

# Steps to Launch, Build, Deploy, and Store Java Web Application Artifacts in Nexus

## 1. Server Setup

Launch three servers:

1. **\*\*Build Server\*\*** – Install Java and Maven.

Select the required image and the instance type as shown below

**name and tags** [Info](#)

Name

[Add additional tags](#)

**▼ Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Recents

Quick Start

Amazon Linux

aws

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Debian

debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-0ddac4b9aed8d5d46 (64-bit (x86)) / ami-0d36c13877cfad64e (64-bit (Arm))

Virtualization: hvm    ENA enabled: true    Root device type: ebs

**Description**

Ubuntu Server 22.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 22.04, amd64 jammy image

Architecture

64-bit (x86)

AMI ID

ami-0ddac4b9aed8d5d46

Publish Date

2025-08-22

Username

ubuntu

Verified provider

**▼ Instance type** [Info](#) | [Get advice](#)

Instance type

t3.micro

Free tier eligible

Family: t3    2 vCPU    1 GiB Memory    Current generation: true    On-Demand SUSE base pricing: 0.0124 USD per Hour

On-Demand Linux base pricing: 0.0124 USD per Hour    On-Demand Ubuntu Pro base pricing: 0.0159 USD per Hour

On-Demand Windows base pricing: 0.0216 USD per Hour    On-Demand RHEL base pricing: 0.0412 USD per Hour

☒ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

## Create key pair

### Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

Build1-server

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA  
RSA encrypted private and public key pair

☐ ED25519  
ED25519 encrypted private and public key pair

Private key file format

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn](#)

Cancel

Create key pair

After checking all the necessary configurations, now Launch the instance

aws

Search

[Alt+S]

United States (N. California)

Account ID: 4486-5873-6202

Indugon

EC2 > Instances > Launch an instance

Help to learn about Amazon EC2 instances.

▼ Configure storage Info

Advanced

1x 16 GiB gp2 Root volume, Not encrypted

Add new volume

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Edit

► Advanced details Info

▼ Summary

Number of instances Info

1

Software image (AMI)

Canonical, Ubuntu, 22.04, amd64...[read more](#)

