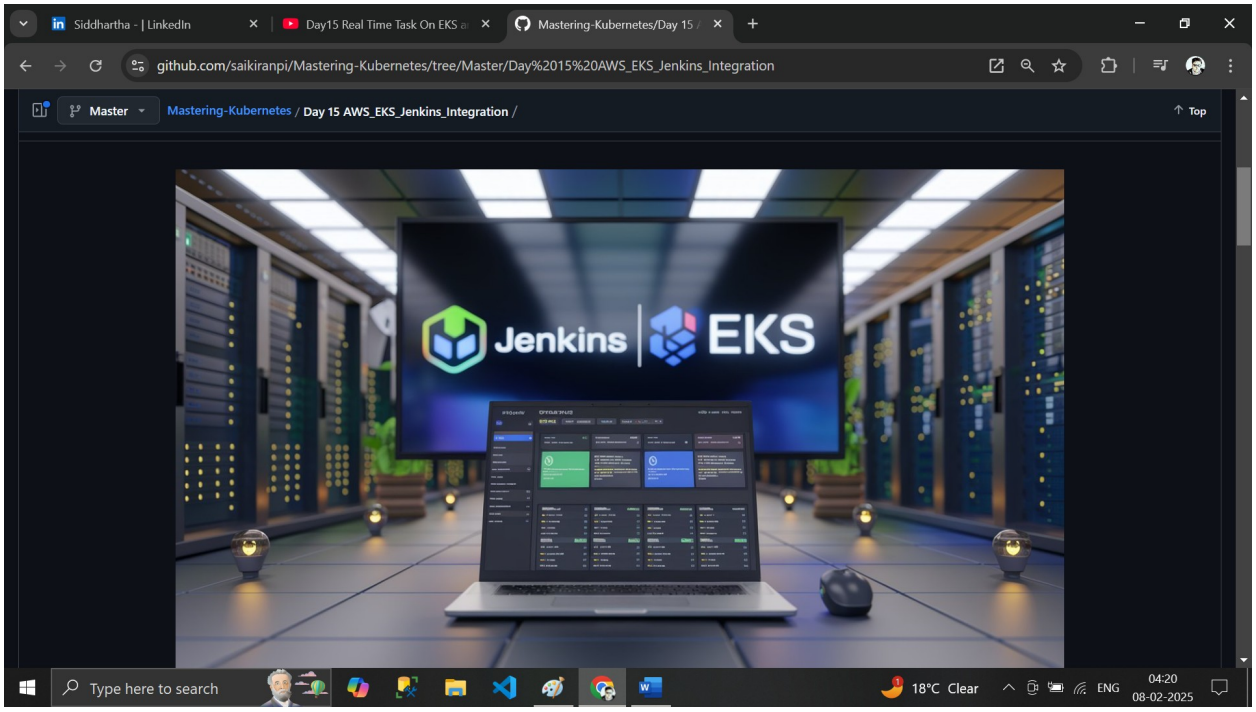
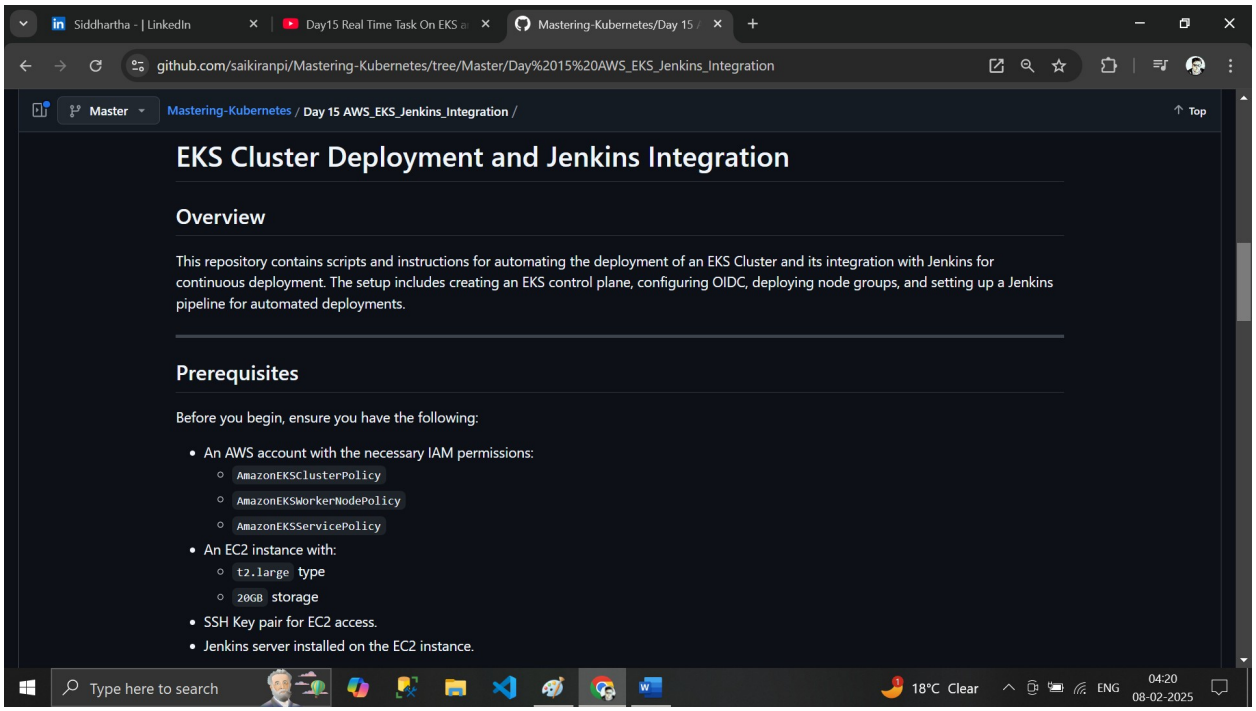
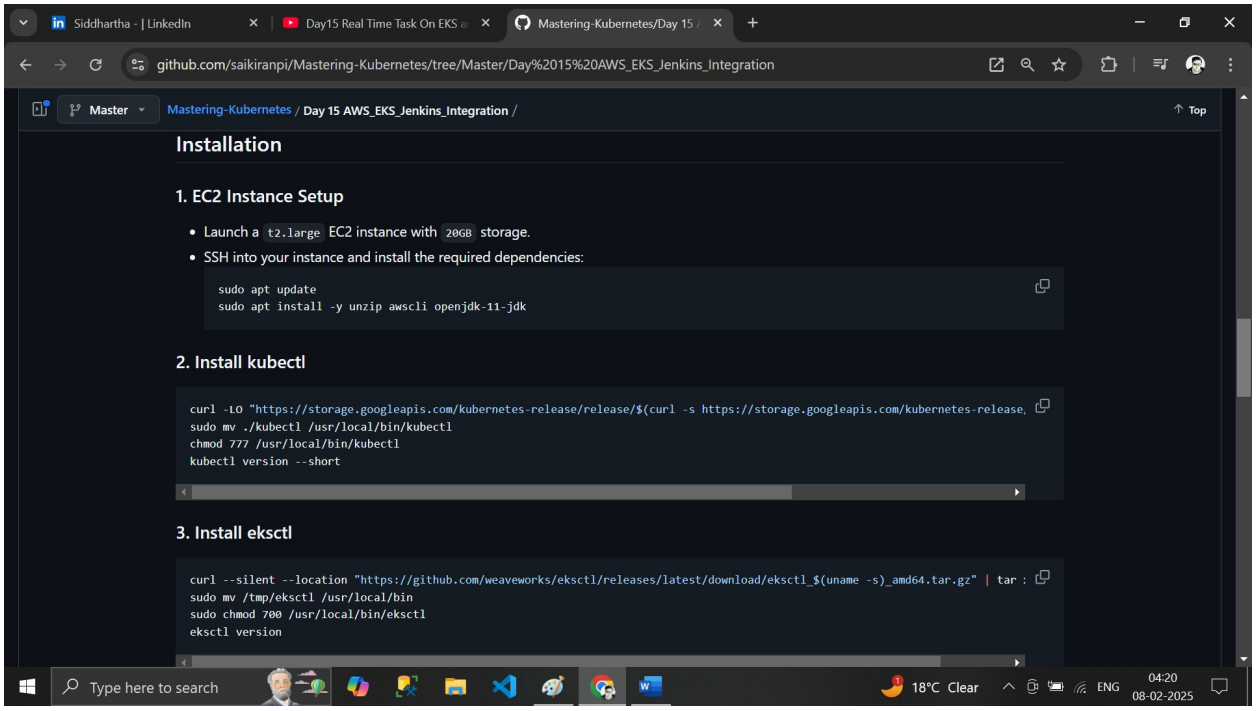
15 Real Time Task On EKS and Jenkins End to End Pipeline Integration

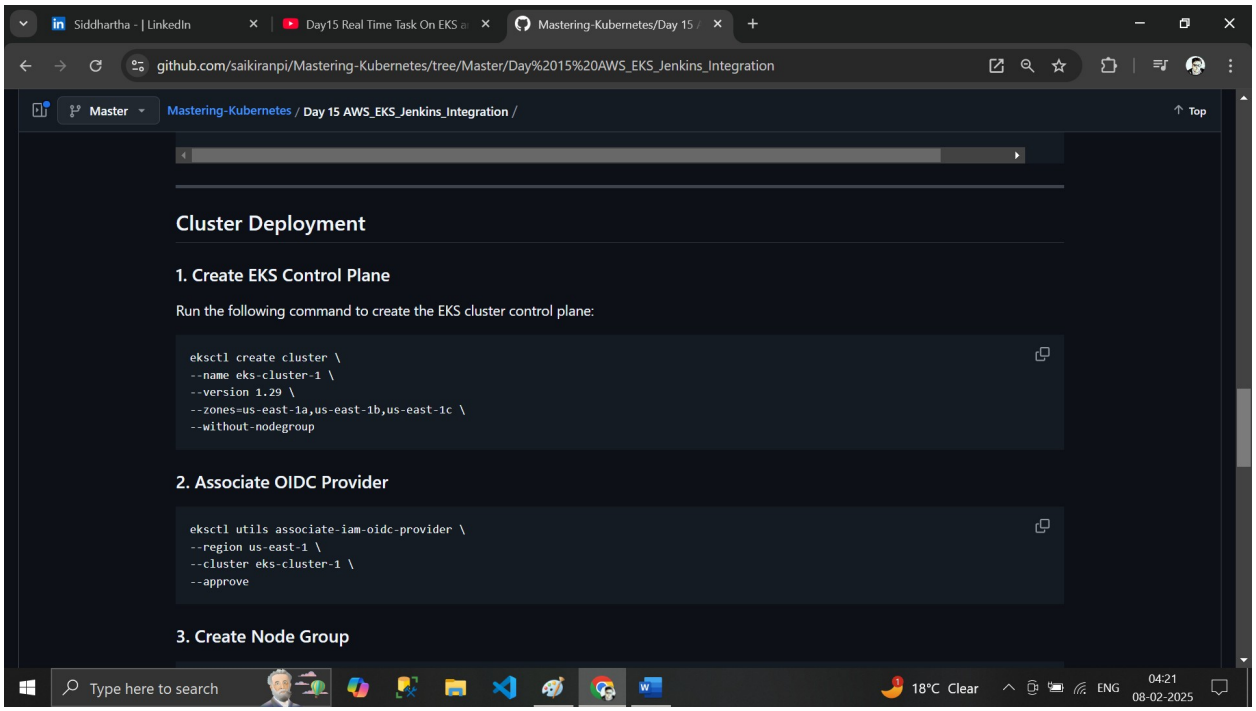
how to set up an Amazon EKS cluster and connect it with Jenkins for automated deployments. We'll start by launching an EC2 instance, setting up the necessary tools, and then deploying an EKS cluster.Finally, I'll walk you through integrating Jenkins with GitHub to automatically trigger deployments based on branch changes. Whether you're new to DevOps or just looking to automateyour workflow,this step-by-step guide will help you get started quickly!

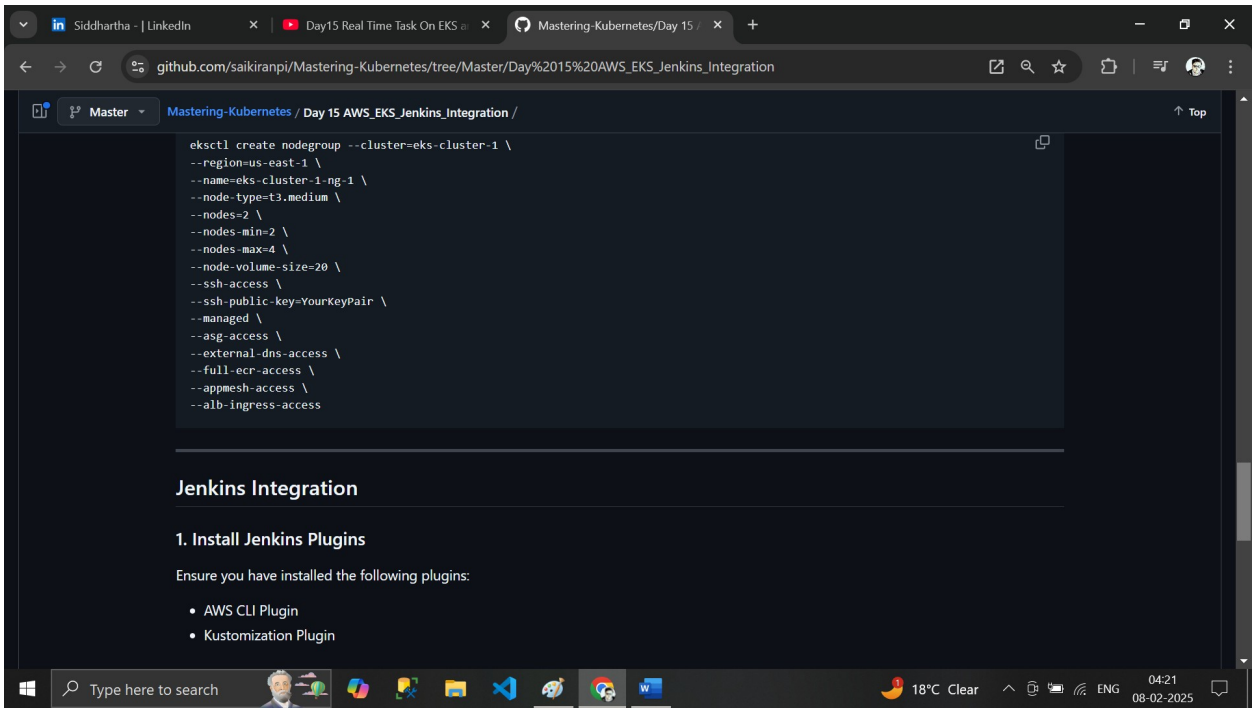
1. **Intro**
2. **Creating T2 Large Instance with 20GB**
3. **Creating IAM User**
4. **Installing Unzip and AwsCLi**
5. **Installing Kubectl and Ekctl**
6. **Installing Control plan**

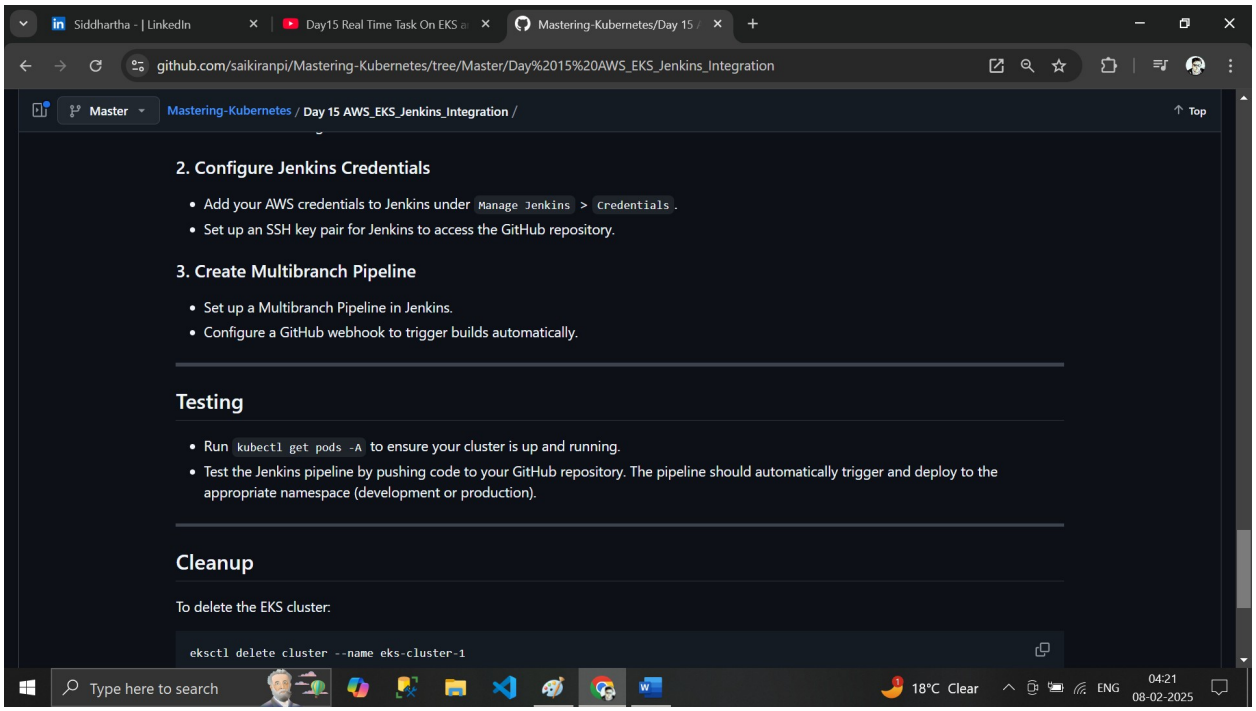



Siddhartha - | LinkedIn
Day15 Real Time Task On EKS
Mastering-Kubernetes/Day 15
github.com/saikiranpi/Mastering-Kubernetes/tree/Master/Day%2015%20AWS_EKS_Jenkins_Integration
Master
Mastering-Kubernetes / Day 15 AWS_EKS_Jenkins_Integration /
Top
EKS Cluster Deployment and Jenkins Integration
Overview
This repository contains scripts and instructions for automating the deployment of an EKS Cluster and its integration with Jenkins for continuous deployment. The setup includes creating an EKS control plane, configuring OIDC, deploying node groups, and setting up a Jenkins pipeline for automated deployments.
Prerequisites
Before you begin, ensure you have the following:
An AWS account with the necessary IAM permissions:
AmazonEKSClusterPolicy
AmazonEKSWorkerNodePolicy
AmazonEKSServicePolicy
An EC2 instance with:
t2.large type
20GB storage
SSH Key pair for EC2 access.
Jenkins server installed on the EC2 instance.
Type here to search
18°C Clear
ENG
04:20
08-02-2025


Siddhartha - | LinkedIn
Day15 Real Time Task On EKS
Mastering-Kubernetes/Day 15
github.com/saikiranpi/Mastering-Kubernetes/tree/Master/Day%2015%20AWS_EKS_Jenkins_Integration
Master
Mastering-Kubernetes / Day 15 AWS_EKS_Jenkins_Integration /
Top
Installation
1. EC2 Instance Setup
Launch a t2.large EC2 instance with 20GB storage.
SSH into your instance and install the required dependencies:
sudo apt update
sudo apt install -y unzip awscli openjdk-11-jdk
2. Install kubectl
curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release,
sudo mv ./kubectl /usr/local/bin/kubectl
chmod 777 /usr/local/bin/kubectl
kubectl version --short
3. Install eksctl
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar :
sudo mv /tmp/eksctl /usr/local/bin
sudo chmod 700 /usr/local/bin/eksctl
eksctl version
Type here to search
18°C Clear
ENG
04:20
08-02-2025


Siddhartha - | LinkedIn
Day15 Real Time Task On EKS
Mastering-Kubernetes/Day 15
github.com/saikiranpi/Mastering-Kubernetes/tree/Master/Day%2015%20AWS_EKS_Jenkins_Integration
Master
Mastering-Kubernetes / Day 15 AWS_EKS_Jenkins_Integration /
Top
Cluster Deployment
1. Create EKS Control Plane
Run the following command to create the EKS cluster control plane:
eksctl create cluster \
--name eks-cluster-1 \
--version 1.29 \
--zones=us-east-1a,us-east-1b,us-east-1c \
--without-nodegroup
2. Associate OIDC Provider
eksctl utils associate-iam-oidc-provider \
--region us-east-1 \
--cluster eks-cluster-1 \
--approve
3. Create Node Group
Type here to search
18°C Clear
ENG
04:21
08-02-2025


Siddhartha - | LinkedIn
Day15 Real Time Task On EKS
Mastering-Kubernetes/Day 15
github.com/saikiranpi/Mastering-Kubernetes/tree/Master/Day%2015%20AWS_EKS_Jenkins_Integration
Master
Mastering-Kubernetes / Day 15 AWS_EKS_Jenkins_Integration /
Top
eksctl create nodegroup --cluster=eks-cluster-1 \
--region=us-east-1 \
--name=eks-cluster-1-ng-1 \
--node-type=t3.medium \
--nodes=2 \
--nodes-min=2 \
--nodes-max=4 \
--node-volume-size=20 \
--ssh-access \
--ssh-public-key=YourKeyPair \
--managed \
--asg-access \
--external-dns-access \
--full-ecr-access \
--appmesh-access \
--alb-ingress-access
Jenkins Integration
1. Install Jenkins Plugins
Ensure you have installed the following plugins:
AWS CLI Plugin
Kustomization Plugin
Type here to search
18°C Clear
ENG
04:21
08-02-2025

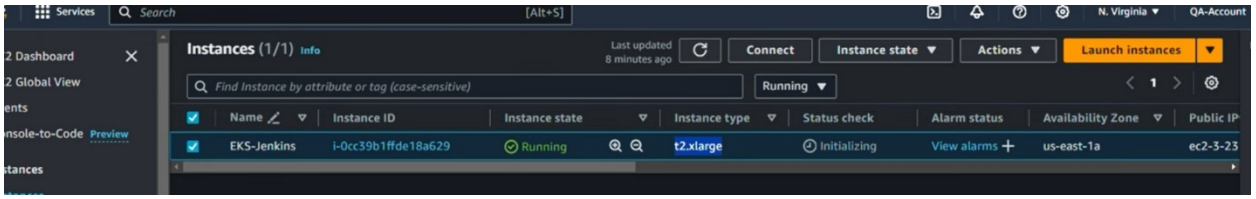


 # Practical

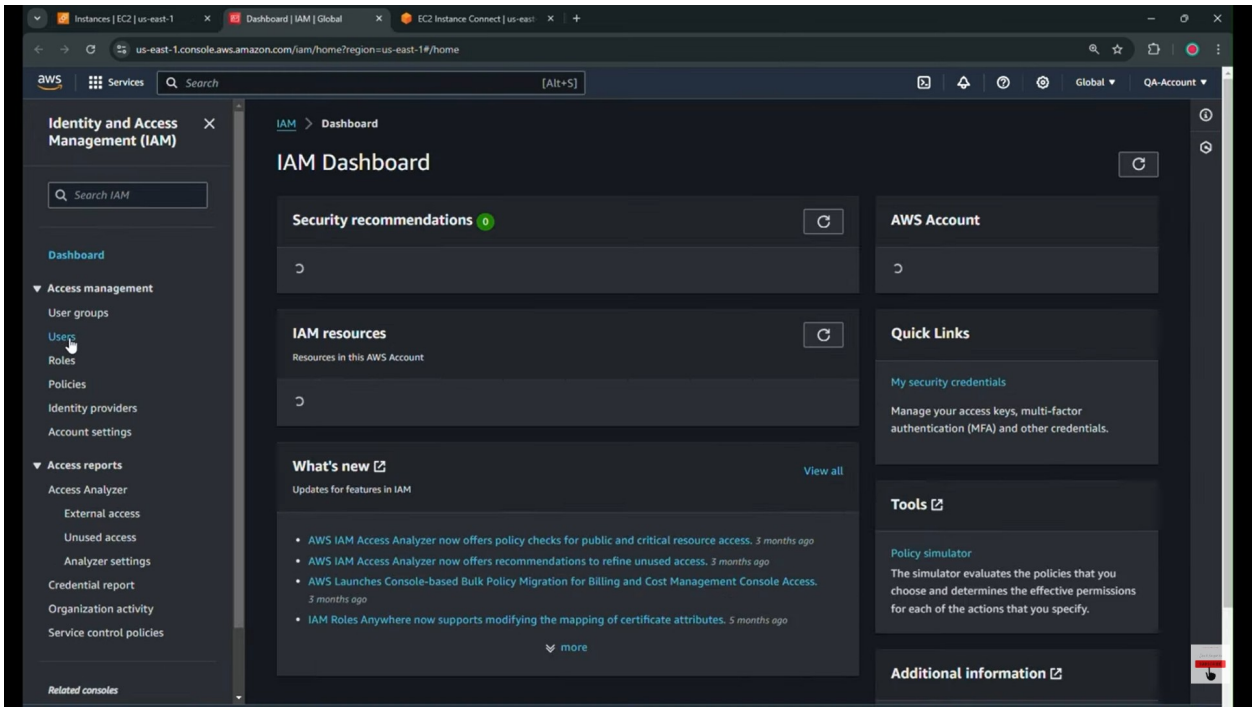
 # steps

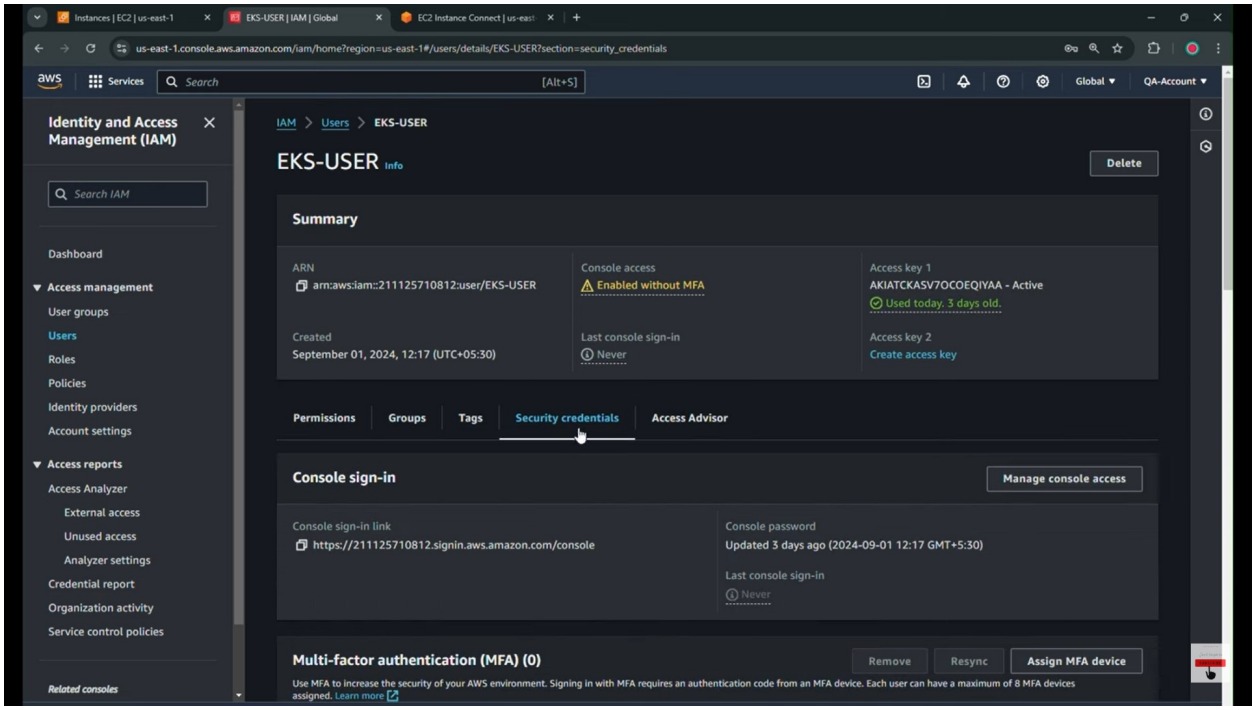
```
# Create 1 t2-Large Instance with 20gb

# Create IAM USER with Administrator access or individual access anything is fine
# AmazonEKSClusterPolicy
# AmazonEKSWorkerNodePolicy
# AmazonEKSServicePolicy
```

EC 2 instance

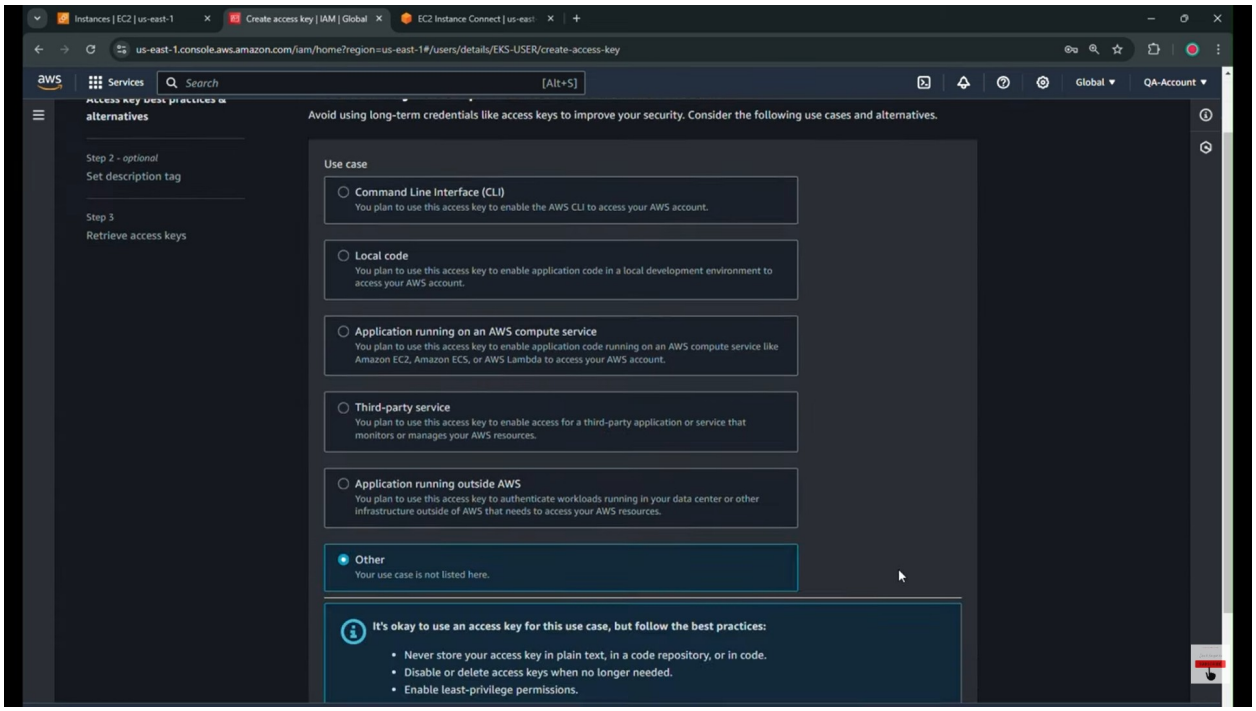


 # IAM user





 # security credentials



 # access+ secret key

 # step

```
# Install unzip and AWSCLI

#sudo apt update && sudo apt install unzip -y
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

```
# aws configure and pass the keys.
```

```
root@ip-10-0-9-146:~# aws configure
AWS Access Key ID [****************IYAA]:
AWS Secret Access Key [****************91T7]:
Default region name [us-east-1]:
Default output format [json]:
```

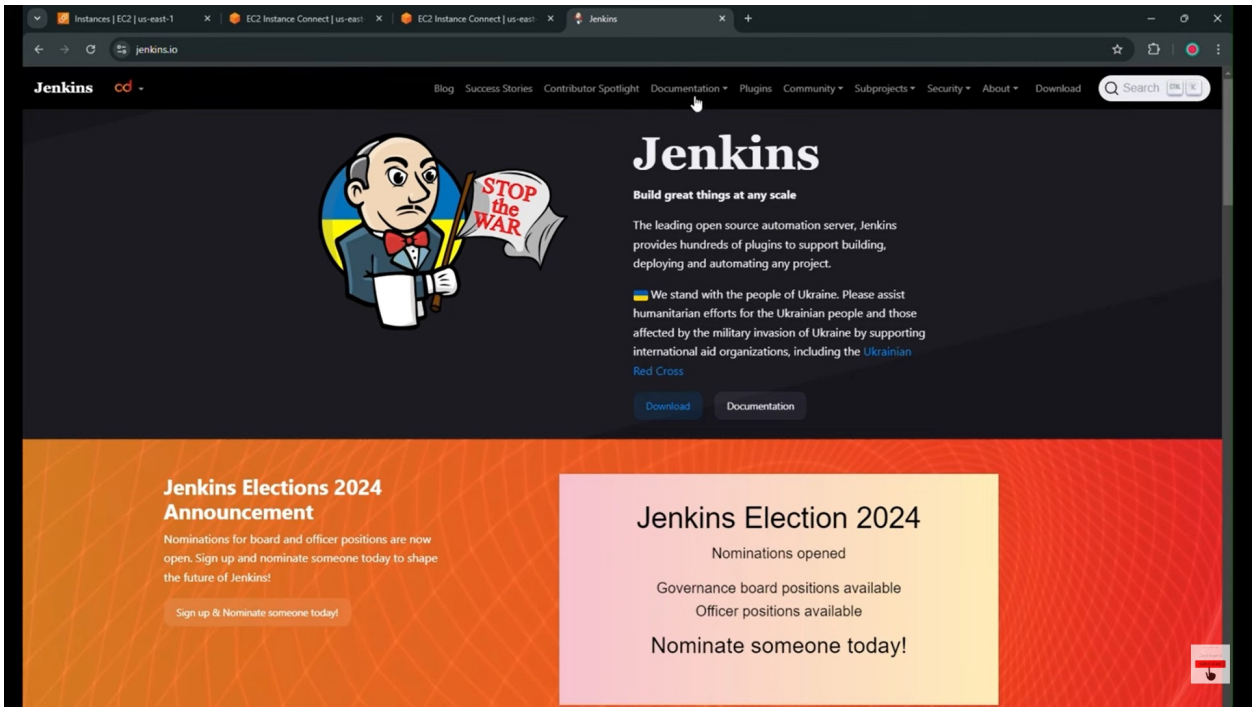
```
root@ip-10-0-9-146:~# aws s3 ls
2024-07-23 17:38:35 cloudvishwakarma.in
root@ip-10-0-9-146:~#
```

 # step

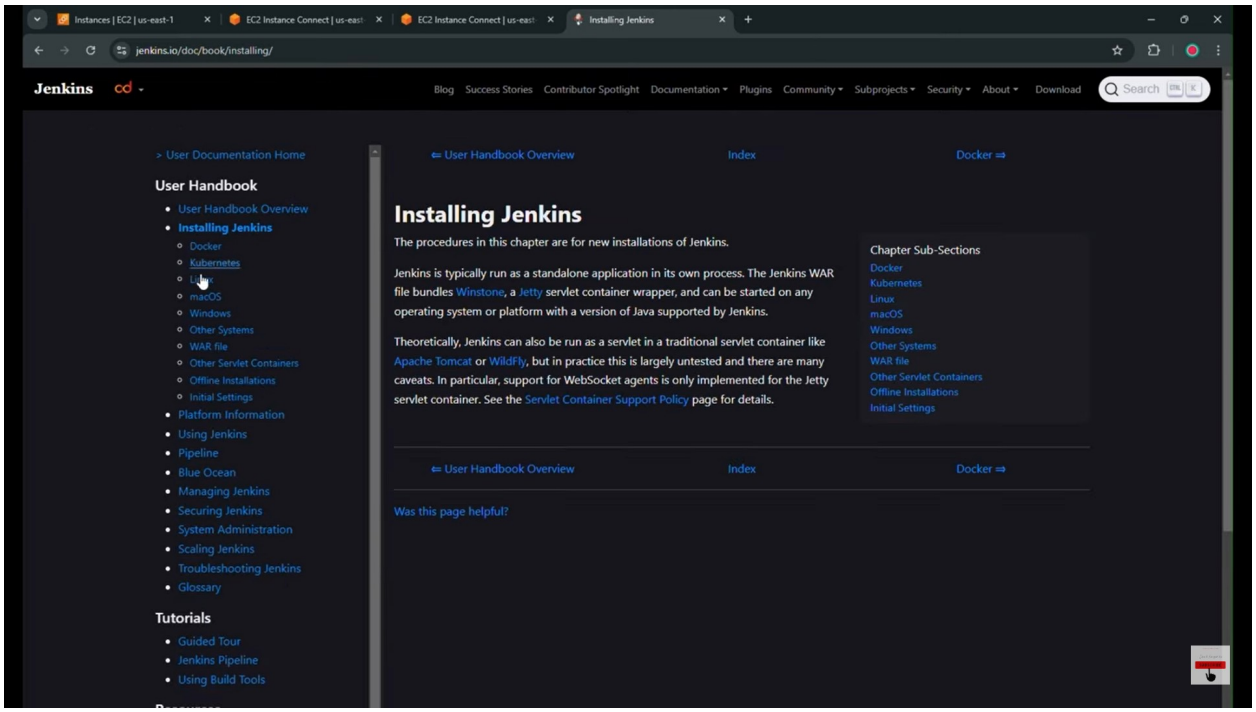
```
# install kubectl

curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/
sudo mv ./kubectl /usr/local/bin/kubectl
chmod 777 /usr/local/bin/kubectl
kubectl version -short

