

Experiment No. 8

Title: Demonstrate Maven Build Life Cycle

Objective:

The objective of this experiment is to understand and demonstrate the complete **Maven Build Lifecycle**, including:

- Maven installation and project creation.
- Understanding source code structure.
- Executing lifecycle phases (compile, test, package, install, deploy).
- Configuring and deploying artifacts to **Nexus Repository** (Maven Artifactory).
- Emulating real-time Java Developer to DevOps Engineer handover.

Introduction:

Maven is a widely-used build automation and project management tool in the Java ecosystem. It provides a clear and standardized build lifecycle for Java projects, allowing developers to perform various tasks such as compiling code, running tests, packaging applications, and deploying artifacts. This experiment aims to demonstrate the Maven build lifecycle and its different phases.

Key Maven Concepts:

- **Project Object Model (POM):** The POM is an XML file named pom.xml that defines a project's configuration, dependencies, plugins, and goals. It serves as the project's blueprint and is at the core of Maven's functionality.
- **Build Lifecycle:** Maven follows a predefined sequence of phases and goals organized into build lifecycles. These lifecycles include clean, validate, compile, test, package, install, and deploy, among others.
- **Plugin:** Plugins are extensions that provide specific functionality to Maven. They enable tasks like compiling code, running tests, packaging artifacts, and deploying applications.
- **Dependency Management:** Maven simplifies dependency management by allowing developers to declare project dependencies in the POM file. Maven downloads these dependencies from repositories like Maven Central.

- **Repository:** A repository is a collection of artifacts (compiled libraries, JARs, etc.) that Maven uses to manage dependencies. Maven Central is a popular public repository, and organizations often maintain private repositories.

Maven Build Life Cycle:

The Maven build process is organized into a set of build lifecycles, each comprising a sequence of phases. Here are the key build lifecycles and their associated phases:

Clean Lifecycle:

- **clean:** Deletes the target directory, removing all build artifacts.

Default Lifecycle:

- **validate:** Validates the project's structure.
- **compile:** Compiles the project's source code.
- **test:** Runs tests using a suitable testing framework.
- **package:** Packages the compiled code into a distributable format (e.g., JAR, WAR).
- **verify:** Runs checks on the package to verify its correctness.
- **install:** Installs the package to the local repository.
- **deploy:** Copies the final package to a remote repository for sharing.

Site Lifecycle:

- **site:** Generates project documentation.

Each phase within a lifecycle is executed in sequence, and the build progresses from one phase to the next. Developers can customize build behavior by configuring plugins and goals in the POM file.

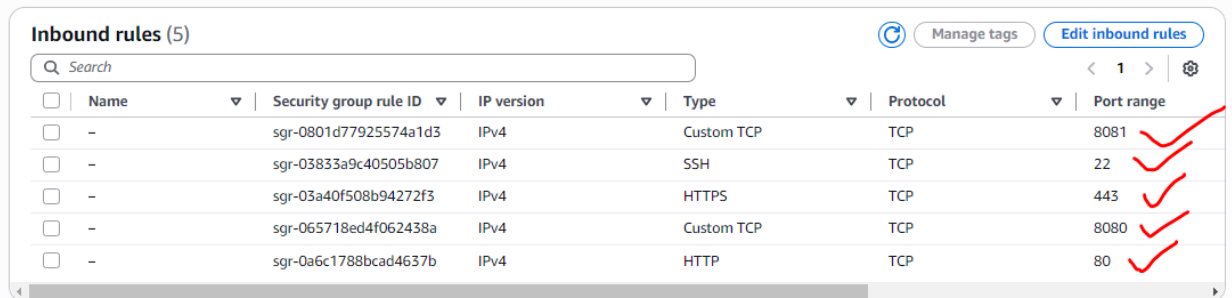
Prerequisites:

- AWS EC2 (Ubuntu 22.04)
- OpenJDK 11
- Apache Maven
- Nexus Repository Manager 3 (Artifactory)
- Spring Boot Web Application

Experiment Steps:

Step 1: Launch and Prepare EC2 Instance

1. Go to AWS Console → EC2 → Launch Instance
2. Choose:
 - **AMI:** Ubuntu Server 22.04
 - **Instance Type:** t2.medium or t2.large
 - **Storage:** 20 GB
 - **Key Pair:** Use existing or create a new one
 - **Security Group:**
 - Allow **SSH (22)** – for remote access
 - Allow **HTTP (80)** – for web app access
 - Allow **Custom TCP (8080, 8081)** – for Spring Boot and Nexus



The screenshot shows the 'Inbound rules (5)' section of an AWS Security Group. It contains a table with five rules, each with a checkbox, a name, a security group rule ID, IP version, type, protocol, and port range. Red checkmarks are placed to the right of each rule's port range.

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0801d77925574a1d3	IPv4	Custom TCP	TCP	8081 ✓
<input type="checkbox"/>	-	sgr-03833a9c40505b807	IPv4	SSH	TCP	22 ✓
<input type="checkbox"/>	-	sgr-03a40f508b94272f3	IPv4	HTTPS	TCP	443 ✓
<input type="checkbox"/>	-	sgr-065718ed4f062438a	IPv4	Custom TCP	TCP	8080 ✓
<input type="checkbox"/>	-	sgr-0a6c1788bcad4637b	IPv4	HTTP	TCP	80 ✓

3. Launch the instance

4. SSH into EC2:

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

```
ls -la
```

```
ubuntu@ip-172-31-33-44:~$ ls -la
total 28
drwxr-x--- 4 ubuntu ubuntu 4096 Jun  9 03:52 .
drwxr-xr-x 3 root    root   4096 Jun  9 03:50 ..
-rw-r--r-- 1 ubuntu ubuntu  220 Mar 31  2024 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31  2024 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Jun  9 03:52 .cache
-rw-r--r-- 1 ubuntu ubuntu  807 Mar 31  2024 .profile
drwx----- 2 ubuntu ubuntu 4096 Jun  9 03:50 .ssh
ubuntu@ip-172-31-33-44:~$ |
```

Step 2: Install Java and Maven

```
sudo apt update && sudo apt install -y openjdk-11-jdk maven git
```

```
java -version
```

