OPNFV Installation instructions for the Arno release of OPNFV when using Foreman as a deployment tool

Table of Contents

Abstract

This document describes how to install the Arno SR1 release of OPNFV when using Foreman/Quickstack as a deployment tool covering it's limitations, dependencies and required system resources.

License

Version history

Date	Ver.	Author	Comment
2015-05-07	0.0.1	Tim Rozet (RedHat)	First draft
2015-05-27	0.0.2	Christopher Price (Ericsson AB)	Minor changes & formatting
2015-06-02	0.0.3	Christopher Price (Ericsson AB)	Minor changes & formatting
2015-06-03	0.0.4	Ildiko Vancsa (Ericsson)	Minor changes
2015-09-10	0.2.0	Tim Rozet (Red Hat)	Update to SR1
2015-09-25	0.2.1	Randy Levensalor (CableLabs)	Added CLI verification

Introduction

This document describes the steps to install an OPNFV Arno SR1 reference platform, as defined by the Bootstrap/Getting-Started (BGS) Project using the Foreman/QuickStack installer.

The audience is assumed to have a good background in networking and Linux administration.

Preface

Foreman/QuickStack uses the Foreman Open Source project as a server management tool, which in turn manages and executes Genesis/QuickStack. Genesis/QuickStack consists of layers of Puppet modules that are capable of provisioning the OPNFV Target System (3 controllers, n number of compute nodes).

The Genesis repo contains the necessary tools to get install and deploy an OPNFV target system using Foreman/QuickStack. These tools consist of the Foreman/QuickStack bootable ISO (`arno.2015.2.0.foreman.iso`), and the automatic deployment script (`deploy.sh`).

An OPNFV install requires a "Jumphost" in order to operate. The bootable ISO will allow you to install a customized CentOS 7 release to the Jumphost, which then gives you the required packages needed to run `deploy.sh`. If you already have a Jumphost with CentOS 7 installed, you may choose to ignore the ISO step and instead move directly to cloning the git repository and running `deploy.sh`. In this case, `deploy.sh` will install the necessary packages for you in order to execute.

`deploy.sh` installs Foreman/QuickStack VM server using Vagrant with VirtualBox as its provider. This VM is then used to provision the OPNFV target system (3 controllers, n compute nodes). These nodes can be either virtual or bare metal. This guide contains instructions for installing both.

Setup Requirements

Jumphost Requirements

The Jumphost requirements are outlined below:

1. CentOS 7 (from ISO or self-installed).

2. Root access.

3. libvirt or other hypervisors disabled (no kernel modules loaded).

4. 3-4 NICs for bare metal deployment/only 1 NIC required for virtual deployment, untagged (no 802.1Q tagging), with IP addresses.

5. Internet access for downloading packages, with a default gateway configured.

6. 4 GB of RAM for a bare metal deployment, 18 GB (HA) or 8 GB (non-HA) of RAM for a VM deployment.

Network Requirements

Network requirements include:

1. No DHCP or TFTP server running on networks used by OPNFV (bare metal deployment only).

2. 1, 3, or 4 separate VLANs (untagged) with connectivity between Jumphost and nodes (bare metal deployment only). These make up the admin, private, public and optional storage networks. If only 1 VLAN network used for baremetal, then all of the previously listed logical networks will be consolidated to that single network.

3. Lights out OOB network access from Jumphost with IPMI node enabled (bare metal deployment only).

4. Admin or public network has Internet access, meaning a gateway and DNS availability.

Note: Storage network will be consolidated to the private network if only 3 networks are used.

Bare Metal Node Requirements

Bare metal nodes require:

1. IPMI enabled on OOB interface for power control.

2. BIOS boot priority should be PXE first then local hard disk.

3. BIOS PXE interface should include admin network mentioned above.

Execution Requirements (Bare Metal Only)

In order to execute a deployment, one must gather the following information:

1. IPMI IP addresses for the nodes.
2. IPMI login information for the nodes (user/pass).

3. MAC address of admin interfaces on nodes.

4. MAC address of private interfaces on 3 nodes that will be controllers.

Note: For single NIC/network barmetal deployment, the MAC address of the admin and private interface will be the same.

Installation High-Level Overview - Bare Metal Deployment

The setup presumes that you have 6 bare metal servers and have already setup connectivity on at least 1 or 3 interfaces for all servers via a TOR switch or other network implementation.

