

# CI/CD Pipeline for React App using Jenkins, Docker, SonarQube & Nexus

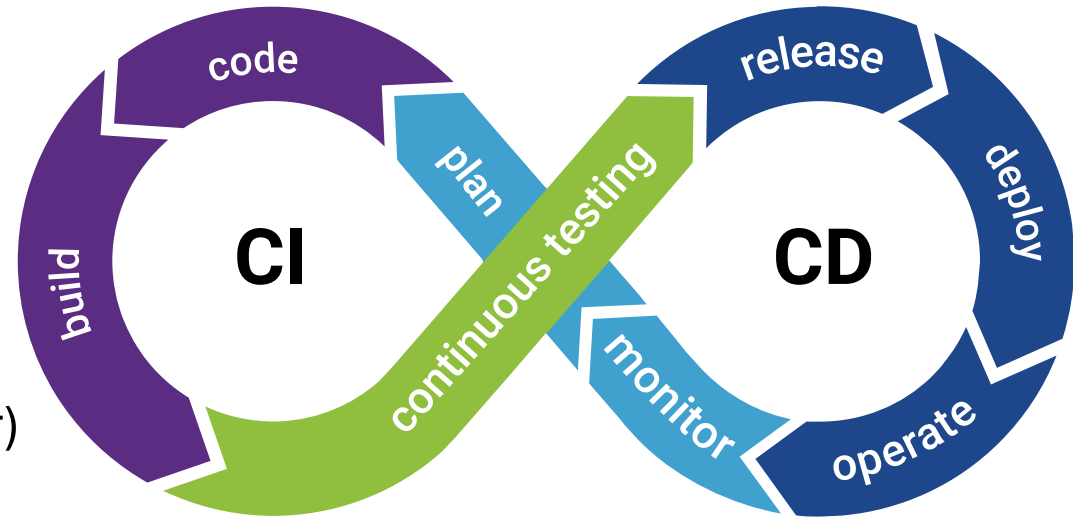
# Objective

- To automate the build, code quality check, Docker image creation, and deployment of a React app using Jenkins, with SonarQube for static analysis and Nexus for image hosting.
- The process includes tool setup and running a complete Jenkins pipeline.

# Set Up Environment

- Install Required Tools

- 🐝 Jenkins
- 🐝 Docker
- 🐝 Git
- 🐝 SonarQube (Container)
- 🐝 Nexus Repository Manager (Container)



# STEP 1 : Install Required Tools

- Install Jenkins

## Add Jenkins repo

🐛 sudo wget -O /etc/yum.repos.d/jenkins.repo \  
https://pkg.jenkins.io/redhat-stable/jenkins.repo

🐛 sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

## Add required dependencies for the jenkins package

🐛 sudo yum install fontconfig java-21-openjdk

🐛 sudo yum install jenkins

🐛 sudo systemctl daemon-reload

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for satthya:
--2025-08-07 10:26:58-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.46.133, 2a04:4e42:7b::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.46.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'
```

```
/etc/yum.repos.d/jenkins.repo 100%[=====>] 85 --.-KB/s in 0s
```

```
2025-08-07 10:27:03 (1.32 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]
```

```
[satthya@test-env ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[satthya@test-env ~]$ sudo yum install fontconfig java-21-openjdk -y
Updating Subscription Management repositories.
Jenkins-stable
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
Package fontconfig-2.14.0-2.el9_1.x86_64 is already installed.
Dependencies resolved.
```

1.2 kB/s		32 kB	00:25
3.2 MB/s		65 MB	00:20
3.1 MB/s		70 MB	00:22

Package	Architecture	Version	Repository	Size
Installing:				
java-21-openjdk	x86_64	1:21.0.8.0.9-1.el9	rhel-9-for-x86_64-appstream-rpms	426 k
Installing dependencies:				
ModemManager-glib	x86_64	1.20.2-1.el9	rhel-9-for-x86_64-baseos-rpms	337 k
adwaita-cursor-theme	noarch	40.1.1-3.el9	rhel-9-for-x86_64-appstream-rpms	655 k
adwaita-icon-theme	noarch	40.1.1-3.el9	rhel-9-for-x86_64-appstream-rpms	12 M
alsa-lib	x86_64	1.2.13-2.el9	rhel-9-for-x86_64-appstream-rpms	530 k
at-spi2-atk	x86_64	2.38.0-4.el9	rhel-9-for-x86_64-appstream-rpms	90 k
at-spi2-core	x86_64	2.40.3-1.el9	rhel-9-for-x86_64-appstream-rpms	199 k
atk	x86_64	2.36.0-5.el9	rhel-9-for-x86_64-appstream-rpms	296 k
avahi-glib	x86_64	0.8-20.el9	rhel-9-for-x86_64-appstream-rpms	16 k
bluez-libs	x86_64	5.72-4.el9	rhel-9-for-x86_64-baseos-rpms	84 k
bubblewrap	x86_64	0.4.1-8.el9_5	rhel-9-for-x86_64-baseos-rpms	52 k
colord-libs	x86_64	1.4.5-6.el9_6	rhel-9-for-x86_64-appstream-rpms	233 k
composefs-libs	x86_64	1.15.0-1.el9	rhel-9-for-x86_64-appstream-rpms	57 k
copy-jdk-configs	noarch	4.1-1.el9	rhel-9-for-x86_64-appstream-rpms	29 k
cups-libs	x86_64	1.2.2-40.el9_5	rhel-9-for-x86_64-baseos-rpms	265 k
exempi	x86_64	2.6.0-0.2.20211007gite23c213.el9	rhel-9-for-x86_64-appstream-rpms	528 k
exiv2-libs	x86_64	0.27.5-2.el9	rhel-9-for-x86_64-appstream-rpms	782 k
fdk-aac-free	x86_64	2.0.0-8.el9	rhel-9-for-x86_64-appstream-rpms	327 k
flac-libs	x86_64	1.3.3-10.el9_2.1	rhel-9-for-x86_64-appstream-rpms	221 k
flatpak-selinux	noarch	1.12.9-3.el9_4	rhel-9-for-x86_64-appstream-rpms	22 k
flatpak-session-helper	x86_64	1.12.9-3.el9_4	rhel-9-for-x86_64-appstream-rpms	75 k
fuse	x86_64	2.9.9-17.el9	rhel-9-for-x86_64-baseos-rpms	84 k

# Jenkins Installation

# STEP 1 : Install Required Tools

- Install Docker

## Uninstall old versions

