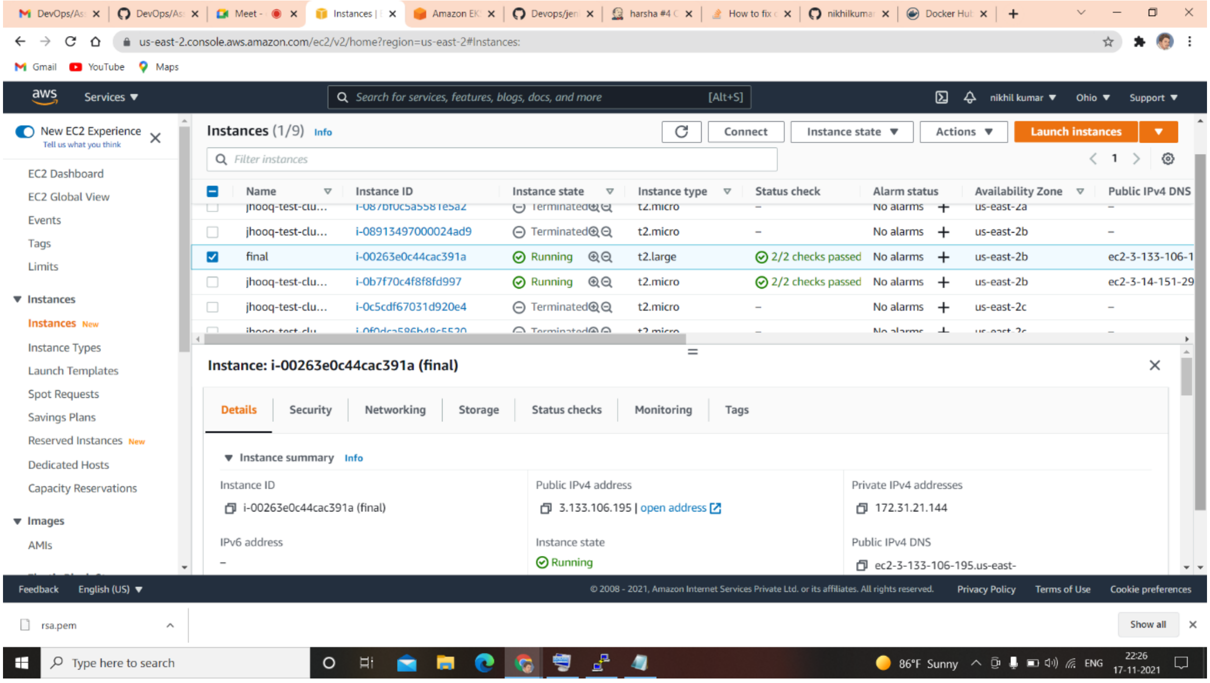
DevOps Final Assignment CI CD Pipeline

N. Nikhil Kumar 18bec031

Y. Santhi Swarup 18bec051

Create aws Ubuntu 20.04 instance type t2.large….



Instances (1/9) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	jhooq-test-clu...	i-087bf0c5a5581e5a2	Terminated	t2.micro	–	No alarms	us-east-2a	–
☐	jhooq-test-clu...	i-08913497000024ad9	Terminated	t2.micro	–	No alarms	us-east-2b	–
☑	final	i-00263e0c44cac391a	Running	t2.large	2/2 checks passed	No alarms	us-east-2b	ec2-3-133-106-1
☐	jhooq-test-clu...	i-0b7f70c4f8f8fd997	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-3-14-151-29
☐	jhooq-test-clu...	i-0c5cdf67031d920e4	Terminated	t2.micro	–	No alarms	us-east-2c	–
☐	jhooq-test-clu...	i-0f0dca586b48c5520	Terminated	t2.micro	–	No alarms	us-east-2c	–

Instance: i-00263e0c44cac391a (final)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Instance ID: i-00263e0c44cac391a (final)

Public IPv4 address: 3.133.106.195 | open address

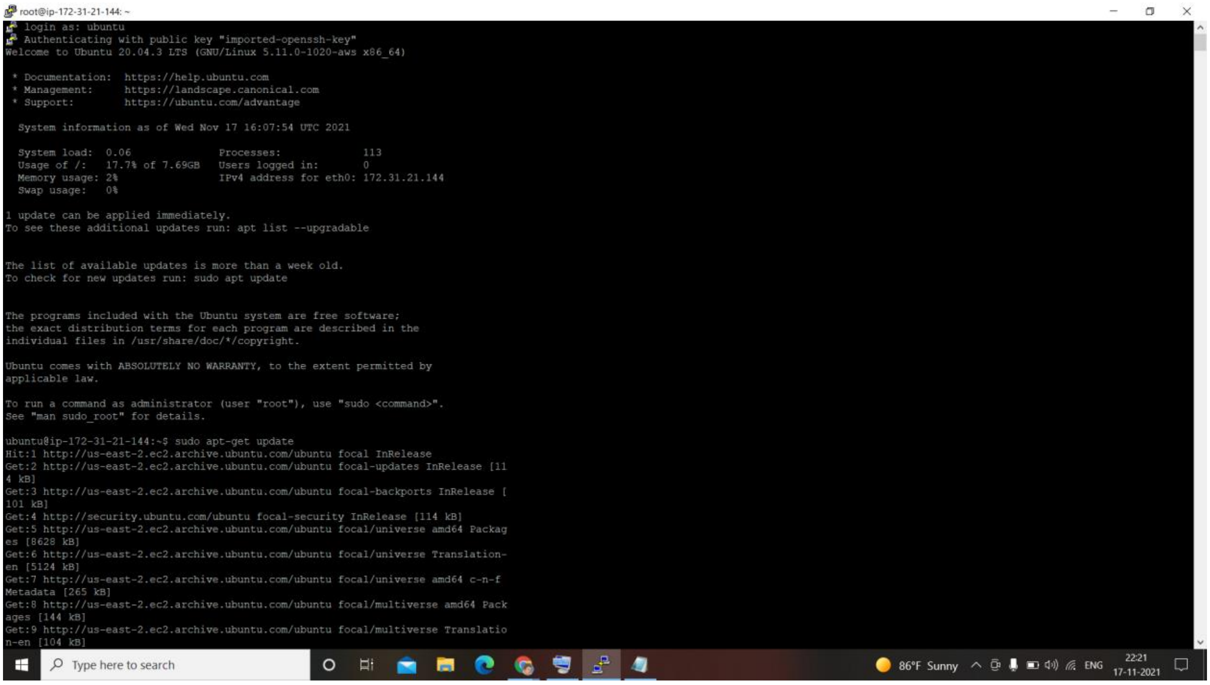
Private IPv4 addresses: 172.31.21.144

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-133-106-195.us-east-

Configure the instance using putty and install jdk and Jenkins



```
 login as: ubuntu
 Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Nov 17 16:07:54 UTC 2021

  System load:  0.06              Processes:             113
  Usage of /:   17.7% of 7.69GB   Users logged in:       0
  Memory usage: 2%                IPv4 address for eth0: 172.31.21.144
  Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-21-144:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [11
4 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [
101 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packag
es [8628 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-
en [5124 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f
Metadata [265 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Pack
ages [144 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translatio
n-en [104 kB]
```

Commands to install JDK and Jenkins

sudo apt-get update

sudo apt install openjdk-11-jre-headless

java –version

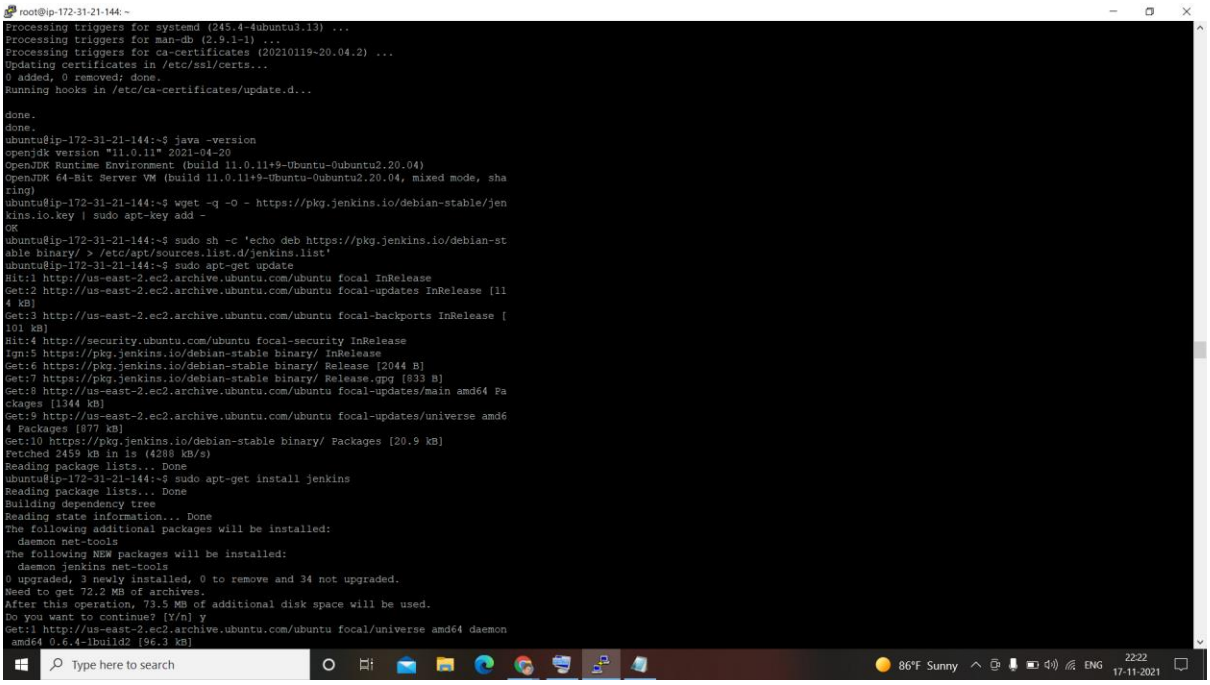
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