ami-0ddac4b9aed8d5d46

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

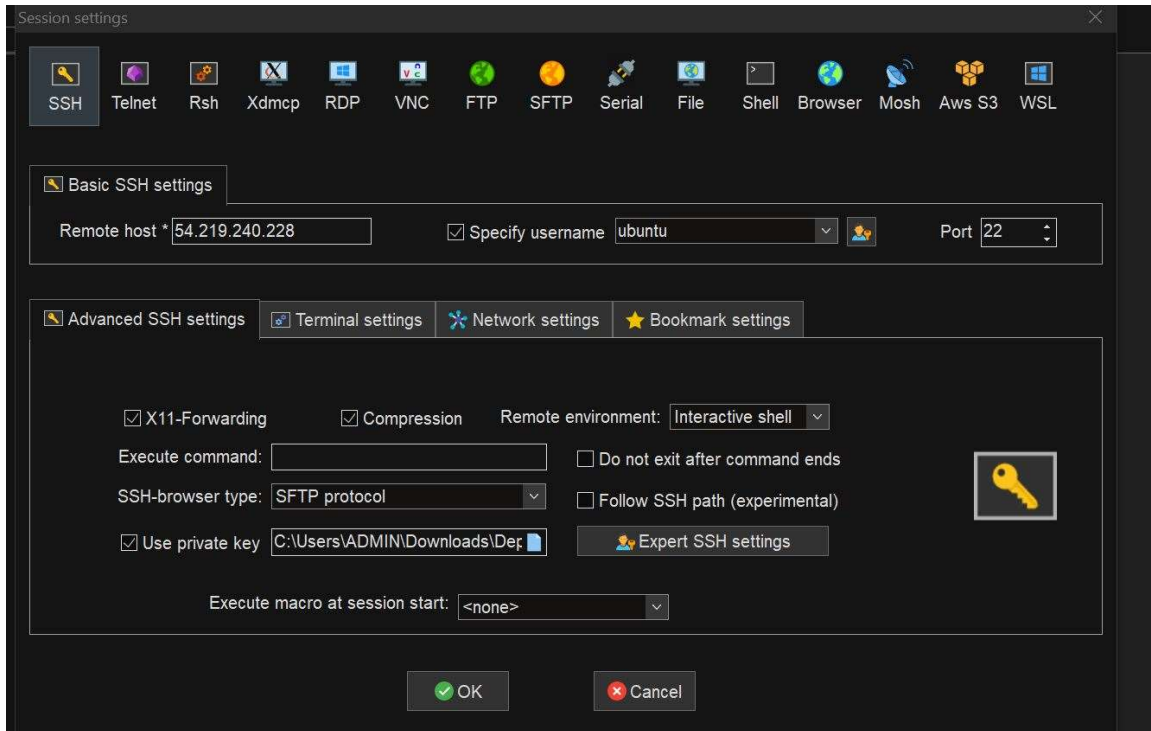
1 volume(s) - 16 GiB

Cancel

Launch instance

Preview code

Connect to the instance by copying the public address of the instance through mobaxtreme



After connecting to the server, now update the server by using command

**sudo apt update -y**

```
ubuntu@ip-172-31-30-110:~$ sudo apt update -y
```

Install java by using command

**sudo apt install openjdk-17-jdk-headless**

```
ubuntu@ip-172-31-30-110:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-11-jre-headless # version 11.0.28+6-1ubuntu1~22.04.1, or
sudo apt install default-jre             # version 2:1.11-72build2
sudo apt install openjdk-17-jre-headless # version 17.0.16+8~us1-0ubuntu1~22.04.1
sudo apt install openjdk-18-jre-headless # version 18.0.2+9-2~22.04
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-0ubuntu3~22.04
sudo apt install openjdk-21-jre-headless # version 21.0.8+9~us1-0ubuntu1~22.04.1
sudo apt install openjdk-25-jre-headless # version 25+36-1~22.04.2
sudo apt install openjdk-8-jre-headless  # version 8u462-ga~us1-0ubuntu2~22.04.2
ubuntu@ip-172-31-30-110:~$ sudo apt install openjdk-17-jre-headless
```

Install maven by using command

**sudo apt install maven -y**

```
ubuntu@ip-172-31-30-110:~$ sudo apt install maven -y
```

Now clone the code by using the command

**git clone <https://github.com/akracad/JavaWebCal.git>**

```
ubuntu@ip-172-31-30-110:~$ git clone https://github.com/akracad/JavaWebCal.git
Cloning into 'JavaWebCal'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 29 (delta 3), reused 29 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (29/29), 5.78 KiB | 1.93 MiB/s, done.
Resolving deltas: 100% (3/3), done.
ubuntu@ip-172-31-30-110:~$
```

2. **\*\*Deploy Server\*\*** – Install Java and Tomcat.

Here, follow the same step on Build server to install java

Now, install tomcat by navigating it to tomcat official page

Archives

**Documentation**

Tomcat 11.0  
Tomcat 10.1  
Tomcat 9.0  
Upgrading  
Tomcat Connectors  
Tomcat Native 2  
Tomcat Native 1.3  
Wiki  
Migration Guide  
Presentations  
Specifications

**Problems?**

Security Reports  
Find help  
FAQ  
Mailing Lists  
Bug Database  
IRC

## 9.0.111

Please see the [README](#) file for packaging information. It explains what every distribution contains.

