



OPNFV MOON installation guide

Release draft (51a65a7)

OPNFV

August 12, 2016

CONTENTS

1	Introduction	1
1.1	keystone-moon	1
1.2	keystonemiddleware-moon	1
1.3	python-moonclient	1
2	Packages creation	3
2.1	keystone-moon package	3
2.2	keystonemiddleware-moon package	3
2.3	python-moonclient package	3
3	Installation	5
3.1	Pre-requisite	5
3.2	Installation	5
3.3	Configuration	5
3.4	Running tests	8

INTRODUCTION

The Moon platform is composed of 3 components : * keystone-moon * keystonemiddleware-moon * python-moonclient

1.1 keystone-moon

This component replaces the Keystone component of the OpenStack platform. All basic functions of the original component were maintained but we add some new functions (specially authorization functions)

1.2 keystonemiddleware-moon

This component replaces the KeystoneMiddleware component of the OpenStack platform. The main function added was to intercept all actions from Nova and Swift in order to retrieve an authorization token from the Keystone-moon component.

1.3 python-moonclient

The MoonClient is an interactive script to drive the Keystone-Moon component through the network.

PACKAGES CREATION

Packages can be found on <https://github.com/dthom/moon-bin>

2.1 keystone-moon package

The Keystone-Moon can be package into 2 forms. The first form is in traditional Python package :

```
cd moon_repo/keystone-moon
python setup.py sdist
ls dist
```

We develop a script to build a Debian package, this script is located in *moon_repo/debian/keystone-moon*

```
cd moon_repo/debian/keystone-moon
python create_deb.py
```

2.2 keystonemiddleware-moon package

The KeystoneMiddleware-Moon can be package into 2 forms. The first form is in traditional Python package :

```
cd moon_repo/keystonemiddleware-moon
python setup.py sdist
ls dist
```

We develop a script to build a Debian package, this script is located in *moon_repo/debian/keystonemiddleware-moon*

```
cd moon_repo/debian/keystonemiddleware-moon
python create_deb.py
```

2.3 python-moonclient package

There is only one type of package for the Moon client:

```
cd moon_repo/moonclient
python setup.py sdist
ls dist
```


INSTALLATION

This installation procedure only describe the installation of a standalone Moon platform.

3.1 Pre-requisite

To install the Moon platform, you will need a working Linux server box. The platform is tested on an up-to-date Ubuntu 16.04 box.

You can build your own packages or you can download stable ones on <https://github.com/dthom/moon-bin>

3.2 Installation

First of all, you must install dependencies for the Keystone-moon package, then you can download pre-built packages or create them by yourself. Endly, you can install Keystone-Moon and MoonClient packages:

```
cd /tmp
wget https://github.com/dthom/moon-bin/archive/master.zip
unzip master.zip
PKGS = $(python3 /tmp/moon-bin-master/tools/get_deb_depends.py /tmp/moon-bin-master/*.deb)
sudo apt-get install $PKGS
sudo dpkg -i /tmp/moon-bin-master/keystone_latest-moon_all.deb
sudo pip install --upgrade /tmp/moon-bin-master/python-moonclient-latest.tar.gz
```

At this point, the Nova and Swift components must be installed on the same box or on an other box. See <http://docs.openstack.org/> for more explanation.

Nova and Swift components automatically installed the python-keystonemiddleware package. We have to replace it with the dedicated Moon one:

```
cd /tmp
sudo dpkg -i /tmp/moon-bin-master/python3-keystonemiddleware_latest-moon_all.deb
sudo dpkg -i /tmp/moon-bin-master/python-keystonemiddleware_latest-moon_all.deb
```

Note: if you installed Nova and Swift in 2 different nodes, you must install python-keystonemiddleware in those 2 nodes.

3.3 Configuration

For Keystone, the following files must be configured, some modifications may be needed, specially passwords:

/etc/keystone/keystone-paste.ini

```
sudo cp /etc/keystone/keystone-paste.ini /etc/keystone/keystone-paste.ini.bak
sudo sed "3i[pipeline:moon_pipeline]\npipeline = sizelimit url_normalize request_id build_auth_context" /etc/keystone/keystone-paste.ini
sudo cp /tmp/keystone-paste.ini /etc/keystone/keystone-paste.ini
sudo sed "s/use = egg:Paste#urlmap/use = egg:Paste#urlmap\n\/moon = moon_pipeline/" /etc/keystone/keystone-paste.ini
sudo cp /tmp/keystone-paste.ini /etc/keystone/keystone-paste.ini
```