The physical TOR switches are **not** automatically configured from the OPNFV reference platform. All the networks involved in the OPNFV infrastructure as well as the provider networks and the private tenant VLANs needs to be manually configured.

The Jumphost can be installed using the bootable ISO. The Jumphost should then be configured with an IP gateway on its admin or public interface and configured with a working DNS server. The Jumphost should also have routable access to the lights out network.

`deploy.sh` is then executed in order to install the Foreman/QuickStack Vagrant VM. `deploy.sh` uses a configuration file with YAML format in order to know how to install and provision the OPNFV target system. The information gathered under section Execution Requirements (Bare Metal Only) is put into this configuration file.

`deploy.sh` brings up a CentOS 7 Vagrant VM, provided by VirtualBox. The VM then executes an Ansible project called Khaleesi in order to install Foreman and QuickStack. Once the Foreman/QuickStack VM is up, Foreman will be configured with the nodes' information. This includes MAC address, IPMI, OpenStack type (controller, compute, OpenDaylight controller) and other information. At this point Khaleesi makes a REST API call to Foreman to instruct it to provision the hardware.

Foreman will then reboot the nodes via IPMI. The nodes should already be set to PXE boot first off the admin interface. Foreman will then allow the nodes to PXE and install CentOS 7 as well as Puppet. Foreman/QuickStack VM server runs a Puppet Master and the nodes query this master to get their appropriate OPNFV configuration. The nodes will then reboot one more time and once back up, will DHCP on their private, public and storage NICs to gain IP addresses. The nodes will now check in via Puppet and start installing OPNFV.

Khaleesi will wait until these nodes are fully provisioned and then return a success or failure based on the outcome of the Puppet application.

Installation High-Level Overview - VM Deployment

The VM nodes deployment operates almost the same way as the bare metal deployment with a few differences. `deploy.sh` still installs Foreman/QuickStack VM the exact same way, however the part of the Khaleesi Ansible playbook which IPMI reboots/PXE boots the servers is ignored. Instead, `deploy.sh` brings up N number more Vagrant VMs (where N is 3 control nodes + n compute). These VMs already come up with CentOS 7 so instead of re-provisioning the entire VM, `deploy.sh` initiates a small Bash script that will signal to Foreman that those nodes are built and install/configure Puppet on them.

To Foreman these nodes look like they have just built and register the same way as bare metal nodes.

VM deployment will automatically use the default gateway interface on the host for all of the VMs internet access via bridging the VMs NICs (public network). The other networks - such as admin, private, storage will all be created as internal VirtualBox networks. Therefore only a single interface on the host is needed for VM deployment.

Installation Guide - Bare Metal Deployment

This section goes step-by-step on how to correctly install and provision the OPNFV target system to bare metal nodes.

Install Bare Metal Jumphost

1. If your Jumphost does not have CentOS 7 already on it, or you would like to do a fresh install, then download the Foreman/QuickStack bootable ISO <http://artifacts.opnfv.org/arno.2015.2.0/foreman/arno.2015.2.0.foreman.iso> here. If you already have a CentOS 7 install that you would like to use then go to step 3.

   2. Boot the ISO off of a USB or other installation media and walk through installing OPNFV CentOS 7.

3. After OS is installed login to your Jumphost as root. If /root/genesis does not exist then `git clone -b arno.2015.2.0 https://gerrit.opnfv.org/gerrit/genesis /root/genesis`

4. Configure IP addresses on 3-4 interfaces that you have selected as your admin, private, public, and storage (optional) networks.

   5. Configure the IP gateway to the Internet either, preferably on the public interface.

   6. Configure your `/etc/resolv.conf` to point to a DNS server (8.8.8.8 is provided by Google).

   7. Disable selinux:

- `setenforce 0`
- `sed -i 's/SELINUX=.*/SELINUX=permissive/' /etc/selinux/config`

8. Disable firewalld:

- `systemctl stop firewalld`
- `systemctl disable firewalld`

Creating an Inventory File

You now need to take the MAC address/IPMI info gathered in section Execution Requirements (Bare Metal Only) and create the YAML inventory (also known as configuration) file for `deploy.sh`.

1. Copy the `opnfv_ksgen_settings.yml` file (for HA) or `opnfv_ksgen_settings_no_HA.yml` from `/root/genesis/foreman/ci/` to another directory and rename it to be what you want Example: `/root/my_ksgen_settings.yml`

2. Edit the file in your favorite editor. There is a lot of information in this file, but you really only need to be concerned with the "nodes:" dictionary.

