

1. Part 1: Deployment of webpage on EC2 instance (Experience Infrastructure as a Service by deploying your personal profile web page using AWS EC2 and AWS S3)

Step 1: First I have created EC2(Amazon Linux AMI 2016.09.1 (HVM), SSD Volume Type - ami-f173cc91)

Step2: I choose t2.micro.

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
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
General purpose	m4.large	2	8	EBS only	Yes	Moderate	Yes
General purpose	m4.xlarge	4	16	EBS only	Yes	High	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Step 3: I kept everything as default

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

Purchasing option: Request Spot instances

Network: vpc-547dfa33 (default)

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

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: None

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.

Step 4: I added below tag , this is also not mandatory

Step 5: Add Tags

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)
Assignment First	Aws Instance setup <input type="button" value="X"/>

(Up to 50 tags maximum)

Step 5: After Review we can select launch option

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 2016.09.1 (HVM), SSD Volume Type - ami-f173cc91

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

Edit AMI

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

Edit instance type

Security Groups

Security Group ID	Name	Description
sg-634f2f1b	MywebServer	MywebServer

Edit security groups

Cancel **Previous** **Launch**

Step 8: I have created a key pair to login this EC2 instance via cli named Myec2KeyPair, earlier I had generated this file and downloaded this .pem file.

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 2016.09.1 (HVM)

Free tier eligible The Amazon Linux AMI is an EBS-backed, PostgreSQL, and other packages.
Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs
t2.micro	Variable	1

Security Groups

Security Group ID	Name	Description
sg-634f2f1b	MywebServer	MywebServer

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
Myec2KeyPair
 I acknowledge that I have access to the selected private key file (Myec2KeyPair.pem), and that without this file, I won't be able to log into my instance.

Cancel **Launch Instances**

Edit AMI

Edit instance type

Network Performance

Edit security groups

Cancel **Previous** **Launch**

Step 9: Below steps we need to do host our website on EC2:

- a) Update our instance so that it will remove all other dependency.
- b) Install apache server
- c) Now add our html file to server

First Login using below command

```
ssh ec2-user@54.202.236.226 -i MyEC2KeyPair.pem privilege
```

Sudo su to gain root access and gain

Use below command to update:

```
Yum update -y
```

Install Apache:

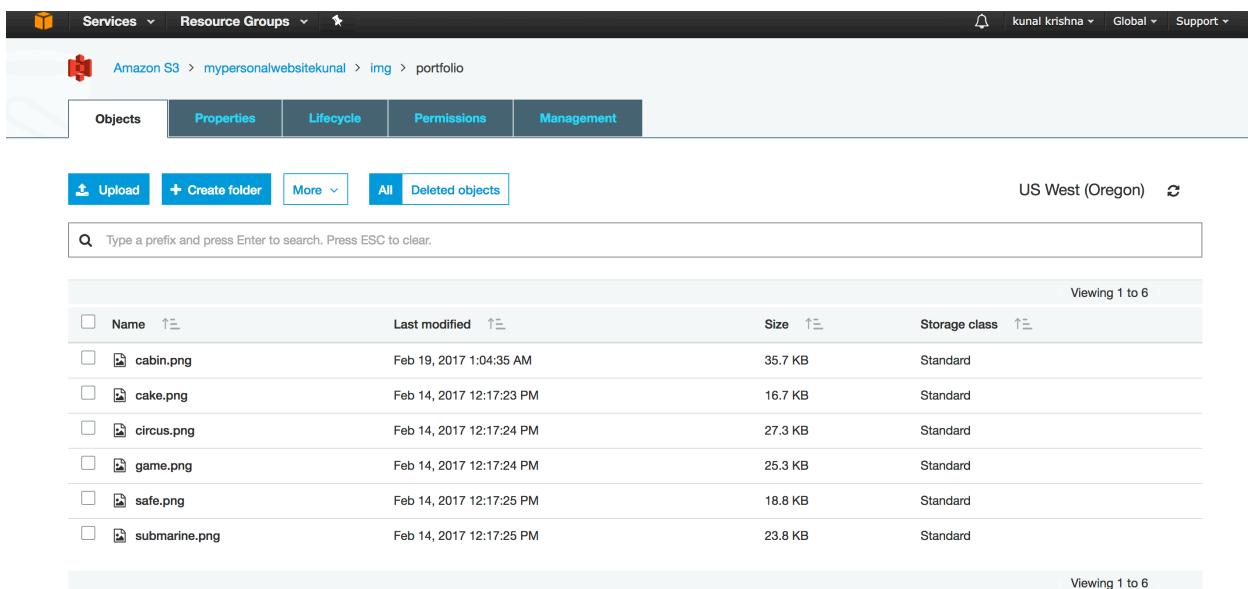
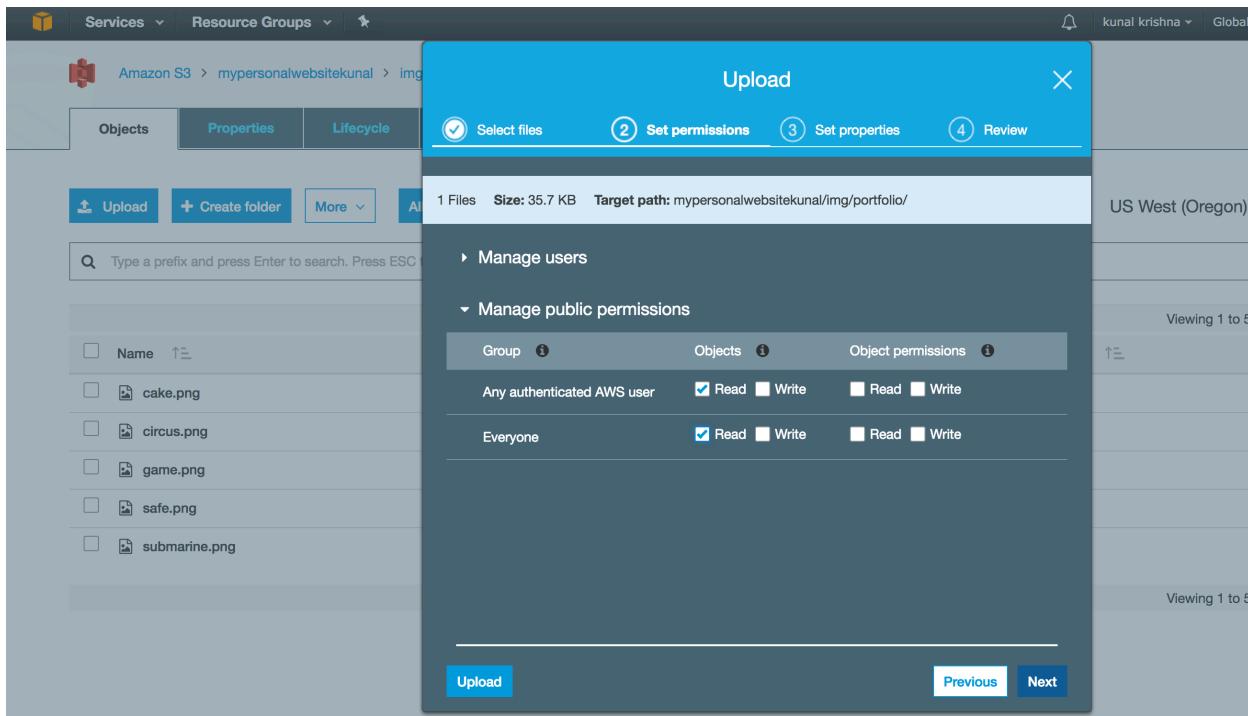
```
yum install httpd -y
```

Now I use command cd /var/www/html so that I can put my index.html folder in this location.

