

Task for today:

=====

1) Setup jenkins CICD pipeline using freestyle job using Docker containers using below code.

<https://github.com/betawins/hiring-app.git>

Stages:

1) Git Clone

2) Sonarqube Integration

3) Maven Compilation

4) Nexus Artifactory

5) Slack Notification

6) Deploy On tomcat.

Setup jenkins CICD pipeline using freestyle job

- To do that these task first I need server.
- Here my server is Available.
- Connect the server and install java, After that install Jenkins.

The screenshot shows the AWS EC2 Instances page. The left sidebar has a tree view with 'Instances' expanded, showing 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', and 'Capacity Reservations'. The main content area shows a table titled 'Instances (1/4)'. A single row is selected for an instance named 'Jenkins' with the ID 'i-087564c29427cc419'. The instance is listed as 'Running' on an 't2.xlarge' instance type. The table includes columns for Name, Instance ID, Instance state, Status check, and Alarm status. Below the table, a detailed view for 'i-087564c29427cc419 (Jenkins)' is shown. It includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under 'Details', there's an 'Instance summary' section with fields for Instance ID (i-087564c29427cc419), Public IPv4 address (54.241.156.9), Private IPv4 addresses (172.31.11.35), Instance state (Running), and Public IPv4 DNS (ec2-54-241-156-9.us-west-).

To install java CMD: sudo yum -y install java-17*

Click on the below URL here you find script to install Jenkins.

<https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos>

After perform the all Script you need to below commands.

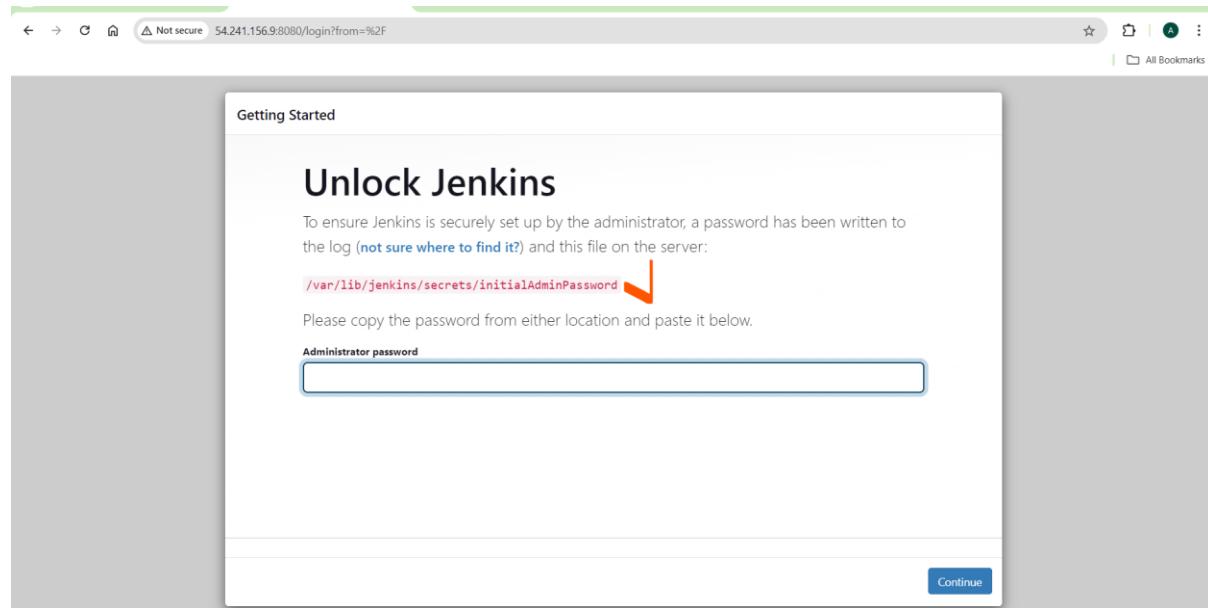
```
total download size: 89 M
installed size: 89 M
Is this ok [y/d/N]: y
Downloading packages:
jenkins-2.462.3-1.1.noarch.rpm
running transaction check
running transaction test
transaction test succeeded
running transaction
  Installing : jenkins-2.462.3-1.1.noarch
  Verifying  : jenkins-2.462.3-1.1.noarch
Installed:
  jenkins.noarch 0:2.462.3-1.1

Complete!
[ec2-user@ip-172-31-11-35 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-11-35 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
     Active: active (running) since Wed 2024-10-09 13:07:28 UTC; 9s ago
       Main PID: 7678 (java)
      Group: /system.slice/jenkins.service
             └─7678 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Oct 09 13:07:25 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: 28b91b53429481a9807f7851e46ad30
Oct 09 13:07:25 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 09 13:07:25 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: ****
Oct 09 13:07:25 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: ****
Oct 09 13:07:25 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: ****
Oct 09 13:07:28 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: 2024-10-09 13:07:28.038+0000 [id=39]      INFO  jenkins.InitReact...zation
Oct 09 13:07:28 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: 2024-10-09 13:07:28.052+0000 [id=25]      INFO  hudson.lifecycle...unning
Oct 09 13:07:28 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: Started Jenkins Continuous Integration Server.
Oct 09 13:07:29 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: 2024-10-09 13:07:29.125+0000 [id=55]      INFO  h.m.DownloadServ...taller
Oct 09 13:07:29 ip-172-31-11-35.us-west-1.compute.internal jenkins[7678]: 2024-10-09 13:07:29.126+0000 [id=55]      INFO  hudson.util.Retri...mpt #1
int: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-11-35 ~]$ |
```

Now go to the browser.

In browser paste <ip address:8080> --- you will be see the below interface.



Go to server there you will find the password of Jenkins.

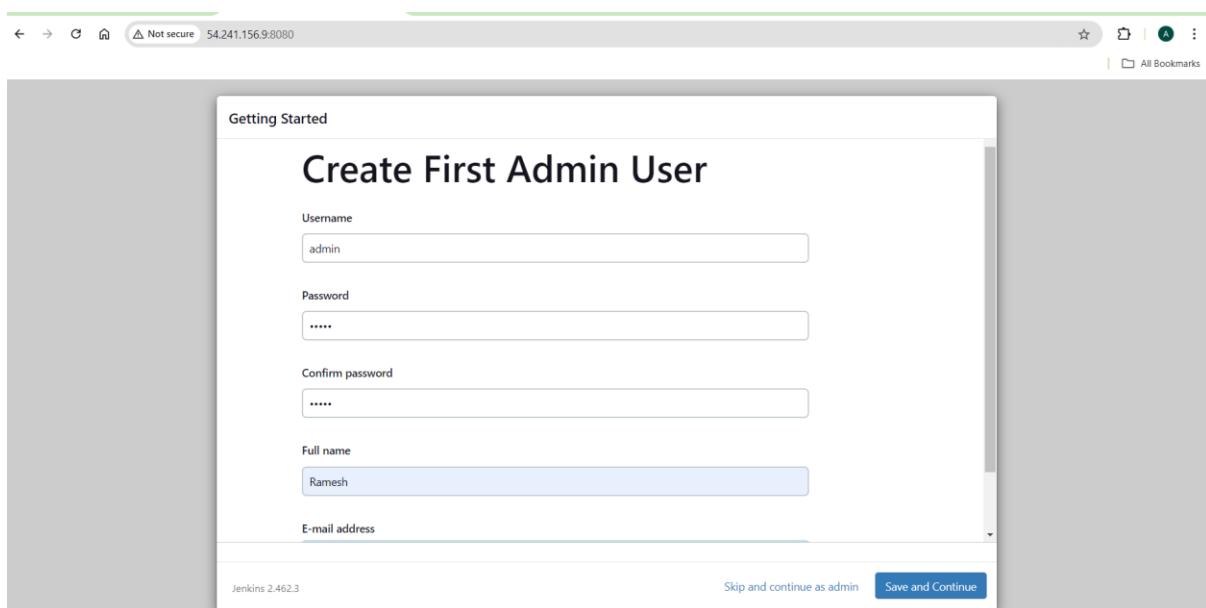
```
[ec2-user@ip-172-31-11-35 ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
28b91b53429481a9807f7851e46ad30
[ec2-user@ip-172-31-11-35 ~]$ |
```

After enter the password click on the install plugins.

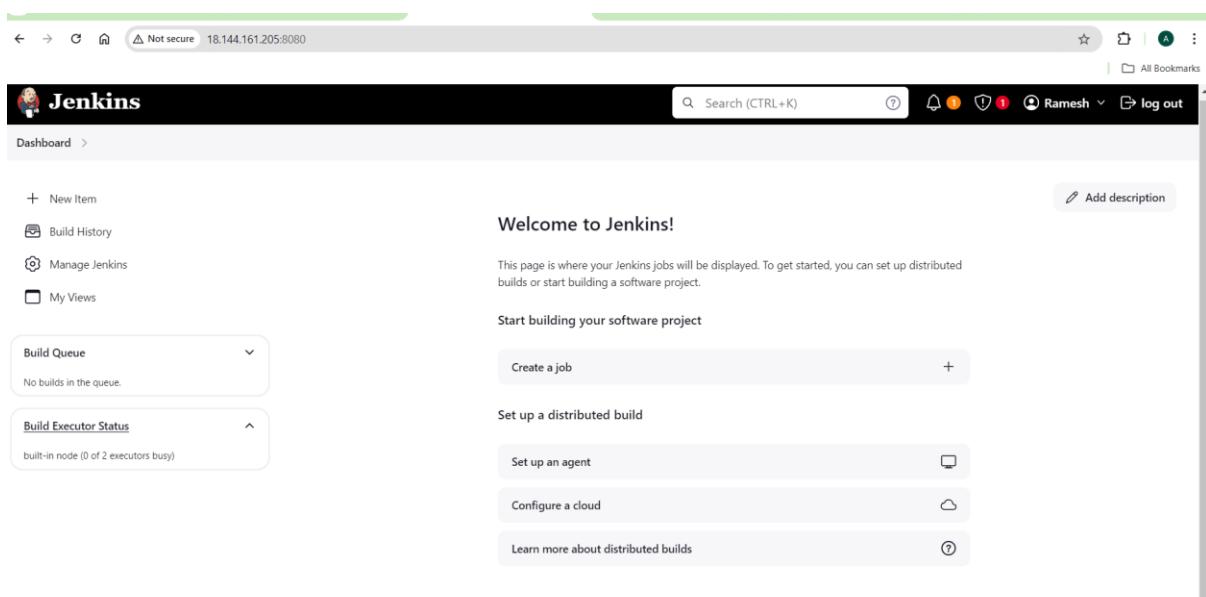


Here give user name and password.

Just click on save and continue.



Now you will see Jenkins Dashboard.



Jenkins installation and Login us successful.

Sonarqube Integration

- Now you need to install Sonarqube.
 - Using the Docker install Sonarqube.
 - First need to install docker, enable, start, and check the status.
 - CMD: sudo yum -y install docker --- to install docker

```
Installing : docker-25.0.6-1.amzn2.0.2.x86_64 [########################################]
Installing : docker-25.0.6-1.amzn2.0.2.x86_64 [########################################]
Installing : docker-25.0.6-1.amzn2.0.2.x86_64 [########################################]
Verifying : runc-1.1.13-1.amzn2.x86_64
Verifying : docker-25.0.6-1.amzn2.0.2.x86_64
Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64
Verifying : containerd-1.7.22-1.amzn2.0.2.x86_64
Verifying : libcgroup-0.41-21.amzn2.x86_64

nstalled:
docker.x86_64 0:25.0.6-1.amzn2.0.2

endency Installed:
containerd.x86_64 0:1.7.22-1.amzn2.0.2      libcgroup.x86_64 0:0.41-21.amzn2          pigz.x86_64 0:2.3.4-1.amzn2.0.1      runc.x86_64 0:1.1.13-1.amzn2

omplete!
ec2-user@ip-172-31-11-35 ~]$ sudo systemctl enable docker
reated symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
ec2-user@ip-172-31-11-35 ~]$ sudo systemctl start docker
ec2-user@ip-172-31-11-35 ~]$ sudo systemctl status docker
docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2024-10-09 17:02:48 UTC; 9s ago
     Docs: https://docs.docker.com
  Process: 5337 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
  Process: 5335 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
Main PID: 5340 (dockerd)
   Tasks: 10
  Memory: 29.3M
 CGroup: /system.slice/docker.service
         └─5340 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

ct 09 17:02:48 ip-172-31-11-35.us-west-1.compute.internal systemd[1]: Starting Docker Application Container Engine...
ct 09 17:02:48 ip-172-31-11-35.us-west-1.compute.internal dockerd[5340]: time="2024-10-09T17:02:48.494676132Z" level=info msg="Starting up"
ct 09 17:02:48 ip-172-31-11-35.us-west-1.compute.internal dockerd[5340]: time="2024-10-09T17:02:48.494697934Z" level=info msg="Loading containers: start."
ct 09 17:02:48 ip-172-31-11-35.us-west-1.compute.internal dockerd[5340]: time="2024-10-09T17:02:48.756623591Z" level=info msg="Loading containers: done."
ct 09 17:02:48 ip-172-31-11-35.us-west-1.compute.internal dockerd[5340]: time="2024-10-09T17:02:48.771543355Z" level=info msg="Docker daemon" commit=b0ba51f co...=25.0.6
ct 09 17:02:48 ip-172-31-11-35.us-west-1.compute.internal dockerd[5340]: time="2024-10-09T17:02:48.814617662Z" level=info msg="API listen on /run/docker.sock"
ct 09 17:02:48 ip-172-31-11-35.us-west-1.compute.internal systemd[1]: Started Docker Application Container Engine.
Int: Some lines were ellipsized, use -l to show in full.
ec2-user@ip-172-31-11-35 ~]$
```

