Using AWS environment to learn druid capabilities

http://metamarkets.com/

Table of Contents

1. Introduction

This document will help users to understand Druid and its capabilities in managing real time and stored data. Document shows how users can instantiate a Stack based on pre-existing template on AWS domain. This automatically generates various components i.e. master, compute nodes, broker, mysql and zookeeper to get user started on Druid technology. A pre-built database with set of segments allows them to instantly make search operation. Also provided is an easy to use Query interface to fetch data stored in compute nodes. Interface shows various querying capabilities available in Druid i.e. dimensions, filters, granularity etc.

The document also briefly covers explains the setup points needed to configure templates using Cloud Formation API (AWS) and how to instantiate stacks using existing template. Cloud Formation and Stacks are extensive AWS topics and out of scope of this document. Basic introduction is provided in document to get user started.

2. About Druid

Druid is the first open source, distributed, analytical data store designed to scale horizontally. Druid is better to meet the needs of analytics workloads requiring fast, real-time access to data at scale.

Druid's key features are:

1. **Distributed architecture**. Swappable read-only data segments using swapping protocol. Per-segment replication relieves load on hot segments. Supports both in-memory and memory-mapped versions.
2. **Real-time ingestion.** Real-time ingestion coupled with broker servers to query across real-time and historical data. Automated migration of real-time to historical as it ages.
3. **Column-oriented for speed**. Data is laid out in columns so that scans are limited to specific data being searched. Compression decreases overall data footprint.
4. **Fast filtering**. Bitmap indices with CONCISE compression.
5. **Operational simplicity**. Fault tolerant due to replication. Supports rolling deployments and restarts. Allows simple scale up and scale down – just add or remove nodes.

3. Objectives of Druid

1. It gives an, efficient way to make Real-time analytics data store.
2. Druid will give maximum storage with minimum storage utility
3. Druid is fast and cheap for Big Data Analytics.
4. Druid data store's performance scales up well to a 6TB in-memory cluster and degrades gracefully under memory pressure.
5. Druid will give Real Time Analytics with billions of rows per second.

4. Druid Infrastructure Setup Points

1. Druid Infrastructure contains master node and compute nodes. Master will work with various compute nodes to handle a given data request. Data Query request is sent to Broker which internally communicates with master and routes the same via master to various compute nodes.
2. Below diagram gives schematic representation of 3 compute nodes and 1 master node which will be created using AWS template.
3. Master AMI that will act as Master and Broker, and would have Zookeeper and MySQL database.
4. Compute node AMI will be instantiated over three nodes to allow infrastructure to cater big data request.

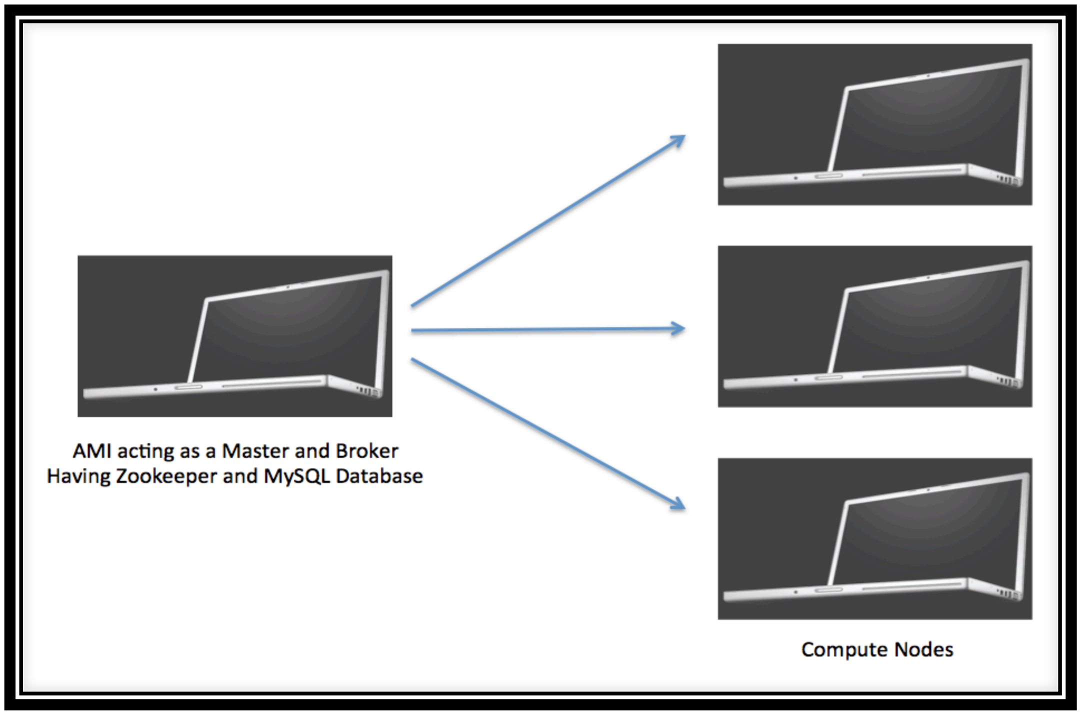

AMI acting as a Master and Broker
Having Zookeeper and MySQL Database
Compute Nodes

Figure 1

5. Creation of User Account in AWS

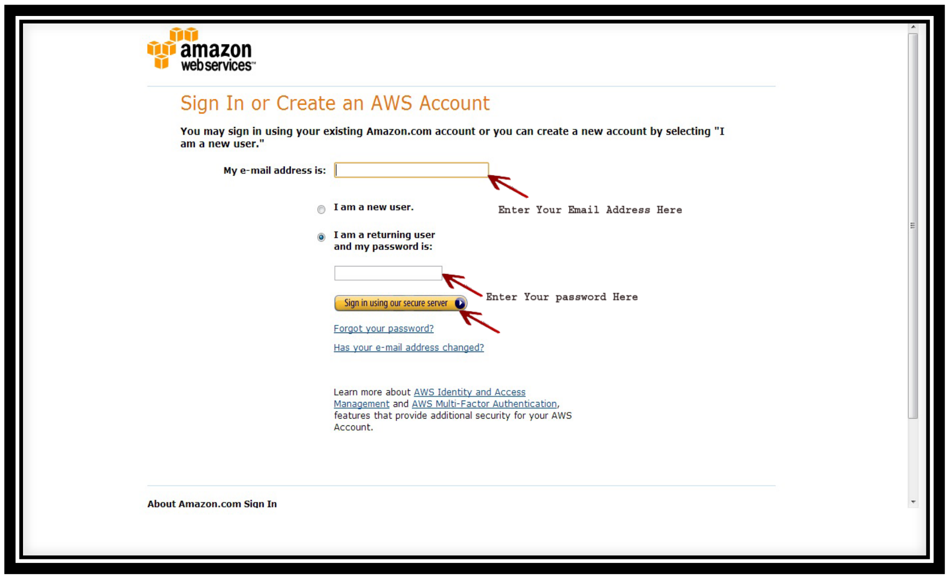
To create an account on AWS follows this link http://aws.amazon.com/ to Sign Up for one (In case you already have an AWS account please skips this section and move to next section)

6. AWS Console& Stack Creation

AWS Console is used to Access and manage Amazon Web Services through a simple and intuitive web-based user interface. The AWS Management Console provides a simple web interface for Amazon Web Services.

1. Log in using https://console.aws.amazon.com/console/home

2. Enter the Username Password and press "Sign in using our secure server"



Sign In or Create an AWS Account

You may sign in using your existing Amazon.com account or you can create a new account by selecting "I am a new user."

My e-mail address is:

Enter Your Email Address Here

I am a new user.

I am a returning user and my password is:

Enter Your password Here

Sign in using our secure server

Forgot your password?

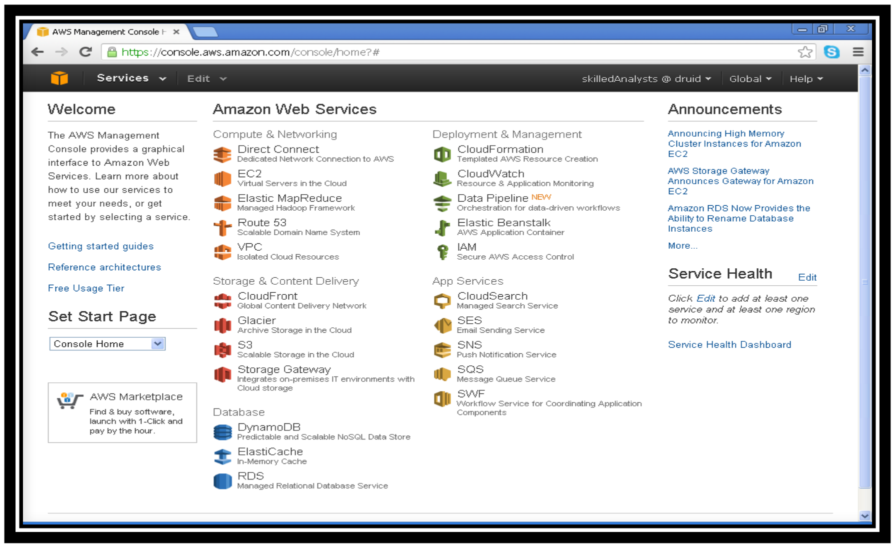
Has your e-mail address changed?

Learn more about AWS Identity and Access Management and AWS Multi-Factor Authentication, features that provide additional security for your AWS Account.

