and configure for services, like: system
packages, MySQL database, Redis Server, RabbitMQ Server,
Monit service
Set the
SERVICE parameter (space separated list of statements) to
define services, which update is needed.
Possible
reserved statements are:
rpms -
for 'system packages' upgrade;
mysql -
for MySQL databse upgrade ond configuring;
redis -
for ERedis Server upgrade and configuring;
rabbitmq
- for RabbitMQ Server upgrade and configuring;
monit -
for Monit upgrade and configuring.

--accept-eula : Automatically accept OnApp's End User
License Agreement
(DEPRICATED)

-c CONFIG_FILE : Custom
installer configuration file. Otherwise, preinstalled one is
used.

-h : print this info

```

<b>Where:</b>	
<b>Database server options:</b>	Default database SQL server is MySQL Server. Please use one of the following option to install LOCALLY.
<b>--mariadb</b>	MariaDB Server
<b>--community</b>	MySQL Community Server
<b>--percona</b>	Percona Server
<b>--percona-cluster</b>	Percona Cluster
<b>MYSQL_*</b>	Options are useful if MySQL is already installed and configured.

<b>Where:</b>	
<b>-m MYSQL_ HOST</b>	MySQL host. Default is 'localhost'
<b>--mysql- port=MYS QL_POR T</b>	TCP port where MySQL Server serves connections. Default values is 3306 for the local installation
<b>--mysql- sock[=M YSQL_S OCK]</b>	Unix socket on which MySQL Server serves connections. Default values is /var/lib/mysql/mysql.sock. Used if local server only. The socket is unset if the option's argument isn't specified.
<b>-p MYSQL_ PASSWD</b>	MySQL password. Random is generated if is not set or specified.
<b>-d MYSQL_ DB</b>	OnApp MySQL database name. Default is 'onapp'.
<b>-u MYSQL_ USER</b>	MySQL user. Default is 'root'.
<b>Redis Server options:</b>	
<b>REDIS_*</b>	Options are useful if Redis Server is already installed and configured.
<b>--redis- host=RE DIS_HOS T</b>	IP address/FQDN where Redis Server runs. It is used by Control Panel to connect to Redis Server. The Redis Server will be installed and configured on the current box if localhost/127.0.0.1 or box's public IP address (listed in SNMP_TRAP_IPS) is specified. Default value is 127.0.0.1. If local Redis, it will serve as well on the unix socket 'PORT' (if --redis-sock without argument isn't specified).
<b>--redis- bind[=RE DIS_BIND ]</b>	The IP address for Redis Server to serve connections (to listen). The option isn't mandatory.
<b>--redis- port=RED IS_PORT</b>	Redis Server listen port. Defaults are: 0 - if local server 6379 - if remote server
<b>--redis- passwd[= REDIS_P ASSWD]</b>	Redis Server password to authenticate. Random password is generated if the option's argument isn't specified. By default no password is used for local Redis.
<b>--redis- sock[=RE DIS_SOC K]</b>	Path to the Redis Server's socket. Used if local server only. Default is /var/run/redis/redis.sock. The socket is unset if the option's argument isn't specified.
<b>Options to manage the OnApp Control Panel administrator account:</b>	

<b>Where:</b>	
<b>ADMIN_*</b>	Options are used to configure OnApp Control Panel administrator data. Please note that these options are for NEW INSTALL only and not for upgrade
<b>-P ADMIN_P ASSWD</b>	CP administrator password
<b>-F ADMIN_F IRSTNAM E</b>	CP administrator first name
<b>-L ADMIN_L ASTNAM E</b>	CP administrator last name
<b>-E ADMIN_E MAIL</b>	CP administrator e-mail
RabbitMQ Server and vCloud options:	
<b>-- rbthost RBT_HO ST</b>	IP address/FQDN where RabbitMQ Server runs. The RabbitMQ will be installed and configured on the current box if localhost/127.0.0.1 or box's public IP address (enlisted in SNMP_TRAP_IPS) Default value is 127.0.0.1.
<b>VCD_*</b>	Options are usefull if vCloud/RabbitMQ are already installed and configured.
<b>-- vcdlogin VCD_LO GIN</b>	RabbitMQ/vCloud user. Default value is 'rbtvcd'.
<b>-- vcdpass wd VCD_PA SSWD</b>	RabbitMQ/vCloud user password. The random password is generated if isn't specified.
<b>-- vcdvhost VCD_VH OST</b>	RabbitMQ/vCloud vhost. Default value is '/'
<b>RBT_*</b>	Options are used to configure RabbitMQ manager account. If local RabbitMQ server.
<b>-- rbtlogin RBT_LO GIN</b>	RabbitMQ manager login. The default value is 'rbtmgr'.
<b>-- rbtpassw d</b>	RabbitMQ manager password. The random password is generated if isn't specified.

<b>Where:</b>	
<b>RBT_PAS SWD</b>	
<b>General options:</b>	
<b>--rake RAKE_T ASKS</b>	List of OnApp Control Panel rake tasks (separated with space) to run at the very end of install or upgrade.
<b>-v ONAPP_ VERSION</b>	Install custom OnApp CP version
<b>-i SNMP_T RAP_IPS</b>	IP addresses separated with coma for snmp to trap
<b>-y</b>	Update OS packages (except of OnApp provided) on the box with 'yum update'.
<b>-a</b>	Is not interactive. Process with automatic installation. Please note, this will continue OnApp Control Panel install/upgrade even if there is transaction currently running.
<b>-t</b>	Add to the database and download Base Templates. For new installs only. If this option is not used, then only the following mandatory System Templates will be added by default during fresh install: OnApp CDN Appliance; Load Balancer Virtual Appliance; Application Server Appliance.
<b>-- noservice s</b>	Do not start OnApp services: monit, onapp and httpd Please note, crond and all OnApp's cron tasks remain running. They could be disabled by stopping crond service manually for your own risk.
<b>-D</b>	Do not make database dump, and make sure it is disabled in the cron and not running at the moment.
<b>-w</b>	Do not disable the iptables service. It is applicable on fresh installs only.
<b>--quick -- quick- update[= SERVICE ]</b>	Proceed with quick update procedure. This will skip update and configuration for services, such as system packages, MySQL database, Redis Server, RabbitMQ Server, and Monit service. Set the SERVICE parameter (space separated list of statements) to define services, which need to be updated. Possible reserved statements are: rpms - for 'system packages' upgrade; mysql - for MySQL databse upgrade ond configuring; redis - for ERedis Server upgrade and configuring; rabbitmq - for RabbitMQ Server upgrade and configuring; monit - for Monit upgrade and configuring.
<b>--accept- eula</b>	Automatically accept OnApp's End User License Agreement.
<b>-c CONFIG_ FILE</b>	Custom installer configuration file. Otherwise, preinstalled one is used.
<b>-h</b>	Print this info

**Perform the steps from six to nine (6-9) only if you are going to use CloudBoot and/or OnApp Storage.**

6. Install CloudBoot dependencies:  
Install CloudBoot default ramdisk:

```
# yum install onapp-ramdisk-centos7-default
```

Depending on a compute resource type, you should install *onapp-ramdisk-DISTRO-FLAVOR* package(s) where:

```
DISTRO: centos6, centos7  
FLAVOR: xen, kvm
```

Depending on the need compute resource type(s), install only the corresponding package(s):

```
# yum install onapp-ramdisk-centos6-kvm  
# yum install onapp-ramdisk-centos6-xen  
# yum install onapp-ramdisk-centos7-kvm  
# yum install onapp-ramdisk-centos7-xen
```

7. Run the following utility to configure and customize ramdisks:

```
# /onapp/onapp-store-install/onapp-store-install.sh
```

8. Install an OnApp license to activate your Control Panel. Enter a valid license key via the OnApp UI. Your default OnApp credentials are **admin/changeme**. You can change a password via the Control Panel > **Users** menu.

After you enter a license key, it may take up to 15 minutes to activate the key.

9. Restart the OnApp service:

```
# service onapp restart
```

**Perform the following step (10) only if you plan to deploy Accelerator.**

10. If you plan to configure an Accelerator, run the following command:

- o For all compute resources:

```
cd /onapp/interface  
# rake hypervisor:messaging:configure
```

- o For certain compute resources only:

```
cd /onapp/interface
# rake hypervisor:messaging:configure['11.0.50.111 11.0.50.112']
```

To perform the configuration for a number of compute resources, separate their IP addresses with a space.

The command above runs on compute resources that are online. If some compute resources are offline, you should run the command again when they are online. The `rabbitmq_host` parameter in the `on_app.yml` file should contain the real IP address of a server with RabbitMQ installed. The `rabbitmq_host` parameter should not be set to 'localhost' or '127.0.0.1'. The server with RabbitMQ installed should be available from the compute resources. For information on manual configuration for Accelerator, refer to [RabbitMQ Configuration for Accelerator](#).

**Perform the steps from 11 to 13 if there are compute resources added in the cloud.**

11. Generate SSH keys that OnApp requires for you to access various elements of the cloud. The script provided generates and transfers keys as necessary. The script needs to be run on your Control Panel server. It overwrites any keys that already exist so if you have custom keys already installed, you need to add them again after running the script. You need to provide your login details to various servers during the script execution. Please follow the onscreen instructions.
12. If you install a new compute resource, connect to your Control Panel server via SSH, download, and run the script:

```
# wget http://downloads.repo.onapp.com/install-all-keys.sh
# /bin/sh install-all-keys.sh
```

Note that the script is applicable only if you configure your first compute resources in the cloud. If you already have some existing compute resources and install new ones, the script sets new SSH keys to all the compute resources, both new and existing.

13. If you add additional compute resources to an existing cloud, update the `authorized_keys` file by running the following script on the Control Panel server:

```
# ssh-copy-id -i /home/onapp/.ssh/id_rsa.pub root@HV_HOST_IP
```

**Perform the following step (14) if you do not plan to install a dedicated backup server.**

14. Mount the locations for templates and backups. If you do not have a dedicated backup server, you must mount your template and backup repositories to compute resources. If your template and backup repositories are located on the Control Panel server, you can mount them as follows:  
Add the repositories to `/etc/exports` on the Control Panel server and then restart the NFS service:

```
# /onapp/templates 192.168.10.0/24(rw,no_root_squash)
# /onapp/backups 192.168.10.0/24(rw,no_root_squash)
```

15. After you installed the Control Panel server, configure your [Cloud Settings](#).

After the configuration is completed, Control Panel will be available in both http and https protocols. For security reasons, we recommend either closing port 80 or opening port 443. This port is used for secure web browser communication. Data transferred across such connections are highly resistant to interception. Moreover, the identity of the remotely connected server can be verified with significant confidence.

If you use a time zone with 30-minute or 45-minute offsets, you need to modify the configuration file `/etc/crontab` and change the startup time (`rake vm:generate_hourly_stats`) from 0th minute to 30th or 45th minute, depending on a time zone.

In the script, you should replace

```
0 * * * * onapp cd /onapp/interface; RAILS_ENV=production rake
vm:generate_hourly_stats
```

with

