

AWS Services

| Template details | |
|------------------|-----------------|
| AWS Services | AWS_Services2.0 |

1 Contents

| | |
|--|----|
| 1. EC2 MONITORING AND MANAGEMENT | 6 |
| 1.1 Objective | 6 |
| 1.2 Assumptions..... | 6 |
| 1.3 Procedure | 6 |
| 1.3.1 Create EC2 Instance | 6 |
| 1.3.2 Resizing the EC2 instance..... | 15 |
| 1.3.3 Modify EC2 instance | 18 |
| 1.3.4 Terminating EC2 instance..... | 20 |
| 1.2.5 MONITORING EC2 INSTANCE | 22 |
| 2. VPC MANAGEMENT | 25 |
| 2.1 Objective | 25 |
| 2.2 Assumptions..... | 25 |
| 2.3 Procedure | 26 |
| 2.3.1 Creation of new VPC | 27 |
| 2.3.2 Subnets..... | 27 |
| 2.3.3 NAT Gateway..... | 29 |
| 2.3.4 Configuration of AWS VPN devices and on premise VPN devices..... | 31 |
| 3. IDENTITY AND SECURITY MANAGEMENT | 35 |
| 3.1 Objective | 35 |
| 3.2 Assumption..... | 35 |
| 3.3 Procedure | 35 |
| 4. RDS SUPPORT | 43 |
| 4.1 Objective | 43 |
| 4.2 Assumption..... | 43 |
| 4.3 Procedure | 43 |
| 5. ELASTICACHE MANAGEMENT..... | 47 |
| 5.1 Objective | 47 |
| 5.2 Assumption..... | 47 |
| 5.3 Procedure | 47 |
| 5.3.1 Launch a Cluster | 47 |

| | |
|---|----|
| 5.3.2 View Cluster Details | 49 |
| 5.3.3 Authorize Access..... | 51 |
| 5.3.4 Connect to a Cluster's Node..... | 52 |
| 5.3.5 Delete Your Cluster [Avoid Unnecessary Charges] | 53 |
| 6. SES MANAGEMENT | 54 |
| 6.1 Objective | 54 |
| 6.2 Assumption..... | 54 |
| 6.3 Procedure | 54 |
| 6.3.1 Verifying Email Address | 55 |
| 6.3.2 Verifying Domain Address | 56 |
| 6.3.3 Using SMTP interface to Send Email..... | 57 |
| 6.3.4 Receiving Email..... | 58 |
| 6.3.5 Monitoring Your Sending Activity..... | 61 |
| 6.3.6 AWS SES Limits..... | 63 |
| 7. SQS MANAGEMENT | 63 |
| 7.1 Objective | 63 |
| 7.2 Procedure | 63 |
| 7.2.1 Create New Queue | 63 |
| 7.2.2 Sending Message..... | 66 |
| 7.2.3 Receiving Message..... | 68 |
| 7.2.4 Modify Queue..... | 69 |
| 7.2.5 Monitoring Queue..... | 70 |
| 7.2.6 Deleting a Message | 71 |
| 7.2.7 Purging Queue..... | 72 |
| 8. SNS MANAGEMENT | 73 |
| 8.1 Objective | 73 |
| 8.2 Assumptions..... | 73 |
| 8.3 Procedure | 74 |
| 8.3.1 Create SNS(Simple Notification Service). | 74 |
| 8.3.2 Modifying SNS(Simple Notification Service)..... | 80 |
| 9.1 Objective | 81 |
| 9.2 Procedure | 82 |

| | |
|---|-----|
| 9.2.1 Create Bucket | 82 |
| 9.2.2 Delete Bucket..... | 84 |
| 9.2.4 Delete Folder..... | 85 |
| 10. EFS MANAGEMENT | 86 |
| 10.1 Objective | 86 |
| 10.2 Assumption..... | 87 |
| 10.3 Procedure | 87 |
| 10.3.1 Create an EC2 Instance..... | 87 |
| 10.3.2 Create Amazon EFS File System..... | 93 |
| 10.3.3 Connect to Your Amazon EC2 Instance and Mount the Amazon EFS File System.. | 96 |
| 11. CLOUD FORMATION..... | 100 |
| 11.1 Objective | 100 |
| 11.2.1 Create the Initial Stack | 100 |
| 11.2.2 Update the Application | 101 |
| 11.2.3 Updating Auto Scaling Groups | 106 |
| 11.2.4 Changing Resource Properties..... | 107 |
| 11.2.5 Update the Instance Type..... | 107 |
| 11.2.6 Update the AMI on an Amazon EC2 instance | 110 |
| 11.2.6 Update the Amazon EC2 Launch Configuration for an Auto Scaling Group..... | 110 |
| 11.2.7 Adding Resource Properties..... | 110 |
| 11.2.8 Add a Key Pair to an Instance..... | 110 |
| 11.2.9 Change the Stack's Resources..... | 112 |
| 12. CLOUDWATCH MANAGEMENT..... | 124 |
| 12.1 Objective | 124 |
| 12.2 Assumptions..... | 124 |
| 12.3 Procedure | 124 |
| 12.3.1 Enable Billing Alerts..... | 124 |
| 12.3.2 Create a Billing Alarm..... | 125 |
| 12.3.3 Check the Alarm Status..... | 126 |
| 12.3.4 Edit a Billing Alarm | 127 |
| 12.3.5 Delete a Billing Alarm..... | 127 |
| 13. ROUTE 53 MANAGEMENT..... | 128 |

| | |
|---|------------|
| 13.1 Objective | 128 |
| 13.2 Assumptions..... | 128 |
| 13.3 Procedure | 128 |
| 14. CLOUDFRONT MANAGEMENT | 141 |
| 14.1 Objective | 141 |
| 14.2 Procedure..... | 141 |
| 15. OPSWORKS | 146 |
| 15.1 AWS OpsWork (Chef Automate) | 146 |
| 15.1.1 Objective..... | 146 |
| 15.1.2 Assumptions | 146 |
| 15.1.3 Procedure..... | 147 |
| 15.1.3.1 Integrating GitHub Repository (Hosted) with Chef Automate Server | 156 |
| 15.2 AWS OpsWork Stacks..... | 164 |
| 15.2.1 Objective..... | 164 |
| 15.2.2 Pre-requisite..... | 164 |
| 15.2.3 Procedure..... | 164 |
| 16. AWS Lambda | 183 |
| 16.1 Objective | 183 |
| 16.2 Procedure..... | 183 |
| 16.2.1 Creating Lamda Function | 183 |
| 16.2.2 Testing Lamda Function | 187 |
| 17. APPLICATION & DEVELOPER TOOL SUPPORT..... | 188 |
| 17.1 Objective | 188 |
| 17.2 Procedure..... | 188 |
| 17.2.1 Configuring GitHub and AWS S3 bucket | 188 |
| 17.2.2. Configuring CodeBuild | 191 |
| 17.2.3 Delete CodeBuild | 196 |

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 Security 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 Elastic IPs |
| 0 Dedicated Hosts | 1 Snapshots |
| 0 Volumes | 1 Load Balancers |
| 3 Key Pairs | 3 Security Groups |
| 0 Placement 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):**

✓ US East (Ohio): This service is operating normally

Scheduled Events

No events

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Feedback English

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)

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

| | | |
|--|---|---|
| <input type="checkbox"/> My AMIs | <input checked="" type="checkbox"/> Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-8a7859ef | Select |
| <input type="checkbox"/> AWS Marketplace | 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 |
| <input type="checkbox"/> Community AMIs | | |
| <input type="checkbox"/> Free tier only | | |
| | <input type="checkbox"/> SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-61a7fd04 | Select |
| | 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 |
| | <input type="checkbox"/> Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type - ami-11aa8c74 | Select |
| | 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 |

Cancel and Exit

1 to 32 of 32 AMIs

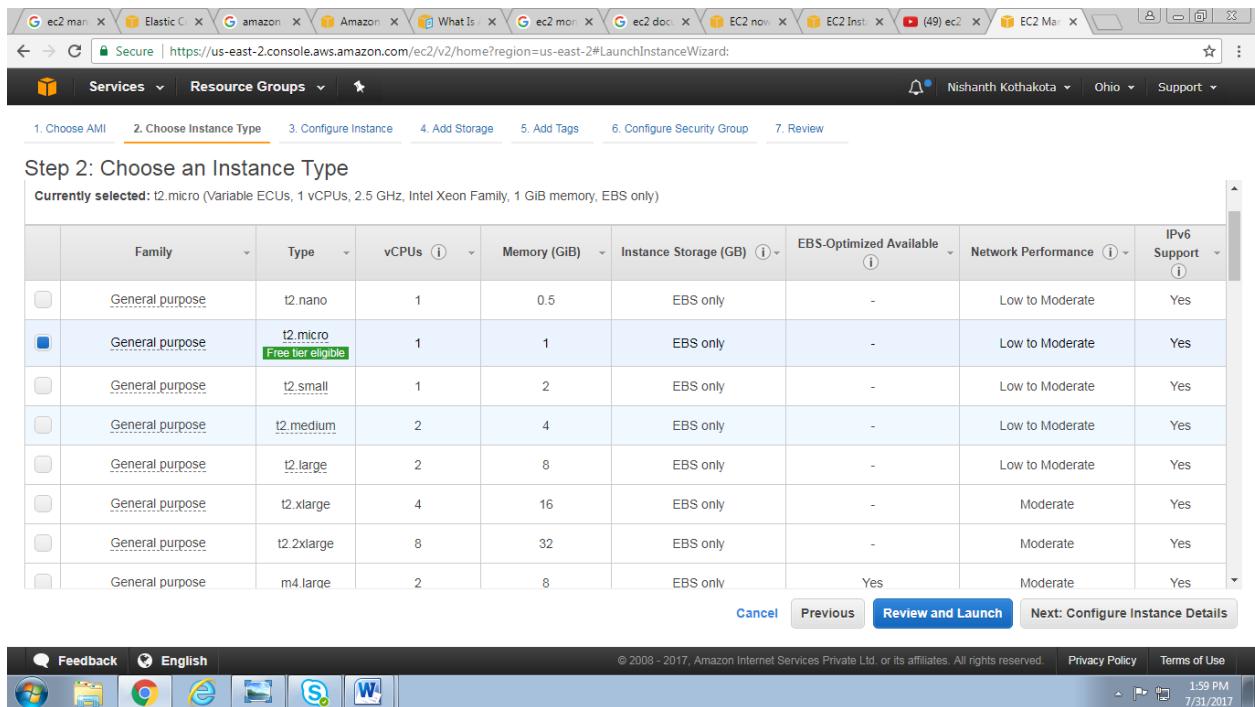
Secure | https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#

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



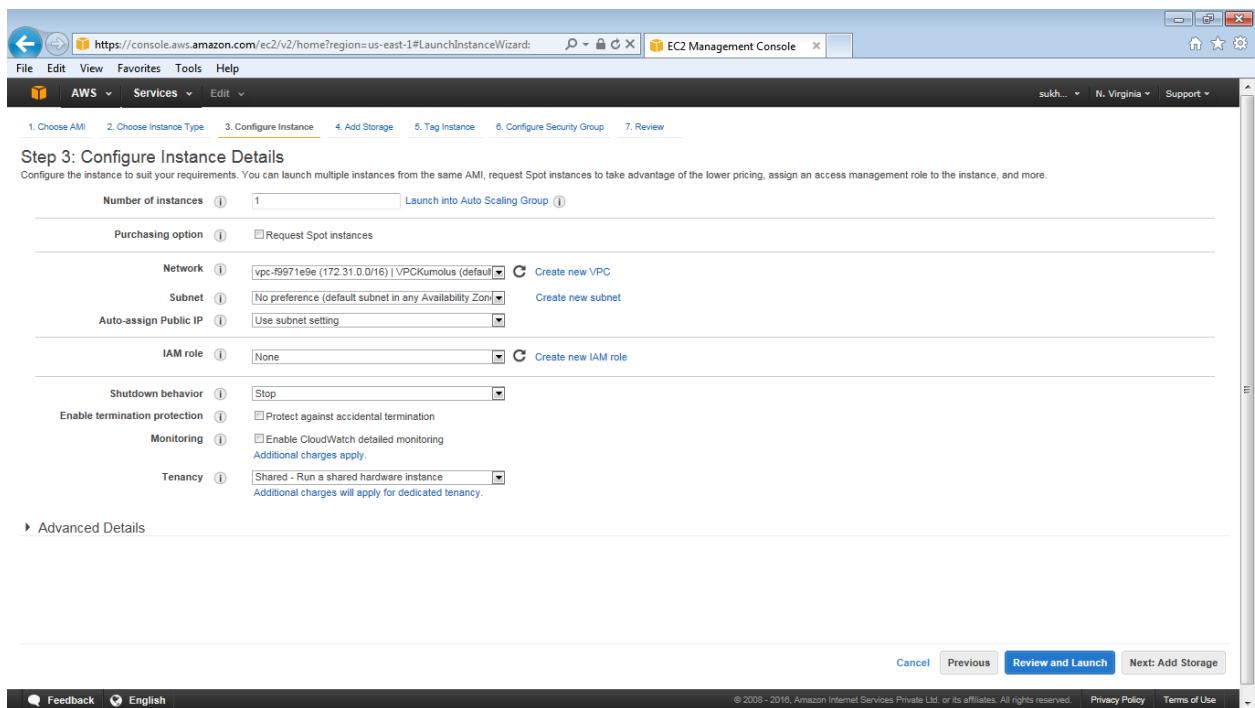
Step 2: Choose an Instance Type

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|-------------------------------------|-----------------|--------------------------------|-------|--------------|-----------------------|-------------------------|---------------------|--------------|
| <input type="checkbox"/> | General purpose | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate | Yes |
| <input checked="" type="checkbox"/> | General purpose | t2.micro Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.small | 1 | 2 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.medium | 2 | 4 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.large | 2 | 8 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.xlarge | 4 | 16 | EBS only | - | Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.2xlarge | 8 | 32 | EBS only | - | Moderate | Yes |
| <input type="checkbox"/> | General purpose | m4.large | 2 | 8 | EBS only | Yes | Moderate | Yes |

Cancel Previous Review and Launch Next: Configure Instance Details

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



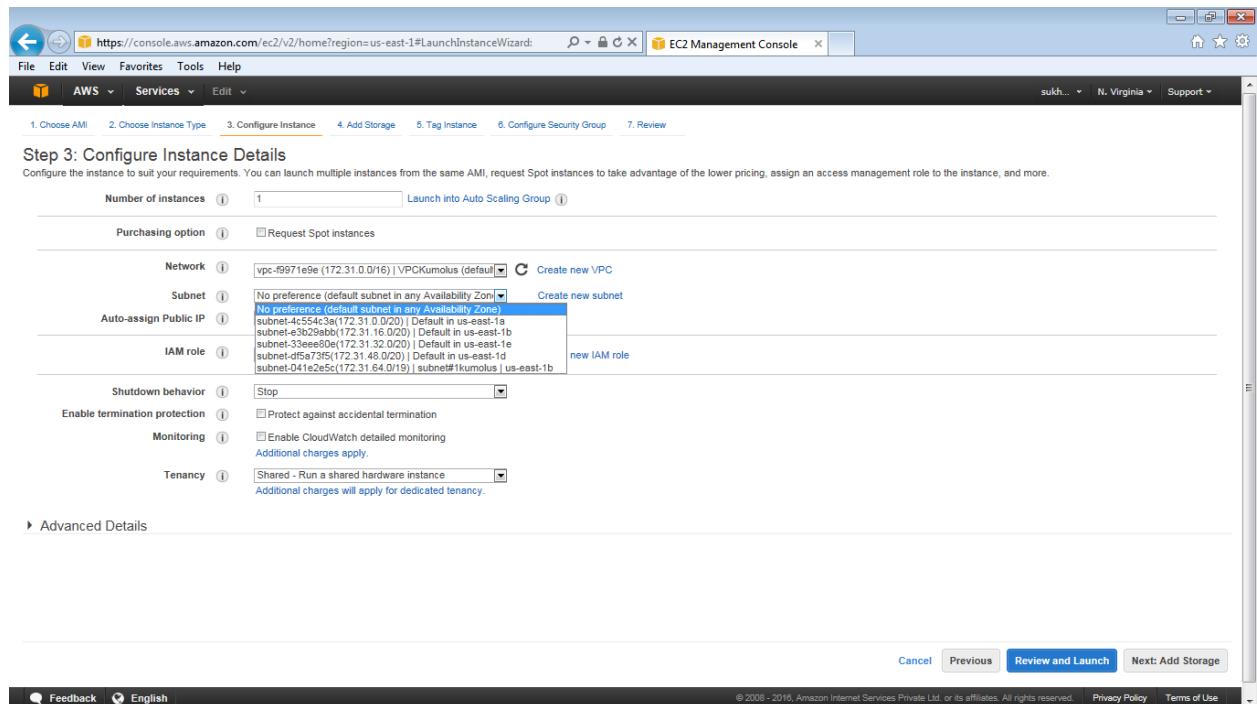
Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

| | | |
|-------------------------------|---|--------------------------------|
| Number of instances | 1 | Launch into Auto Scaling Group |
| Purchasing option | <input type="checkbox"/> Request Spot instances | |
| Network | vpc-f9971e9e (172.31.0.0/16) VPC Kumolus (default) <input type="checkbox"/> Create new VPC | |
| Subnet | No preference (default subnet in any Availability Zone) <input type="checkbox"/> Create new subnet | |
| Auto-assign Public IP | Use subnet setting | |
| IAM role | None <input type="checkbox"/> Create new IAM role | |
| Shutdown behavior | Stop | |
| Enable termination protection | <input type="checkbox"/> Protect against accidental termination | |
| Monitoring | <input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply. | |
| Tenancy | Shared - Run a shared hardware instance <input type="checkbox"/> Additional charges will apply for dedicated tenancy. | |
| Advanced Details | | |

Cancel Previous Review and Launch Next: Add Storage

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



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-f9971e9e (172.31.0.0/16) | VPCKumulus (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP: subnets-4c5543b (172.31.0.0/20) Default in us-east-1a
subnets-e30295b (172.31.16.0/20) Default in us-east-1b
subnets-33ee80a (172.31.32.0/20) Default in us-east-1e
subnets-df573f5 (172.31.48.0/20) Default in us-east-1d
subnets-241e2e5c (172.31.64.0/19) subnet#tkumulus#us-east-1b

IAM role: new IAM role

Shutdown behavior: Stop

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring Additional charges apply.

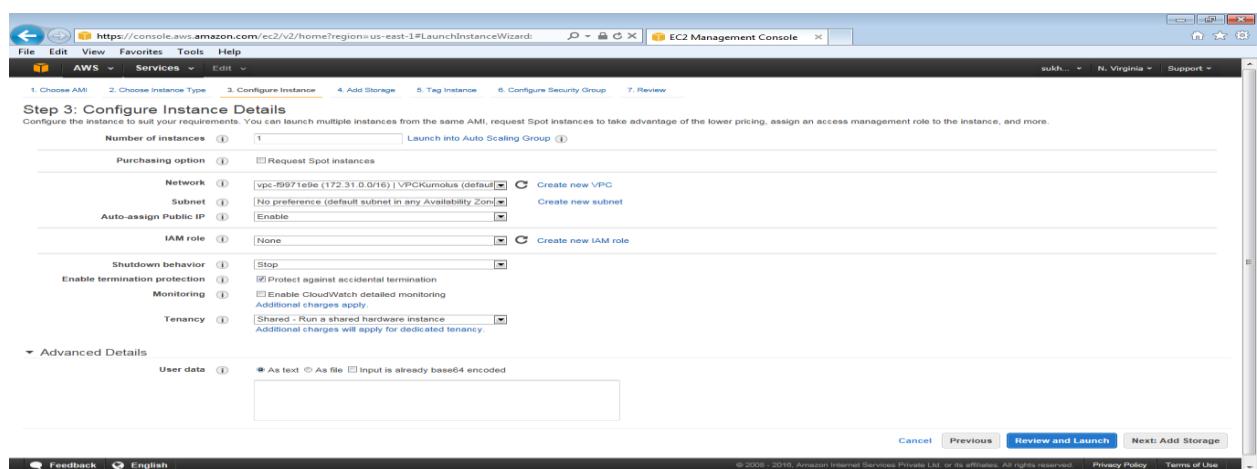
Tenancy: Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.

Advanced Details

Cancel Previous Review and Launch Next: Add Storage

Feedback English © 2008 - 2010, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-f9971e9e (172.31.0.0/16) | VPCKumulus (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP: Enable

IAM role: None Create new IAM role

Shutdown behavior: Stop

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring Additional charges apply.

Tenancy: Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.

Advanced Details

User data: As text As file Input is already base64 encoded

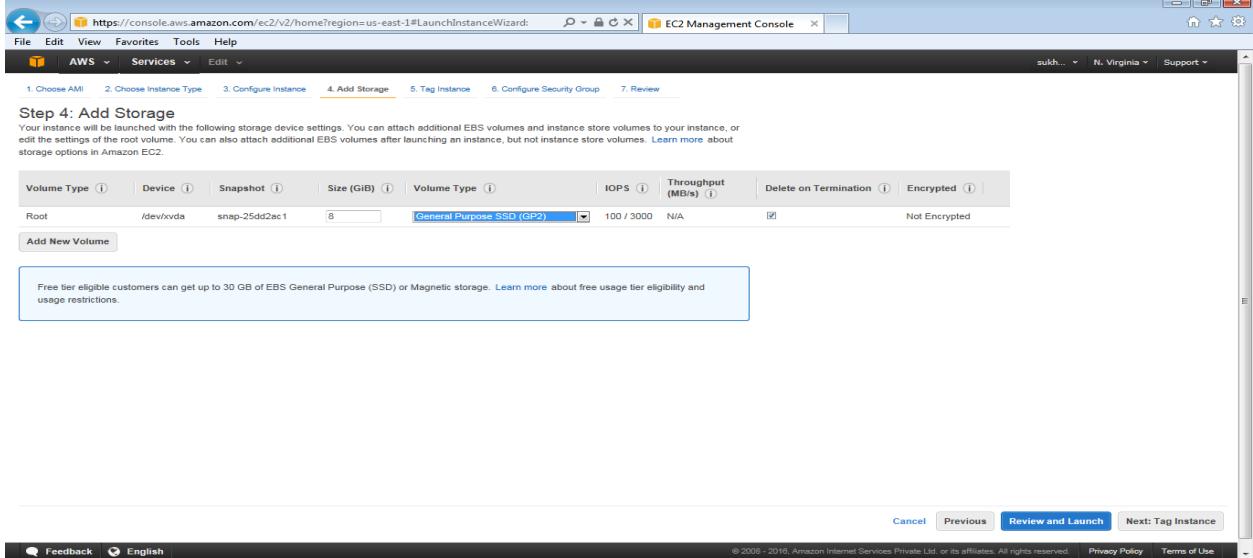
Cancel Previous Review and Launch Next: Add Storage

Feedback English © 2008 - 2010, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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/>



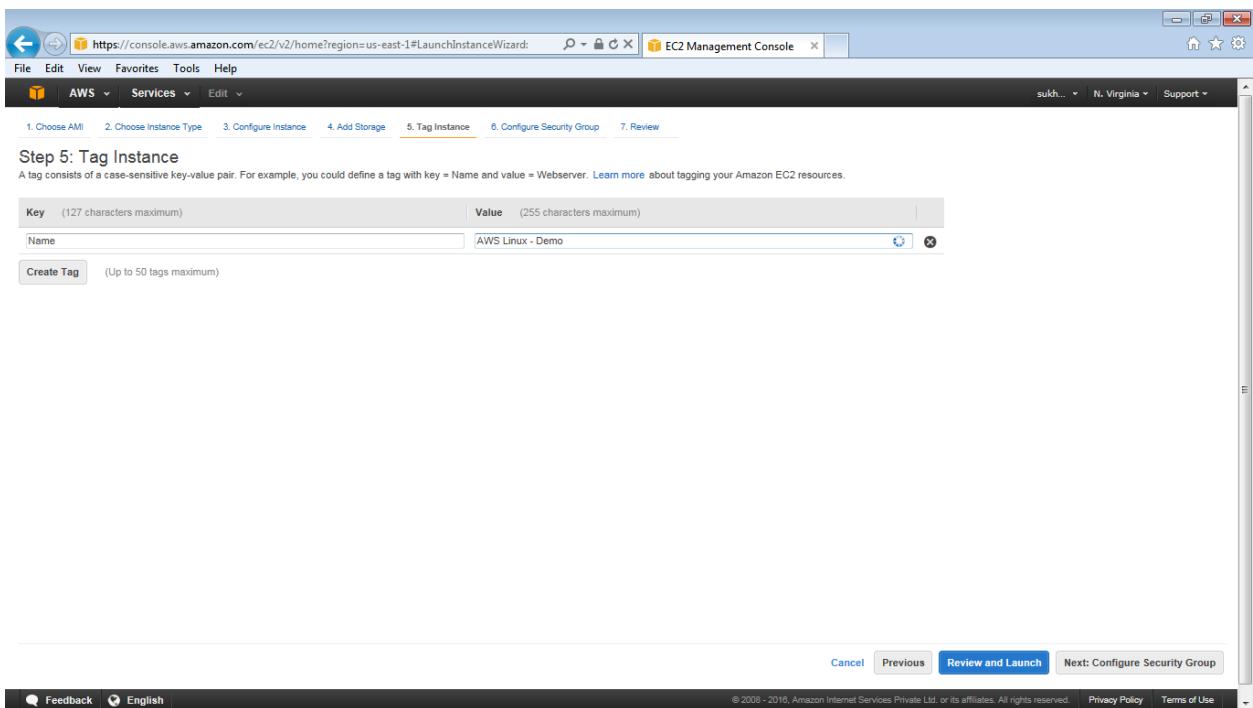
| Volume Type | Device | Snapshot | Size (GiB) | Volume Type | IOPS | Throughput (MB/s) | Delete on Termination | Encrypted |
|-------------|-----------|---------------|------------|---------------------------|------------|-------------------|-------------------------------------|---------------|
| Root | /dev/xvda | snap-25dd2ec1 | 8 | General Purpose SSD (GP2) | 100 / 3000 | N/A | <input checked="" type="checkbox"/> | Not Encrypted |

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Tag Instance

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



Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

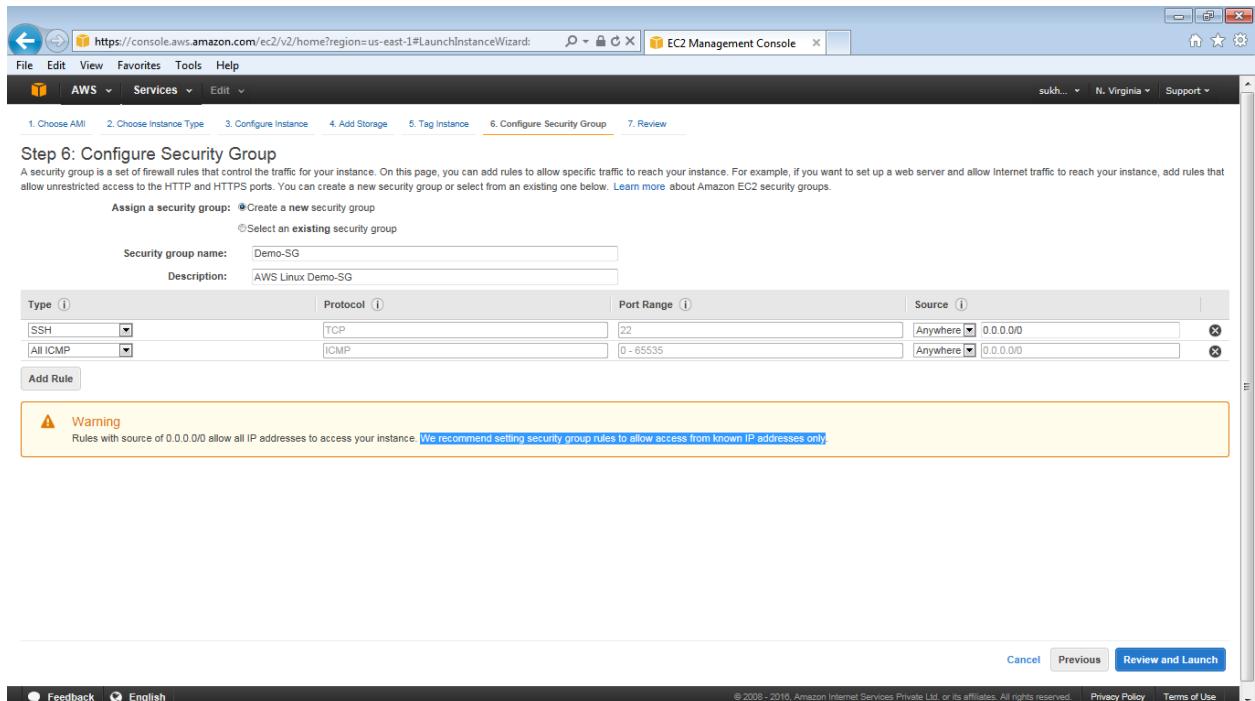
| | |
|------------------------------|--------------------------------|
| Key (127 characters maximum) | Value (255 characters maximum) |
| Name | AWS Linux - Demo |

Create Tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

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

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



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group Select an existing security group

Security group name: Demo-SG
Description: AWS Linux Demo-SG

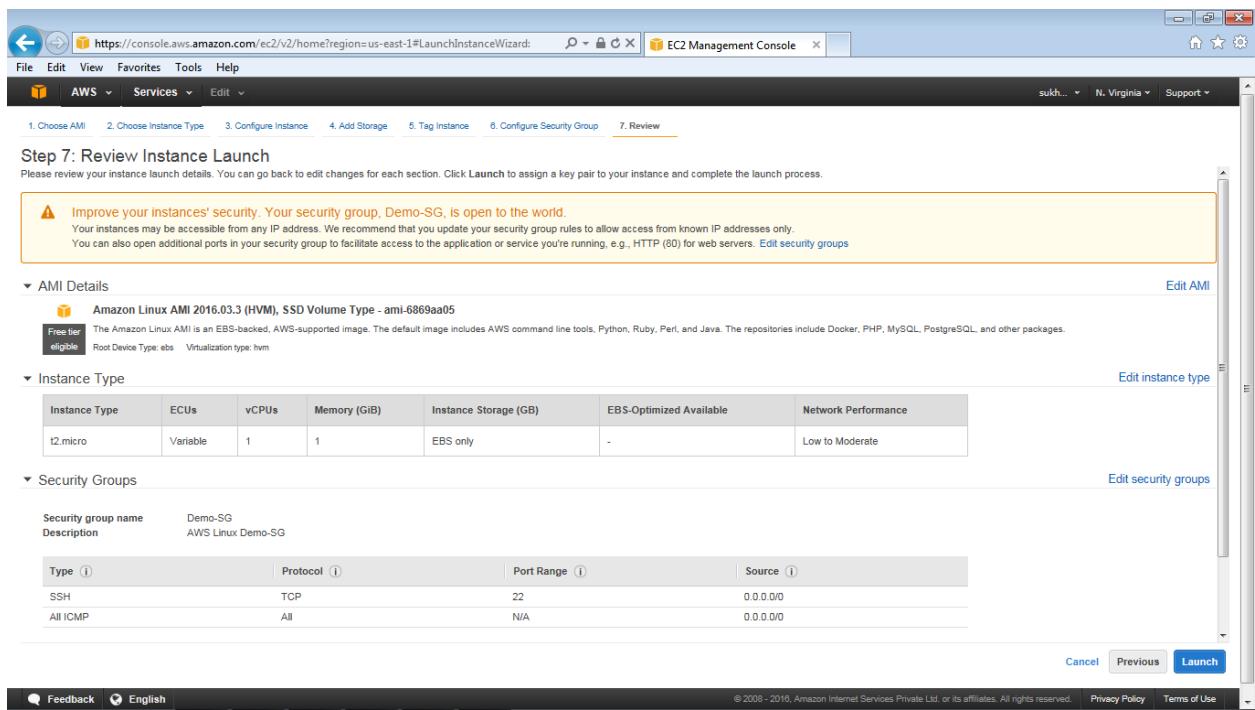
| Type | Protocol | Port Range | Source |
|----------|----------|------------|----------------------|
| SSH | TCP | 22 | Anywhere (0.0.0.0/0) |
| All ICMP | ICMP | 0 - 65535 | Anywhere (0.0.0.0/0) |

[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. [We recommend setting security group rules to allow access from known IP addresses only.](#)

Cancel Previous [Review and Launch](#)

- Review the instance details



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.

AMI Details
Amazon Linux AMI 2016.03 (HVM), SSD Volume Type - ami-6869aa05
Free tier eligible
Root Device Type: ebs Virtualization type: hvm

Instance Type

| Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|----------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro | Variable | 1 | 1 | EBS only | - | Low to Moderate |

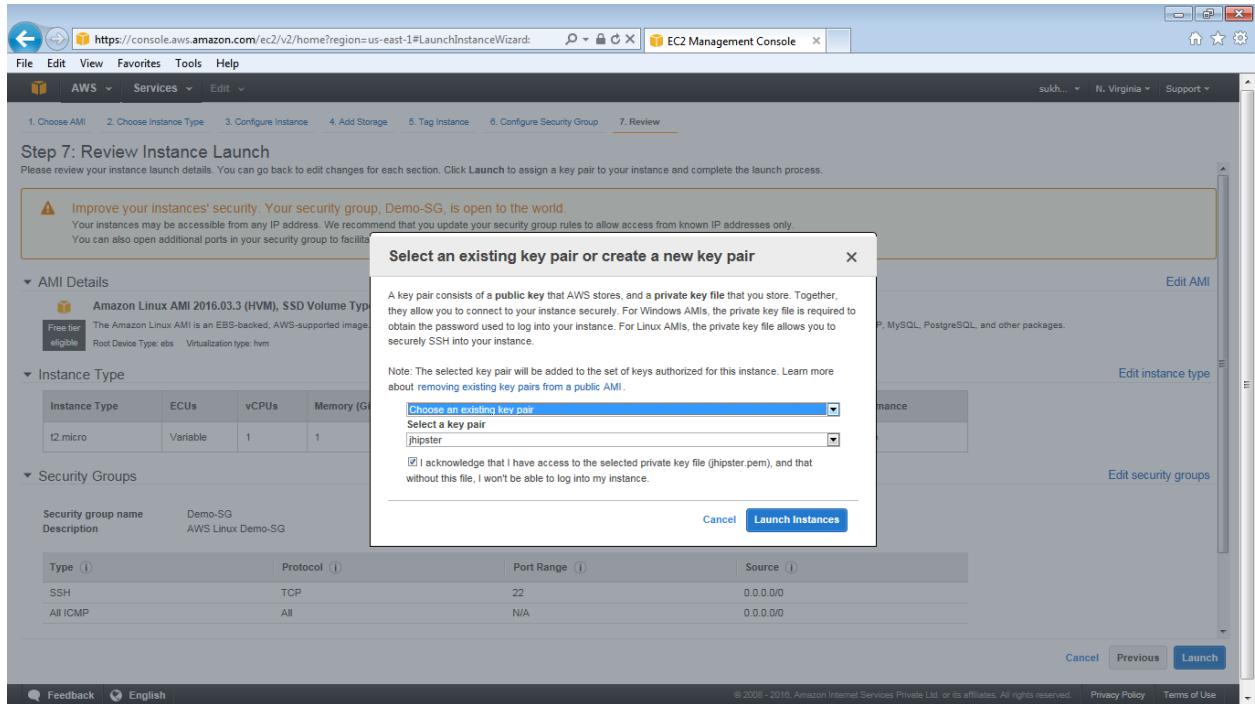
Security Groups

Security group name: Demo-SG
Description: AWS Linux Demo-SG

| Type | Protocol | Port Range | Source |
|----------|----------|------------|-----------|
| SSH | TCP | 22 | 0.0.0.0/0 |
| All ICMP | All | N/A | 0.0.0.0/0 |

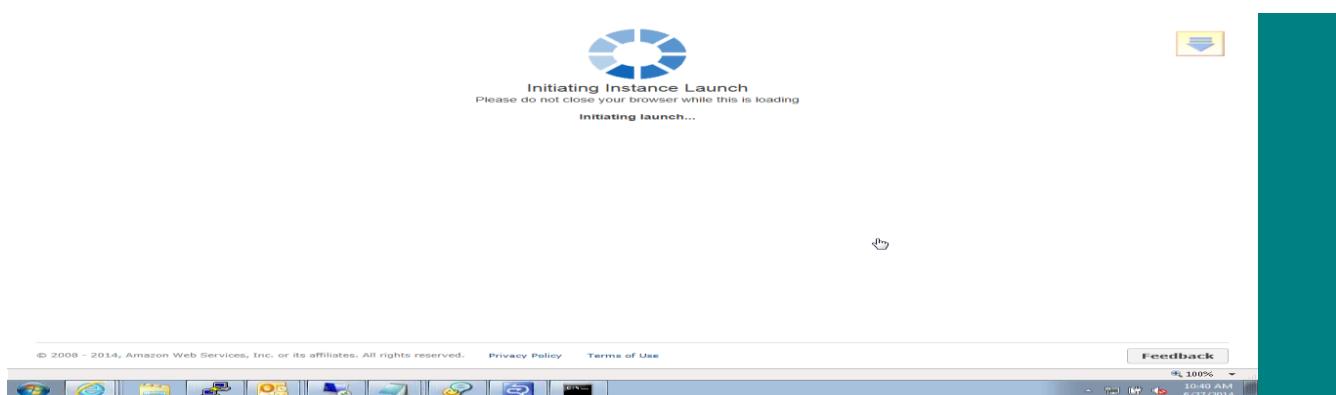
Cancel Previous [Launch](#)

- Select the option 'choose an existing key pair'.

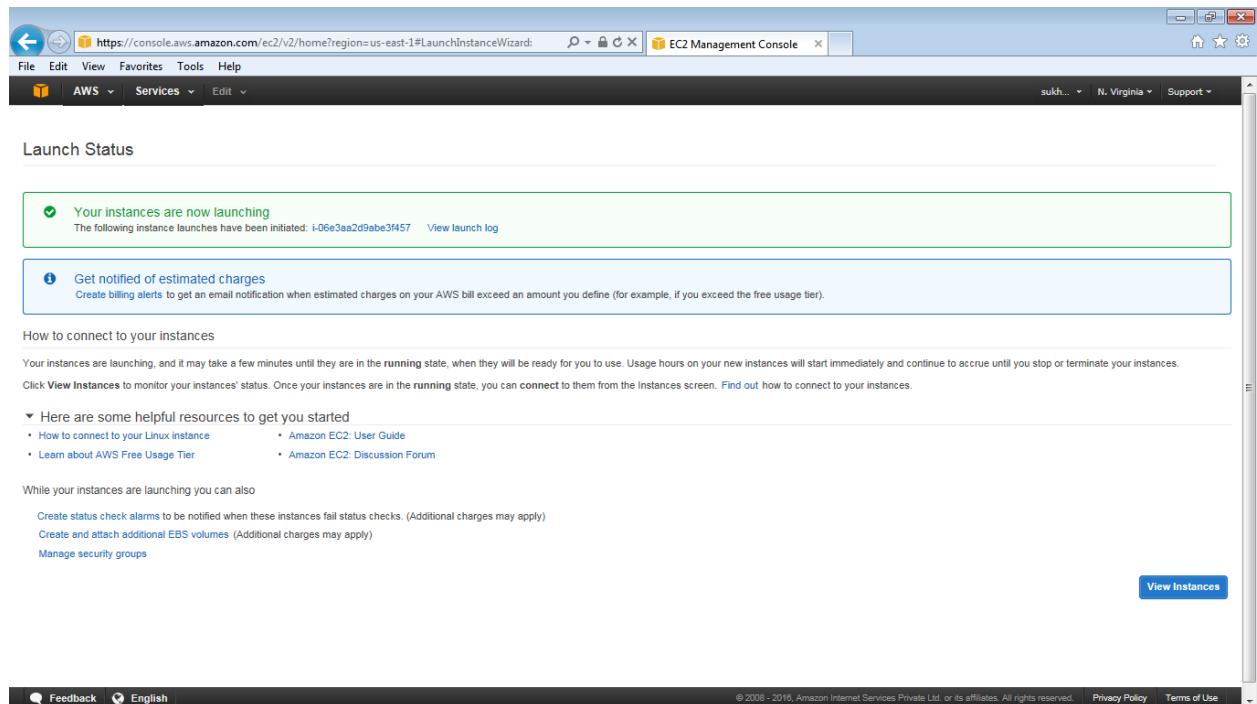


The screenshot shows the AWS EC2 Management Console interface. The main window displays the 'Step 7: Review Instance Launch' step. On the left, there are sections for 'AMI Details' (Amazon Linux AMI 2016.03.3 (HVM), SSD Volume Type), 'Instance Type' (t2.micro), and 'Security Groups' (Demo-SG). The 'Security Groups' section shows a table with one row: Type (SSH), Protocol (TCP), Port Range (22), and Source (0.0.0.0/0). The right side of the screen has tabs for 'Edit AMI', 'Edit instance type', and 'Edit security groups'. A status bar at the bottom says 'Initiating launch...'. A modal dialog box in the center is titled 'Select an existing key pair or create a new key pair'. It contains a dropdown menu set to 'Choose an existing key pair', a dropdown menu showing 'jhipster', and a checkbox labeled 'I acknowledge that I have access to the selected private key file (jhipster.pem), and that without this file, I won't be able to log into my instance.' Below the checkbox are 'Cancel' and 'Launch Instances' buttons.

- 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.



Your instances are now launching
The following instance launches have been initiated: i-06e3aa2d9abe3f457 [View launch log](#)

Get notified of estimated charges
[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances. Click [View Instances](#) to monitor your instances' status. Once your instances are in the running state, you can connect to them from the instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

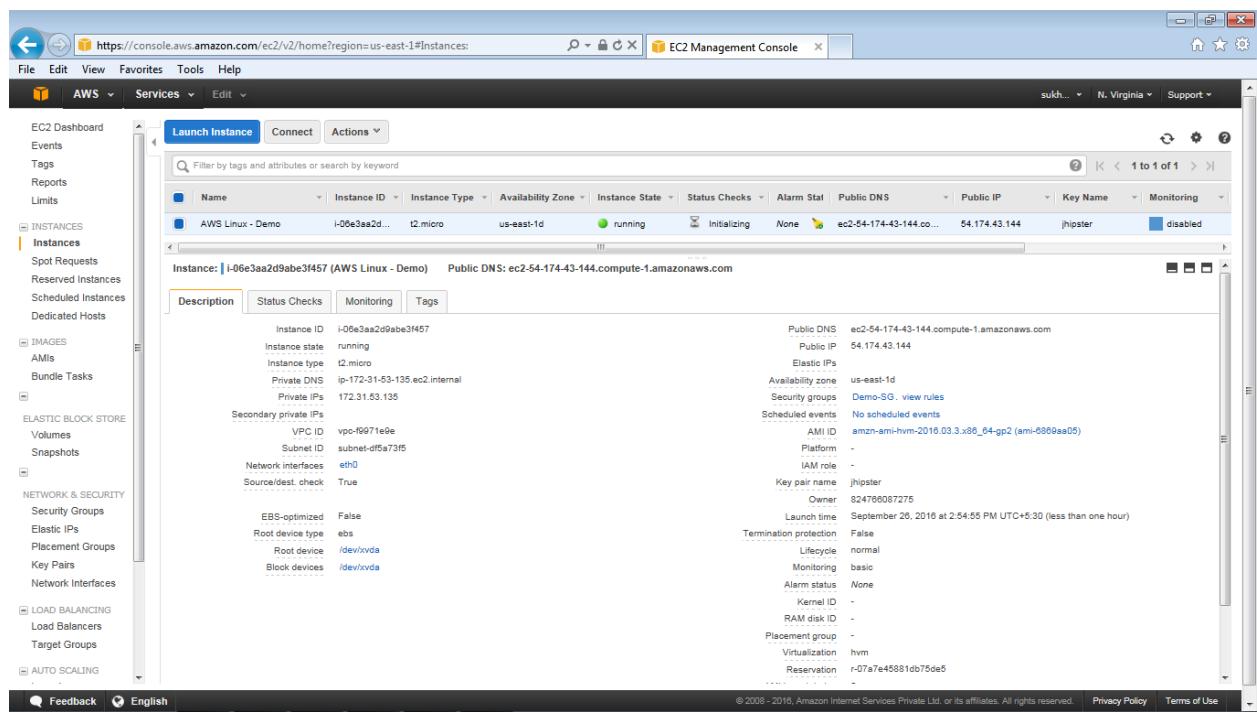
- How to connect to your Linux instance
- Amazon EC2 User Guide
- Learn about AWS Free Usage Tier
- Amazon EC2 Discussion Forum

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

[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.



| Name | Instance ID | Instance Type | Availability Zone | Status Checks | Alarm Stat | Public DNS | Public IP | Key Name | Monitoring |
|------------------|---------------------|---------------|-------------------|---------------|--------------|------------|---|----------|------------|
| AWS Linux - Demo | i-06e3aa2d9abe3f457 | t2.micro | us-east-1d | running | Initializing | None | ec2-54-174-43-144.compute-1.amazonaws.com | jhipster | disabled |

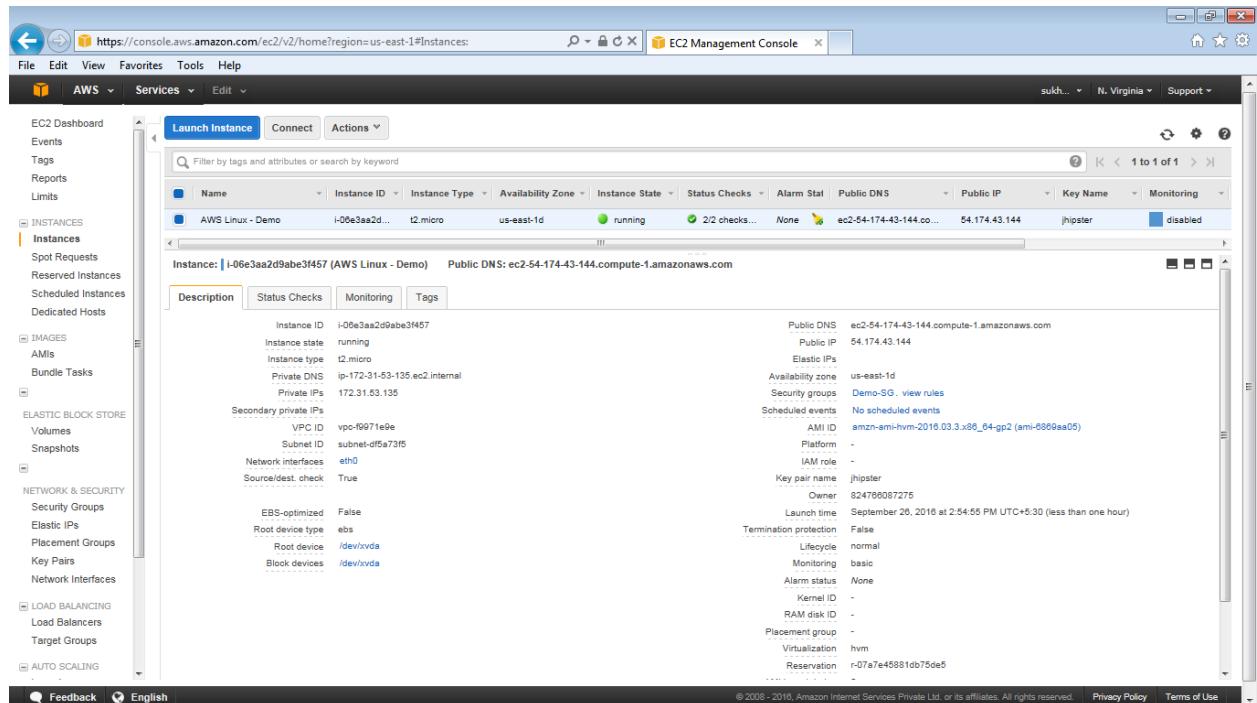
Instance: i-06e3aa2d9abe3f457 (AWS Linux - Demo) Public DNS: ec2-54-174-43-144.compute-1.amazonaws.com

Description **Status Checks** **Monitoring** **Tags**

Instance ID: i-06e3aa2d9abe3f457
 Instance state: running
 Instance type: t2.micro
 Private DNS: ip-172-31-53-135.ec2.internal
 Private IPs: 172.31.53.135
 Secondary private IPs:
 VPC ID: vpc-f9971e9e
 Subnet ID: subnet-df5a73f5
 Network interfaces: eth0
 Source/dest. check: True

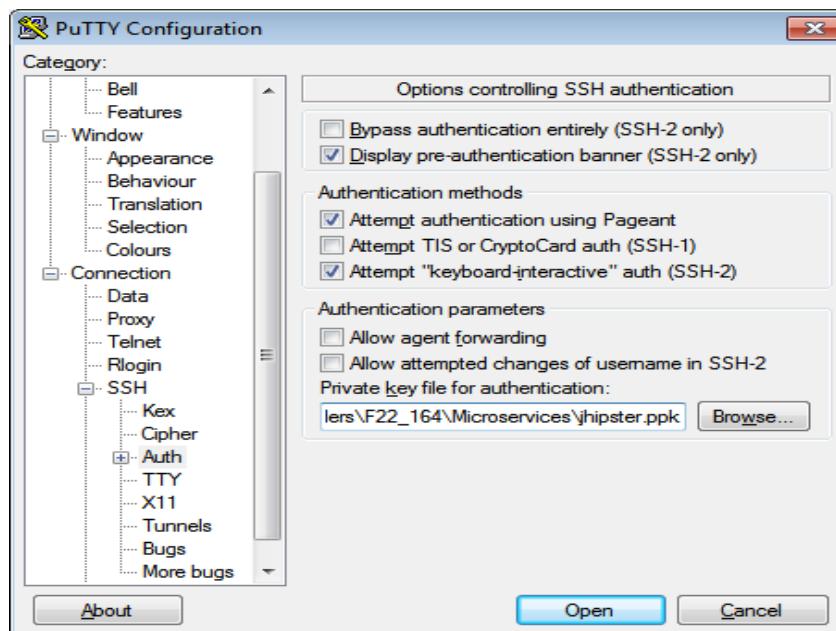
Public DNS: ec2-54-174-43-144.compute-1.amazonaws.com
 Public IP: 54.174.43.144
 Availability zone: us-east-1d
 Security groups: Demo-SG, [view rules](#)
 Scheduled events: No scheduled events
 AMI ID: amzn-ami-hvm-2016.03.3.x86_64-gp2 (ami-6869aa05)
 Platform: -
 IAM role: -
 Key pair name: jhipster
 Owner: 924786087276
 Launch time: September 26, 2016 at 2:54:55 PM UTC+5:30 (less than one hour)
 Termination protection: False
 Lifecycle: normal
 Monitoring: basic
 Alarm status: None
 Kernel ID: -
 RAM disk ID: -
 Placement group: -
 Virtualization: hvm
 Reservation: r-07a7e45881db75de5

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



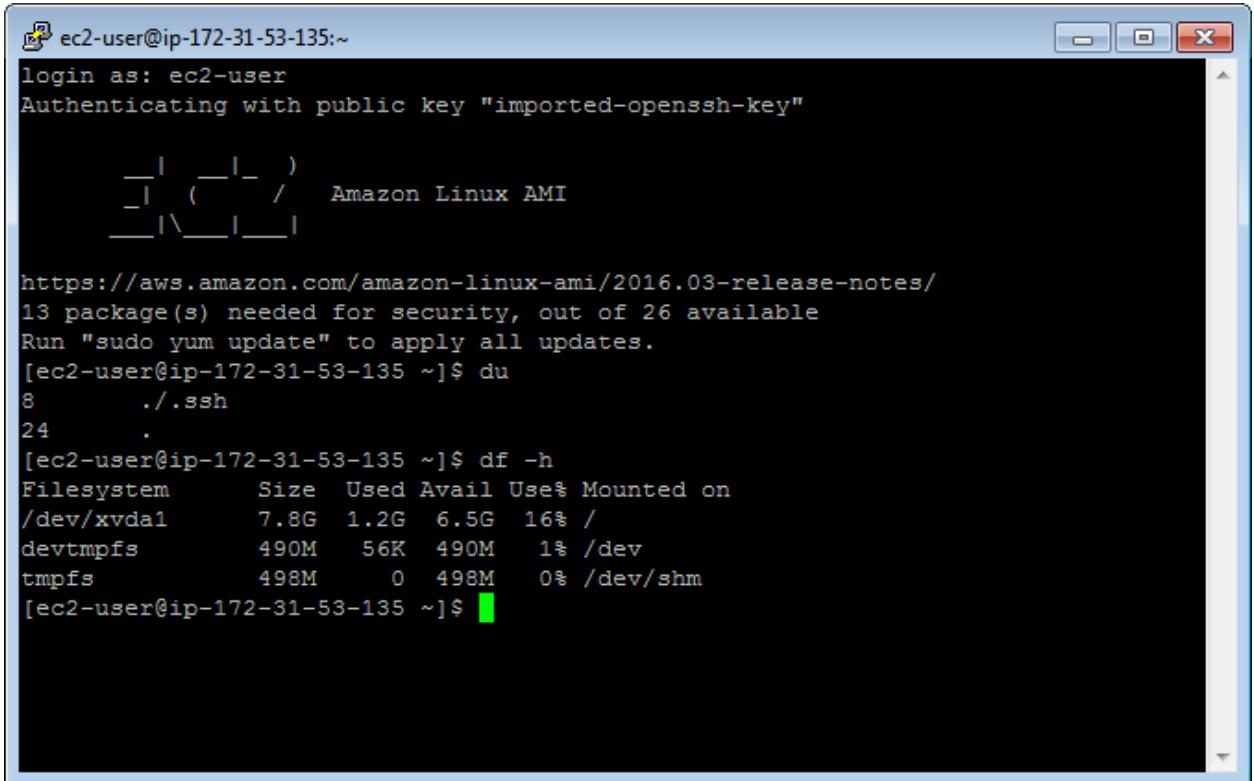
The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with links like EC2 Dashboard, Events, Tags, Limits, Instances, Images, AMIs, Bundle Tasks, ELASTIC BLOCK STORE, Volumes, Snapshots, NETWORK & SECURITY, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, LOAD BALANCING, Load Balancers, Target Groups, and AUTO SCALING. The main panel displays a table of instances. One instance is selected: "AWS Linux - Demo" (Instance ID: i-06e3aa2d9abe3f457). The table includes columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Stat, Public DNS, Public IP, Key Name, and Monitoring. The instance state is "running" with a status of "2/2 checks...". The Public DNS is "ec2-54-174-43-144.compute-1.amazonaws.com" and the Public IP is "54.174.43.144". The Key Name is "jhipster" and the Monitoring is "disabled". Below the table, there are tabs for Description, Status Checks, Monitoring, and Tags. The "Description" tab shows detailed instance information such as Instance ID (i-06e3aa2d9abe3f457), Instance state (running), Instance type (t2.micro), Private DNS (ip-172-31-53-135.ec2.internal), Private IPs (172.31.53.135), Secondary private IPs (VPC ID: vpc-f9971e9e, Subnet ID: subnet-d5fa73f5, Network interfaces: eth0, Source/dest. check: True), EBS-optimized (False), Root device type (ebs), Root device (/dev/xvda), and Block devices (/dev/xvda). The "Status Checks" tab shows a green status with "2/2 checks...". The "Monitoring" tab shows "disabled". The "Tags" tab is empty. At the bottom of the main panel, there are sections for Public DNS, Public IP, Availability zone, Security groups, Scheduled events, IAM ID, Platform, IAM role, Key pair name, Owner, Launch time, Termination protection, Lifecycle, Monitoring, Alarm status, Kernel ID, RAM disk ID, Placement group, Virtualization, and Reservation.

- 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



```

ec2-user@ip-172-31-53-135:~ 
login as: ec2-user
Authenticating with public key "imported-openssh-key"

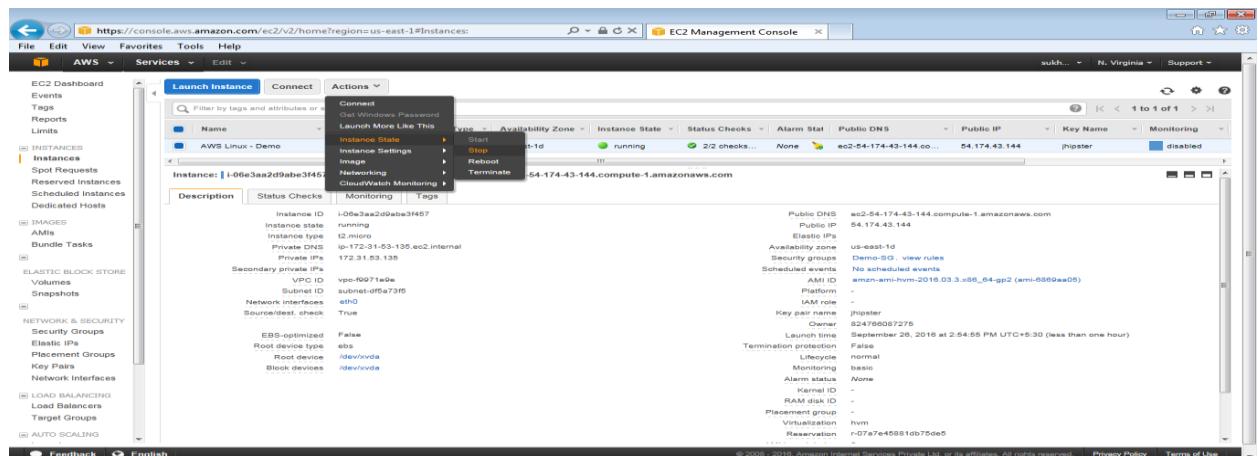
 _|_ |_) 
_| ( /   Amazon Linux AMI
 \_|_|_|

https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
13 package(s) needed for security, out of 26 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-53-135 ~]$ du
8      ./.ssh
24     .
[ec2-user@ip-172-31-53-135 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       7.8G  1.2G  6.5G  16% /
devtmpfs        490M   56K  490M   1% /dev
tmpfs          498M     0  498M   0% /dev/shm
[ec2-user@ip-172-31-53-135 ~]$ 

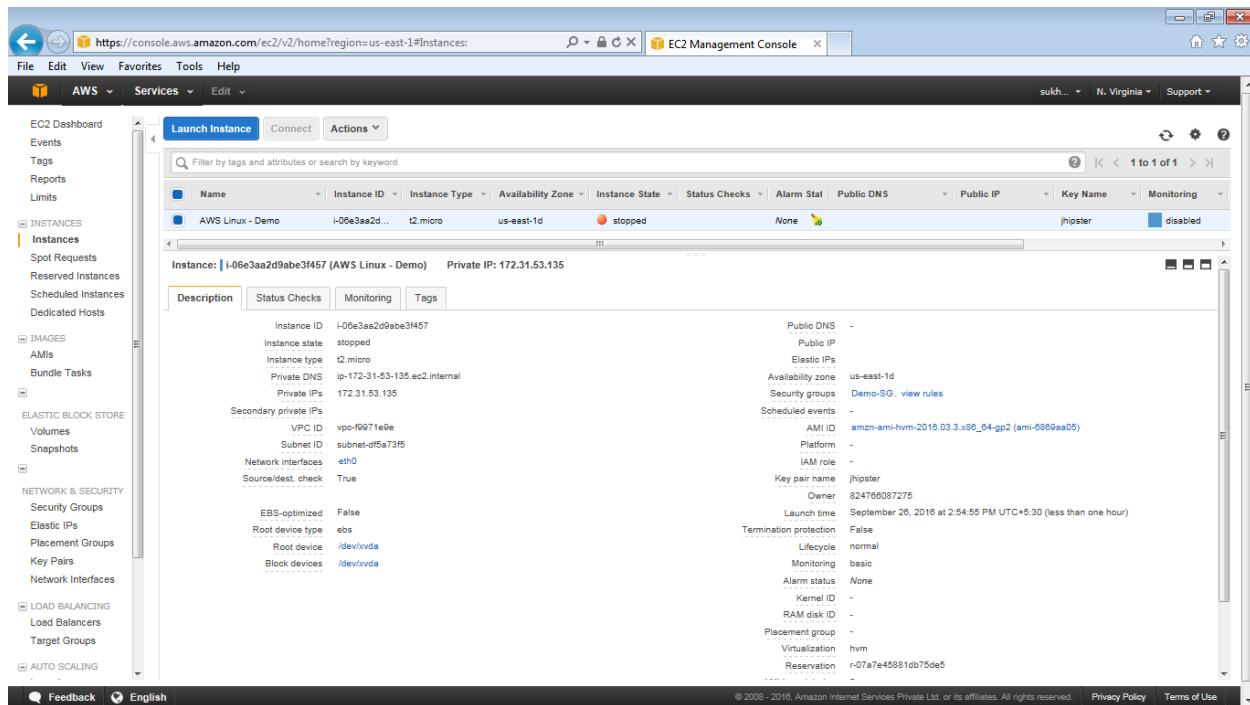
```

1.3.2 Resizing the EC2 instance

1. Go to the EC2 Dashboard and select the instances
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

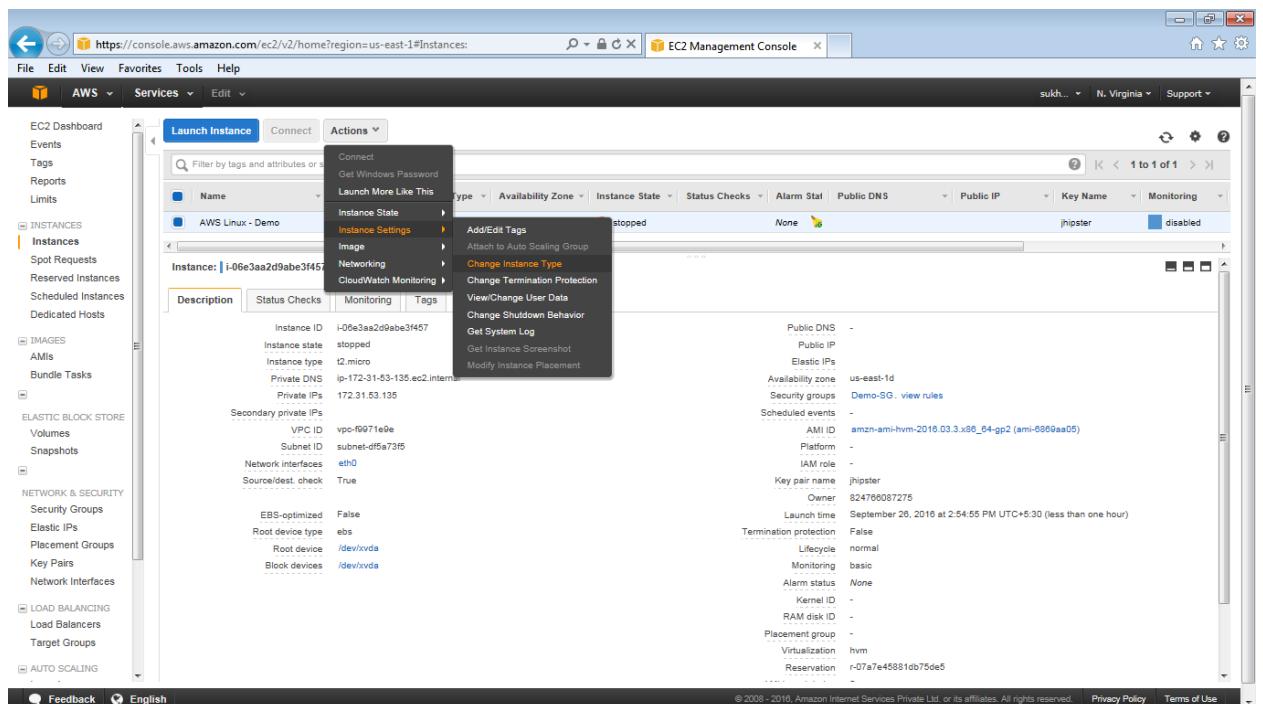


4. Instance status code is shown as “stopped”

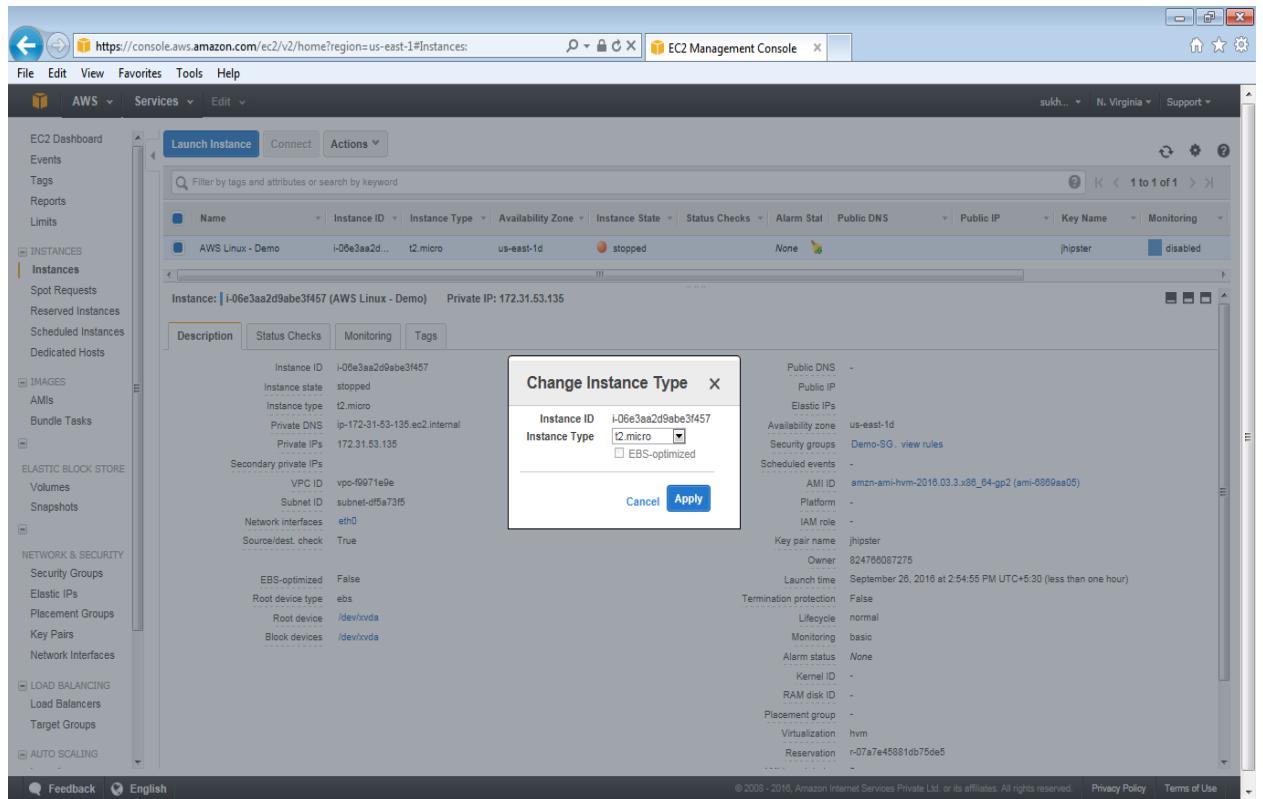


The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, and more. The main area displays a table of instances. One instance is selected: "AWS Linux - Demo" (i-06e3aa2d9abe3f457). The instance details show it is currently "stopped". Other details include its instance ID, type (t2.micro), availability zone (us-east-1d), private IP (172.31.53.135), and various network and storage configurations. The right side of the screen shows detailed configuration settings for the selected instance.

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

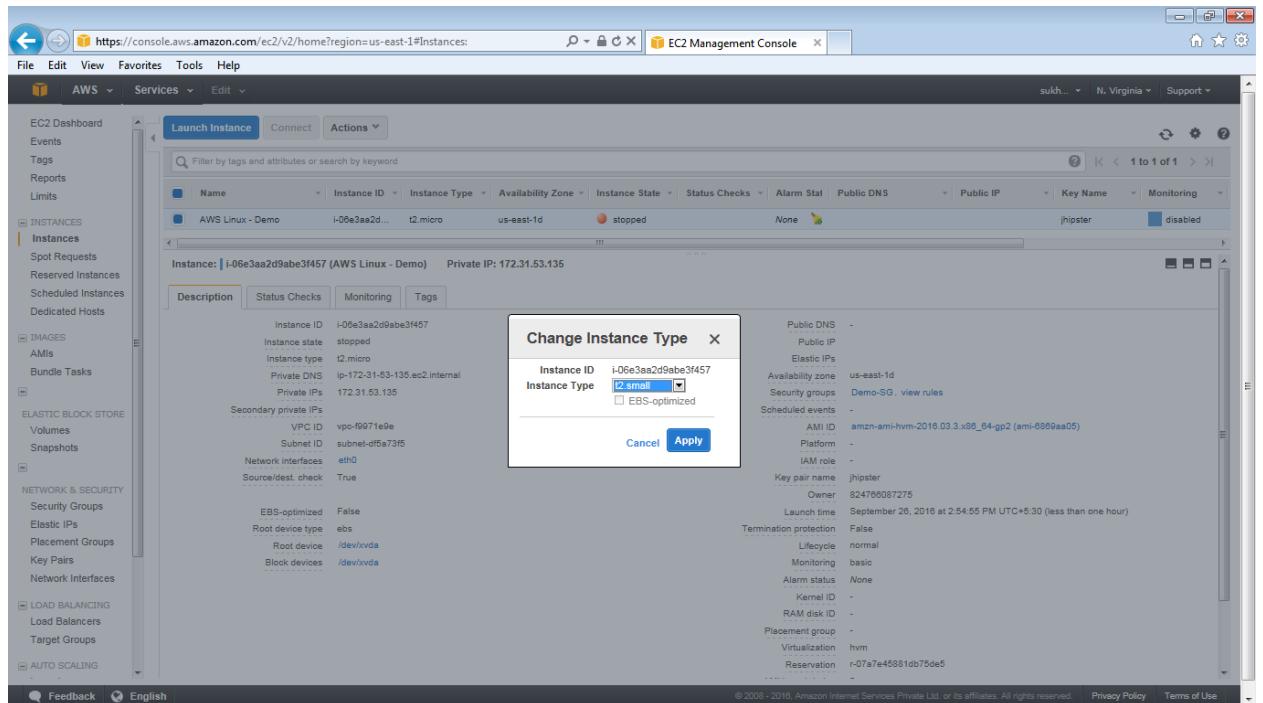


This screenshot shows the same EC2 Management Console interface as above, but with a different focus. The "Actions" dropdown menu is open over the instance table. The "Instance Settings" option is selected, and its submenu is visible, showing options like "Add/Edit Tags", "Attach to Auto Scaling Group", "Change Instance Type", "Change Termination Protection", "View/Change User Data", "Change Shutdown Behavior", "Get System Log", "Get Instance Screenshot", and "Modify Instance Placement". The "Change Instance Type" option is highlighted with a red box. The rest of the interface remains consistent with the previous screenshot, showing the same instance details and configuration options.



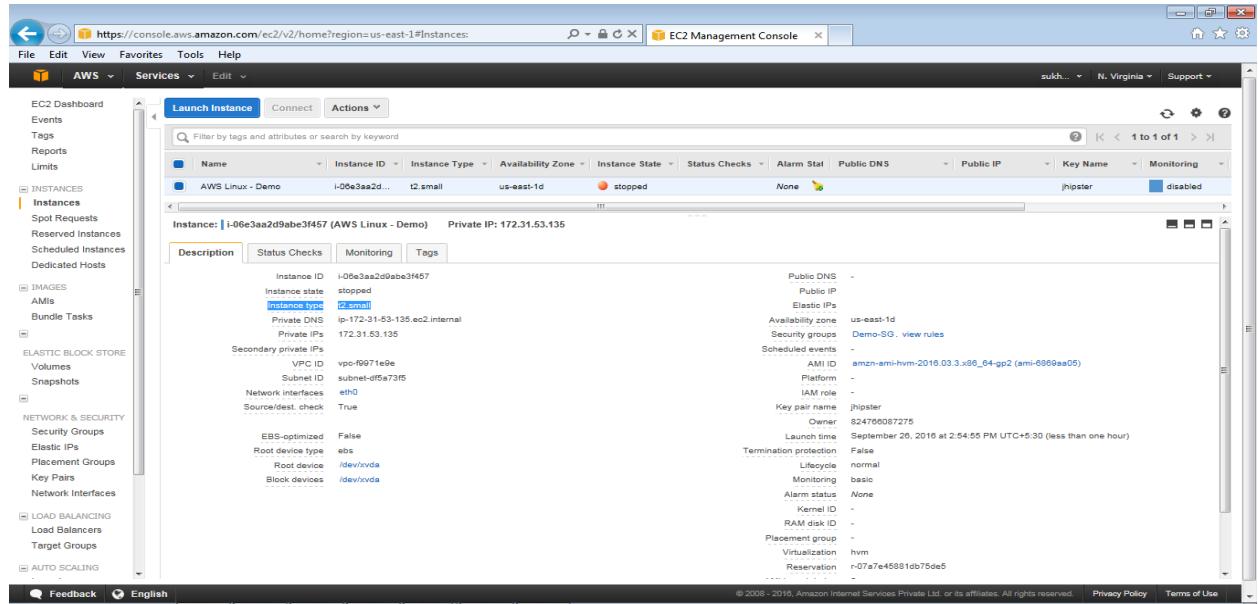
The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, and Auto Scaling. The main area displays a table of instances. One instance, 'AWS Linux - Demo', is selected. A modal dialog box titled 'Change Instance Type' is open over the instance details. In the dialog, the 'Instance Type' dropdown is set to 'd2.small'. Other visible fields include 'Public DNS', 'Availability zone (us-east-1d)', 'Security groups (Demo-SG, view rules)', and various launch configuration details.

6. Click on apply button



This screenshot is identical to the one above it, showing the 'Change Instance Type' dialog box. The difference is that the 'Apply' button has been clicked, and the dialog now displays a confirmation message: 'Your changes have been saved. Your instance will restart in approximately 2 minutes.' The rest of the interface remains the same, showing the EC2 dashboard and the selected instance details.

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

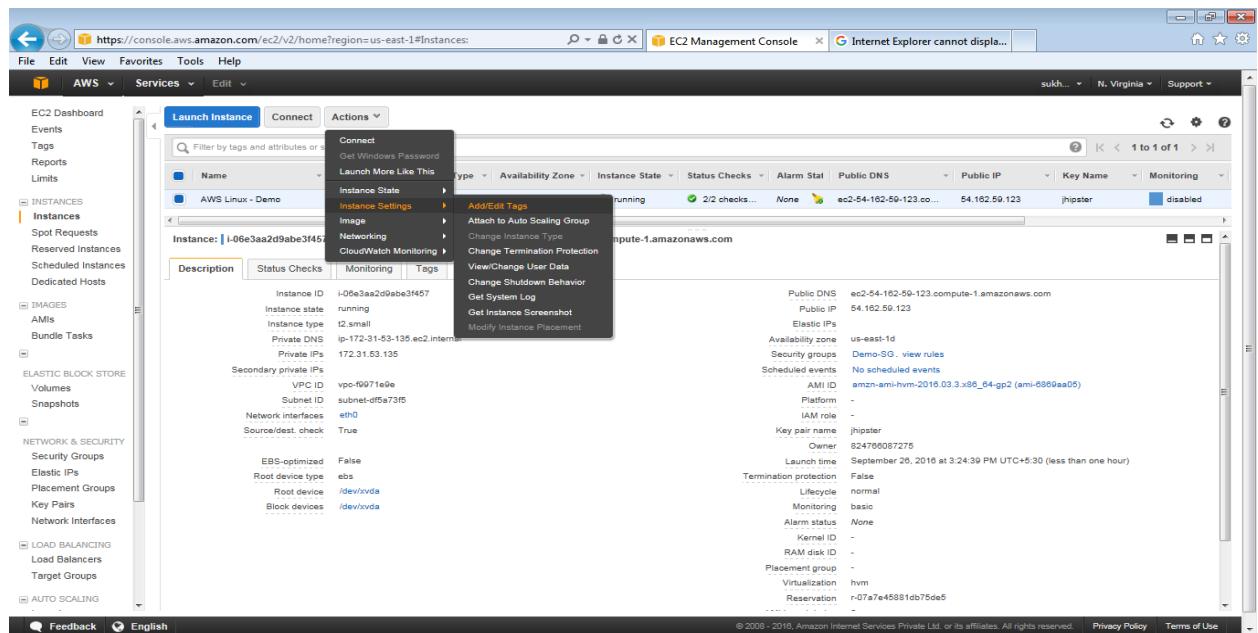


The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, and Auto Scaling. The main area is titled 'Launch Instance' and shows a table with one row for 'AWS Linux - Demo'. The table columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Stat, Public DNS, Public IP, Key Name, and Monitoring. The instance details pane below shows the instance ID (i-06e3aa2d9abe3f457), instance state (stopped), instance type (t2.micro), private DNS (ip-172-31-53-135.ec2.internal), private IP (172.31.53.135), secondary private IPs, VPC ID (vpo-f0071e9e), subnet ID (subnet-d5a73f5), network interfaces (eth0), source/dest. check (True), EBS-optimized (False), root device type (ebs), root device (/dev/xvda), and block devices (/dev/xvda). The right side of the screen displays detailed instance metadata, including Public DNS, Public IP, Elastic IPs, Availability zone (us-east-1d), Security groups (Demo-SG, view rules), Scheduled events, AMI ID (amzn-ami-hvm-2016.03.3.x86_64-gp2 (ami-6069aa05)), Platform, IAM role, Key pair name (jhipster), Owner (824760087275), Launch time (September 20, 2016 at 2:54:55 PM UTC+5:30 (less than one hour)), Termination protection (False), Lifecycle (normal), Monitoring (basic), Alarm status (None), Kernel ID, RAM disk ID, Placement group, Virtualization (hvm), and Reservation (r-07a7e45881db75de5).

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

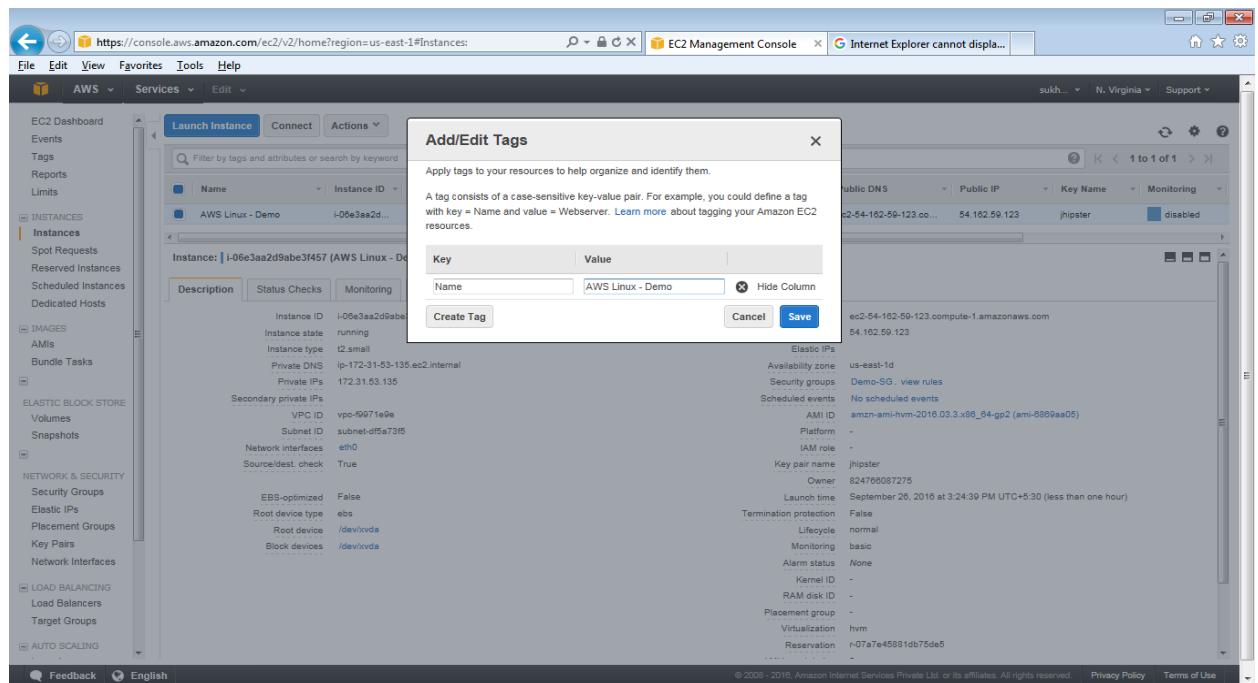
1.3.3 Modify EC2 instance

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



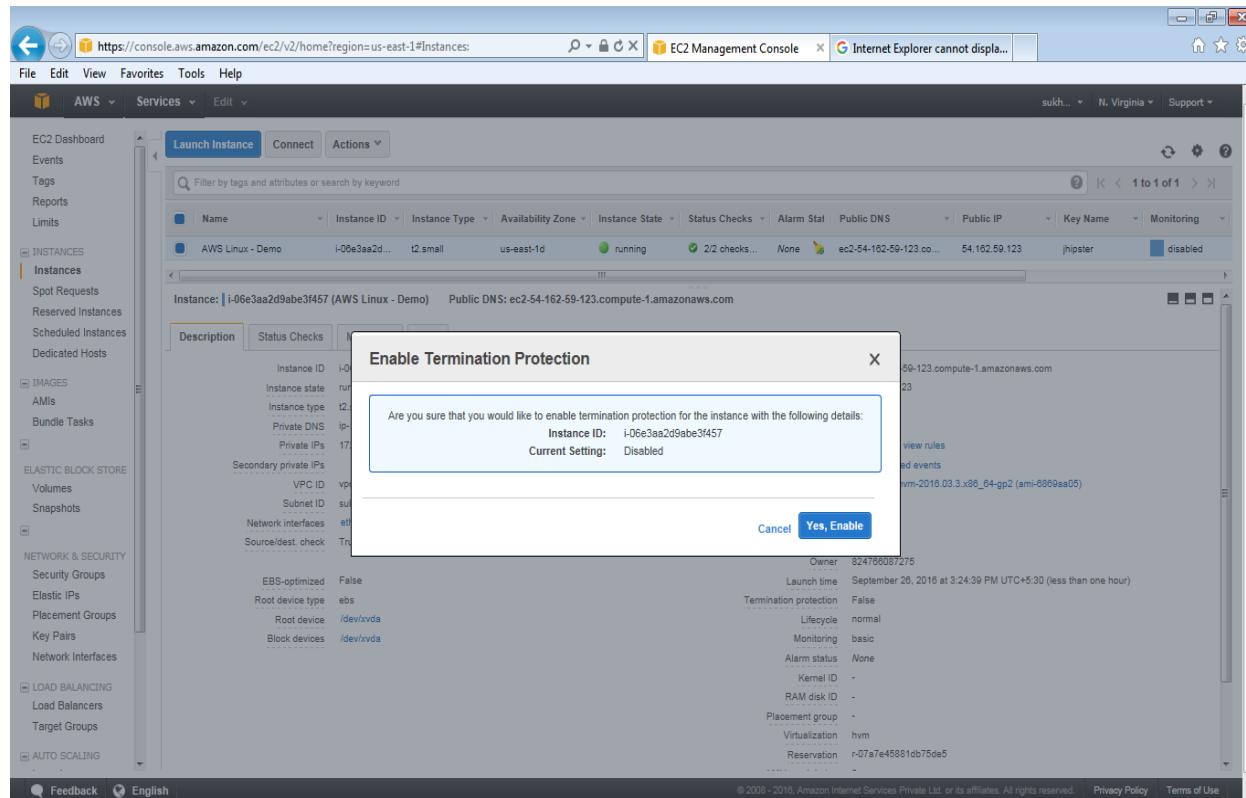
The screenshot shows the AWS EC2 Management Console interface, similar to the previous one. The left sidebar has the same navigation links. The main area shows the 'Launch Instance' configuration for 'AWS Linux - Demo'. The 'Actions' dropdown menu is open, revealing options like Connect, Get Windows Password, Launch More Like This, Instance State (which is currently running), Instance Settings, Add/Edit Tags, Attach to Auto Scaling Group, Change Instance Type, Change Termination Protection, View/Change User Data, Change Shutdown Behavior, Get System Log, Get Instance Screenshot, and Modify Instance Placement. The instance details pane and detailed metadata on the right are identical to the first screenshot.

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



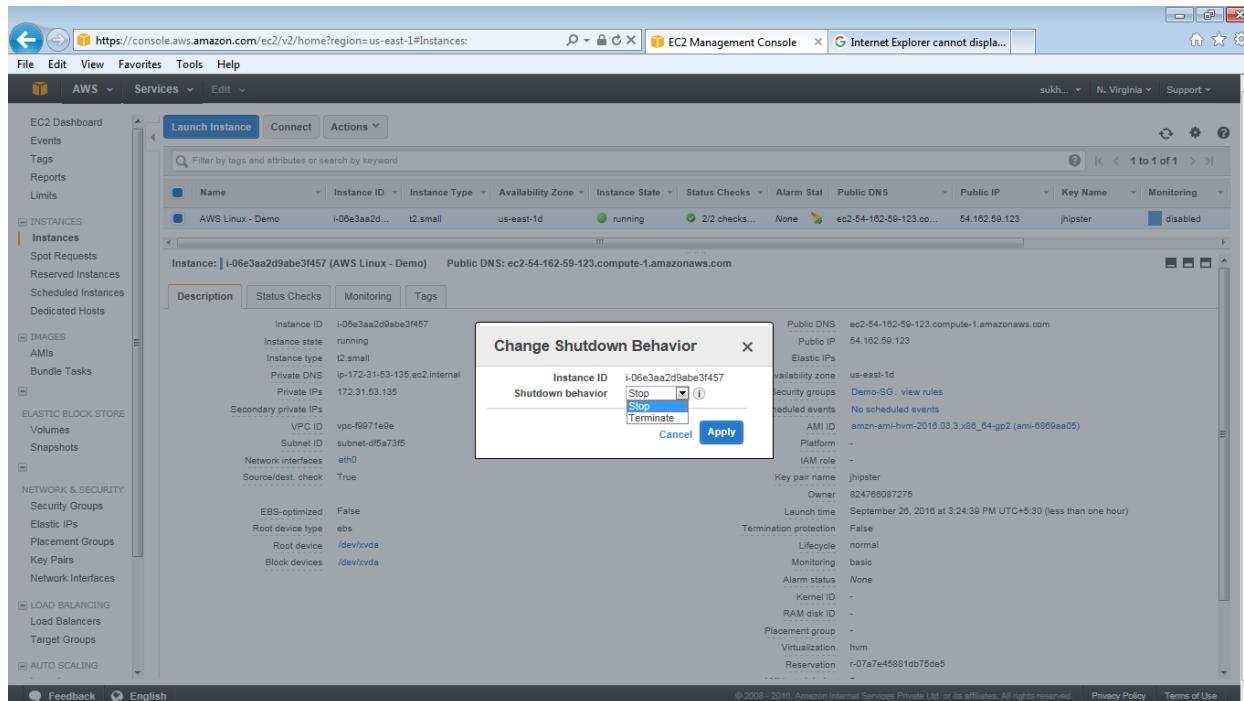
The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, etc. The main area displays a list of instances, with one instance selected: 'AWS Linux - Demo' (Instance ID: i-06e3aa2d9abe3f457). Below the instance details, there's a 'Description' tab showing various metadata fields such as Instance ID, Instance state, Instance type, Private DNS, Private IPs, Secondary private IPs, Network interfaces, and EBS-optimized status. A 'Tags' tab is also visible. A modal window titled 'Add/Edit Tags' is open over the instance details, prompting the user to apply tags to organize and identify resources. The 'Name' tag is already present with the value 'AWS Linux - Demo'. There are 'Create Tag' and 'Save' buttons at the bottom of the modal.

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



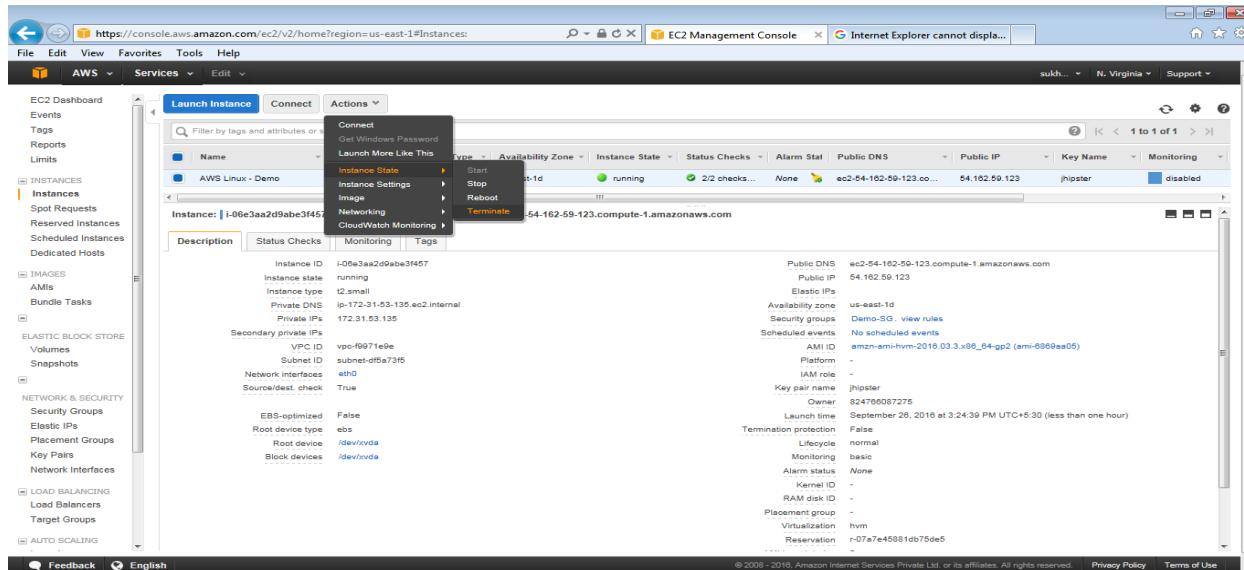
This screenshot shows the same AWS EC2 Management Console interface as the previous one. The instance 'AWS Linux - Demo' is still selected. A modal window titled 'Enable Termination Protection' is displayed, asking the user if they are sure about enabling termination protection for the instance. The window shows the instance ID 'i-06e3aa2d9abe3f457' and the current setting 'Disabled'. At the bottom of the modal are 'Cancel' and 'Yes, Enable' buttons. The background shows the detailed instance information and the EC2 dashboard sidebar.

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 instances
2. From Action select Terminate option , it will show warning message



3. Press “Yes Terminate” button

Screenshot of the AWS EC2 Management Console showing the process of terminating an instance.

The instance details for "AWS Linux - Demo" (i-06e3aa2d9abe3f457) are displayed, including its state as "running". A "Terminate Instances" dialog box is open, containing a warning message about the default action for EBS-backed 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.

The dialog also asks, "Are you sure you want to terminate these instances?" with "Yes, Terminate" as the primary button.

Screenshot of the AWS EC2 Management Console showing the instance status after termination.

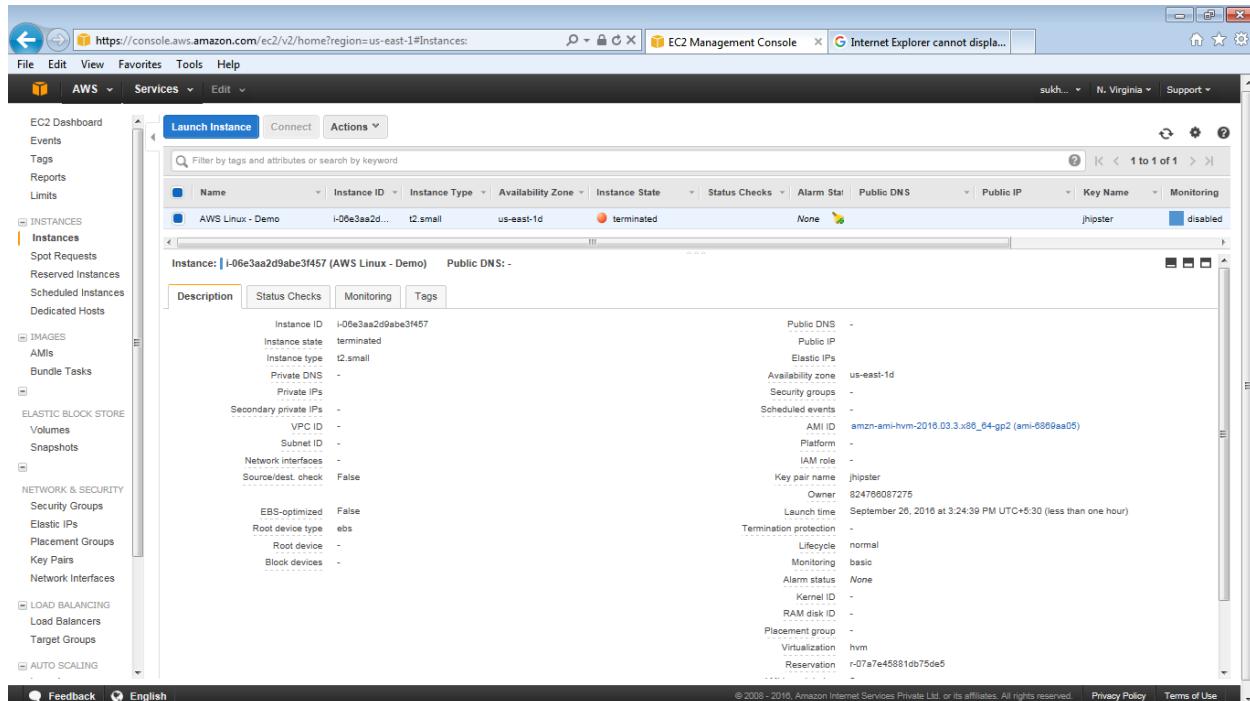
The instance "AWS Linux - Demo" (i-06e3aa2d9abe3f457) is now in a "shutting-down" state. The detailed instance information shows the following details:

- Description:** AWS Linux - Demo
- Status Checks:** None
- Monitoring:** disabled
- Instance ID:** i-06e3aa2d9abe3f457
- Instance state:** shutting-down
- Instance type:** t2.small
- Availability zone:** us-east-1d
- Public DNS:** -
- Public IP:** -
- Elastic IPs:** -
- Private DNS:** -
- Private IPs:** -
- Secondary private IPs:** -
- VPC ID:** -
- Subnet ID:** -
- Network interfaces:** eth0
- Source/dest. check:** False
- EBS-optimized:** False
- Root device type:** ebs
- Root device:** /dev/xvda
- Block devices:** /dev/xvda

The instance's termination protection is set to "False". Other metadata fields include:

- AMI ID: amzn-ami-hvm-2016.03.3.x86_64-gp2 (ami-8869aa05)
- Platform: -
- IAM role: -
- Key pair name: jhipster
- Owner: 824766097275
- Launch time: September 26, 2016 at 3:24:39 PM UTC+5:30 (less than one hour)
- Termination protection: False
- Lifecycle: normal
- Monitoring: basic
- Alarm status: None
- Kernel ID: -
- RAM disk ID: -
- Placement group: -
- Virtualization: hvm
- Reservation: r-07a7e45881db75de5

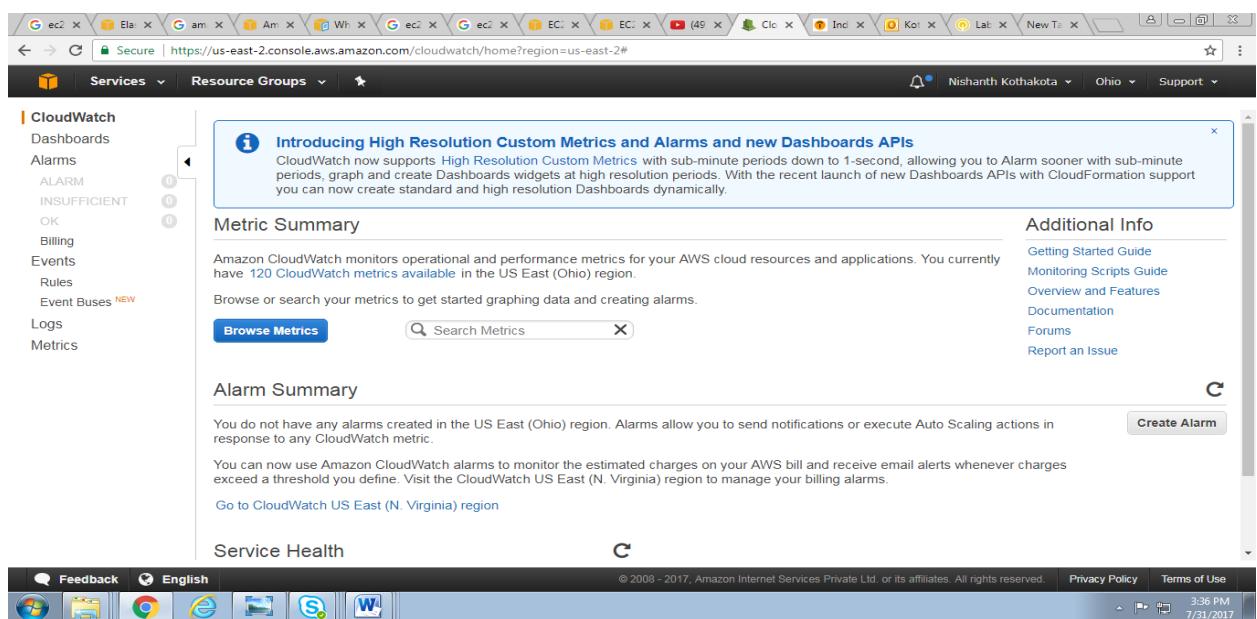
4. The Instance status code is changed to Terminated



The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area has tabs for Launch Instance, Connect, and Actions. A search bar at the top says "Filter by tags and attributes or search by keyword". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Star, Public DNS, Public IP, Key Name, and Monitoring. One row is selected: "AWS Linux - Demo" with Instance ID i-06e3aa2d9abe3f457, Instance Type t2.small, Availability Zone us-east-1d, and Instance State terminated. The "Monitoring" column shows "disabled". Below the table, there's a detailed view of the instance: Description, Status Checks, Monitoring, and Tags. The "Status Checks" tab is active, showing various metrics like Instance ID, Instance state, Instance type, and so on. The "Monitoring" tab shows that monitoring is disabled.

