CS457

DevOps Final Assignment -1

Submitted to:

Dr. Uma S

Submitted by:

S Namratha
(18BCS083)

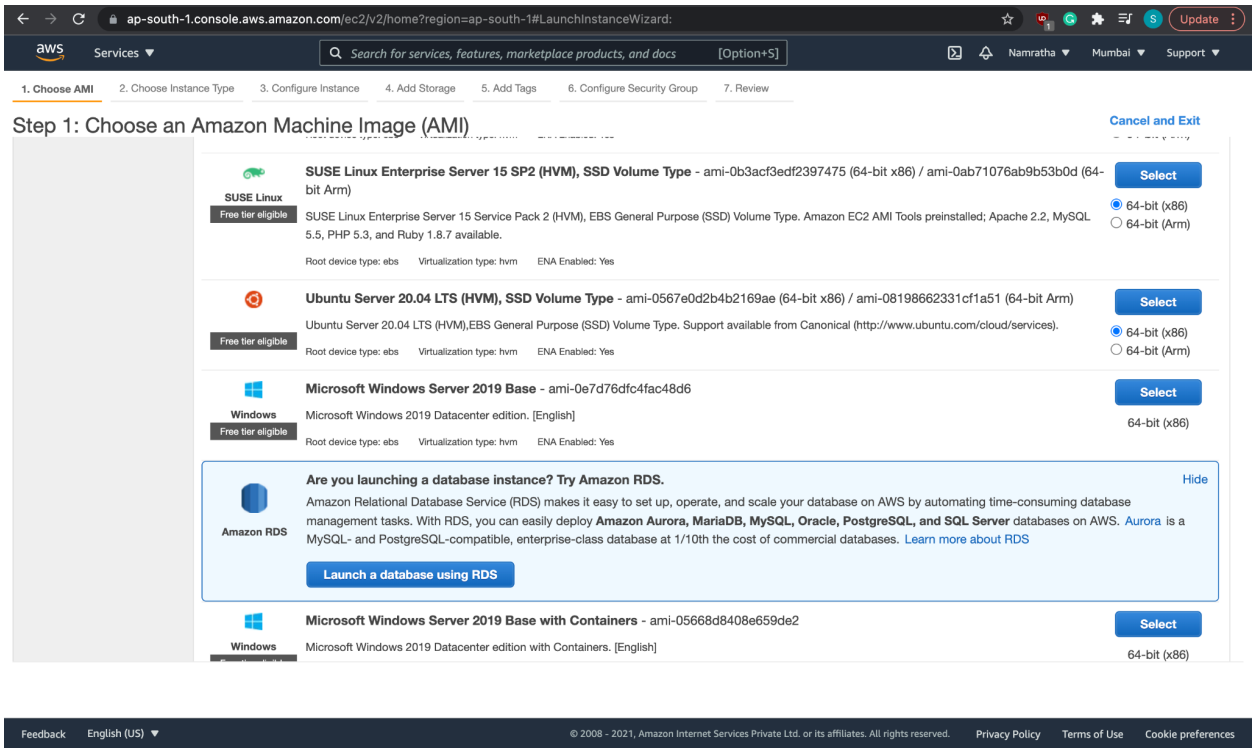
DevOps Assignment 1

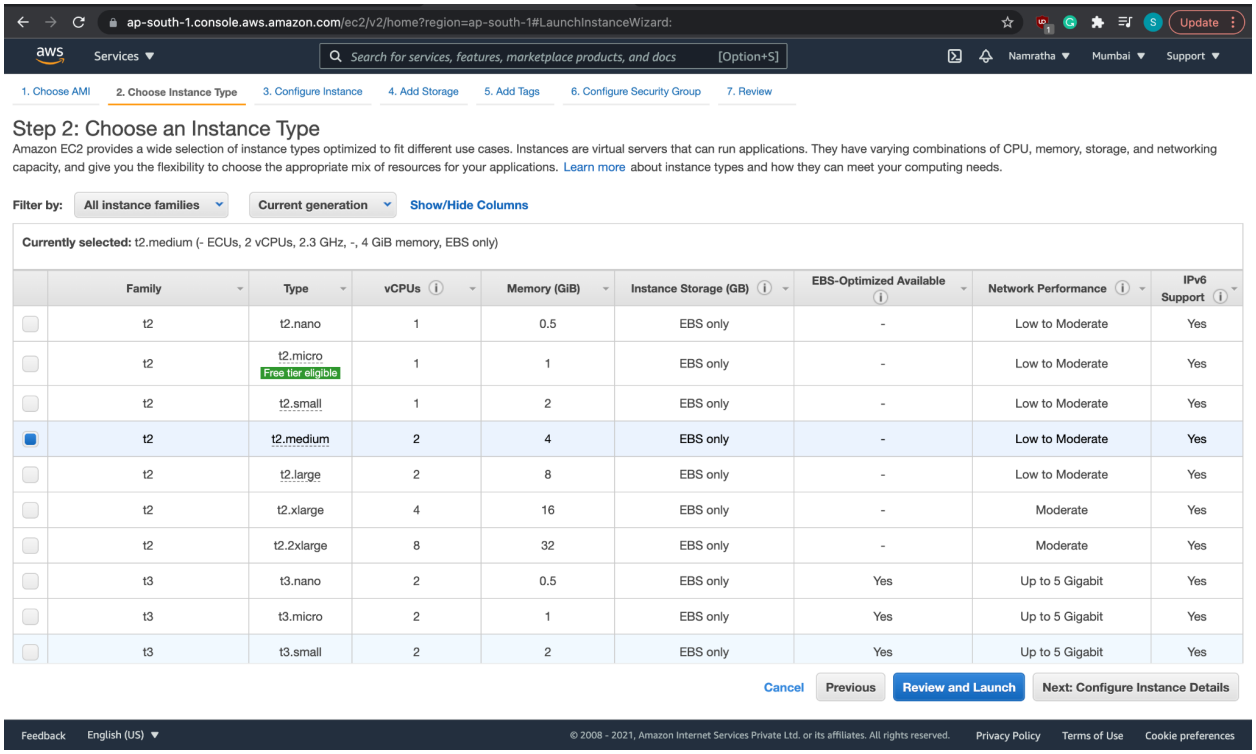
Set up complete CI/CD jenkins pipeline for kubernetes (AWS)

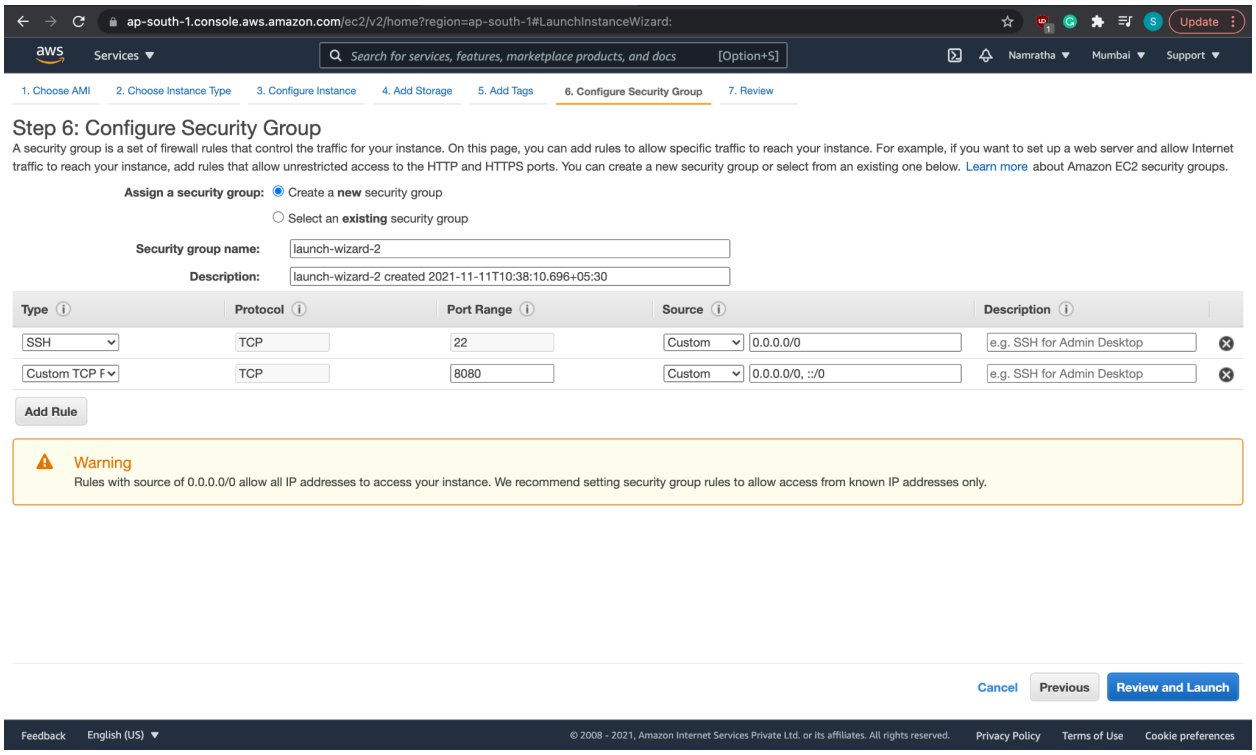
Note- Github repo containing files of this assignment : https://github.com/Namratha-shivaraju/k8s-jenkins-aws

1. Set up and connect an AWS EC2 instance

   Select Ubuntu server



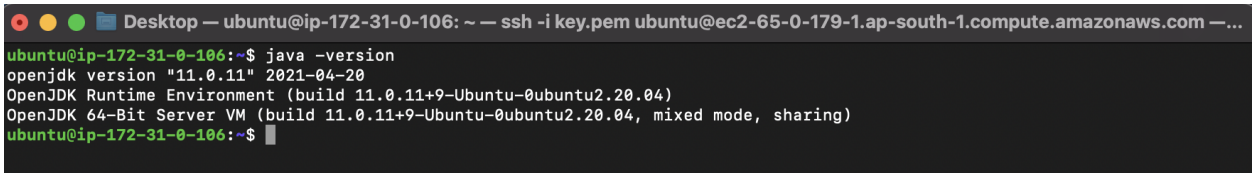




2. Install JDK on AWS EC2 Instance

Run the below commands to install JDK

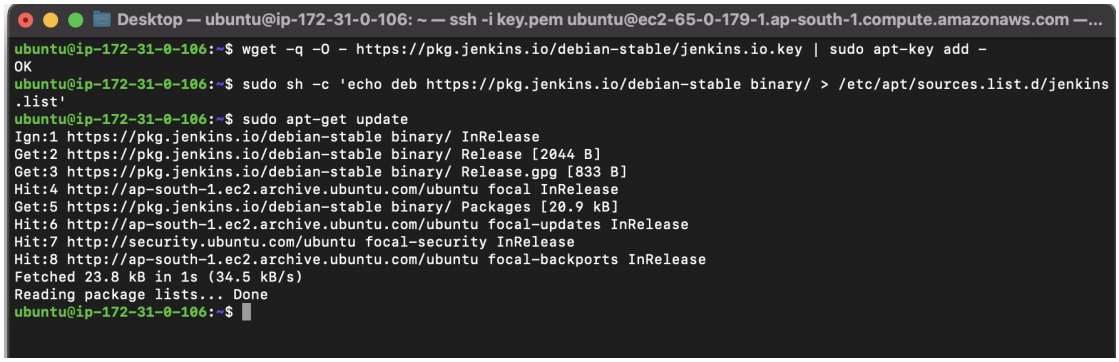
```
sudo apt-get update

sudo apt install openjdk-11-jre-headless

java -version
```