OR

```
java --version
```

```
ubuntu@ip-172-31-33-44:~$ java -version
openjdk version "11.0.27" 2025-04-15
OpenJDK Runtime Environment (build 11.0.27+6-post-Ubuntu-0ubuntu124.04)
OpenJDK 64-Bit Server VM (build 11.0.27+6-post-Ubuntu-0ubuntu124.04, mixed mode, sharing)
ubuntu@ip-172-31-33-44:~$
ubuntu@ip-172-31-33-44:~$
ubuntu@ip-172-31-33-44:~$ java --version
openjdk 11.0.27 2025-04-15
OpenJDK Runtime Environment (build 11.0.27+6-post-Ubuntu-0ubuntu124.04)
OpenJDK 64-Bit Server VM (build 11.0.27+6-post-Ubuntu-0ubuntu124.04, mixed mode, sharing)
ubuntu@ip-172-31-33-44:~$
```

```
mvn -version
```

OR

```
mvn --version
```

OR

```
mvn -v
```

```

ubuntu@ip-172-31-33-44:~$ mvn -version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 11.0.27, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1029-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-33-44:~$
ubuntu@ip-172-31-33-44:~$ mvn --version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 11.0.27, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1029-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-33-44:~$
ubuntu@ip-172-31-33-44:~$
ubuntu@ip-172-31-33-44:~$ mvn -v
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 11.0.27, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1029-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-33-44:~$ |

```

Step 3: Install and Configure Nexus Repository

1. Create a user for Nexus:

```
sudo useradd -m -s /bin/bash nexus
```

```
sudo su - nexus
```

```

ubuntu@ip-172-31-33-44:~$ sudo useradd -m -s /bin/bash nexus
ubuntu@ip-172-31-33-44:~$ sudo su - nexus
nexus@ip-172-31-33-44:~$ |

```

```
ls -la
```

```

nexus@ip-172-31-33-44:~$ ls -la
total 20
drwxr-x--- 2 nexus nexus 4096 Jun  9 04:00 .
drwxr-xr-x 4 root  root  4096 Jun  9 04:00 ..
-rw-r--r-- 1 nexus nexus  220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 nexus nexus 3771 Mar 31 2024 .bashrc
-rw-r--r-- 1 nexus nexus  807 Mar 31 2024 .profile
nexus@ip-172-31-33-44:~$ |

```

2. Download and install Nexus:

```
wget https://download.sonatype.com/nexus/3/nexus-3.80.0-06-  
linux-x86_64.tar.gz
```

```
nexus@ip-172-31-33-44:~$ wget https://download.sonatype.com/nexus/3/nexus-3.80.0-06-linux-x86_64.tar.gz  
--2025-06-09 04:07:03-- https://download.sonatype.com/nexus/3/nexus-3.80.0-06-linux-x86_64.tar.gz  
Resolving download.sonatype.com (download.sonatype.com)... 52.8.65.131, 54.215.207.7  
Connecting to download.sonatype.com (download.sonatype.com)|52.8.65.131|:443... connected.  
HTTP request sent, awaiting response... 302 Moved Temporarily  
Location: https://sonatype-download.global.ssl.fastly.net/repository/downloads-prod-group/3/nexus-3.80.0-06-linux-x86_64.tar.gz [following]  
--2025-06-09 04:07:04-- https://sonatype-download.global.ssl.fastly.net/repository/downloads-prod-group/3/nexus-3.80.0-06-linux-x86_64.tar.gz  
Resolving sonatype-download.global.ssl.fastly.net (sonatype-download.global.ssl.fastly.net)... 151.101.153.194  
Connecting to sonatype-download.global.ssl.fastly.net (sonatype-download.global.ssl.fastly.net)|151.101.153.194|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 426428974 (407M) [application/x-gzip]  
Saving to: 'nexus-3.80.0-06-linux-x86_64.tar.gz'  
  
nexus-3.80.0-06-linux-x86_64.tar. 100%[=====] 406.67M 16.5MB/s in 24s  
2025-06-09 04:07:28 (17.0 MB/s) - 'nexus-3.80.0-06-linux-x86_64.tar.gz' saved [426428974/426428974]  
nexus@ip-172-31-33-44:~$
```

```
ls -la
```

```
nexus@ip-172-31-33-44:~$ ls -la  
total 416460  
drwxr-x--- 2 nexus nexus 4096 Jun 9 04:07 .  
drwxr-xr-x 4 root root 4096 Jun 9 04:00 ..  
-rw-r--r-- 1 nexus nexus 220 Mar 31 2024 .bash_logout  
-rw-r--r-- 1 nexus nexus 3771 Mar 31 2024 .bashrc  
-rw-r--r-- 1 nexus nexus 807 Mar 31 2024 .profile  
-rw-rw-r-- 1 nexus nexus 194 Jun 9 04:07 .wget-hsts  
-rw-rw-r-- 1 nexus nexus 426428974 May 6 16:15 nexus-3.80.0-06-linux-x86_64.tar.gz  
nexus@ip-172-31-33-44:~$
```

```
tar -xvzf nexus-3.80.0-06-linux-x86_64.tar.gz
```

```
ls -la
```

```
nexus@ip-172-31-33-44:~$ ls -la  
total 416468  
drwxr-x--- 4 nexus nexus 4096 Jun 9 04:10 .  
drwxr-xr-x 4 root root 4096 Jun 9 04:00 ..  
-rw-r--r-- 1 nexus nexus 220 Mar 31 2024 .bash_logout  
-rw-r--r-- 1 nexus nexus 3771 Mar 31 2024 .bashrc  
-rw-r--r-- 1 nexus nexus 807 Mar 31 2024 .profile  
-rw-rw-r-- 1 nexus nexus 194 Jun 9 04:07 .wget-hsts  
drwxr-xr-x 6 nexus nexus 4096 Jun 9 04:10 nexus-3.80.0-06  
-rw-rw-r-- 1 nexus nexus 426428974 May 6 16:15 nexus-3.80.0-06-linux-x86_64.tar.gz  
drwxr-xr-x 3 nexus nexus 4096 May 2 20:55 sonatype-work  
nexus@ip-172-31-33-44:~$
```

```
mv nexus-3.80.0-06 nexus
```

```
#Rename
```

```
nexus@ip-172-31-33-44:~$ mv nexus-3.80.0-06 nexus
nexus@ip-172-31-33-44:~$ ls -la
total 416468
drwxr-x--- 4 nexus nexus      4096 Jun  9 04:11 .
drwxr-xr-x 4 root  root      4096 Jun  9 04:00 ..
-rw-r--r-- 1 nexus nexus      220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 nexus nexus    3771 Mar 31 2024 .bashrc
-rw-r--r-- 1 nexus nexus      807 Mar 31 2024 .profile
-rw-rw-r-- 1 nexus nexus      194 Jun  9 04:07 .wget-hsts
drwxr-xr-x 6 nexus nexus      4096 Jun  9 04:10 nexus ✓
-rw-rw-r-- 1 nexus nexus 426428974 May  6 16:15 nexus-3.80.0-06-linux-x86_64.tar.gz
drwxr-xr-x 3 nexus nexus      4096 May  2 20:55 sonatype-work
nexus@ip-172-31-33-44:~$ |
```

