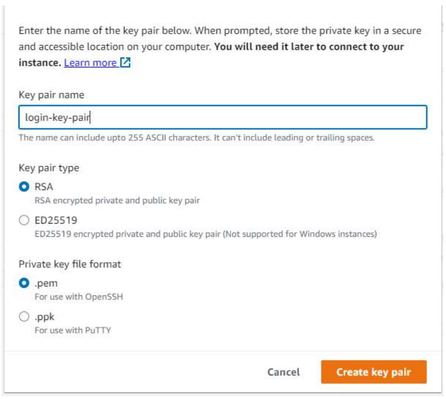
HA JENKINS WITH AWS AUTO-SCALING GROUP, EFS AND ELB FOR SMALL TEAMS:

Creating JENKINS_AMI for Auto-scaling group:

1. Login into AWS account and Launch an EC2 instance with the following details:
   Name: JENKINS_AMI (You can give any thing)
   OS_Type: Amazon Linux
   instance_Type: t2.micro
   Create a key-pair to login from your local system:
   - Click on Create new pair hyperlink in the EC2 Launch page.
   - Give name and leave other options as default and click on create key pair button.


Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

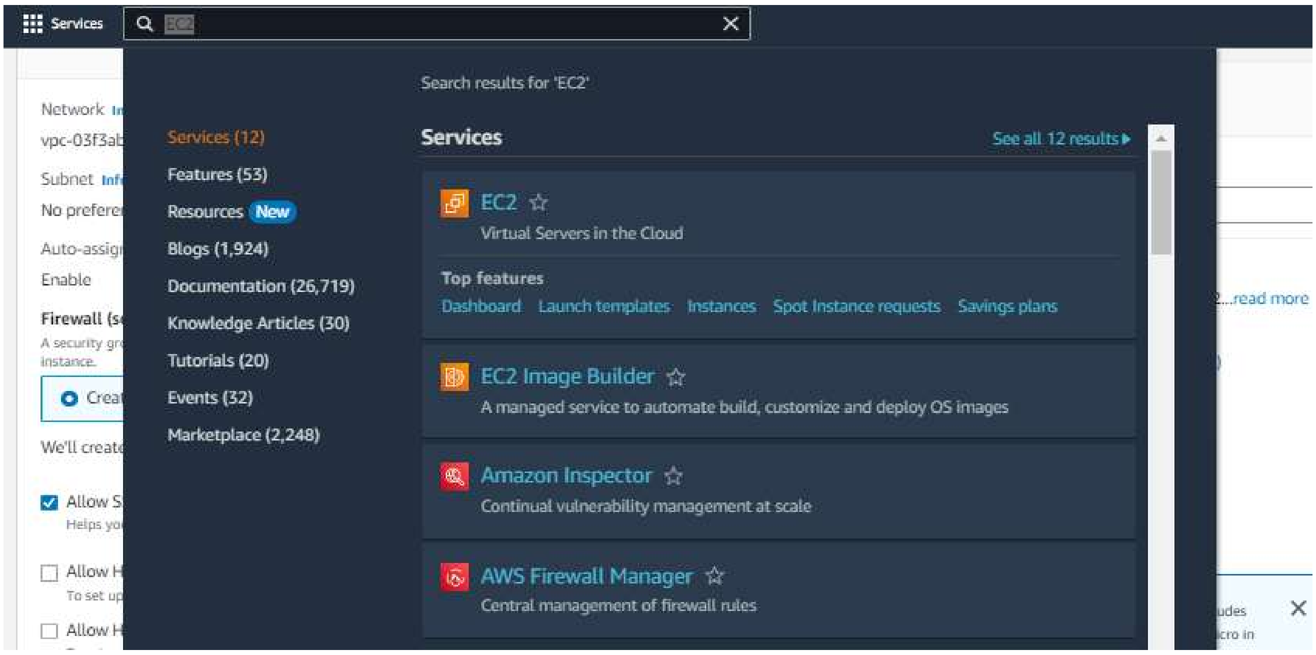
Key pair name: login-key-pair

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

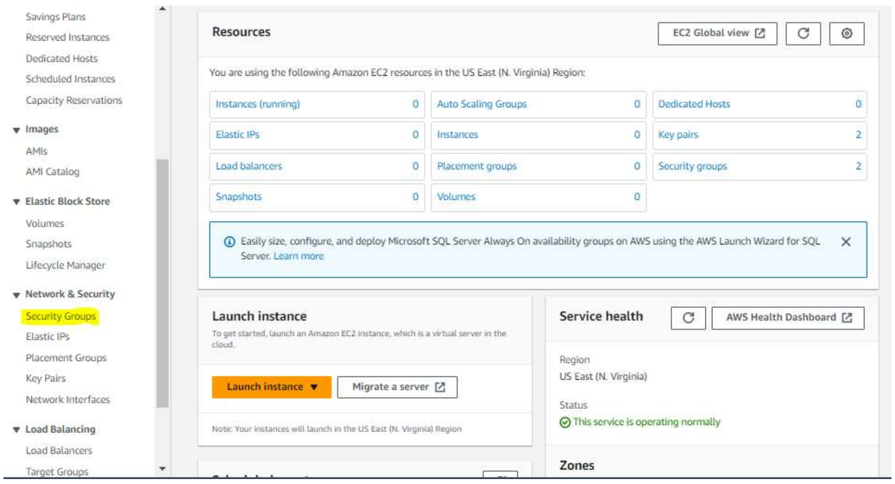
Key pair type
- [✓ RSA] RSA encrypted private and public key pair
- ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format
- [✓ .pem] For use with OpenSSH
- .ppk For use with PuTTY

   - Key-pair will download to our local.
   - Select the key-pair that you have created in the previous step.
2. In the Network settings, Select your VPC. I am using the default VPC.
3. Lets create a security group:
   - Search for EC2 service on search bar, and Open it in a new tab as shown below:



- Click on create a security group



You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	2
Load balancers	0	Placement groups	0	Security groups	2
Snapshots	0	Volumes	0		

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

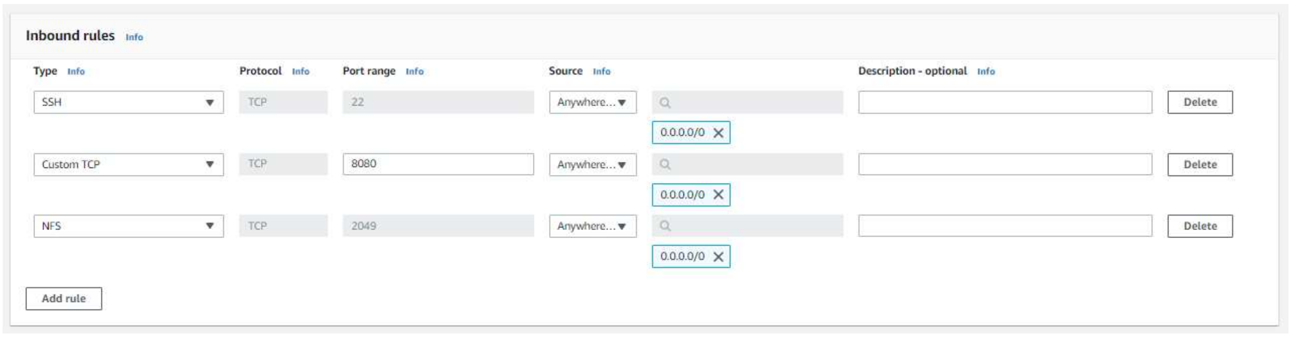
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

Region
US East (N. Virginia)

Status
This service is operating normally

- Add security group name and give description
- click on add rule below in inbound rules and add rules as shown below:

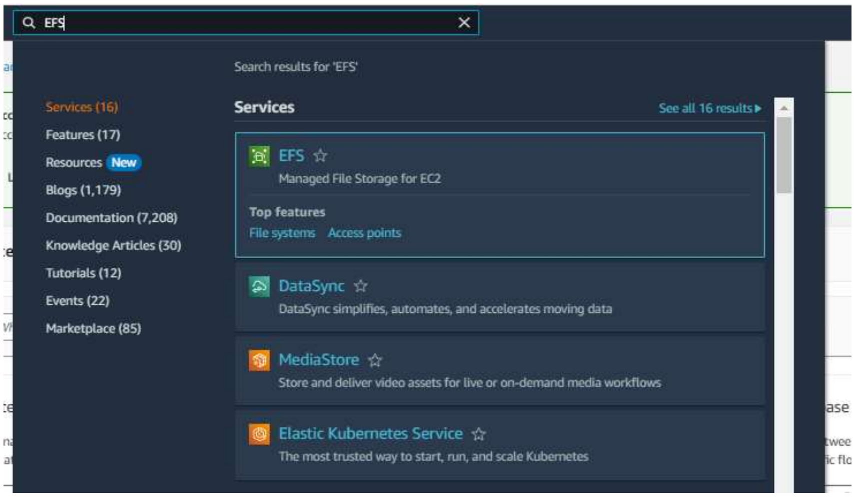

Inbound rules Info

Type	Protocol	Port range	Source		Description - optional	
SSH	TCP	22	Anywhere...	0.0.0.0/0		Delete
Custom TCP	TCP	8080	Anywhere...	0.0.0.0/0		Delete
NFS	TCP	2049	Anywhere...	0.0.0.0/0		Delete

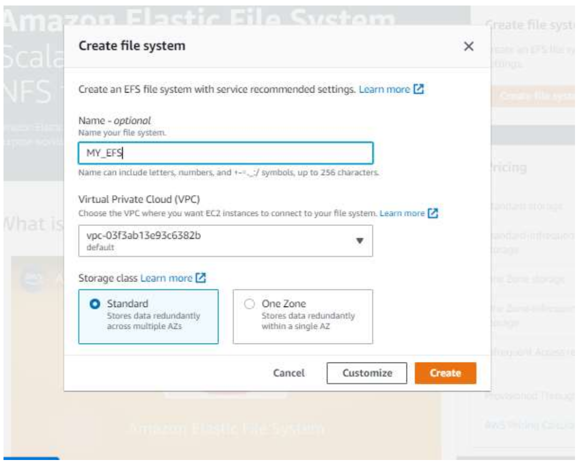
Add rule

- Port 22 is to connect ec2 instance from our local machine and Port 8080 exposes our jenkins and Port 2049 is for EFS mount. You can fine tune source IP's, If you want. I have allowed anywhere.
- Come back to previous tab and select the existing secuirty groups and click on refresh button.
- Select the security group that you have created.
- Leave all the remaining options as default and click on Launch instane.