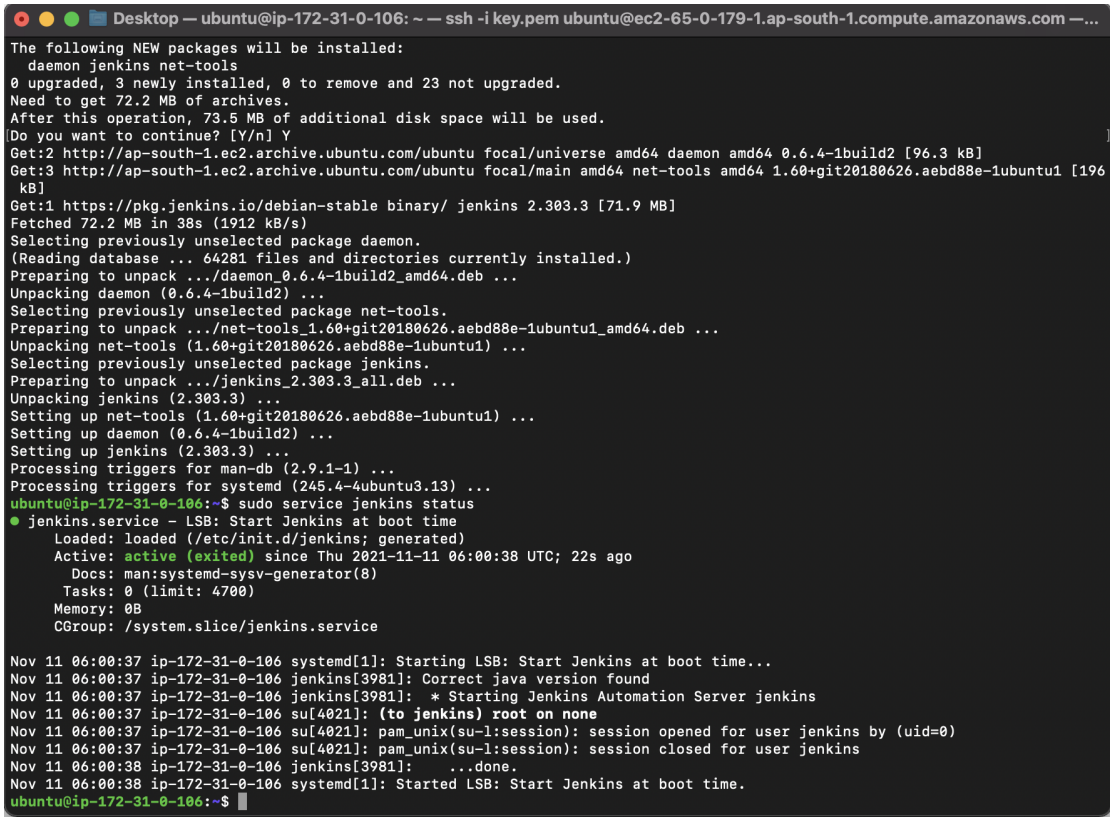


3. Install and Setup Jenkins

Run the below commands to install Jenkins

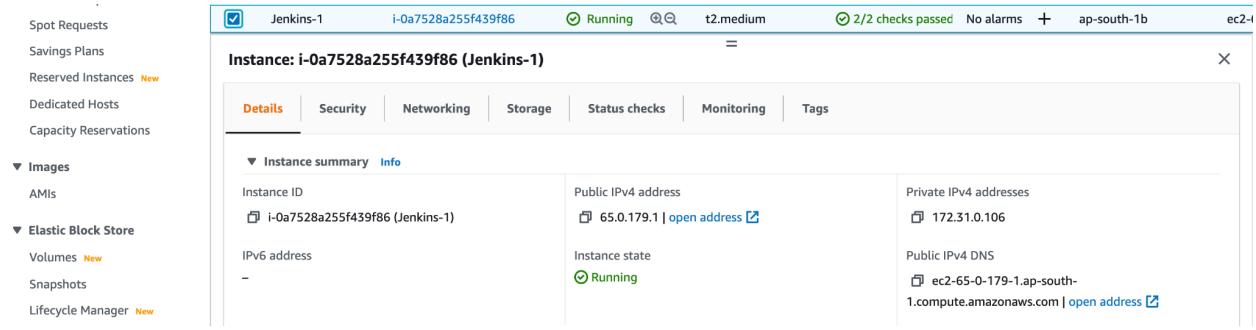
```
wget -q -O -
https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo
apt-key add -

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable
binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins

sudo service jenkins status
```

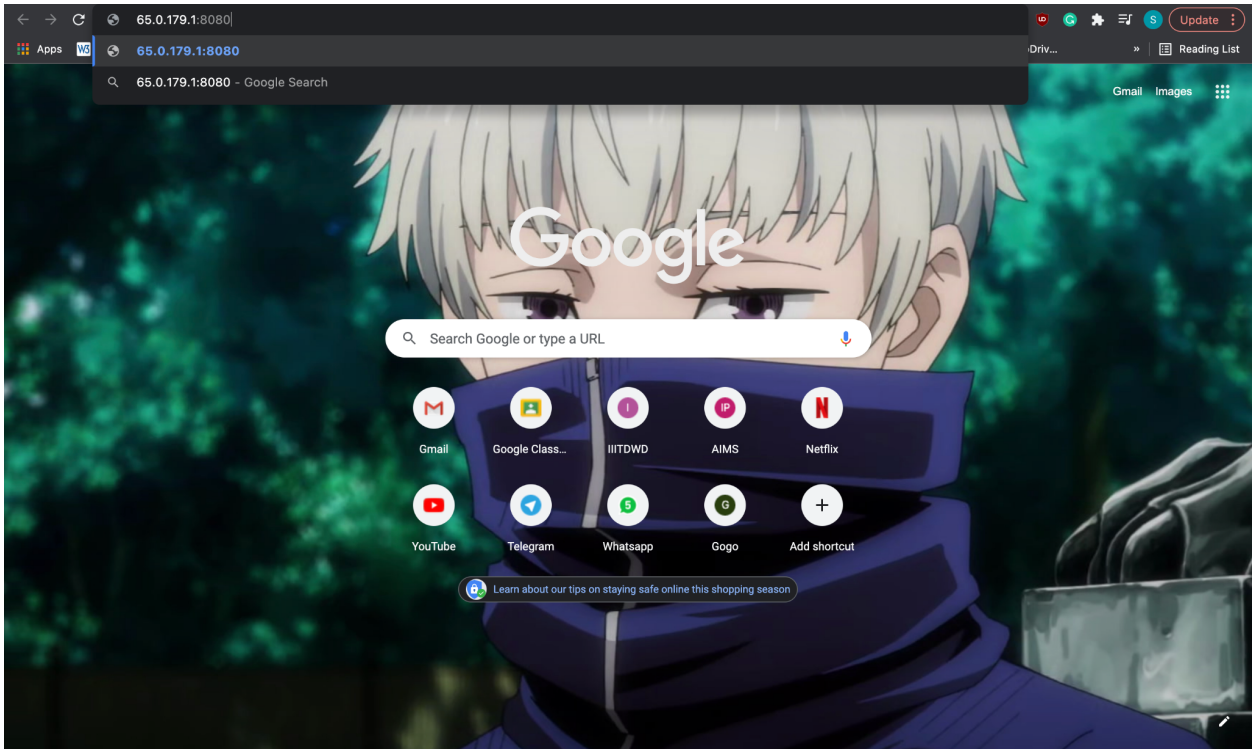


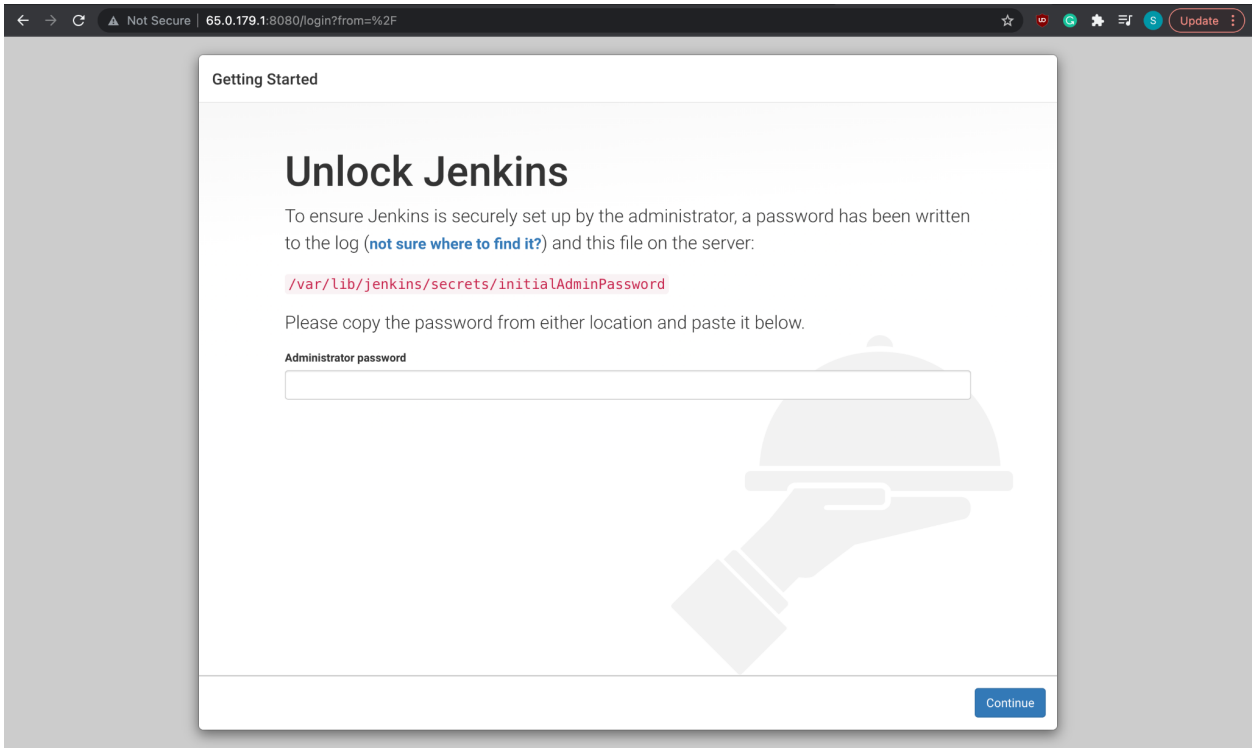


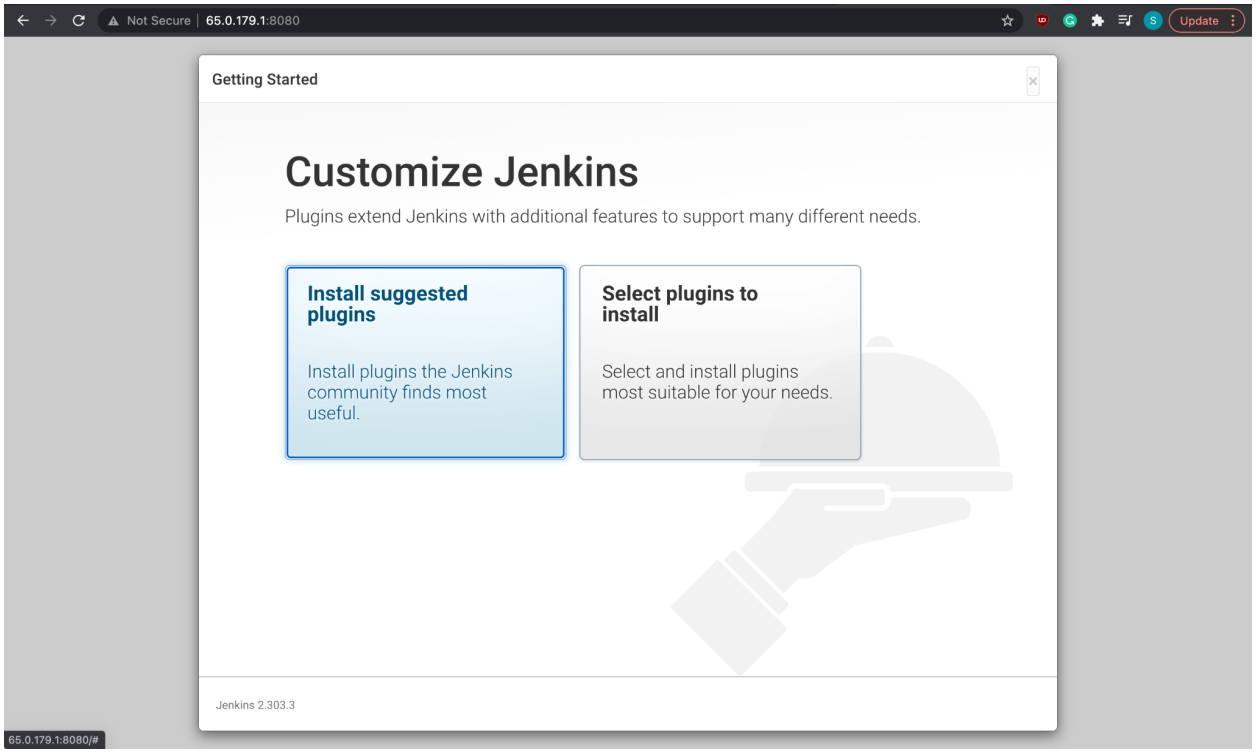
Let’s verify if jenkins is installed.