### Binary Distributions

- Core:
  - [zip](#) ([pgp](#), [sha512](#))
  - [tar.gz](#) ([pgp](#), [sha512](#))
  - [32-bit Windows zip](#) ([pgp](#), [sha512](#))
  - [64-bit Windows zip](#) ([pgp](#), [sha512](#))
  - [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [sha512](#))
- Full documentation:
  - [tar.gz](#) ([pgp](#), [sha512](#))
- Deployer:
  - [zip](#) ([pgp](#), [sha512](#))
  - [tar.gz](#) ([pgp](#), [sha512](#))
- Embedded:
  - [tar.gz](#) ([pgp](#), [sha512](#))
  - [zip](#) ([pgp](#), [sha512](#))

After copying the required version of tomcat, install it on Linux terminal by using command

**wget <https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.111/bin/apache-tomcat-9.0.111.tar.gz>**

```
ubuntu@ip-172-31-31-51:~$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.111/bin/apache-tomcat-9.0.111.tar.gz
```

After installing, we find a tar file and we need to untar it by

```
ubuntu@ip-172-31-31-51:~$ ls
apache-tomcat-9.0.111.tar.gz
ubuntu@ip-172-31-31-51:~$ tar -xvf apache-tomcat-9.0.111.tar.gz
```

Allow 8080 port number in inbound rules of security groups of deploy server to access it on browser

**Edit inbound rules** info  
Inbound rules control the incoming traffic that's allowed to reach the instance.

| Security group rule ID | Type       | Protocol | Port range | Source  | Description - optional |        |
|------------------------|------------|----------|------------|---------|------------------------|--------|
| sgr-03d09095c3eddb403  | SSH        | TCP      | 22         | Custom  | 0.0.0.0/0              | Delete |
| -                      | Custom TCP | TCP      | 8080       | Anyw... | 0.0.0.0/0              | Delete |

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

### 3. \*\*Nexus Repository Server\*\* – Install Java and Nexus Repository Manager.

Here Here, follow the same step on Build and Deploy server to install java

After installing java, we need to install Nexus on this server by navigating it to Nexus official page

 **sonatype** | [DOCUMENTATION](#)

Download

- Nexus Repository 3.70.x Downloads with OrientDB
- Download Archives - Repository Manager 3
- Release Notes
- Sonatype Nexus Repository System Requirements
- Performance Data
- Nexus Repository Professional Features
- Nexus Repository Feature Matrix

Sonatype Help / Sonatype Nexus Repository / Product Information / Download

|                          |  |
|--------------------------|--|
| <b>Unix/Linux x86-64</b> | Unix-like operating system running on a 64-bit x86-compatible processor.<br><a href="#">Nexus Repository 3.85.0 for Linux x86-64 (MD5, SHA1, SHA256, SHA512)</a>           |
| <b>Unix/Linux Arch64</b> | Unix-like operating system running on a processor using the AArch64 architecture.<br><a href="#">Nexus Repository 3.85.0 for Linux Aarch64 (MD5, SHA1, SHA256, SHA512)</a> |
| <b>Windows</b>           | Windows operating system running on computers with 64-bit Intel processors.  |

We need to install by using wget command followed by the url link

Wget [https://download.sonatype.com/nexus/3/nexus-3.85.0-03-linux-x86\\_64.tar.gz](https://download.sonatype.com/nexus/3/nexus-3.85.0-03-linux-x86_64.tar.gz)



After getting the tar file, we need to untar it by using command

**tar -xvf nexus-3.85.0-03-linux-x86\_64.tar.gz**

```
ubuntu@ip-172-31-8-199:~$ ls
nexus-3.85.0-03-linux-x86_64.tar.gz
ubuntu@ip-172-31-8-199:~$ tar -xvf nexus-3.85.0-03-linux-x86_64.tar.gz
```

We need to navigate to nexus-3.85.0-03/bin to start the nexus

**cd nexus-3.85.0-03/bin**

After moving to the folder, we need to start nexus by using command

**./nexus start**

```
ubuntu@ip-172-31-8-199:~$ cd nexus-3.85.0-03/
ubuntu@ip-172-31-8-199:~/nexus-3.85.0-03$ ls
NOTICE.txt  OSS-LICENSE.txt  bin  deploy  etc  jdk
ubuntu@ip-172-31-8-199:~/nexus-3.85.0-03$ cd bin/
ubuntu@ip-172-31-8-199:~/nexus-3.85.0-03/bin$ ls
nexus  nexus.vmoptions  sonatype-nexus-repository-3.85.0-03.jar
ubuntu@ip-172-31-8-199:~/nexus-3.85.0-03/bin$ ./nexus start
Starting nexus
ubuntu@ip-172-31-8-199:~/nexus-3.85.0-03/bin$
```

