AWS Services

Template details	
AWS Services	AWS_Services2.0

1 Contents

1. EC2 MONITORING AND MANAGEMENT

Amazon Elastic Compute Cloud (Amazon **EC2**) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

1.1 Objective

To provide the high level guidance's to setup the AWS EC2 Monitoring and Management on AWS Cloud.

1.2 Assumptions

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

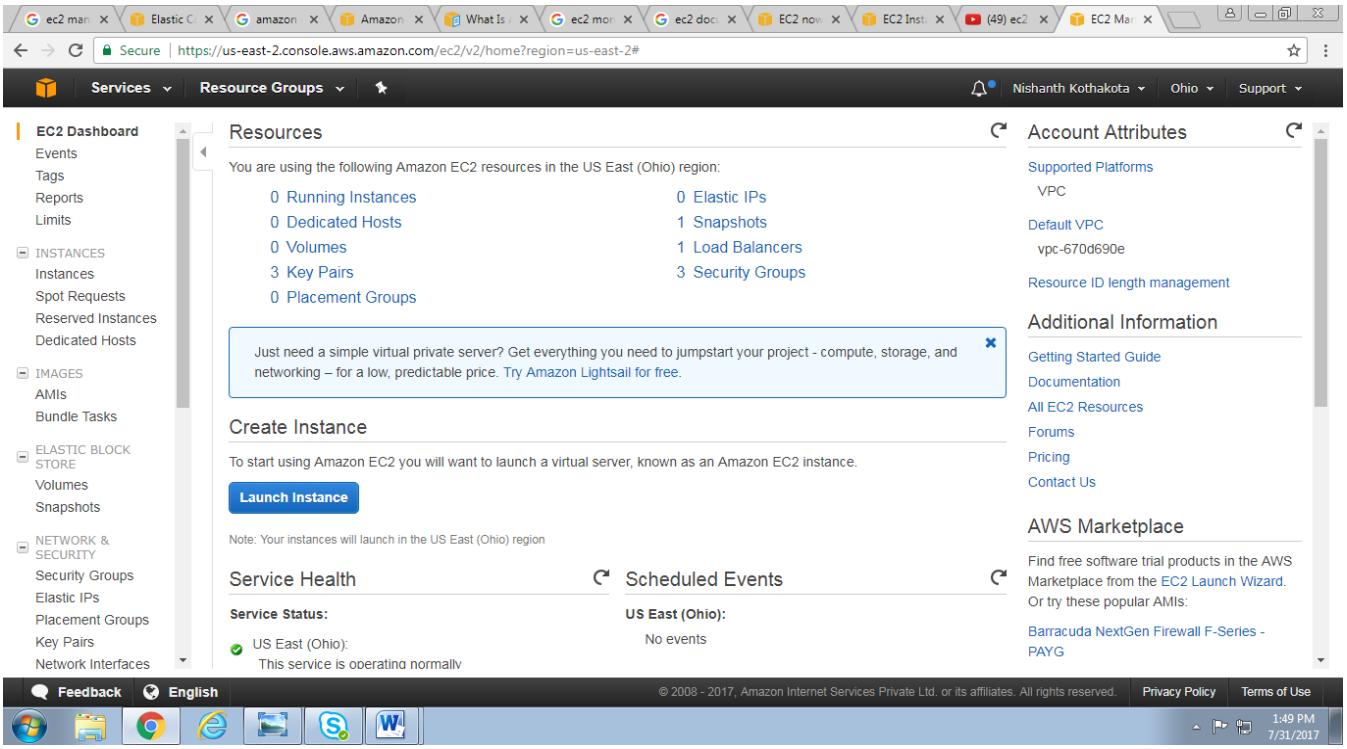
1.3 Procedure

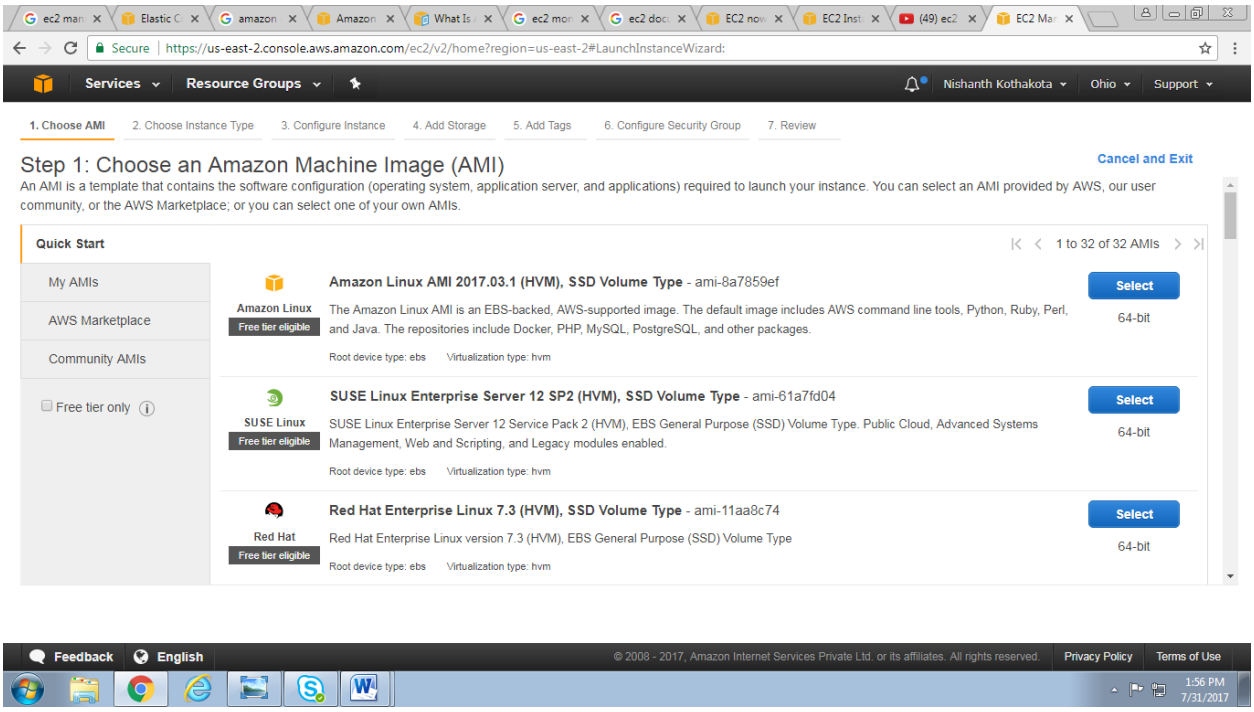
To launch the EC2 instance, you need to complete the following steps:

Step 1: Login to your AWS account
Step 2: Choose an Amazon Machine Images(AMI) from Quick start
Step 3: Select Instance type
Step 4: Configure the all instance details
Step 5: Add storage to your instance
Step 6: Create Sectuity group to your instance

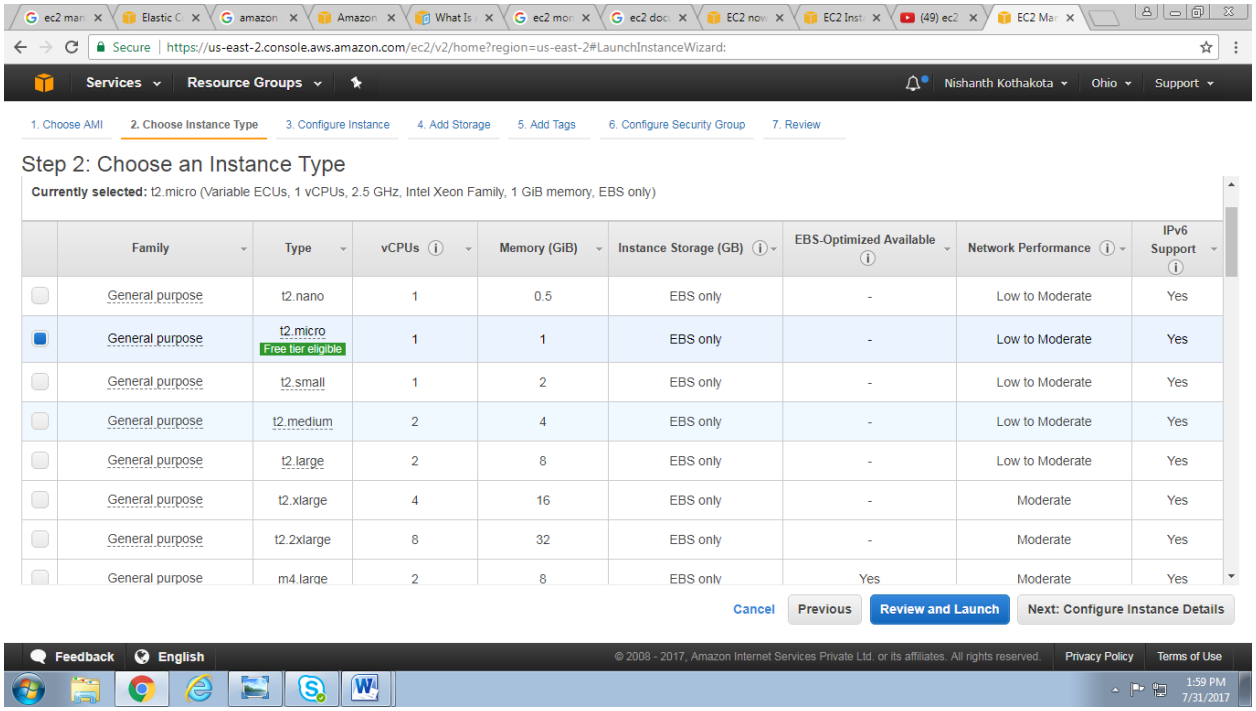
1.3.1 Create EC2 Instance

- Go to the EC2 Dashboard and click on Launch Instance.
- Select Quick start instance – Amazon Linux [Note: As required you can select MyAMI's / Marketplace / community AMI]

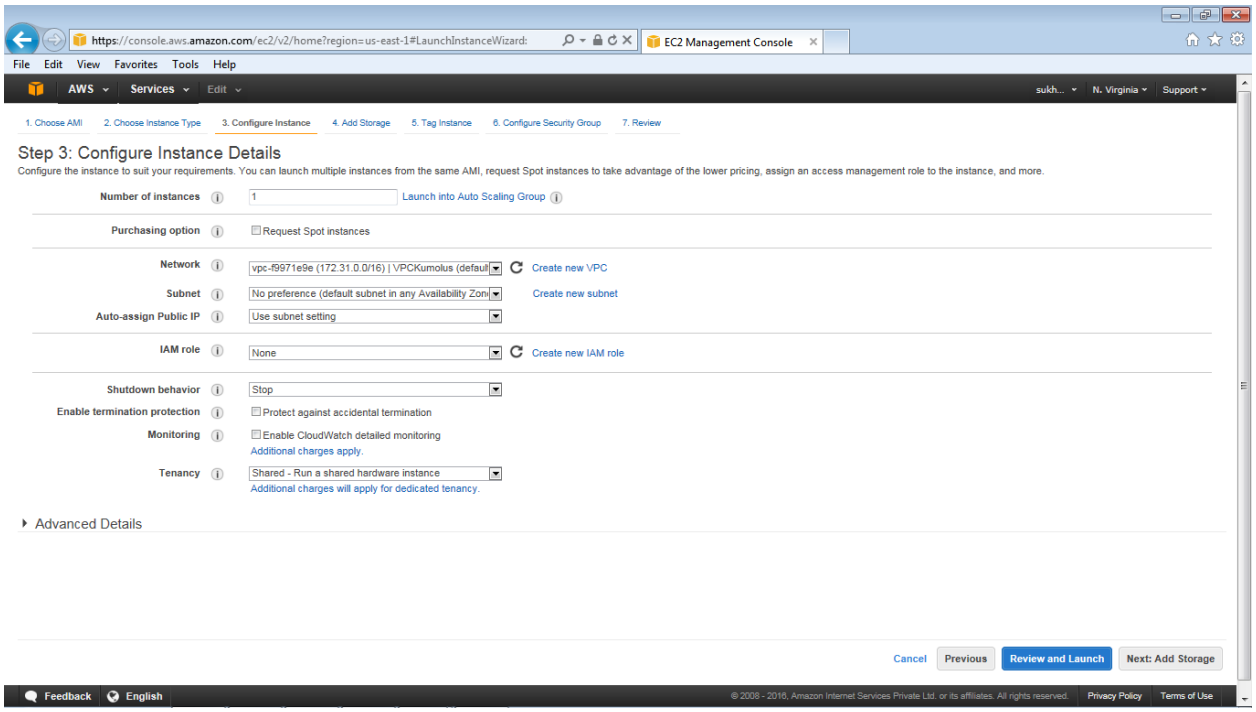

Secure | https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#
Services
Resource Groups
Nishanth Kothakota
Ohio
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Spot Requests
Reserved Instances
Dedicated Hosts
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Resources
You are using the following Amazon EC2 resources in the US East (Ohio) region:
0 Running Instances
0 Dedicated Hosts
0 Volumes
3 Key Pairs
0 Placement Groups
0 Elastic IPs
1 Snapshots
1 Load Balancers
3 Security Groups
Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. Try Amazon Lightsail for free.
Create Instance
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
Launch Instance
Note: Your instances will launch in the US East (Ohio) region
Service Health
Service Status:
US East (Ohio):
This service is operating normally
Scheduled Events
US East (Ohio):
No events
Account Attributes
Supported Platforms
VPC
Default VPC
vpc-670d690e
Resource ID length management
Additional Information
Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us
AWS Marketplace
Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:
Barracuda NextGen Firewall F-Series - PAYG
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
1:49 PM
7/31/2017


Secure | https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
Services
Resource Groups
Nishanth Kothakota
Ohio
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
Quick Start
1 to 32 of 32 AMIs
My AMIs
AWS Marketplace
Community AMIs
Free tier only
Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-8a7859ef
Amazon Linux
Free tier eligible
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs
Virtualization type: hvm
Select
64-bit
SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-61a7fd04
SUSE Linux
Free tier eligible
SUSE Linux Enterprise Server 12 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs
Virtualization type: hvm
Select
64-bit
Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type - ami-11aa8c74
Red Hat
Free tier eligible
Red Hat Enterprise Linux version 7.3 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs
Virtualization type: hvm
Select
64-bit
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
1:56 PM
7/31/2017

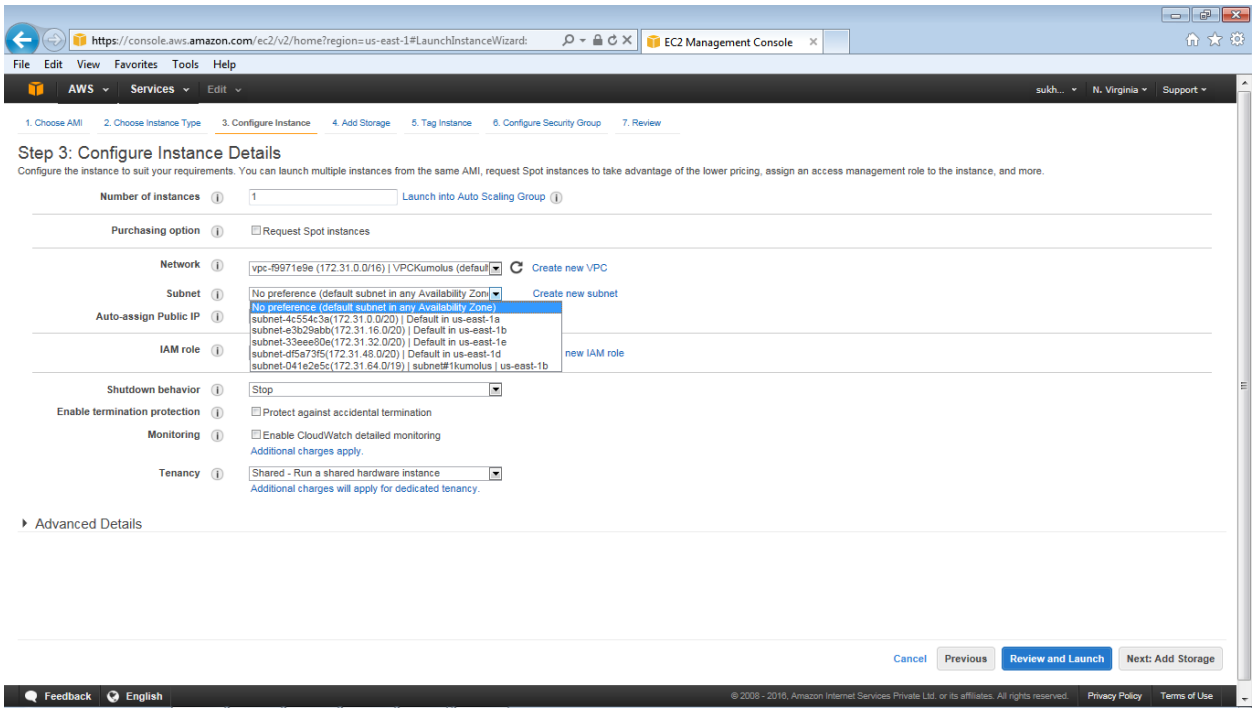
- Select the Instance type as per the requirement and click on Next to start Configure Instance Details



- In Configure Instance tab select Network (Sample: VPCKumolus) .

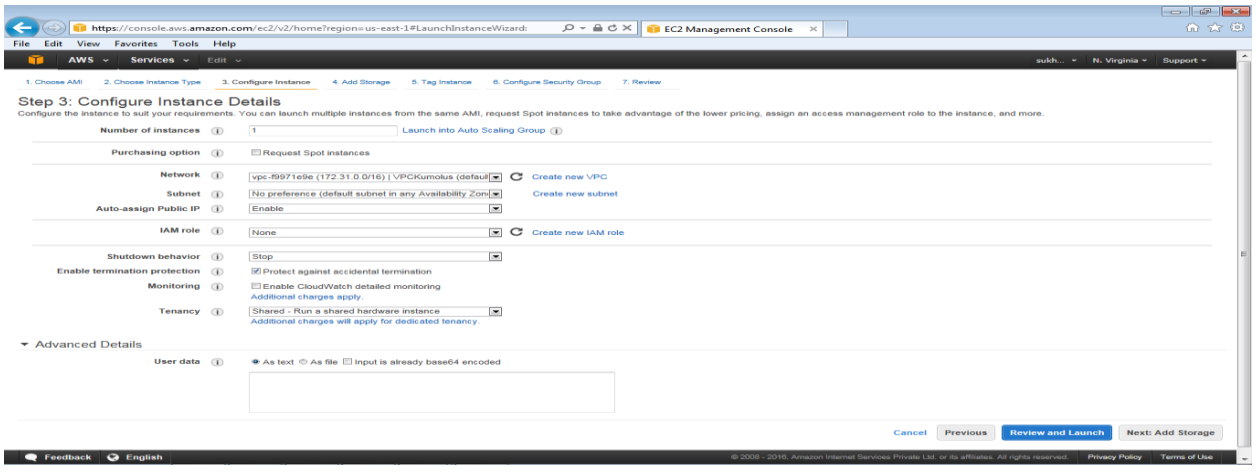


- Select the default Subnet / Subnet as per available IP to be assigned



- Select Auto-assign Public IP (Enable / Disable). This is required for services offered in AWS to work with the Instance.

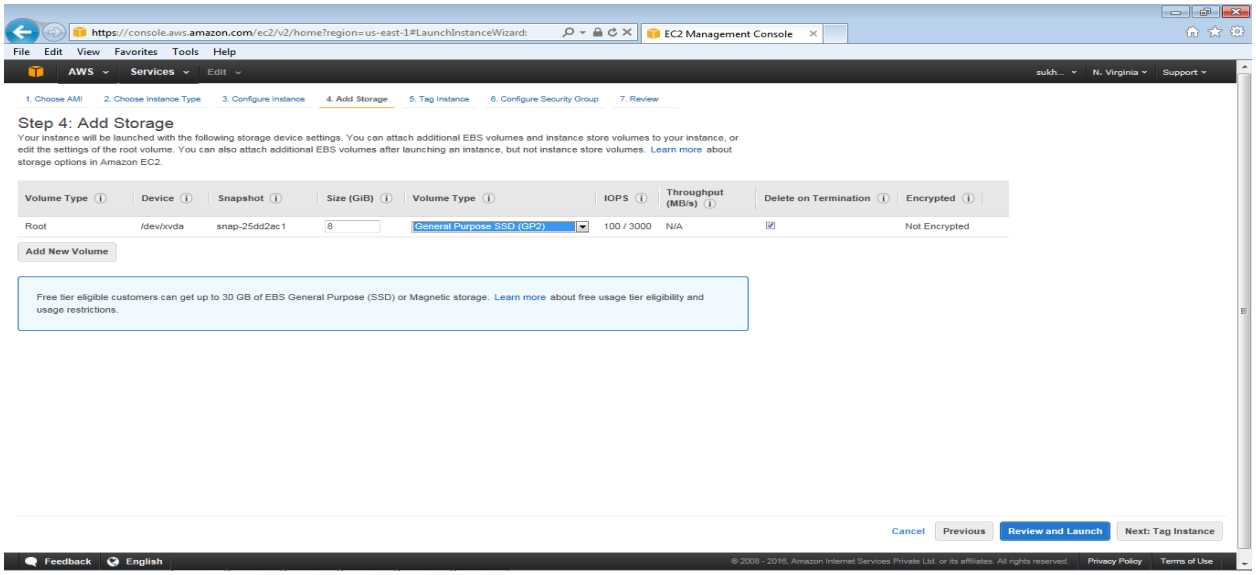
- Check option Protect against accidental termination under Enable Termination protection option.



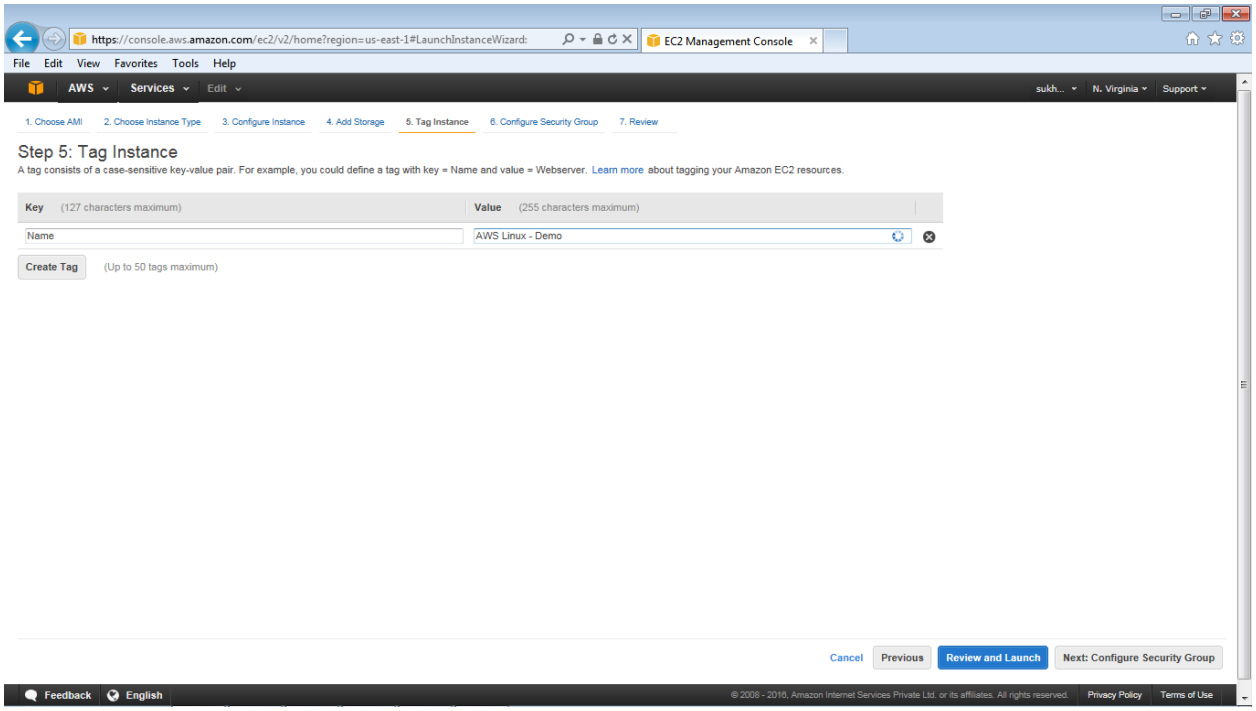
- Proceed to next step to add Storage.

- Select Volume type(General Purpose) as per the requirement and proceed to next step to add Tags to an instance. Other EBS Volume Type are EBS Provisioned IOPS (SSD) & EBS Magnetic

  http://aws.amazon.com/ebs/details/

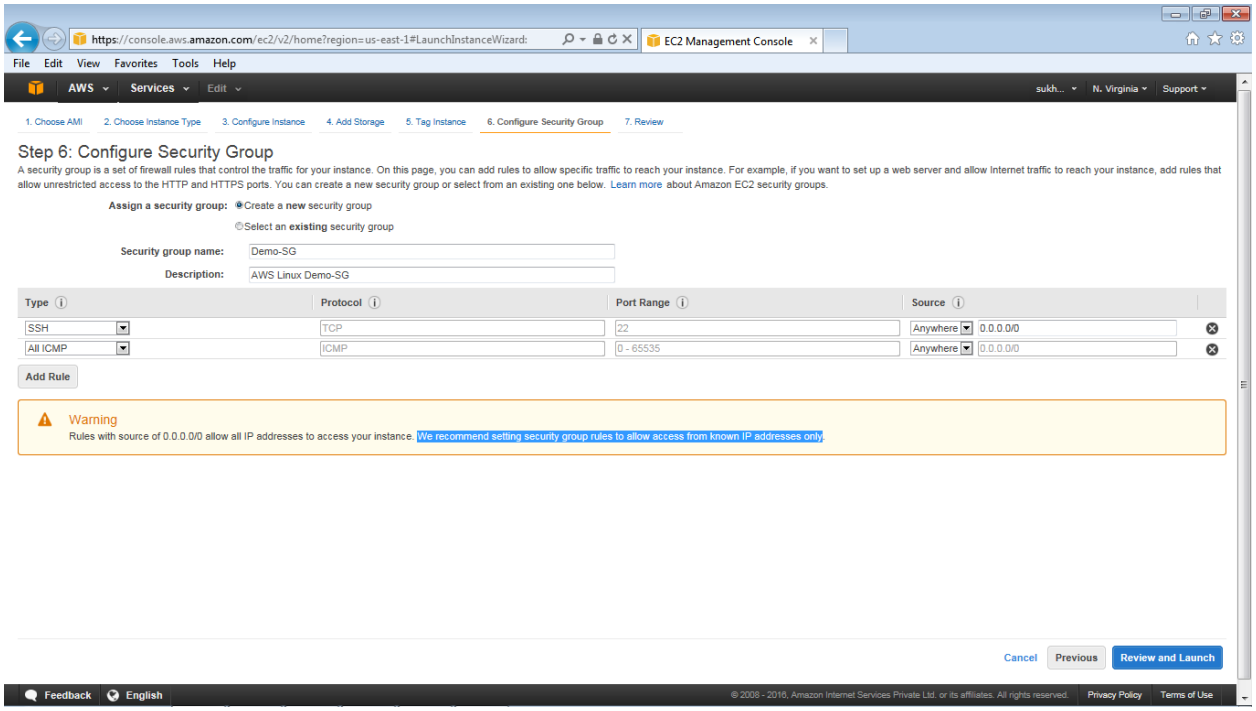


- Tag Instance – provide the Key Value of the server and proceed now to next step to Configure Security group.

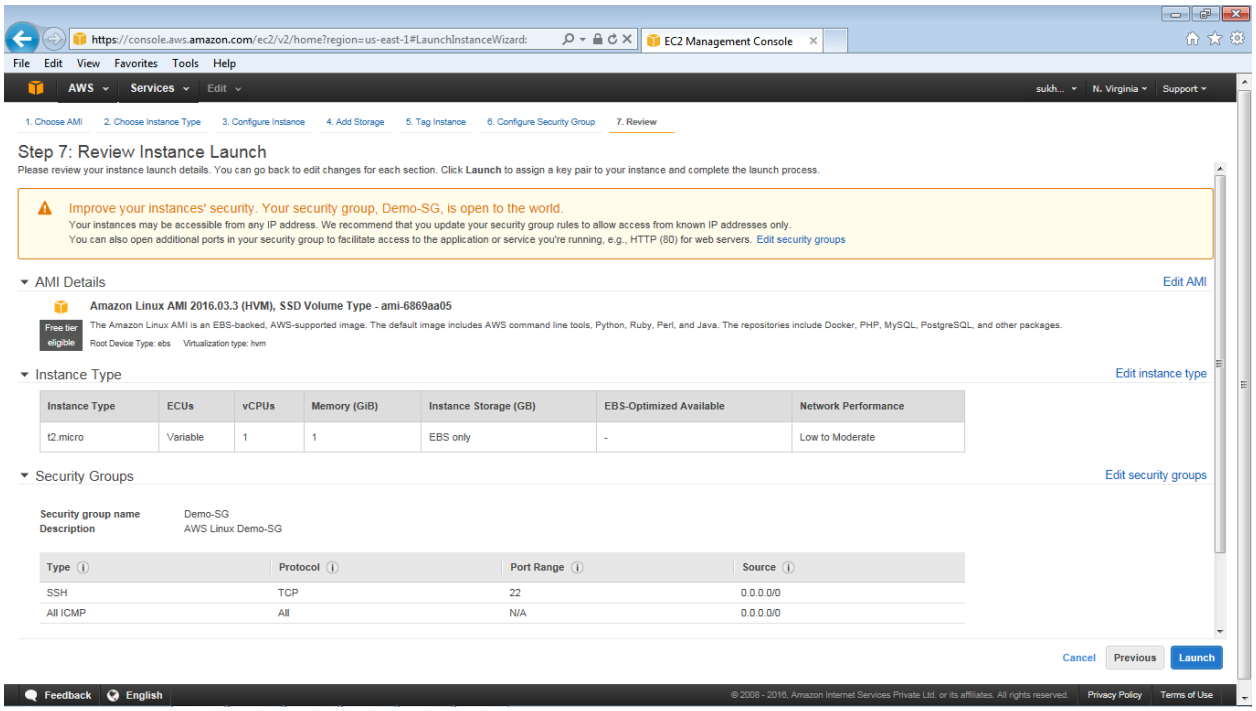


- Create a new security group, provide the required protocals and proceed with Review and Launch.

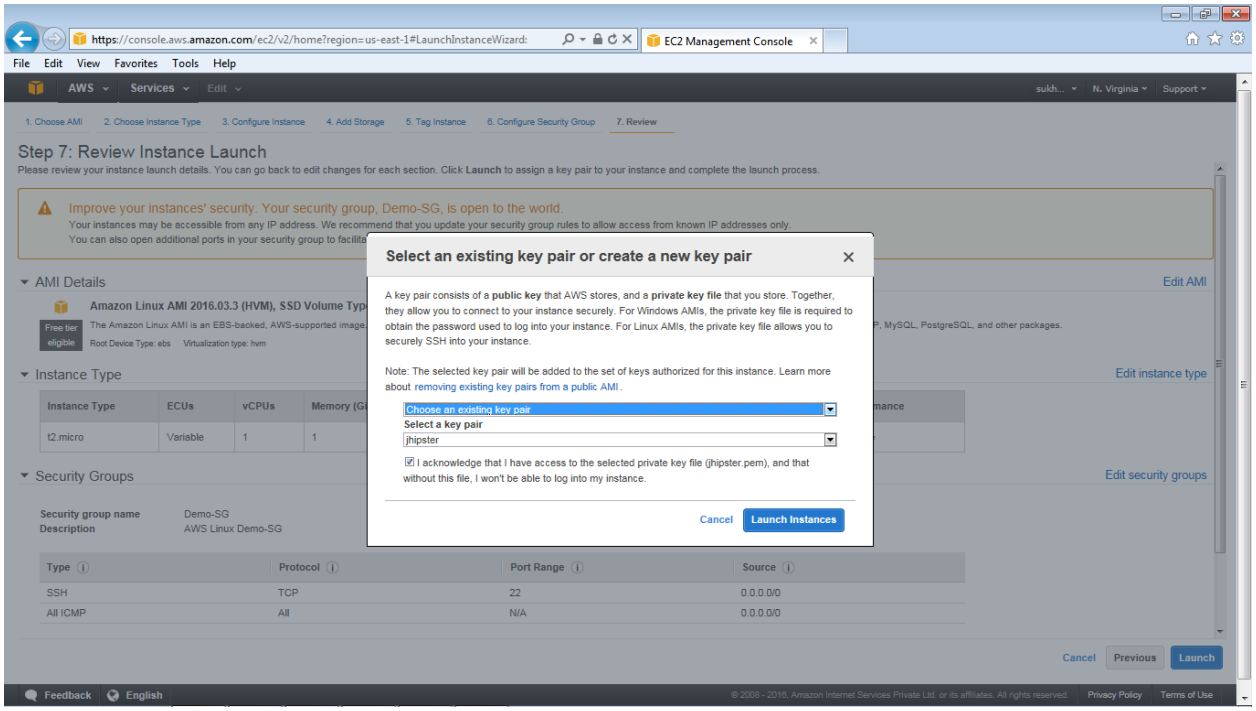
  (Note: if SG is already defined earlier then use option as Select existing security group)



- Review the instance details

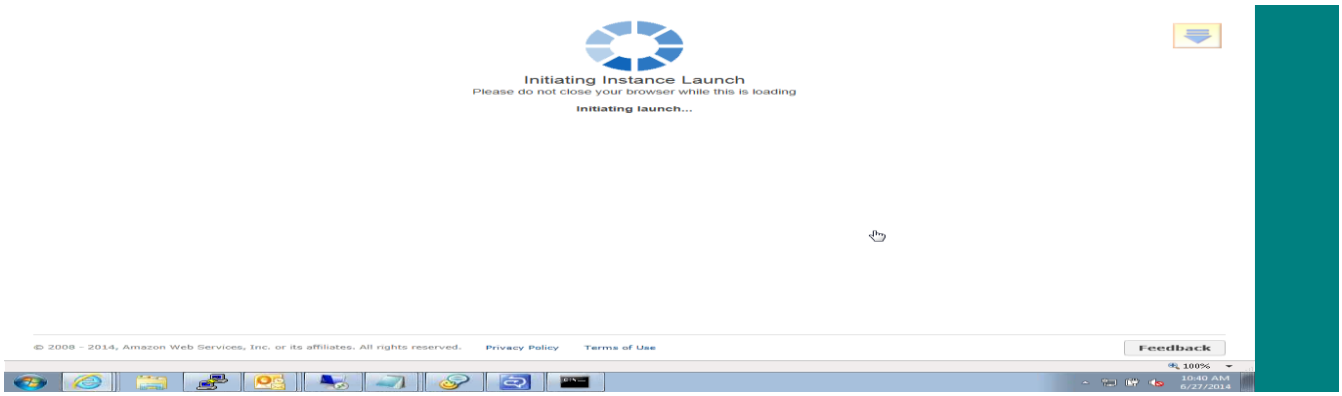


- Select the option ‘choose an existing key pair’.

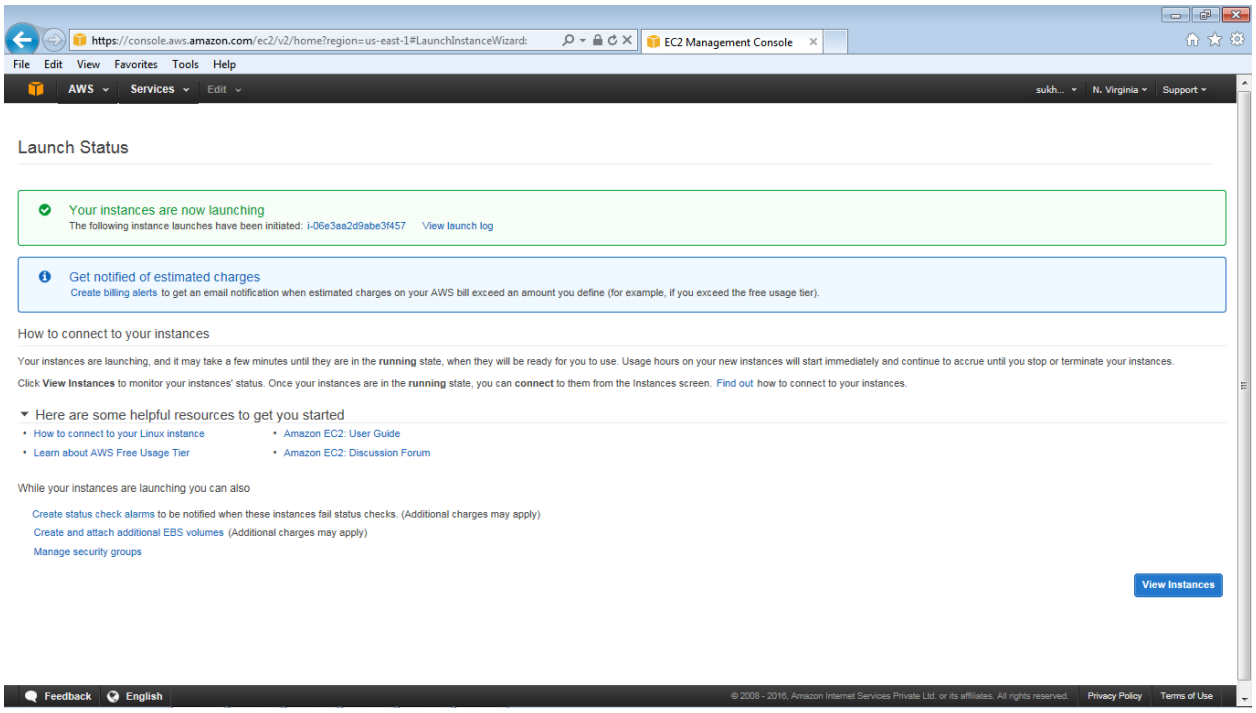


- Select the existing key pair (Sample: Jhipster), Select I acknowledge checkbox and click on Launch instances.

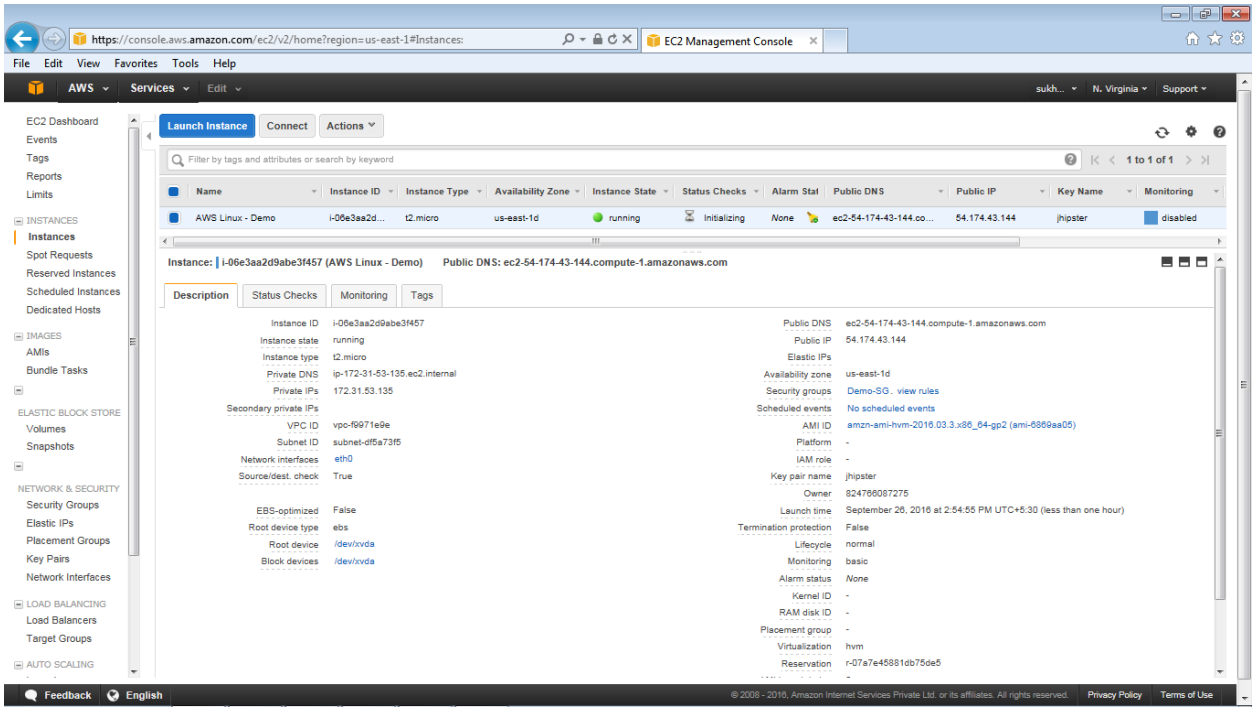
- The dashboard will show the Instance launch is in progress.



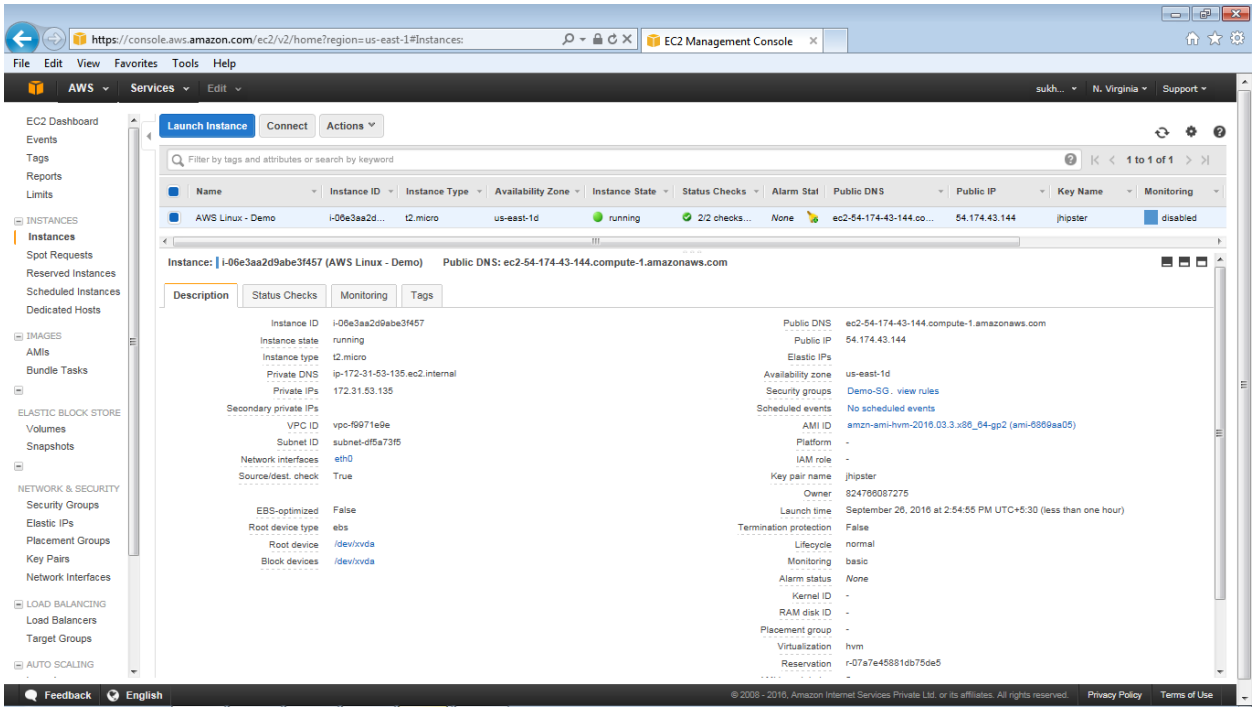
- Click on view instances.



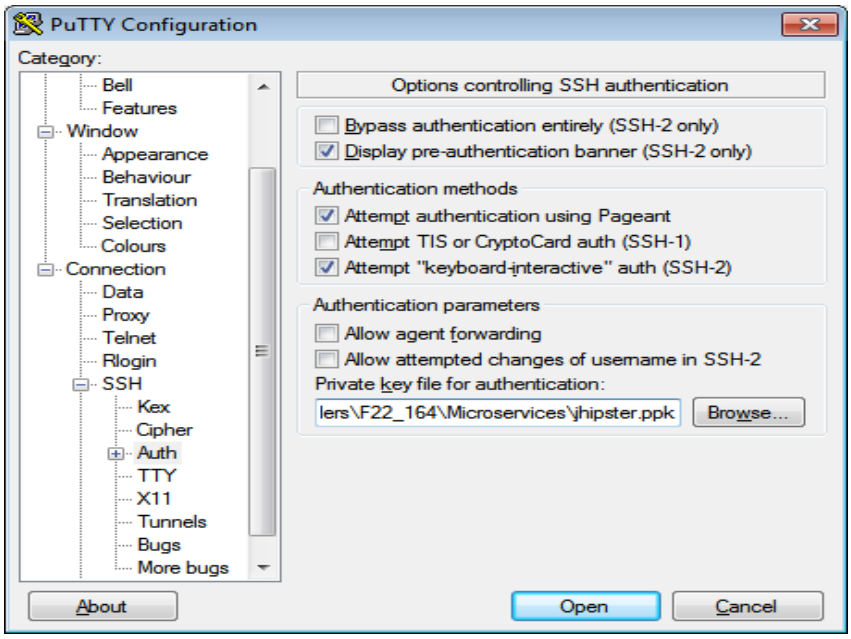
- Initially the instance state is pending . It will change to running state once the VM build is complete and then the instance is ready to be accessed.



- Select the instance which is created and verify the details. [Status code: 2/2 checks]

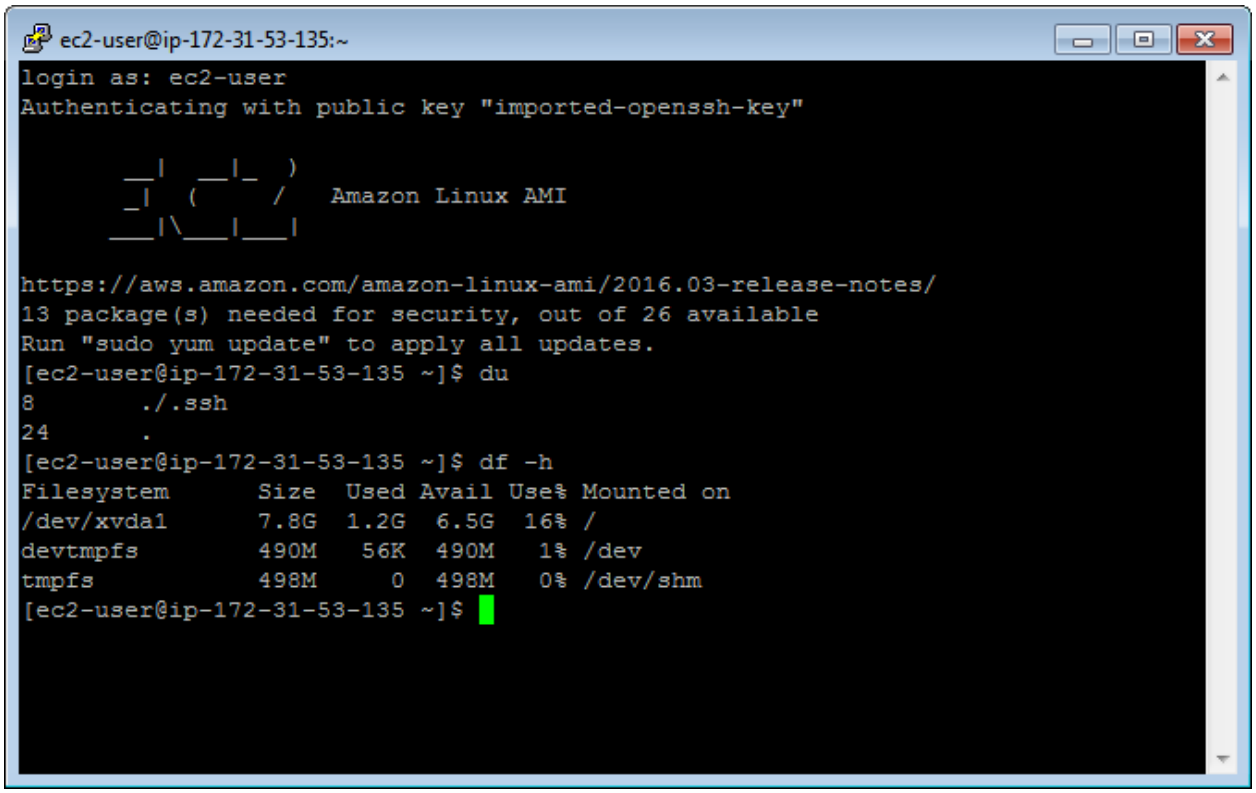


- Access the instance using putty and login using Public DNS and ppk file.



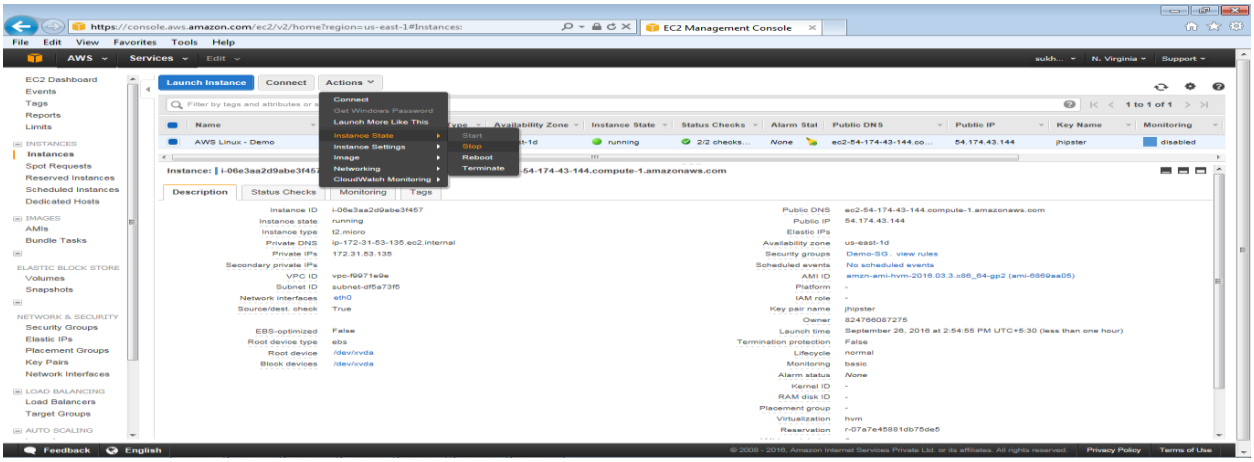
- Login as ec2-user and press enter and check the disk space using “df -h”

Note: For security reason you can assign password while generating the ppk file using puttygen tool

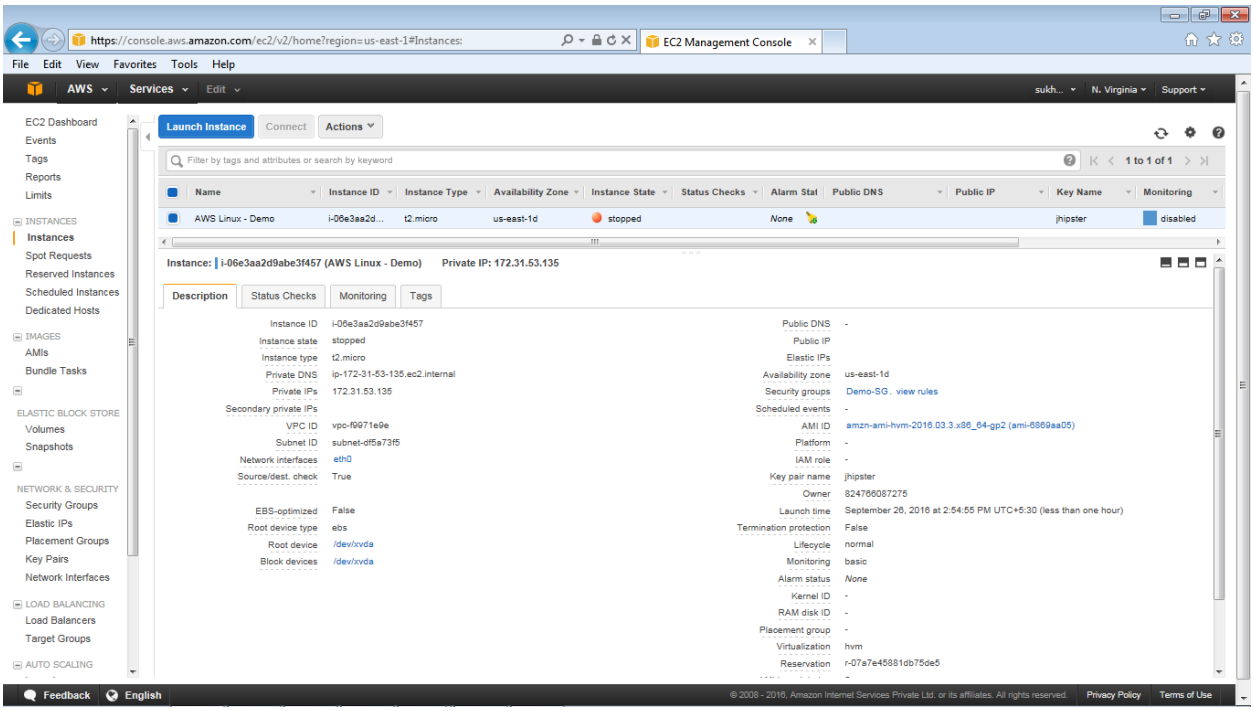


1.3.2 Resizing the EC2 instance

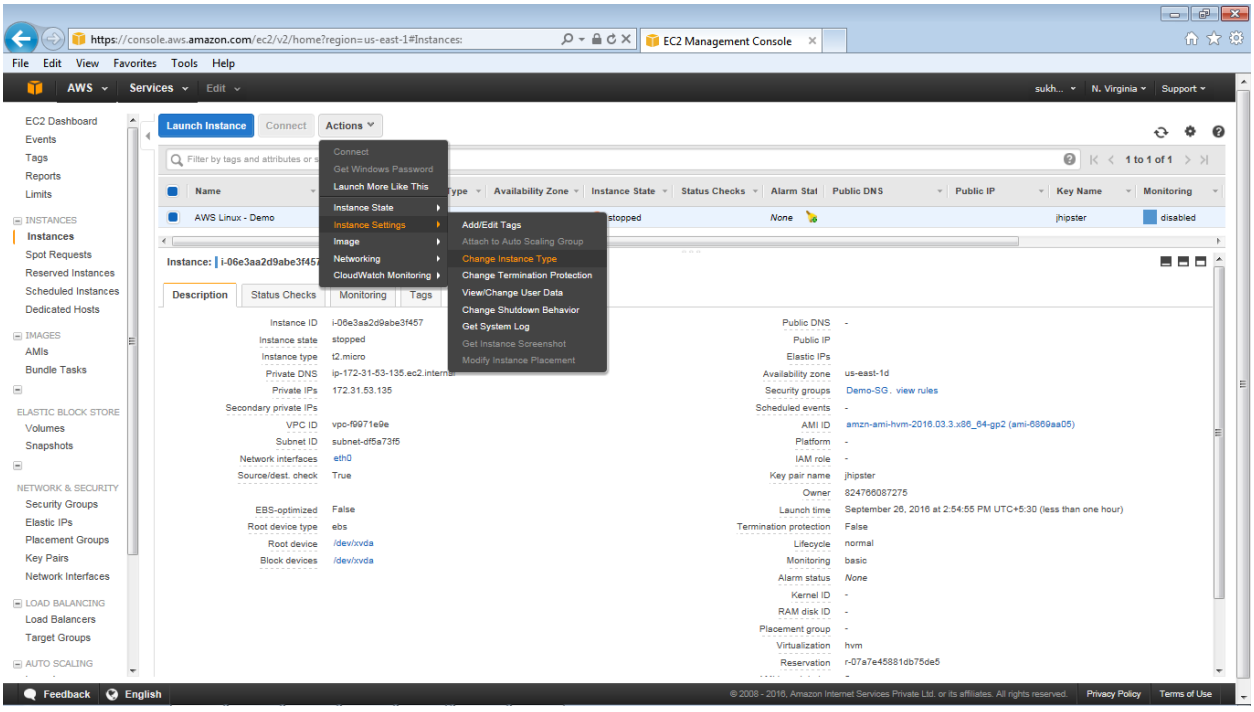
1. Go to the EC2 Dashboard and select the intances
2. Choose **Actions**, select **Instance State**, and then choose **Stop**
3. In the confirmation dialog box, choose **Yes, Stop**. It can take a few minutes for the instance to stop

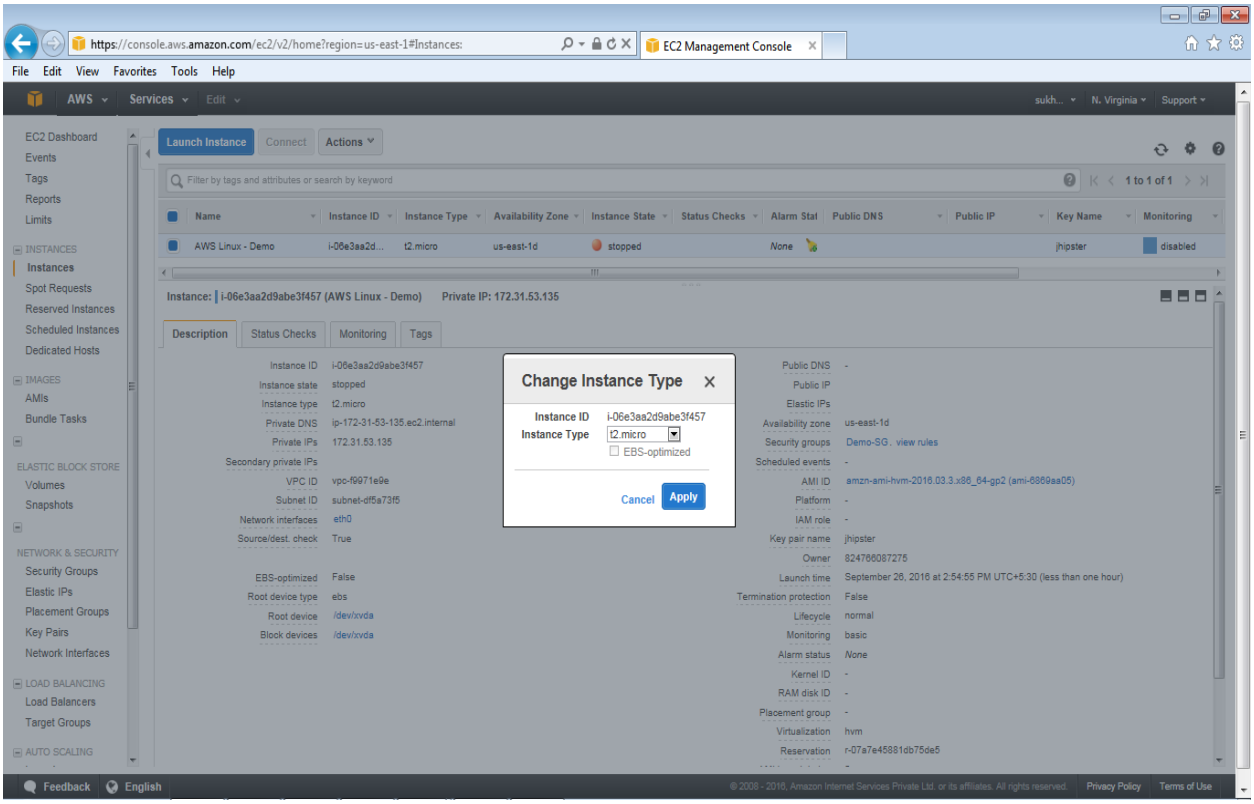


4. Instance status code is shown as “stopped”

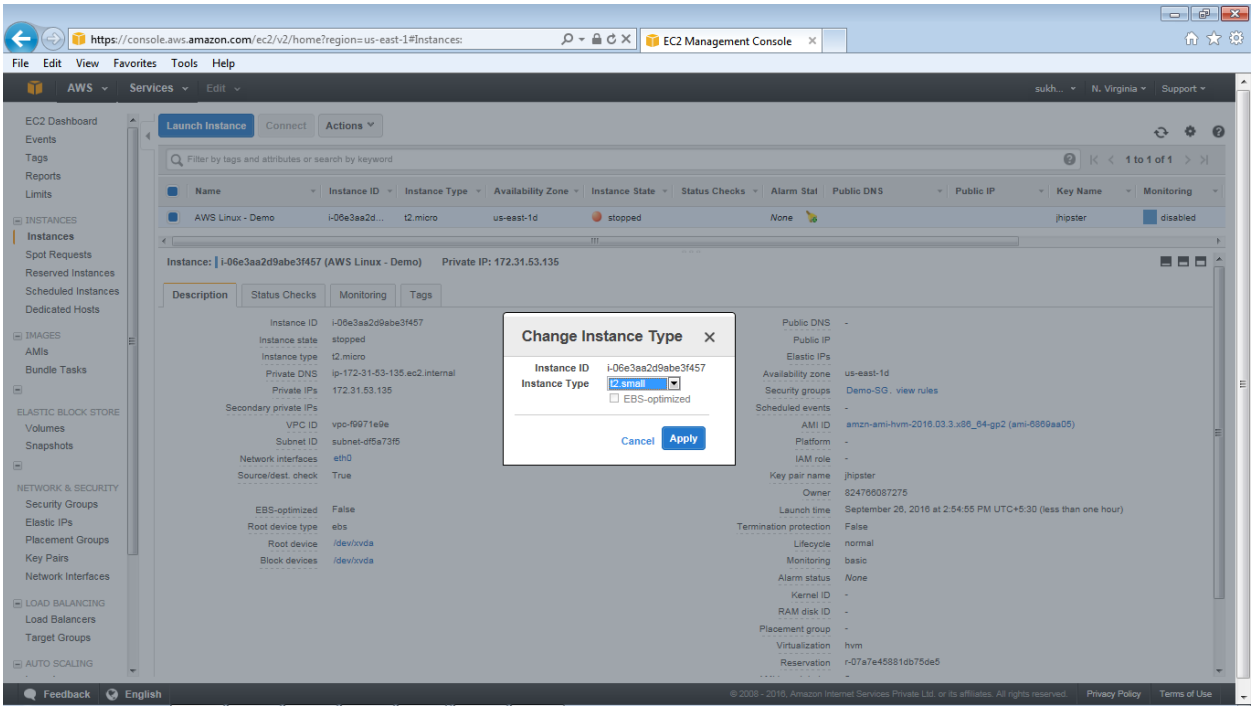


5. With the instance still selected, choose Actions, select Instance Settings, and then choose Change Instance Type. Note that this action is disabled if the instance state is not stopped

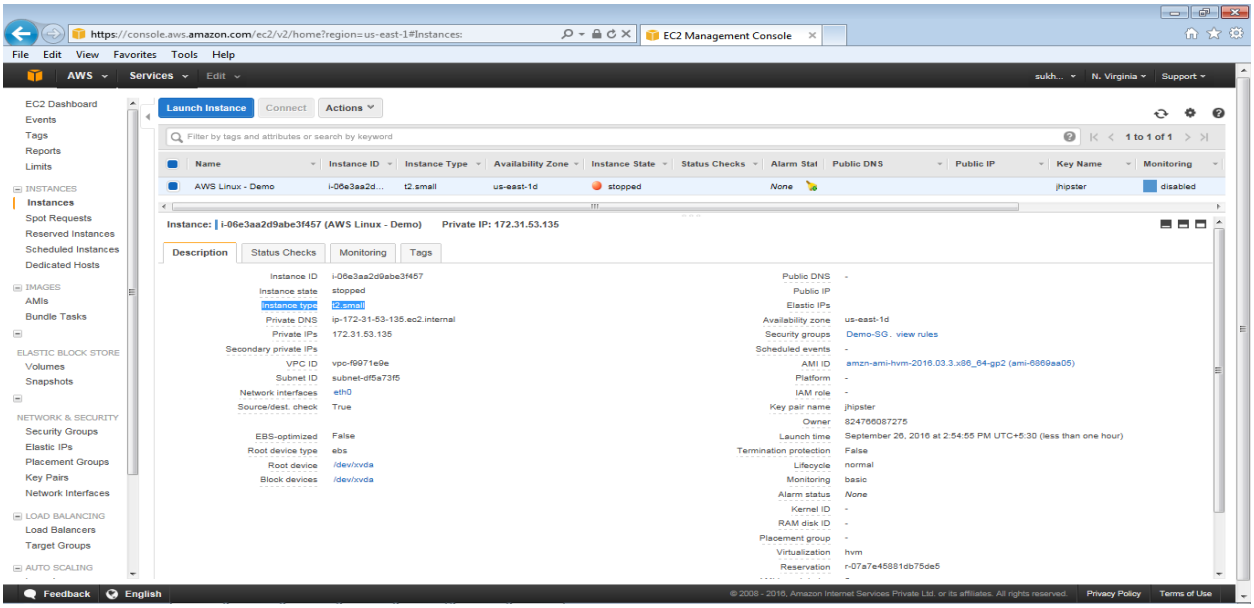




6. Click on apply button



7. Choose Apply to accept the new settings. This resize the instance from (t2.micro to t2.small)

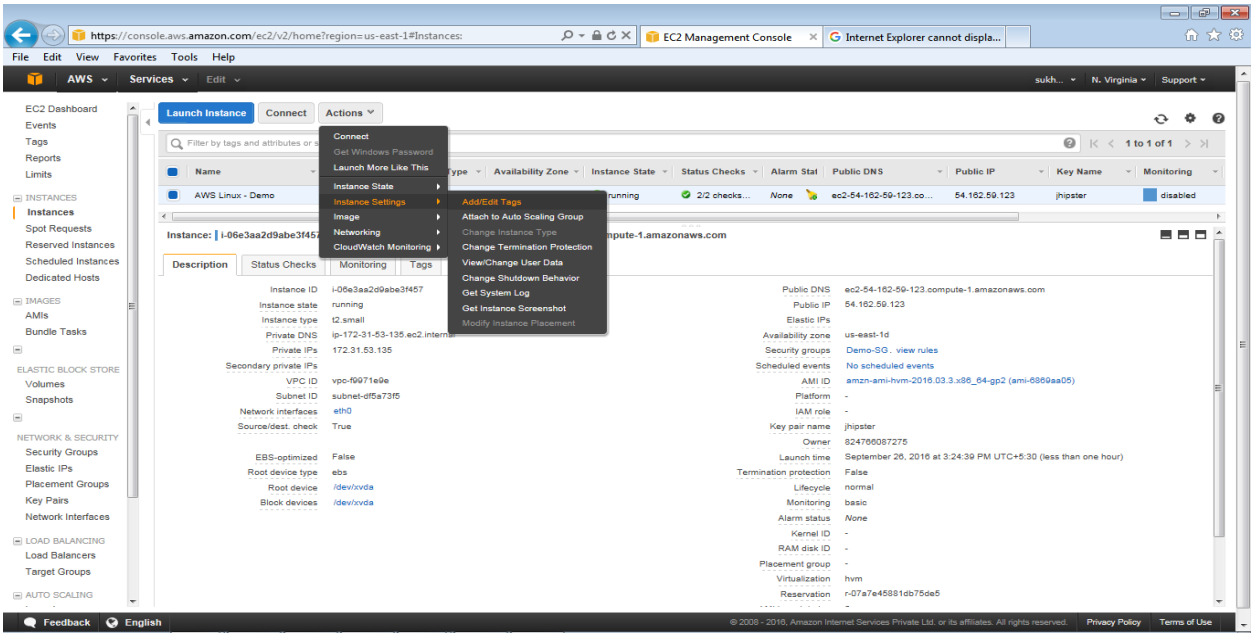


8. From Action option start the instance, login using putty and view the instances details

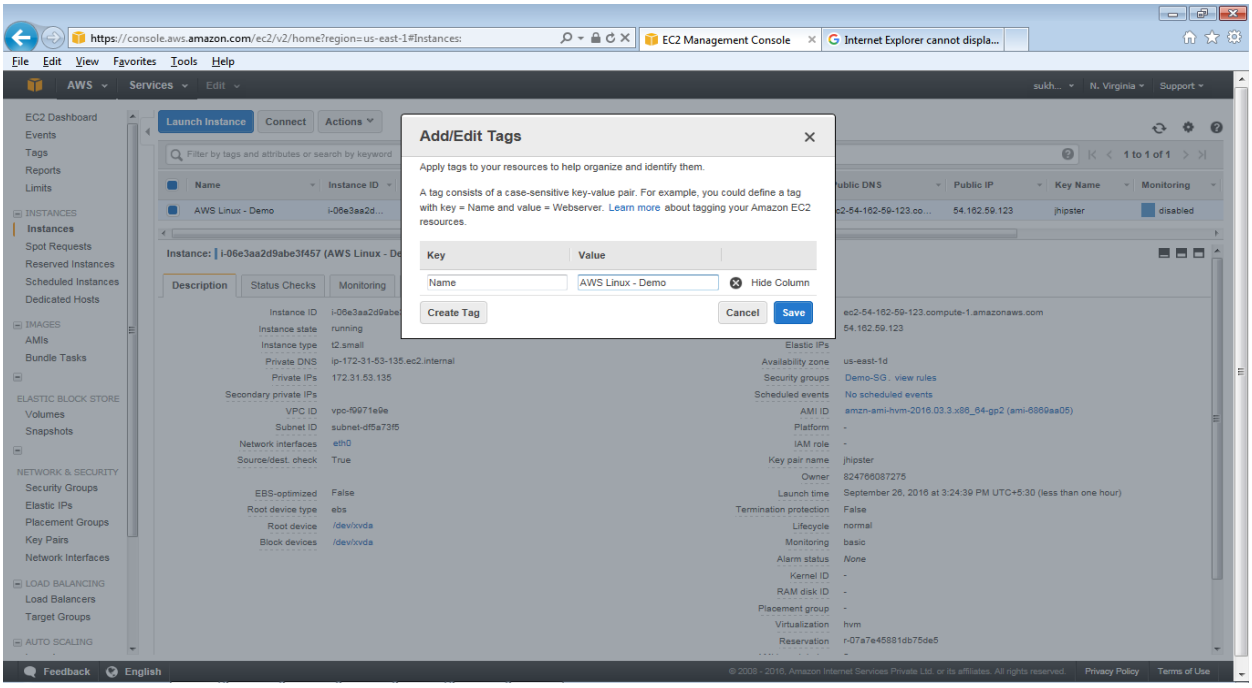
1.3.3 Modify EC2 instance

1. Go to the EC2 Dashboard and select the intances
2. Start the instance (incase stopped)

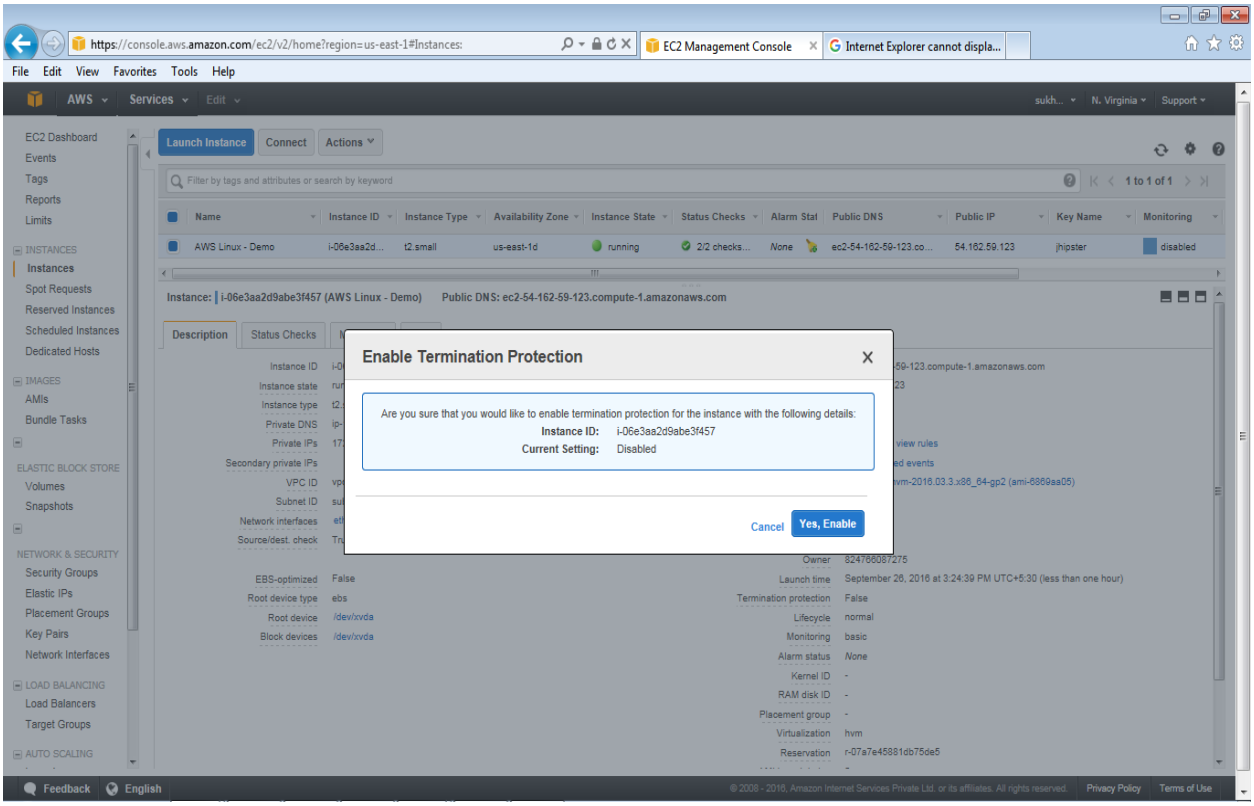
   Note: If EIP is not associated with the instance it will generate the new Public IP address)
3. Add/ Edit the Tags – Action – Add/Edit Tags



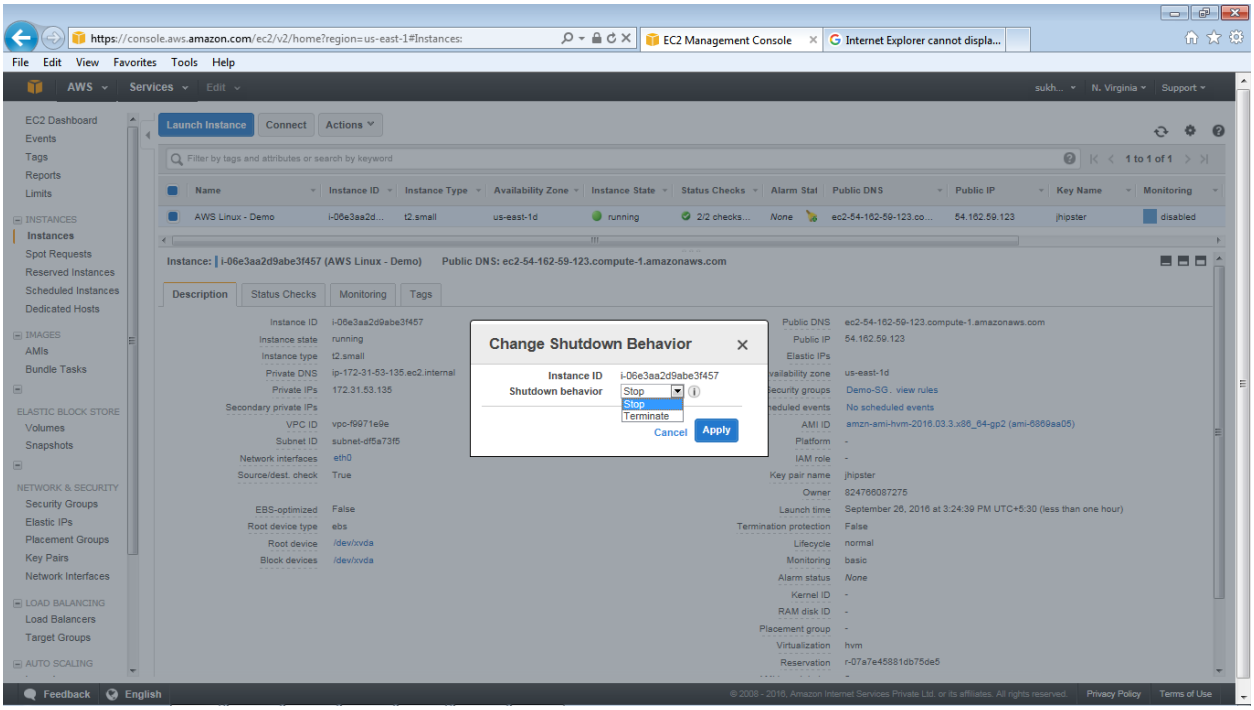
4. Add/ Edit the Tags of the instance and press save



5. Reset the change Termination protection of the instance using “Yes Enable” button

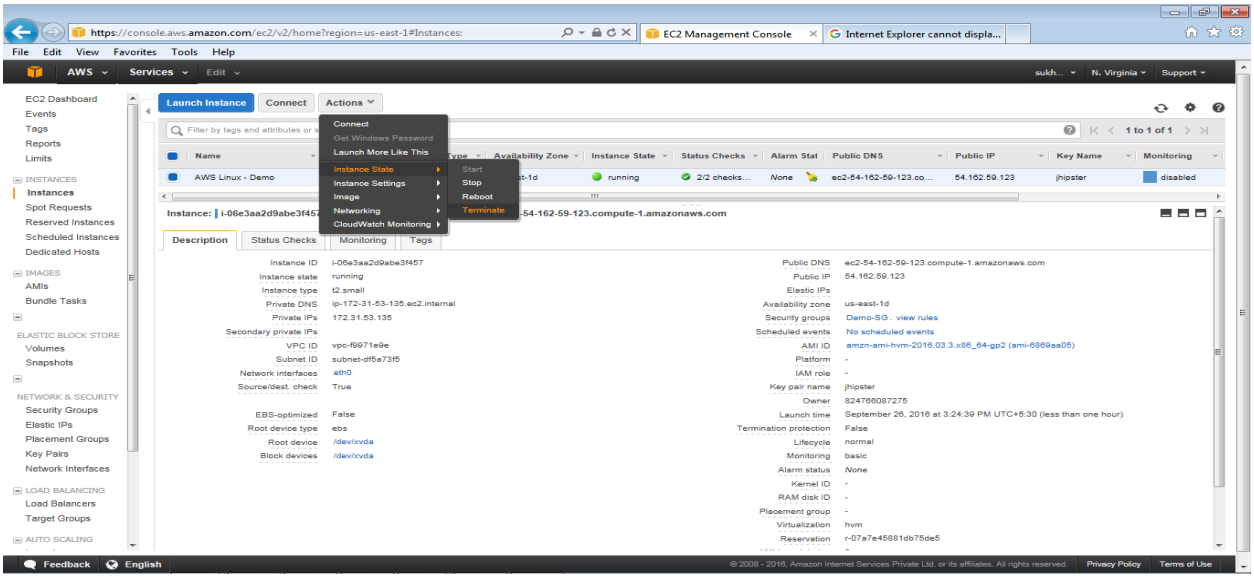


6. Reset the change the shutdown behavior of the instance selecting option “ Stop/Terminate”

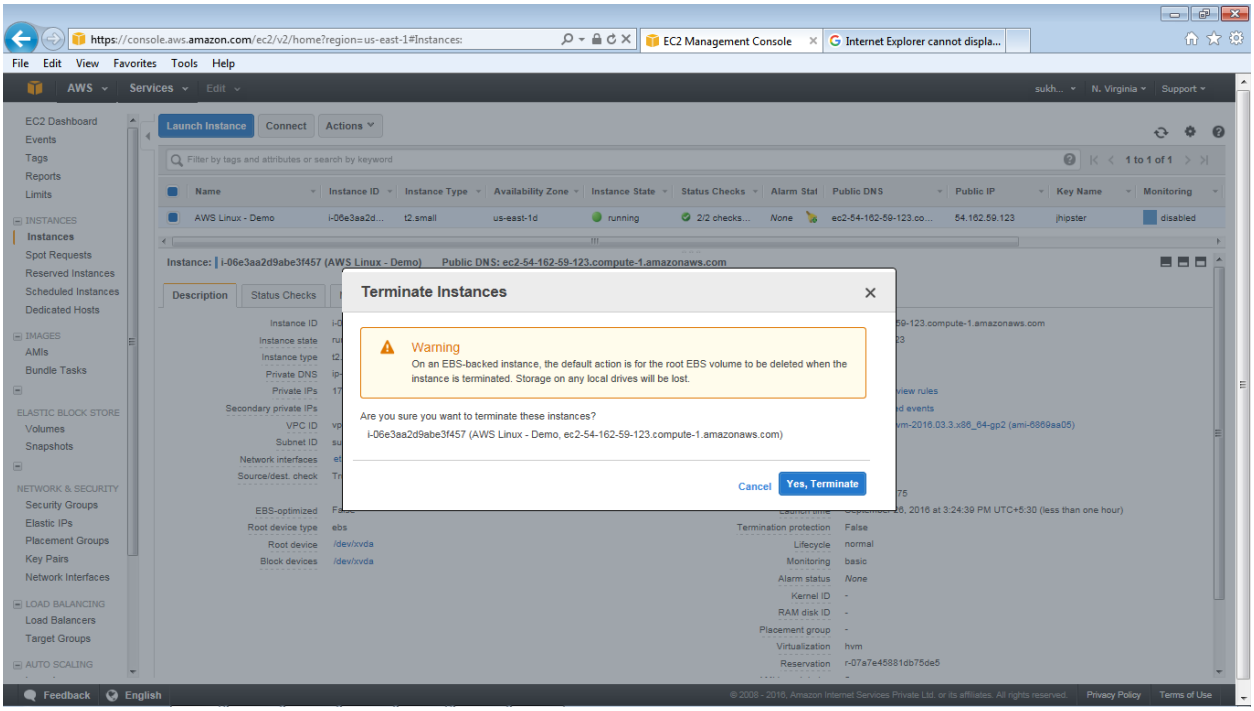


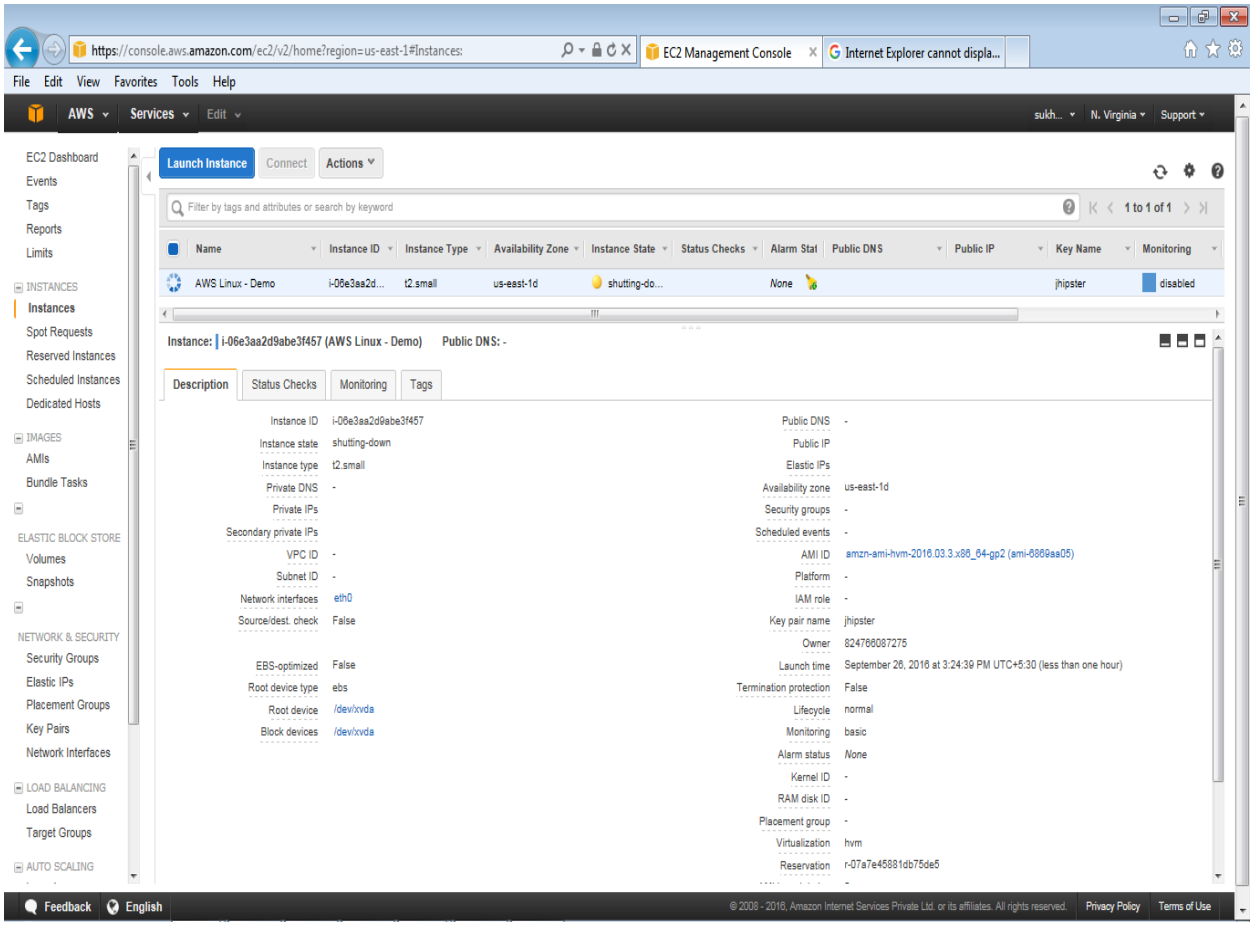
1.3.4 Terminating EC2 instance

1. Go to the EC2 Dashboard and select the intances
2. From Action select Terminate option , it will show warning message

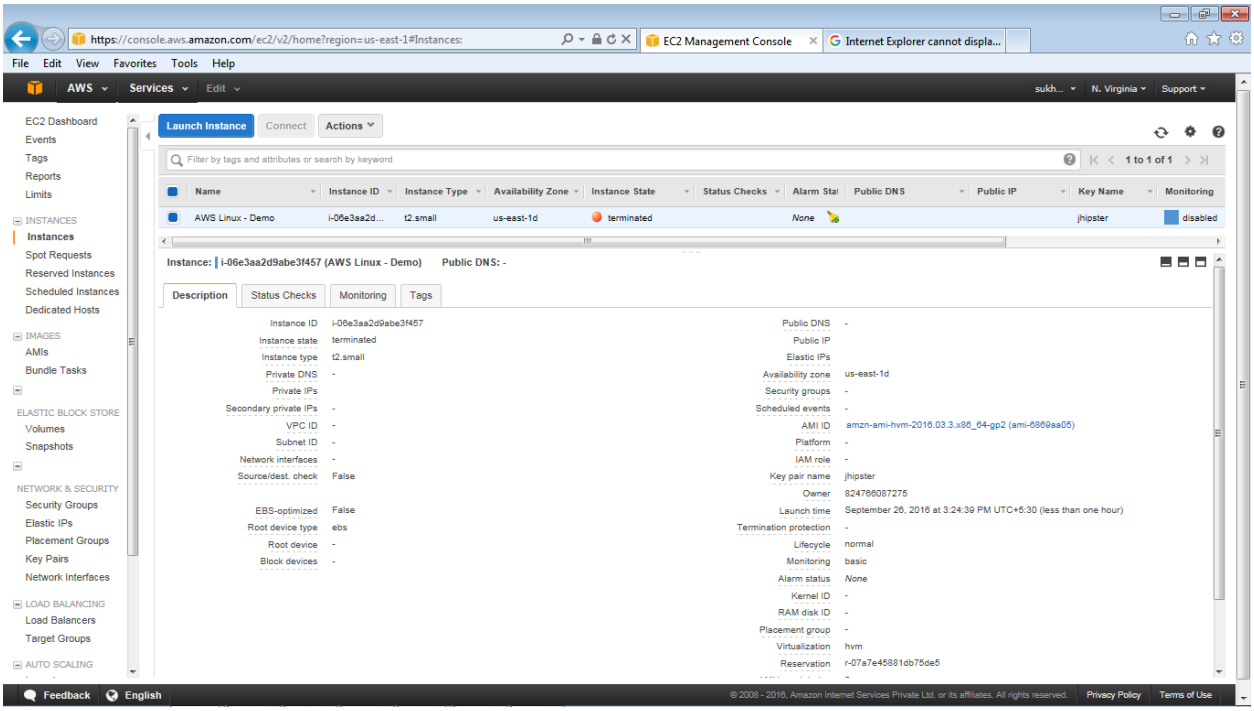


3. Press “Yes Terminate” button


Terminate Instances
Warning
On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.
Are you sure you want to terminate these instances?
i-06e3aa2d9abe3f457 (AWS Linux - Demo, ec2-54-162-59-123.compute-1.amazonaws.com)
Cancel
Yes, Terminate

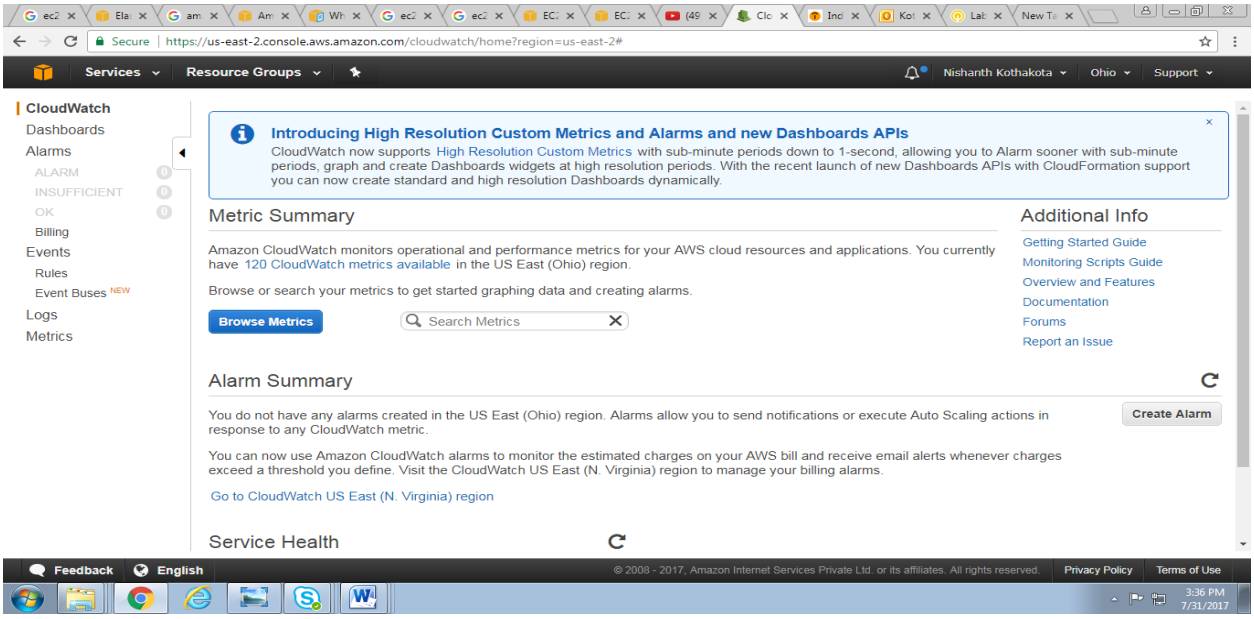

AWS Linux - Demo
i-06e3aa2d...
t2.small
us-east-1d
shutting-do...
None
jhipster
disabled
Instance: i-06e3aa2d9abe3f457 (AWS Linux - Demo)
Public DNS: -
Description
Status Checks
Monitoring
Tags

4. The Instance status code is changed to Terminated

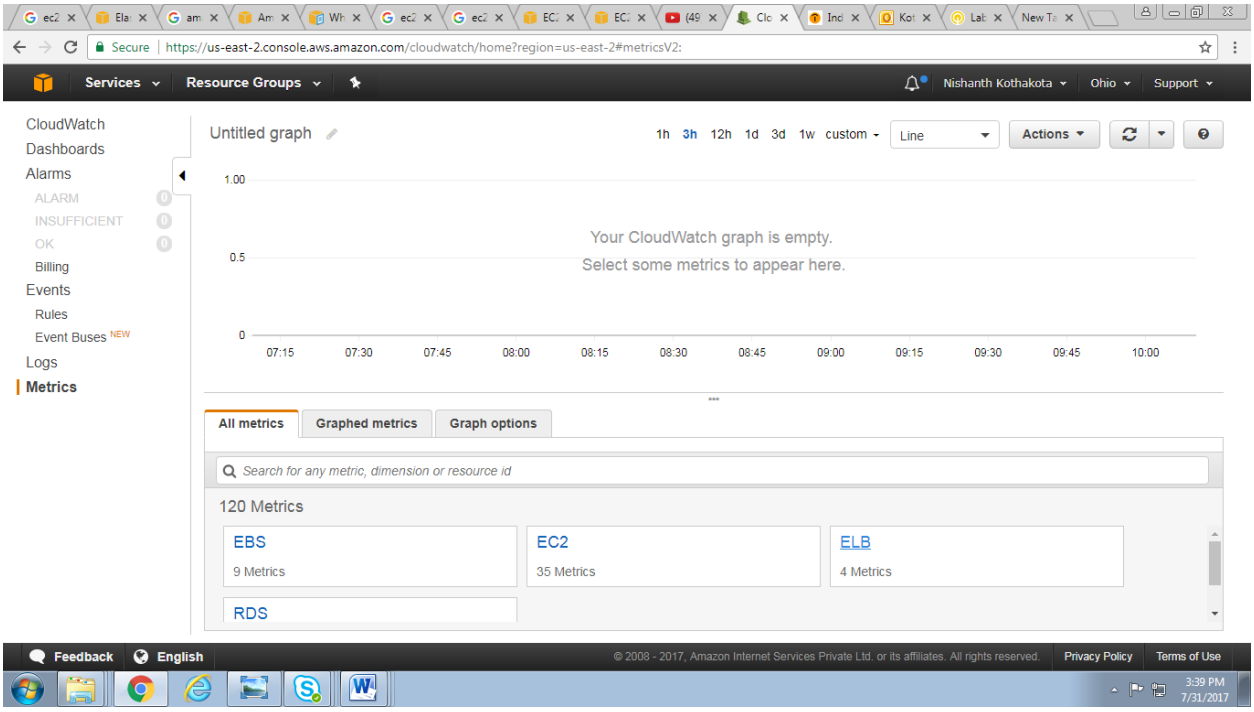


1.2.5 MONITORING EC2 INSTANCE

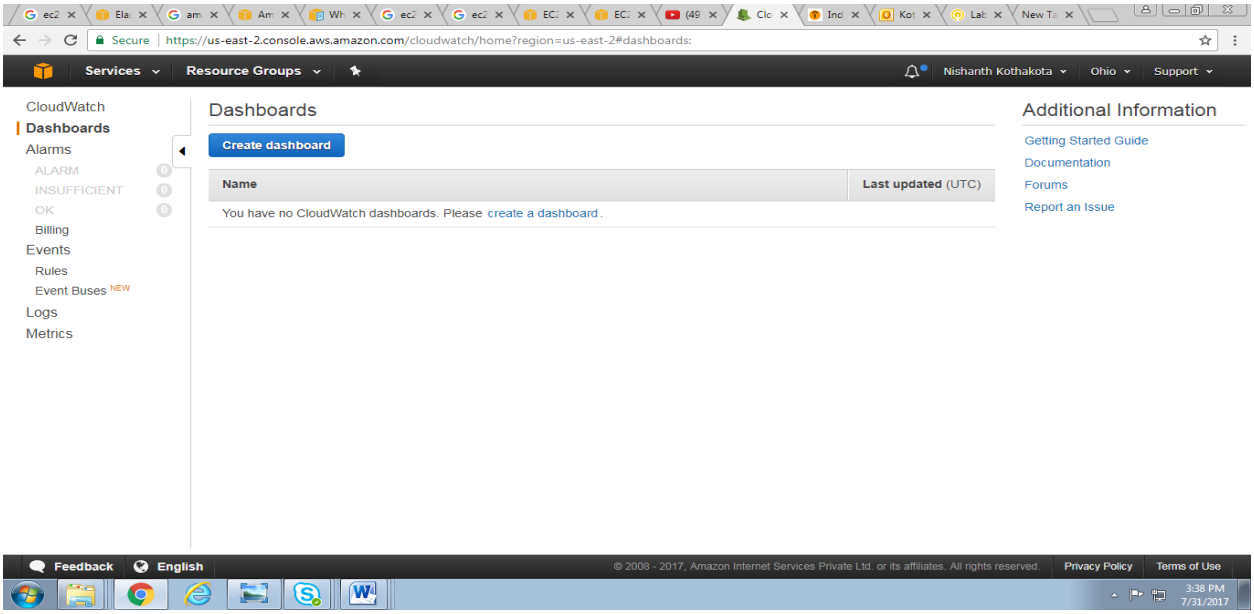
1. Log on the AWS console page and click on cloud watch

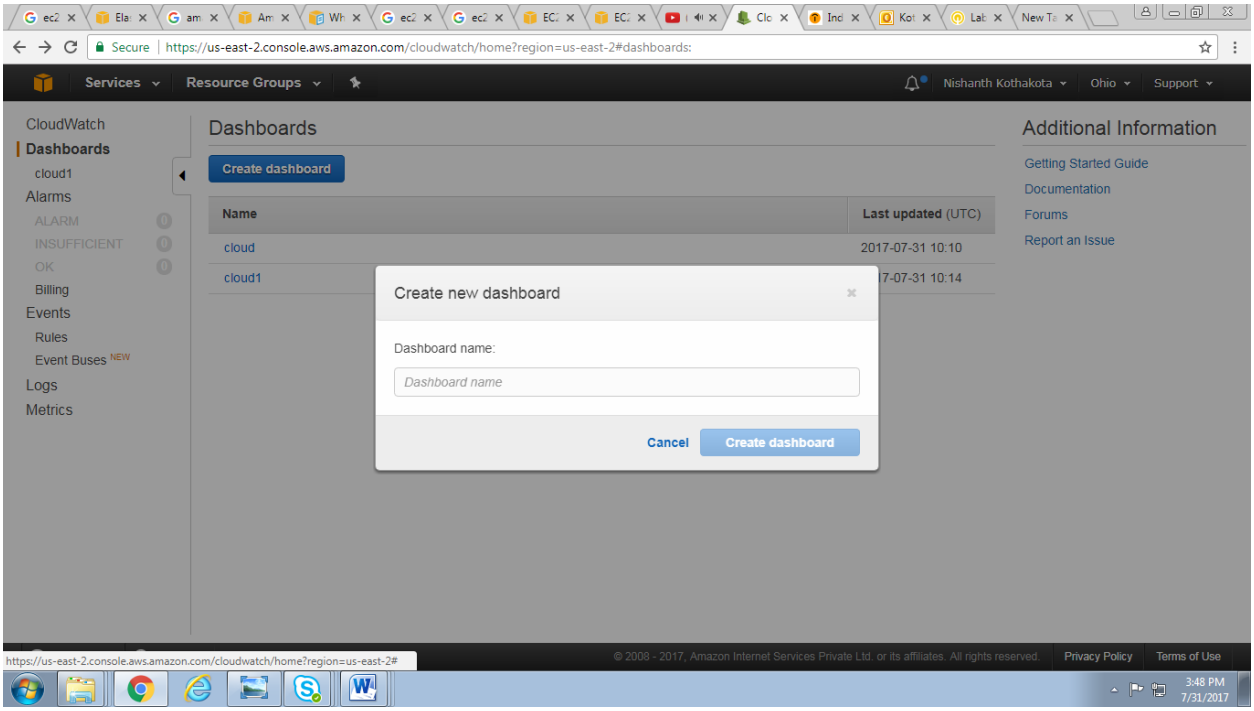


2. Click on Browse Metrics

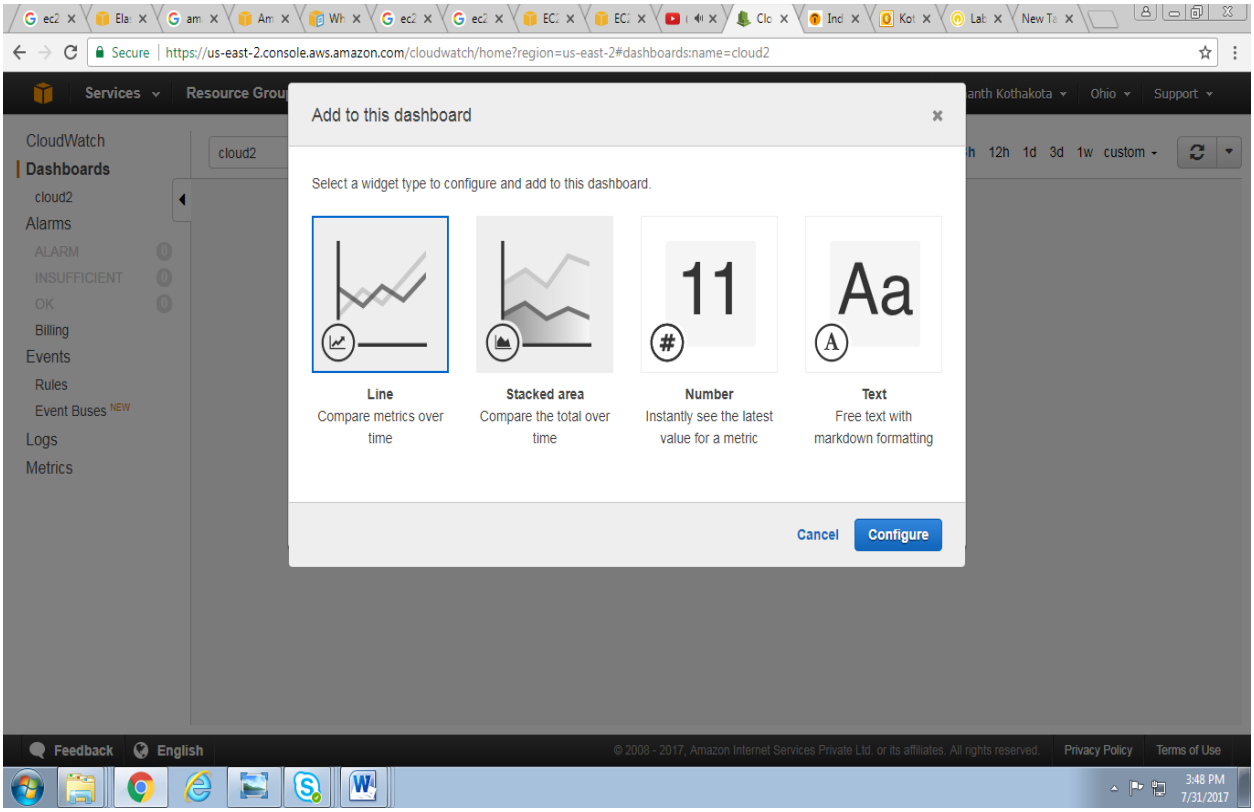


3. Click on Dash Board and create new Dashboard

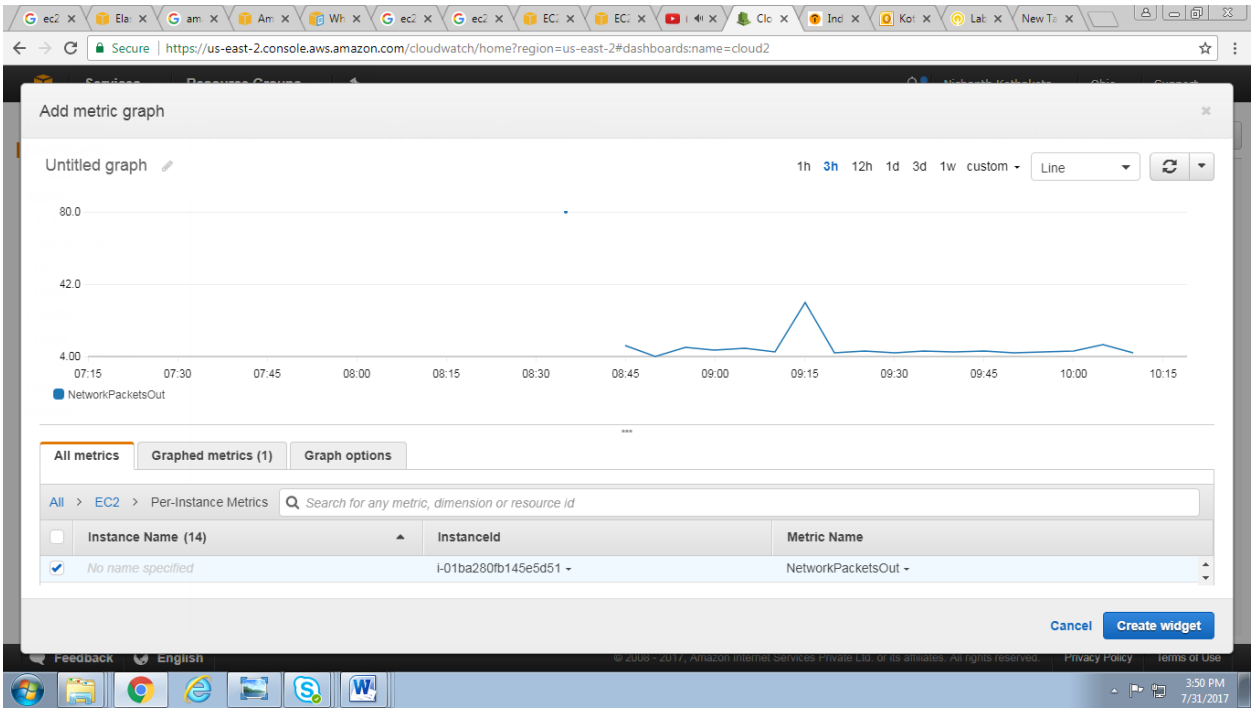




4. After that configure the metric graph and selct the instance



5. After adding the instance it shows utilization of instance



2. VPC MANAGEMENT

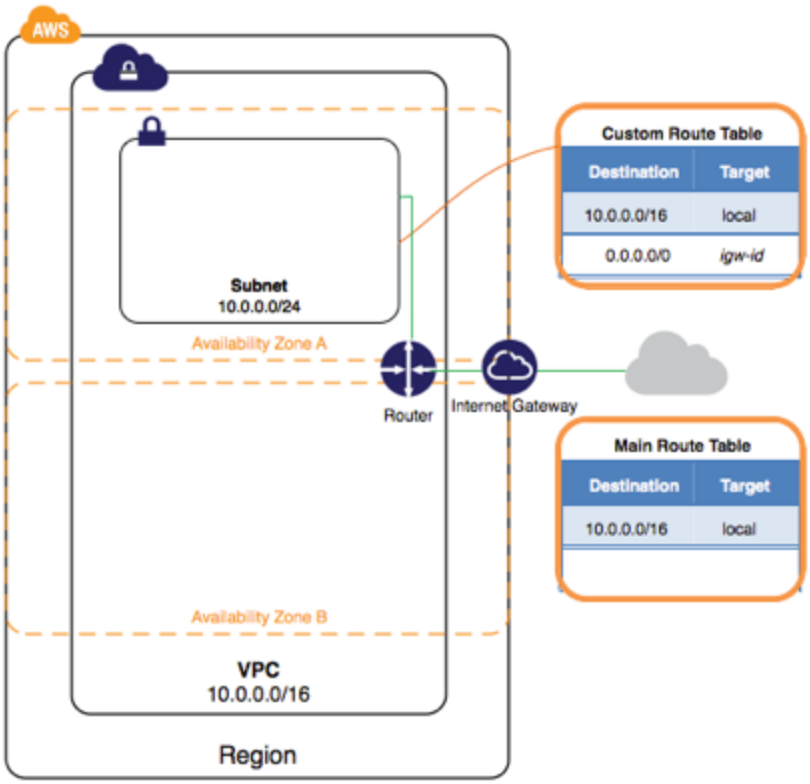
2.1 Objective

To provide the high level guidance's to setup the AWS Virtual Private Cloud (VPC) on AWS Cloud.

2.2 Assumptions

- AWS Account is available with AWS Identity access Management (IAM) or Amazon management console
- VPCs and Subnets. A virtual private cloud (**VPC**) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud

The below diagram shows AWS VPC Architecture.



2.3 Procedure

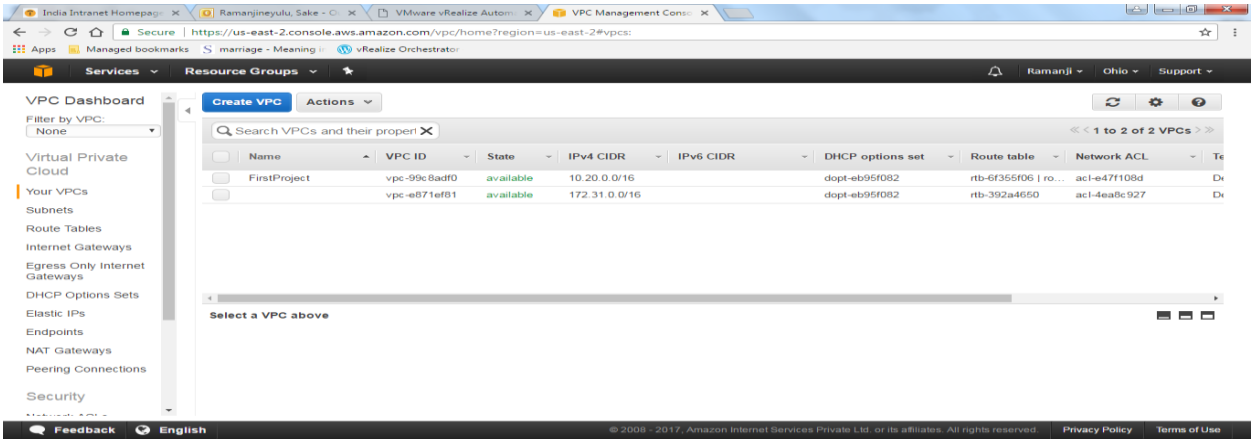
To set up a VPC, you need to complete the following steps:

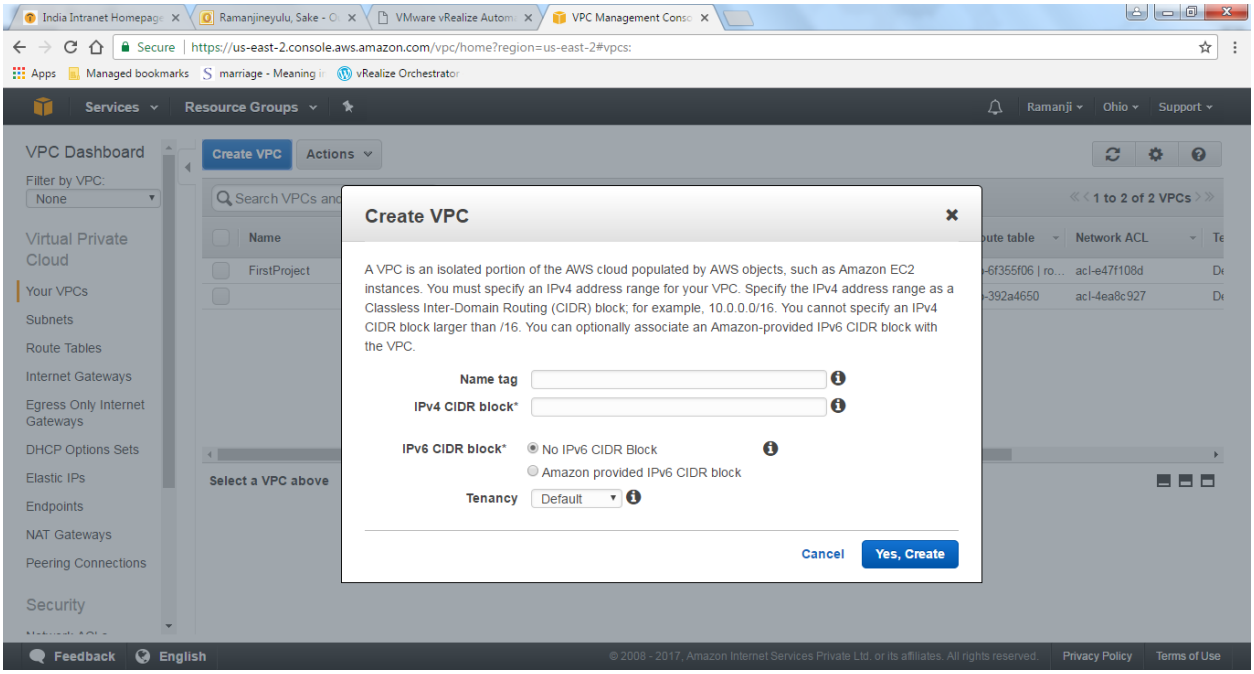
Step 1: Create a new VPC

Step 2: Configuration and management of subnets and network ACLs and Configuration and management of security groups

Step 3: Configuration of AWS VPN devices and on premise VPN devices

2.3.1 Creation of new VPC





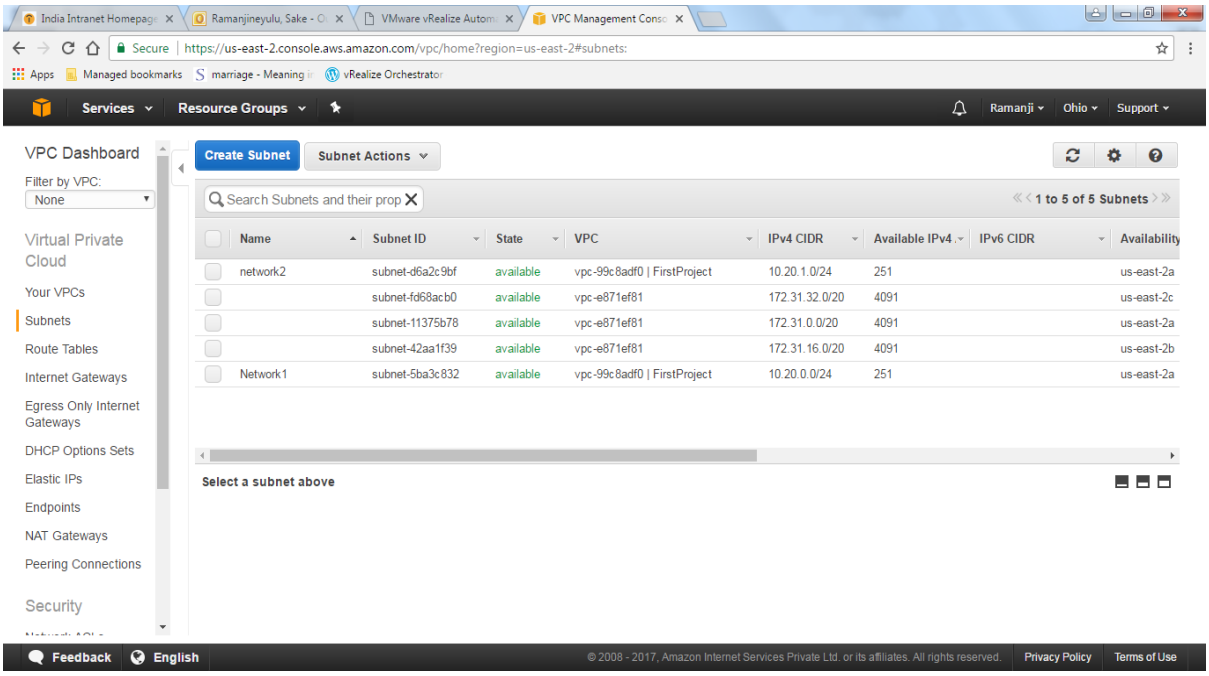
- Give your name for your VPC.
- Specify the range for your VPC in the form of CIDR block.

