



JOID Configuration guide

Release brahmaputra.1.0 (3335efa)

OPNFV

March 16, 2016

CONTENTS

1	JOID Configuration	1
1.1	Bare Metal Installations:	1
1.2	Requirements as per Pharos:	1
1.3	Networking:	1
1.4	Configure JOID for your lab	3
1.5	deployment.yaml file	3
1.6	modify deployment.yaml	3

JOID CONFIGURATION

1.1 Bare Metal Installations:

1.2 Requirements as per Pharos:

1.3 Networking:

Minimum 2 networks

1. First for Admin network with gateway to access external network
2. Second for public network to consume by tenants for floating ips

NOTE: JOID support multiple isolated networks for data as well as storage. Based on your network options for Openstack.

Minimum 6 physical servers

1. Jump host server:

Minimum H/W Spec needed

CPU cores: 16

Memory: 32 GB

Hard Disk: 1(250 GB)

NIC: eth0(Admin, Management), eth1 (external network)

2. Control node servers (minimum 3):

Minimum H/W Spec

CPU cores: 16

Memory: 32 GB

Hard Disk: 1(500 GB)

NIC: eth0(Admin, Management), eth1 (external network)

3. Compute node servers (minimum 2):

Minimum H/W Spec

CPU cores: 16

Memory: 32 GB

Hard Disk: 1(1 TB) this includes the space for ceph as well

NIC: eth0(Admin, Management), eth1 (external network)

NOTE: Above configuration is minimum and for better performance and usage of the Openstack please consider higher spec for each nodes.

Make sure all servers are connected to top of rack switch and configured accordingly. No DHCP server should be up and configured. Only gateway at eth0 and eth1 network should be configure to access the network outside your lab.

1.3.1 Jump node configuration:

1. Install Ubuntu 14.04 LTS server version of OS on the nodes. 2. Install the git and bridge-utils packages on the server and configure minimum two bridges on jump host:

brAdm and brPublic cat /etc/network/interfaces

```
# The loopback network interface
auto lo
iface lo inet loopback
iface eth0 inet manual
auto brAdm
iface brAdm inet static
    address 10.4.1.1
    netmask 255.255.248.0
    network 10.4.0.0
    broadcast 10.4.7.255
    gateway 10.4.0.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 10.4.0.2
    bridge_ports eth0
auto brPublic
iface brPublic inet static
    address 10.2.66.2
    netmask 255.255.255.0
    bridge_ports eth2
```

NOTE: If you choose to use the separate network for management, data and storage then you need to create bridge for each interface. In case of VLAN tags use the appropriate network on jump-host depend upon VLAN ID on the interface.

1.4 Configure JOID for your lab

Get the joid code from gerrit

```
git clone https://gerrit.opnfv.org/gerrit/p/joid.git
```

```
cd joid/ci
```

Enable MAAS

- Create a directory in maas/<company name>/<pod number>/ for example

```
mkdir maas/intel/pod7/
```

- Copy files from pod5 to pod7

```
cp maas/intel/pod5/* maas/intel/pod7/
```

4 files will get copied: deployment.yaml environments.yaml interfaces.host lxc-add-more-interfaces

1.5 deployment.yaml file

1.5.1 Prerequisite:

1. Make sure Jump host node has been configured with bridges on each interface, so that appropriate MAAS and JUJU bootstrap VM can be created. For example if you have three network admin, data and public then I would suggest to give names like brAdm, brData and brPublic. 2. You have information about the node MAC address and power management details (IPMI IP, username, password) of the nodes used for control and compute node.

1.6 modify deployment.yaml

This file has been used to configure your maas and bootstrap node in a VM. Comments in the file are self explanatory and we expect fill up the information according to match lab infrastructure information. Sample deployment.yaml can be found at <https://gerrit.opnfv.org/gerrit/gitweb?p=joid.git;a=blob;f=ci/maas/intel/pod5/deployment.yaml>

1.6.1 modify joid/ci/01-deploybundle.sh

under section case \$3 add the intelpod7 section and make sure you have information provided correctly. Before example consider your network has 192.168.1.0/24 your default network. and eth1 is on public network which will be used to assign the floating ip.

```
'intelpod7' )

# As per your lab vip address list be default uses 10.4.1.11 - 10.4.1.20
sed -i -- 's/10.4.1.1/192.168.1.2/g' ./bundles.yaml
# Choose the external port to go out from gateway to use.

sed -i -- 's/#          "ext-port": "eth1"/          "ext-port": "eth1"/g' ./bundles.yaml
;;
```

NOTE: If you are using separate data network then add this line below also along with other changes. which represents network 10.4.9.0/24 will be used for data network for openstack

```
sed -i -- 's/#os-data-network: 10.4.8.0\21/os-data-network: 10.4.9.0\24/g' ./bun
```

1.6.2 modify joid/ci/02-maasdeploy.sh

under section case \$1 add the intelpod7 section and make sure you have information provided correctly.

```
'intelpod7' )
  cp maas/intel/pod7/deployment.yaml ./deployment.yaml
;;
```

NOTE: If you are using VLAN tags or more network for data and storage then make sure you modify the case \$1 section under Enable vlan interface with maas appropriately. In the example below eth2 has been used as separate data network for tenants in openstack with network 10.4.9.0/24 on compute and control nodes.

```
'intelpod7' )
  maas refresh
  enableautomodebyname eth2 AUTO "10.4.9.0/24" compute || true
  enableautomodebyname eth2 AUTO "10.4.9.0/24" control || true
;;
```

1.6.3 MAAS Install

After integrating the changes as mentioned above run the MAAS install. Suppose you name the integration lab as intelpod7 then run the below commands to start the MAAS deployment.

```
./02-maasdeploy.sh intelpod7
```

This will take approximately 40 minutes to couple hours depending on your environment. This script will do the following:

1. Create 2 VMs (KVM).
2. Install MAAS in one of the VMs.
3. Configure the MAAS to enlist and commission a VM for Juju bootstrap node.
4. Configure the MAAS to enlist and commission bare metal servers.