```
🐛 sudo dnf remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-engine \
    podman \
    runc
```

satthya@test-env:~

×

+

▼

[satthya@test-env ~]\$ sudo dnf remove docker \

docker-client \

docker-client-latest \

docker-common \

docker-latest \

docker-latest-logrotate \

docker-logrotate \

docker-engine \

podman \

runc

Updating Subscription Management repositories.

No match for argument: docker

No match for argument: docker-client

No match for argument: docker-client-latest

No match for argument: docker-common

No match for argument: docker-latest

No match for argument: docker-latest-logrotate

No match for argument: docker-logrotate

No match for argument: docker-engine

No match for argument: runc

Dependencies resolved.

Package	Architecture	Version	Repository	Size
Removing:				
podman	x86_64	4:4.9.4-16.el9_4	@rhel-9-for-x86_64-appstream-rpms	53 M
Removing dependent packages:				
cockpit-podman	noarch	84.1-1.el9	@rhel-9-for-x86_64-appstream-rpms	682 k
Removing unused dependencies:				
common	x86_64	2:2.1.10-1.el9	@rhel-9-for-x86_64-appstream-rpms	170 k

Transaction Summary

Remove 3 Packages

Freed space: 54 M

Is this ok [y/N]: y

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

Preparing :

Erasing : cockpit-podman-84.1-1.el9.noarch

Running scriptlet: podman-4:4.9.4-16.el9\_4.x86\_64

Erasing : podman-4:4.9.4-16.el9\_4.x86\_64

Erasing : common-2:2.1.10-1.el9.x86\_64

Running scriptlet: common-2:2.1.10-1.el9.x86\_64

Verifying : cockpit-podman-84.1-1.el9.noarch

Verifying : common-2:2.1.10-1.el9.x86\_64

Verifying : podman-4:4.9.4-16.el9\_4.x86\_64

Installed products updated.

Removed:

cockpit-podman-84.1-1.el9.noarch

common-2:2.1.10-1.el9.x86\_64

podman-4:4.9.4-16.el9\_4.x86\_64

Complete!

Uninstall Old Docker Version

# STEP 1 : Install Required Tools

- Install Docker

## Add Docker repo

🐝 sudo dnf -y install dnf-plugins-core

🐝 sudo dnf config-manager --add-repo <https://download.docker.com/linux/rhel/docker-ce.repo>

## Install Docker

🐝 sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin



```
satthya@test-env:~  
[satthya@test-env ~]$ sudo dnf -y install dnf-plugins-core  
[sudo] password for satthya:  
Updating Subscription Management repositories.  
Last metadata expiration check: 0:26:54 ago on Thu 07 Aug 2025 10:53:21 AM +08.  
Package dnf-plugins-core-4.3.0-13.el9.noarch is already installed.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository
---------	--------------	---------	------------

```
=====
```

Upgrading:

dnf-plugins-core	noarch	4.3.0-20.el9	rhel-9-for-x86_64-baseos-rpms
python3-dnf-plugins-core	noarch	4.3.0-20.el9	rhel-9-for-x86_64-baseos-rpms

Transaction Summary

```
=====
```

Upgrade 2 Packages

Total download size: 310 k  
Downloading Packages:  
(1/2): dnf-plugins-core-4.3.0-20.el9.noarch.rpm 7.4 kB/s | 41 kB 00:00  
(2/2): python3-dnf-plugins-core-4.3.0-20.el9.noarch.rpm 27 kB/s | 268 kB 00:10  
-----

Total	30 kB/s   310 kB 00:10
-------	------------------------

```
Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.  
Running transaction  
Preparing :  
Upgrading : python3-dnf-plugins-core-4.3.0-20.el9.noarch  
Upgrading : dnf-plugins-core-4.3.0-20.el9.noarch  
Cleanup : dnf-plugins-core-4.3.0-13.el9.noarch  
Cleanup : python3-dnf-plugins-core-4.3.0-13.el9.noarch  
Running scriptlet: python3-dnf-plugins-core-4.3.0-13.el9.noarch  
Verifying : dnf-plugins-core-4.3.0-20.el9.noarch  
Verifying : dnf-plugins-core-4.3.0-13.el9.noarch  
Verifying : python3-dnf-plugins-core-4.3.0-20.el9.noarch  
Verifying : python3-dnf-plugins-core-4.3.0-13.el9.noarch  
Installed products updated.  
  
Upgraded:  
dnf-plugins-core-4.3.0-20.el9.noarch python3-dnf-plugins-core-4.3.0-20.el9.noarch  
  
Complete!  
[satthya@test-env ~]$ sudo dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo  
Updating Subscription Management repositories.  
Adding repo from: https://download.docker.com/linux/rhel/docker-ce.repo  
[satthya@test-env ~]$ sudo dnf install docker-ce docker-ce-cli containerd.io dock  
Updating Subscription Management repositories.  
Docker CE Stable - x86_64 2.1 kB/s | 42 kB 00:20  
Last metadata expiration check: 0:00:10 ago on Thu 07 Aug 2025 11:21:52 AM +08.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository
---------	--------------	---------	------------

```
=====
```

Installing:

containerd.io	x86_64	1.7.27-3.1.el9	docker-ce-stable
---------------	--------	----------------	------------------