# install eksctl

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tm
sudo mv /tmp/eksctl /usr/local/bin
sudo chmod 700 /usr/local/bin/eksctl
eksctl version

#deploy control plane to delete ""eksctl delete cluster --name eks-cluster-1"""

eksctl create cluster \
--name eks-cluster-1 \
--version 1.29 \
--zones=us-east-1a,us-east-1b,us-east-1c \
--without-nodegroup
```

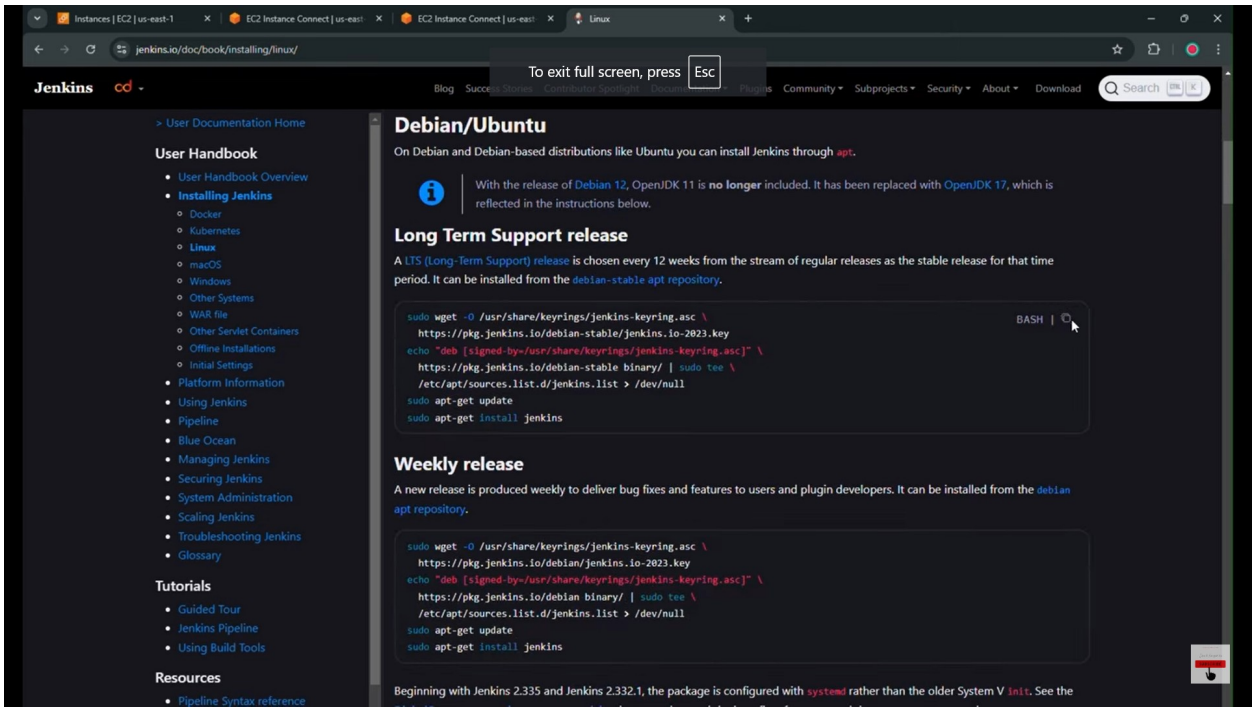
```
#Install java & jenkins

sudo apt install -y openjdk-11-jdk
```

```
root@ip-10-0-9-146:~# sudo apt install -y openjdk-11-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig-config fonts-dejavu-core fonts-dejavu-extra
  gsettings-desktop-schemas java-common libasound2 libasound2-data libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data
  libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcups2 libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
  libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libice-dev
  libice6 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm15 libpciaccess0 libpcsclite1 libpthread-stubs0-dev libsensors-config libsensors5 libsm-dev libsm6 libx11-dev
  libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0
  libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt-dev
  libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless session-migration x11-common x11-utils
  x11proto-dev xorg-sgml-doctools xtrans-dev
```

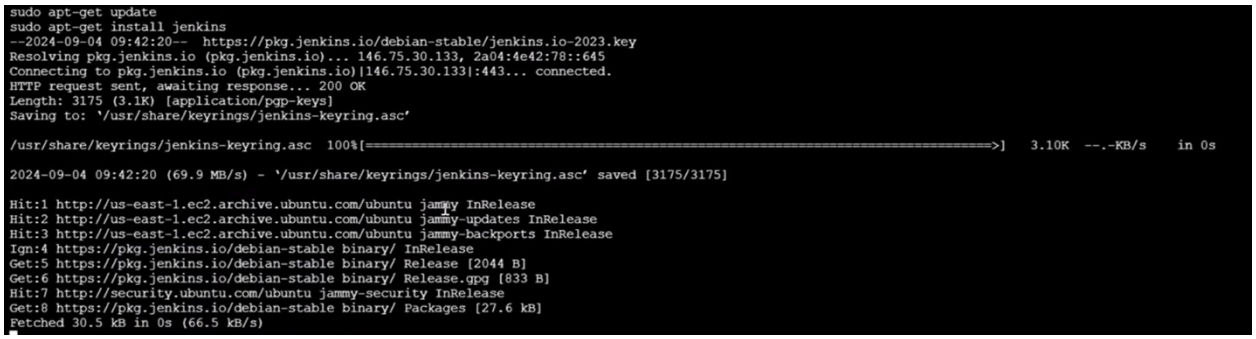


 # documentation





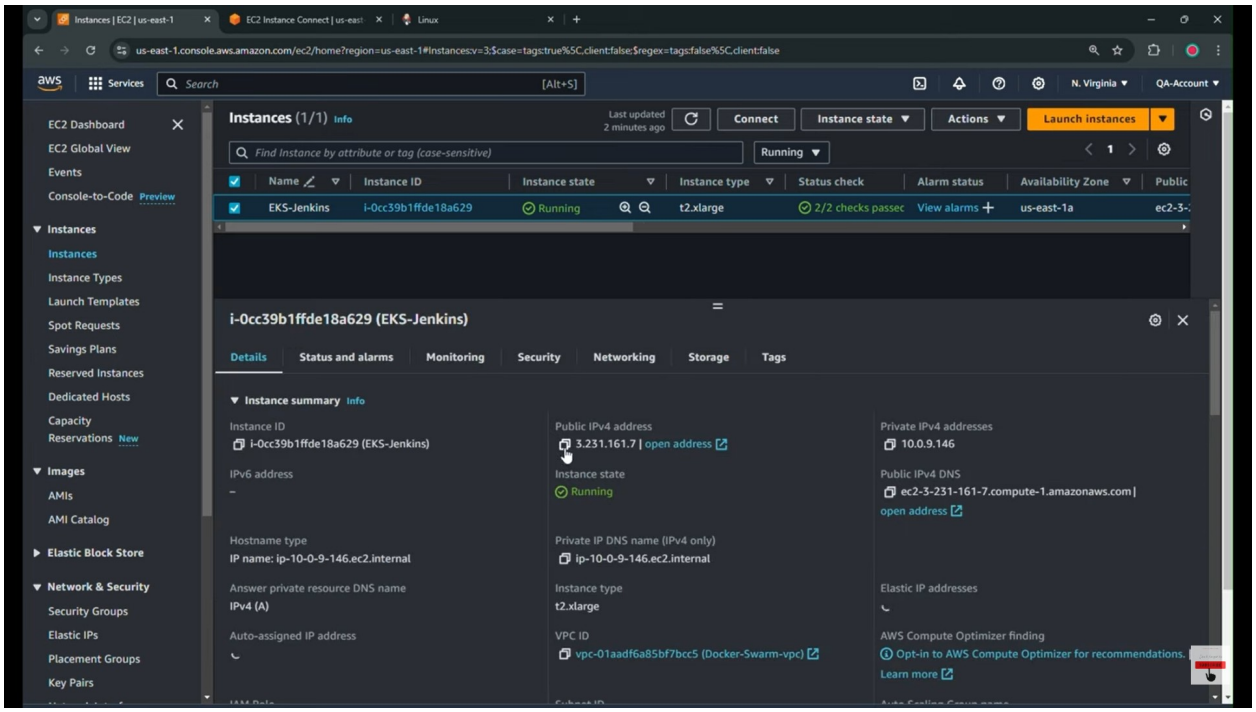
 # now install Jenkins



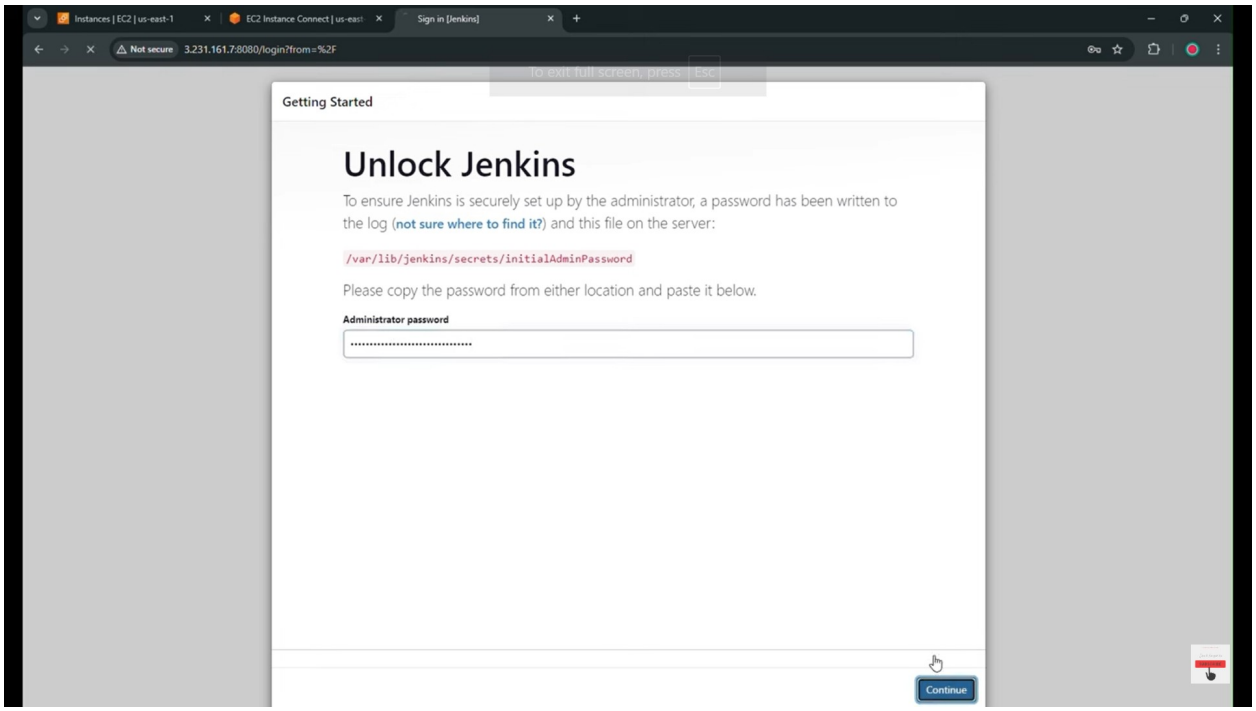
 # Jenkins up + running

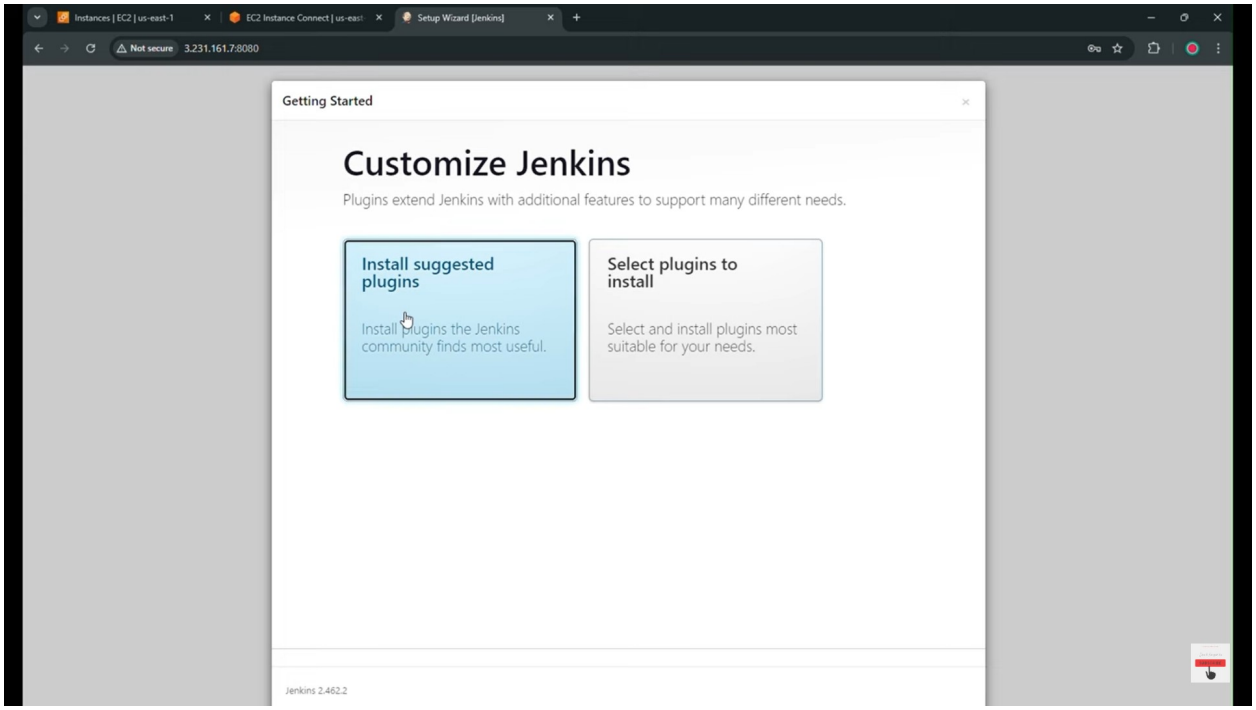
```
root@ip-10-0-9-146:~# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2024-09-04 09:42:37 UTC; 30s ago
   Main PID: 6126 (java)
      Tasks: 52 (limit: 19159)
     Memory: 665.8M
        CPU: 17.454s
     CGroup: /system.slice/jenkins.service
             └─6126 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Sep 04 09:42:35 ip-10-0-9-146 jenkins[6126]: f1255038fda046bb97920cd9ae3b378e
Sep 04 09:42:35 ip-10-0-9-146 jenkins[6126]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 04 09:42:35 ip-10-0-9-146 jenkins[6126]: ************************************************************
Sep 04 09:42:35 ip-10-0-9-146 jenkins[6126]: ************************************************************
Sep 04 09:42:35 ip-10-0-9-146 jenkins[6126]: ************************************************************
Sep 04 09:42:37 ip-10-0-9-146 jenkins[6126]: 2024-09-04 09:42:37.640+0000 [id=36]        INFO        jenkins.InitReactorRunner$1#onAttained: Completed initializat
Sep 04 09:42:37 ip-10-0-9-146 jenkins[6126]: 2024-09-04 09:42:37.663+0000 [id=23]        INFO        hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and r
Sep 04 09:42:37 ip-10-0-9-146 systemd[1]: Started Jenkins Continuous Integration Server.
Sep 04 09:42:37 ip-10-0-9-146 jenkins[6126]: 2024-09-04 09:42:37.684+0000 [id=52]        INFO        h.m.DownloadService$Downloadable#load: Obtained the updated d
Sep 04 09:42:37 ip-10-0-9-146 jenkins[6126]: 2024-09-04 09:42:37.685+0000 [id=52]        INFO        hudson.util.Retrier#start: Performed the action check updates
lines 1-20/20 (END)
```

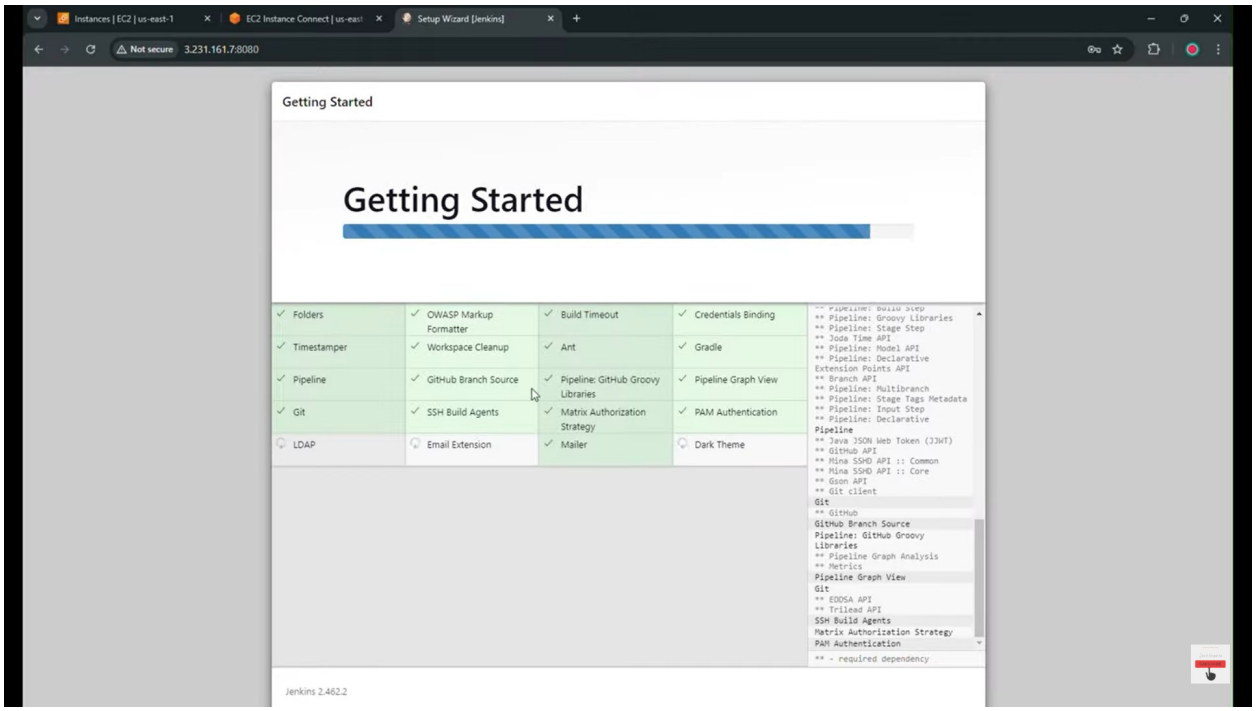
 # copy public ip of the instance to access jenking port :8080

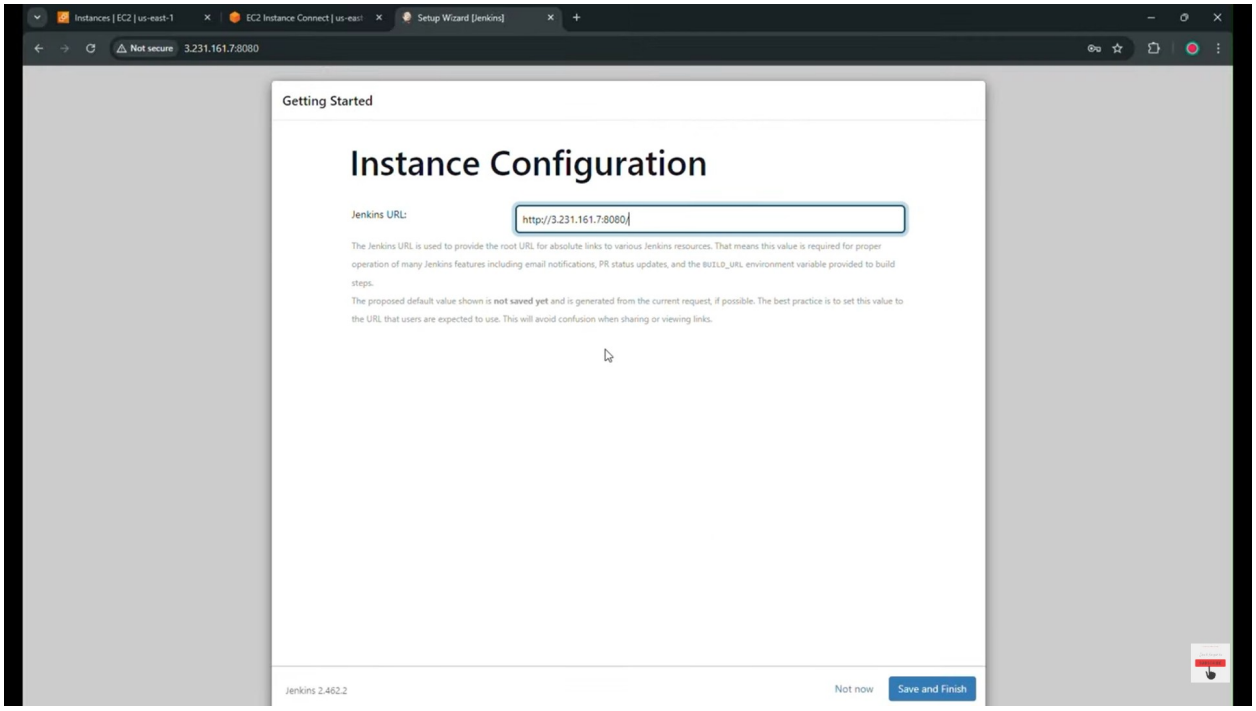


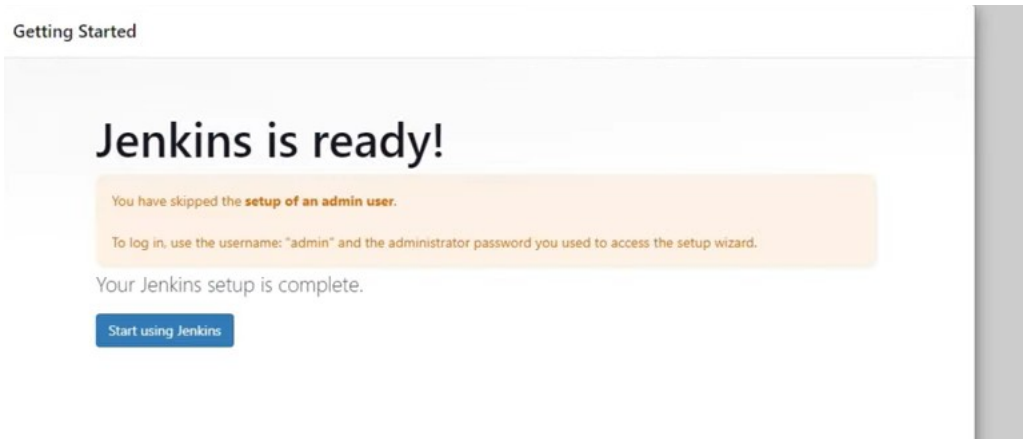

```
root@ip-10-0-9-146:~# jenkins --version
2.462.2
root@ip-10-0-9-146:~# cat /var/lib/jenkins/secrets/initialAdminPassword

f1255038fda046bb97920cd9ae3b378e
root@ip-10-0-9-146:~#
```


Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
Sign in [Jenkins]
Not secure
3.231.161.7:8080/login?from=%2F
To exit full screen, press Esc
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

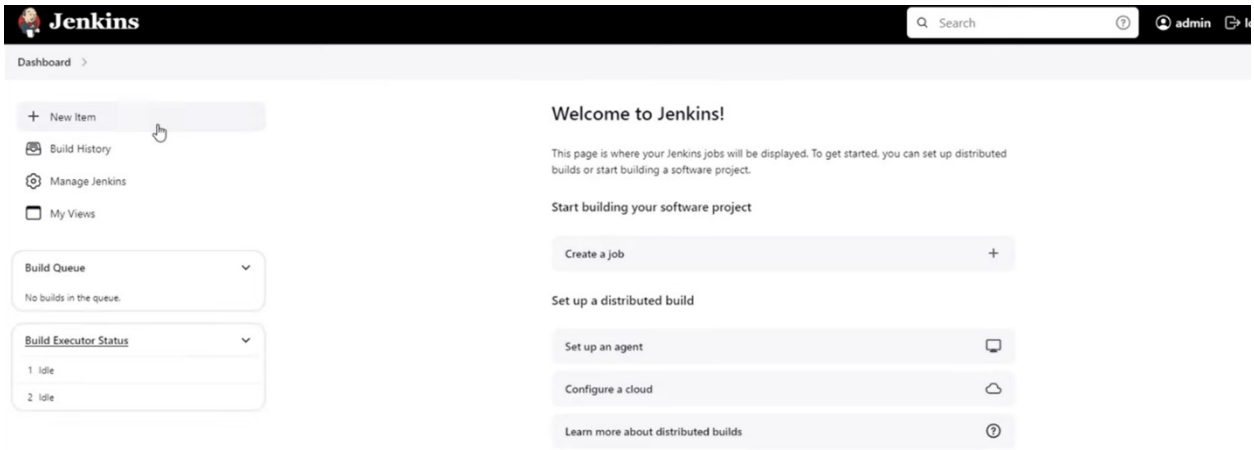

Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
Setup Wizard [Jenkins]
Not secure
3.231.161.7:8080
Getting Started
Customize Jenkins
Plugins extend Jenkins with additional features to support many different needs.
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.
Jenkins 2.462.2


Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
Setup Wizard [Jenkins]
Not secure 3.231.161.7:8080
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 Graph View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
Dark Theme
Jenkins 2.462.2

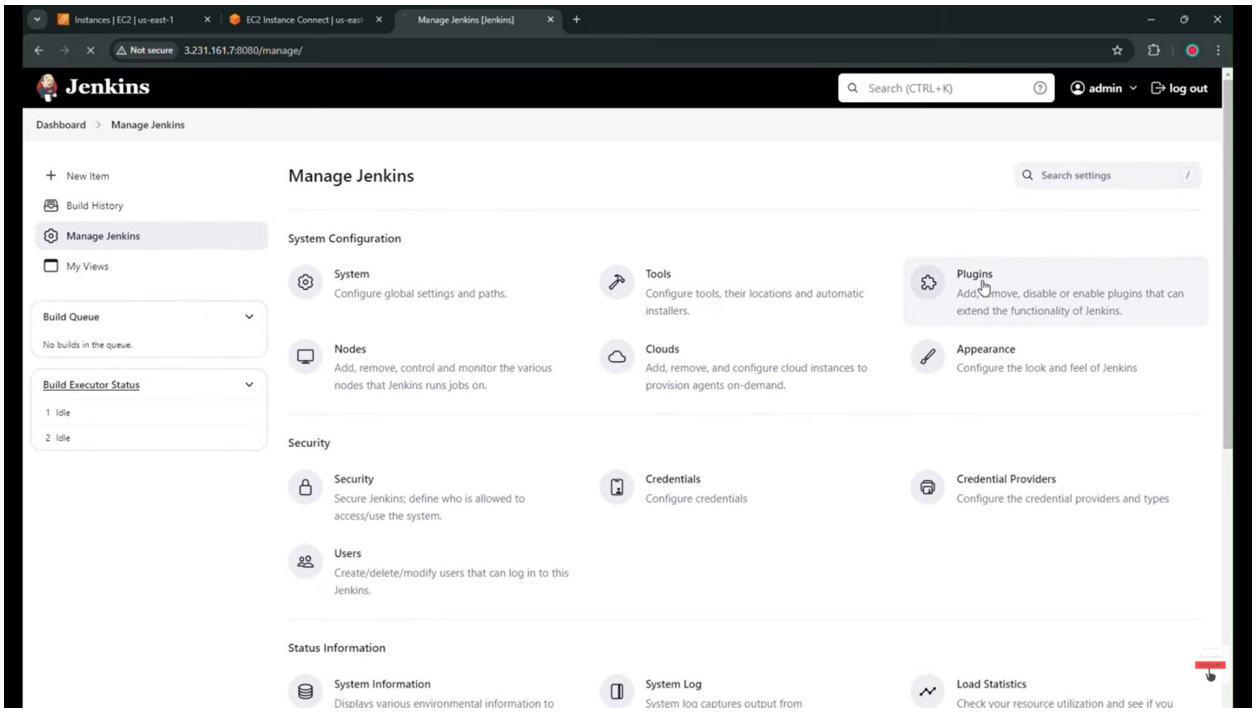

Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
Setup Wizard [Jenkins]
Not secure 3.231.161.7:8080
Getting Started
Instance Configuration
Jenkins URL:
http://3.231.161.7:8080/
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.
The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.
Jenkins 2.462.2
Not now
Save and Finish



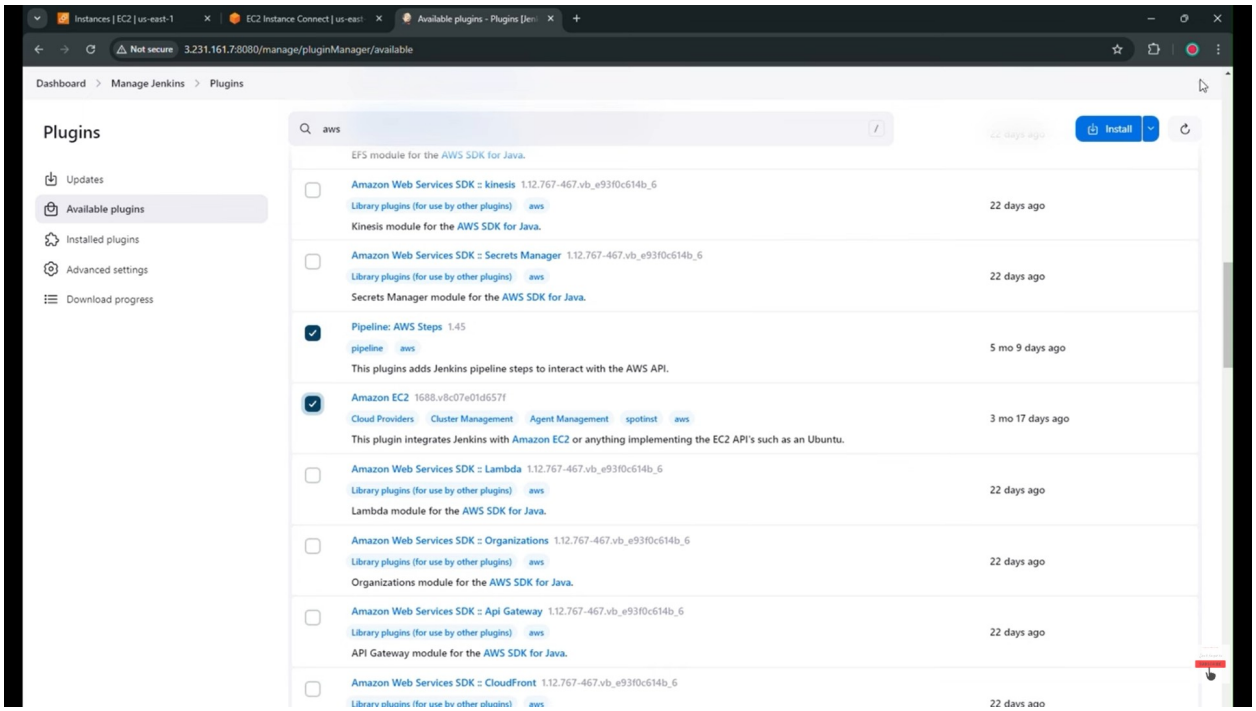
 # manage Jenkins

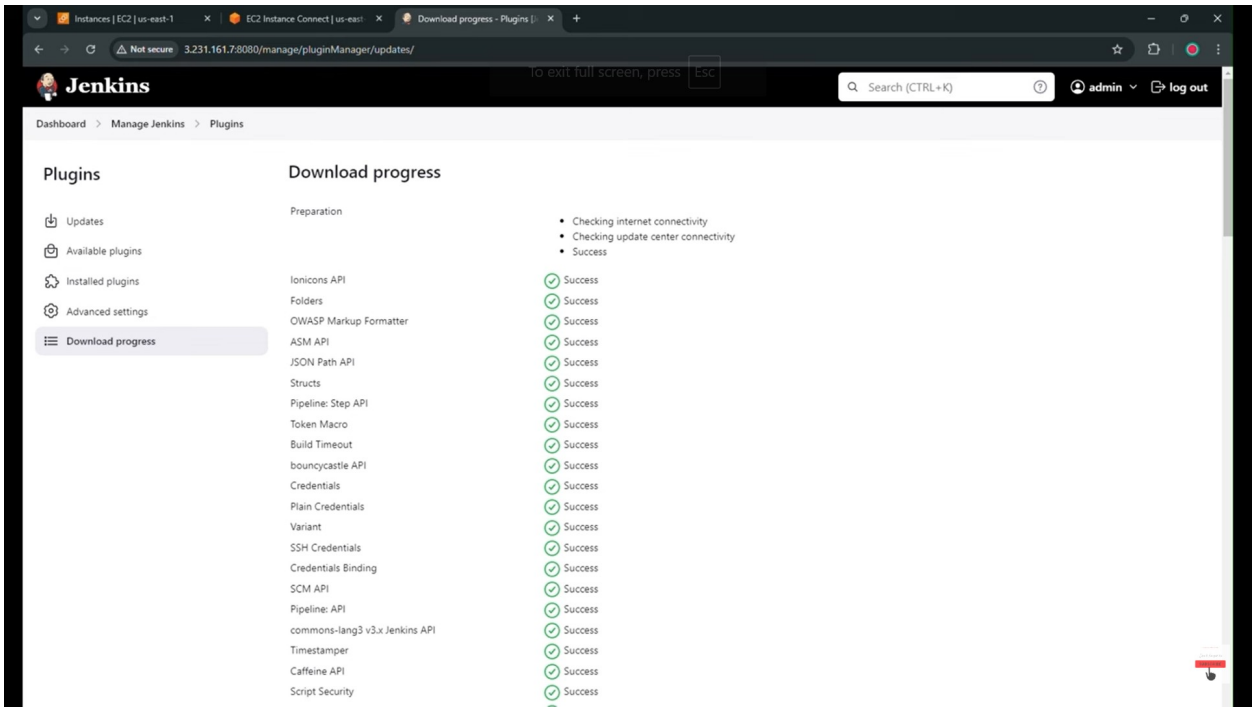


#



select pluginsa – AWS






```
#Once Cluster is up
export KUBECONFIG=/root/.kube/config
```

+

```
#Smoketest
ku get pods -A
ku get ns
ku get all -n kube-system
```

```
root@ip-10-0-9-146:~# export KUBECONFIG=/root/.kube/config
root@ip-10-0-9-146:~# kubectl get pods -A
NAMESPACE     NAME                       READY   STATUS    RESTARTS   AGE
kube-system   coredns-54d6f577c6-5mnhj   0/1     Pending   0          7m5s
kube-system   coredns-54d6f577c6-rdftn   0/1     Pending   0          7m5s
root@ip-10-0-9-146:~# alias ku=kubectl
root@ip-10-0-9-146:~# ku cluster-info
Kubernetes control plane is running at https://0F0680CC3DD41D8681A1D1A8FA19E3C8.gr7.us-east-1.eks.amazonaws.com
CoreDNS is running at https://0F0680CC3DD41D8681A1D1A8FA19E3C8.gr7.us-east-1.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
root@ip-10-0-9-146:~# ku get ns
NAME              STATUS   AGE
default           Active   9m39s
kube-node-lease   Active   9m39s
kube-public       Active   9m39s
kube-system       Active   9m39s
root@ip-10-0-9-146:~#
```

```
# Now we need to install nodePools before that we need to do OIDC authentication.

eksctl utils associate-iam-oidc-provider \
  --region us-east-1 \
  --cluster eks-cluster-1 \
  --approve
```

 # nodepools means---- worker node only .

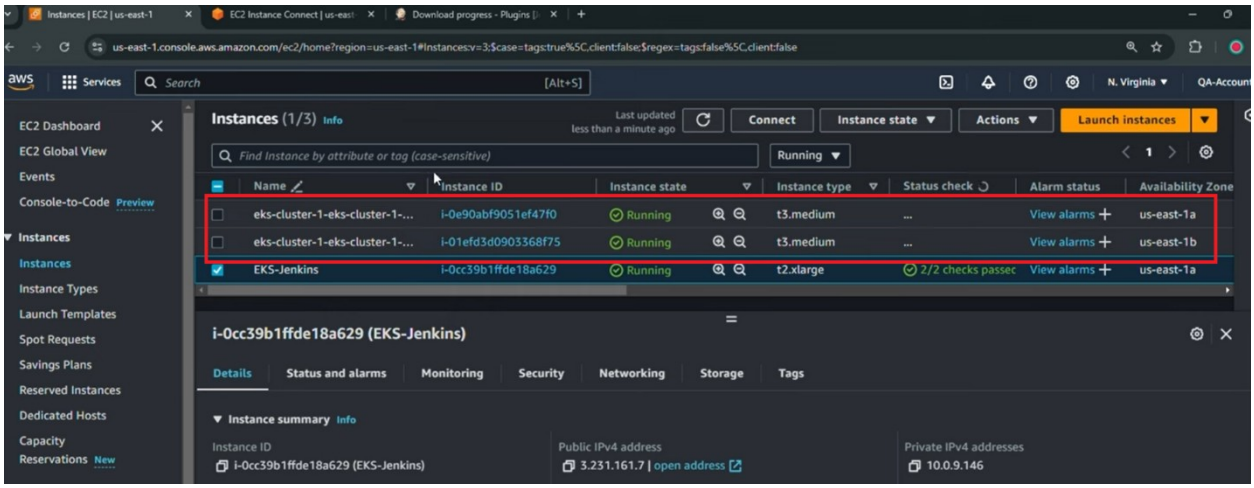
```
root@ip-10-0-9-146:~# eksctl utils associate-iam-oidc-provider \
        --region us-east-1 \
        --cluster eks-cluster-1 \
        --approve
2024-09-04 09:46:06 [ℹ]  will create IAM Open ID Connect provider for cluster "eks-cluster-1" in "us-east-1"
2024-09-04 09:46:06 [✓]  created IAM Open ID Connect provider for cluster "eks-cluster-1" in "us-east-1"
root@ip-10-0-9-146:~#
```

 # create nodegroup, If production go with the private . We using local so lets o with the Public only.

