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

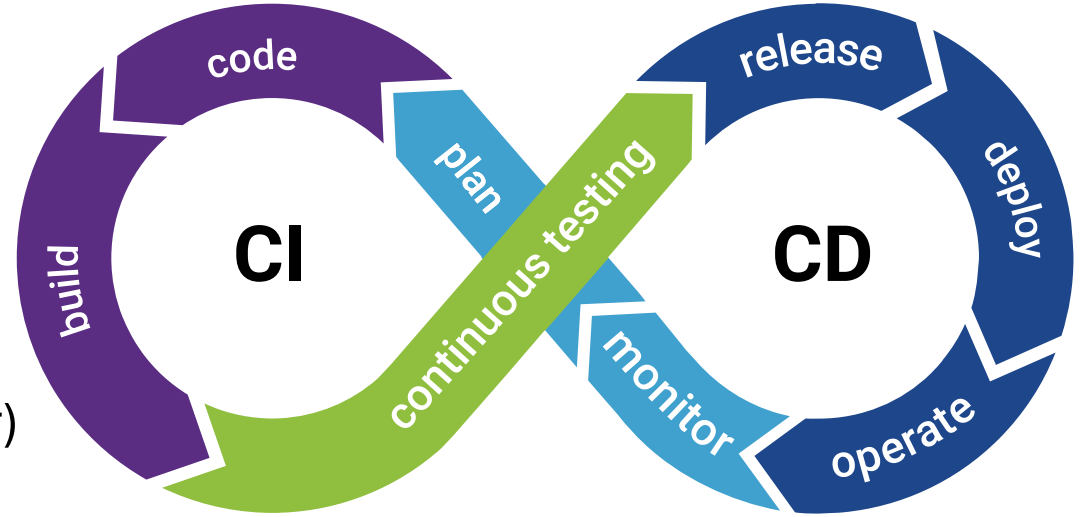
Objective

- 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                                                                  1.2 kB/s |  32 kB     00:25
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)                        3.2 MB/s |  65 MB     00:20
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)                           3.1 MB/s |  70 MB     00:22
Package fontconfig-2.14.0-2.el9_1.x86_64 is already installed.
Dependencies resolved.
================================================================================================================================================
 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          1.0.2-1.el9                             rhel-9-for-x86_64-appstream-rpms        57 k
 copy-jdk-configs                           noarch          4.0-2.el9                               rhel-9-for-x86_64-appstream-rpms        29 k
 cups-libs                                  x86_64          1:2.3.3op2-24.el9                       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

<u>Uninstall old versions</u>

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:
 conmon                  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          : conmon-2:2.1.10-1.el9.x86_64
  Running scriptlet: conmon-2:2.1.10-1.el9.x86_64
  Verifying        : cockpit-podman-84.1-1.el9.noarch
  Verifying        : conmon-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          conmon-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  00:0
(2/2): python3-dnf-plugins-core-4.3.0-20.el9.noarch.rpm   27 kB/s | 268 kB  00:1
--------------------------------------------------------------------------------
Total                                                     30 kB/s | 310 kB  00:1
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  00:2
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

```
[satthya@test-env ~]$ sudo systemctl enable docker.service
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[satthya@test-env ~]$ sudo systemctl start docker.service
[satthya@test-env ~]$ sudo systemctl enable jenkins.service
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[satthya@test-env ~]$ sudo systemctl start jenkins.service
[satthya@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
[satthya@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="2025-08-07T11:26:13.816548386+08:00" level=info msg="Loading containers: start."
Aug 07 11:26:14 test-env dockerd[32149]: time="2025-08-07T11:26:14.910428396+08:00" level=info msg="Firewalld: 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.[illegible]" level=info msg="[illegible]" bridge=false storage-driver=overlay2 version=28.
Aug 07 11:26:18 test-env dockerd[32149]: time="2025-08-07T11:26:18.[illegible]" level=info msg="[illegible]"
Aug 07 11:26:18 test-env dockerd[32149]: time="2025-08-07T11:26:18.[illegible]" level=info msg="[illegible]"
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.
[satthya@test-env ~]$
```