3. The nodes dictionary contains each bare metal host you want to deploy. You can have 1 or more compute nodes and must have 3 controller nodes (these are already defined for you) if ha_flag is set to true. If ha_flag is set to false, please only define 1 controller node. It is optional at this point to add more compute nodes into the dictionary. You must use a different name, hostname, short_name and dictionary keyname for each node.

4. Once you have decided on your node definitions you now need to modify the MAC address/IPMI info dependent on your hardware. Edit the following values for each node:

- `mac_address`: change to MAC address of that node's admin NIC (defaults to 1st NIC)
- `bmc_ip`: change to IP Address of BMC (out-of-band)/IPMI IP
- `bmc_mac`: same as above, but MAC address
- `bmc_user`: IPMI username
- `bmc_pass`: IPMI password

   5. Also edit the following for only controller nodes:

- `private_mac` - change to MAC address of node's private NIC (default to 2nd NIC)

   6. You may also define a unique domain name by editing the `domain_name` global parameter.

7. Save your changes.

Running `deploy.sh`

You are now ready to deploy OPNFV! `deploy.sh` will use your `/var/opt/opnfv/` directory to store its Vagrant VMs. Your Foreman/QuickStack Vagrant VM will be running out of `/var/opt/opnfv/foreman_vm/`.

It is also recommended that you power off your nodes before running `deploy.sh` If there are DHCP servers or other network services that are on those nodes it may conflict with the installation.

Follow the steps below to execute:

1. `cd /root/genesis/foreman/ci/`

2. `./deploy.sh -base_config /root/my_ksgen_settings.yml`

Note: This is for default detection of at least 3 VLAN/interfaces configured on your jumphost with defaulting interface assignment by the NIC order (1st Admin, 2nd Private, 3rd Public). If you wish to use a single interface for baremetal install, see help output for "-single_baremetal_nic". If you would like to specify the NIC mapping to logical network, see help output for "-admin_nic", "-private_nic", "-public_nic", "-storage_nic".

3. It will take about 20-25 minutes to install Foreman/QuickStack VM. If something goes wrong during this part of the process, it is most likely a problem with the setup of your Jumphost. You will also notice different outputs in your shell. When you see messages that say "TASK:" or "PLAY:" this is Khalessi running and installing Foreman/QuickStack inside of your VM or deploying your nodes. Look for "PLAY [Deploy Nodes]" as a sign that Foreman/QuickStack is finished installing and now your nodes are being rebuilt.

4. Your nodes will take 40-60 minutes to re-install CentOS 7 and install/configure OPNFV. When complete you will see "Finished: SUCCESS"

Verifying the Setup

Now that the installer has finished it is a good idea to check and make sure things are working correctly. To access your Foreman/QuickStack VM:

1. As root: `cd /var/opt/opnfv/foreman_vm/`

2. `vagrant ssh` (no password is required)

3. You are now in the VM and can check the status of Foreman service, etc. For example: `systemctl status foreman`

4. Type "exit" and leave the Vagrant VM. Now execute: `cat /var/opt/opnfv/foreman_vm/opnfv_ksgen_settings.yml | grep foreman_url`

5. This is your Foreman URL on your public interface. You can go to your web browser, `http://<foreman_ip>`, login will be "admin"/"octopus". This way you can look around in Foreman and check that your hosts are in a good state, etc.

6. In Foreman GUI, you can now go to Infrastructure -> Global Parameters. This is a list of all the variables being handed to Puppet for configuring OPNFV. Look for `horizon_public_vip`. This is your IP address to Horizon GUI.

Note: You can find out more about how to use Foreman by going to http://www.theforeman.org/ or by watching a walkthrough video here: https://bluejeans.com/s/89gb/

7. Now go to your web browser and insert the Horizon public VIP. The login will be "admin"/"octopus".

8. You are now able to follow the OpenStack Verification section.

OpenStack Verification

Now that you have Horizon access, let's make sure OpenStack the OPNFV target system are working correctly:

1. In Horizon, click Project -> Compute -> Volumes, Create Volume

2. Make a volume "test_volume" of size 1 GB

3. Now in the left pane, click Compute -> Images, click Create Image

4. Insert a name "cirros", Insert an Image Location `http://download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-disk.img`