Allow 8080 port number in inbound rules of security groups of deploy server to access it on browser

**Edit inbound rules** [Info](#)  
Inbound rules control the incoming traffic that's allowed to reach the instance.

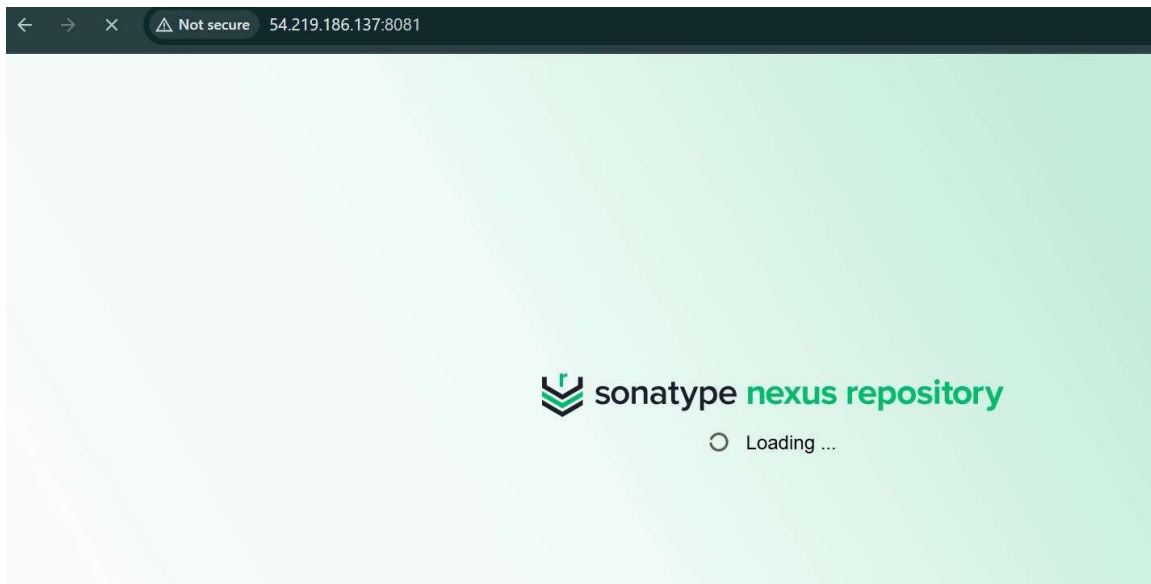
| Security group rule ID | Type <a href="#">Info</a> | Protocol <a href="#">Info</a> | Port range <a href="#">Info</a> | Source <a href="#">Info</a> | Description - optional <a href="#">Info</a> |                        |
|------------------------|---------------------------|-------------------------------|---------------------------------|-----------------------------|---|------------------------|
| sgr-03d09095c3eddb403  | SSH                       | TCP                           | 22                              | Custom                      | <input type="text"/>                        | <a href="#">Delete</a> |
| -                      | Custom TCP                | TCP                           | 8080                            | Anyw...                     | <input type="text"/>                        | <a href="#">Delete</a> |

