

ENGR 6971 PROJECT REPORT

Deployment of an auto scaling online web
services in cloud environment

Deep Pandya
Student ID: 7162391

Contents

ABSTRACT	2
INTRODUCTION	2
ENGINEERING APPROACH.....	3
MYSQL CLUSTER ON AMAZON WEB SERVICES	3
AMAZON AURORA	4
Steps to create a DB cluster and configure Amazon Aurora:	5
To connect to a DB Cluster Instance:.....	8
Pros and cons to use Amazon Aurora:.....	9
PLUGINS TO CONFIGURE MYSQL CLUSTER ON AMAZON WEB SERVICES	9
NORMAL CONFIGURATION USING SHELL SCRIPTS TO DEPLOY MYSQL ON AWS	10
Basic Idea to set up MySQL cluster on Amazon EC2:	10
What is Amazon Virtual Private Cloud?.....	11
Benefits of using a VPC:.....	11
Steps to set up virtual Private cloud according to project requirement:	12
Steps to set up MySQL cluster on Amazon EC2:	13
AMAZON OPSWORK	16
Steps to configure AWS OpsWorks to deploy Dell DVD store application:	17
DELL DVD STORE APPLICATION.....	18
Steps to setup Dell DVD store application on MySQL Database Engine:	19
NETFLIX ASGARD	20
What is Auto-Scaling?	20
Benefits of Auto scaling.....	21
Two Distinct ways to deploy new code through Asgard:.....	22
Steps to use Netflix Asgard to set up Auto scaling Groups for the Dell DVD store Application:	23
Create an application and security group:.....	24
Create a Cluster	24
RESULT AND CONCLUSION.....	26
REFERENCES	30

ABSTRACT

The project report refers to work completed during my project work with Dr. Yan Liu in Summer-2 2015.

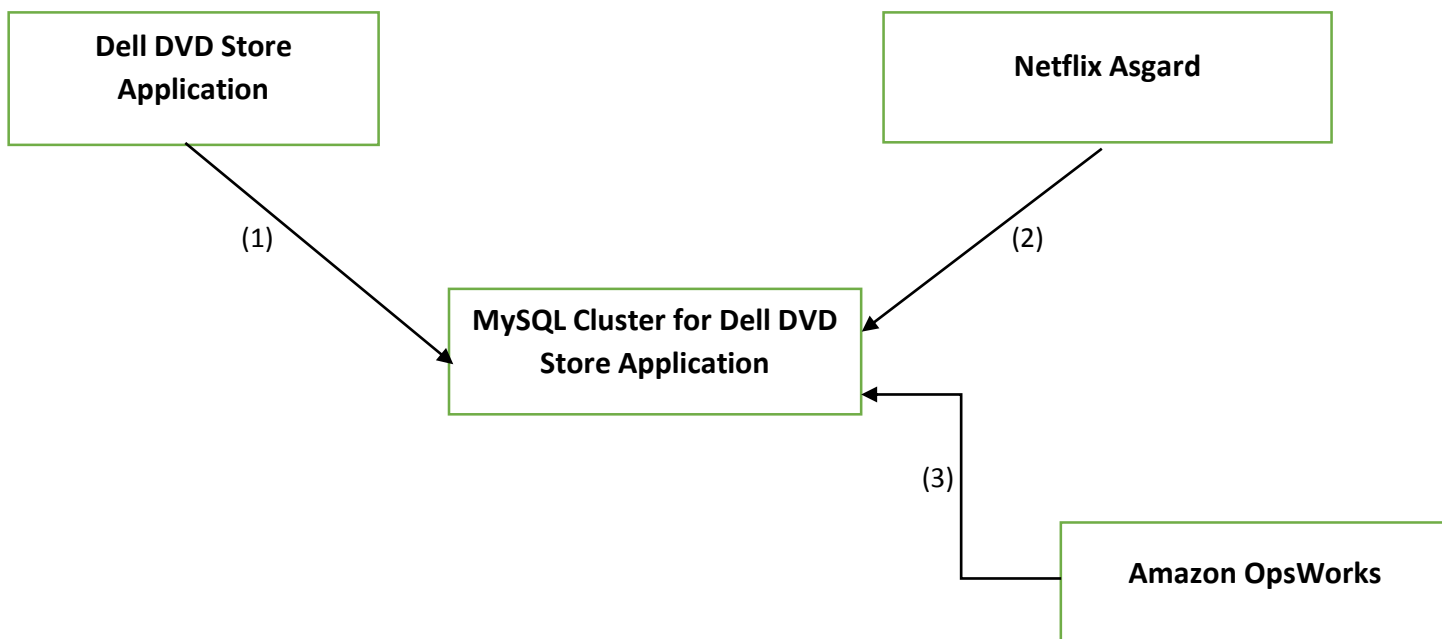
My main objective during the project work was to deploy an auto scaling online web services in cloud services in Amazon Web Services. According to the project definition, I should be able to deploy Dell DVD Store application on Amazon Web Services which is used to check the performance of the database instance configuration and the infrastructure.

Netflix has created an open source library named Netflix Asgard for application deployment and cloud management. As per the project requirement, I used Netflix Asgard library to configure the auto scaling groups for the Dell DVD Store Application on MySQL Cluster.

After creating auto scaling groups, I used AWS Opsworks to automate deployment process of the Dell DVD Application on AWS. AWS Opsworks is an application management service to deploy and operate applications of all size.

INTRODUCTION

Objective: To Deploy Dell DVD Store Application on Amazon EC2 using an auto scaling technic in cloud environment.



Dell DVD Store Application is benchmark tool to test performance and scalability of different database engines. So main objective of the project is to setup MySQL cluster on Amazon Ec2 on which Dell DVD Store Application can be deployed.

- (1) Set Up MySQL cluster on deploy Dell DVD Store Application on created MySQL cluster on Amazon EC2
- (2) Use Netflix Asgard Library to create an auto scaling group for the management node of the MySQL cluster.
- (3) Use Amazon OpsWorks script to automate the deployment of Dell DVD Store Application.

ENGINEERING APPROACH

For the project, I have worked on different modules.

- [MySQL Cluster on Amazon Web Services](#) : Describes what MySQL Cluster is and why we need to use it with Amazon Web Service and how to configure and set up everything to run MySQL Cluster on AWS.
- [Amazon OpsWorks](#) : Describes what AWS OpsWorks is and why we should use to automate the process of deployment of an application. It also shows how to use AWS OpsWorks according to project application.
- [Dell DVD Store Application](#) : Describes what Dell DVD Store Application and how to set up it on MySQL Cluster.
- [Netflix Asgard](#) : Describes what Netflix Asgard library is and why we should use it to create auto scaling group for the application. It also gives you an information to set up everything for Netflix Asgard.

MYSQL CLUSTER ON AMAZON WEB SERVICES

MySQL cluster provide shared-nothing clustering and auto-sharding for the MySQL database management system. It also provides high availability and high throughput with low latency, while allowing near linear scalability. NDB (Network DataBase) or NDBCLUSTER storage engine for MySQL is used to implement MySQL Cluster.

MySQL is the most popular database in the cloud. BY using MySQL Database for Amazon EC2, Developers can deliver fully backed, cost efficient web-scale database application in the cloud. Amazon EC2 and MySQL are a great fit for organizations wanting to reduce the capital expenditures and operating costs and at the same time dynamically scale their applications with minimal cost and effort.

There are three ways to configure MySQL cluster in Amazon Web Services.

