**Install Apache Tomcat 8 on CentOS 7**

**Below URl :** <https://www.vultr.com/docs/how-to-install-apache-tomcat-8-on-centos-7>

**How to install Tomcat 8 on centos 7. Below are complete installation steps.**

Prerequisites

Before further reading, you need to:

1. Deploy a fresh Vultr CentOS 7 server instance.
2. Log into this machine from your SSH terminal as a non-root sudo user.

**Step 1: Update your CentOS system**

First things first, you need to update the system to the latest stable status:

>> sudo yum install epel-release

>> sudo yum update -y && sudo reboot

Use the same sudo user to log into the system after the reboot finishes.

**Step 2: Install Java**

You need to install Java SE 7.0 or later before Apache Tomcat can run properly. Here, I will install OpenJDK Runtime Environment 1.8.0 using YUM:

>> sudo yum install java-1.8.0-openjdk.x86\_64

Now, you can confirm your installation with:

>> java -version

The output will resemble the following:

>> openjdk version "1.8.0\_91"

>> OpenJDK Runtime Environment (build 1.8.0\_91-b14)

>> OpenJDK 64-Bit Server VM (build 25.91-b14, mixed mode)

**Step 3: Create a dedicated user for Apache Tomcat**

For security purposes, you need to create a dedicated non-root user "tomcat" who belongs to the "tomcat" group:

>> sudo groupadd tomcat

>> sudo mkdir /opt/tomcat

>> sudo useradd -s /bin/nologin -g tomcat -d /opt/tomcat tomcat

**Step 4: Download and install the latest Apache Tomcat**

<http://www-us.apache.org/dist/tomcat/tomcat-8/>

>> cd ~

<http://www-us.apache.org/dist/tomcat/tomcat-8/v8.0.45/bin/apache-tomcat-8.0.45.tar.gz>

>> sudo tar -zxvf apache-tomcat-8.0.33.tar.gz -C /opt/tomcat --strip-components=1

**Step 5: Setup proper permissions**

Before you can run Apache Tomcat, you need to setup proper permissions for several directories:

>> cd /opt/tomcat

>> sudo chgrp -R tomcat conf

>> sudo chmod g+rwx conf

>> sudo chmod g+r conf/\*

>> sudo chown -R tomcat logs/ temp/ webapps/ work/

>> sudo chgrp -R tomcat bin

>> sudo chgrp -R tomcat lib

>> sudo chmod g+rwx bin

>> sudo chmod g+r bin/\*

**Step 6: Setup a Systemd unit file for Apache Tomcat**

As a matter of convenience, you should setup a Systemd unit file for Apache Tomcat:

>> sudo vi /etc/systemd/system/tomcat.service

Populate the file with:

[Unit]

Description=Apache Tomcat Web Application Container

After=syslog.target network.target

[Service]

Type=forking

Environment=JAVA\_HOME=/usr/lib/jvm/jre

Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'

Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

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

ExecStop=/bin/kill -15 $MAINPID

User=tomcat

Group=tomcat

[Install]

WantedBy=multi-user.target

Save and quit:

:wq

**Step 7: Install haveged, a security-related program**

For security purposes, you should install haveged as well:

>> sudo yum install haveged

>> sudo systemctl start haveged.service

>> sudo systemctl enable haveged.service

**Step 8: Start and test Apache Tomcat**

Now, start the Apache Tomcat service and set it run on system boot:

>> sudo systemctl start tomcat.service

>> sudo systemctl enable tomcat.service

Then, you can test your installation of Apache Tomcat by visiting the following URL from a web browser:

http://[your-Vultr-server-IP]:8080

If nothing goes wrong, you will see the default Apache Tomcat front page.

**Hardening and Security Guide**

**1.0 Session Timeout**

The session timeout for all web applications must be set to 20 minutes.

This can be done by editing the file in the CATALINA\_HOME/conf/web.xml and setting the following configuration option:

<session-config>

<session-timeout>20</session-timeout>

</session-config>

* 1. **Secure Authentication**

local and certificate based authentication, an account lockout must be enforced3 . To prevent brute force attacks, the authentication realm in use must be placed within a lockout realm using the following configuration:

Edit the file CATALINA\_HOME/conf/server.xml and add the lockout realm with the following option:

<Realm className="org.apache.catalina.realm.LockOutRealm" failureCount="5" lockOutTime="30">

<!-- AUTHENTICATION REALM -->

</Realm>

**1.2 . Change SHUTDOWN port and Command**

If the functionality to shut down Tomcat using a network port on the local Tomcat system down

is needed, the password must be set to a strong and hard to guess passphrase.

Edit the file CATALINA\_HOME/conf/server.xml and set the shutdown passphrase:

<Server port="8333"

shutdown=" NonDeterministicWordSoShutdownPWisNotEasyToGuess ">

**1.3 Disable Automatic Deployment:**

Tomcat allows automatic deployment of applications while Tomcat is running. This functionality must be disabled as it may allow malicious or untested applications to be deployed.

/conf/server.xml

autoDeploy=”false”

**1.5 Remove Server Banner**

Removing Server Banner from HTTP Header is one of the first things to do as hardening. Having server banner expose the product you are using and leads to information leakage vulnerability.

**Implementation:**

Go to $tomcat/conf folder

Modify server.xml by using vi

Add following under Connector port and save the file

<Connector port="8040" protocol="HTTP/1.1"

connectionTimeout="20000"

**server =" "**

redirectPort="8443" />

**1.6 Enable access log logging**

The default configuration doesn’t capture access logs. The access log is very useful in troubleshooting to check request type, requester IP address, status code, etc.

**Implementation:**

Go to $tomcat/conf

Modify server.xml by using vi

Go to the end of the file and uncomment Valve entry for valves.AccessLogValue

<Valve className="org.apache.catalina.valves.AccessLogValve"

directory="logs"

prefix="localhost\_access\_log."

suffix=".txt"

pattern="common" resolveHosts="false"/>

**1.7 Add Secure flag in cookie**

It is possible to steal or manipulate web application session and cookies without having a Secure flag in HTTP Header as Set-Cookie.

**Implementation:**

Go to $tomcat/conf folder

Modify server.xml by using vi

Add following in Connector port

<Connector port="8040" protocol="HTTP/1.1"

connectionTimeout="20000"

server=" "

**secure="true"**

redirectPort="8443" />

**1.8 Add HttpOnly in cookie**

Best practice to have this enabled at application code level. However, due to bad programming or developer’s unawareness, it comes to Web Infrastructure.

**Implementation:**

Go to $tomcat/conf folder

Modify context.xml by using vi

Add following in Context directive

<context usehttponly=’true’> ... </context>

### 1.9 Remove default/unwanted applications

### By default, Tomcat comes with following web applications, which may or not be required in a production environment. You can delete them to keep it clean and avoid any known security risk with Tomcat default application.

### Remove below directory from Webapps directory(/opt/tomcat/webapps).

* ROOT – Default welcome page
* Docs – Tomcat documentation
* Examples – JSP and servlets for demonstration
* Manager, host-manager – Tomcat administration