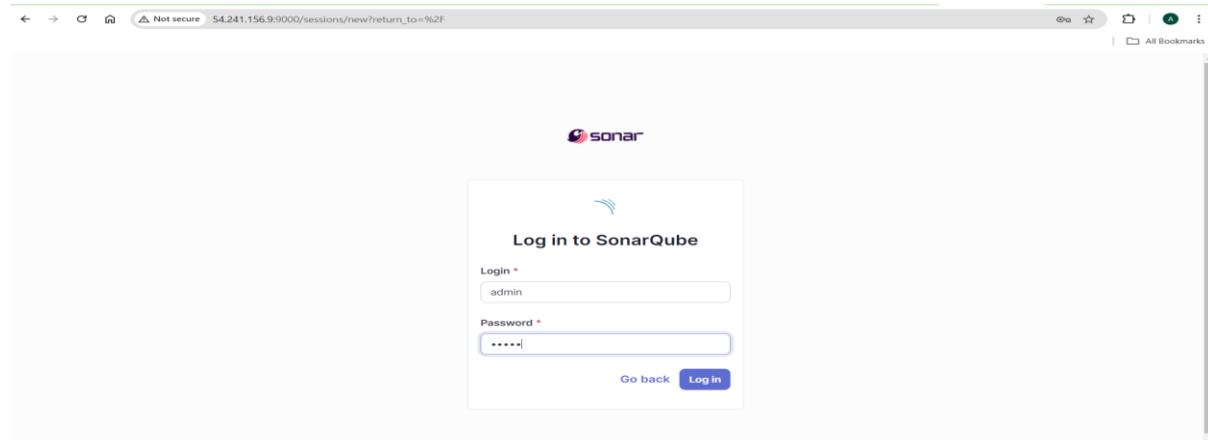
Now you need to install sonarqube using the docker.

CMD: sudo docker run -d --name sonarqube -p 9000:9000 sonarqube

```
[ec2-user@ip-172-31-11-35 ~]$ sudo docker run -d --name sonarqube -p 9000:9000 sonarqube
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
96a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4cb: Pull complete
1a5fd5c7e184: Pull complete
fbe03067fd0d: Pull complete
8f68213fa028: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:9187cdb52bb41c3261bead1793c0b92b1e66f957969f3984f6151ac41d20e24e
Status: Downloaded newer image for sonarqube:latest
954b447d99b4d8da867963facb2f8f529dc0e79941454a0492521e06cb3c33f
[ec2-user@ip-172-31-11-35 ~]$ |
```

Now you are able to access the sonarqube.

Login – admin, password – admin by default



Update new password

The screenshot shows a web browser window with the URL 54.241.156.9:9000/account/reset_password. The page title is "Update your password". There is a yellow warning box with the text "This account should not use the default password." Below it, there are three input fields: "Old Password *", "Password *", and "Confirm Password *". Each field has a green placeholder text "*****". At the bottom is a blue "Update" button.

After you will see the below Sonarqube Dashboard.

The screenshot shows a web browser window with the URL 54.241.156.9:9000/projects/create. The page title is "How do you want to create your project?". It features a heading "Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform." Below this, it says "First, you need to set up a DevOps platform configuration." There are five buttons for importing from "Import from Azure DevOps", "Import from Bitbucket Cloud", "Import from Bitbucket Server", "Import from GitHub", and "Import from GitLab", each with a "Setup" button. At the bottom, there is a "Create a local project" button.

Now we sonarqube integrate with Jenkins we have follow the below steps.

To generate token follow the below steps:

- Sonarqube dash board top right click on My account.
- There you will see the security option click on that.
- Give token name ---- what you want
- Type --- global.

The screenshot shows a web browser window with the URL 18.144.161.205:9000/account/security. The page title is "Administrator". The navigation bar includes "Profile", "Security" (which is highlighted with a red box), "Notifications", and "Projects". A red arrow points to the "Security" tab. Below the tabs, it says "If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network." The main section is titled "Generate Tokens" and contains a table with columns "Name", "Type", and "Expires in". A row shows a token named "Sonar_token" with type "Global", expiration on "November 9, 2024", and a "Revoke" button.

- Token is generated now go to the Jenkins dashboard.
- Now you need to install the plugin --- soanrqube scanner

The screenshot shows the Jenkins Plugins page. A search bar at the top right contains the text 'sonar'. Below it, a table lists three plugins:

- SonarQube Scanner 2.17.2**: Released 7 months 23 days ago. It is highlighted with a checkmark and an 'Install' button.
- Sonar Quality Gates 315.v1ff12b_e81a_3a_4**: Released 1 month 14 days ago. It is a library plugin for Sonar analysis.
- Quality Gates 2.5**: Released 8 years 4 months ago. It fails the build whenever Quality Gates criteria are not met.

A red callout box highlights the warning message for the Quality Gates plugin: "Warning: This plugin version may not be safe to use. Please review the following security notices: • Credentials transmitted in plain text".

- Install without restart.
- Next Go to the mange Jenkins and click on credentials.

The screenshot shows the Jenkins Manage Jenkins page under the 'System Configuration' tab. The 'Security' section is highlighted with a yellow arrow pointing to the 'Credentials' link. Other sections shown include System, Tools, Plugins, Nodes, Clouds, Appearance, and Credential Providers.

- Click on global and add credentials.

The screenshot shows the Jenkins Global credentials (unrestricted) page. A yellow arrow points to the '+ Add Credentials' button. The page displays a table with columns: ID, Name, Kind, and Description. A message at the bottom says: "This credential domain is empty. How about adding some credentials?"

- give the credentials and click on create.

New credentials

Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret:

ID: sonar

Description: sonar

Create

- Now Go to the manage Jenkins and system configuration there you need to pass the Name, URL and select the token After click on save.

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name: Sonar

Server URL: Default is http://localhost:9000
http://18.144.161.205:9000

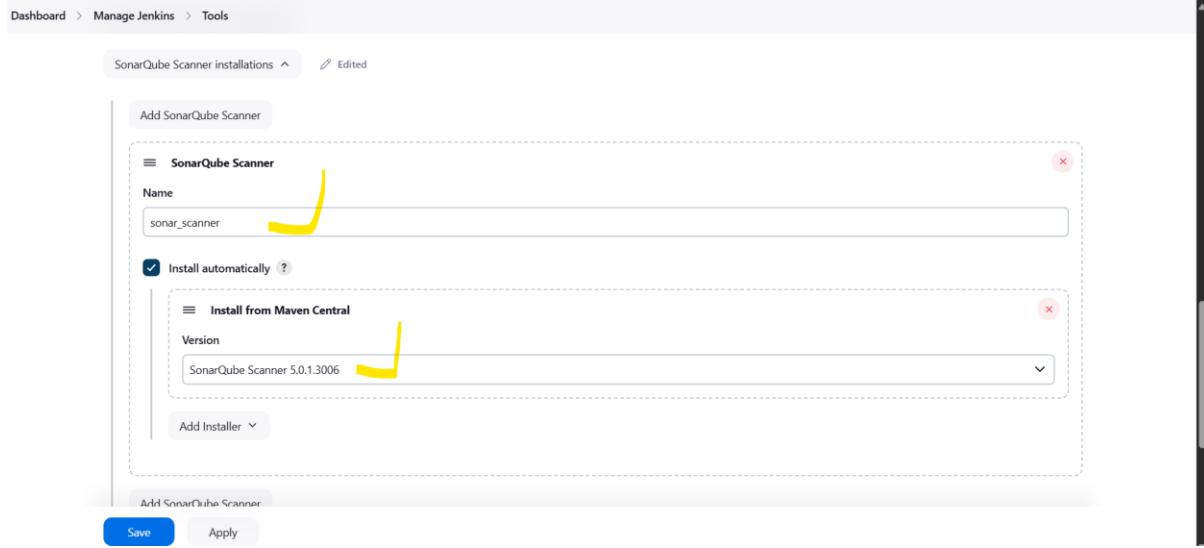
Server authentication token: SonarQube authentication token. Mandatory when anonymous access is disabled.
sonar

Save **Apply**

Now sonarqube integration is done.

After Go to the Jenkins Dashboard.

- Manage Jenkins > Tools
- Sonarqube Scanner installations ---- give name and select install automatically.
- Here we can install manually we facing the issue. Like version issue's.



- Just click on save.
- The Sonarqube and sonar_scanner Configurations are done.

Maven Compilation

- Now go to the maven step.
- To build the code you need to install maven.
- CMD: sudo yum -y install maven

```
[ec2-user@ip-172-31-11-35 ~]$ sudo yum -y install maven
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package maven.noarch 0:3.0.5-17.amzn2 will be installed
--> Processing Dependency: aether-api for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: aether-connector-wagon for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: aether-impl for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: aether-spi for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: aether-util for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: aopalliance for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: apache-commons-cli for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: apache-commons-codec for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: apache-commons-logging for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: atfj for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: cglib for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: google-guice for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: guava for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: httpcomponents-client for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: httpcomponents-core for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: maven-wagon for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn/commons-cli/commons-cli for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.apache.maven.wagon:wagon-provider-api) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.codehausplexus:plexus-classworlds) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.codehausplexus:plexus-component-annotations) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.codehausplexus:plexus-container-default) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.codehausplexus:plexus-interpolation) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.codehausplexus:plexus-utils) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.aether:aether-api) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.aether:aether-impl) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.aether:aether-spi) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.aether:aether-util) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatypeplexus:plexus-cipher) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatypeplexus:sec-dispatcher) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: mvn(org.sonatype.sisu:sisu-inject-plexus) for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-cipher for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-containers-component-annotations for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-interpolation for package: maven-3.0.5-17.amzn2.noarch
--> Processing Dependency: plexus-sec-dispatcher for package: maven-3.0.5-17.amzn2.noarch
```

After that check the all versions like maven and java.

Setting environment variables in your .bash_profile allows you to configure application settings and customize behavior globally. Modifying the PATH makes it easier to run commands without specifying their full paths, enhancing productivity.

- I went to the .bash_profile edited the file.

```
# .bash_profile

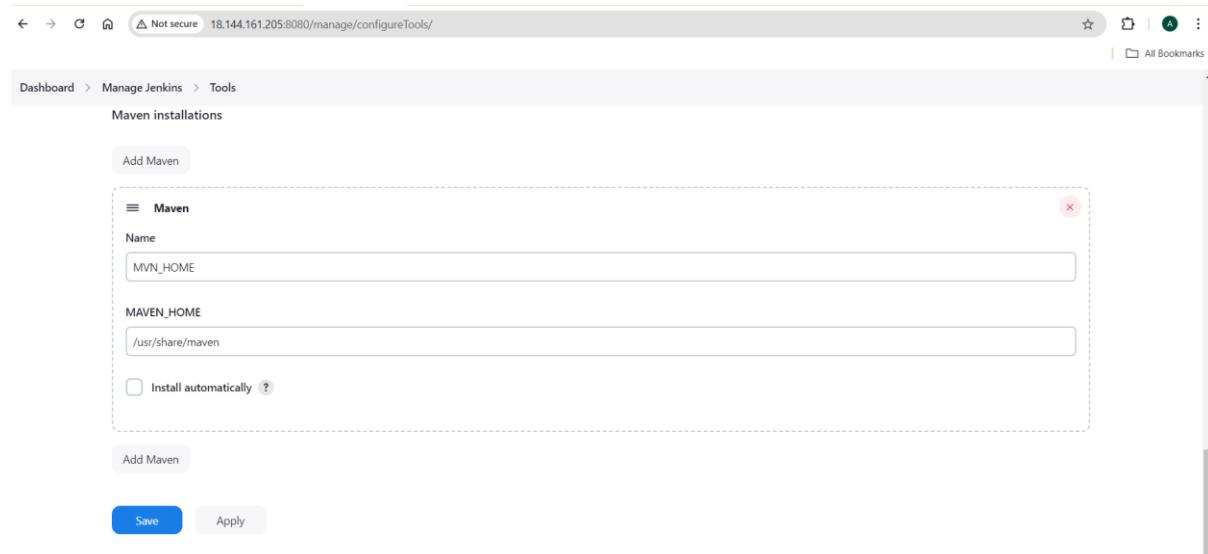
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User-specific environment and startup programs
MVN_HOME=/usr/share/maven
JAVA_HOME=/usr/lib/jvm/java-17-amazon-corretto.x86_64

# Append to PATH
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin:$MVN_HOME/bin:$HOME/.local/bin

# Export PATH
export PATH
|
```

- Save these file.
- Now go to Jenkins GUI.
- Go to manage Jenkins, Tools.
- Give name and path. After click on save.



- The maven configurations also Done.

