CloudConfig

Setting up Cassandra in the Cloud

If you have done work to optimize your Cassandra install in the cloud, please take a moment to contribute some of that knowledge to this page

Amazon Web Services (AWS/EC2)

- There is an ec2snitch to make Cassandra rack-aware in the ec2 cloud.
- Chef install for Cassandra, including ec2snitch setup, or the cluster_chef framework (which includes commands to launch and provision a full cluster)
- The steps listed below only require an AWS EC2 account and do not utilize the ec2snitch.

Optimizing Volume Performance for a Transient Cluster

Depending on node size, and on how many EBS volumes are attached, most EC2 nodes will have many independent attached volumes.

- How should the Cassandra config be modified to take advantage of multiple attached volumes?
- What are the tradeoffs for EBS vs local drives as backing store for a persistent cluster?

Cloud clusters are often transient: when all jobs are finished, the nodes are terminated. Any data held on EBS volumes will remain until next restart, and any data on the local scratch disks disappears forever. When the cluster restarts, each node will have a new ip addresses and identity.

- For a non-persistent cluster, can Cassandra take advantage of the scratch disks (assume they are fast but could disappear at once across the whole cluster at any time)

Monitoring Cassandra in the Cloud

CloudKick offers an out-of-the-box solution for monitoring Cassandra in the cloud. See their Cassandra Checks page to learn how to use that service to monitor.

Note on using Cloudkick for monitoring Cassandra running on Debian: Set the "Path" argument to "/usr/".

Step-By-Step Guide to Installing Cassandra on EC2 & Debian

Assumptions

We will assume that the goal is to install Cassandra in a multi-Availability Zone configuration. However, all nodes must be in one Region because we will use the private IP addresses for the nodes to talk to each other.

We also will setup Security Groups for the Cassandra nodes to talk to one another, and also for other nodes to talk to Cassandra.

In the course of this document, we reference 'lwp-request' and 'ec2_signer.pl'. These are just simple perl programs that send HTTP requests (lwp-request) and that construct a signed URL based on the parameters given (ec2_signer.pl).

Steps

Step 1. Setup the "talk to Cassandra" Security Group

This group will contain any machine which desires to communicate with Cassandra.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=CreateSecurityGroup&GroupName=Talk+To+Cassandra+Local+Zone&GroupDescription=Group+for+any+machine+that+ta
lks+to+Cassandra"`
```

Also, let us open SSH to the machines in this group.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=AuthorizeSecurityGroupIngress&GroupName=Talk+To+Cassandra+Local+Zone&IpPermissions.1.
IpProtocol=tcp&IpPermissions.1.FromPort=22&IpPermissions.1.ToPort=22&IpPermissions.1.IpRanges.1.CidrIp=0.0.0.0
/0"`
```

If you are just testing stuff, here's how you can delete the Security Group.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=DeleteSecurityGroup&GroupName=Talk+To+Cassandra+Local+Zone"`
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=DeleteSecurityGroup&GroupName=Cassandra+Nodes"`
```

Step 2. Get the OwnerID of the "talk to Cassandra" Security Group.

We need to know what the numeric OwnerID of the security group is. Write down the value returned here.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=DescribeSecurityGroups&GroupName.1=Talk+To+Cassandra+Local+Zone"`
```

It will be something like 2931201231.

Step 3. Create the "Cassandra Nodes" Security Group.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=CreateSecurityGroup&GroupName=Cassandra+Nodes&GroupDescription=Group+for+any+Cassandra+machine"`
```

Open SSH so we can get there as well.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=AuthorizeSecurityGroupIngress&GroupName=Cassandra+Nodes&IpPermissions.1.IpProtocol=tcp&IpPermissions.1.
FromPort=22&IpPermissions.1.ToPort=22&IpPermissions.1.IpRanges.1.CidrIp=0.0.0.0/0"`
```

Step 4. Get the OwnerID of the "Cassandra Nodes" Security Group.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=DescribeSecurityGroups&GroupName.1=Cassandra+Nodes"`
```

Step 5. Allow access between the Cassandra nodes on the Cassandra ports

There are 3 ports that Cassandra nodes use to talk to each other: Gossip, Thrift, and JMX.

Gossip port

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=AuthorizeSecurityGroupIngress&GroupName=Cassandra+Nodes&IpPermissions.1.IpProtocol=tcp&IpPermissions.1.
FromPort=7000&IpPermissions.1.ToPort=7000&IpPermissions.1.Groups.1.UserId=152252226102&IpPermissions.1.Groups.1.
GroupName=Cassandra+Nodes"`
```

Thrift Port

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=AuthorizeSecurityGroupIngress&GroupName=Cassandra+Nodes&IpPermissions.1.IpProtocol=tcp&IpPermissions.1.
FromPort=9160&IpPermissions.1.ToPort=9160&IpPermissions.1.Groups.1.UserId=152252226102&IpPermissions.1.Groups.1.
GroupName=Cassandra+Nodes"`
```

JMX Port

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=AuthorizeSecurityGroupIngress&GroupName=Cassandra+Nodes&IpPermissions.1.IpProtocol=tcp&IpPermissions.1.
FromPort=8080&IpPermissions.1.ToPort=8080&IpPermissions.1.Groups.1.UserId=152252226102&IpPermissions.1.Groups.1.
GroupName=Cassandra+Nodes"`
```

Step 6. Allow access between "Talk to Cassandra" nodes and the Cassandra nodes themselves.