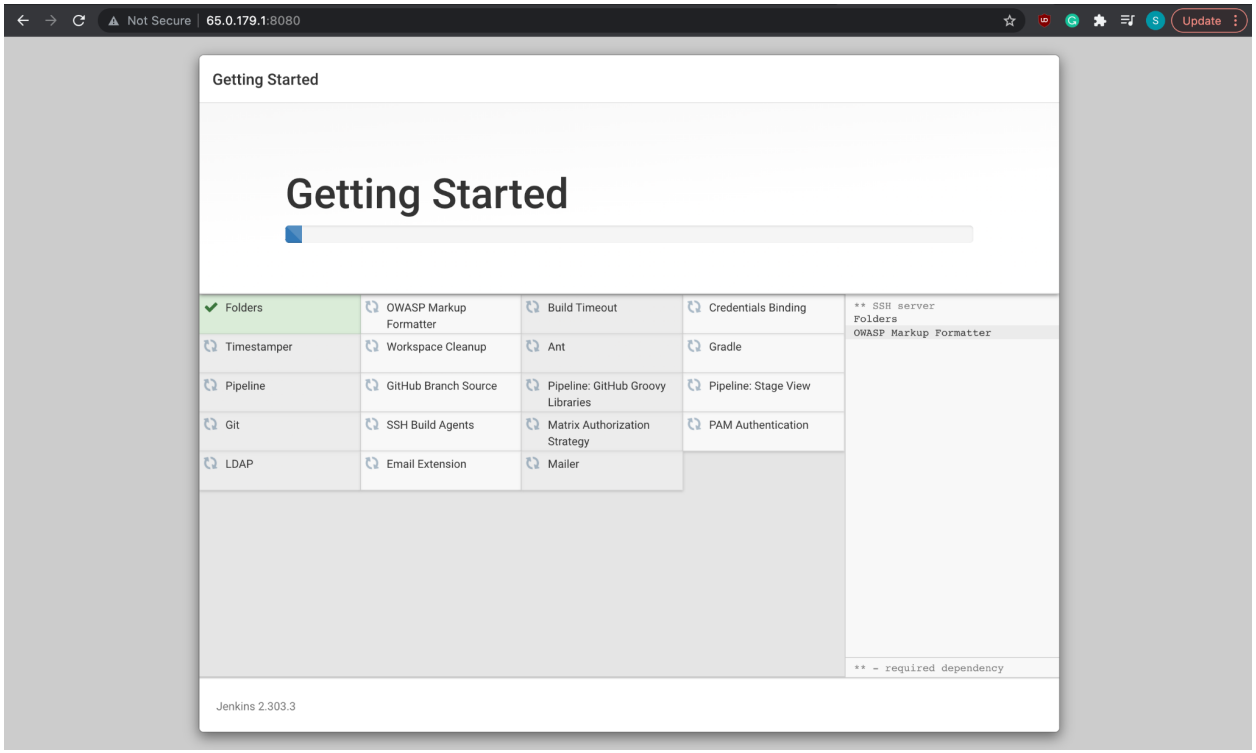
Go to EC2 instance -> copy public IP address -> open a new tab -> type “ ip_add:8080 and enter.

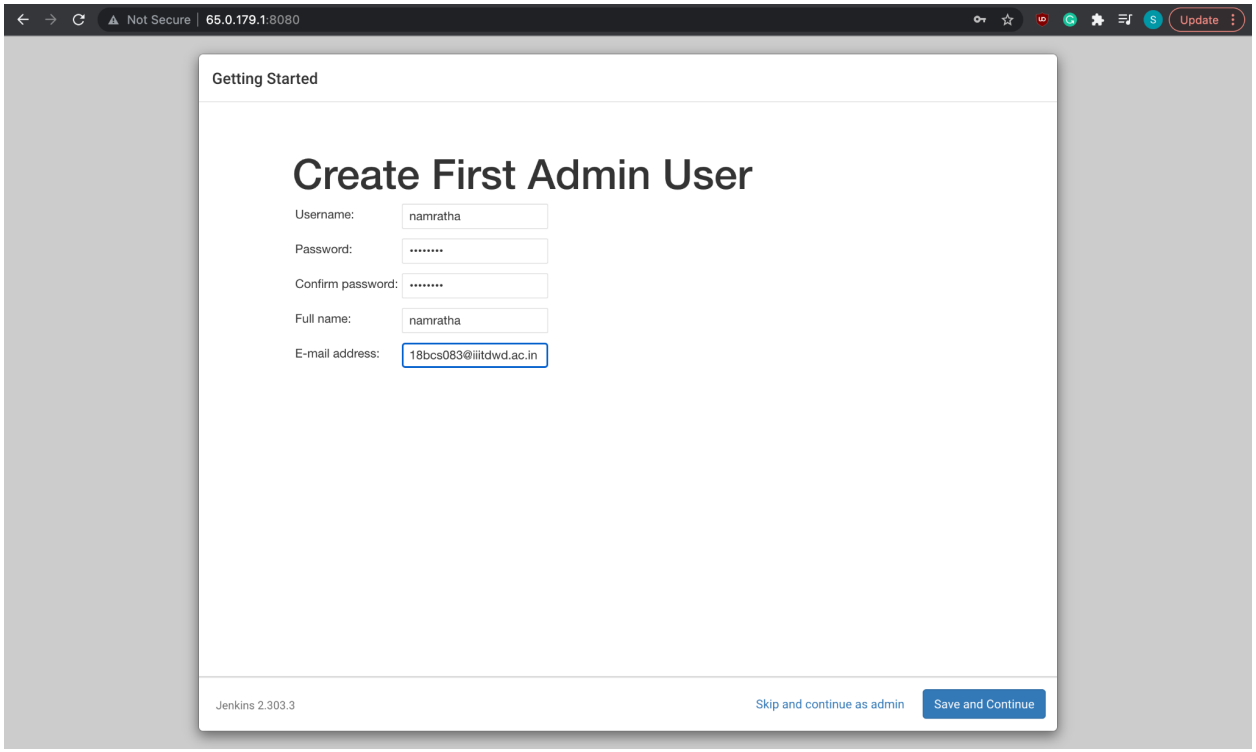

Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes New
Snapshots
Lifecycle Manager New
Jenkins-1
i-0a7528a255f439f86
Running
t2.medium
2/2 checks passed
No alarms
ap-south-1b
Instance: i-0a7528a255f439f86 (Jenkins-1)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0a7528a255f439f86 (Jenkins-1)
Public IPv4 address
65.0.179.1 | open address
Private IPv4 addresses
172.31.0.106
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-65-0-179-1.ap-south-1.compute.amazonaws.com | open address

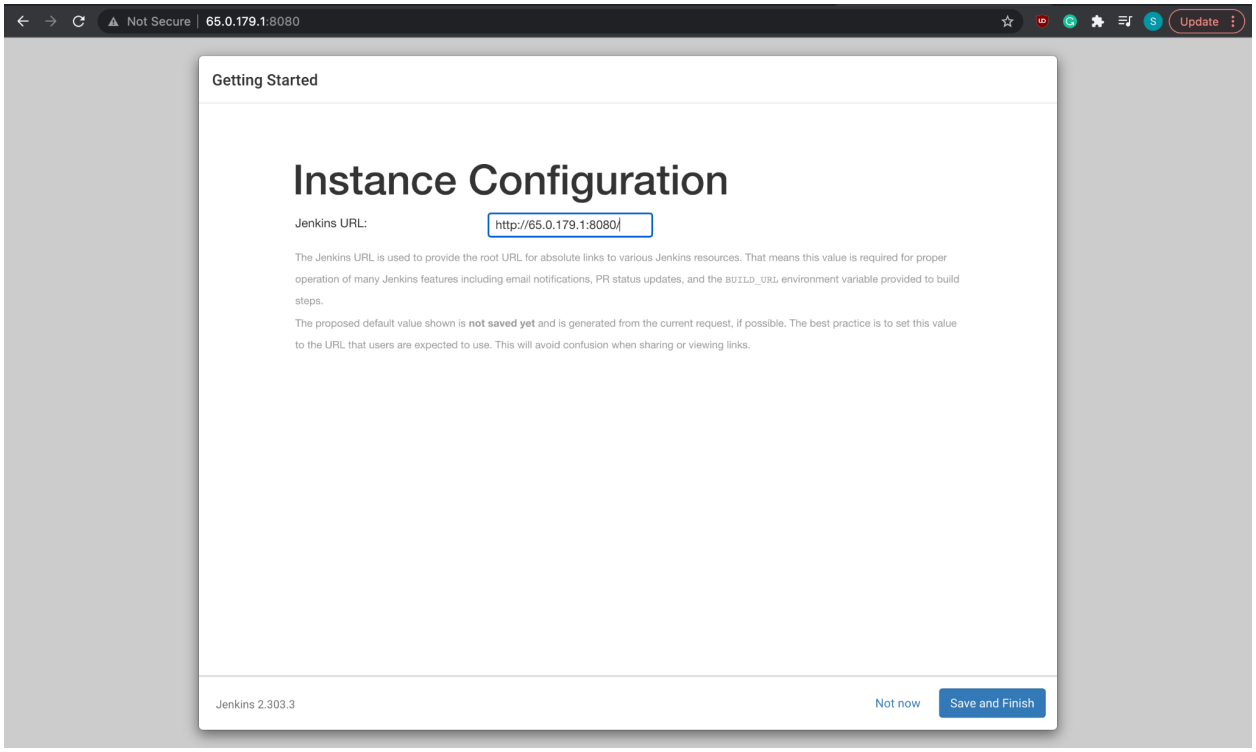

65.0.179.1:8080
Apps
65.0.179.1:8080
65.0.179.1:8080 - Google Search
Update
Reading List
Gmail
Images
Google
Search Google or type a URL
Gmail
Google Class...
IIITDWD
AIMS
Netflix
YouTube
Telegram
Whatsapp
Gogo
Add shortcut
Learn about our tips on staying safe online this shopping season


Not Secure | 65.0.179.1:8080/login?from=%2F
Update
Getting Started
Unlock Jenkins
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:
/var/lib/jenkins/secrets/initialAdminPassword
Please copy the password from either location and paste it below.
Administrator password
Continue

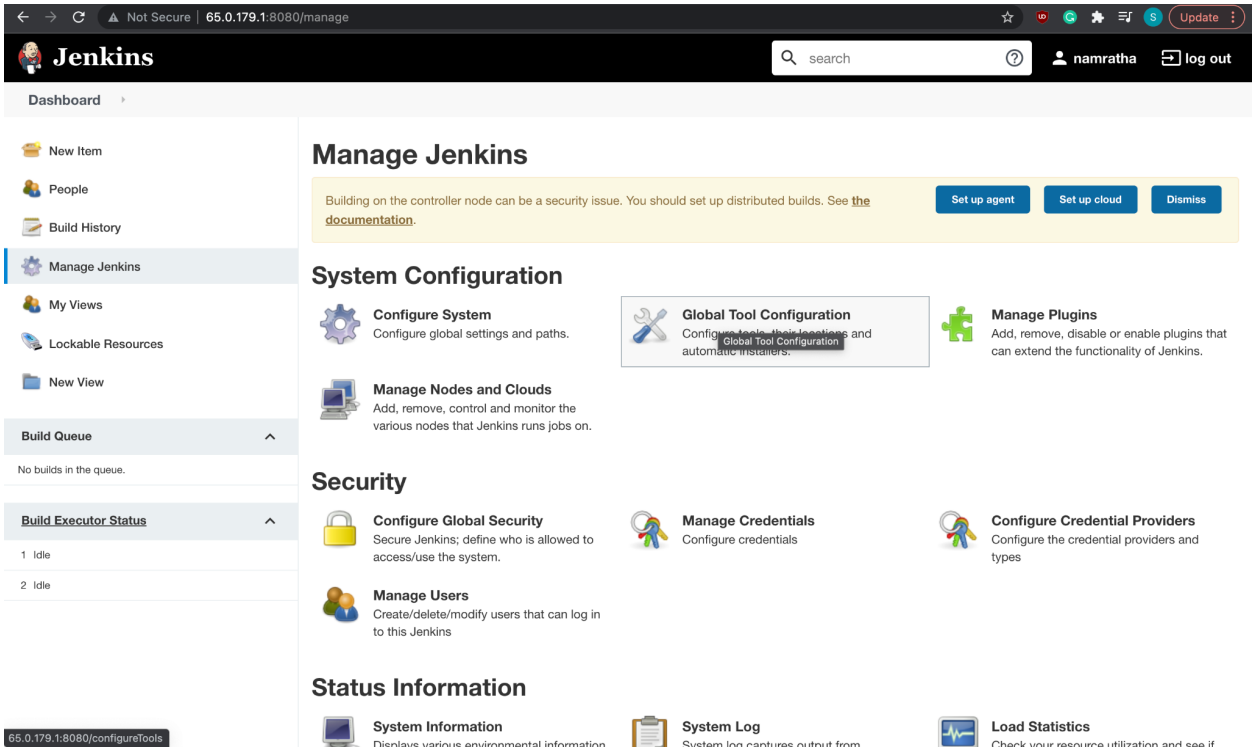

Not Secure | 65.0.179.1:8080
Update
Getting Started
Customize Jenkins
Plugins extend Jenkins with additional features to support many different needs.
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.
Jenkins 2.303.3
65.0.179.1:8080/#