Configuration and management of subnets and network ACLs and Configuration and management of security groups

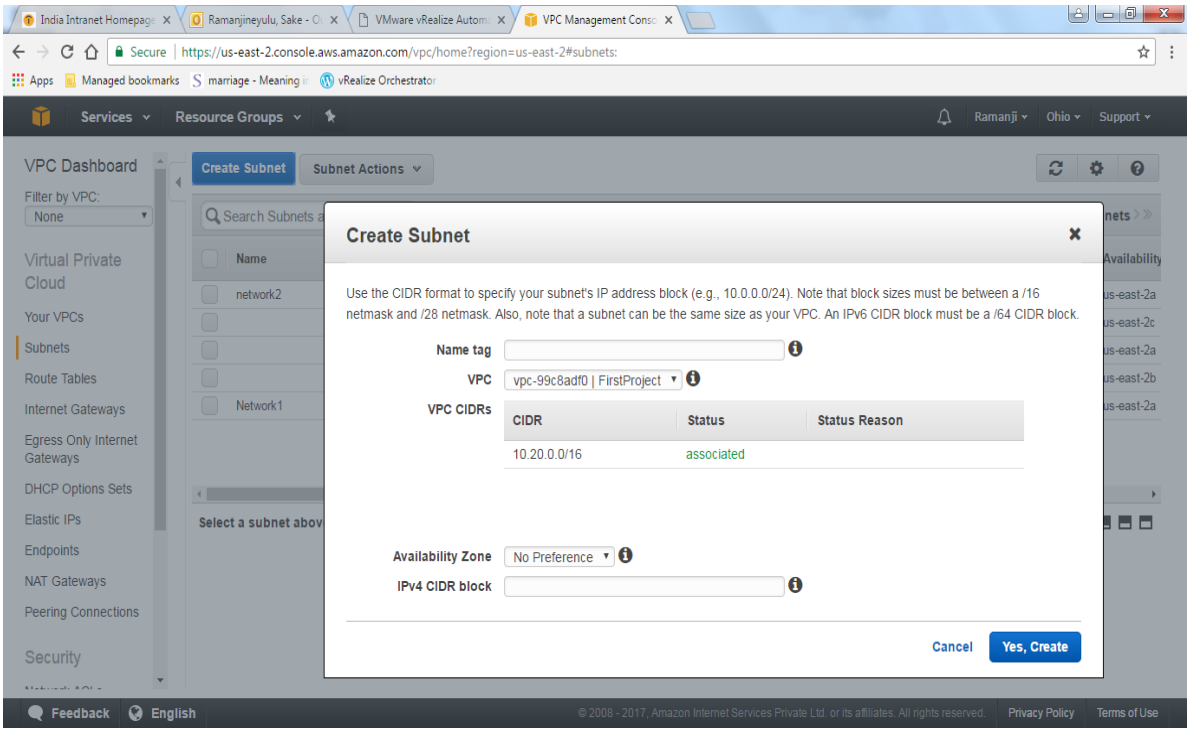
2.3.2 Subnets

- Subnet is certain Range of IP address.

- When your creating subnets you have to specify the range for your subnet.
- You can create public subnets as well as private subnets.



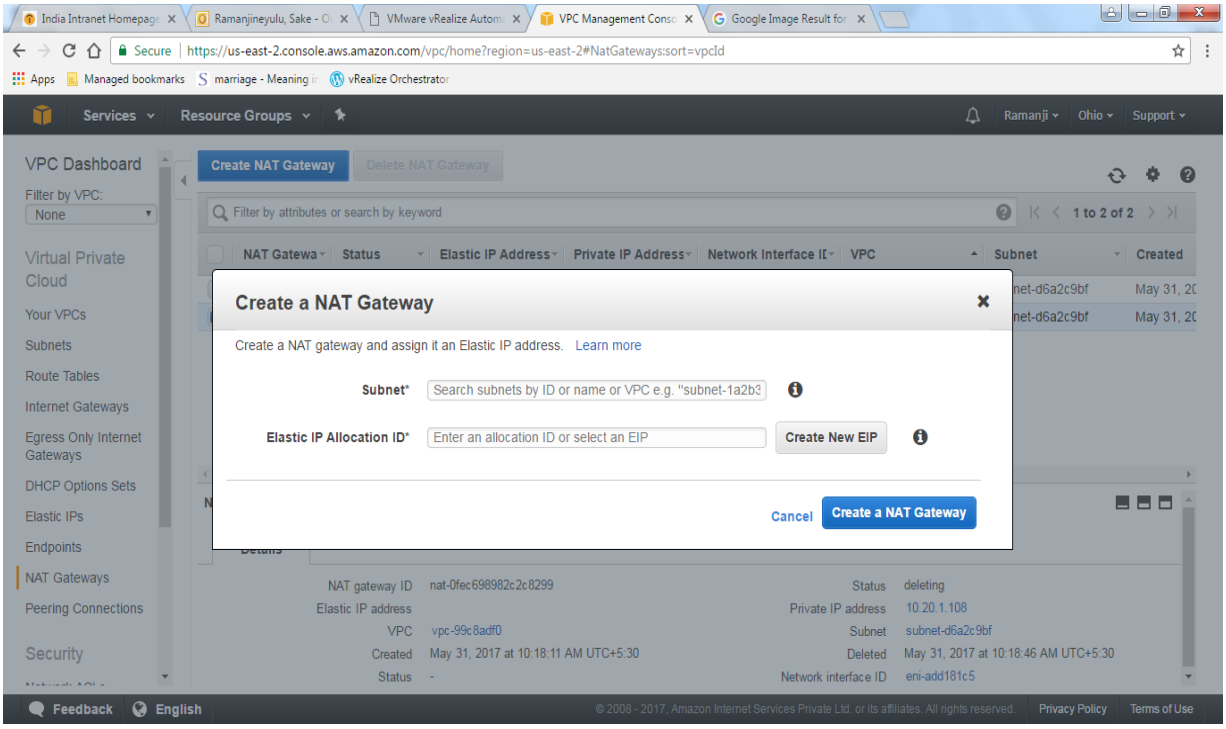
Click on Create Subnet



- Give your name for your subnet
- Select VPC
- Select availability zone
- Specify IP address range for your subnet.

2.3.3 NAT Gateway

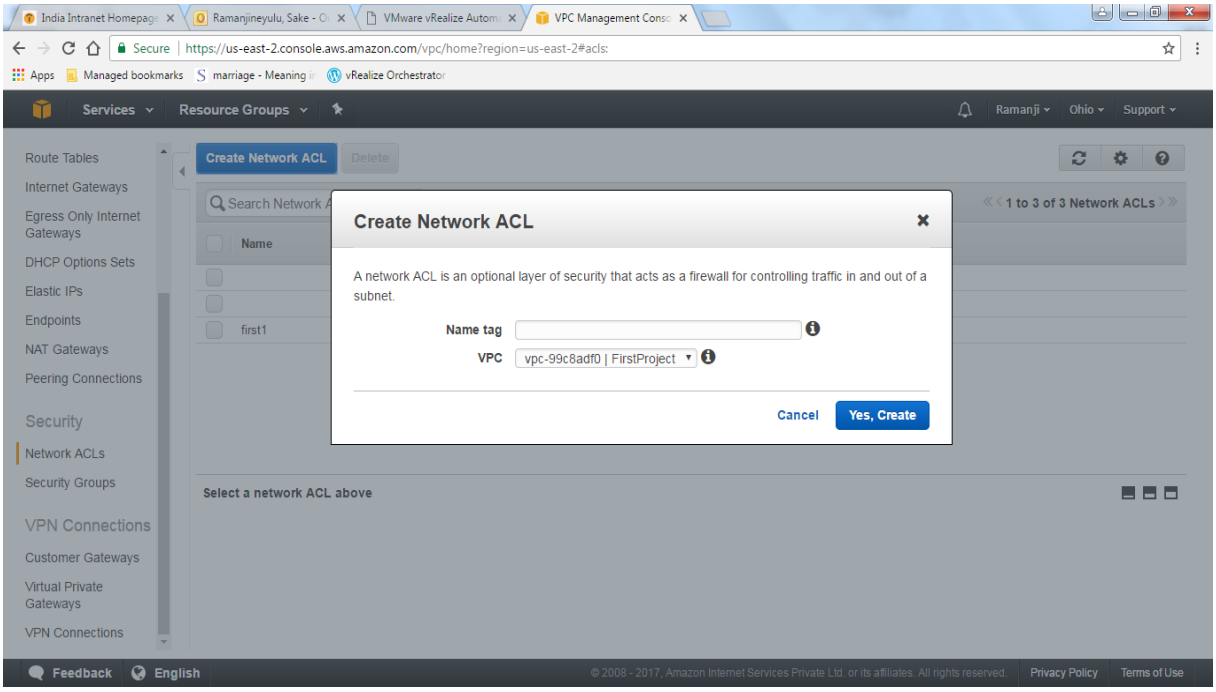
- It is used to enable the instances in a private subnet to connect internet or public cloud.



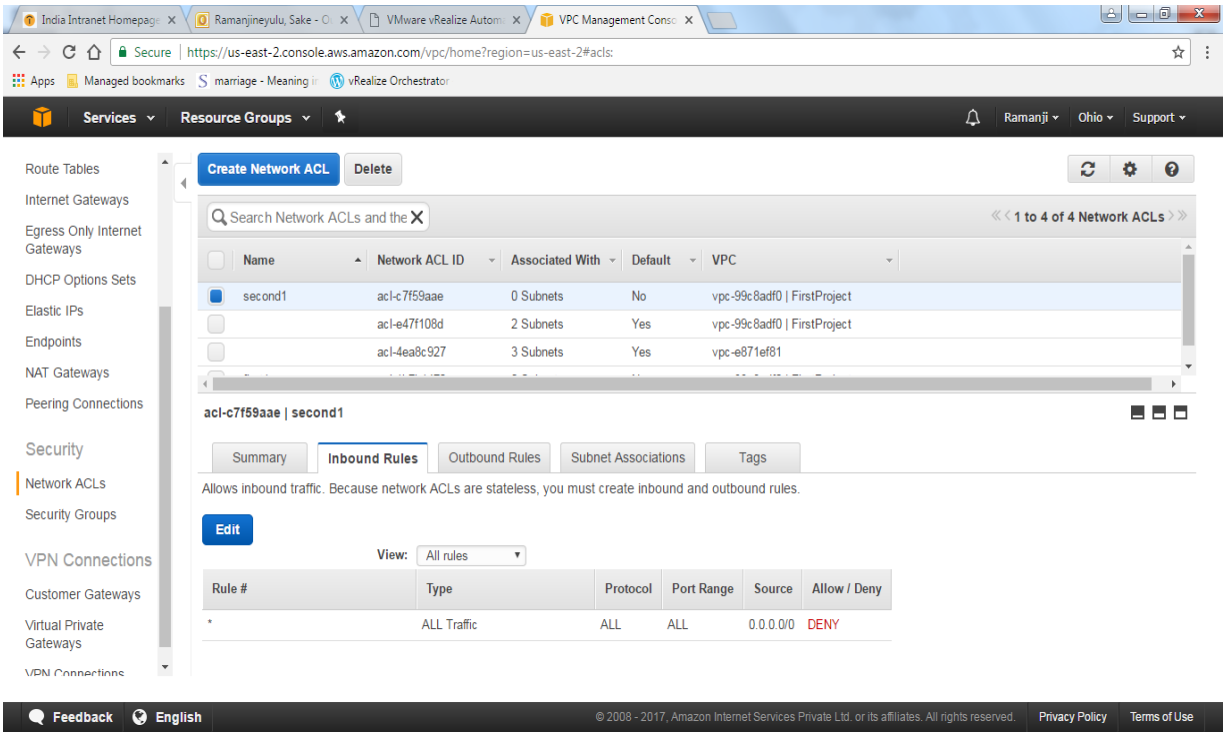
- Select the subnet which you want connect your NAT gateway
- Select Elastic IP if it is already created or you can create new one from here.

NACL (Network Access control list)

- This list control the traffic at the VPC level.
- You have to specify the inbound and outbound rules for VPC in the form of three digits rule numbers.



- Give name for your ACL
- Select your VPC.
- After clicking on Yes, create



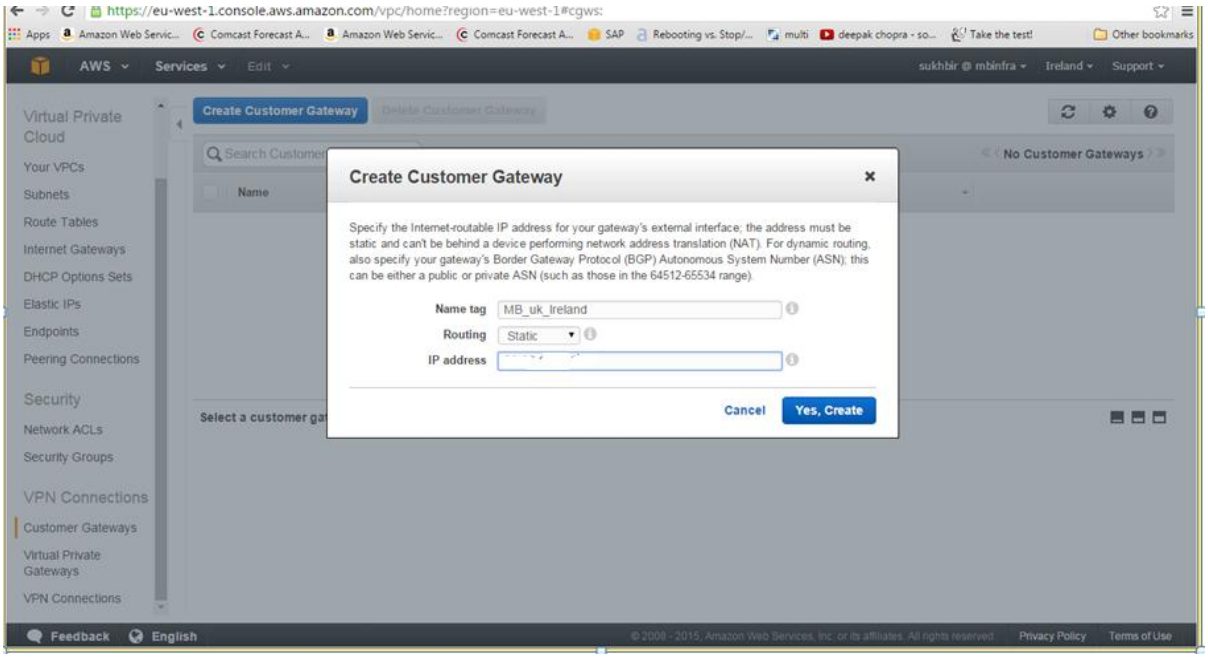
- Click on edit and give inbound rules for your VPC which control the incoming traffic to your VPC.

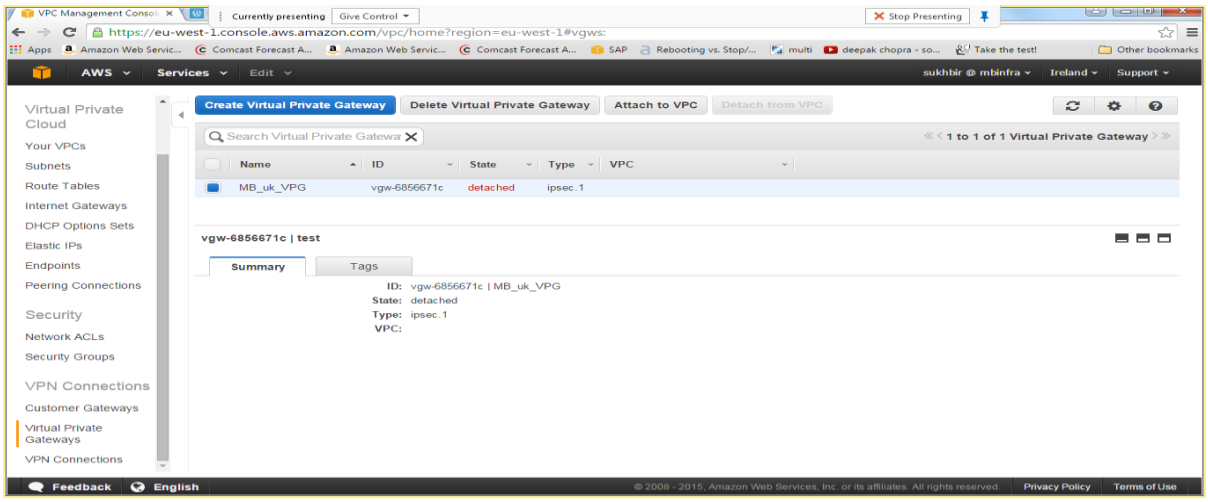
2.3.4 Configuration of AWS VPN devices and on premise VPN devices

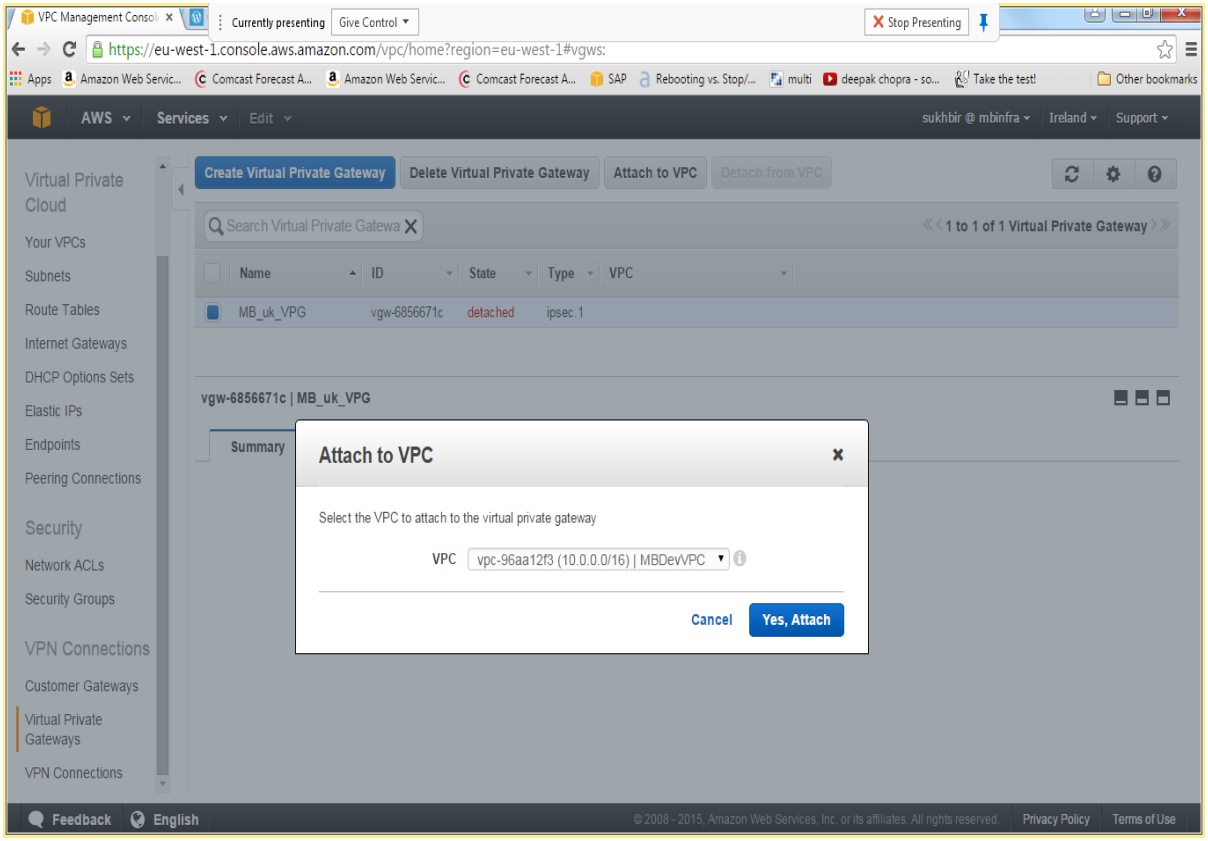
To set up a VPN connection, you need to complete the following steps:

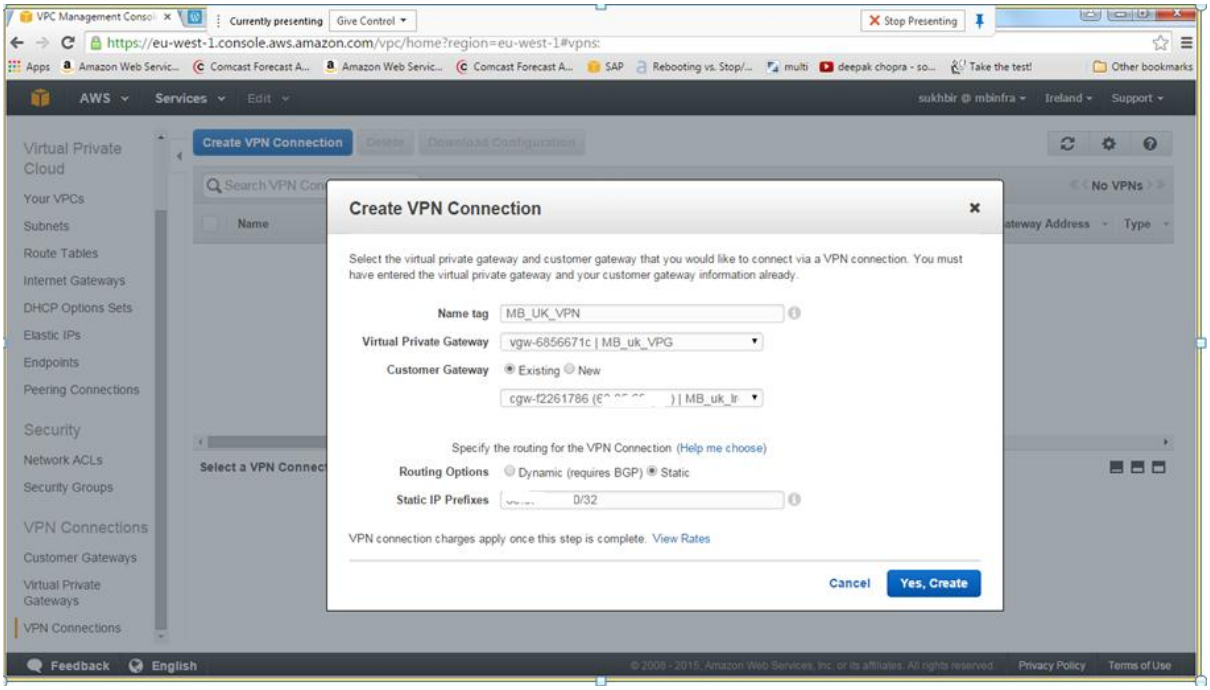
Step 1: Create a Customer Gateway
Step 2: Create a Virtual Private Gateway
Step 3: Enable Route Propagation in Your Route Table
Step 4: Update Your Security Group to Enable Inbound SSH, RDP and ICMP Access
Step 5: Create a VPN Connection and Configure the Customer Gateway
Step 6: Launch an Instance into Your Subnet

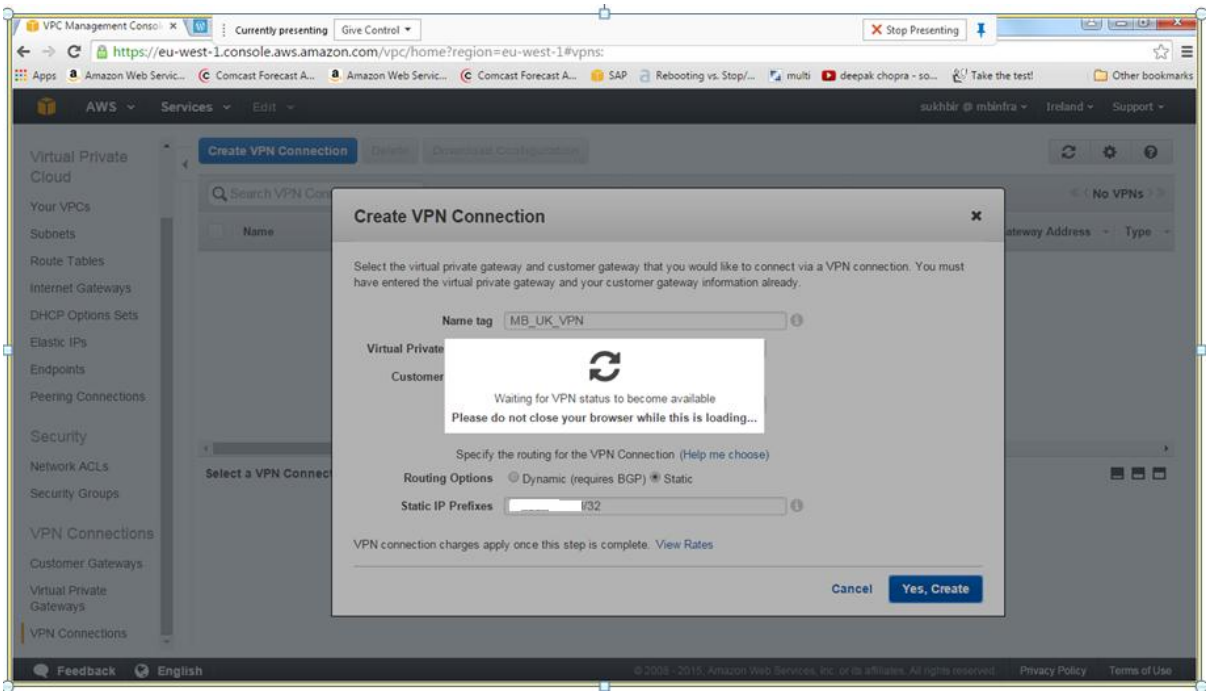
Go to the VPC and click on the VPN option to create the **Create Customer Gateway**

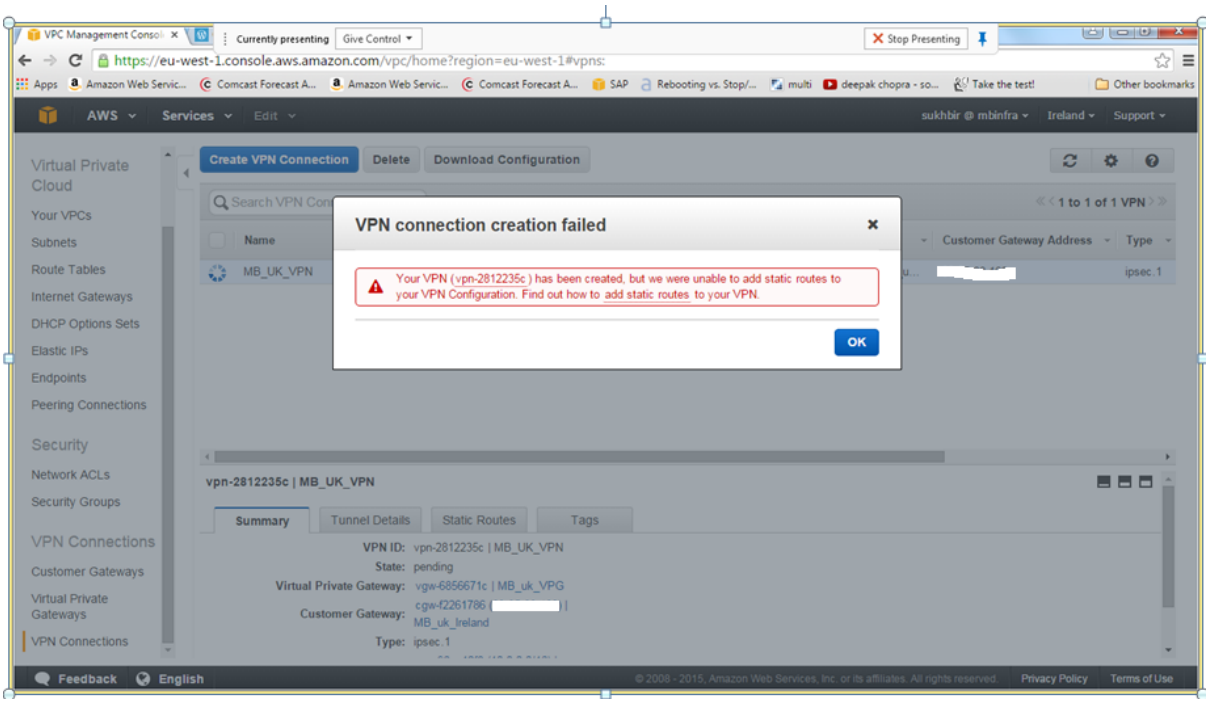



VPC Management Console
Currently presenting
Give Control
Stop Presenting
https://eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#vgws:
AWS
Services
Edit
sukhbir @ mbinfra
Ireland
Support
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Peering Connections
Security
Network ACLs
Security Groups
VPN Connections
Customer Gateways
Virtual Private Gateways
VPN Connections
Create Virtual Private Gateway
Delete Virtual Private Gateway
Attach to VPC
Detach from VPC
Search Virtual Private Gateways
1 to 1 of 1 Virtual Private Gateway
Name
ID
State
Type
VPC
MB_uk_VPG
vgw-6856671c
detached
ipsec.1
vgw-6856671c | test
Summary
Tags
ID: vgw-6856671c | MB_uk_VPG
State: detached
Type: ipsec.1
VPC:
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

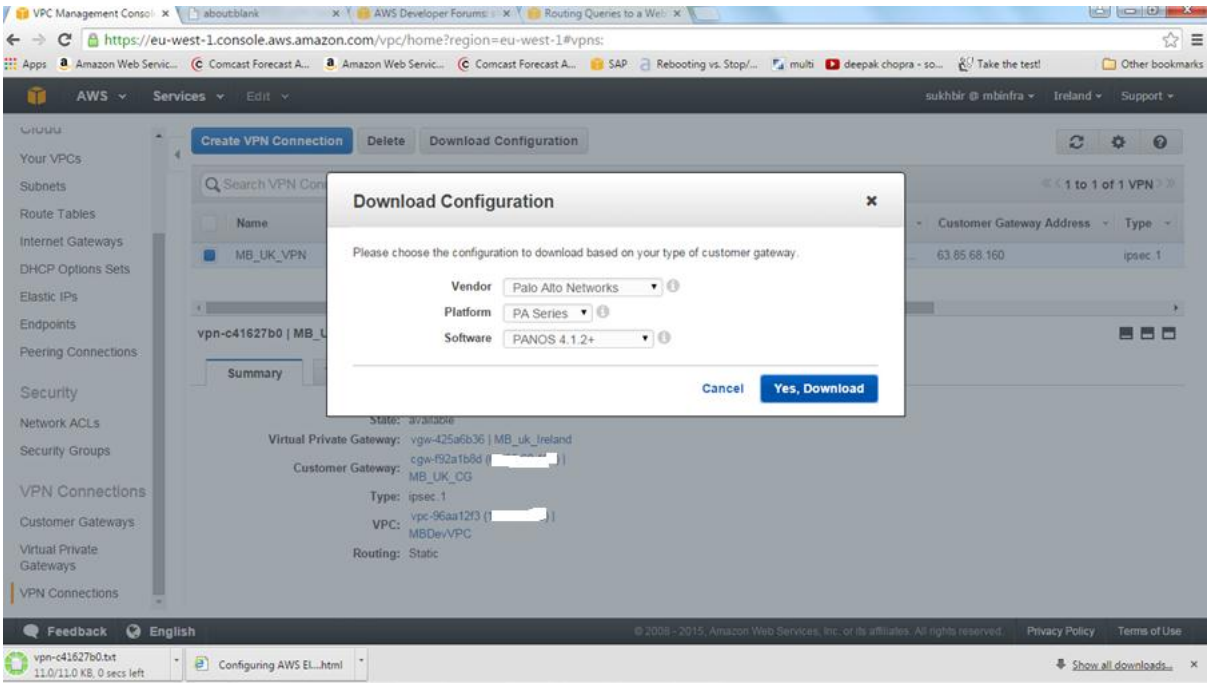

VPC Management Console
Currently presenting
Give Control
Stop Presenting
https://eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#vgws:
AWS
Services
Edit
sukhbir @ mbinfra
Ireland
Support
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Peering Connections
Security
Network ACLs
Security Groups
VPN Connections
Customer Gateways
Virtual Private Gateways
VPN Connections
Create Virtual Private Gateway
Delete Virtual Private Gateway
Attach to VPC
Detach from VPC
Search Virtual Private Gateways
1 to 1 of 1 Virtual Private Gateway
Name
ID
State
Type
VPC
MB_uk_VPG
vgw-6856671c
detached
ipsec.1
vgw-6856671c | MB_uk_VPG
Summary
Attach to VPC
Select the VPC to attach to the virtual private gateway
VPC
vpc-96aa12f3 (10.0.0.0/16) | MBDevVPC
Cancel
Yes, Attach
Feedback
English
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Create VPN Connection
Select the virtual private gateway and customer gateway that you would like to connect via a VPN connection. You must have entered the virtual private gateway and your customer gateway information already.
Name tag
MB_UK_VPN
Virtual Private Gateway
vgw-6856671c | MB_uk_VPG
Customer Gateway
Existing
New
Specify the routing for the VPN Connection (Help me choose)
Routing Options
Dynamic (requires BGP)
Static
Static IP Prefixes
VPN connection charges apply once this step is complete. View Rates
Cancel
Yes, Create


Create VPN Connection
Select the virtual private gateway and customer gateway that you would like to connect via a VPN connection. You must have entered the virtual private gateway and your customer gateway information already.
Name tag
MB_UK_VPN
Waiting for VPN status to become available
Please do not close your browser while this is loading...
Specify the routing for the VPN Connection (Help me choose)
Routing Options
Dynamic (requires BGP)
Static
Static IP Prefixes
VPN connection charges apply once this step is complete. View Rates
Cancel
Yes, Create



Wait till it is available



Download the VPN configuration (Ex: vpn-c415670_GUI.txt) and provide to the Networking team to establish the connection

You can launch the AWs instance and test the connectivity to on premise.

3. IDENTITY AND SECURITY MANAGEMENT

AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS services.

3.1 Objective

To understand the access control creation of users and roles using AWS IAM.

3.2 Assumption

- AWS Account is available with AWS Identity access Management (IAM).

3.3 Procedure

Step 1: Create an Administrator group and give permission to access all of your AWS account's resources.

Step 2: Create a user for yourself add that user to the Administrators group.

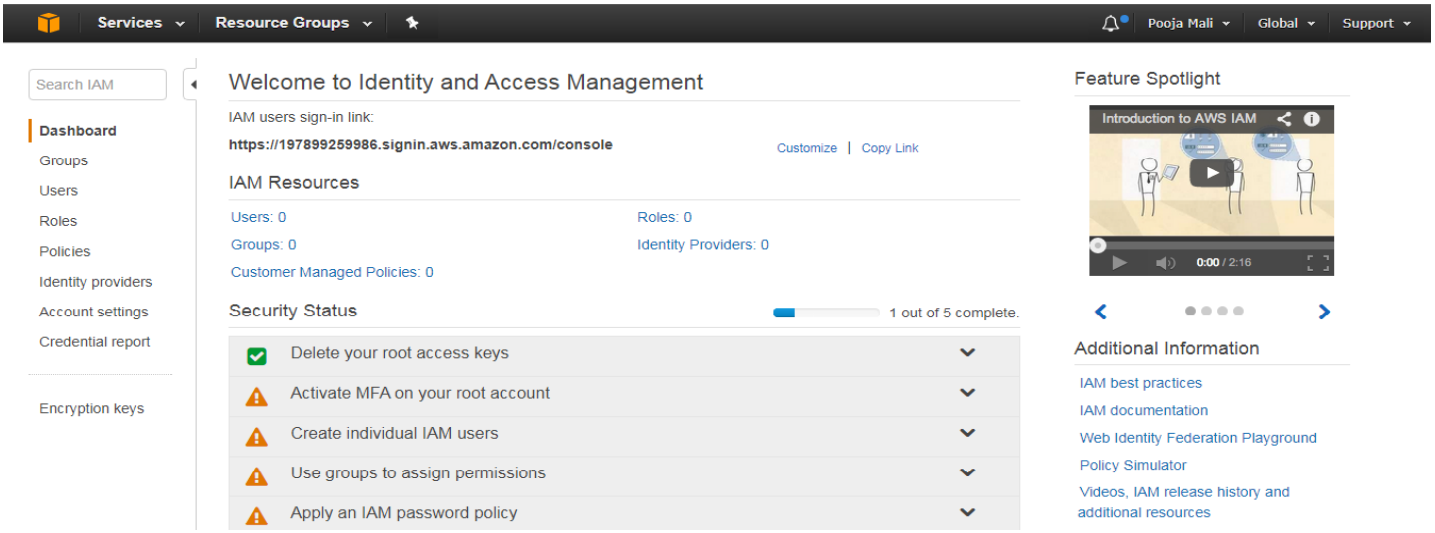
Step 3: Create a password for your user so you can sign in to the AWS Management Console.

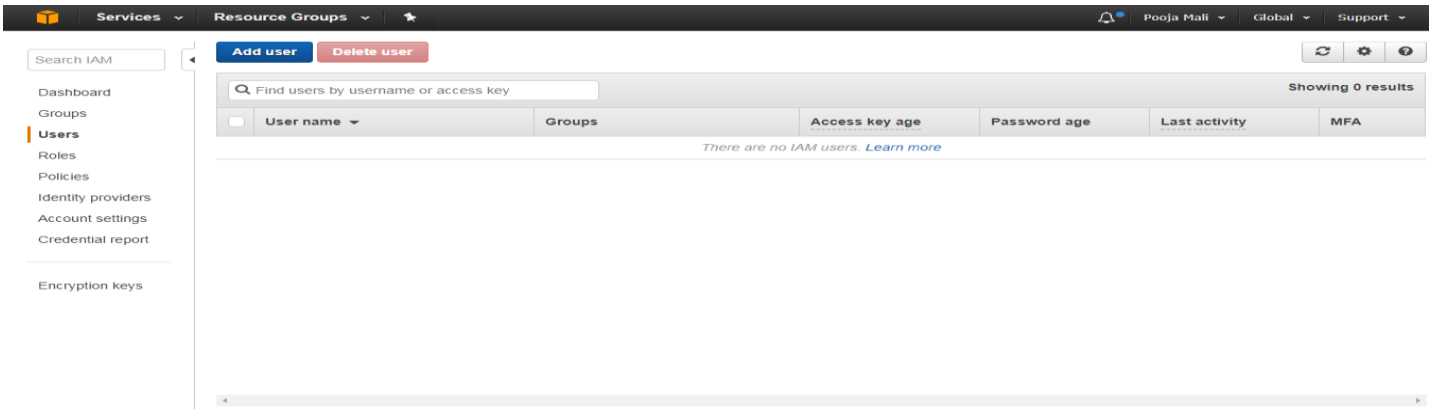
Step 4: Create Group and add users to the group.

Step 5: Create Policy and Roles.

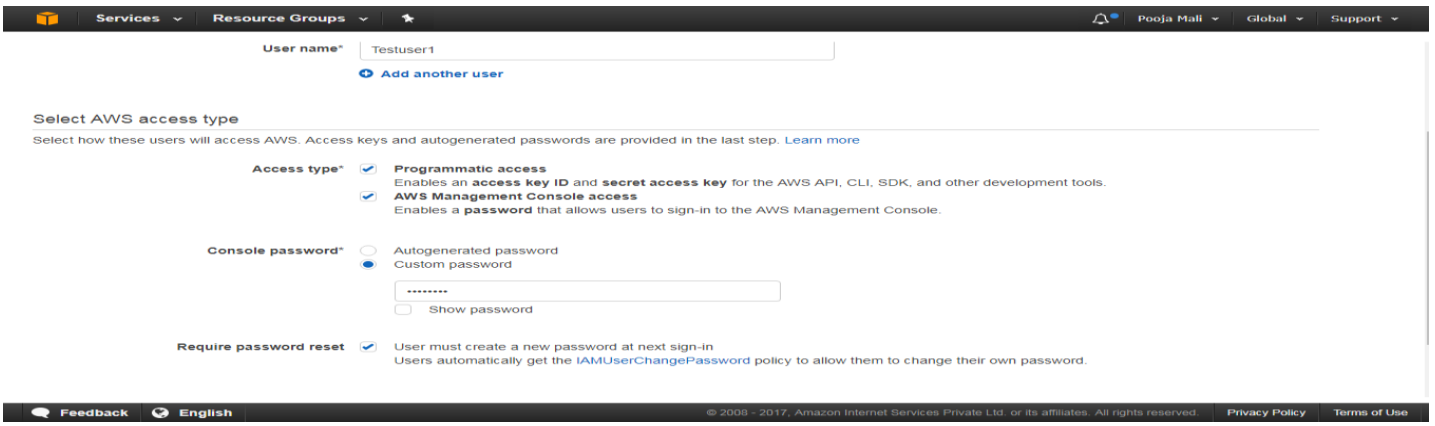
Step 1**:**

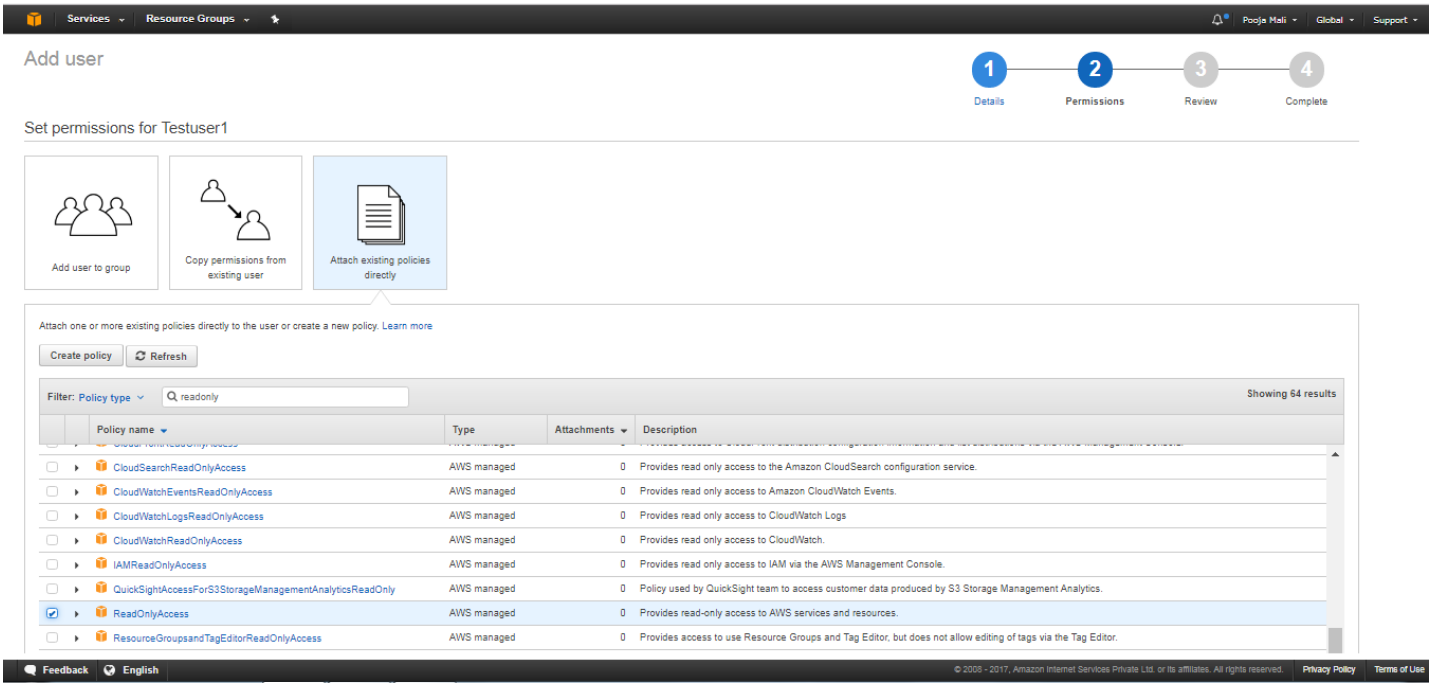
Go to the Security, Identity and Compliances and click on the option IAM.

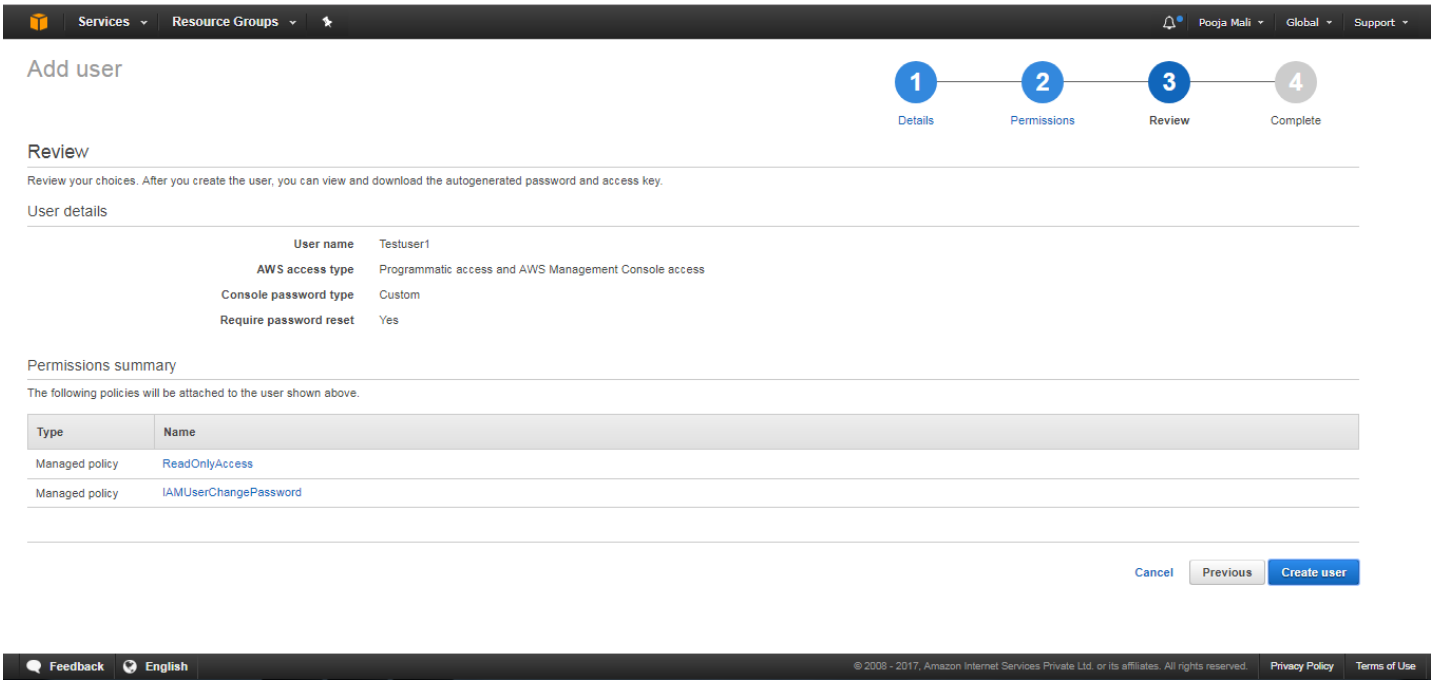

Services
Resource Groups
Pooja Mali
Global
Support
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Welcome to Identity and Access Management
IAM users sign-in link:
https://197899259986.signin.aws.amazon.com/console
Customize | Copy Link
IAM Resources
Users: 0
Roles: 0
Groups: 0
Identity Providers: 0
Customer Managed Policies: 0
Security Status
1 out of 5 complete.
Delete your root access keys
Activate MFA on your root account
Create individual IAM users
Use groups to assign permissions
Apply an IAM password policy
Feature Spotlight
Introduction to AWS IAM
0:00 / 2:16
Additional Information
IAM best practices
IAM documentation
Web Identity Federation Playground
Policy Simulator
Videos, IAM release history and additional resources

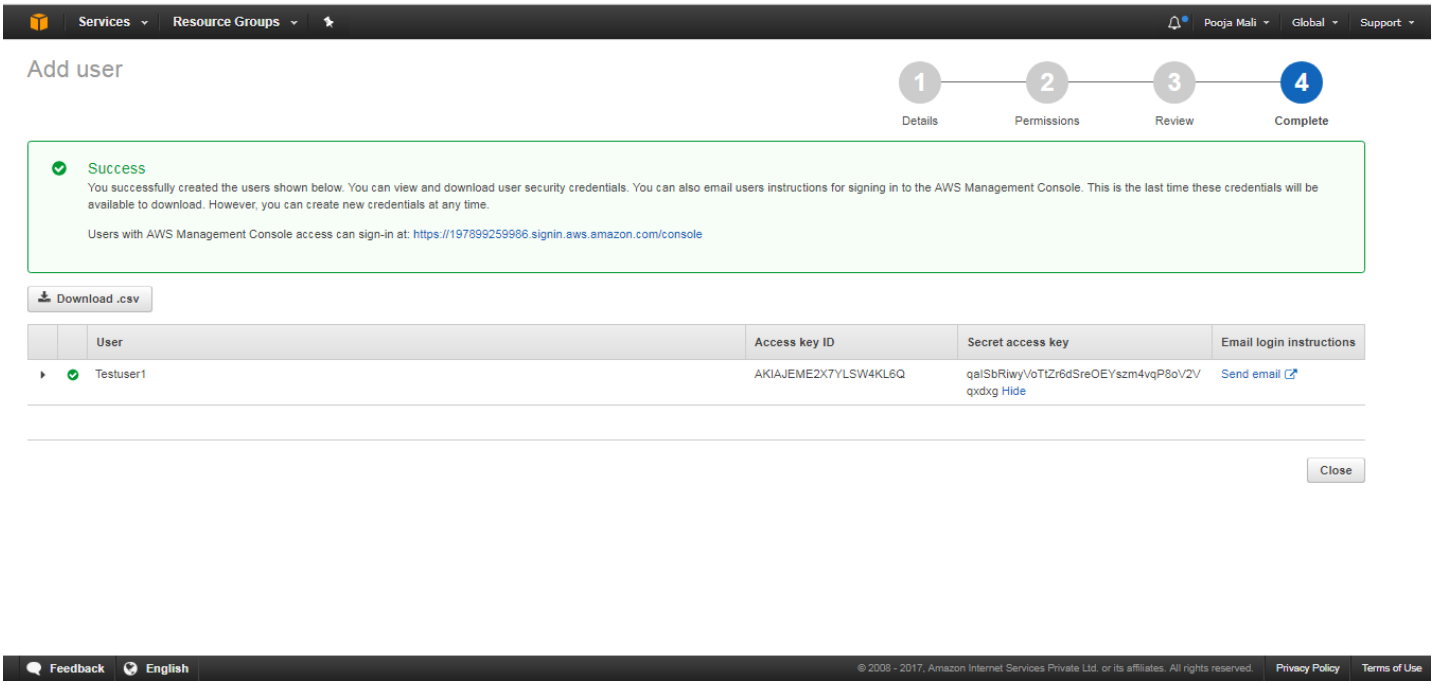

Services
Resource Groups
Pooja Mali
Global
Support
Search IAM
Add user
Delete user
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Find users by username or access key
Showing 0 results
User name
Groups
Access key age
Password age
Last activity
MFA
There are no IAM users. Learn more

Step 2:

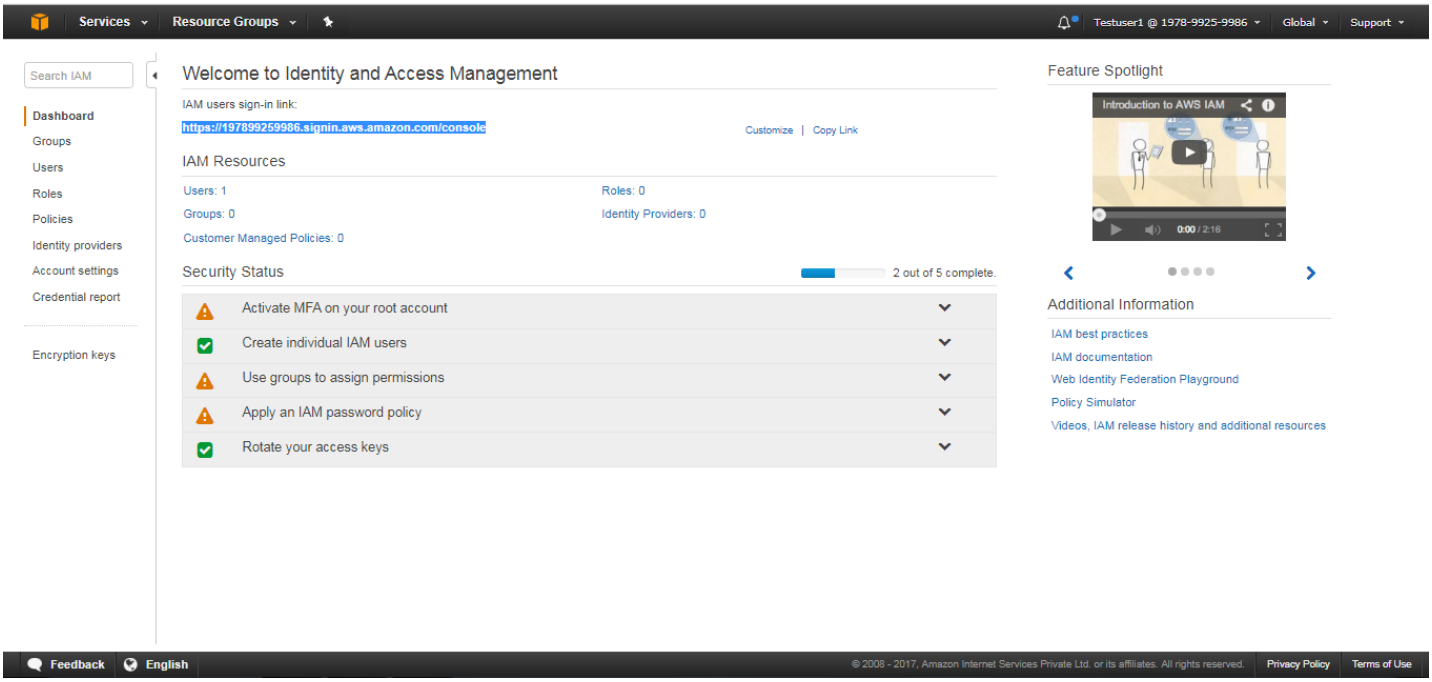

Services
Resource Groups
Pooja Mali
Global
Support
User name*
Testuser1
Add another user
Select AWS access type
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more
Access type*
Programmatic access
Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.
AWS Management Console access
Enables a password that allows users to sign-in to the AWS Management Console.
Console password*
Autogenerated password
Custom password
Show password
Require password reset
User must create a new password at next sign-in
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

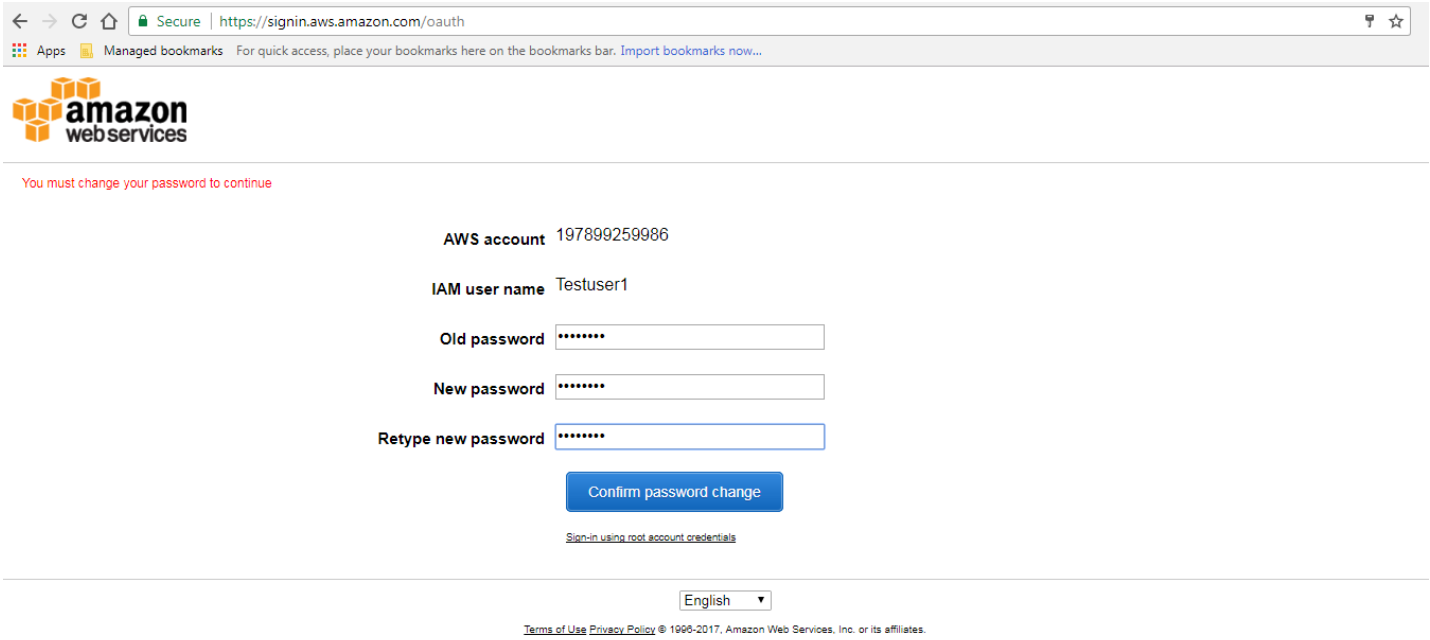

Services
Resource Groups
Pooja Mali
Global
Support
Add user
1 Details
2 Permissions
3 Review
4 Complete
Set permissions for Testuser1
Add user to group
Copy permissions from existing user
Attach existing policies directly
Attach one or more existing policies directly to the user or create a new policy. Learn more
Create policy
Refresh
Filter: Policy type
readonly
Showing 64 results
Policy name
Type
Attachments
Description
CloudSearchReadOnlyAccess AWS managed 0 Provides read only access to the Amazon CloudSearch configuration service.
CloudWatchEventsReadOnlyAccess AWS managed 0 Provides read only access to Amazon CloudWatch Events.
CloudWatchLogsReadOnlyAccess AWS managed 0 Provides read only access to CloudWatch Logs
CloudWatchReadOnlyAccess AWS managed 0 Provides read only access to CloudWatch.
IAMReadOnlyAccess AWS managed 0 Provides read only access to IAM via the AWS Management Console.
QuickSightAccessForS3StorageManagementAnalyticsReadOnly AWS managed 0 Policy used by QuickSight team to access customer data produced by S3 Storage Management Analytics.
ReadOnlyAccess AWS managed 0 Provides read-only access to AWS services and resources.
ResourceGroupsandTagEditorReadOnlyAccess AWS managed 0 Provides access to use Resource Groups and Tag Editor, but does not allow editing of tags via the Tag Editor.
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Services
Resource Groups
Pooja Mali
Global
Support
Add user
1 Details
2 Permissions
3 Review
4 Complete
Review
Review your choices. After you create the user, you can view and download the autogenerated password and access key.
User details
User name Testuser1
AWS access type Programmatic access and AWS Management Console access
Console password type Custom
Require password reset Yes
Permissions summary
The following policies will be attached to the user shown above.
Type
Name
Managed policy ReadOnlyAccess
Managed policy IAMUserChangePassword
Cancel
Previous
Create user
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

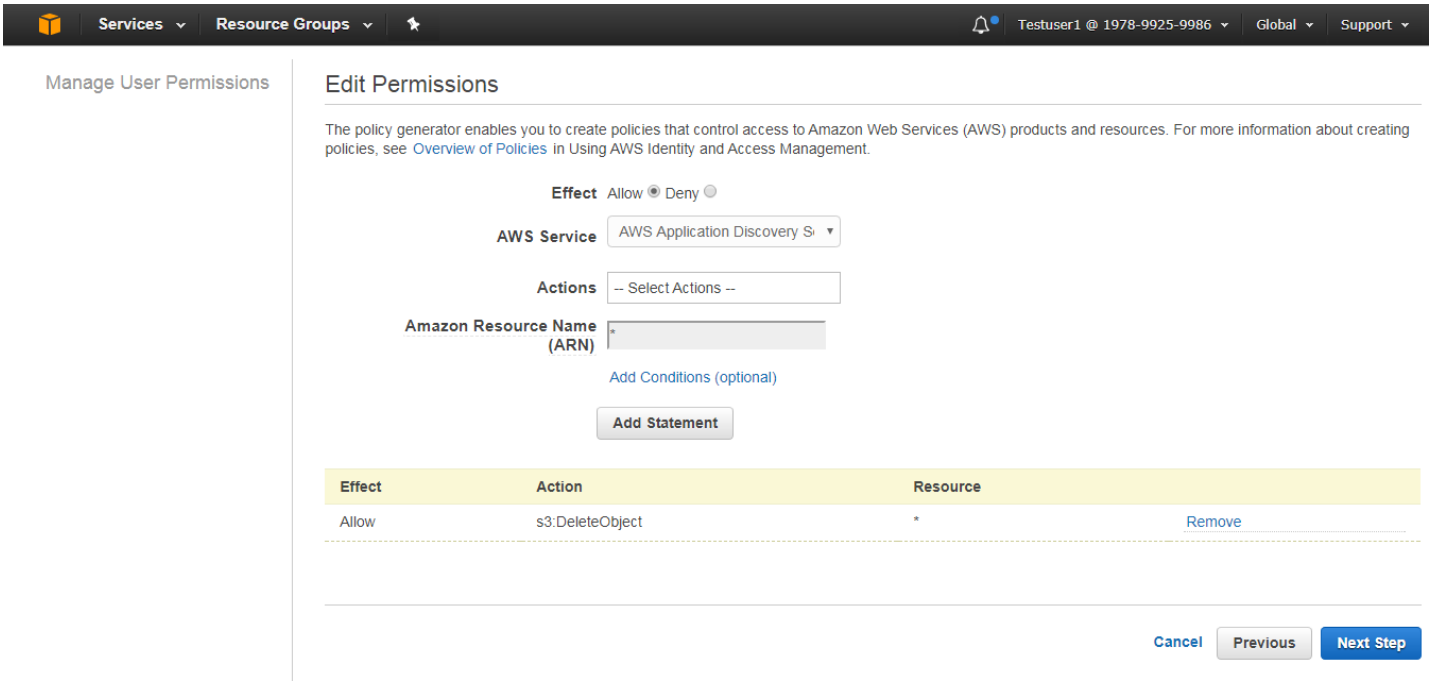


Go back to dashboard, copy the link and open it in new web browser.





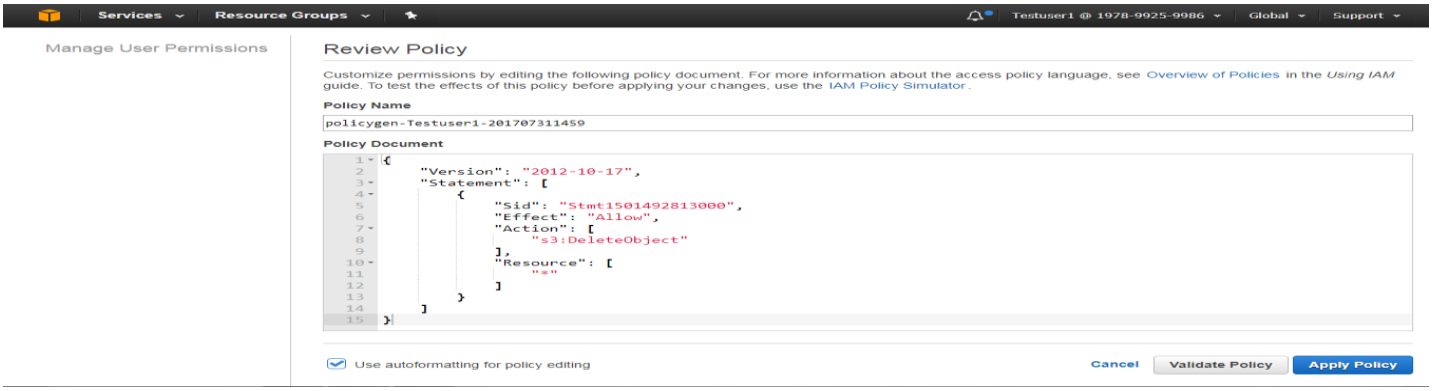
In home page of IAM, Go to User click on the option “add inline policy” next click on “select” option.



In above, “ * ” is to give all permissions on action selected. Otherwise to give specific permission on action selected then click on Amazon Resourse Name afterwards we get like below one.

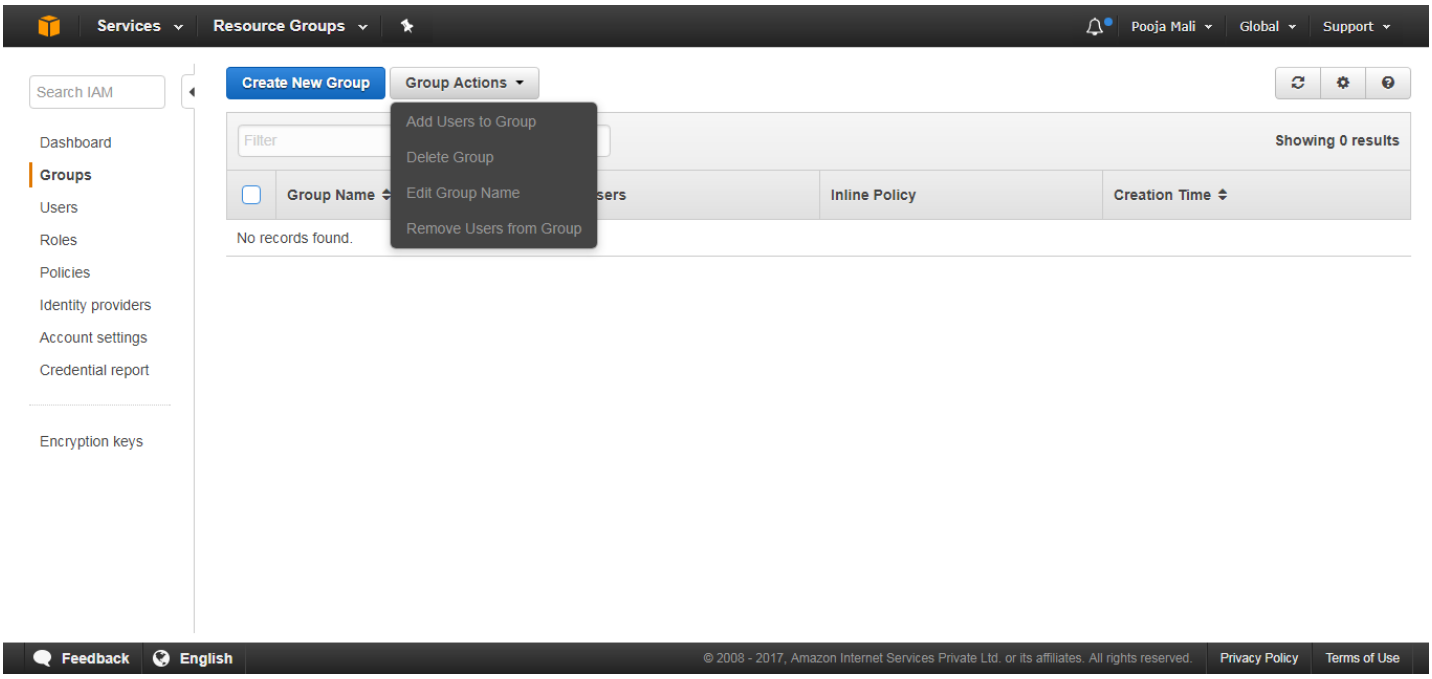
```
<!-- Object in an Amazon S3 bucket -->
arn:aws:s3:::my_corporate_bucket/exampleobject.png
```

Validate and Apply policy

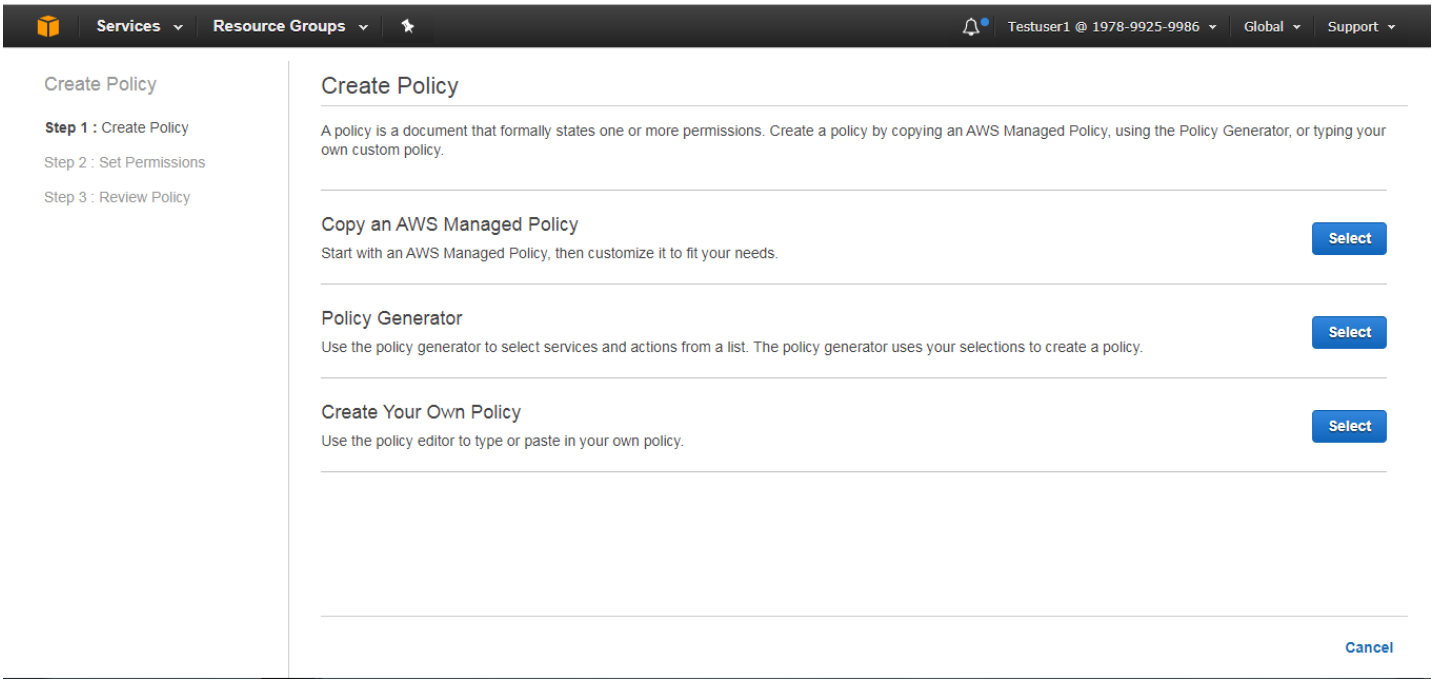


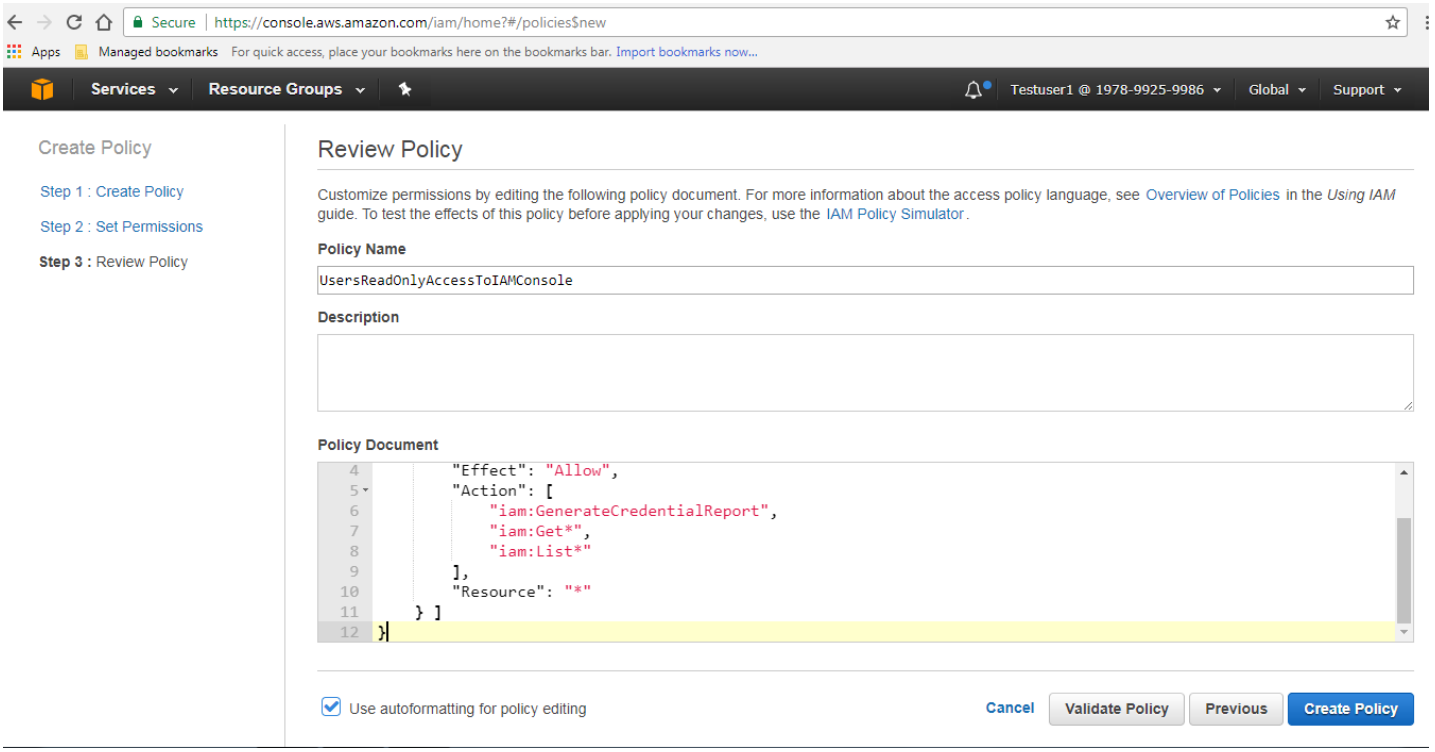
Step 3:

Go to Groups, Click on Group Actions Thus we will add user to the Group Created.



Step 5:

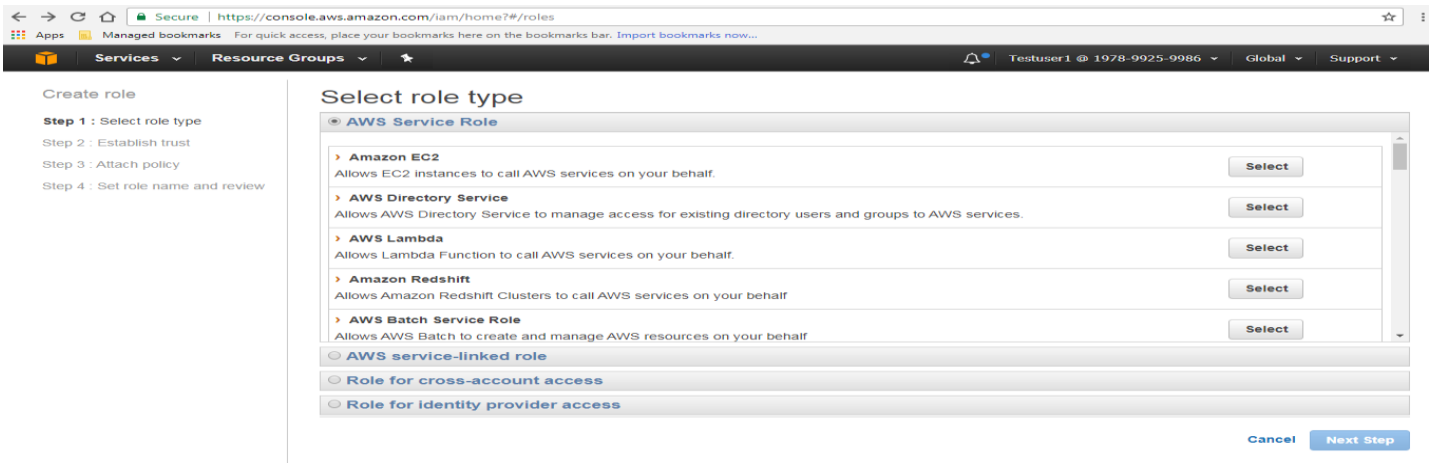

Services
Resource Groups
Testuser1 @ 1978-9925-9986
Global
Support
Create Policy
Step 1 : Create Policy
Step 2 : Set Permissions
Step 3 : Review Policy
Create Policy
A policy is a document that formally states one or more permissions. Create a policy by copying an AWS Managed Policy, using the Policy Generator, or typing your own custom policy.
Copy an AWS Managed Policy
Start with an AWS Managed Policy, then customize it to fit your needs.
Select
Policy Generator
Use the policy generator to select services and actions from a list. The policy generator uses your selections to create a policy.
Select
Create Your Own Policy
Use the policy editor to type or paste in your own policy.
Select
Cancel

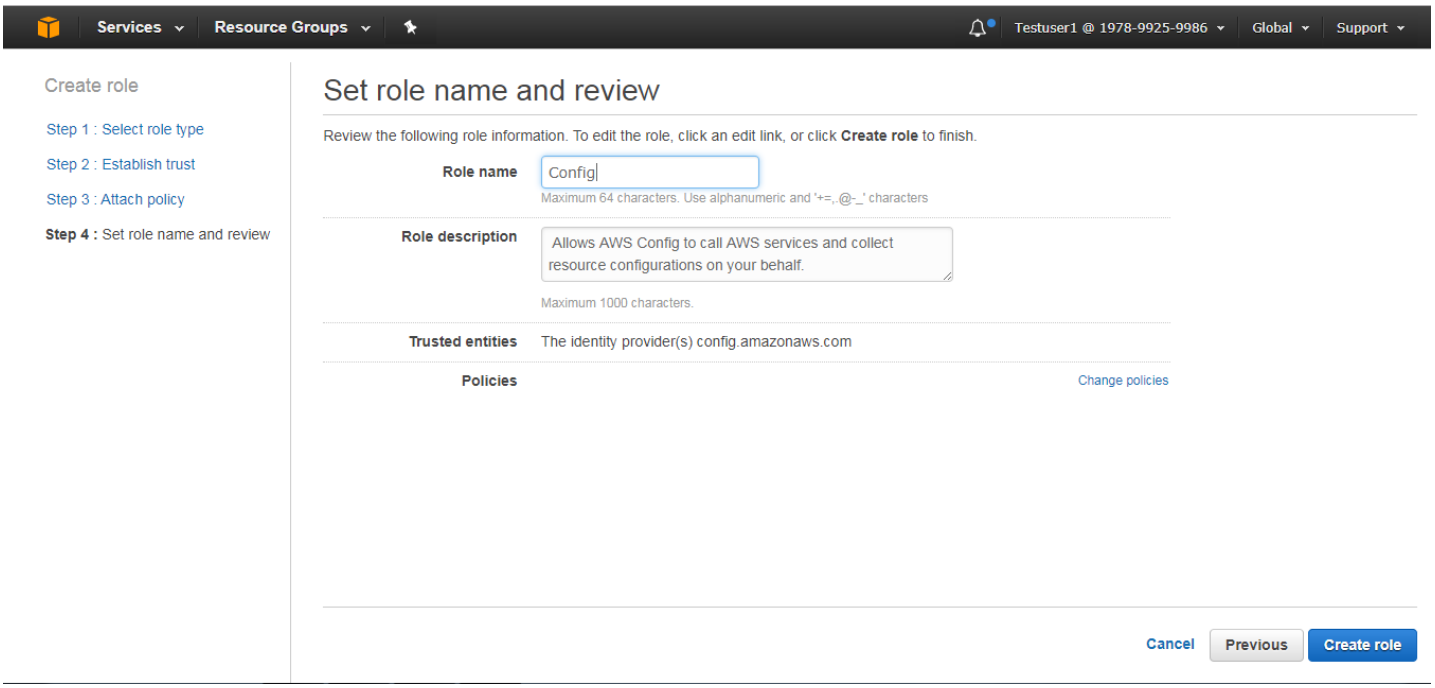

Secure
https://console.aws.amazon.com/iam/home?#/policies$new
Apps
Managed bookmarks
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Services
Resource Groups
Testuser1 @ 1978-9925-9986
Global
Support
Create Policy
Step 1 : Create Policy
Step 2 : Set Permissions
Step 3 : Review Policy
Review Policy
Customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in the Using IAM guide. To test the effects of this policy before applying your changes, use the IAM Policy Simulator.
Policy Name
UsersReadOnlyAccessToIAMConsole
Description
Policy Document
"Effect": "Allow",
"Action": [
"iam:GenerateCredentialReport",
"iam:Get*",
"iam:List*"
],
"Resource": "*"
}]
}
Use autoformatting for policy editing
Cancel
Validate Policy
Previous
Create Policy

Now validate and create a policy.

Step 5 Creating IAM Roles:

Before an IAM user, application, or service can use a role that you created, you must grant permissions to switch to the role. You can use any policy attached to one of an IAM user's groups or to the user itself to grant the necessary permissions. This section describes how to grant users permission to use a role, and then how the user can switch to a role using the AWS Management Console, the Tools for Windows PowerShell, the AWS Command Line Interface (AWS CLI) and the AssumeRole AP.





4. RDS SUPPORT

Amazon Relational Database Service (Amazon **RDS**) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud.

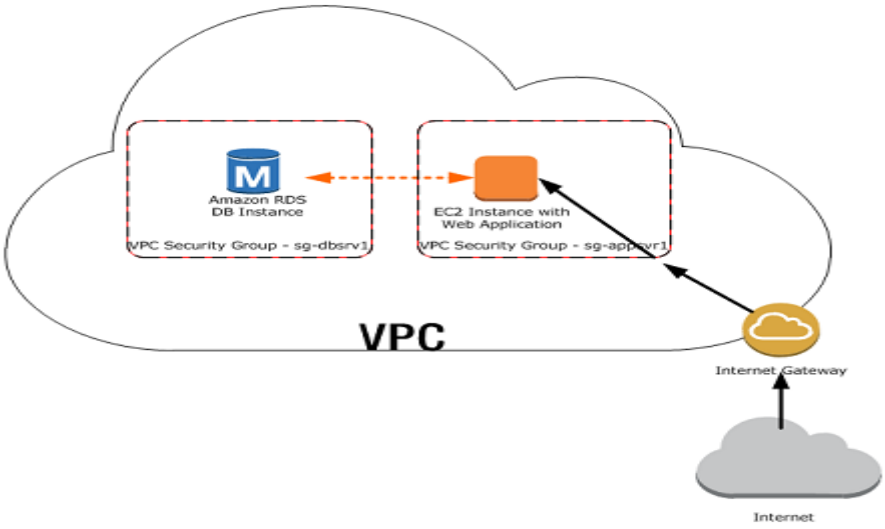
4.1 Objective

To provide the high level guidance's to Monitoring usage, health and Provisioning of RDS Instances.

4.2 Assumption

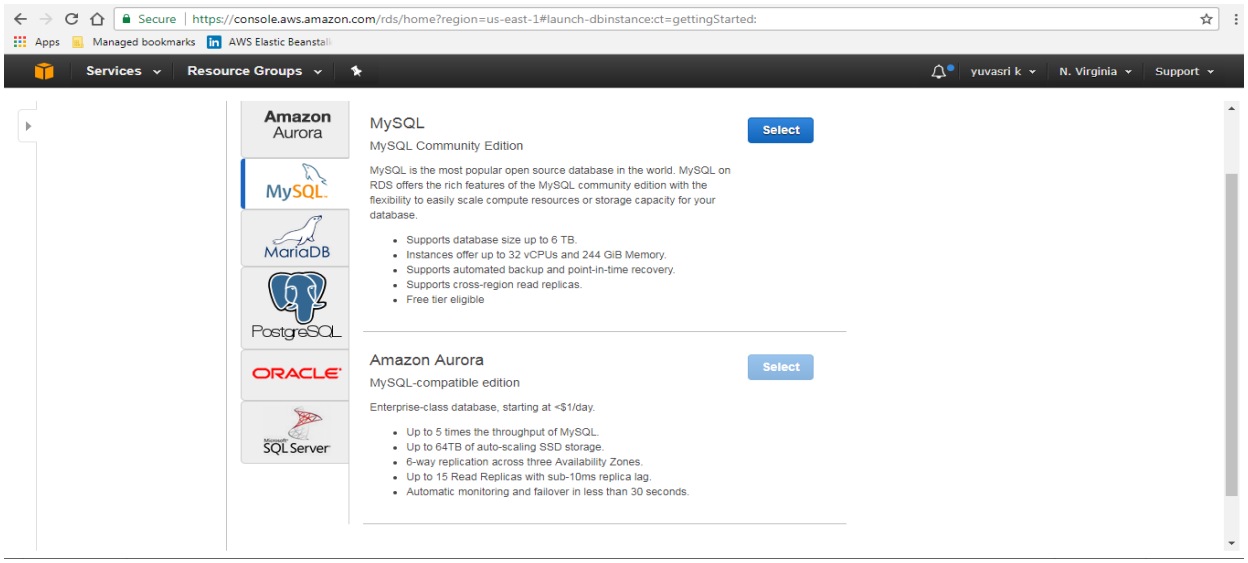
- Amazon RDS is available on several database instance types, optimized for Memory, Performance and I/O, and Provides with six familiar database engines to choose from Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle and Microsoft SQL Server.

The below diagram shows how the AWS RDS is connected:

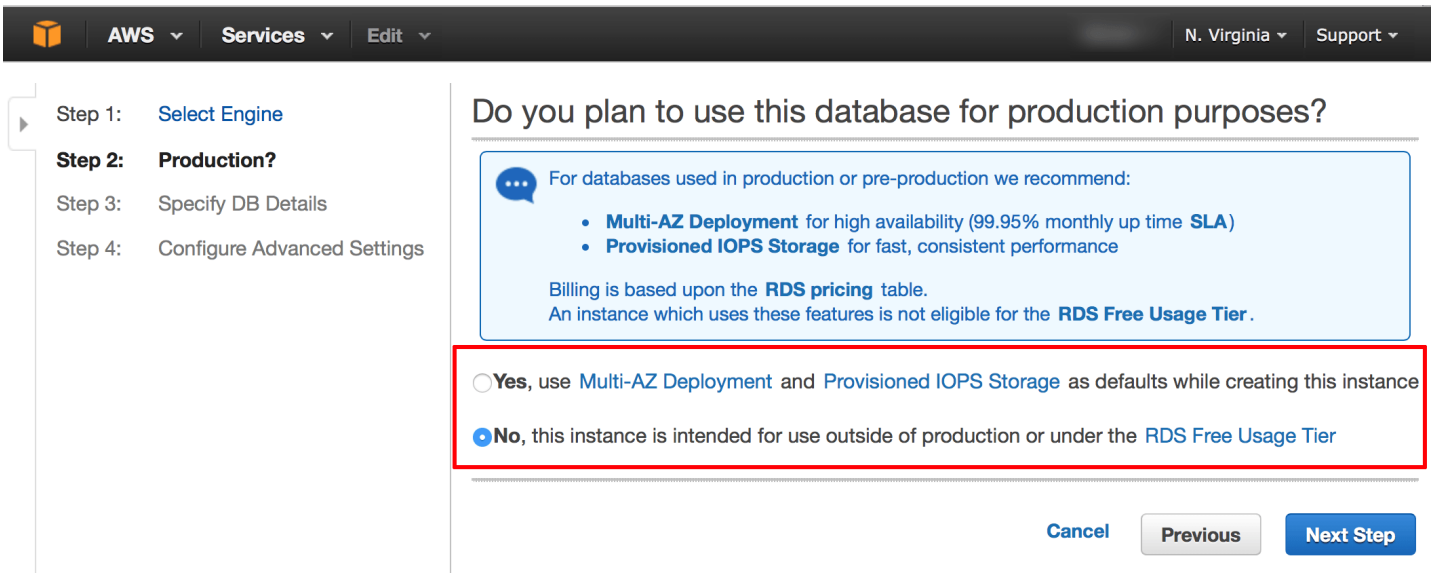


4.3 Procedure

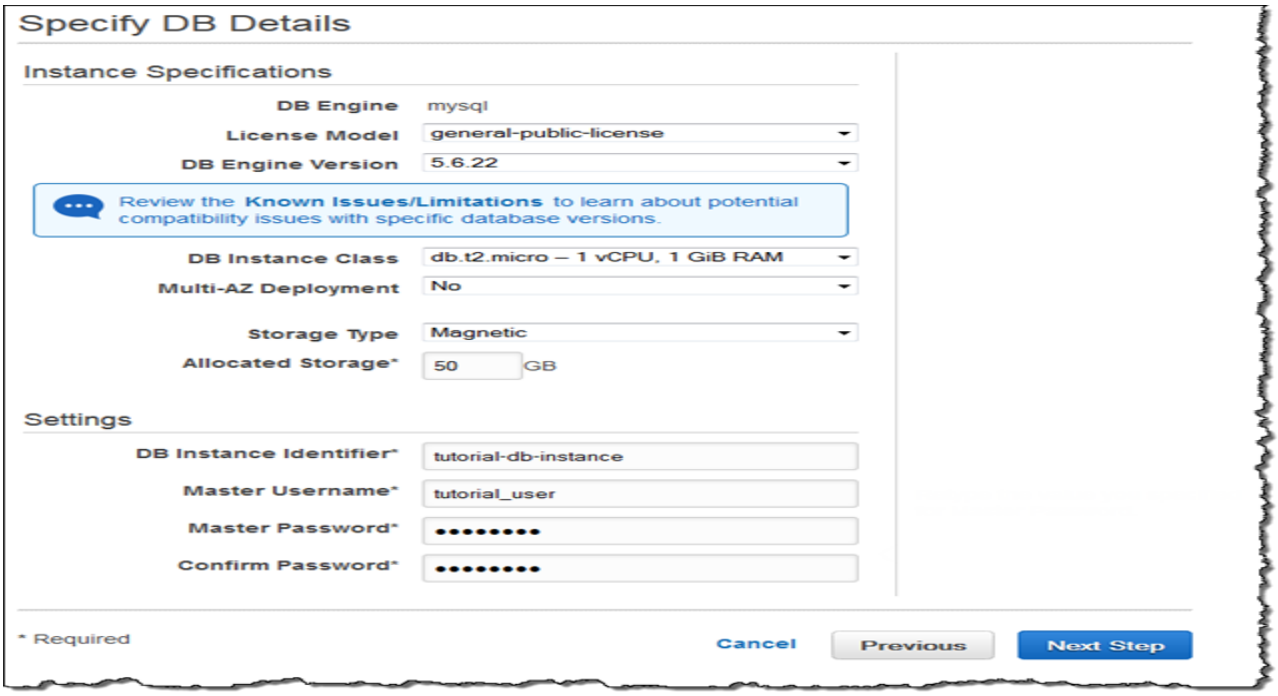
Go to the RDS and click on the DB Instance and follow the below steps:



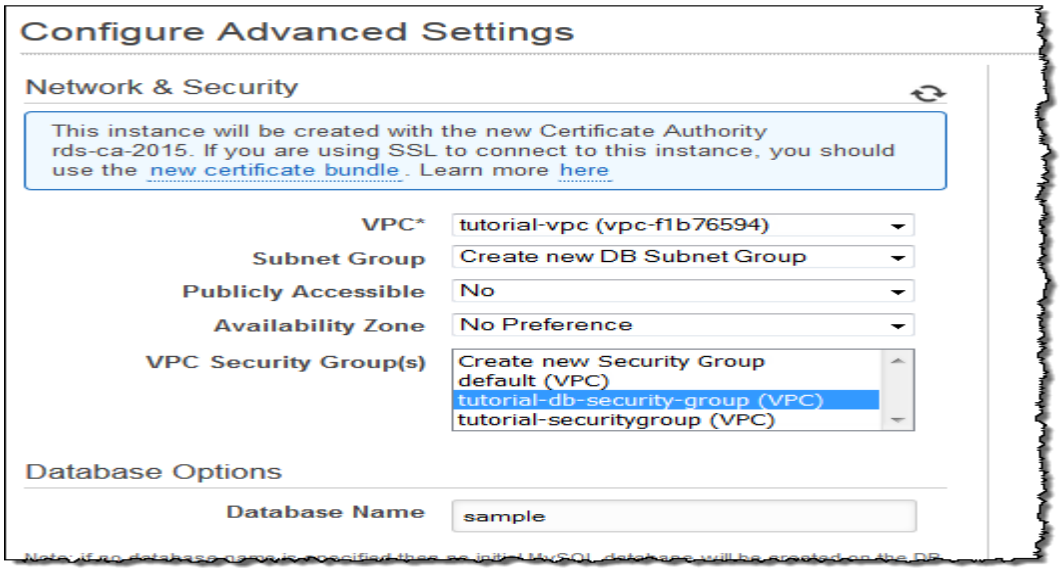
Select the type of the database to be used for the launching the RDS Instance.



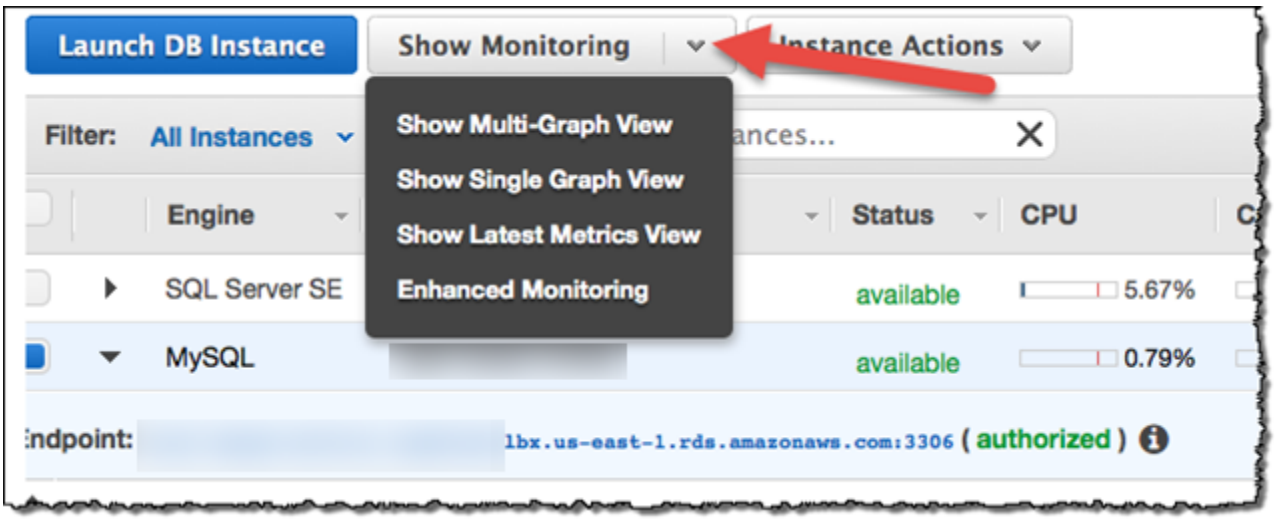
Specify the production type, so that we will come to know weather database is used for production purpose or outside the production.



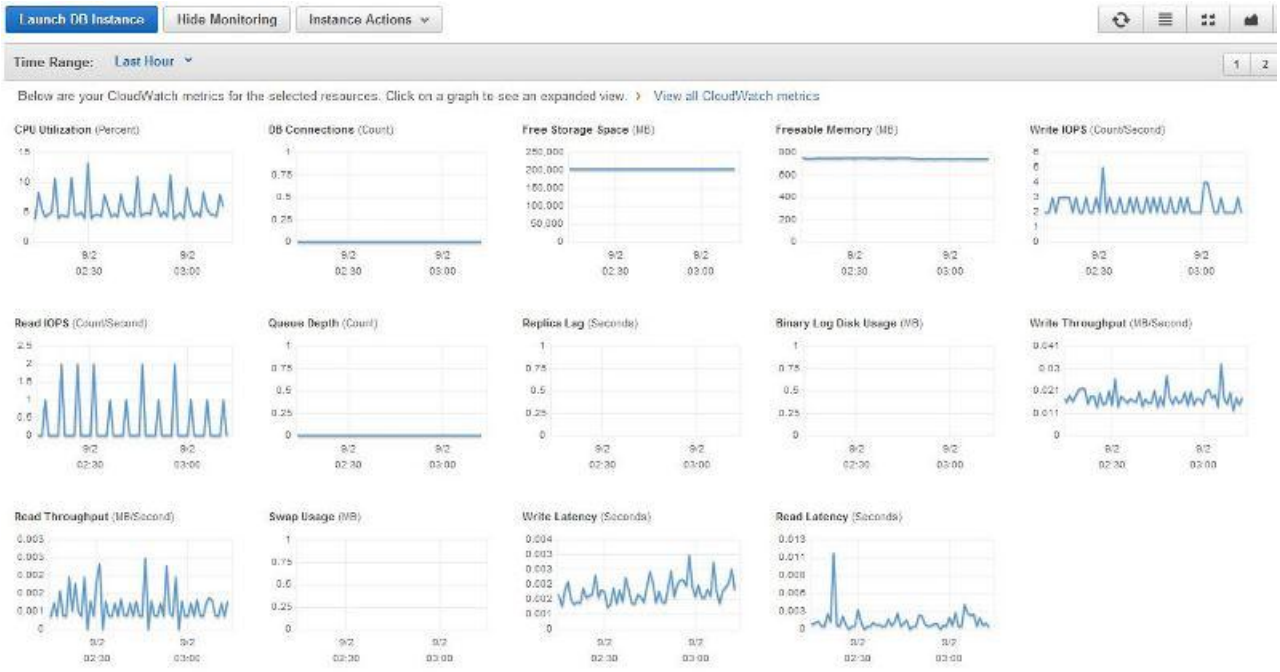
Specifying the DB details asks for the instance class and we can make the Instance Identifier can be identified by Master Username and Password.



Configuring the Advanced Settings helps to specify the network and security settings for the Database and launch the Instance.



Monitoring the usage of the Memory utilization of the RDS Instance.



You can launch the AWs instance and test the connectivity to on premise.

5. ELASTICACHE MANAGEMENT

Amazon ElastiCache is a web service that makes it easy to set up, manage and scale a distributed in-memory data store or cache in the cloud.

5.1 Objective

To provide the high level guidance's to setup the AWS Elasticache Management on AWS Cloud

5.2 Assumption

- AWS Account is available with AWS Elasticache.
- AWS EC2 required to create a security groups.