```
=====
```

# Install Docker

# STEP 1 : Install Required Tools

- Enable Docker and Jenkin service

## Enable and start Docker service

 `sudo systemctl enable docker.service`

 `sudo systemctl start docker.service`

## Enable and start Jenkins service

 `sudo systemctl enable Jenkins.service`


 `sudo systemctl start Jenkins.service`

```
sathya@test-env:~  
[sathya@test-env ~]$ sudo systemctl enable docker.service  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.  
[sathya@test-env ~]$ sudo systemctl start docker.service  
[sathya@test-env ~]$ sudo systemctl enable jenkins.service  
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.  
[sathya@test-env ~]$ sudo systemctl start jenkins.service  
[sathya@test-env ~]$ sudo systemctl status jenkins.service  
● jenkins.service - Jenkins Continuous Integration Server  
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)  
   Active: active (running) since Thu 2025-08-07 11:28:13 +08; 1min 19s ago  
 Main PID: 32424 (java)  
    Tasks: 37 (limit: 23145)  
  Memory: 392.9M  
     CPU: 32.101s  
   CGroup: /system.slice/jenkins.service  
           └─32424 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080  
  
Aug 07 11:28:01 test-env jenkins[32424]: c0030563f51b4820b54133b04ad6990c  
Aug 07 11:28:01 test-env jenkins[32424]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword  
Aug 07 11:28:01 test-env jenkins[32424]: *****  
Aug 07 11:28:01 test-env jenkins[32424]: *****  
Aug 07 11:28:01 test-env jenkins[32424]: *****  
Aug 07 11:28:13 test-env jenkins[32424]: 2025-08-07 03:28:13.917+0000 [id=39] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization  
Aug 07 11:28:13 test-env jenkins[32424]: 2025-08-07 03:28:13.954+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running  
Aug 07 11:28:13 test-env systemd[1]: Started Jenkins Continuous Integration Server.  
Aug 07 11:28:16 test-env jenkins[32424]: 2025-08-07 03:28:16.042+0000 [id=55] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.Mave  
Aug 07 11:28:16 test-env jenkins[32424]: 2025-08-07 03:28:16.043+0000 [id=55] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt  
[sathya@test-env ~]$ sudo systemctl status docker.service  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)  
   Active: active (running) since Thu 2025-08-07 11:26:18 +08; 3min 26s ago  
TriggeredBy: ● docker.socket  
    Docs: https://docs.docker.com  
 Main PID: 32149 (dockerd)  
    Tasks: 11  
  Memory: 59.3M  
     CPU: 674ms  
   CGroup: /system.slice/docker.service  
           └─32149 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
  