/etc/keystone/keystone.conf

```
cat << EOF | sudo tee -a /etc/keystone/keystone.conf
[moon]

# Configuration backend driver
configuration_driver = keystone.contrib.moon.backends.memory.ConfigurationConnector

# Tenant backend driver
tenant_driver = keystone.contrib.moon.backends.sql.TenantConnector

# Authorisation backend driver
authz_driver = keystone.contrib.moon.backends.flat.SuperExtensionConnector

# IntraExtension backend driver
intraextension_driver = keystone.contrib.moon.backends.sql.IntraExtensionConnector

# InterExtension backend driver
interextension_driver = keystone.contrib.moon.backends.sql.InterExtensionConnector

# Logs backend driver
log_driver = keystone.contrib.moon.backends.flat.LogConnector

# Local directory where all policies are stored
policy_directory = /etc/keystone/policies

# Local directory where Root IntraExtension configuration is stored
root_policy_directory = policy_root

# URL of the Moon master
master = 'http://localhost:35357/'

# Login of the Moon master
master_login = 'admin'

# Password of the Moon master
master_password = 'nomoresecrete'
EOF
```

The logging system must be configured :

```
sudo mkdir /var/log/moon/
sudo chown keystone /var/log/moon/

sudo addgroup moonlog

sudo chgrp moonlog /var/log/moon/

sudo touch /var/log/moon/keystonemiddleware.log
sudo touch /var/log/moon/system.log

sudo chgrp moonlog /var/log/moon/keystonemiddleware.log
```

```

sudo chgrp moonlog /var/log/moon/system.log
sudo chmod g+rw /var/log/moon
sudo chmod g+rw /var/log/moon/keystonemiddleware.log
sudo chmod g+rw /var/log/moon/system.log

sudo adduser keystone moonlog
sudo adduser swift moonlog
sudo adduser nova moonlog

```

The Keystone database must be updated:

```

sudo /usr/bin/keystone-manage db_sync
sudo /usr/bin/keystone-manage db_sync --extension moon

```

And, Apache must be restarted:

```

sudo systemctl restart apache.service

```

In order to Nova to be able to communicate with Keystone-Moon, you must update the Nova KeystoneMiddleware configuration file. To achieve this, a new filter must be added in */etc/nova/api-paste.ini* and this filter must be added to the composite data. The filter is:

```

[filter:moon]
paste.filter_factory = keystonemiddleware.moon_agent:filter_factory
authz_login=admin
authz_password=password
logfile=/var/log/moon/keystonemiddleware.log

```

Here is some bash lines to insert this into the Nova configuration file:

```

sudo cp /etc/nova/api-paste.ini /etc/nova/api-paste.ini.bak2
sudo sed "/^keystone = / s/keystonecontext/keystonecontext moon/" /etc/nova/api-paste.ini > /tmp/api-
sudo cp /tmp/api-paste.ini /etc/nova/api-paste.ini

echo -e "\n[filter:moon]\npaste.filter_factory = keystonemiddleware.moon_agent:filter_factory\nauthz_

```

Nova can then be restarted:

```

for service in nova-compute nova-api nova-cert nova-conductor nova-consoleauth nova-scheduler ; do
    sudo service ${service} restart
done

```

In order to Swift to be able to communicate with Keystone-Moon, you must update the Swift KeystoneMiddleware configuration file. To achieve this, a new filter must be added in */etc/swift/proxy-server.conf* and this filter must be added to the composite data. The filter is (exactly the same as Nova):

```

[filter:moon]
paste.filter_factory = keystonemiddleware.moon_agent:filter_factory
authz_login=admin
authz_password=password
logfile=/var/log/moon/keystonemiddleware.log

```

Here is some bash lines to insert this into the Nova configuration file:

```

sudo cp /etc/swift/proxy-server.conf /etc/swift/proxy-server.conf.bak2
sudo sed "/^pipeline = / s/proxy-server/moon proxy-server/" /etc/swift/proxy-server.conf > /tmp/proxy-
sudo cp /tmp/proxy-server.conf /etc/swift/proxy-server.conf

echo -e "\n[filter:moon]\npaste.filter_factory = keystonemiddleware.moon_agent:filter_factory\nauthz_

```

Swift can then be restarted:

```
for service in swift-account swift-account-replicator \
    swift-container-replicator swift-object swift-object-updater \
    swift-account-auditor swift-container swift-container-sync \
    swift-object-auditor swift-proxy swift-account-reaper swift-container-auditor \
    swift-container-updater swift-object-replicator ; do
    sudo service ${service} status
done
```

3.4 Running tests

After a successful installation of the Moon platform, you can execute some tests to see if the platform is up and running. Be patient, the latest test takes time (5 to 20 minutes).

```
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_REGION_NAME=What_ever_you_want
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://localhost:5000/v2.0

# See if Nova is up and running:
nova list

# See if Swift is up and running:
swift stat

# See if Keystone-Moon is up and running
moon intraextension list
# you must see one extension (named root)
moon test --self
```

Revision:

Build date: August 12, 2016