

Zomato Clone App Deployment DevOps project

docker & K8S deployment on ubuntu host machine

<https://github.com/MaheshBabu-DevOPs>

1. Docker Deployment Pipeline

Tools used:

Git, GitHub (Source control)

Jenkins (CI/CD server)

Docker & Docker Hub (Containerization & registry)

SonarQube (Code quality analysis)

Trivy & Docker Scout (Container security scanning)

OWASP Dependency Check (Vulnerability scanning for dependencies)

Gmail (for email notifications)

This pipeline builds, scans, and deploys Docker containers on the Ubuntu host.

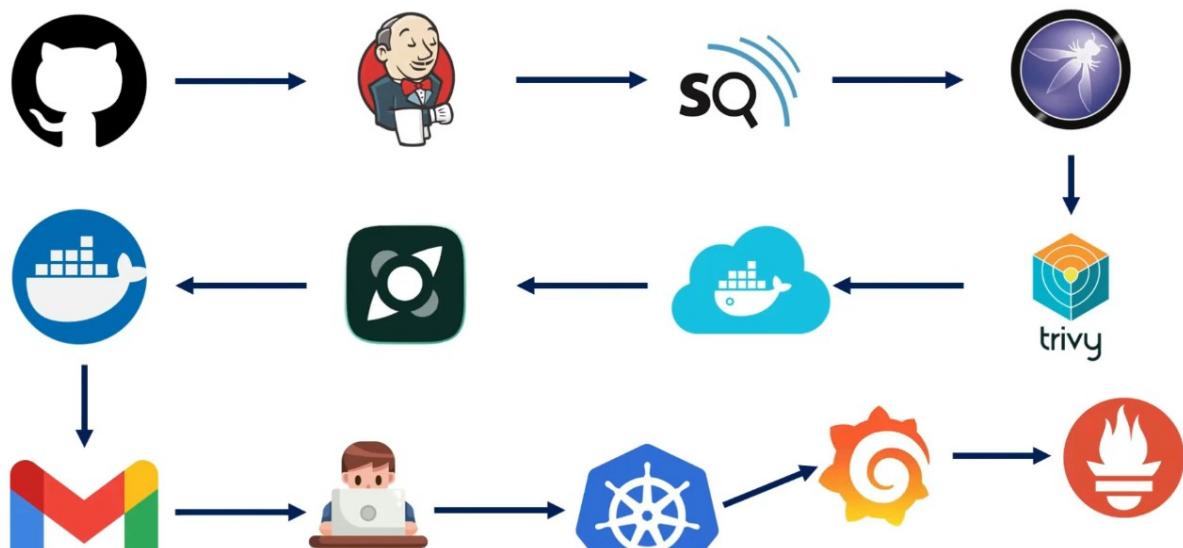
2. Kubernetes Deployment

Tools used:

Kubernetes (Container orchestration)

Prometheus (Monitoring)

Grafana (Visualization/dashboard)



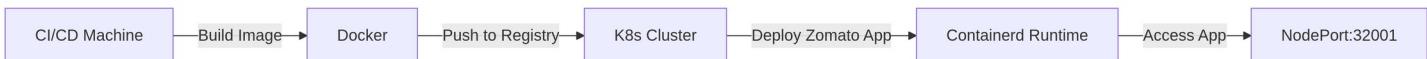


This screenshot shows a documentation page for Jenkins on Linux. The URL is jenkins.io/doc/book/installing/linux/#debianubuntu. The page has a sidebar with "User Documentation Home" and a list of topics under "User Handbook". The main content discusses the choice between "Long Term Support release" and "Weekly release". It explains that an LTS release is chosen every 12 weeks from the stream of regular releases as the stable release for that time period. It can be installed from the [debian-stable apt repository](#). Below this, a terminal command for the LTS release is shown:

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/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
sudo apt-get install jenkins
```

The page also mentions that a new release is produced weekly to deliver bug fixes and features to users and plugin developers. It can be installed from the [debian apt repository](#). Below this, another terminal command for the weekly release is shown:

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian/jenkins.io-2023.key
```



```
dell@mahesh:~$ sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/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
sudo apt-get install jenkins
[sudo] password for dell:
--2025-08-10 15:20:25-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 2a04:4e42:59::645, 151.101.158.133
Connecting to pkg.jenkins.io (pkg.jenkins.io)|2a04:4e42:59::645|:443... connected.
HTTP request sent, awaiting response...
```

```
dell@mahesh:~$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.5 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.5 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
dell@mahesh:~$
```

```

dell@mahesh: ~
dell@mahesh: ~ 122x27
dell@mahesh:~$ sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/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
sudo apt-get install jenkins
--2025-08-10 15:21:37-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 2a04:4e42:25::645, 151.101.158.133
Connecting to pkg.jenkins.io (pkg.jenkins.io)|2a04:4e42:25::645|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/etc/apt/keyrings/jenkins-keyring.asc'

/etc/apt/keyrings/jenkins-keyr 100%[=====] 3.10K --.-KB/s in 0s

2025-08-10 15:21:38 (9.82 MB/s) - '/etc/apt/keyrings/jenkins-keyring.asc' saved [3175/3175]

0% [Working]

```

(2) DevOps Real-time Projects x Linux x +

jenkins.io/doc/book/installing/linux/#debianubuntu

Jenkins cd Blog Success Stories Contributor Spotlight Documentation Plugins Community Subprojects Security About Download Search

> User Documentation Home

User Handbook

- User Handbook Overview
- Installing Jenkins**
 - Docker
 - Kubernetes
 - Linux**
 - macOS
 - Windows
 - Other Systems
 - WAR file
 - Other Servlet Containers
 - Offline Installations
 - Initial Settings
- Platform Information
- Using Jenkins
- Pipeline
- Blue Ocean
- Managing Jenkins
- Securing Jenkins
- System Administration

Here, "8081" was chosen but you can put another port available.

Installation of Java

Jenkins requires Java to run, yet not all Linux distributions include Java by default. Additionally, [not all Java versions are compatible](#) with Jenkins.

There are multiple Java implementations which you can use. [OpenJDK](#) is the most popular one at the moment, we will use it in this guide.

Update the Debian apt repositories, install OpenJDK 21, and check the installation with the commands:

```

sudo apt update
sudo apt install fontconfig openjdk-21-jre
java -version
openjdk version "21.0.3" 2024-04-16
OpenJDK Runtime Environment (build 21.0.3+11-Debian-2)
OpenJDK 64-Bit Server VM (build 21.0.3+11-Debian-2, mixed mode, sharing)

```

BASH | ⓘ

Why use [apt](#) and not [apt-get](#) or another command? The apt command has been available since 2014. It has a command structure that is similar to [apt-get](#) but was created to be a more pleasant experience for typical users. Simple software management tasks like install, search and remove are easier with [ant](#).

```
dell@mahesh: ~
dell@mahesh: ~ 122x27
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-21-jre-headless amd64 21.0.8+9~us1-0ubuntu1~22.04.1 [46.8 MB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ca-certificates-java all 20190909ubuntu1.2 [12.1 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2,041 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java-jni amd64 0.38.0-5build1 [49.0 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-21-jre amd64 21.0.8+9~us1-0ubuntu1~22.04.1 [234 kB]
Fetched 49.4 MB in 13s (3,909 kB/s)
Selecting previously unselected package fontconfig.
(Reading database ... 269681 files and directories currently installed.)
Preparing to unpack .../0-fontconfig_2.13.1-4.2ubuntu5_amd64.deb ...
Unpacking fontconfig (2.13.1-4.2ubuntu5) over (2.13.1-4.2ubuntu5) ...
Selecting previously unselected package java-common.
Preparing to unpack .../1-java-common_0.72build2_all.deb ...
Unpacking java-common (0.72build2) ...
Selecting previously unselected package openjdk-21-jre-headless:amd64.
Preparing to unpack .../2-openjdk-21-jre-headless_21.0.8+9~us1-0ubuntu1~22.04.1_amd64.deb ...
Unpacking openjdk-21-jre-headless:amd64 (21.0.8+9~us1-0ubuntu1~22.04.1) ...
Selecting previously unselected package ca-certificates-java.
Preparing to unpack .../3-ca-certificates-java_20190909ubuntu1.2_all.deb ...
Unpacking ca-certificates-java (20190909ubuntu1.2) ...
Selecting previously unselected package fonts-dejavu-extra.
Preparing to unpack .../4-fonts-dejavu-extra_2.37-2build1_all.deb ...
Unpacking fonts-dejavu-extra (2.37-2build1) ...

Progress: [ 31%] [ #####.....]
```

```
dell@mahesh: ~
dell@mahesh: ~ 122x27
dell@mahesh:~$ jenkins --version
2.516.1
dell@mahesh:~$ java --version
openjdk 21.0.8 2025-07-15
OpenJDK Runtime Environment (build 21.0.8+9-Ubuntu-0ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu122.04.1, mixed mode, sharing)
dell@mahesh:~$ git --version
git version 2.34.1
dell@mahesh:~$
```

DevOps Real-time Project

Linux

jenkins.io/doc/book/installing/linux/#debianubuntu

Jenkins

User Documentation Home

User Handbook

- User Handbook Overview
- Installing Jenkins
 - Docker
 - Kubernetes
 - Linux
 - macOS
 - Windows
 - Other Systems
 - WAR file
 - Other Servlet Containers
 - Offline Installations
 - Initial Settings
- Platform Information
- Using Jenkins
- Pipeline
- Blue Ocean
- Managing Jenkins
- Securing Jenkins
- System Administration

sudo dnf install jenkins

Start Jenkins

You can enable the Jenkins service to start at boot with the command:

```
sudo systemctl enable jenkins
```

You can start the Jenkins service with the command:

```
sudo systemctl start jenkins
```

You can check the status of the Jenkins service using the command:

```
sudo systemctl status jenkins
```

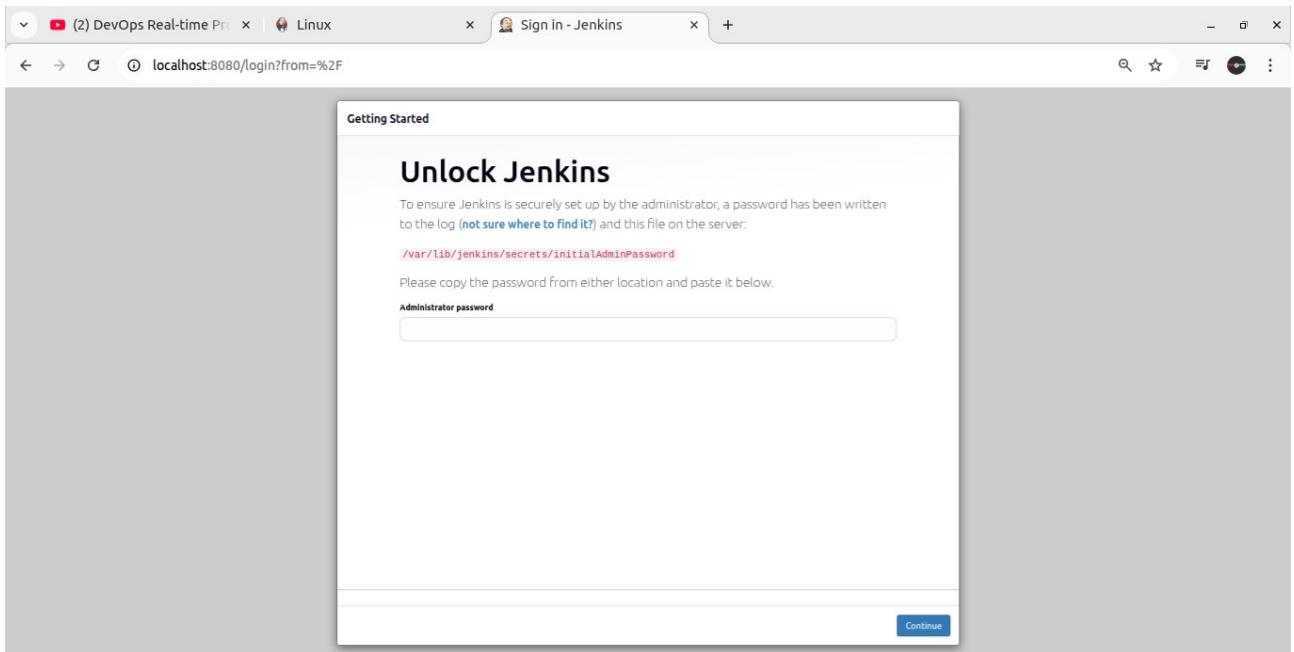
If everything has been set up correctly, you should see an output like this:

```
Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
Active: active (running) since Tue 2018-11-13 16:19:01 +03; 4min 57s ago
```

```
dell@mahesh:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
dell@mahesh:~$ sudo systemctl start jenkins
dell@mahesh:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
    Active: active (running) since Sun 2025-08-10 15:24:40 IST; 3s ago
      Main PID: 12159 (java)
         Tasks: 48 (limit: 14096)
        Memory: 647.6M
          CPU: 1min 5.786s
        CGroup: /system.slice/jenkins.service
                └─12159 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins

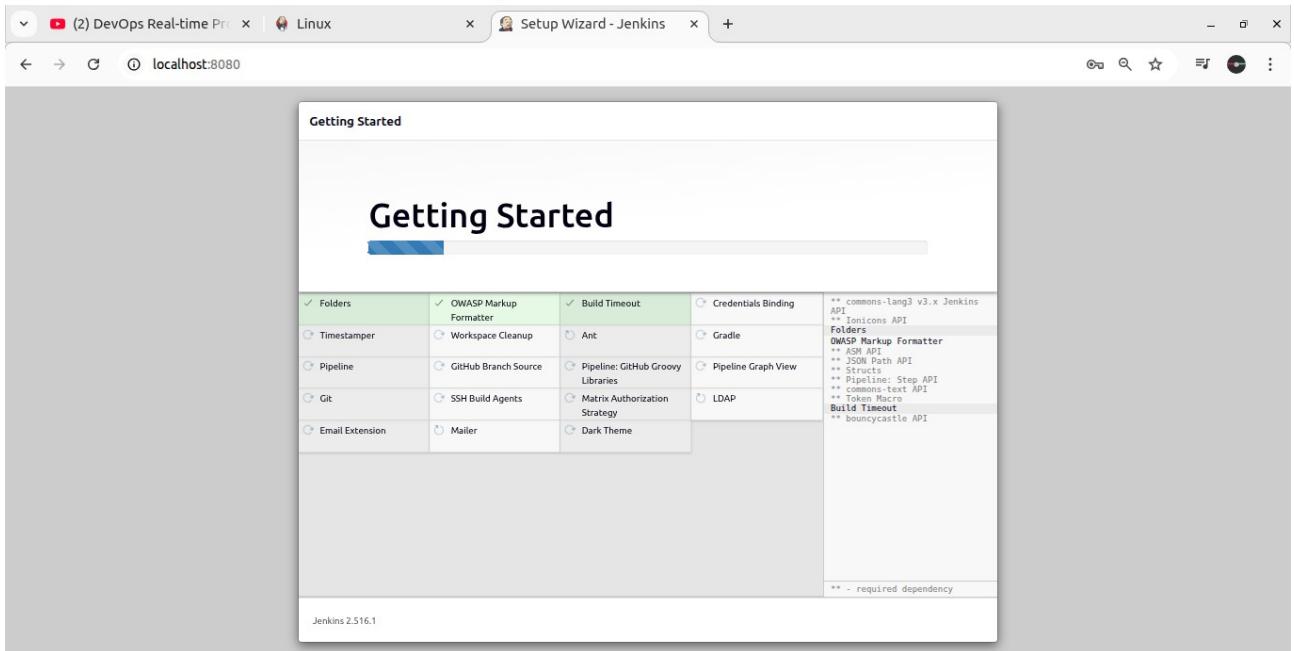
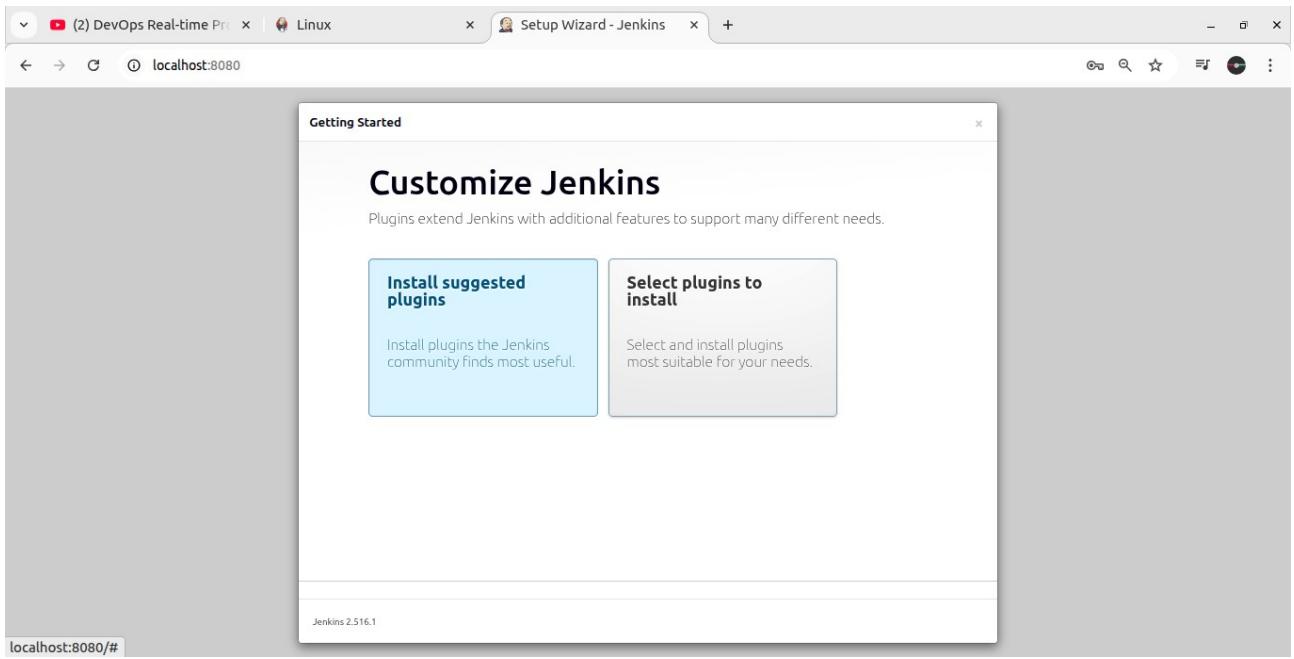
Aug 10 15:24:30 mahesh jenkins[12159]: c6ee7fd7df7946ec85bf5bbcbba239be
Aug 10 15:24:30 mahesh jenkins[12159]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Aug 10 15:24:30 mahesh jenkins[12159]: ****
Aug 10 15:24:30 mahesh jenkins[12159]: ****
Aug 10 15:24:30 mahesh jenkins[12159]: ****
Aug 10 15:24:40 mahesh jenkins[12159]: 2025-08-10 09:54:40.365+0000 [id=47]           INFO    jenkins.InitReactorRunner$>
Aug 10 15:24:40 mahesh jenkins[12159]: 2025-08-10 09:54:40.394+0000 [id=33]           INFO    hudson.lifecycle.Lifecycle$>
Aug 10 15:24:40 mahesh systemd[1]: Started Jenkins Continuous Integration Server.
Aug 10 15:24:42 mahesh jenkins[12159]: 2025-08-10 09:54:42.330+0000 [id=63]           INFO    h.m.DownloadService$Download$>
Aug 10 15:24:42 mahesh jenkins[12159]: 2025-08-10 09:54:42.331+0000 [id=63]           INFO    hudson.util.Retrier#start:>
```

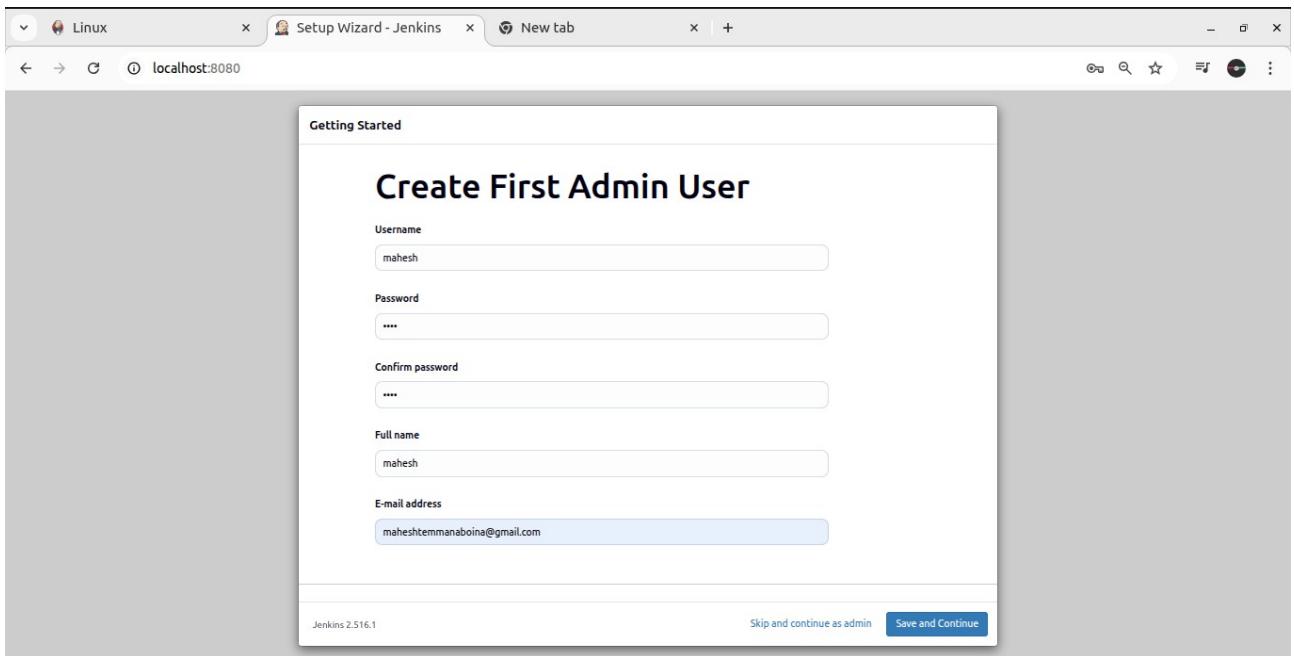
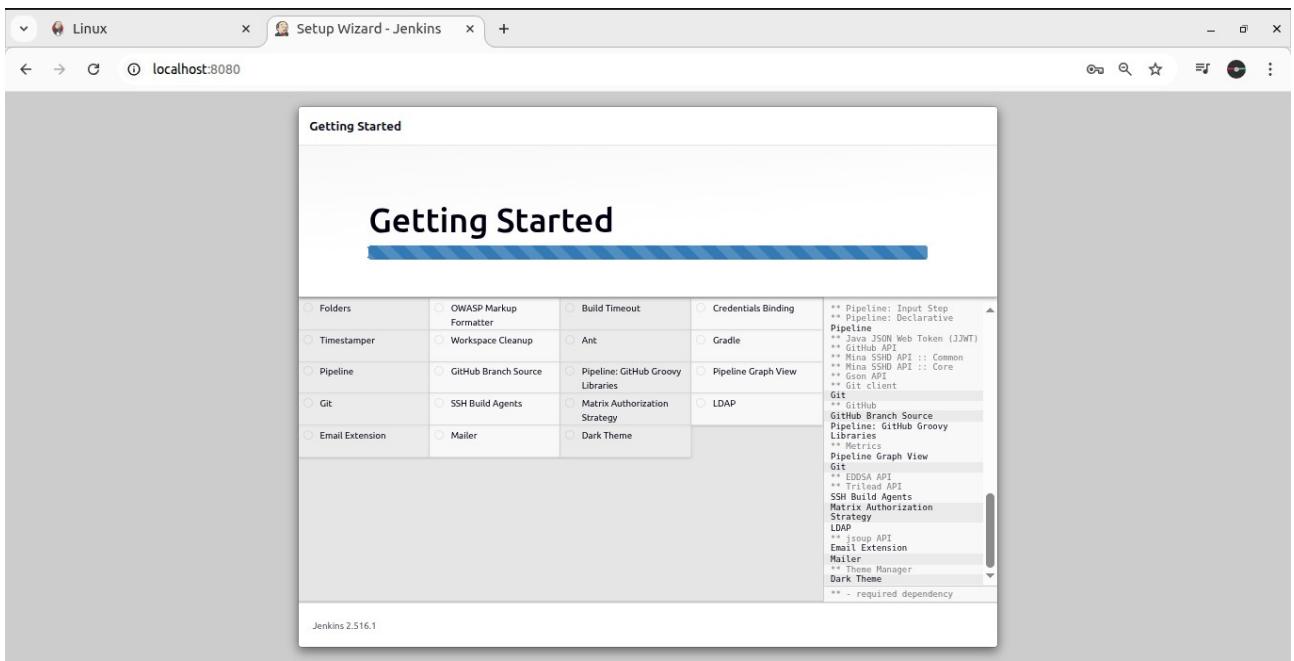
dell@mahesh:~\$