```
#For Node Group In Public Subnet

eksctl create nodegroup --cluster=eks-cluster-1 \
--region=us-east-1 \
--name=eks-cluster-1-cluster-ng-1 \
--node-type=t3.medium \
--nodes=2 \
--nodes-min=2 \
--nodes-max=4 \
--node-volume-size=20 \
--ssh-access \
--ssh-public-key=QA-Account \
--managed \
--asg-access \
--external-dns-access \
--full-ecr-access \
--appmesh-access \
--alb-ingress-access
```

```
2024-09-04 09:46:06 [✓]  created IAM Open ID Connect provider for cluster "eks-cluster-1" in "us-east-1"
root@ip-10-0-9-146:~# eksctl create nodegroup --cluster=eks-cluster-1 \
--region=us-east-1 \
--name=eks-cluster-1-cluster-ng-1 \
--node-type=t3.medium \
--nodes=2 \
--nodes-min=2 \
--nodes-max=4 \
--node-volume-size=20 \
--ssh-access \
--ssh-public-key=QA-Account \
--managed \
--asg-access \
--external-dns-access \
--full-ecr-access \
--appmesh-access \
--alb-ingress-access
2024-09-04 09:46:52 [ℹ]  will use version 1.29 for new nodegroup(s) based on control plane version
2024-09-04 09:46:53 [ℹ]  nodegroup "eks-cluster-1-cluster-ng-1" will use "" [AmazonLinux2/1.29]
2024-09-04 09:46:53 [ℹ]  using EC2 key pair "QA-Account"
2024-09-04 09:46:53 [ℹ]  1 nodegroup (eks-cluster-1-cluster-ng-1) was included (based on the include/exclude rules)
2024-09-04 09:46:53 [ℹ]  will create a CloudFormation stack for each of 1 managed nodegroups in cluster "eks-cluster-1"
2024-09-04 09:46:53 [ℹ]
2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "eks-cluster-1-cluster-ng
```

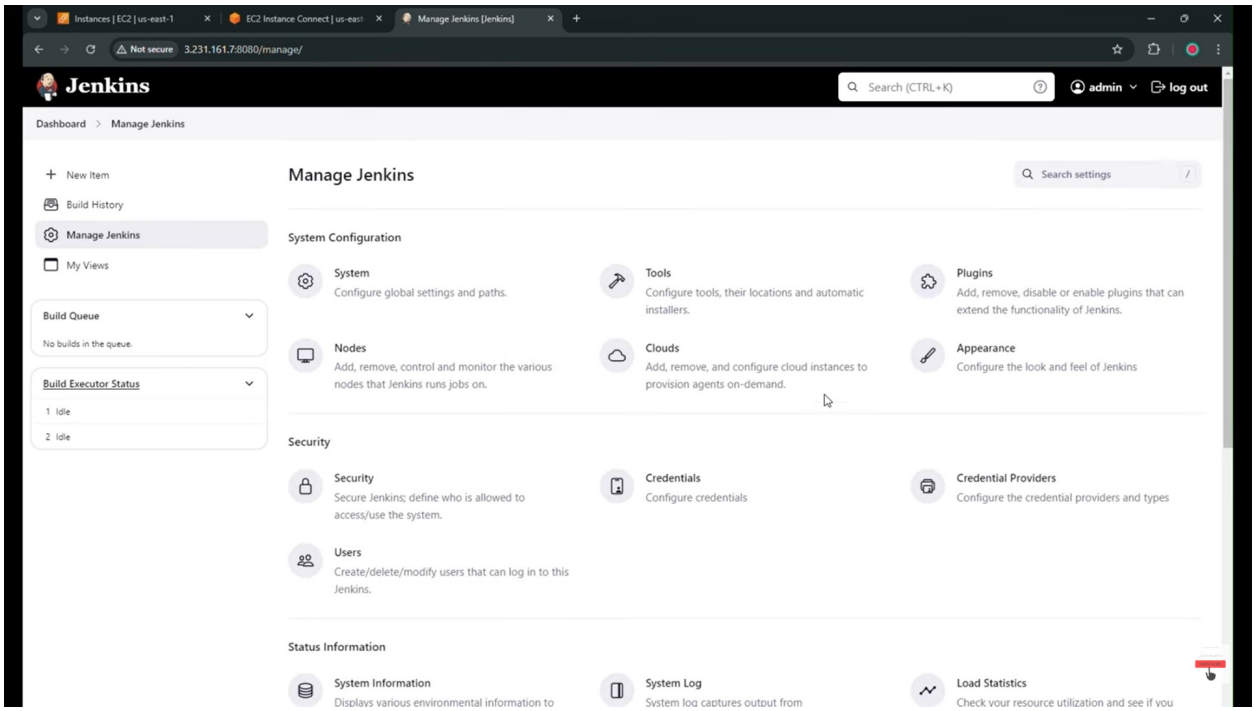
 # 2 instances created



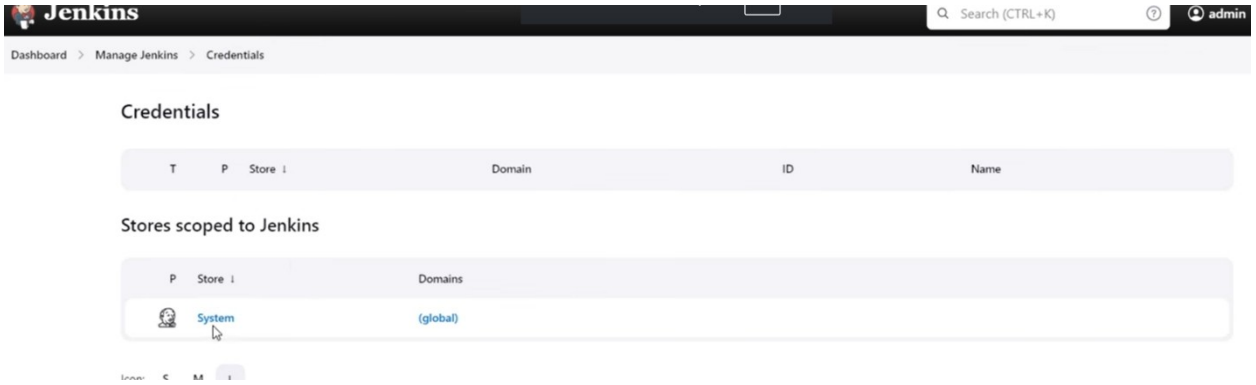
 # handshake B/w EKS + jenkins

```
# NOW THE TASK IS OUR JENKINS SERVER SHOULD BE INTEGARTED WITH THE EKS
# WE WLL DO IT IN A CREDENTIAL PLUGIN WAY

# jenkins > credentials  > system > global > Awscred > Global > aws-access-creds - copy paste the keys
```

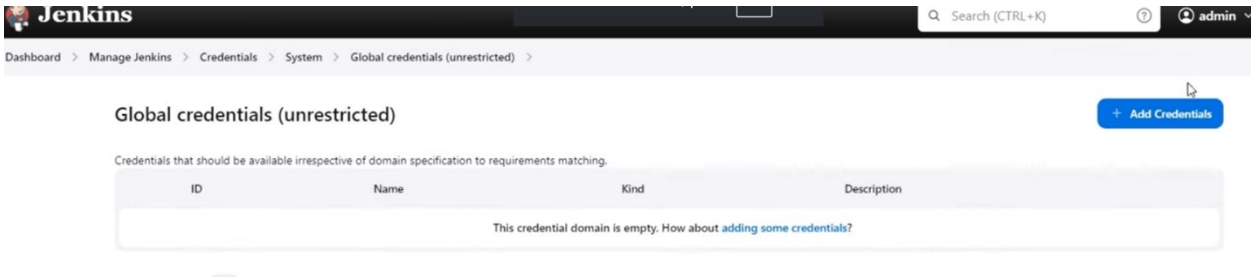
 # manage Jenkins credentials

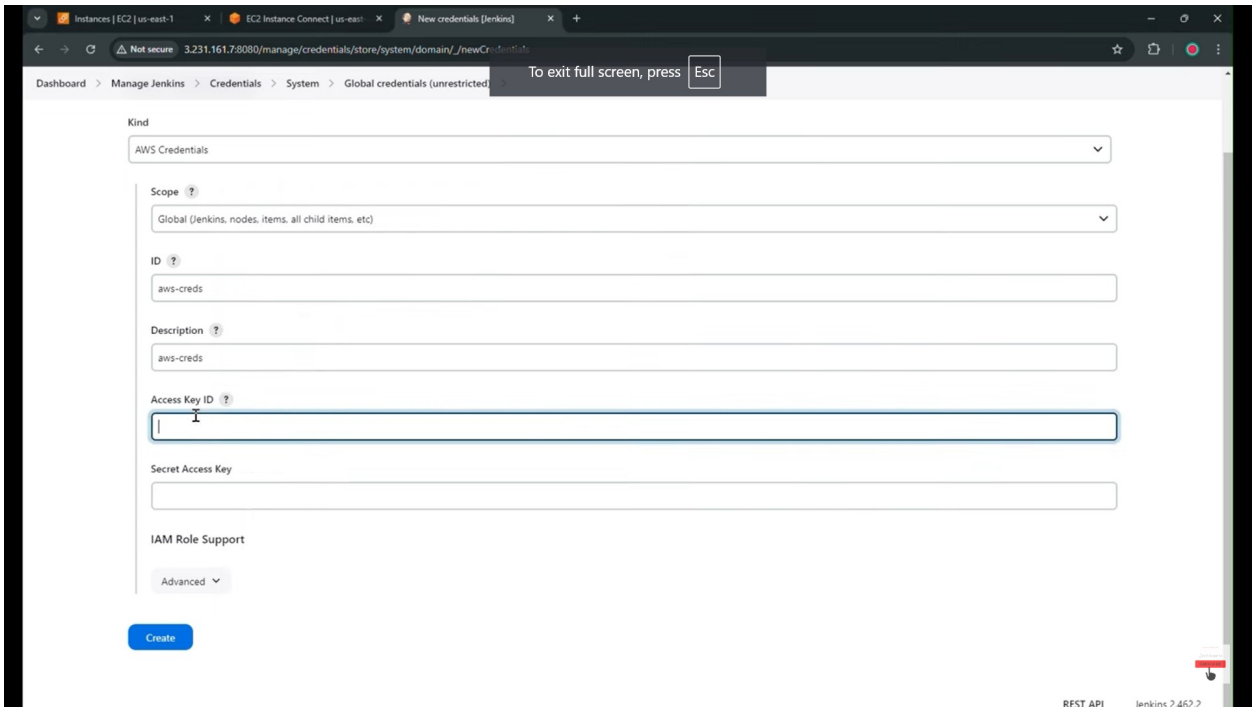


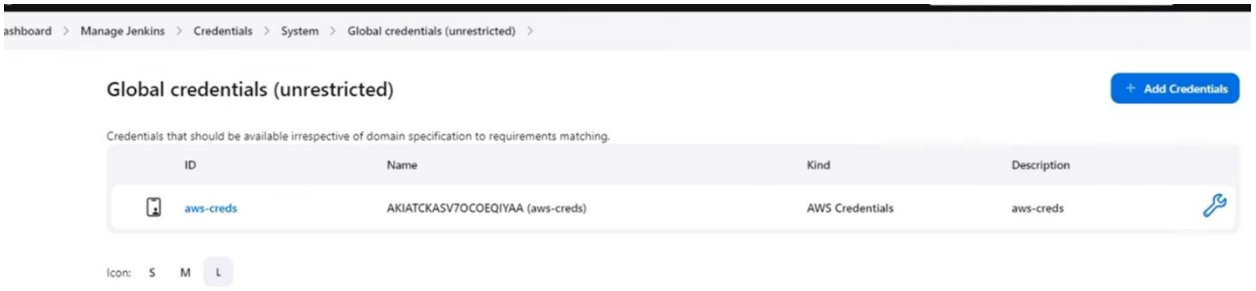
 # Credentials – sytem



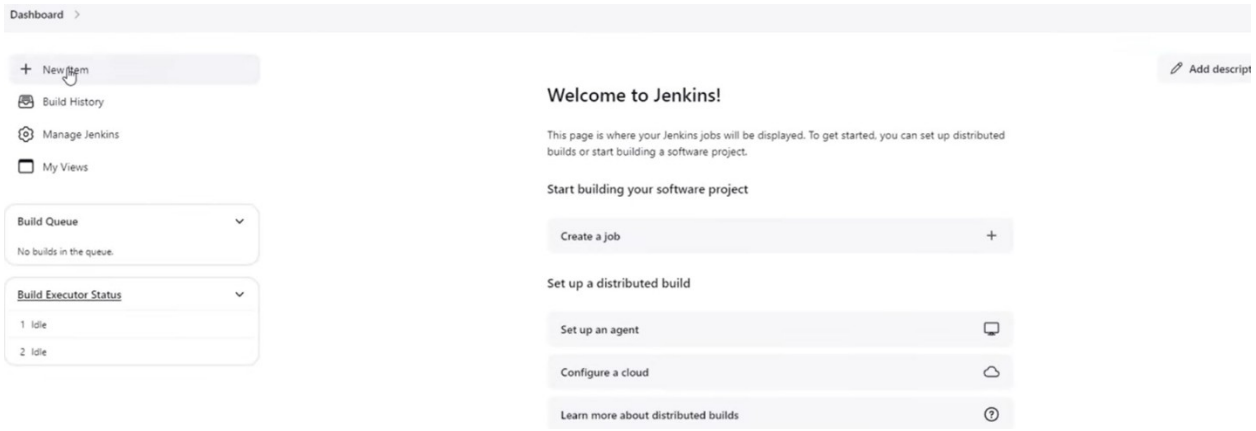
 # Global credentials + add Credentials

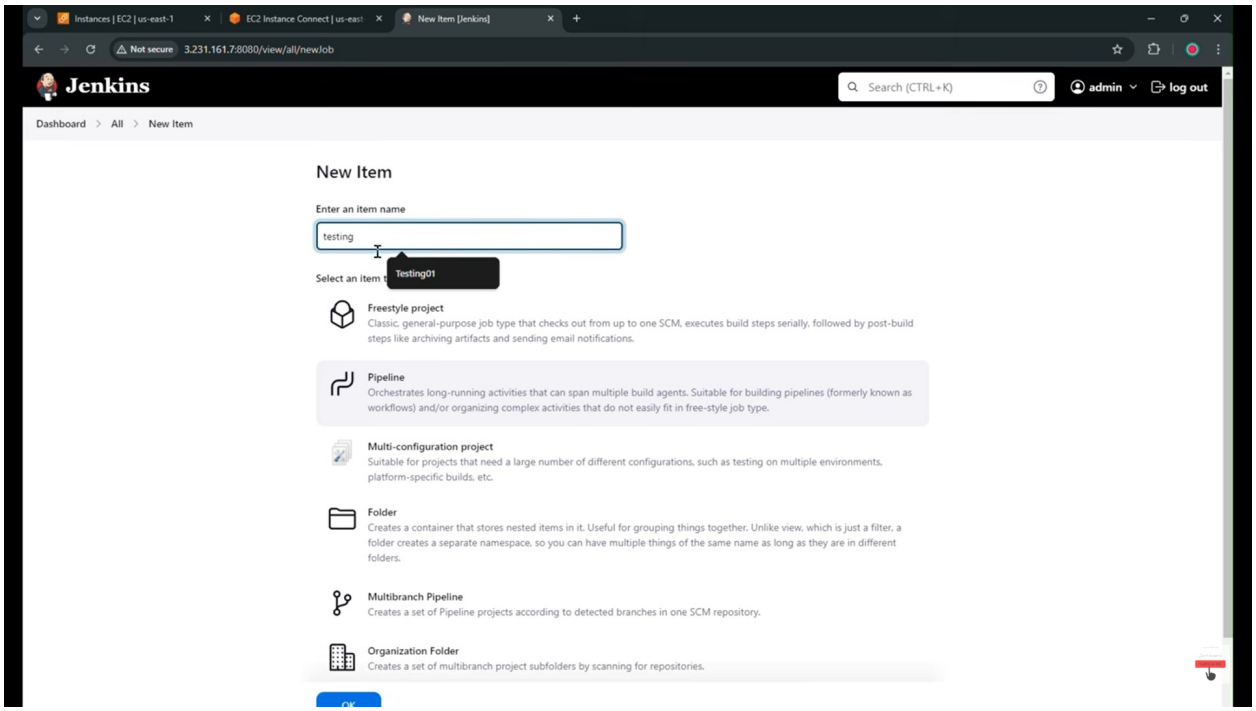




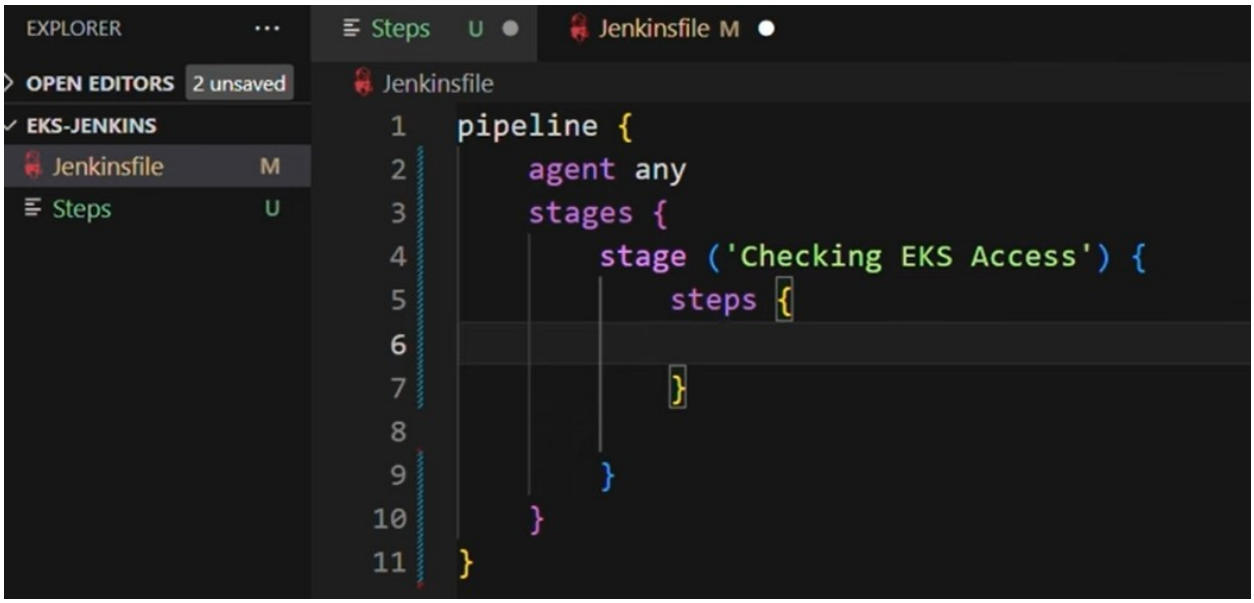


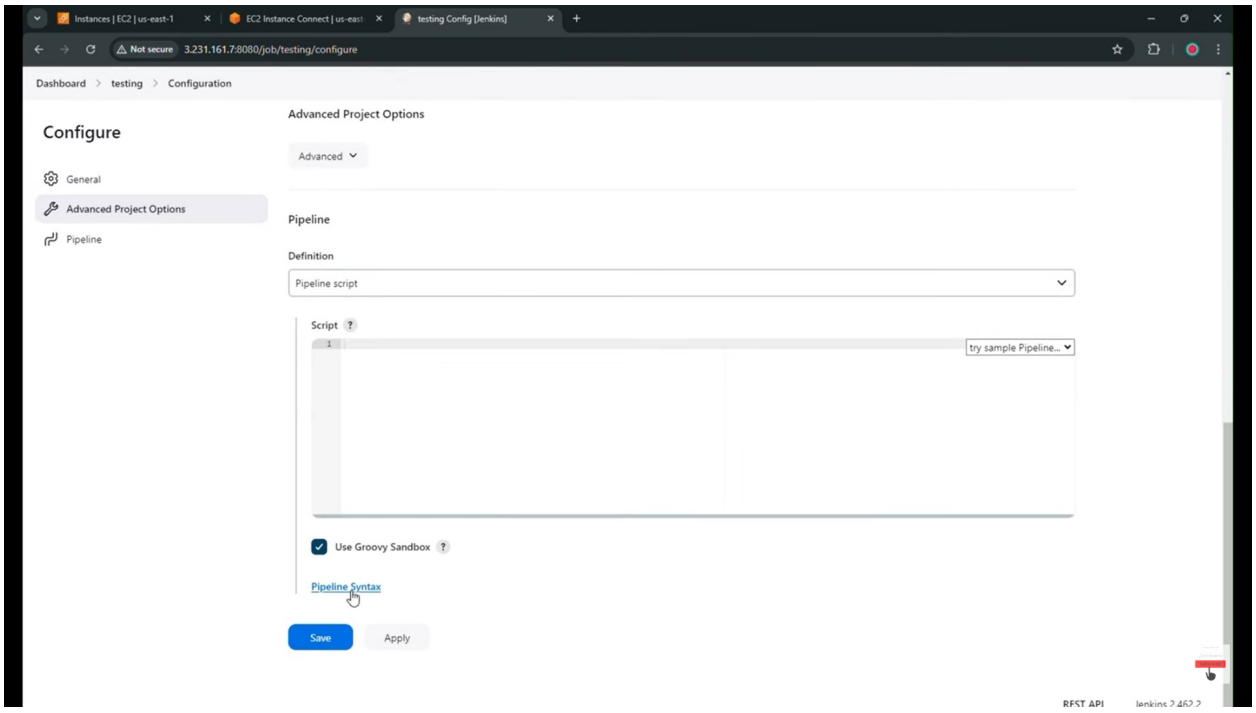
Create a Jenkins Pipeline

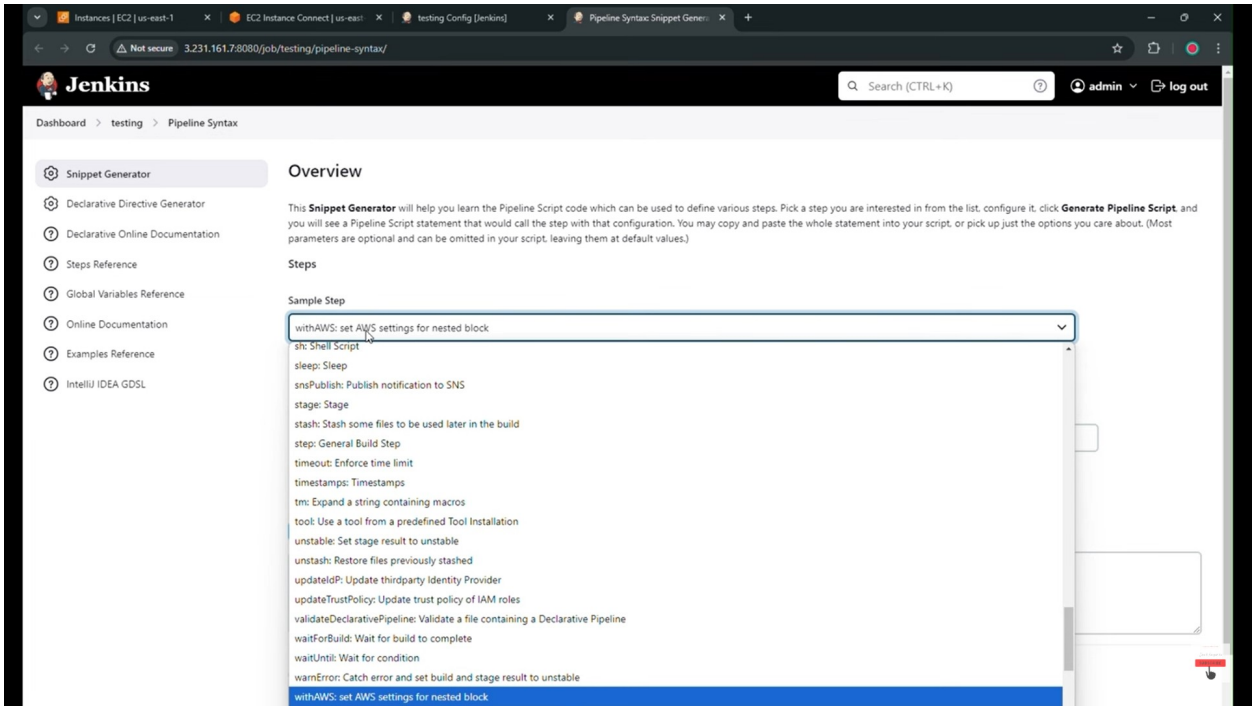


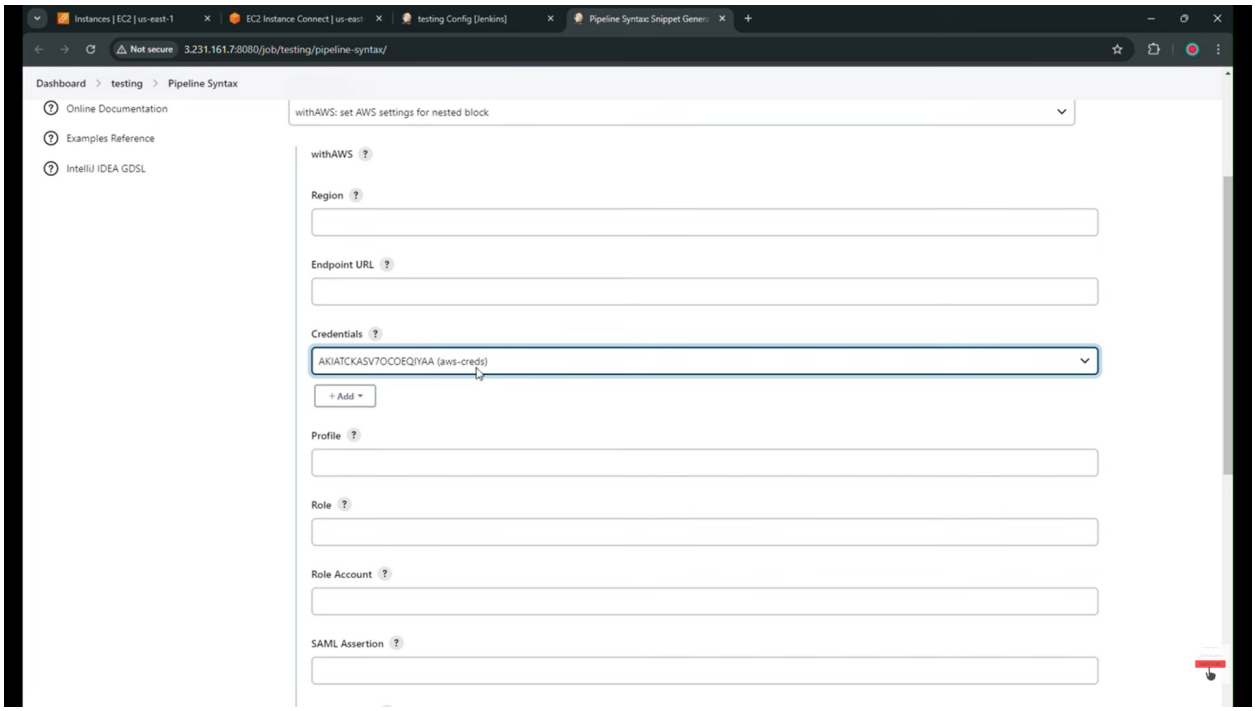


 # click pipeline syntax




Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
testing Config [Jenkins]
Not secure
3.231.161.7:8080/job/testing/configure
Dashboard > testing > Configuration
Configure
General
Advanced Project Options
Pipeline
Advanced Project Options
Advanced
Pipeline
Definition
Pipeline script
Script
try sample Pipeline...
Use Groovy Sandbox
Pipeline Syntax
Save
Apply
REST API


Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
testing Config [Jenkins]
Pipeline Syntax Snippet Gener
Not secure
3.231.161.7:8080/job/testing/pipeline-syntax/
Jenkins
Search (CTRL+K)
admin
log out
Dashboard > testing > Pipeline Syntax
Snippet Generator
Declarative Directive Generator
Declarative Online Documentation
Steps Reference
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
Overview
This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)
Steps
Sample Step
withAWS: set AWS settings for nested block
sh: Shell Script
sleep: Sleep
snsPublish: Publish notification to SNS
stage: Stage
stash: Stash some files to be used later in the build
step: General Build Step
timeout: Enforce time limit
timestamps: Timestamps
tm: Expand a string containing macros
tool: Use a tool from a predefined Tool Installation
unstable: Set stage result to unstable
unstash: Restore files previously stashed
updateIdP: Update thirdparty Identity Provider
updateTrustPolicy: Update trust policy of IAM roles
validateDeclarativePipeline: Validate a file containing a Declarative Pipeline
waitForBuild: Wait for build to complete
waitUntil: Wait for condition
warnError: Catch error and set build and stage result to unstable
withAWS: set AWS settings for nested block

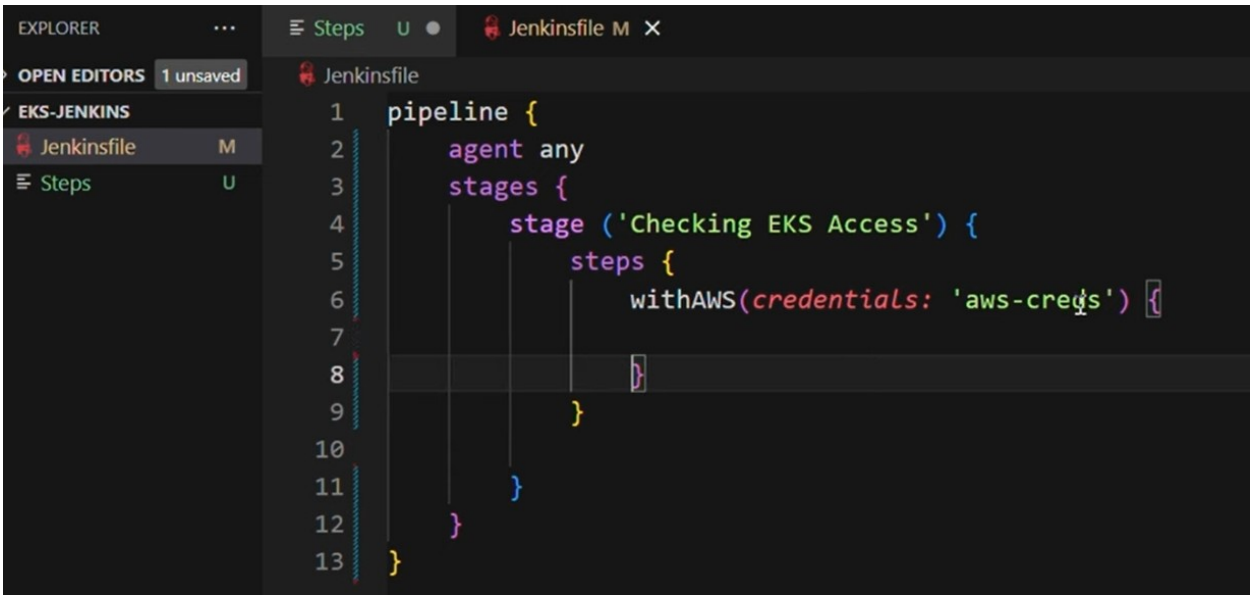


Federated User ID

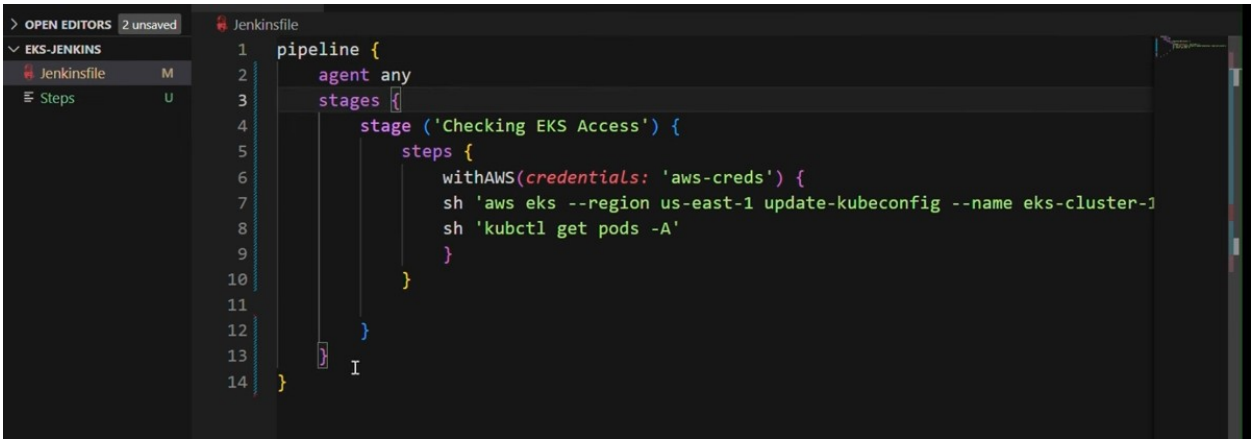
Generate Pipeline Script