Below steps required to create S3 Bucket and uploaded images and other important files to support my html.index file.

Below steps to upload files:

Created bucket name mypersonalwebsitekunal. It should be unique



The screenshot shows the AWS S3 console interface. At the top, there are navigation links for Services, Resource Groups, and a user profile for 'kunal krishna'. Below the header, the path 'Amazon S3 > mypersonalwebsitekunal' is displayed. A navigation bar at the top of the main content area includes tabs for Objects, Properties, Lifecycle, Permissions, and Management, with 'Objects' being the active tab. Below the navigation bar are buttons for Upload, Create folder, More, All, and Deleted objects. To the right of these buttons, it says 'US West (Oregon)' and has a refresh icon. A search bar below the buttons contains the placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' A table below the search bar lists the objects in the bucket. The table has columns for Name, Last modified, Size, and Storage class. The objects listed are:

Name	Last modified	Size	Storage class
css	--	--	--
img	--	--	--
js	--	--	--
less	--	--	--
mail	--	--	--
vendor	--	--	--
index.html	Feb 14, 2017 4:17:42 PM	29.2 KB	Standard

At the bottom of the table, it says 'Viewing 1 to 7'.

Submission Part for Part 1:

Web Page link:

<http://ec2-54-202-121-2.us-west-2.compute.amazonaws.com/>

or

<http://54.202.121.2/>

Note(Above link will change if I stop and start my instance)

Below are S3 links of files (Images and some important files related to website) that I have used.

Images:

Profile pic:

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/img/profile.jpeg>

Resume:

https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/-Kunal_Krishna.pdf

Other images:

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/img/portfolio/cabin.png>

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/img/portfolio/cake.png>

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/img/portfolio/circus.png>

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/img/portfolio/game.png>

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/img/portfolio/safe.png>
<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/img/portfolio/submarine.png>

Problems I faced and solutions:

If I update any file on S3 it immediately not updates my website it takes some time to update.

Another problem which I faced was, when I stopped my Instance and re-start my instance then my public IP address gets changed so solution for this is to I need to keep my instance in running state if I do not want to change IP address.

2. Part 2: Experience scalability

Client Server Program:

I am using Matrix multiplication program using java socket programming method I connected my local host client program to server program using public ip address and port number 6769.

First I ran server program on EC2 server.

After installing java on ec2 server, below commands to run the java program.

```
SSH — root@ip-172-31-2-240:/home/ec2-user — ssh ec2-user@54.202.236....  
Transaction Summary  
=====  
Install 1 Package  
  
Total download size: 11 M  
Installed size: 36 M  
[Is this ok [y/d/N]: y  
Downloading packages:  
java-1.7.0-openjdk-devel-1.7.0.131-2.6.9.0.71.amzn1.x86_ | 11 MB 00:00  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
Installing : 1:java-1.7.0-openjdk-devel-1.7.0.131-2.6.9.0.71.amzn1.x86_ 1/1  
Verifying : 1:java-1.7.0-openjdk-devel-1.7.0.131-2.6.9.0.71.amzn1.x86_ 1/1  
  
Installed:  
java-1.7.0-openjdk-devel.x86_64 1:1.7.0.131-2.6.9.0.71.amzn1  
  
Complete!  
[[root@ip-172-31-2-240 ec2-user]# vi Server.java  
[[root@ip-172-31-2-240 ec2-user]# javac Server.java  
[[root@ip-172-31-2-240 ec2-user]#
```

Client side input:

After running server program, I ran client program on my local system.

First I took IP address of server:

Second I took number of rows and columns for matrix A and B. So, that I can multiply A and B and produce output for Matrix C.

I used various number to experiment this.

Finally I used : 4000 rows and columns for both the matrix and I got more than 70 percentage utilization and element of each row and column for first matrix is 999 and for second matrix 999.

For more details about client and server program you can see at below links:

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/Client.java>

<https://s3-us-west-2.amazonaws.com/mypersonalwebsitekunal/Server.java>

The screenshot shows the IntelliJ IDEA interface with the ClientServer project open. The Client.java file is the active editor, displaying the main() method which handles communication with a server via sockets. The terminal window below shows the client program being run, requesting the IP address of the server (54.202.236.226) and specifying matrix dimensions (1000x1000). The client then waits for input from the server, indicated by the red vertical line in the terminal window.

```

Client / src / Client /
Project  Server.java x Client.java x ReadBusiness.java x Cli.java x cli2.java x Cli3.java x Ser.java x
ClientServer ~/Downloads/ClientServer
src
  Cli
  cli2
  Cli3
  Client
  ReadBusiness
  Ser
  Server
  ClientServer.iml
External Libraries

Client.java
public class Client {
    public static void main() {
        // Client logic
    }
}

Run: Client Client
Enter IP Server :
54.202.236.226
Input Rows Matrix A :
1000
Input Cols Matrix A :
1000
Input Rows Matrix B :
1000
Input Cols Matrix B :
1000

```

Below steps we need

a) Created a new instance.

b) Created auto scaling group and created a rule, alarm for that

c) install java:

yum install java-devel

Run client server program.

Below Screen shots explained the step by step to create auto scaling group.

Step First: Give group name and select all available subnet, for example if you want to have maximum 3 instance then select 3 subnets so that once you instances are created each instance get a particular subnet.

Step 2: Add new Alarm in configure scaling policies, in this case I have added rule in which new instance will be created once CPU utilization reach more than or equal to 70 percentage. I have given scale range from 1 to 3 . This is for scaling up.

Create Auto Scaling Group

Keep this group at its initial size
 Use scaling policies to adjust the capacity of this group

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

Increase Group Size

Name: Increase Group Size

Execute policy when: CPU Utilization Kunal

breaches the alarm threshold: CPUUtilization >= 70 for 300 seconds for the metric dimensions Instanceld = i-08ec19378e3ec65a9

Take the action: Add instances when <= CPUUtilization < +infinity

Instances need: 100 seconds to warm up after each step

Create a simple scaling policy

Decrease Group Size

Set alarm for Scaling Down: I have given rule if CPU utilization is less than 50 percentage and consecutive time period, I have selected as 1 minutes and given a name of the alarm.