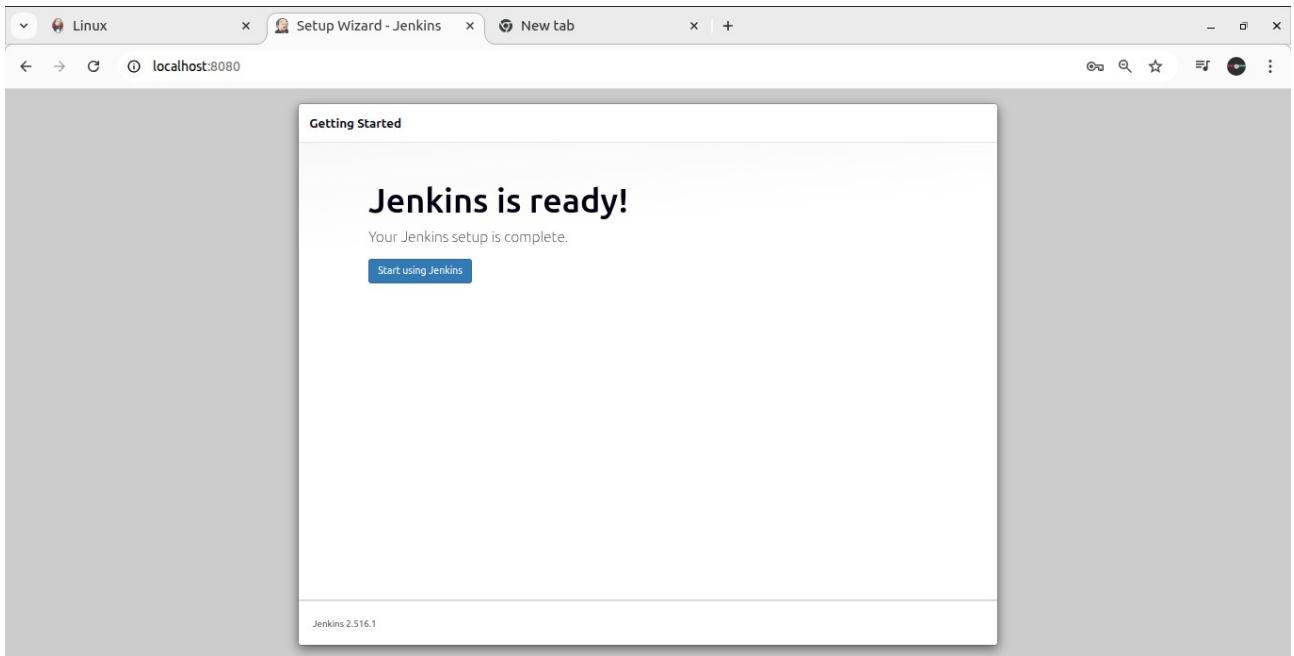
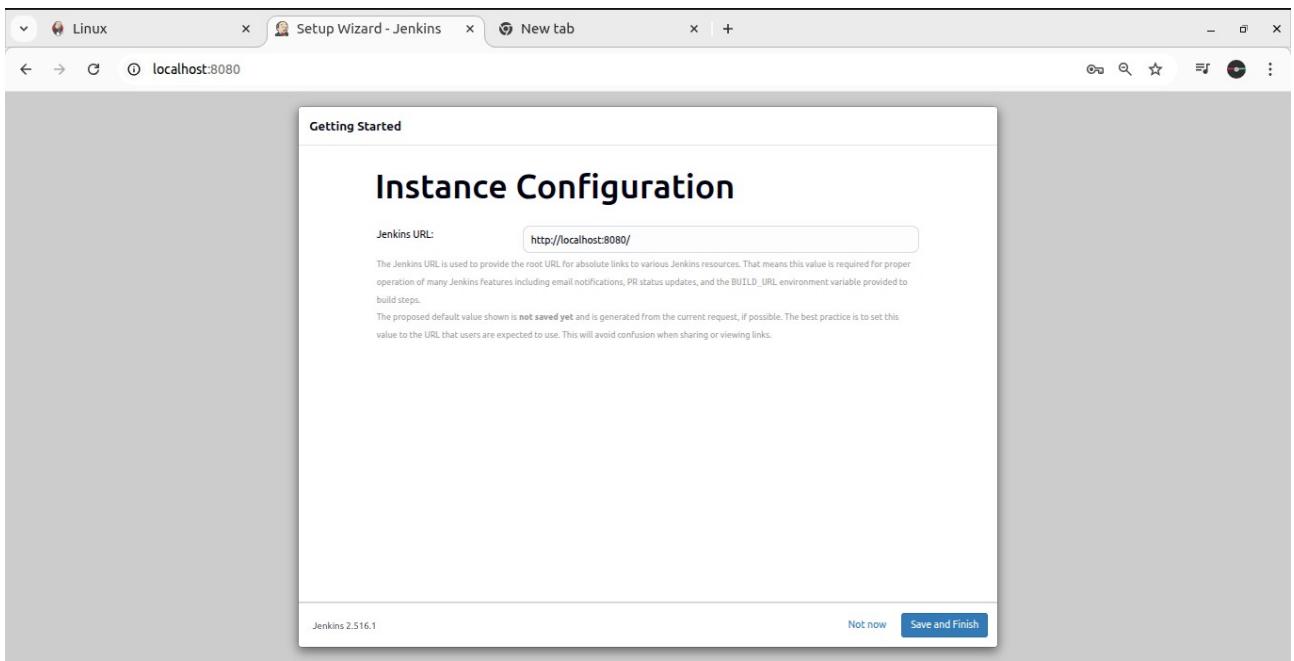


```
dell@mahesh:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2025-08-10 15:24:40 IST; 43s ago
    Main PID: 12159 (java)
      Tasks: 51 (limit: 14096)
     Memory: 648.6M
        CPU: 1min 6.360s
       CGroup: /system.slice/jenkins.service
               └─12159 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins

Aug 10 15:24:30 mahesh jenkins[12159]: c6ee7fd7df7946ec85bf5bbcba239be
Aug 10 15:24:30 mahesh jenkins[12159]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Aug 10 15:24:30 mahesh jenkins[12159]: ****
Aug 10 15:24:30 mahesh jenkins[12159]: ****
Aug 10 15:24:30 mahesh jenkins[12159]: ****
Aug 10 15:24:40 mahesh jenkins[12159]: 2025-08-10 09:54:40.365+0000 [id=47]           INFO      jenkins.InitReactorRunner$1@12159
Aug 10 15:24:40 mahesh jenkins[12159]: 2025-08-10 09:54:40.394+0000 [id=33]           INFO      hudson.lifecycle.Lifecycle$1@12159
Aug 10 15:24:40 mahesh systemd[1]: Started Jenkins Continuous Integration Server.
Aug 10 15:24:42 mahesh jenkins[12159]: 2025-08-10 09:54:42.330+0000 [id=63]           INFO      h.m.DownloadService$DownloadJob@12159
Aug 10 15:24:42 mahesh jenkins[12159]: 2025-08-10 09:54:42.331+0000 [id=63]           INFO      hudson.util.Retriger#start:@12159
dell@mahesh:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
c6ee7fd7df7946ec85bf5bbcba239be
```







The screenshot shows the Jenkins dashboard at `localhost:8080`. The top navigation bar includes links for 'New Item', 'Build History', 'Add description', search, and other system icons. The main content area features the 'Welcome to Jenkins!' message, a note about the build queue, and a 'Start building your software project' section. It also includes sections for 'Set up a distributed build' and links for 'Create a job', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. At the bottom right, there are links for 'REST API' and 'Jenkins 2.516.1'.

This screenshot is identical to the one above, showing the Jenkins dashboard at `localhost:8080`. The only difference is the tab title, which now says 'Linux' instead of 'Dashboard - Jenkins'. The rest of the interface, including the 'Welcome to Jenkins!' message, build queue status, build executor status, and various configuration links, remains the same.

#Docker

The screenshot shows a web browser window with the URL docs.docker.com/engine/install/debian/. The page is titled '#Docker' and features a navigation bar with links to 'Get started', 'Guides', 'Manuals', 'Reference', 'Search', and 'Ask AI'. On the left, there's a sidebar with a tree view of 'OPEN SOURCE' sections: 'Docker Engine' (selected), 'Install' (selected), 'Ubuntu' (selected), 'Debian' (selected), 'RHEL', 'Fedora', 'Raspberry Pi OS (32-bit)', 'CentOS', 'SLES (s390x)', 'Binaries', 'Post-installation steps', 'Storage', 'Networking', 'Containers', 'CLI', 'Daemon', and '...'. The main content area contains instructions for setting up Docker's apt repository. It includes a numbered list '1. Set up Docker's apt repository.' followed by a code block:

```
# Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl  
sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
# Add the repository to Apt sources:  
echo \  
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian  
  \_ $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

Below the code block is a note icon with the text 'Note'.

On the right side, there's a 'Table of contents' sidebar with links to 'Prerequisites', 'Firewall limitations', 'OS requirements', 'Uninstall old versions', 'Installation methods', 'Install using the apt repository' (which is selected), 'Upgrade Docker Engine', 'Install from a package', 'Upgrade Docker Engine', 'Install using the convenience script', 'Install pre-releases', 'Upgrade Docker after using the convenience script', 'Uninstall Docker Engine', and 'Next steps'. There are also 'Edit this page', 'Request changes', and 'Give feedback' buttons.

The screenshot shows a terminal window with a black background and white text. The prompt is 'dell@mahesh:~\$'. The user has run the following commands:

```
dell@mahesh:~$ # Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl  
sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
# Add the repository to Apt sources:  
echo \  
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian  
  \_ $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

Output from the 'apt-get update' command is shown, including hits for various repositories like 'jammy', 'stable', and 'jammy-security'.

```
dell@mahesh:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-08-10 15:37:29 IST; 2min 9s ago
   TriggeredBy: ● docker.socket
   Main PID: 17352 (dockerd)
     Tasks: 10
    Memory: 20.8M
      CPU: 865ms
     CGroup: /system.slice/docker.service
             └─17352 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Aug 10 15:37:28 mahesh dockerd[17352]: time="2025-08-10T15:37:28.911725728+05:30" level=info msg="detected 127.0.0.53 nam>
Aug 10 15:37:28 mahesh dockerd[17352]: time="2025-08-10T15:37:28.939109256+05:30" level=info msg="Creating a containerd c>
Aug 10 15:37:29 mahesh dockerd[17352]: time="2025-08-10T15:37:29.019509528+05:30" level=info msg="Loading containers: sta>
Aug 10 15:37:29 mahesh dockerd[17352]: time="2025-08-10T15:37:29.642660127+05:30" level=info msg="Loading containers: don>
Aug 10 15:37:29 mahesh dockerd[17352]: time="2025-08-10T15:37:29.714997397+05:30" level=info msg="Docker daemon" commit=b>
Aug 10 15:37:29 mahesh dockerd[17352]: time="2025-08-10T15:37:29.716456868+05:30" level=info msg="Initializing buildkit"
Aug 10 15:37:29 mahesh dockerd[17352]: time="2025-08-10T15:37:29.819851599+05:30" level=info msg="Completed buildkit init>
Aug 10 15:37:29 mahesh dockerd[17352]: time="2025-08-10T15:37:29.838456982+05:30" level=info msg="Daemon has completed in>
Aug 10 15:37:29 mahesh dockerd[17352]: time="2025-08-10T15:37:29.838570611+05:30" level=info msg="API listen on /run/dock>
Aug 10 15:37:29 mahesh systemd[1]: Started Docker Application Container Engine.
dell@mahesh:~$
```

```
dell@mahesh:~$ docker --version
Docker version 28.3.3, build 980b856
dell@mahesh:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
dell@mahesh:~$
```

```
dell@mahesh:~$ sudo usermod -aG docker jenkins
dell@mahesh:~$
```

#Trivy

The screenshot shows a browser window with three tabs: "Debian | Docker Docs", "Dashboard - Jenkins", and "Installation - Trivy". The "Installation - Trivy" tab is active, displaying the Trivy documentation. The left sidebar has a "Trivy" heading and a navigation menu with links like Overview, Installation, Quick Start, Scanning, Modes, Examples, Integrations, Private Docker Registries, Vulnerability Detection, Usage, Plugins, Air-Gapped Environment, Comparison with Other Scanners, Further Reading, FAQ, Maintainer, and Credits. The main content area is divided into two sections: "Debian/Ubuntu" and "Arch Linux". The "Debian/Ubuntu" section contains two code snippets for installing Trivy via apt. The first snippet uses curl to download the public key and add it to apt-key, then adds the repository to /etc/apt/sources.list.d/trivy.list and installs Trivy. The second snippet uses wget to download the deb package from GitHub and installs it using dpkg. The "Arch Linux" section states that trivy-bin can be installed from the Arch User Repository and provides a terminal screenshot showing the successful execution of the command "sudo pacman -Sv trivy-bin".

The screenshot shows a terminal window with a black background and white text. The prompt is "dell@mahesh:~". The user runs several commands to check the versions of various tools:

```
dell@mahesh:~$ jenkins --version
2.516.1
dell@mahesh:~$ java --version
openjdk 21.0.8 2025-07-15
OpenJDK Runtime Environment (build 21.0.8+9-Ubuntu-0ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu122.04.1, mixed mode, sharing)
dell@mahesh:~$ docker --version
Docker version 28.3.3, build 980b856
dell@mahesh:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
dell@mahesh:~$ trivy --version
Version: 0.65.0
dell@mahesh:~$
```

#dockerhub

The screenshot shows the Docker Hub interface for the user 'mahi320'. The left sidebar includes options like 'Repositories', 'Collaborations', 'Settings', 'Default privacy', 'Notifications', 'Billing', 'Usage', 'Pulls', and 'Storage'. The main area displays a table of repositories:

Name	Last Pushed	Contains	Visibility	Scout
mahi320/zomato	about 5 hours ago	IMAGE	Public	Inactive
mahi320/starbucks	19 days ago	IMAGE	Public	Inactive
mahi320/netflix	about 1 month ago	IMAGE	Public	Inactive
mahi320/mahesh-banking-app-cicd	over 1 year ago	IMAGE	Public	Inactive

This screenshot shows the same Docker Hub interface for user 'mahi320'. The repository 'mahi320/zomato' is now in the process of being deleted, indicated by a red 'DELETING' status next to its name. The right sidebar shows the user profile 'mahi320' with options for 'What's new', 'My profile', 'Account settings', and a 'Sign out' button.

#docker scout

The screenshot shows the Docker Scout documentation page. The URL is docs.docker.com/scout/. The page title is "Docker Scout". The left sidebar has a "Docker Scout" section under "PRODUCTS" with sub-links for Install, Quickstart, Explore, How-tos, Deep dive, Policy Evaluation, and Integrations. The main content area describes Docker Scout as a solution for proactively enhancing software supply chain security by analyzing images to compile an inventory of components (SBOM) against a continuously updated vulnerability database.

The screenshot shows the "Install Docker Scout" documentation page. The URL is docs.docker.com/scout/install/. The page title is "Install Docker Scout". The left sidebar has an "Install" link under the "Docker Scout" section. The main content area explains that the Docker Scout CLI plugin is pre-installed with Docker Desktop. It also provides instructions for manual installation if Docker Engine is used without Docker Desktop, including a command-line script:

```
$ curl -fsSL https://raw.githubusercontent.com/docker/scout-cli/main/  
$ sh install-scout.sh
```

The screenshot shows a terminal window with the following session:

```
dell@mahesh:~$ docker login  
Authenticating with existing credentials... [Username: mahi320]  
Info → To login with a different account, run 'docker logout' followed by 'docker login'  
Login Succeeded  
dell@mahesh:~$
```

```
dell@mahesh:~$ docker scout version
dell@mahesh:~$ docker scout version
version: v1.18.2 (go1.24.3 - linux/amd64)
git commit: fe75cbf34a86031408d13c036ca79a46db0769aa
dell@mahesh:~$
```

#sonarqube install

The screenshot shows the Docker Hub search interface with the query 'sonarqube' entered. The results page displays eight search results, each showing a card with the image name, publisher, description, and metrics (Pulls, Stars, Last Updated). The results are filtered by 'Images' under 'Products' and 'Docker Official Image' under 'Trusted content'.

Image	Publisher	Description	Pulls	Stars	Last Updated
sonarqube	Docker Official Images	Official images for SonarQube, code analysis tool for code quality and security	1B+	2515	4 days
mcp/sonarqube	mcp	Interact with SonarQube Cloud, Server and Community build over the web API. Analyze...	3.4K	16	3 days
bitnami/sonarqube	bitnami VMware	Bitnami container image for SonarQube	1M+	34	3 days
bitnamicharts/sonarqube	bitnami VMware	Bitnami Helm chart for SonarQube(TM)	1M+	0	2 days
owasp/sonarqube	OWASP	This project aims to enable more security functionalities to SonarQube and use it as an...	100K+	33	over 6 years
igohr/sonarqube	igohr	Concourse CI Resource for getting results from Sonarqube.	10M+	2	10 months
siamaksade/sonarqube	siamaksade	SonarQube image for OpenShift	1M+	0	about 6 years
openshiftdemos/sonarqube	openshiftdemos	An OpenShift-focused build of SonarQube	1M+	5	over 7 years

sonarqube - Official Image

hub.docker.com/_/sonarqube

Overview Tags

Quick reference

- Maintained by: SonarSource
- Where to get help: the SonarSource Community forum, the Docker Community Forums, the Docker Community Slack, or Stack Overflow

Supported tags and respective Dockerfile links

- 2025.4.2-developer, 2025.4-developer, developer
- 2025.4.2-enterprise, 2025.4-enterprise, enterprise
- 2025.4.2-datacenter-app, 2025.4-datacenter-app, datacenter-app
- 2025.4.2-datacenter-search, 2025.4-datacenter-search, datacenter-search
- 2025.1.3-developer, 2025.1-developer, 2025-lta-developer
- 2025.1.3-enterprise, 2025.1-enterprise, 2025-lta-enterprise
- 2025.1.3-datacenter-app, 2025.1-datacenter-app, 2025-lta-datacenter-app
- 2025.1.3-datacenter-search, 2025.1-datacenter-search, 2025-lta-datacenter-search
- 25.8.0.112029-community, community, latest
- 9.9.8-community, 9.9-community, 9-community, lts, lts-community

Tag summary

Recent tags: latest

Content type: Image

Digest: sha256:7106d7732...

```
$ > Docker pull command
$ docker pull sonarqube
```

Run In Docker Desktop

Requires Docker Desktop 4.37.1 or later.

This weeks pulls

Pulls: 1,798,044
Last week

```
dell@mahesh:~$ docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
32f112e3802c: Pull complete
6c884a837f25: Pull complete
acb031cbe5aa: Pull complete
93a246f2d3c0: Pull complete
67feb6c558c2: Pull complete
6d1b567fd265: Pull complete
8586f24f5df4: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:7106d77329a6fdac1a0daa8fc797da4f790f88f7cb796cc6b09375e7c889203b
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest
dell@mahesh:~$
```

```
dell@mahesh:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
sonarqube        latest   39223a90c240  3 days ago   1.22GB
dell@mahesh:~$ docker run -d --name sonarqube -p 9000:9000 sonarqube:latest
a60b83770e8e3e789ce24b31f51aad9478ea25e49c7d4ee422370214683ad842
dell@mahesh:~$ docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS
NAMES
a60b83770e8e    sonarqube:latest   "/opt/sonarqube/dock..."   2 seconds ago   Up 2 seconds   0.0.0.0:9000->9000/tcp, [::]:9000
sonarqube
dell@mahesh:~$
```

Log in - SonarQube

localhost:9000/sessions/new?return_to=%2F

Sonar

Log in to SonarQube

Login *

Password *

Go back Log In

SonarQube™ technology is powered by [SonarSource SA](#)

SonarQube Server Terms & Conditions [Community](#) [Documentation](#) [Plugins](#)

Update password - SonarQube

localhost:9000/account/reset_password

Update your password

⚠ This account should not use the default password.

Enter a new password

Old Password *

.....

Password *

Confirm Password *

Update

How do you want to create your project? [Learn more](#)

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](#)

SonarQube™ technology is powered by [SonarSource SA](#). Community Build - v25.8.0.112029 - MQR MODE. [LGPL v3](#) [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

#jenkins plugins setup

How do you want to create your project? [Sign in - Jenkins](#)

localhost:8080/login?from=%2F



Sign in to Jenkins

Username

Password

Keep me signed in

[Sign in](#)

How do you want to create your Jenkins instance? x Manage Jenkins - Jenkins x +

localhost:8080/manage/ Jenkins / Manage Jenkins

Manage Jenkins

Search settings

System Configuration

- System** Configure global settings and paths.
- Tools** Configure tools, their locations and automatic installers.
- Plugins** Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Nodes** Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

Configure the look and feel of Jenkins.

Security

- Security** Secure Jenkins; define who is allowed to access/use the system.
- Credentials** Configure credentials.
- Credential Providers** Configure the credential providers and types.
- Users** Create/delete/modify users that can log in to this Jenkins.