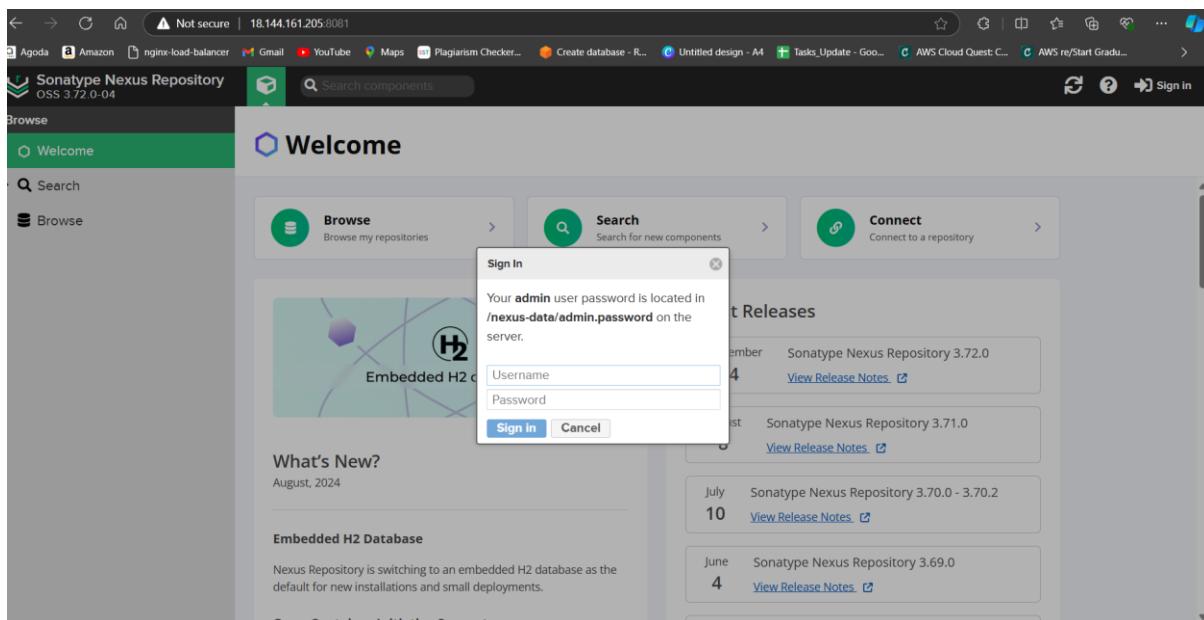
Nexus Artifactory

- Using the Docker I am installing Nexus.

CMD: docker run -d --name nexus -p 8081:8081 sonatype/nexus3

```
[ec2-user@ip-172-31-11-35 ~]$ docker run -d --name nexus -p 8081:8081 sonatype/nexus3
715e840932b617eb33f88dc15ff36494b5257183acafdd4257e58c5e4b9fe826
[ec2-user@ip-172-31-11-35 ~]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
715e840932b6 sonatype/nexus3 "/opt/sonatype/nexus..." About a minute ago Up About a minute 0.0.0.0:8081->8081/tcp, :::8081->8081/tcp nexus
f2bcad8b2b3 sonarqube "/opt/sonarqube/dock..." 14 minutes ago Up 14 minutes 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp busy_hamilton
```

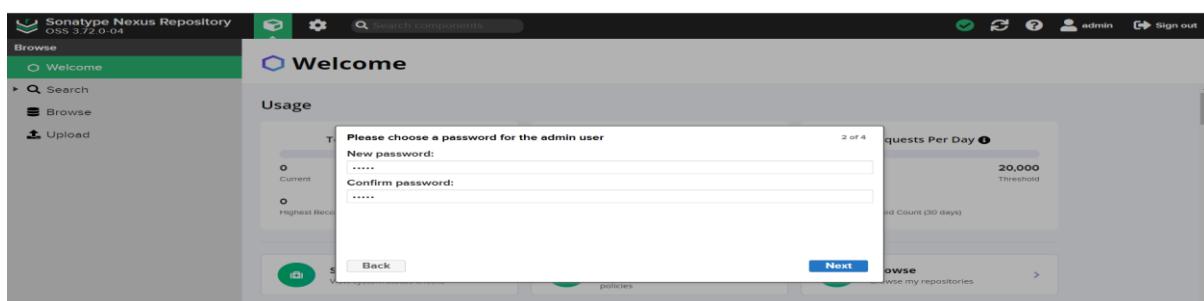
- Now try to access the Nexus with <ip:8081> --- you see the below interface.
- Default user is Admin.



- If you want password go to server run below Command.
- CMD: docker exec <container Id> cat /nexus-data/admin.password

```
[ec2-user@ip-172-31-11-35 ~]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
715e840932b6 sonatype/nexus3 "/opt/sonatype/nexus..." 6 hours ago Up 5 hours 0.0.0.0:8081->8081/tcp, :::8081->8081/tcp nexus
f2bcad8b2b3 sonarqube "/opt/sonarqube/dock..." 6 hours ago Up 5 hours 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp busy_hamilton
[ec2-user@ip-172-31-11-35 ~]$ docker exec 715e84 cat /nexus-data/admin.password
433500ab-e141-4cf6-8f15-1652a0a4d10e[ec2-user@ip-172-31-11-35 ~]$ |
```

- After that you need to setup your user and password.



Just click on Next.

- disable anonymous access and click on nexus.
- Now you need to create repo in nexus And repositories.

Click on settings

Sonatype Nexus Repository OSS 3.72.0-04

Administration

Repository

Repositories

Blob Stores

Proprietary Repositories

Content Selectors

Cleanup Policies

Routing Rules

Security

Privileges

Roles

Users

Anonymous Access

Repository

Blob Stores

Cleanup Policies

Proprietary Repositories

Content Selectors

Repositories

Routing Rules

Click on create repository.

Sonatype Nexus Repository OSS 3.72.0-04

Administration

Repository

Repositories

Blob Stores

Proprietary Repositories

Content Selectors

Cleanup Policies

Routing Rules

Security

Privileges

Roles

Users

Anonymous Access

Repositories

Name

Type

Format

Blob Store

Status

URL ↑

Health check

Firewall Re...

Name	Type	Format	Blob Store	Status	URL ↑	Health check	Firewall Re...
maven-central	proxy	maven2	default	Online - Ready to Conn...	[copy]	Analyze	[]
maven-public	group	maven2	default	Online	[copy]	[]	[]
maven-releases	hosted	maven2	default	Online	[copy]	[]	[]
maven-snapshots	hosted	maven2	default	Online	[copy]	[]	[]
nuget-group	group	nuget	default	Online	[copy]	[]	[]
nuget-hosted	hosted	nuget	default	Online	[copy]	[]	[]
nuget.org-proxy	proxy	nuget	default	Online - Ready to Conn...	[copy]	Analyze	[]

After click on maven(hosted).

Here you need to fill repo Name, version policy—release, layout policy—permissive.

Sonatype Nexus Repository OSS 3.72.0-04

Administration

Repository

Repositories

Blob Stores

Proprietary Repositories

Content Selectors

Cleanup Policies

Routing Rules

Security

Privileges

Roles

Users

Anonymous Access

LDAP

Realms

Repositories / hiring_app

Delete repository

Rebuild index

Settings

Name: hiring_app

Format: maven2

Type: hosted

URL: http://54.183.255.255:8081/repository/hiring_app/

Online: If checked, the repository accepts incoming requests

Maven 2

Version policy: Release

Layout policy: Validate that all paths are maven artifact or metadata paths

Content Disposition: Add Content-Disposition header as 'Attachment' to disable some content from being inline in a browser

Storage

Blob store:

After that go to Deployment policy – redeploy and click on create repository.

Sonatype Nexus Repository OSS 3.72.0-04

Administration

Repository

Repositories

Hosted

Deployment policy:

Controls if deployments of and updates to artifacts are allowed

Allow redeploy

Proprietary Components:

Components in this repository count as proprietary for namespace conflict attacks (requires Sonatype Nexus Firewall)

Now your repo is Available.

Sonatype Nexus Repository OSS 3.72.0-04

Administration

Repository

Repositories

Name	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...
nuget-hosted	hosted	nuget	default	Online			
nuget.org-proxy	proxy	nuget	default	Online - Ready to Con...			
nuget-group	group	nuget	default	Online			
maven-releases	hosted	maven2	default	Online			
maven-snapshots	hosted	maven2	default	Online			
maven-central	proxy	maven2	default	Online - Ready to Con...			
maven-public	group	maven2	default	Online			
Simple_customer_app	hosted	maven2	default	Online			
hiring_app	hosted	maven2	default	Online			
V-profile	hosted	maven2	default	Online			

Now go to the Jenkins Dashboards and manage Jenkins.

- GO to Available Jenkins, search for Nexus Artifacts Uploader.
- Select check box and click on install.

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Search (CTRL+K)

Install

Nexus Artifact Uploader 2.14

Artifact Uploaders

This plugin to upload the artifact to Nexus Repository.

This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.

Released 1 yr 10 mo ago

Maven Artifact ChoiceListProvider (Nexus) 1.17

Maven Build Parameters

This Plugin adds a new ChoiceListProvider for the "Extensible Choice Plugin" which is able to read Artifact information from a Nexus repository.

11 mo ago

artifact-promotion 0.5.2

Pipeline Build Wrappers External Site/Tool Integrations Artifact Uploaders Maven Other Post-Build Actions

This is a simple plugin to promote artifacts. The promotion is done on the repository server(s) by moving the artifact from a 'staging' repository into a 'release' repository. Currently, only Sonatype Nexus (Open Source) is supported.

This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.

5 yr 3 mo ago

The Nexus Configuration Is Done.

Slack Configuration.

Now we have to setup slack configuration.

- First install the slack notification plugin.

The screenshot shows the Jenkins Plugins page. On the left, there's a sidebar with links for Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. The main area has a search bar at the top with the query 'slack'. Below it, a table lists several plugins:

Install	Name	Released
<input checked="" type="checkbox"/>	Slack Notification 751.v2e44153c8fe1	2 days 15 hr ago
<input type="checkbox"/>	Global Slack Notifier 1.5	5 yr 7 mo ago
<input type="checkbox"/>	Build Notifications 1.5.0	7 yr 0 mo ago
<input type="checkbox"/>	Slack Upload 1.7	

A warning message is displayed for the 'Build Notifications' plugin: "Warning: This plugin version may not be safe to use. Please review the following security notices:
• Tokens stored in plain text".

After installing the Slack plugin I am going to slack.

- Go to the slack click on Add Apps and search for the Jenkins ci.

The screenshot shows the Slack App Marketplace. A search bar at the top contains the query 'jenkins ci'. Below it, a list of available apps includes 'Jenkins CI' and 'BuildPulse'. The 'Jenkins CI' app card is highlighted with a yellow arrow pointing to its 'Add' button.

You have to click on Add Jenkins CI.

After that you see the below interface and just click on the Add to slack.

The screenshot shows the Jenkins CI app details page on the Slack Marketplace. The app icon features a cartoon Jenkins head holding a coffee cup. The title 'Jenkins CI' is displayed above a description: 'Jenkins CI is a customizable continuous integration server with over 600 plugins, allowing you to configure it to meet your needs. This integration will post build notifications to a channel in Slack.' A large green 'Add to Slack' button is prominently featured. At the bottom, there are links for 'Learn more & Support', 'Privacy policy', 'Terms', and 'Categories'.

- Select your channel and click on Add Jenkins CI integration.

The screenshot shows the Jenkins CI app page in the Slack Marketplace. At the top, there's a search bar and navigation links for 'Browse', 'Manage', and 'Build'. The main content area has a heading 'Jenkins CI' with a brief description: 'An open source continuous integration server.' Below this, it says 'Jenkins CI is a customizable continuous integration server with over 600 plugins, allowing you to configure it to meet your needs.' and 'This integration will post build notifications to a channel in Slack.' There's a section titled 'Post to Channel' with a dropdown menu set to '# notifications' and a button 'Add Jenkins CI integration' at the bottom. A yellow arrow points to this button.

The below you can see the step by step instructions.

The screenshot shows the 'Setup Instructions' for the Jenkins CI app. It starts with 'Step 1' which says 'In your Jenkins dashboard, click on Manage Jenkins from the left navigation.' followed by a screenshot of the Jenkins dashboard. Then it goes to 'Step 2' which says 'Click on Manage Plugins and search for Slack Notification in the Available tab. Click the checkbox and install the plugin.' followed by a screenshot of the Manage Plugins screen. A yellow arrow points from the Jenkins dashboard in Step 1 to the Manage Plugins screen in Step 2.