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

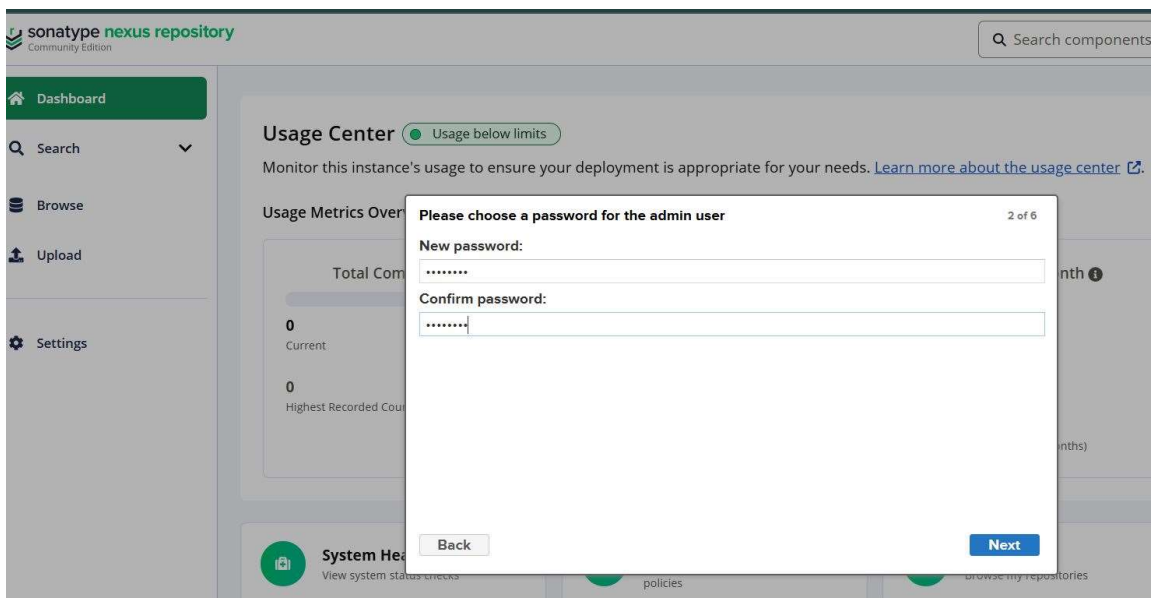
Access the Nexus by copying public ip address on the browser followed by 8081 (port number)



After entering into home page, get the password by copying the given path on home page

```
ubuntu@ip-172-31-31-171:~/nexus-3.85.0-03/bin$ cat /home/ubuntu/sonatype-work/nexus3/admin.password  
e86f44d6-f250-4d1c-a106-d5178dd98ce0ubuntu@ip-172-31-31-171:~/nexus-3.85.0-03/bin$
```

After entering the password, it will ask for password change



Now create a repository by navigating to settings

The screenshot shows the Sonatype Nexus Community Edition interface. On the left is a sidebar with a 'Settings' menu containing 'Repository' (selected), 'Repositories', 'Blob Stores', 'Data Store', 'Proprietary Repositories', 'Content Selectors', 'Cleanup Policies', 'Routing Rules', 'Security', 'Support', 'Logging', 'Logs', 'Status', 'Support ZIP', 'System Information', 'System', and 'IQ Server'. The main panel is titled 'Repositories' and shows the 'Create Repository: maven2 (hosted)' form. The form fields are: 'Name' (Java-Cal-App), 'Online' (checked), 'Maven 2' section with 'Version policy' (Release), 'Layout policy' (Strict), 'Content Disposition' (Inline), 'Storage' section with 'Blob store' (default), 'Strict Content Type Validation' (checked), 'Hosted' section with 'Deployment policy' (Disable redeploy), 'Proprietary Components' (unchecked), and 'Cleanup' section with 'Cleanup Policies' (empty). At the bottom are 'Create repository' and 'Cancel' buttons.

Now, in the build server where we have cloned our git repo, navigate to directory where index.jsp is present

```
ubuntu@ip-172-31-30-110:~/JavaWebCal/src/main/webapp$ pwd
/home/ubuntu/JavaWebCal/src/main/webapp
ubuntu@ip-172-31-30-110:~/JavaWebCal/src/main/webapp$ ls
WEB-INF  index.jsp
ubuntu@ip-172-31-30-110:~/JavaWebCal/src/main/webapp$
```



After navigating to the above path, we need to change the functions in order to create different versions of the code

```
<input type="text" name="n1" />
<br />
<label>Second number : </label>
<input type="text" name="n2" />
<br />
  <div>
    <label>
      <input type="radio" name="r1" value="add" />addition
    <br />
  </label>

</div>
<input type="submit" value="submit" />
</form>
</body>
</html>
~
~
```

After editing the index.jsp, now edit the pom.xml in which we need to give the nexus repository link and also the required information regarding Nexus.

