

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

## Introduction

Bright Cluster-on-Demand for OpenStack is a component of Bright OpenStack. It allows Bright OpenStack users to create virtual Bright clusters inside a Bright OpenStack private cloud. The virtual clusters look and feel the same as a real world physical cluster would. Throughout this article we will refer to Cluster-on-Demand as COD, and where OpenStack is mentioned, this generally refers to Bright OpenStack.

## Prerequisites

Obviously since COD creates clusters in OpenStack, a Bright OpenStack deployment is required. Please refer to our OpenStack documentation on how to deploy Bright OpenStack.

The compute nodes in each virtual cluster will use PXE boot for installation, and this requires tenant networks to support a standard MTU of 1500 bytes. (This is a limitation in iPXE, it does not accept any MTU options set by the DHCP server.) We therefore recommend using VLAN-based tenant network isolation. If VXLAN-based isolation is used, then make sure the underlying network infrastructure is configured for a larger MTU, so that the tenant networks can support an MTU of 1500 bytes.

The head nodes of all virtual clusters will be deployed using an image. This image can be relatively large. (As a minimum it will be about 12 GB.) In order to ensure quick cluster creation it is important to use a storage back-end, which supports copy-on-write (COW) cloning between OpenStack's image (Glance) and volume (Cinder) service. We strongly recommend using Ceph for this purpose. Refer to our manuals on how to set up Ceph and Bright OpenStack.

Finally, COD will create a tenant network for each internal network within the virtual clusters, and a Cinder volume for each (head) node. Each head node will be assigned a floating IP. Therefore, we advise that the quota settings of tenants are reviewed in OpenStack. Specifically, make sure that the tenant can create a sufficient number of vCPUs, instances, networks, subnets, ports, floating IPs, and volumes.

## Setup

First of all, you will have to install the following package on your head node :

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

```
yum install -y cm-cluster-on-demand-openstack
```

```
yum update -y cm-cluster-on-demand
```

You can add the following alias at the end of your headnode `~/.bashrc` :

```
alias cod='cm-cluster-on-demand-openstack'
```

You can also add the following as an alias at the end of the `.bashrc` in `skel` directory for your users at `/etc/skel/.bashrc` :

```
alias cod=cm-cluster-on-demand-openstack
```

By default, COD makes use of a specific set of flavors. To create them, run the following commands as root, from the head node of the clusters:

```
openstack flavor create --ram 1024 --vcpus 1 cod.xsmall
```

```
openstack flavor create --ram 2048 --vcpus 2 cod.small
```

```
openstack flavor create --ram 4096 --vcpus 2 cod.medium
```

```
openstack flavor create --ram 8192 --vcpus 4 cod.large
```

```
openstack flavor create --ram 16384 --vcpus 8 cod.xlarge
```

Obviously, the specific values can be tweaked to suit custom requirements. Note that by default the tools will use `cod.medium` for head node instances and `cod.xsmall` for compute node instances, and that these are the recommended minimum settings for these instances.

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

Next, enable the port security plugin. This is required in order for the head node instances to be able to serve DHCP leases to the compute node instances. Using `cmsh`, execute the following commands:

```
[mycluster]% openstack
[mycluster->openstack[default]]% settings
[mycluster->openstack[default]->settings]% networking
[mycluster->openstack[default]->settings->networking]% set
enableml2portsecurityplugin yes
[mycluster->openstack*[default*]->settings*->networking*]% commit
```

Now generate a base configuration file for the `cm-cluster-on-demand-openstack` tool. To do this make sure the `cluster-tools` environment module is loaded and run:

For Bright 7.3 :

```
cm-cluster-on-demand-openstack config dump >
/etc/cm-cluster-on-demand.conf
```

For Bright 8. 0:

```
cm-cluster-on-demand-openstack config dump --output-type ini > /etc/cm-cluster-on-demand.ini
```

Edit the configuration file and set the following fields:

```
license_product_key=xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx
```

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

Note that this field can be omitted if each user will have his or her own Bright product keys.

```
cluster_password=password123
```

This would set a global default of `password123` as the password for all clusters. Most likely it makes more sense to let users specify their passwords in their own configuration file, or on the command line when creating a cluster.

Specify the image to boot nodes from:

```
node_boot_image = <image_name>
```

Set the default flavors:

```
head_node_type=cod.medium
```

```
default_node_type=cod.xsmall
```

Use `neutron net-show bright-external-flat-externalnet` and take the `id` property:

```
floating_ip_network_uuid=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
```

Finally, a head node image needs to be downloaded from the Bright public image repository, and then pushed into the local OpenStack (Glance) image service.