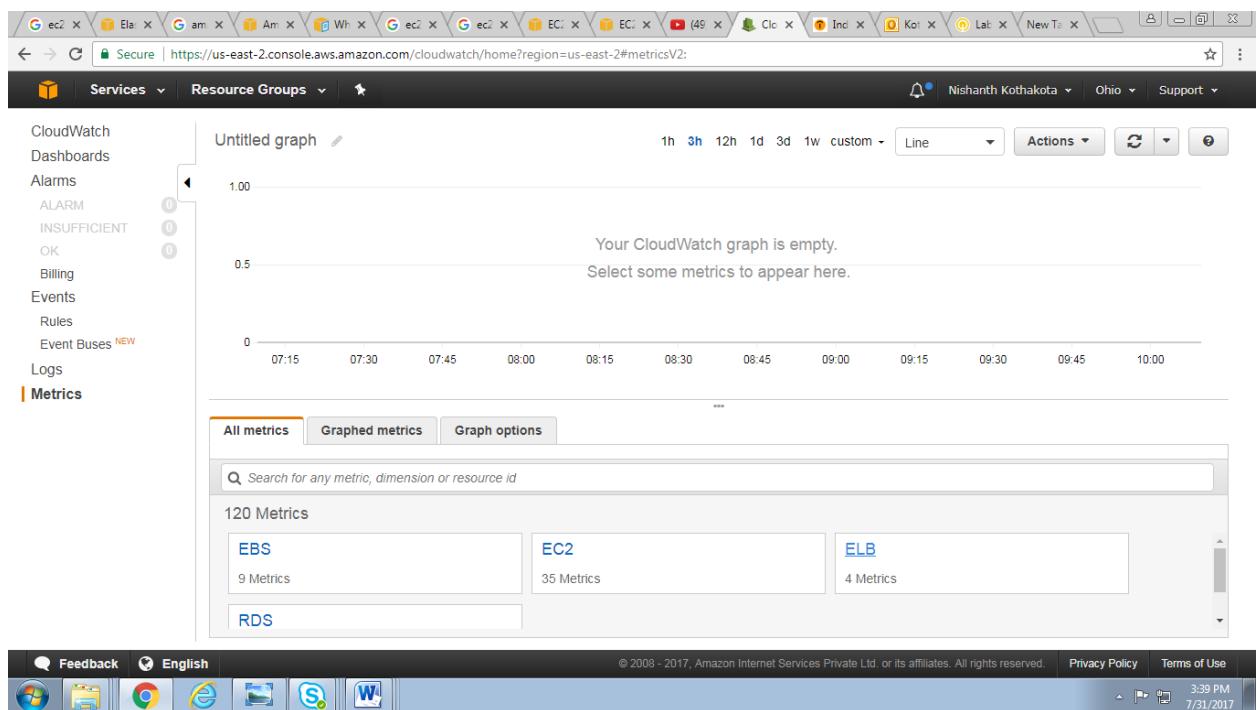
1.2.5 MONITORING EC2 INSTANCE

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

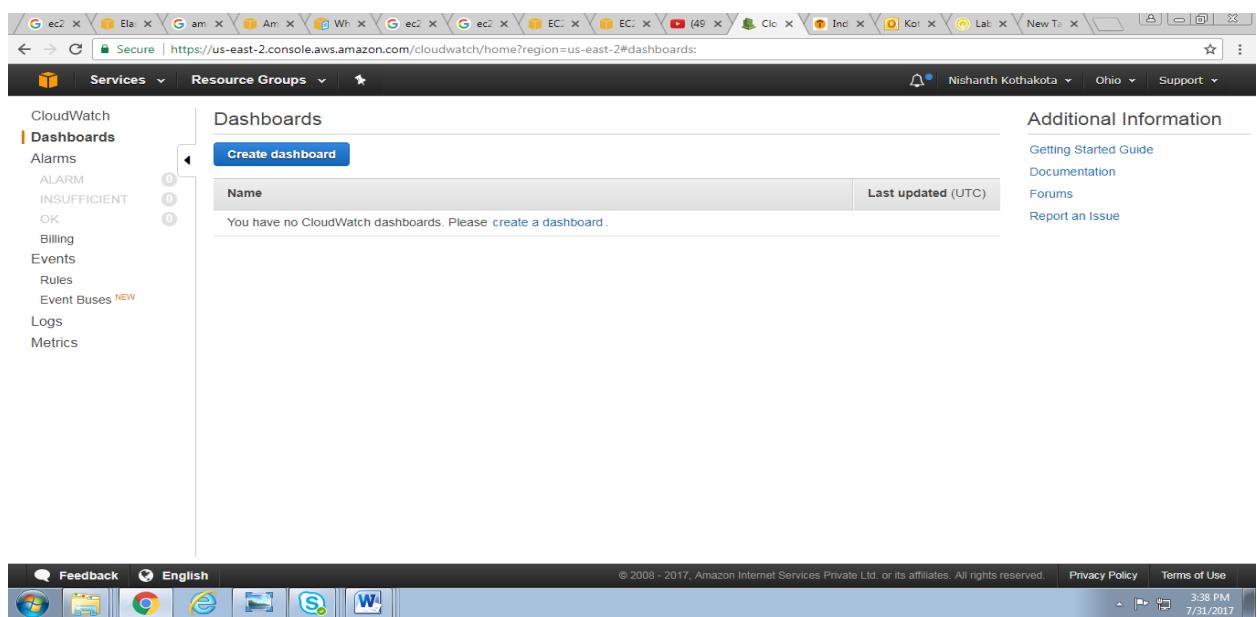


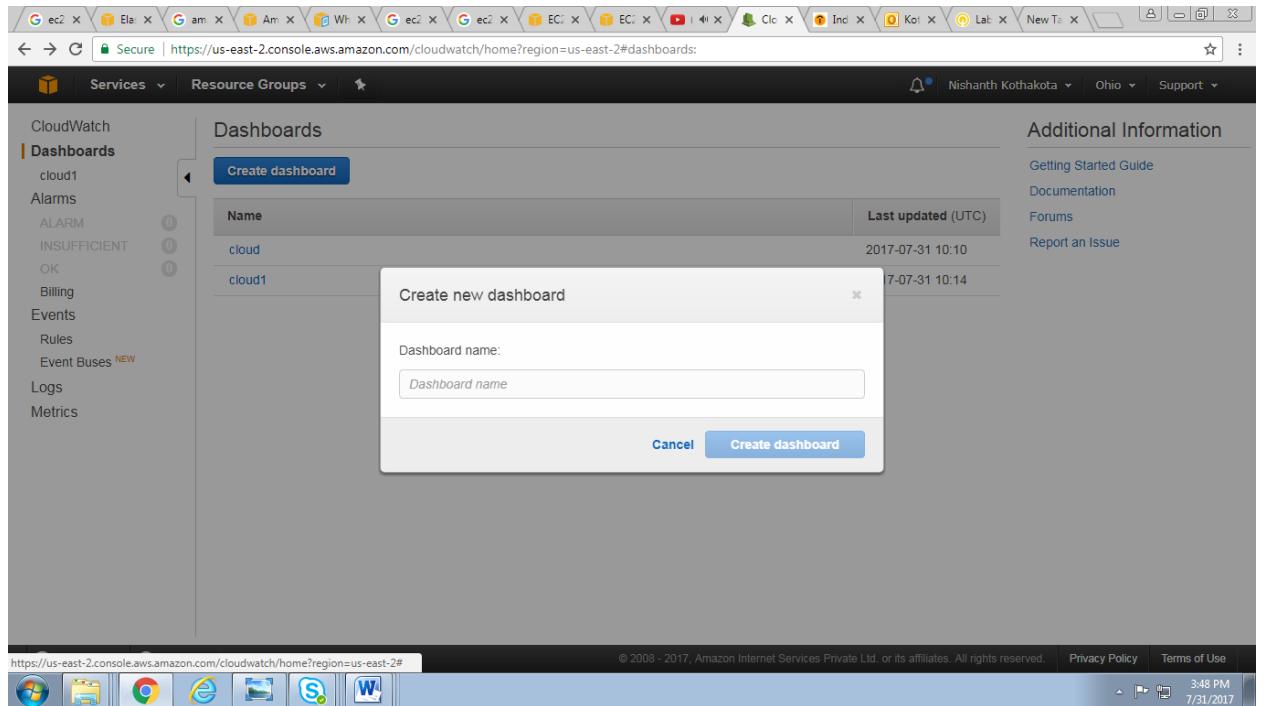
The screenshot shows the AWS CloudWatch Metrics and Alarms page. The left sidebar lists CloudWatch services: Dashboards, Alarms, ALARM, INSUFFICIENT, OK, Billing, Events, Rules, Event Buses (NEW), Logs, and Metrics. The main content area has a banner about High Resolution Custom Metrics and Alarms. Below it is the "Metric Summary" section, which says "Amazon CloudWatch monitors operational and performance metrics for your AWS cloud resources and applications. You currently have 120 CloudWatch metrics available in the US East (Ohio) region." It includes a "Browse Metrics" button and a search bar. To the right is the "Additional Info" section with links to Getting Started Guide, Monitoring Scripts Guide, Overview and Features, Documentation, Forums, and Report an Issue. Below that is the "Alarm Summary" section, which states "You do not have any alarms created in the US East (Ohio) region. Alarms allow you to send notifications or execute Auto Scaling actions in response to any CloudWatch metric." It includes a "Create Alarm" button. At the bottom is the "Service Health" section, which is currently healthy. The footer contains standard AWS links: Feedback, English, Privacy Policy, Terms of Use, and a timestamp (3:36 PM 7/31/2017).

2. Click on Browse Metrics



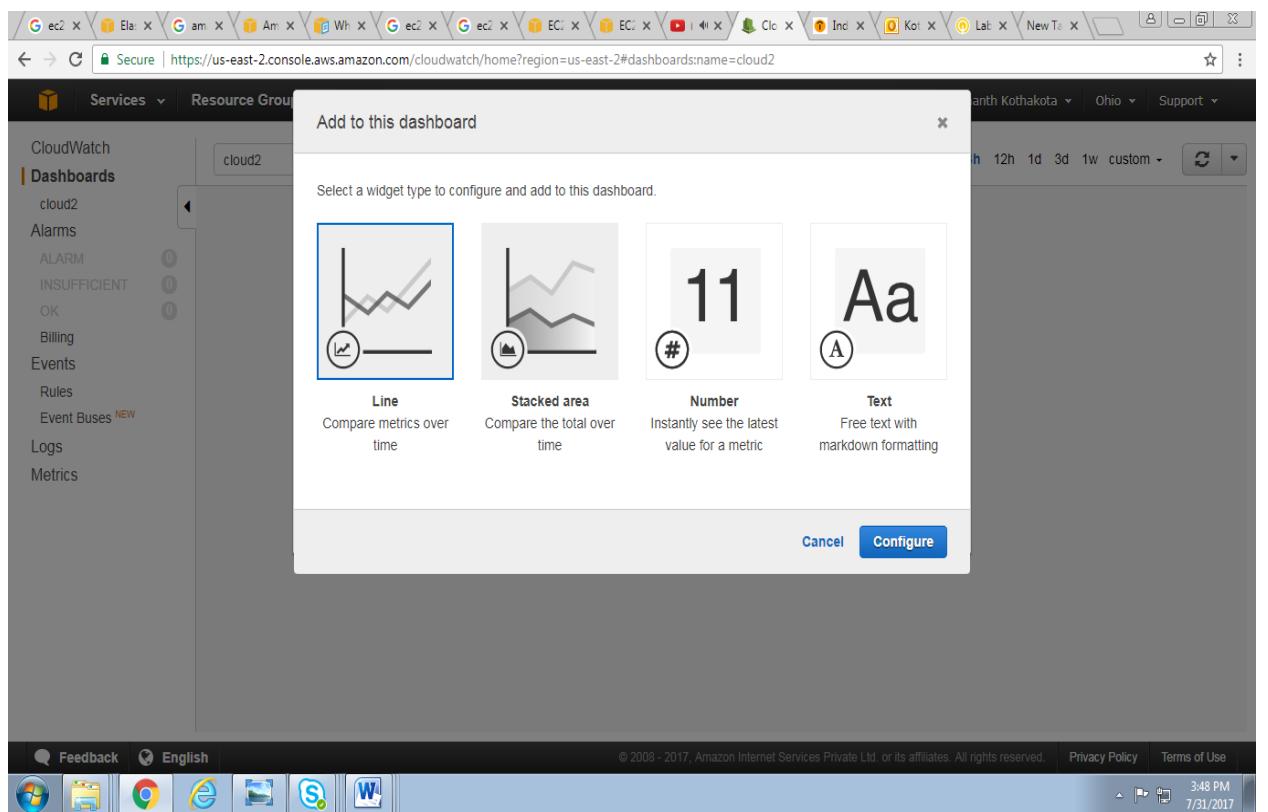
3. Click on Dash Board and create new Dashboard





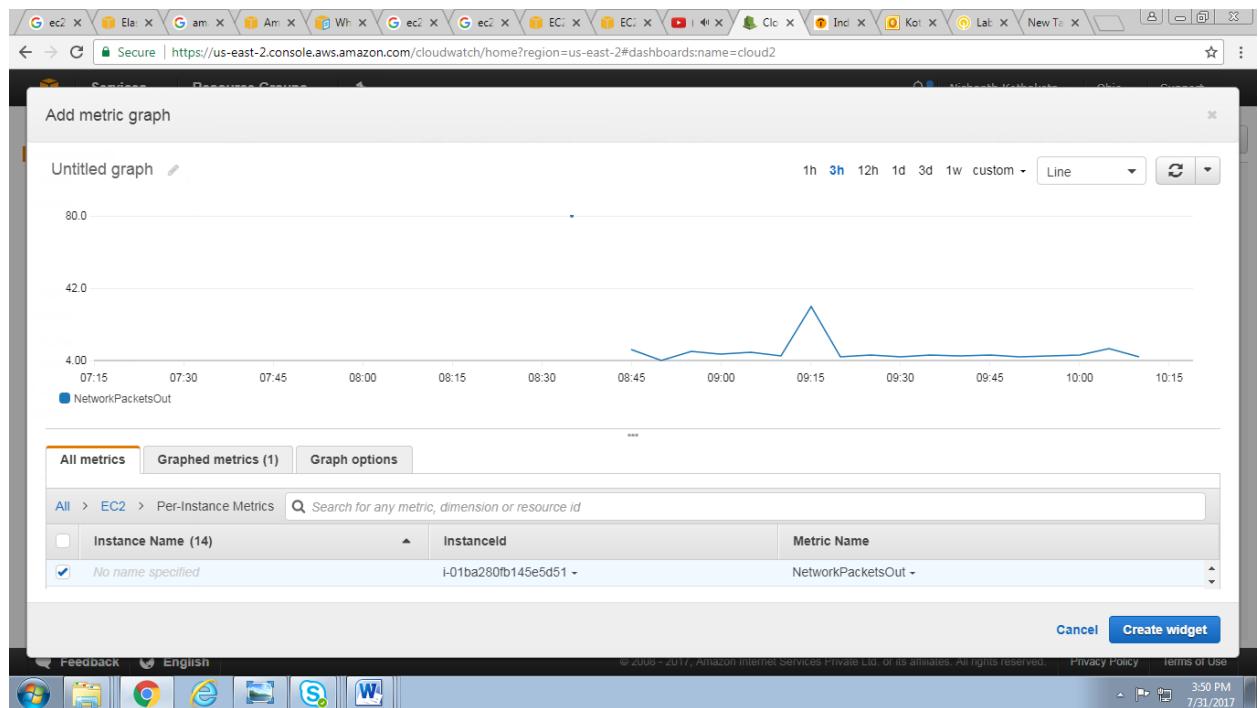
The screenshot shows the AWS CloudWatch Metrics Dashboard creation interface. On the left, there's a sidebar with navigation links for CloudWatch, Dashboards, Alarms, Rules, Event Buses, Logs, and Metrics. The main area is titled "Dashboards" and contains a "Create dashboard" button. Below it, there's a table showing existing dashboards: "cloud" (Last updated: 2017-07-31 10:10) and "cloud1" (Last updated: 2017-07-31 10:14). A modal window titled "Create new dashboard" is open, prompting for a "Dashboard name". The bottom of the screen shows the browser's address bar with the URL <https://us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#dashboards>, and the status bar indicates the date and time as 7/31/2017 3:48 PM.

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



The screenshot shows the AWS CloudWatch Metrics Dashboard configuration interface. The sidebar on the left lists "CloudWatch", "Dashboards" (selected), "cloud2", "Alarms", "Rules", "Event Buses", "Logs", and "Metrics". The main area is titled "Add to this dashboard" and displays four widget options: "Line" (Compare metrics over time), "Stacked area" (Compare the total over time), "Number" (Instantly see the latest value for a metric), and "Text" (Free text with markdown formatting). At the bottom of the modal are "Cancel" and "Configure" buttons. The bottom of the screen shows the browser's address bar with the URL <https://us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#dashboards:name=cloud2>, and the status bar indicates the date and time as 7/31/2017 3:48 PM.

5. After adding the instance it shows utilization of instance



2. VPC MANAGEMENT

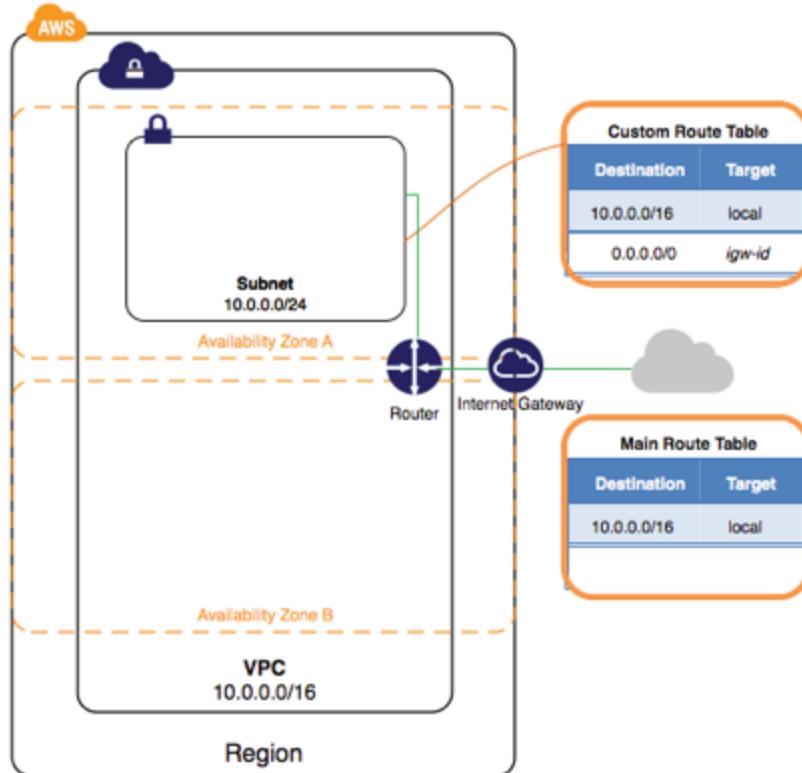
2.1 Objective

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

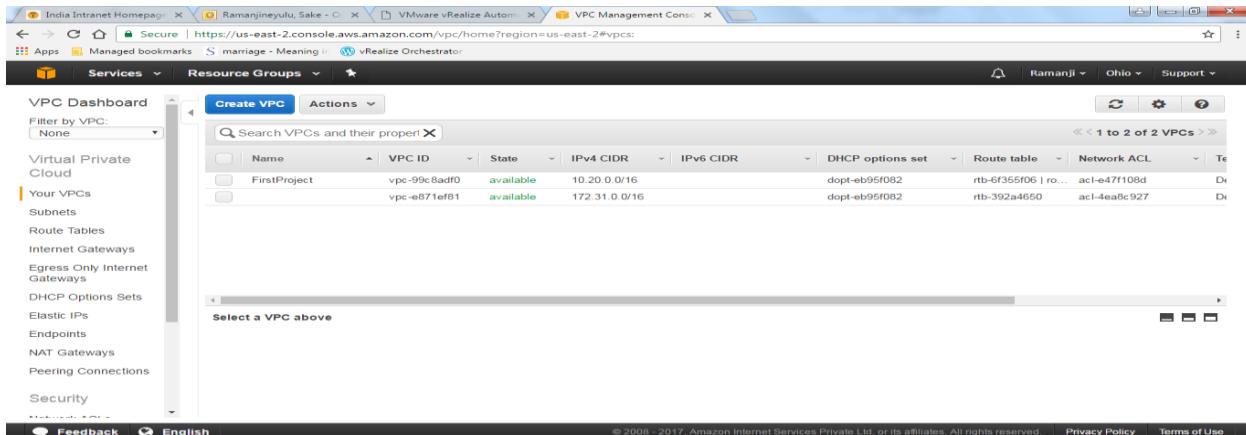
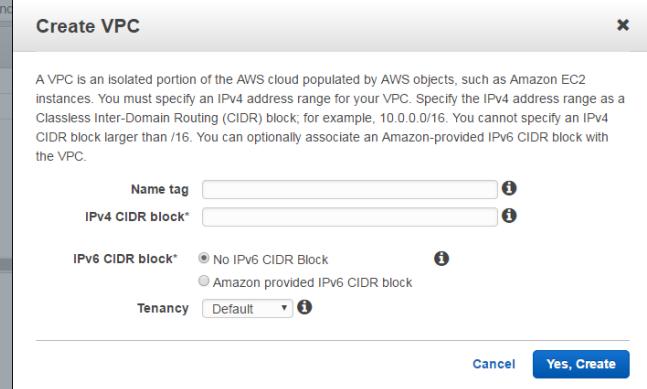
To set up a VPC, you need to complete the following steps:

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

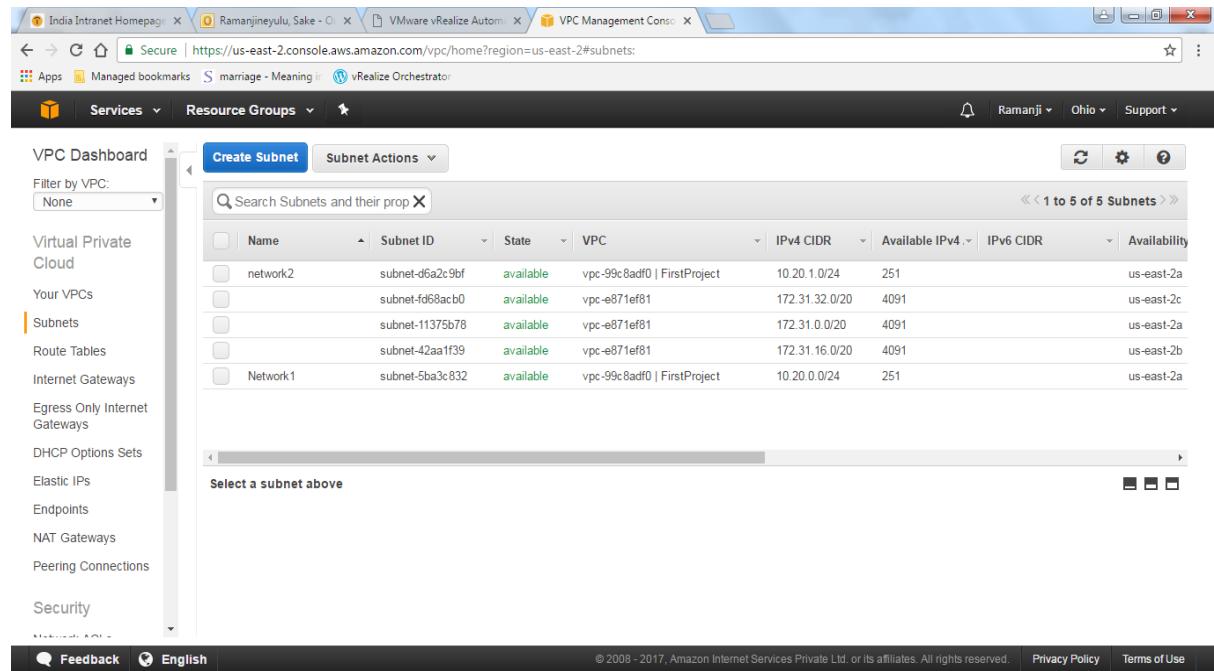
- Give your name for your VPC.
- Specify the range for your VPC in the form of CIDR block.

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

2.3.2 Subnets

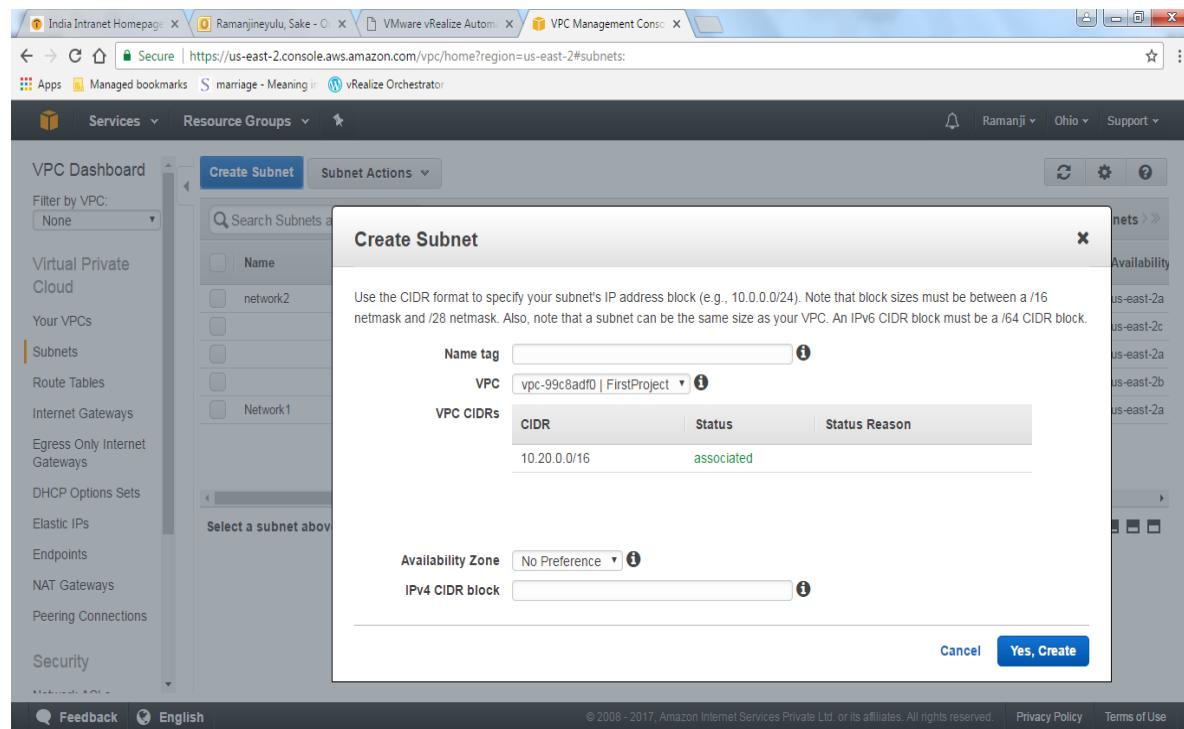
- Subnet is certain Range of IP address.

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



| Name | Subnet ID | State | VPC | IPv4 CIDR | Available IPv4 | IPv6 CIDR | Availability |
|----------|-----------------|-----------|-----------------------------|----------------|----------------|-----------|--------------|
| network2 | subnet-d6a2c9bf | available | vpc-99c8adf0 FirstProject | 10.20.1.0/24 | 251 | | us-east-2a |
| | subnet-fd68acb0 | available | vpc-e871ef81 | 172.31.32.0/20 | 4091 | | us-east-2c |
| | subnet-11375b78 | available | vpc-e871ef81 | 172.31.0.0/20 | 4091 | | us-east-2a |
| | subnet-42aa1f39 | available | vpc-e871ef81 | 172.31.16.0/20 | 4091 | | us-east-2b |
| Network1 | subnet-5ba3c832 | available | vpc-99c8adf0 FirstProject | 10.20.0.0/24 | 251 | | us-east-2a |

Click on Create Subnet

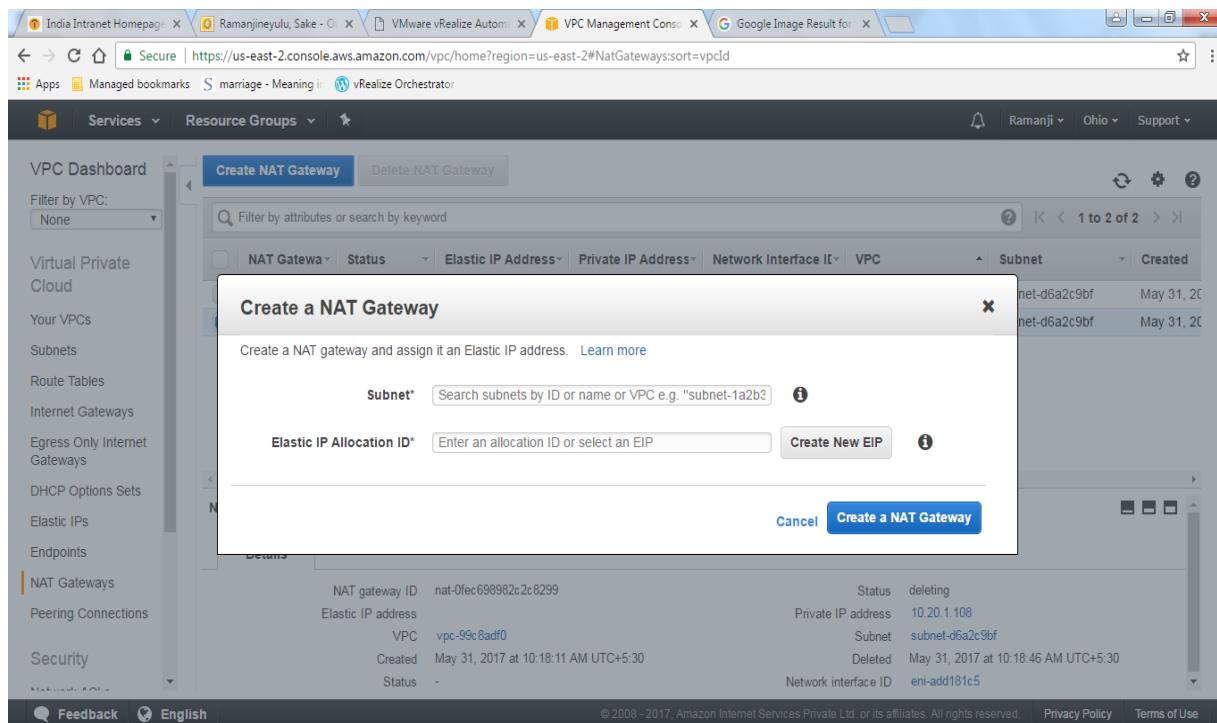


| CIDR | Status | Status Reason |
|--------------|------------|---------------|
| 10.20.0.0/16 | associated | |

- 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.



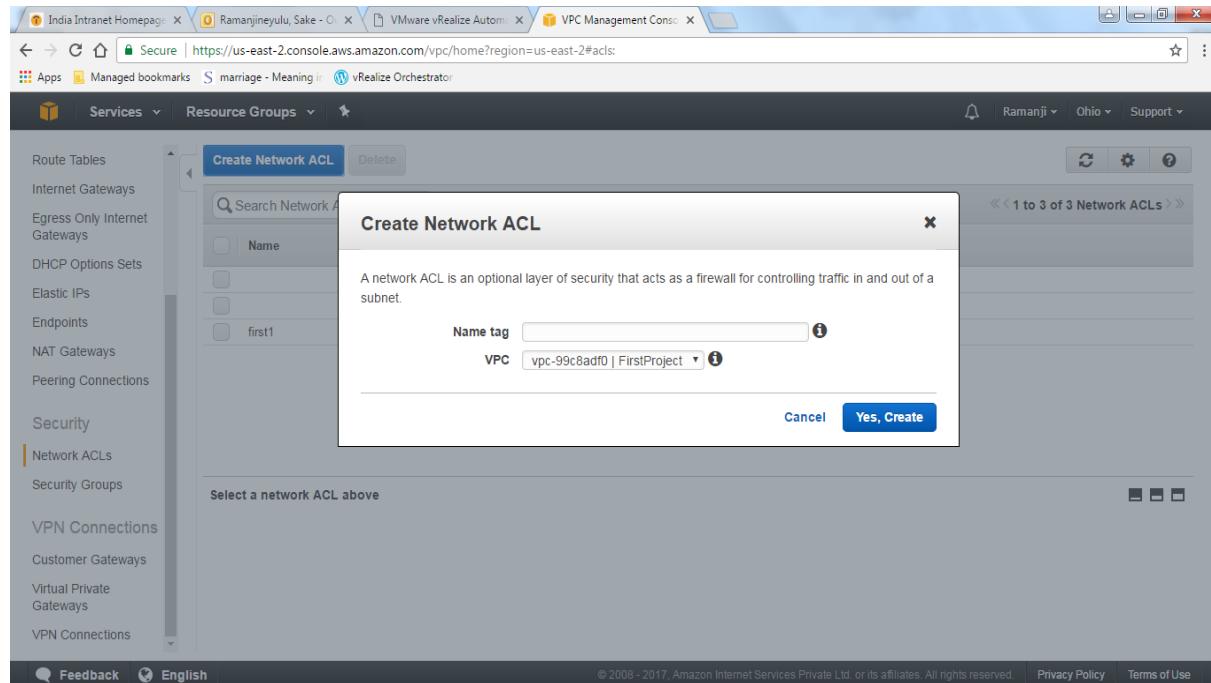
The screenshot shows the AWS VPC Management Console interface. On the left, there's a sidebar with navigation links like 'Virtual Private Cloud', 'Subnets', 'Route Tables', etc. The main area has a title 'Create NAT Gateway' and a search bar. Below it, there's a table showing existing NAT gateways. A modal window titled 'Create a NAT Gateway' is open in the center. It contains fields for 'Subnet*' (with a dropdown menu) and 'Elastic IP Allocation ID*' (with a text input field and a 'Create New EIP' button). At the bottom of the modal are 'Cancel' and 'Create a NAT Gateway' buttons. The background table lists two entries:

| Subnet | Created |
|-----------------|--------------|
| subnet-d6a2c9bf | May 31, 2017 |
| subnet-d6a2c9bf | May 31, 2017 |

- 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.

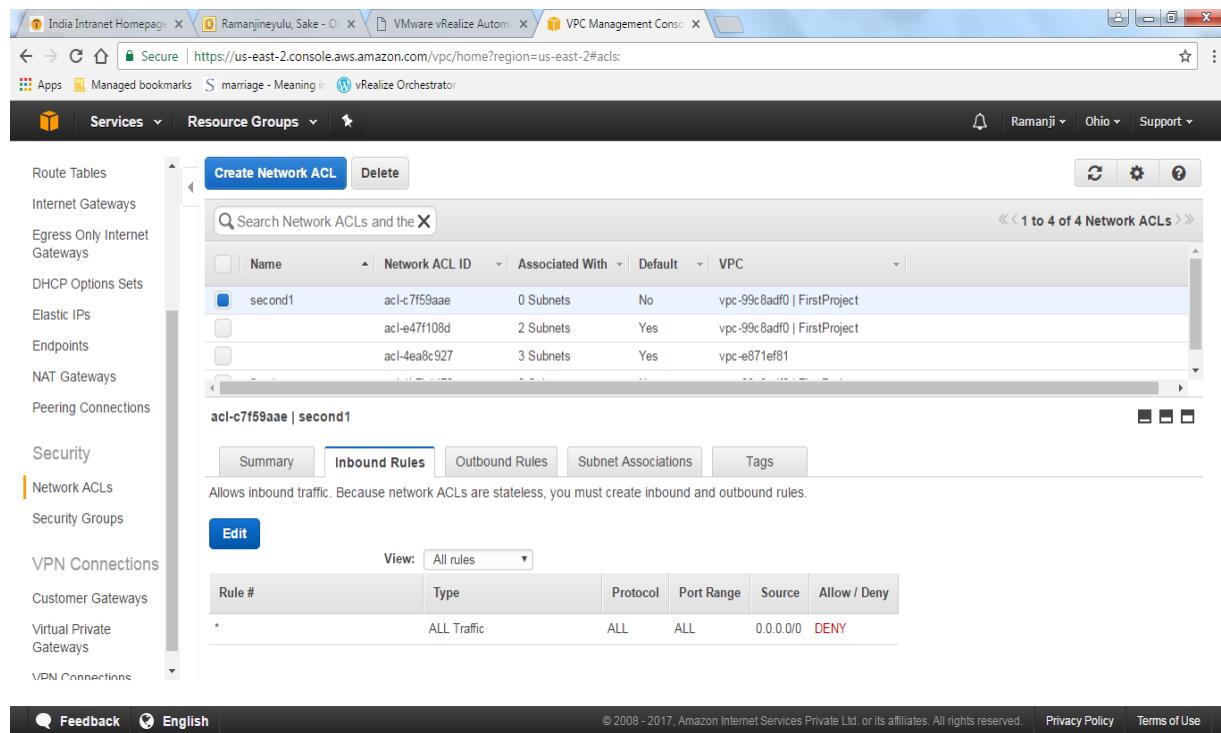
NACL (Network Access control list)

- 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.



The screenshot shows the AWS VPC Management Console. On the left, there's a sidebar with various network-related services like Route Tables, Internet Gateways, and Security Groups. The 'Network ACLs' option under 'Security' is selected. A central modal window titled 'Create Network ACL' is open, asking for a 'Name tag' which is set to 'first1'. Below this, it shows the 'VPC' associated with the new ACL. At the bottom of the modal are 'Cancel' and 'Yes, Create' buttons.

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



This screenshot shows the list of Network ACLs in the AWS VPC Management Console. There are four entries: 'second1', 'acl-e47f108d', 'acl-4ea8c927', and 'acl-e871ef81'. The 'second1' row is highlighted. Below the table, a modal window for 'Edit Inbound Rules' is open. It has tabs for 'Summary', 'Inbound Rules' (which is selected), 'Outbound Rules', 'Subnet Associations', and 'Tags'. Under 'Edit Inbound Rules', there's a table with columns: Rule #, Type, Protocol, Port Range, Source, and Allow / Deny. The first rule listed is '*' with 'ALL Traffic' as the type, 'ALL' for both protocol and port range, and '0.0.0.0/DENY' as the source and action.

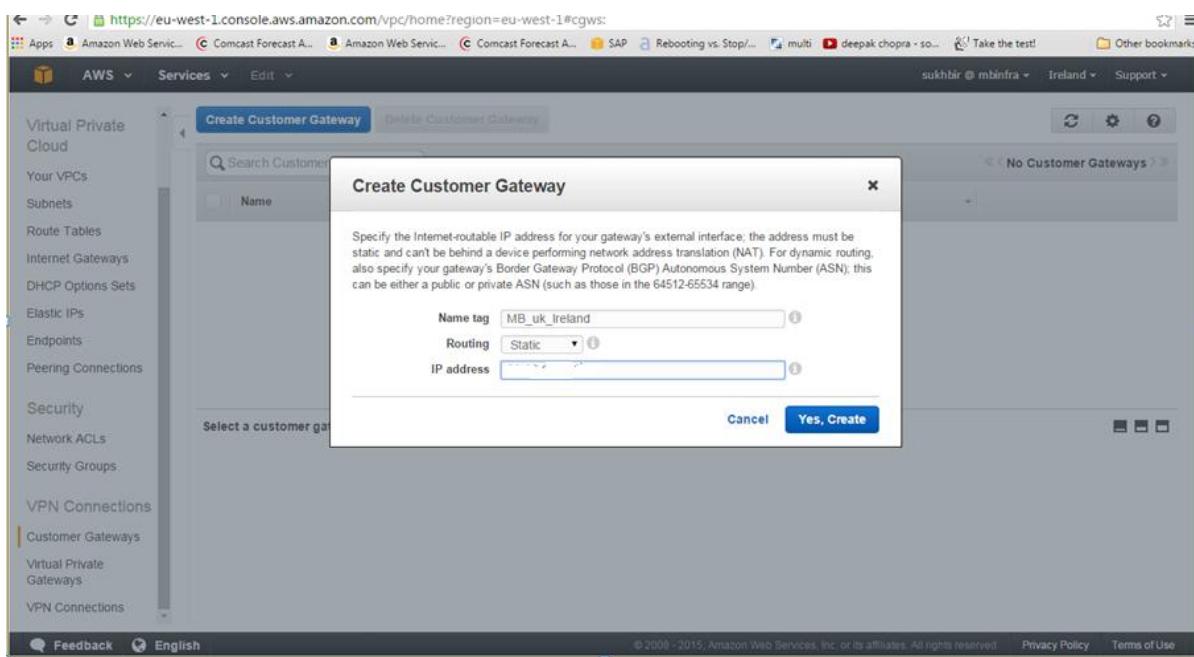
- Click on edit and give inbound rules for your VPC which control the incoming traffic to your VPC.

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 | AWS Services Edit | https://eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#vgws: Apps Amazon Web Service... Comcast Forecast A... Amazon Web Service... Comcast Forecast A... SAP Rebooting vs. Stop... multi deepak chopra - so... Take the test! Other bookmarks 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

| 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 | AWS Services Edit | https://eu-west-1.console.aws.amazon.com/vpc/home?region=eu-west-1#vgws: Apps Amazon Web Service... Comcast Forecast A... Amazon Web Service... Comcast Forecast A... SAP Rebooting vs. Stop... multi deepak chopra - so... Take the test! Other bookmarks 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

| Name | ID | State | Type | VPC |
|-----------|--------------|----------|---------|-----|
| MB_uk_VPG | vgw-6856671c | detached | ipsec.1 | |

vgw-6856671c | MB_uk_VPG

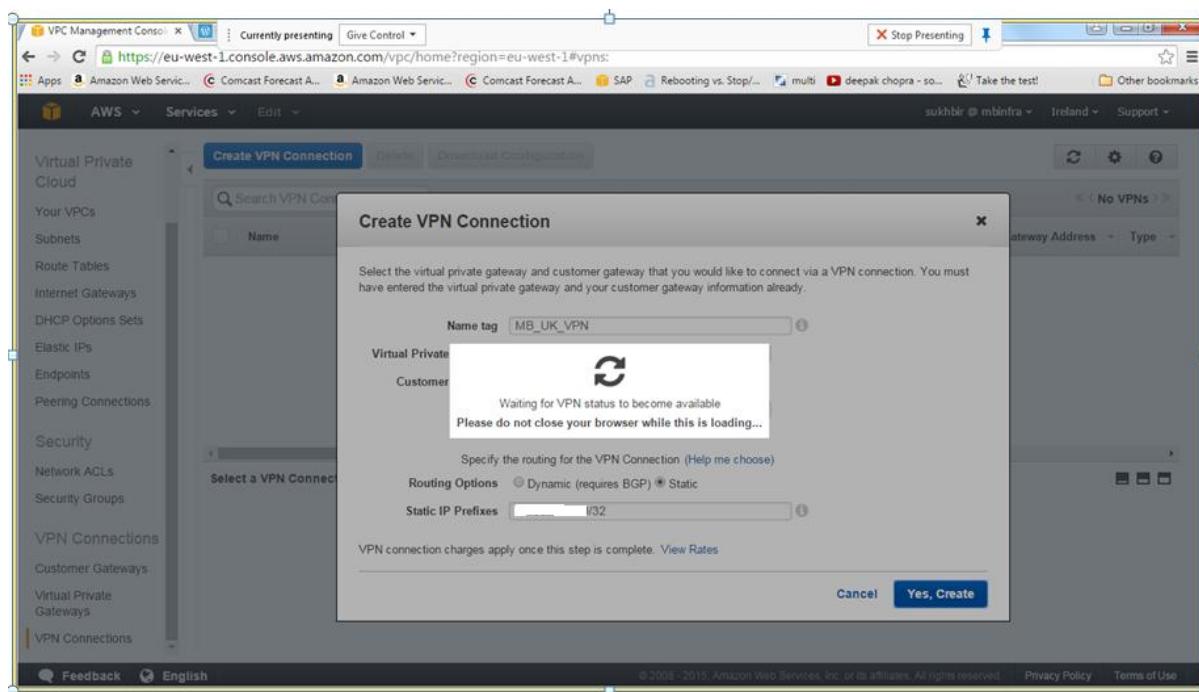
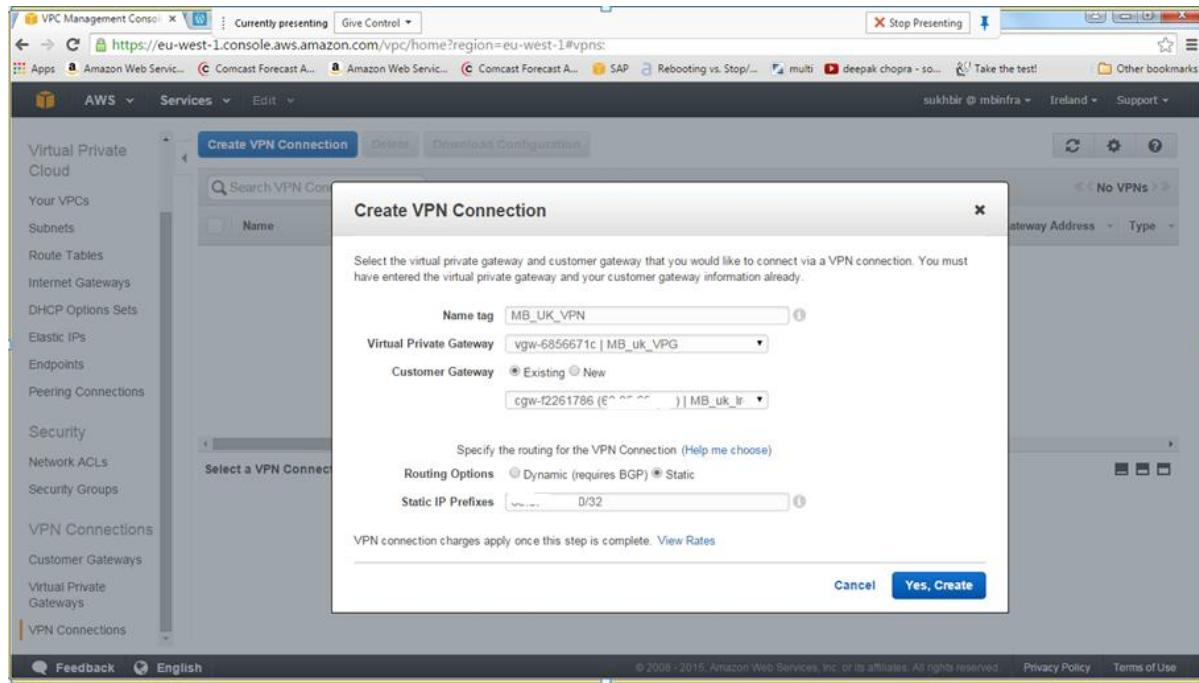
Attach to VPC

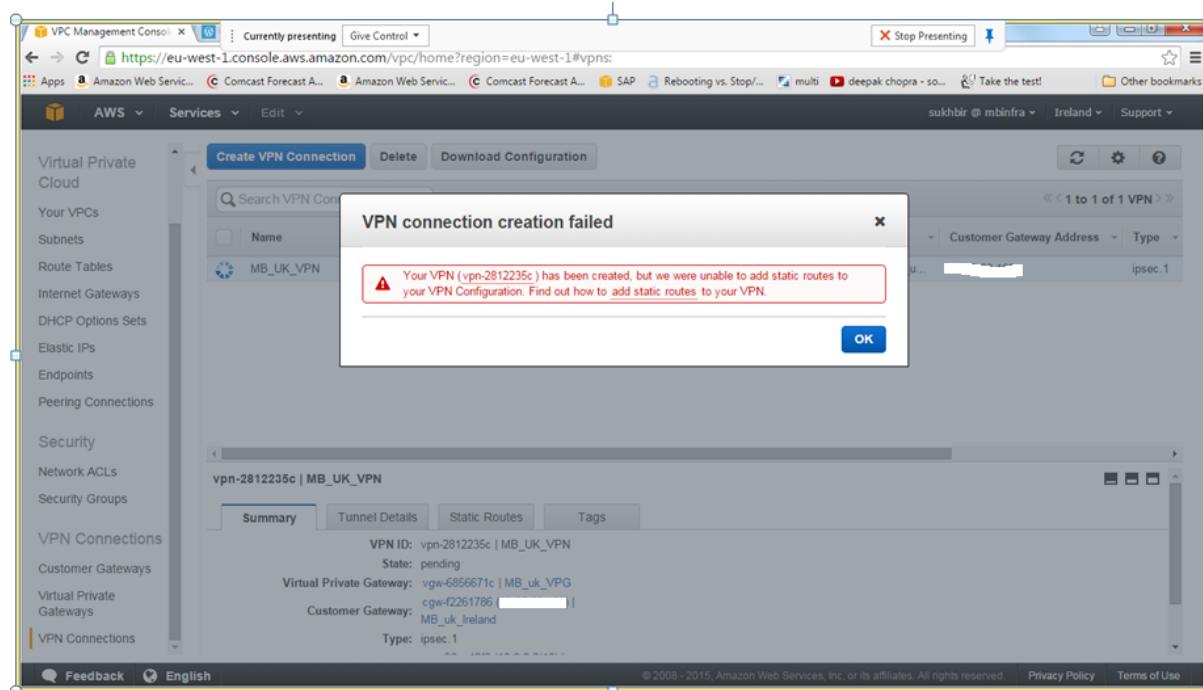
Select the VPC to attach to the virtual private gateway

VPC vpc-96aa12f3 (10.0.0.0/16) | MBDevVPC

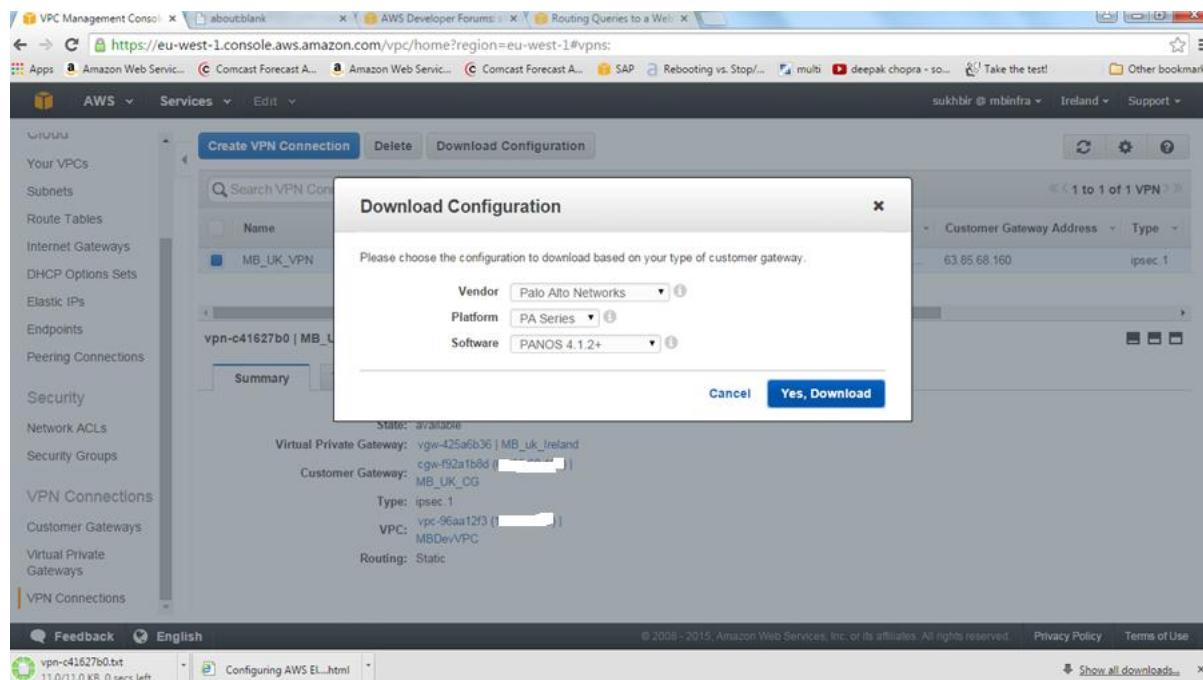
Cancel Yes, Attach

Feedback English © 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use





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.

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

Additional Information

- [IAM best practices](#)
- [IAM documentation](#)
- [Web Identity Federation Playground](#)
- [Policy Simulator](#)
- [Videos, IAM release history and additional resources](#)

Add user **Delete user**

Find users by username or access key

| User name | Groups | Access key age | Password age | Last activity | MFA |
|--|--------|----------------|--------------|---------------|-----|
| There are no IAM users. Learn more | | | | | |

Step 2:

User name* [+ 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.

Services ▾ Resource Groups ▾ Pooja Mali Global Support

Add user

Set permissions for Testuser1

1 Details 2 Permissions 3 Review 4 Complete

Attach existing policies directly

Add one or more existing policies directly to the user or create a new policy. Learn more

Create policy Refresh

Filter: Policy type ▾ Q: readonly Showing 64 results

| | Policy name | Type | Attachments | Description |
|-------------------------------------|---|-------------|-------------|---|
| <input type="checkbox"/> | CloudSearchReadOnlyAccess | AWS managed | 0 | Provides read only access to the Amazon CloudSearch configuration service. |
| <input type="checkbox"/> | CloudWatchEventsReadOnlyAccess | AWS managed | 0 | Provides read only access to Amazon CloudWatch Events. |
| <input type="checkbox"/> | CloudWatchLogsReadOnlyAccess | AWS managed | 0 | Provides read only access to CloudWatch Logs |
| <input type="checkbox"/> | CloudWatchReadOnlyAccess | AWS managed | 0 | Provides read only access to CloudWatch |
| <input type="checkbox"/> | IAMReadOnlyAccess | AWS managed | 0 | Provides read only access to IAM via the AWS Management Console. |
| <input type="checkbox"/> | QuickSightAccessForS3StorageManagementAnalyticsReadOnly | AWS managed | 0 | Policy used by QuickSight team to access customer data produced by S3 Storage Management Analytics. |
| <input checked="" type="checkbox"/> | ReadOnlyAccess | AWS managed | 0 | Provides read-only access to AWS services and resources. |
| <input type="checkbox"/> | 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 |

Create user

Cancel Previous Create user

Feedback English © 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Add user

1 2 3 4

Details Permissions Review Complete

Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://197899259986.signin.aws.amazon.com/console>

[Download .csv](#)

| User | Access key ID | Secret access key | Email login instructions |
|-----------|----------------------|---|----------------------------|
| Testuser1 | AKIAJEME2X7YLSW4KL6Q | qaiSbRiwyoTzr6dSreOEYszm4vqP8oV2V qxdxg Hide | Send email |

[Close](#)

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

Welcome to Identity and Access Management

IAM users sign-in link:
<https://197899259986.signin.aws.amazon.com/console>

Customize | Copy Link

Security Status
2 out of 5 complete.

- ⚠ Activate MFA on your root account
- ✓ Create individual IAM users
- ⚠ Use groups to assign permissions
- ⚠ Apply an IAM password policy
- ✓ Rotate your access keys

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

Secure | https://signin.aws.amazon.com/oauth

Apps Managed bookmarks For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

amazon web services

You must change your password to continue

AWS account 197899259986

IAM user name Testuser1

Old password
New password
Retype new password

Confirm password change

[Sign-in using root account credentials](#)

English ▾

[Terms of Use](#) [Privacy Policy](#) © 1996-2017, Amazon Web Services, Inc. or its affiliates.

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

Services ▾ | Resource Groups ▾ | ★

Testuser1 @ 1978-9925-9986 ▾ | Global ▾ | Support ▾

Manage User Permissions

Edit Permissions

The policy generator enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [Overview of Policies](#) in Using AWS Identity and Access Management.

Effect Allow Deny

AWS Service AWS Application Discovery S...

Actions -- Select Actions --

Amazon Resource Name (ARN)

Add Conditions (optional)

Add Statement

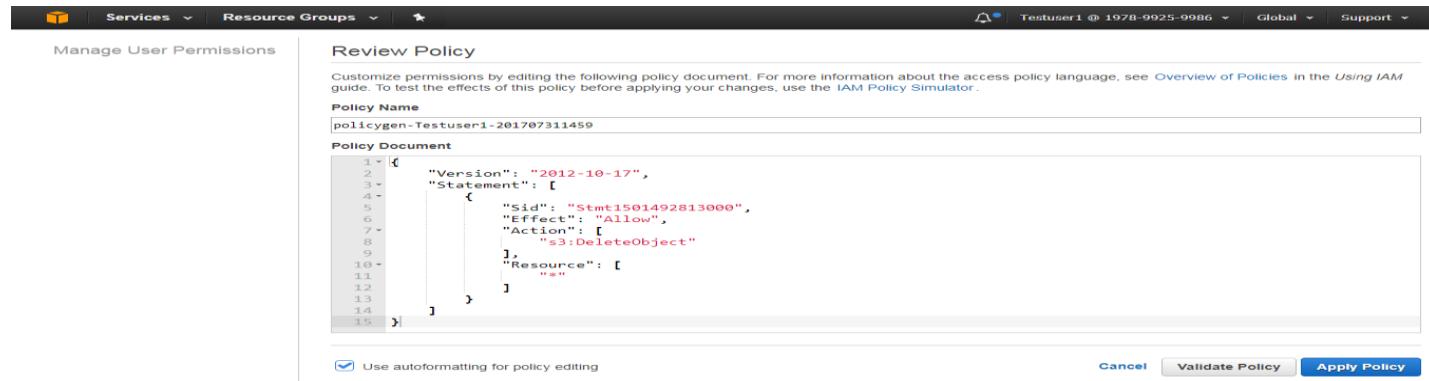
| Effect | Action | Resource | |
|--------|-----------------|----------|--------|
| Allow | s3:DeleteObject | * | Remove |

Cancel Previous Next Step

In above, “ * ” is to give all permissions on action selected. Otherwise to give specific permission on action selected then click on Amazon Resource 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



The screenshot shows the 'Review Policy' page in the AWS IAM console. The policy name is 'policygen-Testuser1-201707311459'. The policy document is displayed as follows:

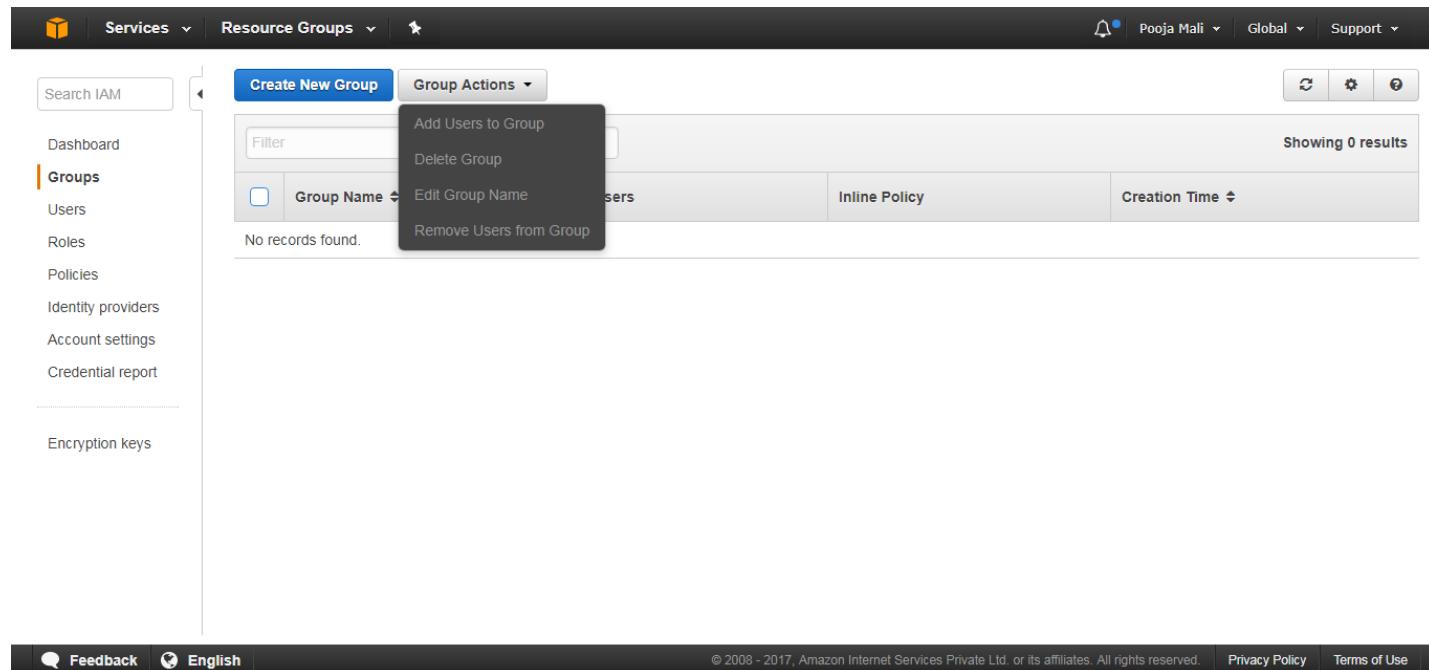
```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "Stmt1501492813000",
6        "Effect": "Allow",
7        "Action": [
8          "s3:DeleteObject"
9        ],
10       "Resource": [
11         "*"
12       ]
13     }
14   ]
15 }
```

Below the policy document, there is a checkbox for 'Use autoformatting for policy editing'. At the bottom right, there are 'Cancel', 'Validate Policy', and 'Apply Policy' buttons.

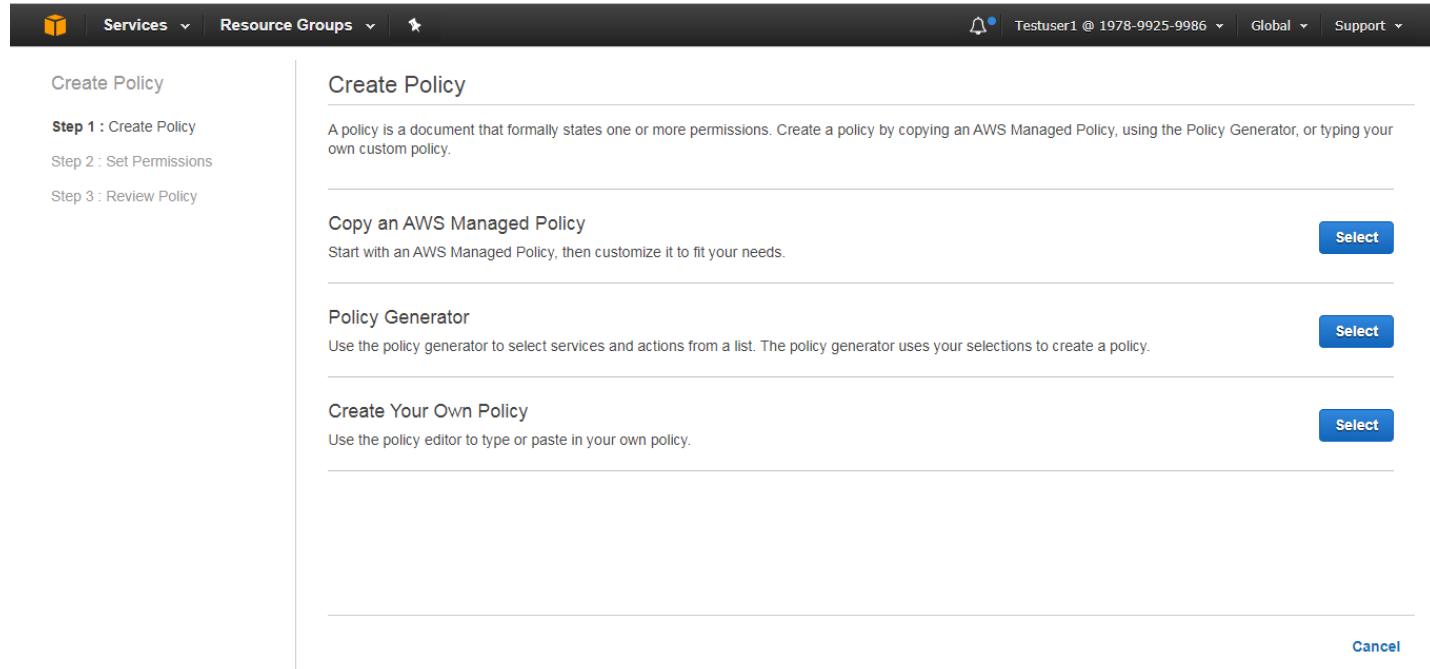
Step 3:

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



The screenshot shows the 'Groups' page in the AWS IAM console. The 'Groups' link in the left sidebar is highlighted. A dropdown menu titled 'Group Actions' is open, showing options: 'Add Users to Group', 'Delete Group', 'Edit Group Name', and 'Remove Users from Group'. The main table lists groups with columns for 'Group Name', 'Users', 'Inline Policy', and 'Creation Time'. A message at the bottom of the table says 'No records found.'.

Step 5:



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.

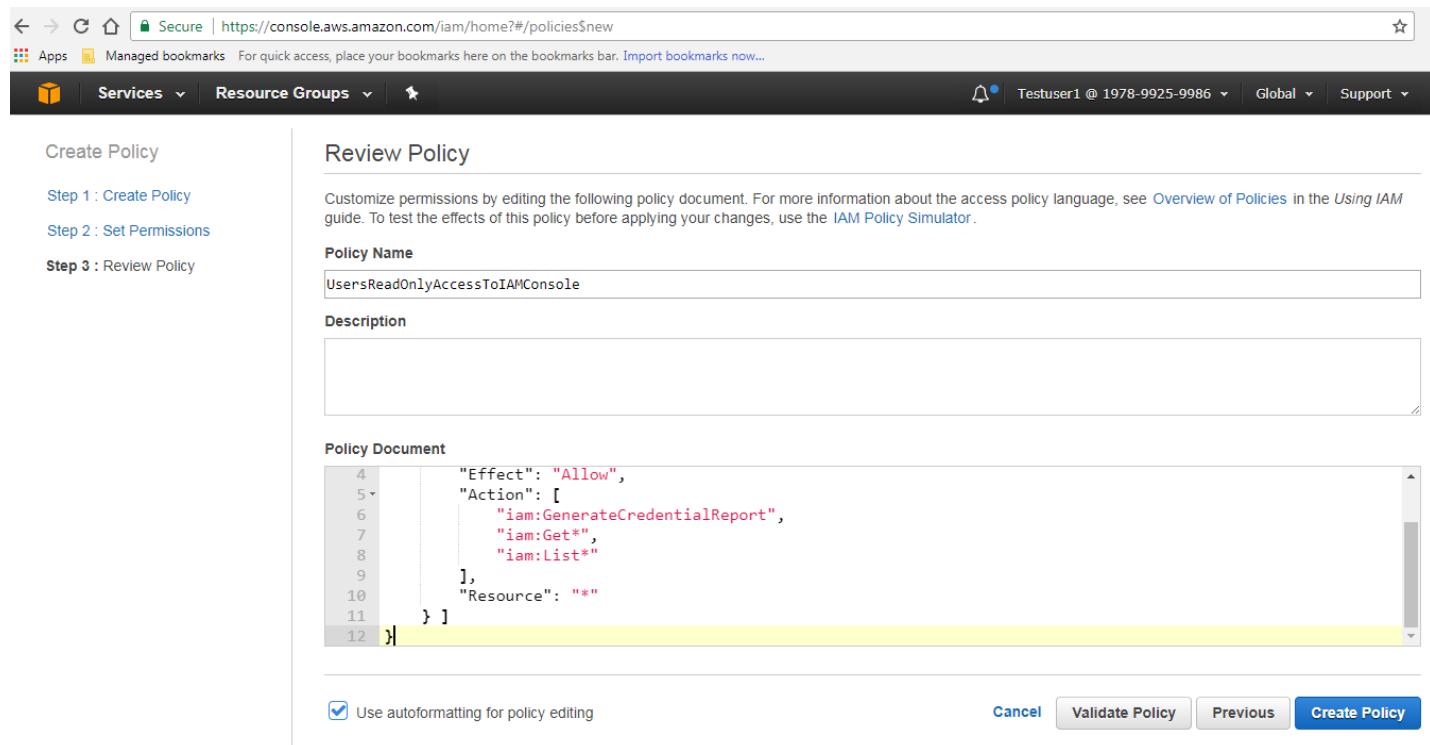
Select

Create Your Own Policy

Use the policy editor to type or paste in your own policy.

Select

Cancel



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

```

4   "Effect": "Allow",
5   "Action": [
6     "iam:GenerateCredentialReport",
7     "iam:Get*",
8     "iam>List*"
9   ],
10  "Resource": "*"
11 }
12

```

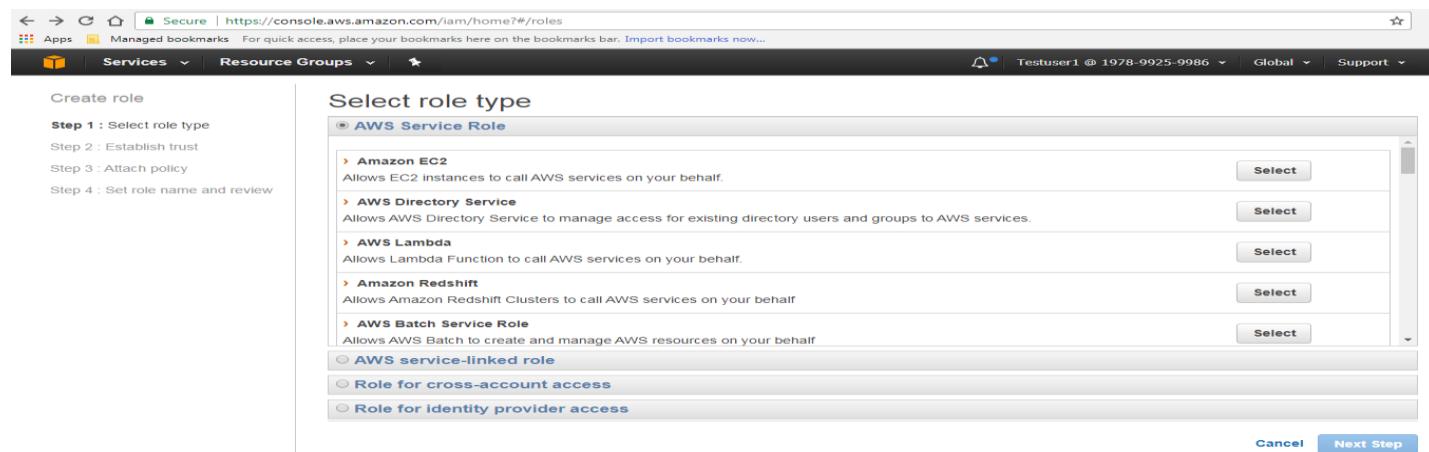
Use autoformatting for policy editing

Create Policy

Now validate and create a policy.

Step 5 Creating IAM Roles:

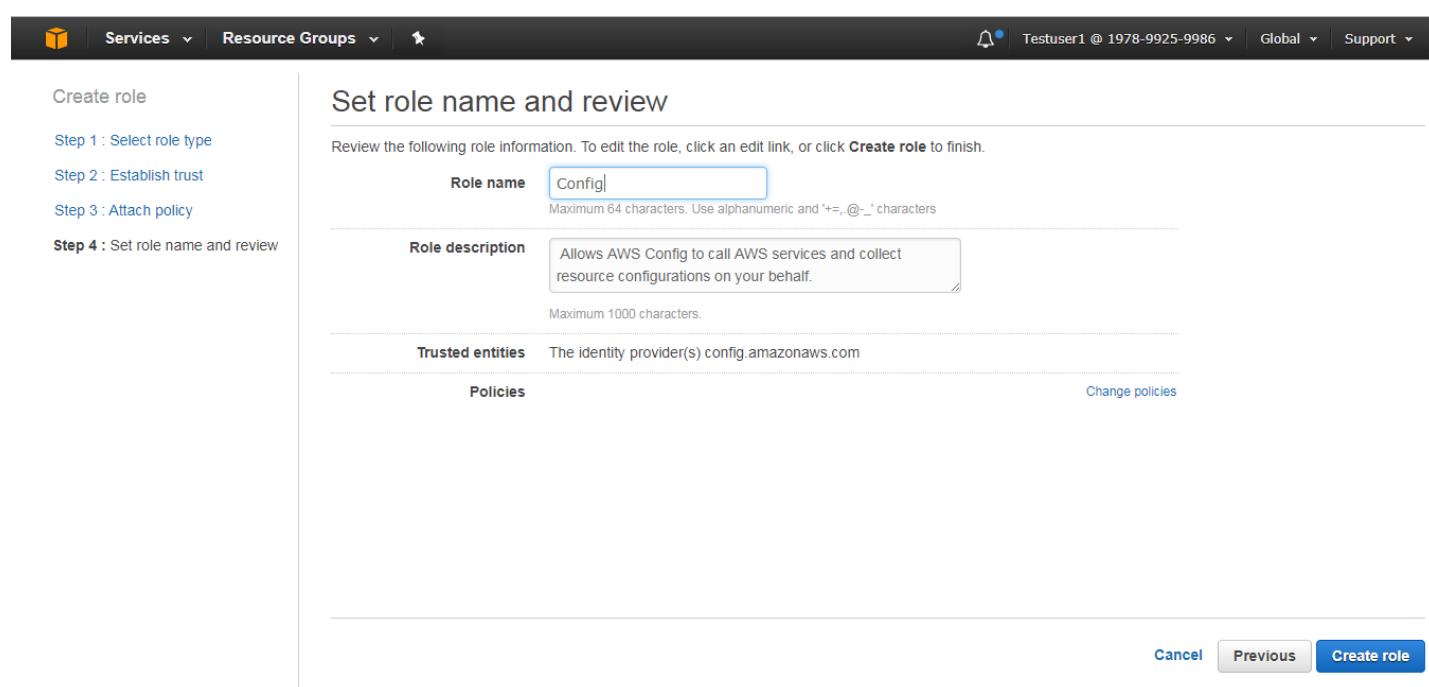
Before an IAM user, application, or service can use a role that you created, you must grant permissions to switch to the role. You can use any policy attached to one of an IAM user's groups or to the user itself to grant the necessary permissions. This section describes how to grant users permission 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.



Select role type

AWS Service Role

- Amazon EC2**
Allows EC2 instances to call AWS services on your behalf.
- AWS Directory Service**
Allows AWS Directory Service to manage access for existing directory users and groups to AWS services.
- AWS Lambda**
Allows Lambda Function to call AWS services on your behalf.
- Amazon Redshift**
Allows Amazon Redshift Clusters to call AWS services on your behalf.
- AWS Batch Service Role**
Allows AWS Batch to create and manage AWS resources on your behalf.
- AWS service-linked role**
- Role for cross-account access**
- Role for identity provider access**



Create role

Step 1 : Select role type
Step 2 : Establish trust
Step 3 : Attach policy
Step 4 : Set role name and review

Set role name and review

Review the following role information. To edit the role, click an edit link, or click **Create role** to finish.

| | |
|---|--|
| Role name | Config |
| Maximum 64 characters. Use alphanumeric and '+-, @-' characters | |
| Role description | Allows AWS Config to call AWS services and collect resource configurations on your behalf. |
| Maximum 1000 characters. | |
| Trusted entities | The identity provider(s) config.amazonaws.com |
| Policies | <input type="button" value="Change policies"/> |

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.

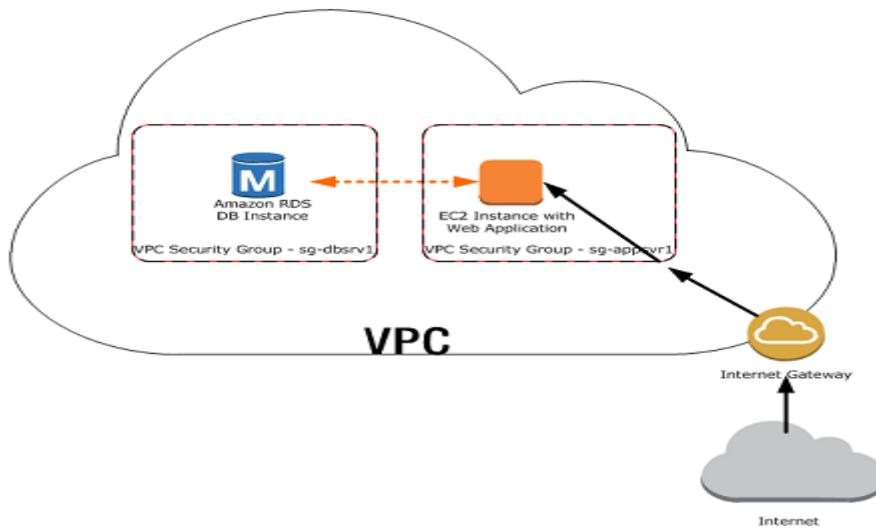
4.1 Objective

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

4.2 Assumption

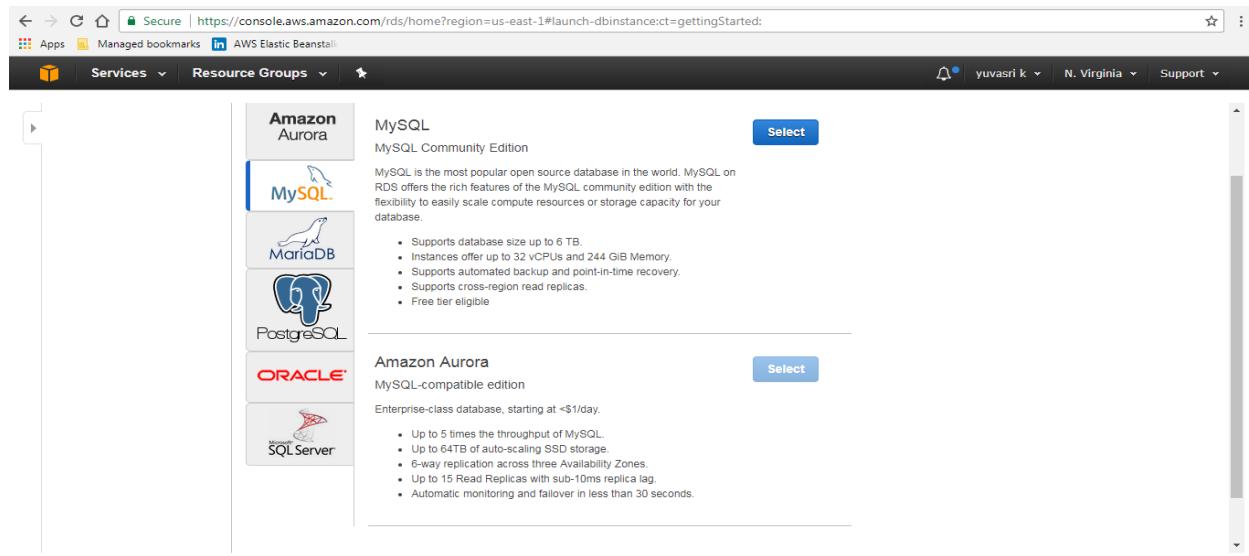
- Amazon RDS is available on several database instance types, optimized for Memory, Performance and I/O, and Provides with six familiar database engines to choose from Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle and Microsoft SQL Server.

The below diagram shows how the AWS RDS is connected:



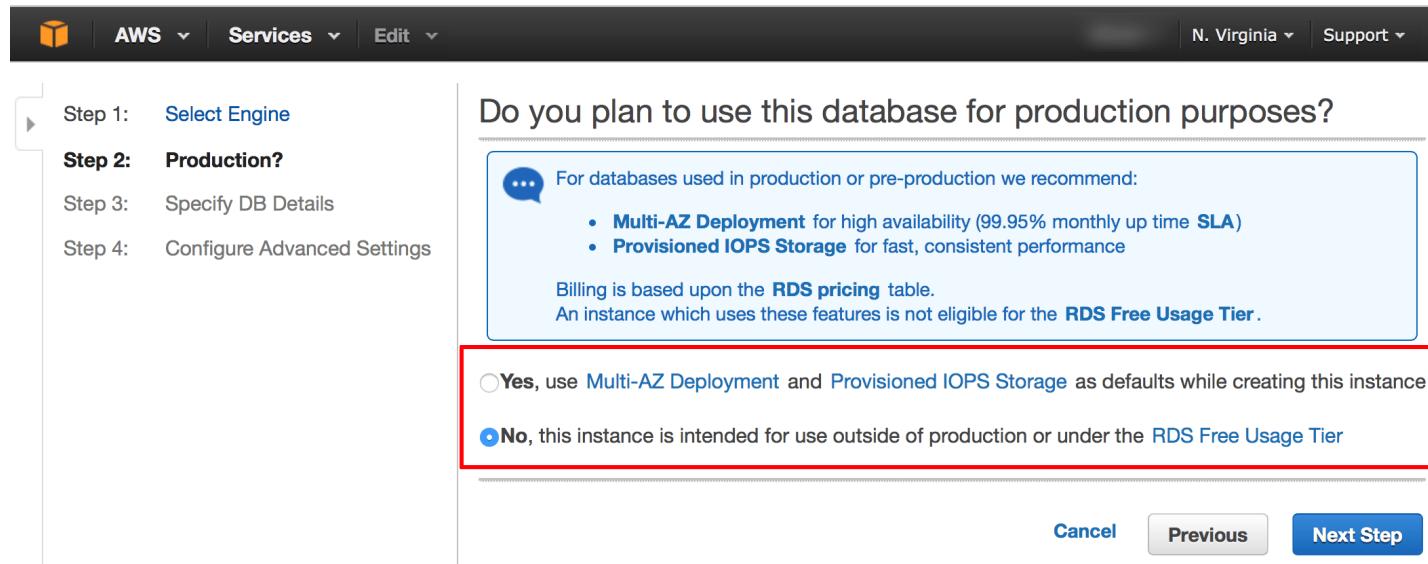
4.3 Procedure

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



The screenshot shows the AWS RDS console at the URL <https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstancect=gettingStarted>. The left sidebar shows 'Services' and 'Resource Groups'. The main area displays a list of database engines: MySQL, Amazon Aurora, MariaDB, PostgreSQL, Oracle, and SQL Server. MySQL is currently selected, indicated by a blue border around its icon and the word 'Select' next to its name. A detailed description of MySQL Community Edition follows, listing its features like support for up to 6 TB of database size and automated backups.

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



The screenshot shows the AWS RDS instance creation wizard at Step 2: Select Engine. The top navigation bar includes 'AWS', 'Services', 'Edit', 'N. Virginia', and 'Support'. On the left, a sidebar lists steps: Step 1: Select Engine, Step 2: Production?, Step 3: Specify DB Details, and Step 4: Configure Advanced Settings. The main content area asks 'Do you plan to use this database for production purposes?'. It provides a recommendation for production databases: 'For databases used in production or pre-production we recommend: • Multi-AZ Deployment for high availability (99.95% monthly up time SLA) • Provisioned IOPS Storage for fast, consistent performance'. Below this, it states 'Billing is based upon the [RDS pricing](#) table. An instance which uses these features is not eligible for the [RDS Free Usage Tier](#)'. At the bottom, two radio button options are shown: Yes, use [Multi-AZ Deployment](#) and [Provisioned IOPS Storage](#) as defaults while creating this instance and No, this instance is intended for use outside of production or under the [RDS Free Usage Tier](#). The 'Yes' option is highlighted with a red box.

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

Specify DB Details

Instance Specifications

| | |
|--|---------------------------------|
| DB Engine | mysql |
| License Model | general-public-license |
| DB Engine Version | 5.6.22 |
| Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions. | |
| DB Instance Class | db.t2.micro – 1 vCPU, 1 GiB RAM |
| Multi-AZ Deployment | No |
| Storage Type | Magnetic |
| Allocated Storage* | 50 GB |

Settings

| | |
|-------------------------|----------------------|
| DB Instance Identifier* | tutorial-db-instance |
| Master Username* | tutorial_user |
| Master Password* | ***** |
| Confirm Password* | ***** |

* Required [Cancel](#) [Previous](#) [Next Step](#)

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

Configure Advanced Settings

Network & Security

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

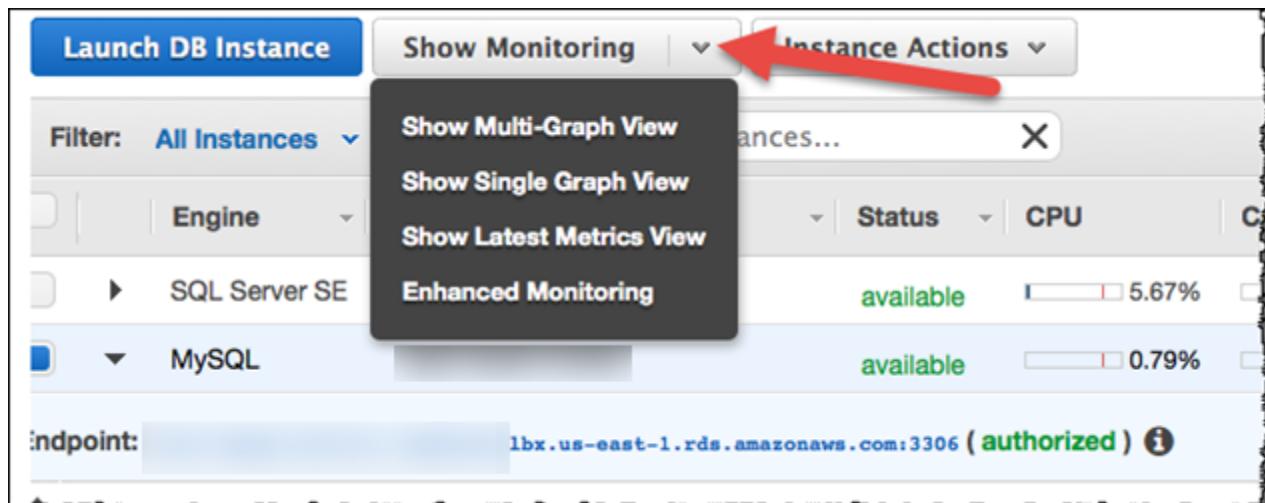
| | |
|-----------------------|---|
| VPC* | tutorial-vpc (vpc-f1b76594) |
| Subnet Group | Create new DB Subnet Group |
| Publicly Accessible | No |
| Availability Zone | No Preference |
| VPC Security Group(s) | Create new Security Group default (VPC) tutorial-db-security-group (VPC) tutorial-securitygroup (VPC) |

Database Options

| | |
|---------------|--------|
| Database Name | sample |
|---------------|--------|

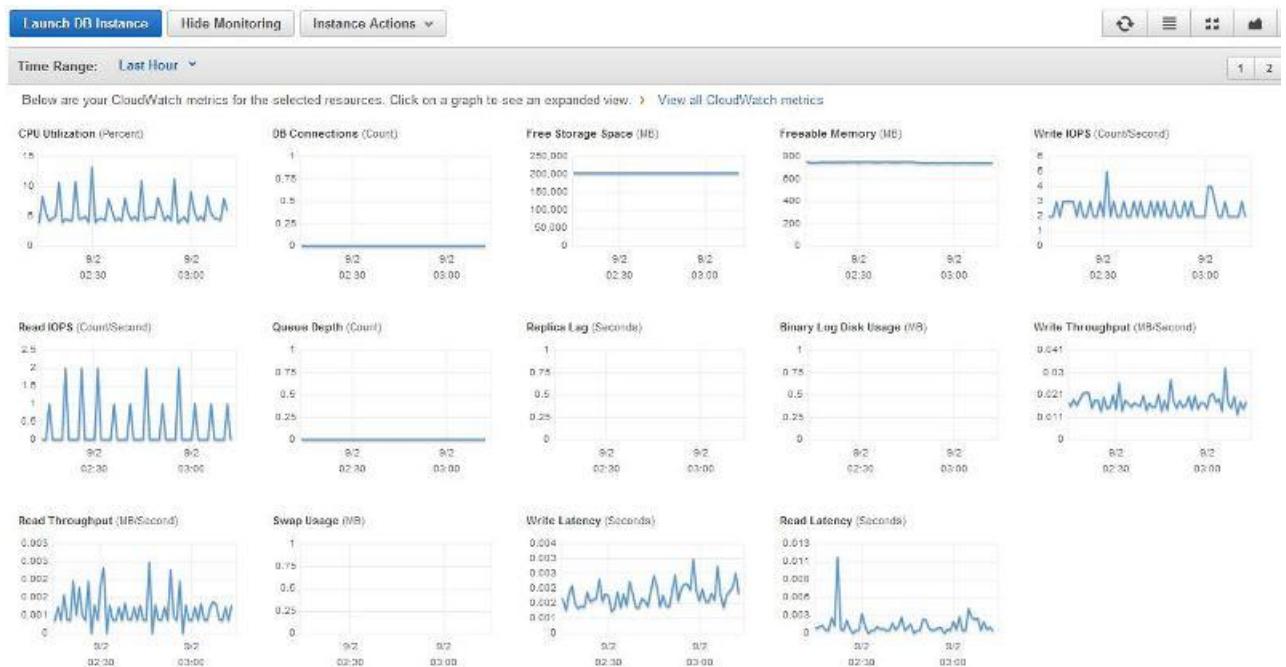
Note: Two databases will be created. One for MySQL and one for Oracle. Both databases will be mounted on the DB instance.

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



The screenshot shows the AWS RDS Instances page. At the top, there are buttons for "Launch DB Instance" and "Show Monitoring". A red arrow points to the "Instance Actions" dropdown menu, which is open and displays four options: "Show Multi-Graph View", "Show Single Graph View", "Show Latest Metrics View", and "Enhanced Monitoring". Below this, there is a table listing RDS instances. The first instance is "SQL Server SE" and the second is "MySQL". Both are marked as "available". On the right side of the table, there are status indicators for CPU usage (5.67% and 0.79%) and memory utilization.

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

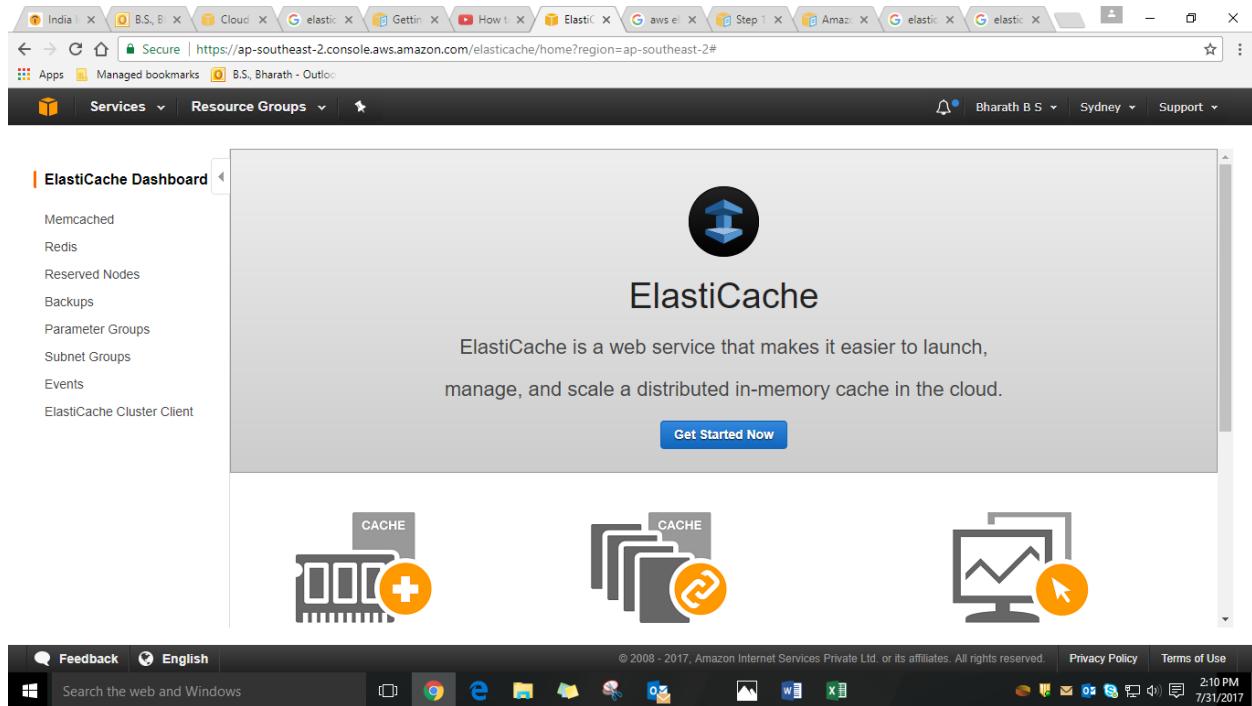
Step 4: Authorize Access

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 Elasticache



ElastiCache Dashboard

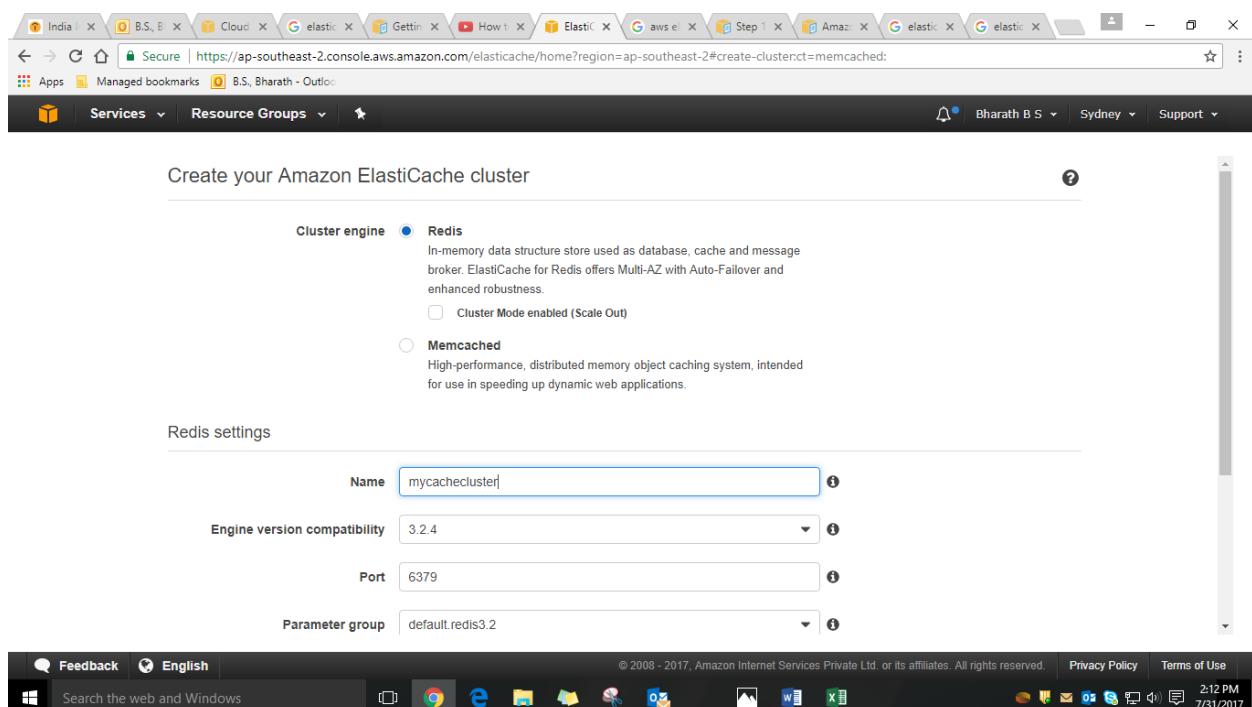
- Memcached
- Redis
- Reserved Nodes
- Backups
- Parameter Groups
- Subnet Groups
- Events
- ElastiCache Cluster Client

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](#)

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



Create your Amazon ElastiCache cluster

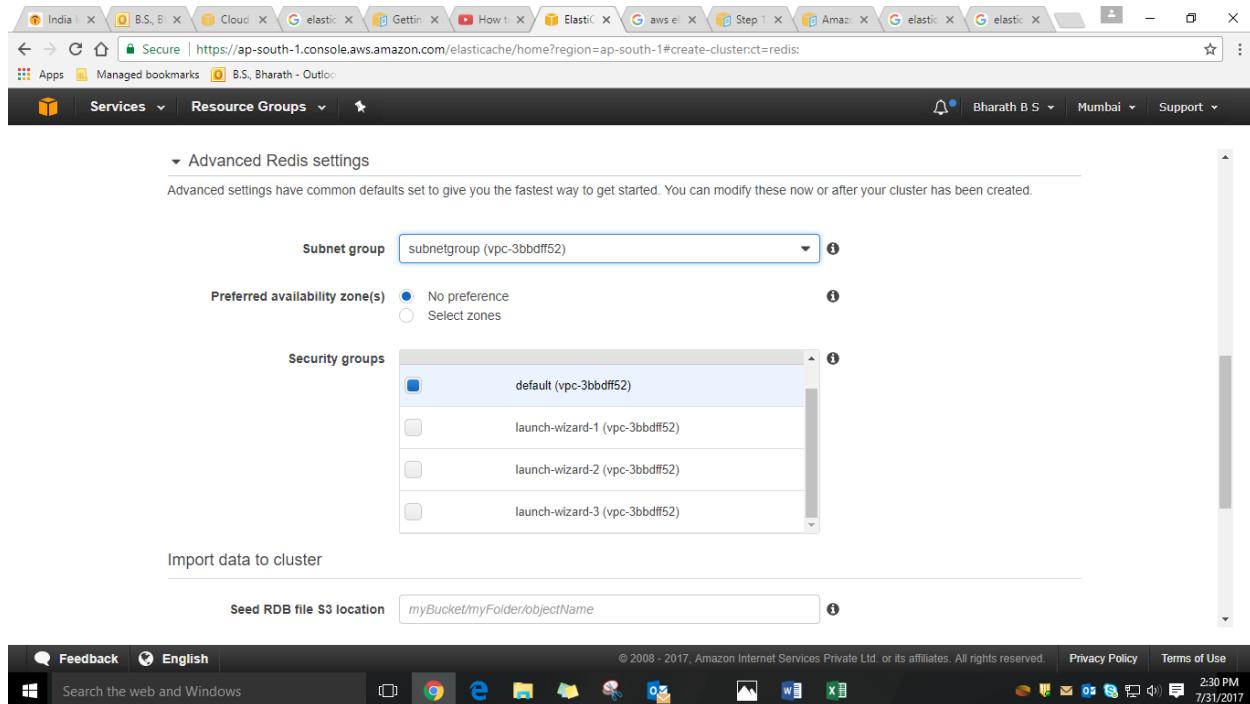
Cluster engine **Redis**
 In-memory data structure store used as database, cache and message broker. ElastiCache for Redis offers Multi-AZ with Auto-Failover and enhanced robustness.
 Cluster Mode enabled (Scale Out)

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 |

© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



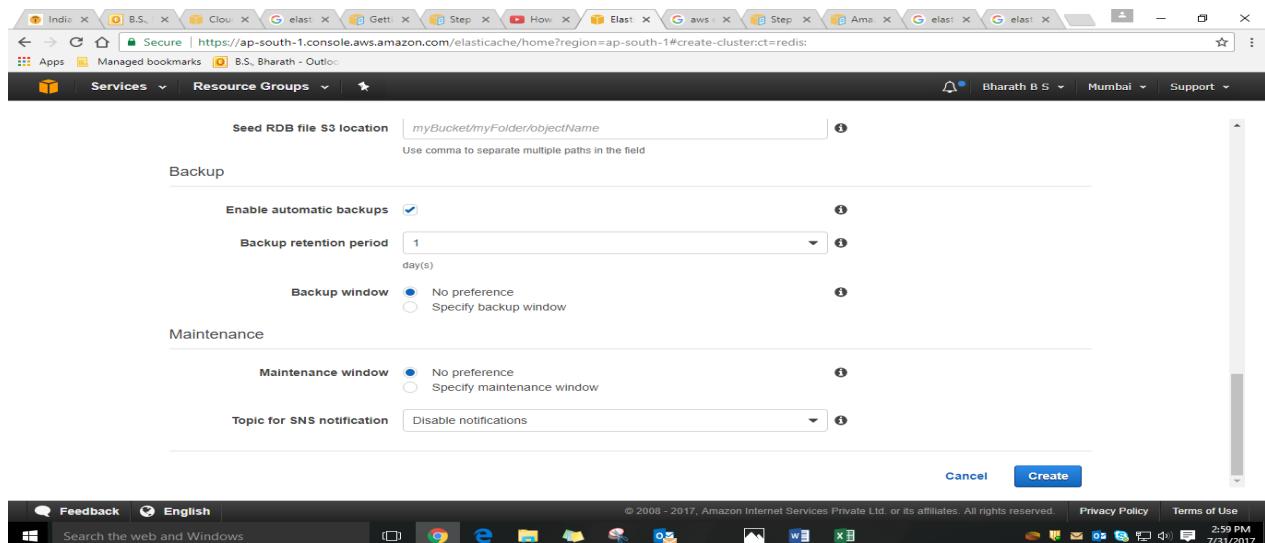
Subnet group: subnetgroup (vpc-3bbdff52)

Preferred availability zone(s): No preference

Security groups: default (vpc-3bbdff52)

Import data to cluster: Seed RDB file S3 location: myBucket/myFolder/ObjectName

5.3.2 View Cluster Details



Seed RDB file S3 location: myBucket/myFolder/ObjectName

Backup

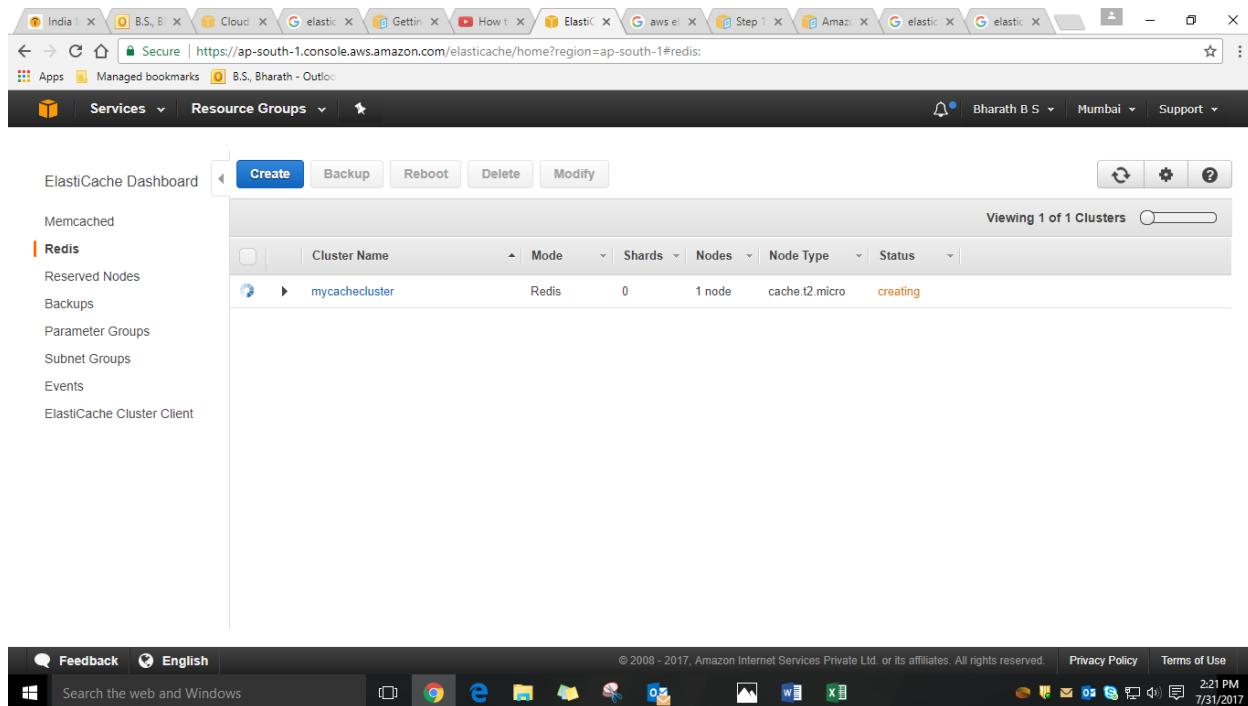
- Enable automatic backups: checked
- Backup retention period: 1 day(s)
- Backup window: No preference

Maintenance

- Maintenance window: No preference
- Topic for SNS notification: Disable notifications

Create

Wait till it is available



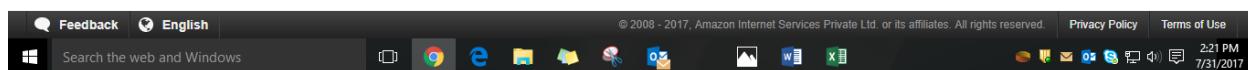
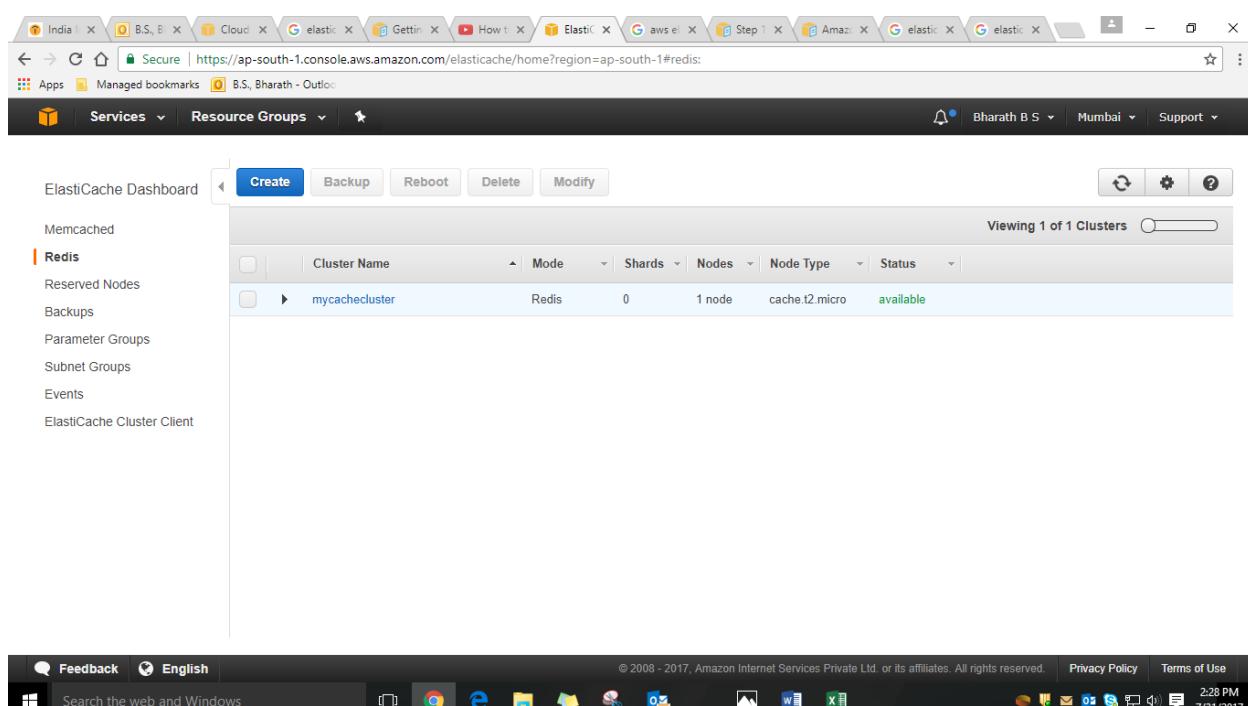
ElastiCache Dashboard

Create Backup Reboot Delete Modify

Viewing 1 of 1 Clusters

| | Cluster Name | Mode | Shards | Nodes | Node Type | Status |
|---|----------------|-------|--------|--------|----------------|----------|
| ▶ | mycachecluster | Redis | 0 | 1 node | cache.t2.micro | creating |

- Memcached
- Redis**
- Reserved Nodes
- Backups
- Parameter Groups
- Subnet Groups
- Events
- ElastiCache Cluster Client

ElastiCache Dashboard

Create Backup Reboot Delete Modify

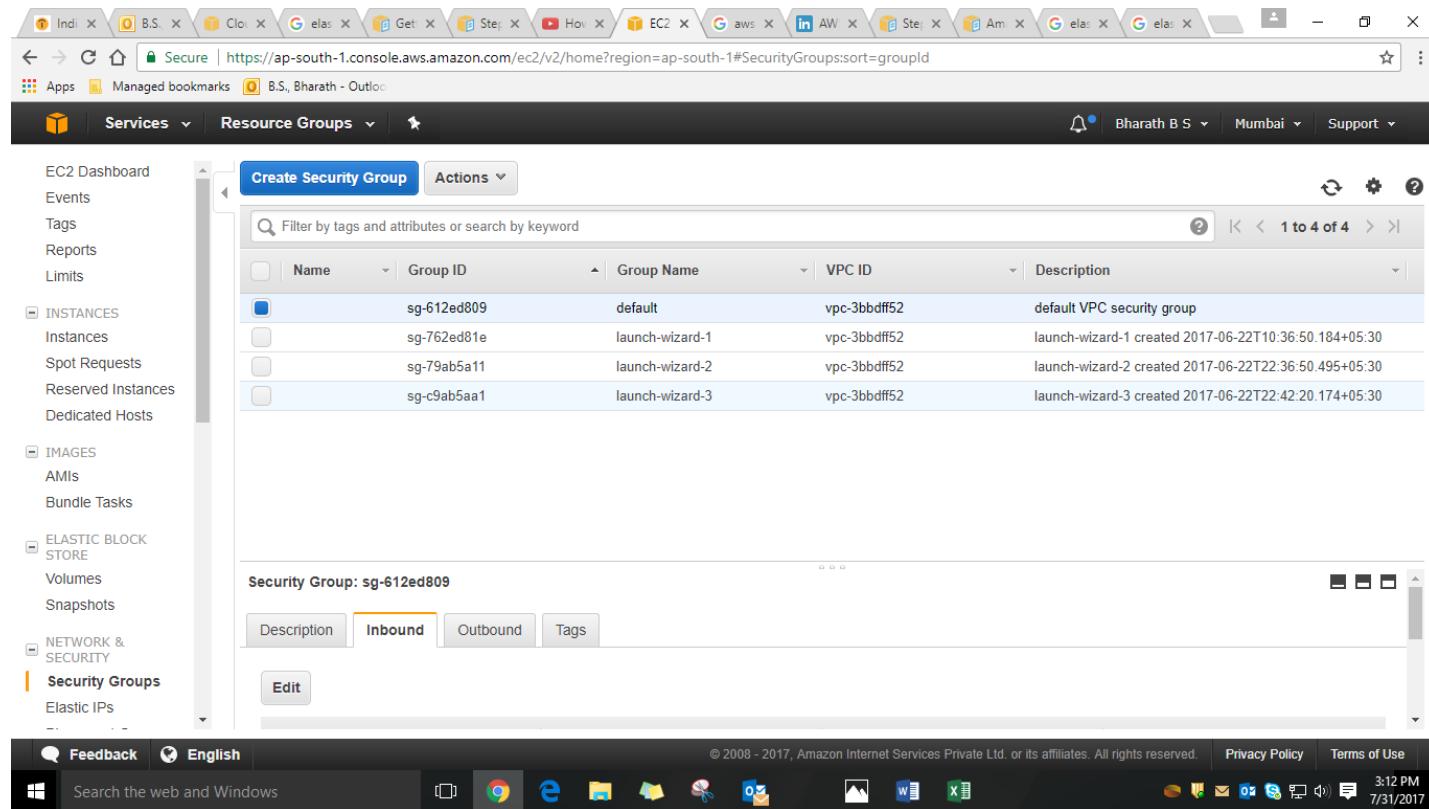
Viewing 1 of 1 Clusters

| | Cluster Name | Mode | Shards | Nodes | Node Type | Status |
|---|----------------|-------|--------|--------|----------------|-----------|
| ▶ | mycachecluster | Redis | 0 | 1 node | cache.t2.micro | available |

- Memcached
- Redis**
- Reserved Nodes
- Backups
- Parameter Groups
- Subnet Groups
- Events
- ElastiCache Cluster Client



5.3.3 Authorize Access



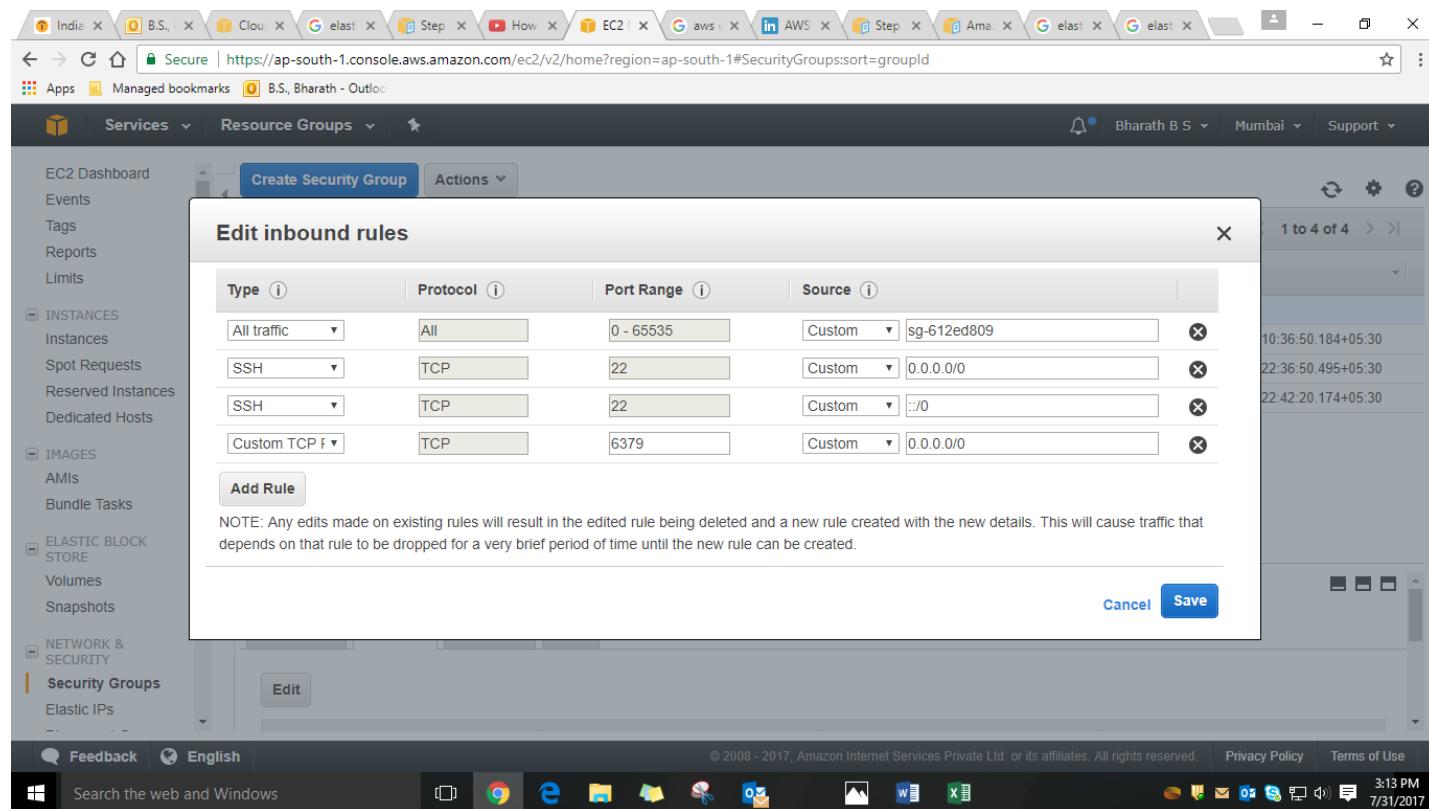
The screenshot shows the AWS Management Console interface for managing security groups. The left sidebar navigation includes: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (Instances, Spot Requests, Reserved Instances, Dedicated Hosts), IMAGES (AMIs, Bundle Tasks), ELASTIC BLOCK STORE (Volumes, Snapshots), and NETWORK & SECURITY (Security Groups, Elastic IPs). The main content area displays a table of security groups:

| Name | Group ID | Group Name | VPC ID | Description |
|-------------|----------|-----------------|--------------|---|
| sg-612ed809 | | default | vpc-3bbdff52 | default VPC security group |
| sg-762ed81e | | launch-wizard-1 | vpc-3bbdff52 | launch-wizard-1 created 2017-06-22T10:36:50.184+05:30 |
| sg-79ab5a11 | | launch-wizard-2 | vpc-3bbdff52 | launch-wizard-2 created 2017-06-22T22:36:50.495+05:30 |
| sg-c9ab5aa1 | | launch-wizard-3 | vpc-3bbdff52 | launch-wizard-3 created 2017-06-22T22:42:20.174+05:30 |

Below the table, a specific security group is selected: "Security Group: sg-612ed809". The "Inbound" tab is active, showing the current inbound rules:

- All traffic (Protocol: All, Port Range: 0 - 65535, Source: sg-612ed809)
- SSH (Protocol: TCP, Port Range: 22, Source: Custom 0.0.0.0/0)
- SSH (Protocol: TCP, Port Range: 22, Source: Custom ::/0)
- Custom TCP (Protocol: TCP, Port Range: 6379, Source: Custom 0.0.0.0/0)

A "Save" button is visible at the bottom right of the rule editor.