5.3 Procedure

To set up a Elasticache connection, you need to complete the following steps:

Step 1: Create an AWS Account [One time]

Step 2: Launch a Cluster

Step 3: (Optional) View Cluster Details

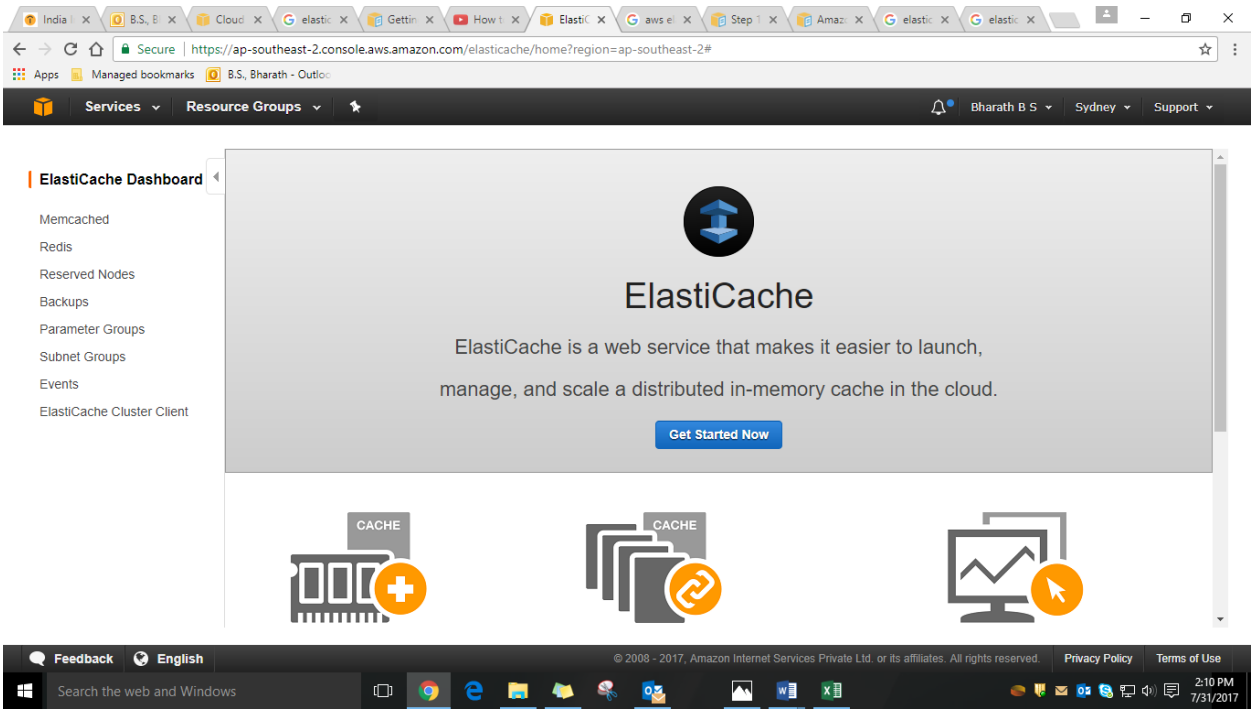
Step 4: Authorize Access

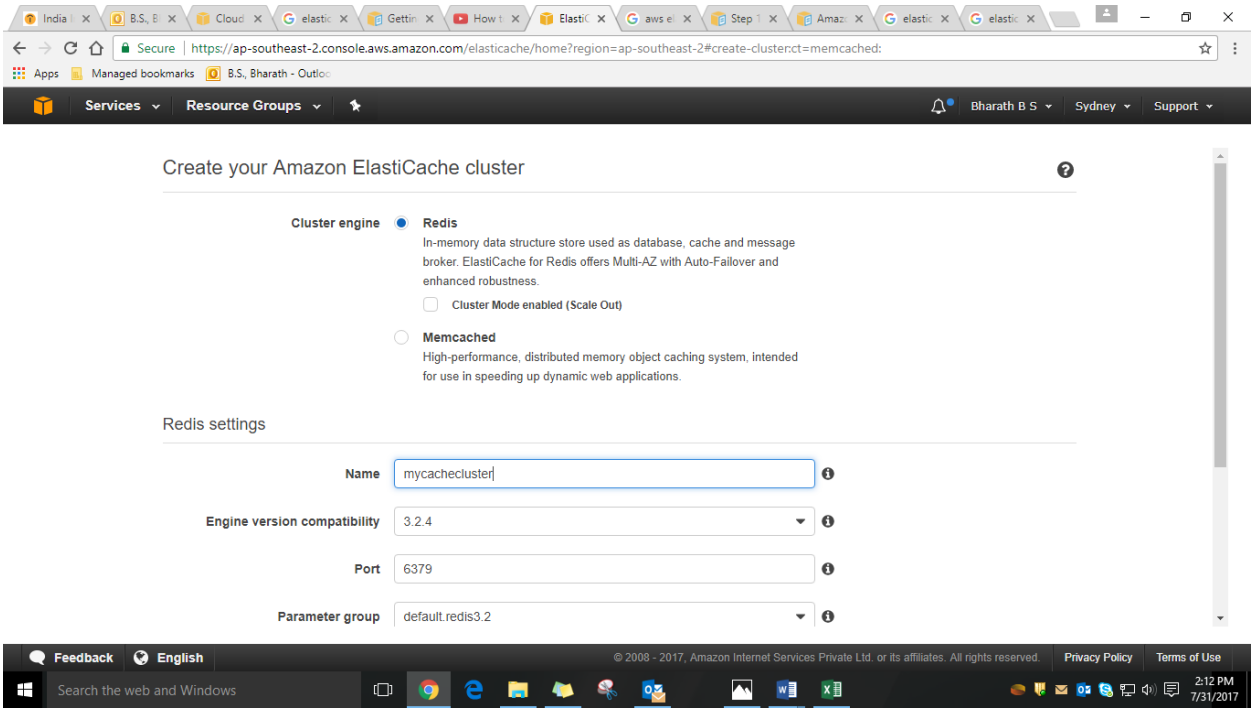
Step 5: Connect to a Cluster's Node

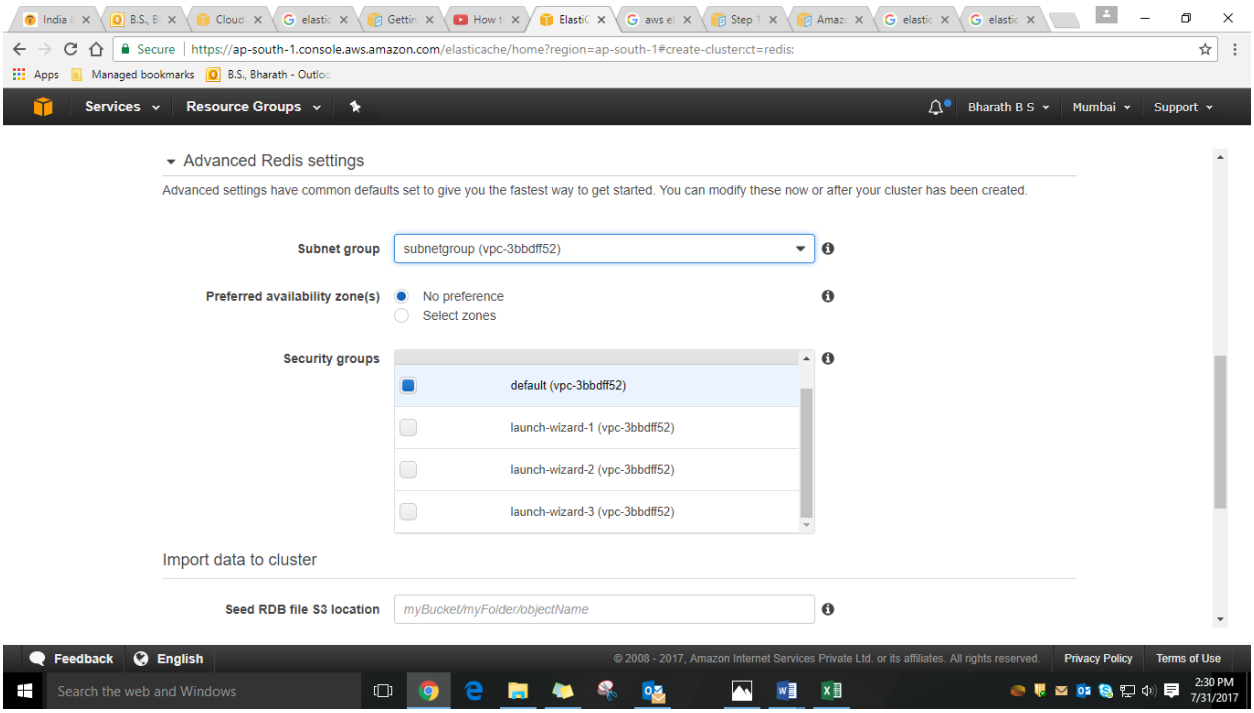
Step 6: Delete Your Cluster [Avoid Unnecessary Charges]

5.3.1 Launch a Cluster

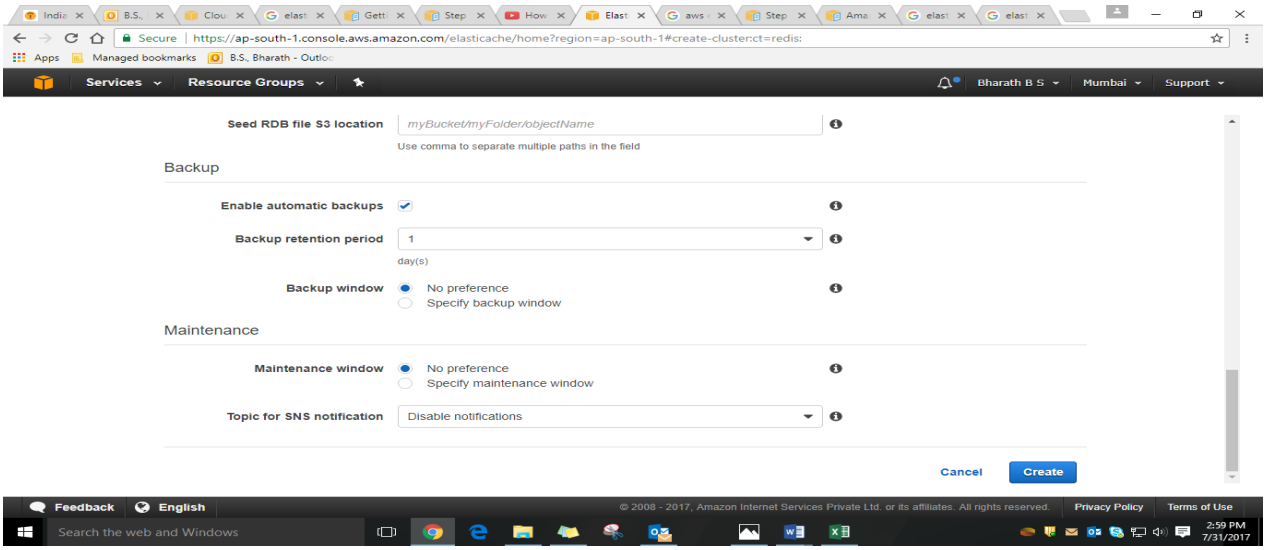
Go to the Elasticache and click on the Get Started Now option to create the Create Elasticcache


Secure | https://ap-southeast-2.console.aws.amazon.com/elasticache/home?region=ap-southeast-2#
Services
Resource Groups
Bharath B S
Sydney
Support
ElastiCache Dashboard
Memcached
Redis
Reserved Nodes
Backups
Parameter Groups
Subnet Groups
Events
ElastiCache Cluster Client
ElastiCache
ElastiCache is a web service that makes it easier to launch,
manage, and scale a distributed in-memory cache in the cloud.
Get Started Now
CACHE
CACHE
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Search the web and Windows
2:10 PM
7/31/2017

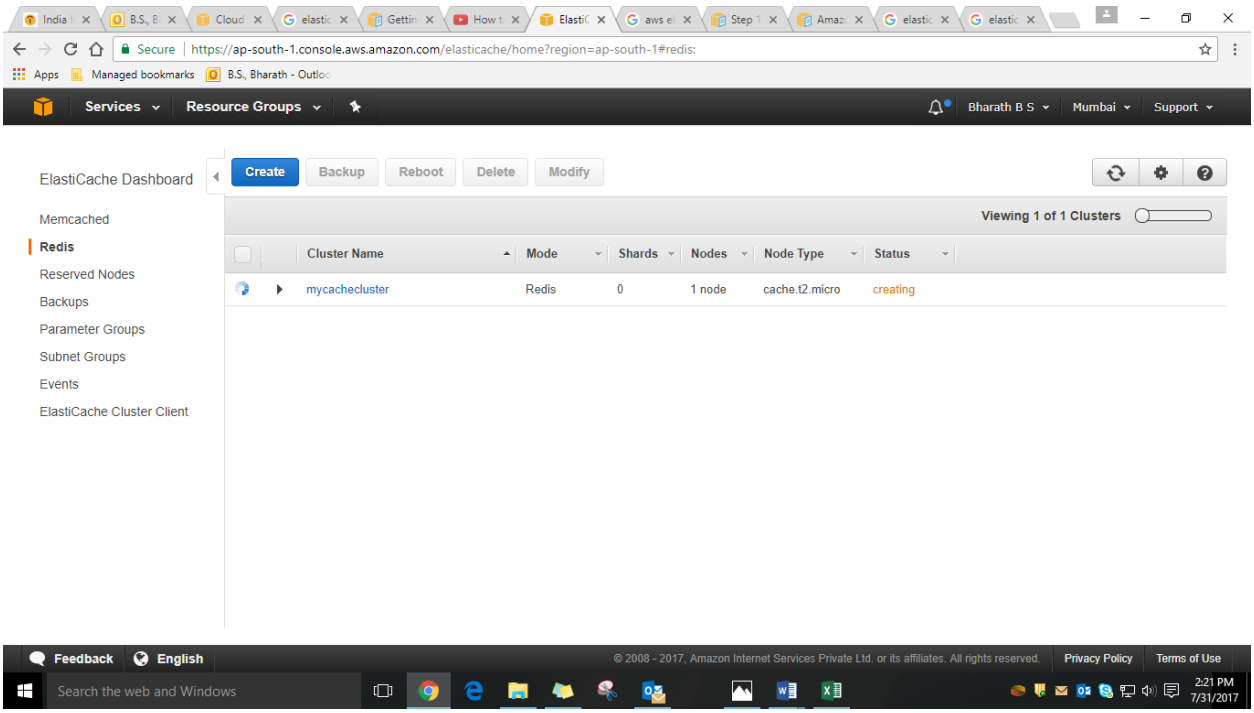

Secure | https://ap-southeast-2.console.aws.amazon.com/elasticache/home?region=ap-southeast-2#create-cluster:ct=memcached:
Services
Resource Groups
Bharath B S
Sydney
Support
Create your Amazon ElastiCache cluster
Cluster engine
Redis
In-memory data structure store used as database, cache and message broker. ElastiCache for Redis offers Multi-AZ with Auto-Failover and enhanced robustness.
Cluster Mode enabled (Scale Out)
Memcached
High-performance, distributed memory object caching system, intended for use in speeding up dynamic web applications.
Redis settings
Name
mycachecluster
Engine version compatibility
3.2.4
Port
6379
Parameter group
default.redis3.2
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Search the web and Windows
2:12 PM
7/31/2017

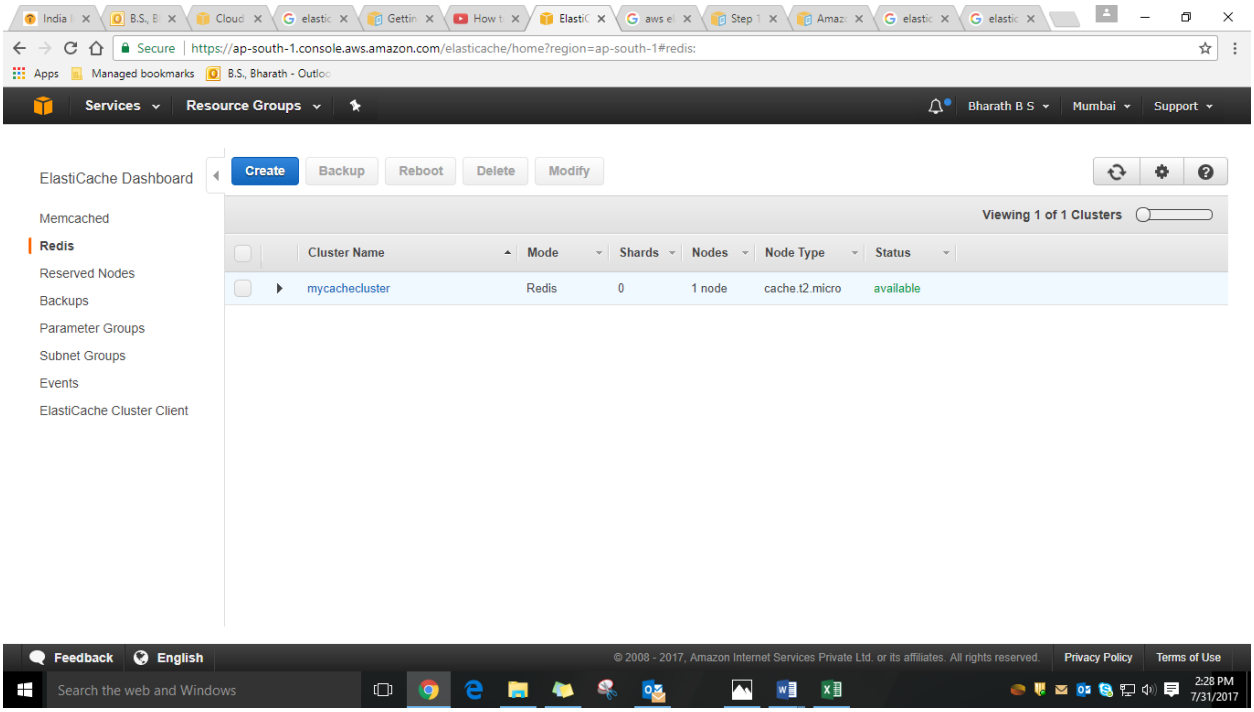


5.3.2 View Cluster Details

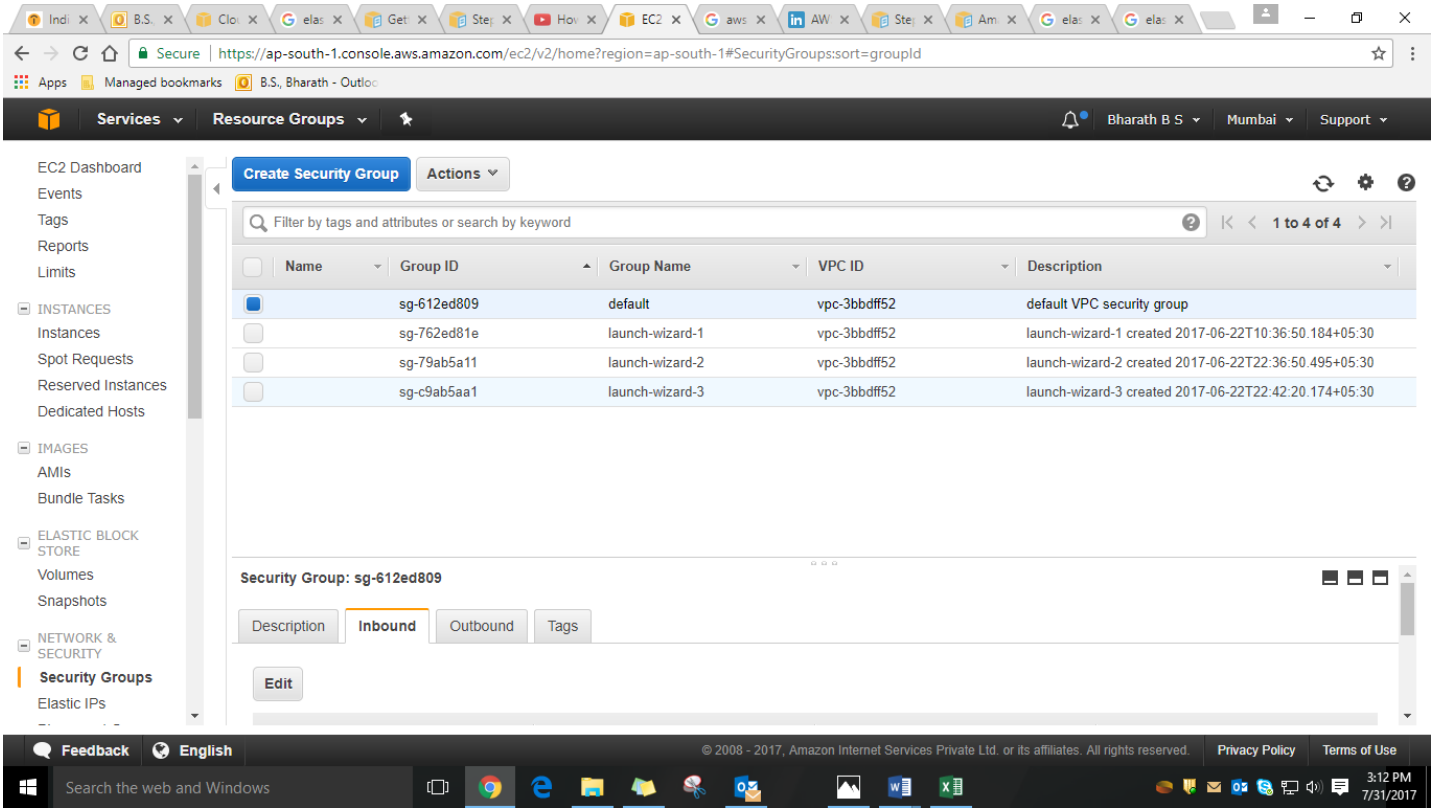


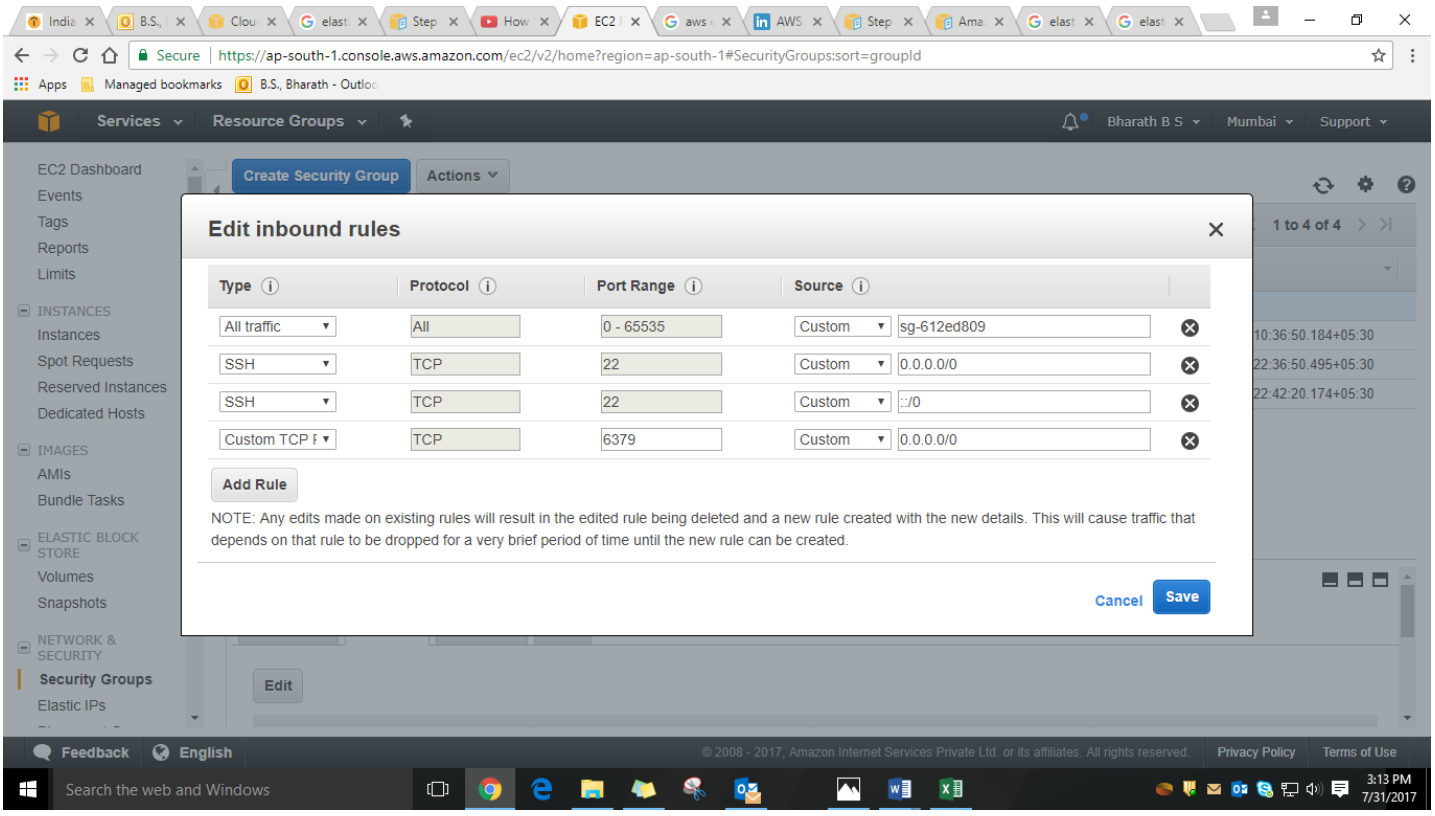
Wait till it is available


Secure | https://ap-south-1.console.aws.amazon.com/elasticache/home?region=ap-south-1#redis:
Services
Resource Groups
Bharath B S
Mumbai
Support
ElastiCache Dashboard
Create
Backup
Reboot
Delete
Modify
Memcached
Redis
Reserved Nodes
Backups
Parameter Groups
Subnet Groups
Events
ElastiCache Cluster Client
Viewing 1 of 1 Clusters
Cluster Name
Mode
Shards
Nodes
Node Type
Status
mycachecluster
Redis
0
1 node
cache.t2.micro
creating
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Search the web and Windows
2:21 PM
7/31/2017


Secure | https://ap-south-1.console.aws.amazon.com/elasticache/home?region=ap-south-1#redis:
Services
Resource Groups
Bharath B S
Mumbai
Support
ElastiCache Dashboard
Create
Backup
Reboot
Delete
Modify
Memcached
Redis
Reserved Nodes
Backups
Parameter Groups
Subnet Groups
Events
ElastiCache Cluster Client
Viewing 1 of 1 Clusters
Cluster Name
Mode
Shards
Nodes
Node Type
Status
mycachecluster
Redis
0
1 node
cache.t2.micro
available
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Search the web and Windows
2:28 PM
7/31/2017

5.3.3 Authorize Access





5.3.4 Connect to a Cluster's Node

Download and install the *telnet* utility on your Amazon EC2 instance. At the command prompt of your Amazon EC2 instance, type the following command and type *y* at the command prompt.

```
sudo yum install telnet

Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                          |  63 kB

...(output omitted)...

Complete!
```

At the command prompt of your Amazon EC2 instance, type the following command, substituting the endpoint of your node for the one shown in this example.

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

Output similar to the following appears.

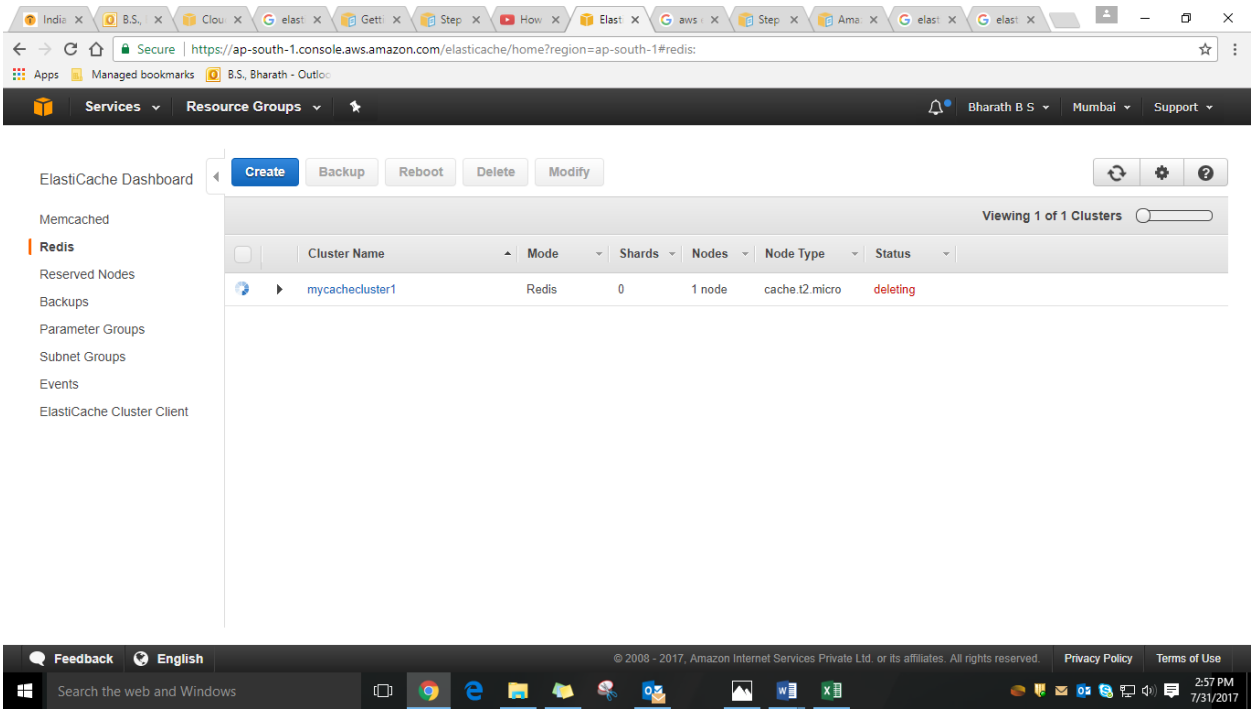
```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^]'.
>
```

Run Redis commands.

You are now connected to a node, and you can run Redis commands. The following is an example.

```
set a 0 0 5       // Set key "a" with no expiration and 5 byte value
hello             // Set value as "hello"
STORED
get a             // Get value for key "a"
VALUE a 0 5
hello
END
get b             // Get value for key "b" results in miss
END
>
```

5.3.5 Delete Your Cluster [Avoid Unnecessary Charges]



You can launch the AWs instance and test the connectivity to on premise.

6. SES MANAGEMENT

Amazon SES lets you send marketing and transactional email to customers in a quick and cost-effective manner

6.1 Objective

To provide the high level guidance's to setup the AWS SES (Simple Email Service) on AWS Cloud.

6.2 Assumption

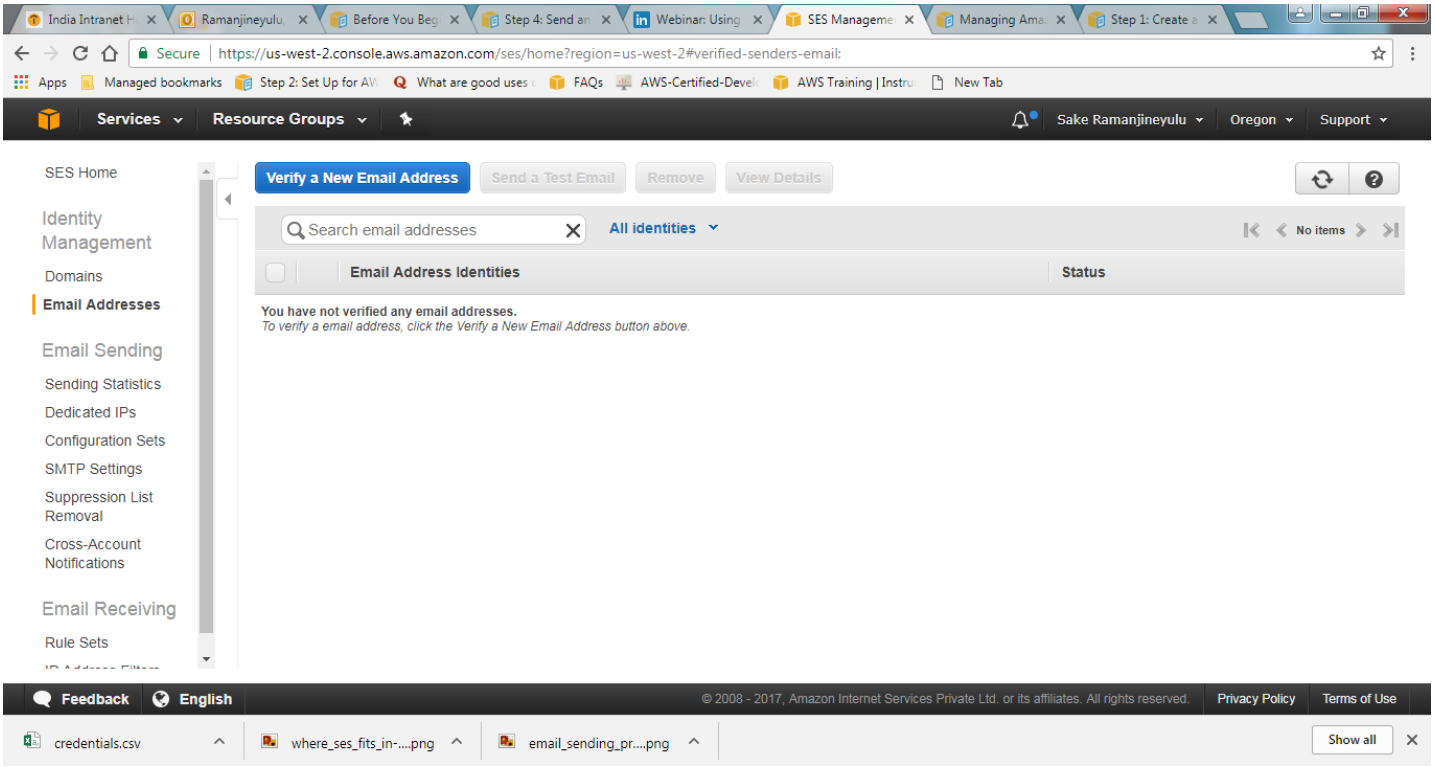
- Minimal setup, maximum deliverability send a lot of email as often necessary. All aspects of email management built in Focus on your business.
- You can send email in minutes and stay informed with on-demand feedback.
- This service is Available in three regions only. Those are EU (Ireland) , US East (N. Virginia) and US West (Oregon)

6.3 Procedure

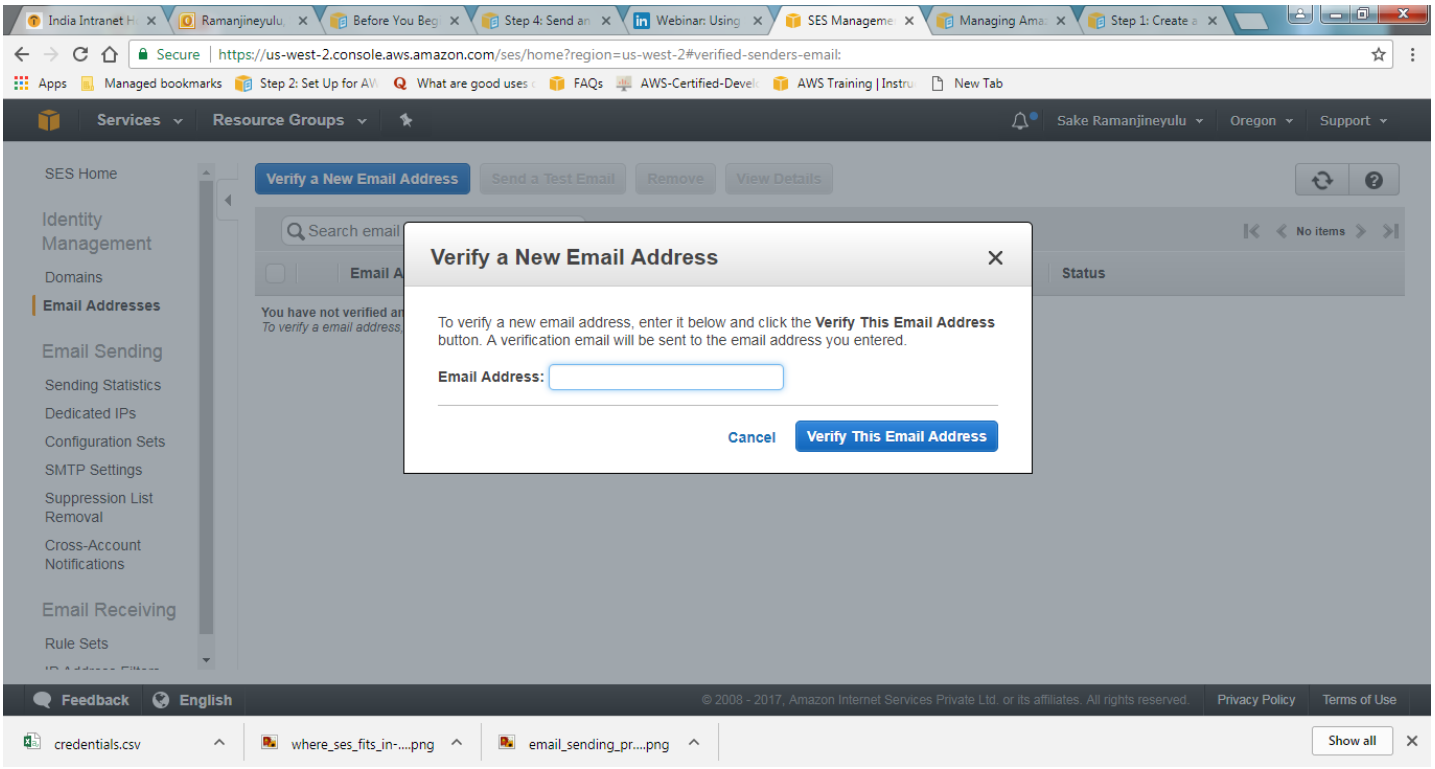
To set up a SES, you need to complete the following steps:

Go to the AWS console and click on SES and follow the screen shots to complete the overall SES set up.

6.3.1 Verifying Email Address

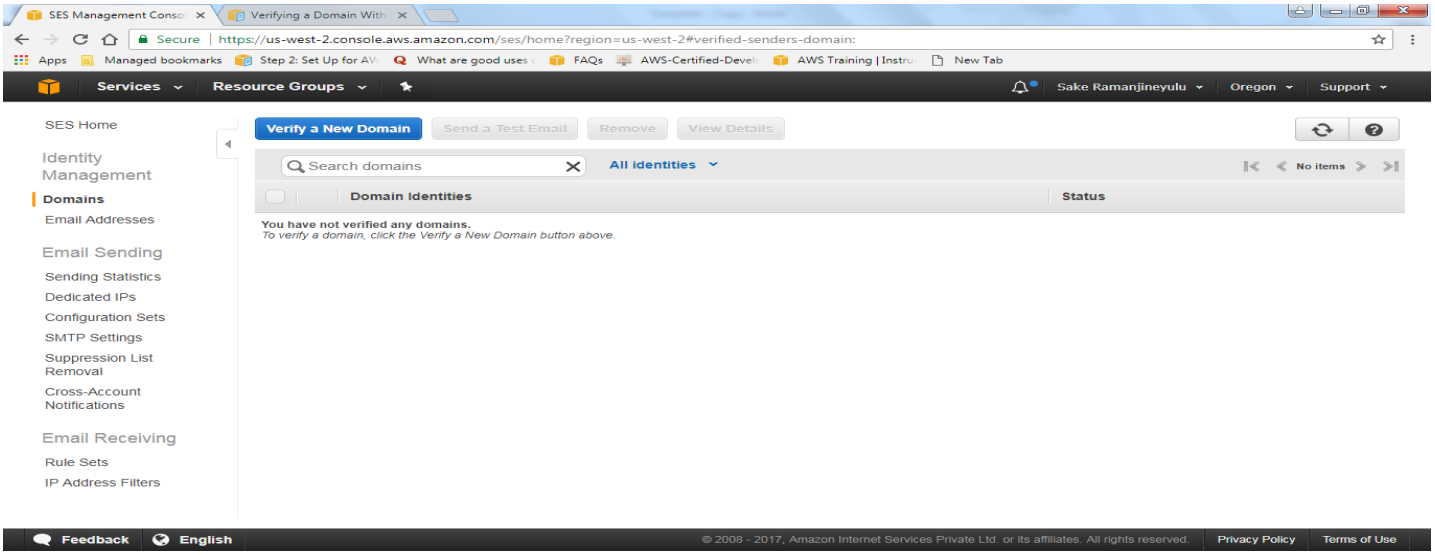


Click on Verify new Email Address.

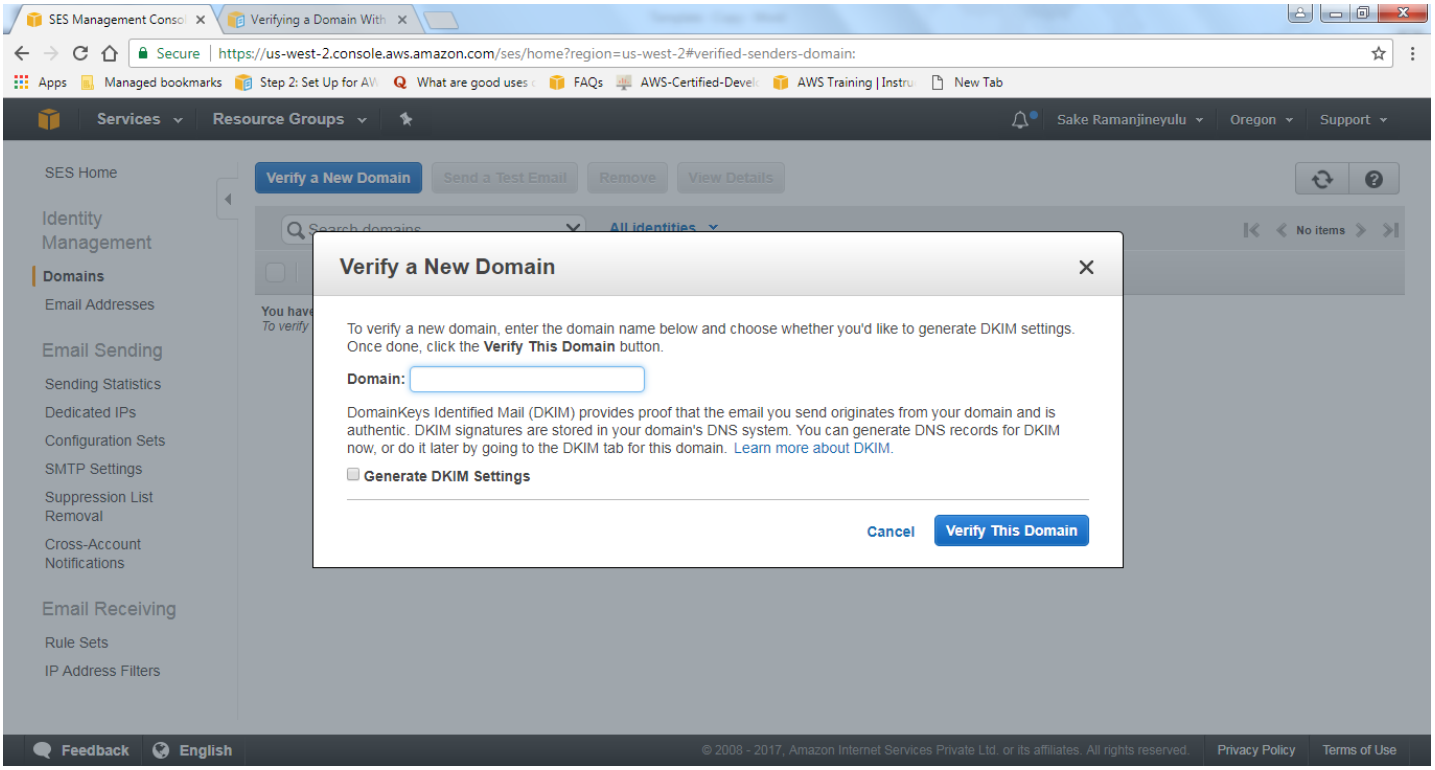


Enter your Email Address and click on Verify this Email Address. After this Your Email address receives a Confirmation mail link and click on that on that link Verify your email address.

6.3.2 Verifying Domain Address



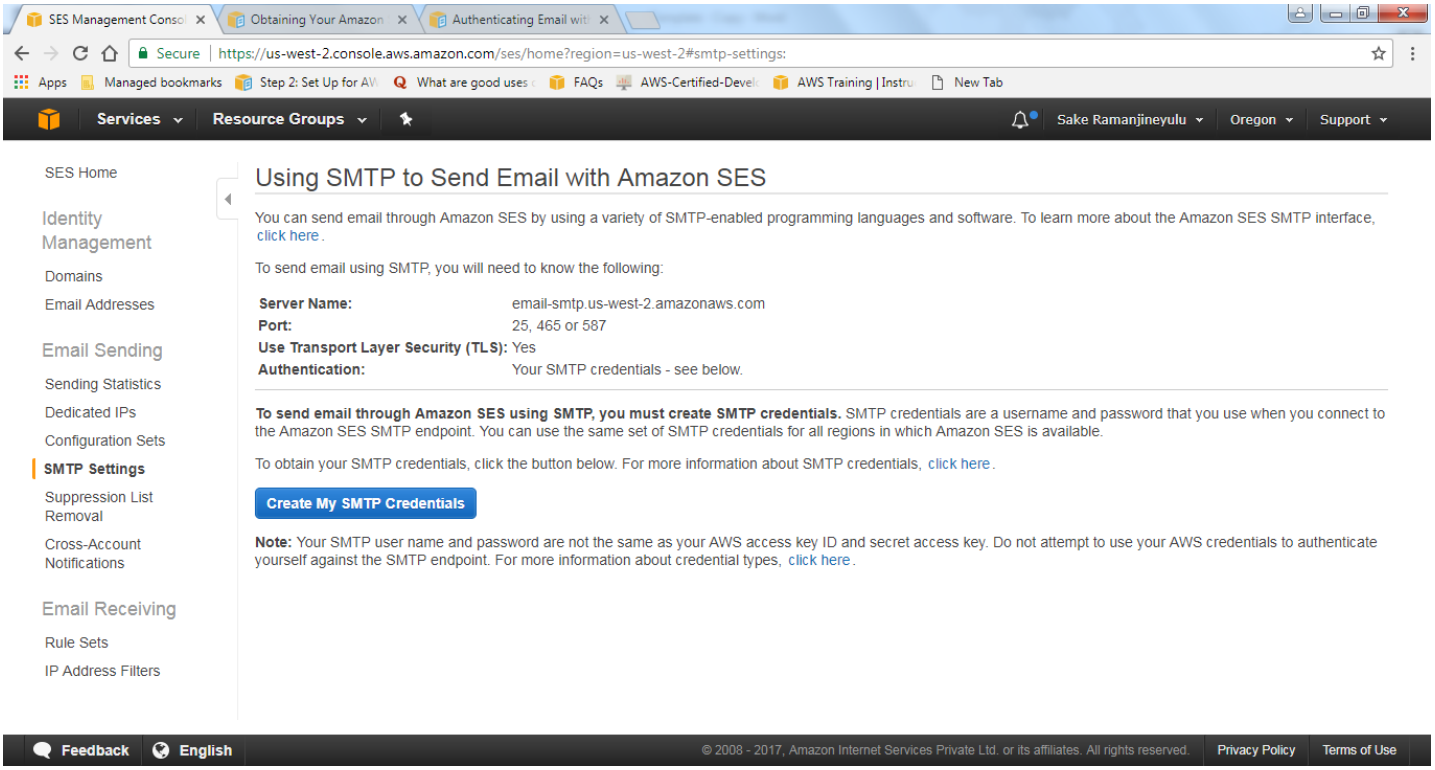
Click on Verify a new domain



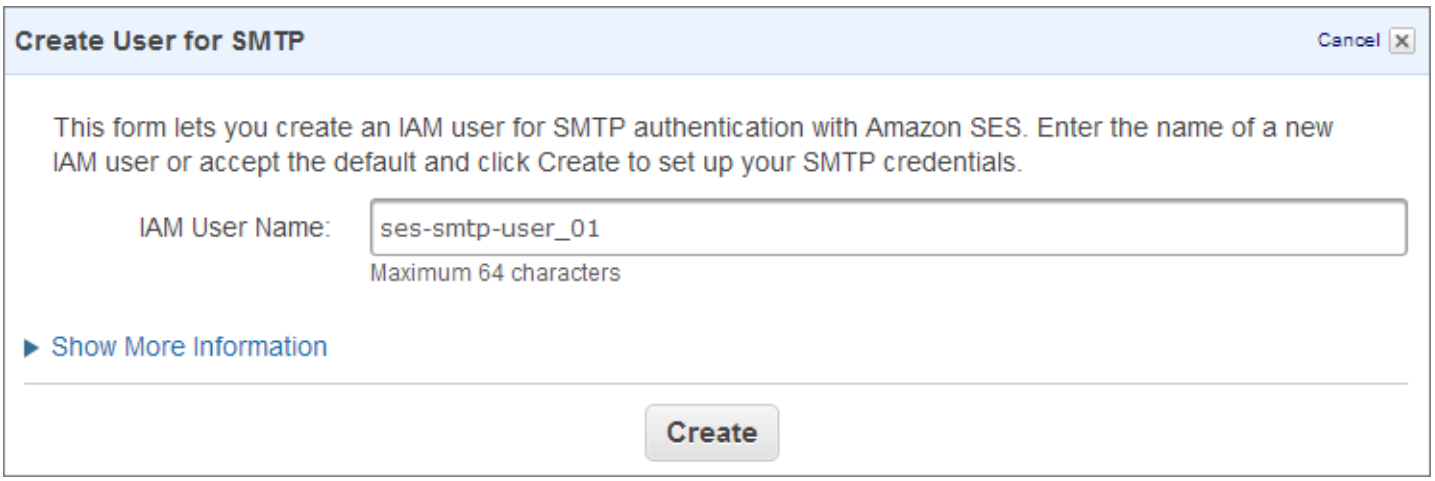
Enter Your Domain name and if you want Generate the DKIM settings then click on check box and after clicked on Verify this Domain You have to fill the Name, Type and Value for your domain.

6.3.3 Using SMTP interface to Send Email

In the navigation pane click on SMTP settings and choose create My SMTP Credentials



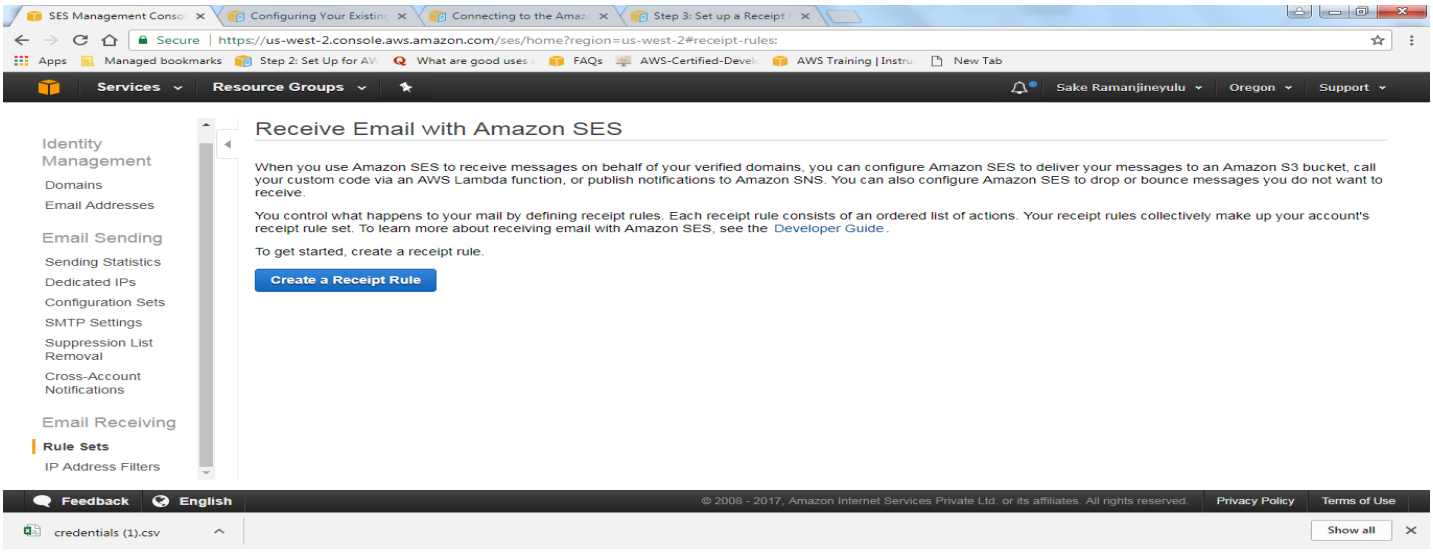
After Click on create My SMTP Credentials it will take you into IAM user



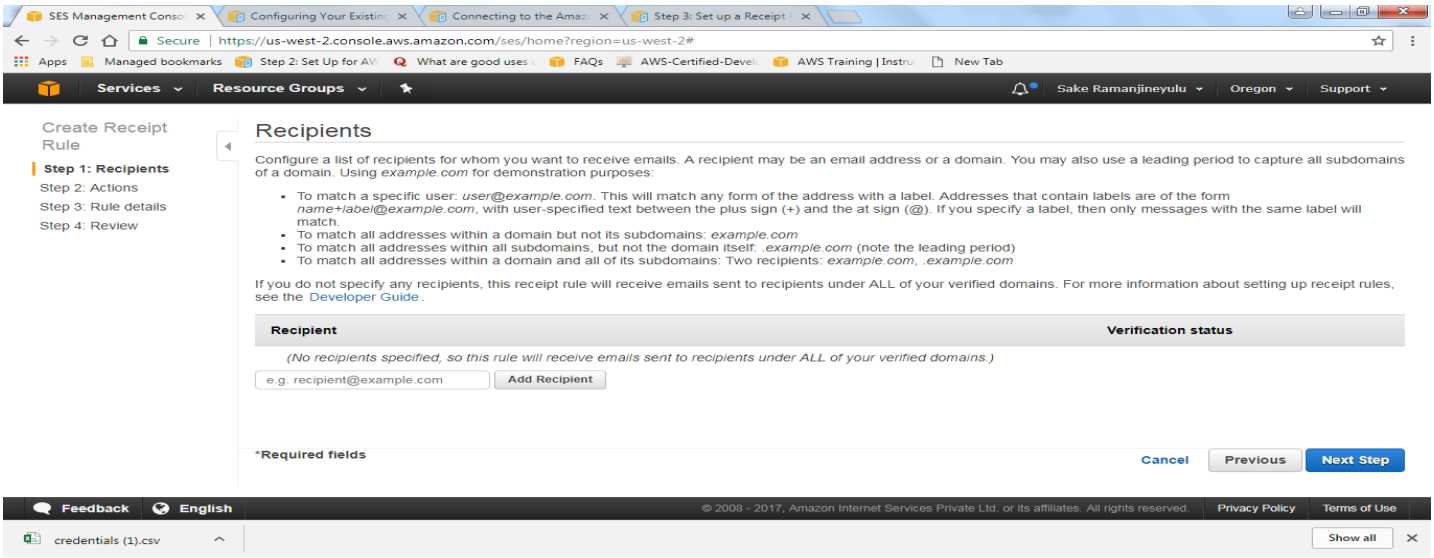
Click on Create and Download Your SMTP credentials

6.3.4 Receiving Email

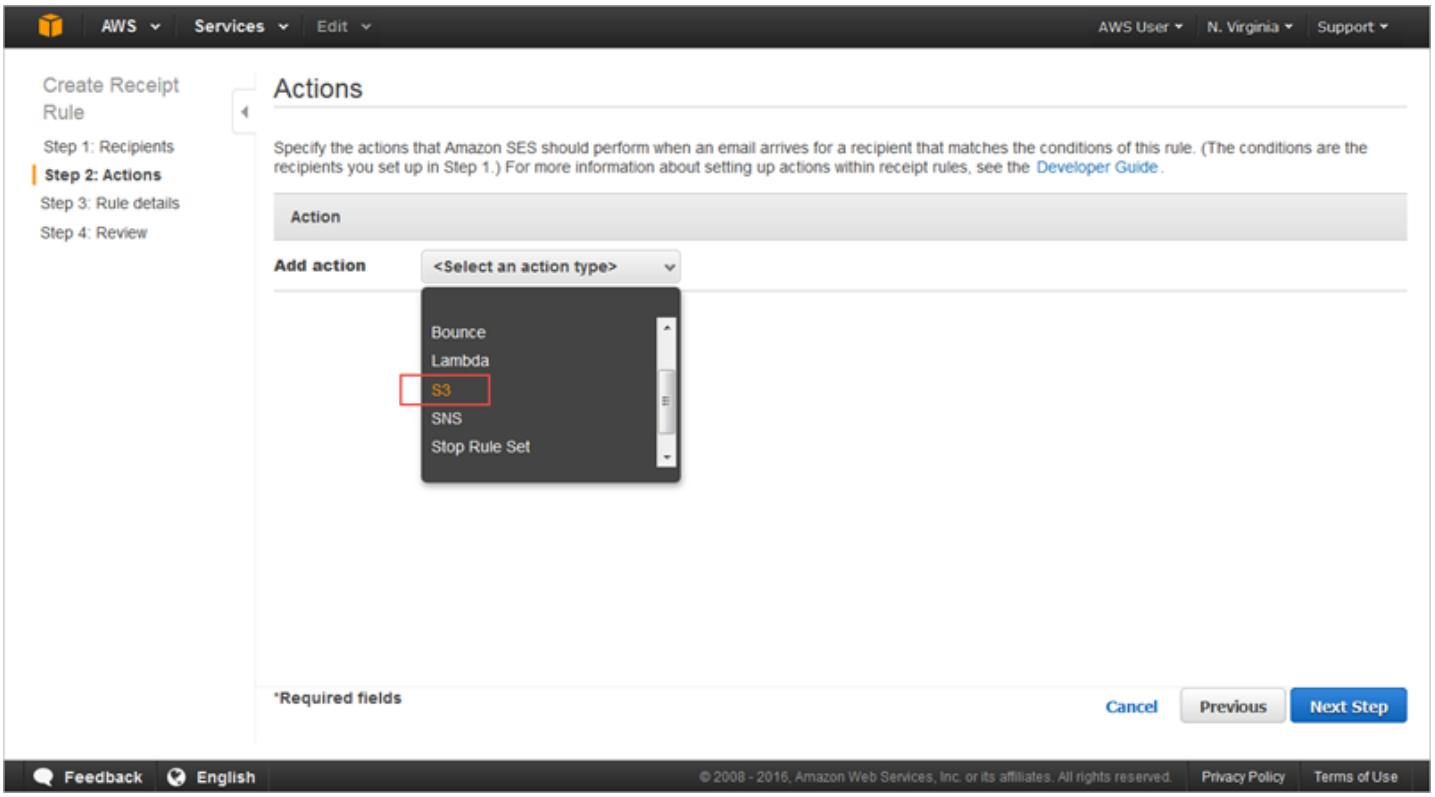
For this you have to create one **receipt rule**

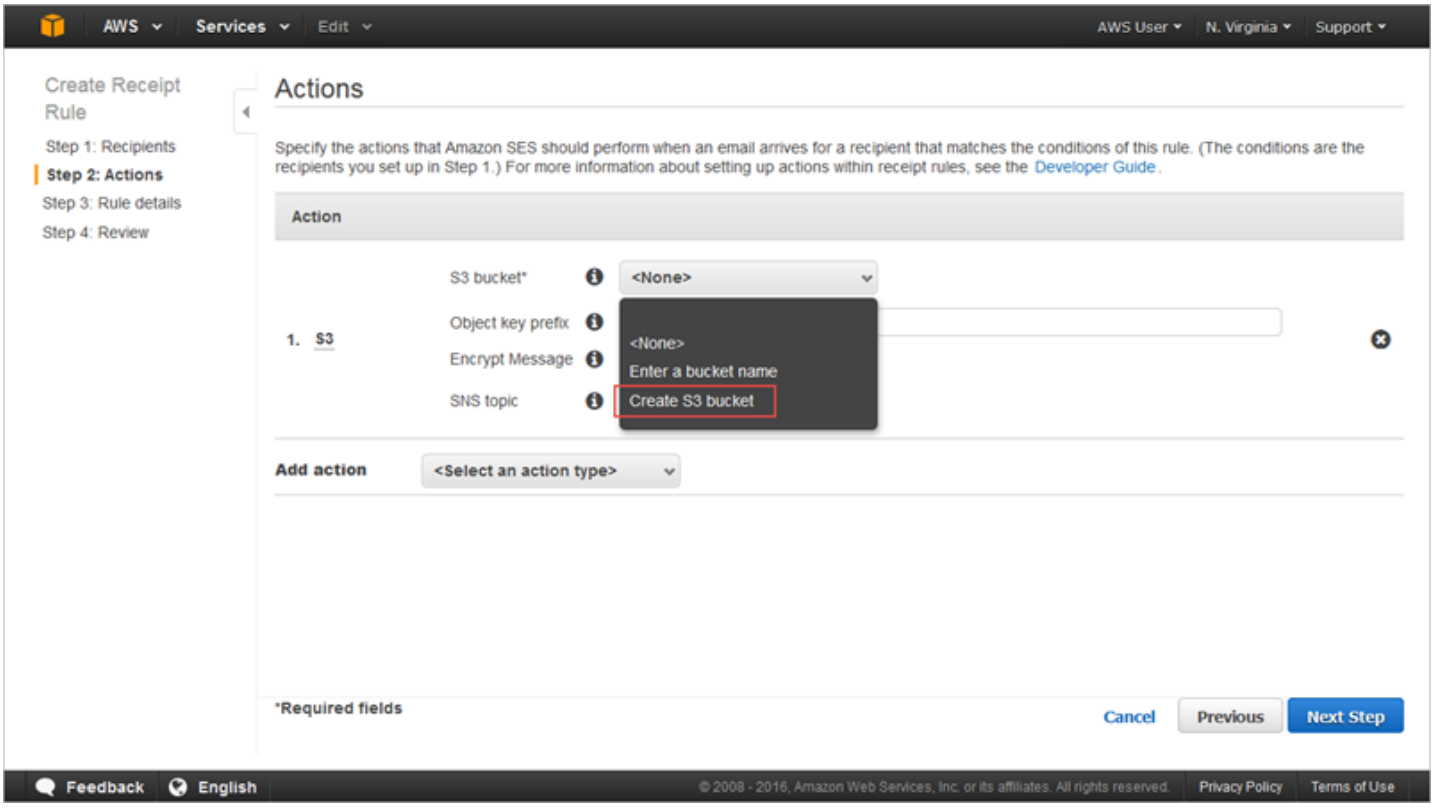


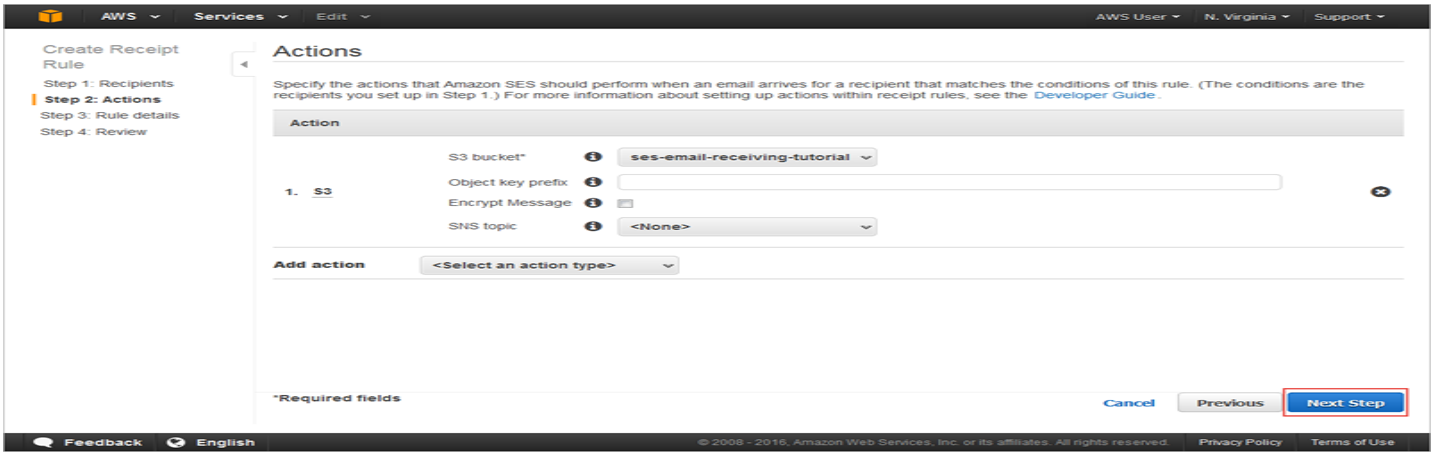
After click on Create a Receipt rule it will display window like this



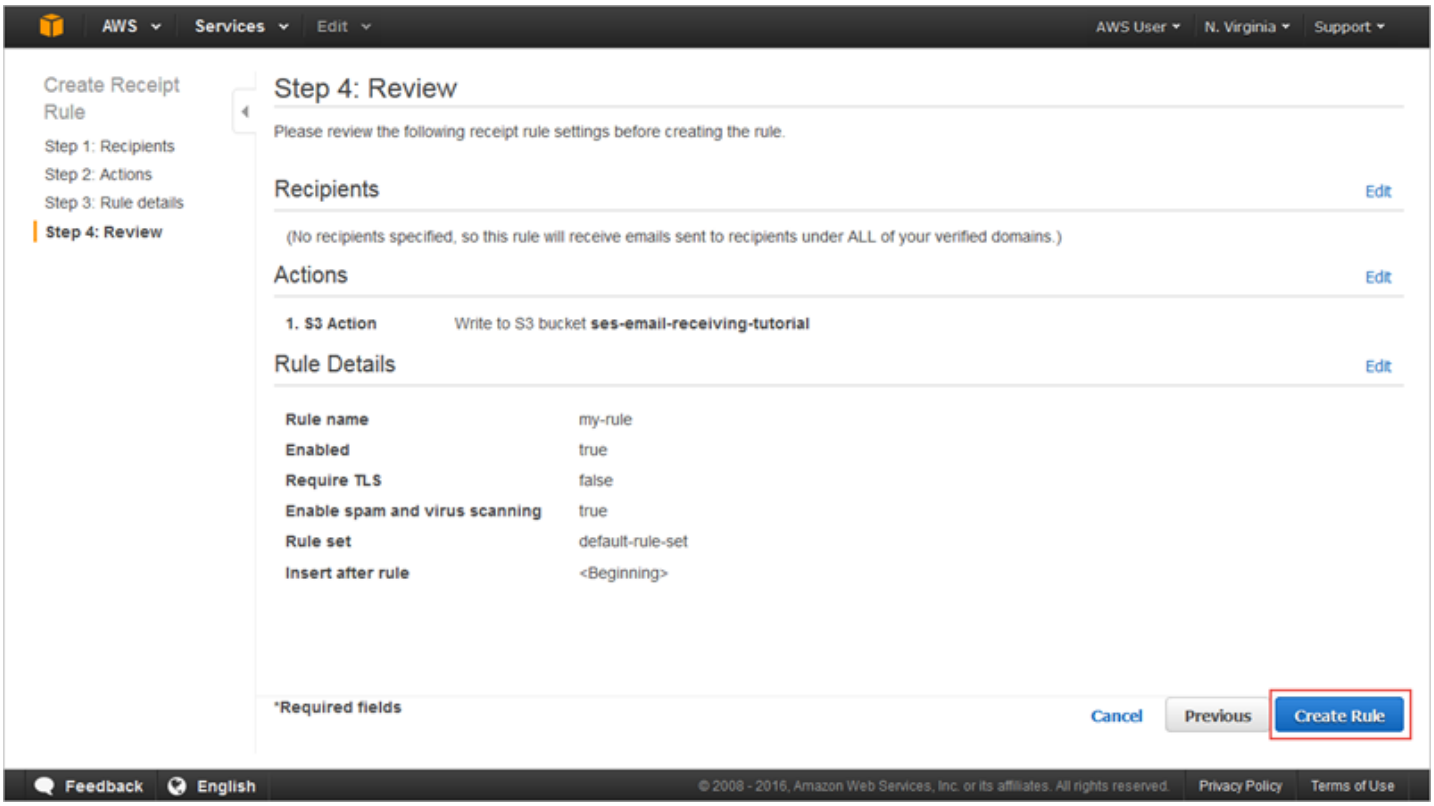
In this above Window you have to add your Recipient which you have previously created domain.


AWS
Services
Edit
AWS User
N. Virginia
Support
Create Receipt Rule
Step 1: Recipients
Step 2: Actions
Step 3: Rule details
Step 4: Review
Actions
Specify the actions that Amazon SES should perform when an email arrives for a recipient that matches the conditions of this rule. (The conditions are the recipients you set up in Step 1.) For more information about setting up actions within receipt rules, see the Developer Guide.
Action
Add action
<Select an action type>
Bounce
Lambda
S3
SNS
Stop Rule Set
*Required fields
Cancel
Previous
Next Step
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


AWS
Services
Edit
AWS User
N. Virginia
Support
Create Receipt Rule
Step 1: Recipients
Step 2: Actions
Step 3: Rule details
Step 4: Review
Actions
Specify the actions that Amazon SES should perform when an email arrives for a recipient that matches the conditions of this rule. (The conditions are the recipients you set up in Step 1.) For more information about setting up actions within receipt rules, see the Developer Guide.
Action
1. S3
S3 bucket*
Object key prefix
Encrypt Message
SNS topic
<None>
<None>
Enter a bucket name
Create S3 bucket
Add action
<Select an action type>
*Required fields
Cancel
Previous
Next Step
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



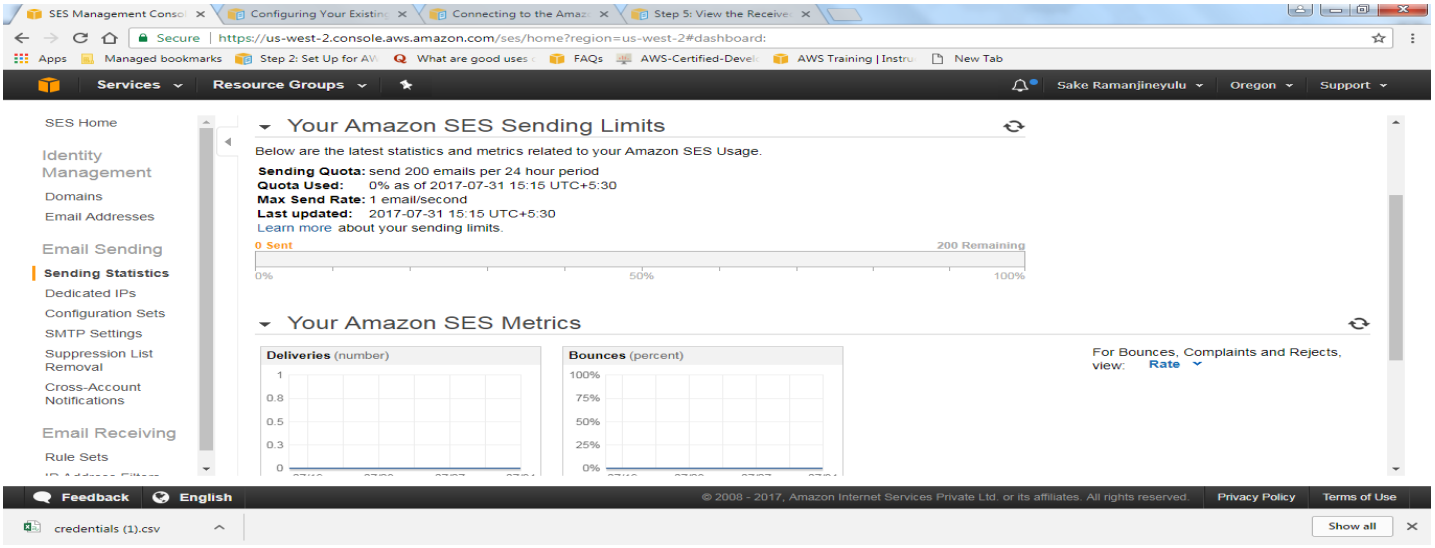
Click on Next step

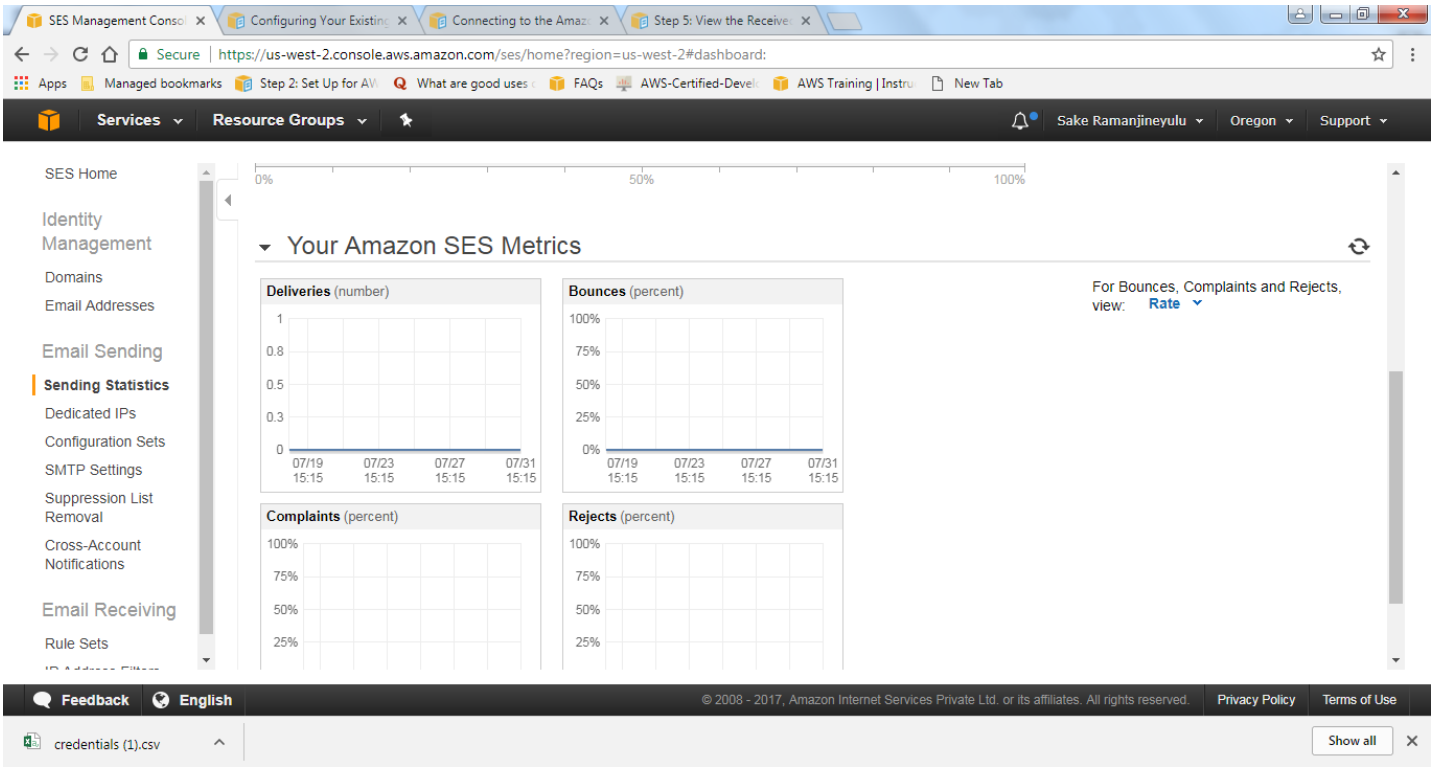


This above Window is the Overview of Rule check it once and Create rule. Send a mail from your Domain and check it in S3 Bucket which you have recently created.

6.3.5 Monitoring Your Sending Activity

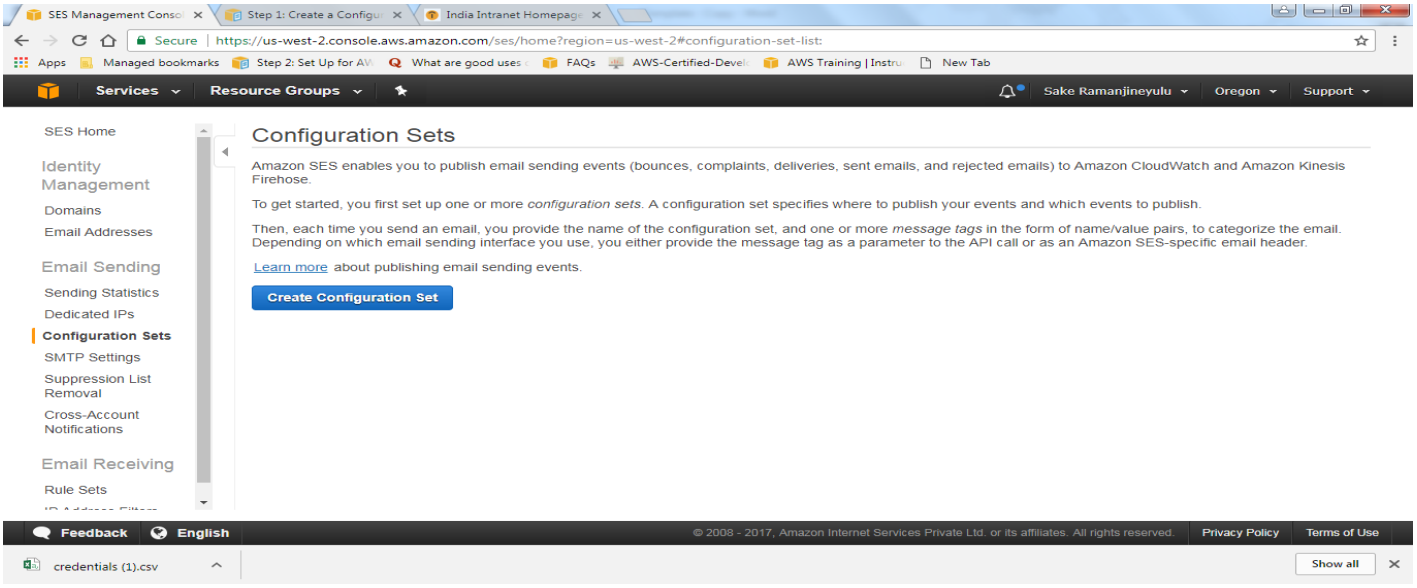
By using AWS console: In the navigation pane click on Sending Statistics it will show like this

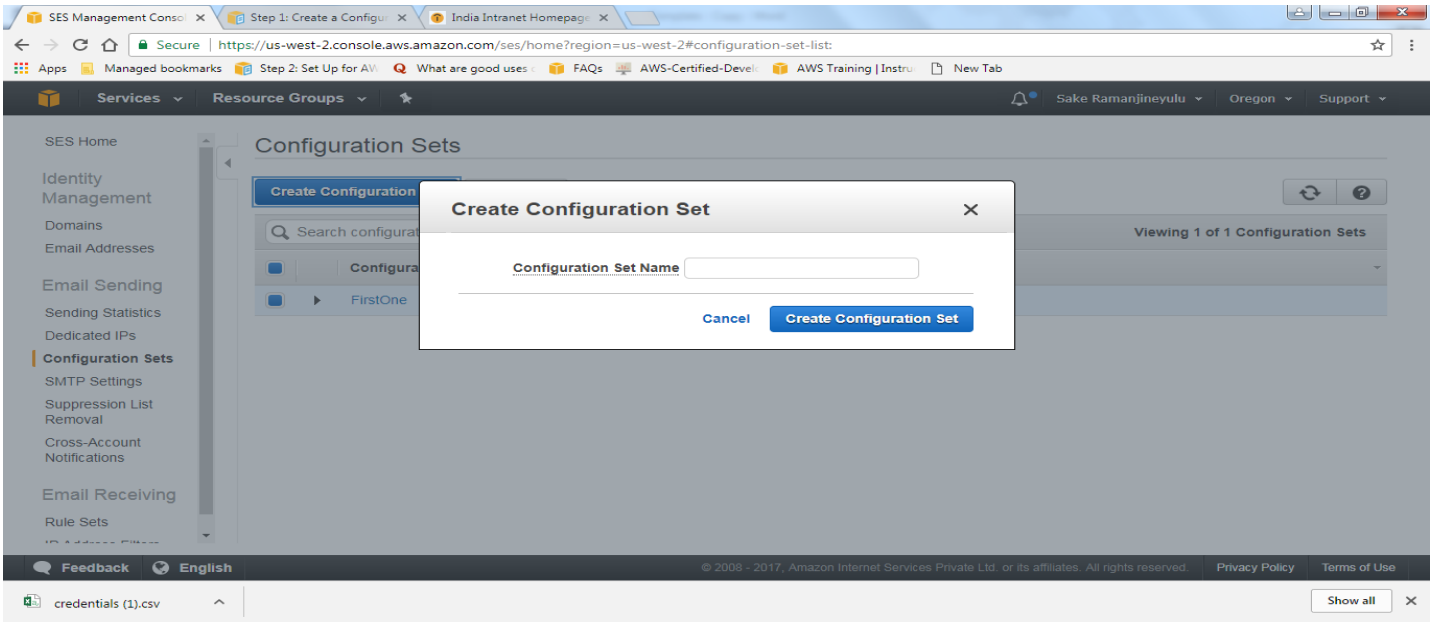




By using Event Publishing:

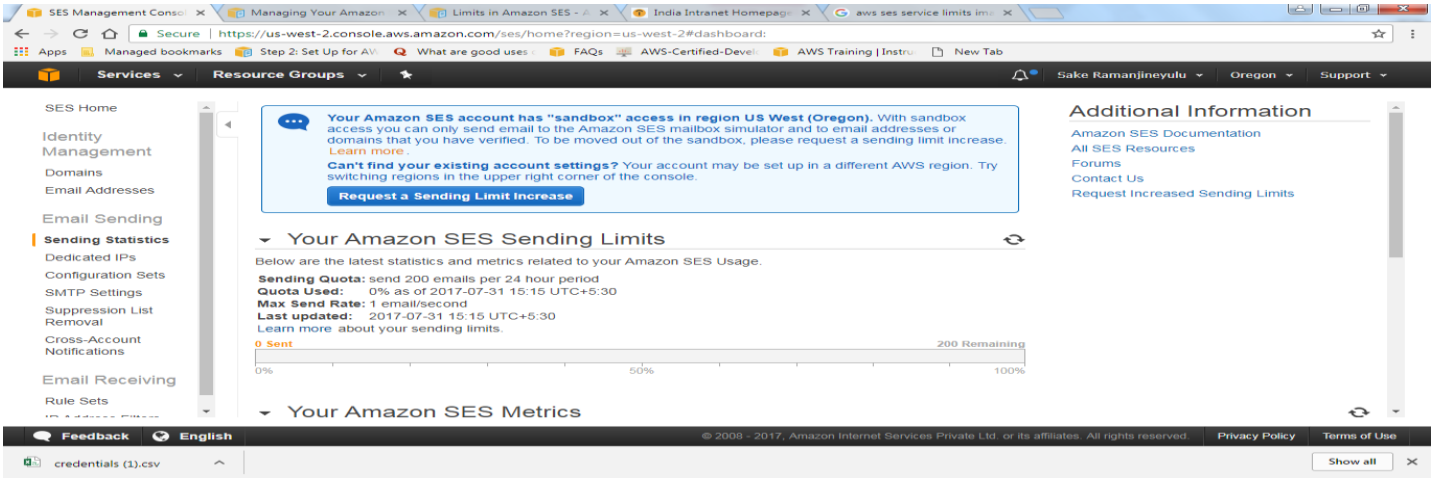
For this Purpose you have to create on configuration set. Configuration set enable you to publish email sending events to Amazon Cloud Watch. In the navigation pane click on Configuration sets





Give your configuration set name and click on create configuration set.

6.3.6 AWS SES Limits



7. SQS MANAGEMENT

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications.

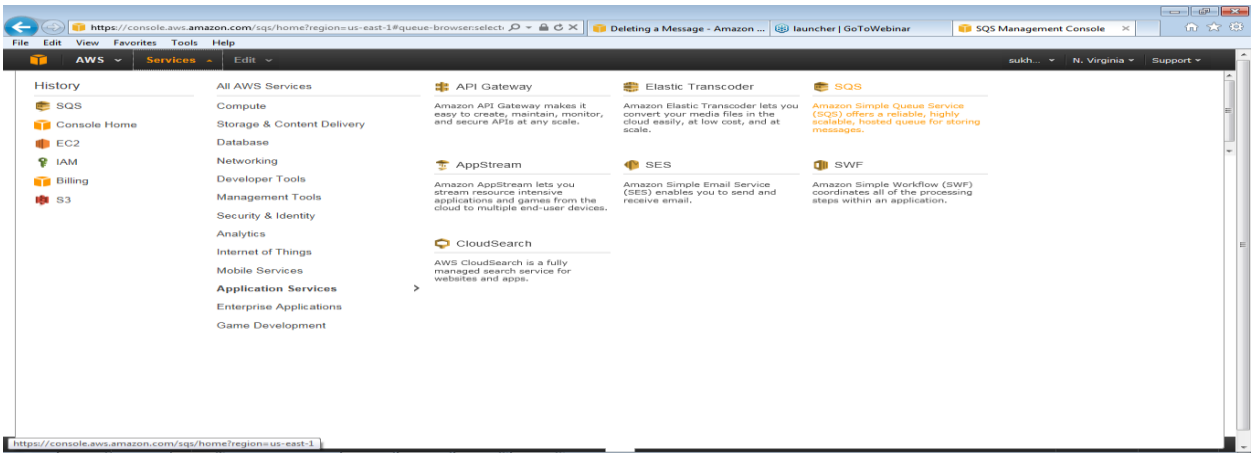
7.1 Objective

To understand the process of creating, modifying and managing SQS queues in AWS.

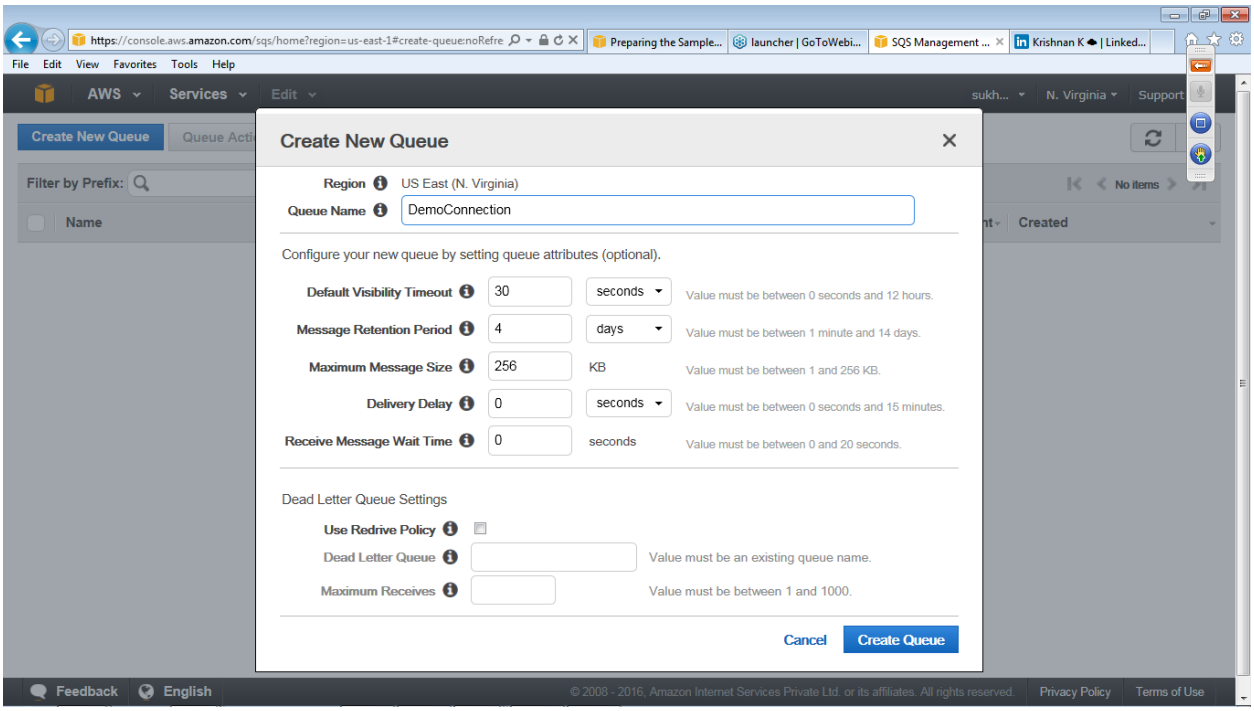
7.2 Procedure

7.2.1 Create New Queue

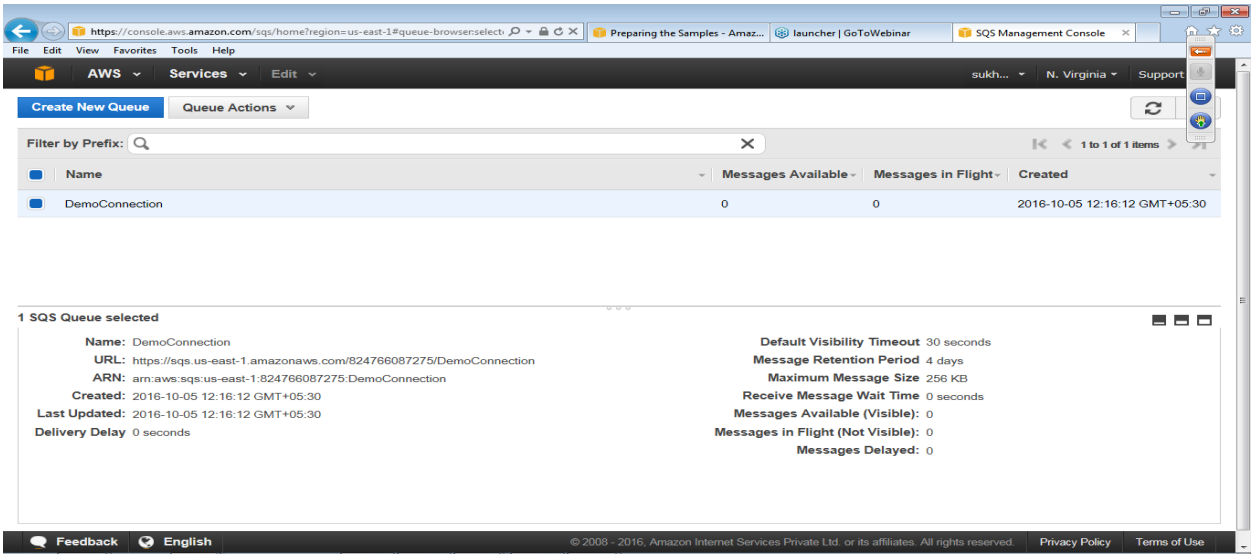
1 Sign in to AWS Management Console → Application Servics-> SQS



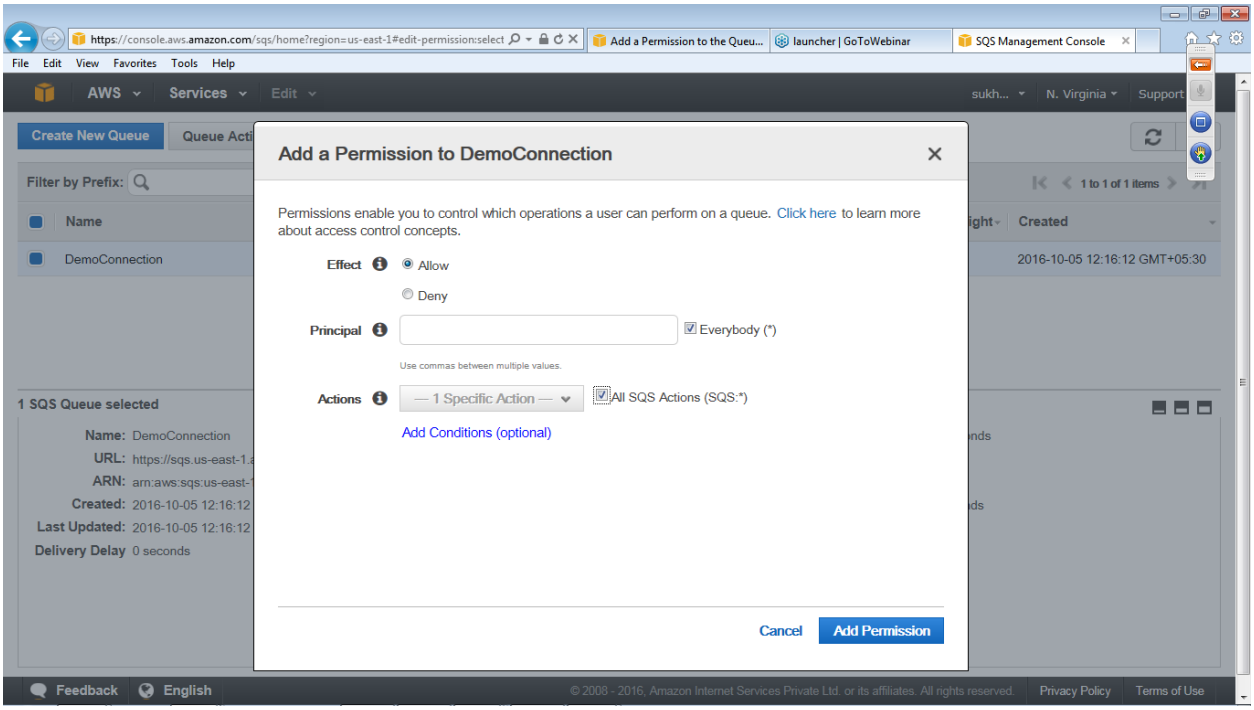
2 In the **Create New Queue** dialog box, enter DemoConnection in the Queue Name field, and leave the default value settings for the remaining fields



The console displays all of your queues in that region

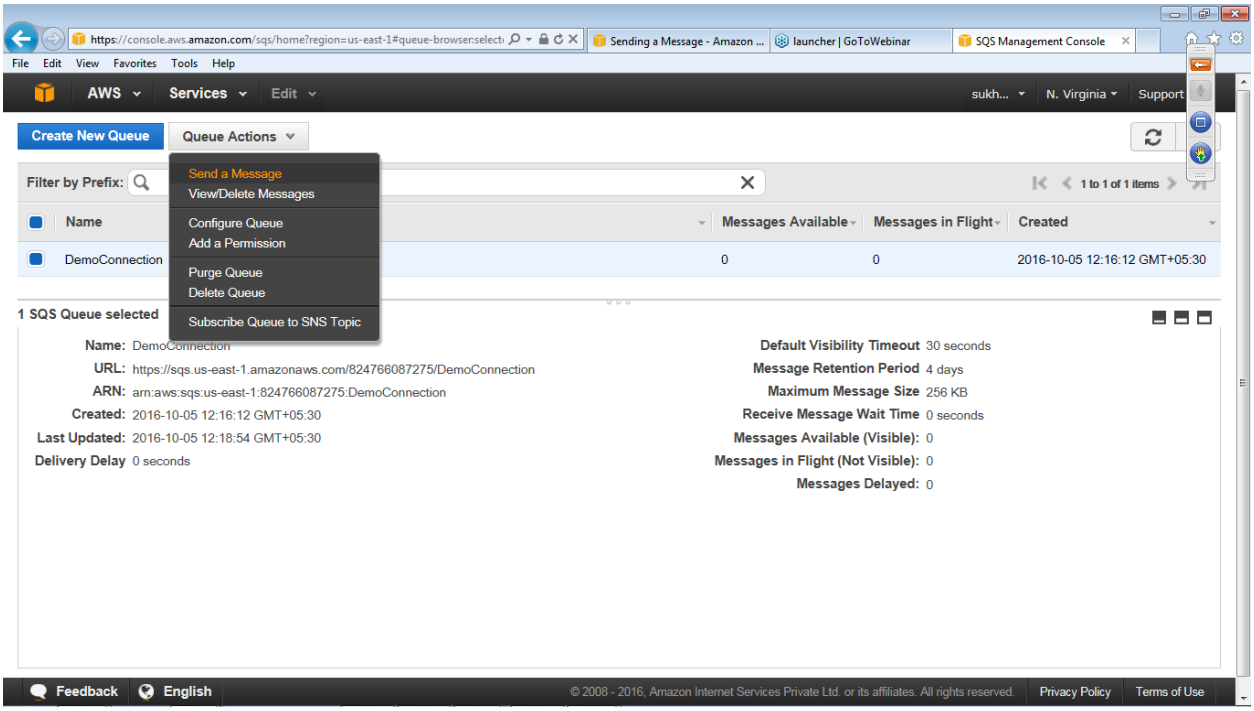


3 Select **Add a Permission** from the **Queue Actions** drop-down list and specify the permission's settings.

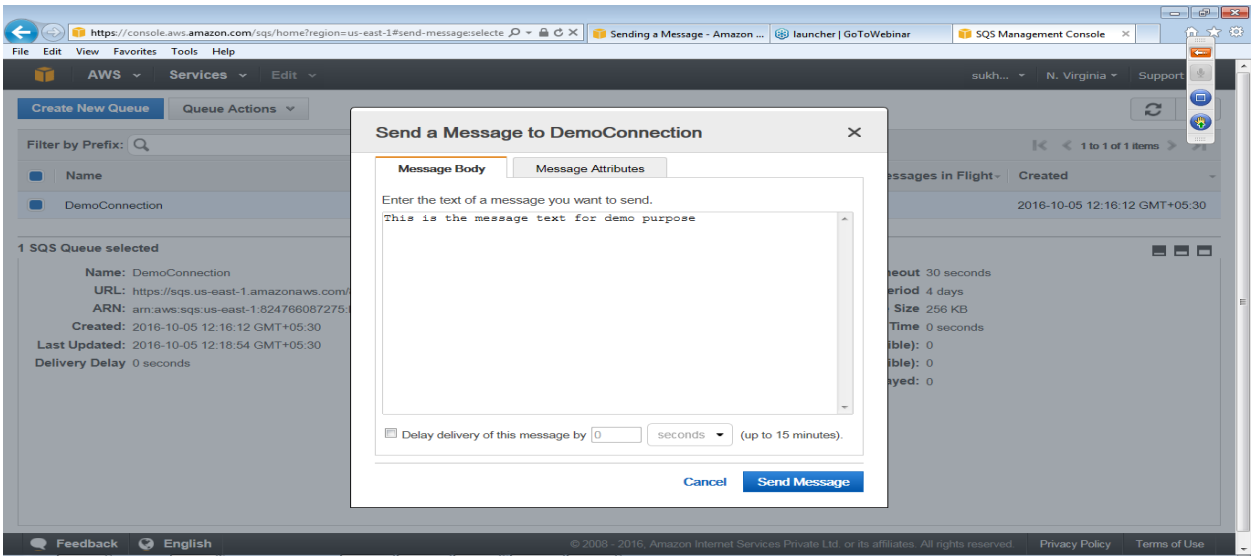


7.2.2 Sending Message

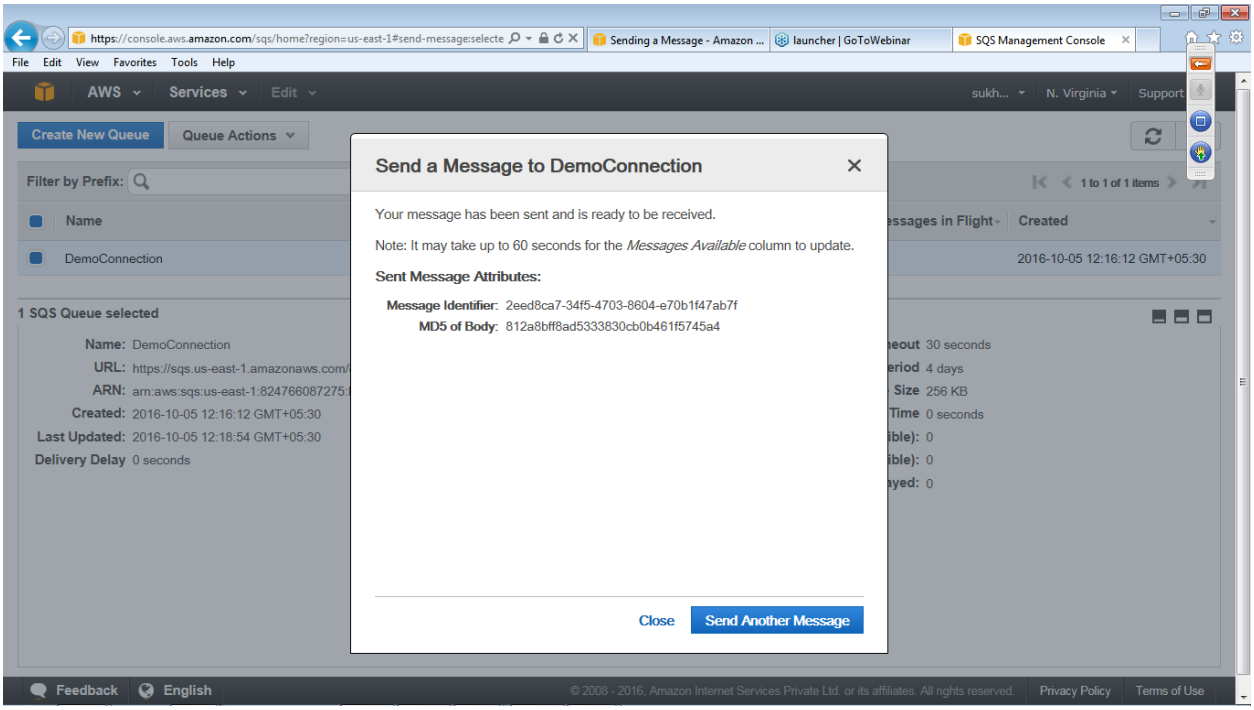
1 In the AWS Management Console select a queue

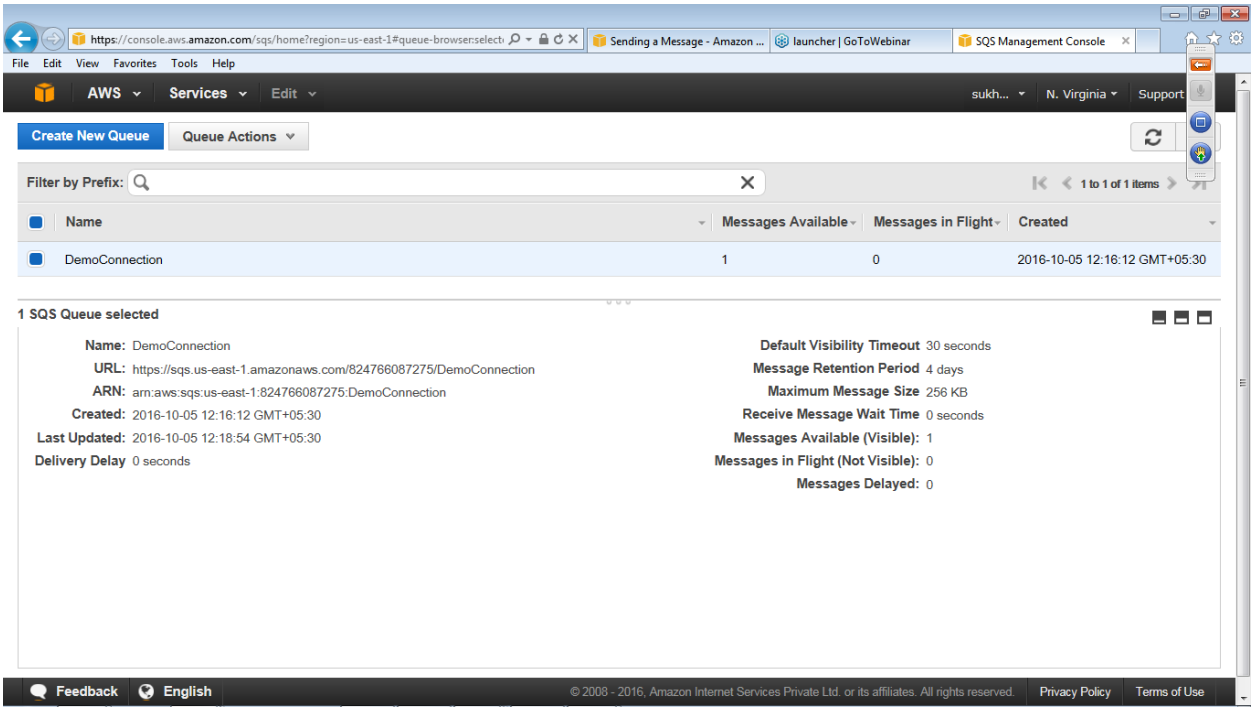


2 In the **Send a Message to** dialog box, enter This is the message text for demo purpose and click **Send Message**



3 In the **Send a Message** confirm and click **Close.**

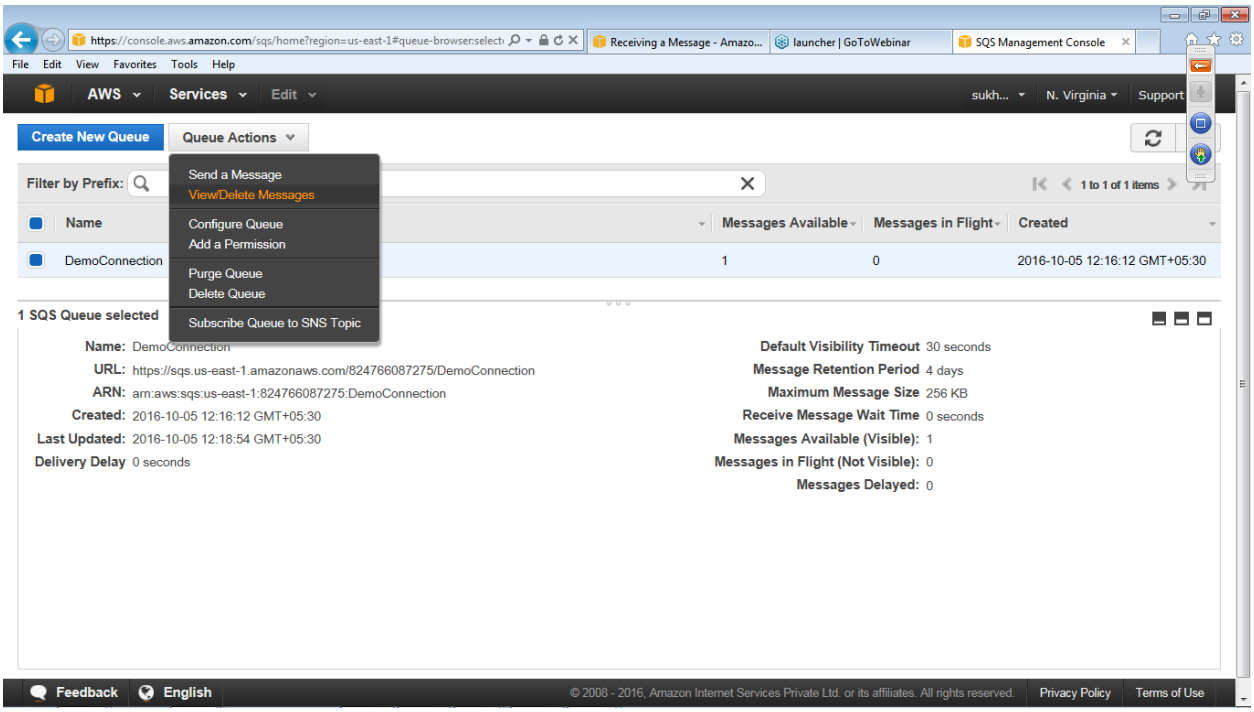




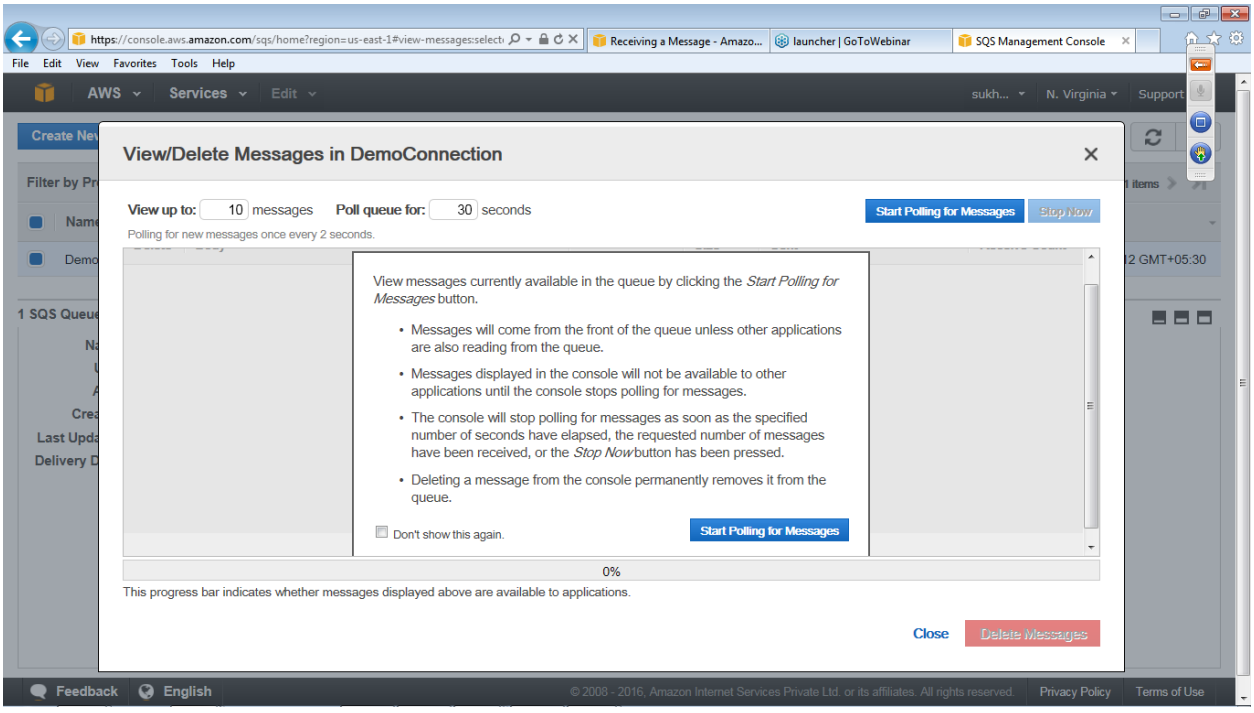
7.2.3 Receiving Message

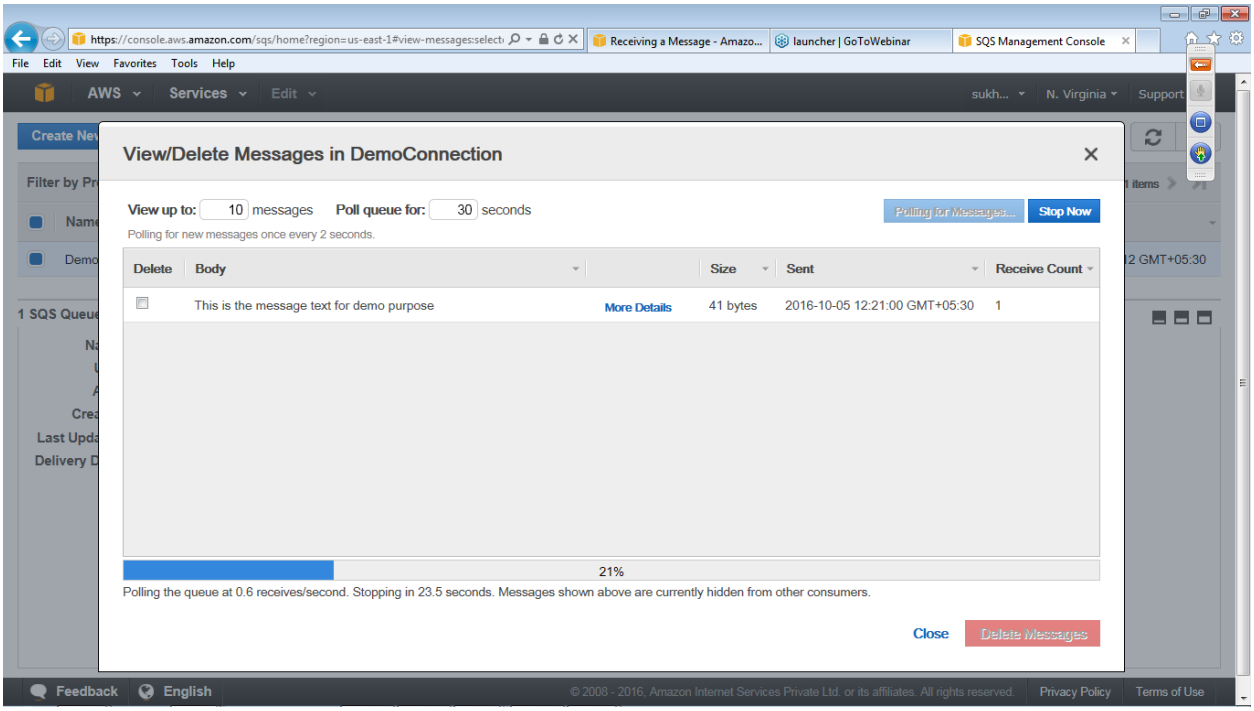
1 In the AWS Management Console, select a queue

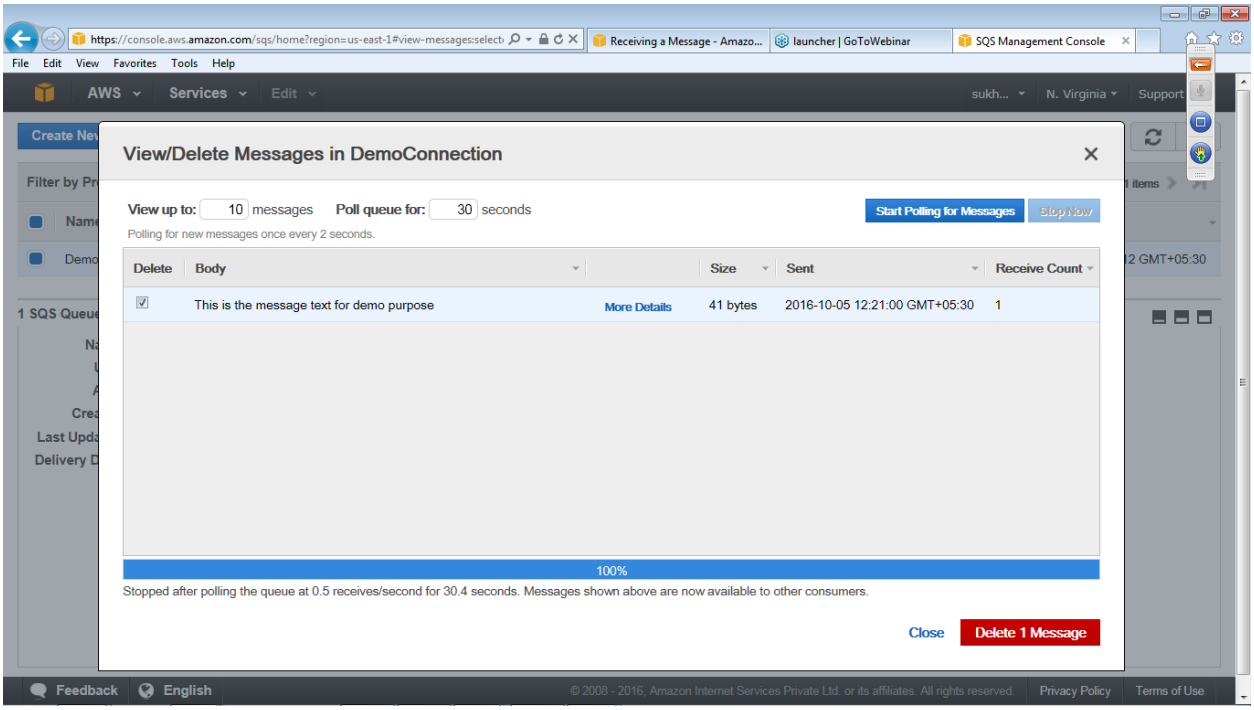
2 Select **View/Delete Messages** from the **Queue Actions** drop-down list



3 Click **Start Polling for Messages** to receive a message from the queue.



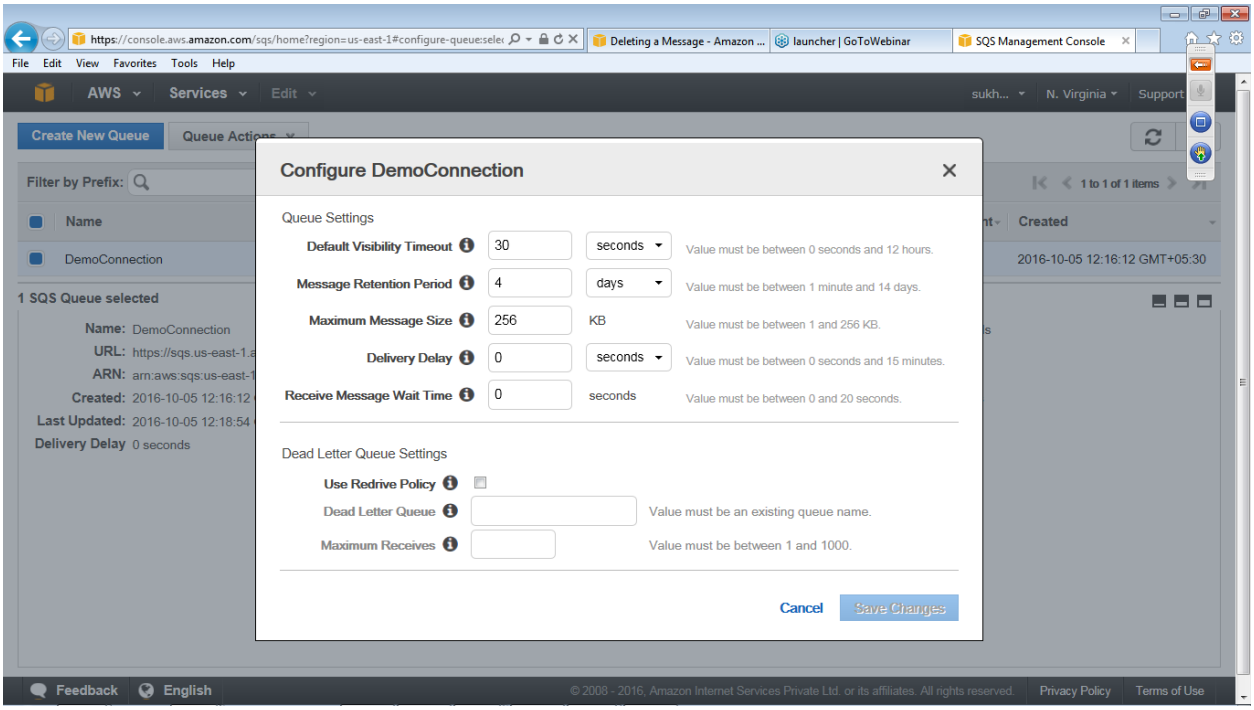




7.2.4 Modify Queue

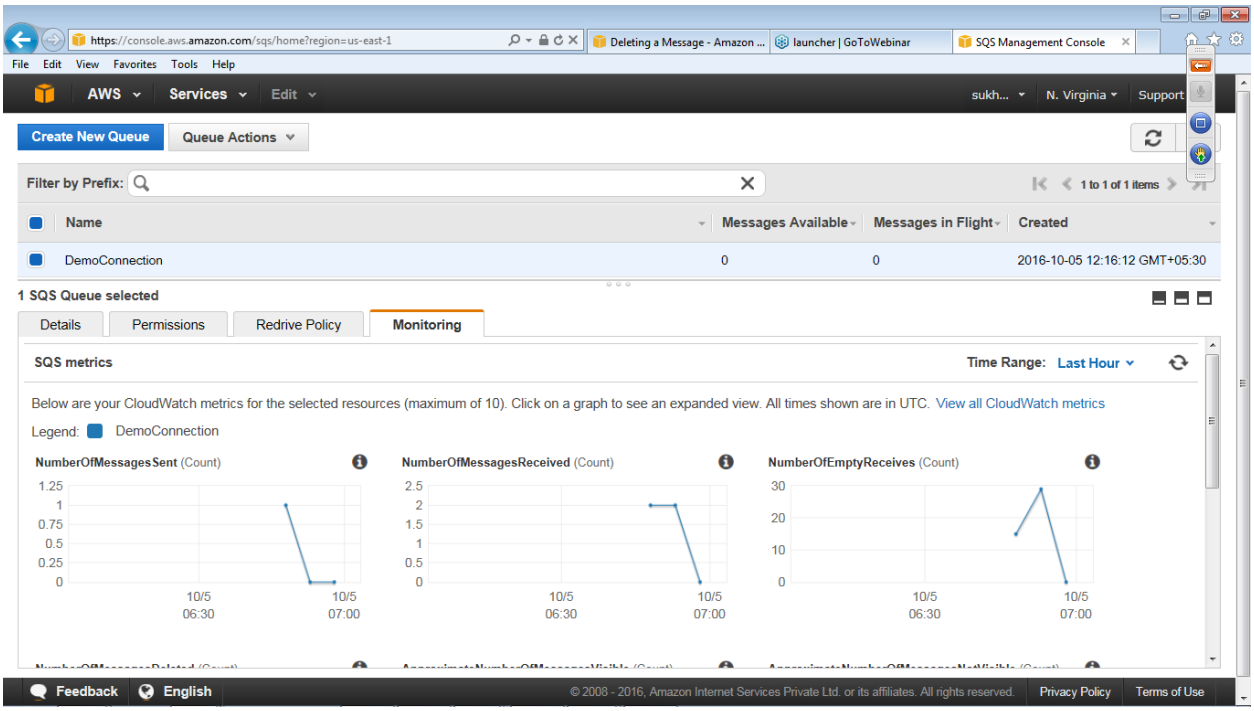
1 In the AWS Management Console, select a queue

2 From the **Queue Action** drop down select the option **Configure Queues**

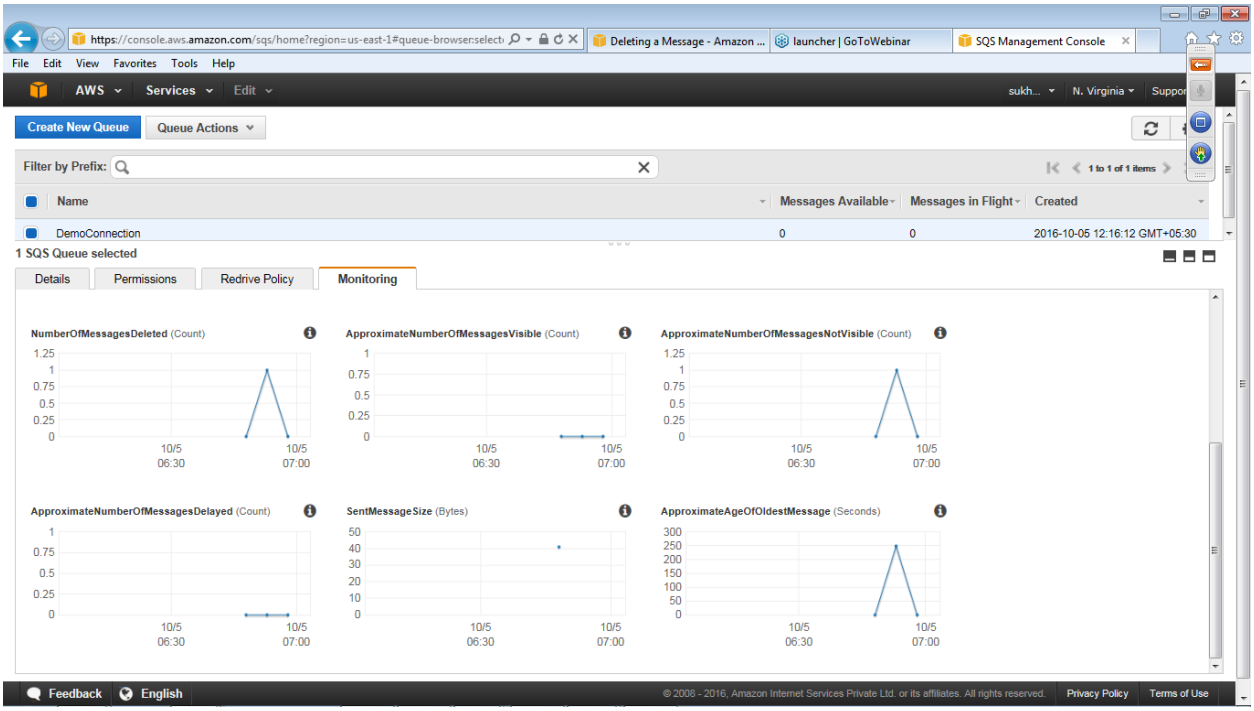


7.2.5 Monitoring Queue

1 In the AWS Management Console, select a queuClick on the **Monitoring tab** and get the details of SQS Metrics



2



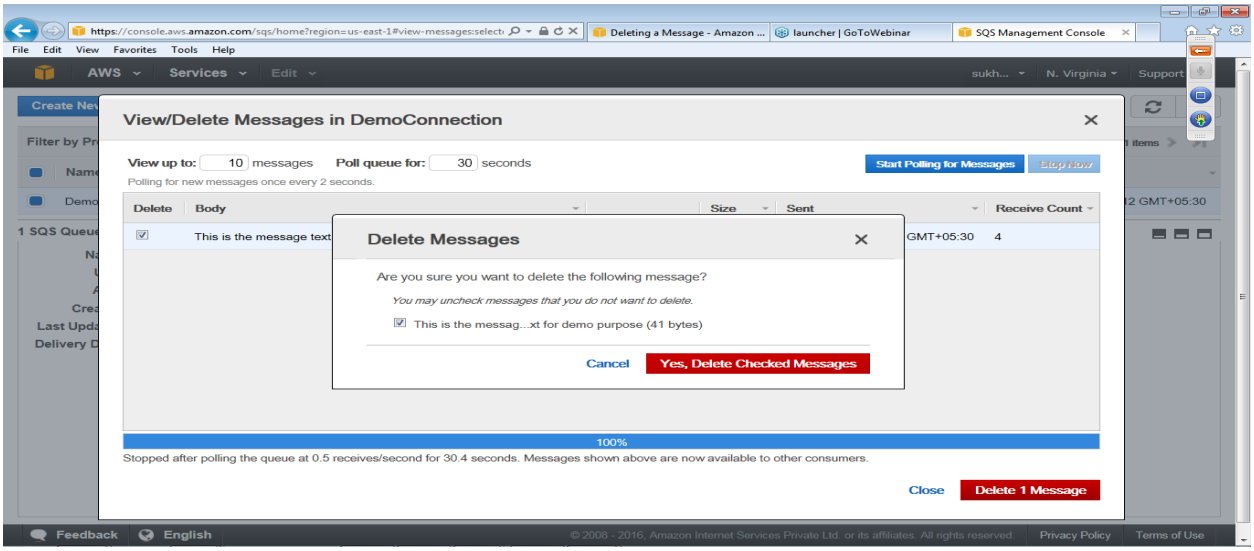
7.2.6 Deleting a Message

If you want to delete messages from your queue one at a time use Delete option

1 In the AWS Management Console, select a queue

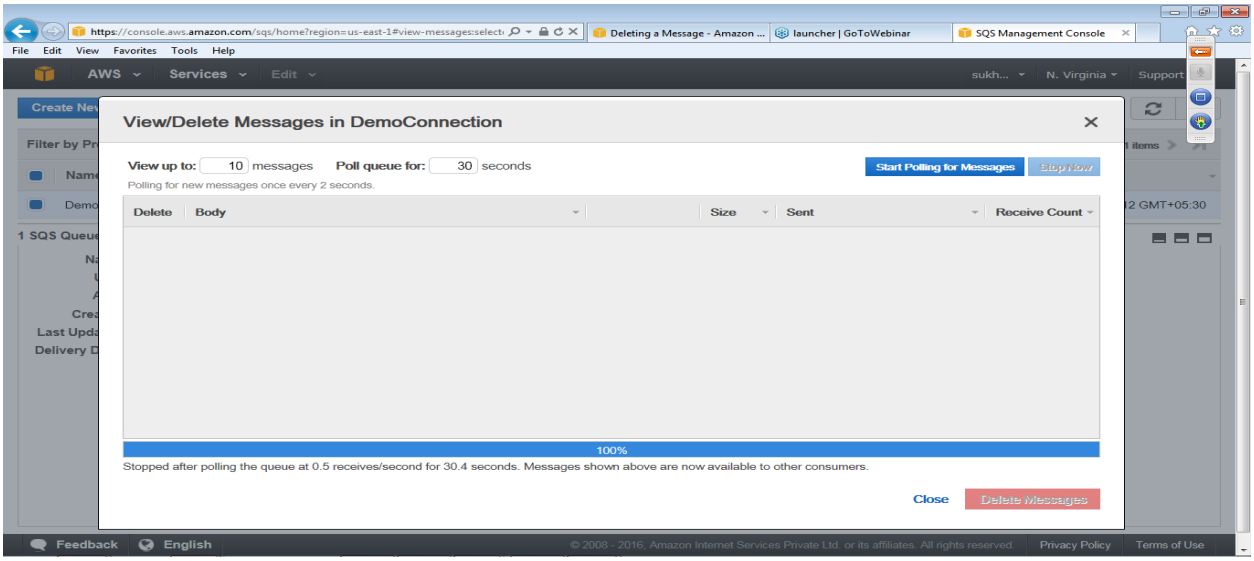
2 Select **View/Delete Messages** from the **Queue Actions** drop-down list

3 Select the message you want to delete



4 Click **Delete 1 Message** to delete the selected message

5 Click **Close** to close the **View/Delete Messages** dialog box



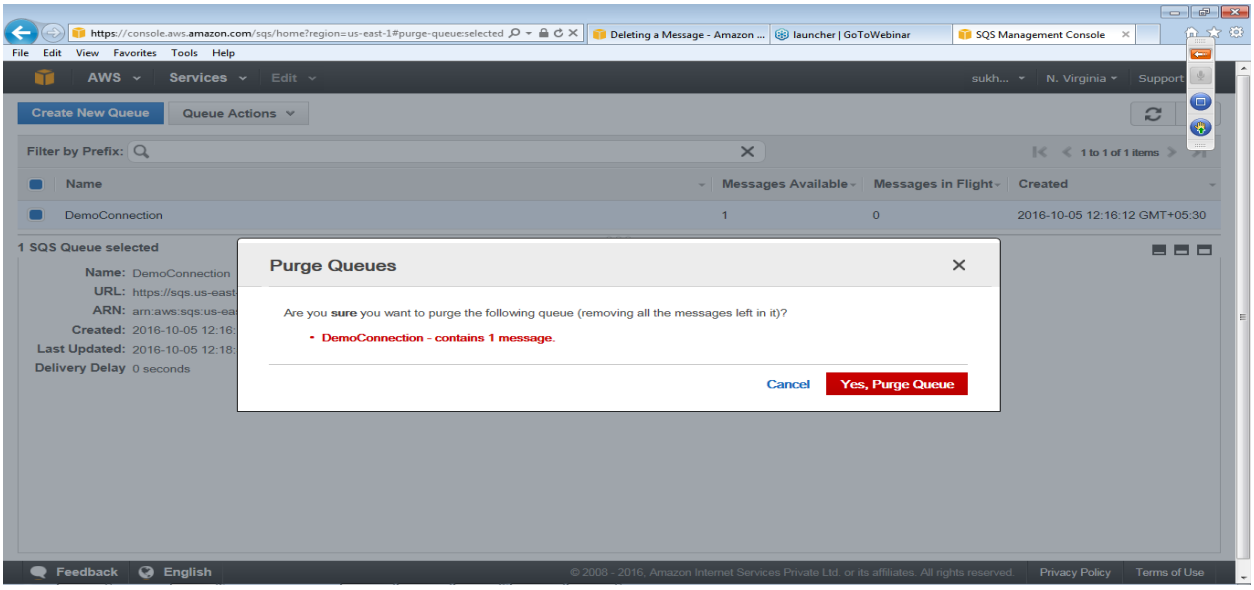
7.2.7 Purging Queue

If you want to delete many messages from your queue at once use purging

1 In the AWS Management Console select a queue

2 Select **Purge Queue** from the **Queue Actions** drop-down list

3 In the **Purge Queues** dialog box, click **Yes, Purge Queue**



In the **Purge Queues** confirmation box click **OK**.