Status Information

- System Information** Displays various environmental information to assist trouble-shooting.
- System Log** System log captures output from java.util.logging output related to Jenkins.
- Load Statistics** Check your resource utilization and see if you need more computers for your builds.
- About Jenkins** See the version and license information.

localhost:8080/manage/pluginManager

How do you want to create your Jenkins instance? x Available plugins - Plugin Manager x +

localhost:8080/manage/pluginManager/available Jenkins / Manage Jenkins / Plugins

Plugins

Search available plugins

Install Name Released Health

Install	Name	Released	Health
<input type="checkbox"/>	Pipeline Graph Analysis 241.vc3d48fb_b_2582	2 mo 10 days ago	100
<input type="checkbox"/>	PAM Authentication 1.12	5 mo 8 days ago	97
<input type="checkbox"/>	JavaMail API 1.6.2-11	5 mo 18 days ago	100
<input type="checkbox"/>	Command Agent Launcher 123.v37cfcd92ef67	4 mo 1 day ago	100
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d	6 mo 20 days ago	100
<input type="checkbox"/>	SSH server 3.372.v5d04a_e92d8cf	1 mo 9 days ago	100
<input type="checkbox"/>	Pipeline: REST API 2.38	3 mo 12 days ago	99

How do you want to create this page? Available plugins - Plugin Manager Jenkins Manage Jenkins Plugins

localhost:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Search: sonar

Install	Name	Released	Health
<input checked="" type="checkbox"/>	Pipeline Graph Analysis 241.vc3d48fb_b_2582 Library plugins (for use by other plugins)	2 mo 10 days ago	100
<input checked="" type="checkbox"/>	Pipeline: REST API 2.38 User Interface	3 mo 12 days ago	99
<input checked="" type="checkbox"/>	Pipeline: Stage View 2.38 User Interface	3 mo 12 days ago	99
<input checked="" type="checkbox"/>	Common API for Blue Ocean 1.27.21 External Site/Tool Integrations User Interface	1 mo 24 days ago	99
<input checked="" type="checkbox"/>	Eclipse Temurin installer 146.v1898676a_f04e Provides an installer for the JDK tool that downloads the Eclipse Temurin™ build based upon OpenJDK from the Adoptium Working Group .	1 mo 6 days ago	100
<input checked="" type="checkbox"/>	SonarQube Scanner 2.18 External Site/Tool Integrations Build Reports	6 mo 13 days ago	88
<input checked="" type="checkbox"/>	Sonar Quality Gates 352.vdcdb_d7994fb_6 Library plugins (for use by other plugins) analysis Other Post-Build Actions	5 mo 18 days ago	100

How do you want to create this page? Available plugins - Plugin Manager Jenkins Manage Jenkins Plugins

localhost:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Search: docker

Install	Name	Released	Health
<input checked="" type="checkbox"/>	Docker 1274.vc0203fdf2e74 Cloud Providers Cluster Management docker	5 mo 5 days ago	100
<input checked="" type="checkbox"/>	Docker Commons 457.v0f62a_94f11a_3 Library plugins (for use by other plugins) docker	2 mo 11 days ago	100
<input checked="" type="checkbox"/>	Docker Pipeline 621.va_73f881d9232 pipeline DevOps Deployment docker	2 mo 11 days ago	97
<input checked="" type="checkbox"/>	Docker API 3.5.3-122.v156e51f30c0a_ Library plugins (for use by other plugins) docker	7 days 19 hr ago	100
<input checked="" type="checkbox"/>	docker-build-step 2.12 Build Tools docker	1 yr 2 mo ago	64
<input type="checkbox"/>	Amazon ECR 1.155.v3d884c1b_dee1 aws		

How do you want to create your Jenkins instance? x Available plugins - Plugin Manager x +

localhost:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Search: oswap

CSRF vulnerability and missing permission check

Email Extension Template 233.v1eb_88fc160b_5
Build Notifiers emailxlet

This plugin allows administrators to create global templates for the Extended Email Publisher.

OWASP Dependency-Check 5.6.1
Security DevOps Build Tools Build Reports

This plug-in can independently execute a **Dependency-Check** analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.

OWASP Dependency-Track 6.0.1
pipeline Security Build Tools Build Reports

This plug-in publishes Software Bill-of-Materials (SBOM) to Dependency-Track for continuous analysis. **Dependency-Track** is an intelligent Software Supply Chain Component Analysis platform that allows organizations to identify and reduce risk from the use of third-party and open source components.

Official OWASP ZAP 1.1.0
The Official OWASP ZAP Jenkins Plugin extends the functionality of the ZAP security tool into a CI Environment.

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

Credentials stored in plain text

OWASP ZAP 1.0.7
Other Post-Build Actions

1 yr 2 mo ago

3 mo 13 days ago

3 mo 21 days ago

4 mo 19 days ago

8 yr 1 mo ago

10 yr ago

Install

How do you want to create your Jenkins instance? x Download progress - Plugin Manager x +

localhost:8080/manage/pluginManager/updates/

Jenkins / Manage Jenkins / Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

commons-lang3 v3.x Jenkins API Success

Ionicons API Success

Folders Success

OWASP Markup Formatter Success

ASM API Success

JSON Path API Success

Structs Success

Pipeline: Step API Success

commons-text 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

Timestamper Success

Caffeine API Success

Srini Security Success

How do you want to create your Jenkins instance? x Download progress - Plugins x +

localhost:8080/manage/pluginManager/updates/

Jenkins / Manage Jenkins / Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Plugin	Status
Pipeline Graph Analysis	Success
Pipeline: REST API	Success
Pipeline: Stage View	Success
Common API for Blue Ocean	Success
Eclipse Temurin installer	Pending
SonarQube Scanner	Pending
Apache HttpComponents Client 5.x API	Pending
Sonar Quality Gates	Pending
Cloud Statistics	Pending
Authentication Tokens API	Pending
Docker Commons	Pending
Commons Compress API	Pending
Docker API	Pending
Docker	Pending
Docker Commons	Pending
Docker Pipeline	Pending
Docker API	Pending
Dev Tools Symbols API	Pending
Javadoc	Pending
JSch dependency	Pending
Maven Integration	Pending
docker-build-step	Pending
Email Extension Template	Pending
OWASP Dependency-Check	Pending
Loading plugin extensions	Pending

→ Go back to the top page

How do you want to create your Jenkins instance? x Download progress - Plugins x +

localhost:8080/manage/pluginManager/updates/

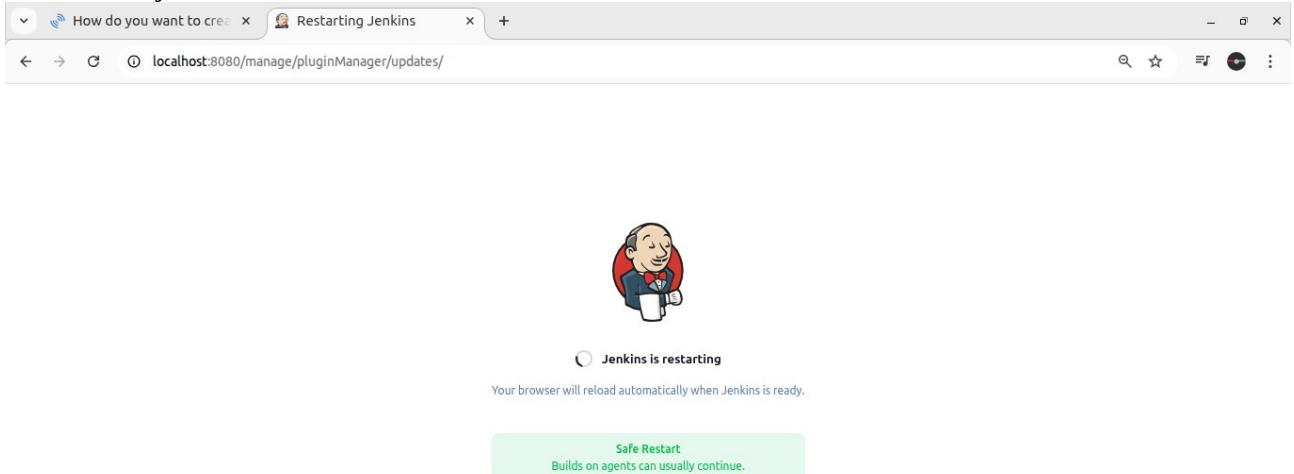
Jenkins / Manage Jenkins / Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Plugin	Status
Pipeline Graph Analysis	Success
Pipeline: REST API	Success
Pipeline: Stage View	Success
Common API for Blue Ocean	Success
Eclipse Temurin installer	Success
SonarQube Scanner	Success
Apache HttpComponents Client 5.x API	Success
Sonar Quality Gates	Success
Cloud Statistics	Success
Authentication Tokens API	Success
Docker Commons	Success
Commons Compress API	Success
Docker API	Success
Docker	Success
Docker Commons	Success
Docker Pipeline	Success
Docker API	Success
Dev Tools Symbols API	Success
Javadoc	Success
JSch dependency	Success
Maven Integration	Installing
docker-build-step	Pending
Email Extension Template	Pending
OWASP Dependency-Check	Pending
Loading plugin extensions	Pending

#restart the jenkins



#jenkins tools setup

A screenshot of the Jenkins "Tools" configuration page. The URL is "localhost:8080/manage/configureTools/". The page has a header with "Jenkins / Manage Jenkins / Tools". It includes sections for "Maven Configuration", "Default settings provider" (with a dropdown menu set to "Use default maven settings"), "Default global settings provider" (with a dropdown menu set to "Use default maven global settings"), "JDK installations" (with an "Add JDK" button), and "Git installations" (with a "Git" entry in a list and a "Name" field). At the bottom are "Save" and "Apply" buttons.

Users - Administration x Tools - Jenkins x localhost:8080/manage/configureTools/

Jenkins / Manage Jenkins / Tools

JDK installations

Add JDK

JDK

Name: jdk21

Install automatically ?

Install from adoptium.net ?

Version: jdk-17.0.8.1+1

Add Installer ▾

Add JDK

Save **Apply**

This screenshot shows the 'Tools' configuration page in Jenkins. Under the 'JDK installations' section, a new entry for 'jdk21' is being added. The 'Name' field contains 'jdk21'. The 'Install automatically' checkbox is checked. Under 'Install from adoptium.net', the 'Version' dropdown is set to 'jdk-17.0.8.1+1'. There is also an 'Add Installer' button. At the bottom, there are 'Save' and 'Apply' buttons.

Users - Administration x Tools - Jenkins x localhost:8080/manage/configureTools/

Jenkins / Manage Jenkins / Tools

Git installations

Add Git

Git

Name: Default

Path to Git executable: git

Install automatically ?

Add Git ▾

Gradle installations

Add Gradle

Save **Apply**

This screenshot shows the 'Tools' configuration page in Jenkins. Under the 'Git installations' section, a new entry for 'Default' is being added. The 'Name' field contains 'Default'. The 'Path to Git executable' field contains 'git'. There is an 'Install automatically' checkbox which is unchecked. There is also an 'Add Git' button. At the bottom, there are 'Save' and 'Apply' buttons.

SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name
sonar-scanner

Install automatically ?

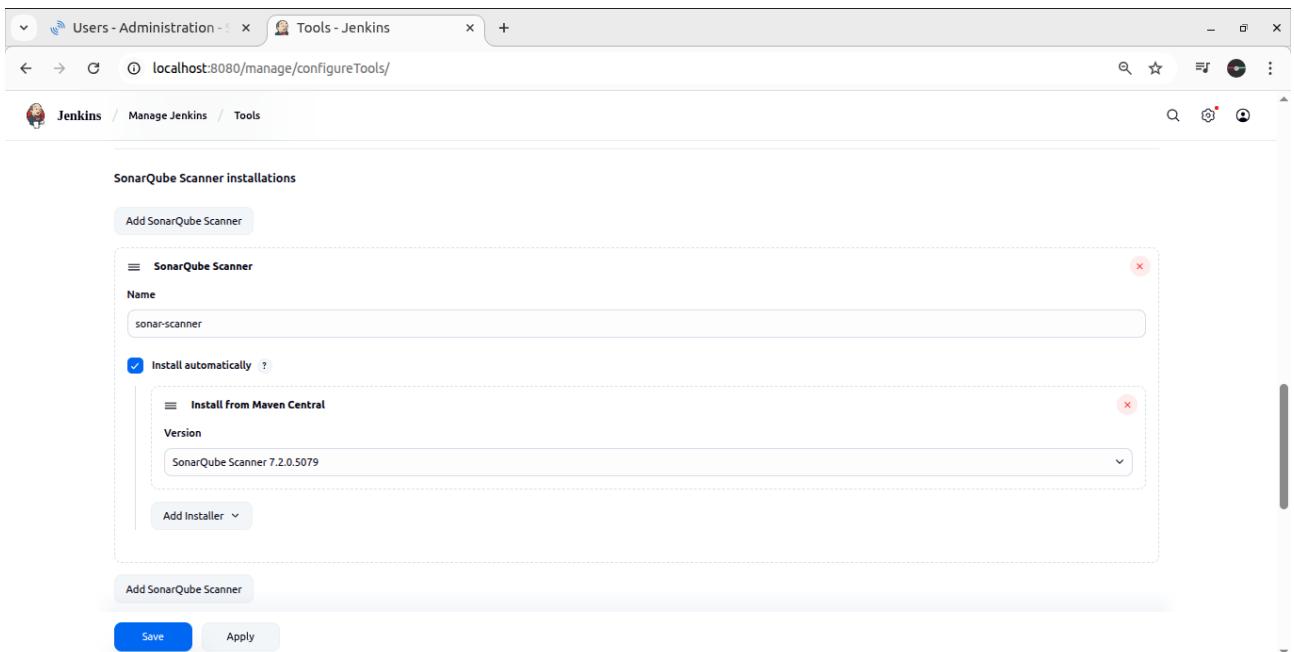
Install from Maven Central

Version
SonarQube Scanner 7.2.0.5079

Add Installer ▾

Add SonarQube Scanner

Save **Apply**



Add NodeJS

NodeJS

Name
node24

Install automatically ?

Install from nodejs.org

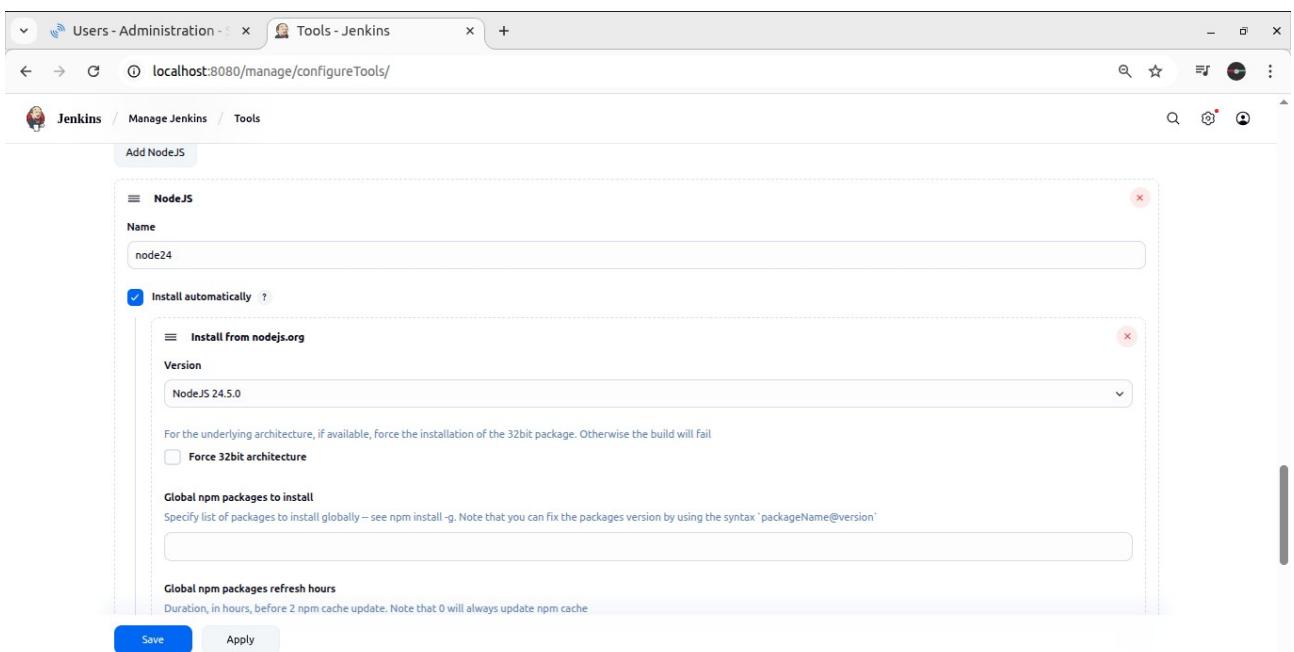
Version
NodeJS 24.5.0

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail
 Force 32bit architecture

Global npm packages to install
Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'
[empty input field]

Global npm packages refresh hours
Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache
[empty input field]

Save **Apply**



Users - Administration x Tools - Jenkins x localhost:8080/manage/configureTools/

Jenkins / Manage Jenkins / Tools

Dependency-Check installations

Add Dependency-Check

Dependency-Check

Name: DP-Check

Install automatically:

Add Installer

Add Dependency-Check

Docker installations

Add Docker

Save Apply

This screenshot shows the Jenkins 'Tools' configuration page. Under the 'Dependency-Check installations' section, a new tool named 'DP-Check' is being added. The 'Install automatically' checkbox is checked. The 'Docker installations' section is also visible below.

Users - Administration x Tools - Jenkins x localhost:8080/manage/configureTools/

Jenkins / Manage Jenkins / Tools

Docker installations

Add Docker

Docker

Name: docker

Install automatically:

Download from docker.com

Docker version: latest

Add Installer

Add Docker

Save Apply

This screenshot shows the Jenkins 'Tools' configuration page. Under the 'Docker installations' section, a new tool named 'docker' is being added. It is set to download from docker.com and use the latest version. The 'Add Docker' button is visible above the configuration fields.

The screenshot shows the Jenkins 'Tools' configuration page at localhost:8080/manage/configureTools/. The 'Docker installations' section is displayed, featuring a 'Docker' configuration card. The card has a 'Name' field set to 'docker', an 'Install automatically' checkbox checked, and a 'Download from docker.com' section with a 'Docker version' dropdown set to 'latest'. A 'Save' and 'Apply' button are located at the bottom.

```
#credentials setup  
sonarqube key, docker credentials, gmail key
```

The screenshot shows the Jenkins 'Manage Jenkins' dashboard at localhost:8080/manage/. The 'Security' section is active, showing links for 'Security', 'Users', and 'Credential Providers'. Other sections like 'Status Information' (System Information, System Log, Load Statistics) and 'Troubleshooting' (Manage Old Data) are also visible.

The screenshot shows the Jenkins 'Credentials' management page. At the top, there are tabs for 'T' (Text), 'P' (Password), 'Store' (selected), and 'Domain'. Below the tabs, a table header includes columns for 'Domain', 'ID', and 'Name'. A section titled 'Stores scoped to Jenkins' lists a single store named 'System' under the 'Domains' column, with '(global)' selected. At the bottom left, there are icons for 'Icon:', 'S' (Secret text), 'M' (Masked password), and 'L' (List). The URL in the address bar is `localhost:8080/manage/credentials/`. The bottom right corner shows 'REST API' and 'Jenkins 2.516.1'.

The screenshot shows the 'New credentials' creation form. The title bar indicates the URL is `localhost:8080/manage/credentials/store/system/domain/_/newCredentials`. The form fields include:

- Kind:** Secret text (dropdown menu)
- Scope:** Global (Jenkins, nodes, items, all child items, etc) (dropdown menu)
- Secret:** (text input field)
- ID:** (text input field)
- Description:** (text input field)

A blue 'Create' button is at the bottom. The bottom right corner shows 'REST API' and 'Jenkins 2.516.1'.

SonarQube™ technology is powered by SonarSource SA

Community Build v25.8.0.112029 - MQR MODE

LGPL v3 Community Documentation Plugins Web API

The screenshot shows the 'Users' section of the SonarQube administration interface. A single user, 'Administrator' (admin), is listed. The table includes columns for Name, SCM Accounts, Last connection, Last SonarQube for IDE connection ?, Groups, Tokens, and Actions. The 'Groups' column shows 2 groups assigned, and the 'Tokens' column shows 0 tokens issued. A 'Create User' button is located in the top right corner.

SonarQube™ technology is powered by SonarSource SA

Community Build v25.8.0.112029 - MQR MODE

LGPL v3 Community Documentation Plugins Web API

The screenshot shows the 'Users' section with a modal dialog titled 'Tokens of Administrator'. The modal contains a 'Generate Tokens' form with fields for 'Name' (Administrator) and 'Expires in' (30 days). A success message states 'New token "token" has been created. Make sure you copy it now, you won't be able to see it again!' with a generated token value: 'squ_ae9197e129cf415f11648ddcd141642ad5f36f59'. A 'Close' button is at the bottom right of the modal.

Users - Administration × New credentials - Jenkins × +

localhost:8080/manage/credentials/store/system/domain/_newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

New credentials

Kind

Secret text

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

Secret
.....

ID ?
Sonar-token

Description ?
Sonar-token

Create

REST API Jenkins 2.516.1

This screenshot shows the 'New credentials' page for a 'Secret text' type credential. The 'Kind' field is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field contains several redacted characters. The 'ID' field is filled with 'Sonar-token', and the 'Description' field also contains 'Sonar-token'. A blue 'Create' button is located at the bottom left of the form.

Users - Administration × New credentials - Jenkins × +

localhost:8080/manage/credentials/store/system/domain/_newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

New credentials

Kind

Username with password

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

Username ?
mahi320

Treat username as secret ?

Password ?
.....

ID ?
docker

Description ?
docker

Create

This screenshot shows the 'New credentials' page for a 'Username with password' type credential. The 'Kind' field is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field is filled with 'mahi320'. There is a checkbox labeled 'Treat username as secret ?' which is unchecked. The 'Password' field contains several redacted characters. The 'ID' field is filled with 'docker', and the 'Description' field also contains 'docker'. A blue 'Create' button is located at the bottom left of the form.

Users - Administration > System » Global credentials

localhost:8080/manage/credentials/store/system/domain/_/

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
Sonar-token	Sonar-token	Secret text	Sonar-token
docker	mahi320/***** (docker)	Username with password	docker

Icon: S M L

REST API Jenkins 2.516.1

#in jenkins setup sonarqbe

Users - Administration > System - Jenkins

localhost:8080/manage/configure

Jenkins / Manage Jenkins / System

Environment variables
 Tool Locations

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
Add SonarQube

Metrics

Access keys ?
Add new access key

Pipeline Speed / Durability

Default Speed / Durability Level ?
None: use pipeline default (Maximum survivability/durability but slowest)

Save Apply

Users - Administration x System - Jenkins x

localhost:8080/manage/configure

Jenkins / Manage Jenkins / System

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

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

Server authentication token: Sonar-token

+ Add

Advanced ▾

Save Apply



Users - Administration x Dashboard - Jenkins x

localhost:9000/admin/users

⚠️ 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](#)

SonarQube community Projects Issues Rules Quality Profiles Quality Gates Administration More

Administration

Configuration Security Projects System Marketplace

General Settings

Sind administer individual users.

