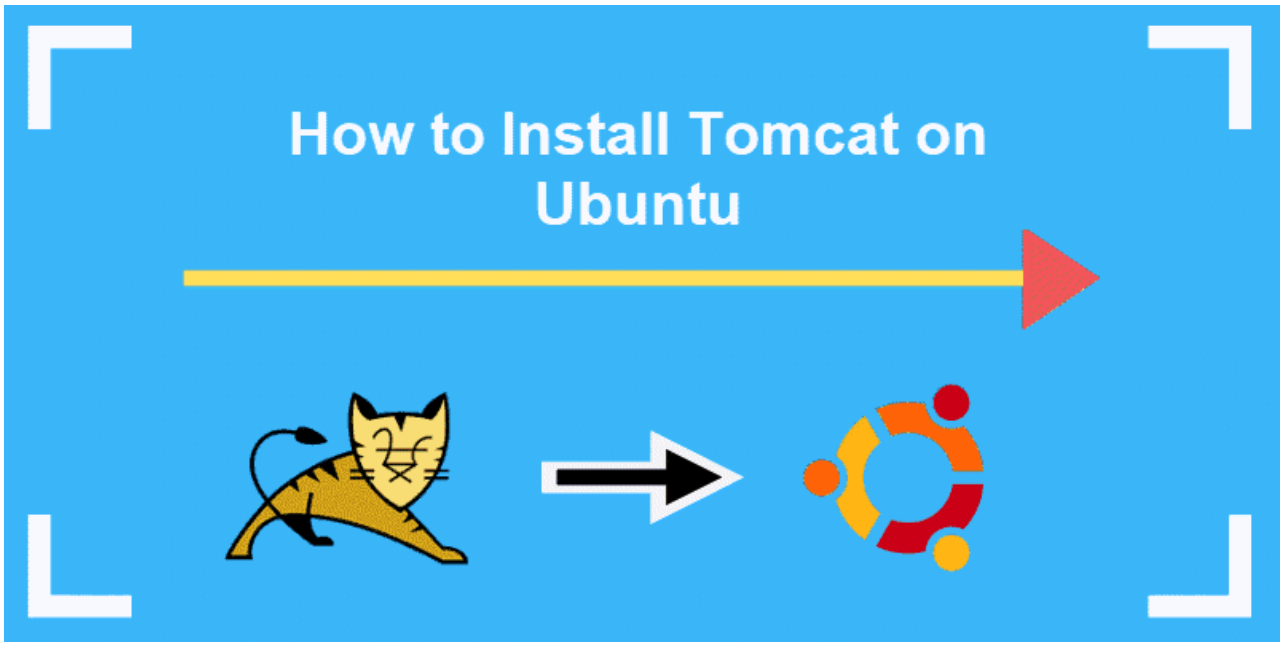
# How to Install Apache Tomcat 9 on Ubuntu 18.04

## Introduction

**Apache Tomcat** is a free, open-source, lightweight application server used for Java-based web applications. Developers use it to implement Java Servlet and JavaServer Pages technologies (including Java Expression Language and Java WebSocket).

**Read this guide to learn how to install and configure Apache Tomcat on Ubuntu 18.04.**




**Note:** These steps apply to other Ubuntu-based distributions, including Ubuntu 16.04, Linux Mint, and Elementary OS.

## Prerequisites

- An Ubuntu-based distribution (such as Ubuntu 18.04)
- A user account with **sudo** privileges
- A terminal window (**Ctrl**–**Alt**–**T**)
- The apt package manager, included by default

# Steps for Installing Tomcat 9 on Ubuntu

Follow the steps below to Install Tomcat on Ubuntu.

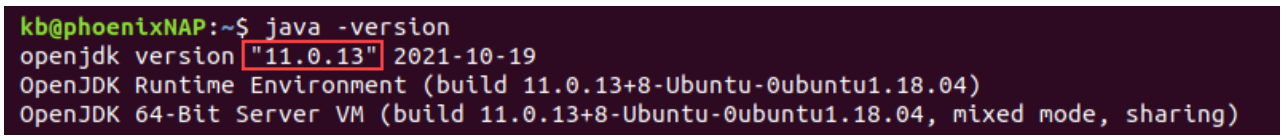
## Step1: Check if Java is Installed

Before you can download and install **Tomcat**, make sure you have the required Java installation for Ubuntu (**OpenJDK**).

