riptano / **ComboAMI**

The AMI takes a set of input parameters via the EC2 user-data to install, RAID, ring, and launch a DataStax Enterprise/Community cluster.

mlococo Document deprecated ami names — Latest commit `0dc0728` on Dec 17, 2015

build	Create ami_id_extractor.py and update ami_ids.json	6 months ago
community_cloud	Clarified a few things within the Cloudformation script	2 years ago
repo_keys	Update DataStax package repository key	6 months ago
AMI_LICENSE.txt	2.0	5 years ago
EXPANDING.md	Added back spaces to EXPANDING.	4 years ago
FILES.md	Remove the presetup directory, replaced by packer build	8 months ago
README.md	Document deprecated ami names	2 months ago
VERSION	Prep for 2.6 release	6 months ago
ami_ids.json	Update ami_ids.json for 2.6.3	6 months ago
conf.py	Python Coding Standards mostly met. Breaking down and proper naming o…	4 years ago
ds0_updater.py	Fix the git-update code	6 months ago
ds0_utils.py	Prep for 2.6.3 - Built 2.6.2 without pushing	6 months ago
ds1_launcher.py	wait_for_first_node.sh	10 months ago
ds2_configure.py	removed 'shlex.split' processing of base64postscript var	2 months ago
ds3_after_init.py	Fixing issue with getting public hostname from inside a VPC	2 years ago
ds4_motd.py	Provide a default –repository	8 months ago
logger.py	Trim trailing whitespace in logger.py	8 months ago
reflector.php	Current reflector code with JSON response	4 years ago
tokentool.html	Better nodetool that takes into account preceding blank datacenters	4 years ago
tokentoolv2.py	Hide excessive tokentool debugs	4 years ago
wait_for_first_node.sh	wait_for_first_node.sh	10 months ago

README.md

Summary

DataStax's Amazon Machine Image a quick way to test a DataStax Community or DataStax Enterprise cluster on EC2.

Quickstart

1. Log into the AWS console with your web browser
2. Select the EC2 service
3. Find the ami-id's for your region. Note that the AMI with the name of "DataStax Auto-Clustering AMI" that has no version number specified is from the 2.4 series and is deprecated. Select an ami-id from from the list in this repo to ensure you're getting the latest fixes.
4. "Launch Instance" -> Community AMI's -> Search for your ami-id -> "Select"
5. Select an instance type (m3.medium is good for low-throughput testing)
6. "Next: Configure Instance Details" -> "Advanced Details" -> add "User Data" of "--clustername test-cluster --totalnodes 1 --version community"
7. "Review and Launch" -> "Launch" -> Select keypair
8. SSH to your new cassandra cluster and run `nodetool status`

If you frequently launch scratch clusters, you may be interested in cassandralauncher

For detailed instructions on launching, visit http://docs.datastax.com/en/latest-dsc-ami

Options

Basic AMI Switches:

```
--clustername <name>
    The name of the Cassandra cluster
    REQUIRED

--totalnodes <#>
    Cluster size
    REQUIRED

--version [ community | enterprise ]
    Installs either DataStax Enterprise or
    DataStax Community Edition
    REQUIRED

--rpcbinding
    Binds the rpc_address to the private IP
    address of the instance
    Default: false, uses 0.0.0.0
```

DataStax Enterprise Specific:

```
--username <user>
    The username provided during DSE registration
    --password is REQUIRED in order to use this option
    REQUIRED for a DSE installation

--password <pass>
    The password provided during DSE registration
    --username is REQUIRED in order to use this option
    REQUIRED for a DSE installation

--analyticsnodes <#>
    Number of analytics nodes that run with Spark
    Note: Uses Hadoop in versions earlier than DSE 4.5
    Default: 0
```

```
--searchnodes <#>
    Number of search nodes that run with Solr
    Default: 0

--hadoop
    Force Hadoop over Spark on analytics nodes
    Default: false, uses Spark on 4.5+
```

Advanced:

```
--release <release_version>
    Allows for the installation of a previous DSE version
    Example: 1.0.2
    Default: Ignored

--cfsreplicationfactor <#>
    The CFS replication factor
    Note: --cfsreplicationfactor must be <= --analyticsnodes
    Default: 1

--opscenter no
    Disables the installation of OpsCenter on the cluster
    Default: yes

--reflector <url>
    Allows you to use your own reflector
    Default: http://reflector2.datastax.com/reflector2.php

--repository <repository>
    Allows you to set a custom repository to pull configuration files from
    Default: none, falls back to repository used to create the AMI
    Examples: https://github.com/riptano/ComboAMI#2.5
              https://github.com/riptano/ComboAMI#e5e3d41fb5f12461509aa1b6079413b381930d81

--postscript_url <url>
    Allows you to download and execute a post install custom script
    Default: none

--base64postscript <base64_encoded_commands>
    Allows you to specify a list of base64 encoded semi-colon/newline separated commands to be executed post installa
    Default: none
    Example: ZWNobyAtbiAiY2FzcyI7IGVjaG8gLW4gImFuZHJhIg== (echo -n "cass"; echo -n "andra")
             c3VkbyBhcHQtZ2V0IGluc3RhbGwgY29sbGVjdGQ= (sudo apt-get install collectd)
```

Security Groups

For information on setting up security groups, see the Datastax Documentation

Contributing

Pull requests are welcome. Consider creating an issue to discus the feature before doing the development work, or just fork and create a PR based off the dev-2.6 branch.