1. Normal Configuration using shell scripts to deploy MySQL on AWS.
2. By using Third Party Plugins like several nines Galera Cluster, Start Cluster, etc.
3. Amazon Aurora

AMAZON AURORA

- Amazon Aurora is a MySQL-compatible, relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases.
- It provides up to five times better performance than standard MySQL.
- It is designed to offer greater than 99.9% availability, replicating 6 copies of data across 3 Availability Zones and backing up data to Amazon S3.

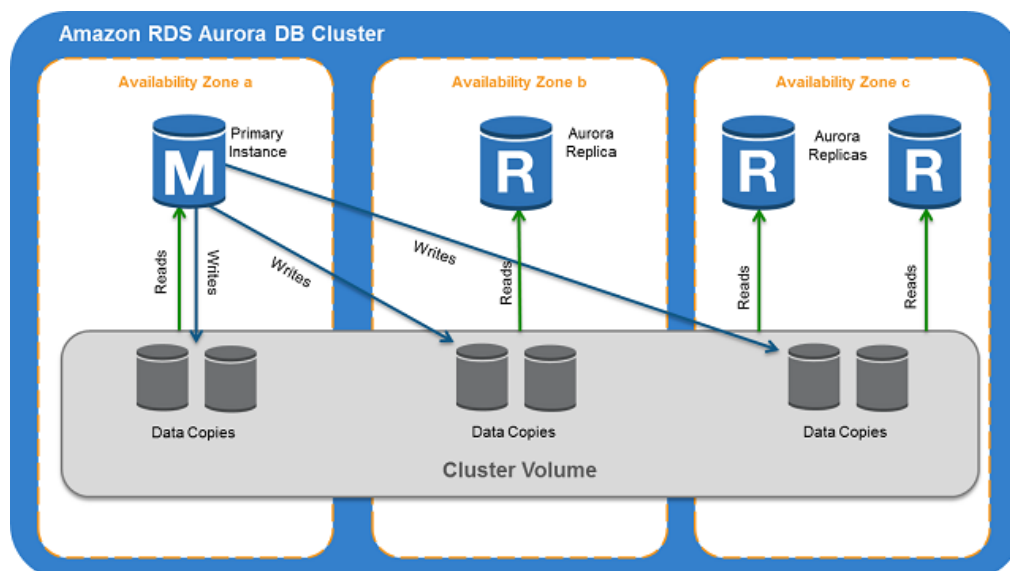
Amazon Aurora Instance = DB cluster Instance

A DB Cluster includes one or more instances and a cluster volume to manage data for instances. An Aurora cluster volume is a virtual database storage volume which spans multiple availability Zones. Each Zone has a copy of the cluster data.

Aurora DB Cluster consists two types of instances:

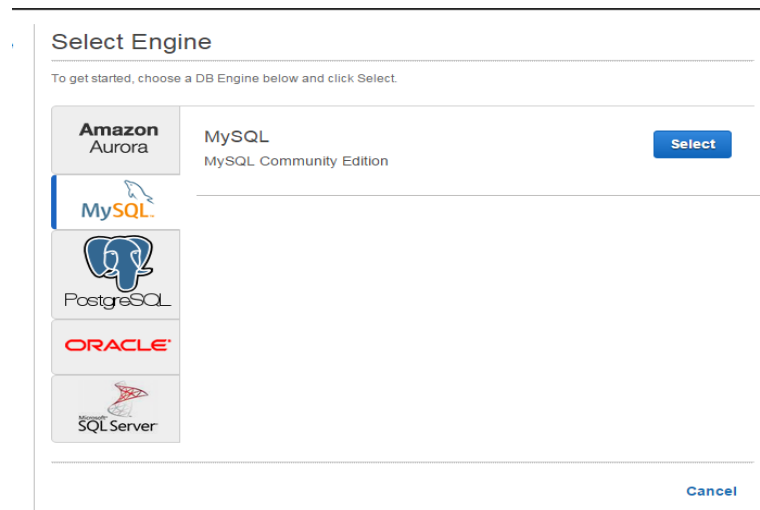
1. Primary Instance: Supports read-write blocks and perform data modifications to the cluster data.
2. Aurora Replica: Supports only read operations.

Each Db instance can have one primary instance and up to 15 Aurora Replicas.



Steps to create a DB cluster and configure Amazon Aurora:

1. Open the Amazon RDS for Aurora console at <https://console.aws.amazon.com/rds>.
2. In the top-right corner of the AWS Management Console, select the region that you want to create your DB cluster in. This example uses the US East (N. Virginia) region. Amazon Aurora is only available in the US East (N. Virginia), US West (Oregon), or EU (Ireland) regions.
3. In the left navigation pane, click DB Instances.
4. Click Launch DB Instance to start the Launch DB Instance Wizard. The wizard opens on the Select Engine page.
5. On the Select Engine page, click the Select button for the Aurora DB engine.



6. Set the following values on the Specify DB Details page as shown in figure :

Specify DB Details

Instance Specifications

DB Engine	mysql
License Model	<input type="text" value="general-public-license"/>
DB Engine Version	<input type="text" value="5.6.23"/>

Review the [Known Issues/Limitations](#) to learn about potential compatibility issues with specific database versions.

Version number of the database engine to be used for this instance.

DB Instance Class	<input type="text" value="db.m3.xlarge — 4 vCPU, 15 GiB RAM"/>
Multi-AZ Deployment	<input type="text" value="Yes"/>
Storage Type	<input type="text" value="Provisioned IOPS (SSD)"/>
Allocated Storage*	<input type="text" value="100"/> GB
Provisioned IOPS	<input type="text" value="1000"/>

Settings

DB Instance Identifier*	<input type="text" value="Test"/>
Master Username*	<input type="text" value="deeppandya"/>
Master Password*	<input type="password" value="....."/>
Confirm Password*	<input type="password" value="....."/>

* Required

Cancel

- Click Next and set the following values on the Configure Advanced Settings page as shown in below figure :

Configure Advanced Settings

Network & Security

This instance will be created with the new Certificate Authority rds-ca-2015. If you are using SSL to connect to this instance, you should use the [new certificate bundle](#). [Learn more here](#)

VPC*

Subnet Group

Publicly Accessible

Availability Zone

VPC Security Group(s)

Database Options

Database Name

Note: If no database name is specified then no initial MySQL database will be created on the DB Instance.

Database Port

DB Parameter Group

DB Cluster Parameter Group

Option Group

Copy Tags To Snapshots ☐

Enable Encryption

Backup

Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to detail [here](#).

Backup Retention Period days

Backup Window

Maintenance

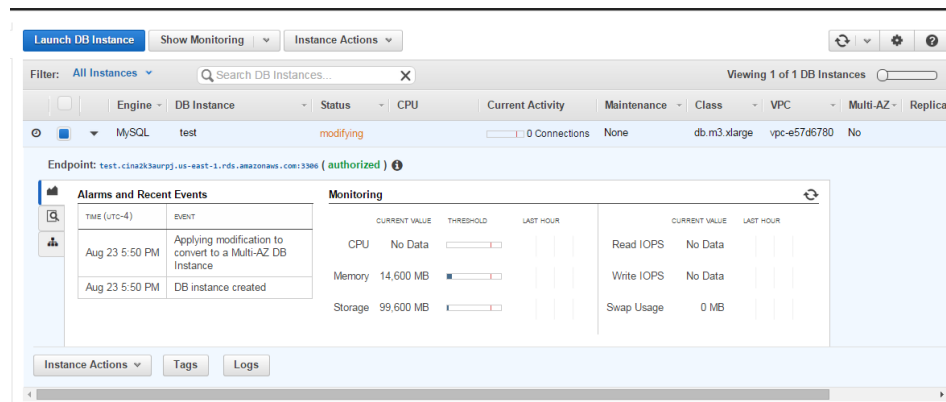
Auto Minor Version Upgrade

Maintenance Window

* Required

Specify a string of up to 8 alpha-numeric characters that define the name given to a database that Amazon RDS creates when it creates the DB Instance, as in 'mydb'. If you do not specify a database name, Amazon RDS does not create a database when it creates the DB Instance.

