

# Phase 4: CI/CD Automation - Setup Jenkins Pipeline for Hello World Spring Boot App

## Table of Contents

1. [Environment Setup](#)
2. [Java and Maven Installation](#)
3. [Jenkins Installation](#)
4. [Nexus Repository Installation](#)
5. [Spring Boot Application Setup](#)
6. [Jenkins Configuration](#)
7. [Nexus Configuration](#)
8. [Pipeline Creation](#)
9. [Testing and Execution](#)

## Environment Setup

### 1. Update System Packages

```
sudo apt update && sudo apt upgrade -y
```

### 2. Install Required Dependencies

```
sudo apt install -y curl wget gnupg2 software-properties-common apt-tran  
sport-https ca-certificates lsb-release
```

## Java and Maven Installation

### 1. Install OpenJDK 17

```
sudo apt install -y openjdk-17-jdk openjdk-17-jre
```

## 2. Verify Java Installation

```
java -version  
javac -version
```

```
suhaib@IND-147:~$ java -version  
openjdk version "17.0.15" 2025-04-15  
OpenJDK Runtime Environment (build 17.0.15+6-Debian-1deb12u1)  
OpenJDK 64-Bit Server VM (build 17.0.15+6-Debian-1deb12u1, mixed mode, sharing)  
suhaib@IND-147:~$ javac -version  
javac 17.0.15
```

## 3. Set JAVA\_HOME Environment Variable

```
echo 'export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64' >> ~/.bash  
rc  
echo 'export PATH=$PATH:$JAVA_HOME/bin' >> ~/.bashrc  
source ~/.bashrc
```

## 4. Install Maven

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

```
no vm guests are running outdated hypervisor (qemu) binaries on this host.  
suhaib@IND-147:~$ mvn -version  
Apache Maven 3.8.7  
Maven home: /usr/share/maven  
Java version: 17.0.15, vendor: Debian, runtime: /usr/lib/jvm/java-17-openjdk-amd64  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "5.15.167.4-microsoft-standard-WSL2", arch: "amd64", family: "unix"
```

# Jenkins Installation

## 1. Install Jenkins

```
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
```

```
suhaib@IND-147:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-06-05 10:37:49 IST; 45s ago
     Main PID: 19216 (java)
       Tasks: 62 (limit: 9331)
      Memory: 802.9M
    CGroup: /system.slice/jenkins.service
            └─19216 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --http=

Jun 05 10:37:44 IND-147 jenkins[19216]: b72478fa055e4edaad49832912a8ff81
Jun 05 10:37:44 IND-147 jenkins[19216]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Jun 05 10:37:44 IND-147 jenkins[19216]: *****
Jun 05 10:37:44 IND-147 jenkins[19216]: *****
Jun 05 10:37:44 IND-147 jenkins[19216]: *****
Jun 05 10:37:49 IND-147 jenkins[19216]: 2025-06-05 05:07:49.196+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttain
Jun 05 10:37:49 IND-147 jenkins[19216]: 2025-06-05 05:07:49.225+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady:
Jun 05 10:37:49 IND-147 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Jun 05 10:37:51 IND-147 jenkins[19216]: 2025-06-05 05:07:51.052+0000 [id=71] INFO h.m.DownloadService$Downloadable#loa
Jun 05 10:37:51 IND-147 jenkins[19216]: 2025-06-05 05:07:51.052+0000 [id=71] INFO hudson.util.Retrier#start: Performed
```