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



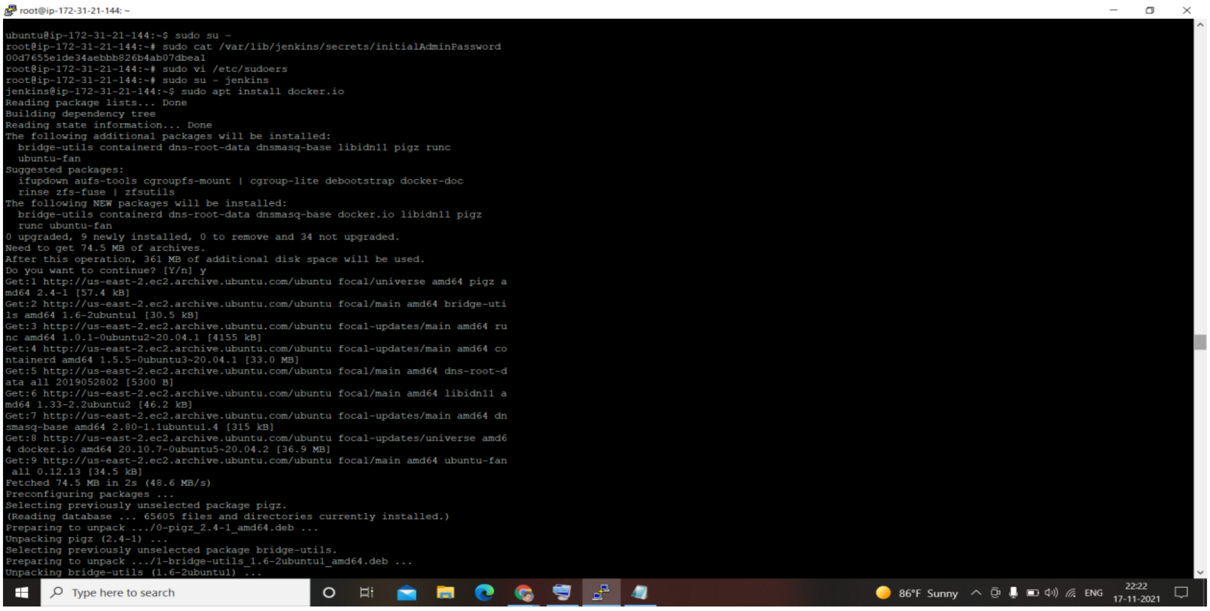
```
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for ca-certificates (20210119~20.04.2) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
ubuntu@ip-172-31-21-144:~$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sha
ring)
ubuntu@ip-172-31-21-144:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jen
kins.io.key | sudo apt-key add -
OK
ubuntu@ip-172-31-21-144:~$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-st
able binary/ > /etc/apt/sources.list.d/jenkins.list'
ubuntu@ip-172-31-21-144:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [11
4 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [
101 kB]
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Pa
ckages [1344 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd6
4 Packages [877 kB]
Get:10 https://pkg.jenkins.io/debian-stable binary/ Packages [20.9 kB]
Fetched 2459 kB in 1s (4288 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-21-144:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon net-tools
The following NEW packages will be installed:
  daemon jenkins net-tools
0 upgraded, 3 newly installed, 0 to remove and 34 not upgraded.
Need to get 72.2 MB of archives.
After this operation, 73.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 daemon
 amd64 0.6.4-1build2 [96.3 kB]
```

Now we need to start using Jenkins from the public ipv4 address of the instance created above. And Jenkins by default runs on the port 8080.

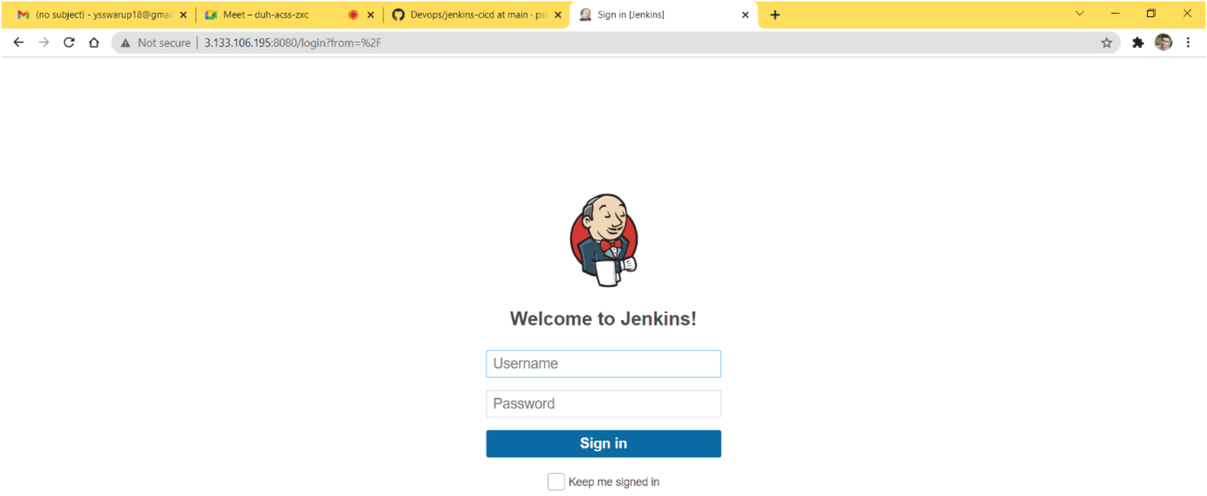
Now use the command to unlock Jenkins to access it

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

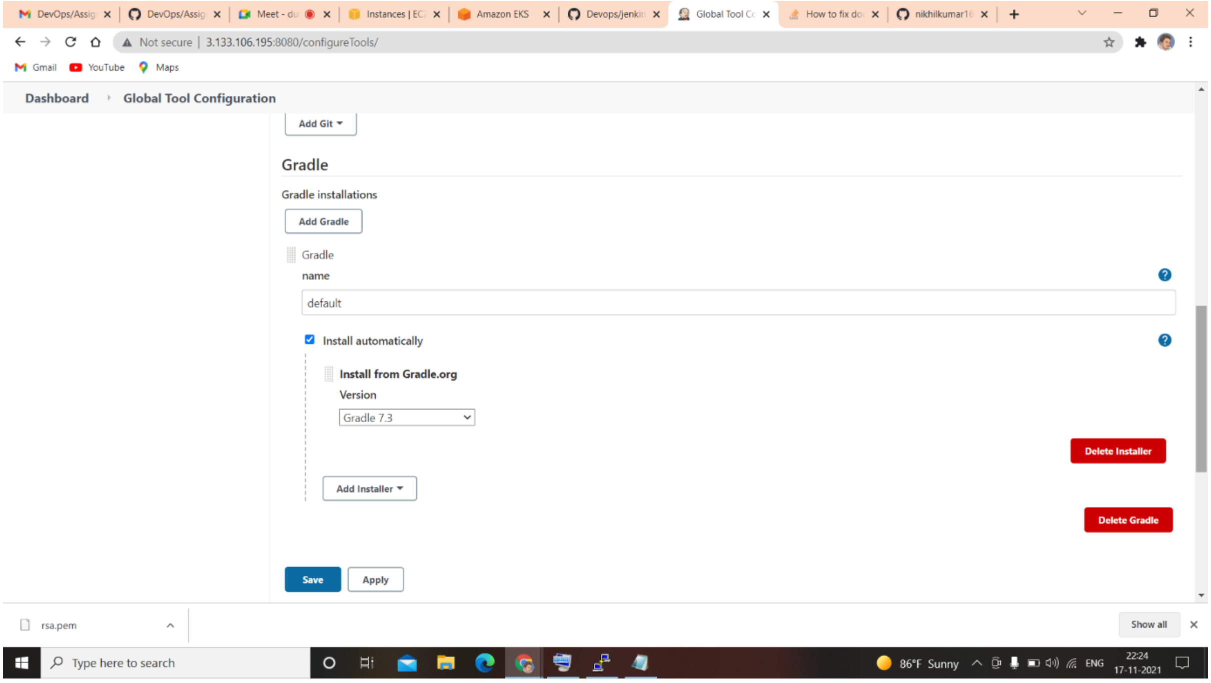
Opt for install suggested plugin. After completing the installation of the suggested plugin you need to set the First Admin User for Jenkins. Also, check the instance configuration because it will be used for accessing the Jenkins. Now Jenkins is ready to be used



```
ubuntu@ip-172-31-21-144:~$ sudo su -
root@ip-172-31-21-144:~# sudo cat /var/lib/jenkins/secrets/initialAdminPassword
00d7655e1de34aebbb826b4ab07dbea1
root@ip-172-31-21-144:~# sudo vi /etc/sudoers
root@ip-172-31-21-144:~# sudo su - jenkins
jenkins@ip-172-31-21-144:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc
  ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc
  rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz
  runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 34 not upgraded.
Need to get 74.5 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz a
md64 2.4-1 [57.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-uti
ls amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 ru
nc amd64 1.0.1-0ubuntu2~20.04.1 [4155 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 co
ntainerd amd64 1.5.5-0ubuntu3~20.04.1 [33.0 MB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-d
ata all 2019052802 [5300 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 a
md64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dn
smasq-base amd64 2.80-1.1ubuntu1.4 [315 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd6
4 docker.io amd64 20.10.7-0ubuntu5~20.04.2 [36.9 MB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan
 all 0.12.13 [34.5 kB]
Fetched 74.5 MB in 2s (48.6 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 65605 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
```


(no subject) - ysswarup18@gmai
Meet – duh-acss-zxc
Devops/jenkins-cicd at main · ps
Sign in [Jenkins]
Not secure | 3.133.106.195:8080/login?from=%2F
Welcome to Jenkins!
Username
Password
Sign in
□ Keep me signed in