- Leave the rest of the values as their defaults, and click Launch DB Instance to create the DB cluster and primary instance.



To connect to a DB Cluster Instance:

1. Open the Amazon RDS for Aurora console at <https://console.aws.amazon.com/rds>.
2. Select Instances and click the arrow icon to show the DB cluster details. On the details page, copy the value for the endpoint. This endpoint is the cluster endpoint.
3. Type the following command at a command prompt on a client computer to connect to a database on a DB cluster using the MySQL monitor. Use the cluster endpoint to connect to the primary instance, and the master user name and password that you created previously. If you supplied a port value other than 3306, use that for the -P parameter instead.

```
PROMPT> mysql -h <endpoint> -P 3306 -u <mymasteruser> -p <password>
```

You will see output similar to the following.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 350
Server version: 5.1.32-log MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Pros and cons to use Amazon Aurora:

- Amazon Aurora is created by Amazon developers which is recognised by the industry.
- It is easy to create and use MySQL cluster.
- It provides Fast, Durable, Highly scalable and highly secure solution to the MySQL cluster.
- It provides perfect solution for the high-end commercial databases. But it's costly to use for small databases.
- Currently Amazon Aurora is in preview state so as a developer, if anyone wants to use it he/she has to ask for the special permission.

After setting up everything for the Amazon Aurora in AWS, Follow the instruction in section [DELL DVD STORE APPLICATION](#) to set up Dell DVD Store application in Amazon Aurora since Aurora provides only MySQL cluster for the project objective.

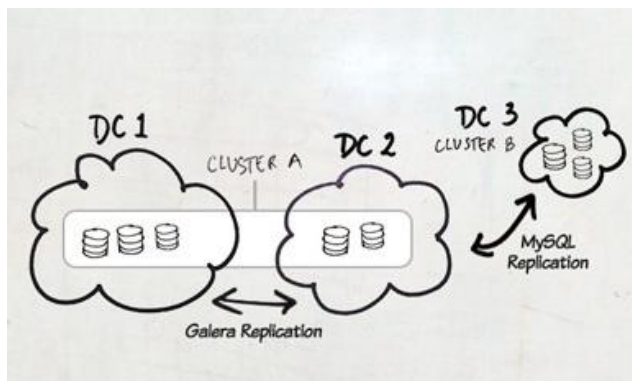
See section [RESULT AND CONCLUSION](#) to check the created MySQL cluster up and running.

Note: by using Amazon aurora, you can create a MySQL cluster but to set up Dell DVD Store Application on Aurora, you have to download a dataset of Dell DVD Store Application from open source resource <https://s3-us-west-2.amazonaws.com/selection-tasks/ds2-small.zip> or you can generate your own dataset and set up the Dell DVD Store Application.

PLUGINS TO CONFIGURE MYSQL CLUSTER ON AMAZON WEB SERVICES

For this project, I have tried Two Cluster Plugins.

1. Galera Cluster for MySQL on Amazon VPC developed by severalnines.
 - Galera cluster for MySQL provides Multimaster Cluster based on synchronous replication.
 - Galera cluster provides an easy-to-use, high-availability solution with high system uptime, no data loss and scalability for the growth.

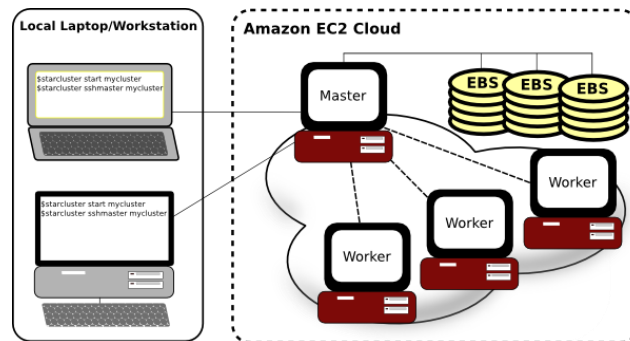


Check below link to know more about Galera Cluster,

<http://www.severalnines.com/deploying-galera-cluster-mysql-amazon-vpc>

2. Star Cluster

- StarCluster is an open source cluster-computing toolkit for Amazon's Elastic Compute Cloud (EC2).
- StarCluster has been designed to automate and simplify the process of building, configuring, and managing clusters of virtual machines on Amazon's EC2 cloud.
- StarCluster allows anyone to easily create a cluster computing environment in the cloud suited for distributed and parallel computing applications and systems.



Check below link to know more about Star Cluster,

<https://www.youtube.com/watch?v=2Ym7epCYnSk>

NORMAL CONFIGURATION USING SHELL SCRIPTS TO DEPLOY MYSQL ON AWS

For the project, I have used EC2 windows instances. Because Dell DVD store Application in MySQL cluster works better in windows platform and it provides more functionality.

Basic Idea to set up MySQL cluster on Amazon EC2:

I have defined one Virtual Private Cloud in Amazon environment, which includes one public subnet and one private subnet.

For MySQL cluster on AWS EC2, It should have at least one management node which manages the cluster and one or more data nodes to store the data.

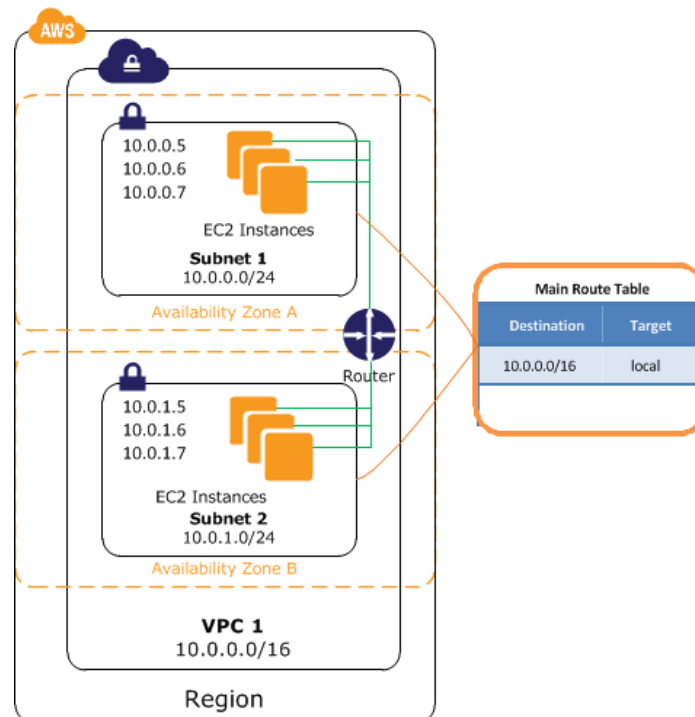
For the project, I have used one windows instance as a management node which resides in public facing subnet. And two windows instances as data nodes which reside in Private subnet and they can be accessed by public subnet node only.

What is Amazon Virtual Private Cloud?

Amazon virtual private cloud provides a logically isolated section of amazon web services cloud where AWS resources can be launched in a private network.

The complete control is provided over the virtual networking environment, including section of the range of IP address, configuring route tables and network gateways and creation of the subnets.

A subnet is a range of IP addresses in the VPC. Amazon VPC can be configured to use a public subnet for resources that must be connected to the Internet, and a private subnet for resources that won't be connected to the Internet (I have used the same configuration to deploy MySQL cluster on EC2).