About Amazon.com Sign In

Figure 2

3. After click you will directly go to the Welcome Screen on AWS Console.



The AWS Management Console provides a graphical interface to Amazon Web Services. Learn more about how to use our services to meet your needs, or get started by selecting a service.

Find & buy software, launch with 1-Click and pay by the hour.

Announcing High Memory Cluster Instances for Amazon EC2

AWS Storage Gateway Announces Gateway for Amazon EC2

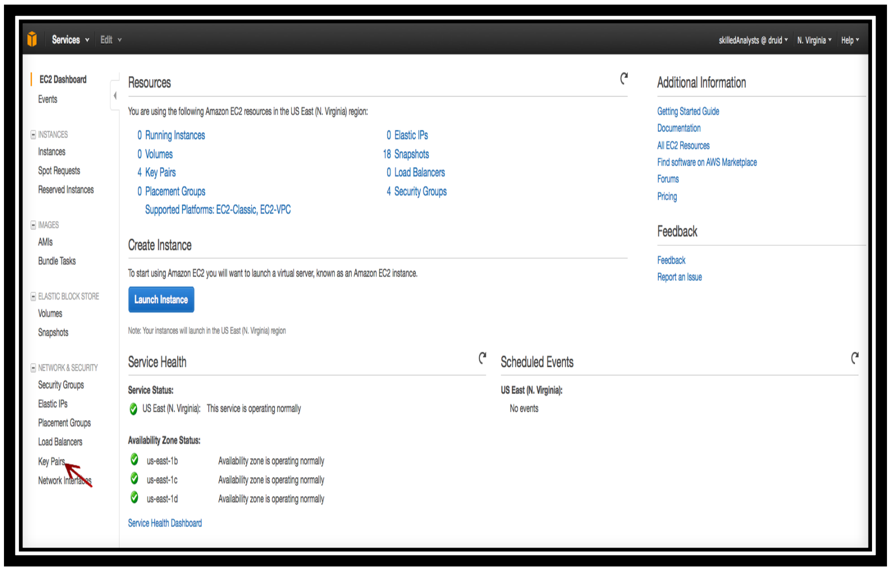
Amazon RDS Now Provides the Ability to Rename Database Instances

Click Edit to add at least one service and at least one region to monitor.

Figure 3

In case you do not see Deployment & Management (CloudFormation) in AWS console, please refer AWS help manual for the same.

4. Select EC2 Dashboard and click on " Key Pairs"



You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Service Status:

US East (N. Virginia): This service is operating normally

Availability Zone Status:

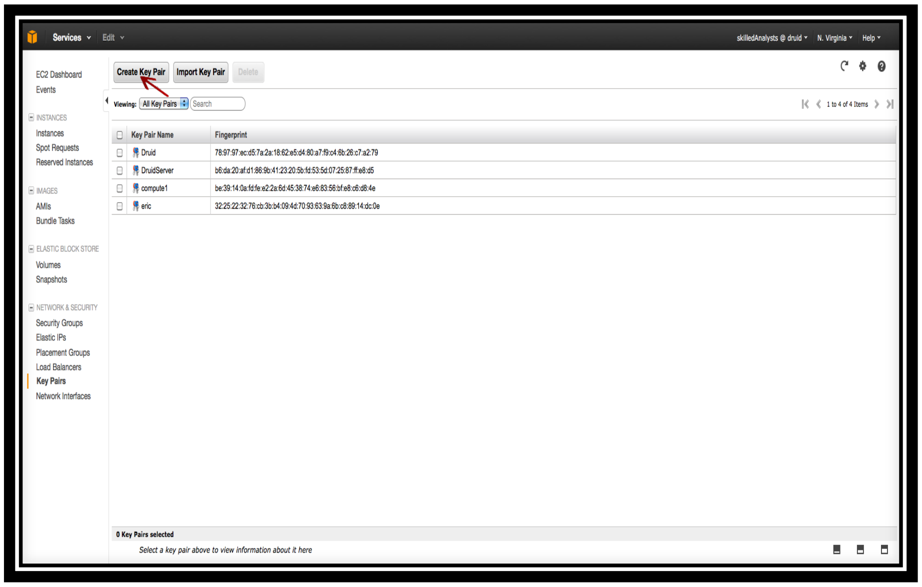
us-east-1b	Availability zone is operating normally
us-east-1c	Availability zone is operating normally
us-east-1d	Availability zone is operating normally

US East (N. Virginia):

No events

Figure 4

5. Press the “Create Key Pair” button.



Key Pair Name	Fingerprint
Druid	78:97:97:ec:d5:7a:2a:18:62:e5:d4:80:a7:f9:c4:6b:26:c7:a2:79
DruidServer	b6:da:20:af:d1:86:9b:41:23:20:5b:fd:53:5d:07:25:87:ff:e8:d5
compute1	be:39:14:0a:fd:fe:e2:2a:6d:45:38:74:e6:83:56:bf:e8:c6:d8:4e
eric	32:25:22:32:76:cb:3b:b4:09:4d:70:93:63:9a:6b:c8:89:14:dc:0e

Figure 5

6. A dialogue box will appear, enter the Key Pair name you want to add and press the “Create” button.

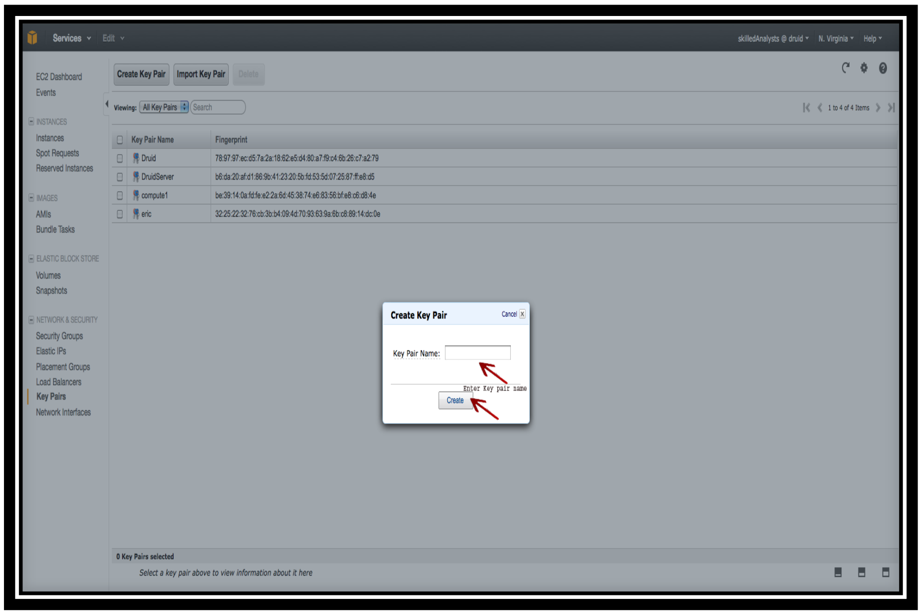
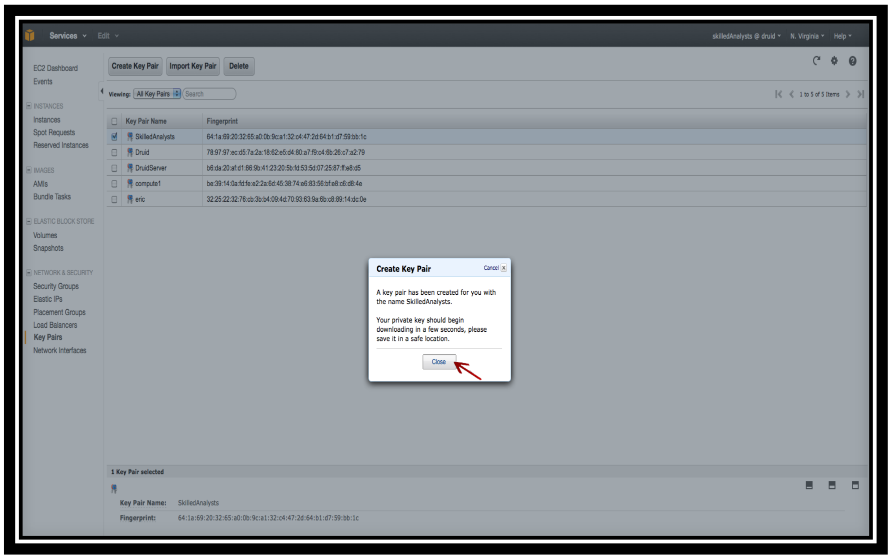


Figure 6

7. Again a dialogue box will open saying that the Key Pair has been successfully created and press the “Close” button.



Key Pair Name	Fingerprint
SkilledAnalysts	64:1a:69:20:32:65:a0:0b:9c:a1:32:c4:47:2d:64:b1:d7:59:bb:1c
Druid	78:97:97:ec:d5:7a:2a:18:62:e5:d4:80:a7:f9:c4:6b:26:c7:a2:79
DruidServer	b6:da:20:af:d1:86:9b:41:23:20:5b:fd:53:5d:07:25:87:ff:e8:d5
compute1	be:39:14:0a:fd:fe:e2:2a:6d:45:38:74:e6:83:56:bf:e8:c6:d8:4e
eric	32:25:22:32:76:cb:3b:b4:09:4d:70:93:63:9a:6b:c8:89:14:dc:0e

Create Key Pair

A key pair has been created for you with the name SkilledAnalysts.

Your private key should begin downloading in a few seconds, please save it in a safe location.

