

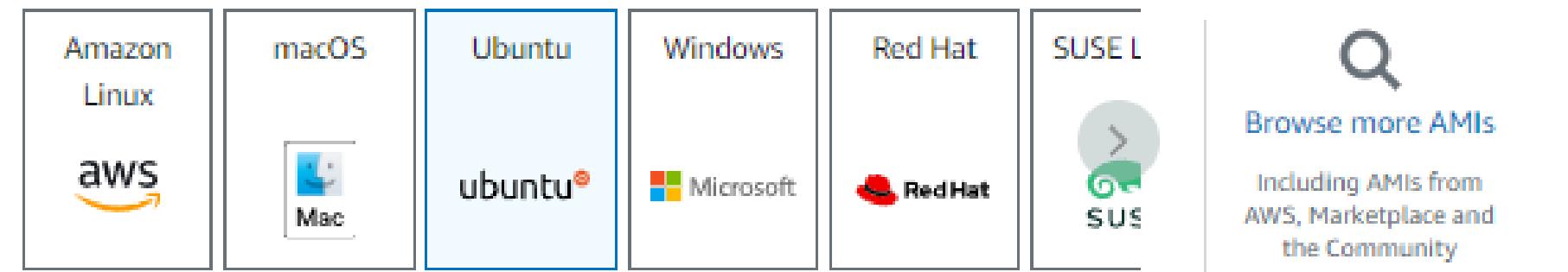
Project on

CI PIPELINE SETUP WITH VARIOUS TOOLS

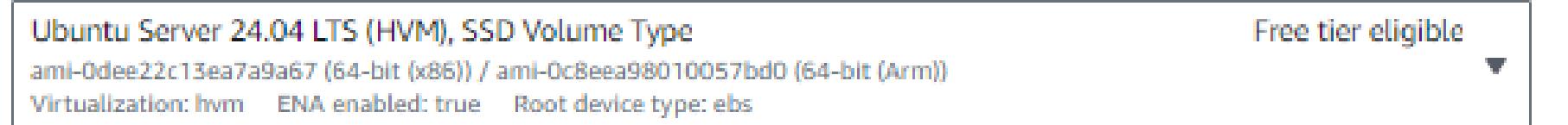
(JENKINS, DOCKER, SONARQUBE, MAVEN, TRIVY(FOR DOCKER IMAGE SCAN))

Step 1 – Create the Ec2 instance in AWS account with these parameters

- EC2 type – Ubuntu t2.medium
- EBS volume – 30 GB
- Region – Asia Pacific (Mumbai)ap-south-1

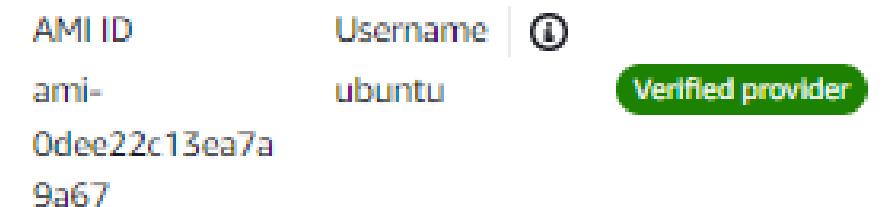
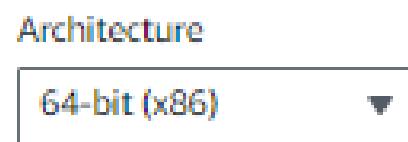


Amazon Machine Image (AMI)