Benefits of using a VPC:

- Run the instances on single-tenant hardware
- Assign static private IP addresses to the instances that persist across starts and stops
- Change security group membership for the instances while they're running

- Assign multiple IP addresses to the instances
- Define network interfaces, and attach one or more network interfaces to the instances
- Control the outbound traffic from the instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to the instances in the form of network access control lists (ACL)

Check below link to know more about Amazon Virtual Private Subnet,

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Introduction.html

Steps to set up virtual Private cloud according to project requirement:

1. Use Amazon's VPC wizard to create VPC that has a public and private subnet.
2. Go into the VPC console dashboard and verify the region that is desired for the VPC and click on the select button.

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via a Network Address Translation (NAT) instance in the public subnet. (Hourly charges for NAT instances apply.)

Select

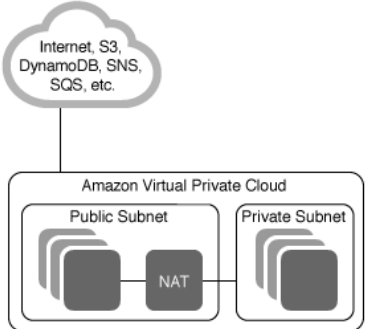
Internet, S3, DynamoDB, SNS, SQS, etc.

Amazon Virtual Private Cloud

Public Subnet

Private Subnet

NAT



3. From the next screen, configure IP CIDR block, public subnet mask and private subnet mask as shown in below figure and press on Create VPC.

Step 2: VPC with Public and Private Subnets

IP CIDR block: (507 IP addresses available)

VPC name:

Public subnet: (59 IP addresses available)

Availability Zone:

Public subnet name:

Private subnet: (11 IP addresses available)

Availability Zone:

Private subnet name:

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT instance:

Instance type:

Key pair name:

Note: Instance rates apply. [View Rates](#).

Add endpoints for S3 to your subnets

Subnet:

Enable DNS hostnames: ☒ Yes ☐ No

Hardware tenancy:

[Cancel and Exit](#) [Back](#) [Create VPC](#)

- After the previous, it will create 1 VPC (another one for the default configuration), 2 subnets, 1 internet gateway and 2 route tables.

[Start VPC Wizard](#) [Launch EC2 Instances](#)

Note: Your Instances will launch in the US West (N. California) region.

You are using the following Amazon VPC resources in the US West (N. California) region:

2 VPCs	2 Internet Gateways
4 Subnets	3 Route Tables
2 Network ACLs	1 Elastic IP
19 Security Groups	1 Running Instance
0 VPC Peering Connections	0 Virtual Private Gateways
0 VPN Connections	0 Customer Gateways
0 Endpoints	

Steps to set up MySQL cluster on Amazon EC2:

- After creating Virtual Private Cloud as shown above. Start defining different windows instances from EC2. One windows instance for management node and it should have one elastic IP address so that it can be accessed from the outside world and it should be created in Public subnet. And define two Windows Instances for Data Nodes and they should be created in private subnet which includes private IP addresses for both of them.

Check below link on how to create an instance in EC2,

http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/EC2Win_GetStarted.html

<div> <div>Launch Instance</div> <div>Connect</div> <div>Actions</div> </div> <div> <div>Filter by tags and attributes or search by keyword</div> <div>K</div> </div>											
<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
<input type="checkbox"/>		i-16d25ad8	t2.micro	us-west-1a	running	Initializing	None			keypair	disabled
<input type="checkbox"/>		i-16d25ad8	t2.micro	us-west-1a	running	Initializing	None			keypair	disabled
<input type="checkbox"/>		i-454e0187	t2.micro	us-west-1c	terminated		None			starcluster	disabled
<input type="checkbox"/>	test	i-4b1fee9e	t2.micro	us-west-1c	terminated		None			keypair	disabled
<input type="checkbox"/>		i-5bd579b0	m1.small	us-west-1a	running	2/2 checks ...	None	ec2-52-8-247-84.us-wes...	52.8.247.84	keypair	disabled
<input type="checkbox"/>		i-afd35b0f	t2.micro	us-west-1a	running	Initializing	None			keypair	disabled

- At the end of step 1, it will have 3 windows instances, one with elastic IP from public subnet and which is used as a management node. And two other windows Instances as a data nodes.
- To configure MySQL cluster on VPC, First step is to download and install MySQL software on each of the windows machine. For the project, I have used zip version of the software.
- Extract the contents of the Zip archive to C:\mysqlcluster directory.
- Keep the configuration and data files separate from the software. Create a new folder MySQL_Cluster\My_Cluster in the home directory and in there create conf and data folders.
- Within the data folder on management node, create a sub-folder for each of the two MySQL Server (mysqld) processes and then copy c:\mysqlcluster\data\mysql to each – this creates the MySQL database for each of the MySQL nodes containing data such as privileges and password.
- To get access to the executables from the command line, add c:\mysqlcluster\bin to the PATH environment variable.
- Create a config.ini file within the conf folder that was created on the host to run the management node (ndb_mgmd).

And add below text content in the config.ini file,

```
[ndbd default]
Noofreplicas=2
datadir=C:\Documents\MySQL_Cluster\My_Cluster\data
```

```
[ndbd]
hostname=First Data node Private IP
id=2
```

```
[ndbd]
hostname=Second Data node private IP
id=3
```

```
[ndb_mgmd]
id=1
hostname=Management node IP
```

```
mysqld]
id=101
hostname= Management node IP

[mysqld]
id=102
hostname= Management node IP
```

9. Create a cnf file for each of the MySQL server data in conf folder.

First.cnf

```
[mysqld]
ndb-nodeid=101
ndbcluster
datadir=C:\Documents\MySQL_Cluster\My_Cluster\data\mysqld_1
port=3306
ndb-connectstring=Management Node IP with port 1186
```

Second.cnf

```
[mysqld]
ndb-nodeid=102
ndbcluster
datadir= C:\Documents\MySQL_Cluster\My_Cluster\data\mysqld_2
port=3307
ndb-connectstring= Management Node IP with port 1186
```

10. The first process to start is the management node. Run the following command from the management node,

```
ndb_mgmd --initial -f conf/config.ini --configdir=./conf
```

11. Next, start the data nodes on Data node1 and Data node 2:

```
On Data node 1 => ndbd -c management node with IP --initial
```

```
On Data node 1 => ndbd -c management node with IP --initial
```

12. And then finally start the MySQL Server nodes on Management node:

```
mysqld --defaults-file=conf\First.cnf
mysqld --defaults-file=conf\Second.cnf
```


13. Finally, check that all of the nodes are up and running:

```
On management Node => ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @Data node 1 (mysql-5.1.39 ndb-7.0.9, Nodegroup: 0, Master)
id=3 @Data node 2 (mysql-5.1.39 ndb-7.0.9, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @Management node (mysql-5.1.39 ndb-7.0.9)

[mysqld(API)] 2 node(s)
id=101 @Management node (mysql-5.1.39 ndb-7.0.9)
id=102 @Management node (mysql-5.1.39 ndb-7.0.9)
```

By following the above steps, MySQL cluster will be up and running for the Dell DVD Store Application.

AMAZON OPSWORK

- AWS OpsWorks is an application management service that makes it easy to deploy and operate applications of all shapes and sizes.
- AWS OpsWorks includes automation to scale your application based on time or load and dynamic configuration to orchestrate changes as your environment scales.
- To manage the applications over their lifetime with the support for auto healing and automatic instance scaling, AWS OpsWorks provides dynamic configuration and orchestration.
- AWS OpsWorks allows reproduction of the software configuration on new instances and apply changes to all running instances.
- AWS OpsWorks automates operational tasks like package installations, software configurations, code deployment and database setups n any Linux servers.