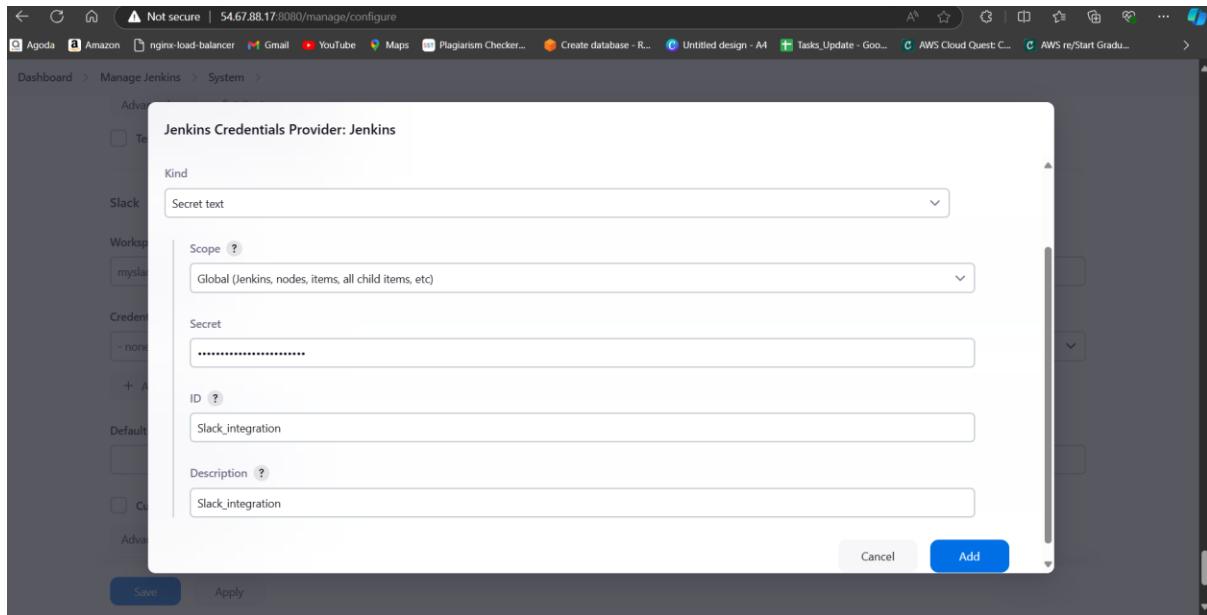
In these 3rd step you can see the domain name and token name.

These credentials we have to add in Jenkins.

The screenshot shows the configuration steps for Jenkins. 'Step 3' shows the 'Global Slack Notifier Settings' in Jenkins, with fields for 'Team Subdomain' (set to 'jenkins-slack-plugin') and 'Integration Token Credential ID' (set to 'some text (bot user slack token)'). A note says 'Exposing your Integration Token is a security risk. Please use the Integration Token Credential ID' and 'some text (bot user slack token)' is highlighted. 'Step 4' shows the configuration for individual projects, stating 'For each Project that you would like receive notifications for, choose Configure from the project's menu.' A yellow arrow points from the Jenkins configuration in Step 3 to the project configuration in Step 4.

Now go to the manage Jenkins > system > global credentials

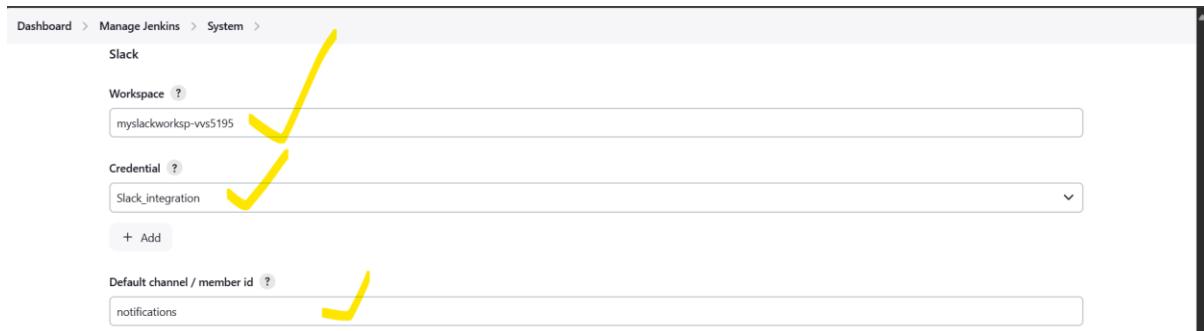
- There need to select → Kind – secret text
- There give secret – what we in slack
- Give ID --- give any name what you want and description also.



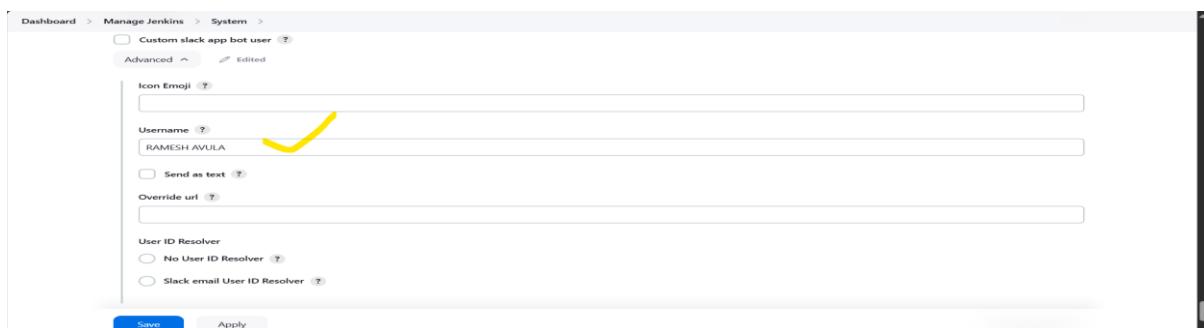
After that give user name and click on save.

Now go to the Jenkins Dashboard, manage Jenkins, system global configurations.

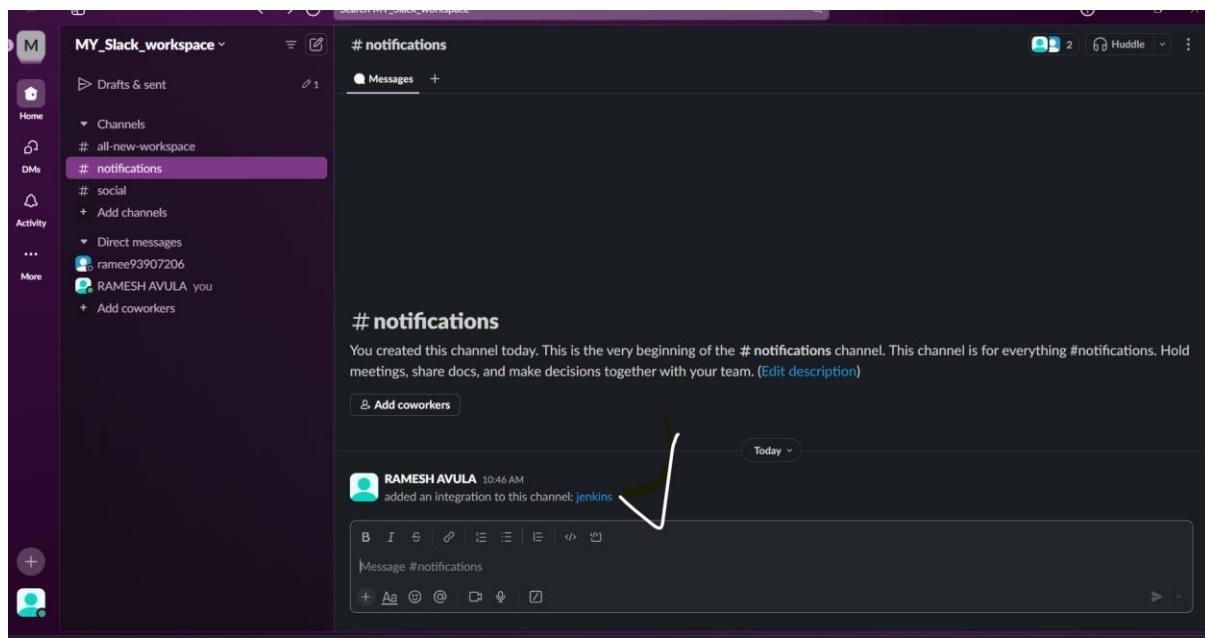
- There under the Slack see workspace, credential and channel.
- There you need to give workspace name, credential what we give in global credentials and give channel name.



Give user name also.



After that you will see in the slack will be Jenkins integrated your slack channel.



The Last Stage is tomcat configuration.

To download the tomcat click on the link --- [Apache Tomcat® - Apache Tomcat 9 Software Downloads](https://tomcat.apache.org/download-90.cgi)

There you will see tar.gz(ppg, sha512) --- copy the link

The screenshot shows the Apache Tomcat 9.0.96 download page. The left sidebar contains links for Home, Taglibs, Maven Plugin, Download (Tomcat 11, 10, 9, Migration Tool, Connectors, Native, Archives), Documentation (Tomcat 11.0, 10.1, 9.0, Upgrading, Connectors, Native 2, Native 1.3, Wiki, Migration Guide, Presentations, Specifications), Problems (Security Reports, Find help, FAQ, Mailing Lists, Bug Database, IRC), and Finance headline (India reported 1...). The main content area has sections for Release Integrity (instructions about OpenPGP signatures), Mirrors (current mirror https://dlcdn.apache.org/), and 9.0.96 (instructions to read README). The Binary Distributions section lists various file types for download, including Core (zip (ppg, sha512), tar.gz (ppg, sha512) [highlighted by a yellow arrow], 32-bit Windows zip (ppg, sha512), 64-bit Windows zip (ppg, sha512), 32-bit/64-bit Windows Service Installer (ppg, sha512)), Full documentation (tar.gz (ppg, sha512)), Deployer (zip (ppg, sha512), tar.gz (ppg, sha512)), and Embedded (tar.gz (ppg, sha512), zip (ppg, sha512)). The Source Code Distributions section is also present. The bottom of the screen shows a Windows taskbar with various icons.

Now go to the server there you need to install tomacat.

CMD: wget<link> ---- to download form the internet

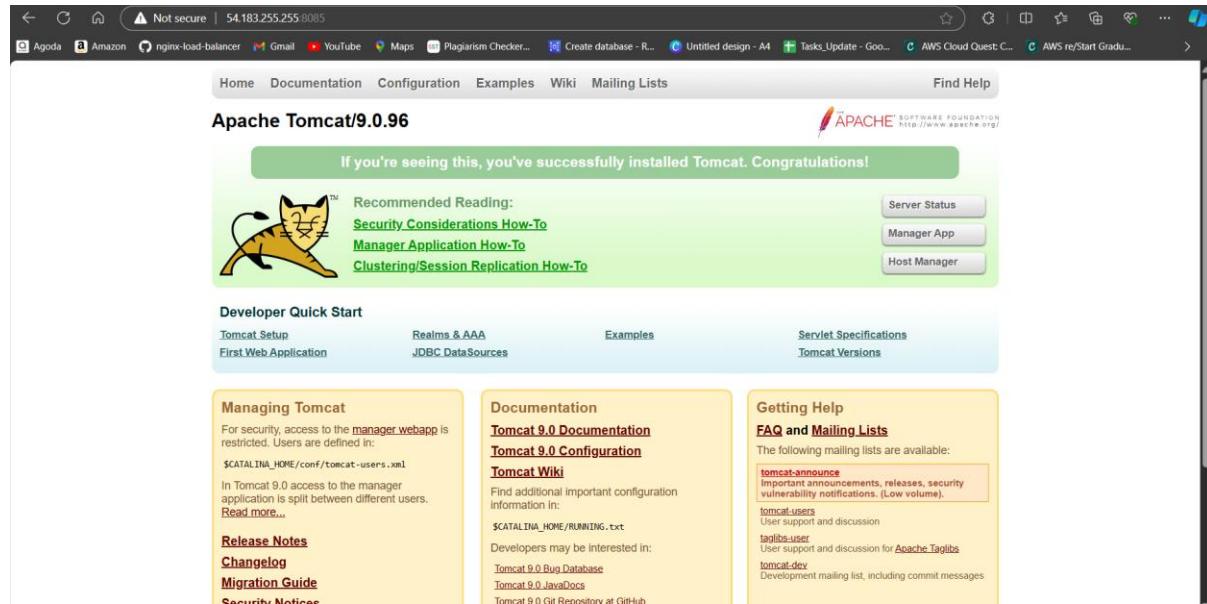
CMD: tar -xvf <link> -- to extract the tar.file

Now you have to start the tomcat server. CMD-- ./startup.sh

```
[root@ip-172-31-11-35 opt]# ls
apache-tomcat-9.0.96 apache-tomcat-9.0.96.tar.gz aws containerd rh sonar_scanner
[root@ip-172-31-11-35 opt]# cd apache-tomcat-9.0.96/
[root@ip-172-31-11-35 apache-tomcat-9.0.96]# ls
bin BUILDING.txt conf CONTRIBUTING.md lib LICENSE logs NOTICE README.md RELEASE-NOTES RUNNING.txt temp webapps work
[root@ip-172-31-11-35 apache-tomcat-9.0.96]# cd bin
[root@ip-172-31-11-35 bin]# ls
bootstrap.jar ciphers.sh daemon.sh makebase.sh startup.bat ✓ tool-wrapper.sh
catalina.bat commons-daemon.jar digest.bat setclasspath.bat startup.sh version.bat
catalina.sh commons-daemon-native.tar.gz digest.sh setclasspath.sh tomcat-juli.jar ✓ version.sh
catalina-tasks.xml configtest.bat html shutdown.bat tomcat-native.tar.gz
ciphers.bat configtest.sh makebase.bat shutdown.sh tool-wrapper.bat
[root@ip-172-31-11-35 bin]#
```

Now Go to the browser Paste IP address with Port Number.

