

# OpenStack: How do I deploy Bright OpenStack 7.1 (Juno) on an HA cluster?

Deploying a Bright OpenStack 7.1 cluster is explained here. The deployment is described for OpenStack running with or without Ceph storage. For Deployments without Ceph, the Ceph-related paragraphs can be skipped.

## Prerequisites:

- Bright Cluster Manager 7.1 (Bright OpenStack Juno)
- RHEL7.1+ based distro
- One headnode is configured
- Second headnode is not yet configured (no HA yet)
- At least two compute nodes (in this article those will be named "networknode" and "node001").
- Both compute nodes are down (or CMDaemon is stopped on them)
- Shared storage for the headnodes /cm/shared is not configured yet

## End result of following the article:

- Bright OpenStack deployment with two headnodes

---

Let's start.

Install the primary headnode

Deploy a regular cluster according to the Bright Cluster Manager Administrator Manual.

## Prepare HA configuration

Run the "configure" setup step of **cmha-setup** utility

```
cmha-setup -> setup -> configure
```

At the end of this step you will have a new headnode configuration in Bright (needed for OpenStack deployment).

# OpenStack: How do I deploy Bright OpenStack 7.1 (Juno) on an HA cluster?

## [Ceph-only] Prepare for running cm-ceph-setup

*(skip this stage if you won't be using Ceph in your deployment)*

Prepare for running cm-ceph-setup. The Ceph deployment process needs at least one of the nodes which will be later specified as the ceph monitor nodes to be UP during the deployment. This node must have the 'ceph' rpm preinstalled:

Configure the Software image used by the slave nodes which will be later specified as the ceph monitor nodes. Just do "yum install ceph" for the software image of those nodes. There's no need to install this rpm to software images used by OSDs (cm-ceph-setup will do that).

Power on all (or at least one) of the slave nodes which will be later specified as the Ceph monitor nodes, and let them be provisioned with the software image mentioned in the previous step. In the case of the minimum example Ceph deployment, that would be the "networknode". Not all monitors have to be up, so even if you will use multiple monitor nodes later on, powering one at this stage is enough.

Power off the to-be Ceph OSDs nodes (nodes which you will later specify as the OSDs). This is because the to-be OSD nodes cannot be UP when deploying Ceph. Instead of physically powering off the nodes, sometimes it's easier to simply stop the CMDaemon running on those nodes. You can easily do this with pdsh:

```
pdsh -w node001,node002,node003 service cmd stop
```

[Ceph-only] Install Ceph

*(skip this stage if you won't be using Ceph in your deployment)*

This will configure ceph. The Headnode should NOT be used neither as the Ceph OSD nor as the Ceph monitor node. (However, the headnode can be used as the monitor node if needed (but since there must be a odd number of monitor nodes, this means that both headnodes would have to be monitor nodes + a odd number of additional monitor nodes).

A minimum functioning Ceph deployment is composed of two nodes. Since in the case of HA clusters the Ceph nodes cannot be the headnodes, the minimal working ceph-based bright HA deployment is composed of two headnodes, and two slave nodes. One slave node being the Ceph monitor, the other being the Ceph OSD. Note, that with such a minimal deployment data stored on the OSD is not replicated, across other OSDs, so this deployment is only recommended for tests. Note that at least one future monitor node must be up when running cm-ceph-setup and have ceph packages must installed (as described in the previous step).

Run **cm-ceph-setup** to deploy Ceph. The install process will ask which nodes are to be configured as monitor nodes, which are to be configured as OSDs.

Page 2 / 7

# OpenStack: How do I deploy Bright OpenStack 7.1 (Juno) on an HA cluster?

In the case of minimum example deployment discussed in this document the user should configure “networknode” as the monitor node, and “node001” as the OSD node.