```
30 * * * * onapp cd /onapp/interface; RAILS_ENV=production rake
vm:generate_hourly_stats
```

## 6 Install Compute Resources

Please do not create mixed compute zones. Do not add CloudBoot and static compute resources to one compute zone, as well as Xen and KVM compute resources to one compute zone. The reason is that Xen virtual servers cannot be migrated to a KVM compute resource and vice versa.

After you successfully installed the Control Panel server, you can follow the next procedures to set up Xen or KVM compute resources:

- [Install CloudBoot Compute Resources](#)
- [Install Static Compute Resources](#)
- [Install Static KVM Compute Resources with Integrated Storage](#) - proceed to this section if you plan to use OnApp Integrated Storage on static resources

**On this page:**

[Install CloudBoot Compute Resources](#)

- [Configure CloudBoot Settings in BIOS](#)
- [Configure InfiniBand](#)

[Install Static Compute Resources](#)

[Install Static KVM Compute Resources with Integrated Storage](#)

**See also:**

[Install Control Panel Server](#)

[Install Data Stores](#)

[Install Backup Server](#)

### 6.1 Install CloudBoot Compute Resources

---

#### Before you proceed

- The CloudBoot compute resource installation enables dynamic boot of compute resource servers without any persistent installation requirements.
- The servers must support and have PXE boot enabled on the Network Interface Card (set up in BIOS if not already enabled by default). See [Configure CloudBoot Settings in BIOS](#) for details.
- Disable CPU power-saving features in BIOS before you proceed to the compute resource installation.
- We strongly recommend you to deploy one or more backup servers for backups and virtual servers provisioning when using CloudBoot.

Follow this installation to enable CloudBoot for your compute resources.

1. Enable CloudBoot on the Control Panel:
  - Go to **Admin > Settings > Configuration > System > CloudBoot**.
  - Scroll down to the CloudBoot section and check the **Enable** box.
2. Enable Storage on the Control Panel:
  - Go to **Admin > Settings > Configuration > System > OnApp Storage**.
  - Scroll down to the OnApp Storage section and check the **Enable OnApp Storage** box.
  - Tick the **Use Local Read Path** checkbox to minimize the network throughput dependency for read heavy workloads.
3. Enter IP addresses for static content target and Control Panel server CloudBoot interface. Enter the relevant IPs in **Settings > Configuration > System > CloudBoot**. Static content, such as CloudBoot images, kernels, virtual server templates, can be hosted on a standalone NFS server if you want. The default configuration is to install everything on the Control Panel server.
4. Add CloudBoot IP addresses for compute resources:
  - Create a CloudBoot IP Net:  
**Admin > Settings > Compute Resources > CloudBoot IPs tab > New IP Net**.
  - Add an IP range to the new IP Net:  
**Admin > Settings > Compute Resources > CloudBoot IPs > Actions** icon next to the *IP net* > **Add New IP Range**.
  - Add CloudBoot IP Addresses:  
**Admin > Settings > Compute Resources > CloudBoot IPs > Create IP Address**.
5. Power on servers and allow them to boot the default image. Add servers to the Control Panel by selecting MAC addresses and assigning IP address via **Admin > Settings > Compute Resources > Add New CloudBoot Compute Resource**.
 

If you want to expose drives in compute resources to OnApp Storage, our integrated storage platform, then you must select them at this point. For more information on setting up and configuring CloudBoot, see the [CloudBoot Compute resources](#) section of the Admin guide.
6. CloudBoot compute resources mount the following locations automatically at boot:
  - */data* to */onapp/tools/recovery*
  - */onapp/templates* to */.templates*

If you are using CentOS 6, you need to manually create symbolic link */data* to */onapp/tools/recovery* folder adding this command to Custom Config on a compute resource:

1.
  - `ln -s /onapp/tools/recovery /data`
  - a. The NFS server from which these are mounted is defined by the **Static Config Target** parameter (see [Edit System Configuration](#) section for details). You can set the default Control Panel server IP to any other server. This change will affect all CloudBoot compute resources.

The following paths must be available in the static config target to make it possible to use CloudBoot:

- */tftpboot/export*
- */data*

- `/ftpboot/images`

Compute resources will use local templates (mounted from Static Config target) during the server provisioning if the **Use SSH File Transfer** configuration setting is disabled or the template has null `backup_server_id`.

2. If you do not have a Dedicated Backup Server in place, please use Custom Config to mount `/onapp/templates` and `/onapp/backup` from your Control Panel server or another NFS export.
3. After you have installed CloudBoot compute resource, proceed to the [Configure CloudBoot Settings in BIOS](#) section.

If you do not have a dedicated backup server, you must mount your template and backup repositories to compute resources. If your template and backup repositories are located on the Control Panel server, you can mount them as follows:

Add locations to `/etc/exports` on the Control Panel server:

```
# /onapp/templates 192.168.10.0/24(rw,no_root_squash)
# /onapp/backups 192.168.10.0/24(rw,no_root_squash)
```

Add locations to Custom Config on a compute resource and run them manually on the command line (in this example, we are mounting from 192.168.10.101):

```
# unlink /onapp/templates
# mkdir -p /onapp/backups && mount -t nfs
192.168.10.101:/onapp/backups /onapp/backups
# mkdir -p /onapp/templates && mount -t nfs
192.168.10.101:/onapp/templates /onapp/templates
```

## Accelerator Deployment

If you want to deploy Accelerator, run the following steps:

1. Run the following command:

```
# su onapp
# cd /onapp/interface
```

2. Run the following command on the Control Panel server:

- For all compute resources:

```
# rake hypervisor:messaging:configure
```

- For certain compute resources only:

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

To perform the configuration for a number of compute resources, separate their IP addresses with a space. The command above should be run after every reboot.

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

```
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{y3l?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
```

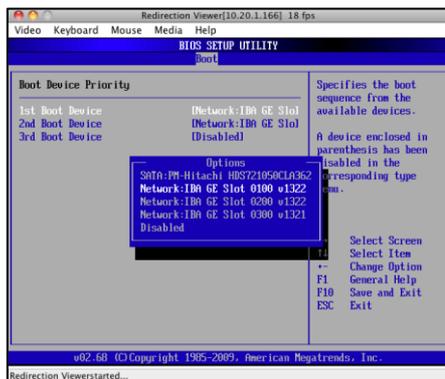
For information on manual configuration for Accelerator, refer to [RabbitMQ Configuration for Accelerator](#).

### 6.1.1 Configure CloudBoot Settings in BIOS

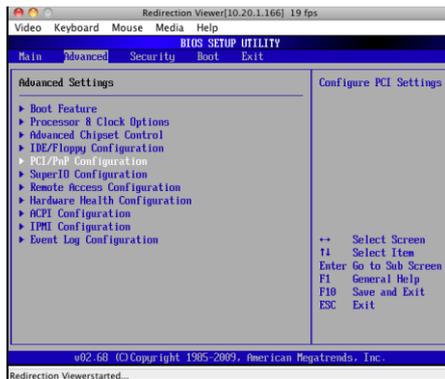
Your BIOS settings may vary from the example provided in this section.

To use PXE boot, you have to make sure it is enabled in BIOS.

1. Select the required Ethernet card supporting PXE as a boot device:



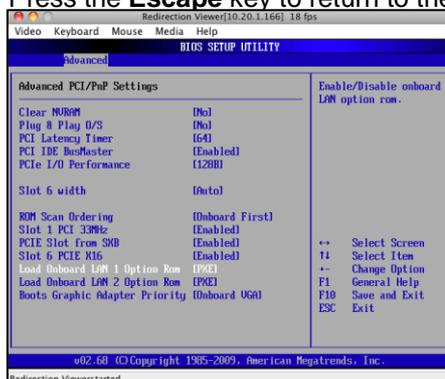
2. Go to the **Advanced** settings > **PCI/PnP Configuration**.



- In the **Advanced** settings, select the first/primary Onboard LAN/NIC Option ROM and press **Enter**.

Use up and down arrow keys to set Option ROM settings to enabled and press **Enter**.

Press the **Escape** key to return to the **Advanced** menu.



- Set a local disk as a second boot device.

## 6.1.2 Configure InfiniBand

Your hardware must meet the following requirements for the Ethernet mode utilization:

- VPI enabled switches (including a proper license key)
- VPI adapter cards (HCAs)

There are the following limitations for InfiniBand:

- To avoid compatibility issues, do not enable InfiniBand for the cloud with compute resources other than CentOS 6 KVM.
- Be aware that InfiniBand is not supported for CloudBoot CentOS 7 due to the lack of integrated OFED stack.
- InfiniBand in Ethernet mode is supported only for CentOS 6 KVM nodes.
- InfiniBand is supported only for the SAN network, not PXE boot.

To enable the InfiniBand mode:

1. Go to **Control Panel > Admin > Settings > Configuration** on the OnApp CP.
2. Move the *Enable InfiniBand Boot* slider in the CloudBoot section.
3. Click the **Save Configuration** button.

## 6.2 Install Static Compute Resources

---

### Before You Proceed

- Install base CentOS packages on the local drive before compute resource installation, depending on which virtualization method you choose:
  - Xen 4 compute resources: CentOS 6.x x86/64 or CentOS 7.x x86/64
  - KVM compute resources: CentOS 6.x x64 or CentOS 7.x x86/64
- We recommend installing CentOS from the minimal CentOS ISO for static compute resources.
- Disable CPU power-saving features in BIOS before you proceed to the compute resource installation.
- If you are not using a dedicated backup server in your cloud setup, configure NFS server with the following options to preserve files owner and group settings during template unpacking on NFS storage:
  - `no_root_squash`
  - `no_all_squash`
- Pay attention that smart and baremetal servers cannot be installed using the static compute resource installation method.
- If you plan to deploy baremetal servers, you can [enable recovery mode for baremetal servers](#).

### To install a compute resource:

1. Add a compute resource to your cloud using the OnApp Control Panel: **Admin > Settings > Compute resources > Add New Compute Resource**. Make sure that the compute resource is visible in the Control Panel and at this point is shown as inactive.
2. Update your server:

```
# yum update
```

3. Download the OnApp repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo-6.4.noarch.rpm
```

4. Install the OnApp compute resource installer package:

```
# yum install onapp-hv-install
```

5. Edit custom configuration for a compute resource. Custom values must be set before the installer script runs.

If you deploy Xen to a server running CentOS 6, it is important to specify a number for `XEN_DOM0_MAX_VCPUS`. We recommend to set the value to 2 if the compute resource has 12 cores or less; or 4 if the compute resource has more than 12 cores.

```
# vi /onapp/onapp-hv.conf
```

### The full list of custom values for a compute resource.

- o OnApp HV tools [custom version](#)

```
HV_VERSION=""
```

- o OnApp StorageAPI [custom version](#)

```
API_VERSION=""
```

- o Enable monit - tool for managing and monitoring Unix systems

```
ENABLE_MONIT=1
```

- o Default server to sync time on the compute resource

```
NTP_TIME_SERVER='pool.ntp.org'
```

- o Xen HV (Domain-0) related configuration

```
XEN_DOM0_MEM_MIN=409600
XEN_DOM0_MEM_DEVISOR=48
XEN_DOM0_MAX_VCPUS=""
XEN_DOM0_VCPUS_PIN_ENABLE=0
XEN_DOM0_SCHEDULER_WEIGHT=65535
XEN_DOM0_SCHEDULER_CAP=200
# 4.2.x and higher versions only
XEN_DOM0_SCHEDULER_RATELIMIT_US=100
XEN_DOM0_SCHEDULER_TIMESLICE_MS=5
```

- o The number of loopback devices created

```
LOOPBACKS=128
```

- The maximum size of the connection tracking table.

The value can't be greater than 65536 if the total memory of Xen Domain-0 or KVM is less than 1Gb.  
The value can be doubled (or even more, depending on the memory amount).

```
NET_IPV4_NETFILTER_IP_CONTRACK_MAX=""
```

- The divisor to calculate the hash table. The recommended value is 8.

```
hashsize = nf_contrack_max / 8  
CONTRACK_TO_HASHSIZE=8
```

- Outdated Xen compute resource's (Domain-0) configuration parameters

```
XEN_DOM0_MEM_OVERHEAD_MIN=262144  
P_TO_VCPUS=4
```

6. Run the OnApp compute resource installer script, depending on the compute resource:

- **For Xen**

```
# /onapp/onapp-hv-install/onapp-hv-xen-install.sh
```

### The full list of Xen installer options.

#### Usage:

```
# /onapp/onapp-hv-install/onapp-hv-xen-install.sh [-c CONFIG_FILE] [-v  
HV_VERSION] [-p API_VERSION] [-a] [-y] [-t] [-s] [-x] [-h]
```

#### Where:

-c CONFIG_FILE	Custom installer configuration file. Otherwise, the pre-installed one is used.
-v HV_VERSION	Custom compute resource Tools version
-p API_VERSION	Custom StorageAPI version
-a	Do NOT be interactive. Process with automatic installation.
-y	Update OS packages (except for OnApp provided) with 'yum update'.
-t	Initiate Recovery templates and ISO(s), which are used to provision FreeBSD guests, download. The download is initiated if the '-a' option is used.

-s	Skip packages management: install, remove, upgrade. None of the packages are installed/updated/removed.
-x	Skip xen*, kernel*, and libvirt* (compute resource related RPM) packages management. Useful for update (not for fresh install).

o **For KVM**

```
# /onapp/onapp-hv-install/onapp-hv-kvm-install.sh
```

Please refer to the [Install Integrated Storage on Static Compute Resources](#) section if you plan to use OnApp Integrated Storage on the compute resource (CentOS7 KVM only).

