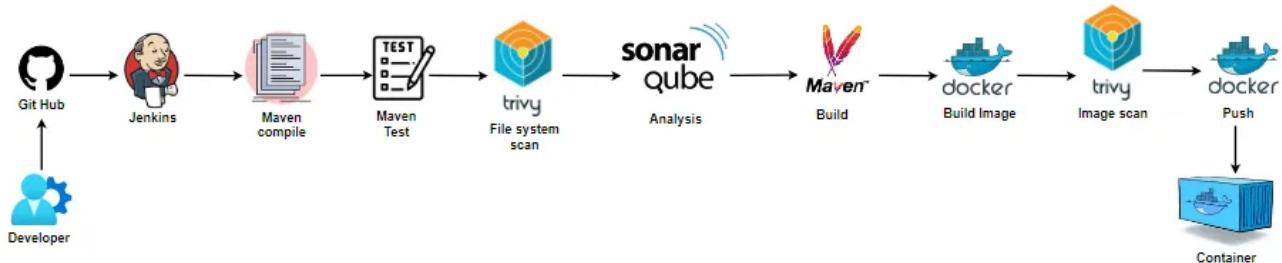


CI PIPELINE SETUP WITH VARIOUS TOOLS



Step 1: Create an EC2 Instance

1. Log in to your AWS account.
2. Navigate to **EC2 Dashboard -> Launch Instance**.
3. Choose **Ubuntu t2.medium**.
4. Allocate **30GB EBS volume**.
5. Select **US-EAST-1** as the region.
6. Configure security group rules to allow necessary ports (8080 for Jenkins, 9000 for SonarQube, etc.).

The screenshot shows the AWS EC2 'Launch an instance' wizard. The current step is 'Step 1 of 7: Set instance details'. The 'Name and tags' section has 'Name' set to 'jenkins-project'. The 'Application and OS Images (Amazon Machine Image)' section shows 'Amazon Linux 2023 AMI 2023.6.2...' selected. The 'Virtual server type (instance type)' is 't2.micro'. The 'Storage (volumes)' section shows '1 volume(s) - 8 GiB'. The 'Summary' section shows 'Number of instances' as 1. The 'Launch instance' button is highlighted in orange at the bottom right.

Network settings

VPC - required [Info](#)
vpc-0369b0461f6d73d06 (default)

Subnet [Info](#)
subnet-0c1a273a2d2f73c97 VPC: vpc-0369b0461f6d73d06 Owner: 982534386480 Availability Zone: us-east-1a Zone type: Availability Zone IP addresses available: 4091 CIDR: 172.31.0.0/20

Create new subnet [Create new subnet](#)

Auto-assign public IP [Info](#)
Enable Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required
launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-~/[@#]=;&:[]\$*

Description - required [Info](#)
launch-wizard-1 created 2025-03-27T03:36:39.818Z

Summary

Number of instances [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64... [read more](#)
ami-084568db4383264da

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

EC2

Instances (1) [Info](#)

Last updated less than a minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) [Clear filters](#)

Instance ID = i-0d5b8767659e3f952 All states

<input type="checkbox"/>	Name D	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	jenkins-project	i-0d5b8767659e3f952	Running Q Q	t2.medium	Initializing	View alarms +	us-east-1a	ec2-44-193-219-155.co.

Select an instance

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 2: Connect to EC2 and Install Required Tools

1. Connect to the instance using SSH:

```
ssh -i <your-key.pem> ubuntu@<EC2 PUBLIC IP>
```

2. Switch to root user:

```
sudo -i
```

Step 3: Install Jenkins

Run the following script to install Jenkins:

```
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
```

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
```

```
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
```

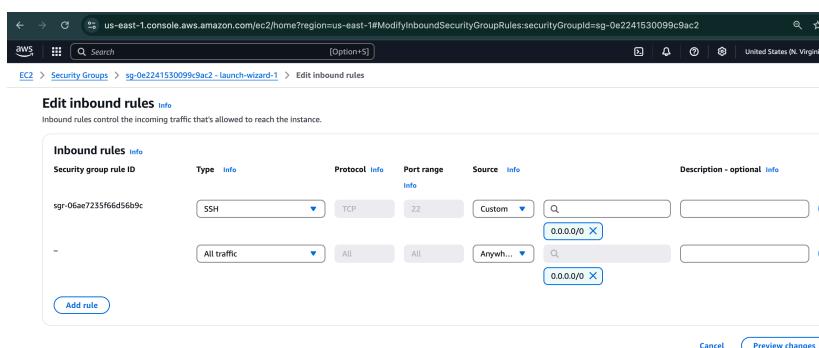
```
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
sudo apt-get update -y
```