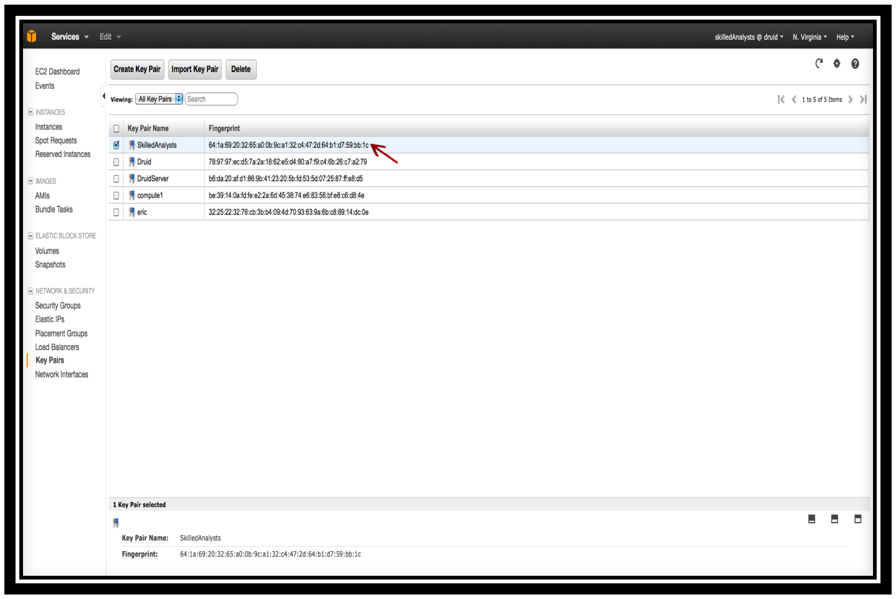
1 Key Pair selected

Key Pair Name: SkilledAnalysts

Fingerprint: 64:1a:69:20:32:65:a0:0b:9c:a1:32:c4:47:2d:64:b1:d7:59:bb:1c

Figure 7

8. After then you can see the created Key Pair list.



Key Pair Name	Fingerprint
SkilledAnalysts	64:1a:69:20:32:65:a0:0b:9c:a1:32:c4:47:2d:64:b1:d7:59:bb:1c
Druid	78:97:97:ec:d5:7a:2a:18:62:e5:d4:80:a7:f9:c4:6b:26:c7:a2:79
DruidServer	b6:da:20:af:d1:86:9b:41:23:20:5b:fd:53:5d:07:25:87:ff:e8:d5
compute1	be:39:14:0a:fd:fe:e2:2a:6d:45:38:74:e6:83:56:bf:e8:c6:d8:4e
eric	32:25:22:32:76:cb:3b:b4:09:4d:70:93:63:9a:6b:c8:89:14:dc:0e

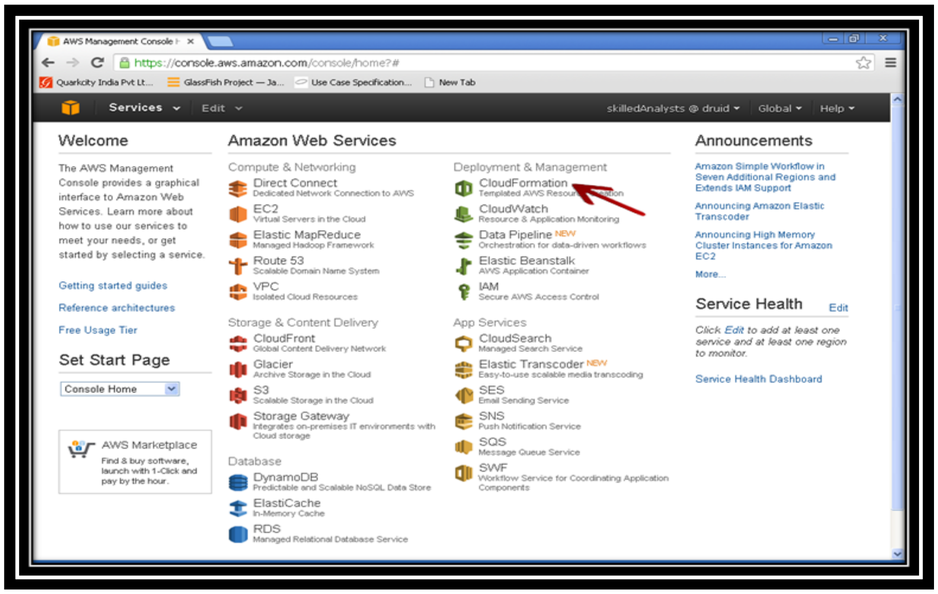
1 Key Pair selected

Key Pair Name: SkilledAnalysts

Fingerprint: 64:1a:69:20:32:65:a0:0b:9c:a1:32:c4:47:2d:64:b1:d7:59:bb:1c

Figure 8

9. Go to the homepage and Click on the CloudFormation under Deployment & Management.


AWS Management Console
https://console.aws.amazon.com/console/home?#
Quarkcity India Pvt Lt... GlassFish Project — Ja... Use Case Specification... New Tab
Services Edit skilledAnalysts @ druid Global Help

Welcome

The AWS Management Console provides a graphical interface to Amazon Web Services. Learn more about how to use our services to meet your needs, or get started by selecting a service.

Getting started guides
Reference architectures
Free Usage Tier

Set Start Page
Console Home

AWS Marketplace
Find & buy software, launch with 1-Click and pay by the hour.

Amazon Web Services

Compute & Networking
Direct Connect — Dedicated Network Connection to AWS
EC2 — Virtual Servers in the Cloud
Elastic MapReduce — Managed Hadoop Framework
Route 53 — Scalable Domain Name System
VPC — Isolated Cloud Resources

Storage & Content Delivery
CloudFront — Global Content Delivery Network
Glacier — Archive Storage in the Cloud
S3 — Scalable Storage in the Cloud
Storage Gateway — Integrates on-premises IT environments with Cloud storage

Database
DynamoDB — Predictable and Scalable NoSQL Data Store
ElastiCache — In-Memory Cache
RDS — Managed Relational Database Service

Deployment & Management
CloudFormation — Templated AWS Resource Creation
CloudWatch — Resource & Application Monitoring
Data Pipeline NEW — Orchestration for data-driven workflows
Elastic Beanstalk — AWS Application Container
IAM — Secure AWS Access Control

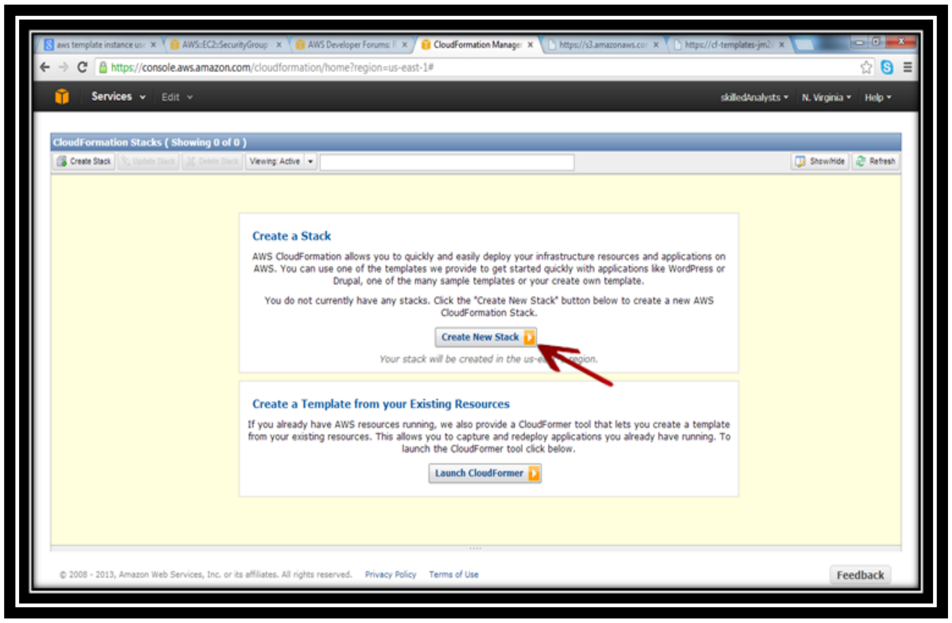
App Services
CloudSearch — Managed Search Service
Elastic Transcoder NEW — Easy-to-use scalable media transcoding
SES — Email Sending Service
SNS — Push Notification Service
SQS — Message Queue Service
SWF — Workflow Service for Coordinating Application Components

Announcements
Amazon Simple Workflow in Seven Additional Regions and Extends IAM Support
Announcing Amazon Elastic Transcoder
Announcing High Memory Cluster Instances for Amazon EC2
More...

Service Health Edit
Click Edit to add at least one service and at least one region to monitor.
Service Health Dashboard

Figure 9

10. To create a new Stack, press “Create New Stack” button


https://console.aws.amazon.com/cloudformation/home?region=us-east-1#
Services Edit skilledAnalysts N. Virginia Help

CloudFormation Stacks (Showing 0 of 0)
Create Stack | Update Stack | Delete Stack | Viewing: Active | Show/Hide | Refresh

Create a Stack

AWS CloudFormation allows you to quickly and easily deploy your infrastructure resources and applications on AWS. You can use one of the templates we provide to get started quickly with applications like WordPress or Drupal, one of the many sample templates or your create own template.