Next setup Gradle in Jenkins



Gradle

Gradle installations

Gradle

name: default

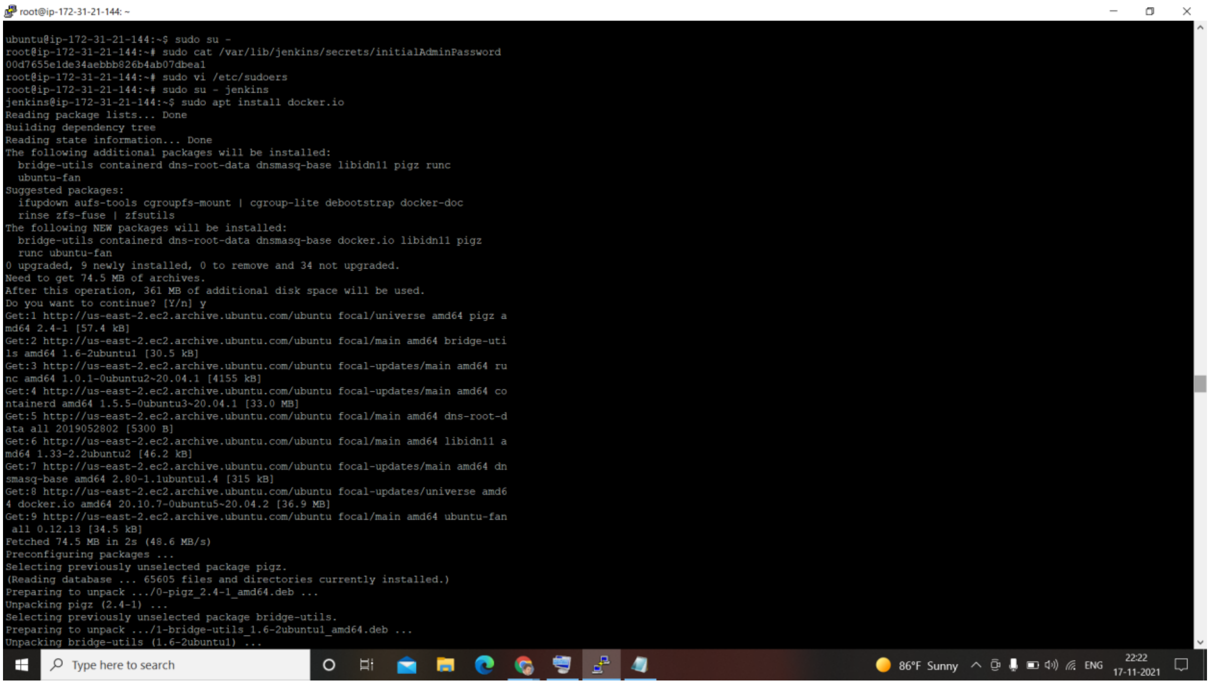
☑ Install automatically

Install from Gradle.org

Version: Gradle 7.3

To interact with the Kubernetes cluster Jenkins will be executing the shell script with the Jenkins user, so the Jenkins user should have an administration (super user) role assigned forehand. Let’s add Jenkins user as an administrator and also ass NOPASSWD so that during the pipeline run it will not ask for root password. Open the file /etc/sudoers in vi mode.

sudo vi /etc/sudoers



```
ubuntu@ip-172-31-21-144:~$ sudo su -
root@ip-172-31-21-144:~# sudo cat /var/lib/jenkins/secrets/initialAdminPassword
00d7655e1de34aebbb826b4ab07dbea1
root@ip-172-31-21-144:~# sudo vi /etc/sudoers
root@ip-172-31-21-144:~# sudo su - jenkins
jenkins@ip-172-31-21-144:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc
  ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc
  rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz
  runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 34 not upgraded.
Need to get 74.5 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz a
md64 2.4-1 [57.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-uti
ls amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 ru
nc amd64 1.0.1-0ubuntu2~20.04.1 [4155 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 co
ntainerd amd64 1.5.5-0ubuntu3~20.04.1 [33.0 MB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-d
ata all 2019052802 [5300 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 a
md64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dn
smasq-base amd64 2.80-1.1ubuntu1.4 [315 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd6
4 docker.io amd64 20.10.7-0ubuntu5~20.04.2 [36.9 MB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan
 all 0.12.13 [34.5 kB]
Fetched 74.5 MB in 2s (48.6 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 65605 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
```

jenkins ALL=(ALL) NOPASSWD: ALL

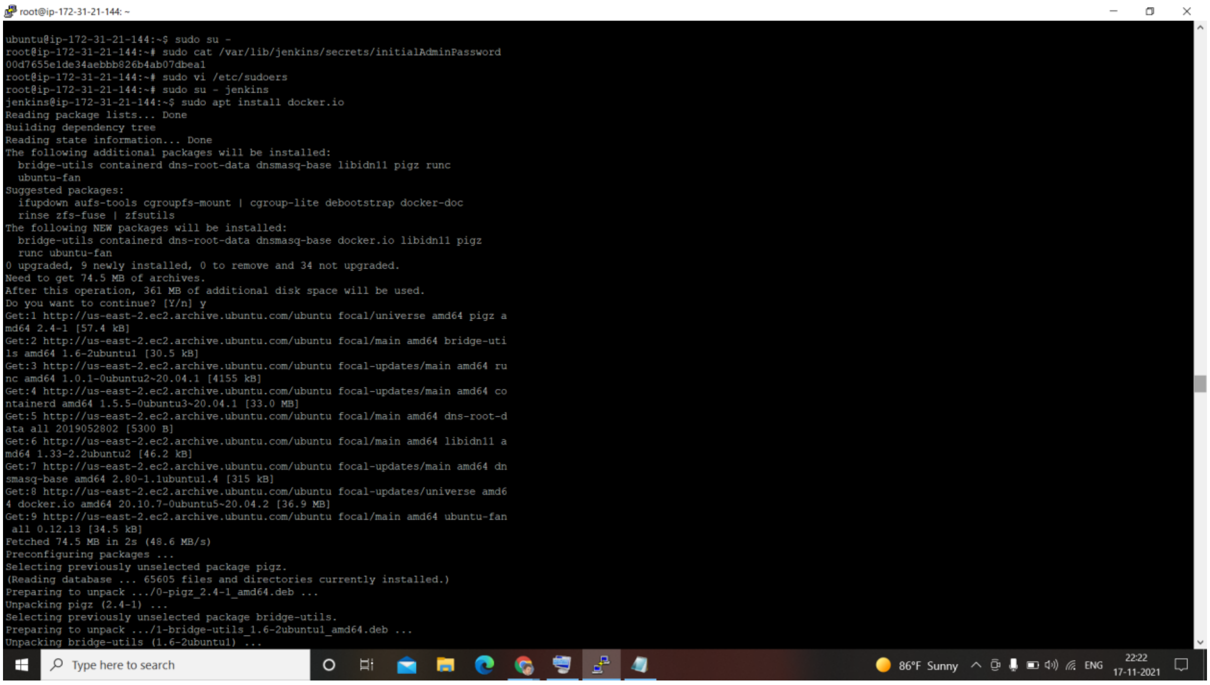
sudo su - jenkins

Now we need to install the docker after installing the Jenkins. The docker installation will be done by the Jenkins user because now it has root user privileges.

Use the following command for installing the docker

sudo apt install docker.io

sudo usermod -aG docker jenkins

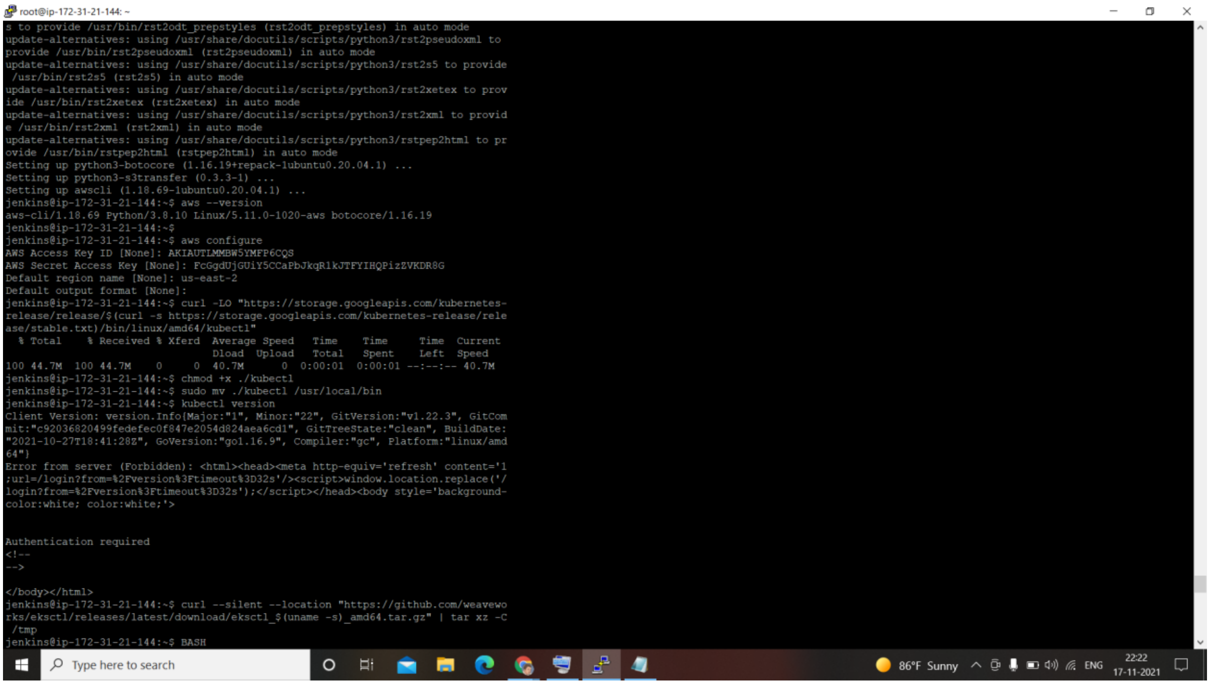


```
ubuntu@ip-172-31-21-144:~$ sudo su -
root@ip-172-31-21-144:~# sudo cat /var/lib/jenkins/secrets/initialAdminPassword
00d7655e1de34aebbb826b4ab07dbea1
root@ip-172-31-21-144:~# sudo vi /etc/sudoers
root@ip-172-31-21-144:~# sudo su - jenkins
jenkins@ip-172-31-21-144:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc
  ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc
  rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz
  runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 34 not upgraded.
Need to get 74.5 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 pigz a
md64 2.4-1 [57.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 bridge-uti
ls amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 ru
nc amd64 1.0.1-0ubuntu2~20.04.1 [4155 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 co
ntainerd amd64 1.5.5-0ubuntu3~20.04.1 [33.0 MB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-d
ata all 2019052802 [5300 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 a
md64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 dn
smasq-base amd64 2.80-1.1ubuntu1.4 [315 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd6
4 docker.io amd64 20.10.7-0ubuntu5~20.04.2 [36.9 MB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan
 all 0.12.13 [34.5 kB]
Fetched 74.5 MB in 2s (48.6 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 65605 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
```