Steps to configure AWS OpsWorks to deploy Dell DVD store application:

- Select Register existing instances to select MySQL Cluster EC2 image to deploy using AWS OpsWorks

Start Fresh

Create an OpsWorks stack to model your application components, deploy code, automate tasks, and control the resources used by your application.

[Add Your First Stack](#)

Register Existing Instances

Use OpsWorks with existing **EC2 instances**, as well as **on-premises servers** for an easy way to install packages, run scripts, manage operating system users, and more.

[Register Instances](#)

- Select EC2 Instances

Step 1: Choose Instance Type

Choose the type of instance you want to register. You can start with one instance and register more later. Once a computing resource is registered, it can be managed along with other AWS OpsWorks resources. [Learn more.](#)



EC2 Instances

Register existing EC2 instances that you created using the Amazon EC2 console, CLI or API.



On-premises Instances

Register on-premises instances running on your own hardware such as servers and virtual machines.

[Cancel](#) [Next: Select Instances](#)

- Select Running EC2 instance from the given option of the selected region.

Step 2: Select Instances

Create Stack

An OpsWorks stack is a way to group your instances and related resources. Create a stack in order to be able to register your instances. [Learn more.](#)

Name	<input type="text" value="test"/>	
Region	<input type="text" value="US West (N. California)"/>	Select a region for instances created in your stack.
VPC	<input type="text" value="vpc-aaf12fcf (default)"/>	
IAM role	<input type="text" value="aws-opsworks-service-role"/>	

Select Instances to Register

Before registering an instance, please ensure that it is compatible with AWS OpsWorks. [Learn more.](#)

<input type="text" value="Search"/>						
<input type="checkbox"/>	Instance ID	Name	Instance Type	Availability Zone	Instance State	Public IP
<input type="checkbox"/>	i-454e6187	–	t2.micro	us-west-1c	stopped	
<input checked="" type="checkbox"/>	i-4b1fee8e	test	t2.micro	us-west-1c	running	52.8.221.21

[Cancel](#) [Previous: Choose Instance Type](#) [Next: Install AWS CLI](#)

- Follow the instruction on the page and click on the register instances

Step 3: Install AWS CLI

OpsWorks uses an agent to manage instances. You must install a current version of the AWS CLI on the computer on which you will run the register command to install the OpsWorks agent. You can run the register command directly on the instance, or from a remote host.

For more information about installing and configuring the AWS CLI, see [Installing the AWS Command Line Interface](#).

Installation type

- ☒ Run the register command from a separate workstation
- ☐ Run the command from the instance to be imported

AWS CLI Installation

- ☒ I have installed the AWS CLI tools on the machine that will run the command

Set up user credentials

The register command must have appropriate permissions. [Learn more](#) about which permissions are required and how to configure the CLI tools to use them.

[Cancel](#)

[Previous: Select Instances](#)

[Next: Register Instances](#)

- Press Done to register instance.

Step 4: Register Instances

The register command installs the OpsWorks agent on an instance and registers the instance with the stack. Registered instances show up in the list below. [What's going to happen to my server?](#)

Run this command

- ☒ I use SSH keys to connect to my instances. [Learn more](#).

```
1 aws opsworks register --infrastructure-class ec2 --region us-east-1 --stack-id f20f90a3-989e-44f6-b21c-62c049a2294e --ssh-username [username] --ssh-private-key [key-path] i-4b1fee8e
```

Registered Instances (1)

Instances will start appearing in the list once you run the register command.

[Previous: Install AWS CLI](#)

[Done](#)

DELL DVD STORE APPLICATION

- Dell Engineers have created a synthetic benchmark tool to evaluate the performance and scalability of the different types of database engines like SQL Server, Oracle, MySQL and PostgreSQL.
- The application generates a benchmark value in the form of orders placed per test cycle and tests the performance of the instance configuration and the infrastructure underneath it.

- The application includes a back-end database component, a Web application and driver programmes.
- The synthetic workload dataset can be created with an included tool and load created dataset into the instance via a new test database (DS2) and then run an included load testing tool to simulate orders of DVDs off of a Web Store.
- The main goal to create Dell DVD store application is to provide advanced database features such as transactions, stored procedures, triggers and referential integrity.
- The DS2 database schema consists eight tables. It includes implementations of indexes, primary keys, foreign keys, and stored procedures.
- The driver program and Web tier implement a multistep transactional commit or rollback, depending on success or failure of multiple conditions.
- The database can be three sizes: 10MB, 1GB or 100GB. A 10MB database size is too small. Due to locking contention, 1GB database size does not allow to scale well.

Steps to setup Dell DVD store application on MySQL Database Engine:

1. Connect to the Already created Amazon EC2 server and follow the below instructions.
2. Download the two pre-requisite test bundles. A base bundle for all database engines from <http://linux.dell.com/dvdstore/ds21.tar.gz> and the extra bundle for MySQL database located at http://linux.dell.com/dvdstore/ds21_mysql.tar.gz.
3. Create a folder called C:\ds2and extract both tar files to the same directory.
4. It will create a file structure which includes Install_DVDStore.pl file. This Perl file is used to generate different Test databases with different size. But to run Perl file, download Active State's Active Perl tool.
5. Open a command prompt and go to the C:\ds2 directory and run Install_DVDStore.pl script. Enter the database size, enter the measure of space and enter the database platform (here MySQL) and the platform.
6. Enter the location where the new database objects will be placed on the system. Use C:\ds2\DatabaseFiles\
7. Press enter to start the process. It will begin to generate the CSV dataset files.
8. The process will take some time to finish and it will generate SQL script based on the selected configuration.
9. Run the SQL script. It will create the new DS2 database in MySQL and load the CSV workload files with bulk insert commands.

By following above steps, The Dataset is installed for the Dell DVD Store application.

To Test the performance of the MySQL cluster on EC2,

- Create a batch file in the location C:\DS2\mysqlds2 directory. And add below line to the batch file.

```
c:\ds2\mysqlds2\ds2mysqldriver --target=EC2 instance target --run_time=60 --db_size=2GB --n_threads=4 --ramp_rate=10 --pct_newcustomers=0 --warmup_time=0 --think_time=0.085 > testresults.txt 2>&1
```

- Change the target if it's intend to run this script across a network.
- Set your database size according to the configured workload files.
- Adjust the number of threads to set the expected load.
- The think_time setting is the amount of time that a simulated user would 'think' before clicking again.
- The output of the process is stored to a testresults.txt file.
- Run the recently created batch file to check the performance of the MySQL cluster on EC2 instance. Check testresults.txt file for the output.

NETFLIX ASGARD

- Asgard is a web interface for application deployments and cloud management in amazon Web Services.
- Asgard is created by Netflix to make it easier to work with Amazon web Services and It is an open source application and available on GitHub.
- Asgard removes a lot of the cryptic command line tools required for several AWS features.
- Asgard provides log files about the changes that are being made to the AWS environment.
- Asgard is available as a self-contained war and a standalone jar file.