3. Create systemd service for Nexus:

Exit to root:

exit

```
nexus@ip-172-31-33-44:~$ exit
logout
ubuntu@ip-172-31-33-44:~$ |
```

Then:

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

Paste the following:

```
[Unit]
Description=Nexus Repository
After=network.target

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

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

```
[Unit]
Description=Nexus Repository
After=network.target

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

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

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

```
ubuntu@ip-172-31-33-44:~$ sudo cat /etc/systemd/system/nexus.service
[Unit]
Description=Nexus Repository
After=network.target

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

[Install]
WantedBy=multi-user.target

ubuntu@ip-172-31-33-44:~$ |
```

4. Start Nexus:

```
sudo chown -R nexus:nexus /home/nexus
```

```
sudo systemctl daemon-reexec
```

```
sudo systemctl status nexus
```



```

ubuntu@ip-172-31-33-44:~$ sudo chown -R nexus:nexus /home/nexus
ubuntu@ip-172-31-33-44:~$ sudo systemctl daemon-reexec
ubuntu@ip-172-31-33-44:~$ sudo systemctl status nexus
○ nexus.service - Nexus Repository
   Loaded: loaded (/etc/systemd/system/nexus.service; disabled; preset: enabled)
   Active: inactive (dead)
ubuntu@ip-172-31-33-44:~$ |

```

sudo systemctl enable nexus

```

ubuntu@ip-172-31-33-44:~$ sudo systemctl enable nexus
Created symlink /etc/systemd/system/multi-user.target.wants/nexus.service → /etc/systemd/system/nexus.service.

```

sudo systemctl status nexus

```

ubuntu@ip-172-31-33-44:~$ sudo systemctl status nexus
○ nexus.service - Nexus Repository
   Loaded: loaded (/etc/systemd/system/nexus.service; enabled; preset: enabled)
   Active: inactive (dead)
ubuntu@ip-172-31-33-44:~$ |

```

sudo systemctl start nexus

sudo systemctl status nexus

```

ubuntu@ip-172-31-33-44:~$ sudo systemctl start nexus
ubuntu@ip-172-31-33-44:~$ sudo systemctl status nexus
● nexus.service - Nexus Repository
   Loaded: loaded (/etc/systemd/system/nexus.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-06-09 04:17:20 UTC; 4s ago
     Process: 3961 ExecStart=/home/nexus/nexus/bin/nexus start (code=exited, status=0/SUCCESS)
    Main PID: 4202 (java)
      Tasks: 23 (limit: 9501)
     Memory: 312.5M (peak: 312.9M)
        CPU: 8.500s
    CGroup: /system.slice/nexus.service
            └─4202 /home/nexus/nexus/jdk/temurin_17.0.13_11_linux_x86_64/jdk-17.0.13+11/bin/java -server -Dnexus.installer.type=linux
Jun 09 04:17:20 ip-172-31-33-44 systemd[1]: Starting nexus.service - Nexus Repository...
Jun 09 04:17:20 ip-172-31-33-44 nexus[3961]: /home/nexus/nexus/bin/nexus: 155: [: not found
Jun 09 04:17:20 ip-172-31-33-44 nexus[3961]: Starting nexus
Jun 09 04:17:20 ip-172-31-33-44 systemd[1]: Started nexus.service - Nexus Repository.
lines 1-15/15 (END)

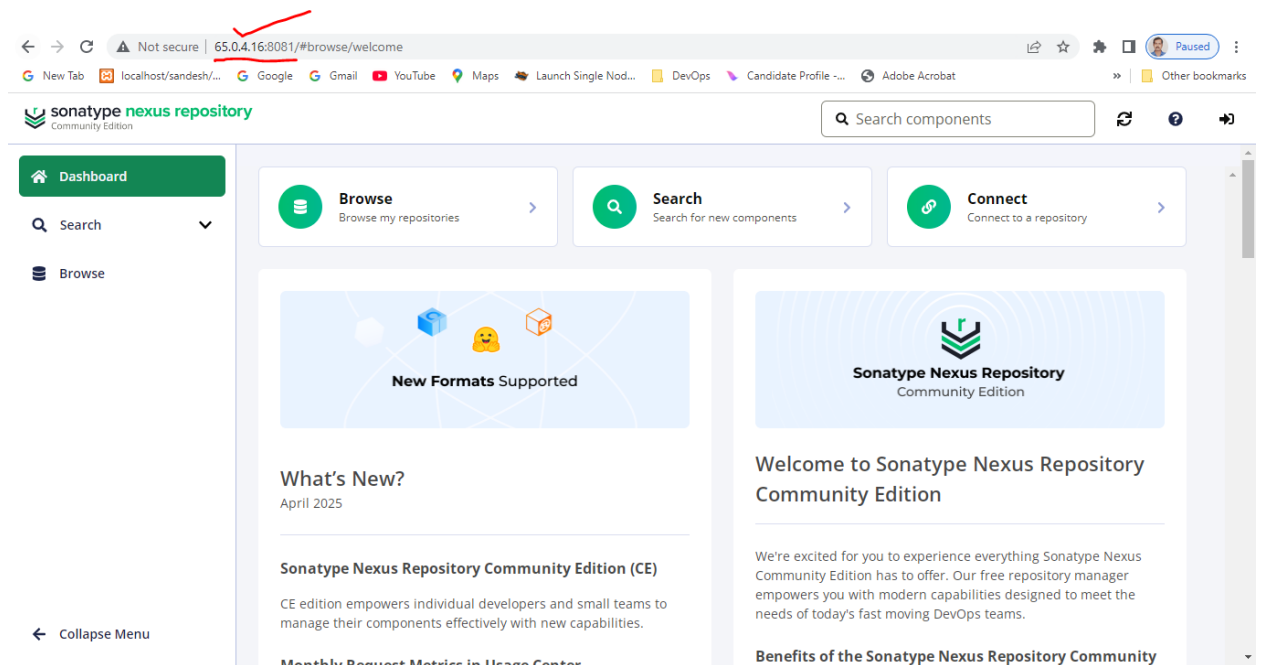
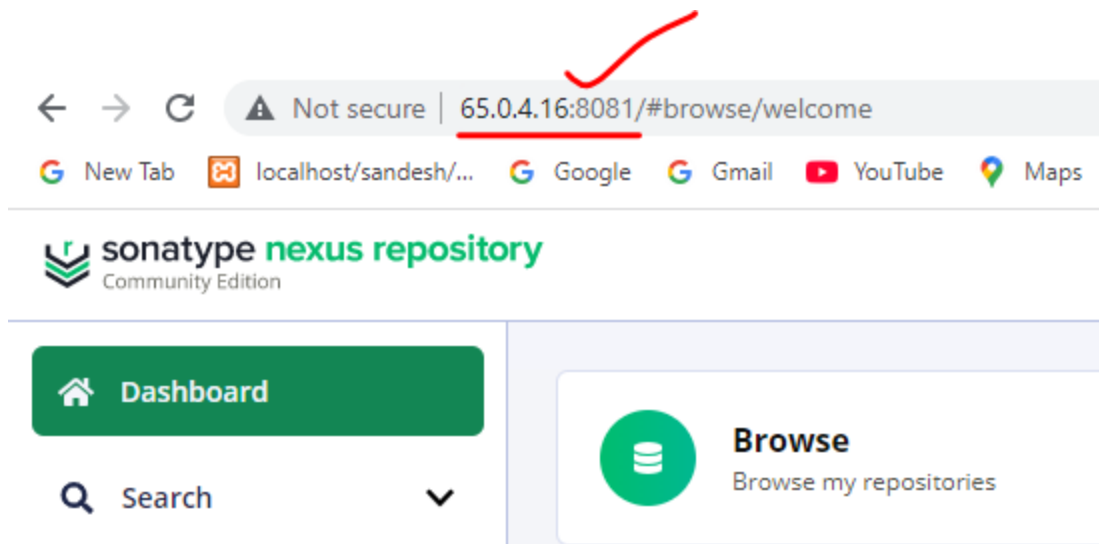
```