The screenshot shows the "Edit inbound rules" dialog box overlaid on the AWS EC2 Security Groups page. The dialog box has columns for Type, Protocol, Port Range, and Source. It lists four rules:

| Type | Protocol | Port Range | Source |
|-------------|----------|------------|--------------------|
| All traffic | All | 0 - 65535 | Custom sg-612ed809 |
| SSH | TCP | 22 | Custom 0.0.0.0/0 |
| SSH | TCP | 22 | Custom ::/0 |
| Custom TCP | TCP | 6379 | Custom 0.0.0.0/0 |

At the bottom of the dialog box, there is a note: "NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created." Below the note are "Cancel" and "Save" buttons.

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.

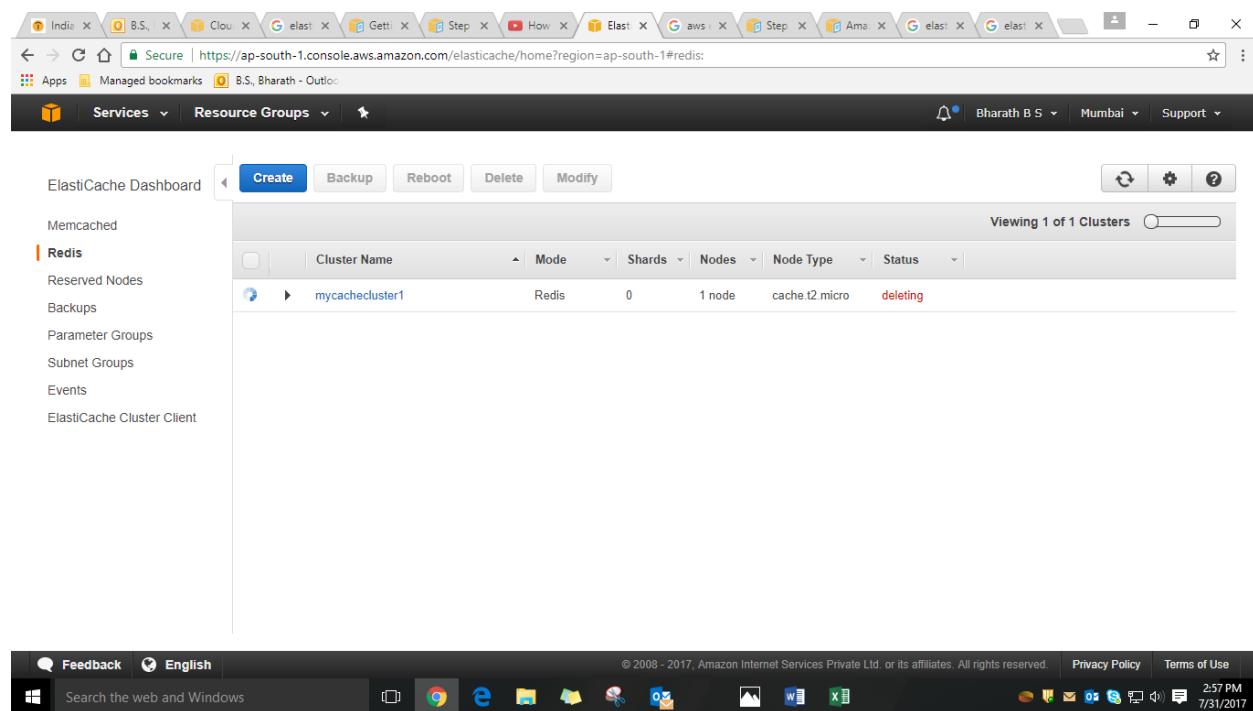
You are now connected to a node, and you can run Redis commands. The following is an example.

```

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]



The screenshot shows the AWS ElastiCache Dashboard. On the left sidebar, under the Redis section, there are links for Memcached, Reserved Nodes, Backups, Parameter Groups, Subnet Groups, Events, and ElastiCache Cluster Client. The main area displays a table with one cluster entry:

| | Cluster Name | Mode | Shards | Nodes | Node Type | Status |
|--|-----------------|-------|--------|--------|----------------|----------|
| | mycachecluster1 | Redis | 0 | 1 node | cache.t2.micro | deleting |

At the bottom of the dashboard, there is a footer with links for Feedback, English, Privacy Policy, Terms of Use, and a search bar. The search bar contains the text "Search the web and Windows".

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

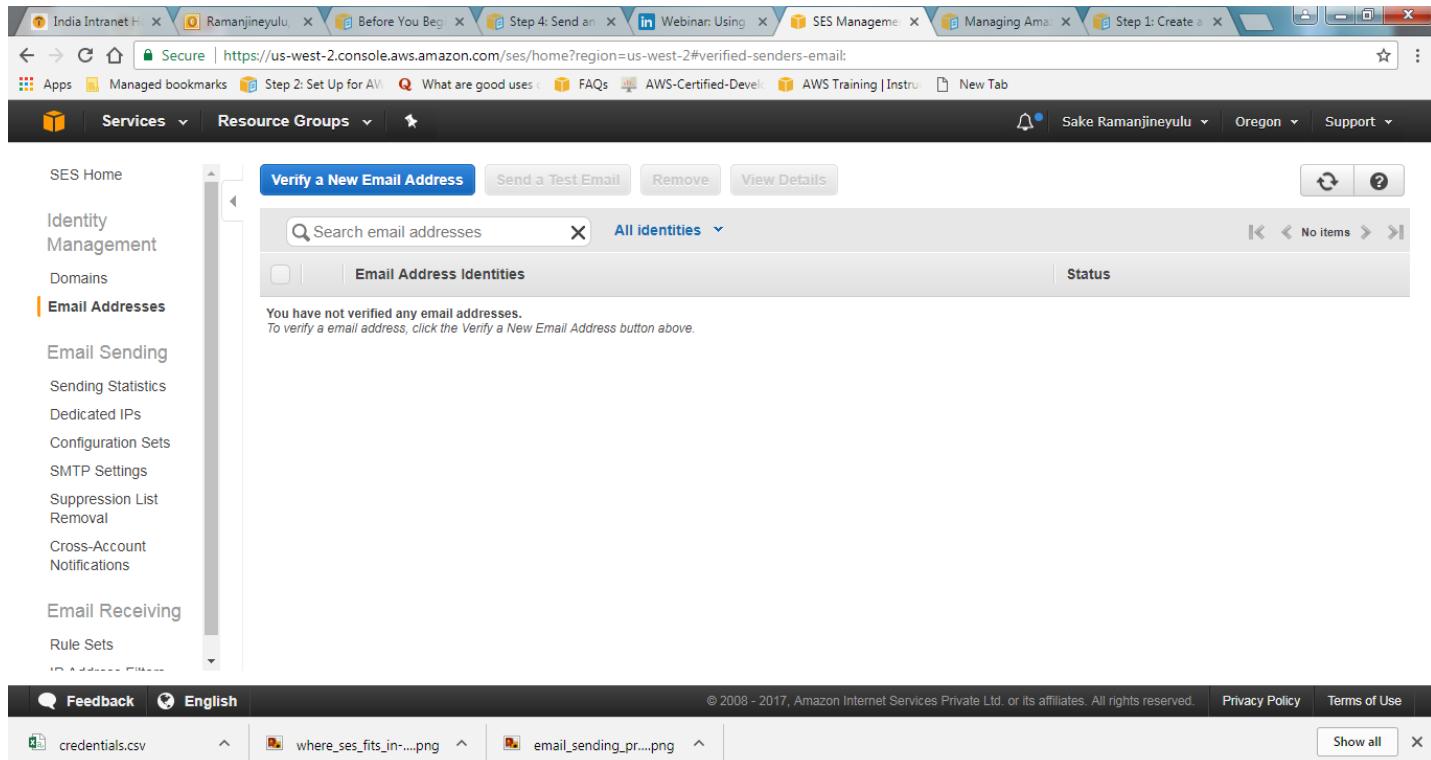
- Minimal setup, maximum deliverability send a lot of email as often necessary. All aspects of email management built in Focus on your business.
- You can send email in minutes and stay informed with on-demand feedback.
- This service is Available in three regions only. Those are EU (Ireland) , US East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

6.3 Procedure

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

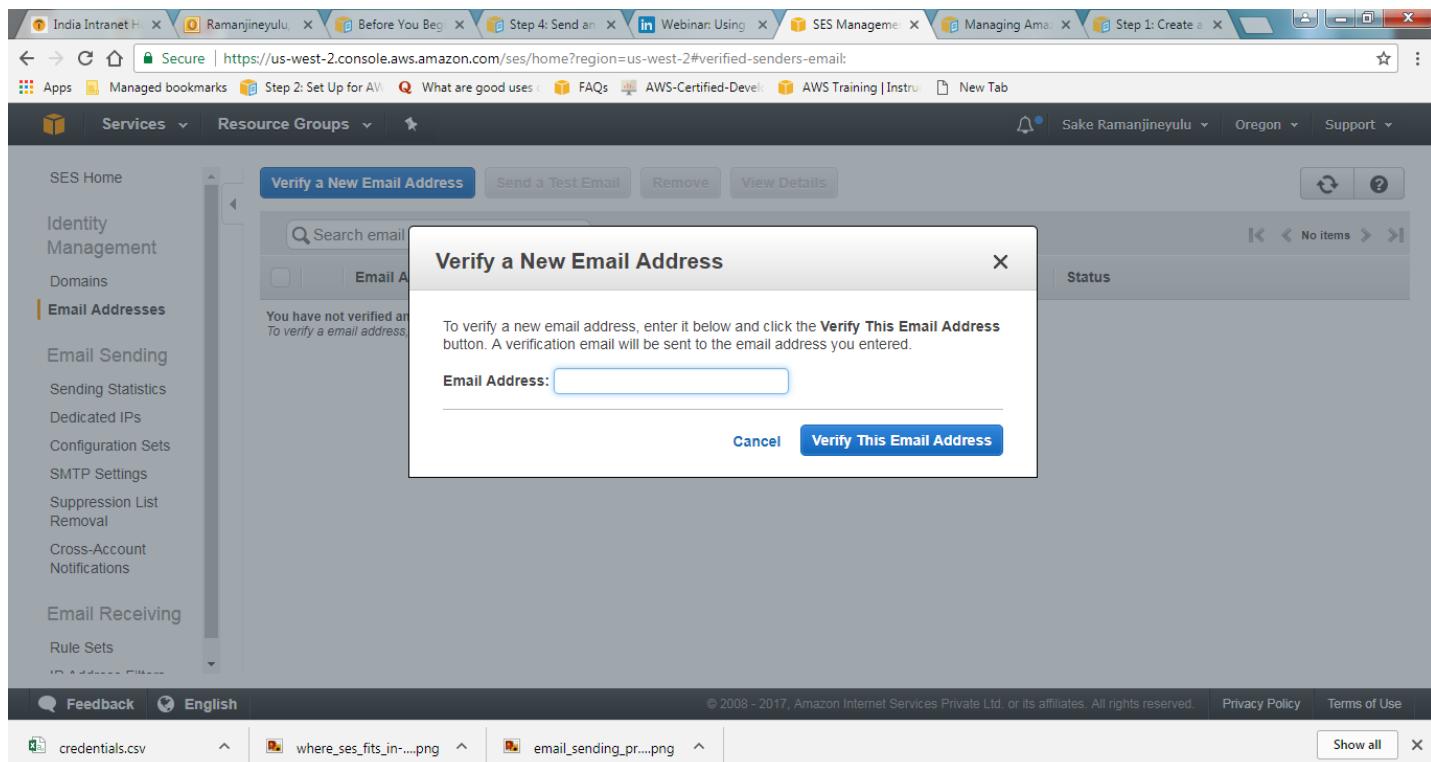
Go to the AWS console and click on SES and follow the screen shots to complete the overall SES set up.

6.3.1 Verifying Email Address



The screenshot shows the AWS SES Management Console interface. On the left, a sidebar lists various service categories. Under 'Email Addresses', the 'Verify a New Email Address' button is highlighted. A modal window titled 'Verify a New Email Address' is displayed in the center, containing a search bar and a text input field labeled 'Email Address:' with a placeholder 'Enter email address'. Below the input field are 'Cancel' and 'Verify This Email Address' buttons.

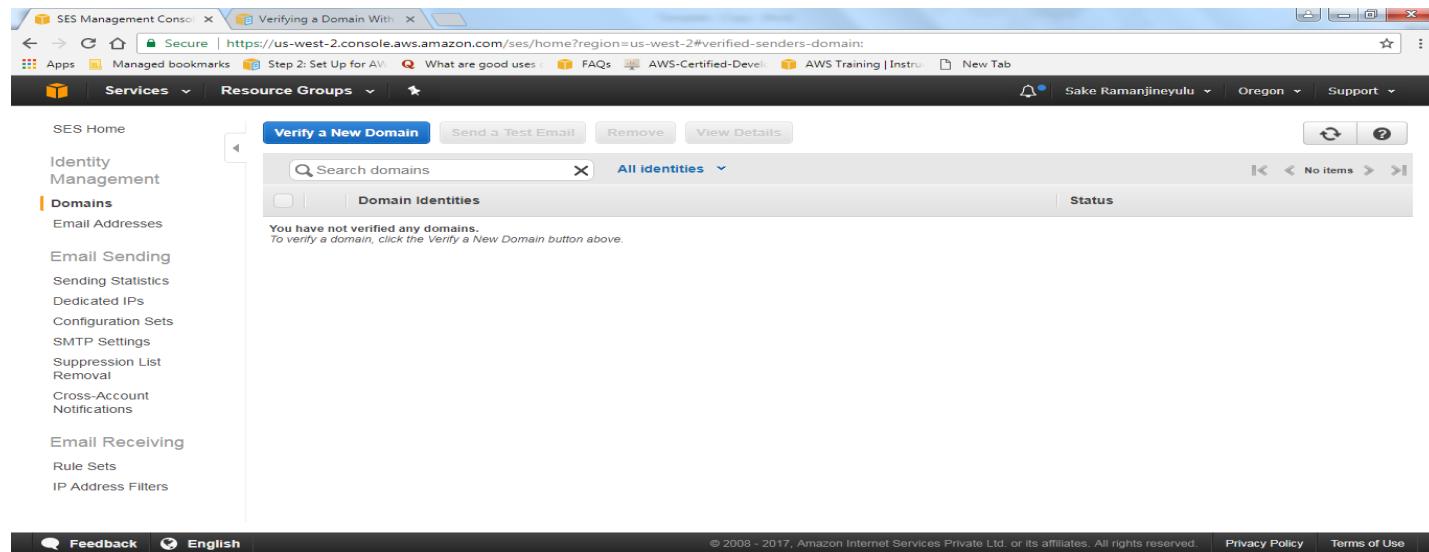
Click on Verify new Email Address.



The screenshot shows the AWS SES Management Console interface. On the left, a sidebar lists various service categories. Under 'Email Addresses', the 'Verify a New Email Address' button is highlighted. A modal window titled 'Verify a New Email Address' is displayed in the center, containing a search bar and a text input field labeled 'Email Address:' with a placeholder 'Enter email address'. Below the input field are 'Cancel' and 'Verify This Email Address' buttons.

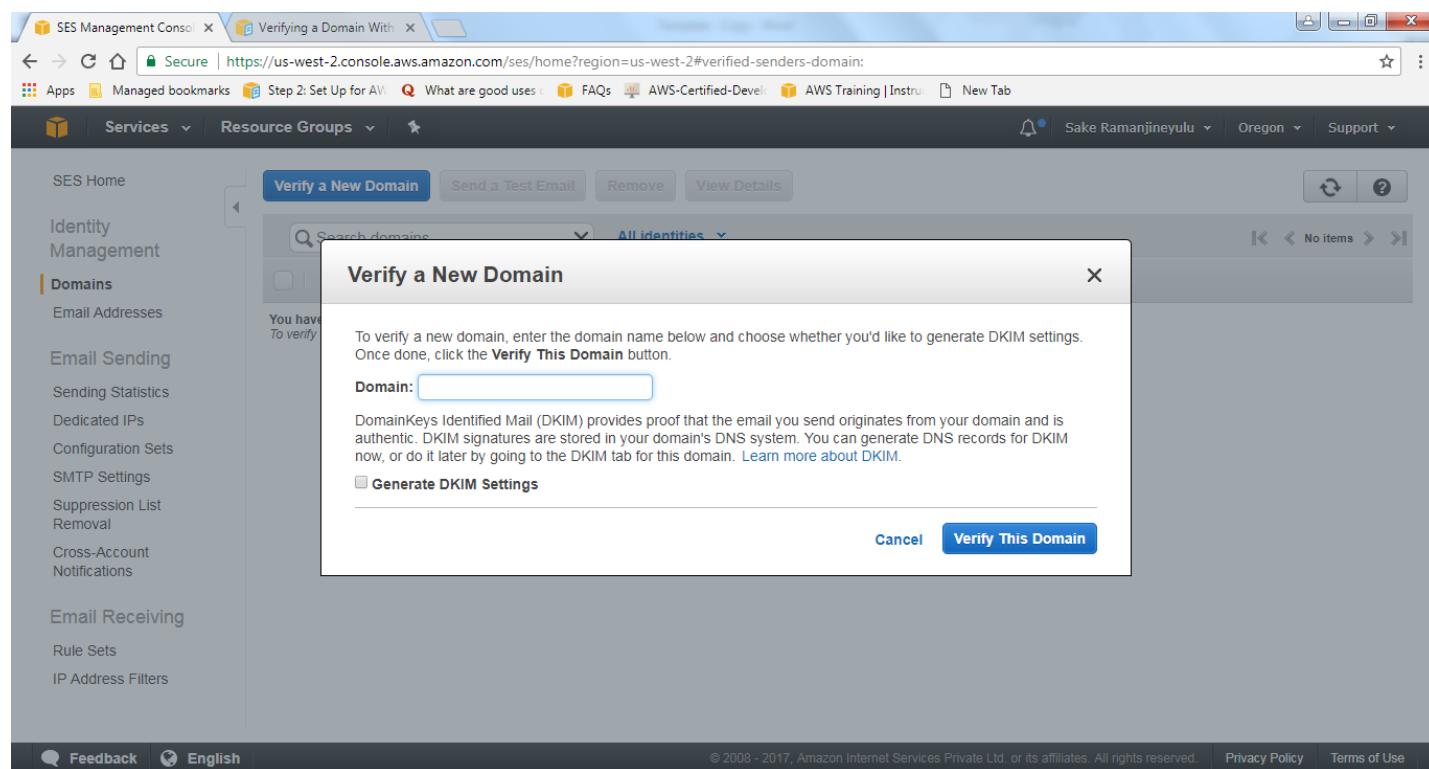
Enter your Email Address and click on Verify this Email Address. After this Your Email address receives a Confirmation mail link and click on that on that link Verify your email address.

6.3.2 Verifying Domain Address



The screenshot shows the AWS SES Management Console. The left sidebar is titled 'Domains' and lists various options like 'Email Addresses', 'Email Sending', and 'Email Receiving'. The main area has a search bar and a table with one row: 'You have not verified any domains. To verify a domain, click the Verify a New Domain button above.' A large blue button labeled 'Verify a New Domain' is visible at the top of this section.

Click on Verify a new domain

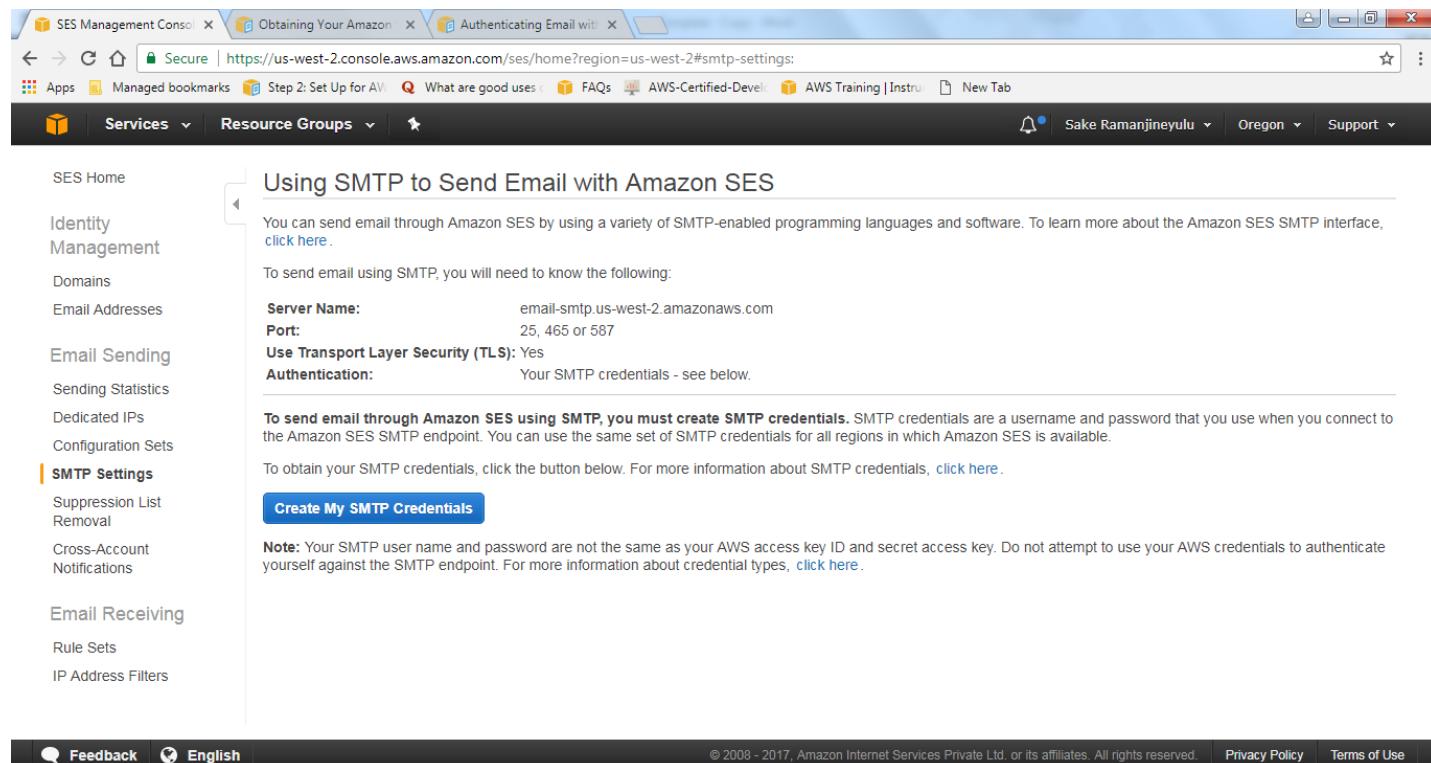


The screenshot shows the 'Verify a New Domain' dialog box. It contains instructions: 'To verify a new domain, enter the domain name below and choose whether you'd like to generate DKIM settings. Once done, click the Verify This Domain button.' Below this is a 'Domain:' input field, a 'Generate DKIM Settings' checkbox, and two buttons: 'Cancel' and 'Verify This 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



Using SMTP to Send Email with Amazon SES

You can send email through Amazon SES by using a variety of SMTP-enabled programming languages and software. To learn more about the Amazon SES SMTP interface, [click here](#).

To send email using SMTP, you will need to know the following:

| | |
|--|------------------------------------|
| Server Name: | email-smtp.us-west-2.amazonaws.com |
| Port: | 25, 465 or 587 |
| Use Transport Layer Security (TLS): | Yes |
| Authentication: | Your SMTP credentials - see below. |

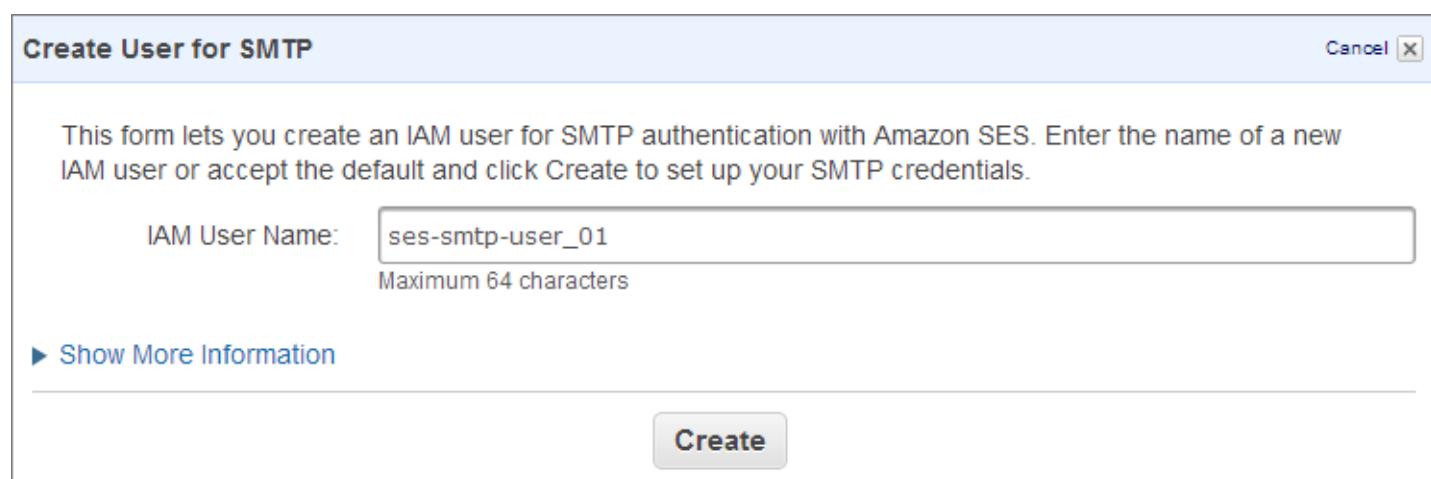
To send email through Amazon SES using SMTP, you must create SMTP credentials. SMTP credentials are a username and password that you use when you connect to the Amazon SES SMTP endpoint. You can use the same set of SMTP credentials for all regions in which Amazon SES is available.

To obtain your SMTP credentials, click the button below. For more information about SMTP credentials, [click here](#).

Create My SMTP Credentials

Note: Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credential types, [click here](#).

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



Create User for SMTP

This form lets you create an IAM user for SMTP authentication with Amazon SES. Enter the name of a new IAM user or accept the default and click Create to set up your SMTP credentials.

IAM User Name: Maximum 64 characters

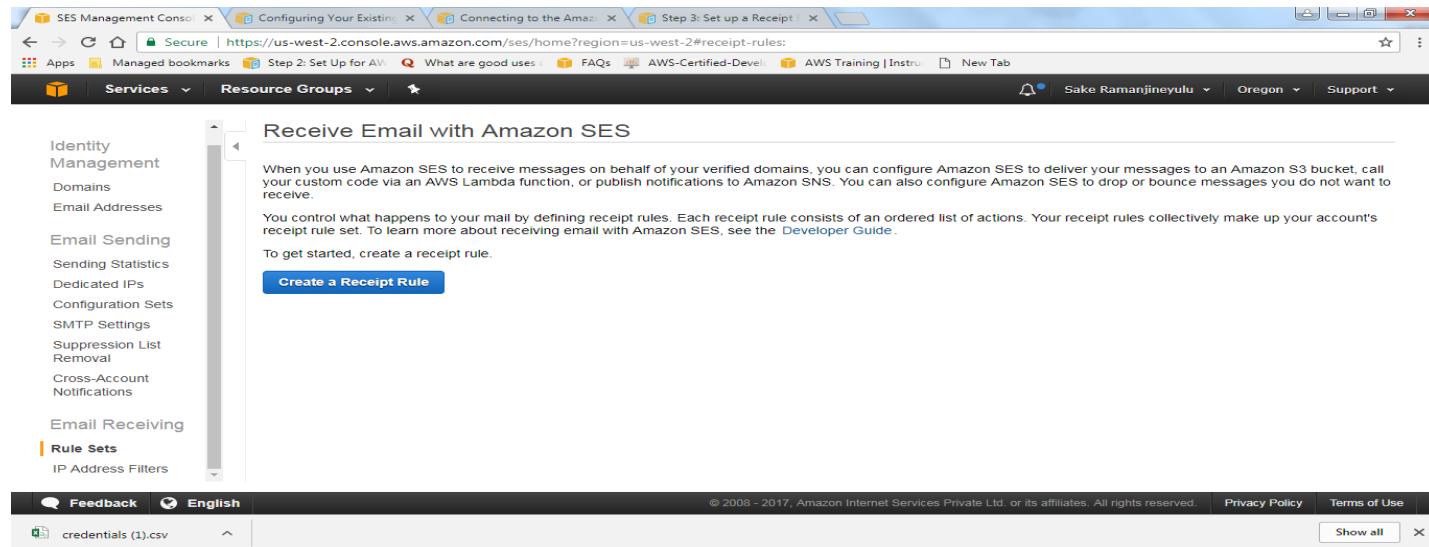
▶ Show More Information

Create

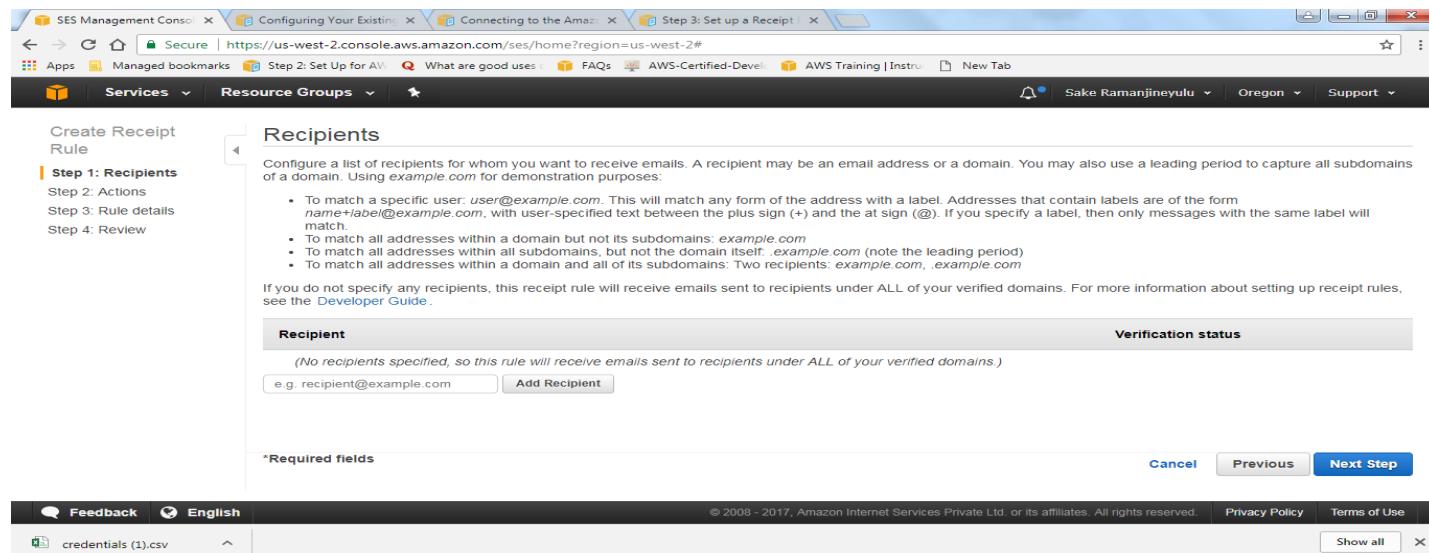
Click on Create and Download Your SMTP credentials

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

| | |
|-------------------|-------------------------------|
| S3 bucket* | <None> |
| Object key prefix | <None> Enter a bucket name |
| Encrypt Message | |
| SNS topic | 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

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* ses-email-receiving-tutorial
Object key prefix
Encrypt Message
SNS topic <None>

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

AWS Services Edit AWS User N. Virginia Support

Create Receipt Rule
Step 1: Recipients
Step 2: Actions
Step 3: Rule details
Step 4: Review

Step 4: Review

Please review the following receipt rule settings before creating the rule.

Recipients [Edit](#)
(No recipients specified, so this rule will receive emails sent to recipients under ALL of your verified domains.)

Actions [Edit](#)
1. S3 Action Write to S3 bucket ses-email-receiving-tutorial

Rule Details [Edit](#)

| | |
|--------------------------------|------------------|
| Rule name | my-rule |
| Enabled | true |
| Require TLS | false |
| Enable spam and virus scanning | true |
| Rule set | default-rule-set |
| Insert after rule | <Beginning> |

*Required fields

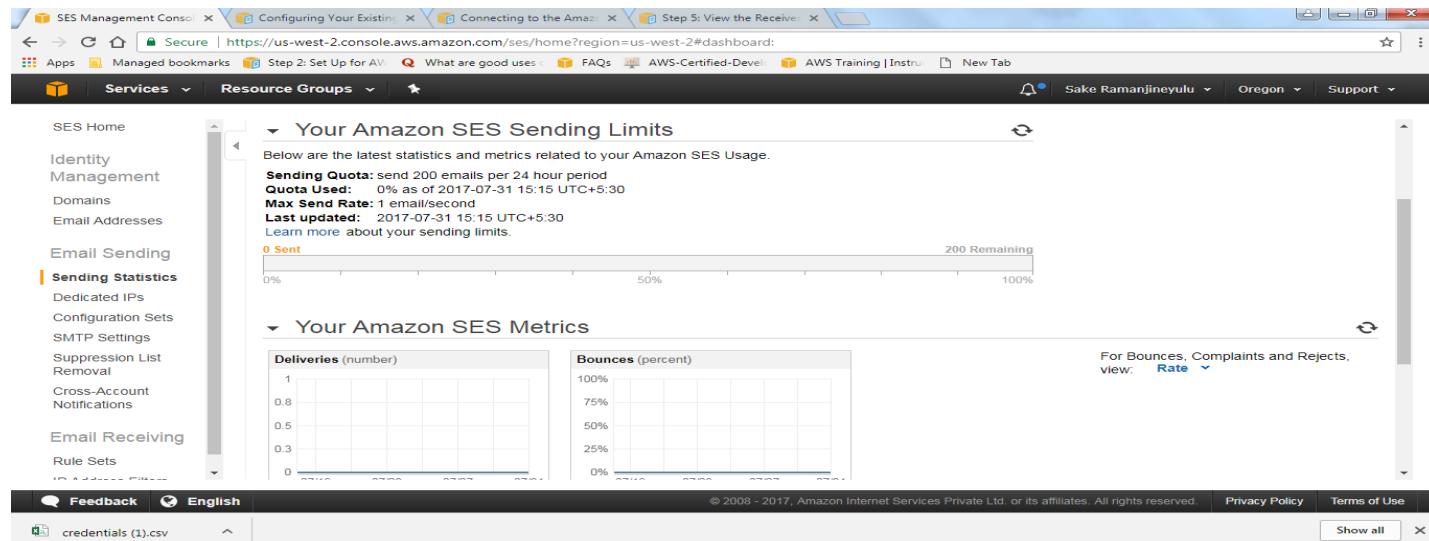
Cancel Previous **Create Rule**

Feedback English © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

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

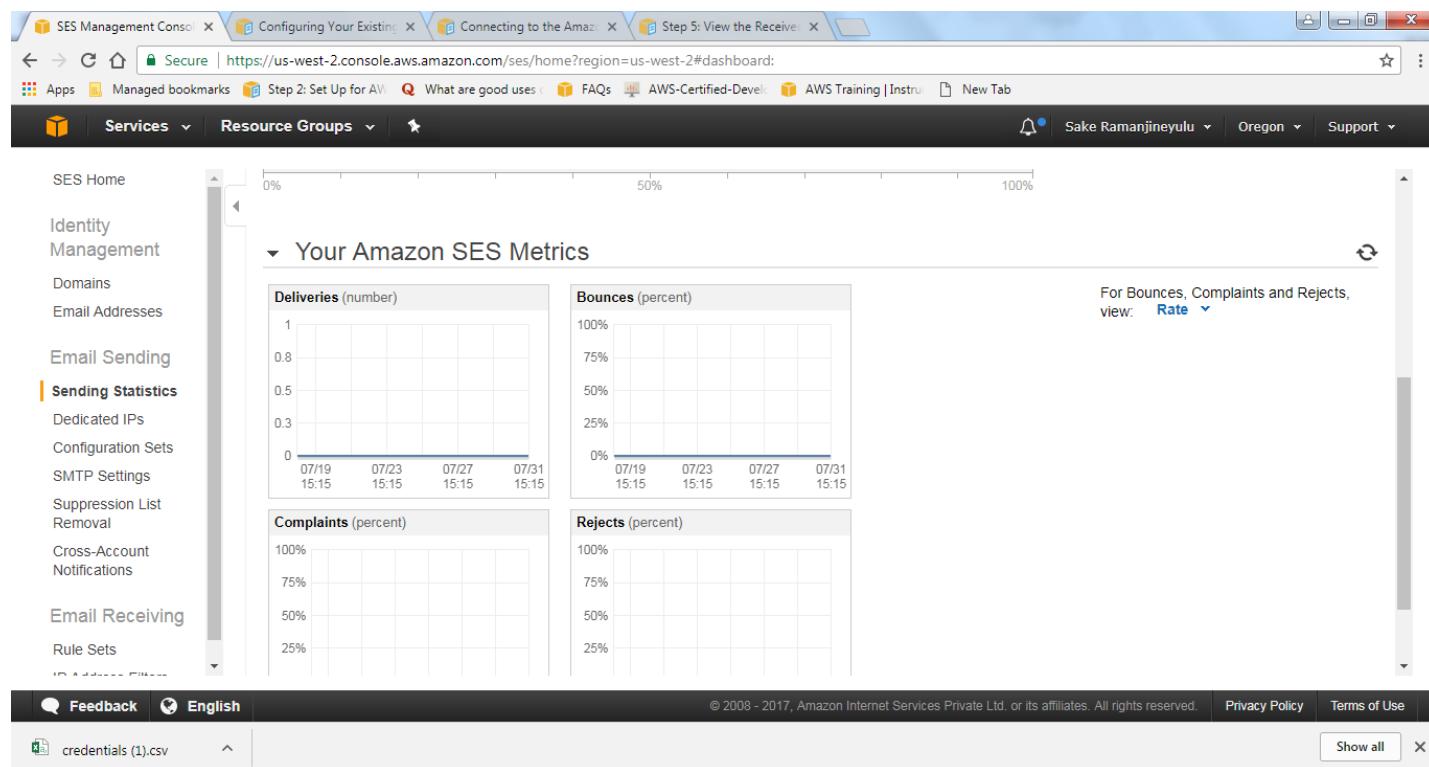
By using AWS console: In the navigation pane click on Sending Statistics it will show like this



The screenshot shows the SES Management Console dashboard for the 'Your Amazon SES Sending Limits' section. On the left, the navigation pane is visible with various options like SES Home, Identity Management, Domains, Email Addresses, Email Sending, and Email Receiving. The 'Sending Statistics' option is currently selected. The main content area displays the latest statistics and metrics related to Amazon SES usage. It includes a summary table with the following details:

| |
|--|
| Sending Quota: send 200 emails per 24 hour period |
| Quota Used: 0% as of 2017-07-31 15:15 UTC+5:30 |
| Max Send Rate: 1 email/second |
| Last updated: 2017-07-31 15:15 UTC+5:30 |

Below this, there is a progress bar showing '0 Sent' and '200 Remaining' at 100%. To the right, there is a chart titled 'Deliveries (number)' showing a single data point of 1. Further down, there is another chart titled 'Bounces (percent)' showing a single data point of 0%. A note indicates that for Bounces, Complaints and Rejects, view the 'Rate' metric.



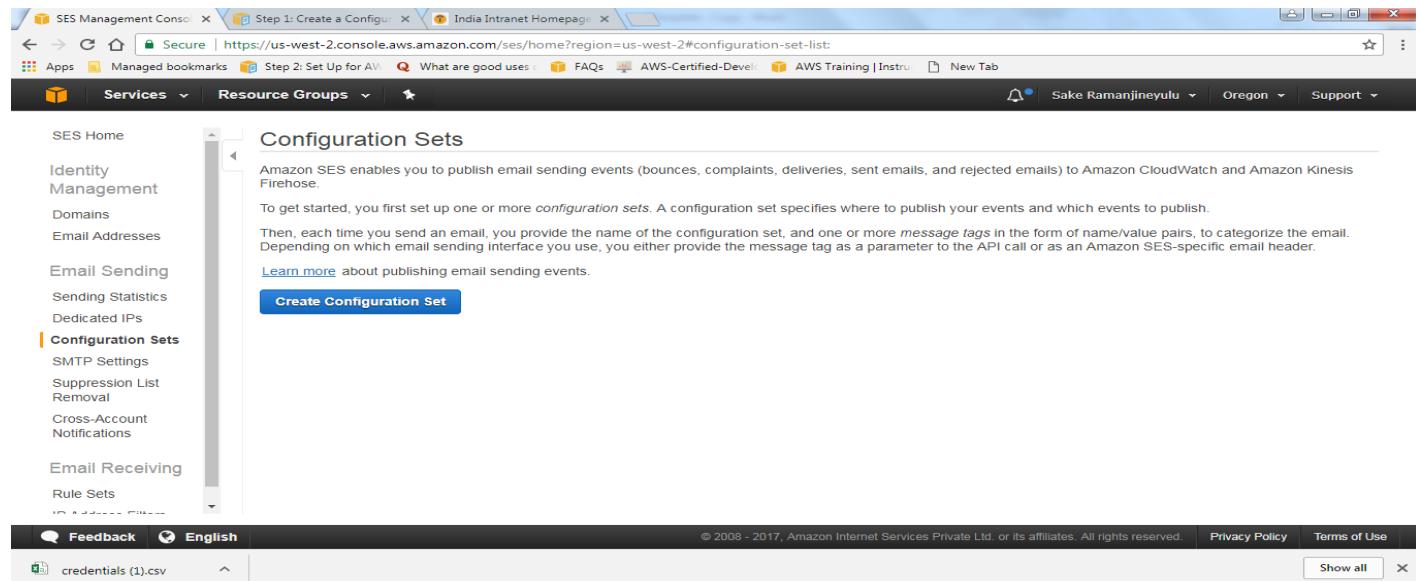
The screenshot shows the SES Management Console dashboard for the 'Your Amazon SES Metrics' section. The navigation pane is identical to the previous screenshot. The main content area displays four line charts showing metrics over time from July 19 to July 31, 2017, at 15:15 UTC+5:30. The charts are:

- Deliveries (number):** Shows a single data point of 1.
- Bounces (percent):** Shows a single data point of 0%.
- Complaints (percent):** Shows a single data point of 0%.
- Rejects (percent):** Shows a single data point of 0%.

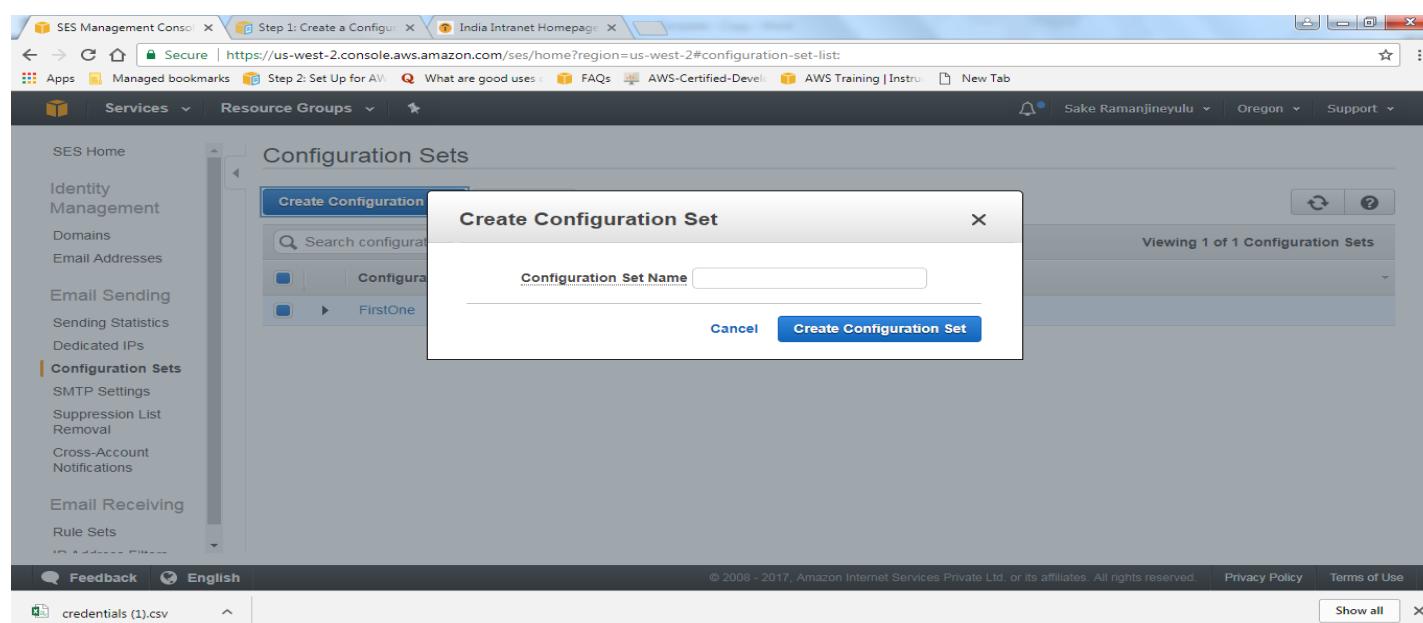
A note at the bottom right of the dashboard indicates that for Bounces, Complaints and Rejects, view the 'Rate' metric.

By using Event Publishing:

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



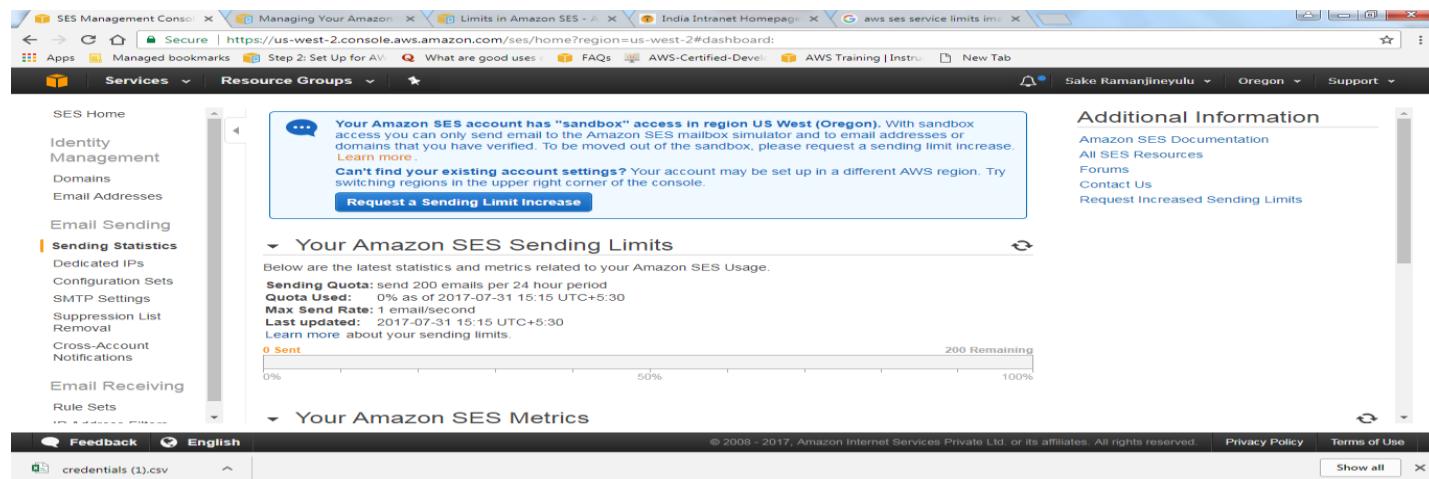
The screenshot shows the AWS SES Management Console. The left sidebar has a 'Configuration Sets' section selected. A prominent blue button labeled 'Create Configuration Set' is centered on the page. The main content area contains descriptive text about configuration sets and their purpose.



The screenshot shows the 'Create Configuration Set' dialog box overlaid on the main SES Management Console interface. The dialog box has a search bar at the top and a single input field labeled 'Configuration Set Name'. Below the input field are two buttons: 'Cancel' and 'Create Configuration Set'.

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

6.3.6 AWS SES Limits



The screenshot shows the AWS SES Management Console. On the left, there's a sidebar with options like SES Home, Identity Management, Domains, Email Addresses, Email Sending (with a 'Request a Sending Limit Increase' button), and Email Receiving. The main content area displays 'Your Amazon SES Sending Limits' with a progress bar showing 0% sent and 200 remaining out of a quota of 200 emails per 24-hour period. It also shows 'Your Amazon SES Metrics'. On the right, there's an 'Additional Information' section with links to documentation, forums, and support.

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.

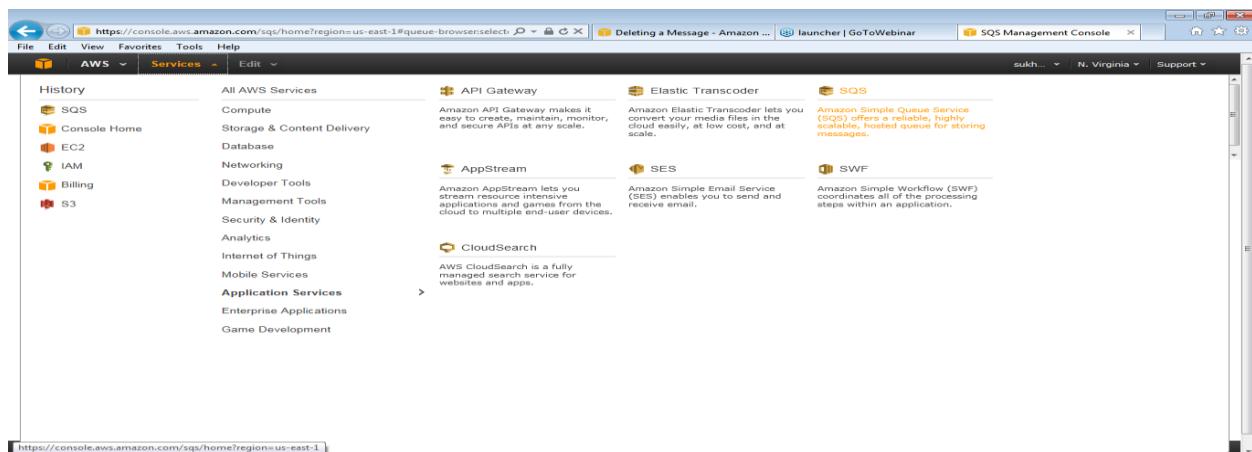
7.1 Objective

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

7.2 Procedure

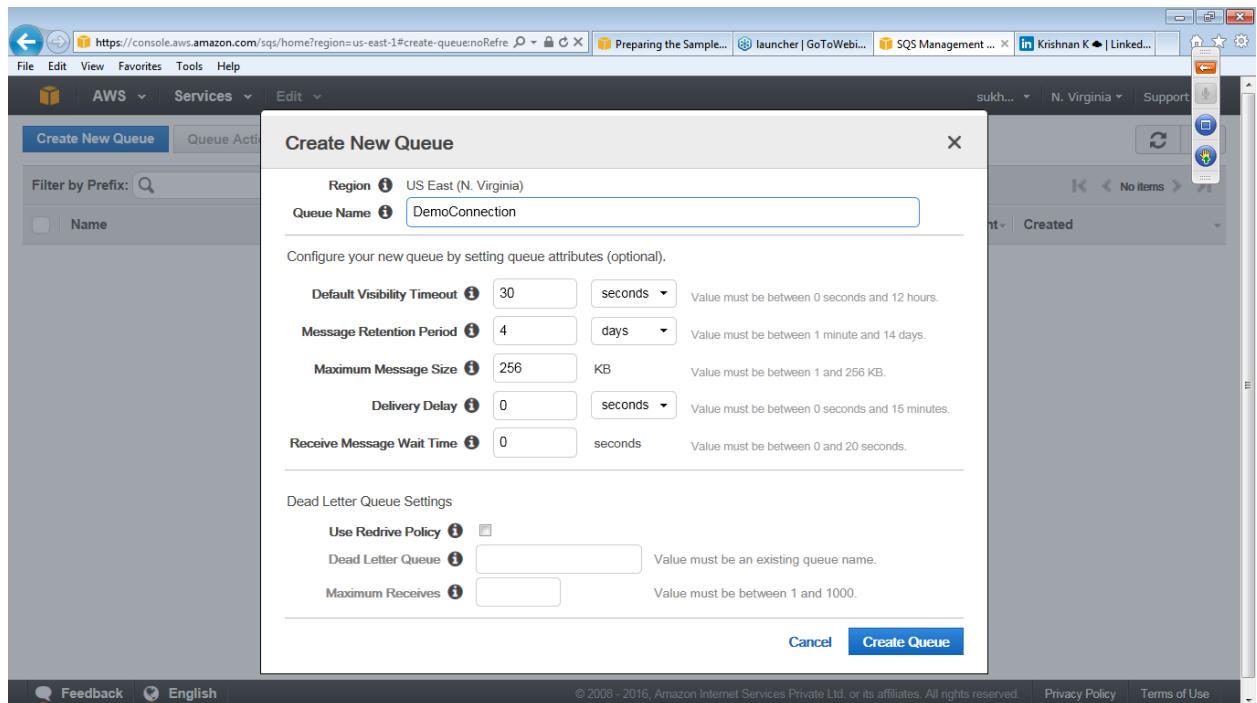
7.2.1 Create New Queue

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

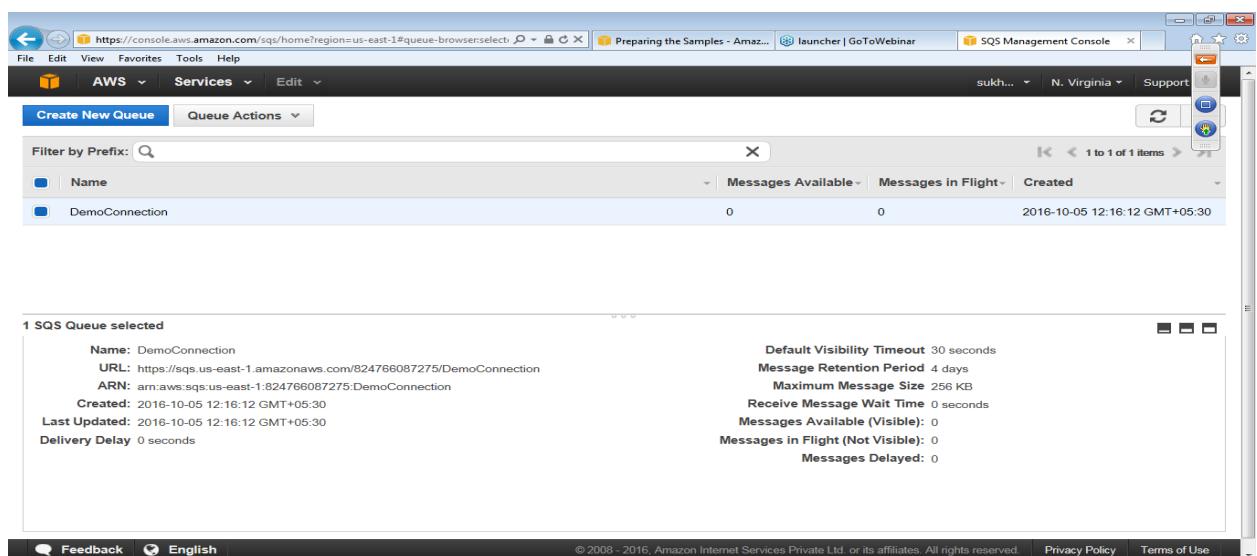


The screenshot shows the AWS Management Console with the SQS service selected under 'Application Services'. Other services listed include API Gateway, Elastic Transcoder, SNS, SES, SWF, CloudSearch, AppStream, and others. The browser address bar shows the URL for the SQS home page.

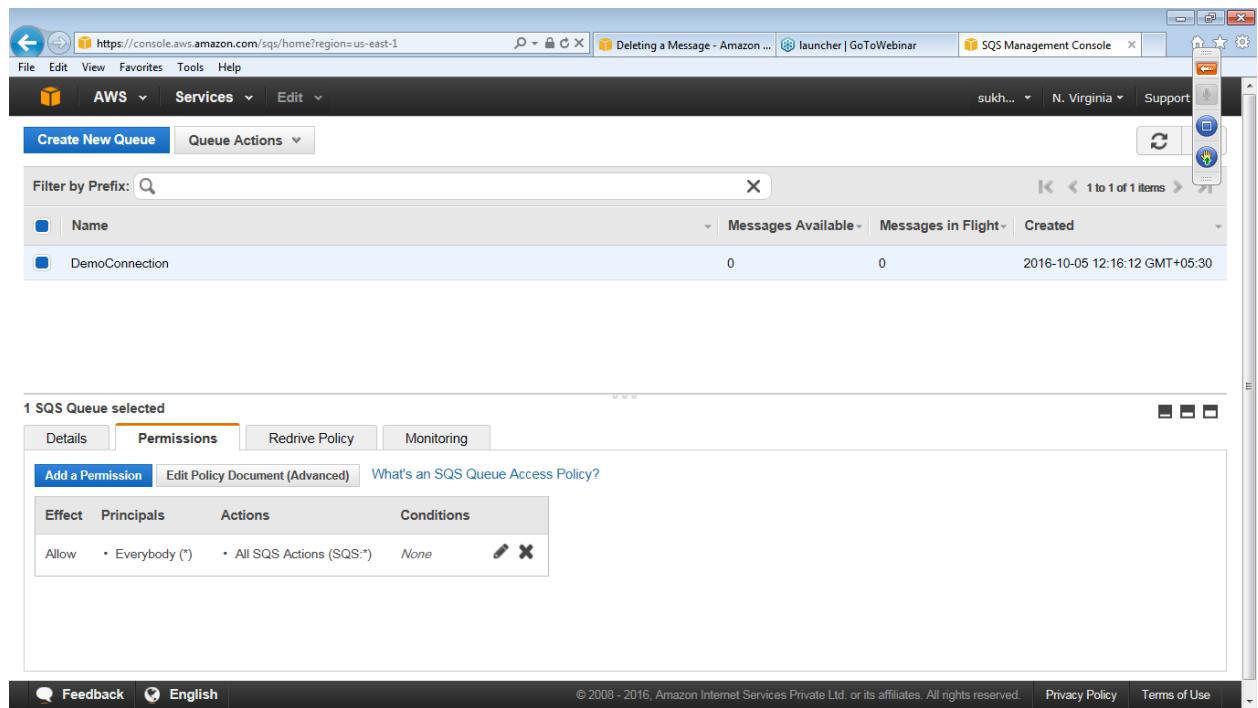
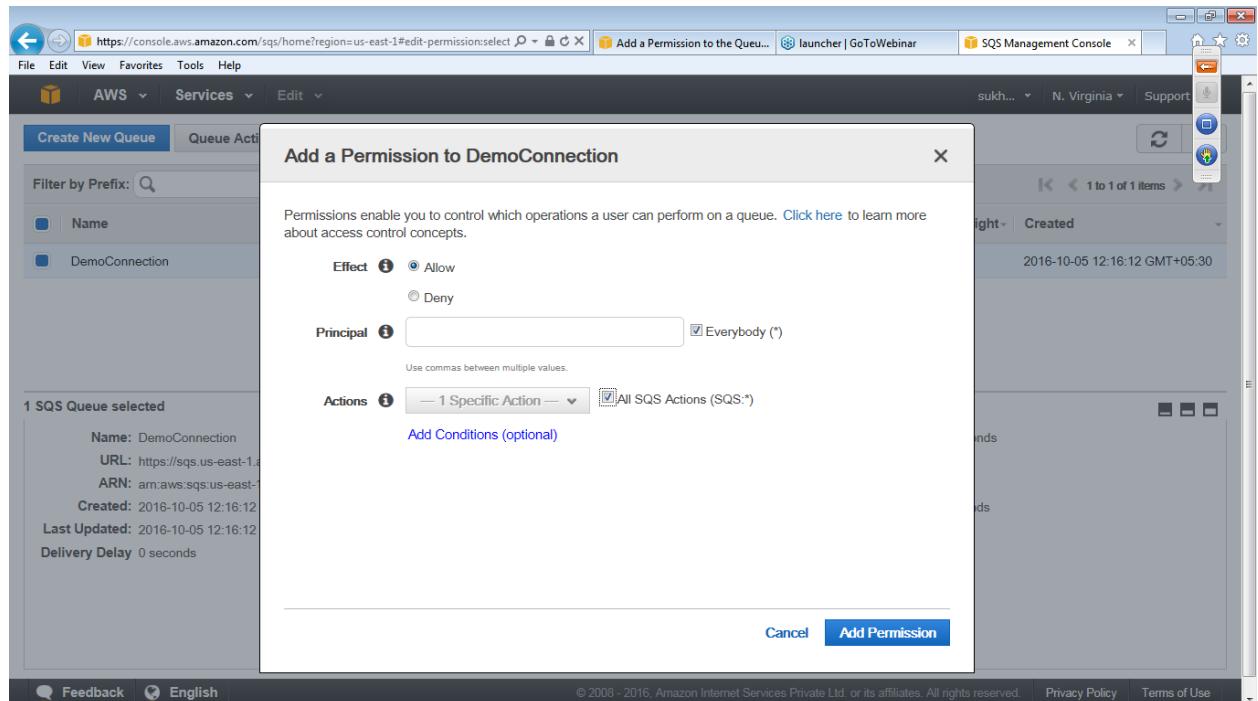
- 2 In the **Create New Queue** dialog box, enter DemoConnection in the Queue Name field, and leave the default value settings for the remaining fields



The console displays all of your queues in that region

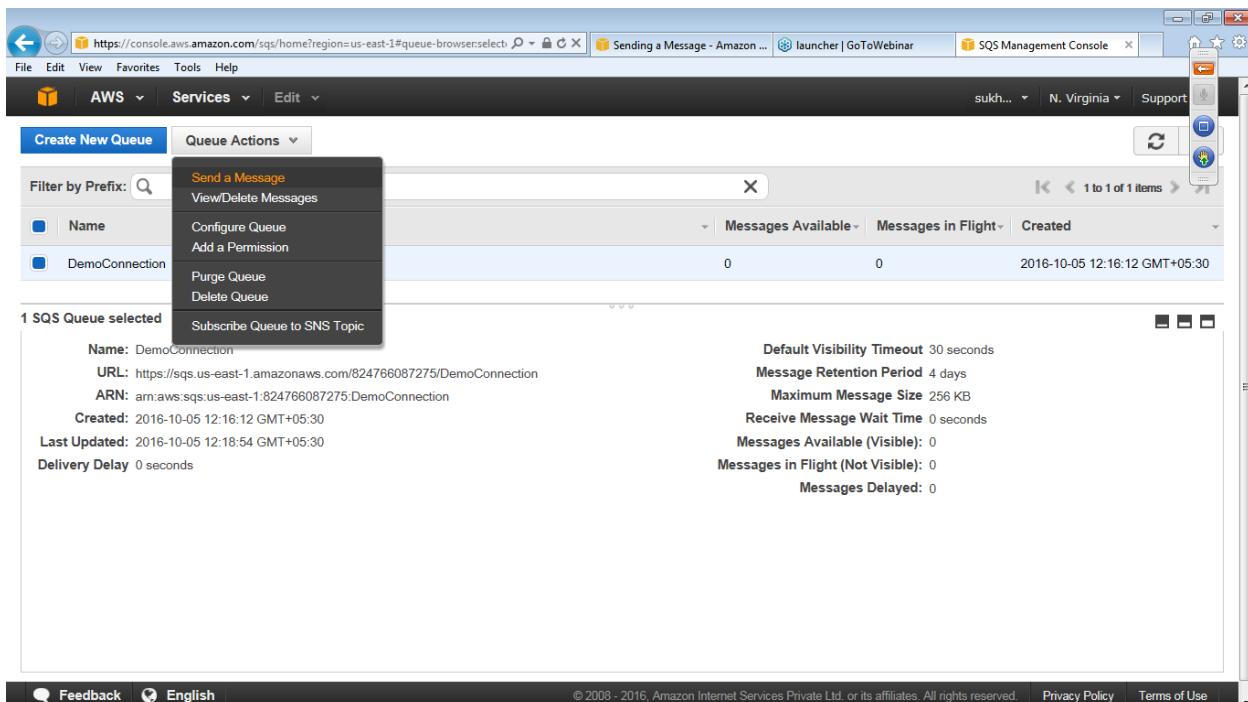


- 3 Select **Add a Permission** from the **Queue Actions** drop-down list and specify the permission's settings.



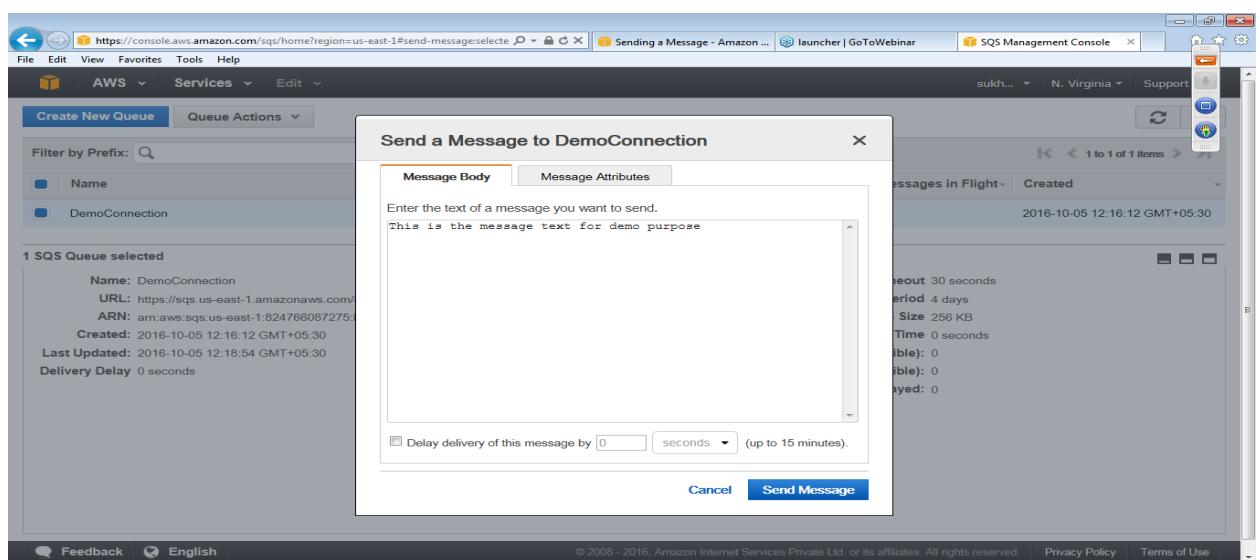
7.2.2 Sending Message

- 1 In the AWS Management Console select a queue



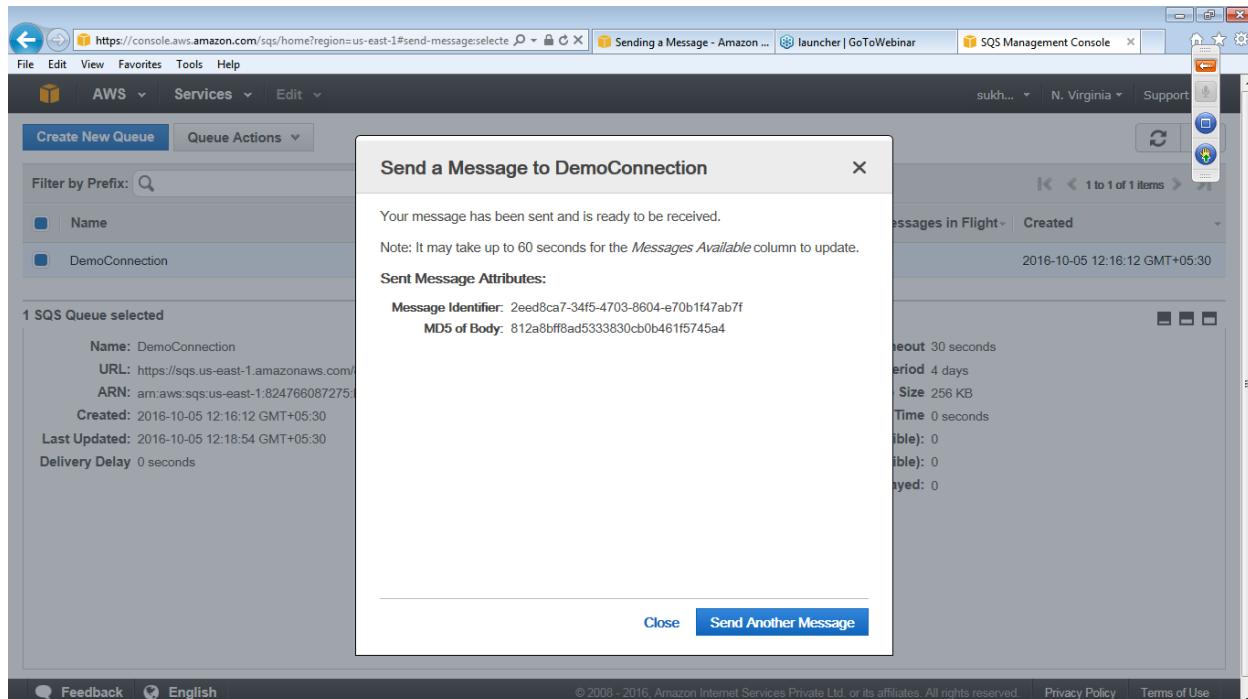
The screenshot shows the AWS Management Console interface for the SQS service. A context menu is open over the 'DemoConnection' queue, with the 'Send a Message' option highlighted in orange. The main pane displays details about the queue, including its name, URL, ARN, and creation and last update times. On the right, there are statistics for message visibility and retention.

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



The screenshot shows the 'Send a Message to DemoConnection' dialog box. The 'Message Body' tab is selected, and the text 'This is the message text for demo purpose' is entered into the message body field. Below the message body, there is a checkbox for 'Delay delivery of this message by [] seconds (up to 15 minutes)'. At the bottom of the dialog are 'Cancel' and 'Send Message' buttons.

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



Send a Message to DemoConnection

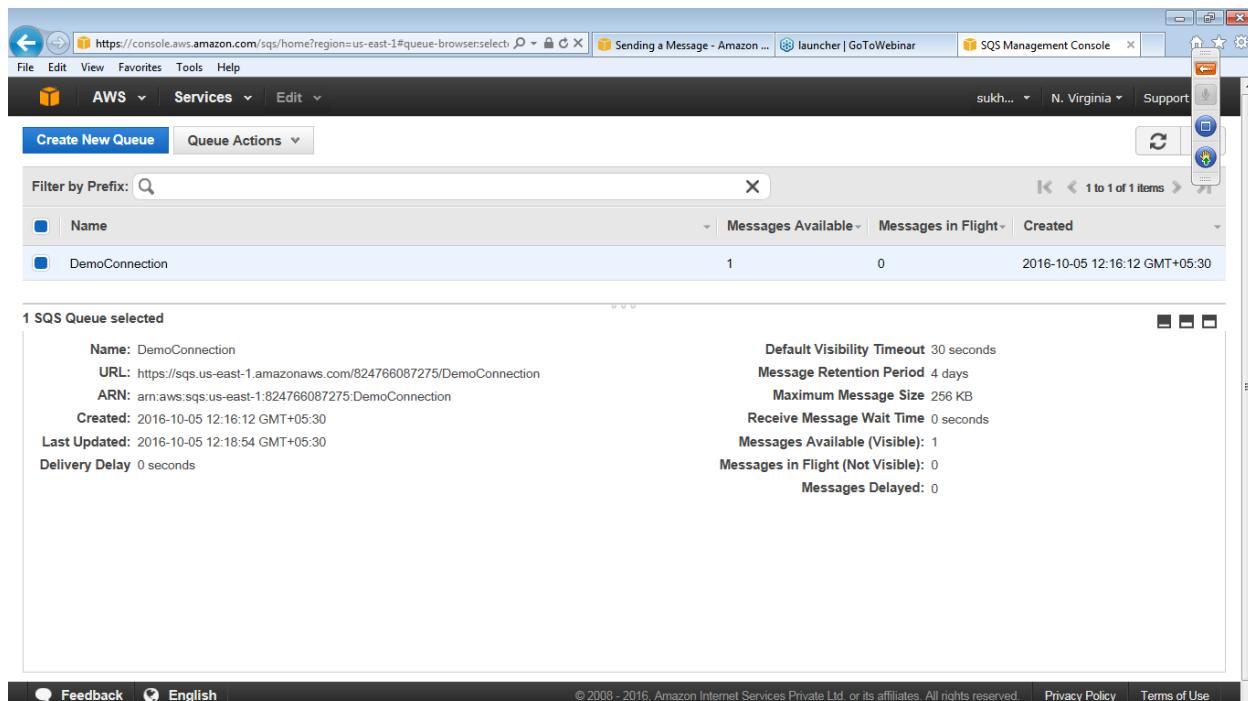
Your message has been sent and is ready to be received.

Note: It may take up to 60 seconds for the *Messages Available* column to update.

Sent Message Attributes:

- Message Identifier: 2eed8ca7-34f5-4703-8604-e70b1f47ab7f
- MD5 of Body: 812a8ff8ad5333830cb0b461f5745a4

Close **Send Another Message**



Filter by Prefix:

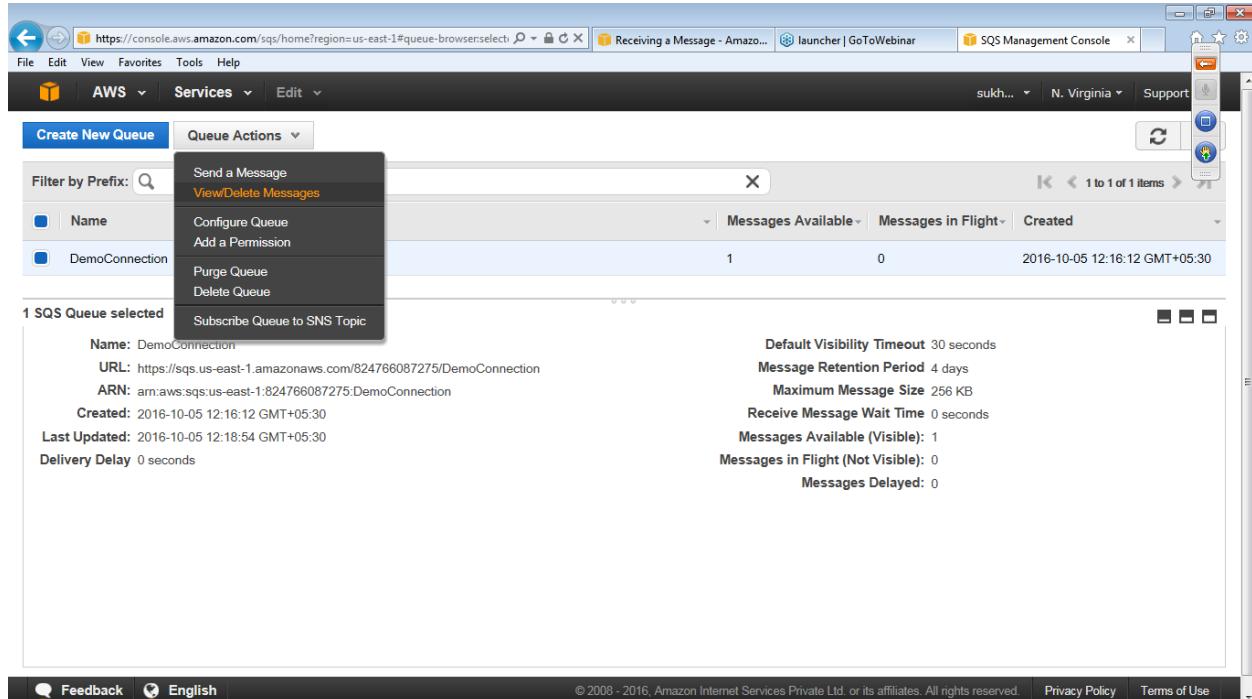
| | Messages Available | Messages in Flight | Created |
|----------------|--------------------|--------------------|-------------------------------|
| DemoConnection | 1 | 0 | 2016-10-05 12:16:12 GMT+05:30 |

1 SQS Queue selected

Name: DemoConnection **Default Visibility Timeout:** 30 seconds
URL: https://sqs.us-east-1.amazonaws.com/824766087275/DemoConnection **Message Retention Period:** 4 days
ARN: arn:aws:sqs:us-east-1:824766087275:DemoConnection **Maximum Message Size:** 256 KB
Created: 2016-10-05 12:16:12 GMT+05:30 **Receive Message Wait Time:** 0 seconds
Last Updated: 2016-10-05 12:18:54 GMT+05:30 **Messages Available (Visible):** 1
Delivery Delay: 0 seconds **Messages in Flight (Not Visible):** 0
Delayed: 0

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



The screenshot shows the AWS Management Console interface for the SQS Management Console. A context menu is open over a selected queue named 'DemoConnection'. The menu items include:

- Send a Message
- View/Delete Messages** (highlighted in yellow)
- Configure Queue
- Add a Permission
- Purge Queue
- Delete Queue

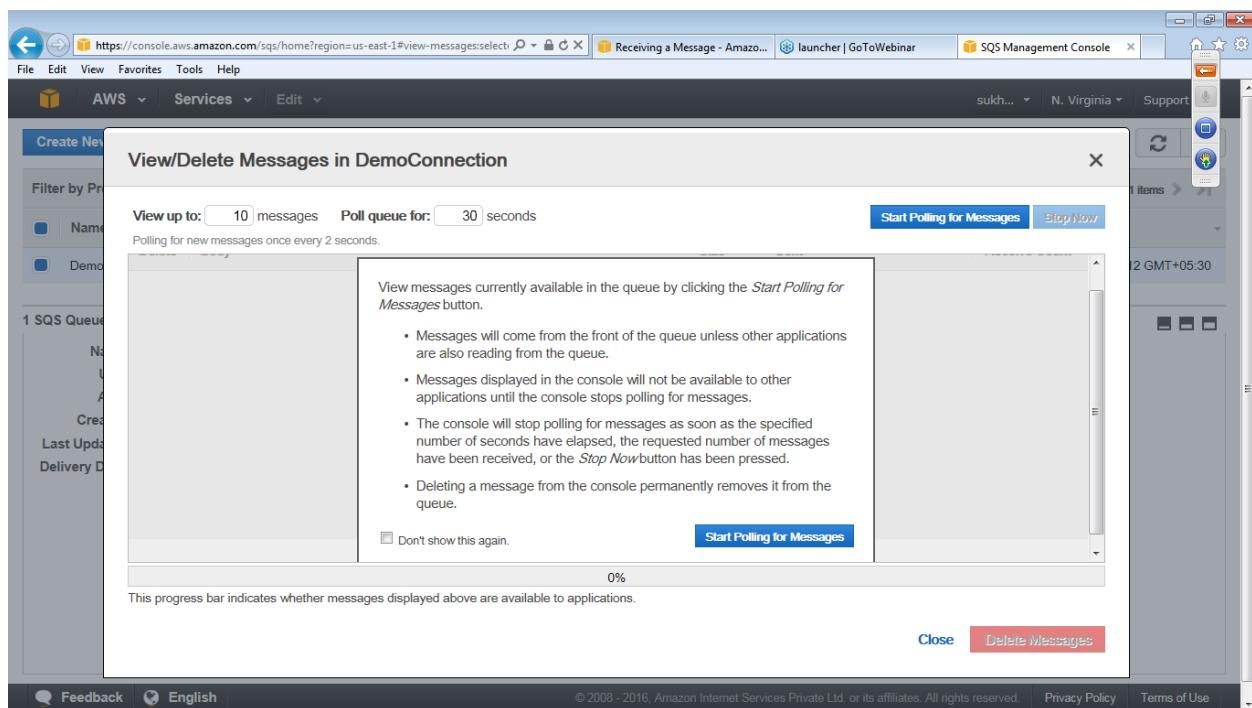
Below the menu, the queue details are displayed:

- 1 SQS Queue selected**
- Name:** DemoConnection
- URL:** https://sns.us-east-1.amazonaws.com/824766087275/DemoConnection
- ARN:** arn:aws:sns:us-east-1:824766087275:DemoConnection
- Created:** 2016-10-05 12:16:12 GMT+05:30
- Last Updated:** 2016-10-05 12:18:54 GMT+05:30
- Delivery Delay:** 0 seconds

On the right side, queue statistics are shown:

- Default Visibility Timeout: 30 seconds
- Message Retention Period: 4 days
- Maximum Message Size: 256 KB
- Receive Message Wait Time: 0 seconds
- Messages Available (Visible): 1
- Messages in Flight (Not Visible): 0
- Messages Delayed: 0

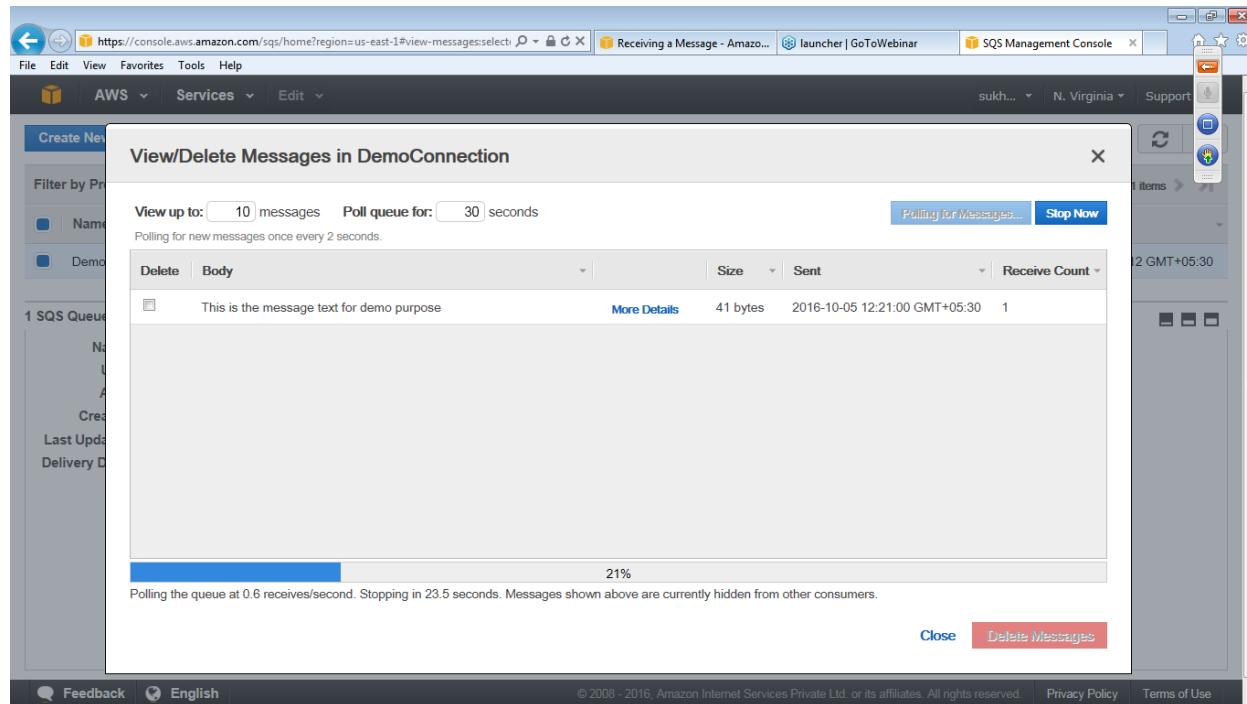
- 3 Click **Start Polling for Messages** to receive a message from the queue.



The screenshot shows the 'View/Delete Messages in DemoConnection' dialog box. It includes the following fields and controls:

- View up to:** 10 messages
- Poll queue for:** 30 seconds
- Start Polling for Messages** button
- Stop Now** button
- A descriptive text area explaining the polling process.
- Don't show this again.** checkbox
- Start Polling for Messages** button (inside the dialog)
- Close** and **Delete Messages** buttons at the bottom right.

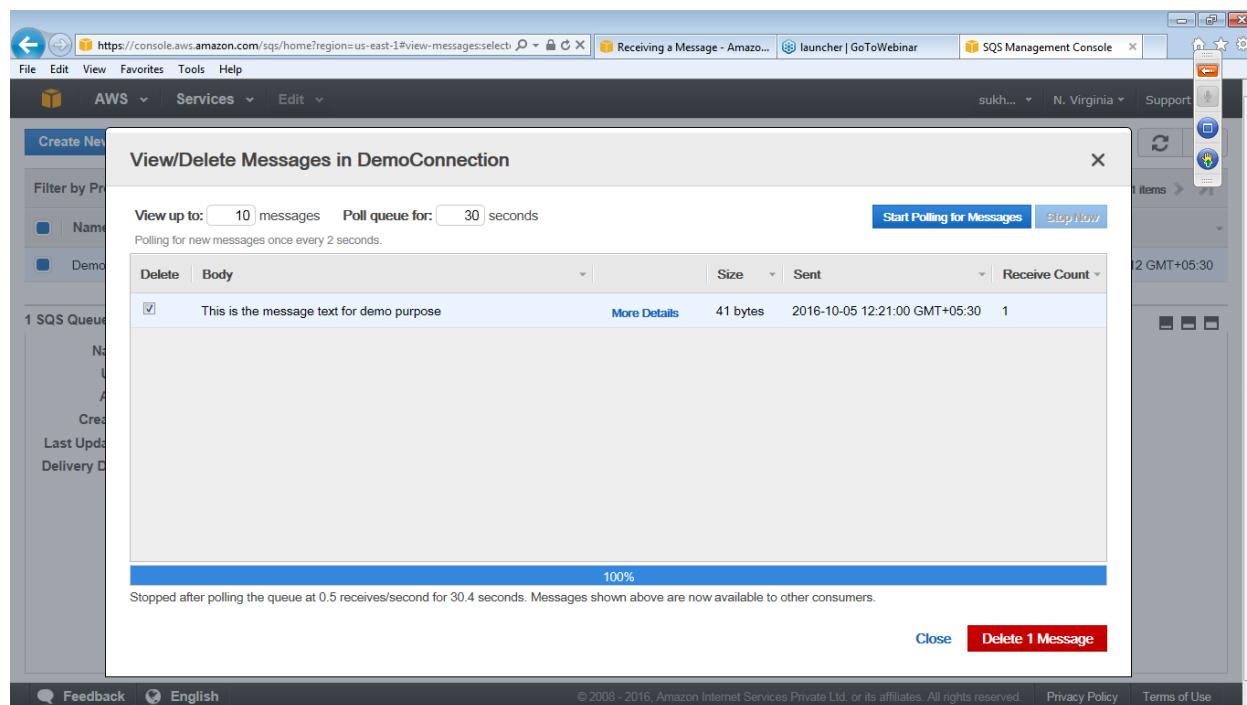
At the bottom left of the dialog, it says: "This progress bar indicates whether messages displayed above are available to applications." A progress bar is shown at 0%.



The screenshot shows the AWS SQS Management Console interface. A modal window titled "View/Delete Messages in DemoConnection" is open. It displays a single message in a table:

| Delete | Body | More Details | Size | Sent | Receive Count |
|--------------------------|---|--------------|----------|-------------------------------|---------------|
| <input type="checkbox"/> | This is the message text for demo purpose | | 41 bytes | 2016-10-05 12:21:00 GMT+05:30 | 1 |

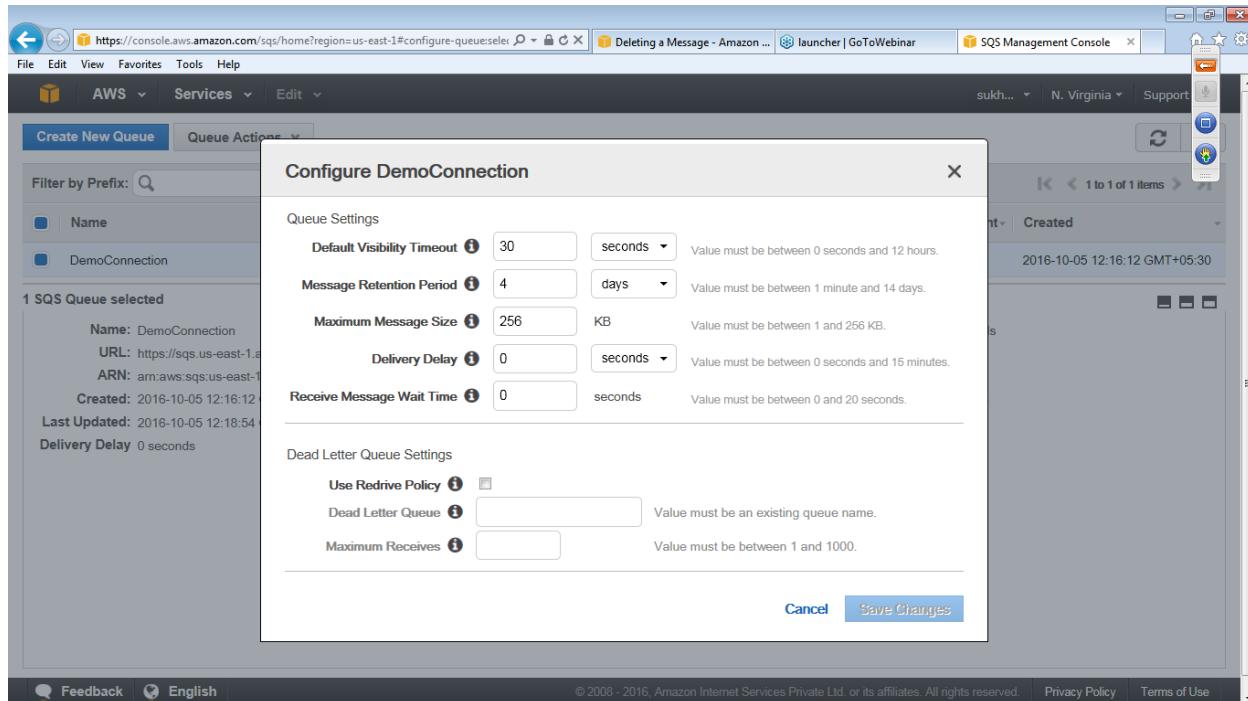
Below the table, a progress bar indicates "21%" completion of a poll operation. A status message says "Polling the queue at 0.6 receives/second. Stopping in 23.5 seconds. Messages shown above are currently hidden from other consumers." At the bottom right of the modal are "Close" and "Delete Messages" buttons.



The screenshot shows the same AWS SQS Management Console interface, but the modal window now displays a different state. The progress bar is at 100%, and the status message says "Stopped after polling the queue at 0.5 receives/second for 30.4 seconds. Messages shown above are now available to other consumers." The "Delete" checkbox is checked for the message. At the bottom right of the modal are "Close" and "Delete 1 Message" buttons.

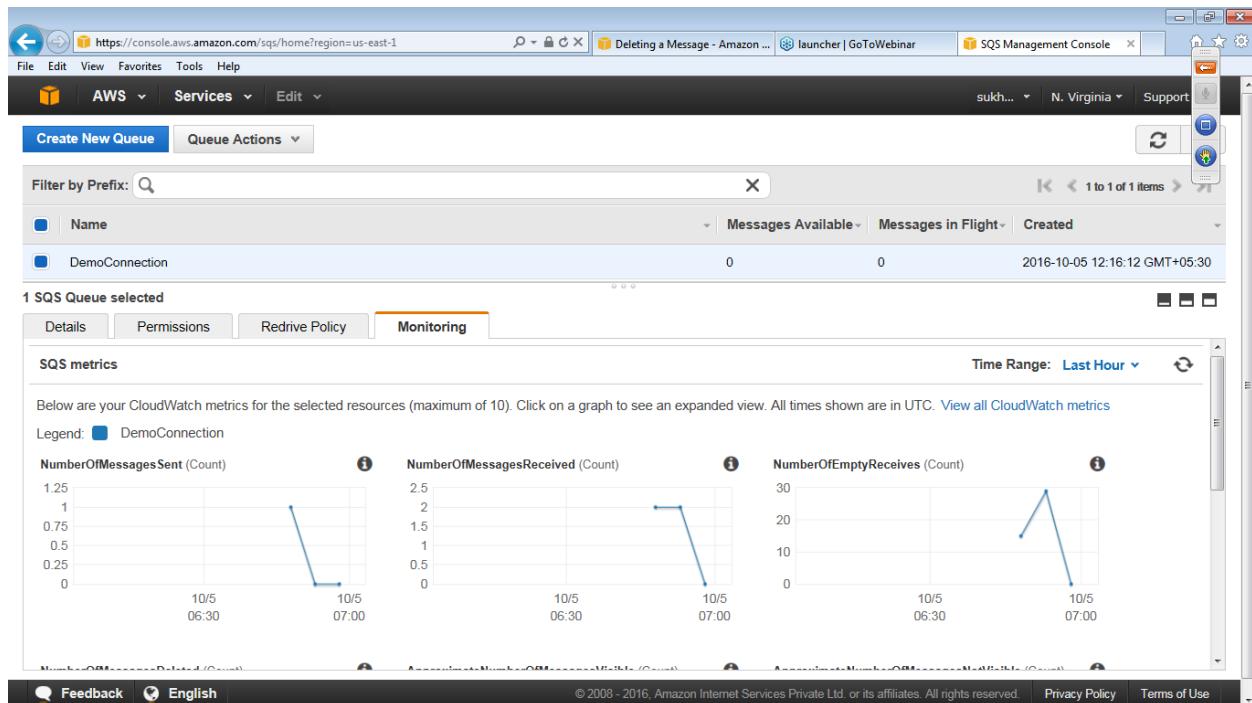
7.2.4 Modify Queue

- 1 In the AWS Management Console, select a queue
- 2 From the **Queue Action** drop down select the option **Configure Queues**

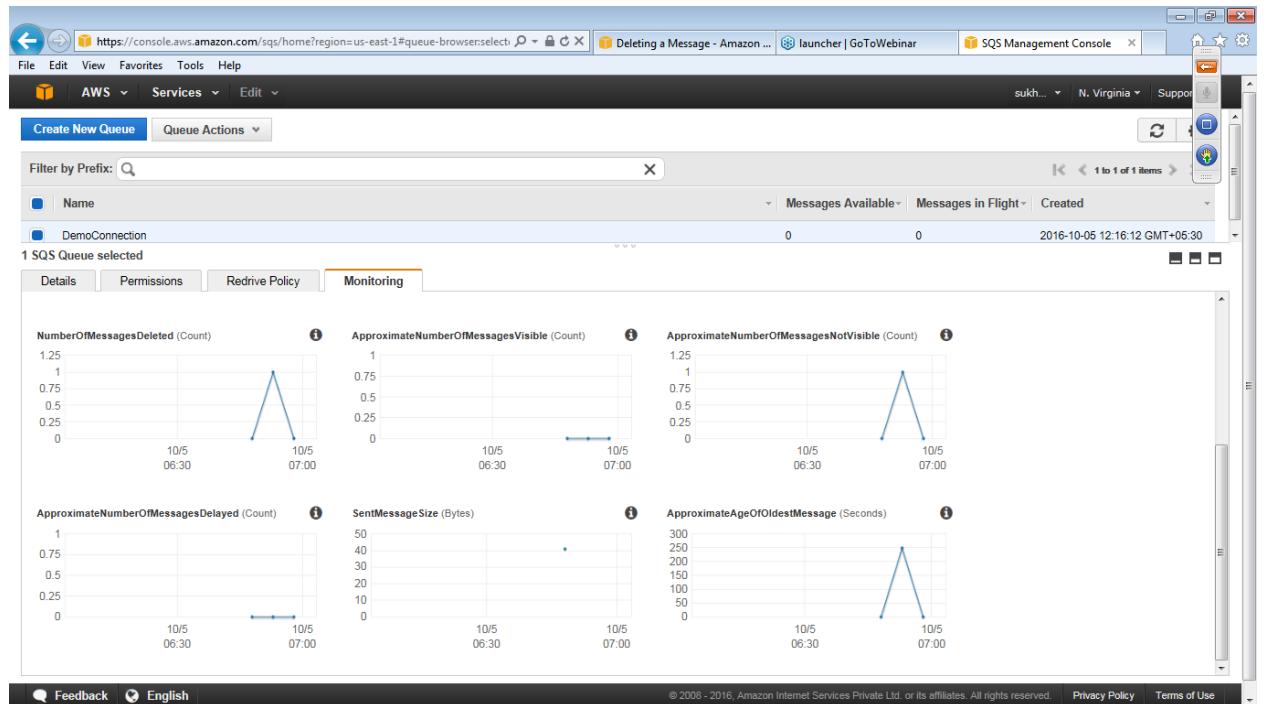


7.2.5 Monitoring Queue

- In the AWS Management Console, select a queue and click on the **Monitoring tab** and get the details of SQS Metrics



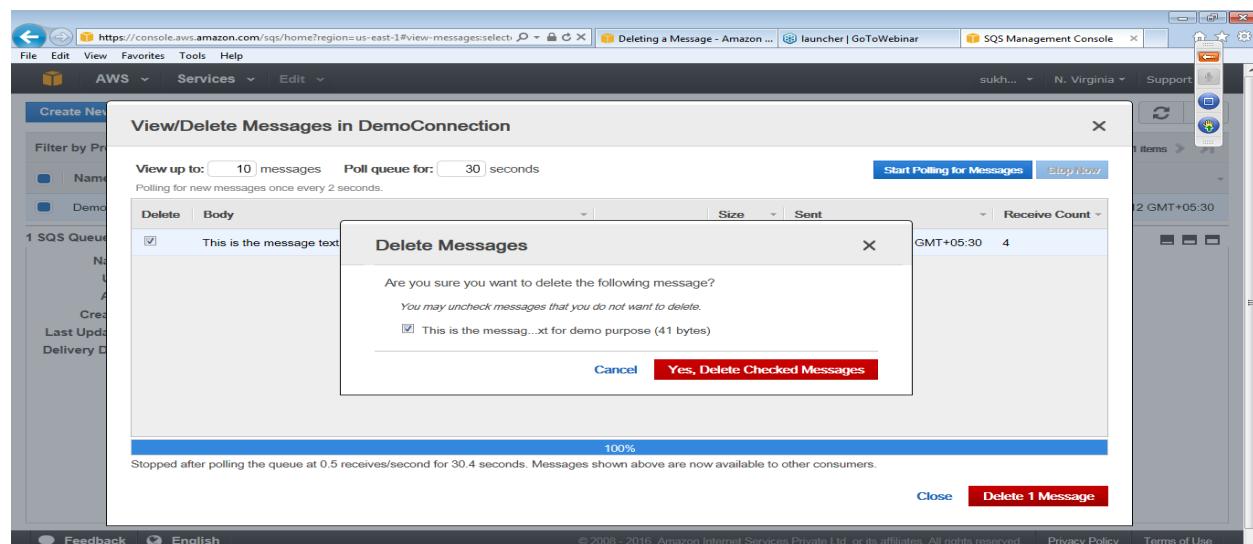
2



7.2.6 Deleting a Message

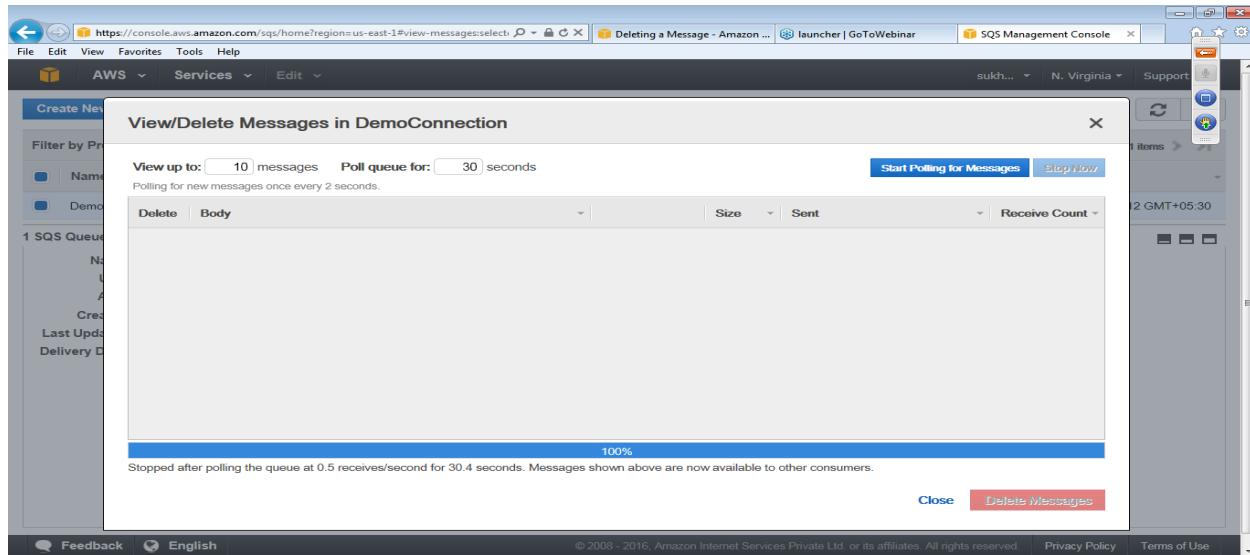
If you want to delete messages from your queue one at a time use Delete option

- 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

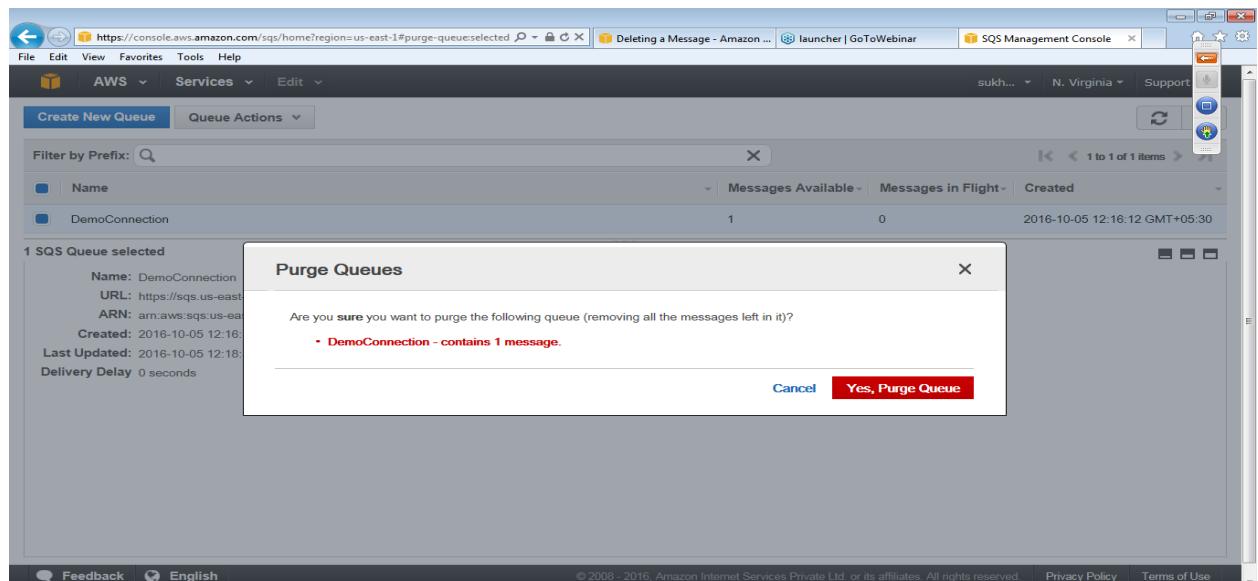
5 Click **Close** to close the **View/Delete Messages** dialog box



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.

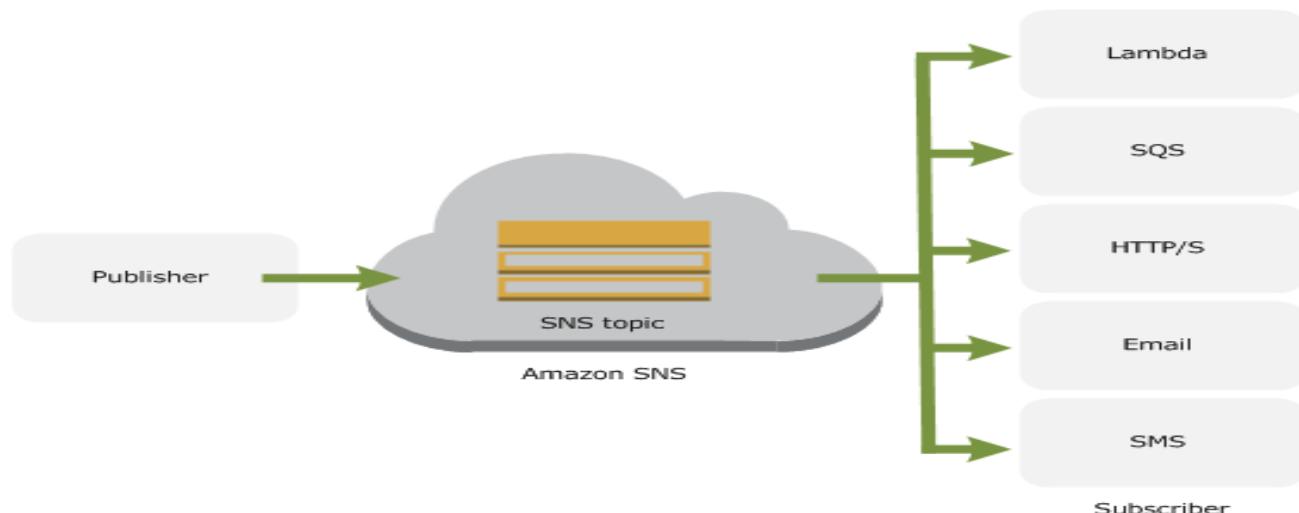
8.1 Objective

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.



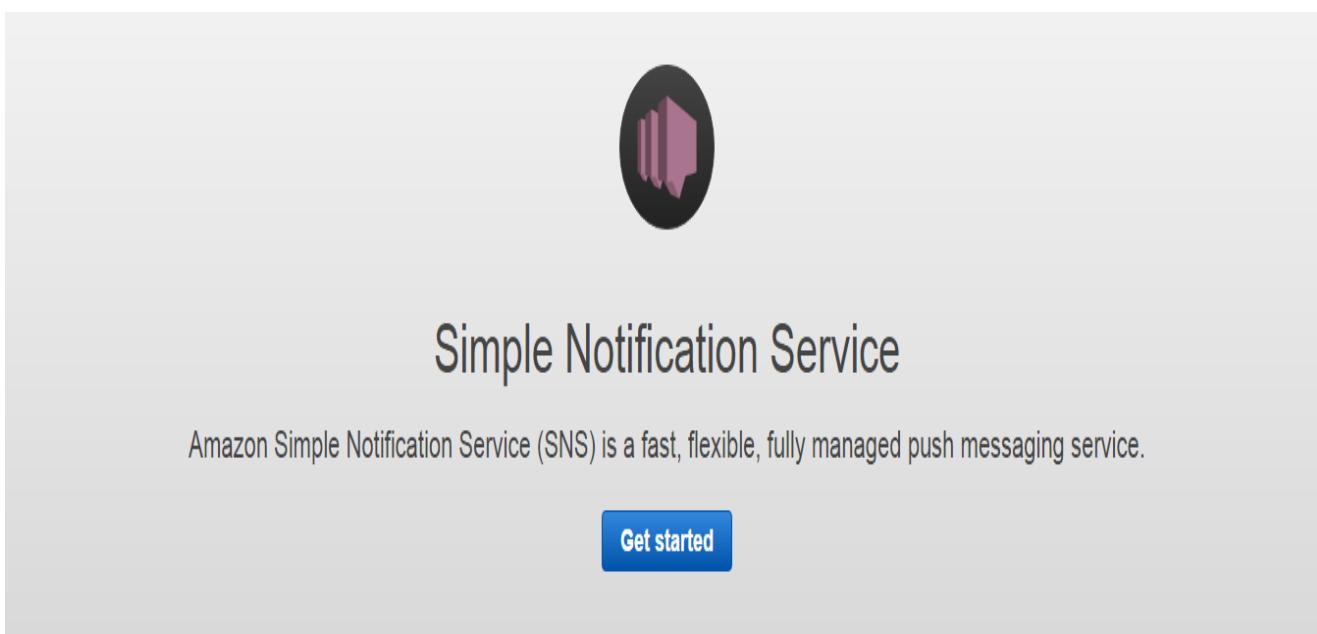
8.3 Procedure

8.3.1 Create SNS(Simple Notification Service).

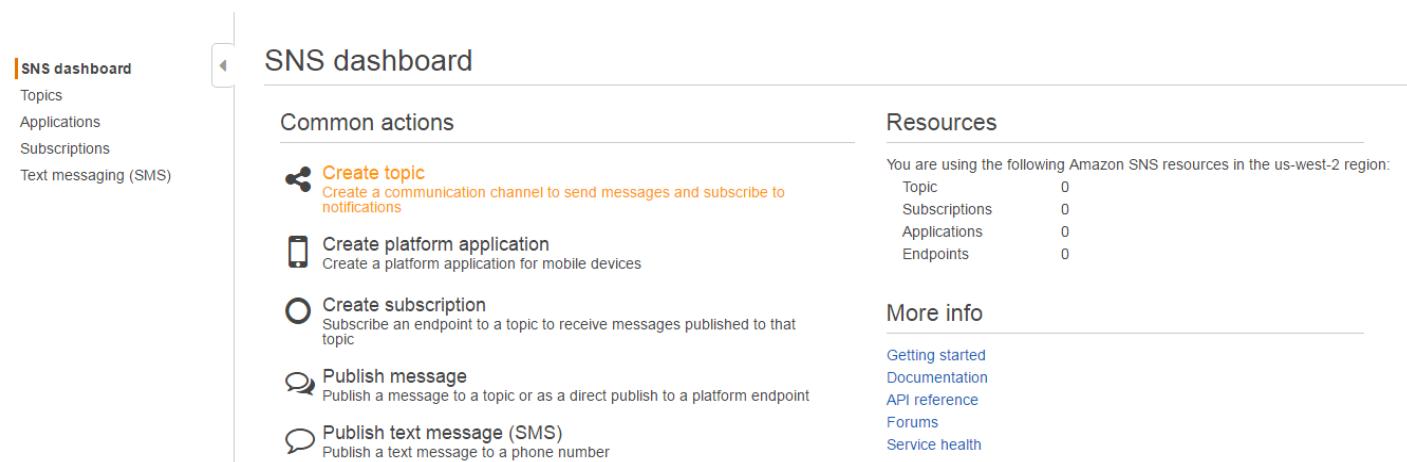
1. Sign in to AWS Management Console and select **SNS(Simple Notification Service)**.

| | | | |
|---|---|---|---|
|  Run and Manage Web Apps  Lambda Run Code without Thinking about Servers |  Release Software using Continuous Delivery |  Deploy and Scale Session-based Multiplayer Games | <a data-bbox="1295 375 1400 382" href="#">Create a Group <a data-bbox="1426 375 1504 382" href="#">Tag Editor |
| Storage & Content Delivery <ul style="list-style-type: none">  Scalable Storage in the Cloud  Global Content Delivery Network  Fully Managed File System for EC2  Archive Storage in the Cloud  Large Scale Data Transport  Hybrid Storage Integration | Management Tools <ul style="list-style-type: none">  Monitor Resources and Applications  Create and Manage Resources with Templates  Track User Activity and API Usage  Track Resource Inventory and Changes  Automate Operations with Chef  Create and Use Standardized Products  Optimize Performance and Security | Mobile Services <ul style="list-style-type: none">  Build, Test, and Monitor Mobile Apps  User Identity and App Data Synchronization  Test Android, iOS, and Web Apps on Real Devices in the Cloud  Collect, View and Export App Analytics  Push Notification Service | Additional Resources <ul style="list-style-type: none"> Getting Started  Read our documentation or view our training to learn more about AWS. AWS Console Mobile App  View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes. AWS Marketplace  Find and buy software, launch with 1-Click and pay by the hour. AWS re:Invent Announcements  Explore the next generation of AWS cloud capabilities. See what's new |
| Database <ul style="list-style-type: none">  Managed Relational Database Service  Managed NoSQL Database  In-Memory Cache  Petabyte-scale, Cost-Effective Data Warehousing  Managed Database Migration Service | Security & Identity <ul style="list-style-type: none">  Manage User Access and Encryption Keys  Host and Manage Active Directory  Analyze Application Security  Protect Malicious Web Traffic  Provision, Manage, and Deploy SSL/TLS Certificates | Application Services <ul style="list-style-type: none">  Build, Deploy and Manage APIs  Stream Application Streaming  Managed Search Service  Easy-to-Use Scalable Media Transcoding  Email Sending and Receiving Service  Message Queue Service  Scalable Work Flow Service | Service Health <p> All services operating normally.</p> <p>Updated: Sep 30 2016 11:26:00 GMT-0530</p> |

- ## 2. Click on Get Started.

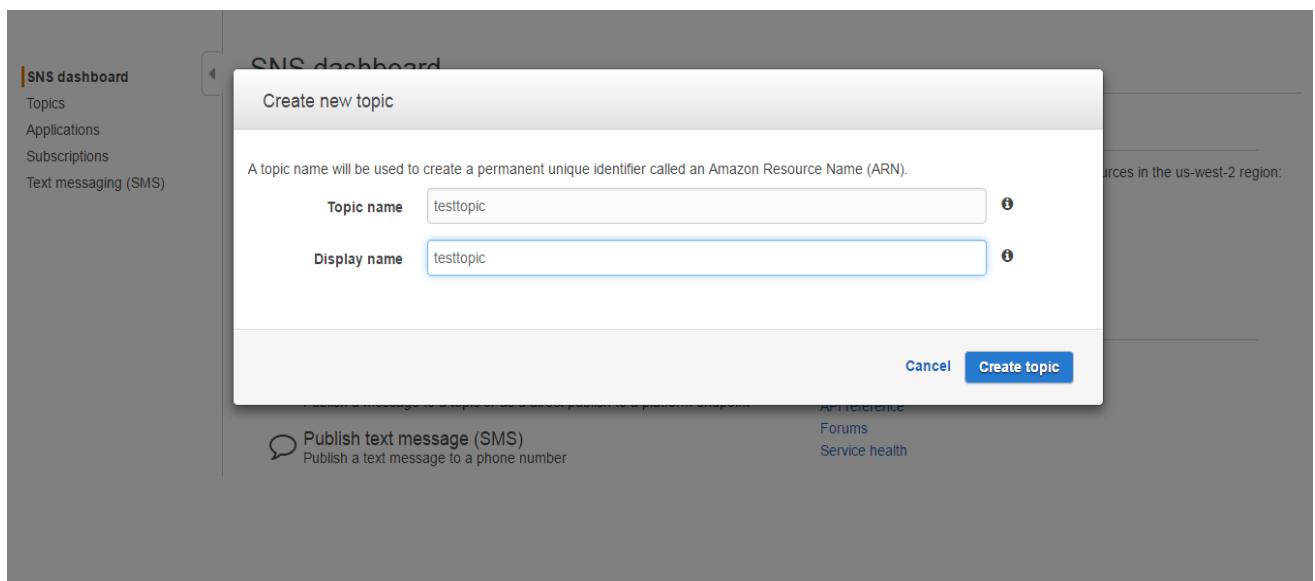


3. Now Click on **Create Topic**.