8. SNS MANAGEMENT

Amazon Simple Notification Service (SNS) is a simple, fully-managed "push" messaging service that allows users to push texts, alerts or notifications, like an auto-reply message, or a notification that a package has shipped.

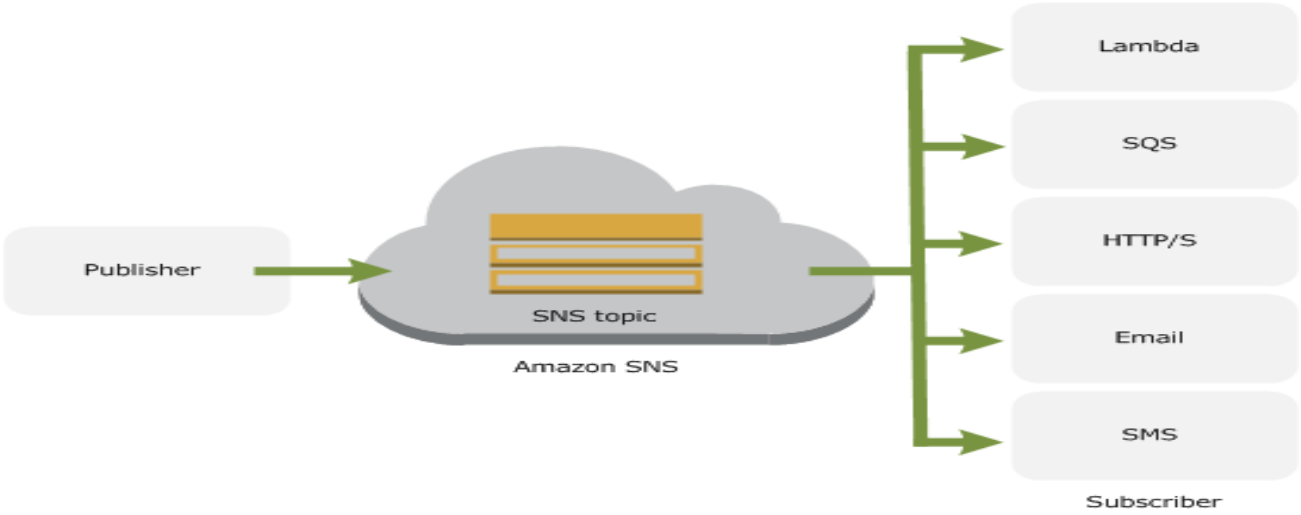
8.1 Objective

Understand the process of Creating, modifying and managing SNS (Simple Notification Service).

8.2 Assumptions

- Simple Notification Service(SNS) is available in the AWS console
- You are using any AWS services such as Amazon CloudWatch, Amazon Ec2 instances or Amazon S3.

  Now the below diagram shows how subscriber can be notified directly from the publisher through SNS for certain created topic.

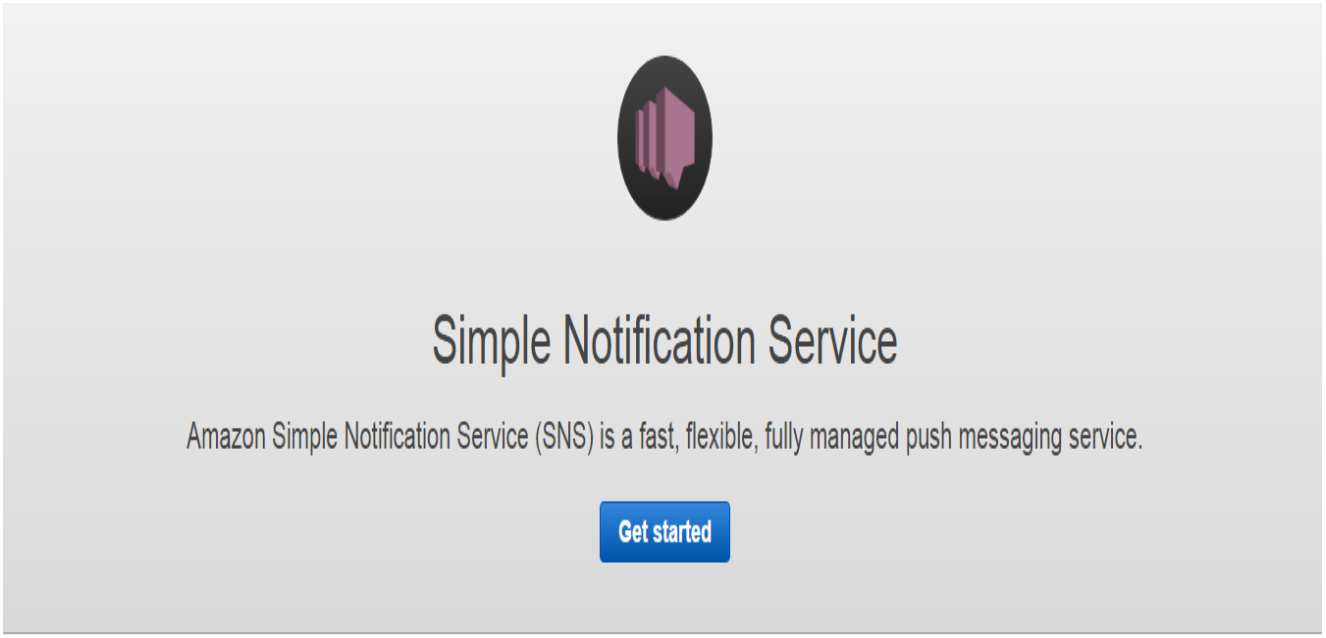


8.3 Procedure

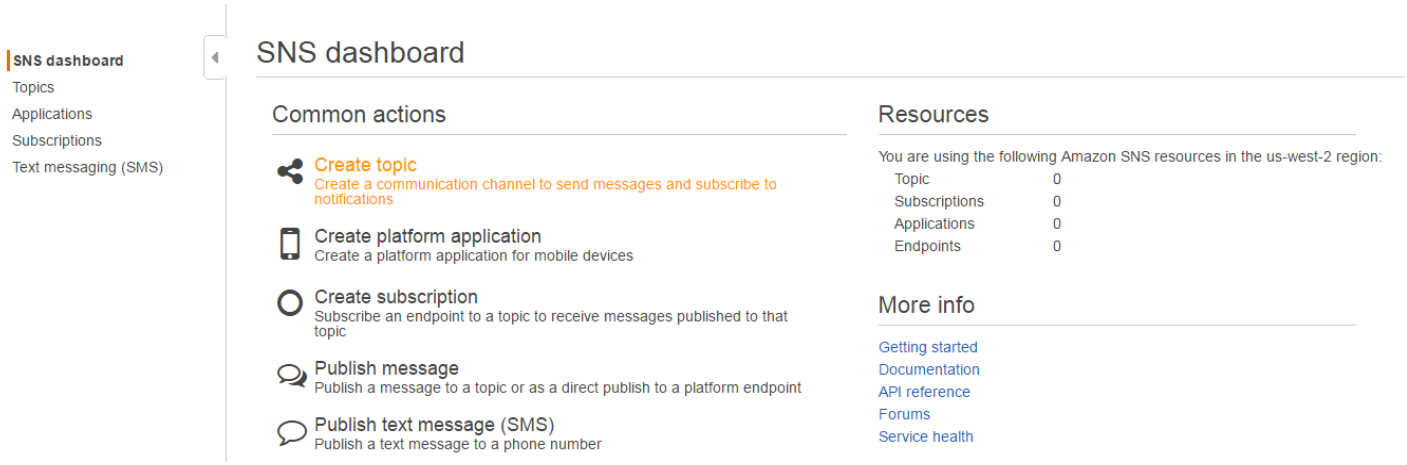
8.3.1 Create SNS(Simple Notification Service).

1. Sign in to AWS Management Console and select **SNS(Simple Notification Service).**

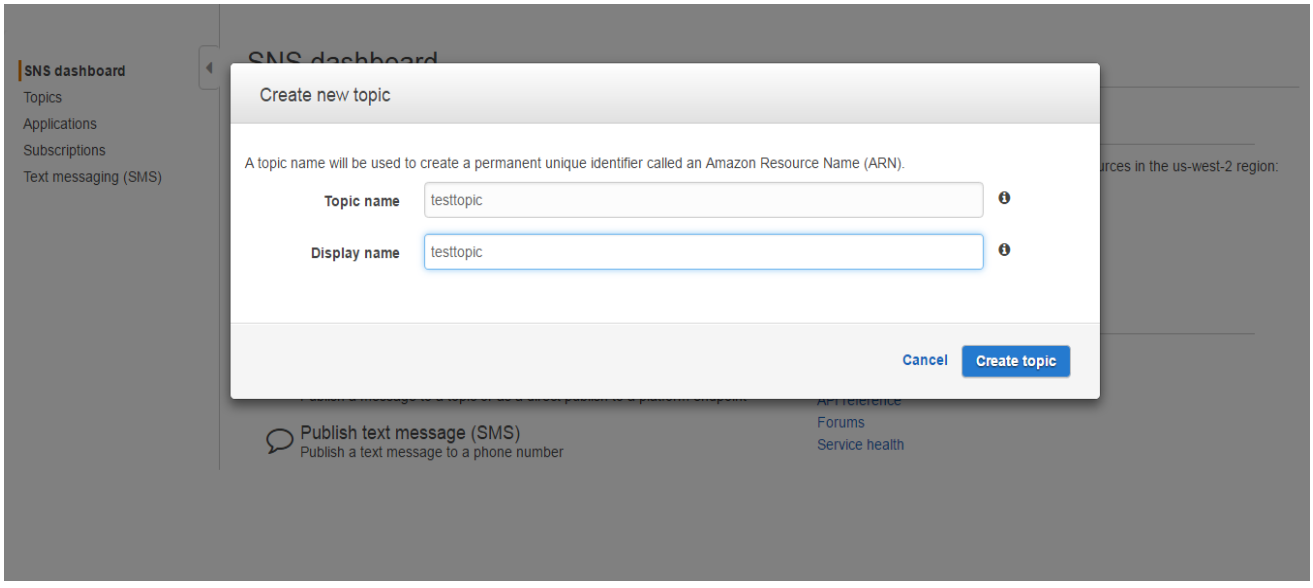
2. Click on Get Started.



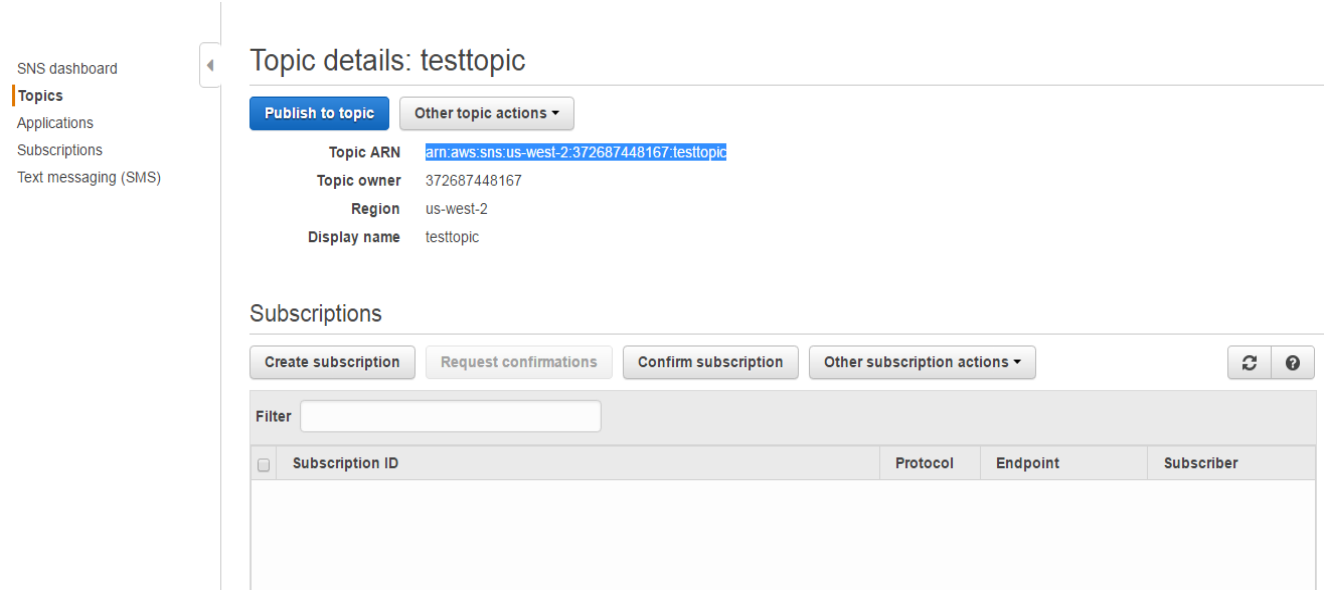
3. Now Click on **Create Topic**.



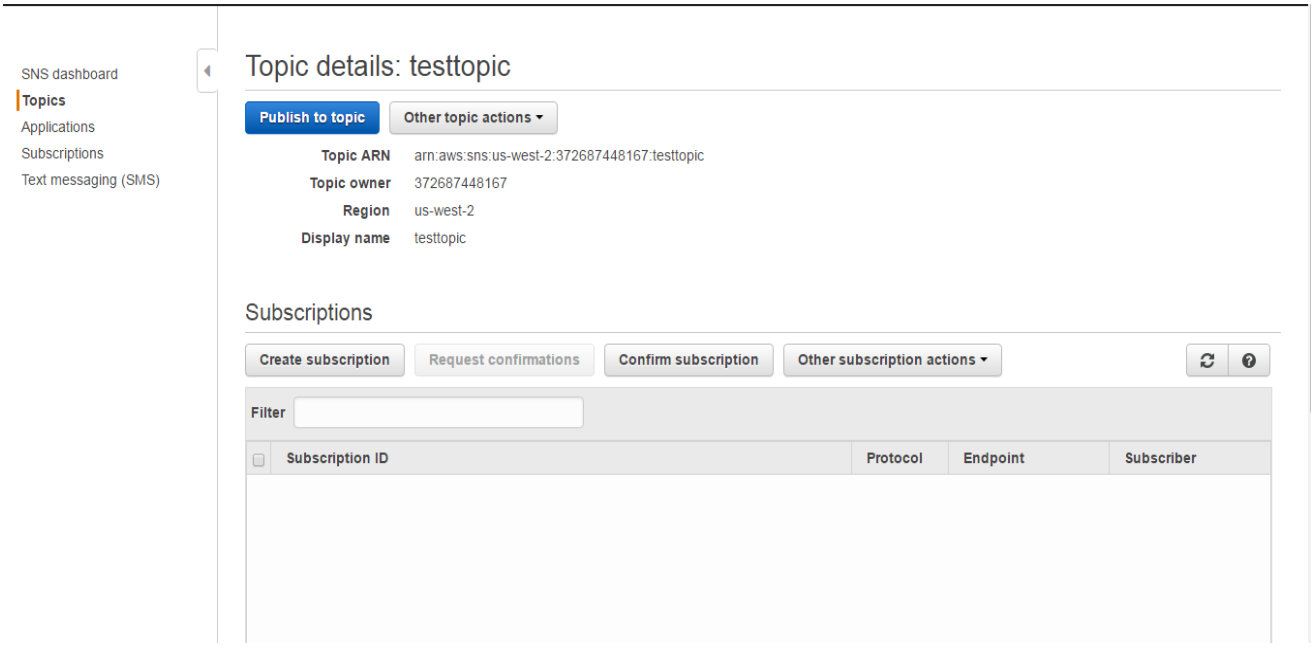
4. In the Create New Topic, Give the **Topic name** and **Displayname** and Click **Create Topic.**



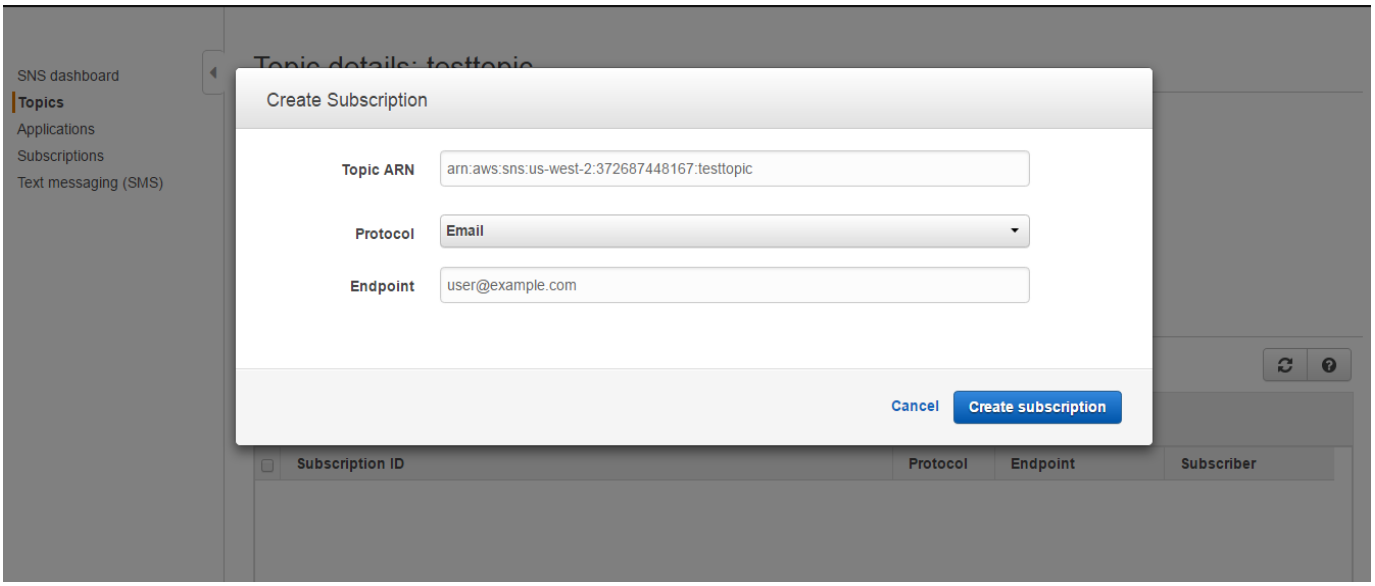
5. In this, copy the **Topic ARN** which will be required in next step.



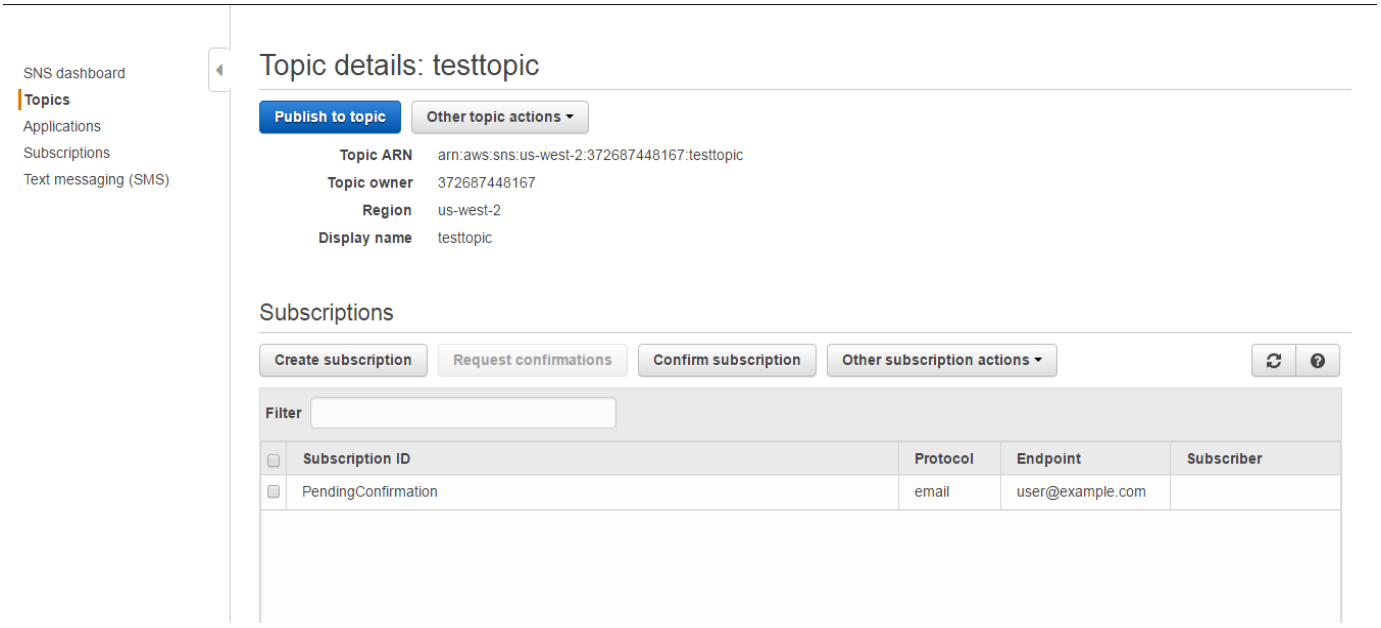
6. In this window Click on **Create Subscription**.



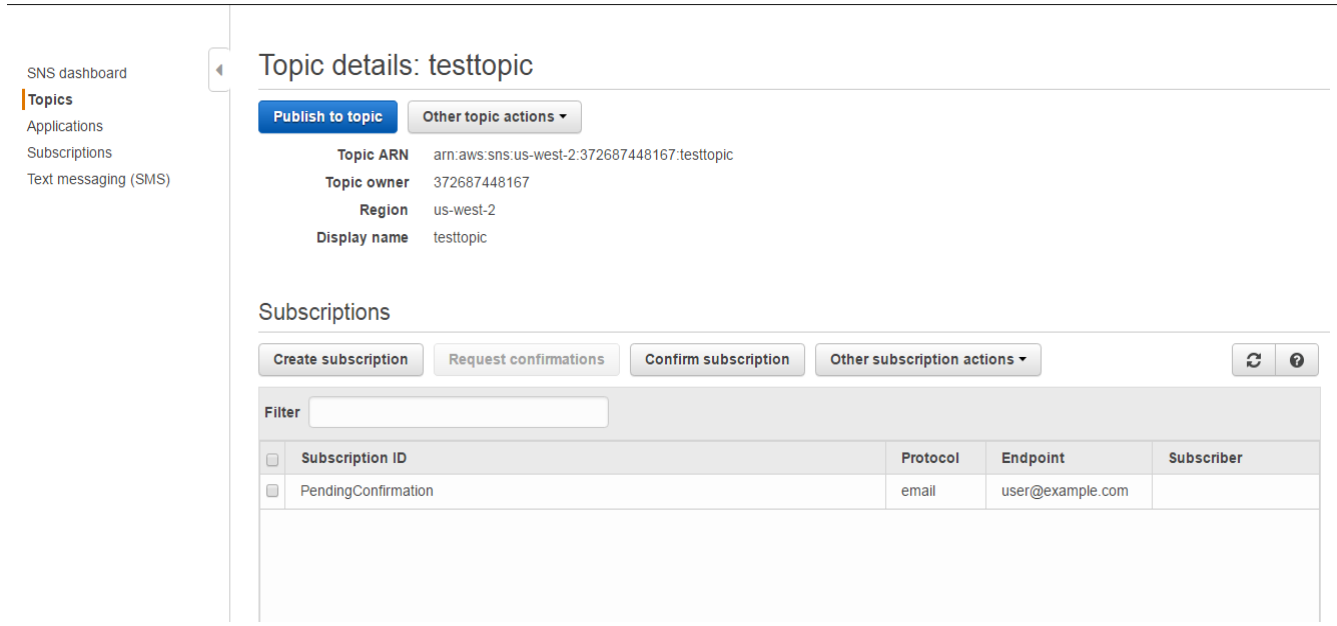
7.After Clicking **Create Subscription**, paste the copied Topic ARN which was copied in earlier step ,in Protocol select Email and in Endpoint Give the Email ID and click **Create Subscription**.



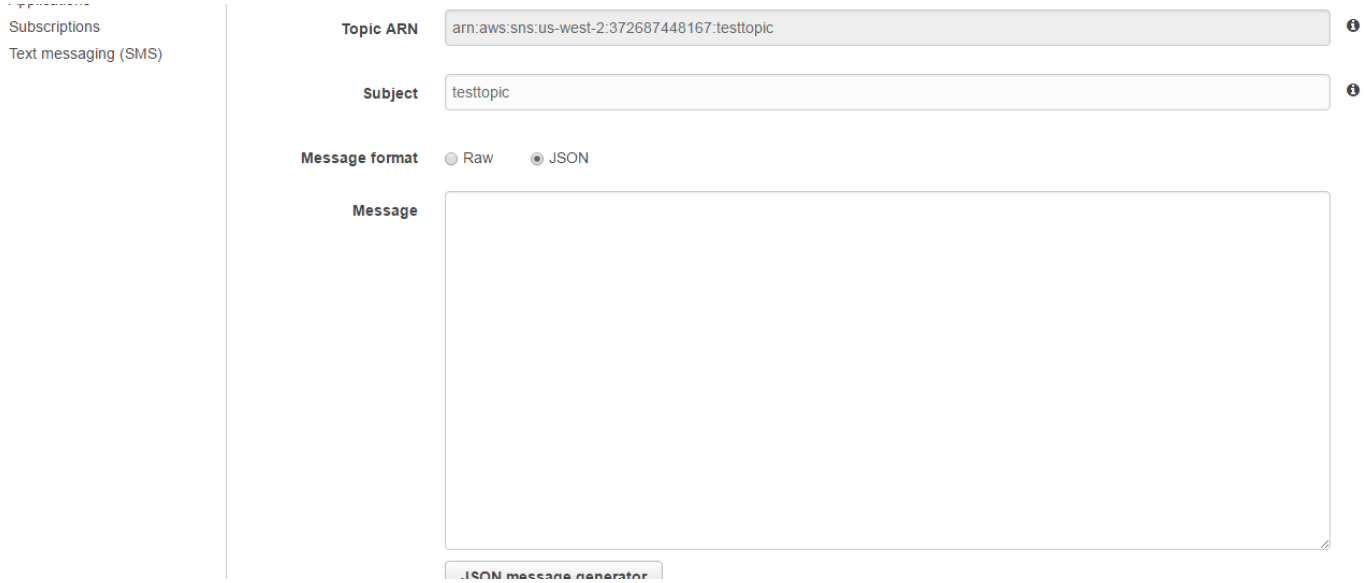
8.After clicking on **Create Subscription** you will receive an email for confirmation just sign in and click on link for confirmation.



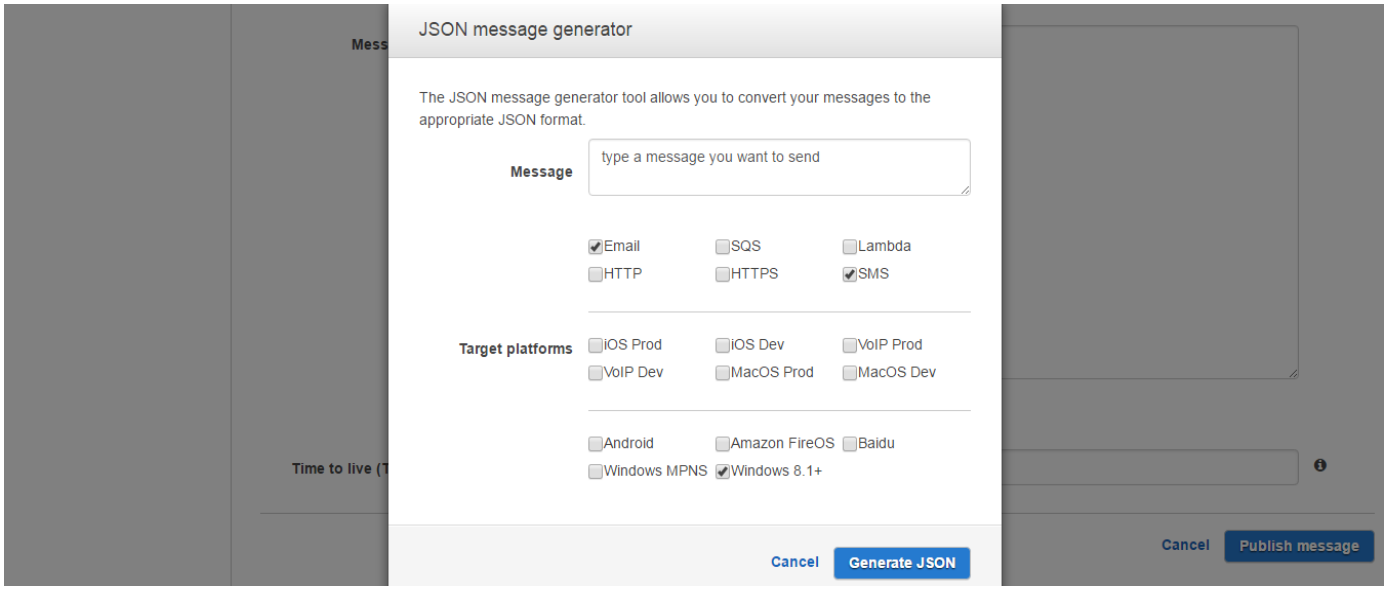
9.In this,click on **Publish to topic.**



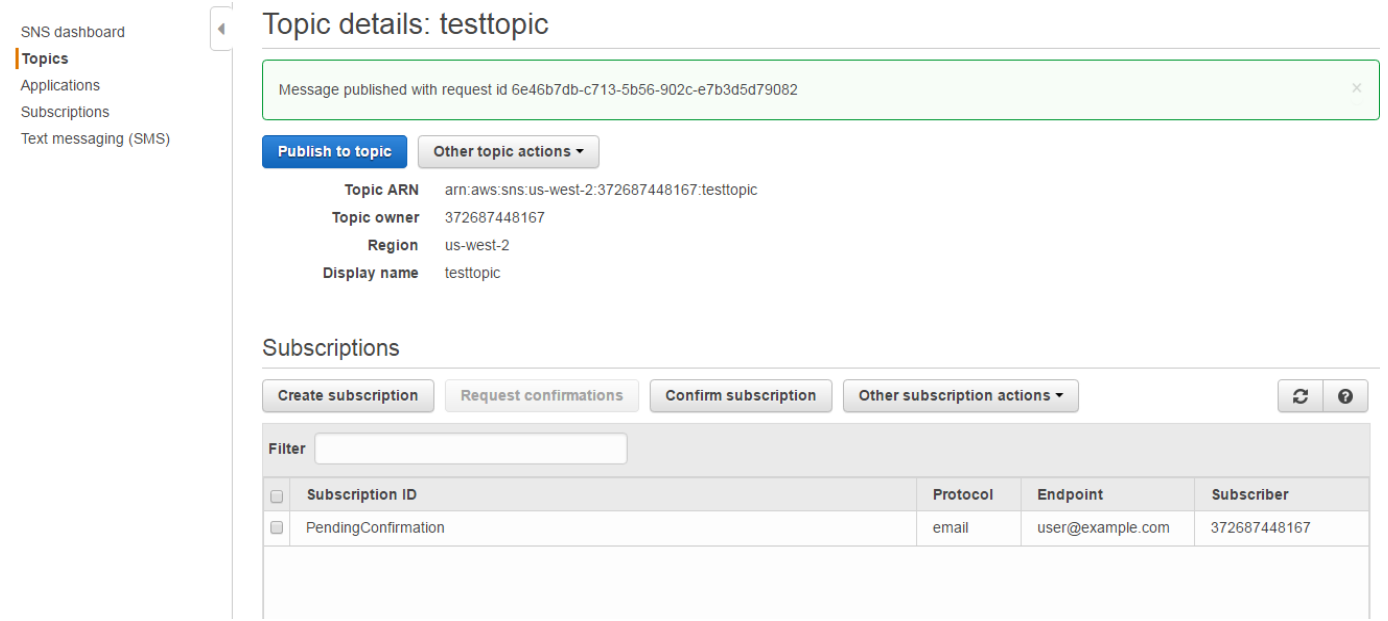
10.In this, type the Subject Name and then click on **JSON message generator**.



11.In the **Message** Box type the brief message and in **Target Platforms** select **Email, SMS** and Select **Windows 8.1+** and then click on **Generate JSON.**

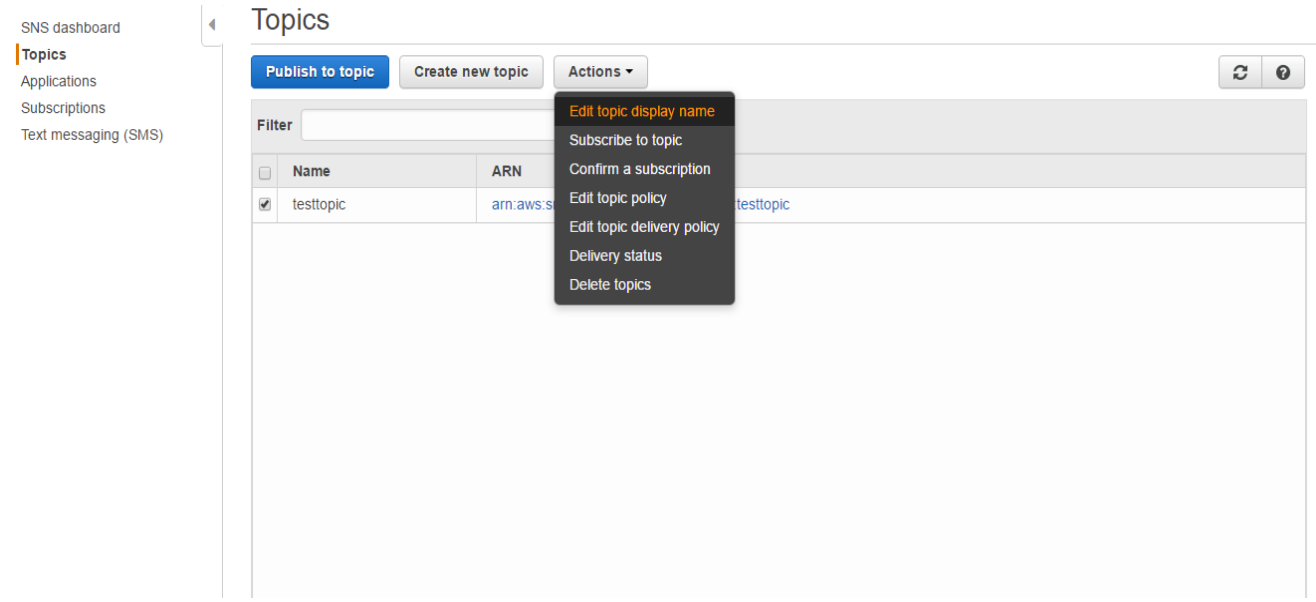


12. After clicking **Generate JSON** following message will be displayed.

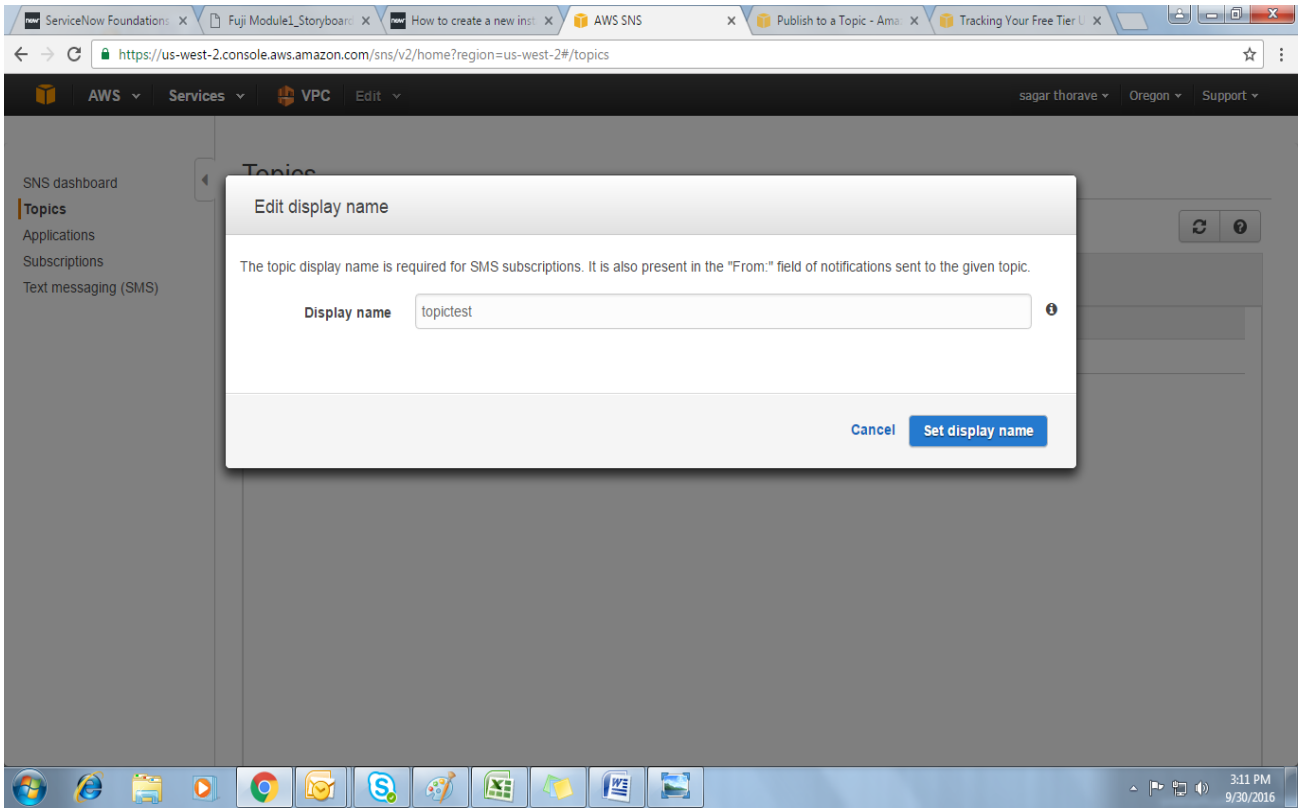


8.3.2 Modifying SNS(Simple Notification Service).

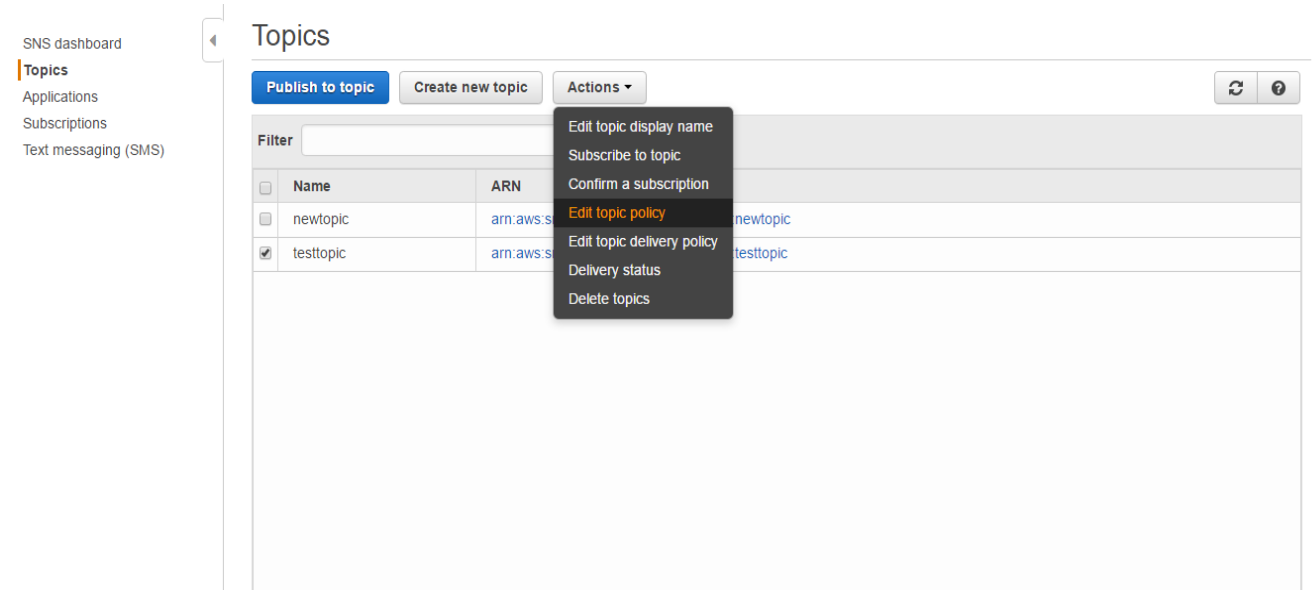
1. Click on the topic created then click on **Actions** and select **Edit topic display Name**.



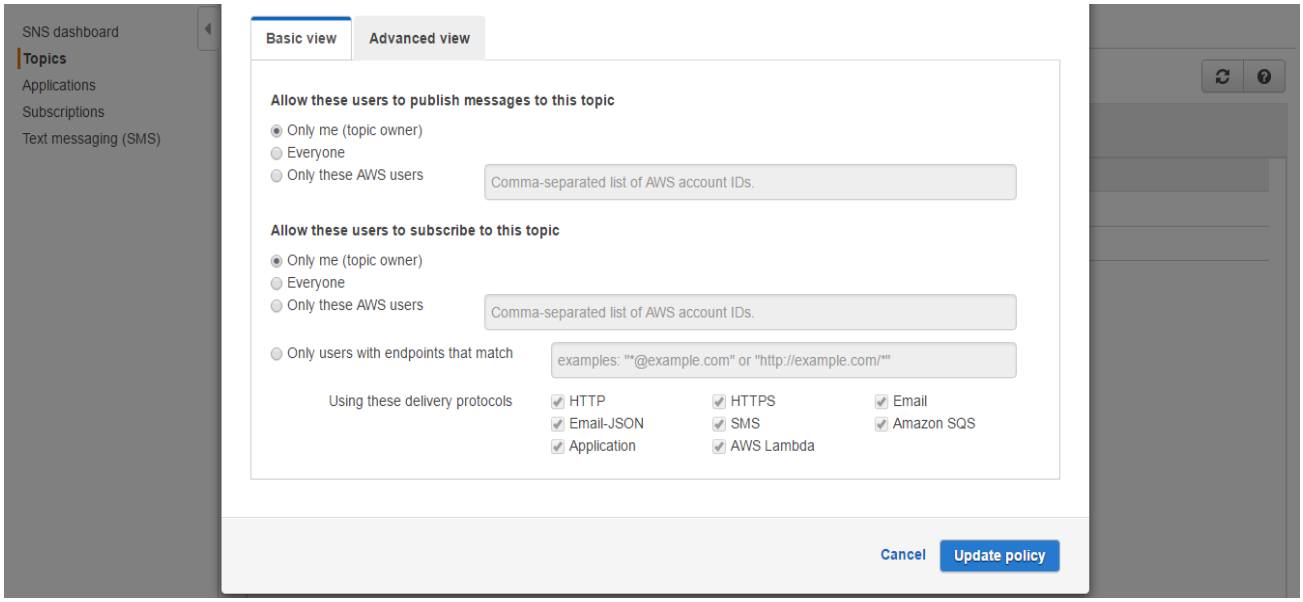
2. Type the name you want to display in **Display Name** and click on **Set display name**.



3. Select topic name and click on **Actions** and select **Edit topic policy**.



4. In this you can allow user restriction,if you don't want any other user to publish the messages in this topic then just click on **only me (topic owner)** and then click on **Update Policy**.



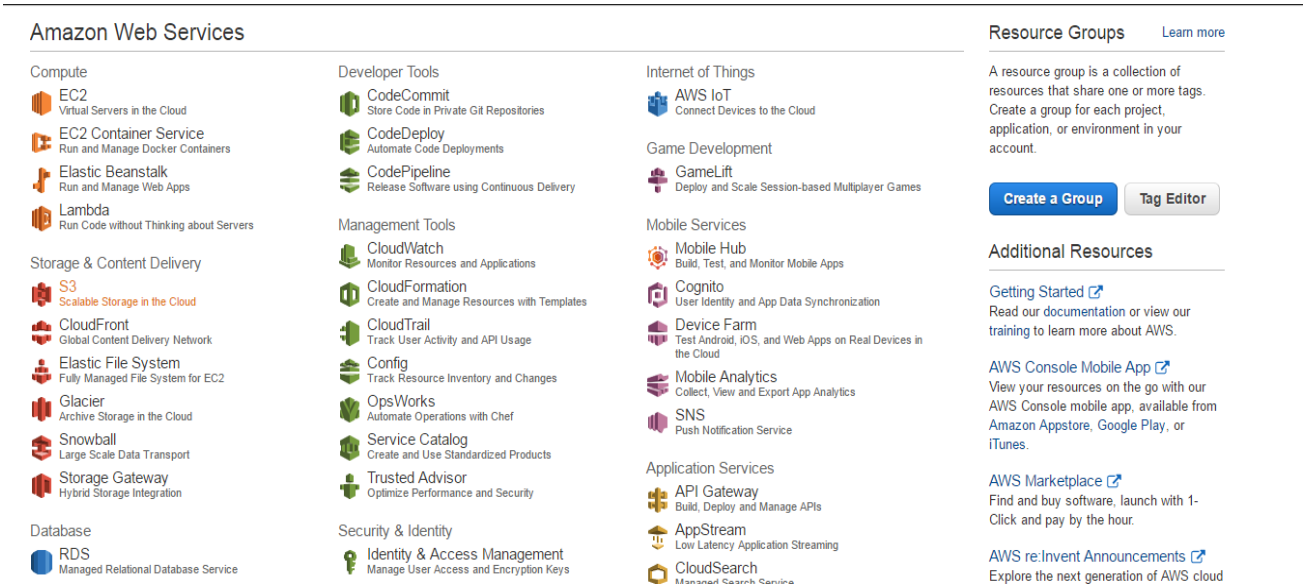
9. S3 & GLACIER MANAGEMENT

9.1 Objective

To understand the process of Creating and deleting S3 Buckets and Folders

9.2 Procedure

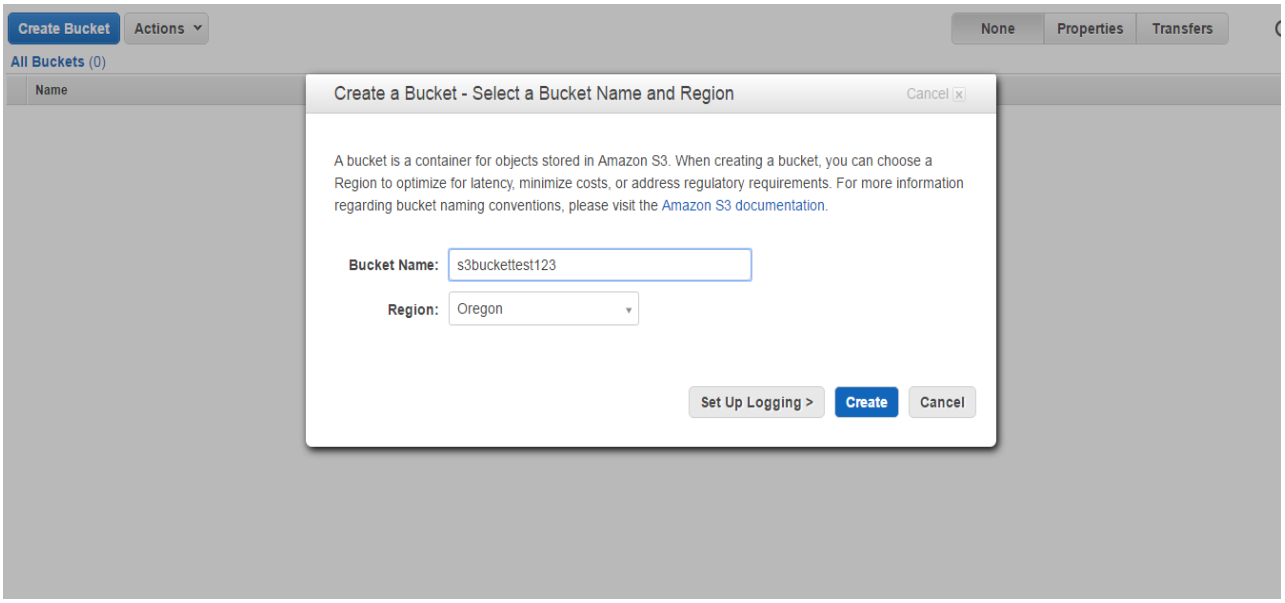
Sign in to AWS Management Console and select **S3(Simple Storage Service)**



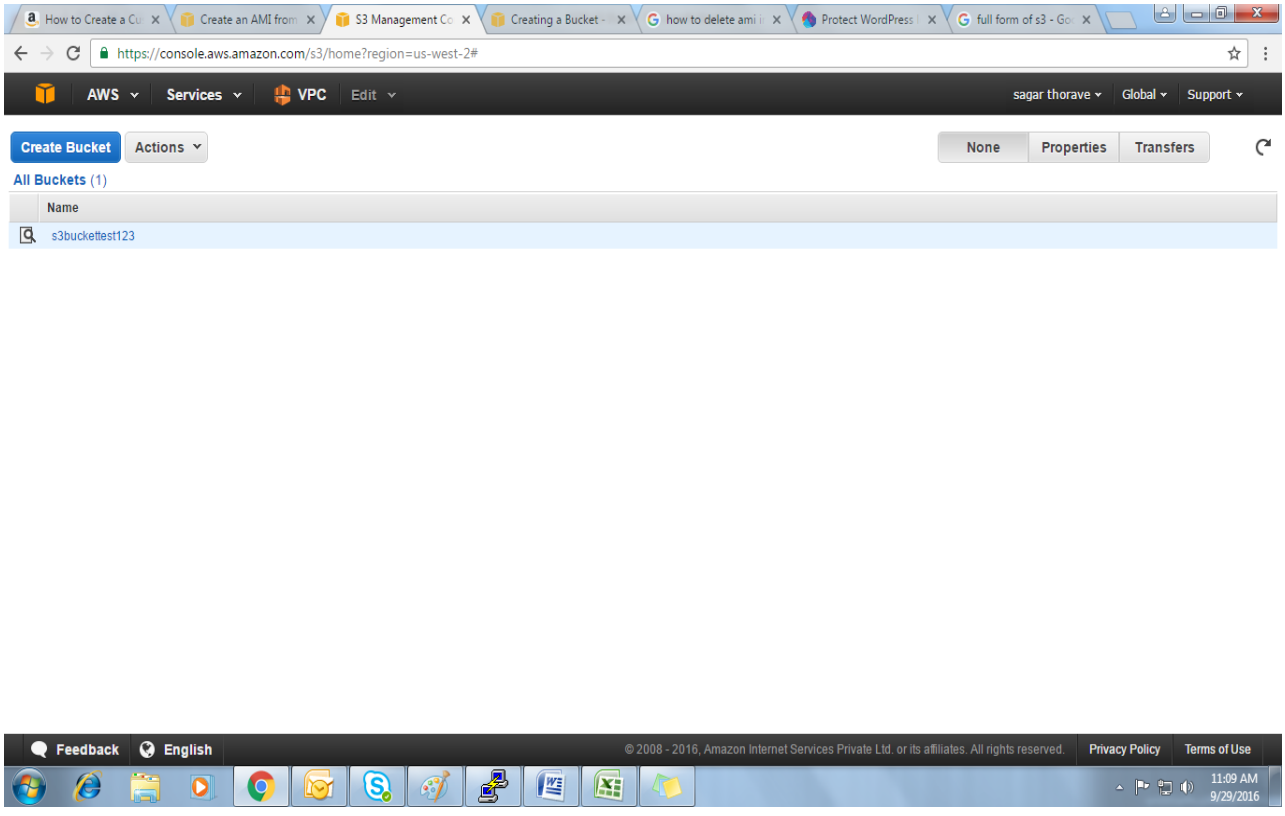
9.2.1 Create Bucket

Click on **Create Bucket** button.

7. After clicking **Create Bucket** Button , give the bucket name and select the region in which you want to create bucket and click **Create**.

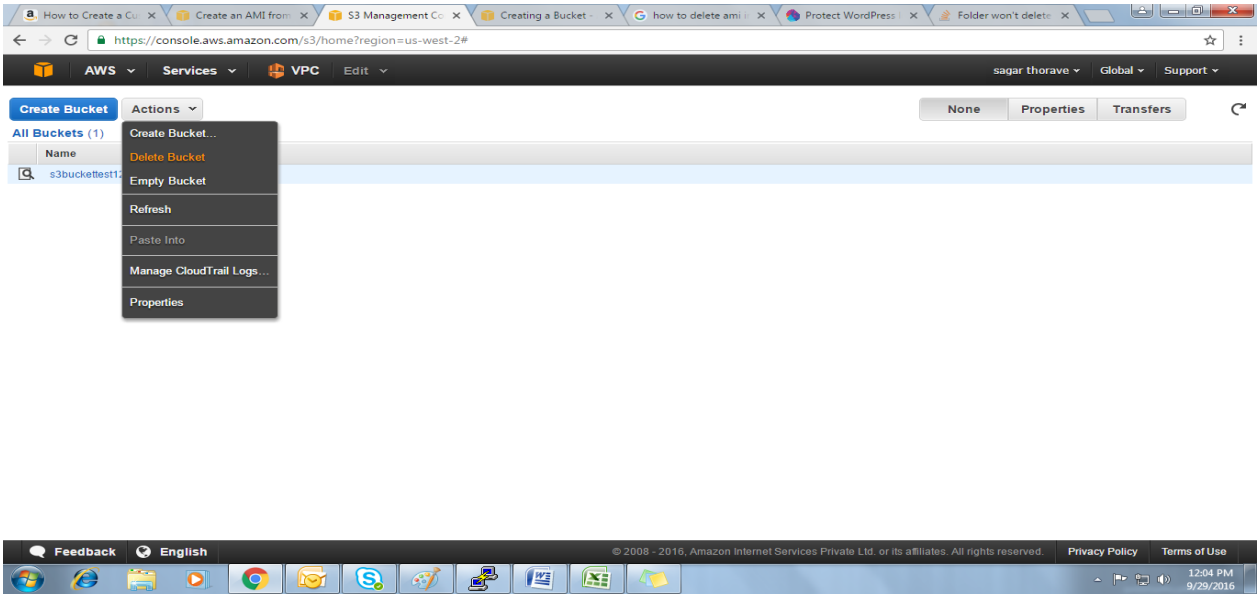


8. Here you can see that the Bucket named as s3buckettest123 has been created.

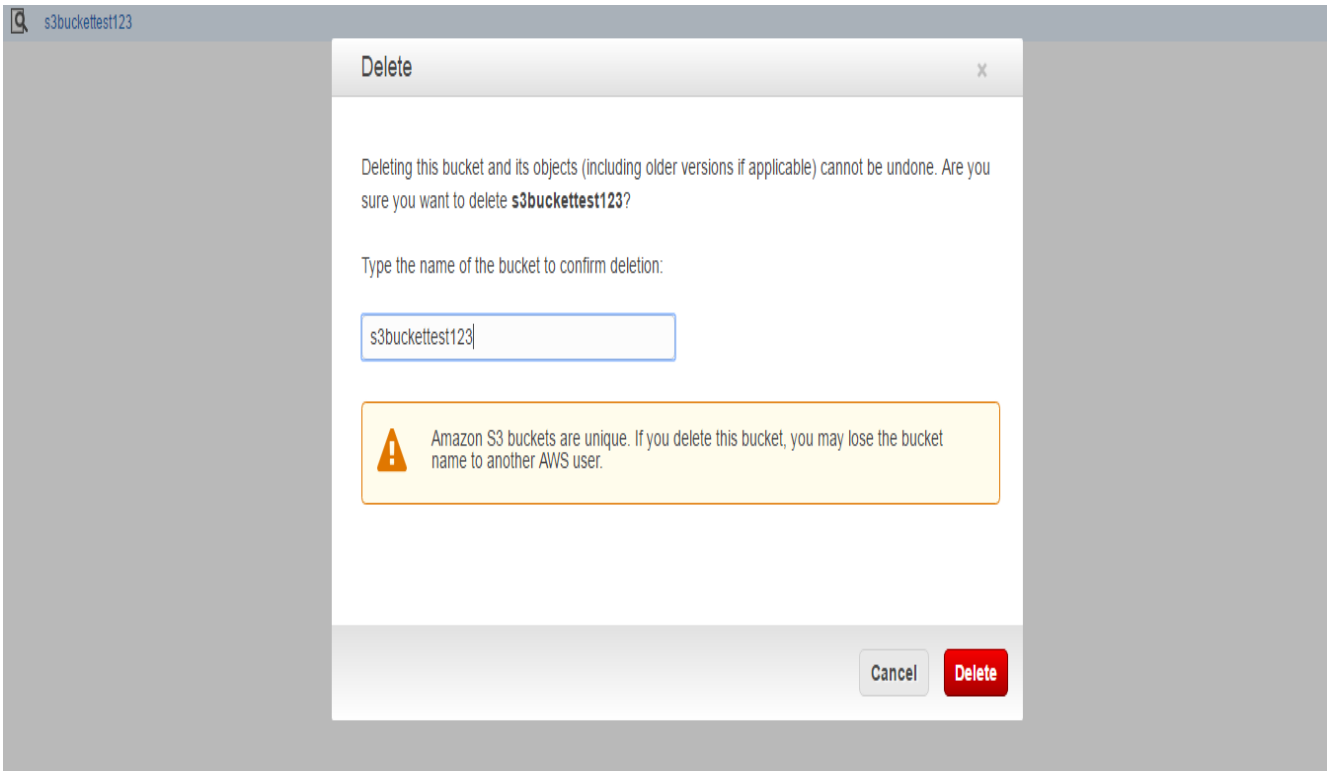


9.2.2 Delete Bucket

1. Select the bucket you created and click on **Actions** and select **Delete Bucket**.

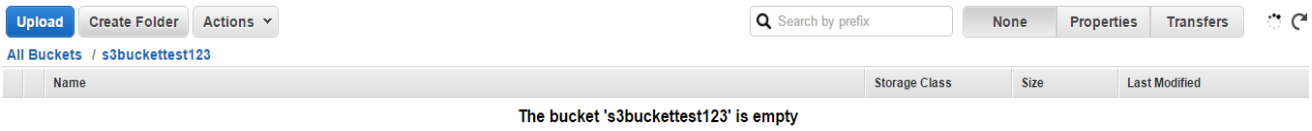


2. After selecting **Delete Bucket** option give the name of the bucket and click **Delete**, the bucket will be deleted.

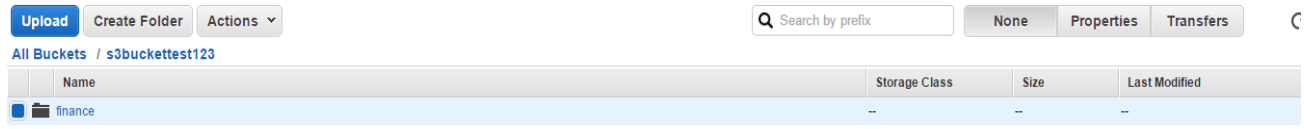


9.2.3 Create Folder

5. Click on the bucket which you have created and then click on **Create Folder** button.

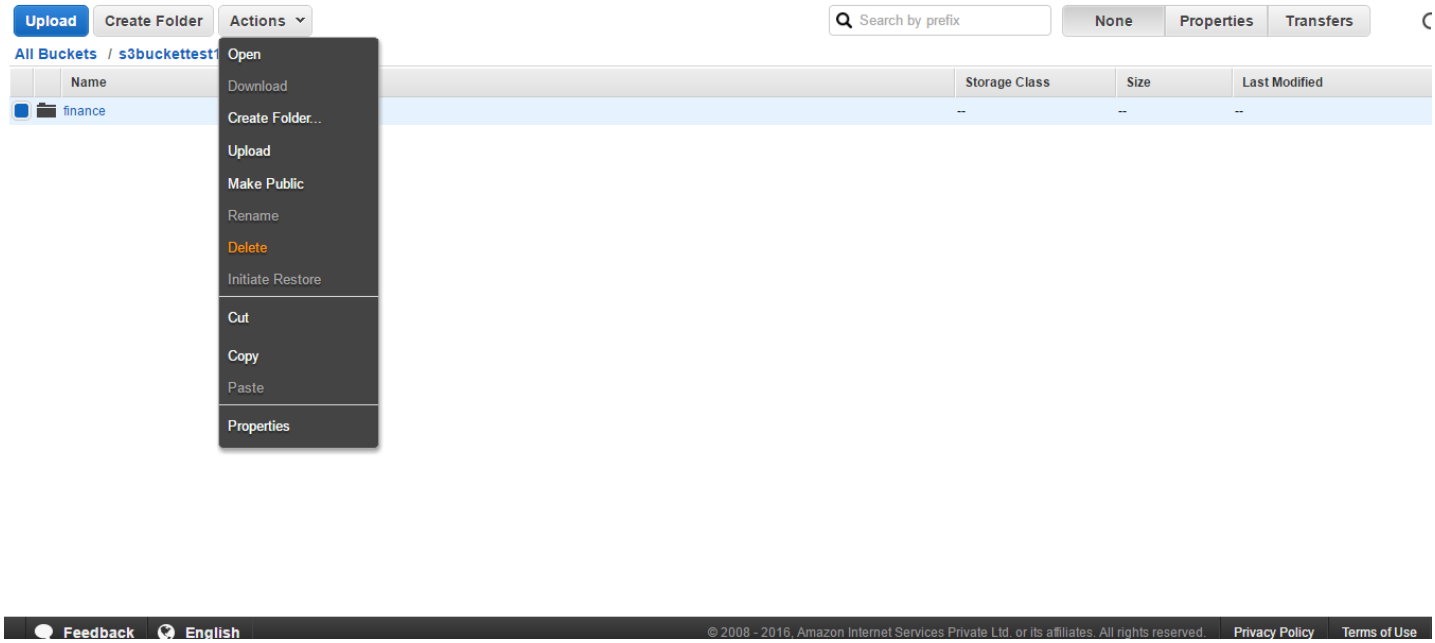


6. After clicking **Create Folder**,give the name of the folder you want to create in my case I have given finance.

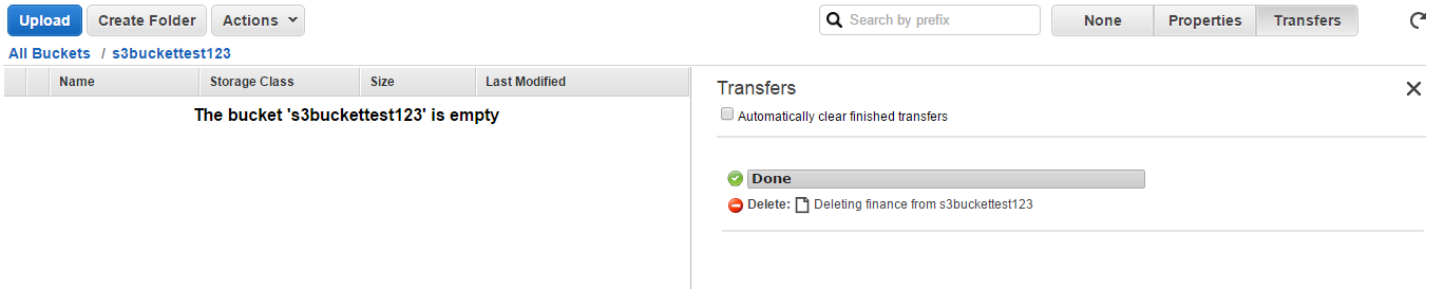


9.2.4 Delete Folder

1. Select Folder you created and then click on **Actions** and select **Delete.**



2. Click **yes** when prompted for Confirmation and then you can see that the folder named finance has been deleted.



10. EFS MANAGEMENT

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud. Amazon EFS is easy to use and offers a simple interface that allows you to create and configure file systems quickly and easily.

10.1 Objective

To understand the process of Elastic File System

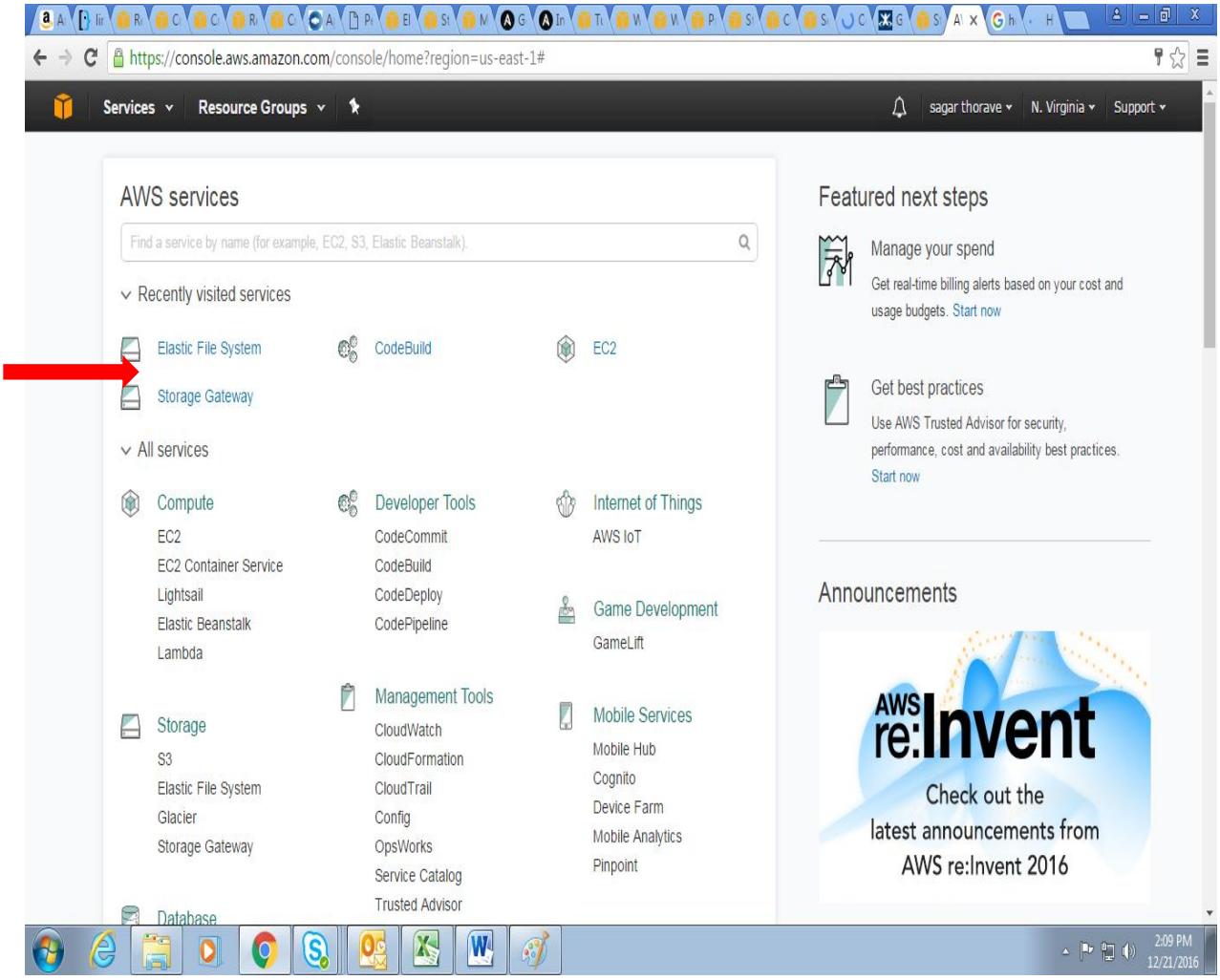
10.2 Assumption

- You’re having an AWS account.
- You're familiar with using the Amazon EC2 console to launch instances.
- You have a default VPC in the region that you're using for this Getting Started exercise. If you don't have a default VPC, or if you want to mount your file system from a new VPC with new or existing security groups, you can still use this Getting Started exercise as long as you configure Security Groups for EC2 Instances and Mount Targets. You have not changed the default inbound access rule for the default security group.

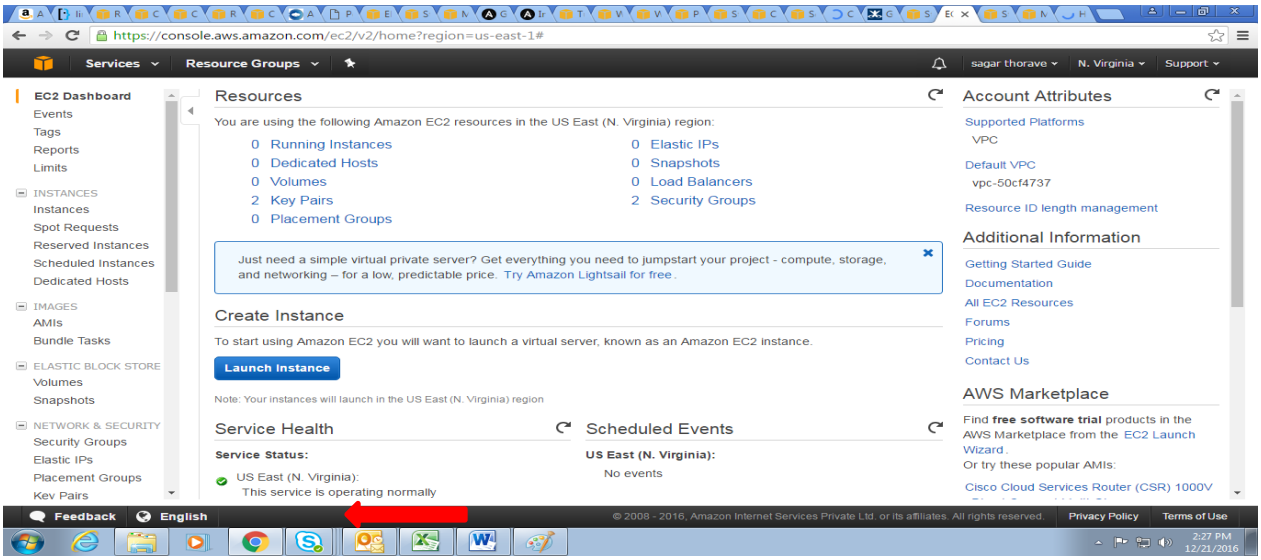
10.3 Procedure

10.3.1 Create an EC2 Instance

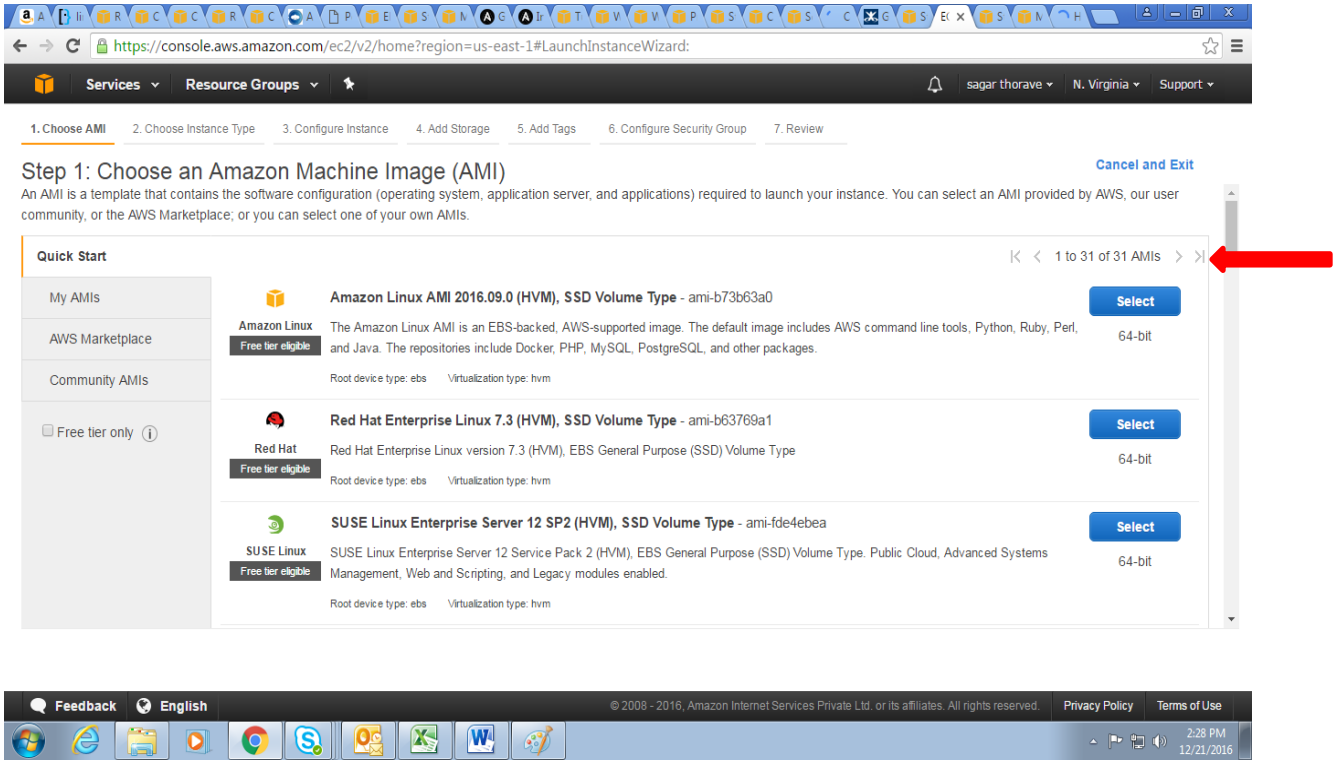
- Sign in to AWS Management Console https://console.aws.amazon.com/efs/ and click on EC2 service



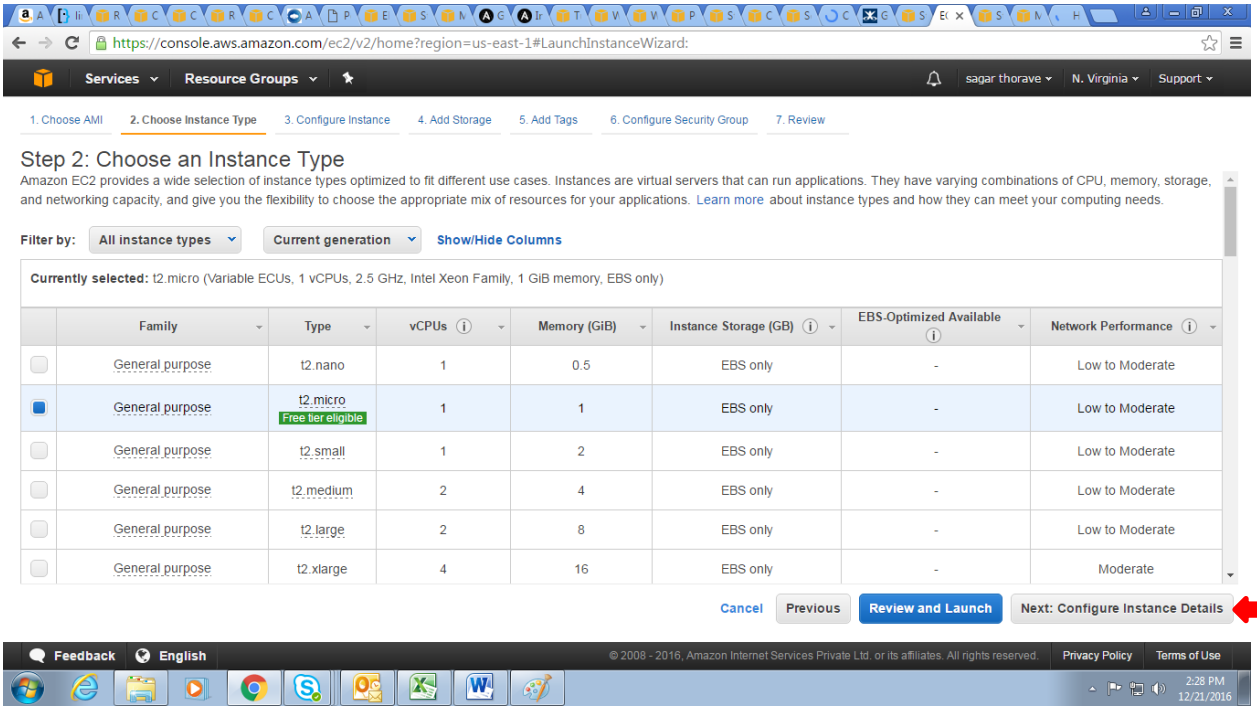
4 Click on Launch Instance button



5 Choose an Amazon Machine Image (AMI), select Amazon Linux AMI 2016.9.0 (HVM) as it has preinstalled nfs-utils which is required to install for the following process

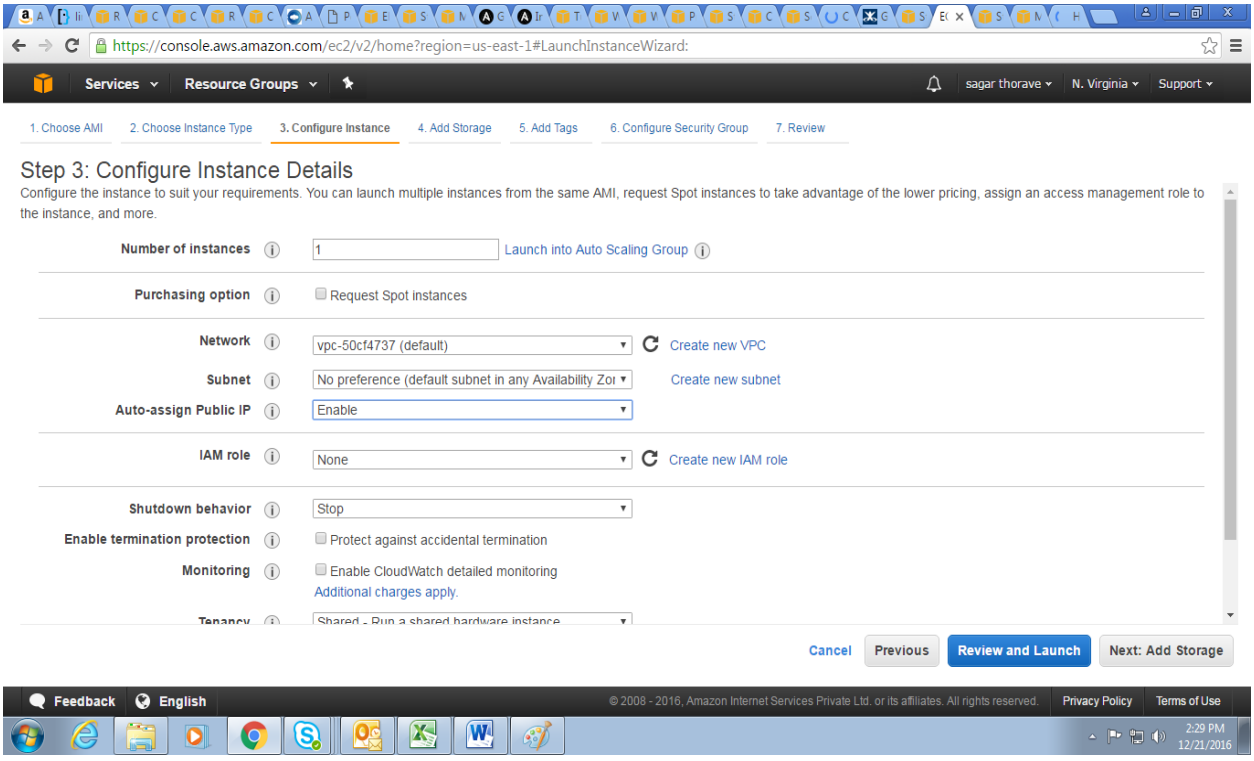


6 Choose an Instance type and then choose **Configure Instance Details**

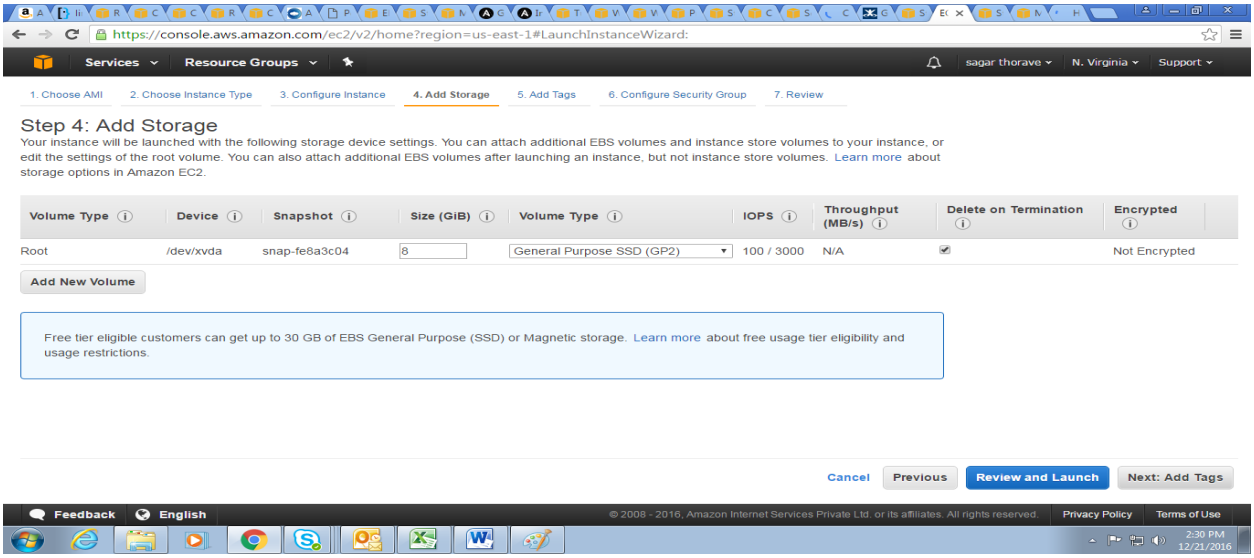


7 **Configure Instance Details**, choose **Network**, and then choose the entry for your default VPC.

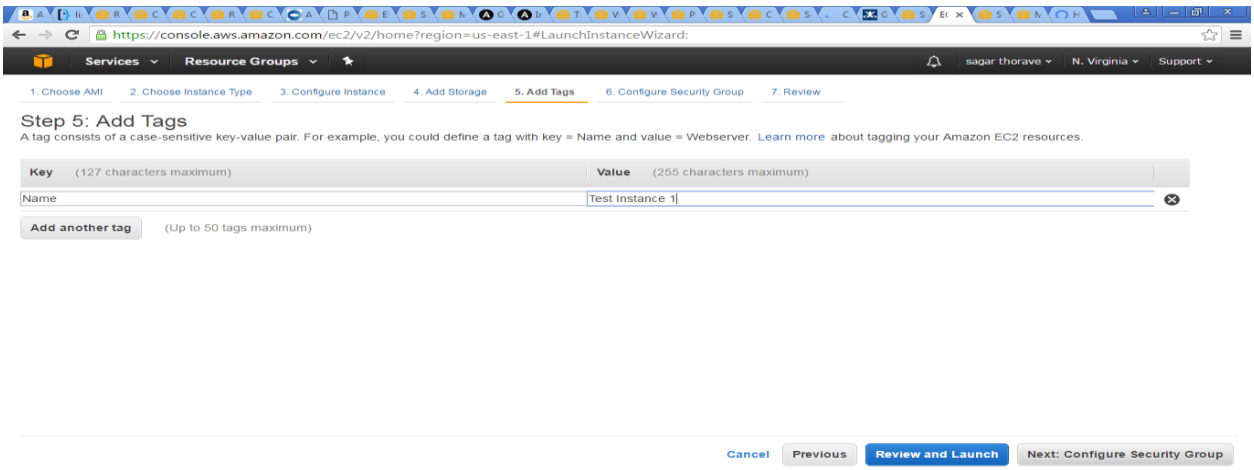
It will look something like vpc-xxxxxxx (172.31.0.0/16) (default) and select **Add storage.**



8 Select the storage you want and choose **Add Tags**



9 In **Add Tags**, give the name to an instance and choose **Configure Security Group**



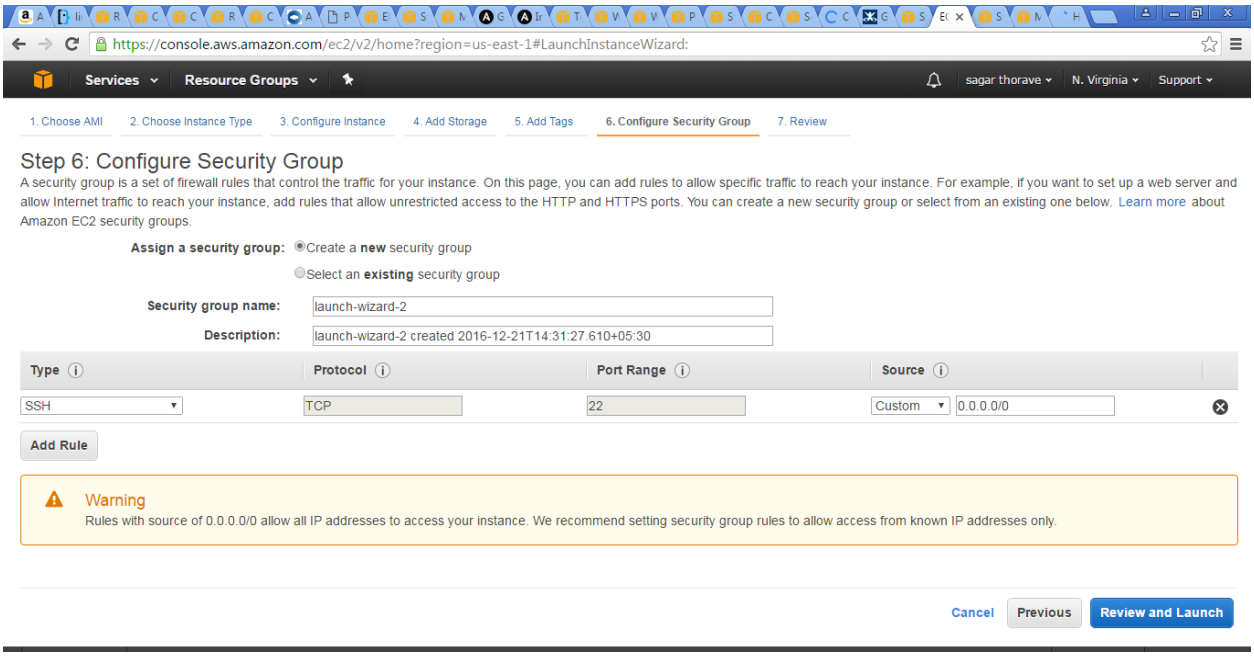
10 In **Configure Security Group**, review the contents of this page, ensure that **Assign a security group** is set to Create a **new** security group, and verify that the inbound rule being created has the following default values and then Choose **Review and Launch**

i] Type: SSH

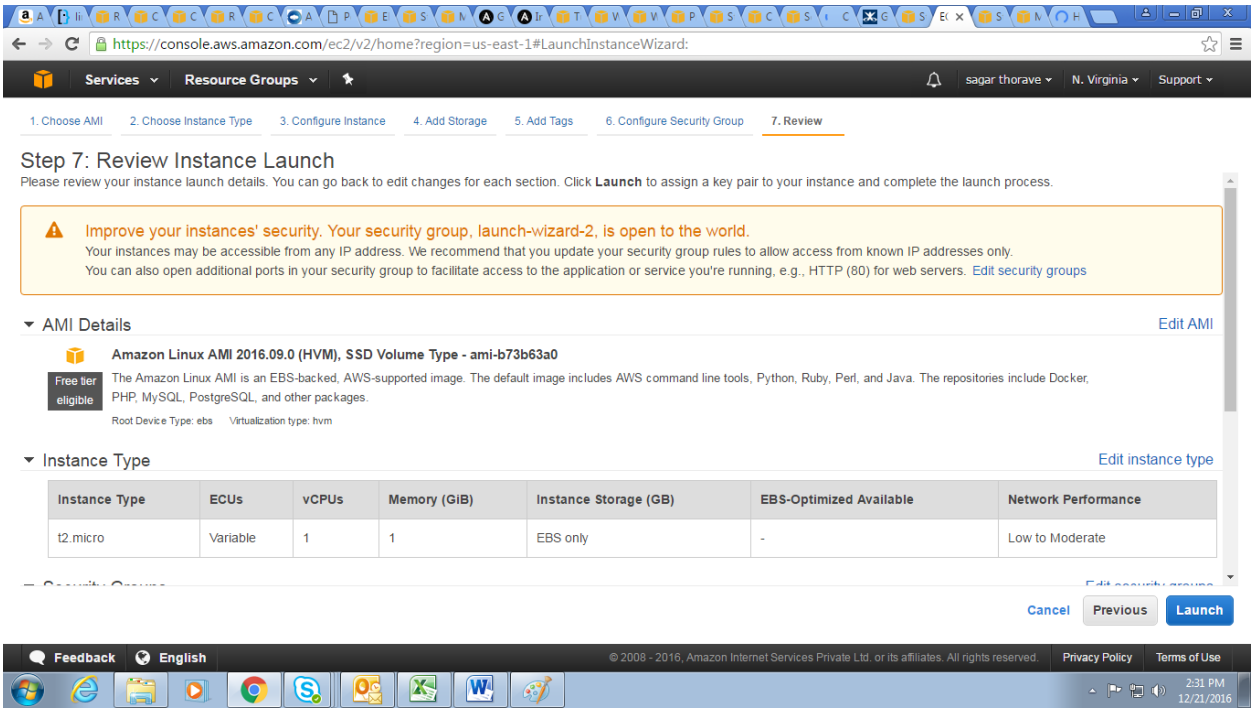
ii] Protocol: TCP

iii] Port Range: 22

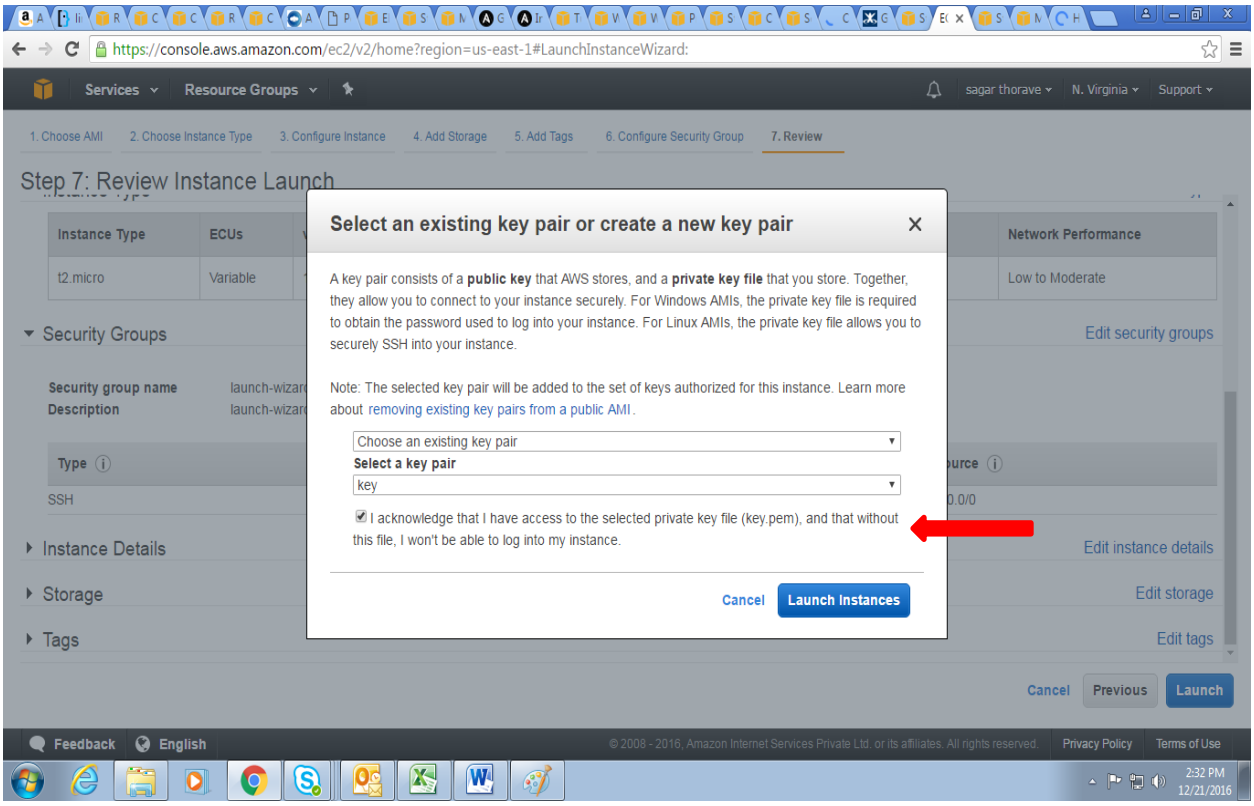
iv] Source: Anywhere 0.0.0.0/0



11 In Review Instance Launch, choose **Launch**

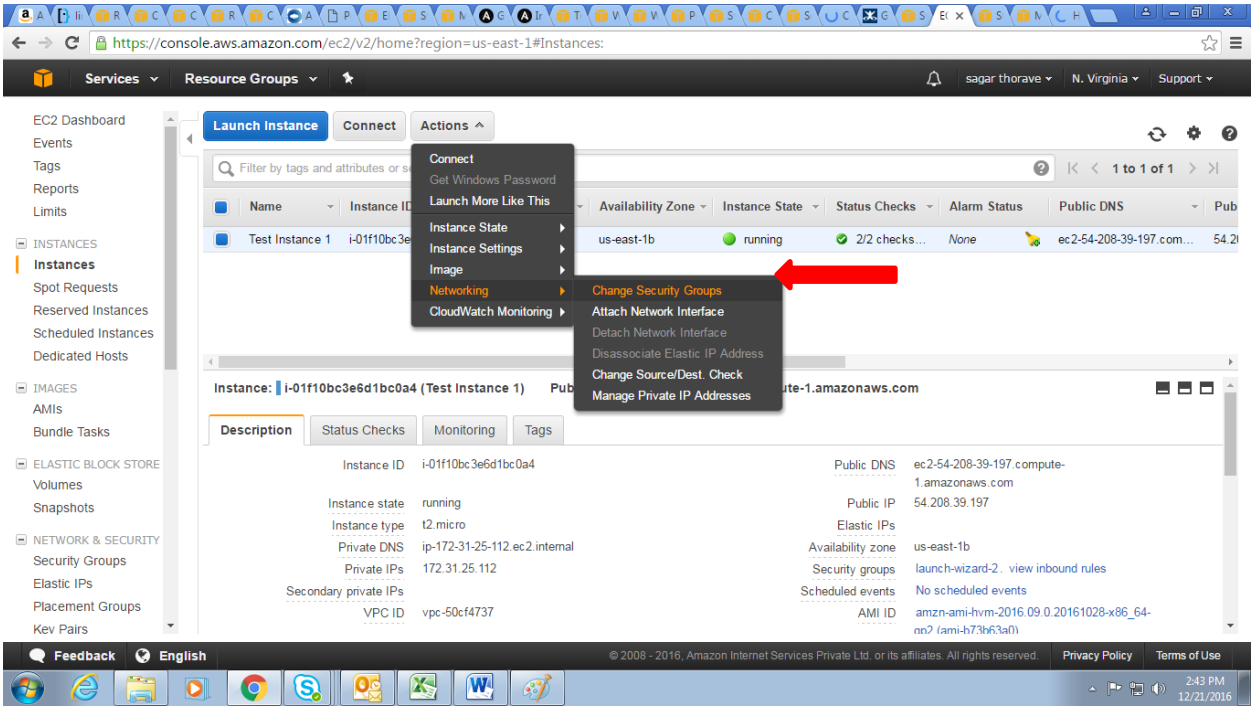


12 Here the Key you want to use for authentication and choose **Launch Instances**

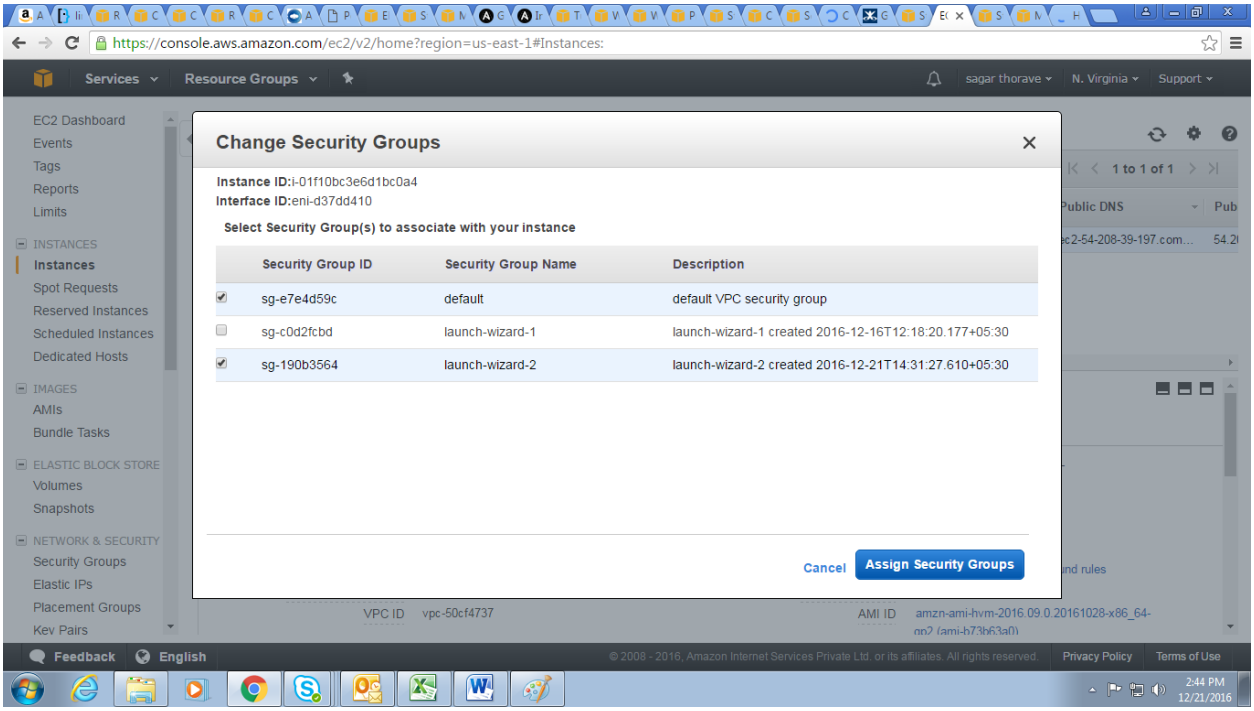


13 Choose the name of the instance you just created from the list, and then choose **Actions.**

In **Actions** Select **Networking** and then choose **Change Security Groups**

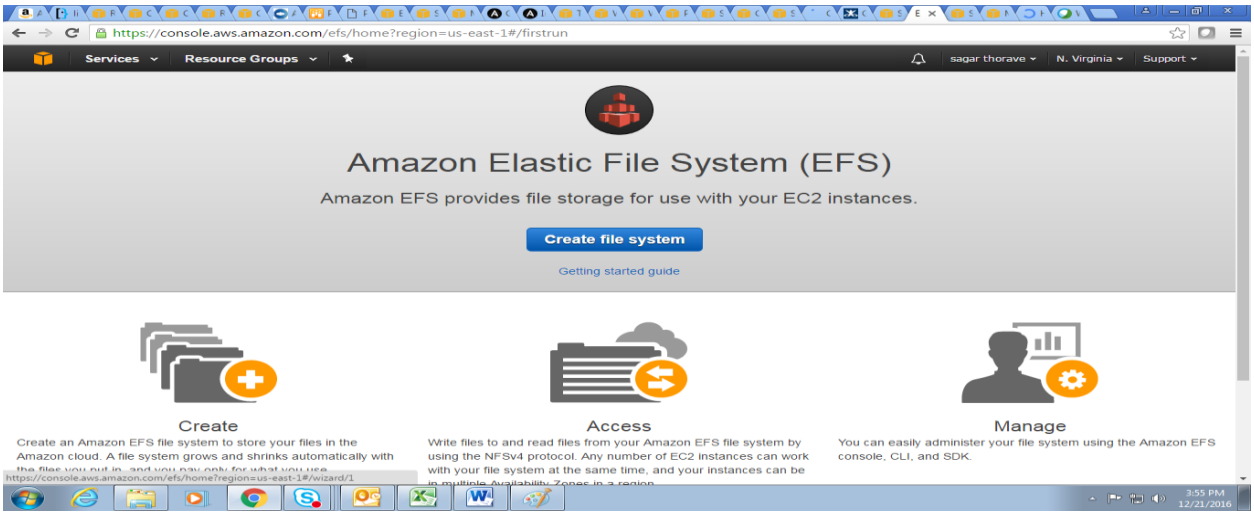


Select the check box next to the security group with the description **default VPC security group** And Choose **Assign Security Groups** .Make a note of the values listed next to VPC ID and Public DNS. You'll need those values later in this exercise.