Create User

Encryption Webhooks

search by login Filter by All users ?

Name	SCM Accounts	Last connection	Last SonarQube for IDE connection ?	Groups	Tokens	Actions
A Administrator admin		< 1 hour ago	Never	2	1	⋮

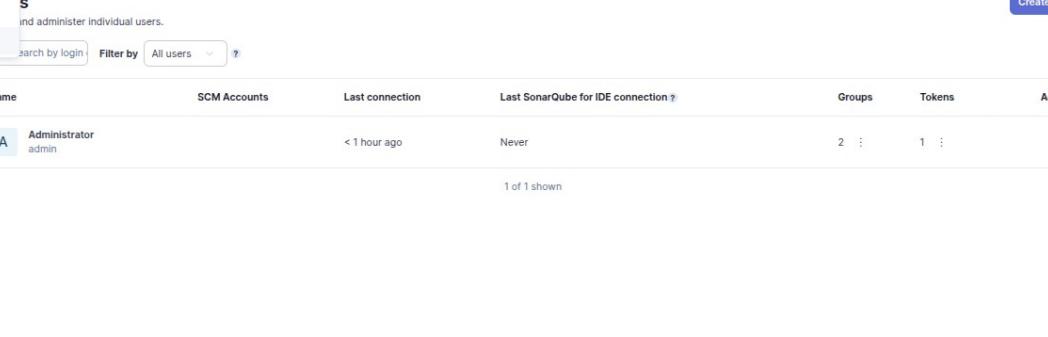
1 of 1 shown

SonarQube™ technology is powered by SonarSource SA

Community Build - v25.8.0.112029 - MQR MODE

LGPL v3 Community Documentation Plugins Web API

localhost:9000/admin/webhooks



Webhooks - Administration x Dashboard - Jenkins x | +

localhost:9000/admin/webhooks

⚠️ 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](#)

SonarQube community Projects Issues Rules Quality Profiles Quality Gates Administration More

Administration Configuration Security Projects System Marketplace

Webhooks

Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs.

No webhook defined.

Create Webhook

Name * jenkins

URL * http://localhost:8080/sonarqube-webhook

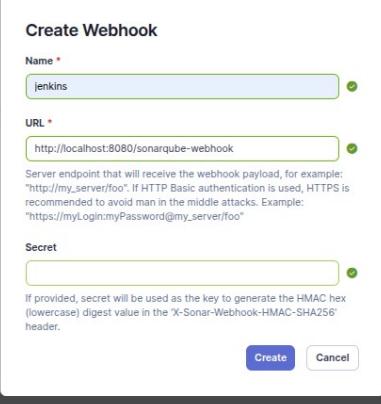
Secret

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

SonarQube™ technology is powered by [SonarSource SA](#)

LGPL v3 Community Documentation Plugins Web API



How do you want to create x New Item - Jenkins x +

localhost:9000/projects/create

⚠️ 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](#)

SonarQube community Projects Issues Rules Quality Profiles Quality Gates Administration 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 Import from Bitbucket Cloud Import from Bitbucket Server

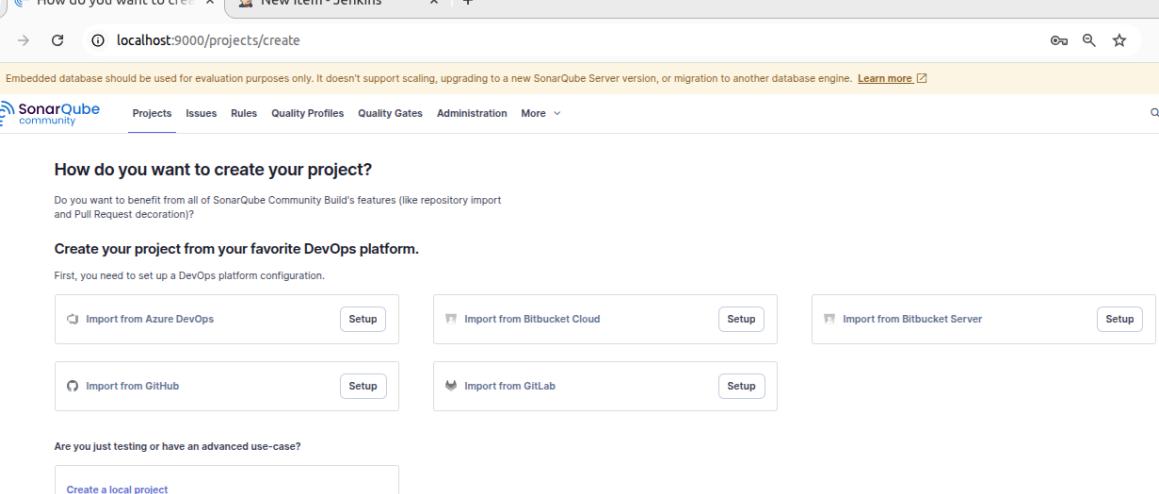
Import from GitHub Import from GitLab

Are you just testing or have an advanced use-case?

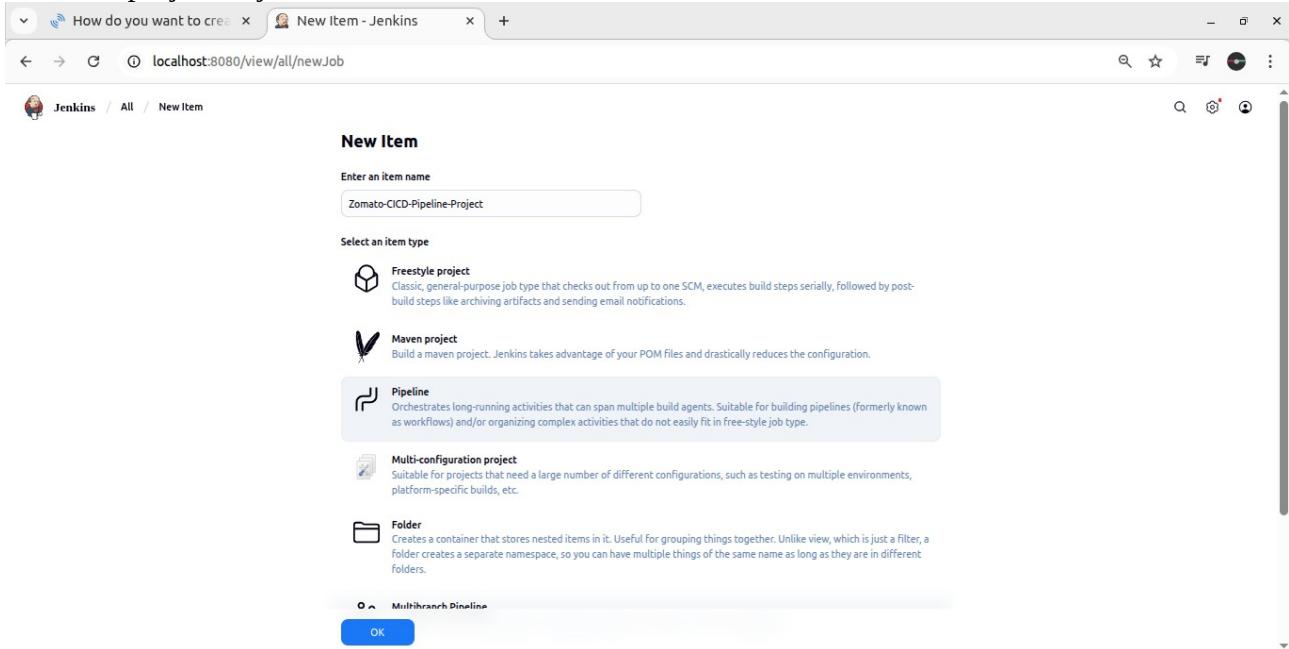
Create a local project

SonarQube™ technology is powered by [SonarSource SA](#)

LGPL v3 Community Documentation Plugins Web API



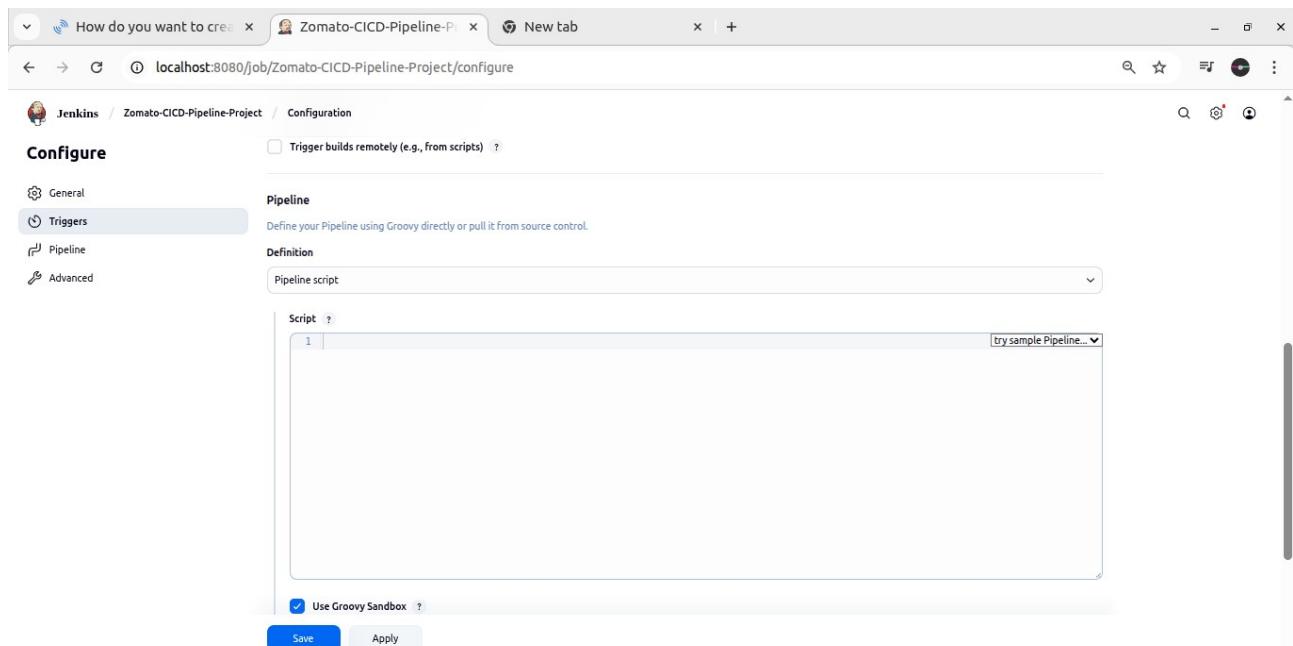
#create a project in jenkins



The screenshot shows the Jenkins 'New Item' creation dialog. At the top, there's a search bar with the placeholder 'Enter an item name' containing the text 'Zomato-CI_CD-Pipeline-Project'. Below it, a section titled 'Select an item type' lists several options:

- 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.
- Maven project**: Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- 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**: (This option is partially visible at the bottom)

At the bottom right of the dialog is a blue 'OK' button.



The screenshot shows the Jenkins Pipeline configuration dialog for the 'Zomato-CI_CD-Pipeline-Project'. The title bar indicates the URL is 'localhost:8080/job/Zomato-CI_CD-Pipeline-Project/configure'. The left sidebar has tabs for 'General', 'Triggers' (which is selected), 'Pipeline', 'Definition', and 'Advanced'. The 'Pipeline' tab is active, showing the sub-section 'Definition' with a dropdown menu set to 'Pipeline script'. A large text area labeled 'Script' contains a single line of Groovy code: '1'. To the right of the script area is a preview window showing the text 'try sample Pipeline...'. At the bottom of the dialog are two buttons: 'Save' (in blue) and 'Apply'.

Webhooks - Administration x Zomato-CI_CD-Pipeline-Project x New tab x | +

localhost:8080/job/Zomato-CI_CD-Pipeline-Project/configure

Jenkins / Zomato-CI_CD-Pipeline-Project / Configuration

Configure

Pipeline script

General Triggers Pipeline Advanced

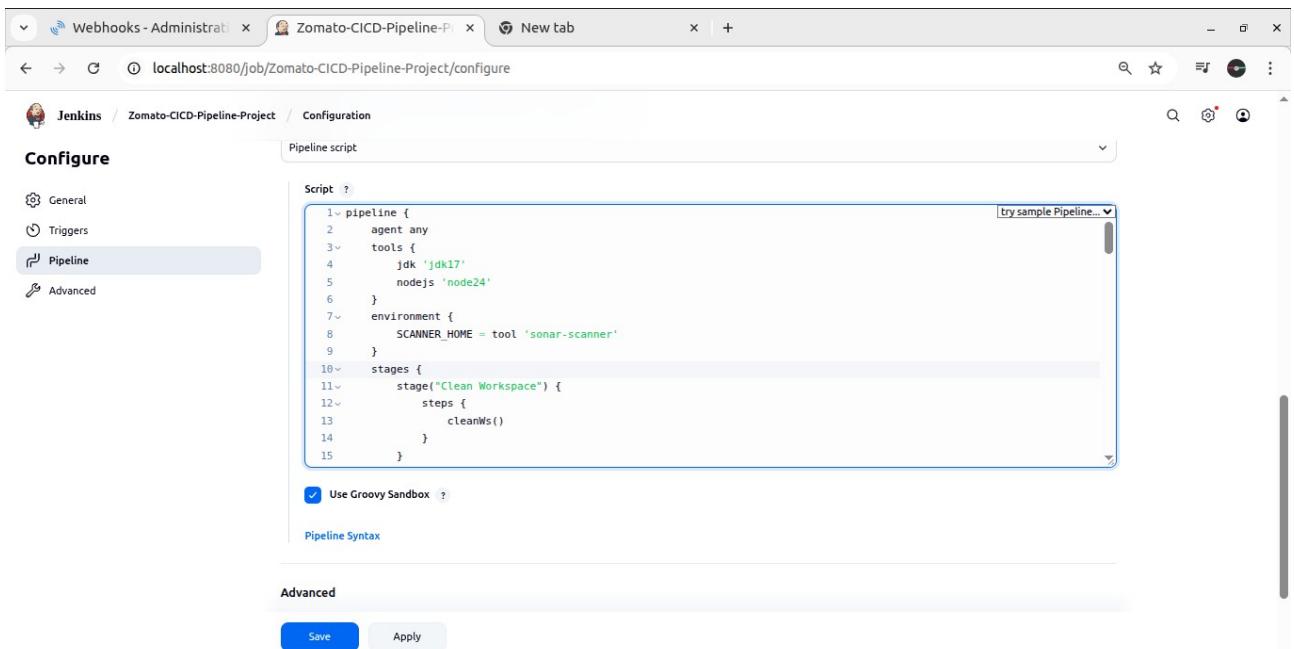
```
1 pipeline {  
2     agent any  
3     tools {  
4         jdk 'jdk17'  
5         nodejs 'node24'  
6     }  
7     environment {  
8         SCANNER_HOME = tool 'sonar-scanner'  
9     }  
10    stages {  
11        stage("Clean Workspace") {  
12            steps {  
13                cleanWs()  
14            }  
15        }  
16    }  
17}
```

Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Save Apply



Webhooks - Administration x Zomato-CI_CD-Pipeline-Project x New tab x | +

localhost:8080/job/Zomato-CI_CD-Pipeline-Project/configure

Jenkins / Zomato-CI_CD-Pipeline-Project / Configuration

Configure

General Triggers Pipeline Advanced

```
56    }  
57  
58    stage("Build Docker Image") {  
59        steps {  
60            sh 'docker build -t zomato-app .'  
61        }  
62    }  
63  
64    stage("Push to DockerHub") {  
65        steps {  
66            script {  
67                withDockerRegistry(credentialsId: 'docker') {  
68                    sh 'docker push zomato-app'  
69                }  
70            }  
71        }  
72    }
```

Use Groovy Sandbox ?

Pipeline Syntax

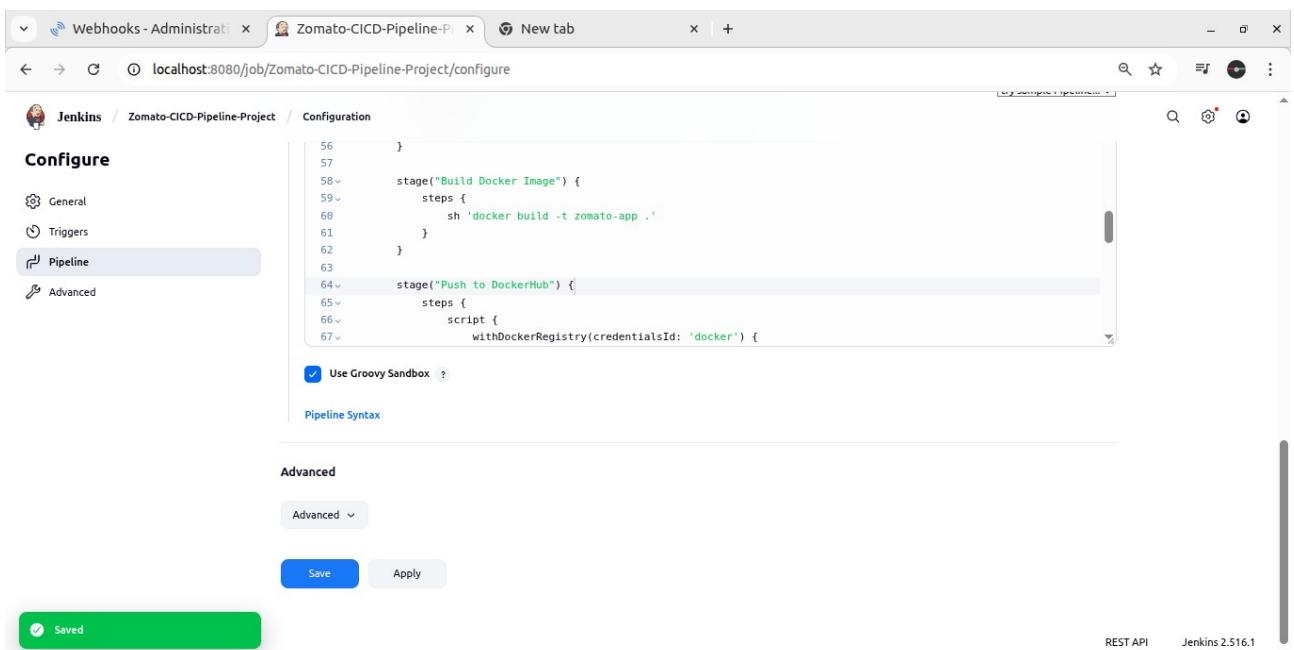
Advanced

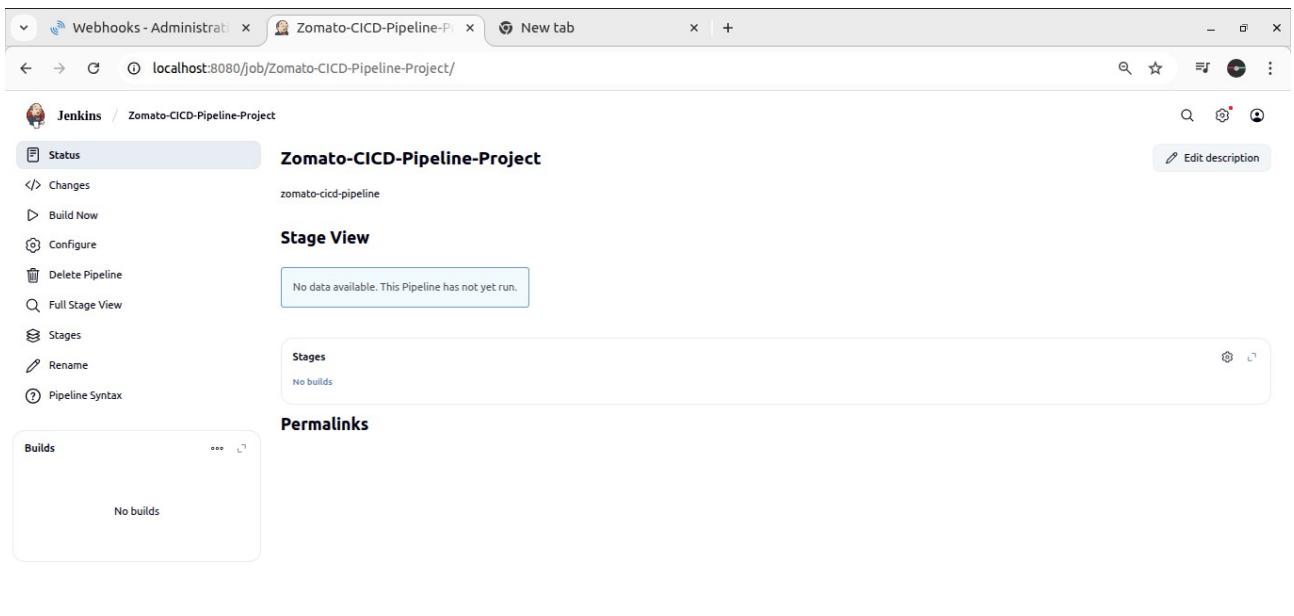
Advanced ▾

Save Apply

 Saved

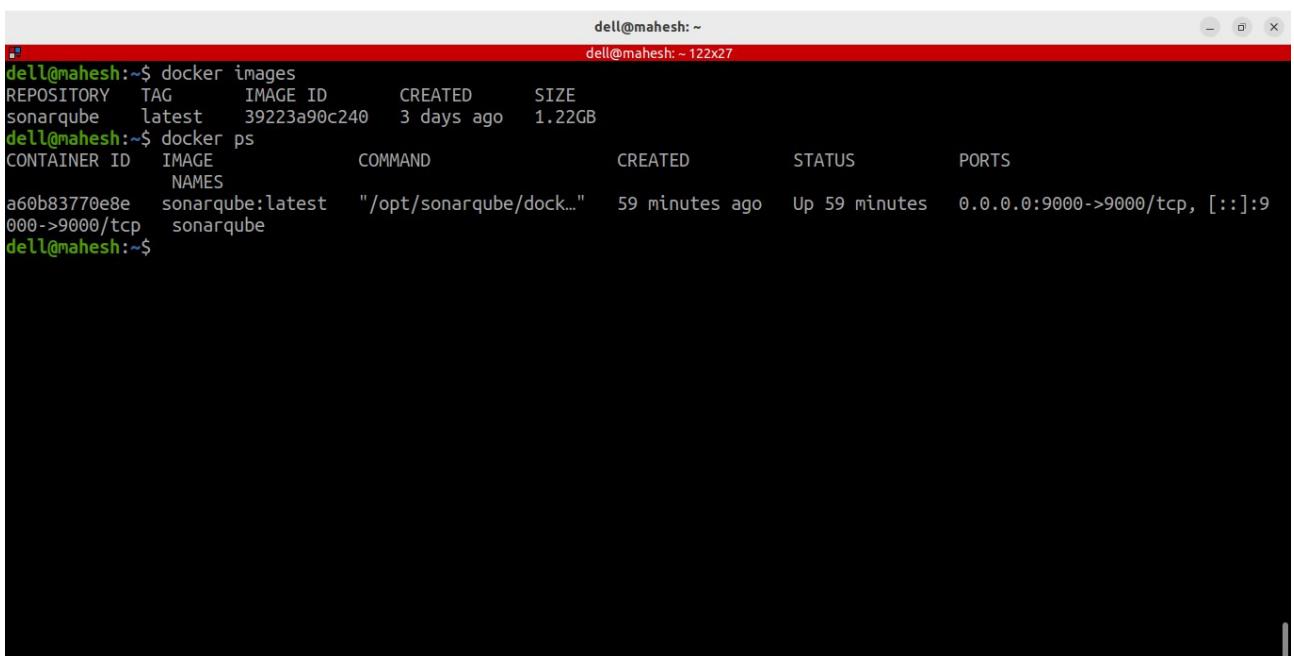
REST API Jenkins 2.516.1



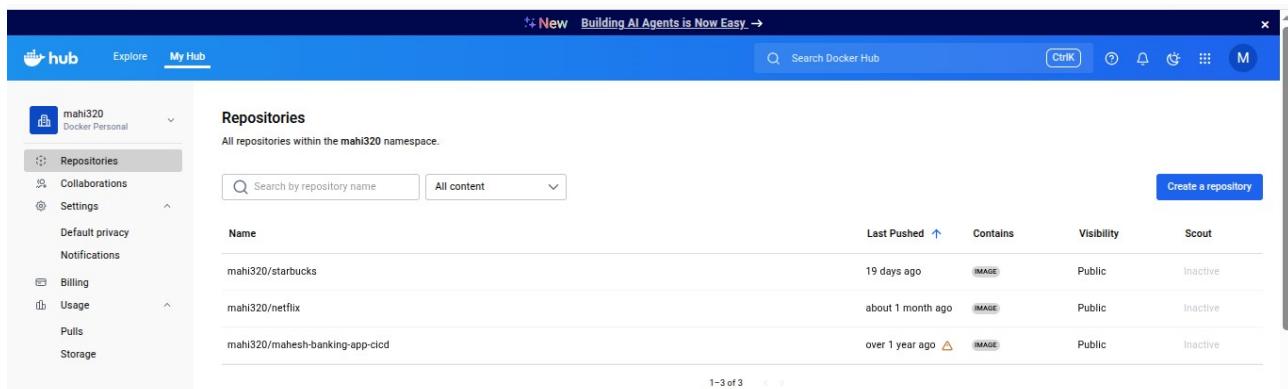


The screenshot shows the Jenkins interface for the 'Zomato-CI-CD-Pipeline-Project'. The left sidebar contains links for Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main content area is titled 'Stage View' and displays a message: 'No data available. This Pipeline has not yet run.' Below this is a 'Stages' section showing 'No builds'. A 'Permalinks' section includes a 'Builds' card with the message 'No builds'. At the bottom right, there are links for REST API and Jenkins 2.516.1.