### The full list of KVM installer options.

#### Usage:

```
# /onapp/onapp-hv-install/onapp-hv-kvm-install.sh [-c CONFIG_FILE] [-a] [-y] [-t] [-s] [-x] [-v HV_VERSION] [-p API_VERSION] [-h]
```

#### Where:

-c CONFIG_FILE	Custom installer configuration file. Otherwise, the pre-installed one is used.
-a	Do NOT be interactive. Process with automatic installation.
-v HV_VERSION	Custom compute resource Tools version
-p API_VERSION	Custom StorageAPI version
-t	Initiate Recovery templates and ISO(s), which are used to provision FreeBSD guests, download. The download is initiated if the '-a' option is used.
-y	Update OS packages (except those provided by OnApp for a compute resource) with 'yum update'. Useful for update (not for fresh install).
-s	Skip packages management: install, remove, upgrade. Useful for update (not for fresh install).
-x	Skip kvm, kernel, and libvirt (compute resource related RPM) packages management. Useful for update (not for fresh install).
-d	Install OnApp Storage related packages. Applicable for CentOS 7.x only.
-h	Print this info

7. Configure the compute resource for your cloud. This step is also required for the SNMP statistics receiver configuration:

```
# /onapp/onapp-hv-install/onapp-hv-config.sh -h <CP_HOST_IP> -p
<HV_HOST_IP> -b <HV_BSNET_IP>
```

**The full list of compute resource (both Xen and KVM) configuration options.**

**Usage:**

```
# /onapp/onapp-hv-install/onapp-hv-config.sh [-h CP_HOST_IP] [-p
HV_HOST_IP] [-b HV_BSNET_IP] [-f FTS_IP] [-l LVM_HOST_ID] [-a|-i
[USER:PASSWD]] [-s] -?
```

**Where:**

-h CP_HOST_IP	FQDN or IP Addresses (separated by a comma) of the management server, which should receive all status reports and are authoritative for this compute resource. Used by <i>snmpd</i> and <i>snmptrapd</i> .
-p HV_HOST_IP	FQDN or IP Address of server (the compute resource) which will serve all stats-related and other requests send by the CP_HOST_IP. Used by <i>snmpd</i> , <i>snmptrapd</i> , and StorageAPI.
-b HV_BSNET_IP	Compute resource's IP Address from Backup Servers' network. Used to bind the SCSI target daemon.
-f FTS_IP	File Transfer Server FQDN or IP address, used for daily cron update recovery ISO by <i>recovery.sh</i>  <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; width: fit-content; margin: 10px auto;"> <p>If unsure, set the Control Panel server's management IP as CP_HOST_IP and FILE_TRANSFER_SERVER_IP.</p> </div>
-l LVM_HOST_ID	The <i>lvmlckd</i> sanlock <i>host_id</i> . The ID should be unique for each compute resource in the cloud. It's value for Static compute resources are between 257 and 2000.
-a	Install AoE
-s	Install sshfs
-?	Print this help info

8. If requested by the installer, reboot the compute resource to complete the installation:

```
# shutdown -r now
```

**Perform the steps from 9 to 11 if you haven't installed SSH keys for the compute resource yet. You may proceed to step 10 if you want to install all compute resources or install a particular compute resource at step 11.**

9. Generate SSH keys that OnApp requires for you to access various elements of the cloud. The script provided will generate and transfer keys as necessary. The script needs to be run on your Control Panel server. It will overwrite any keys that already exist, so if you have custom keys already installed, you will need to add them again after running the script. The script will ask you for your login details to various servers during the execution. Please follow the onscreen instructions.
10. If you install a new cloud instance, connect to your Control Panel server via SSH, download, and run the script:

```
# wget http://downloads.repo.onapp.com/install-all-keys.sh
# /bin/sh install-all-keys.sh
```

11. If you are add additional compute resources to an existing cloud, update the `authorized_keys` file by running the following script on the Control Panel server:

```
# ssh-copy-id -i /home/onapp/.ssh/id_rsa.pub root@HV_HOST_IP
```

**Perform the following step (12) if you do not plan to install a dedicated backup server.**

12. If you do not have a dedicated backup server, you must mount your template and backup repositories to compute resources. If your template and backup repositories are located on the Control Panel server, you can mount them as follows:

Make sure that locations are added to `/etc/exports` on the Control Panel server and then reboot.

```
# /onapp/templates 192.168.10.0/24(rw,no_root_squash)
# /onapp/backups 192.168.10.0/24(rw,no_root_squash)
```

Add locations to `/etc/rc.local` on the compute resource and run them manually on the command line (in this example, we are mounting from 192.168.10.101).

```
# mkdir -p /onapp/backups && mount -t nfs 192.168.10.101:/onapp/backups
/onapp/backups
# mkdir -p /onapp/templates && mount -t nfs
192.168.10.101:/onapp/templates /onapp/templates
```

13. Mount ISO locations:

To build virtual servers from ISO images, it is required to mount and share the location where the ISOs are stored at the Control Panel server with all the compute resources. When virtual servers are booted from ISOs, the ISO image is taken from the compute resource server. The location is preconfigured in `onapp.yml` config file:

- o `iso_path_on_cp` - specifies the location where ISOs are stored on the Control Panel server. By default, the location is `/data`. You can change it to any other

suitable location. Make sure that this location is shared with the specified `iso_path_on_hv` location.

- o `iso_path_on_hv` - specifies the location where ISOs are located on the compute resource servers. By default, the location is `/data`. You can change it to any other suitable location with the `onappowner` user and read/write access. Make sure that this location is mounted to the specified `iso_path_on_cp` location.

CloudBoot compute resources mount the `/data` location automatically at boot to the `/onapp/tools/recovery` on a compute resource.

ISOs can be hosted on a dedicated server at any desired location with an arbitrary name if you want. In this case, it is necessary to mount the ISOs location on this server to the Control Panel `iso_path_on_cp` directory and all the compute resources `iso_path_on_hv` locations. This can be a backup server to avoid the excess usage of the Control Panel resources.

### Accelerator deployment for static compute resources

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

1. Copy file:

```
# cp /home/mq/onapp/messaging/credentials{,_example,}.yaml
```

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

```
---
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?s81}' #RABBITMQ ACCESS PASSWORD
queue: 'hv-10.0.50.102' # hv-[IP Address of Compute Resource]
exchange:
  name: 'acceleration'
  type: 'direct'
  durable: True
```

3. Change an owner:

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

4. Run the following commands:

```
# service onapp-messaging start
# monit monitor onapp-messaging
```

## 6.3 Install Static KVM Compute Resources with Integrated Storage

---

**Before you proceed**

There are the following limitations for static compute resources with Integrated Storage:

- Integrated storage is supported only for CentOS 7 KVM compute resources.
- You must be running OnApp 6.1 version and higher.
- You must enable Storage in the system configuration first (**Settings > Configuration > OnApp Storage**). Visit [Configuration Settings](#) chapter for more details.
- Use the following recommended partitioning schema for your compute resource. The `/etc/fstab` should look like:

```
/dev/centos/root    /                xfs    defaults    0 0
UUID=...           /boot           xfs    defaults    0 0
/dev/centos/swap   /swap           swap   defaults    0 0
```

To install integrated storage on static compute resources:

1. Add a static compute resource to your cloud using the OnApp Control Panel: **Admin > Settings > Compute resources > Add New Compute Resource**. Make sure that the compute resource is visible in the Control Panel and at this point is shown as inactive.

2. Update your server:

```
# yum update
```

3. Download the OnApp repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo-6.4.noarch.rpm
```

4. Install the OnApp compute resource installer package:

```
# yum install onapp-hv-install
```

5. Run the OnApp compute resource installer script:

```
# /onapp/onapp-hv-install/onapp-hv-kvm-install.sh -d
```

6. Configure the compute resource for your cloud. This step is also required for the SNMP statistics receiver configuration:

```
# /onapp/onapp-hv-install/onapp-hv-config.sh -h <CP_HOST_IP> -p
<HV_HOST_IP> -b <HV_BSNET_IP>
```

7. Update the `authorized_keys` file by running the following script on the Control Panel server:

```
# ssh-copy-id -i /home/onapp/.ssh/id_rsa.pub root@HV_HOST_IP
```

If you do not have a dedicated backup server, you must mount your template and backup repositories to compute resources. If your template and backup repositories are located on the Control Panel server, you can mount them as follows:

Add locations to `/etc/exports` on the Control Panel server:

```
# /onapp/templates 192.168.10.0/24(rw,no_root_squash)
# /onapp/backups 192.168.10.0/24(rw,no_root_squash)
```

8. After you have installed integrated storage on static compute resource, configure your [Static Compute Resource Devices](#).
9. Reboot the compute resource to complete the installation.

## 7 Install Data Stores

After you successfully installed the Control Panel server and compute resources, you can follow one of the next procedures to set up data stores:

- [Install LVM Data Store](#)
- [Install Integrated Storage Data Store](#)
- [Install SolidFire Data Store](#)

### Before You Begin

- To configure an Integrated Storage data store, see the [Storage](#) guide.
- The following procedures assume that you have already configured a compute resource to see the iSCSI/ATAoE block device to connect to and that the SAN disk will be shown when running `fdisk -l`.
- All compute resources need access to the same data store. Ensure that you have the block device visible on all compute resources.
- OnApp doesn't support the 4 K block size for local LVM data stores.
- OnApp provides and maintains integration with StorPool, whereas each VS disk becomes a separate volume in the StorPool storage system. For details, refer to [StorPool section](#) of our documentation.
- **VERY IMPORTANT:** Perform the procedures only once per data store! Take care when selecting the disk/partition you want to use for storing virtual servers data!

### See also:

[Install Control Panel Server](#)

[Install Backup Server](#)

### 7.1 Install LVM Data Store

---

To install an LVM data store, follow the next procedure:

1. Add the new data store to OnApp via the Control Panel user interface:
  - Go to your Control Panel > **Admin** > **Settings** menu.
  - Click the **Data Stores** icon.
  - Click the **Create Data Store** button at the bottom of the screen.
  - Follow the steps in the creation wizard:

#### Step 1

- Enter a *Label* and *IP Address* for your data store.
- Select the data store type: **LVM**.
- Move the slider to the right to enable a data store. When disabled, OnApp will not allow new disks to be created automatically on that data store. This is useful to prevent an established data store from becoming too full. It also lets you prevent the automatic creation of root disks on 'special' data stores (high speed, etc).
- Click **Next**.

## Step 2

- Set disk capacity in GB.
  - If required, you can also bind the data store with a local compute resource. This is helpful if you want that the data store and a compute resource were located on the same physical server, thus decreasing the time needed for a compute resource-data store connection.
  - If required, you can also assign the data store to a data store zone. The drop-down menu lists all data store zones set up in the cloud (to add or edit data store zones, see [Data Store Zones](#)).
- When you are finished configuring the data store, click the **Create Data Store** button.  
To use the data store, you have to assign it either to a compute resource or a compute zone.
2. Find the data store unique identifier (this is needed to create your volume group in step #4): (Read the IDENTIFIER from the data stores screen: [http://xxx.xxx.xxx.xxx/settings/data\\_stores](http://xxx.xxx.xxx.xxx/settings/data_stores))
  3. Connect via SSH to a compute resource that is able to connect to this data store. Create the physical volume:

```
# pvcreate --metadatasize 50M /dev/xxx
```

Replace xxx with the real device.

4. Create the volume group:

```
# vgcreate onapp-IDENTIFIER /dev/xxx
```

Replace xxx with the real device and *IDENTIFIER* with the info from the data store page on the UI.

5. Test compute resource/volume group visibility. Now when you have the new data store formatted, you should be able to see the volume group from all compute resources. To test this, run *pvscan* and *vgscan* on all compute resources. Make sure you can see all identifiers on all compute resources.

## 7.2 Install Integrated Storage Data Store

---

Before creating an Integrated Storage data store:

1. Create one or more Xen or KVM compute resource with Integrated Storage enabled to group their drives together into a virtual data store.
2. Create a compute zone.
3. Add your compute resources to the compute zone.

After running these steps, you can proceed to the Integrated Storage data store creation.

To create a new Integrated Storage data store:

1. Go to your Control Panel > **Cloud > Integrated Storage > Data Stores** menu.
2. On the screen that appears, you'll see the list of all Integrated Storage data stores in the cloud.
3. To create a new data store, click the **Create New Integrated Storage Data Store** button, and complete the wizard that follows:
  - **Name** - give your data store a name
  - **Show advanced options** - select this check box to reveal the list of advanced settings:
    - *Replicas* - specify the number of data copies to increase the resilience to individual drive failure. You can specify 1, 2 or 4 replicas.
    - *Stripes* - specify the number of data splittings to increase the number of physical disks included to the virtual disk. You can specify 0, 2 or 4 stripes.
    - *Overcommit* - specify the over-provisioning percentage. You can set the following overcommit values: none (0%), 20%, 50% or unlimited (100%).

In order for your hard drives (nodes) to be detected and active, multicast traffic should be enabled on your switch, for the Onapp Integrated Storage Network/VLAN.

- **Storage Nodes**
    - *Filter by compute resource* - use this to filter the nodes (disks) available for inclusion in this data store, by specific compute resources.
    - *Filter by performance* - use this to filter the nodes available for inclusion in this data store by performance.
4. Click the **Save** button to create the data store. The data store must be assigned to a compute zone and data store zone before you can provision storage to a VS.

## 7.3 Install SolidFire Data Store

---

You can create one SolidFire data store per cloud that will represent the space available at the SolidFire side.

To create a SolidFire data store:

1. Go to your Control Panel > **Admin > Settings** menu.
2. Click the **Data Stores** icon.
3. Click the **Create Data Store** button at the bottom of the screen.

## 4. Follow the steps in the creation wizard:

**Step 1 of 3**

- Enter a data store label.
- Specify an IP address to be used for managing the data store via CP (Inasmuch SolidFire data stores have two interfaces, you'll have to specify the IP address for the cluster admin later.)
- Select a **SolidFire** data store type.
- Move the slider to the right to enable a data store. When disabled, OnApp will not allow new disks to be created automatically on that data store. This is useful to prevent an established data store from becoming too full. It also lets you prevent the automatic creation of root disks on 'special' data stores (high speed, etc).
- Click **Next**.

**Step 2 of 3**

- Set disk capacity in GB.
- If required, you can also bind the data store with a local compute resource. This is helpful if you wish that the data store and a compute resource were located on the same physical server thus decreasing the time needed for a compute resource-data store connection.
- If required, you can also assign the data store to a data store zone. The drop-down menu lists all data store zones set up in the cloud (to add or edit data store zones, see the section on Data store zones in the Settings section of this guide).

**Step 3**

- Specify the cluster Admin settings:
  - *iSCSI IP* - iSCSI IP address
  - *Username* - specify a username for cluster authorization
  - *Password* - specify a password for cluster authorization
- Specify the SolidFire Account settings:
  - *Username* - specify SolidFire account username
  - *Initiator secret* - specify iSCSI initiator secret (optional)
  - *Target secret* - specify iSCSI initiator secret (optional)

*Initiator secret* and *target secret* are optional parameters. They are created automatically for a newly created account. For the new account they will be taken from the SolidFire database. If you specify target and initiator secrets for an existing user, they will be overwritten.

5. When you've finished configuring the store, click the **Create Data Store** button.

## 8 Install Backup Server

Follow one of the following procedures to set up a backup server in your cloud:

- [Install Static Backup Server](#)
- [Install CloudBoot Backup Server](#)
- [Install Static Backup Server with Integrated Storage](#) - proceed to this section if you plan to use OnApp Integrated Storage on static backup servers

**On this page:**

[Install Static Backup Server](#)

[Install CloudBoot Backup Server](#)

[Install Integrated Storage on Static Backup Servers](#)

**See also:**

[Install Control Panel Server](#)

[Install Data Stores](#)

### 8.1 Install Static Backup Server

---

To install a static backup server, run the following procedure. Skip this section if you use CloudBoot. We recommend installing CentOS from the minimal CentOS ISO for static backup servers.

1. Add a backup server via the Control Panel user interface:
  - Go to your Control Panel > **Admin** > **Settings** menu and click the **Backup Servers** icon.
  - Click the **Create Backup Server** button.
  - Fill in the form that appears:
    - *Label* - give your backup server a label
    - *IP address* - enter the backup server IP address (IPv4)
    - *Backup IP address* - add a provisioning network IP address
    - *Capacity* - set the backup server capacity (in GB)
    - *Backup server zone* - select the backup server zone to which this backup server will be assigned.
  - Move the **Enabled** slider to the right to enable the backup server.
  - Click the **Add Backup Server** button.
  - Update your server:

```
# yum update
```

- Download the OnApp repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo-6.4.noarch.rpm
```

- Install the OnApp Backup Server installer package:

```
# yum install onapp-bk-install
```

- Check and set the backup server default settings:

```
# vi /onapp/onapp-bk.conf
```

### **Edit the backup server default settings in the /onapp/onapp-bk.conf file.**

#### **OnApp BK tools custom version**

```
BK_VERSION=""
```

#### **OnApp StorageAPI custom version**

```
API_VERSION=""
```

#### **Default server to synch time on the HV**

```
NTP_TIME_SERVER='pool.ntp.org'
```

#### **The number of retries for WGET to download the file**

```
WGET_TRIES=5
```

#### **OnApp templates directory**

Please refer to the corresponding settings at OnApp Control Panel web interface

```
TEMPLATES_DIR='/onapp/templates'
```

#### **OnApp backups directory**

Please refer to the corresponding settings at OnApp Control Panel web interface

```
BACKUPSS_DIR='/onapp/backups'
```

- Run the installer. It is recommended to download Base, Load Balancer, and CDN templates while running the installer. You may rerun the installer later with the `-t` option.

```
# sh /onapp/onapp-bk-install/onapp-bk-install.sh
```

## The full list of installer options.

### Usage:

```
# /onapp/onapp-bk-install/onapp-bk-install.sh [-c CONFIG_FILE] [-a] [-y]
[-t] [-b] [-v BK_VERSION] [-p API_VERSION] [-h]
```

### Where:

-c CONFIG_FILE	Custom installer configuration file. Otherwise, the pre-installed one is used.
-a	Do NOT be interactive. Process with automatic installation.
-v BK_VERSION	Custom backup tools version
-p API_VERSION	Custom StorageAPI version
-y	Update OS packages (except for the ones provided by OnApp) with 'yum update'.
-t	Initiate the Load Balancer Virtual Appliance, OnApp CDN Appliance, and Application Server Appliance templates download. The download is initiated if the '-a' option is used.
-b	Initiate the Base templates download
-h	Print this info

- Configure the backup server for your cloud. This step is also required for the SNMP statistics receiver configuration:

```
# /onapp/onapp-bk-install/onapp-bk-config.sh -h <CP_HOST_IP> -p
<BK_HOST_IP>
```

## The full list of configuration options.

### Usage:

```
# /onapp/onapp-bk-install/onapp-bk-config.sh [-h CP_HOST_IP] [ -p
BK_HOST_IP] [-a|-i [USER:PASSWD]] [-s] -?
```

### Where:

-h CP_HOST_IP	FQDN or IP Address of the management server which should receive all status reports and is authoritative for this backup server.
-p BK_HOST_IP	FQDN or IP Address of Backup Server which will serve all stats related and other requests send by the CP_HOST_IP. Used by snmpd and StorageAPI.
-a	Install AoE
-i [USER:PASSWD]	Install iSCSI utils and configure with USER and PASSWD (if specified)
-s	Install sshfs
-?	Print this help info

- Install SSH keys for the backup server by running the following script on the Control Panel server:

```
# ssh-copy-id -i /home/onapp/.ssh/id_rsa.pub root@BS_HOST_IP
```

### Perform the following step (9) if you plan to deploy VSs from OVA.

- Mount the locations for OVA. If you enable building VSs from [OVAs](#) in your cloud, you should mount the source location to which OVA is uploaded on CP with a destination location where OVA is stored on a backup server. The location is preconfigured in the *onapp.yml* config file:
  - **Source Path** (*ova\_path*) - specifies the location where OVAs are downloaded/uploaded on the Control Panel server. By default the location is */data*.
  - **Destination Path** (*ova\_path*) - specifies the location where OVAs are stored on backup servers. By default the location is */data*.

To mount the locations:

- Add locations (e.g. */data*) to */etc/exports* on the Control Panel server and then restart the nfs service:

```
# /data 192.168.10.0/24(rw,no_root_squash)
```

- Add locations to */etc/rc.local* on the backup server and run them manually on the command line (in this example we are mounting from 192.168.10.101):

```
# mkdir -p /data && mount -t nfs 192.168.10.101:/data /data
```

## 8.2 Install CloudBoot Backup Server

---

CloudBoot backup servers are CloudBoot KVM compute resources that can be used as backup servers. Follow the step-by-step instructions provided in this section to configure CloudBoot backup servers in your cloud.

### Before You Begin

- You should configure some local or remote attached storage for persistent backups on the provisioning/backup server. We strongly recommend you to deploy one or more backup servers for backups and virtual servers provisioning when using a CloudBoot functionality.
- We strongly recommend you to deploy one or more backup servers on your cloud. Incremental backups are only supported with a dedicated backup server.
- CentOS now defaults to NFSv4. This is known to cause compatibility issues so we strongly recommend that you use NFSv3 for all mounts. This can be done by passing `-t nfs -o vers=3` in any mount commands.
- To use a backup server, you have to add it to a [backup server zone](#) and assign it either to a [compute resource](#) or a [compute zone](#).

To create a CloudBoot backup server:

1. Add a backup server via the Control Panel user interface:
  - Go to your Control Panel > **Admin** > **Settings** menu and click the **Backup Servers** icon.
  - Click the **Create Backup Server** button.
  - Fill in the form that appears:
    - *Label* - give your backup server a label
    - *IP address* - enter the backup server IP address (IPv4)
    - *Backup IP address* - add a provisioning network IP address
    - *Capacity* - set the backup server capacity (in GB)
    - *Backup server zone* - select the backup server zone to which this backup server will be assigned.
  - Move the **Enabled** slider to the right to enable the backup server.
  - Click the **Add Backup Server** button.
- Update CloudBoot and CP server RPMs:

```
# yum update onapp-store-install yum update onapp-cp-install
```

- Configure CloudBoot settings:

```
# /onapp/onapp-store-install/onapp-store-install.sh
```

- Create a new CloudBoot compute resource: at the first step of the creation wizard, select the type - Backup (CloudBoot Provisioning and Backup Resource). Refer to the [Create CloudBoot Compute Resource](#) section of this guide for details.
- Go to your Control Panel > **Admin** > **Settings** menu and then click the **Compute Resources** icon.
- Click the **Add New CloudBoot Compute Resource** button.
- In the wizard menu, click Backup (CloudBoot Provisioning and Backup Resource).
- Select right MAC address of it.
- Fill in the form that appears:
  - *Label* - give the compute resource a name
  - *Pxe IP address* - select an IP address for this compute resource from the address pool available
  - *Enabled* - move the slider to the right to allow VSs to be installed/booted on this compute resource
  - *Compute Zone* - select the compute zone, to which this compute resource will be assigned, from the drop-down list
  - *Custom Config* - specify any custom commands you want to run when compute resource is booted
- At this step, specify the Backup Server properties:
  - *Label* - give your backup server a label
  - *IP address* - enter the IP address of a Compute resource you have created at step 1
  - *Backup IP address* - add a provisioning network IP address
  - *Capacity* - set the backup server capacity (in GB)
- Then, you shouldn't assign any drive, but it needs right NIC to be applied to the storage network.
- After that, assign your backup server to the backup server zone. If you want to attach LVM-based storage and create backups, you should also add the IP address of the KVM Compute resource added in step 1 in the 'Backup IP address' field of each of your compute resources.

#### Further steps:

1. Format and mount the local storage:
  - SSH to the backup server
  - Format the storage with your preferred filesystem type, e.g.:

```
# mkfs.ext4 /dev/sda
```

- Create a directory for backups if it does not exist

```
# mkdir /backupstorage
```

- Mount the storage to /onapp/backups:

```
# mount /dev/sda /backupstorage
```

- Create a directory for storing templates:

```
# mkdir /backupstorage/templates
```

- Create a directory for storing backups:

```
# mkdir /backupstorage/backups
```

- Create symbolic links in /onapp:

```
# ln -s /backupstorage/backups /onapp/backups
# unlink /onapp/templates
# ln -s /backupstorage/templates /onapp/templates
# ln -s /onapp/tools/recovery /data
```

- Add the following to custom config file:

```
# mkdir /backupstorage
# mount /dev/sda /backupstorage
# ln -s /backupstorage/backups /onapp/backups
# unlink /onapp/templates
# ln -s /backupstorage/templates /onapp/templates
# ln -s /onapp/tools/recovery /data
```

- Another example of the custom config files:

- If you have a separate partition for backups and templates (/dev/sda1 and /dev/sda2):

```
# mkdir /backupstorage
# mkdir /backupstorage/templates
# mkdir /backupstorage/backups
# mount /dev/sda1 /backupstorage/backups
# mount /dev/sda2 /backupstorage/templates
# ln -s /backupstorage/backups /onapp/backups
# unlink /onapp/templates
# ln -s /backupstorage/templates /onapp/templates
```

- If you current array is detected as /dev/sda1 and currently everything is located in /onapp within templates and backup directories within:

```
# mkdir /backupstorage
# mount /dev/sda1 /backupstorage/
# ln -s /backupstorage/backups /onapp/backups

# unlink /onapp/templates
# ln -s /backupstorage/templates /onapp/templates
```

2. Update the database so that the location of the templates is known:

- o Find the database password:

```
# cat /onapp/interface/config/database.yml |grep password
```

- o Open the onapp database in MySQL:

```
# mysql -p # use onapp;
```

- o Find the ID of the backup server:

```
# select * from backup_servers;
```

- o For all of the templates, set the required backup\_server\_id:

```
# update templates set backup_server_id='[your_id]';
```

3. To download the base templates during the installation to your Control Panel, download and run the following script:

```
# wget http://downloads.repo.onapp.com/get_template.sh
# /bin/sh get_template.sh
```

## 8.3 Install Static Backup Servers with Integrated Storage

### Before you proceed

There are the following limitations for installation of static backup servers with Integrated Storage:

- Integrated storage is supported only for CentOS 7 KVM compute resources.
- You must be running OnApp 6.1 version and higher.
- You must enable Storage in the system configuration first (**Settings > Configuration > OnApp Storage**). Visit [Configuration Settings](#) chapter for more details.

