OpenNebula 5.8 Enterprise Add-ons Documentation

Release 5.8

OpenNebula Systems

Mar 07, 2019

CONTENTS

CHAPTER

ONE

OPENNEBULA CLI EXTENSIONS

1.1 Overview

CLI extensions is a feature intended to enhance OpenNebula's CLI for customers. This addon will add several sub-commands to existing OpenNebula's CLI commands.

1.1.1 How Should I Read This Chapter

This chapter should be read after the infrastructure is properly setup using OpenNebula's Private Enterprise Repositories.

1.1.2 Hypervisor Compatibility

This chapter applies to all the hypervisors.

1.2 Resolved Issues in 5.8.0

No new features or corrected bugs were found on this release

1.3 Resolved Issues in 5.6.2

No new features or corrected bugs were found on this release

1.4 Installation

You should already have enterprise repositories configured and a working OpenNebula's installation.

1.4.1 Installing on Ubuntu

apt install opennebula-cli-extensions

1.4.2 Installing on CentOS

yum install opennebula-cli-extensions

1.5 onezone serversync

1.5.1 Overview

This command is designed to help administrators to sync OpenNebula's configurations across different federated and High Availability (HA) zones and fix lagging nodes in HA environments. It will first check for inconsistencies between local and remote configuration files inside **/etc/one/** directory. In case they exist, the local version will be replaced by the remote version and only the affected service will be restarted. Whole configuration files will be replaced with the only exception of **/etc/one/oned.conf**. In this case, the local **FEDERATION** configuration will be maintained, but the rest of the content will be overwritten. A backup will be made inside **/etc/one/** before replacing any file.

Note: This addon will connect to your local and remote OpenNebula's Ruby OCA interface. Because of this, OpenNebula should be up and running on both servers. Oneadmin user will be used to log in on each server and credentials will be obtained from ***/var/lib/one/.one/one_auth*** on each server.

This script can also be used to replace your local database with a remote version. Useful if your HA node was left behind.

Warning:

Only use this option between HA nodes, never across federated nodes.

This is the list of files that will be checked and replaced:

- Individual files:
  - /etc/one/az_driver.conf
  - /etc/one/az_driver.default
  - /etc/one/ec2_driver.conf
  - /etc/one/ec2_driver.default
  - /etc/one/econe.conf
  - /etc/one/oneflow-server.conf
  - /etc/one/onegate-server.conf
  - /etc/one/sched.conf
  - /etc/one/sunstone-logos.yaml
  - /etc/one/sunstone-server.conf
  - /etc/one/vcenter_driver.default
- Folders:
  - /etc/one/sunstone-views
  - /etc/one/auth

- /etc/one/ec2query_templates
- /etc/one/hm
- /etc/one/sunstone-views
- /etc/one/vmm_exec

Note: Any file inside previous folders that doesn't exist on the remote server (like backups) will **not** be removed.

1.5.2 Usage

```
$ onezone serversync <remote_opennebula_server> [--db]
```

remote_opennebula_server is the server that will be used to fetch configuration files from.

If **–db** option is used, local database will be synced with the one located on **remote_opennebula_server**.

1.6 onevcenter cleartags

1.6.1 Overview

This command is designed to remove every OpenNebula's custom attribute on a VMware Virtual Machine. When OpenNebula imports a VMware's VM, it adds some attributes. In some operations, like re-importing a VM to OpenNebula, there might be a conflict and errors could appear. These are the attributes that will be removed:

- opennebula.vm.running
- opennebula.vm.id
- opennebula.disk.*

1.6.2 Usage

A use case example for this addon is for re-importing a VMware VM to OpenNebula. The workflow in such scenario would be like this:

- Use onevcenter cleartags on the VM that will be removed:

```
$ onevcenter cleartags <vmid>
```

vmid is the id of the VM whose attributes will be cleared.

- Un-register VM

```
$ onevm recover --delete-db <vmid>
```

- Re-import VM: on the next host's monitoring cycle you will find this VM under **Wilds** tab, and it can be safely imported.

CHAPTER

TWO

OPENNEBULA MARKET ADDON

2.1 Overview

OpenNebula's Market Addon provides new marketplaces to use with OpenNebula. This addon will give access to new appliances and OS images.

2.1.1 How Should I Read This Chapter

This chapter should be read after the infrastructure is properly setup using OpenNebula's Private Enterprise Repositories.

2.1.2 Hypervisor Compatibility

This chapter applies to all the hypervisors.

2.2 Resolved Issues in 5.8.0

No new features or corrected bugs were found on this release

2.3 Installation

You should already have enterprise repositories configured and a working OpenNebula's installation.

2.3.1 Installing on Ubuntu

apt install opennebula-market-addon

2.3.2 Installing on CentOS

yum install opennebula-market-addon

2.4 Turnkey Linux Images Marketplace

2.4.1 Overview

This marketplace enables OpenNebula users to directly fetch images from Turnkey Linux Image Server. Every image listed on that server will be discovered by the marketplace driver and will be shown in the form of **Appliance**_**Distribution**. Amd64 architecture will be used.