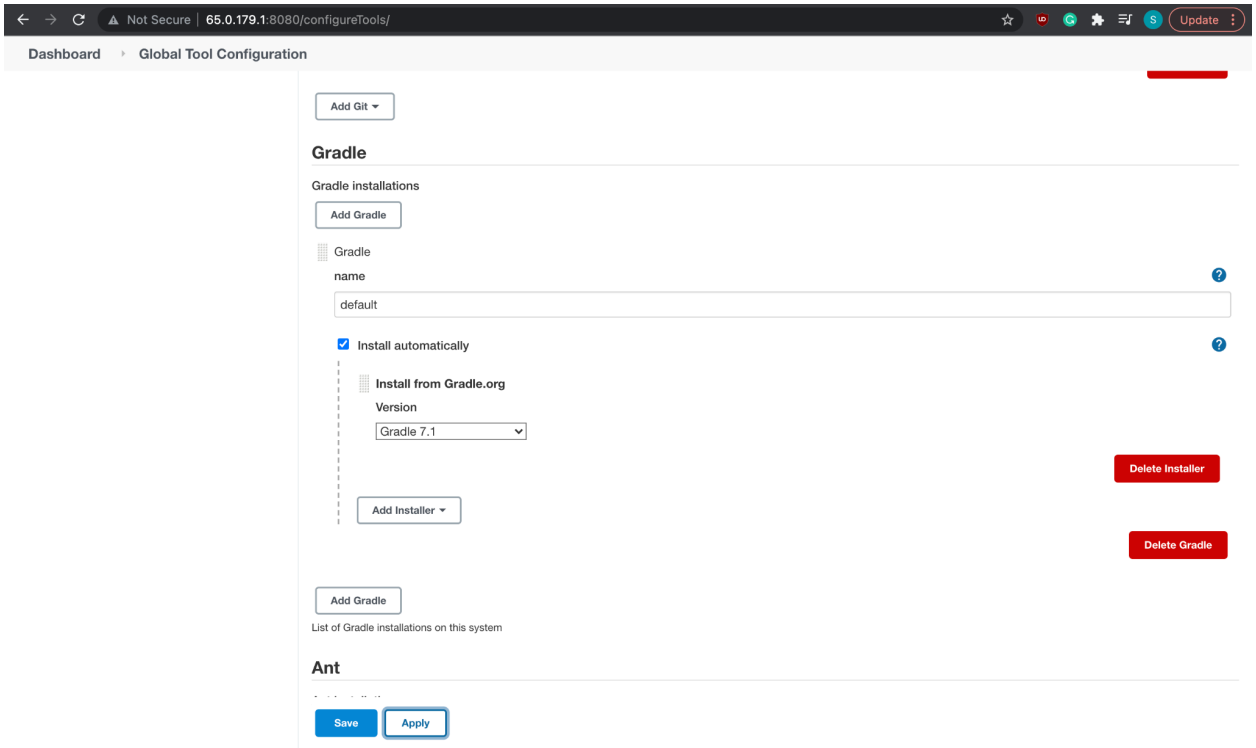

Not Secure | 65.0.179.1:8080
Update
Getting Started
Getting Started
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline: Stage View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
** SSH server
Folders
OWASP Markup Formatter
** - required dependency
Jenkins 2.303.3


Not Secure | 65.0.179.1:8080
Update
Getting Started
Create First Admin User
Username:
namratha
Password:
Confirm password:
Full name:
namratha
E-mail address:
18bcs083@iiitdwd.ac.in
Jenkins 2.303.3
Skip and continue as admin
Save and Continue



Goto -> Manage Jenkins -> Global Tool Configuration -> Gradle





4. Update visudo and assign administration privileges to jenkins use

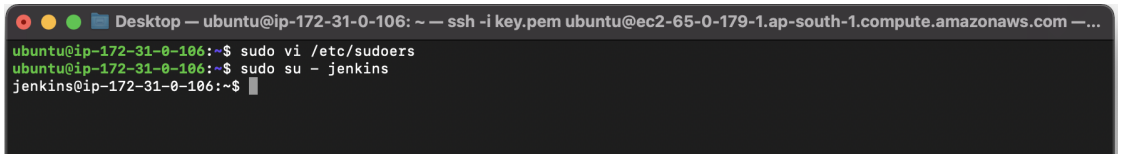
Add jenkins user as administrator - This step is not necessary but it avoids the process of entering password every time the pipeline runs.

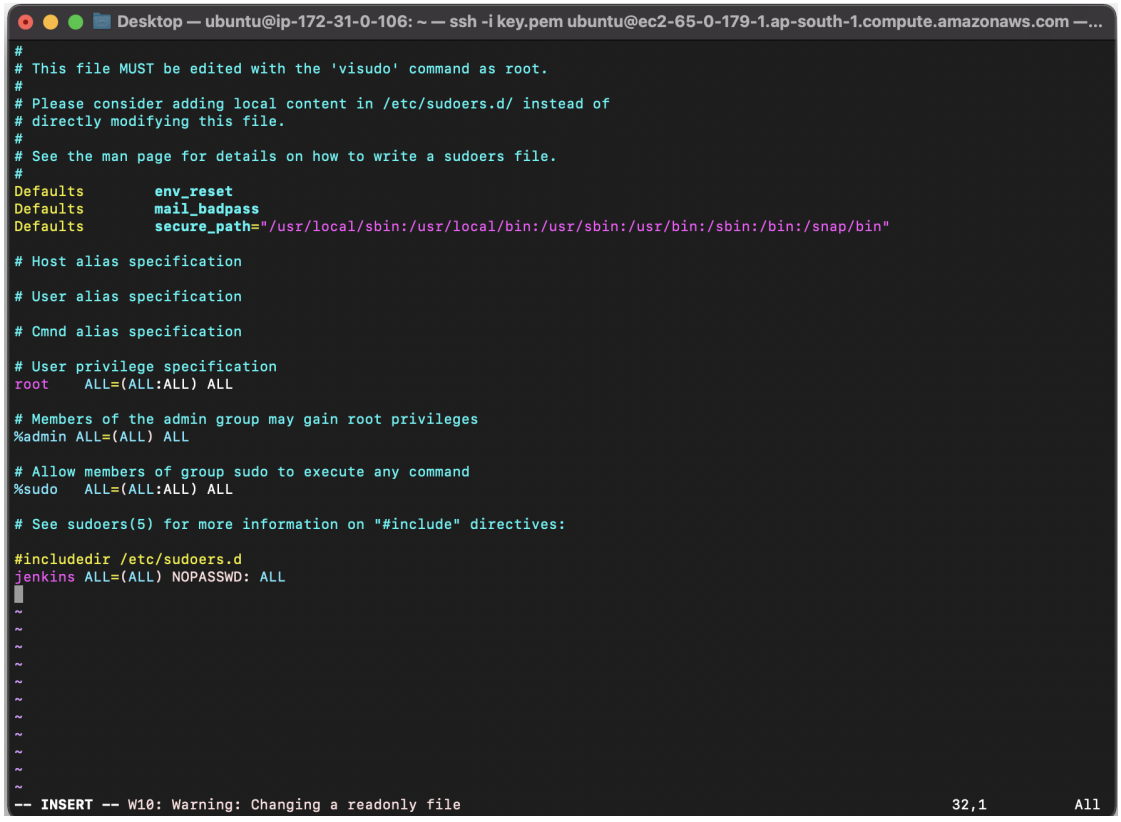
```
sudo vi /etc/sudoers
```

```
jenkins ALL=(ALL) NOPASSWD: ALL
```

To use jenkins as root user, run the following

```
sudo su - jenkins
```





5. Install Docker

Run the below commands to install Docker

```
sudo apt install docker.io
```

```
docker --version
```

To add jenkins user to Docker group, run the following

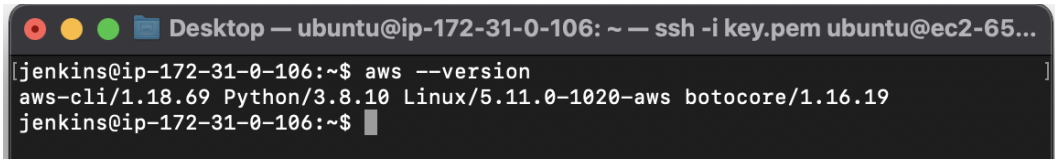
```
sudo usermod -aG docker jenkins
```

6. Install and Setup AWS CLI

To use eksctl we install AWS CLI.
Run the below commands to install AWS CLI

```
sudo apt install awscli
```

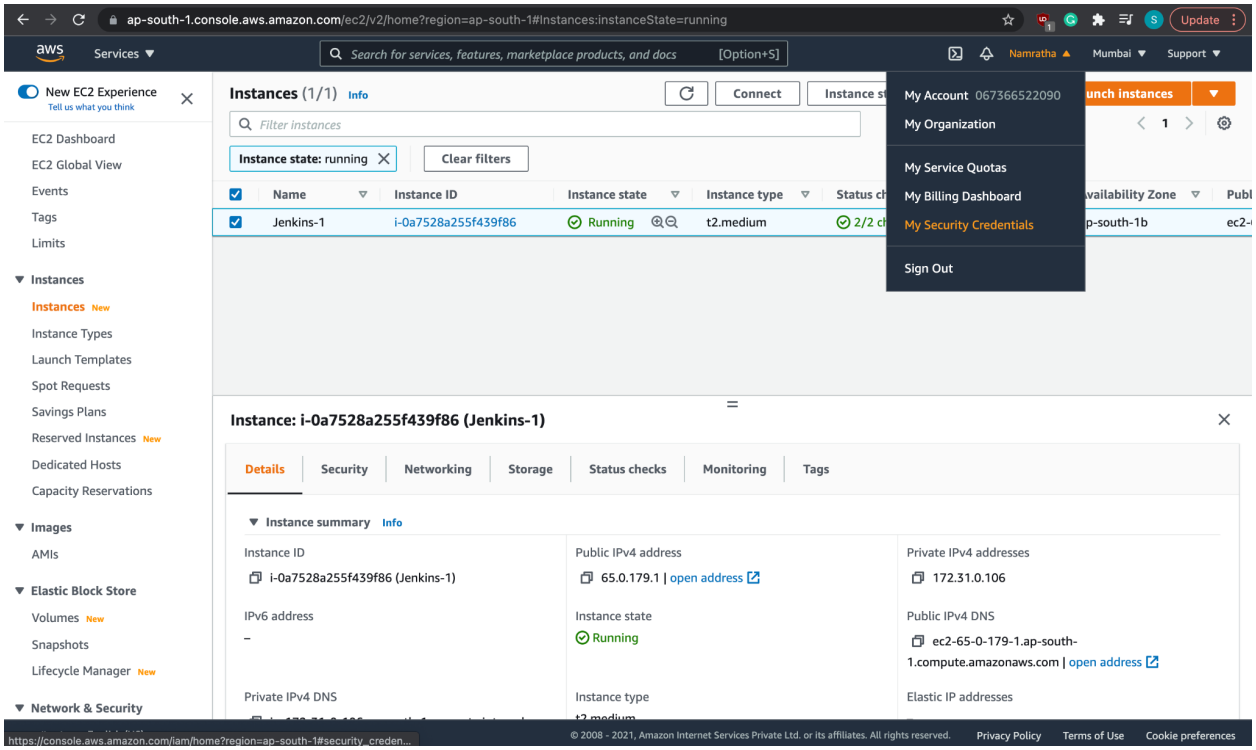
```
aws --version
```

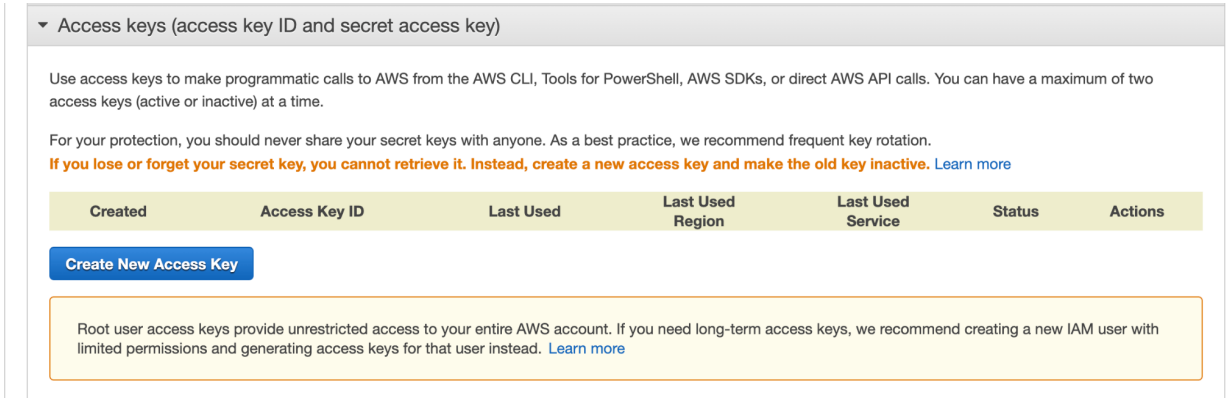


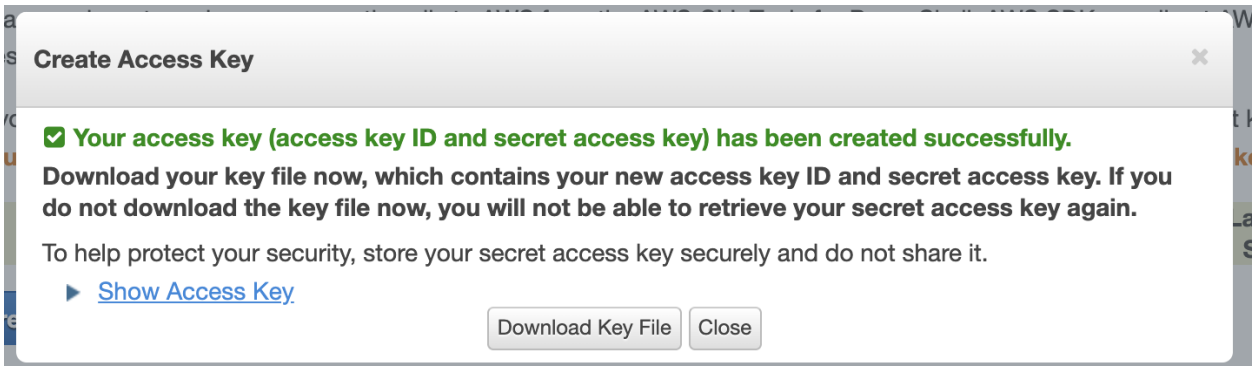
7. Configure AWS CLI

Configure the AWS CLI to authenticate and communicate with AWS environment.

Go to AWS -> My Security Credentials -> Access keys -> Create New Access Key.


ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:instanceState=running
Update
aws
Services
Search for services, features, marketplace products, and docs [Option+S]
Namratha
Mumbai
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes New
Snapshots
Lifecycle Manager New
Network & Security
Instances (1/1) Info
Connect
Filter instances
Instance state: running
Clear filters
Name
Instance ID
Instance state
Instance type
Jenkins-1
i-0a7528a255f439f86
Running
t2.medium
My Account 067366522090
My Organization
My Service Quotas
My Billing Dashboard
My Security Credentials
Sign Out
Instance: i-0a7528a255f439f86 (Jenkins-1)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0a7528a255f439f86 (Jenkins-1)
Public IPv4 address
65.0.179.1 | open address
Private IPv4 addresses
172.31.0.106
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-65-0-179-1.ap-south-1.compute.amazonaws.com | open address
Private IPv4 DNS
Instance type
Elastic IP addresses
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

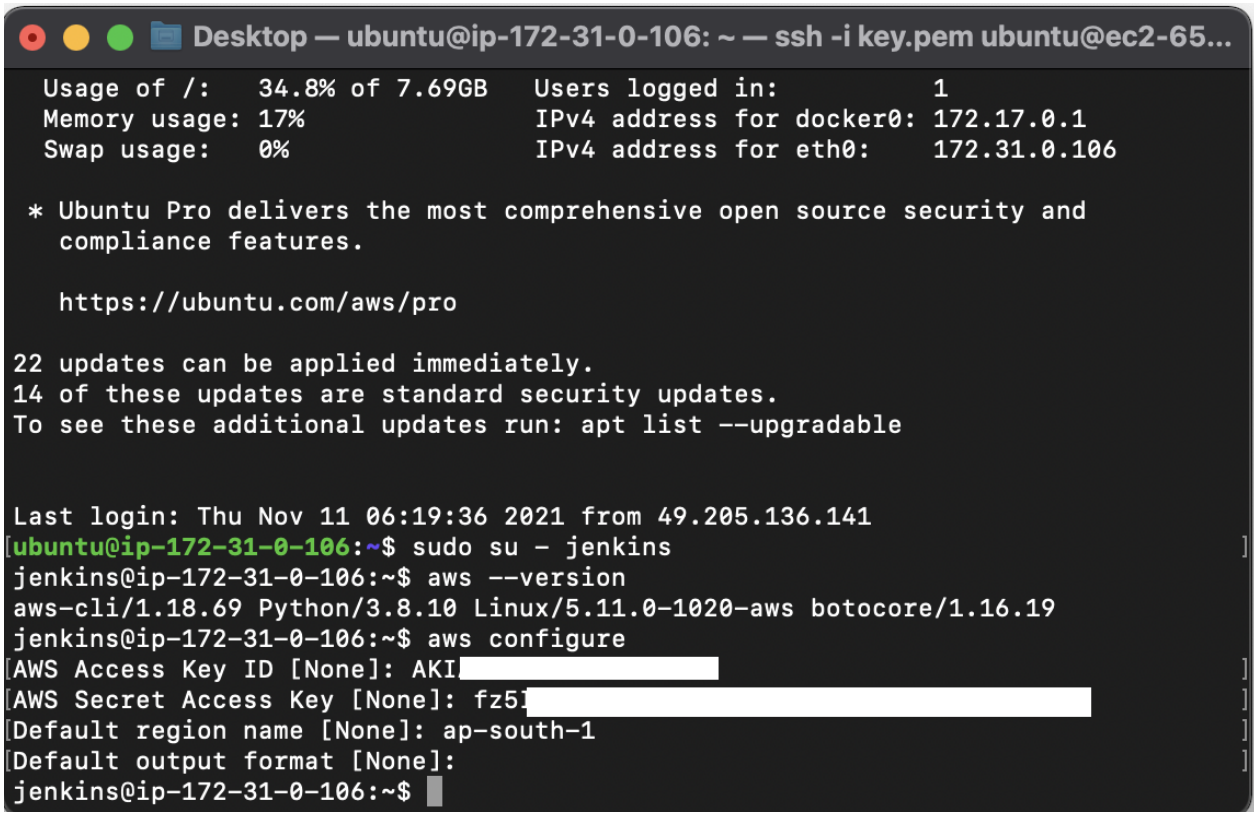

Access keys (access key ID and secret access key)
Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.
For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. Learn more
Created
Access Key ID
Last Used
Last Used Region
Last Used Service
Status
Actions
Create New Access Key
Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead. Learn more


Create Access Key
Your access key (access key ID and secret access key) has been created successfully.
Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.
To help protect your security, store your secret access key securely and do not share it.
Show Access Key
Download Key File
Close

Run the below commands to configure.

```
aws configure
```

Enter the Access Key ID and Secret Access Key obtained after creation of access key.



8. Install and Setup Kubectl

Run the below commands to install Kubectl

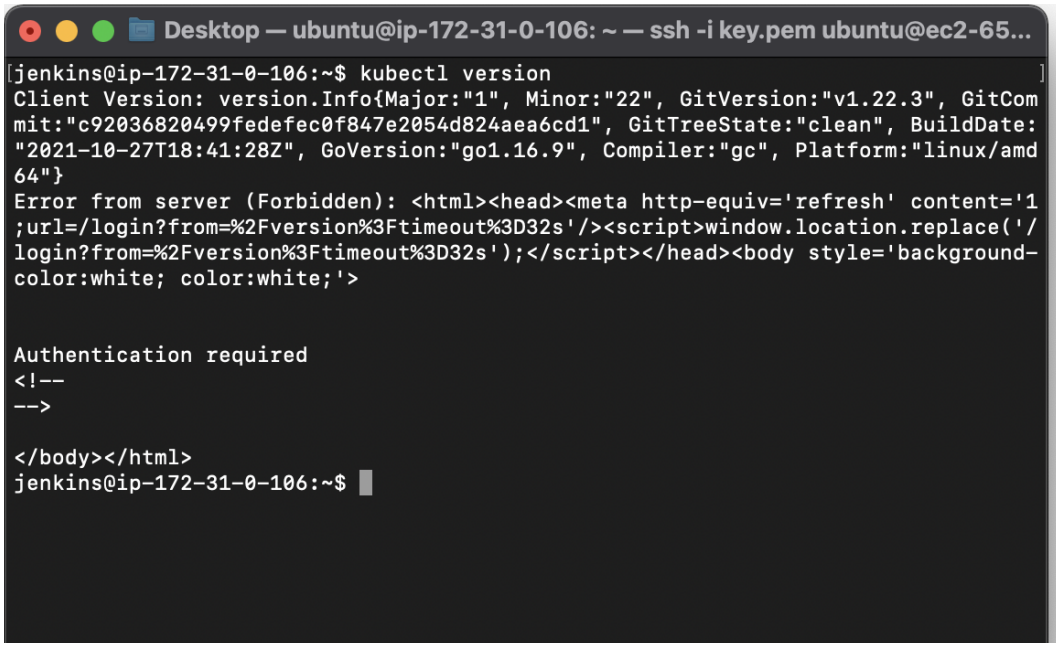
```
curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
chmod +x ./kubectl
```

```
sudo mv ./kubectl /usr/local/bin
```

To verify the installation, run the following

```
Kubectl version
```



9. Install and Setup eksctl

Run the below commands to install eksctl

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

```
sudo mv /tmp/eksctl /usr/local/bin
```

To verify the installation, run the following

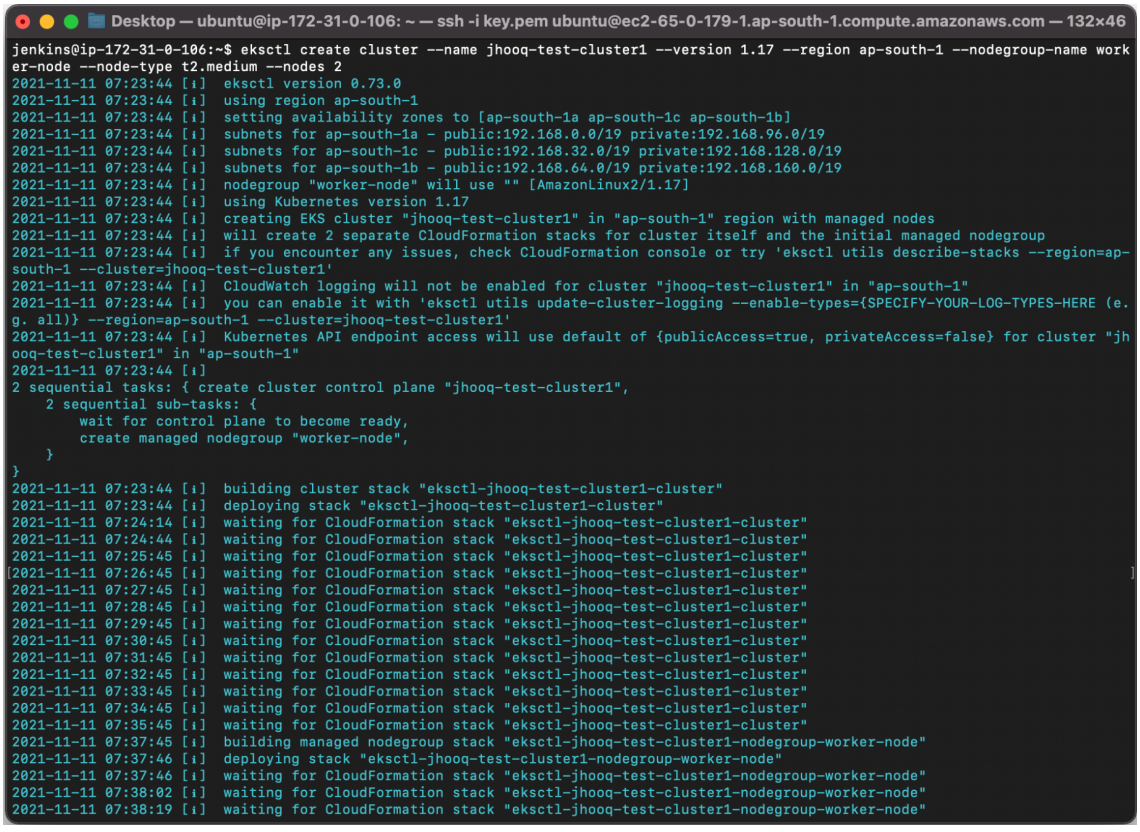
```
eksctl version
```

```
jenkins@ip-172-31-0-106:~$  curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
jenkins@ip-172-31-0-106:~$ sudo mv /tmp/eksctl /usr/local/bin
jenkins@ip-172-31-0-106:~$ eksctl version
0.73.0
jenkins@ip-172-31-0-106:~$
```

10. Create eks cluster using eksctl

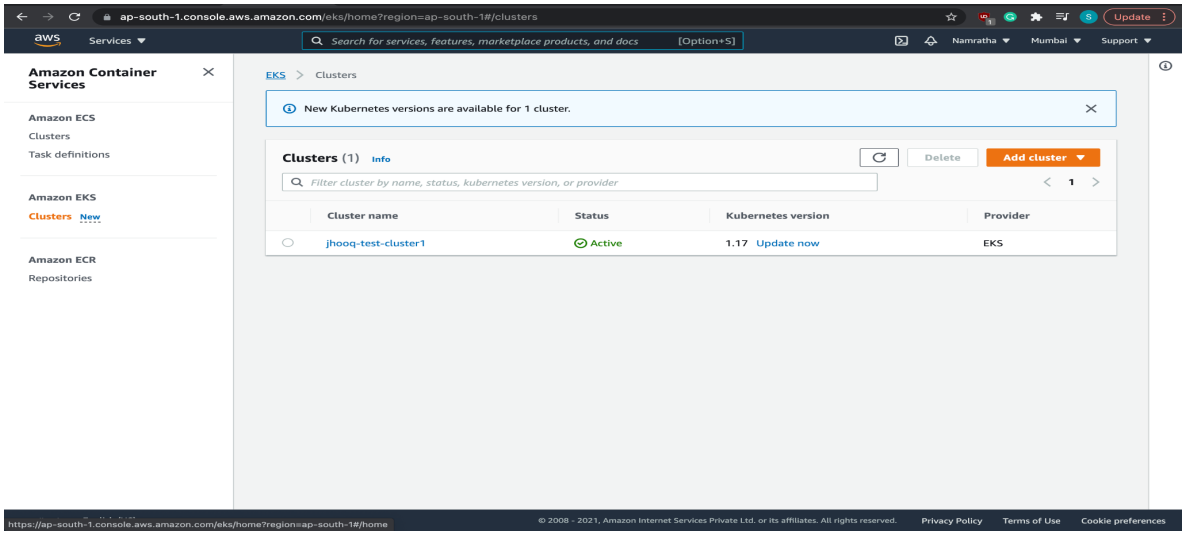
To create a cluster, run the eksctl command