Then you see the apache tomcat GUI.



Here you need to did the configurations and add the user in tomacatuser.xml file.

To find the file's use the command – find -name context.xml

Here leave first one only remaining the four file's you need to edit.

```
[root@ip-172-31-11-35 apache-tomcat-9.0.96]# find / -name context.xml
find: '/proc/15860': No such file or directory
/opt/apache-tomcat-9.0.96/conf/context.xml ✓
/opt/apache-tomcat-9.0.96/webapps/docs/META-INF/context.xml ✓
/opt/apache-tomcat-9.0.96/webapps/examples/META-INF/context.xml ✓
/opt/apache-tomcat-9.0.96/webapps/host-manager/META-INF/context.xml ✓
/opt/apache-tomcat-9.0.96/webapps/manager/META-INF/context.xml ✓
```

Use vi editor to open file

CMD: ---- vi /opt/apache-tomcat-9.0.96/webapps/docs/META-INF/context.xml

You wil see below interface.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more
    contributor license agreements. See the NOTICE file distributed with
    this work for additional information regarding copyright ownership.
    The ASF licenses this file to You under the Apache License, Version 2.0
    (the "License"); you may not use this file except in compliance with
    the License. You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<Context antiResourceLocking="false" >
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" /> |
</Context>
~
```

Under the context you see the Valve.

here just add like this <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"

```
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" /> -->
```

like this change the all file's.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more
    contributor license agreements. See the NOTICE file distributed with
    this work for additional information regarding copyright ownership.
    The ASF licenses this file to You under the Apache License, Version 2.0
    (the "License"); you may not use this file except in compliance with
    the License. You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<Context antiResourceLocking="false" >
    <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:1" /> -->
</Context>
~
```

After changing the all file's we have to attach user's.

These location you see tomcat-users.xml file there you need to attach user's.

```
[root@ip-172-31-11-35 conf]# pwd
/opt/apache-tomcat-9.0.96/conf
[root@ip-172-31-11-35 conf]# ls
Catalina          catalina.properties  jaspic-providers.xml  logging.properties  tomcat-users.xml  web.xml
catalina.policy   context.xml         jaspic-providers.xsd  server.xml           tomcat-users.xsd
[root@ip-172-31-11-35 conf]# |
```

Open file through vi editor add the bellow user's if you want change the user name and password.

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<user username="admin" password="password" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
<user username="Ramesh" password="Ramesh123" roles="manager-gui,manager-script"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

```

to operate the "/manager/html" web application. If you wish to use this app,
you must define such a user - the username and password are arbitrary.

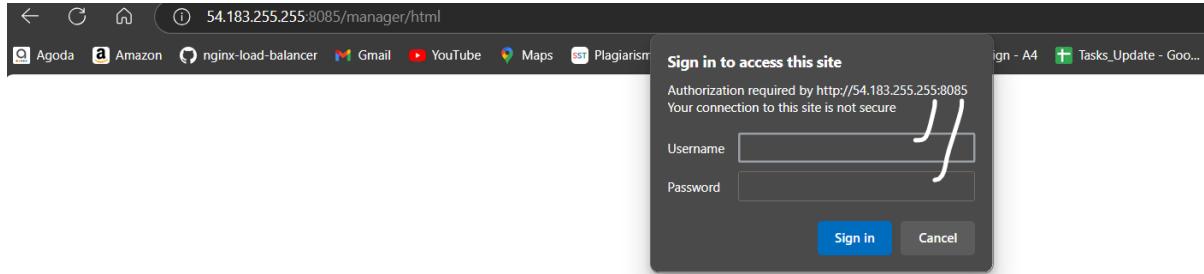
Built-in Tomcat manager roles:
- manager-gui      - allows access to the HTML GUI and the status pages
- manager-script   - allows access to the HTTP API and the status pages
- manager-jmx      - allows access to the JMX proxy and the status pages
- manager-status   - allows access to the status pages only

The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <...> that surrounds them. You
will also need to set the passwords to something appropriate.
-->
<!--
<user username="admin" password="" roles="manager-gui"/>
<user username="robot" password="" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <...> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="" roles="tomcat"/>
<user username="both" password="" roles="tomcat,role1"/>
<user username="role1" password="" roles="role1"/>
-->
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<user username="admin" password="password" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
<user username="Ramesh" password="Ramesh123" roles="manager-gui,manager-script"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>

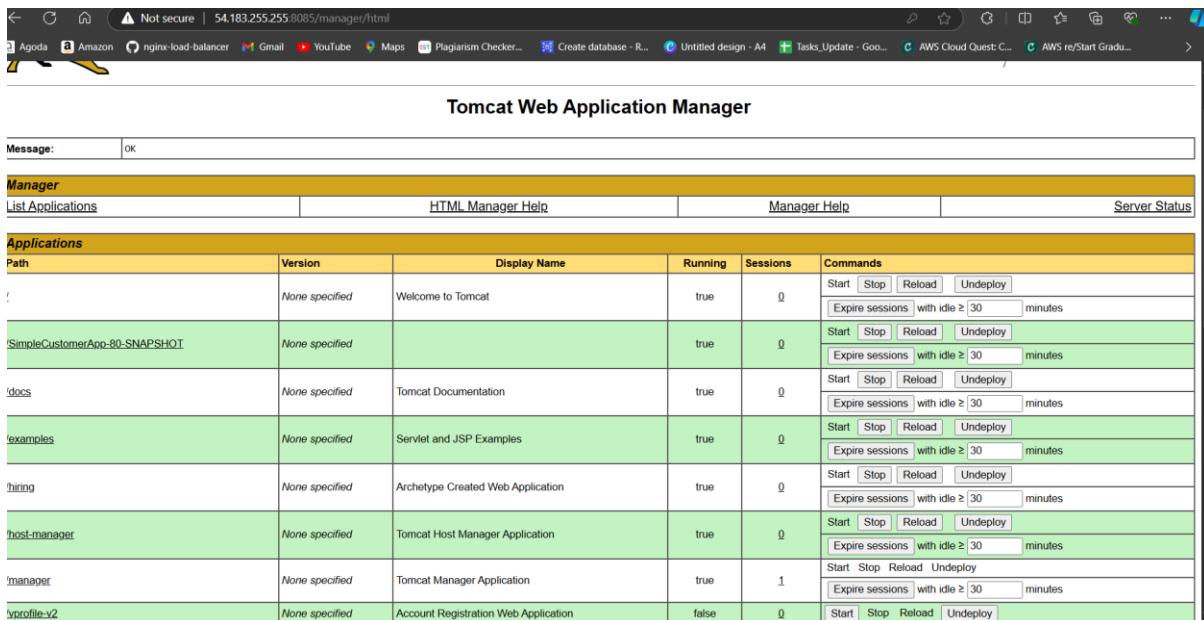
```

Save the file.

Now try to access the manager app and give the user name and password.



Now we are access.



Without did this all configurations and adding the user's it's not possible to login.

Now the all configurations all done.

Good to go create job and build the job.

Go to manage Jenkins.

Go system global configurations.

Now we did all configuration so we have create Job.

To create job click on new item, give name, and select the freestyle project and click on ok.

The screenshot shows the Jenkins 'New Item' creation interface. The 'Enter an item name' field is empty and highlighted with a yellow arrow. A validation message below it reads: 'This field cannot be empty, please enter a valid name.' The 'Select an item type' section contains four options: 'Freestyle project' (selected), 'Maven project', 'Pipeline', and 'Multi-configuration project'. Each option has a brief description. At the bottom right of the form is a blue 'OK' button, which is also highlighted with a yellow arrow.

Now go to the source code management.

Repository URL --- <https://github.com/Avularamesh/hiring-app.git>

Give branch as -- main

The screenshot shows the Jenkins 'Configuration' page for the 'Hiring_app' job. In the left sidebar, under 'Configure', the 'Source Code Management' section is expanded, showing the 'Git' configuration. The 'Repository URL' field is populated with 'https://github.com/Avularamesh/hiring-app.git' and is highlighted with a yellow arrow. The 'Credentials' dropdown is currently set to '- none -'. Other sections like 'General', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions' are visible in the sidebar.

Now go to the build step and click on Add build step.

There you see – Execute SonarQube Scanner.

In the Analysis properties ---- there you need to give the what developer given analysis code.

The screenshot shows a GitHub repository interface. On the left, there's a sidebar with navigation links like 'Dashboard', 'All', 'Hiring_app', and 'Configuration'. The main area displays a list of files: Dockerfile, Jenkinsfile, README.md, Untitled Diagram.drawio, jenkinsfile-cicd, and pom.xml. Below this is the 'README' section, which contains the following text:

```
In SonarAnalysis, use the below code...
sonar.projectKey=Test
sonar.sources=.
```

JDK--- inherit from job

The screenshot shows the Jenkins configuration page for the 'Hiring_app' job. The left sidebar has sections for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. In the 'Build Steps' section, there is a configuration for 'Execute SonarQube Scanner'. The 'JDK' dropdown is set to '(Inherit From Job)'. The 'Path to project properties' field is empty. The 'Analysis properties' field contains the following code:

```
sonar.projectKey=Test
sonar.sources=.
```

Next stage is – Build

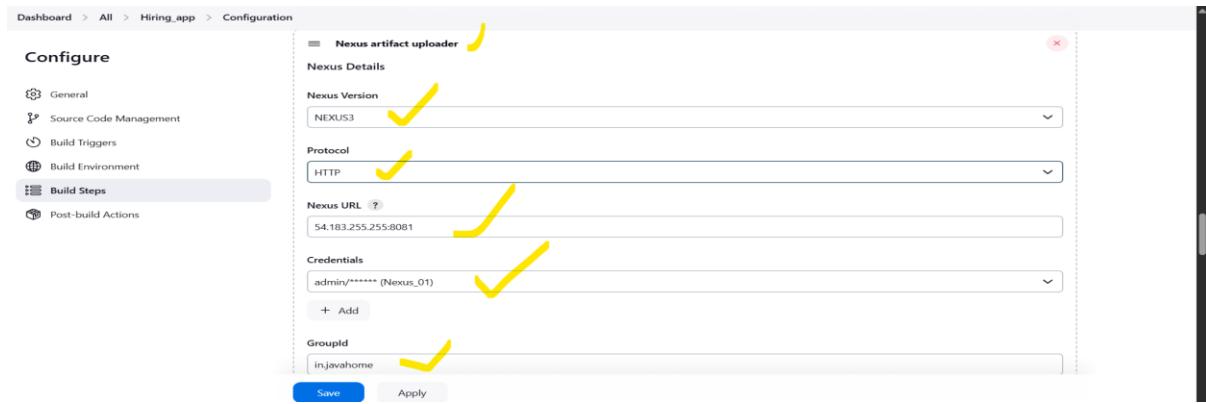
- Again build step and click on Add build step.
- There you seen the Invoke top-level maven targets.
- Maven version – select MVN_HOME
- Goal – clean install

The screenshot shows the Jenkins configuration page for the 'Hiring_app' job. The left sidebar has sections for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', and 'Build Steps' (which is selected). In the 'Build Steps' section, there is a configuration for 'Invoke top-level Maven targets'. The 'Maven Version' dropdown is set to 'MVN_HOME'. The 'Goals' field contains the following text:

```
clean install
```

Now Next stage nexus

- Again build step and click on Add build step.
- Select Nexus artefact uploader and version --- NEXUS3
- Protocol --- HTTP and give Nexus URL --- <URL>
- Select the credentials what you saved in the global credentials.