Now we have our EC2 machine and Jenkins installed. Now we need to set up the AWS CLI on the EC2 machine so that we can use eksctl in the later stages. Let us get the installation done for AWS CLI

sudo apt install awscli

aws –version



```
s to provide /usr/bin/rst2odt_prepstyles (rst2odt_prepstyles) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2pseudoxml to
provide /usr/bin/rst2pseudoxml (rst2pseudoxml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2s5 to provide
 /usr/bin/rst2s5 (rst2s5) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xetex to prov
ide /usr/bin/rst2xetex (rst2xetex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xml to provid
e /usr/bin/rst2xml (rst2xml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rstpep2html to pr
ovide /usr/bin/rstpep2html (rstpep2html) in auto mode
Setting up python3-botocore (1.16.19+repack-1ubuntu0.20.04.1) ...
Setting up python3-s3transfer (0.3.3-1) ...
Setting up awscli (1.18.69-1ubuntu0.20.04.1) ...
jenkins@ip-172-31-21-144:~$ aws --version
aws-cli/1.18.69 Python/3.8.10 Linux/5.11.0-1020-aws botocore/1.16.19
jenkins@ip-172-31-21-144:~$
jenkins@ip-172-31-21-144:~$ aws configure
AWS Access Key ID [None]: AKIAUTLMMBW5YMFP6CQS
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]:
jenkins@ip-172-31-21-144:~$ curl -LO "https://storage.googleapis.com/kubernetes-
release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/rele
ase/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 44.7M  100 44.7M    0     0  40.7M      0  0:00:01  0:00:01 --:--:-- 40.7M
jenkins@ip-172-31-21-144:~$ chmod +x ./kubectl
jenkins@ip-172-31-21-144:~$ sudo mv ./kubectl /usr/local/bin
jenkins@ip-172-31-21-144:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.3", GitCom
mit:"c92036820499fedefec0f847e2054d824aea6cd1", GitTreeState:"clean", BuildDate:
"2021-10-27T18:41:28Z", GoVersion:"go1.16.9", Compiler:"gc", Platform:"linux/amd
64"}
Error from server (Forbidden): <html><head><meta http-equiv='refresh' content='1
;url=/login?from=%2Fversion%3Ftimeout%3D32s'/><script>window.location.replace('/
login?from=%2Fversion%3Ftimeout%3D32s');</script></head><body style='background-
color:white; color:white;'>


Authentication required
<!--
-->

</body></html>
jenkins@ip-172-31-21-144:~$ curl --silent --location "https://github.com/weavewo
rks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C
 /tmp
jenkins@ip-172-31-21-144:~$ BASH
```

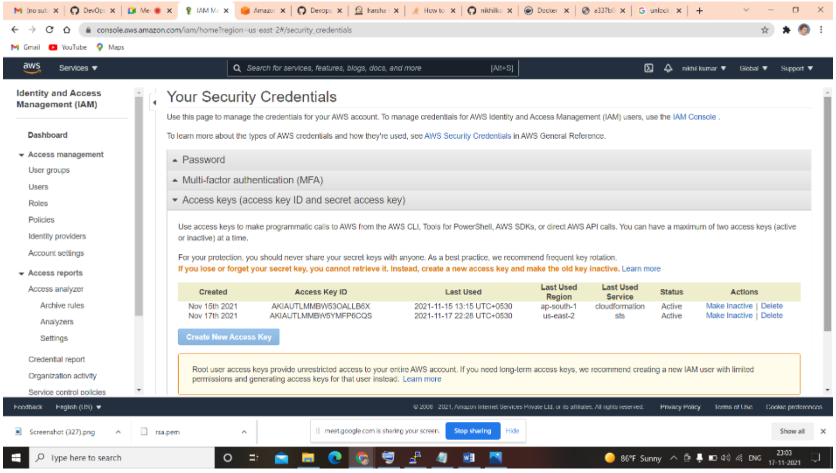
Now after installing the AWS CLI, let's configure the AWS CLI so that it can authenticate and communicate with the AWS environment. To configure the AWS the first command we are going to run is:

aws configure

Once you execute the above command it will ask for the following information

AWS Access Key ID [None] , AWS Secret Access Key [None], Default region name [None], Default output format [None]

we can find this information by going into AWS -> My Security Credentials. Then navigate to Access Keys (access key ID and secret access key). You can click on the Create New Access Key and it will let you generate - AWS Access Key ID, AWS Secret Access Key. Default region name - You can find it from the menu.



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console .

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

- Password
- Multi-factor authentication (MFA)
- 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
Nov 15th 2021	AKIAUTLMMBW53OALLB6X	2021-11-15 13:15 UTC+0530	ap-south-1	cloudformation	Active	Make Inactive | Delete
Nov 17th 2021	AKIAUTLMMBW5YMFP6CQS	2021-11-17 22:28 UTC+0530	us-east-2	sts	Active	Make Inactive | Delete

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

Install and Setup Kubectl and eksctl

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

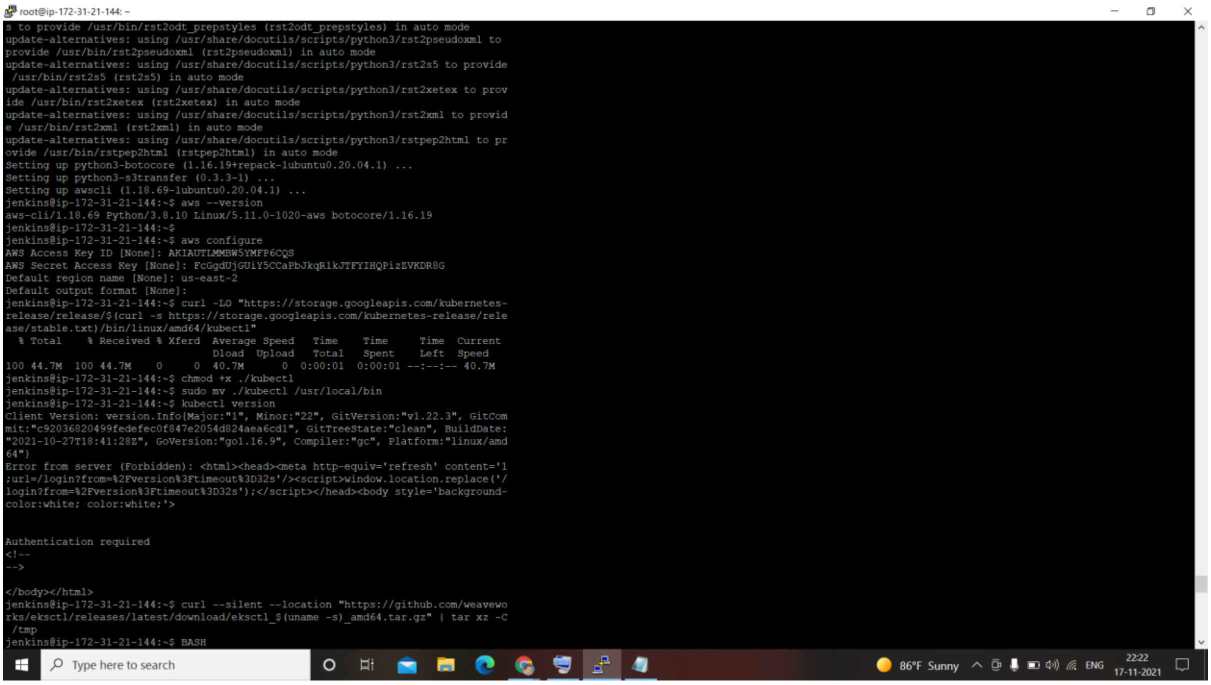
sudo mv ./kubectl /usr/local/bin

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

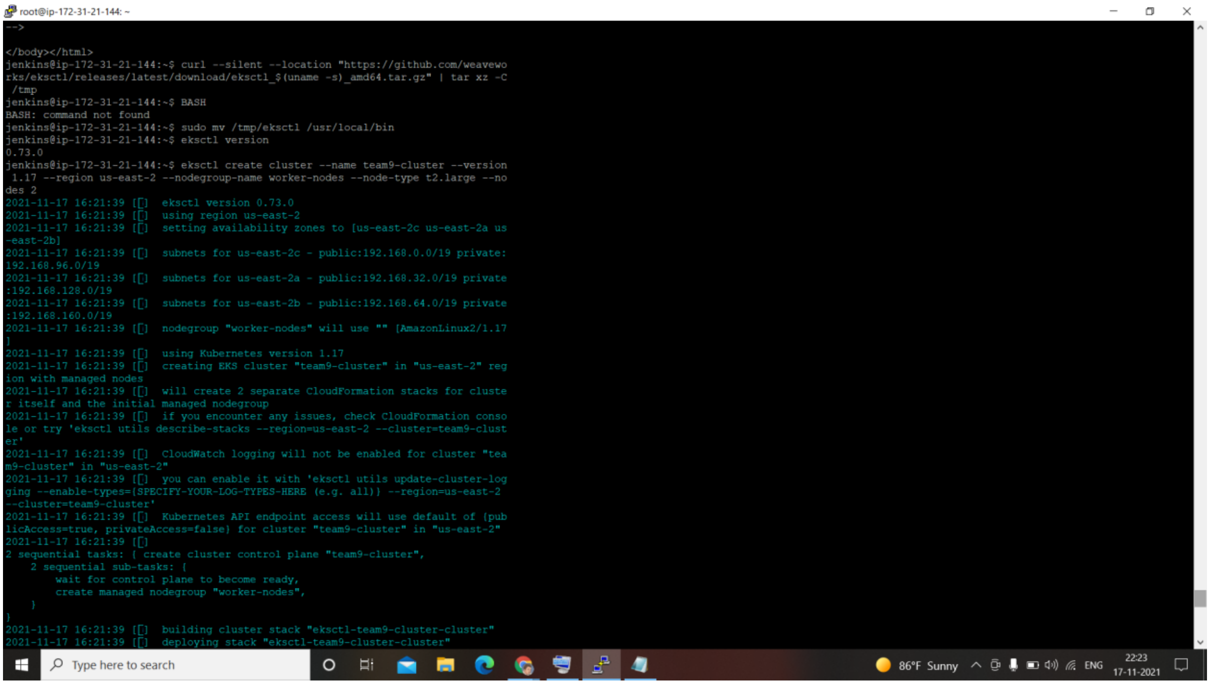
sudo mv /tmp/eksctl /usr/local/bin

then run eksctl command

eksctl create cluster --name team9-cluster --version 1.17 --region us-east-2 --nodegroup-name worker-nodes --node-type t2.large --nodes 2