To install static backup server with Integrated Storage:

1. Update your server:

```
# yum update
```

2. Download the OnApp repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo-6.4.noarch.rpm
```

3. Install the OnApp compute resource installer package:

```
# yum install onapp-bk-install
```

4. Run the OnApp compute resource installer script:

```
# /onapp/onapp-bk-install/onapp-bk-install.sh -d
```

5. Configure the compute resource for your cloud. This step is also required for the SNMP statistics receiver configuration:

```
# /onapp/onapp-bk-install/onapp-bk-config.sh -h <CP_HOST_IP> -p  
<BK_HOST_IP>
```

6. Update the `authorized_keys` file by running the following script on the Control Panel server:

```
# ssh-copy-id -i /home/onapp/.ssh/id_rsa.pub root@HV_HOST_IP
```

If you do not have a dedicated backup server, you must mount your template and backup repositories to compute resources. If your template and backup repositories are located on the Control Panel server, you can mount them as follows:

Add locations to `/etc/exports` on the Control Panel server and then reboot:

```
# /onapp/templates 192.168.10.0/24(rw,no_root_squash)  
# /onapp/backups 192.168.10.0/24(rw,no_root_squash)
```

Add locations to `/etc/rc.local` on the compute resource and run them manually on the command line (in this example, we are mounting from 192.168.10.101):

```
# mkdir -p /onapp/backups && mount -t nfs 192.168.10.101:/onapp/backups  
/onapp/backups  
# mkdir -p /onapp/templates && mount -t nfs  
192.168.10.101:/onapp/templates /onapp/templates
```

7. After you have installed integrated storage on static backup server, [create your backup server](#) via the Control Panel user interface.
8. Reboot the backup server to complete the installation.

## 9 Configure Cloud

After you set up your infrastructure, the final step is to configure your cloud on your Control Panel user interface. This section provides information on how to configure a basic cloud. After you complete the following steps, you can create virtual servers in your cloud.



### 9.1 Configure Control Panel Settings

After you installed OnApp, you can apply the necessary Control Panel configurations. For more information, see the [Configuration Settings](#) section.



### 9.2 Configure Compute Resources

To deploy virtual servers, you need to create the [compute zone](#) and [compute resources](#). Next, you [add compute resources to compute zones](#) as follows:

- You can add Xen compute resources to compute zones of the Virtual and Baremetal types
- You can add KVM compute resources to compute zones of the Virtual and Smart types
- You can add vCloud Director compute resources to compute zones of the VPC type
- You can add vCenter compute resources to compute zones of the Virtual type

If you want to use Integrated Storage, select a compute zone as a storage API endpoint and make sure to enable Integrated Storage in the **Settings > Admin > Configuration** to group compute resource drives together into a virtual data store.



### 9.3 Configure Data Stores

To provide your virtual servers with storage space, you need to create [data store zones](#) and [data stores](#). You can [create LVM](#) and [Integrated Storage](#) data stores. You should also [attach data stores to data store zone](#). If you deploy traditional storage, you need to configure data stores on your compute resource. The commands below use `/dev/sda5` as an example. You can find a *volume group identifier* we're using in the second command on the **Data Stores** page in the Control Panel. Follow these steps for each local storage block on a compute resource:

```
# pvcreate --metadatasize=50M /dev/sda5
# vgcreate onapp-ar0akk2wyer3tf /dev/sda5
```



## 9.4 Configure Networks

To provide IP addresses to your future virtual servers, you need to perform the necessary network configurations. Create the [network zone](#) and [network](#). When adding the network, select the network zone you created. The network will be automatically attached to the network zone.



## 9.5 Configure Backup Servers

If you plan to use backup servers to store templates, ISO images, or backups, you need to add the [backup server](#) and [backup server zone](#) to your cloud. Next you add the [backup server to backup server zone](#).



## 9.6 Configure Relations Between Entities

After you added all the necessary resources to your cloud, you need to associate them with the compute resource you created. For this, assign the [data store](#) and [network](#) to the compute resource or compute zone. You also need to assign a backup server to [compute resources](#) or [compute zones](#). All data stores, networks, and backup servers you assign to a compute zone or resource should be from zones with the same type as the compute zone or resource. Zones can have one of the following types: *Virtual*, *Baremetal*, *Smart*, or *VPC*.

- To *Virtual* compute zones and resources, you can assign networks from the Virtual network zones and LVM, Integrated Storage, VMware and SolidFire data stores from the Virtual data store zones.
- To *Baremetal* compute zones and resources, you can assign networks from the Baremetal network zones.

- To *Smart* compute zones and resources, you can assign networks from the Smart network zones and LVM, Integrated Storage, and SolidFire data stores from the Smart data store zones.
- To *VPC* compute zones and resources, you can assign external, org, and vApp networks from the VPC network zones, data stores, and data store zones. All vCloud Director related resources have the *VPC* type.



## 9.7 Configure Templates

To build Linux virtual servers, you need to download templates using the UI downloader. For this, you [install templates](#) and [template store](#). You should also add the installed templates to the Template Store.



## 9.8 Configure ISOs

To be able to build and boot virtual servers from ISO images, additional steps are required. For more information, refer to the [Additional Considerations for ISOs](#) section.



## 9.9 Configure OVAs

To be able to build and boot virtual servers from OVA images, additional steps are required. For more information, refer to the [Additional Considerations for OVAs](#) section.



## 9.10 Configure Buckets

To control user access to resources that are available in your cloud, create and configure [Buckets](#). Buckets enable you to set up resources allocation and pricing separately. If you want to configure resources to which a user has access, you can edit Access Control. If you want to set up pricing for resources, edit Rate Card where the prices and the number of free resources are set.

## 10 Configure VMware Cloud Director Integration

You can integrate OnApp with VMware solutions such as VMware Cloud Director and vCenter. For detailed instructions, see [OnApp and VMware Cloud Director Configuration Guide](#) and [vCenter Implementation Guide](#).

# 11 Support

## 11.1 24/7 Support

OnApp provides 24/7 support that includes remote installation, free upgrades, and 24/7 global support by telephone and email. If you need our help, you can reach to us as follows:

- [Submit a request by email](#)
- **Call (+1) 888 876 8666**

## 11.2 What Does OnApp Support in my Cloud?

OnApp provides support for everything directly related to our core products such as [OnApp Cloud](#), [OnApp CDN](#), [OnApp for VMware](#), and [other products](#). We take responsibility for the software, bug fixes, patches, and general maintenance of our products.

Although there are some areas that are not covered under the standard OnApp support, we attempt to offer suggestions or put you in touch with our Professional Services team. Typically the following areas are not covered by the standard OnApp support:

- Switch, router, and firewall configuration.
- SAN configuration and optimization.
- Attaching, removing, and resizing LUNs.
- The compute resource and Control Panel server hardware support.
- Operating System installation and support.
- Maintenance of your passwords or [Whitelists](#).
- Configuration and troubleshooting inside virtual servers.
- VMware vSphere installation and configuration.
- Known bugs and limitations within virtualization platforms.
- Third party integrations.
- Alpha and Beta releases.
- Coding for [Recipes](#).
- Coding for custom configs on CloudBoot compute resources.

## 11.3 Professional Services

Get in touch with our [Professional Services](#) to get expert help from launch to production and beyond.

## 11.4 OnApp Community

Visit [OnApp Community](#) that is a public forum where you can share your feedback and product ideas. Only OnApp customers can give suggestions but anyone can explore them.

## 11.5 Knowledge Base

Visit [Knowledge Base](#) where you can find a lot of how-to articles to resolve questions that you may face while working with OnApp.

## 11.6 Documentation

You are now located in one of the OnApp documentation spaces that is [Installation Guide](#). Go to [Documentation Home](#) to browse other available spaces.

## 12 Install Xen Compute Resources

In OnApp 7.0, support status of Xen changes from Full Support to Technical Guidance with further End of Support in OnApp 7.1. Refer to [Xen End of Support](#) for more information.

If you need to add a Xen compute resource to your Control Panel, you can follow the procedures described in this guide to set up Xen compute resources.

### 12.1 Install CloudBoot Compute Resources

#### Before you proceed

- The CloudBoot compute resource installation enables dynamic boot of compute resource servers without any persistent installation requirements.
- The servers must support and have PXE boot enabled on the Network Interface Card (set up in BIOS if not already enabled by default). See [Configure CloudBoot Settings in BIOS](#) for details.
- Disable CPU power-saving features in BIOS before you proceed to the compute resource installation.
- We strongly recommend you to deploy one or more backup servers for backups and virtual servers provisioning when using CloudBoot.

Follow this installation to enable CloudBoot for your Xen compute resources.

1. Enable CloudBoot on the Control Panel:
  - Go to **Admin > Settings > Configuration > System > CloudBoot**.
  - Scroll down to the CloudBoot section and check the **Enable** box.
  - Enable Storage on the Control Panel:
    - Go to **Admin > Settings > Configuration > System > OnApp Storage**.
    - Scroll down to the OnApp Storage section and check the **Enable OnApp Storage** box.
    - Tick the **Use Local Read Path** checkbox to minimize the network throughput dependency for read heavy workloads.
    - Enter IP addresses for static content target and Control Panel server CloudBoot interface. Enter the relevant IPs in **Settings > Configuration > System > CloudBoot**. Static content, such as CloudBoot images, kernels, virtual server templates, can be hosted on a standalone NFS server if you want. The default configuration is to install everything on the Control Panel server.
  - Add CloudBoot IP addresses for compute resources:

- Create a CloudBoot IP Net:  
**Admin > Settings > Compute Resources > CloudBoot IPs tab > New IP Net.**
- Add an IP range to the new IP Net:  
**Admin > Settings > Compute Resources > CloudBoot IPs > Actions** icon next to the *IP net* > **Add New IP Range.**
- Add CloudBoot IP Addresses:  
**Admin > Settings > Compute Resources > CloudBoot IPs > Create IP Address.**
- Power on servers and allow them to boot the default image. Add servers to the Control Panel by selecting MAC addresses and assigning IP address via **Admin > Settings > Compute Resources > Add New CloudBoot Compute Resource.**  
If you want to expose drives in compute resources to OnApp Storage, our integrated storage platform, then you must select them at this point. For more information on setting up and configuring CloudBoot, see the [CloudBoot Compute resources](#) section of the Admin guide.
- CloudBoot compute resources mount the following locations automatically at boot:
  - */data* to */onapp/tools/recovery*
  - */onapp/templates* to */.templates*

If you are using CentOS 6, you need to manually create symbolic link */data* to */onapp/tools/recovery* folder adding this command to Custom Config on a compute resource:

- `ln -s /onapp/tools/recovery /data`
- The NFS server from which these are mounted is defined by the **Static Config Target** parameter (see [Edit System Configuration](#) section for details). You can set the default Control Panel server IP to any other server. This change will affect all CloudBoot compute resources.
- The following paths must be available in the static config target to make it possible to use CloudBoot:
  - */ftpboot/export*
  - */data*
  - */ftpboot/images*
- Compute resources will use local templates (mounted from Static Config target) during the server provisioning if the **Use SSH File Transfer** configuration setting is disabled or the template has null *backup\_server\_id*.

If you do not have a Dedicated Backup Server in place, please use Custom Config to mount */onapp/templates* and */onapp/backup* from your Control Panel server or another NFS export.

After you have installed CloudBoot compute resource, proceed to the [Configure CloudBoot Settings in BIOS](#) section.

If you do not have a dedicated backup server, you must mount your template and backup repositories to compute resources. If your template and backup repositories are located on the Control Panel server, you can mount them as follows:

Add locations to `/etc/exports` on the Control Panel server:

```
/onapp/templates 192.168.10.0/24(rw,no_root_squash)
/onapp/backups 192.168.10.0/24(rw,no_root_squash)
```

Add locations to Custom Config on a compute resource and run them manually on the command line (in this example, we are mounting from 192.168.10.101):

```
unlink /onapp/templates
mkdir -p /onapp/backups && mount -t nfs 192.168.10.101:/onapp/backups
/onapp/backups
mkdir -p /onapp/templates && mount -t nfs 192.168.10.101:/onapp/templates
/onapp/templates
```

## Accelerator Deployment

If you want to deploy Accelerator, run the following steps:

1. Run the following command:

```
su onapp
cd /onapp/interface
```

2. Run the following command on the Control Panel server:

- o For all compute resources:

```
rake hypervisor:messaging:configure
```

- o For certain compute resources only:

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

To perform the configuration for a number of compute resources, separate their IP addresses with a space. The command above should be run after every reboot. However, you can avoid the necessity to run the command repeatedly after every reboot by copying the following information (using your parameters) from `/home/mq/onapp/messaging/credentials.yml` to the custom config:

```

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?s81}' #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

```

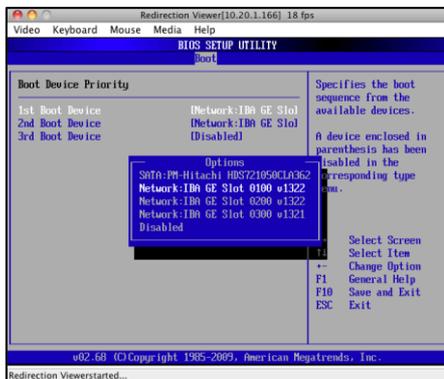
For information on manual configuration for Accelerator, refer to [RabbitMQ Configuration for Accelerator](#).

## 12.1.1 Configure CloudBoot Settings in BIOS

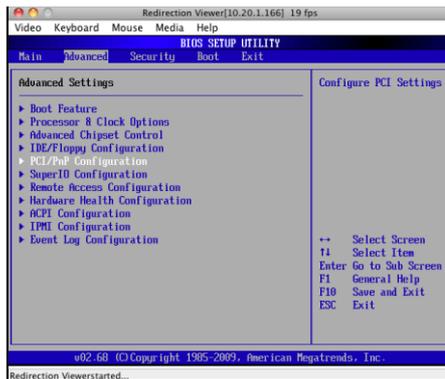
Your BIOS settings may vary from the example provided in this section.

To use PXE boot, you have to make sure it is enabled in BIOS.

1. Select the required Ethernet card supporting PXE as a boot device:



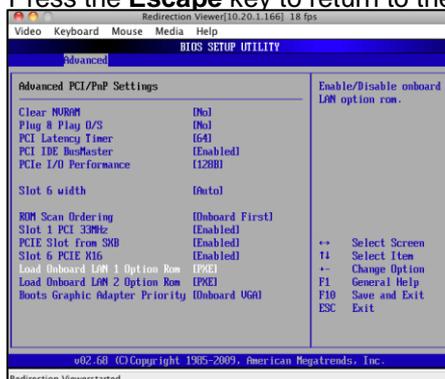
2. Go to the **Advanced settings > PCI/PnP Configuration**.



3. In the **Advanced** settings, select the first/primary Onboard LAN/NIC Option ROM and press **Enter**.

Use up and down arrow keys to set Option ROM settings to enabled and press **Enter**.

Press the **Escape** key to return to the **Advanced** menu.



4. Set a local disk as a second boot device.

## 12.2 Install Static Compute Resources

### Before you proceed

- Install base CentOS packages on the local drive before Xen 4 compute resource installation: CentOS 6.x x86/64 or CentOS 7.x x86/64
- We recommend installing CentOS from the minimal CentOS ISO for static compute resources.
- Disable CPU power-saving features in BIOS before you proceed to the compute resource installation.
- If you are not using a dedicated backup server in your cloud setup, configure NFS server with the following options to preserve files owner and group settings during template unpacking on NFS storage:
- no\_root\_squash

- no\_all\_squash
- Pay attention that smart and baremetal servers cannot be installed using the static compute resource installation method.

If you plan to deploy baremetal servers, you can [enable recovery mode for baremetal servers](#).

To install a compute resource:

1. Add a compute resource to your cloud using the OnApp Control Panel: **Admin > Settings > Compute resources > Add New Compute Resource**. Make sure that the compute resource is visible in the Control Panel and at this point is shown as inactive.
2. Update your server:

```
# yum update
```

3. Download the OnApp repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo-6.2.noarch.rpm
```

4. Install the OnApp compute resource installer package:

```
# yum install onapp-hv-install
```

5. Edit custom configuration for a compute resource. Custom values must be set before the installer script runs.

If you deploy Xen to a server running CentOS 6, it is important to specify a number for `XEN_DOM0_MAX_VCPUS`. We recommend to set the value to 2 if the compute resource has 12 cores or less; or 4 if the compute resource has more than 12 cores.

```
#vi /onapp/onapp-hv.conf
```

### The full list of custom values for Xen ompute resources

- OnApp HV tools [custom version](#):

```
HV_VERSION=""
```

- OnApp StorageAPI [custom version](#):

```
API_VERSION=""
```

- Enable monit - tool for managing and monitoring Unix systems:

```
ENABLE_MONIT=1
```

- Default server to sync time on the compute resource:

```
NTP_TIME_SERVER='pool.ntp.org'
```

- Xen HV (Domain-0) related configuration:

```
XEN_DOM0_MEM_MIN=409600
XEN_DOM0_MEM_DEVISOR=48
XEN_DOM0_MAX_VCPUS=""
XEN_DOM0_VCPUS_PIN_ENABLE=0
XEN_DOM0_SCHEDULER_WEIGHT=65535
XEN_DOM0_SCHEDULER_CAP=200
# 4.2.x and higher versions only
XEN_DOM0_SCHEDULER_RATELIMIT_US=100
XEN_DOM0_SCHEDULER_TIMESLICE_MS=5
```

- The number of loopback devices created:

```
LOOPBACKS=128
```

- The maximum size of the connection tracking table:

The value can't be greater than 65536 if the total memory of Xen Domain-0 or KVM is less than 1Gb.  
The value can be doubled (or even more, depending on the memory amount).

```
NET_IPV4_NETFILTER_IP_CONTRACK_MAX=""
```

- The divisor to calculate the hash table. The recommended value is 8.

```
hashsize = nf_contrack_max / 8
CONTRACK_TO_HASHSIZE=8
```

- Outdated Xen compute resource's (Domain-0) configuration parameters:

```
XEN_DOM0_MEM_OVERHEAD_MIN=262144
P_TO_VCPUS=4
```

6. Run the OnApp compute resource installer script:

```
# /onapp/onapp-hv-install/onapp-hv-xen-install.sh
```

## The full list of Xen installer options

### Usage:

```
/onapp/onapp-hv-install/onapp-hv-xen-install.sh [-c CONFIG_FILE] [-v HV_VERSION] [-p API_VERSION] [-a] [-y] [-t] [-s] [-x] [-h]
```

### Where:

-c CONFIG_FILE	Custom installer configuration file. Otherwise, the pre-installed one is used.
-v HV_VERSION	Custom compute resource Tools version
-p API_VERSION	Custom StorageAPI version
-a	Do NOT be interactive. Process with automatic installation.
-y	Update OS packages (except for OnApp provided) with 'yum update'.
-t	Initiate Recovery templates and ISO(s), which are used to provision FreeBSD guests, download. The download is initiated if the '-a' option is used.
-s	Skip packages management: install, remove, upgrade. None of the packages are installed/updated/removed.
-x	Skip xen*, kernel*, and libvirt* (compute resource related RPM) packages management. Useful for update (not for fresh install).

7. Configure the compute resource for your cloud. This step is also required for the SNMP statistics receiver configuration:

```
# /onapp/onapp-hv-install/onapp-hv-config.sh -h <CP_HOST_IP> -p
<HV_HOST_IP> -b <HV_BSNET_IP>
```

## The full list of compute resource configuration options

### Usage:

```
/onapp/onapp-hv-install/onapp-hv-config.sh [-h CP_HOST_IP] [-p HV_HOST_IP]
[-b HV_BSNET_IP] [-f FTS_IP] [-l LVM_HOST_ID] [-a|-i [USER:PASSWD]] [-s] -
?
```

### Where:

-h CP_HOST_IP	FQDN or IP Addresses (separated by a comma) of the management server, which should receive all status reports and are authoritative for this compute resource. Used by <i>snmpd</i> and <i>snmptrapd</i> .
-p HV_HOST_IP	FQDN or IP Address of server (the compute resource) which will serve all stats-related and other requests send by the CP_HOST_IP. Used by <i>snmpd</i> , <i>snmptrapd</i> , and <i>StorageAPI</i> .
-b HV_BSNET_IP	Compute resource's IP Address from Backup Servers' network. Used to bind the SCSI target daemon.
-f FTS_IP	File Transfer Server FQDN or IP address, used for daily cron update recovery ISO by <i>recovery.sh</i>  <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px; width: fit-content; margin: 10px auto;"> <p>If unsure, set the Control Panel server's management IP as CP_HOST_IP and FILE_TRANSFER_SERVER_IP.</p> </div>
-l LVM_HOST_ID	The <i>lvmlockd</i> <i>sanlock</i> <i>host_id</i> . The ID should be unique for each compute resource in the cloud. It's value for Static compute resources are between 257 and 2000.
-a	Install AoE
-s	Install <i>sshfs</i>
-?	Print this help info

8. Reboot the compute resource to complete the installation:

```
# shutdown -r now
```

9. Generate SSH keys that OnApp requires for you to access various elements of the cloud. The script provided will generate and transfer keys as necessary. The script needs to be run on your Control Panel server. It will overwrite any keys that already exist so if you have custom keys already installed, you will need to add them again after running the script. The script will ask you for your login details to various servers during the execution. Please follow the onscreen instructions.

10. If you install a new cloud instance, connect to your Control Panel server via SSH, download, and run the script:

```
# wget http://downloads.repo.onapp.com/install-all-keys.sh
# /bin/sh install-all-keys.sh
```

11. If you are add additional compute resources to an existing cloud, update the `authorized_keys` file by running the following script on the Control Panel server:

```
# ssh-copy-id -i /home/onapp/.ssh/id_rsa.pub root@HV_HOST_IP
```

12. If you do not have a dedicated backup server, you must mount your template and backup repositories to compute resources. If your template and backup repositories are located on the Control Panel server, you can mount them as follows:

Make sure that locations are added to `/etc/exports` on the Control Panel server and then reboot.

```
/onapp/templates 192.168.10.0/24(rw,no_root_squash)
/onapp/backups 192.168.10.0/24(rw,no_root_squash)
```

Add locations to `/etc/rc.local` on the compute resource and run them manually on the command line (in this example, we are mounting from 192.168.10.101).

```
mkdir -p /onapp/backups && mount -t nfs 192.168.10.101:/onapp/backups
/onapp/backups
mkdir -p /onapp/templates && mount -t nfs 192.168.10.101:/onapp/templates
/onapp/templates
```

13. Mount ISO locations:

To build virtual servers from ISO images, it is required to mount and share the location where the ISOs are stored at the Control Panel server with all the compute resources. When virtual servers are booted from ISOs, the ISO image is taken from the compute resource server. The location is preconfigured in `onapp.yml` config file:

- o `iso_path_on_cp` - specifies the location where ISOs are stored on the Control Panel server. By default, the location is `/data`. You can change it to any other suitable location. Make sure that this location is shared with the specified `iso_path_on_hv` location.
- o `iso_path_on_hv` - specifies the location where ISOs are located on the compute resource servers. By default, the location is `/data`. You can change it to any other suitable location with the `onappowner` user and read/write access. Make sure that this location is mounted to the specified `iso_path_on_cp` location.

CloudBoot compute resources mount the `/data` location automatically at boot to the `/onapp/tools/recovery` on a compute resource.

ISOs can be hosted on a dedicated server at any desired location with an arbitrary name if you want. In this case, it is necessary to mount the ISOs location on this server to the Control Panel `iso_path_on_cp` directory and all the compute resources `iso_path_on_hv` locations. This can be a backup server to avoid the excess usage of the Control Panel resources.

14. Reboot static compute resources.

## Accelerator deployment for static compute resources

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

1. Copy file:

```
cp /home/mq/onapp/messaging/credentials{_example,}.yaml
```

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

```
---
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?s81}' #RABBITMQ ACCESS PASSWORD
queue: 'hv-10.0.50.102' # hv-[IP Address of Compute Resource]
exchange:
  name: 'acceleration'
  type: 'direct'
  durable: True
```

3. Change an owner:

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

4. Run the following commands:

```
service onapp-messaging start
monit monitor onapp-messaging
```

## 13 Configuration and Installer Options

This document provides the full list of configuration and installer options for such entities as Control Panel, compute resources and backups servers.

## 13.1 Control Panel Configuration File Custom Values

**# Template server URL**

```
TEMPLATE_SERVER_URL='http://templates-manager.onapp.com/'
```

**# IPs (separated with coma) list for the snmp to trap**

```
SNMP_TRAP_IPS=
```

**# OnApp Control Panel custom version**

```
ONAPP_VERSION=
```

**# OnApp MySQL/MariaDB connection data (database.yml)**

```
ONAPP_CONN_WAIT_TIMEOUT=15
ONAPP_CONN_POOL=30
ONAPP_CONN_RECONNECT='true'
ONAPP_CONN_ENCODING='utf8'
ONAPP_CONN_SOCKET='/var/lib/mysql/mysql.sock'
```

**# MySQL/MariaDB server configuration data (in case of local server)**

```
MYSQL_WAIT_TIMEOUT=604800
MYSQL_MAX_CONNECTIONS=500
MYSQL_PORT=3306
```

**# Use MariaDB instead of MySQL as OnApp database server**

```
WITH_MARIADB=0
```

**# Configure the database server relative amount of available RAM**

```
TUNE_DB_SERVER=0
```

**# The number of C data structures that can be allocated before triggering the garbage collector. It defaults to 8 million**

```
RUBY_GC_MALLOC_LIMIT=16000000
```

**# sysctl.conf net.core.somaxconn value**

```
NET_CORE_SOMAXCONN=2048
```

# The root of OnApp database dump directory (on the Control Panel box)

```
ONAPP_DB_DUMP_ROOT=
```

# Remote server's (to store database dumps) IP, user, path, openssh connection options and number of dumps to keep

```
DB_DUMP_SERVER=
DB_DUMP_USER=root
DB_DUMP_SERVER_ROOT=/onapp/backups
DB_DUMP_SERVER_SSH_OPT="-o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null -o PasswordAuthentication=no"
KEEP_DUMPS=168
DB_DUMP_CRON='40 * * * *'
```

# Enable monit - tool for managing and monitoring Unix systems

```
ENABLE_MONIT=1
```

# If enabled (the 1 value is set) - install (if local box) and configures RabbitMQ Server (messaging system) for the vCloud support.

```
ENABLE_RABBITMQ=1
```

# Rotate transactions' log files created more than TRANS\_LOGS\_ROTATE\_TIME day(s) ago

```
TRANS_LOGS_ROTATE_TIME=30
```

# Maximum allowed for uploading file size in bytes, from 0 (meaning unlimited) to 2147483647 (2GB). Default is 1GB

```
MAX_UPLOAD_SIZE=1073741824
```

# Timeout before ping Redis Server to check if it is started. Default is 5 sec.

```
REDIS_PING_TIMEOUT=5
```

# OnApp Control Panel SSL certificates (please do not change if you aren't familiar with SSL certificates)

# \* The data below to generate self-signed PEM-encoded X.509 certificate

```
SSL_CERT_COUNTRY_NAME=UK
SSL_CERT_ORGANIZATION_NAME='OnApp Limited'
SSL_CERT_ORGANIZATION_ALUNITNAME='OnApp Cloud'
SSL_CERT_COMMON_NAME=`hostname --fqdn 2>/dev/null`
```

```
# SSLCertificateFile, SSLCertificateKeyFile Apache directives' values
# ssl_certificate, ssl_certificate_key Nginx directives' values
```

```
SSLCERTIFICATEFILE=/etc/pki/tls/certs/ca.crt
SSLCERTIFICATECSRFILE=/etc/pki/tls/private/ca.csr
SSLCERTIFICATEKEYFILE=/etc/pki/tls/private/ca.key
```

```
# * PEM-encoded CA Certificate (if custom one exists)
# SSLCACertificateFile, SSLCertificateChainFile Apache directives' values
# ssl_client_certificate Nginx directives' values
```

```
SSLCACERTIFICATEFILE=
SSLCERTIFICATECHAINFILE=
```

```
# SSLCipherSuite, SSLProtocol Apache directives' values
# ssl_ciphers, ssl_protocols Nginx directives' values
```

```
SSLCIPHERSUITE=
SSLPROTOCOL=
```

## 13.2 The Full List of OnApp Hypervisor Custom Values

- OnApp HV tools custom version

```
HV_VERSION=""
```

- OnApp StorageAPI custom version

```
API_VERSION=""
```

- Default server to sync time on the HV

```
NTP_TIME_SERVER='pool.ntp.org'
```

- Xen HV (Domain-0) related configuration

```
XEN_DOM0_MEM_MIN=409600
XEN_DOM0_MEM_DEVISOR=48
XEN_DOM0_MAX_VCPUS=""
XEN_DOM0_VCPUS_PIN_ENABLE=0
XEN_DOM0_SCHEDULER_WEIGHT=65535
XEN_DOM0_SCHEDULER_CAP=200

4.2.x and higher versions only:
XEN_DOM0_SCHEDULER_RATELIMIT_US=100
XEN_DOM0_SCHEDULER_TIMESLICE_MS=5
```

- The number of loopback devices created

```
LOOPBACKS=128
```

- The maximum size of the connection tracking table.

