High Availability Across Availability Zones

In this lab, we are going to implement a Highly Available architecture for Windows Server based Internet Information Services (IIS) Web Applications.

Though we use technologies like VMware, OpenStack, AWS Cloud, Azure Cloud or GCP, all the technologies are implemented on Physical Infrastructure. There are more chances of failure of any physical component like HDD failures, Power Supply Failures, OS crash or any Catastrophic Events. Though Cloud provider companies provide high level of Service Level Agreements (SLAs), you have to always architect your Infrastructure in such a way that it has a redundant or standby path in-case of failure so that your applications will be running 24x7 and your customers face less downtime.

On AWS cloud, we can configure Highly Available architecture in multiple ways. For this lab, we are going to design a Highly Available architecture across Availability Zones.

Since you are more exposed to Linux based Apache server configurations through the Cloud Architecting and Cloud Foundations labs, I've selected Windows Server based IIS to let you know how configure IIS role on Windows Servers and how to access Windows Servers hosted on AWS Cloud using Key Pairs.

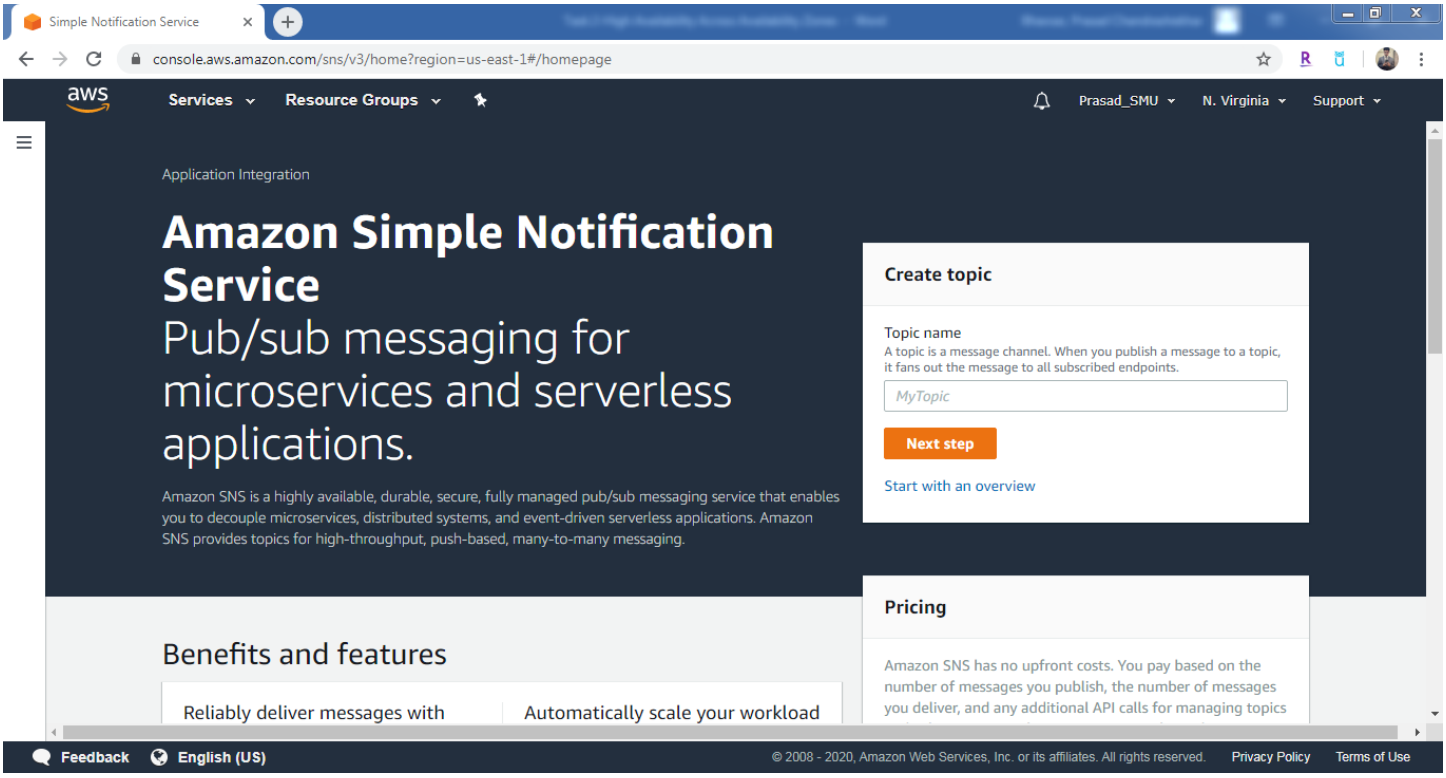
This lab consists of following topics:

- Task 1: Create a SNS topic.
- Task 2: Create Custom Security Groups for EC2 Instances and Load Balancer.
- Task 3: Launch an EC2 Instance-Windows Server 2016 and configure IIS role on it.
- Task 4: Create an Amazon Machine Image (AMI) of the EC2 Instance.
- Task 5: Create an Internet-facing application load balancer.
- Task 6: Create Launch Configurations, Auto Scaling groups (ASG), Auto Scaling Policies and CloudWatch Alarms.
- Task 7: Test the Failover.

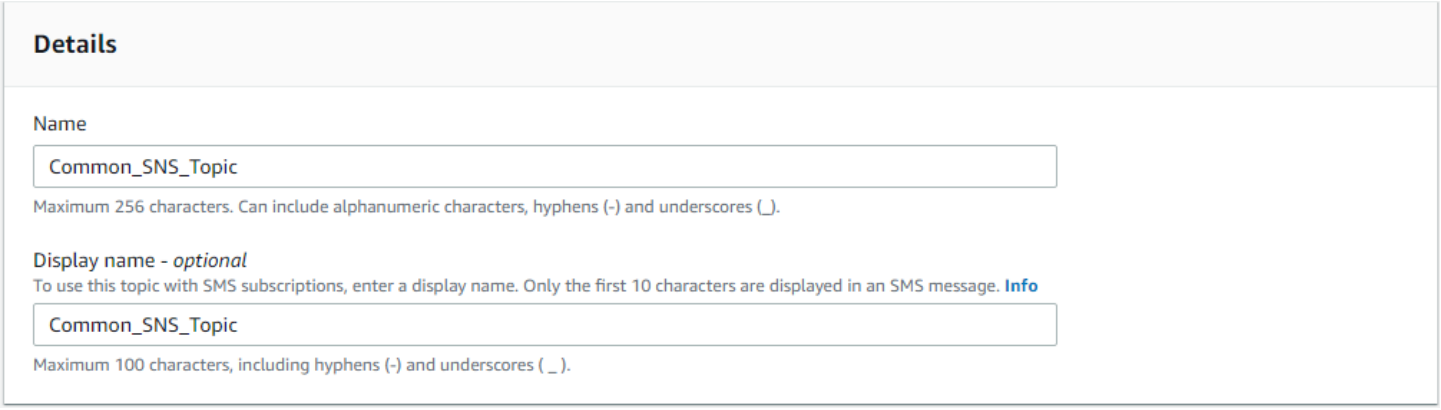
Task 1: Create a SNS topic.

We are now going to create a common SNS Topic and Subscriptions. We are going to use for this SNS topic for the next set of labs as well.

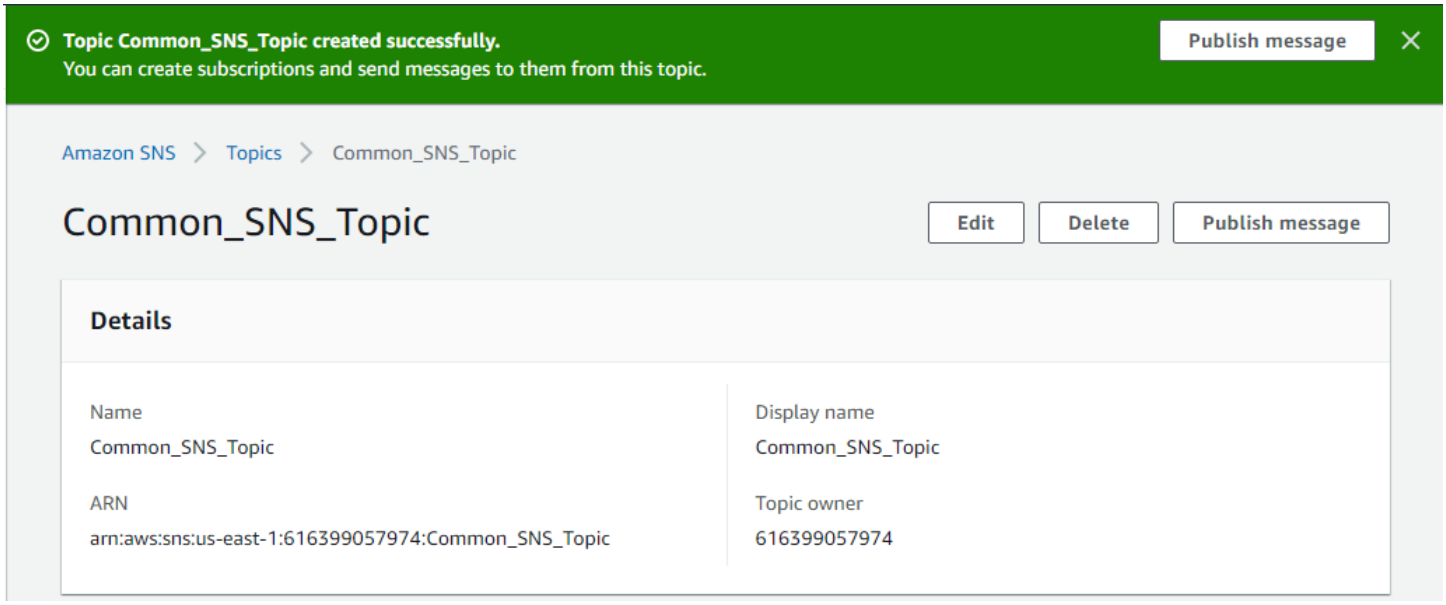
Open AWS Management console and navigate to SNS service.



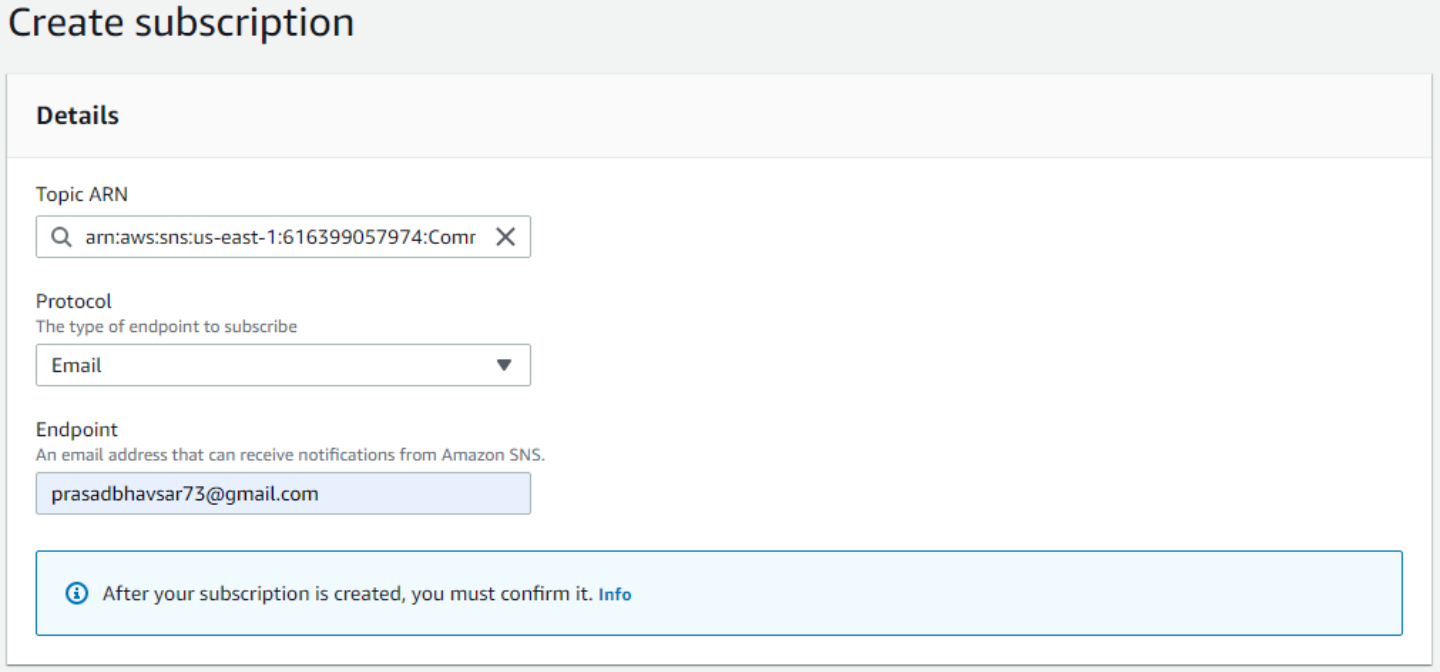
Under Create Topic, click on Next. Give topic name as of your choice and click Create Topic.



You've now successfully configured a SNS topic.



Next Click on Subscriptions and Click on Protocol. I'm going to select Protocol as Email. Enter the endpoint as your Email Address and click on Create Subscription.



On the Email you might have now received an Email for Subscription Confirmation. Confirm your Email Address Subscription and now you're ready to go.

amazon webservices™ Simple Notification Service

Subscription confirmed!

You have subscribed prasadbhavsar73@gmail.com to the topic: **Common_SNS_Topic**.

Your subscription's id is:
arn:aws:sns:us-east-1:616399057974:Common_SNS_Topic:a448bf30-469d-482a-813a-9ef0f8136c75

If it was not your intention to subscribe, click here to unsubscribe.