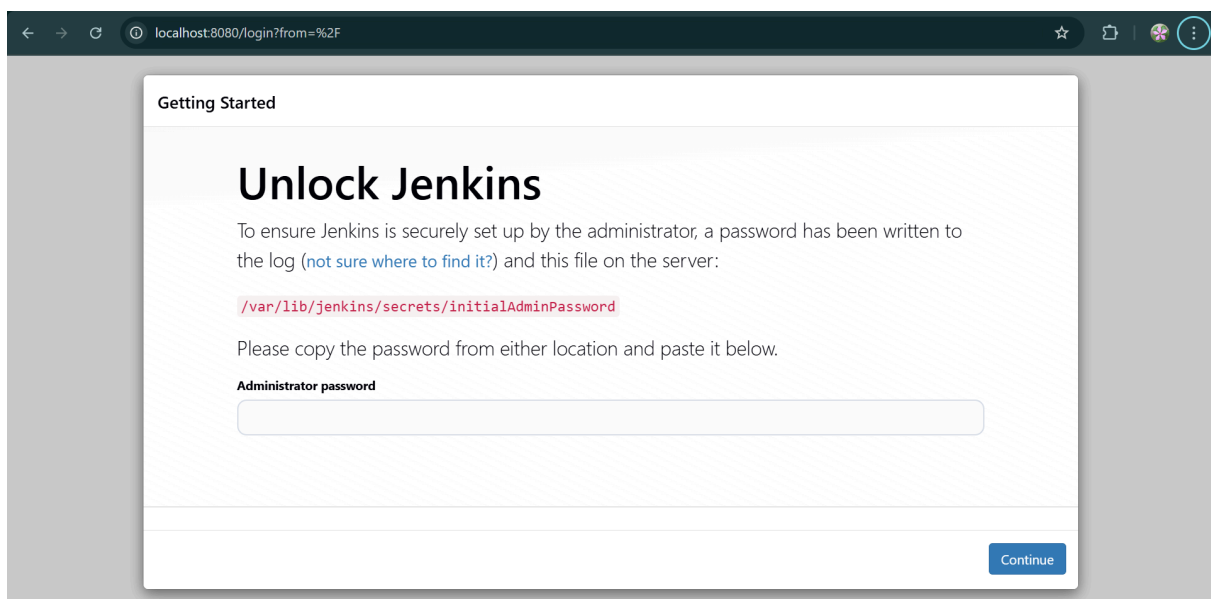
## 2. Get Initial Admin Password

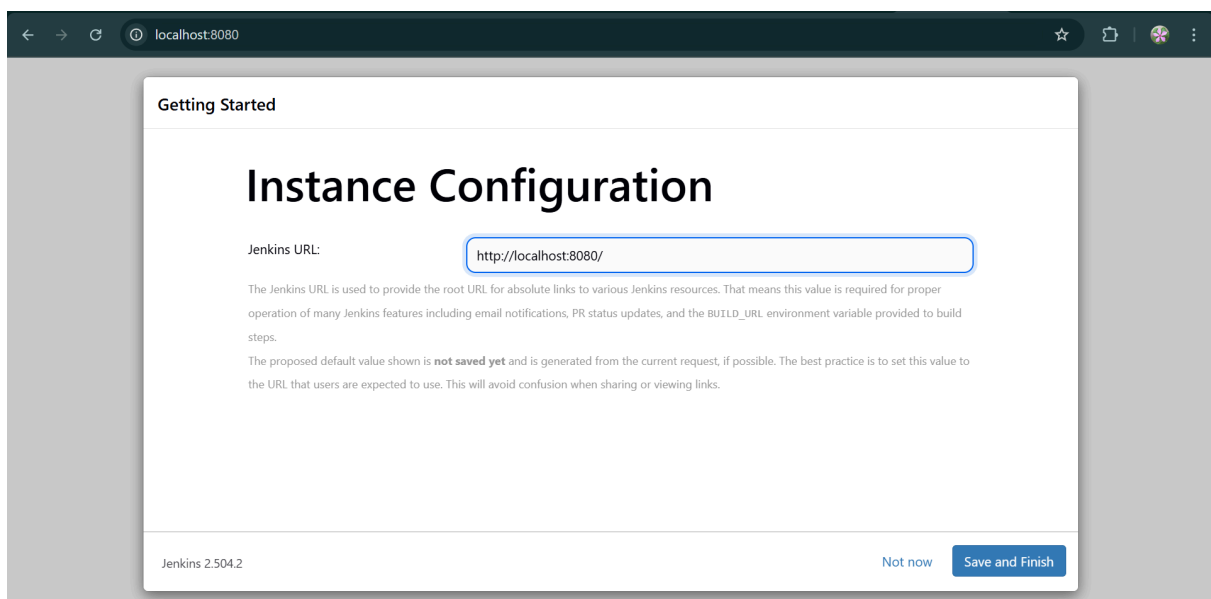
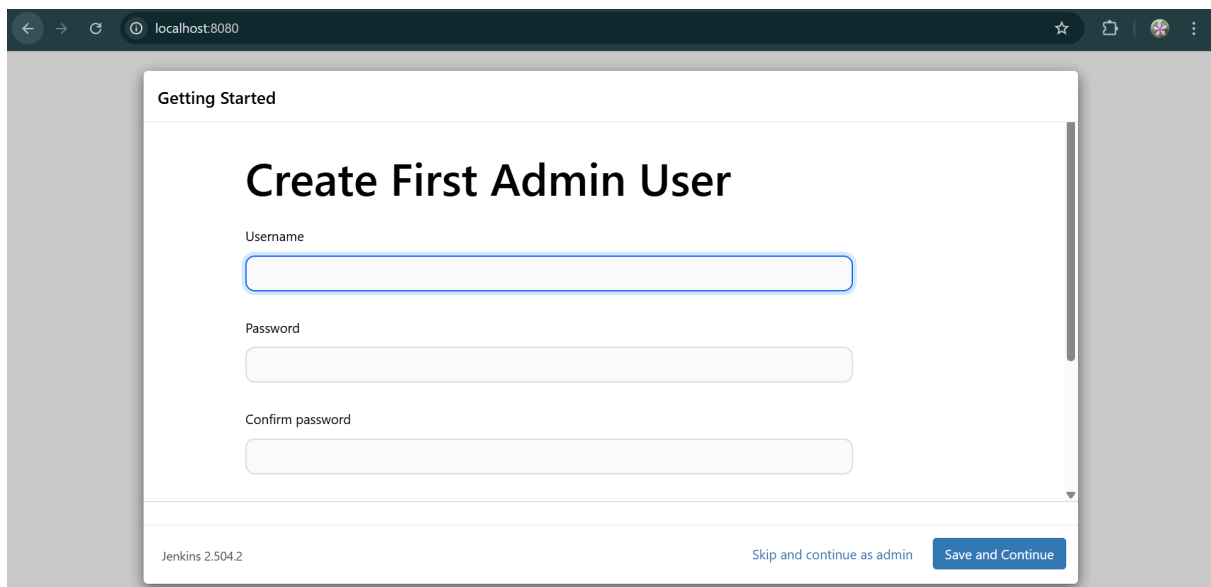
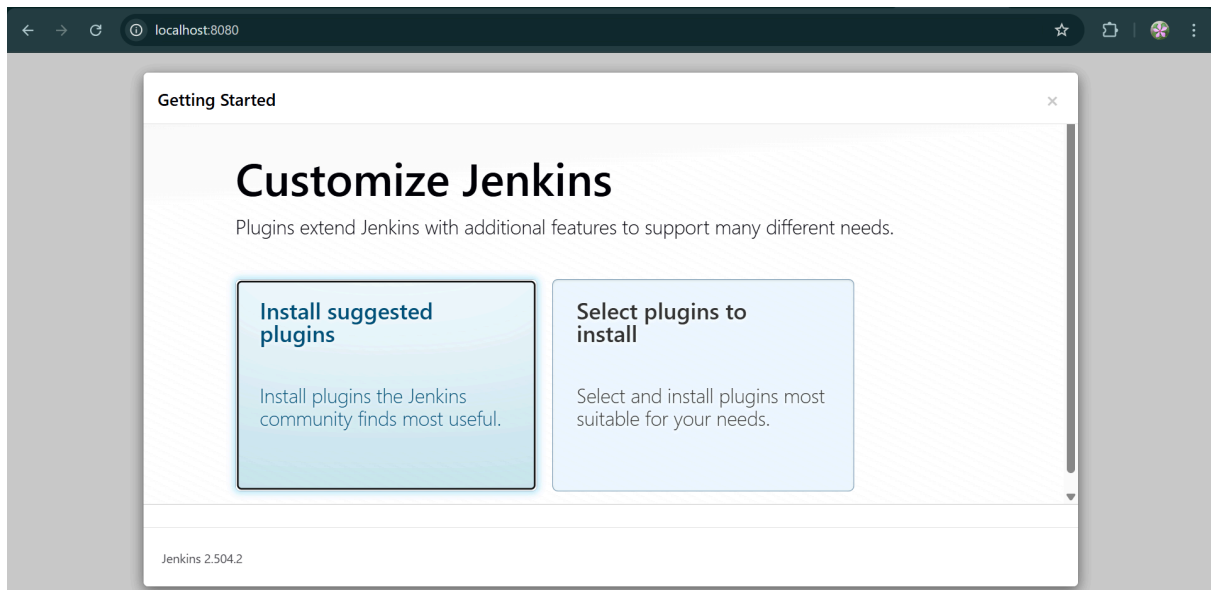
```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

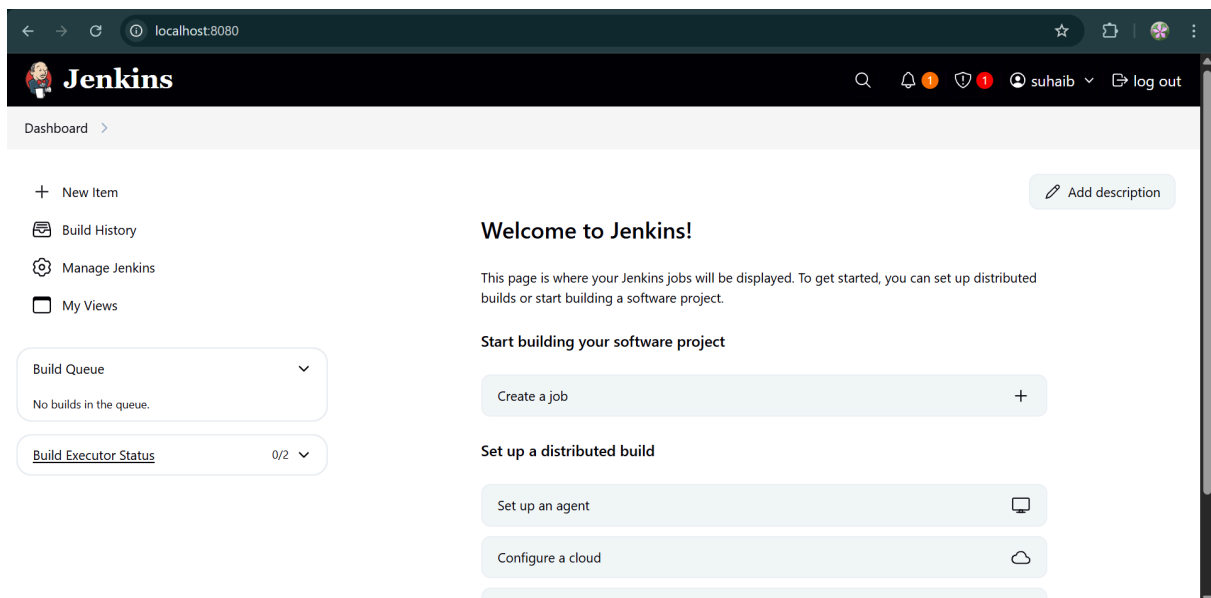
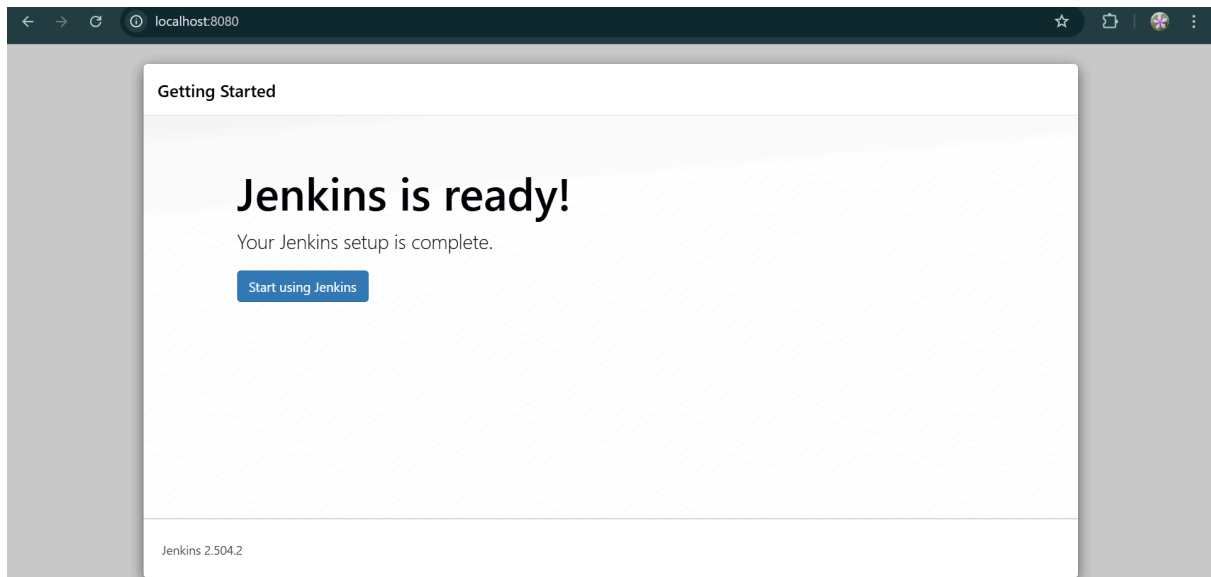
```
suhaib@IND-147:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
b72478fa055e4edaad49832912a8ff81
suhaib@IND-147:~$
```

## 3. Access Jenkins Web Interface

Open browser and navigate to: <http://localhost:8080>







# Nexus Repository Installation

## 1. Create Nexus User

```
sudo useradd -m -d /opt/nexus -s /bin/bash nexus
sudo passwd nexus
```

```
suhaib@IND-147:~$ sudo useradd -m -d /opt/nexus -s /bin/bash nexus
useradd: user 'nexus' already exists
suhaib@IND-147:~$ sudo passwd nexus
New password:
Retype new password:
passwd: password updated successfully
suhaib@IND-147:~$
```

root

## 2. Download and Install Nexus

```
cd /tmp
wget https://download.sonatype.com/nexus/3/nexus-3.80.0-06-linux-x86_64.tar.gz
sudo tar -xzf nexus-3.80.0-06-linux-x86_64.tar.gz -C /opt/
sudo mv /opt/nexus-3* /opt/nexus
sudo chown -R nexus:nexus /opt/nexus
sudo chown -R nexus:nexus /opt/sonatype-work
```

```
suhaib@IND-147:/tmp$ sudo tar -xzf nexus-3.80.0-06-linux-x86_64.tar.gz -C /opt/
suhaib@IND-147:/tmp$ ls
cron_test.log
hsperfdata_jenkins
hsperfdata_suhaib
jetty-0_0_0-8080-war-_-any-14746703135340567815
nexus-3.80.0-06-linux-x86_64.tar.gz
systemd-private-a355896401234b86b91982c9859ce120-ntpsec.service-ntLbIV
systemd-private-a355896401234b86b91982c9859ce120-postfix-mta-sts-resolver.service-wB17VN
systemd-private-a355896401234b86b91982c9859ce120-systemd-logind.service-X0fBPD
winstone8372487243049490731.jar
suhaib@IND-147:/tmp$ sudo mv /opt/nexus-3* /opt/nexus
suhaib@IND-147:/tmp$ sudo chown -R nexus:nexus /opt/nexus
suhaib@IND-147:/tmp$ sudo chown -R nexus:nexus /opt/sonatype-work
suhaib@IND-147:/tmp$
```

## 3. Configure Nexus as Service

```
sudo nano /etc/systemd/system/nexus.service
```

Add the following content:

```
[Unit]
Description=nexus service
After=network.target

[Service]
Type=forking
LimitNOFILE=65536
ExecStart=/opt/nexus/bin/nexus start
ExecStop=/opt/nexus/bin/nexus stop
User=nexus
Restart=on-abort
TimeoutSec=600
```

[Install]

WantedBy=multi-user.target

```
suhaib@IND-147: /etc/systemd/system/nexus.service *
[Unit]
Description=nexus service
After=network.target

[Service]
Type=forking
LimitNOFILE=65536
ExecStart=/opt/nexus/bin/nexus start
ExecStop=/opt/nexus/bin/nexus stop
User=nexus
Restart=on-abort
TimeoutSec=600

[Install]
WantedBy=multi-user.target
```

## 4. Start Nexus Service

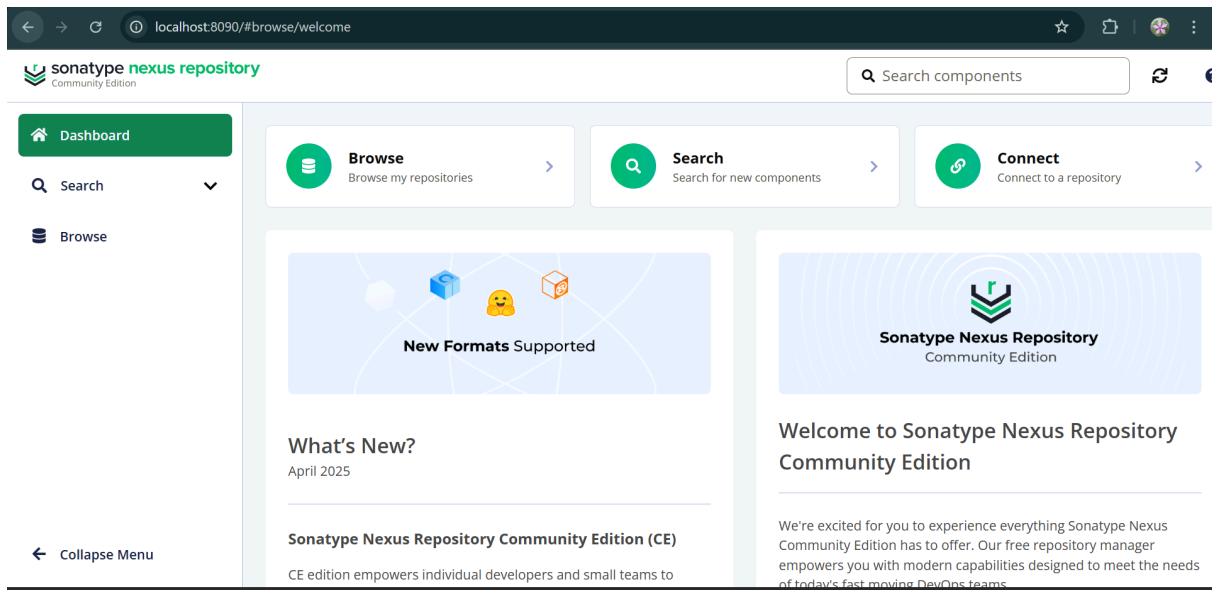
```
sudo systemctl daemon-reload
sudo systemctl enable nexus
sudo systemctl start nexus
sudo systemctl status nexus
```