Enable and started required services

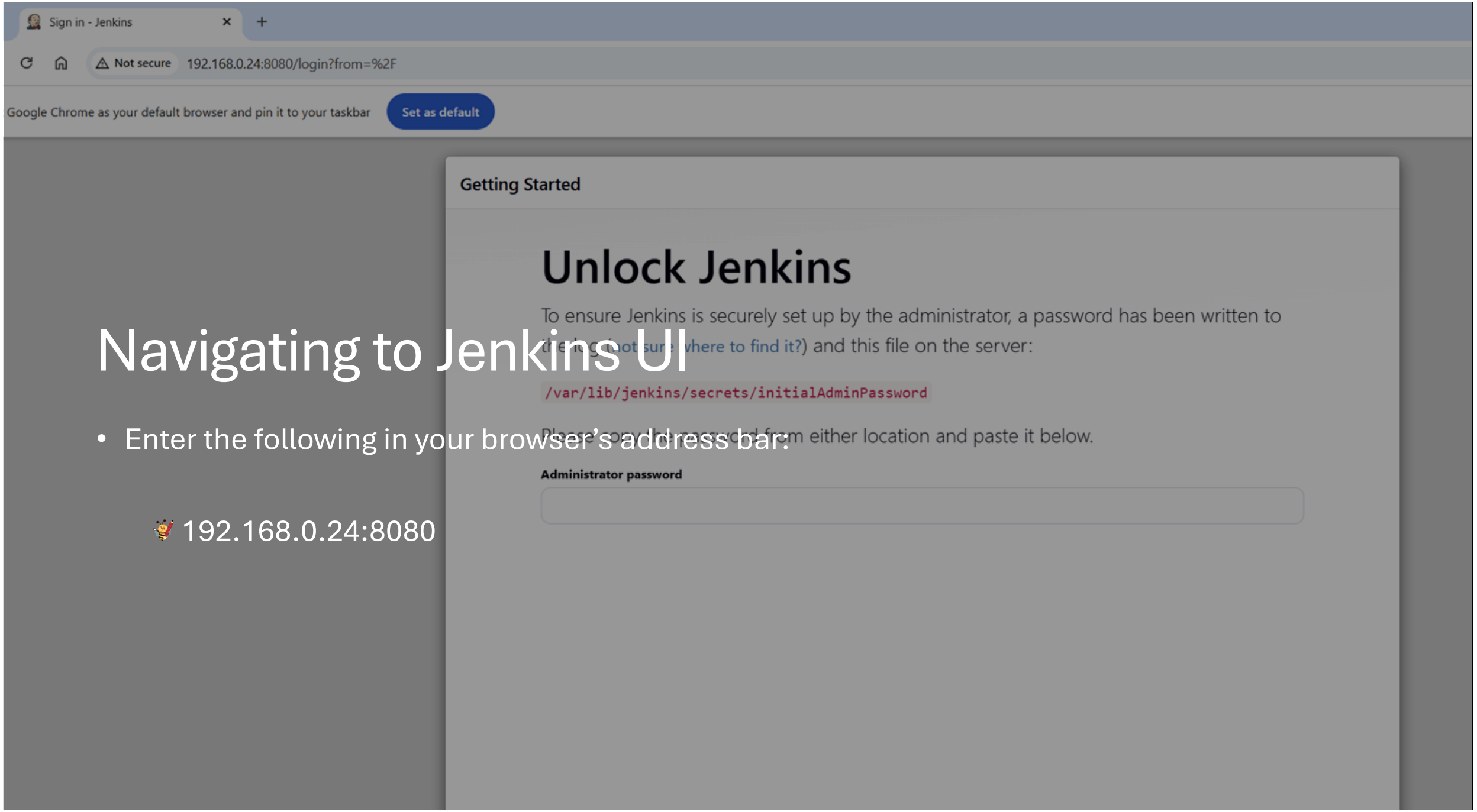
STEP 1 : Install Required Tools

- 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

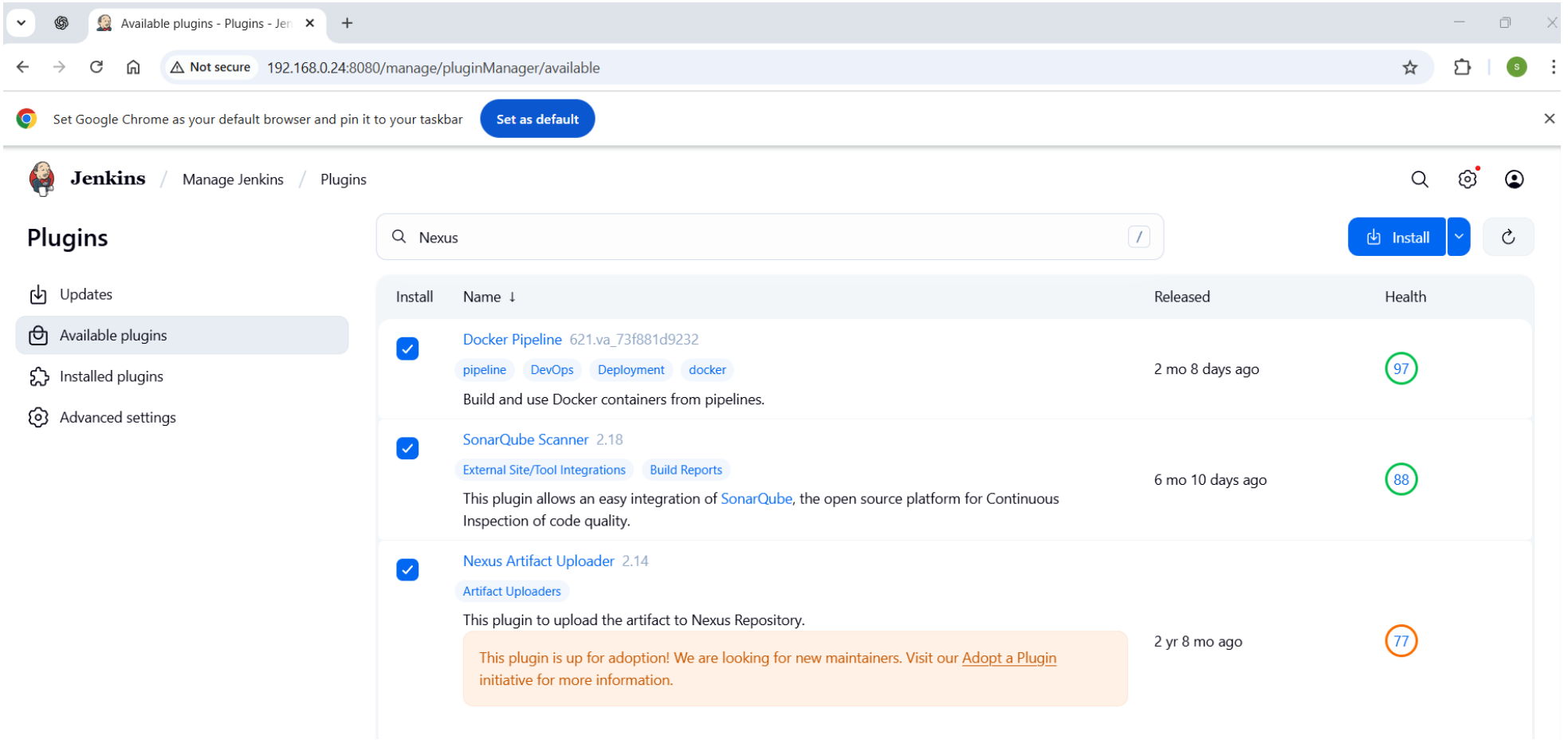
Configure Jenkins with required plugins and global tools.

Installed Jenkins plugins:

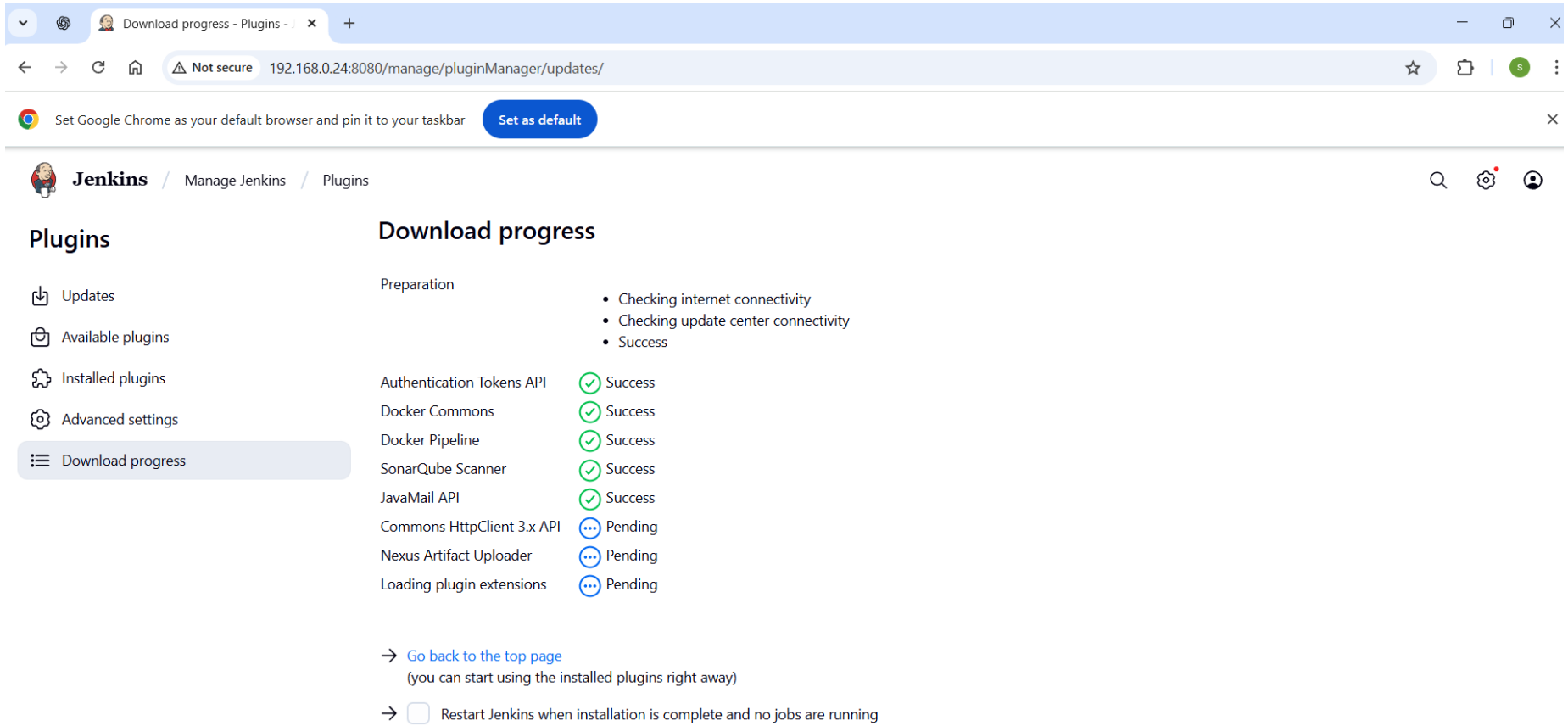
- Docker Pipeline
- Git Plugin
- SonarQube Scanner
- Nexus Artifact

Configure global tools under Manage Jenkins > Global Tool Configuration:

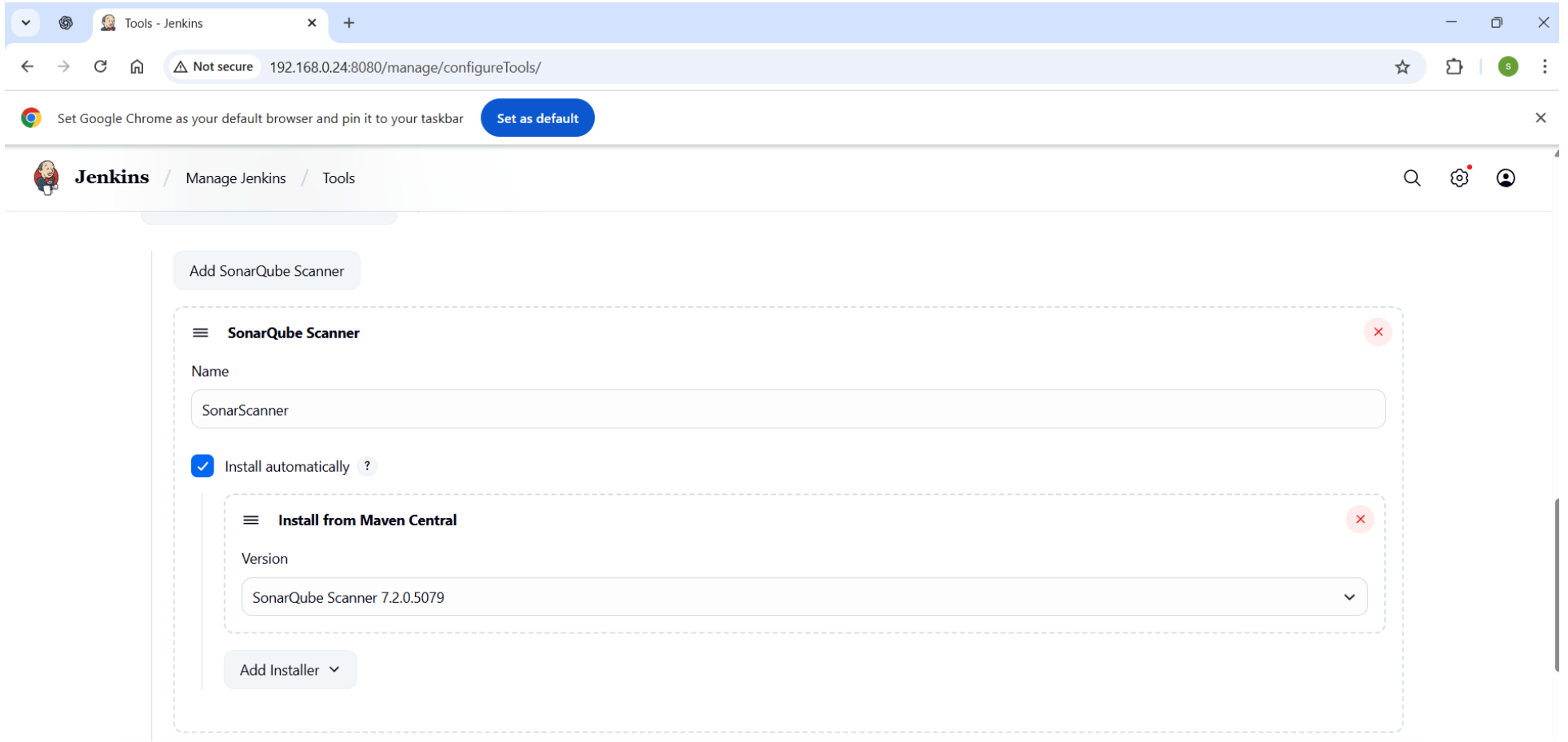
- SonarQube Scanner
- Verified Git and Docker path availability