#check images status



```
dell@mahesh:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
sonarqube latest 39223a90c240 3 days ago 1.22GB
dell@mahesh:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
a60b83770e8e sonarqube:latest "/opt/sonarqube/dock..." 59 minutes ago Up 59 minutes 0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp
sonarqube
dell@mahesh:~$
```



The screenshot shows the Docker Hub interface for the user 'mahi320'. The left sidebar includes sections for Repositories, Collaborations, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main content area is titled 'Repositories' and lists three repositories: 'mahi320/starbucks', 'mahi320/netflix', and 'mahi320/mahesh-banking-app-cicd'. Each repository entry includes the name, last pushed date, image count, visibility, and status.

Name	Last Pushed	Contains	Visibility	Scout
mahi320/starbucks	19 days ago	IMAGE	Public	Inactive
mahi320/netflix	about 1 month ago	IMAGE	Public	Inactive
mahi320/mahesh-banking-app-cicd	over 1 year ago	IMAGE	Public	Inactive

The screenshot shows the Jenkins Pipeline configuration page for the 'Zomato-CI-CD-Pipeline-Project'. The 'Pipeline' tab is selected in the left sidebar. The main area contains a Groovy script block:

```
post {  
    always {  
        echo "Pipeline completed with status: ${currentBuild.currentResult}"  
    }  
    success {  
        echo "Application deployed successfully! Access at: http://<your-server-ip>:3001"  
    }  
    unstable {  
        echo "Pipeline completed with warnings (OWASP scan)"  
    }  
    failure {  
        echo "Pipeline failed. Check logs for errors."  
    }  
}
```

A checkbox labeled 'Use Groovy Sandbox' is checked. Below the script, there's a 'Pipeline Syntax' link. At the bottom, there are 'Advanced' and 'Saved' buttons, along with 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins Pipeline status page for the 'Zomato-CI-CD-Pipeline-Project'. The 'Status' tab is selected in the left sidebar. The main area displays the pipeline stages:

- Stage View**: Shows a message: "No data available. This Pipeline has not yet run."
- Stages**: Shows a message: "No builds".

On the left sidebar, other options like 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Full Stage View', 'Stages', 'Rename', and 'Pipeline Syntax' are listed. On the right, there's an 'Add description' button. At the bottom, there's a green button labeled 'Build scheduled'.

#pipeline is scheduled

The screenshot shows the Jenkins Pipeline Project status page for 'Zomato-CICD-Pipeline-Project'. The pipeline consists of four stages: Declarative Tool Install, Clean Workspace, Git Checkout, and SonarQube Analysis. The 'Average stage times' are listed as 326ms, 870ms, 8s, and 17s respectively. A build history section shows build #1 scheduled for Aug 10 at 17:30. The status bar indicates the build has completed successfully. The Jenkins version is 2.516.1.

#Sonarqube-code-status

The screenshot shows the SonarQube dashboard for the 'zomato' project. The main summary indicates a 'Passed' status with 1.3k Lines of Code and a last analysis 1 minute ago. Key metrics shown include 0 New issues, 0 Accepted issues, 0 Duplications, and 0 Security Hotspots. The coverage is listed as 'Not computed'. The dashboard also includes tabs for Issues, Security Hotspots, Code, Measures, and Activity.

Webhooks - Administration | Zomato-CI-CD-Pipeline-Project | Projects - SonarQube Community | Docker Hub

localhost:9000/projects

⚠️ 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](#)

ⓘ The way in which security, reliability, and maintainability counts and ratings are calculated has changed. [Learn more in SonarQube documentation](#)

SonarQube community Projects Issues Rules Quality Profiles Quality Gates Administration More

My Favorites All

Filters

Quality Gate

- Passed 1
- Failed 0

Security

- A ≥ 0 info issues 0
- B ≥ 1 low issue 0
- C ≥ 1 medium issue 1
- D ≥ 1 high issue 0
- E ≥ 1 blocker issue 0

Reliability

- A ≥ 0 info issues 0

Search projects (minimum 2 characters) Perspective Overall Status Sort by Name

zomato Public Last analysis: 2 minutes ago - 1.3k Lines of Code - CSS, JavaScript, ...

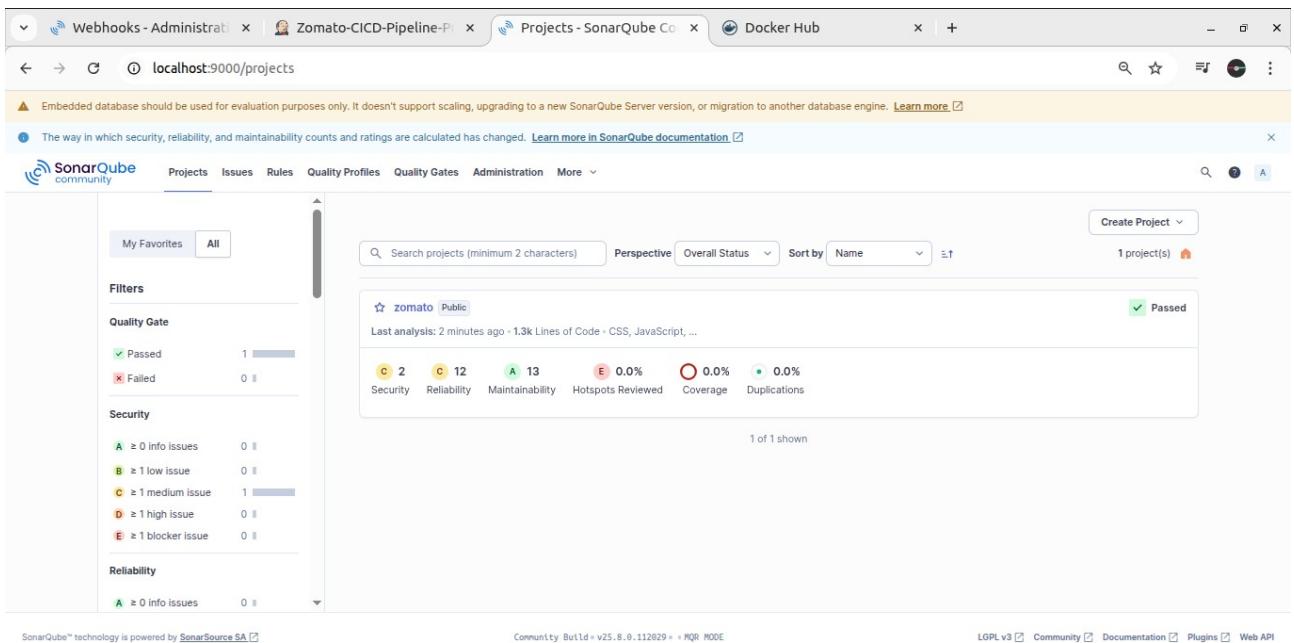
Passed

C 2 C 12 A 13 E 0.0% H 0.0% Coverage 0.0% Duplications 0.0%

1 of 1 shown

SonarQube™ technology is powered by [SonarSource SA](#) Community Build v25.8.0.112029 + MQR MODE

LGPL v3 Documentation Plugins Web API



Webhooks - Administration | Zomato-CI-CD-Pipeline-Project | zomato - Overview - SonarQube Community | Docker Hub

localhost:9000/dashboard?id=zomato

⚠️ 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](#)

SonarQube community Projects Issues Rules Quality Profiles Quality Gates Administration More

zomato Bind project / main ✓

Overview Issues Security Hotspots Code Measures Activity Project Settings Project Information

main 1.3k Lines of Code · Version not provided Set as homepage Take the Tour

Quality Gate Passed Last analysis 2 minutes ago

New Code Overall Code

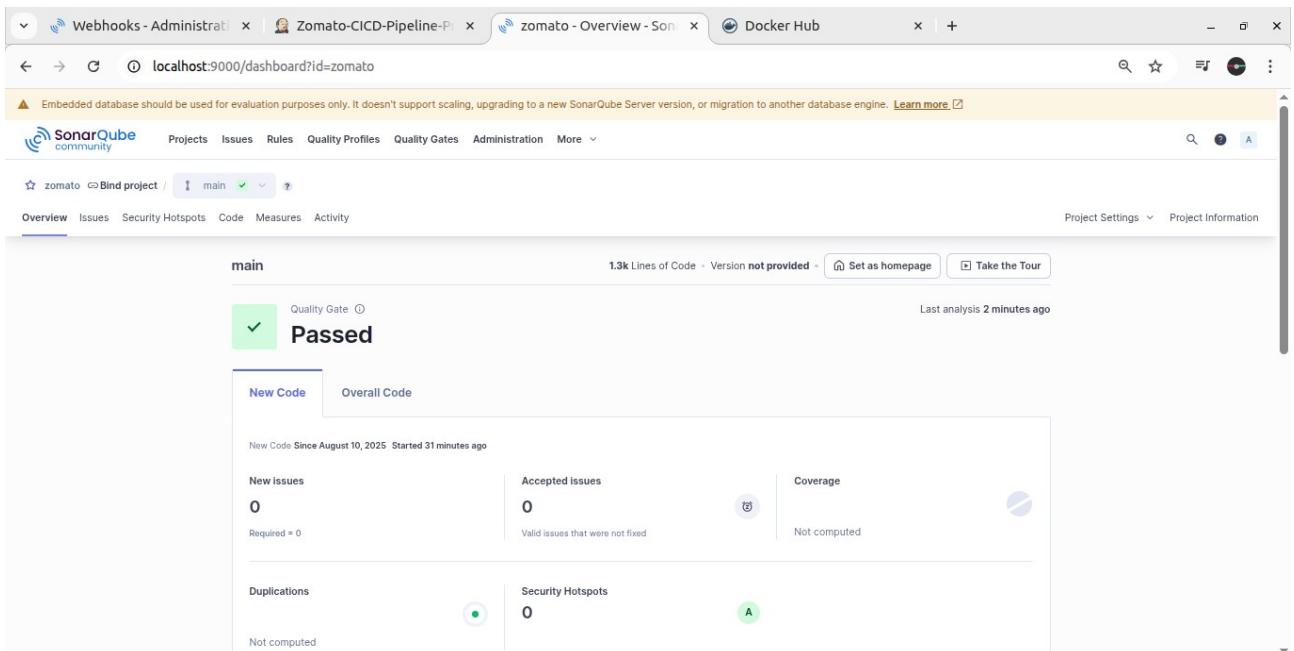
New Code Since August 10, 2025 Started 31 minutes ago

New issues	Accepted issues	Coverage
0	0	Not computed

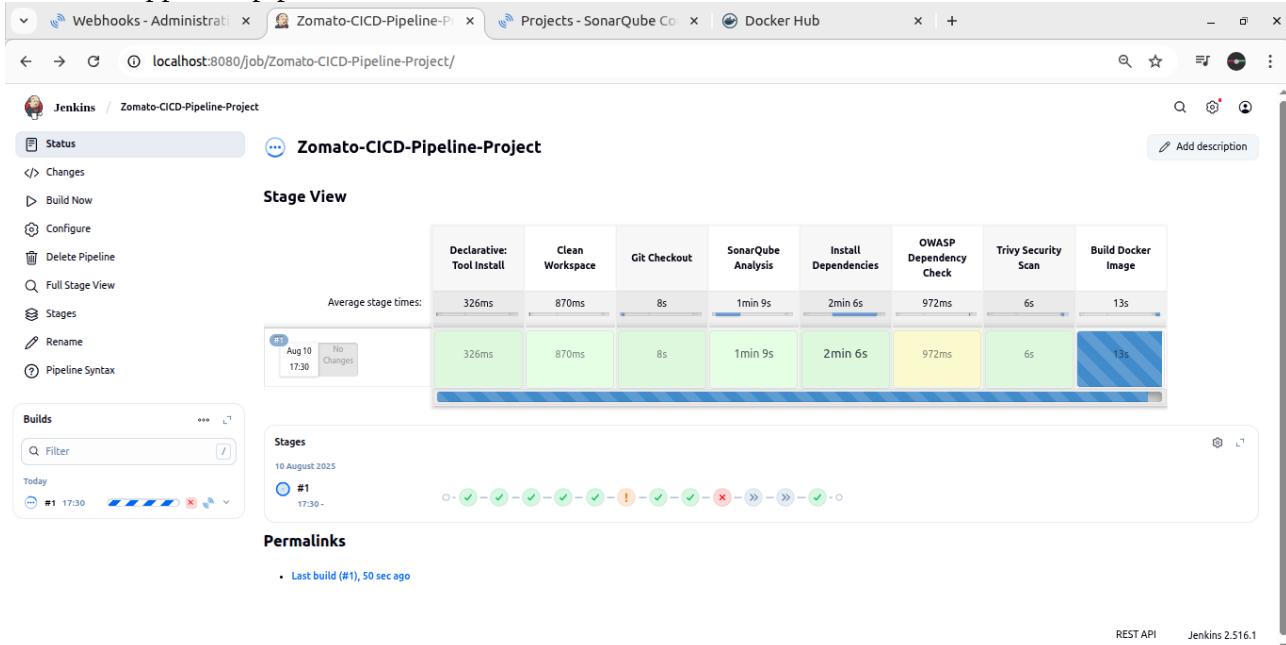
New issues Accepted issues Coverage

Duplications Security Hotspots

Not computed 0 A



#Zomato-app-cicd-pipeline



The screenshot shows the Jenkins Pipeline Project status page for 'Zomato-CI_CD-Pipeline-Project'. The pipeline consists of eight stages: Declarative: Tool Install, Clean Workspace, Git Checkout, SonarQube Analysis, Install Dependencies, OWASP Dependency Check, Trivy Security Scan, and Build Docker Image. The total average stage time is 1min 9s. The most recent build (#1) was run on Aug 10 at 17:30 and passed.

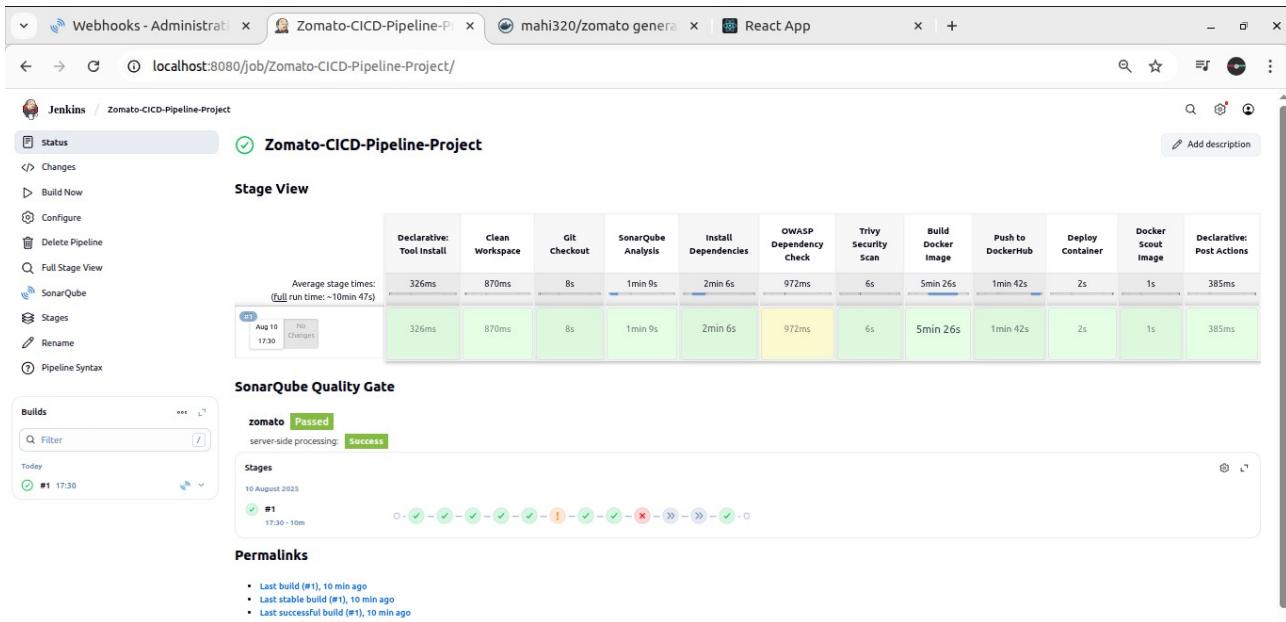
Stage	Time
Declarative: Tool Install	326ms
Clean Workspace	870ms
Git Checkout	8s
SonarQube Analysis	1min 9s
Install Dependencies	2min 6s
OWASP Dependency Check	972ms
Trivy Security Scan	6s
Build Docker Image	13s

Builds
Last build (#1), 50 sec ago

Stages
10 August 2025
#1 17:30 -

Permalinks
REST API Jenkins 2.516.1

#Pipeline has sucess



The screenshot shows the Jenkins Pipeline Project status page for 'Zomato-CI_CD-Pipeline-Project'. The pipeline consists of twelve stages: Declarative: Tool Install, Clean Workspace, Git Checkout, SonarQube Analysis, Install Dependencies, OWASP Dependency Check, Trivy Security Scan, Build Docker Image, Push to DockerHub, Deploy Container, Docker Scout Image, and Declarative: Post Actions. The total average stage time is 10min 47s. The most recent build (#1) was run on Aug 10 at 17:30 and passed. The SonarQube Quality Gate also passed.

Stage	Time
Declarative: Tool Install	326ms
Clean Workspace	870ms
Git Checkout	8s
SonarQube Analysis	1min 9s
Install Dependencies	2min 6s
OWASP Dependency Check	972ms
Trivy Security Scan	6s
Build Docker Image	5min 26s
Push to DockerHub	1min 42s
Deploy Container	2s
Docker Scout Image	1s
Declarative: Post Actions	385ms

Builds
Last build (#1), 10 min ago

Stages
10 August 2025
#1 17:30 - 10m

SonarQube Quality Gate
zomato Passed
server-side processing: Success

Permalinks

The screenshot shows the Jenkins Pipeline Overview for job #1. The pipeline graph indicates a sequence of steps: Start, Tool Install (green checkmark), Clean Workspace (green checkmark), Git Checkout (green checkmark), SonarQube Analysis (green checkmark), Install Dependencies (green checkmark), OWASP Dependency Check (orange exclamation mark), Trivy Security Scan (red X), Build Docker Image (green checkmark), Push to DockerHub (red X), Deploy Container (grey question mark), Docker Scout Image (grey question mark), Post Actions (green checkmark), and End. The 'Tool Install' step is expanded, showing its sub-steps: Use a tool from a predefined Tool installation (dk17), Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step., Use a tool from a predefined Tool installation (node24), and Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. The entire pipeline run started 12 minutes ago and took 10 minutes.

The screenshot shows the SonarQube dashboard for the 'zomato' project. A warning message at the top states: "⚠️ 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](#)". The main interface shows the 'main' module with a green 'Passed' status. Key metrics displayed include: 1.3k Lines of Code, Version not provided, Set as homepage, Take the Tour, Last analysis 12 minutes ago, New Code Since August 10, 2025, Started 41 minutes ago, New issues 0 (Required = 0), Accepted issues 0 (Valid issues that were not fixed), Coverage Not computed, Duplications Not computed, and Security Hotspots 0. Navigation tabs include Overview, Issues, Security Hotspots, Code, Measures, and Activity. Project Settings and Project Information are also visible.

The screenshot shows the Jenkins dashboard at localhost:8080. The main header includes tabs for 'Webhooks - Administra...', 'Dashboard - Jenkins', 'Projects - SonarQube', 'mahi320/zomato gener...', and 'React App'. The Jenkins dashboard has a search bar and navigation icons. On the left, there are links for 'New Item', 'Build History', and 'Build Executor Status'. A 'Build Queue' section indicates 'No builds in the queue.' Below it, the 'Build Executor Status' shows 0/2 executors available. The central part of the dashboard displays a table for the 'Zomato-CI_CD-Pipeline-Project'. The table columns are 'S' (Status), 'W' (Waiting), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The project status is green ('S') with a checkmark icon, labeled 'Zomato-CI_CD-Pipeline-Project'. Its last success was 13 min ago, and its last failure was N/A. The duration was 10 min. There is a 'More' button (three dots) next to the row. At the bottom, there are icons for 'Icon: S M L' and a 'More' link.

