# APACHE TOMCAT INSTALLATION (as service)

1. Download the package and required version of tomcat using the below command,

Note: You can download the package in any directory you want, but /opt/tomcat is recommendable.

1. Create the Directory where you want to store the downloaded packages.

|  |
| --- |
| sudo mkdir -p /opt/tomcat  sudo wget -P /opt/tomcat  [http://mirror.cogentco.com/pub/apache/tomcat/tomcat-8/v8.5.57/bin/apache-tomcat-8.5.57.tar.gz](http://mirror.cogentco.com/pub/apache/tomcat/tomcat-8/v8.5.55/bin/apache-tomcat-8.5.55.tar.gz) |

1. Once downloaded, extract the downloaded binary zip file in the same directory.

|  |
| --- |
| sudo tar -xzvf /opt/tomcat/apache-tomcat-8.5.57.tar.gz |

1. Tomcat requires a system user and group name called “tomcat” to interact and run the config files/scripts with system user permissions.

|  |
| --- |
| sudo groupadd tomcat  sudo useradd -s /bin/false -d /opt/tomcat -g tomcat tomcat  sudo chgrp -R tomcat /opt/tomcat/ |

1. Allow the read permissions for all the conf file located on /opt/tomcat/apache-tomcat-8.5.57/conf/ and execute permission for /opt/tomcat/apache-tomcat-8.5.57/conf directory

|  |
| --- |
| sudo chmod -R g+r /opt/tomcat/apache-tomcat-8.5.57/conf/  sudo chmod g+X /opt/tomcat/apache-tomcat-8.5.57/conf |

1. Change the ownership for required directories as a tomcat user using below command.

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

1. After completing all the above steps, create a tomcat as a service file for service configuration and paste the below contents

Note: You have to provide the exact installation path if you have changed your installation path directory on above commands. Make sure you set a correct path before restarting the service.

|  |
| --- |
| Insert the below contents  [Unit]  Description=Apache Tomcat Web Application Container  After=network.target  [Service]  Type=forking  Environment=JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64  Environment=CATALINA\_PID=/opt/tomcat/apache-tomcat-8.5.55/temp/tomcat.pid  Environment=CATALINA\_HOME=/opt/tomcat/apache-tomcat-8.5.57  Environment=CATALINA\_BASE=/opt/tomcat/apache-tomcat-8.5.57  Environment='CATALINA\_OPTS=-Xms1024M -Xmx2048M -server -XX:+UseParallelGC'  Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'  ExecStart=/opt/tomcat/apache-tomcat-8.5.57/bin/startup.sh  ExecStop=/opt/tomcat/apache-tomcat-8.5.57/bin/shutdown.sh  User=tomcat  Group=tomcat  UMask=0007  RestartSec=10  Restart=always  [Install]  WantedBy=multi-user.target |

1. Restart the service daemon and check the tomcat status. If tomcat is in-active, start it using the below command.

|  |
| --- |
| sudo systemctl daemon-reload  sudo systemctl start tomcat  sudo systemctl status tomcat |

1. Once the Tomcat is up, enable the service using below command to start it automatically whenever the system boots up.

|  |
| --- |
| sudo systemctl enable tomcat |

# REMOVE DEFAULT APPS

# Tomcat comes with some default applications by default inside webapps directory(/opt/tomcat/apache-tomcat-8.5.57/webapps), so we can remove those applications from this folder using below commands if we do not want to use them.

Eg: Apps names are Root,Docs,Examples,Manager and host-manager.

|  |
| --- |
| cd opt/tomcat/apache-tomcat-8.5.57/webapps  ls -l  rm -rf <app-name> or \*(if you want to delete all) |

# MEMORY SETTINGS

If any performance issues are happening on tomcat, then you might need to look up the memory size of tomcat first. By default, JVM allocated the heap memory size as 64 MB or 128 MB to tomcat, so if we want to increase or assign the memory size more for our applications, then we should define the required memory size in our tomcat.service file (Refer Step 7 file where env path has been set as CATALINA\_OPTS for memory).

Note: You must ensure that you are defining the size which is almost less than from your base system memory size.

Exa: You can set this environment on any startup script file, but defining it in the tomcat service file is much recommended. Refer the below command where you need to increase your size following by -server -XMS<Your-memory-size>

|  |
| --- |
| Environment='CATALINA\_OPTS=**-Xms1024M -Xmx2048M -server** |

Once changes are done on the service file and daemon reloaded, just verify it by accessing the tomcat on browser and checking the status where JVM memory will be displayed.

Note: Restarting the tomcat is must for all the changes to take effect.

Below snaps for your reference:



# 

# 

# 

# PORT CHANGE

All the Configuration files regarding tomcat should be located under the path where you downloaded and configured the tomcat. In this case it is /opt/tomcat/apache-tomcat-8.5.57/conf directory.

Find the server.xml file (where server network configuration has been configured) inside the conf directory and modify the port section as per below details.

By default, Tomcat runs on port 8080, and must change the port of tomcat if you’ve any other service already running on port 8080. Also make sure it doesn’t overlap with any other services.

|  |
| --- |
| **<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443"/>**  **<change the port as you like, eg: port=”80”>** |

# 

# 

# HTTPS CONFIGURATION

By default, it will redirect HTTP port so if we want to enable HTTPS then we need to modify the below lines on server.xml and add the required details to make a SSL/TLS redirect for all the web requests.

Uncomment the below line on server.xml and modify the same with additional SSL/TLS security details with certificates or keystore files.

|  |
| --- |
| **<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" maxThreads="150" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" keystoreFile=”<file-path>” keystorePass=”<keystorefile-password>”/>** |

Once configured the above lines on server.xml, restart the tomcat and verify the change via browser using this URL [**https://localhost:8443**](https://localhost:8443)

You can still access it via the same port as well[**https://localhost:8443**](https://localhost:8443)

Once the above step is done and if we want to enforce only HTTPS requests should come to our application via tomcat, then we should need to add the below lines into the web.xml file.

Note: Add following before </web-app> syntax

|  |
| --- |
| <security-constraint>  <web-resource-collection>  <web-resource-name>**name-you-want-to-give**</web-resource-name>  <url-pattern>/\*</url-pattern>  </web-resource-collection>  <user-data-constraint>  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  </user-data-constraint>  </security-constraint> |

# 