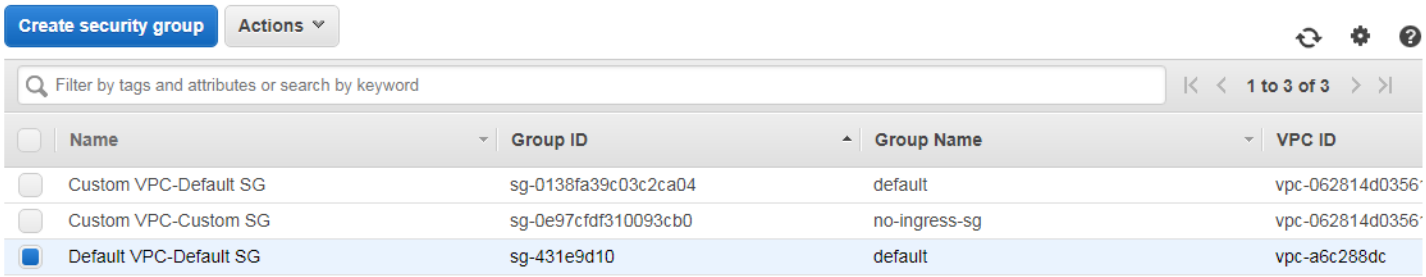
Go back to AWS Console, you'll now see the status of the Subscription as CONFIRMED.

	ID	Endpoint	Status	Protocol	Topic
○	a448bf30-469d-482a-813a-9ef0f8136c75	prasadbhavsar73@gmail.com	Confirmed	EMAIL	Common_SNS_Topic

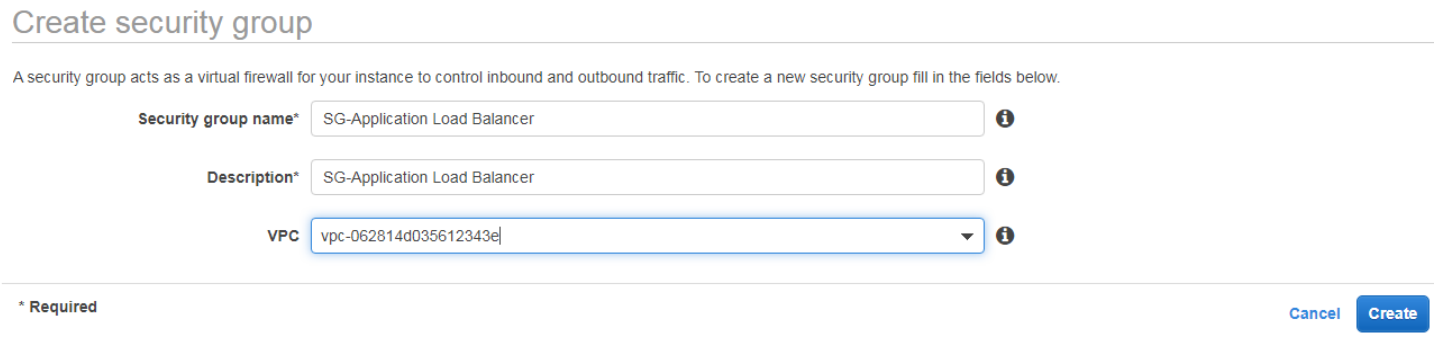
You've now setup a SNS Topic (Common_SNS_Topic) with Subscription as Email of your choice.

Task 2: Create Custom Security Groups for EC2 Instances and Load Balancer.

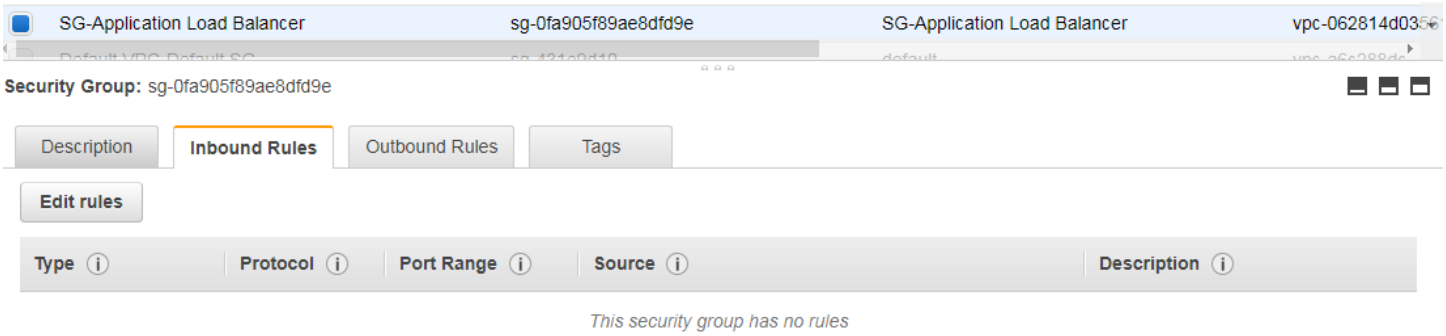
Navigate to VPC Service and Click on Security Groups.



Click on Create Security Group. We'll now create a Security Group for Application Load Balancer. Give Security Group name as per your choice and select the VPC as Custom VPC which we deployed in the previous lab and click on Create.



Select the Security Group and click on Inbound Rules and then click on Edit Rules.



You're now going to allow HTTP and HTTPS traffic on the Load Balancer. Click on Add Rule and add both the rules as follows.

Type	Protocol	Port Range	Source		Description
HTTPS	TCP	443	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Now click on Save Rules. You can leave Outbound Rules as default since it is allowing all the traffic by default.

SG-Application Load Balancer sg-0fa905f89ae8dfd9e SG-Application Load Balancer vpc-062814d03

Security Group: sg-0fa905f89ae8dfd9e

Description | **Inbound Rules** | Outbound Rules | Tags

Edit rules

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	::/0	

Again, click on Create Security Group and create a Security Group for Windows Servers with allowing Inbound Rules to only HTTP and RDP traffic. Make sure to allow RDP traffic to only to your Systems Public IP. You will be the only one who can take RDP of this Server now.

Type	Protocol	Port Range	Source		Description
RDP	TCP	3389	My IP	70.123.124.218/32	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

SG-Windows Servers sg-04bd568afe8fa65c4 SG-Windows Servers vpc-062814d03

Security Group: sg-04bd568afe8fa65c4

Description | **Inbound Rules** | Outbound Rules | Tags

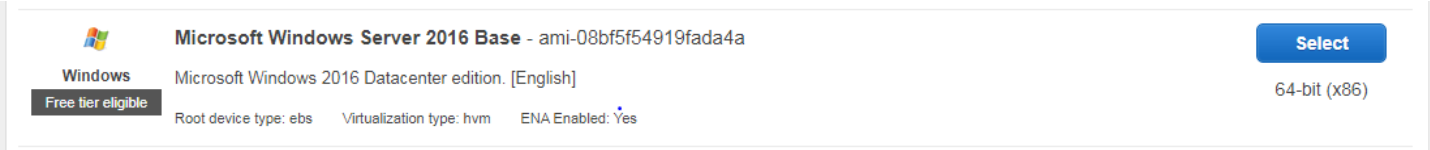
Edit rules

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
RDP	TCP	3389	70.123.124.218/32	

Task 3: Launch an EC2 Instance-Windows Server 2016 and configure IIS role on it.

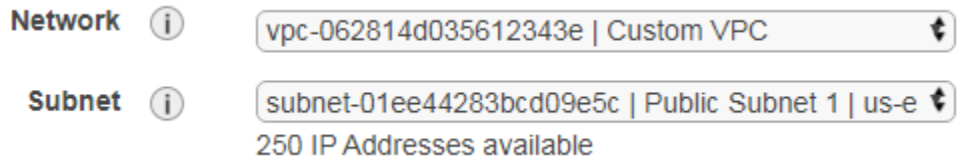
Navigate to EC2 service and Click on Launch Instance.

Click on Free Tier Only and Select **Windows Server 2016 Base**.

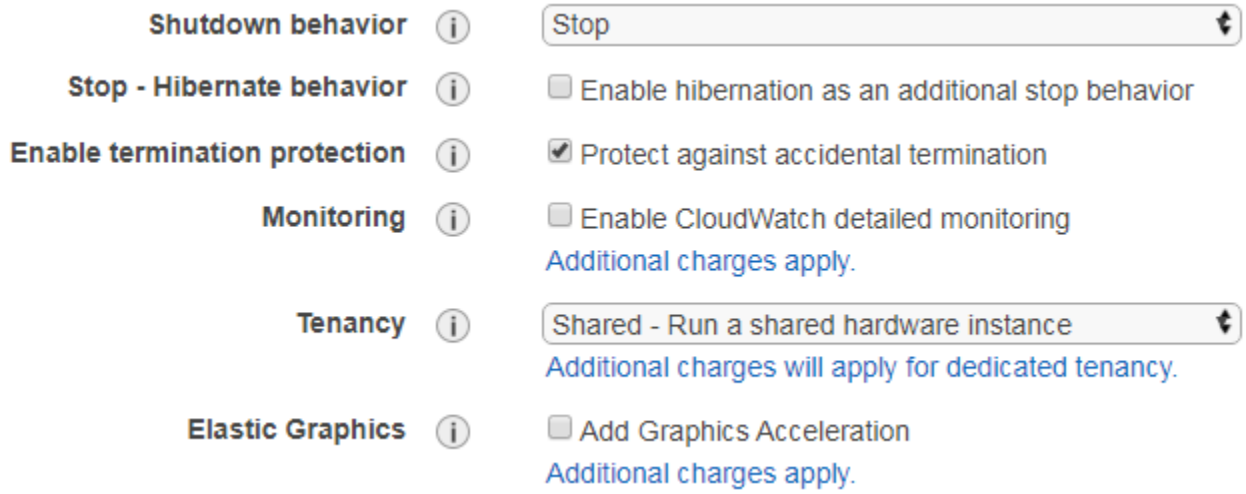


Choose Default Instance Type as **t2.micro** and configure Instance Details.

Select the Network as Custom VPC and Subnet as Public Subnet 1.



Select the Shutdown behavior as STOP, it means when your EC2 Instance crashes at the OS level, the Instance will get STOP automatically. Also enable the Protect against accidental termination.



Click on Next: Add Storage.

Keep the default EBS Root Volumes Settings as it is and Click Next: Add Tags.

We are not going to add any Tags here, hence click Next: Configure Security Groups.

Click on “Select an existing Security Group” and click on SG-Windows Servers security group that we’ve configured in the previous task.

Security Group ID	Name	Description	Actions
sg-0138fa39c03c2ca04	default	default VPC security group	Copy to new
sg-0e97cfdf310093cb0	no-ingress-sg	Security group with no ingress rule	Copy to new
sg-0fa905f89ae8dfd9e	SG-Application Load Balancer	SG-Application Load Balancer	Copy to new
sg-04bd568afe8fa65c4	SG-Windows Servers	SG-Windows Servers	Copy to new

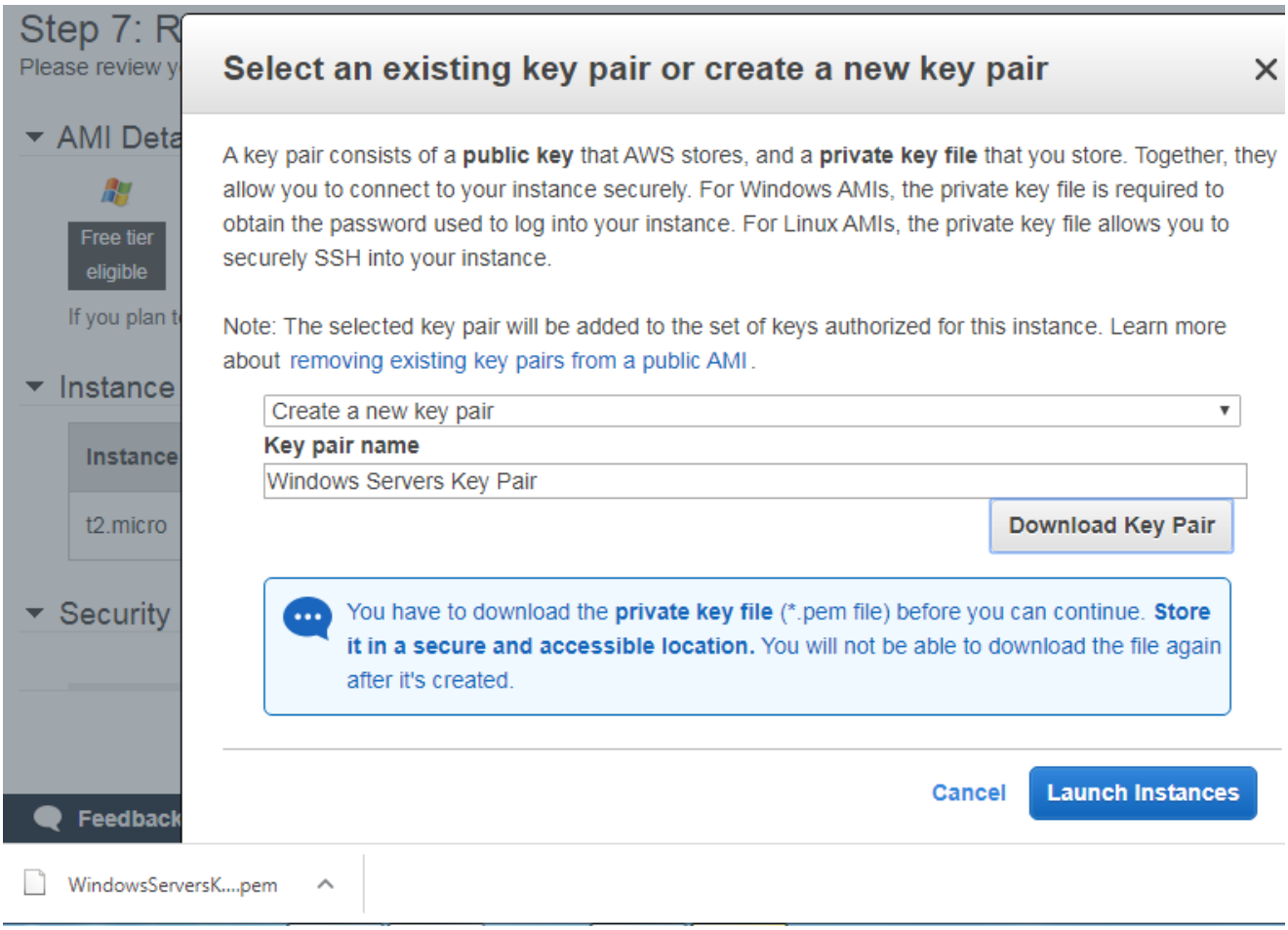
Inbound rules for sg-04bd568afe8fa65c4 (Selected security groups: sg-04bd568afe8fa65c4)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	

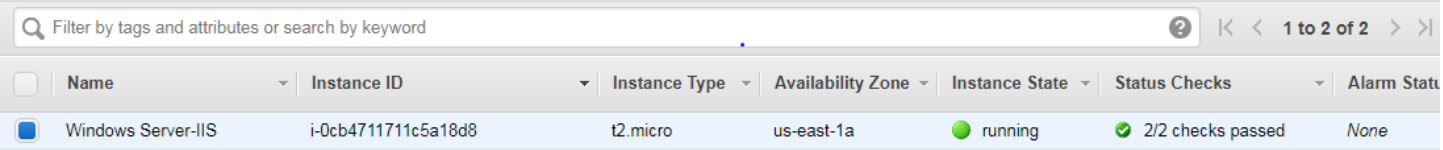
Click on Next: Review Instance Launch.