Check below link to get latest binary of the Netflix Asgard,

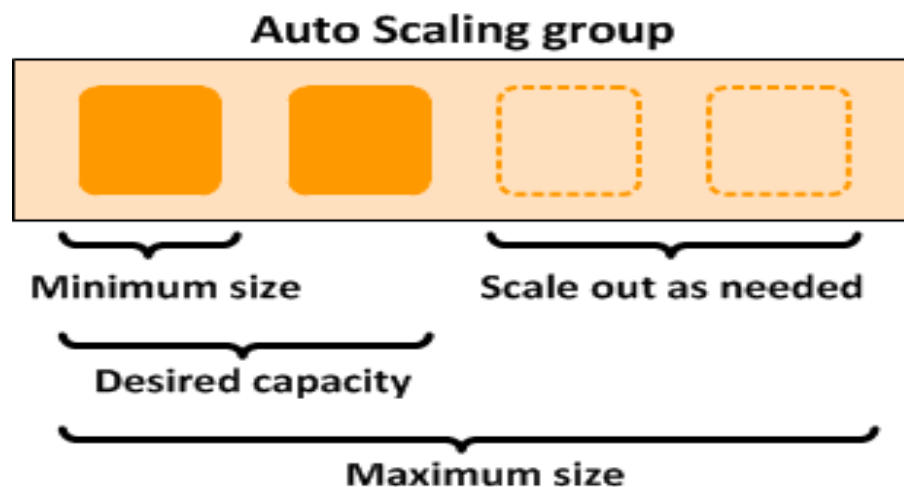
<https://github.com/Netflix/asgard/releases>

Before starting anything with the Netflix Asgard,

What is Auto-Scaling?

- Auto scaling is an automation service provided by the Amazon.
- Auto scaling provides features to manage a running pool of servers, including the capability to replace failed instances and automatically grow and shrink resource pool.
- Auto scaling helps to ensure that the number of EC2 instances available to handle the load the application.

- The collection of EC2 instances can be created, called Auto scaling groups. The minimum number of instances can be specified in each Auto scaling group and Auto scaling ensures that the group never goes below the minimum number of instance.
- The maximum number of instances in each group can also be specified and Auto scaling ensures that the group never goes above the given size.
- The desired capacity can be specified, either at the time of group creation or at any time thereafter, Auto scaling ensures that the group has that many instances.
- The scaling policies can be specified and Auto scaling can launch or terminate instances as demand on the application's load increases or decreases.



Benefits of Auto scaling

Adding Auto Scaling to the application architecture is one way to maximize the benefits of the AWS cloud.

When Auto scaling is used, the applications gain the following benefits:

- Better fault tolerance. Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it.
- Better availability. Auto Scaling can be configured to use multiple Availability Zones. If one Availability Zone becomes unavailable, Auto Scaling can launch instances in another one to compensate.
- Better cost management. Auto Scaling can dynamically increase and decrease capacity as needed.

Now let's get back to Netflix Asgard,

- Netflix Asgard provides is used to set up Configurations and Auto scaling groups for the AWS.
- Netflix Asgard provides two additional abstractions on top of AWS – Applications and Clusters.

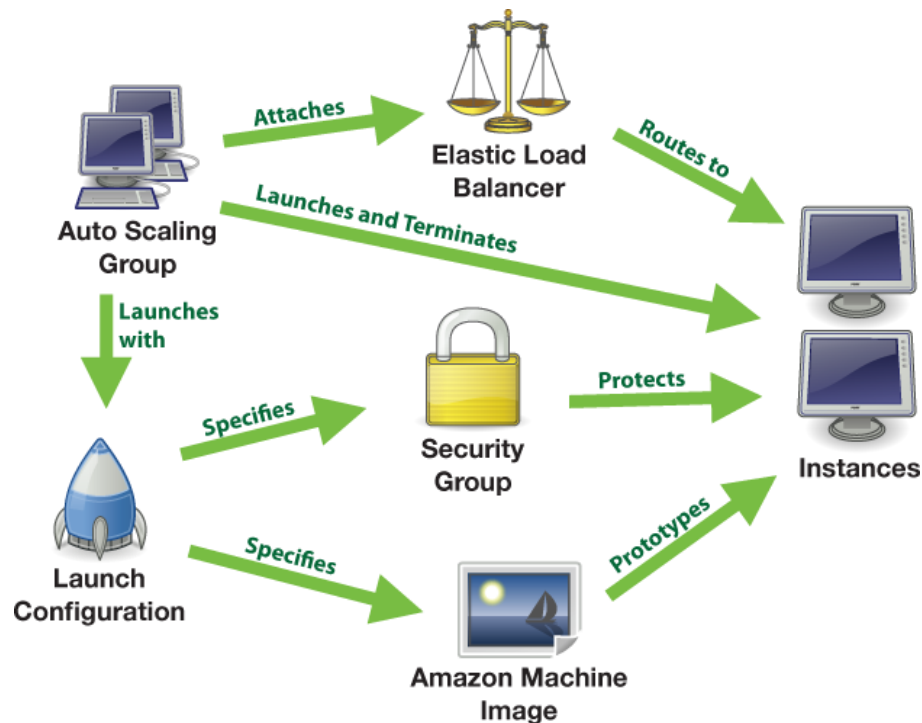
- One or more Clusters, some Auto scaling Groups, a Launch Configuration, an Elastic Load Balancer, Security groups, Some Ec2 instances and An AMI are included in an application.
- Each application contains an email address and an owner to connect the objects and the person responsible for creating and managing the applications.
- The Cluster of an application contains one or more Auto scaling Groups. The incremented version numbers are assigned by Asgard to newly created Auto scaling Groups.

Two Distinct ways to deploy new code through Asgard:

- The rolling push deployment model launches instances with new code and deleting and replacing old instances one or two at a time.
- The cluster based deployment model launches a new instance and starts to route traffic to it through an Elastic Load Balancer. The old cluster is disabled but remains available in case of rollback.

Both models depends on the AWS components that they are using should be dynamic and can be created programmatically.

Asgard provides a simple GUI for setting up and managing Auto scaling Groups.



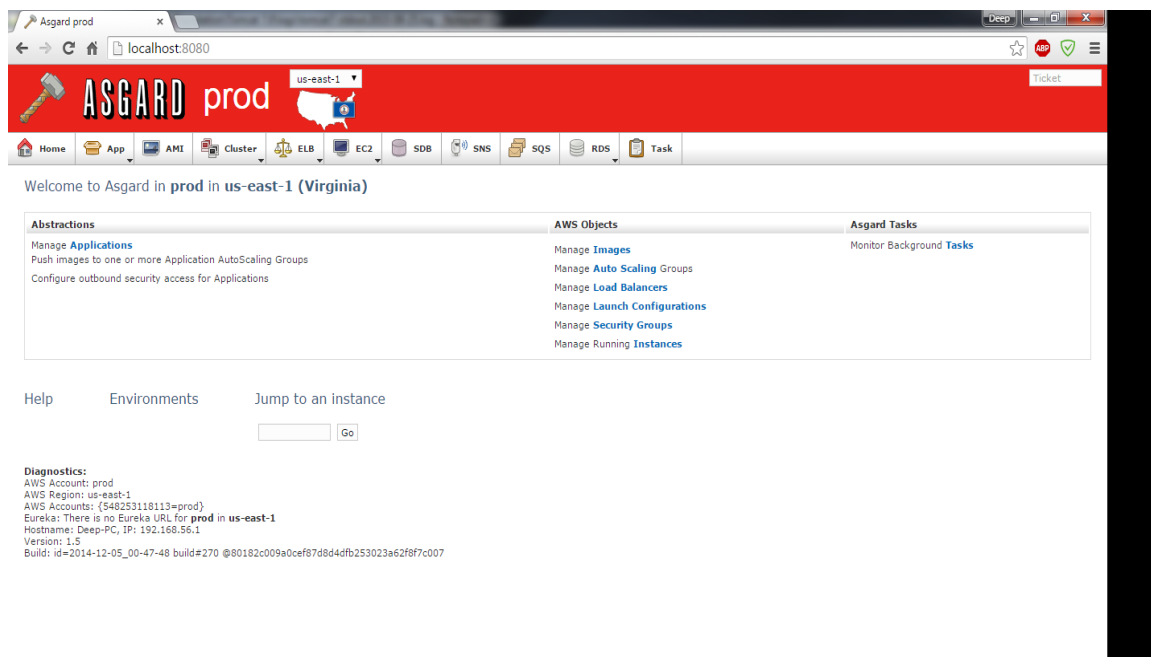
- Zero or more Elastic Load Balancers (ELBs) to new instances can be attached to An Auto Scaling Group (ASG).
- An Elastic Load Balancer can send user traffic to instances.
- An Auto scaling Group can launch and terminate instances.