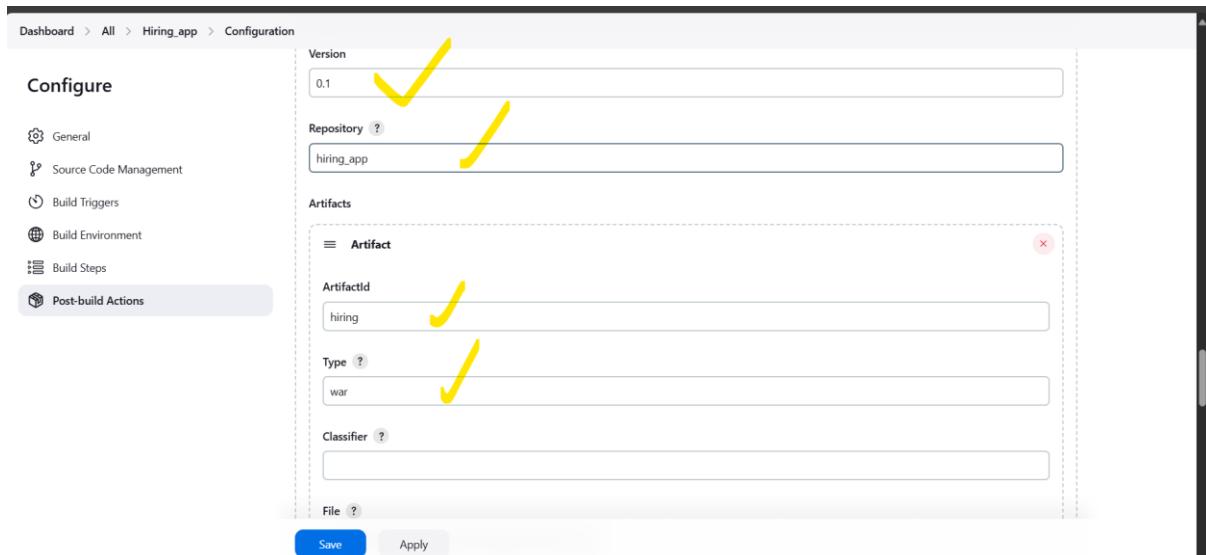


the above I am given group ID we have took in Pom.xml file

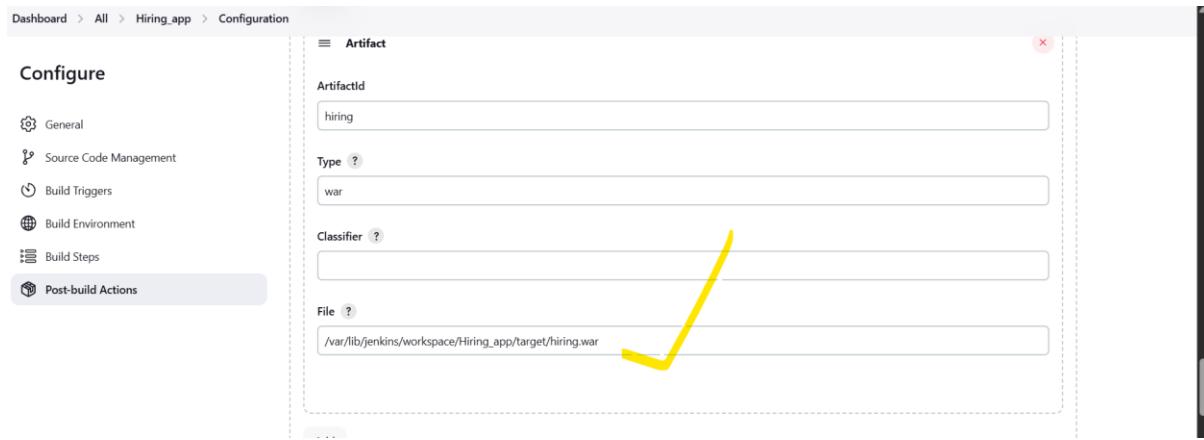
```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>in.java.home</groupId>
  <artifactId>hiring</artifactId>
  <packaging>war</packaging>
  <version>0.1</version>
  <name>hiring Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
  
```

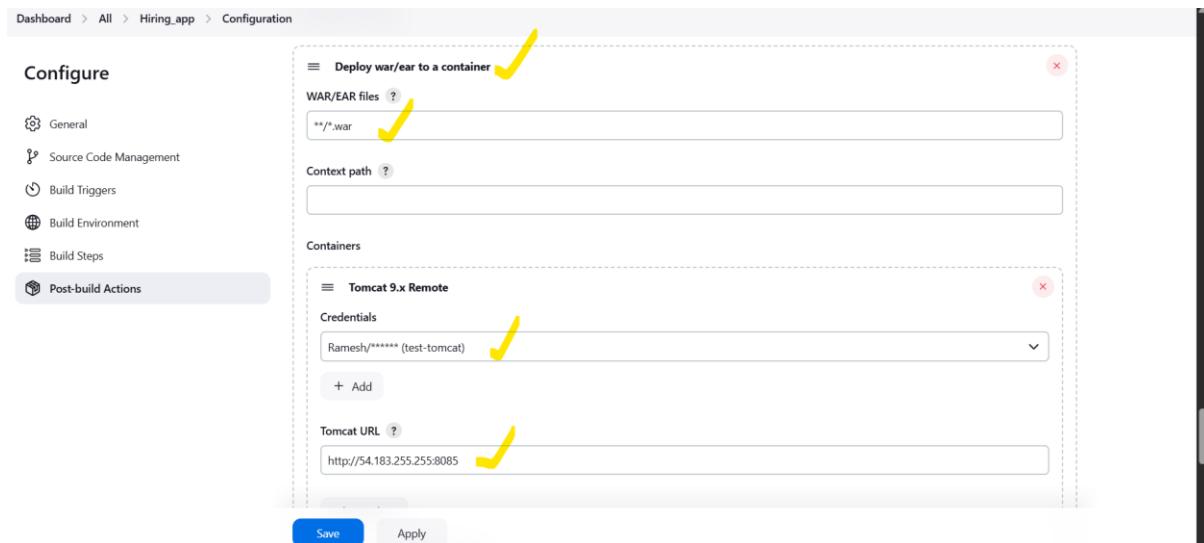
- Here version --- 0.1
- Repository name ---- give nexus what we created repo name.
- ArtifactId --- hiring and type ---war



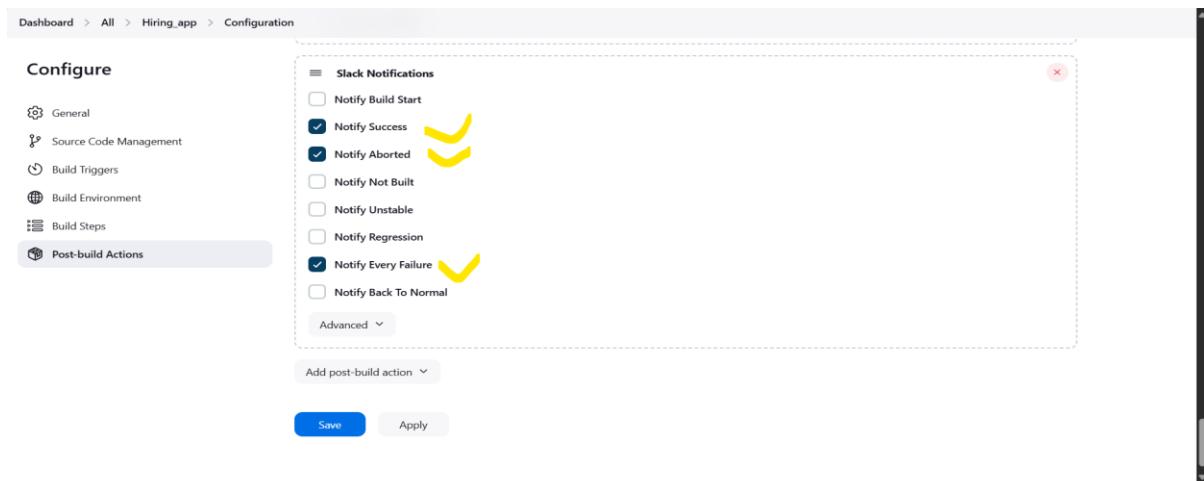
File – give your file location that will be generated after build.



- After Nexus we have to deploy war file in a tomcat.
- Go to the post build actions click on --- Deploy war/ear to container.
- WAR/EAR files ---- **/*.war always
- After that give the credentials of tomcat and give URL of tomcat.

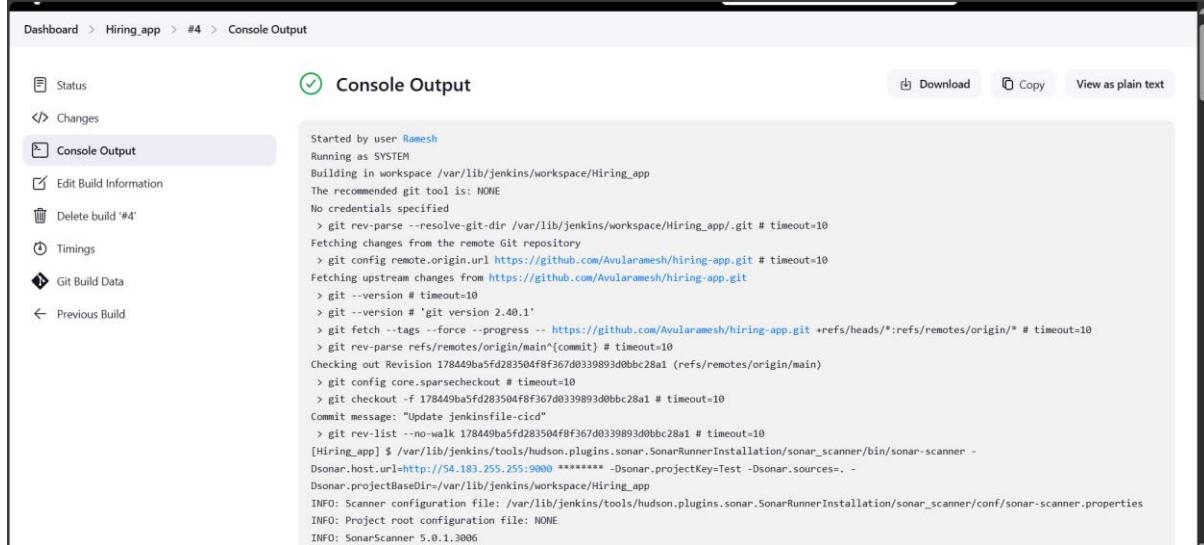


- Again go the Post build actions click on Slack Notifications.
- There you need to select – Notify success, Notify Aborted, notify every failure.
- After made this changes click on apply and save.



Now build the Job

The first stage is successfule.

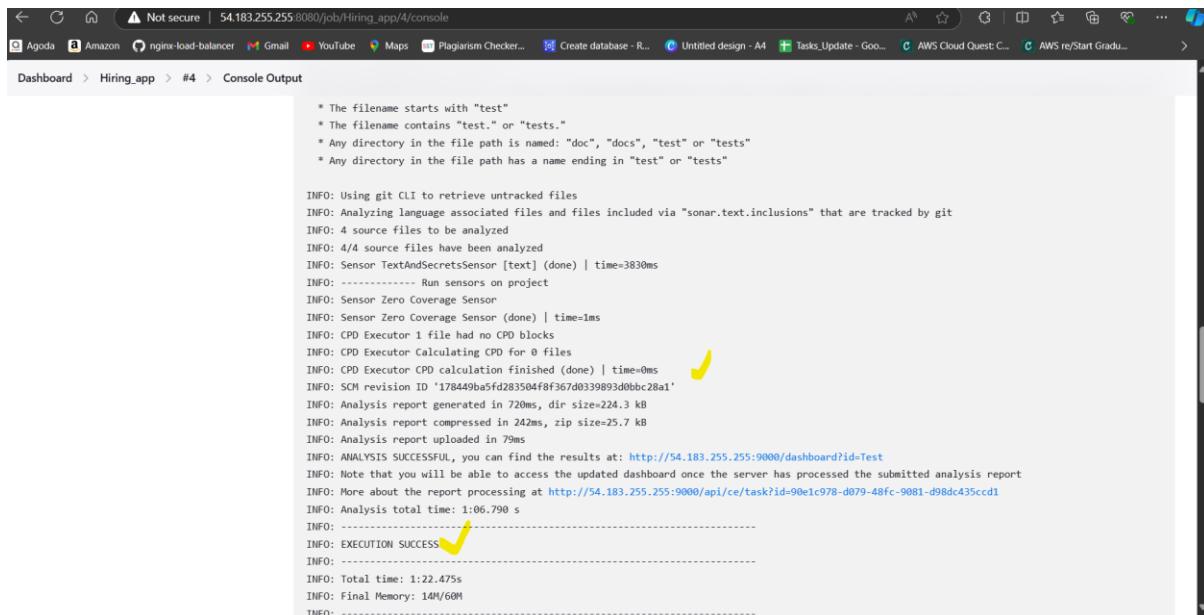


Dashboard > Hiring_app > #4 > Console Output

Status Changes Console Output Edit Build Information Delete build #4 Timings Git Build Data Previous Build

Started by user Ramesh
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Hiring_app
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Hiring_app/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Avularamesh/hiring-app.git # timeout=10
Fetching upstream changes from https://github.com/Avularamesh/hiring-app.git
> git --version # timeout=10
> git -version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/Avularamesh/hiring-app.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 178449ba5fd283504f8f87367d0339893d0bbc28a1 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 178449ba5fd283504f8f87367d0339893d0bbc28a1 # timeout=10
Commit message: "Update jenkinsfile-cicd"
> git rev-list --no-walk 178449ba5fd283504f8f87367d0339893d0bbc28a1 # timeout=10
[Hiring_app] \$ /var/lib/jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonar_scanner/bin/sonar-scanner -Dsonar.host.url=http://54.183.255.9000 ***** -Dsonar.projectKey=Test -Dsonar.sources=. -Dsonar.projectBaseDir=/var/lib/jenkins/workspace/Hiring_app
INFO: Scanner configuration file: /var/lib/jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonar_scanner/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 5.0.1.3006

Second stage also successfule.