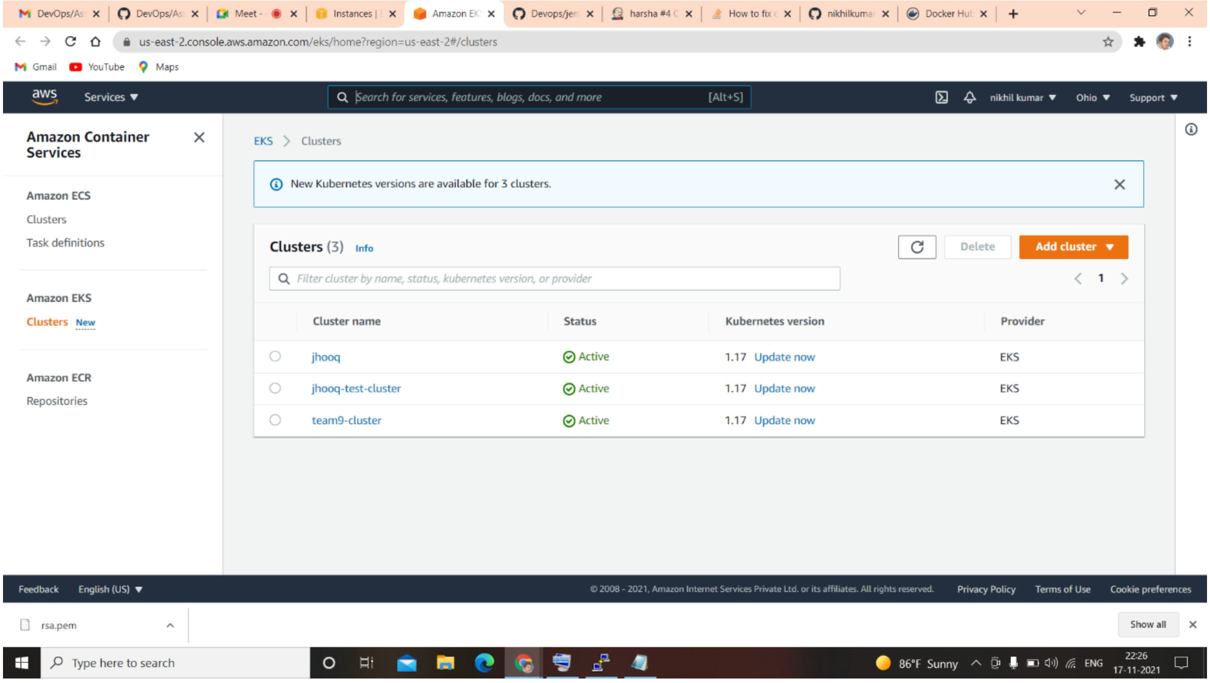


```
s to provide /usr/bin/rst2odt_prepstyles (rst2odt_prepstyles) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2pseudoxml to
provide /usr/bin/rst2pseudoxml (rst2pseudoxml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2s5 to provide
 /usr/bin/rst2s5 (rst2s5) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xetex to prov
ide /usr/bin/rst2xetex (rst2xetex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xml to provid
e /usr/bin/rst2xml (rst2xml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rstpep2html to pr
ovide /usr/bin/rstpep2html (rstpep2html) in auto mode
Setting up python3-botocore (1.16.19+repack-1ubuntu0.20.04.1) ...
Setting up python3-s3transfer (0.3.3-1) ...
Setting up awscli (1.18.69-1ubuntu0.20.04.1) ...
jenkins@ip-172-31-21-144:~$ aws --version
aws-cli/1.18.69 Python/3.8.10 Linux/5.11.0-1020-aws botocore/1.16.19
jenkins@ip-172-31-21-144:~$
jenkins@ip-172-31-21-144:~$ aws configure
AWS Access Key ID [None]: AKIAUTLMMBW5YMFP6CQS
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]:
jenkins@ip-172-31-21-144:~$ curl -LO "https://storage.googleapis.com/kubernetes-
release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/rele
ase/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 44.7M  100 44.7M    0     0  40.7M      0  0:00:01  0:00:01 --:--:-- 40.7M
jenkins@ip-172-31-21-144:~$ chmod +x ./kubectl
jenkins@ip-172-31-21-144:~$ sudo mv ./kubectl /usr/local/bin
jenkins@ip-172-31-21-144:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.3", GitCom
mit:"c92036820499fedefec0f847e2054d824aea6cd1", GitTreeState:"clean", BuildDate:
"2021-10-27T18:41:28Z", GoVersion:"go1.16.9", Compiler:"gc", Platform:"linux/amd
64"}
Error from server (Forbidden): <html><head><meta http-equiv='refresh' content='1
;url=/login?from=%2Fversion%3Ftimeout%3D32s'/><script>window.location.replace('/
login?from=%2Fversion%3Ftimeout%3D32s');</script></head><body style='background-
color:white; color:white;'>


Authentication required
<!--
-->

</body></html>
jenkins@ip-172-31-21-144:~$ curl --silent --location "https://github.com/weavewo
rks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C
 /tmp
jenkins@ip-172-31-21-144:~$ BASH
```



```
-->

</body></html>
jenkins@ip-172-31-21-144:~$ curl --silent --location "https://github.com/weavewo
rks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C
 /tmp
jenkins@ip-172-31-21-144:~$ BASH
BASH: command not found
jenkins@ip-172-31-21-144:~$ sudo mv /tmp/eksctl /usr/local/bin
jenkins@ip-172-31-21-144:~$ eksctl version
0.73.0
jenkins@ip-172-31-21-144:~$ eksctl create cluster --name team9-cluster --version
 1.17 --region us-east-2 --nodegroup-name worker-nodes --node-type t2.large --no
des 2
2021-11-17 16:21:39 [ℹ]  eksctl version 0.73.0
2021-11-17 16:21:39 [ℹ]  using region us-east-2
2021-11-17 16:21:39 [ℹ]  setting availability zones to [us-east-2c us-east-2a us
-east-2b]
2021-11-17 16:21:39 [ℹ]  subnets for us-east-2c - public:192.168.0.0/19 private:
192.168.96.0/19
2021-11-17 16:21:39 [ℹ]  subnets for us-east-2a - public:192.168.32.0/19 private
:192.168.128.0/19
2021-11-17 16:21:39 [ℹ]  subnets for us-east-2b - public:192.168.64.0/19 private
:192.168.160.0/19
2021-11-17 16:21:39 [ℹ]  nodegroup "worker-nodes" will use "" [AmazonLinux2/1.17
]
2021-11-17 16:21:39 [ℹ]  using Kubernetes version 1.17
2021-11-17 16:21:39 [ℹ]  creating EKS cluster "team9-cluster" in "us-east-2" reg
ion with managed nodes
2021-11-17 16:21:39 [ℹ]  will create 2 separate CloudFormation stacks for cluste
r itself and the initial managed nodegroup
2021-11-17 16:21:39 [ℹ]  if you encounter any issues, check CloudFormation conso
le or try 'eksctl utils describe-stacks --region=us-east-2 --cluster=team9-clust
er'
2021-11-17 16:21:39 [ℹ]  CloudWatch logging will not be enabled for cluster "tea
m9-cluster" in "us-east-2"
2021-11-17 16:21:39 [ℹ]  you can enable it with 'eksctl utils update-cluster-log
ging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-2
--cluster=team9-cluster'
2021-11-17 16:21:39 [ℹ]  Kubernetes API endpoint access will use default of {pub
licAccess=true, privateAccess=false} for cluster "team9-cluster" in "us-east-2"
2021-11-17 16:21:39 [ℹ]
2 sequential tasks: { create cluster control plane "team9-cluster",
    2 sequential sub-tasks: {
        wait for control plane to become ready,
        create managed nodegroup "worker-nodes",
    }
}
2021-11-17 16:21:39 [ℹ]  building cluster stack "eksctl-team9-cluster-cluster"
2021-11-17 16:21:39 [ℹ]  deploying stack "eksctl-team9-cluster-cluster"
```

Check for eks cluster creation in aws

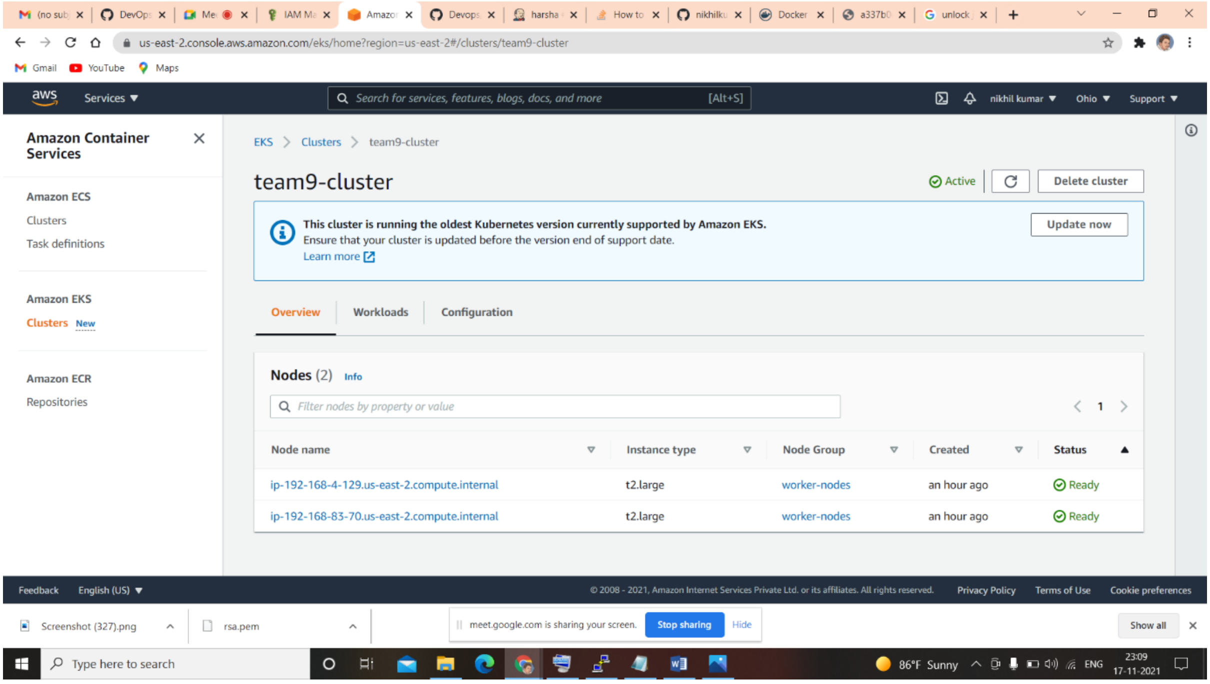


New Kubernetes versions are available for 3 clusters.