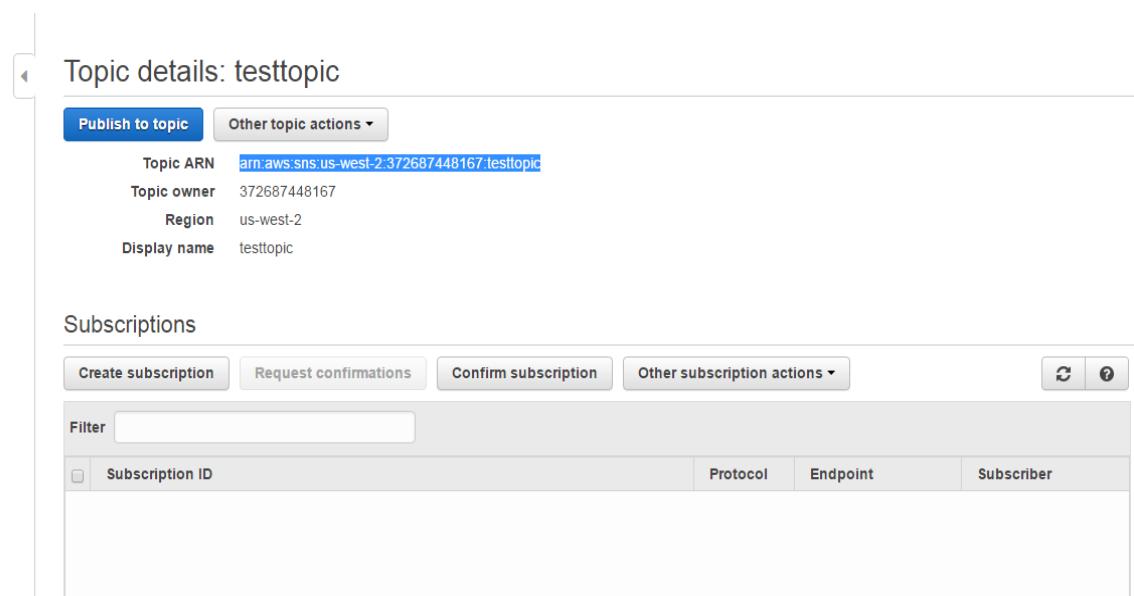
The screenshot shows the AWS SNS dashboard. On the left, there's a sidebar with links: 'Topics', 'Applications', 'Subscriptions', and 'Text messaging (SMS)'. The main area is titled 'SNS dashboard' and contains a 'Common actions' section with several buttons: 'Create topic' (highlighted in orange), 'Create platform application', 'Create subscription', 'Publish message', and 'Publish text message (SMS)'. To the right, there's a 'Resources' section showing usage statistics for the us-west-2 region: Topic (0), Subscriptions (0), Applications (0), and Endpoints (0). Below that is a 'More info' section with links to 'Getting started', 'Documentation', 'API reference', 'Forums', and 'Service health'.

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



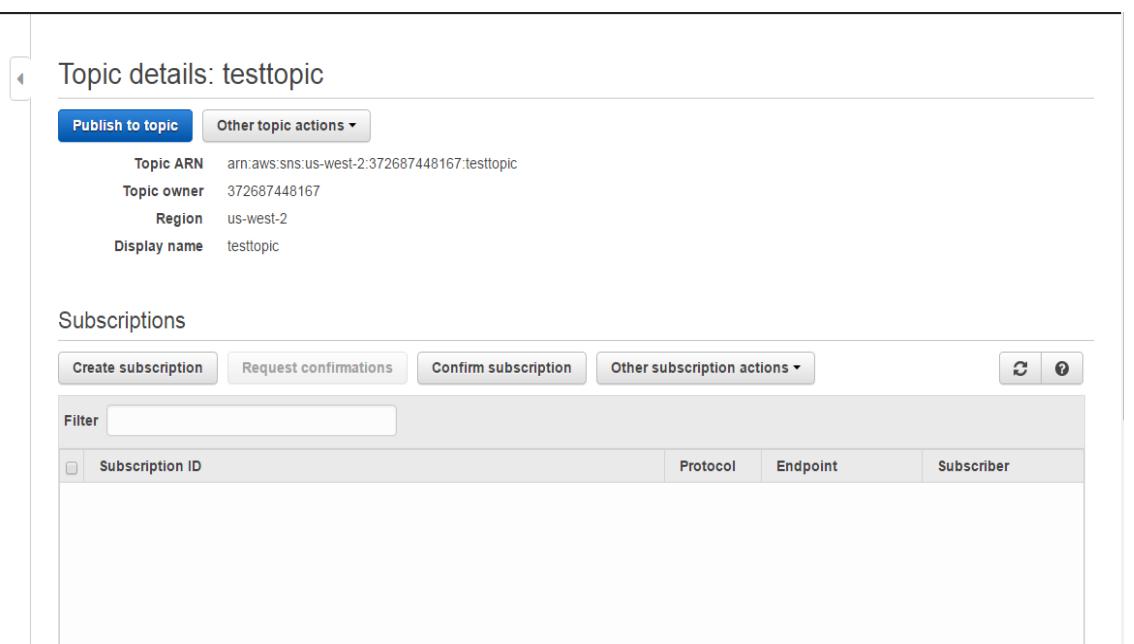
The screenshot shows a modal dialog box titled 'Create new topic'. It contains two input fields: 'Topic name' with the value 'testtopic' and 'Display name' with the value 'testtopic'. Below the inputs is a note: 'A topic name will be used to create a permanent unique identifier called an Amazon Resource Name (ARN)'. At the bottom of the dialog are 'Cancel' and 'Create topic' buttons. The background shows the SNS dashboard interface with a 'Publish text message (SMS)' button visible.

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



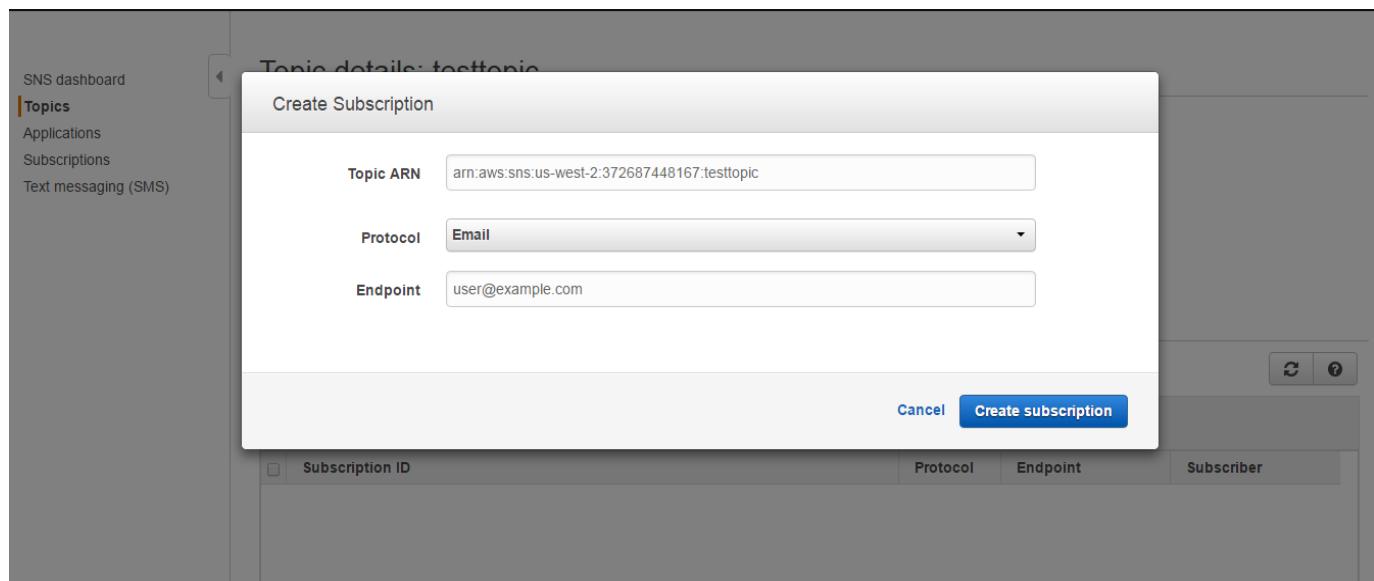
The screenshot shows the 'Topic details: testtopic' page. On the left, there's a sidebar with links: SNS dashboard, Topics (which is selected), Applications, Subscriptions, and Text messaging (SMS). The main area has tabs: 'Publish to topic' (selected) and 'Other topic actions'. Below these are topic details: Topic ARN (arn:aws:sns:us-west-2:372687448167:testtopic), Topic owner (372687448167), Region (us-west-2), and Display name (testtopic). A large section titled 'Subscriptions' contains buttons: 'Create subscription', 'Request confirmations', 'Confirm subscription', and 'Other subscription actions'. A 'Filter' input field is above a table with columns: Subscription ID, Protocol, Endpoint, and Subscriber. The table is currently empty.

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

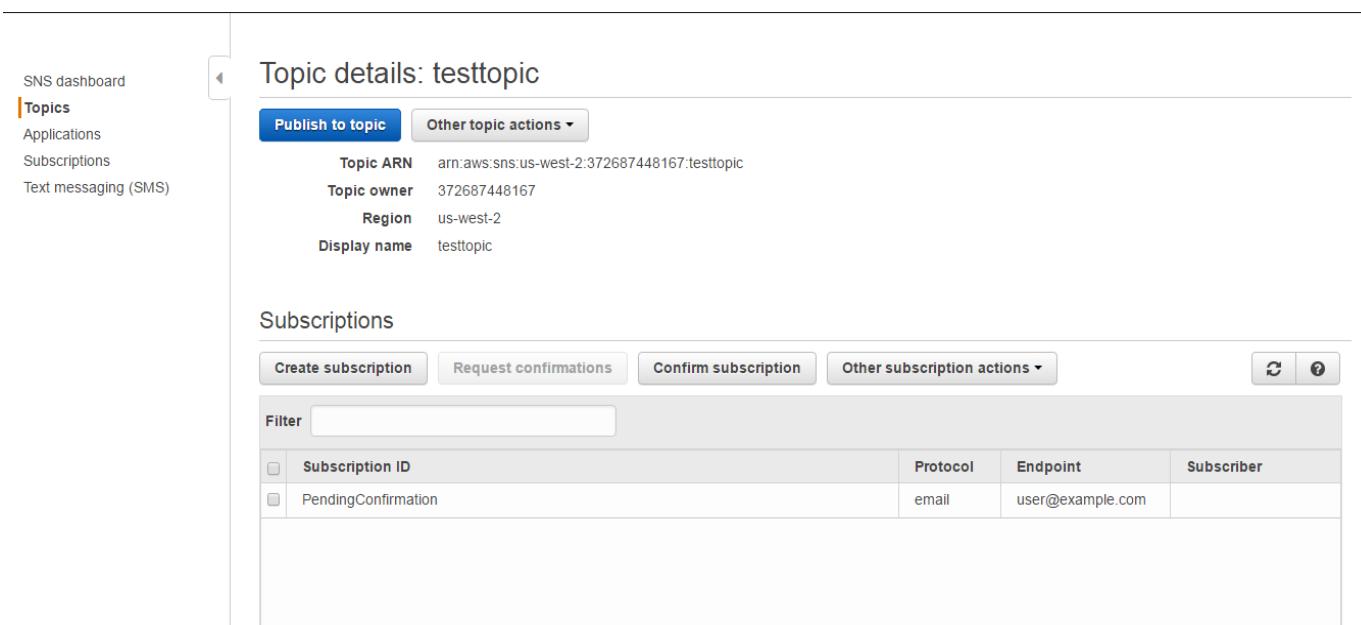


This screenshot is identical to the one above, showing the 'Topic details: testtopic' page. The 'Create subscription' button is highlighted in red, indicating it is the target for the next step. The rest of the interface, including the sidebar, topic details, and subscription table, remains the same.

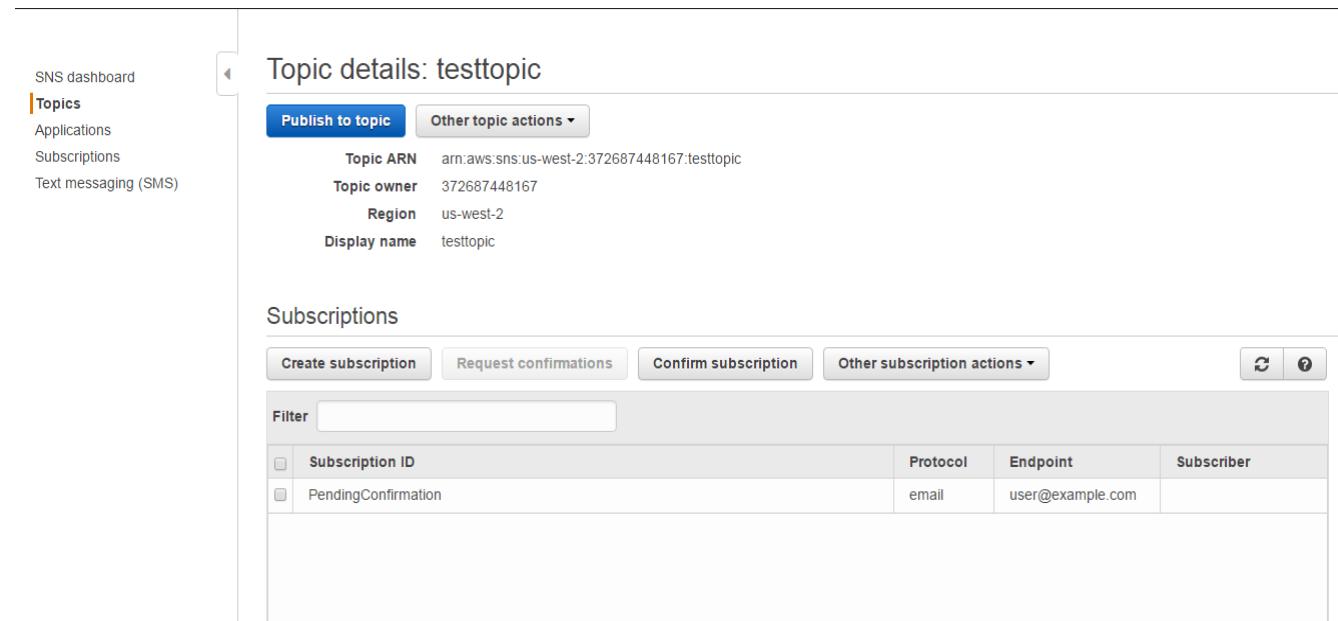
7.After Clicking **Create Subscription**, paste the copied Topic ARN which was copied in earlier step ,in Protocol select Email and in Endpoint Give the Email ID and click **Create Subscription**.



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

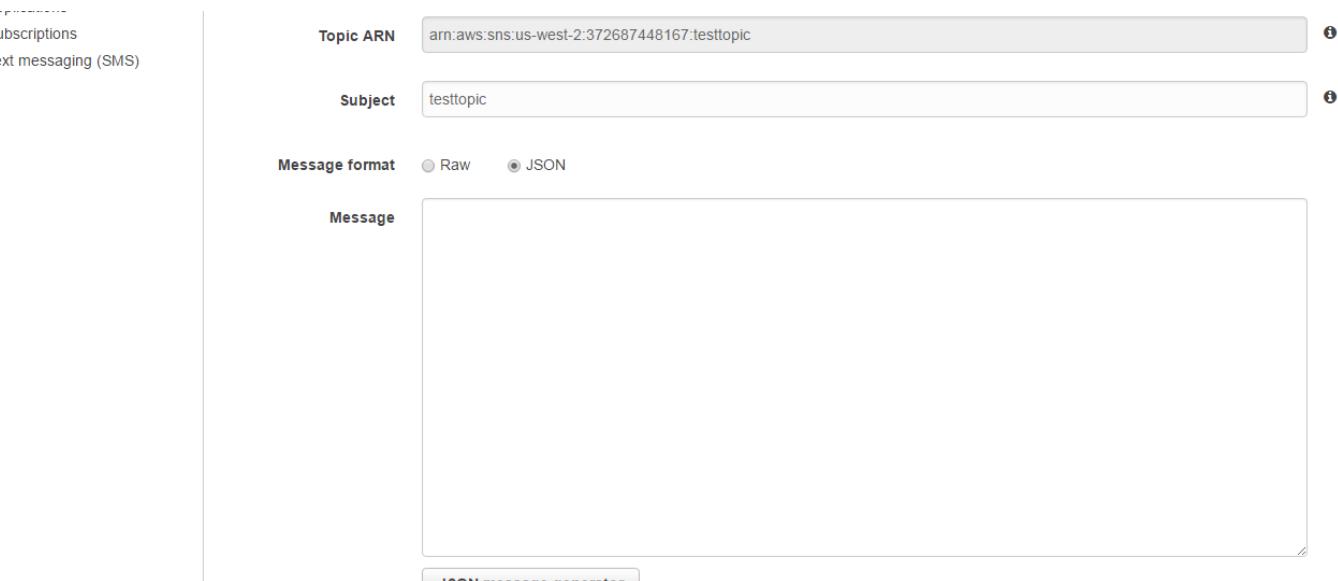


9.In this,click on **Publish to topic**.



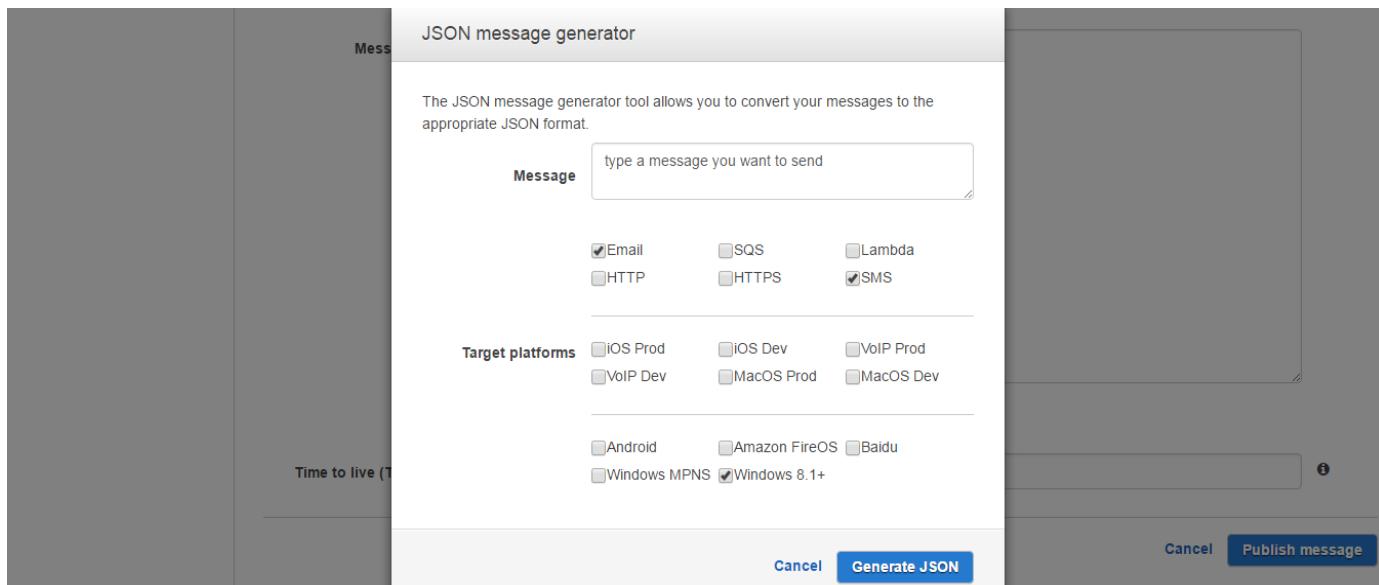
The screenshot shows the AWS SNS Topic Details page for a topic named 'testtopic'. The left sidebar has 'Topics' selected. The main area displays topic details: Topic ARN (arn:aws:sns:us-west-2:372687448167:testtopic), Topic owner (372687448167), Region (us-west-2), and Display name (testtopic). Below this is a 'Subscriptions' section with a table showing one subscription: Subscription ID (PendingConfirmation), Protocol (email), Endpoint (user@example.com), and Subscriber (empty). At the top right of the main area, there is a 'Publish to topic' button.

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



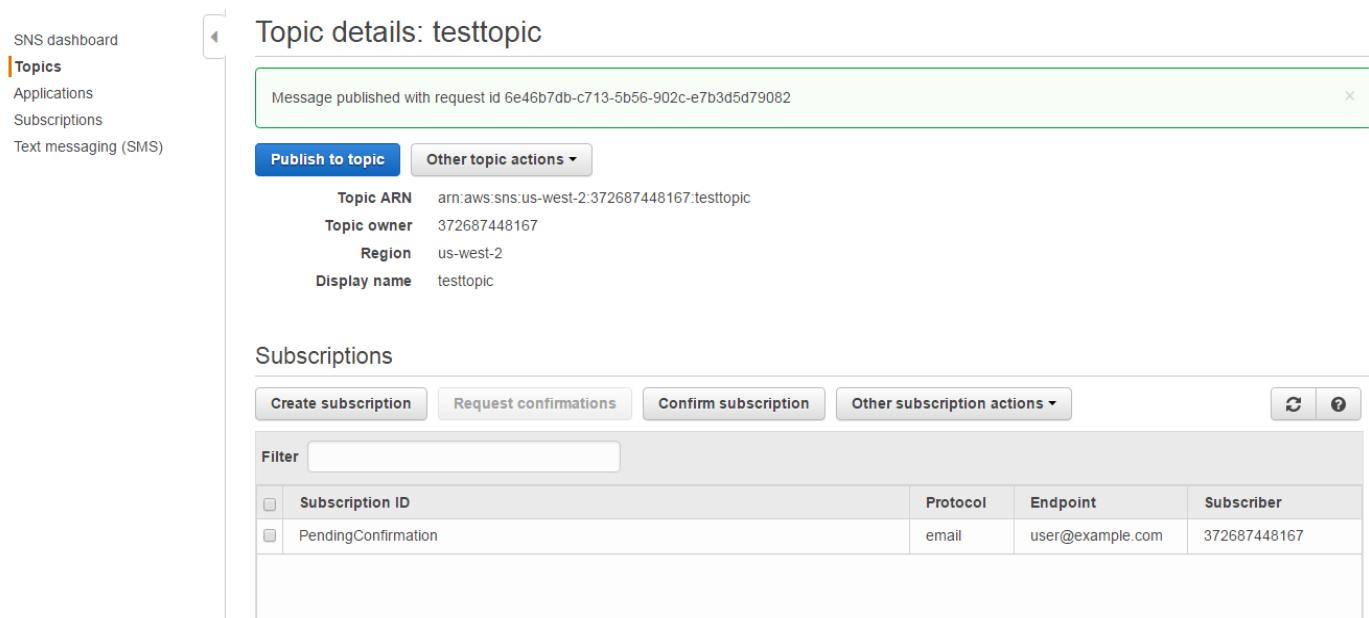
The screenshot shows the AWS SNS JSON message generator interface. On the left sidebar, 'Text messaging (SMS)' is selected. The main area has fields for Topic ARN (arn:aws:sns:us-west-2:372687448167:testtopic) and Subject (testtopic). Below these, a 'Message format' section has 'JSON' selected. A large text area labeled 'Message' is present, with a 'JSON message generator' button at the bottom.

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**.



The screenshot shows the "JSON message generator" interface. It includes a message input field, several checkboxes for target platforms (Email, SMS, Windows 8.1+ are selected), and a "Generate JSON" button.

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

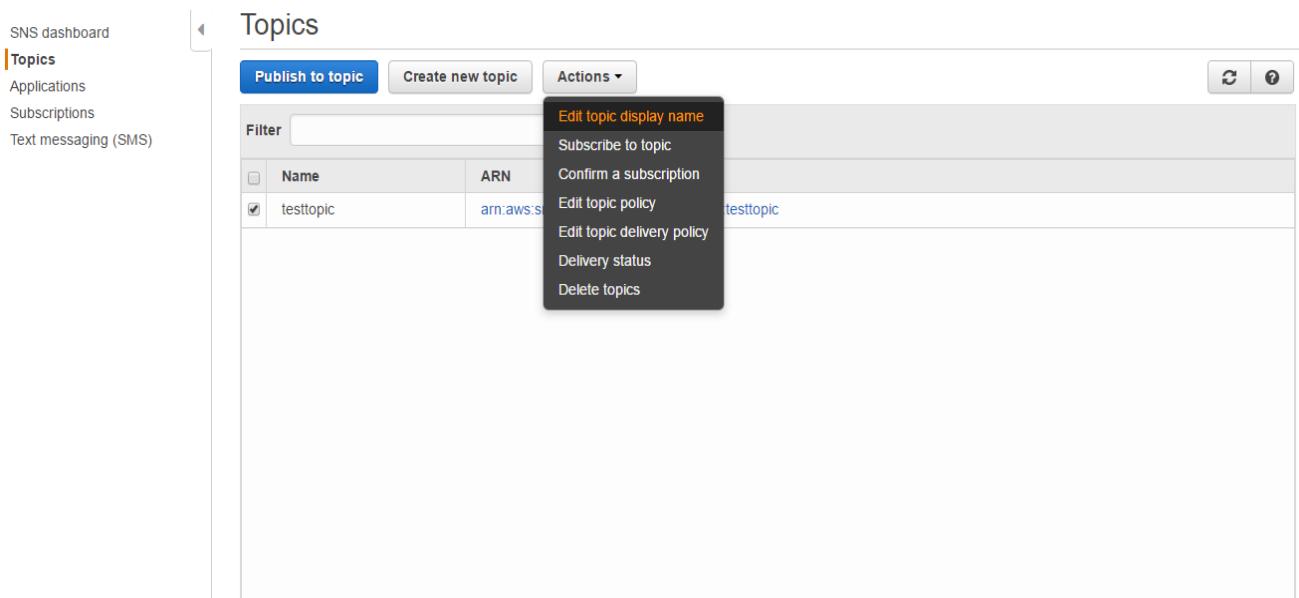


The screenshot shows the "Topic details: testtopic" page. It displays a message published with request id 6e46b7db-c713-5b56-902c-e7b3d5d79082. Below this, it shows topic ARN, owner, region, and display name. The "Subscriptions" section lists a single subscription for "PendingConfirmation" with email "user@example.com".

| Subscription ID | Protocol | Endpoint | Subscriber |
|---------------------|----------|------------------|--------------|
| PendingConfirmation | email | user@example.com | 372687448167 |

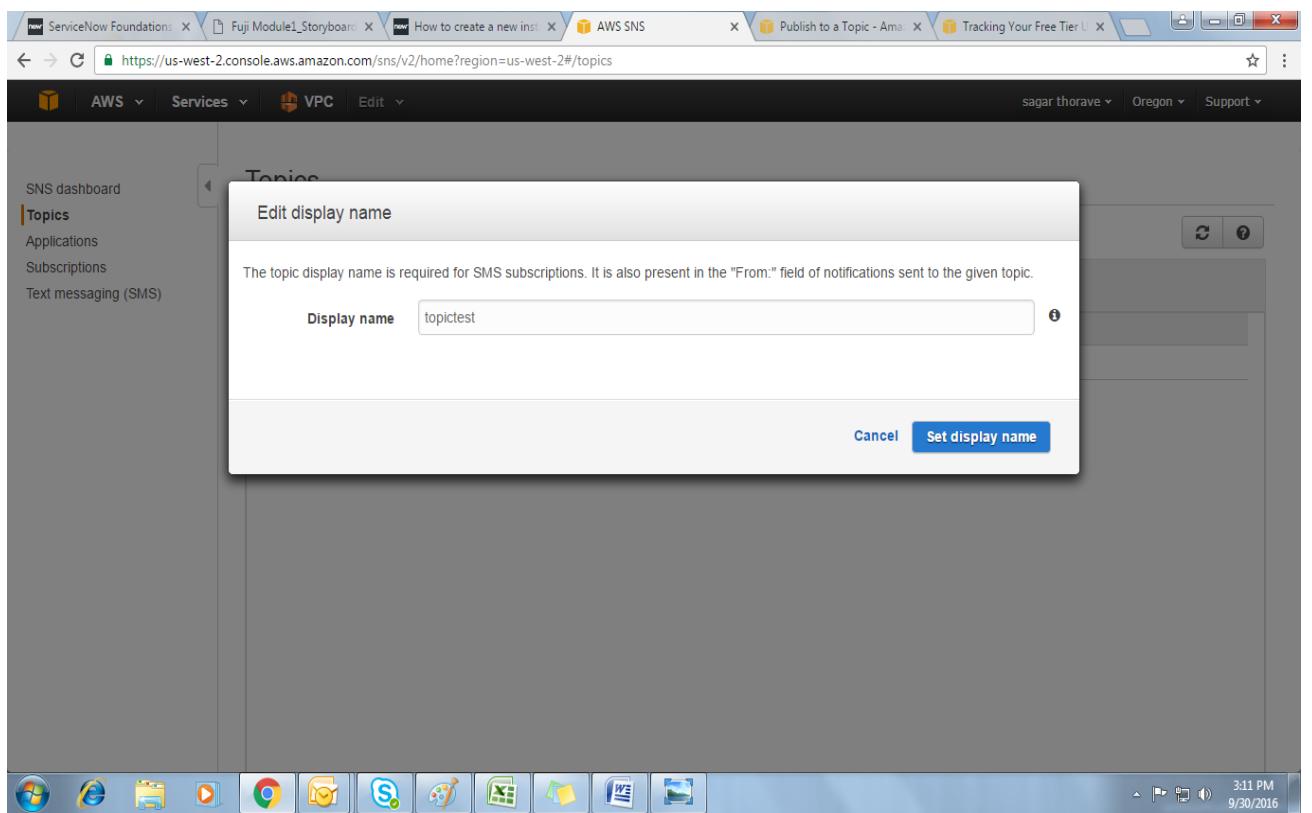
8.3.2 Modifying SNS(Simple Notification Service).

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



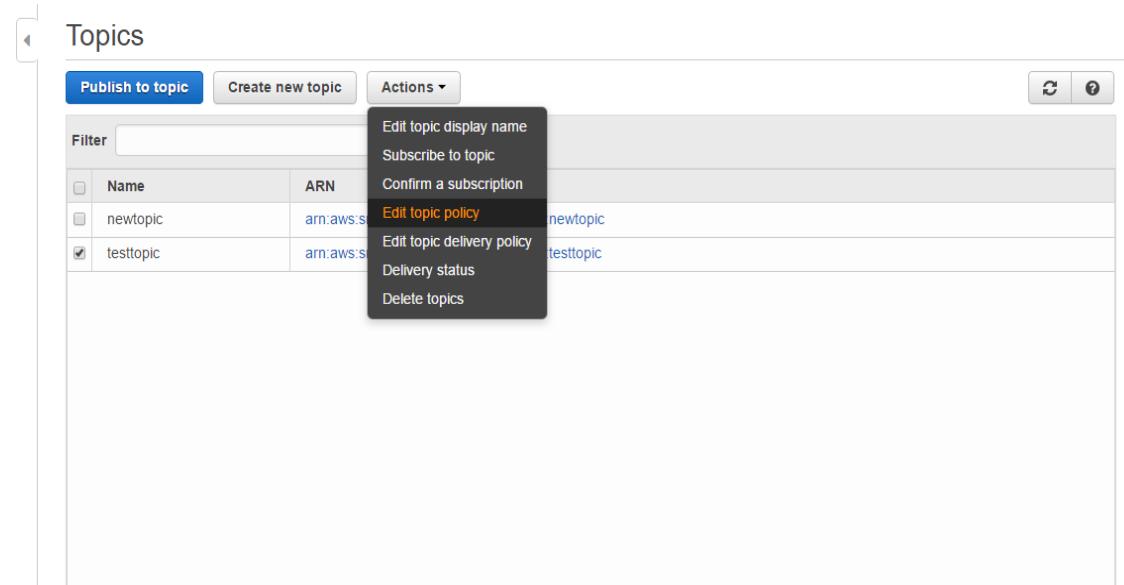
The screenshot shows the AWS SNS Topics page. On the left, there's a sidebar with links: SNS dashboard, Topics (which is selected and highlighted in orange), Applications, Subscriptions, and Text messaging (SMS). The main area has tabs: Publish to topic, Create new topic, and Actions (with a dropdown arrow). A context menu is open over a row for a topic named "testtopic". The menu items are: Edit topic display name (highlighted in orange), Subscribe to topic, Confirm a subscription, Edit topic policy, Edit topic delivery policy, Delivery status, and Delete topics.

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

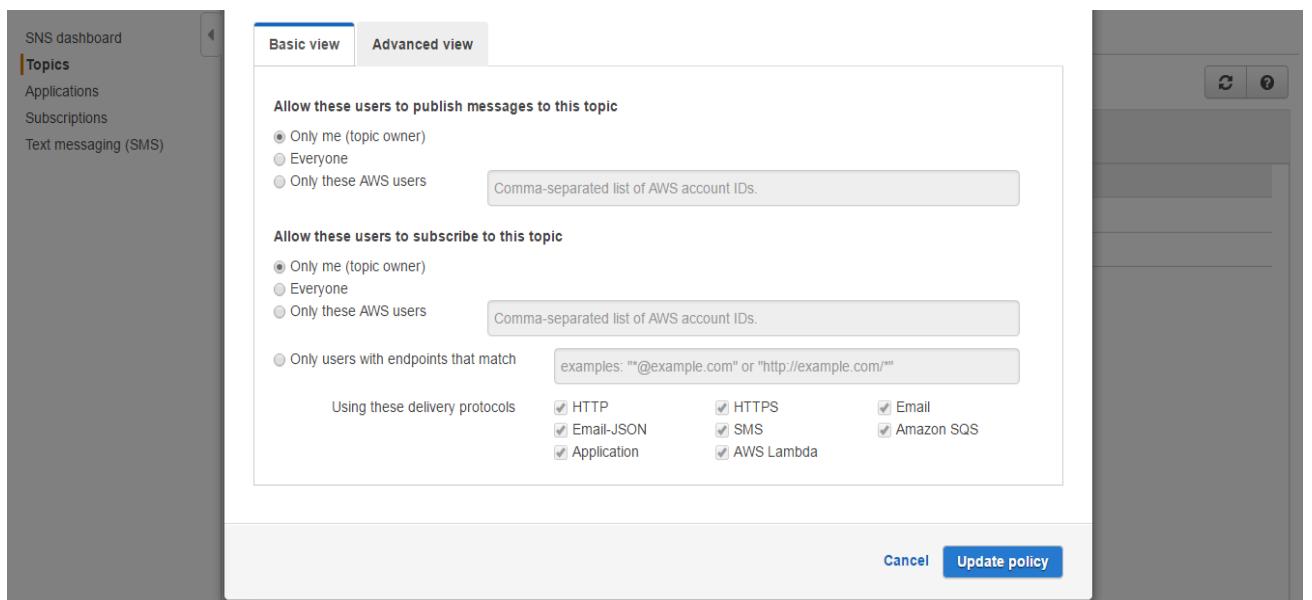


The screenshot shows the AWS SNS Topics page with a modal dialog box titled "Edit display name". The dialog contains a single input field labeled "Display name" with the value "topicstest". At the bottom right of the dialog are two buttons: "Cancel" and "Set display name". The background of the main SNS page is visible, showing the list of topics.

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**.



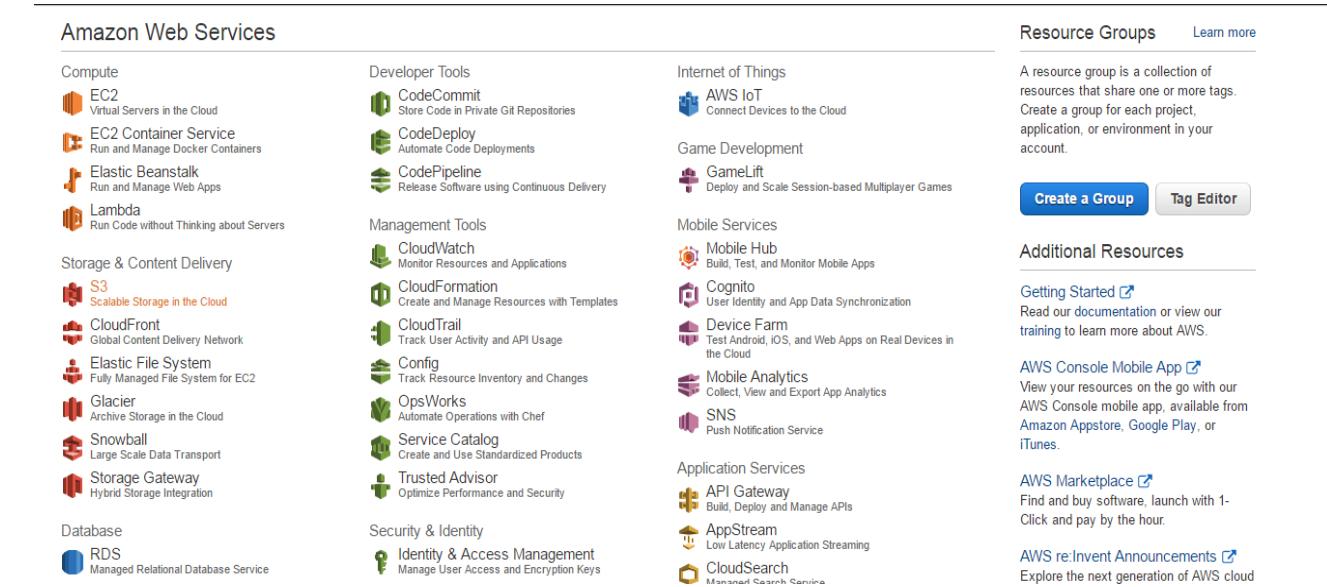
9. S3 & GLACIER MANAGEMENT

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)**



Amazon Web Services

- Compute**
 - EC2** Virtual Servers in the Cloud
 - EC2 Container Service** Run and Manage Docker Containers
 - Elastic Beanstalk** Run and Manage Web Apps
 - Lambda** Run Code without Thinking about Servers
- Storage & Content Delivery**
 - S3** Scalable Storage in the Cloud
 - CloudFront** Global Content Delivery Network
 - Elastic File System** Fully Managed File System for EC2
 - Glacier** Archive Storage in the Cloud
 - Snowball** Large Scale Data Transport
 - Storage Gateway** Hybrid Storage Integration
- Database**
 - RDS** Managed Relational Database Service

Developer Tools

- CodeCommit** Store Code in Private Git Repositories
- CodeDeploy** Automate Code Deployments
- CodePipeline** Release Software using Continuous Delivery

Management Tools

- CloudWatch** Monitor Resources and Applications
- CloudFormation** Create and Manage Resources with Templates
- CloudTrail** Track User Activity and API Usage
- Config** Track Resource Inventory and Changes
- OpsWorks** Automate Operations with Chef
- Service Catalog** Create and Use Standardized Products
- Trusted Advisor** Optimize Performance and Security

Internet of Things

- AWS IoT** Connect Devices to the Cloud

Game Development

- GameLift** Deploy and Scale Session-based Multiplayer Games

Mobile Services

- Mobile Hub** Build, Test, and Monitor Mobile Apps
- Cognito** User Identity and App Data Synchronization
- Device Farm** Test Android, iOS, and Web Apps on Real Devices in the Cloud
- Mobile Analytics** Collect, View and Export App Analytics
- SNS** Push Notification Service

Application Services

- API Gateway** Build, Deploy and Manage APIs
- AppStream** Low Latency Application Streaming
- CloudSearch** Managed Search Service

Resource Groups [Learn more](#)

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Create a Group **Tag Editor**

Additional Resources

- Getting Started** Read our documentation or view our training to learn more about AWS.
- AWS Console Mobile App** View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
- AWS Marketplace** Find and buy software, launch with 1-Click and pay by the hour.
- AWS re:Invent Announcements** Explore the next generation of AWS cloud

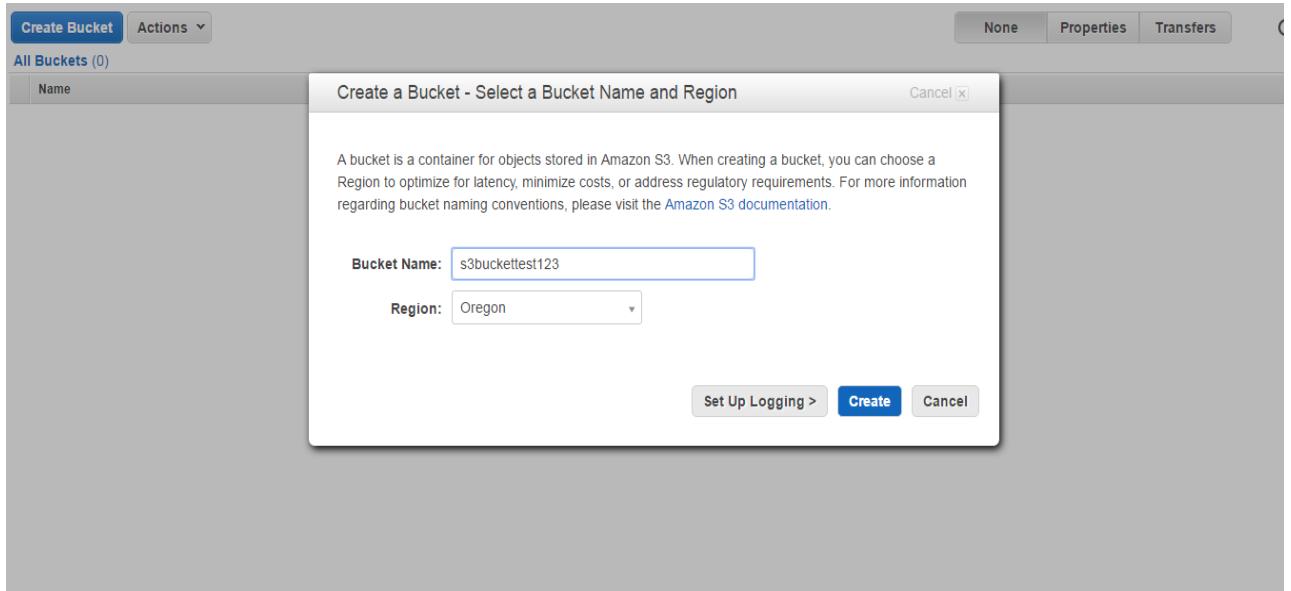
9.2.1 Create Bucket

Click on **Create Bucket** button.

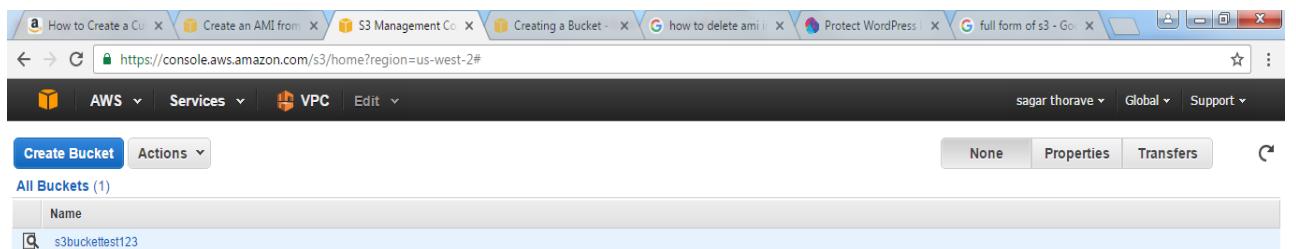


| Name |
|------|
| |

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

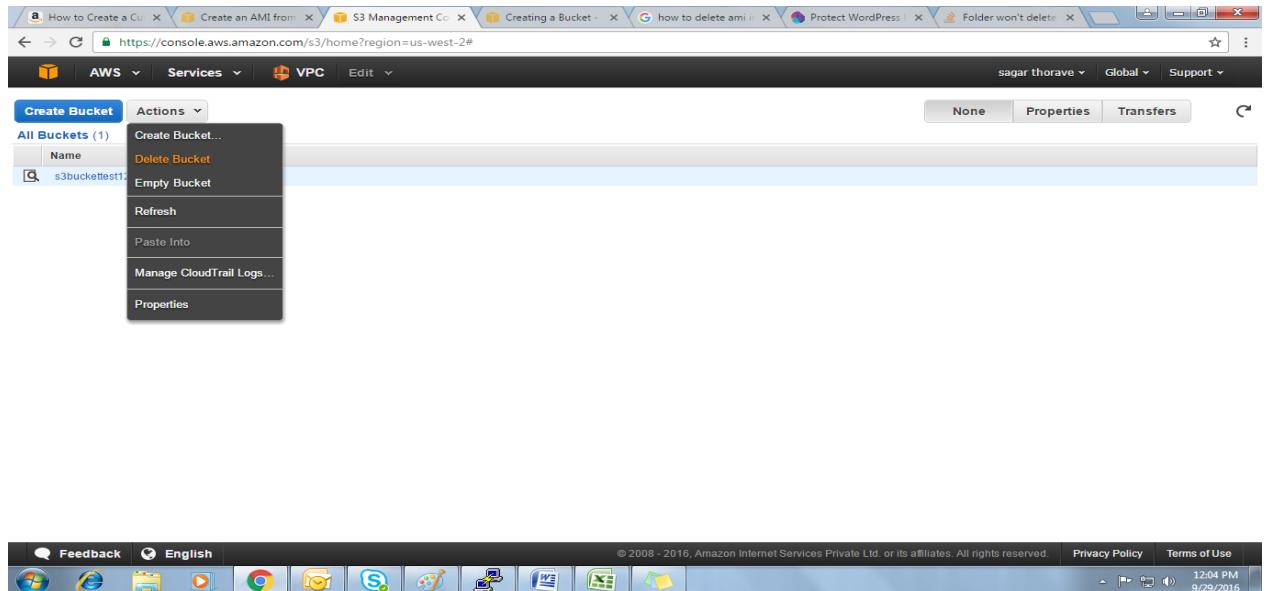


8. Here you can see that the Bucket named as s3buckettest123 has been created.

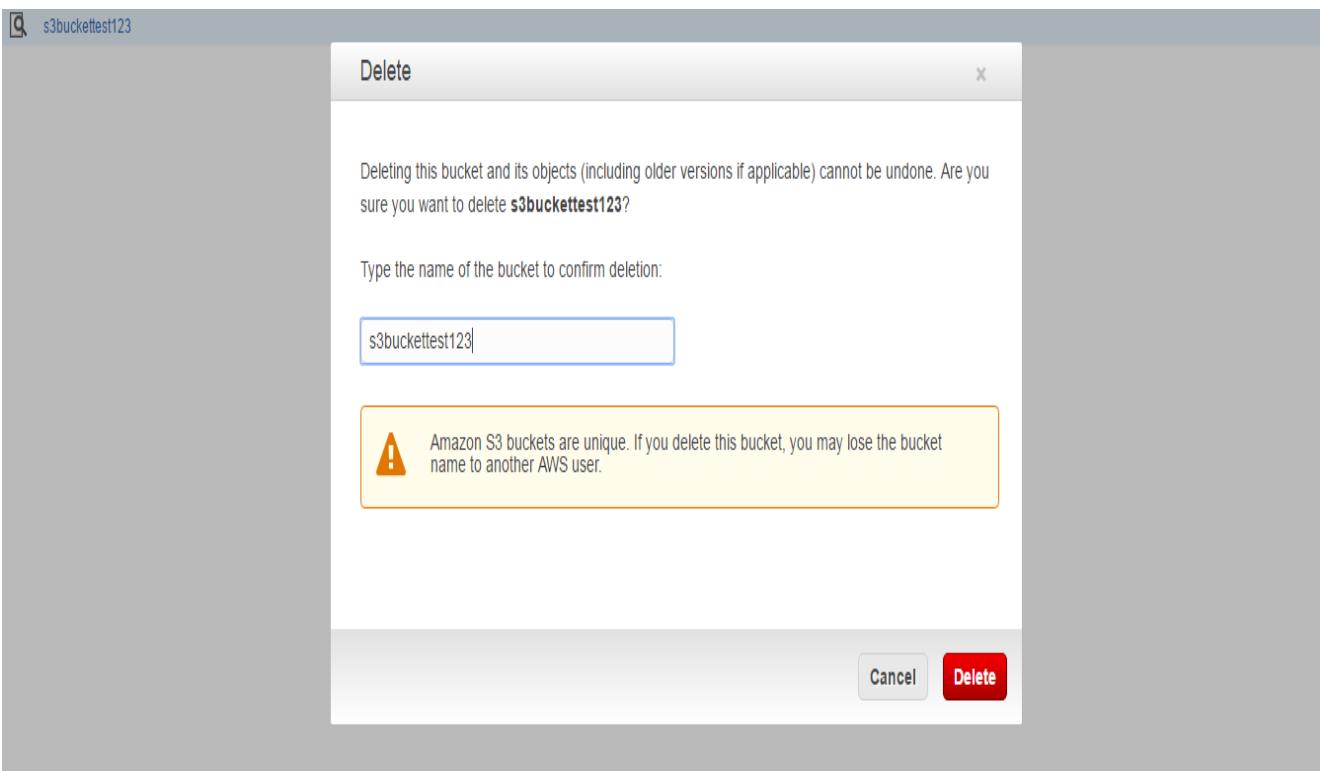


9.2.2 Delete Bucket

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.



The screenshot shows the AWS S3 console interface. At the top, there are three buttons: 'Upload', 'Create Folder', and 'Actions'. Below these are tabs for 'All Buckets' and the selected bucket 's3buckettest123'. A search bar labeled 'Search by prefix' is followed by buttons for 'None', 'Properties', and 'Transfers'. A refresh icon is also present. The main area displays a table with columns for 'Name', 'Storage Class', 'Size', and 'Last Modified'. A message at the top of the table says 'The bucket 's3buckettest123' is empty'.

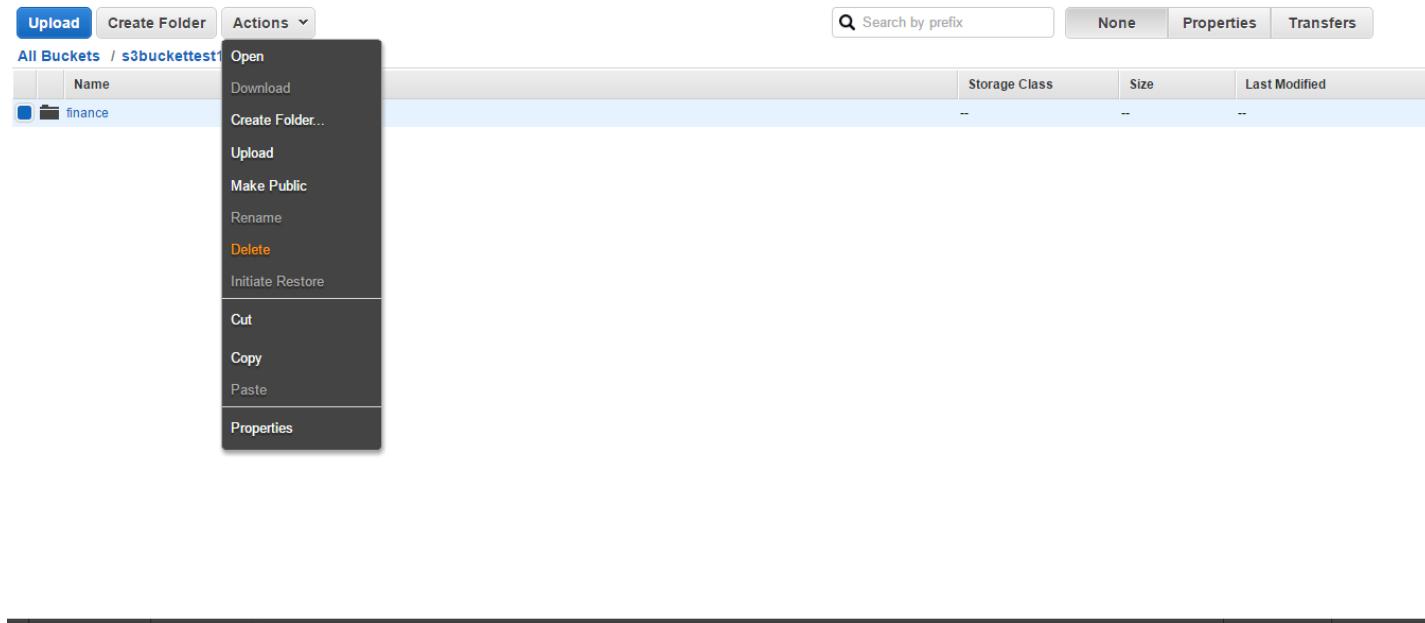
6. After clicking **Create Folder**, give the name of the folder you want to create in my case I have given finance.



The screenshot shows the same AWS S3 console interface after creating a folder. The table now lists one item: 'finance' under the 'Name' column. The other columns ('Storage Class', 'Size', 'Last Modified') are shown with their respective icons and values.

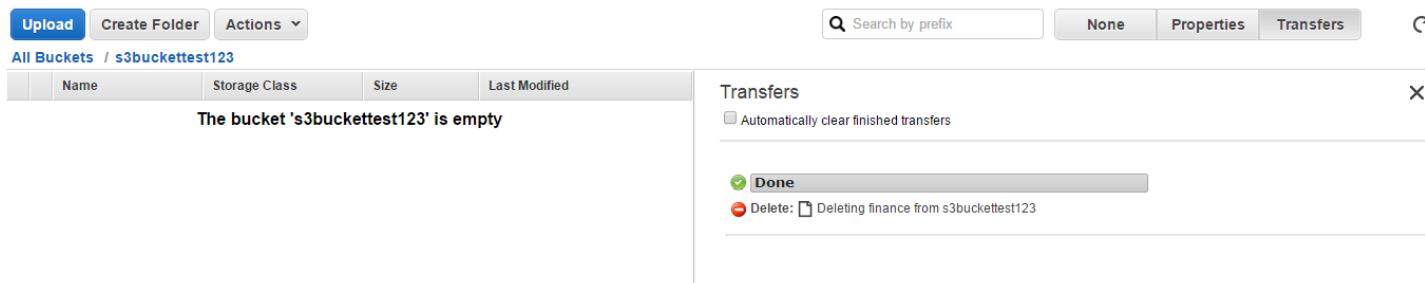
9.2.4 Delete Folder

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



The screenshot shows the AWS S3 console interface. At the top, there are buttons for 'Upload', 'Create Folder', and 'Actions'. A dropdown menu is open over a folder named 'finance' in the 's3buckettest1' bucket. The menu options include 'Open', 'Download', 'Create Folder...', 'Upload', 'Make Public', 'Rename', 'Delete' (which is highlighted in orange), 'Initiate Restore', 'Cut', 'Copy', 'Paste', and 'Properties'. Below the menu, the main content area shows a table with columns for Name, Storage Class, Size, and Last Modified. There is one entry: 'finance'. At the bottom of the screen, there are links for 'Feedback', 'English', and 'Privacy Policy/Terms of Use'.

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



The screenshot shows the AWS S3 console after a deletion. The main area displays the message 'The bucket 's3buckettest123' is empty'. On the right side, there is a 'Transfers' sidebar with a section for 'Automatically clear finished transfers'. Below it, a progress bar indicates a task is 'Done', and a note says 'Delete: Deleting finance from s3buckettest123'.

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

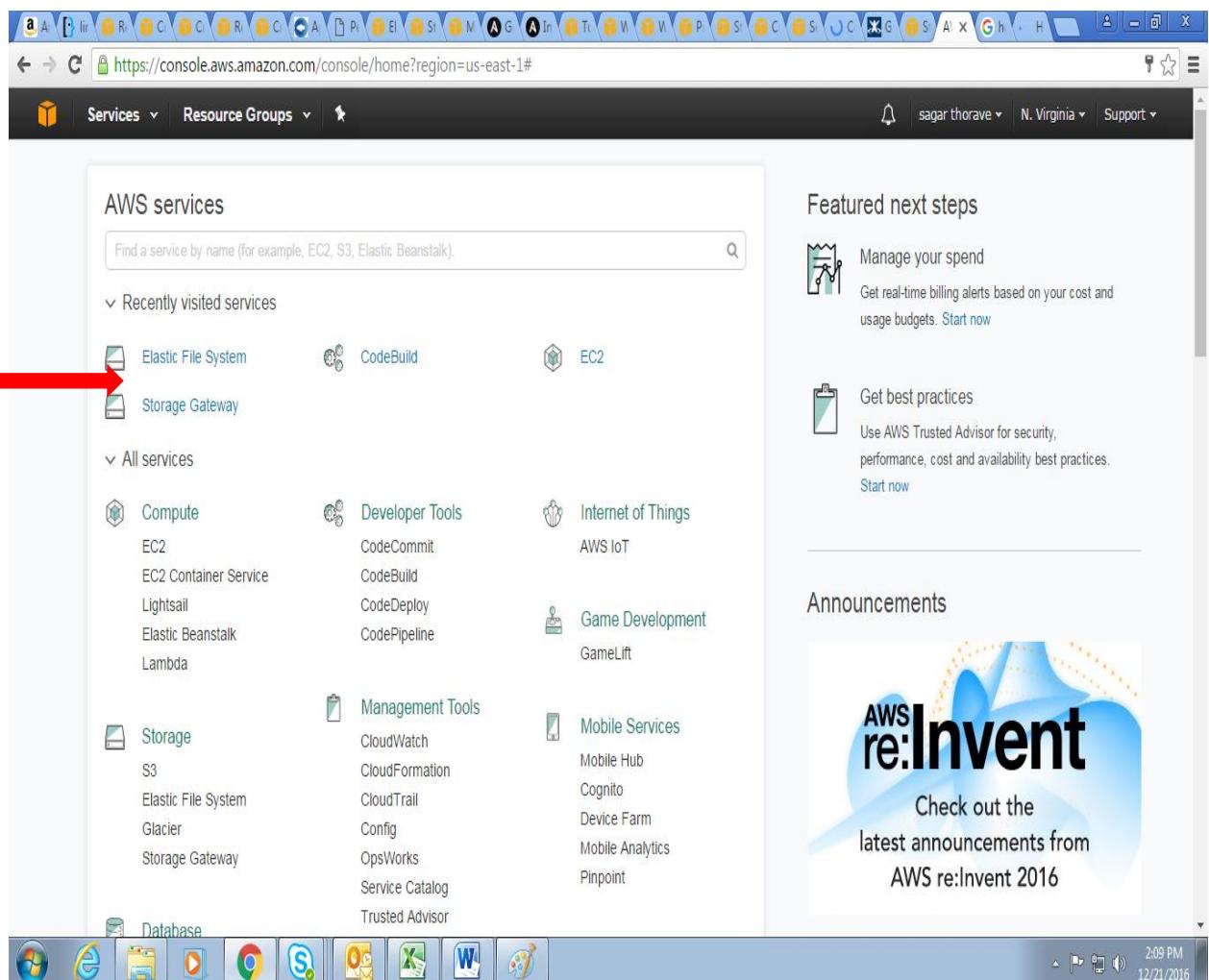
10.2 Assumption

- 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.

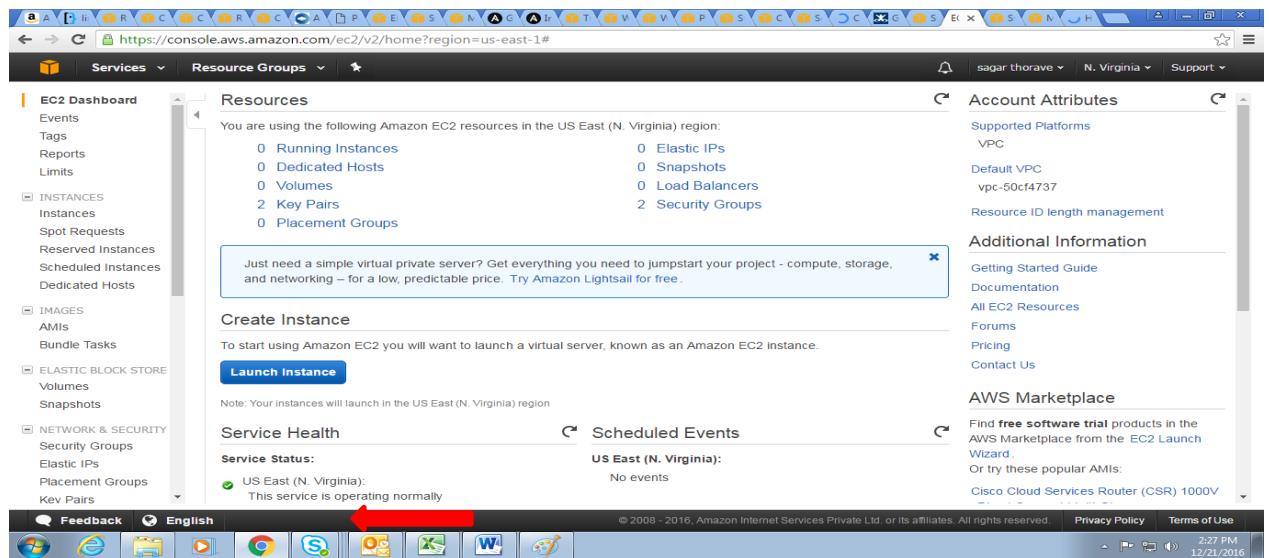
10.3 Procedure

10.3.1 Create an EC2 Instance

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

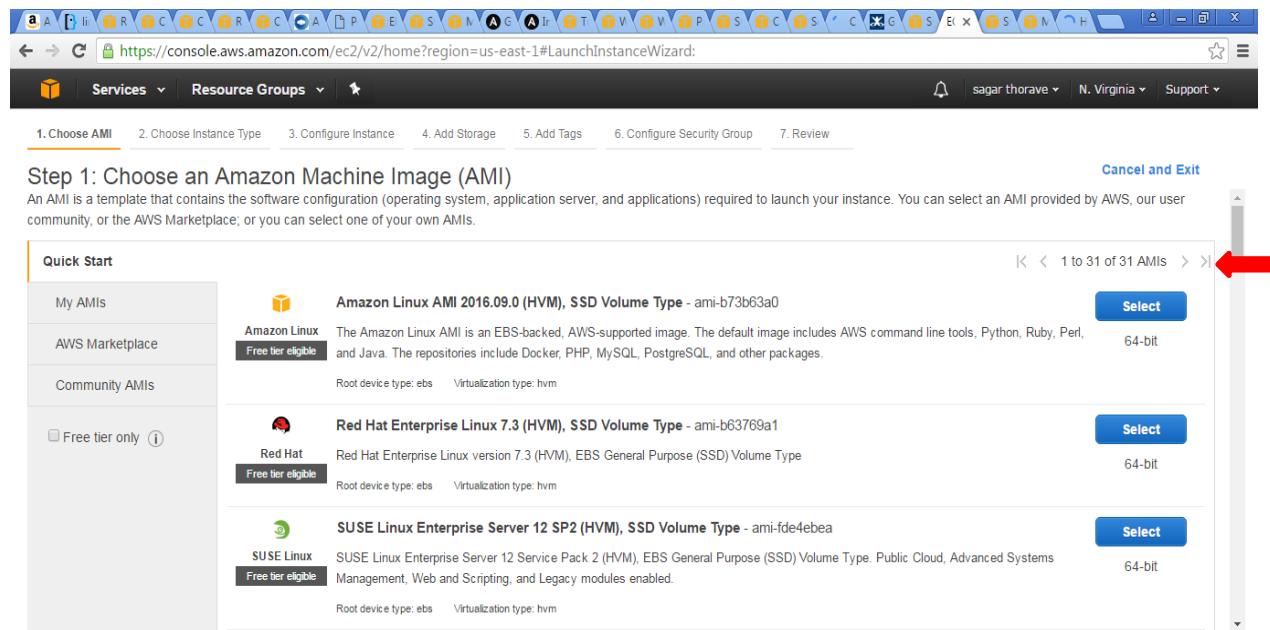


4 Click on Launch Instance button

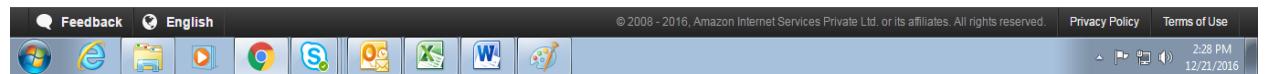


The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with various navigation links like EC2 Dashboard, Instances, Images, and Network & Security. The main area is titled 'Resources' and shows statistics for Running Instances, Dedicated Hosts, Volumes, Key Pairs, and Placement Groups. Below this is a callout box with a link to 'Amazon Lightsail for free'. Underneath is a 'Create Instance' section with a 'Launch Instance' button highlighted by a red arrow. To the right, there are sections for 'Service Health' (status: US East (N. Virginia) - operating normally), 'Scheduled Events' (no events), and 'AWS Marketplace' (with a note about finding trial products). The bottom of the screen shows the Windows taskbar.

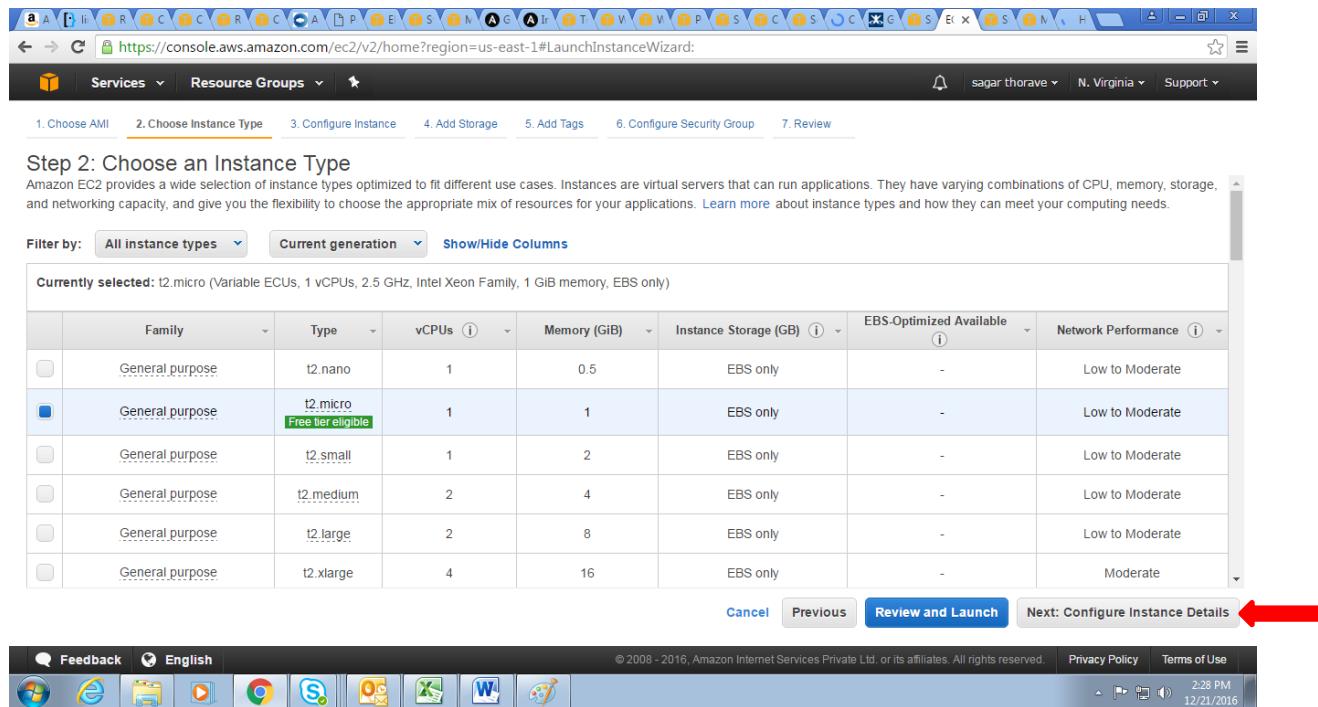
5 Choose an Amazon Machine Image (AMI), select Amazon Linux AMI 2016.9.0 (HVM) as it has preinstalled nfs-utils which is required to install for the following process



The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page of the Launch Instance Wizard. It lists several AMIs under 'Quick Start': Amazon Linux AMI 2016.9.0 (HVM), SSD Volume Type (ami-b73b63a0), Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type (ami-b63769a1), and SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type (ami-fde4ebea). Each entry includes a 'Select' button. A red arrow points to the 'Select' button for the Amazon Linux AMI entry. The bottom of the screen shows the Windows taskbar.



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



Step 2: Choose an Instance Type

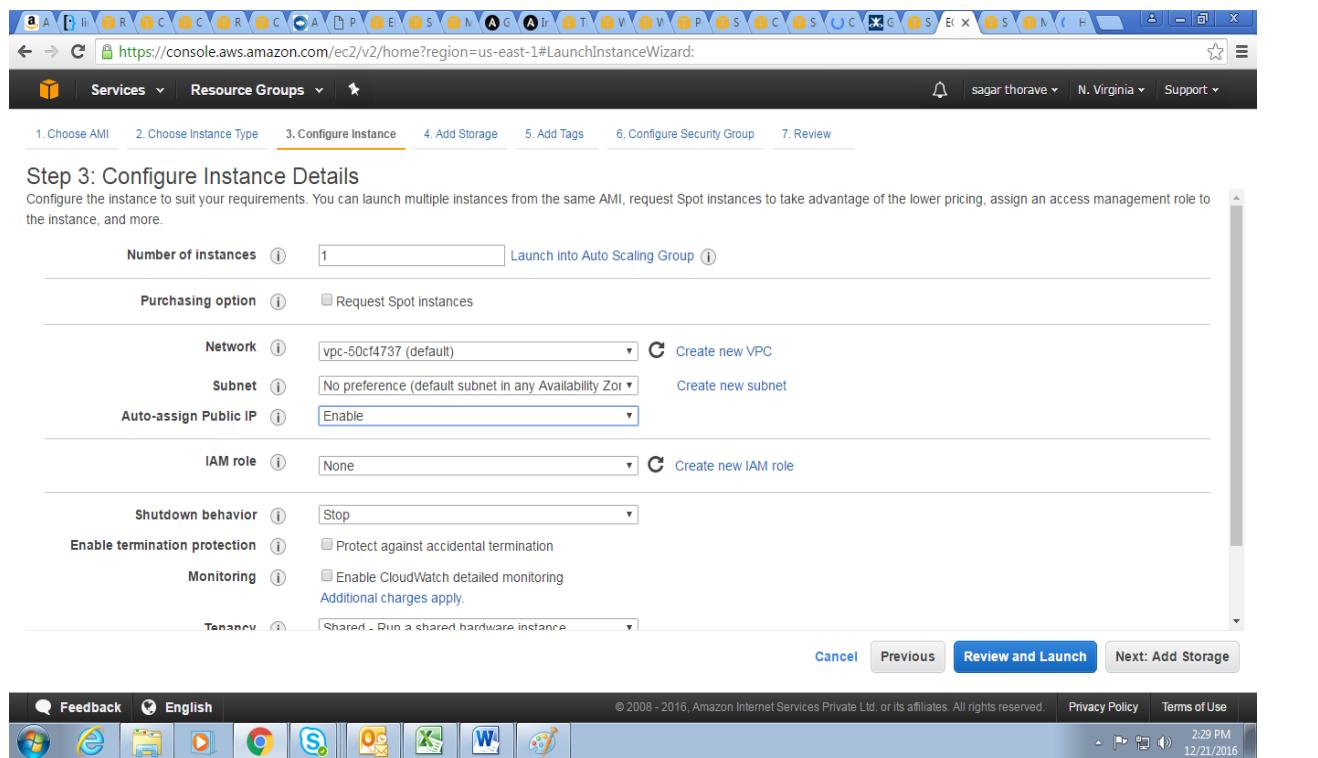
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

| Filter by: | All instance types | Current generation | Show/Hide Columns | | | | |
|---|--------------------|---|-------------------|--------------|-----------------------|-------------------------|---------------------|
| Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only) | | | | | | | |
| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
| <input type="checkbox"/> | General purpose | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate |
| <input checked="" type="checkbox"/> | General purpose | t2.micro <small>Free tier eligible</small> | 1 | 1 | EBS only | - | Low to Moderate |
| <input type="checkbox"/> | General purpose | t2.small | 1 | 2 | EBS only | - | Low to Moderate |
| <input type="checkbox"/> | General purpose | t2.medium | 2 | 4 | EBS only | - | Low to Moderate |
| <input type="checkbox"/> | General purpose | t2.large | 2 | 8 | EBS only | - | Low to Moderate |
| <input type="checkbox"/> | General purpose | t2.xlarge | 4 | 16 | EBS only | - | Moderate |

Review and Launch Next: Configure Instance Details

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

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



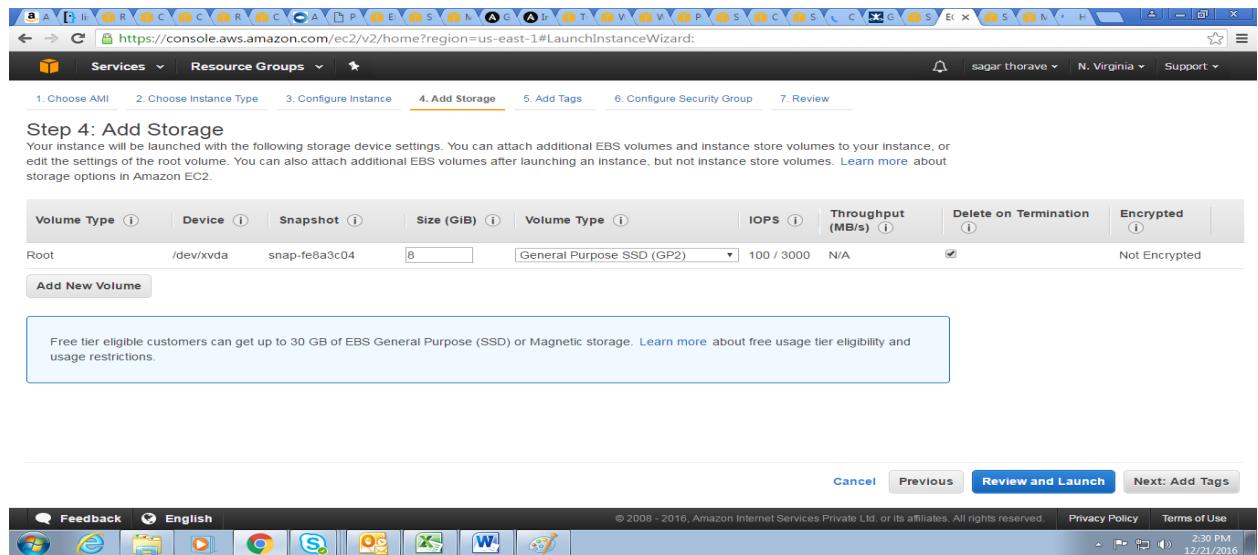
Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

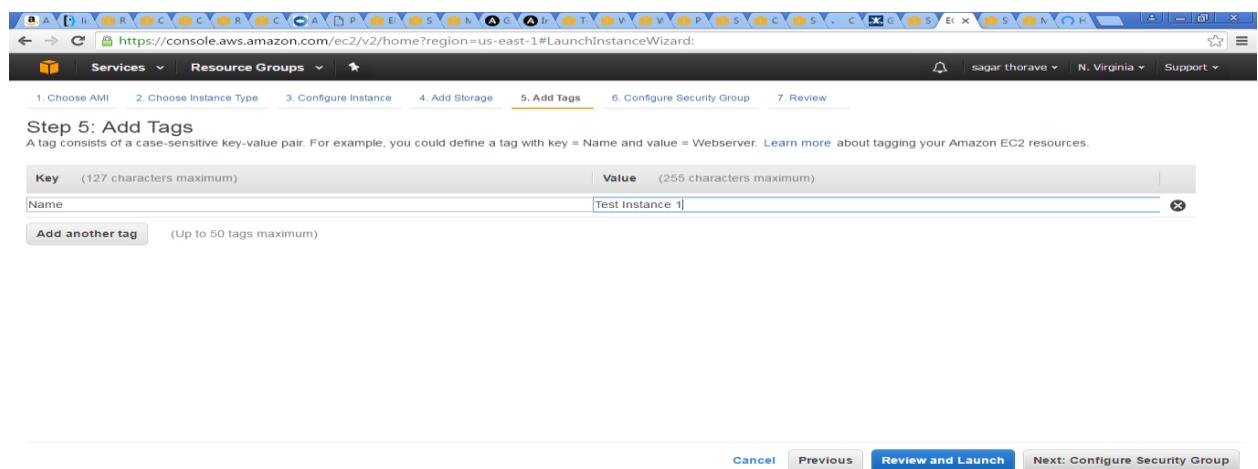
| | | |
|-------------------------------|--|--|
| Number of instances | <input type="text" value="1"/> | Launch into Auto Scaling Group |
| Purchasing option | <input type="checkbox"/> Request Spot instances | |
| Network | vpc-50cf4737 (default) | <input type="checkbox"/> Create new VPC |
| Subnet | No preference (default subnet in any Availability Zone) | <input type="checkbox"/> Create new subnet |
| Auto-assign Public IP | Enable | |
| IAM role | <input type="text" value="None"/> | <input type="checkbox"/> Create new IAM role |
| Shutdown behavior | Stop | |
| Enable termination protection | <input type="checkbox"/> Protect against accidental termination | |
| Monitoring | <input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small> | |
| Tenancy | Shared - Run a shared hardware instance | |

Review and Launch Next: Add Storage

8 Select the storage you want and choose Add Tags

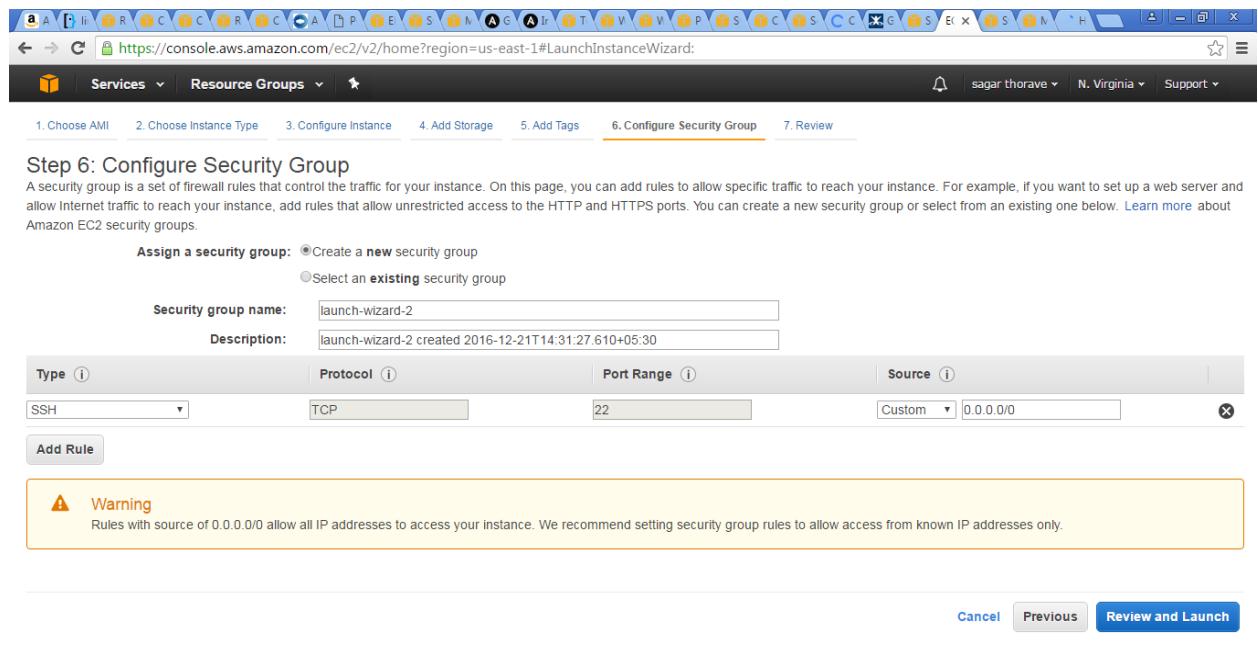


9 In Add Tags, give the name to an instance and choose Configure Security Group



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



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group:

- Create a **new** security group
- Select an **existing** security group

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2016-12-21T14:31:27.610+05:30

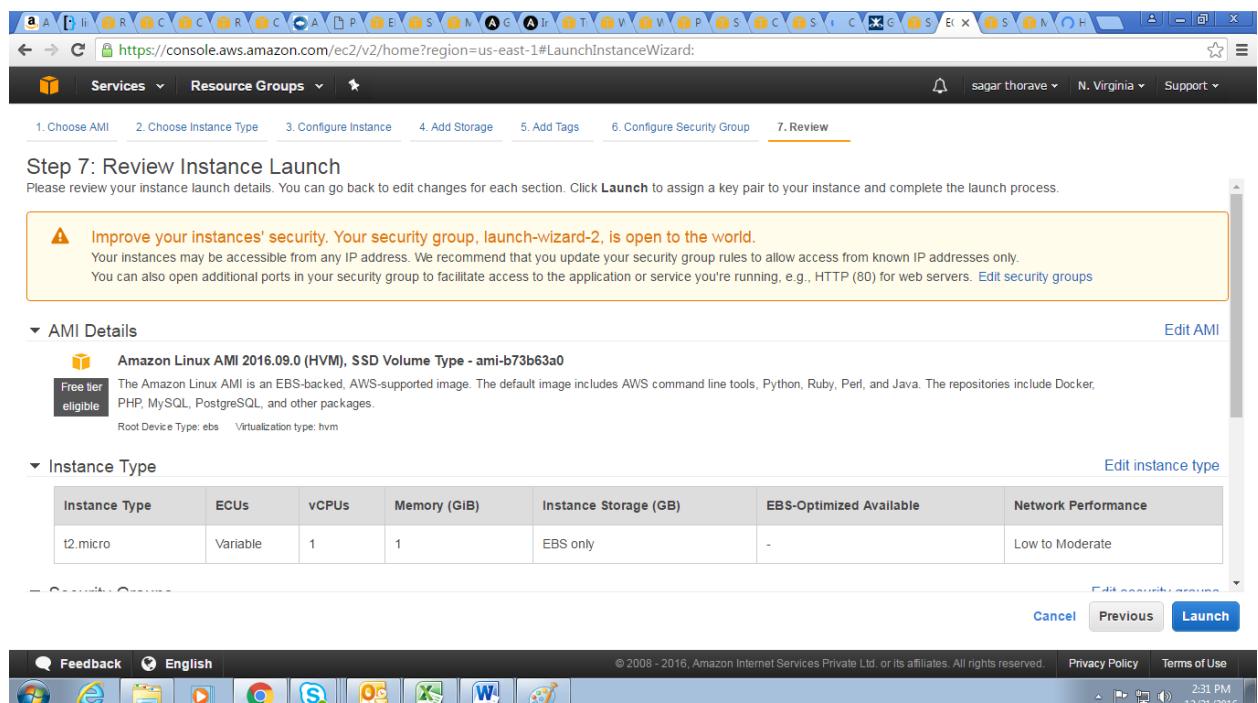
| Type | Protocol | Port Range | Source |
|------|----------|------------|------------------|
| SSH | TCP | 22 | Custom 0.0.0.0/0 |

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

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



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Amazon Linux AMI 2016.09.0 (HVM), SSD Volume Type - ami-b73b63a0

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.

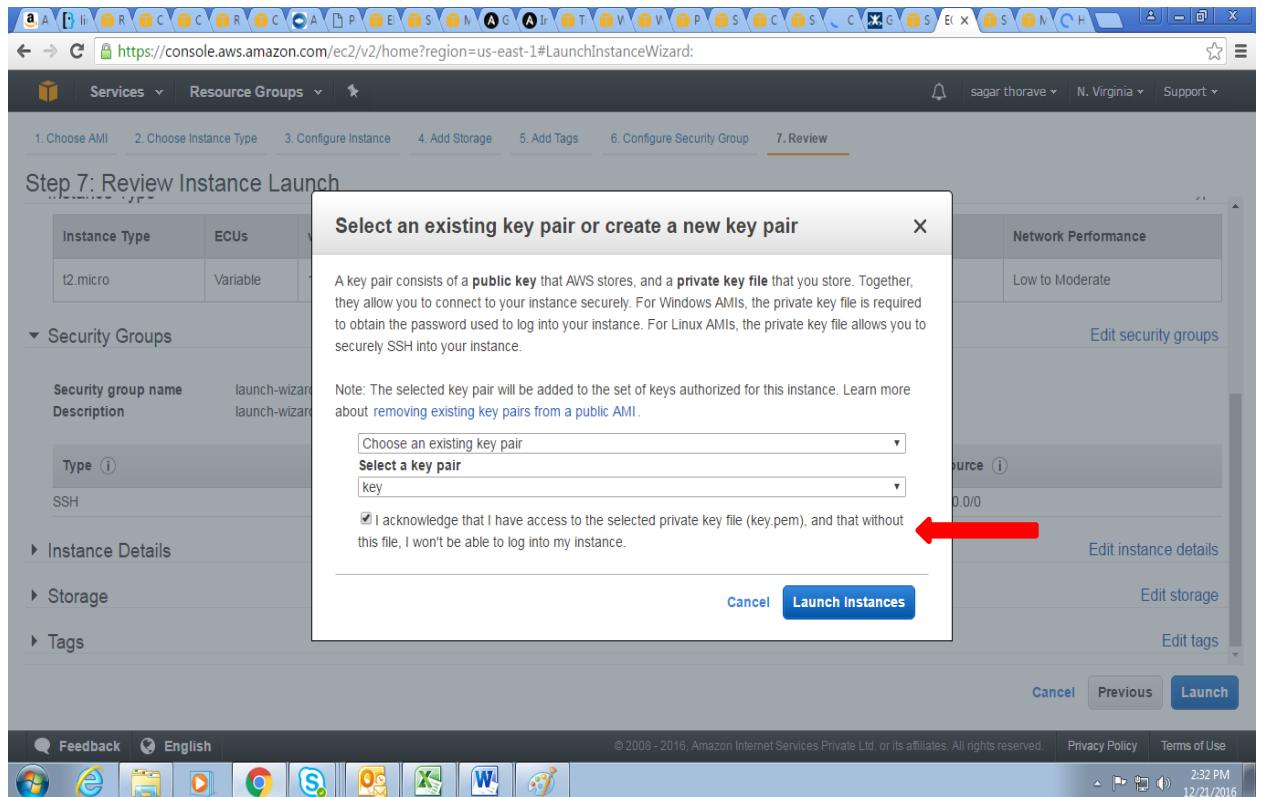
Root Device Type: ebs Virtualization type: hvm

Instance Type

| Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|----------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro | Variable | 1 | 1 | EBS only | - | Low to Moderate |

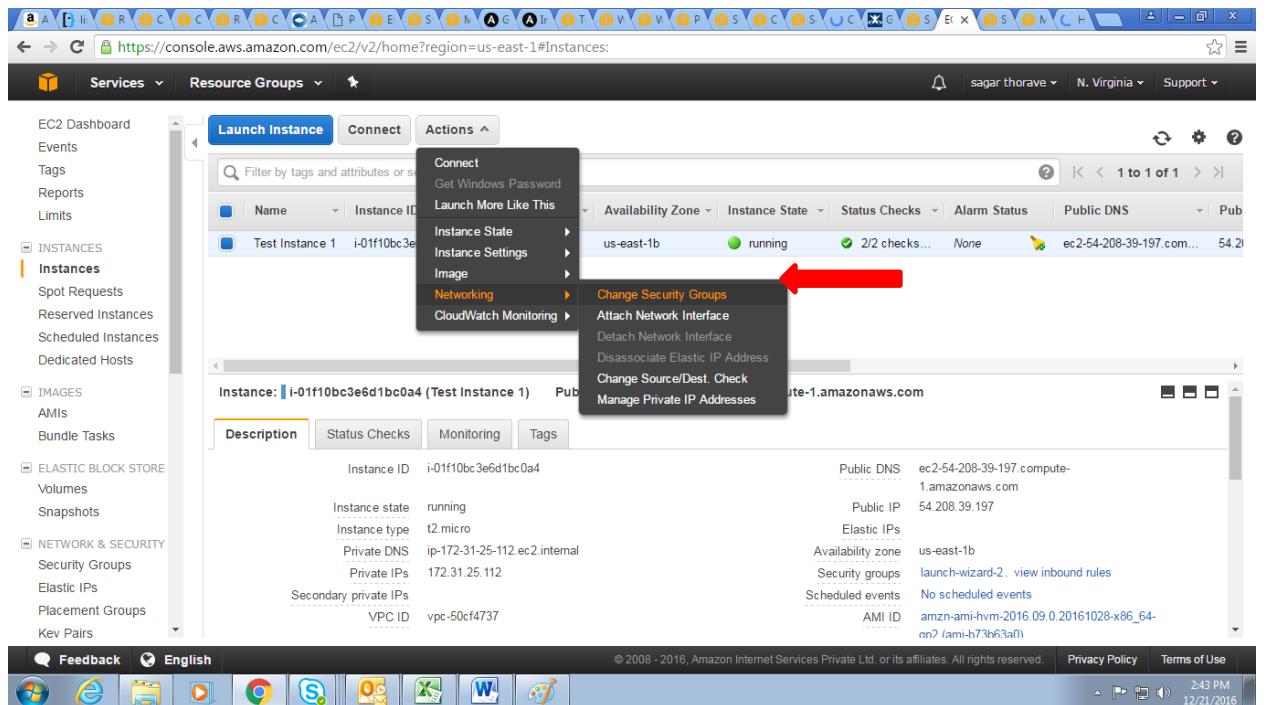
Launch

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

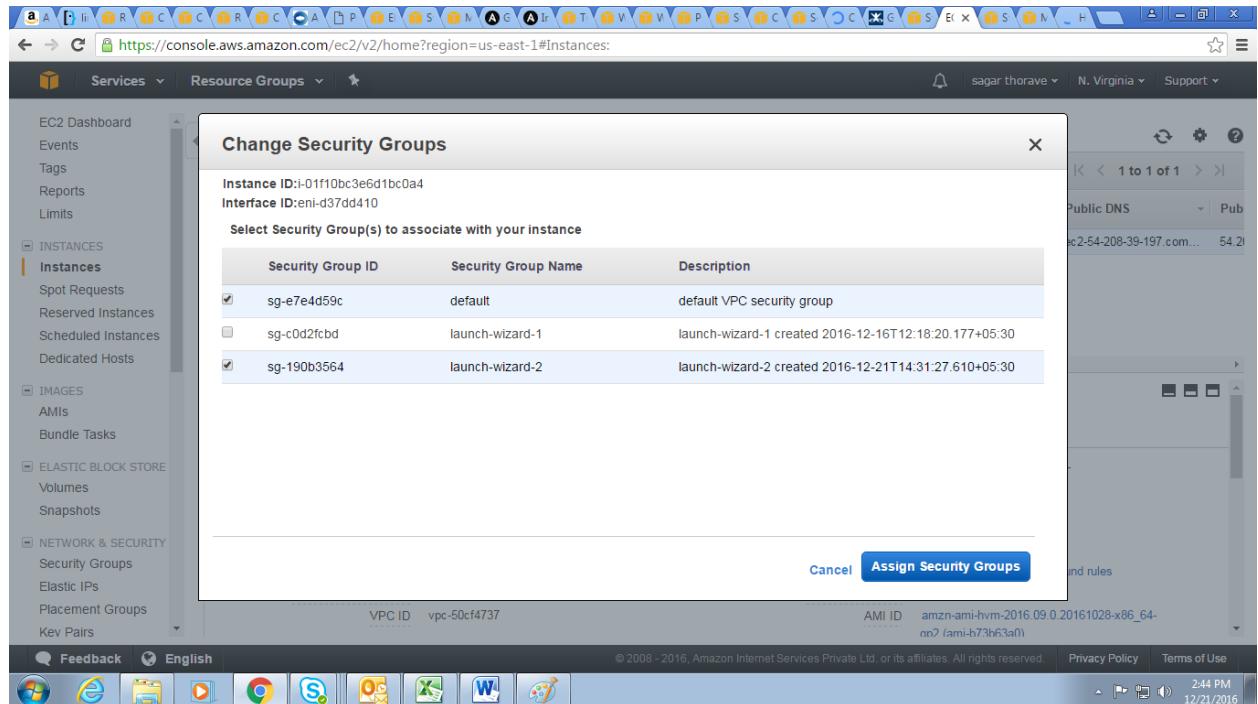


13 Choose the name of the instance you just created from the list, and then choose **Actions**.

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

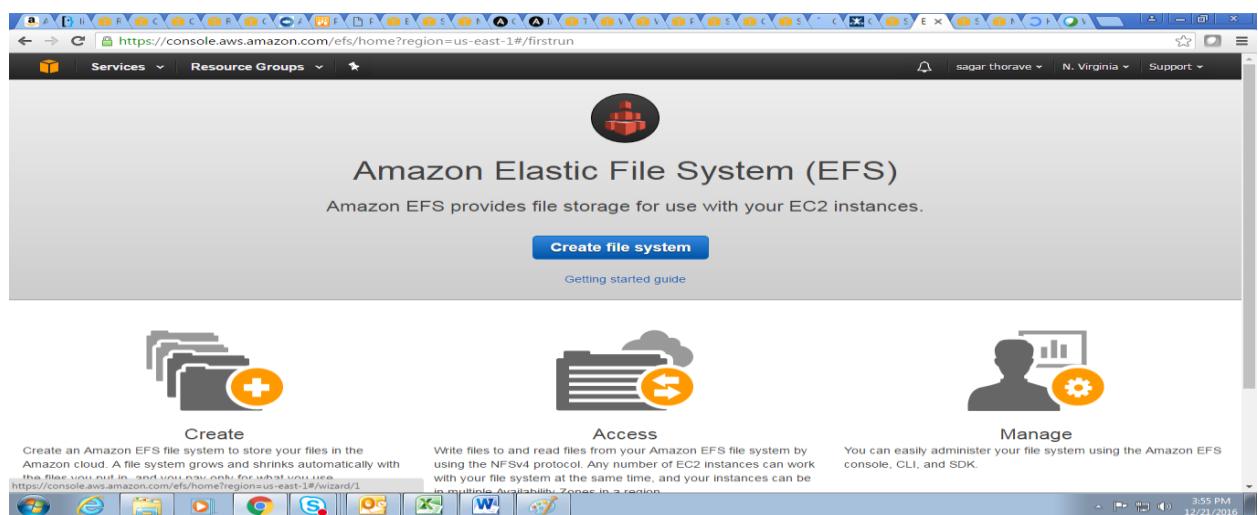


Select the check box next to the security group with the description **default VPC security group** And Choose **Assign Security Groups**. Make a note of the values listed next to VPC ID and Public DNS. You'll need those values later in this exercise.



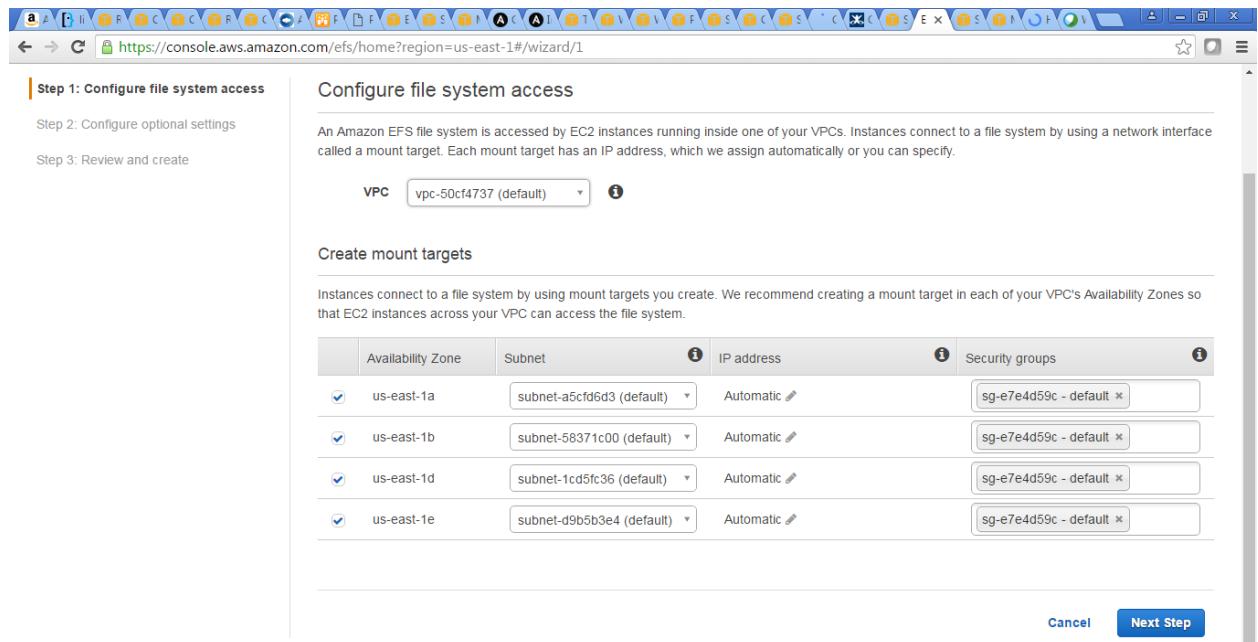
10.3.2 Create Amazon EFS File System

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**



Step 1: Configure file system access

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

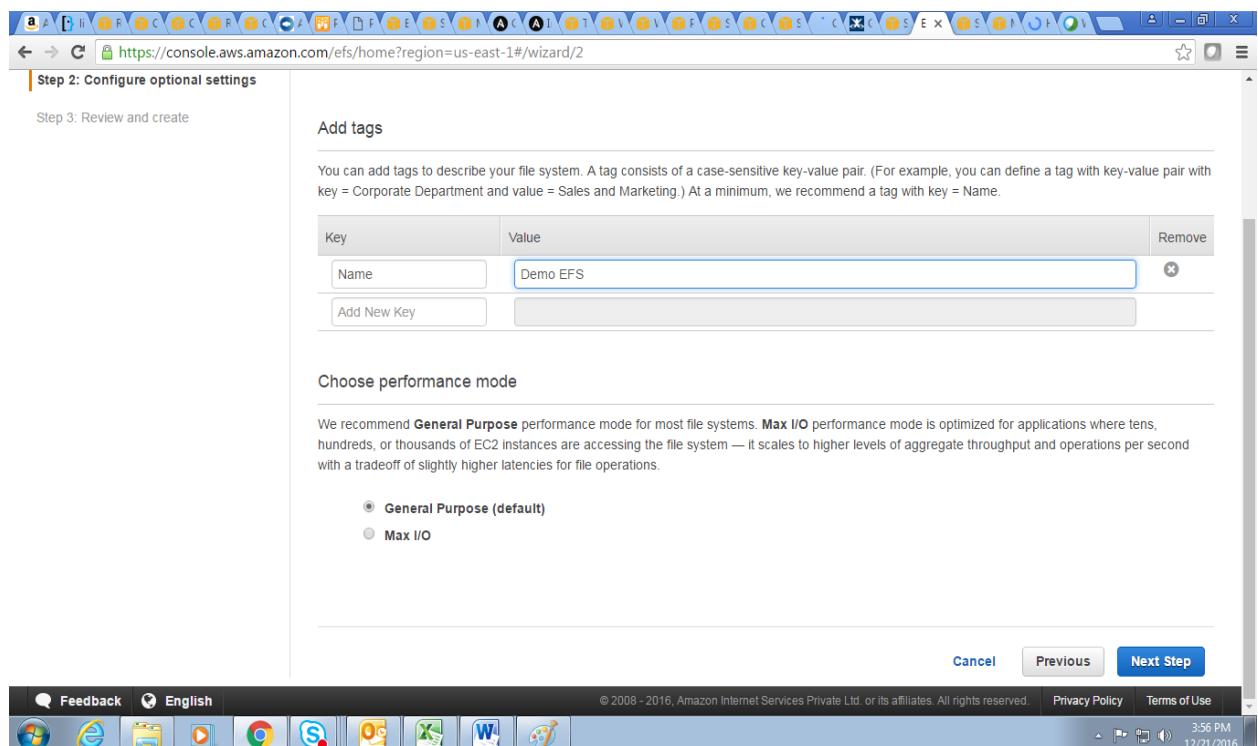
| VPC | vpc-50cf4737 (default) | | |
|-------------------|---------------------------|------------|-----------------------|
| Availability Zone | Subnet | IP address | Security groups |
| us-east-1a | subnet-a5cf6d3 (default) | Automatic | sg-e7e4d59c - default |
| us-east-1b | subnet-58371c00 (default) | Automatic | sg-e7e4d59c - default |
| us-east-1d | subnet-1cd5fc36 (default) | Automatic | sg-e7e4d59c - default |
| us-east-1e | subnet-d9b5b3e4 (default) | Automatic | sg-e7e4d59c - default |

Create mount targets

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

Cancel **Next Step**

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



Step 2: Configure optional settings

Add tags

You can add tags to describe your file system. A tag consists of a case-sensitive key-value pair. (For example, you can define a tag with key-value pair with key = Corporate Department and value = Sales and Marketing.) At a minimum, we recommend a tag with key = Name.

| Key | Value | Remove |
|-------------|----------|--------|
| Name | Demo EFS | X |
| Add New Key | | |

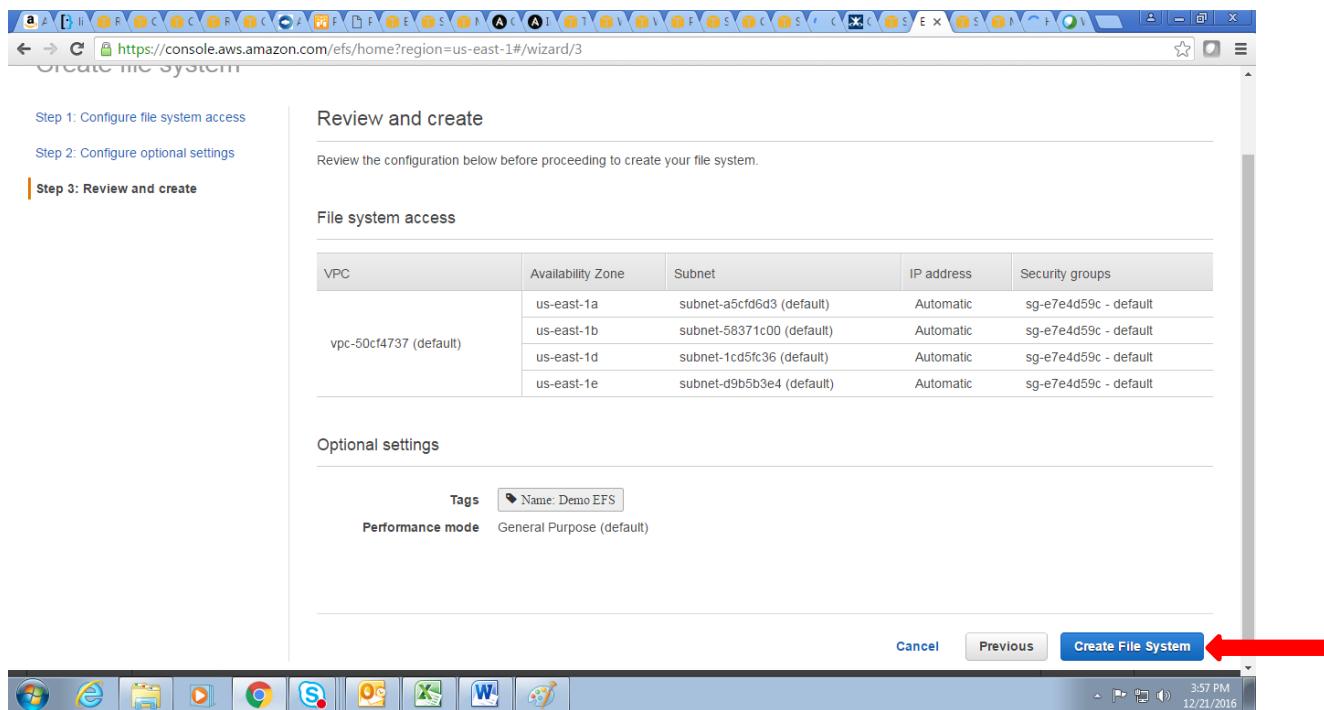
Choose performance mode

We recommend **General Purpose** performance mode for most file systems. **Max I/O** performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system — it scales to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

General Purpose (default)
 Max I/O

Cancel **Previous** **Next Step**

4. Choose **Create File System**



Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

Review and create

Review the configuration below before proceeding to create your file system.

File system access

| VPC | Availability Zone | Subnet | IP address | Security groups |
|------------------------|-------------------|---------------------------|------------|-----------------------|
| vpc-50cf4737 (default) | us-east-1a | subnet-a5cf6d3 (default) | Automatic | sg-e7e4d59c - default |
| | us-east-1b | subnet-58371c00 (default) | Automatic | sg-e7e4d59c - default |
| | us-east-1d | subnet-1cd5fc36 (default) | Automatic | sg-e7e4d59c - default |
| | us-east-1e | subnet-d9b5b3e4 (default) | Automatic | sg-e7e4d59c - default |

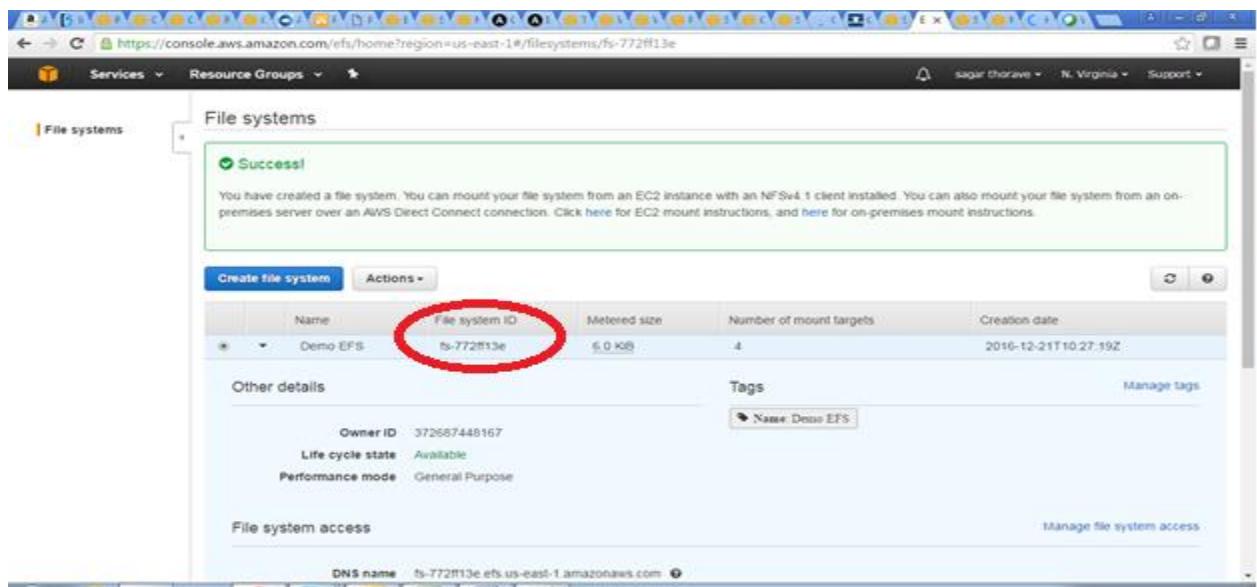
Optional settings

Tags: Name: Demo EFS

Performance mode: General Purpose (default)

Create File System (highlighted with a red arrow)

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.



File systems

Success!

You have created a file system. You can mount your file system from an EC2 instance with an NFSv4.1 client installed. You can also mount your file system from an on-premises server over an AWS Direct Connect connection. Click here for EC2 mount instructions, and here for on-premises mount instructions.

| Name | File system ID | Metered size | Number of mount targets | Creation date |
|----------|----------------|--------------|-------------------------|----------------------|
| Demo EFS | fs-772ff13e | 6.0 KiB | 4 | 2016-12-21T10:27:19Z |

Other details

Owner ID: 372687448167
 Life cycle state: Available
 Performance mode: General Purpose

Tags

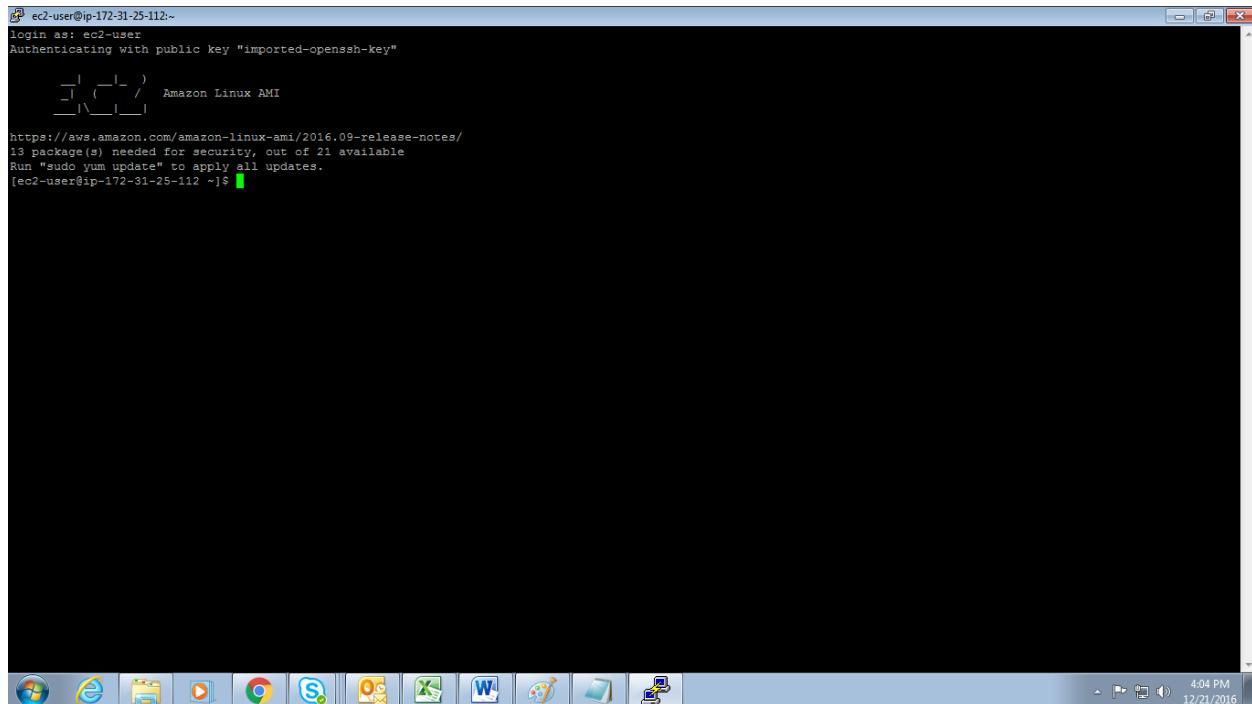
Name: Demo EFS

File system access

DNS name: fs-772ff13e.efs.us-east-1.amazonaws.com

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