Review the Instance setting and click on Launch.

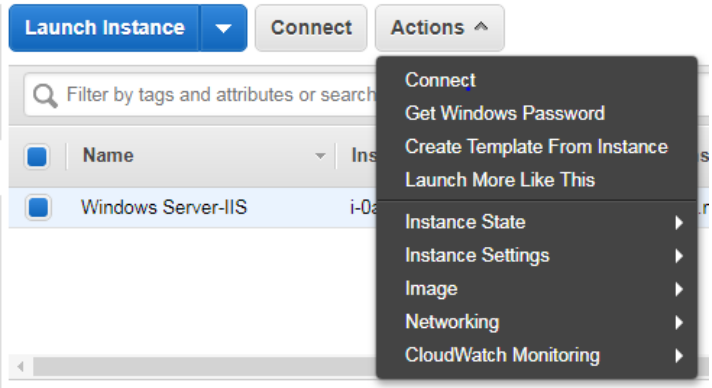
Now in **Select an existing key pair or create new key pair** box, click on Create a Key Pair and give name as per your choice and download the Key. Keep this key in safe location in your desktop. Now click on Launch Instances.



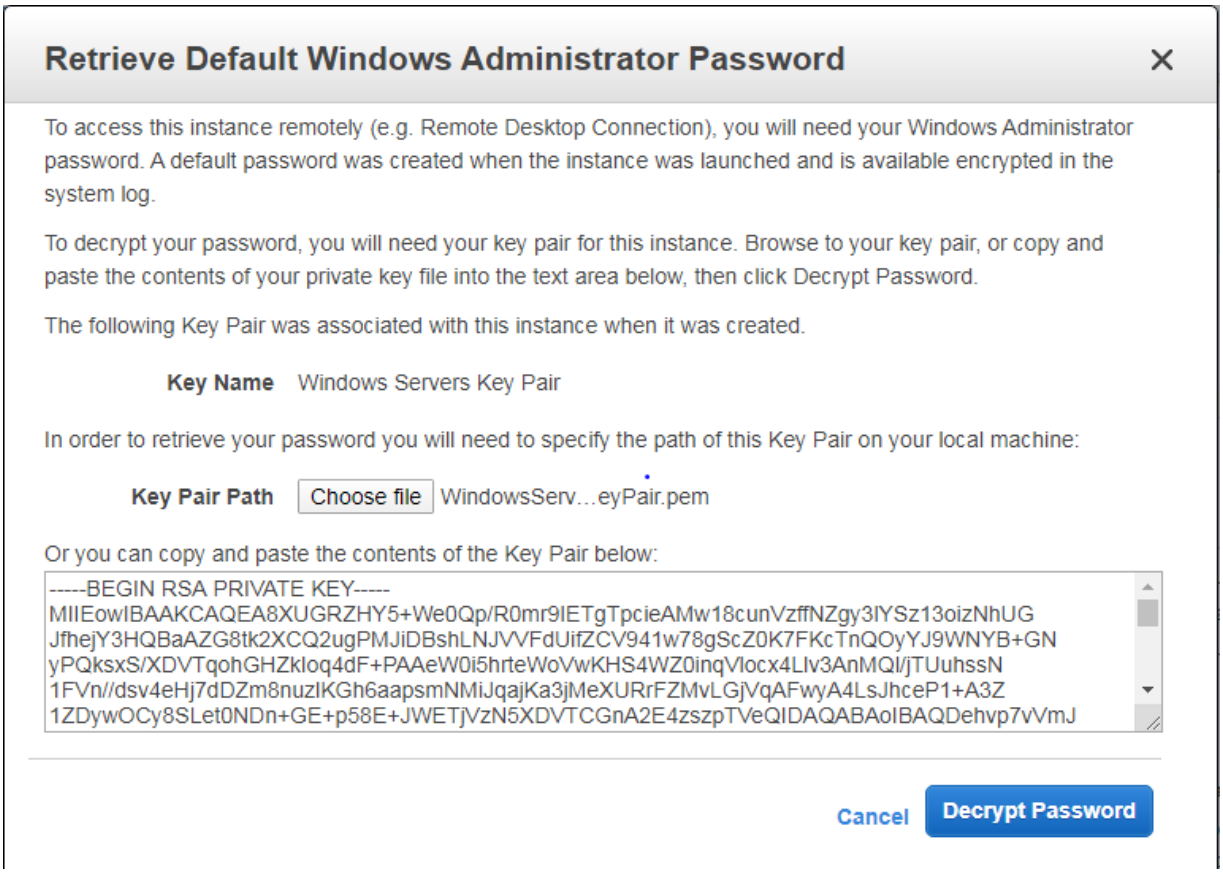
Wait till you observe the Instance State as running and Status Checks as 2/2 Checks Passed.



Select the EC2 Instance and click on Actions and then click on Get Windows Password.

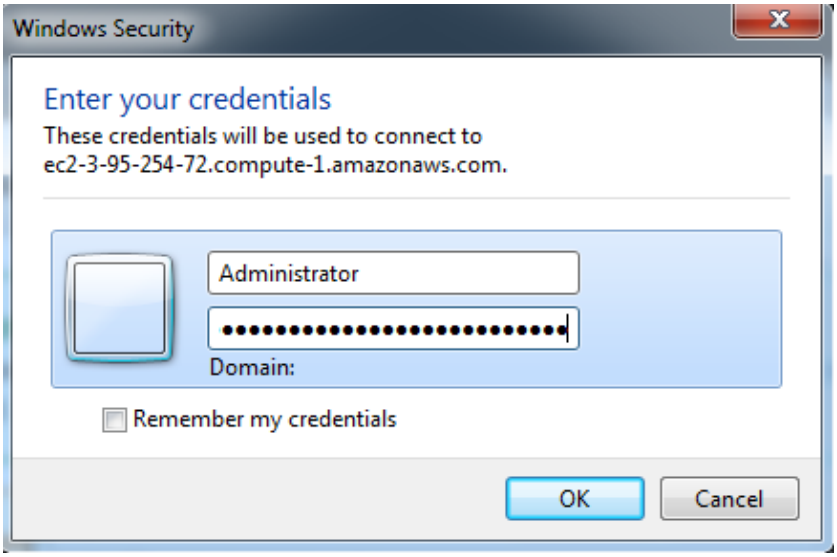


In the Retrieve Default Windows Administrator Password box, browse to the Key Pair Path and Choose the Key Pair that you've Downloaded.



Now click on Decrypt Password. Now copy the Public DNS, Username and Password to Notepad. You'll need this information whenever you want to take RDP of this EC2 Instance.

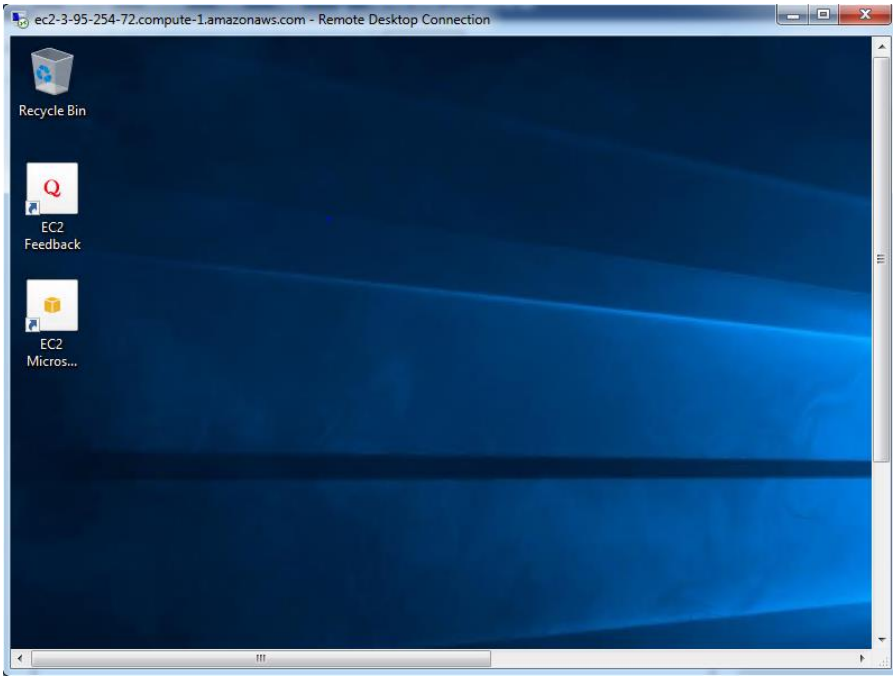
On your Home Computer, click on Start and Select Remote Desktop Connection. Put the DNS name or the Public IP Address of the EC2 Instance, Username and Password and hit ok.



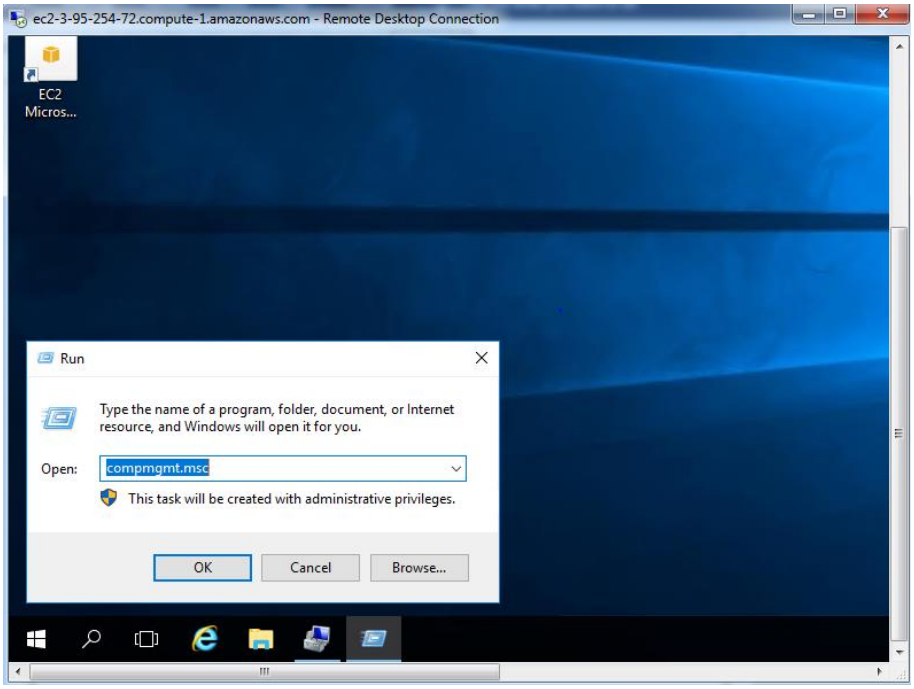
Windows Server Administration Tasks:

In this part, you'll perform some of the Windows Systems Administrations Tasks.

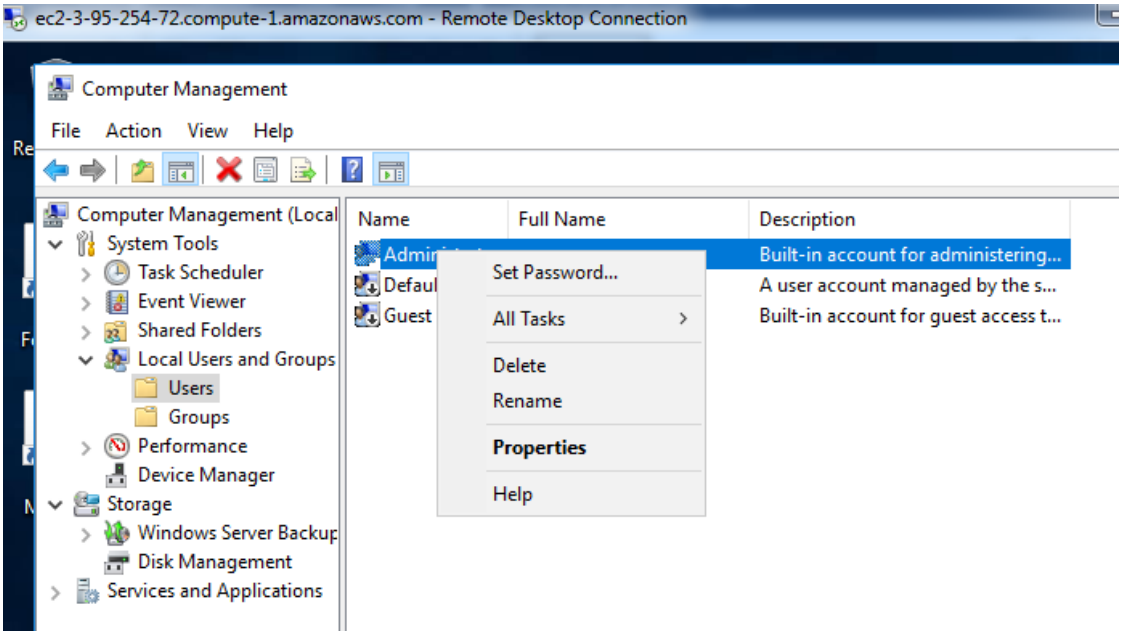
Congratulations, you have now launched your first Windows Server 2016 on AWS Cloud.