If the cluster is not connected to the internet, then the image will have to be placed manually as explained in <http://kb.brightcomputing.com/faq/index.php?action=artikel&id=386>

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

## Picking up the image from the internet

If the cluster is connected to the internet, then, in order to list the available images, run the following command as root:

If the cluster is behind a proxy, then please make exopt the `http_proxy` and `https_proxy` before running the following commands.

For Bright 7.3:

```
cm-cluster-on-demand-openstack imagerepo list
```

For Bright 8.0 :

Add the following to `.bashrc` after the line which sources `.openstackrc` :

```
export OS_PROJECT_NAME="bright"
```

Source your `.bashrc`:

```
source ~/.bashrc
```

And then execute the following which will contact Bright computing image repository:

```
cm-cluster-on-demand-openstack image repo-list
```

Pick an image, and using the ID from the list output, run:

Page 5 / 9

(c) 2019 Bright Computing <[kb@brightcomputing.com](mailto:kb@brightcomputing.com)> | 2019-12-11 01:08

URL: <http://kb.brightcomputing.com/faq/index.php?action=artikel&cat=24&id=328&artlang=en>

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

For Bright 7.3 :

```
cm-cluster-on-demand-openstack imagerepo apply --is-public <id>
```

For Bright 8.0:

```
cm-cluster-on-demand-openstack image install --is-public yes <id>
```

Once the image upload is complete, create a regular Bright user with a corresponding OpenStack account. (See other manuals for details.)

For Bright 8.0, see Chapter 5 of the OpenStack Deployment manual:

<http://support.brightcomputing.com/manuals/8.0/openstack-deployment-manual.pdf#chapter.5>

For Bright 7.3, see Chapter 4 of the OpenStack Deployment manual:

<http://support.brightcomputing.com/manuals/7.3/openstack-deployment-manual.pdf#chapter.4>

For the purpose of this document we will be using OpenStack SQL backend.

We will configure OpenStack user setting with the proper initialization and migration scripts found in 5.1.2 of the openstack-deployment-manual.pdf;

## Using COD

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

In case of using Bright 8.0 you will have to create a volume type first :

```
cinder type-create --is-public True default
```

The regular user can now SSH to the cluster, and start creating clusters using the command:

```
cm-cluster-on-demand-openstack cluster create
```

To list the available head node images:

```
cm-cluster-on-demand-openstack image list
```

To list created clusters:

```
cm-cluster-on-demand-openstack cluster list
```

Append "--help" to any COD command to see a full list of available command line arguments.

```
cm-cluster-on-demand-openstack cluster --help
```

```
cm-cluster-on-demand-openstack cluster create --help
```

**Example on creating a CoD cluster :**

Switch to your user :

```
su - username
```

Make sure that ~/.bashrc has the following at the end of it :

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

```
source ~/.openstackrc
source ~/.openstackrc_password
export OS_PROJECT_NAME="USER PROJECT HERE"
```

List the images that you have :

```
cm-cluster-on-demand-openstack image list
```

Use an image to create a cluster

```
cm-cluster-on-demand-openstack create --image <image> -n 2 Test-Cluster
```

## Additional changes needed for virtualizing OpenStack clusters

On the odd chance that one of your use cases for using COD is to run virtualized Bright OpenStack clouds, you might need to introduce some additional configuration changes. Running virtual Openstack clouds is fairly uncommon use case, due to having to deal with nested virtualization.

If you're not planning to run virtual Bright OpenStack clusters, you can skip the steps in this section.

To make sure your users connect to the API endpoints of your virtual OpenStack cluster, and to the nova-novncproxy service, certain ports will have to be open. This can be done by modifying the security groups of an existing cluster. However, if you find yourself creating virtual OpenStack cluster often, it's typically easier to open those ports by default. To do so globally, open the following ports in the `/etc/cm-cluster-on-demand.conf`. To do so only for a specific user, use `~/cm-cluster-on-demand.conf` instead.

# OpenStack: How to setup Cluster-on-Demand (COD) for Bright OpenStack 7.3 and 8.0

```
ingress_ports: [22, 6080, 8081, 10080, 500, 8004, 8774, 8776, 9292, 9696]
```

Port "6080" is required for the Horizon dashboard of the virtual cluster to be able to connect to the console of the VM. The remaining ports (apart from 22, and 10080) are OpenStack API endpoints.

Unique solution ID: #1328

Author: Teun

Last update: 2018-04-18 12:21