2.4.2 Usage

To use this marketplace, you need to install first **OpenNebula Market Addon**, see: *Installation*. Then you will need to modify **/etc/one/oned.conf** and add the following snippet under **MarketPlace Driver Behavior Configuration** section:

oned.conf section:

```
MARKET_MAD_CONF = [
   NAME = "turnkeylinux",
   SUNSTONE_NAME  = "Turnkey_Linux",
   REQUIRED_ATTRS = "",
   APP_ACTIONS = "monitor",
   PUBLIC = "yes"
]
```

Also, you will have to modify the following section on *oned.conf*:

```
MARKET_MAD = [
    EXECUTABLE = "one_market",
    ARGUMENTS  = "-t 15 -m http,s3,one,linuxcontainers"
]
```

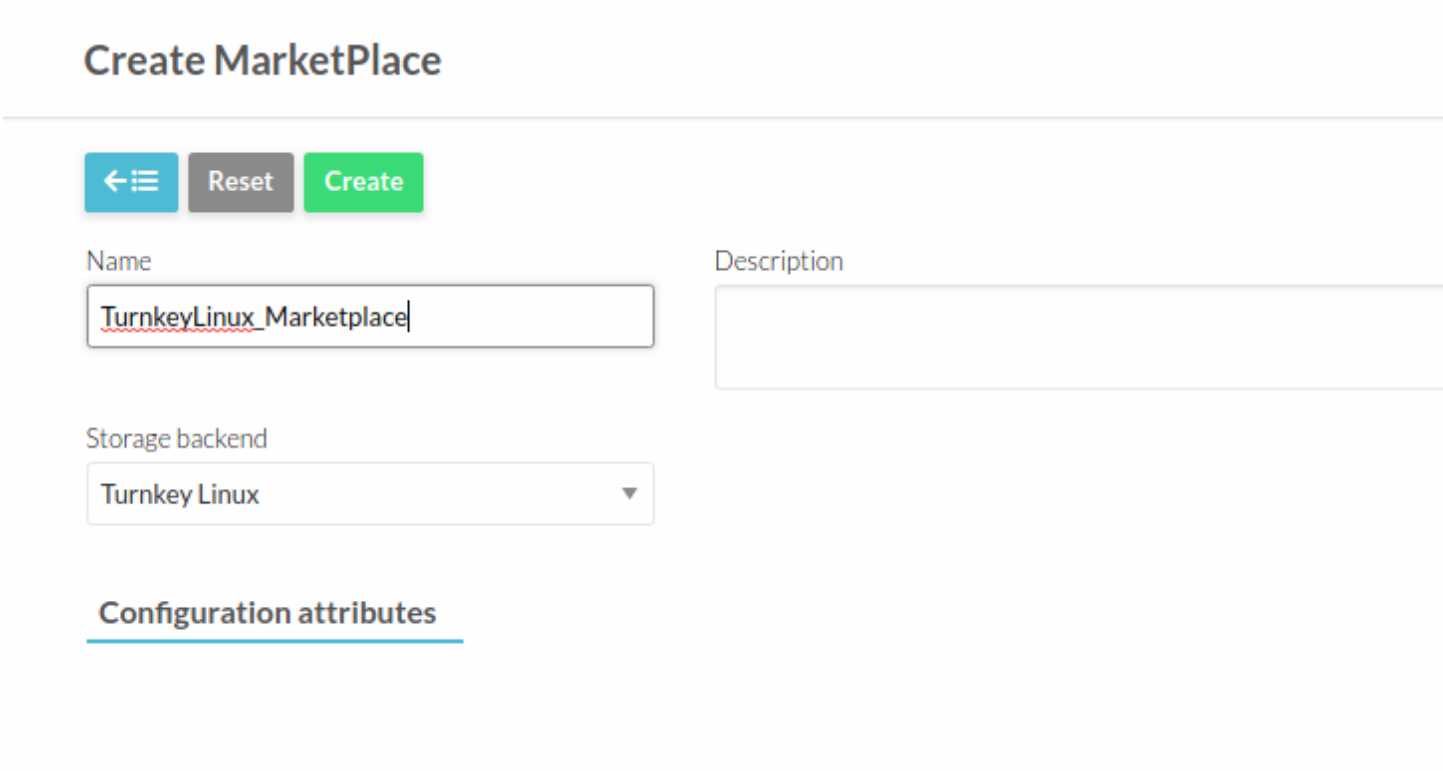
and add **turnkeylinux** as shown below:

```
MARKET_MAD = [
    EXECUTABLE = "one_market",
    ARGUMENTS  = "-t 15 -m http,s3,one,linuxcontainers,turnkeylinux"
]
```

Restart OpenNebula and you are ready to add the new marketplace via Sunstone.

```
# systemctl restart opennebula
```

Open Sunstone and go to **Storage –> MarketPlaces –> Add**. Set a custom Name and select **Turnkey_Linux** as storage backend.


Create MarketPlace
Reset
Create
Name
TurnkeyLinux_Marketplace
Description
Storage backend
Turnkey Linux
Configuration attributes