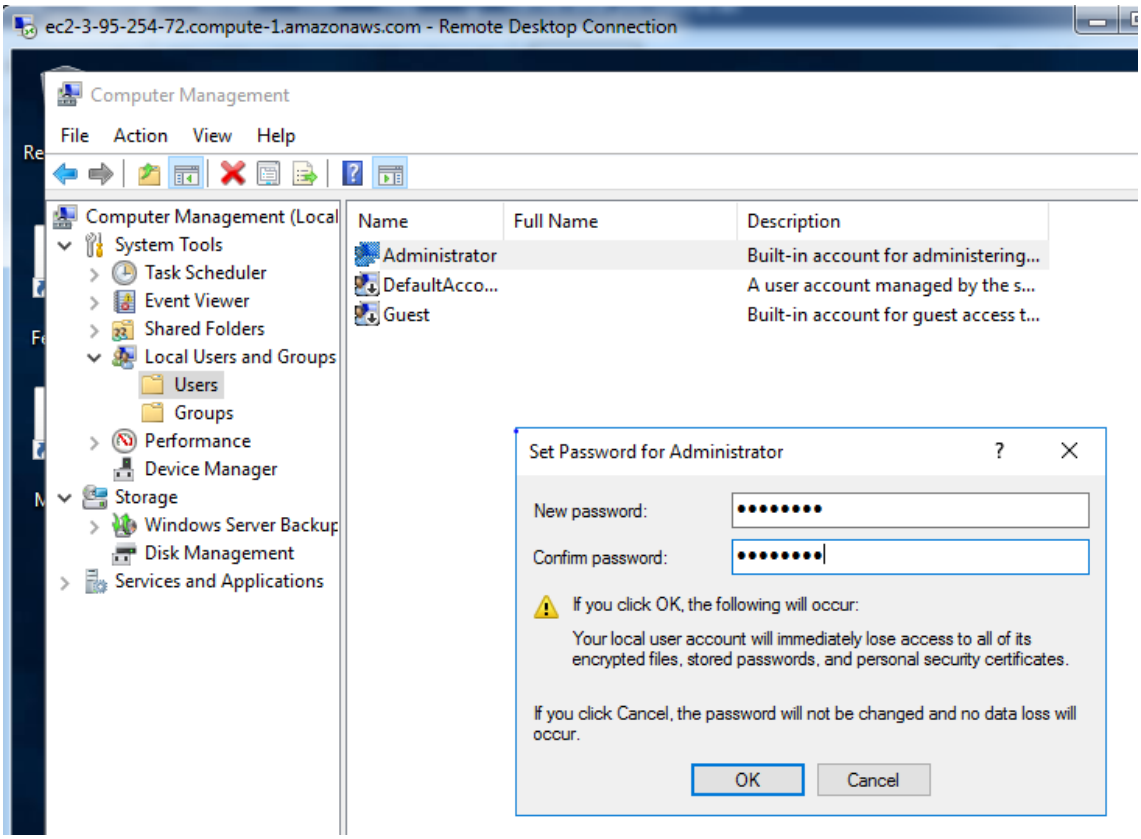
On your Windows EC2 Instance, click on Run and write compmgmt.msc.



This will open Computer Management Console. Click on Local Users and Groups and Click on Users. Right Click on Administrator User and click on Set Password.

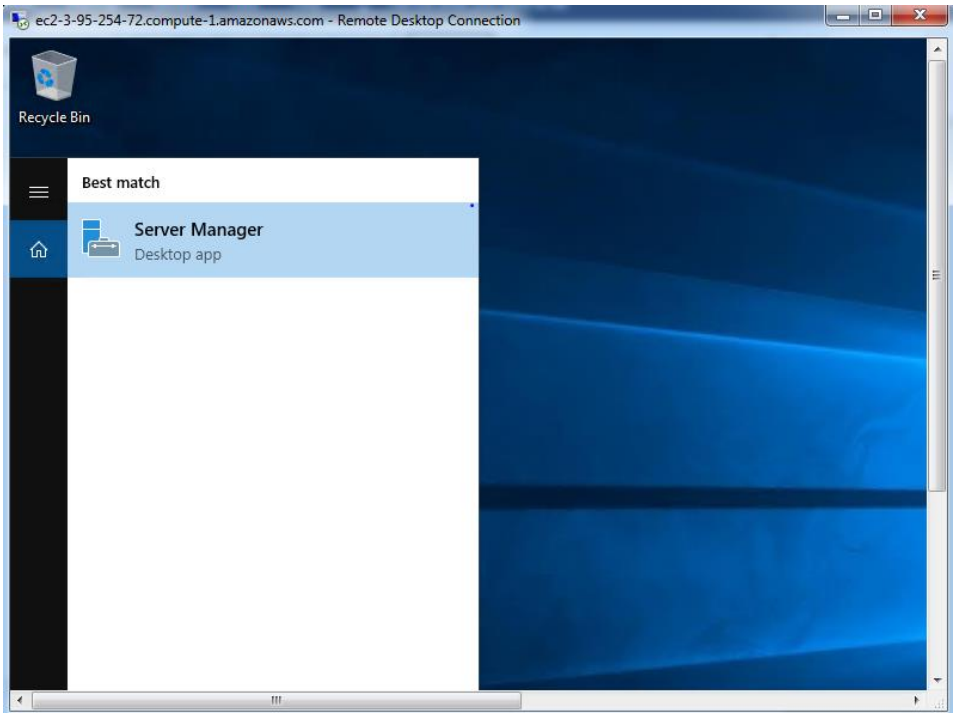


Give Password of your choice and hit Ok.

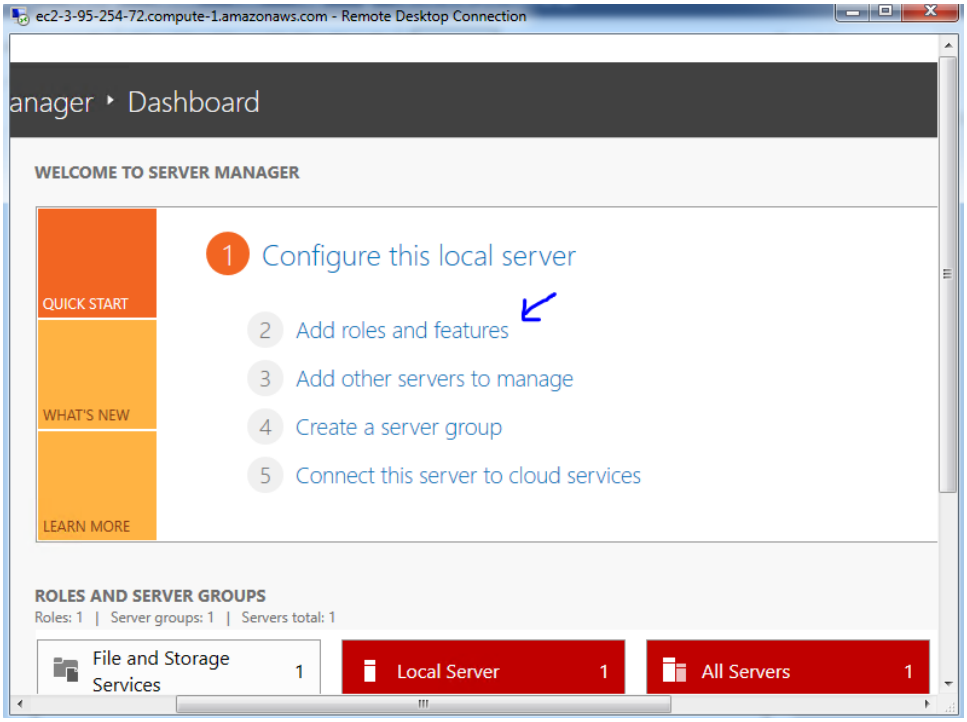


Now even if you lose the decrypted password, you do not have to worry as you've reset the Administrator's Account Password.

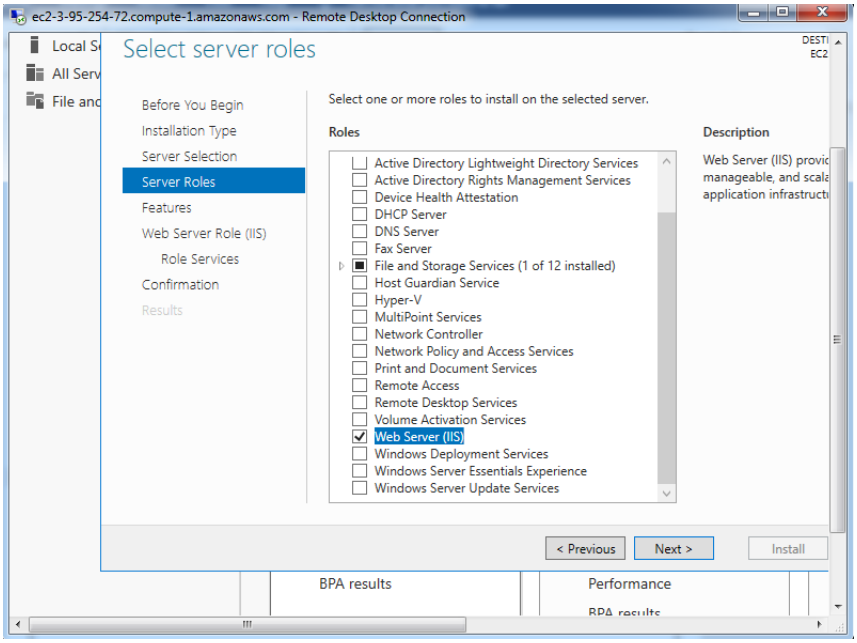
Now in EC2 Instance, Search for Server Manager.



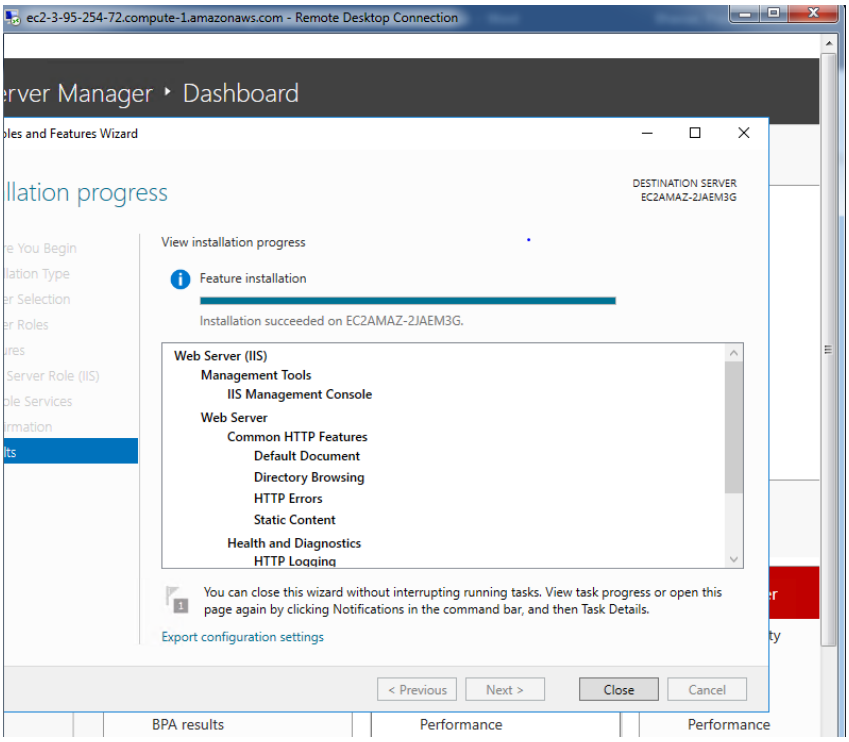
On the Server Manager Dashboard, Click on Add Roles and Features.



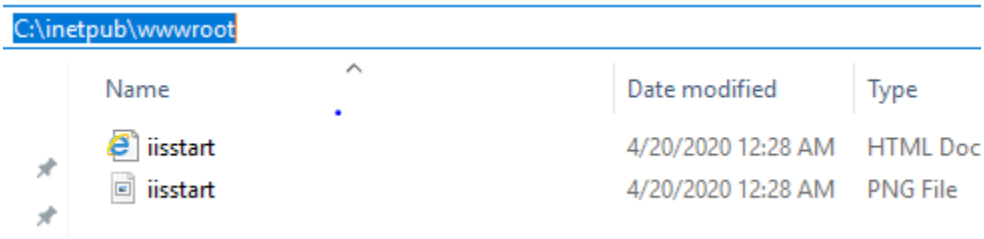
Keep all the default Settings for Before you Begin, Installation Type and Server Selection. Come to Server Roles. Click on Web Server (IIS) and Click on Add Features.



Keep default values for rest of the tasks and Start Installation. Once the Installation is complete, you'll observe the below status. Then click on Close.