5. Select format "QCOW2", select Public, then hit Create Image

6. Now click Project -> Network -> Networks, click Create Network

7. Enter a name "test_network", click Next

8. Enter a subnet name "test_subnet", and enter Network Address `10.0.0.0/24`, click Next

9. Enter `10.0.0.5,10.0.0.9` under Allocation Pools, then hit Create

Note: You may also want to expand this pool by giving a larger range, or you can simply hit Create with entering nothing and the entire subnet range will be used for DHCP

10. Go to Project -> Network -> Routers

11. Click "provider_router". Then select "Add Interface"

12. From the pop up menu, select test_subnet in the "Subnet" field. Press "Add interface"

13. Verify your Network Topology looks correct in Project -> Network -> Network Topology

14. Now go to Project -> Compute -> Instances, click Launch Instance

15. Enter Instance Name "cirros1", select Instance Boot Source "Boot from image", and then select Image Name "cirros"

16. Click Launch, status should show "Spawning" while it is being built

17. You can now repeat steps 15 and 16, but create a "cirros2" named instance

18. Once both instances are up you can see their IP addresses on the Instances page. Click the Instance Name of cirros1.

19. Now click the "Console" tab and login as "cirros"/"cubswin:)"

20. Verify you can ping the IP address of cirros2

21. Continue to the next steps to provide external network access to cirros1.

22. Go to Project -> Compute -> Instances. From the drop down menu under "Actions" select "Associate Floating IP"

23. Press the "+" symbol next under "IP Address". Select "Allocate IP" on the new pop up.

24. You should now see an external IP address filled into the "IP Address" field. Click "Associate".

25. Now from your external network you should be able to ping/ssh to the floating IP address.

Congratulations you have successfully installed OPNFV!

OpenStack CLI Verification

This section is for users who do not have web access or prefer to use command line rather than a web browser to validate the OpenStack installation. Do not run this if you have already completed the OpenStack verification, since this uses the same names.

1. Install the OpenStack CLI tools or log-in to one of the compute or control servers.

2. Find the IP of keystone public VIP. As root:

```
cat /var/opt/opnfv/foreman_vm/opnfv_ksgen_settings.yml | \
grep keystone_public_vip
```

3. Set the environment variables. Substitute the keystone public VIP for <VIP> below.

```
export OS_AUTH_URL=http://<VIP>:5000/v2.0
export OS_TENANT_NAME="admin"
export OS_USERNAME="admin"
export OS_PASSWORD="octopus"
```

4. Load the CirrOS image into glance.

```
glance image-create --copy-from \
http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img \
--disk-format qcow2 --container-format bare --name 'CirrOS'
```

5. Verify the image is downloaded. The status will be "active" when the download completes.

```
glance image-show CirrOS
```

6. Create a private tenant network.

```
neutron net-create test_network
```

7. Verify the network has been created by running the command below.

```
neutron net-show test_network
```

8. Crate a subnet for the tenant network.

```
neutron  subnet-create  test_network  --name  test_subnet  --dns-nameserver
8.8.8.8 10.0.0.0/24
```

9. Verify the subnet was created.

```
neutron subnet-show test_subnet
```

10. Add an interface from the test_subnet to the provider router.

```
neutron router-interface-add provider_router test_subnet
```

11. Verify the interface was added.

```
neutron router-port-list
```

12. Deploy a VM.

```
nova boot --flavor 1 --image CirrOS cirros1
```

13. Wait for the VM to complete booting. This can be completed by viewing the console log until a login prompt appears.

```
nova console-log cirros1
```

14. Get the local ip from the VM.

```
nova show cirros1 | grep test_network
```

15. Get the port ID for the ip from the previous command. Replace <IP> with the IP from the previous command. The port id is the first series of numbers and letters.

```
neutron port-list | grep 10.0.0.2 | awk ' { print $2 } '
```

16. Assign a floating ip to the VM. Substitue the port-id from the previous command for <PORT_ID>

```
neutron floatingip-create --port-id <PORT_ID> provider_network
```

17. Log into the vm. Substitute FLOATING_IP for the floating_ip_address displayed in the output in the above command.

```
ssh cirros@<FLOATING_IP>
```

18. Logout and create a second VM.

```
nova boot --flavor 1 --image CirrOS cirros2
```

19. Get the ip for cirros2.

```
nova show cirros2 | grep test_network
```

20. Redo step 17 to log back into cirros1 and ping cirros2. Replace <CIRROS2> with the ip from the previous step.

```
ping <CIRROS2>
```

Installation Guide - VM Deployment

This section goes step-by-step on how to correctly install and provision the OPNFV target system to VM nodes.