After running `cm-ceph-setup`, power on the nodes which participate in the Ceph cluster. This will effectively enable Ceph. When deploying OpenStack with Ceph backends, a functioning Ceph cluster will be required for the OpenStack deployment process to work. That's why we need to make sure right now that Ceph is working fine.

In the case of minimum example deployment discussed in this document, the user would have to power on node001 (node which had the OSD role).

At this point, “`ceph -s`” should report the monitor and OSD nodes to be up. Also the Ceph overview in `cmsh` and `cmgui` should be displaying Ceph-related information obtained from Ceph.

## Deploy OpenStack

Deploy OpenStack on your primary headnode (using CMGUI wizard, or `cm-openstack-setup` script). Simply follow the regular OpenStack deployment process as you would have for a single headnode cluster..

In the case of a HA cluster, this step will silently depend on the fact that you already have the second Headnode object configuration created in bright, as it will assign the default roles to both of the headnodes (You've done that during the `cmha-setup 'configure'` step). What this means is that all the roles which would normally be assigned to the single headnode of your cluster, will be automatically assigned to both headnodes.

In other words, after deploying OpenStack, at this point your API endpoints, HAProxy node, AMQP server, will be configured on your primary headnode.

After deploying OpenStack, this would be the point to decide whether you want to stick with AMQP server, and/or HAProxy nodes running on the headnodes. Or whether in the case your deployment you would prefer to run those on a separate set of nodes.

When running AMQP server (RabbitMQ) on the headnodes, the service will run on the currently active headnode (in a un-clustered mode).

If you want to run `rabbitmq` in a clustered mode, you will need to configure it manually, and plug it in later to the OpenStack deployment by:

```
[headnode->openstack[default]->settings:advanced]% set customamqphost amqpcluster;
commit
```

## Disable OpenStack, stop OpenStack services

Disable OpenStack:

```
cmsh -c "openstack; set enabled 0; commit"
```

Page 3 / 7

# OpenStack: How do I deploy Bright OpenStack 7.1 (Juno) on an HA cluster?

Stop all OpenStack services running on the headnode. This can be easily done with CMGUI -- select multiple services, and stop them

## Prepare the Secondary Headnode

If you're deploying a virtualized Bright HA cluster, e.g. inside of a OpenStack Cloud, create your secondary headnode instance via dashboard, on the cluster's internal network and external network, boot it with an iPXE image PXE booting off of eth0. Quickly go to the node's console, and boot the node into the rescue mode.

Make it wait there for now.

If you're deploying a physical cluster, simply boot your new secondary headnode off of the primary headnode, and enter the rescue mode.

## Clone the Primary Headnode to Secondary Headnode

Run `cmha-setup -> setup -> clone install`, and follow the on screen instruction. The instructions bold down to going to the console of you secondary headnode, logging in, and the running:

```
/cm/cm-clone-install --failover
```

This will clone the current content of your headnode (with OpenStack already installed in the previous step), to your secondary headnode.

Decide where you want to store OpenStack databases

By default OpenStack Databases will be stored in the MariaDB database services running on the Headnodes, and thus will be automatically replicated between those two for High Availability.

Advanced users might want to consider deploying their own Galera database cluster (as described in the OpenStack manual). If that's what you will want to do, remove the file

`/cm/local/apps/cluster-tools/ha/conf/extradbclone_openstack.xml`. That file has been created as part of OpenStack deployment process, and in the next step will result in the headnodes being configured to replicate OpenStack databases between each other. If you will want to run your own Galera cluster, replicating OpenStack databases between the headnode will not be needed, you should thus remove this file before proceeding.

## Finalize HA setup

Run:

```
cmha-setup -> setup -> finalize
```

# OpenStack: How do I deploy Bright OpenStack 7.1 (Juno) on an HA cluster?

This will clone the databases to the failover node. Note, OpenStack service must still be down here, otherwise they will proceed writing to the databases while those are being cloned.

## Configure AMQP on the secondary headnode

When running AMQP servers on the headnode, the secondary RabbitMQ configuration on the secondary headnode should be cleaned up.