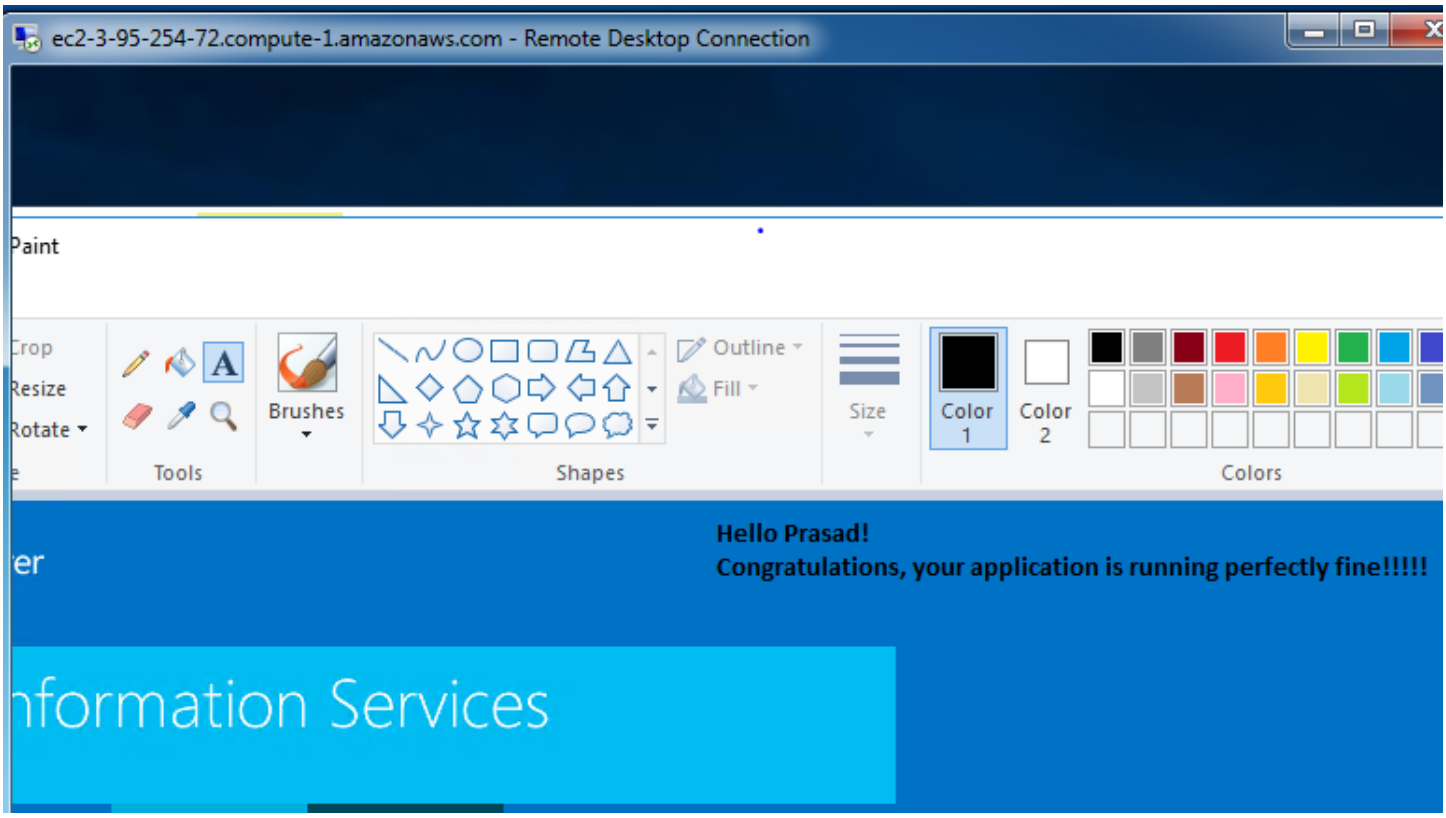


Once the Installation is Complete, navigate to the below directory in EC2 Instance.



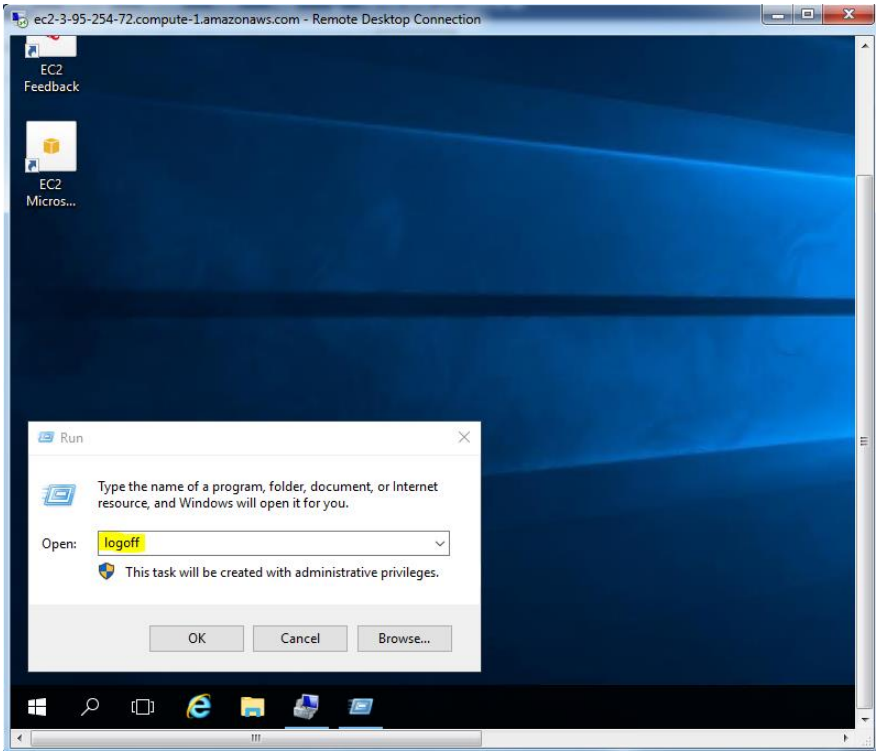
Since we do not have any Web Application running in our environment, we consider the default IIS webpage as the Web Application.

Now click on iistart.png and edit the image as per your choice.

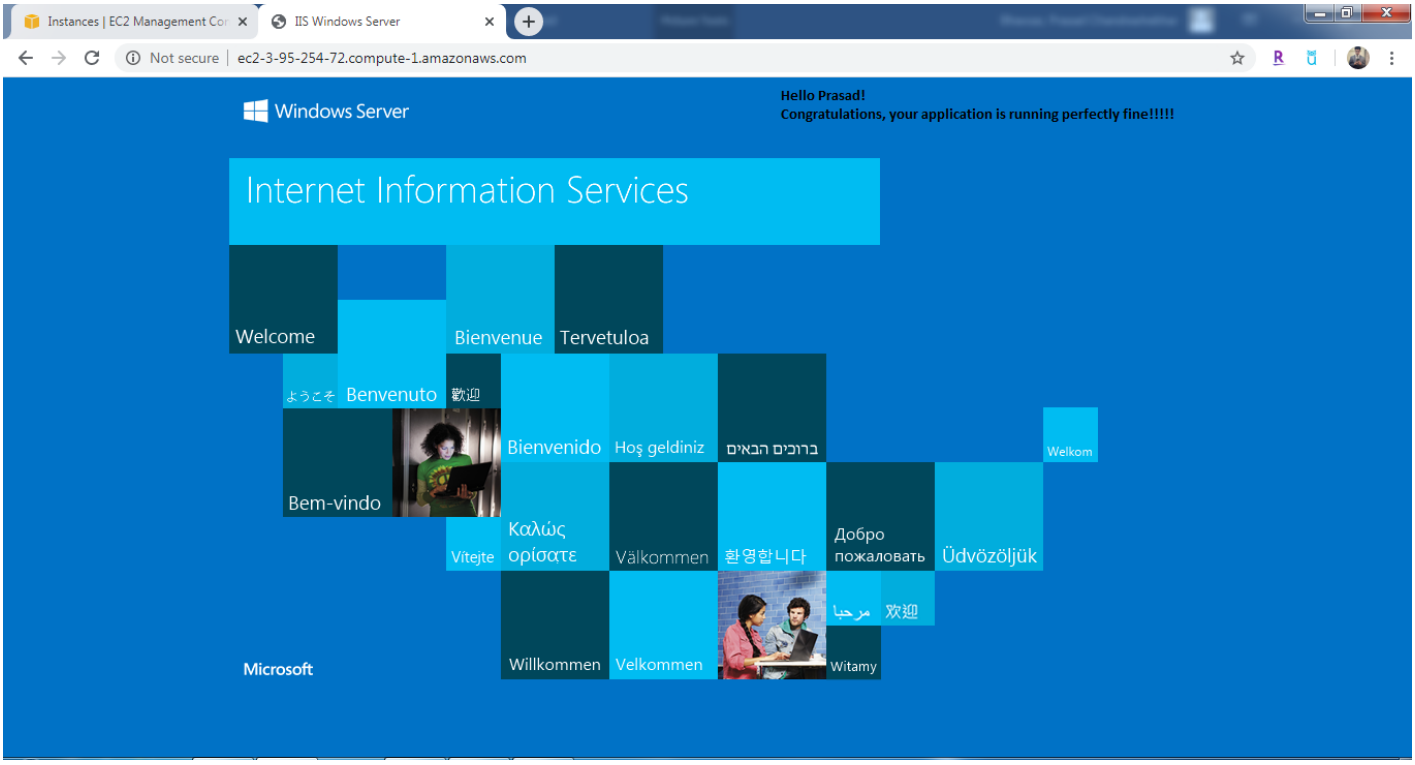


Web Server Installation and Configuration is now Complete.

Sign out from the EC2 Instance. <u>DO NOT SHUTDOWN THE EC2 INSTANCE.</u>

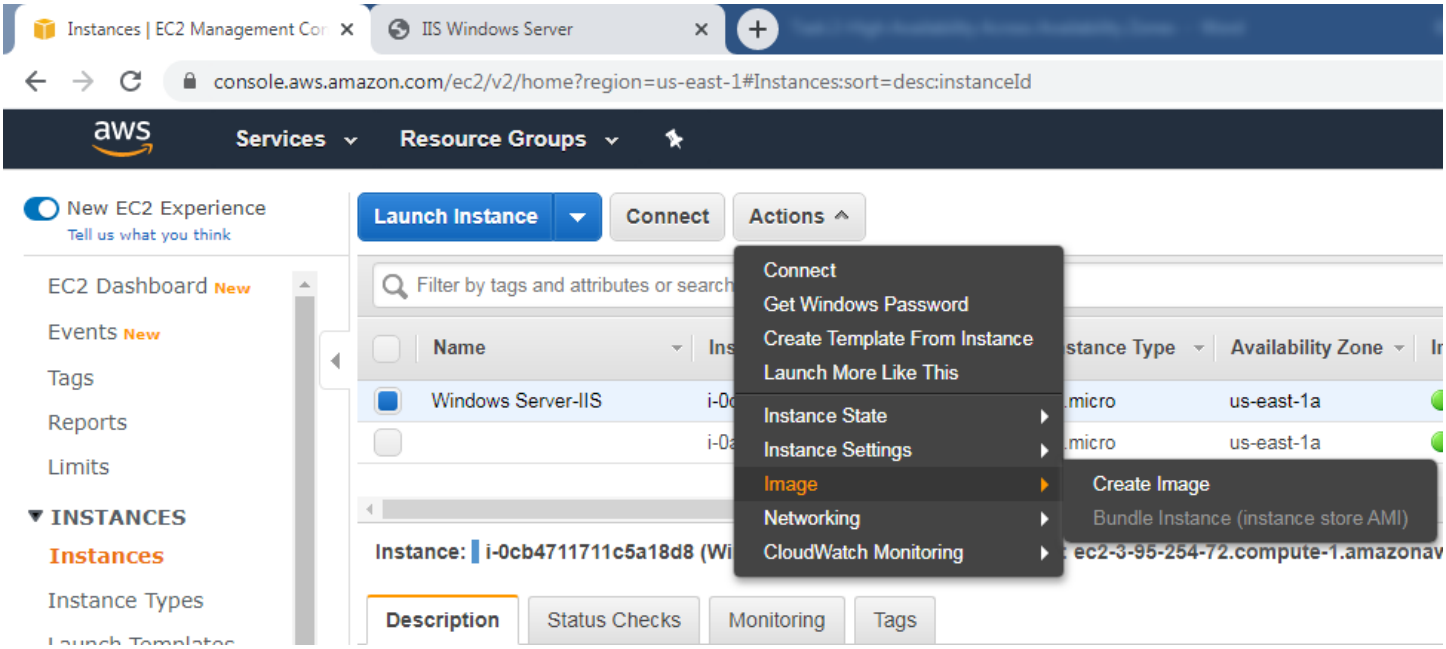


Come to your own Computer, open Browser and paste the EC2 Instance DNS or Public IP. You should see the Web Application web page that you have just configured.

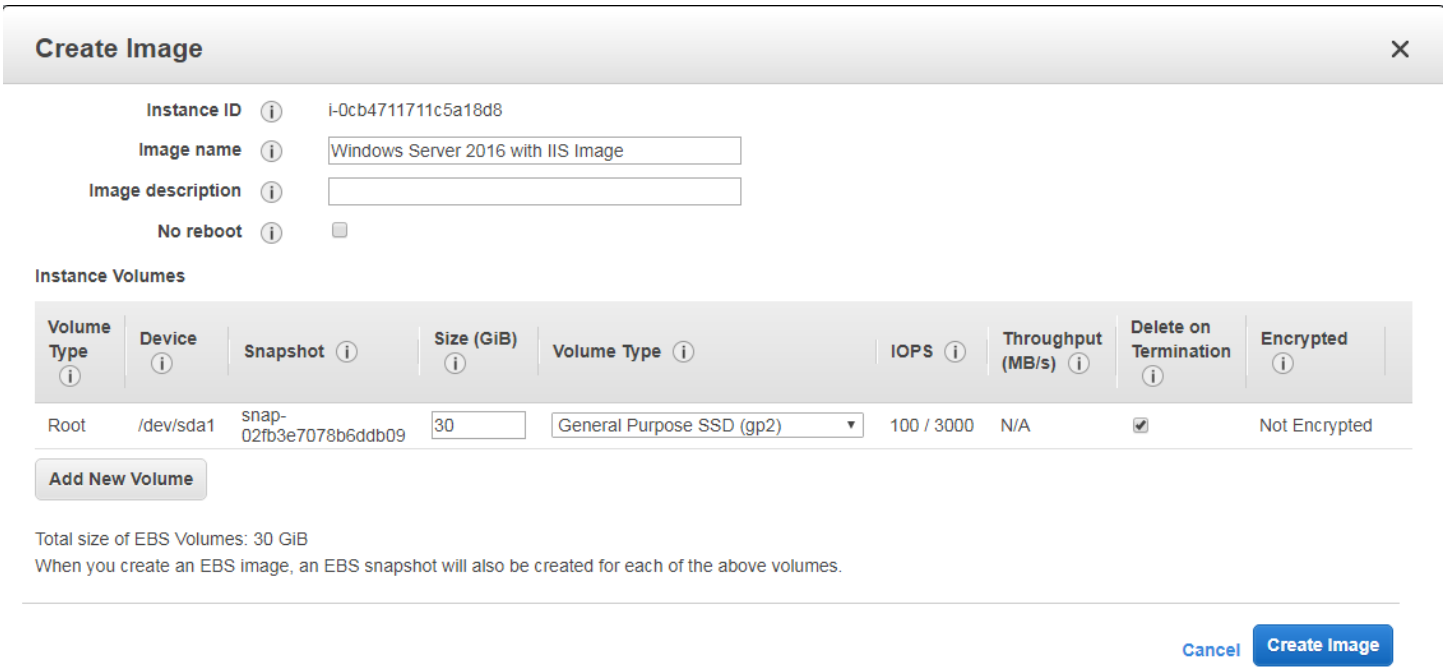


Task 4: Create an Amazon Machine Image (AMI) of the EC2 Instance.