4. Lets create a EFS for our jenkins instances. Meanwhile, the jenkins instance will get started.
    - Search EFS on search bar and click on it as shown below:



- Click on create EFS and give name for EFS and select the vpc that you are using.
- Select standard for storage class as we will run instances in different AZs.


Create file system

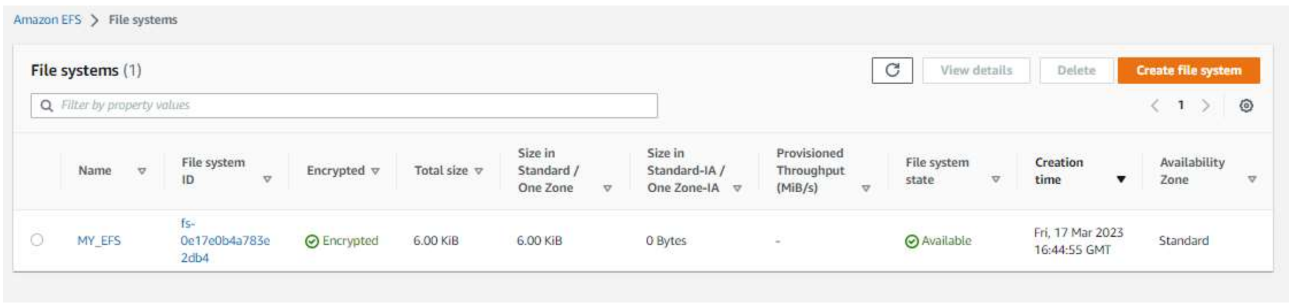
Create an EFS file system with service recommended settings. Learn more

Name - *optional*
Name your file system.
MY_EFS
Name can include letters, numbers, and +-=._:/ symbols, up to 256 characters.

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system. Learn more
vpc-03f3ab13e93c6382b
default

Storage class Learn more
[✓ Standard — Stores data redundantly across multiple AZs] One Zone — Stores data redundantly within a single AZ

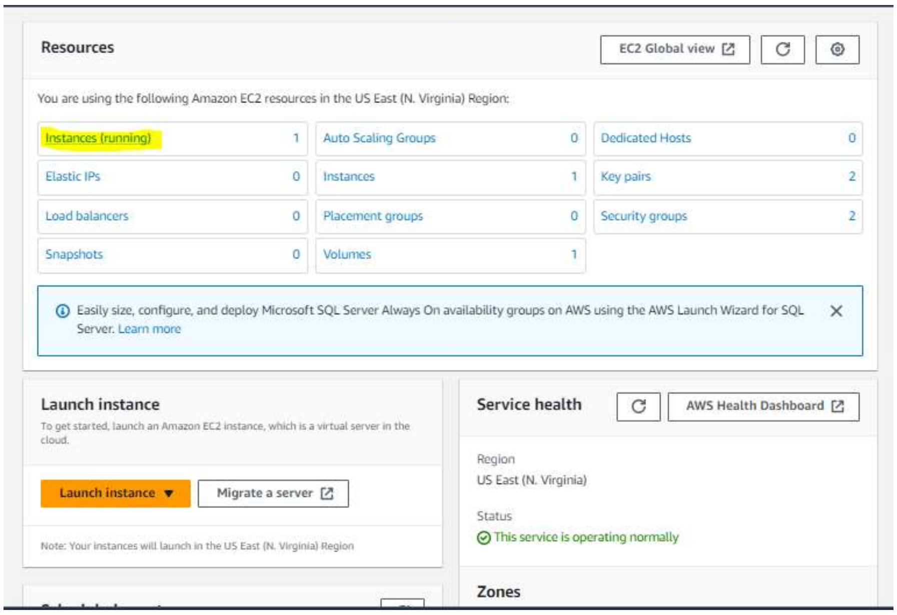
- Click on create and EFS will be created



File systems (1)

	Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	File system state	Creation time	Availability Zone
○	MY_EFS	fs-0e17e0b4a783e2db4	✓ Encrypted	6.00 KiB	6.00 KiB	0 Bytes	-	✓ Available	Fri, 17 Mar 2023 16:44:55 GMT	Standard

5. Open Gitbash in the path where your key-pair is downloaded.
6. Connect to the EC2 with that key-pair and public IP of EC2 using the following steps:
    - Open the EC2 service in AWS console and click on instances as shown below:



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	2
Load balancers	0	Placement groups	0	Security groups	2
Snapshots	0	Volumes	1		

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

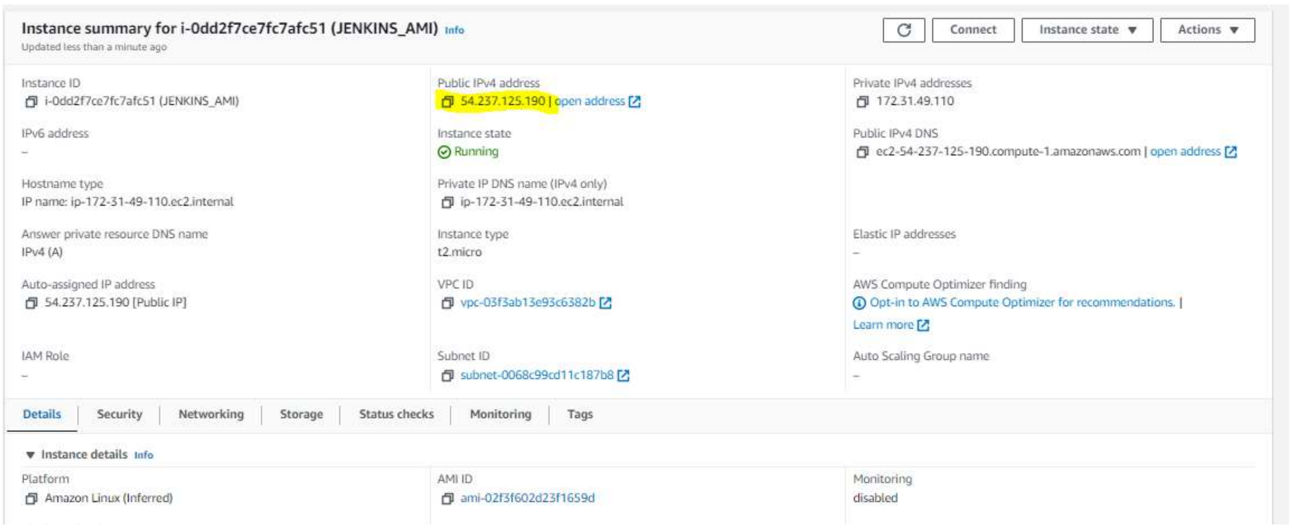
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

Service health
Region
US East (N. Virginia)
Status
This service is operating normally

Zones

    - Click on the instance that you created and copy the public IP of EC2 as shown below:



Instance summary for i-0dd2f7ce7fc7afc51 (JENKINS_AMI) Info
Updated less than a minute ago

Instance ID i-0dd2f7ce7fc7afc51 (JENKINS_AMI)	Public IPv4 address 54.237.125.190 | open address	Private IPv4 addresses 172.31.49.110
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-54-237-125-190.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-49-110.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-49-110.ec2.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 54.237.125.190 [Public IP]	VPC ID vpc-03f3ab13e93c6382b	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0068c99cd11c187b8	Auto Scaling Group name –

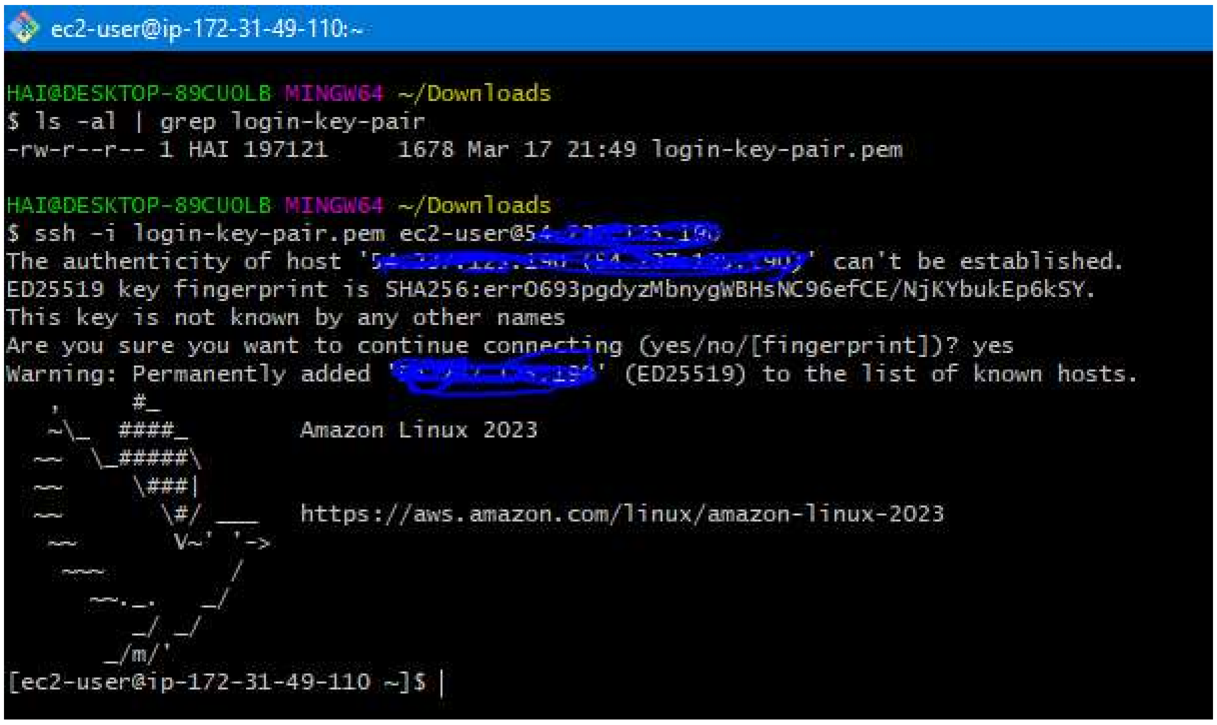
Instance details Info

Platform Amazon Linux (Inferred)	AMI ID ami-02f3f602d23f1659d	Monitoring disabled