The value can't be greater than 65536 if the total memory of Xen Domain-0 or KVM is less than 1Gb.  
The value could be doubled (or even more, depends on memory amount).

```
NET_IPV4_NETFILTER_IP_CONTRACK_MAX=""
```

- The divisor to calculate the hash table. The recommended value is 8.

```
hashsize = nf_contrack_max / 8
CONTRACK_TO_HASHSIZE=8
```

- Outdated Xen HVs' (Domain-0) configuration parameters

```
XEN_DOM0_MEM_OVERHEAD_MIN=262144
P_TO_VCPUS=4
```

### 13.3 The Full List of Installer Options for Static Hypervisors

#### Usage:

```
/onapp/onapp-hv-install/onapp-hv-xen-install.sh [-c CONFIG_FILE] [-a] [-y]
[-t] [-s] [-v HV_VERSION] [-p API_VERSION] [-h]
```

#### Where:

-c CONFIG_FILE	custom installer configuration file. Otherwise, preinstalled one is used.
-a	do NOT be interactive. Process with automatic installation.
-v HV_VERSION	custom Hypervisor Tools version
-p API_VERSION	custom StorageAPI version
-y	update OS packages (except for OnApp provided) on the box with 'yum update'.
-t	initiate Recovery templates and ISO(s), which are used to provision FreeBSD guests, download  The download is initiated if '-a' option is used
-s	skip packages management: install, remove, upgrade
-h	print this info

## 13.4 The Full List of Configuration Options for Static Hypervisors

### Usage:

```
/onapp/onapp-hv-install/onapp-hv-config.sh[-h CP_HOST_IP] [-p HV_HOST_IP]
[-b HV_BSNET_IP] [-f FTS_IP] [-a|-i [USER:PASSWD]] [-s] -?
```

### Where:

-h CP_HOST_IP	FQDN or IP Address of the management server which should receive all status reports and is authoritative for this hypervisor. For High Availability cluster, specify the comma-separated IP addresses of the network interfaces that connect the CPs and the hypervisor.
-p HV_HOST_IP	FQDN or IP Address of Server (the Hypervisor) which will serve all stats related and other requests send by the CP_HOST_IP.  Used by snmpd, snmptrapd and StorageAPI.
-b HV_BSNET_IP	Hypervisor's IP Address from Backup Servers' network  Used to bind the SCSI target daemon.
-f FTS_IP	File Transfer Server FQDN or IP address, used for daily cron update recovery ISO by recovery.sh
-a	Install AoE
-s	Install sshfs
-?	Print this help info

## 13.5 Backup Server Default Settings Configuration

### OnApp BK tools custom version

```
BK_VERSION=""
```

### OnApp StorageAPI custom version

```
API_VERSION=""
```

### Default server to synch time on the HV

```
NTP_TIME_SERVER='pool.ntp.org'
```

### The number of retries for WGET to download the file

```
WGET_TRIES=5
```

### OnApp templates directory.

Please refer to the corresponding settings at OnApp Control Panel web interface

```
TEMPLATES_DIR='/onapp/templates'
```

### OnApp backups directory.

Please refer to the corresponding settings at OnApp Control Panel web interface

```
BACKUPSS_DIR='/onapp/backups'
```

## 13.6 The Full List of Installer Options for Static Backup Server

Usage:

```
/onapp/onapp-bk-install/onapp-bk-install.sh [-c CONFIG_FILE] [-a] [-y] [-t] [-h]
```

**Where:**

-c CONFIG_FILE	Custom installer configuration file. Otherwise, preinstalled one is used.
-a	Do NOT be interactive. Processes with automatic installation.
-y	Update OS packages (except of OnApp provided) on the box with 'yum update'.
-t	Initiate Base, Load Balancer and CDN templates download. The download is initiated if '-a' option is used.
-h	Print this info

## 13.7 The Full List of Configuration Options for Static Backup Server

**Usage:**

```
/onapp/onapp-bk-install/onapp-bk-config.sh [-h CP_HOST_IP] [-p BK_HOST_IP] [-f FTS_IP] [-a|-i [USER:PASSWD]] [-s] -?
```

**Where:**

-h CP_HOST_IP	FQDN or IP Address of the management server which should receive all status reports and is authoritative for this backup server.
-p BK_HOST_IP	FQDN or IP Address of Backup Server which will serve all stats related and other requests send by the CP_HOST_IP. Used by snmpd and StorageAPI.
-f FTS_IP	File Transfer Server FQDN or IP address used for daily cron update recovery ISO by recovery.sh
-a	Install AoE
-i [USER:PASSWD]	Install iSCSI utils and configure with USER and PASSWD (if specified)
-s	Install sshfs
-?	Print this help info

## 13.8 The Full List of Installer Options for Static Backup Server Complete Installation

**Usage:**

```
/onapp/onapp-bk-install/onapp-bk-install.sh [-c CONFIG_FILE] [-a] [-y] [-t] [-v BK_VERSION] [-p API_VERSION] [-h]
```

**Where:**

-c CONFIG_FILE	Custom installer configuration file. Otherwise, preinstalled one is used.
-a	Do NOT be interactive. Prozesse with automatic installation.
-v BK_VERSION	Custom Backup Tools version
-p API_VERSION	Custom StorageAPI version
-y	Update OS packages (except of OnApp provided) on the box with 'yum update'.
-t	Initiate Base, Load Balancer and CDN templates download. The download is initiated if '-a' option is used.
-h	Print this info

**13.9 The Full List of Control Panel Installer Options****Usage:**

```
/onapp/onapp-cp-install/onapp-cp-install.sh [-c CONFIG_FILE] [-m MYSQL_HOST] [-p MYSQL_PASSWD] [-d MYSQL_DB] [-u MYSQL_USER] [-U ADMIN_LOGIN] [-P ADMIN_PASSWD] [-F ADMIN_FIRSTNAME] [-L ADMIN_LASTNAME] [-E ADMIN_EMAIL] [-v ONAPP_VERSION] [-i SNMP_TRAP_IPS] [--redis-host=REDIS_HOST] [--redis-passwd[=REDIS_PASSWD] [--redis-port=REDIS_PORT] [--redis-sock=REDIS_PATH] [-a] [-y] [-D] [-h]
```

**Where:**

<b>MYSQL_*</b>	<b>Options are useful if MySQL is already installed and configured.</b>
-m MYSQL_HOST	<b>MySQL host</b>
-p MYSQL_PASSWD	<b>MySQL password</b>
-d MYSQL_DB	<b>OnApp MySQL database name</b>
-u MYSQL_USER	<b>MySQL user</b>
<b>REDIS_*</b>	<b>Options are useful if Redis Server is already installed and configured.</b>
--redis-host=REDIS_HOST	IP address/FQDN where Redis Server runs. The Redis Server will be installed and configured on the current box if localhost/127.0.0.1 or box's public IP

**Usage:**

```
/onapp/onapp-cp-install/onapp-cp-install.sh [-c CONFIG_FILE] [-m
MYSQL_HOST] [-p MYSQL_PASSWD] [-d MYSQL_DB] [-u MYSQL_USER] [-U
ADMIN_LOGIN] [-P ADMIN_PASSWD] [-F ADMIN_FIRSTNAME] [-L ADMIN_LASTNAME] [-
E ADMIN_EMAIL] [-v ONAPP_VERSION] [-i SNMP_TRAP_IPS] [--redis-
host=REDIS_HOST] [--redis-passwd[=REDIS_PASSWD] [--redis-port=REDIS_PORT]
[--redis-sock=REDIS_PATH] [-a] [-y] [-D] [-h]
```

	address (listed in SNMP_TRAP_IPS) is specified. If local Redis, it will serve as well on the unix socket '/tmp/redis.sock'. Default value is 127.0.0.1.
<b>--redis-port=REDIS_PORT</b>	Redis Server listen port. Defaults are: 0 - if local server 6379 - if remote server
<b>--redis-passwd[=REDIS_PASSWD]</b>	Redis Server password to authenticate. Random password is generated if the option's argument isn't specified. By default no password is used for local Redis.
<b>--redis-sock=REDIS_PATH :</b>	Path to the Redis Server's socket. Used if local server only. Default is /tmp/redis.sock
<b>ADMIN_*</b>	Options are used to configure OnApp Control Panel administrator data. Please note, that these options are for NEW INSTALL only and not for upgrade
<b>-P ADMIN_PASSWD</b>	<b>CP administrator password</b>
<b>-F ADMIN_FIRSTNAME</b>	<b>CP administrator first name</b>
<b>-L ADMIN_LASTNAME</b>	<b>CP administrator last name</b>
<b>-E ADMIN_EMAIL</b>	<b>CP administrator e-mail</b>
<b>-v ONAPP_VERSION</b>	<b>Install custom OnApp CP version</b>
<b>-i SNMP_TRAP_IPS</b>	<b>IP addresses separated with coma for snmp to trap</b>
<b>-c CONFIG_FILE</b>	<b>Custom installer configuration file. Otherwise, preinstalled one is used.</b>
<b>--accept-eula</b>	<b>accept End User License Agreement automatically</b>
<b>-y</b>	<b>update OS packages (except of OnApp provided) on the box with 'yum update'.</b>
<b>-a</b>	<b>do not be interactive. Process with automatic installation.</b>

**Usage:**

```

/onapp/onapp-cp-install/onapp-cp-install.sh [-c CONFIG_FILE] [-m
MYSQL_HOST] [-p MYSQL_PASSWD] [-d MYSQL_DB] [-u MYSQL_USER] [-U
ADMIN_LOGIN] [-P ADMIN_PASSWD] [-F ADMIN_FIRSTNAME] [-L ADMIN_LASTNAME] [-
E ADMIN_EMAIL] [-v ONAPP_VERSION] [-i SNMP_TRAP_IPS] [--redis-
host=REDIS_HOST] [--redis-passwd[=REDIS_PASSWD] [--redis-port=REDIS_PORT]
[--redis-sock=REDIS_PATH] [-a] [-y] [-D] [-h]

```

<b>-D</b>	<b>do not make database dump, and make sure it is disabled in the cron and not running at the moment</b>
<b>-h</b>	<b>print this info</b>

## 14 OnApp Installation Components

You can install the following components from command-line:

- [OnApp Database Server](#)
- [RabbitMQ Server](#)
- [Redis Server](#)

### 14.1 OnApp Database Server Installation

- For database server, each table is stored per single file and the server is configured using predefined config (my.cnf) based on total available amount of RAM.
- MariaDB new version 10.4 has been using to serve OnApp's database since version 6.3 Edge 1. An extra `--mariadb-custom` (on CentOS 7.x) and `--mariadb` (on CentOS 6.x) options are required by database installer.