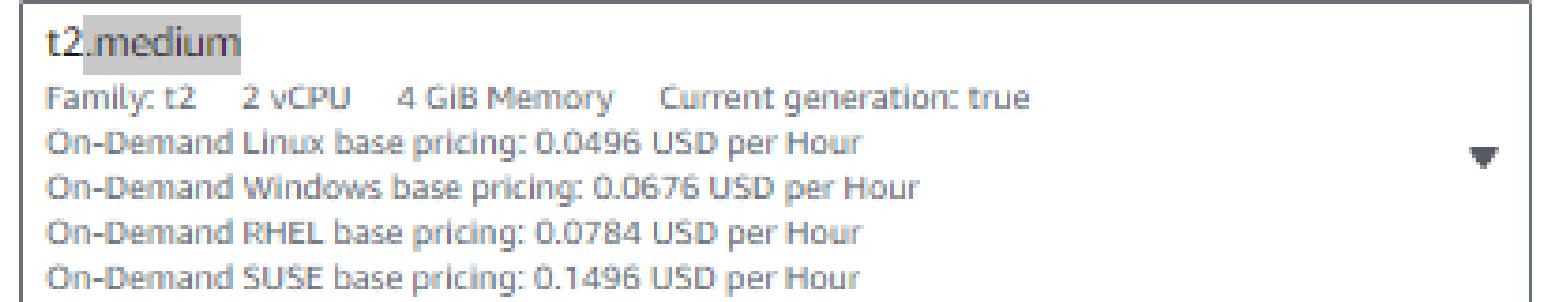


Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).



Instance type



Configure storage

1x GiB

Root volume (Not encrypted)

All generations

[Compare instance types](#)

Number of instances | Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64 noble image
ami-0dee22c13ea7a9a67

Virtual server type (instance type)
t2.medium

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 30 GiB

Cancel Launch instance Review commands

Step 2 – Connect to EC2 and Install all tools in that system as root user

The screenshot shows the AWS EC2 Instances page with one instance listed:

- Name:** CI_Pipeline-Ubuntu
- Instance ID:** i-0717e7e6b747c9db4
- Instance type:** t2.medium (highlighted with a red box)
- State:** Running
- Status check:** 2/2 checks passed
- Alarm status:** View alarms
- Availability Zone:** ap-south-1a
- Public IPv4 DNS:** ec2-3-109-217-190.ap-south-1.compute.amazonaws.com

The instance details page for i-0717e7e6b747c9db4 (CI_Pipeline-Ubuntu) shows:

- Public IPv4 address:** 3.109.217.190 (highlighted with a red box)
- Private IPv4 addresses:** 172.31.36.240 (highlighted with a red box)
- Public IPv4 DNS:** ec2-3-109-217-190.ap-south-1.compute.amazonaws.com (highlighted with a red box)

The AWS Block Devices page shows:

- Volume ID:** vol-0f99740294d84a9ec
- Device name:** /dev/sda1
- Volume size (GiB):** 30 (highlighted with a red box)
- Attachment status:** Attached
- Attachment time:** 2024/10/02 09:45:23

- Use your EC2 Private key properly
- Take SSH of your instance through MobXterm, VS code, Putty or CMD
- To login as root user - sudo su

```
ubuntu@ip-172-31-36-240:~$ sudo su
root@ip-172-31-36-240:/home/ubuntu# curl ifconfig.io
3.109.217.190
root@ip-172-31-36-240:/home/ubuntu# cd
root@ip-172-31-36-240:~/#
root@ip-172-31-36-240:~# ifconfig
enX0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
        inet 172.31.36.240 netmask 255.255.240.0 broadcast 172.31.47.255
              ether 02:fc:60:13:6c:7b txqueuelen 1000 (Ethernet)
              ether 02:fc:60:13:6c:7b txqueuelen 1000 (Ethernet)
```

Step 3 – Install Jenkins on Ubuntu

Follow step-by-step properly

- `sudo apt update -y`
- `sudo apt upgrade -y`
- `sudo apt install openjdk-17-jre -y`
- `curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null`
- `cat /usr/share/keyrings/jenkins-keyring.asc` =====> To see the public key of Jenkins
- `echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null`
- `cat /etc/apt/sources.list.d/jenkins.list`
- `sudo apt-get update -y`
- `sudo apt-get install jenkins -y`
- `dpkg -s jenkins`

```
root@ip-172-31-36-240:~# dpkg -s jenkins
Package: jenkins
Status: install ok installed
Priority: optional
Section: devel
Installed-Size: 91248
Maintainer: Kohsuke Kawaguchi <kk@kohsuke.org>
Architecture: all
Version: 2.462.3
Depends: adduser, lsb-base (>= 3.2-14), net-tools, sysvinit-utils (>= 2.88dsf-50)
Pre-Depends: init-system-helpers (>= 1.54~)
```