Create Auto Scaling Group

Name: Increase Group Size

Execute policy when: CPU Utilization Kunal

breaches the alarm threshold: CPUUtilization >= 70 for 300 seconds

Take the action:

Instances need:

Create a simple scaling policy

Decrease Group Size

Name: Decrease Instance for scaling Down

Execute policy when:

Take the action:

Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define. To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send a notification to: MyCPUUtilization (kxk155230@utdallas)

Whenever: Average of CPU Utilization Is: < 50 Percent

For at least: 1 consecutive period(s) of 1 Minute

Name of alarm: Decrease Instance for scaling Down

CPU Utilization Percent

50
40
30
20
10
0

2/19 04:00 2/19 06:00 2/19 08:00

sdfasdfs

Decrease Group Size: Now after setting alarm set the rules that remove one instances when CPU utilization is less than 50 percentage.

Decrease Group Size

Name: Decrease Group Size

Execute policy when: Decrease Instance for scaling Down [Edit](#) [Remove](#)
breaches the alarm threshold: CPUUtilization < 50 for 60 seconds
for the metric dimensions AutoScalingGroupName = sdfasdf

Take the action: [Remove](#) 1 Instances when 50 >= CPUUtilization > -infinity

[Add step](#)

[Create a simple scaling policy](#)

[Cancel](#) [Previous](#) [Review](#) [Next: Configure Notifications](#)

Configure notification: We can add notification we our CPU utilization increases

Services ▾ Resource Groups ▾ [Add](#)

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

Create Auto Scaling Group
Configure your Auto Scaling group to send notifications to a specified endpoint, such as an email address, whenever a specified event takes place, including: successful launch of an instance, failed instance launch, instance termination, and failed instance termination.

If you created a new topic, check your email for a confirmation message and click the included link to confirm your subscription. Notifications can only be sent to confirmed addresses.

[Add notification](#)

[Cancel](#) [Previous](#) [Review](#) [Next: Configure Tags](#)

Configure Tag: I have just added below tag , though it is not mandatory.

Create Auto Scaling Group

A tag consists of a case sensitive key-value pair that you can use to identify your group. For example, you could define a tag with Key = Environment and Value = Production. You can optionally choose to apply these tags to instances in the group when they launch. [Learn more](#).

Key	Value	Tag New Instances
CheckScalingLab	incase of cpu utilization greater than 70	<input checked="" type="checkbox"/>
checkScalingLab	incase of cpu utilization is less than 50	<input checked="" type="checkbox"/>

Add tag 48 remaining

Cancel Previous Review

Review: In this section it is mentioned that minimum group size is one instance and Maximum group size is 3 and health check grace period is 150 seconds. Once we click on create auto scaling group we will create a auto scaling group.

Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

Auto Scaling Group Details

Group name	Assingment1AutoscalingGroup
Group size	1
Minimum Group Size	1
Maximum Group Size	3
Subnet(s)	subnet-0409d363, subnet-d5dd219c, subnet-3be8cc63
Health Check Grace Period	150
Detailed Monitoring	No
Instance Protection	None

Scaling Policies

- Increase Group Size: With alarm = awsec2-Assingment1AutoscalingGroup-CPU-Utilization; Add 3 instances and 100 seconds for instances to warm up
- Decrease Group Size: With alarm = awsec2-Assingment1AutoscalingGroup-High-CPU-Utilization; Remove 1 instances

Notifications

Tags

CheckScalingLab	incase of cpu utilization greater than 70	tag new instances
checkScalingLab	incase of cpu utilization is less than 50	tag new instances

Cancel Previous Create Auto Scaling group

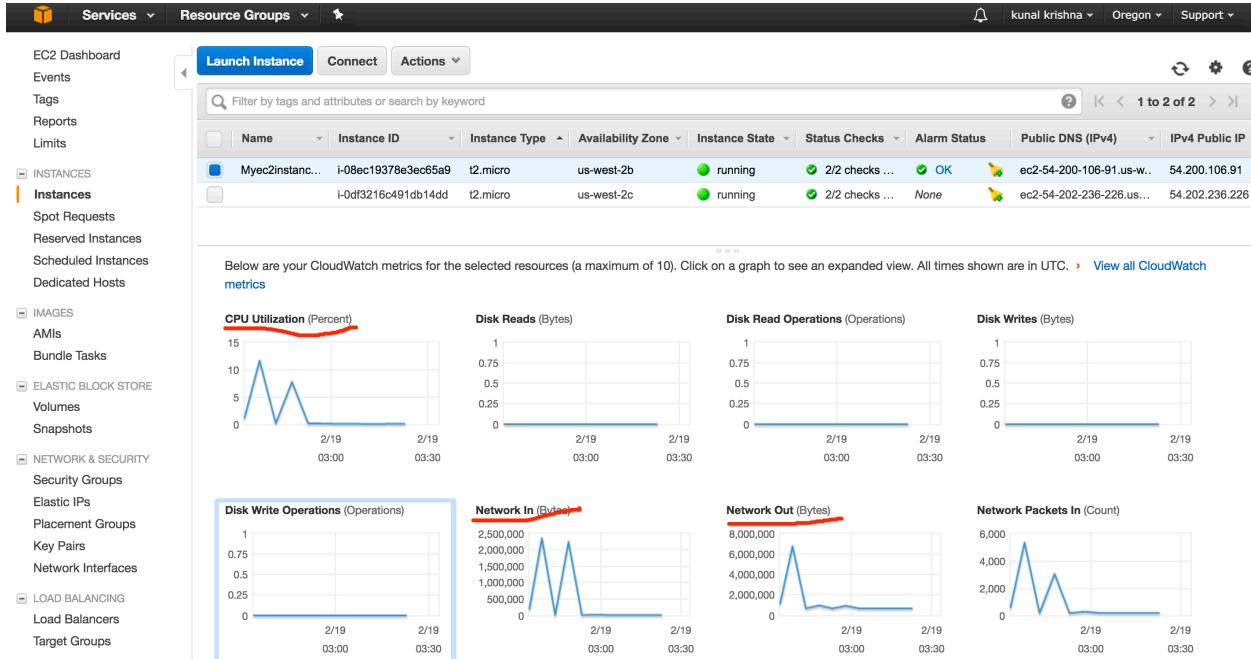
Now once our Auto Scaling group is created now we can add our existing instance in this group using below methods: Select instance and then action ->instance setting-> attach to auto scaling group. My case it is my Myec2Instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The 'Instances' link is underlined. In the main content area, a table lists two instances: 'Myec2instanc...' and 'i-0df3216c491db14dd'. Below the table, detailed information for 'Myec2instanc...' is shown, including Instance ID (i-08ec19378e3ec65a9), Instance state (running), Instance type (t2.micro), Availability zone (us-west-2b), and Security groups (MywebServer). A context menu is open over the first instance, with 'Actions' expanded. The 'Attach to Auto Scaling Group' option is visible and highlighted with a red box.