#check the docker images

```
dell@mahesh:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED       SIZE
mahi320/zomato  latest   55eba30ad407  4 minutes ago  807MB
zomato          latest   55eba30ad407  4 minutes ago  807MB
sonarqube       latest   39223a90c240  3 days ago    1.22GB
dell@mahesh:~$ docker ps
CONTAINER ID   IMAGE           COMMAND        CREATED       STATUS        PORTS
NAMES
fb2ab546ee9f   mahi320/zomato:latest "docker-entrypoint.s..."  About a minute ago Up  About a minute  0.0.0.0:3001->3000
0/tcp, [::]:3001->3000/tcp  zomato
a60b83770e8e   sonarqube:latest   "/opt/sonarqube/dock..."  2 hours ago     Up  2 hours      0.0.0.0:9000->9000
0/tcp, [::]:9000->9000/tcp  sonarqube
dell@mahesh:~$
```

#check in docker hub

The screenshot shows the Docker Hub interface for the user 'mahi320'. The left sidebar includes options like Repositories, Collaborations, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main area displays a list of repositories under the 'Repositories' tab. The repository 'mahi320/zomato' is highlighted, showing it was last pushed 3 minutes ago, is public, and is inactive. Other repositories listed are 'mahi320/starbucks', 'mahi320/netflix', and 'mahi320/mahesh-banking-app-ci-cd'.

The screenshot shows the detailed view for the 'mahi320/zomato' repository. The left sidebar remains the same. The main content area shows the repository's general information: last pushed 3 minutes ago, repository size 228 MB, and a note about using 0 of 1 private repositories. It includes sections for Docker commands (with a 'docker push' command example), Tags (listing 'latest' tag as an image type, pulled less than 1 day ago, and pushed 3 minutes ago), and a Buildcloud advertisement.

Webhooks - Administra x | Zomato-CI_CD-Pipeline... x | Projects - SonarQube C x | Explore Docker's Content x | React App x | + -

hub.docker.com/search?q=mahi320%2Fzomato

*** New Building AI Agents is Now Easy →**

hub Explore My Hub Search Docker Hub CtrlK M

Filter by 1 - 28 of 1,053 results for mahi320/zomato.

Products Images Extensions Plugins

Trusted content Docker Official Image Verified Publisher Sponsored OSS

Categories Networking Security Languages & frameworks Integration & delivery

mah320/zomato mah320

zomato/protoc-gen zomato

rcms/rcms-zomato-web-app rcms

iamejtas23/zomato-clone iamejtas23

acecloudacademy/zomato-ap... acecloudacademy

acecloudacademy/zomato-clo... acecloudacademy

kastrov/zomato kastrov

shaikmustafa/zomato shaikmustafa

Best match Give Feedback

https://hub.docker.com/r/mahi320/zomato

Webhooks - Administra x | Zomato-CI_CD-Pipeline... x | Projects - SonarQube C x | mahi320/zomato - Docker x | React App x | + -

hub.docker.com/r/mahi320/zomato

mahi320/zomato

By [mahi320](#) · Updated 3 minutes ago

No overview available

This repository doesn't have an overview

Tag summary

Recent tags latest

Content type Image

Digest sha256:217e05e15...

Docker pull command docker pull mahi320/zomato

Run in Docker Desktop

Requires Docker Desktop 4.37.1 or later.

#OUTPUT OF ZOMATO-APP

Successfully deployed in a docker container on ubuntu host machine

The screenshot shows the Zomato homepage. At the top, there's a navigation bar with links for 'Investor Relations', 'Add restaurant', 'Log in', and 'Sign up'. Below the header, the word 'zomato' is prominently displayed in a large, lowercase, bold font. Underneath it, the tagline 'Discover the best food & drinks in Patna' is visible. A search bar contains the text 'Chennai'. Below the main title, there are three images: a restaurant interior, silhouettes of people at a concert, and a plate of food.

The screenshot displays the Jenkins pipeline status for the 'Zomato-CI-CD-Pipeline-Project'. The top navigation bar includes tabs for 'Webhooks - Administration', 'Zomato-CI-CD-Pipeline-Project', 'mahi320/zomato - Docker', and 'React App'. The main content area shows the 'Stage View' for the pipeline stages: Declarative: Tool Install, Clean Workspace, Git Checkout, SonarQube Analysis, Install Dependencies, OWASP Dependency Check, Trivy Security Scan, Build Docker Image, Push to DockerHub, Deploy Container, Docker Scout Image, and Declarative: Post Actions. The 'Build Docker Image' stage is highlighted in blue. Below the stage view is the 'SonarQube Quality Gate' section, which shows a green 'Passed' status with 'server-side processing: Success'. The 'Builds' section lists a single build named 'zomato' with a status of 'Passed'. The 'Stages' section shows the execution history for build #1 on August 10, 2025, from 17:30 to 10:00. The 'Permalinks' section provides links to the last five builds.

Stage	Time	Status
Declarative: Tool Install	Aug 10 17:30	326ms
Clean Workspace		870ms
Git Checkout		8s
SonarQube Analysis		1min 9s
Install Dependencies		2min 6s
OWASP Dependency Check		972ms
Trivy Security Scan		6s
Build Docker Image		5min 26s
Push to DockerHub		1min 42s
Deploy Container		2s
Docker Scout Image		1s
Declarative: Post Actions		385ms

Webhooks - Administration | Zomato-CI_CD-Pipeline-Project | mahi320/zomato - Dockerfile | React App

localhost:8080/job/Zomato-CI_CD-Pipeline-Project/1/

Jenkins / Zomato-CI_CD-Pipeline-Project / #1

Status: #1 (10 Aug 2025, 17:30:17)

Started by user mahesh

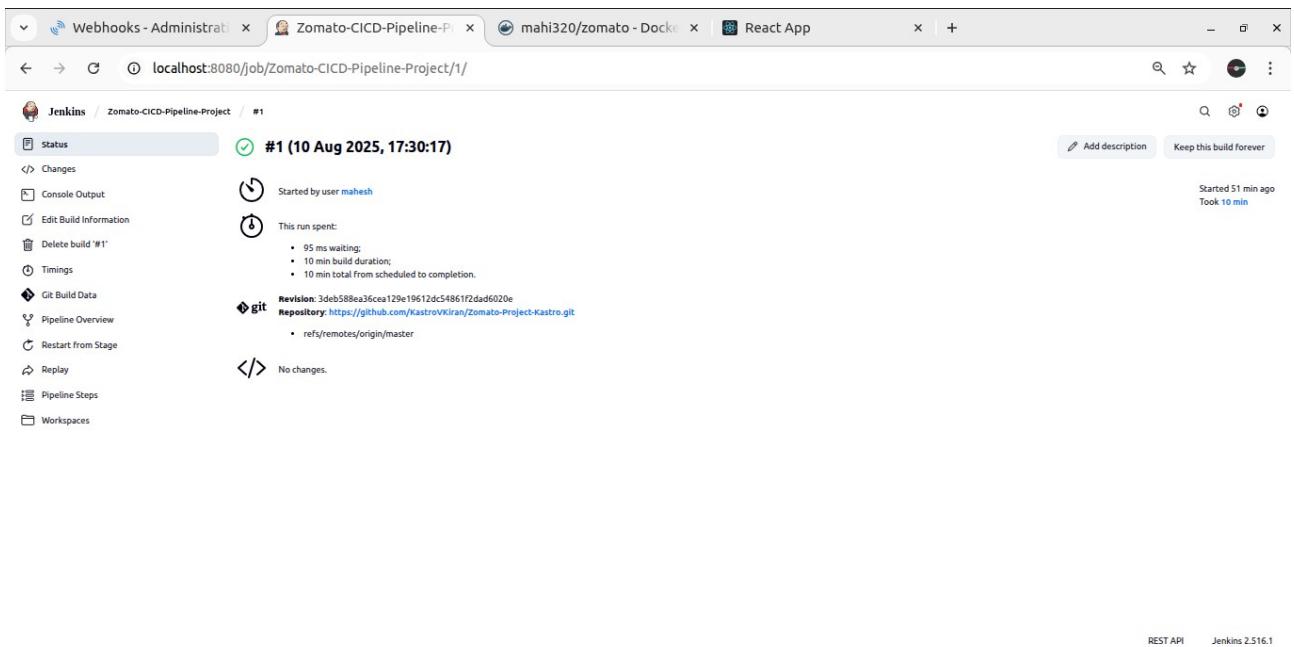
This run spent:

- 95 ms waiting;
- 10 min build duration;
- 10 min total from scheduled to completion.

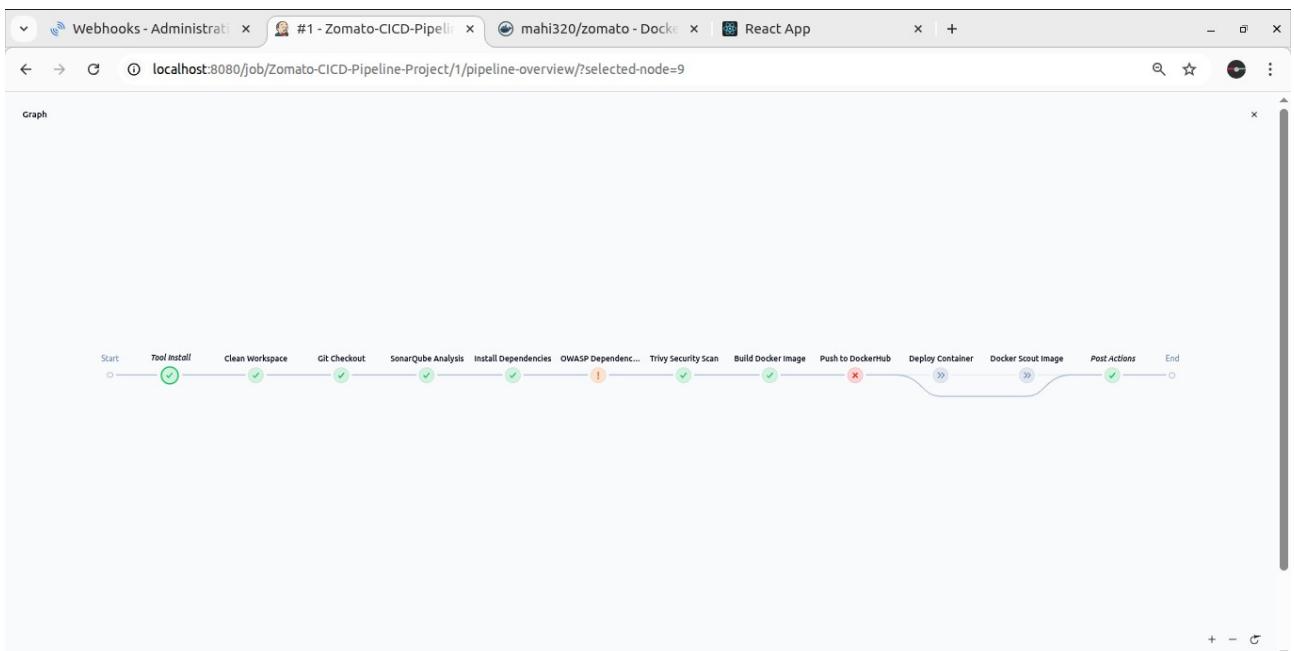
git Revision: 3deb588ea36cea129e19612dc54861f2dad6020e
Repository: https://github.com/kastrovKirjan/Zomato-Project-Kastro.git
refs/remotes/origin/master

</> No changes.

REST API Jenkins 2.516.1



The screenshot shows the Jenkins pipeline status page for job #1. It displays the build number (#1), timestamp (10 Aug 2025, 17:30:17), and the fact that it was started by user mahesh. It also shows the duration of the run, which includes waiting, build, and total times. The git section provides revision details, including the commit hash and repository URL. A note indicates 'No changes.' This is a standard Jenkins interface for monitoring the progress of a pipeline build.



Deploying to Kubernetes on Ubuntu Host Machine (Without AWS EKS)

Kubernetes Single Node Architecture with Zomato App Deployment
Stage-2

#List of my images in docker hub

The screenshot shows the Docker Hub interface. At the top, there's a blue header bar with the Docker logo, 'hub', 'Explore', 'My Hub', and a search bar. Below the header, a profile picture of a person with a blue 'M' is shown, followed by the username 'mahi320' and a link to their GitHub profile. The main area is titled 'Repositories' with a 'Starred' tab. A search bar says 'Search by repository name' with the placeholder 'Displaying 1 to 4 of 4 repositories'. There are four repository cards:

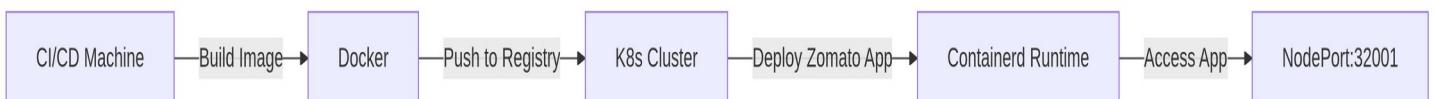
- mahi320/zomato**: A Node.js-based Starbucks clone application built with Docker and CI/CD pipeline. Last updated about 1 hour ago.
- mahi320/starbucks**: A Node.js-based Starbucks clone application built with Docker and CI/CD pipeline. Last updated 19 days ago.
- mahi320/netflix**: Jenkins-based secure CI/CD pipeline for deploying a Netflix-style streaming app. Last updated about 1 month ago.
- mahi320/maheh-banking-app-cicd**: Jenkins-based secure CI/CD pipeline for deploying a Netflix-style streaming app. Last updated over 1 year ago.

#Note:-

Recommended Setup for Production

CI/CD Server	Docker Only	Building container images	Jenkins/GitLab runners need Docker for image builds
---------------------	-------------	---------------------------	---

Application Runtime	Kubernetes Only	Running production workloads	Dedicated cluster (avoid mixing Docker and K8s on same nodes)
----------------------------	-----------------	------------------------------	---



1. Do install docker first after stop docker and kubernetes instal and deployment on ubuntu host machine

```
# Verify Docker is stopped  
sudo systemctl stop docker && sudo systemctl disable docker
```

On K8s nodes:

```
sudo systemctl is-active docker # Must return 'inactive'  
sudo systemctl is-active containerd # Must return 'active'
```

```
dell@mahesh:~$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.5 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.5 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
dell@mahesh:~$
```

#Deployment yaml files

```
Open ▾ F1 zomato.md ~/ Save ▾ Minimize ▾ Maximize ▾ Close ▾
020
627 Given your Zomato project is already built into a Docker image and pushed to Docker Hub, and you're working on an Ubuntu host,
628
629
630 #Installation of K8S on ubuntu-machine
631
632 # 1 kubeadm Setup on Ubuntu (Single Node)-Kubernetes (Single Node, kubeadm)
633 #Step 1: System Preparation
634 # Update packages
635 sudo apt update && sudo apt upgrade -y
636
637 # Disable swap
638 sudo swapoff -a
639 sudo sed -i '/ swap / s/^/#/' /etc/fstab
640
641 # Load required kernel modules
642 cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
643 overlay
644 br_netfilter
645 EOF
646 sudo modprobe overlay
647 sudo modprobe br_netfilter
648
649 # Set sysctl params for Kubernetes networking
650 cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
651 net.bridge.bridge-nf-call-iptables = 1
652 net.bridge.bridge-nf-call-ipTables = 1
653 net.ipv4.ip_forward = 1
654 EOF
655 sudo sysctl --system
656
657
658
659
```

Markdown ▾ Tab Width: 8 ▾ Ln 632, Col 1 ▾ INS

```
Open ▾  zomato.md ~/ Save    655 sudo sysctl --system  
656  
657  
658  
659  
660 # 2 Install Container Runtime (containerd)  
661 sudo apt install -y containerd  
662  
663 # Generate default containerd config  
664 sudo mkdir -p /etc/containerd  
665 containerd config default | sudo tee /etc/containerd/config.toml > /dev/null  
666  
667 # Enable systemd cgroup driver for kubelet compatibility  
668 sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc/containerd/config.toml  
669  
670 # Restart & enable containerd  
671 sudo systemctl restart containerd  
672 sudo systemctl enable containerd  
673  
674  
675  
676 # 3 Install Kubernetes Components  
677 # Install dependencies  
678 sudo apt install -y apt-transport-https ca-certificates curl gpg  
679  
680 # Add Kubernetes signing key  
681 sudo mkdir -p -m 755 /etc/apt/keyrings  
682 curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key \  
683 | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg  
684  
685 # Add Kubernetes apt repo  
686 echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] \  
687 https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /" \  
688 | sudo tee /etc/apt/sources.list.d/kubernetes.list  
689  
Markdown ▾ Tab Width: 8 ▾ Ln 674, Col 1 ▾ INS
```

```
Open ▾  zomato.md ~/ Save    670 # Restart & enable containerd  
671 sudo systemctl restart containerd  
672 sudo systemctl enable containerd  
673  
674  
675  
676 # 3 Install Kubernetes Components  
677 # Install dependencies  
678 sudo apt install -y apt-transport-https ca-certificates curl gpg  
679  
680 # Add Kubernetes signing key  
681 sudo mkdir -p -m 755 /etc/apt/keyrings  
682 curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key \  
683 | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg  
684  
685 # Add Kubernetes apt repo  
686 echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] \  
687 https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /" \  
688 | sudo tee /etc/apt/sources.list.d/kubernetes.list  
689  
690 # Install kubeadm, kubelet, kubectl  
691 sudo apt update  
692 sudo apt install -y kubelet kubeadm kubectl  
693 sudo apt-mark hold kubelet kubeadm kubectl  
694  
695  
696  
697  
698 # 4 Initialize Control Plane  
699 sudo kubeadm reset -f  
700 sudo systemctl restart containerd  
701  
702 sudo kubeadm init \  
703 --pod-network-cidr=10.240.0.0/16 \  
Markdown ▾ Tab Width: 8 ▾ Ln 694, Col 1 ▾ INS
```

```
Open ▾  ↗ zomato.md Save ⌂ ⌃ ⌄ ⌅ ⌆ ⌇
687 https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /" \
688 | sudo tee /etc/apt/sources.list.d/kubernetes.list
689
690 # Install kubeadm, kubelet, kubectl
691 sudo apt update
692 sudo apt install -y kubelet kubeadm kubectl
693 sudo apt-mark hold kubelet kubeadm kubectl
694
695
696 # 4 Initialize Control Plane
697 sudo kubeadm reset -f
698 sudo systemctl restart containerd
699
700 sudo kubeadm init \
701 --pod-network-cidr=192.168.0.0/16 \
702 --cri-socket unix:///run/containerd/containerd.sock
703
704
705
706 # 5 Configure kubectl for Your User
707 mkdir -p $HOME/.kube
708 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
709 sudo chown $(id -u):$(id -g) $HOME/.kube/config
710
711
712 # 6 Install Calico CNI (Pod Networking)
713 kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/calico.yaml
714
715
716
717 # 7 (Single Node Only) Allow Scheduling on Control Plane
718 kubectl taint nodes --all node-role.kubernetes.io/control-plane-
719
720
721
722
723
724
725
726
727
728 cd ~
729 mkdir zomato-k8s
730 cd zomato-k8s
731
732
```

Markdown ▾ Tab Width: 8 ▾ Ln 719, Col 1 ▾ INS

```
719
720
721 kubectl run nginx --image=nginx --port=80
722 kubectl expose pod nginx --type=NodePort --port=80
723
724
725 kubectl get pods
726
727
728 cd ~
729 mkdir zomato-k8s
730 cd zomato-k8s
731
732
```

```
Open ▾  ↗ zomato.md Save ⌂ ⌃ ⌄ ⌅ ⌆ ⌇
726
727
728 cd ~
729 mkdir zomato-k8s
730 cd zomato-k8s
731
732
733 #How to create YAML files?
734 sudo nano zomato-deployment.yaml
735
736 apiVersion: apps/v1
737 kind: Deployment
738 metadata:
739   name: zomato-app
740 spec:
741   replicas: 1
742   selector:
743     matchLabels:
744       app: zomato
745   template:
746     metadata:
747       labels:
748         app: zomato
749     spec:
750       containers:
751         - name: zomato-contalner
752           image: mahl320/zomato:latest
753         ports:
754           - containerPort: 3000      # changed to 3000 to match app
755
756
757
758 sudo nano zomato-service.yaml
759 apiVersion: v1
760 kind: Service
```

Markdown ▾ Tab Width: 8 ▾ Ln 757, Col 1 ▾ INS

```
#service.yaml
```

```
#Install k8s on ubuntu machine
```

```
dell@mahesh:~$ sudo apt-get update
dell@mahesh:~$ 122x27
dell@mahesh:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 https://aquasecurity.github.io/trivy-repo/deb jammy InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:7 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:8 https://packages.grafana.com/oss/deb stable InRelease
Hit:9 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:10 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:11 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:12 https://ppa.launchpadcontent.net/marin-m/songrec/ubuntu jammy InRelease
Hit:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb InRelease
Reading package lists... Done
dell@mahesh:~$ 
```



```
755
756
757
758 sudo nano zomato-service.yaml
759 apiVersion: v1
760 kind: Service
761 metadata:
762   name: zomato-service
763 spec:
764   selector:
765     app: zomato
766   type: NodePort
767   ports:
768     - protocol: TCP
769       port: 3001      # external service port (can stay 3001)
770       targetPort: 3000 # must match containerPort/app port inside pod
771       nodePort: 32001
772
773
774 #Apply the YAML files
775 kubectl apply -f zomato-deployment.yaml
776 kubectl apply -f zomato-service.yaml
777
778
779 #note
780 - Kubernetes will try to pull the image specified in your deployment YAML from Docker Hub (or any other container registry if specified).
781
782
783 #How it works:
784 Your zomato-deployment.yaml references the image:
785
786 #Image: mahi320/zomato:latest
787 When Kubernetes schedules the pod, the kubelet on the node will check if this image exists locally.
788 
```