Step 4 – Change the security group of EC2 instance

The screenshot shows the AWS Management Console with the EC2 service selected. The user is editing the inbound rules for a security group named 'sg-0734bb623ce1ace54 - default'. Two rules are present: one for 'All traffic' (Type: All traffic) and one for 'SSH' (Type: SSH). Both 'Type' dropdowns are highlighted with red boxes. In the 'Details' section, the 'Security group ID' (sg-0734bb623ce1ace54) is also highlighted with a red box. The 'Inbound rules' table at the bottom shows two entries, both with their 'Type' columns highlighted with red boxes.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0242ab9bdb8576345	All traffic	All	All	Anyw... 0.0.0.0/0	
-	SSH	TCP	22	Anyw... 0.0.0.0/0	

Cancel Preview changes Save rules

Details

Security group name	Security group ID	Description	VPC ID
default	sg-0734bb623ce1ace54	default VPC security group	vpc-0ed78ddb4cc11fd3f

Inbound rules Outbound rules Tags

Inbound rules (2)

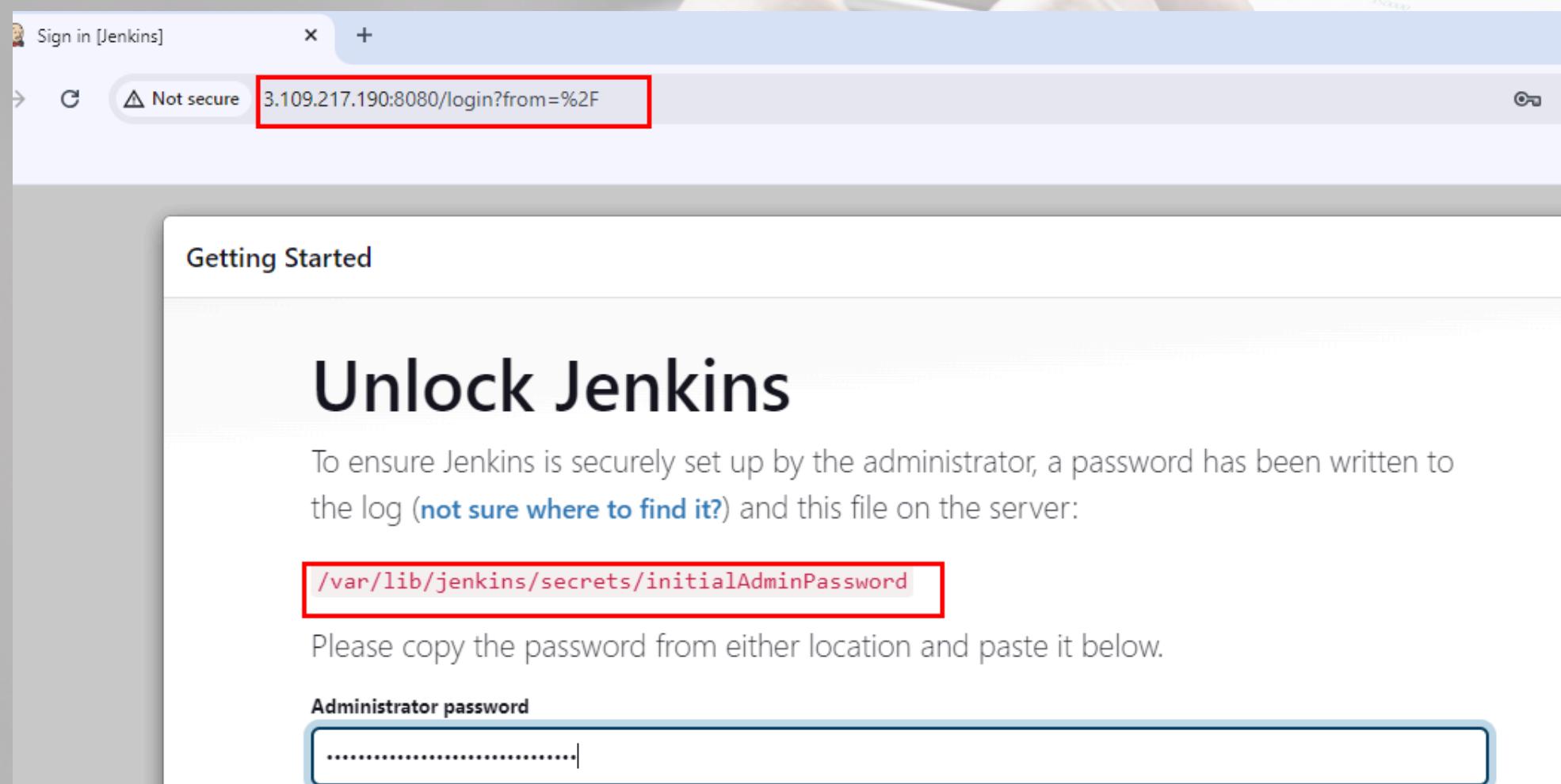
Search	
Security group rule... IP version Type	IPv4 All traffic
sgr-0242ab9bdb8576... IPv4 All	0.0.0.0/0
sgr-08b0b340cae8bbd... IPv4 TCP	0.0.0.0/0