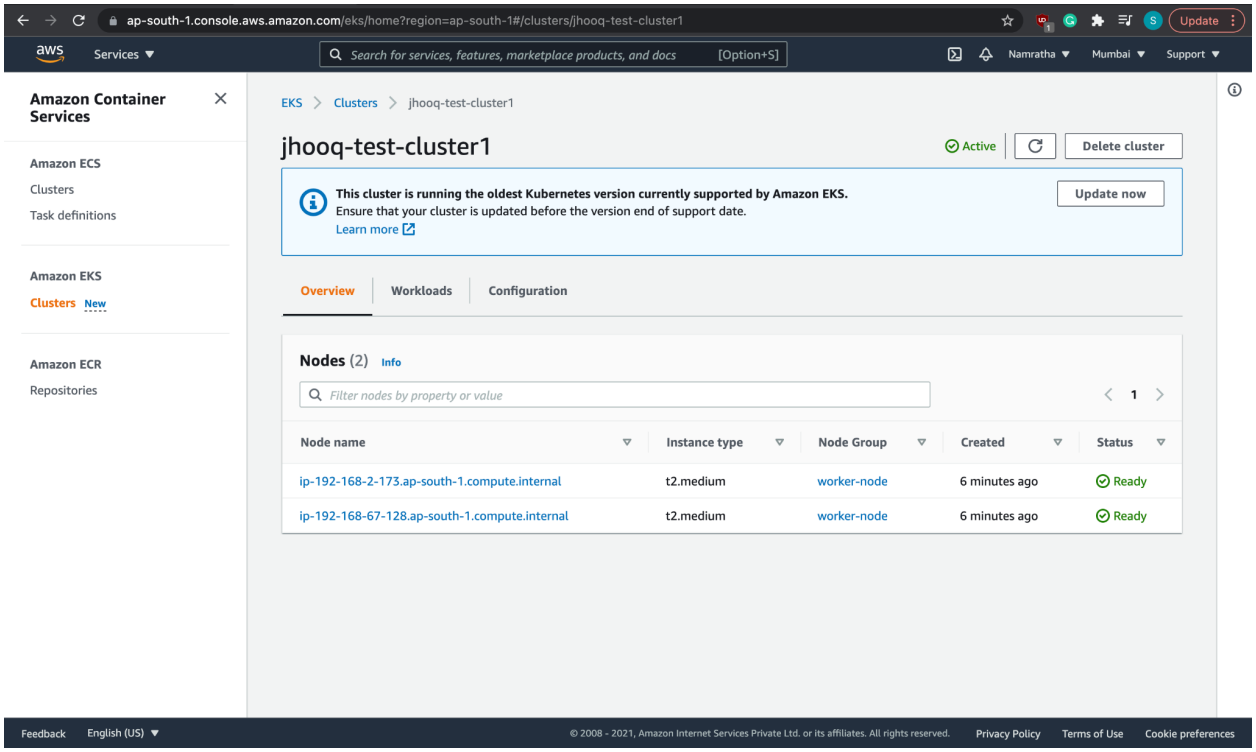
```
eksctl create cluster \
--name jhooq-test-cluster1 \
--version 1.17 \
--region ap-south-1b \
--nodegroup-name worker-node \
--node-type t2.micro \
--nodes 2
```



```
Desktop — ubuntu@ip-172-31-0-106: ~ — ssh -i key.pem ubuntu@ec2-65-0-179-1.ap-south-1.compute.amazonaws.com — 132×46
2021-11-11 07:23:44 [ℹ]  building cluster stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:23:44 [ℹ]  deploying stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:24:14 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:24:44 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:25:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:26:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:27:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:28:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:29:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:30:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:31:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:32:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:33:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:34:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:35:45 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-cluster"
2021-11-11 07:37:45 [ℹ]  building managed nodegroup stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:37:46 [ℹ]  deploying stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:37:46 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:38:02 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:38:19 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:38:38 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:38:55 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:39:15 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:39:34 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:39:53 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:40:10 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:40:28 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:40:44 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:41:00 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:41:19 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:41:34 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:41:51 [ℹ]  waiting for CloudFormation stack "eksctl-jhooq-test-cluster1-nodegroup-worker-node"
2021-11-11 07:41:51 [ℹ]  waiting for the control plane availability...
2021-11-11 07:41:51 [✔]  saved kubeconfig as "/var/lib/jenkins/.kube/config"
2021-11-11 07:41:51 [ℹ]  no tasks
2021-11-11 07:41:51 [✔]  all EKS cluster resources for "jhooq-test-cluster1" have been created
2021-11-11 07:41:51 [ℹ]  nodegroup "worker-node" has 2 node(s)
2021-11-11 07:41:51 [ℹ]  node "ip-192-168-2-173.ap-south-1.compute.internal" is ready
2021-11-11 07:41:51 [ℹ]  node "ip-192-168-67-128.ap-south-1.compute.internal" is ready
2021-11-11 07:41:51 [ℹ]  waiting for at least 2 node(s) to become ready in "worker-node"
2021-11-11 07:41:51 [ℹ]  nodegroup "worker-node" has 2 node(s)
2021-11-11 07:41:51 [ℹ]  node "ip-192-168-2-173.ap-south-1.compute.internal" is ready
2021-11-11 07:41:51 [ℹ]  node "ip-192-168-67-128.ap-south-1.compute.internal" is ready
2021-11-11 07:41:52 [ℹ]  kubectl command should work with "/var/lib/jenkins/.kube/config", try 'kubectl get nodes'
2021-11-11 07:41:52 [✔]  EKS cluster "jhooq-test-cluster1" in "ap-south-1" region is ready
jenkins@ip-172-31-0-106:~$
```

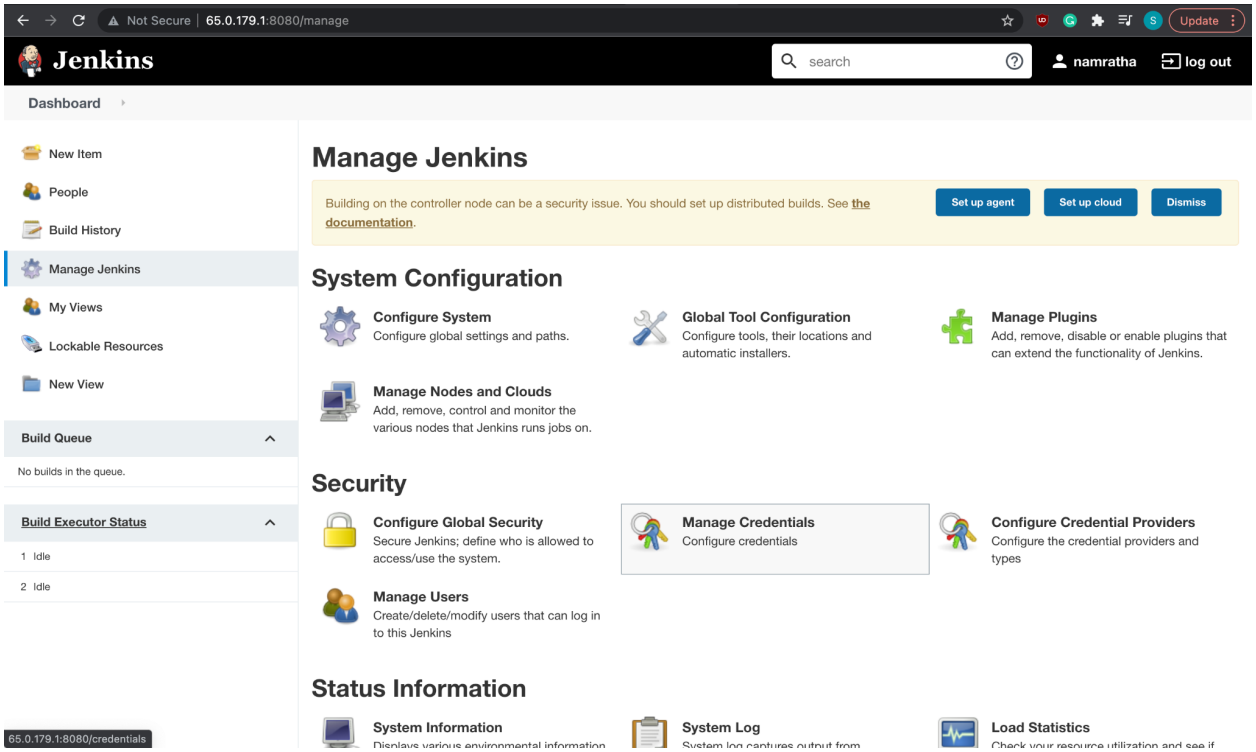
Verify the EKS kubernetes cluster from AWS.