You do not currently have any stacks. Click the "Create New Stack" button below to create a new AWS CloudFormation Stack.

Create New Stack

Your stack will be created in the us-east-1 region.

Create a Template from your Existing Resources

If you already have AWS resources running, we also provide a CloudFormer tool that lets you create a template from your existing resources. This allows you to capture and redeploy applications you already have running. To launch the CloudFormer tool click below.

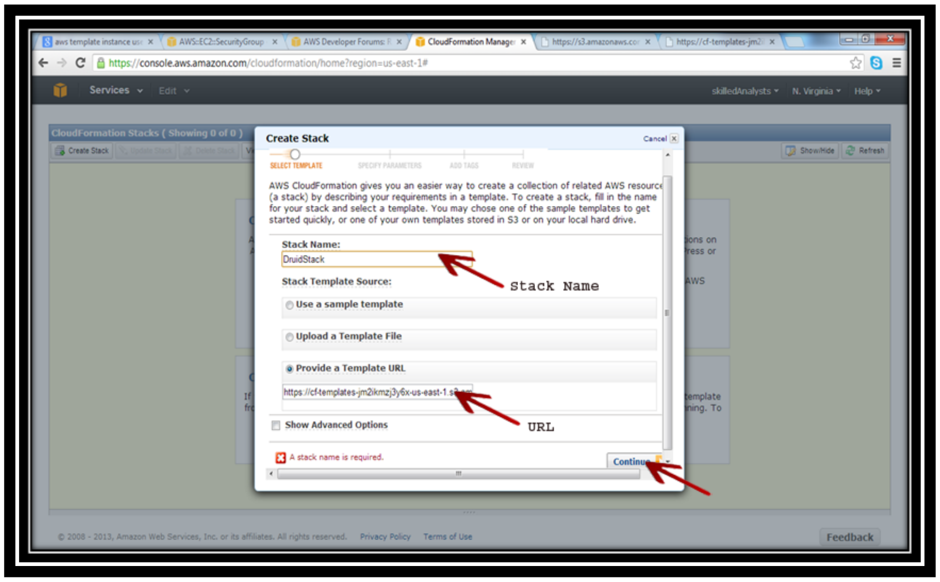
Launch CloudFormer

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Figure 10

Stack is predefined set of AMI’s instances launched based on user provided template file or URL. In our case we have 1 Master node (Ubuntu 12.0.4 machine) and 3 Compute nodes (Ubuntu 12.0.4 machine). On the stack creation, the master node will register itself to zookeeper and launch a preconfigured database. So the user will be able to query the compute nodes for results. More technical details are explained under Administrator Details section.

11. Enter the Stack name and provide a template URL https://s3.amazonaws.com/cf-templates-jm2ikmzj3y6x-us-east-1/2013081cA9-Druid04012013.template and press “Continue”



Create Stack

AWS CloudFormation gives you an easier way to create a collection of related AWS resource (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of your own templates stored in S3 or on your local hard drive.

Stack Name: DruidStack

Stack Template Source:

Use a sample template

Upload a Template File

[✓ Provide a Template URL]

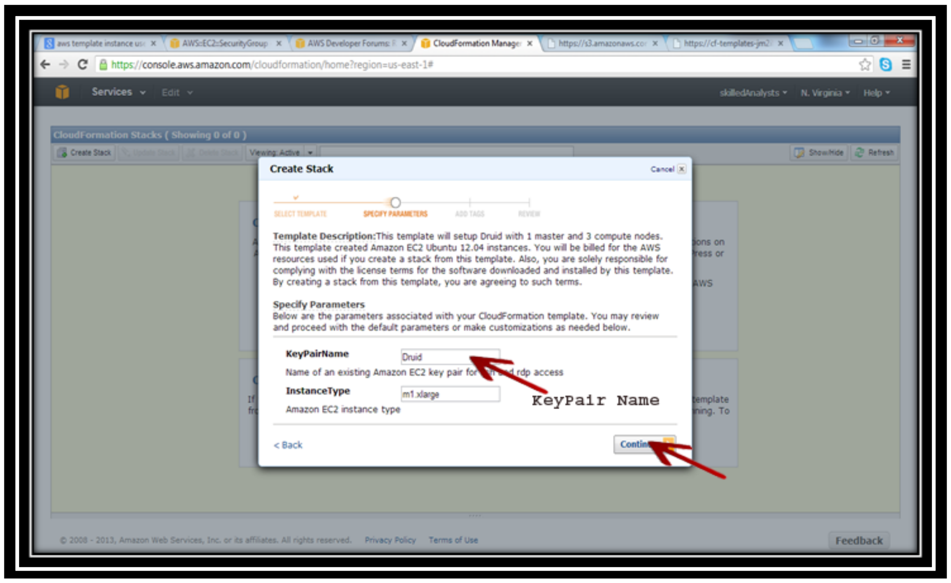
https://cf-templates-jm2ikmzj3y6x-us-east-1.s3.am

Show Advanced Options

A stack name is required.

Figure 11

12. Enter the Key Pair name and press “Continue” button.
    Note: ***Here we are using Druid as Keypair name.***



Create Stack

Template Description: This template will setup Druid with 1 master and 3 compute nodes. This template created Amazon EC2 Ubuntu 12.04 instances. You will be billed for the AWS resources used if you create a stack from this template. Also, you are solely responsible for complying with the license terms for the software downloaded and installed by this template. By creating a stack from this template, you are agreeing to such terms.

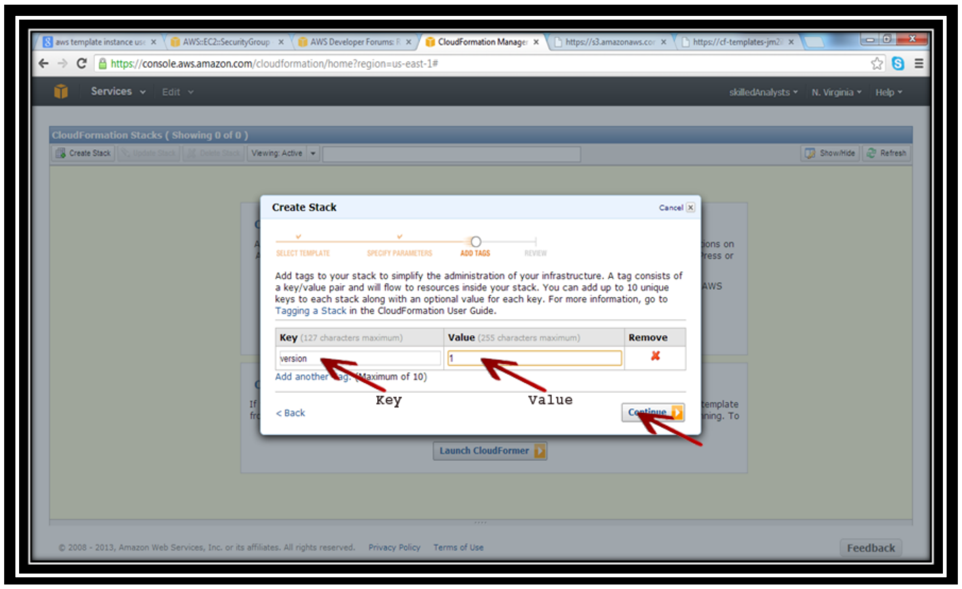
Specify Parameters
Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

KeyPairName: Druid
Name of an existing Amazon EC2 key pair for ssh and rdp access

InstanceType: m1.xlarge
Amazon EC2 instance type

Figure 12

13. Enter the Key and its Value, then press “Continue” button



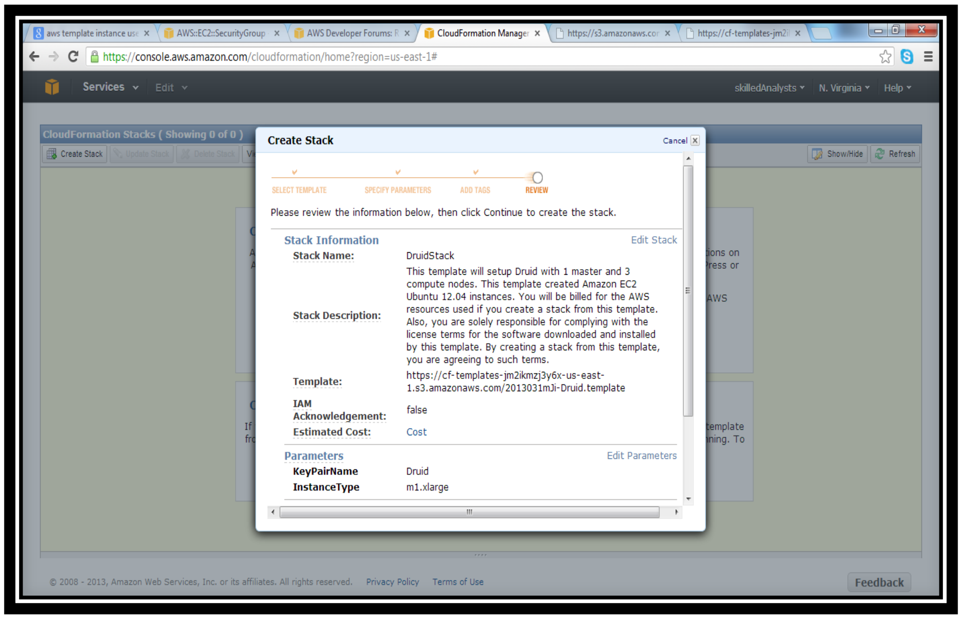
Add tags to your stack to simplify the administration of your infrastructure. A tag consists of a key/value pair and will flow to resources inside your stack. You can add up to 10 unique keys to each stack along with an optional value for each key. For more information, go to Tagging a Stack in the CloudFormation User Guide.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
version	1	✘

Add another tag. (Maximum of 10)

Figure13

14. After continue you will directly go to the Stack Information page



Please review the information below, then click Continue to create the stack.