When it's done, you should be able to view MAAS webpage (<http://<MAAS IP>/MAAS>) and see 1 bootstrap node and bare metal servers in the 'Ready' state on the nodes page.

1.6.4 Virtual deployment

By default, just running the script ./02-maasdeploy.sh will automatically create the KVM VMs on a single machine and configure everything for you.

OPNFV Install

JOID allows you to deploy different combinations of OpenStack release and SDN solution in HA or non-HA mode.

For OpenStack, it supports Juno and Liberty. For SDN, it supports Openvswitch, OpenContrail, OpenDayLight and ONOS.

In addition to HA or non-HA mode, it also supports to deploy the latest from the development tree (tip).

The `deploy.sh` in the `joid/ci` directory will do all the work for you. For example, the following deploy OpenStack Liberty with OpenDayLight in a HA mode in the Intelpod7.

```
./deploy.sh -o liberty -s odl -t ha -l intelpod7 -f none
```

By default, the SDN is Openvswitch, non-HA, Liberty, Intelpod5, OPNFV Brahmaputra release and ODL_L2 for the OPNFV feature.

Possible options for each choice are as follows:

```
[-s ]
nosdn: openvswitch only and no other SDN.
odl: OpenDayLight Lithium version.
opencontrail: OpenContrail SDN.
onos: ONOS framework as SDN.
```

```
[-t ]
nonha: NO HA mode of Openstack.
ha: HA mode of openstack.
tip: the tip of the development.
```

```
[-o ]
juno: OpenStack Juno version.
liberty: OpenStack Liberty version.
```

```
[-l ] etc...
```

default: For virtual deployment where installation will be done on KVM created using `./02-r`

intelpod5: Install on bare metal OPNFV pod5 of Intel lab.

intelpod6

orangepod2

..

(other pods)

Note: if you make changes as per your pod above then please use your pod.

```
[-f ]
none: no special feature will be enabled.
ipv6: ipv6 will be enabled for tenant in openstack.
```

By default debug is enabled in script and error messages will be printed on the SSH terminal where you are running the scripts. It could take an hour to couple hours (max) to complete.

Is the deployment done successfully?

Once juju-deployer is complete, use `juju status` to verify that all deployed unit are in the ready state.

```
juju status --format tabular
```

Find the Openstack-dashboard IP address from the `juju status` output, and see if you can log in via browser. The username and password is admin/openstack.

Optionall, see if you can log in Juju GUI. Juju GUI is on the Juju bootstrap node which is the second VM you define in the `02-maasdeploy.sh`. The username and password is admin/admin.

If you deploy ODL, OpenContrail or ONOS, find the IP address of the web UI and login. Please refer to each SDN bundle.yaml for username/password.

1.6.5 Troubleshoot

To access to any deployed units, `juju ssh` for example to login into nova-compute unit and look for `/var/log/juju/unit-<of interest>` for more info.

```
juju ssh nova-compute/0
```

Example:

```
ubuntu@R4N4B1:~$ juju ssh nova-compute/0
Warning: Permanently added '172.16.50.60' (ECDSA) to the list of known
hosts.
Warning: Permanently added '3-r4n3b1-compute.maas' (ECDSA) to the list of
known hosts.
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-77-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
<skipped>
Last login: Tue Feb 2 21:23:56 2016 from bootstrap.maas
ubuntu@3-R4N3B1-compute:~$ sudo -i
root@3-R4N3B1-compute:~# cd /var/log/juju/
root@3-R4N3B1-compute:/var/log/juju# ls
machine-2.log unit-ceilometer-agent-0.log unit-ceph-osd-0.log
unit-neutron-contrail-0.log unit-nodes-compute-0.log unit-nova-compute-0.log
unit-ntp-0.log
root@3-R4N3B1-compute:/var/log/juju#
```

By default juju will add the Ubuntu user keys for authentication into the deployed server and only ssh access will be available.

Once you resolve the error, go back to the jump host to rerun the charm hook with:

```
juju resolved --retry <unit>
```

Configure OpenStack

In each SDN directory, for example `joid/ci/opencontrail`, there is a folder for Juju deployer where you can find the charm bundle yaml files that the `deploy.sh` uses to deploy.

In the same directory, there is **scripts** folder where you can find shell scripts to help you configure the OpenStack cloud that you just deployed. These scripts are created to help you configure a basic OpenStack Cloud to verify the cloud. For more info on OpenStack Cloud configuration, please refer to the OpenStack Cloud Administrator Guide on docs.openstack.org. Similarly, for complete SDN configuration, please refer to the respective SDN administrator guide.

Each SDN solution requires slightly different setup, please refer to the **README** in each SDN folder. Most likely you will need to modify the **openstack.sh** and **cloud-setup.sh** scripts for the floating IP range, private IP network, and SSH keys. Please go through **openstack.sh**, **glance.sh** and **cloud-setup.sh** and make changes as you see fit.