Manage tags Edit inbound rules

Step 5– Sign Into Jenkins console

- http://<EC2_Public_IP>:8080/
- <http://3.109.217.190:8080/>

Step 6 – Get the Administrator password by hitting the below command in EC2



```
root@ip-172-31-36-240:~# cat /var/lib/jenkins/secrets/initialAdminPassword  
a2fe583cb8e14f83a6e98f6ae2ae92ad ←  
root@ip-172-31-36-240:~#
```

Step 7 – Install all suggested plugins

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.

Not secure 3.109.217.190:8080

Getting Started

Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding Plugin	** Pipeline: Supporting APIs ** Plugin Utilities API ** Font Awesome API ** Bootstrap 5 API ** JQuery3 API ** ECharts API ** Display URL API ** Checks API ** JUnit ** Matrix Project ** Resource Disposer Workspace Cleanup Ant ** OkHttp ** Durable Task ** Pipeline: Nodes and Processes
✓ Timestamper	✓ Workspace Cleanup Plugin	✓ 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	

Step 8 –Create first user

Not secure 3.109.217.190:8080

Getting Started

Create First Admin User

Username: admin

Password:

Confirm password:

Full name: admin

Not secure 3.109.217.190:8080

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

Step 9 – Create a pipeline Job

Jenkins

Search (CTRL+K)

admin

Dashboard > All > New Item

New Item

Enter an item name
pipeline

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. **Red box highlights this option.**
- 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 actually stores items.

OK

Not secure 3.109.217.190:8080/job/pipeline/configure

Jenkins

Search (CTRL+K)

admin

Dashboard > pipeline > Configuration

Configure General

General

Description

Plain text [Preview](#)

Discard old builds ?

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

GitHub project

Pipeline speed/durability override ?

Dashboard > CI_CD pipeline > Configuration

Configure Pipeline

Definition

Pipeline script

Script ?

```
1 try sample Pipeline...
```

Use Groovy Sandbox ?

Save Apply

Step 10 – Add pipeline script as SCM

- Add git repository as: https://github.com/DandaleAman/Java_app_3.0.git
- Add branch Specifier: */main
- Add Script Path: Jenkinsfile

The screenshot shows a GitHub repository page for 'Java_app_3.0'. The repository has one branch, 'main'. The code tab is selected, showing files like Jenkinsfile, Dockerfile, and README.md. The Jenkinsfile is present in the code listing.

The screenshot shows the Jenkins Pipeline configuration screen. The 'Pipeline' section is selected. Under 'Definition', 'Pipeline script from SCM' is chosen. The 'SCM' dropdown is set to 'Git'. The 'Repository URL' field contains 'https://github.com/DandaleAman/Java_app_3.0.git'. The 'Branch Specifier' field contains '/main'. The 'Script Path' field contains 'Jenkinsfile'. The 'Save' and 'Apply' buttons at the bottom are highlighted with red boxes.

Step 11 –Add the Plugins

- Follow: Dashboard -> Manage Jenkins -> Plugins -> Available plugins

The screenshot shows the Jenkins plugin manager interface. The left sidebar has a red box around the 'Available plugins' link under the 'Updates' section. The main area lists several SonarQube-related plugins:

- Sonar Gerrit (388.v9b_f1cb_e42306)
- SonarQube Scanner (2.17.2)
- SonarQube Generic Coverage (1.0)
- Sonar Quality Gates (315.v1f12b_e81a_3a_4)
- Artifactory (4.0.8)
- JFrog (1.5.3)
- Quality Gates (2.5)

A red arrow points to the 'Install' button at the top right of the list.