```
withAWS(credentials: 'aws-creds') {
    // some block
}
```

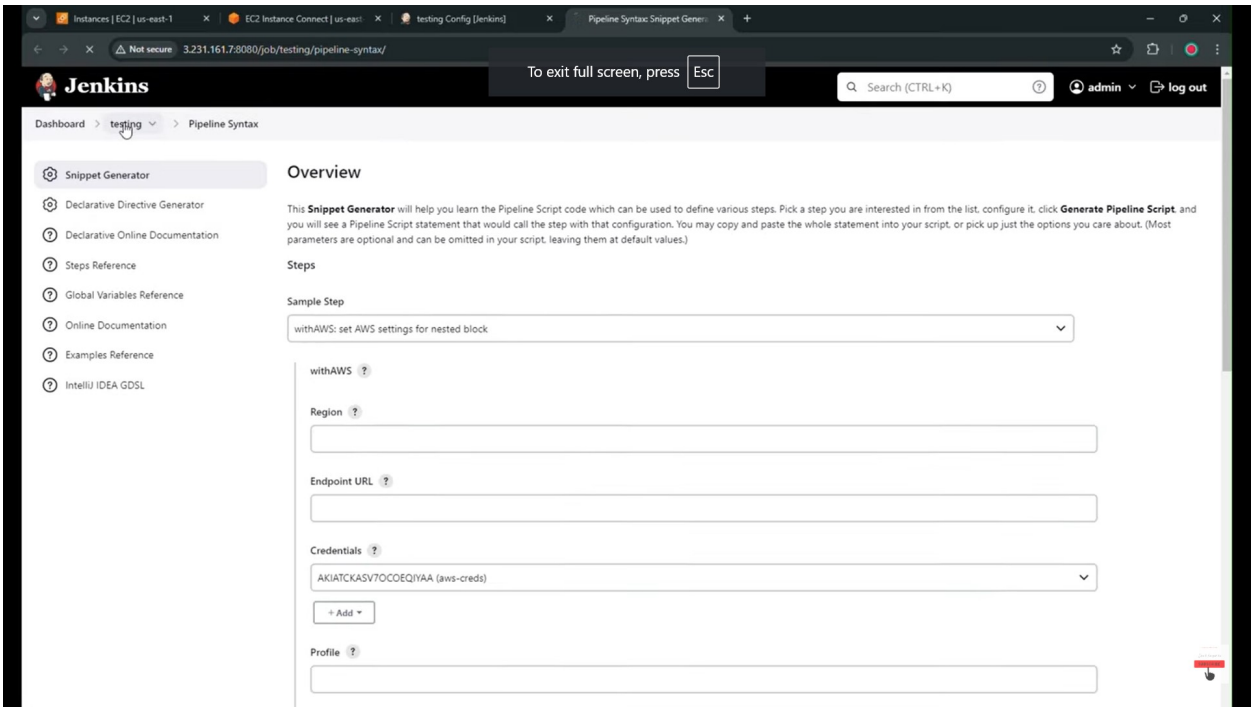
Global Variables



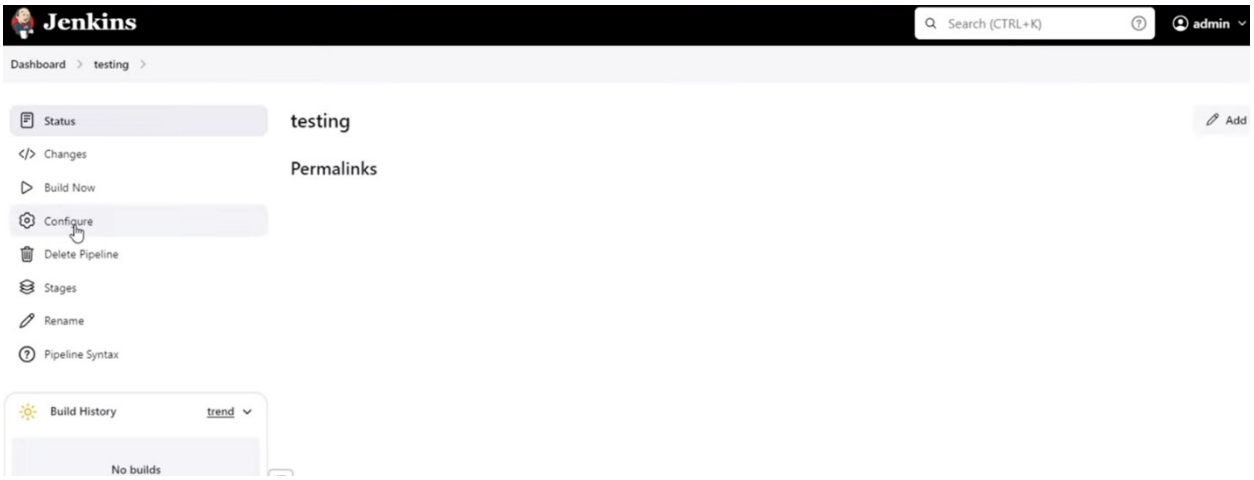

```
# sh 'aws eks --region us-east-1 update-kubeconfig --name eks-cluster-1'
# sh 'kubectl get pods -A'
```

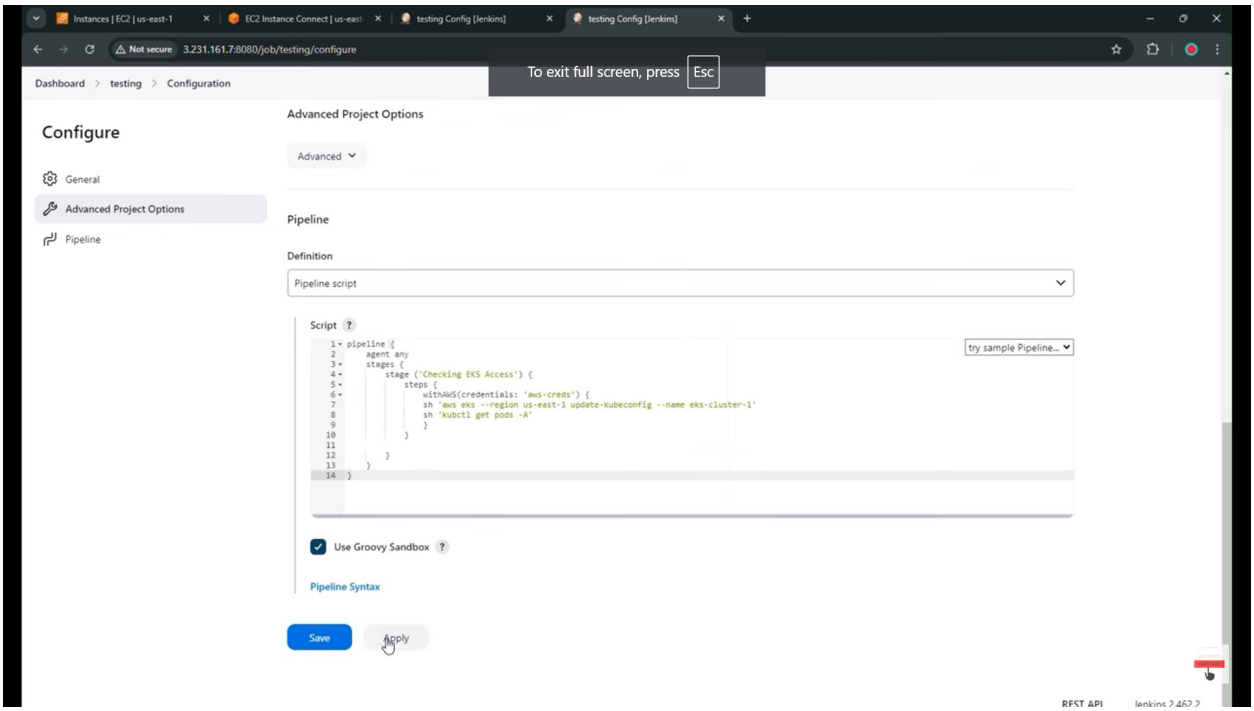


now go back to Jenkins pipeline .. & check configuration status + proper integration with cluster ??



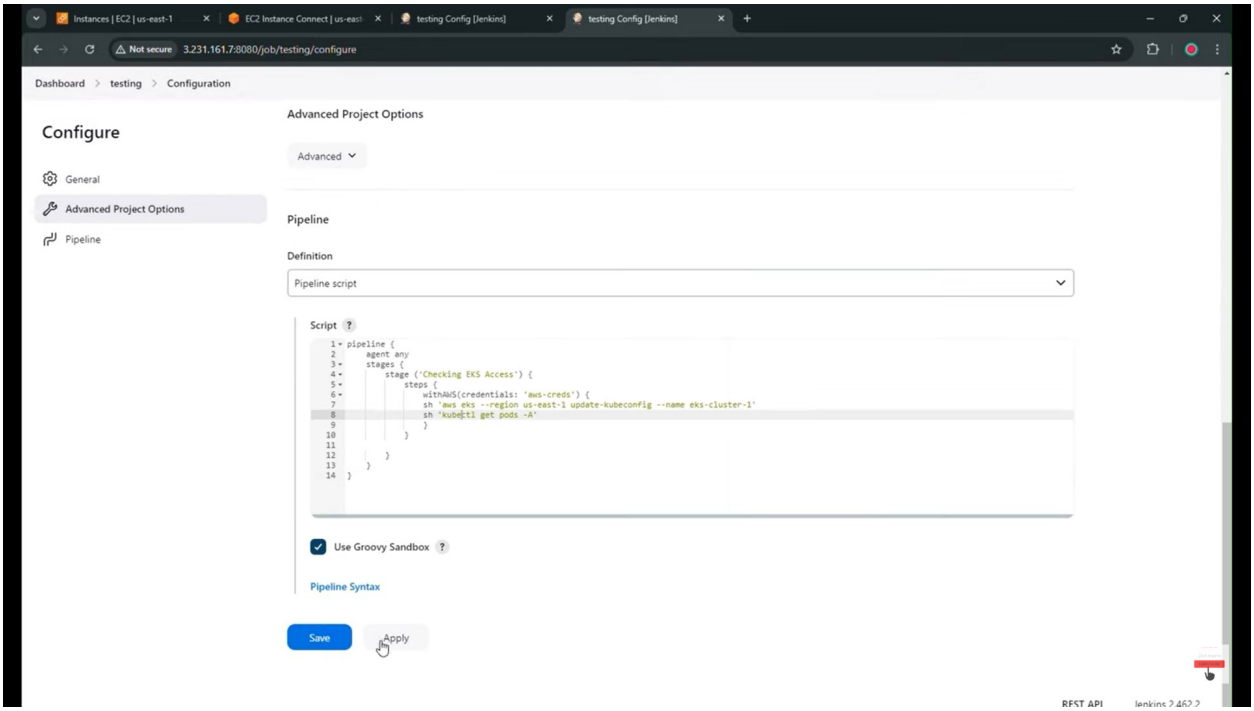
click testing - click configure

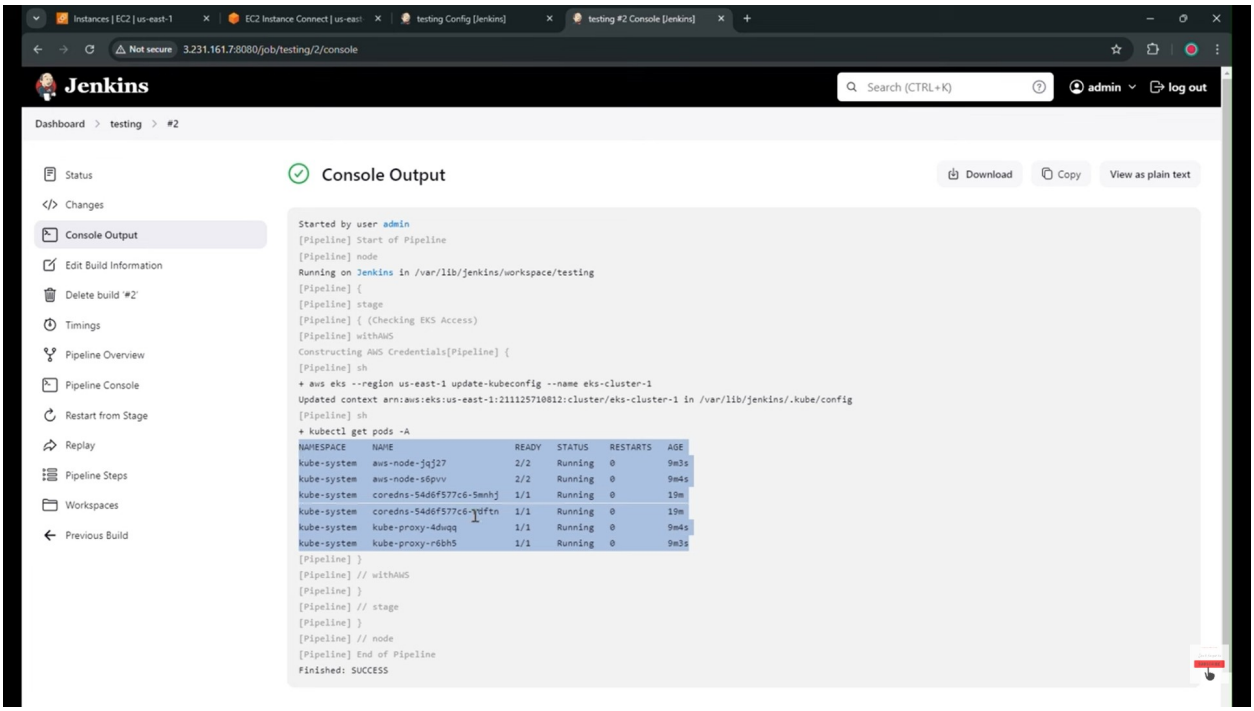




error . rename .. kubectl

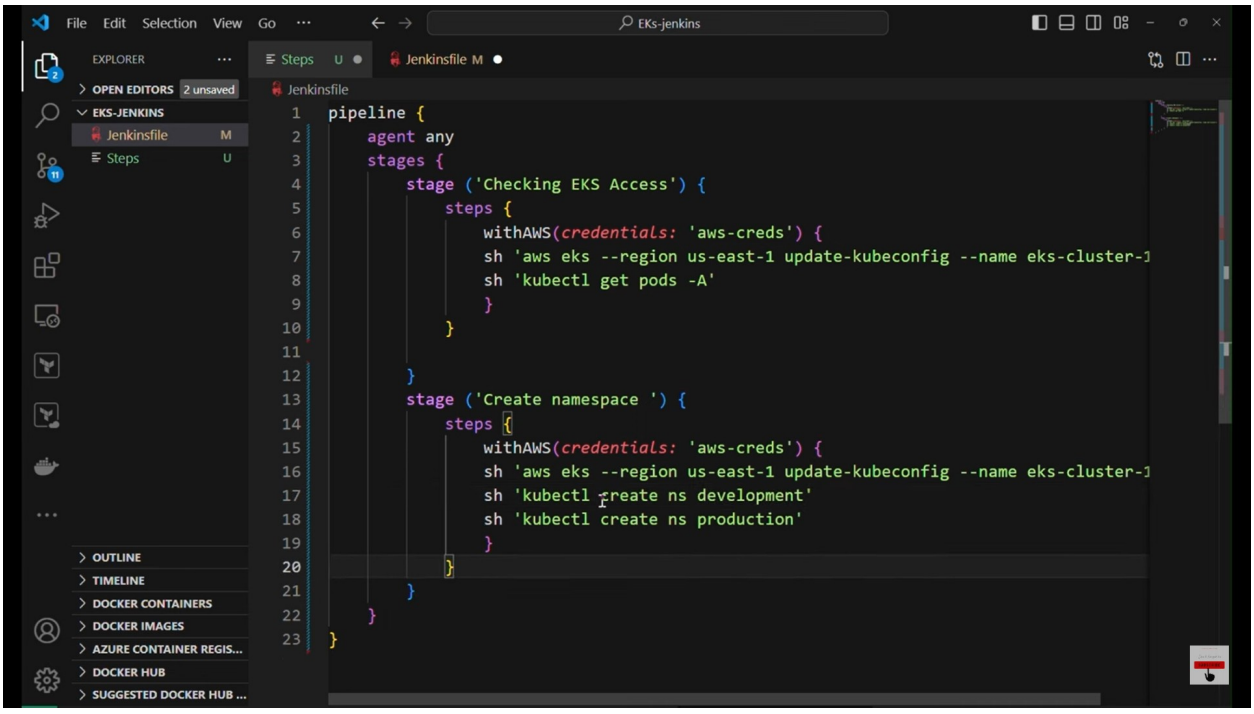
--- testing- configure --- Buildnow



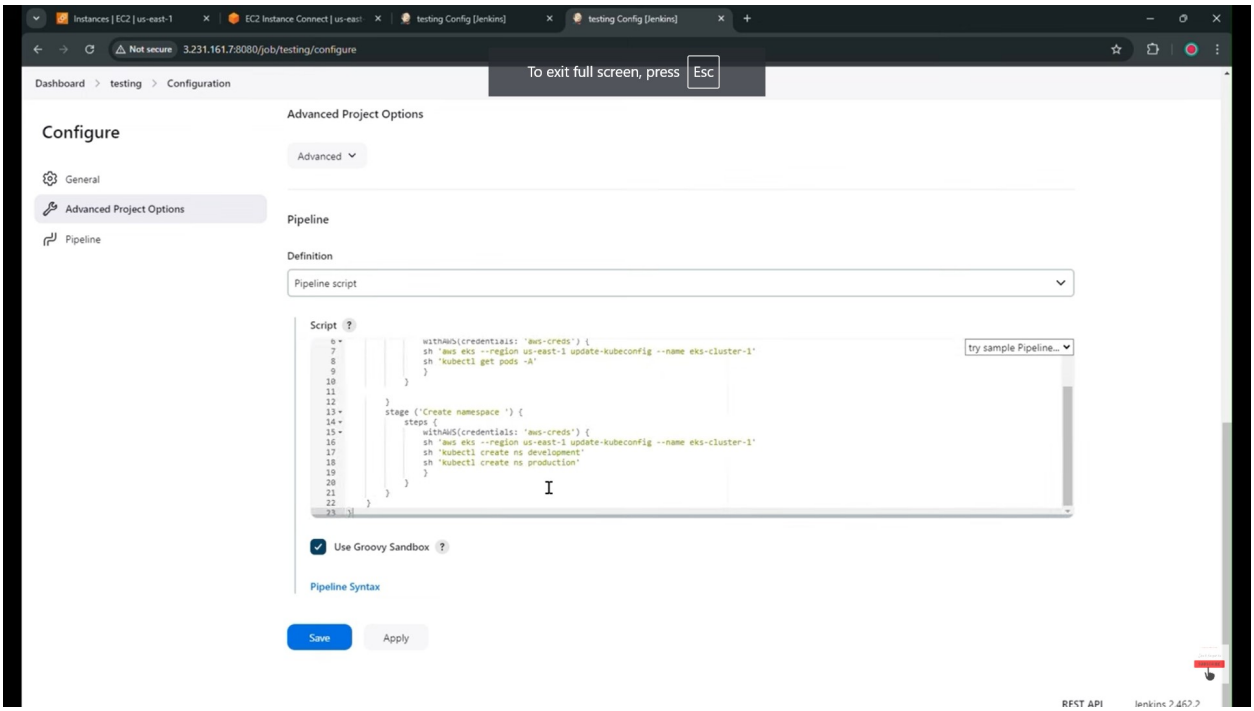


now Jenkins able to access eks cluster + aws credentials as I have allotted admin access

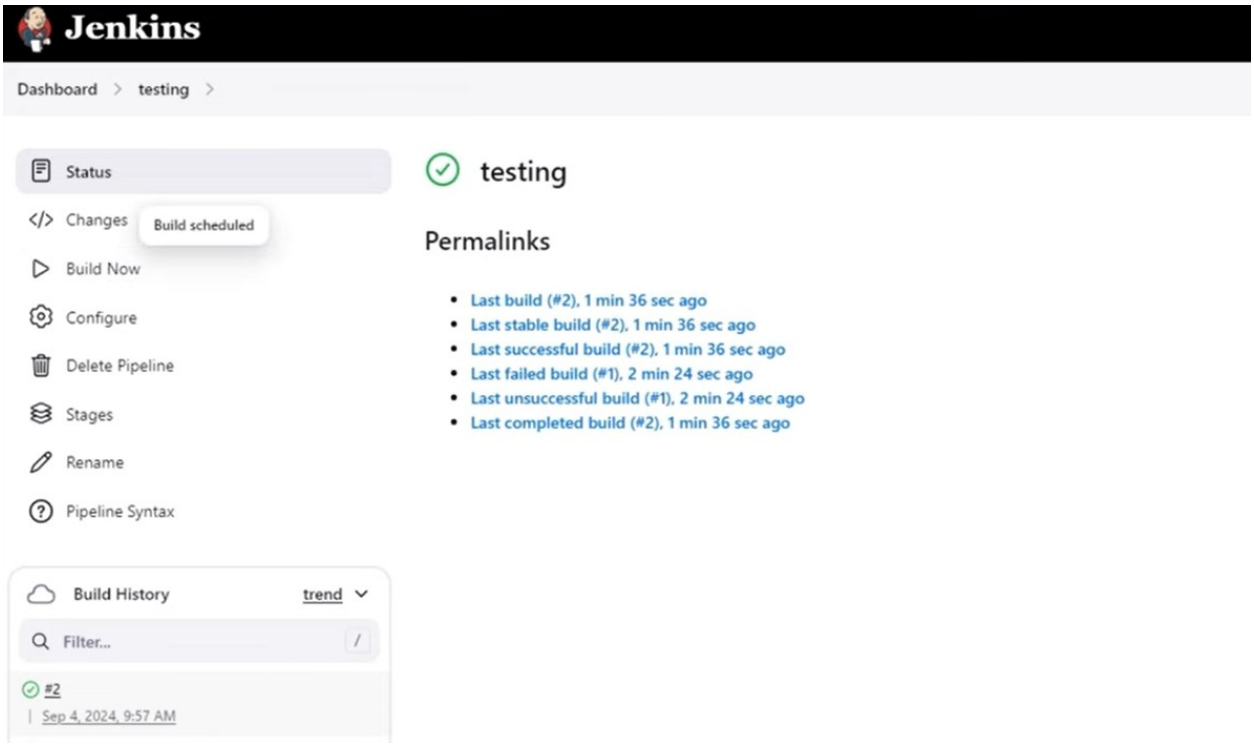
2nd .. create namespace -ns



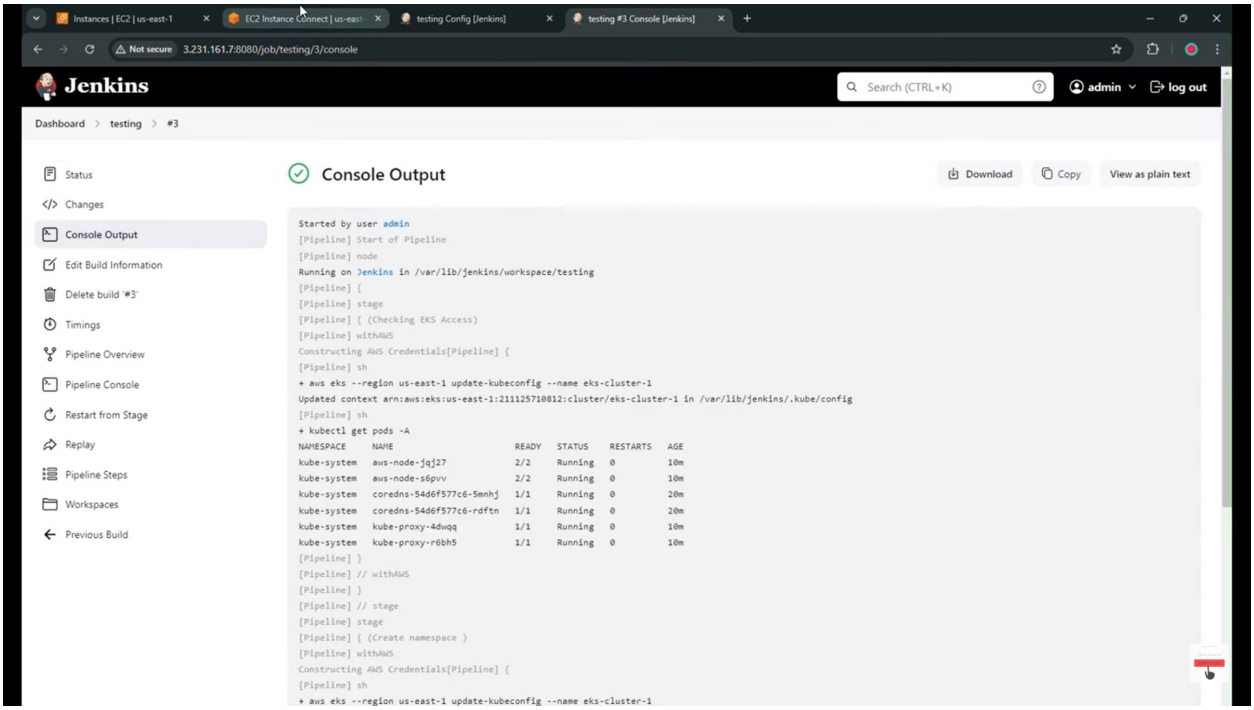
go back to Jenkins- testing- configure- Groovy script


Not secure 3.231.161.7:8080/job/testing/configure
To exit full screen, press Esc
Dashboard > testing > Configuration
Configure
General
Advanced Project Options
Pipeline
Advanced Project Options
Advanced
Pipeline
Definition
Pipeline script
Script
try sample Pipeline...
Use Groovy Sandbox
Pipeline Syntax
Save
Apply

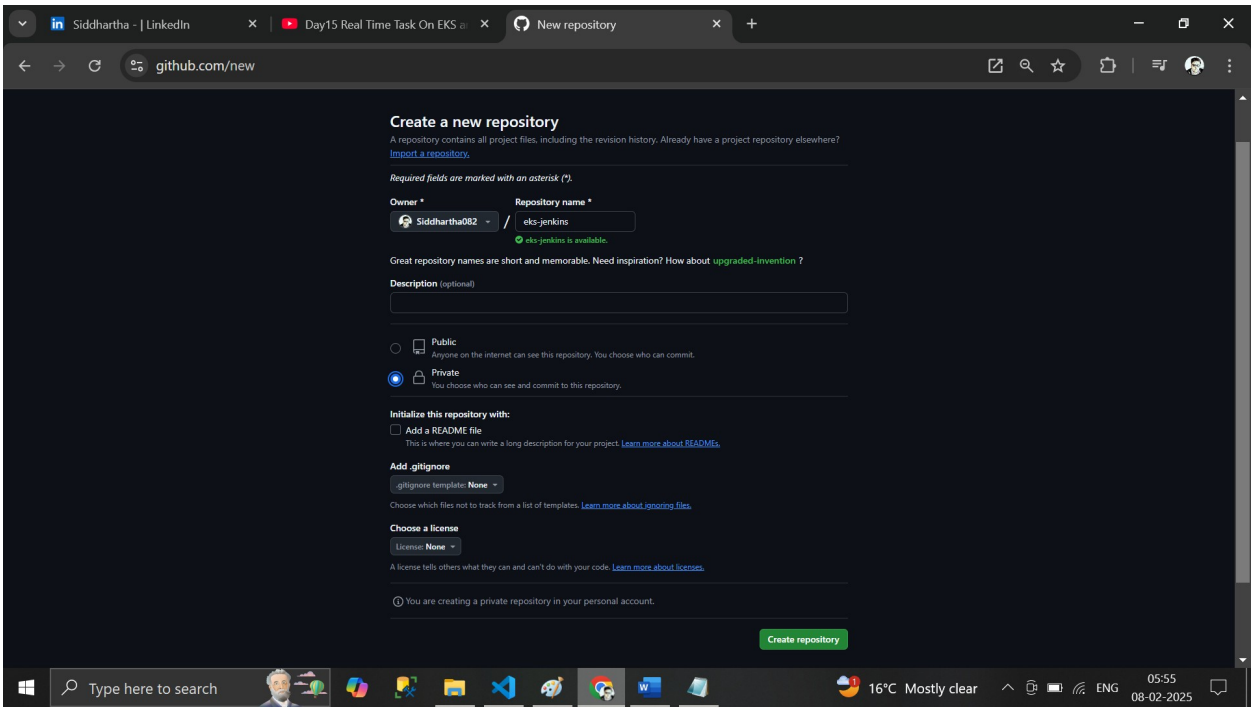
Buildnow –

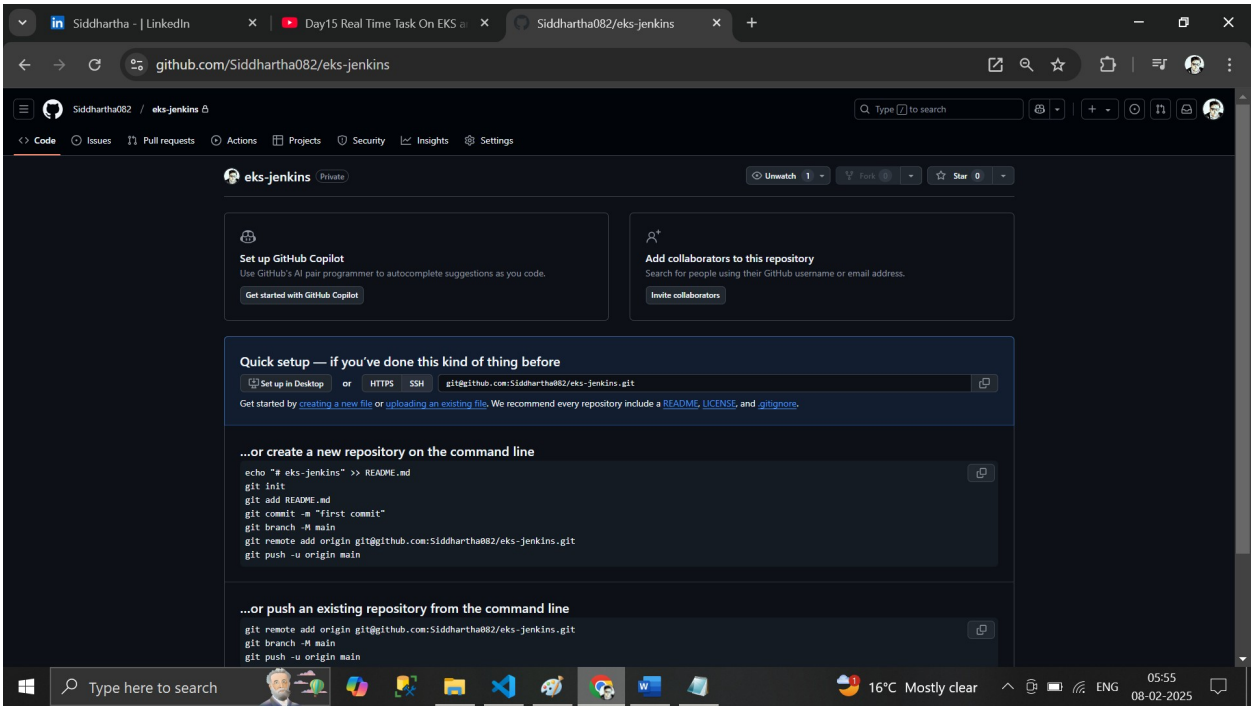

Jenkins
Dashboard > testing >
Status
Changes
Build scheduled
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax
Build History
trend
Filter...
#2
Sep 4, 2024, 9:57 AM
testing
Permalinks
Last build (#2), 1 min 36 sec ago
Last stable build (#2), 1 min 36 sec ago
Last successful build (#2), 1 min 36 sec ago
Last failed build (#1), 2 min 24 sec ago
Last unsuccessful build (#1), 2 min 24 sec ago
Last completed build (#2), 1 min 36 sec ago

```
root@ip-10-0-9-146:~# ku get ns
NAME              STATUS   AGE
default           Active   23m
development       Active   0s
kube-node-lease   Active   23m
kube-public       Active   23m
kube-system       Active   23m
root@ip-10-0-9-146:~# ku get ns
NAME              STATUS   AGE
default           Active   23m
development       Active   22s
kube-node-lease   Active   23m
kube-public       Active   23m
kube-system       Active   23m
production        Active   21s
root@ip-10-0-9-146:~#
```




```
# LETS-CREATE A PRIVATE GITHUB REPOSITORY AND MAKE A HANDSHAKE METHOD BETWEEN GITHUB AND JENKINS
```


Siddhartha - | LinkedIn
Day15 Real Time Task On EKS
New repository
github.com/new
Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.
Required fields are marked with an asterisk (*).
Owner *
Repository name *
Siddhartha082
eks-jenkins
eks-jenkins is available.
Great repository names are short and memorable. Need inspiration? How about upgraded-invention ?
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Add a README file
This is where you can write a long description for your project. Learn more about READMEs.
Add .gitignore
.gitignore template: None
Choose which files not to track from a list of templates. Learn more about ignoring files.
Choose a license
License: None
A license tells others what they can and can't do with your code. Learn more about licenses.
You are creating a private repository in your personal account.
Create repository
Type here to search
16°C Mostly clear
ENG
05:55
08-02-2025


Siddhartha - | LinkedIn
Day15 Real Time Task On EKS
Siddhartha082/eks-jenkins
github.com/Siddhartha082/eks-jenkins
Siddhartha082 / eks-jenkins
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
eks-jenkins
Private
Unwatch 1
Fork 0
Star 0
Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot
Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
git@github.com:Siddhartha082/eks-jenkins.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# eks-jenkins" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:Siddhartha082/eks-jenkins.git
git push -u origin main
...or push an existing repository from the command line
git remote add origin git@github.com:Siddhartha082/eks-jenkins.git
git branch -M main
git push -u origin main
Type here to search
16°C Mostly clear
ENG
05:55
08-02-2025

#