- An Auto scaling Group uses a Launch Configuration for each instance launch.
- The Launch Configuration specifies which Amazon Machine Image (AMI) and which Security Groups to use when launching an instance.
- The AMI contains all the bits that will be on each instance, including the operating system, common infrastructure such as Apache and Tomcat, and a specific version of a specific Application.
- The traffic sources and ports to the instances can be restricted by the Security Groups.

Steps to use Netflix Asgard to set up Auto scaling Groups for the Dell DVD store Application:

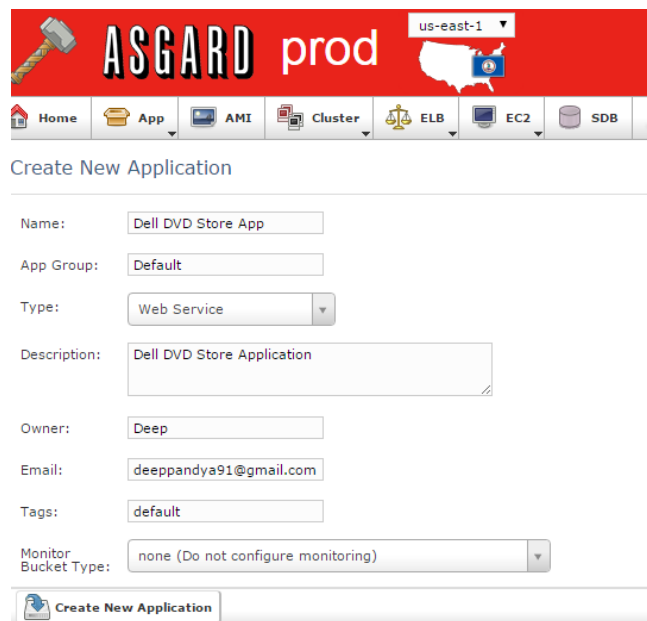
For the project I have used self-contained war file with tomcat server.

- For the first step create an AWS key pair, set that key pair path in the Environment variables.
- Set java heap memory `-Xmx1024M -XX:MaxPermSize=128m` in tomcat configuration. Follow the steps from below link to set up war file in tomcat server, <https://github.com/Netflix/asgard/wiki/Tomcat-configuration>
- Start the service in Apache Tomcat.
Open a browser and go to <http://localhost:8080> and you will have below screen,



Create an application and security group:

1. Click App
2. Click Create New Application
 - Name: **DelIDVDStoreApplication**
 - Description: Dell DVD Store application
 - Owner: Deep
 - Email: deeppandya91@gmail.com
 - Don't change dropdown menu
3. Submit the form by clicking Create New Application at the bottom



The screenshot shows the 'Create New Application' form in the AWS Management Console. The header bar is red with the 'ASGARD prod' logo and a 'us-east-1' region dropdown. The navigation bar includes links for Home, App, AMI, Cluster, ELB, EC2, and SDB. The form fields are as follows:

- Name: Dell DVD Store App
- App Group: Default
- Type: Web Service (dropdown menu)
- Description: Dell DVD Store Application
- Owner: Deep
- Email: deeppandya91@gmail.com
- Tags: default
- Monitor Bucket Type: none (Do not configure monitoring) (dropdown menu)

At the bottom of the form is a 'Create New Application' button with a blue icon.

4. On the Application Details screen, click Create Security Group

Create a Cluster

1. Click Cluster
2. Click **Create New Auto Scaling Group**
3. Select the **DelIDVDStoreApplication** application
4. Scroll down to the **Launch Configuration** section

5. Change the AMI Image ID to the AMI that of the MySQL Cluster to deploy as your **DellDVDStoreApplication** example.
6. Select the **DellDVDStoreApplication** security group
7. Submit the form by clicking **Create New Auto Scaling Group** at the bottom

The screenshot shows the 'Create New Auto Scaling Group' page in the AWS Management Console. The top navigation bar includes 'Home', 'App', 'AMI', 'Cluster', 'ELB', 'EC2', 'SDB', 'SNS', 'SQS', 'RDS', and 'Task'. The 'App' menu is expanded, showing 'Applications', 'Stacks', 'Owners', and 'Security Groups'. The 'Name' field is 'delldvd'. The 'Optional v1' section includes 'Env', 'Req', and 'Selection of task property values'. The 'Name Preview' section shows 'delldvd'. The 'Stack' section has a dropdown for 'delldvd' and a 'New Stack' button. The 'Free-form details' section includes 'Countries' (Ex: latam), 'Dev phase' (Ex: stage), 'Hardware devices' (Ex: phones), 'Partners' (Ex: sony), and 'Revision' (Ex: 27). The 'Auto Scaling' section includes 'Instance Bounds' (Min: , Max:), 'Desired Capacity' (instances), 'Cooldown' (10 seconds), 'ASG Health Check Type' (EC2 (Replace terminated instances)), 'ASG Health Check Grace Period' (600 seconds), 'Termination Policy' (Default), and 'Availability Zones' (us-east-1a, us-east-1b, us-east-1d, us-east-1e). The 'AZ Rebalancing' section has radio buttons for 'Keep Zones Balanced' (selected) and 'Don't Rebalance Zones'. The 'Load Balancers' section states 'There are no load balancers in this account-region'. The 'Launch Configuration' section includes 'AMI Image ID' (Image Id), 'Instance Type' (m3.medium \$48.24/mo), 'SSH Key' (), 'Security Groups' (Disable (more reliable) selected, Enable (reduced costs) unselected), 'Kernel ID' (), 'Ram Disk ID' (), and 'IAM Instance Profile' (). The 'Create New Auto Scaling Group' button is at the bottom.

By Following Above Steps, Auto Scaling Group can be configured for the MySQL cluster for Dell DVD Store Application.

Note: Netflix library can be used with any Amazon Memory image. And Amazon Aurora creates Amazon Memory Image for MySQL cluster so Netflix library can be used with Amazon Aurora. But for the project I have used Netflix Asgard with MySQL cluster using EC2 instances.

RESULT AND CONCLUSION

Dell DVD Store Application on MySQL Cluster

After setting up everything according to section [DELL DVD STORE APPLICATION](#),

Start MySQL Monitor in the management node of the MySQL Cluster.