  - Connect to EC2 with the following command in your git bash:

```
$ ls -al | grep login-key-pair
$ ssh -i login-key-pair.pem ec2-user@54.237.125.190
```

then type yes and hit enter, You will be logged into the ec2 as shown below:



```
ec2-user@ip-172-31-49-110:~

HAI@DESKTOP-89CUOLB MINGW64 ~/Downloads
$ ls -al | grep login-key-pair
-rw-r--r-- 1 HAI 197121     1678 Mar 17 21:49 login-key-pair.pem

HAI@DESKTOP-89CUOLB MINGW64 ~/Downloads
$ ssh -i login-key-pair.pem ec2-user@[redacted]
The authenticity of host '[redacted]' can't be established.
ED25519 key fingerprint is SHA256:err0693pgdyzMbnygWBHsNC96efCE/NjKYbukEp6kSY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[redacted]' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-49-110 ~]$ |
```

7. Lets mount the EFS and Install Jenkins:

- Update the packages by running the following cmds:

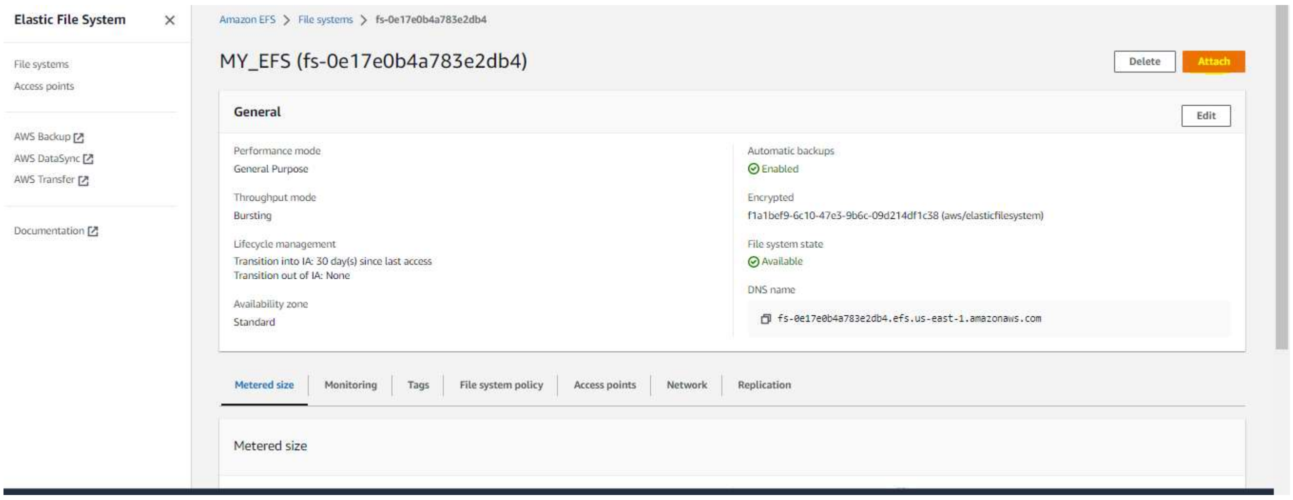
```
$ sudo -i
$ yum update -y
```

- Mount EFS to the path /usr/lib/jenkins (As it is the default Jenkins homepage) by running the following cmds:

  - create a directory by running below cmd:

    ```
    $ mkdir /var/lib/jenkins
    ```

  - Go to EFS page and click on EFS which you have created. Click on Attach as shown below:



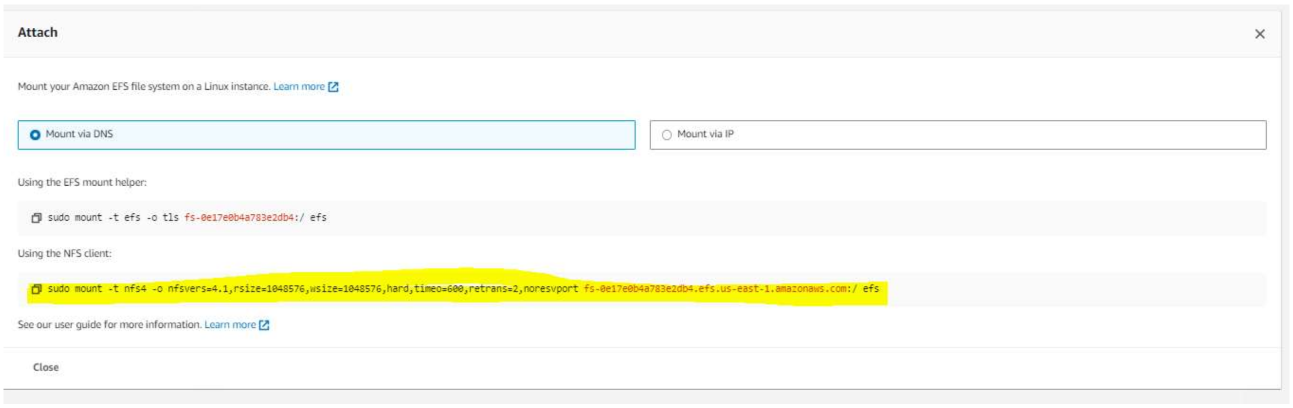
MY_EFS (fs-0e17e0b4a783e2db4)

General

Performance mode	General Purpose
Throughput mode	Bursting
Lifecycle management	Transition into IA: 30 day(s) since last access Transition out of IA: None
Availability zone	Standard
Automatic backups	Enabled
Encrypted	f1a1bef9-6c10-47e3-9b6c-09d214df1c38 (aws/elasticfilesystem)
File system state	Available
DNS name	fs-0e17e0b4a783e2db4.efs.us-east-1.amazonaws.com

Metered size

- Copy the command in the page as shown in below and edit the path efs to /var/lib/jenkins as shown below:



Mount your Amazon EFS file system on a Linux instance. Learn more

Using the EFS mount helper:

```
sudo mount -t efs -o tls fs-0e17e0b4a783e2db4:/ efs
```

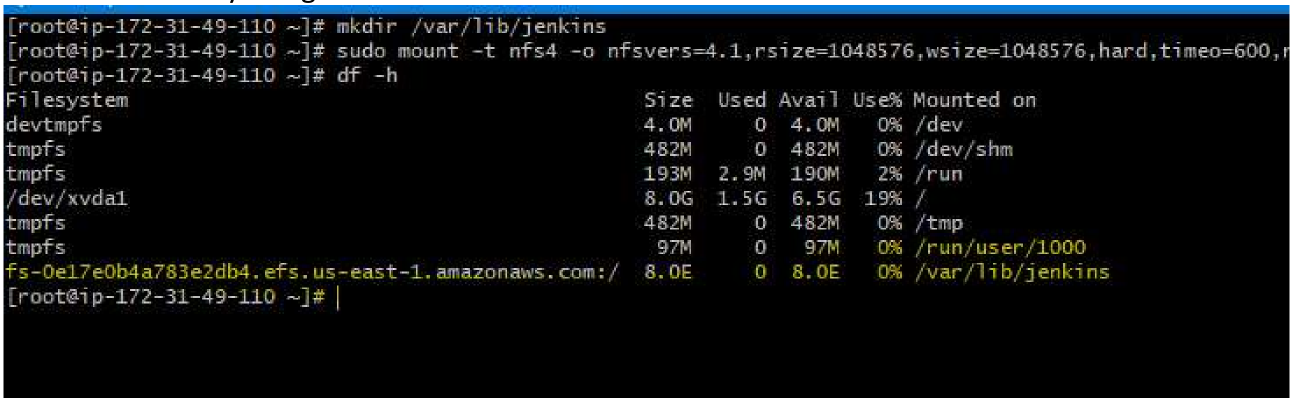
Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0e17e0b4a783e2db4.efs.us-east-1.amazonaws.com:/ efs
```

See our user guide for more information. Learn more

$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0e17e0b4a783e2db4.efs.us-east-1.amazonaws.com:/ /var/lib/jenkins

- You can check by using the cmd 'df -h' as shown below:



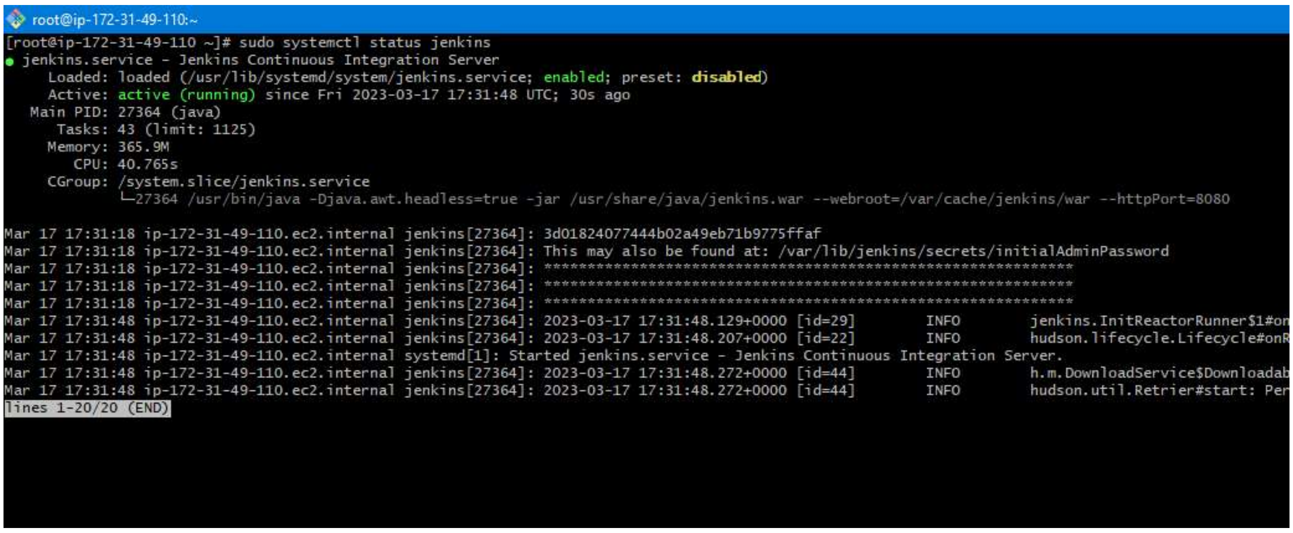
```
[root@ip-172-31-49-110 ~]# mkdir /var/lib/jenkins
[root@ip-172-31-49-110 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,r
[root@ip-172-31-49-110 ~]# df -h
Filesystem                                       Size  Used Avail Use% Mounted on
devtmpfs                                         4.0M     0  4.0M   0% /dev
tmpfs                                            482M     0  482M   0% /dev/shm
tmpfs                                            193M  2.9M  190M   2% /run
/dev/xvda1                                       8.0G  1.5G  6.5G  19% /
tmpfs                                            482M     0  482M   0% /tmp
tmpfs                                             97M     0   97M   0% /run/user/1000
fs-0e17e0b4a783e2db4.efs.us-east-1.amazonaws.com:/  8.0E     0  8.0E   0% /var/lib/jenkins
[root@ip-172-31-49-110 ~]# |
```

- Install the jenkins by running following cmds:

```
$ sudo wget -O /etc/yum.repos.d/jenkins.repo
https://pkg.jenkins.io/redhat-stable/jenkins.repo
$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
$ sudo yum upgrade
$ sudo yum install java-11-amazon-corretto-headless -y
$ sudo yum install jenkins -y
$ sudo systemctl daemon-reload
$ sudo systemctl start jenkins
```

- Now you can check the status of jenkins as shown below:

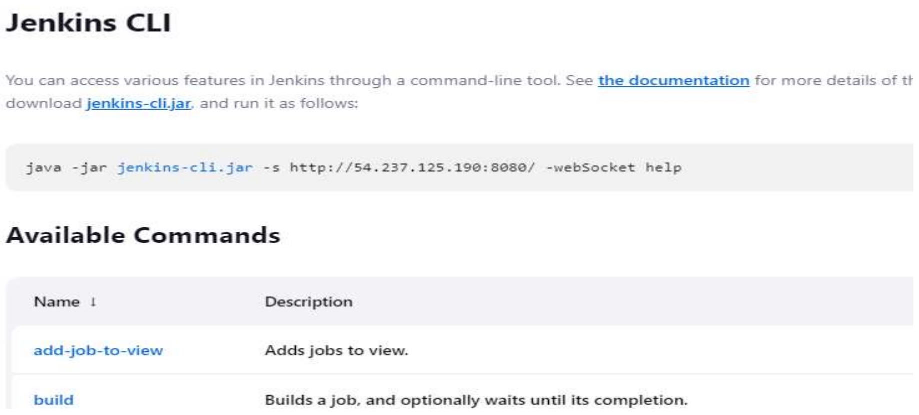
```
$ sudo systemctl status jenkins
```



```
root@ip-172-31-49-110:~
[root@ip-172-31-49-110 ~]# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
     Active: active (running) since Fri 2023-03-17 17:31:48 UTC; 30s ago
   Main PID: 27364 (java)
      Tasks: 43 (limit: 1125)
     Memory: 365.9M
        CPU: 40.765s
     CGroup: /system.slice/jenkins.service
             └─27364 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 17 17:31:18 ip-172-31-49-110.ec2.internal jenkins[27364]: 3d01824077444b02a49eb71b9775ffaf
Mar 17 17:31:18 ip-172-31-49-110.ec2.internal jenkins[27364]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 17 17:31:18 ip-172-31-49-110.ec2.internal jenkins[27364]: *************************************************************
Mar 17 17:31:18 ip-172-31-49-110.ec2.internal jenkins[27364]: *************************************************************
Mar 17 17:31:18 ip-172-31-49-110.ec2.internal jenkins[27364]: *************************************************************
Mar 17 17:31:48 ip-172-31-49-110.ec2.internal jenkins[27364]: 2023-03-17 17:31:48.129+0000 [id=29]        INFO        jenkins.InitReactorRunner$1#on
Mar 17 17:31:48 ip-172-31-49-110.ec2.internal jenkins[27364]: 2023-03-17 17:31:48.207+0000 [id=22]        INFO        hudson.lifecycle.Lifecycle#onR
Mar 17 17:31:48 ip-172-31-49-110.ec2.internal systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 17 17:31:48 ip-172-31-49-110.ec2.internal jenkins[27364]: 2023-03-17 17:31:48.272+0000 [id=44]        INFO        h.m.DownloadService$Downloadab
Mar 17 17:31:48 ip-172-31-49-110.ec2.internal jenkins[27364]: 2023-03-17 17:31:48.272+0000 [id=44]        INFO        hudson.util.Retrier#start: Per
lines 1-20/20 (END)
```

- Open jenkins page by http://PUBLIC_IP:8080/
- setup the jenkins with one admin account and install all the plugins by following the jenkins dashboard.
- Now Jenkins is up. Login into Jenkins and download the cli to EC2 instance by following the steps:
    - Go to Manage Jenkins > click on Jenkins CLI > Copy the jenkinscli link address
- make a directory as 'jenkins_sync' in root path and download jenkinscli by following cmd:



Jenkins CLI

You can access various features in Jenkins through a command-line tool. See the documentation for more details of th
download jenkins-cli.jar, and run it as follows:

```
java -jar jenkins-cli.jar -s http://54.237.125.190:8080/ -webSocket help
```

Available Commands

Name ↓	Description
add-job-to-view	Adds jobs to view.
build	Builds a job, and optionally waits until its completion.

$ mkdir /root/jenkins_sync
$ cd /jenkins_sync
$ curl -O http://localhost:8080/jnlpJars/jenkins-cli.jar
$ touch reload.sh
$ echo '#!/bin/bash' > reload.sh
$ echo 'java -jar /root/jenkins_sync/jenkins-cli.jar -s http://localhost:8080/ -auth ADMIN:admin reload-configuration' >> reload.sh

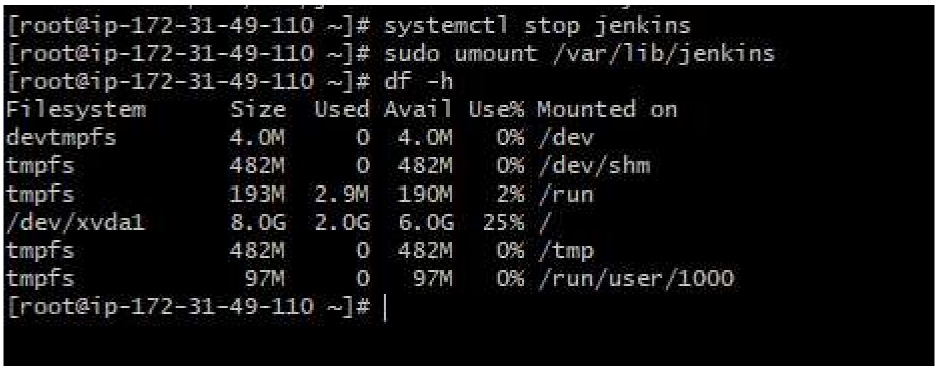
- Replace the ADMIN:admin with the your jenkins user and password.
- Change the permissions of reload.sh file

$ chmod 744 reload.sh

- Lets run this reload.sh file every minute to be in sync with cronjob in all jenkins instances after configuring with Autoscaling group.
  - Add the following line in the /etc/crontab file.
  - * * * * * root /root/jenkins_sync/reload.sh
  - execute the following cmds:

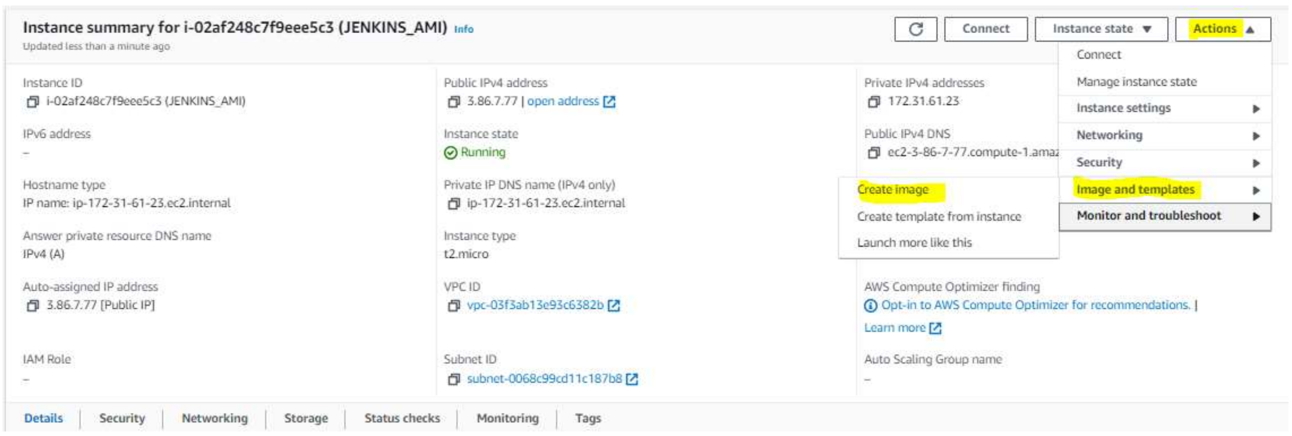
    $ yum install cronie -y
    $ systemctl start crond
    $ systemctl enable crond
- Now, we are good with Jenkins installation. Lets remove the efs and create AMI.
  - Run the below cmds:

    $ sudo systemctl stop jenkins
    $ sudo umount /var/lib/jenkins
    $ df -h