```
root@ip-10-0-9-146:~# su - jenkins
jenkins@ip-10-0-9-146:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/jenkins/.ssh/id_rsa):
Created directory '/var/lib/jenkins/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ZAwcoxjnU56NypbK2bHTVuXGKISK0AQyqdz4IyzKQHg jenkins@ip-10-0-9-146
The key's randomart image is:
+---[RSA 3072]----+
|+oo ..=.         |
|o+ = =.B         |
|= = = = = .      |
|+=E+ = o =       |
|+.o * . S +      |
|o+ B + o .       |
|= = = o          |
|..   o           |
|                 |
+----[SHA256]-----+
jenkins@ip-10-0-9-146:~$
```

i-0cc39b1ffde18a629 (EKS-Jenkins)

PublicIPs: 3.231.161.7 PrivateIPs: 10.0.9.146

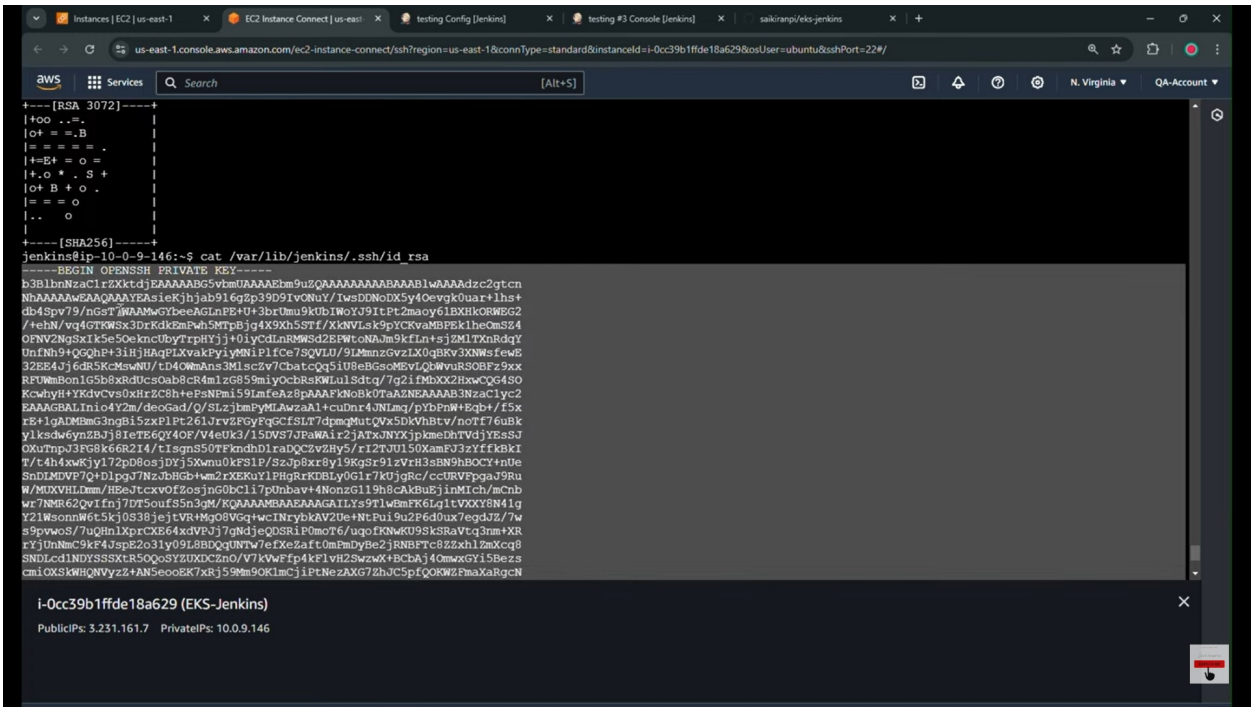
```
su - jenkins
ssh-keyhen #generating ket for authentication.
```

 # cp the Pvt Key .paste it in Jenkins credentials

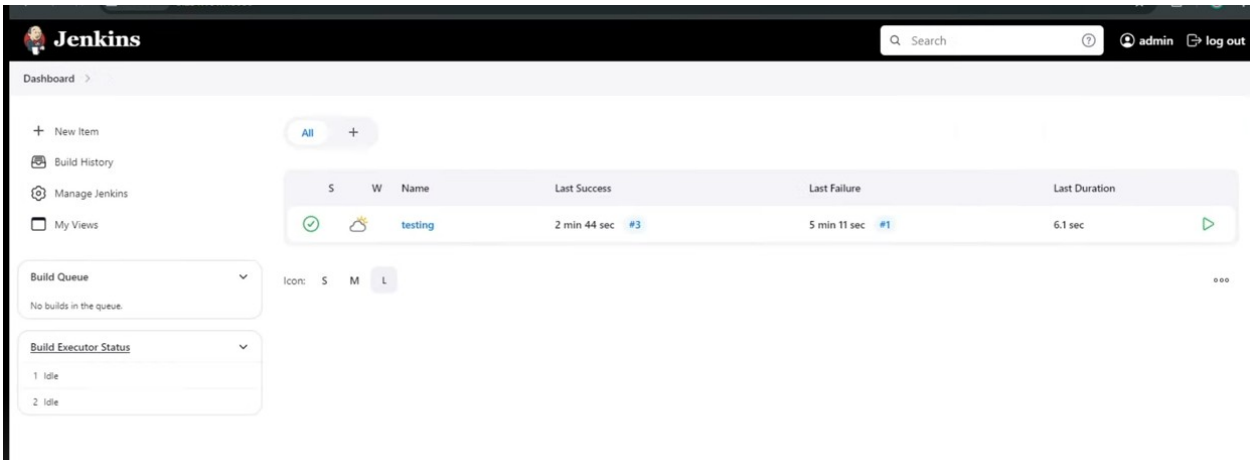
```
jenkins@ip-10-0-9-146:~$ cat /var/lib/jenkins/.ssh/id_rsa
```

i-0cc39b1ffde18a629 (EKS-Jenkins)

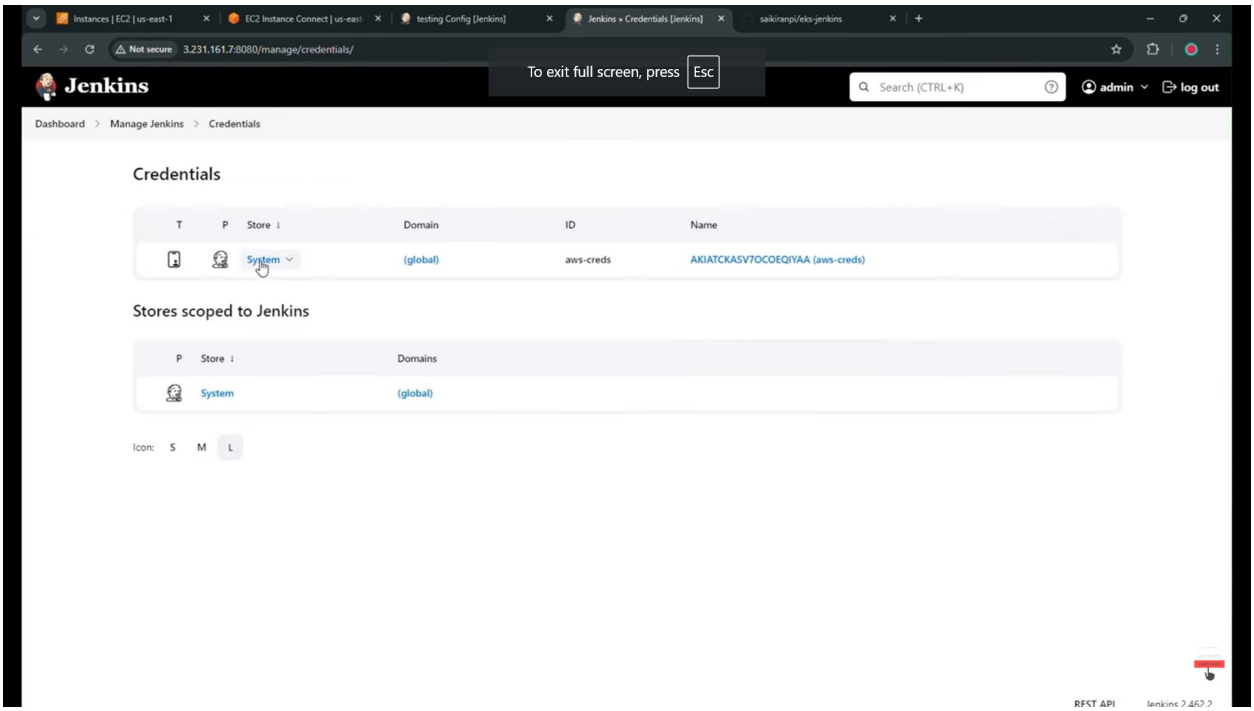
PublicIPs: 3.231.161.7 PrivateIPs: 10.0.9.146



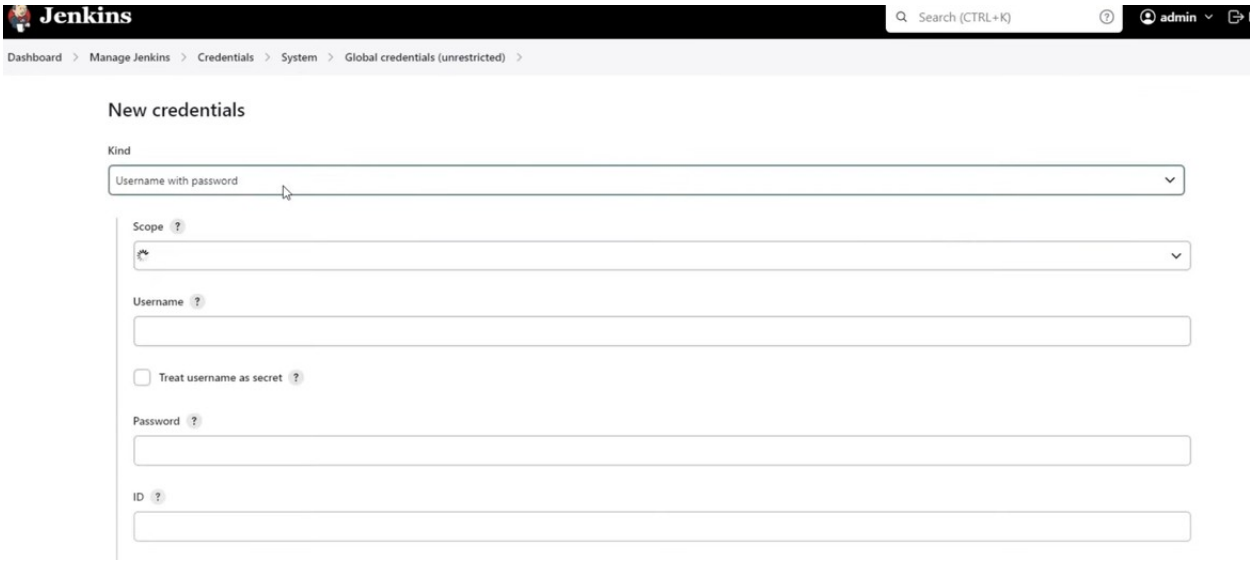
come back to Jenkins dashboard

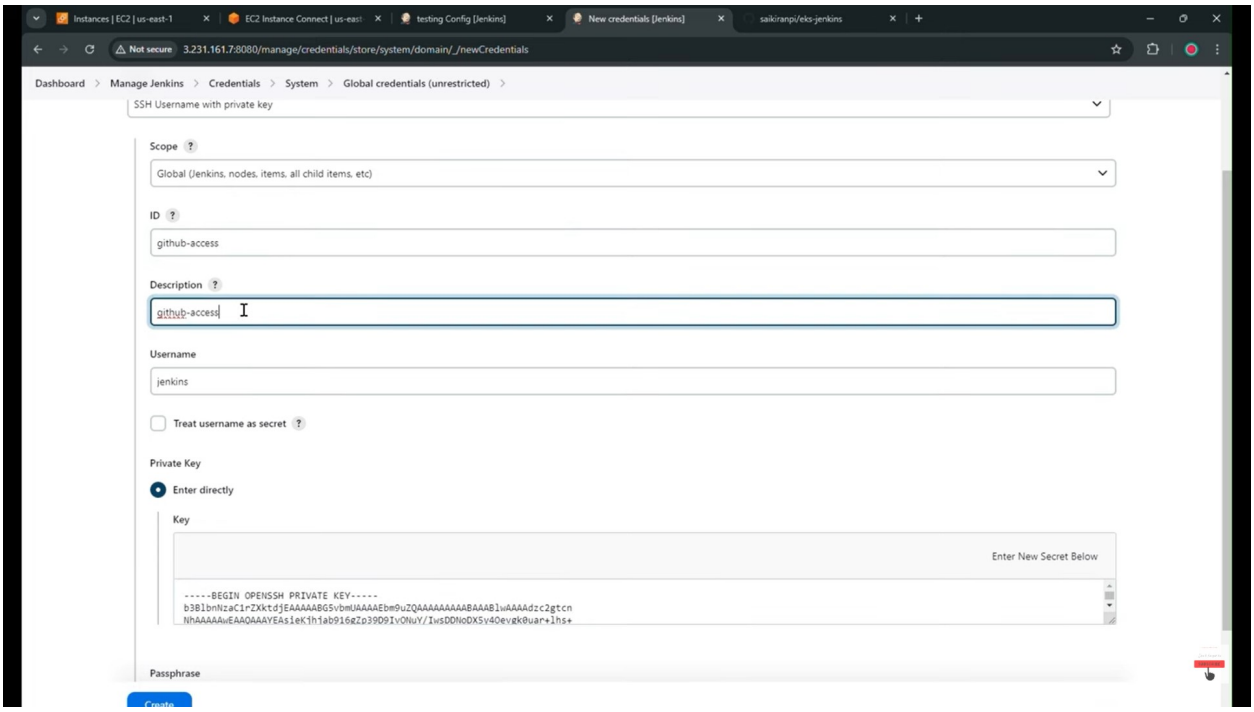


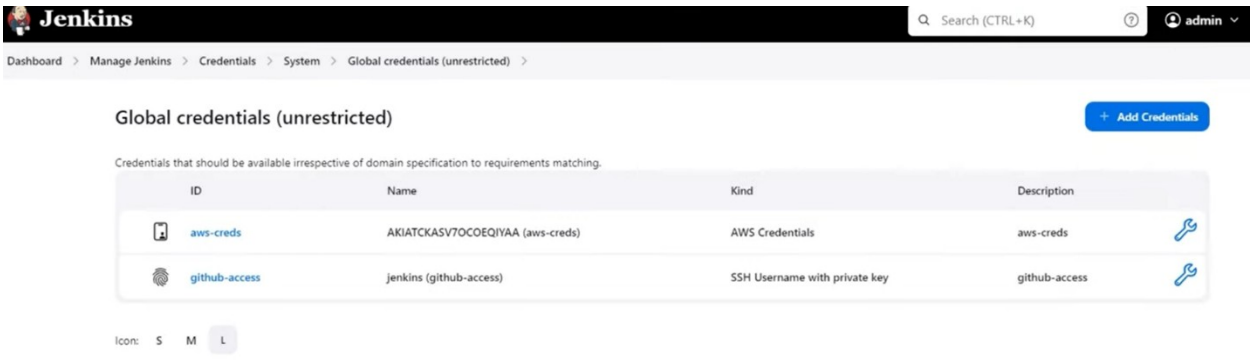
manage jenkins- credentials-system



global credentials- Add Credentials





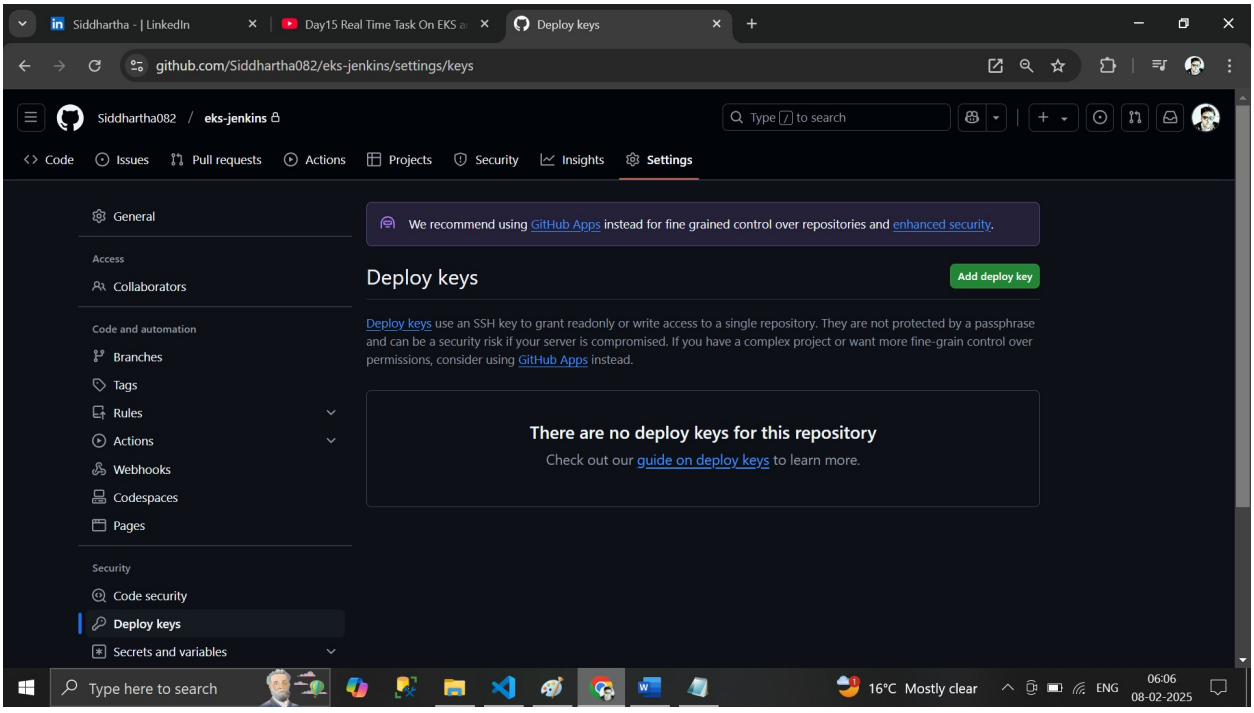


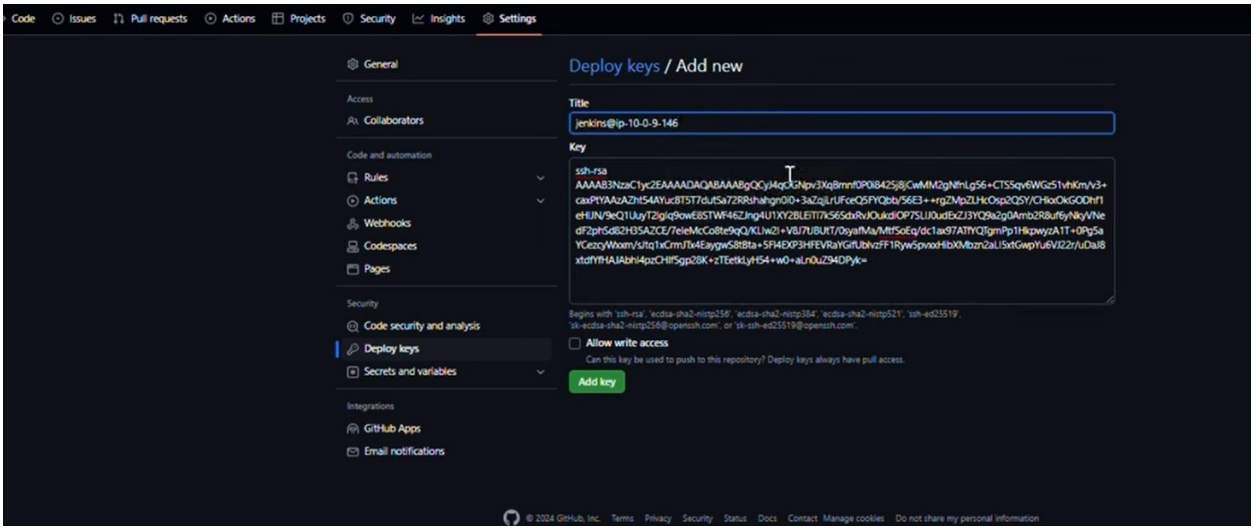
 # do it in GitHub repo -- go to server ssh . cp the public key & paste it in github credentials

-NOW ON GITHUB PRIVATE REPO WITH PUBLIC key
CAT PUBLICKKEY

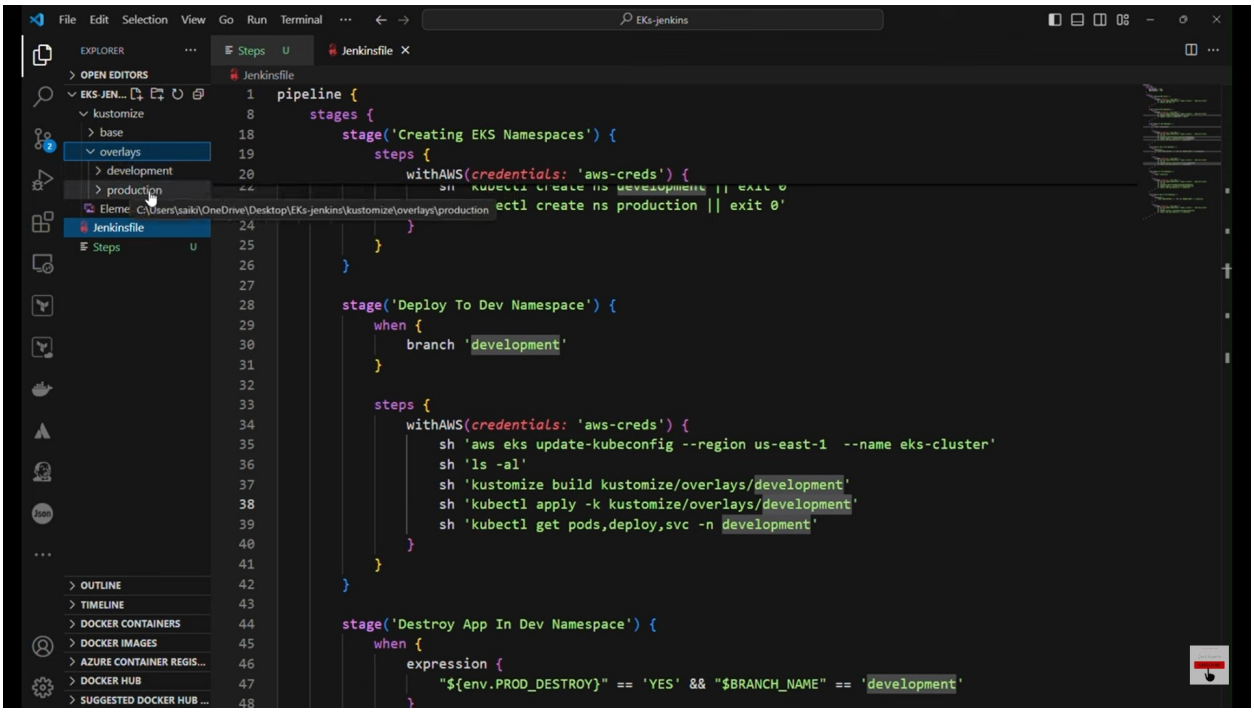
```
jenkins@ip-10-0-9-146:~$ cat /var/lib/jenkins/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCyJ4qOGNpv3XqBmnf0P0i8425j8jCwMM2gNfnLg56+CTS5qv6WGz51vhKm/v3+caxPtYAAzAZht54AYuc8T5T7dutSa72RRshahgn0i0+3aZqjLrUFceQ5FYQbb/56E3++rgZMpZLHcOsp2QSY/CHkxOkGODhf1eHlJN/9eQ1UuyT2lgIq9owE8STWF46ZJng4U1XY2BLEiTl7k56SdxRvJOukdiOP7SLIJ0udEx2J3YQ9a2g0Amb2R8uf6yNkyVNedF2phSd82H35AZCE/7eleMcCo8te9qQ/KLiw2I+V8J7tJBUtT/0syafMa/MtfSoEq/dc1ax97ATfYQTgmPp1HkpwyzA1T+0Pg5aYCezcyWxxm/sJtq1xCrmJTx4EaygwS8tBta+5FI4EXP3HFEVRaYGifUb1vzFFlRyw5pvxxHibXMbzn2aLI5xtGwpYu6VJ22r/uDaJ8xtdfYfHAJAbhI4pzCHIf5gp28K+zTEetkLyH54+w0+aLn0uZ94DPyk= jenkins@ip-10-0-9-146
jenkins@ip-10-0-9-146:~$
```

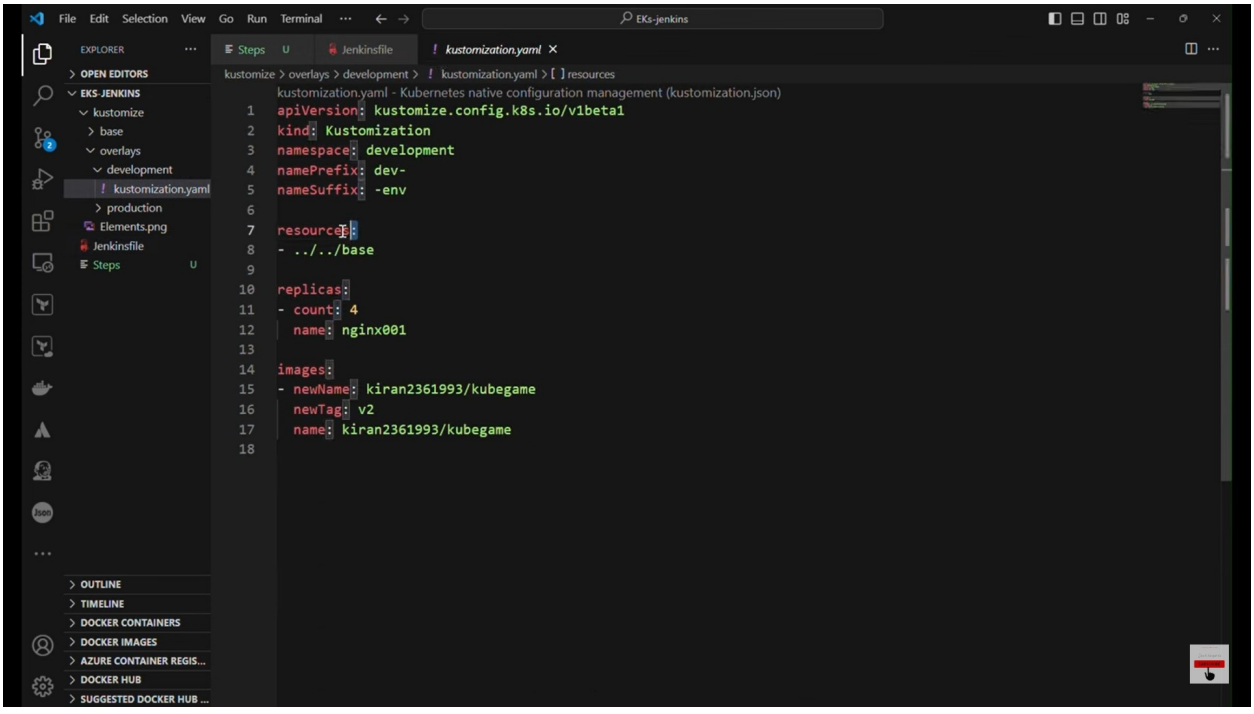
#github repo - setting – deploy keys

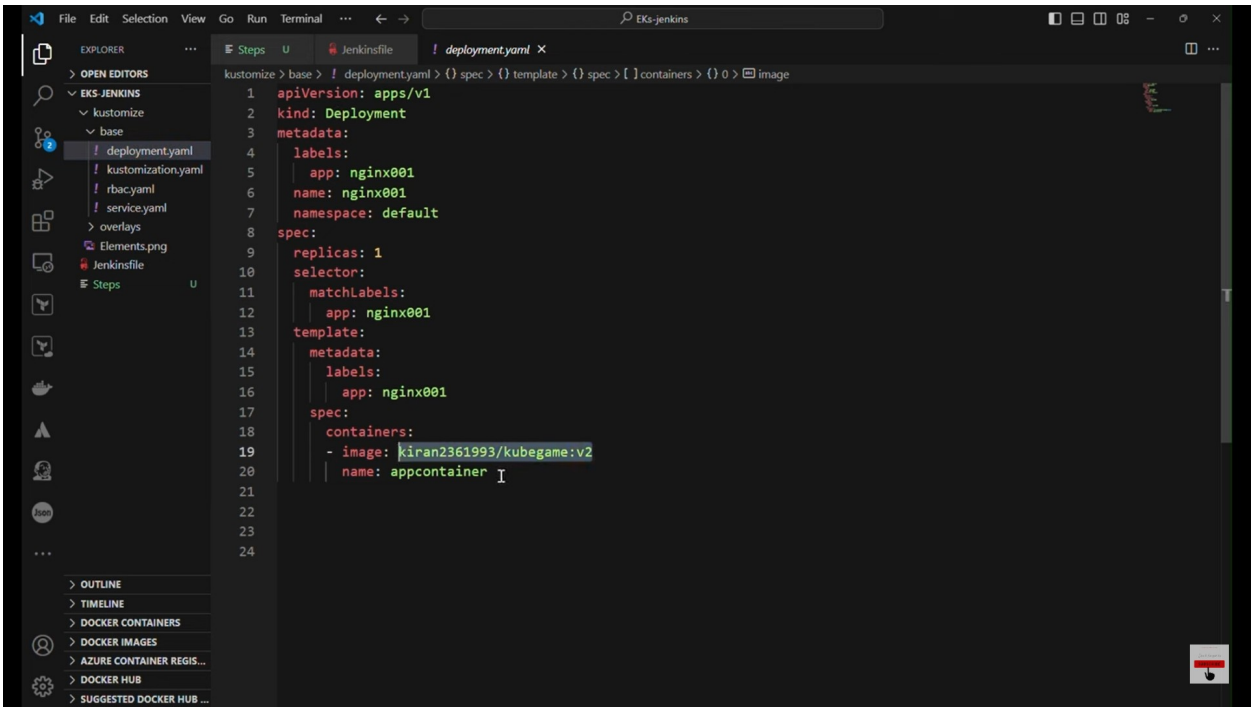


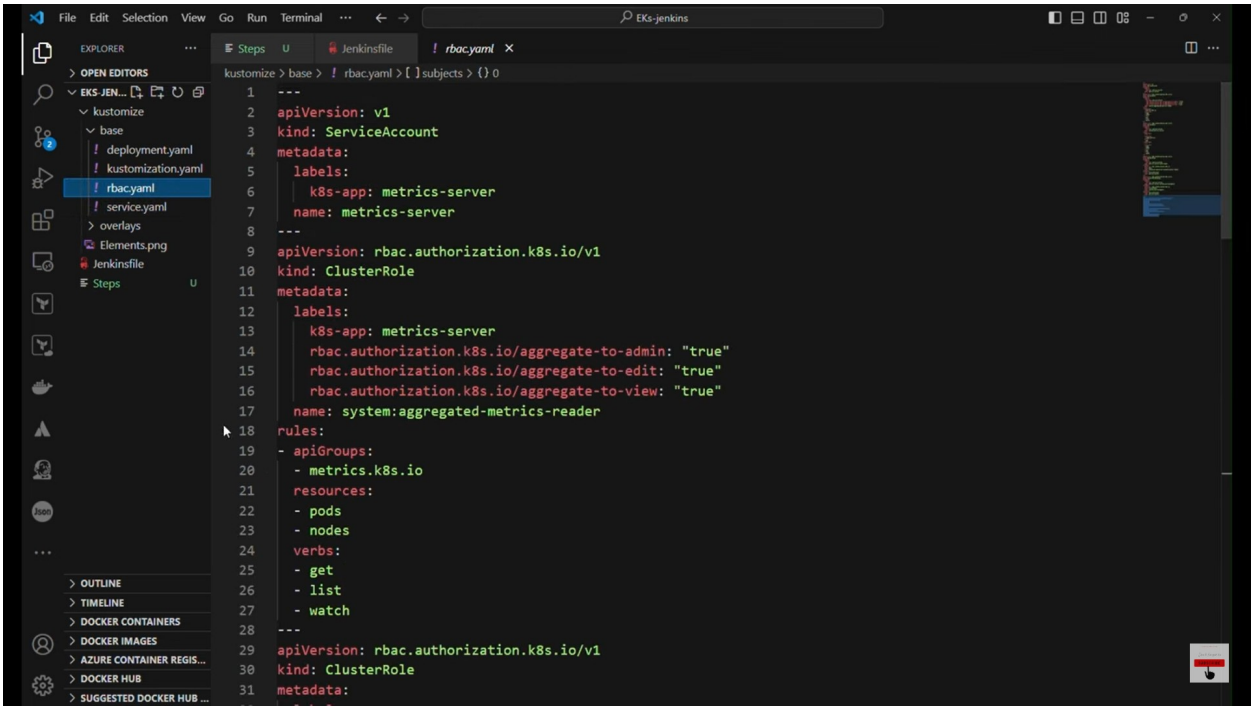


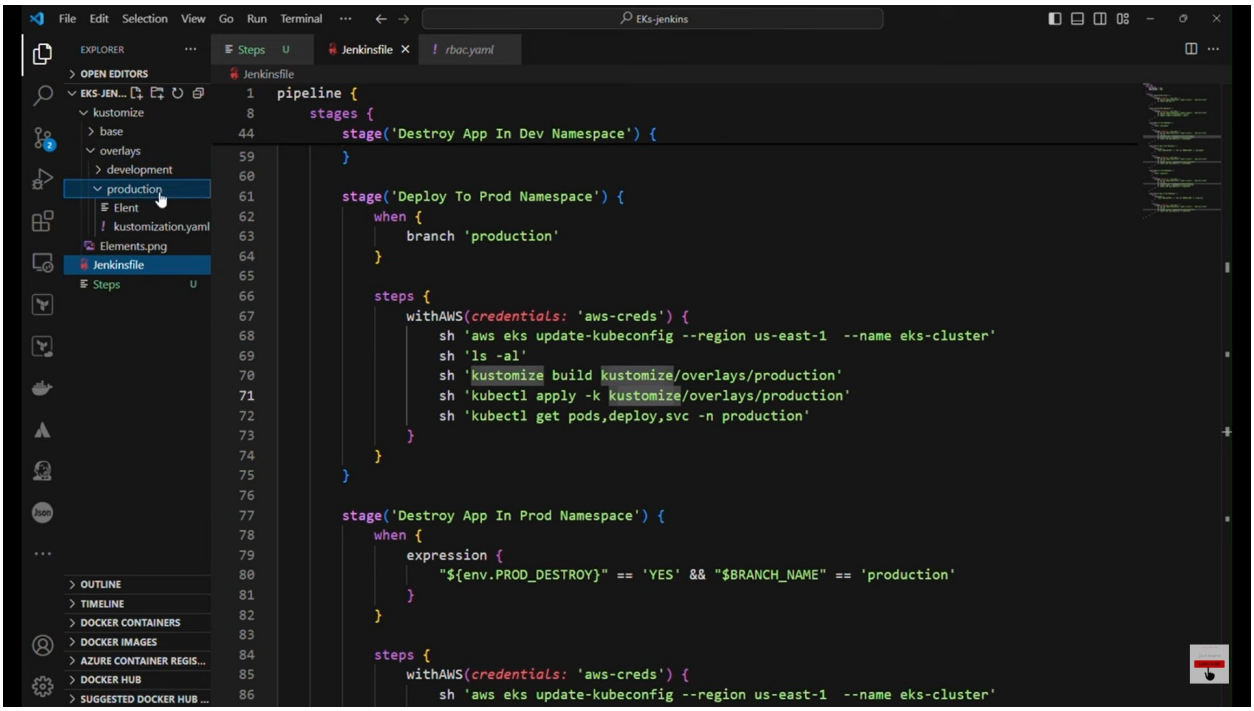
now create a developmnet branch and push the code to private REPO

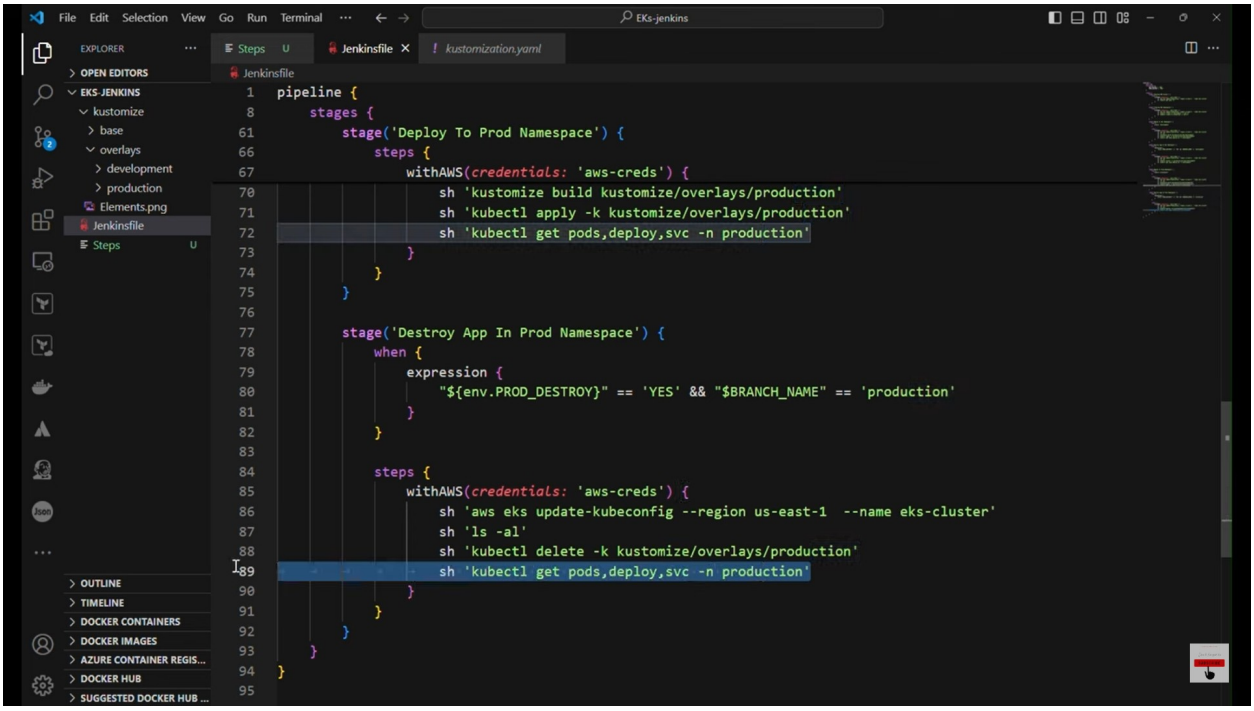




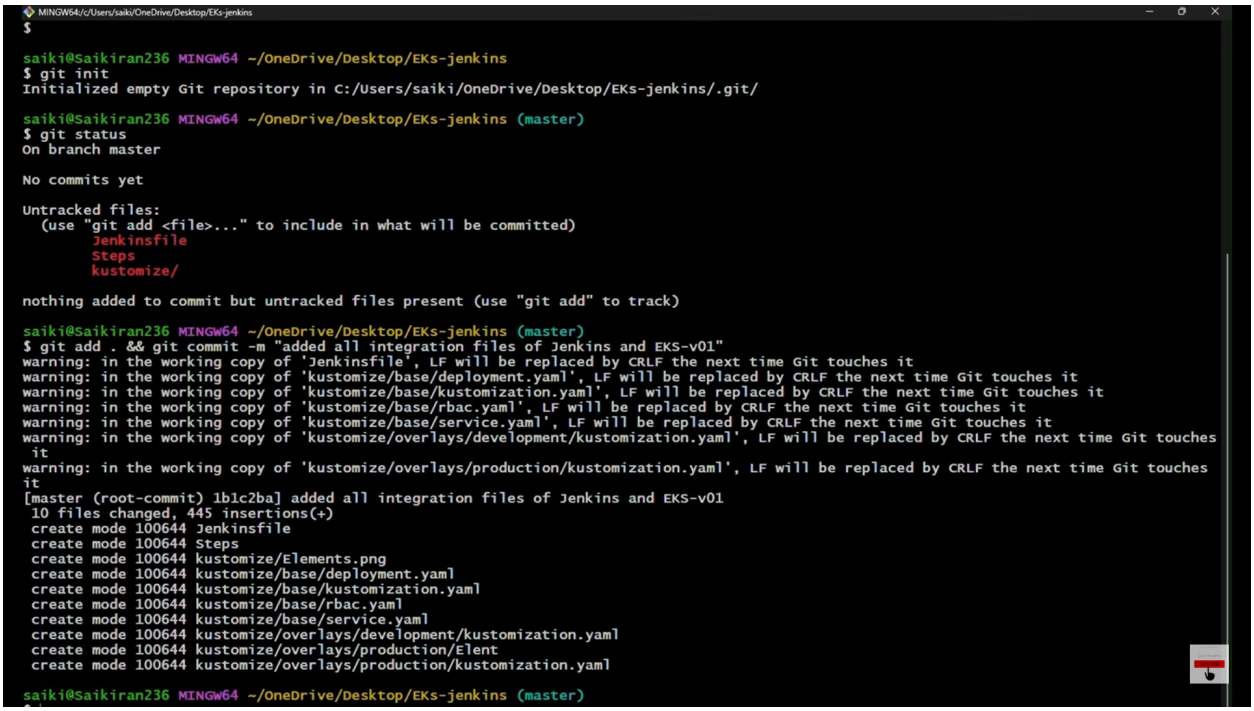

File Edit Selection View Go Run Terminal
EKs-jenkins
EXPLORER
Steps U
Jenkinsfile
deployment.yaml
kustomize > base > deployment.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > image
OPEN EDITORS
EKS-JENKINS
kustomize
base
deployment.yaml
kustomization.yaml
rbac.yaml
service.yaml
overlays
Elements.png
Jenkinsfile
Steps U
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGIS...
DOCKER HUB
SUGGESTED DOCKER HUB ...
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 labels:
5 app: nginx001
6 name: nginx001
7 namespace: default
8 spec:
9 replicas: 1
10 selector:
11 matchLabels:
12 app: nginx001
13 template:
14 metadata:
15 labels:
16 app: nginx001
17 spec:
18 containers:
19 - image: kiran2361993/kubegame:v2
20 name: appcontainer
21
22
23
24


File Edit Selection View Go Run Terminal
EKs-jenkins
EXPLORER
Steps U
Jenkinsfile
rbac.yaml
kustomize > base > rbac.yaml > [ ] subjects > {} 0
OPEN EDITORS
EKS-JEN...
kustomize
base
deployment.yaml
kustomization.yaml
rbac.yaml
service.yaml
overlays
Elements.png
Jenkinsfile
Steps U
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGIS...
DOCKER HUB
SUGGESTED DOCKER HUB ...
1 ---
2 apiVersion: v1
3 kind: ServiceAccount
4 metadata:
5 labels:
6 k8s-app: metrics-server
7 name: metrics-server
8 ---
9 apiVersion: rbac.authorization.k8s.io/v1
10 kind: ClusterRole
11 metadata:
12 labels:
13 k8s-app: metrics-server
14 rbac.authorization.k8s.io/aggregate-to-admin: "true"
15 rbac.authorization.k8s.io/aggregate-to-edit: "true"
16 rbac.authorization.k8s.io/aggregate-to-view: "true"
17 name: system:aggregated-metrics-reader
18 rules:
19 - apiGroups:
20 - metrics.k8s.io
21 resources:
22 - pods
23 - nodes
24 verbs:
25 - get
26 - list
27 - watch
28 ---
29 apiVersion: rbac.authorization.k8s.io/v1
30 kind: ClusterRole
31 metadata:



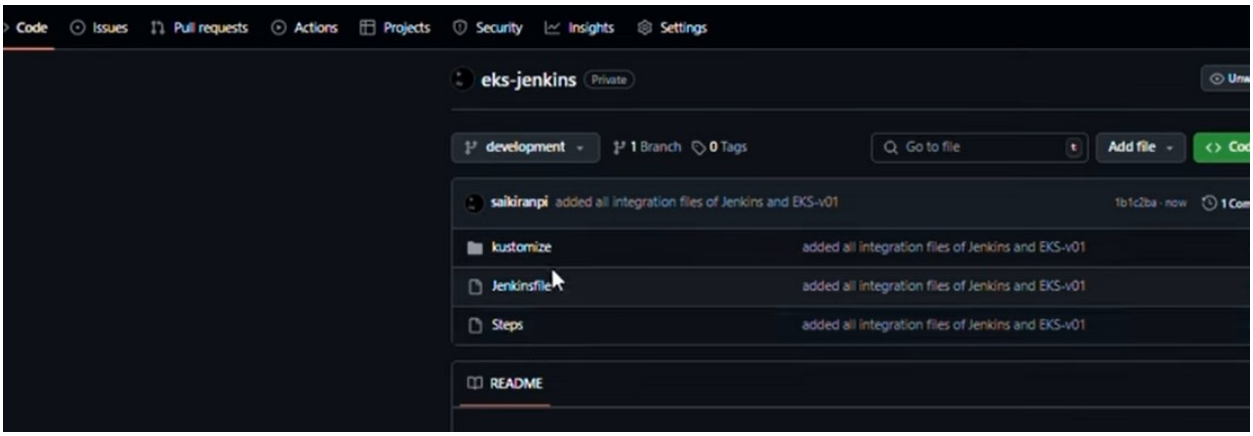


#push repo in GitHub