- Plugins for Sonar/Jfrog
 - Sonar Gerrit
 - SonarQube Scanner
 - SonarQube Generic Coverage
 - Sonar Quality Gates
 - Quality Gates
 - Artifactory
 - Jfrog

Step 12– Setup Docker

- `sudo apt update -y`
- `sudo apt install apt-transport-https ca-certificates curl software-properties-common -y`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" -y`
- `sudo apt update -y`
- `apt-cache policy docker-ce -y`
- `sudo apt install docker-ce -y`
- `sudo systemctl status docker`
- `sudo chmod 777 /var/run/docker.sock`
- `docker -v`

Step 13- Install SonarQube

- docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube

Step 13.1 - Start docker container if it's not up

- docker ps -a
- docker start <Container ID>

```
root@ip-172-31-36-240:~# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
sonarqube        latest   862025bf322c  4 days ago   1.1GB

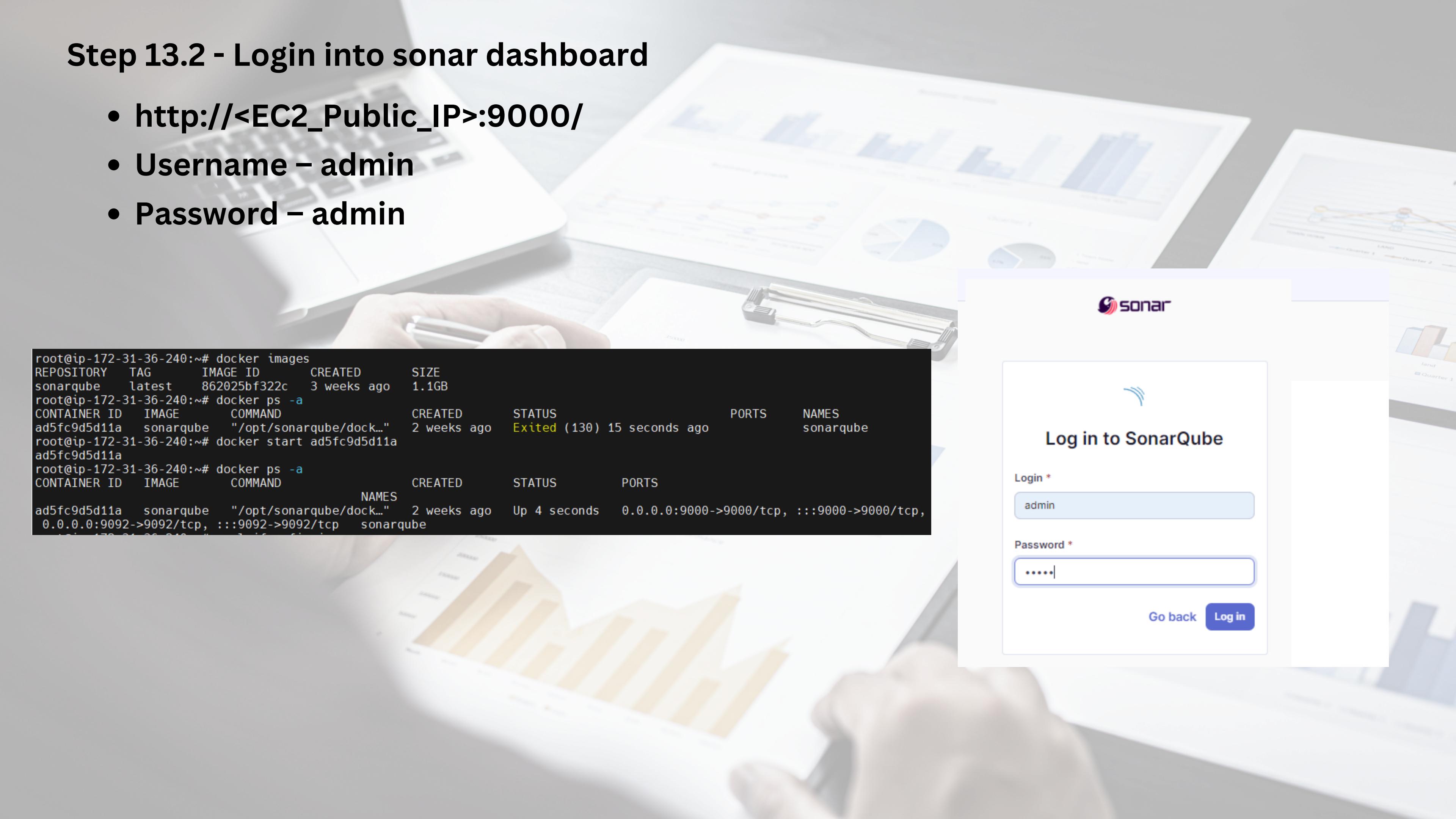
root@ip-172-31-36-240:~# docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS
NAMES
ad5fc9d5d11a      sonarqube    "/opt/sonarqube/dock.."    7 hours ago   Up 7 hours   0.0.0.0:9000->9000/tcp, :::9000->9092/tcp, 0.0.0.0:9092->9092/tcp, :::9092->9092/tcp      sonarqube

root@ip-172-31-36-240:~#
```