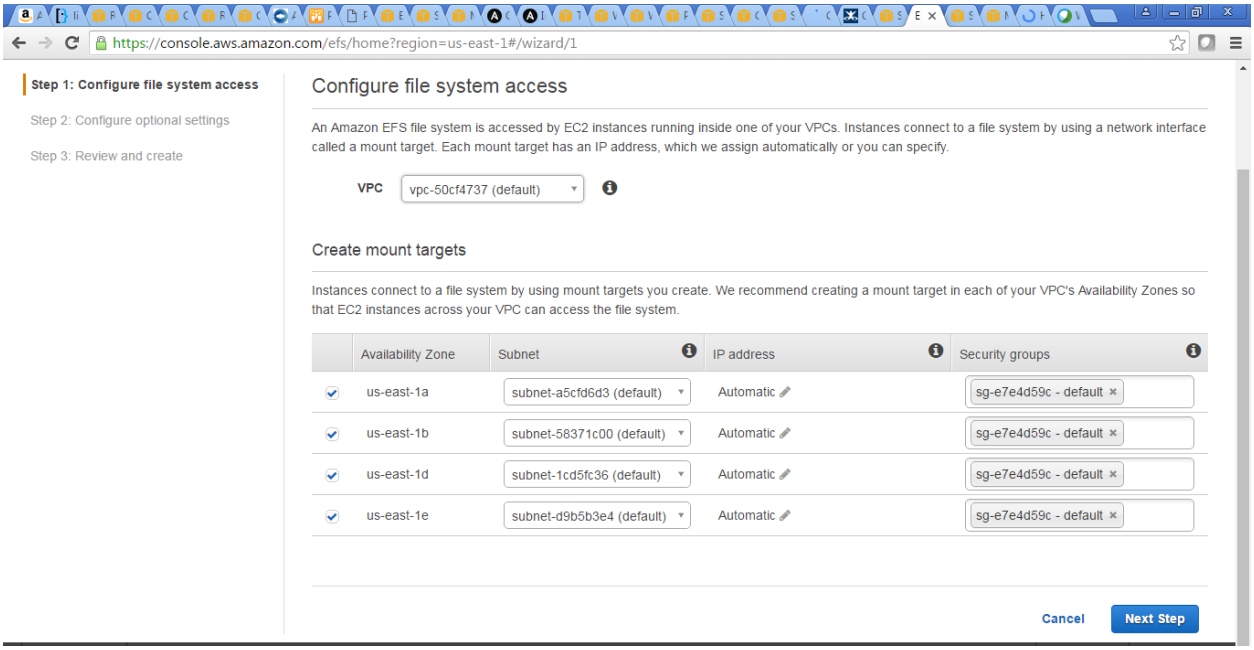
10.3.2 Create Amazon EFS File System

1. Open the Amazon EFS console and Choose **Create file system**

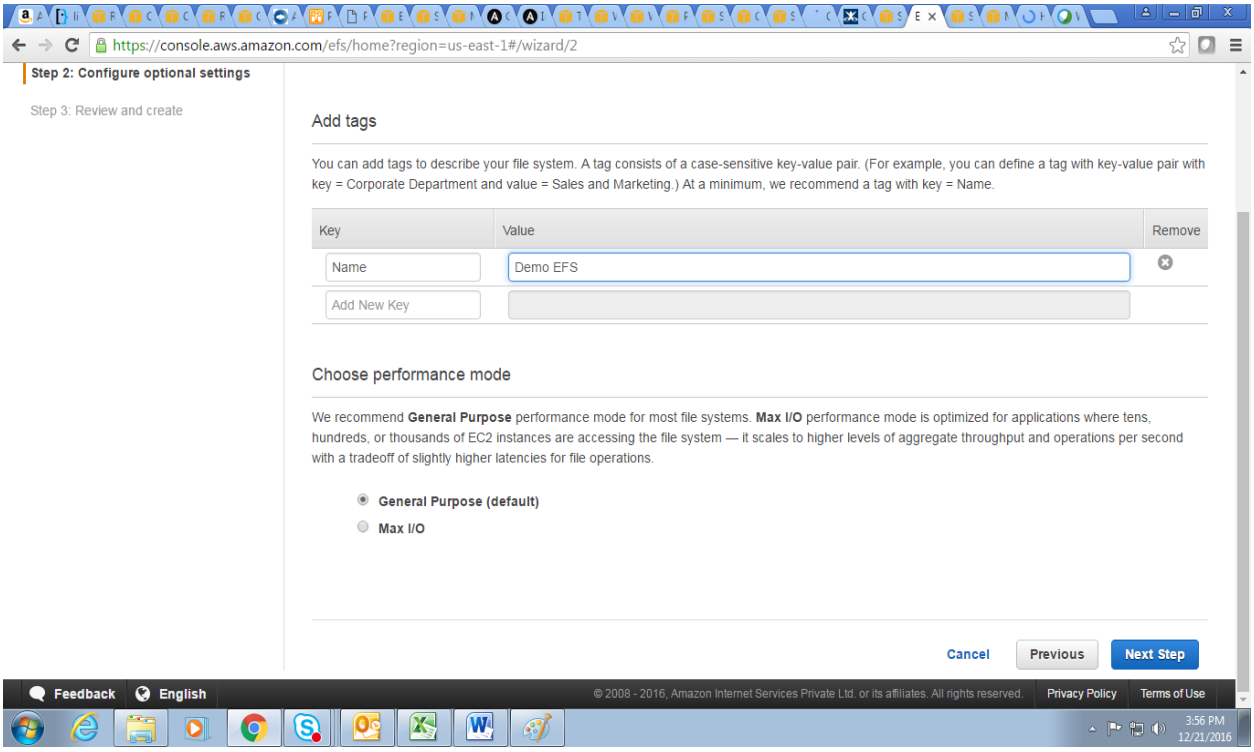


2. Choose your default vpc from the **VPC** list. Select the check boxes for all of the Availability Zones. Make sure that they all have the default subnets, automatic IP

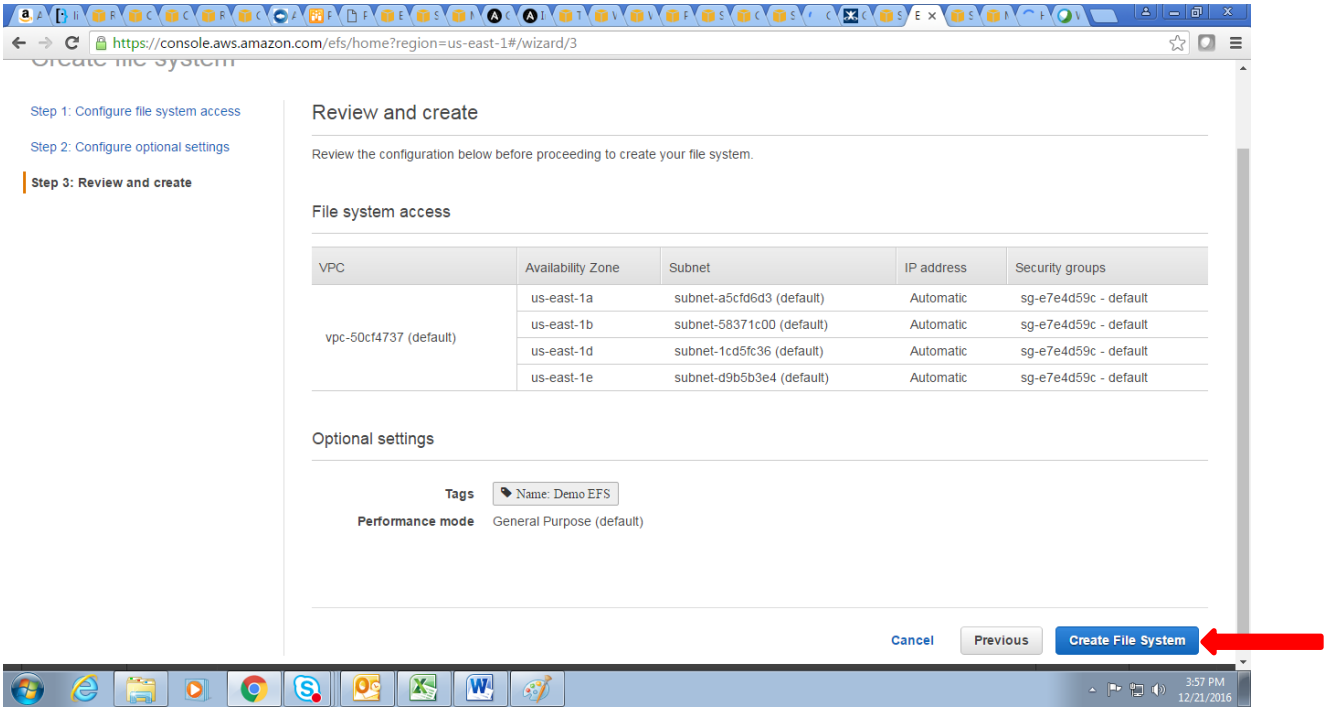
addresses, and the default security groups chosen. These are your mount targets. And then choose **Next Step**



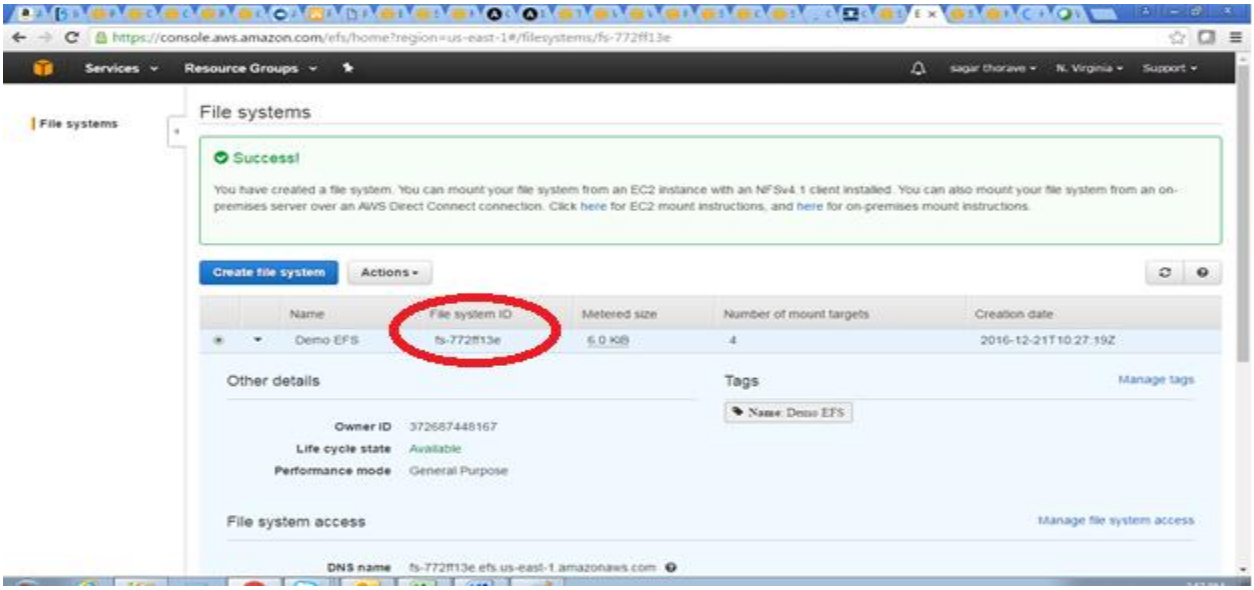
3. Name your file system, keep **general purpose** selected as your default performance mode, and choose **Next Step**



4. Choose **Create File System**

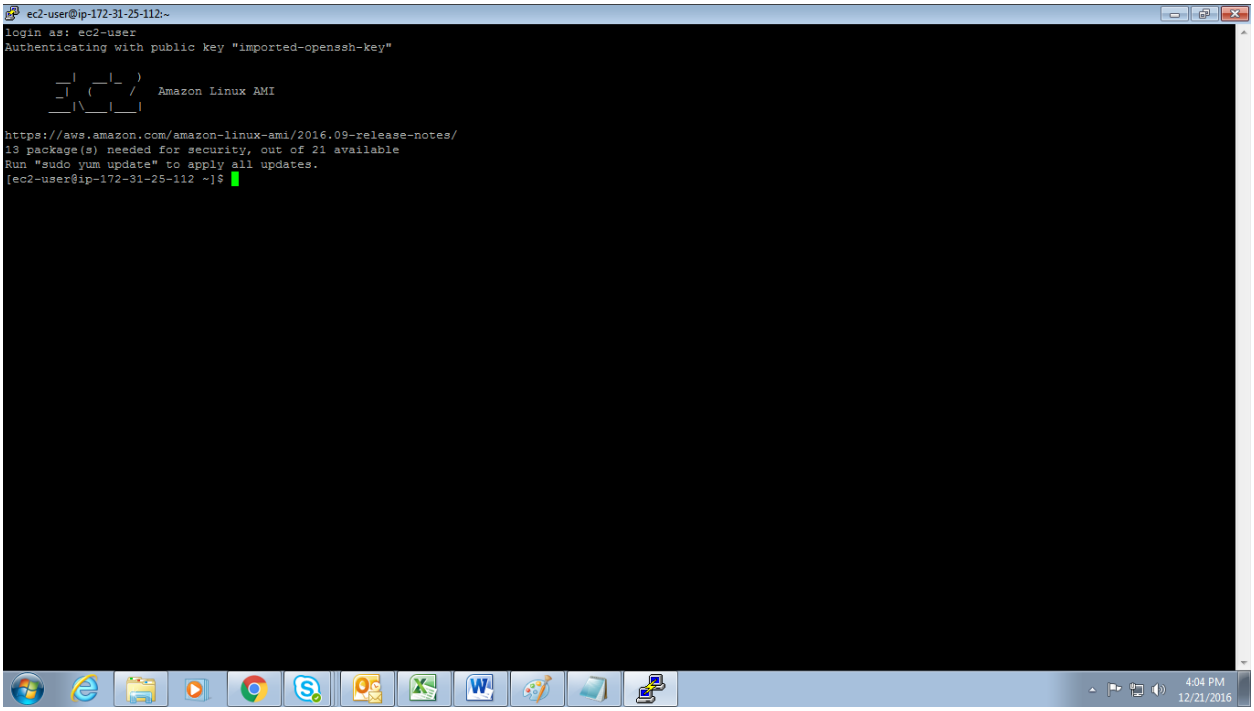


5. Choose your file system from the list and make a note of the **File system ID** value. You'll need this value for the next step.



10.3.3 Connect to Your Amazon EC2 Instance and Mount the Amazon EFS File System

Connect to your Amazon EC2 instance using ssh with the help of putty, the below screen should be displayed



1. After you've connected, install the NFS client in my case it is preinstalled so it is showing Nothing to do

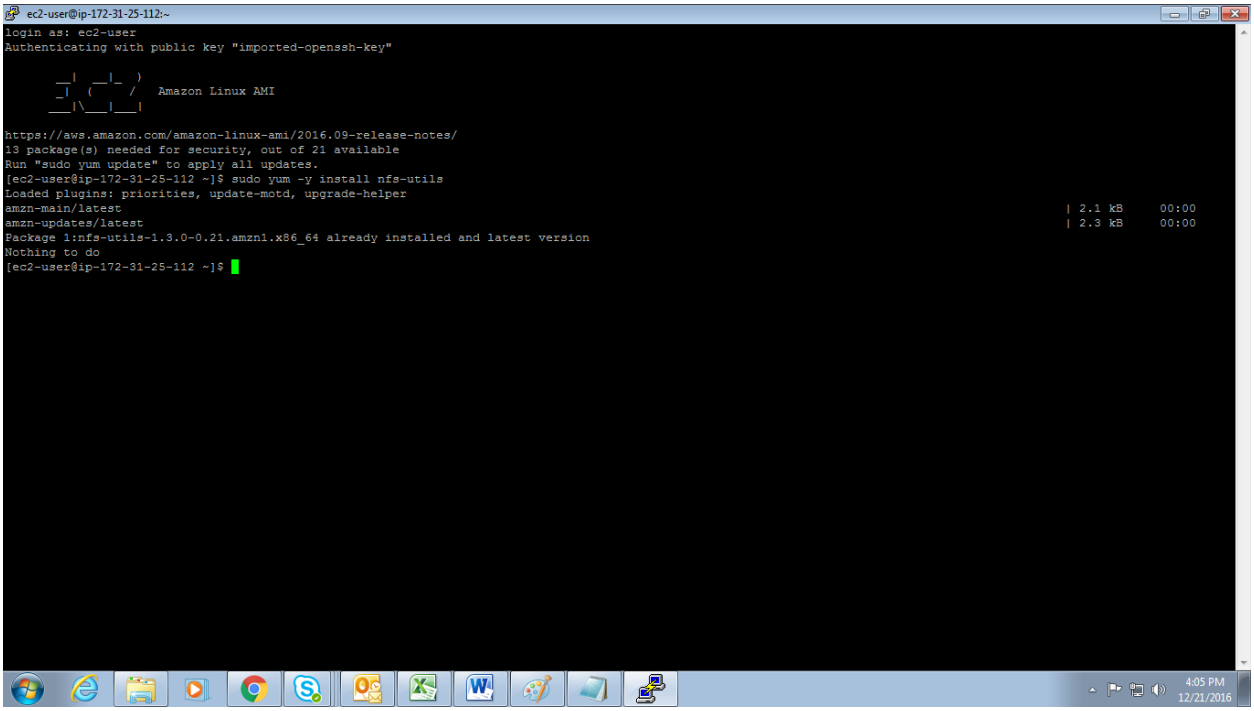
   If you're using an Amazon Linux AMI or RedHat Linux AMI, install the NFS client with the following command

   ```
   $ sudo yum -y install nfs-utils
   ```

   If you're using an Ubuntu AMI, install the NFS client with the following command

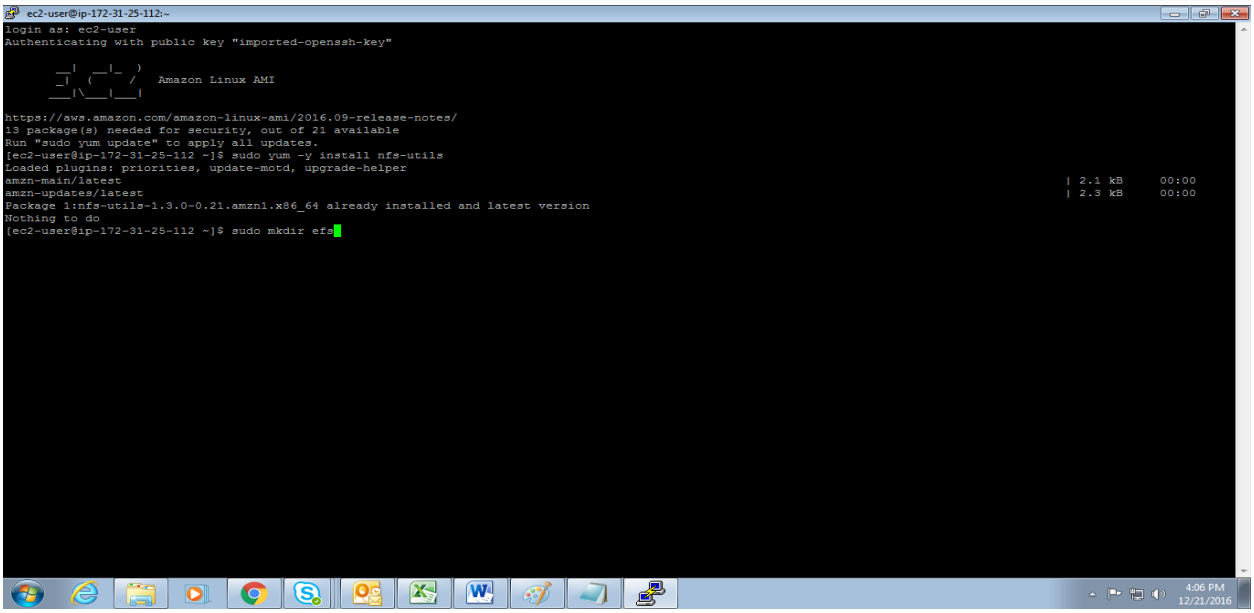
   ```
   $ sudo apt-get -y install nfs-common
   ```

   .



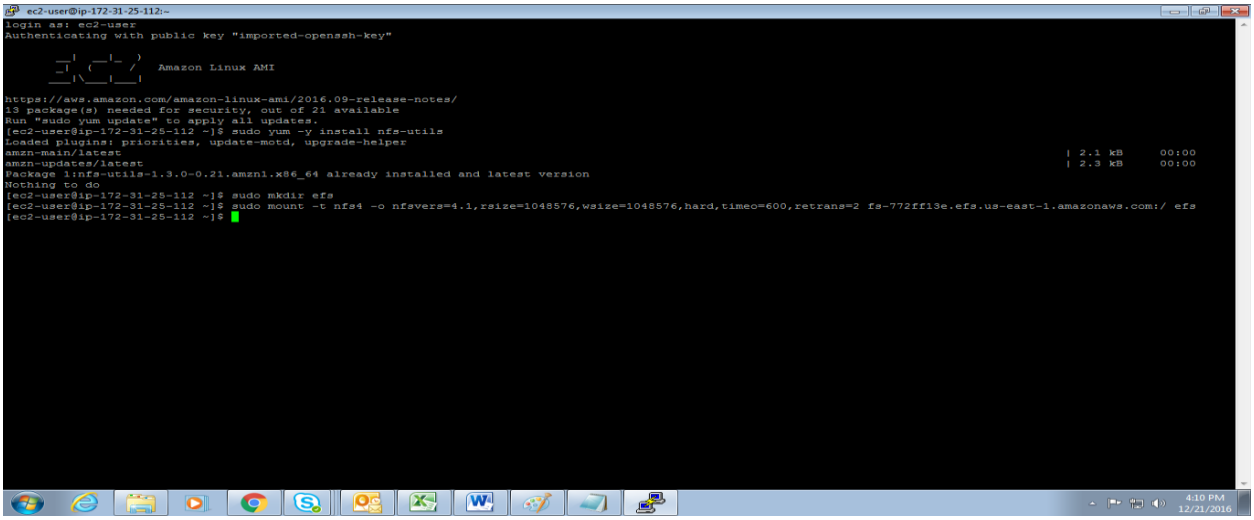
2. Make a directory for the mount point with the following command

```
$ sudo mkdir efs
```



3. Mount the Amazon EFS file system to the directory that you created. Use the following command and replace the file-system-id and aws-region placeholders with your File System ID value and AWS Region, respectively.

sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 file-system-id.efs.aws-region.amazonaws.com:/ efs

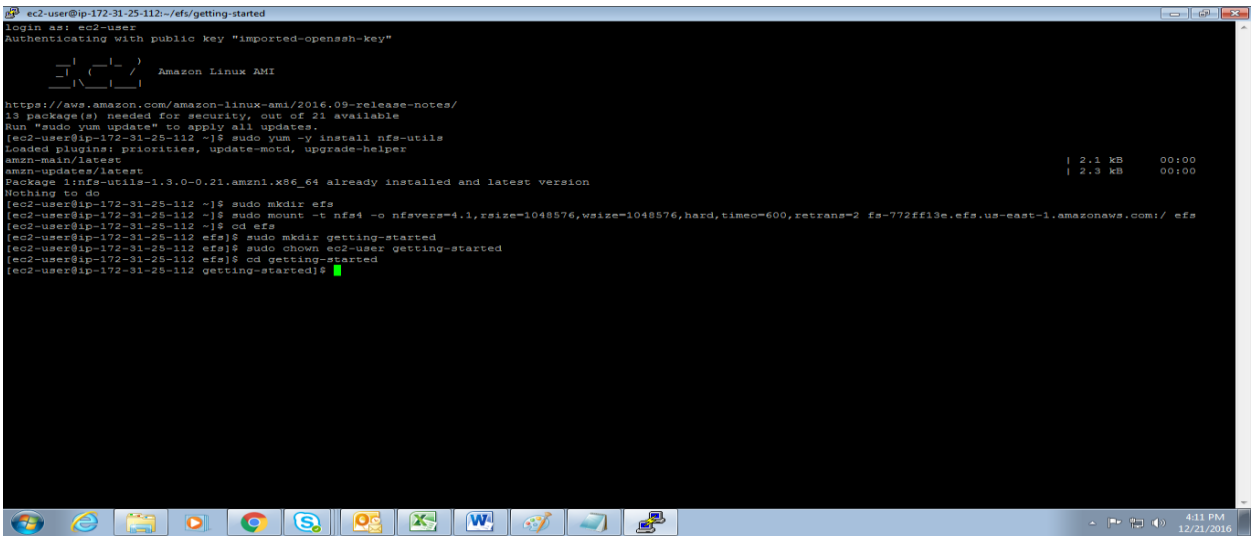


4. Change directories to the new directory that you created with the following command

```
$ cd efs
```

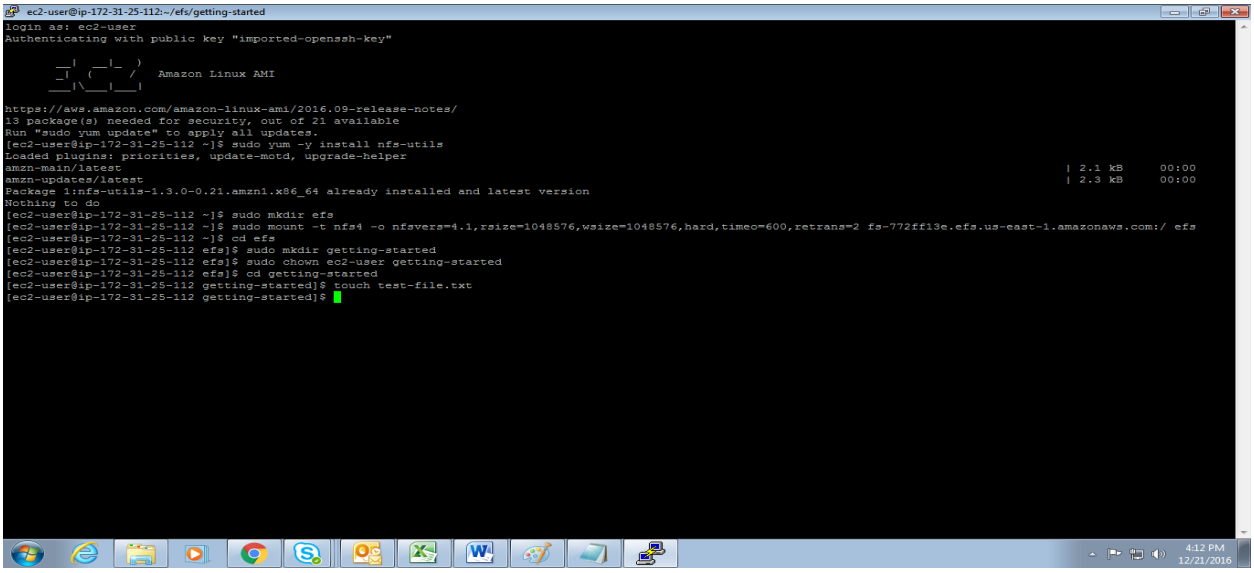
Make a subdirectory and change the ownership of that subdirectory to your EC2 instance user. Then, navigate to that new directory with the following commands

```
$ sudo mkdir getting-started
$ sudo chown ec2-user getting-started
$ cd getting-started
```



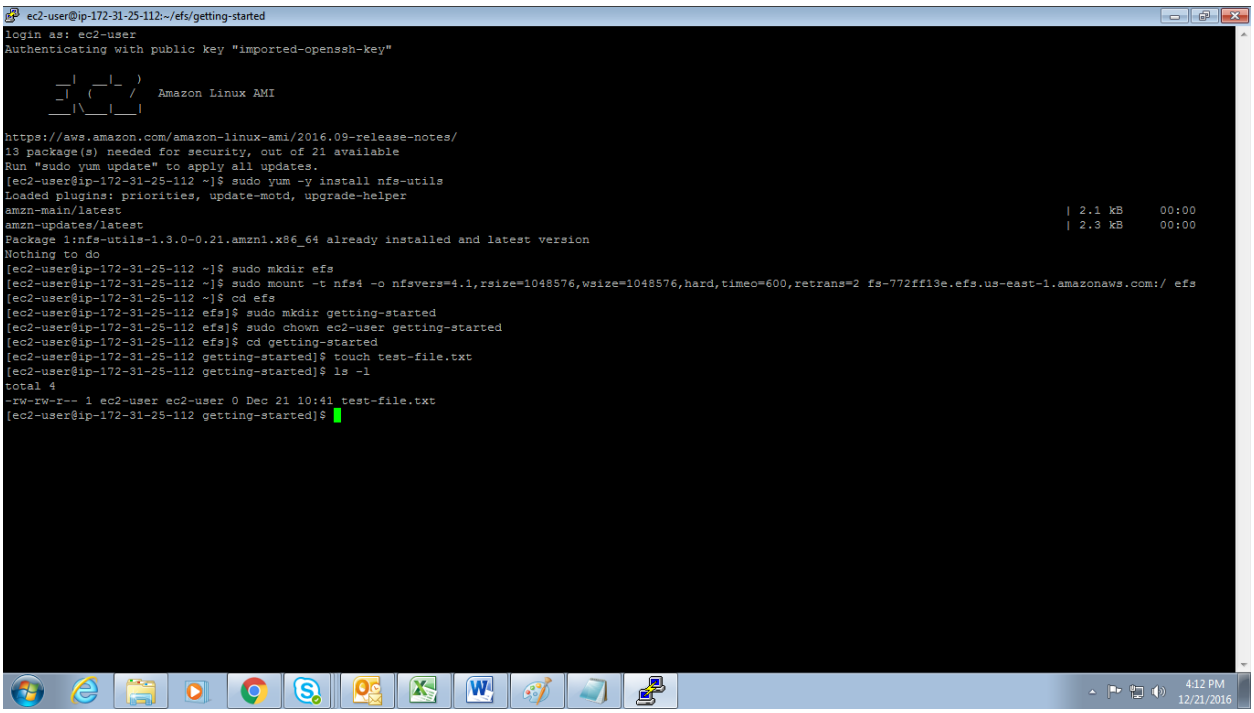
5. Create a text file with the following command

```
$ touch test-file.txt
```



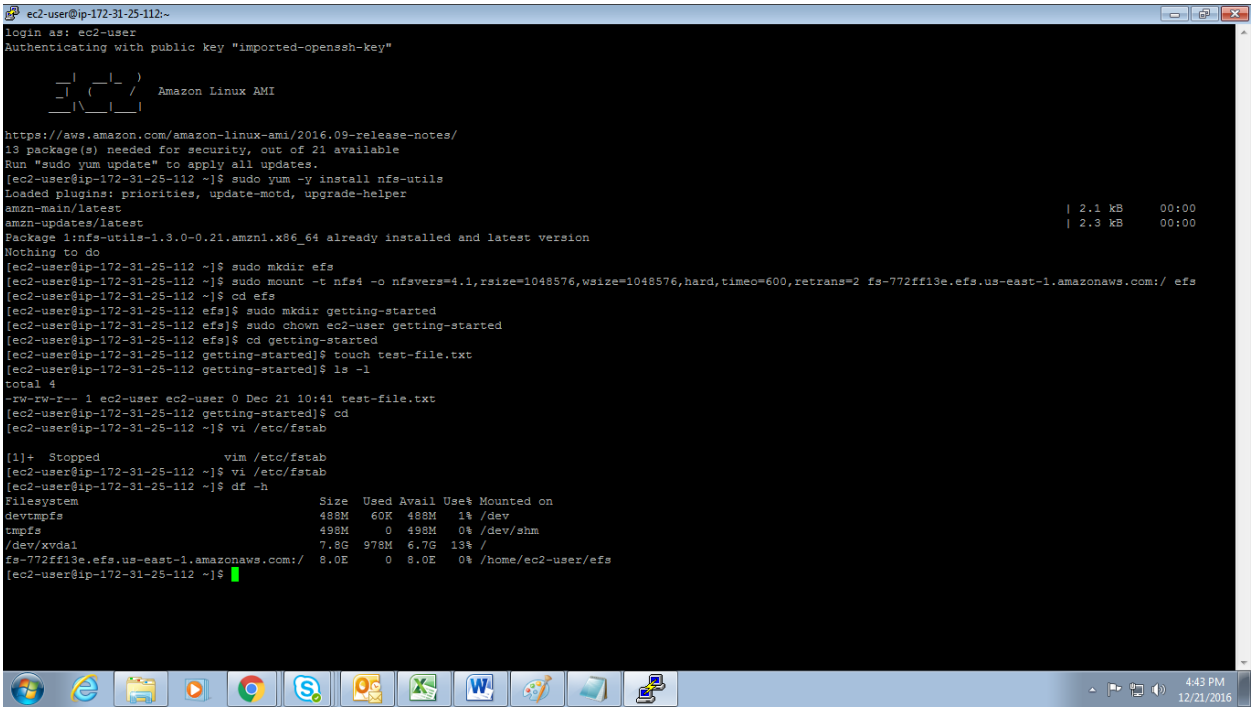
6. List the directory contents with the following command.

```
$ ls -al
```



Here you can see that the file test-file.txt has been created, To check whether the efs has been mounted or not run the following command

$ df -h



Here you can see that we have successfully mounted the Elastic File System.

11. CLOUD FORMATION

AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.

11.1 Objective

To Use AWS CloudFormation Designer to update a stack

11.2 Procedure

11.2.1 Create the Initial Stack

For the purposes of this example, we'll use the AWS Management Console to create an initial stack from the sample template.

Warning

Completing this procedure will deploy live AWS services. You will be charged the standard usage rates as long as these services are running.

To create the stack from the AWS Management Console

Copy the previous template and save it locally on your system as a text file. Note the location because you'll need to use the file in a subsequent step.

Log in to the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation .

Click Create New Stack.

In the Create New Stack wizard, on the Select Template screen, type UpdateTutorial in the Name field. On the same page, select Upload a template to Amazon S3 and browse to the file that you downloaded in the first step, and then click Next.

On the Specify Parameters screen, in the Instance Type box, type t1.micro. Then click Next.

On the Options screen, click Next.

On the Review screen, verify that all the settings are as you want them, and then click Create.

After the status of your stack is CREATE_COMPLETE, the output tab will display the URL of your website. If you click the value of the WebsiteURL output, you will see your new PHP application working.

11.2.2 Update the Application

Now that we have deployed the stack, let's update the application. We'll make a simple change to the text that is printed out by the application. To do so, we'll add an echo command to the index.php file as shown in this template snippet:

Copy

```
"WebServerInstance": {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "AWS::CloudFormation::Init" : {
          "config" : {
```

```
                :


            "files" : {


              "/var/www/html/index.php" : {
                "content" : { "Fn::Join" : ["", [
                  "<?php\n",
                  "echo '<h1>AWS CloudFormation sample PHP
application</h1>';\n",
                  "echo 'Updated version via UpdateStack';\n ",
                  "?>\n"
                ]]},
                "mode"    : "000644",
                "owner"   : "apache",
                "group"   : "apache"
              },


              :


      }
    },
```

Use a text editor to manually edit the template file that you saved locally.

Now, we'll update the stack.

To update the stack from the AWS Management Console

Log in to the AWS CloudFormation console, at: https://console.aws.amazon.com/cloudformation.

On the AWS CloudFormation dashboard, click the stack you created previously, and then click Update Stack.

In the Update Stack wizard, on the Select Template screen, select Upload a template to Amazon S3, select the modified template, and then click Next.

On the Options screen, click Next.

Click Next because the stack doesn't have a stack policy. All resources can be updated without an overriding policy.

On the Review screen, verify that all the settings are as you want them, and then click Update.

If you update the stack from the AWS Management Console, you will notice that the parameters that were used to create the initial stack are prepopulated on the **Parameters**page of the **Update Stack** wizard. If you use the `aws cloudformation update-stack`command, be sure to type in the same values for the parameters that you used originally to create the stack.

When your stack is in the UPDATE_COMPLETE state, you can click the WebsiteURL output value again to verify that the changes to your application have taken effect. By default, the cfn-hup daemon runs every 15 minutes, so it may take up to 15 minutes for the application to change once the stack has been updated.

To see the set of resources that were updated, go to the AWS CloudFormation console. On the **Events** tab, look at the stack events. In this particular case, the metadata for the Amazon EC2 instance WebServerInstance was updated, which caused AWS CloudFormation to also reevaluate the other resources (`WebServerSecurityGroup`) to ensure that there were no other changes. None of the other stack resources were modified. AWS CloudFormation will update only those resources in the stack that are affected by any changes to the stack. Such changes can be direct, such as property or metadata changes, or they can be due to dependencies or data flows through Ref, GetAtt, or other intrinsic template functions.

This simple update illustrates the process; however, you can make much more complex changes to the files and packages that are deployed to your Amazon EC2 instances. For example, you might decide that you need to add MySQL to the instance, along with PHP support for MySQL. To do so, simply add the

additional packages and files along with any additional services to the configuration and then update the stack to deploy the changes. In the following template snippet, the changes are highlighted in red:

Copy

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "Comment" : "Install a simple PHP application",
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd"             : [],
            "php"               : [],
            "php-mysql"         : [],
            "mysql-server"      : [],
            "mysql-libs"        : [],
            "mysql"             : []
          }
        },

        :

        "services" : {
          "sysvinit" : {
```

```
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true"
},
                "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
                    "files" : ["/etc/cfn/cfn-hup.conf",
"/etc/cfn/hooks.d/cfn-auto-reloader.conf"]},
                "mysqld"   : { "enabled" : "true", "ensureRunning" : "true" }
              }
            }
          }
        }
      },

      "Properties": {
           :
      }
    }
```

You can update the CloudFormation metadata to update to new versions of the packages used by the application. In the previous examples, the version property for each package is empty, indicating that cfn-init should install the latest version of the package.

Copy

```
     "packages" : {
       "yum" : {
         "httpd"             : [],
         "php"               : []
```

```
    }
```

You can optionally specify a version string for a package. If you change the version string in subsequent update stack calls, the new version of the package will be deployed. Here's an example of using version numbers for RubyGems packages. Any package that supports versioning can have specific versions.

Copy

```
"packages" : {
  "rubygems" : {
    "mysql"           : [],
    "rubygems-update" : ["1.6.2"],
    "rake"            : ["0.8.7"],
    "rails"           : ["2.3.11"]
  }
  }
```

11.2.3 Updating Auto Scaling Groups

If you are using Auto Scaling groups in your template, as opposed to Amazon EC2 instance resources, updating the application will work in exactly the same way; however, AWS CloudFormation does not provide any synchronization or serialization across the Amazon EC2 instances in an Auto Scaling group. The cfn-hup daemon on each host will run independently and update the application on its own schedule. When you use cfn-hup to update the on-instance configuration, each instance will run the cfn-hup hooks on its own schedule; there is no coordination between the instances in the stack. You should consider the following:

If the cfn-hup changes run on all Amazon EC2 instances in the Auto Scaling group at the same time, your service might be unavailable during the update.

If the cfn-hup changes run at different times, old and new versions of the software may be running at the same.

To avoid these issues, consider forcing a rolling update on your instances in the Auto Scaling group. For more information, see UpdatePolicy.

11.2.4 Changing Resource Properties

With AWS CloudFormation, you can change the properties of an existing resource in the stack. The following sections describe various updates that solve specific problems; however, any property of any resource that supports updating in the stack can be modified as necessary.

11.2.5 Update the Instance Type

The stack we have built so far uses a t1.micro Amazon EC2 instance. Let's suppose that your newly created website is getting more traffic than a t1.micro instance can handle, and now you want to move to an m1.small Amazon EC2 instance type. If the architecture of the instance type changes, the instance will be created with a different AMI. If you check out the mappings in the template, you will see that both the t1.micro and m1.small are the same architectures and use the same Amazon Linux AMIs.

Copy

```
"Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "PV64"   },
      "t2.micro"    : { "Arch" : "HVM64"  },
      "t2.small"    : { "Arch" : "HVM64"  },
      "t2.medium"   : { "Arch" : "HVM64"  },
      "m1.small"    : { "Arch" : "PV64"   },
      "m1.medium"   : { "Arch" : "PV64"   },
      "m1.large"    : { "Arch" : "PV64"   },
      "m1.xlarge"   : { "Arch" : "PV64"   },
      "m2.xlarge"   : { "Arch" : "PV64"   },
      "m2.2xlarge"  : { "Arch" : "PV64"   },
      "m2.4xlarge"  : { "Arch" : "PV64"   },
      "m3.medium"   : { "Arch" : "HVM64"  },
      "m3.large"    : { "Arch" : "HVM64"  },
      "m3.xlarge"   : { "Arch" : "HVM64"  },
      "m3.2xlarge"  : { "Arch" : "HVM64"  },
      "c1.medium"   : { "Arch" : "PV64"   },
      "c1.xlarge"   : { "Arch" : "PV64"   },
      "c3.large"    : { "Arch" : "HVM64"  },
      "c3.xlarge"   : { "Arch" : "HVM64"  },
      "c3.2xlarge"  : { "Arch" : "HVM64"  },
      "c3.4xlarge"  : { "Arch" : "HVM64"  },
      "c3.8xlarge"  : { "Arch" : "HVM64"  },
```

```
      "g2.2xlarge"  : { "Arch" : "HVMG2"  },
      "r3.large"    : { "Arch" : "HVM64"  },
      "r3.xlarge"   : { "Arch" : "HVM64"  },
      "r3.2xlarge"  : { "Arch" : "HVM64"  },
      "r3.4xlarge"  : { "Arch" : "HVM64"  },
      "r3.8xlarge"  : { "Arch" : "HVM64"  },
      "i2.xlarge"   : { "Arch" : "HVM64"  },
      "i2.2xlarge"  : { "Arch" : "HVM64"  },
      "i2.4xlarge"  : { "Arch" : "HVM64"  },
      "i2.8xlarge"  : { "Arch" : "HVM64"  },
      "hi1.4xlarge" : { "Arch" : "HVM64"  },
      "hs1.8xlarge" : { "Arch" : "HVM64"  },
      "cr1.8xlarge" : { "Arch" : "HVM64"  },
      "cc2.8xlarge" : { "Arch" : "HVM64"  }
    },

    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "PV64" : "ami-50842d38", "HVM64" : "ami-08842d60",
"HVMG2" : "ami-3a329952"  },
      "us-west-2"      : { "PV64" : "ami-af86c69f", "HVM64" : "ami-8786c6b7",
"HVMG2" : "ami-47296a77"  },
      "us-west-1"      : { "PV64" : "ami-c7a8a182", "HVM64" : "ami-cfa8a18a",
"HVMG2" : "ami-331b1376"  },
      "eu-west-1"      : { "PV64" : "ami-aa8f28dd", "HVM64" : "ami-748e2903",
"HVMG2" : "ami-00913777"  },
      "ap-southeast-1" : { "PV64" : "ami-20e1c572", "HVM64" : "ami-d6e1c584",
"HVMG2" : "ami-fabe9aa8"  },
      "ap-northeast-1" : { "PV64" : "ami-21072820", "HVM64" : "ami-35072834",
"HVMG2" : "ami-5dd1ff5c"  },
      "ap-southeast-2" : { "PV64" : "ami-8b4724b1", "HVM64" : "ami-fd4724c7",
"HVMG2" : "ami-e98ae9d3"  },
      "sa-east-1"      : { "PV64" : "ami-9d6cc680", "HVM64" : "ami-956cc688",
"HVMG2" : "NOT_SUPPORTED" },
      "cn-north-1"     : { "PV64" : "ami-a857c591", "HVM64" : "ami-ac57c595",
"HVMG2" : "NOT_SUPPORTED" },
      "eu-central-1"   : { "PV64" : "ami-a03503bd", "HVM64" : "ami-b43503a9",
"HVMG2" : "ami-b03503ad"  }
    }
```

```
}
```

Let's use the template that we modified in the previous section to change the instance type. Because InstanceType was an input parameter to the template, we don't need to modify the template; we can simply change the value of the parameter in the Stack Update wizard, on the Specify Parameters page.

To update the stack from the AWS Management Console

Log in to the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.

On the AWS CloudFormation dashboard, click the stack you created previously, and then click Update Stack.

In the Update Stack wizard, on the Select Template screen, select Use current template, and then click Next.

The Specify Details page appears with the parameters that were used to create the initial stack are pre-populated in the Specify Parameters section.

Change the value of the InstanceType text box from t1.micro to t2.small. Then, click Next.

On the Options screen, click Next.

Click Next because the stack doesn't have a stack policy. All resources can be updated without an overriding policy.

On the Review screen, verify that all the settings are as you want them, and then click Update.

You can dynamically change the instance type of an EBS-backed Amazon EC2 instance by starting and stopping the instance. AWS CloudFormation tries to optimize the change by updating the instance type and restarting the instance, so the instance ID does not change. When the instance is restarted, however, the public IP address of the instance does change. To ensure that the Elastic IP address is bound correctly after the change, AWS CloudFormation will also update the Elastic IP address. You can see the changes in the AWS CloudFormation console on the Events tab.

To check the instance type from the AWS Management Console, open the Amazon EC2 console, and locate your instance there.

11.2.6 Update the AMI on an Amazon EC2 instance

Now let's look at how we might change the Amazon Machine Image (AMI) running on the instance. We will trigger the AMI change by updating the stack to use a new Amazon EC2 instance type, such as t2.medium, which is an HVM64 instance type.

11.2.6 Update the Amazon EC2 Launch Configuration for an Auto Scaling Group

If you are using Auto Scaling groups rather than Amazon EC2 instances, the process of updating the running instances is a little different. With Auto Scaling resources, the configuration of the Amazon EC2 instances, such as the instance type or the AMI ID is encapsulated in the Auto Scaling launch configuration. You can make changes to the launch configuration in the same way as we made changes to the Amazon EC2 instance resources in the previous sections. However, changing the launch configuration does not impact any of the running Amazon EC2 instances in the Auto Scaling group. An updated launch configuration applies only to new instances that are created after the update.

If you want to propagate the change to your launch configuration across all the instances in your Auto Scaling group, you can use an update attribute. For more information, see UpdatePolicy.

11.2.7 Adding Resource Properties

So far, we've looked at changing existing properties of a resource in a template. You can also add properties that were not originally specified in the template. To illustrate that, we'll add an Amazon EC2 key pair to an existing EC2 instance and then open up port 22 in the Amazon EC2 Security Group so that you can use Secure Shell (SSH) to access the instance.

11.2.8 Add a Key Pair to an Instance

To add SSH access to an existing Amazon EC2 instance

Add two additional parameters to the template to pass in the name of an existing Amazon EC2 key pair and SSH location.

Copy

```
"Parameters" : {

  "KeyName" : {
```

```
      "Description" : "Name of an existing Amazon EC2 key pair for SSH
access",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
    "SSHLocation" : {
      "Description" : " The IP address range that can be used to SSH to
the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern":
"(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the
form x.x.x.x/x."
    }
    :
  },
```

Add the KeyName property to the Amazon EC2 instance.

Copy

```
          "WebServerInstance": {
          "Type" : "AWS::EC2::Instance",
          :
          "Properties": {
          :
          "KeyName" : { "Ref" : "KeyName" },
          :
          }
          },
```

Add port 22 and the SSH location to the ingress rules for the Amazon EC2 security group.

Copy

```
    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP and SSH",
```

```
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22",
"CidrIp" : { "Ref" : "SSHLocation"}},
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80",
"CidrIp" :  "0.0.0.0/0"}
        ]
      }
    },
```

Update the stack, either from the AWS Management Console as explained in Update the Application or by using the AWS command aws cloudformation update-stack.

11.2.9 Change the Stack's Resources

Since application needs can change over time, AWS CloudFormation allows you to change the set of resources that make up the stack. To demonstrate, we'll take the single instance application from Adding Resource Properties and convert it to an auto-scaled, load-balanced application by updating the stack.

This will create a simple, single instance PHP application using an Elastic IP address. We'll now turn the application into a highly available, auto-scaled, load balanced application by changing its resources during an update.

Add an Elastic Load Balancer resource.

Copy

```
    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "CrossZone" : "true",
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "LBCookieStickinessPolicy" : [ {
          "PolicyName" : "CookieBasedPolicy",
          "CookieExpirationPeriod" : "30"
        } ],
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : "80",
```

```
          "Protocol" : "HTTP",
          "PolicyNames" : [ "CookieBasedPolicy" ]
        } ],
        "HealthCheck" : {
          "Target" : "HTTP:80/",
          "HealthyThreshold" : "2",
          "UnhealthyThreshold" : "5",
          "Interval" : "10",
          "Timeout" : "5"
        }
      }
    }
```

Convert the EC2 instance in the template into an Auto Scaling Launch Configuration. The properties are identical, so we only need to change the type name from:

Copy

```
"WebServerInstance": {
  "Type" : "AWS::EC2::Instance",
```

to:

Copy

```
"LaunchConfig": {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
```

For clarity in the template, we changed the name of the resource from *WebServerInstance* to *LaunchConfig*, so you'll need to update the resource name referenced by cfn-init and cfn-hup (just search for WebServerInstance and replace it with LaunchConfig, except for cfn-signal). For cfn-signal, you'll need to signal the Auto Scaling group (WebServerGroup) not the instance, as shown in the following snippet:

Copy

```
              "# Signal the status from cfn-init\n",
              "/opt/aws/bin/cfn-signal -e $? ",
```

```
            "         --stack ", { "Ref" : "AWS::StackName" },
            "         --resource WebServerGroup ",
            "         --region ", { "Ref" : "AWS::Region" }, "\n"
```

1. Add an Auto Scaling Group resource.

Copy

```
"WebServerGroup" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
    "MinSize" : "1",
    "DesiredCapacity" : "1",
    "MaxSize" : "5",
    "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
  },
  "CreationPolicy" : {
    "ResourceSignal" : {
      "Timeout" : "PT15M"
    }
  },
  "UpdatePolicy": {
    "AutoScalingRollingUpdate": {
      "MinInstancesInService": "1",
      "MaxBatchSize": "1",
      "PauseTime" : "PT15M",
      "WaitOnResourceSignals": "true"
    }
  }
}
```

Update the Security Group definition to lock down the traffic to the instances from the load balancer.

Copy

```
"WebServerSecurityGroup" : {
  "Type" : "AWS::EC2::SecurityGroup",
  "Properties" : {
```

```
        "GroupDescription" : "Enable HTTP access via port 80 locked
down to the ELB and SSH access",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80",
"SourceSecurityGroupOwnerId" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
"SourceSecurityGroup.OwnerAlias"]},
"SourceSecurityGroupName" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
"SourceSecurityGroup.GroupName"]}},
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22",
"CidrIp" : { "Ref" : "SSHLocation"}}
        ]
      }
    }
```

Update the Outputs to return the DNS Name of the Elastic Load Balancer as the location of the application from:

Copy

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : ["", ["http://",
      { "Fn::GetAtt" : [ "WebServerInstance", "PublicDnsName" ]}]]},
  "Description" : "Application URL"
}
```

to:

Copy

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : ["", ["http://",
      { "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ]}]]},
  "Description" : "Application URL"
}
```

For reference, the follow sample shows the complete template. If you use this template to update the stack, you will convert your simple, single instance application into a highly available, multi-AZ, auto-

scaled and load balanced application. Only the resources that need to be updated will be altered, so had there been any data stores for this application, the data would have remained intact. Now, you can use AWS CloudFormation to grow or enhance your stacks as your requirements change.

Copy

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template: Sample template that
can be used to test EC2 updates. **WARNING** This template creates an Amazon
Ec2 Instance. You will be billed for the AWS resources used if you create a
stack from this template.",

  "Parameters" : {

    "KeyName": {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access
to the instance",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription" : "must be the name of an existing EC2
KeyPair."
    },

    "SSHLocation" : {
      "Description" : " The IP address range that can be used to SSH to the
EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern":
"(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form
x.x.x.x/x."
    },

    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
```

```
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro", "t2.micro", "t2.small", "t2.medium",
"m1.small", "m1.medium", "m1.large", "m1.xlarge", "m2.xlarge",
"m2.2xlarge", "m2.4xlarge", "m3.medium", "m3.large", "m3.xlarge",
"m3.2xlarge", "c1.medium", "c1.xlarge", "c3.large", "c3.xlarge",
"c3.2xlarge",
"c3.4xlarge", "c3.8xlarge", "g2.2xlarge", "r3.large", "r3.xlarge",
"r3.2xlarge", "r3.4xlarge", "r3.8xlarge", "i2.xlarge", "i2.2xlarge",
"i2.4xlarge",
"i2.8xlarge", "hi1.4xlarge", "hs1.8xlarge", "cr1.8xlarge", "cc2.8xlarge",
"cg1.4xlarge"],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "PV64"   },
      "t2.micro"    : { "Arch" : "HVM64"  },
      "t2.small"    : { "Arch" : "HVM64"  },
      "t2.medium"   : { "Arch" : "HVM64"  },
      "m1.small"    : { "Arch" : "PV64"   },
      "m1.medium"   : { "Arch" : "PV64"   },
      "m1.large"    : { "Arch" : "PV64"   },
      "m1.xlarge"   : { "Arch" : "PV64"   },
      "m2.xlarge"   : { "Arch" : "PV64"   },
      "m2.2xlarge"  : { "Arch" : "PV64"   },
      "m2.4xlarge"  : { "Arch" : "PV64"   },
      "m3.medium"   : { "Arch" : "HVM64"  },
      "m3.large"    : { "Arch" : "HVM64"  },
      "m3.xlarge"   : { "Arch" : "HVM64"  },
      "m3.2xlarge"  : { "Arch" : "HVM64"  },
      "c1.medium"   : { "Arch" : "PV64"   },
      "c1.xlarge"   : { "Arch" : "PV64"   },
      "c3.large"    : { "Arch" : "HVM64"  },
      "c3.xlarge"   : { "Arch" : "HVM64"  },
      "c3.2xlarge"  : { "Arch" : "HVM64"  },
```

```
      "c3.4xlarge"  : { "Arch" : "HVM64"  },
      "c3.8xlarge"  : { "Arch" : "HVM64"  },
      "g2.2xlarge"  : { "Arch" : "HVMG2"  },
      "r3.large"    : { "Arch" : "HVM64"  },
      "r3.xlarge"   : { "Arch" : "HVM64"  },
      "r3.2xlarge"  : { "Arch" : "HVM64"  },
      "r3.4xlarge"  : { "Arch" : "HVM64"  },
      "r3.8xlarge"  : { "Arch" : "HVM64"  },
      "i2.xlarge"   : { "Arch" : "HVM64"  },
      "i2.2xlarge"  : { "Arch" : "HVM64"  },
      "i2.4xlarge"  : { "Arch" : "HVM64"  },
      "i2.8xlarge"  : { "Arch" : "HVM64"  },
      "hi1.4xlarge" : { "Arch" : "HVM64"  },
      "hs1.8xlarge" : { "Arch" : "HVM64"  },
      "cr1.8xlarge" : { "Arch" : "HVM64"  },
      "cc2.8xlarge" : { "Arch" : "HVM64"  }
    },

    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "PV64" : "ami-50842d38", "HVM64" : "ami-08842d60",
"HVMG2" : "ami-3a329952"  },
      "us-west-2"      : { "PV64" : "ami-af86c69f", "HVM64" : "ami-8786c6b7",
"HVMG2" : "ami-47296a77"  },
      "us-west-1"      : { "PV64" : "ami-c7a8a182", "HVM64" : "ami-cfa8a18a",
"HVMG2" : "ami-331b1376"  },
      "eu-west-1"      : { "PV64" : "ami-aa8f28dd", "HVM64" : "ami-748e2903",
"HVMG2" : "ami-00913777"  },
      "ap-southeast-1" : { "PV64" : "ami-20e1c572", "HVM64" : "ami-d6e1c584",
"HVMG2" : "ami-fabe9aa8"  },
      "ap-northeast-1" : { "PV64" : "ami-21072820", "HVM64" : "ami-35072834",
"HVMG2" : "ami-5dd1ff5c"  },
      "ap-southeast-2" : { "PV64" : "ami-8b4724b1", "HVM64" : "ami-fd4724c7",
"HVMG2" : "ami-e98ae9d3"  },
      "sa-east-1"      : { "PV64" : "ami-9d6cc680", "HVM64" : "ami-956cc688",
"HVMG2" : "NOT_SUPPORTED" },
      "cn-north-1"     : { "PV64" : "ami-a857c591", "HVM64" : "ami-ac57c595",
"HVMG2" : "NOT_SUPPORTED" },
```

```
      "eu-central-1"    : { "PV64" : "ami-a03503bd", "HVM64" : "ami-b43503a9",
"HVMG2" : "ami-b03503ad"  }
    }
  },

  "Resources" : {

    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "CrossZone" : "true",
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "LBCookieStickinessPolicy" : [ {
          "PolicyName" : "CookieBasedPolicy",
          "CookieExpirationPeriod" : "30"
        } ],
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : "80",
          "Protocol" : "HTTP",
          "PolicyNames" : [ "CookieBasedPolicy" ]
        } ],
        "HealthCheck" : {
          "Target" : "HTTP:80/",
          "HealthyThreshold" : "2",
          "UnhealthyThreshold" : "5",
          "Interval" : "10",
          "Timeout" : "5"
        }
      }
    },

    "WebServerGroup" : {
      "Type" : "AWS::AutoScaling::AutoScalingGroup",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
        "MinSize" : "1",
```

```
      "DesiredCapacity" : "1",
      "MaxSize" : "5",
      "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
    },
    "CreationPolicy" : {
      "ResourceSignal" : {
        "Timeout" : "PT15M"
      }
    },
    "UpdatePolicy": {
      "AutoScalingRollingUpdate": {
        "MinInstancesInService": "1",
        "MaxBatchSize": "1",
        "PauseTime" : "PT15M",
        "WaitOnResourceSignals": "true"
      }
    }
  },

  "LaunchConfig": {
    "Type" : "AWS::AutoScaling::LaunchConfiguration",
    "Metadata" : {
      "Comment" : "Install a simple PHP application",
      "AWS::CloudFormation::Init" : {
        "config" : {
          "packages" : {
            "yum" : {
              "httpd"             : [],
              "php"               : []
            }
          },

          "files" : {

            "/var/www/html/index.php" : {
              "content" : { "Fn::Join" : ["", [
                "<?php\n",
```

```
                  "echo '<h1>AWS CloudFormation sample PHP
application</h1>';\n",
                  "echo 'Updated version via UpdateStack';\n ",
                  "?>\n"
                ]]},
                "mode"    : "000644",
                "owner"   : "apache",
                "group"   : "apache"
              },


              "/etc/cfn/cfn-hup.conf" : {
                "content" : { "Fn::Join" : ["", [
                  "[main]\n",
                  "stack=", { "Ref" : "AWS::StackId" }, "\n",
                  "region=", { "Ref" : "AWS::Region" }, "\n"
                ]]},
                "mode"    : "000400",
                "owner"   : "root",
                "group"   : "root"
              },

              "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
                "content": { "Fn::Join" : ["", [
                  "[cfn-auto-reloader-hook]\n",
                  "triggers=post.update\n",

"path=Resources.LaunchConfig.Metadata.AWS::CloudFormation::Init\n",
                  "action=/opt/aws/bin/cfn-init -s ", { "Ref" :
"AWS::StackId" }, " -r LaunchConfig ",
                                                      " --region     ", { "Ref"
: "AWS::Region" }, "\n",
                  "runas=root\n"
                ]]}
              }
            },

            "services" : {
```

```
              "sysvinit" : {
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true"
},
                "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
                    "files" : ["/etc/cfn/cfn-hup.conf",
"/etc/cfn/hooks.d/cfn-auto-reloader.conf"]}
              }
            }
          }
        }
      },

      "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                          { "Fn::FindInMap" : [ "AWSInstanceType2Arch", {
"Ref" : "InstanceType" }, "Arch" ] } ] },
        "InstanceType"   : { "Ref" : "InstanceType" },
        "KeyName"        : { "Ref" : "KeyName" },
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
             "#!/bin/bash -xe\n",
             "yum install -y aws-cfn-bootstrap\n",

             "# Install the files and packages from the metadata\n",
             "/opt/aws/bin/cfn-init -v ",
             "         --stack ", { "Ref" : "AWS::StackName" },
             "         --resource LaunchConfig ",
             "         --region ", { "Ref" : "AWS::Region" }, "\n",

             "# Start up the cfn-hup daemon to listen for changes to the Web
Server metadata\n",
             "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-
hup'\n",

             "# Signal the status from cfn-init\n",
             "/opt/aws/bin/cfn-signal -e $? ",
             "         --stack ", { "Ref" : "AWS::StackName" },
```

```
              "         --resource WebServerGroup ",
              "         --region ", { "Ref" : "AWS::Region" }, "\n"
        ]]}}
      }
    },

    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80 locked down to
the ELB and SSH access",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80",
"SourceSecurityGroupOwnerId" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
"SourceSecurityGroup.OwnerAlias"]},"SourceSecurityGroupName" : {"Fn::GetAtt"
: ["ElasticLoadBalancer", "SourceSecurityGroup.GroupName"]}},
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp"
: { "Ref" : "SSHLocation"}}
        ]
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Description" : "Application URL",
      "Value" : { "Fn::Join" : ["", ["http://", { "Fn::GetAtt" : [
"ElasticLoadBalancer", "DNSName" ]}]] }
    }
  }
}
```

12. CLOUDWATCH MANAGEMENT

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time.

12.1 Objective

To understand the process of AWS CloudWatch.

12.2 Assumptions

- You’re having an AWS account.
- You're familiar with using the AWS CloudWatch console to launch instances.
- You have a default VPC in the region that you're using for this Getting Started exercise. If you don't have a default VPC, or if you want to mount your file system from a new VPC with new or existing security groups, you can still use this Getting Started exercise

12.3 Procedure

Step 1: Enable Billing Alerts

Step 2: Create a Billing Alarm

Step 3: Check the Alarm Status

Step 4: Edit a Billing Alarm

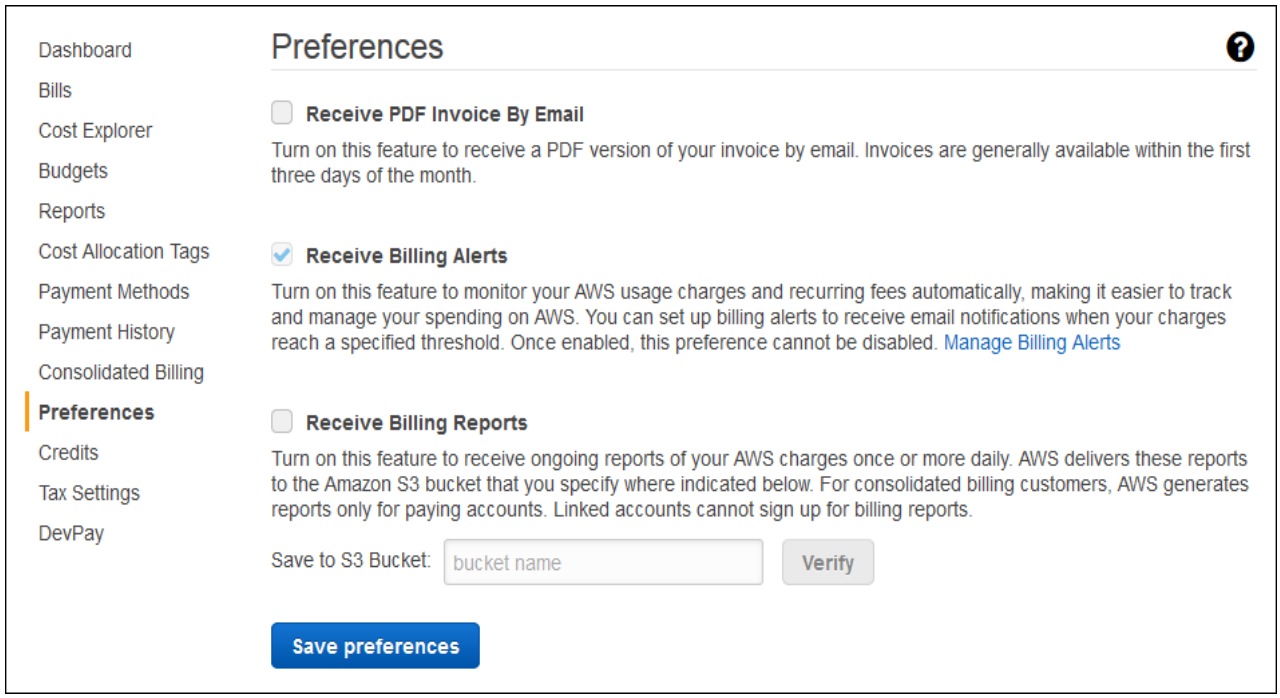
Step 5: Delete a Billing Alarm

12.3.1 Enable Billing Alerts

- Enable billing alerts, so that we can monitor estimated AWS charges and create an alarm using billing metric data.
- After you enable billing alerts for the first time, it takes about 15 minutes before you can view billing data and set billing alarms.

To enable monitoring of your estimated charges

1. Open the Billing and Cost Management console at https://console.aws.amazon.com/billing/home?#.
2. In the navigation pane, choose **Preferences**.
3. Select **Receive Billing Alerts**.



4. Choose **Save preferences**.

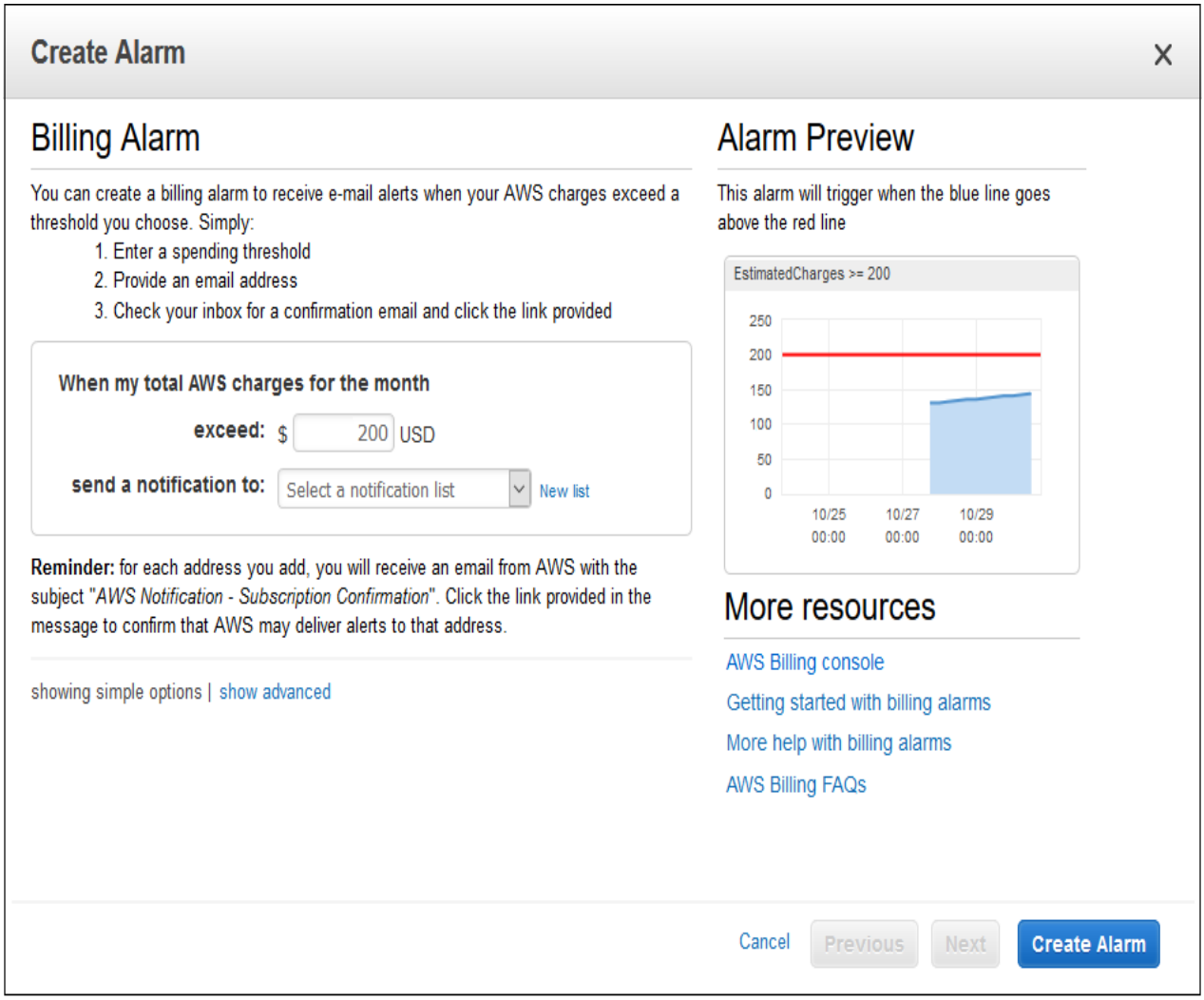
12.3.2 Create a Billing Alarm

- After enabled billing alerts, create a billing alarm.
- An alarm will sends an email message when your estimated charges for AWS exceed a specified.

To create a billing alarm

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. Change the region to US East (N. Virginia).
3. In the navigation pane, choose **Alarms**, **Billing**.

4. For **whenever my total AWS charges for the month exceed**, specify the monetary amount (for example, 200) that must be exceeded to trigger the alarm and send an email notification.



5. For **send a notification to**, choose an existing notification list or create a new one.
6. Choose **Create Alarm**.

12.3.3 Check the Alarm Status

- Check the status of the billing alarm that you just created.

To check the alarm status

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. Change the region to US East (N. Virginia). Billing metric data is stored in this region and reflects worldwide charges.
3. In the navigation pane, choose **Alarms**, **Billing**.
4. Select the check box next to the alarm after the subscription is confirmed, refresh the console to show the updated status.

12.3.4 Edit a Billing Alarm

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region to US East (N. Virginia). Billing metric data is stored in this region and reflects worldwide charges.
3. In the navigation pane, choose **Alarms**, **Billing**.
4. Select the check box next to the alarm and then choose **Actions**, **Modify**.
5. For **whenever my total AWS charges for the month exceed**, specify the new amount that must be exceeded to trigger the alarm and send an email notification.
6. Choose **Save Changes**.

12.3.5 Delete a Billing Alarm

You can delete your billing alarm if you no longer need it.

To delete a billing alarm

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. If necessary, change the region to US East (N. Virginia). Billing metric data is stored in this region and reflects worldwide charges.
3. In the navigation pane, choose **Alarms**, **Billing**.
4. Select the check box next to the alarm and then choose **Actions**, **Delete**.
5. When prompted for confirmation, choose **Yes, Delete**.

13. ROUTE 53 MANAGEMENT

Amazon Route 53 provides highly available and scalable Domain Name System (DNS), domain name registration, and health-checking web services.

13.1 Objective

To understand the process of AWS Route 53

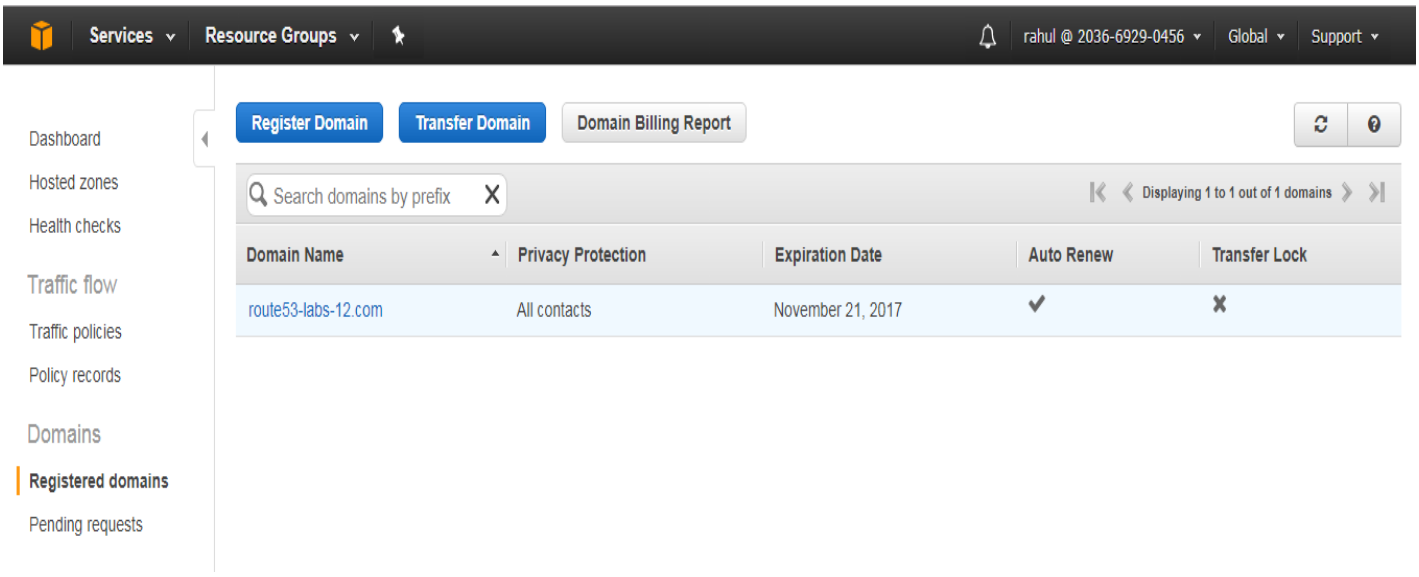
13.2 Assumptions

- You can use Amazon Route 53 to help you get a website or web application up and running
- Route 53 performs three main functions:
  1) **Register domain names**
  2) **Route internet traffic to the resources for your domain**
  3) **Check the health of your resources**

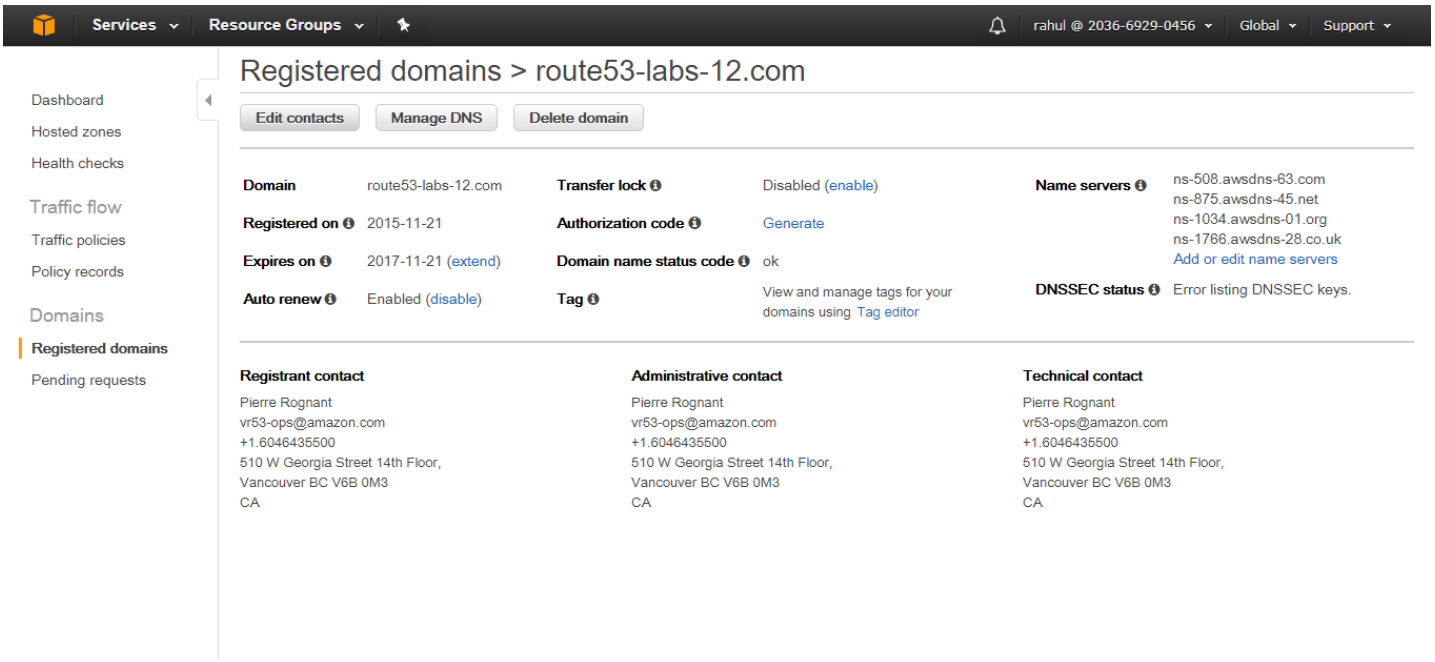
13.3 Procedure

You will first need to know the domain name you are working with.

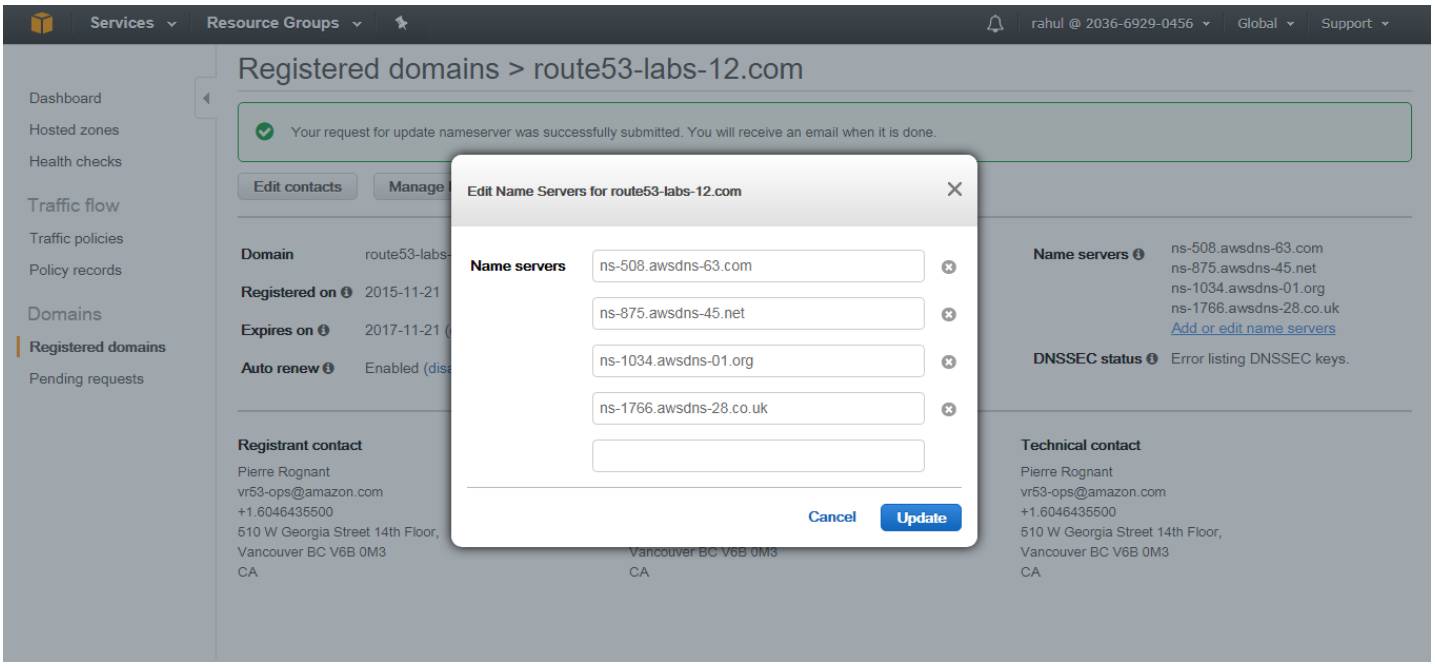
1) In navigation plane, click **Register Domains**.



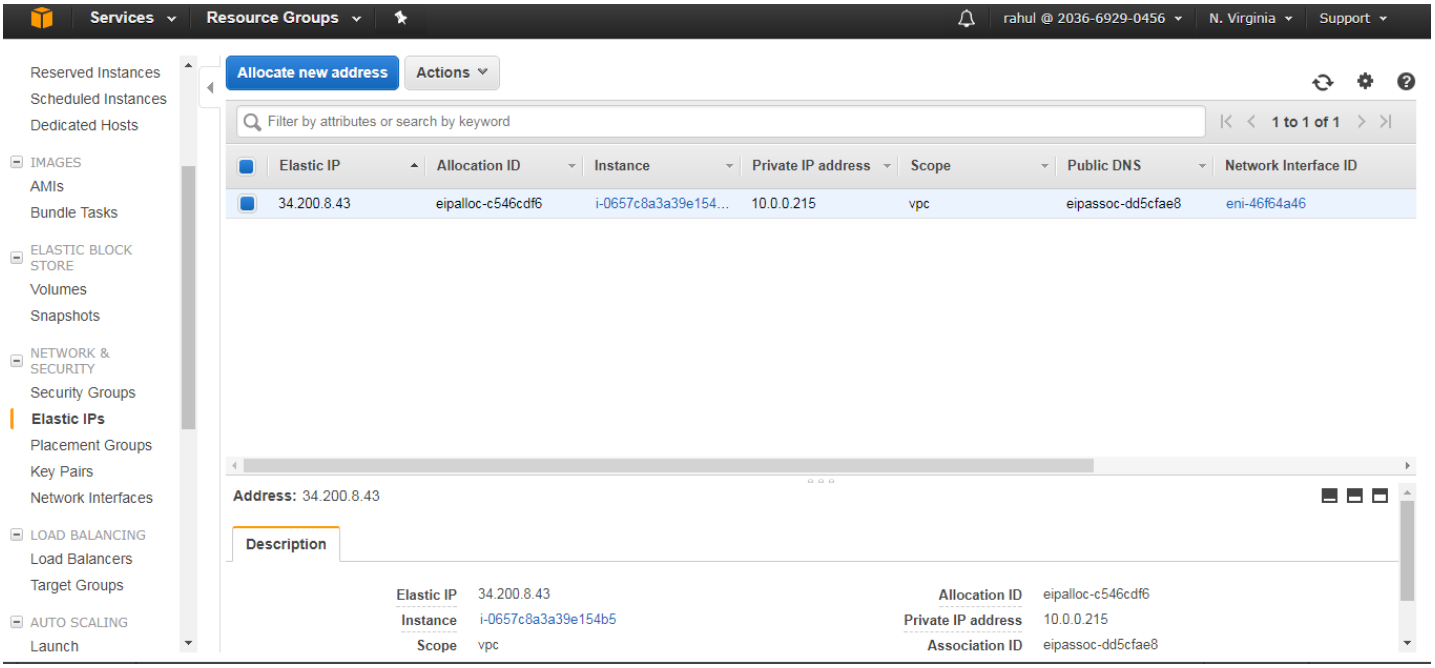
2) Click **Add or Edit name Server**



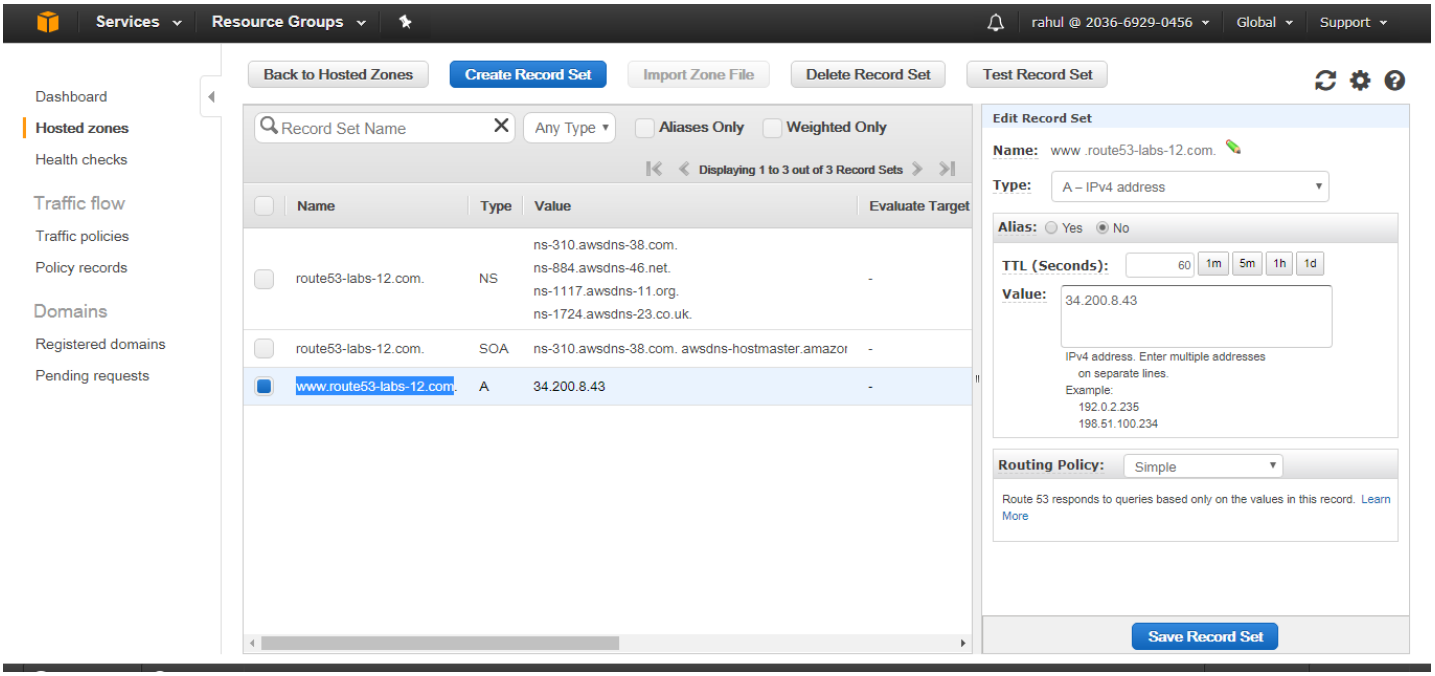
3) Paste the **name server value** for a **Domain Name**:



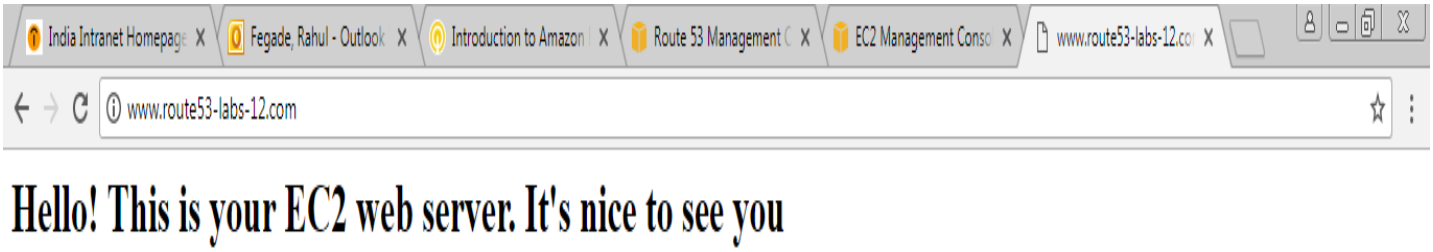
4) Open the new **EC2 console management** and check for the **Elastic IP**



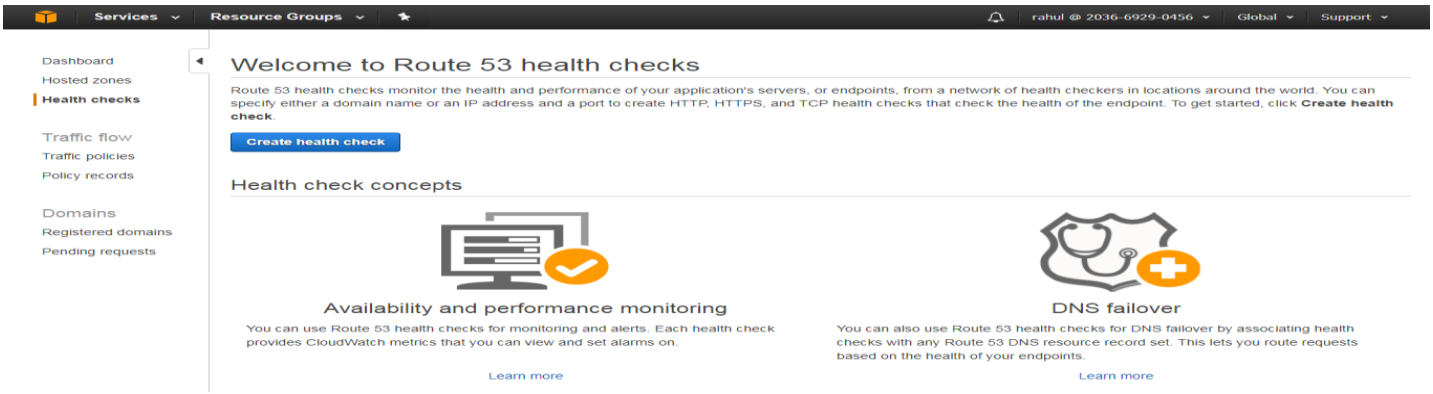
5) In **Route 53 management console**, Click **Hosted Zones** and copy the Domain name to the left of **A**

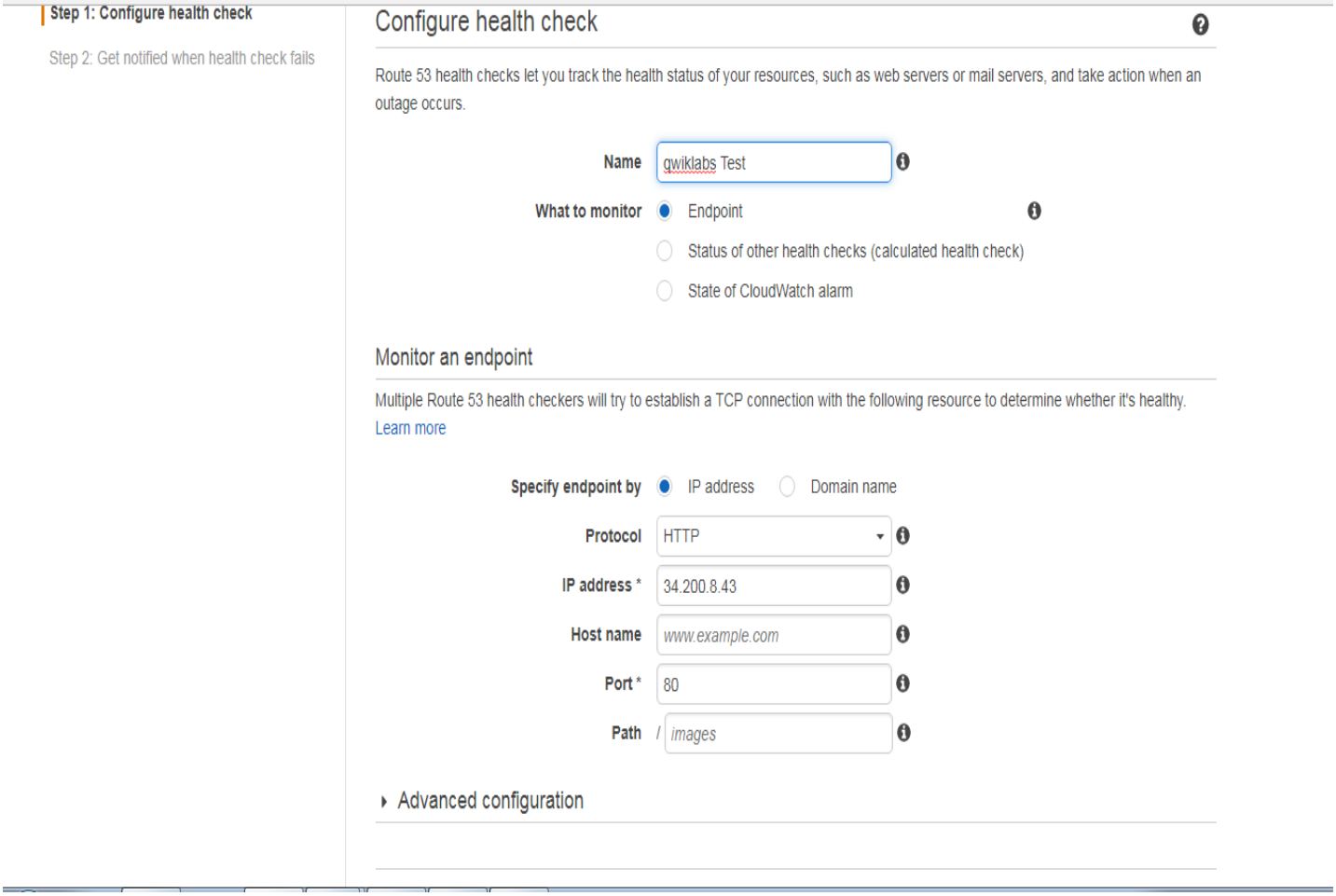


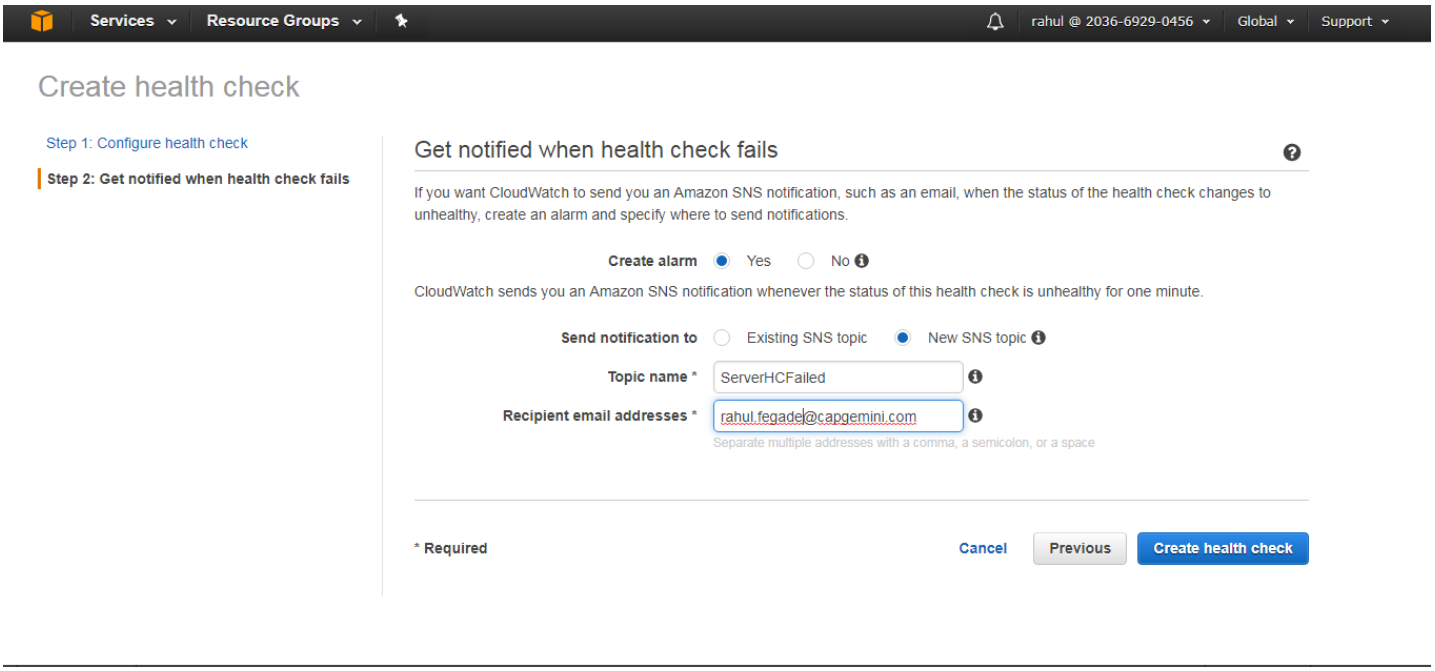
6) Paste the **Domain name** in the Web browser and you will see the following web page



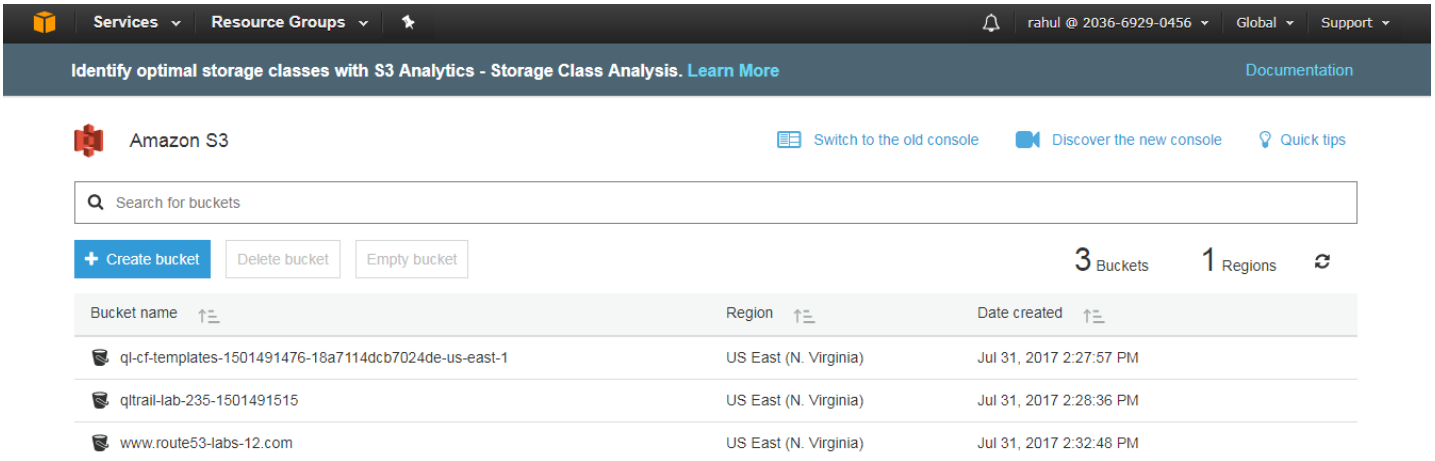
7) In **Route 53 console management**, click on **health checks.** click **Create Health check**




Step 1: Configure health check
Step 2: Get notified when health check fails
Configure health check
Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.
Name
qwiklabs Test
What to monitor
Endpoint
Status of other health checks (calculated health check)
State of CloudWatch alarm
Monitor an endpoint
Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy.
Learn more
Specify endpoint by
IP address
Domain name
Protocol
HTTP
IP address *
34.200.8.43
Host name
www.example.com
Port *
80
Path
/ images
Advanced configuration



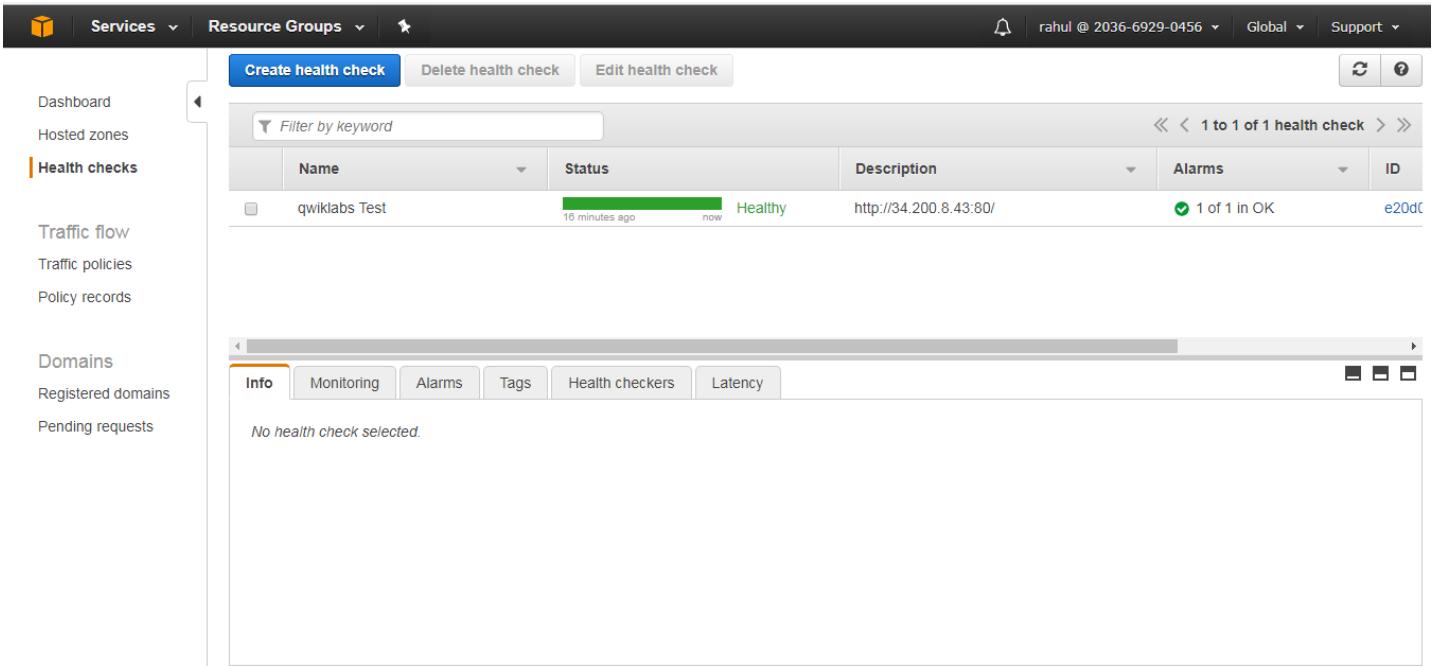
8) In **S3 bucket**, right click on **Domain name** the page will as per the following



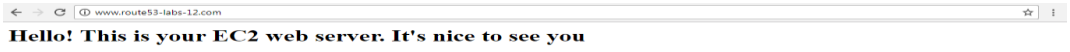
9) The web Page will show an error msg.

Error, this service is not working

10) In the **Health checks** navigation bar, The Status Should be **healthy**

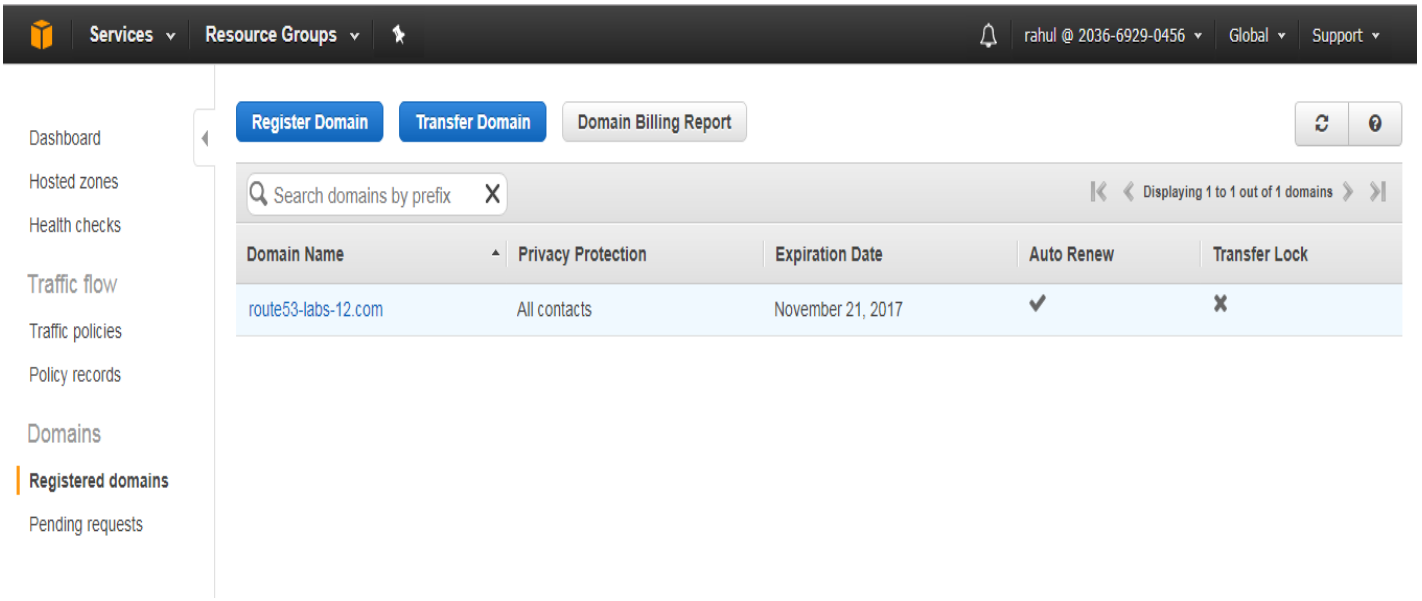


11) And again paste the domain name in the browser, the Web Browser successfully created

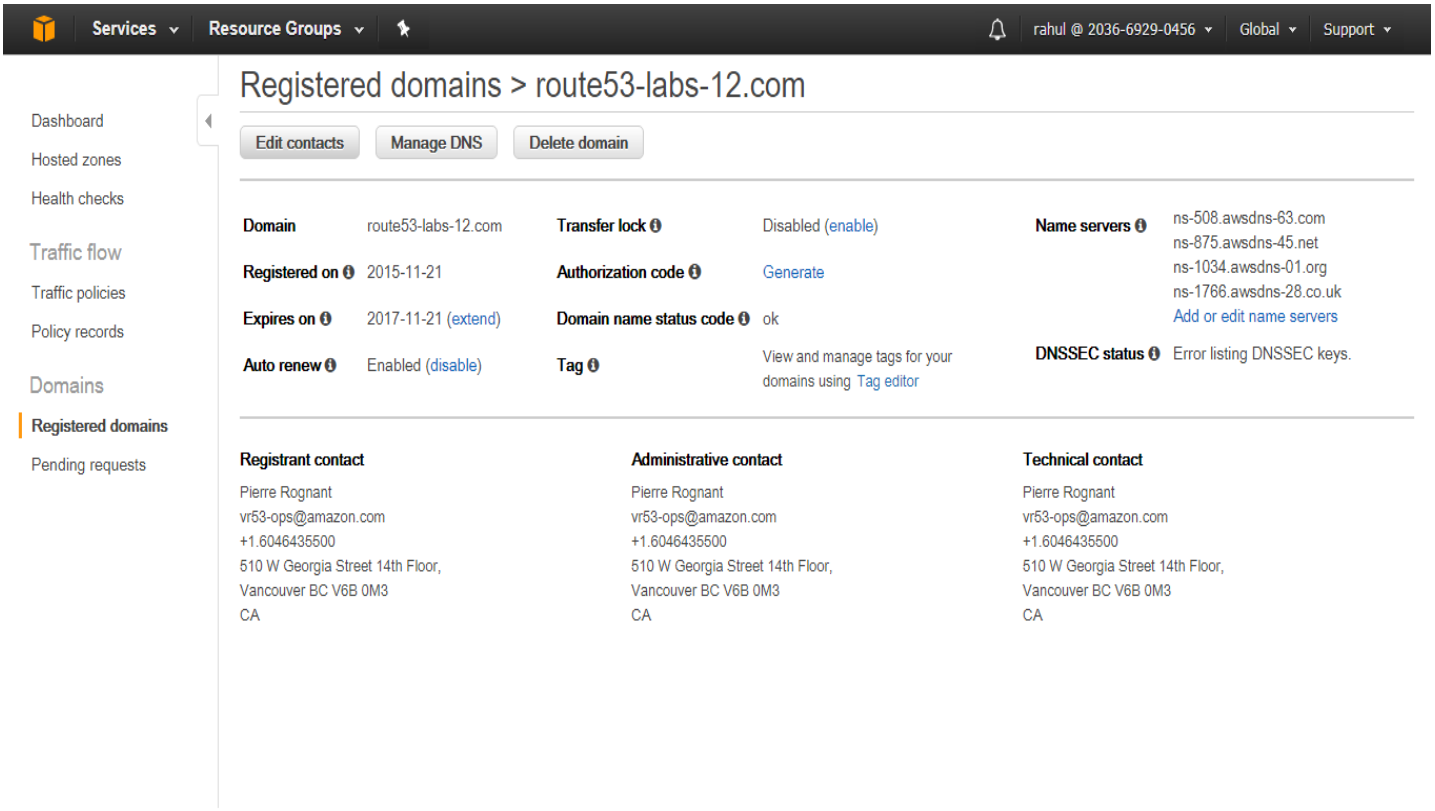


You will first need to know the domain name you are working with.

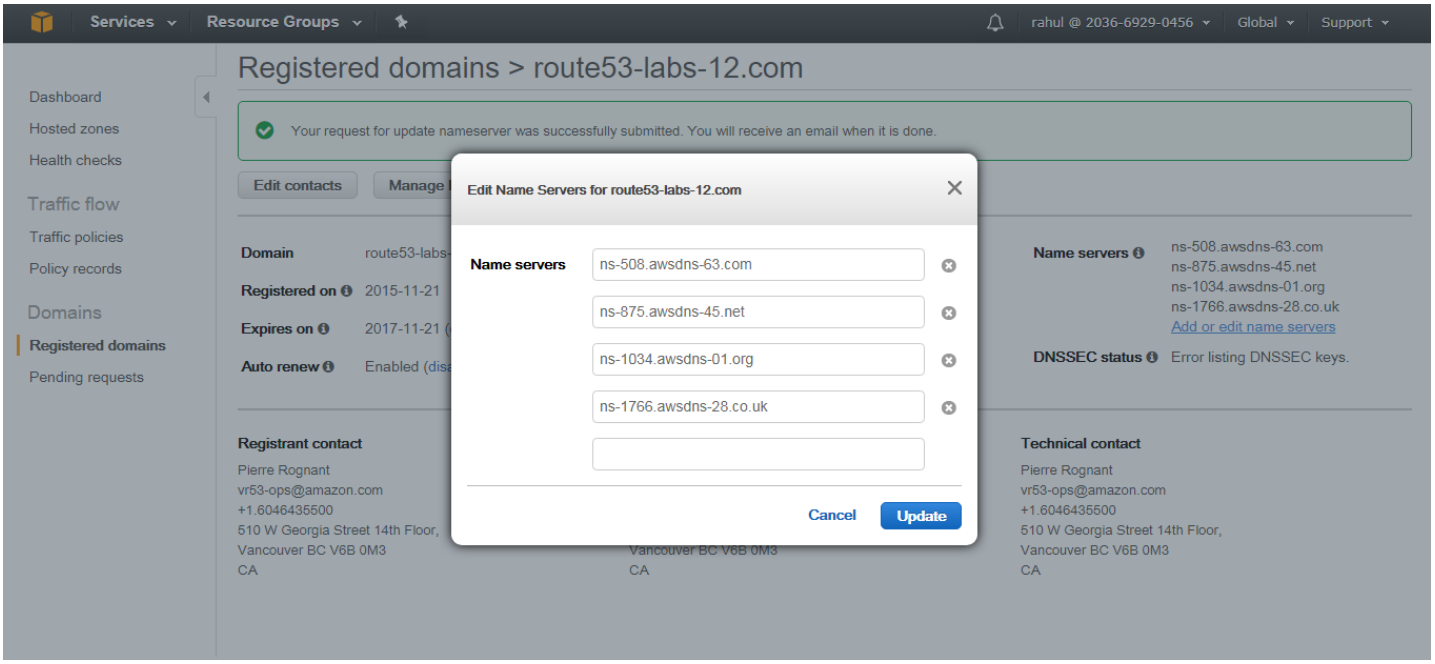
1) In navigation plane, click **Register Domains**.