After entering into pom.xml search for dependency management and edit as shown below

```
  <distributionManagement>
<repository>
<id>Java-web-app</id>
<url>http://54.219.186.137:8081/repository/Java-web-app/</url>
</repository>
</distributionManagement>
</project>
```

Now, here search for the version and given the desired name of the artifact

```
<modelVersion>4.0.0</modelVersion>

    <groupId>com.web.cal</groupId>
<artifactId>webapp-add</artifactId>
<version>0.0.1</version>
<packaging>war</packaging>
<name>WebAppCal Maven Webapp</name>
<url>http://maven.apache.org</url>
```

After editing the pom.xml file

We need to navigate to etc/maven and here we should give the credentials of the nexus server

```
ubuntu@ip-172-31-30-110:/etc/maven$ pwd
/etc/maven
ubuntu@ip-172-31-30-110:/etc/maven$ ls
logging  m2.conf  settings.xml  toolchains.xml
ubuntu@ip-172-31-30-110:/etc/maven$
```

After entering into settings.xml, search for the lines <server> and give the appropriate credentials of the nexus server for ensure the connection between maven and Nexus server

```
<!--
<server>
  <id>Java-web-app</id>
  <username>admin</username>
  <password>admin123</password>
</server>
```

After making all the necessary configurations, now go to directory where we have pom.xml

Enter command mvn package, then you can see output like

```
[INFO] --- maven-war-plugin:3.3.2:war (default-war) @ webapp-add ---
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp-add] in [/home/ubuntu/JavaWebCal/target/webapp-add-0.0.1]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ubuntu/JavaWebCal/src/main/webapp]
[INFO] Building war: /home/ubuntu/JavaWebCal/target/webapp-add-0.0.1.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.173 s
[INFO] Finished at: 2025-10-15T13:16:18Z
[INFO] -----
ubuntu@ip-172-31-30-110:~/JavaWebCal$
```

Enter command mvn deploy, then you can see output like

```
[INFO] --- maven-deploy-plugin:2.7:deploy (default-deploy) @ webapp-add ---
Uploading to Java-web-app: http://54.219.186.137:8081/repository/Java-web-app/com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.war
Uploaded to Java-web-app: http://54.219.186.137:8081/repository/Java-web-app/com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.war (3.8 kB at 13 kB/s)
Uploading to Java-web-app: http://54.219.186.137:8081/repository/Java-web-app/com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.pom
Uploaded to Java-web-app: http://54.219.186.137:8081/repository/Java-web-app/com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.pom (1.4 kB at 10 kB/s)
Downloading from Java-web-app: http://54.219.186.137:8081/repository/Java-web-app/com/web/cal/webapp-add/maven-metadata.xml
Uploading to Java-web-app: http://54.219.186.137:8081/repository/Java-web-app/com/web/cal/webapp-add/maven-metadata.xml
Uploaded to Java-web-app: http://54.219.186.137:8081/repository/Java-web-app/com/web/cal/webapp-add/maven-metadata.xml (301 B at 7.0 kB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.317 s
[INFO] Finished at: 2025-10-15T13:27:23Z
[INFO] -----
```

Now you can see the artifact is being stored in Nexus, go and check repository of Nexus which is accessed on the browser.

← → ↻ Not secure 54.153.85.253:8081/#browse/search/generic=keyword%3Dwebapp-add:maven2-4f1bbccd

sonatype nexus repository  
Community Edition

Dashboard

Search

Custom

Maven

NuGet

Browse

Upload

Settings

Search / webapp-add

| Repository | Java-web-app | Group   | com.web.cal | Most popular version |  |
|------------|--------------|---------|-------------|----------------------|--|
| Format     | maven2       | Name    | webapp-add  | Age                  |  |
|            |              | Version | 0.0.1       | Popularity           |  |