SSH to the secondary headnode.

Make sure that rabbitmq-server is stopped on the secondary headnode.

Remove the rabbitmq database, it's a leftover of the cloning process of the primary headnode to the secondary headnode

```
rm -rf /var/lib/rabbitmq/mnesia/*
```

Find out rabbitmq's 'openstack' username and password:

```
cmsh -c 'openstack settingscredentials; get messagequeueusername; get messagequeuepassword'
openstack #username
y0cSR8D2aDtiDJKI2Dh59VAr #password
```

Start 'rabbitmq-server' on the secondary headnode

Add the rabbitmq's 'openstack' user with the obtained credentials

```
rabbitmqctl add_user <username> <password>
```

Now set the permissions (again, on the secondary headnode)

```
rabbitmqctl set_permissions -p / <username> '.*' '.*' '.*'
```

## Configure Shared Storage

The last thing to do is to configure the shared storage for the headnodes. i.e. the storage location the /cm/shared and /home will be stored.

If you are deploying this cluster in a virtual environment, you could simply create a VM with a NFS export, and later configure it as NAS.

Run **cmha-setup** , and select "Shared Storage". Follow the steps as described on screen. Administrator Manual can be used to learn more information on this step.

# OpenStack: How do I deploy Bright OpenStack 7.1 (Juno) on an HA cluster?

The end result of this step is that /cm/shared located on your primary headnode will be copied to an external share at this point. Deploying OPenStack can potentially modify the content of /cm/shared, that's why it's important to deploy it OpenStack before configuring Shared Storage.

*[advanced users] If OpenStack needs to be deployed after Shared storage has been configure, the Shared Storage step must be repeated. (Also, make sure no /cm/shared is externally mounted over the local filesystem copy of /cm/shared. This will be the case with already existing nas NAS)*

## Enable OpenStack, start OpenStack services

```
cmsh -c "openstack; set enabled 1; commit"
```

## Start OpenStack services

Testing:

- use the dashboard to create a network
- trigger manual failover event (run "cmha makeactive" on the passive headnode)
- log in to the dashboard to see if the network is still there
- use cmsh/cmgui to create a object in OpenStack (e.g. Network)
- trigger manual failover (this time run cmha makeactive on the other headnode)
- check if the object is still there

## Optional next steps:

- Depending on you network config, you might need to force a specific hostname to be your main public Auth Host. You can do this with "OpenStackKeystoneMainPublicAuthHost=master" configured in Advanced Config section of cmd.conf on your headnodes
- If you are not using Ceph for OpenStack storage, you will want to configure your storage driver at this point.
- If you want to run OpenStack::Dashboard roles on your headnodes (or simply multiple instances of the dashboard hosts), you might want to add a Memcached node, and reconfigure the Dashboard roles to use that memcached cost as the backend for storing sessions (so that in the event of failover users won't have to sign in again). If you want to provider HA for the memcached node, configure multiple of them, in active/passive mode, and put them behind a HAProxy (or behind a cluster of HAProxies).
- Do some more performance/usability test and:
  - decide if you want to stay with RabbitMQ in active/passive running on the headnodes, or whether you want to configure your own active/active Highly available cluster of clustered RabbitMQ nodes
  - decide how many Backend HAProxie do you need. By default the HAProxie on the headnode will be the backend proxies use by OpenStack service to contact other OpenStack services. In some deployments it might make sense to have a separate set of backend HAProxies, to offload the headnodes.
  - decide if you want to users to access the deployment via frontend HAProxies running on the headnode, or maybe via HAProxies located on some other nodes. If the latter seems better for your case, you can try configuring a "Failover node group".

Unique solution ID: #1272

# OpenStack: How do I deploy Bright OpenStack 7.1 (Juno) on an HA cluster?

Author: Piotr Wachowicz

Last update: 2015-07-13 15:56