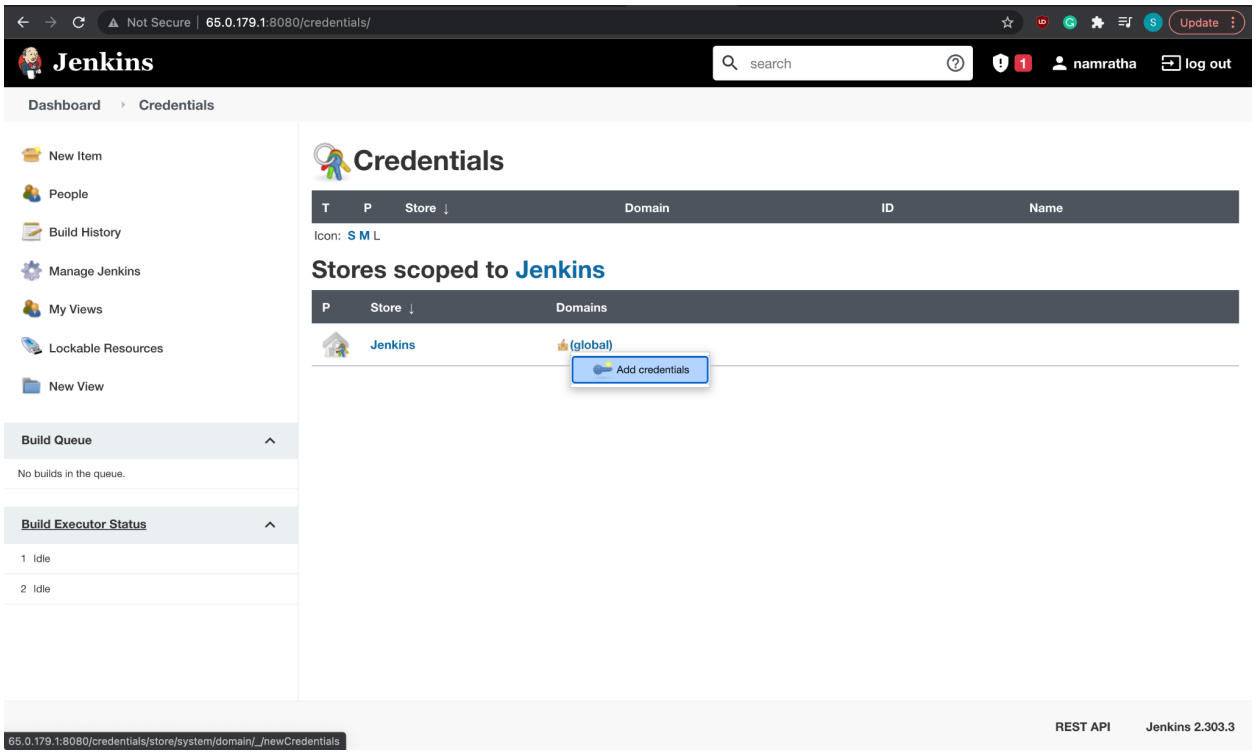


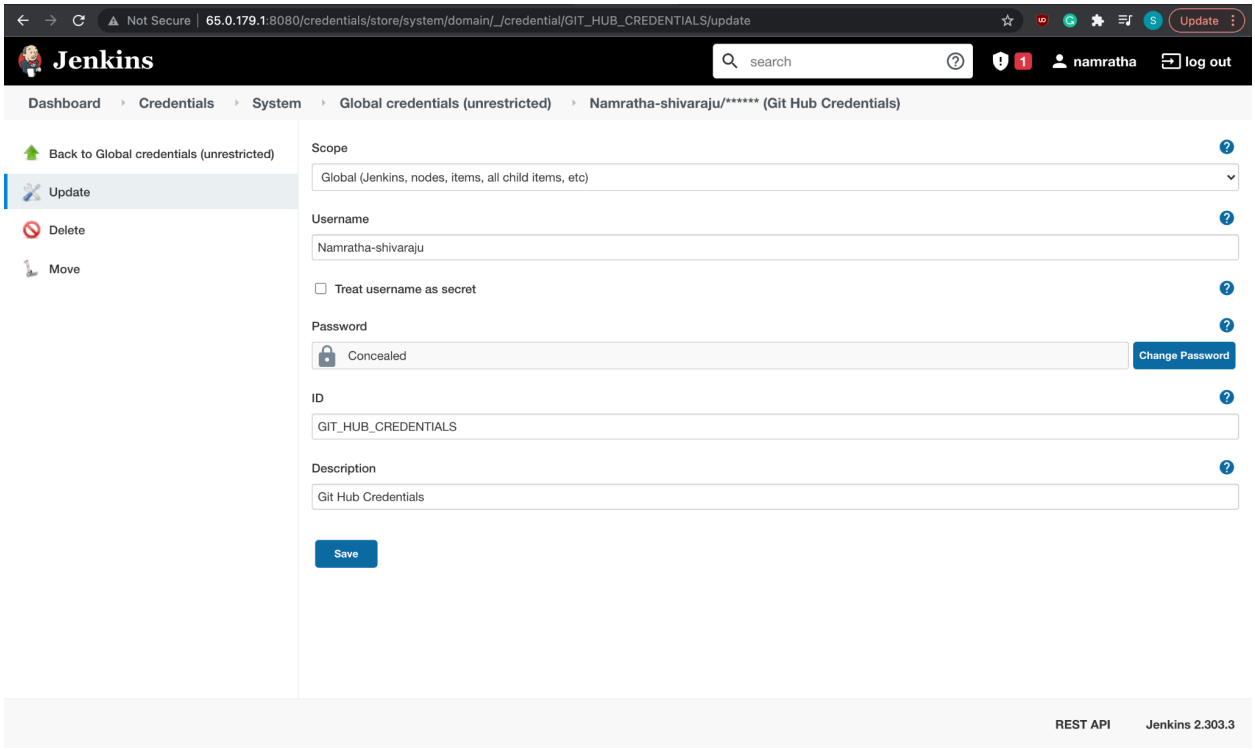


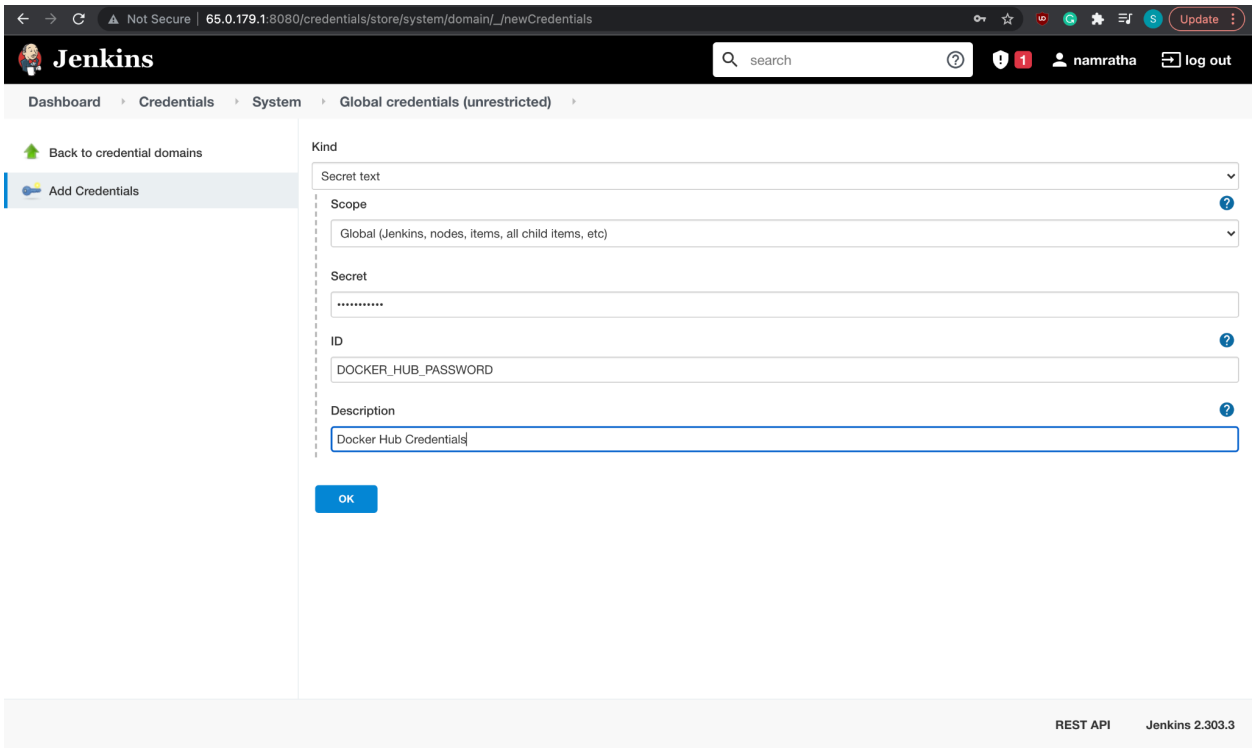
11. Add Docker and GitHub Credentials into Jenkins

Goto -> Jenkins -> Manage Jenkins -> Manage Credentials -> Stored scoped to jenkins -> global -> Add Credentials.


Not Secure | 65.0.179.1:8080/manage
Update
Jenkins
search
namratha
log out
Dashboard
New Item
People
Build History
Manage Jenkins
My Views
Lockable Resources
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Manage Jenkins
Building on the controller node can be a security issue. You should set up distributed builds. See the documentation.
Set up agent
Set up cloud
Dismiss
System Configuration
Configure System
Configure global settings and paths.
Global Tool Configuration
Configure tools, their locations and automatic installers.
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
Security
Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.
Manage Credentials
Configure credentials
Configure Credential Providers
Configure the credential providers and types
Manage Users
Create/delete/modify users that can log in to this Jenkins
Status Information
System Information
System Log
Load Statistics
65.0.179.1:8080/credentials

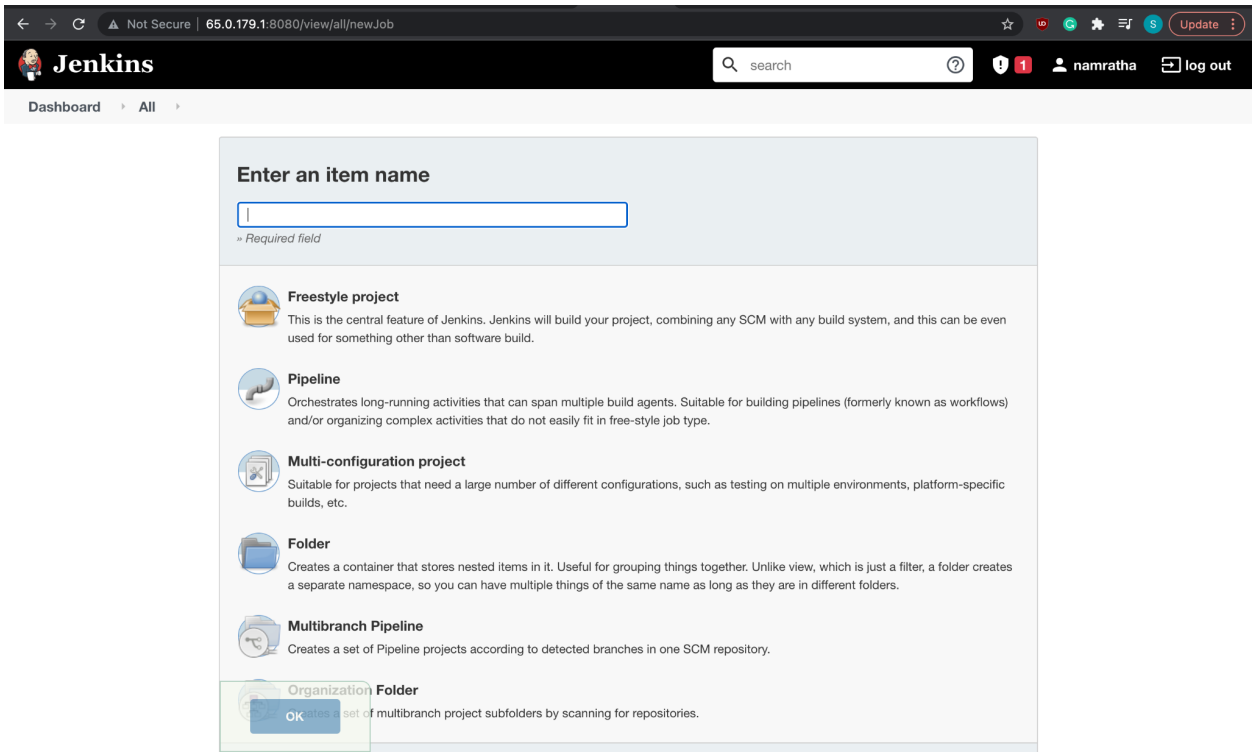

Not Secure | 65.0.179.1:8080/credentials/
Update
Jenkins
search
1
namratha
log out
Dashboard
Credentials
New Item
People
Build History
Manage Jenkins
My Views
Lockable Resources
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Credentials
T
P
Store ↓
Domain
ID
Name
Icon: S M L
Stores scoped to Jenkins
P
Store ↓
Domains
Jenkins
(global)
Add credentials
REST API
Jenkins 2.303.3
65.0.179.1:8080/credentials/store/system/domain/_/newCredentials

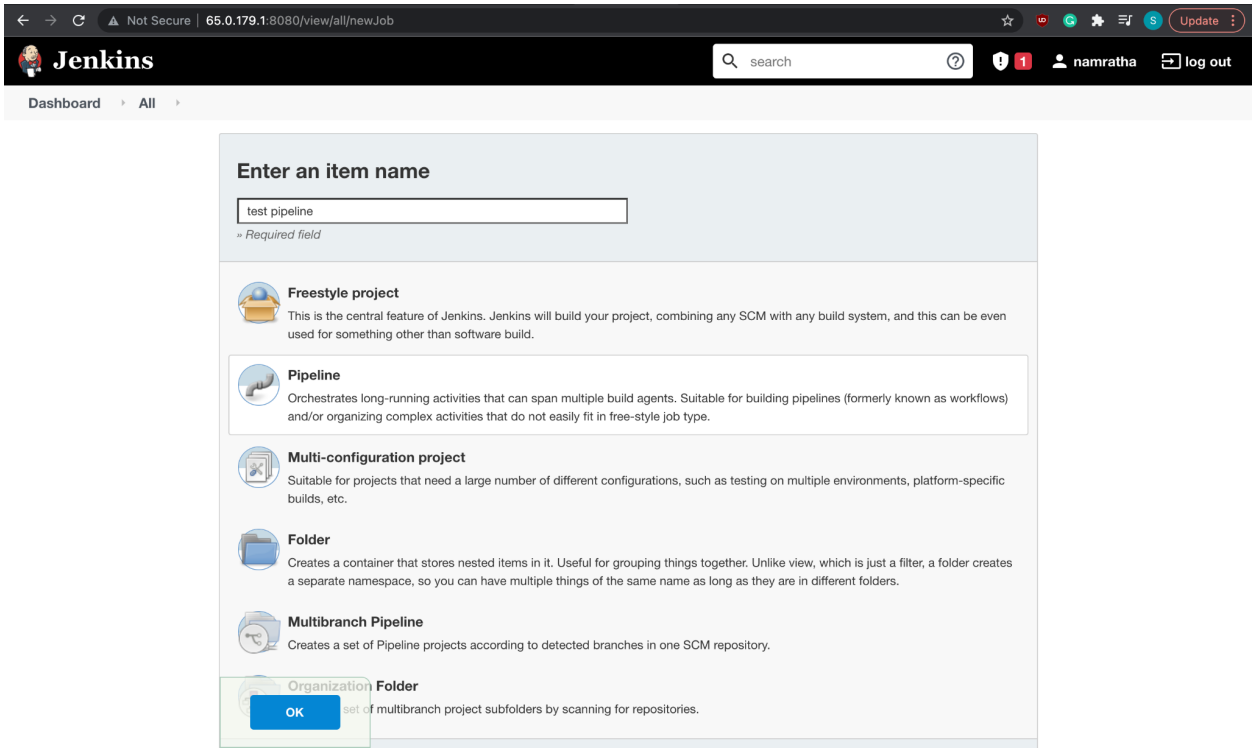

Not Secure | 65.0.179.1:8080/credentials/store/system/domain/_/credential/GIT_HUB_CREDENTIALS/update
Update
Jenkins
search
1
namratha
log out
Dashboard › Credentials › System › Global credentials (unrestricted) › Namratha-shivaraju/****** (Git Hub Credentials)
Back to Global credentials (unrestricted)
Update
Delete
Move
Scope
Global (Jenkins, nodes, items, all child items, etc)
Username
Namratha-shivaraju
Treat username as secret
Password
Concealed
Change Password
ID
GIT_HUB_CREDENTIALS
Description
Git Hub Credentials
Save
REST API
Jenkins 2.303.3


Not Secure | 65.0.179.1:8080/credentials/store/system/domain/_/newCredentials
Update
Jenkins
search
1
namratha
log out
Dashboard › Credentials › System › Global credentials (unrestricted) ›
Back to credential domains
Add Credentials
Kind
Secret text
Scope
Global (Jenkins, nodes, items, all child items, etc)
Secret
ID
DOCKER_HUB_PASSWORD
Description
Docker Hub Credentials
OK
REST API
Jenkins 2.303.3

12. CI/CD pipeline

Go to jenkins -> new item-> enter the name of the item -> select “Pipeline” -> click Ok.





13. Add jenkins stages

We need to write jenkins pipeline to deploy our application.
Combine the stages and add it in the pipeline script.
Click “Apply” and “Save”.