Markdown ▾ Tab Width: 8 ▾ Ln 778, Col 1 ▾ INS

```
dell@mahesh: ~
dell@mahesh: ~ 135x30
kernel.core_uses_pid = 1
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.default.accept_source_route = 0
sysctl: setting key "net.ipv4.conf.all.accept_source_route": Invalid argument
net.ipv4.conf.default.promote_secondaries = 1
sysctl: setting key "net.ipv4.conf.all.promote_secondaries": Invalid argument
net.ipv4.ping_group_range = 0 2147483647
net.core.default_qdisc = fq_codel
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
fs.protected_regular = 1
fs.protected_fifos = 1
* Applying /usr/lib/sysctl.d/50-pid-max.conf ...
kernel.pid_max = 4194304
* Applying /usr/lib/sysctl.d/99-protect-links.conf ...
fs.protected_fifos = 1
fs.protected_hardlinks = 1
fs.protected_regular = 2
fs.protected_symlinks = 1
* Applying /etc/sysctl.d/99-sysctl.conf ...
* Applying /etc/sysctl.d/k8s.conf ...
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
* Applying /etc/sysctl.d/kubernetes.conf ...
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
* Applying /etc/sysctl.conf ...
dell@mahesh:~$
```

```
dell@mahesh: ~
dell@mahesh: ~ 135x30
dell@mahesh:~$ sudo apt install -y containerd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  runc
The following packages will be REMOVED:
  containerd.io docker-ce
The following NEW packages will be installed:
  containerd runc
0 upgraded, 2 newly installed, 2 to remove and 0 not upgraded.
Need to get 45.9 MB of archives.
After this operation, 42.8 MB disk space will be freed.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.2.5-0ubuntu1~22.04.1 [8,093 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.27-0ubuntu1~22.04.1 [37.8 MB]
Fetched 45.9 MB in 13s (3,437 kB/s)
(Reading database ... 270316 files and directories currently installed.)
Removing docker-ce (5:28.3.3-1~ubuntu.22.04~jammy) ...
[Progress: [ 15%] [#####.....]
```

#Containerized

```
dell@mahesh: /etc/containerd$ sudo systemctl restart containerd
dell@mahesh: /etc/containerd$ sudo systemctl enable containerd
dell@mahesh: /etc/containerd$ sudo systemctl start containerd
dell@mahesh: /etc/containerd$ sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/lib/systemd/system/containerd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-08-10 19:09:51 IST; 1s ago
     Docs: https://containerd.io
     Main PID: 34337 (containerd)
        Tasks: 9
       Memory: 13.8M
          CPU: 506ms
        CGroup: /system.slice/containerd.service
                 └─34337 /usr/bin/containerd

Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.084080740+05:30" level=error msg="failed to load cni during init, >
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.086363507+05:30" level=info msg=serving... address=/run/containerd>
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.086658132+05:30" level=info msg=serving... address=/run/containerd>
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.088806818+05:30" level=info msg="Start subscribing containerd even>
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.089207326+05:30" level=info msg="Start recovering state"
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.092534665+05:30" level=info msg="Start event monitor"
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.095025961+05:30" level=info msg="Start snapshots syncer"
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.096526269+05:30" level=info msg="Start cni network conf syncer for"
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.096602429+05:30" level=info msg="Start streaming server"
Aug 10 19:09:51 mahesh containerd[34337]: time="2025-08-10T19:09:51.097027359+05:30" level=info msg="containerd successfully booted in<
dell@mahesh: /etc/containerd$
```

```
dell@mahesh: ~$ # Install dependencies
dell@mahesh: ~$ sudo apt install -y apt-transport-https ca-certificates curl gpg
dell@mahesh: ~$ Reading package lists... Done
dell@mahesh: ~$ Building dependency tree... Done
dell@mahesh: ~$ Reading state information... Done
dell@mahesh: ~$ ca-certificates is already the newest version (20240203~22.04.1).
dell@mahesh: ~$ curl is already the newest version (7.81.0-1ubuntu1.20).
dell@mahesh: ~$ gpg is already the newest version (2.2.27-3ubuntu2.4).
dell@mahesh: ~$ apt-transport-https is already the newest version (2.4.14).
dell@mahesh: ~$ The following packages were automatically installed and are no longer required:
dell@mahesh: ~$   docker-ce-rootless-extras libssl1.0.2g slirp4netns
dell@mahesh: ~$ Use 'sudo apt autoremove' to remove them.
dell@mahesh: ~$ 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
dell@mahesh: ~$ # Add Kubernetes signing key
dell@mahesh: ~$ sudo mkdir -p -m 755 /etc/apt/keyrings
dell@mahesh: ~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key \
dell@mahesh: ~$   | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
dell@mahesh: ~$ 
dell@mahesh: ~$ # Add Kubernetes apt repo
dell@mahesh: ~$ echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] \
dell@mahesh: ~$   https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /" \
dell@mahesh: ~$   | sudo tee /etc/apt/sources.list.d/kubernetes.list
dell@mahesh: ~$ File '/etc/apt/sources.list.d/kubernetes.list' exists. Overwrite? (y/N) y
dell@mahesh: ~$ deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.30/deb/ /
dell@mahesh: ~$
```

```
#install`kubectl, kubelet, kubeadm
```

```
dell@mahesh:~$ sudo apt install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 93.8 MB of archives.
After this operation, 343 MB of additional disk space will be used.
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]
Get:1 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb cri-tools 1.30.1-1.1 [21.3 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb kubeadm 1.30.14-1.1 [10.5 MB]
30% [3 kubeadm 6,472 kB/10.5 MB 62%] 1,410 kB/s 46s
```

```
dell@mahesh:~$ sudo apt install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  conntrack cri-tools kubeadm kubelet kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 93.8 MB of archives.
After this operation, 343 MB of additional disk space will be used.
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 conntrack amd64 1:1.4.6-2build2 [33.5 kB]
Get:1 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb cri-tools 1.30.1-1.1 [21.3 MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb kubeadm 1.30.14-1.1 [10.5 MB]
Get:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb kubelet 1.30.14-1.1 [10.9 MB]
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb kubernetes-cni 1.4.0-1.1 [32.9 MB]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb kubelet 1.30.14-1.1 [10.2 MB]
Fetched 93.8 MB in 31s (2,979 kB/s)
Selecting previously unselected package conntrack.
(Reading database ... 270359 files and directories currently installed.)
Preparing to unpack .../conntrack_1.4.6-2build2_amd64.deb ...
Unpacking conntrack (1:1.4.6-2build2) ...
Selecting previously unselected package cri-tools.
Preparing to unpack .../cri-tools_1.30.1-1.1_amd64.deb ...
Unpacking cri-tools (1.30.1-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../2-kubeadm_1.30.14-1.1_amd64.deb ...
Unpacking kubeadm (1.30.14-1.1) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../3-kubelet_1.30.14-1.1_amd64.deb ...
Unpacking kubelet (1.30.14-1.1) ...
Selecting previously unselected package kubernetes-cni.
Preparing to unpack .../4-kubernetes-cni_1.4.0-1.1_amd64.deb ...
Unpacking kubernetes-cni (1.4.0-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../5-kubectl_1.30.14-1.1_amd64.deb ...
Unpacking kubectl (1.30.14-1.1) ...
Setting up conntrack (1:1.4.6-2build2) ...
Setting up kubelet (1.30.14-1.1) ...
Setting up cri-tools (1.30.1-1.1) ...
Setting up kubernetes-cni (1.4.0-1.1) ...
```

```
Unpacking kubelet (1.30.14-1.1) ...
Setting up conntrack (1:1.4.6-2build2) ...
Setting up kubectl (1.30.14-1.1) ...
Setting up cri-tools (1.30.1-1.1) ...
Setting up kubernetes-cni (1.4.0-1.1) ...
Setting up kubeadm (1.30.14-1.1) ...
Setting up kubelet (1.30.14-1.1) ...
Processing triggers for man-db (2.10.2-1) ...
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
dell@mahesh:~$
```

```
dell@mahesh:~ dell@mahesh: ~ 135x30
^C
dell@mahesh:~$ sudo kubeadm init --pod-network-cidr=192.168.0.0/16 --cri-socket unix:///run/containerd/containerd.sock
I0810 19:14:26.301419 35846 version.go:256] remote version is much newer: v1.33.3; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.14
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0810 19:14:29.121805 35846 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is in
consistent with that used by kubeadm. It is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Using existing ca certificate authority
[certs] Using existing apiserver certificate and key on disk
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [localhost mahesh] and IPs [192.168.1.150 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [localhost mahesh] and IPs [192.168.1.150 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
```

```
dell@mahesh:~ dell@mahesh: ~ 135x30
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate cre
dentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.1.150:6443 --token 4rmfo5.r7kkm77bd3dez29w \
  --discovery-token-ca-cert-hash sha256:239048c966f27f504b073a29d5dca741251cb7d3f9f4a72cd49f656ac1a41d29
dell@mahesh:~$
```

##Configure kubectl for Your User

```
dell@mahesh:~$ # 5 Configure kubectl for Your User
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

dell@mahesh:~$ # 6 Install Calico CNI (Pod Networking)
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
serviceaccount/calico-cni-plugin created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodesstatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/iptables.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
```

##Run a Temporary Nginx Pod (for testing)

```
dell@mahesh:~$ kubectl get pods
NAME    READY  STATUS    RESTARTS   AGE
nginx  1/1    Running   0          110s
dell@mahesh:~$ kubectl get nodes
NAME      STATUS  ROLES   AGE     VERSION
mahesh  Ready   control-plane  3m20s  v1.30.14
dell@mahesh:~$
```

```
#deployment.yaml
```

```
dell@mahesh: ~/zomato-k8s
dell@mahesh: ~/zomato-k8s 122x27
zomato-deployment.yaml

GNU nano 6.2
apiVersion: apps/v1
kind: Deployment
metadata:
  name: zomato-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: zomato
  template:
    metadata:
      labels:
        app: zomato
    spec:
      containers:
        - name: zomato-container
          image: mahi320/zomato:latest
          ports:
            - containerPort: 3000      # changed to 3000 to match app

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^I Go To Line M-E Redo
                                         M-A Set Mark
                                         M-6 Copy
```

```
#service.yaml
```

```
dell@mahesh: ~/zomato-k8s
dell@mahesh: ~/zomato-k8s 122x27
zomato-service.yaml

GNU nano 6.2
apiVersion: v1
kind: Service
metadata:
  name: zomato-service
spec:
  selector:
    app: zomato
  type: NodePort
  ports:
    - protocol: TCP
      port: 3001      # external service port (can stay 3001)
      targetPort: 3000 # must match containerPort/app port inside pod
      nodePort: 32001
```

```
^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^I Go To Line M-E Redo
                                         M-A Set Mark
                                         M-6 Copy
```

##Apply the YAML files

```
dell@mahesh: ~/zomato-k8s$ ls
zomato-deployment.yaml zomato-service.yaml
dell@mahesh:~/zomato-k8s$ kubectl apply -f zomato-deployment.yaml
deployment.apps/zomato-app created
dell@mahesh:~/zomato-k8s$ kubectl apply -f zomato-service.yaml
service/zomato-service created
dell@mahesh:~/zomato-k8s$
```

#output

```
dell@mahesh: ~/zomato-k8s$ kubectl get pods
dell@mahesh: ~/zomato-k8s 122x27
NAME           READY   STATUS    RESTARTS   AGE
nginx          1/1     Running   0          6m25s
zomato-app-64fc466984-ln6g2 1/1     Running   0          2m11s
dell@mahesh:~/zomato-k8s$ kubectl get nodes
NAME      STATUS   ROLES   AGE   VERSION
mahesh   Ready    control-plane   8m4s   v1.30.14
dell@mahesh:~/zomato-k8s$ kubectl get svc
NAME        TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      8m5s
nginx       NodePort   10.104.233.38  <none>        80:30312/TCP  6m34s
zomato-service  NodePort   10.102.147.96  <none>        3001:32001/TCP  2m20s
dell@mahesh:~/zomato-k8s$
```

#logs

```
dell@mahesh: ~/zomato-k8s$ kubectl get pods -l app=zomato
dell@mahesh: ~/zomato-k8s$ kubectl logs zomato-app-64fc466984-ln6g2
dell@mahesh: ~/zomato-k8s$ kubectl logs zomato-app-64fc466984-ln6g2

> zomato_clone@0.1.0 start
> react-scripts start

(node:26) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:26) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...

Compiled successfully!

You can now view zomato_clone in the browser.

  Local:          http://localhost:3000
  On Your Network: http://192.168.5.133:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
Compiling...
Compiled successfully!
webpack compiled successfully
dell@mahesh: ~/zomato-k8s$
```

#Output of zomato-app deployment in k8s



#docker

The screenshot shows the Zomato homepage with a banner featuring various food dishes. The banner includes the text "Get the App", "Investor Relations", "Add restaurant", "Log in", and "Sign up". Below the banner is a search bar with "Chennai" selected and a placeholder "Search for restaurant, cuisine or a dish". A large "zomato" logo is centered with the tagline "Discover the best food & drinks in Patna". Below the main banner are three smaller images with captions: "Order Online", "Nightlife and Clubs", and "Dinning".

The screenshot shows a page encouraging users to download the Zomato app. The heading is "Get the Zomato app". Below it, a message says "We will send you a link, open it on your phone to download the app". There are two radio button options: "Email" and "Phone", with "Email" selected. A "Share App Link" button is next to the input field. Below this, there's a "Download app from" section with links to "GET IT ON Google Play" and "Download on the App Store".

Explore options near me

Popular cuisines near me

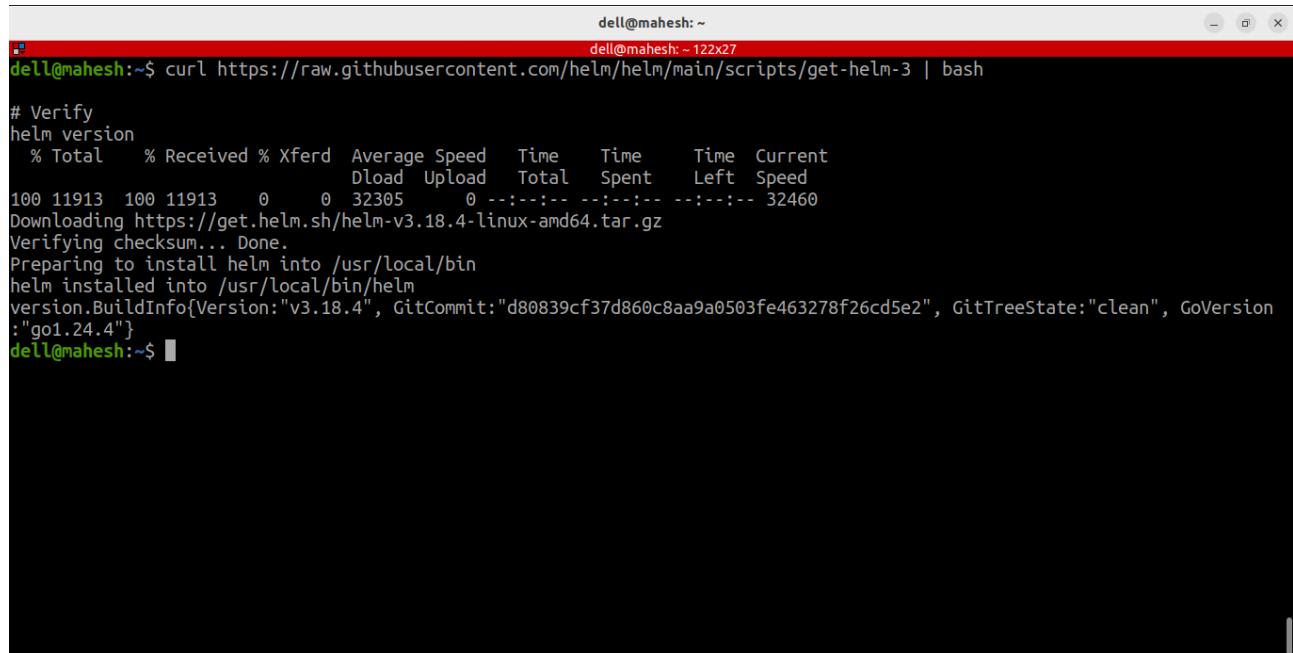
Setup prometheus and grafana

Kubernetes (kubeadm) is running, deploy Prometheus + Grafana using Helm on Ubuntu Machine

STAGE-2

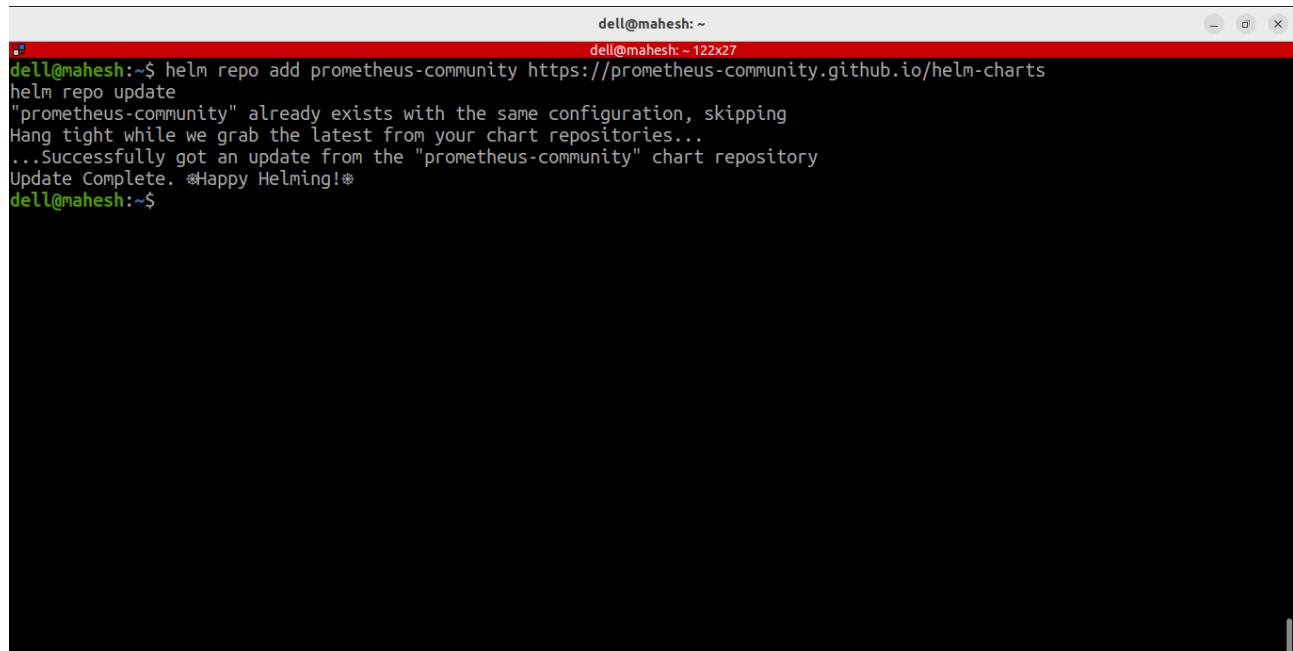
1. Prometheus collects real-time metrics from Kubernetes components, nodes, and pods.
2. Grafana visualizes those metrics in easy-to-read dashboards.

```
##install helm
```



```
dell@mahesh:~ dell@mahesh:~ 122x27
dell@mahesh:~$ curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
# Verify
helm version
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
100 11913  100 11913    0      0 32305      0 --::-- --::-- --::-- 32460
Downloading https://get.helm.sh/helm-v3.18.4-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
version.BuildInfo{Version:"v3.18.4", GitCommit:"d80839cf37d860c8aa9a0503fe463278f26cd5e2", GitTreeState:"clean", GoVersion :"go1.24.4"}
dell@mahesh:~$
```

Add Prometheus Community Helm repo



```
dell@mahesh:~ dell@mahesh:~ 122x27
dell@mahesh:~$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo update
"prometheus-community" already exists with the same configuration, skipping
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. *Happy Helming!*
dell@mahesh:~$
```

Install kube-prometheus-stack (Prometheus + Grafana)

```
dell@mahesh:~$ # Create monitoring namespace
kubectl create namespace monitoring

# Install chart
helm install monitoring prometheus-community/kube-prometheus-stack \
--namespace monitoring
namespace/monitoring created
NAME: monitoring
LAST DEPLOYED: Mon Aug 11 11:43:16 2025
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace monitoring get pods -l "release=monitoring"

Get Grafana 'admin' user password by running:
  kubectl --namespace monitoring get secrets monitoring-grafana -o jsonpath=".data.admin-password" | base64 -d ; echo

Access Grafana local instance:
  export POD_NAME=$(kubectl --namespace monitoring get pod -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=monitoring" -o name)
  kubectl --namespace monitoring port-forward $POD_NAME 3000

Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus
instances using the Operator.
dell@mahesh:~$
```

4 Access Grafana 5 Access Prometheus

```
dell@mahesh:~$ kubectl get secret monitoring-grafana -n monitoring \
--o jsonpath=".data.admin-password" | base64 --decode; echo
prom-operator
dell@mahesh:~$ kubectl port-forward svc/monitoring-grafana -n monitoring 3000:80
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000

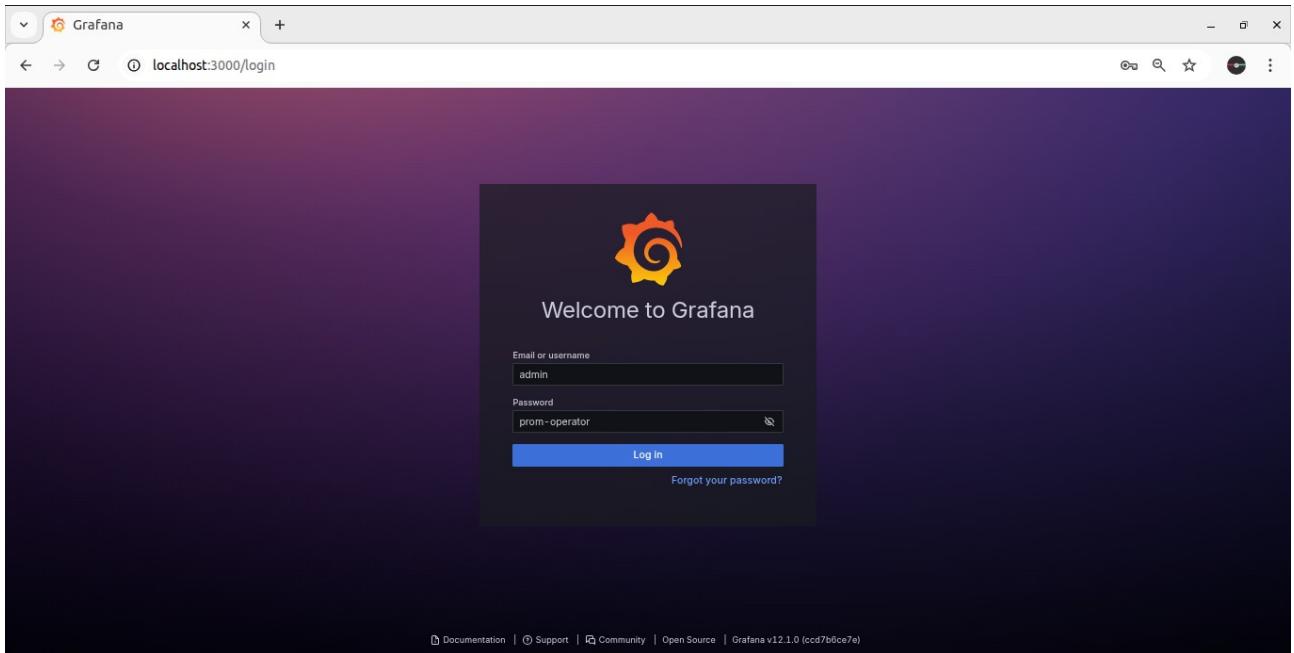
dell@mahesh:~$ kubectl port-forward svc/monitoring-kube-prometheus-prometheus -n monitoring 9090:9090
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
```

#Browser

http://localhost:3000

<http://localhost:9090>

#Grafana



A screenshot of a web browser window showing the Grafana Home - Dashboards page. The title bar says "Home - Dashboards - Gra...". The address bar shows "localhost:3000/?orgId=1&from=now-6h&to=now&timezone=browser". The left sidebar has a "Home" tab selected, along with Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections, and Administration. The main content area has a "Welcome to Grafana" header. On the right, there's a "Need help?" section with links to Documentation, Tutorials, Community, and Public Slack. Below that are three cards: "Basic" (TUTORIAL DATA SOURCE AND DASHBOARDS, Grafana fundamentals), "COMPLETE Add your first data source", and "COMPLETE Create your first dashboard". At the bottom, there are sections for "Dashboards" (Starred dashboards, Recently viewed dashboards) and "Latest from the blog" (an article by Wilfried Roset titled "How to use SQL to learn more about your Grafana usage").