Install Plugins



Install Plugins

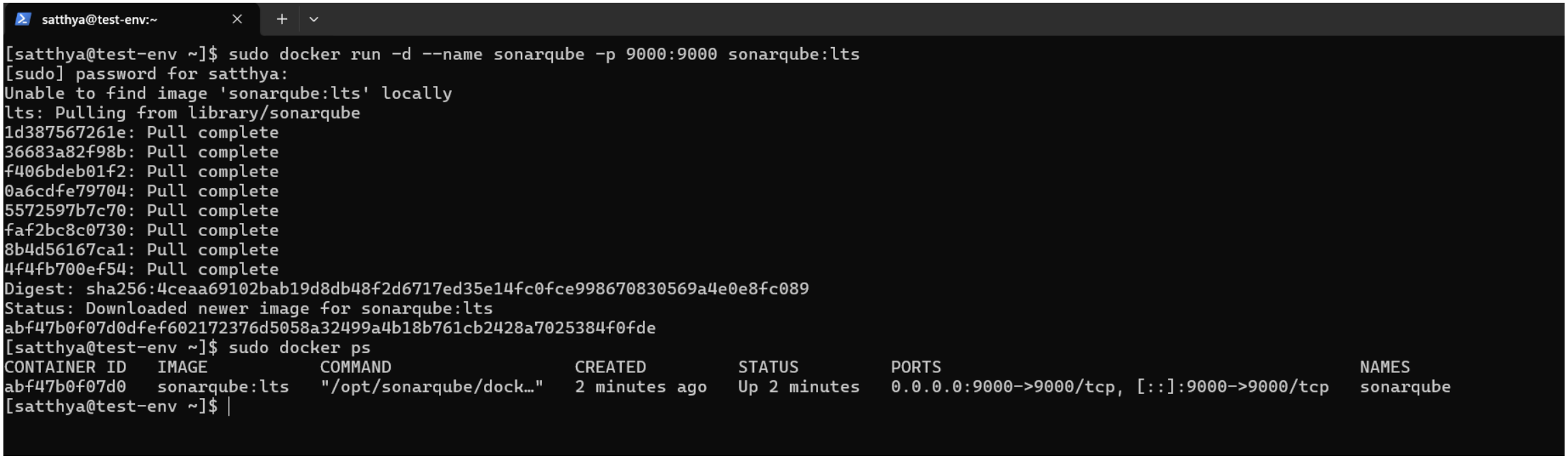


Install Plugins

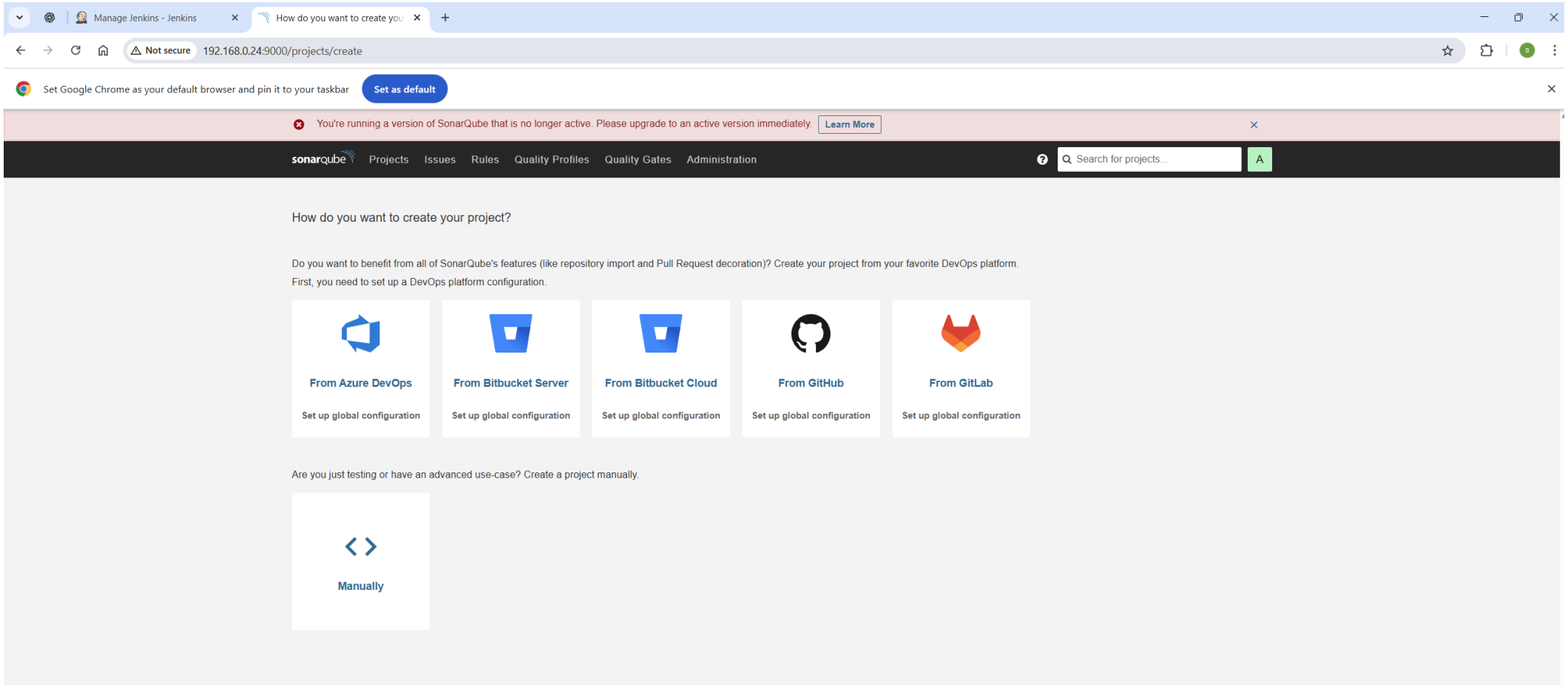
STEP 3 : SonarQube Setup

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