```
saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (master)
$

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (master)
$

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (master)
$ git status
On branch master
nothing to commit, working tree clean

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (master)
$ git branch
* master

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (master)
$ git branch master -m development

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (development)
$ git branch
* development

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (development)
$ git
```

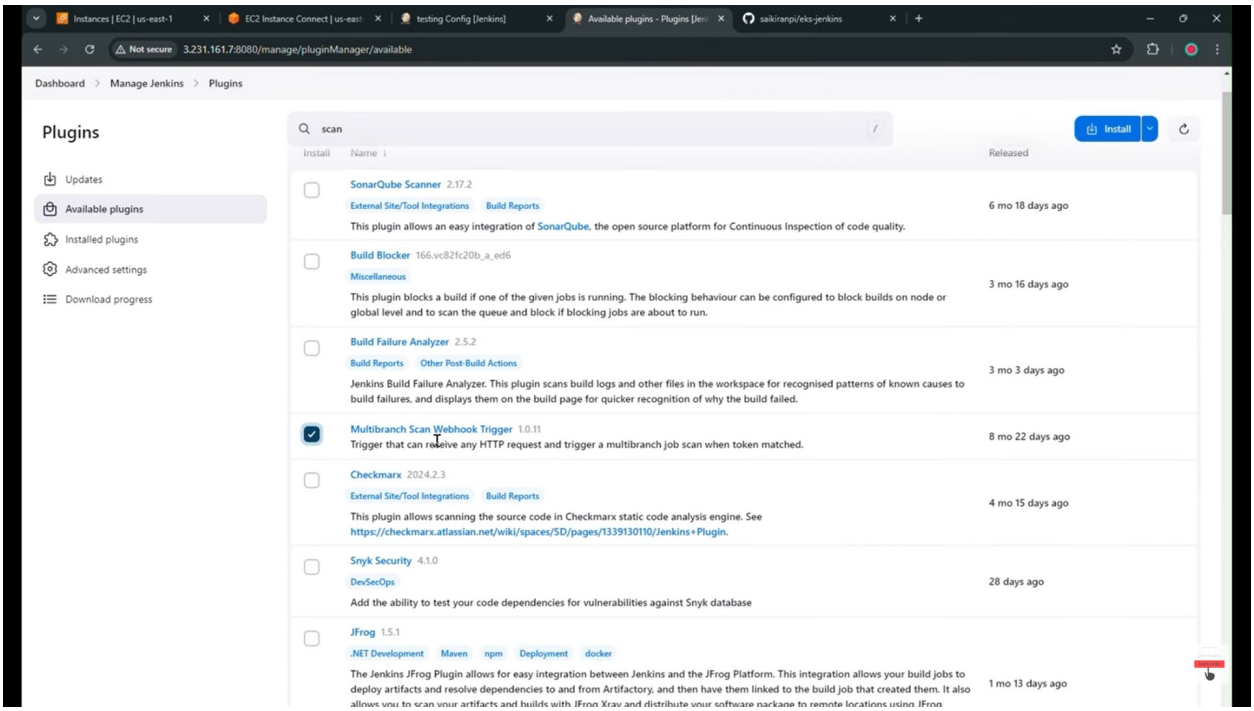
```
saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (master)
$ git branch master -m development

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (development)
$ git branch
* development

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (development)
$ git remote add origin https://github.com/saikiranpi/eks-jenkins.git

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (development)
$ git push origin development
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 12 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (17/17), 101.40 KiB | 14.48 MiB/s, done.
Total 17 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/saikiranpi/eks-jenkins.git
 * [new branch]      development -> development

saiki@Saikiran236 MINGW64 ~/OneDrive/Desktop/EKs-jenkins (development)
$
```

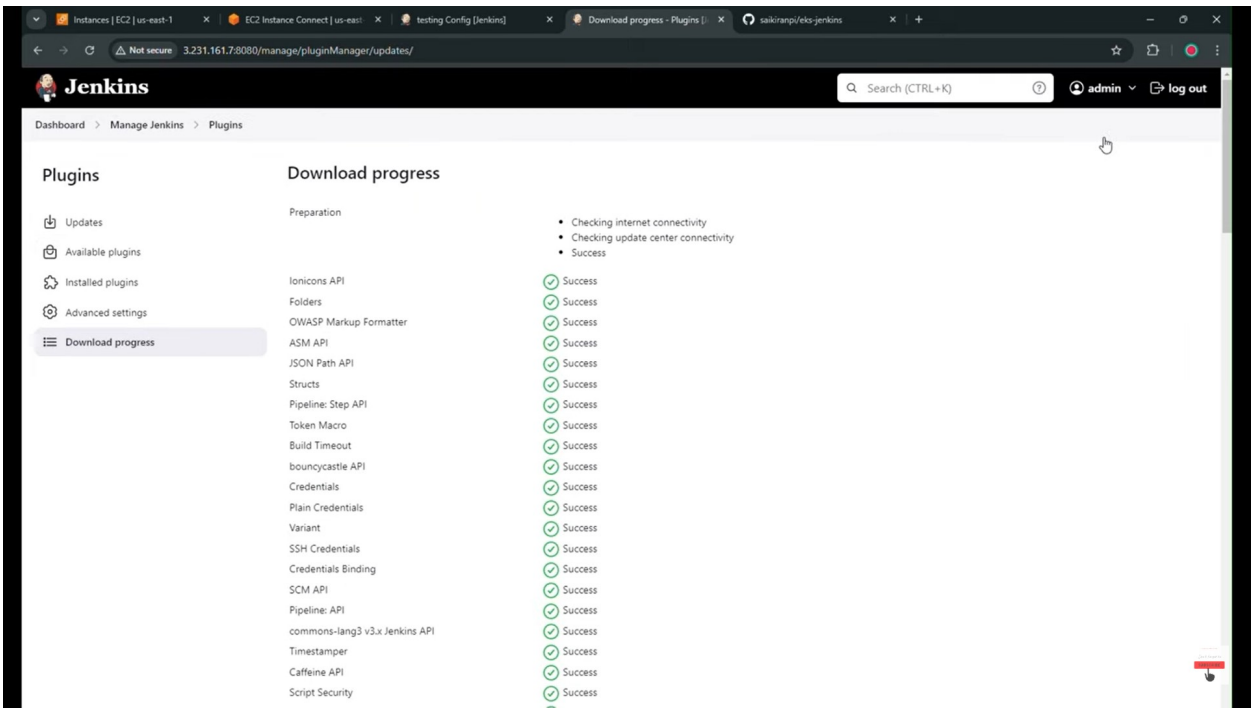


 # step

```
for automatic trigger need to add a plugin on jenins server  called 'multi branch scan webhook triger
```

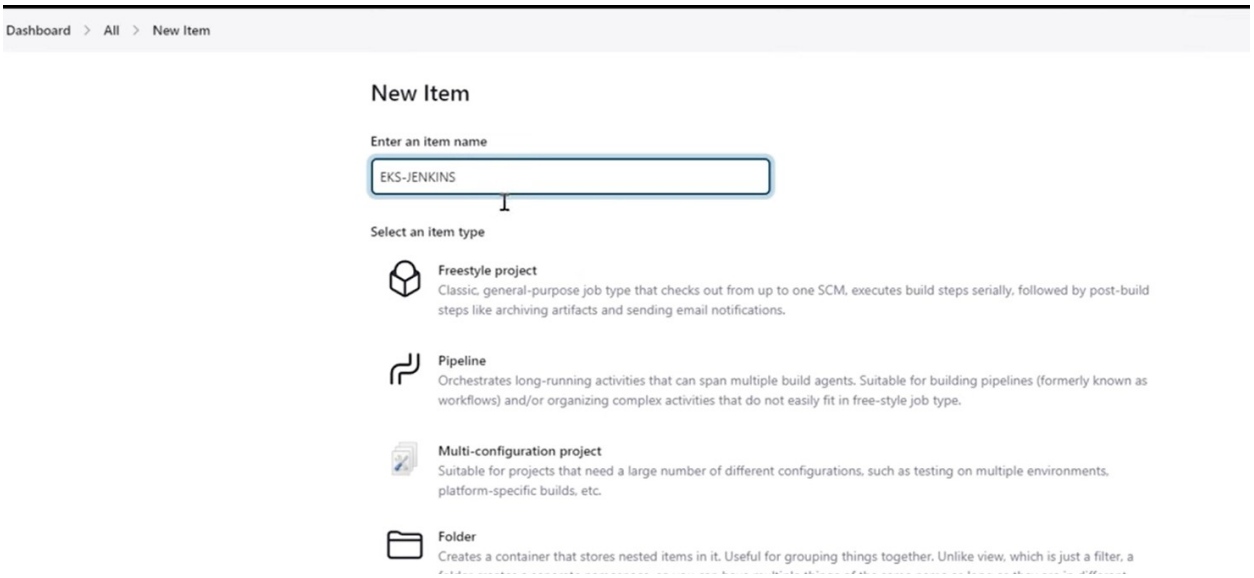
 # Come back to Jenkins- manage Jenkins-plugins – available plugins

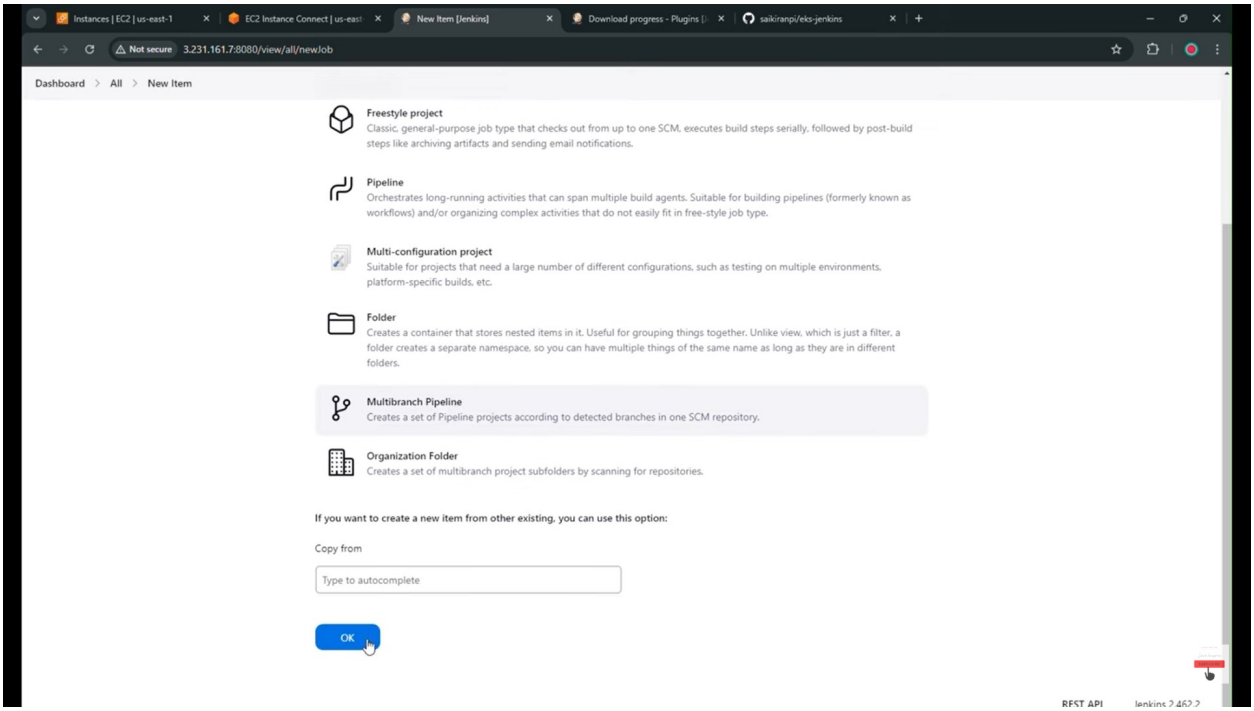


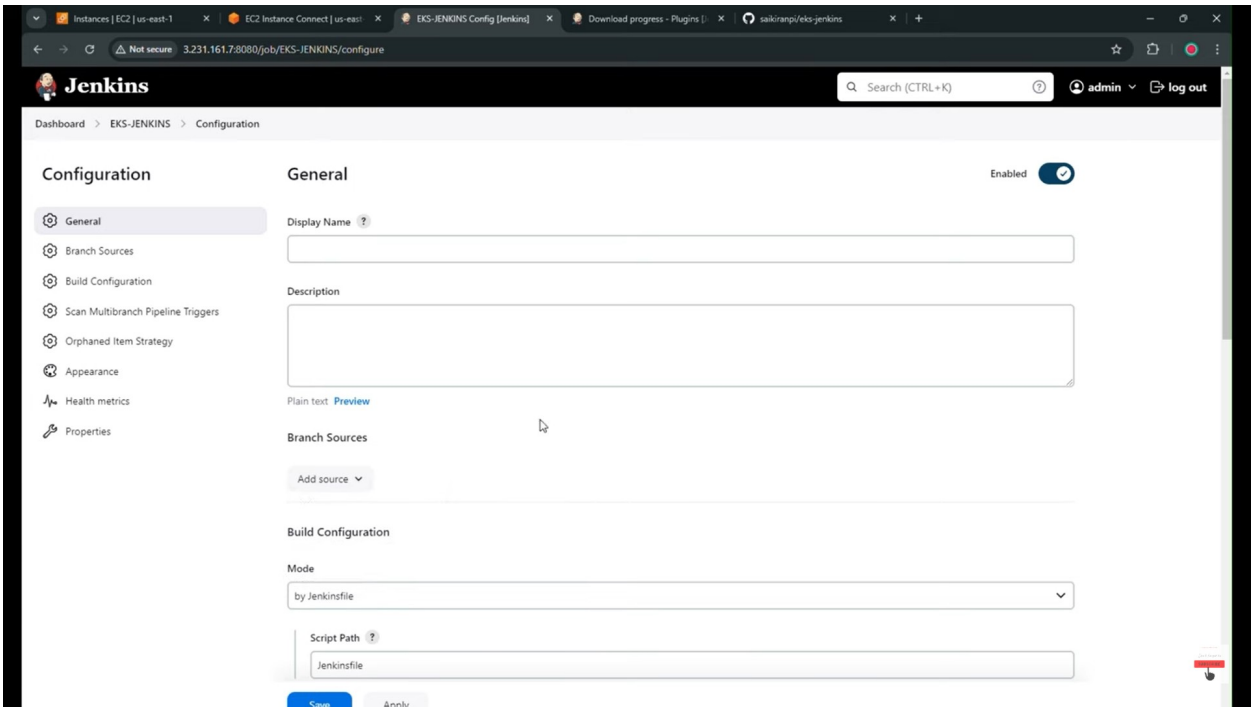


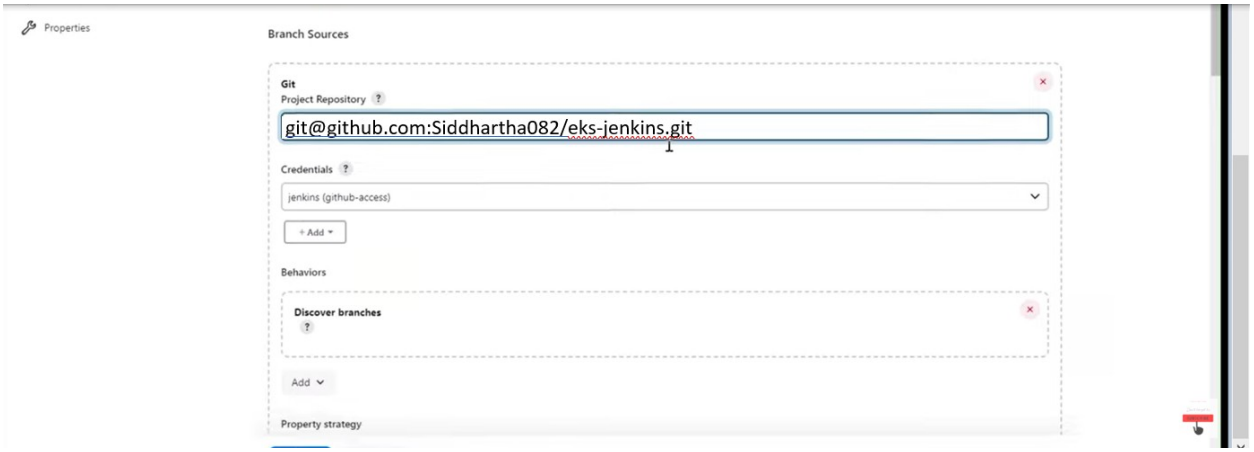

```
Create a new pipeline with Multibranch and select git and give the credentials accoridngly .
```

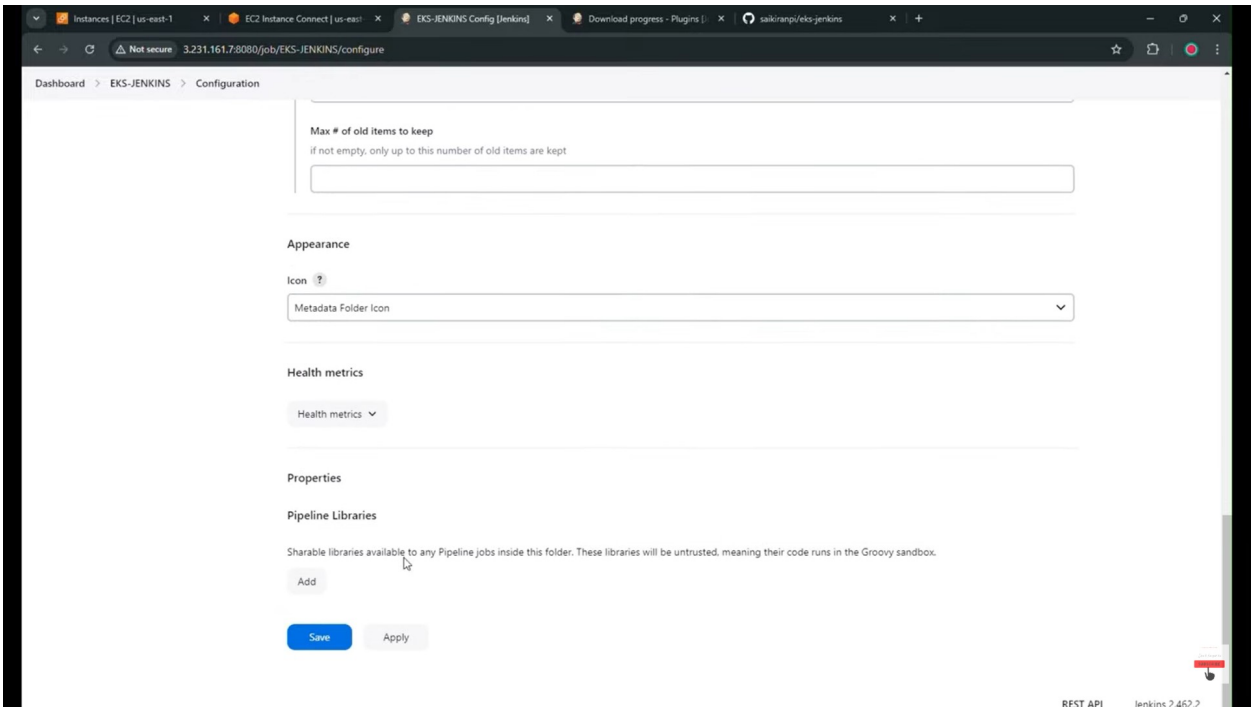
 # Jenkins- new item


Dashboard > All > New Item
New Item
Enter an item name
EKS-JENKINS
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a

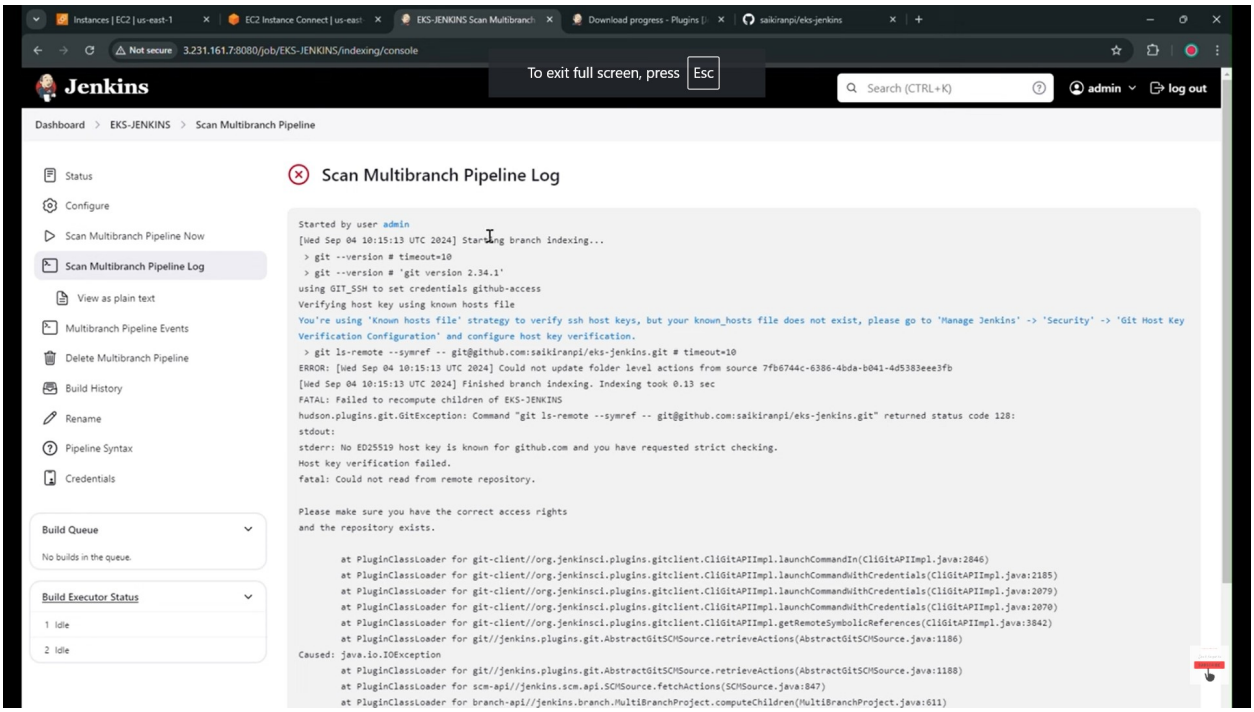

Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
New Item [Jenkins]
Download progress - Plugins
saikiranpi/eks-jenkins
Not secure 3.231.161.7:8080/view/all/newJob
Dashboard > All > New Item
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.
If you want to create a new item from other existing, you can use this option:
Copy from
Type to autocomplete
OK
REST API
Jenkins 2.462.2


Jenkins
Search (CTRL+K)
admin
log out
Dashboard > EKS-JENKINS > Configuration
Configuration
General
Enabled
General
Branch Sources
Build Configuration
Scan Multibranch Pipeline Triggers
Orphaned Item Strategy
Appearance
Health metrics
Properties
Display Name
Description
Plain text Preview
Branch Sources
Add source
Build Configuration
Mode
by Jenkinsfile
Script Path
Jenkinsfile


Properties
Branch Sources
Git
Project Repository
git@github.com:Siddhartha082/eks-jenkins.git
Credentials
jenkins (github-access)
+ Add
Behaviors
Discover branches
Add
Property strategy

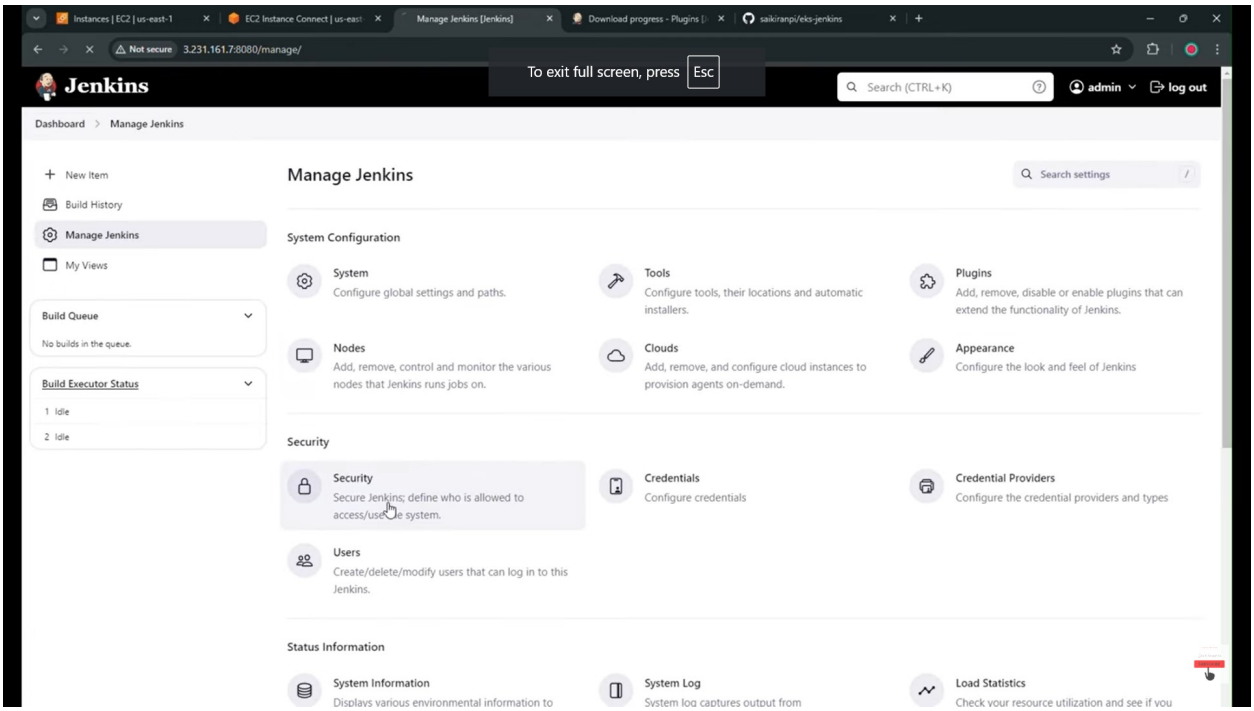


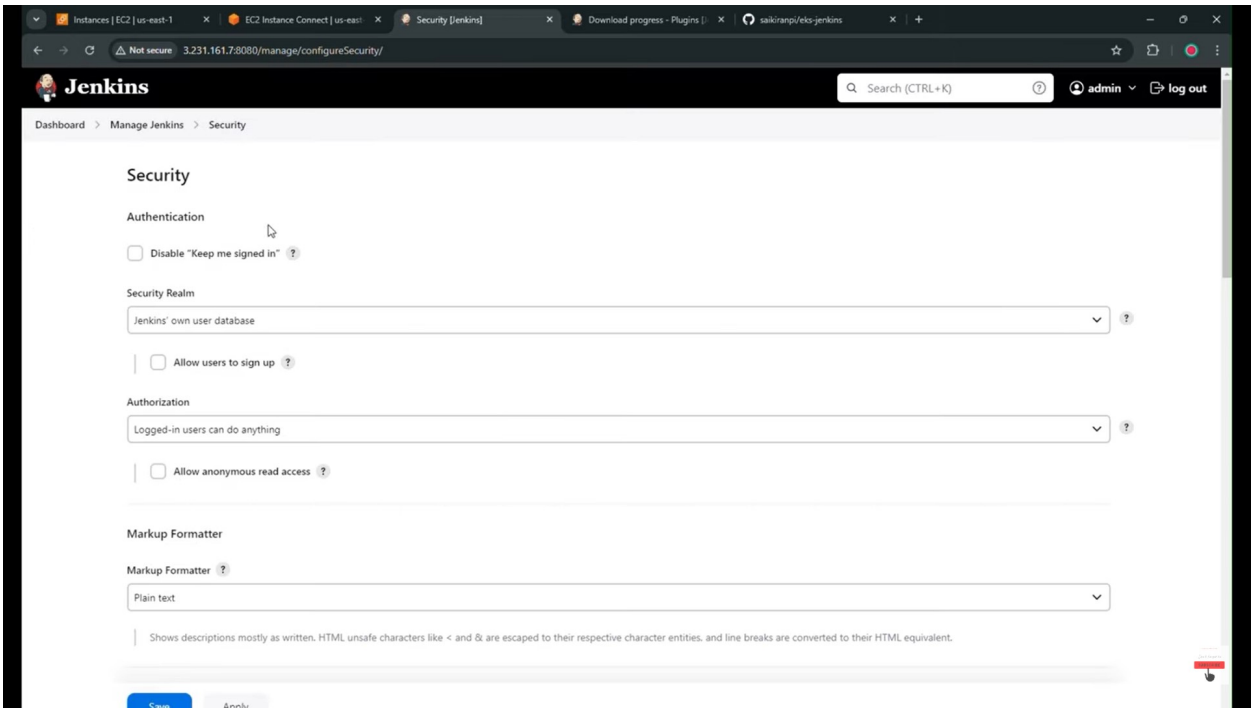
 # error

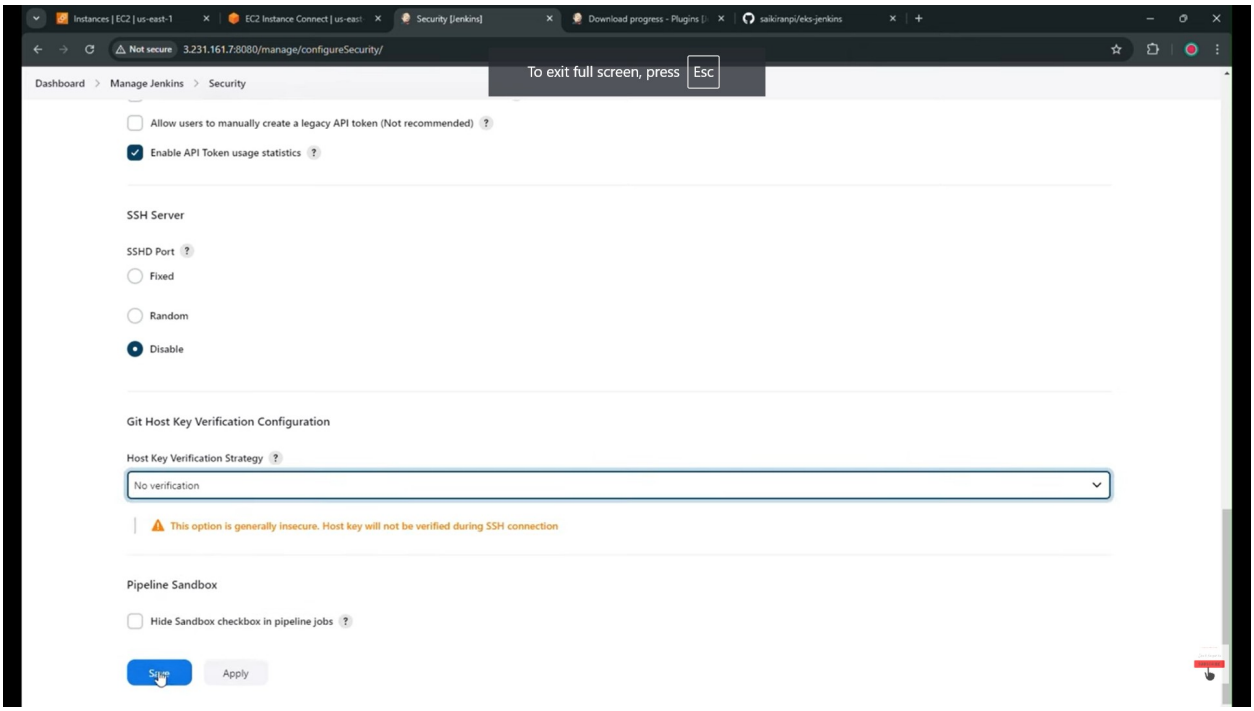


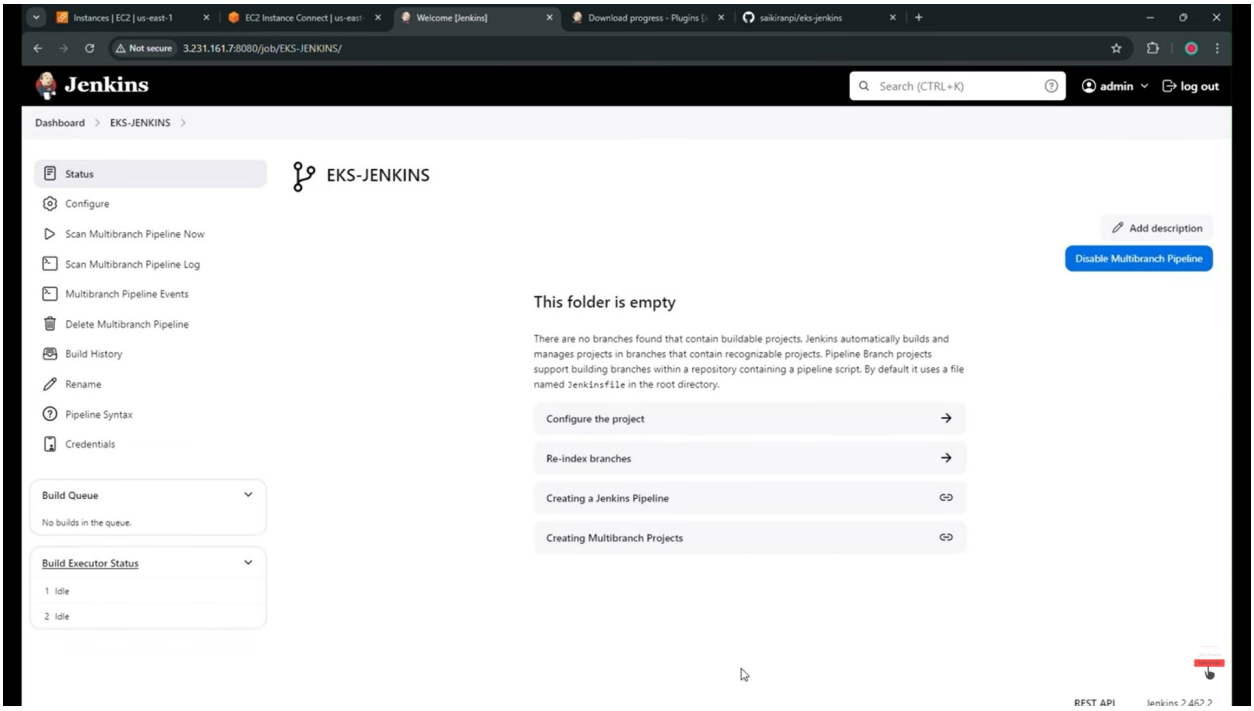