```
suhaib@IND-147:/$ sudo systemctl daemon-reload
suhaib@IND-147:/$ sudo systemctl enable nexus
Created symlink /etc/systemd/system/multi-user.target.wants/nexus.service → /etc/systemd/system/nexus.service.
suhaib@IND-147:/$ sudo systemctl start nexus
suhaib@IND-147:/$ sudo systemctl status nexus
● nexus.service - nexus service
   Loaded: loaded (/etc/systemd/system/nexus.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-06-05 10:53:43 IST; 4s ago
     Process: 20481 ExecStart=/opt/nexus/bin/nexus start (code=exited, status=0/SUCCESS)
    Main PID: 20722 (java)
       Tasks: 31 (limit: 9331)
      Memory: 318.8M
   CGroup: /system.slice/nexus.service
           └─20722 /opt/nexus/jdk/temurin_17.0.13_11_linux_x86_64/jdk-17.0.13+11/bin/java -server -Dnexus.installer.type=linux-x86_64

Jun 05 10:53:43 IND-147 systemd[1]: Starting nexus.service - nexus service...
Jun 05 10:53:43 IND-147 nexus[20481]: /opt/nexus/bin/nexus: 155: [: not found
Jun 05 10:53:43 IND-147 systemd[1]: Started nexus.service - nexus service.
Jun 05 10:53:43 IND-147 nexus[20481]: Starting nexus
lines 1-14/14 (END)
```

## 5. Access Nexus Web Interface

Open browser and navigate to: <http://localhost:8081>

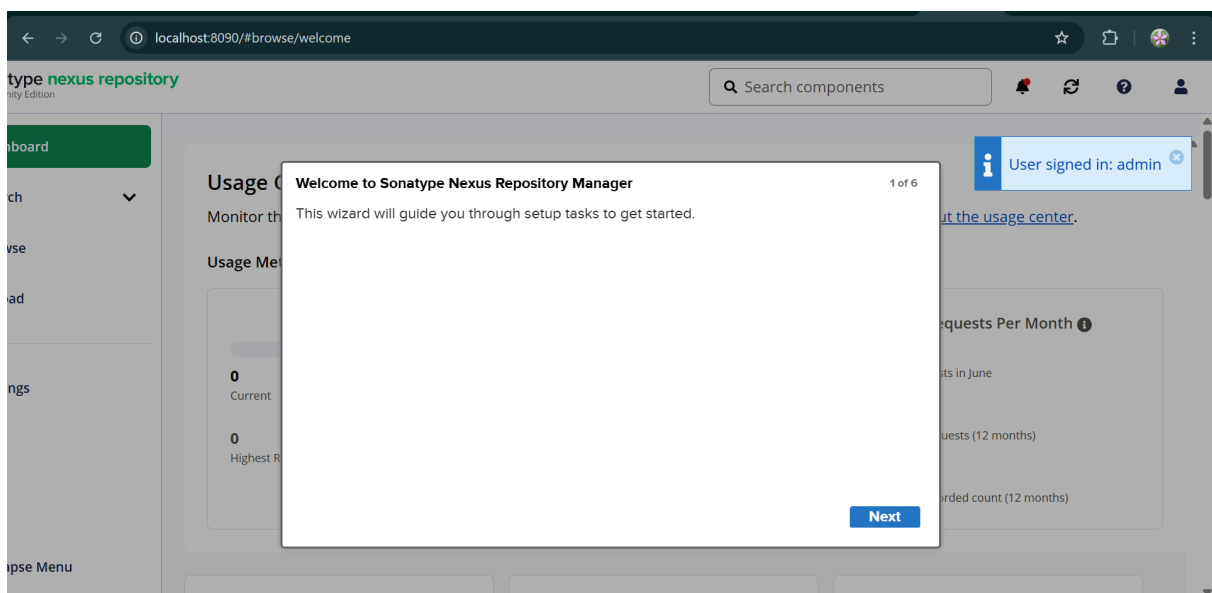


Default credentials:

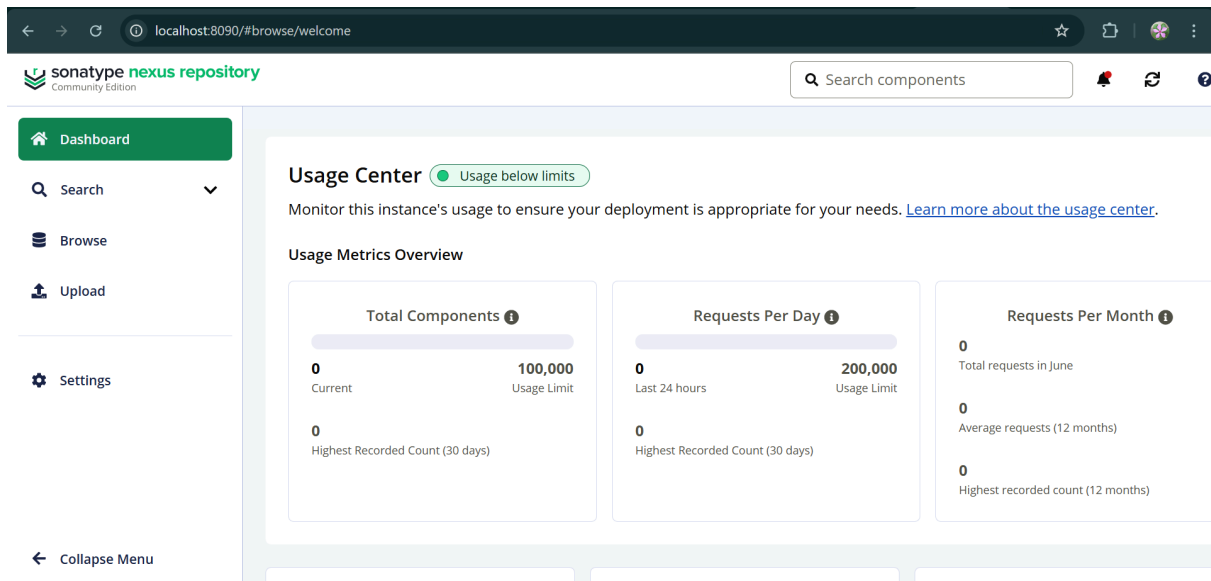
- Username: admin
- Password: Check `/opt/sonatype-work/nexus3/admin.password`

```
sudo cat /opt/sonatype-work/nexus3/admin.password
```

```
suhaib@IND-147:/$ sudo cat /opt/sonatype-work/nexus3/admin.password
03762b38-1adb-4891-8738-d89e21f7145bsuhaib@IND-147:/$
```







# Spring Boot Application Setup

## 1. Create Project Directory

```
mkdir -p ~/jenkins-projects/hello-world-spring-boot
cd ~/jenkins-projects/hello-world-spring-boot
```

## 2. Initialize Maven Project

```
mvn archetype:generate -DgroupId=com.example -DartifactId=hello-world-spring-boot -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
cd hello-world-spring-boot
```

```

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-release-plugin/2.5.3/maven-release-plugin-2.5.3.jar (53 kB at 1.4 MB/s)
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO] >>> maven-archetype-plugin:3.3.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO] <<< maven-archetype-plugin:3.3.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:3.3.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 2.1 MB/s)
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: /home/suhaib/jenkins-projects/hello-world-spring-boot
[INFO] Parameter: package, Value: com.example
[INFO] Parameter: groupId, Value: com.example
[INFO] Parameter: artifactId, Value: hello-world-spring-boot
[INFO] Parameter: packageName, Value: com.example
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/suhaib/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 12.030 s
[INFO] Finished at: 2025-06-05T11:40:06+05:30
[INFO] -----
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot$ |

```

### 3. Update pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.0</version>
    <relativePath/>
  </parent>

  <groupId>com.example</groupId>
  <artifactId>hello-world-spring-boot</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>hello-world-spring-boot</name>
  <description>Hello World Spring Boot Application</description>

  <properties>

```

```

<java.version>17</java.version>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M9</version>
    </plugin>
  </plugins>
</build>

<distributionManagement>
  <repository>

```

```

    <id>nexus-releases</id>
    <name>Nexus Release Repository</name>
    <url>http://localhost:8090/repository/maven-releases/</url>
  </repository>
  <snapshotRepository>
    <id>nexus-snapshots</id>
    <name>Nexus Snapshot Repository</name>
    <url>http://localhost:8090/repository/maven-snapshots/</url>
  </snapshotRepository>
</distributionManagement>
</project>

```

## 4. Create Main Application Class

```
mkdir -p src/main/java/com/example/helloworldspringboot
```

Create `src/main/java/com/example/helloworldspringboot/HelloWorldApplication.java` :

```

package com.example.helloworldspringboot;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class HelloWorldApplication {

    public static void main(String[] args) {
        SpringApplication.run(HelloWorldApplication.class, args);
    }

    @GetMapping("/")
    public String hello() {
        return "Hello World from Spring Boot!";
    }
}

```

```

    @GetMapping("/health")
    public String health() {
        return "Application is running!";
    }
}

```

## 5. Create Test Class

Create `src/test/java/com/example/helloworldspringboot/HelloWorldApplicationTest.java` :

```

package com.example.helloworldspringboot;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit.jupiter.SpringJUnitConfig;

@SpringBootTest
@SpringJUnitConfig
class HelloWorldApplicationTest {

    @Test
    void contextLoads() {
        // Test that Spring context loads successfully
    }
}

```

## 6. Create Jenkinsfile

Create `Jenkinsfile` in the root directory:

```

pipeline {
    agent any

    tools {
        maven 'Maven-3.9.0'
        jdk 'OpenJDK-17'
    }

    environment {

```

```

// Nexus and GitHub info
NEXUS_VERSION = "nexus3"
NEXUS_PROTOCOL = "http"
NEXUS_URL = "localhost:8090"
NEXUS_REPOSITORY = "maven-snapshots"
NEXUS_CREDENTIAL_ID = "nexus-credentials"
GITHUB_REPO = "https://github.com/suhaibmdv/jenkins_springboot_nexus.git"

// POM details (manually entered to avoid security sandbox issues)
ARTIFACT_ID = "hello-world-spring-boot"
GROUP_ID = "com.example"
VERSION = "1.0.0-SNAPSHOT"
PACKAGING = "jar"
}

stages {
    stage('Checkout') {
        steps {
            echo 'Checking out source code from GitHub...'
            git branch: 'main', url: "${GITHUB_REPO}"
        }
    }

    stage('Build') {
        steps {
            echo 'Building the application...'
            sh 'mvn clean compile'
        }
    }

    stage('Test') {
        steps {
            echo 'Running tests...'
            sh 'mvn test'
        }
        post {
            always {