```
ec2-user@ip-172-31-25-112:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
[ec2-user@ip-172-31-25-112 ~]$ https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/  
13 package(s) needed for security, out of 21 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-25-112 ~]$
```

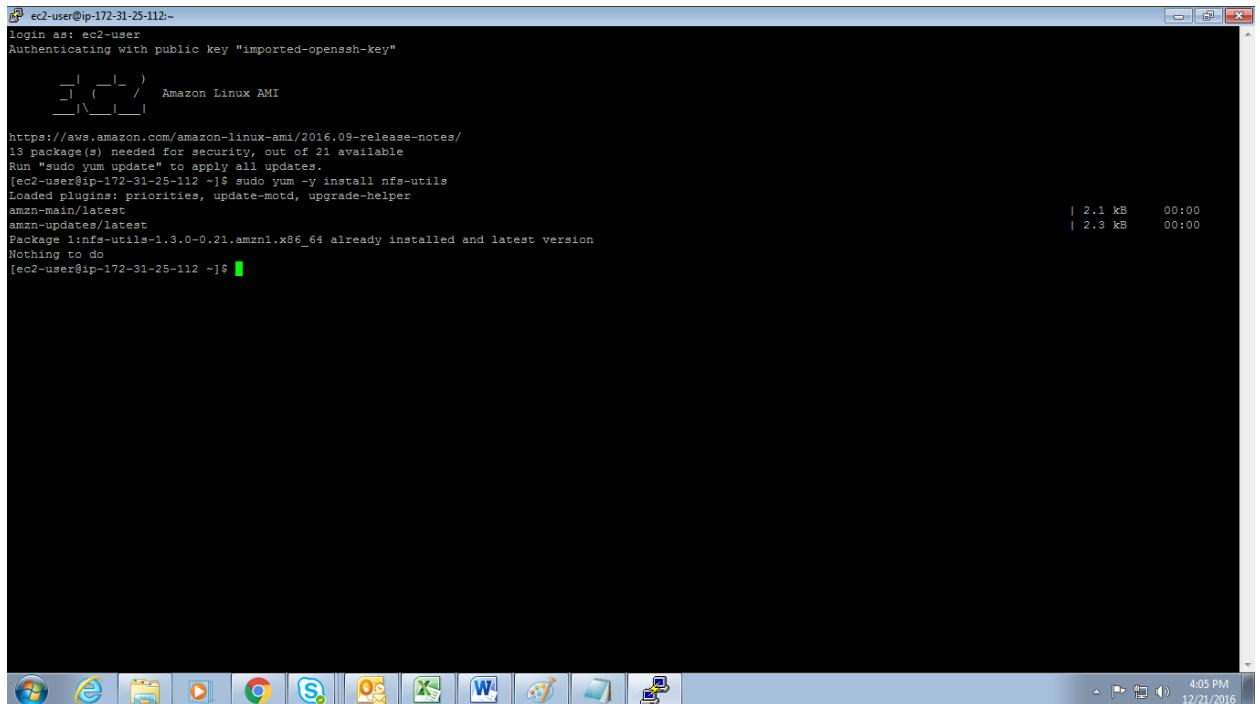
1. After you've connected, install the NFS client in my case it is preinstalled so it is showing Nothing to do

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
```

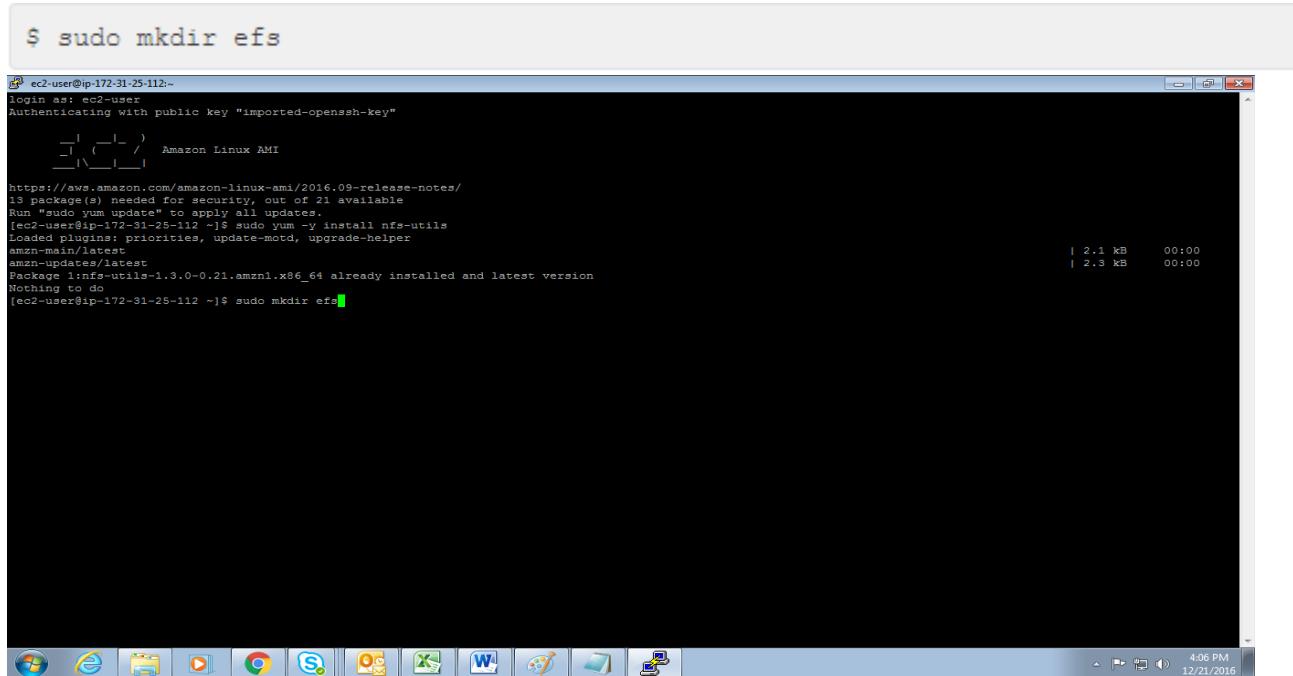
If you're using an Ubuntu AMI, install the NFS client with the following command

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



```
ec2-user@ip-172-31-25-112:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-25-112 ~]$ sudo yum update
https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
13 package(s) needed for security, out of 21 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-25-112 ~]$ sudo yum -y install nfs-utils
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main/latest
amzn-updates/latest
Package 1:nfs-utils-1.3.0-0.21.amzn1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-25-112 ~]$
```

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

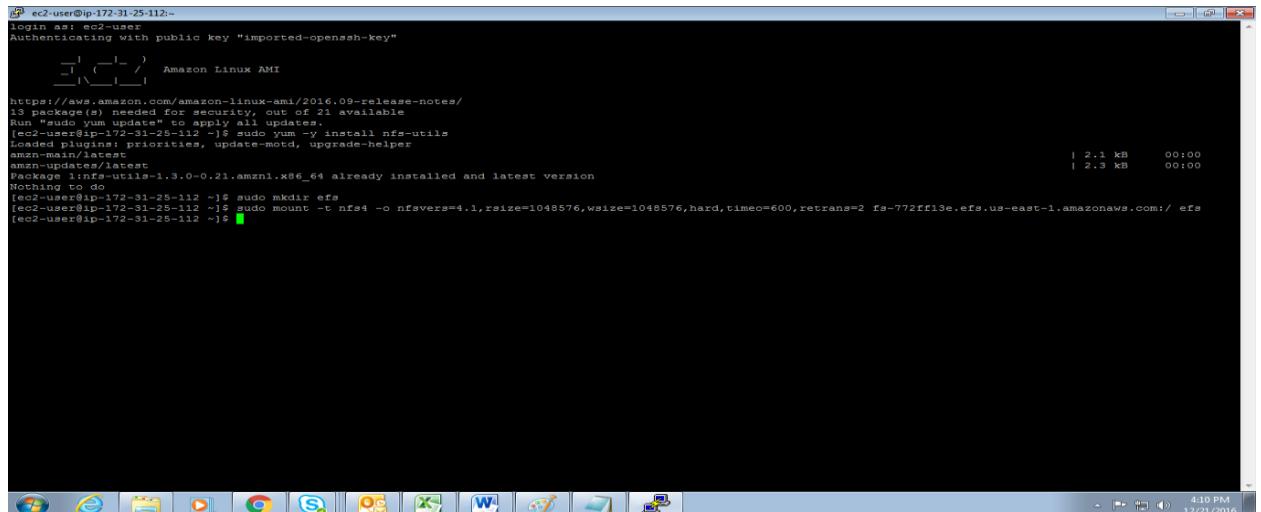


```
$ sudo mkdir efs
ec2-user@ip-172-31-25-112:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-25-112 ~]$ sudo yum update
https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
13 package(s) needed for security, out of 21 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-25-112 ~]$ sudo yum -y install nfs-utils
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main/latest
amzn-updates/latest
Package 1:nfs-utils-1.3.0-0.21.amzn1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-25-112 ~]$ 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,wszie=1048576,hard,timeo=600,retrans=2 file-system-
id.efs.aws-region.amazonaws.com:/ efs
```



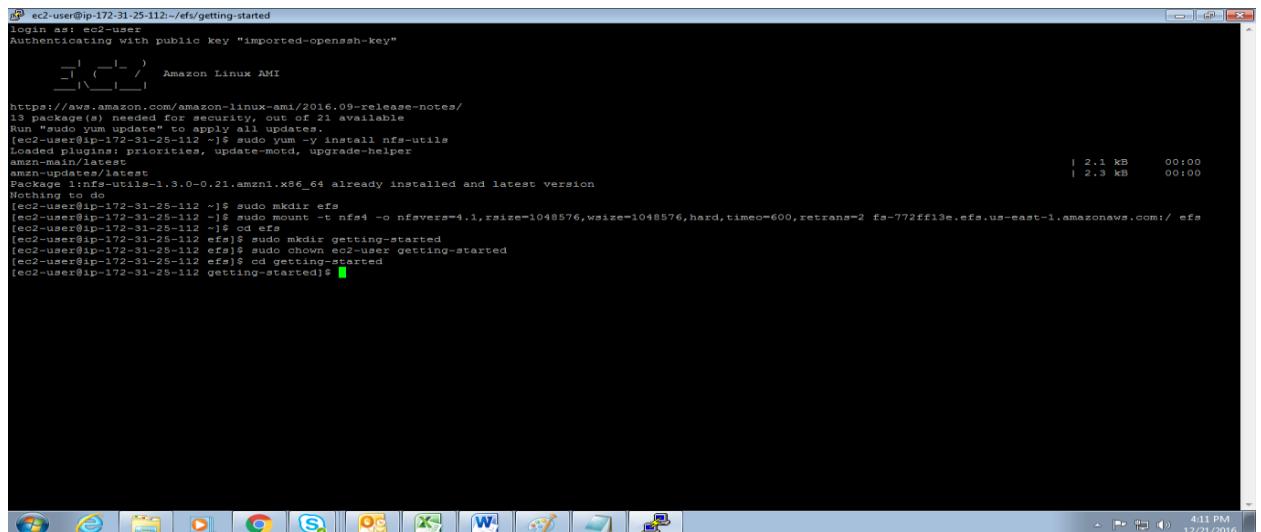
```
[ec2-user@ip-172-31-25-112: ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wszie=1048576,hard,timeo=600,retrans=2 file-system-
id.efs.aws-region.amazonaws.com:/ efs
[ec2-user@ip-172-31-25-112: ~]$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-25-112: ~]$ Amazon Linux AMI
[ec2-user@ip-172-31-25-112: ~]$ http://aws.amazon.com/amazon-linux-ami/2016.09/release-notes/
13 package(s) needed for security, out of 21 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-25-112: ~]$ sudo yum -y install nfs-utils
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main/latest
amzn-updates/latest
Package nfs-utils-1.3.0-0.21.amzn1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-25-112: ~]$ sudo mkdir efs
[ec2-user@ip-172-31-25-112: ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wszie=1048576,hard,timeo=600,retrans=2 fs-772ff13e.efs.us-east-1.amazonaws.com:/ efs
[ec2-user@ip-172-31-25-112: ~]$ [ec2-user@ip-172-31-25-112: ~]$
```

- 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

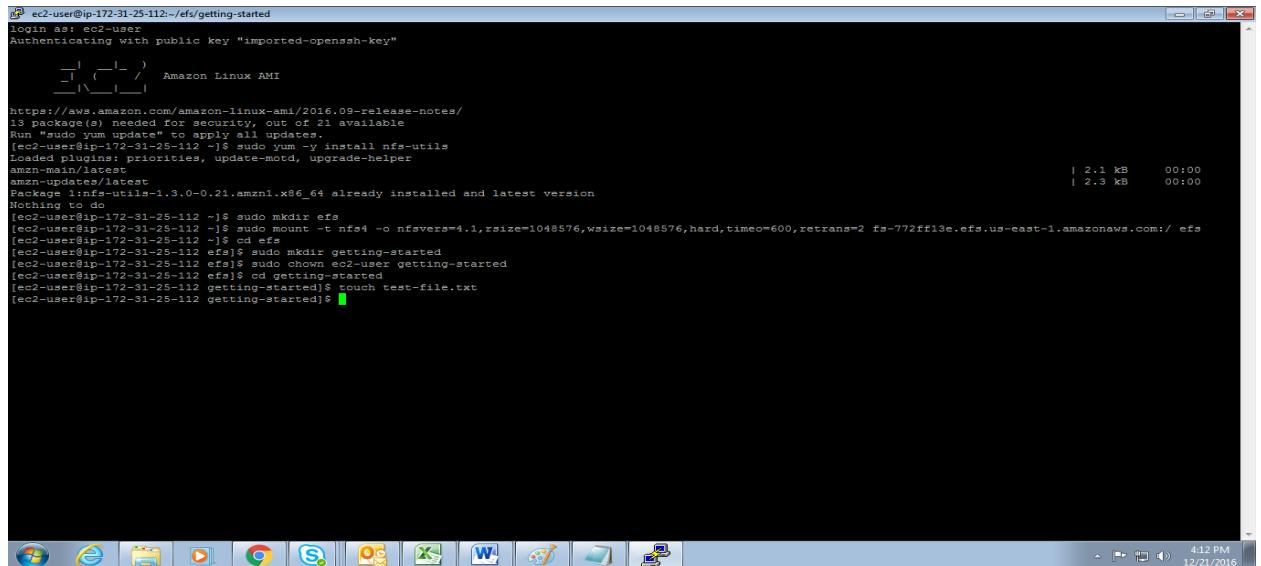
```
$ sudo mkdir getting-started
$ sudo chown ec2-user getting-started
$ cd getting-started
```



```
[ec2-user@ip-172-31-25-112: ~]$ sudo mkdir getting-started
[ec2-user@ip-172-31-25-112: ~]$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-25-112: ~]$ Amazon Linux AMI
[ec2-user@ip-172-31-25-112: ~]$ http://aws.amazon.com/amazon-linux-ami/2016.09/release-notes/
13 package(s) needed for security, out of 21 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-25-112: ~]$ sudo yum -y install nfs-utils
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main/latest
amzn-updates/latest
Package nfs-utils-1.3.0-0.21.amzn1.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-25-112: ~]$ sudo mkdir efs
[ec2-user@ip-172-31-25-112: ~]$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wszie=1048576,hard,timeo=600,retrans=2 fs-772ff13e.efs.us-east-1.amazonaws.com:/ efs
[ec2-user@ip-172-31-25-112: ~]$ [ec2-user@ip-172-31-25-112: ~]$ cd getting-started
[ec2-user@ip-172-31-25-112: getting-started]$ sudo mkdir getting-started
[ec2-user@ip-172-31-25-112: getting-started]$ sudo chown ec2-user:ec2-user getting-started
[ec2-user@ip-172-31-25-112: getting-started]$ cd getting-started
[ec2-user@ip-172-31-25-112: getting-started]$ [ec2-user@ip-172-31-25-112: getting-started]$
```

5. Create a text file with the following command

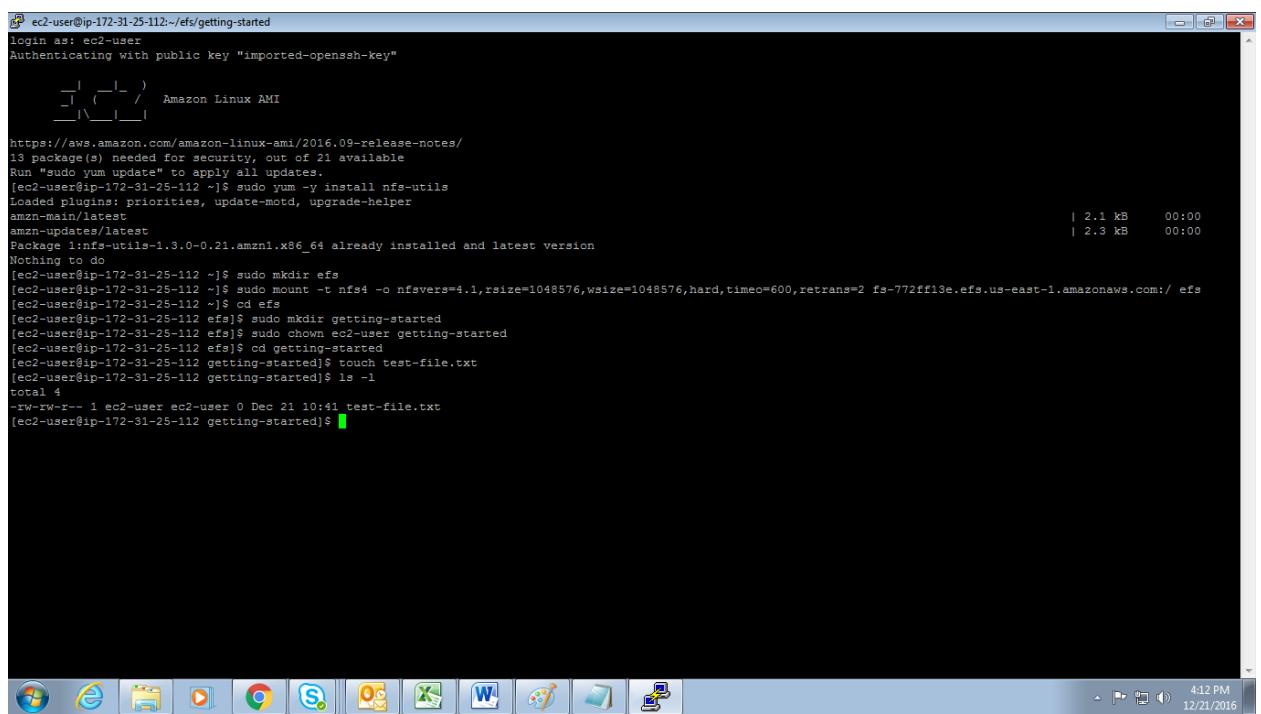
```
$ touch test-file.txt
```



```
[ec2-user@ip-172-31-25-112:~/efs/getting-started]
login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-25-112 ~]$ touch test-file.txt
[ec2-user@ip-172-31-25-112 ~]$
```

6. List the directory contents with the following command.

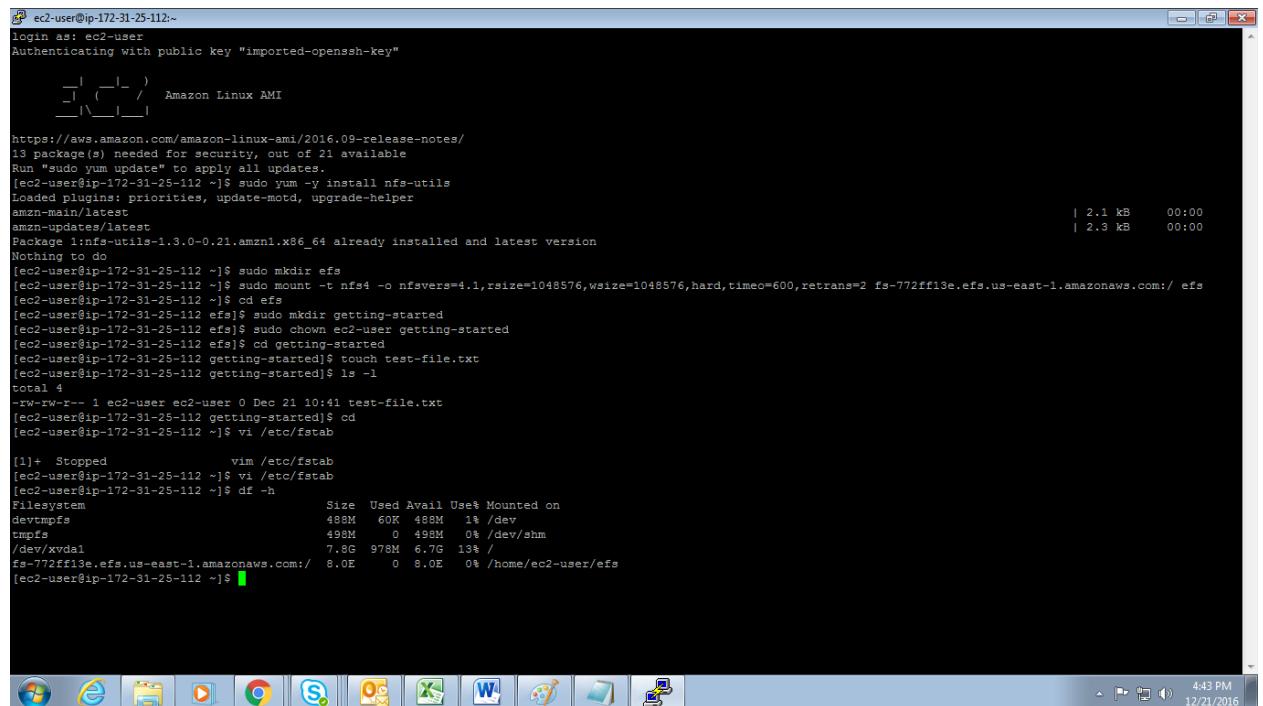
```
$ ls -al
```



```
[ec2-user@ip-172-31-25-112:~/efs/getting-started]
login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-25-112 ~]$ ls -al
total 4
-rw-r--r-- 1 ec2-user ec2-user 0 Dec 21 10:41 test-file.txt
[ec2-user@ip-172-31-25-112 ~]$
```

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
```



```
ec2-user@ip-172-31-25-112:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/devtmpfs        488M   60K  488M  1% /dev
tmpfs           498M     0  498M  0% /dev/shm
/dev/xvda1       7.8G  978M  6.7G 13% /
efs-772ff13e.efs.us-east-1.amazonaws.com/  8.0E    0  8.0E  0% /home/ec2-user/efs
[ec2-user@ip-172-31-25-112 ~]$
```

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{1,2})",
    "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", "hil.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",
        "MaxSize" : "2",
        "DesiredCapacity" : "1",
        "VPCZoneIdentifier" : "subnet-00000000,subnet-00000001"
    }
}
}
```

```

        "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" : {