```
[root@ip-172-31-49-110 ~]# systemctl stop jenkins
[root@ip-172-31-49-110 ~]# sudo umount /var/lib/jenkins
[root@ip-172-31-49-110 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           482M     0  482M   0% /dev/shm
tmpfs           193M  2.9M  190M   2% /run
/dev/xvda1      8.0G  2.0G  6.0G  25% /
tmpfs           482M     0  482M   0% /tmp
tmpfs            97M     0   97M   0% /run/user/1000
[root@ip-172-31-49-110 ~]# |
```

8. Lets create an AMI, Go to EC2 console and click on your EC2 instance. Click on " Actions > image and template > create image " as shown below:



Instance summary for i-02af248c7f9eee5c3 (JENKINS_AMI) Info
Updated less than a minute ago

Instance ID
i-02af248c7f9eee5c3 (JENKINS_AMI)

IPv6 address
–

Hostname type
IP name: ip-172-31-61-23.ec2.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
3.86.7.77 [Public IP]

IAM Role
–

Public IPv4 address
3.86.7.77 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-61-23.ec2.internal

Instance type
t2.micro

VPC ID
vpc-03f3ab13e93c6382b

Subnet ID
subnet-0068c99cd11c187b8

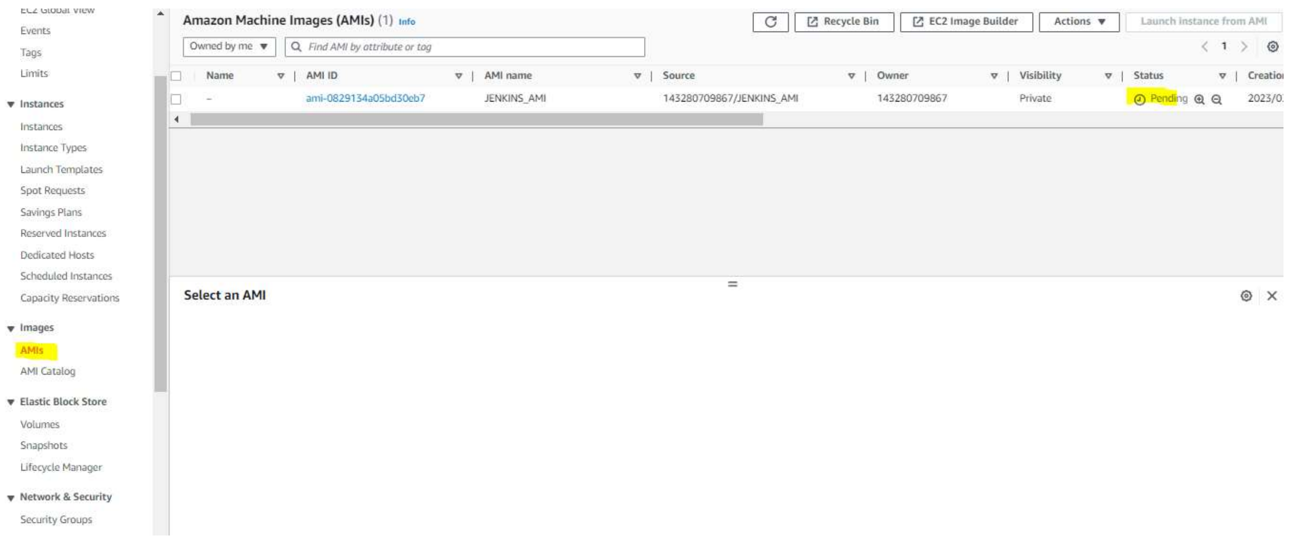
Private IPv4 addresses
172.31.61.23

Public IPv4 DNS
ec2-3-86-7-77.compute-1.ama

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. |
Learn more

Auto Scaling Group name
–

- Give name and description, leave remaining values as it is and click on create image.
- Click on AMI's and you can see the status of AMI as shown below:


Amazon Machine Images (AMIs) (1)

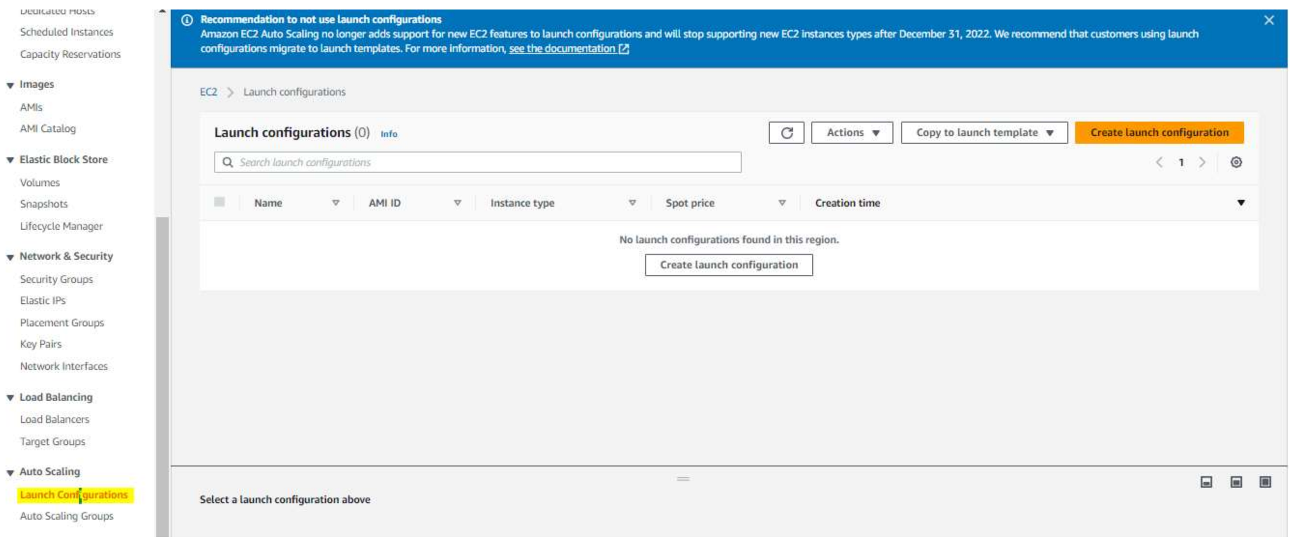
Name	AMI ID	AMI name	Source	Owner	Visibility	Status	Creatio
–	ami-0829134a05bd30eb7	JENKINS_AMI	143280709867/JENKINS_AMI	143280709867	Private	Pending	2023/0

Select an AMI

- Terminate your EC2 once the AMI image is available.

Creating the Jenkins servers Launch Configuration:

1. Once your AMI is created, we will use it to configure the Autoscaling group.
2. Go to EC2 console and click on Launch configuration as shown below:


Recommendation to not use launch configurations
Amazon EC2 Auto Scaling no longer adds support for new EC2 features to launch configurations and will stop supporting new EC2 instances types after December 31, 2022. We recommend that customers using launch configurations migrate to launch templates. For more information, see the documentation

EC2 > Launch configurations

Launch configurations (0)

Name	AMI ID	Instance type	Spot price	Creation time

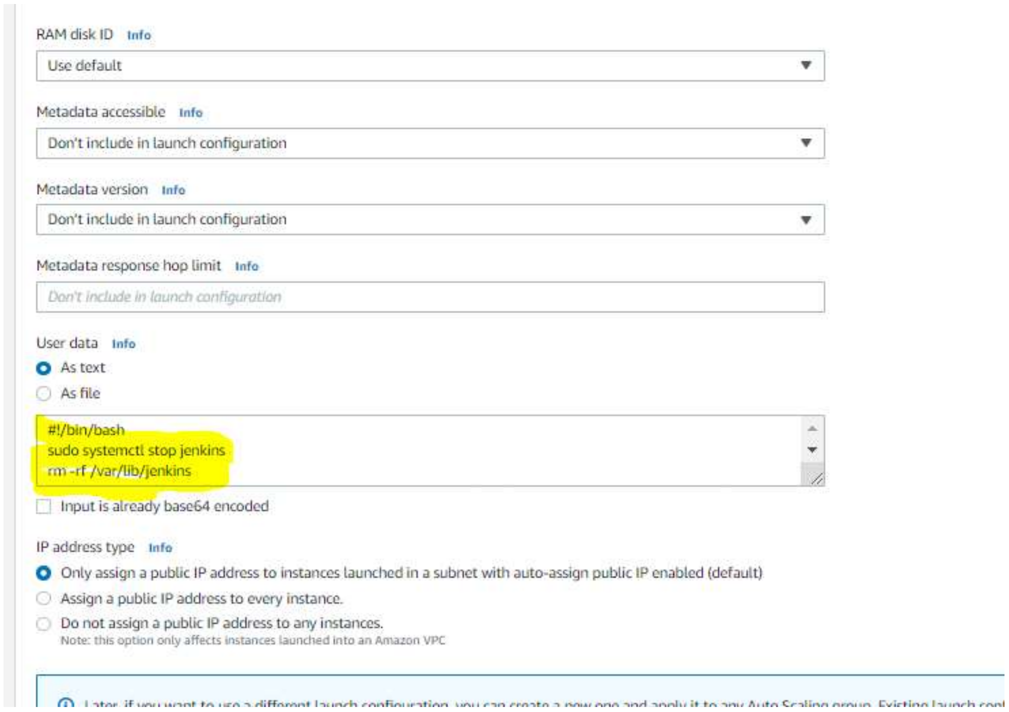
No launch configurations found in this region.

Select a launch configuration above

3. Click on Create Launch configuration and select the AMI image (created image), security groups and key-pair login.

4. Click on advance settings and give the below cmds in user data as shown below:

```
#!/bin/bash
sudo systemctl stop jenkins
rm -rf /var/lib/jenkins
mkdir /var/lib/jenkins
sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-01d6e34d4fcfa691a.efs.us-east-1.amazonaws.com:/ /var/lib/jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
```



RAM disk ID Info

Use default

Metadata accessible Info

Don't include in launch configuration

Metadata version Info

Don't include in launch configuration

Metadata response hop limit Info

Don't include in launch configuration

User data Info

[✓ As text]

As file

```
#!/bin/bash
sudo systemctl stop jenkins
rm -rf /var/lib/jenkins
```

Input is already base64 encoded

IP address type Info

[✓ Only assign a public IP address to instances launched in a subnet with auto-assign public IP enabled (default)]

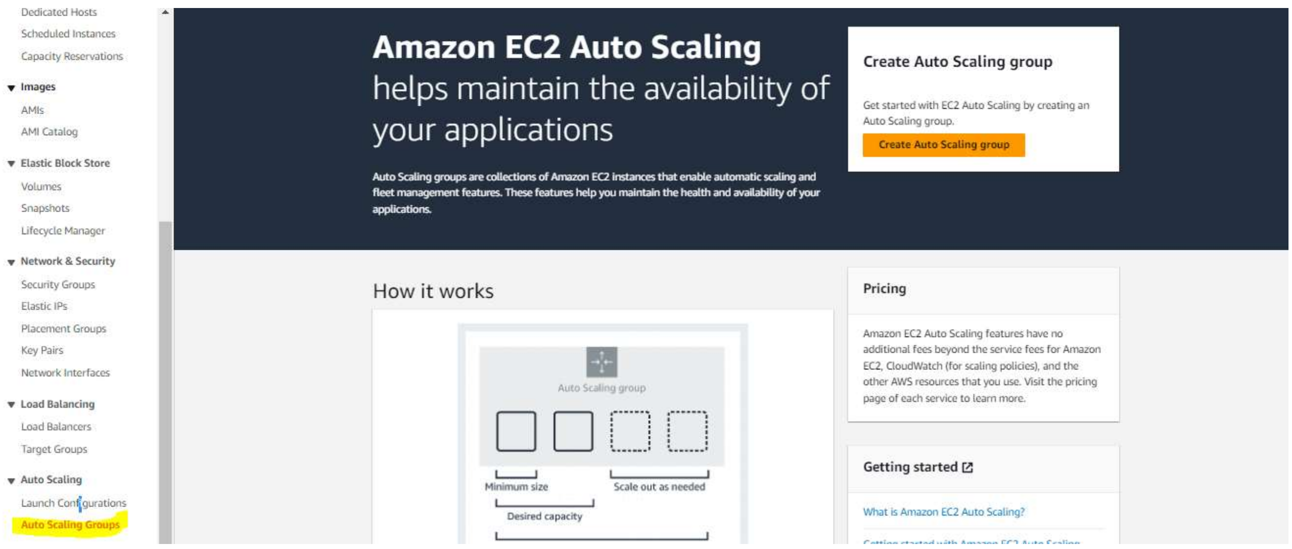
Assign a public IP address to every instance.

Do not assign a public IP address to any instances.
Note: this option only affects instances launched into an Amazon VPC

5. click on Launch configuration. Now we are ready with Launch configuration.

Creating Auto-Scaling Group :

1. Go to EC2 console, and click Autoscaling group as shown below:



Amazon EC2 Auto Scaling
helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

How it works

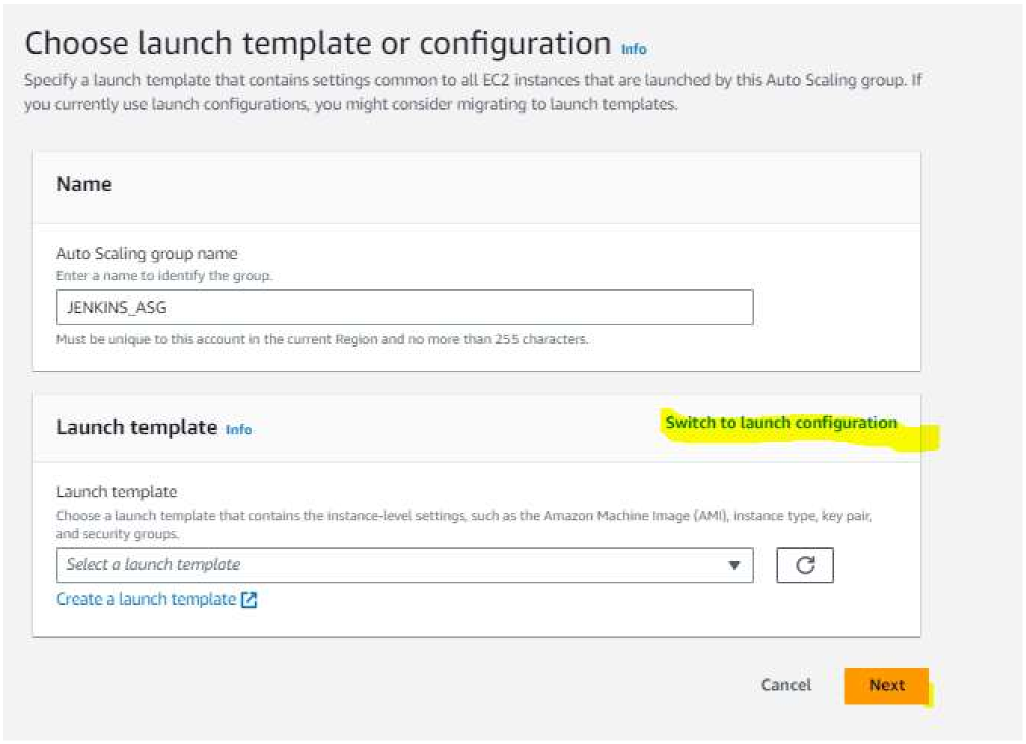
Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started

What is Amazon EC2 Auto Scaling?

2. Give name for ASG and Click on switch to Launch configuration and select the previous configured launch configuration as shown below:



Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name

Enter a name to identify the group.

JENKINS_ASG

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

Switch to launch configuration

Launch template

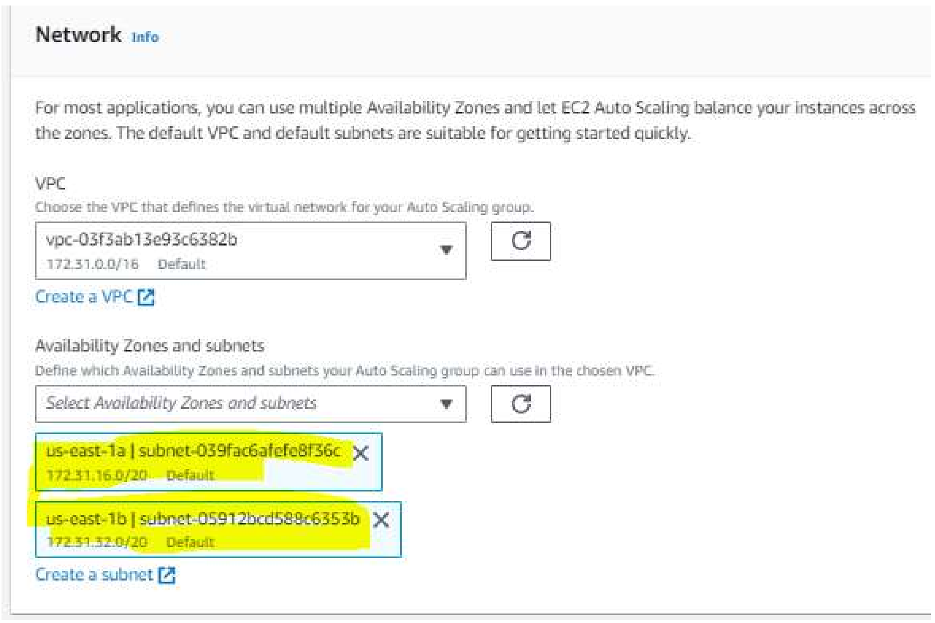
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Select a launch template

Create a launch template

Cancel Next

3. Select the VPC and Subnet regions on which you need to run these jenkins servers. I have selected two regions as shown below:



Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-03f3ab13e93c6382b
172.31.0.0/16 Default

Create a VPC

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

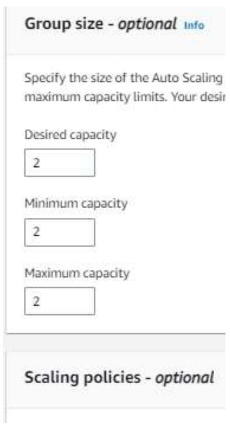