```

```

        junit testResults: 'target/surefire-reports/*.xml', allowEmptyRes
ults: true
    }
}

stage('Package') {
    steps {
        echo 'Packaging the application...'
        sh 'mvn package -DskipTests'
    }
    post {
        success {
            archiveArtifacts artifacts: 'target/*.jar', fingerprint: true
        }
    }
}

stage('Deploy to Nexus') {
    steps {
        echo 'Deploying artifacts to Nexus...'
        script {
            def repositoryName = VERSION.endsWith("SNAPSHOT") ? "ma
ven-snapshots" : "maven-releases"
            def artifactPath = "target/${ARTIFACT_ID}-${VERSION}.${PAC
KAGING}"

            if (!fileExists(artifactPath)) {
                error "Artifact file not found: ${artifactPath}"
            }

            nexusArtifactUploader(
                nexusVersion: NEXUS_VERSION,
                protocol: NEXUS_PROTOCOL,
                nexusUrl: NEXUS_URL,
                groupId: GROUP_ID,
                version: VERSION,
                repository: repositoryName,

```

```

        credentialsId: NEXUS_CREDENTIAL_ID,
        artifacts: [
            [
                artifactId: ARTIFACT_ID,
                classifier: '',
                file: artifactPath,
                type: PACKAGING
            ],
            [
                artifactId: ARTIFACT_ID,
                classifier: '',
                file: "pom.xml",
                type: "pom"
            ]
        ]
    )
    echo "Artifact deployed successfully to Nexus"
}
}

stage('Integration Tests') {
    steps {
        echo 'Running integration tests...'
        sh 'mvn verify -DskipUTs'
    }
}

post {
    always {
        echo 'Cleaning workspace...'
        cleanWs()
    }
    success {
        echo 'Pipeline executed successfully!'
        // Optional email notification
        /*

```



```

        emailx(
            subject: "SUCCESS: Job '${env.JOB_NAME} ${env.BUILD_NUMB
ER}'",
            body: "Good news! The build ${env.BUILD_URL} completed succe
ssfully.",
            to: "${env.CHANGE_AUTHOR_EMAIL}"
        )
    */
}
failure {
    echo 'Pipeline failed!'
    // Optional email notification
    /*
    emailx(
        subject: "FAILED: Job '${env.JOB_NAME} ${env.BUILD_NUMBE
R}'",
        body: "Build failed. Check console output at ${env.BUILD_URL}",
        to: "${env.CHANGE_AUTHOR_EMAIL}"
    )
    */
}
}
}
}

```

```

suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$ tree
.
├── Jenkinsfile
├── pom.xml
├── src
│   ├── main
│   │   ├── java
│   │   │   └── com
│   │   │       └── example
│   │   │           ├── App.java
│   │   │           └── helloworldspringboot
│   │   │               └── HelloWorldApplication.java
│   └── test
│       ├── java
│       │   ├── com
│       │   │   └── example
│       │   │       └── helloworldspringboot
│       │   │           └── HelloWorldApplicationTest.java
└── target

```

## 7. Initialize Git Repository

```

git init
git add .
git commit -m "Initial commit with Spring Boot Hello World application"
git branch -M main

```

```
git remote add origin https://github.com/suhaibmdv/jenkins_springboot_nexus.git
git push -u origin main
```

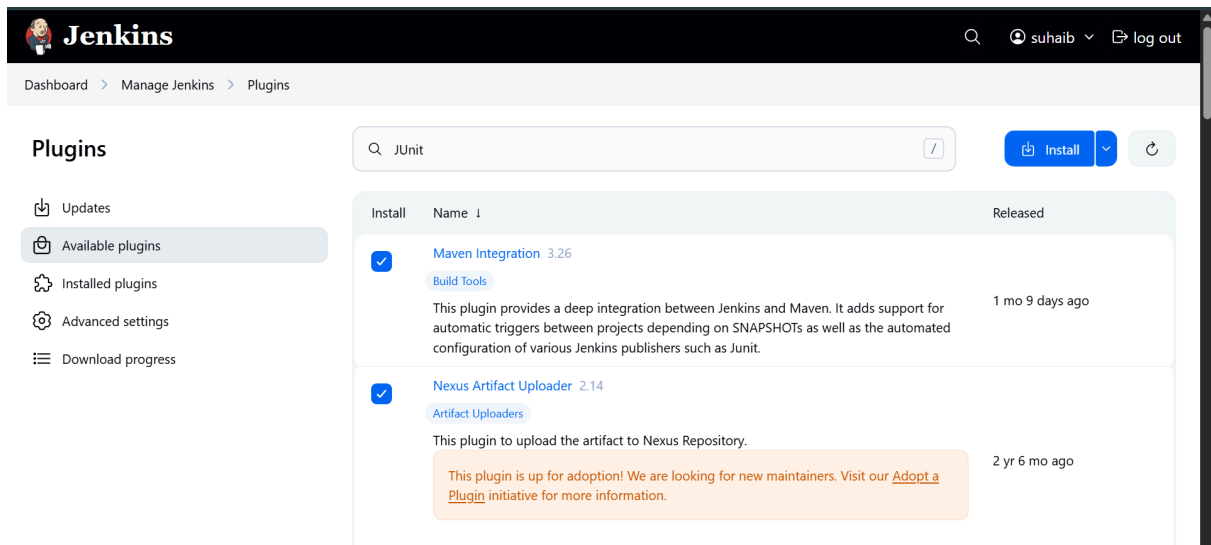
```
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$ git add .
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$ git commit -m "Initial commit with Spring Boot Hello World application"
[master (root-commit) ec2a9cb] Initial commit with Spring Boot Hello World application
6 files changed, 279 insertions(+)
create mode 100644 Jenkinsfile
create mode 100644 pom.xml
create mode 100644 src/main/java/com/example/App.java
create mode 100644 src/main/java/com/example/helloworldspringboot/HelloWorldApplication.java
create mode 100644 src/main/java/com/example/helloworldspringboot/HelloWorldApplicationTest.java
create mode 100644 src/test/java/com/example/AppTest.java
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$ git branch -M main
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$ git remote add origin https://github.com/suhaibmdv/jenkins_springboot_nexus.git
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$ git push -u origin main
Username for 'https://github.com': suhaibmdv
Password for 'https://suhaibmdv@github.com':
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (18/18), 3.43 KiB | 702.00 KiB/s, done.
Total 18 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/suhaibmdv/jenkins_springboot_nexus.git
* [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$ |
```

# Jenkins Configuration

## 1. Install Required Plugins

Access Jenkins at <http://localhost:8080> and install these plugins:

- Maven Integration Plugin
- Nexus Artifact Uploader
- Pipeline Plugin
- Git Plugin
- Email Extension Plugin
- JUnit Plugin
- Pipeline Utility Steps

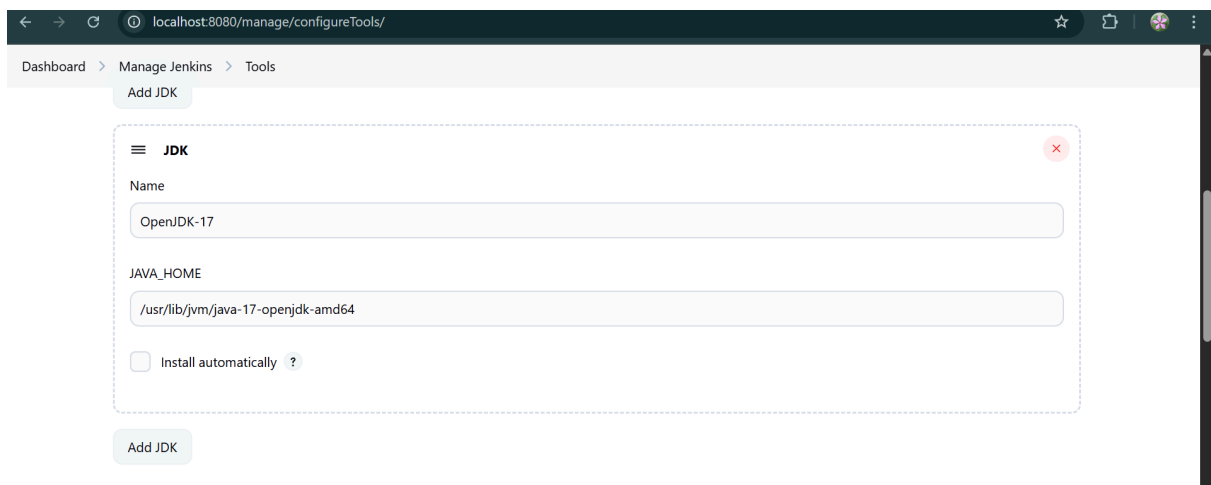


## 2. Configure Global Tools

Go to **Manage Jenkins** → **Global Tool Configuration**:

### Configure JDK:

- Name: `OpenJDK-17`
- JAVA\_HOME: `/usr/lib/jvm/java-17-openjdk-amd64`



### Configure Maven:

- Name: `Maven-3.9.0`
- Install automatically or set MAVEN\_HOME

## 3. Configure Credentials

Go to **Manage Jenkins** → **Manage Credentials** → **Global**:

## Add Nexus credentials:

- ID: `nexus-credentials`
- Username: `admin`
- Password: [Nexus admin password]

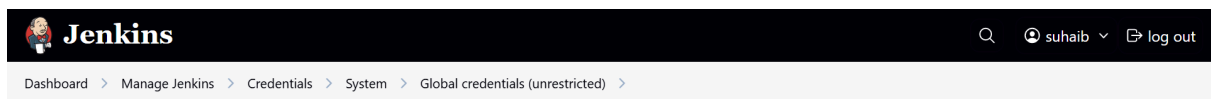
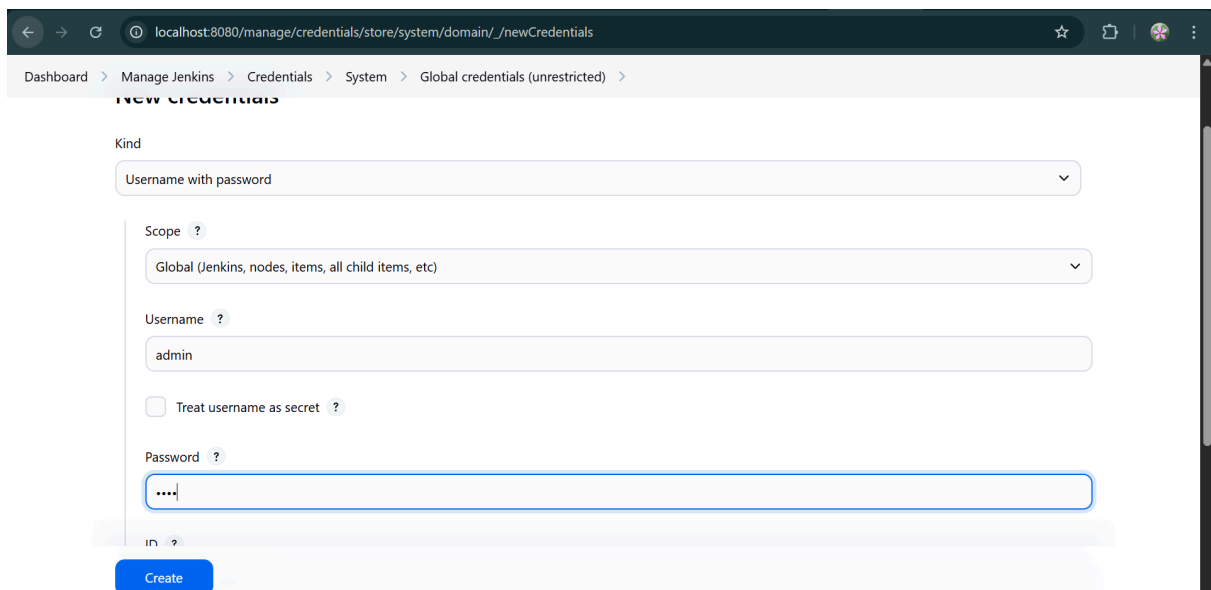