To install MySQL Server standalone (separately from Control Panel box) with the installer:

- OnApp < 6.3:

1. Install OnApp's YUM repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo.noarch.rpm
```

2. Install the `onapp-mysql` package:

```
# yum -y install onapp-mysql
```

3. Run the installer:

- a. For CentOS 6:

```
# /onapp/onapp-mysql/onapp-mysql.sh
```

- b. For CentOS 7:

```
# /onapp/onapp-mysql/onapp-mysql.sh --mariadb
```

- OnApp >= 6.3:

1. Install OnApp's YUM repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo-6.3.noarch.rpm
```

2. Install the `onapp-mysql` package:

```
# yum -y install onapp-mysql
```

### 3. Run the installer:

#### a. For CentOS 6:

```
# /onapp/onapp-mysql/onapp-mysql.sh --mariadb
```

#### b. For CentOS 7:

```
# /onapp/onapp-mysql/onapp-mysql.sh --mariadb-custom
```

### The installer options list:

```
# /onapp/onapp-mysql/onapp-mysql.sh -h
Usage: /onapp/onapp-mysql/onapp-mysql.sh [--mysqlsock MYSQL_SOCKET | --mysqlhost MYSQL_HOST [--mysqlport MYSQL_PORT]] [-mysqluser MYSQL_USER] [-mysqlpasswd MYSQL_PASSWD] [--mariadb | --percona | --percona-cluster [--bootstrap] ] [-h|--help]
```

<code>--mysqlhost MYSQL_HOST</code>	IP address/FQDN where MySQL Server runs. Default values is 127.0.0.1.
<code>--mysqlport MYSQL_PORT</code>	TCP port where MySQL Server serves connections. Default values is 3306.
<code>--mysqlsock MYSQL_SOCKET</code>	Unix socket on which MySQL Server serves connections. Default values is /var/lib/mysql/mysql.sock.
<code>--mysqluser MYSQL_LOGIN</code>	MySQL Server user with administrative privileges. Default value is 'root'.
<code>--mysqlpasswd MYSQL_PASSWD</code>	MySQL Server "root" user's password.
<code>--grantfrom GRANTED_HOSTS</code>	A comma-separated list of hosts' IP addresses which are granted to access the MySQL Server with administrative privileges.
<code>--with-grant- option</code>	Grant access for GRANTED_HOSTS with grant option. Can be used with <code>--grantfrom</code> only.
<code>--installonly</code>	Just install corresponded packages. Do not configure server (except "root" user password) Conflicts with: <code>--bootstrap</code> , <code>--grantfrom</code>
Default is MySQL Server. It is also possible to use one of the following "MySQL compatible" servers:	
<code>--mariadb</code>	MariaDB Server
<code>--percona</code>	Percona Server
<code>--percona-cluster</code>	Percona Cluster
<code>--bootstrap</code>	Percona Cluster specific option to start the server in bootstrap mode

-h --help	Print this info
-----------	-----------------

The following example will install MySQL Server, and set root's random password:

```
# /onapp/onapp-mysql/onapp-mysql.sh
```

*root user's connection properties to the server are at*

```
# cat /onapp/.my.cnf
port=3306
user=root
password=PASSWORD
```

Just run the **OnApp Control Panel's** installer with the following options (MYSQL\_HOST - remote MySQL Server standalone IP address):

```
# /onapp/onapp-cp-install/onapp-cp-install.sh -m MYSQL_HOST -p PASSWORD --
mysql-sock
```

It is also possible to use the remote MySQL Server for the Zabbix Server. To do so, run **Zabbix's** installer with the following options (MYSQL\_HOST - remote MySQL Server standalone IP address):

```
# /onapp/onapp-zabbix/onapp-zabbix.sh --mysqlhost=MYSQL_HOST --
mysqlpasswd='PASSWORD'
```

It is required to grant access to the MySQL Server from outside if it is going to be used remotely: for example by OnApp Control Panel, or Zabbix Server. To do so, the mysql installer should be called with `--grantfrom` option:

```
# /onapp/onapp-mysql/onapp-mysql.sh --grantfrom='CP_HOST,ZABBIX_HOST'
```

where, the CP\_HOST and ZABBIX\_HOST are Control Panel and Zabbix Server IP addresses (separated with comma).

To install other MySQL compatible standalone databases:

### MariaDB Server

- for OnApp < 6.3:

```
# /onapp/onapp-mysql/onapp-mysql.sh --mariadb
```

- ForOnApp >= 6.3:

- CentOS 6.x:

```
# /onapp/onapp-mysql/onapp-mysql.sh --mariadb
```

- CentOS 7.x:

```
# /onapp/onapp-mysql/onapp-mysql.sh --mariadb-custom
```

### Percona Server

```
# /onapp/onapp-mysql/onapp-mysql.sh --percona
```

- 

- **Percona Cluster**

```
# /onapp/onapp-mysql/onapp-mysql.sh --percona-cluster --
grantfrom='WSREP_NODE_INCOMING_IP' --bootstrap
```

where, the `WSREP_NODE_INCOMING_IP` - IP address at which the node accepts client connections. `--bootstrap` is useful when installing the first node of the Percona Cluster.

## 14.2 RabbitMQ Server Installation

To install RabbitMQ Server standalone (separately from Control Panel box) with the installer:

1. Install OnApp's YUM repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo.noarch.rpm
```

2. Install the `onapp-rabbitmq` package:

```
# yum -y install onapp-rabbitmq
```

3. Run the installer:

```
# /onapp/onapp-rabbitmq/onapp-rabbitmq.sh
```

### The installer options list:

```
# /onapp/onapp-rabbitmq/onapp-rabbitmq.sh -h
Usage: /onapp/onapp-rabbitmq/onapp-rabbitmq.sh [--rbthost RBT_HOST] [-
-onapplogin ONAPP_LOGIN] [--onapppasswd ONAPP_PASSWD] [--onappvhost
ONAPP_VHOST] [--rbtlogin RBT_LOGIN] [--rbtpasswd RBT_PASSWD] [-h|--
help]
```

<code>--rbthost RBT_HOST</code>	IP address/FQDN where RabbitMQ Server runs. The RabbitMQ will be installed and configured on the current box if <code>localhost/127.0.0.1</code> or box's public IP address (listed in <code>SNMP_TRAP_IPS</code> ) is specified. Default values is <code>127.0.0.1</code> .
<code>ONAPP_*</code>	Options to configure OnApp's account on RabbitMQ.

--onapplogin ONAPP_LOGIN	The user used by OnApp to connect to the RabbitMQ. Default value is 'onapp'.
--onapppasswd ONAPP_PASSWD	OnApp user's password. The random password is generated if isn't specified.
--onappvhost ONAPP_VHOST	OnApp user's vhost. Default value is '/'
RBT_*	Options are used to configure RabbitMQ manager account. If local RabbitMQ server.
--rbtlogin RBT_LOGIN	abbitMQ manager login. The default value is 'rbtmgr'.
--rbtpasswd RBT_PASSWD	RabbitMQ manager password. The random password is generated if isn't specified.
-h --help	Print this info

The following example will install RabbitMQ Server and configure manager and onapp accounts with random password generated and ready to use by vCloud:

```
# /onapp/onapp-rabbitmq/onapp-rabbitmq.sh
```

*The ready to use credentials.yml could be used to configure OnApp's Control Panel and vCloud access.*

```
# cat /onapp/onapp-rabbitmq/credentials.yml
---
port: 5672
user: 'onapp'
password: 'PASSWORD'
vhost: '/'
host: '127.0.0.1'
```

Just run the CP's installer with the following options (RBT\_HOST - remote RabbitMQ Server standalone IP address):

```
# /onapp/onapp-cp-install/onapp-cp-install.sh --rbthost=RBT_HOST --
vcdlogin=onapp --vcdpasswd='PASSWORD'
```

or copy /onapp/onapp-rabbitmq/credentials.yml from the RabbitMQ Server standalone box, into Control Panel /onapp/configuration/rabbit\_mq/vcloud/credentials.yml.

As soon as the Control Panel installer completes, add 'onapp' system user's RSA public key to authorize root user login on remote RabbitMQ Server via ssh:

```
# ssh-copy-id -i /home/onapp/.ssh/id_rsa root@RBT_HOST
```

The RabbitMQ manager's account credentials are available at the .rabbitmq.mgr file:

```
# cat /onapp/onapp-rabbitmq/.rabbitmq.mgr
rbtmgr MANAGER_PASSWORD
```

Also you can monitor RabbitMQ server using a [command-line tool](#) for it.

## 14.3 Redis Server Installation

To install Redis Server standalone (separately from Control Panel box) with the installer:

1. Install OnApp's YUM repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo.noarch.rpm
```

2. Install the `onapp-redis` package:

```
# yum -y install onapp-redis
```

3. Run the installer:

```
# /onapp/onapp-redis/onapp-redis.sh
```

### The installer options list:

```
# /onapp/onapp-redis/onapp-redis.sh -h
Usage: /onapp/onapp-redis/onapp-redis.sh [--passwd[=PASSWD] [--
nopasswd] [--port=PORT] [--sock=PATH] [-h|--help]
```

<code>--host=HOST</code>	IP address of the interface to bind. Default is 127.0.0.1
<code>--port=PORT</code>	Redis Server listen port. Defaults are:0 - if local server; 6379 - if remote server.
<code>--sock=PATH</code>	Path to the Redis Server's socket. Used if local server only. Default is <code>/tmp/redis.sock</code> .
<code>--nopasswd</code>	Connect to the Redis Server without password
<code>--passwd[=PASSWD]</code>	Redis Server password to authenticate. Random password is generated if the option's argument isn't specified. By default, no password is used for local Redis.
<code>-h --help</code>	Print this info

The following example will install Redis Server, configure access to it from any (including external boxes) and set random password:

```
# /onapp/onapp-redis/onapp-redis.sh --host=REDIS_HOST --passwd
```

where, `REDIS_HOST` - IP address of the interface for Redis to bind. Default is 127.0.0.1.

The ready to use `credentials.yml` could be used to configure OnApp's Control Panel access.

```
# cat /onapp/onapp-redis/credentials.yml
---

host: 'REDIS_HOST'
port: 6379
path: '/var/run/redis/redis.sock'
password: 'PASSWORD'
```

Just run the CP's installer with the following options (REDIS\_HOST - remote Redis Server standalone IP address):

```
# /onapp/onapp-cp-install/onapp-cp-install.sh --redis-host=REDIS_HOST --
redis-passwd='PASSWORD'
```

## 14.4 Zabbix Server Installation

It is not recommended to install Zabbix on the Control Panel server. You can use a separate server or a VS (if your network allows it) as the Zabbix server.

To install Zabbix Server with the installer:

1. Install OnApp's YUM repository:

```
# rpm -Uvh http://rpm.repo.onapp.com/repo/onapp-repo.noarch.rpm
```

2. Install the `onapp-zabbix` package:

```
# yum -y install onapp-zabbix
```

3. Run the installer:

```
# /onapp/onapp-zabbix/onapp-zabbix.sh
```

### The installer options list:

```
# /onapp/onapp-zabbix/onapp-zabbix.sh -h
Usage: /onapp/onapp-zabbix/onapp-zabbix.sh [--mysqlsock MYSQL_SOCKET | -
--mysqlhost MYSQL_HOST [--mysqlport MYSQL_PORT]] [-mysqluser
MYSQL_USER] [-mysqlpasswd MYSQL_PASSWD] [--mysqldb ZABBIX_DB] [--
zabbixhost ZABBIX_HOST] [--zabbixlogin ZABBIX_LOGIN] [--zabbixpasswd
ZABBIX_PASSWD] [--zabbixurl ZABBIX_URL] [-h|--help]
```

MYSQL\_\*

Options are used for MySQL server connection.

--mysqlhost MYSQL_HOST	IP address/FQDN where MySQL Server runs. Default values is 127.0.0.1.
--mysqlport MYSQL_HOST	TCP port where MySQL Server serves connections. Default values is 3306.
--mysqlsock MYSQL_SOCKET	Unix socket on which MySQL Server serves connections. Default values is /var/lib/mysql/mysql.sock.
--mysqluser MYSQL_LOGIN	MySQL Server user with administrative privileges. Default value is 'root'.
--mysqlpasswd MYSQL_PASSWD	MySQL Server "root" user's password.
--mysqldb ZABBIX_DB	MySQL database name for the Zabbix Server. Default value is 'zabbix'
ZABBIX_*	Options are used to configure Zabbix Server and its web UI.
--zabbixlogin ZABBIX_LOGIN	MySQL Server user for the Zabbix Server. The default value is 'zabbix'.
--zabbixpasswd ZABBIX_PASSWD	MySQL Server "zabbix" user's password. The random password is generated if isn't specified.
--zabbixurl ZABBIX_URL	Zabbix Server web UI path (against to VirtualHost). The default value is '/zabbix'.
--zabbixhost ZABBIX_HOST	IP address/FQDN where Zabbix Server runs.
-h --help	Print this info

*The following example will install zabbix Server and configure it to work with ready to use, remote MySQL Server at **MYSQL\_HOST**. If this configuration is not made, Zabbix will set up a separate database on the server on which it is installed.*

```
# /onapp/onapp-zabbix/onapp-zabbix.sh --mysqlhost=MYSQL_HOST --
mysqluser=root --mysqlpasswd='MYSQL_PASSWD'
```

## 14.5 Zabbix Server Uninstall

To uninstall the Zabbix Server from a particular box:

1. Run the following command:

```
# yum remove zabbix zabbix-release zabbix-server zabbix-server-mysql
zabbix-web zabbix-web-mysql onapp-zabbix
```

2. Move/remove "OnApp's for Zabbix" webinterface config and restart Apache:

```
# mv /etc/httpd/conf.d/onapp-zabbix.conf /etc/httpd/conf.d/onapp-
zabbix.conf.rpmsave
# service httpd restart
```

3. *(This step is optional)* To remove Zabbix MySQL user and database, run the following command:

```
# mysql -u root -h <MYSQL_HOST> -e "drop database zabbix;"  
# mysql -u root -h <MYSQL_HOST> -e "drop user 'zabbix'@'<HOST>';"
```

Where:

- <MYSQL\_HOST> - the IP address/FQDN of the host where MySQL server runs
  - <HOST> - the IP address/FQDN of the host, from which the access for *zabbix* user was granted to the *zabbix* database (please find more details at *onapp-zabbix.log*)
4. Move/remove Zabbix webinterface configuration file:

```
mv /etc/zabbix/web/zabbix.conf.php /etc/zabbix/web/zabbix.conf.php.rpmsave
```