Start SonarQube using Docker



SonarQube UI

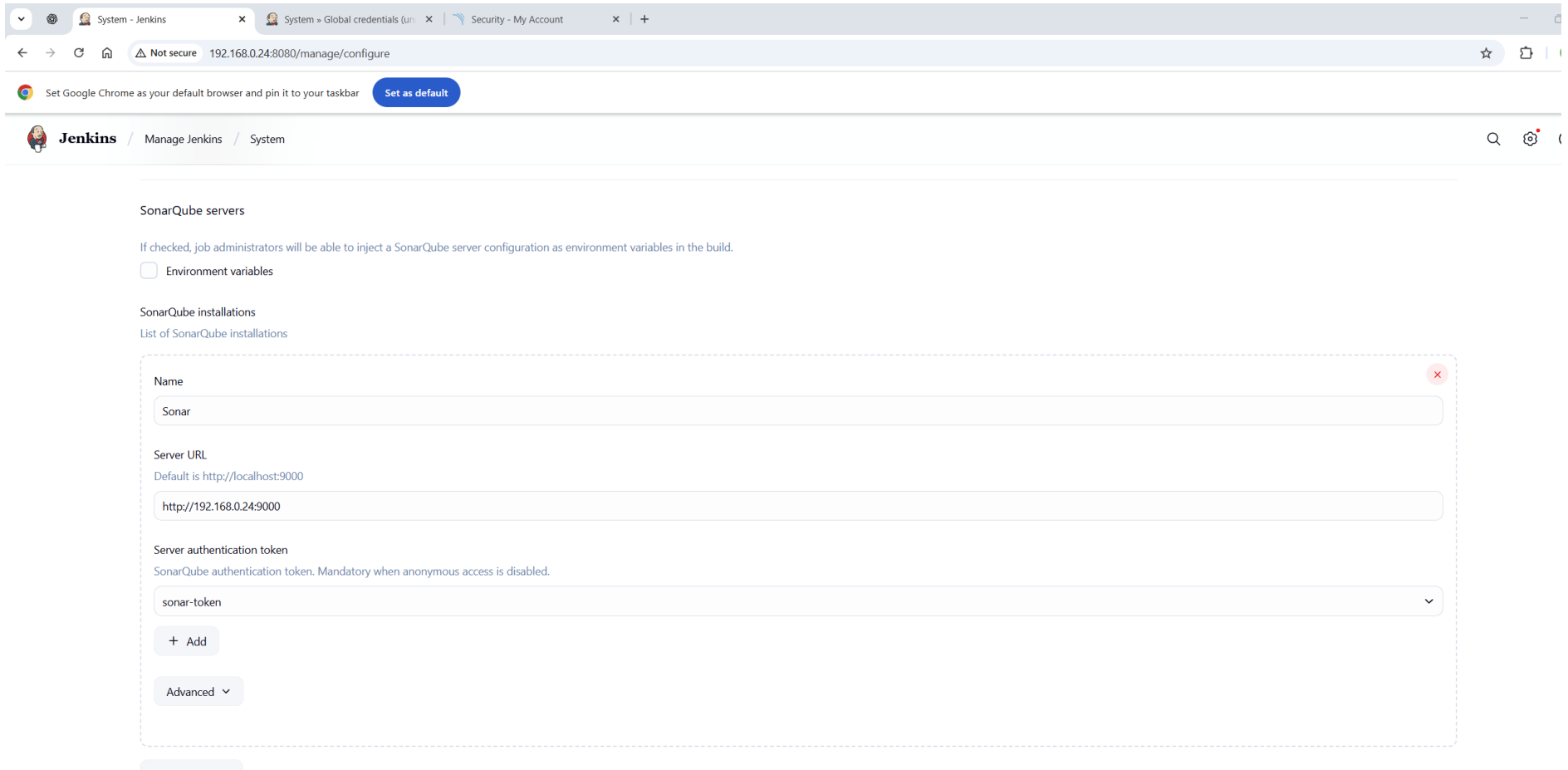
STEP 3 : SonarQube Setup

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



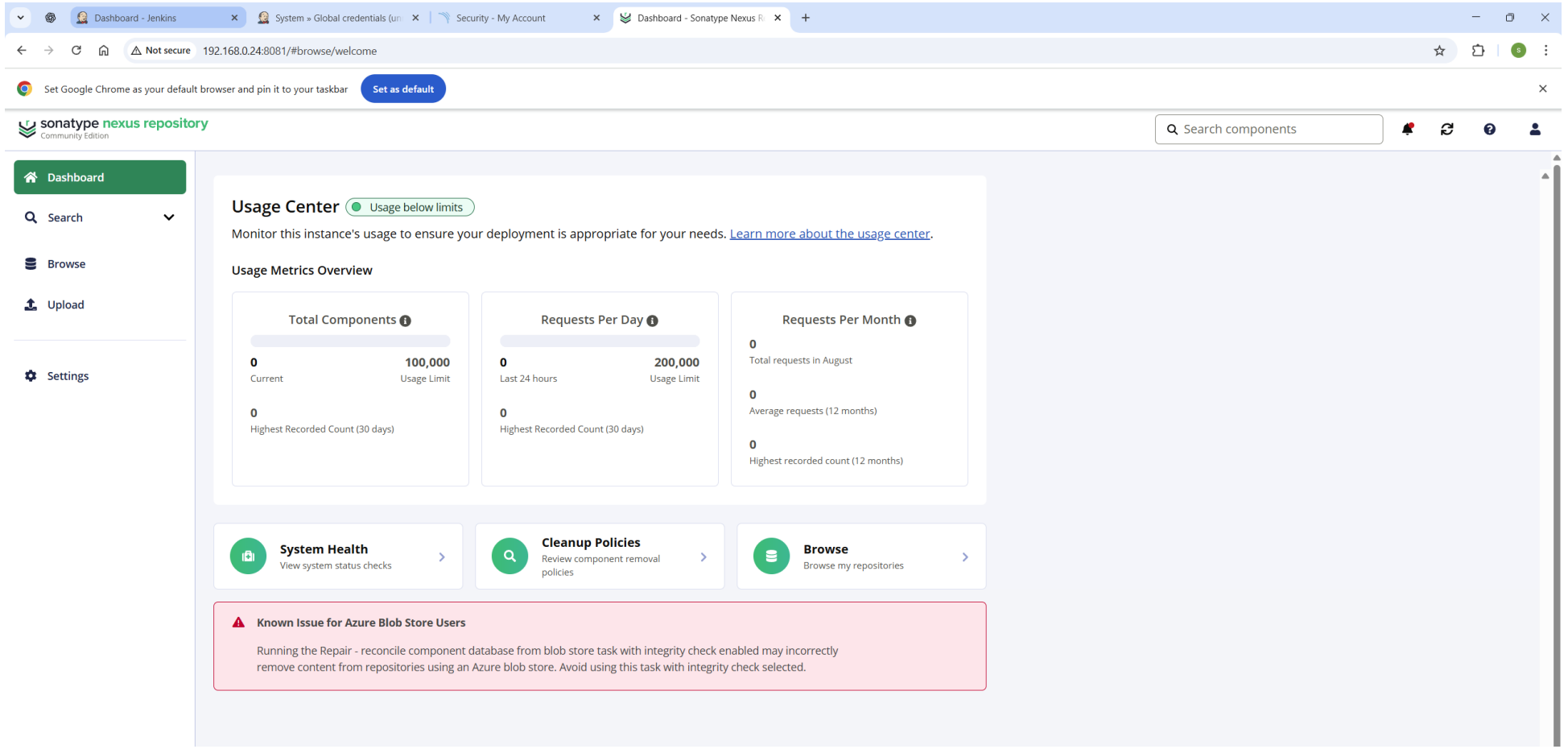