Step 13.2 - Login into sonar dashboard

- **http://<EC2_Public_IP>:9000/**
- **Username – admin**
- **Password – admin**

```
root@ip-172-31-36-240:~# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
sonarqube        latest   862025bf322c  3 weeks ago   1.1GB
root@ip-172-31-36-240:~# docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
ad5fc9d5d11a     sonarqube  "/opt/sonarqube/dock..."  2 weeks ago   Exited (130)  15 seconds ago
root@ip-172-31-36-240:~# docker start ad5fc9d5d11a
ad5fc9d5d11a
root@ip-172-31-36-240:~# docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
ad5fc9d5d11a     sonarqube  "/opt/sonarqube/dock..."  2 weeks ago   Up  4 seconds  0.0.0.0:9000->9000/tcp, :::9000->9000/tcp,
0.0.0.0:9092->9092/tcp, :::9092->9092/tcp   sonarqube
```



The background of the slide features a blurred photograph of a person's hands working on a laptop. The laptop screen displays various charts and graphs, suggesting a data analysis or software development environment.

 **Log in to SonarQube**

Login *

Password *

[Go back](#) [Log in](#)

Step 13.3 - Create Sonar token for Jenkins

Sonar Dashboard -> Administration -> My Account -> Security -> Create token -> Save the token to some text file

The screenshot shows the SonarQube Security page. At the top, there is a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation bar, the user profile is shown as 'Administrator'. A red box highlights the 'Security' tab in the navigation menu. The main content area is titled 'Security' and contains a paragraph about User Tokens. Below this, the 'Generate Tokens' section is displayed. It includes fields for 'Name' (with placeholder 'Enter Token Name'), 'Type' (with a dropdown menu labeled 'Select Token Type' and a red arrow pointing to it), and 'Expires in' (set to '30 days'). A 'Generate' button is also present. A table below lists existing tokens, showing one entry for 'Jenkins' with details: Type 'Global', Project 'Never', Created 'October 25, 2024', Expiration 'November 24, 2024', and a 'Revoke' button. In the top right corner of the page, a small dropdown menu is open, showing options: 'Administrator', 'My Account' (which is also highlighted with a red box), and 'Log out'.

Name	Type	Project	Last use	Created	Expiration
Jenkins	Global	Never	October 25, 2024	November 24, 2024	Revoke

Step 13.4 - Integrate Sonar to Jenkins

Sonar Dashboard -> Administration -> Configuration -> webhooks -> Add the below name and url and save

- **http://<EC2_Public_Ip>:8080/sonarqube-webhook/**

Create Webhook

Name *

URL *

Server endpoint that will receive the webhook payload, for example:
"http://my_server/foo". If HTTP Basic authentication is used, HTTPS is
recommended to avoid man in the middle attacks. Example:
"https://myLogin:myPassword@my_server/foo"

Secret

If provided, secret will be used as the key to generate the HMAC hex
(lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256'
header.

Create **Cancel**

Step 13.5– Add Sonarqube to Jenkins

Jenkins Dashboard -> Manage Jenkins -> system configuration

- Click on sonarqube servers -> add url and name -> Click on add token -> Select Secret text -> Add the sonar token from step13.3 -> Give name of token as *sonarqube-api*

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-api
Server URL	Default is http://localhost:9000 http://3.109.217.190:8080/
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled. sonarqube-api

[Jenkins](#) [Add](#) [View](#) [Provision](#)

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

- Secret text (selected)
- Username with password
- GitHub App
- SSH Username with private key
- Secret file
- Certificate