Press Ctrl+C to exit.

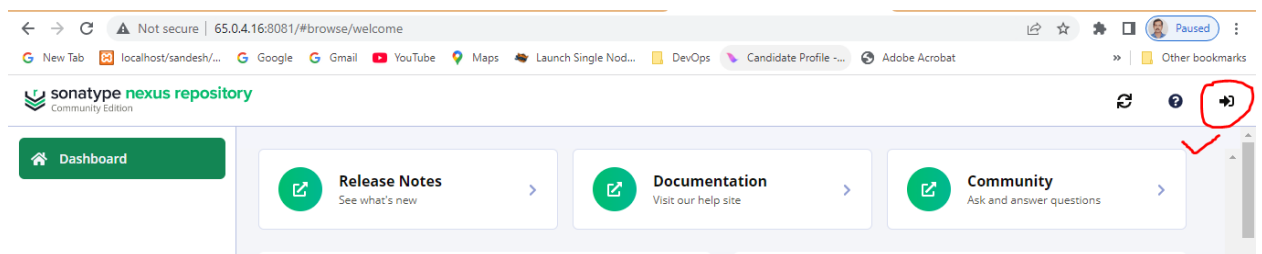
5. Access Nexus:

Visit in browser:

<http://<EC2-PUBLIC-IP>:8081>



Click on **Login**:



User name: admin

Get the password by running the below command:

```
sudo cat /home/nexus/sonatype-work/nexus3/admin.password
```

```
ubuntu@ip-172-31-33-44:~$ sudo cat /home/nexus/sonatype-work/nexus3/admin.password
[REDACTED]
ubuntu@ip-172-31-33-44:~$
```

Sign In

Sign in

Cancel

Then change the password.

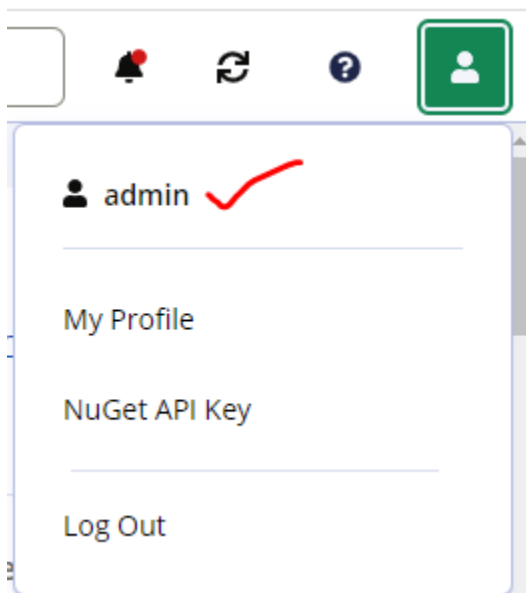
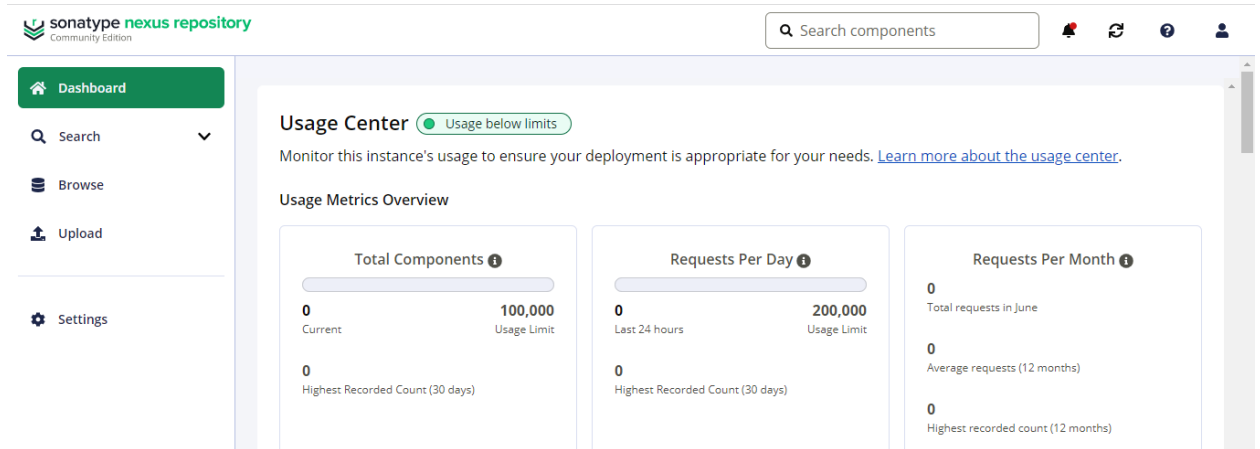
Please choose a password for the admin user2 of 6