Stack Information

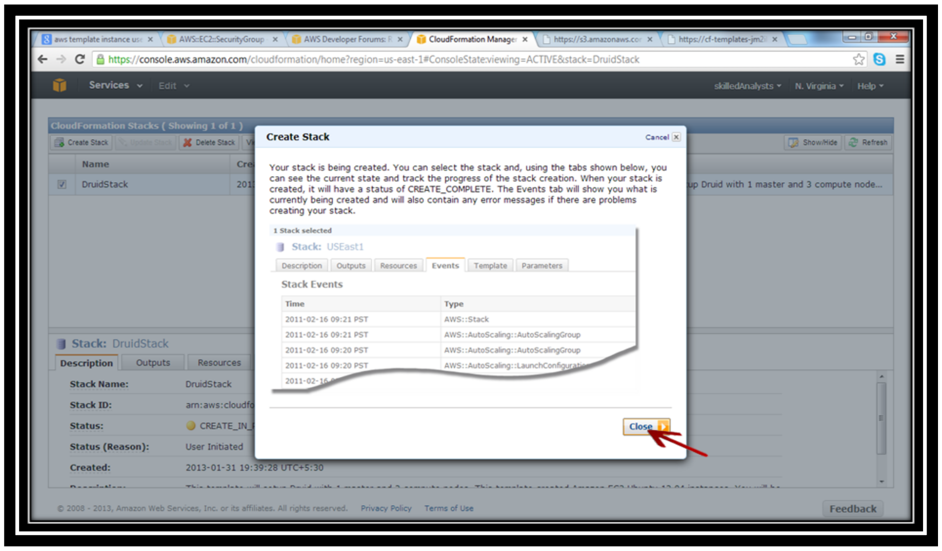
Stack Name:	DruidStack
Stack Description:	This template will setup Druid with 1 master and 3 compute nodes. This template created Amazon EC2 Ubuntu 12.04 instances. You will be billed for the AWS resources used if you create a stack from this template. Also, you are solely responsible for complying with the license terms for the software downloaded and installed by this template. By creating a stack from this template, you are agreeing to such terms.
Template:	https://cf-templates-jm2ikmzj3y6x-us-east-1.s3.amazonaws.com/2013031mJi-Druid.template
IAM Acknowledgement:	false
Estimated Cost:	Cost

Parameters

KeyPairName	Druid
InstanceType	m1.xlarge

Figure 14

15. You will be able to see the created Stack and then press “Close” button



Create Stack

Your stack is being created. You can select the stack and, using the tabs shown below, you can see the current state and track the progress of the stack creation. When your stack is created, it will have a status of CREATE_COMPLETE. The Events tab will show you what is currently being created and will also contain any error messages if there are problems creating your stack.

1 Stack selected

Stack: USEast1

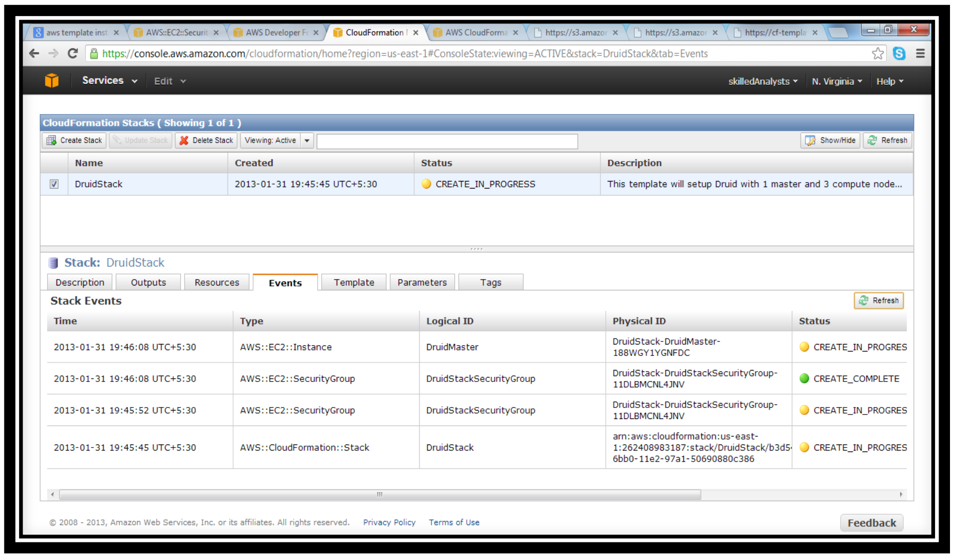
Stack Events

Time	Type
2011-02-16 09:21 PST	AWS::Stack
2011-02-16 09:21 PST	AWS::AutoScaling::AutoScalingGroup
2011-02-16 09:20 PST	AWS::AutoScaling::AutoScalingGroup
2011-02-16 09:20 PST	AWS::AutoScaling::LaunchConfigura

<u>Figure 15</u>

16. It will redirect you on the Cloud Formation Stacks page and show you the progress bar of created Stacks.

Note: *<u>Go to the Events and press the “Refresh” button it will show the progress of the Stack. User needs to press “Refresh” button till the Creation of Stack completes. Once the Stack creation is completed then only the User can see the created instances up and running.</u>*



https://console.aws.amazon.com/cloudformation/home?region=us-east-1#ConsoleState:viewing=ACTIVE&stack=DruidStack&tab=Events

CloudFormation Stacks (Showing 1 of 1)

	Name	Created	Status	Description
✓	DruidStack	2013-01-31 19:45:45 UTC+5:30	● CREATE_IN_PROGRESS	This template will setup Druid with 1 master and 3 compute node...

Stack: DruidStack

Stack Events

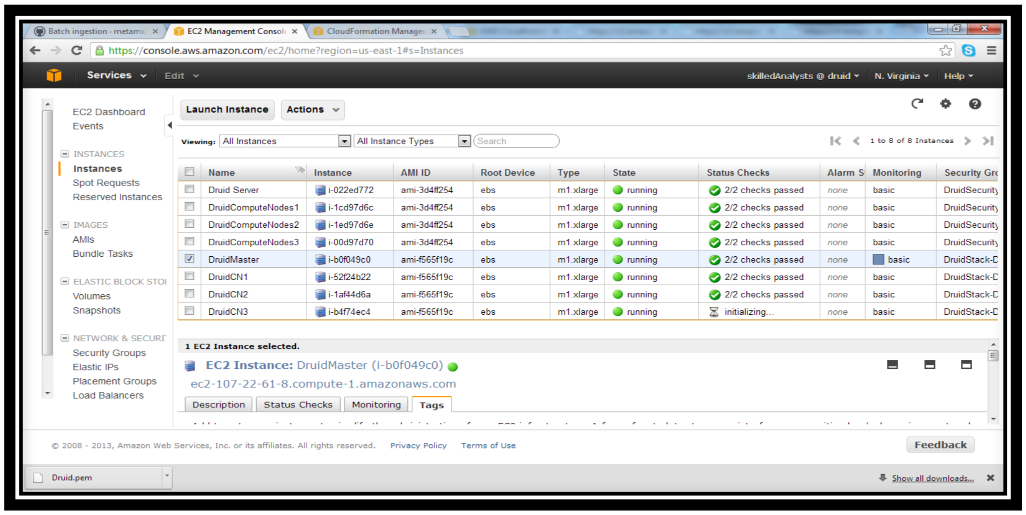
Time	Type	Logical ID	Physical ID	Status
2013-01-31 19:46:08 UTC+5:30	AWS::EC2::Instance	DruidMaster	DruidStack-DruidMaster-188WGY1YGNFDC	● CREATE_IN_PROGRES
2013-01-31 19:46:08 UTC+5:30	AWS::EC2::SecurityGroup	DruidStackSecurityGroup	DruidStack-DruidStackSecurityGroup-11DLBMCNL4JNV	● CREATE_COMPLETE
2013-01-31 19:45:52 UTC+5:30	AWS::EC2::SecurityGroup	DruidStackSecurityGroup	DruidStack-DruidStackSecurityGroup-11DLBMCNL4JNV	● CREATE_IN_PROGRES
2013-01-31 19:45:45 UTC+5:30	AWS::CloudFormation::Stack	DruidStack	arn:aws:cloudformation:us-east-1:262408983187:stack/DruidStack/b3d5-6bb0-11e2-97a1-50690880c386	● CREATE_IN_PROGRES

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved.

<u>Figure 16</u>

17. You can see the created Instances in terms of Druid Server and Compute Nodes by clicking on Instances option in EC2 Dashboard.

