ILLINOIS DATA SCIENCE INITIATIVE

SETTING UP AMBARI ON OPENSTACK NEBULA

Version: 0.0.1

Author(s): Quinn Jarrell, Professor Brunner

April 11, 2017

Setting up Ambari on Openstack Nebula

QUINN JARRELL[1,2,3] AND PROFESSOR BRUNNER[1,2,3]

[1] *National Center For Supercomputing Applications (NCSA)*
[2] *Laboratory for Computation, Data, and Machine Learning*
[3] *Univeristy of Illinois*

Compiled April 11, 2017

https://github.com/lcdm-uiuc

INTRODUCTION

This technical report details the advantages, disadvantages, and a guide on setting up Ambari on Openstack. We refer to individual virtual machines as nodes. The included scripts used within this report are all included in the scripts directory under the name openstack_setup.py.

ABOUT OPENSTACK

The Openstack software enables many private organizations to setup private clouds running on their own datacenters. Openstack presents a management system for virtual computers with similar features to public clouds like Amazon Web Services and Google Cloud Platform. The National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign (UIUC) runs an extensive deployment of Openstack. Many research groups within UIUC depend on Openstack to provide a reliable infrastructure to deploy distributed programs.

ABOUT AMBARI

Some of the most desirable distributed applications deployed on NCSA are Apache Hadoop and Apache Spark. Apache Ambari provides a managed system of deploying Apache Hadoop, Spark, and several other distributed applications to a cluster of computers on Openstack. It provides a Web interface which provides not only a method installing new services but also an extensive monitoring system to maintain distributed applications. It has enabled the Illinois Data Science Initiative (iDSI) to quickly set up Hadoop and Spark clusters required for teaching an Applied Cloud Computing class and facilitating IDSi's research.

VIRTUAL MACHINE SIZES

We have found deploying Hadoop/Spark clusters to very few large machines works much better than deploying to lots of medium sized virtual machines. The goal is always to avoid network communication costs and to have as much RAM as possible. Our clusters worked very well with 64 GB of ram and 24 VCPUs. There is an extensive guide on sizing your cluster from Hortonworks. We rely on network drives to provide our storage which does hamper the performance of our cluster but is provision from Openstack. Typically each node has two volumes of 1 TB each.

SETTING UP THE VIRTUAL NETWORK ON OPENSTACK

We first need to configure Openstack to allocate all virtual machines for our cluster in a private network. Our network was limited to 254 nodes so we used the settings:

```
Network Address : 192.168.0.0/24
Gateway IP: 192.168.0.1
Allocation Pools: 192.168.0.2,192.168.0.254
```

CREATING A SECURE AMBARI SECURITY GROUP

Ambari should be only accessible through the web interface. Openstack provides a set of connection blocking rules called Security Groups. The Security Groups are listed under Compute/Access_and_Security in the Openstack Dashboard. To secure Ambari a new security group should be created and assigned to all nodes launched. We used the following rules for our security group:

```
Egress  IPv4  Any Any 0.0.0.0/0
Egress  IPv6  Any Any ::/0
Ingress IPv4  ICMP  Any 192.168.100.0/24
Ingress IPv4  TCP 1 - 65535 192.168.100.0/24
Ingress IPv4  TCP 22 (SSH)  0.0.0.0/0
Ingress IPv4  TCP 8080  0.0.0.0/0
```

This restricts external access to any node only through port 8080 or SSH. Port 8080 is the default Ambari web interface port. This prevents exposing PostgreSQL databases and other services to the open internet.

SETTING UP THE NETWORK FOR CENTOS 6

Ambari requires each node in the cluster to know every other node's domain name. There are two ways of doing this, either through a Domain Name Server(DNS) or through editing a file called hosts. The hosts file is a slightly easier but more rigid method as it only works well with a small private network (Less than 10000 hosts). To create the hosts file we generate a mapping like

192.168.0.0 192-168-0-0.local

With dots replaced with dashes, it is an acceptable domain name and the .local is an unused suffix which can be changed to anything. We translate every address within the local subnet. The 192.168.x.x is dedicated to a local area network so it will not affect requests to other IPs. It should be noted that this requires an entirely private network. If you are running on a shared network, you should only map the private IP addresses of the nodes on your cluster and check that they do not change.

Hosts File Example

So our hosts file ends up looking this:

```
127.0.0.1   localhost localhost.localdomain
127.0.0.1   localhost4 localhost4.localdomain4
192.168.100.0 192-168-100-0.local
192.168.100.1 192-168-100-1.local
192.168.100.2 192-168-100-2.local
192.168.100.3 192-168-100-3.local
192.168.100.4 192-168-100-4.local
192.168.100.5 192-168-100-5.local
192.168.100.6 192-168-100-6.local
192.168.100.7 192-168-100-7.local
192.168.100.8 192-168-100-8.local
192.168.100.9 192-168-100-9.local
::1         localhost localhost.localdomain
::1         localhost6 localhost6.localdomain6
```

With lots of more similar lines for the rest of the local subnet.

PREPARING SERVERS

We use the tool Fabric to provision servers. We assume you have installed Fabric on one of the computers in the cluster and configured it.