New password:

Confirm password:

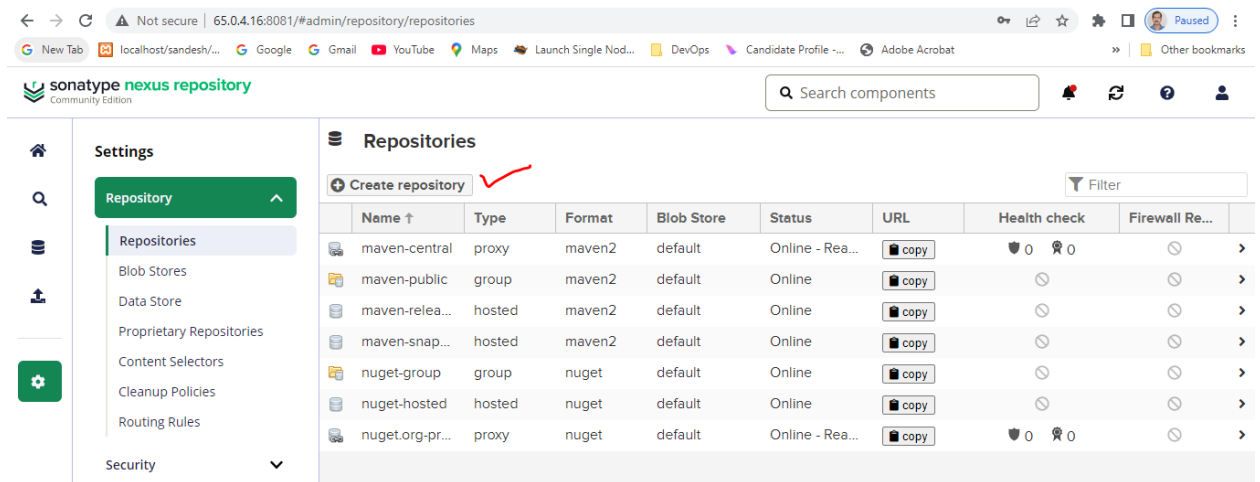
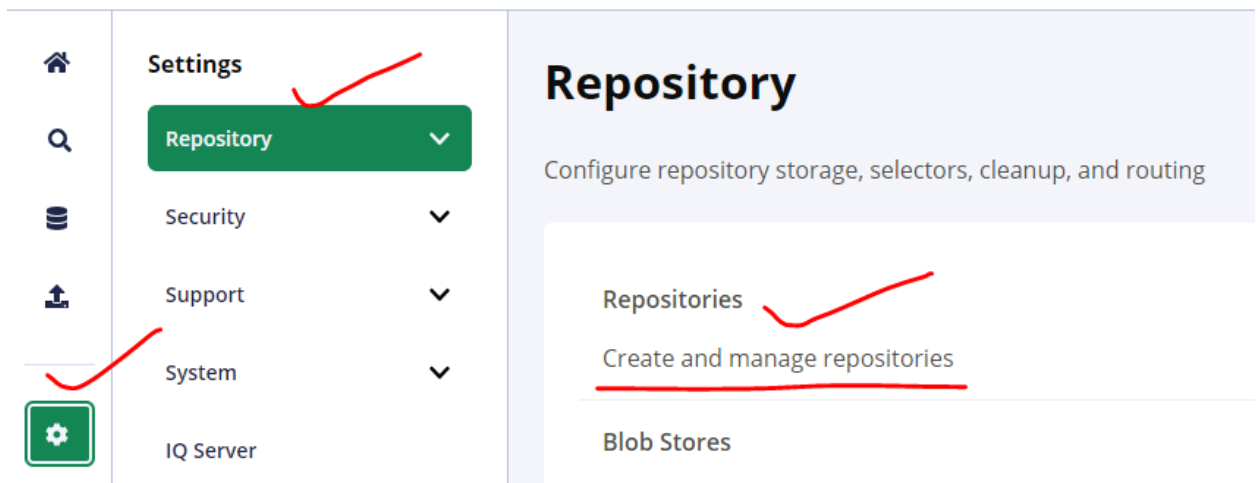
Back

Next
























6. Create a Maven Hosted Repository:

Settings -> Repository -> Create and manage repositories -> Create repository







- Recipe: maven2 (hosted)

 **Repositories** /  **Select Recipe**

Recipe ↑ 
 docker (group)
 docker (hosted)
 docker (proxy)
 gitlfs (hosted)
 go (group)
 go (proxy)
 helm (hosted)
 helm (proxy)
 huggingface (proxy)
 maven2 (group)
 maven2 (hosted) 
 maven2 (proxy)
 npm (group)
 npm (hosted)
 npm (proxy)
 nuget (group)
 nuget (hosted)

- Name: sdm-maven-releases

 **Repositories** /  **Select Recipe** /  **Create Repository: maven2 (hosted)**

Name: A unique identifier for this repository
 

Online: ☒ If checked, the repository accepts incoming requests


- Version Policy: Release

Maven 2

Version policy:

What type of artifacts does this repository store?