<u>Stack will generate its instances automatically, wait for few minutes to let it get completed. User is not required to create any AMI/Instances manually.</u>



	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Alarm S	Monitoring	Security Gro
☐	Druid Server	i-022ed772	ami-3d4ff254	ebs	m1.xlarge	● running	✓ 2/2 checks passed	none	basic	DruidSecurity
☐	DruidComputeNodes1	i-1cd97d6c	ami-3d4ff254	ebs	m1.xlarge	● running	✓ 2/2 checks passed	none	basic	DruidSecurity
☐	DruidComputeNodes2	i-1ed97d6e	ami-3d4ff254	ebs	m1.xlarge	● running	✓ 2/2 checks passed	none	basic	DruidSecurity
☐	DruidComputeNodes3	i-00d97d70	ami-3d4ff254	ebs	m1.xlarge	● running	✓ 2/2 checks passed	none	basic	DruidSecurity
☑	DruidMaster	i-b0f049c0	ami-f565f19c	ebs	m1.xlarge	● running	✓ 2/2 checks passed	none	basic	DruidStack-D
☐	DruidCN1	i-52f24b22	ami-f565f19c	ebs	m1.xlarge	● running	✓ 2/2 checks passed	none	basic	DruidStack-D
☐	DruidCN2	i-1af44d6a	ami-f565f19c	ebs	m1.xlarge	● running	✓ 2/2 checks passed	none	basic	DruidStack-D
☐	DruidCN3	i-b4f74ec4	ami-f565f19c	ebs	m1.xlarge	● running	initializing...	none	basic	DruidStack-D

1 EC2 Instance selected.

EC2 Instance: DruidMaster (i-b0f049c0)

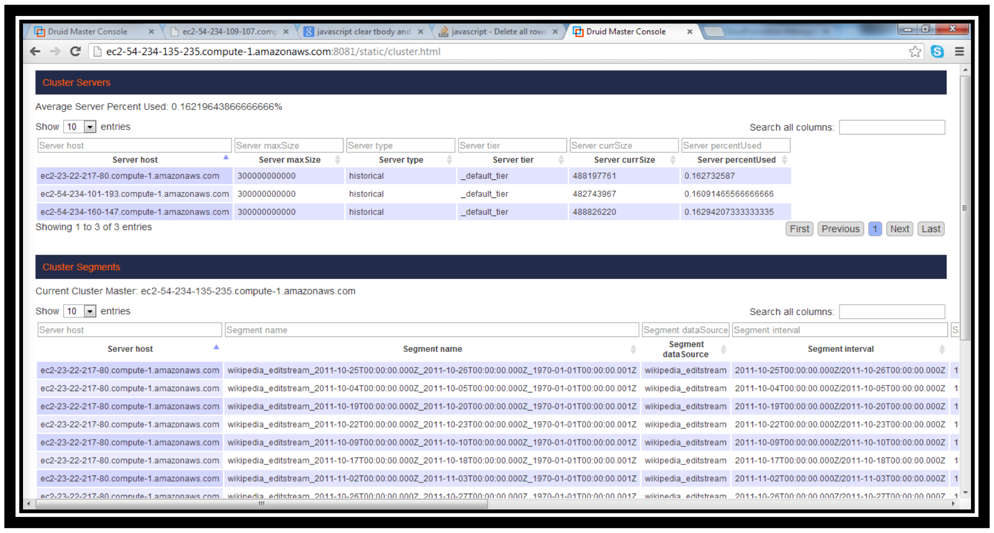
ec2-107-22-61-8.compute-1.amazonaws.com

<u>Figure 17</u>

7. Status of the Clusters

1. To see the status of clusters Use the mentioned URL http://IPAddressDruidMaster:8081/static/cluster.html

2. Replace IPAddressDruidMaster with the actual Druid Master IP Address


Druid Master Console
ec2-54-234-109-107.comp
javascript clear tbody and
javascript - Delete all rows
Druid Master Console
ec2-54-234-135-235.compute-1.amazonaws.com:8081/static/cluster.html

Cluster Servers

Average Server Percent Used: 0.16219643866666666%

Show 10 entries   Search all columns:

Server host	Server maxSize	Server type	Server tier	Server currSize	Server percentUsed
ec2-23-22-217-80.compute-1.amazonaws.com	300000000000	historical	_default_tier	488197761	0.162732587
ec2-54-234-101-193.compute-1.amazonaws.com	300000000000	historical	_default_tier	482743967	0.16091465566666666
ec2-54-234-160-147.compute-1.amazonaws.com	300000000000	historical	_default_tier	488826220	0.16294207333333335

Showing 1 to 3 of 3 entries   First Previous 1 Next Last

Cluster Segments

Current Cluster Master: ec2-54-234-135-235.compute-1.amazonaws.com

Show 10 entries   Search all columns:

Server host	Segment name	Segment dataSource	Segment interval	
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-10-25T00:00:00.000Z_2011-10-26T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-10-25T00:00:00.000Z/2011-10-26T00:00:00.000Z	1
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-10-04T00:00:00.000Z_2011-10-05T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-10-04T00:00:00.000Z/2011-10-05T00:00:00.000Z	1
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-10-19T00:00:00.000Z_2011-10-20T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-10-19T00:00:00.000Z/2011-10-20T00:00:00.000Z	1
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-10-22T00:00:00.000Z_2011-10-23T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-10-22T00:00:00.000Z/2011-10-23T00:00:00.000Z	1
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-10-09T00:00:00.000Z_2011-10-10T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-10-09T00:00:00.000Z/2011-10-10T00:00:00.000Z	1
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-10-17T00:00:00.000Z_2011-10-18T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-10-17T00:00:00.000Z/2011-10-18T00:00:00.000Z	1
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-11-02T00:00:00.000Z_2011-11-03T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-11-02T00:00:00.000Z/2011-11-03T00:00:00.000Z	1
ec2-23-22-217-80.compute-1.amazonaws.com	wikipedia_editstream_2011-10-26T00:00:00.000Z_2011-10-27T00:00:00.000Z_1970-01-01T00:00:00.001Z	wikipedia_editstream	2011-10-26T00:00:00.000Z/2011-10-27T00:00:00.000Z	1

Figure 18

8. Administrator Details

When Amazon launches an instance, template file can specify *user data*, which is available for all instances in the reservation to retrieve.

The user-data is available to the instance with a simple HTTP request at this URL:
http://169.254.169.254/latest/user-data

Amazon does not put this user-data on the instance itself, though the AMI may have code that causes the instance to download and process the user-data automatically. We have put code in the cloud formation template user data section to automate the process configuration, start and logging via user data. The instances use this data at launch time to ensure compute node connect to correct MySQL instance configured in master node.

Below section shows our user data snippet

```
"UserData": {
"Fn::Base64": {
        "Fn::Join": [
                "",
        [
"#!/bin/bash -v\n"
"# Helper function\n",
"dnsName=$(curl -s http://169.254.169.254/latest/meta-data/public-hostname)\n",
"sudosed    -i    \"s/ec2-23-20-118-187.compute-1.amazonaws.com/${dnsName}/g\"
/etc/mysql/my.cnf\n",
"sudo service mysql restart\n",
"cd /usr/share/zookeeper/bin\n",
"sudo ./zkServer.sh start\n",
"sudosed    -i    \"s/ec2-50-19-184-93.compute-1.amazonaws.com/${dnsName}/g\"
/home/ubuntu/index.html\n",
"cd /home/ubuntu/druid/druid-services\n",
"sudosed    -i    \"s/ec2-50-19-184-93.compute-1.amazonaws.com/${dnsName}/g\"
/home/ubuntu/druid/druid-services/master/runtime.properties\n",
"sudosed    -i    \"s/ec2-50-19-184-93.compute-1.amazonaws.com/${dnsName}/g\"
/home/ubuntu/druid/druid-services/broker/runtime.properties\n",
"nohup  java  -Duser.timezone=UTC  -Dfile.encoding=UTF-8  -cp  master:druid-services-0.2.0-
SNAPSHOT-selfcontained.jar  com.metamx.druid.http.MasterMain>nohup_masterLog  2>&1
&\n",
"nohup    java    -Duser.timezone=UTC    -Dfile.encoding=UTF-8    -
```

cpbroker:DruidSkilledAnalystsBroker.jarcom.skilledanalysts.druid.demo.SkilledAnalystsDemo Main>nohup_skilledAnalystsDemoLog 2>&1 &\n"

]

]

}

1. How things are laid out:

Zookeeper is kept under /usr/share/zookeeper. It is started with below highlighted section automatically using Userdata settings explained above.

"cd /usr/share/zookeeper/bin\n",
"sudo ./zkServer.sh start\n",

MySQL is located at /etc/mysql/

Using the Amazon utility URL first the DNS name of generated instance is replaced in my.cnf at /etc/mysql. Then the MySQL server is started prior to starting the druid node (master or compute). Below lines show this configuration in Userdata portion of Cloud Formation template file.

"sudosed -i \"s/ec2-23-20-118-187.compute-1.amazonaws.com/${dnsName}/g\" /etc/mysql/my.cnf\n",

"sudo service mysql restart\n",

Druid related files are kept under /home/ubuntu/druid/druid-services

Master is started with below highlighted section automatically using Userdata settings explained above. Master's DNS name (based on new AWS instance) is put into runtime properties before firing it up.

"cd /home/ubuntu/druid/druid-services\n",

"sudosed -i \"s/ec2-50-19-184-93.compute-1.amazonaws.com/${dnsName}/g\" /home/ubuntu/druid/druid-services/master/runtime.properties\n",

"nohup java -Duser.timezone=UTC -Dfile.encoding=UTF-8 -cp master:druid-services-0.2.0-SNAPSHOT-selfcontained.jar com.metamx.druid.http.MasterMain>nohup_masterLog 2>&1 &\n",

2. Details of how the processes are started up:

As explained above based on Userdata DNS name is picked and configuration managed using SED (Stream editor filter). Nohupis used to run the processes. Nohup is a POSIX command to ignore the HUP (hangup) signal. The HUP (hangup) signal is by convention the way a terminal warns depending processes of logout.