13.1. Jenkins stage-1 : Checkout the GitHub Repository

```
stage("Git Clone"){

git branch: 'main', credentialsId: 'GIT_HUB_CREDENTIALS',
url:
'https://github.com/Namratha-shivaraju/k8s-jenkins-aws.git
'
}
```

13.2. Jenkins stage-2 : Gradle compilation and build

```
stage('Gradle Build') {
sh './gradlew build'
```

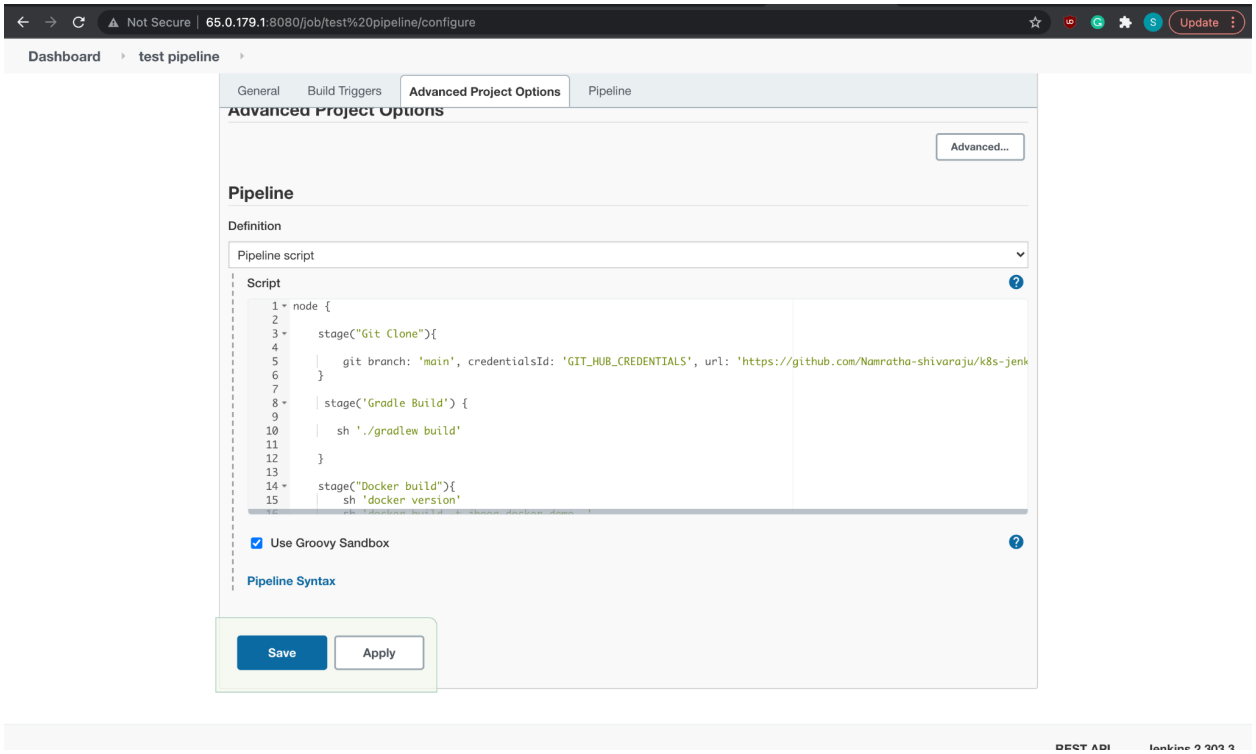
```
}
```

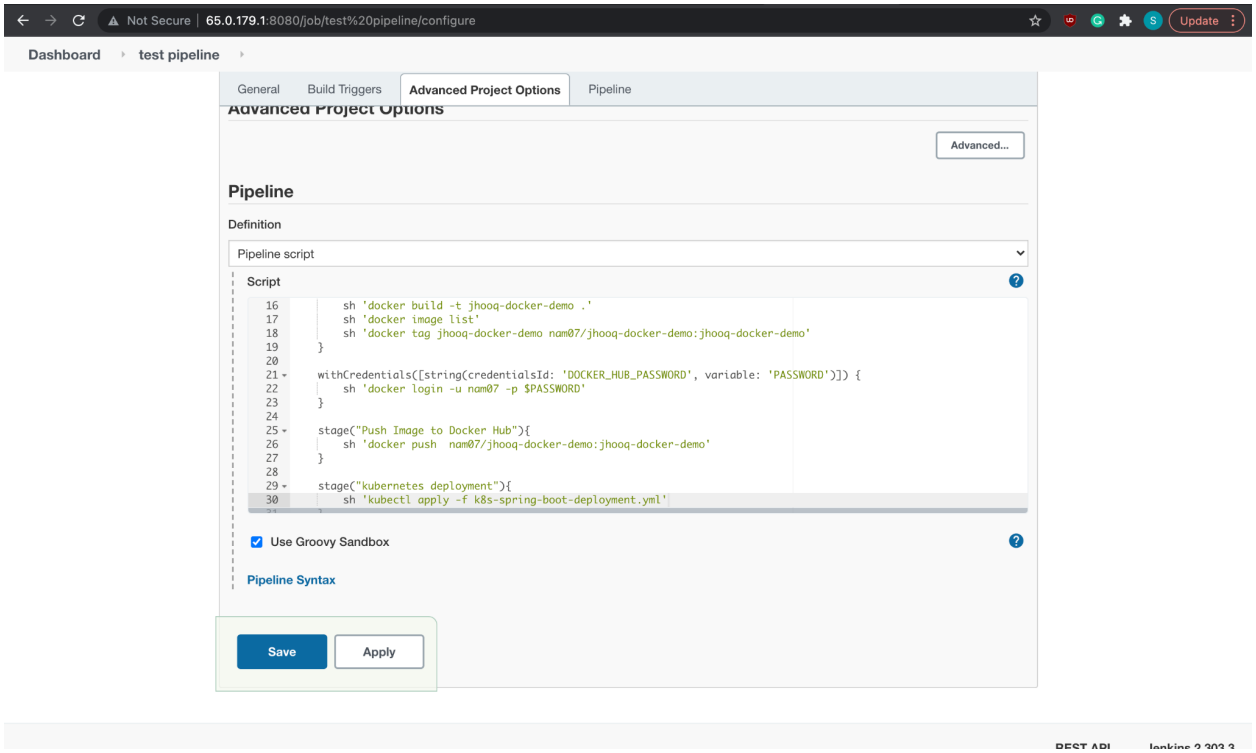
13.3. Jenkins stage-3 : Create Docker Container and push to Docker Hub

```
stage("Docker build"){
        sh 'docker version'
        sh 'docker build -t jhooq-docker-demo .'
        sh 'docker image list'
        sh 'docker tag jhooq-docker-demo
nam07/jhooq-docker-demo:jhooq-docker-demo'
    }

withCredentials([string(credentialsId:
'DOCKER_HUB_PASSWORD', variable: 'PASSWORD')]) {
        sh 'docker login -u nam07 -p $PASSWORD'
    }

stage("Push Image to Docker Hub"){
        sh 'docker push
nam07/jhooq-docker-demo:jhooq-docker-demo'
    }
```

13.4. Jenkins stage-4 : Kubernetes deployment

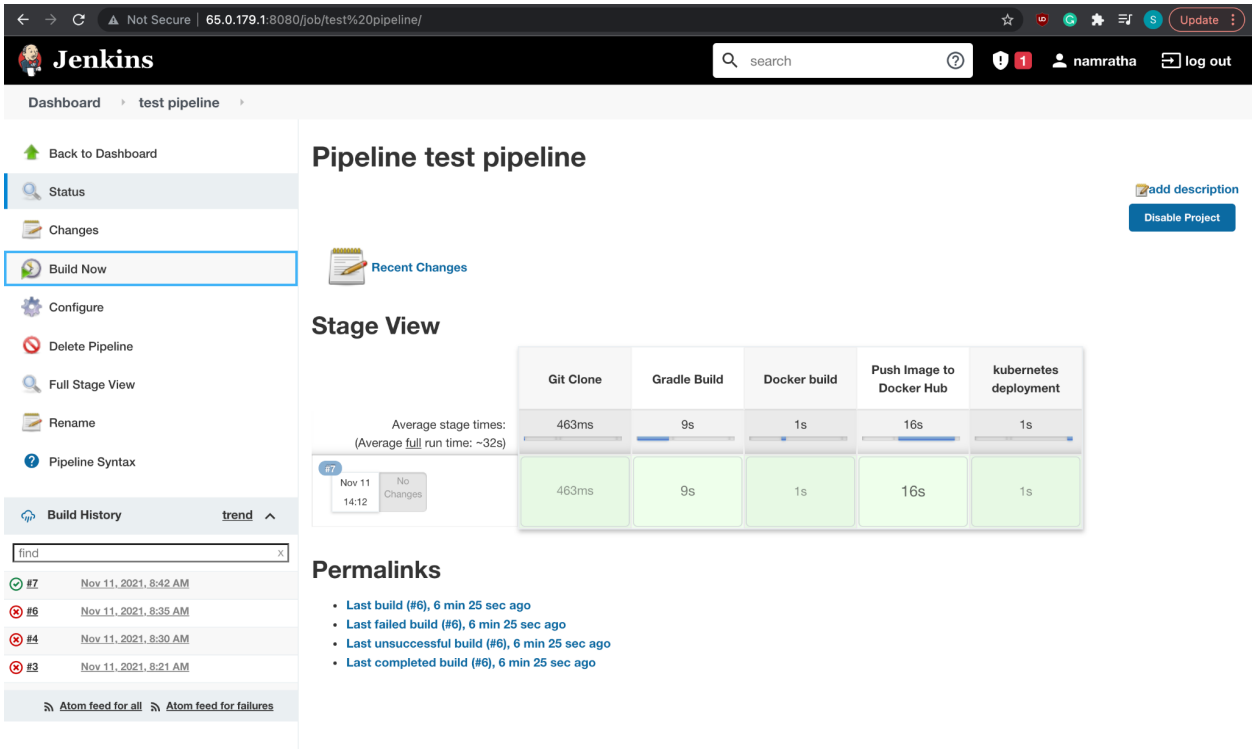
```
stage("kubernetes deployment"){
        sh 'kubectl apply -f
k8s-spring-boot-deployment.yml'
    }
```





14. Build and run the pipeline

Go to the pipeline and click on “Build Now”.

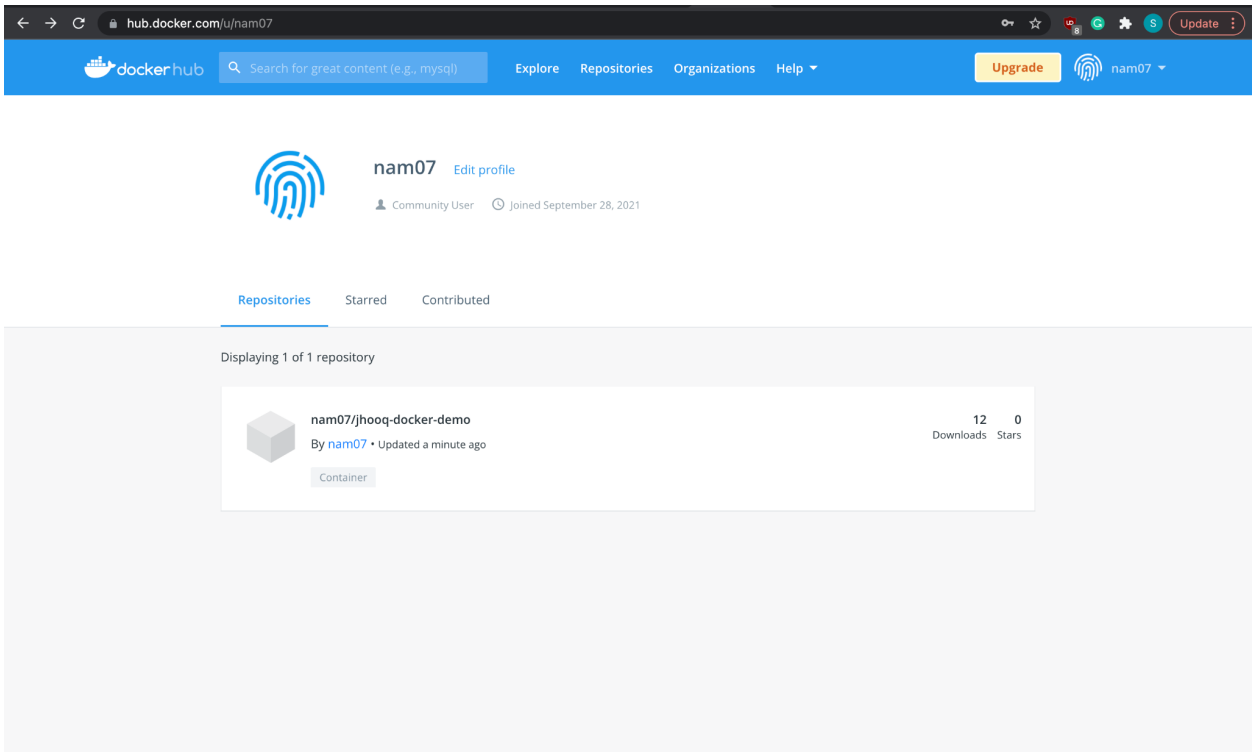


15. Verify using kubectl commands

Run the below commands to verify the deployment.

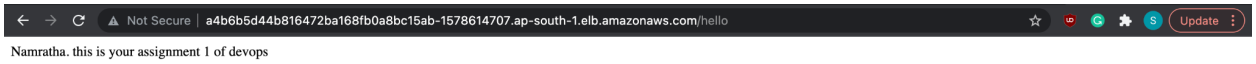
```
kubectl get deployments
```

```
kubectl get service
```



16. Access the rest end point from browser

Copy the expernal IP address from terminal and paste it in the browser with /hello.



Namratha. this is your assignment 1 of devops