The non-Cassandra nodes use Thrift to talk to Cassandra. So we need to open that port to the talkers. We also open the JMX port so that monitoring can occur.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=AuthorizeSecurityGroupIngress&GroupName=Cassandra+Nodes&IpPermissions.1.IpProtocol=tcp&IpPermissions.1.
FromPort=9160&IpPermissions.1.ToPort=9160&IpPermissions.1.Groups.1.UserId=152252226102&IpPermissions.1.Groups.1.
GroupName=Talk+To+Cassandra+Local+Zone"`
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=AuthorizeSecurityGroupIngress&GroupName=Cassandra+Nodes&IpPermissions.1.IpProtocol=tcp&IpPermissions.1.
FromPort=8080&IpPermissions.1.ToPort=8080&IpPermissions.1.Groups.1.UserId=152252226102&IpPermissions.1.Groups.1.
GroupName=Talk+To+Cassandra+Local+Zone"`
```

Step 7. Create a Key Pair

You might already have a key pair. You must have a key-pair to log into an EC2 instance. If you already have one and have the private key for it, you can safely skip this step.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=CreateKeyPair&KeyName=CassandraLauncher-East"`
```

Record the keyMaterial in the response.

Step 8. Pick the correct seed instances.

For Cassandra, we use 64bit Debian Lenny. Currently, the available AMIs are:

- us-east-1: ami-f0f61599
- us-west-1: ami-4d3d6c08
- eu-west-1: ami-80446ff4
- ap-southeast-1: ami-a3f38cf1

These come from http://alestic.com/.

Step 9. Select the Availability Zone you want.

For the us-east-1, there are 4 AZs:

- us-east-1a
- us-east-1b
- us-east-1c
- us-east-1d

Step 10. Start up your SEED instance

NOTE, the KeyName must be one for which you have the security key (from the KeyPair). In the example below, we use the key name 'dviner' and the AZ us-east-1a. Replace these with your key name and AZ selection.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=RunInstances&ImageId=ami-f0f61599&MinCount=1&MaxCount=1&KeyName=dviner&SecurityGroup.
1=Cassandra+Nodes&InstanceType=m1.large&DisableApiTermination=false&Monitoring.Enabled=false&Placement.
AvailabilityZone=us-east-1a"`
```

RECORD THE INSTANCE ID (/RunInstancesResponse/instancesSet/item/instanceId)

Step 11. Get the IP address of the SEED instance

You must insert the InstanceId obtained in Step 10.

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=DescribeInstances&InstanceId.1=INSTANCE-ID"`
```

Step 12. Setup Cassandra on SEED

See below for this information.

Step 13. Start up a NON-SEED instance

This is exactly like launching a SEED instance.

Step 14. Setup Cassandra on NON-SEED

See below.

Step 15. Shutting Down an Instance

```
% lwp-request -SUse `perl ../../ec2/ec2_signer.pl "https://ec2.us-east-1.amazonaws.com/?
Action=TerminateInstances&InstanceId=INSTANCE-ID"`
```

Cassandra Basic Setup

The steps here assume that you have an instance running and you can connect to it. These steps are applicable for both seed nodes and non-seed nodes.

Step 1. Add the Cassandra APT repository

```
% cat setup/cassandra.list
    # taken from http://wiki.apache.org/cassandra/DebianPackaging
    deb http://www.apache.org/dist/cassandra/debian unstable main
    deb-src http://www.apache.org/dist/cassandra/debian unstable main
% scp -i ../your-private-key cassandra.list root@YOUR-INSTANCE.amazonaws.com:/etc/apt/sources.list.d/
```

Step 2. Add the GPG keys on the instance

```
% apt-get update
% apt-get upgrade
% gpg --keyserver wwwkeys.pgp.net --recv-keys F758CE318D77295D
% gpg --export --armor F758CE318D77295D | apt-key add -
```

Step 3. Install the Debian package for Cassandra

```
% apt-get update
% apt-get install cassandra
```

At this point, Cassandra will be installed and running. However, it's not configured for a multi-node cluster. So we need to continue.

Step 4. Turn off the default Cassandra

```
% /etc/init.d/cassandra stop
% rm /var/lib/cassandra/commitlog/*
% rm -r /var/lib/cassandra/data/
```

Cassandra Seed & Non-Seed Configurations

In setting up the seed and non-seed nodes, we just alter the configuration file (which lives at /etc/cassandra/storage-conf.xml). These are the important differences:

SEED CONF:

- LISTENADDRESS is IP of this node
- ThriftAddress is IP of this node (or 0.0.0.0)
- SEED is IP of this node
- AutoBootstrap to off

NON-SEED

- LISTENADDRESS is IP of this node
- ThriftAddress is IP of this node (or 0.0.0.0)
- SEED is the IP of the earlier seed
- AutoBootstrap to on

Booting up Cassandra

After installing the appropriate config file, you boot up Cassandra:

```
% tail -f /var/log/cassandra/output.log
% tail -f /var/log/cassandra/system.log
```

```
% /etc/init.d/cassandra start
```

It definitely takes a few minutes for each new node to find all the other nodes. You can query any node to see what it thinks the cluster is composed of:

```
% nodetool  -h localhost ring
```