```
You will get error which took 2 hours to resolve got to manage jenkins   > Security > turnoff host key verification
```

 # Jenkins – manage Jenkins- security


Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
Manage Jenkins [Jenkins]
Download progress - Plugins
saikiranpi/eks-jenkins
Not secure 3.231.161.7:8080/manage/
To exit full screen, press Esc
Jenkins
Search (CTRL+K)
admin
log out
Dashboard > Manage Jenkins
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Manage Jenkins
Search settings
System Configuration
System
Configure global settings and paths.
Tools
Configure tools, their locations and automatic installers.
Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
Clouds
Add, remove, and configure cloud instances to provision agents on-demand.
Appearance
Configure the look and feel of Jenkins
Security
Security
Secure Jenkins; define who is allowed to access/use the system.
Credentials
Configure credentials
Credential Providers
Configure the credential providers and types
Users
Create/delete/modify users that can log in to this Jenkins.
Status Information
System Information
Displays various environmental information to
System Log
System log captures output from
Load Statistics
Check your resource utilization and see if you


Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
Security [Jenkins]
Download progress - Plugins
saikiranpi/eks-jenkins
Not secure 3.231.161.7:8080/manage/configureSecurity/
Jenkins
Search (CTRL+K)
admin
log out
Dashboard > Manage Jenkins > Security
Security
Authentication
Disable "Keep me signed in"
Security Realm
Jenkins' own user database
Allow users to sign up
Authorization
Logged-in users can do anything
Allow anonymous read access
Markup Formatter
Markup Formatter
Plain text
Shows descriptions mostly as written. HTML unsafe characters like < and & are escaped to their respective character entities. and line breaks are converted to their HTML equivalent.
Save
Apply





Once done Install kustomize

```
cd /usr/local/bin
https://github.com/kubernetes-sigs/kustomize/releases/download/kustomize%2Fv5.2.1/kustomize_v5.2.1_linux_amd64.tar.gz
```

```
jenkins@ip-10-0-9-146:~$ exit
logout
root@ip-10-0-9-146:~#
```

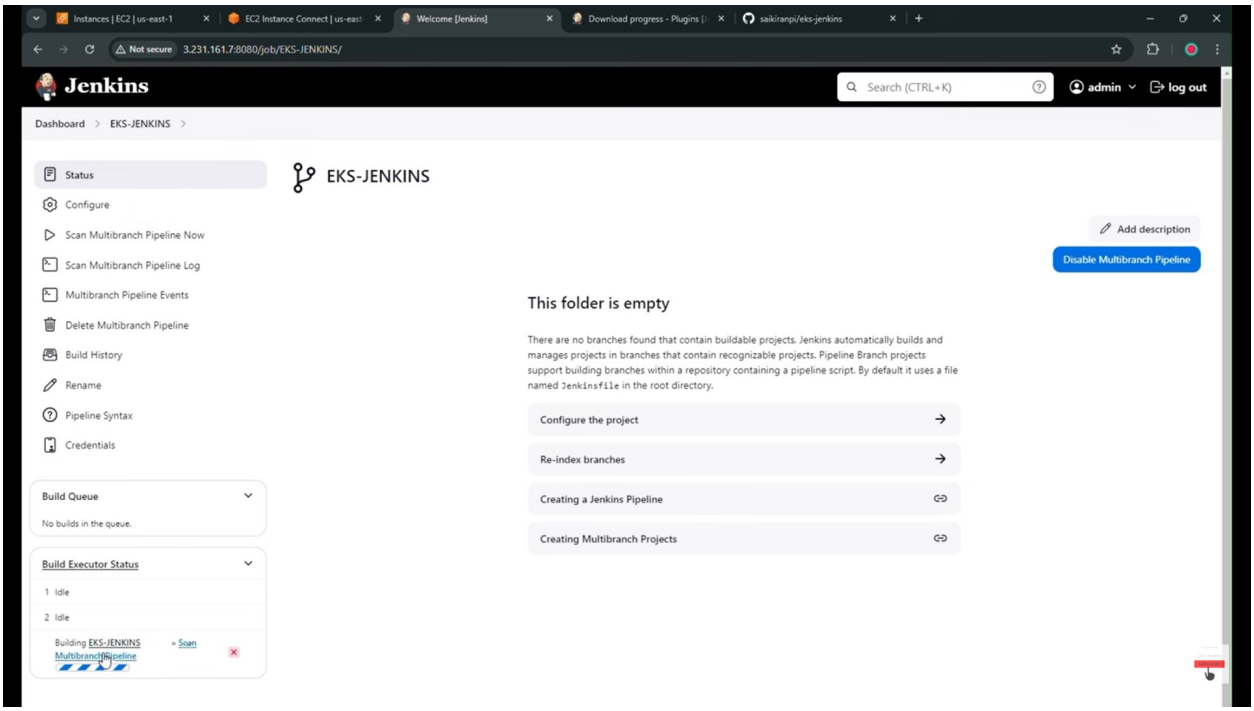
```
root@ip-10-0-9-146:~# cd /usr/local/bin/
root@ip-10-0-9-146:/usr/local/bin# wget https://github.com/kubernetes-sigs/kustomize/releases/download/kustomize%2Fv5.2.1/kustomize_v5.2.1_linux_amd64.tar.gz
--2024-09-04 10:16:44--  https://github.com/kubernetes-sigs/kustomize/releases/download/kustomize%2Fv5.2.1/kustomize_v5.2.1_linux_amd64.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/133067498/88dbd3c1-b048-4327-a04f-834d97f5acb6?X-Amz-Algorithm=AWS4-HMAC-S
256&X-Amz-Credential=releaseassetproduction%2F20240904%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240904T101644Z&X-Amz-Expires=300&X-Amz-Signature=d81f97c2cef4
6d846e2013427af79496b5bbcf1d632ba6377684b19925bf36&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=133067498&response-content-disposition=attachment%3B%20fi
name%3Dkustomize_v5.2.1_linux_amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-09-04 10:16:44--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/133067498/88dbd3c1-b048-4327-a04f-834d97f5acb6?X-Amz-Algori
m=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240904%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240904T101644Z&X-Amz-Expires=300&X-Amz-Signatu
=d81f97c2cef4896d846e2013427af79496b5bbcf1d632ba6377684b19925bf36&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=133067498&response-content-disposition=att
hment%3B%20filename%3Dkustomize_v5.2.1_linux_amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5946650 (5.7M) [application/octet-stream]
Saving to: 'kustomize_v5.2.1_linux_amd64.tar.gz'

kustomize_v5.2.1_linux_amd64.tar.gz      100%[=====================================================================================>]   5.67M  --.-KB/s    in 0.05s

2024-09-04 10:16:44 (104 MB/s) - 'kustomize_v5.2.1_linux_amd64.tar.gz' saved [5946650/5946650]
```

```
root@ip-10-0-9-146:/usr/local/bin# tar zxvf kustomize_v5.2.1_linux_amd64.tar.gz
kustomize
root@ip-10-0-9-146:/usr/local/bin#
```

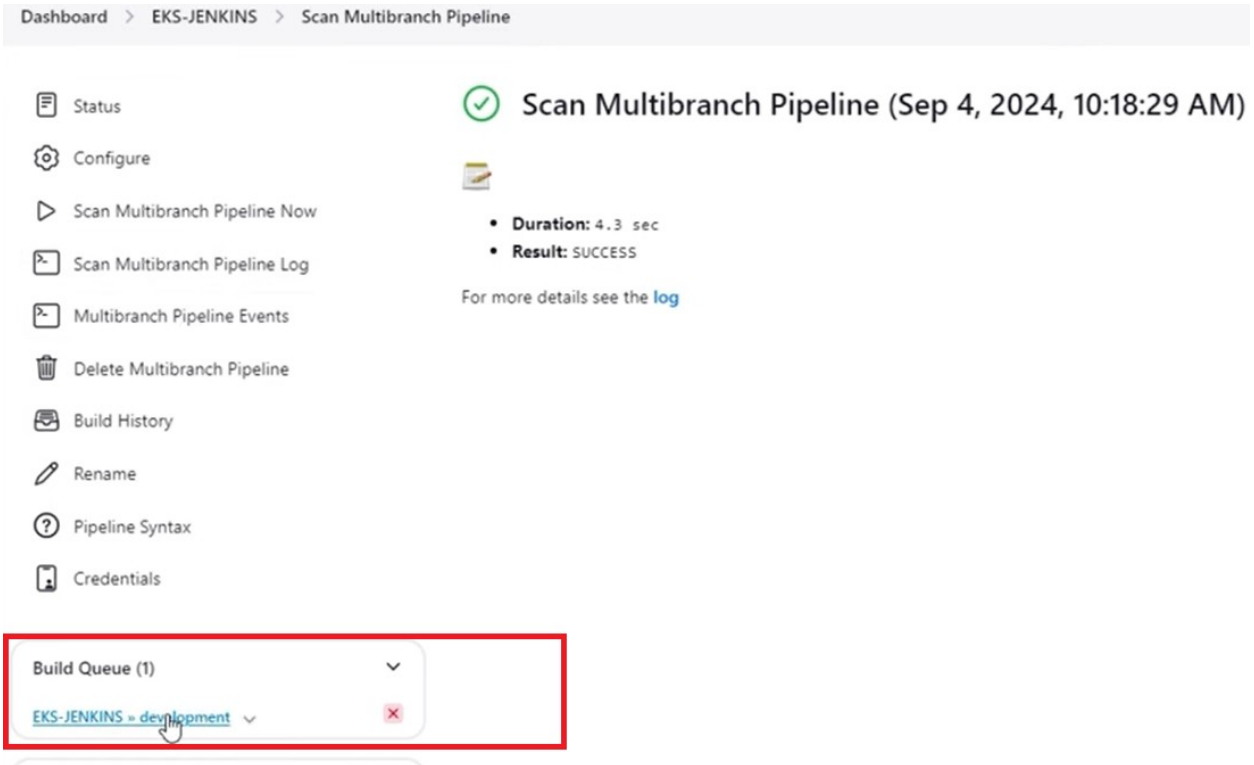
```
root@ip-10-0-9-146:/usr/local/bin#
root@ip-10-0-9-146:/usr/local/bin# ls -al
total 215644
drwxr-xr-x  2 root root      4096 Sep  4 10:16 .
drwxr-xr-x 11 root root      4096 Sep  4 09:28 ..
lrwxrwxrwx  1 root root        37 Sep  4 09:28 aws -> /usr/local/aws-cli/v2/current/bin/aws
lrwxrwxrwx  1 root root        47 Sep  4 09:28 aws_completer -> /usr/local/aws-cli/v2/current/bin/aws_completer
-rwx------  1 1001  127 143462400 Aug 19 11:52 eksctl
-rwxrwxrwx  1 root root  56381592 Sep  4 09:30 kubectl
-rwxr-xr-x  1 1001  127  15015936 Oct 19  2023 kustomize
-rw-r--r--  1 root root   5946650 Oct 19  2023 kustomize_v5.2.1_linux_amd64.tar.gz
root@ip-10-0-9-146:/usr/local/bin# rm -rf kustomize_v5.2.1_linux_amd64.tar.gz
root@ip-10-0-9-146:/usr/local/bin#
```

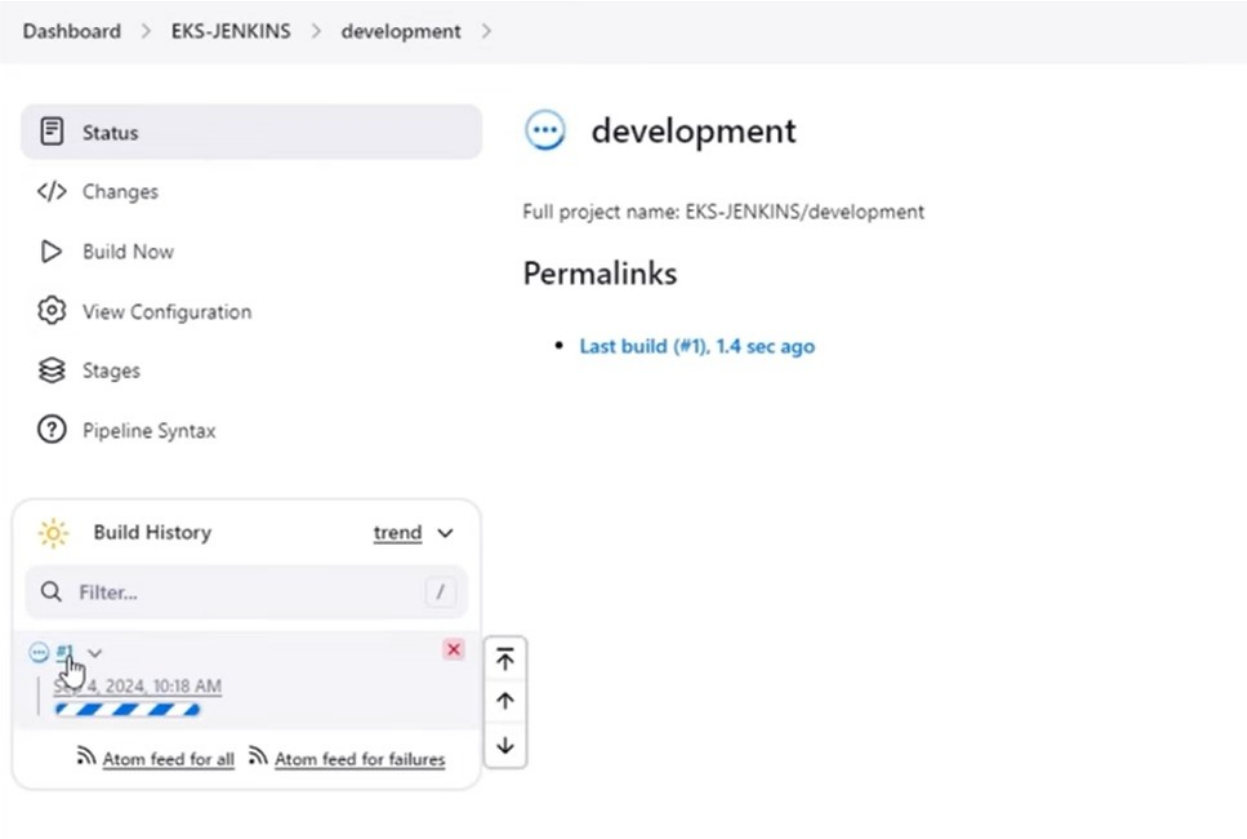
```
# do dummy testing by creating the file and push , Automatically a build should be starated when you select scan multibranch pipeline
```

 # click on scan mulibranch pipeline

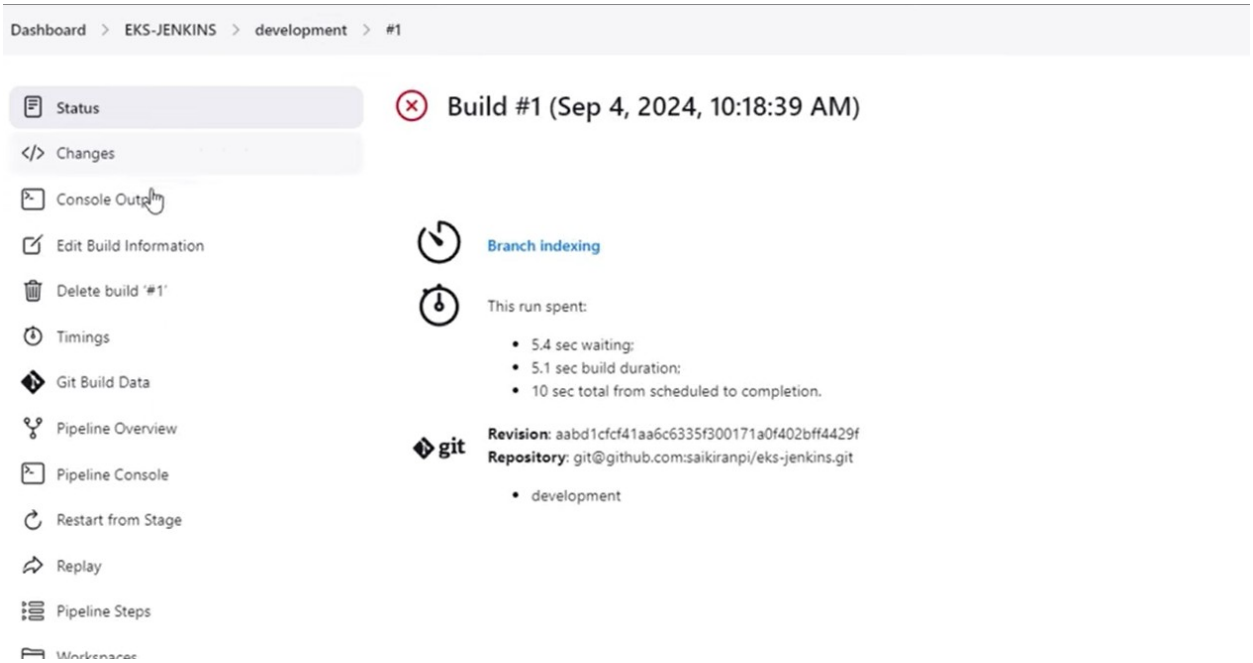

Jenkins
Search (CTRL+K)
admin
log out
Dashboard
EKS-JENKINS
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
Build History
Rename
Pipeline Syntax
Credentials
EKS-JENKINS
Add description
Disable Multibranch Pipeline
This folder is empty
There are no branches found that contain buildable projects. Jenkins automatically builds and manages projects in branches that contain recognizable projects. Pipeline Branch projects support building branches within a repository containing a pipeline script. By default it uses a file named Jenkinsfile in the root directory.
Configure the project
Re-index branches
Creating a Jenkins Pipeline
Creating Multibranch Projects
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Building EKS-JENKINS » Scan Multibranch Pipeline

 # job started

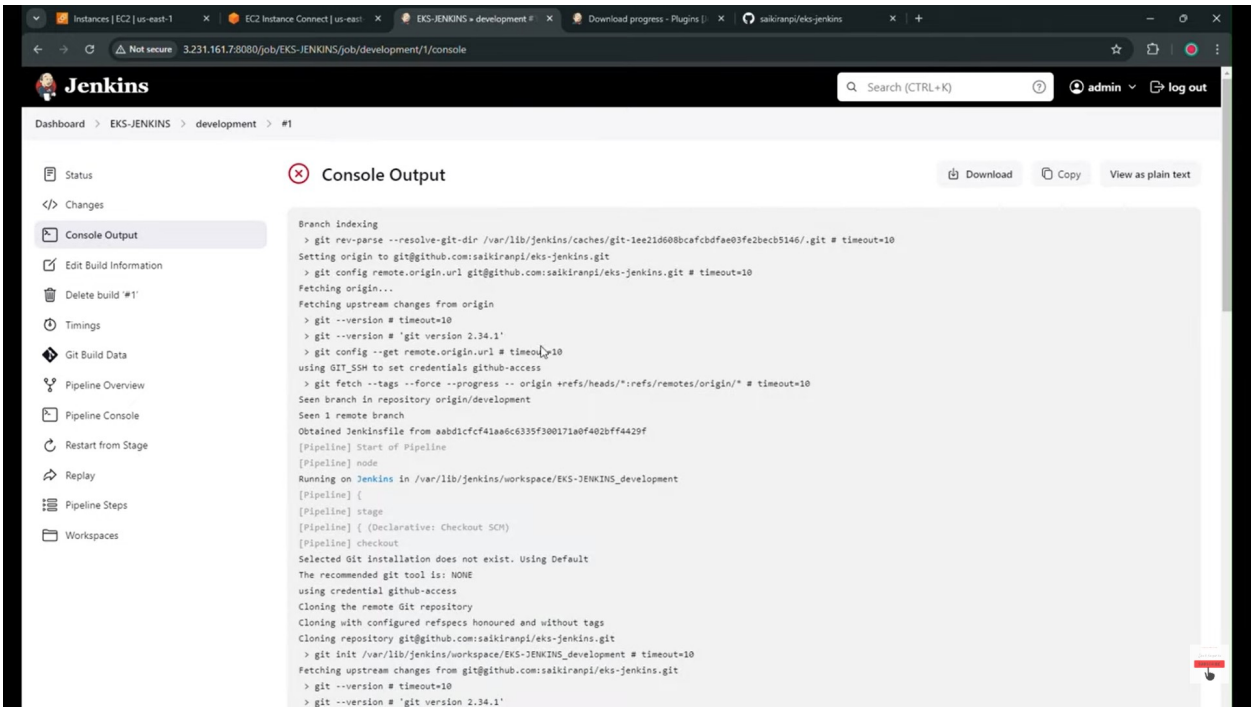

Dashboard
EKS-JENKINS
Scan Multibranch Pipeline
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
Build History
Rename
Pipeline Syntax
Credentials
Scan Multibranch Pipeline (Sep 4, 2024, 10:18:29 AM)
Duration: 4.3 sec
Result: SUCCESS
For more details see the log
Build Queue (1)
EKS-JENKINS » development

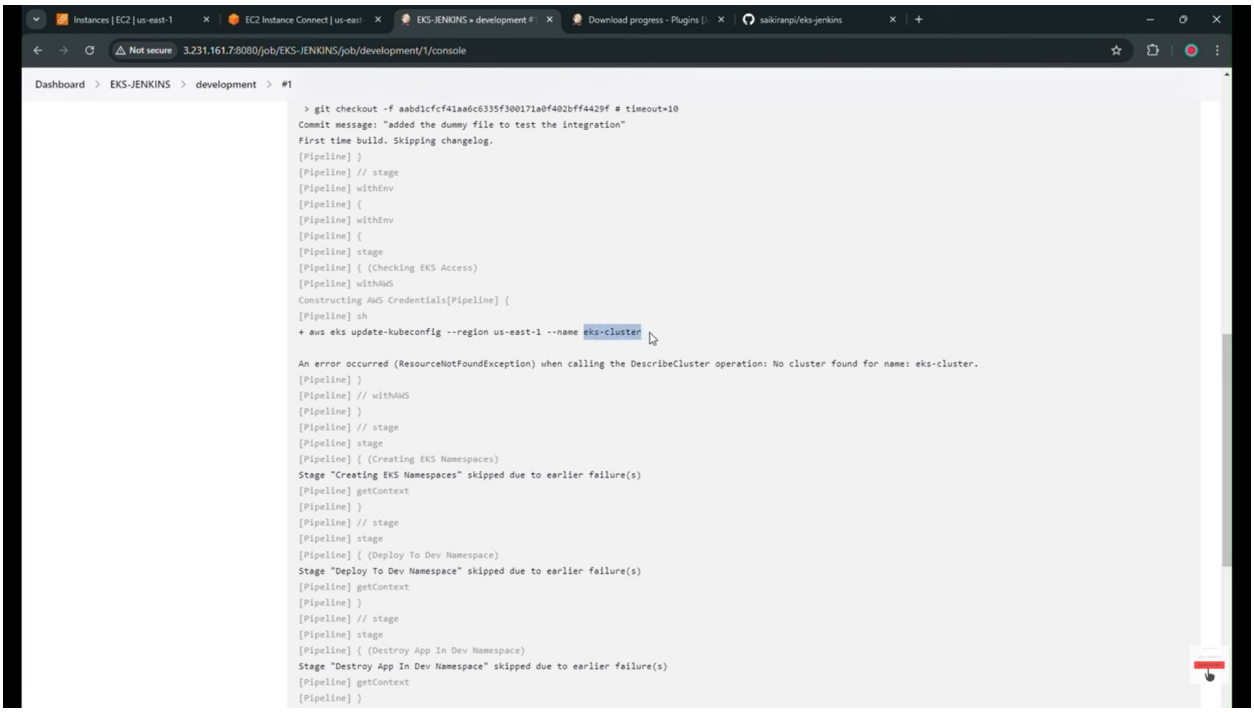


For status click above

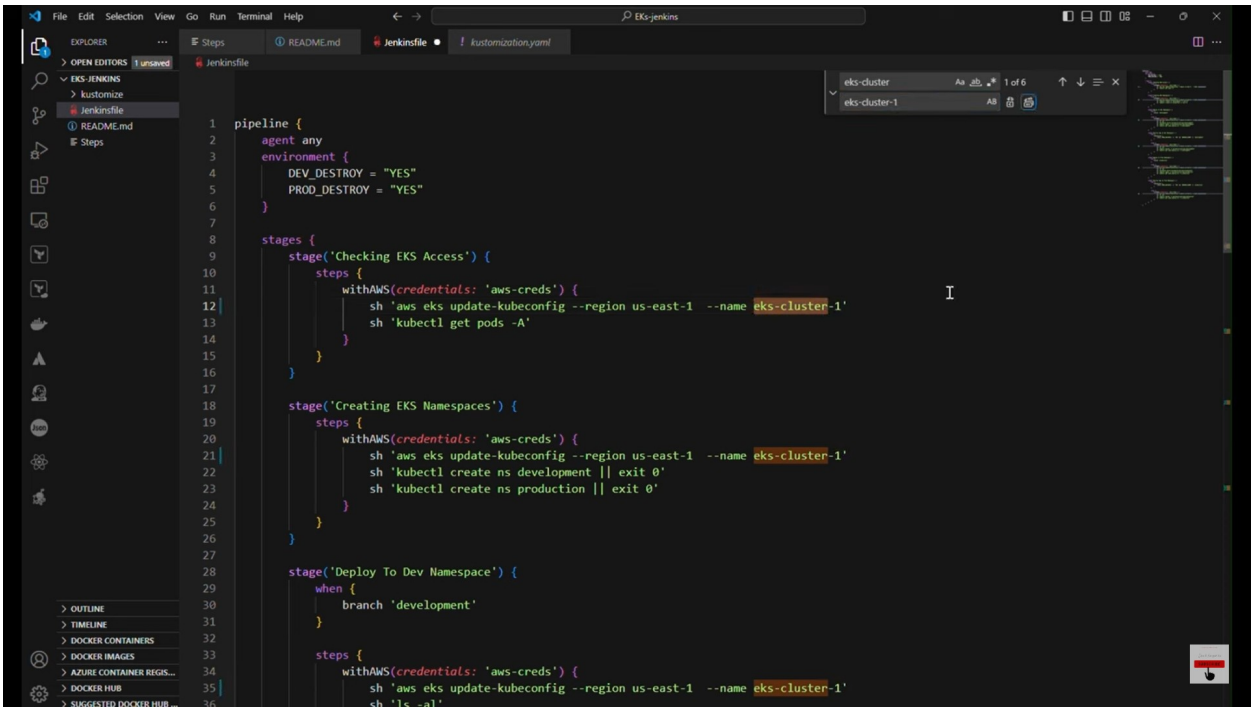


 # console Output – error

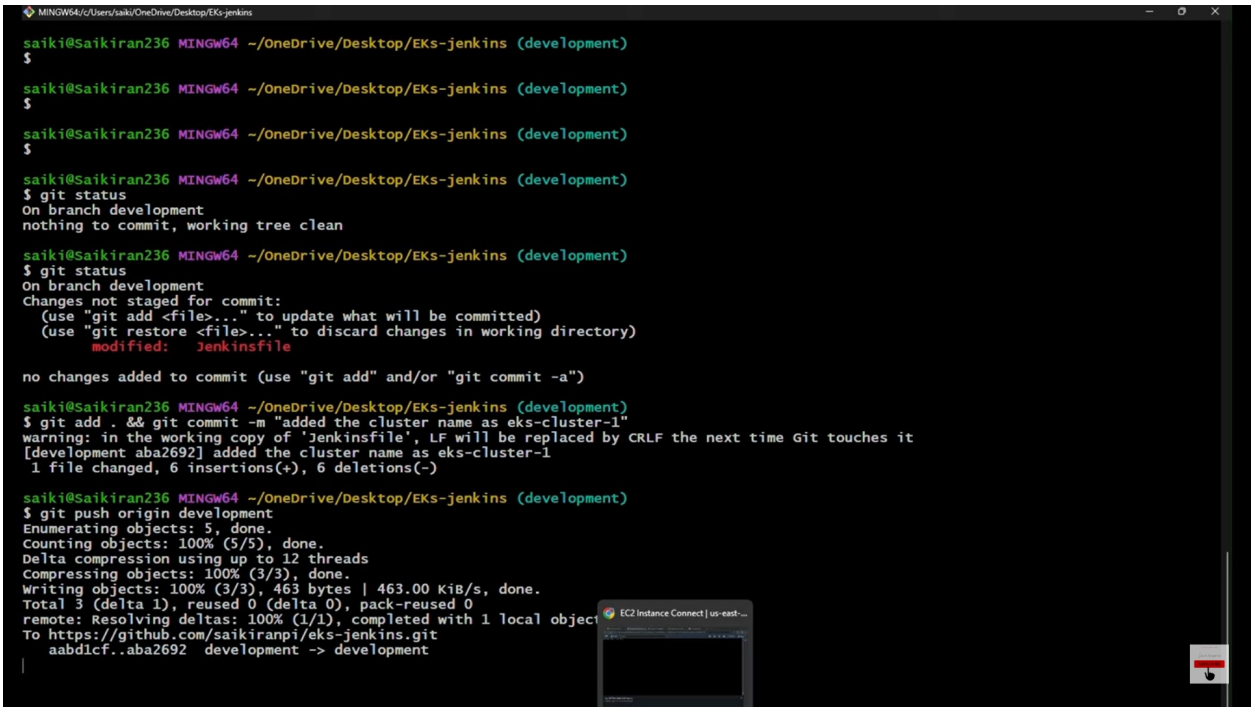


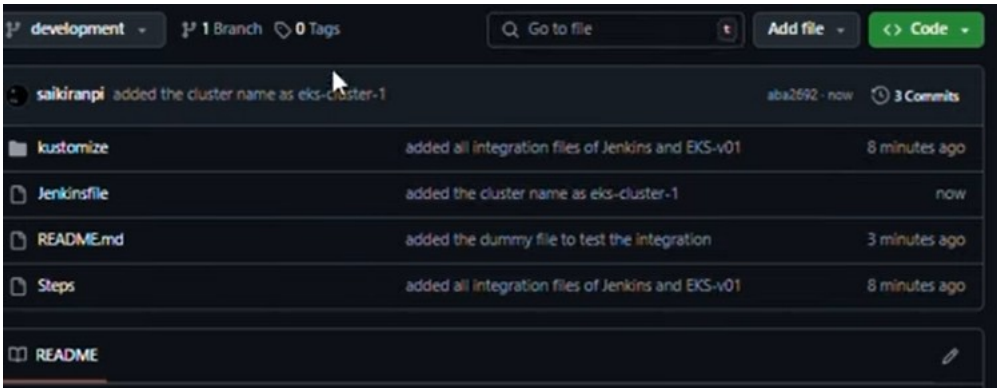


 # rename eks-cluster to eks-cluster1

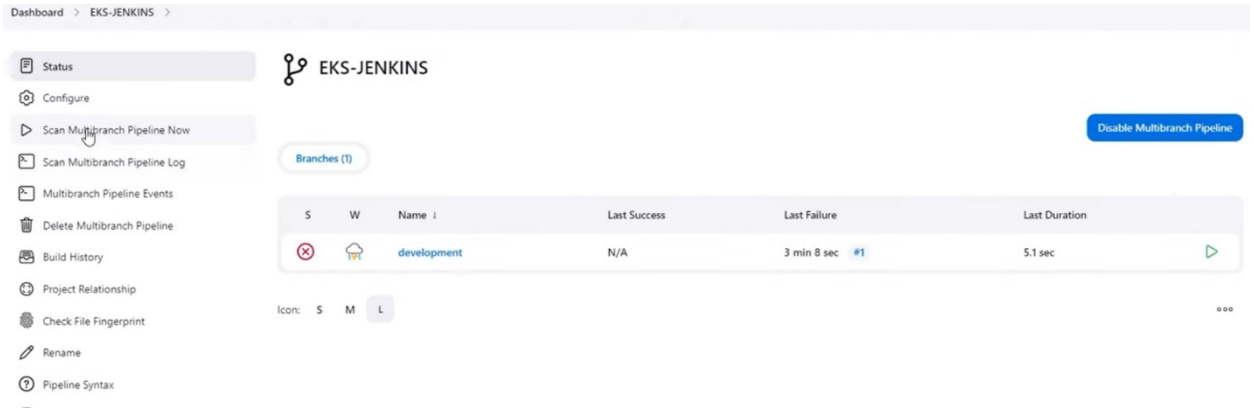


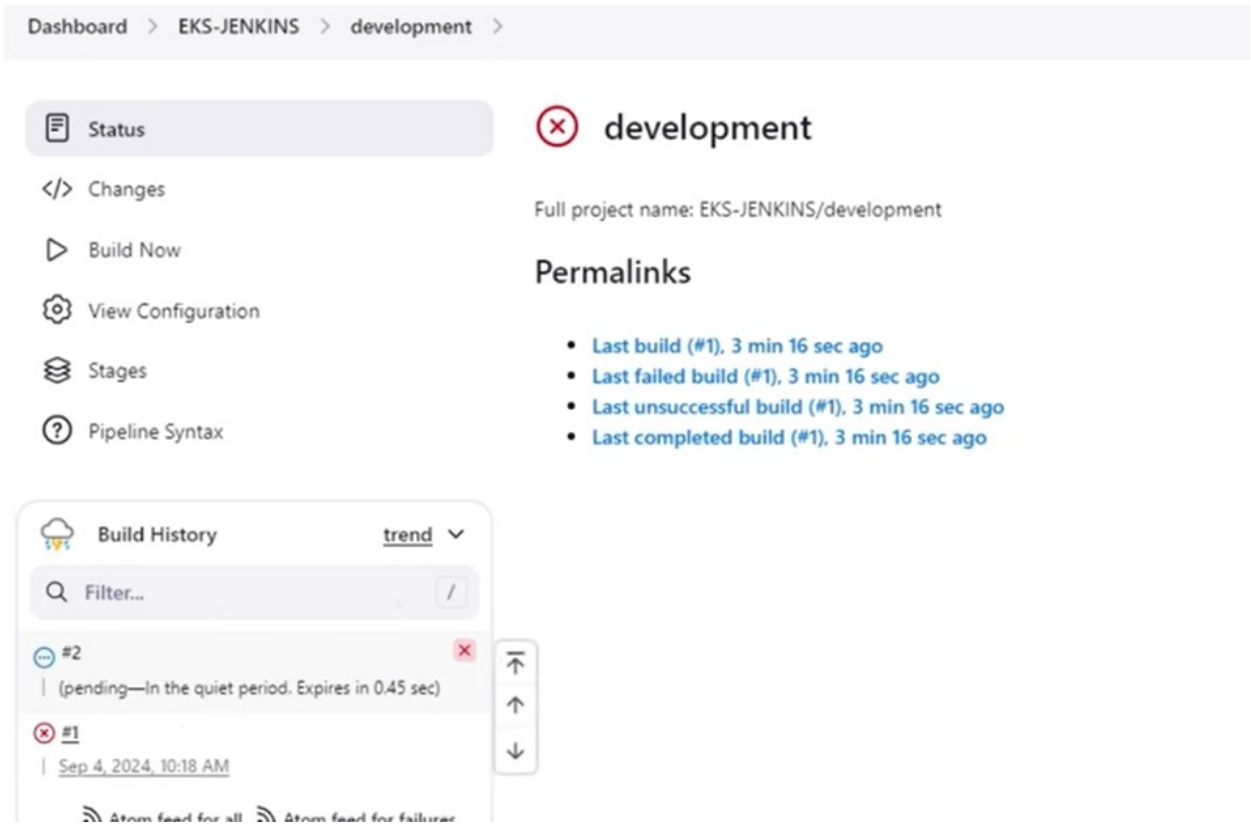