Delete component Analyze application

| Name  |
|---|
| /com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.pom      |
| /com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.pom.md5  |
| /com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.pom.sha1 |
| /com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.war      |
| /com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.war.md5  |
| /com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.war.sha1 |

After this if you want to deploy it on tomcat, then copy the endpoint from the artifact in Nexus and paste it in deploy server using wget command

```
ubuntu@ip-172-31-31-171:~/nexus-3.85.0-03/bin$ cd
ubuntu@ip-172-31-31-171:~$ wget http://com/web/cal/webapp-add/0.0.1/webapp-add-0.0.1.war
```

Now it is being copied to webapps in tomcat, you can access it in browser



## Calculator

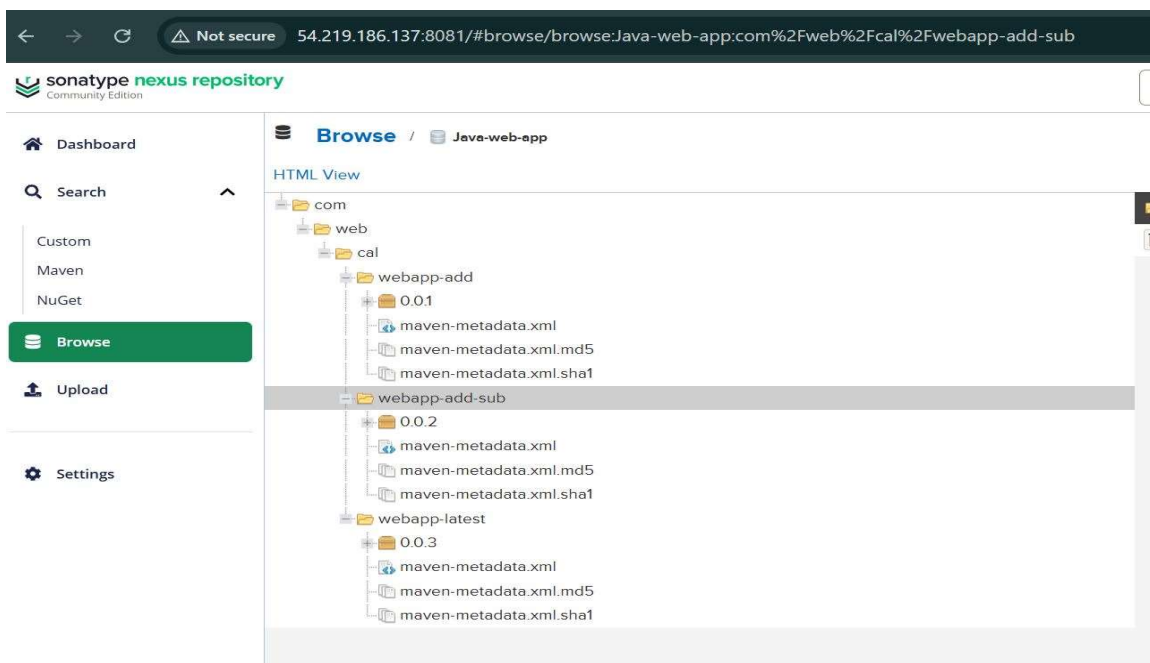
first number:

Second number :

☐ addition

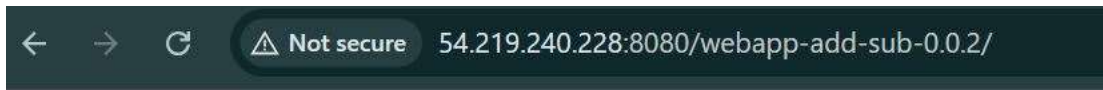
Now, follow the same steps for all the other versions and get the versions deployed.

Below is the screenshot of all the versions in Nexus repository



Now use wget command and deploy all the other versions as show above steps

The output of 2 functions applications is



## Calculator

first number:

Second number :

☐ addition

☐ Subtraction

In this way, we can store the artifacts build through maven into Nexus repository manager and deploy them using the artifacts on Tomcat webserver.