### Global credentials (unrestricted)

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

ID	Name	Kind	Description
This credential domain is empty. How about <a href="#">adding some credentials?</a>			

Icon: S M L



### Global credentials (unrestricted)

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

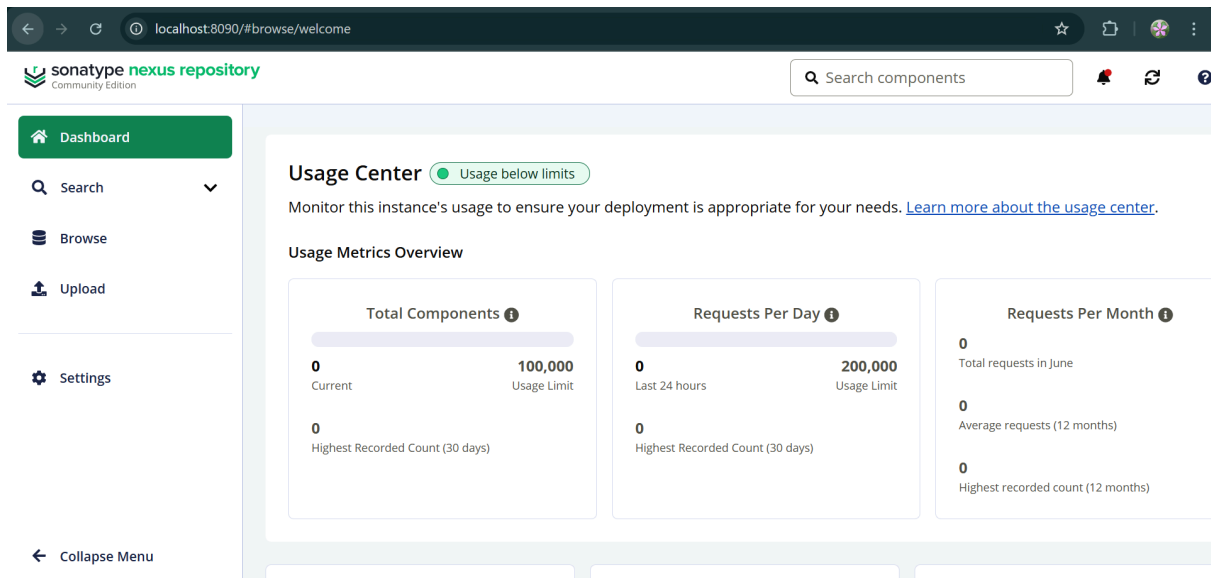
ID	Name	Kind	Description
<code>nexus-credentials</code>	admin/*****	Username with password	

Icon: S M L

# Nexus Configuration

## 1. Initial Setup

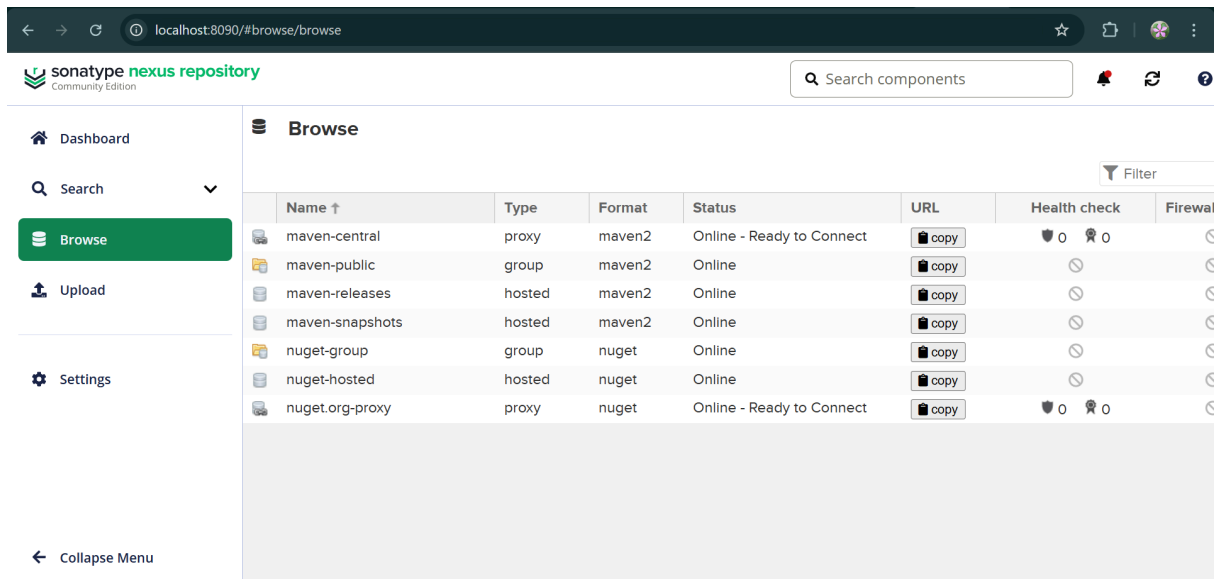
1. Login to Nexus at `http://localhost:8081` or whatever port you are running it on
2. Change default admin password
3. Configure anonymous access (optional)



## 2. Create Maven Repositories

Default repositories should already exist:

- `maven-releases`
- `maven-snapshots`
- `maven-public` (group repository)



### 3. Configure Settings.xml for Maven

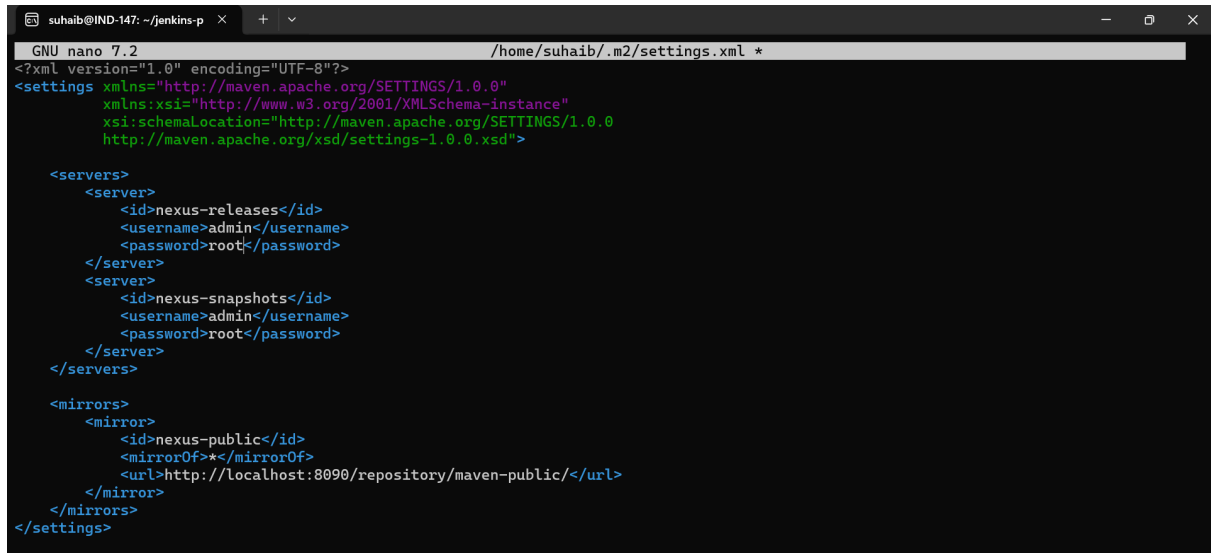
Create `~/.m2/settings.xml` :

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <servers>
    <server>
      <id>nexus-releases</id>
      <username>admin</username>
      <password>root</password>
    </server>
    <server>
      <id>nexus-snapshots</id>
      <username>admin</username>
      <password>root</password>
    </server>
  </servers>

  <mirrors>
    <mirror>
      <id>nexus-public</id>
```

```
</mirrorOf>*</mirrorOf>
<url>http://localhost:8090/repository/maven-public/</url>
</mirror>
</mirrors>
</settings>
```



```
GNU nano 7.2 /home/suhaib/.m2/settings.xml *
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    http://maven.apache.org/xsd/settings-1.0.0.xsd">