```
sudo apt-get install jenkins -y
```

Step 4: Update EC2 Security Group

- Add an **inbound rule** to allow **port 8080** (for Jenkins access).
- Save the changes.



Step 5: Access Jenkins Dashboard

1. Open a browser and enter:

```
http://<EC2_PUBLIC_IP>:8080/
```

2. Retrieve the administrator password:

```
cat /var/lib/jenkins/secrets/initialAdminPassword
```

3. Install all suggested plugins.

4. Create the first admin user.

Unlock Jenkins

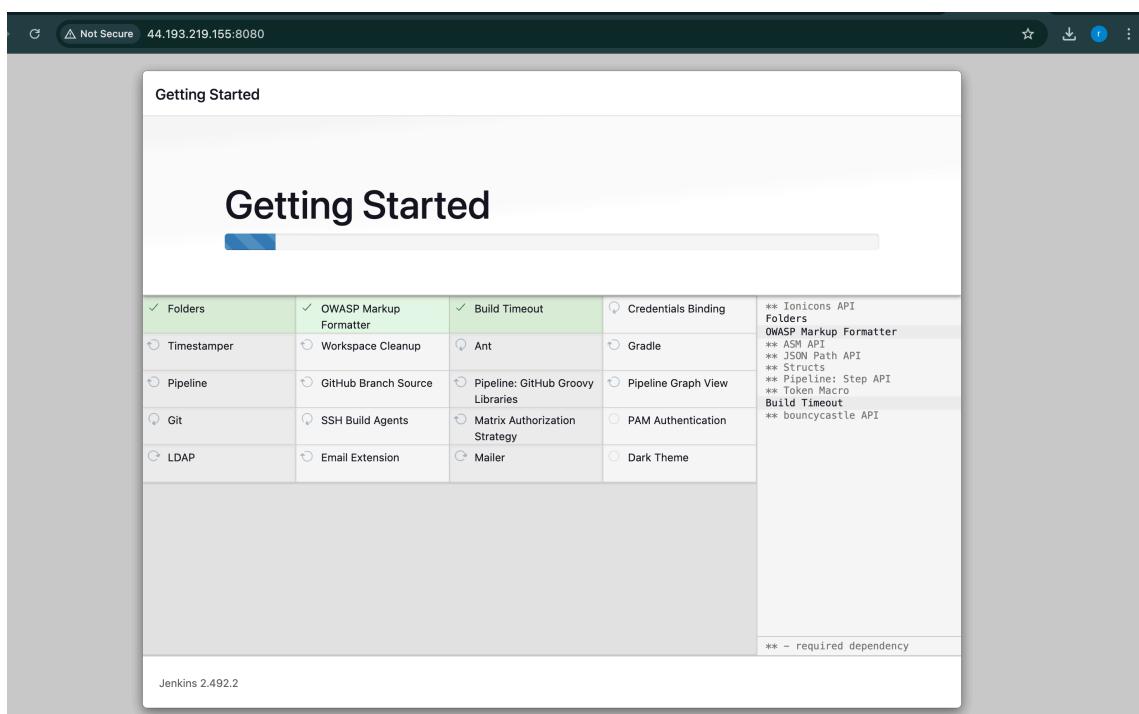
To ensure Jenkins is securely set up by the administrator, a password is in the log (**not sure where to find it?**) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

```
.....
```



The screenshot shows the Jenkins dashboard. At the top, there's a header with the Jenkins logo, a search bar, and user information (raja). Below the header, the left sidebar contains links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The main content area has a title 'Welcome to Jenkins!' and a message: 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' It features a section titled 'Start building your software project' with a 'Create a job' button and a '+' icon. Below this, there's a 'Set up a distributed build' section with three items: 'Set up an agent' (with a monitor icon), 'Configure a cloud' (with a cloud icon), and 'Learn more about distributed builds' (with a question mark icon). On the far right of the dashboard, there are links for 'REST API' and 'Jenkins 2.492.2'.

Step 6: Setup CI/CD Pipeline in Jenkins

1. Create a New Pipeline Job

2. Use SCM for Pipeline Script:

https://github.com/venkatraja1234/Java_app_3.0.git

3. Install Required Plugins:

- SonarQube Scanner
- Quality Gates
- Artifactory
- JFrog
- Docker



New Item

Enter an item name

first-project

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 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**

OK



Plugins

jfrog

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Install Name ↓

Released

**Sonar Gerrit** 388.v9b_f1cb_e42306

External Site/Tool Integrations

9 mo 25 days ago

This plugin allows to submit issues from SonarQube to Gerrit as comments directly.

**SonarQube Scanner** 2.18

External Site/Tool Integrations Build Reports

1 mo 27 days ago

This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.

**SonarQube Generic Coverage** 1.0

TODO

5 yr 8 mo ago

**Sonar Quality Gates** 352.vdccb_d7994fb_6

Library plugins (for use by other plugins) analysis Other Post-Build Actions

1 mo 2 days ago

Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")

**Quality Gates** 2.5

Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")

Warning: This plugin version may not be safe to use. Please review the following security notices:

[Credentials transmitted in plain text](#)

8 yr 10 mo ago

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
commons-lang3 v3.x Jenkins API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

Step 7: Install Docker

Run the following script to install 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 chmod 777 /var/run/docker.sock
```