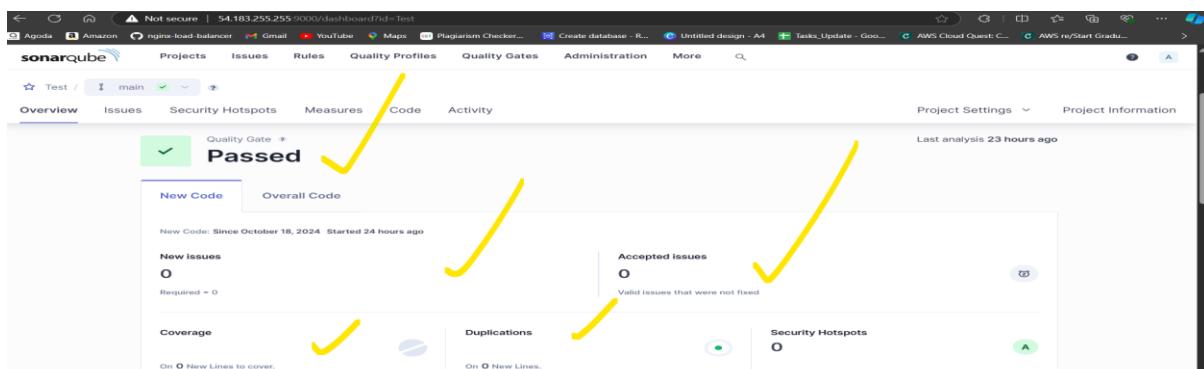


Dashboard > Hiring_app > #4 > Console Output

* The filename starts with "test"
* The filename contains "test." or "tests."
* Any directory in the file path is named: "doc", "docs", "test" or "tests"
* Any directory in the file path has a name ending in "test" or "tests"

INFO: Using git CLI to retrieve untracked files
INFO: Analyzing language associated files and files included via "sonar.text.inclusions" that are tracked by git
INFO: 4 source files to be analyzed
INFO: 4/4 source files have been analyzed
INFO: Sensor TextAndSecretsSensor [text] (done) | time=3830ms
INFO: ----- Run sensors on project
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=1ms
INFO: CPD Executor 1 file had no CPD blocks
INFO: CPD Executor Calculating CPD for 0 files
INFO: CPD Executor CPD calculation finished (done) | time=0ms
INFO: SCM revision ID '178449ba5fd283504f8f87367d0339893d0bbc28a1'
INFO: Analysis report generated in 720ms, dir size=224.3 kB
INFO: Analysis report compressed in 242ms, zip size=25.7 kB
INFO: Analysis report uploaded in 79ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://54.183.255.9000/dashboard?id=Test
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://54.183.255.9000/api/ce/task?id=90e1c978-d079-48fc-9081-d98dc435cc1
INFO: Analysis total time: 1:06.790 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 1:22.475s
INFO: Final Memory: 14M/60M
INFO: -----

You see in sonarqube also like what bugs are there the quality gate is passed or not and Duplications.



The Next stage is build stage.

This stage also succesful.

```
Dashboard > Hiring_app > #4 > Console Output
[INFO] No tests to run.
[INFO] Surefire report directory: /var/lib/jenkins/workspace/Hiring_app/target/surefire-reports
-----
TESTS
-----
Results :
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO] --- maven-war-plugin:3.3.1:war (default-war) @ hiring ---
[INFO] Packaging webapp [hiring] in [/var/lib/jenkins/workspace/Hiring_app/target/hiring]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/Hiring_app/src/main/webapp]
[INFO] Building war: /var/lib/jenkins/workspace/Hiring_app/target/hiring.war
[INFO] 
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ hiring ---
[INFO] Installing /var/lib/jenkins/workspace/Hiring_app/target/hiring.war to /var/lib/jenkins/.m2/repository/in/javahome/hiring/0.1/hiring-0.1.war
[INFO] Installing /var/lib/jenkins/workspace/Hiring_app/pom.xml to /var/lib/jenkins/.m2/repository/in/javahome/hiring/0.1/hiring-0.1.pom
[INFO] 
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.316s
[INFO] Finished at: Fri Oct 18 06:31:30 UTC 2024
[INFO] Final Memory: 11M/30M
[INFO] 
Uploading artifact hiring.war started...
```

This location war is created.

```
workspace
[root@ip-172-31-11-35 jenkins]# cd workspace/
[root@ip-172-31-11-35 workspace]# ls
clone_job Hiring_app Project Sonarqube_Integration test-project V-profile
[root@ip-172-31-11-35 workspace]# cd Hiring_app/
[root@ip-172-31-11-35 Hiring_app]# ls
Dockerfile Jenkinsfile jenkinsfile-cicd pom.xml README.md src target Untitled Diagram.drawio
[root@ip-172-31-11-35 Hiring_app]# cd target/
[root@ip-172-31-11-35 target]# ls
hiring hiring.war maven-archiver surefire
[root@ip-172-31-11-35 target]# pwd
/var/lib/jenkins/workspace/Hiring_app/target
[root@ip-172-31-11-35 target]# |
```

Upload the war file into nexus.

```
Uploading artifact hiring.war started...
GroupId: in.javahome
ArtifactId: hiring
Classifier:
Type: war
Version: 0.1
File: hiring.war
Repository:hiring_app
Uploading: http://54.183.255.255:8081/repository/hiring_app/in/javahome/hiring/0.1/hiring-0.1.war
100 % completed (1.7 kB / 1.7 kB).
Uploaded: http://54.183.255.255:8081/repository/hiring_app/in/javahome/hiring/0.1/hiring-0.1.war (1.7 kB at 3.0 kB/s)
Uploading artifact hiring.war completed.
```

War file aslo uploaded in to nexues.

The screenshot shows the Sonatype Nexus Repository interface. The left sidebar has a green 'Browse' button selected. The main area shows a tree view of uploaded components. Under 'in/javahome/hiring/0.1', there are three files: 'hiring-0.1.war', 'hiring-0.1.war.md5', and 'hiring-0.1.war.sha1'. The 'hiring-0.1.war' file is highlighted with a yellow arrow.

Now Deploy to tomcat server.

```
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/Hiring_app/target/hiring.war to container Tomcat 9.x Remote with context null
[/var/lib/jenkins/workspace/Hiring_app/target/hiring.war] is not deployed. Doing a fresh deployment.
Deploying [/var/lib/jenkins/workspace/Hiring_app/target/hiring.war]
[Slack Notifications] found #3 as previous completed, non-aborted build
[Slack Notifications] will send OnSuccessNotification because build matches and user preferences allow it
Finished: SUCCESS
```

Just click on /hiring

The screenshot shows the Apache Tomcat Manager interface. The 'Applications' section lists several web applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes
/SimpleCustomerApp-80.SNAPSHOT	None specified		true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes
/docs	None specified	Tomcat Documentation	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes
/examples	None specified	Servlet and JSP Examples	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes
/hiring	None specified	Archetype Created Web Application	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes
/manager	None specified	Tomcat Manager Application	true	1	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes
/profile-v2	None specified	Account Registration Web Application	false	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button>
/webapps	None specified		true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="text"/> Expire sessions with idle ≥ [30] minutes

Deploy
Deploy directory or WAR file located on server

Context Path:

You see like the these.

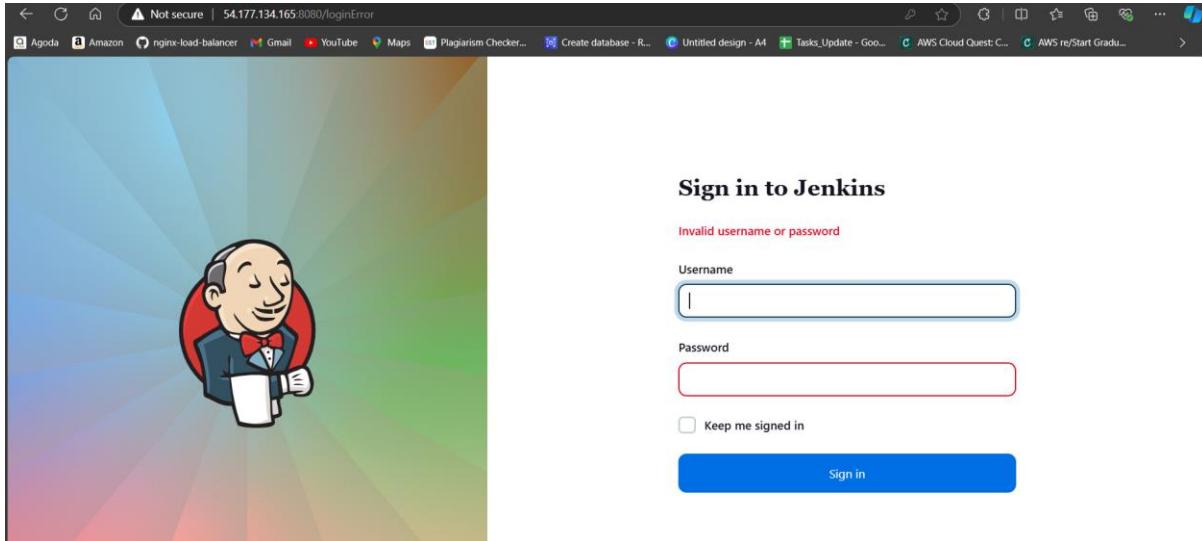
The screenshot shows a browser window displaying the 'Welcome to Techie Horizon' page. The URL in the address bar is 54.183.255.255:8085/hiring/.

slack notification also Done.

A screenshot of a Slack message from the 'jenkins' app. The message is timestamped at 12:01 PM and reads: "Hiring_app - #4 Success after 1 min 47 sec (Open)". The message has a green checkmark icon and several other small icons below it.

Task is completed.

Issue: in the badcase I forgot the Jenkins admin password that time I am unable to login that point off time I followed below the steps.



Then I connect to the ec2 server.

I edit the file.

CMD: vi /var/lib/jenkins/config.xml

<usesecurity>true</usesecurity>

I change it as the false. <usesecurity>false</usesecurity>

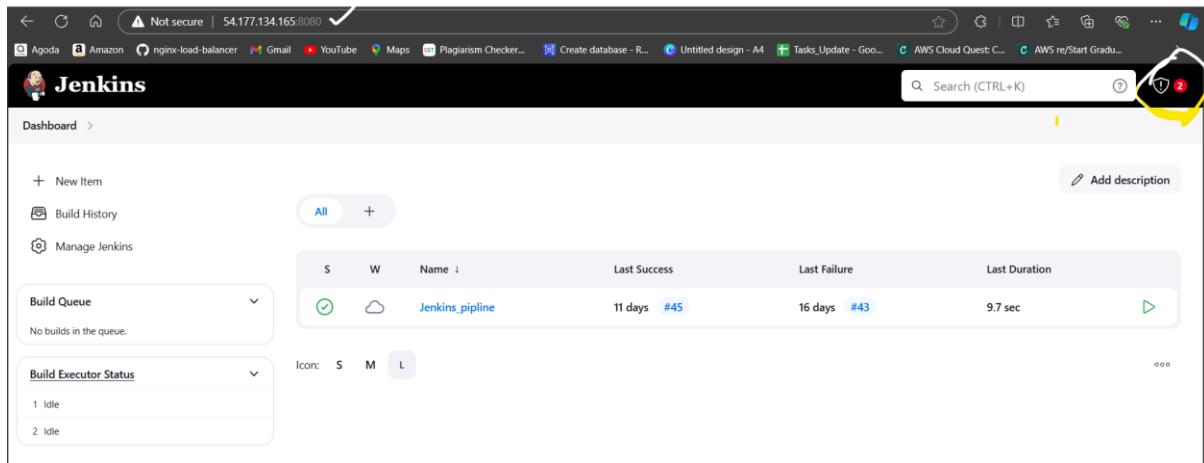
```
<?xml version='1.1' encoding='UTF-8'?>
<hudson>
  <disabledAdministrativeMonitors/>
  <version>2.462.3</version>
  <numExecutors>2</numExecutors>
  <mode>NORMAL</mode>
  <useSecurity>true</useSecurity>
  <authorizationStrategy class="hudson.security.FullControlOnceLoggedInAuthorizationStrategy">
    <denyAnonymousReadAccess>true</denyAnonymousReadAccess>
  </authorizationStrategy>
  <securityRealm class="hudson.security.HudsonPrivateSecurityRealm">
    <disableSignup>true</disableSignup>
    <enableCaptcha>false</enableCaptcha>
  </securityRealm>
  <disableRememberMe>false</disableRememberMe>
  <projectNamingStrategy class="jenkins.model.ProjectNamingStrategy$DefaultProjectNamingStrategy"/>
  <workspaceDir>${JENKINS_HOME}/workspace/${ITEM_FULL_NAME}</workspaceDir>
  <buildsDir>${ITEM_ROOTDIR}/builds</buildsDir>
  <jdk></jdk>
  <viewsTabBar class="hudson.views.DefaultViewsTabBar"/>
  <myViewsTabBar class="hudson.views.DefaultMyViewsTabBar"/>
  <clouds/>
  <scmCheckoutRetryCount>0</scmCheckoutRetryCount>
  <views>
    <hudson.model.AllView>
      <owner class="hudson" reference="...."/>
      <name>all</name>
      <filterExecutors>false</filterExecutors>
      <filterQueue>false</filterQueue>
      <properties class="hudson.model.View$PropertyList"/>
    </hudson.model.AllView>
  </views>
  <primaryView>all</primaryView>
  <slaveAgentPort>1</slaveAgentPort>
  <label></label>
  <crumbIssuer class="hudson.security.csrf.DefaultCrumbIssuer">
    <excludeClientIPFromCrumb>false</excludeClientIPFromCrumb>
  </crumbIssuer>
  <nodeProperties/>
  <globalNodeProperties/>
</var/lib/jenkins/config.xml" [noeol] 42L, 1660B
```

```
<?xml version='1.1' encoding='UTF-8'?>
<hudson>
  <disabledAdministrativeMonitors/>
  <version>2.462.3</version>
  <numExecutors>2</numExecutors>
  <mode>NORMAL</mode>
  <useSecurity>false</useSecurity>
  <authorizationStrategy class="hudson.security.FullControlOnceLoggedInAuthorizationStrategy">
    <denyAnonymousReadAccess>true</denyAnonymousReadAccess>
  </authorizationStrategy>
  <securityRealm class="hudson.security.HudsonPrivateSecurityRealm">
    <disableSignup>true</disableSignup>
    <enableCaptcha>false</enableCaptcha>
  </securityRealm>
  <disableRememberMe>false</disableRememberMe>
  <projectNamingStrategy class="jenkins.model.ProjectNamingStrategy$DefaultProjectNamingStrategy"/>
  <workspaceDir>${JENKINS_HOME}/workspace/${ITEM_FULL_NAME}</workspaceDir>
  <buildsDir>${ITEM_ROOTDIR}/builds</buildsDir>
  <jdk></jdk>
```

After changing as false save the file.