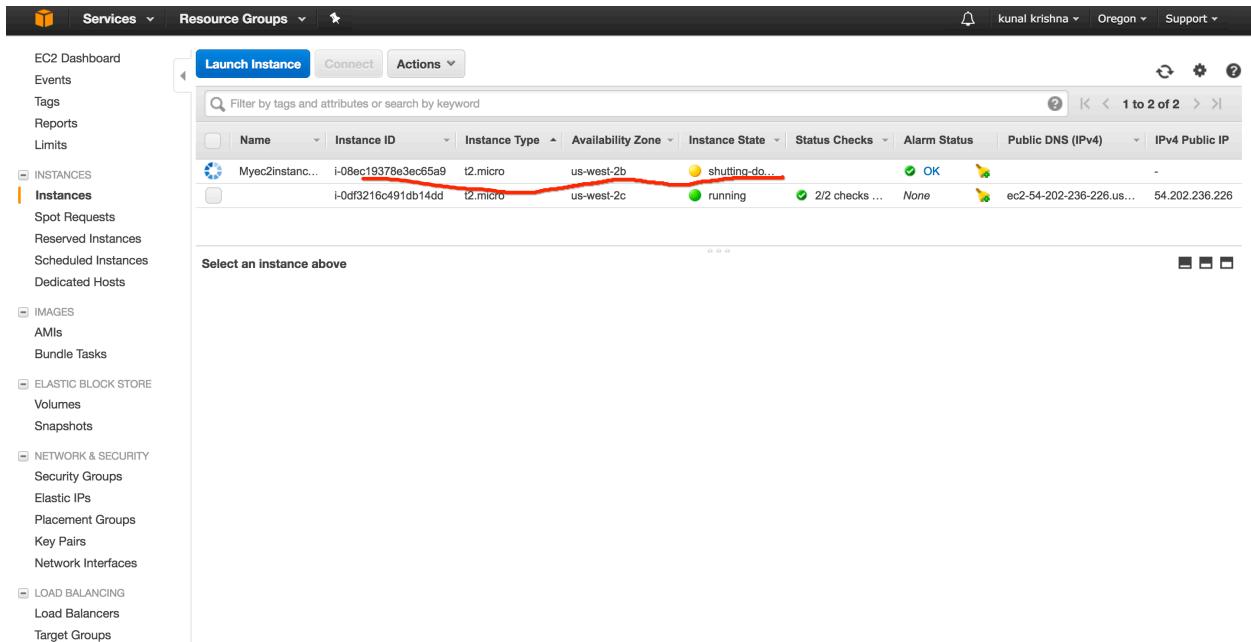
Below screen shot will attach my instance to auto scaling

This screenshot shows the 'Attach to Auto Scaling Group' dialog box. It has a title bar 'Attach to Auto Scaling Group' with a close button 'X'. The main area says 'Attach an instance to:' with two radio button options: 'a new Auto Scaling group' and 'an existing Auto Scaling group'. The second option is selected. Below that is a dropdown menu titled 'Auto Scaling Group' with a search bar. It lists one item: 'Assigment1AutoscalingGroup' (with 'us-west-2a, us-west-2b, us-west-2c' listed under 'Availability Zones'). This item is also highlighted with a red box. At the bottom of the dialog, there's a note about proceeding with the action and a confirmation message: 'Are you sure you want to attach i-08ec19378e3ec65a9 (ec2-54-200-106-91.us-west-2.compute.amazonaws.com)?'. At the very bottom are 'Cancel' and 'Attach' buttons, with 'Attach' being highlighted with a blue box.

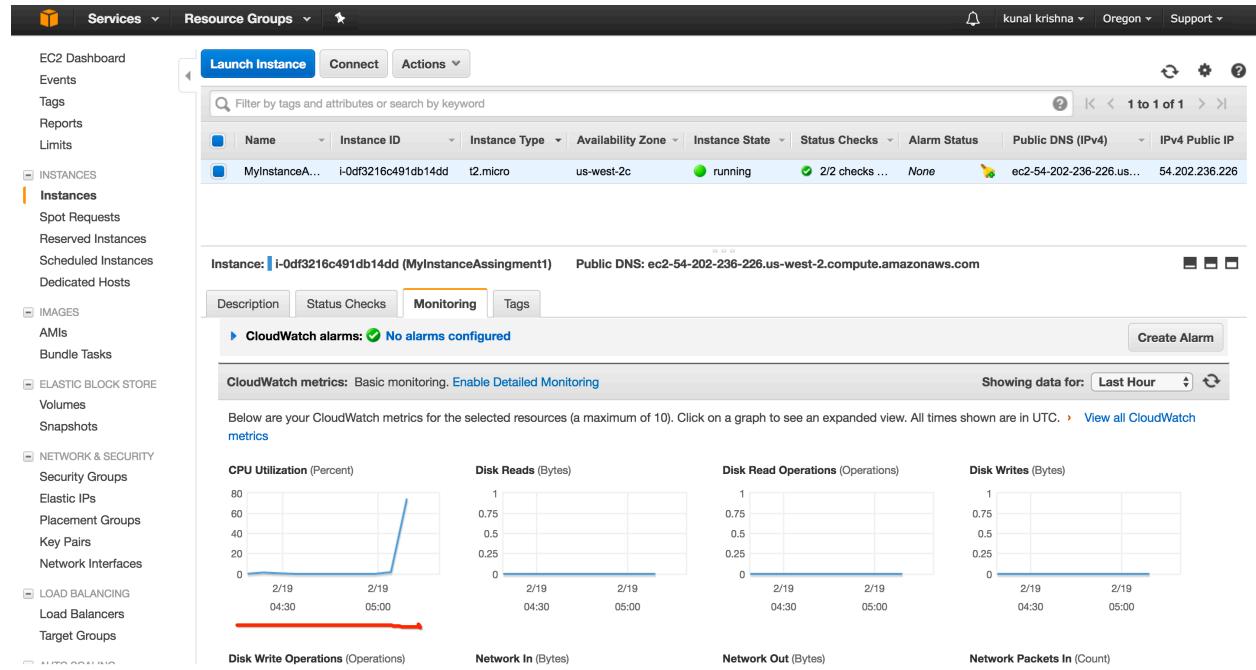
Below screen shots shows my initial CPU utilization.