Clusters (3) Info

Filter cluster by name, status, kubernetes version, or provider

	Cluster name	Status	Kubernetes version	Provider
○	jhooq	Active	1.17 Update now	EKS
○	jhooq-test-cluster	Active	1.17 Update now	EKS
○	team9-cluster	Active	1.17 Update now	EKS



us-east-2.console.aws.amazon.com/eks/home?region=us-east-2#/clusters/team9-cluster

Amazon Container Services

Amazon ECS
Clusters
Task definitions

Amazon EKS
Clusters New

Amazon ECR
Repositories

EKS > Clusters > team9-cluster

team9-cluster

Active | Delete cluster

This cluster is running the oldest Kubernetes version currently supported by Amazon EKS.
Ensure that your cluster is updated before the version end of support date.
Learn more

Update now

Overview | Workloads | Configuration

Nodes (2) Info

Filter nodes by property or value

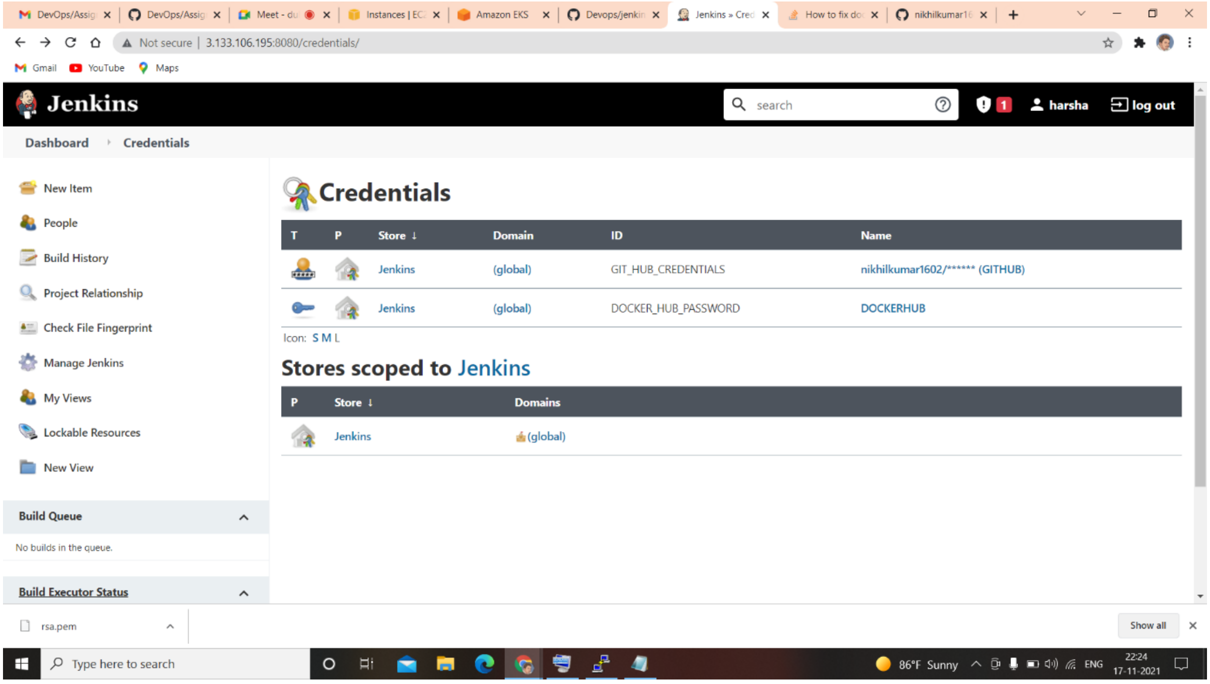
Node name	Instance type	Node Group	Created	Status
ip-192-168-4-129.us-east-2.compute.internal	t2.large	worker-nodes	an hour ago	✓ Ready
ip-192-168-83-70.us-east-2.compute.internal	t2.large	worker-nodes	an hour ago	✓ Ready

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

meet.google.com is sharing your screen. Stop sharing Hide

86°F Sunny 23:09 17-11-2021

Add GitHub and Docker hub credentials in credential manager



Not secure | 3.133.106.195:8080/credentials/

Jenkins

Dashboard > Credentials

New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Lockable Resources
New View

Build Queue
No builds in the queue.

Build Executor Status

Credentials

T	P	Store ↓	Domain	ID	Name
		Jenkins	(global)	GIT_HUB_CREDENTIALS	nikhilkumar1602/****** (GITHUB)
		Jenkins	(global)	DOCKER_HUB_PASSWORD	DOCKERHUB

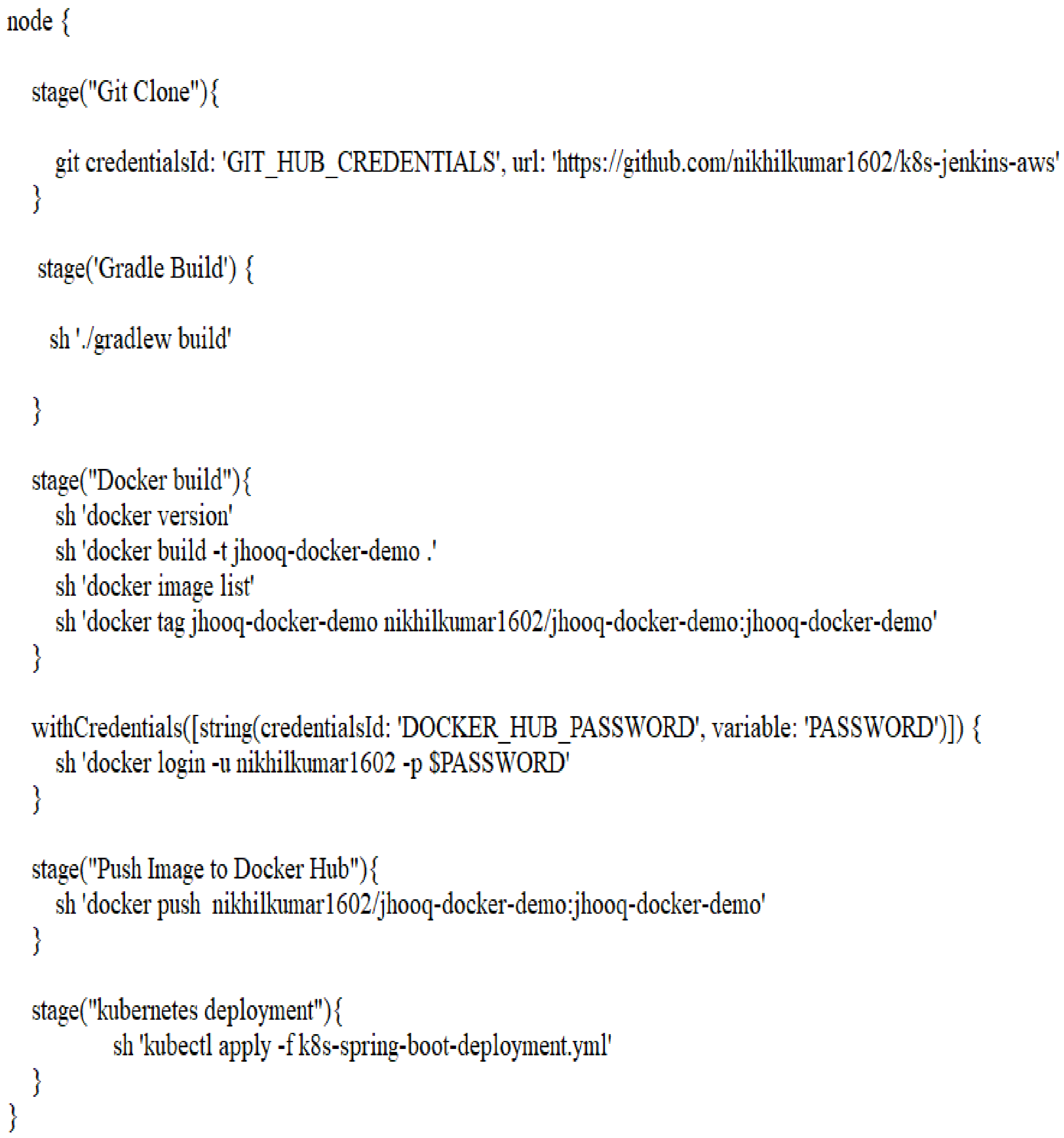
Icon: S M L

Stores scoped to Jenkins

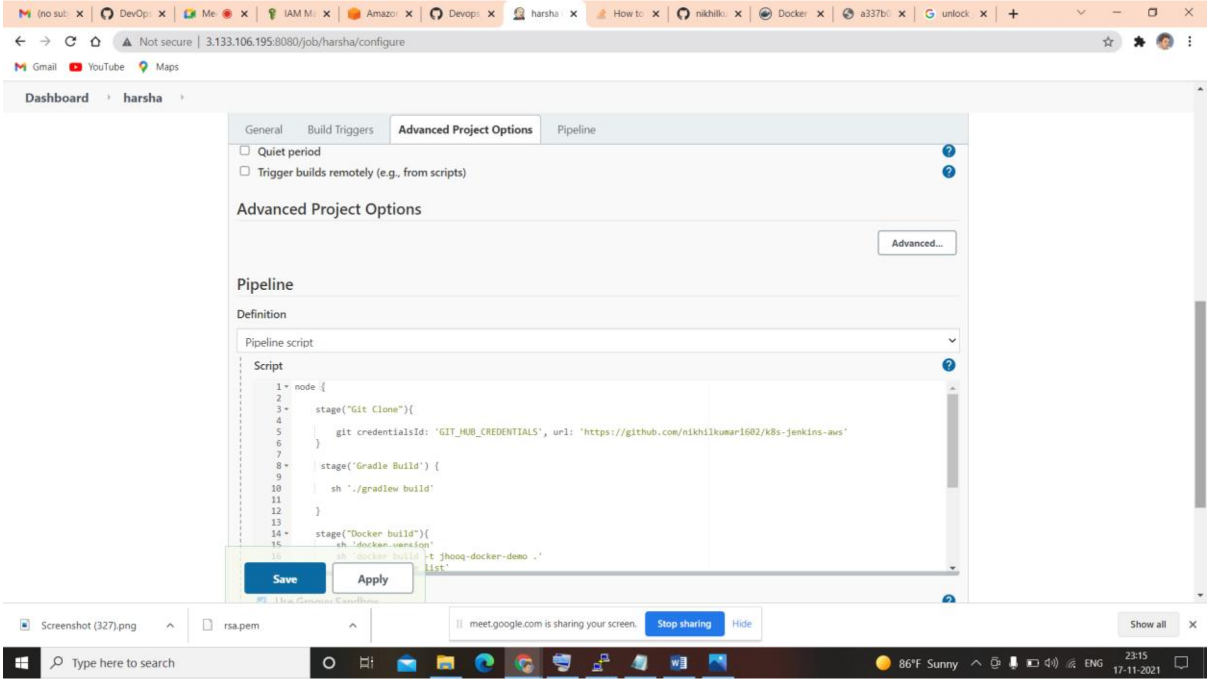
P	Store ↓	Domains
	Jenkins	(global)