This way output that would normally go to the terminal goes to a file called nohup.out if it has not already been redirected.

- Logs are generated in /home/ubuntu/druid-services.
- Since Nohup is used in command line to start java process the logs for master are logged innohup_master log file,
- For compute its nohup_compute log file and
- For broker it is nohup_skilledanalystsdemo log file.

3. Code Location& Data fetch URL:

Jar is kept at /home/ubuntu/druid-services
There is a demo servlet which outputs index.html located at /home/Ubuntu/index.html containing basic query interface. This query interface generates a JSON, which is sent to broker, and results retrieved are shown back to user in a simple table.

Logs for looking at this process are kept in file nohup_skilledAnalystsDemoLog located at /home/ubuntu/druid-services. The demo application is bundled into broker jar and run at instance start by User data using the below highlighted text:

"nohup java -Duser.timezone=UTC -Dfile.encoding=UTF-8 -cpbroker:DruidSkilledAnalystsBroker.jarcom.skilledanalysts.druid.demo.SkilledAnalystsDemoMain>nohup_skilledAnalystsDemoLog 2>&1 &\n"

4. How to adjust configuration:

Druid utilizes various configuration parameters, which are located under respective application (master, broker, compute). More information on these properties is present at wiki (https://github.com/metamx/druid/wiki/Configuration). For master, this file is located in /home/Ubuntu/druid/druid-services/master/runtime.properties.

For compute node this file is located in /home/Ubuntu/druid/druid-services/compute/runtime.properties.

5. How to view log files:

Below are various log files generated by nohup (via userdata) to generate various druid components

- Master
  /home/ubuntu/druid/druid-services/nohup_masterLog

- Broker (and Demo Servlet- Fetch Query Interface)
  /home/ubuntu/druid/druid-services/nohup_skilledAnalystsDemoLog

- Compute
  /home/ubuntu/druid/druid-services/nohup_computeLog

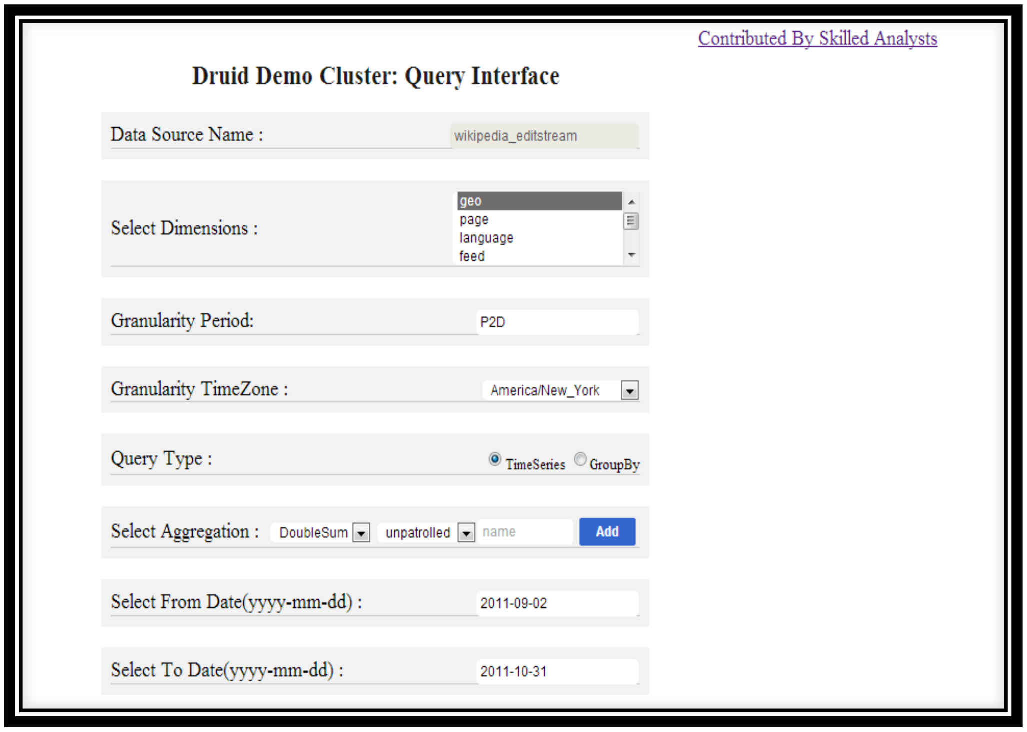
- Druid query logs
  /tmp/druid/log/DATE.log (where DATE represent current date)

Important Note:*In case of any error in query execution, always first check the logs of nohup_skilledAnalystsDemoLog. In case there is syntax error then it will be logged there. In case Druid startup has faced any issue (database not present) or zookeeper not allowing new connections, then it will be logged in nohup_masterLog log file. Compute node will log its operations and issues in nohup_computeLog.*

9. Data Query

1. For Data Query hit the mentioned URL: http://IPAddressDruidMaster:8082/druid/v3/demoServlet

2. Replace **IPAddressDruidMaster** with the actual druid master IP Address



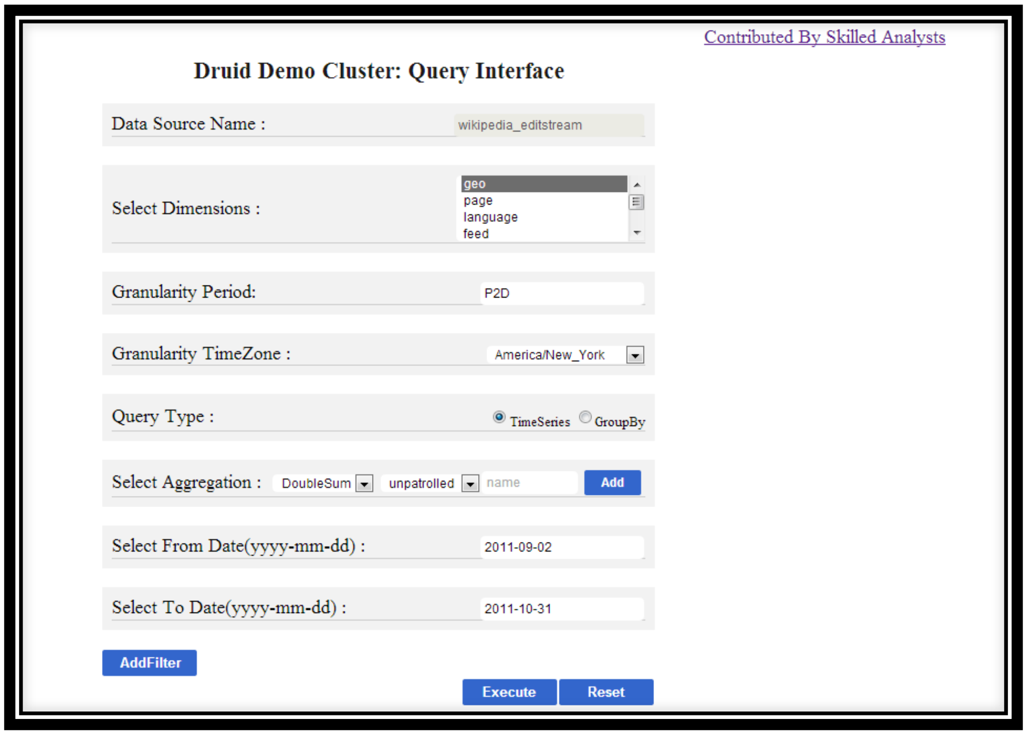
Contributed By Skilled Analysts

Druid Demo Cluster: Query Interface

Data Source Name :	wikipedia_editstream
Select Dimensions :	geo, page, language, feed
Granularity Period:	P2D
Granularity TimeZone :	America/New_York
Query Type :	[✓ TimeSeries] GroupBy
Select Aggregation :	DoubleSum, unpatrolled, name, Add
Select From Date(yyyy-mm-dd) :	2011-09-02
Select To Date(yyyy-mm-dd) :	2011-10-31

Figure 14

3. Browser will have prefilled values in Dimensions, granularity (which will bring result). Select Aggregation (Press add next to aggregation). Press "Execute" button.