Aug 07 11:26:13 test-env dockerd[32149]: time="2025-08-07T11:26:13.640596925+08:00" level=info msg="Creating a containerd client" address=/run/containerd/containerd.sock timeout=1m0s  
Aug 07 11:26:13 test-env dockerd[32149]: time="2025-08-07T11:26:13.816548286+08:00" level=info msg="Loading containers: start."  
Aug 07 11:26:14 test-env dockerd[32149]: time="2025-08-07T11:26:14.710426390+08:00" level=info msg="Firewall: created docker-forwarding policy"  
Aug 07 11:26:18 test-env dockerd[32149]: time="2025-08-07T11:26:18.537395974+08:00" level=info msg="Loading containers: done."  
Aug 07 11:26:18 test-env dockerd[32149]: time="2025-08-07T11:26:18.893548437+08:00" level=info msg="Daemon has completed initialization"  
Aug 07 11:26:18 test-env dockerd[32149]: time="2025-08-07T11:26:18.894141401+08:00" level=info msg="API listen on /run/docker.sock"  
Aug 07 11:26:18 test-env systemd[1]: Started Docker Application Container Engine.  
[sathya@test-env ~]$ |
```

# Enable and started required services

# STEP 1 : Install Required Tools

- Add Jenkins user to docker group

 `sudo usermod -aG docker Jenkins`

- Allow firewall rule

 `firewall-cmd --add-port=8080/tcp --permanent`

 `firewall-cmd -- reload`

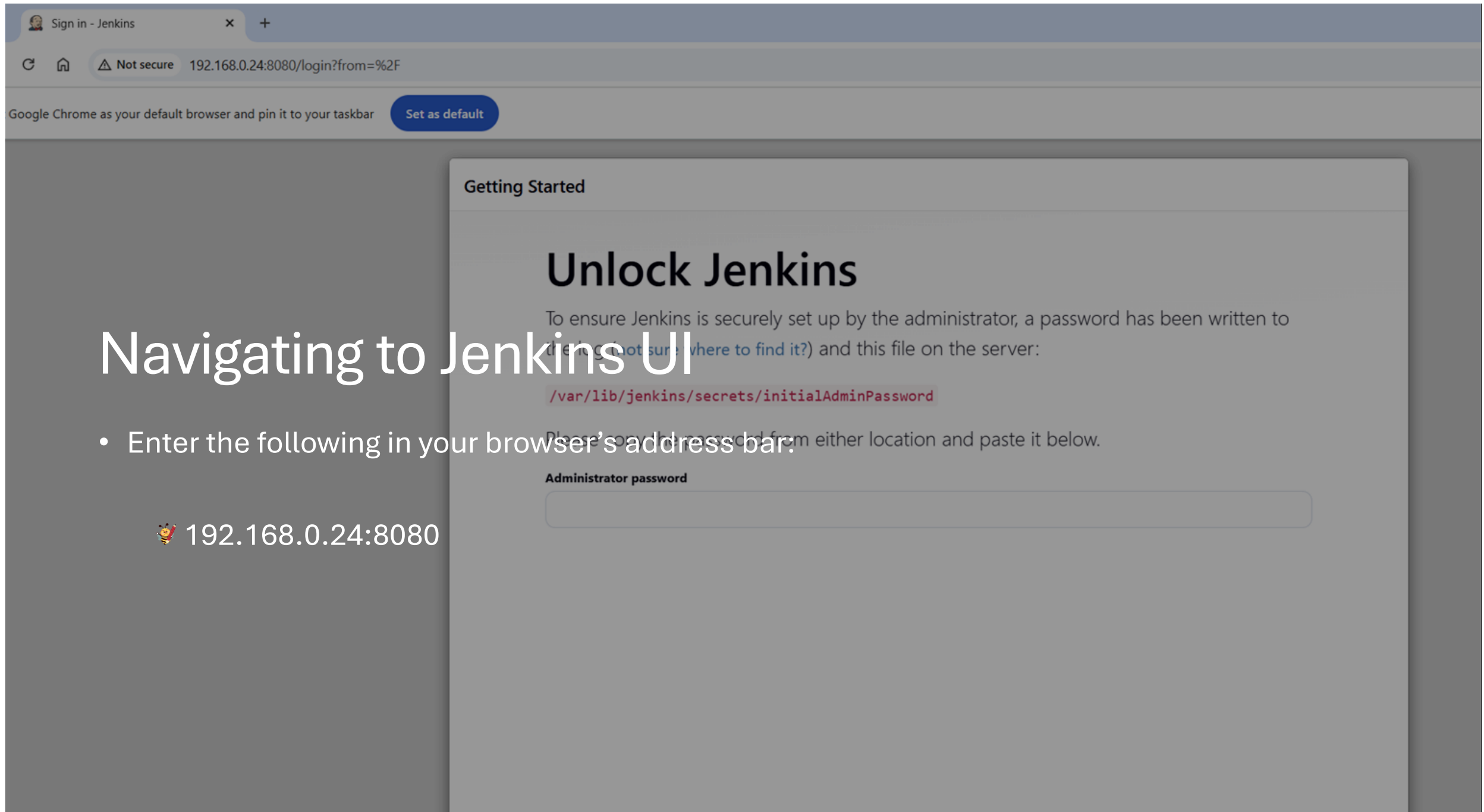
```
satthya@test-env:~  
[satthya@test-env ~]$ sudo usermod -aG docker jenkins  
[sudo] password for satthya:  
[satthya@test-env ~]$ sudo firewall-cmd --add-port=8080/tcp --permanent  
success  
[satthya@test-env ~]$ sudo firewall-cmd --reload  
success  
[satthya@test-env ~]$ sudo firewall-cmd --list-all  
public (active)  
  target: default  
  icmp-block-inversion: no  
  interfaces: eth0  
  sources:  
  services: cockpit dhcpv6-client ssh  
  ports: 8080/tcp  
  protocols:  
  forward: yes  
  masquerade: no  
  forward-ports:  
  source-ports:  
  icmp-blocks:  
  rich rules:  
[satthya@test-env ~]$ |
```

## Allow Firewall Rules

# Navigating to Jenkins UI

- Enter the following in your browser's address bar:

🐝 192.168.0.24:8080



## STEP 2 : Jenkins Configuration

Configure Jenkins with required plugins and global tools.

🐝 Installed Jenkins plugins:

- Docker Pipeline
- Git Plugin
- SonarQube Scanner
- Nexus Artifact

🐝 Configure global tools under Manage Jenkins > Global Tool Configuration:

- SonarQube Scanner
- Verified Git and Docker path availability

Available plugins - Plugins - Jen x +

Not secure 192.168.0.24:8080/manage/pluginManager/available

Set Google Chrome as your default browser and pin it to your taskbar Set as default

Jenkins / Manage Jenkins / Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Search Nexus

Install

Install	Name ↓	Released	Health
<input checked="" type="checkbox"/>	<b>Docker Pipeline</b> 621.va_73f881d9232 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	2 mo 8 days ago	97
<input checked="" type="checkbox"/>	<b>SonarQube Scanner</b> 2.18 External Site/Tool Integrations Build Reports This plugin allows an easy integration of <a href="#">SonarQube</a> , the open source platform for Continuous Inspection of code quality.	6 mo 10 days ago	88
<input checked="" type="checkbox"/>	<b>Nexus Artifact Uploader</b> 2.14 Artifact Uploaders This plugin to upload the artifact to Nexus Repository. <div>This plugin is up for adoption! We are looking for new maintainers. Visit our <a href="#">Adopt a Plugin</a> initiative for more information.</div>	2 yr 8 mo ago	77

# Install Plugins



Download progress - Plugins - x

Not secure 192.168.0.24:8080/manage/pluginManager/updates/

Set Google Chrome as your default browser and pin it to your taskbar Set as default

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

Authentication Tokens API	✓ Success
Docker Commons	✓ Success
Docker Pipeline	✓ Success
SonarQube Scanner	✓ Success
JavaMail API	✓ Success
Commons HttpClient 3.x API	⋮ Pending
Nexus Artifact Uploader	⋮ Pending
Loading plugin extensions	⋮ Pending

→ [Go back to the top page](#)  
(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

# Install Plugins

Tools - Jenkins

Not secure 192.168.0.24:8080/manage/configureTools/

Set Google Chrome as your default browser and pin it to your taskbar [Set as default](#)

Jenkins / Manage Jenkins / Tools

Add SonarQube Scanner

**SonarQube Scanner**

Name

SonarScanner

☒ Install automatically ?

**Install from Maven Central**

Version

SonarQube Scanner 7.2.0.5079

Add Installer

# Install Plugins

## STEP 3 : SonarQube Setup

Enable static code analysis for React app.

🐝 Started SonarQube using Docker

- `sudo docker run -d --name sonarqube -p 9000:9000 sonarqube:lts`

🐝 Allow firewall rule

- `sudo firewall-cmd --add-port=9000/tcp --permanent`
- `sudo firewall-cmd --reload`