86°F Sunny 22:24 17-11-2021

now we can start writing out the Jenkins pipeline for deploying the Spring Boot Application into the Kubernetes Cluster. Jenkins stage-



```
node {

    stage("Git Clone"){

        git credentialsId: 'GIT_HUB_CREDENTIALS', url: 'https://github.com/nikhilkumar1602/k8s-jenkins-aws'
    }

     stage('Gradle Build') {

       sh './gradlew build'

    }

    stage("Docker build"){
        sh 'docker version'
        sh 'docker build -t jhooq-docker-demo .'
        sh 'docker image list'
        sh 'docker tag jhooq-docker-demo nikhilkumar1602/jhooq-docker-demo:jhooq-docker-demo'
    }

    withCredentials([string(credentialsId: 'DOCKER_HUB_PASSWORD', variable: 'PASSWORD')]) {
        sh 'docker login -u nikhilkumar1602 -p $PASSWORD'
    }

    stage("Push Image to Docker Hub"){
        sh 'docker push  nikhilkumar1602/jhooq-docker-demo:jhooq-docker-demo'
    }

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



Not secure | 3.133.106.195:8080/job/harsha/configure

Dashboard › harsha ›

General | Build Triggers | Advanced Project Options | Pipeline

□ Quiet period

□ Trigger builds remotely (e.g., from scripts)

Advanced Project Options

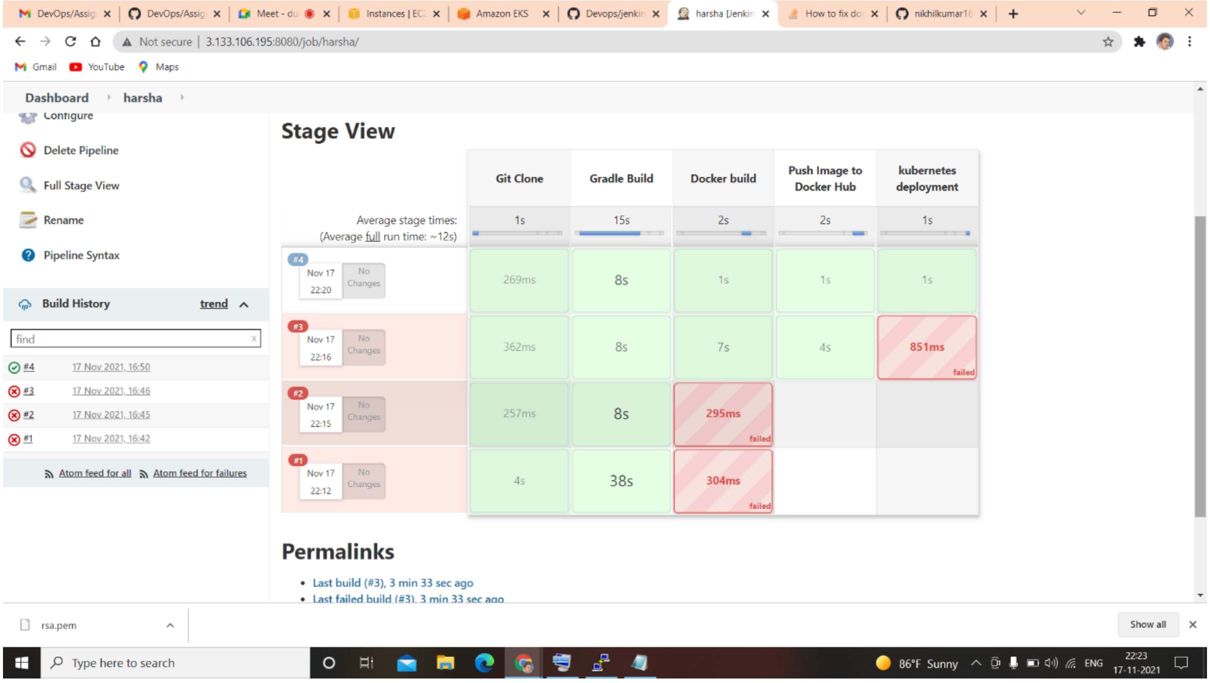
Pipeline

Definition: Pipeline script

Script

```
node {

    stage("Git Clone"){

        git credentialsId: 'GIT_HUB_CREDENTIALS', url: 'https://github.com/nikhilkumar1602/k8s-jenkins-aws'
    }

     stage('Gradle Build') {

       sh './gradlew build'