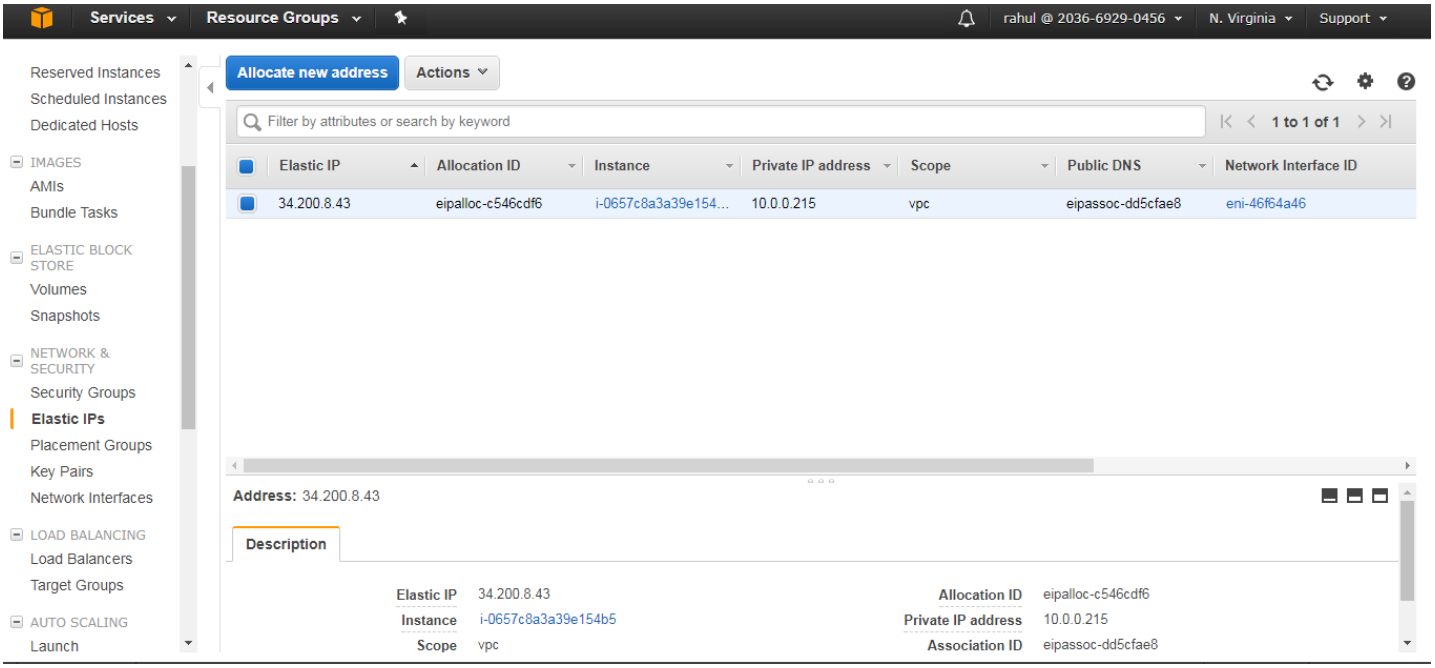
2) Click **Add or Edit name Server**



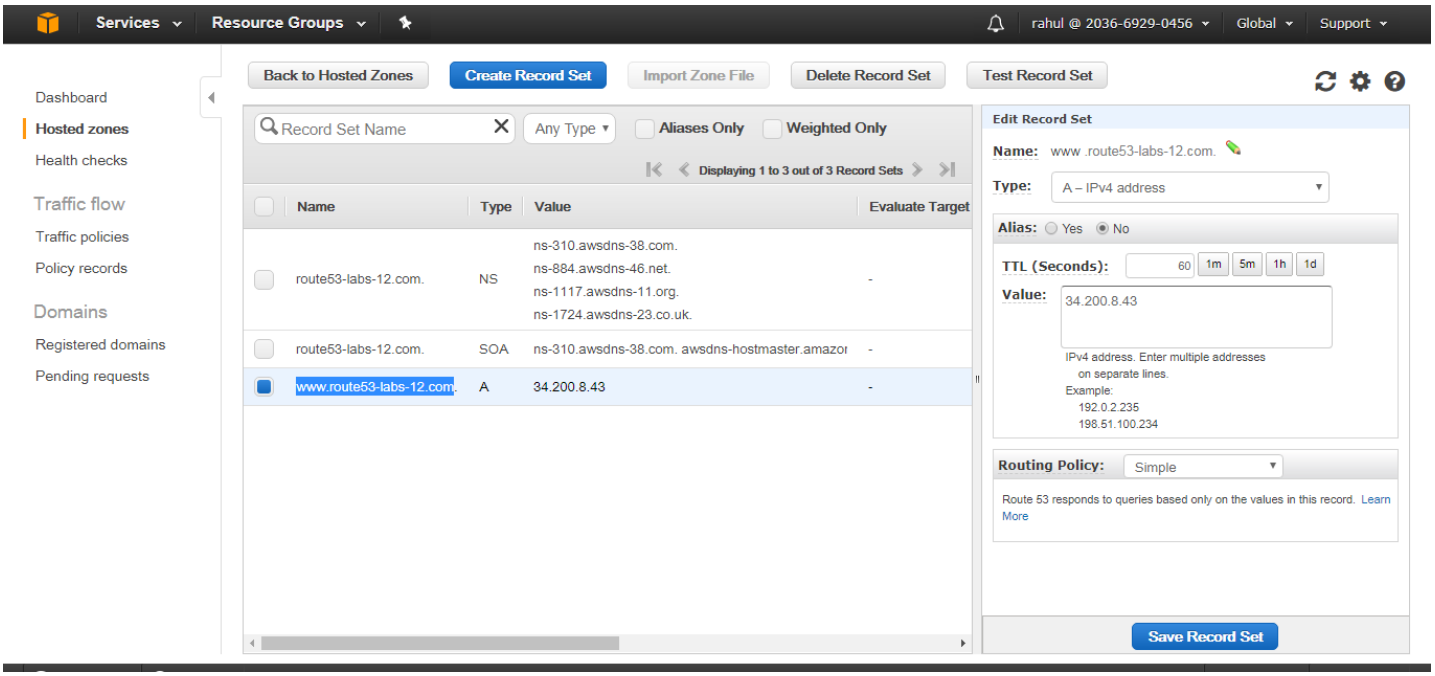
3) Paste the **name server value** for a **Domain Name**:



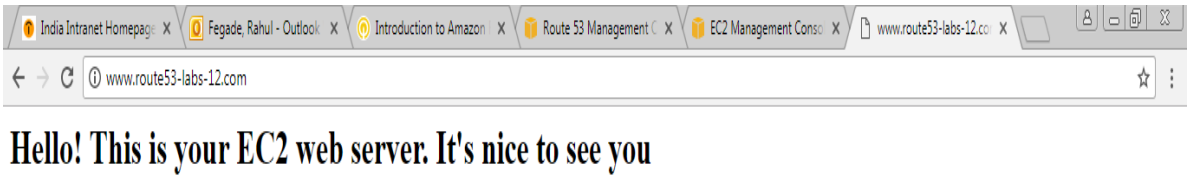
4) Open the new **EC2 console management** and check for the **Elastic IP**



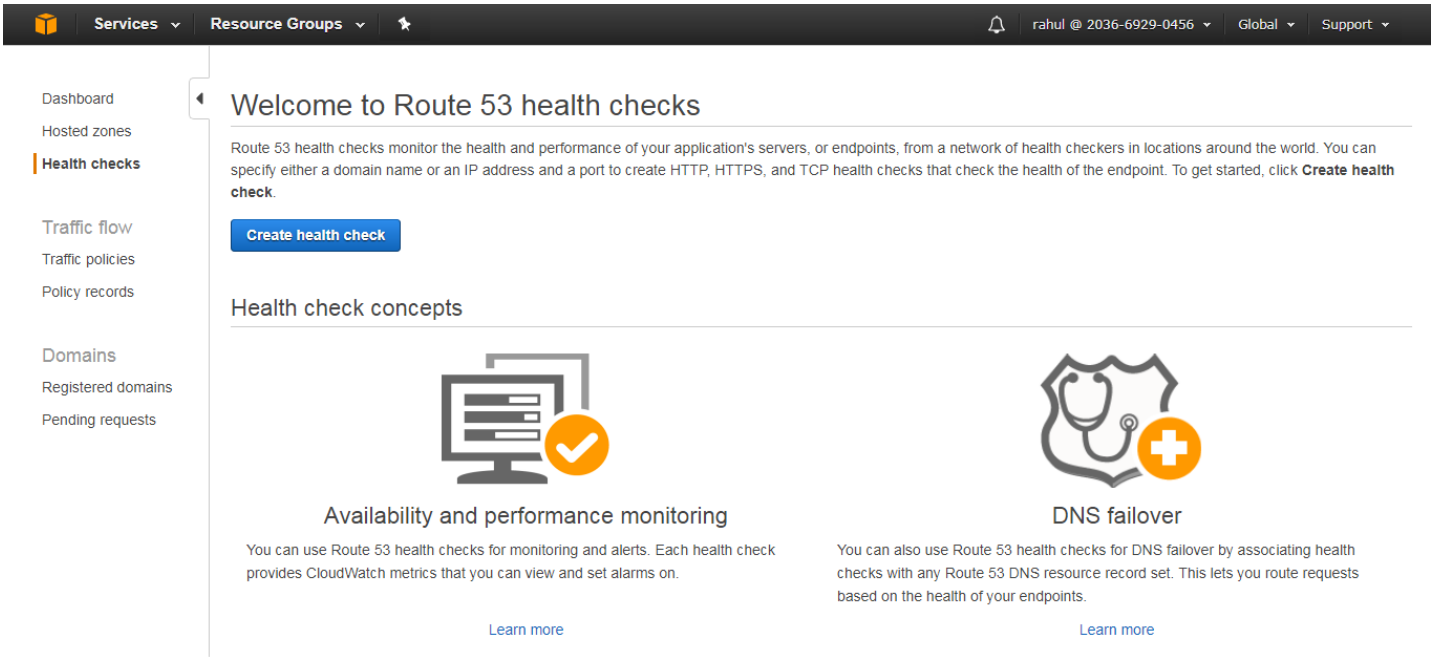
5) In **Route 53 management console**, Click **Hosted Zones** and copy the Domain name to the left of **A**

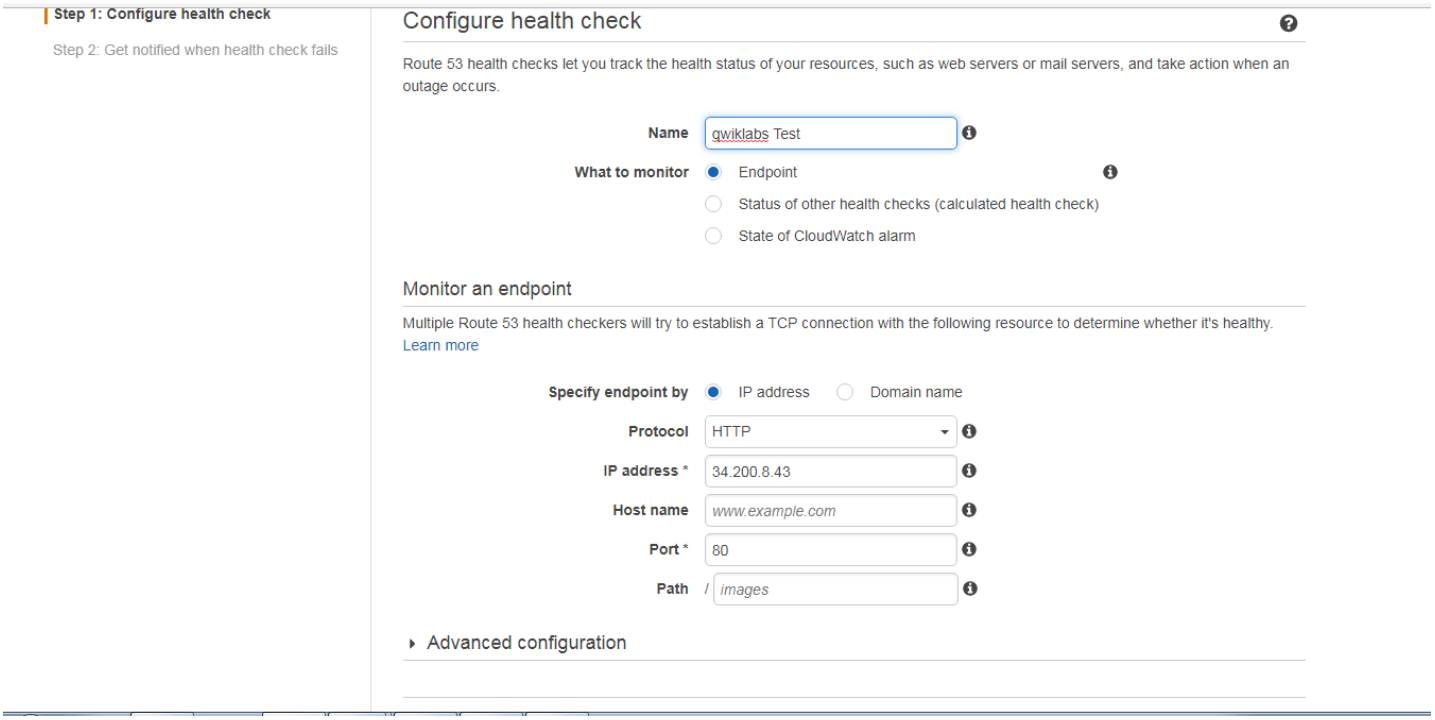


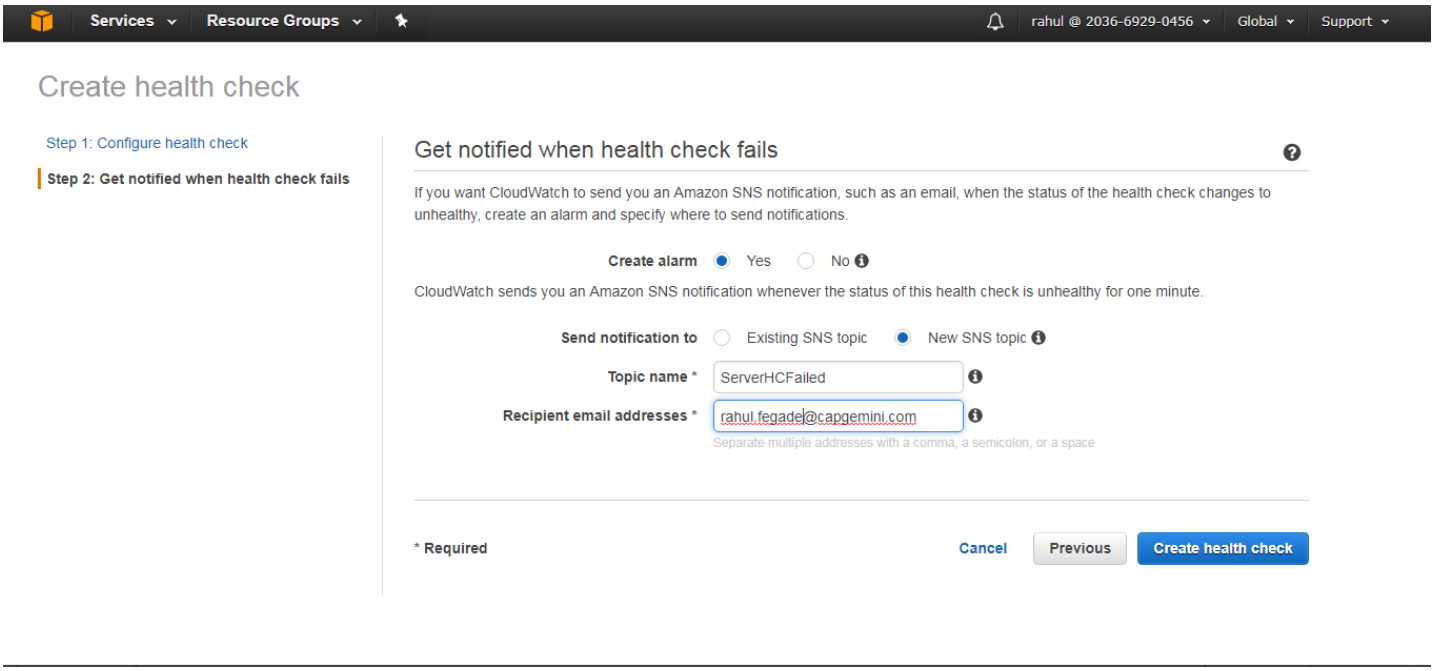
6) Paste the **Domain name** in the Web browser and you will see the following web page



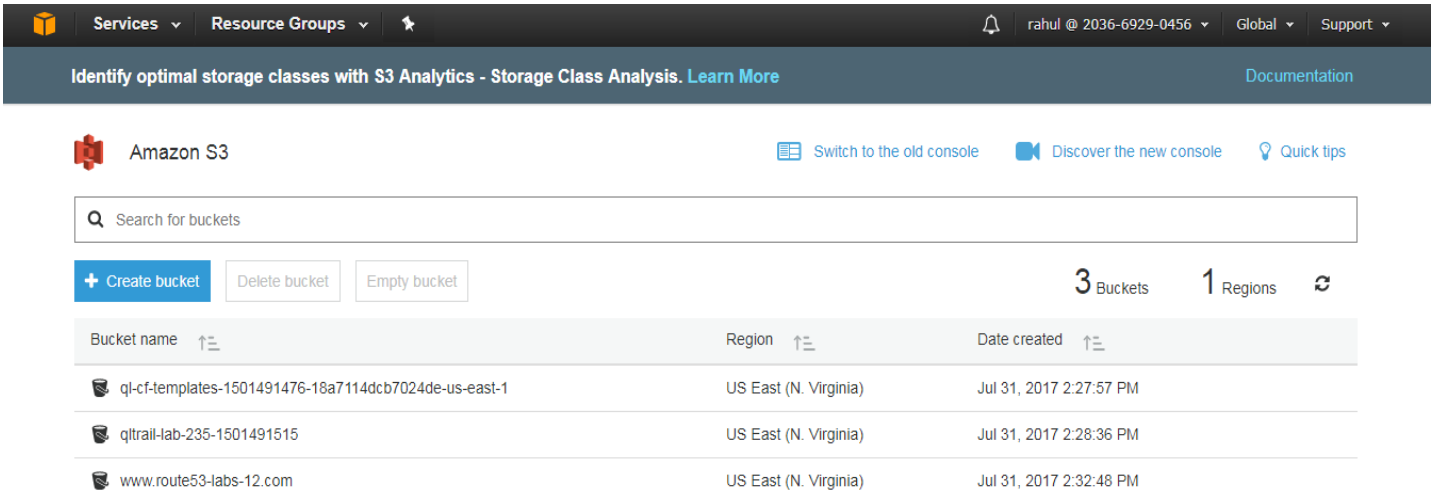
7) In **Route 53 console management**, click on **health checks.** click **Create Health check**







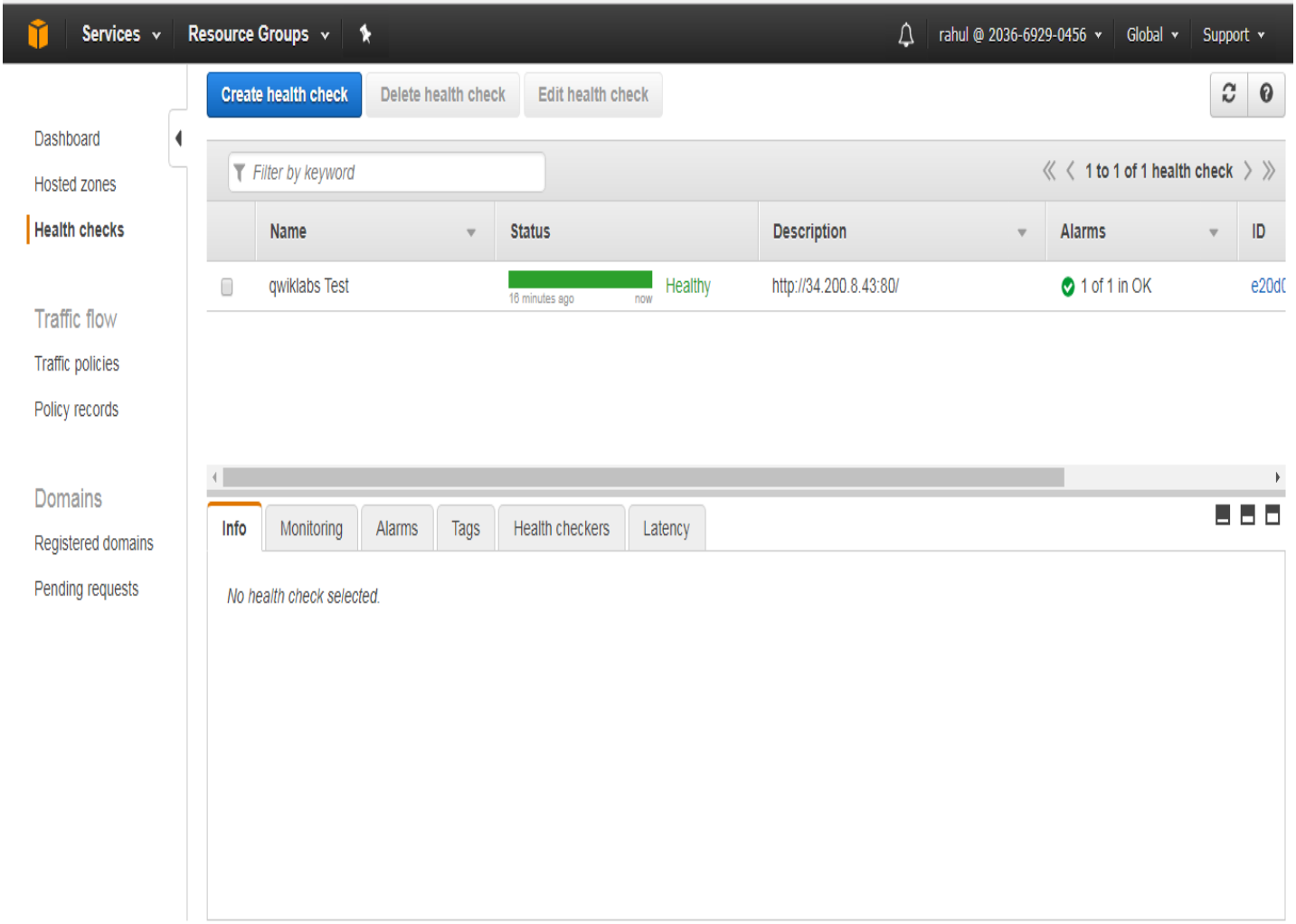
8) In **S3 bucket**, right click on **Domain name** the page will as per the following



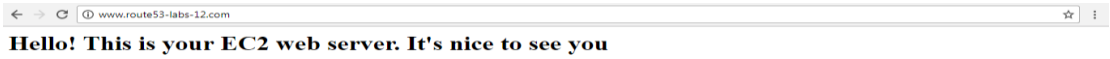
9) The web Page will show a error msg.

Error, this service is not working

10) In the **Health checks** navigation bar, The Status Should be **healthy**



11) And again paste the domain name in the browser, the Web Browser successfully created



14. CLOUDFRONT MANAGEMENT

Amazon Cloud Front is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .php, and image files, to your users.

14.1 Objective

To provide high level guidance's to setup the AWS CloudFront on AWS Cloud.

14.2 Procedure

To setup a AWS Cloud Front, you need to complete the following steps:

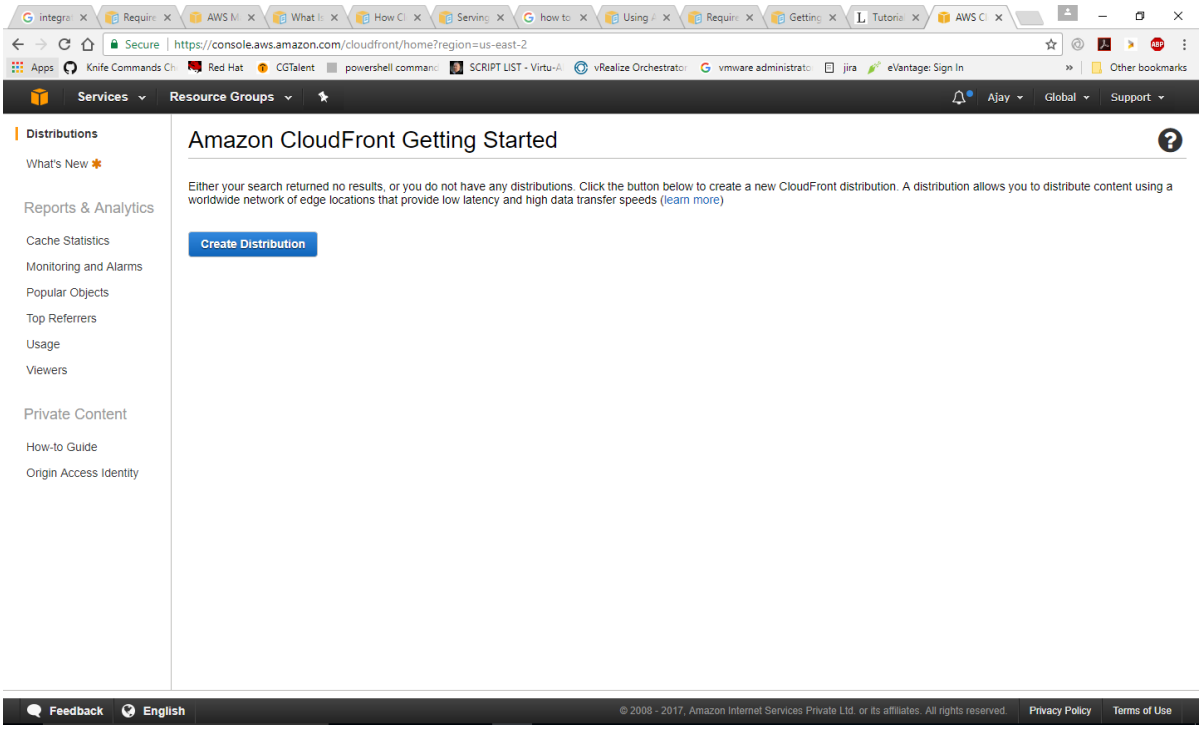
Step 1: Create a S3 Bucket with objects and grant public access

Step 2: Create a Cloud Front Web Distribution

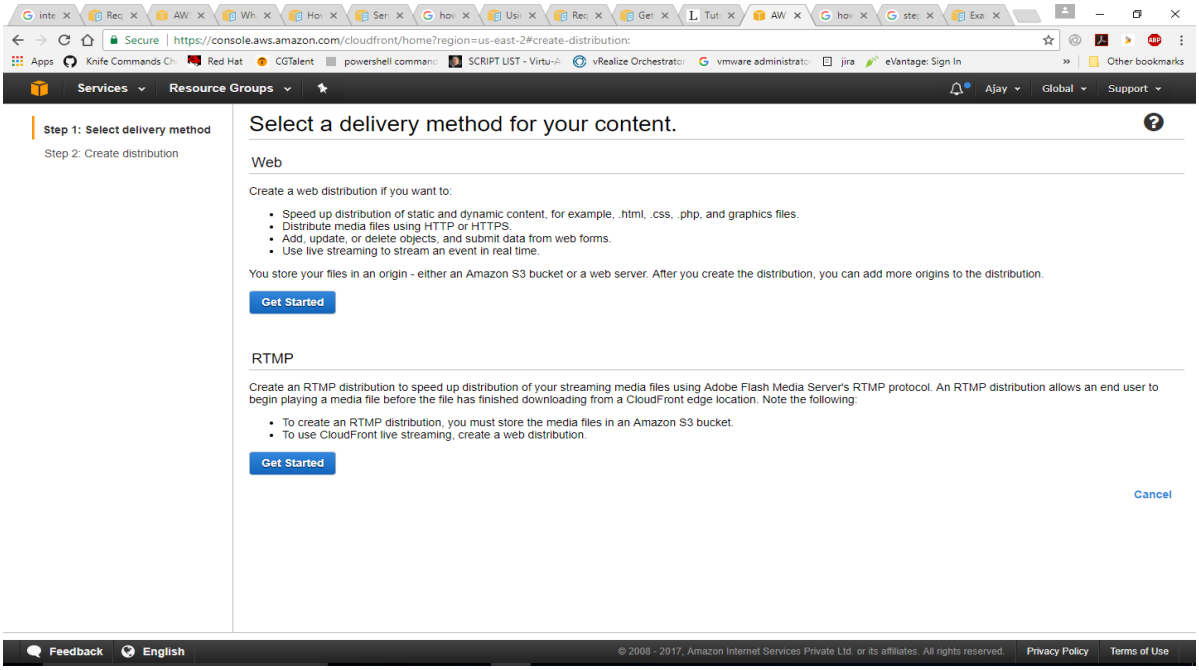
Step 3: Test your Links

Steps to follow

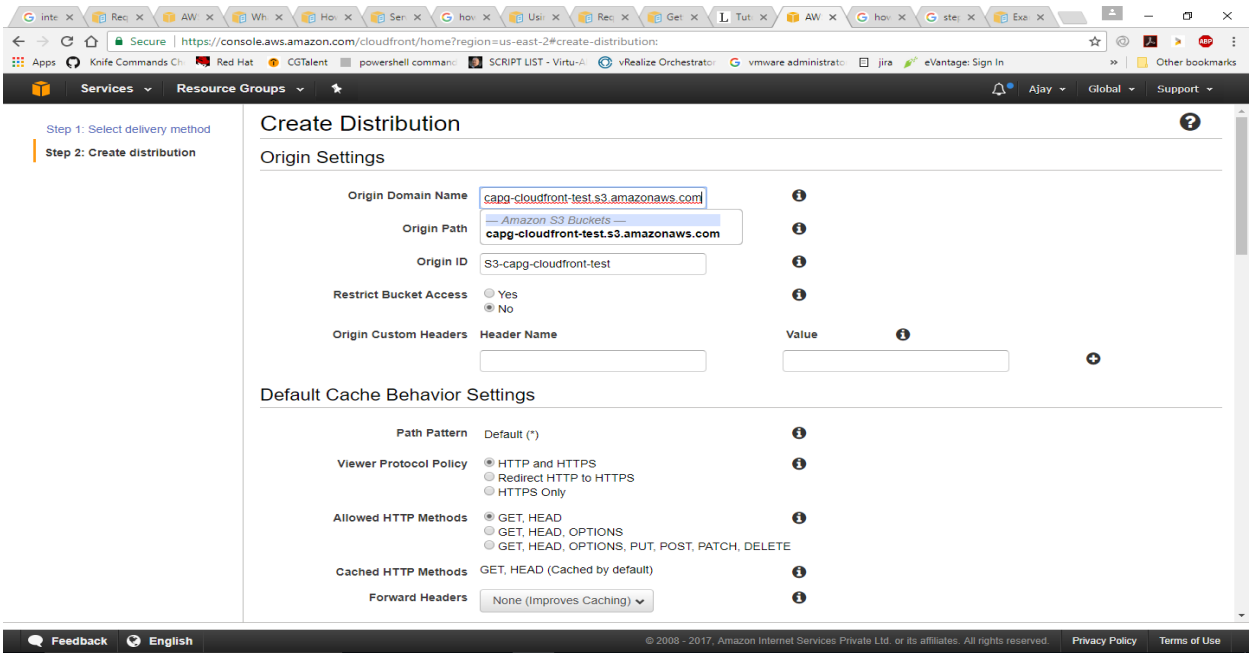
Go to Services on the AWS Management Console and under Networking & Content Delivery select Cloud Front

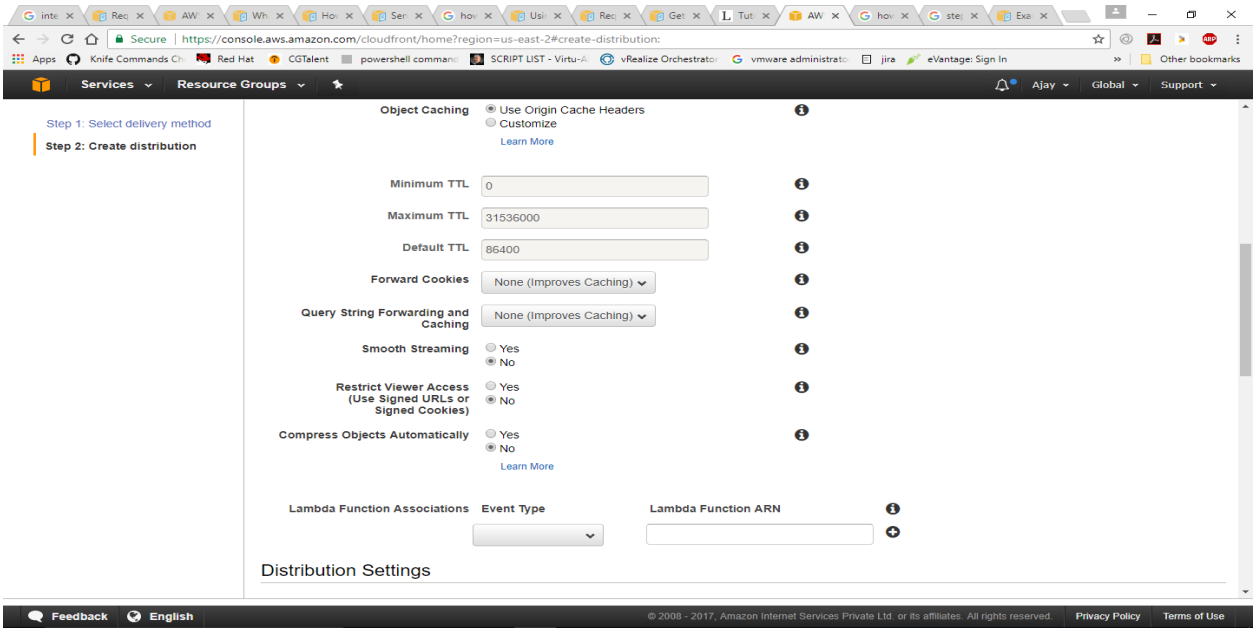


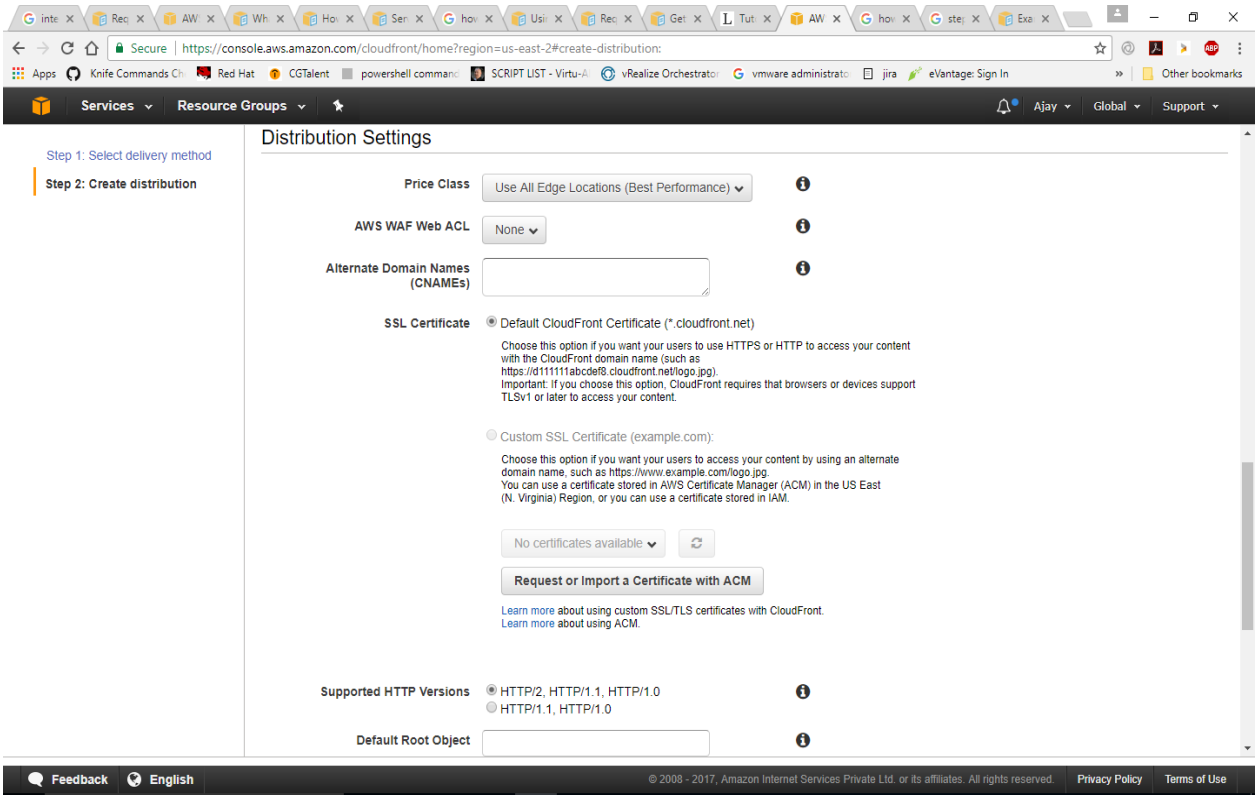
Select delivery method for content

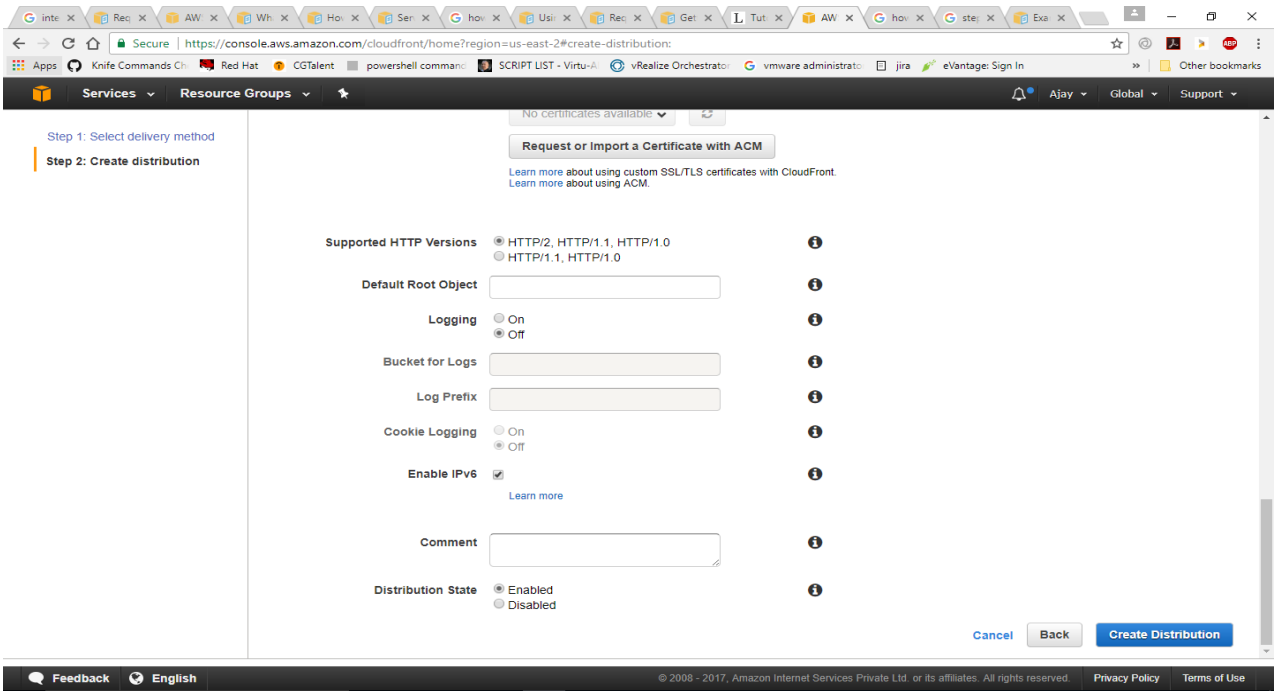


Give the origin domain name

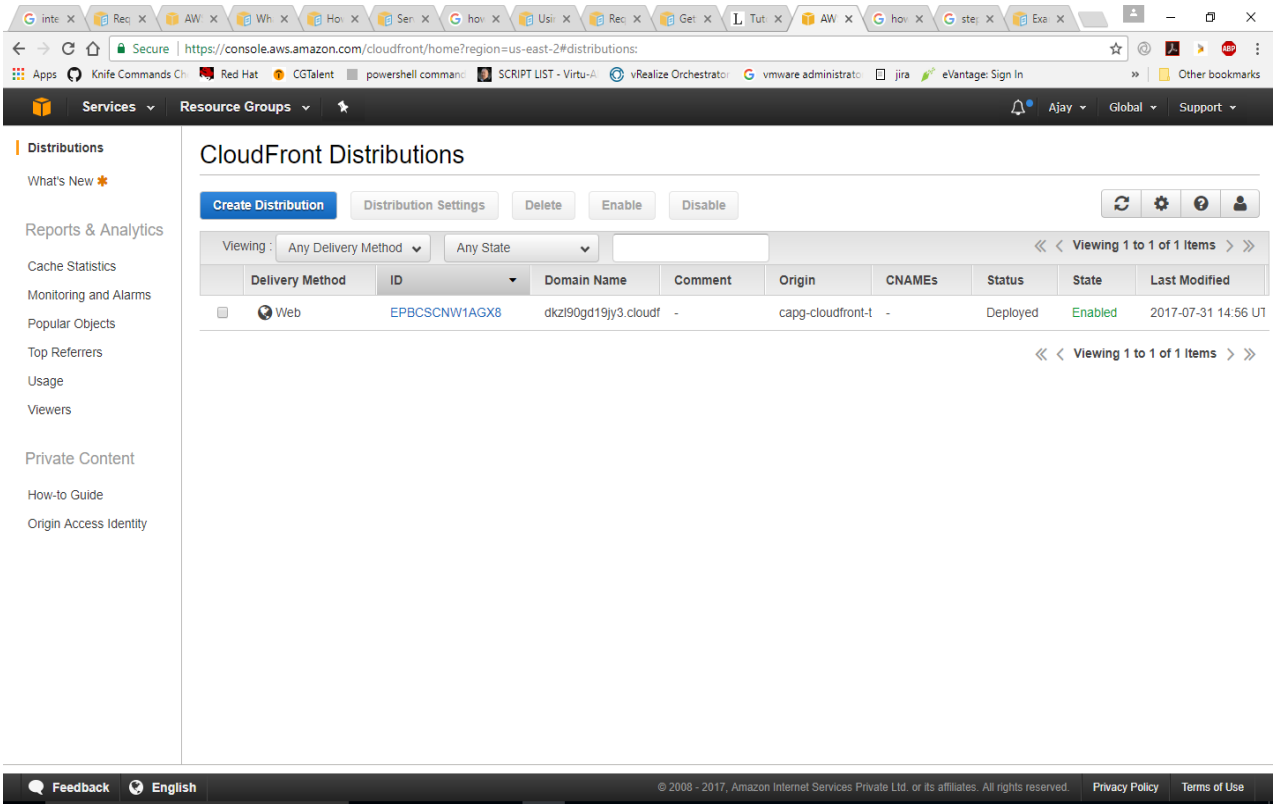




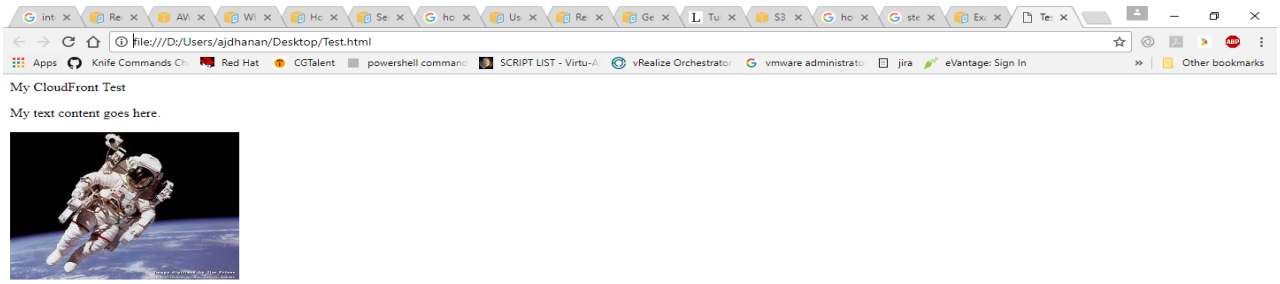

Secure | https://console.aws.amazon.com/cloudfront/home?region=us-east-2#create-distribution:
Services
Resource Groups
Ajay
Global
Support
Step 1: Select delivery method
Step 2: Create distribution
Distribution Settings
Price Class
Use All Edge Locations (Best Performance)
AWS WAF Web ACL
None
Alternate Domain Names (CNAMEs)
SSL Certificate
Default CloudFront Certificate (*.cloudfront.net)
Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as https://d111111abcdef8.cloudfront.net/logo.jpg).
Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.
Custom SSL Certificate (example.com):
Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.jpg.
You can use a certificate stored in AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.
No certificates available
Request or Import a Certificate with ACM
Learn more about using custom SSL/TLS certificates with CloudFront.
Learn more about using ACM.
Supported HTTP Versions
HTTP/2, HTTP/1.1, HTTP/1.0
HTTP/1.1, HTTP/1.0
Default Root Object
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Secure | https://console.aws.amazon.com/cloudfront/home?region=us-east-2#create-distribution:
Services
Resource Groups
Ajay
Global
Support
Step 1: Select delivery method
Step 2: Create distribution
No certificates available
Request or Import a Certificate with ACM
Learn more about using custom SSL/TLS certificates with CloudFront.
Learn more about using ACM.
Supported HTTP Versions
HTTP/2, HTTP/1.1, HTTP/1.0
HTTP/1.1, HTTP/1.0
Default Root Object
Logging
On
Off
Bucket for Logs
Log Prefix
Cookie Logging
On
Off
Enable IPv6
Learn more
Comment
Distribution State
Enabled
Disabled
Cancel
Back
Create Distribution
Feedback
English
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Wait till the status changes to Deployed from In Progress



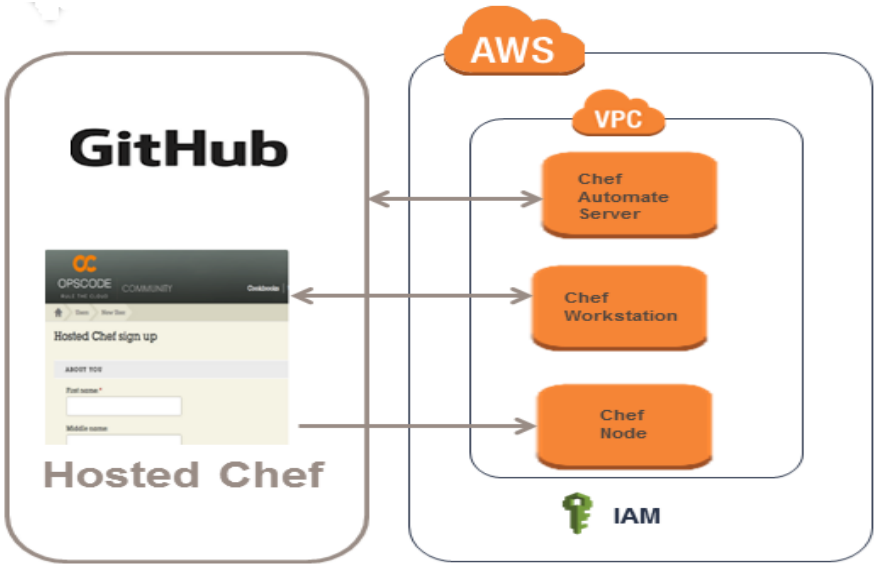
Test Your Links



15. OPSWORKS

15.1 AWS OpsWork (Chef Automate)

A Chef Automate server manages nodes in your environment, stores information about those nodes, and serves as a central repository for your Chef cookbooks.



As per the above diagram, we have used AWS Chef Automated Server, Chef Workstation and Chef Node on AWS inside the VPC. Chef Automate server interacts as follows

- Opscode hosted Chef interacts with Chef Automated Server
- Chef Automated Server interacts with GitHub
- The Chef workstation interacts with the hosted chef used to manage the chef nodes.

15.1.1 Objective

Understand the process to create AWS OpsWork (Chef Automate)

15.1.2 Assumptions

- AWS account, VPC configuration, EC2 Keypair and Security groups , IAM user access
- Chef servers, workstation and node 1 is already running

Server Name	Instance Type	OS Type
Chef Server	Hosted Chef Server	
Chef Workstation	Micro instance	RHEL 7.2

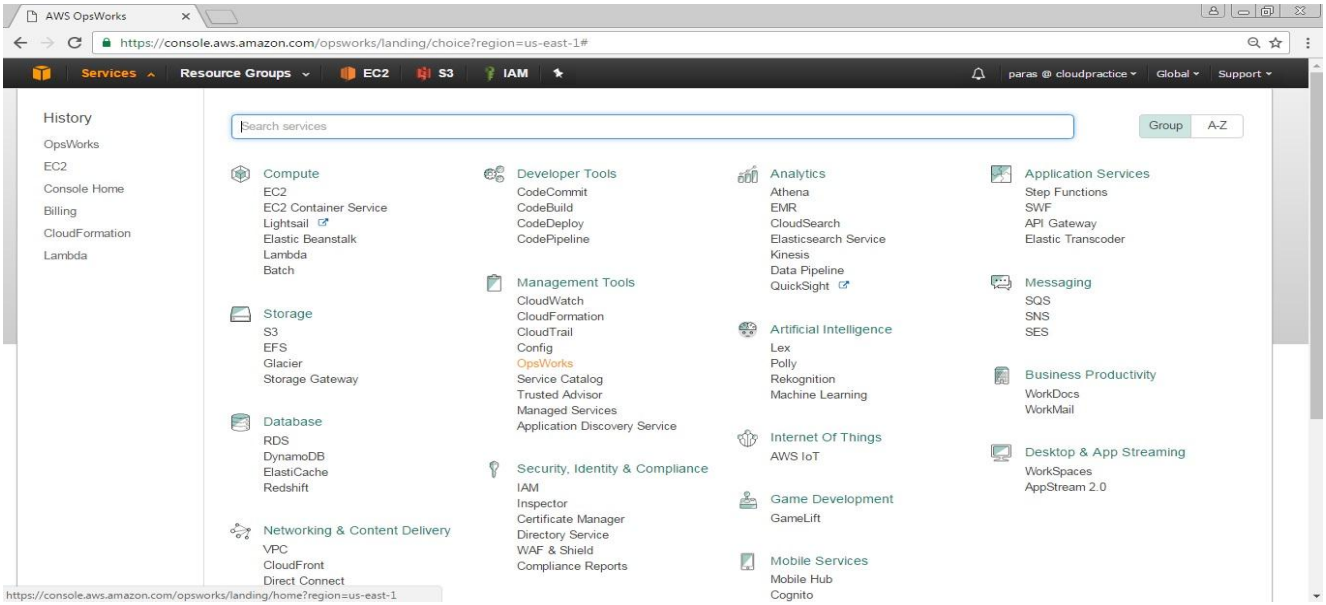
Node 1	Micro instance	RHEL 7.2

- Git / GitHub

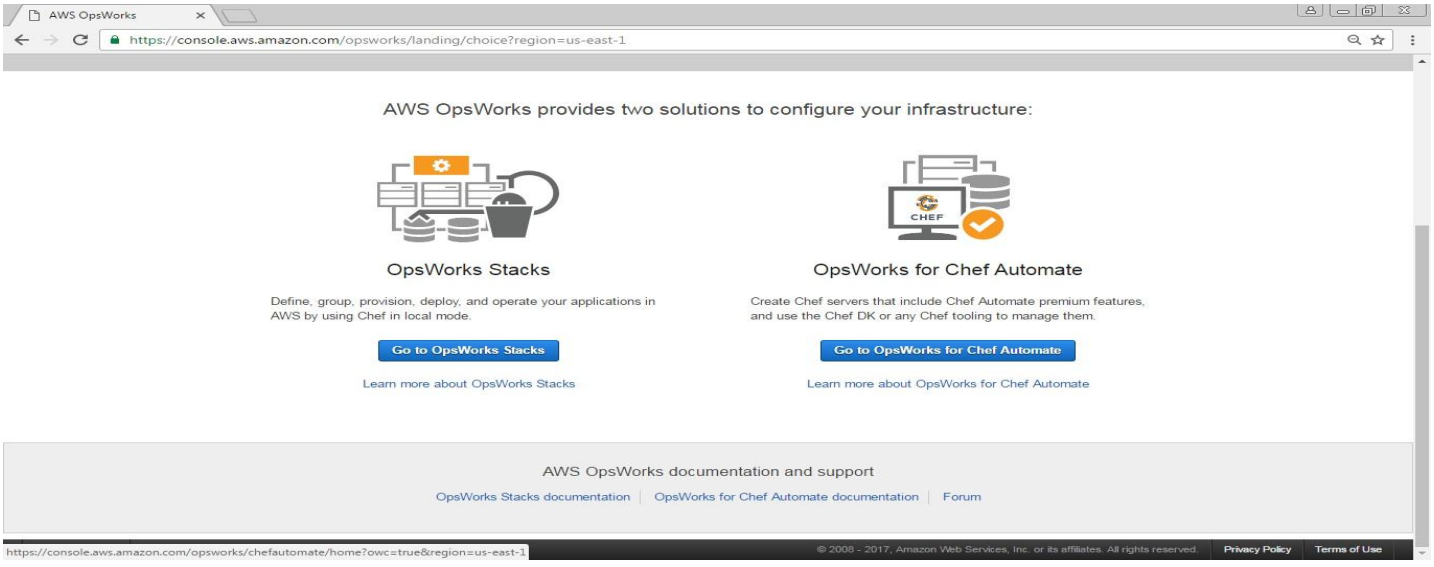
15.1.3 Procedure

15.1.3.1 Create a Chef Automate Server

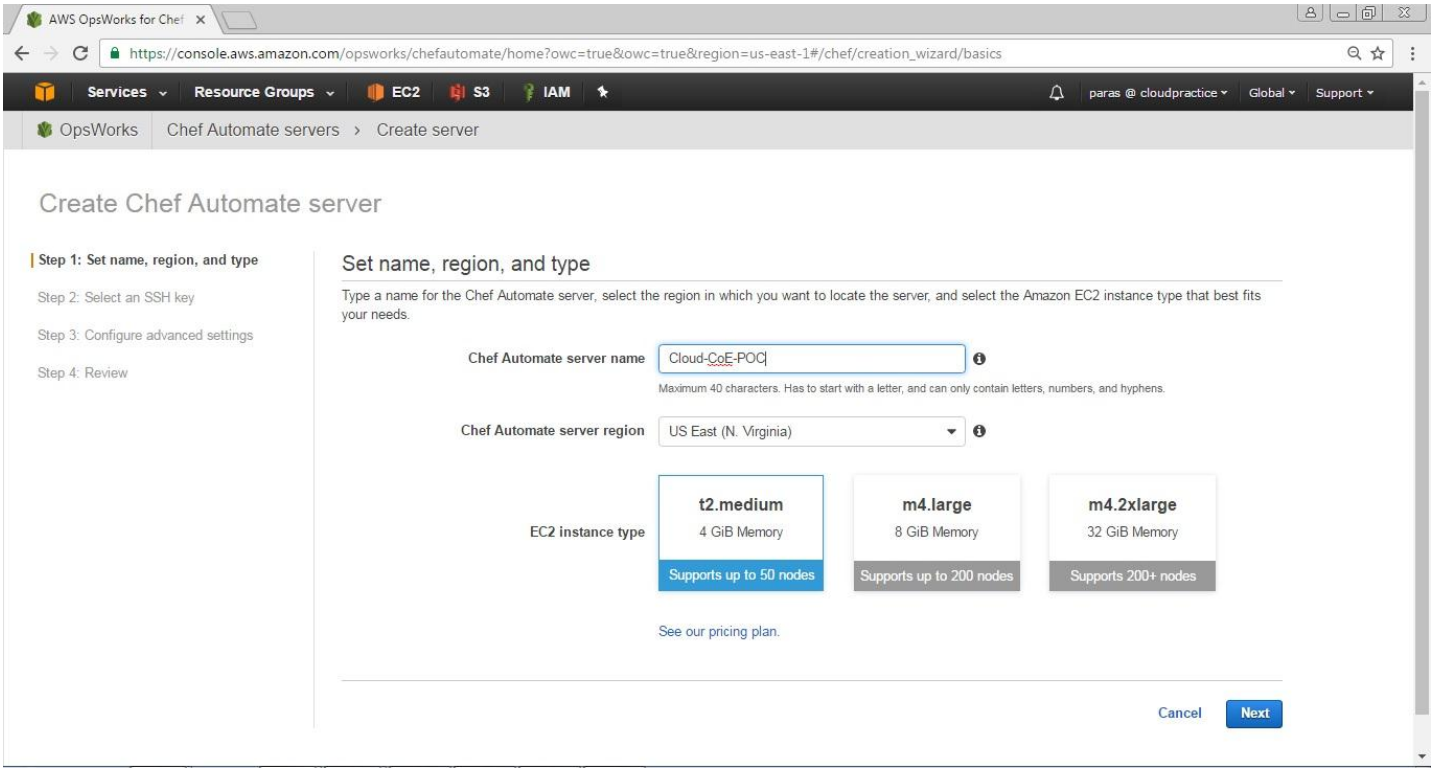
- Open the AWS Management console
- Click on **OpsWorks** under **Services**.



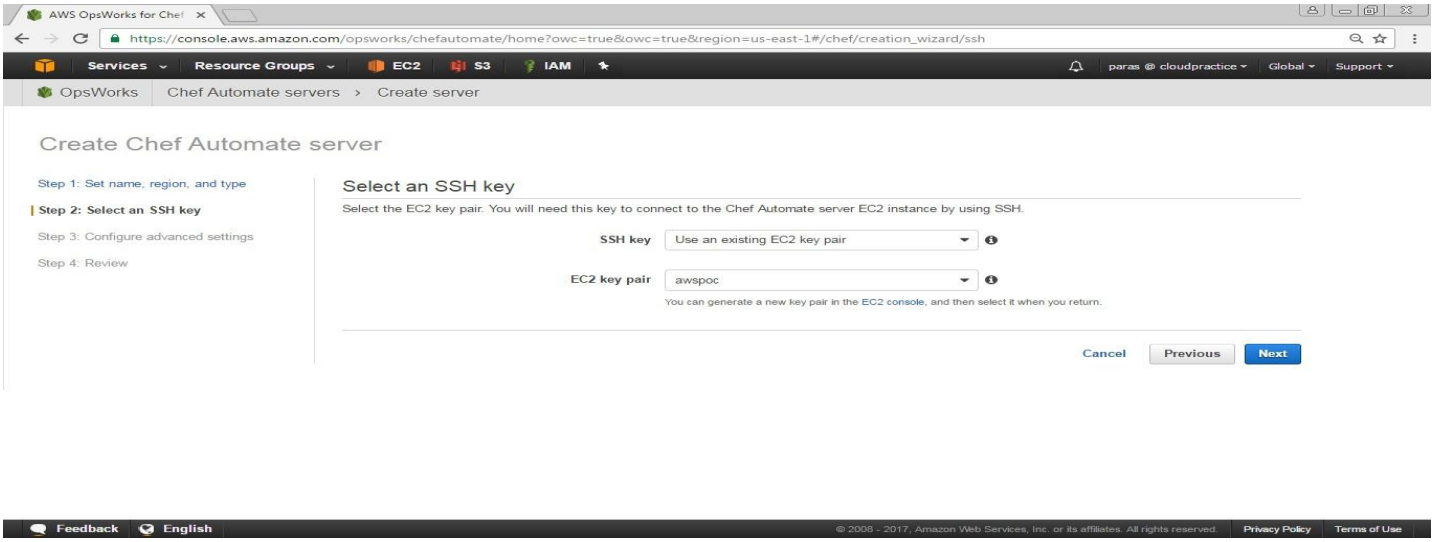
- From **AWS OpsWork** console, click on **Go to OpsWorks for Chef Automate** button.



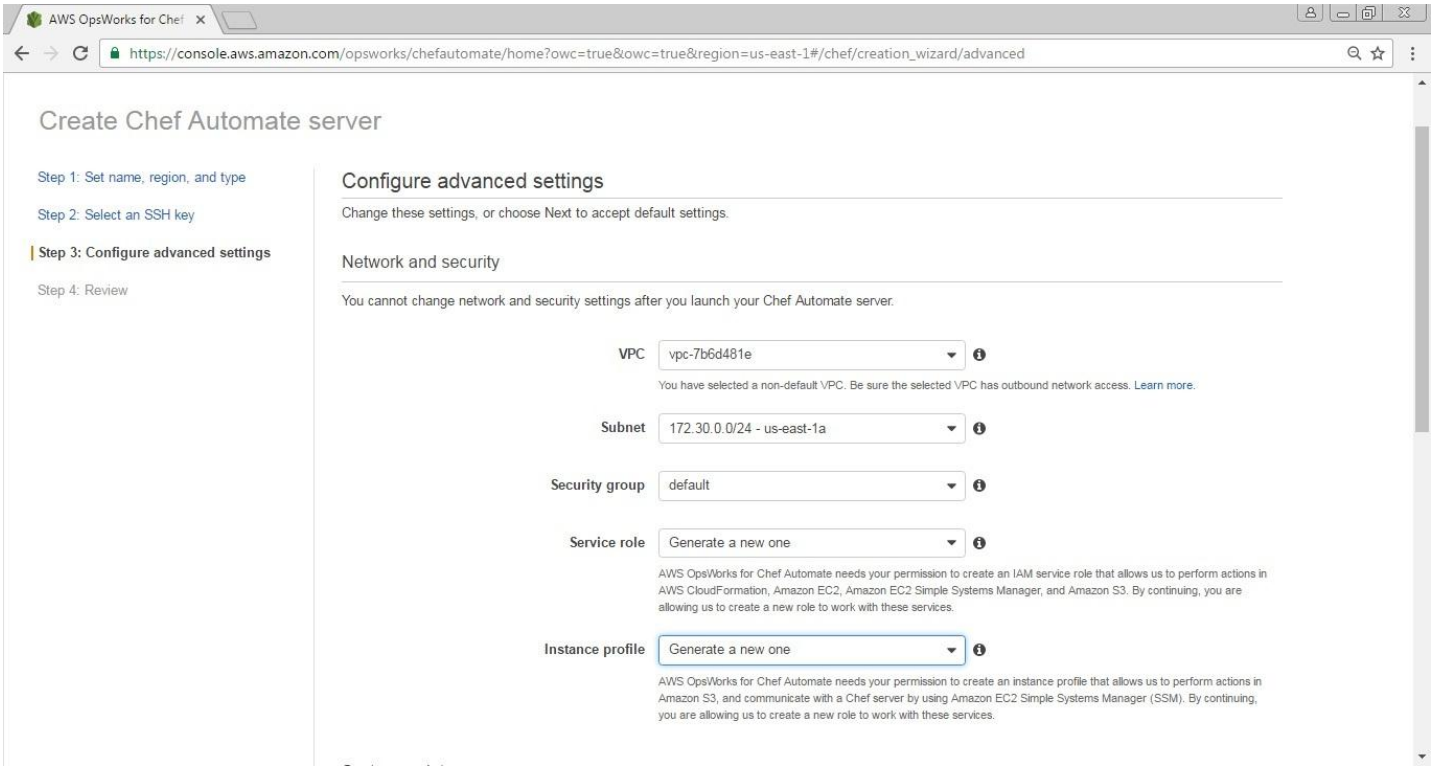
- Provide the **Chef Automate server name,** select the **Chef Automate server region** where you want to host it, **EC2 intstance type (** select EC2 type as t2.medium as it supports upto 50 nodes **)** and click on **NEXT** button.



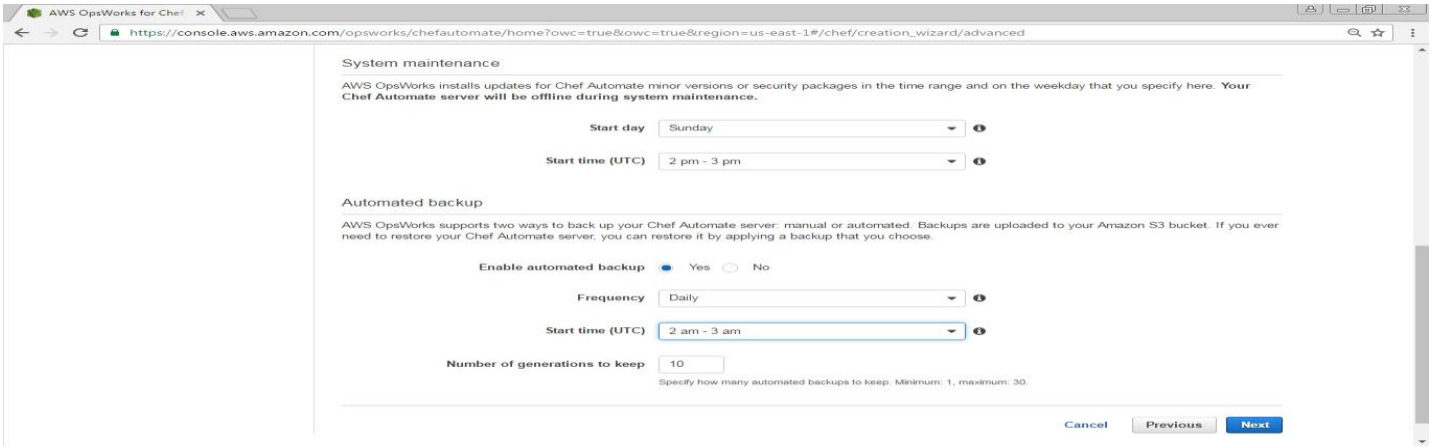
- In the **SSH Key**, select the option **Use an existing EC2 Key pair.** In **EC2 key pair** select the key you want to use for Chef Automate Server (Ex. awspoc key pair).



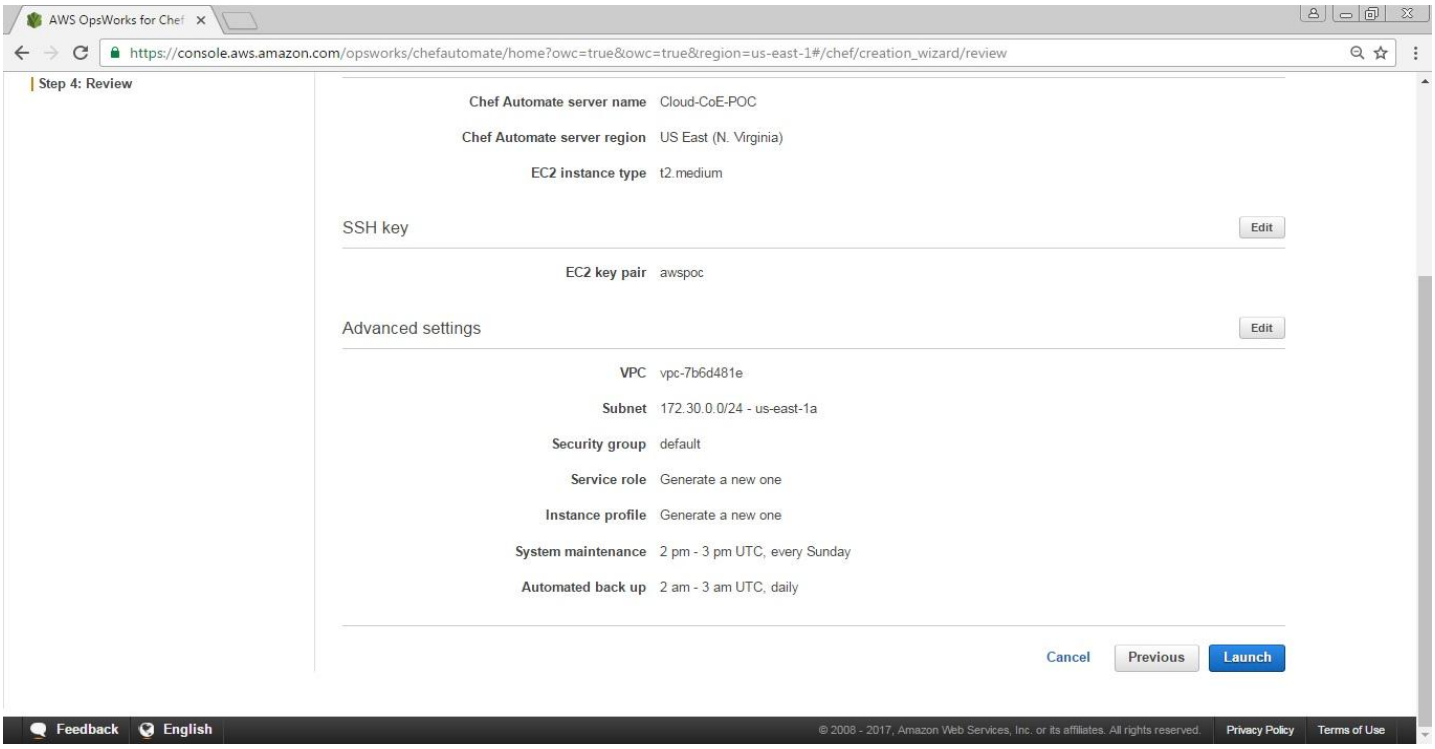
- In **Configure advanced settings**, Under **Network and security** select your **VPC, Subnet, Security group** (like in my case it is default group that opens port 443 – https and port 22 -ssh), Select **Service role** Generate a new one and the same applies to **Instance Profile**.



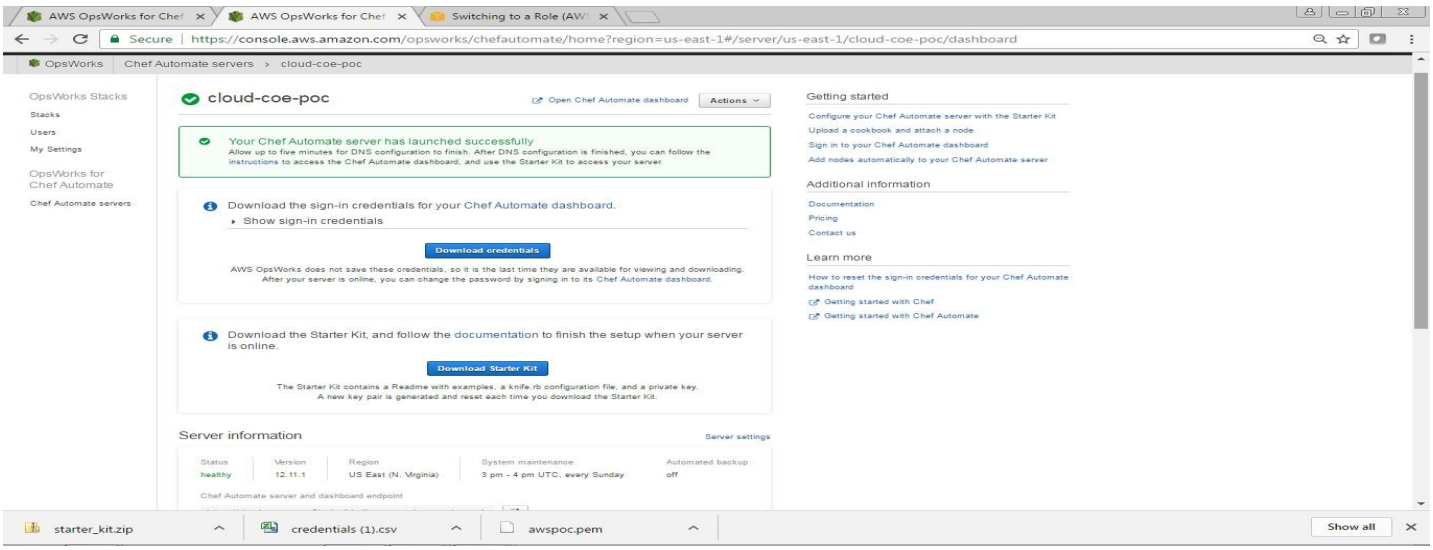
- In **Configure advanced settings**, Under **System maintenance** selects the **Start day** and **Start time.** At the selected day and time your server will be offline.
  Under **Automated backup,** Select **Yes** *(if required only)* radio button for **Enable automated backup,** select the **Frequency, Start time** and **Number of generations to keep.** Now click on **Next** button.



- Review your configuration and click on **Launch** button



- Your Chef Automated server has launched successfully. Now press ***Download Credentials*** and ***Download starter kit***.



Server information

Server settings

Status	Version	Region	System maintenance	Automated backup
healthy	12.11.1	US East (N. Virginia)	3 pm - 4 pm UTC, every Sunday	off

Chef Automate server and dashboard endpoint

https://cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io

Before you can work with your Chef Automate server, make sure you have Chef DK installed.

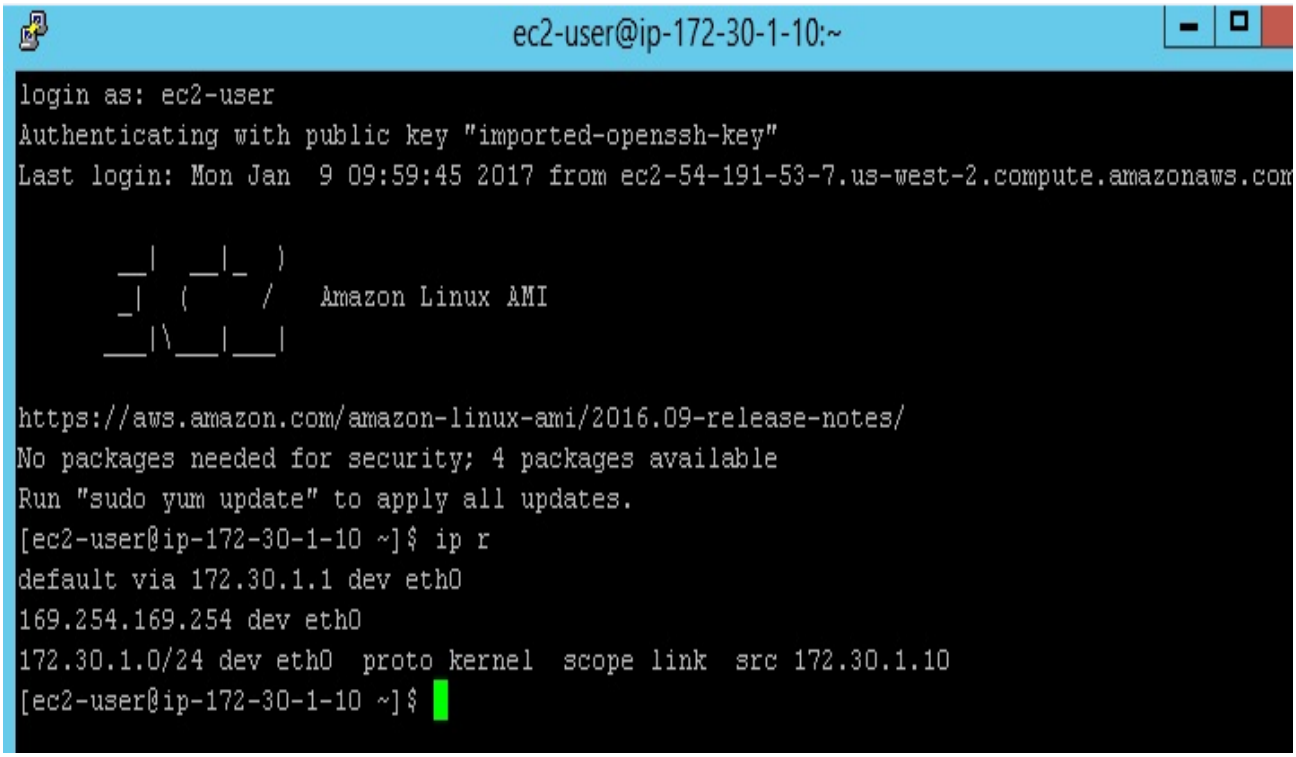
Recent events

Time (UTC)	Description
2017-01-09T09:27:59Z	Successfully launched Server cloud-coe-poc
2017-01-09T09:27:59Z	Created DNS cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io
2017-01-09T09:27:58Z	Finished CloudFormation stack arn:aws:cloudformation:us-east-1:201417269482:stack/aws-opsworks...
2017-01-09T09:20:51Z	Start creating CloudFormation stack arn:aws:cloudformation:us-east-1:201417269482:stack/aws-ops...

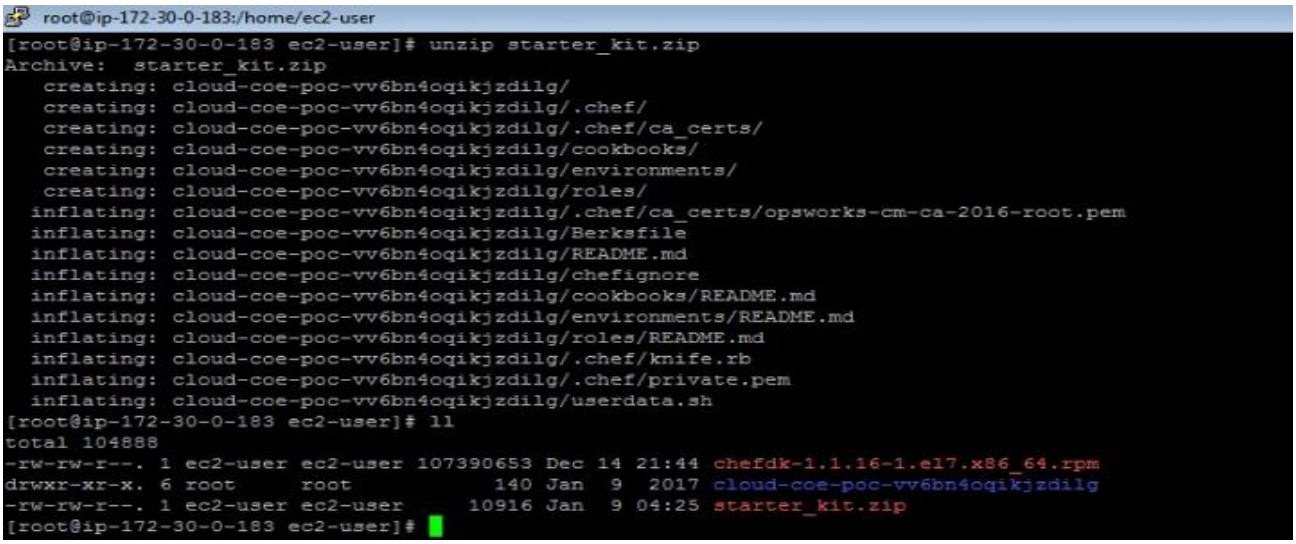
Recent backups

Status	Creation time (UTC)	Type	Description
			No items.

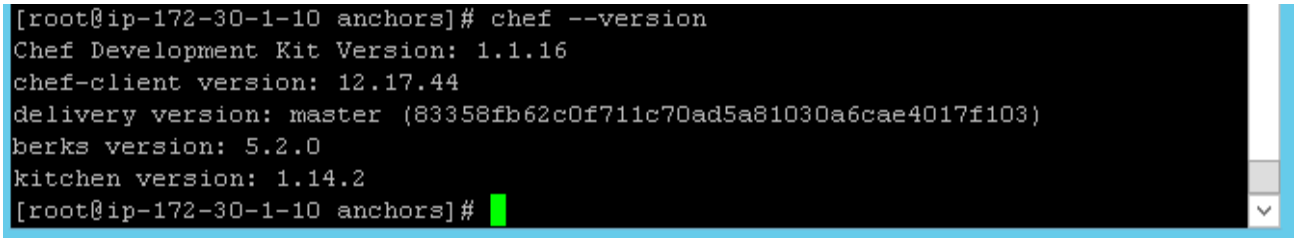
- Open the Chef automation server using putty and login as ec2-user



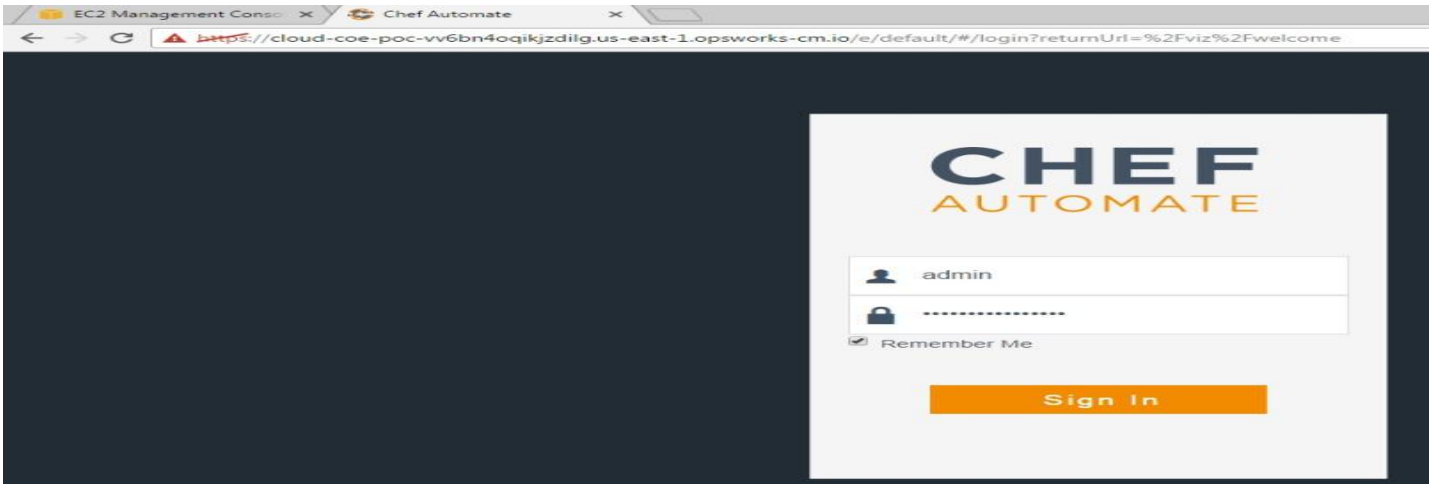
- Use the downloaded starter kit and copy it in the Chef Automate Server



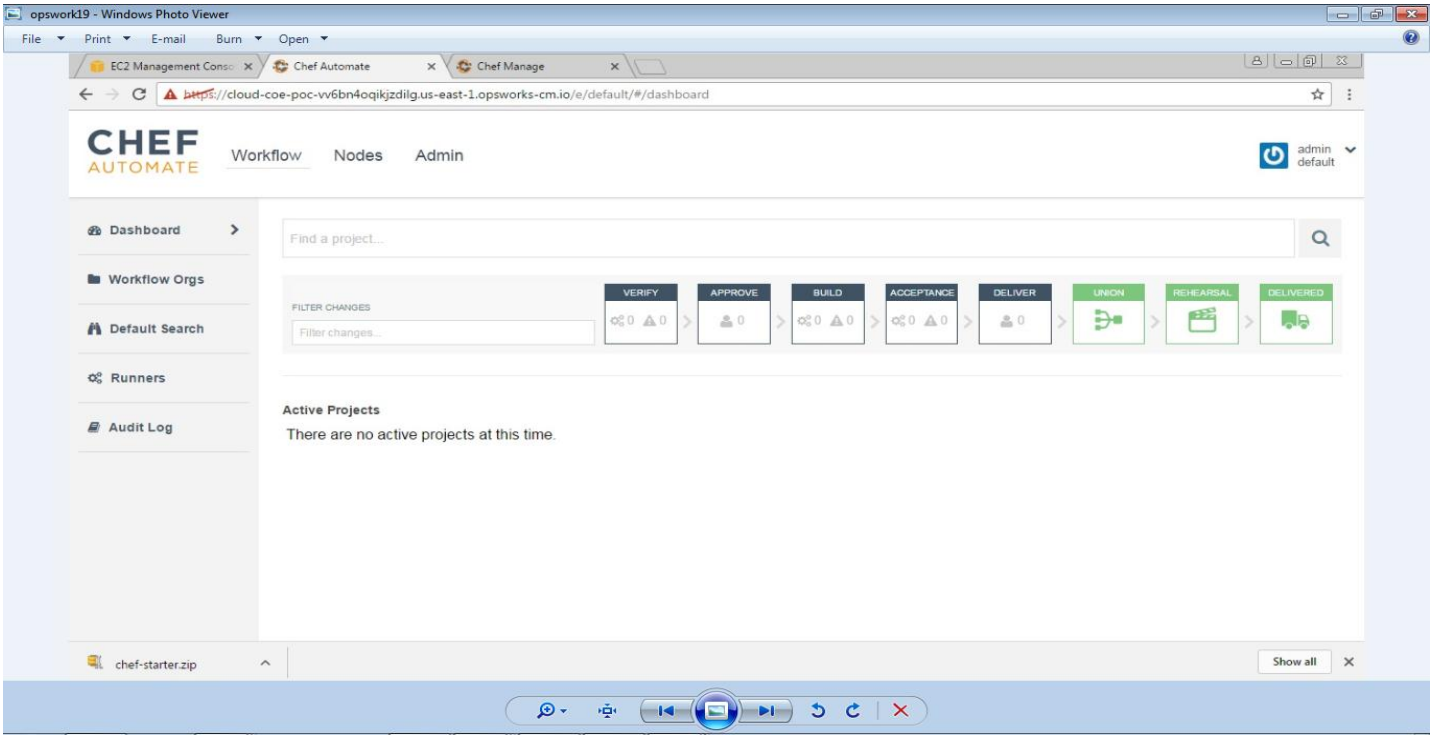
- Verify the ChefDK installation using chef commands on the Terminal



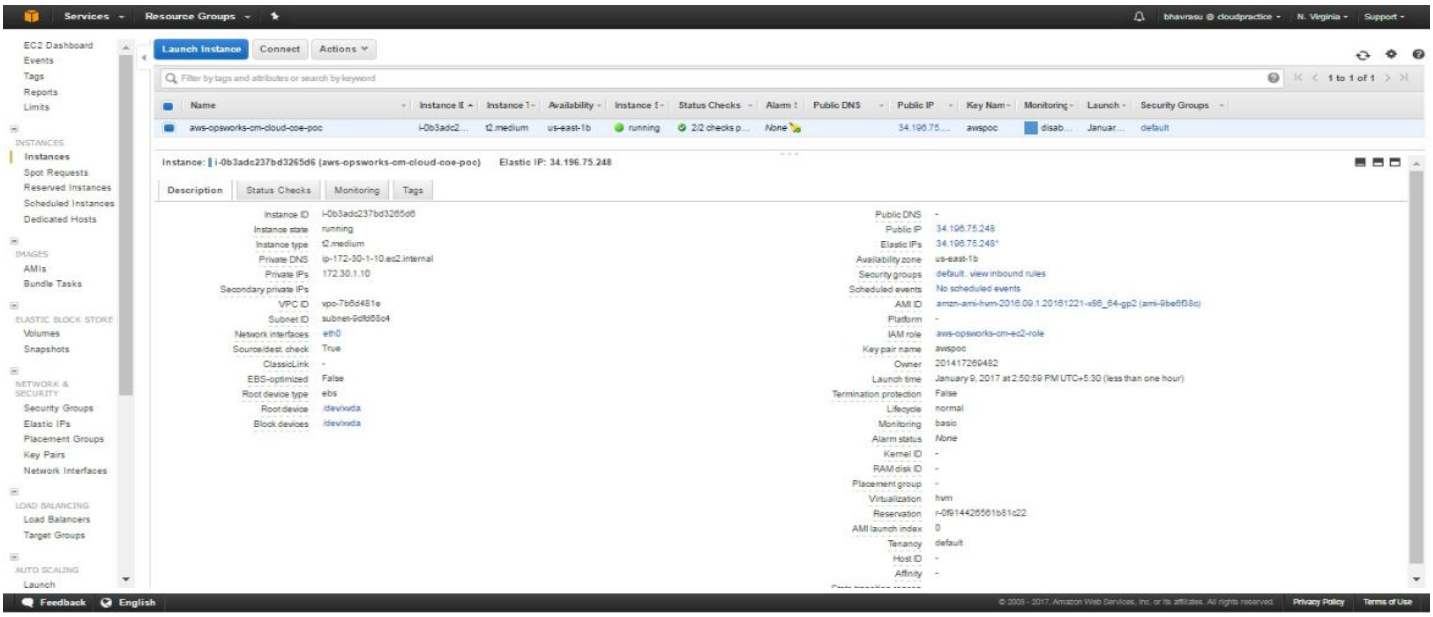
- Open the browser and copy paste the URL provided in the above (step 9) inside the server information section



- Open the browser and verify the Chef Automate on hosted Chef. Below screen shows default view



- Open AWS Management console to view the AWS Chef automation server instance details



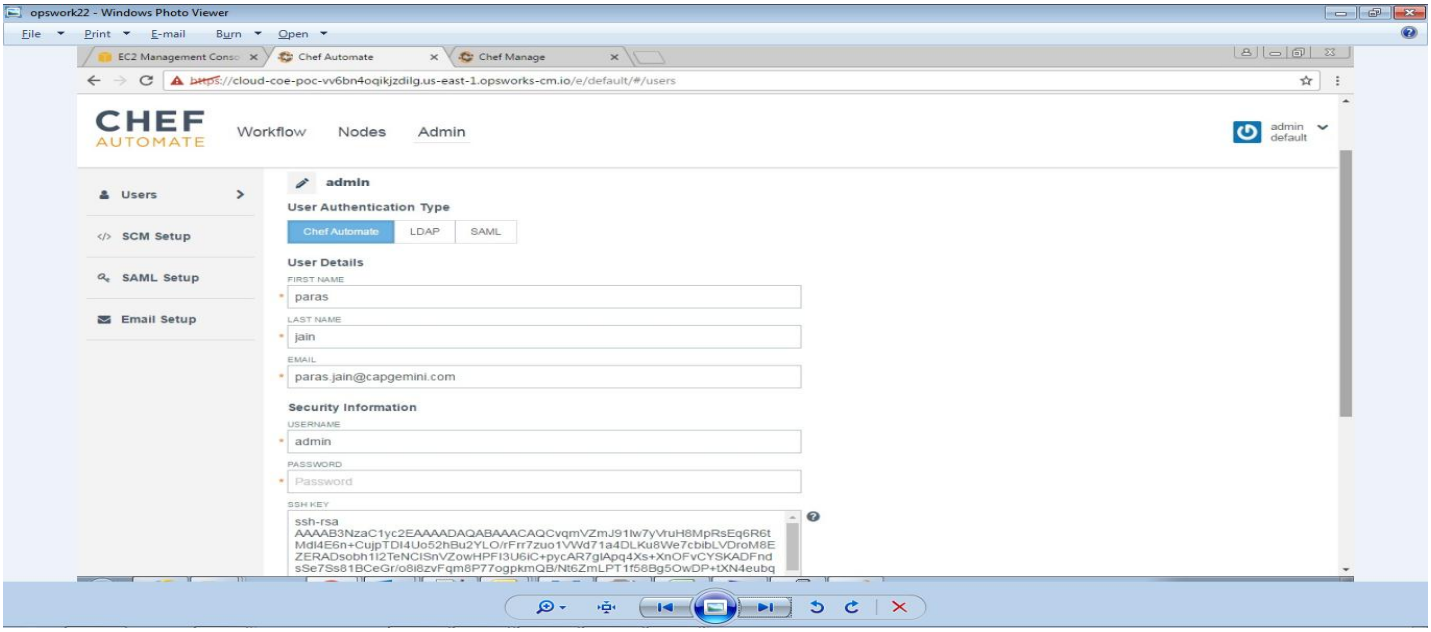
Note: You can integrate Chef Automate server using either Git / GitHub option

15.1.3.2 Integrating Git Repository with Chef Automate Server

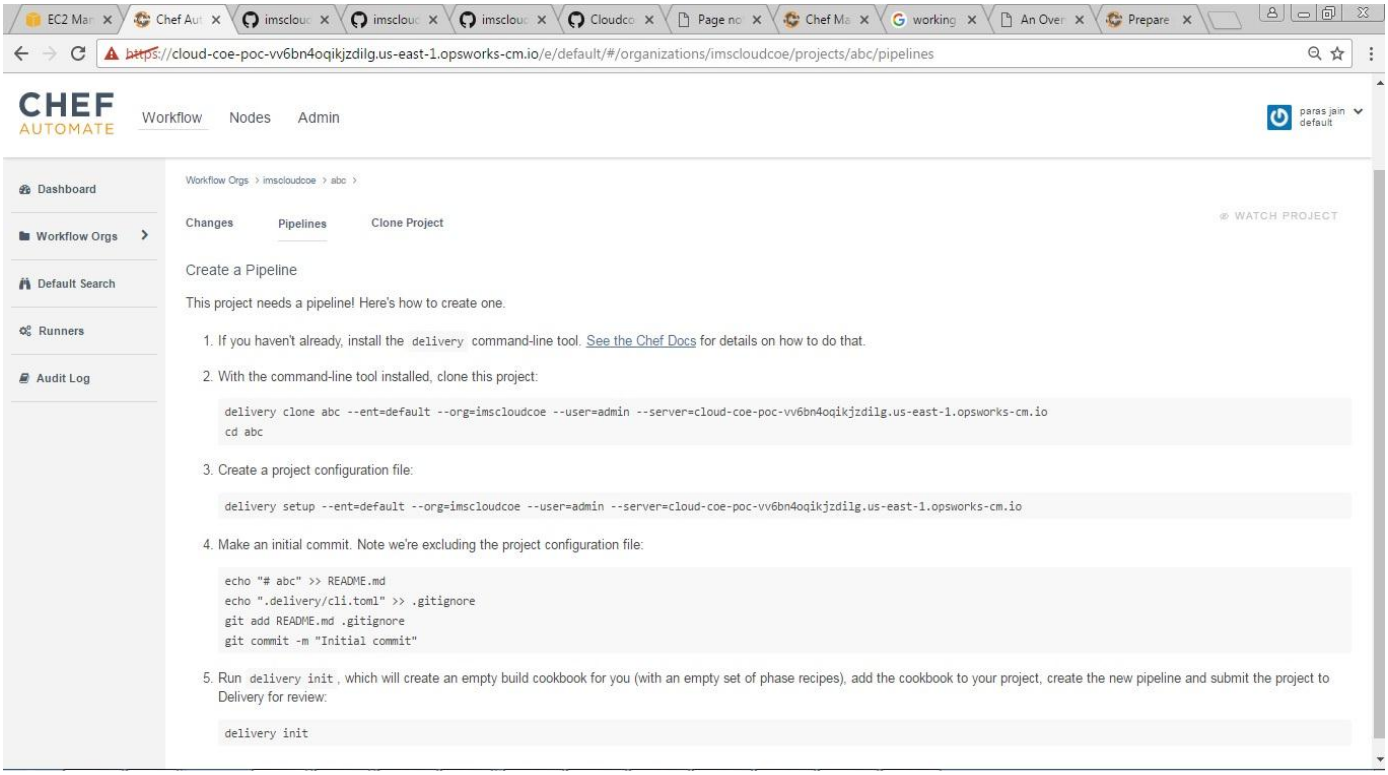
- Now install Git on the Chef automation server using following commands

  Commands

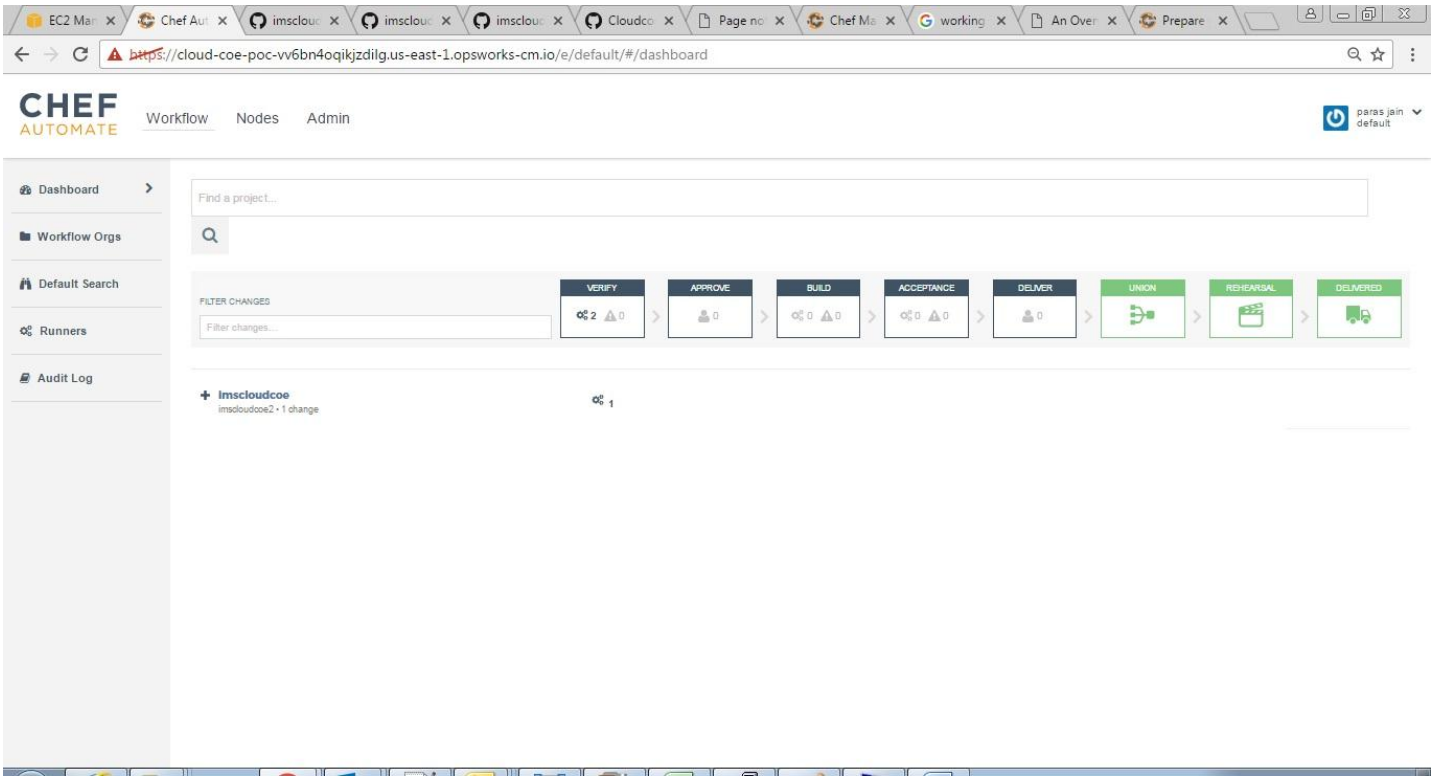
  - yum install git
  - git
  - delivery clone mytest --ent=default --org=imscloudcoe --user=admin --server=cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io
  - delivery
- On the Chef Automate server generate the rsa key as follows
  - ssh-keygen -t rsa -b 4096
  - cd /root/.ssh/
  - vim id_rsa.pub
  - cd /root/
  - vi id_rsa.pub
- Open the Chef Automate web UI, make an admin user and copy the generated rsa key details in the SSH-Key text box. This will connect the chef automate server to the Chef Automate web



- Execute the following commands in the chef automated server console



- After running the above commands, you will get the first pipeline (*imscloudcoe*) step as shown below on the Chef Automate GUI



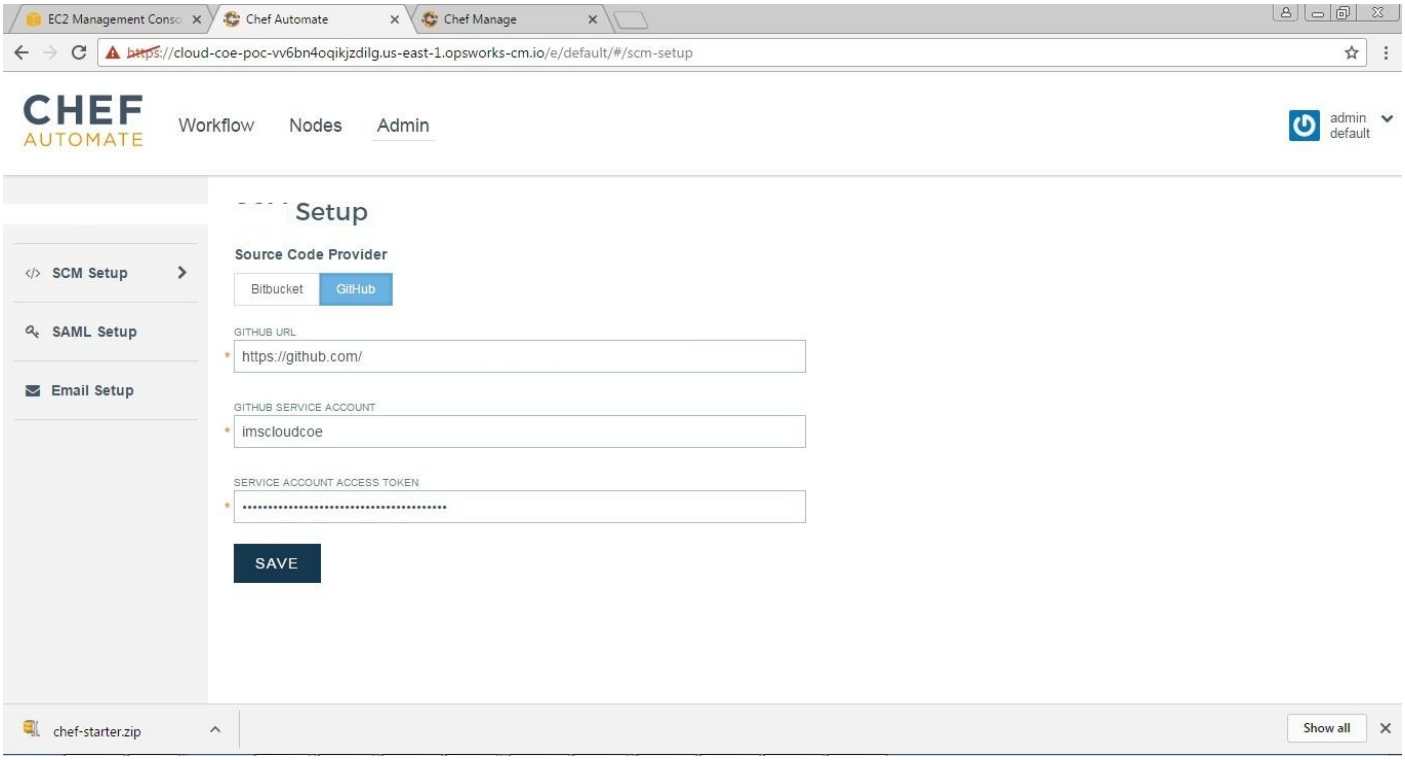
In this way, we can integrate the Git within the Chef Automate Server. We can also integrate the hosted GitHub with Chef automate Server as shown below.

GitHub Access

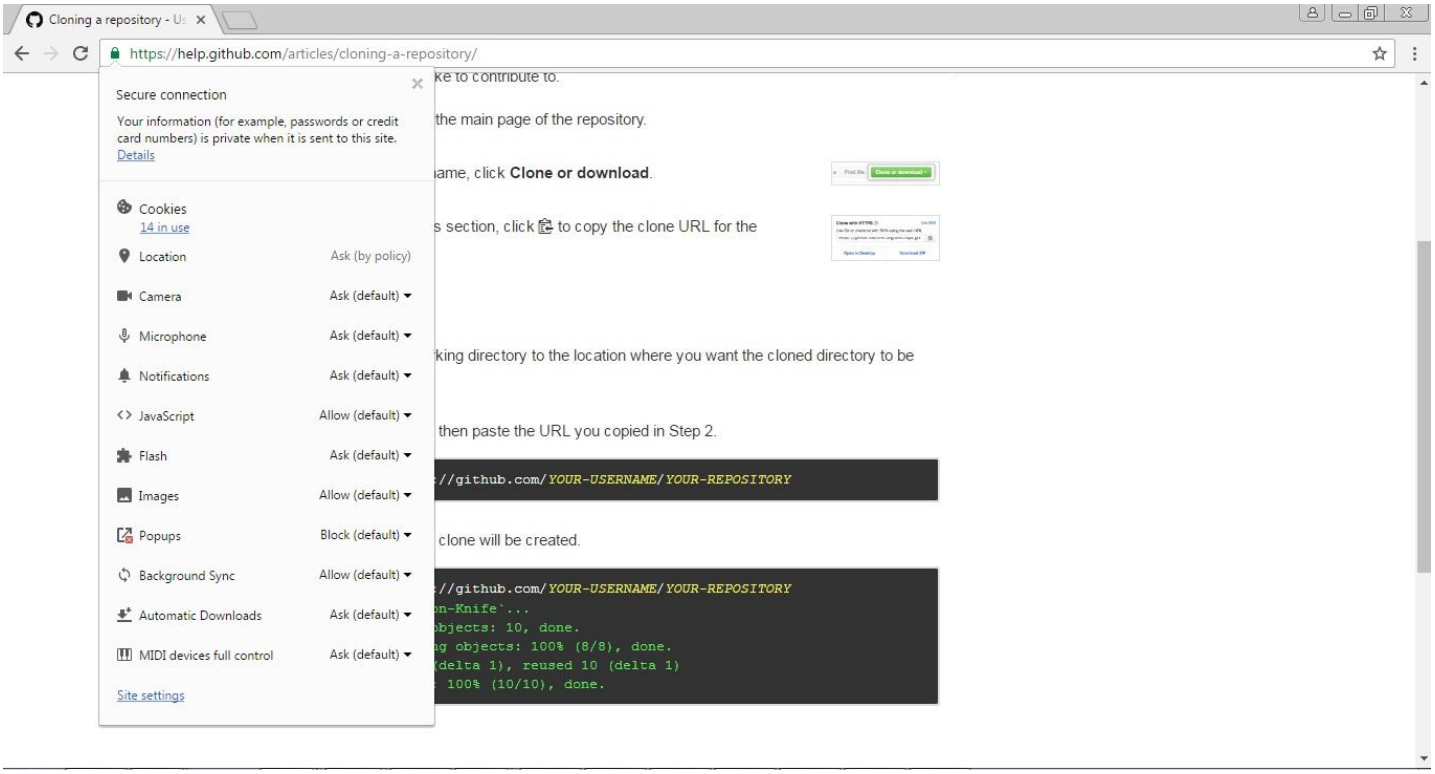
- GitHub Details
- Token Generation
- Sample Project

15.1.3.1 Integrating GitHub Repository (Hosted) with Chef Automate Server

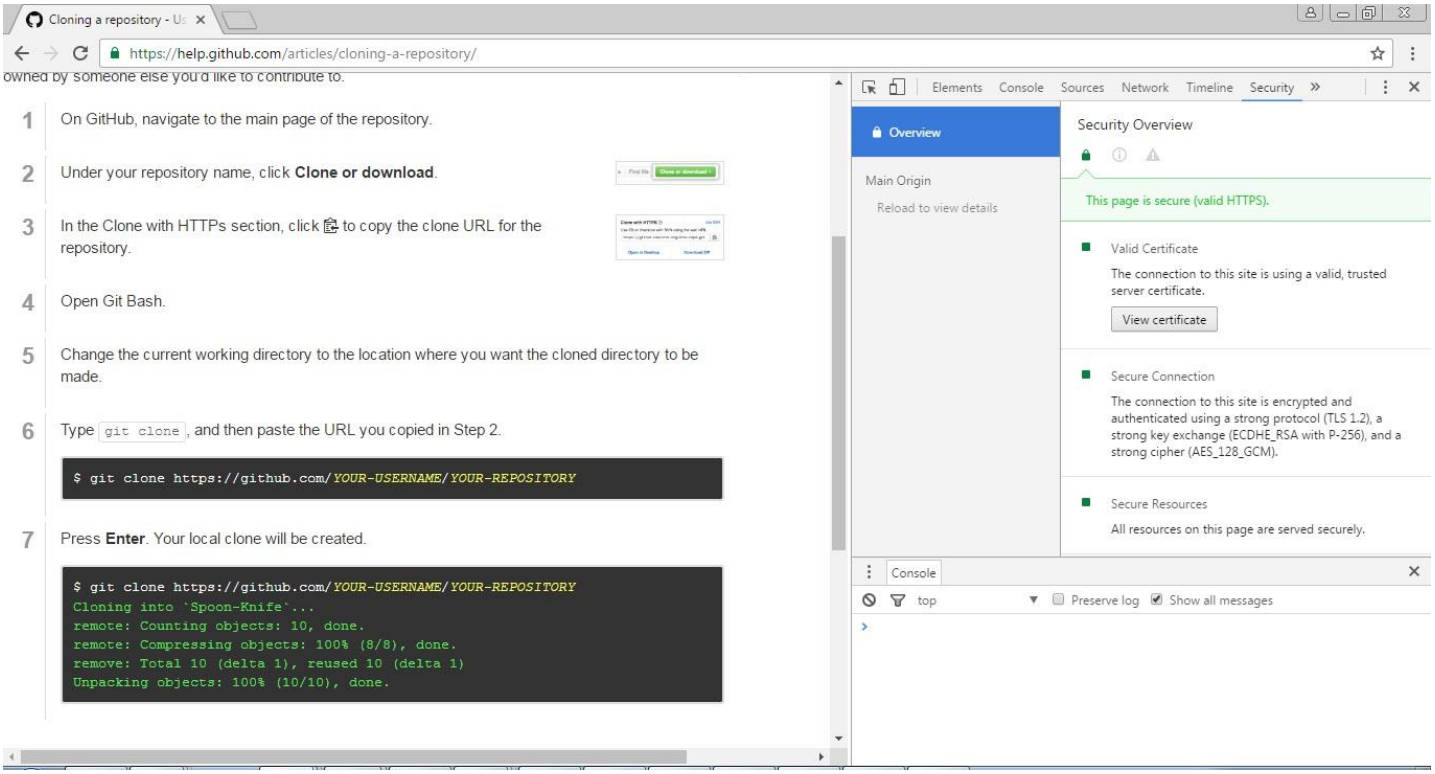
- Go to Chef Automate web UI, Select **Admin** tab, click on SCM setup from the left panel, select GitHub tab and give your GitHub details as shown below



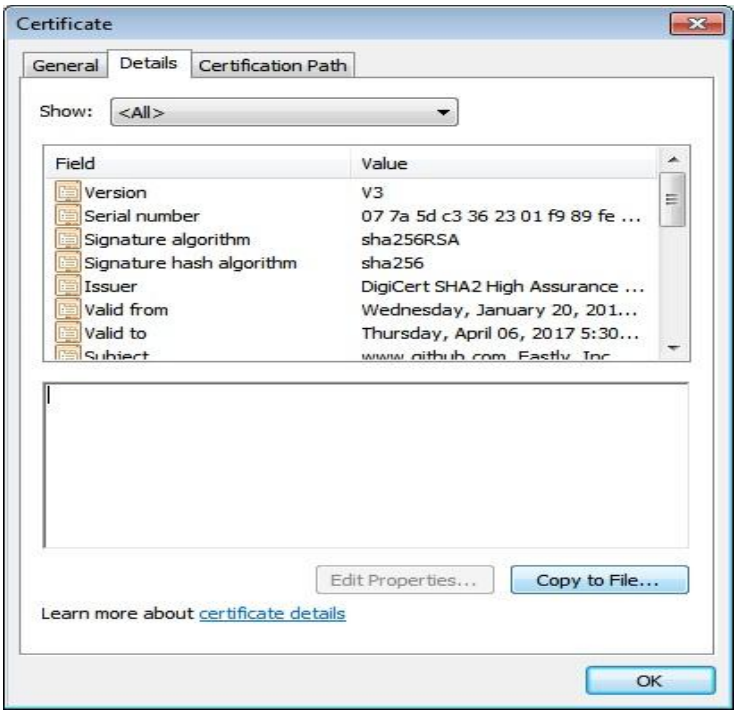
- Click on Save button
- Login to GitHub, download the certificate from your GitHub.
- On the URL menu, Click on lock button (before https) a drop down menu will appear click on Details link.