Release



- Deployment Policy: Allow Redeploy

Hosted

Deployment policy:

Controls if deployments of and updates to artifacts are allowed

Allow redeploy



Proprietary Components:

☐ Components in this repository count as proprietary for namespace conflict attacks (requires Sonatype Nexus Firewall)

- Create repository



Repositories

[Select Recipe](#)[Create Repository: maven2 \(hosted\)](#)

Proprietary Components:

☐ Components in this repository count as proprietary for namespace conflict attacks (requires Sonatype Nexus Firewall)

Cleanup

Cleanup Policies:

Components that match any of the Applied policies will be deleted

Available

Filter



Applied

[Create repository](#)

[Cancel](#)



Repositories

[+ Create repository](#)

Filter

	Name ↑	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...	
	maven-central	proxy	maven2	default	Online - R...	copy	0 0		
	maven-public	group	maven2	default	Online	copy			
	maven-releases	hosted	maven2	default	Online	copy			
	maven-snapshots	hosted	maven2	default	Online	copy			
	nuget-group	group	nuget	default	Online	copy			
	nuget-hosted	hosted	nuget	default	Online	copy			
	nuget.org-proxy	proxy	nuget	default	Online - R...	copy	0 0		
	<u>sdm-maven-releases</u>	<u>hosted</u>	<u>maven2</u>	default	Online	copy			

Step 4: Create Spring Boot Web Application


```
mvn archetype:generate \
  -DgroupId=com.example \
  -DartifactId=SpringBootApplication \
  -DarchetypeArtifactId=maven-archetype-quickstart \
  -DinteractiveMode=false
```

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.jar (4.3 kB at 614 kB/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/ubuntu
[INFO] Parameter: package, Value: com.example
[INFO] Parameter: groupId, Value: com.example
[INFO] Parameter: artifactId, Value: SpringBootApplication
[INFO] Parameter: packageName, Value: com.example
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /home/ubuntu/SpringBootApplication
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.025 s
[INFO] Finished at: 2025-06-09T04:46:52Z
[INFO] -----
ubuntu@ip-172-31-33-44:~$
```

```
ls -la
```

```
ubuntu@ip-172-31-33-44:~$ ls -la
total 36
drwxr-x--- 6 ubuntu ubuntu 4096 Jun  9 04:46 .
drwxr-xr-x 4 root    root   4096 Jun  9 04:00 ..
-rw-r--r-- 1 ubuntu ubuntu  220 Mar 31  2024 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31  2024 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Jun  9 03:52 .cache
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:46 .m2
-rw-r--r-- 1 ubuntu ubuntu  807 Mar 31  2024 .profile
drwx----- 2 ubuntu ubuntu 4096 Jun  9 03:50 .ssh
-rw-r--r-- 1 ubuntu ubuntu    0 Jun  9 03:55 .sudo_as_admin_successful
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:46 SpringBootApplication ✓
ubuntu@ip-172-31-33-44:~$
```

```
cd SpringBootApplication
```

```
ls -la
```

```
ubuntu@ip-172-31-33-44:~$ cd SpringBootApplication
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ ls -la
total 16
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:46 .
drwxr-x--- 6 ubuntu ubuntu 4096 Jun  9 04:46 .. ✓
-rw-rw-r-- 1 ubuntu ubuntu  650 Jun  9 04:46 pom.xml
drwxrwxr-x 4 ubuntu ubuntu 4096 Jun  9 04:46 src ✓
ubuntu@ip-172-31-33-44:~/SpringBootApplication$
```

```
rm -rf src/ # Removing src to create a real-time SpringBoot App
```

```
ls -la
```

```
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ rm -rf src/
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ ls -la
total 12
drwxrwxr-x 2 ubuntu ubuntu 4096 Jun  9 04:50 .
drwxr-x--- 6 ubuntu ubuntu 4096 Jun  9 04:46 ..
-rw-rw-r-- 1 ubuntu ubuntu  650 Jun  9 04:46 pom.xml
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ |
```

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

```
ls -la
```

```
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ mkdir -p src/main/java/com/example
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ ls -la
total 16
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:51 .
drwxr-x--- 6 ubuntu ubuntu 4096 Jun  9 04:46 ..
-rw-rw-r-- 1 ubuntu ubuntu  650 Jun  9 04:46 pom.xml
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:51 src ✓
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ |
```

```
vi src/main/java/com/example/App.java
```

Paste this:

```
package com.example;

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

@SpringBootApplication
@RestController
public class App {

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

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

```

package com.example;

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

@SpringBootApplication
@RestController
public class App {

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

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

```

cat src/main/java/com/example/App.java

```

ubuntu@ip-172-31-33-44:~/SpringBootTest$ cat src/main/java/com/example/App.java
package com.example;

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

@SpringBootApplication
@RestController
public class App {

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

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

ubuntu@ip-172-31-33-44:~/SpringBootTest$ |

```

Step 5: Replace pom.xml with Spring Boot Configuration

Edit pom.xml:

```
<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>
  <groupId>com.example</groupId>
  <artifactId>SpringBootApplication</artifactId>
  <version>1.0.0</version>
  <packaging>jar</packaging>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.5</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>

  <distributionManagement>
    <repository>
      <id>nexus</id>
      <name>Nexus Release Repository</name>
      <url>http://<EC2-PUBLIC-IP>:8081/repository/sdm-maven-releases/</url>
    </repository>
  </distributionManagement>
</project>
```

Step 6: Add Nexus Credentials to Maven Settings

```
vi ~/.m2/settings.xml
```

Paste:

```
<settings>
  <servers>
    <server>
      <id>nexus</id>
      <username>admin</username>
      <password>YourNexusPassword</password>
    </server>
  </servers>
</settings>
```