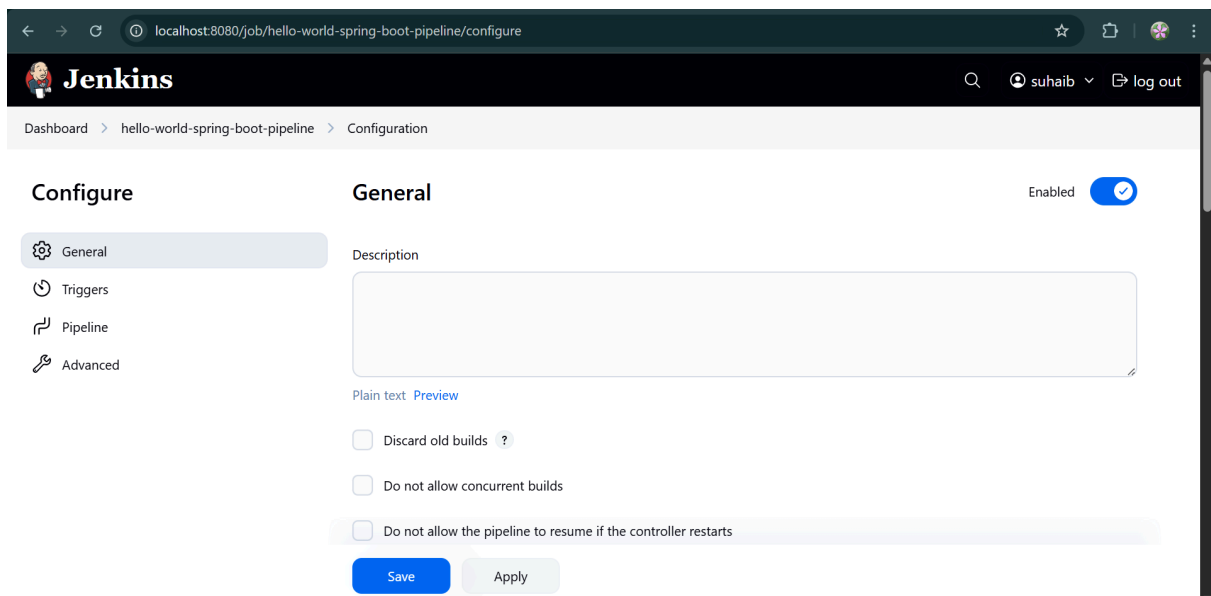
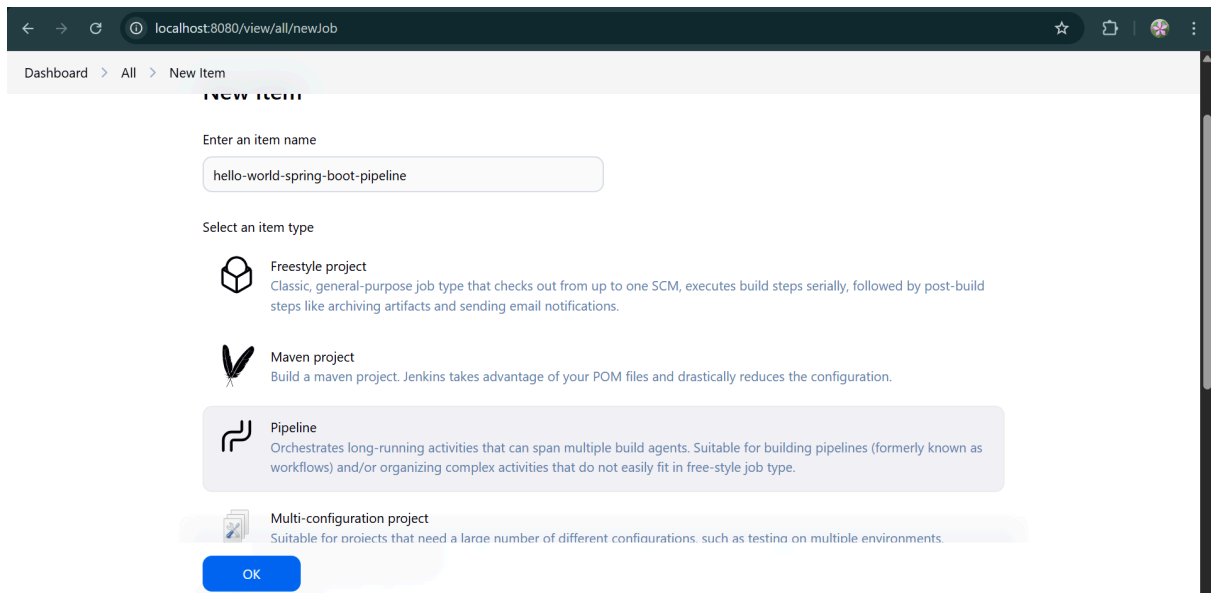
  <servers>
    <server>
      <id>nexus-releases</id>
      <username>admin</username>
      <password>root</password>
    </server>
    <server>
      <id>nexus-snapshots</id>
      <username>admin</username>
      <password>root</password>
    </server>
  </servers>

  <mirrors>
    <mirror>
      <id>nexus-public</id>
      <mirrorOf>*</mirrorOf>
      <url>http://localhost:8090/repository/maven-public/</url>
    </mirror>
  </mirrors>
</settings>
```

## Pipeline Creation

### 1. Create New Pipeline Job

1. Go to Jenkins Dashboard
2. Click **New Item**
3. Enter name: `hello-world-spring-boot-pipeline`
4. Select **Pipeline**
5. Click **OK**

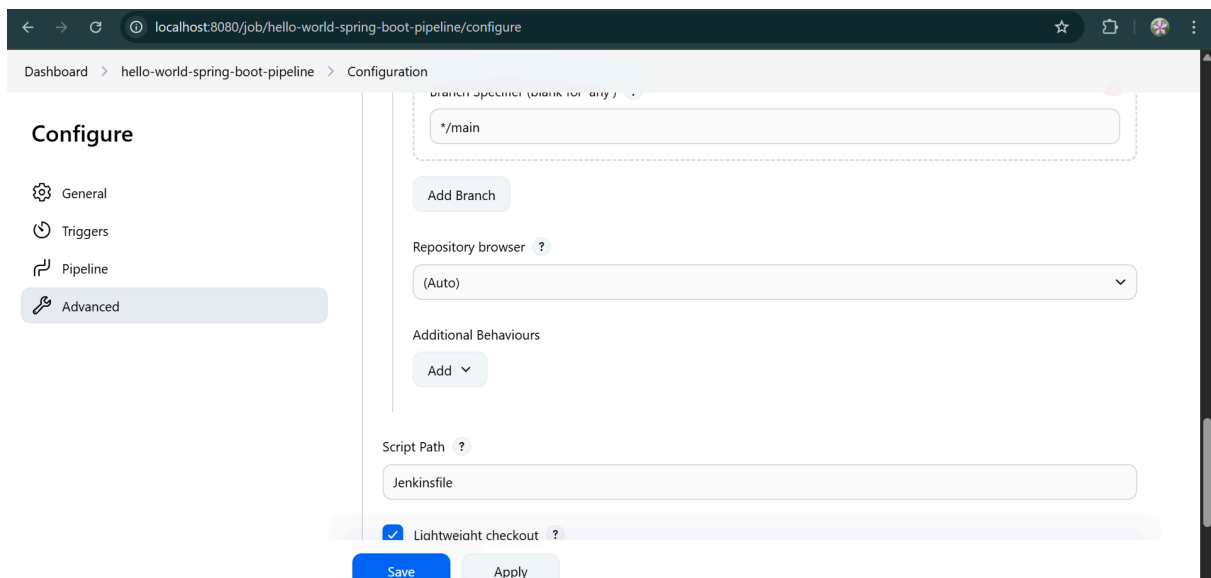
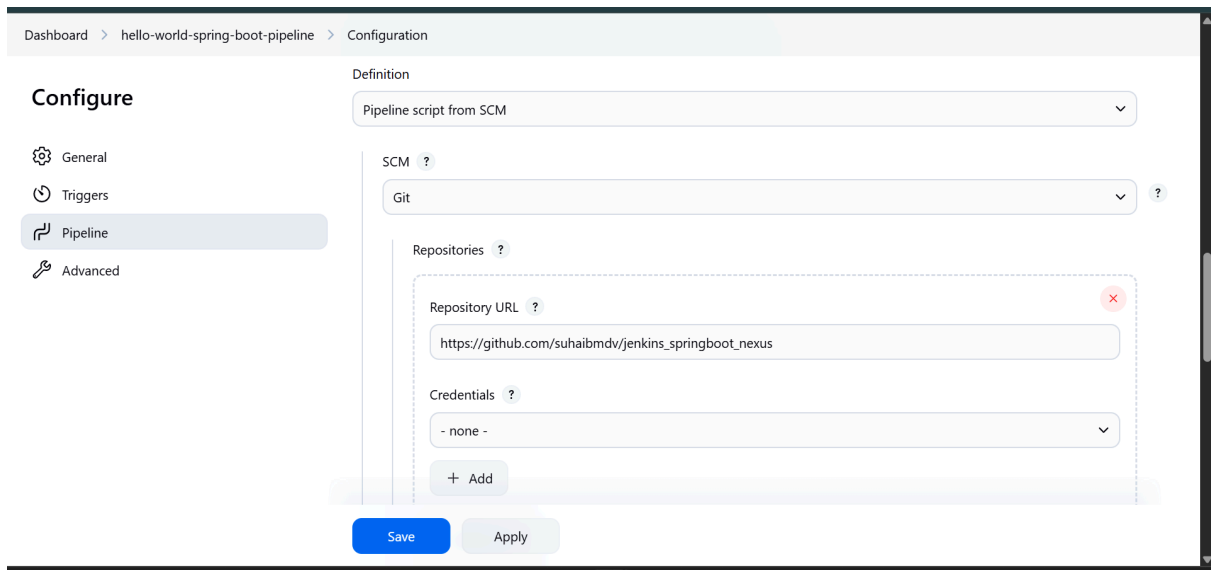


## 2. Configure Pipeline

In the pipeline configuration:

1. Under **Pipeline** section
2. Select **Pipeline script from SCM**
3. Choose **Git** as SCM
4. Enter repository URL :  
`https://github.com/suhaibmdv/jenkins\_springboot\_nexus`
5. Set branch to `/main` or `/master`
6. Script Path: `Jenkinsfile`





### 3. Alternative: Direct Pipeline Script

Or paste the Jenkinsfile content directly in the **Pipeline Script** text area.

## Testing and Execution

### 1. Test Spring Boot Application Locally

```
cd ~/jenkins-projects/hello-world-spring-boot
mvn clean test
mvn spring-boot:run -Dspring-boot.run.arguments=--server.port=8087
```

```

12:43:14.889 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper -- Found @SpringBootConfiguration com.example.helloworldspringboot.HelloWorldApplication for test class com.example.helloworldspringboot.HelloWorldApplicationTest

  ____  _
 / ___|| | | |
| |___| |_| |
 \___ \|  __/
      |_| |_|

:: Spring Boot ::
              (v3.2.0)

2025-06-05T12:43:15.527+05:30 INFO 29763 --- [main] c.e.h.HelloWorldApplicationTest : Starting HelloWorldApplicationTest using Java 17.0.15 with PID 29763 (started by suhaib in /home/suhaib/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot)
2025-06-05T12:43:15.529+05:30 INFO 29763 --- [main] c.e.h.HelloWorldApplicationTest : No active profile set, falling back to 1 default profile: "default"
2025-06-05T12:43:17.192+05:30 INFO 29763 --- [main] c.e.h.HelloWorldApplicationTest : Started HelloWorldApplicationTest in 2.084 seconds (process running for 4.319)
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.115 s - in com.example.helloworldspringboot.HelloWorldApplicationTest
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 01:12 min
[INFO] Finished at: 2025-06-05T12:43:18+05:30
[INFO]
suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$

```

```

[INFO] --- spring-boot-maven-plugin:3.2.0:run (default-cli) @ hello-world-spring-boot ---
[INFO] Attaching agents: []

  ____  _
 / ___|| | | |
| |___| |_| |
 \___ \|  __/
      |_| |_|

:: Spring Boot ::
              (v3.2.0)

2025-06-05T12:46:36.417+05:30 INFO 30138 --- [main] c.e.h.HelloWorldApplication : Starting HelloWorldApplication using Java 17.0.15 with PID 30138 (/home/suhaib/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot/target/classes started by suhaib in /home/suhaib/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot)
2025-06-05T12:46:36.424+05:30 INFO 30138 --- [main] c.e.h.HelloWorldApplication : No active profile set, falling back to 1 default profile: "default"
2025-06-05T12:46:38.420+05:30 INFO 30138 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8087 (http)
2025-06-05T12:46:38.436+05:30 INFO 30138 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-06-05T12:46:38.437+05:30 INFO 30138 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.16]
2025-06-05T12:46:38.615+05:30 INFO 30138 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-06-05T12:46:38.618+05:30 INFO 30138 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2100 ms
2025-06-05T12:46:39.117+05:30 INFO 30138 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8087 (http) with context path ''
2025-06-05T12:46:39.130+05:30 INFO 30138 --- [main] c.e.h.HelloWorldApplication : Started HelloWorldApplication in 3.706 seconds (process running for 4.067)

```

Test endpoints:

```

curl http://localhost:8087/
curl http://localhost:8087/health

```

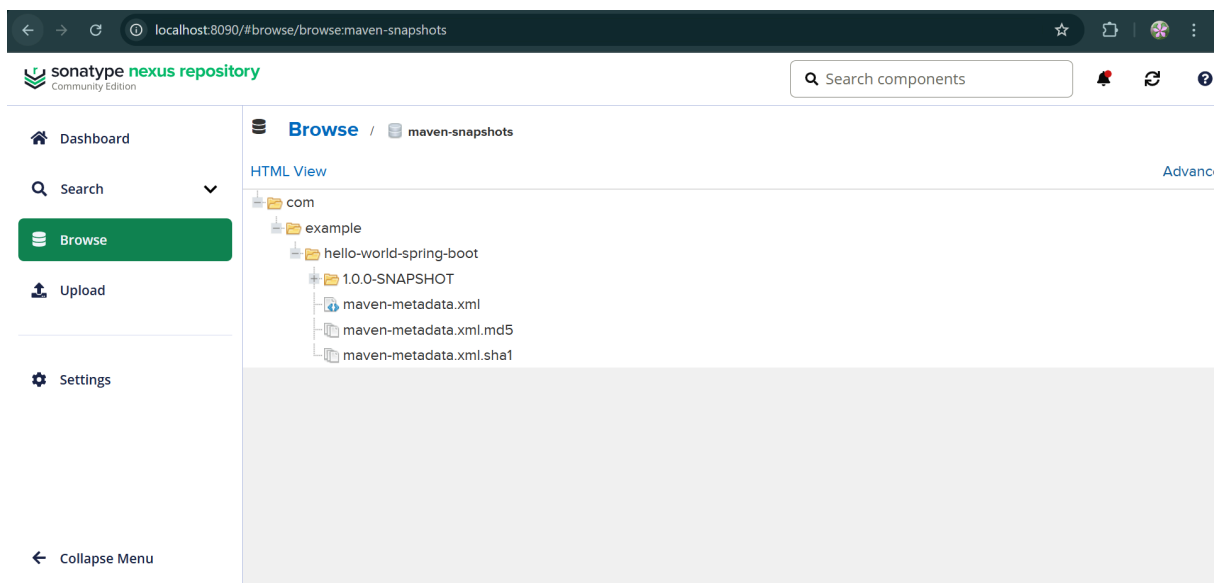
A screenshot of a web browser window. The address bar shows 'localhost:8087'. The page content displays 'Hello World from Spring Boot!'.



## 2. Test Maven Deploy to Nexus

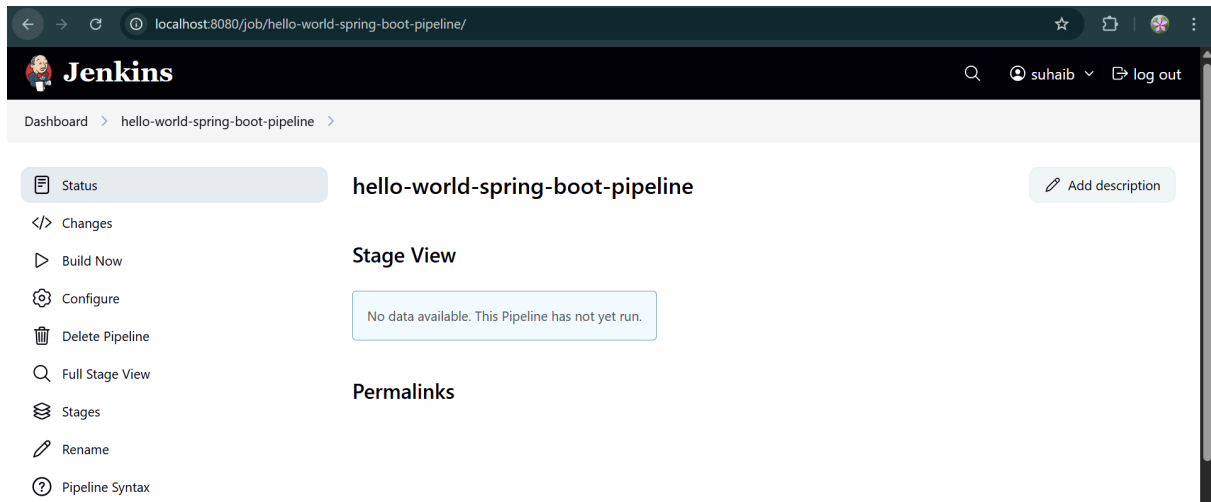
mvn clean deploy