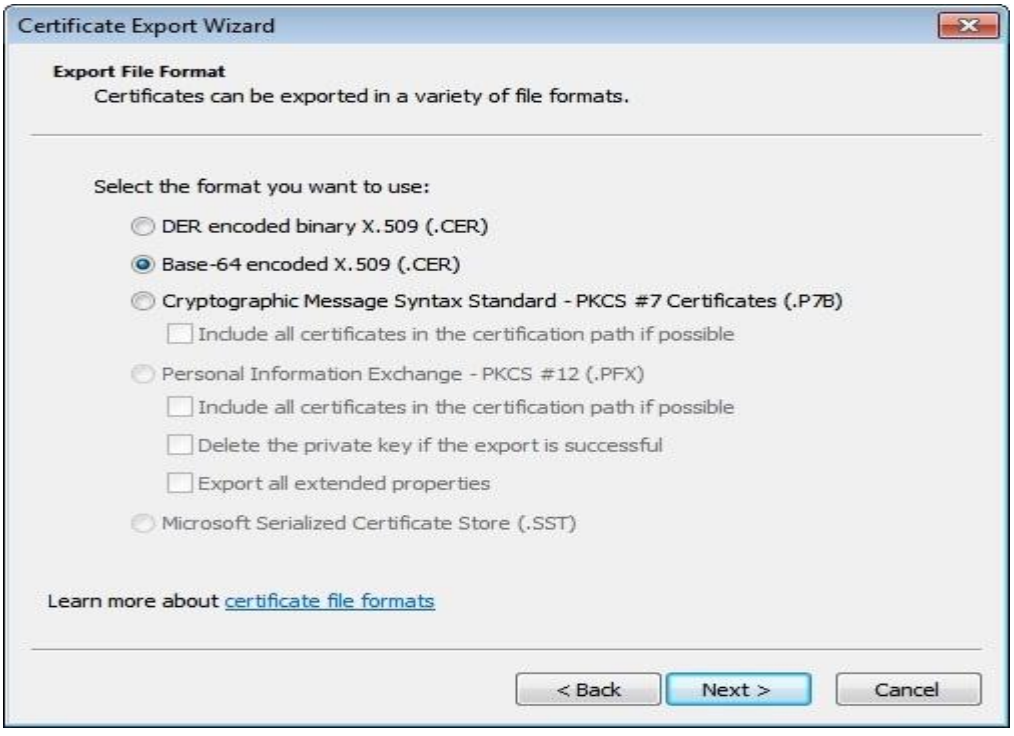
- Click on View certificate button



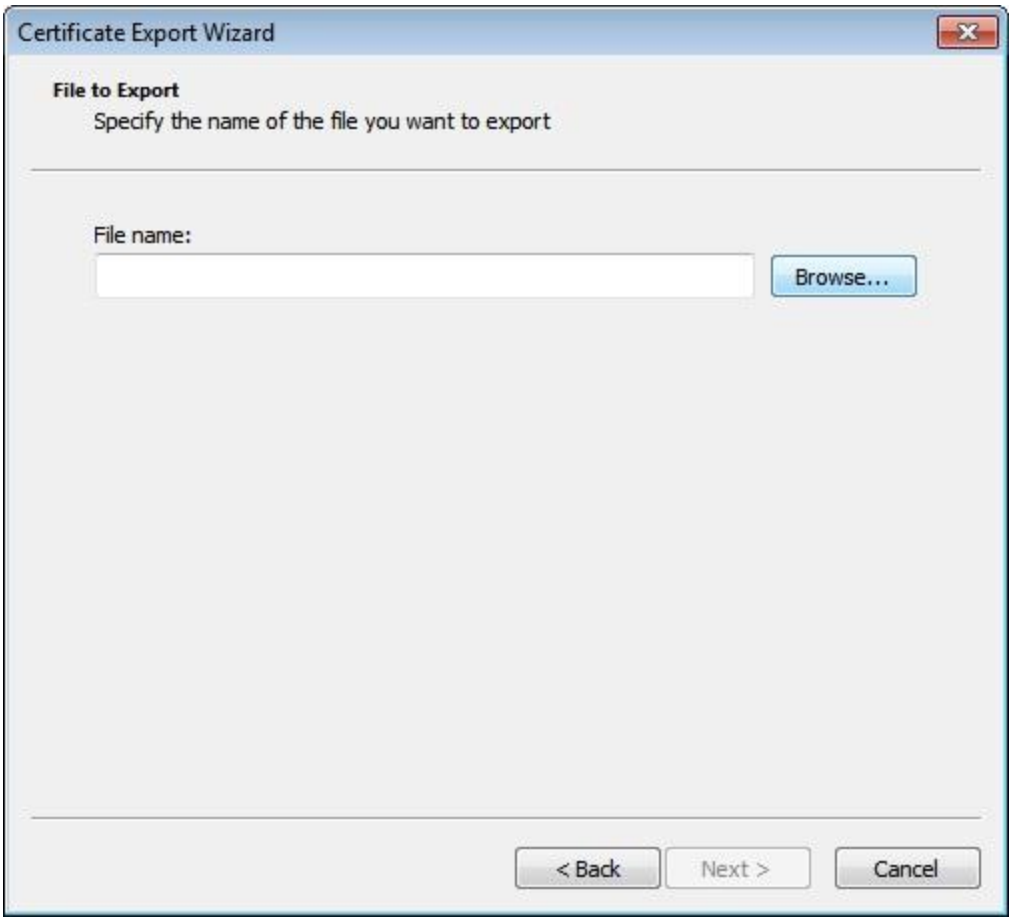
- Now click on **Copy to file** button



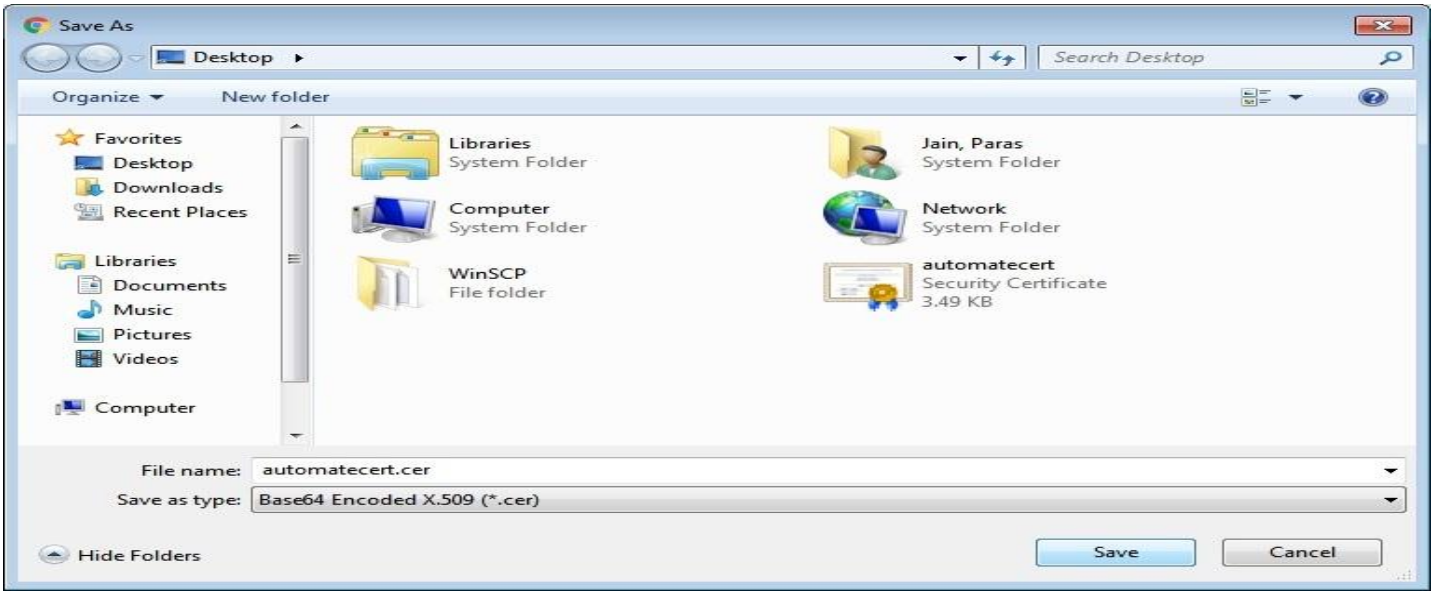
- Select **Base-64 encoded X.509 (.CER)** radio button , click on **Next** button



- Click on **Browse** button and provide the location where you want to save it



- Provide the **Filename** and click on **Save** Button



- Now move the certificate to the following location in chef automate server **/etc/pki/ca-trust/source/anchors/**

- Login into your chef automate server as a root user and run the following commands

  yum install ca-certificates

  update-ca-trust force-enable

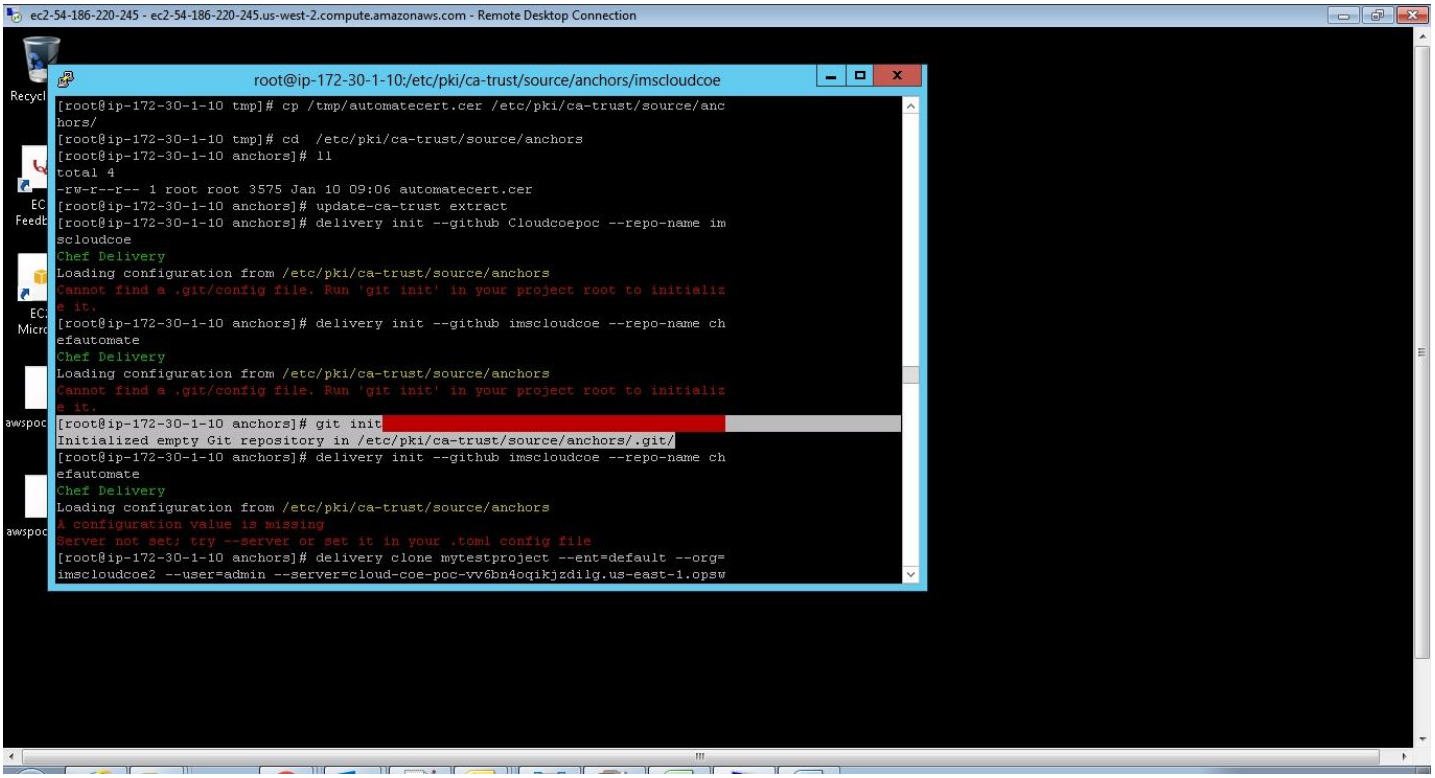
  cd /etc/pki/ca-trust/source/anchors/

  update-ca-trust extract

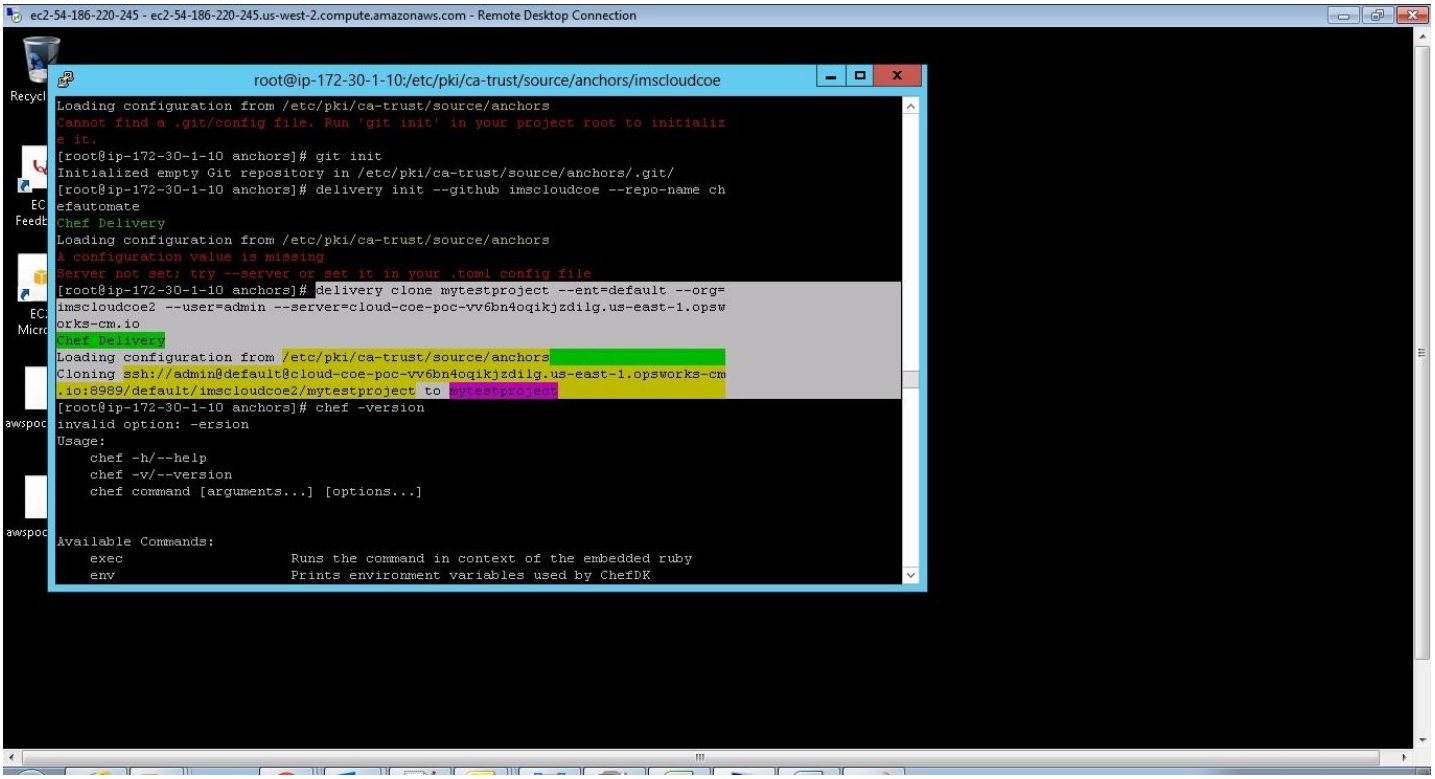
- Create a new GitHub-integrated project, the project repository in GitHub must have at least one commit

- Run the following command in chef automate server

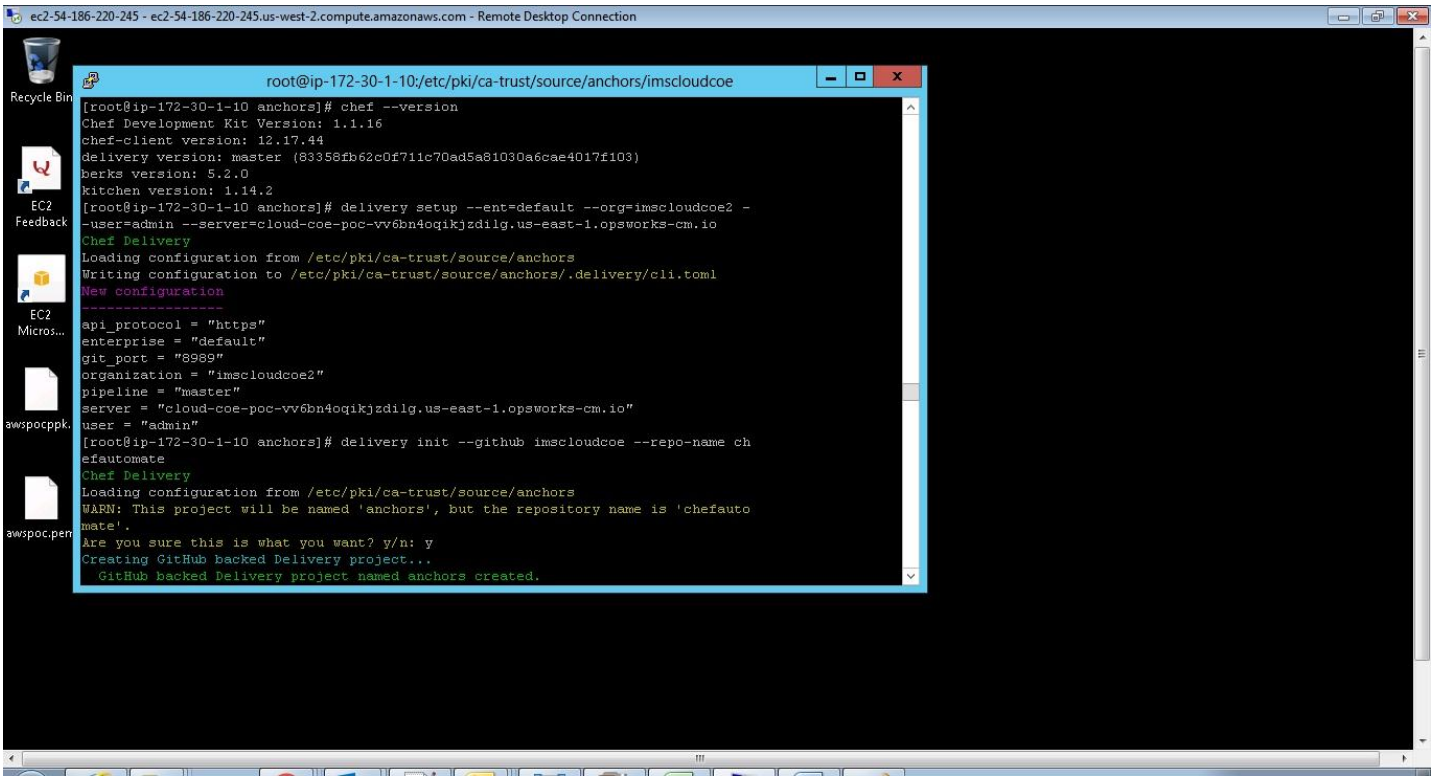
  git init



  delivery clone mytestprotect --ent=default --org=imscloudcoe2 --user=admin --server=cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io

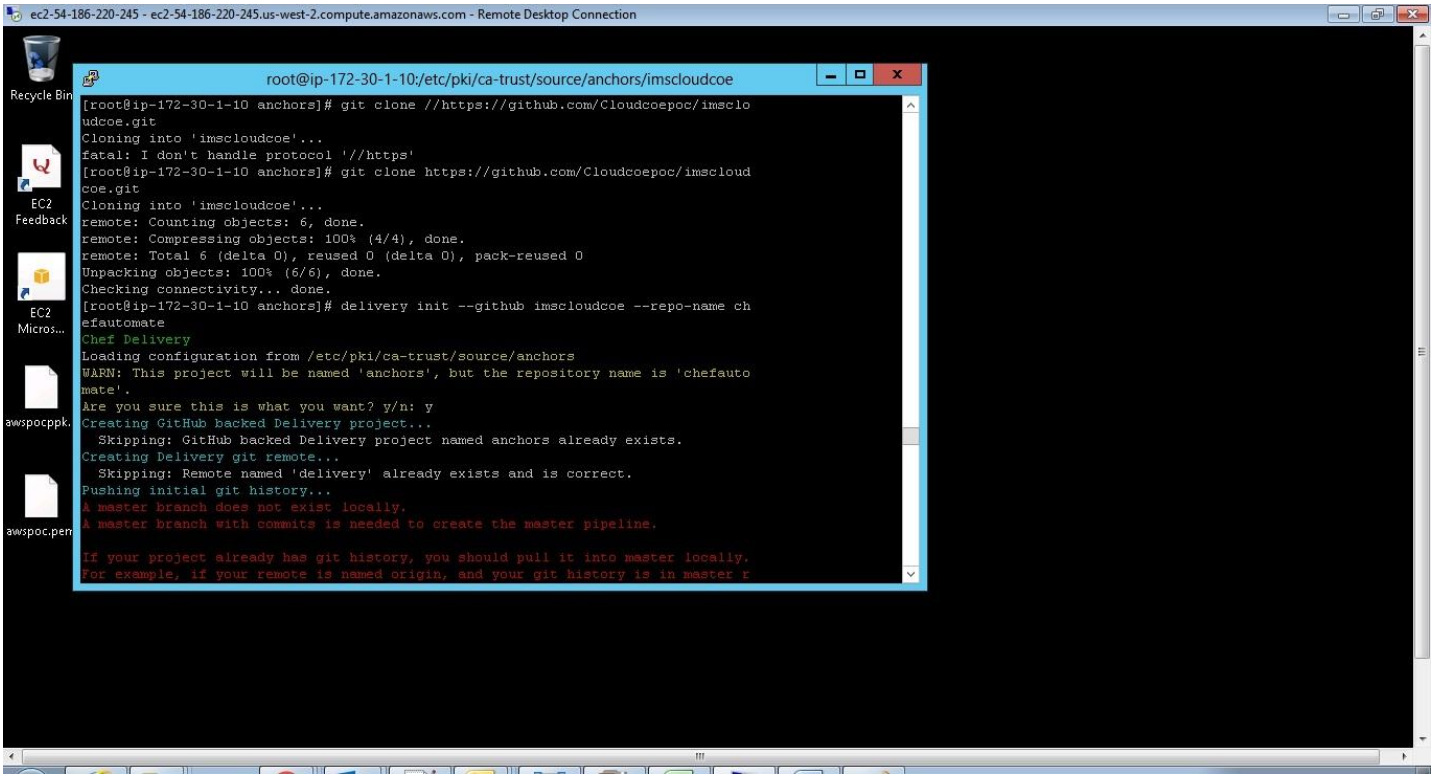


delivery setup --ent=default --org=imscloudcoe2 --user=admin --server=cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io



git clone https://github.com/Cloudcoepoc/imscloudcoe.git

delivery init --github Cloudcoepoc --repo-name imscloudcoe

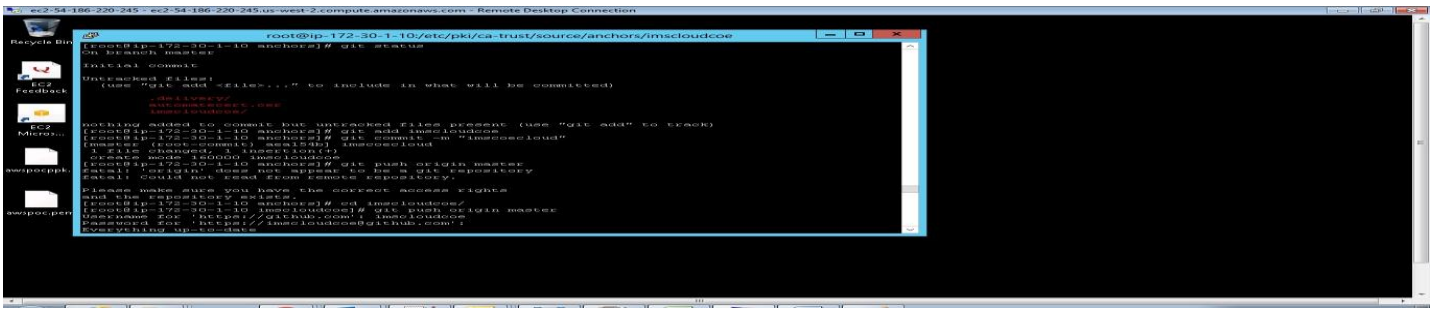


git status

git add imscloudcoe

git commit -m "imscoecloud"

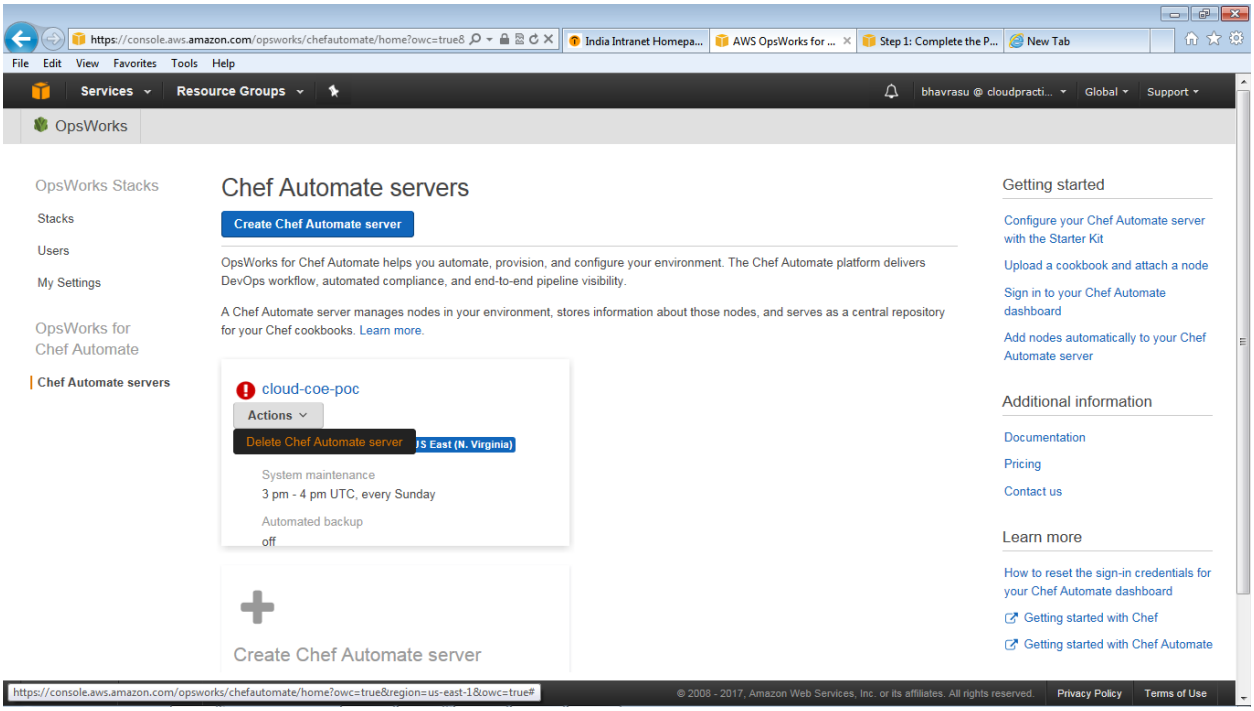
git push origin master

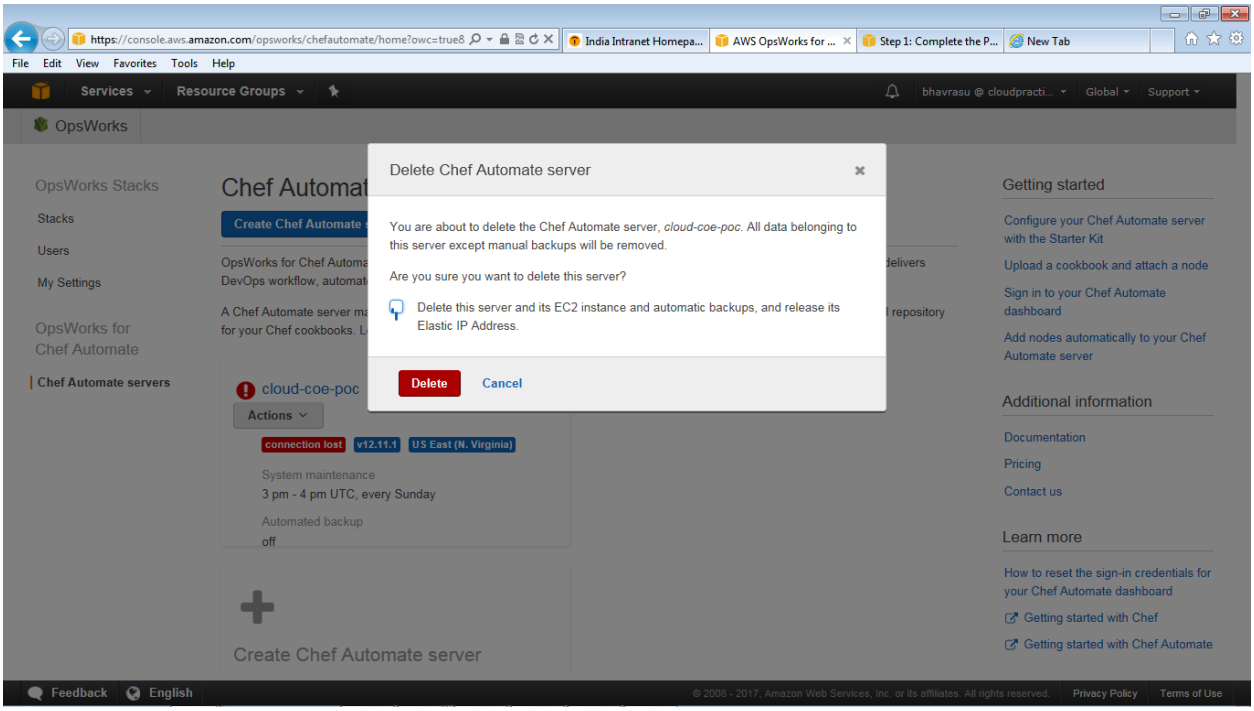


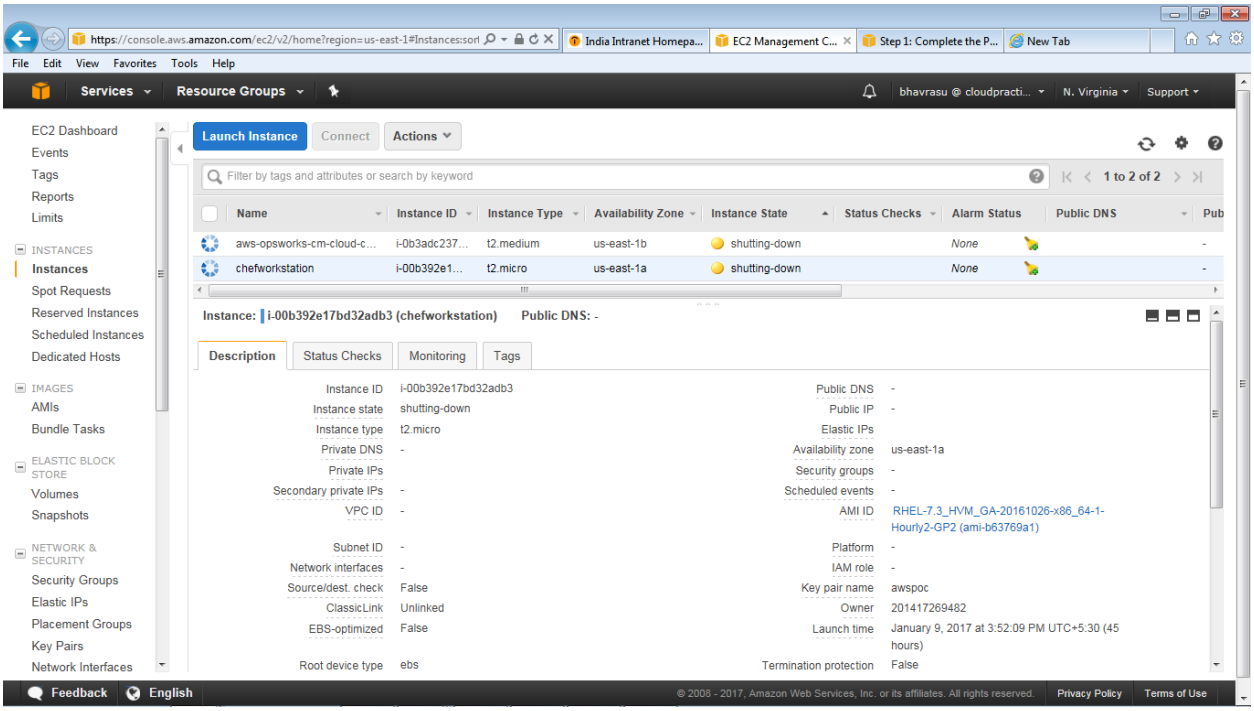
git status

Files are pushed to Chef Automated Server

15.1.3.4 Delete Chef Automate Server







15.2 AWS OpsWork Stacks

15.2.1 Objective

Understand the process to create AWS OpsWork Stacks

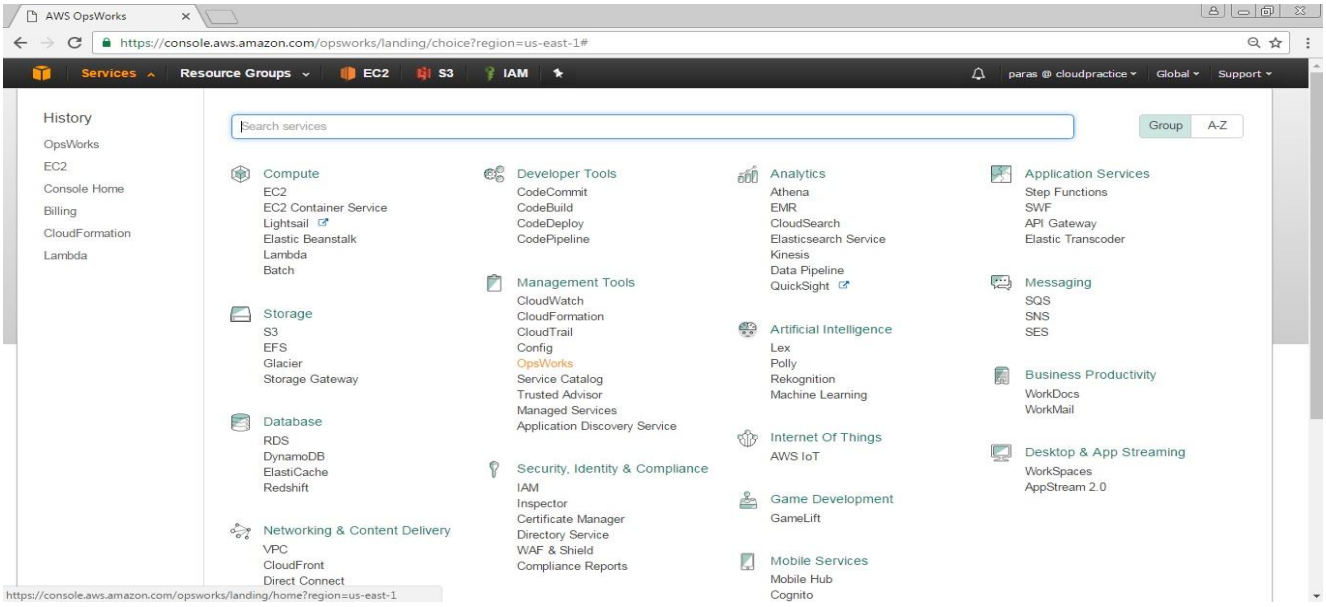
15.2.2 Pre-requisite

AWS account, VPC configuration, IAM user access (attach the AWSOpsWorksFullAccess and AmazonS3FullAccess)

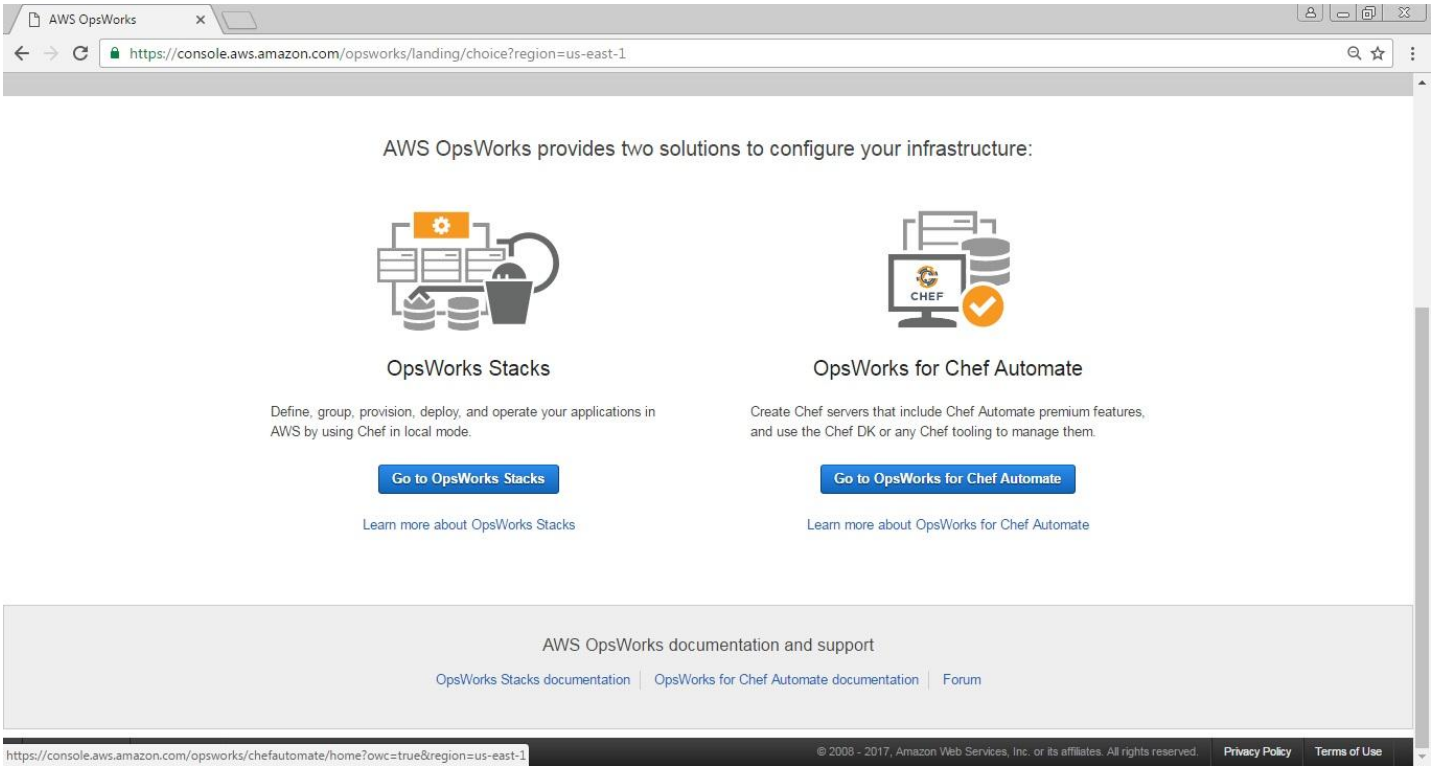
15.2.3 Procedure

15.2.3 .1 Create a Create a Stack

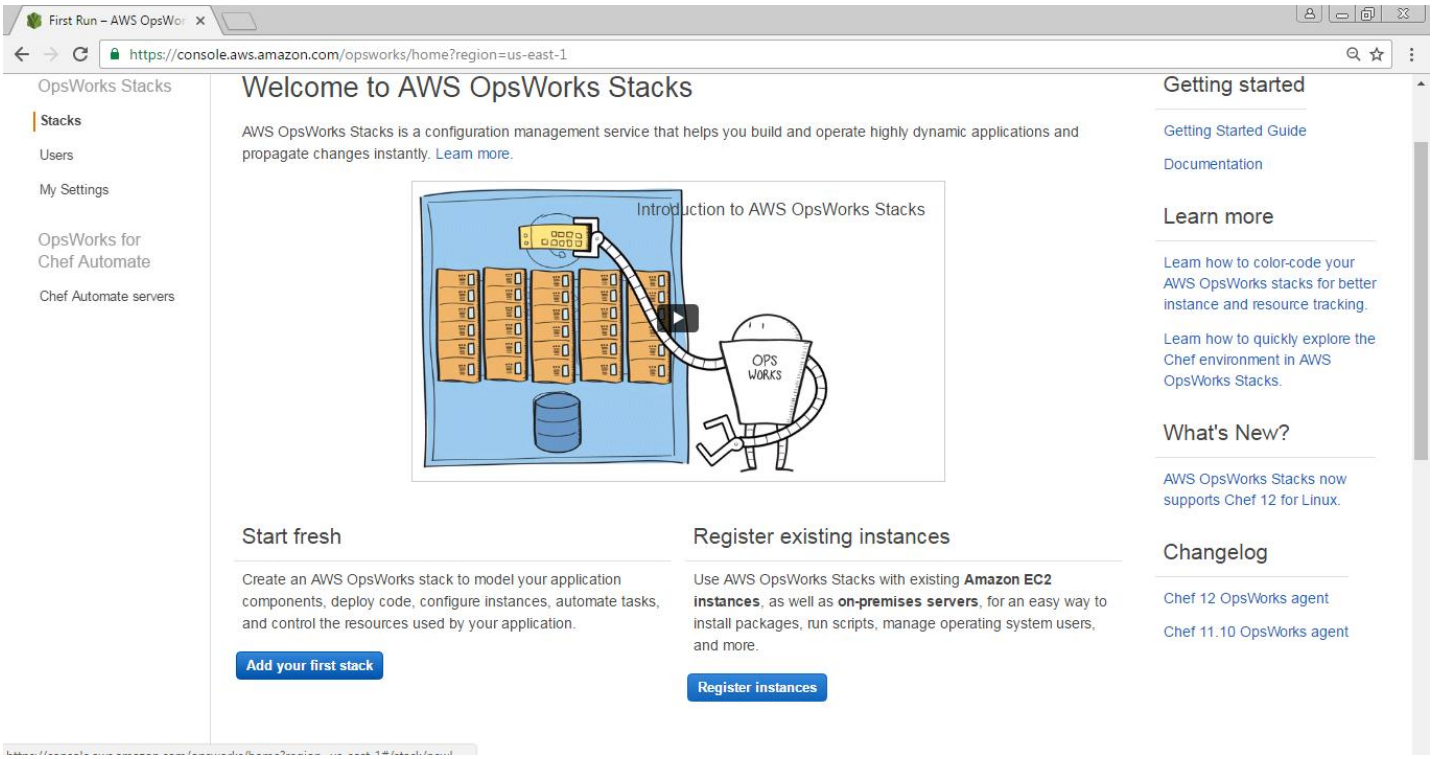
- Open the AWS Management console
- Click on **OpsWorks** under **Services**.



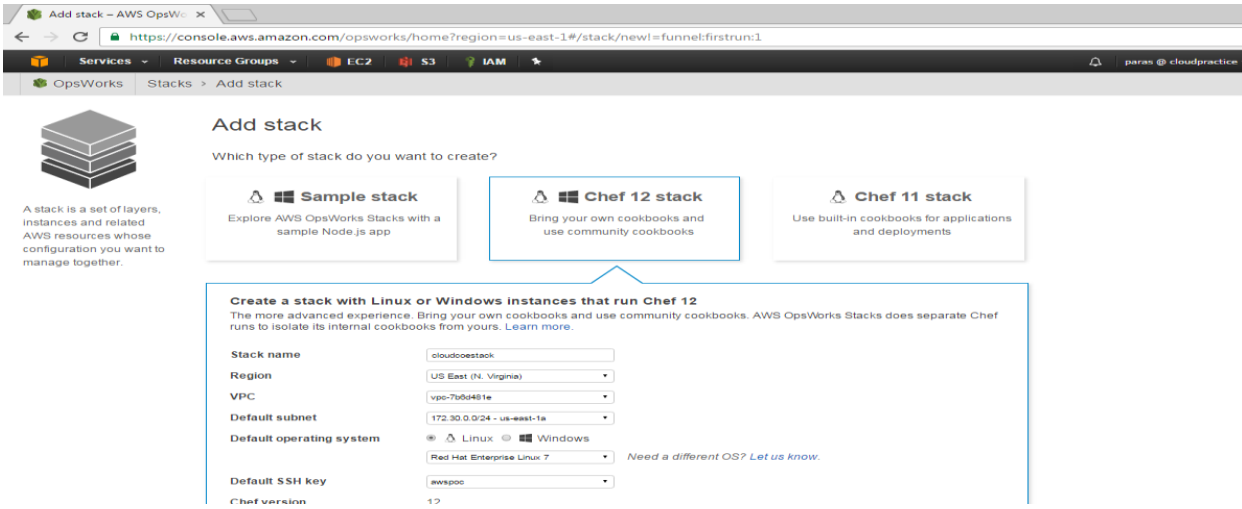
- From **AWS OpsWork** console, click on **Go to OpsWorks Stacks** button.



- Click on **Add your first stack** button



- With the **Add stack** page displayed, choose **Chef 12 stack** if it is not already chosen for you.



In the **Chef 12 stack** drop down box do the following configuration:

In the **Stack name** box provide type the name, for example **cloudcoe**

For **Region** choose US East (N. Virginia)

For **VPC:** If a VPC is available, choose it else choose No VPC

For **Default subnet** choose any subnet available in your VPC

For **Default operating system**, choose **Linux** and from the drop down list choose **Red Hat Enterprise Linux 7**

Default **SSH key**, choose **Do not use a default SSH key**

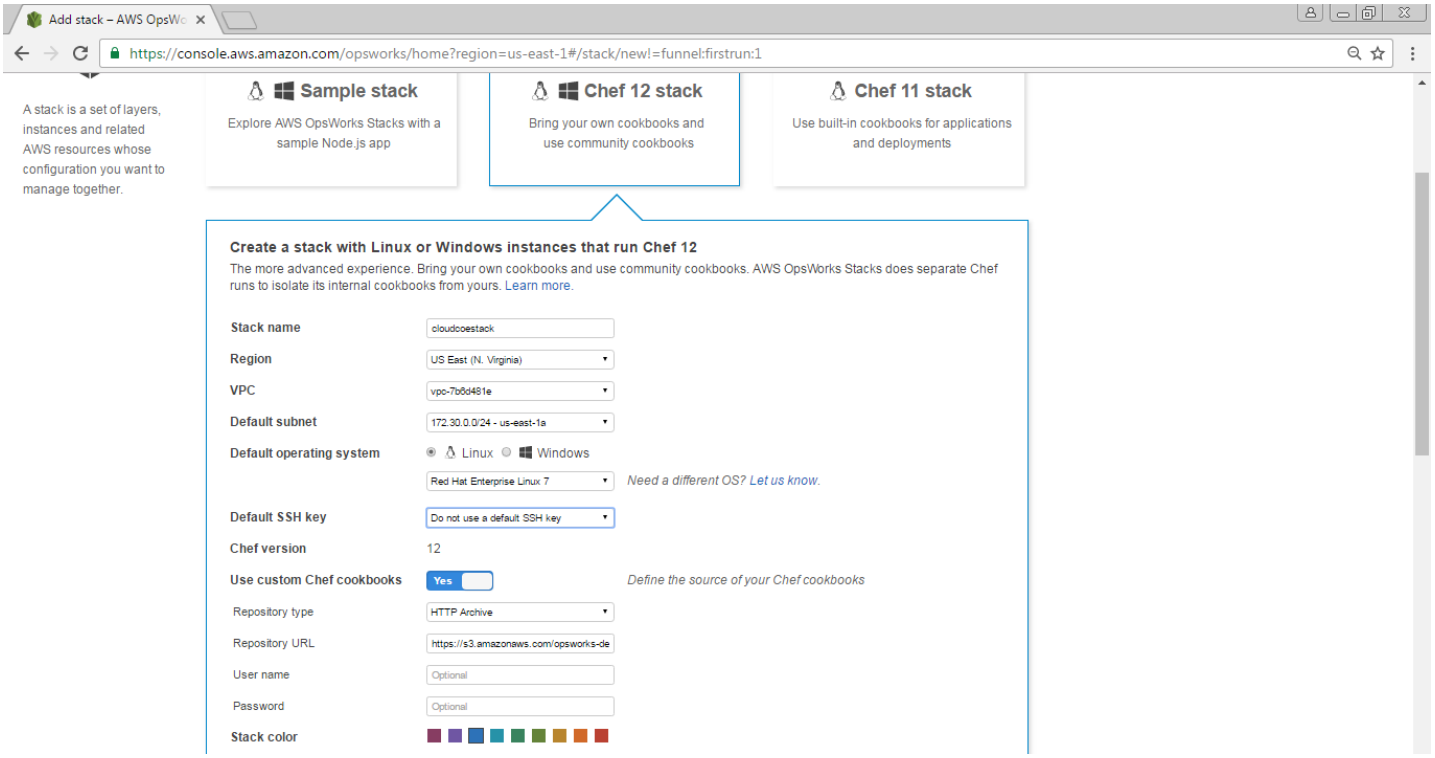
For **Use custom Chef cookbooks**, choose **Yes**

For **Repository type**, choose **Http Archive**

For **Repository URL**, type *https://s3.amazonaws.com/opsworks-demo-assets/opsworks-linux-demo-cookbooks-nodejs.tar.gz*

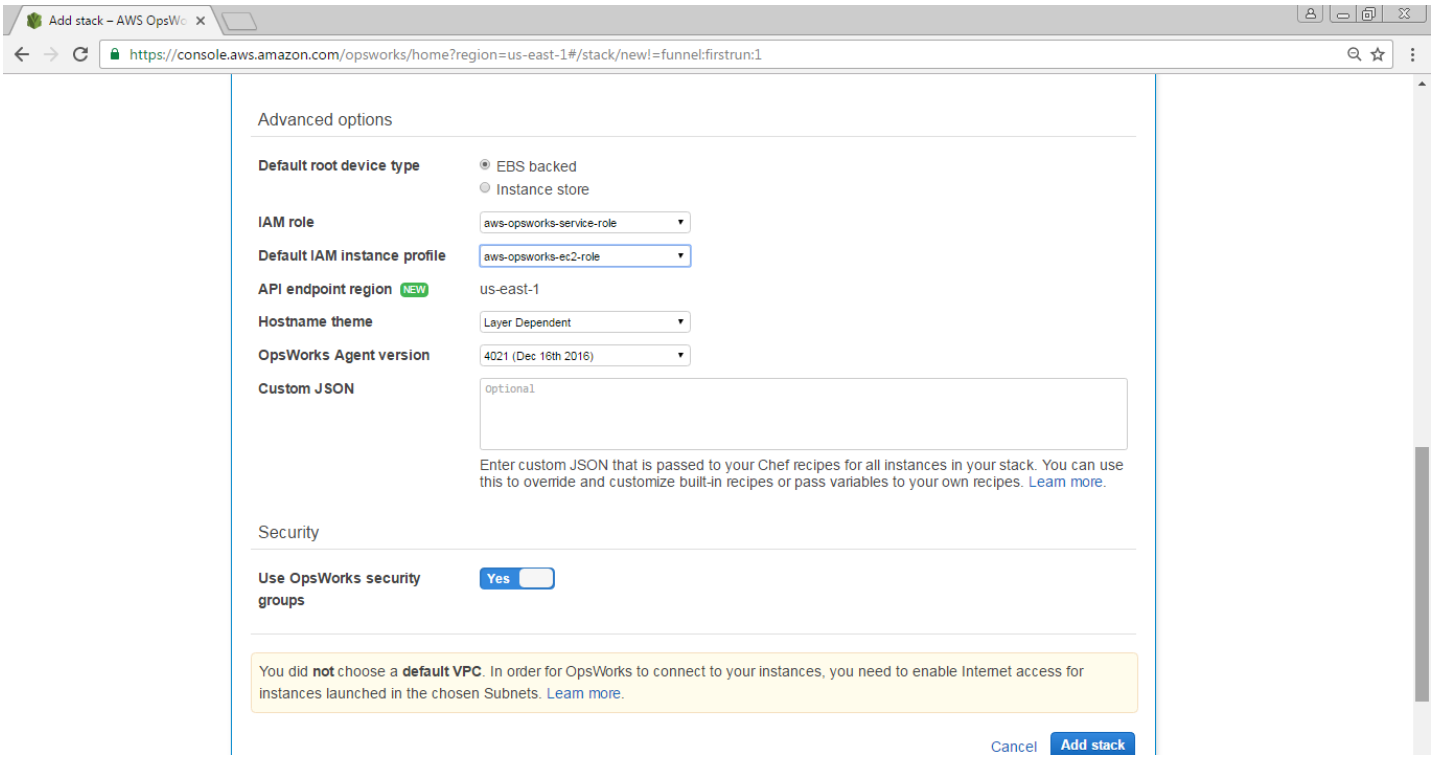
For **User name** and **Password** leave the textbox blank

For **Stack color** choose **dark blue**



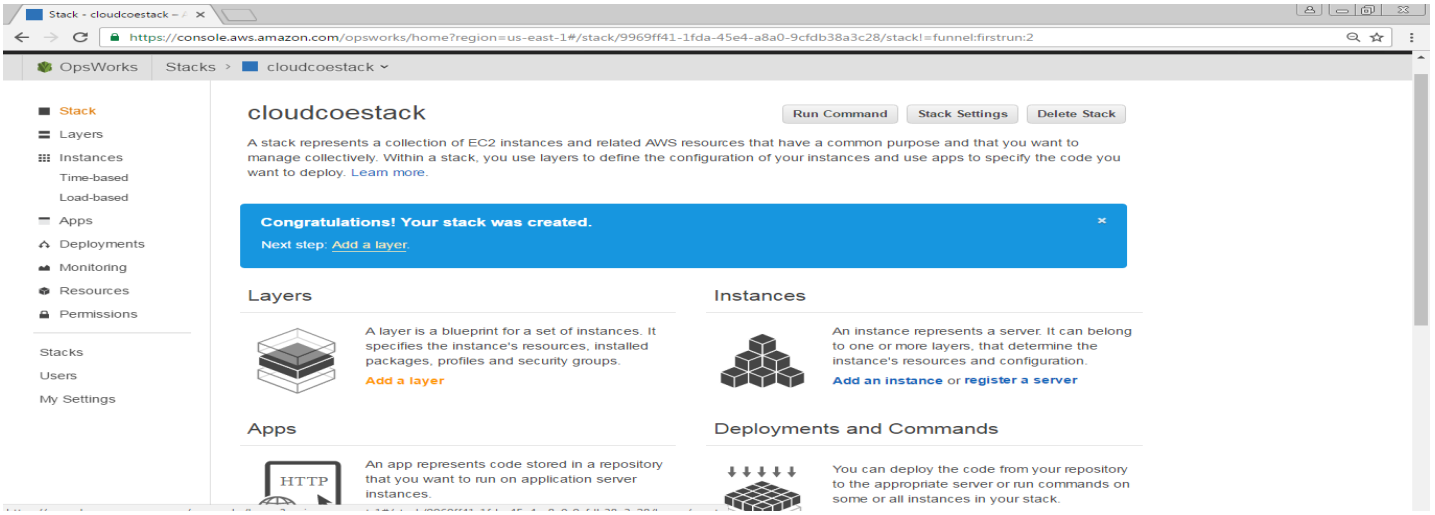
- Choose **Advanced Options,** do the following
    - ✓ For **Default root device type** select **EBS backed**
    - ✓ For **IAM role**: If **aws-opsworks-service-role** is available choose it else choose **New IAM role**
    - ✓ For **Default IAM instance profile:** If **aws-opsworks-ec2-role** is available, choose it else choose **New IAM instance profile**
    - ✓ For **Hostname theme** select **Layer Dependent**
    - ✓ For **OpsWorks Agent version** select most recent version
    - ✓ For **Custom JSON** leave textbox blank
    - ✓ For **Use OpsWorks security groups** select **Yes**

- Click on **Add Stack** button

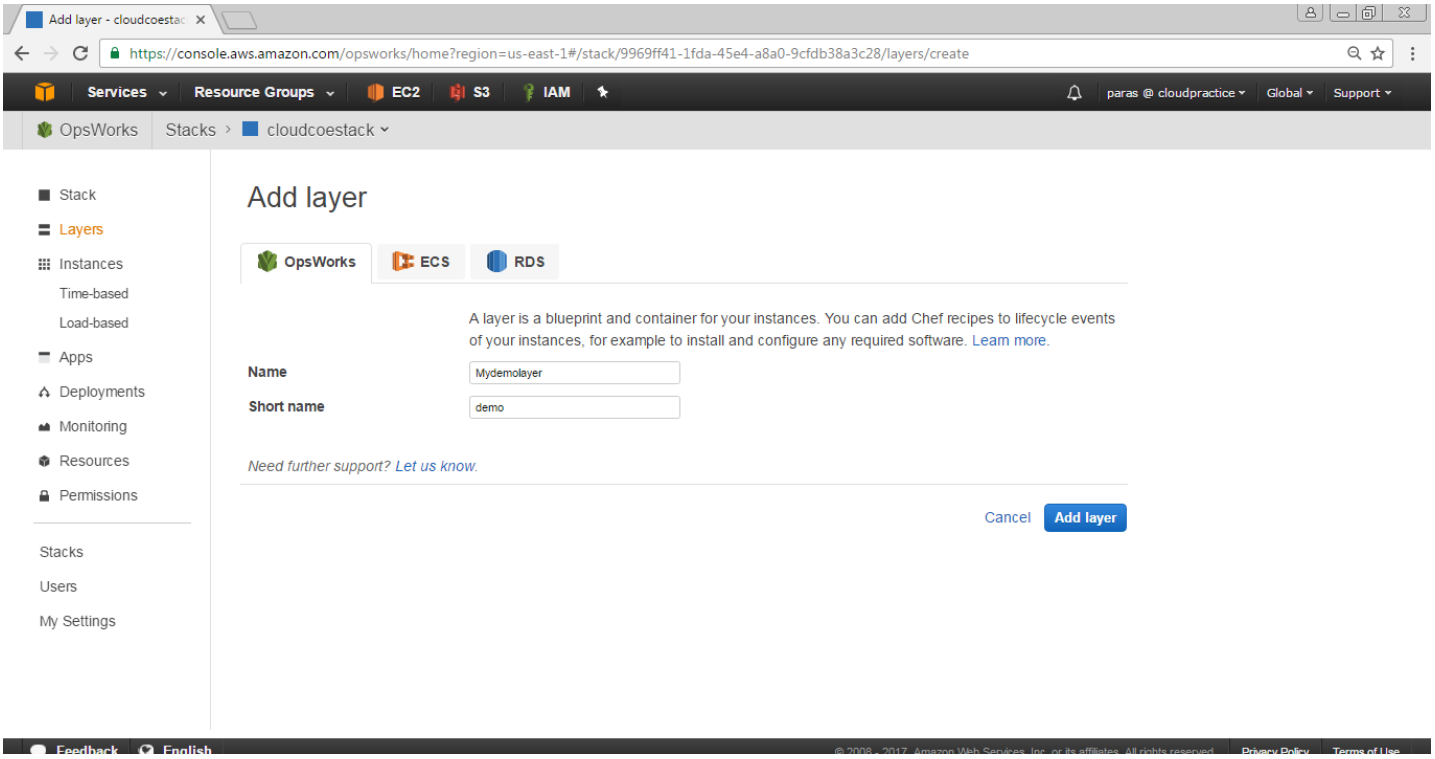


15.2.3.2 Add a Layer to the Stack

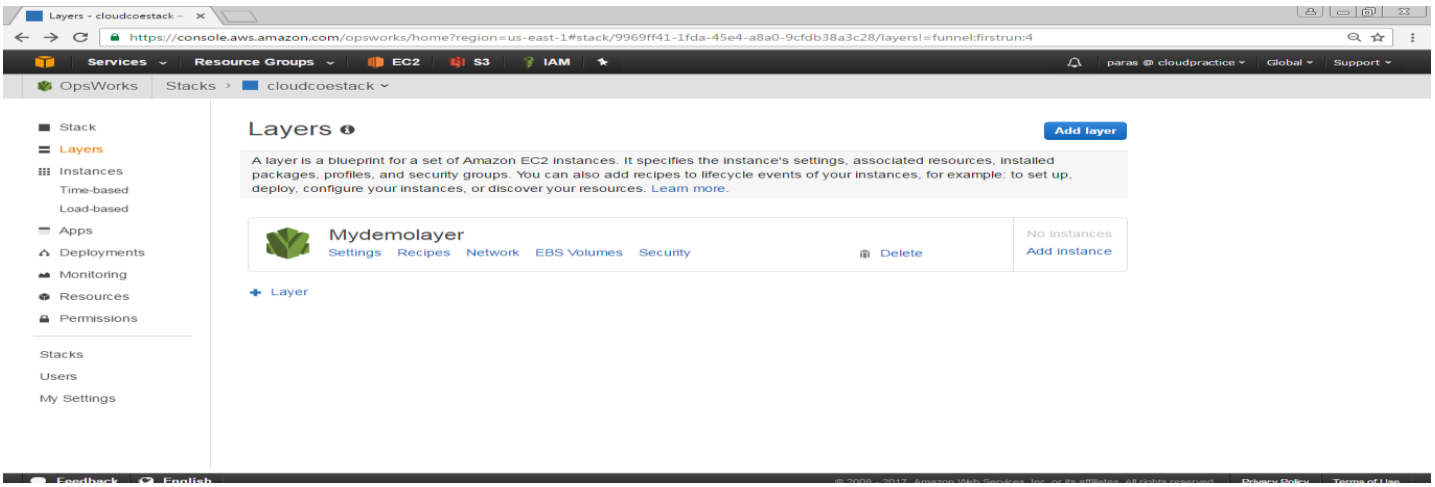
15 You will get following screen showing cloudcoestack, click Add a layer



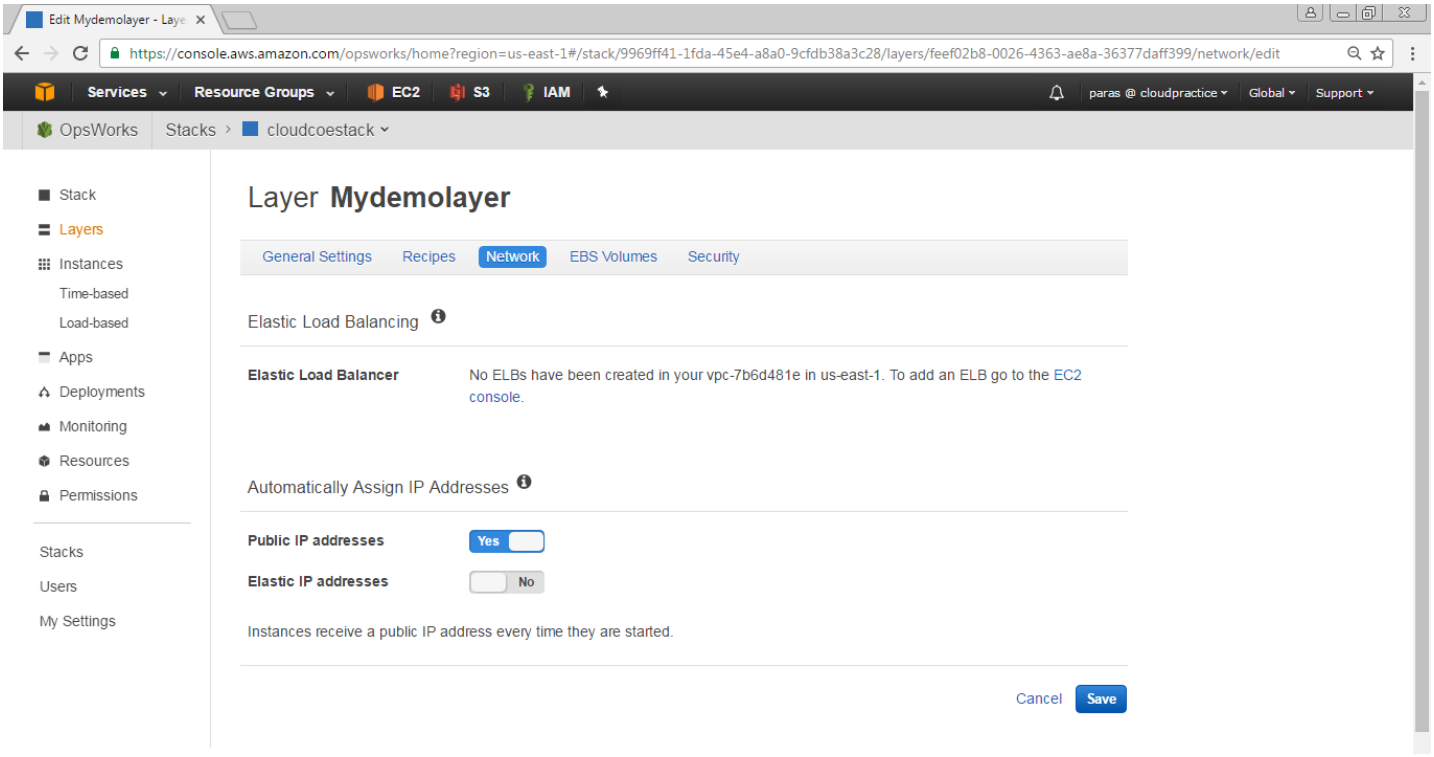
16 The Add Layer page is displayed. On the OpsWorks tab, for Name, type **Mydemolayer.** For **Short name,** type **demo** and click on **Add layer**



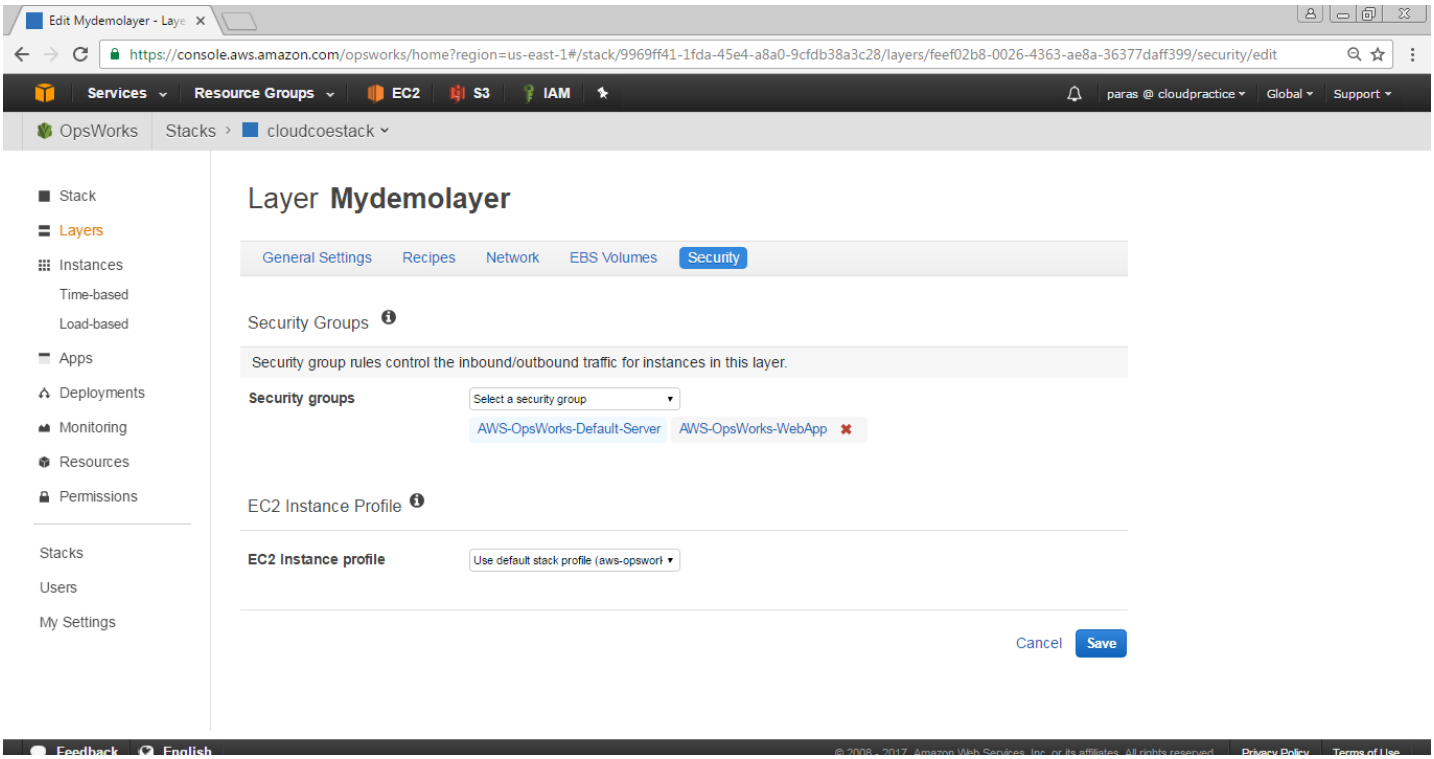
17 On the Layers page, for **MyDemoLayer**, choose **Network**



18 On the **Network** tab, under **Automatically Assign IP Addresses**, verify that **Public IP addresses** is set to yes. If you've made changes, click on **Save** button

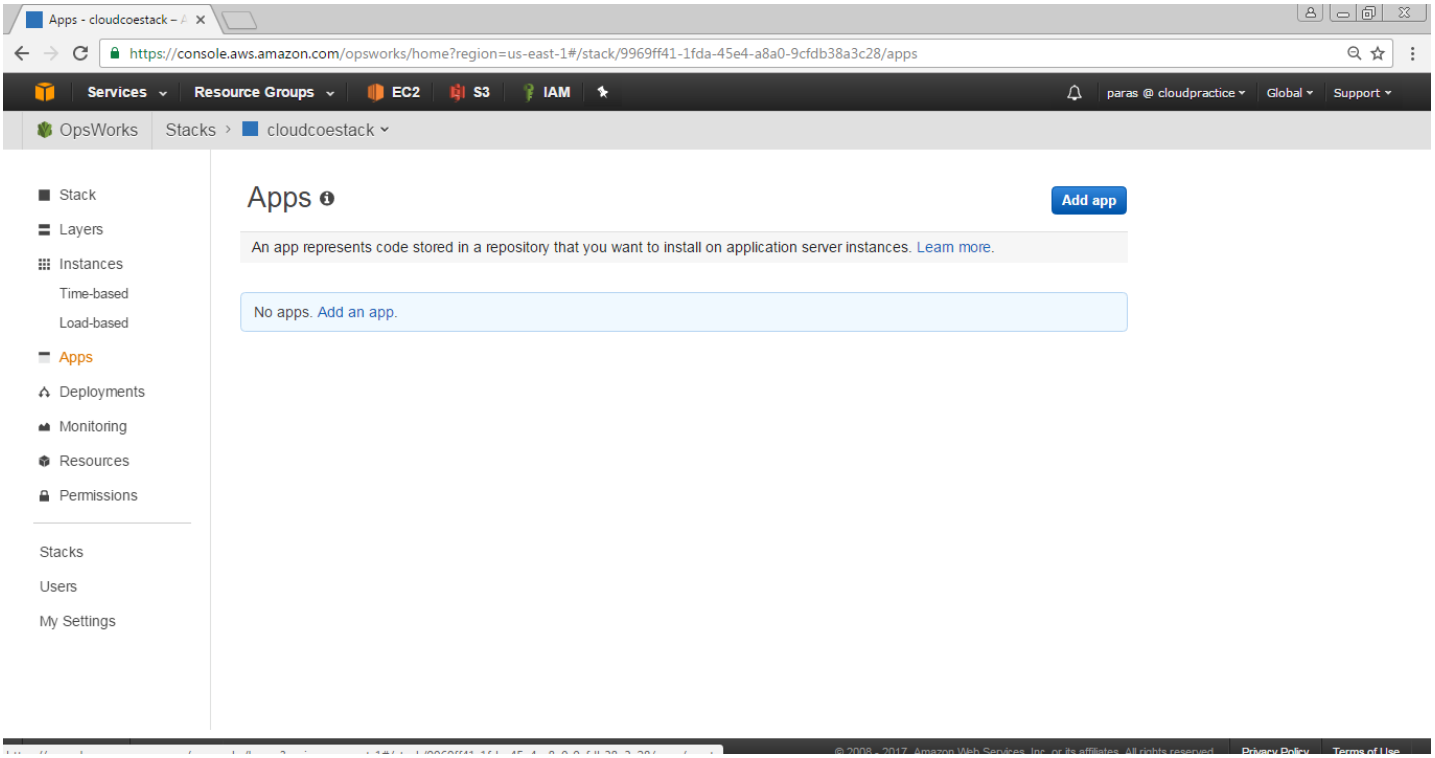


19 On the Layers page, choose **Security**. The Layer **MyDemoLayer** page is displayed with the Security tab open. For **Security groups**, choose **AWS-OpsWorks-WebApp**, and then click on **Save** button

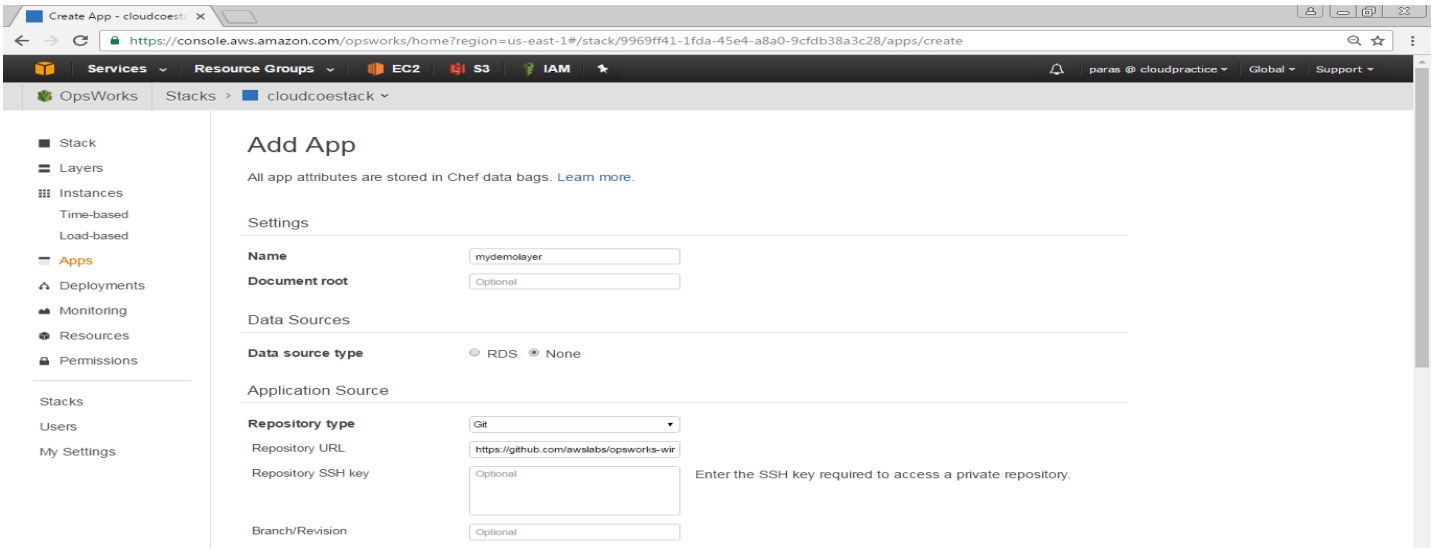


15.2.3.3 Specify the App to Deploy to the Instance

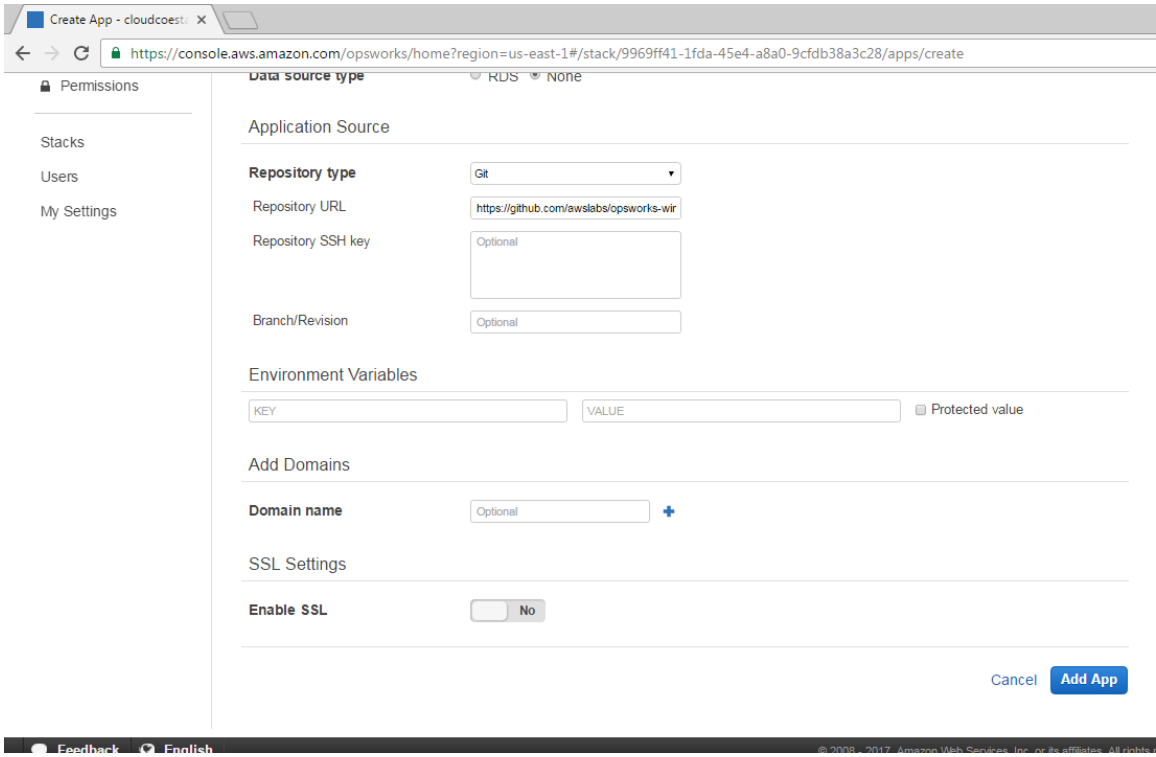
- In the service navigation pane, choose **Apps**. The **Apps** page is displayed. Click on **Add an app**. The Add App page is displayed



- For Settings, do the following
    - ✓ For **Name**, type **mydemolayer**
    - ✓ For **Document root** leave blank
    - ✓ For **Data Sources** select **None Data source type**
    - ✓ For **Application Source ,** for **Repository type** select **Git**
    - ✓ For **Application Source**, for **Repository URL**, type *https://github.com/awslabs/opsworks-windows-demo-nodejs.git*
    - ✓ For **Repository SSH key** leave blank
    - ✓ For **Branch/Revision** leave blank

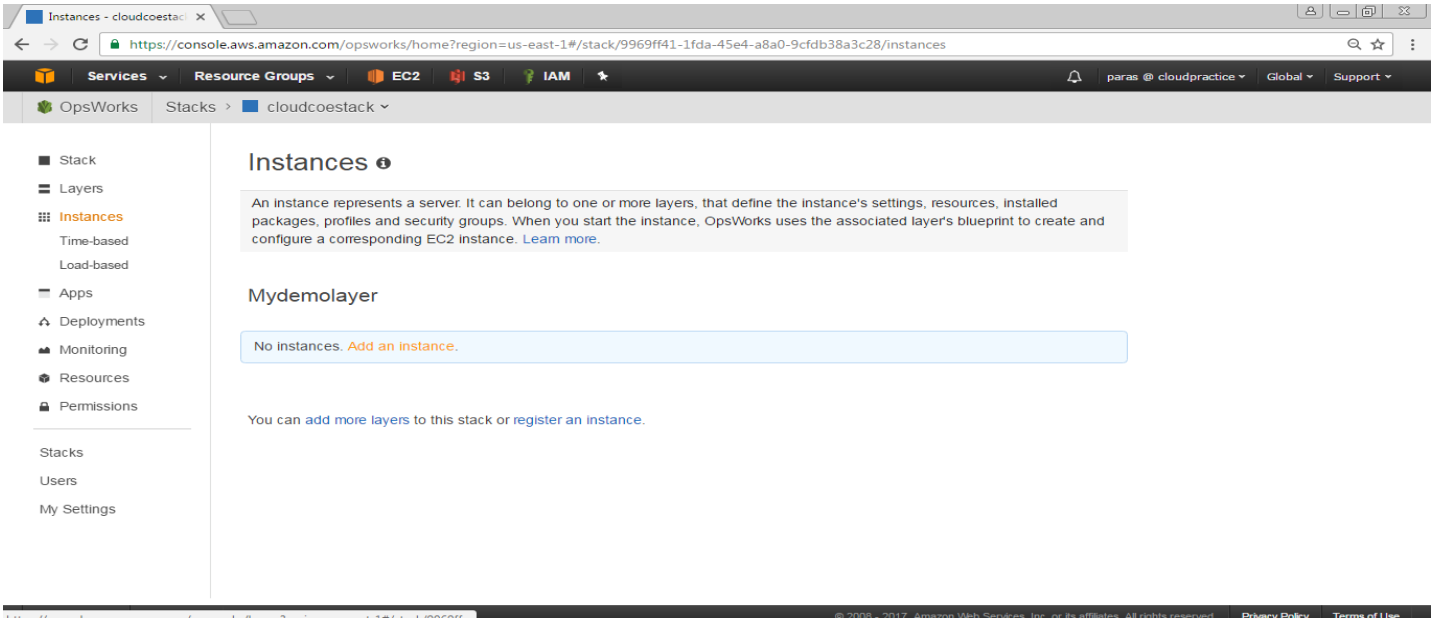


- ✓ For **Environment Variables, KEY** leave blank, **VALUE** leave blank, unchecked **Protected Value**
- ✓ For **Add Domains**, **Domain Name** leave blank
- ✓ For **SSL Settings**, select **No for Enable SSL**
- ✓ Click on **Add App** button. AWS OpsWorks Stacks adds the app and displays the **Apps** page

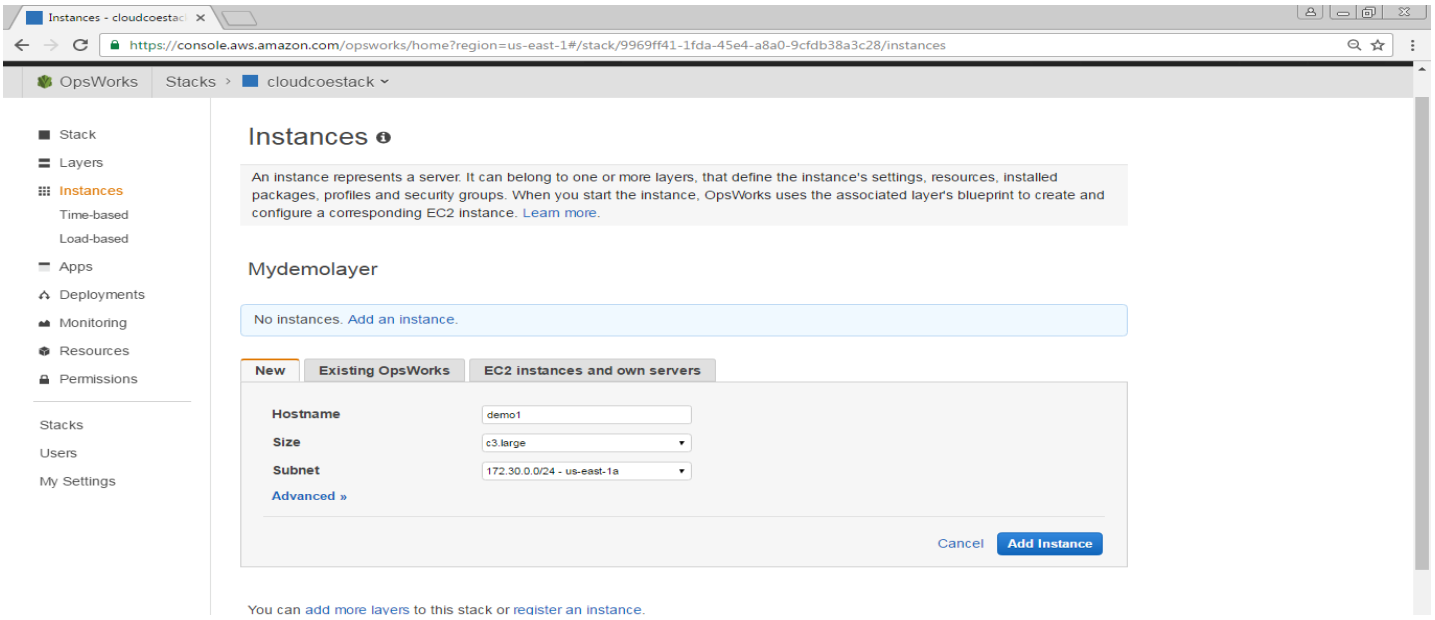


15.2.3.4 Launch an Instance

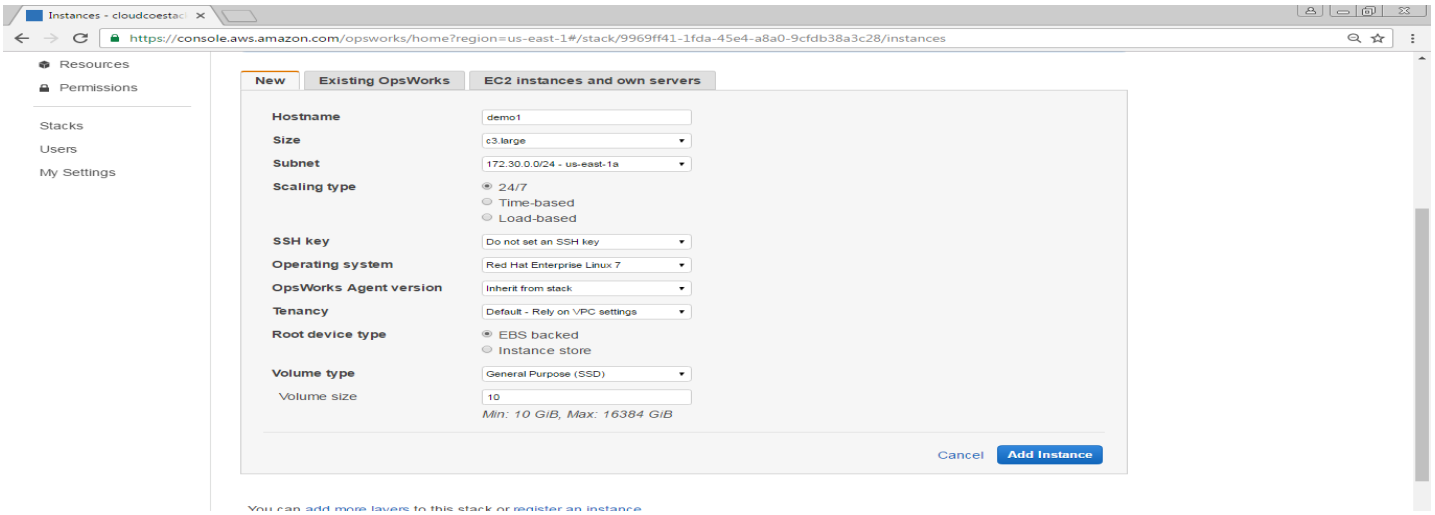
- In the service navigation pane, choose **Instances**. The Instances page is displayed. For **MyDemoLayer**, click **Add an instance**



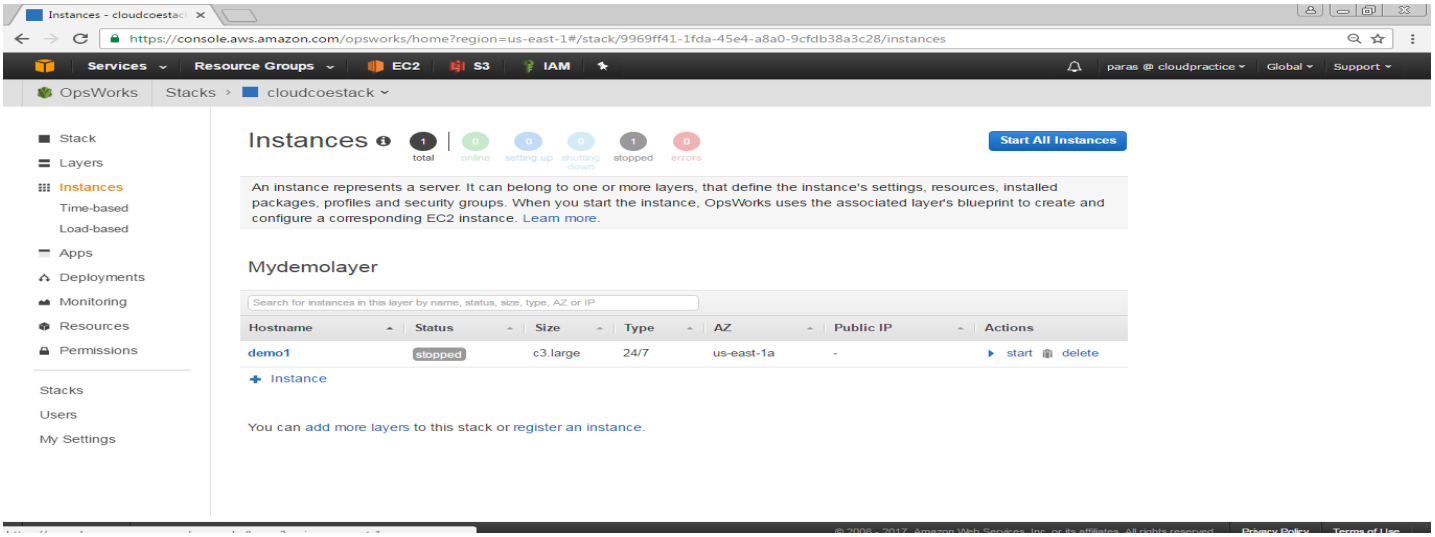
- On the New tab, do the following
- For **Hostname** give a hostname for example demo1
- For **Size** choose **c3.large** *(default size)*
- For **Subnet** choose your subnet



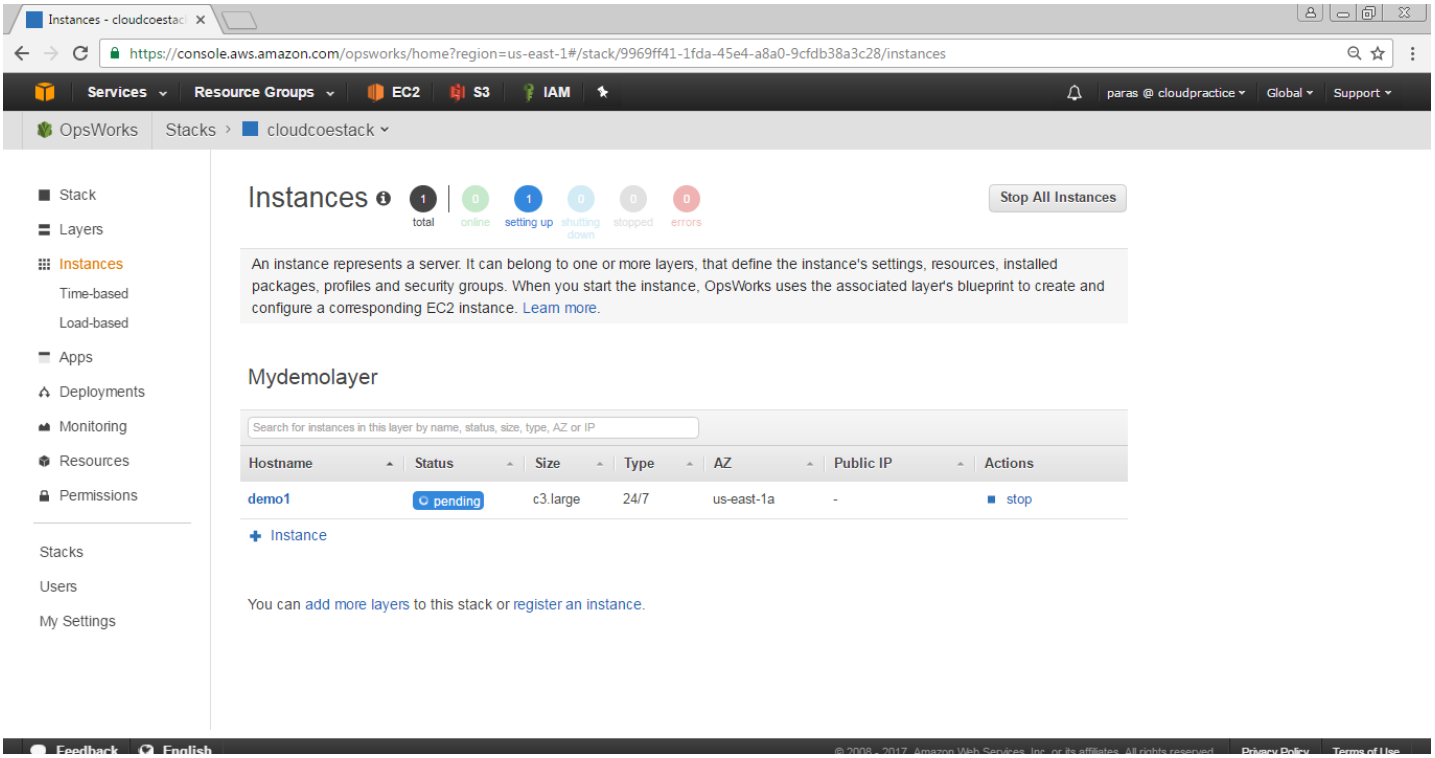
- Choose Advanced, do the following configuration
    - For **Scaling type** select **24/7**
    - For **SSH key** select **Do not use a default SSH key**
    - For **Operating system** select **Red Hat Enterprise Linux 7**
    - For **OpsWorks Agent version** select **Inherit from stack**
    - For **Tenancy** select **Default - Rely on VPC settings**
    - For **Root device type** select **EBS backed**
    - For **Volume type** select **General Purpose (SSD)**
    - For **Volume size** select **10**
    - Click on **Add Instance**

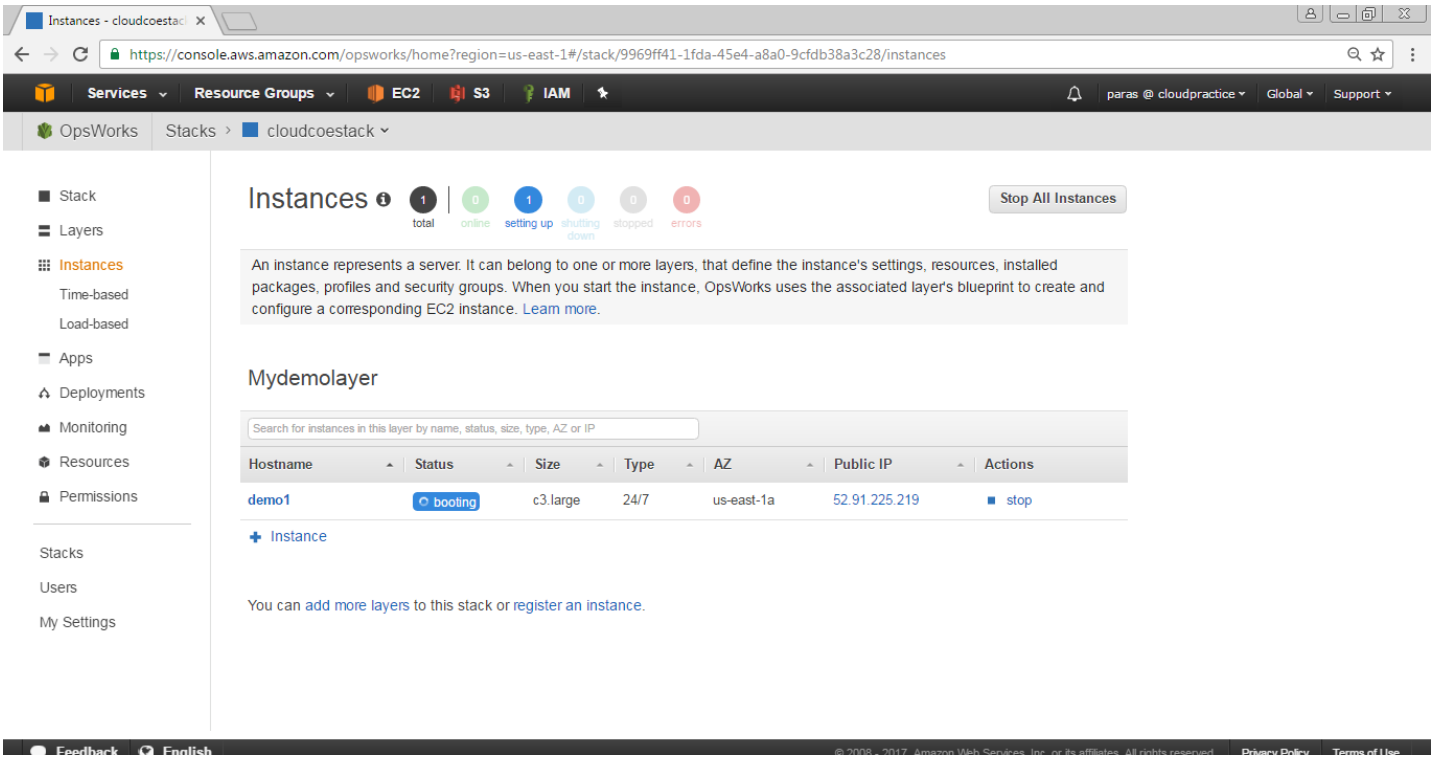


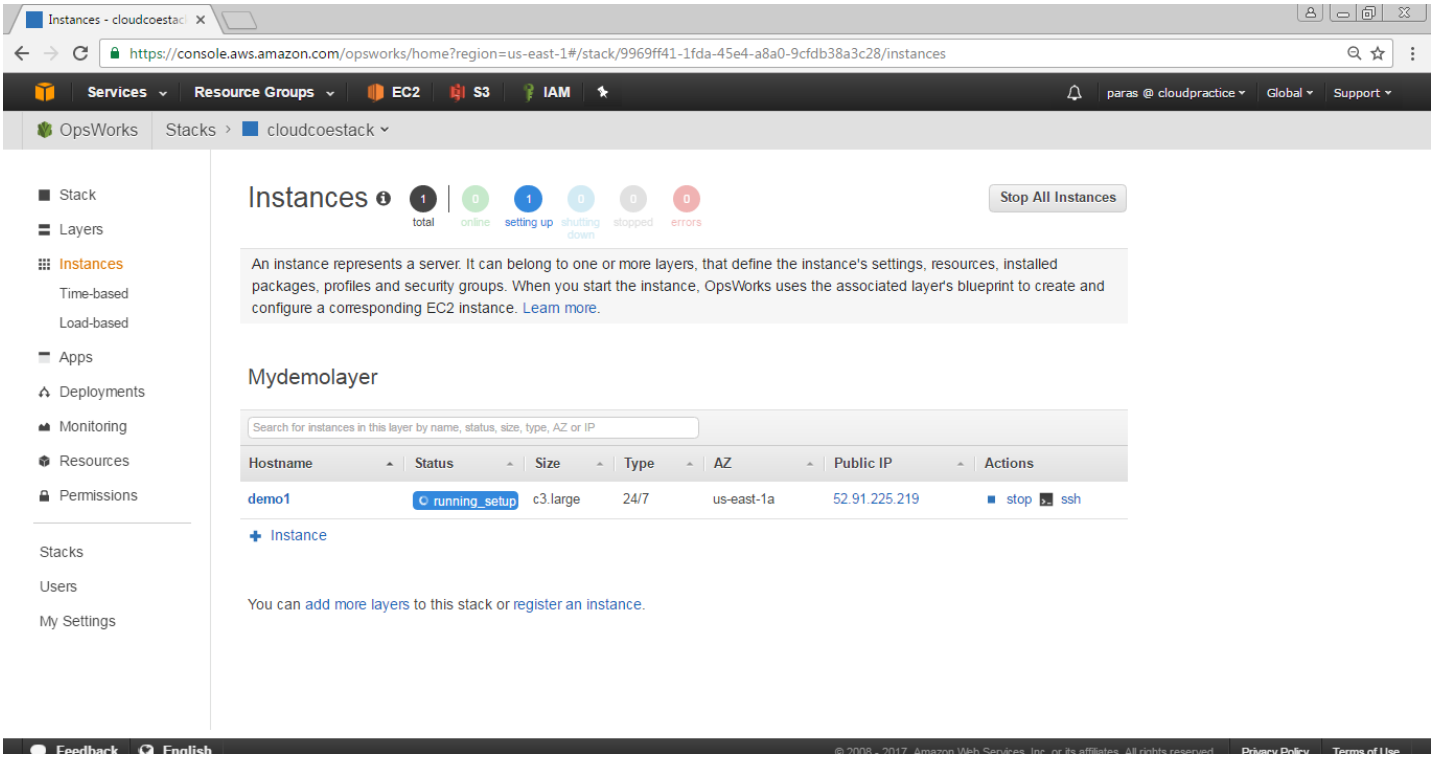
- AWS OpsWorks Stacks adds the instance to the layer and displays the **Instances** page. For **MyDemoLayer**, for **demo1**, for **Actions**, click on **start**



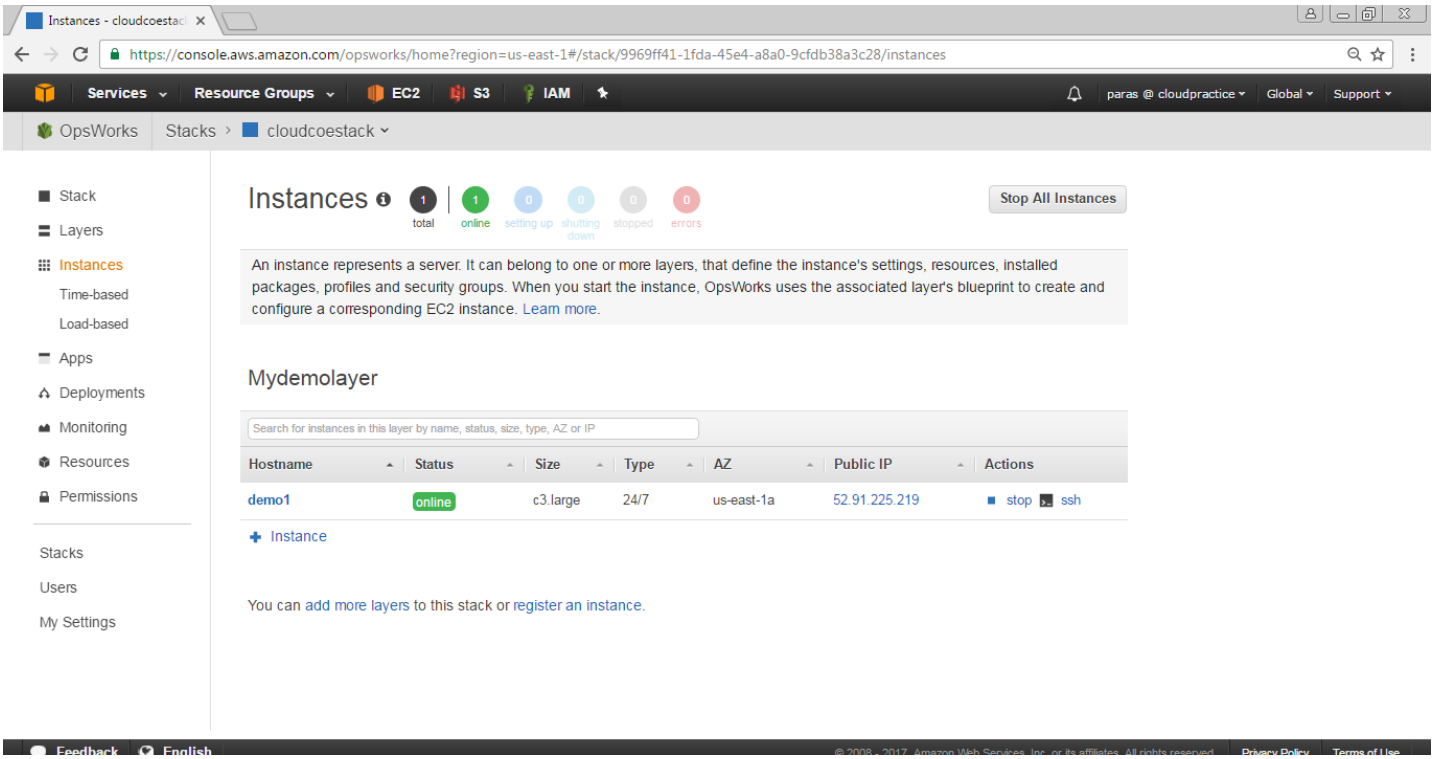
- Over the course of several minutes. Status turns from stopped to requested, to pending, to booting, to running_setup, and then finally to online. Note that this process can take several minutes. Status is shown in the following screen shots







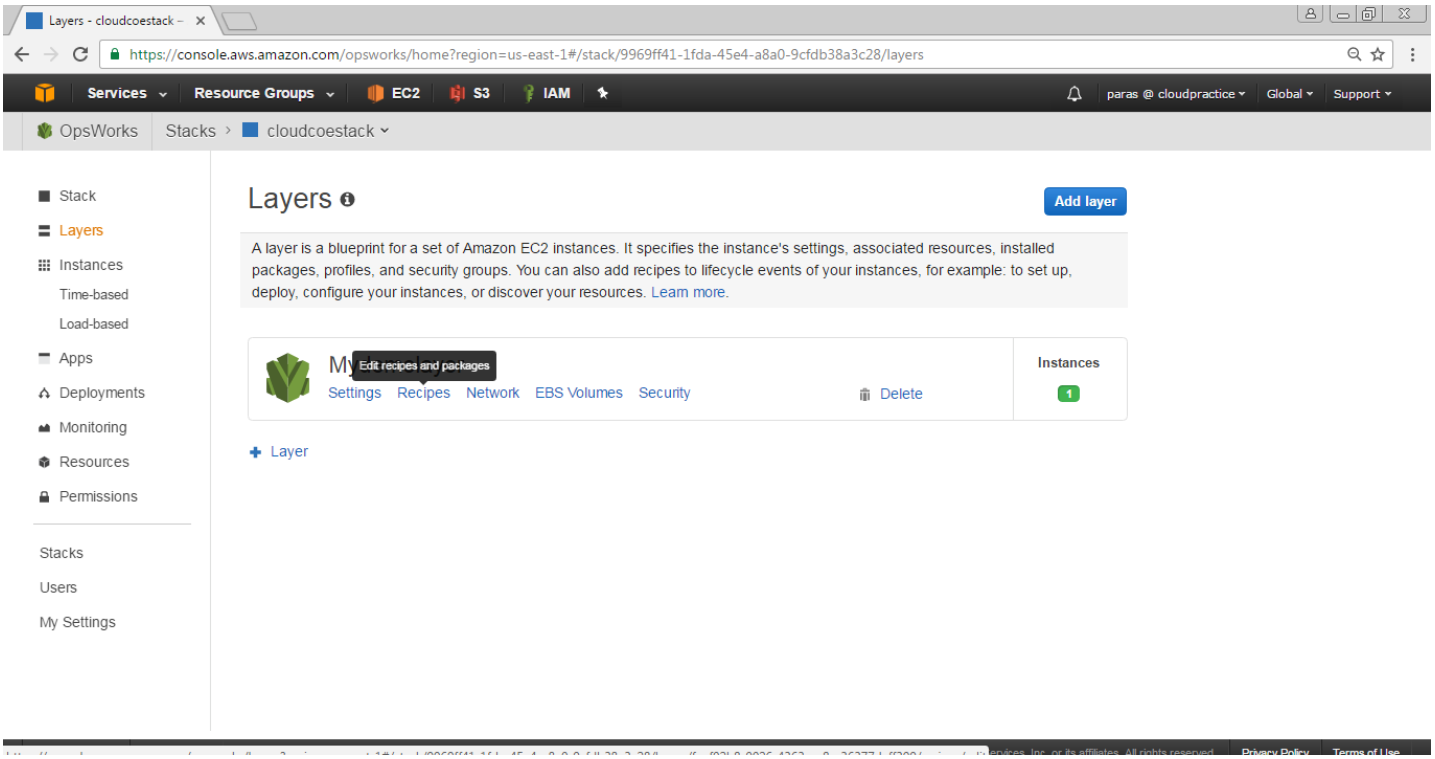
- After **Status** changes to **online**, the **setting up** circle indicator changes from **1** to **0**, and the **online** circle changes from **0** to **1** and changes to bright green. Do not proceed until the **online** circle changes to bright green, and shows **1** instance online



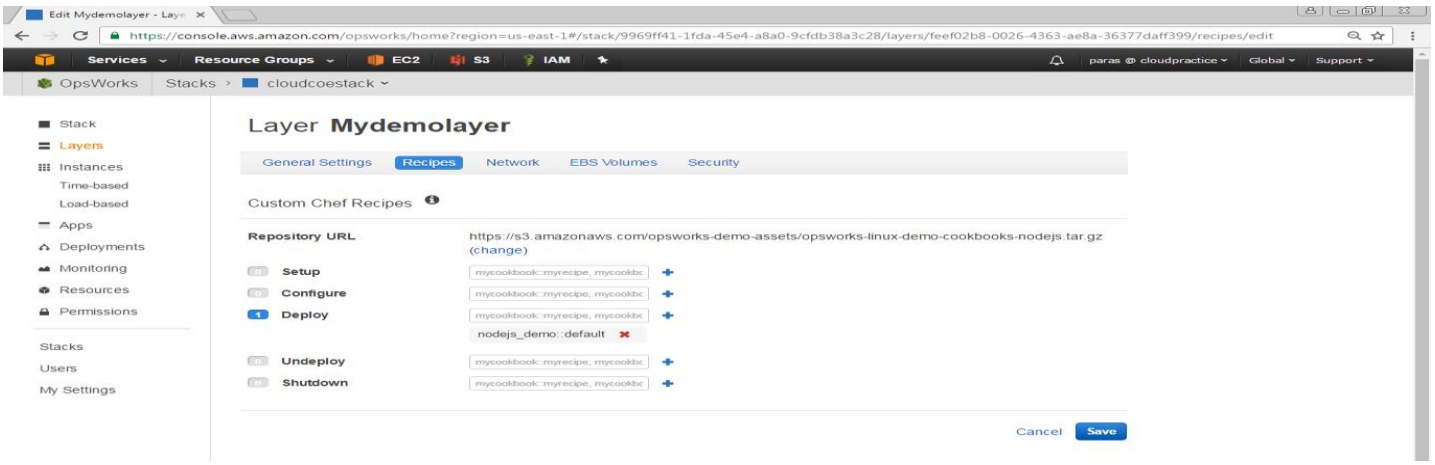
Note: You now have an instance that is ready for the app to be deployed to it

15.2.3.5 Deploy the App to the Instance

1. In the service navigation pane, choose **Layers**. The Layers page is displayed. For **MyDemoLayer**, choose **Recipes**



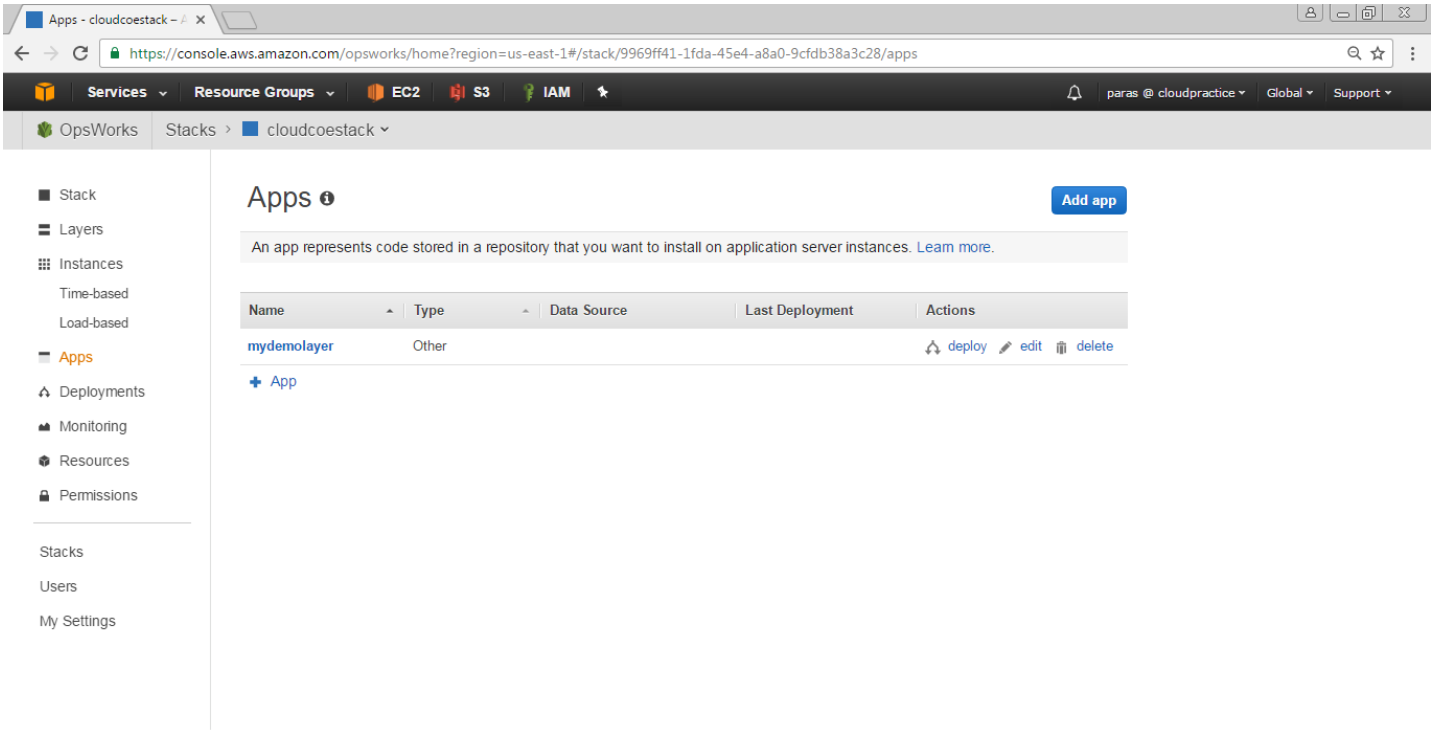
2. For **Custom Chef Recipes**, for **Deploy**, type *nodejs_demo::default*, and then press **Enter** (Here nodejs_demo is the name of the cookbook and default is the name of the target recipe within the cookbook). Click on **Save** button



3. **To deploy the app to the instance,**

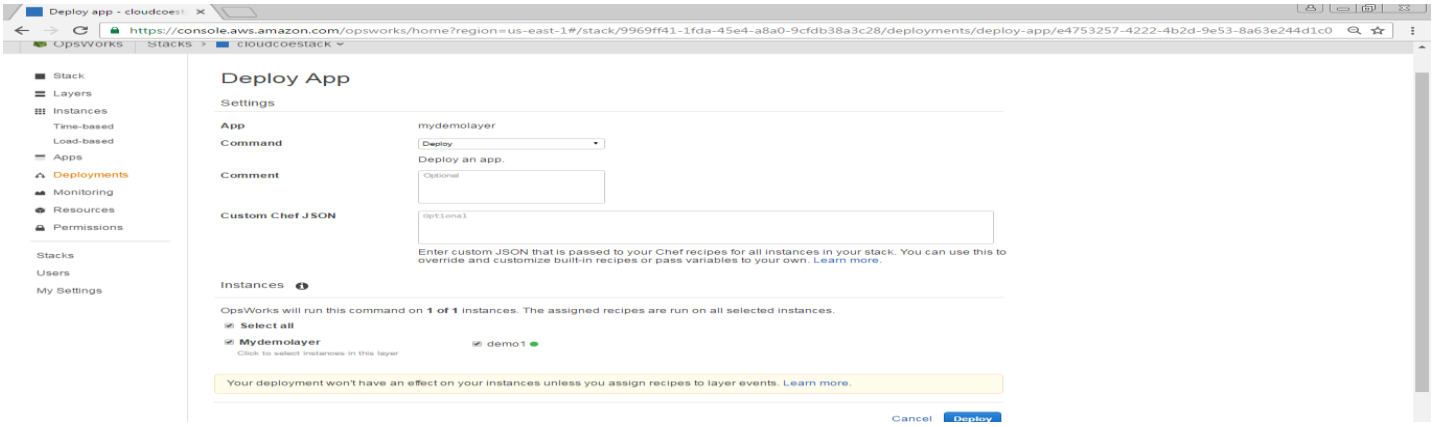
   In the service navigation pane, choose **Apps**. The Apps page display

   For **MyDemolayer**, for **Actions**, choose **deploy**

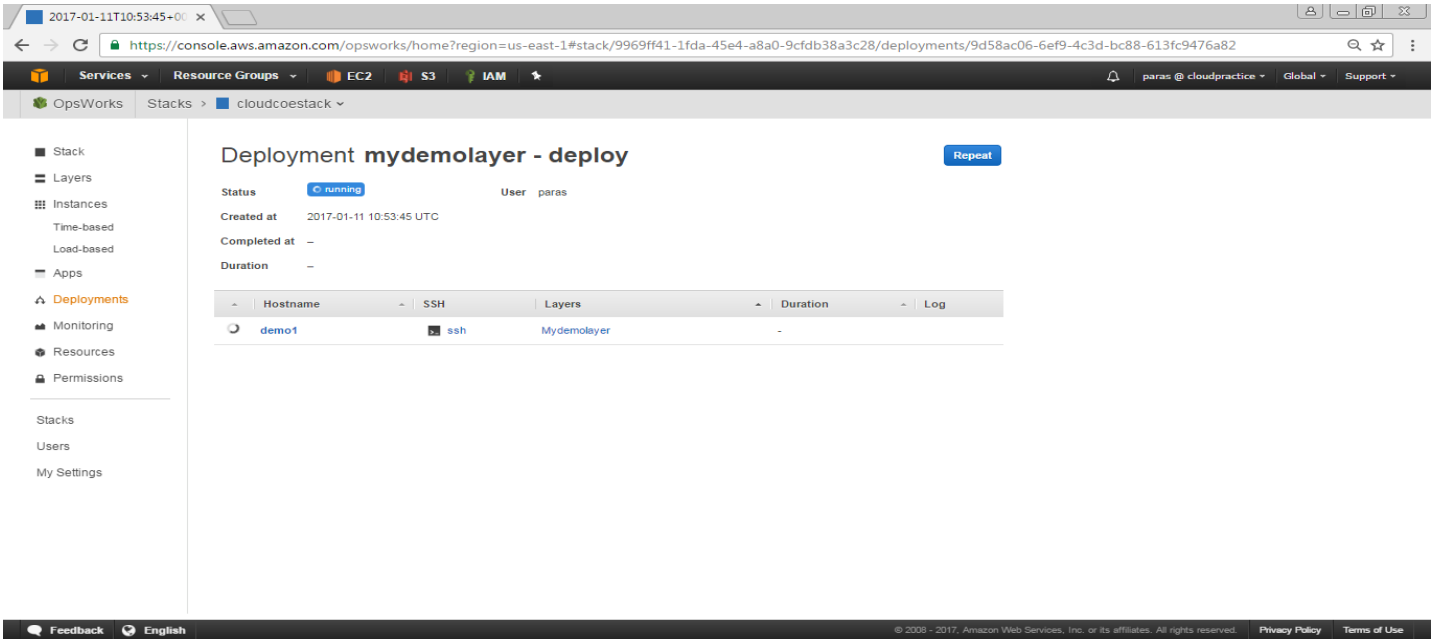


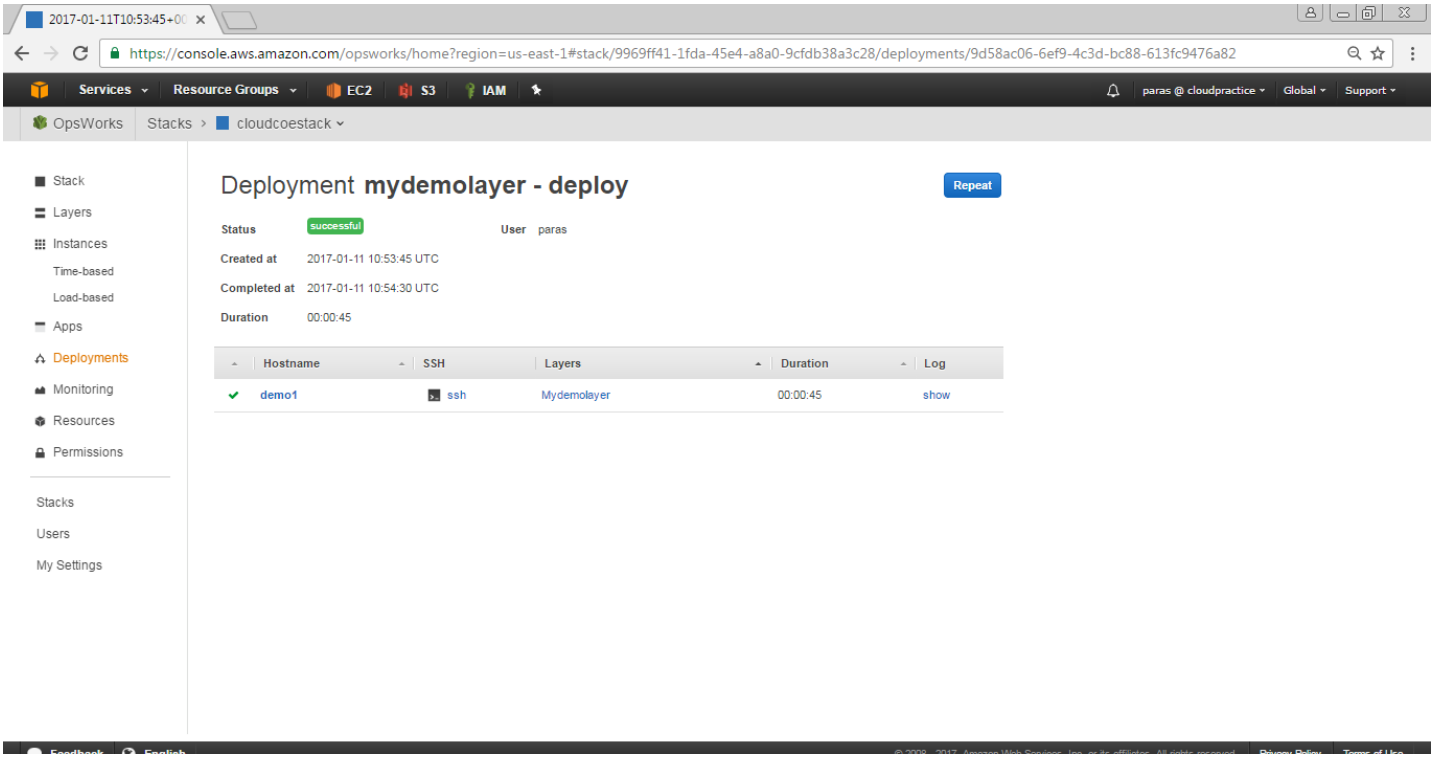
- On the Deploy App page, do the following configuration
- For **Command** select **Deploy**
- For **Comment** leave blank
- For **Settings**, **Custom Chef JSON** leave blank
- For **Instances**, checked **Select all**, checked **MyDemoLayer**, checked **demo1**

4. Click on **Deploy** button

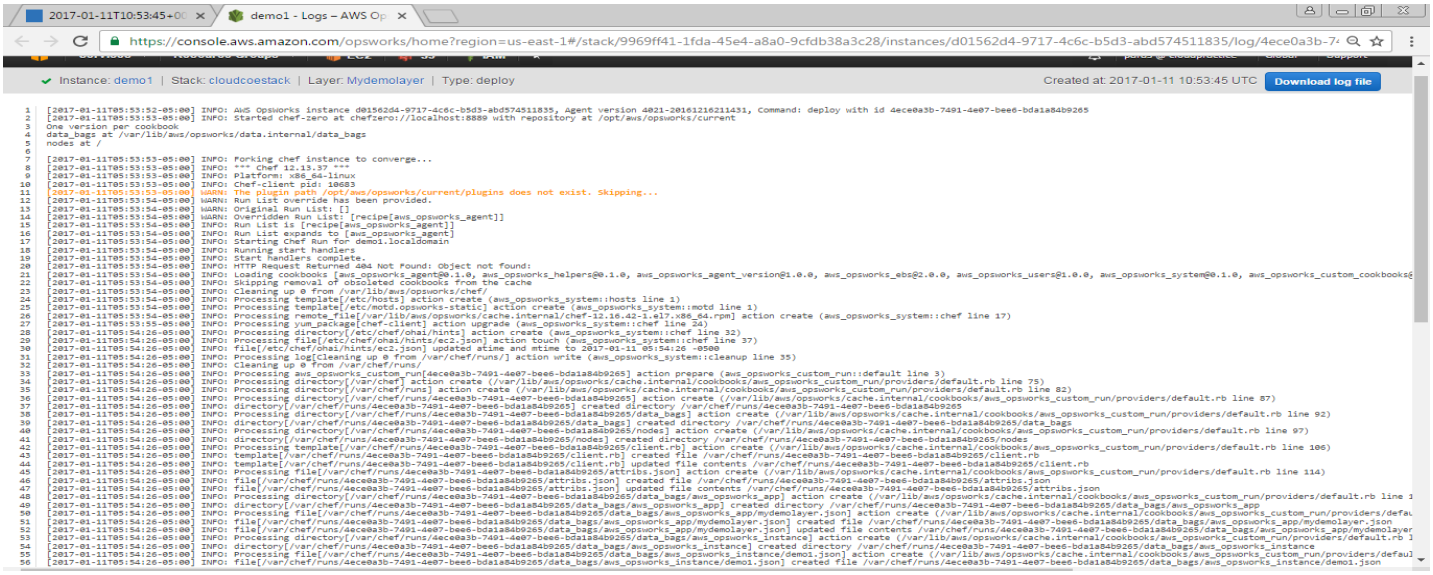


5. The **Deployment MyDemolayer – deploy** page is displayed. **Status** changes from **running** to **successful**. A spinning circle displays next to **demo1**, which then changes to a green check mark. Note that this process can take several minutes. Do not proceed until you see both a **Status** of **successful** and the green check mark icon.



