**Install Java 8 on CentOS 7**

Posted on [February 8, 2019](https://www.liquidweb.com/kb/install-java-8-on-centos-7/) by [Echo Diaz](https://www.liquidweb.com/kb/author/ediaz/) | Updated: September 10, 2019  
Category: [Technical Support](https://www.liquidweb.com/kb/category/technical-support/), [Tutorials](https://www.liquidweb.com/kb/category/tutorials/) | Tags: [CentOS 7](https://www.liquidweb.com/kb/tag/centos-7/), [Installation](https://www.liquidweb.com/kb/tag/installation/), [Java](https://www.liquidweb.com/kb/tag/java/), [JDK](https://www.liquidweb.com/kb/tag/jdk/), [jre](https://www.liquidweb.com/kb/tag/jre/)

Reading Time: 2 minutes

In this tutorial, we’ll be showing you how to install Oracle’s Java 8 programming language specifically onto a CentOS 7 server. This simple object-oriented language is used for many of the applications and websites you come across today.  Let’s jump right in!

[](https://www.youtube.com/embed/90-0dRxs1fs?rel=)

Pre-flight

1. Open the terminal and login as root.  If you are logged in as another user, you will need to add **sudo** before each command.
2. Working on a [Linux CentOS 7 server](https://www.liquidweb.com/products/vps)
3. No installations of previous Java versions

Installing Java 8 on CentOS 7

Step 1: Update

As a matter of best practice we’ll update before installing any new programs:

yum -y update

Step 2: Install Java 8

yum install java-1.8.0-openjdk

Step 3:  Verify Java is Installed

java -version

**Example Output:**

java -version  
openjdk version "1.8.0\_191"  
OpenJDK Runtime Environment (build 1.8.0\_191-b12)  
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)

**Set Java’s Home Environment**

Step 1: Find Java’s Path

Let’s set the**JAVA\_HOME** variable, using the following command will give us a path so we can set the variable.

update-alternatives --config java

You’ll see a prompt to “Enter to keep the current selection[+], or type selection number:”, if you had multiple Java version you could set the default here, but all we need is the path of Java so we can exit pressing enter.  The highlighted area is the path we will need to copy/paste into our .bash\_profile file.

Selection    Command  
-----------------------------------------------  
\*+ 1           java-1.8.0-openjdk.x86\_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1.el7\_6.x86\_64/jre/bin/java)</code?

Step 2: Setting Java’s Path in Your Environment

After copying your Java’s path, open the .bash\_profile with your text editor.

vim .bash\_profile

Export your Java path into the .bash\_profile by adding the following to the bottom of the file. (Your path may look different from mine, and it’s not important that they vary.)

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1.el7\_6.x86\_64/jre/bin/java

Refresh the File:

Source .bash\_profile

When you use the**JAVA\_HOME** variable you’ll now be able to see the path you set.

echo $JAVA\_HOME

**Example Output:**

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.191.b12-1.el7\_6.x86\_64/jre/bin/java

# How to Install Apache Tomcat 8 on CentOS 7

Last Updated: Thu, May 12, 2016

CentOSJavaLinux GuidesPopularSystem AdminWeb Servers

Apache Tomcat is an open-source web server that is designed to serve Java web pages. It is widely deployed and powers various mission-critical web applications around the world.

As a starter guide, this article explains how to install Apache Tomcat 8, the latest stable version of Apache Tomcat, onto a Vultr CentOS 7 server instance.

### Prerequisites

Before further reading, you need to:

* Deploy a fresh Vultr CentOS 7 server instance.
* 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

In this fashion, you created a user "tomcat" who belongs to the group "tomcat". You cannot use this user account to log into the system. The home directory is /opt/tomcat, which is where the Apache Tomcat program will reside.

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

You can always find the latest stable version of Apache Tomcat 8 from its [official download page](http://tomcat.apache.org/tomcat-8.0-doc/index.html), which is 8.0.33 as of writing.

Under the "Binary Distributions" section and then the "Core" list, use the link pointing to the "tar.gz" archive to compose a wget command:

cd ~

wget http://www-us.apache.org/dist/tomcat/tomcat-8/v8.0.33/bin/apache-tomcat-8.0.33.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

In order to test Apache Tomcat in a web browser, you need to modify the firewall rules:

sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp

sudo firewall-cmd --reload

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.

### Step 9: Configure the Apache Tomcat web management interface

In order to use the "Manager App" and the "Host manager" in the Apache Tomcat web interface, you need to create an admin user for your Apache Tomcat server:

sudo vi /opt/tomcat/conf/tomcat-users.xml

Within the </tomcat-users ...>...</tomcat-users> segment, insert a line to define a admin user:

<user username="yourusername" password="yourpassword" roles="manager-gui,admin-gui"/>

Remember to replace "yourusername" and "yourpassword" with your own ones, the less common the better.

Save and quit:

:wq

Restart Apache Tomcat to put your modifications into effect:

sudo systemctl restart tomcat.service

Refresh the Apache Tomcat front page from your web browser. Log in the "Manager App" and the "Host manager" using the credentials you had setup earlier.

The Apache Tomcat setup is complete. You can now use it to deploy your own applications.

# How To Open A Port In CentOS / RHEL 7

By [admin](https://www.thegeekdiary.com/author/sandeep_patil/)

A TCP/IP network connection may be either blocked, dropped, open, or filtered. These actions are generally controlled by the IPtables firewall the system uses and is independent of any process or program that may be listening on a network port. This post will outline the steps to open a port required by a application. For this post example, we will be opening Application Specific (Apache) Port 55555.

Server details are as below:

# uname -a

Linux geeklab 3.10.0-693.17.1.el7.x86\_64 #1 SMP Thu Jan 25 20:13:58 UTC 2018 x86\_64 x86\_64 x86\_64 GNU/Linux

# cat /etc/redhat-release

CentOS Linux release 7.4.1708 (Core)

## 1. Check Port Status

Check that the port is not open and Apache is not showing that port:

# netstat -na | grep 55555

# lsof -i -P |grep http

httpd 5823 root 4u IPv6 42212 0t0 TCP \*:80 (LISTEN)

## 2. Check Port Status in iptables

Check that iptables are not showing that port open:

# iptables-save | grep 55555

## 3. Add the port

Add the test port in **/etc/services** file and allow the port to accept packets. Test port can be added by editing /etc/services file in below format:

# vi /etc/services

service-name port/protocol [aliases ...] [# comment]

# vi /etc/services

testport 55555/tcp # Application Name

## 4. Open firewall ports

Add Firewall rule to allow the port to accept packets:

# firewall-cmd --zone=public --add-port=55555/tcp --permanent

success

# firewall-cmd --reload

success

# iptables-save | grep 55555

-A IN\_public\_allow -p tcp -m tcp --dport 55555 -m conntrack --ctstate NEW -j ACCEPT

## 5. Check newly added port status

After adding the port for httpd and reloading httpd services, notice now httpd is also listening to newly added port 55555:

# lsof -i -P |grep http

httpd 6595 root 4u IPv6 43709 0t0 TCP \*:80 (LISTEN)

httpd 6595 root 6u IPv6 43713 0t0 TCP \*:55555 (LISTEN)

# netstat -na |grep 55555

tcp6 0 0 :::55555 :::\* LISTEN