STEP 4 : Nexus Repository Setup

- 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



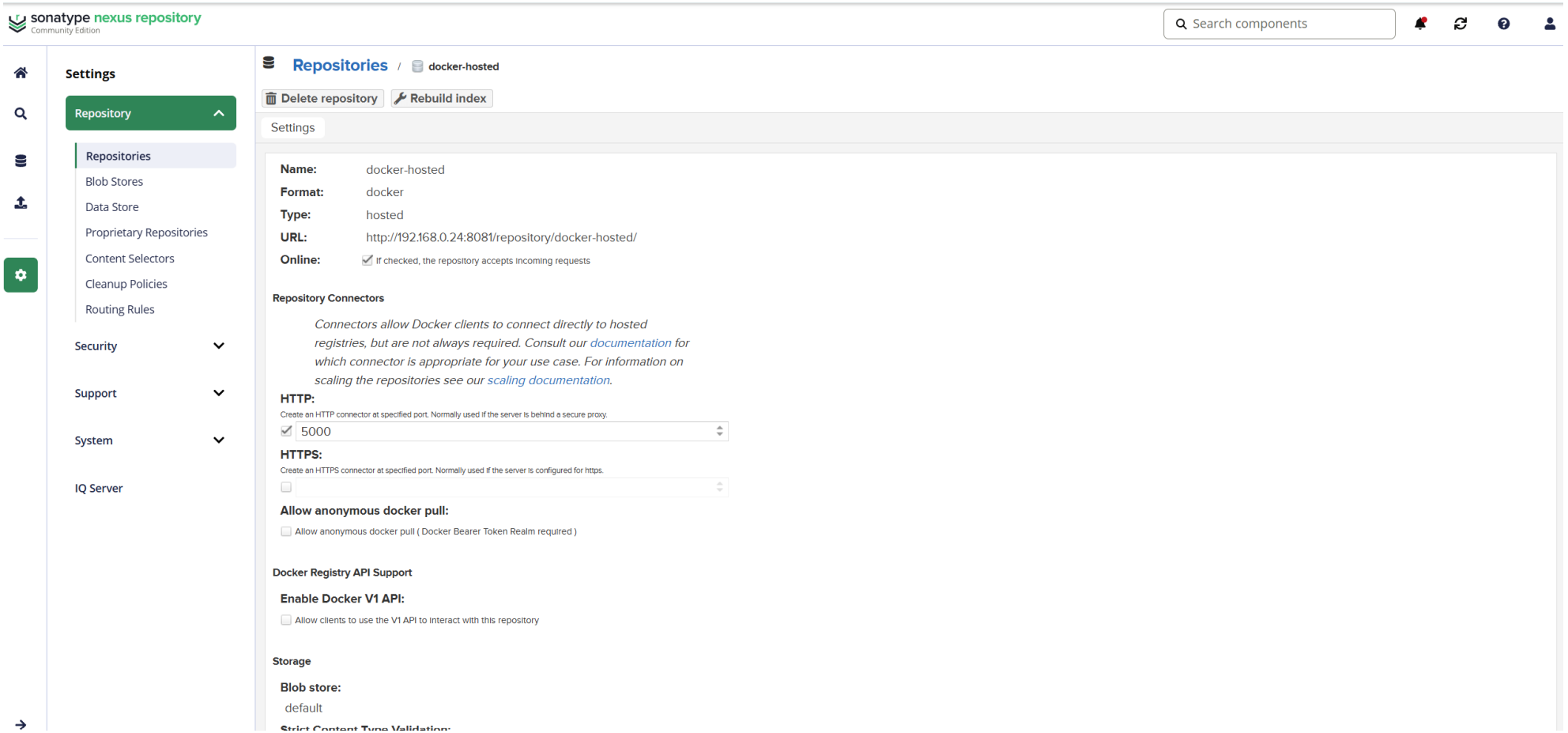
Sonatype Nexus UI

STEP 4 : Nexus Repository Setup

🍯 Create Docker (hosted repository)