From here you can run the following lines to have Ambari fully installed on a new Centos 6 node

```
fab -f setup-fqdn.py \
install_packages copy_hosts_file fix_fqdn \
disable_huge_pages disable_firewall turn_on_ntp \
disable_selinux import_ambari_repo \
-H $NODE1,$NODE2,$NODE3

fab -f setup-fqdn.py \
setup_ambari_server \
-H $MASTERNODE
```

where $NODEn is the IP or hostname of each node in your cluster and $MASTERNODE is the node you want to run the Ambari web interface on. The $MASTERNODE should also be one of $NODEn as it requires the same resources as the nodes.

FABRIC FUNCTIONS

This section will explain each Fabric function. If you have not run the previous command then you will need to run each command below in order.

Installing Packages

Running the install_packages function installs Wget, ntpdate, ntp, epel-release, and tmux.

```
fab -f setup-fqdn.py \
install_packages \
-H $NODE1,$NODE2,$NODE3
```

For the rest of the report we assume you have installed these packages.

Copying Host File to Remote Nodes

Each node in the cluster requires the same /etc/hosts file in order to match IP to FQDN. The function copy_hosts_file will copy the file `ip_to_host` from the current directory to `/etc/hosts` on each node.

```
fab -f setup-fqdn.py \
copy_hosts_file \
-H $NODE1,$NODE2,$NODE3
```

Fixing FQDN

Ambari requires the Fully Qualified Domain Name (FQDN) to be set for each node in the cluster. A FQDN is a domain name ending in a .suffix. The hosts file set up previously implicitly defines an FQDN for every node in the cluster. To make each node in the cluster aware of it's own FQDN, run the fix_fqdn function with the host names of the computers in your cluster.

```
fab -f setup-fqdn.py \
fix_fqdn \
-H $NODE1,$NODE2,$NODE3
```

where $NODEn is the IP or hostname of each node in your cluster.

Disabling Huge Pages

Transparent huge pages allow applications to allocate very large amounts of memory very quickly. Unfortunately this memory allocation sometimes causes the Linux kernel to move memory around a lot. This reduces Spark's and Hadoop's throughput because they request large allocations very often. To disable them, run the `disable_huge_pages` function

```
fab -f setup-fqdn.py \
disable_huge_pages \
-H $NODE1,$NODE2,$NODE3
```

Disabling Firewalls

To turn off the firewall for all nodes within the cluster. We suggest implementing the firewall with Openstack security groups instead. See the Openstack configuration report for details. To turn the firewall off run the disable_firewall function with the host names of the computers in your cluster.

```
fab -f setup-fqdn.py \
disable_firewall \
-H $NODE1,$NODE2,$NODE3
```

Enabling NTP

Hadoop, Spark and many other services provided by Ambari require as accurate time as possible. The nodes should be synced explicitly during setup through the Network Time Protocol (NTP) service.

```
fab -f setup-fqdn.py \
turn_on_ntp \
-H $NODE1,$NODE2,$NODE3
```

Disabling SELinux

SELinux is enabled by default for CentOS 6. It provides a system of access security for the kernel. Unfortunately many application including Ambari do not work well with SELinux. To disable it run

```
fab -f setup-fqdn.py \
disable_selinux \
-H $NODE1,$NODE2,$NODE3
```

Importing the Ambari Repository

The Hortonworks company provides public Ambari package repositories for the package manager Yum. We use these packages as a base to avoid recompiling Ambari from source. To download the repository, run

```
fab -f setup-fqdn.py \
import_ambari_repo \
-H $NODE1,$NODE2,$NODE3
```

Setting up Ambari Server

Finally you should set up the Ambari web server on *one* of the nodes in the cluster. This node should have all of the above functions run on it already. This node will operate the web interface to manage your cluster. It will also most likely be the only node accessible from the outside internet.

```
fab -f setup-fqdn.py \
setup_ambari_server \
-H $MASTERNODE
```

where $MASTERNODE is an IP address or FQDN of one of the nodes in your cluster.

Follow the Ambari setup questions, the defaults are fine.

CONFIGURING AMBARI

Now the web interface will be available at your master node's floating IP address at port 8080. For instance our cluster was available at 165.2.1.3:8080 . When you first access the interface, the default username and password is admin. After logging in, **you should immediately change the default password for the admin account** This can be done by clicking on Users, then on the admin account, and finally clicking Change Password. **If you do not do this, your Ambari will be exposed to the general internet.**

Adding Servers to Ambari

After naming your new cluster, you should add new hosts by listing their FQDNs like 192-168-0-2.local. Note the dashes instead of dots for the FQDN. Several nodes can be added through simple pattern expressions like 192-168-0-[2-5].local . This will add hosts 192-168-0-2.local, 192-168-0-3.local, 192-168-0-4.local, 192-168-0-5.local. After entering the hosts, select your private SSH key or copy paste it into the box. It is needed for Ambari to SSH into each machine to install services. Also **make sure to change the SSH User Account from root to centos** since Centos 6 does not allow root login by default. After adding the hosts, registration should take a few minutes to confirm the hosts are healthy. Then select the services you want and follow the on-screen guide. The default assignments for masters and slaves should work.

SOURCES

1. https://ambari.apache.org/1.2.2/installing-hadoop-using-ambari/content/ambari-chap1.html 2. https://cwiki.apache.org/confluence/display/AMBARI/Start+Guide+Usin
3. Cluster Sizes https://docs.hortonworks.com/HDPDocuments/HDP2/HD
2.3.0/bk$_{c}luster - planning - guide/bk_{c}luster - planning - guide -$ 20150721.pdf