```

satthya@test-env:~
[satthya@test-env ~]$ sudo docker run -d --name sonarqube -p 9000:9000 sonarqube:lts
[sudo] password for satthya:
Unable to find image 'sonarqube:lts' locally
lts: Pulling from library/sonarqube
1d387567261e: Pull complete
36683a82f98b: Pull complete
f406bdeb01f2: Pull complete
0a6cdfe79704: Pull complete
5572597b7c70: Pull complete
faf2bc8c0730: Pull complete
8b4d56167ca1: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:4ceaa69102bab19d8db48f2d6717ed35e14fc0fce998670830569a4e0e8fc089
Status: Downloaded newer image for sonarqube:lts
abf47b0f07d0dfe602172376d5058a32499a4b18b761cb2428a7025384f0fde
[satthya@test-env ~]$ sudo docker ps

```

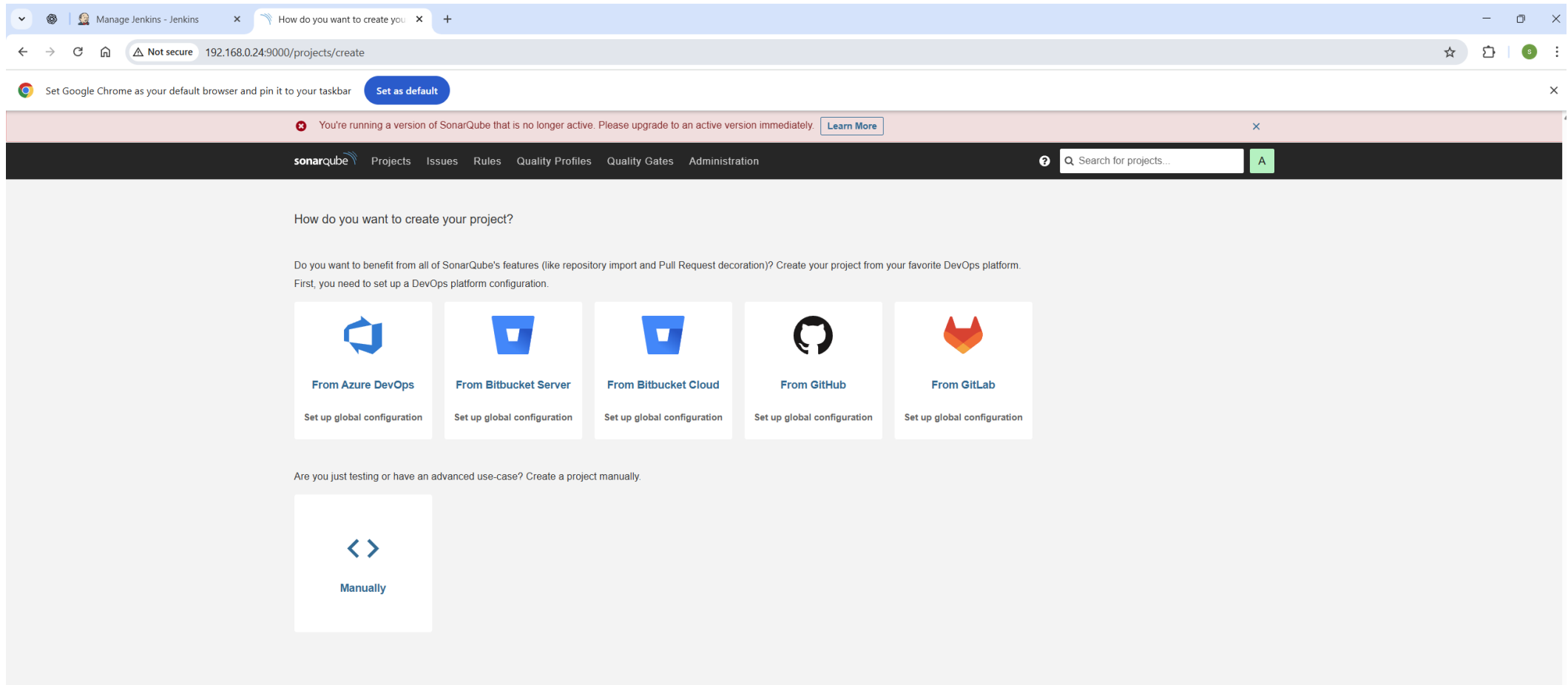
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
abf47b0f07d0	sonarqube:lts	"/opt/sonarqube/dock..."	2 minutes ago	Up 2 minutes	0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp	sonarqube