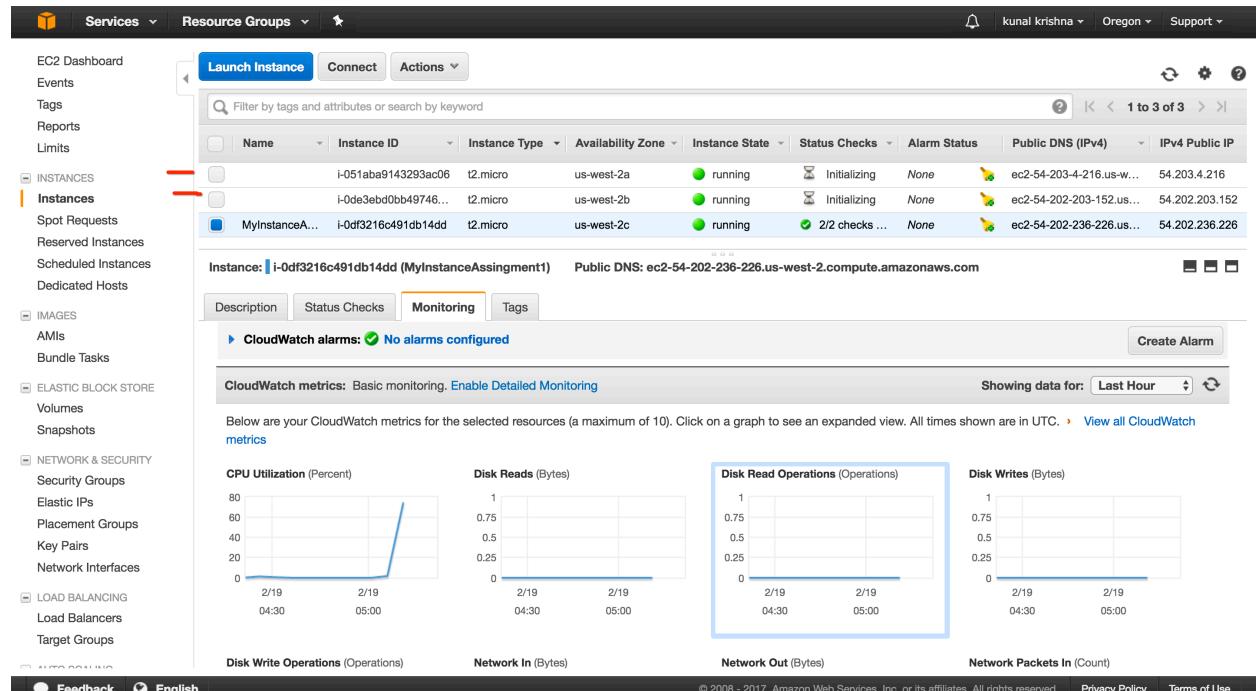
As I have attached my existing instance with auto scaling group and now my cpu utilization is less than 50 percentage then my attached instance will be shut down.



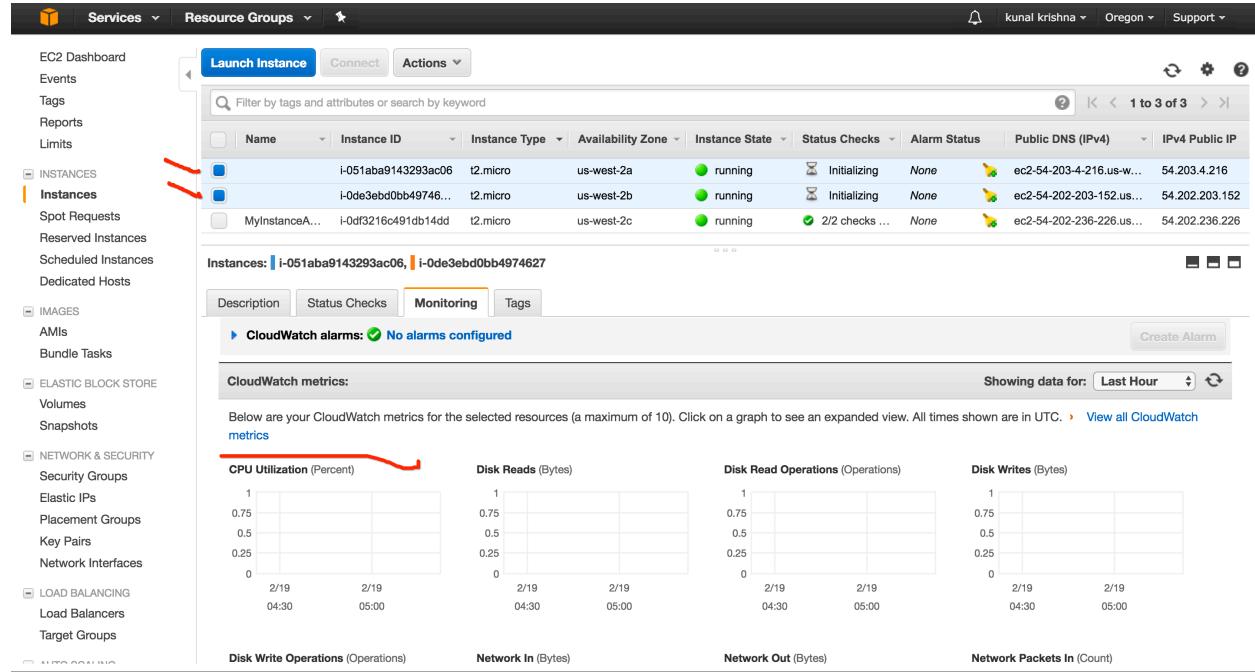
Now I am running my client server program on EC2 instance. Below screen print shows my utilization is more than 70 percentage. Once I get an email notification or after 150 seconds new instance will be launched for scaling purpose.