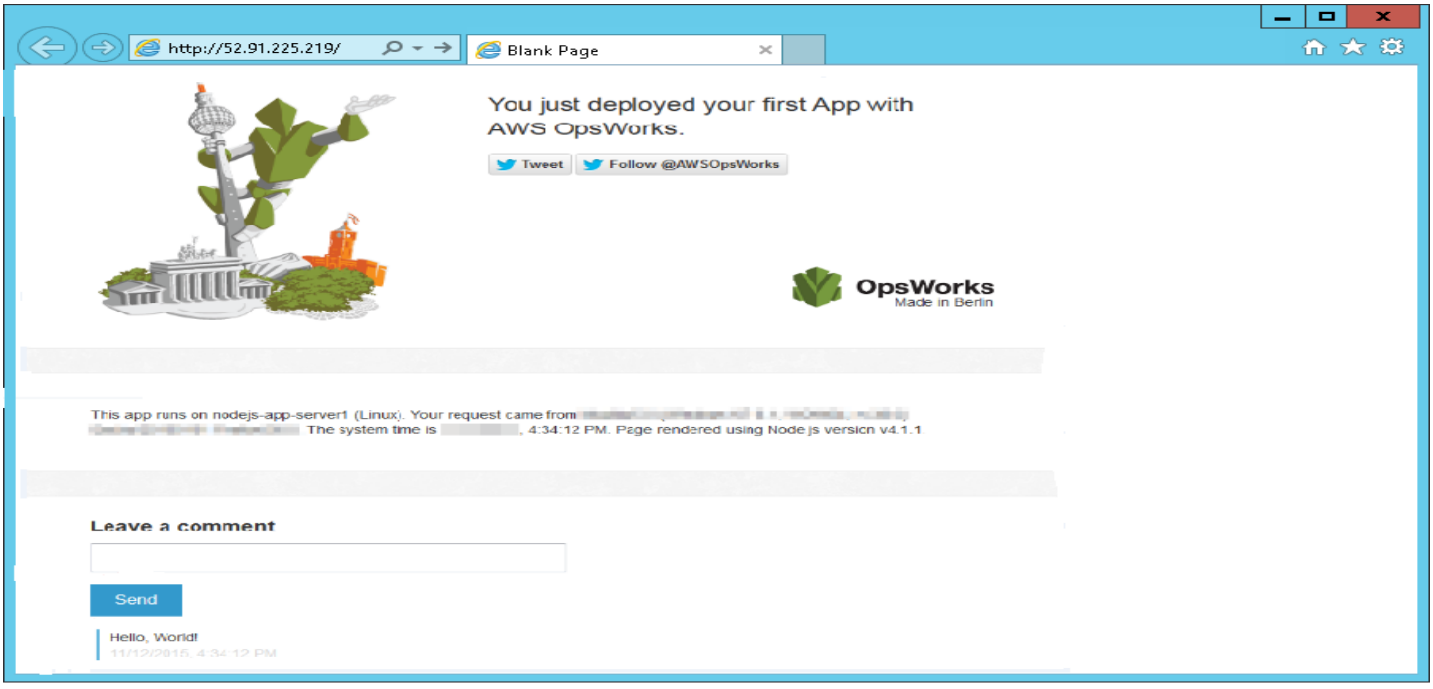


6. To see the Logs, You can click on **show** under **Log** (inside the table). A new link will open showing you the logs



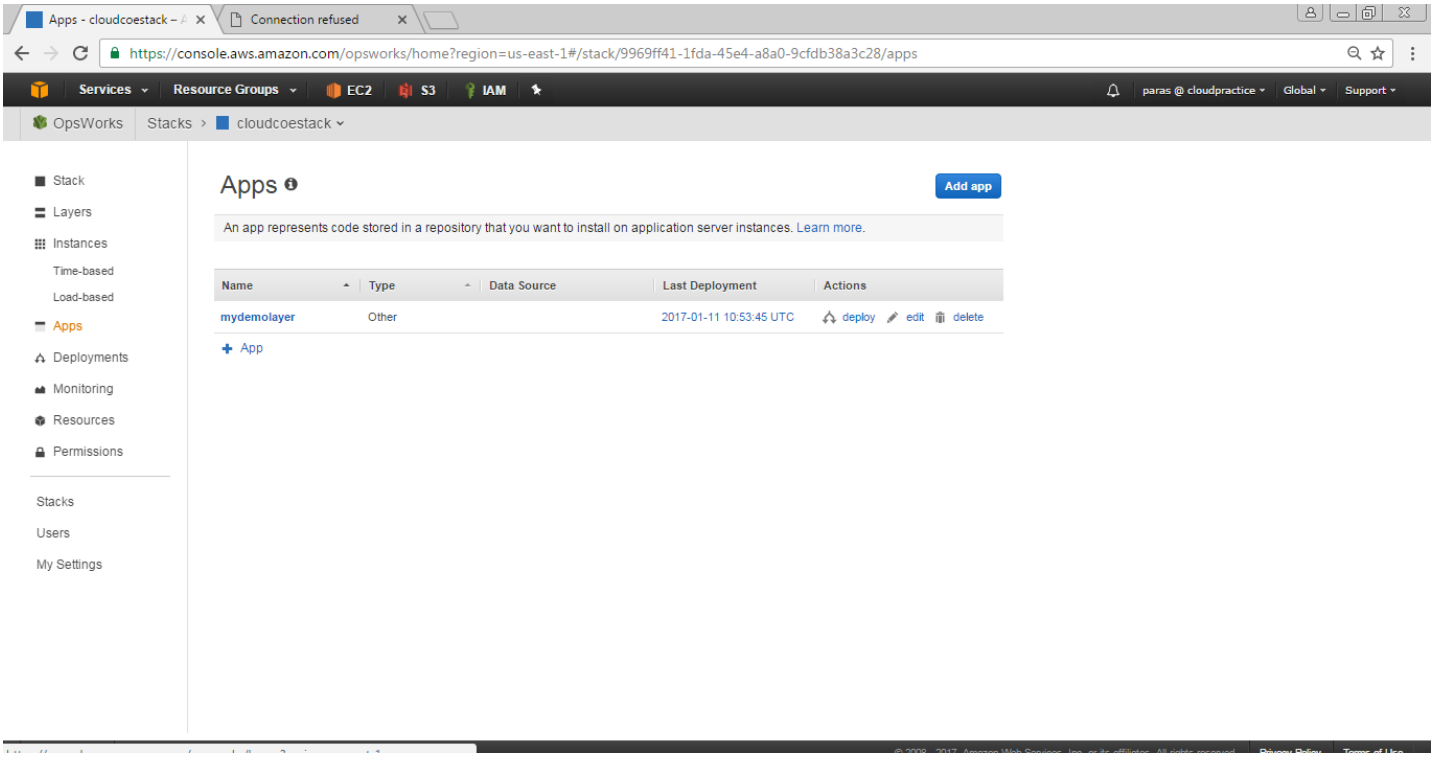
15.2.3.6 Test the Deployed App on the Instance

- To test the deployment on the instance, In the service navigation pane, choose **Instances**. The **Instances** page is displayed. For **MyDemoLayer**, for **demo1**, for **Public IP**, choose the **IP address**
- On the congratulatory web page, in the Leave a comment text box, type a comment, and then choose Send to test the app. The app adds your comment to the web page

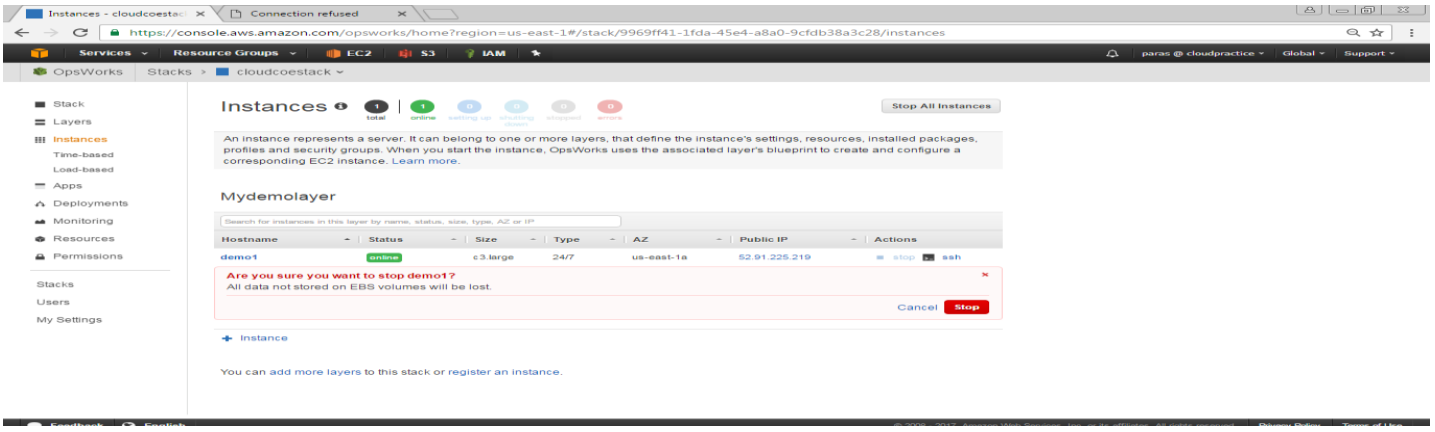


NOTE: You have now successfully tested the deployed app on the instance

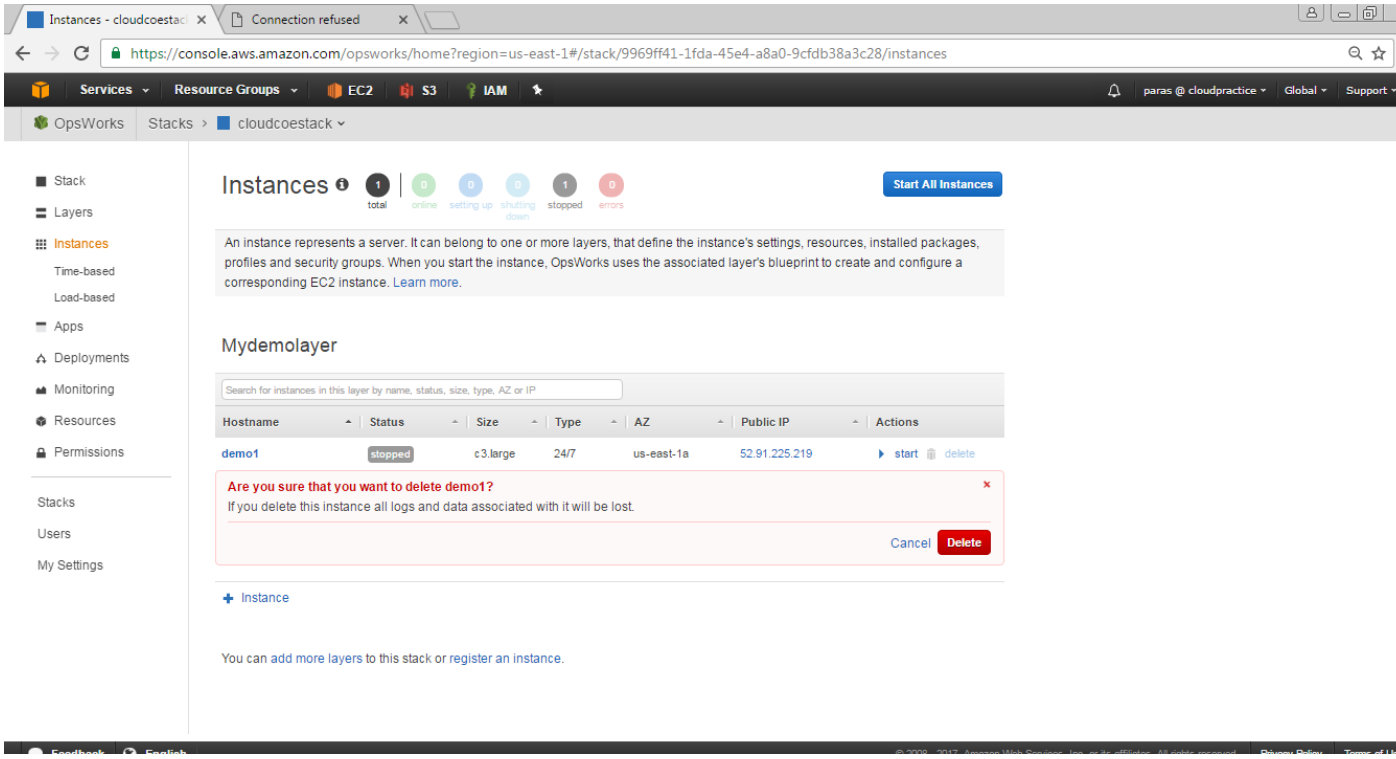
- To delete the app from the stack. In the service navigation pane, choose **Apps**. The **Apps** page is displayed. For **MyDemolayer**, for **Actions**, choose **delete**. When the confirmation message is displayed, choose **Delete**. AWS OpsWorks Stacks deletes the app



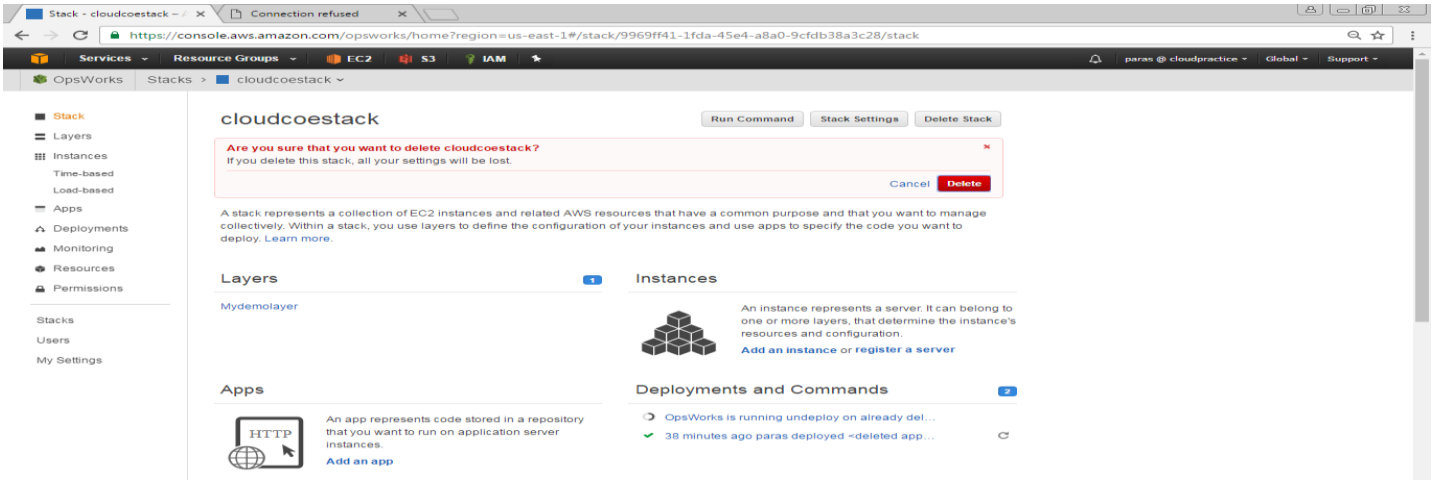
- To delete the instance for the stack. In the service navigation pane, choose **Instances**. The Instances page is displayed. For **MyDemoLayer**, for **demo1**, for **Actions**, choose **stop**. When you see the confirmation message, choose **Stop.**



- When the instance is stopped in the **Actions**, click on **delete.** When you see the confirmation message, click on **Delete** button



- To delete the stack. In the service navigation pane, choose **Stack**. The **cloudcoestack** page is displayed. Click on **Delete Stack**. When you see the confirmation message, click on **Delete.**



16. AWS Lambda

AWS Lambda is a compute service that makes it easy for you to build applications that respond quickly to new information.

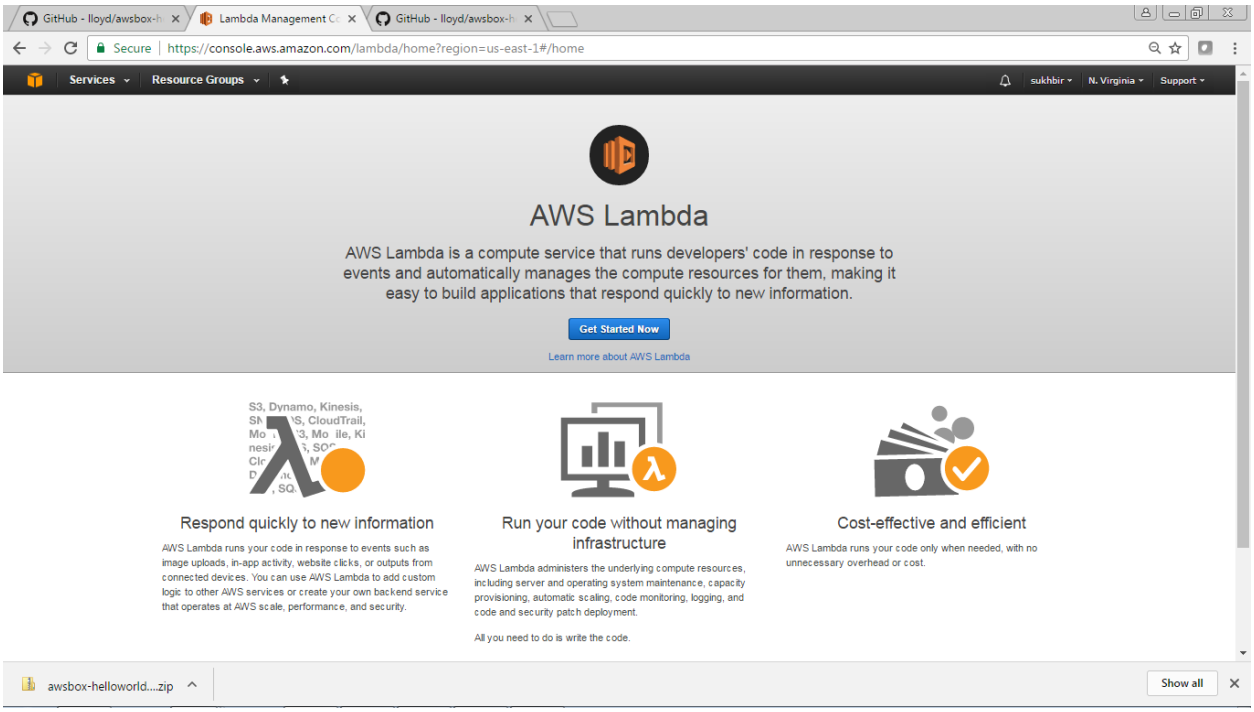
16.1 Objective

To understand the process of using AWS Lamda service for creating and testing the function written in python.

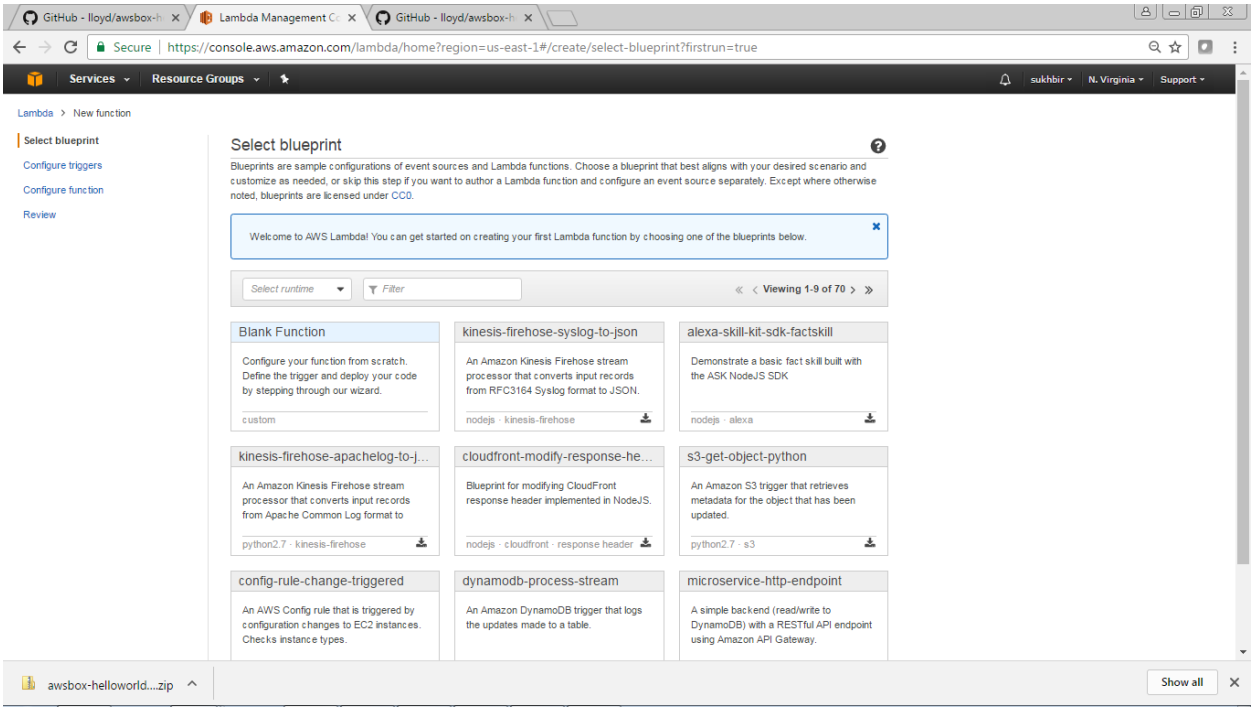
16.2 Procedure

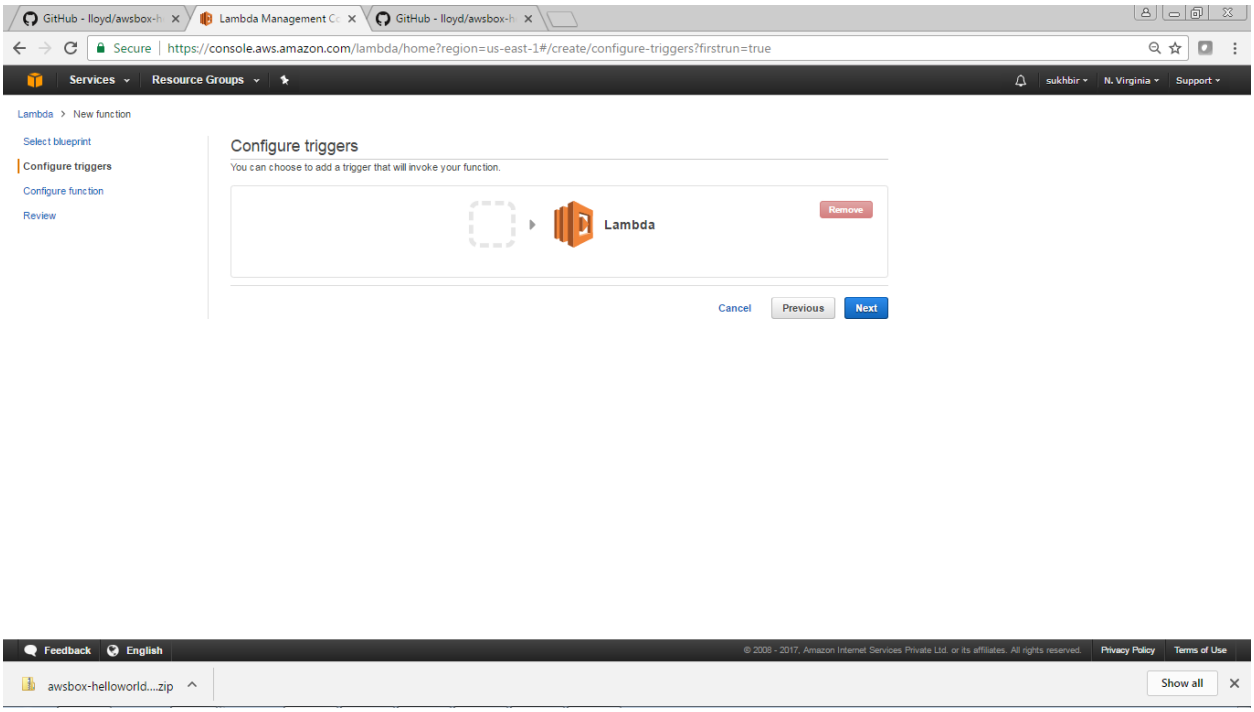
16.2.1 Creating Lamda Function

1. Sign in to the AWS Management Console and open the AWS Lambda console (Assuming user is having IAM access rights for AWS Lamda.



2. Choose Create a Lambda function
3. On the Select blueprint page, choose Skip
4. On the Configure Function page, enter lambdaTest for Name, and choose Python 2.7 for Runtime

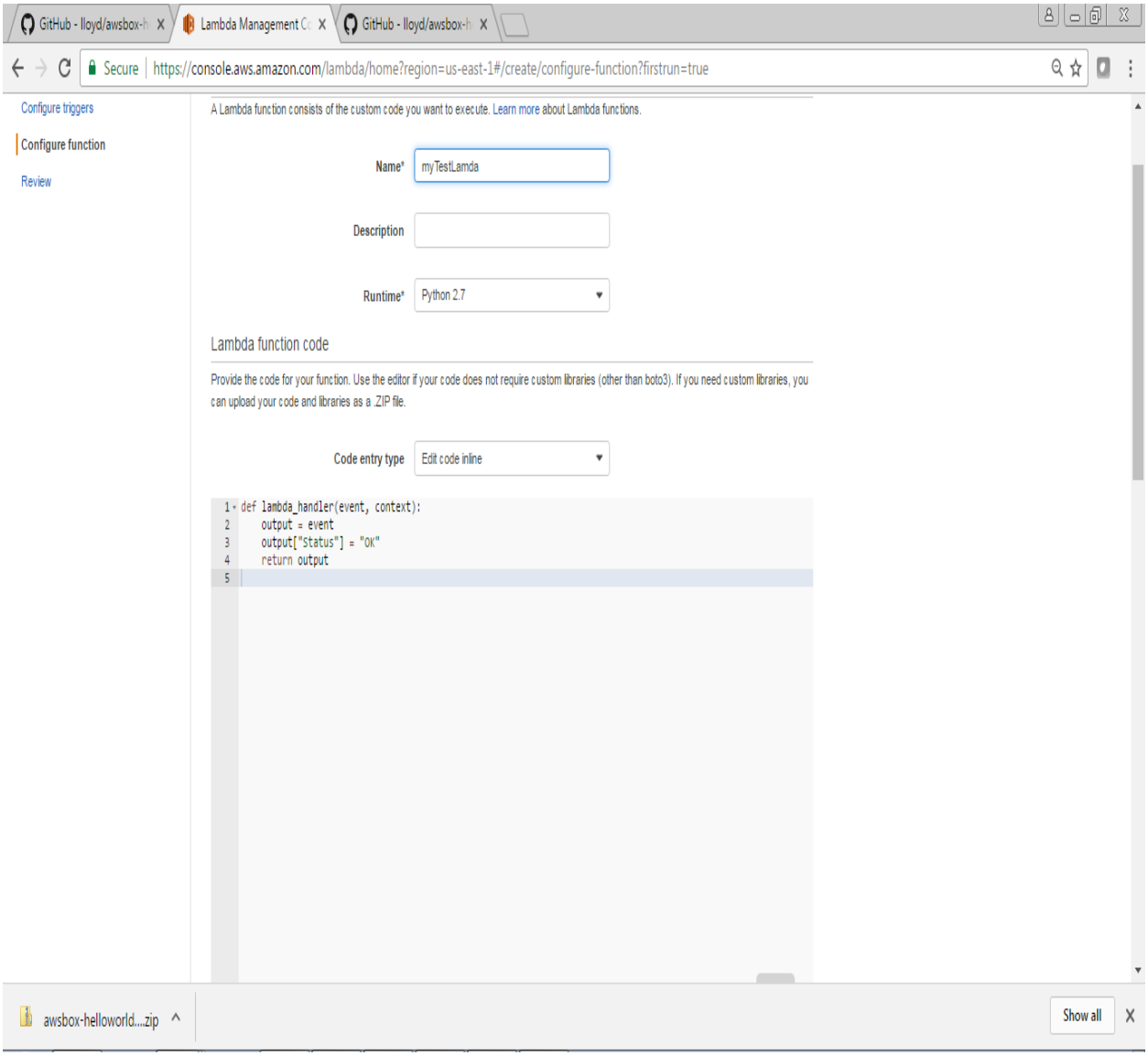




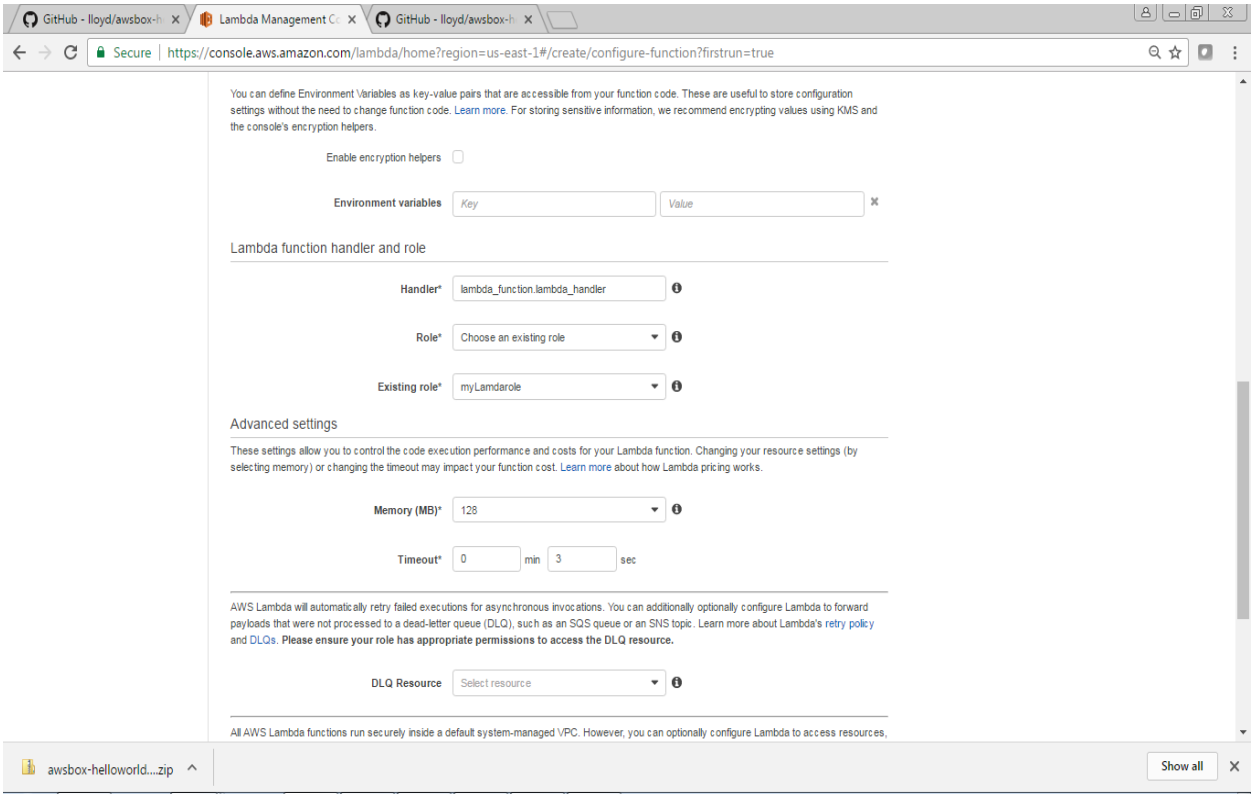
5. Paste the following code for Lambda Function Code:

```
def lambda_handler(event, context):
```

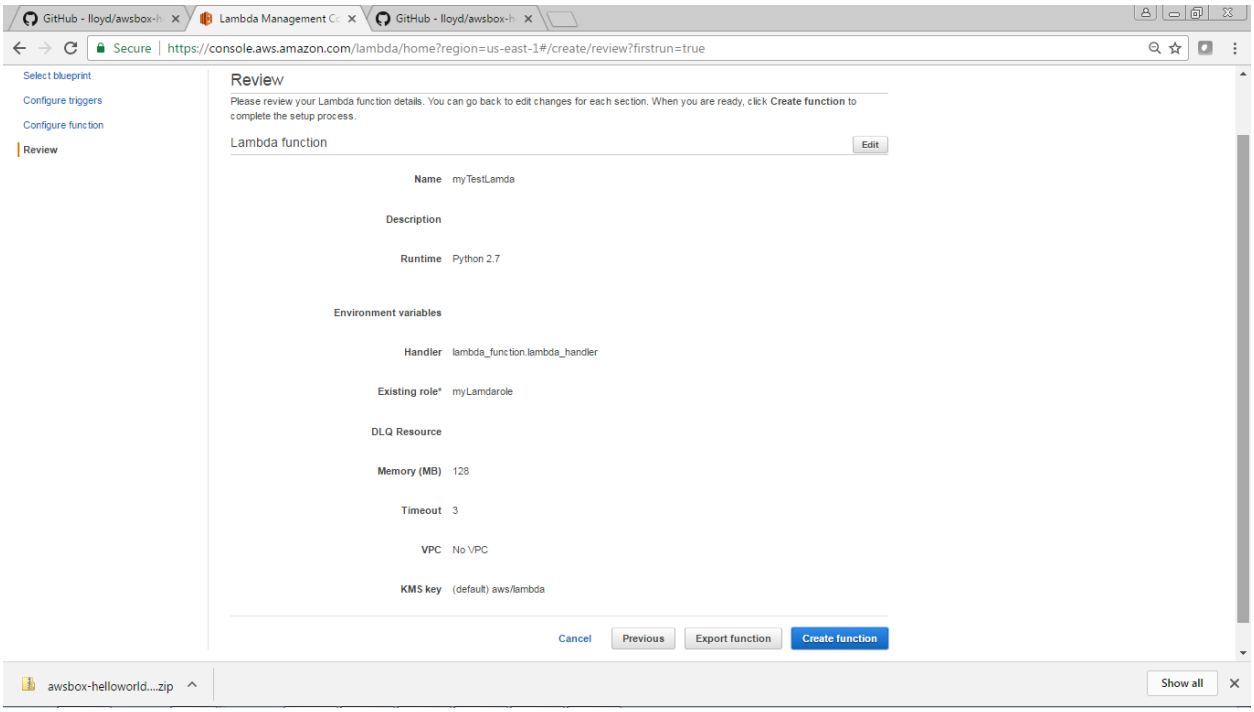
```
    output = event
    output["Status"] = "OK"
    return output
```



6. Assign a value for Role to the function. If you have used Lambda before, you can select an existing role; otherwise, select the option to create a new Basic execution role.
7. The value of Timeout defaults to 3 seconds, which is fine for this function. Other functions may need more time to execute.

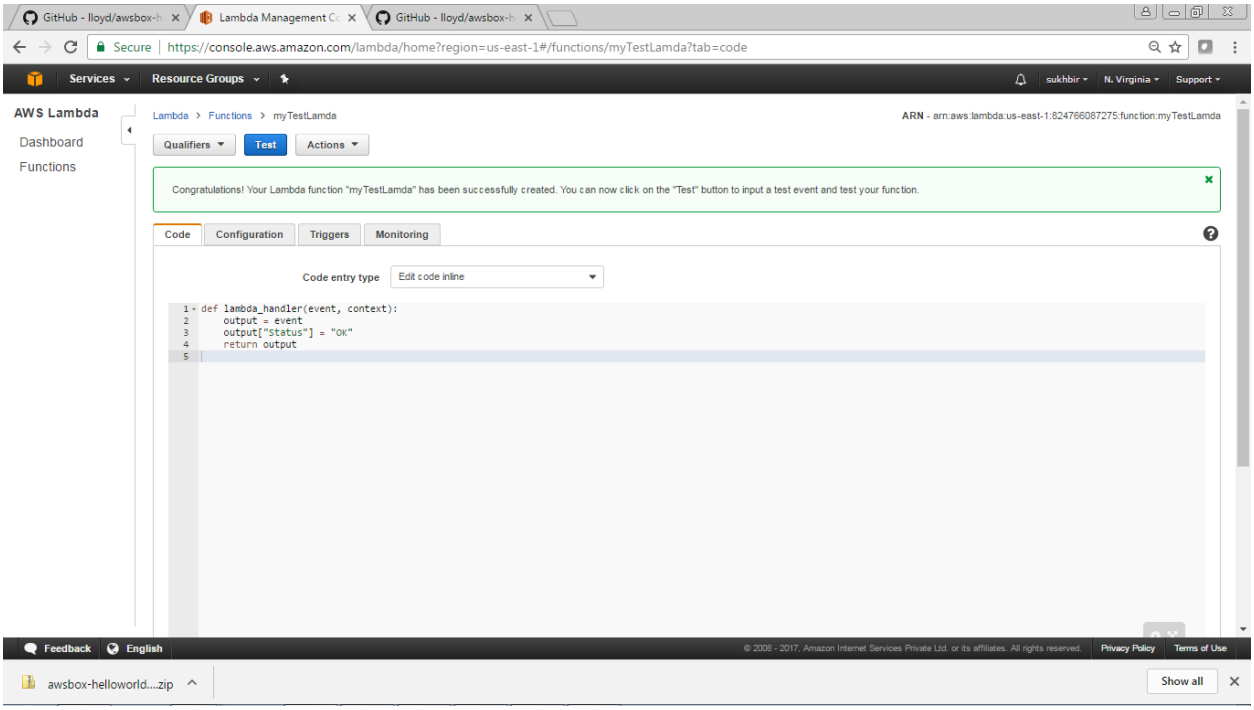


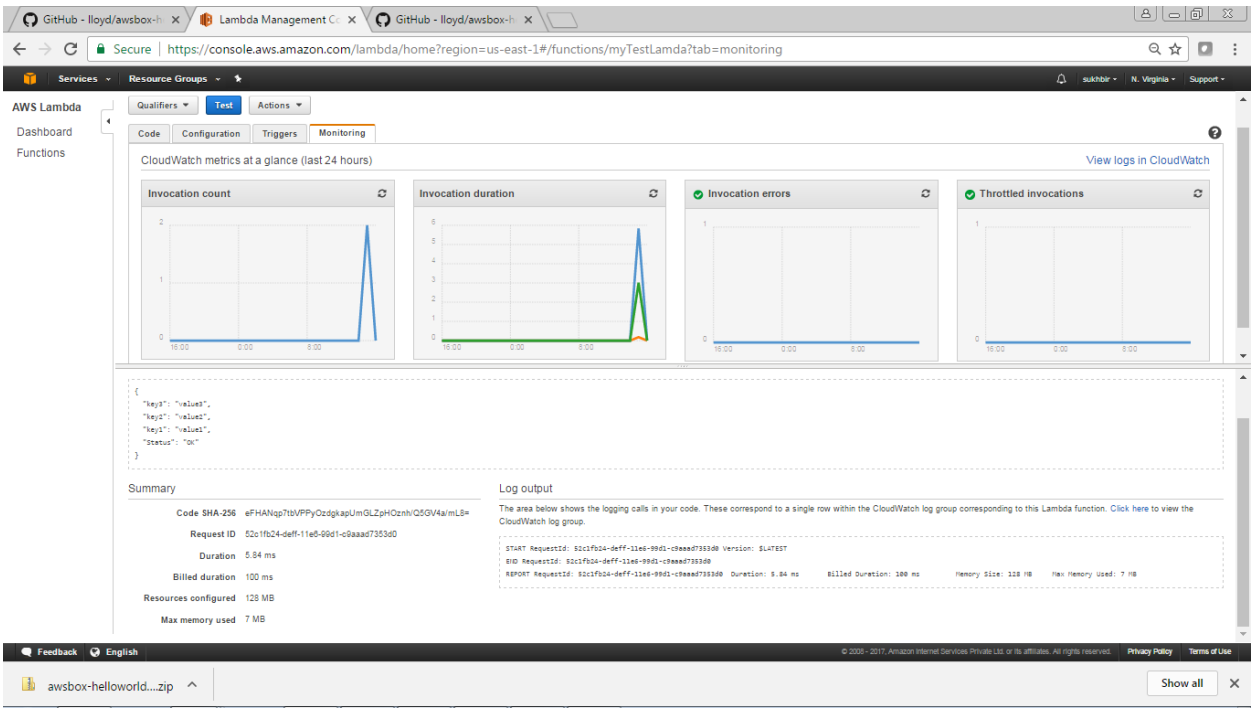
Accept all other defaults and choose Next, Create Function.



16.2.2 Testing Lamda Function

1. Test your function from the console by choosing Test, and verifying that it runs with no errors and that it returns a JSON object reflecting the input data with an added “Status” key.





17. APPLICATION & DEVELOPER TOOL SUPPORT

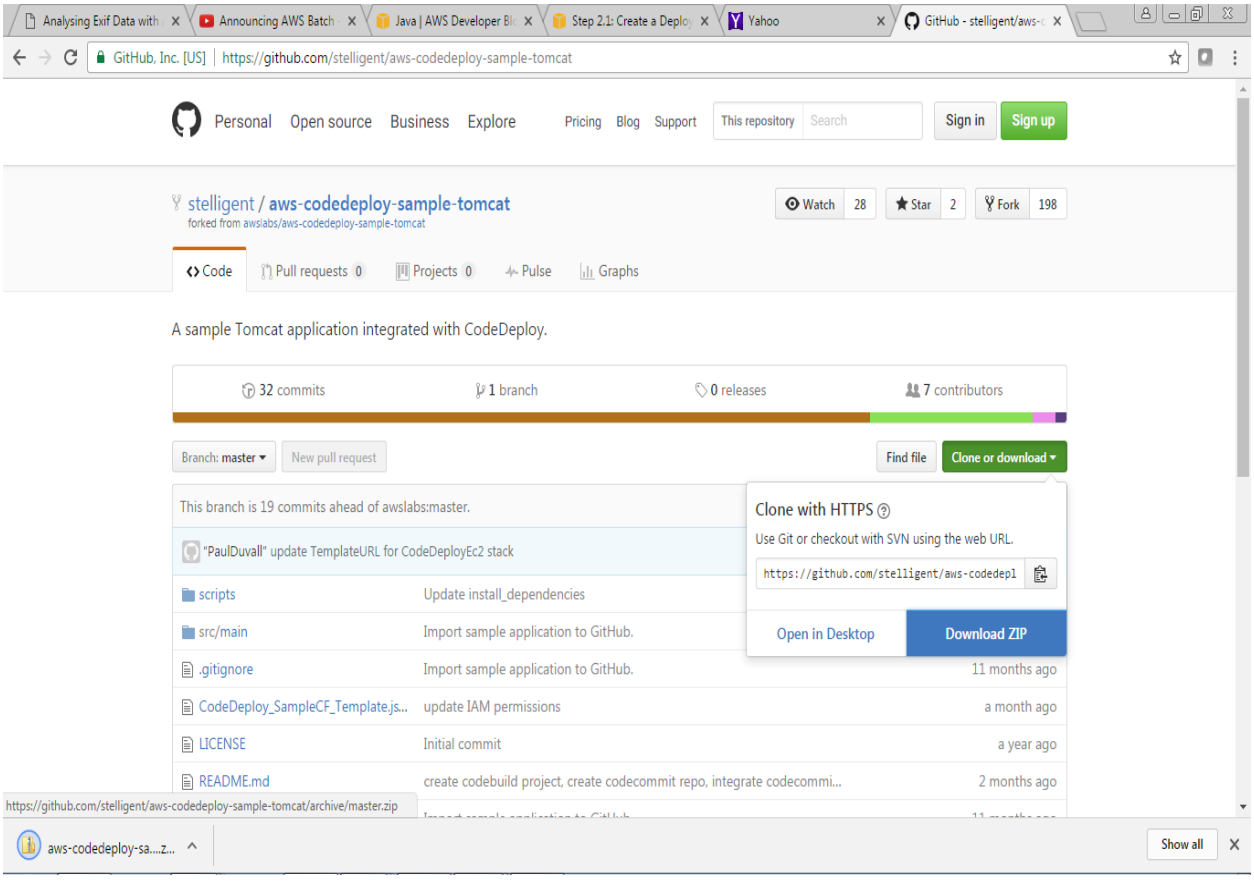
17.1 Objective

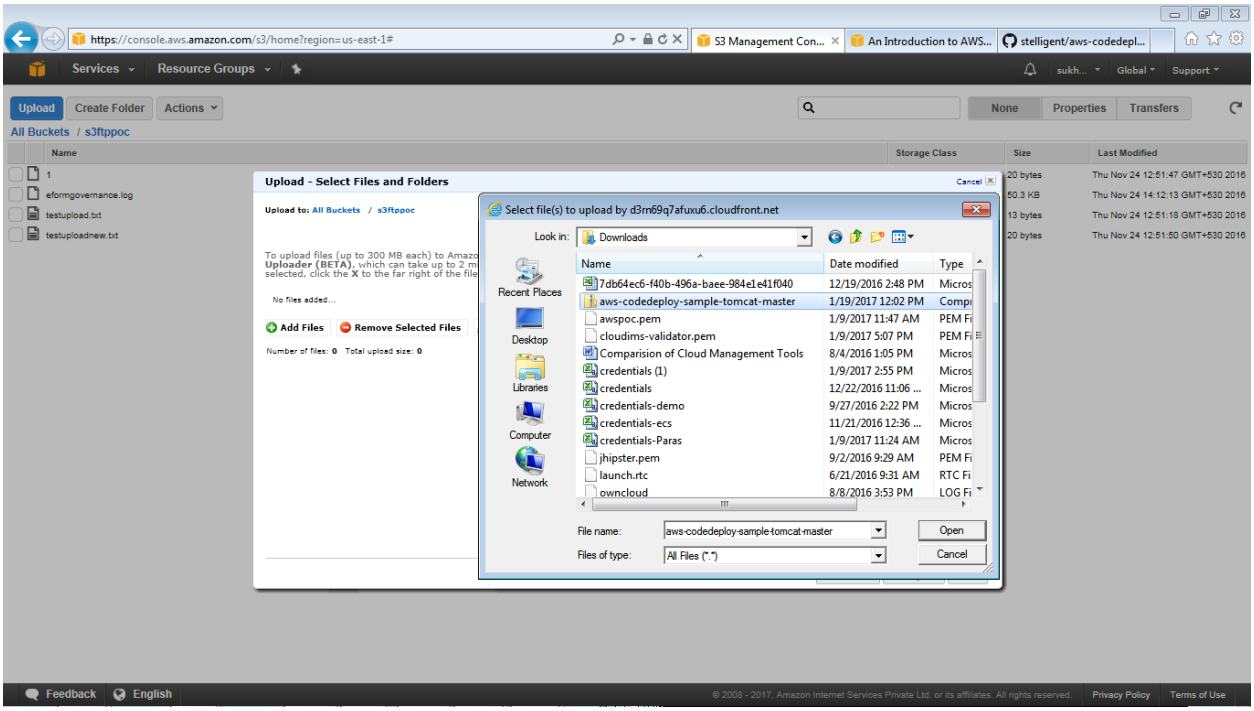
To understand the process for AWS CodeCommit Repository, AWS CodeBuild Project using java application and AWS Pipeline Creation.

17.2 Procedure

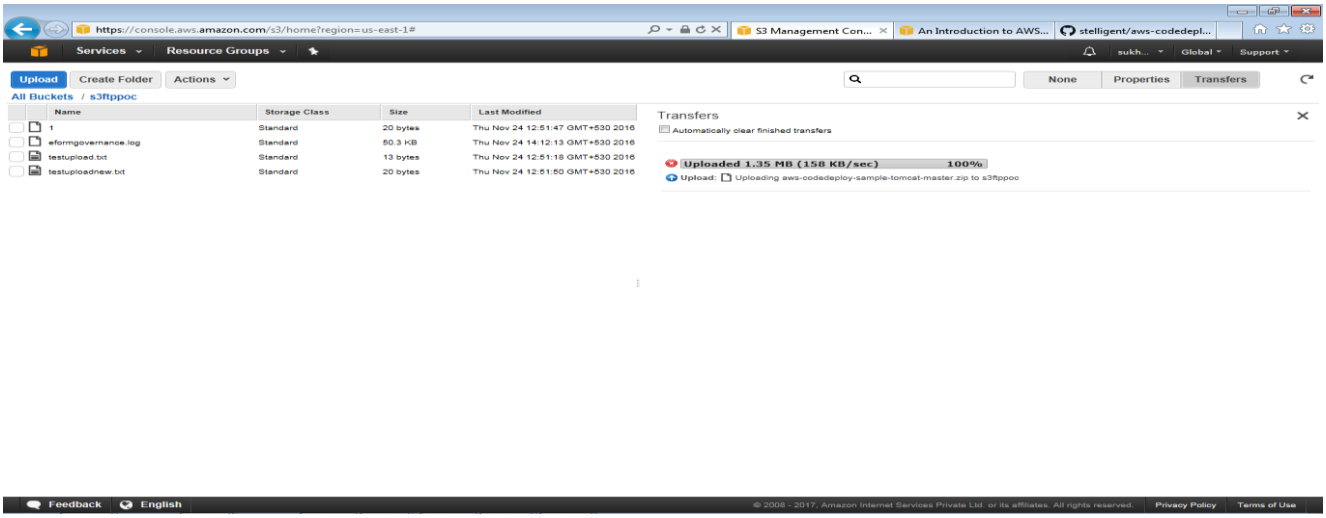
17.2.1 Configuring GitHub and AWS S3 bucket

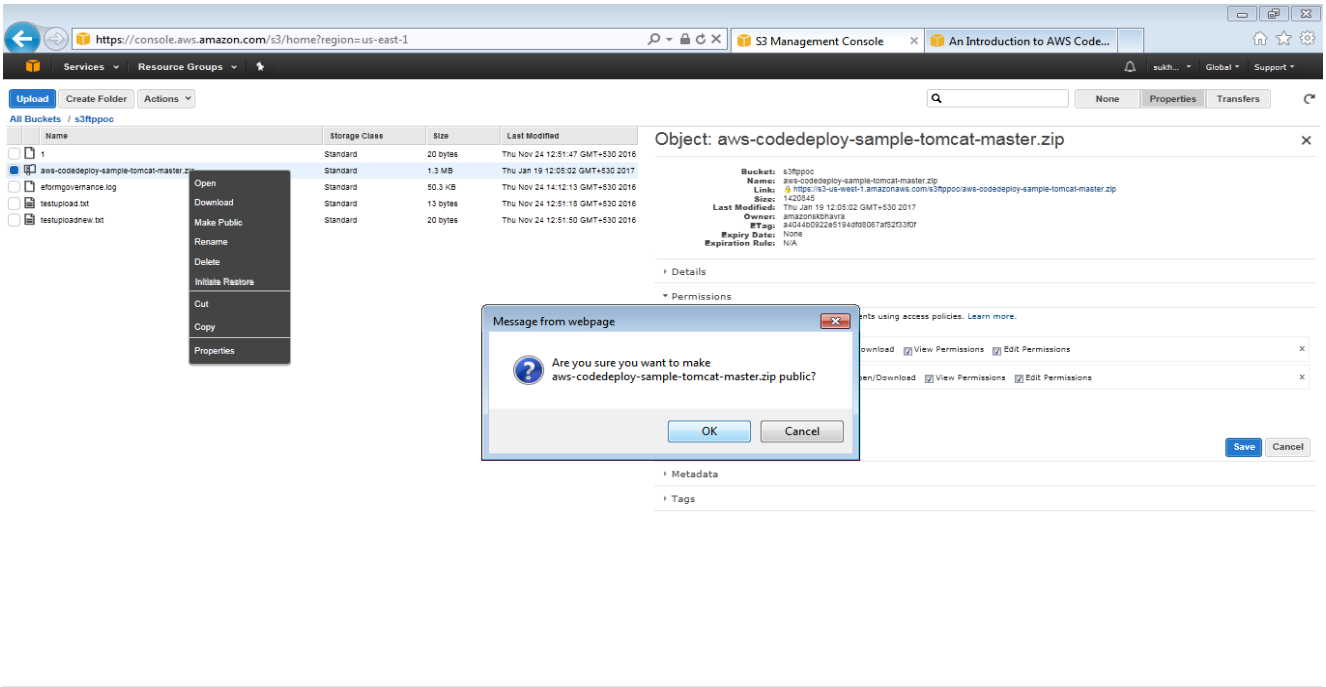
- Login to the GitHub and download the GitHub package: https://github.com/stelligent/aws-codedeploy-sample-tomcat

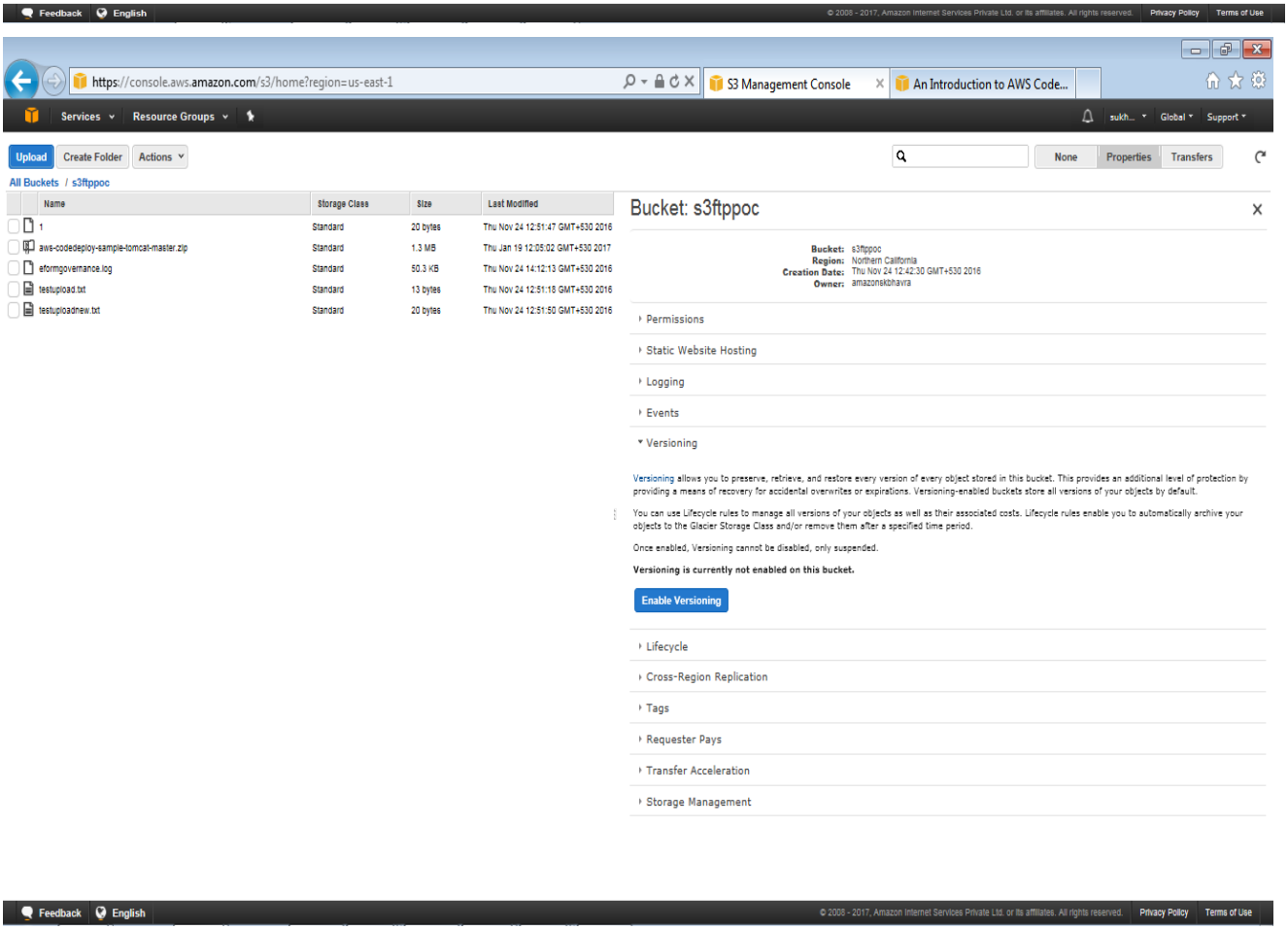


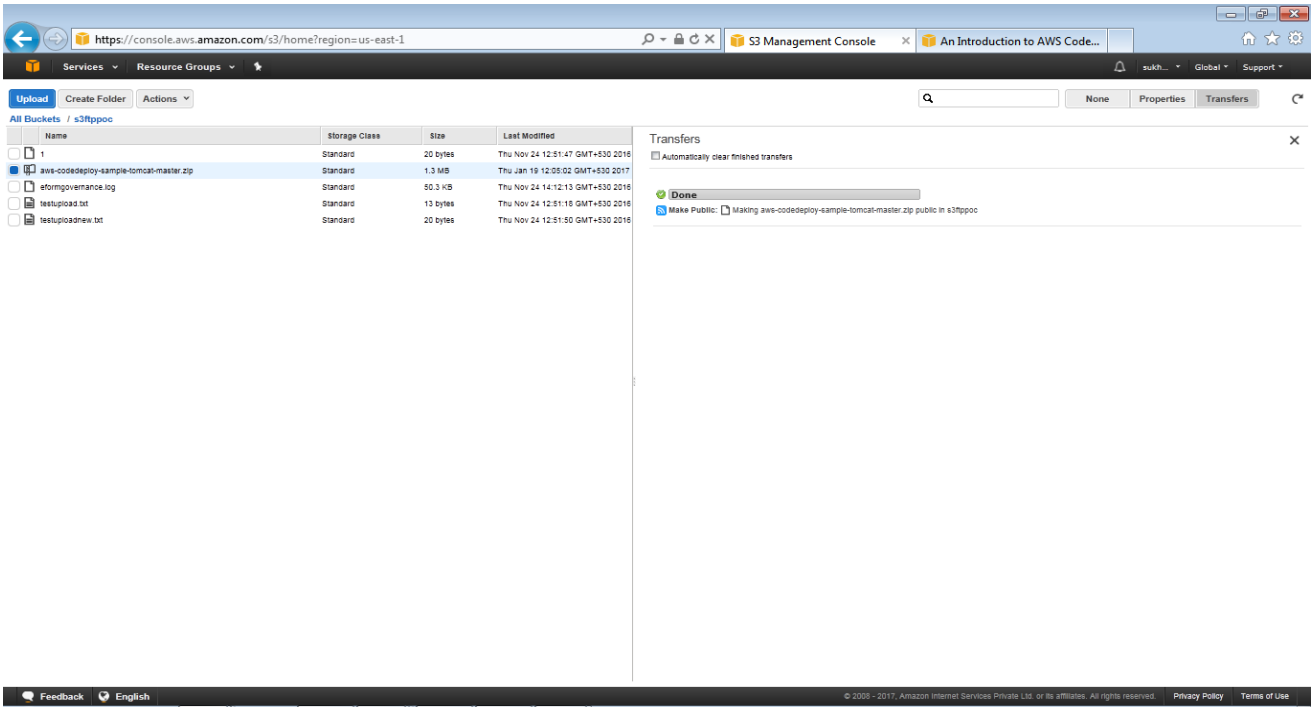


- Signin AWS Management Console
- Create a AWS S3 bucket. Be sure that S3 versioning is enabled for the bucket.
- Upload the GitHub package: https://github.com/stelligent/aws-codedeploy-sample-tomcat.**zip** to an S3 bucket.
- Make S3 bucket as public and **enable the versioning**



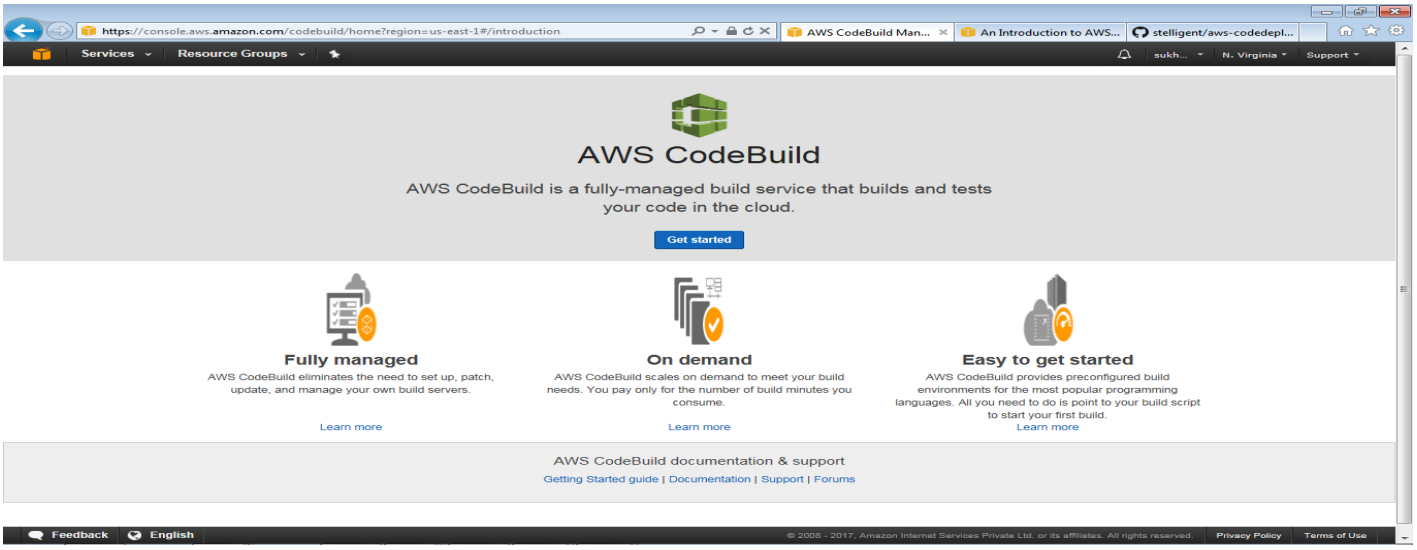

https://console.aws.amazon.com/s3/home?region=us-east-1
S3 Management Console
An Introduction to AWS Code...
Services
Resource Groups
Upload
Create Folder
Actions
None
Properties
Transfers
All Buckets / s3ftppoc
Name
Storage Class
Size
Last Modified
1
Standard
20 bytes
Thu Nov 24 12:51:47 GMT+530 2016
aws-codedeploy-sample-tomcat-master.zip
Standard
1.3 MB
Thu Jan 19 12:05:02 GMT+530 2017
eformgovernance.log
Standard
50.3 KB
Thu Nov 24 14:12:13 GMT+530 2016
testupload.txt
Standard
13 bytes
Thu Nov 24 12:51:18 GMT+530 2016
testuploadnew.txt
Standard
20 bytes
Thu Nov 24 12:51:50 GMT+530 2016
Open
Download
Make Public
Rename
Delete
Initials Restore
Cut
Copy
Properties
Object: aws-codedeploy-sample-tomcat-master.zip
Bucket: s3ftppoc
Name: aws-codedeploy-sample-tomcat-master.zip
Link: https://s3-us-west-1.amazonaws.com/s3ftppoc/aws-codedeploy-sample-tomcat-master.zip
Size: 1420845
Last Modified: Thu Jan 19 12:05:02 GMT+530 2017
Owner: amazon&dhavra
ETag: a4044b0922e5194dfd8067af62f33f0f
Expiry Date: None
Expiration Rule: N/A
Details
Permissions
Message from webpage
Are you sure you want to make aws-codedeploy-sample-tomcat-master.zip public?
OK
Cancel
Save
Cancel
Metadata
Tags

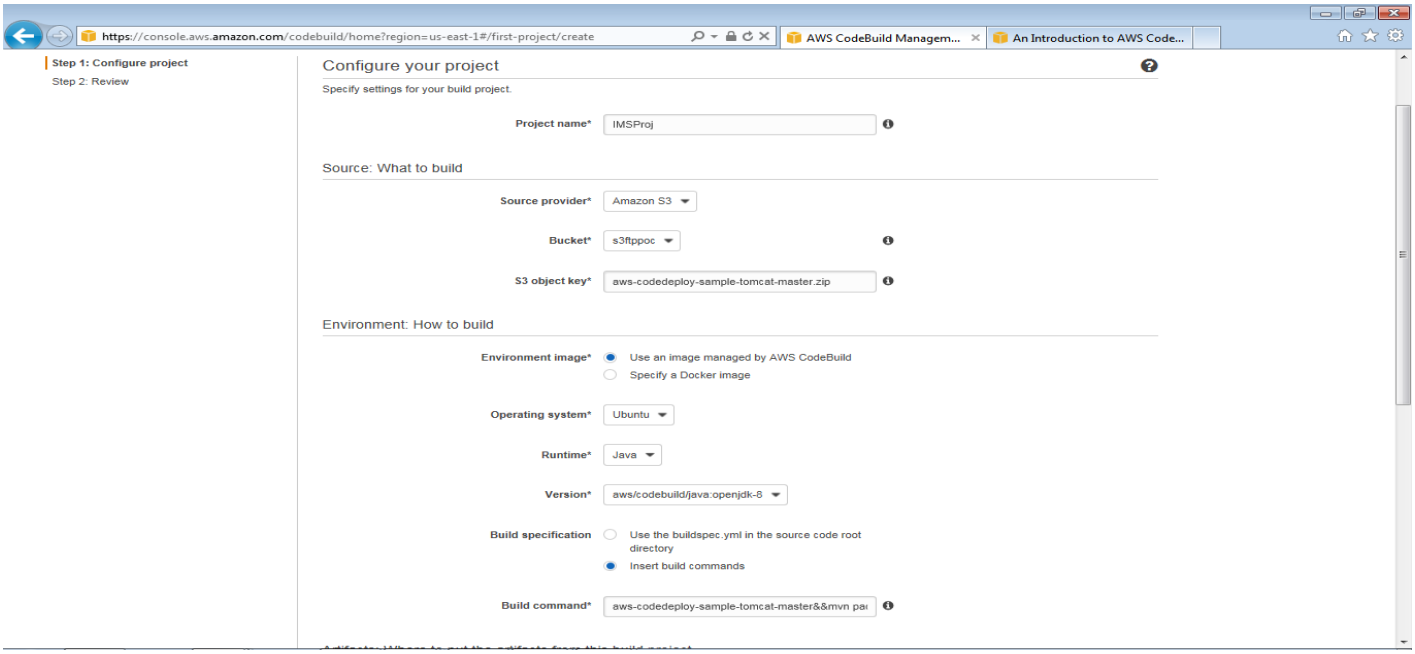

https://console.aws.amazon.com/s3/home?region=us-east-1
S3 Management Console
An Introduction to AWS Code...
Services
Resource Groups
Upload
Create Folder
Actions
None
Properties
Transfers
All Buckets / s3ftppoc
Name
Storage Class
Size
Last Modified
1
Standard
20 bytes
Thu Nov 24 12:51:47 GMT+530 2016
aws-codedeploy-sample-tomcat-master.zip
Standard
1.3 MB
Thu Jan 19 12:05:02 GMT+530 2017
eformgovernance.log
Standard
50.3 KB
Thu Nov 24 14:12:13 GMT+530 2016
testupload.txt
Standard
13 bytes
Thu Nov 24 12:51:18 GMT+530 2016
testuploadnew.txt
Standard
20 bytes
Thu Nov 24 12:51:50 GMT+530 2016
Bucket: s3ftppoc
Bucket: s3ftppoc
Region: Northern California
Creation Date: Thu Nov 24 12:42:30 GMT+530 2016
Owner: amazon&dhavra
Permissions
Static Website Hosting
Logging
Events
Versioning
Versioning allows you to preserve, retrieve, and restore every version of every object stored in this bucket. This provides an additional level of protection by providing a means of recovery for accidental overwrites or expirations. Versioning-enabled buckets store all versions of your objects by default.
You can use Lifecycle rules to manage all versions of your objects as well as their associated costs. Lifecycle rules enable you to automatically archive your objects to the Glacier Storage Class and/or remove them after a specified time period.
Once enabled, Versioning cannot be disabled, only suspended.
Versioning is currently not enabled on this bucket.
Enable Versioning
Lifecycle
Cross-Region Replication
Tags
Requester Pays
Transfer Acceleration
Storage Management



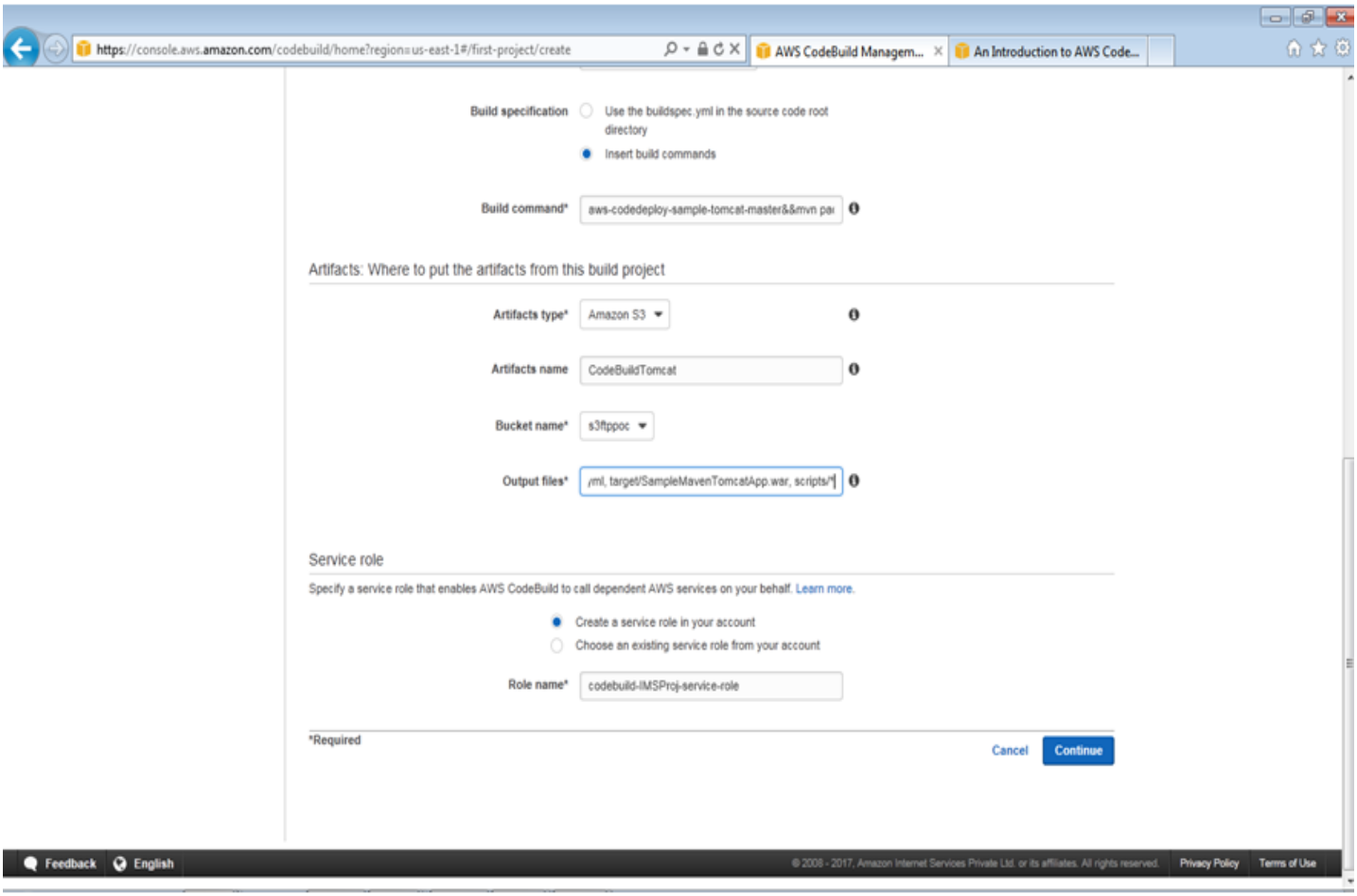
17.2.2. Configuring CodeBuild

1. Click on the AWS **Code Build** service and **Click on Get Started** button
2. Provide the following details





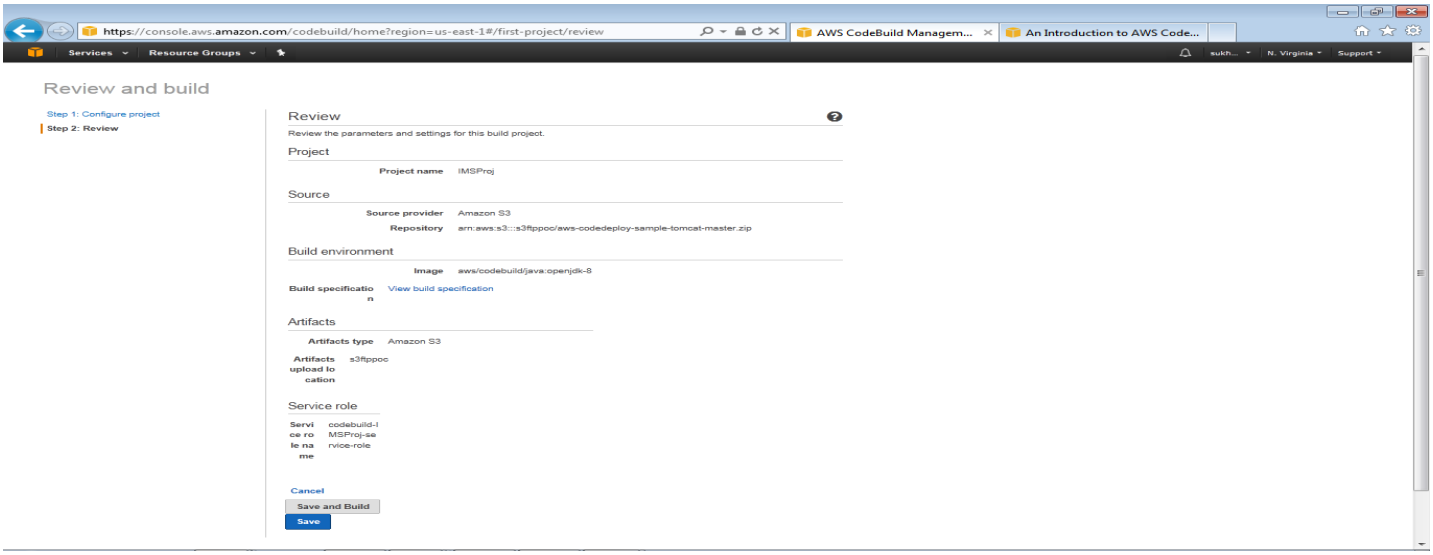
Build command value: cd aws-codedeploy-sample-tomcat-master && mvn package

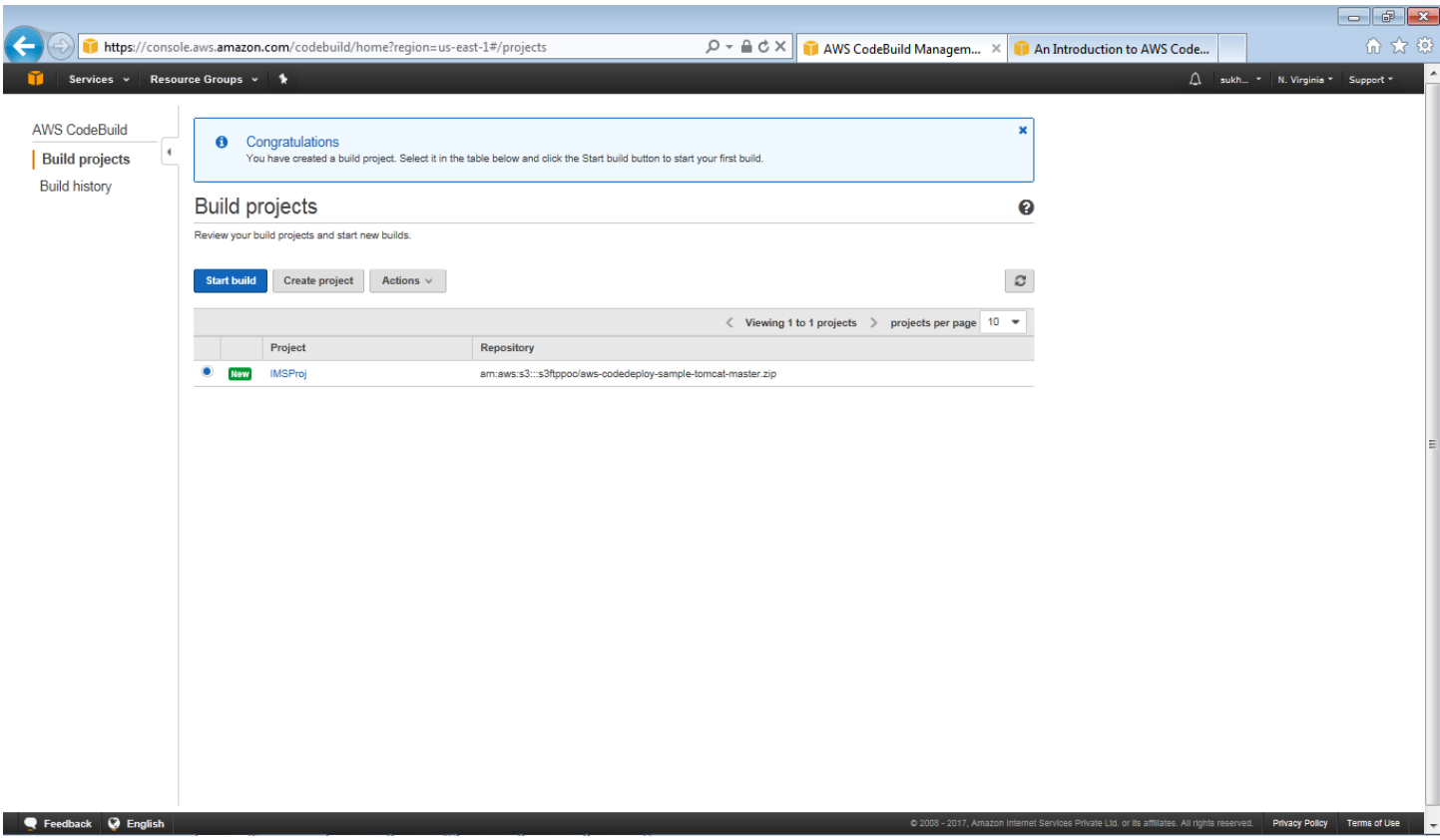


We are going to store the artifact in S3, with the name **CodeBuildTomcat**, and in the same bucket we used to store the source code.

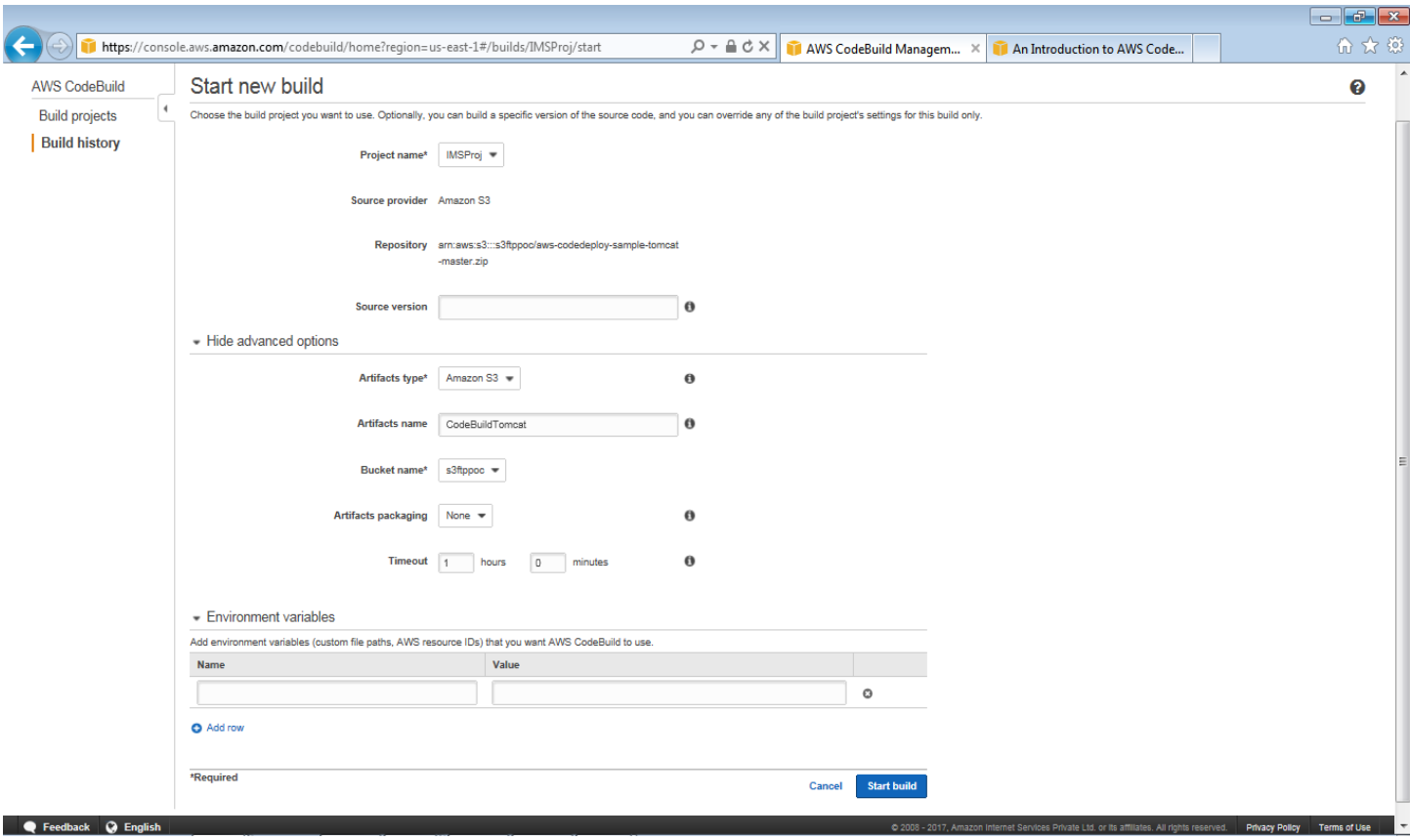
Output file value: appspec.yml, target/SampleMavenTomcatApp.war, scripts/*

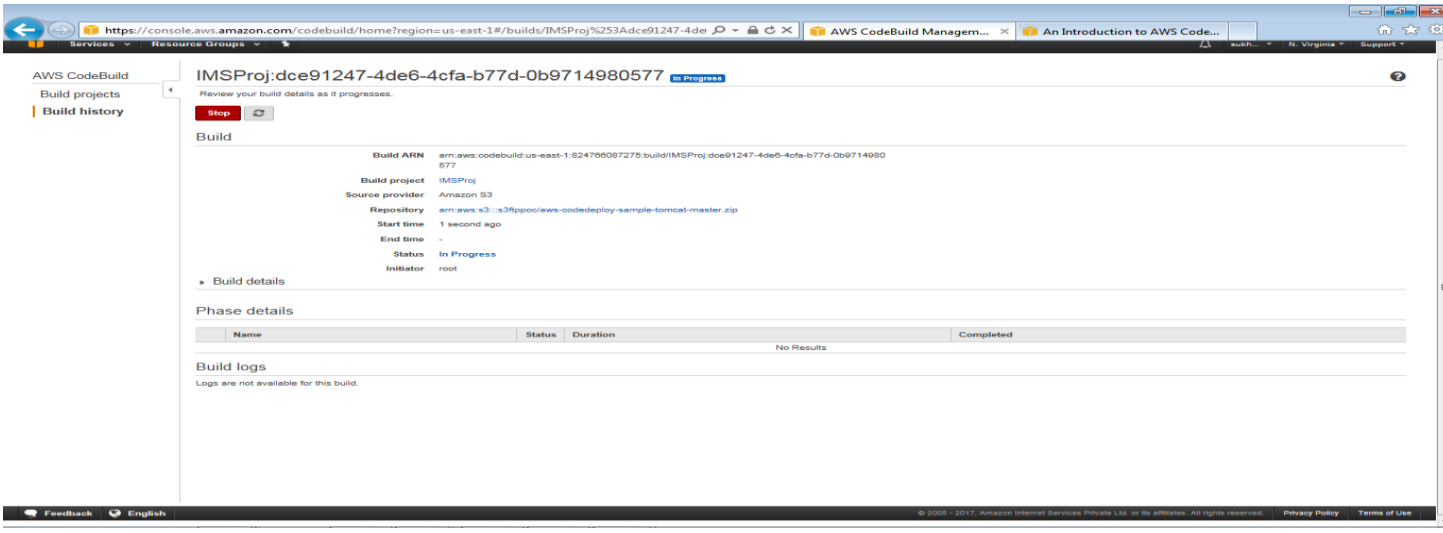
3. Click on Review and build the project



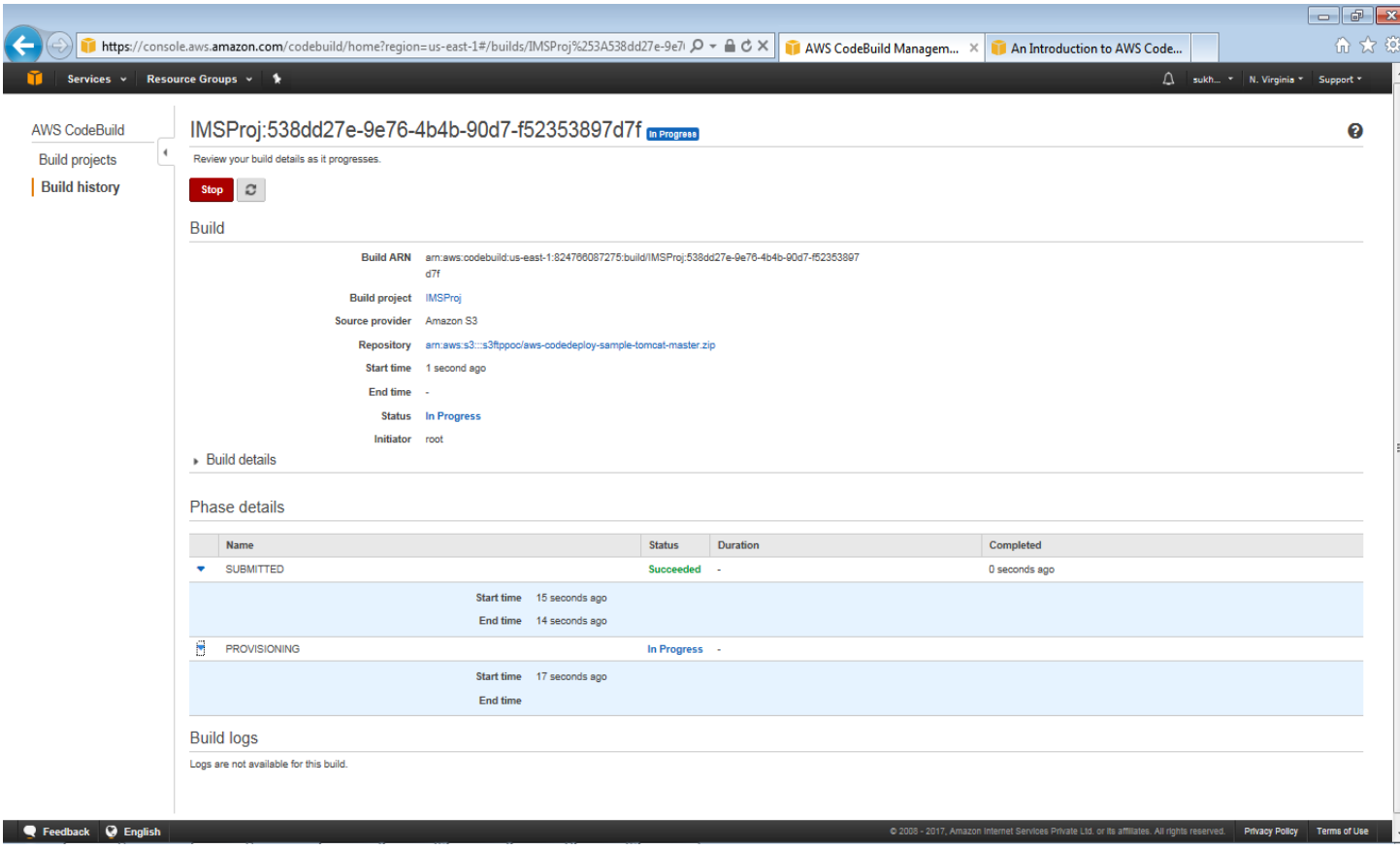


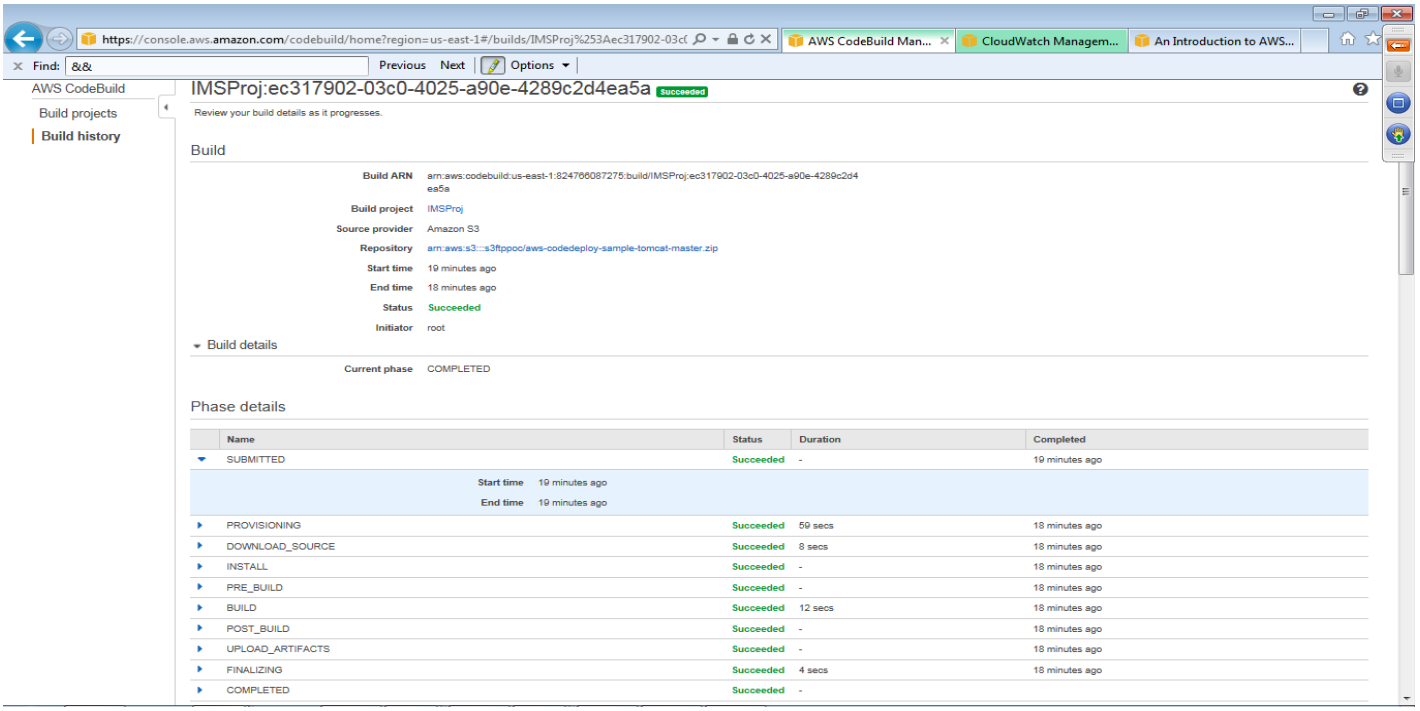
4. Click on Start build and check the status of the build



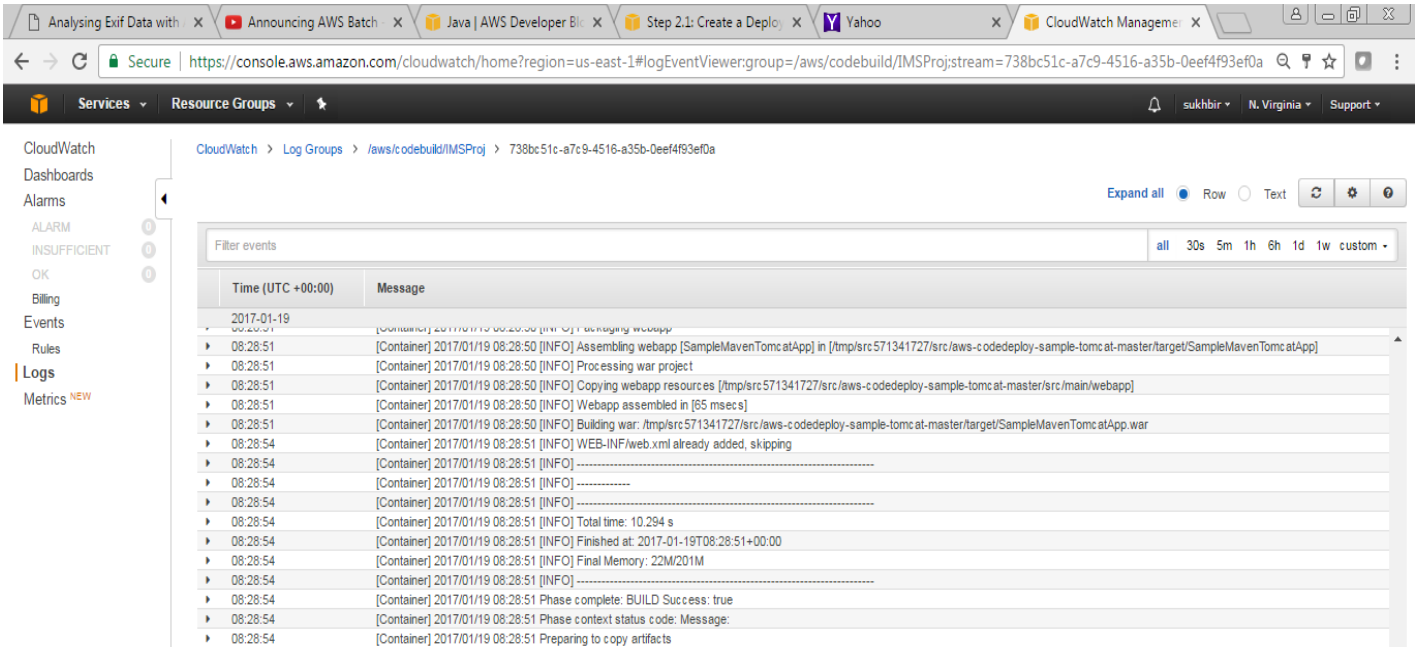


Wait for the build to succeed





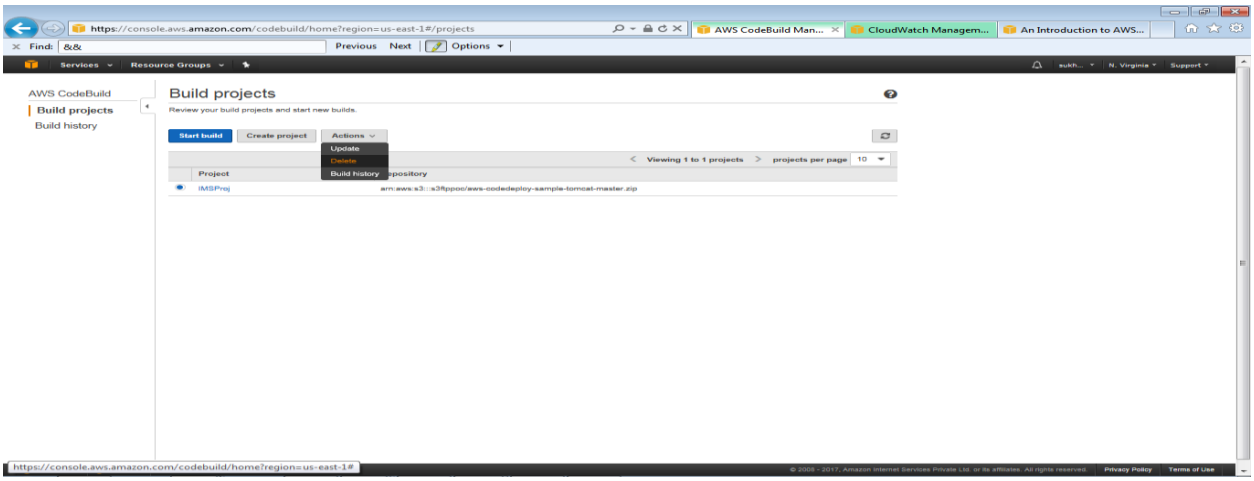
As CodeBuild is building artifacts, it goes through several distinct phases that are logged in AWS CloudWatch Logs.

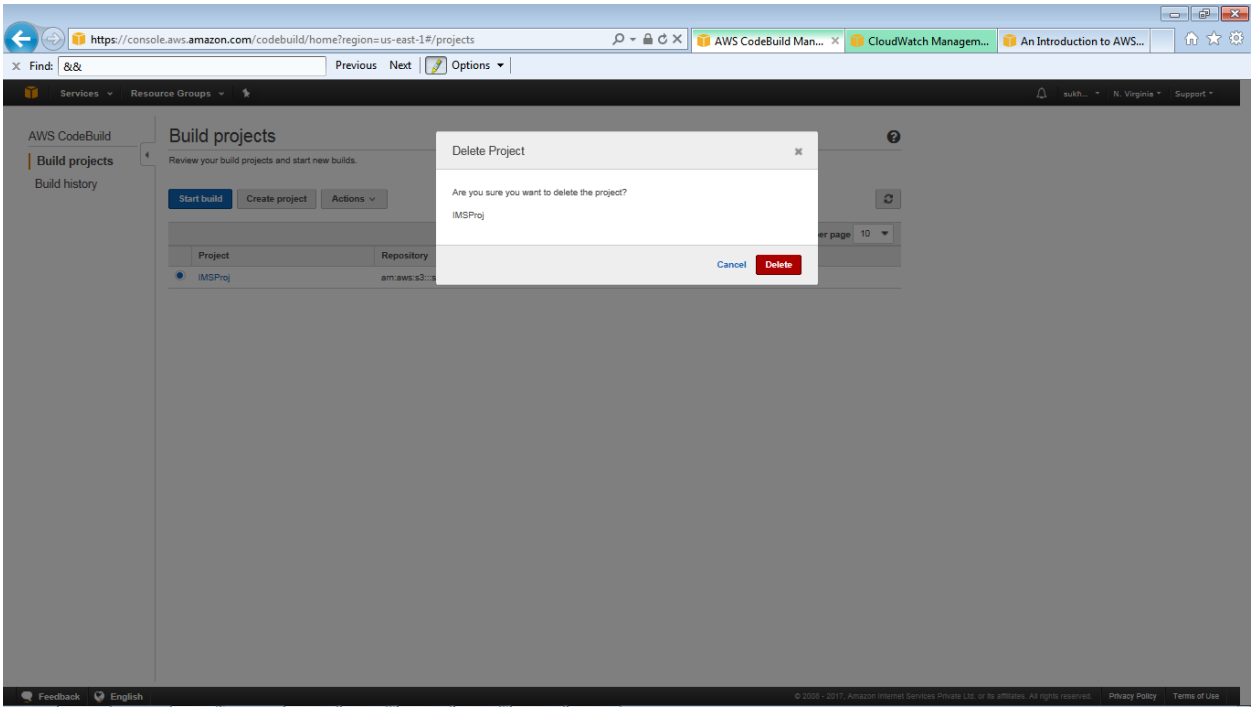


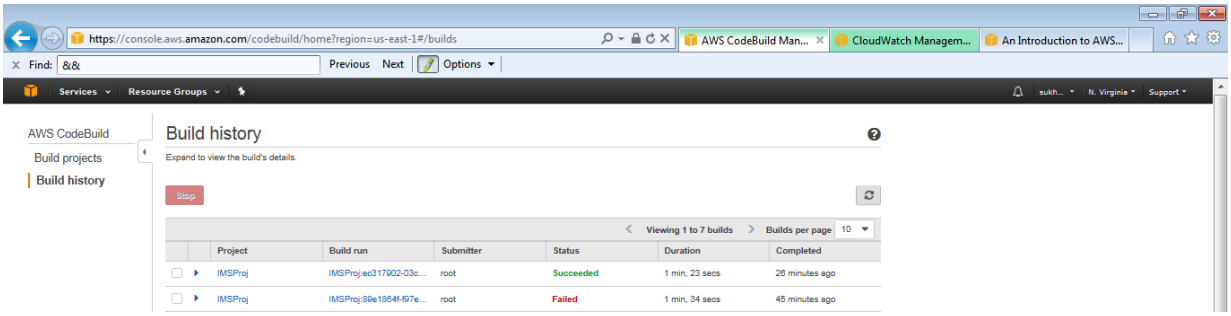
Once it has completed, you can open the S3 bucket you specified for the artifact and view the S3 bucket with the *SampleMavenTomcatApp.war* file.

Further, you can automate your release process by using AWS CodePipeline to run your builds with AWS CodeBuild.

17.2.3 Delete CodeBuild




https://console.aws.amazon.com/codebuild/home?region=us-east-1#/projects
AWS CodeBuild Man...
CloudWatch Managem...
An Introduction to AWS...
Find: &&
Previous
Next
Options
Services
Resource Groups
AWS CodeBuild
Build projects
Build history
Build projects
Review your build projects and start new builds.
Start build
Create project
Actions
Project
Repository
IMSProj
Delete Project
Are you sure you want to delete the project?
IMSProj
Cancel
Delete
Feedback
English
Privacy Policy
Terms of Use


https://console.aws.amazon.com/codebuild/home?region=us-east-1#/builds
AWS CodeBuild Man...
CloudWatch Managem...
An Introduction to AWS...
Find: &&
Previous
Next
Options
Services
Resource Groups
AWS CodeBuild
Build projects
Build history
Build history
Expand to view the build's details.
Stop
Viewing 1 to 7 builds
Builds per page
10
Project
Build run
Submitter
Status
Duration
Completed
IMSProj
IMSProj:ec317902-03c...
root
Succeeded
1 min, 23 secs
26 minutes ago
IMSProj
IMSProj:89e1864f-f97e...
root
Failed
1 min, 34 secs
45 minutes ago