Step 7: Build, Run, and Deploy the App

1. Build the app:

```
mvn clean package
```

```
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/28.2-android/guava-28.2-android.jar (2.6 MB at 5.3 MB/s)
[INFO] Replacing main artifact with repackaged archive
[INFO] BUILD SUCCESS
[INFO] Total time: 9.004 s
[INFO] Finished at: 2025-06-09T04:59:53Z
[INFO]
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ |
```

```
ls -la
```

```
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ ls -la
total 20
drwxrwxr-x 4 ubuntu ubuntu 4096 Jun  9 04:59 .
drwxr-x--- 6 ubuntu ubuntu 4096 Jun  9 04:58 ..
-rw-rw-r-- 1 ubuntu ubuntu 1280 Jun  9 04:58 pom.xml
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:51 src
drwxrwxr-x 6 ubuntu ubuntu 4096 Jun  9 04:59 target ✓
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ |
```

```
ls -la target/
```

```
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ ls -la target/
total 17244
drwxrwxr-x 6 ubuntu ubuntu 4096 Jun  9 04:59 .
drwxrwxr-x 4 ubuntu ubuntu 4096 Jun  9 04:59 ..
-rw-rw-r-- 1 ubuntu ubuntu 17627490 Jun  9 04:59 SpringBootApplication-1.0.0.jar ✓
-rw-rw-r-- 1 ubuntu ubuntu 2437 Jun  9 04:59 SpringBootApplication-1.0.0.jar.original
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:59 classes
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:59 generated-sources
drwxrwxr-x 2 ubuntu ubuntu 4096 Jun  9 04:59 maven-archiver
drwxrwxr-x 3 ubuntu ubuntu 4096 Jun  9 04:59 maven-status
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ |
```

2. Run the app:

```
java -jar target/SpringBootApplication-1.0.0.jar
```

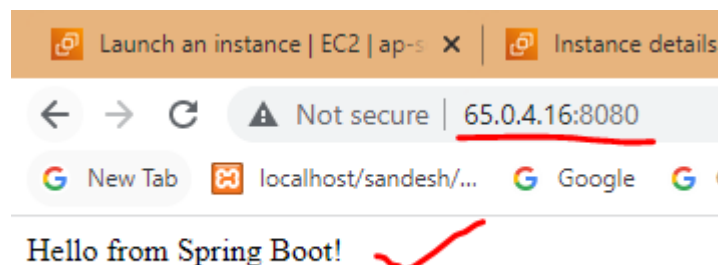
```
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ java -jar target/SpringBootApplication-1.0.0.jar

:: Spring Boot ::
(v2.7.5)

2025-06-09 05:02:43.712 INFO 4564 --- [main] com.example.App : Starting App v1.0.0 using Java 11.0.27 on ip-172-31-33-44 with PID 4564 (/home/ubuntu/SpringBootApplication/target/SpringBootApplication-1.0.0.jar started by ubuntu in /home/ubuntu/SpringBootApplication)
2025-06-09 05:02:43.720 INFO 4564 --- [main] com.example.App : No active profile set, falling back to 1 default profile: "default"
2025-06-09 05:02:45.210 INFO 4564 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2025-06-09 05:02:45.226 INFO 4564 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-06-09 05:02:45.226 INFO 4564 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.68]
2025-06-09 05:02:45.331 INFO 4564 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-06-09 05:02:45.332 INFO 4564 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1494 ms
2025-06-09 05:02:45.802 INFO 4564 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2025-06-09 05:02:45.817 INFO 4564 --- [main] com.example.App : Started App in 2.889 seconds (JVM running for 3.438)
```

3. Access it in Browser:

<http://<EC2-PUBLIC-IP>:8080>



Output:

Hello from Spring Boot!

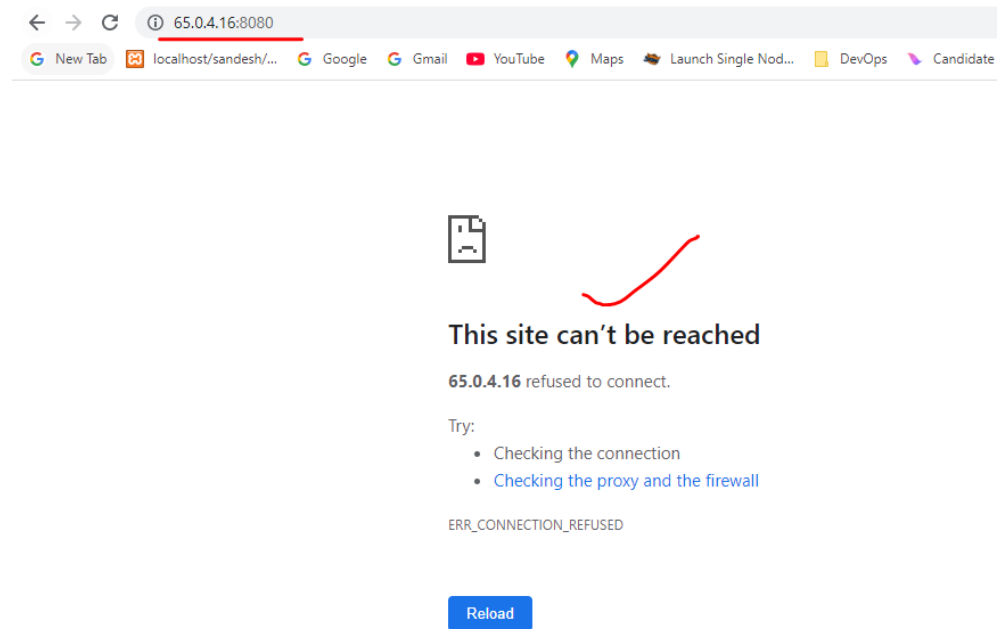
Stop the application by pressing Ctrl+C.

```
ubuntu@ip-172-31-33-44:~/SpringBootApplication$ java -jar target/SpringBootApplication-1.0.0.jar

:: Spring Boot ::
(v2.7.5)

2025-06-09 05:02:43.712 INFO 4564 --- [main] com.example.App : Starting App v1.0.0 using Java 11.0.27 on ip-172-31-33-44 with PID 4564 (/home/ubuntu/SpringBootApplication/target/SpringBootApplication-1.0.0.jar started by ubuntu in /home/ubuntu/SpringBootApplication)
2025-06-09 05:02:43.720 INFO 4564 --- [main] com.example.App : No active profile set, falling back to 1 default profile: "default"
2025-06-09 05:02:45.210 INFO 4564 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2025-06-09 05:02:45.226 INFO 4564 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-06-09 05:02:45.226 INFO 4564 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.68]
2025-06-09 05:02:45.331 INFO 4564 --- [main] o.a.c.c.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-06-09 05:02:45.332 INFO 4564 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1494 ms
2025-06-09 05:02:45.802 INFO 4564 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2025-06-09 05:02:45.817 INFO 4564 --- [main] com.example.App : Started App in 2.889 seconds (JVM running for 3.438)
2025-06-09 05:03:22.194 INFO 4564 --- [nio-8080-exec-2] o.a.c.c.c.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-06-09 05:03:22.195 INFO 4564 --- [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-06-09 05:03:22.196 INFO 4564 --- [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
ubuntu@ip-172-31-33-44:~/SpringBootApplication$
```