Install Jumphost

Follow the instructions in the Install Bare Metal Jumphost section, except that you only need 1 network interface on the host system with internet connectivity.

Creating an Inventory File

It is optional to create an inventory file for virtual deployments. Since the nodes are virtual you are welcome to use the provided opnfv_ksgen_settings files. You may also elect to customize your deployment. Those options include modifying domain name of your deployment as well as allocating specific resources per node.

Modifying VM resources is necessary for bigger virtual deployments in order to run more nova instances. To modify these resources you can edit each of the follow node paramters in the Inventory file:

1. memory - set in KiB

2. cpus - number of vcpus to allocate to this VM

3. disk - size in GB (cannot be less than 40)

Running `deploy.sh`

You are now ready to deploy OPNFV! `deploy.sh` will use your `/var/opt/opnfv/` directory to store its Vagrant VMs. Your Foreman/QuickStack Vagrant VM will run out of `/var/opt/opnfv/foreman_vm/`. Your compute and subsequent controller nodes will run in:

- `/var/opt/opnfv/compute`
- `/var/opt/opnfv/controller1`
- `/var/opt/opnfv/controller2`
- `/var/opt/opnfv/controller3`

Each VM will be brought up and bridged to your Jumphost NIC for the public network. `deploy.sh` will first bring up your Foreman/QuickStack Vagrant VM and afterwards it will bring up each of the nodes listed above, in order of controllers first.

Follow the steps below to execute:

1. `cd /root/genesis/foreman/ci/`

2. `./deploy.sh -virtual -static_ip_range <your_range>`, Where <your_range> is a range of at least 20 IP addresses (non-HA you need only 5) that are useable on your public subnet. `Ex: -static_ip_range 192.168.1.101,192.168.1.120`

Note: You may also wish to use other options like manually selecting the NIC to be used on your host, etc. Please use ``deploy.sh -h`` to see a full list of options available.

3. It will take about 20-25 minutes to install Foreman/QuickStack VM. If something goes wrong during this part of the process, it is most likely a problem with the setup of your Jumphost. You will also notice different outputs in your shell. When you see messages that say "TASK:" or "PLAY:" this is Khalessi running and installing Foreman/QuickStack inside of your VM or deploying your nodes. When you see "Foreman is up!", that means deploy will now move on to bringing up your other nodes.

4. `deploy.sh` will now bring up your other nodes, look for logging messages like "Starting Vagrant Node <node name>", "<node name> VM is up!" These are indicators of how far along in the process you are. `deploy.sh` will start each Vagrant VM, then run provisioning scripts to inform Foreman they are built and initiate Puppet.

5. The speed at which nodes are provisioned is totally dependent on your Jumphost server specs. When complete you will see "All VMs are UP!"

6. The deploy will then print out the URL for your foreman server as well as the URL to access horizon.

Verifying the Setup - VMs

Follow the instructions in the Verifying the Setup section.

Also, for VM deployment you are able to easily access your nodes by going to `/var/opt/opnfv/<node name>` and then `vagrant ssh` (password is "vagrant"). You can use this to go to a controller and check OpenStack services, OpenDaylight, etc.

OpenStack Verification - VMs

Follow the steps in OpenStack Verification section.

Frequently Asked Questions

Please see the Arno FAQ.

License

All Foreman/QuickStack and "common" entities are protected by the Apache 2.0 License.

References

OPNFV

OPNFV Home Page

OPNFV Genesis project page

OpenStack

OpenStack Juno Release artifacts

OpenStack documentation

OpenDaylight

Upstream OpenDaylight provides a number of packaging and deployment options meant for consumption by downstream projects like OPNFV.

Currently, OPNFV Foreman uses OpenDaylight's Puppet module, which in turn depends on OpenDaylight's RPM hosted on the CentOS Community Build System.

Foreman

Foreman documentation

Authors:	Tim Rozet (trozet@redhat.com)
Version:	0.2.0

Documentation tracking

Revision: 9c9a2ec11f2baebe31a20de16706ae58f303665f

Build date: Wed Sep 30 16:51:59 UTC 2015