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

Sonatype Nexus UI

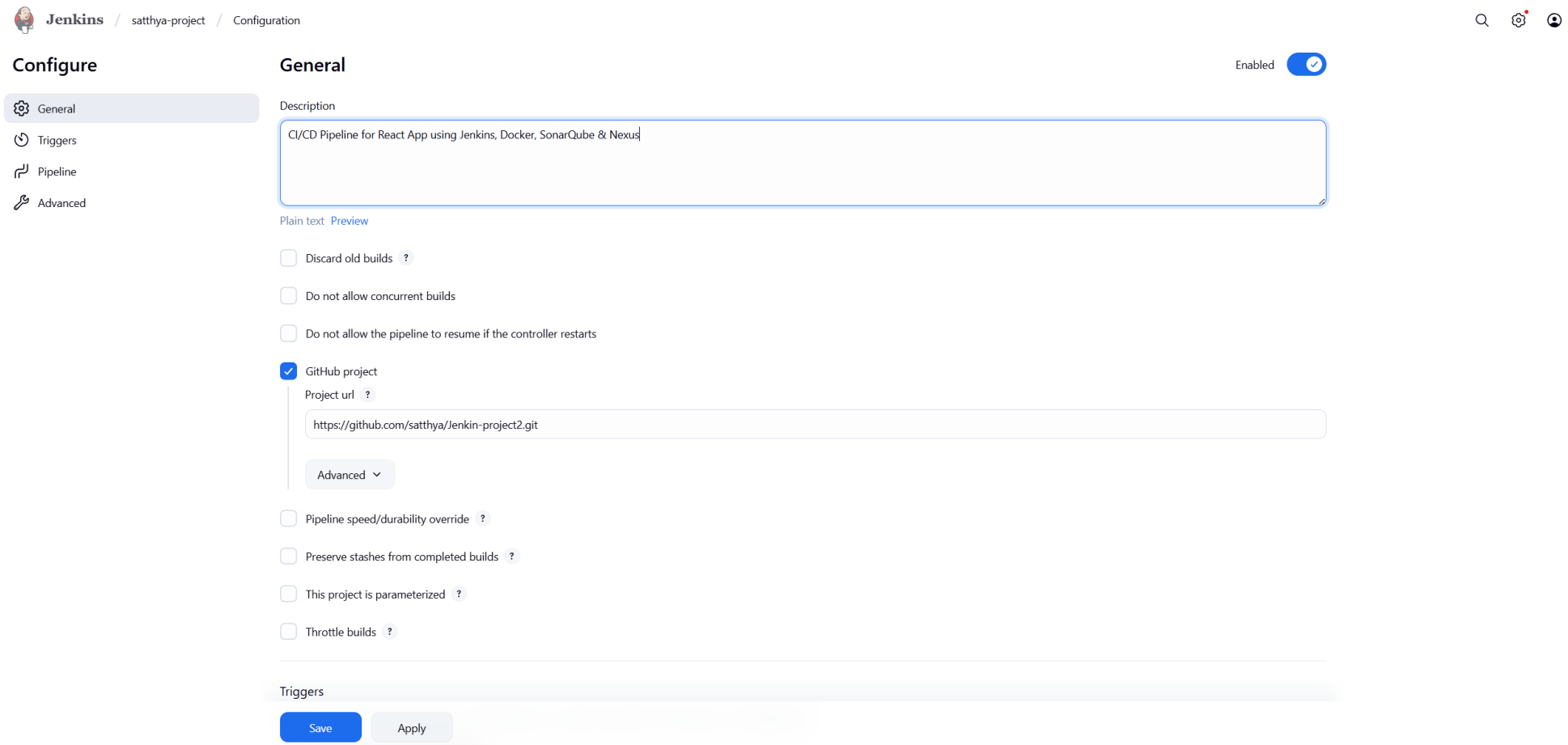


STEP 5: Jenkins Pipeline Setup

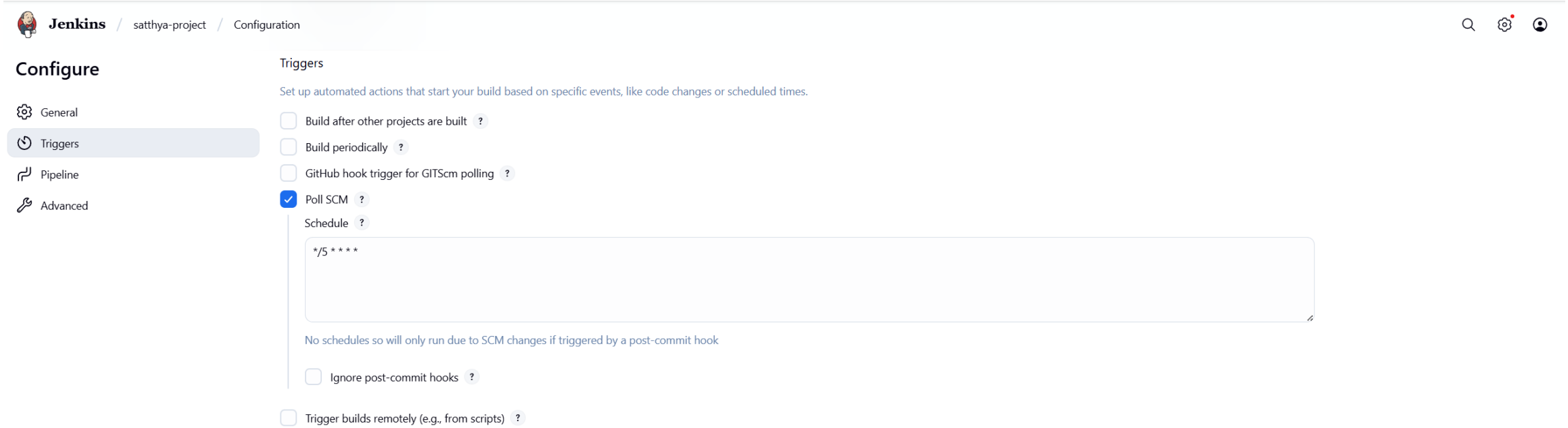
🐝 Automate build, scan, push, and deploy process