```



```

        "           --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

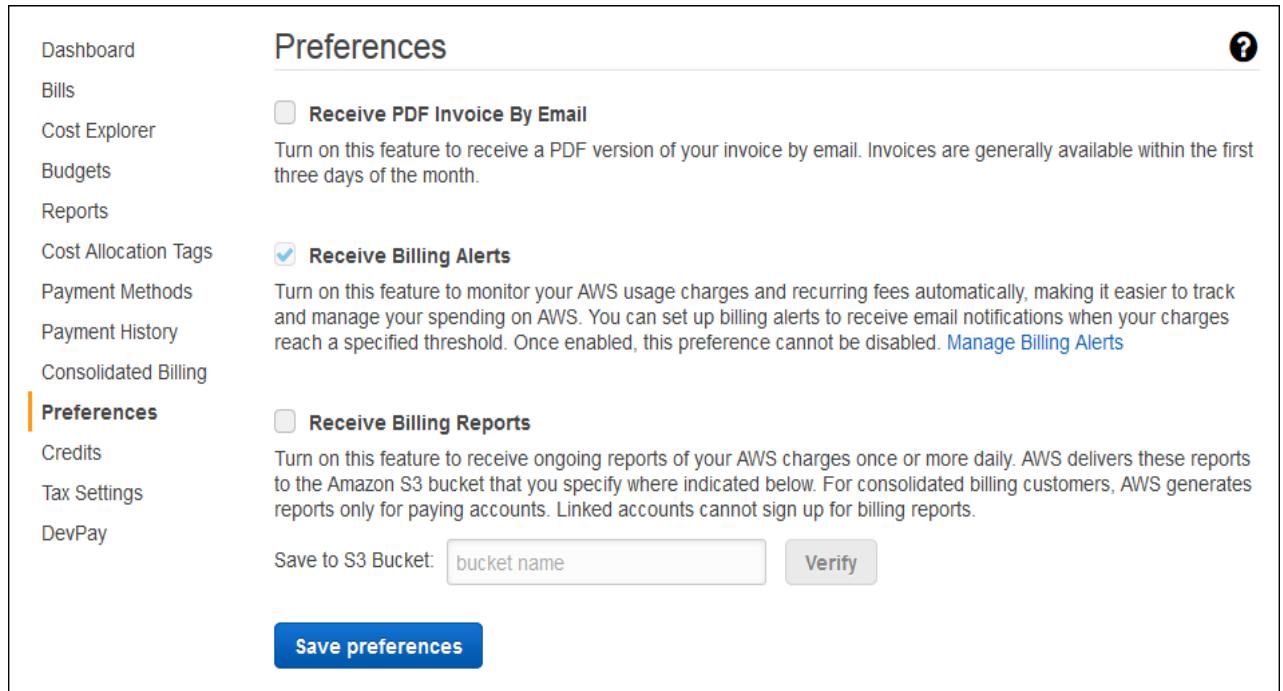
Step 5: Delete a Billing Alarm

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**.



Preferences

Receive PDF Invoice By Email
Turn on this feature to receive a PDF version of your invoice by email. Invoices are generally available within the first three days of the month.

Receive Billing Alerts
Turn on this feature to monitor your AWS usage charges and recurring fees automatically, making it easier to track and manage your spending on AWS. You can set up billing alerts to receive email notifications when your charges reach a specified threshold. Once enabled, this preference cannot be disabled. [Manage Billing Alerts](#)

Receive Billing Reports
Turn on this feature to receive ongoing reports of your AWS charges once or more daily. AWS delivers these reports to the Amazon S3 bucket that you specify where indicated below. For consolidated billing customers, AWS generates reports only for paying accounts. Linked accounts cannot sign up for billing reports.

Save to S3 Bucket:

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.

Create Alarm

Billing Alarm

You can create a billing alarm to receive e-mail alerts when your AWS charges exceed a threshold you choose. Simply:

1. Enter a spending threshold
2. Provide an email address
3. Check your inbox for a confirmation email and click the link provided

When my total AWS charges for the month

exceed: \$ USD

send a notification to: [New list](#)

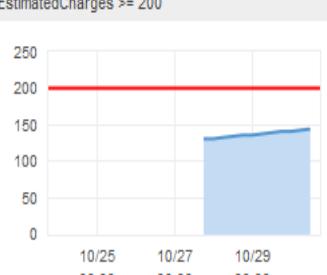
Reminder: for each address you add, you will receive an email from AWS with the subject "AWS Notification - Subscription Confirmation". Click the link provided in the message to confirm that AWS may deliver alerts to that address.

[showing simple options](#) | [show advanced](#)

Alarm Preview

This alarm will trigger when the blue line goes above the red line

EstimatedCharges >= 200



| Date | EstimatedCharges |
|-------------|------------------|
| 10/25 00:00 | 100 |
| 10/27 00:00 | 120 |
| 10/29 00:00 | 150 |

Cancel
Previous
Next
Create Alarm

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

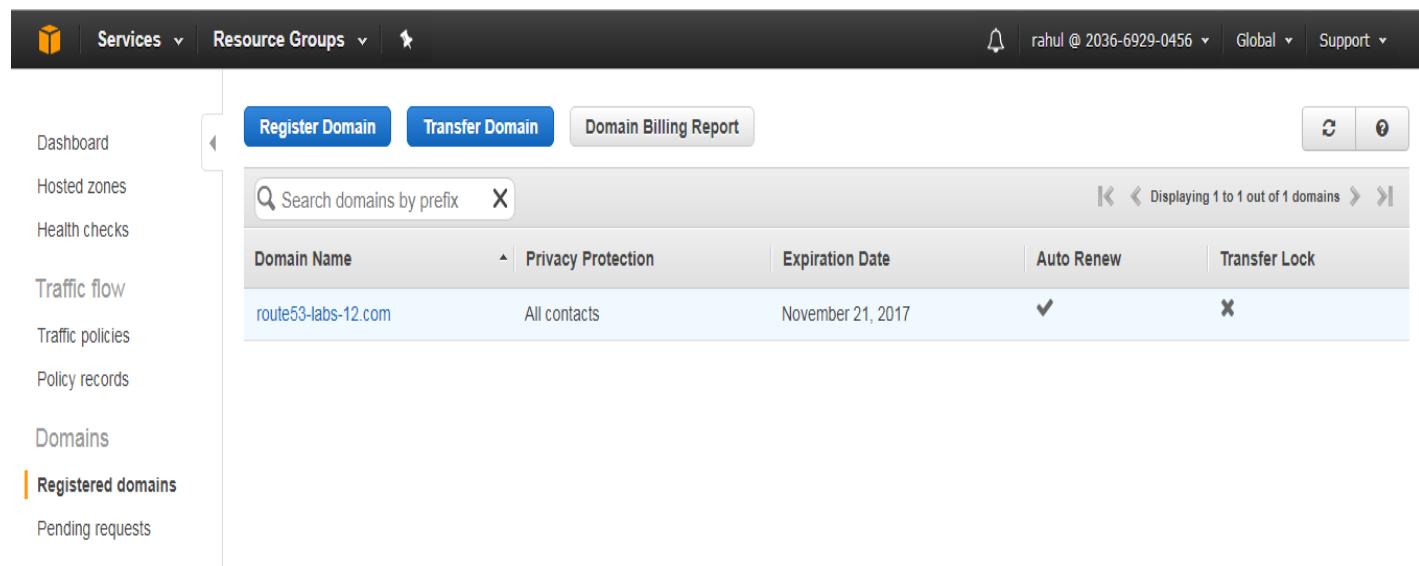
13.2 Assumptions

- 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**.



The screenshot shows the AWS Route 53 service console. The top navigation bar includes 'Services', 'Resource Groups', and user information ('rahul @ 2036-6929-0456'). The left sidebar lists navigation options: Dashboard, Hosted zones, Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains (which is selected and highlighted in orange), and Pending requests. The main content area displays a table of registered domains. The table has columns for 'Domain Name', 'Privacy Protection', 'Expiration Date', 'Auto Renew', and 'Transfer Lock'. One domain, 'route53-labs-12.com', is listed with its details: 'All contacts' under Privacy Protection, 'November 21, 2017' under Expiration Date, a checked checkbox under Auto Renew, and an unchecked checkbox under Transfer Lock. A search bar at the top of the main content area allows searching by domain prefix.

| Domain Name | Privacy Protection | Expiration Date | Auto Renew | Transfer Lock |
|---------------------|--------------------|-------------------|------------|---------------|
| route53-labs-12.com | All contacts | November 21, 2017 | ✓ | ✗ |

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

Services | Resource Groups | ★

Registered domains > route53-labs-12.com

Edit contacts Manage DNS Delete domain

| | | | | | |
|----------------------|---------------------------------------|--------------------------------|--|----------------------|--|
| Domain | route53-labs-12.com | Transfer lock | Disabled (enable) | Name servers | ns-508.awsdns-63.com ns-875.awsdns-45.net ns-1034.awsdns-01.org ns-1766.awsdns-28.co.uk Add or edit name servers |
| Registered on | 2015-11-21 | Authorization code | Generate | DNSSEC status | Error listing DNSSEC keys. |
| Expires on | 2017-11-21 (extend) | Domain name status code | ok | | |
| Auto renew | Enabled (disable) | Tag | View and manage tags for your domains using Tag editor | | |

Registrant contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

Administrative contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

Technical contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

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

Services | Resource Groups | ★

Registered domains > route53-labs-12.com

Your request for update nameserver was successfully submitted. You will receive an email when it is done.

Edit contacts Manage DNS

Edit Name Servers for route53-labs-12.com

Name servers

- ns-508.awsdns-63.com
- ns-875.awsdns-45.net
- ns-1034.awsdns-01.org
- ns-1766.awsdns-28.co.uk
-

Cancel **Update**

| | | | |
|----------------------|---------------------|----------------------|--|
| Domain | route53-labs-12.com | Name servers | ns-508.awsdns-63.com ns-875.awsdns-45.net ns-1034.awsdns-01.org ns-1766.awsdns-28.co.uk Add or edit name servers |
| Registered on | 2015-11-21 | DNSSEC status | Error listing DNSSEC keys. |
| Expires on | 2017-11-21 | | |
| Auto renew | Enabled (disable) | | |

Registrant contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

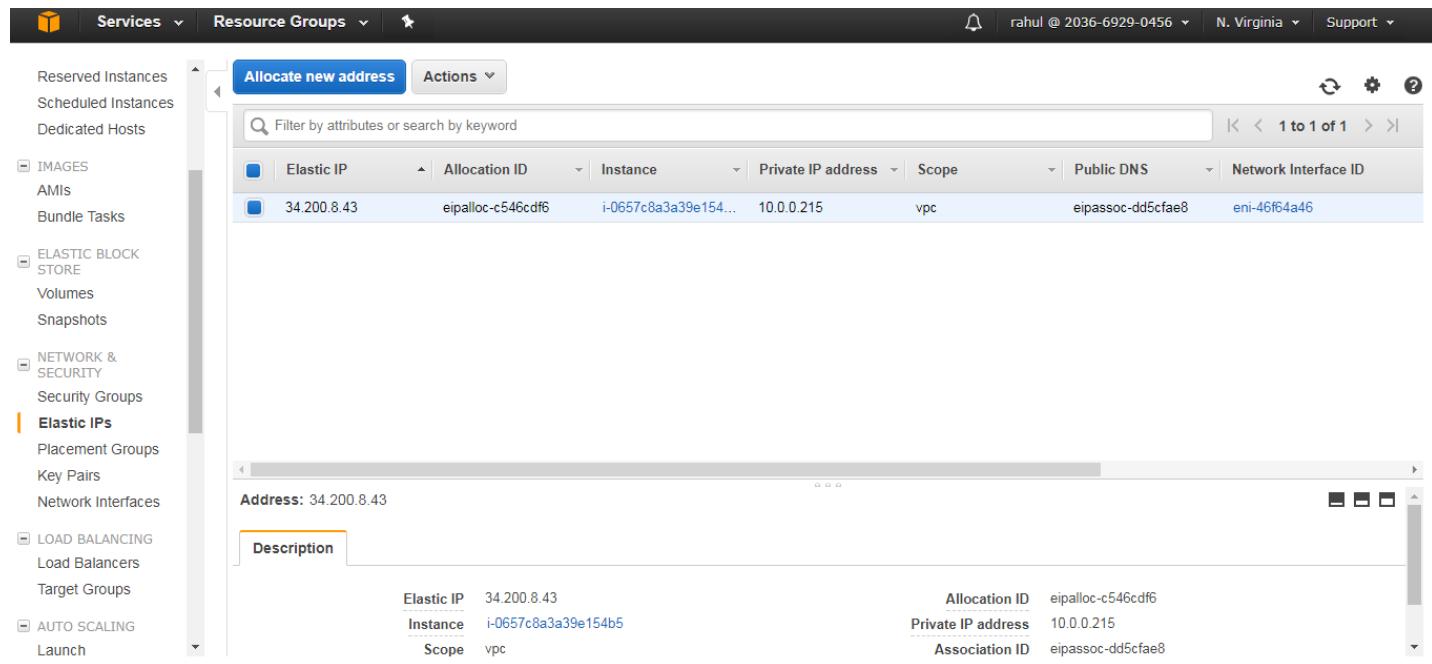
Administrative contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

Technical contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

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



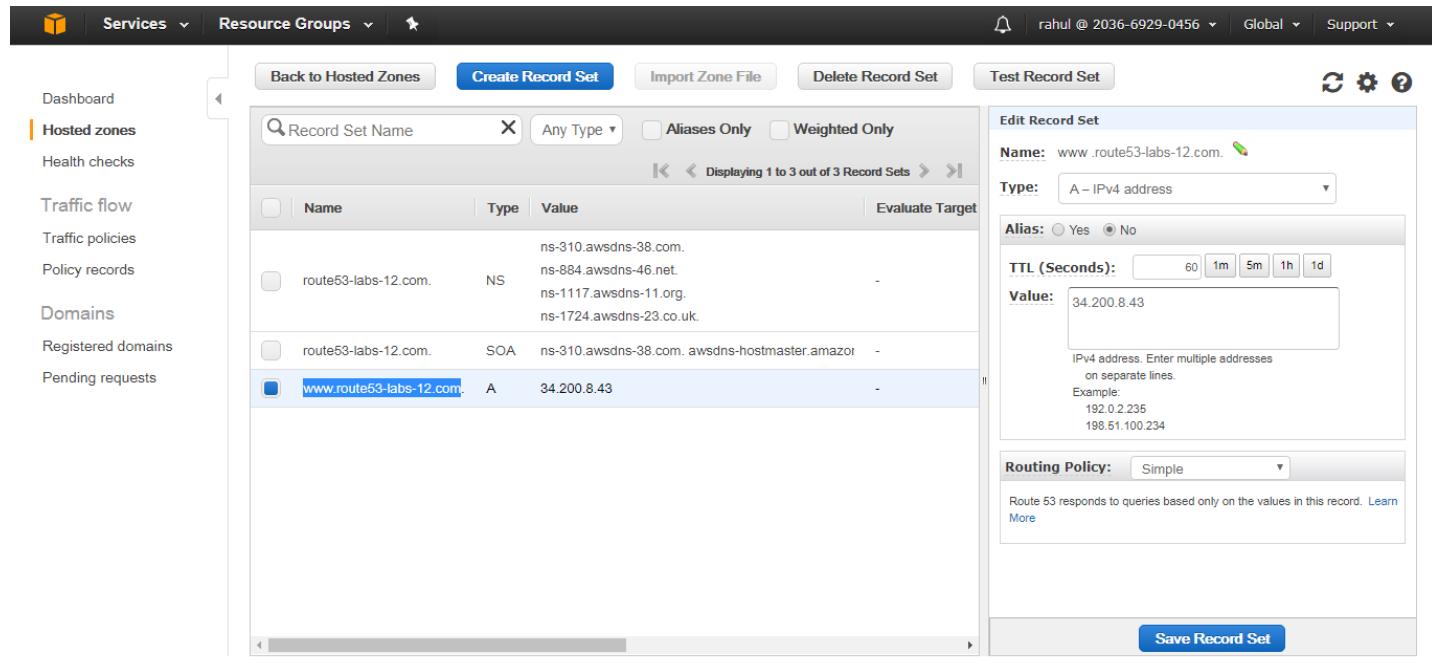
The screenshot shows the AWS EC2 console under the 'Elastic IPs' section. A single elastic IP allocation is listed:

| Allocation ID | Instance | Private IP address | Scope | Public DNS | Network Interface ID |
|-------------------|---------------------|--------------------|-------|-------------------|----------------------|
| eipalloc-c546cdf6 | i-0657c8a3a39e154b5 | 10.0.0.215 | vpc | eipassoc-dd5cf8e8 | eni-46f64a46 |

Below the table, specific details for the allocation are shown:

- Address:** 34.200.8.43
- Description:** (empty)
- Elastic IP:** 34.200.8.43
- Instance:** i-0657c8a3a39e154b5
- Scope:** vpc
- Allocation ID:** eipalloc-c546cdf6
- Private IP address:** 10.0.0.215
- Association ID:** eipassoc-dd5cf8e8

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



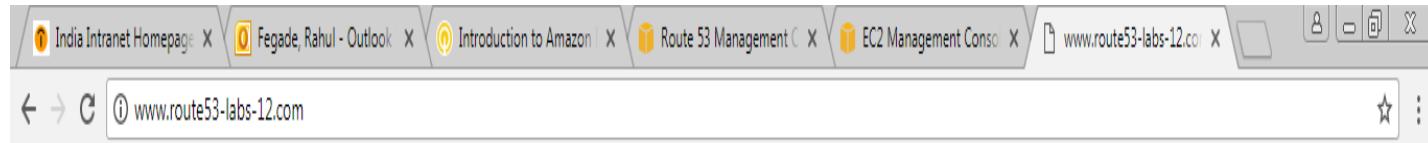
The screenshot shows the AWS Route 53 console under the 'Hosted zones' section. An A record is selected for editing:

| Name | Type | Value |
|---|------|---|
| ns-310.awsdns-38.com. | NS | ns-884.awsdns-46.net. ns-1117.awsdns-11.org. ns-1724.awsdns-23.co.uk. |
| route53-labs-12.com. | SOA | ns-310.awsdns-38.com. awsdns-hostmaster.amazon. |
| www.route53-labs-12.com | A | 34.200.8.43 |

The right panel displays the record settings:

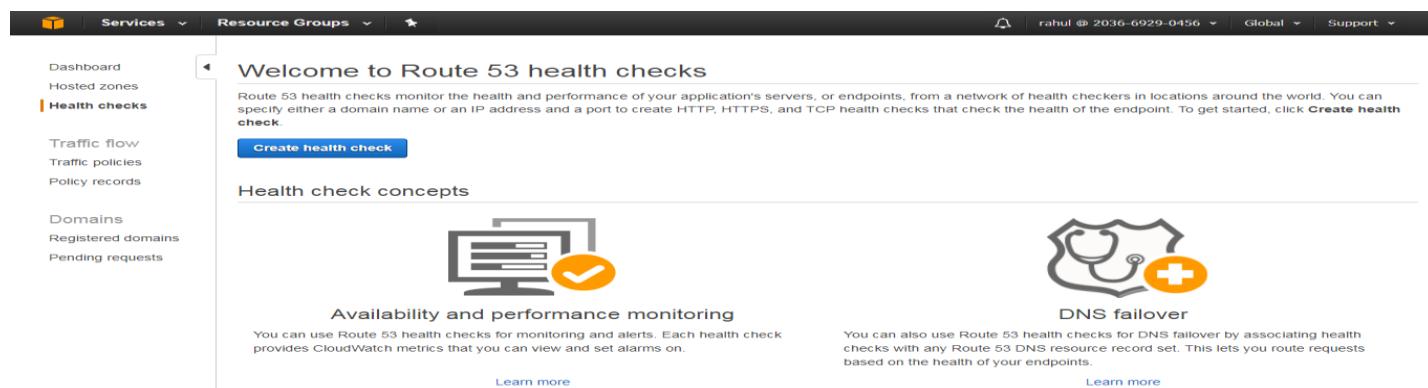
- Name:** www.route53-labs-12.com.
- Type:** A – IPv4 address
- Alias:** No
- TTL (Seconds):** 60, 1m, 5m, 1h, 1d
- Value:** 34.200.8.43
- Routing Policy:** Simple

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



Hello! This is your EC2 web server. It's nice to see you

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



Welcome to Route 53 health checks

Route 53 health checks monitor the health and performance of your application's servers, or endpoints, from a network of health checkers in locations around the world. You can specify either a domain name or an IP address and a port to create HTTP, HTTPS, and TCP health checks that check the health of the endpoint. To get started, click [Create health check](#).

[Create health check](#)

Health check concepts

Availability and performance monitoring

You can use Route 53 health checks for monitoring and alerts. Each health check provides CloudWatch metrics that you can view and set alarms on.

[Learn more](#)

DNS failover

You can also use Route 53 health checks for DNS failover by associating health checks with any Route 53 DNS resource record set. This lets you route requests based on the health of your endpoints.

[Learn more](#)

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 

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 

IP address * 

Host name 

Port * 

Path 

 Advanced configuration

Create health check

Step 1: Configure health check

Step 2: Get notified when health check fails

Get notified when health check fails

If you want CloudWatch to send you an Amazon SNS notification, such as an email, when the status of the health check changes to unhealthy, create an alarm and specify where to send notifications.

Create alarm Yes No

CloudWatch sends you an Amazon SNS notification whenever the status of this health check is unhealthy for one minute.

Send notification to Existing SNS topic New SNS topic

Topic name * ServerHCFailed

Recipient email addresses * rahul.fegade@capgemini.com

Separate multiple addresses with a comma, a semicolon, or a space

* Required

Cancel Previous Create health check

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

Amazon S3

Identify optimal storage classes with S3 Analytics - Storage Class Analysis. [Learn More](#)

Documentation

Switch to the old console Discover the new console Quick tips

Create bucket Delete bucket Empty bucket

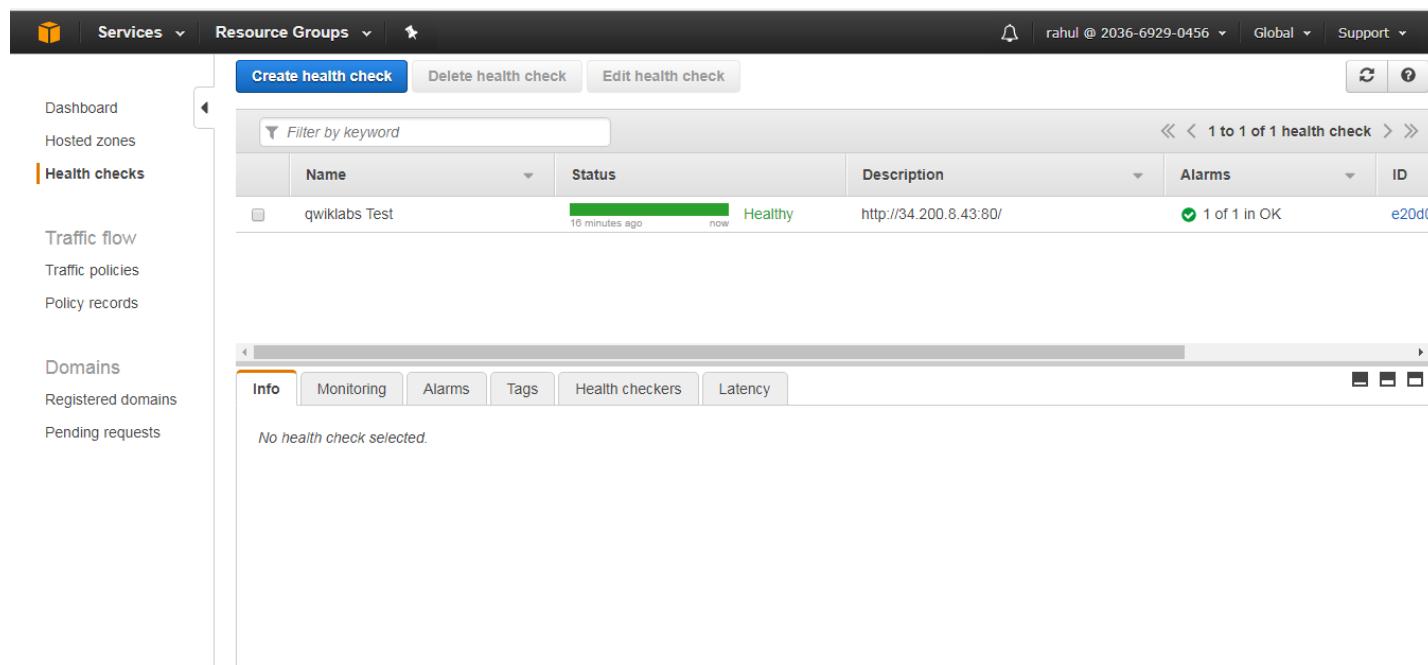
3 Buckets 1 Regions

| Bucket name | Region | Date created |
|---|-----------------------|-------------------------|
| ql-cf-templates-1501491476-18a7114dcb7024de-us-east-1 | US East (N. Virginia) | Jul 31, 2017 2:27:57 PM |
| qltrail-lab-235-1501491515 | US East (N. Virginia) | Jul 31, 2017 2:28:36 PM |
| www.route53-labs-12.com | US East (N. Virginia) | Jul 31, 2017 2:32:48 PM |

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**



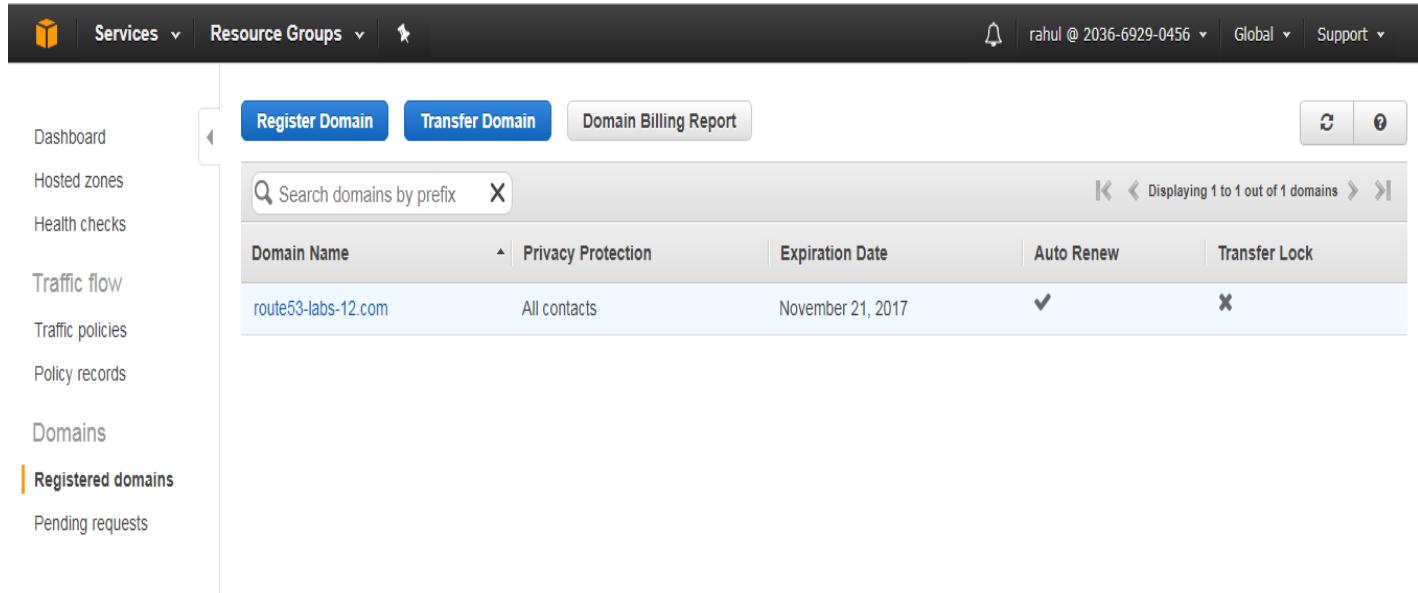
The screenshot shows the AWS Route 53 Health Checks interface. On the left, there's a sidebar with links like Dashboard, Hosted zones, Traffic flow, Domains, and Policy records. The main area has tabs for Create health check, Delete health check, and Edit health check. A search bar says 'Filter by keyword'. Below it is a table with columns: Name, Status, Description, Alarms, and ID. One row is shown: 'qwiklabs Test' (Status: Healthy, Description: http://34.200.8.43:80/, Alarms: 1 of 1 in OK, ID: e20dc). At the bottom, there are tabs for Info, Monitoring, Alarms, Tags, Health checkers, and Latency, with 'Info' selected.

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



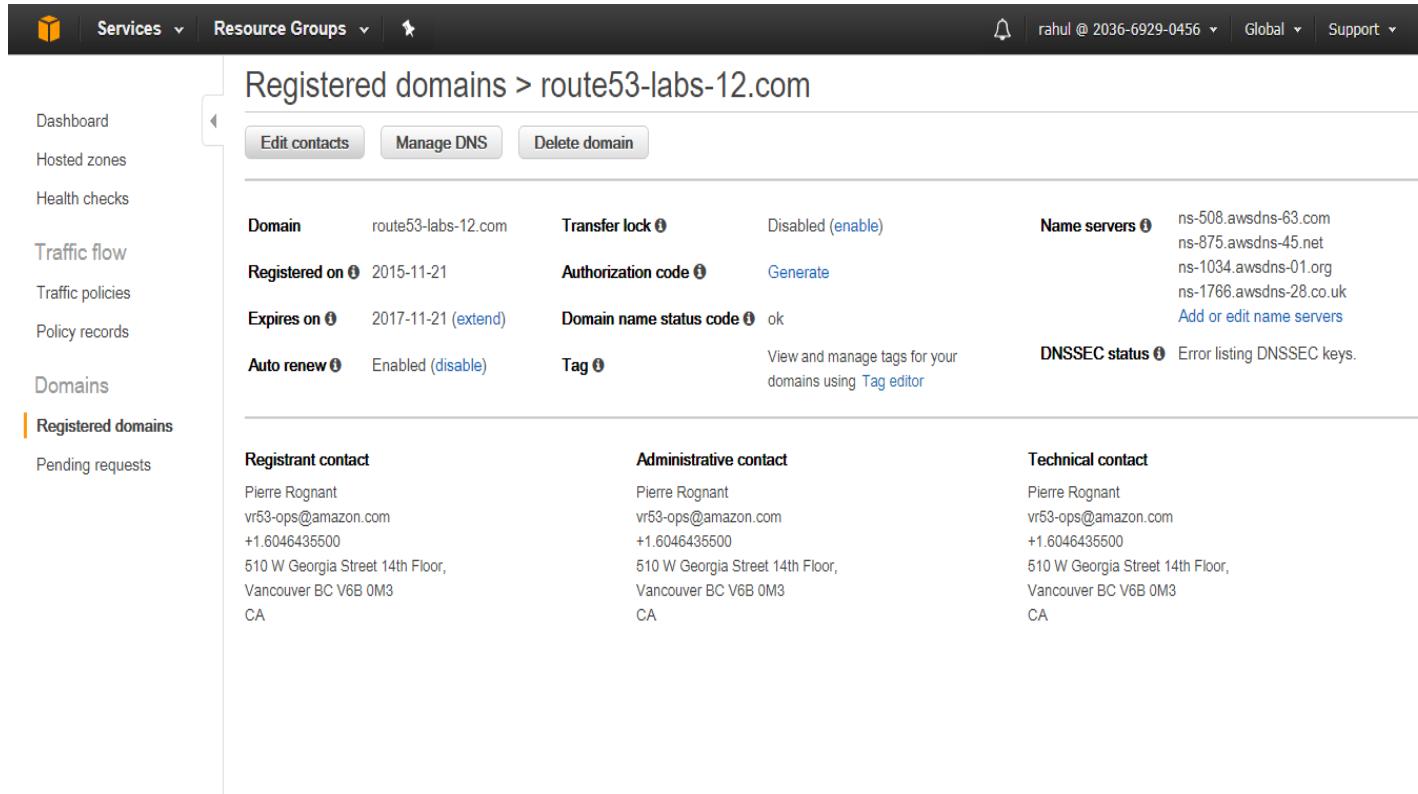
You will first need to know the domain name you are working with.

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



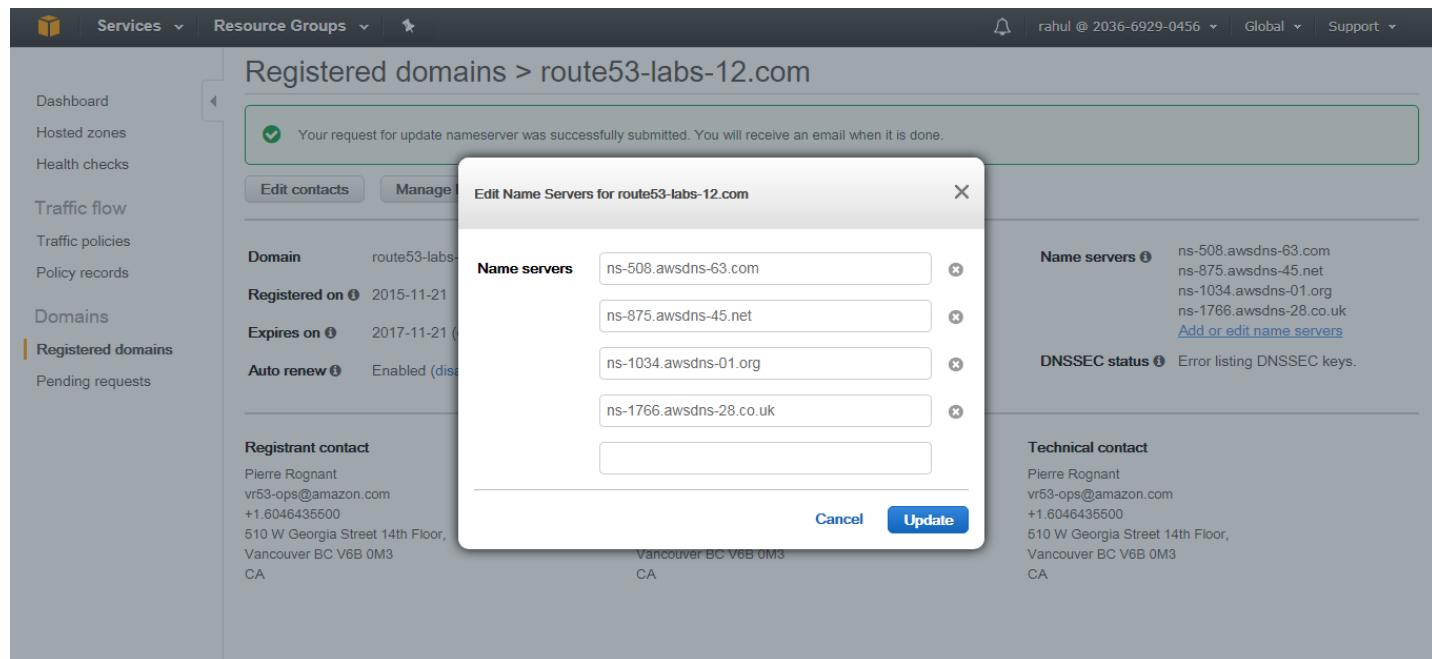
The screenshot shows the AWS Route 53 console. The left sidebar is titled "Domains" and has a sub-section "Registered domains" which is currently selected. The main content area shows a table of registered domains. The first row in the table is for the domain "route53-labs-12.com". The table includes columns for "Domain Name", "Privacy Protection", "Expiration Date", "Auto Renew", and "Transfer Lock". The "Auto Renew" column contains a checked checkbox, and the "Transfer Lock" column contains a crossed-out checkbox.

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



The screenshot shows the "Manage DNS" tab for the domain "route53-labs-12.com". The main content area displays various domain settings. Under "Name servers", it lists "ns-508.awsdns-63.com", "ns-875.awsdns-45.net", "ns-1034.awsdns-01.org", and "ns-1766.awsdns-28.co.uk". There is also a link "Add or edit name servers". Under "DNSSEC status", it says "Error listing DNSSEC keys". Below these, there are sections for "Registrant contact", "Administrative contact", and "Technical contact", each listing the same contact information: Pierre Rognant, vr53-ops@amazon.com, +1.6046435500, 510 W Georgia Street 14th Floor, Vancouver BC V6B 0M3, CA.

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



Your request for update nameserver was successfully submitted. You will receive an email when it is done.

Edit Name Servers for route53-labs-12.com

| Name servers | ns-508.awsdns-63.com |
|-------------------------|----------------------|
| ns-875.awsdns-45.net | |
| ns-1034.awsdns-01.org | |
| ns-1766.awsdns-28.co.uk | |

Registrant contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

Vancouver BC V6B 0M3
CA

Name servers

- ns-508.awsdns-63.com
- ns-875.awsdns-45.net
- ns-1034.awsdns-01.org
- ns-1766.awsdns-28.co.uk

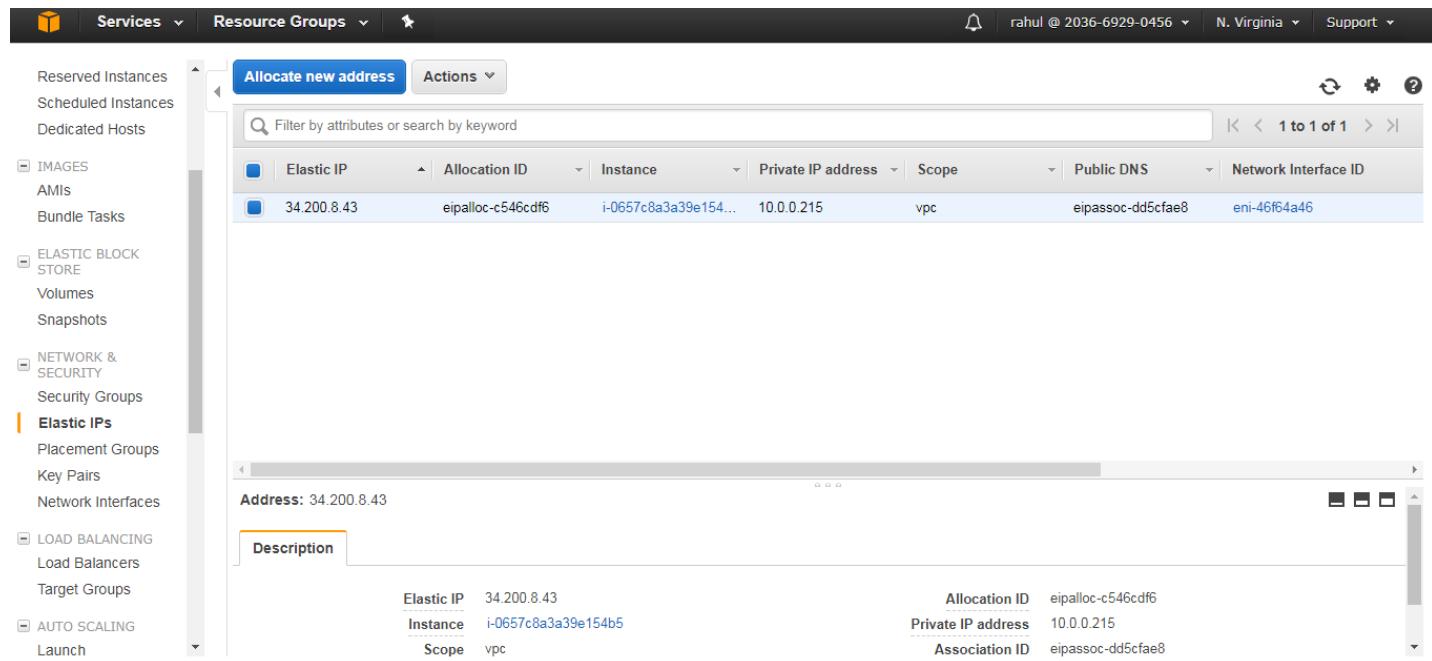
DNSSEC status

Error listing DNSSEC keys.

Technical contact

Pierre Rognant
vr53-ops@amazon.com
+1.6046435500
510 W Georgia Street 14th Floor,
Vancouver BC V6B 0M3
CA

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



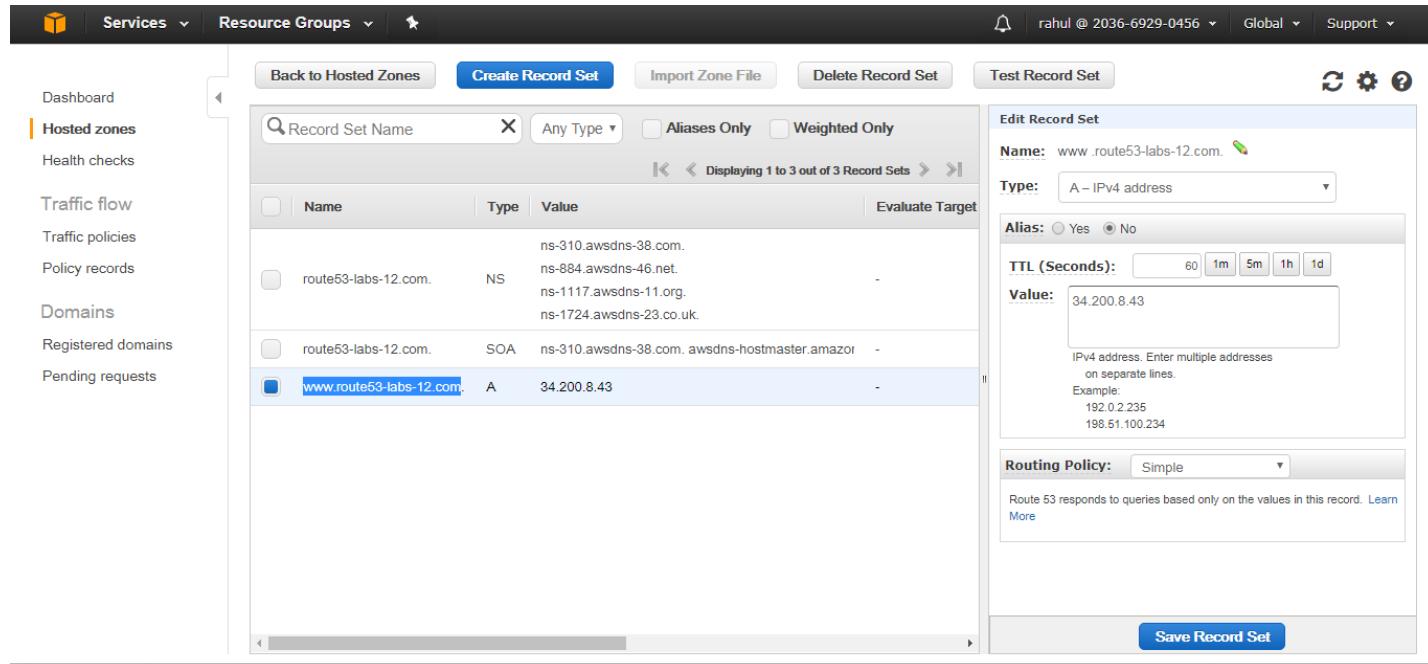
| Elastic IP | Allocation ID | Instance | Private IP address | Scope | Public DNS | Network Interface ID |
|-------------|-------------------|------------------------|--------------------|-------|-------------------|----------------------|
| 34.200.8.43 | eipalloc-c546cdf6 | i-0657c8a3a39e154b5... | 10.0.0.215 | vpc | eipassoc-dd5cfae8 | eni-46f64a46 |

Description

Elastic IP: 34.200.8.43

Elastic IP: 34.200.8.43
Allocation ID: eipalloc-c546cdf6
Instance: i-0657c8a3a39e154b5
Scope: vpc
Private IP address: 10.0.0.215
Association ID: eipassoc-dd5cfae8

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



| Name | Type | Value |
|--------------------------------|----------|--|
| route53-labs-12.com. | NS | ns-310.awsdns-38.com. ns-884.awsdns-46.net. ns-1117.awsdns-11.org. ns-1724.awsdns-23.co.uk. |
| route53-labs-12.com. | SOA | ns-310.awsdns-38.com. awsdns-hostmaster.amazon. |
| www.route53-labs-12.com | A | 34.200.8.43 |

Name: www.route53-labs-12.com

Type: A - IPv4 address

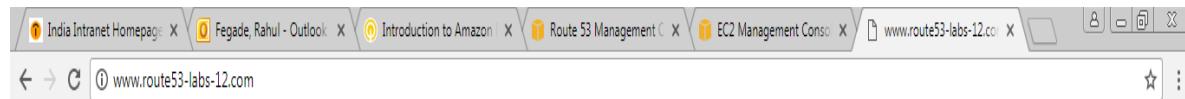
Alias: Yes No

TTL (Seconds):

Value:
 IPv4 address. Enter multiple addresses on separate lines.
 Example:
 192.0.2.235
 198.51.100.234

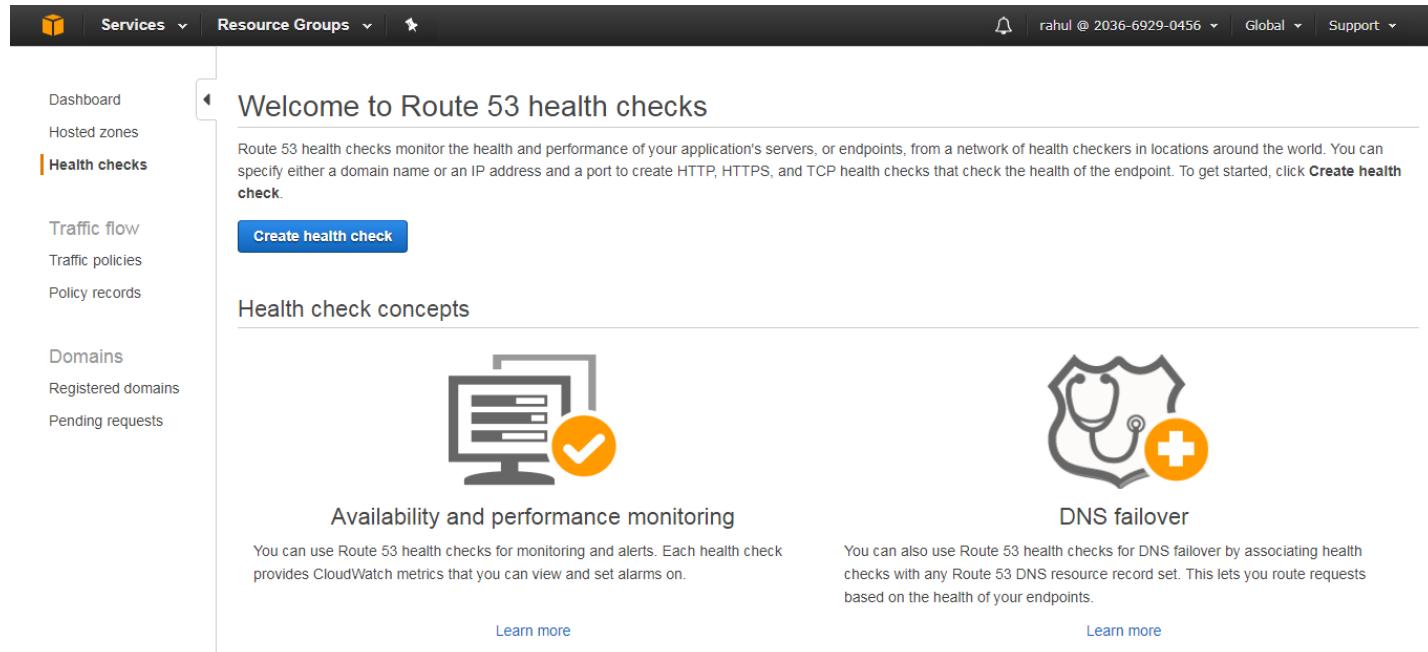
Routing Policy: Simple

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



Hello! This is your EC2 web server. It's nice to see you

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



Welcome to Route 53 health checks

Route 53 health checks monitor the health and performance of your application's servers, or endpoints, from a network of health checkers in locations around the world. You can specify either a domain name or an IP address and a port to create HTTP, HTTPS, and TCP health checks that check the health of the endpoint. To get started, click **Create health check**.

Create health check

Health check concepts



Availability and performance monitoring

You can use Route 53 health checks for monitoring and alerts. Each health check provides CloudWatch metrics that you can view and set alarms on.

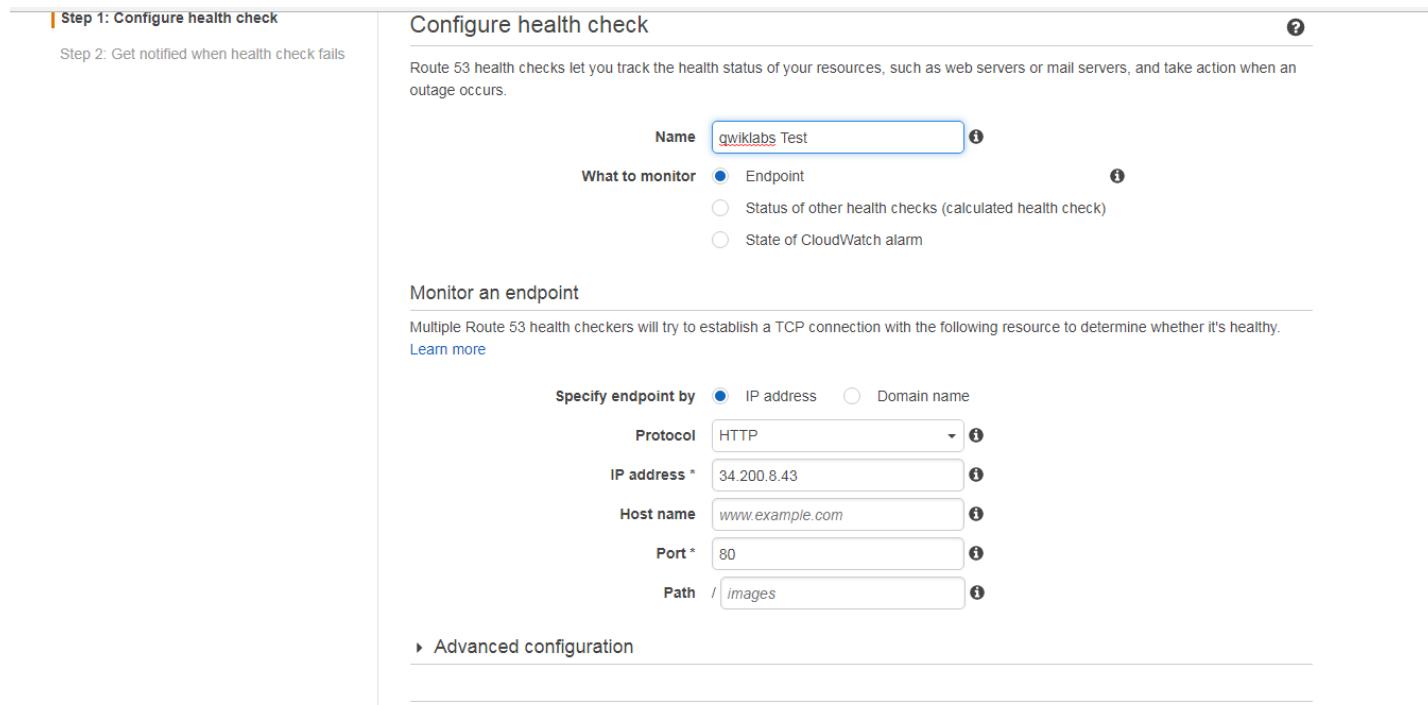
[Learn more](#)



DNS failover

You can also use Route 53 health checks for DNS failover by associating health checks with any Route 53 DNS resource record set. This lets you route requests based on the health of your endpoints.

[Learn more](#)



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 [?](#)

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 | <input type="text" value="HTTP"/> ? |
| IP address * | <input type="text" value="34.200.8.43"/> ? |
| Host name | <input type="text" value="www.example.com"/> ? |
| Port * | <input type="text" value="80"/> ? |
| Path | <input type="text" value="/images"/> ? |

[Advanced configuration](#)

Create health check

Step 1: Configure health check

Step 2: Get notified when health check fails

Get notified when health check fails

If you want CloudWatch to send you an Amazon SNS notification, such as an email, when the status of the health check changes to unhealthy, create an alarm and specify where to send notifications.

Yes No

CloudWatch sends you an Amazon SNS notification whenever the status of this health check is unhealthy for one minute.

Send notification to Existing SNS topic New SNS topic

Topic name * ServerHCFailed

Recipient email addresses * rahul.fegade@capgemini.com

Separate multiple addresses with a comma, a semicolon, or a space

* Required

Cancel Previous Create health check

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

Amazon S3

Identify optimal storage classes with S3 Analytics - Storage Class Analysis. [Learn More](#)

Documentation

Search for buckets

[+ Create bucket](#) [Delete bucket](#) [Empty bucket](#)

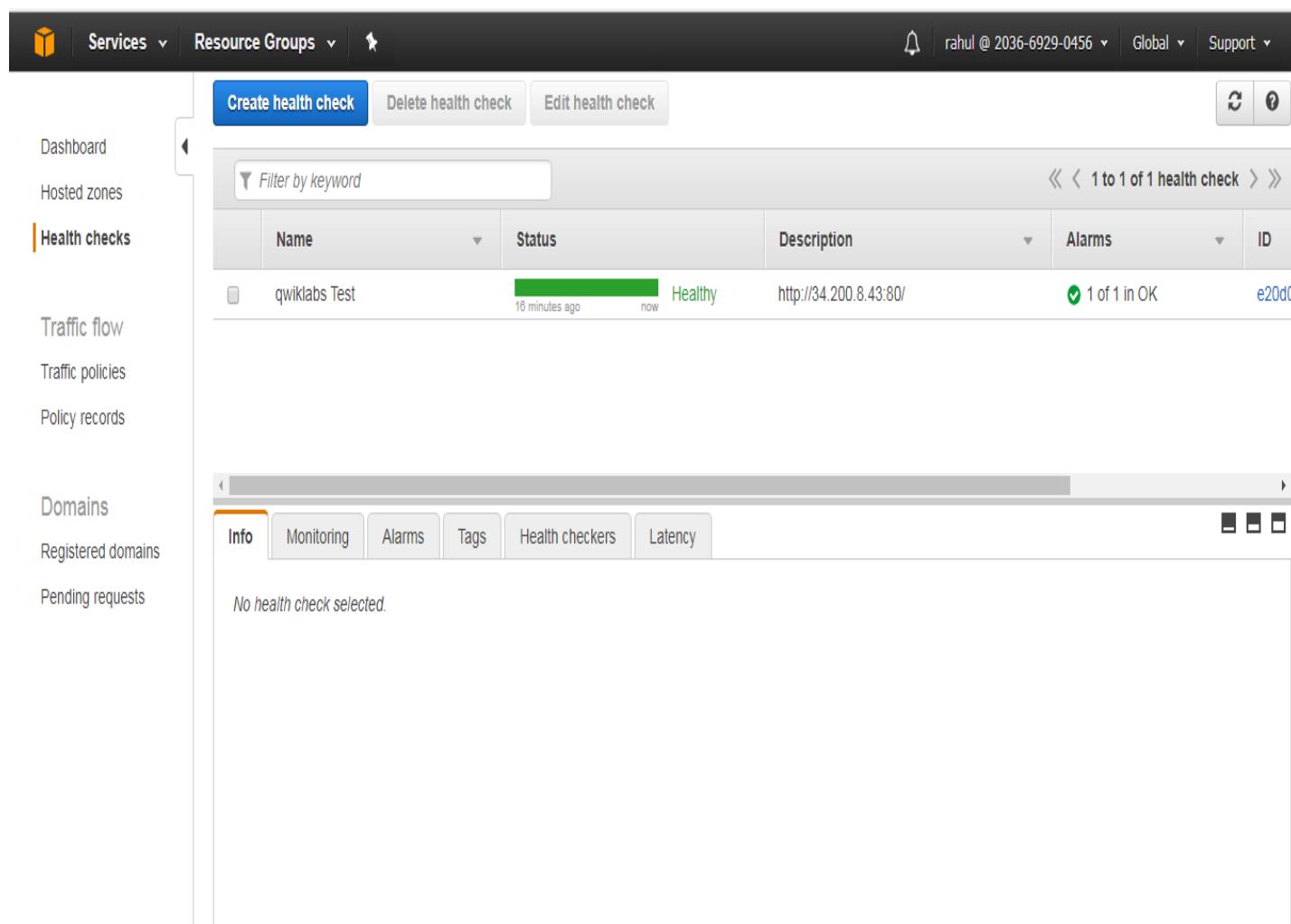
3 Buckets 1 Regions

| Bucket name | Region | Date created |
|---|-----------------------|-------------------------|
| ql-cf-templates-1501491476-18a7114dcb7024de-us-east-1 | US East (N. Virginia) | Jul 31, 2017 2:27:57 PM |
| qltrail-lab-235-1501491515 | US East (N. Virginia) | Jul 31, 2017 2:28:36 PM |
| www.route53-labs-12.com | US East (N. Virginia) | Jul 31, 2017 2:32:48 PM |

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**



The screenshot shows the AWS Route 53 Health Checks page. The left sidebar has a 'Health checks' section selected. The main content area displays a table of health checks. One row is visible:

| | Name | Status | Description | Alarms | ID |
|---|---------------|---|------------------------|--|-------|
|  | qwiklabs Test | Healthy 16 minutes ago now | http://34.200.8.43:80/ |  1 of 1 in OK | e20dc |

Below the table, there is a detailed view for the selected health check, showing tabs for Info, Monitoring, Alarms, Tags, Health checkers, and Latency. The Info tab is active, displaying the message "No health check selected."

- 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:

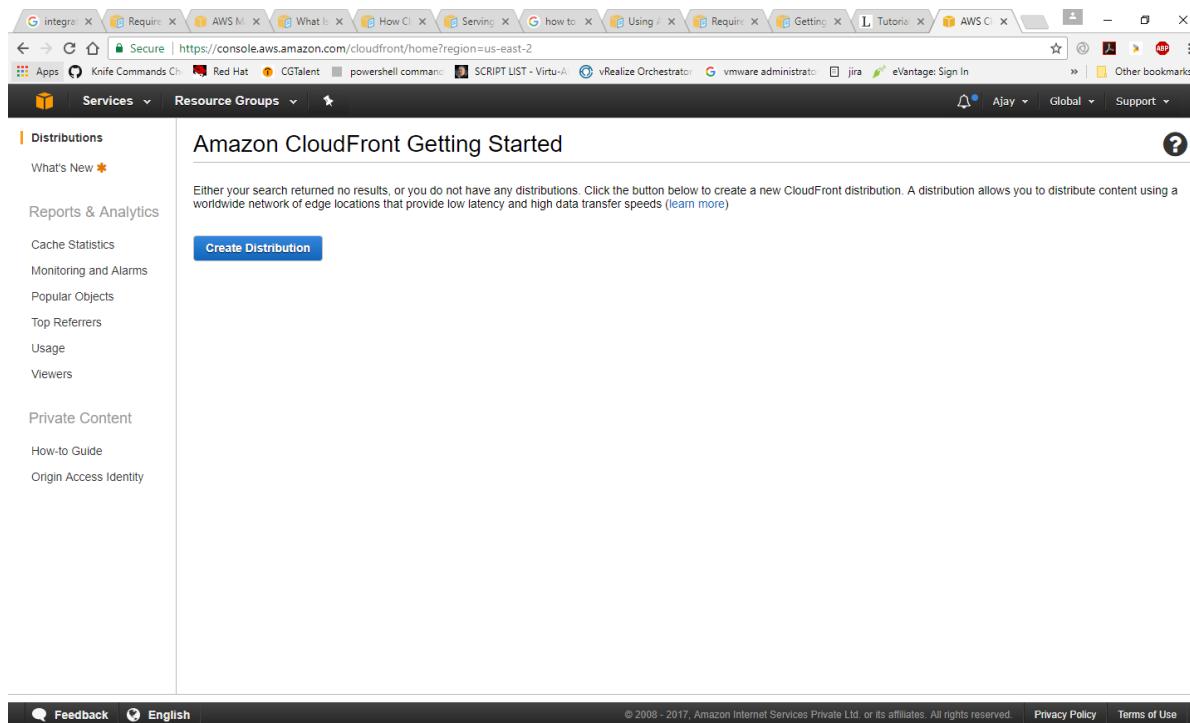
Step 1: Create a S3 Bucket with objects and grant public access

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

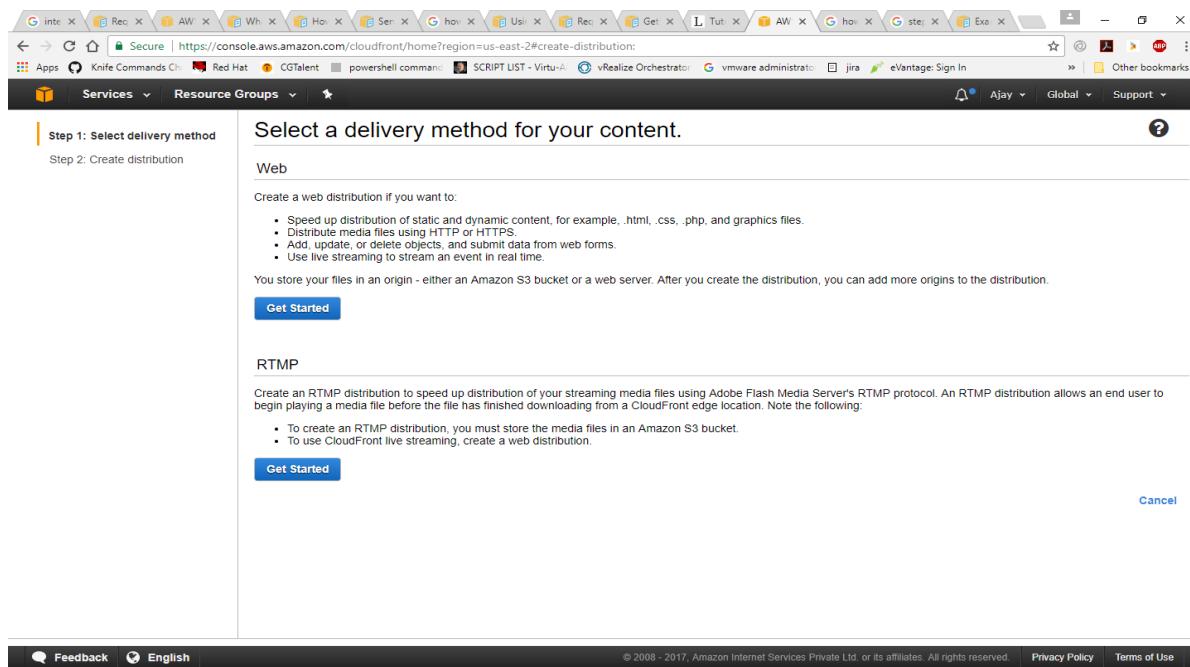


Amazon CloudFront Getting Started

Either your search returned no results, or you do not have any distributions. Click the button below to create a new CloudFront distribution. A distribution allows you to distribute content using a worldwide network of edge locations that provide low latency and high data transfer speeds ([learn more](#))

Create Distribution

Select delivery method for content



Step 1: Select delivery method

[Step 2: Create distribution](#)

Select a delivery method for your content.

Web

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin - either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

Get Started

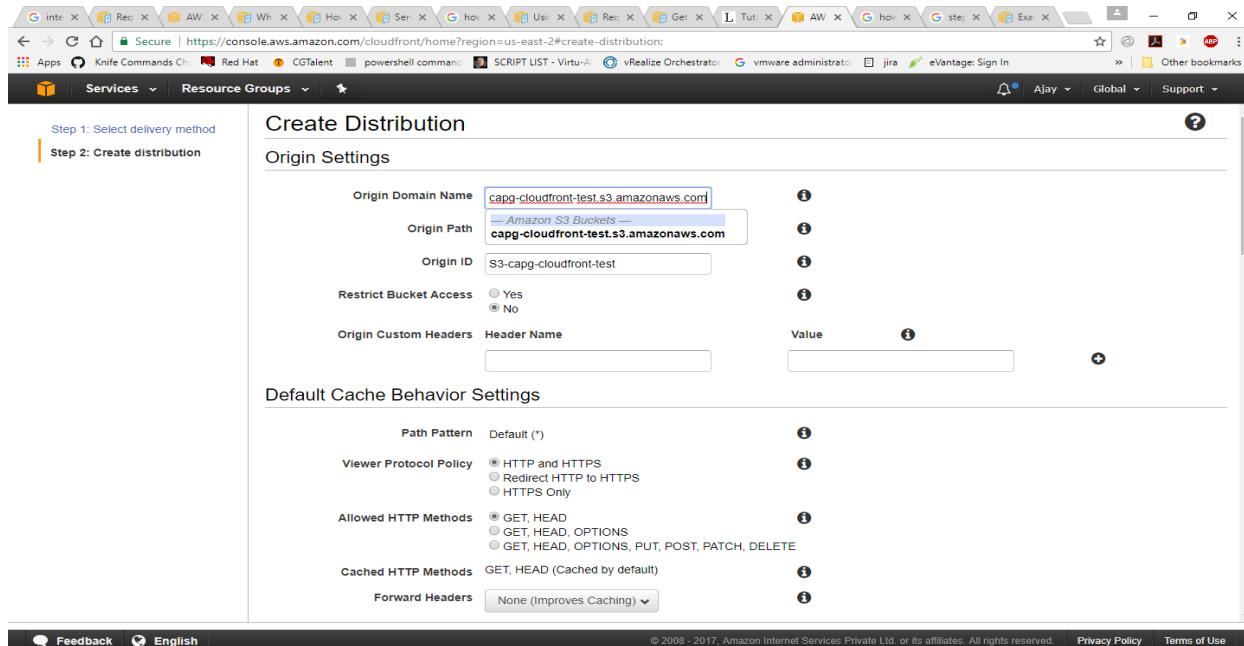
RTMP

Create an RTMP distribution to speed up distribution of your streaming media files using Adobe Flash Media Server's RTMP protocol. An RTMP distribution allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location. Note the following:

- To create an RTMP distribution, you must store the media files in an Amazon S3 bucket.
- To use CloudFront live streaming, create a web distribution.

Get Started **Cancel**

Give the origin domain name



Create Distribution

Step 1: Select delivery method

Step 2: Create distribution

Origin Settings

Origin Domain Name: capg-cloudfront-test.s3.amazonaws.com

Origin Path: capg-cloudfront-test.s3.amazonaws.com

Origin ID: S3-capg-cloudfront-test

Restrict Bucket Access: Yes No

Origin Custom Headers:

| Header Name | Value |
|-------------|-------|
| | |

Default Cache Behavior Settings

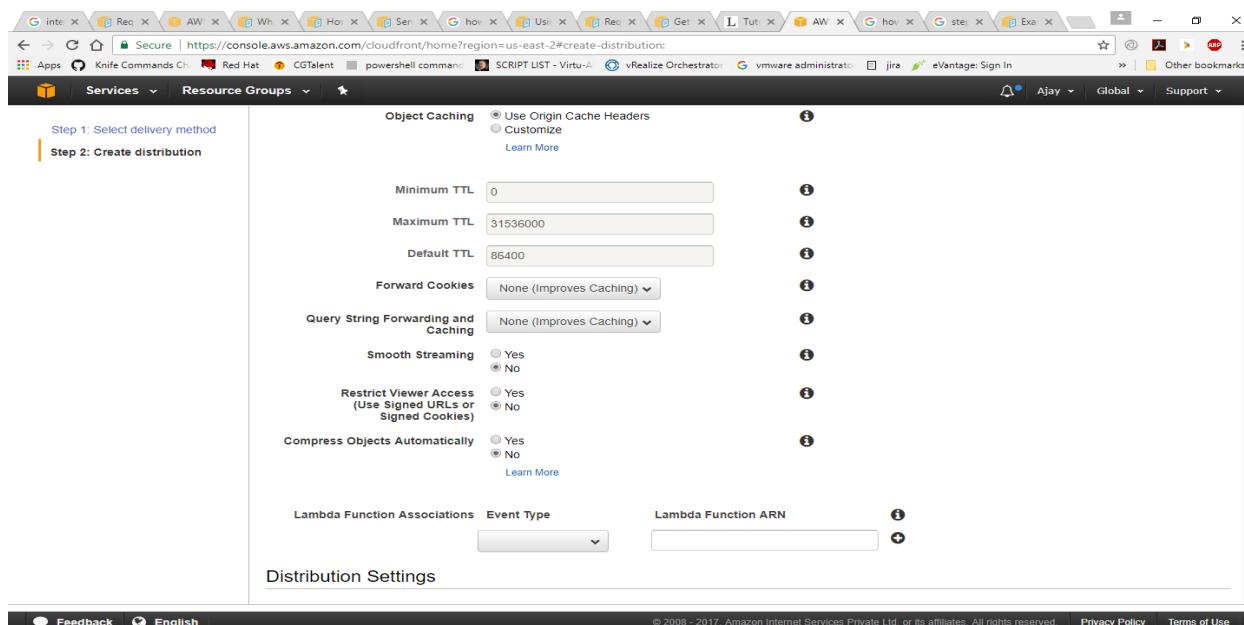
Path Pattern: Default (*)

Viewer Protocol Policy: HTTP and HTTPS Redirect HTTP to HTTPS HTTPS Only

Allowed HTTP Methods: GET, HEAD GET, HEAD, OPTIONS GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cached HTTP Methods: GET, HEAD (Cached by default)

Forward Headers: None (Improves Caching)



Create Distribution

Step 1: Select delivery method

Step 2: Create distribution

Object Caching

Use Origin Cache Headers Customize [Learn More](#)

Minimum TTL: 0

Maximum TTL: 31536000

Default TTL: 86400

Forward Cookies: None (Improves Caching)

Query String Forwarding and Caching: None (Improves Caching)

Smooth Streaming: Yes No

Restrict Viewer Access (Use Signed URLs or Signed Cookies): Yes No

Compress Objects Automatically: Yes No [Learn More](#)

Lambda Function Associations: Event Type: [Select](#) Lambda Function ARN: [Select](#)

Distribution Settings

Secure | https://console.aws.amazon.com/cloudfront/home?region=us-east-2#create-distribution:

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://d11111abcdef8.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:

Step 1: Select delivery method

Step 2: Create distribution

Distribution Settings

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

Screenshot of the AWS CloudFront Distributions page:

The page shows a single distribution entry:

| Delivery Method | ID | Domain Name | Comment | Origin | CNAMEs | Status | State | Last Modified |
|-----------------|---------------|-------------------------|---------|-------------------|--------|----------|---------|---------------------|
| Web | EPBCSCNW1AGX8 | dkzI90gd19jy3.cloudf... | - | capg-cloudfront-t | - | Deployed | Enabled | 2017-07-31 14:56 UT |

Test Your Links

Screenshot of a web browser displaying a test page:

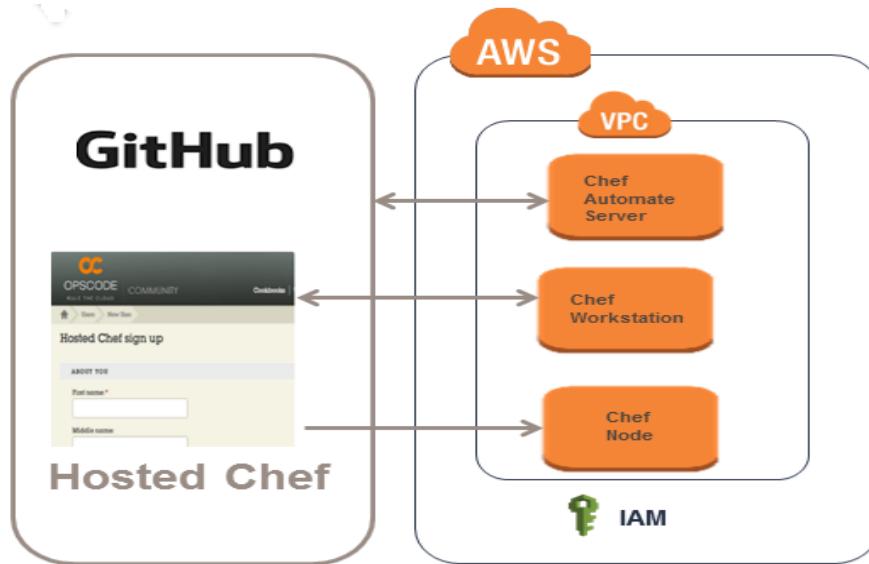
The page content includes:

- My CloudFront Test
- My text content goes here.
- 

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 |

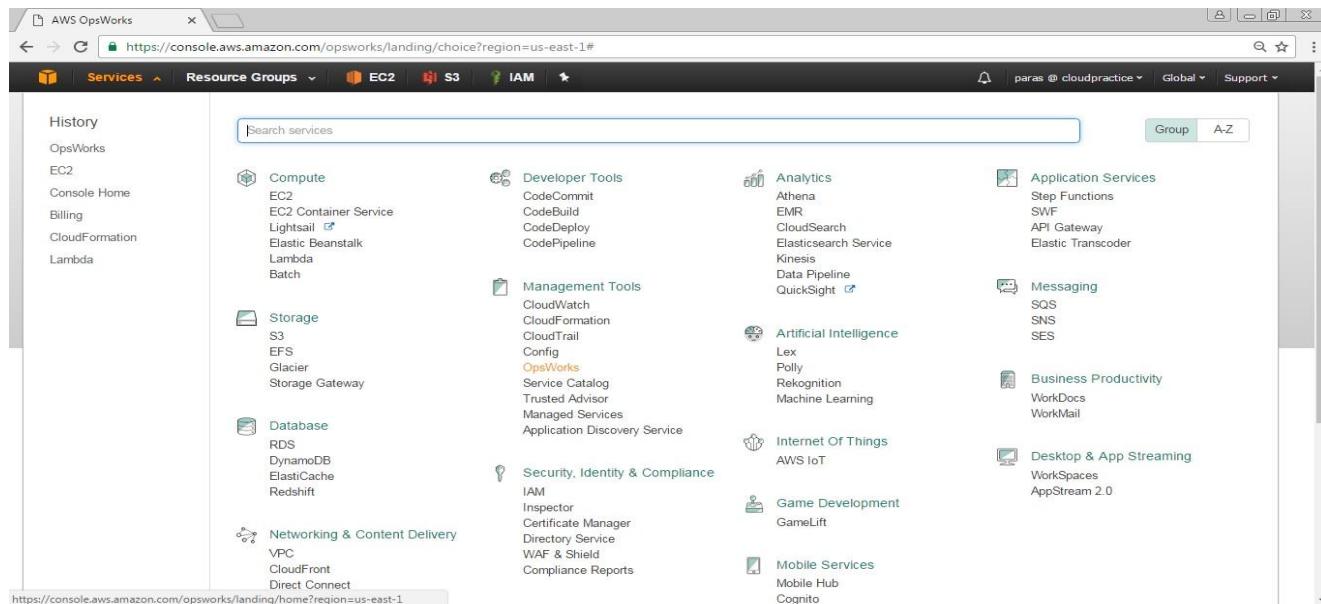
| | | |
|--------|----------------|----------|
| Node 1 | Micro instance | RHEL 7.2 |
|--------|----------------|----------|

- Git / GitHub

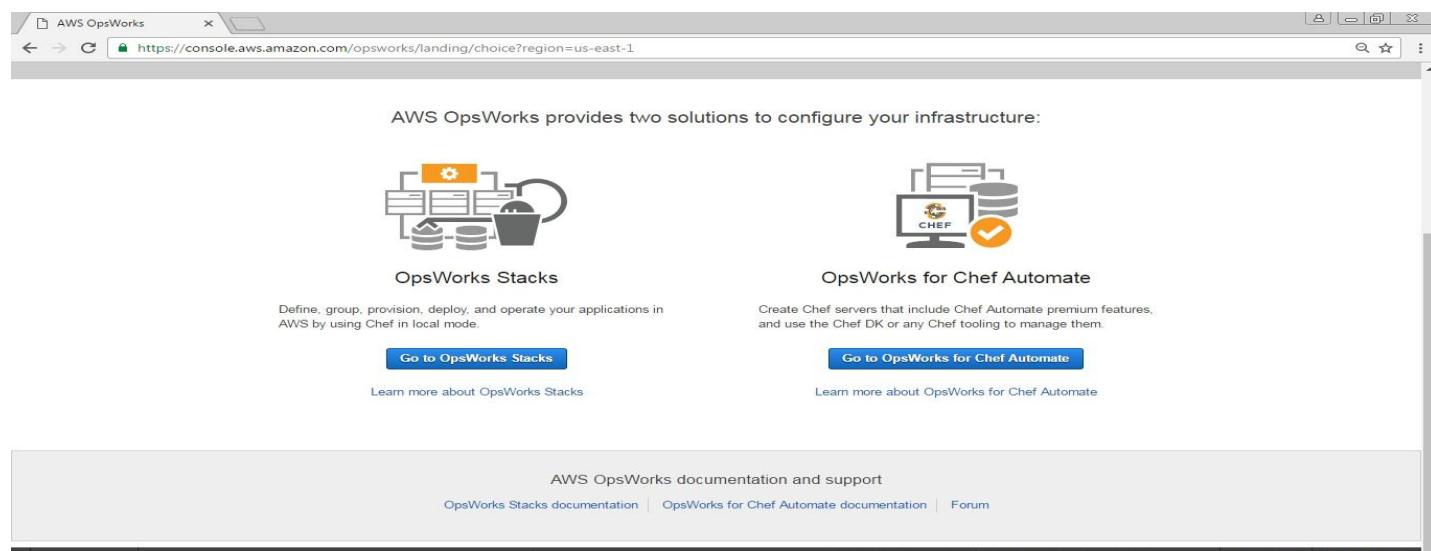
15.1.3 Procedure

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.

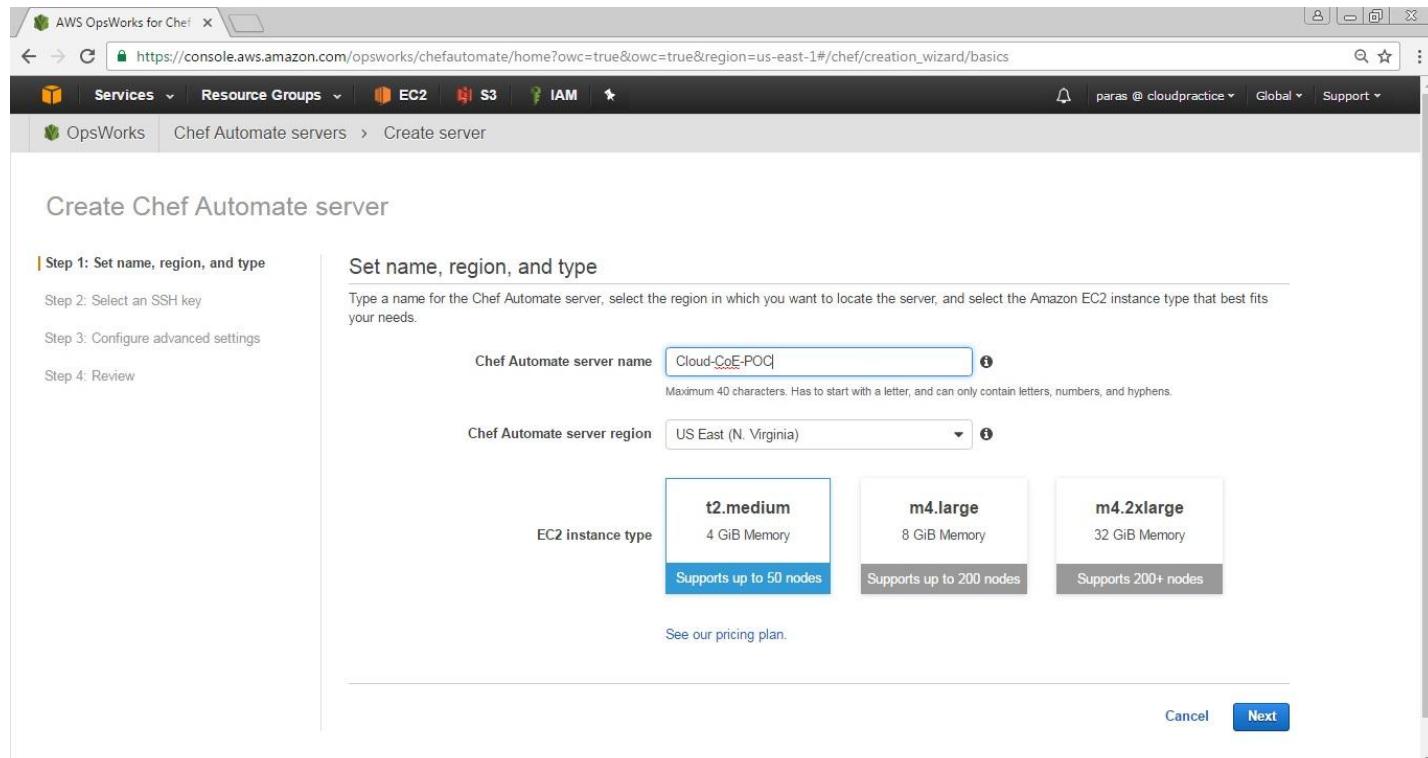


The screenshot shows the AWS OpsWorks landing page. It highlights two main options: "OpsWorks Stacks" and "OpsWorks for Chef Automate".

- OpsWorks Stacks:** Described as providing two solutions to configure your infrastructure. It features a diagram of a stack with databases and a monitor. A blue button says "Go to OpsWorks Stacks". Below it, a link says "Learn more about OpsWorks Stacks".
- OpsWorks for Chef Automate:** Described as creating Chef servers that include Chef Automate premium features. It features a diagram of a computer monitor with the word "CHEF" and a checkmark. A blue button says "Go to OpsWorks for Chef Automate". Below it, a link says "Learn more about OpsWorks for Chef Automate".

At the bottom, there's a footer with links to "AWS OpsWorks documentation and support", "OpsWorks Stacks documentation", "OpsWorks for Chef Automate documentation", and "Forum". The URL in the address bar is https://console.aws.amazon.com/opsworks/chefautomate/home?owc=true®ion=us-east-1#.

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



Create Chef Automate server

Step 1: Set name, region, and type

Type a name for the Chef Automate server, select the region in which you want to locate the server, and select the Amazon EC2 instance type that best fits your needs.

Chef Automate server name: Cloud-CoE-POC

Maximum 40 characters. Has to start with a letter, and can only contain letters, numbers, and hyphens.

Chef Automate server region: US East (N. Virginia)

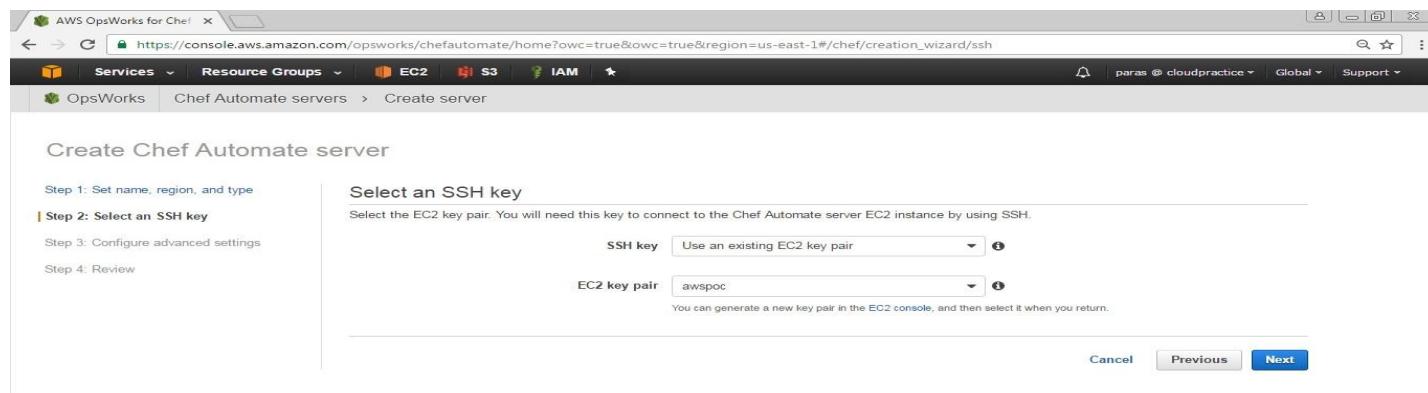
EC2 instance type:

- t2.medium**: 4 GiB Memory. Supports up to 50 nodes.
- m4.large**: 8 GiB Memory. Supports up to 200 nodes.
- m4.2xlarge**: 32 GiB Memory. Supports 200+ nodes.

See our pricing plan.

Cancel **Next**

- 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).



Create Chef Automate server

Step 1: Set name, region, and type

Step 2: Select an SSH key

Select the EC2 key pair. You will need this key to connect to the Chef Automate server EC2 instance by using SSH.

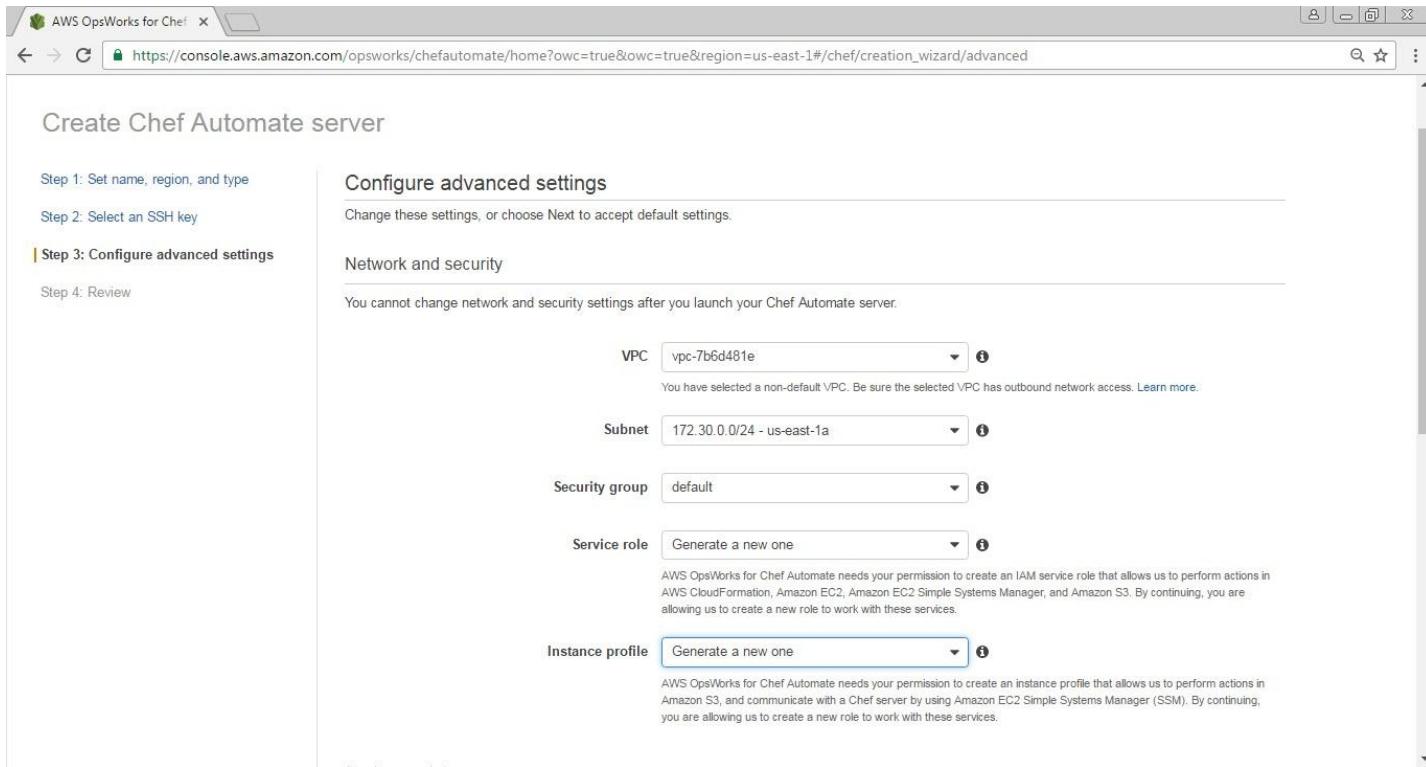
SSH key: Use an existing EC2 key pair

EC2 key pair: awspoc

You can generate a new key pair in the EC2 console, and then select it when you return.

Cancel **Previous** **Next**

- 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**.



Create Chef Automate server

Step 1: Set name, region, and type
Step 2: Select an SSH key
Step 3: Configure advanced settings
Step 4: Review

Configure advanced settings

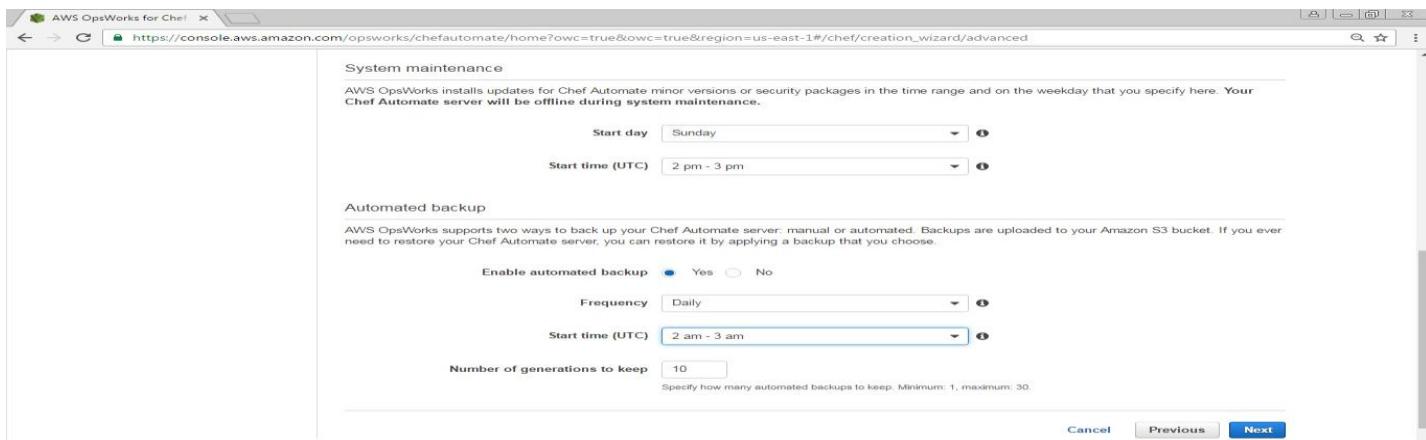
Change these settings, or choose Next to accept default settings.

Network and security

You cannot change network and security settings after you launch your Chef Automate server.

VPC: vpc-7b6d481e
Subnet: 172.30.0.0/24 - us-east-1a
Security group: default
Service role: Generate a new one
Instance profile: Generate a new one

- 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.



System maintenance

AWS OpsWorks installs updates for Chef Automate minor versions or security packages in the time range and on the weekday that you specify here. Your Chef Automate server will be offline during system maintenance.

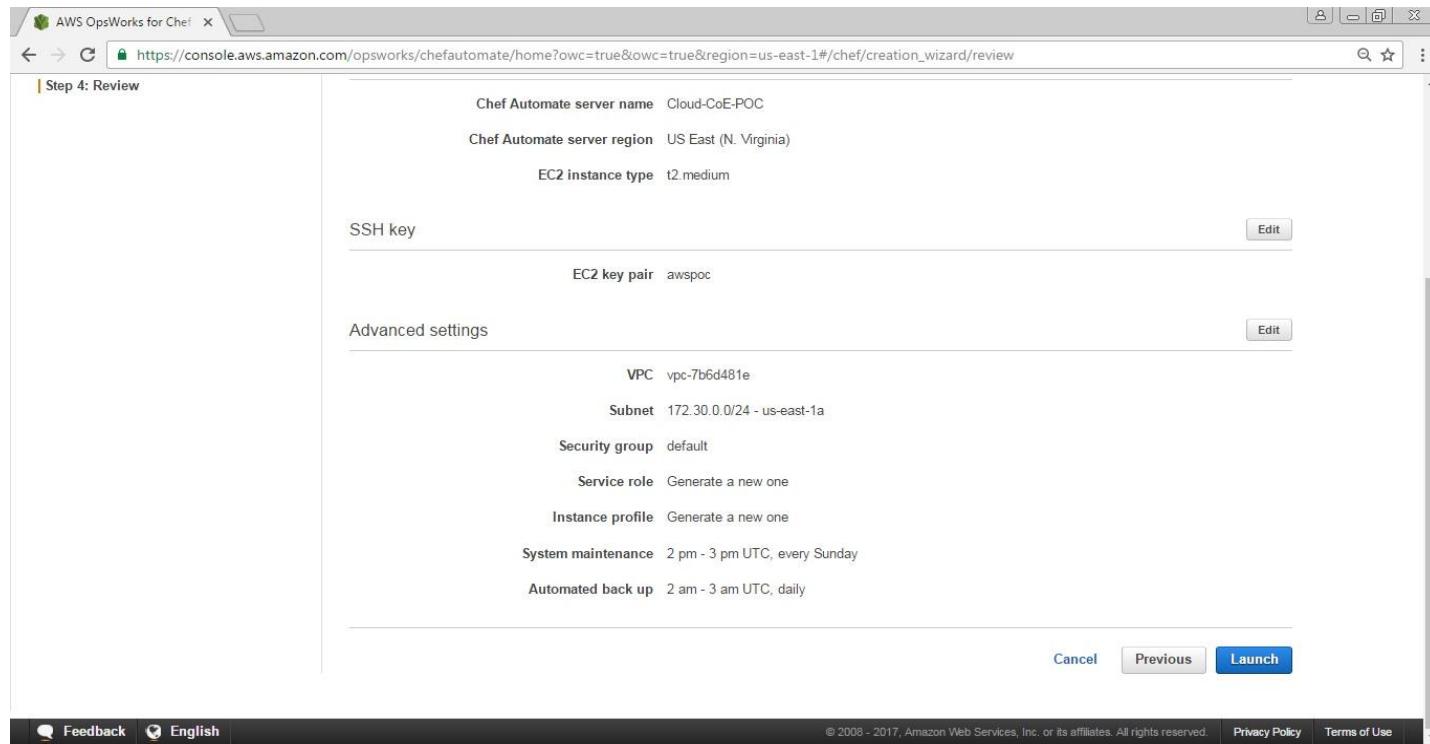
Start day: Sunday
Start time (UTC): 2 pm - 3 pm

Automated backup

AWS OpsWorks supports two ways to back up your Chef Automate server: manual or automated. Backups are uploaded to your Amazon S3 bucket. If you ever need to restore your Chef Automate server, you can restore it by applying a backup that you choose.

Enable automated backup: Yes
Frequency: Daily
Start time (UTC): 2 am - 3 am
Number of generations to keep: 10

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



Chef Automate server name: Cloud-CoE-POC

Chef Automate server region: US East (N. Virginia)

EC2 instance type: t2.medium

SSH key:

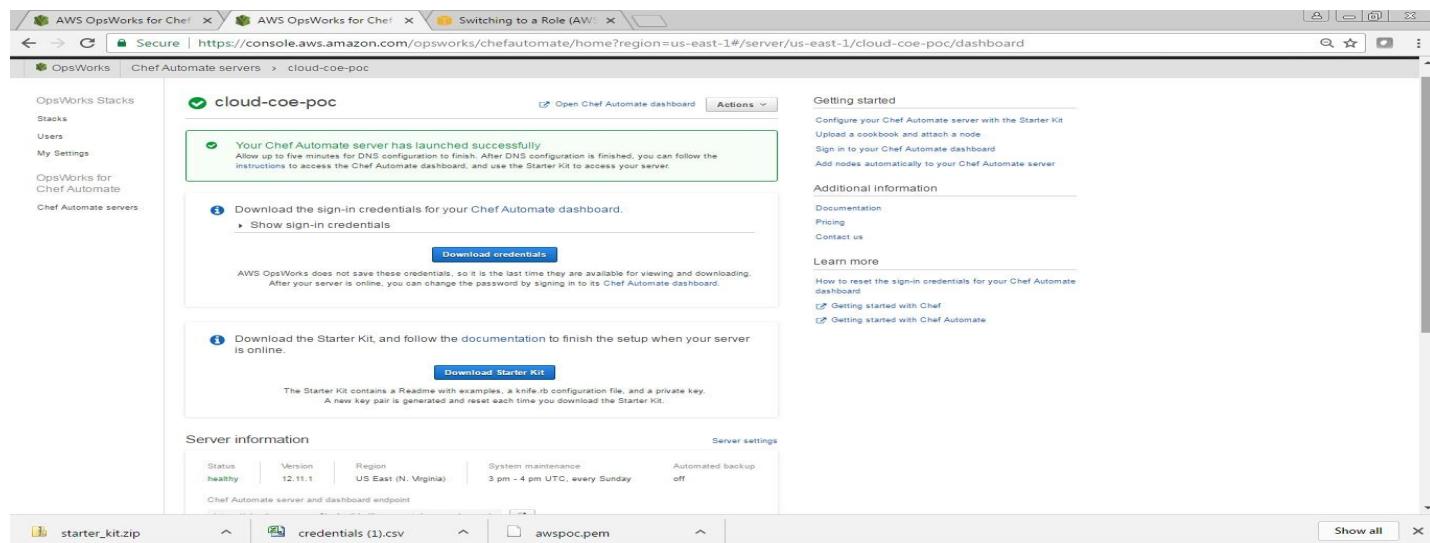
EC2 key pair: awspoc

Advanced settings:

- VPC:** vpc-7b6d481e
- Subnet:** 172.30.0.0/24 - us-east-1a
- Security group:** default
- Service role:** Generate a new one
- Instance profile:** Generate a new one
- System maintenance:** 2 pm - 3 pm UTC, every Sunday
- Automated back up:** 2 am - 3 am UTC, daily

Buttons: Cancel, Previous, Launch

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



Getting started:

- Your Chef Automate server has launched successfully.
- Download the sign-in credentials for your Chef Automate dashboard.
- Download the Starter Kit, and follow the documentation to finish the setup when your server is online.

Server information:

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

Server settings:

Chef Automate server and dashboard endpoint: [redacted]

File browser:

- starter_kit.zip
- credentials (1).csv
- awspoc.pem

Server information

[Server settings](#)

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

Chef Automate server and dashboard endpoint

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



i Before you can work with your Chef Automate server, make sure you have [Chef DK](#) installed.

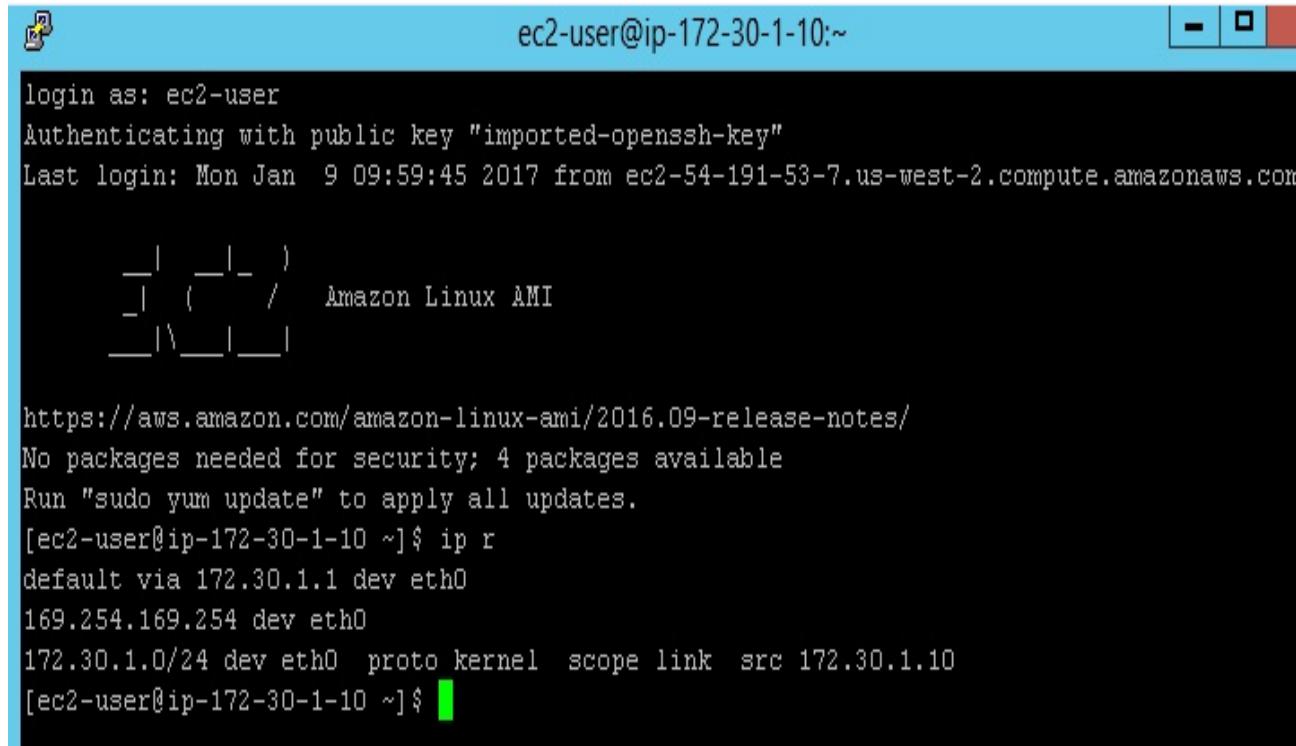
Recent events

| 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 |
| 2017-01-09T09:27:59Z | 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



```

ec2-user@ip-172-30-1-10:~ 
login as: ec2-user
Authenticating with public key "imported-ssh-key"
Last login: Mon Jan  9 09:59:45 2017 from ec2-54-191-53-7.us-west-2.compute.amazonaws.com

[ec2-user@ip-172-30-1-10 ~] $ ip r
default via 172.30.1.1 dev eth0
169.254.169.254 dev eth0
172.30.1.0/24 dev eth0 proto kernel scope link src 172.30.1.10
[ec2-user@ip-172-30-1-10 ~] $ 

```

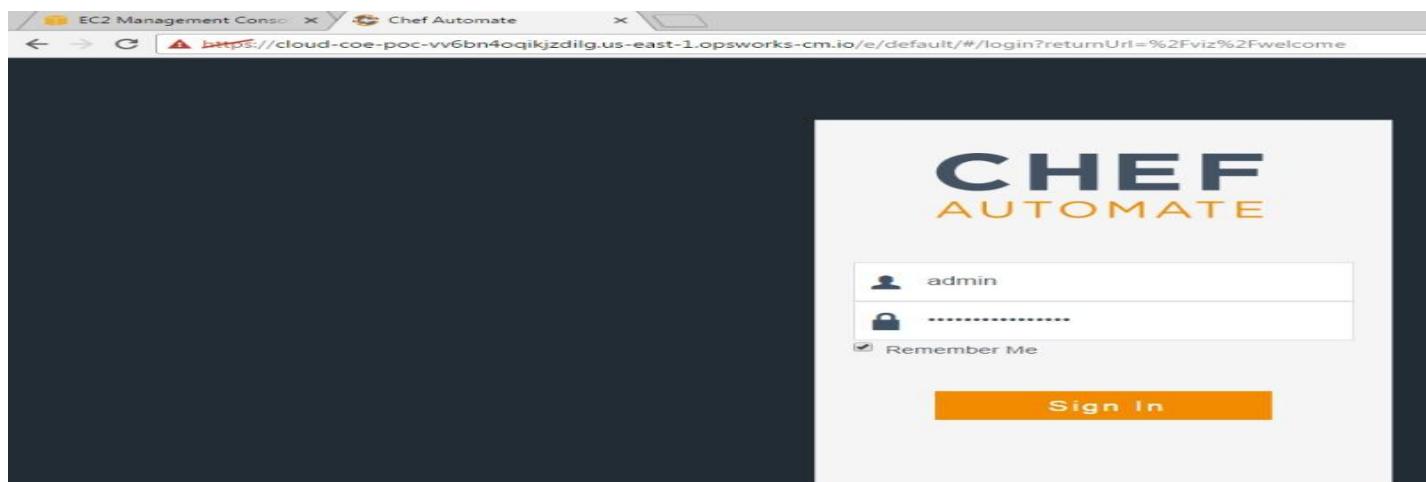
- Use the downloaded starter kit and copy it in the Chef Automate Server

```
[root@ip-172-30-0-183 ~]# unzip starter_kit.zip
Archive: starter_kit.zip
  creating: cloud-coe-poc-vv6bn4oqikjzdilg/
  creating: cloud-coe-poc-vv6bn4oqikjzdilg/.chef/
  creating: cloud-coe-poc-vv6bn4oqikjzdilg/.chef/ca_certs/
  creating: cloud-coe-poc-vv6bn4oqikjzdilg/cookbooks/
  creating: cloud-coe-poc-vv6bn4oqikjzdilg/environments/
  creating: cloud-coe-poc-vv6bn4oqikjzdilg/roles/
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/.chef/ca_certs/opsworks-cm-ca-2016-root.pem
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/Berksfile
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/README.md
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/chefignore
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/cookbooks/README.md
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/environments/README.md
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/roles/README.md
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/.chef/knife.rb
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/.chef/private.pem
  inflating: cloud-coe-poc-vv6bn4oqikjzdilg/userdata.sh
[root@ip-172-30-0-183 ~]# ll
total 104888
-rw-rw-r--. 1 ec2-user ec2-user 107390653 Dec 14 21:44 chefdk-1.1.16-1.el7.x86_64.rpm
drwxr-xr-x. 6 root    root        140 Jan  9 2017 cloud-coe-poc-vv6bn4oqikjzdilg
-rw-rw-r--. 1 ec2-user ec2-user     10916 Jan  9 04:25 starter_kit.zip
[root@ip-172-30-0-183 ~]#
```

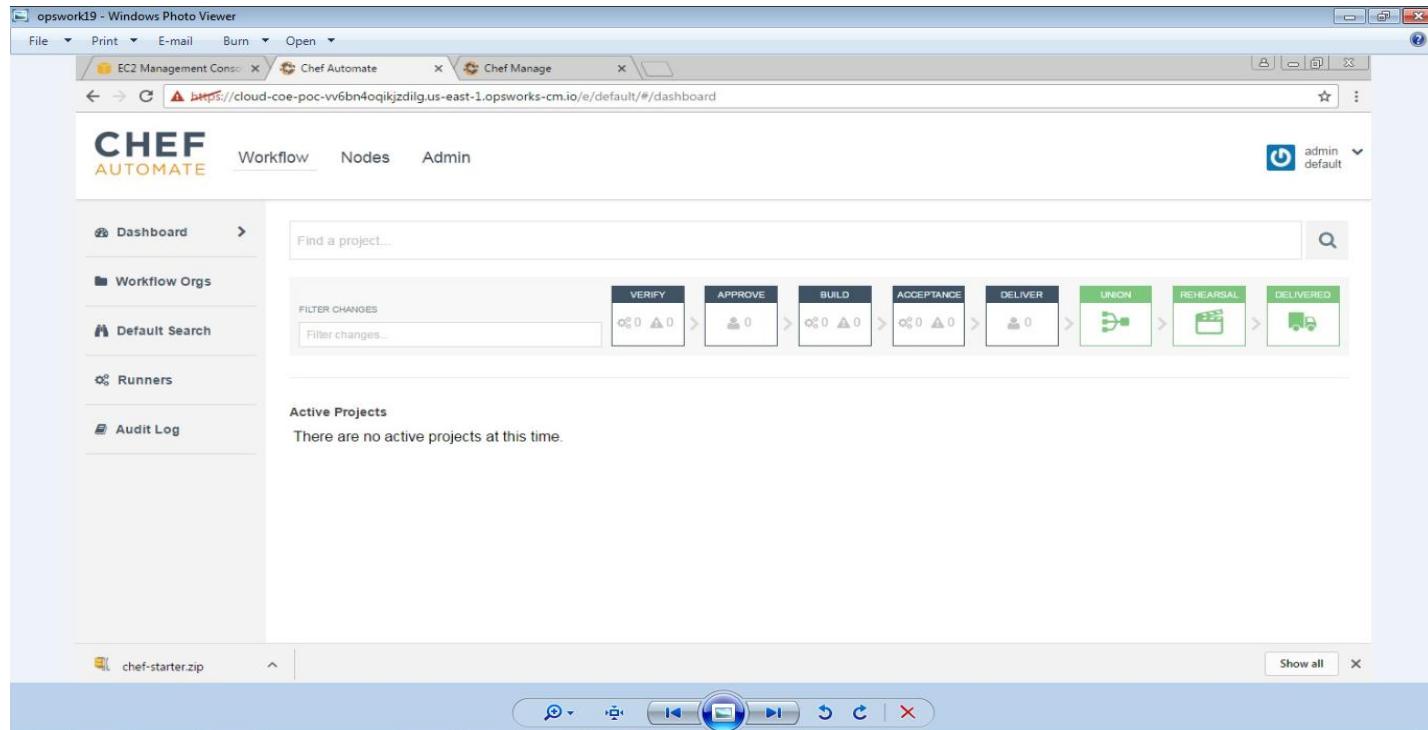
- Verify the [ChefDK](#) installation using chef commands on the Terminal

```
[root@ip-172-30-1-10 anchors]# chef --version
Chef Development Kit Version: 1.1.16
chef-client version: 12.17.44
delivery version: master (83358fb62c0f711c70ad5a81030a6cae4017f103)
berks version: 5.2.0
kitchen version: 1.14.2
[root@ip-172-30-1-10 anchors]#
```

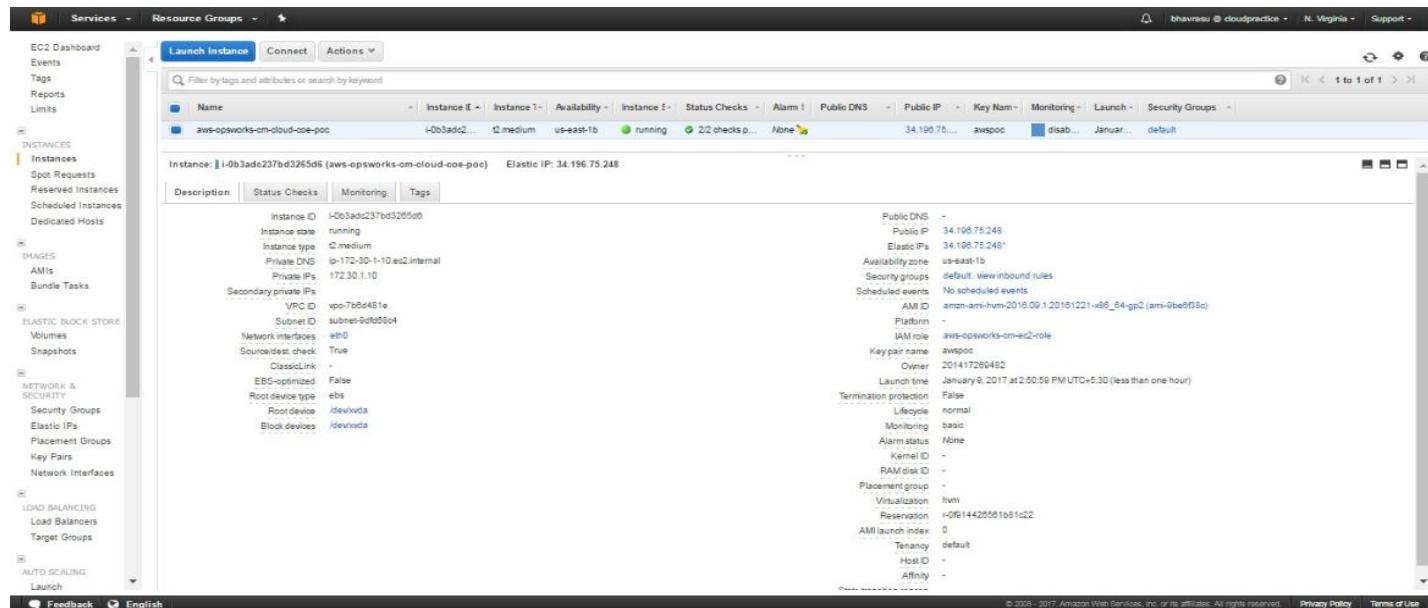
- Open the browser and copy paste the URL provided in the above (step 9) inside the server information section



- 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



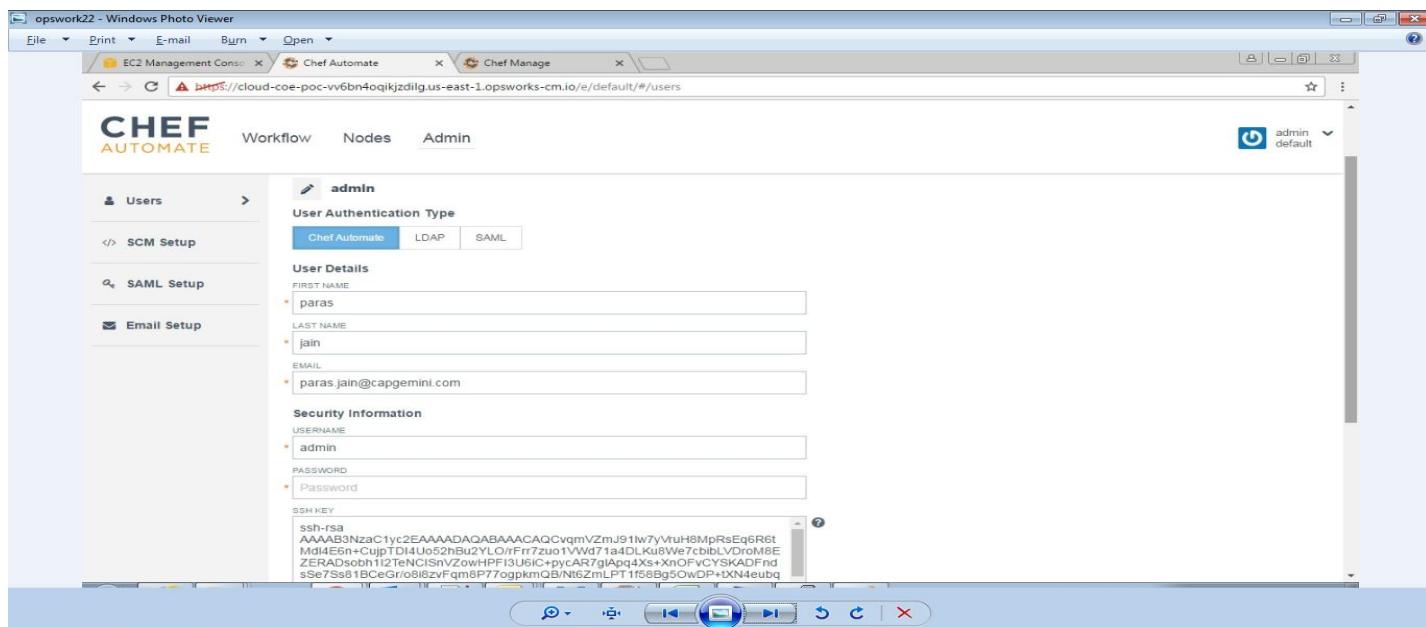
Note: You can integrate Chef Automate server using either Git / GitHub option

15.1.3.2 Integrating Git Repository with Chef Automate Server

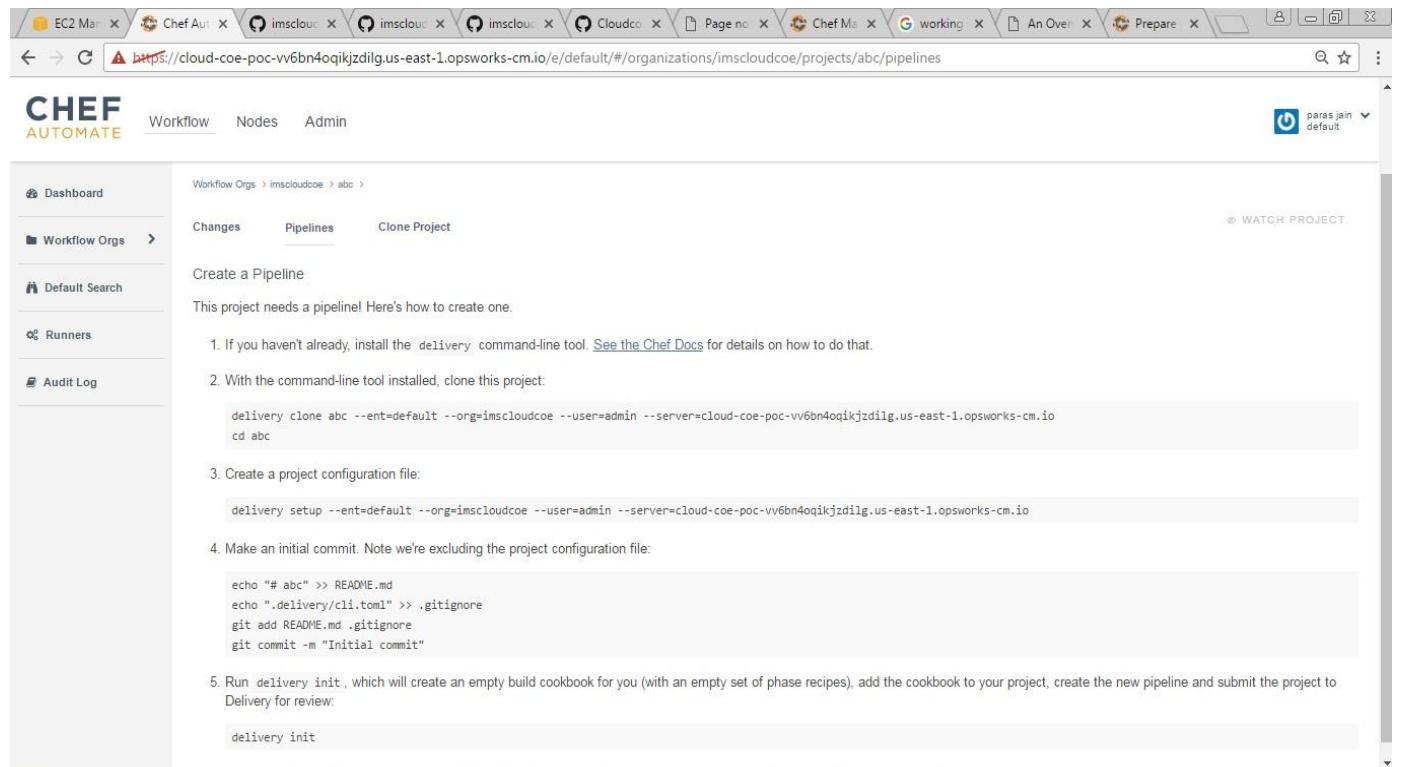
- Now install Git on the Chef automation server using following commands

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



The screenshot shows the Chef Automate interface for a workflow organization named 'imscloudco'. On the left, a sidebar lists options like Dashboard, Workflow Orgs, Default Search, Runners, and Audit Log. The main content area is titled 'Create a Pipeline' and provides instructions for setting up a pipeline. It includes several command-line snippets:

```

1. If you haven't already, install the delivery command-line tool. See the Chef Docs for details on how to do that.

2. With the command-line tool installed, clone this project:

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

3. Create a project configuration file:

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

4. Make an initial commit. Note we're excluding the project configuration file:

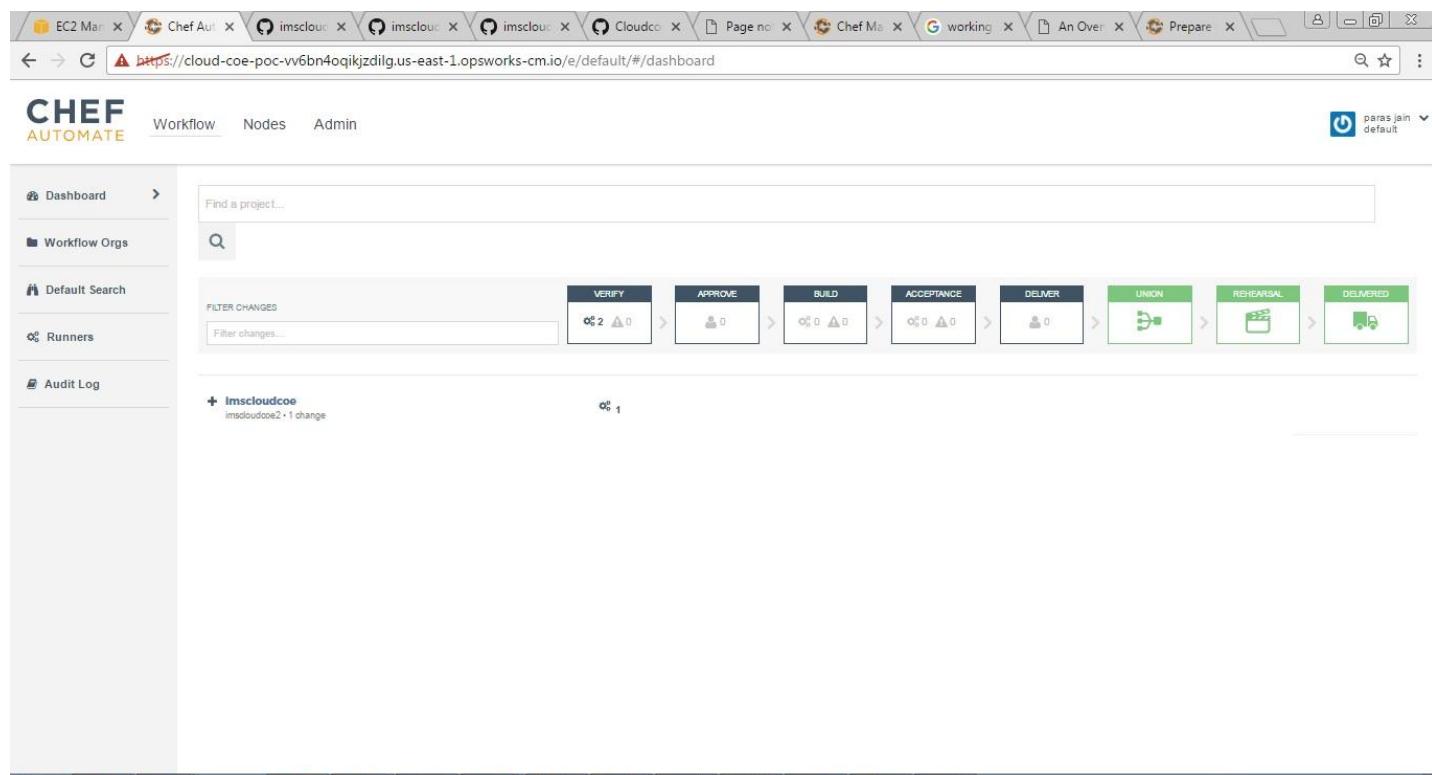
    echo "# abc" >> README.md
    echo ".delivery/cli.toml" >> .gitignore
    git add README.md .gitignore
    git commit -m "Initial commit"

5. Run delivery init , which will create an empty build cookbook for you (with an empty set of phase recipes), add the cookbook to your project, create the new pipeline and submit the project to Delivery for review.

    delivery init

```

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



The screenshot shows the Chef Automate dashboard. On the left, a sidebar lists Dashboard, Workflow Orgs, Default Search, Runners, and Audit Log. The main area displays a workflow pipeline with the following stages and counts of changes:

- VERIFY: 2 changes
- APPROVE: 0 changes
- BUILD: 0 changes
- ACCEPTANCE: 0 changes
- DELIVER: 0 changes
- UNION: 1 change
- REHEARSAL: 0 changes
- DELIVERED: 0 changes

A message at the bottom indicates there is 1 change in the UNION stage.

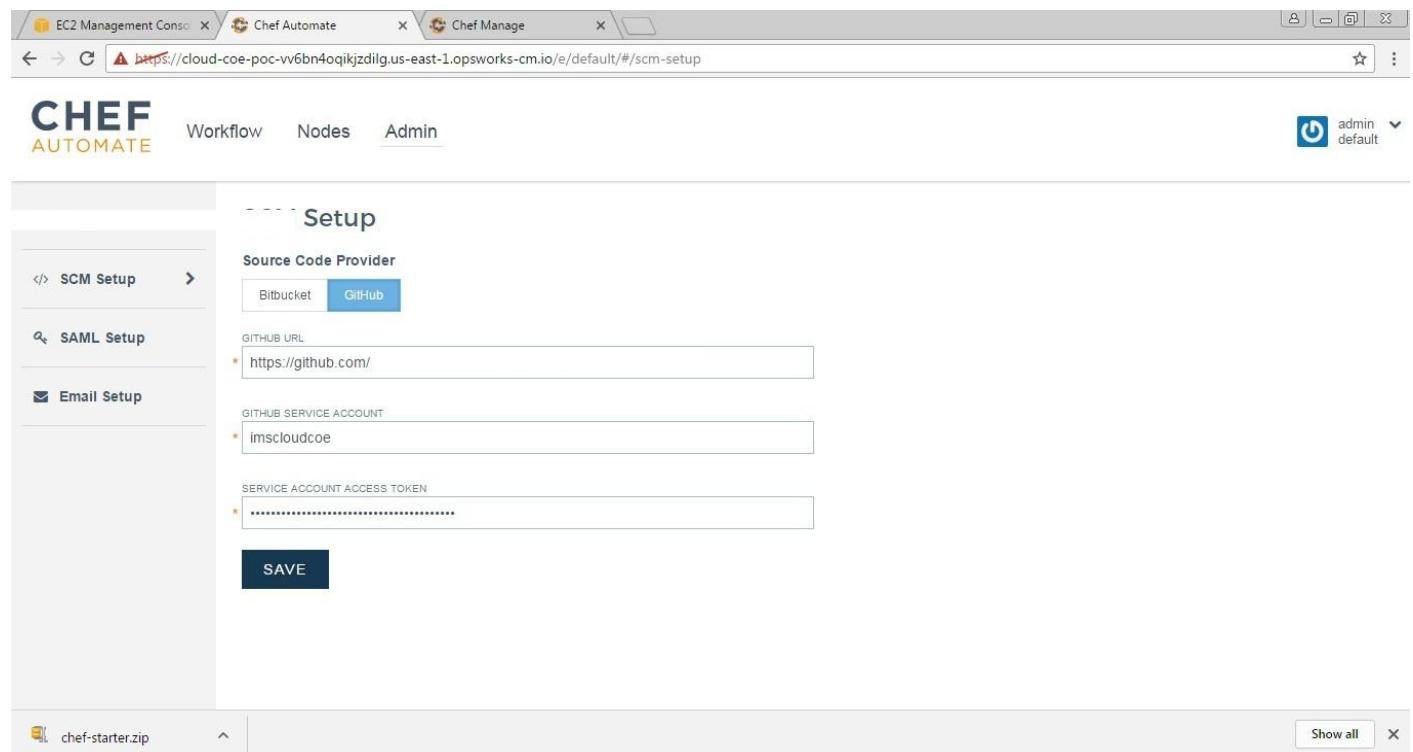
In this way, we can integrate the Git within the Chef Automate Server. We can also integrate the hosted GitHub with Chef automate Server as shown below.

GitHub Access

- GitHub Details
- Token Generation
- Sample Project

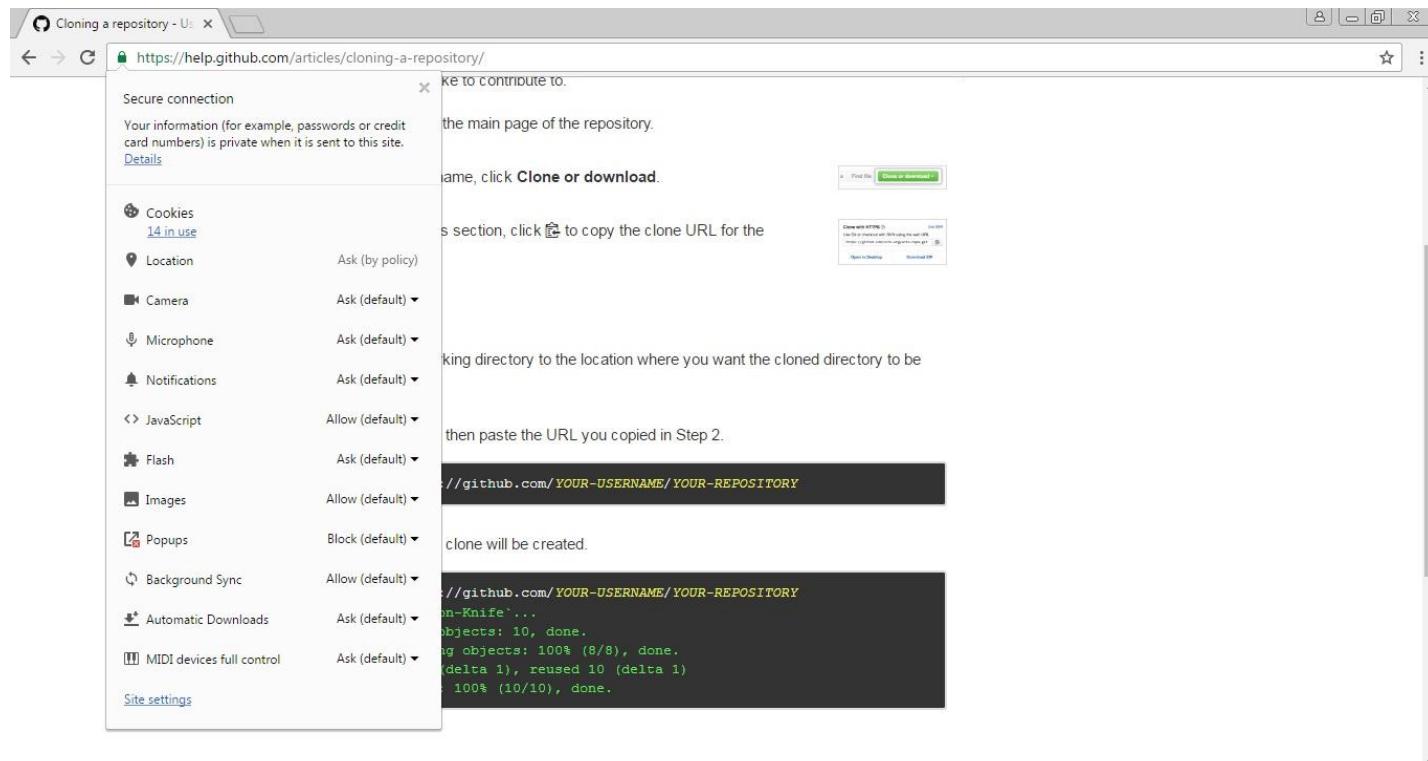
15.1.3.1 Integrating GitHub Repository (Hosted) with Chef Automate Server

- 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

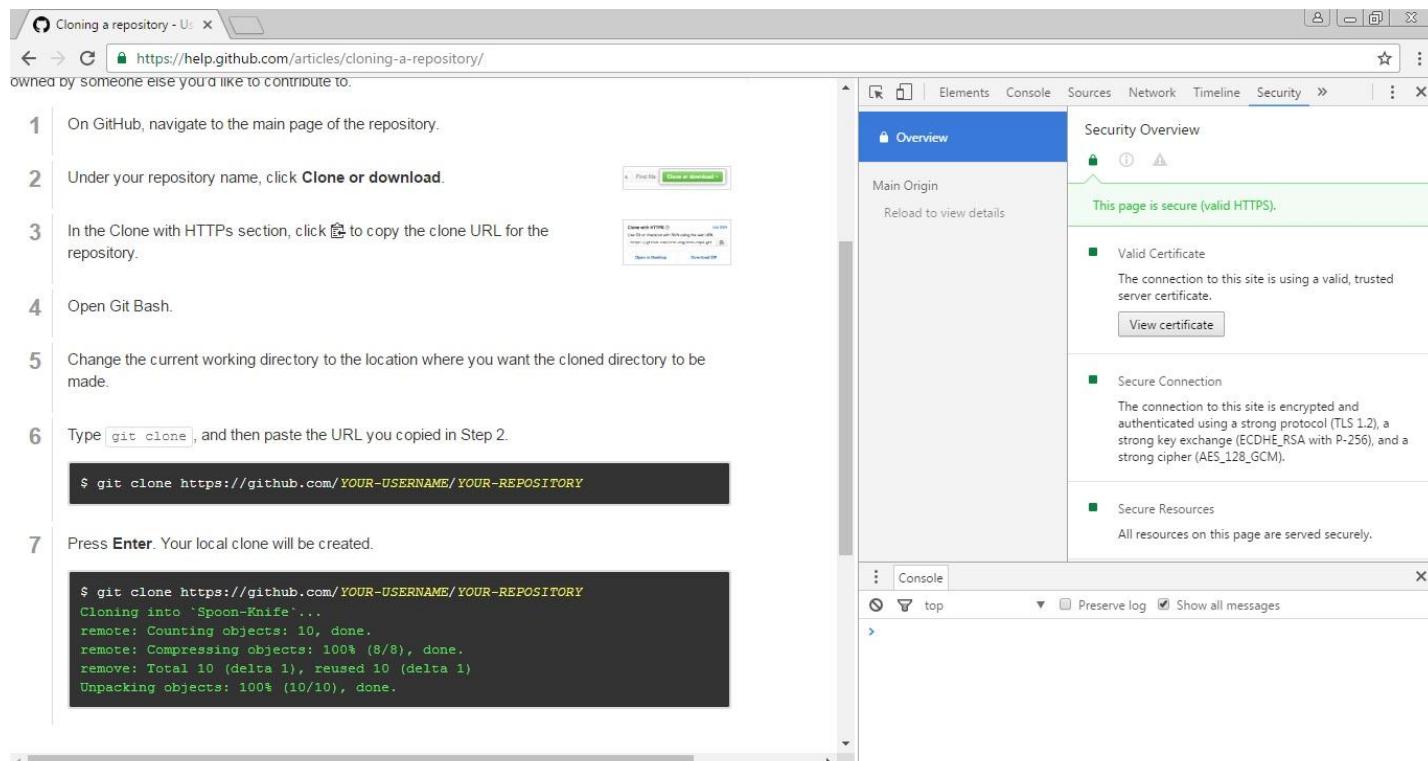


The screenshot shows the Chef Automate web interface. The URL in the browser is <https://cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io/e/default/#/scm-setup>. The page title is "Setup". On the left sidebar, there are three tabs: "SCM Setup" (selected), "SAML Setup", and "Email Setup". The main content area has a heading "Source Code Provider" with two options: "Bitbucket" and "GitHub" (which is selected). Below this, there are fields for "GITHUB URL" (set to "https://github.com/"), "GITHUB SERVICE ACCOUNT" (set to "imscloudcoe"), and "SERVICE ACCOUNT ACCESS TOKEN" (a redacted password). At the bottom of the form is a "SAVE" button. In the bottom right corner of the page, there is a download link for "chef-starter.zip".

- 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



1 On GitHub, navigate to the main page of the repository.

2 Under your repository name, click **Clone or download**.

3 In the Clone with HTTPS section, click to copy the clone URL for the repository.

4 Open Git Bash.

5 Change the current working directory to the location where you want the cloned directory to be made.

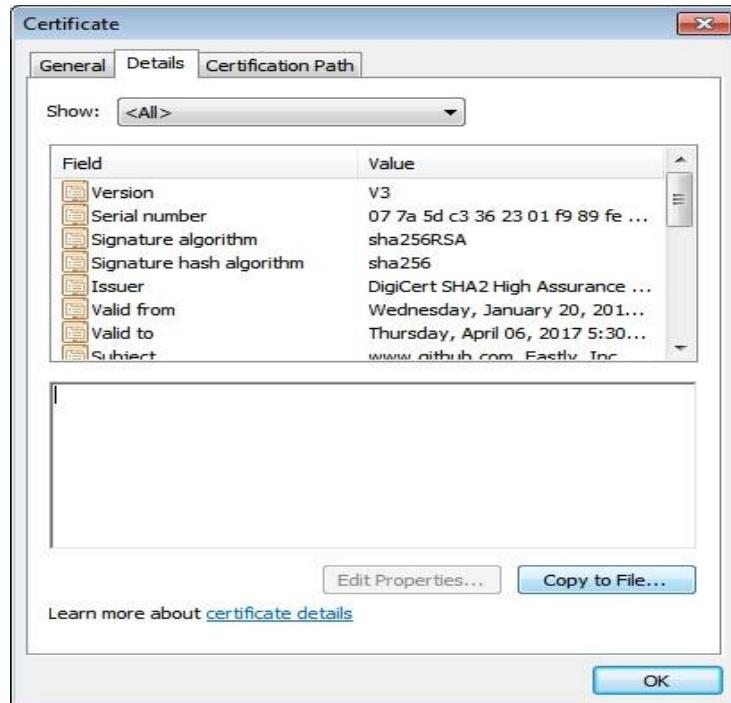
6 Type `git clone`, and then paste the URL you copied in Step 2.

```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
```

7 Press **Enter**. Your local clone will be created.

```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
Cloning into 'Spoon-Knife'...
remote: Counting objects: 10, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 1), reused 10 (delta 1)
Unpacking objects: 100% (10/10), done.
```

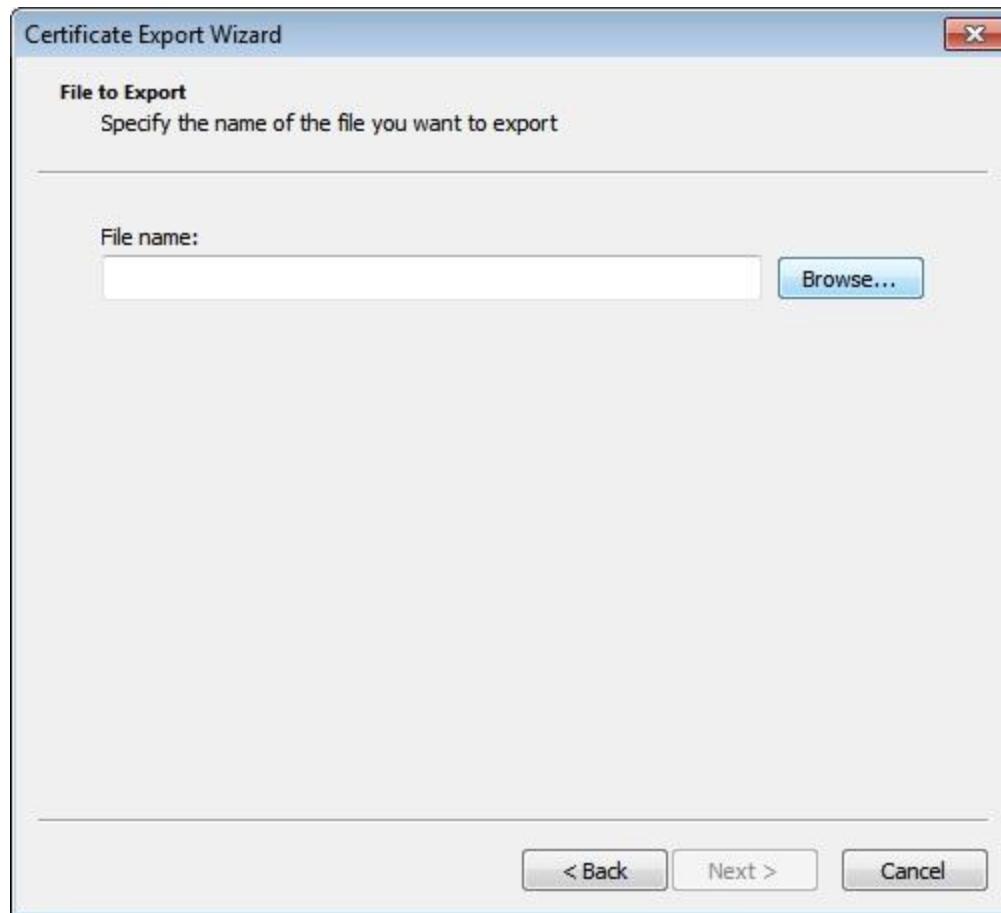
- Now click on **Copy to file** button



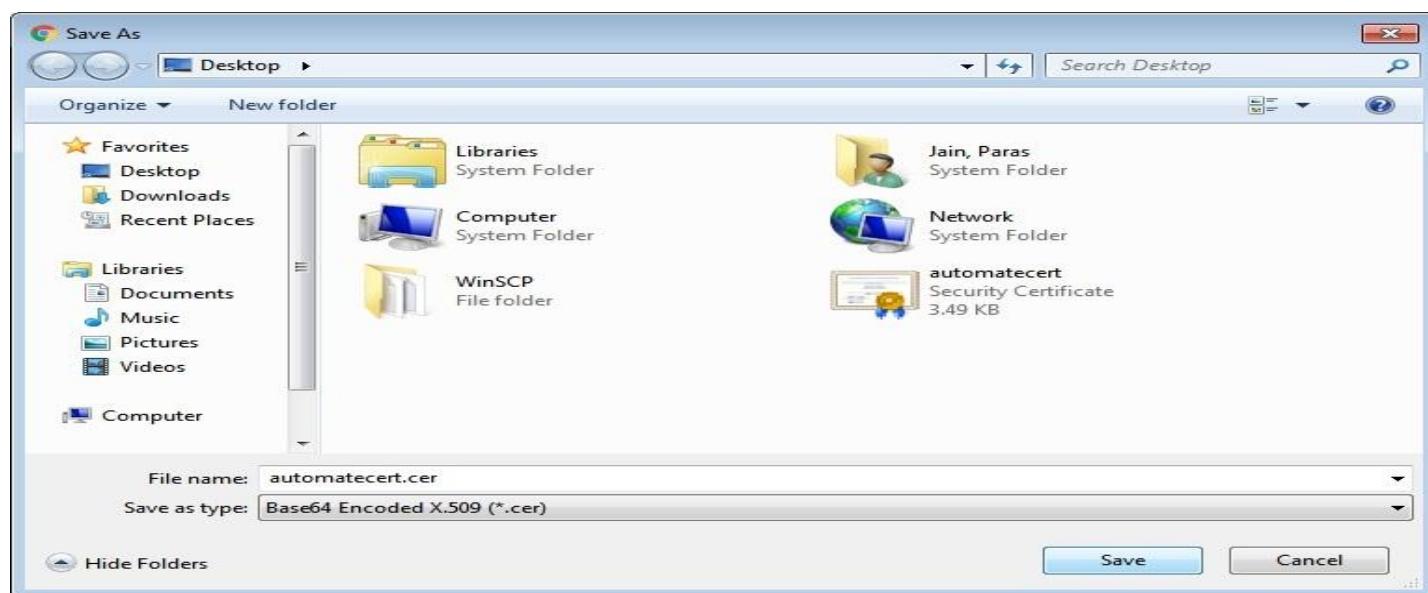
- Select **Base-64 encoded X.509 (.CER)** radio button , click on **Next** button



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



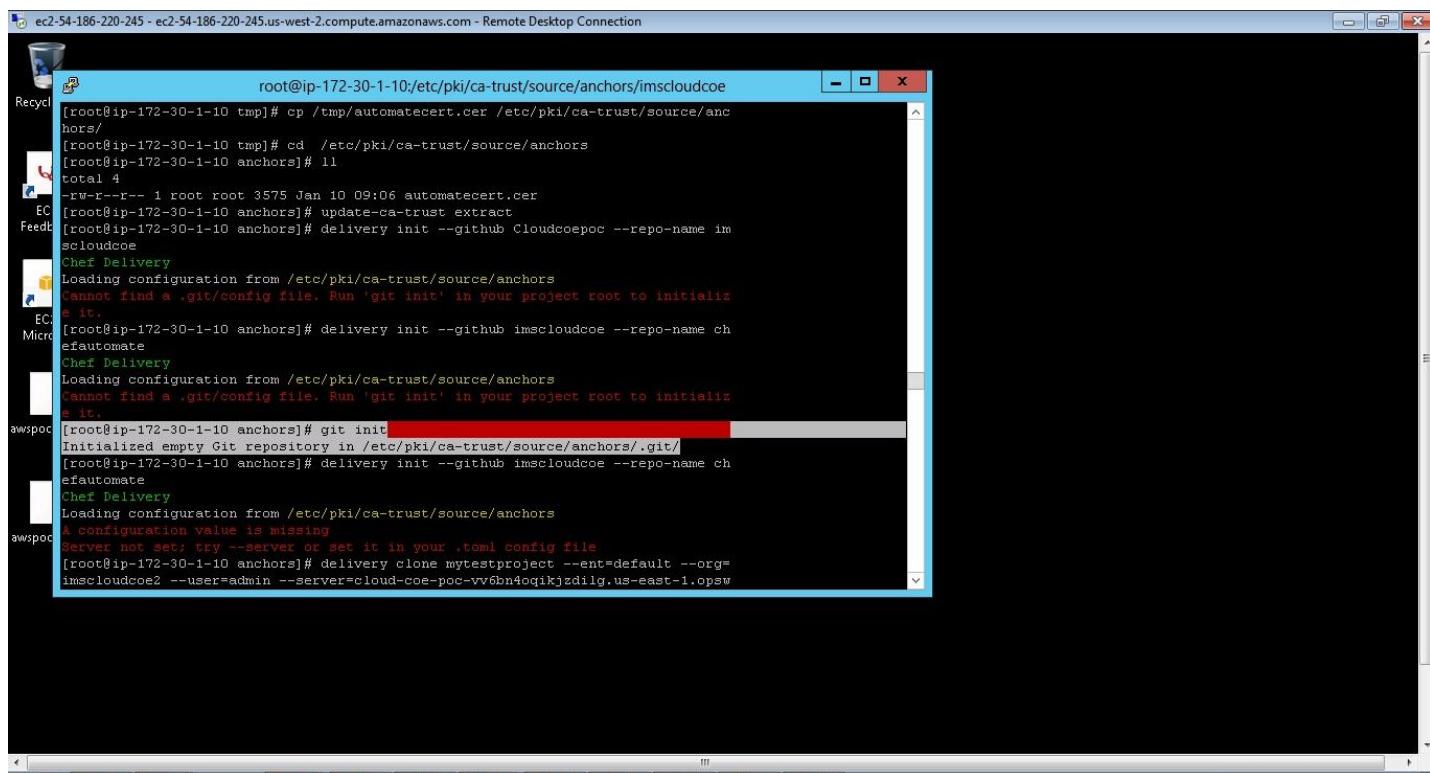
- Provide the **Filename** and click on **Save** Button



- 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
```



```
[root@ip-172-30-1-10 tmp]# cp /tmp/automatecert.cer /etc/pki/ca-trust/source/anchors/
[root@ip-172-30-1-10 tmp]# cd /etc/pki/ca-trust/source/anchors
[root@ip-172-30-1-10 anchors]# 11
total 4
-rw-r--r-- 1 root root 3575 Jan 10 09:06 automatecert.cer
[root@ip-172-30-1-10 anchors]# update-ca-trust extract
[root@ip-172-30-1-10 anchors]# delivery init --github Cloudcoepoc --repo-name imscloudcoe
Chef Delivery
Loading configuration from /etc/pki/ca-trust/source/anchors
cannot find a .git/config file. Run 'git init' in your project root to initialize it.
[root@ip-172-30-1-10 anchors]# delivery init --github imscloudcoe --repo-name chefautomate
Chef Delivery
Loading configuration from /etc/pki/ca-trust/source/anchors
cannot find a .git/config file. Run 'git init' in your project root to initialize it.
[root@ip-172-30-1-10 anchors]# git init
Initialized empty Git repository in /etc/pki/ca-trust/source/anchors/.git/
[root@ip-172-30-1-10 anchors]# delivery init --github imscloudcoe --repo-name chefautomate
Chef Delivery
Loading configuration from /etc/pki/ca-trust/source/anchors
A configuration value is missing
Server not set: try --server or set it in your .toml config file
[root@ip-172-30-1-10 anchors]# delivery clone mytestprotect --ent=default --org=imscloudcoe2 --user=admin --server=cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io
```

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

ec2-54-186-220-245 - ec2-54-186-220-245.us-west-2.compute.amazonaws.com - Remote Desktop Connection

```

root@ip-172-30-1-10:/etc/pki/ca-trust/source/anchors/imscloudcoe
Loading configuration from /etc/pki/ca-trust/source/anchors
Cannot find a .git/config file. Run 'git init' in your project root to initialize it.
[root@ip-172-30-1-10 anchors]# git init
Initialized empty Git repository in /etc/pki/ca-trust/source/anchors/.git/
[root@ip-172-30-1-10 anchors]# delivery init --github imscloudcoe --repo-name chef-automate
Chef Delivery
Loading configuration from /etc/pki/ca-trust/source/anchors
A configuration value is missing
Server not set; try --server or set it in your .toml config file.
[root@ip-172-30-1-10 anchors]# delivery clone mytestproject --ent=default --org=imscloudcoe2 --user=admin --server=cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io:8989/default/imscloudcoe2/mytestproject to mytestproject
[root@ip-172-30-1-10 anchors]# chef -version
invalid option: -ersion
Usage:
  chef -h--help
  chef -v--version
  chef command [arguments...] [options...]
Available Commands:
  exec      Runs the command in context of the embedded ruby
  env       Prints environment variables used by ChefDK

```

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

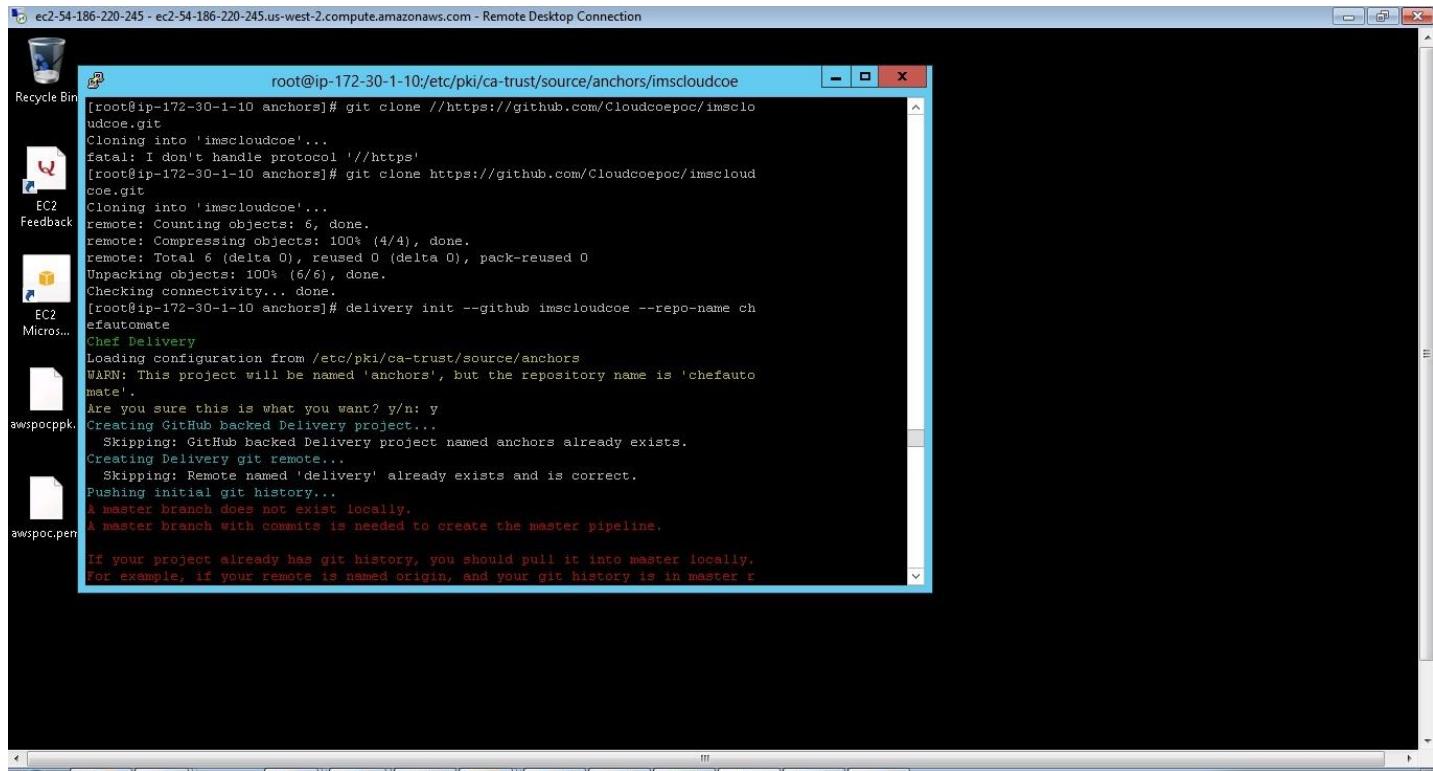
ec2-54-186-220-245 - ec2-54-186-220-245.us-west-2.compute.amazonaws.com - Remote Desktop Connection

```

root@ip-172-30-1-10:/etc/pki/ca-trust/source/anchors/imscloudcoe
[root@ip-172-30-1-10 anchors]# chef --version
Chef Development Kit Version: 1.1.16
chef-client version: 12.17.44
delivery version: master (83358fb62c0f711c70ad5a81030a6cae4017f103)
berks version: 5.2.0
kitchen version: 1.14.2
[root@ip-172-30-1-10 anchors]# delivery setup --ent=default --org=imscloudcoe2 --user=admin --server=cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io
Chef Delivery
Loading configuration from /etc/pki/ca-trust/source/anchors
Writing configuration to /etc/pki/ca-trust/source/anchors/.delivery/cli.toml
New configuration
-----
api_protocol = "https"
enterprise = "default"
git_port = "8989"
organization = "imscloudcoe2"
pipeline = "master"
server = "cloud-coe-poc-vv6bn4oqikjzdilg.us-east-1.opsworks-cm.io"
user = "admin"
[root@ip-172-30-1-10 anchors]# delivery init --github imscloudcoe --repo-name chef-automate
Chef Delivery
Loading configuration from /etc/pki/ca-trust/source/anchors
WARN: This project will be named 'anchors', but the repository name is 'chefautomate'!
Are you sure this is what you want? y/n: y
Creating GitHub backed Delivery project...
GitHub backed Delivery project named anchors created.