Open the terminal (**Ctrl+Alt+T**) and use the following command check the Java version:

```
java -version
```

The output will show the Java version running on your system. Currently, the latest release is **OpenJDK 11.0.3**:




## Step 2: Install OpenJDK

If you do not have OpenJDK or have a version older than Java 8, install the newest release by typing the following:

```
sudo apt install default-jdk
```

## Step 3: Create Tomcat User and Group

For security reasons, do not run Tomcat under the root user. Create a new group and

system user to run the Apache Tomcat service from the **/opt/tomcat** directory.

```
sudo groupadd tomcat
```

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

```
kb@phoenixNAP:~$ sudo groupadd tomcat
kb@phoenixNAP:~$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
kb@phoenixNAP:~$
```

## Step 4: Download Tomcat 9

1. Download the latest binary Tomcat release navigate to the official Apache Tomcat Download page.

2. On it, find the **Binary Distributions** > **Core** list and the **tar.gz** link in it. Copy the link of the file.


3. Go back to the terminal and change to the **/tmp** directory with the command:

```
cd /tmp
```

4. Now, use the curl command with the **tar.gz** link you copied in step 2 to download the package:

```
curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.56/bin/apach
e-tomcat-9.0.56.tar.gz
```

```
kb@phoenixNAP:~$ cd /tmp
kb@phoenixNAP:/tmp$ curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.56/bin/apache-
tomcat-9.0.56.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
```

```
100 11.0M  100 11.0M    0     0  9835k      0  0:00:01  0:00:01 --:--:-- 9835k
kb@phoenixNAP:/tmp$
```

## Step 5: Extract tar.gz File

1. To extract the tar.gz Tomcat file, create a new **/opt/tomcat/** directory with the command:

```
sudo mkdir /opt/tomcat
```

2. Then, extract the file in the new directory with the following command:

```
sudo tar xzvf apache-tomcat-9*tar.gz -C /opt/tomcat --strip-compon
ents=1
```

## Step 6: Modify Tomcat User Permission

The new Tomcat user you created does not have executable privileges, but it needs to access the installation directory. You need to setup execute privileges over the directory.

1. Move to the directory where the Tomcat installation is located:

```
cd /opt/tomcat
```

2. Grant group and user ownership over the installation directory to the **tomcat** group and user with the command:

```
sudo chown -RH tomcat: /opt/tomcat
```

3. Lastly, change script permissions to grant execute access in /opt/tomcat/bin/with:

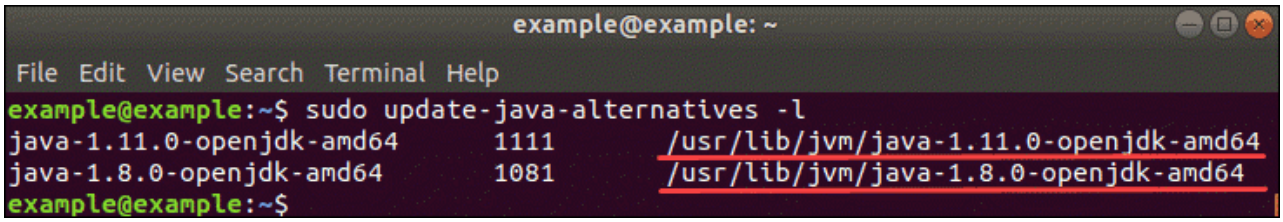
```
sudo sh -c 'chmod +x /opt/tomcat/bin/*.sh'
```

## Step 7: Create System Unit File

Since you are going to to use Tomcat as a service, you need to create a **systemd service file**.

1. To configure the file, you first need to find the **JAVA_HOME** path. This is the exact location of the Java installation package.
To do so, prompt the system to give you information about the Java packages installed on the system. In the terminal, type:

```
sudo update-java-alternatives -l
```




As the output shows, there are two available versions of Java. Accordingly, it also shows two paths displaying their location.

Choose the version you want to use and copy its location. With that, you can move on to create the service file.

2. Create and open a new file in the **/etc/system/system** under the name **tomcat.service**:

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

3. Once the file opens, copy and paste the content below, changing the **JAVA_HOME** value to the information you found in the previous step.

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Dj
ava.awt.headless=true"
```

```
Environment="CATALINA_BASE=/opt/tomcat"

Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UsePara
llelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

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

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"

Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

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

4. **Save** and **Exit** the file (**Ctrl+X**, followed by **y**[es] and **Enter**).