Now come back to the EC2 Service. Select the EC2 Instance and click on Image and then Create Image.

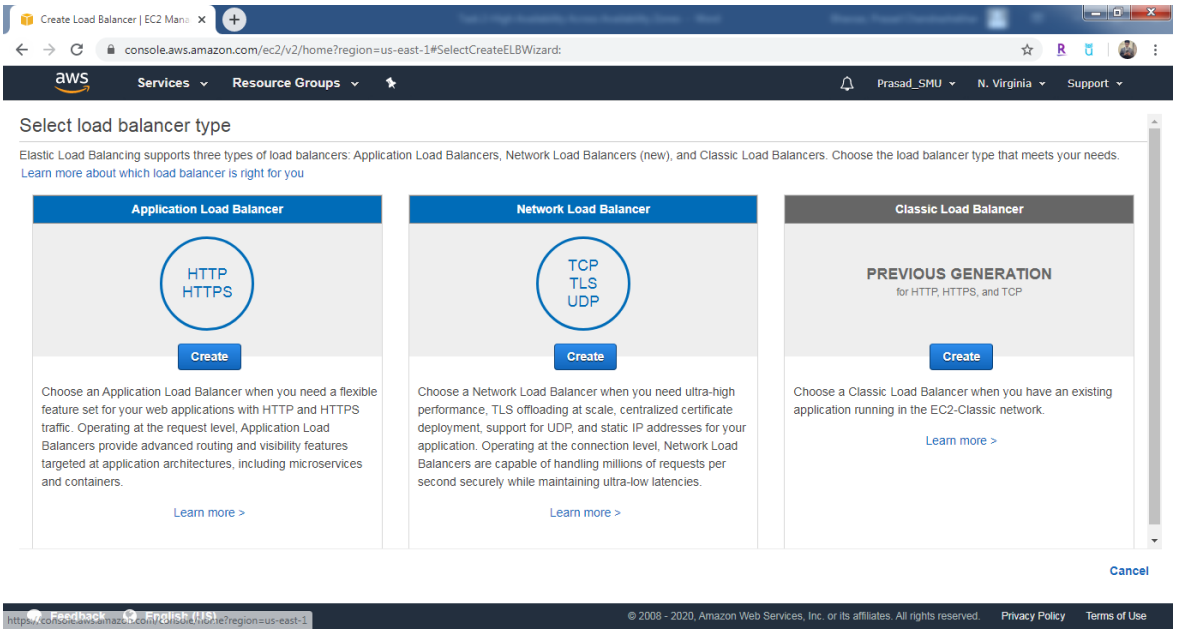


Give Image Name and click on Create Image.



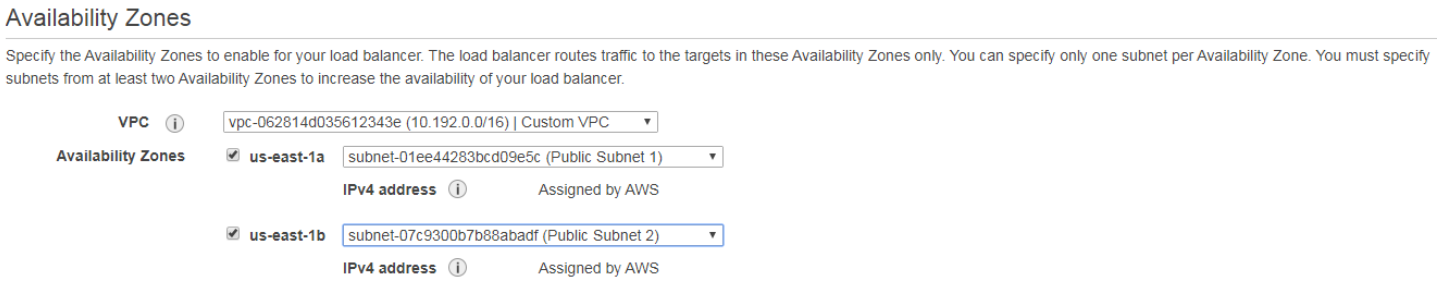
Task 5: Create an Internet-facing application load balancer.

Navigate to the EC2 Service and click on Load Balancer and Click on Create Application Load Balancer.



Give the Load Balancer name of your choice and select Scheme as internet-facing.

As per the best practice the Elastic Load Balancers (ELB) are configured on the Public Subnets while the target EC2 Instances are launched in Private Subnets. Hence select the Public Subnets from both the Availability Zones.



Click on Next: Configure Security Settings, since we are not going to use any SSL/TLS, keep this setting as default and click on Next: Configure Security Groups.

Click on Select an existing Security Group and select the Security Group (SG) that we've configured for the Load Balancer and click Next.

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group: Create a **new** security group

Select an **existing** security group

Filter VPC security groups

Security Group ID	Name	Description	Actions
sg-0138fa39c03c2ca04	default	default VPC security group	Copy to new
sg-0e97cfdf310093cb0	no-ingress-sg	Security group with no ingress rule	Copy to new
sg-0fa905f89ae8dfd9e	SG-Application Load Balancer	SG-Application Load Balancer	Copy to new
sg-04bd568afe8fa65c4	SG-Windows Servers	SG-Windows Servers	Copy to new

In Step 4: Configure Routing, we are now going to create a Target Group. Give the Target Name of your choice and configure the below Advanced Settings.

Target group

Target group New target group

Name TargetGroupofELB

Target type Instance / IP / Lambda function

Protocol HTTP

Port 80

Advanced health check settings

Port traffic port / override

Healthy threshold 2

Unhealthy threshold 2

Timeout 5 seconds

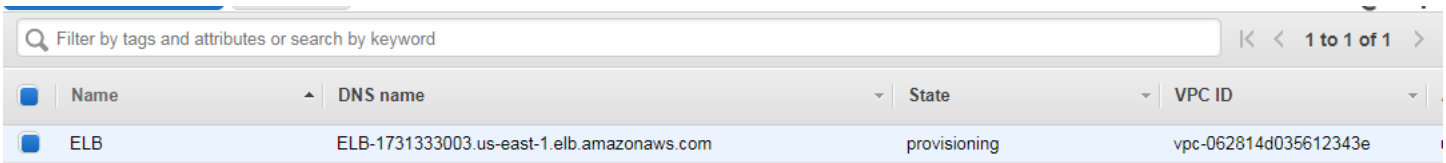
Interval 10 seconds

Success codes 200

Click on Next: Register Targets.

Here we are not going to register any Targets. Later we'll configure our architecture in such a way that the Targets of Load Balancer will be the EC2 Instances launched by AutoScaling group.

Click Next, review the configuration and click on Create.



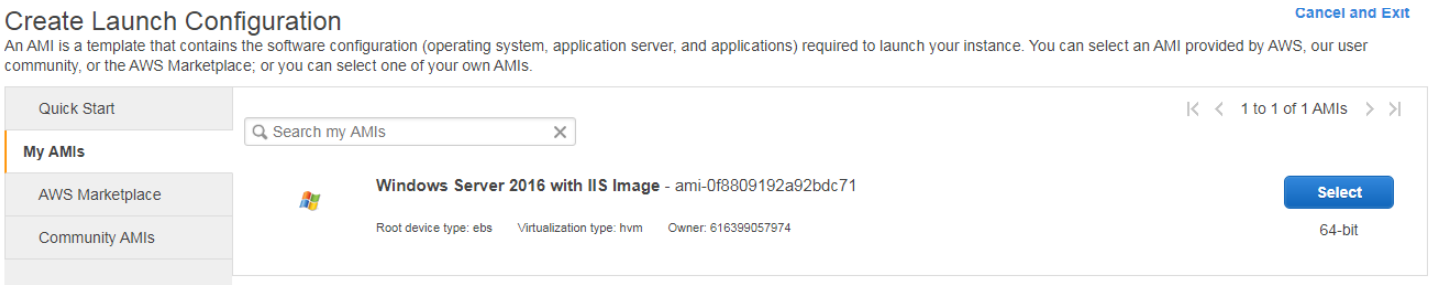
Your Elastic Load Balancer is now configured successfully!!!!!

Task 6: Create Launch Configurations, Auto Scaling groups (ASG), Auto Scaling Policies and CloudWatch Alarms.

Launch Configurations:

Navigate to the EC2 Service and click on Launch Configurations and then click on Create Launch Configurations.

Click on My AMIs, you'll notice the AMI that you've created. Click on Select.



Click on Next, keep the default Instance Type as t2.micro and click Next.

Give the Launch Configurations Name of your Choice and click Next.

Keep the default EBS Storage Settings and click on Next.

For the Security Group, click on Select an existing Security Group and click on Security Group that you've created for Windows Server.

Assign a security group: ○ Create a **new** security group
◉ Select an **existing** security group

	Security Group ID	Name	VPC ID	Description	Actions
☐	sg-0138fa39c03c2ca04	default	vpc-062814d035612343e	default VPC security group	Copy to new
☐	sg-431e9d10	default	vpc-a6c288dc	default VPC security group	Copy to new
☐	sg-0e97cfdf310093cb0	no-ingress-sg	vpc-062814d035612343e	Security group with no ingress rule	Copy to new
☐	sg-0fa905f89ae8dfd9e	SG-Application Load Balancer	vpc-062814d035612343e	SG-Application Load Balancer	Copy to new
☑	sg-04bd568afe8fa65c4	SG-Windows Servers	vpc-062814d035612343e	SG-Windows Servers	Copy to new

Click Next, review the Configurations and click on Create Launch Configurations. Select the existing Key Pair and click on Create Launch Configurations.

Select an existing key pair or create a new key pair ×

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair ▾

Select a key pair

Windows Servers Key Pair ▾

☑ I acknowledge that I have access to the selected private key file (Windows Servers Key Pair.pem), and that without this file, I won't be able to log into my instance.

Cancel **Create launch configuration**

Launch Configurations (LC) is now successfully created.

Launch configuration creation status

✔ **Successfully created launch configuration: LC**
View creation log

▾ View
View your launch configurations
View your Auto Scaling groups

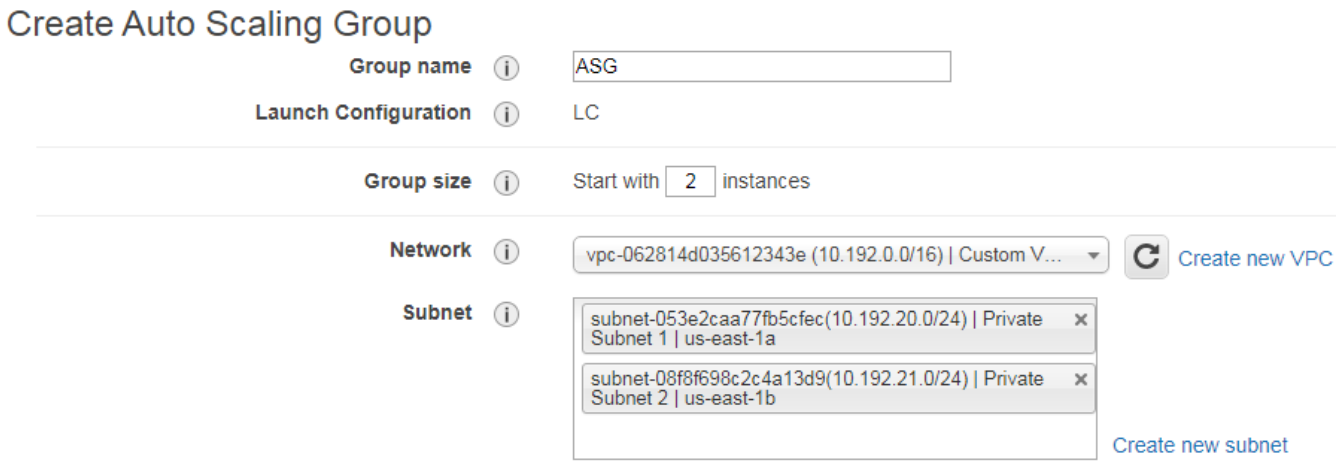
▸ Here are some helpful resources to get you started

Create an Auto Scaling group using this launch configuration **Close**

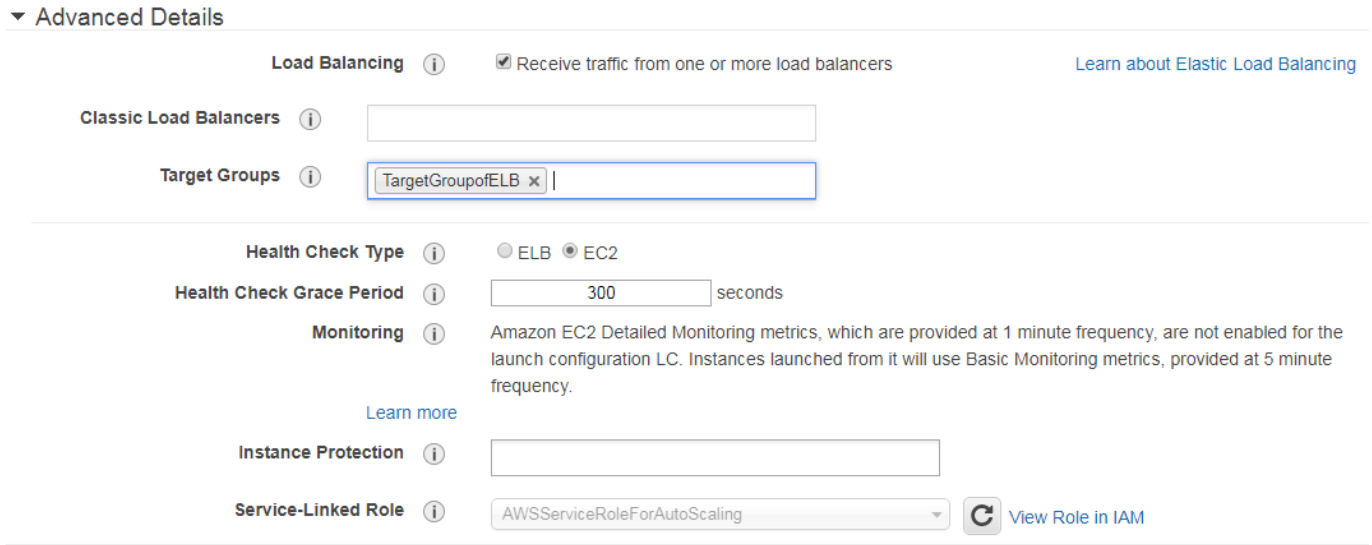
Auto Scaling groups (ASG), Auto Scaling Policies and CloudWatch Alarms:

Now click on Create an **Auto Scaling group using this launch configuration**.

