A Reproducible and Automated Deployment of an HPC Application on a Private Cloud

Nam Pho and Josh Miller

OSDC PIRE
June 11, 2015

## Abstract

The challenge presented to this hackathon was to address gaps within the OSDC ecosystem by exploring tools and applications with ease-of-use and reproducibility being paramount to whatever solution is proposed. We aim to address all of these areas with an automated deployment of a high-performance computing (HPC) cluster on a private cloud. In this proof-of-concept we are using the OpenScienceDataCloud (OSDC) deployment of OpenStack. Since the deployment of an HPC cluster is automated, it can be easily adopted by researchers with less experience and lower the barrier for exploring the OSDC data while minimizing any inefficiency in use of the cluster. Since the platform is consistently deployed it also serves to be a product or application that can be delivered to the researchers as a tool for analyzing data.

GitHub: https://github.com/nampho2/osdc-ansible

## Methods

The configuration management of all virtual machines was handled by Ansible. Ansible is a Python-based tool to manage a collection of servers.

Hardware was abstracted through the use of an OpenStack deployment on the OSDC (Sullivan) with an allocation of 16 vCPU, 32 Gb of RAM, and a 1 Tb uniformly mounted GlusterFS cluster.

The software stack uses the CentOS 6 image provided by the OSDC for all systems and is running the openlava job scheduler. Openlava is an open-source deployment that closely resembles IBM's Platform LSF. Scientific software packages come from the XSEDE repository (Fisher, *et al*. 2014).

## Results

All the Ansible playbooks and code are freely available through a public GitHub repository and should be seen as a continuous work in progress. To use the tool, first we confirm that there are no running VMs.

```
npho@kg14-compute-1:~/osdc-ansible$ nova list
npho@kg14-compute-1:~/osdc-ansible$
```

If any VMs have conflicting names with the machines you're about to spin up you can run `ansible-playbook purge.yml` from the repository to remove any

redundancies. Now we can spin up the cluster from scratch by running the `hpc.yml` playbook as shown below.

**NOTE: The process is long and extensive sections of output have been truncated for this submission.**

```
npho@kg14-compute-1:~/osdc-ansible$ ansible-playbook hpc.yml
 ----------------------------------------
< PLAY [setup a hpc cluster in openstack] >
 ----------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||

<<< output truncated >>>
 ---------------------------------------
< TASK: provision | create nova key-pair >
 ---------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||


ok: [loginnode]
 ---------------------------------
< TASK: provision | create all vms >
 ---------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||


ok: [loginnode] => (item=hn1)
ok: [loginnode] => (item=cn1)
ok: [loginnode] => (item=cn2)
ok: [loginnode] => (item=cn3)
ok: [loginnode] => (item=cn4)
ok: [loginnode] => (item=cn5)
ok: [loginnode] => (item=cn6)
ok: [loginnode] => (item=cn7)


<<< output truncated >>>

npho@kg14-compute-1:~/osdc-ansible$
```

After the process is complete you now have a fully configured HPC cluster. This is roughly as diagrammed in Figure 1.

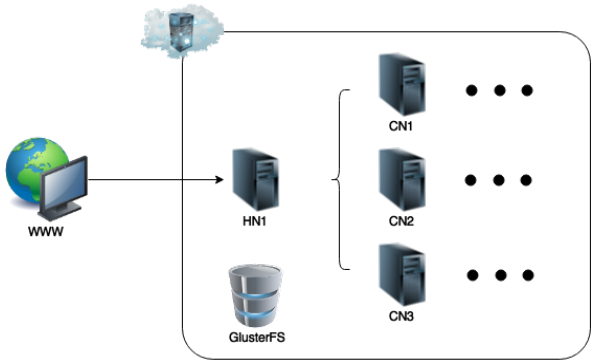


**Figure 1:** layout of the openlava virtual cluster deployment in OSDC.

You can confirm that the VMs are online and active as show below.

```
npho@kg14-compute-1:~/osdc-ansible$ nova list
+--------------------------------------+------+--------+----------------------+
|                  ID                  | Name | Status |       Networks       |
+--------------------------------------+------+--------+----------------------+
| 06f813ee-f13b-4ebd-bf76-795032f9784d | cn6  | ACTIVE | private=172.16.1.180 |
| 10c6db8c-d977-47eb-a2d9-b355866d0029 | hn1  | ACTIVE | private=172.16.1.152 |
| 4cabf3c9-f27a-423c-b653-b4c6cad24dd2 | cn7  | ACTIVE | private=172.16.1.178 |
| 4dae0b88-34ab-454c-a4a2-15620471c461 | cn3  | ACTIVE | private=172.16.1.160 |
| 6b983d4a-7a61-4141-9c94-1fb80b2939ae | cn2  | ACTIVE | private=172.16.1.97  |
| 740360cc-af5d-482b-96fc-fed28a9c358b | cn4  | ACTIVE | private=172.16.1.182 |
| a10cfcb1-a7d9-4183-a9f6-6bb240b4a9d6 | cn5  | ACTIVE | private=172.16.1.175 |
| b2c3cf37-8f5b-44b1-967c-b3e0976cf431 | cn1  | ACTIVE | private=172.16.1.100 |
+--------------------------------------+------+--------+----------------------+
npho@kg14-compute-1:~/osdc-ansible$
```

The default layout has a head node server called `hn1` and seven compute nodes labeled `cn1` through `cn7`. You can log into the head node as shown below and confirm that all compute nodes are visible with their 2 CPUs.

```
npho@kg14-compute-1:~$ ssh root@hn1
Last login: Thu Jun 11 10:40:26 2015 from loginnode
[root@hn1 ~]# bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
cn1                ok              -      2      0      0      0      0      0
cn2                ok              -      2      0      0      0      0      0
cn3                ok              -      2      0      0      0      0      0
cn4                ok              -      2      0      0      0      0      0
cn5                ok              -      2      0      0      0      0      0
cn6                ok              -      2      0      0      0      0      0
cn7                ok              -      2      0      0      0      0      0
hn1                closed          -      0      0      0      0      0      0
[root@hn1 ~]#
```

Now you can run parallel jobs from the head node by prepending your command with the openlava executable, `bsub`. Make sure you write your output to a shared system (or collect from all the machines independently if you write to local disk).

# References

Jeremy Fischer, Richard Kneppe, Matthew Standish, Craig A. Stewart, Resa Alvord, David Lifka, Barbara Hallock, Victor Hazlewood. Methods For Creating XSEDE Compatible Clusters. *XSEDE '14* [PDF]