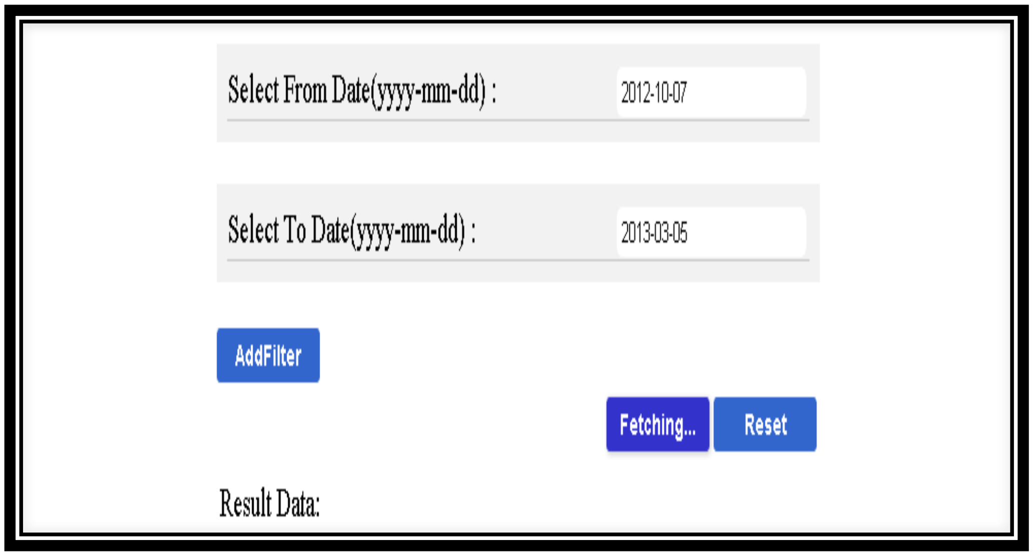
Contributed By Skilled Analysts

Druid Demo Cluster: Query Interface

Data Source Name :	wikipedia_editstream
Select Dimensions :	geo, page, language, feed
Granularity Period:	P2D
Granularity TimeZone :	America/New_York
Query Type :	[✓ TimeSeries] GroupBy
Select Aggregation :	DoubleSum, unpatrolled, name
Select From Date(yyyy-mm-dd) :	2011-09-02
Select To Date(yyyy-mm-dd) :	2011-10-31

Figure 15

4. It will show the Query is in progress by converting “Execute” into Fetching.

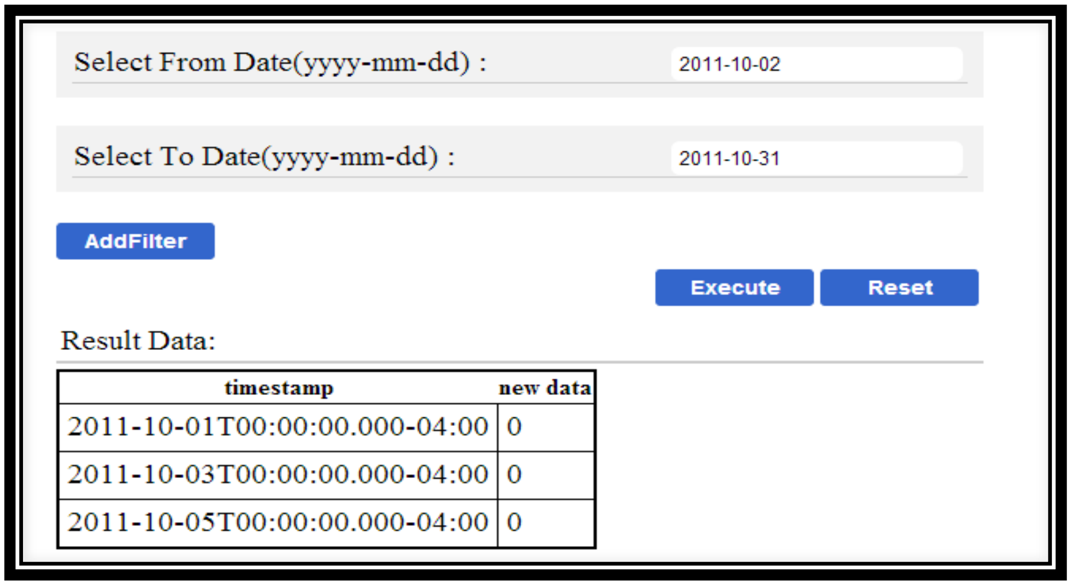


Select From Date(yyyy-mm-dd) :	2012-10-07
Select To Date(yyyy-mm-dd) :	2013-03-05

Result Data:

Figure 16

5. The results will be shown in the same page as shown below:



Select From Date(yyyy-mm-dd) : 2011-10-02

Select To Date(yyyy-mm-dd) : 2011-10-31

Result Data:

timestamp	new data
2011-10-01T00:00:00.000-04:00	0
2011-10-03T00:00:00.000-04:00	0
2011-10-05T00:00:00.000-04:00	0

Figure 17

10. Troubleshooting

<u>**Process Troubleshooting**</u>

- From AWS console, find out the instance IP address you will like to troubleshoot (master or compute)
- SSH is provided in security group for external usage for troubleshooting.
- In case you prefer to use remote desktop then add 3389 (or custom port as per network policy) in Druid Public policy attached with the instance.
- Once you have connected to machine, open command line (terminal) and type “psaux|grep java”. The command will show Druid processes and their id. It will also show zookeeper process id in case the command is executed at master.
- Similarly to find mysql process id, type “psaux|grepmysql”.
- In case you will like to stop/kill any process, type “kill -9 processid”
- Processes can be restarted using commands used in Userdata section (as described in chapter Administrator Details).
- For example to restart Zookeeper use below

  "cd /usr/share/zookeeper/bin"
  "sudo ./zkServer.sh start"

<u>**Query Troubleshooting**</u>

- In case data is not coming on to screen from the browser application, druid allows simple querying capabilities via curl.
- Open a terminal and type below:

  curl -X POST "http://ec2-67-202-13-229.compute-1.amazonaws.com:8082/druid/v2/?pretty" -H 'content-type: application/json' -d @query.json

- The above command assumes that a query.json is located in directory from where the command is invoked. Below is sample query.json, which is similar to what is produced by the browser application (you can copy paste and save in a file query.json in current directory and invoke curl request from there).

  {"queryType":"groupBy","dataSource":"wikipedia_editstream","granularity":{"type": "period", "period": "P2D", "timeZone": "America/Los_Angeles","origin": "2011-09-01T00:00:00-08:00"},"dimensions":["geo"],"aggregations":[{"type":"count","name": "rows"}],"intervals":["2011-09-01/2011-10-31"]}

- The above approach allows trying out all possibilities and options offered by Druid (post aggregations etc.), which might not be intuitive as offered by simple query browser.

11. Appendix

AWS Druid Template

A template is a declaration of the AWS resources that make up a stack. The template is stored as a text file whose format complies with the JavaScript Object Notation (JSON) standard.

The template allows infrastructure deployments that have different configuration values. Templates also provide output properties that can be used to easily communicate configuration or the results of deploying the template back to the user. All AWS resources in a template are identified using logical names, allowing multiple stacks to be created from a template without fear of naming collisions between AWS resources.

Below is section of Druid template used for current deployment:

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "This template will setup Druid with 1 master and 3 compute nodes. This template created Amazon EC2 Ubuntu 12.04 instances. You will be billed for the AWS resources used if you create a stack from this template. Also, you are solely responsible for complying with the license terms for the software downloaded and installed by this template. By creating a stack from this template, you are agreeing to such terms.",
"Parameters" : {
   "KeyPairName" : {
     "Description" : "Name of an existing Amazon EC2 key pair for ssh and rdp access",
   "Type" : "String"
   },
   "InstanceType" : {
     "Description" : "Amazon EC2 instance type",
     "Type" : "String",
     "Default" : "m1.xlarge",
     "AllowedValues" : [ "m1.xlarge"]
   }
  },
  "Resources" : {
            "DruidMaster" : {
                     "Type" : "AWS::EC2::Instance",
                     "Properties" : {
                     "ImageId" : "ami-6e71e907",
"InstanceType" : "m1.xlarge",
"KeyName" : { "Ref" : "KeyPairName" },
"SecurityGroups" : [ { "Ref" : "DruidPublicSecurity" },{ "Ref" : "DruidPrivateSecurity" } ],
"UserData": {
        "Fn::Base64": {
                "Fn::Join": ["", [
"dnsName=$(curl -s http://169.254.169.254/latest/meta-data/public-hostname)\n",
"sudosed    -i    \"s/ec2-23-20-118-187.compute-1.amazonaws.com/${dnsName}/g\"
/etc/mysql/my.cnf\n",
```

As shown above, template has description to help broadcast deployment plan, various EC2 instances that make this deployment and it contains User data. User data is set of meta-data, which is available to EC2 instance post initialization from AWS.

AWS CloudFormation gives developers and systems administrators an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion. It contains a collection of templates that illustrate various usage cases. Stacks can be created from the templates via the AWS Management Console, or also through the AWS CloudFormation command line tools.

You can use the templates as-is or you can use them as a starting point for creating your own templates.***In this document, we will use an existing template. User is not required to create any AMI manually. Template is used to create a stack (predefined set of instances based on template definition).***You can deploy Druid template and its associated collection of resources via the AWS Management Console, Cloud Formation command line tools or APIs.

- Version & URL to download Eclipse: Below link was used to download eclipse
  www.eclipse.org/downloads/

- Version & URL to download Druid: Below link was used to download Druid
  https://github.com/metamx/druid

Copyright – Druid

Engineering team of Skilled Analysts (www.skilledanalysts.com) built the AWS cloud formation template and query interface explained in the current document. Skilled Analysts with offices in North America, Germany and India provides analytic services for big data technologies.

About Metamarkets

Metamarkets is pioneering a new approach to big data analytics: data science-as-a-service. A new generation of mobile applications and cloud services are generating high volume, high velocity transaction streams — from impressions and actions, to purchases and payments. Analyzing these streams can unlock value by increasing revenue, improving user engagement, and enhancing operational awareness.

However, existing tools require complex integrations or simply don't scale. While data science is critical in overcoming these limitations, there aren't enough data scientists to keep pace with the rising demand for insight.

Metamarkets democratizes data science through scalable, smart and intuitive analytics. Metamarkets help data scientists deliver powerful dashboards to business users and accelerate their own discovery efforts.

For additional information please visit *http://metamarkets.com/*