```

[satthya@test-env ~]$

```

## Start SonarQube using Docker



# SonarQube UI

## STEP 3 : SonarQube Setup

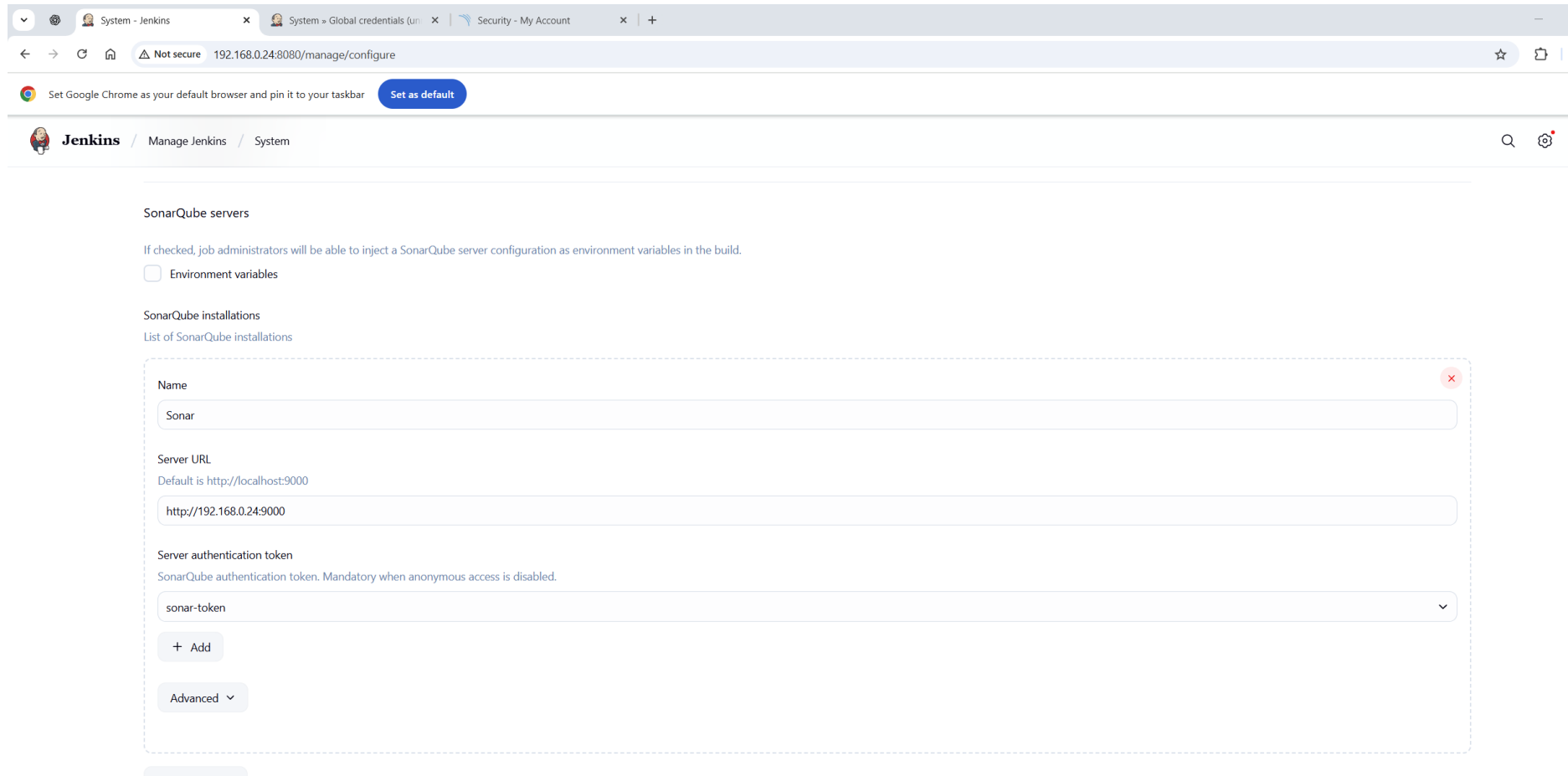
Enable static code analysis for React app.

🐝 Create token via My Account > Security

🐝 Configure Sonar server in Jenkins (configure system)

- Name: Sonar
- URL: <http://192.168.0.24:9000>
- Token added as secret text credential

# Configure Sonar Server in Jenkins



The screenshot shows the Jenkins web interface in a Google Chrome browser. The address bar indicates the URL is `192.168.0.24:8080/manage/configure`. The page title is 'Jenkins / Manage Jenkins / System'. The main content area is titled 'SonarQube servers'. It contains a checkbox for 'Environment variables' which is unchecked. Below this is the 'SonarQube installations' section, which includes a link 'List of SonarQube installations'. A dashed box highlights a configuration entry with the following fields: 'Name' (value: 'Sonar'), 'Server URL' (value: 'http://192.168.0.24:9000'), and 'Server authentication token' (value: 'sonar-token'). There is a '+ Add' button and an 'Advanced' dropdown menu at the bottom of the dashed box.

System - Jenkins x System - Global credentials (un x Security - My Account x +

← → ↺ 🏠 ⚠ Not secure 192.168.0.24:8080/manage/configure ☆ 📦 |

Set Google Chrome as your default browser and pin it to your taskbar Set as default

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 URL

Default is `http://localhost:9000`

`http://192.168.0.24:9000`

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar-token ▼

+ Add

Advanced ▼

# STEP 4 : Nexus Repository Setup

🐝 Start Nexus via Docker

- `sudo docker run -d -p 8081:8081 -p 5000:5000 --name nexus sonatype/nexus3`

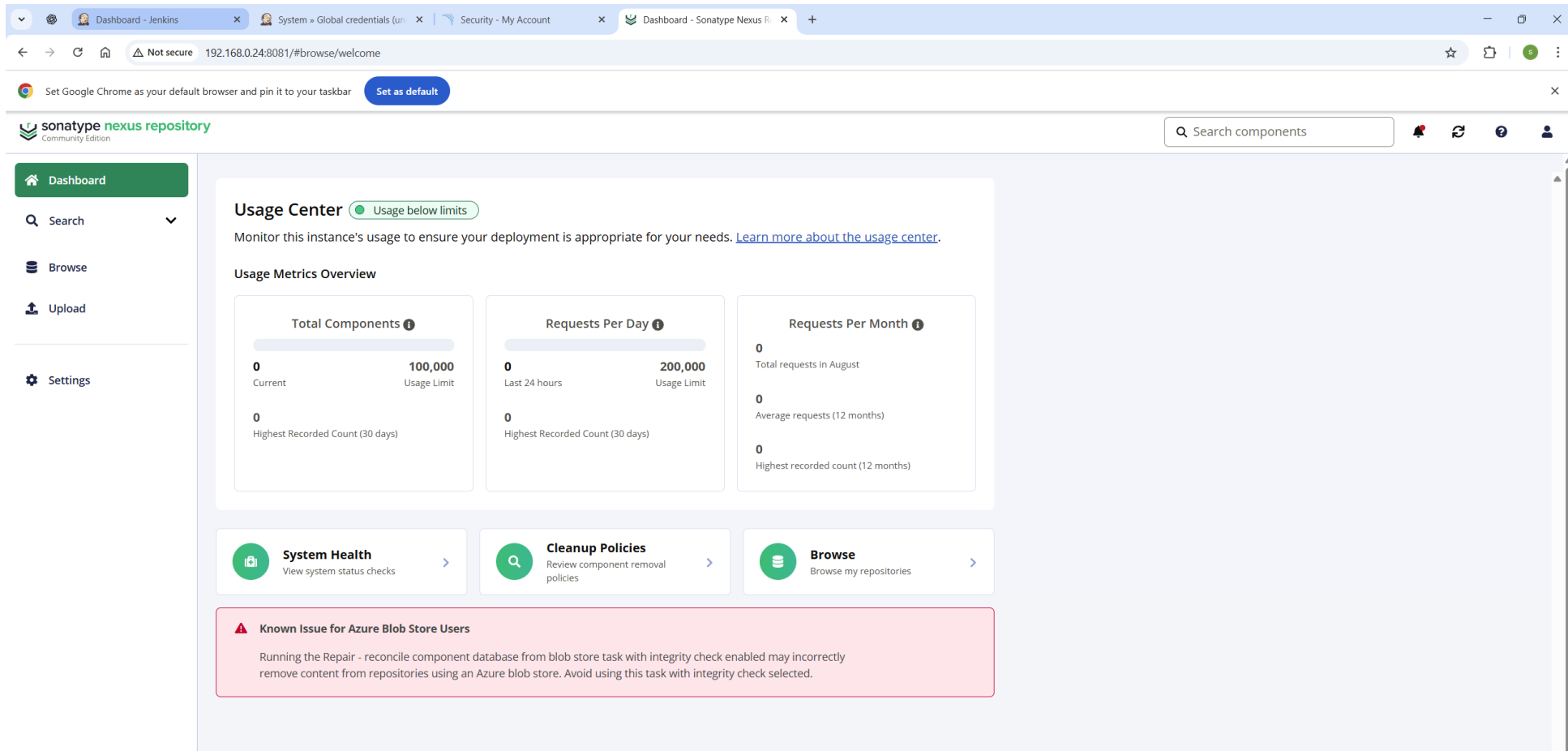
🐝 Allow firewall rule

- `sudo firewall-cmd --add-port=5000/tcp --permanent`
- `sudo firewall-cmd --add-port=8081/tcp --permanent`
- `sudo firewall-cmd --reload`