Verify installation:

```
docker -v
```

```
Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.
```

```
Restarting services...
```

```
Service restarts being deferred:  
/etc/needrestart/restart.d/dbus.service  
systemctl restart networkd-dispatcher.service  
systemctl restart systemd-logind.service  
systemctl restart unattended-upgrades.service
```

```
No containers need to be restarted.
```

```
User sessions running outdated binaries:  
ubuntu @ session #2: sshd[1115]  
ubuntu @ user manager service: systemd[1120]
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

```
root@ip-172-31-14-134:~# chmod 777 /var/run/docker.sock  
root@ip-172-31-14-134:~# docker -v  
Docker version 24.0.2, build cb74dfc  
root@ip-172-31-14-134:~# ps  
  PID TTY      TIME CMD  
18089 pts/3    00:00:00 sudo  
18090 pts/3    00:00:00 bash  
20382 pts/3    00:00:00 ps  
root@ip-172-31-14-134:~# docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube  
Unable to find image 'sonarqube:latest' locally  
latest: Pulling from library/sonarqube  
5a7813e071bf: Pull complete  
dbe46403441a: Pull complete  
f9f4ee04af87: Pull complete  
e3da94a33fa1: Pull complete  
f03e4717322c: Pull complete  
c9439e8e4945: Download complete  
62f1017e9142: Download complete  
4f4fb700ef54: Download complete
```

```
root@ip-172-31-14-134:~# docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube  
Unable to find image 'sonarqube:latest' locally  
latest: Pulling from library/sonarqube  
5a7813e071bf: Pull complete  
dbe46403441a: Pull complete  
f9f4ee04af87: Pull complete  
e3da94a33fa1: Pull complete  
f03e4717322c: Pull complete  
c9439e8e4945: Pull complete  
62f1017e9142: Pull complete  
4f4fb700ef54: Pull complete  
Digest: sha256:c0a734bd7e62c1a35794e3a070b4945f5a081b0053201eb926bcb936b0e5d2e6  
Status: Downloaded newer image for sonarqube:latest  
76fc760871bf4e5749608d9bf8f5b499553fbc48233ea227cc289026c895d40a
```

```
root@ip-172-31-14-134:~# $docker ps -a  
ps-a: command not found  
root@ip-172-31-14-134:~# $docker ps -a  
  PID TTY      TIME CMD  
1275 pts/0    00:00:00 sudo  
1277 pts/1    00:00:00 bash  
18088 pts/2    00:00:00 sudo  
18090 pts/3    00:00:00 bash  
20854 pts/3    00:00:00 ps  
root@ip-172-31-14-134:~# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
76fc760871bf	sonarqube	"/opt/sonarqube/dock..."	2 minutes ago	Up 2 minutes	0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 0.0.0.0:9092->9092/tcp, :::9092->9092/tcp

```
root@ip-172-31-14-134:~# docker start  
"docker start" requires at least 1 argument.  
See 'docker start --help'.
```

```
Usage: docker start [OPTIONS] CONTAINER [CONTAINER...]
```

```
Start one or more stopped containers  
root@ip-172-31-14-134:~# docker start 76fc760871bf  
76fc760871bf  
root@ip-172-31-14-134:~#
```

Step 8: Install and Configure SonarQube

Run the installation script:

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

Start SonarQube if it's not running:

```
docker ps -a # Get container ID
```

```
docker start <containerID>
```

Access SonarQube:

http://<EC2_PUBLIC_IP>:9000/

Login Credentials:

- Username: **admin**
- Password: **admin**

The screenshot shows a web browser window with the URL `http://44.193.219.155:9000/account/reset_password`. The page title is "Update your password". A warning message in a yellow box states: "⚠ This account should not use the default password." Below this, there are three input fields: "Old Password *", "Password *", and "Confirm Password *". Each field has a green border and a small green checkmark icon on its right side. The "Update" button at the bottom is blue.