5. For the changes to take place, reload the system daemon with the command:

```
sudo systemctl daemon-reload
```

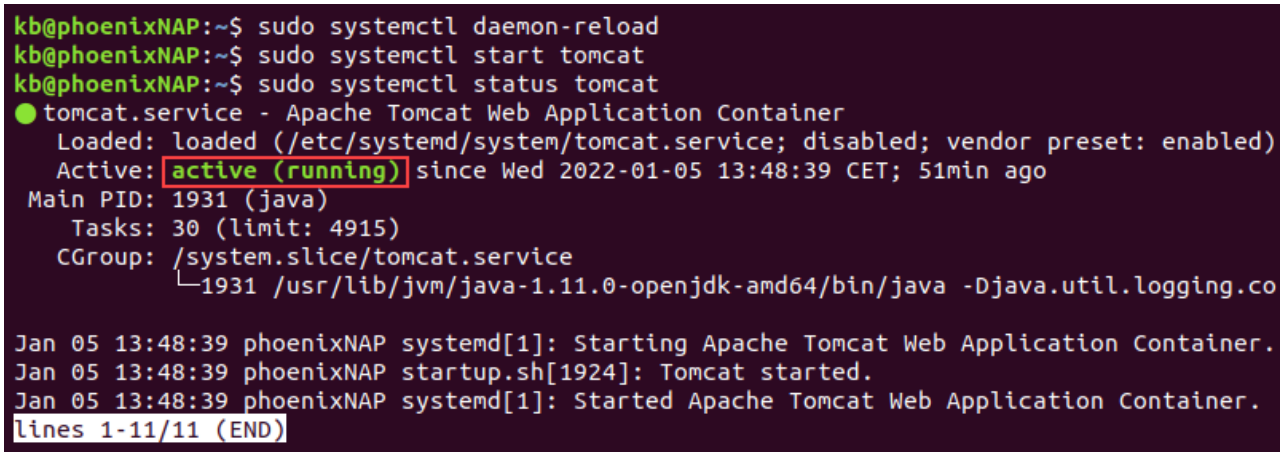
6. Now, you can finally start the Tomcat service:

```
sudo systemctl start tomcat
```

7. Verify the Apache Tomcat service is running with the command:

7. Verify the Apache Tomcat service is running with the command:

```
sudo systemctl status tomcat
```




The message you want to receive is that the service is **active (running)**.

## Step 8: Adjust Firewall

If you are using a firewall to protect your server (as you should), you will not be able to access the Tomcat interface. Tomcat uses Port 8080, which is outside your local network.

1. Open Port 8080 to allow traffic through it with the command:

```
sudo ufw allow 8080/tcp
```

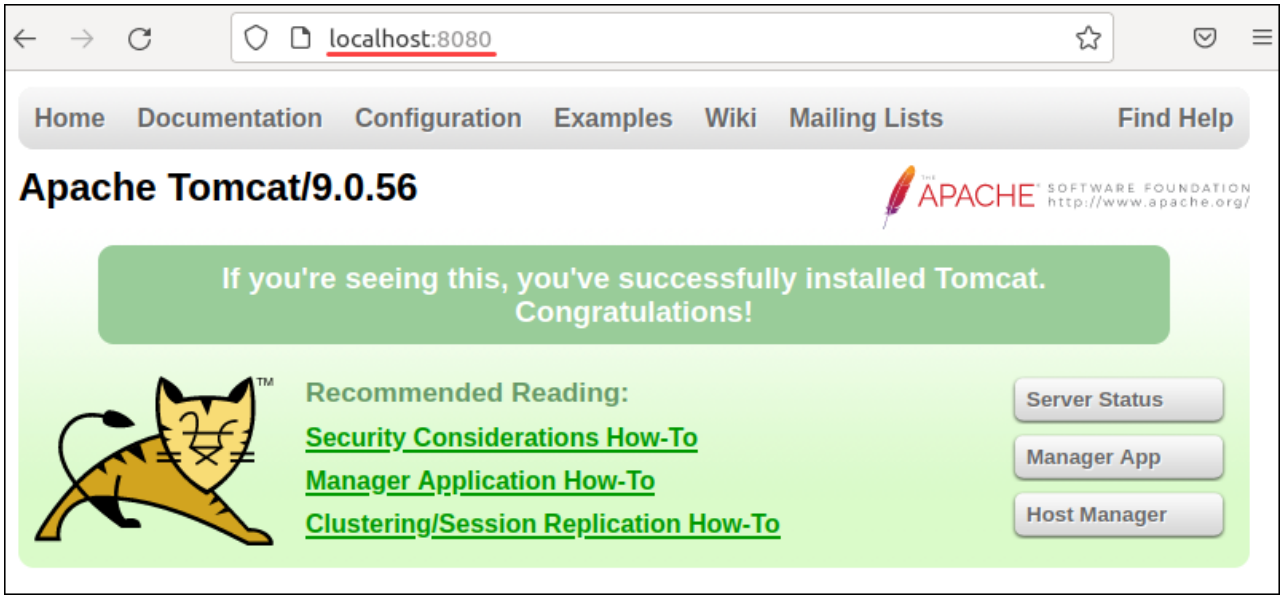
2. If the port is open, you should be able to see the Apache Tomcat splash page. Type the following in the browser window:

```
http://server_ip:8080
```

or

```
http://localhost:8080
```

Your web browser should open the web page as in the image below:




## Step 9: Configure Web Management Interface

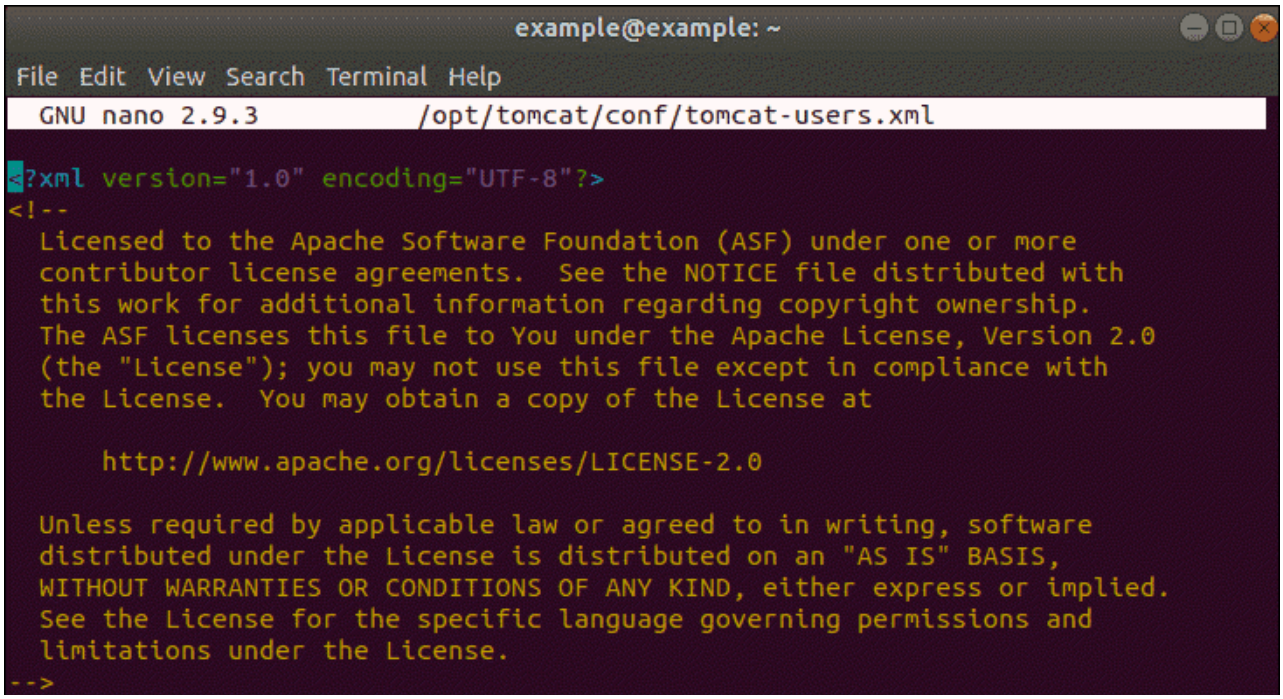
Once you verified the service is running properly, you need to create a user who can use the web management interface.

To do this, open and edit the users file.

1. Open the **users file** with the command:

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

The file should appear like the one in the image below:



```
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

2. Delete everything from the file and add the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users>
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>
  <role rolename="admin-gui"/>
  <role rolename="admin-script"/>
  <user username="admin" password="Your_Password" roles="manager-g
ui, manager-script, manager-jmx, manager-status, admin-gui, admin-
script"/>
</tomcat-users>
```

Make sure to replace the **`Your_Password`** value with a strong password of your preference.

3. **Save** and **Exit** the file.

## Step 10: Configure Remote Access

Finally, you need to configure remote access. This is required. By default, Tomcat is only accessible from the local machine.

1. First, open the **manager** file:

```
sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

2. Next, decide whether to grant access from a. **anywhere** or b. **from a specific IP address**.

a. To make it publicly accessible, add the following lines to the file:

```
<Context antiResourceLocking="false" privileged="true">
<!--
```

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+.d+|::1|0000:1" />


-->

</Context>
```

b. To allow access from a specific IP address, add the IP to the previous command, as follows:

```
<Context antiResourceLocking="false" privileged="true">
<!--

<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0000:1|THE.IP.ADDRESS." />

-->

</Context>
```

3. Repeat the same process for the **host-manager** file.

Start by opening the file with the command:

```
sudo nano /opt/tomcat/latest/webapps/host-manager/META-INF/contex
t.xml
```