```
satthya@test-env:~  
[satthya@test-env ~]$ sudo docker run -d -p 8081:8081 -p 5000:5000 --name nexus sonatype/nexus3  
Unable to find image 'sonatype/nexus3:latest' locally  
latest: Pulling from sonatype/nexus3  
1ec5864c3611: Pull complete  
07fae006c362: Pull complete  
80810cc180ff: Pull complete  
4ef57ae1dd31: Pull complete  
e8a81a779a26: Pull complete  
2cdacfdddb56: Pull complete  
Digest: sha256:e2bfd3bc42743823bdf389be20d78703609a5f8d0e875788d2f55aa59beef85b  
Status: Downloaded newer image for sonatype/nexus3:latest  
9cb5dee88b921d495902e2873363272d9400a4a3e9457d751d8b94d8886cc191  
[satthya@test-env ~]$ firewall-cmd --add-port=8081/tcp --permanent  
Authorization failed.  
Make sure polkit agent is running or run the application as superuser.  
[satthya@test-env ~]$ sudo firewall-cmd --add-port=8081/tcp --permanent  
success  
[satthya@test-env ~]$ sudo firewall-cmd --add-port=5000/tcp --permanent  
success  
[satthya@test-env ~]$ sudo firewall-cmd --reload  
success  
[satthya@test-env ~]$ sudo docker ps  
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                                     NAMES  
9cb5dee88b92   sonatype/nexus3      "/opt/sonatype/nexus..." 2 minutes ago  Up 2 minutes  0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp, 0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp  nexus  
abf47b0f07d0   sonarqube:lts        "/opt/sonarqube/dock..." About an hour ago  Up About an hour  0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp  sonarqube  
[satthya@test-env ~]$
```

# Start Nexus via Docker and Allow Firewall rule



# Sonatype Nexus UI

# STEP 4 : Nexus Repository Setup

🐛 Create Docker (hosted repository)

- Name: docker-hosted
- Port: 5000
- Enabled push access

# Sonatype Nexus UI

The screenshot displays the Sonatype Nexus Community Edition interface. The top navigation bar includes the Sonatype logo, a search bar for components, and user icons. The left sidebar contains a 'Settings' menu with options like Repository, Repositories, Blob Stores, Data Store, Proprietary Repositories, Content Selectors, Cleanup Policies, Routing Rules, Security, Support, System, and IQ Server. The main content area is titled 'Repositories' and shows the configuration for a 'docker-hosted' repository. It includes buttons for 'Delete repository' and 'Rebuild index', a 'Settings' tab, and various configuration sections: Name, Format, Type, URL, Online status, Repository Connectors (with HTTP and HTTPS settings), Docker Registry API Support (with V1 API enablement), Storage (with Blob store), and Strict Content Type Validation.

sonatype nexus repository  
Community Edition

Search components

Repositories / docker-hosted

Delete repository Rebuild index

Settings

**Name:** docker-hosted  
**Format:** docker  
**Type:** hosted  
**URL:** http://192.168.0.24:8081/repository/docker-hosted/  
**Online:** ☒ If checked, the repository accepts incoming requests

**Repository Connectors**

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our [documentation](#) for which connector is appropriate for your use case. For information on scaling the repositories see our [scaling documentation](#).

**HTTP:**  
Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.  
☒ 5000

**HTTPS:**  
Create an HTTPS connector at specified port. Normally used if the server is configured for https.  
☐

**Allow anonymous docker pull:**  
☐ Allow anonymous docker pull ( Docker Bearer Token Realm required )

**Docker Registry API Support**

**Enable Docker V1 API:**  
☐ Allow clients to use the V1 API to interact with this repository

**Storage**

**Blob store:**  
default


**Strict Content Type Validation:**




# STEP 5: Jenkins Pipeline Setup

🐝 Automate build, scan, push, and deploy process


- Create and trigger CI/CD job.
  - Create new pipeline job in Jenkins
  - Configure:
    - GitHub repo URL
    - Script path: Jenkinsfile / Groovy script
    - Trigger: Poll SCM or GitHub webhook
    - Git credentials if needed


# Create new pipeline job


 **Jenkins** / satthya-project / Configuration


  

### Configure


 General

 Triggers

 Pipeline

 Advanced

### General

Enabled 

Description

CI/CD Pipeline for React App using Jenkins, Docker, SonarQube & Nexus

Plain text [Preview](#)

☐ Discard old builds ?


☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ GitHub project

Project url ?

https://github.com/satthya/jenkin-project2.git

Advanced 

☐ Pipeline speed/durability override ?

☐ Preserve stashes from completed builds ?

☐ This project is parameterized ?


☐ Throttle builds ?



Triggers

Save



Apply

# Create new pipeline job

 **Jenkins** / satthya-project / Configuration

## Configure

-  General
-  **Triggers**
-  Pipeline
-  Advanced

### Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☒ **Poll SCM ?**

Schedule ?

\*/\* 5 \* \* \*

No schedules so will only run due to SCM changes if triggered by a post-commit hook

- ☐ Ignore post-commit hooks ?
- ☐ Trigger builds remotely (e.g., from scripts) ?

# Create new pipeline job



Jenkins

/ satthya-project

/ Configuration

## Configure



General



Triggers



Pipeline



Advanced

### Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

#### Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   environment {
5     sonar_token = credentials('sonar-token')
6   }
7
8   stages {
9     stage('Git Checkout') {
10      steps {
11        git branch: 'main', credentialsId: 'Git-Jenkins', url: 'https://github.com/satthya/Jenkin-project2.git'
12      }
13    }
14
15    stage('SonarQube Scan') {
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

### Advanced

Advanced ▾

Save

Apply



# STEP 5: Jenkins Pipeline Setup

## 🐛 Pipeline script (Groovy Language)