#Prometheus

The screenshot shows the Grafana interface at localhost:3000/dashboards. The left sidebar is open, showing the navigation menu with 'Dashboards' selected. The main area displays a list of dashboards under the heading 'Dashboards'. A search bar at the top of the list allows filtering by name or tag. To the right of the list, a 'Tags' section shows color-coded tags for each dashboard: purple for 'alertmanager-mixin', green for 'coredns dns', red for 'etcd-mixin', and various shades of blue for different Kubernetes components like 'kubernetes-mixin'. At the bottom of the screen, there is a query editor with a 'Query' input field containing placeholder text 'Enter expression (press Shift+Enter for newlines)', a 'Table' button, a 'Graph' button, an 'Alerts' button, a 'Status' button, and an 'Execute' button.

Screenshot of the Prometheus Time Series interface showing target health status.

The interface displays four service monitors:

- serviceMonitor/monitoring/monitoring-kube-prometheus-alertmanager/0**: Status: 1 / 1 up. Last scrape: 4.498s ago. State: UP.
- serviceMonitor/monitoring/monitoring-kube-prometheus-alertmanager/1**: Status: 1 / 1 up. Last scrape: 16.597s ago. State: UP.
- serviceMonitor/monitoring/monitoring-kube-prometheus-apiserver/0**: Status: 1 / 1 up. Last scrape: 28.658s ago. State: UP.
- serviceMonitor/monitoring/monitoring-kube-prometheus-coredns/0**: Status: 2 / 2 up. Last scrape: 19.038s ago. State: UP.

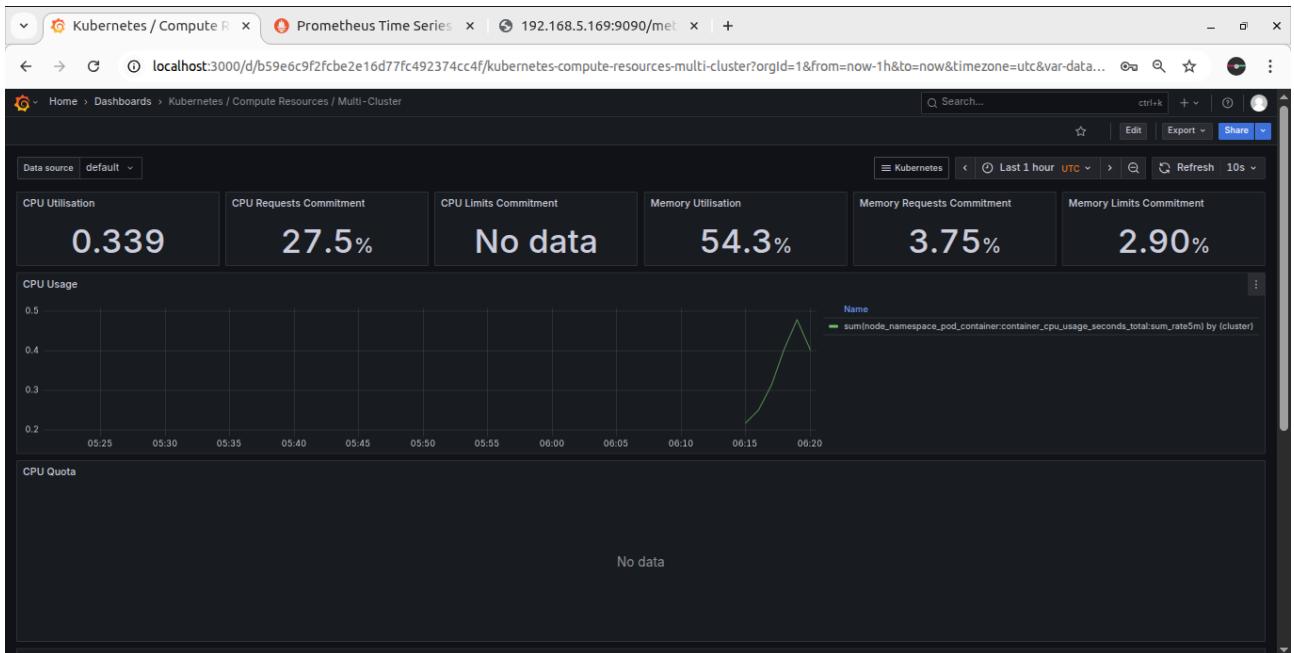
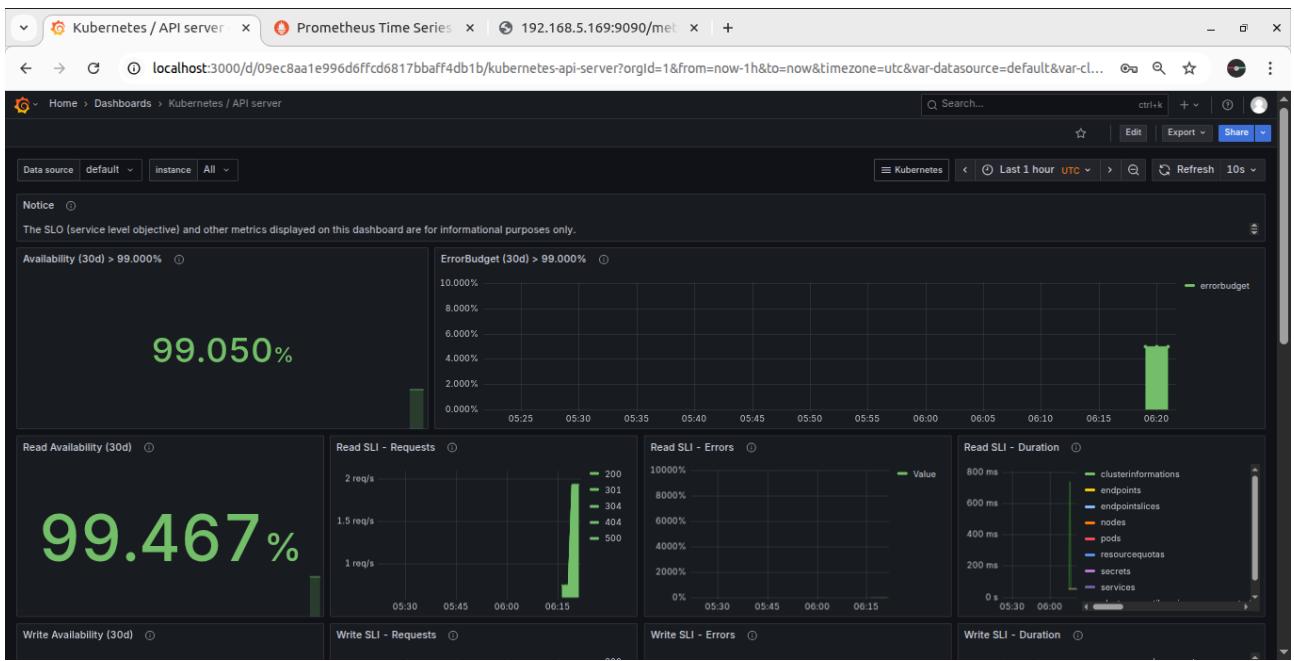
Each entry shows the endpoint URL, labels (container, endpoint, instance, job, namespace, pod), and last scrape time.

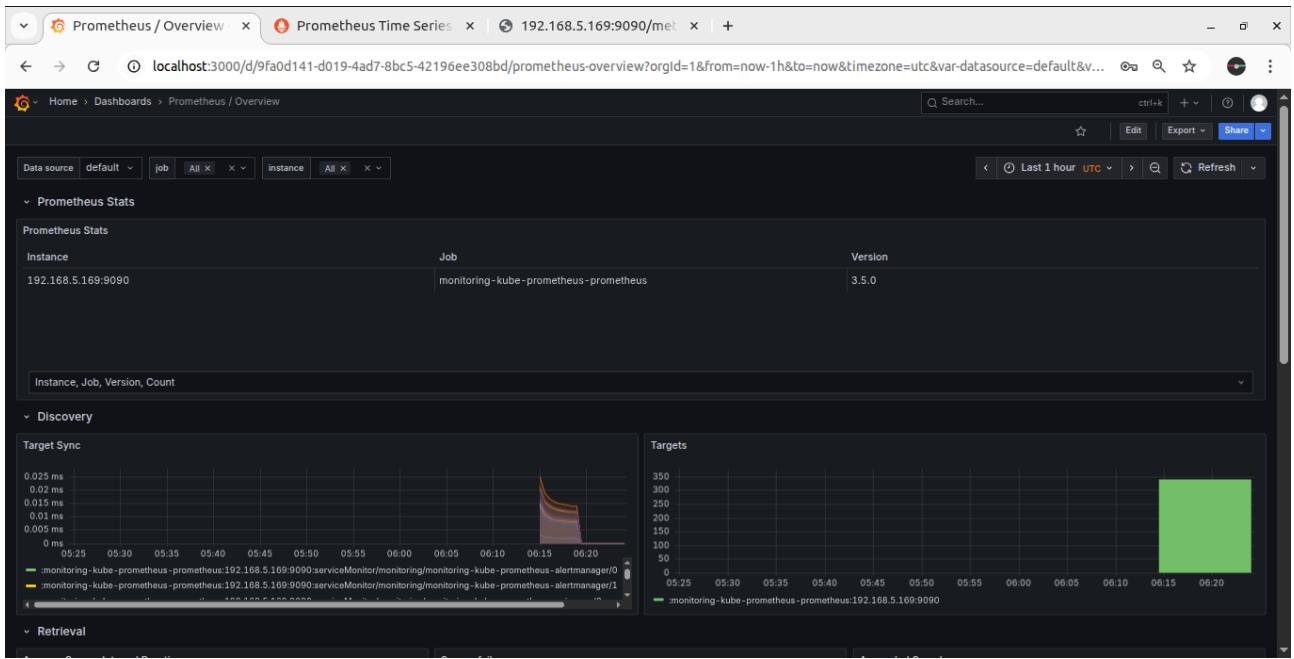
Screenshot of the Grafana Dashboards interface showing a list of available dashboards.

The interface displays a list of dashboards under the "Dashboards" section:

- Alertmanager / Overview
- CoreDNS
- etcfd
- Grafana Overview
- Kubernetes / API server
- Kubernetes / Compute Resources / Multi-Cluster
- Kubernetes / Compute Resources / Cluster
- Kubernetes / Compute Resources / Namespace (Pods)
- Kubernetes / Compute Resources / Namespace (Workloads)
- Kubernetes / Compute Resources / Node (Pods)
- Kubernetes / Compute Resources / Pod
- Kubernetes / Compute Resources / Workload
- Kubernetes / Controller Manager
- Kubernetes / Kubelet

Each dashboard entry includes a preview icon, a title, and a list of associated tags on the right side.





```
dell@mahesh:~$ kubectl get secret monitoring-grafana -n monitoring \
-o jsonpath="{.data.admin-password}" | base64 --decode; echo
prom-operator
dell@mahesh:~$ kubectl port-forward svc/monitoring-grafana -n monitoring 3000:80
Forwarding from 127.0.0.1:3000 -> 3000
Forwarding from [::1]:3000 -> 3000
Handling connection for 3000
dell@mahesh:~$ ^C
dell@mahesh:~$ kubectl port-forward svc/monitoring-kube-prometheus-prometheus -n monitoring 9090:9090
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
^C
dell@mahesh:~$ ^C
dell@mahesh:~$ kubectl port-forward svc/monitoring-kube-prometheus-prometheus -n monitoring 9090:9090
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
Handling connection for 9090
```

```
dell@mahesh:~$ kubectl get all -A
dell@mahesh:~
```

NAME	READY	STATUS	RESTARTS	AGE
default pod/nginx	1/1	Running	0	3h4m
default pod/zomato-app-64fc466984-gcfr6	1/1	Running	0	3h13m
kube-system pod/calico-kube-controllers-78b7fdb74b-hnn6v	1/1	Running	14 (76m ago)	3h15m
kube-system pod/calico-node-bm59f	1/1	Running	0	3h15m
kube-system pod/coredns-55cb5b774-n2c8c	1/1	Running	0	3h16m
kube-system pod/coredns-55cb5b774-rlc5q	1/1	Running	0	3h16m
kube-system pod/etc-d-nahesh	1/1	Running	4	3h17m
kube-system pod/kube-apiserver-nahesh	1/1	Running	13 (72m ago)	3h17m
kube-system pod/kube-controller-manager-nahesh	1/1	Running	1 (112m ago)	3h17m
kube-system pod/kube-proxy-jb2nb	1/1	Running	0	3h16m
kube-system pod/kube-scheduler-nahesh	1/1	Running	5 (112m ago)	3h17m
monitoring pod/alertmanager-monitoring-kube-prometheus-alertmanager-0	2/2	Running	0	130m
monitoring pod/monitoring-grafana-65dd9d9c59-gm4tg	3/3	Running	0	130m
monitoring pod/monitoring-kube-prometheus-operator-7984fbdd4d-shqv	1/1	Running	0	130m
monitoring pod/monitoring-kube-state-metrics-7d84c55c59-rcsgx	1/1	Running	14 (75m ago)	130m
monitoring pod/monitoring-prometheus-node-exporter-xrgn5	0/1	CrashLoopBackOff	30 (105s ago)	130m
monitoring pod/prometheus-monitoring-kube-prometheus-prometheus-0	2/2	Running	0	130m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
default service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	3h17m
default service/nginx	NodePort	10.109.179.192	<none>	80:30665/TCP	3h4m
default service/zomato-service	NodePort	10.110.192.255	<none>	3001:32001/TCP	3h13m
kube-system service/kube-dns	ClusterIP	10.96.0.10	<none>	53/UDP,53/TCP,9153/TCP	3h17m
kube-system service/monitoring-kube-prometheus-coredns	ClusterIP	None	<none>	9153/TCP	130m
kube-system service/monitoring-kube-prometheus-kube-controller-manager	ClusterIP	None	<none>	10257/TCP	130m
kube-system service/monitoring-kube-prometheus-kube-etcdd	ClusterIP	None	<none>	2381/TCP	130m
kube-system service/monitoring-kube-prometheus-kube-proxy	ClusterIP	None	<none>	10249/TCP	130m
kube-system service/monitoring-kube-prometheus-kube-scheduler	ClusterIP	None	<none>	10259/TCP	130m
monitoring service/alertmanager-operated	ClusterIP	None	<none>	10250/TCP,10255/TCP,4194/TCP	143m
monitoring service/monitoring-grafana	ClusterIP	10.101.219.244	<none>	80/TCP	130m
monitoring service/monitoring-kube-prometheus-alertmanager	ClusterIP	10.100.2.54	<none>	9093/TCP,8080/TCP	130m
monitoring service/monitoring-kube-prometheus-operator	ClusterIP	10.98.203.37	<none>	443/TCP	130m
monitoring service/monitoring-kube-prometheus-prometheus	ClusterIP	10.106.24.168	<none>	9090/TCP,8080/TCP	130m
monitoring service/monitoring-kube-state-metrics	ClusterIP	10.106.68.46	<none>	8080/TCP	130m
monitoring service/monitoring-prometheus-node-exporter	ClusterIP	10.108.207.12	<none>	9100/TCP	130m
monitoring service/prometheus-operated	ClusterIP	None	<none>	9090/TCP	130m

```
dell@mahesh:~$ kubectl get nodes -o wide
dell@mahesh:~
```

NAME	READY	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
nahesh	Ready	Control-Plane	3h17m	v1.30.14	192.168.1.150	<none>	Ubuntu 22.04.5 LTS	6.8.0-65-generic	containerd://1.7.27	


```
dell@mahesh:~$ kubectl get pods -A -o wide
dell@mahesh:~
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED-NODE	READYNESS	GATES
default nginx	1/1	Running	0	3h5m	192.168.5.133	nahesh	<none>	<none>	<none>
default zomato-app-64fc466984-gcfr6	1/1	Running	0	3h13m	192.168.5.132	nahesh	<none>	<none>	<none>
kube-system calico-kube-controllers-78b7fdb74b-hnn6v	1/1	Running	14 (76m ago)	3h16m	192.168.5.130	nahesh	<none>	<none>	<none>
kube-system calico-node-bm59f	1/1	Running	0	3h16m	192.168.1.150	nahesh	<none>	<none>	<none>
kube-system coredns-55cb5b774-n2c8c	1/1	Running	0	3h17m	192.168.5.129	nahesh	<none>	<none>	<none>
kube-system coredns-55cb5b774-rlc5q	1/1	Running	0	3h17m	192.168.5.131	nahesh	<none>	<none>	<none>
kube-system etcd-nahesh	1/1	Running	4	3h17m	192.168.1.150	nahesh	<none>	<none>	<none>
kube-system kube-apiserver-nahesh	1/1	Running	13 (73m ago)	3h17m	192.168.1.150	nahesh	<none>	<none>	<none>
kube-system kube-controller-manager-nahesh	1/1	Running	1 (113m ago)	3h17m	192.168.1.150	nahesh	<none>	<none>	<none>
kube-system kube-proxy-jb2nb	1/1	Running	0	3h17m	192.168.1.150	nahesh	<none>	<none>	<none>
kube-system kube-scheduler-nahesh	1/1	Running	5 (113m ago)	3h17m	192.168.1.150	nahesh	<none>	<none>	<none>
monitoring alertmanager-monitoring-kube-prometheus-alertmanager-0	2/2	Running	0	130m	192.168.5.167	nahesh	<none>	<none>	<none>
monitoring monitoring-grafana-65dd9d9c59-gm4tg	3/3	Running	0	130m	192.168.5.166	nahesh	<none>	<none>	<none>
monitoring monitoring-kube-prometheus-operator-7984fbdd4d-shqv	1/1	Running	0	130m	192.168.5.165	nahesh	<none>	<none>	<none>
monitoring monitoring-kube-state-metrics-7d84c55c59-rcsgx	1/1	Running	14 (76m ago)	130m	192.168.5.164	nahesh	<none>	<none>	<none>
monitoring monitoring-prometheus-node-exporter-xrgn5	0/1	CrashLoopBackOff	30 (2n30s ago)	130m	192.168.1.150	nahesh	<none>	<none>	<none>
monitoring prometheus-monitoring-kube-prometheus-prometheus-0	2/2	Running	0	130m	192.168.5.169	nahesh	<none>	<none>	<none>

```
dell@mahesh:~$ kubectl get pods -n monitoring
NAME                               READY   STATUS    RESTARTS   AGE
alertmanager-monitoring-kube-prometheus-alertmanager-0   2/2    Running   0          138m
monitoring-grafana-65d9dd9c59-gm4tg   3/3    Running   0          131m
monitoring-kube-prometheus-operator-7984fbddd4-shqv   1/1    Running   0          131m
monitoring-kube-state-metrics-7d84c55c59-rcsgx   1/1    Running   14 (76m ago) 131m
monitoring-prometheus-node-exporter-xrgms   0/1    CrashLoopBackoff 30 (2m56s ago) 131m
prometheus-monitoring-kube-prometheus-prometheus-0   2/2    Running   0          130m
dell@mahesh:~$ kubectl get svc -A
NAMESPACE     NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
default       kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP         3h18m
default       nginx         NodePort    10.109.179.192 <none>        80:30665/TCP   3h5m
default       zonato-service   ClusterIP   10.119.192.255 <none>        3001:32001/TCP  3h14m
kube-system   kube-dns      ClusterIP   10.96.0.10      <none>        53/UDP,53/TCP,9153/TCP 3h18m
kube-system   monitoring-kube-prometheus-coredns   ClusterIP   None           <none>        9153/TCP        131m
kube-system   monitoring-kube-prometheus-kube-controller-manager   ClusterIP   None           <none>        10257/TCP       131m
kube-system   monitoring-kube-prometheus-kube-etcd      ClusterIP   None           <none>        2381/TCP        131m
kube-system   monitoring-kube-prometheus-kube-proxy     ClusterIP   None           <none>        10249/TCP       131m
kube-system   monitoring-kube-prometheus-kube-scheduler   ClusterIP   None           <none>        10259/TCP       131m
kube-system   monitoring-kube-prometheus-kubelet      ClusterIP   None           <none>        10250/TCP,10255/TCP,4194/TCP 144m
monitoring    alertmanager-operated   ClusterIP   None           <none>        9093/TCP,9094/TCP,9094/UDP 130m
monitoring    monitoring-grafana   ClusterIP   10.101.219.244 <none>        80/TCP          131m
monitoring    monitoring-kube-prometheus-alertmanager   ClusterIP   10.100.2.54      <none>        9093/TCP,8088/TCP  131m
monitoring    monitoring-kube-prometheus-operator     ClusterIP   10.98.203.37    <none>        443/TCP          131m
monitoring    monitoring-kube-prometheus-prometheus   ClusterIP   10.106.24.168    <none>        9090/TCP,8080/TCP  131m
monitoring    monitoring-kube-state-metrics   ClusterIP   10.106.68.46    <none>        8088/TCP        131m
monitoring    monitoring-prometheus-node-exporter   ClusterIP   10.108.207.12   <none>        5100/TCP        131m
monitoring    prometheus-operated      ClusterIP   None           <none>        9090/TCP        130m
dell@mahesh:~
```

```
dell@mahesh:~$ kubectl get deploy -A
NAMESPACE     NAME           READY   UP-TO-DATE   AVAILABLE   AGE
default       zonato-app     1/1    1            1            3h14m
kube-system   calico-kube-controllers   1/1    1            1            3h16m
kube-system   coredns        2/2    2            2            3h18m
monitoring    monitoring-grafana   1/1    1            1            131m
monitoring    monitoring-kube-prometheus-operator   1/1    1            1            131m
monitoring    monitoring-kube-state-metrics   1/1    1            1            131m
dell@mahesh:~$ kubectl get ds -A
NAMESPACE     NAME           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
kube-system   calico-node    1         1         1         1         1         kubernetes.io/os=linux 3h17m
kube-system   kube-proxy     1         1         1         1         1         kubernetes.io/os=linux 3h18m
monitoring    monitoring-prometheus-node-exporter 1         1         0         1         0         kubernetes.io/os=linux 131m
dell@mahesh:~
```

Thank You!

<https://github.com/MaheshBabu-DevOPs>

<https://linkedin.com/in/maheshbabu-devops>