Go back to browser and confirm that the application is not accessible.



4. Deploy the artifact to Nexus:

```
mvn deploy
```

```

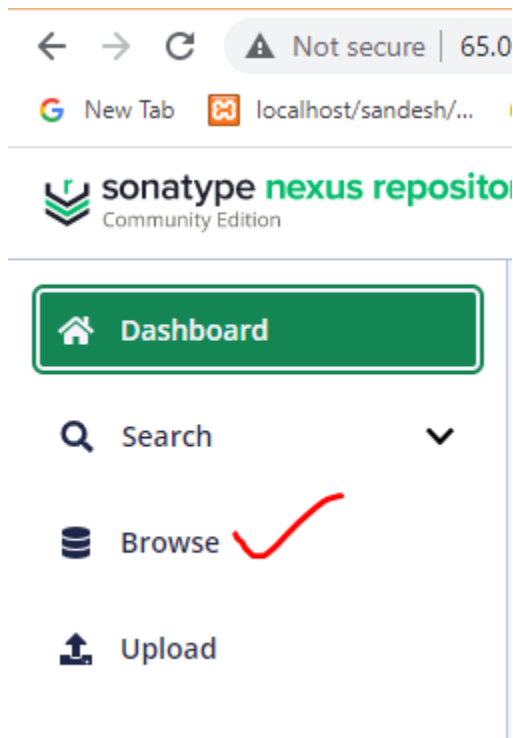
[INFO] Installing /home/ubuntu/SpringBootApplication/target/SpringBootApplication-1.0.0.jar to /home/ubuntu/.m2/repository/com/example/SpringBootApplication/1.0.0/SpringBootApplication-1.0.0.jar
[INFO] Installing /home/ubuntu/SpringBootApplication/pom.xml to /home/ubuntu/.m2/repository/com/example/SpringBootApplication/1.0.0/SpringBootApplication-1.0.0.pom
[INFO]
[INFO] --- maven-deploy-plugin:2.8.2:deploy (default-deploy) @ SpringBootApplication ---
Uploading to nexus: http://65.0.4.16:8081/repository/sdm-maven-releases/com/example/SpringBootApplication/1.0.0/SpringBootApplication-1.0.0.jar (18 MB at 17 MB/s)
Uploading to nexus: http://65.0.4.16:8081/repository/sdm-maven-releases/com/example/SpringBootApplication/1.0.0/SpringBootApplication-1.0.0.pom (1.3 kB at 20 kB/s)
Downloading from nexus: http://65.0.4.16:8081/repository/sdm-maven-releases/com/example/SpringBootApplication/maven-metadata.xml
Uploading to nexus: http://65.0.4.16:8081/repository/sdm-maven-releases/com/example/SpringBootApplication/maven-metadata.xml (304 B at 5.2 kB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.621 s
[INFO] Finished at: 2025-06-09T05:13:22Z
[INFO] -----
ubuntu@ip-172-31-33-44:~/SpringBootApplication$

```

In browser, go to Nexus:

<http://<EC2-PUBLIC-IP>:8081>

Navigate to → Browse → sdm-maven-releases → com/example/SpringBootApplication



Browse							
Filter							
	Name ↑	Type	Format	Status	URL	Health check	Firewall Re...
	maven-central	proxy	maven2	Online - Ready to Connect	copy	0 0	
	maven-public	group	maven2	Online	copy		
	maven-releases	hosted	maven2	Online	copy		
	maven-snapshots	hosted	maven2	Online	copy		
	nuget-group	group	nuget	Online	copy		
	nuget-hosted	hosted	nuget	Online	copy		
	nuget.org-proxy	proxy	nuget	Online - Ready to Connect	copy	0 0	
	<u>sdm-maven-releases</u>	hosted	maven2	Online	copy		

You'll see the .jar published.

⚠ Not secure | 65.0.4.16:8081/#browse/browse:sdm-maven-releases

sonatype nexus repository
 Community Edition

Dashboard
 Search
 Browse
 Upload
 Settings

Browse / sdm-maven-releases

Upload component
 [HTML View](#)

com
 example
 SpringBootApplication
 1.0.0
 maven-metadata.xml
 maven-metadata.xml.md5
 maven-metadata.xml.sha1

Conclusion:

This experiment demonstrates the Maven build lifecycle by creating a simple Java project and executing various Maven build phases. Maven simplifies the build process by providing a standardized way to manage dependencies, compile code, run tests, and package applications. Understanding these build phases is essential for Java developers using Maven in their projects.

Exercise/Questions:

1. What is Maven, and why is it commonly used in software development?
2. Explain the purpose of the pom.xml file in a Maven project.
3. How does Maven simplify dependency management in software projects?
4. What are Maven plugins, and how do they enhance the functionality of Maven?
5. List the key phases in the Maven build lifecycle, and briefly describe what each phase does.
6. What is the primary function of the clean phase in the Maven build lifecycle?
7. In Maven, what does the compile phase do, and when is it typically executed?
8. How does Maven differentiate between the test and verify phases in the build lifecycle?
9. What is the role of the install phase in the Maven build lifecycle, and why is it useful?
10. Explain the difference between a local repository and a remote repository in the context of Maven.