Give Auto Scaling Group name as per your choice, Select Network as your Custom VPC and Subnets as Private Subnet 1 and Private Subnet 2. Also, keep the Group Size as 2. It means the Auto Scaling Group will launch two Windows Server 2016 IIS based EC2 Instances in two different Availability Zones. This is how we are achieving High Availability across multiple Availability Zones.



Now in the bottom, click on the Advanced Settings. Check **Receive traffic from one or more load balancers** and select the Target Group which you configured while configuring the Load Balancer. This means Load Balancer will route the traffic to the EC2 Instances launched in the Private Subnets.

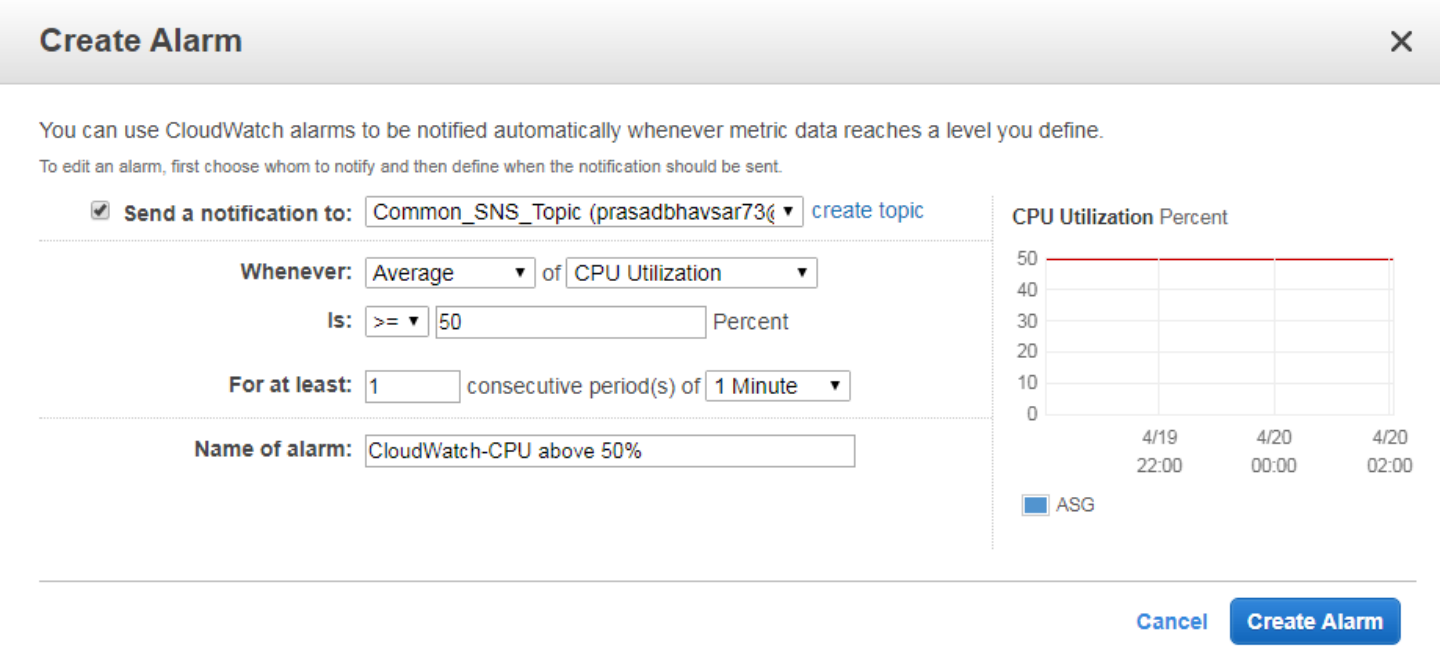


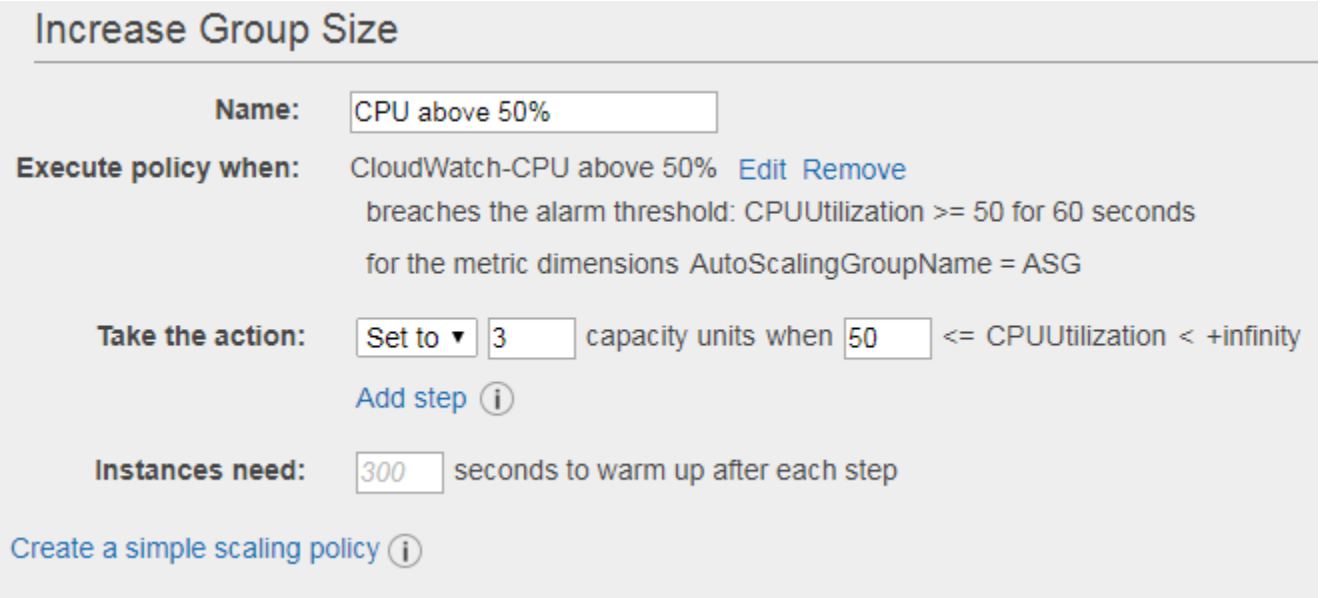
Click on Next: Configure the Scaling Policies. Select **Use scaling policies to adjust the capacity of this group**. At the bottom, click on **Scale the Auto Scaling group using step or simple scaling policies**. Then click on Add Alarms. Here you creating CLOUDWATCH alarms for your ASG policies.

Here you I'm keeping the scaling of Instances from 2-3.

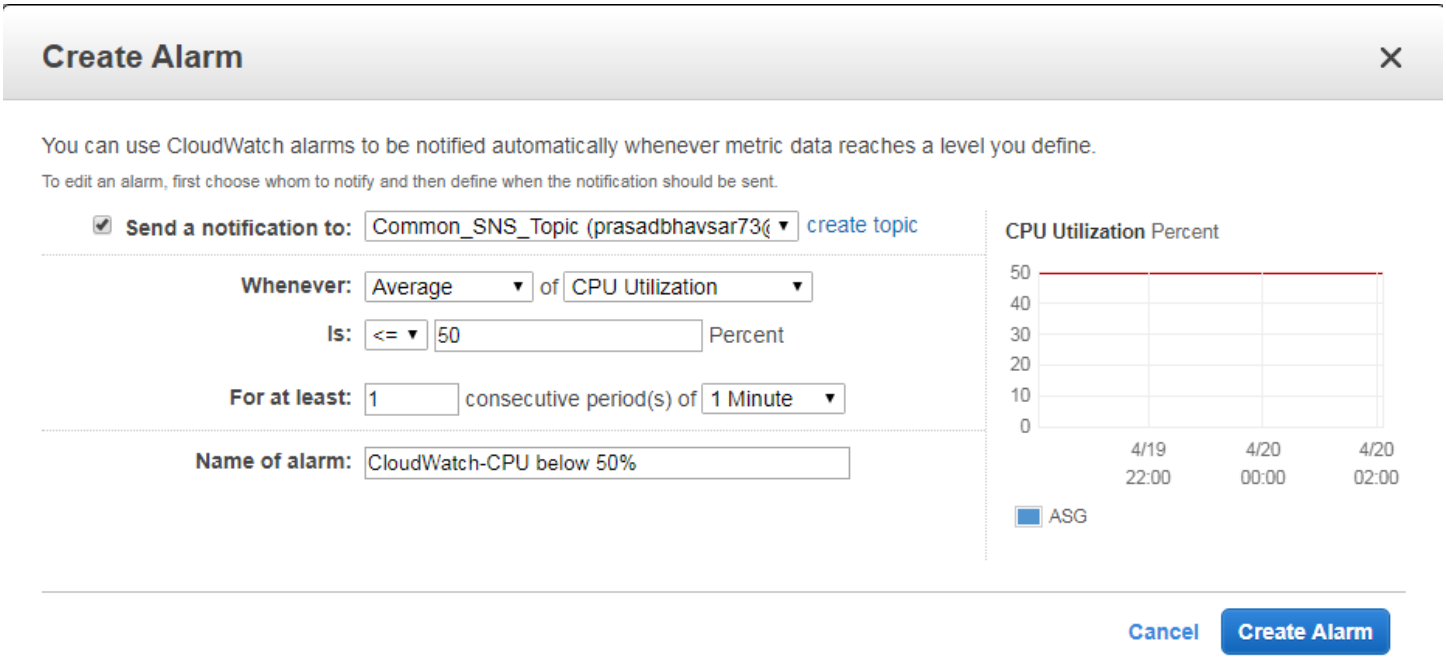
Scale between 2 and 3 instances. These will be the minimum and maximum size of your group.

Now, try to understand the following CloudWatch Alarms. When the CPU Utilization of EC2 Instances goes above 50 %, then a new EC2 Instance will get launched with total Capacity of 3 Instances and an SNS notification will be send to the SNS topic subscribers.





Similarly, when the CPU Utilization of EC2 Instances goes below 50 %, then a new EC2 Instance will get deducted with total Capacity of 2 Instances and an SNS notification will be send to the SNS topic subscribers.



Decrease Group Size

Name: CPU below 50%

Execute policy when: CloudWatch-CPU below 50% Edit Remove
breaches the alarm threshold: CPUUtilization <= 50 for 60 seconds
for the metric dimensions AutoScalingGroupName = ASG

Take the action: Set to 2 capacity units when 50 >= CPUUtilization > -infinity
Add step

Click on Next: Configure Notifications. CloudWatch is going to send notification In-case if any new Instance gets launched or terminated hence, we are not going to configure anything here.

Click Next: Tags, do not mention any Tags, Click Next, review the settings and Create ASG.

Auto Scaling group creation status

Successfully created Auto Scaling group
View creation log

View
View your Auto Scaling groups
View your launch configurations

Here are some helpful resources to get you started

Close

Autoscaling setup is now complete.

Let's now verify whatever we've configured.

First, navigate to the CloudWatch Service and verify the configured Alarms as follows. Do not worry if you see the state as Insufficient Data. ASG is still in process to deploy the EC2 Instances in Private Subnets.

Alarms (2) | Hide Auto Scaling alarms | Clear selection | Create composite alarm | Actions | Create alarm

Search | Insufficient d... | Any type | 1

Name	State	Last state update	Conditions	Actions
CloudWatch-CPU below 50%	Insufficient data	2020-04-19 09:17:46	CPUUtilization <= 50 for 1 datapoints within 1 minute	-
CloudWatch-CPU above 50%	Insufficient data	2020-04-19 09:17:46	CPUUtilization >= 50 for 1 datapoints within 1 minute	-

Wait for some time till the EC2 Instances Status Checks are Passed. You can now observe that two EC2 Instances have been launched by Auto Scaling in two different Availability Zones.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Server 1	i-0f0db1f1a7b08d704	t2.micro	us-east-1b	running	2/2 checks passed	None
Windows Server-IIS	i-0cb4711711c5a18d8	t2.micro	us-east-1a	running	2/2 checks passed	None
	i-07ba595434b420ba3	t2.micro	us-east-1a	terminated		None
	i-06d7d6670c23014d1	t2.micro	us-east-1b	terminated		None
Server 2	i-060878cf7930ebea6	t2.micro	us-east-1a	running	2/2 checks passed	None

If you click on Description of these Instances, you'll notice that no Public IPs have been assigned to these Instances. However, the Instances have Private IP Addresses.