Not Secure 44.193.219.155:9000/projects/create

SonarQube community Projects Issues Rules Quality Profiles Quality Gates Administration More ▾

⚠️ Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. [Learn more](#)

How do you want to create your project?

Do you want to benefit from all of SonarQube Community Build's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

Import from Azure DevOps Setup

Import from Bitbucket Cloud Setup

Import from Bitbucket Server Setup

Import from GitHub Setup

Import from GitLab Setup

Are you just testing or have an advanced use-case? Create a local project.

[Create a local project](#)

Generate a token:

1. Go to **Administration -> My Account -> Security**.
2. Click **Create Token**.
3. Save the token for Jenkins integration.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More ? A

A Administrator Profile Security Notifications Projects

Tokens

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.

Name	Type	Expires in
Jenkins1	Global Analysis Token	30 days

Name	Type	Project	Last use	Created	Expiration	Actions

Step 9: Integrate SonarQube with Jenkins

1. Go to Jenkins Dashboard -> Manage Jenkins -> Configure System.
2. Locate **SonarQube servers**.
3. Add SonarQube URL and Token (generated in Step 8).
4. Configure **webhook**:

http://<EC2_IP>:8080/sonarqube-webhook/

The screenshot shows the Jenkins configuration interface. Under the 'SonarQube Scanner installations' section, there is a form to add a new scanner named 'Sonar'. The 'Install automatically' checkbox is checked. Under 'Install from Maven Central', the version is set to 'SonarQube Scanner 5.0.1.3006'. There is also an 'Add Installer' button.

The screenshot shows the SonarQube administration interface under 'Webhooks'. A 'Create Webhook' dialog is open, prompting for a 'Name' (set to 'jenkins') and a 'URL' (set to 'http://44.193.219.155:8080/sonarqube-webhook/'). The dialog also includes a 'Secret' field and 'Create' and 'Cancel' buttons. The background shows the SonarQube navigation bar and some status information.

Step 10: Install Maven

Run the following script:

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

Step 11: Install Trivy for Security Scanning

Run the script:

```
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 12: Configure Jenkins Credentials

12.1 Add Docker Hub Credentials

1. Go to Jenkins Dashboard -> Manage Jenkins -> Credentials -> System -> Global Credentials.
2. Add Docker Hub credentials with ID: docker.

The screenshot shows the Jenkins Manage Jenkins dashboard. The top navigation bar has 'Dashboard' and 'Manage Jenkins'. The 'Manage Jenkins' option is highlighted in the sidebar. The main content area is titled 'Manage Jenkins' and contains several sections: 'System Configuration' (with 'System' and 'Nodes' options), 'Security' (with 'Security' option), and 'Tools' (with 'Tools' and 'Clouds' options). On the left, there are collapsed sections for 'Build Queue', 'Build Executor Status' (showing 0 of 2 executors busy), and 'Cloud Statistics'.

Credentials

The screenshot shows the Jenkins 'Credentials' page. At the top, there are tabs for 'T' (Text), 'P' (Password), 'Store' (selected), 'Domain', 'ID', and 'Name'. Below this, a section titled 'Stores scoped to Jenkins' lists 'System' and '(global)' under 'Domains'. A 'System' dropdown is open. At the bottom, there's a 'System' section with a 'Domain' dropdown set to 'Global credentials (unrestricted)', a description about unrestricted credentials, and icons for S, M, and L.

12.2 Add Jenkins Shared Library

1. Navigate to **Manage Jenkins -> Configure System -> Global Pipeline Library**.
2. Add:
 - o **Name:** my-shared-library
 - o **Default Version:** main
 - o **Git URL:** https://github.com/venkatraja1234/jenkins_shared_lib.git

Dashboard > Manage Jenkins > System >

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.

Add

Global Untrusted Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be untrusted, meaning they run with "sandbox" restrictions and cannot use @Grab.

Add

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

Cannot validate default version until after saving and reconfiguring.

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

Modern SCM

Legacy SCM

Step 13: Verify the CI/CD Pipeline

After running the pipeline, verify the following:

- Jenkins logs for build errors.
- **Trivy Scan** results for vulnerabilities.
- **SonarQube Dashboard** for code quality reports.

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

minikube-sample / main ?

The last analysis has warnings. See details Version 0.0.1-SNAPSHOT

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Quality Gate Status [?](#)

Quality Gate Passed

Enjoy your sparkling clean code!

Measures

New Code Overall Code

Reliability 0 Bugs A	Maintainability 1 Code Smells A
Security 0 Vulnerabilities A	Security Review 0 Security Hotspots A
Coverage 0.0% Coverage	Duplications 0.0% Duplications