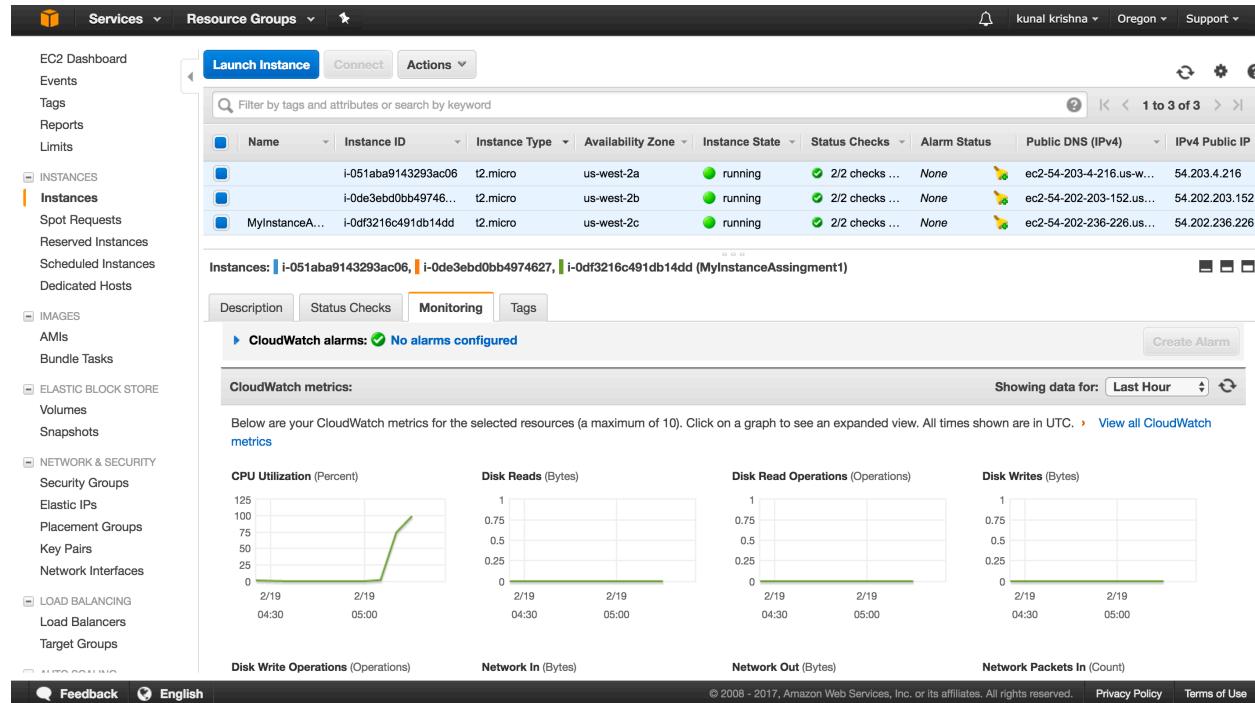
Now below screen print we can see two new instance will be created as CPU utilization is more than 70 percentage.



Below screen print for two new instances and their cpu Utilization.



Below screen print where my CPU utilization increased to more than 90 percentage and now 3 instances are running.



Now since our cpu utilization is now decreasing then our first instance is now being terminated.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for Services (EC2 Dashboard, Events, Tags, Reports, Limits), Instances (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, Network Interfaces), and Load Balancing (Load Balancers, Target Groups). The main area displays a table of instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-051aba9143293ac06	t2.micro	us-west-2a	running	2/2 checks ...	None	ec2-54-203-4-216.us-w...	54.203.4.216
	i-0de3ebd0bb4974627	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-54-202-203-152.us...	54.202.203.152
MyInstanceA...	i-0df3216c491db14dd	t2.micro	us-west-2c	shutting-down		None	-	-

Below the table, it says "Instances: i-051aba9143293ac06, i-0de3ebd0bb4974627". There are tabs for Description, Status Checks, Monitoring (which is selected), and Tags. Under Monitoring, it says "CloudWatch alarms: No alarms configured" and "Create Alarm". Below that is a section titled "CloudWatch metrics:" with four graphs: CPU Utilization (Percent), Disk Reads (Bytes), Disk Read Operations (Operations), and Disk Writes (Bytes). At the bottom, there are four more metrics: Disk Write Operations (Operations), Network In (Bvtes), Network Out (Bytes), and Network Packets In (Count).

This screenshot is similar to the one above, showing the AWS EC2 Dashboard. The sidebar and instance table are identical. However, the status of "MyInstanceA..." has changed from "shutting-down" to "terminated". A red arrow points from the text below the table to this status change.

Below the table, it says "Instances: i-051aba9143293ac06, i-0de3ebd0bb4974627". The "Monitoring" tab is still selected. Under "Description", there is a bulleted list:

- i-051aba9143293ac06: ec2-54-203-4-216.us-west-2.compute.amazonaws.com
- i-0de3ebd0bb4974627: ec2-54-202-203-152.us-west-2.compute.amazonaws.com

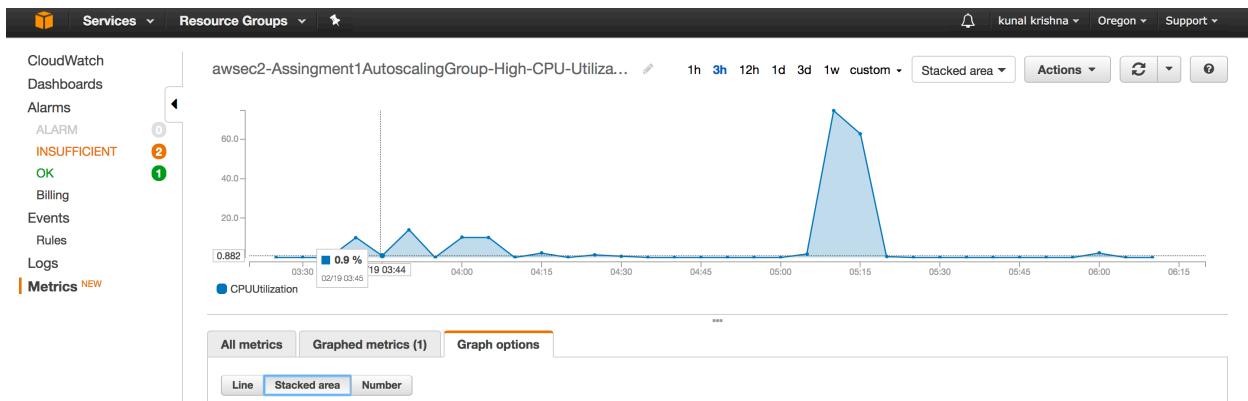
Below Screen print shows our inbound rule which will be used to connect our client server program.

Ports	Protocol	Source	MywebServer
80	tcp	0.0.0.0/:/0	✓
0-65535	tcp	0.0.0.0/:/0	✓
110	tcp	0.0.0.0/:/0	✓
995	tcp	0.0.0.0/:/0	✓
22	tcp	0.0.0.0/:/0	✓
25	tcp	0.0.0.0/:/0	✓
53	udp	0.0.0.0/:/0	✓
53	tcp	0.0.0.0/:/0	✓
0-65535	udp	0.0.0.0/:/0	✓
443	tcp	0.0.0.0/:/0	✓

Now 2nd instance is also shutting down as cpu Utilization is not increasing.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
i-051aba9143293ac06	i-051aba9143293ac06	t2.micro	us-west-2a	shutting-do...	None	-	-	-
i-0de3ebd0bb49746...	i-0de3ebd0bb49746...	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-54-202-203-152.us...	54.202.203.152
MyInstanceA...	i-0df3216c491db14dd	t2.micro	us-west-2c	terminated	None	-	-	-

I have created Cloud watch alarm to show my CPU utilization and in case my free tier limit is over then I will get notification via email.