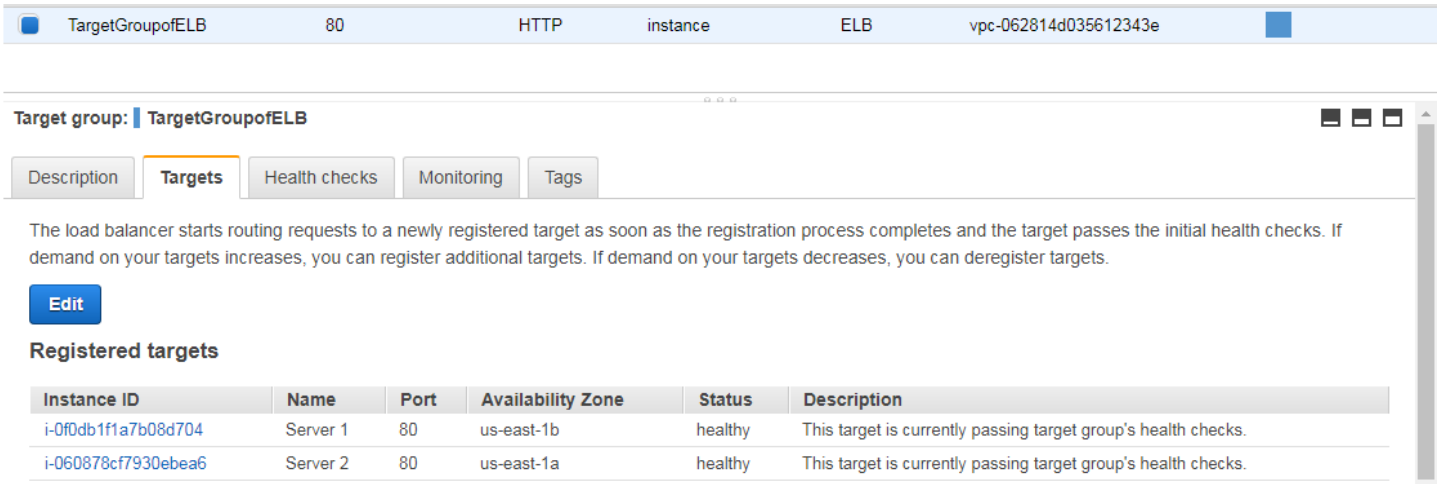
Think of the Reason??????

Description | Status Checks | Monitoring | Tags

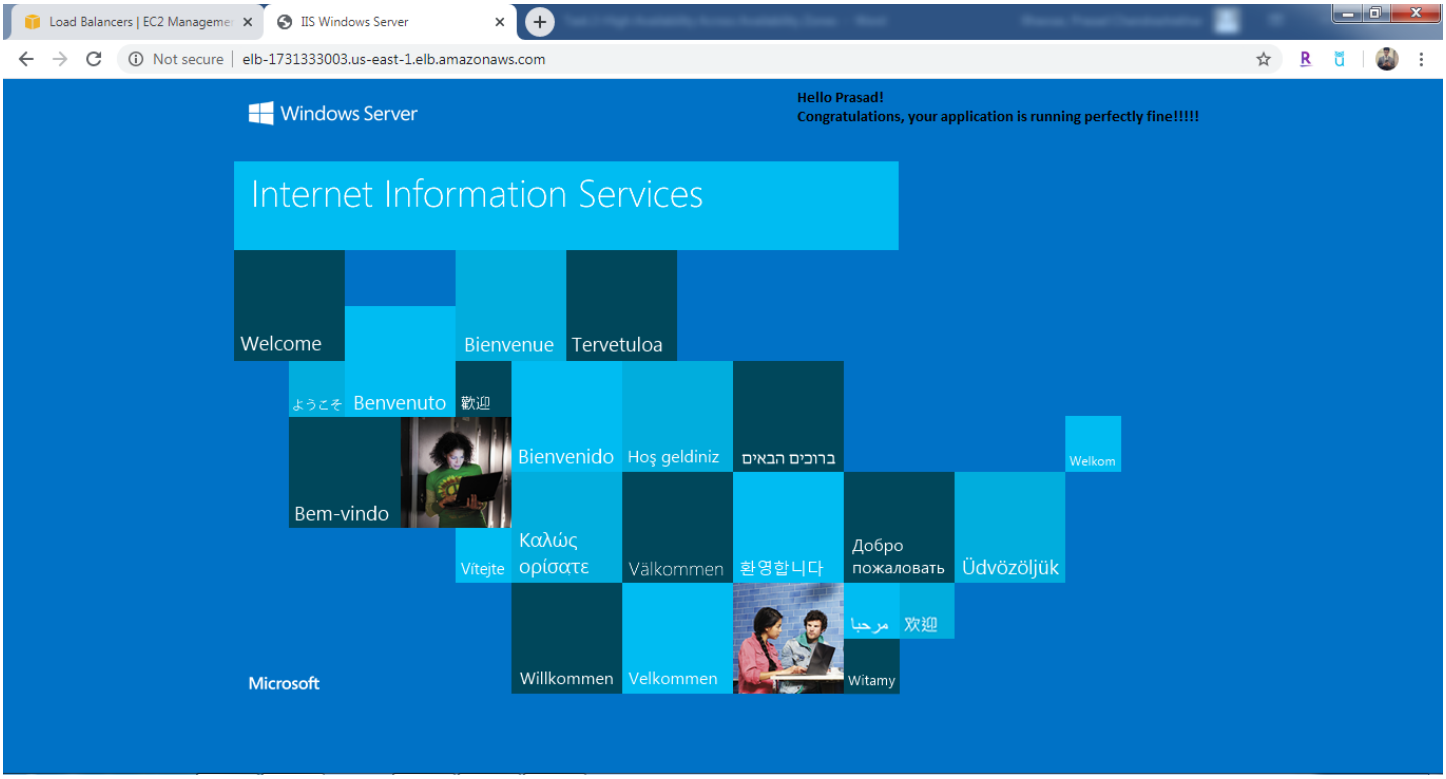
Instance ID	i-060878cf7930ebea6	Public DNS (IPv4)	- ?
Instance state	running	IPv4 Public IP	-
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-10-192-20-106.ec2.internal	Availability zone	us-east-1a
Private IPs	10.192.20.106	Security groups	SG-Windows Servers. view inbound rules. view outbound rules

Answer: Because Autoscaling has launched these EC2 Instances in PRIVATE SUBNETS.

Also, under Load Balancing, click on Target Groups. You'll observe that EC2 Instances launched by AutoScaling will be added under Targets.



Now finally, navigate to Load Balancer and copy the Public DNS of the Load Balancer and paste it in your Browser. You should see your Application is Up and Running and it was served by a EC2 Instance launched by Autoscaling.



Task 7: Test the Failover.

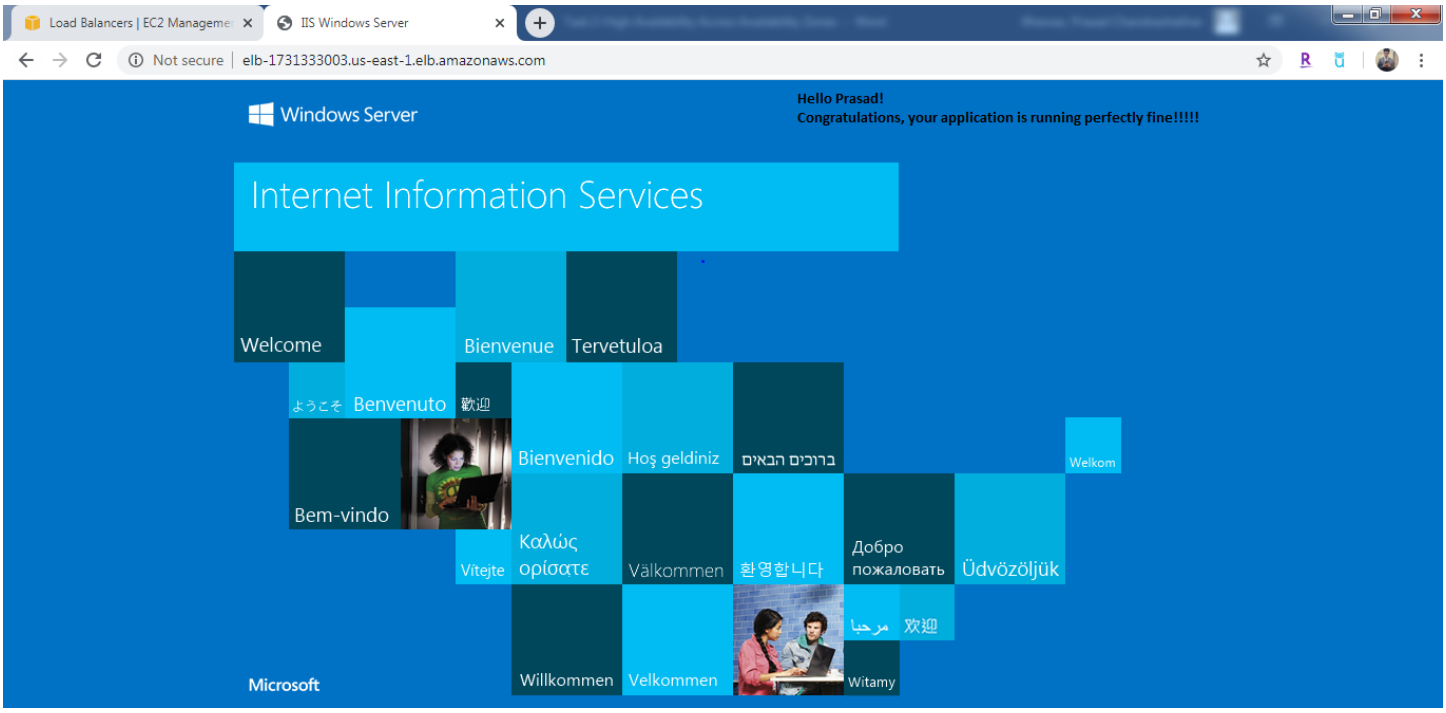
Navigate to EC2 Instances and TERMINATE on of the EC2 Instance which was launched by Autoscaling group.

Let's Terminate Server 1.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Server 1	i-0f0db1f1a7b08d704	t2.micro	us-east-1b	shutting-do...		None

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Server 1	i-0f0db1f1a7b08d704	t2.micro	us-east-1b	terminated		None

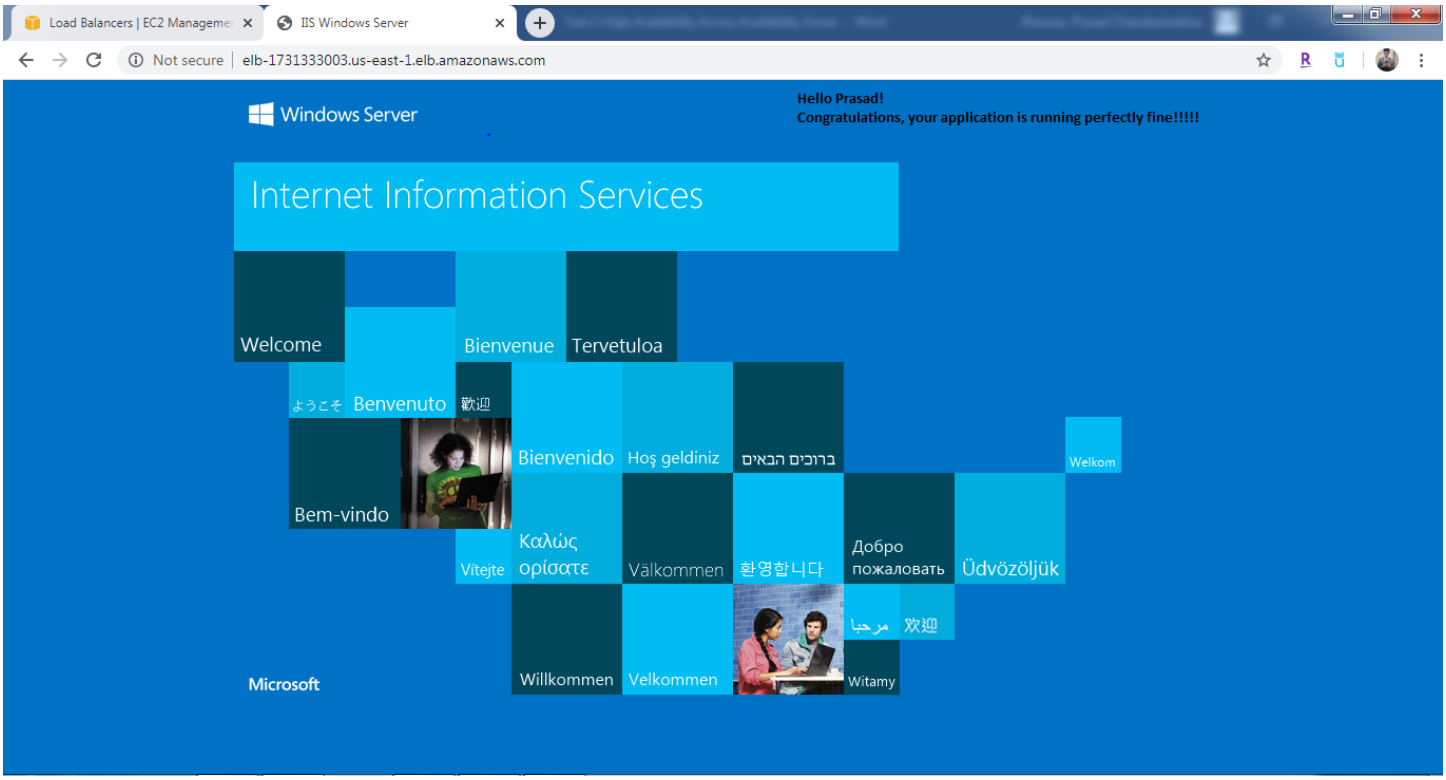
Once the Server 1 is fully terminated, go back to Load Balancer, copy the DNS name and paste it in the browser. You should see your application is still Up and Running and it was served by Server 2 which is in different Availability Zone.



You'll notice that after termination of Server 1, Autoscaling group has launched a new Instance (Server 3) in different Availability Zone than Server 2.

Server 3	i-01043fe380368098b	t2.micro	us-east-1b	pending	Initializing	None
Server 3	i-01043fe380368098b	t2.micro	us-east-1b	running	2/2 checks passed	None

Now your application is served by Server 2 and Server 3 which is in different Availability Zones.



This is how you can achieve the High Availability across multiple Availability Zones along with Autoscaling of EC2 Instances depending on the resource's utilization.

For Questions, contact me on pbhavsar@smu.edu.