Step 14– Install Maven

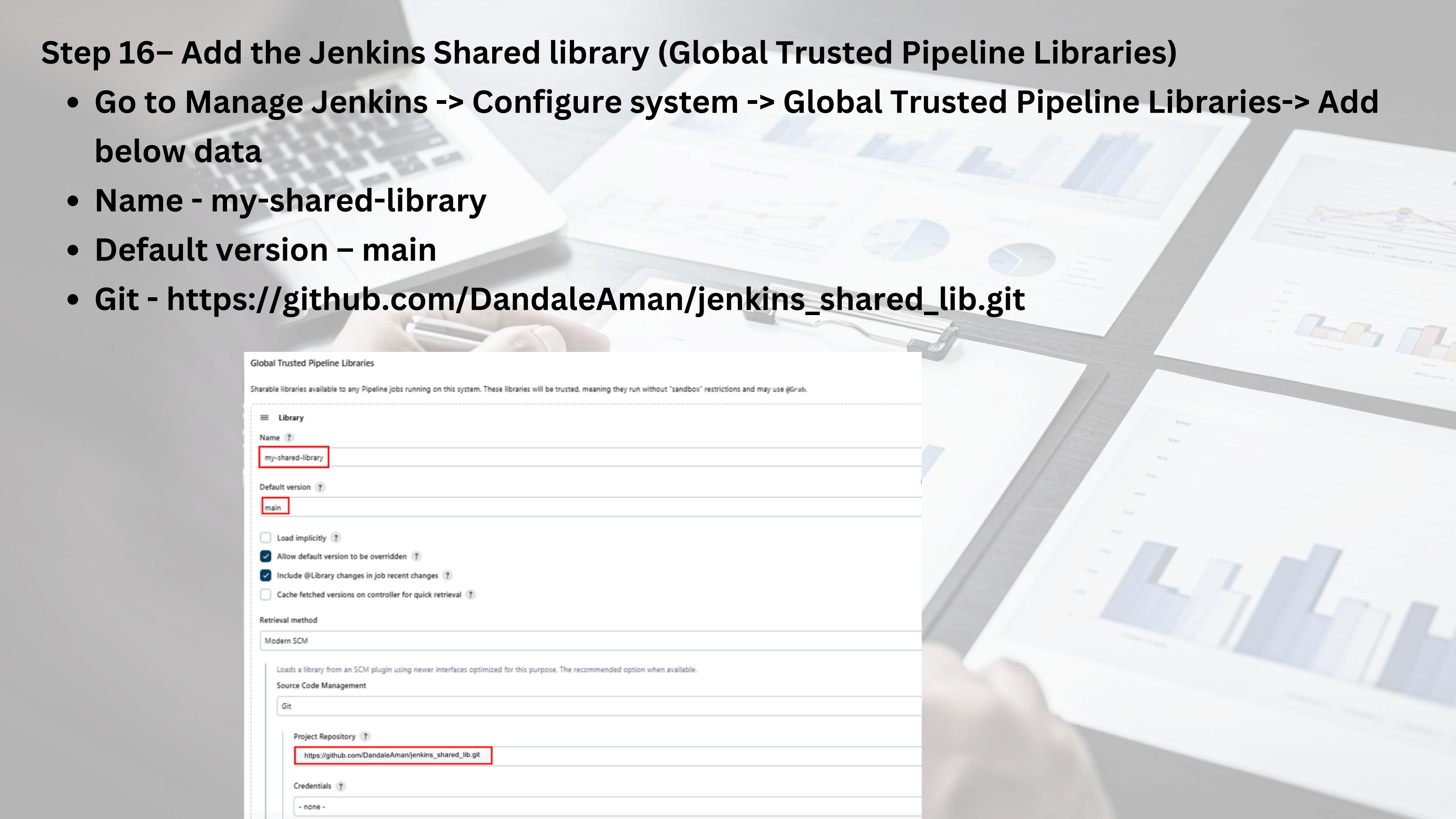
- sudo apt update -y
- sudo apt install maven -y
- mvn -version

Step 15 – Install TRIVY for docker image scan

- sudo apt-get install wget apt-transport-https gnupg lsb-release
- wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -
- echo deb https://aquasecurity.github.io/trivy-repo/deb \$(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list
- sudo apt-get update
- sudo apt-get install trivy

Step 16– Add the Jenkins Shared library (Global Trusted Pipeline Libraries)

- Go to Manage Jenkins -> Configure system -> Global Trusted Pipeline Libraries-> Add below data
- Name - my-shared-library
- Default version – main
- Git - https://github.com/DandaleAman/jenkins_shared_lib.git



The background of the slide features a blurred image of a person's hand holding a pen over a clipboard. The clipboard contains several financial charts and graphs, including bar charts and pie charts, with data labels like 'Quarter 1', 'Quarter 2', 'Quarter 3', and 'Quarter 4'.