Problem and Solution: I got problem while running client server program then I did setting in inbound rule.

Second problem: When I start my scaling group in which minimum instance I set as one and maximum as 3 and also set auto scaling up and down rule. So, when I attached my existing instance in which I install all my java and updated it got terminated as my CPU utilization was not more than 50 percentage (as per auto scaling down rule). So I still need to find solution to figure out how to make sure particular instance should not be deleted or in case of deletion it should copy all the data and setting to remaining running instance. I created one script which can be used whenever new instance is created everything should be set up by using that script and files will be copied.

Part 3: Code to manage AWS instances. Infrastructure as Code

First install aws cli in our system.

I used below command to install in my Mac.

```
brew install awscli
```

Below screen print will showing how to do aws configure before connecting to aws server using aws cli.

For this we need to create a user using IAM

The screenshot shows the AWS IAM 'Users' section. A user named 'MyCliUser' is selected. The top navigation bar includes 'Services', 'Resource Groups', a search bar, and account information for 'kunal krishna'. The left sidebar has links for Dashboard, Groups, Users (selected), Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area displays the user's ARN (arn:aws:iam::900000000000:user/MyCliUser), Path (/), and Creation time (2017-02-16 13:57 CST). Below this, tabs for 'Permissions', 'Groups (0)', 'Security credentials' (selected), and 'Access Advisor' are shown. Under 'Sign-in credentials', it lists Console password (Disabled), Console login link (N/A), Last login (N/A), Assigned MFA device (No), and Signing certificates (None). Under 'Access keys', it says 'Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.' A 'Create access key' button is present. A table shows one access key entry:

Access key ID	Created	Last used	Status
AKIAJ...OSA	2017-02-16 13:57 CST	N/A	Active Make inactive X

Now after than use below command to connect ec2 instance using cli.

```
aws configure
```

Command to Stop EC2 Instance:

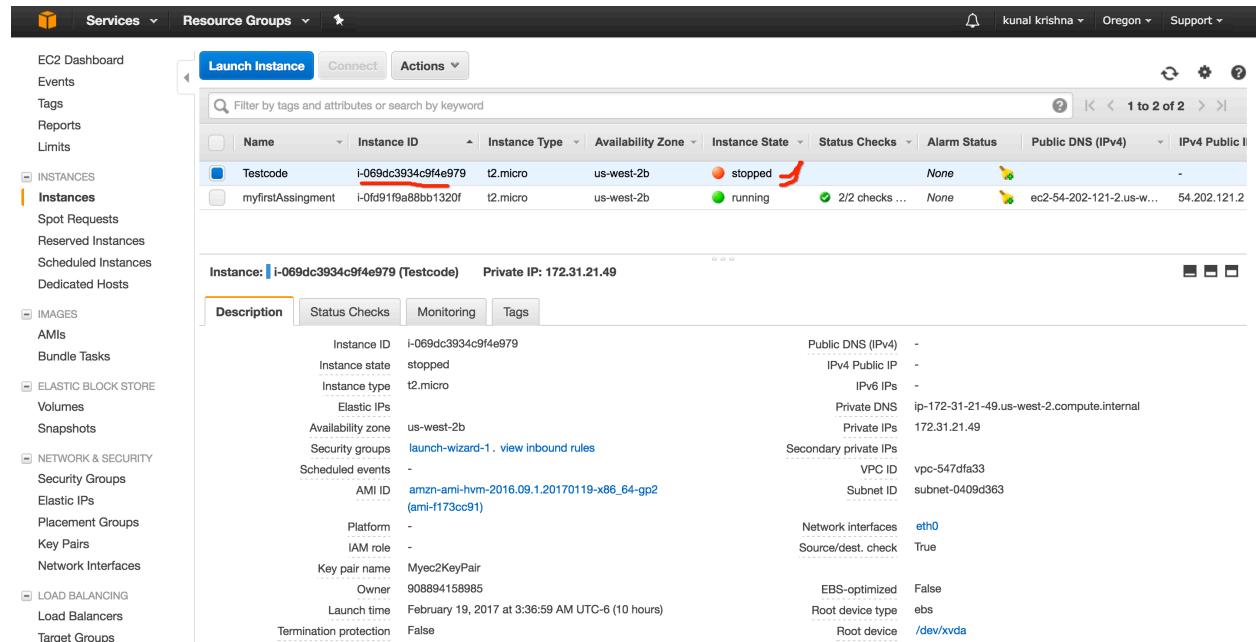
```
aws ec2 stop-instances --instance-ids i-069dc3934c9f4e979
```

Here above command only we need to replace instance id (i-069dc3934c9f4e979) this one we can get from ec2 instance.

```
[root@ip-172-31-21-49 ec2-user]# aws configure
AWS Access Key ID [*****5KOA]: AKIAITH4JL3VQ5UH
AWS Secret Access Key [*****W1I]: kydVWKqfMv...ovBe+W1I
Default region name [us-west-2b]: us-west-2
Default output format [None]:
[root@ip-172-31-21-49 ec2-user]# aws ec2 stop-instances --instance-ids i-069dc3934c9f4e979
{
    "StoppingInstances": [
        {
            "InstanceId": "i-069dc3934c9f4e979",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
[root@ip-172-31-21-49 ec2-user]#
Broadcast message from root@ip-172-31-21-49
(unknown) at 9:45 ...

The system is going down for power off NOW!
Connection to 54.201.201.252 closed by remote host.
Connection to 54.201.201.252 closed.
kunals-MacBook-Pro:ssh kunalkrishna$
```

Below screen print shows stopped the command.



Command to Start EC2 Instance:

```
aws ec2 start-instances --instance-ids i-069dc3934c9f4e979
```

Here above command only we need to replace instance id (i-069dc3934c9f4e979) this one we can get from ec2 instance.

```
[kunals-MacBook-Pro:~ kunalkrishna$ aws ec2 start-instances --instance-ids i-069dc3934c9f4e979
{
  "StartingInstances": [
    {
      "InstanceId": "i-069dc3934c9f4e979",
      "CurrentState": {
        "Code": 0,
        "Name": "pending"
      },
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ]
}
```

Below screen print shows it is in running status.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (which is selected), Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts, IMAGES (AMIs, Bundle Tasks), and ELASTIC BLOCK STORE (Volumes, Snapshots). 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 Status, Public DNS (IPv4), and IPv4 Public IP. There are two rows in the table:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
Testcode	i-069dc3934c9f4e979	t2.micro	us-west-2b	running	Initializing	None	ec2-54-244-234-157.us...	54.244.234.157
myfirstAssignment	i-0fd119a88bb1320f	t2.micro	us-west-2b	running	2/2 checks	None	ec2-54-202-121-2.us-w...	54.202.121.2

Below the table, a message says 'Select an instance above'.