If you check result.txt file from log file that you have mentioned at the time of set up, you'll have below results,

Total number of Threads to be Spawned across multiple servers are n_threads: 32

target= db2 n_threads= 32 ramp rate= 10 runtime= 3 db_size= 2GB warmup_time= 0 thinktime= 0.085

pct_newcustomers= 0 n_searches= 3 search_batch_size= 5 nlineitems= 5 virtdir= ds2 page type= php

Using WIN32 QueryPerformanceCounters for measuring response time

Performance Counter Category Test and Counter MaxRT exist

Thread0: createdforUser 0

Thread1: createdforUser 1

Thread2: createdforUser 2

Thread19:createdforUser 19

Thread3: createdforUser 3

Thread6: createdforUser 6

Thread4: createdforUser 4

Thread5: createdforUser 5

Thread7: createdforUser 7

Thread8: createdforUser 8

Thread9: createdforUser 9

Thread11:createdforUser 11

Thread20:createdforUser 20

Thread12:createdforUser 12

Thread13:createdforUser 13

Thread16:createdforUser 16

Thread10:createdforUser 10

Thread14:createdforUser 14

Thread17:createdforUser 17

Thread15:createdforUser 15

Thread18:createdforUser 18

Thread22:createdforUser 22

Thread23:createdforUser 23

Thread21:createdforUser 21

Thread24:createdforUser 24

Thread25:createdforUser 25

Thread26:createdforUser 26

Thread27:createdforUser 27

Thread28:createdforUser 28

Thread29:createdforUser 29

Thread30:createdforUser 30

Thread31:createdfor User31

Controller (7/24/2013 5:51:53 AM): all threads running

Controller: n_threads_connected = 0 : ConnectionTimeOut remaining 60

Thread 29: connected to db2

Thread 15: connected to db2

Controller: n_threads_connected = 2 : ConnectionTimeOut remaining 59

Thread 3: connected to db2

Thread 20: connected to db2

Thread 1: connected to db2

et= 110.0 n_overall=24868 opm=13564msec=162 rt_tot_avg_msec=55 rt_tot_sampled=37 rollbacks: n=508 %= 2.0

et= 120.0 n_overall=27437 opm=13718 msec=292 rt_tot_avg_msec=53 rt_tot_sampled=37 rollbacks:
n=569 %= 2.1

et= 130.0 n_overall=30066 opm=13876 msec=204 rt_tot_avg_msec=51 rt_tot_sampled=35 rollbacks:
n=623 %= 2.1

et= 140.0 n_overall=31999 opm=13713 msec=232 rt_tot_avg_msec=53 rt_tot_sampled=78 rollbacks:
n=658 %= 2.1

et= 150.0 n_overall=34583 opm=13832 msec=161 rt_tot_avg_msec=52 rt_tot_sampled=37 rollbacks:
n=709 %= 2.1

et= 160.0 n_overall=36793 opm=13797 msec=289 rt_tot_avg_msec=52 rt_tot_sampled=57 rollbacks:
n=755 %= 2.1

et= 170.0 n_overall=39183 opm=13828 msec=98 rt_tot_avg_msec=52 rt_tot_sampled=49 rollbacks: n=810
%= 2.1

et= 180.0 n_overall=41885 opm=13961 rt_tot_lastn_maxmsec=125 rt_tot_avg_msec=51 rt_tot_sampled=32 rollbacks:
n=860 %= 2.1

Final (7/24/2013 5:55:05 AM): et= 180.0 n_overall=41885 opm=13961 rt_tot_lastn_max=125 rt_tot_avg=51
n_login_overall=41885

Thread 0: exiting

Thread 13: exiting

Thread 8: exiting

Thread 18: exiting

Thread 15: exiting

Thread 1: exiting

Thread 5: exiting

Thread 20: exiting

Thread 21: exiting

Thread 12: exiting

Thread 25: exiting

Thread 31: exiting

Thread 10: exiting

Thread 29: exiting

Thread 4: exiting

Thread 3: exiting

Thread 14: exiting

Thread 24: exiting

Thread 11: exiting

Thread 22: exiting

Thread 23: exiting

Thread 16: exiting

Thread 2: exiting

Thread 28: exiting

Thread 9: exiting

Thread 7: exiting

Thread 17: exiting

Thread 6: exiting

Thread 30: exiting

Thread 26: exiting

Thread 27: exiting

Thread 19: exiting

Controller (7/24/2013 5:55:05 AM): all threads stopped, exiting

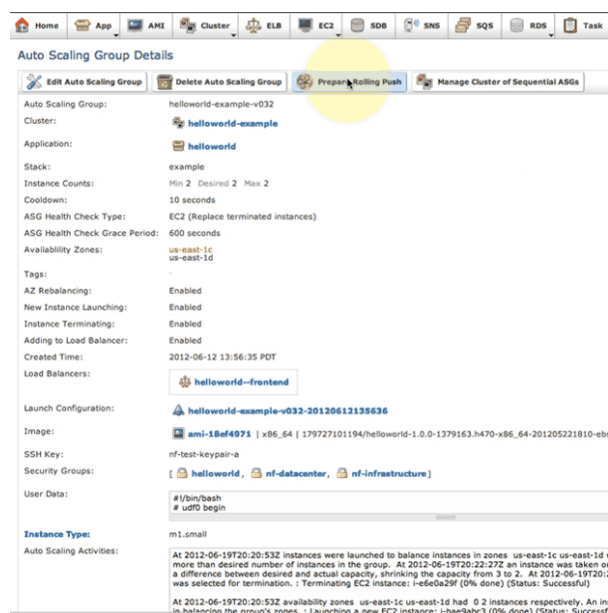
n_purchase_from start= 41891 n_rollbacks_from start= 860

Run over.

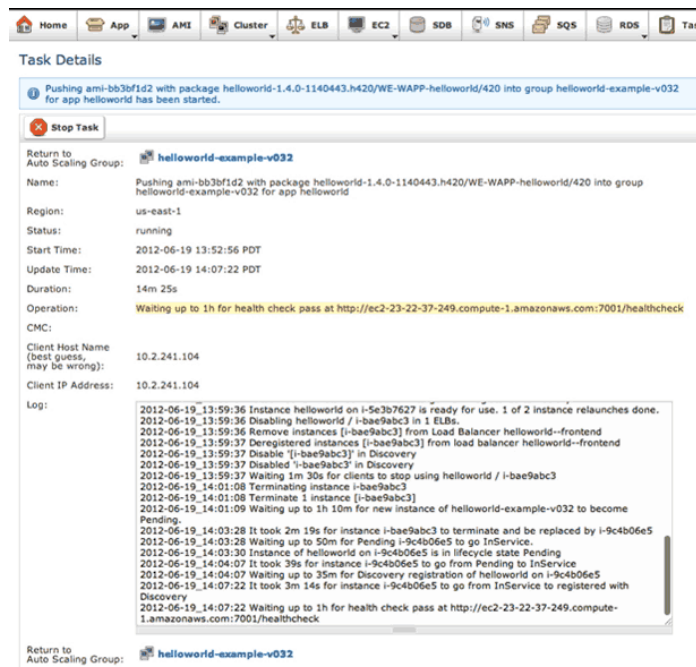
Netflix Asgard

After setting up everything from section [NETFLIX ASGARD](#),

If you want to see auto scaling group settings then go to your desired region and select autoscaling option and you'll have all the information as below,



And if you want to see application's task information then go to task details section for the particular application and you'll have below screen,



So to check results in Dell DVD Store Application, all you have to do is to check the result file that you have mentioned at the time of configuration and to check Netflix Asgard, you have to redirect to particular option for the specific application and you'll have detailed information regarding your application.

REFERENCES

- I. Netflix Asgard
<https://github.com/Netflix/asgard>
<http://techblog.netflix.com/2012/06/asgard-web-based-cloud-management-and.html>
- II. Amazon Aurora – A MySQL compatible, relational database engine
<https://aws.amazon.com/rds/aurora/>
- III. Running MySQL Cluster over multiple Windows Servers.
<http://www.clusterdb.com/mysql-cluster/running-mysql-cluster-over-multiple-servers>
- IV. Amazon OpsWorks
<https://aws.amazon.com/opsworks/>

V. Dell DVD Store Application

<http://linux.dell.com/dvdstore/>

https://www.youtube.com/watch?v=eng7_sG89KA

<http://www.davidklee.net/2013/07/29/benchmark-your-sql-server-instance-with-dvdstore/>

VI. Galera Cluster

<http://www.severalnines.com/galera-configurator/>

VII. Start Cluster

<http://star.mit.edu/cluster/>

<https://www.youtube.com/watch?v=tC2GuuUA5Ak>