```
pipeline {
  agent any

  environment {
    sonar_token = credentials('sonar-token')
    docker_image = 'satthya-reactapp'
  }

  stages {
    stage('Git Checkout') {
      steps {
        git branch: 'main', credentialsId: 'Git-Jenkins', url: 'https://github.com/satthya/Jenkin-project2.git'
      }
    }

    stage('Sonarqube Scan') {
      steps {
        script {
          withSonarQubeEnv('SonarScanner') {
            sh 'sonar-scanner -Dsonar.projectKey=satthya-ReactApp -Dsonar.sources=. -Dsonar.host.url=http://192.168.0.25:9000 -Dsonar.login=$sonar_token'
          }
        }
      }
    }

    stage('Docker Build') {
      steps {
        sh 'docker build -t $docker_image .'
      }
    }

    stage('Nexus Sonatype') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'Nexus-Jenkin', usernameVariable: 'NEXUS_USER', passwordVariable: 'NEXUS_PASS')]) {
          sh 'docker tag satthya-reactapp:latest 192.168.0.25:5000/satthya-reactapp:latest'
          sh 'docker login -u $NEXUS_USER -p $NEXUS_PASS 192.168.0.25:5000'
          sh 'docker push 192.168.0.25:5000/$docker_image'
        }
      }
    }

    stage('Application Deployment') {
      steps {
        sh 'docker run -dit --name satthya-reactapp -p 80:80 $docker_image'
      }
    }
  }
}
```

# Final Jenkin Stage View

 **Jenkins** / satthya-project

Status

</> Changes

▶ Build Now

⚙️ Configure

🗑️ Delete Pipeline

🔍 Full Stage View

🐙 GitHub

📶 SonarQube

📁 Stages

✎ Rename

❓ Pipeline Syntax

📄 Git Polling Log

Builds

🔍 Filter

Today

✅ #57 5:23 PM

❌ #56 5:21 PM

🟢 **satthya-project**

CI/CD Pipeline for React App using Jenkins, Docker, SonarQube & Nexus

Stage View

Average stage times:  
(full run time: ~1min 7s)

	Git Checkout	Sonarqube Scan	Docker Build	Nexus Sonatype	Application Deployment
#57 Aug 09 17:23 No Changes	16s	27s	3s	2s	1s
#56 Aug 09 17:21 No Changes					
#55 Aug 09 17:06 No Changes	16s	24s	5s	32s	
#54 Aug 09 16:54 No Changes	16s	57s	9s	1s failed	
#53 Aug 09 14:08 No Changes	12s	38s	6s	1s failed	

🔍

⚙️

👤

Edit description

# Final SonarQube UI View

SonarQube

community

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

My FavoritesAll

Filters

Quality Gate

Passed

1

Failed

0

Security

A ≥ 0 info issues

1

B ≥ 1 low issue

0

C ≥ 1 medium issue

0

D ≥ 1 high issue

0

Search projects (minimum 2 characters)

PerspectiveOverall StatusSort byName

Create Project

1 project(s)

☆ sathya-ReactAppPublic

Passed

Last analysis: 14 minutes ago • 7 Lines of Code • Docker, HTML

A 0

A 0

A 1

E 0.0%

—

0.0%

Security

Reliability

Maintainability

Hotspots Reviewed

Coverage

Duplications

1 of 1 shown

# Final Nexus Sonatype UI view

 Dashboard

 Search

 Browse




 Upload

 Settings

 **Browse** /  docker-hosted

[HTML View](#)

[Advanced search..](#)

 v2  
   blobs  
   satthya-reactapp

# Final React App interface