```
.jar
[INFO] --- maven-deploy-plugin:3.1.1:deploy (default-deploy) @ hello-world-spring-boot ---
Downloading from nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/1.0.0-SNAPSHOT/maven-metadata.xml
Uploading to nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/1.0.0-SNAPSHOT/hello-world-spring-boot-1.0.0-20250605.072050-1.pom
Uploaded to nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/1.0.0-SNAPSHOT/hello-world-spring-boot-1.0.0-20250605.072050-1.pom (2.5 kB at 21 kB/s)
Uploading to nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/1.0.0-SNAPSHOT/hello-world-spring-boot-1.0.0-20250605.072050-1.jar
Uploaded to nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/1.0.0-SNAPSHOT/hello-world-spring-boot-1.0.0-20250605.072050-1.jar (20 MB at 24 MB/s)
Downloading from nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/maven-metadata.xml
Uploading to nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/1.0.0-SNAPSHOT/maven-metadata.xml (784 B at 16 kB/s)
Uploading to nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/maven-metadata.xml
Uploaded to nexus-snapshots: http://localhost:8090/repository/maven-snapshots/com/example/hello-world-spring-boot/maven-metadata.xml (294 B at 7.0 kB/s)
[INFO] BUILD SUCCESS
[INFO] Total time: 8.921 s
[INFO] Finished at: 2025-06-05T12:50:58+05:30
[INFO] suhaib@IND-147:~/jenkins-projects/hello-world-spring-boot/hello-world-spring-boot$
```

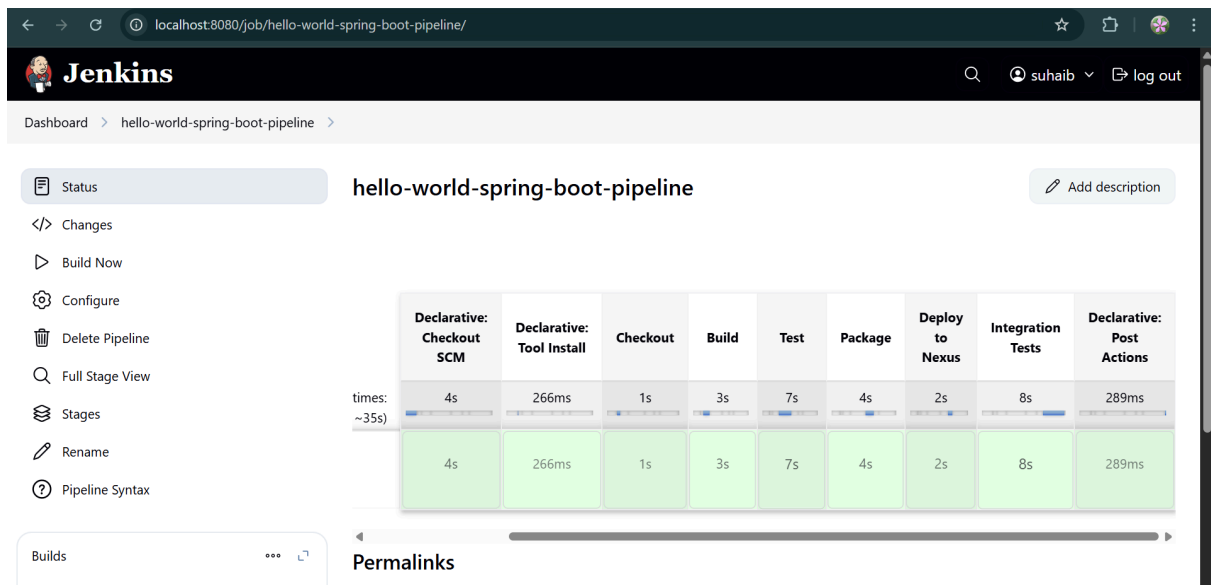


## 3. Run Jenkins Pipeline

1. Go to Jenkins dashboard
2. Click on your pipeline job
3. Click **Build Now**
4. Monitor the build progress in **Console Output**

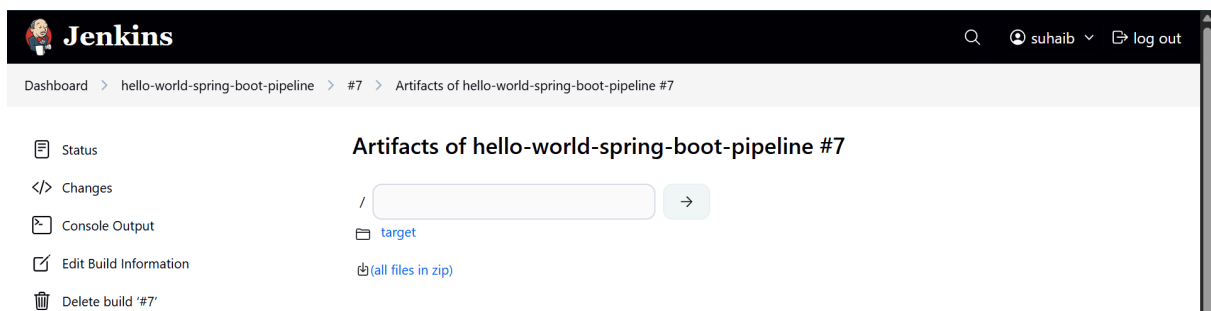
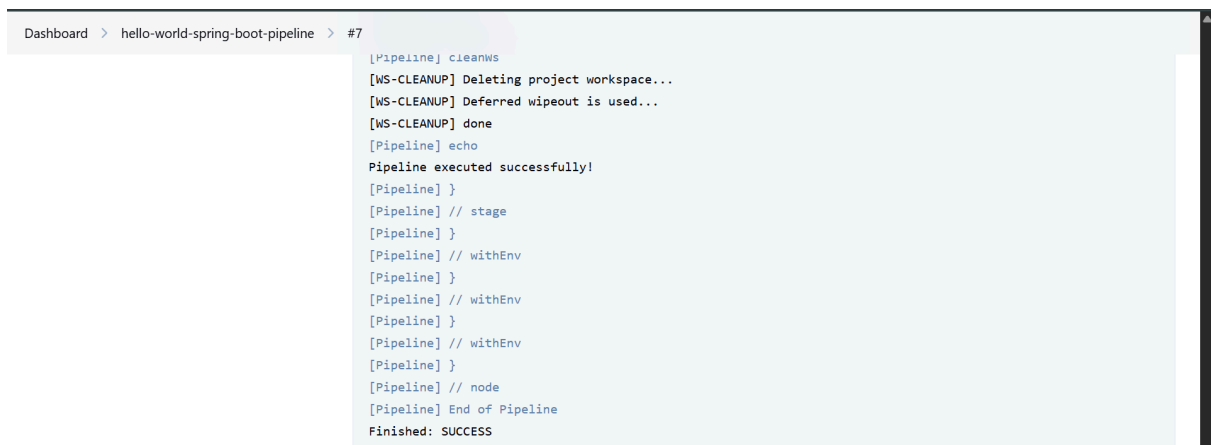
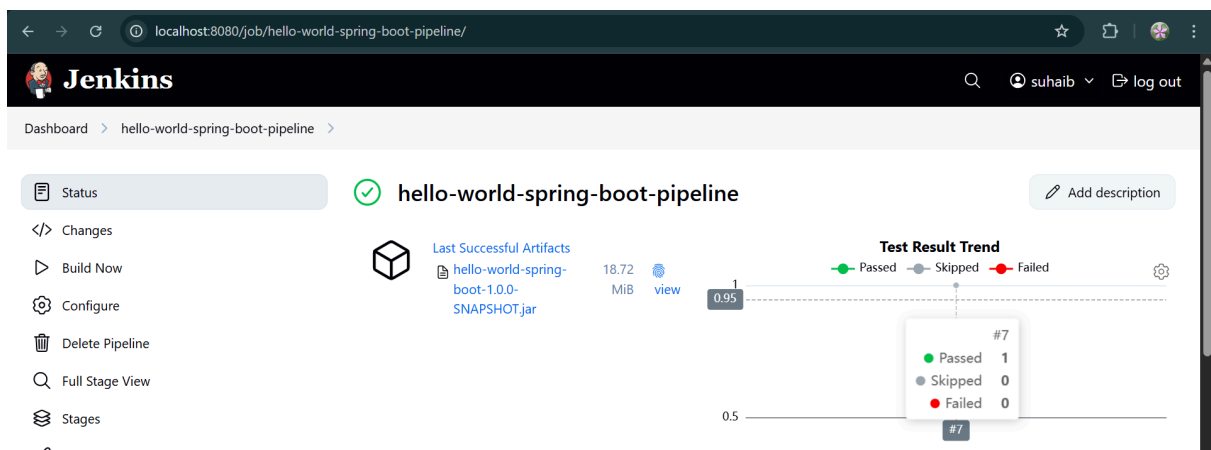
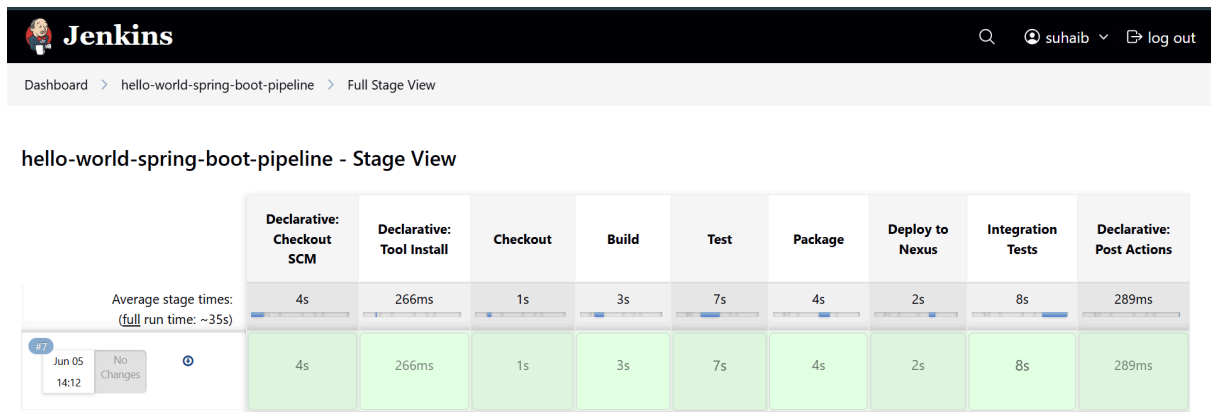


The screenshot shows the Jenkins dashboard for a pipeline named 'hello-world-spring-boot-pipeline'. The left sidebar contains a list of actions: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main area displays the 'Stage View' with a message: 'No data available. This Pipeline has not yet run.'



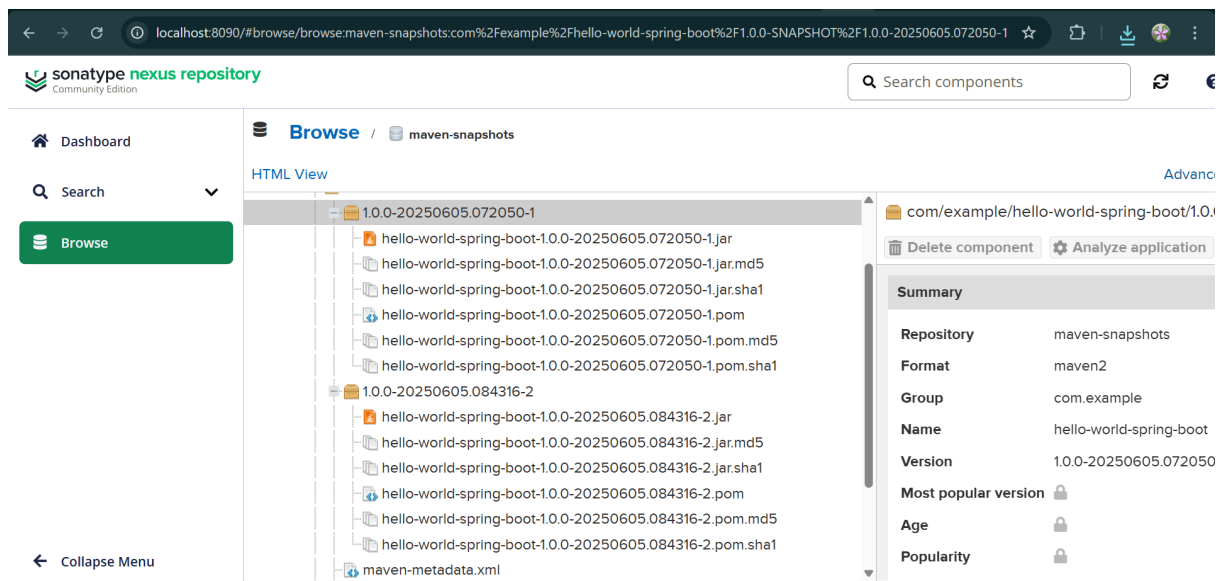
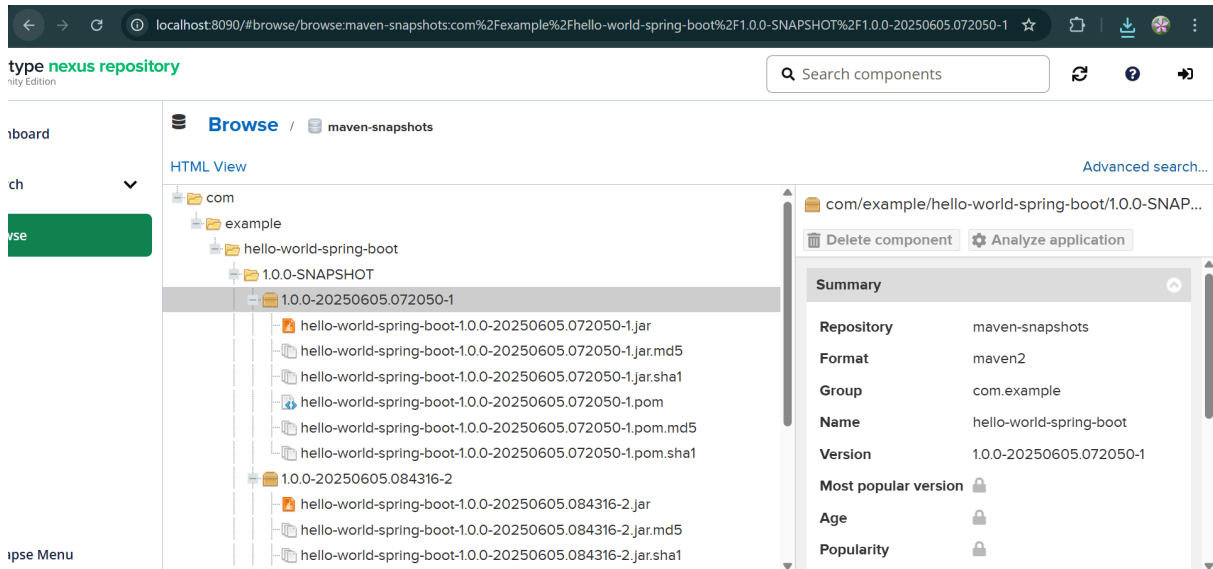
The screenshot shows the Jenkins dashboard for the same pipeline, but now displaying the 'Stages' view. A table shows the build times for each stage. The total time for the build is approximately 35 seconds.

	Declarative: Checkout SCM	Declarative: Tool Install	Checkout	Build	Test	Package	Deploy to Nexus	Integration Tests	Declarative: Post Actions
times:	4s	266ms	1s	3s	7s	4s	2s	8s	289ms
~35s)	4s	266ms	1s	3s	7s	4s	2s	8s	289ms



## 4. Verify Artifacts in Nexus

1. Go to Nexus web interface
2. Browse → maven-snapshots
3. Look for `com/example/hello-world-spring-boot`



## Troubleshooting

### Common Issues and Solutions

#### 1. Jenkins Permission Issues

```
sudo usermod -aG docker jenkins
```

```
sudo systemctl restart jenkins
```

## 2. Maven Not Found

Ensure Maven is properly configured in Jenkins Global Tools.

## 3. Nexus Connection Issues

Check if Nexus is running:

```
sudo systemctl status nexus  
sudo netstat -tlnp | grep 8081
```

## 4. Java Version Issues

Ensure consistent Java versions across tools:

```
update-alternatives --config java
```

## 5. Port Conflicts

If default ports are in use:

- Jenkins: Edit `/etc/default/jenkins` and change `HTTP_PORT`
- Nexus: Edit `/opt/nexus/etc/nexus-default.properties`

## Verification Commands

### Check Services Status

```
sudo systemctl status jenkins  
sudo systemctl status nexus
```

### Check Ports

```
sudo netstat -tlnp | grep :8080 # Jenkins  
sudo netstat -tlnp | grep :8081 # Nexus
```

### Check Logs

```
sudo journalctl -u jenkins -f  
sudo journalctl -u nexus -f
```

## Best Practices

1. **Security:** Use proper credentials management
2. **Backup:** Regular backup of Jenkins and Nexus data
3. **Monitoring:** Set up monitoring for services
4. **Updates:** Keep Jenkins and plugins updated
5. **Testing:** Always test pipeline changes in staging first