Select Availability Zones and subnets

us-east-1a | subnet-039fac6afefe8f36c
172.31.16.0/20 Default

us-east-1b | subnet-05912bcd588c6353b
172.31.32.0/20 Default

Create a subnet

4. Give instances group size as shown below:


Group size - *optional* Info

Specify the size of the Auto Scaling
maximum capacity limits. Your desi

Desired capacity
2

Minimum capacity
2

Maximum capacity
2

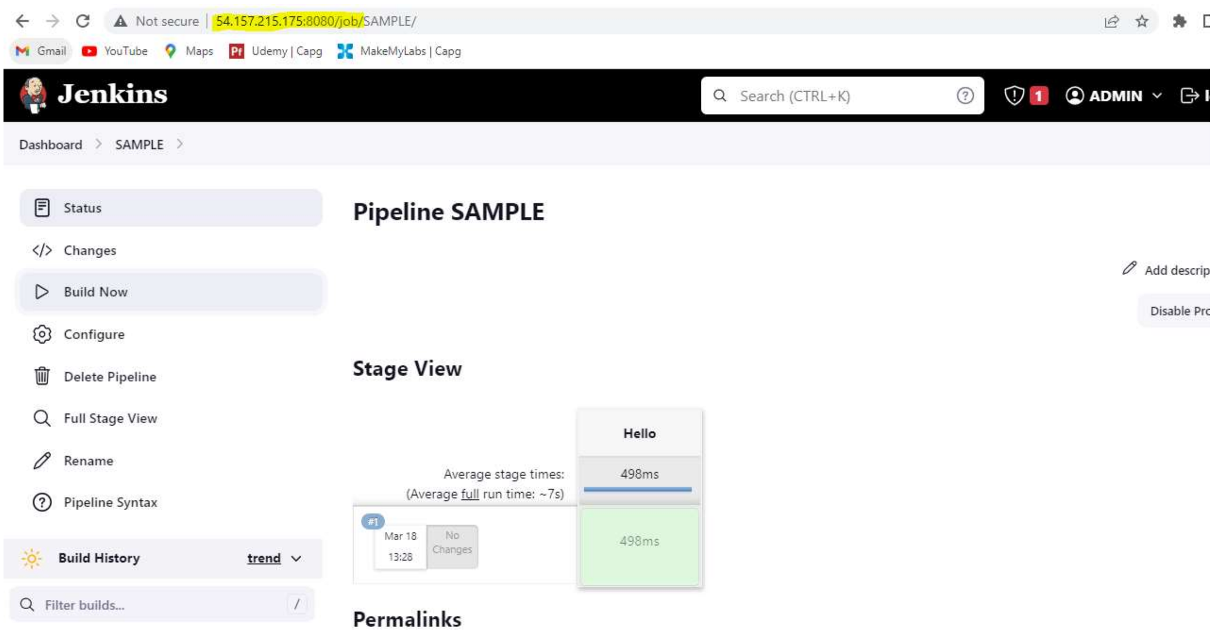
Scaling policies - *optional*

5. Leaving all the options as default, Click Next continously and Launch ASG.
6. It will take time to launch jenkins servers. Go to EC2 console and wait for instances.

Lets Open both server jenkins with their public ip:

1. Go to EC2 console, you will find your EC2 instances there and copy the public ip and open jenkins using http://PUBLIC_IP:8080/

2. create a job and run in first jenkins as shown below:


Not secure | 54.157.215.175:8080/job/SAMPLE/

Pipeline SAMPLE

Stage View

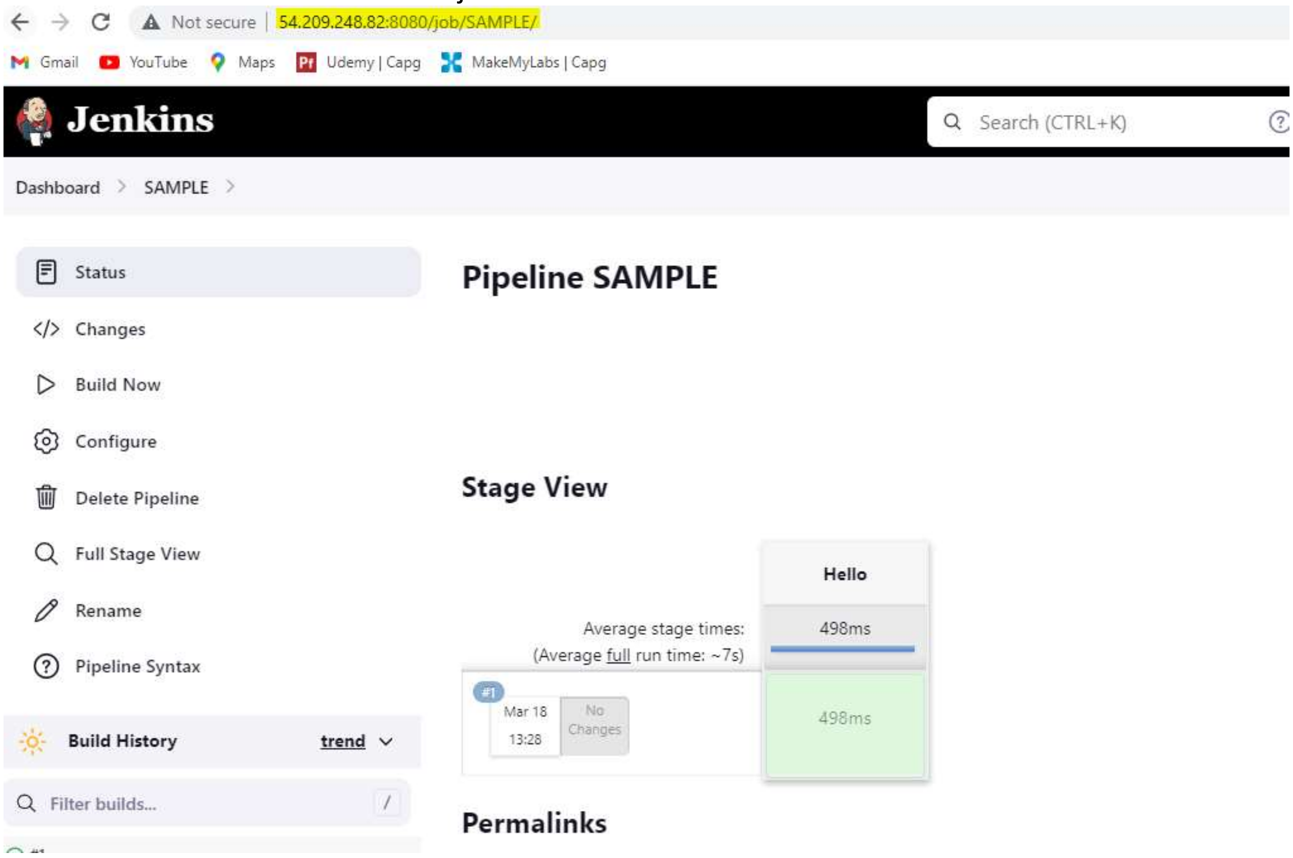
Hello

Average stage times: 498ms
(Average full run time: ~7s)

#1 Mar 18 13:28 No Changes 498ms

Permalinks

3. You will be able to see the same job in the second server also as shown below:

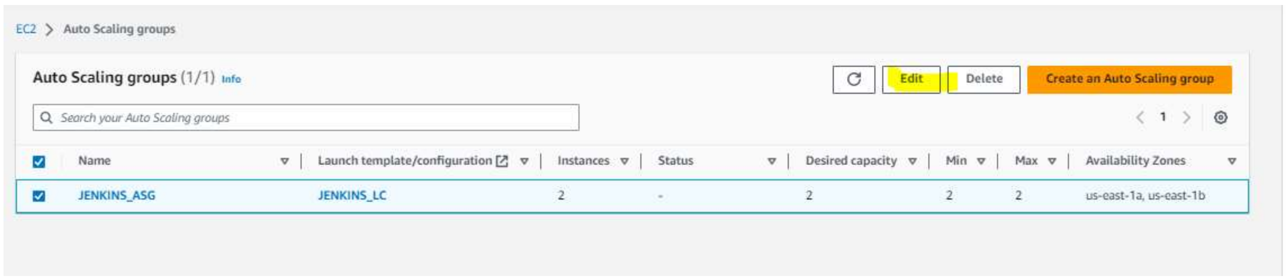


4. This sync will be done by the cronjob that we have configured previously.

Attaching LoadBalancer:

1. Go to EC2 console and click on the Autoscaling group.

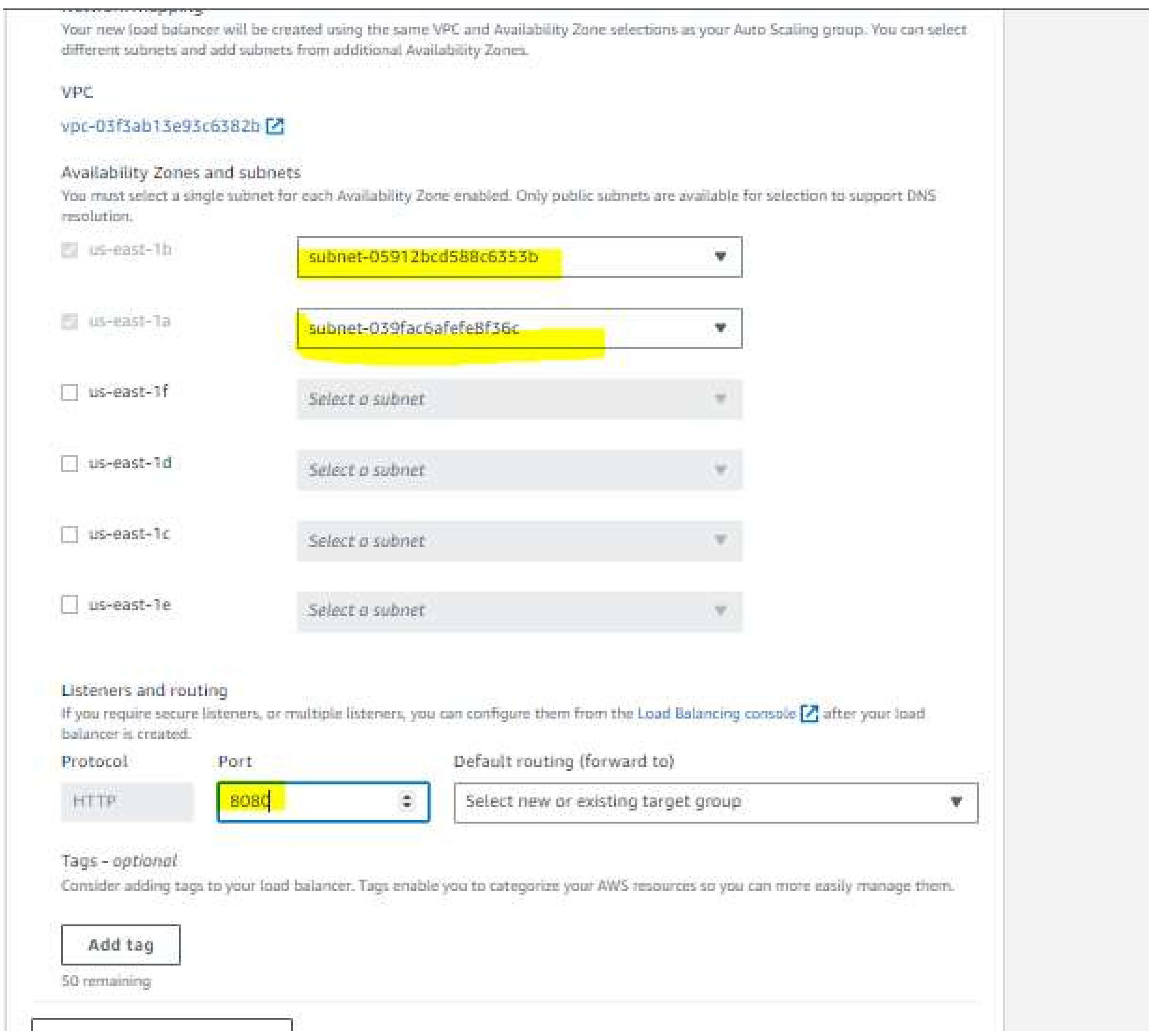
2. Select the Autoscaling group and click on edit as shown below:


EC2
Auto Scaling groups
Auto Scaling groups (1/1) Info
Edit
Delete
Create an Auto Scaling group
Search your Auto Scaling groups
1

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
JENKINS_ASG	JENKINS_LC	2	-	2	2	2	us-east-1a, us-east-1b

3. Go to Load balancer options. Click on Add new load balancer.

4. Give loadbalancer name, select internet-facing, select VPC, subnets, port as 8080 as shown below:



Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
vpc-03f3ab13e93c6382b

Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

✓ us-east-1b	subnet-05912bcd588c6353b
✓ us-east-1a	subnet-039fac6afefe8f36c
☐ us-east-1f	Select a subnet
☐ us-east-1d	Select a subnet
☐ us-east-1c	Select a subnet
☐ us-east-1e	Select a subnet