You need to restart the Jenkins server. CMD: systemctl restart jenkins

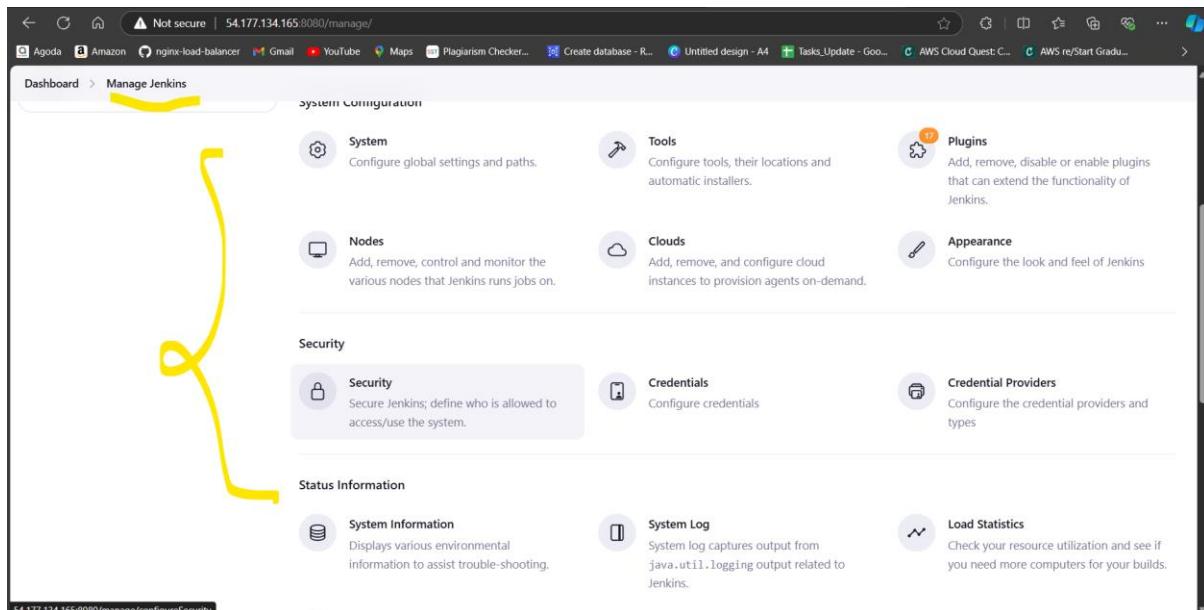
Then now try to login. Now you can access the Jenkins GUI.



- User have the credential that will secure the Jenkins server and this is best practice to secure the Jenkins server.
- But your not Login as admin user. So this this is not best practice. So you need to change the password of admin user.

To change the password you need to follow the below steps:

- Go to the manage Jenkins.
- Here we haven't found and option as users. So go to the security credentials.



- Then you will see the below interface.
- You see the option Security Realm --- None

The screenshot shows the Jenkins 'Manage Jenkins > Security' page. The 'Security' tab is selected. Under 'Authentication', there is a checkbox for 'Disable "Keep me signed in"'. Under 'Authorization', it says 'Anyone can do anything'. A yellow arrow points to the 'Security Realm' dropdown, which is set to 'None'.

- Change as Jenkin's own user database.
- Then save and comeback to manage Jenkins.

The screenshot shows the same Jenkins 'Manage Jenkins > Security' page. The 'Security Realm' dropdown has been changed to 'Jenkins' own user database'. A yellow arrow points to this selection. There is also a checkbox for 'Allow users to sign up'.

Now you can see the user's Option and click on that.

The screenshot shows the Jenkins 'Manage Jenkins > System Configuration' page. In the 'Security' section, there are links for 'Security', 'Users', and 'Credentials'. A yellow arrow points to the 'Users' link.

Click on admin.

The screenshot shows the Jenkins 'Manage Jenkins > Jenkins' own user database > Users' page. It lists one user: 'admin' with 'Ramesh' as the name. A yellow arrow points to the 'admin' entry in the list.

User ID	Name
admin	Ramesh

After that click on configure and scroll down.

You see the password and you need to change the password.

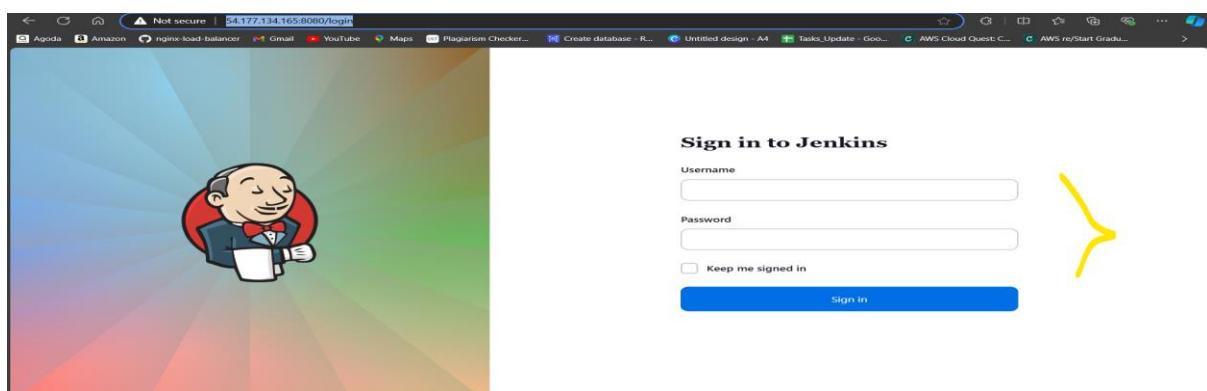
The screenshot shows the Jenkins configuration page for a user named 'Ramesh'. The 'Configure' tab is selected. In the 'Password' section, there are two input fields: 'Password:' and 'Confirm Password:', both containing redacted text. A yellow arrow points from the text 'Here I am given my new password and click on the save button.' to the 'Password:' field. Below the password fields is a 'Session Termination' section with a 'Terminate All Sessions' button. At the bottom of the page are 'Setting for search' options, including 'Case-sensitivity' with a checked 'Insensitive search tool' checkbox, and 'Save' and 'Apply' buttons. The URL in the address bar is 'Dashboard > Ramesh > Configure'.

Here I am given my new password and click on the save button.

The screenshot shows the same Jenkins configuration page after changing the password. The 'Password' section now displays four dots in each of the 'Password:' and 'Confirm Password:' fields, indicating they have been modified. A yellow arrow points from the text 'After that go to the browser--- <ipaddress:8080/login>' to the 'Password:' field. The rest of the page remains the same, including the 'Session Termination' section and the 'Setting for search' options at the bottom. The URL in the address bar is 'Dashboard > Ramesh > Configure'.

After that go to the browser--- <ipaddress:8080/login>

Now give your user name and password.



Now you can access as user.

The screenshot shows the Jenkins dashboard. On the left, there are links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). In the center, a table lists a single job: 'Jenkins_pipeline'. The table columns are 'S' (Status), 'W' (Workflow), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The 'Jenkins_pipeline' row has a green circle icon for status, a cloud icon for workflow, and the name 'Jenkins_pipeline'. It shows '11 days #45' as the last success, '16 days #43' as the last failure, and '9.7 sec' as the last duration. A yellow checkmark is drawn over the 'Jenkins_pipeline' row. At the top right, there is a search bar, a help icon, a user profile for 'Ramesh', and a 'log out' link. Another yellow checkmark is drawn near the 'log out' link.

But here we facing the problem Now.

with our ip address:<port number> any one access the our Jenkins GUI.

This not good for our organization so to solve that we need to change the some configurations.

To secure our Jenkins GUI follow below steps:

- Connect to the server.
- vi /var/lib/jenkins/config.xml open this file

```
root@ip-172-31-1-15 ~]# client_loop: send disconnect: Connection reset
PS C:\Users\ramee\downloads> ssh -i "Ncalifornia.pem" ec2-user@ec2-54-177-134-165.us-west-1.compute.amazonaws.com
A newer release of "Amazon Linux" is available.
Version 2023.6.20241010:
Run "/usr/bin/dnf check-release-update" for full release and version update info
      _#
     /###_
    /#####
   /#####
  /#####
 /#####
V-->
 /_/
 /_/
 /_m/
.last login: Thu Oct 24 05:28:47 2024 from 106.222.235.124
[ec2-user@ip-172-31-1-15 ~]$ sudo su -
.last login: Thu Oct 24 05:29:00 UTC 2024 on pts/1
[root@ip-172-31-1-15 ~]# vi /var/lib/jenkins/config.xml
[root@ip-172-31-1-15 ~]# vi /var/lib/jenkins/config.xml
'/var/lib/jenkins/config.xml' 41L, 1628B written
```

There you need to observe <authorizationstratgey> here you need to change.

The screenshot shows the Jenkins configuration file 'config.xml'. The 'authorizationStrategy' section is highlighted with a yellow box. It contains the following XML code:
<authorizationStrategy class="hudson.security.AuthorizationStrategy\$Unsecured"/>
This indicates that the Jenkins instance is currently configured to be unsecured, which is why anyone can access the GUI via the IP address and port.

This lines you need to add in the file.

```
<authorizationStrategy class="hudson.security.FullControlOnceLoggedInAuthorizationStrategy">  
    <denyAnonymousReadAccess>true</denyAnonymousReadAccess>  
</authorizationStrategy>
```

```
<?xml version='1.1' encoding='UTF-8'?>  
<hudson>  
    <disabledAdministrativeMonitors/>  
    <version>2.462.3</version>  
    <numExecutors>2</numExecutors>  
    <mode>NORMAL</mode>  
    <useSecurity>true</useSecurity>  
    <authorizationStrategy class="hudson.security.FullControlOnceLoggedInAuthorizationStrategy">  
        <denyAnonymousReadAccess>true</denyAnonymousReadAccess>  
    | </authorizationStrategy>  
    <securityRealm class="hudson.security.HudsonPrivateSecurityRealm">  
        <disableSignup>true</disableSignup>  
        <enableCaptcha>false</enableCaptcha>  
    </securityRealm>  
    <disableRememberMe>false</disableRememberMe>  
    <projectNamingStrategy class="jenkins.model.ProjectNamingStrategy$DefaultProjectNamingStrategy"/>  
    <workspaceDir>${JENKINS_HOME}/workspace/${ITEM_FULL_NAME}</workspaceDir>  
    <buildsDir>${ITEM_ROOTDIR}/builds</buildsDir>  
    <markupFormatter class="hudson.markup.EscapedMarkupFormatter"/>  
    <jdks/>  
    <viewsTabBar class="hudson.views.DefaultViewsTabBar"/>  
    <myViewsTabBar class="hudson.views.DefaultMyViewsTabBar"/>  
    <clouds/>  
    <scmCheckoutRetryCount>0</scmCheckoutRetryCount>  
    <views>  
        <hudson.model.AllView>  
            <owner class="hudson" reference=". . . . ."/>  
            <name>all</name>  
            <filterExecutors>false</filterExecutors>  
            <filterQueue>false</filterQueue>  
            <properties class="hudson.model.View$PropertyList"/>  
        </hudson.model.AllView>  
    </views>  
    <primaryView>all</primaryView>  
    <slaveAgentPort>-1</slaveAgentPort>  
    <label></label>  
    <crumbIssuer class="hudson.security.csrf.DefaultCrumbIssuer">  
        <excludeClientIPFromCrumb>false</excludeClientIPFromCrumb>  
    </crumbIssuer>  
    <nodeProperties/>  
>/var/lib/jenkins/config.xml" 43L, 1727B
```

10,2

After that save the file and restart the Jenkins server.

To restart the Jenkins server CMD: systemctl restart Jenkins

Then go to the browser <http://54.177.134.165:8080/>

It will automatically ask the Login credentials.

Give the credentials and Login.