```
root@ip-10-0-9-146:/usr/local/bin# cd
root@ip-10-0-9-146:~# ku get ns
NAME              STATUS   AGE
default           Active   44m
development       Active   21m
kube-node-lease   Active   44m
kube-public       Active   44m
kube-system       Active   44m
production        Active   21m
root@ip-10-0-9-146:~# ku delete ns development production
namespace "development" deleted
namespace "production" deleted
^L
```



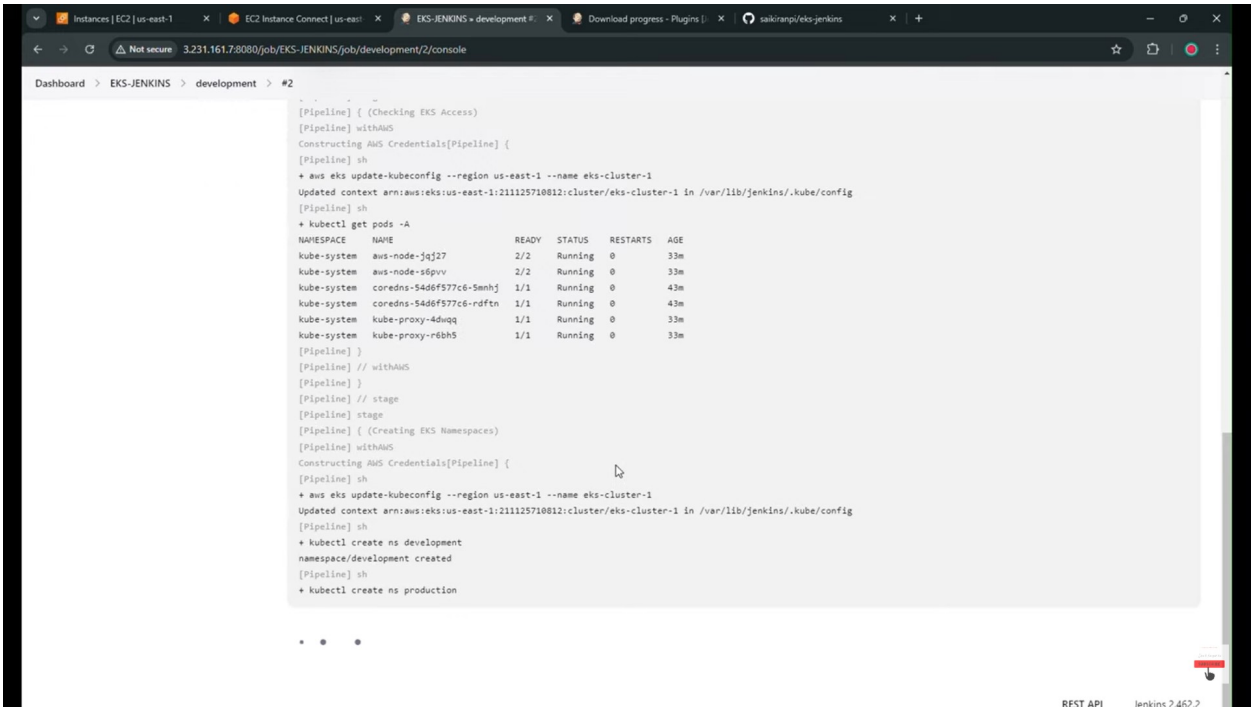


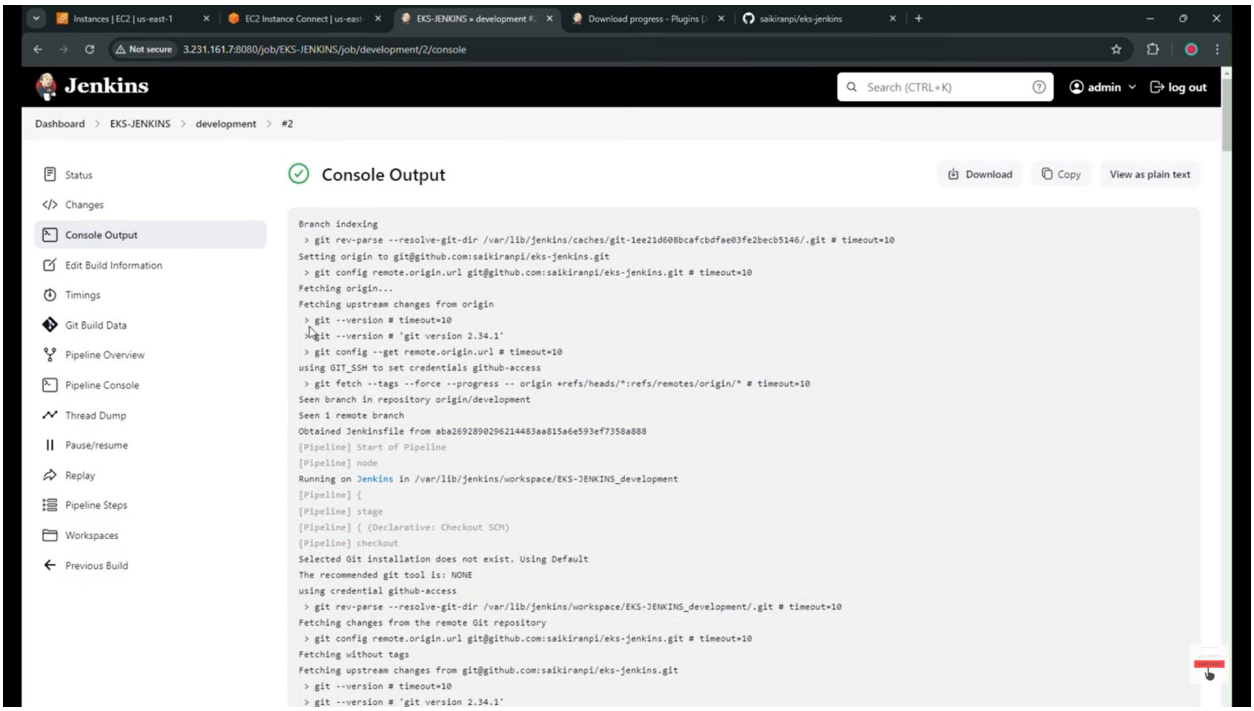
 # eks Jenkins .. scan multi-branch pipeline now




Dashboard > EKS-JENKINS > development >
Status
Changes
Build Now
View Configuration
Stages
Pipeline Syntax
development
Full project name: EKS-JENKINS/development
Permalinks
Last build (#1), 3 min 16 sec ago
Last failed build (#1), 3 min 16 sec ago
Last unsuccessful build (#1), 3 min 16 sec ago
Last completed build (#1), 3 min 16 sec ago
Build History
trend
Filter...
#2
(pending—In the quiet period. Expires in 0.45 sec)
#1
Sep 4, 2024, 10:18 AM

 # console O/p

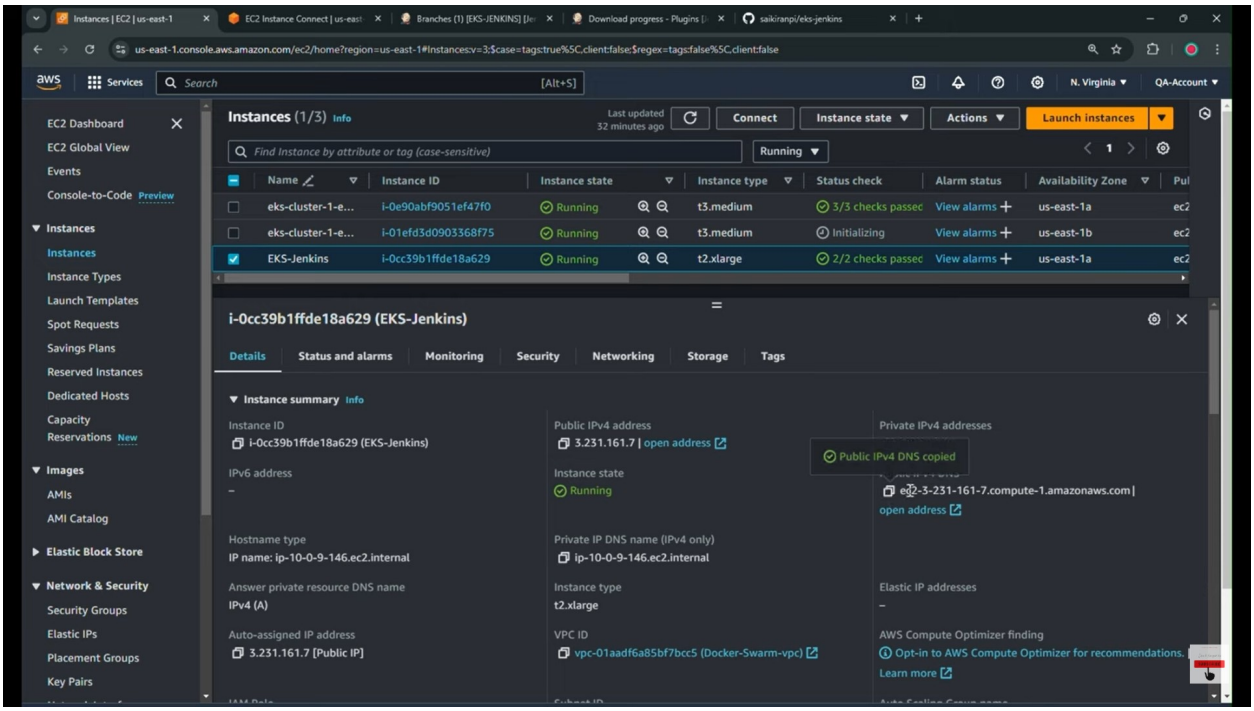

Not secure 3.231.161.7:8080/job/EKS-JENKINS/job/development/2/console
Dashboard > EKS-JENKINS > development > #2
[Pipeline] { (Checking EKS Access)
[Pipeline] withAWS
Constructing AWS Credentials[Pipeline] {
[Pipeline] sh
+ aws eks update-kubeconfig --region us-east-1 --name eks-cluster-1
Updated context arn:aws:eks:us-east-1:211125710812:cluster/eks-cluster-1 in /var/lib/jenkins/.kube/config
[Pipeline] sh
+ kubectl get pods -A
NAMESPACE NAME READY STATUS RESTARTS AGE
kube-system aws-node-jqj27 2/2 Running 0 33m
kube-system aws-node-s6pvv 2/2 Running 0 33m
kube-system coredns-54d6f577c6-5mnhj 1/1 Running 0 43m
kube-system coredns-54d6f577c6-rdftn 1/1 Running 0 43m
kube-system kube-proxy-4dwqq 1/1 Running 0 33m
kube-system kube-proxy-r6bh5 1/1 Running 0 33m
[Pipeline] }
[Pipeline] // withAWS
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Creating EKS Namespaces)
[Pipeline] withAWS
Constructing AWS Credentials[Pipeline] {
[Pipeline] sh
+ aws eks update-kubeconfig --region us-east-1 --name eks-cluster-1
Updated context arn:aws:eks:us-east-1:211125710812:cluster/eks-cluster-1 in /var/lib/jenkins/.kube/config
[Pipeline] sh
+ kubectl create ns development
namespace/development created
[Pipeline] sh
+ kubectl create ns production
REST API



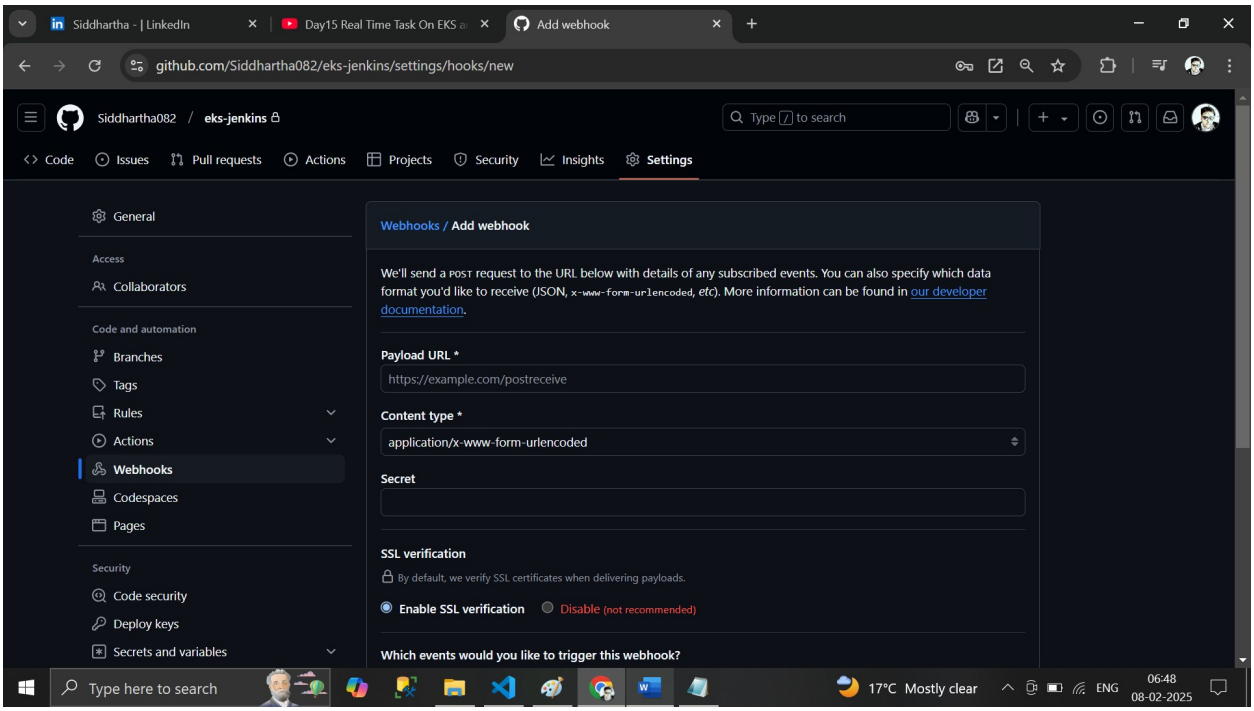
 # Final step automation on Webhook triggers

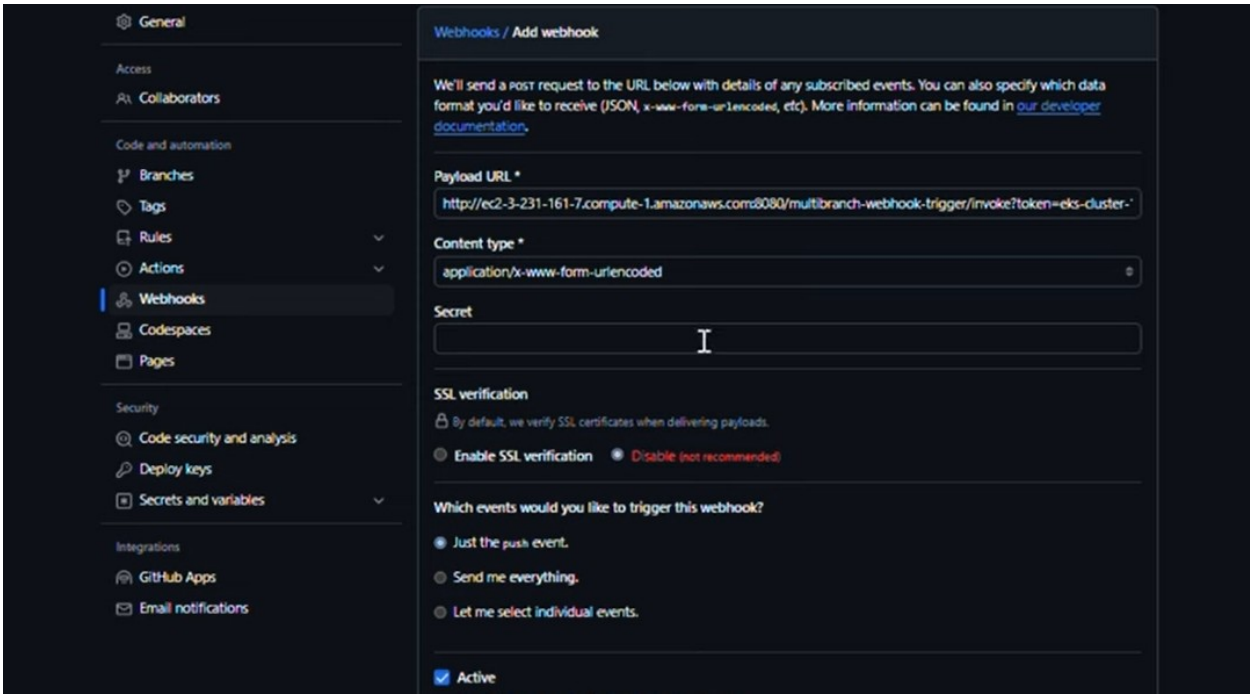
```
# Now i want auto commit not by just clicking on multibranch pipeline
multibranch jenkins > confgure > scan multibranch > Scan Multibranch Pipeline Triggers > scan by webhook > and carefully give the webhook accordingly.

http://ec2-3-231-161-7.compute-1.amazonaws.com:8080/multibranch-webhook-trigger/invoke?token=eks-cluster-1

ADD THE SAME ON GITHUB WEBHOOKS TOO

delete ""eksctl delete cluster --name eks-cluster-1""
```

 # go to instance copy DNS

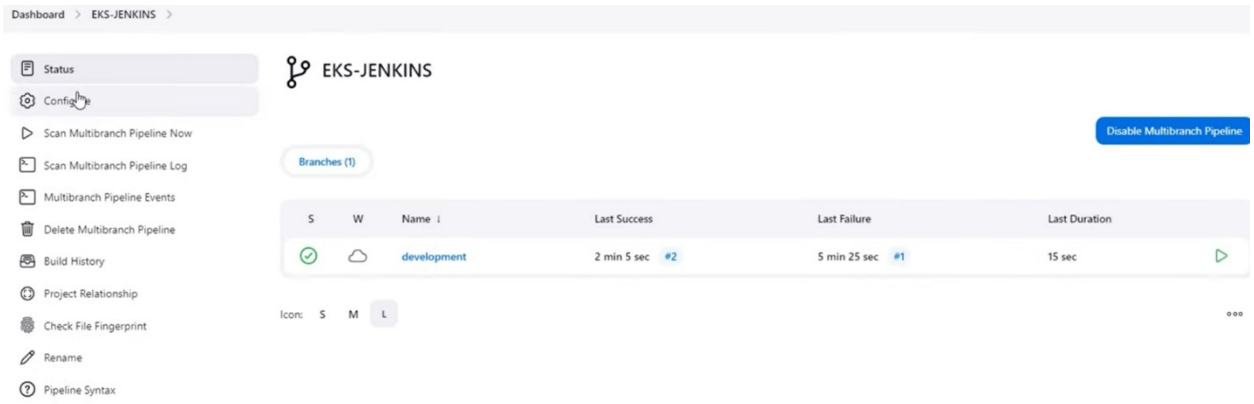


go to Github repo- settings- Webhooks- add

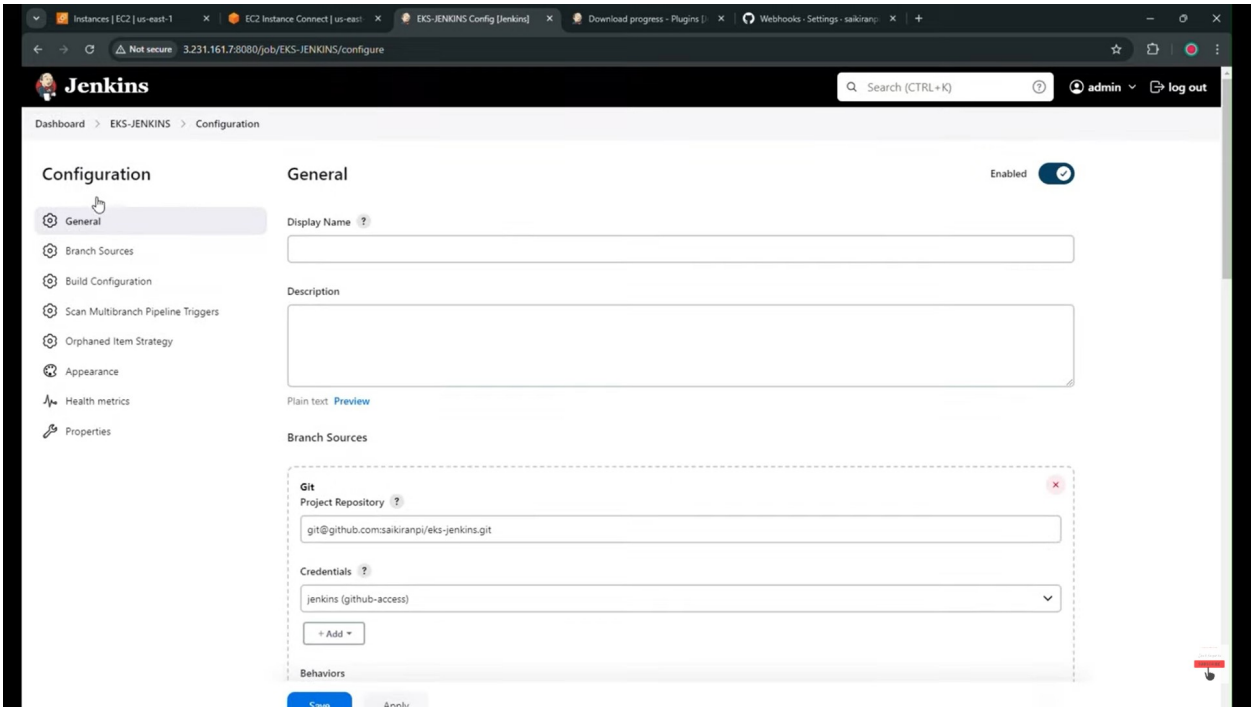




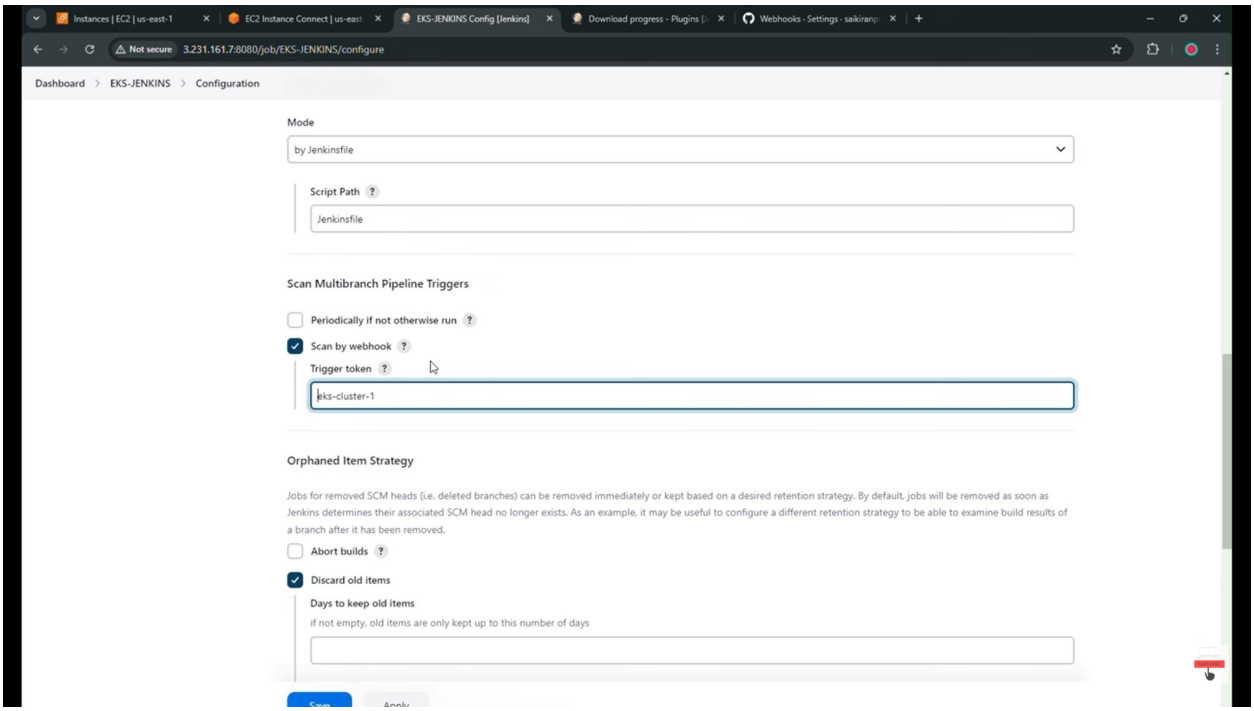

```
multibranch jenkins > confgure > scan multibranch > Scan Multibranch Pipeline Triggers > scan by webhook > and carefully give the webhook accordingly.
```

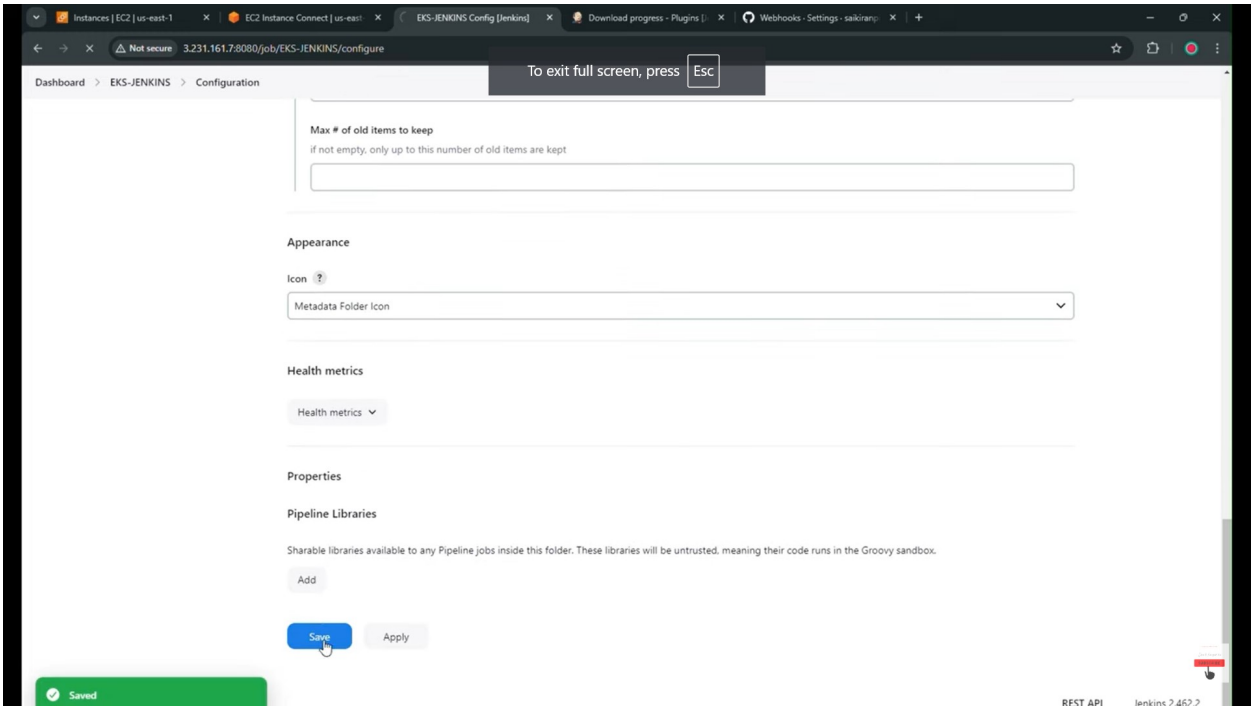


click configure

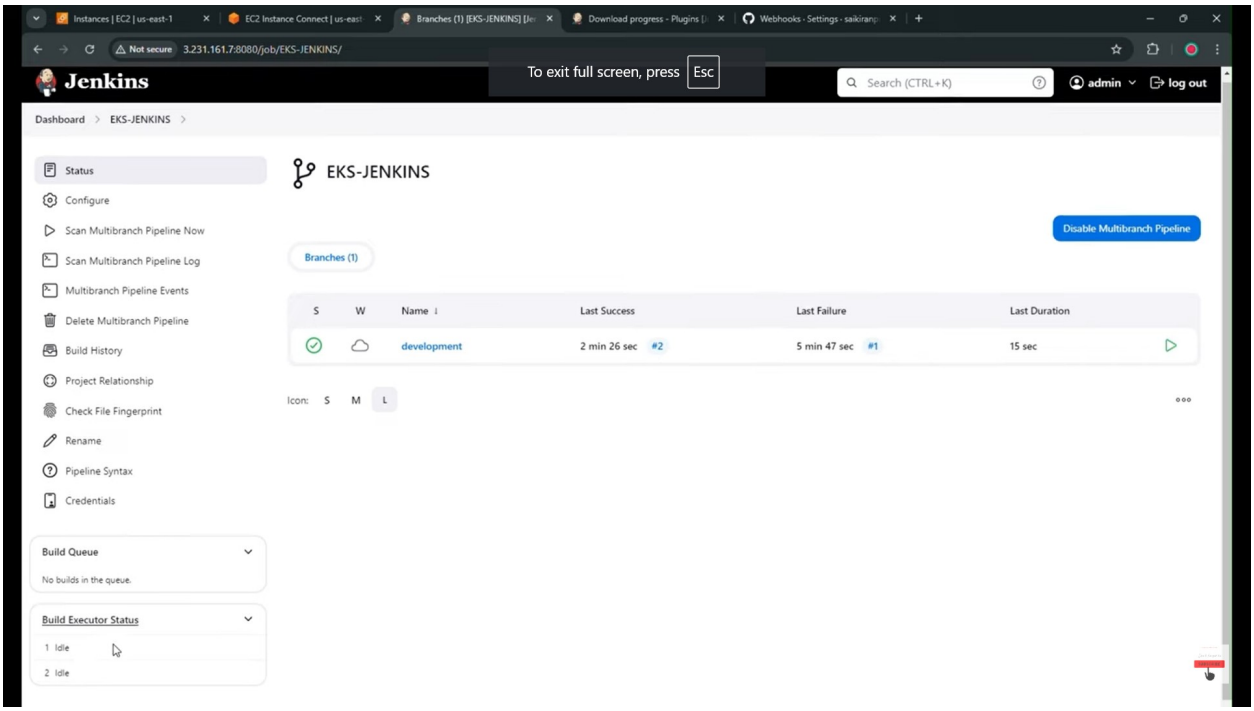


scan multibranch pipeline triggers





Automatically Job running



GitHub Webhook