Global Trusted Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use @Grab.

Library

Name my-shared-library

Default version main

Load implicitly ?

Allow default version to be overridden ?

Include @Library changes in job recent changes ?

Cache fetched versions on controller for quick retrieval ?

Retrieval method

Modern SCM

Loads a library from an SCM plugin using newer interfaces optimized for this purpose. The recommended option when available.

Source Code Management

Git

Project Repository https://github.com/DandaleAman/jenkins_shared_lib.git

Credentials

Step 17– Once pipeline is Run Check

- Console Output
- The Jenkins logs
- The Trivy scan vulnerabilities
- The sonarqube dashboard for report

The screenshot shows the Jenkins pipeline dashboard for the 'Jenkins CI_CD pipeline'. The 'SonarQube' section is highlighted with a red box. It displays the status of the 'minikube-sample' project as 'Passed' and 'server-side processing' as 'Success'. Below this, a 'Builds' section shows the last four builds, all of which are successful. A red arrow points to the SonarQube icon in the sidebar, with the text 'It will take you to the sonarqube Dashboard'.

The screenshot shows the Jenkins console output for build #1 of the 'Jenkins CI_CD pipeline'. The 'Console Output' section is highlighted with a red box. The log output shows the Jenkinsfile being obtained from GitHub, cloning the 'jenkins_shared_lib' repository, and initializing the workspace. The Jenkins UI sidebar on the left is also visible.

```
Started by user admin
Obtained Jenkinsfile from git https://github.com/praveen1994dec/Java_app_3.0.git
Loading library my-shared-library@main
Attempting to resolve main from remote references...
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git ls-remote -h -- https://github.com/praveen1994dec/jenkins_shared_lib.git # timeout=10
Found match: refs/heads/main revision 610e218e6018f9d7e0d3230c2b706eb02794455e
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/praveen1994dec/jenkins_shared_lib.git
> git init /var/lib/jenkins/workspace/Jenkins CI_CD
pipeline@libs/35af354d5868542ce0aa7fce00986f6c3dbdef04102e0d647598e66c5c8f2a17 # timeout=10
Fetching upstream changes from https://github.com/praveen1994dec/jenkins_shared_lib.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
```

Step 18– Sonarqube Dashboard for improve the quality of code

- SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of your project.

The image displays three screenshots of the SonarQube web application interface:

- Screenshot 1 (Top Left): SonarQube Dashboard**
 - Shows the main dashboard for the project "minikube-sample / main".
 - A green box highlights the "Quality Gate" status, which is labeled "Passed".
 - Key metrics displayed include: Security (0 Open issues), Reliability (1 Open issue), Maintainability (2 Open issues), Accepted issues (0), Coverage (0.0%), and Duplications (0.0%).
 - The dashboard also shows 130 Lines of Code and the Version 0.0.1-SNAPSHOT.
- Screenshot 2 (Top Right): SonarQube Issues**
 - Shows the "Issues" tab for the same project.
 - A red box highlights a specific issue: "Remove this field injection and use constructor injection instead." located in "src/.../sample/controller/HomeResource.java".
 - The issue details show it's a "Code Smell" of "Major" severity, introduced 1 year ago, with 5min effort required to fix.
 - The "Clean Code Attribute" section shows a single instance of "Consistency".
 - The "Software Quality" section shows a single instance of "Reliability".
- Screenshot 3 (Bottom Left): SonarQube Code Review**
 - Shows the "Issues" tab again, focusing on the specific code review for "src/.../sample/controller/HomeResource.java".
 - A red box highlights the "Where is the issue?" section, which shows the Java code snippet with the annotation "@Autowired".
 - The code snippet includes the warning: "Remove this field injection and use constructor injection instead.".
 - The code editor shows lines 17 through 29 of the HomeResource.java file.



THANK YOU



<https://www.linkedin.com/in/amandandale/>