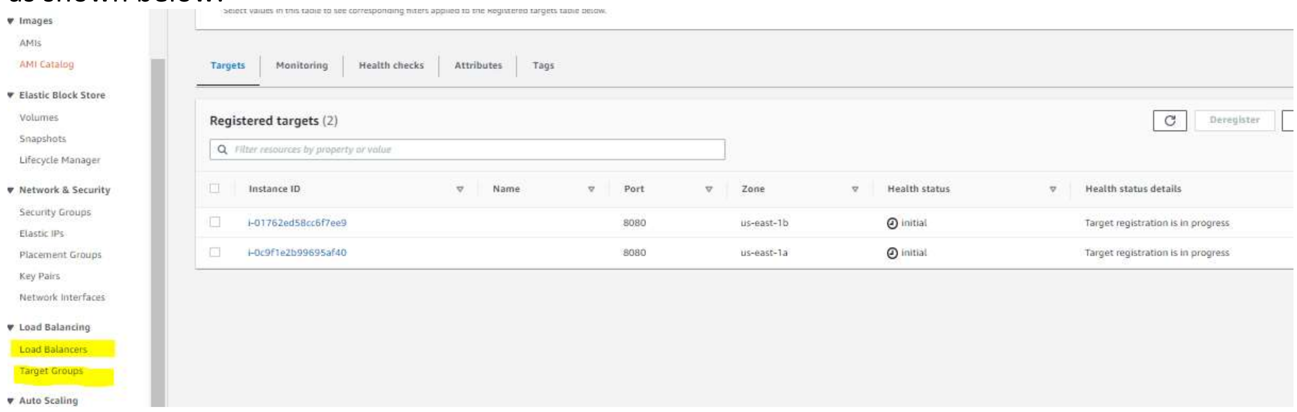
Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the Load Balancing console after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	8080	Select new or existing target group

Tags - *optional*
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

5. Click on default routing drop down and click on create a new target group. Give a name to target group and Click on Update.

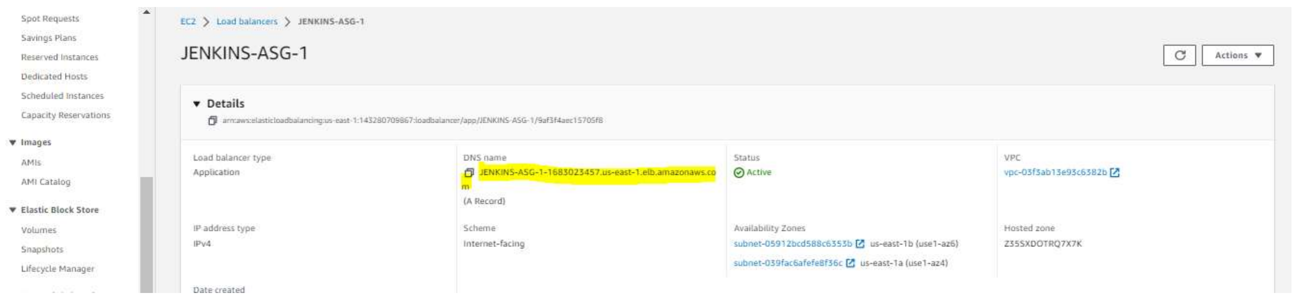
6. Wait for some time and Check the load balancer status and Target groups by clicking on the tabs as shown below:



Registered targets (2)

Instance ID	Name	Port	Zone	Health status	Health status details
i-01762ed58cc6f7ee9		8080	us-east-1b	initial	Target registration is in progress
i-0c9f1e2b99695af40		8080	us-east-1a	initial	Target registration is in progress

7. Once they are ready, Go to load balancer and copy the url as shown below:



EC2 > Load balancers > JENKINS-ASG-1

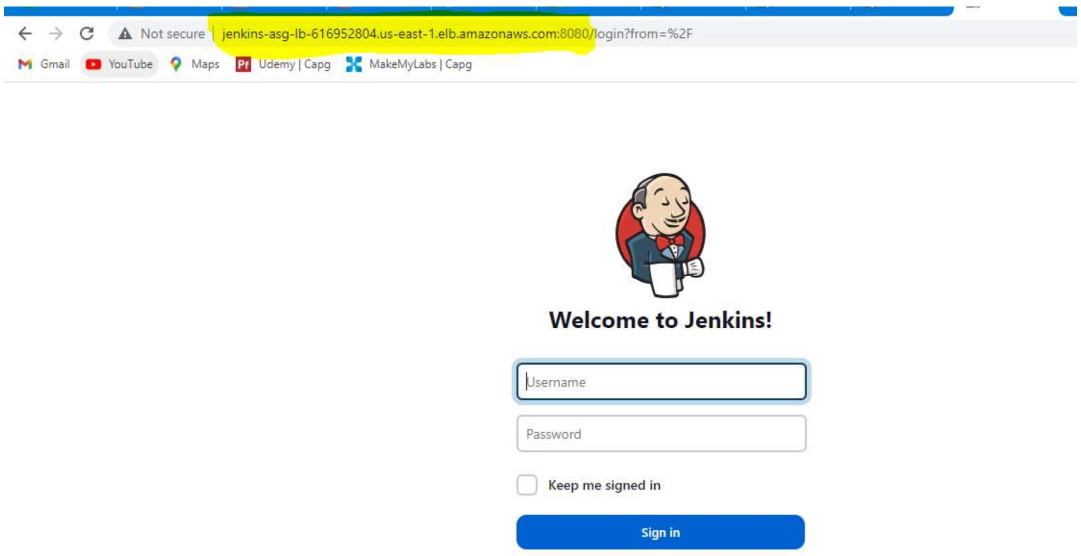
JENKINS-ASG-1

Details

arn:aws:elasticloadbalancing:us-east-1:143280709867:loadbalancer/app/JENKINS-ASG-1/9af3f4aec15705f8

Load balancer type	DNS name	Status	VPC
Application	JENKINS-ASG-1-1683023457.us-east-1.elb.amazonaws.com (A Record)	Active	vpc-03f3ab13e93c6382b
IP address type	Scheme	Availability Zones	Hosted zone
IPv4	Internet-facing	subnet-05912bcd588c6353b us-east-1b (use1-az6) subnet-039fac6afefe8f36c us-east-1a (use1-az4)	Z35SXDOTRQ7X7K
Date created			

8. Open in a new tab as http://LOAD_BALANCER_URL:8080/, You will be at jenkins login page as shown below:



Not secure | jenkins-asg-lb-616952804.us-east-1.elb.amazonaws.com:8080/login?from=%2F

Finally, we have launched our jenkins with autoscaling group. If one of the jenkins server is down, ELB will serve the requests from other available jenkins server. All the jobs run/configuration changes in one server will be synced with other services with the cronjob that we have configured.

DELETE ALL RESOURCES CREATED IN AWS.

- Make all the below resouces are deleted.
- Delete ASG group will delete EC2 instances also.
- Delete Load balancer
- Delete Target groups
- Delete EFS
- Delete AMI
- Delete EBS snapshot
- Delete security groups
- Delete key-pair login