    }

    stage("Docker build"){
        sh 'docker version'
        sh 'docker build -t jhooq-docker-demo .'
                          list'
```

after adding pipeline script, build the pipeline



Not secure | 3.133.106.195:8080/job/harsha/

Dashboard › harsha ›

Stage View

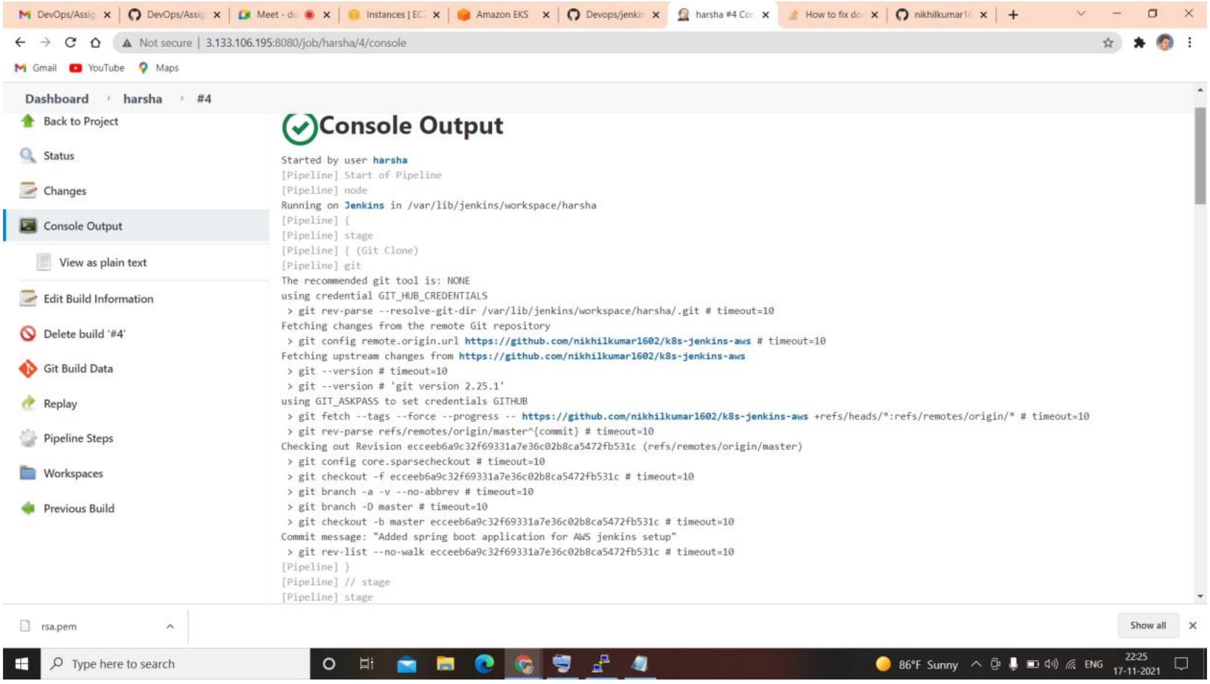
	Git Clone	Gradle Build	Docker build	Push Image to Docker Hub	kubernetes deployment
Average stage times: (Average full run time: ~12s)	1s	15s	2s	2s	1s
#4 Nov 17 22:20 No Changes	269ms	8s	1s	1s	1s
#3 Nov 17 22:16 No Changes	362ms	8s	7s	4s	851ms failed
#2 Nov 17 22:15 No Changes	257ms	8s	295ms failed		
#1 Nov 17 22:12 No Changes	4s	38s	304ms failed		

Build History

#4	17 Nov 2021, 16:50
#3	17 Nov 2021, 16:46
#2	17 Nov 2021, 16:45
#1	17 Nov 2021, 16:42

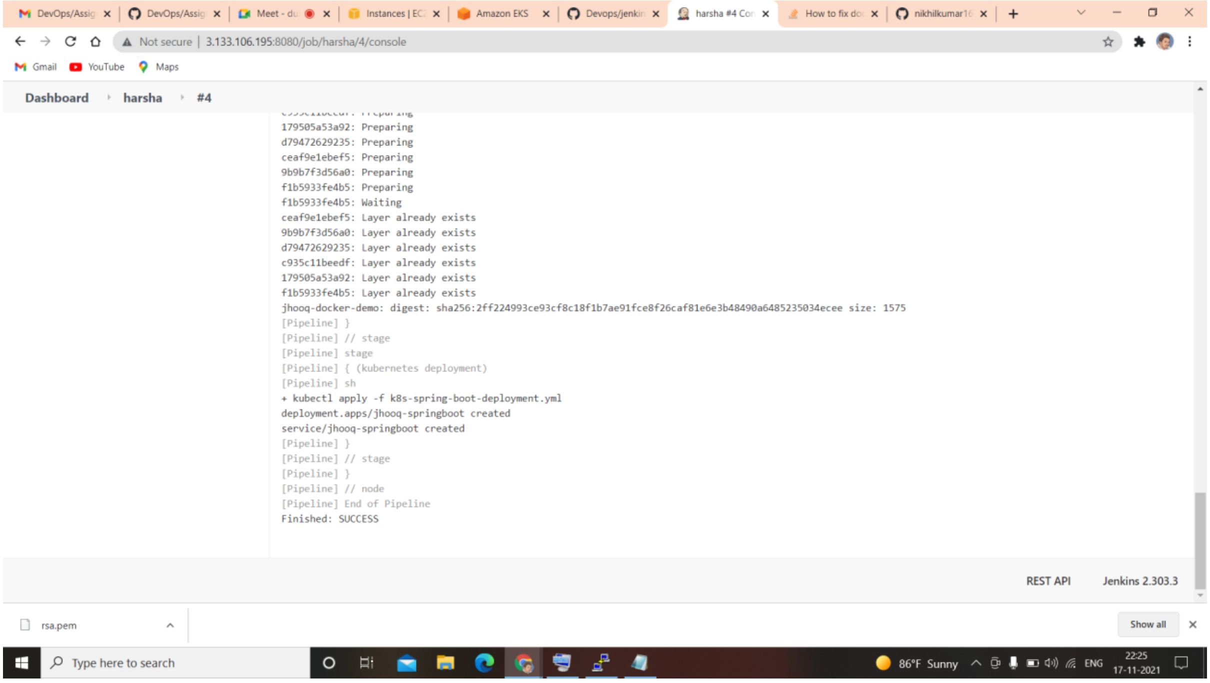
Permalinks

- Last build (#3), 3 min 33 sec ago
- Last failed build (#3), 3 min 33 sec ago



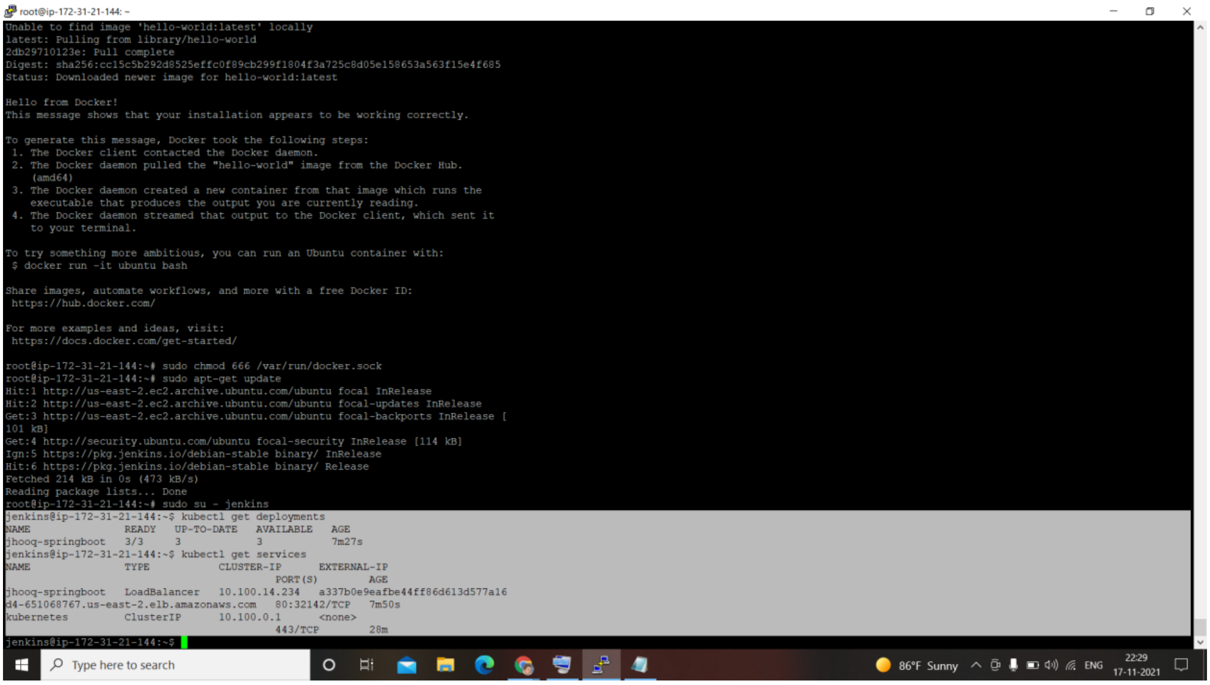
Console Output

```
Started by user harsha
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/harsha
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Git Clone)
[Pipeline] git
The recommended git tool is: NONE
using credential GIT_HUB_CREDENTIALS
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/harsha/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/nikhilkumar1602/k8s-jenkins-aws # timeout=10
Fetching upstream changes from https://github.com/nikhilkumar1602/k8s-jenkins-aws
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
using GIT_ASKPASS to set credentials GITHUB
 > git fetch --tags --force --progress -- https://github.com/nikhilkumar1602/k8s-jenkins-aws +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision ecceeb6a9c32f69331a7e36c02b8ca5472fb531c (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f ecceeb6a9c32f69331a7e36c02b8ca5472fb531c # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D master # timeout=10
 > git checkout -b master ecceeb6a9c32f69331a7e36c02b8ca5472fb531c # timeout=10
Commit message: "Added spring boot application for AWS jenkins setup"
 > git rev-list --no-walk ecceeb6a9c32f69331a7e36c02b8ca5472fb531c # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
```

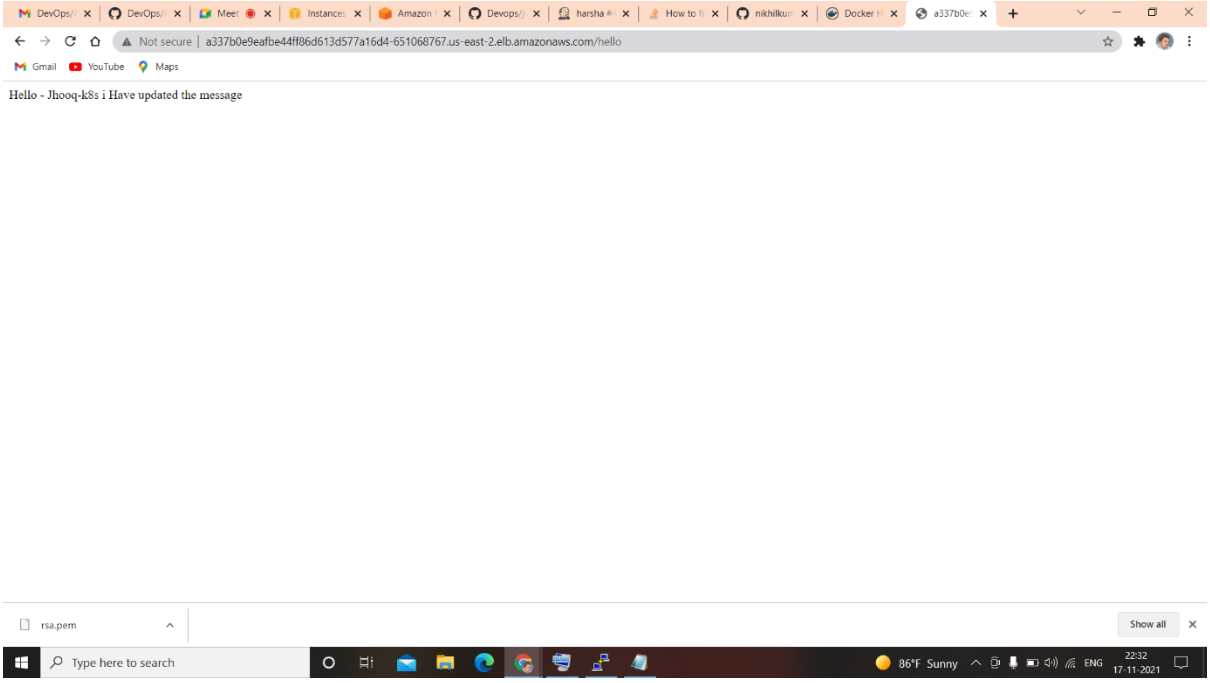


```
179505a53a92: Preparing
d79472629235: Preparing
ceaf9e1ebef5: Preparing
9b9b7f3d56a0: Preparing
f1b5933fe4b5: Preparing
f1b5933fe4b5: Waiting
ceaf9e1ebef5: Layer already exists
9b9b7f3d56a0: Layer already exists
d79472629235: Layer already exists
c935c11beedf: Layer already exists
179505a53a92: Layer already exists
f1b5933fe4b5: Layer already exists
jhooq-docker-demo: digest: sha256:2ff224993ce93cf8c18f1b7ae91fce8f26caf81e6e3b48490a6485235034ecee size: 1575
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (kubernetes deployment)
[Pipeline] sh
+ kubectl apply -f k8s-spring-boot-deployment.yml
deployment.apps/jhooq-springboot created
service/jhooq-springboot created
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

verify the Kubernetes deployment and service with kubectl command .e.g kubectl get deployments, kubectl get service. You can access the rest end point from browser using the EXTERNAL-IP address.



```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:cc15c5b292d8525effc0f89cb299f1804f3a725c8d05e158653a563f15e4f685
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

root@ip-172-31-21-144:~# sudo chmod 666 /var/run/docker.sock
root@ip-172-31-21-144:~# sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [
101 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release
Fetched 214 kB in 0s (473 kB/s)
Reading package lists... Done
root@ip-172-31-21-144:~# sudo su - jenkins
jenkins@ip-172-31-21-144:~$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
jhooq-springboot    3/3     3            3           7m27s
jenkins@ip-172-31-21-144:~$ kubectl get services
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP
                                             PORT(S)        AGE
jhooq-springboot    LoadBalancer   10.100.14.234   a337b0e9eafbe44ff86d613d577a16
d4-651068767.us-east-2.elb.amazonaws.com    80:32142/TCP   7m50s
kubernetes          ClusterIP      10.100.0.1      <none>
                                             443/TCP        28m
jenkins@ip-172-31-21-144:~$
```



Hello - Jhooq-k8s i Have updated the message

--------- Team 9 --------------