```

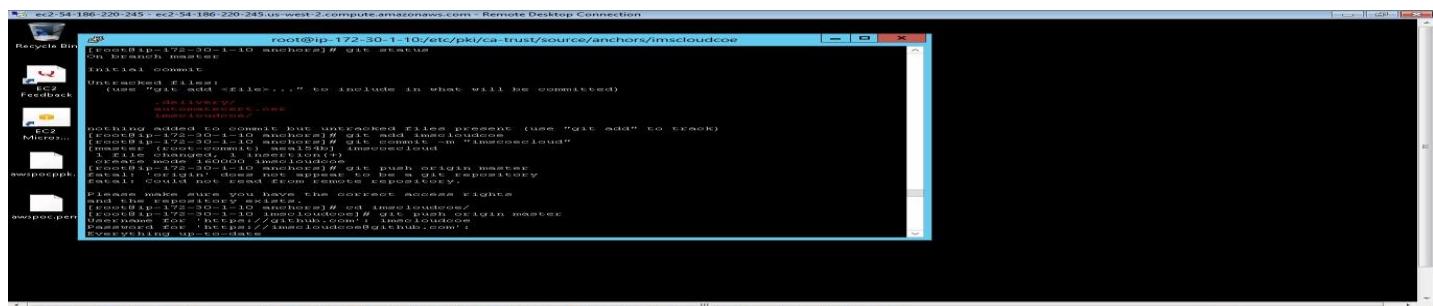
```
git clone https://github.com/Cloudcoepoc/imscloudcoe.git
delivery init --github Cloudcoepoc --repo-name imscloudcoe
```



```
root@ip-172-30-1-10:/etc/pki/ca-trust/source/anchors/imscloudcoe
[root@ip-172-30-1-10 anchors]# git clone //https://github.com/Cloudcoepoc/imscloudcoe.git
Cloning into 'imscloudcoe'...
fatal: I don't handle protocol '//https'
[root@ip-172-30-1-10 anchors]# git clone https://github.com/Cloudcoepoc/imscloudcoe.git
Cloning into 'imscloudcoe'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.
[root@ip-172-30-1-10 anchors]# delivery init --github imscloudcoe --repo-name chefautomate
Chef Delivery
Loading configuration from /etc/pki/ca-trust/source/anchors
WARN: This project will be named 'anchors', but the repository name is 'chefautomate'.
Are you sure this is what you want? y/n: y
Creating GitHub backed Delivery project...
  Skipping: GitHub backed Delivery project named anchors already exists.
Creating Delivery git remote...
  Skipping: Remote named 'delivery' already exists and is correct.
Pushing initial git history...
A master branch does not exist locally.
A master branch with commits is needed to create the master pipeline.

If your project already has git history, you should pull it into master locally.
For example, if your remote is named origin, and your git history is in master r
```

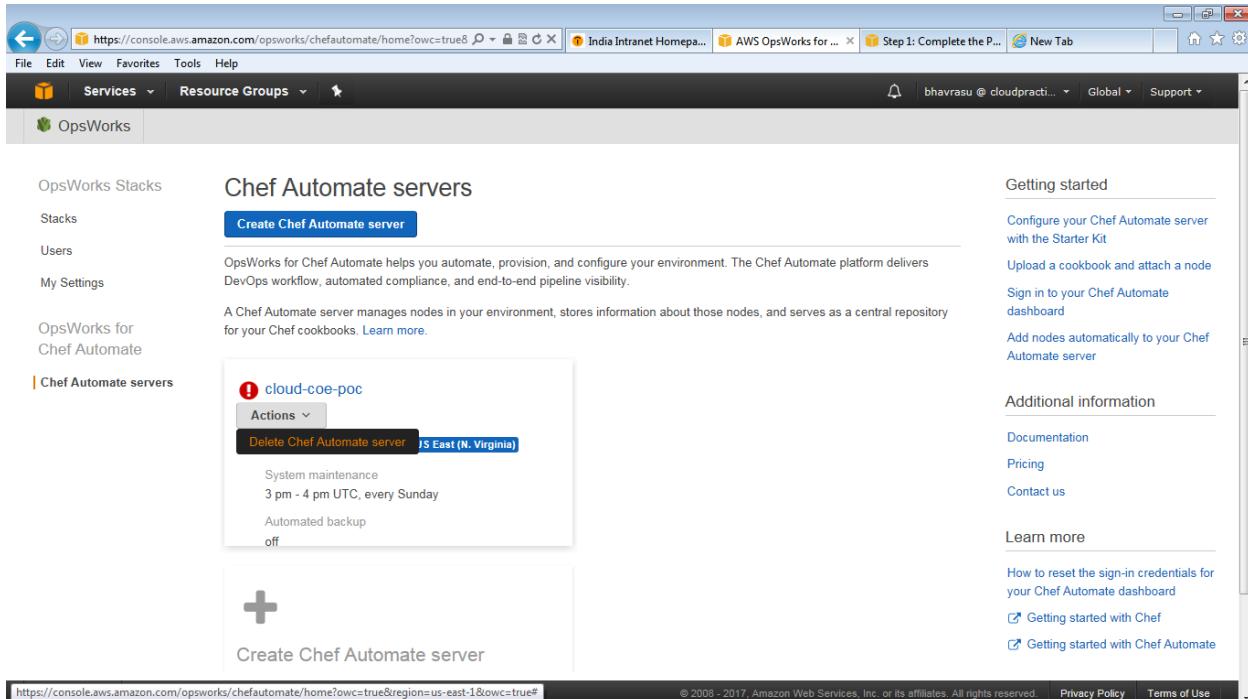
```
git status
git add imscloudcoe
git commit -m "imscoecloud"
git push origin master
```



```
root@ip-172-30-1-10:/etc/pki/ca-trust/source/anchors/imscloudcoe
[root@ip-172-30-1-10 anchors]# git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .chefignore
    .gitignore
nothing to commit but untracked files present (use "git add" to track)
[root@ip-172-30-1-10 anchors]# git add imscloudcoe
[root@ip-172-30-1-10 anchors]# git commit -m "imscoecloud"
[master (root-commit) aee154b] imscoecloud
 1 file changed, 10000 insertions(+)
 create mode 10000 imscloudcoe
[root@ip-172-30-1-10 anchors]# git push origin master
Error! Couldn't find any git repository
Fatal: Couldn't find any git repository
Please make sure you have the correct access rights
[root@ip-172-30-1-10 anchors]# cd imscloudcoe/
[root@ip-172-30-1-10 imscloudcoe]# git push origin master
Username for 'https://github.com': imscloudcoe
Password for 'https://imscloudcoe@github.com': 
Everything up-to-date
```

```
git status
Files are pushed to Chef Automated Server
```

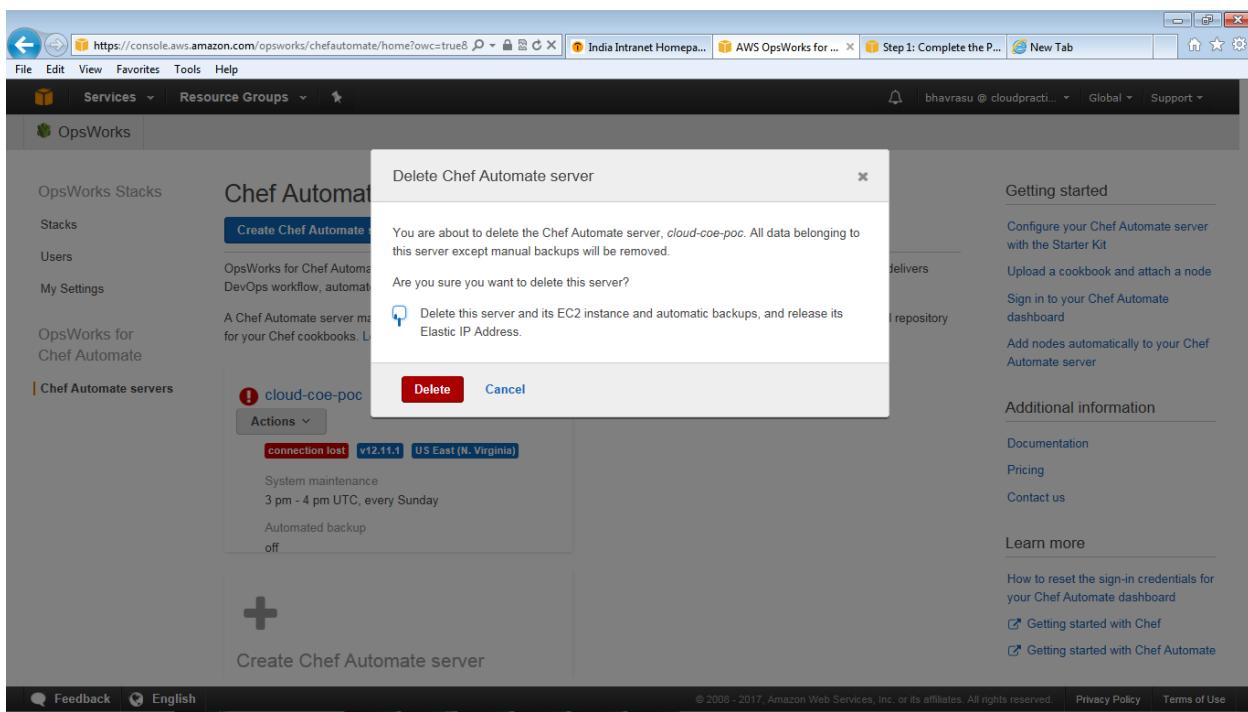
15.1.3.4 Delete Chef Automate Server



The screenshot shows the AWS OpsWorks Chef Automate server list page. On the left sidebar, under 'Chef Automate servers', there is a list item for 'cloud-coe-poc'. In the main content area, there is a card for this server with the following details:

- Status:** ! cloud-coe-poc
- Actions:** Actions ▾
- Delete Chef Automate server**
- Region:** US East (N. Virginia)
- System maintenance:** 3 pm - 4 pm UTC, every Sunday
- Automated backup:** off

To the right of the server list, there is a 'Getting started' sidebar with links to configure the server, upload cookbooks, sign in to the dashboard, and add nodes automatically. Below that is an 'Additional information' sidebar with links to documentation, pricing, and contact us. At the bottom, there is a 'Learn more' section with links to reset sign-in credentials, get started with Chef, and get started with Chef Automate.



The screenshot shows a confirmation dialog box titled 'Delete Chef Automate server' overlying the main page. The dialog contains the following text:

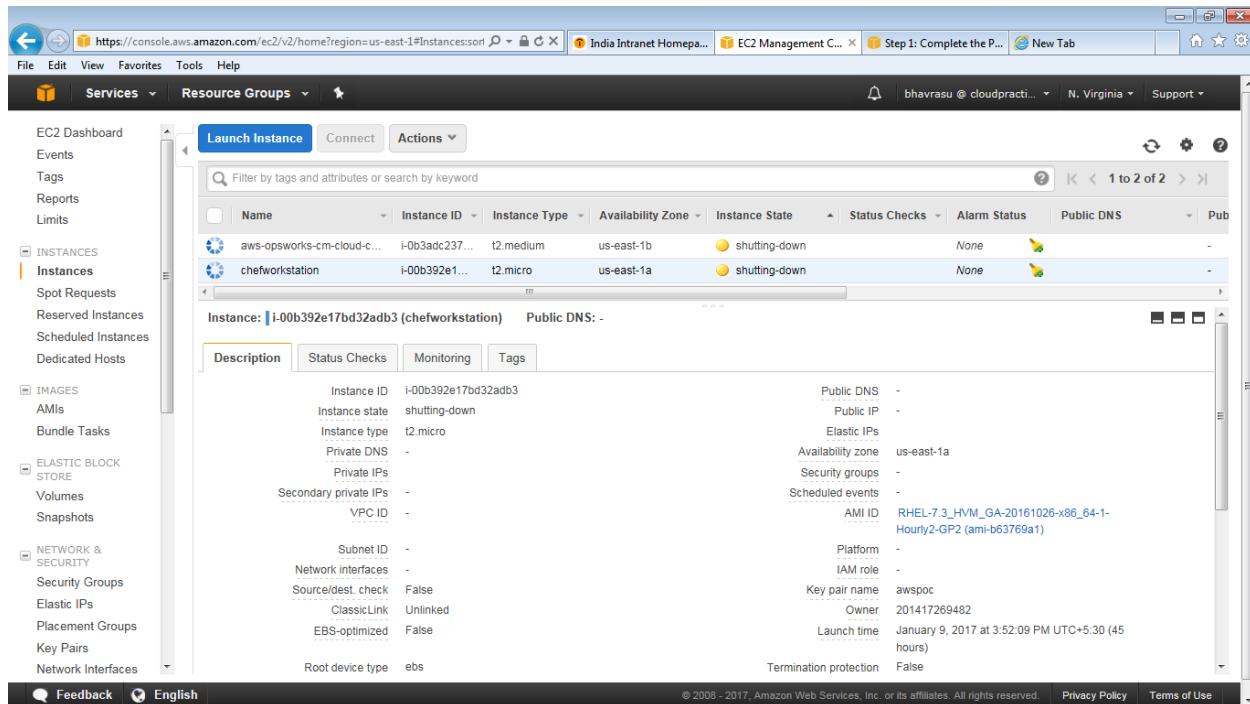
You are about to delete the Chef Automate server, *cloud-coe-poc*. All data belonging to this server except manual backups will be removed.

Are you sure you want to delete this server?

Delete this server and its EC2 instance and automatic backups, and release its Elastic IP Address.

At the bottom of the dialog are two buttons: a red 'Delete' button and a blue 'Cancel' button.

The background page shows the same server list and sidebar as the previous screenshot, with the 'Delete' button being the primary focus.



The screenshot shows the AWS EC2 Management Console interface. The left sidebar lists various services: EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, and Network Interfaces. The main content area is titled 'Instances' and shows two instances: 'aws-opsworks-cm-cloud-c...' and 'chefworkstation'. The 'chefworkstation' instance is selected, and its details are displayed in a large table. The table includes columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, and several other fields like Private DNS, Public IP, and AMI ID.

15.2 AWS OpsWork Stacks

15.2.1 Objective

Understand the process to create AWS OpsWork Stacks

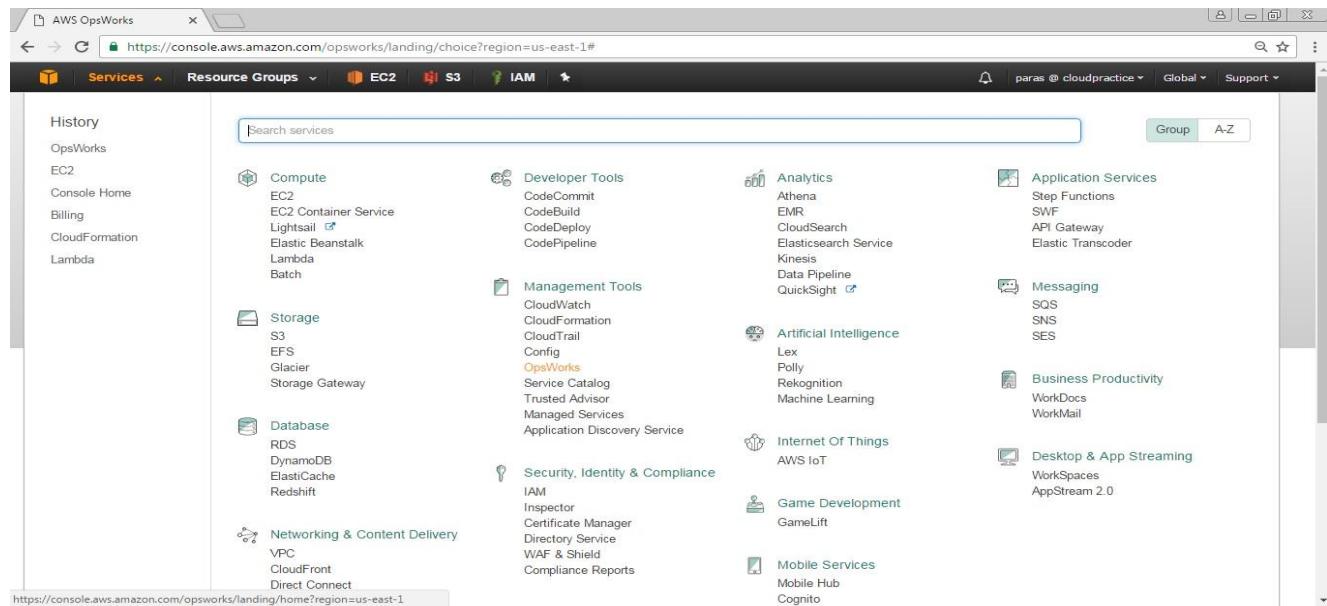
15.2.2 Pre-requisite

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

15.2.3 Procedure

15.2.3.1 Create a Stack

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



The screenshot shows the AWS OpsWorks console interface. On the left, there's a sidebar with links like History, OpsWorks, EC2, Console Home, Billing, CloudFormation, and Lambda. The main area is titled "Search services" and lists various AWS services under categories: Compute (EC2, EC2 Container Service, Lightsail, Elastic Beanstalk, Lambda, Batch), Storage (S3, EFS, Glacier, Storage Gateway), Database (RDS, DynamoDB, ElastiCache, Redshift), Networking & Content Delivery (VPC, CloudFront, Direct Connect), Developer Tools (CodeCommit, CodeBuild, CodeDeploy, CodePipeline), Management Tools (CloudWatch, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog, Trusted Advisor, Managed Services, Application Discovery Service), Security, Identity & Compliance (IAM, Inspector, Certificate Manager, Directory Service, WAF & Shield, Compliance Reports), Analytics (Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, Data Pipeline, QuickSight), Application Services (Step Functions, SWF, API Gateway, Elastic Transcoder), Messaging (SQS, SNS, SES), Artificial Intelligence (Lex, Polly, Rekognition, Machine Learning), Internet Of Things (AWS IoT), Game Development (GameLift), Mobile Services (Mobile Hub, Cognito), Business Productivity (WorkDocs, WorkMail), Desktop & App Streaming (WorkSpaces, AppStream 2.0).

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



The screenshot shows the same AWS OpsWorks console as above, but the focus is on the "Go to OpsWorks Stacks" button located at the bottom of the main content area.

AWS OpsWorks provides two solutions to configure your infrastructure:



OpsWorks Stacks

Define, group, provision, deploy, and operate your applications in AWS by using Chef in local mode.

[Go to OpsWorks Stacks](#)

[Learn more about OpsWorks Stacks](#)



OpsWorks for Chef Automate

Create Chef servers that include Chef Automate premium features, and use the Chef DK or any Chef tooling to manage them.

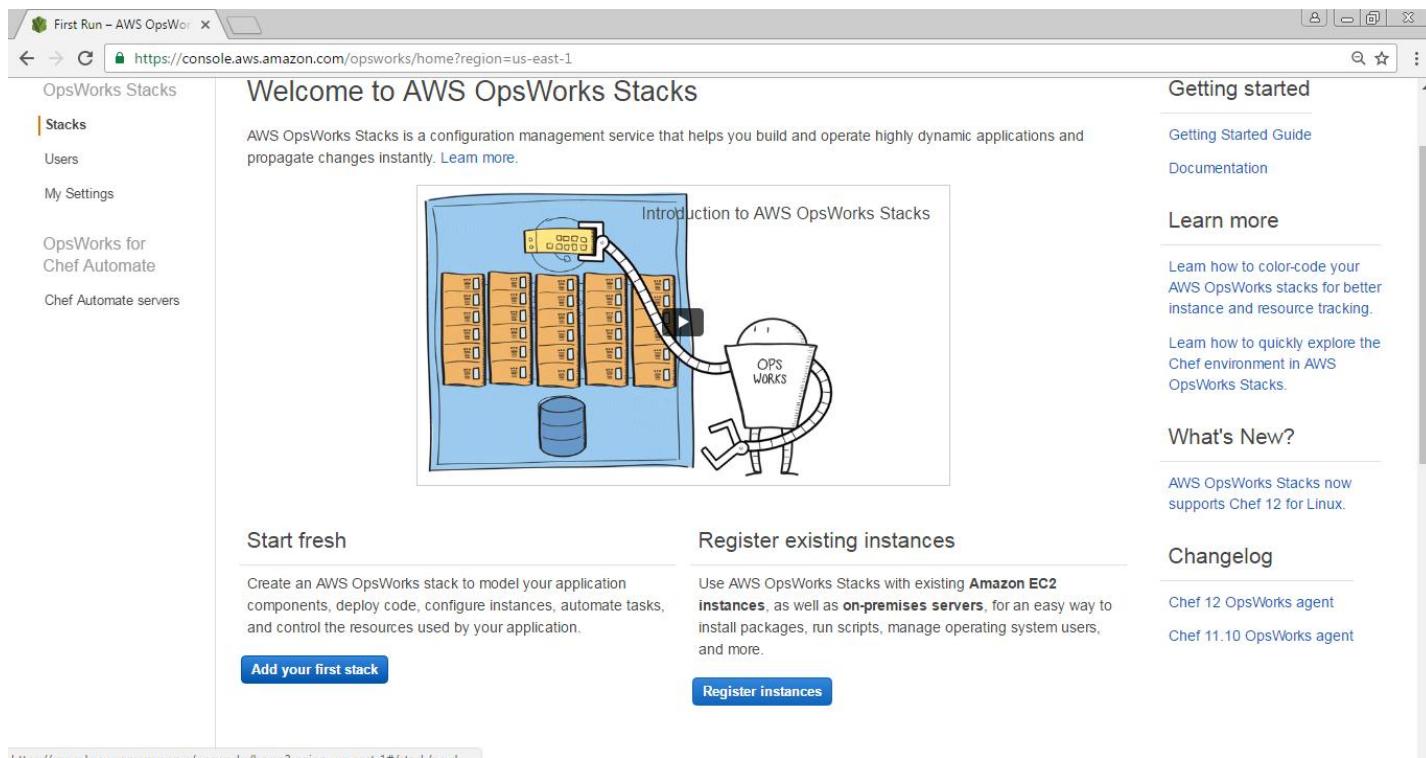
[Go to OpsWorks for Chef Automate](#)

[Learn more about OpsWorks for Chef Automate](#)

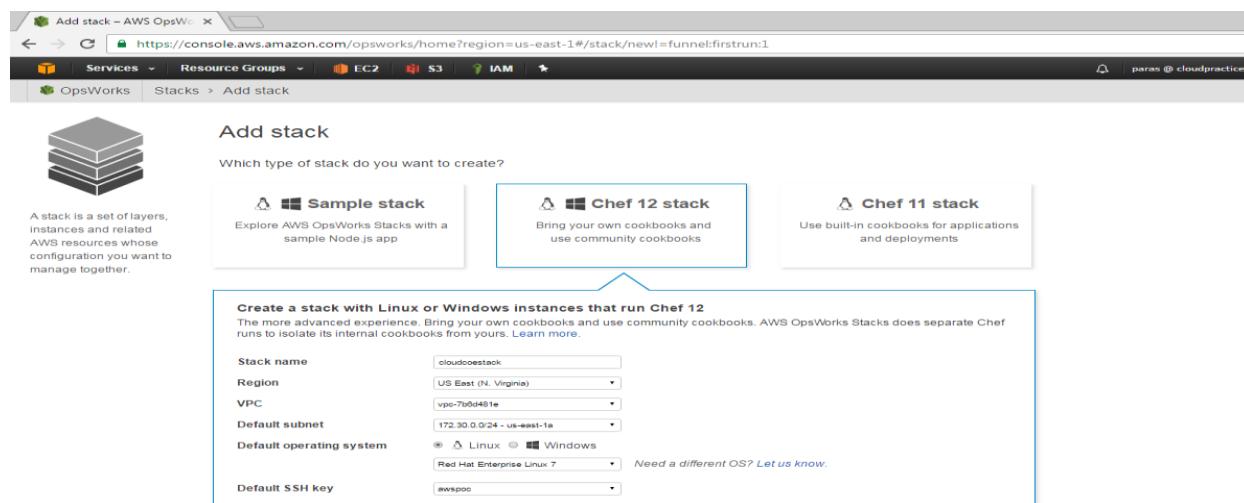
AWS OpsWorks documentation and support

[OpsWorks Stacks documentation](#) | [OpsWorks for Chef Automate documentation](#) | [Forum](#)

- 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 **clouddcoe**

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**

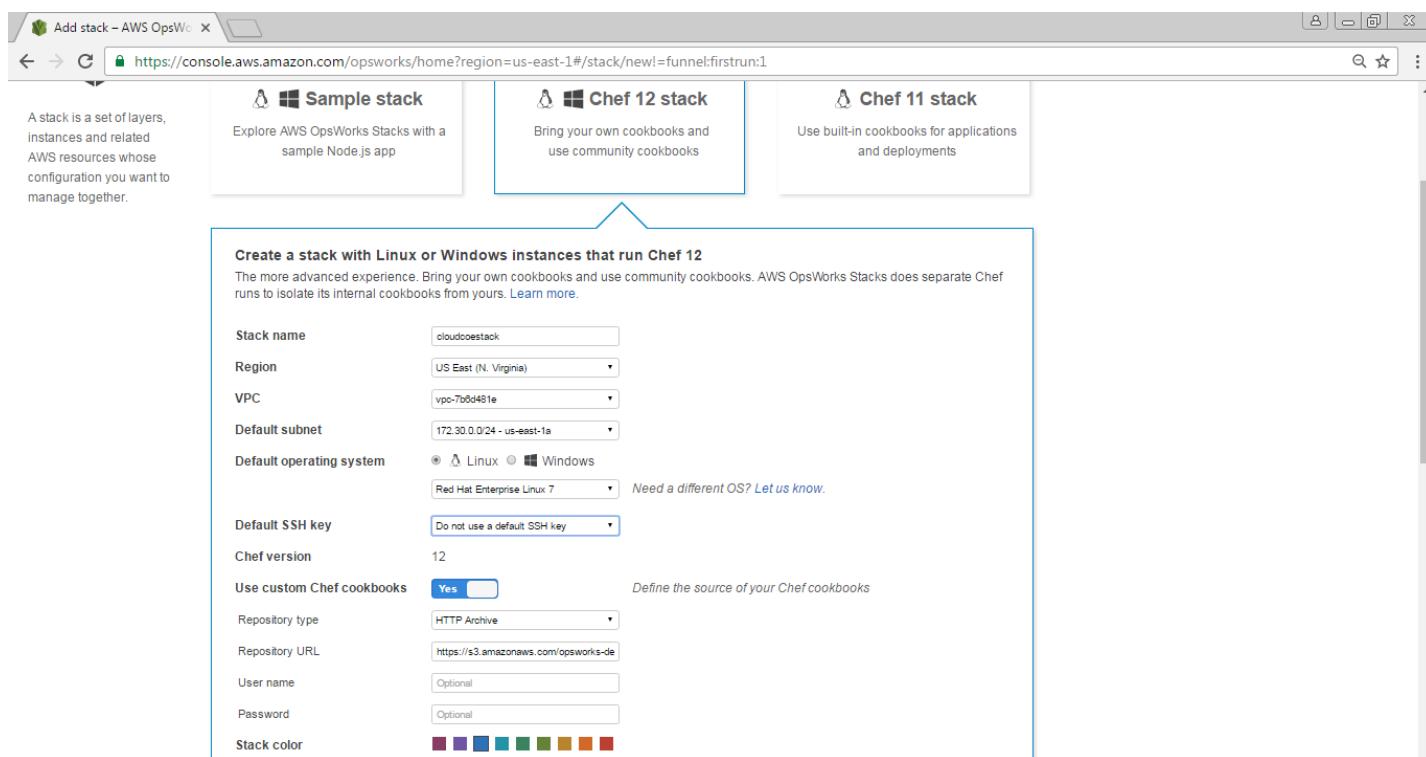
For **Use custom Chef cookbooks**, choose **Yes**

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

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

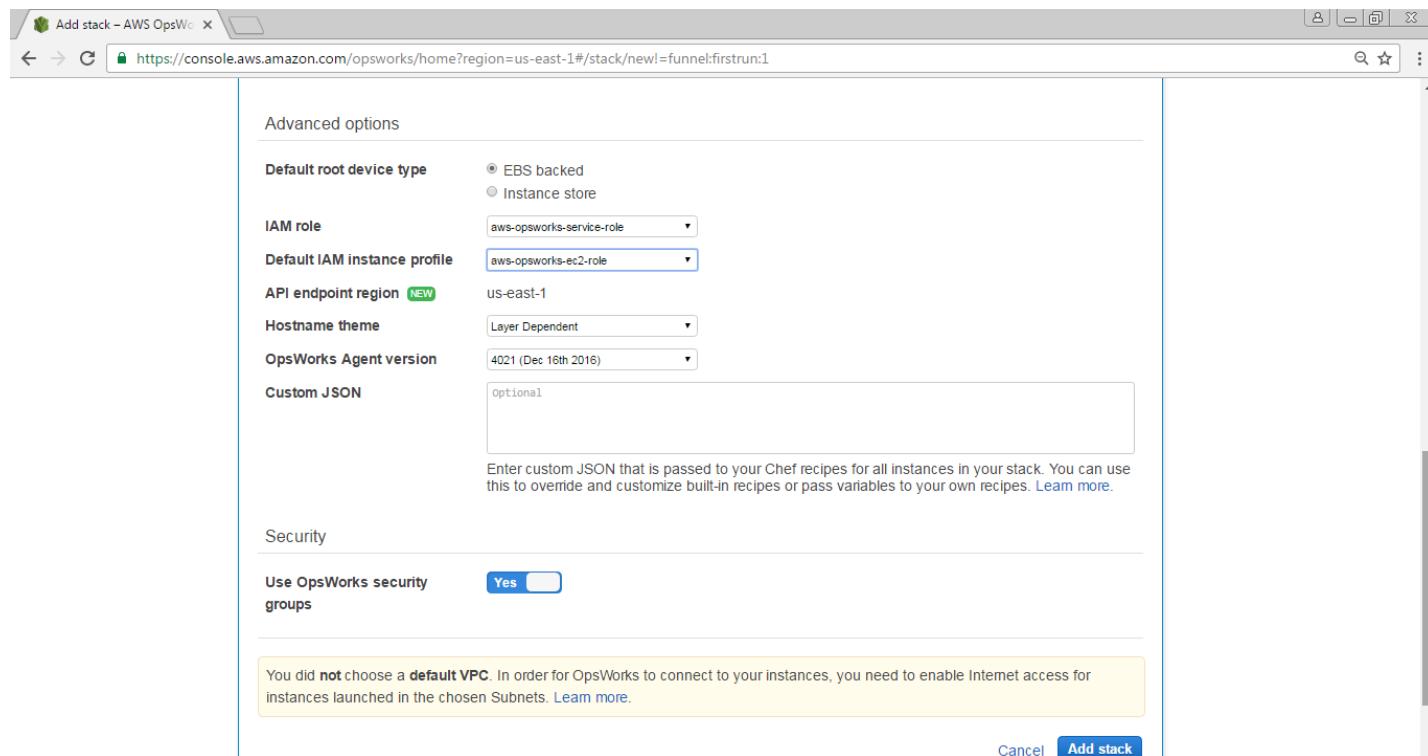
For **User name** and **Password** leave the textbox blank

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



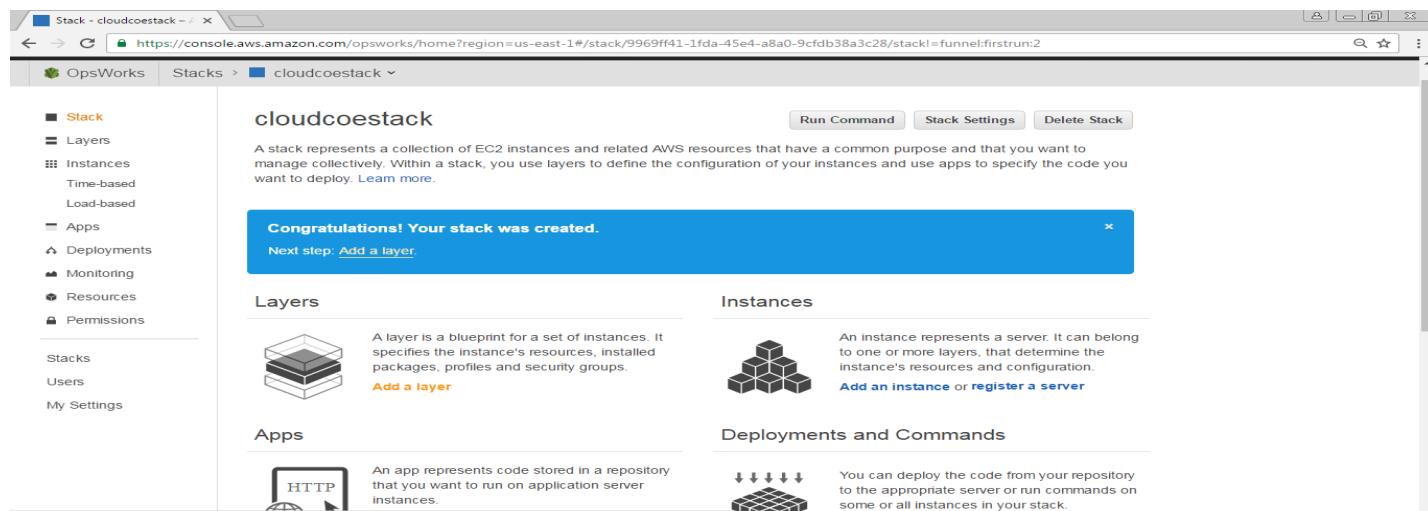
- Choose **Advanced Options**, do the following
 - ✓ For **Default root device type** select **EBS backed**
 - ✓ For **IAM role**: If **aws-opsworks-service-role** is available choose it else choose **New IAM role**
 - ✓ For **Default IAM instance profile**: If **aws-opsworks-ec2-role** is available, 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

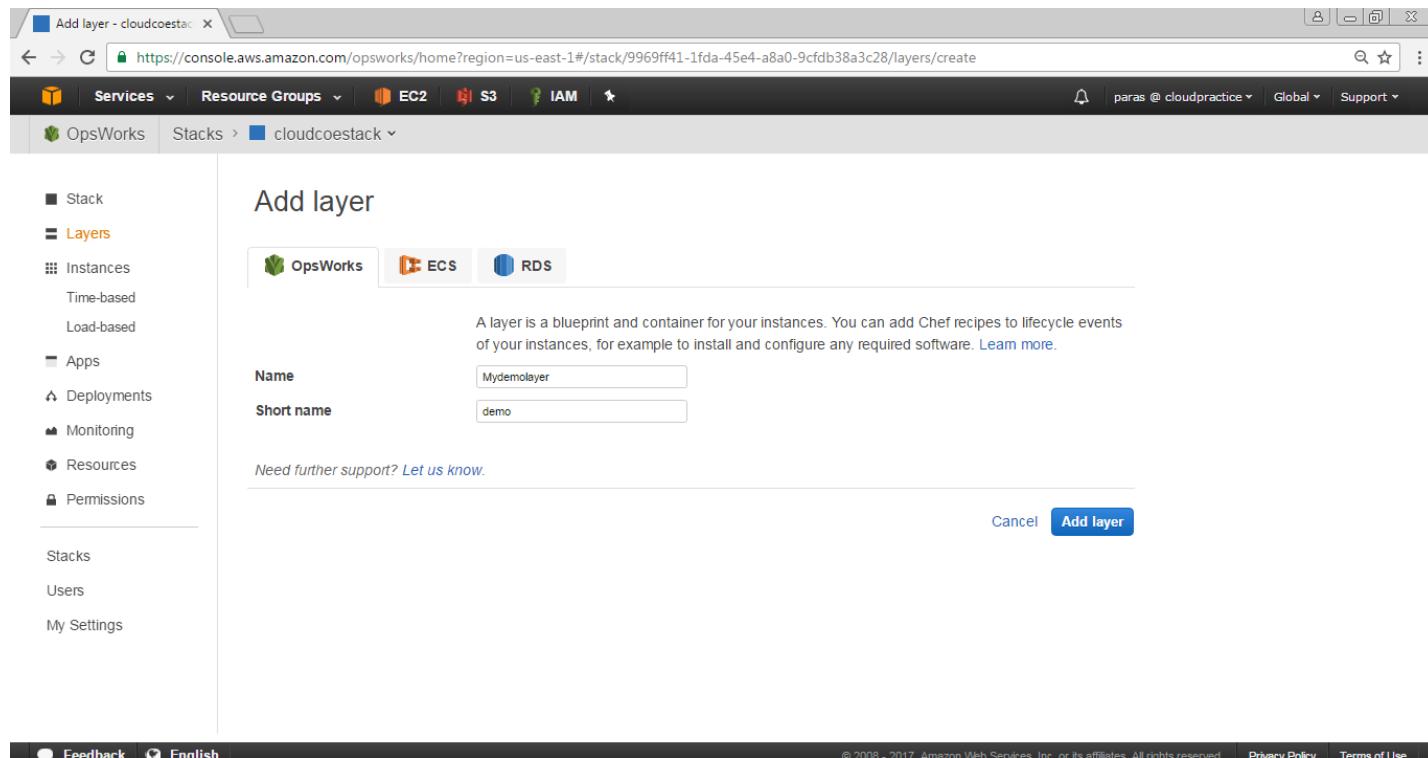


15.2.3.2 Add a Layer to the Stack

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

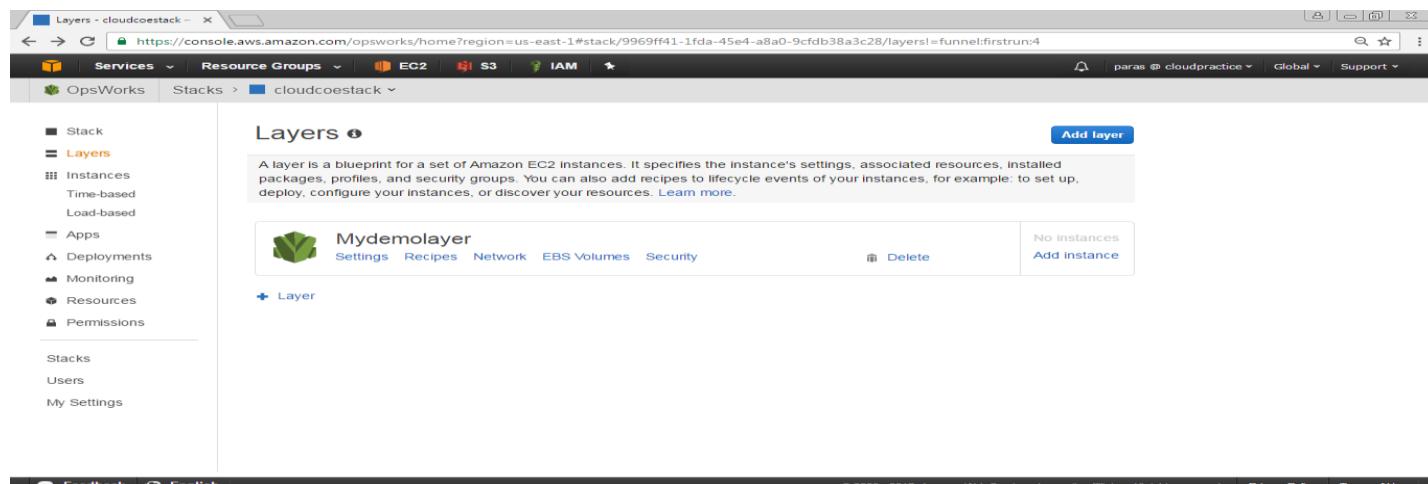


- 16 The Add Layer page is displayed. On the OpsWorks tab, for Name, type **Mydemolayer**. For **Short name**, type **demo** and click on **Add layer**



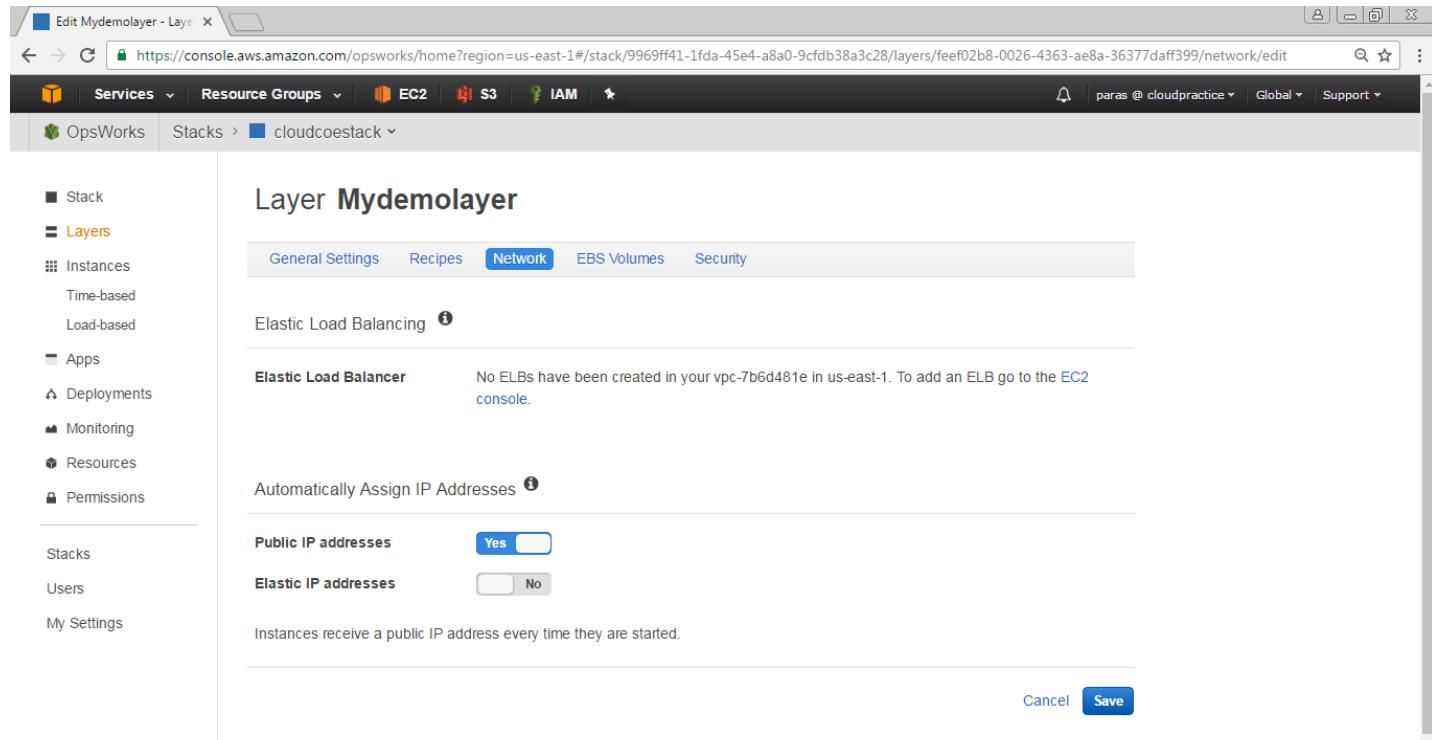
The screenshot shows the AWS OpsWorks console with the URL <https://console.aws.amazon.com/opsworks/home?region=us-east-1#stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/layers/create>. The left sidebar is open with the 'Layers' section selected. The main area is titled 'Add layer' and contains fields for 'Name' (Mydemolayer) and 'Short name' (demo). Below the fields is a link 'Need further support? Let us know.' At the bottom right are 'Cancel' and 'Add layer' buttons, with 'Add layer' being the active one.

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



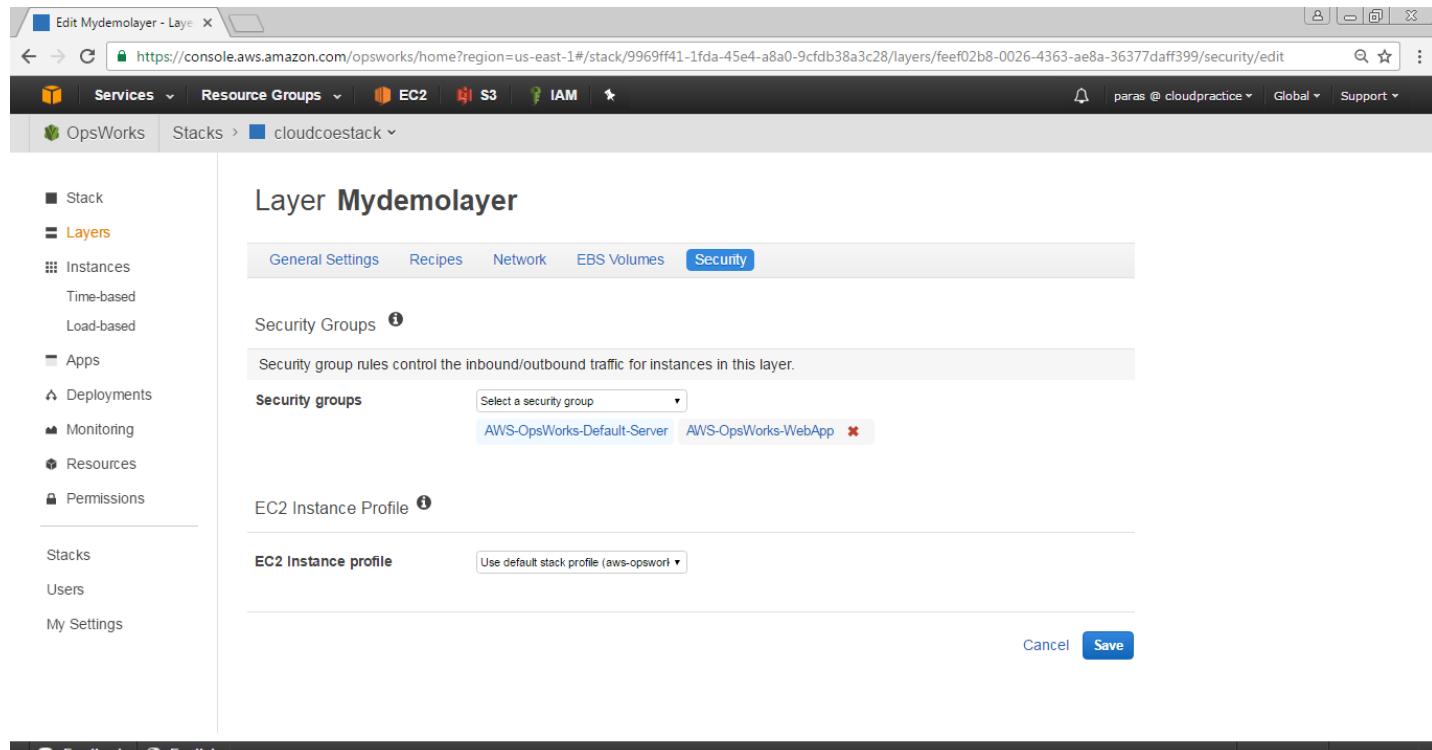
The screenshot shows the AWS OpsWorks console with the URL <https://console.aws.amazon.com/opsworks/home?region=us-east-1#stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/layers/funnel:firstrun:4>. The left sidebar is open with the 'Layers' section selected. The main area is titled 'Layers' and lists a single layer named 'Mydemolayer'. The 'Network' tab is selected, showing options for 'Automatically Assign IP Addresses' (set to 'Yes') and other network settings. At the bottom of the tab, there is a 'Save' button.

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



The screenshot shows the AWS OpsWorks console with the URL <https://console.aws.amazon.com/opsworks/home?region=us-east-1#/stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/layers/feef02b8-0026-4363-ae8a-36377daff399/network/edit>. The left sidebar shows the navigation menu with 'Layers' selected. The main content area is titled 'Layer Mydemolayer' and has tabs for General Settings, Recipes, Network (which is selected), EBS Volumes, and Security. Under the Network tab, there are sections for 'Elastic Load Balancing' and 'Automatically Assign IP Addresses'. The 'Public IP addresses' switch is set to 'Yes' and the 'Elastic IP addresses' switch is set to 'No'. At the bottom right of the form, there are 'Cancel' and 'Save' buttons.

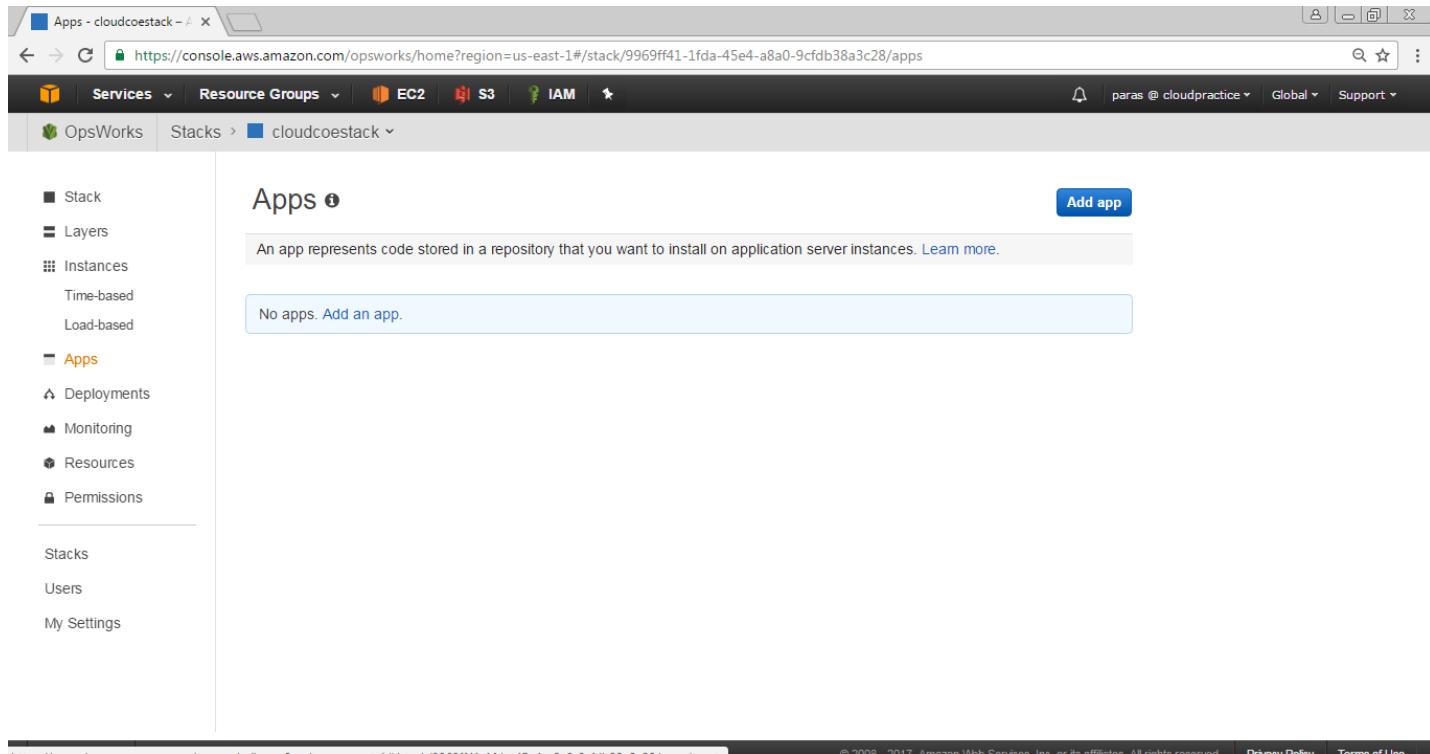
- 19 On the Layers page, choose **Security**. The Layer **MyDemolayer** page is displayed with the Security tab open. For **Security groups**, choose **AWS-OpsWorks-WebApp**, and then click on **Save** button



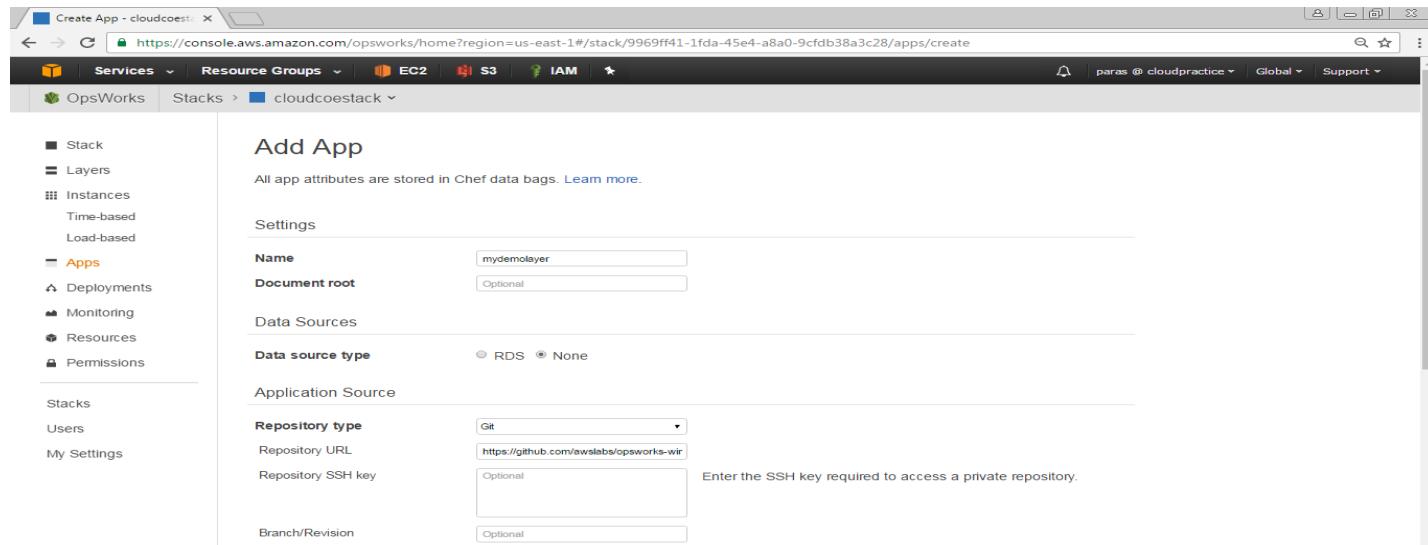
The screenshot shows the AWS OpsWorks console with the URL <https://console.aws.amazon.com/opsworks/home?region=us-east-1#/stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/layers/feef02b8-0026-4363-ae8a-36377daff399/security/edit>. The left sidebar shows the navigation menu with 'Layers' selected. The main content area is titled 'Layer Mydemolayer' and has tabs for General Settings, Recipes, Network, EBS Volumes, and Security (which is selected). Under the Security tab, there is a 'Security Groups' section with a note about security group rules controlling inbound/outbound traffic. A dropdown menu for 'Security groups' shows 'Select a security group' and lists 'AWS-OpsWorks-Default-Server' and 'AWS-OpsWorks-WebApp'. Below that is an 'EC2 Instance Profile' section with a dropdown menu for 'EC2 Instance profile' showing 'Use default stack profile (aws-opswo'. At the bottom right of the form, there are 'Cancel' and 'Save' buttons.

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

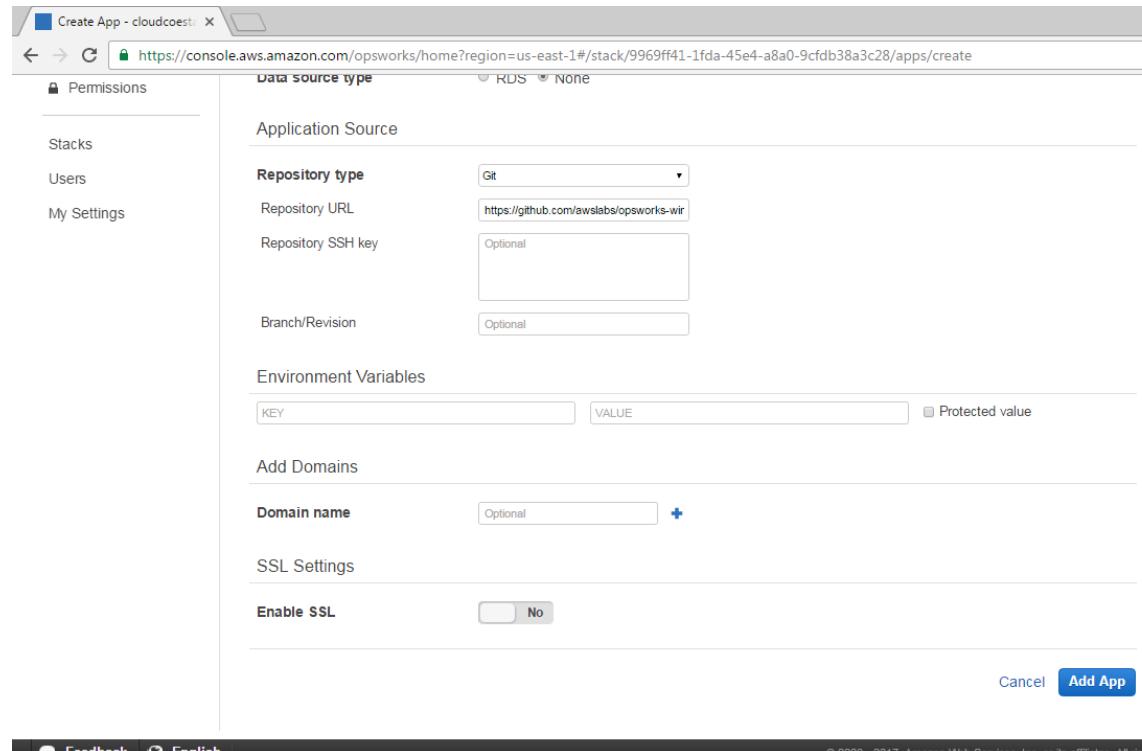


- For Settings, do the following
 - For **Name**, type **mydemolayer**
 - For **Document root** leave blank
 - For **Data Sources** select **None Data source type**
 - For **Application Source**, for **Repository type** select **Git**
 - For **Application Source**, for **Repository URL**, type
`https://github.com/awslabs/opsworks-windows-demo-nodejs.git`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



The screenshot shows the AWS OpsWorks Stacks interface for creating a new app. The left sidebar is collapsed, showing options like 'Stacks', 'Users', and 'My Settings'. The main area is titled 'Add App' with the sub-section 'Settings'. Under 'Name', the value 'mydemolayer' is entered. Under 'Data source type', 'None' is selected. In the 'Application Source' section, 'Repository type' is set to 'Git', 'Repository URL' is 'https://github.com/awslabs/opsworks-wir', and 'Branch/Revision' is 'Optional'. A note says 'Enter the SSH key required to access a private repository.'

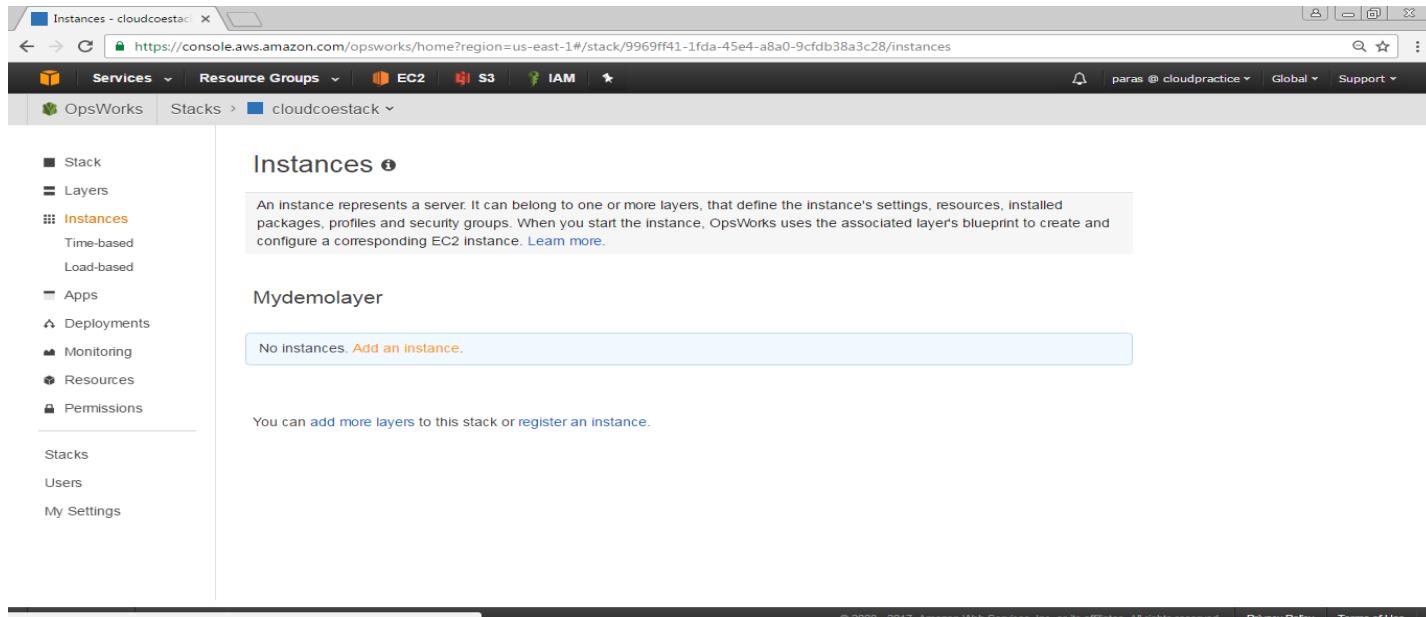
- ✓ For **Environment Variables**, **KEY** leave blank, **VALUE** leave blank, unchecked **Protected Value**
- ✓ For **Add Domains**, **Domain Name** leave blank
- ✓ For **SSL Settings**, select **No for Enable SSL**
- ✓ Click on **Add App** button. AWS OpsWorks Stacks adds the app and displays the **Apps**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



This screenshot shows the 'Add App' form with more detailed settings. The 'Permissions' sidebar is open, showing 'Stacks', 'Users', and 'My Settings'. The main form has 'Data source type' set to 'RDS'. Under 'Application Source', 'Repository type' is 'Git', 'Repository URL' is 'https://github.com/awslabs/opsworks-wir', and 'Branch/Revision' is 'Optional'. Under 'Environment Variables', there are two empty fields: 'KEY' and 'VALUE'. Under 'Add Domains', 'Domain name' is 'Optional'. Under 'SSL Settings', 'Enable SSL' is set to 'No'. At the bottom right are 'Cancel' and 'Add App' buttons.

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**



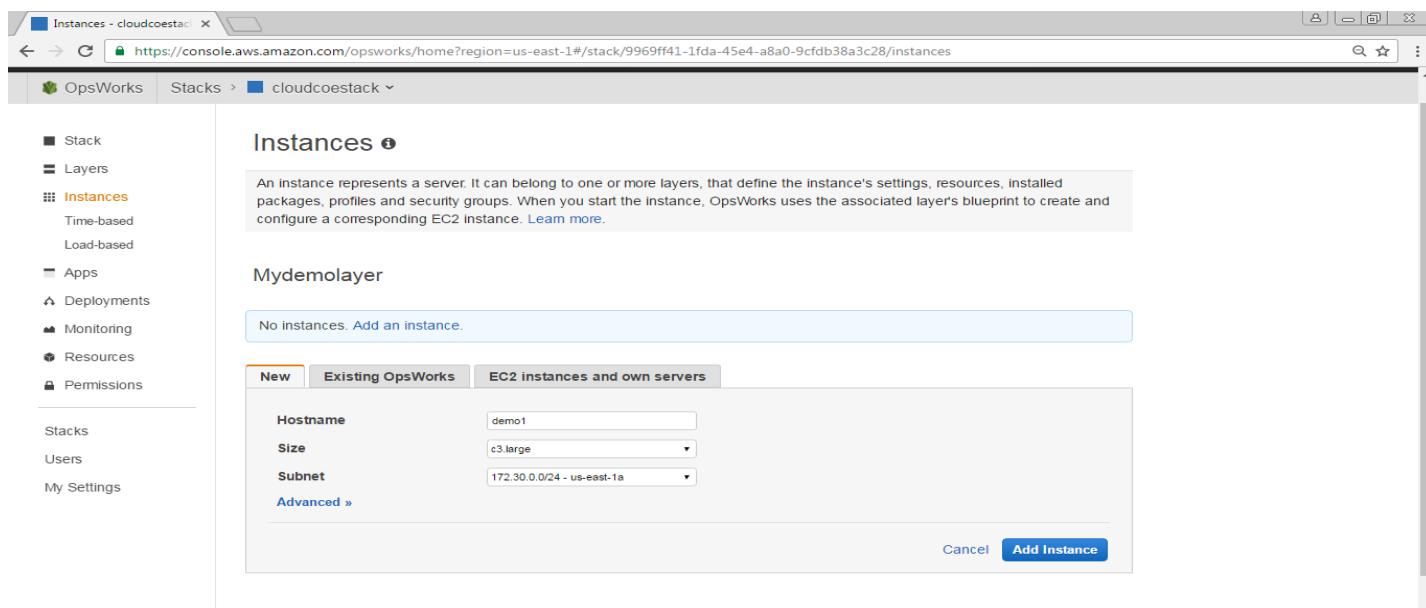
An instance represents a server. It can belong to one or more layers, that define the instance's settings, resources, installed packages, profiles and security groups. When you start the instance, OpsWorks uses the associated layer's blueprint to create and configure a corresponding EC2 instance. [Learn more.](#)

Mydemolayer

No instances. [Add an instance.](#)

You can add more layers to this stack or register 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



New Existing OpsWorks EC2 instances and own servers

Hostname: demo1

Size: c3.large

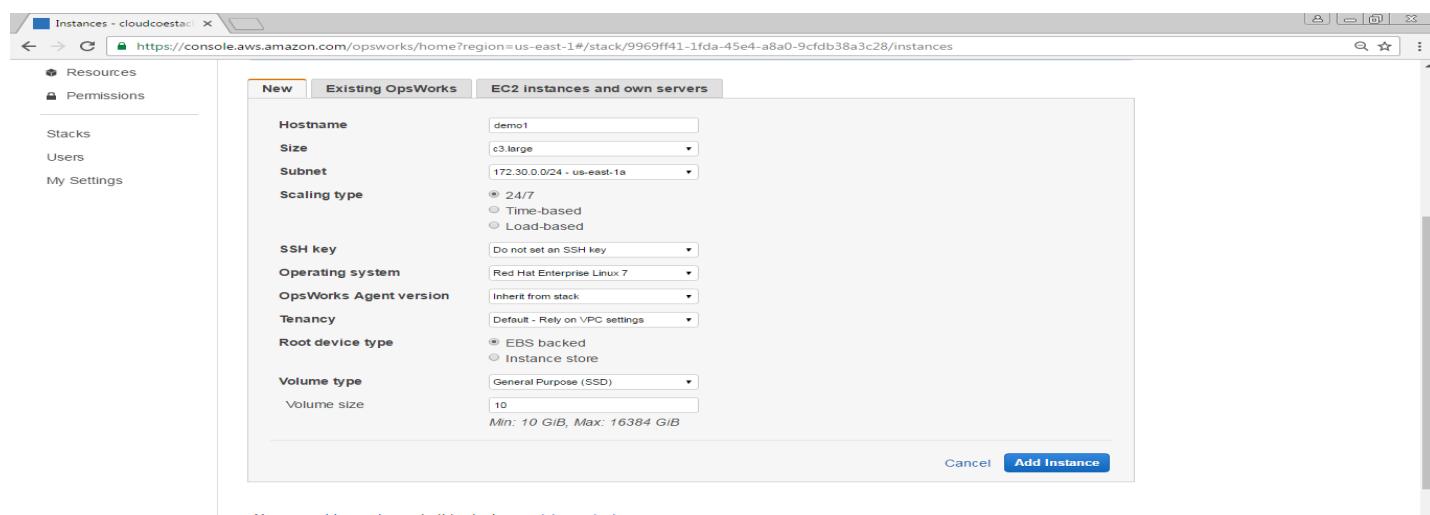
Subnet: 172.30.0.0/24 - us-east-1a

Advanced »

Cancel **Add Instance**

You can add more layers to this stack or register an instance.

- 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**

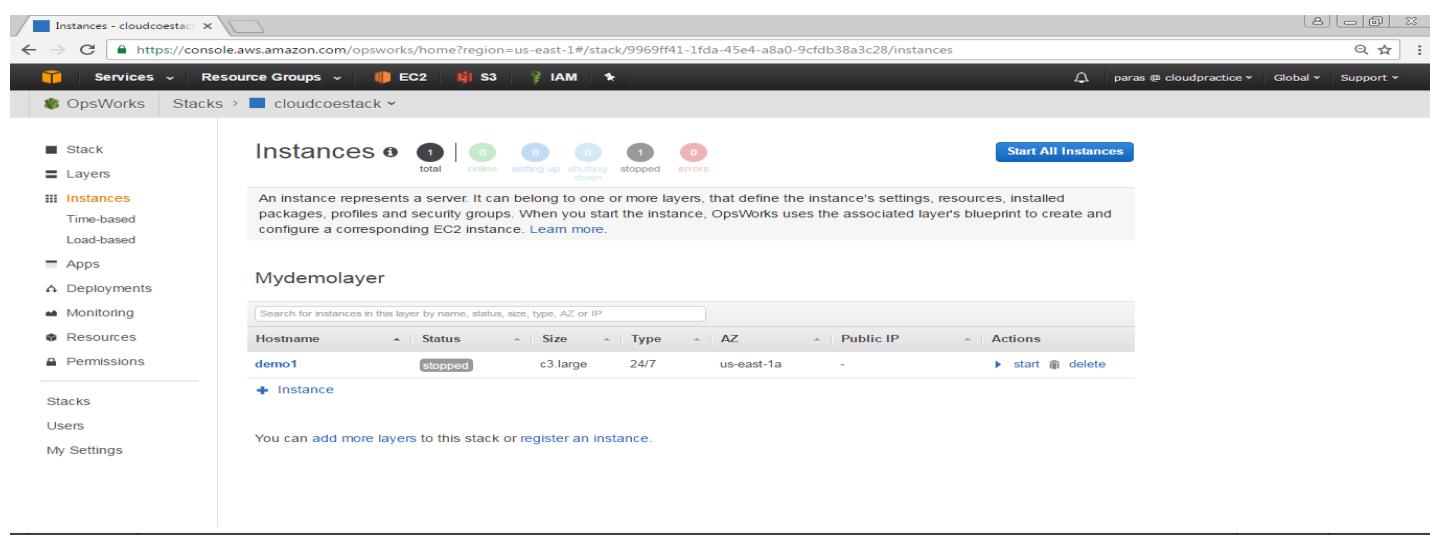


The screenshot shows the 'Instances' creation dialog in the AWS OpsWorks console. The 'New' tab is selected. The form fields are as follows:

- Hostname:** demo1
- Size:** c3.large
- Subnet:** 172.30.0.0/24 - us-east-1a
- Scaling type:** 24/7 (radio button selected)
- SSH key:** Do not set an SSH key
- Operating system:** Red Hat Enterprise Linux 7
- OpsWorks Agent version:** Inherit from stack
- Tenancy:** Default - Rely on VPC settings
- Root device type:** EBS backed (radio button selected)
- Volume type:** General Purpose (SSD)
- Volume size:** 10

At the bottom right are 'Cancel' and 'Add Instance' buttons.

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



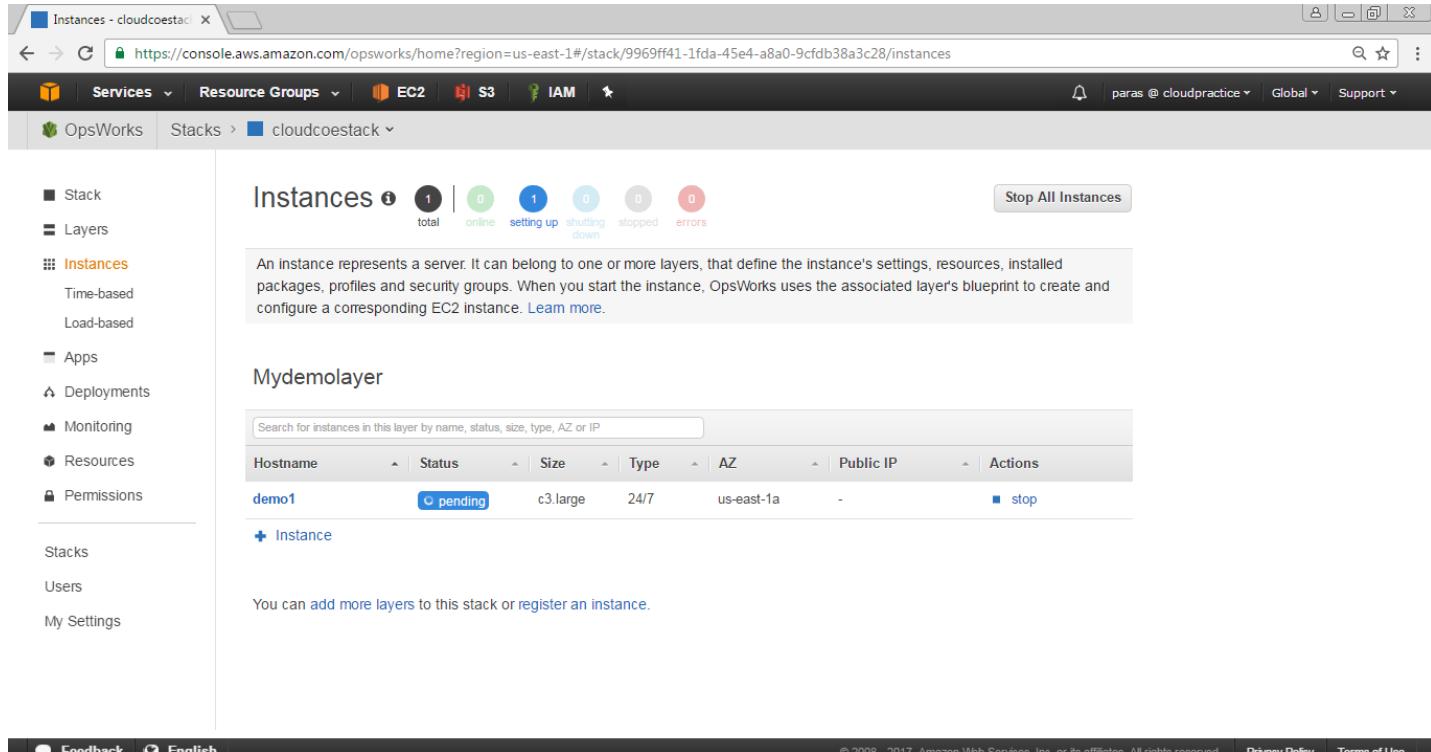
The screenshot shows the 'Instances' page for the 'Mydemolayer' stack. The left sidebar shows the navigation path: Services > Resource Groups > EC2 > S3 > IAM > OpsWorks > Stacks > cloudcostack > Mydemolayer.

The main area displays the 'Instances' section with the following details:

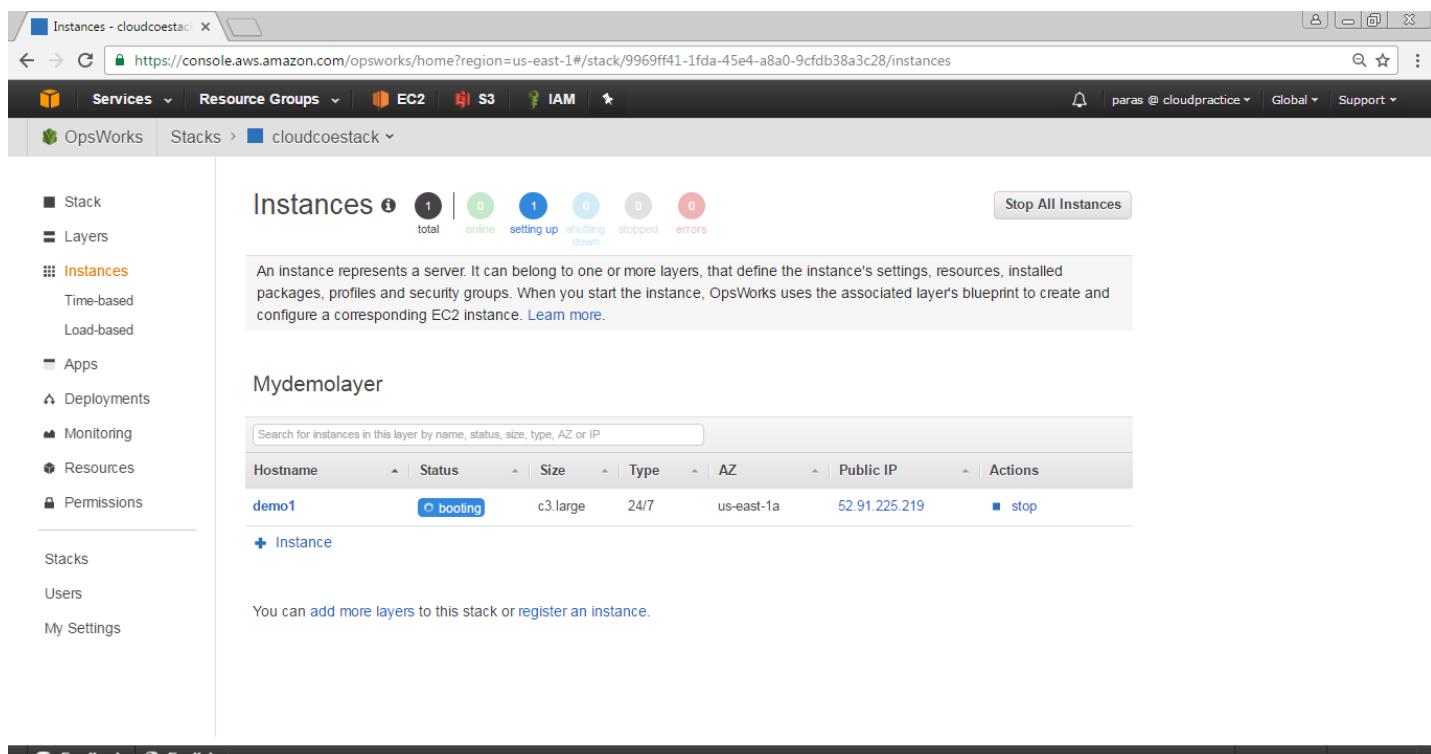
| Hostname | Status | Size | Type | AZ | Public IP | Actions |
|----------|---------|----------|------|------------|-----------|--|
| demo1 | stopped | c3.large | 24/7 | us-east-1a | - | start delete |

At the top right of the instances table is a 'Start All Instances' button. Below the table, there is a note: 'You can add more layers to this stack or register an instance.'

- 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



This screenshot shows the AWS OpsWorks Instances page for the stack 'cloudcoestack'. The left sidebar shows navigation options like Stack, Layers, Instances, Apps, Deployments, Monitoring, Resources, and Permissions. Under Instances, it lists Time-based and Load-based layers. The main area displays the 'Instances' section with a summary bar showing 1 total instance: 0 online, 1 setting up, 0 shut down, 0 stopped, and 0 errors. A 'Stop All Instances' button is available. Below this is a detailed table for the single instance 'demo1': Hostname (demo1), Status (pending), Size (c3.large), Type (24/7), AZ (us-east-1a), Public IP (-), and Actions (stop). A '+' button for Instance is also present. A note at the bottom says 'You can add more layers to this stack or register an instance.'



This screenshot shows the same AWS OpsWorks Instances page after some time has passed. The instance 'demo1' is now in a 'booting' state, indicated by the blue circle in the status column. The rest of the table and interface remain the same, showing the instance's configuration and the note about adding layers.

Instances - cloudcoestack

<https://console.aws.amazon.com/opsworks/home?region=us-east-1#/stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/instances>

Services ▾ Resource Groups ▾ EC2 S3 IAM

paras @ cloudpractice Global Support

OpsWorks Stacks cloudcoestack

Instances 1 total | 1 online | 0 setting up | 0 shutting down | 0 stopped | 0 errors Stop All Instances

An instance represents a server. It can belong to one or more layers, that define the instance's settings, resources, installed packages, profiles and security groups. When you start the instance, OpsWorks uses the associated layer's blueprint to create and configure a corresponding EC2 instance. [Learn more.](#)

Mydemolayer

| Hostname | Status | Size | Type | AZ | Public IP | Actions |
|----------|---------------|----------|------|------------|---------------|----------|
| demo1 | running_setup | c3.large | 24/7 | us-east-1a | 52.91.225.219 | stop ssh |

+ Instance You can add more layers to this stack or register an instance.

Feedback English © 2008–2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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

Instances - cloudcoestack

<https://console.aws.amazon.com/opsworks/home?region=us-east-1#/stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/instances>

Services ▾ Resource Groups ▾ EC2 S3 IAM

paras @ cloudpractice Global Support

OpsWorks Stacks cloudcoestack

Instances 1 total | 1 online | 0 setting up | 0 shutting down | 0 stopped | 0 errors Stop All Instances

An instance represents a server. It can belong to one or more layers, that define the instance's settings, resources, installed packages, profiles and security groups. When you start the instance, OpsWorks uses the associated layer's blueprint to create and configure a corresponding EC2 instance. [Learn more.](#)

Mydemolayer

| Hostname | Status | Size | Type | AZ | Public IP | Actions |
|----------|--------|----------|------|------------|---------------|----------|
| demo1 | online | c3.large | 24/7 | us-east-1a | 52.91.225.219 | stop ssh |

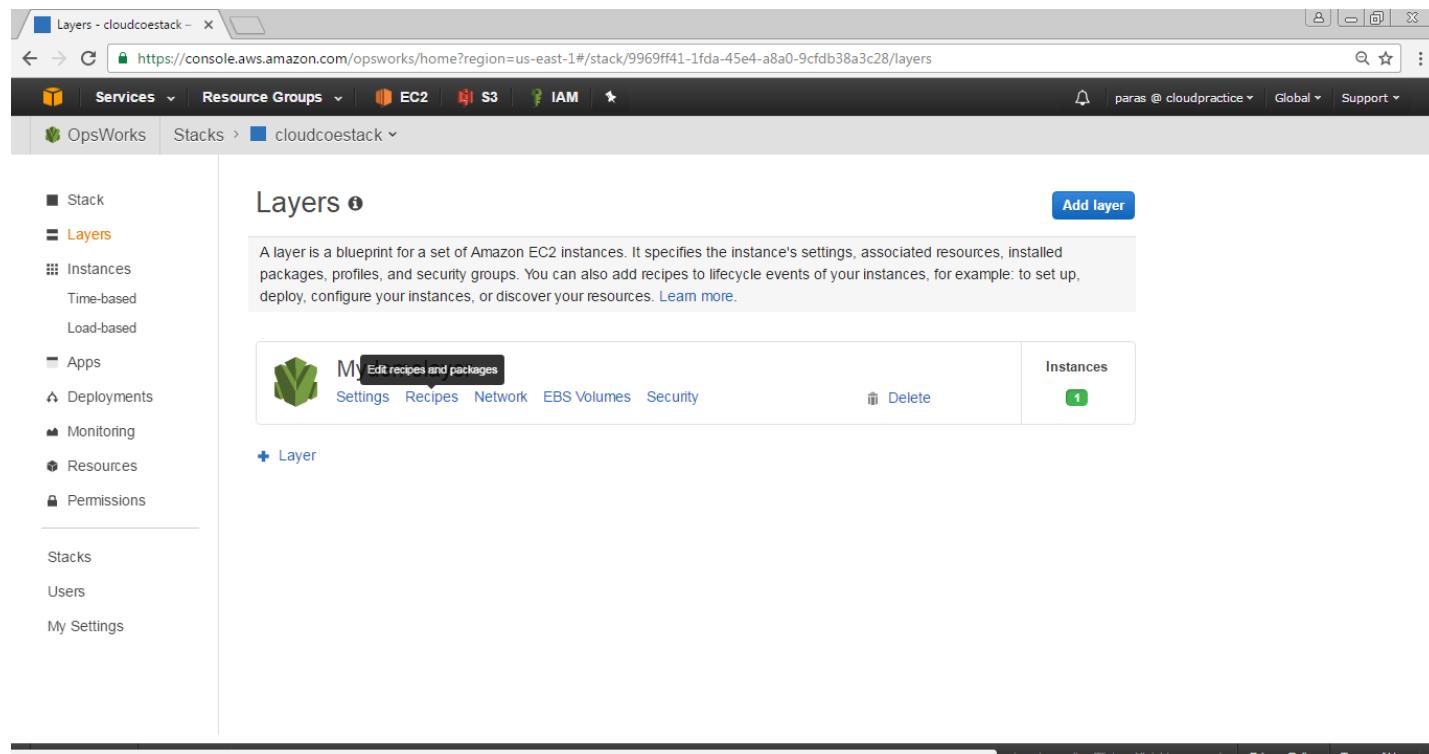
+ Instance You can add more layers to this stack or register an instance.

Feedback English © 2008–2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

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

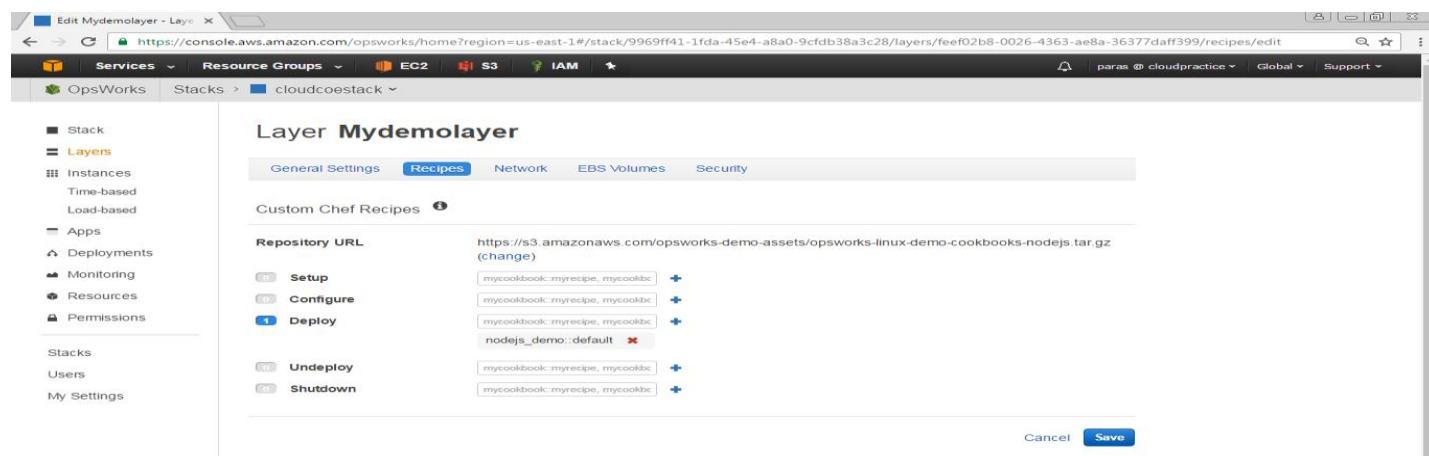
15.2.3.5 Deploy the App to the Instance

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



The screenshot shows the AWS OpsWorks console with the URL <https://console.aws.amazon.com/opsworks/home?region=us-east-1#stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/layers>. The service navigation pane on the left is visible, with 'Layers' selected. The main content area displays the 'Layers' page for the 'MyDemolayer' stack. The 'Recipes' tab is active, showing a list of recipes under the 'Deploy' section. One recipe, 'nodejs_demo::default', is highlighted with a red asterisk.

- 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

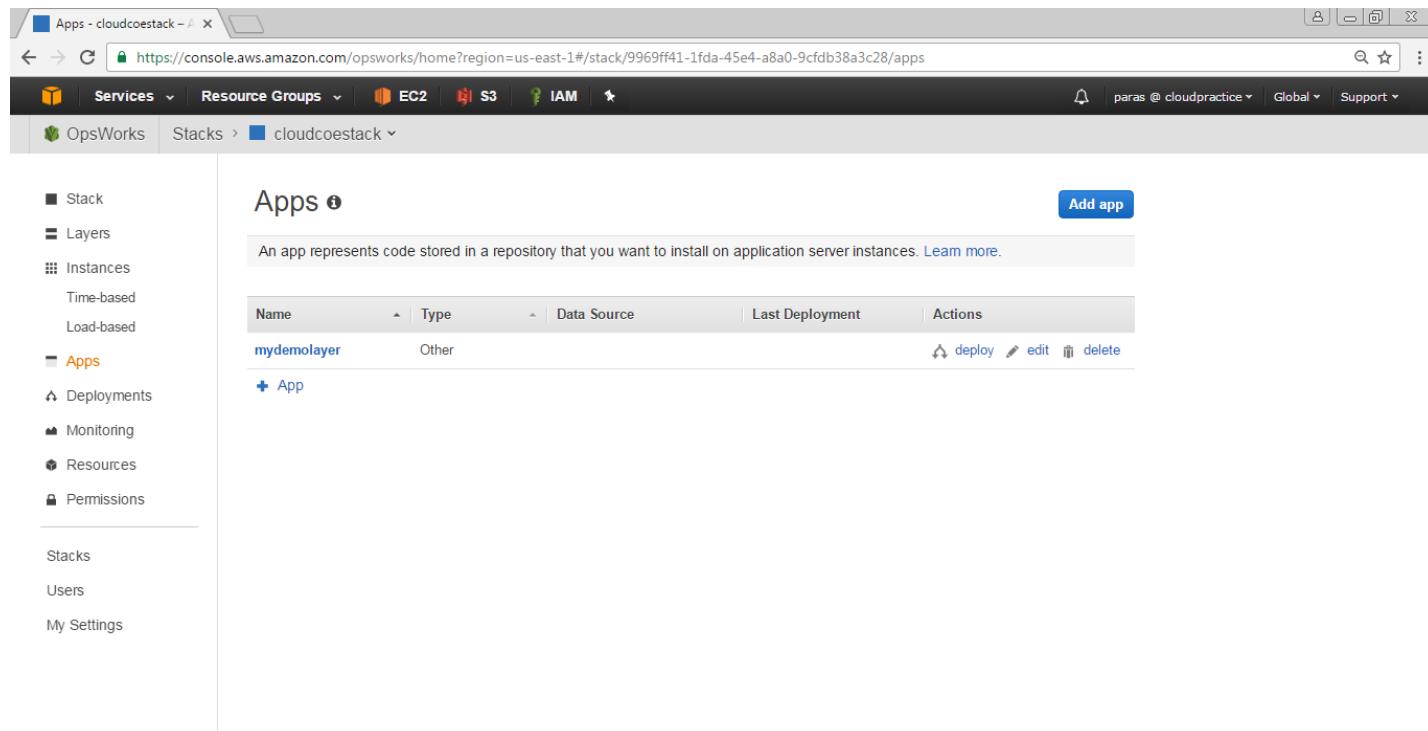


The screenshot shows the AWS OpsWorks console with the URL <https://console.aws.amazon.com/opsworks/home?region=us-east-1#stack/9969ff41-1fda-45e4-a8a0-9cfdb38a3c28/layers/feef02b8-0026-4363-ae8a-36377daff399/recipes/edit>. The service navigation pane on the left is visible, with 'Layers' selected. The main content area displays the 'Layer Mydemolayer' page. The 'Recipes' tab is active, showing the 'Deploy' section with the 'nodejs_demo::default' recipe highlighted. The 'Save' button is visible at the bottom right.