- 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

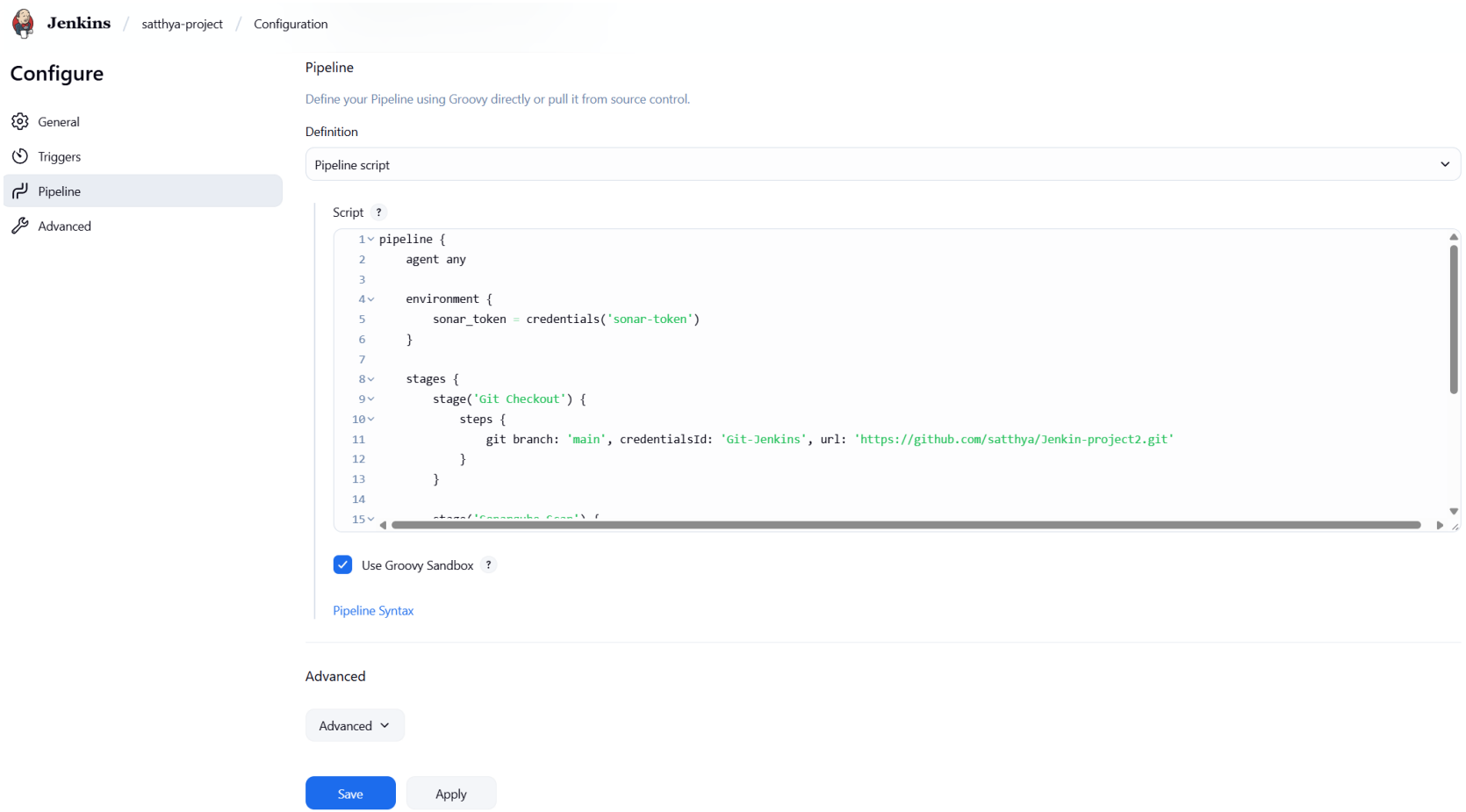
Create new pipeline job



Create new pipeline job



Create new pipeline job



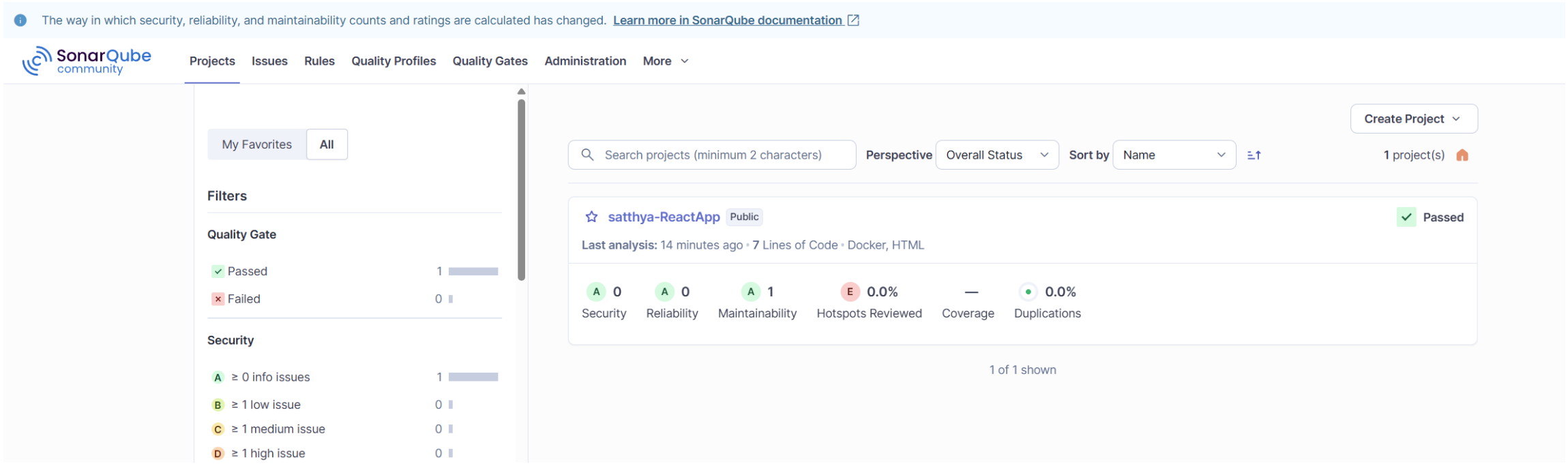
STEP 5: Jenkins Pipeline Setup

- Pipeline script (Groovy Language)

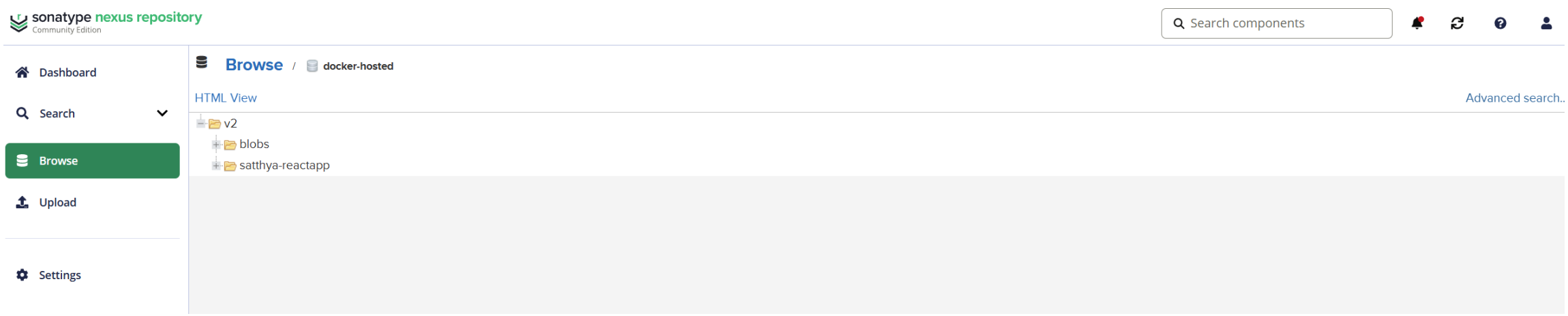
```
pipeline {
    agent any

    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

Final SonarQube UI View



Final Nexus Sonatype UI view



Final React App interface