Command to terminate EC2 Instance:

```
aws ec2 terminate-instances --instance-ids i-069dc3934c9f4e979
```

Here above command only we need to replace instance id (i-069dc3934c9f4e979) this one we can get from ec2 instance.

```
[kunals-MacBook-Pro:~ kunalkrishna$ aws ec2 terminate-instances --instance-ids i-069dc3934c9f4e979
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-069dc3934c9f4e979",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
kunals-MacBook-Pro:~ kunalkrishna$ ]
```

Below screen print shows that instance is terminated.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Scheduled Instances, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, and Network & Security. The 'Instances' link is currently selected. The main area displays a table of instances. The first row, 'Testcode', has its instance ID (i-069dc3934c9f4e979) highlighted with a red box. Its status is 'terminated'. The second row, 'myfirstAssgningment', has its instance ID (i-0fd91f9a88bb1320f) highlighted with a red box. Its status is 'running'. Below the table, a detailed view for the 'Testcode' instance is shown, listing its configuration: Instance ID (i-069dc3934c9f4e979), Instance state (terminated), Instance type (t2.micro), Elastic IPs, Availability zone (us-west-2b), Security groups, and Scheduled events. The Public DNS column is empty for both instances.

Command to launch EC2 Instance:

```
aws ec2 run-instances --image-id ami-f173cc91 --security-group-ids sg-634f2f1b --count 1 --instance-type t2.micro --key-name Myec2KeyPair --query 'Instances[0].InstanceId'.
```

In the above command we need to give image-id this you get it on EC2 screen where all the options are there to select type of instance.

Security group id . I have created one security group id while setting up an instance.

This is the same network security group where we set our inbound and outbound rules,

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. A red circle highlights the 'MywebServer' security group, which has a red X icon next to its name.

Name	Group ID	Group Name	VPC ID	Description
sg-120ebf6a		default	vpc-547dfa33	default VPC security group
sg-634f2f1b		MywebServer	vpc-547dfa33	MywebServer
sg-96770cee		launch-wizard-1	vpc-547dfa33	launch-wizard-1 created 2017-02-19T03:36:36.555-06:00

You can also create security group using command prompt with below command.

```
aws ec2 create-security-group --group-name cligroup-sg --description
"security group for development environment in EC2"
{
    "GroupId": "sg-2ccb654"
}
```

```
aws ec2 authorize-security-group-ingress --group-name cligroup-sg --protocol
tcp --port 22 --cidr 0.0.0.0/0
```

Below screen print to shows command to create security group.

```
kunals-MacBook-Pro:~ kunalkrishna$ aws ec2 create-security-group --group-name cligroup-sg --description "security group for development environment in EC2"
{
    "GroupId": "sg-2ccb654"
```

Below Screen print shows to add inbound rule.

```
kunals-MacBook-Pro:~ kunalkrishna$ aws ec2 authorize-security-group-ingress --group-name cligroup-sg --protocol tcp --port 22 --cidr 0.0.0.0/0
kunals-MacBook-Pro:~ kunalkrishna$
```

Same thing we can verify on EC2 instance.

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. A table lists three security groups:

Name	Group ID	Group Name	VPC ID	Description
sg-120ebf6a		default	vpc-547dfa33	default VPC security group
sg-2cccb654		cligroup-sg	vpc-547dfa33	security group for development environment in EC2
sg-634f2f1b		MywebServer	vpc-547dfa33	MywebServer

Below the table, a detailed view of the 'cligroup-sg' security group is shown. The 'Inbound' tab is selected, displaying a single rule:

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0

Instance-type: I am using t.micro as this is free one.

Key-name: We need to create key value pair while launching an instance, this option we will get once we launch **our very first instance**. In my case I have already created key value pair so I am using the same one,

If you want to create using cli then below command to create key-value pair.:

```
$ aws ec2 create-key-pair --key-name Myec2KeyPair1 --query 'KeyMaterial' --output text > Myec2KeyPair1.pem
```

```
[kunals-MacBook-Pro:~ kunalkrishna$ aws ec2 create-key-pair --key-name Myec2KeyPair1 --query 'KeyMaterial' --output text > Myec2KeyPair1.pem
[REDACTED]
```

We can verify the same thing on EC2 instance.

The screenshot shows the AWS EC2 Key Pairs page. On the left, there's a sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, and Network & Security. The main area has tabs for Create Key Pair, Import Key Pair, and Delete. A search bar is at the top. Below it is a table with columns for Key pair name and Fingerprint. The table contains three rows:

Key pair name	Fingerprint
Myec2KeyPair12	45:2a:3c:bb:a1:65:16:a6:6a:8d:0f:4a:d9:d4:b0:a4:fd:cf:6a:f5
Myec2KeyPair1	18:7f:6d:7d:83:39:9a:72:c6:10:94:4a:75:e6:d0:77:d2:31:8e:1d
Myec2KeyPair	50:b7:42:bb:fb:4d:ab:9b:3d:ea:15:b2:19:97:a7:80:66:7d:70:4d

```
kunals-MacBook-Pro:~ kunalkrishna$ aws ec2 run-instances --image-id ami-f173cc91 --security-group-ids sg-634f2f1b --count 1 --instance-type t2.micro --key-name Myec2KeyPair --query 'Instances[0].InstanceId'  
"i-0f08fae1fda32ba9f"  
kunals-MacBook-Pro:~ kunalkrishna$
```

Now below screen print shows that instance is initialized.

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'INSTANCES' section, the 'Instances' option is selected. The main area displays a table of instances. One instance, with the Instance ID `i-0f08fae1fda328a9f`, is highlighted with a red box around its Instance ID and the 'Initializing' status column. Another instance, named 'myfirstAssingment', is also listed. The table includes columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Public IP.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-0f08fae1fda328a9f	t2.micro	us-west-2a	running	Initializing	None	ec2-54-191-151-151.us...	54.191.151.151
myfirstAssingment	i-0fd91f9a88bb1320f	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-54-202-121-2.us-w...	54.202.121.2

Problem and solution:

Error I was getting as unable to connect endpoint URL from AWS cli.

Solution : correct the value of region

Use `us-west-2` instead of `us-west-2b` in your Endpoint URL. Most probably, this would be the

value of `REGION` in your configuration file that was created by running `aws configure`

`us-west-2` is a region

`us-west-2b` is an availability Zone in that region

I gave `us-west-2b` while putting region.

stop:

```
aws ec2 stop-instances --instance-ids i-069dc3934c9f4e979
```

Start:

```
aws ec2 start-instances --instance-ids i-069dc3934c9f4e979
```

terminate or delete:

```
aws ec2 terminate -instances --instance-ids i-069dc3934c9f4e979
```