3. To deploy the app to the instance,

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

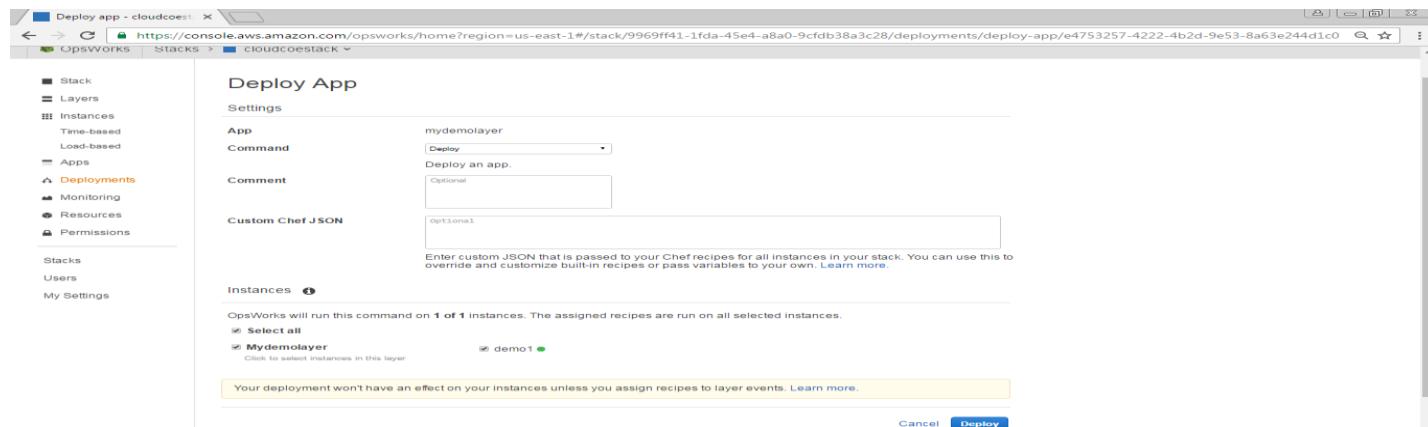
For **MyDemolayer**, for **Actions**, choose **deploy**



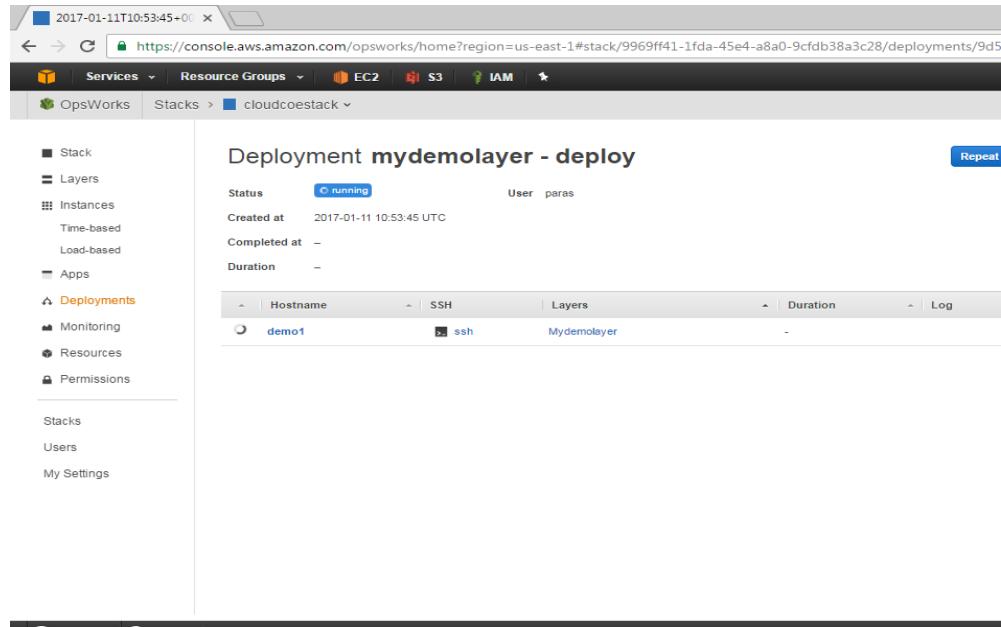
| Name | Type | Data Source | Last Deployment | Actions |
|-------------|-------|-------------|-----------------|--|
| mydemolayer | Other | | | deploy edit delete |

- 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

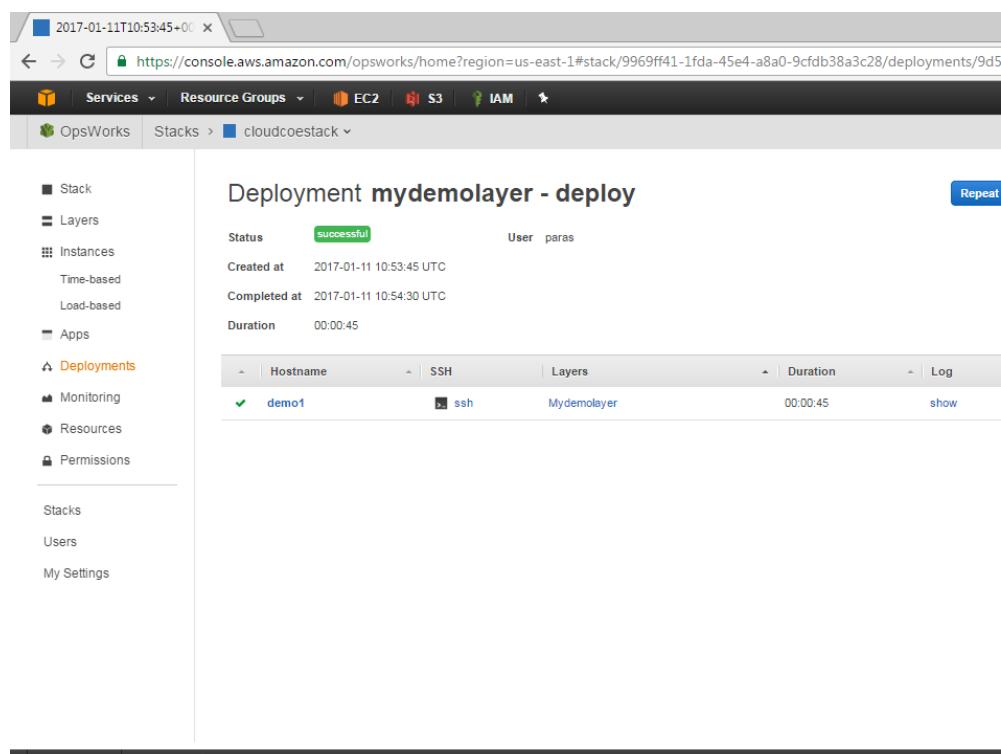


5. The Deployment MyDemolayer – deploy page is displayed. **Status** changes from **running** to **successful**. A spinning circle displays next to **demo1**, which then changes to a green check mark. Note that this process can take several minutes. Do not proceed until you see both a **Status of successful**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



Deployment mydemolayer - deploy

| Hostname | SSH | Layers | Duration | Log |
|----------|-----|-------------|----------|-----|
| demo1 | ssh | Mydemolayer | - | - |



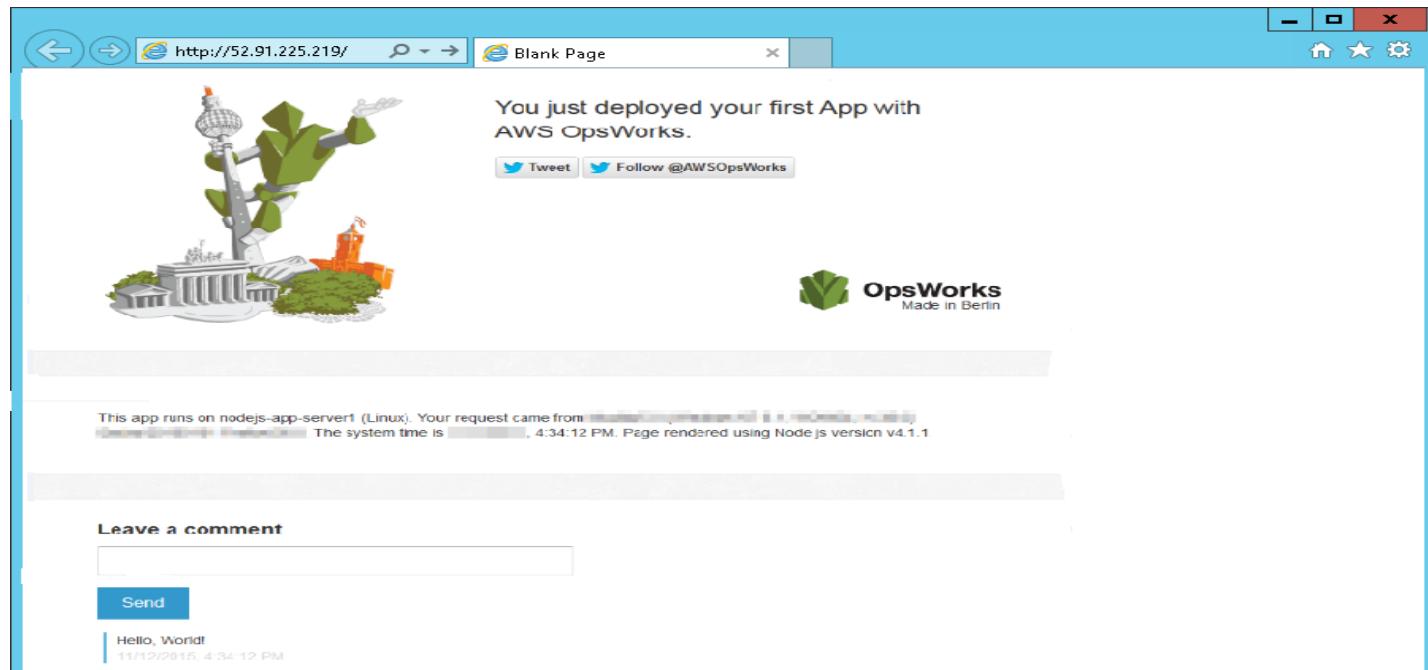
Deployment mydemolayer - deploy

| Hostname | SSH | Layers | Duration | Log |
|----------|-----|-------------|----------|------|
| ✓ demo1 | ssh | Mydemolayer | 00:00:45 | show |

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

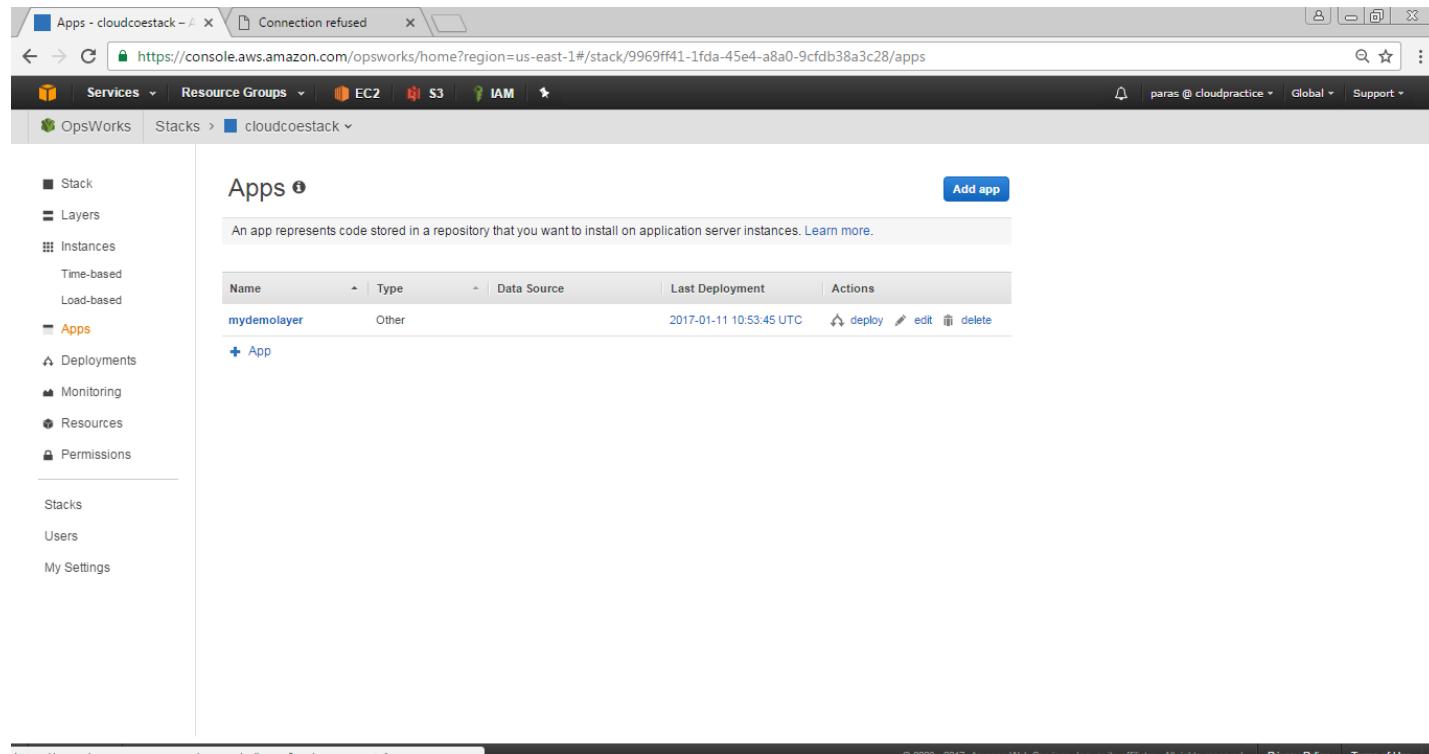
15.2.3.6 Test the Deployed App on the Instance

- 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



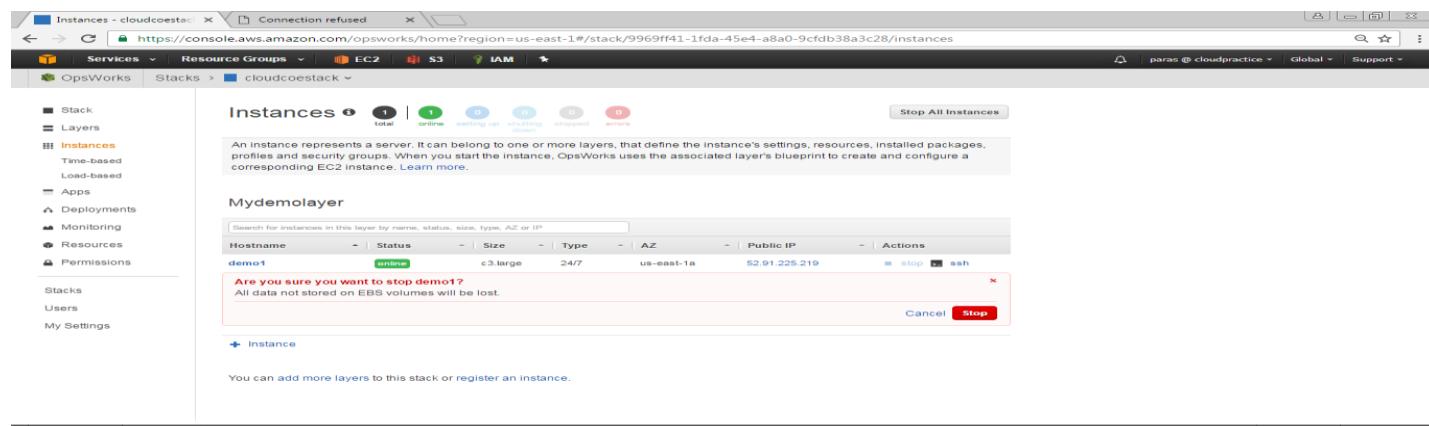
NOTE: You have now successfully tested the deployed app on the instance

- 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



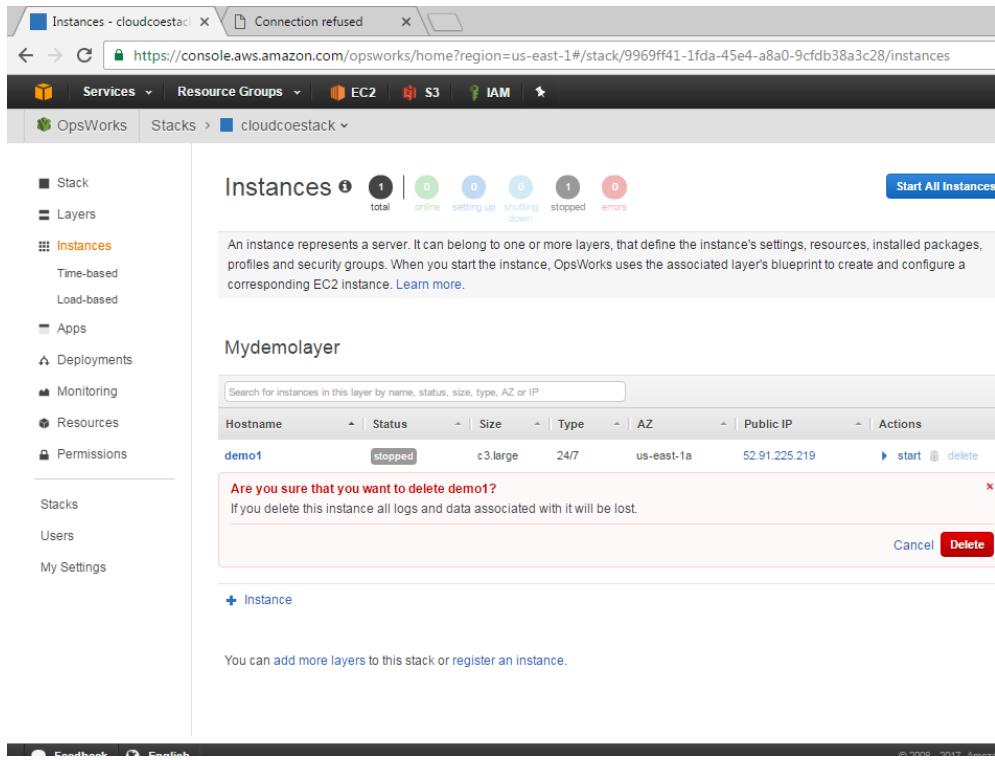
The screenshot shows the AWS OpsWorks Stacks interface. The left sidebar has a tree view with 'Stacks' expanded, showing 'cloudcoestack'. Under 'cloudcoestack', 'Layers', 'Instances', 'Time-based', 'Load-based', and 'Apps' are listed. 'Apps' is selected and highlighted in orange. The main content area is titled 'Apps' and contains a table with one row for 'mydemolayer'. The table columns are 'Name', 'Type', 'Data Source', 'Last Deployment', and 'Actions'. The 'Actions' column for 'mydemolayer' includes links for 'deploy', 'edit', and 'delete'. A blue 'Add app' button is located at the top right of the table area.

- 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**.



The screenshot shows the AWS OpsWorks Stacks Instances page. The left sidebar shows 'Instances' selected under 'cloudcoestack'. The main content area is titled 'Instances' and shows a table with one row for 'demo1'. The table columns are 'Hostname', 'Status', 'Size', 'Type', 'AZ', 'Public IP', and 'Actions'. The 'Actions' column for 'demo1' includes links for 'stop' and 'ssh'. A modal dialog box is open over the table, asking 'Are you sure you want to stop demo1? All data not stored on EBS volumes will be lost.' with 'Cancel' and 'Stop' buttons. Below the table, there's a note: 'You can add more layers to this stack or register an instance.'

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

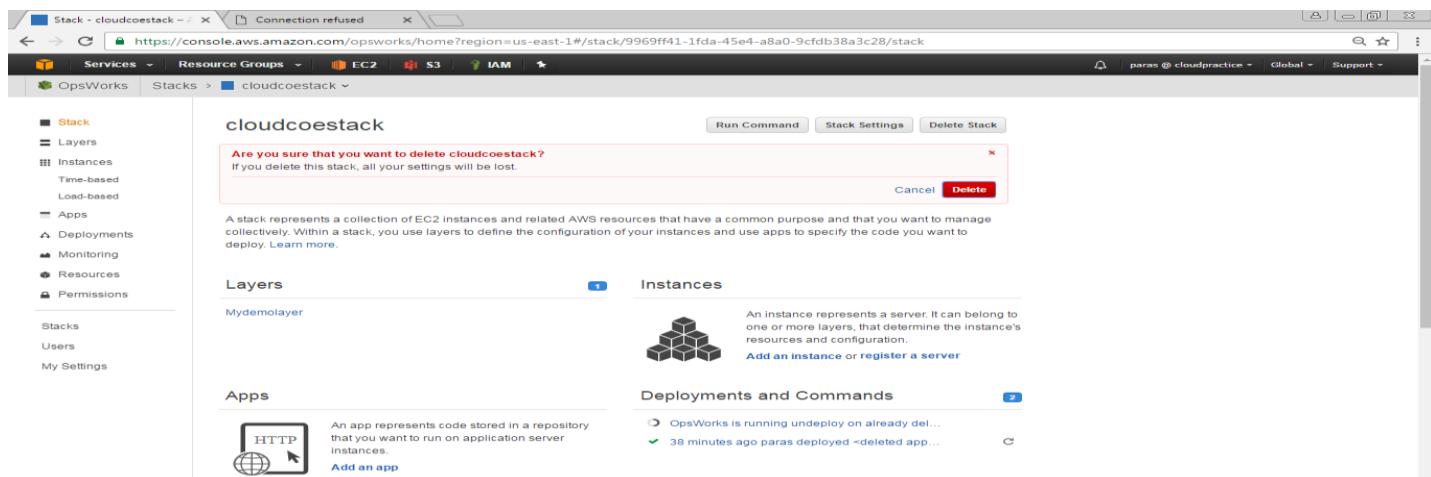


The screenshot shows the AWS OpsWorks Instances page for the stack 'cloudcoestack'. The left sidebar navigation includes 'Stack', 'Layers', 'Instances' (selected), 'Time-based', 'Load-based', 'Apps', 'Deployments', 'Monitoring', 'Resources', 'Permissions', 'Stacks', 'Users', and 'My Settings'. The main content area displays the 'Instances' section with a summary: 1 total, 0 online, 0 setting up, 0 shutting down, 1 stopped, and 0 errors. A 'Start All Instances' button is at the top right. Below is a detailed table:

| Hostname | Status | Size | Type | AZ | Public IP | Actions |
|----------|---------|----------|------|------------|---------------|--|
| demo1 | stopped | c3.large | 24/7 | us-east-1a | 52.91.225.219 | start delete |

A modal dialog box is open over the table, asking 'Are you sure that you want to delete demo1? If you delete this instance all logs and data associated with it will be lost.' It has 'Cancel' and 'Delete' buttons.

- 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**.



The screenshot shows the AWS OpsWorks Stack page for the stack 'cloudcoestack'. The left sidebar navigation is identical to the previous screenshot. The main content area includes sections for 'Run Command', 'Stack Settings', and 'Delete Stack' buttons. Below is a detailed view of the stack components:

- Layers:** Shows 'Mydemolayer'.
- Instances:** Shows a single instance represented by a stack of cubes icon. A tooltip says: 'An instance represents a server. It can belong to one or more layers, that determine the instance's resources and configuration.' It also says 'Add an instance or register a server'.
- Apps:** Shows an 'HTTP' icon with a cursor over it, and a tooltip: 'An app represents code stored in a repository that you want to run on application server instances.' It also says 'Add an app'.
- Deployments and Commands:** Shows deployment status: 'OpsWorks is running undeploy on already del...' and '38 minutes ago paras deployed <deleted app...>'.

16. AWS Lambda

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

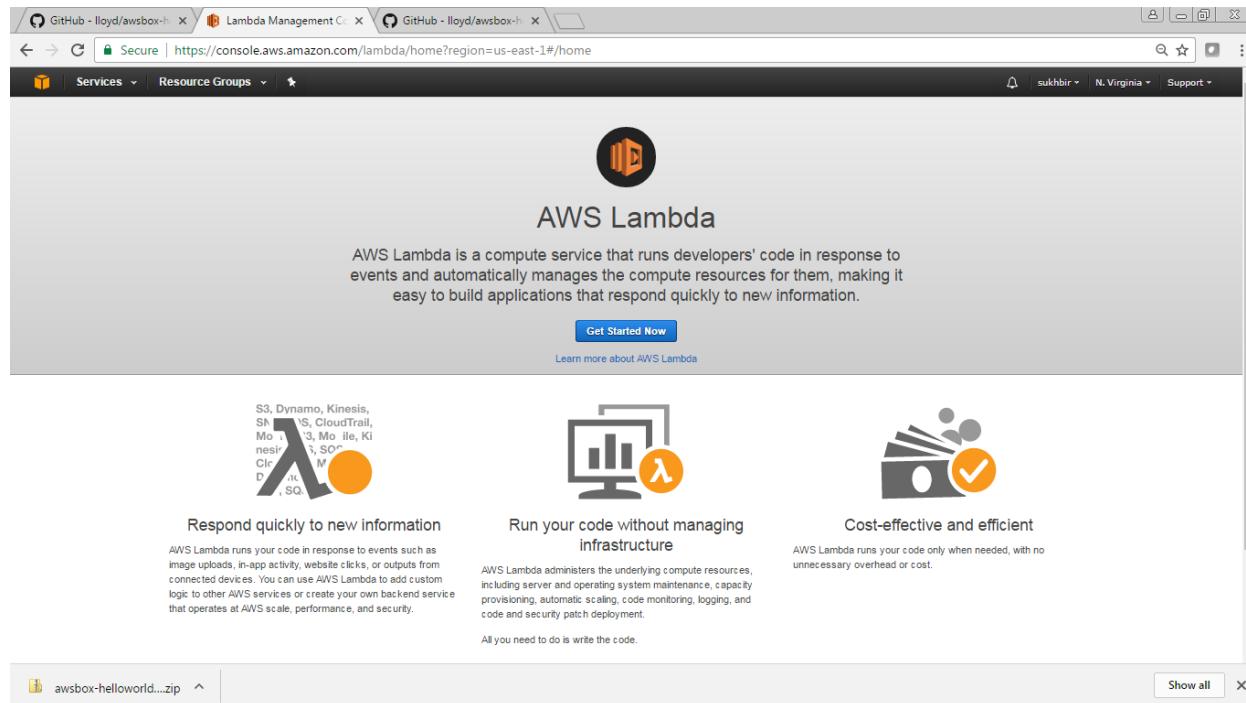
16.1 Objective

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

16.2 Procedure

16.2.1 Creating Lambda Function

1. Sign in to the [AWS Management Console](https://console.aws.amazon.com/lambda/home?region=us-east-1#/) and open the AWS Lambda console
(Assuming user is having IAM access rights for AWS Lambda.



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

Lambda Management C... GitHub - lloyd/awsbox-h...

Secure | https://console.aws.amazon.com/lambda/home?region=us-east-1#/create/select-blueprint?firstrun=true

Lambda Services Resource Groups

Lambda > New function

Select blueprint

Configure triggers

Configure function

Review

Select blueprint

Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under [CC0](#).

Welcome to AWS Lambda! You can get started on creating your first Lambda function by choosing one of the blueprints below.

Select runtime Filter Viewing 1-9 of 70

| | | |
|--|--|---|
| Blank Function Configure your function from scratch. Define the trigger and deploy your code by stepping through our wizard. custom | kinesis-firehose-syslog-to-json An Amazon Kinesis Firehose stream processor that converts input records from RFC3164 Syslog format to JSON. nodejs kinesis-firehose | alexa-skill-kit-sdk-factskill Demonstrate a basic fact skill built with the ASK NodeJS SDK. nodejs alexa |
| kinesis-firehose-apache-log-to-j... An Amazon Kinesis Firehose stream processor that converts input records from Apache Common Log format to python2.7 kinesis-firehose | cloudfront-modify-response-he... Blueprint for modifying CloudFront response header implemented in NodeJS. nodejs cloudfront response header | s3-get-object-python An Amazon S3 trigger that retrieves metadata for the object that has been updated. python2.7 s3 |
| config-rule-change-triggered An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types. | dynamodb-process-stream An Amazon DynamoDB trigger that logs the updates made to a table. | microservice-http-endpoint A simple back end (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway. |

awsbox-helloworld...zip ^ Show all X

Lambda Management C... GitHub - lloyd/awsbox-h...

Secure | https://console.aws.amazon.com/lambda/home?region=us-east-1#/create/configure-triggers?firstrun=true

Lambda Services Resource Groups

Lambda > New function

Select blueprint

Configure triggers

You can choose to add a trigger that will invoke your function.

Lambda

Cancel Previous Next

Feedback English

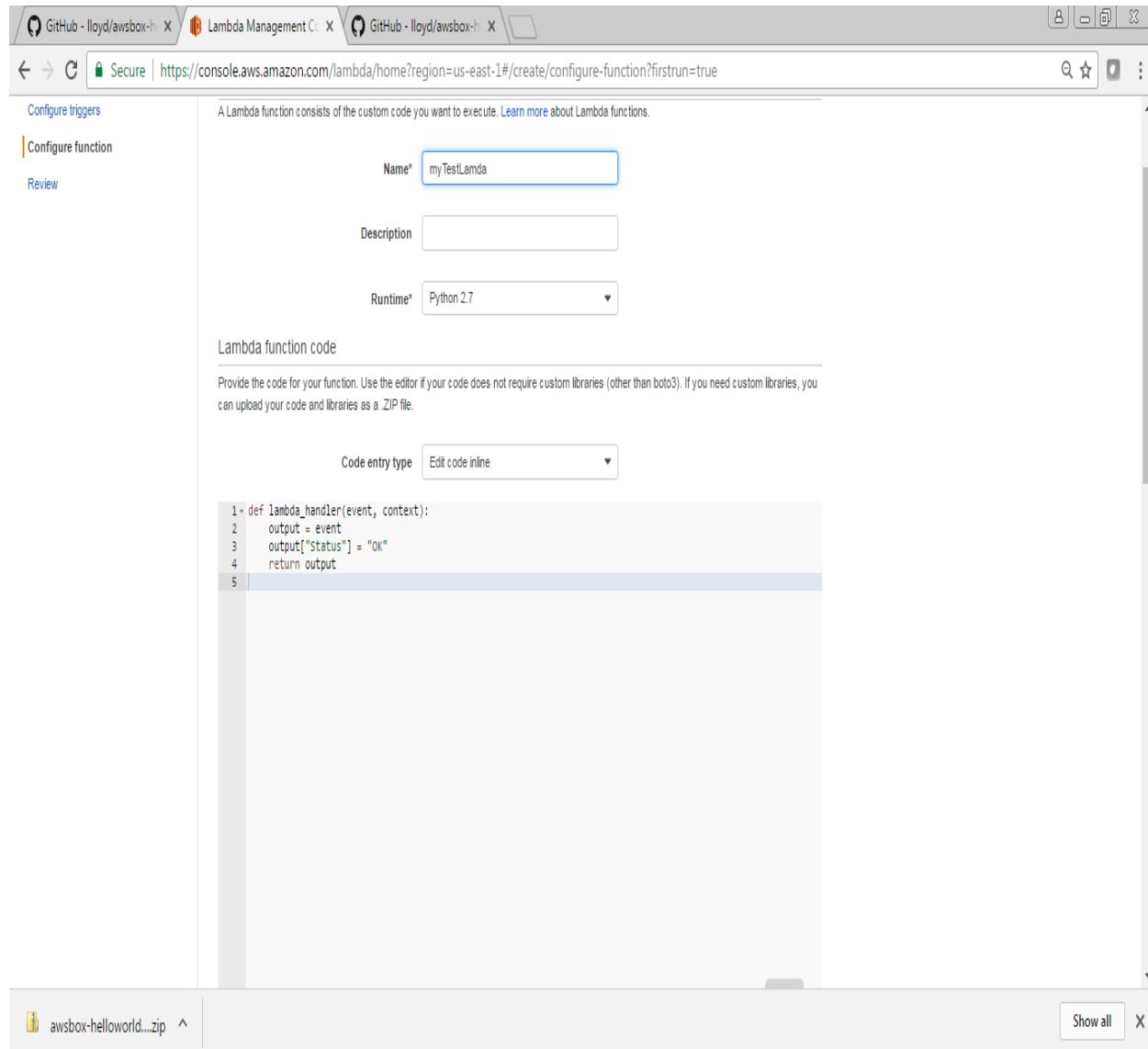
© 2008 - 2017, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

awsbox-helloworld...zip ^ Show all X

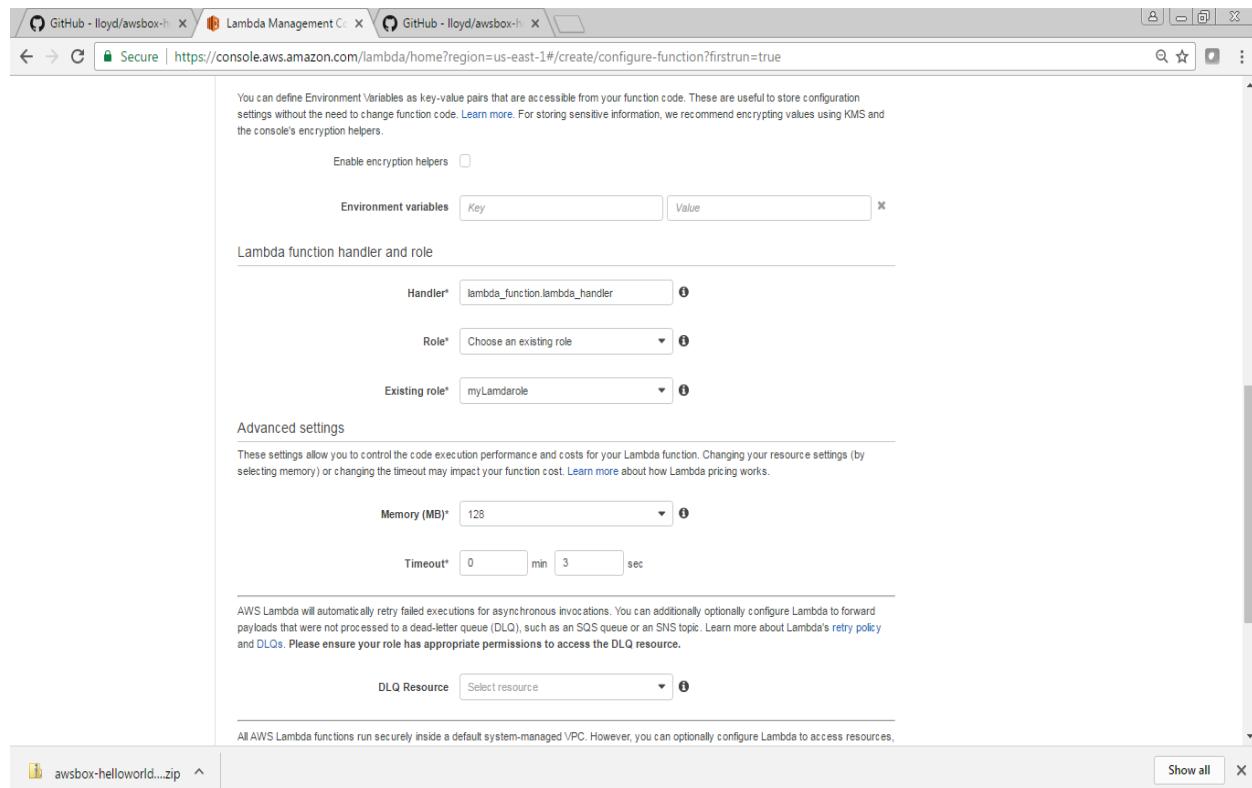
5. Paste the following code for Lambda Function Code:

```
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.



The screenshot shows the AWS Lambda Management Console interface. The URL in the address bar is <https://console.aws.amazon.com/lambda/home?region=us-east-1#/create/configure-function?firstrun=true>.

Environment variables

You can define Environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#). For storing sensitive information, we recommend encrypting values using KMS and the console's encryption helpers.

Enable encryption helpers

| Key | Value |
|-----|-------|
| | |

Lambda function handler and role

Handler*

Role*

Existing role*

Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. Changing your resource settings (by selecting memory) or changing the timeout may impact your function cost. [Learn more](#) about how Lambda pricing works.

Memory (MB)*

Timeout*

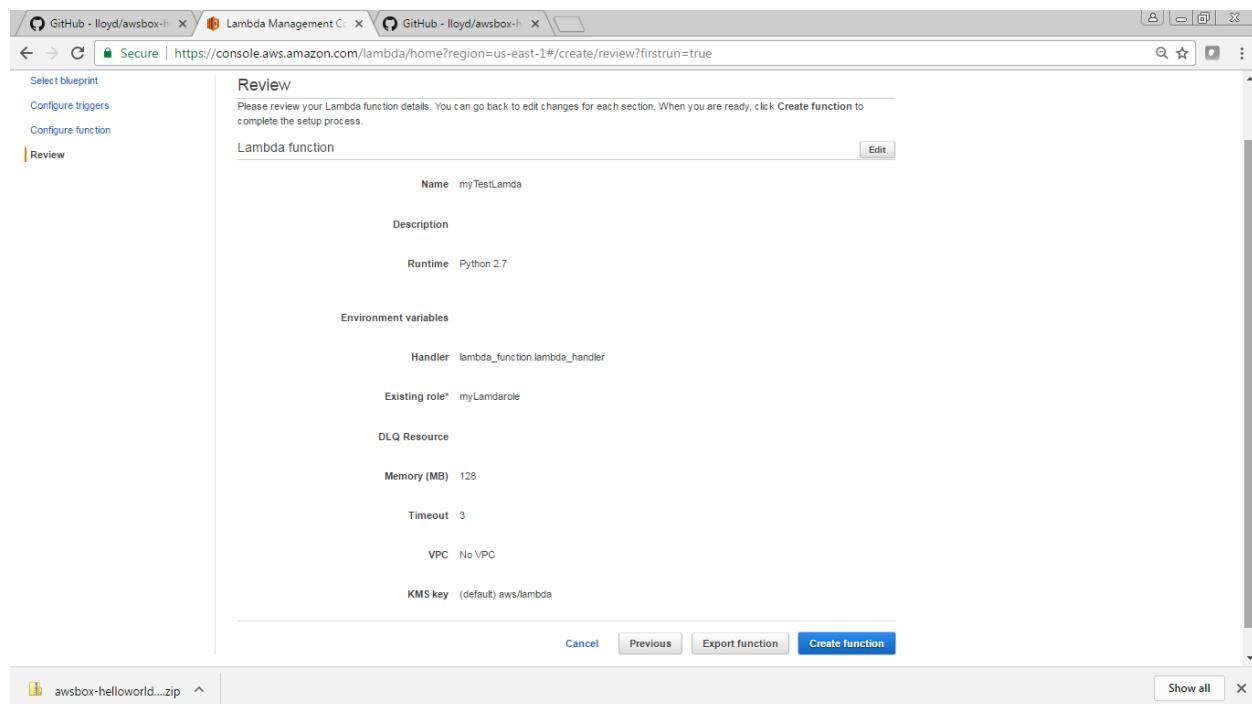
DLQ Resource

Select resource

All AWS Lambda functions run securely inside a default system-managed VPC. However, you can optionally configure Lambda to access resources,



Accept all other defaults and choose Next, Create Function.



The screenshot shows the AWS Lambda Management Console interface. The URL in the address bar is <https://console.aws.amazon.com/lambda/home?region=us-east-1#/create/review?firstrun=true>.

Review

Please review your Lambda function details. You can go back to edit changes for each section. When you are ready, click **Create function** to complete the setup process.

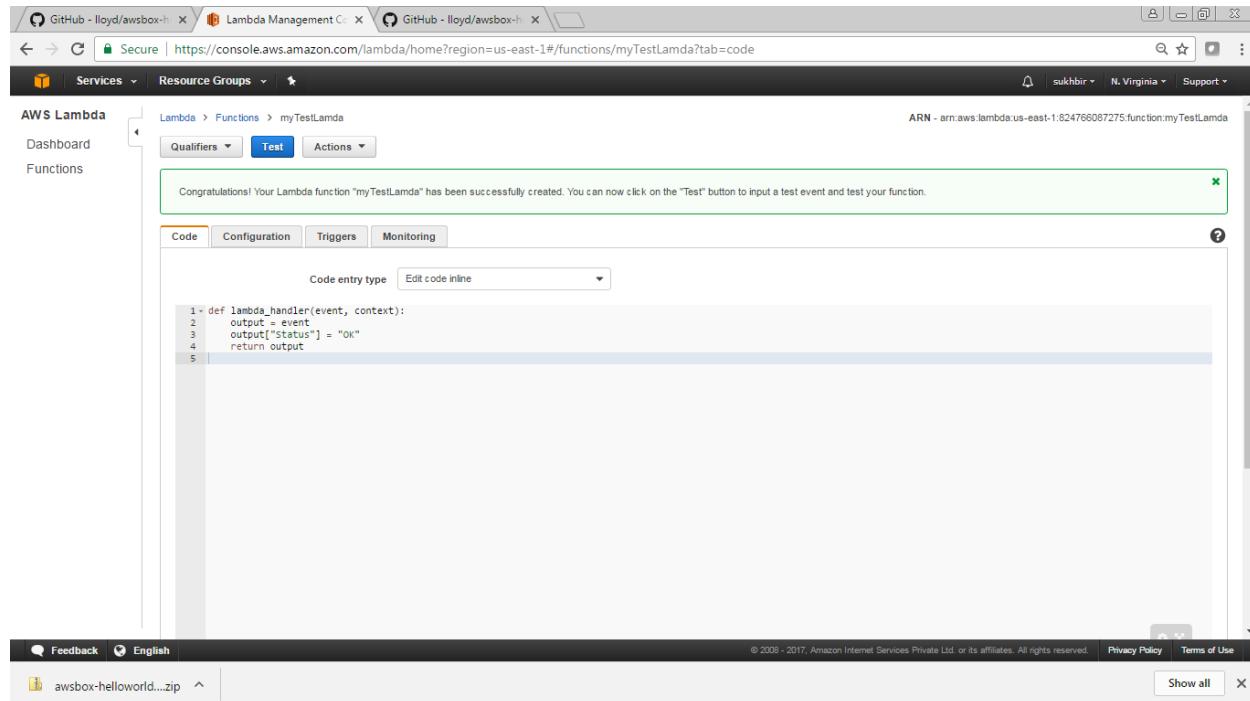
Lambda function

| | | |
|------------------------------|--------------------------------|-------------------------------------|
| Name | myTestLambda | <input type="button" value="Edit"/> |
| Description | | |
| Runtime | Python 2.7 | |
| Environment variables | | |
| Handler | lambda_function.lambda_handler | |
| Existing role* | myLamdarole | |
| DLQ Resource | | |
| Memory (MB) | 128 | |
| Timeout | 3 | |
| VPC | No VPC | |
| KMS key | (default) aws/lambda | |



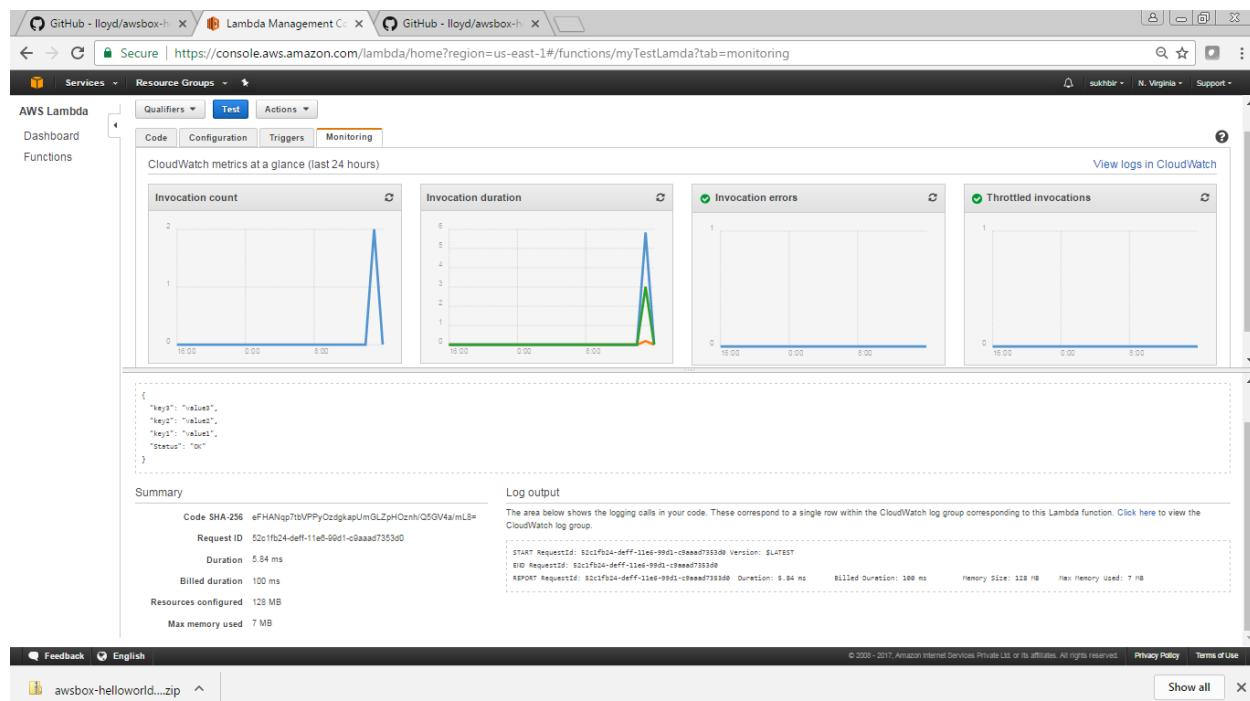
16.2.2 Testing Lambda Function

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.



Congratulations! Your Lambda function "myTestLambda" has been successfully created. You can now click on the "Test" button to input a test event and test your function.

ARN - arn:aws:lambda:us-east-1:824766087275:function:myTestLambda



CloudWatch metrics at a glance (last 24 hours)

Invocation count

Invocation duration

Invocation errors

Throttled invocations

Invocation count chart data:

| Time | Invocation Count |
|-------|------------------|
| 18.00 | 0 |
| 0.00 | 0 |
| 8.00 | 0 |
| 10.00 | 2 |
| 11.00 | 1 |
| 12.00 | 2 |

Invocation duration chart data:

| Time | Duration |
|-------|----------|
| 18.00 | 0 |
| 0.00 | 0 |
| 8.00 | 0 |
| 10.00 | 5 |
| 11.00 | 1 |
| 12.00 | 6 |

Invocation errors chart data:

| Time | Errors |
|-------|--------|
| 18.00 | 0 |
| 0.00 | 0 |
| 8.00 | 0 |
| 10.00 | 1 |
| 11.00 | 0 |
| 12.00 | 0 |

Throttled invocations chart data:

| Time | Invocations |
|-------|-------------|
| 18.00 | 0 |
| 0.00 | 0 |
| 8.00 | 0 |
| 10.00 | 1 |
| 11.00 | 0 |
| 12.00 | 0 |

Log output

```
Code SHA-256 eFHAnQp7tbVPPyOzdgkapUmGLZpHOznhQ5GV4a/mLs#
Request ID 52cf024-def1-11e5-99d1-c9aaad7353d0
Duration 0.84 ms
Billed duration 100 ms
Resources configured 128 MB
Max memory used 7 MB
```

```
START RequestId: 52cf024-def1-11e5-99d1-c9aaad7353d0 Version: $LATEST
EID RequestId: 52cf024-def1-11e5-99d1-c9aaad7353d0
REPORT RequestId: 52cf024-def1-11e5-99d1-c9aaad7353d0 Duration: 0.84 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 7 MB
```

17. APPLICATION & DEVELOPER TOOL SUPPORT

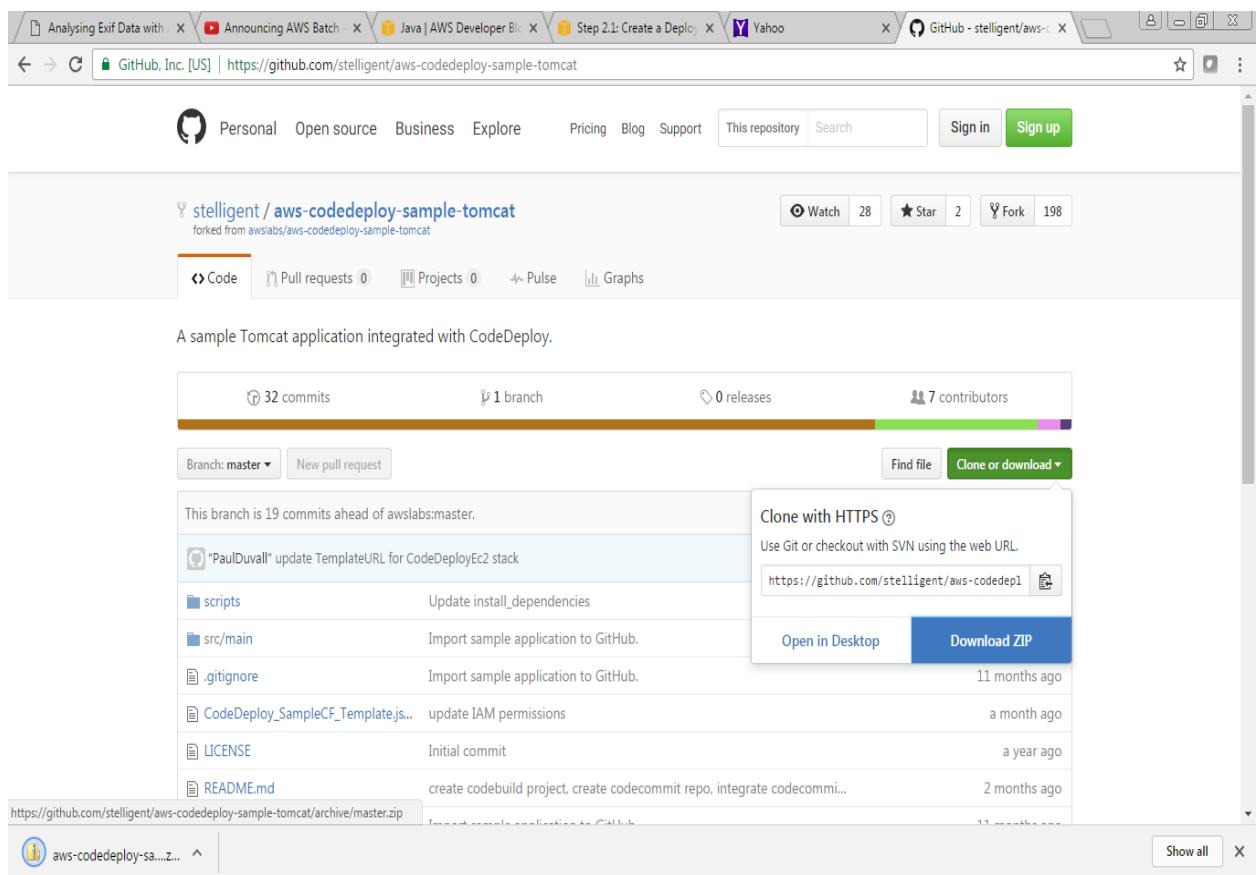
17.1 Objective

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

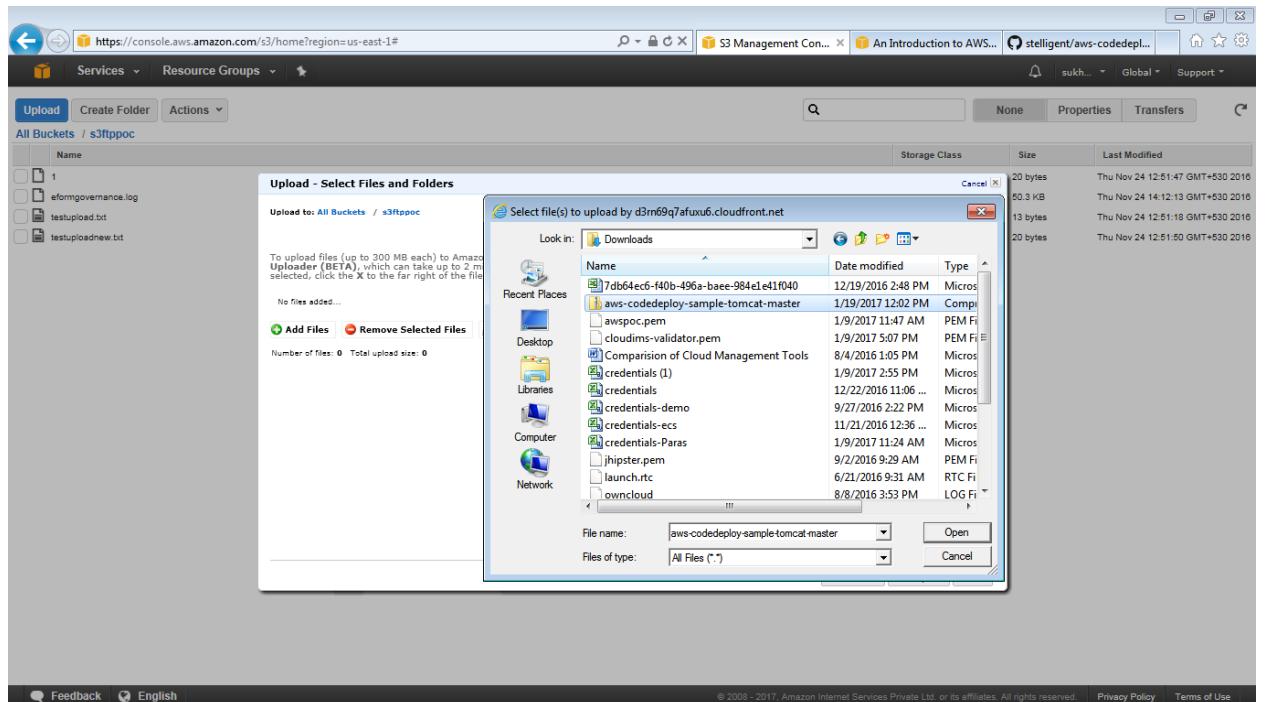
17.2 Procedure

17.2.1 Configuring GitHub and AWS S3 bucket

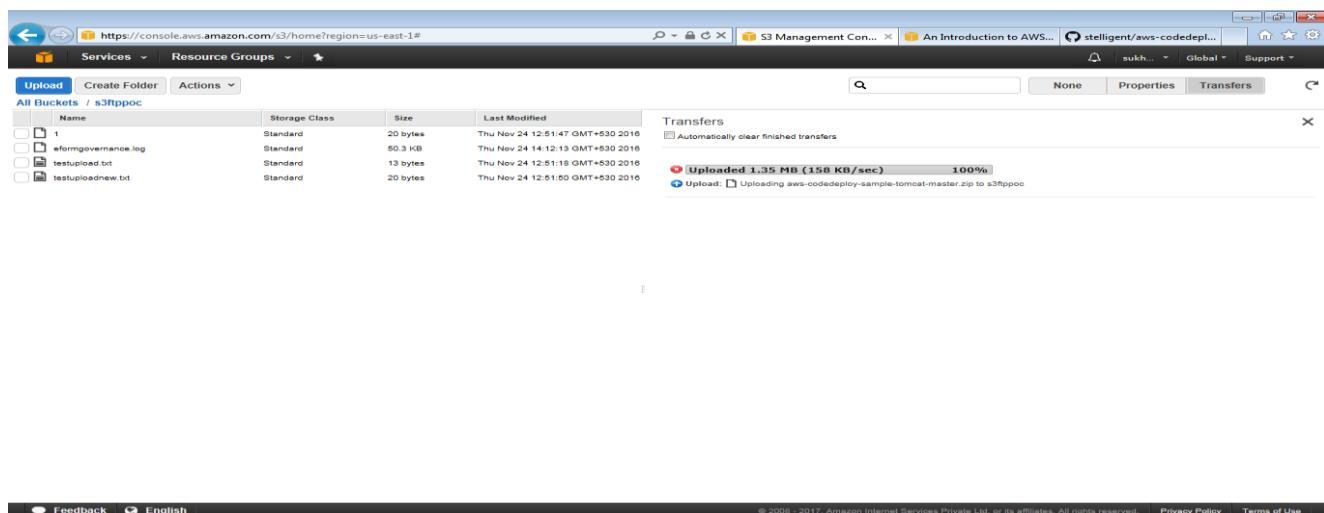
- Login to the GitHub and download the GitHub package: <https://github.com/stelligent/aws-codedeploy-sample-tomcat>



The screenshot shows a GitHub repository page for 'stelligent / aws-codedeploy-sample-tomcat'. The repository has 32 commits, 1 branch, 0 releases, and 7 contributors. It contains files like scripts, src/main, .gitignore, CodeDeploy_SampleCF_Template.js, LICENSE, and README.md. A 'Clone with HTTPS' button is visible, with a 'Download ZIP' link highlighted.



- 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**



Screenshot of the AWS S3 Management Console showing the upload of a file named "aws-codedeploy-sample-tomcat-master.zip". A context menu is open over the file, with the "Make Public" option selected. A confirmation dialog box asks if the user wants to make the object public.

Object: aws-codedeploy-sample-tomcat-master.zip

| 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 |
| etfmgovernance.log | Standard | 50.3 kB | Thu Nov 24 14:12:13 GMT+530 2016 |
| testupload.txt | Standard | 13 bytes | Thu Nov 24 12:51:19 GMT+530 2016 |
| testuploadnew.txt | Standard | 20 bytes | Thu Nov 24 12:51:50 GMT+530 2016 |

Bucket: s3ftppoc

Object: aws-codedeploy-sample-tomcat-master.zip

Permissions

Message from webpage

Are you sure you want to make aws-codedeploy-sample-tomcat-master.zip public?

OK Cancel

Save Cancel

Screenshot of the AWS S3 Management Console showing the properties of a bucket named "s3ftppoc". The "Enable Versioning" button is highlighted.

Bucket: s3ftppoc

Buckets: s3ftppoc
Region: Northern California
Creation Date: Thu Nov 24 12:42:30 GMT+530 2016
Owner: amazon@khavari

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

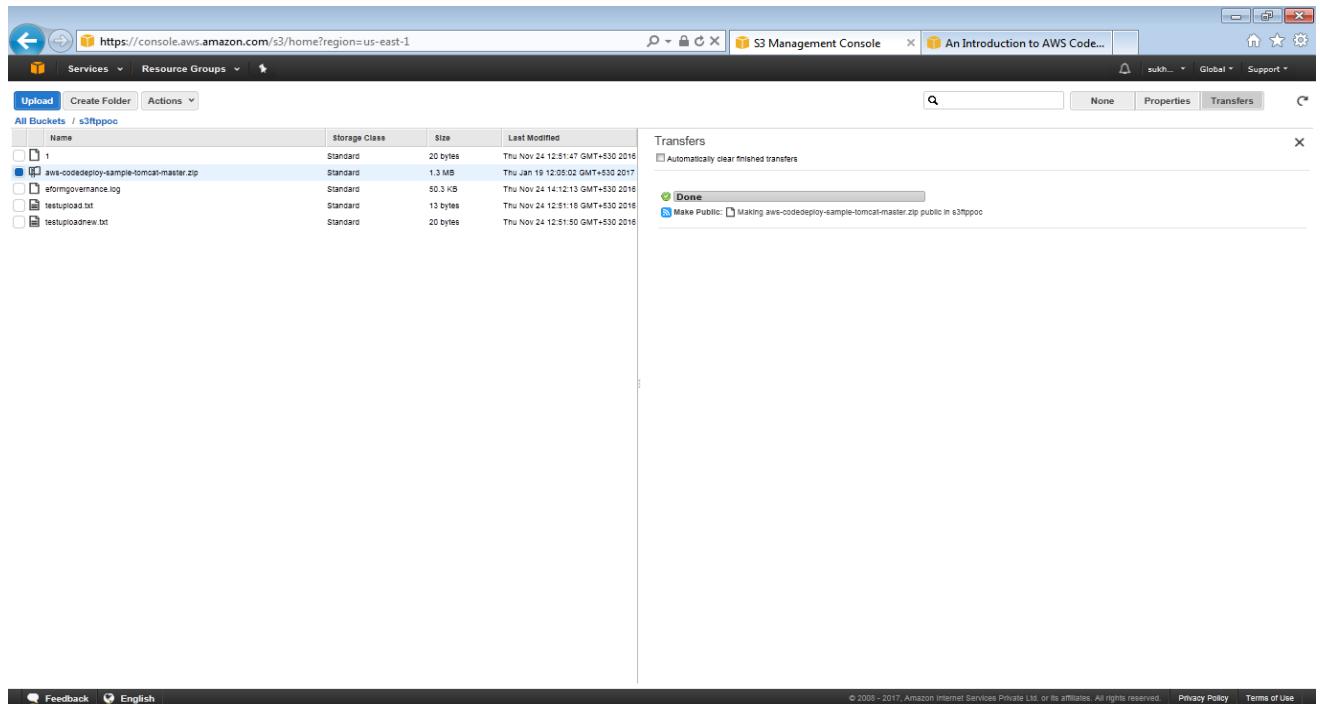
Cross-Region Replication

Tags

Requester Pays

Transfer Acceleration

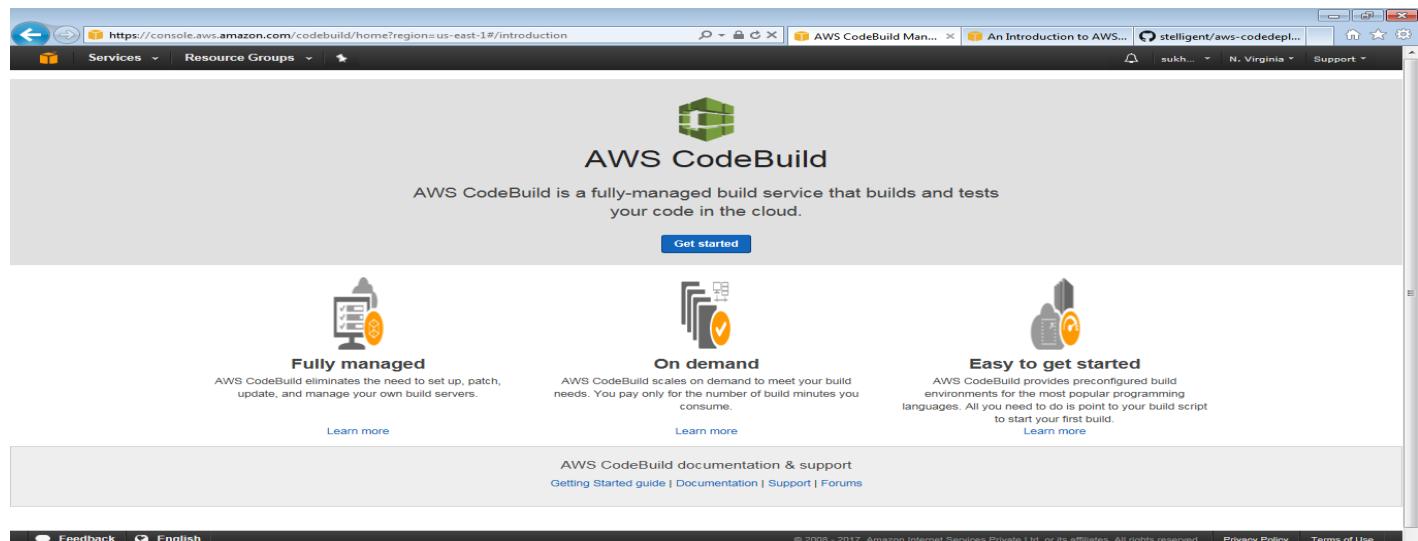
Storage Management



The screenshot shows the AWS S3 Management Console interface. On the left, there's a list of files in a bucket named 's3ftppoc'. One file, 'aws-codedeploy-sample-tomcat-master.zip', is currently being uploaded, indicated by a progress bar at the top of its row. The progress bar shows 'Done' with a green checkmark and the message 'Make Public: Making aws-codedeploy-sample-tomcat-master.zip public in s3ftppoc'. The rest of the files listed are 'testlog.txt', 'testlognew.txt', and 'testupload.txt', all of which have been uploaded successfully.

17.2.2. Configuring CodeBuild

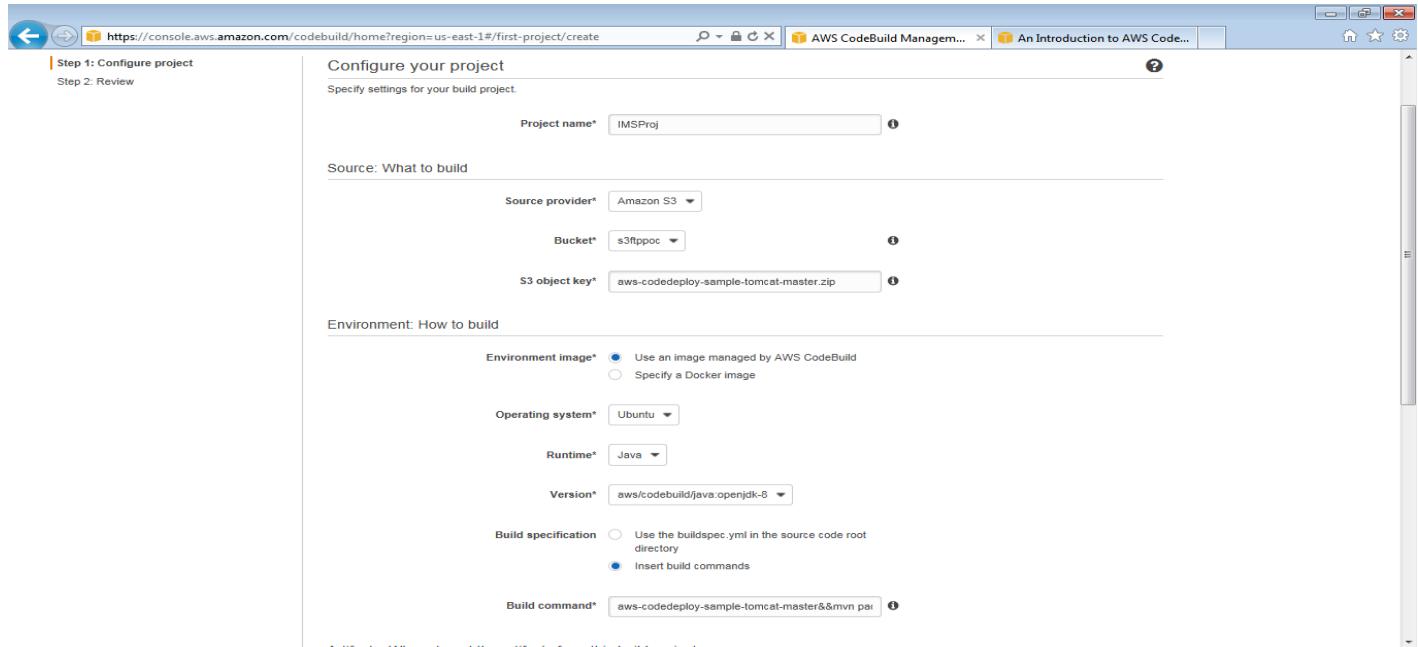
1. Click on the AWS **Code Build** service and **Click on Get Started** button
2. Provide the following details



The screenshot shows the AWS CodeBuild homepage. At the top, it features the AWS logo and the text 'AWS CodeBuild'. Below that, a sub-headline reads 'AWS CodeBuild is a fully-managed build service that builds and tests your code in the cloud.' A prominent blue 'Get started' button is centered below the headline. The page then highlights three main features:

- Fully managed**: An icon of a computer monitor with a checkmark. Description: 'AWS CodeBuild eliminates the need to set up, patch, update, and manage your own build servers.' A 'Learn more' link is provided.
- On demand**: An icon of a stack of documents with a play button. Description: 'AWS CodeBuild scales on demand to meet your build needs. You pay only for the number of build minutes you consume.' A 'Learn more' link is provided.
- Easy to get started**: An icon of a smartphone with a play button. Description: 'AWS CodeBuild provides preconfigured build environments for the most popular programming languages. All you need to do is point to your build script to start your first build.' A 'Learn more' link is provided.

At the bottom of the page, there's a footer with links to 'AWS CodeBuild documentation & support', 'Getting Started guide', 'Documentation', 'Support', and 'Forums'. The footer also includes standard Amazon navigation links like 'Feedback', 'English', and copyright information.



Step 1: Configure project

Configure your project
Specify settings for your build project.

Project name* IMSProj

Source: What to build

Source provider* Amazon S3

Bucket* s3ftppoc

S3 object key* aws-codedeploy-sample-tomcat-master.zip

Environment: How to build

Environment image* Use an image managed by AWS CodeBuild
 Specify a Docker image

Operating system* Ubuntu

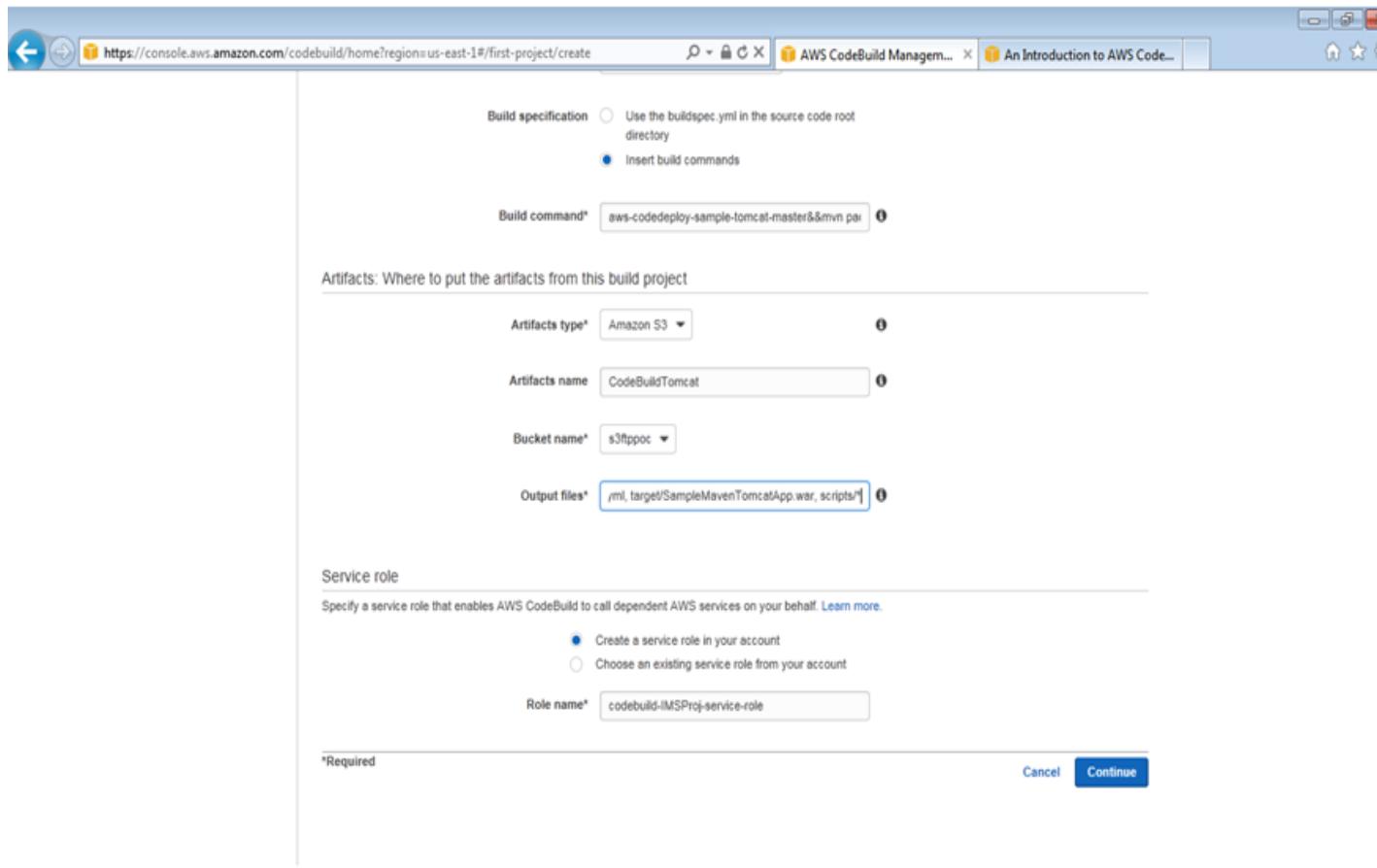
Runtime* Java

Version* aws/codebuild/java:openjdk-8

Build specification Use the buildspec.yml in the source code root directory
 Insert build commands

Build command* aws-codedeploy-sample-tomcat-master&&mvn package

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



Build specification Use the buildspec.yml in the source code root directory
 Insert build commands

Build command* aws-codedeploy-sample-tomcat-master&&mvn package

Artifacts: Where to put the artifacts from this build project

Artifacts type* Amazon S3

Artifacts name CodeBuildTomcat

Bucket name* s3ftppoc

Output files* /m1/target/Sample Maven Tomcat App.war, scripts/*

Service role

Specify a service role that enables AWS CodeBuild to call dependent AWS services on your behalf. [Learn more.](#)

Create a service role in your account
 Choose an existing service role from your account

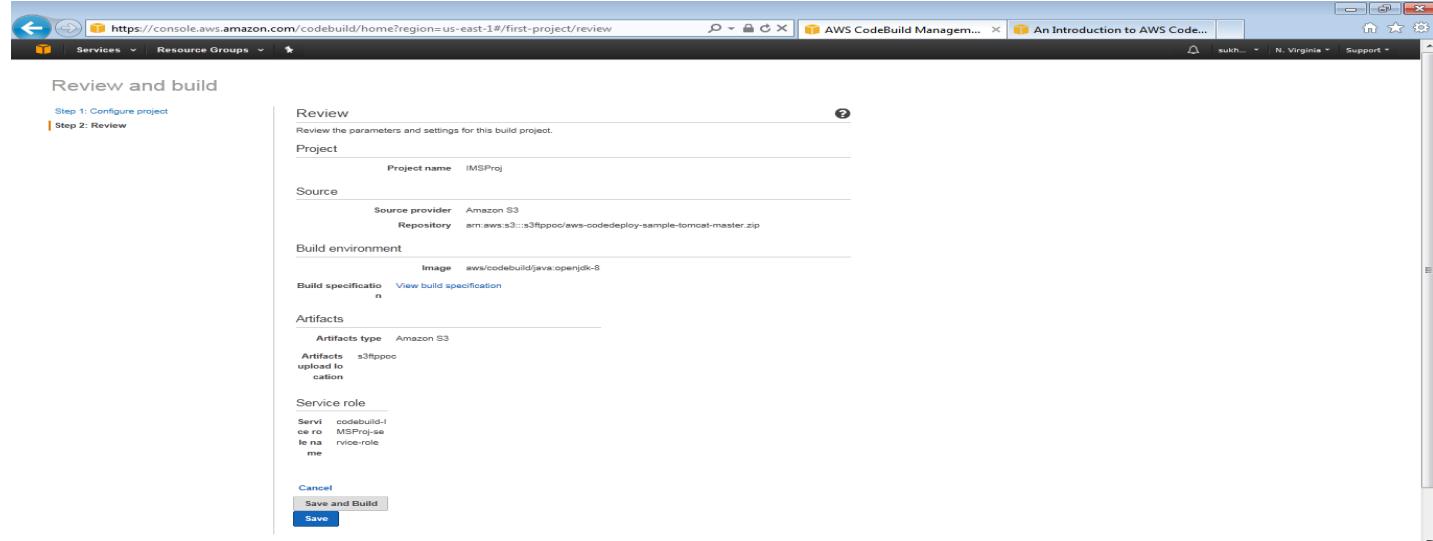
Role name* codebuild-IMSProj-service-role

*Required Cancel Continue

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/*

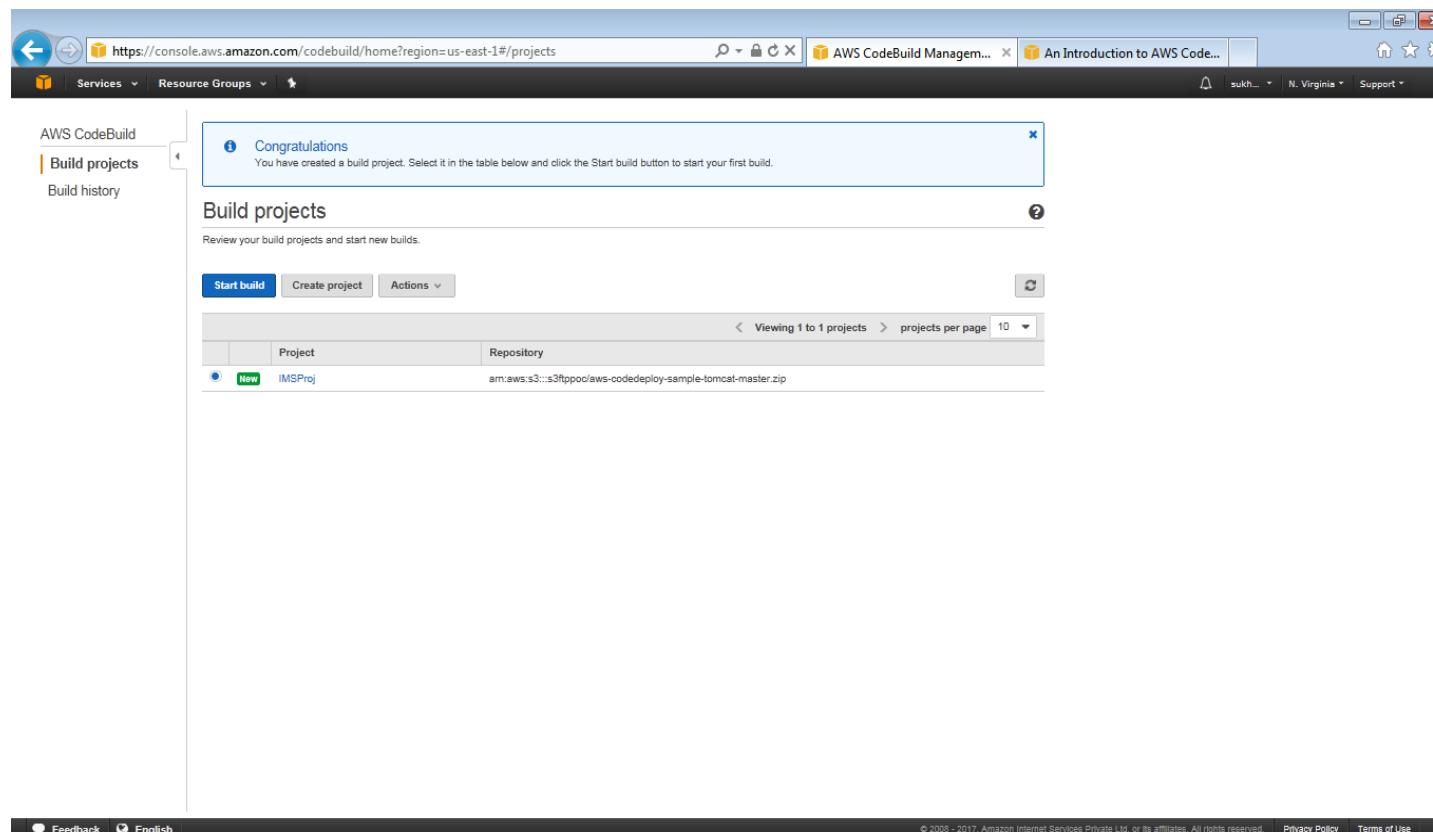
3. Click on Review and build the project



The screenshot shows the 'Review and build' step of the AWS CodeBuild configuration process. It displays the following details:

- Project:** Project name: IMSPProj
- Source:** Source provider: Amazon S3, Repository: arn:aws:s3:::s3fppoo/aws-codedeploy-sample-tomcat-master.zip
- Build environment:** Image: aws/codebuild/java:openjdk-8
- Build specification:** View build specification
- Artifacts:** Artifacts type: Amazon S3, Artifacts: s3fppoo, upload location: s3://s3fppoo
- Service role:** Service role: codebuild-lambda-MSProj-role

At the bottom, there are three buttons: Cancel, Save and Build (highlighted in blue), and Save.

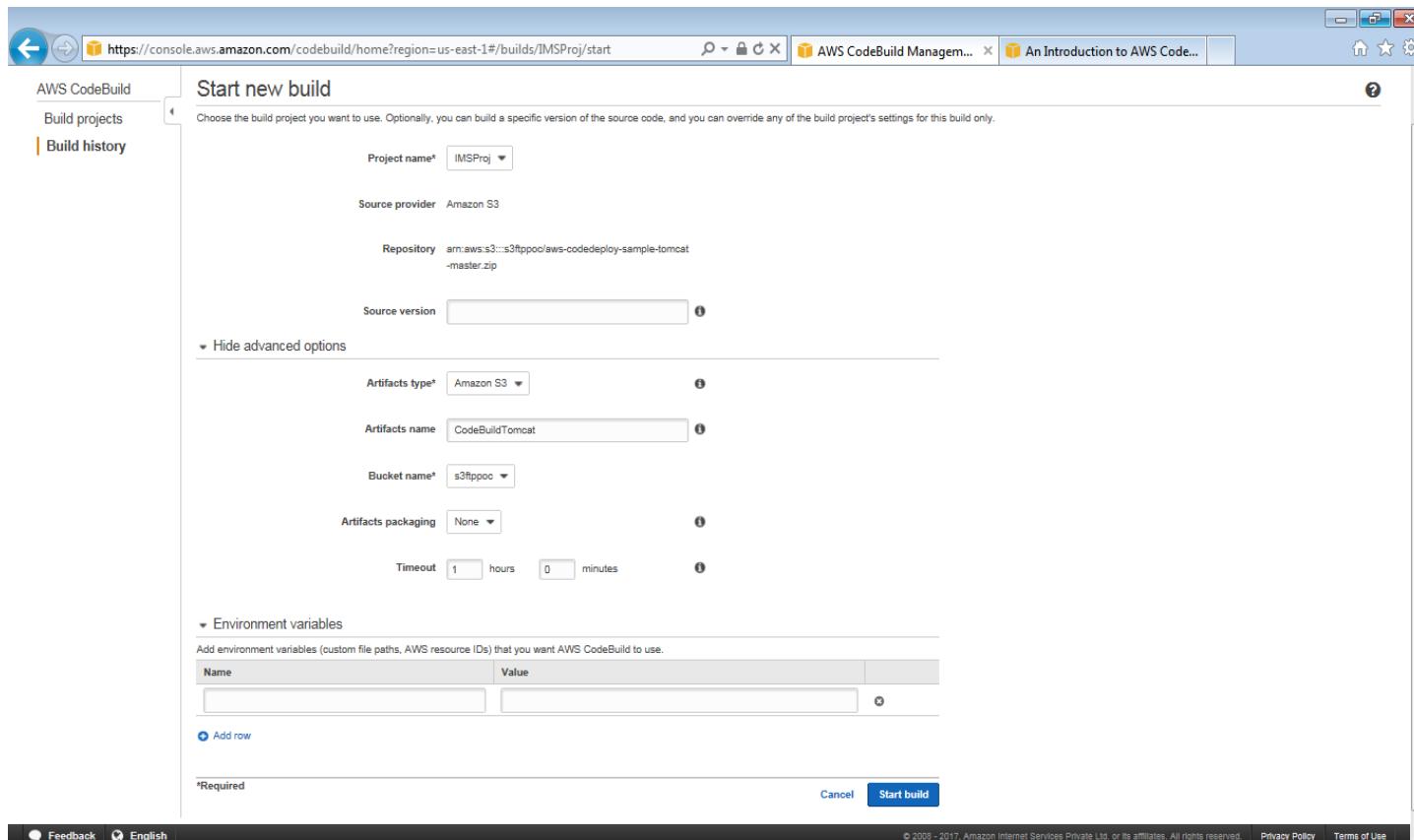


The screenshot shows the 'Build projects' step of the AWS CodeBuild management interface. It displays the following information:

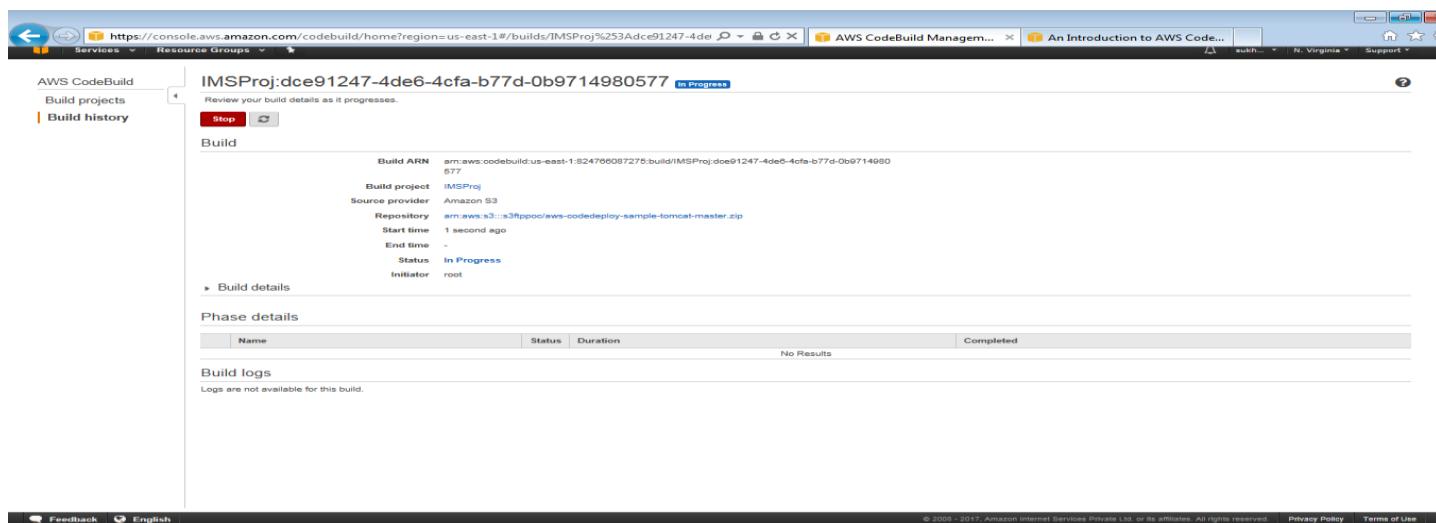
- A message box says: "Congratulations! You have created a build project. Select it in the table below and click the Start build button to start your first build."
- Build projects:** Review your build projects and start new builds.
- Actions:** Start build, Create project, Actions
- Table:** Viewing 1 to 1 projects, projects per page: 10

| | Project | Repository |
|--|----------|--|
| New | IMSPProj | arn:aws:s3:::s3fppoo/aws-codedeploy-sample-tomcat-master.zip |

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

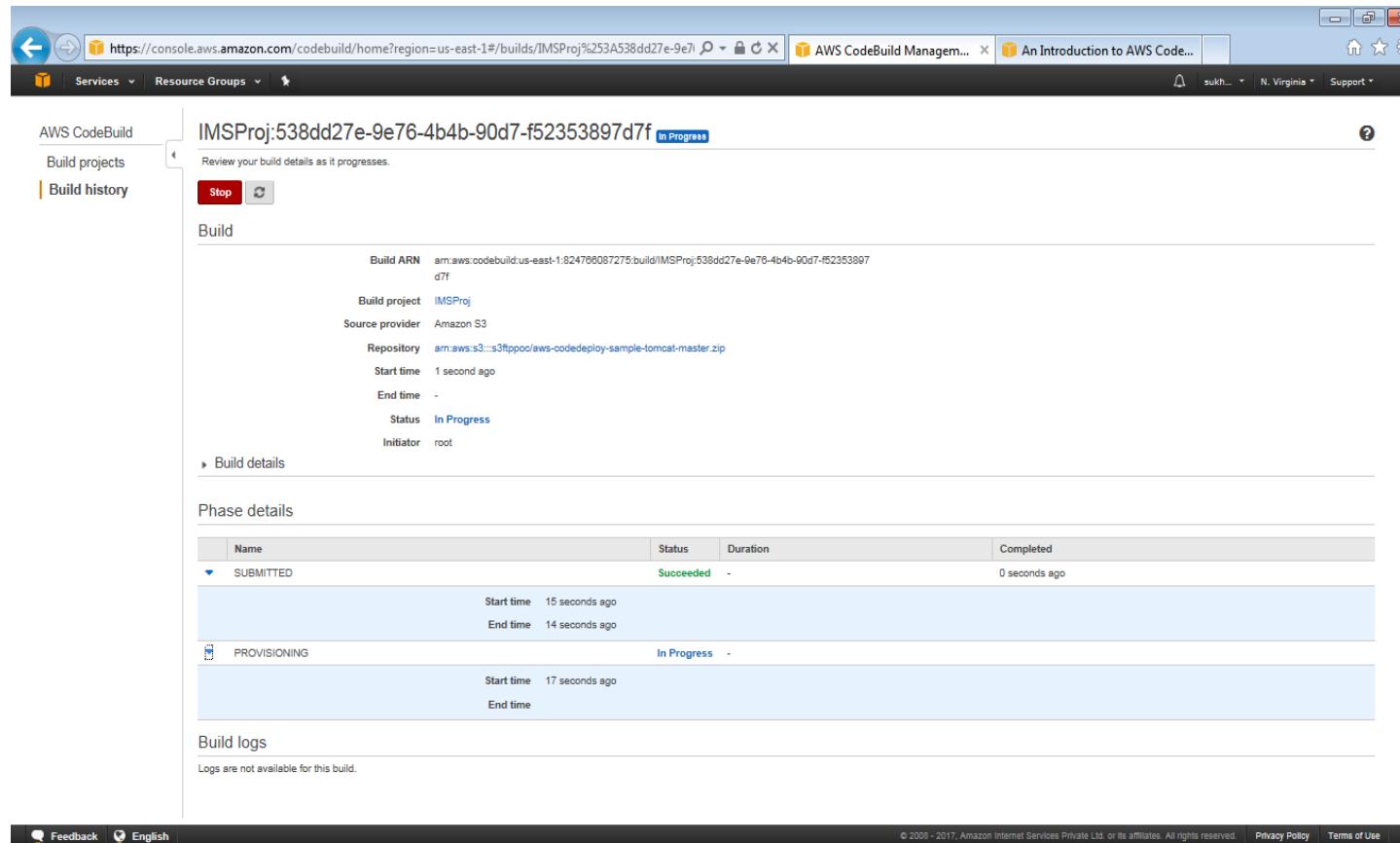


The screenshot shows the 'Start new build' configuration page for the 'IMSProj' project. The 'Project name*' dropdown is set to 'IMSProj'. The 'Source provider' is 'Amazon S3'. The 'Repository' is 'arn:aws:s3:::s3ftppoc/aws-codedeploy-sample-tomcat-master.zip'. The 'Source version' field is empty. Under 'Artifacts type*', 'Amazon S3' is selected. The 'Artifacts name' is 'CodeBuildTomcat'. The 'Bucket name*' is 's3ftppoc'. 'Artifacts packaging' is set to 'None'. The 'Timeout' is '1 hours 0 minutes'. There is a section for 'Environment variables' with a table for adding environment variables, but it is currently empty. At the bottom, there is a note '*Required' and a 'Start build' button.



The screenshot shows the 'Build details' page for build ID 'IMSProj:dce91247-4de6-4cfa-b77d-0b9714980577'. The status is 'In Progress'. The build ARN is 'arn:aws:codebuild:us-east-1:824706087276:build/IMSProj:dce91247-4de6-4cfa-b77d-0b9714980577:577'. The build project is 'IMSProj'. The source provider is 'Amazon S3'. The repository is 'arn:aws:s3:::s3ftppoc/aws-codedeploy-sample-tomcat-master.zip'. The start time was '1 second ago'. The status is 'In Progress' and the initiator is 'root'. Below this, there is a 'Build logs' section which states 'Logs are not available for this build.'

Wait for the build to succeed



AWS CodeBuild

Build projects

Build history

IMSProj:538dd27e-9e76-4b4b-90d7-f52353897d7f In Progress

Review your build details as it progresses.

Stop **Cancel**

Build

| | |
|-----------------|---|
| Build ARN | arn:aws:codebuild:us-east-1:824760087275:build/IMSProj:538dd27e-9e76-4b4b-90d7-f52353897d7f |
| Build project | IMSProj |
| Source provider | Amazon S3 |
| Repository | arn:aws:s3:::s3ftppoc/aws-codedeploy-sample-tomcat-master.zip |
| Start time | 1 second ago |
| End time | - |
| Status | In Progress |
| Initiator | root |

▶ Build details

Phase details

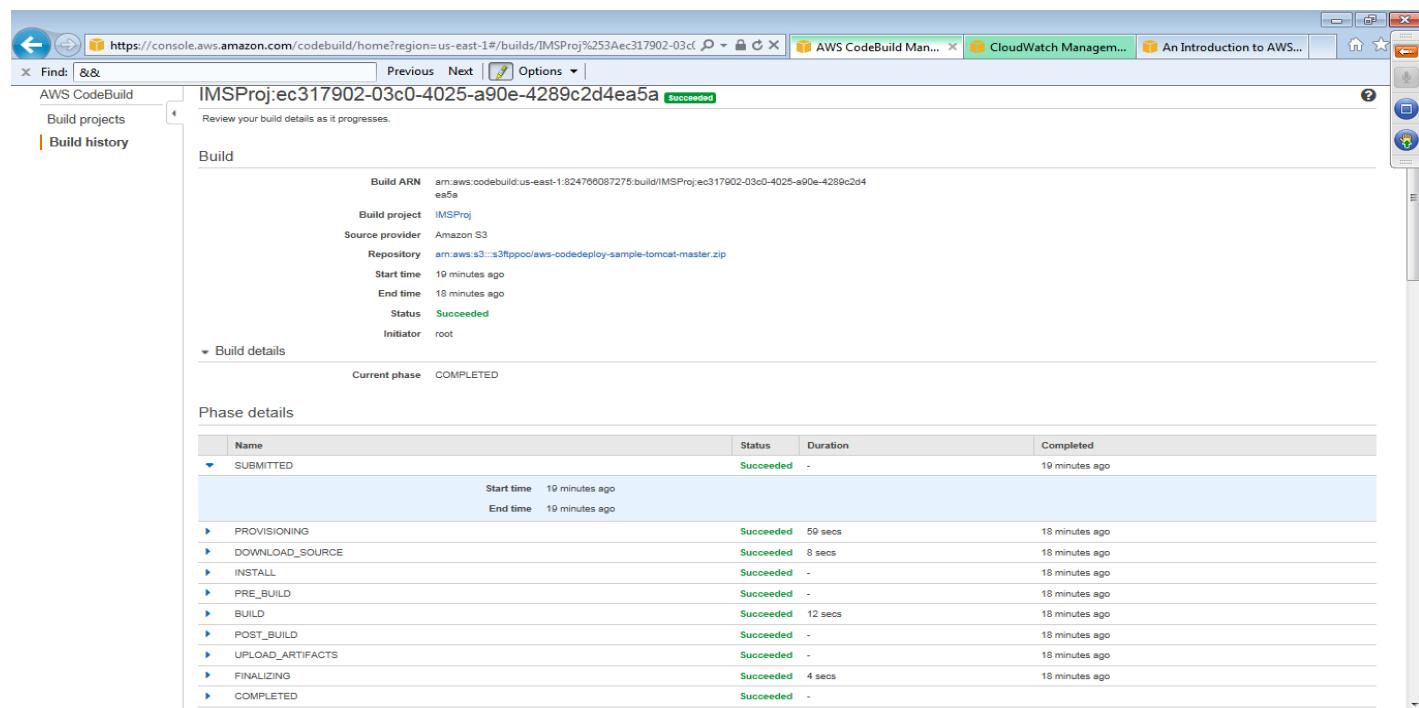
| Name | Status | Duration | Completed |
|--------------|-------------|----------|---------------|
| SUBMITTED | Succeeded | - | 0 seconds ago |
| PROVISIONING | In Progress | - | |

Start time 15 seconds ago
End time 14 seconds ago

Start time 17 seconds ago
End time

Build logs

Logs are not available for this build.



AWS CodeBuild

Build projects

Build history

IMSProj:ec317902-03c0-4025-a90e-4289c2d4ea5a Succeeded

Review your build details as it progresses.

Build

| | |
|-----------------|---|
| Build ARN | arn:aws:codebuild:us-east-1:824760087275:build/IMSProj:ec317902-03c0-4025-a90e-4289c2d4ea5a |
| Build project | IMSProj |
| Source provider | Amazon S3 |
| Repository | arn:aws:s3:::s3ftppoc/aws-codedeploy-sample-tomcat-master.zip |
| Start time | 19 minutes ago |
| End time | 18 minutes ago |
| Status | Succeeded |
| Initiator | root |

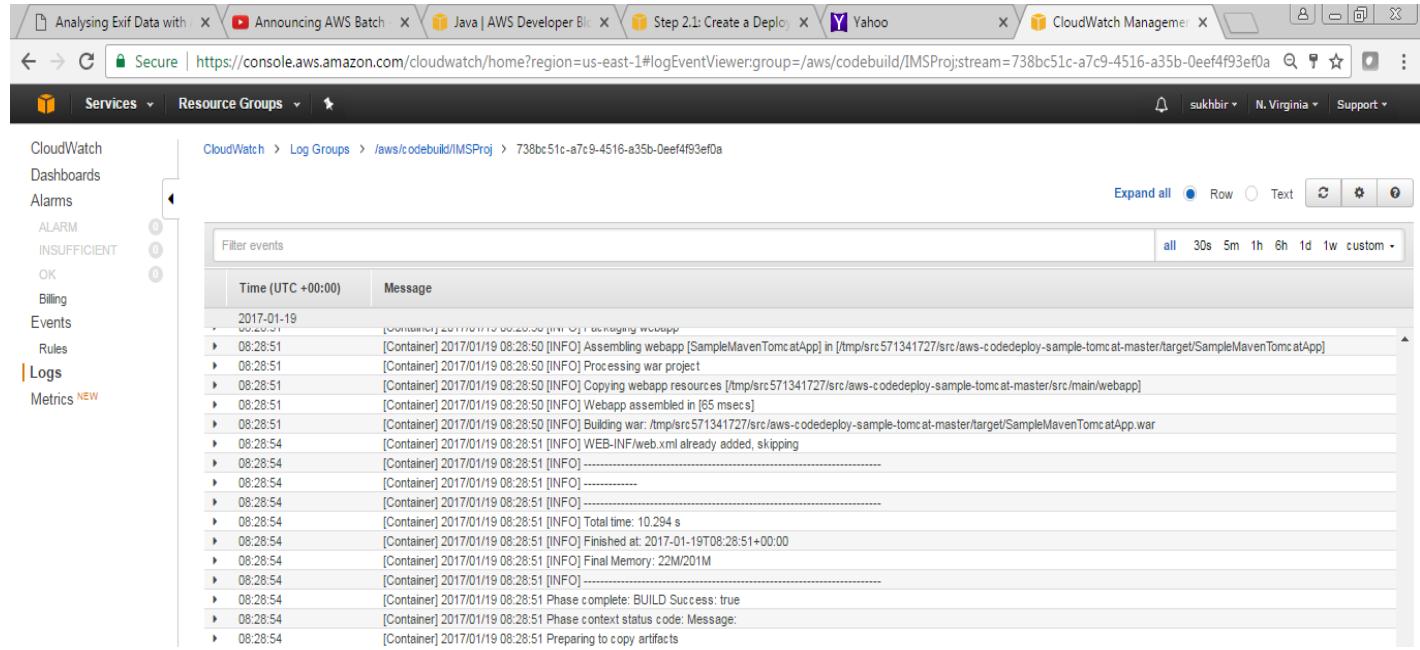
▼ Build details

Current phase: COMPLETED

Phase details

| Name | Status | Duration | Completed |
|------------------|-----------|----------|----------------|
| SUBMITTED | Succeeded | - | 19 minutes ago |
| PROVISIONING | Succeeded | 59 secs | 18 minutes ago |
| DOWNLOAD_SOURCE | Succeeded | 8 secs | 18 minutes ago |
| INSTALL | Succeeded | - | 18 minutes ago |
| PRE_BUILD | Succeeded | - | 18 minutes ago |
| BUILD | Succeeded | 12 secs | 18 minutes ago |
| POST_BUILD | Succeeded | - | 18 minutes ago |
| UPLOAD_ARTIFACTS | Succeeded | - | 18 minutes ago |
| FINALIZING | Succeeded | 4 secs | 18 minutes ago |
| COMPLETED | Succeeded | - | |

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

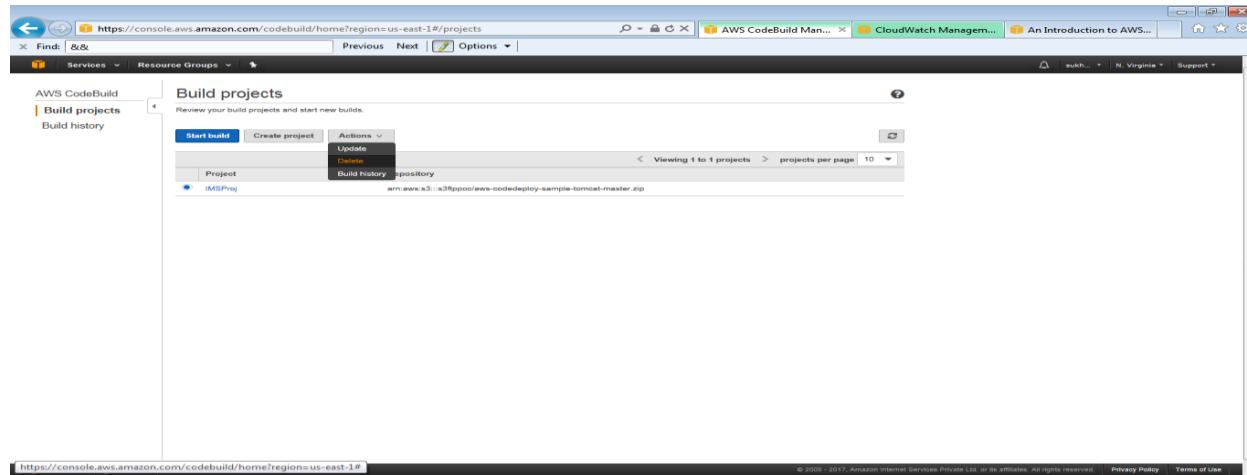


The screenshot shows the AWS CloudWatch Logs interface. On the left, a sidebar lists services: CloudWatch, Dashboards, Alarms, ALARM (INSUFFICIENT, OK), Billing, Events, Rules, Logs (selected), and Metrics. The main area shows a log group path: CloudWatch > Log Groups > /aws/codebuild/IMSPProj > 738bc51c-a7c9-4516-a35b-0eef4f93ef0a. A table titled 'Filter events' displays log messages with columns for Time (UTC +00:00) and Message. The log entries show the build process starting at 2017-01-19 08:28:51, assembling a webapp, processing a war project, copying resources, building the war file, and finally preparing to copy artifacts at 08:28:54.

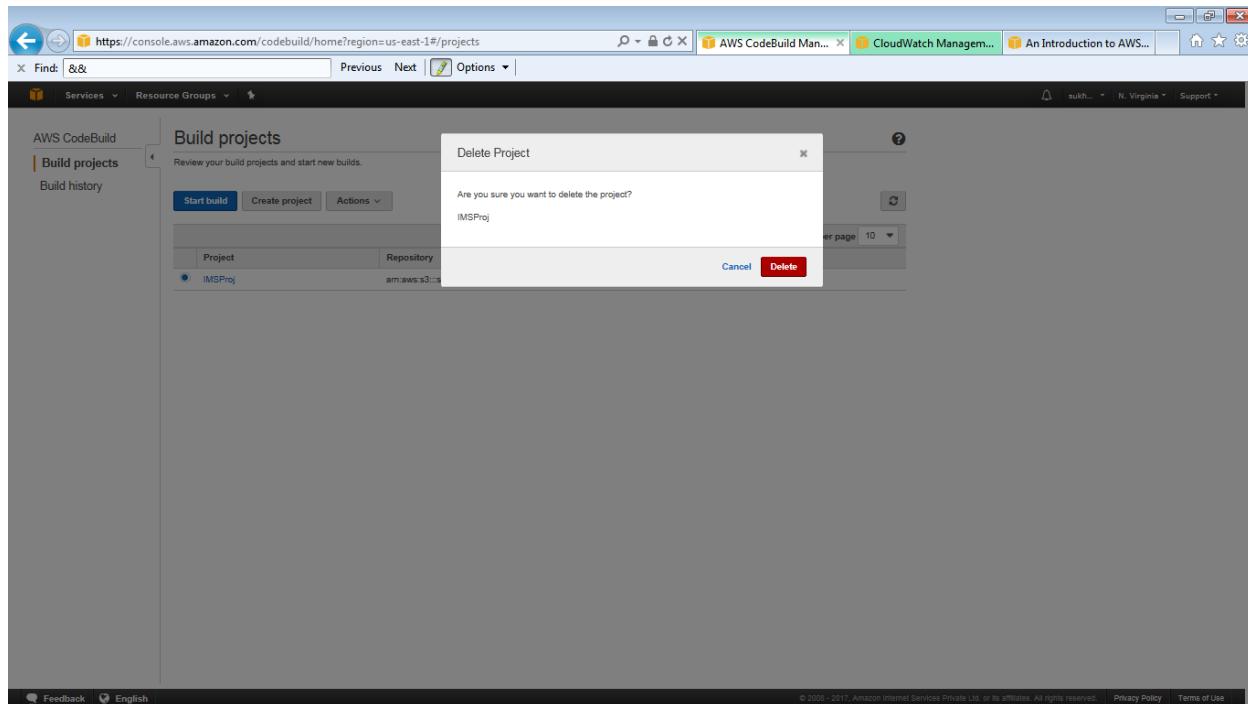
Once it has completed, you can open the S3 bucket you specified for the artifact and view the S3 bucket with the *SampleMavenTomcatApp.war* file.

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

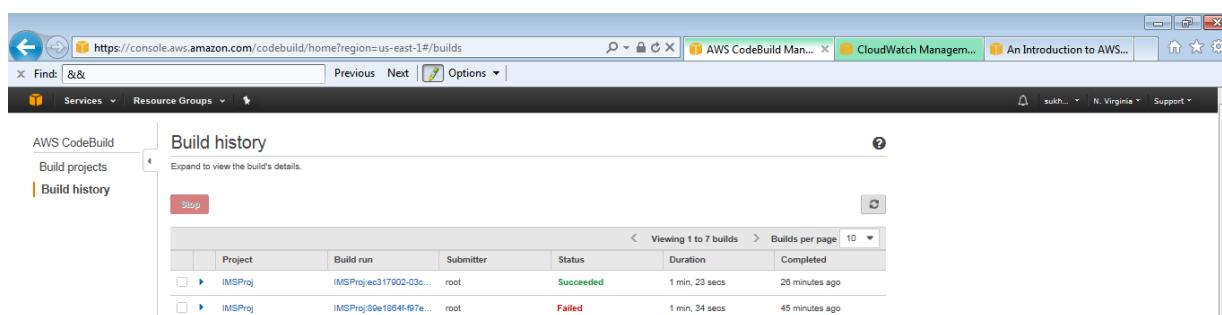
17.2.3 Delete CodeBuild



The screenshot shows the AWS CodeBuild console under the 'Build projects' section. It lists one project, 'IMSPProj', which is associated with the repository 'arn:aws:s3:::s3ftppos/aws-codedeploy-sample-tomcat-master.zip'. There are buttons for 'Start build', 'Create project', 'Actions' (with options for 'Update' and 'Delete'), and 'Build history'.



The screenshot shows the AWS CodeBuild console with the URL <https://console.aws.amazon.com/codebuild/home?region=us-east-1#/projects>. A modal dialog titled "Delete Project" is open, asking "Are you sure you want to delete the project?" with the project name "IMSPProj". The "Delete" button is highlighted in red.



The screenshot shows the AWS CodeBuild console with the URL <https://console.aws.amazon.com/codebuild/home?region=us-east-1#/builds>. The "Build history" section is displayed, showing two build runs for the project "IMSPProj". The first build run is listed as "Succeeded" and the second as "Failed".

| | Project | Build run | Submitter | Status | Duration | Completed |
|--------------------------|----------|--------------------------|-----------|-----------|----------------|----------------|
| <input type="checkbox"/> | IMSPProj | IMSPProj@317902-03e... | root | Succeeded | 1 min, 23 secs | 20 minutes ago |
| <input type="checkbox"/> | IMSPProj | IMSPProj@9e1954f-f97e... | root | Failed | 1 min, 34 secs | 45 minutes ago |