

**INFOPRO SOLUTIONS SDN BHD,**

**Malaysia**

**Tomcat Installation Guide – Training**

|  |  |
| --- | --- |
| **Document Name:** | Tomcat Installation Guide – Training |
| **Document Version No.:** | 2.0 |
| **Document Ownership:** | INFOPRO SOLUTIONS SDN BHD |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Amendments** | **Updated By** | **Date** |
| 1.0 | Tomcat Installation Guide | Izzuddin | 31/05/2022 |
| 2.0 | Update Tomcat Security Configuration | Farah Atiqah | 17/10/2022 |
| 3.0 | Guide Enhancement | Bryan | 16/02/2023 |
| 4.0  5.0  6.0 | Restrict Root User for Startup and Shutdown  Archive and Cron job  Update JDBC Configuration | Shaheeda  Aishah Fatihah  Bryan | 05/04/2023  18/03/2024  07/06/2024 |

Table of Content

[Revision History 2](#_Toc162352225)

[Table of Content 3](#_Toc162352226)

[1.0 FILE REQUIREMENT 5](#_Toc162352227)

[1.1 File needed 5](#_Toc162352228)

[2.0 TOMCAT INSTALLATIONS 5](#_Toc162352229)

[2.1 Create user 5](#_Toc162352230)

[2.2 Tomcat installation 7](#_Toc162352231)

[2.3 Libre Office Installation 8](#_Toc162352232)

[2.4 Java installation 10](#_Toc162352233)

[3.0 TOMCAT CONFIGURATIONS 12](#_Toc162352234)

[3.1 Change owner in directory tomcat. 12](#_Toc162352235)

[3.2 Tomcat Security Configurations 12](#_Toc162352236)

[3.2.1 Change port (optional) 12](#_Toc162352237)

[3.2.2 Shutdown port (optional) 13](#_Toc162352238)

[3.2.3 Disable Firewall and SELinux 14](#_Toc162352239)

[3.2.4 Start Server and Access the Web Interface 15](#_Toc162352240)

[4.0 RESTRICT ROOT USER FOR STARTUP AND SHUTDOWN 15](#_Toc162352241)

[5.0 SECURITY CONFIGURATION 17](#_Toc162352242)

[5.1 Remove localhost\_access\_log logging 17](#_Toc162352243)

[5.2 Remove Software Version Numbers 19](#_Toc162352244)

[5.3 HTTPS setup 19](#_Toc162352245)

[5.4 Logging setup 24](#_Toc162352246)

[5.5 Secure and HttpOnly Flag to Cookie 25](#_Toc162352247)

[5.6 Replace Default 404,403,500 page 29](#_Toc162352248)

[5.7 HTTP Security Configuration 31](#_Toc162352249)

[5.8 Set SameSite Cookie Flag 33](#_Toc162352250)

[5.9 Remove Default/Unwanted Applications 34](#_Toc162352251)

[5.10 JDBC configuration 36](#_Toc162352252)

[5.11 Password Encryption 37](#_Toc162352253)

[5.12 Archive and Cron Job configuration 39](#_Toc162352254)

[Add reboot Startup (OPTIONAL) 39](#_Toc162352255)

[Compress catalina.out file 39](#_Toc162352256)

[Archive application los in work/app\_home/logs 40](#_Toc162352257)

[Remove older logs in archive and temp folder 40](#_Toc162352258)

[5.13 CORS Security Configuration (Optional) 41](#_Toc162352259)

[5.14 Synergy Configuration (Optional) 42](#_Toc162352260)

[POST INSTALLATION 44](#_Toc162352261)

[Change owner and permission for your Tomcat user 44](#_Toc162352262)

[Restart Tomcat 44](#_Toc162352263)

# FILE REQUIREMENT

## File needed

|  |  |
| --- | --- |
| apache-tomcat-9.0.58.zip | I:\ICBA-DOC\INFRA\Installation Guide\SYNERGi Installer\Tomcat 9 RHEL8 |
| libreoffice7.0.tar | I:\ICBA-DOC\INFRA\Installation Guide\SYNERGi Installer\Tomcat 9 RHEL8 |
| JDK Installer | I:\ICBA-DOC\INFRA\Installation Guide\java installer |
| ssl folder | I:\ICBA-DOC\INFRA\Installation Guide\SYNERGi Installer\Tomcat 9 RHEL8 |
| jdbc folder | I:\ICBA-DOC\INFRA\Installation Guide\SYNERGi Installer\Tomcat 9 RHEL8 |
| encryption folder | I:\ICBA-DOC\INFRA\Installation Guide\SYNERGi Installer\Tomcat 9 RHEL8 |
| archivelog.sh | I:\ICBA-DOC\INFRA\Installation Guide\SYNERGi Installer\Tomcat 9 RHEL8\bin |
| remove\_older\_logs.sh | I:\ICBA-DOC\INFRA\Installation Guide\SYNERGi Installer\Tomcat 9 RHEL8\bin |

# TOMCAT INSTALLATIONS

## Create user

Write this command below using putty in winSCP to create a separate non-privileged user for Tomcat to protect other services running in case of any of account get compromised. Now we create username ‘tomcat\_din’

|  |
| --- |
| sudo useradd -r -m -U -d /home/tomcat\_danish -s /bin/bash tomcat\_danish |

**Example:**



Command above will create user named tomcat\_din and group with home directory /home/tomcat\_din. Run the command as root user (super user).

**Explanation the command above:**

**sudo: -** Allows user to execute the following command with administrative privileges.

* It is commonly used to perform tasks that require root or super user access.

**useradd: -** Used to create a new user account on the system.

**-r: -** This option indicates that the user being created will be a system account.

- System accounts are typically used for running services or daemons on the system, rather than for regular user interactions.

**-m: -** This option specifies that a home directory should be created for the new user. In this case, the home directory will be locatedat **/home/tomcat\_din**.

**-U: -** This flag instructs the command to create a new group with the same name as the user. So, in this case, a group named “tomcat\_din” will also be created.

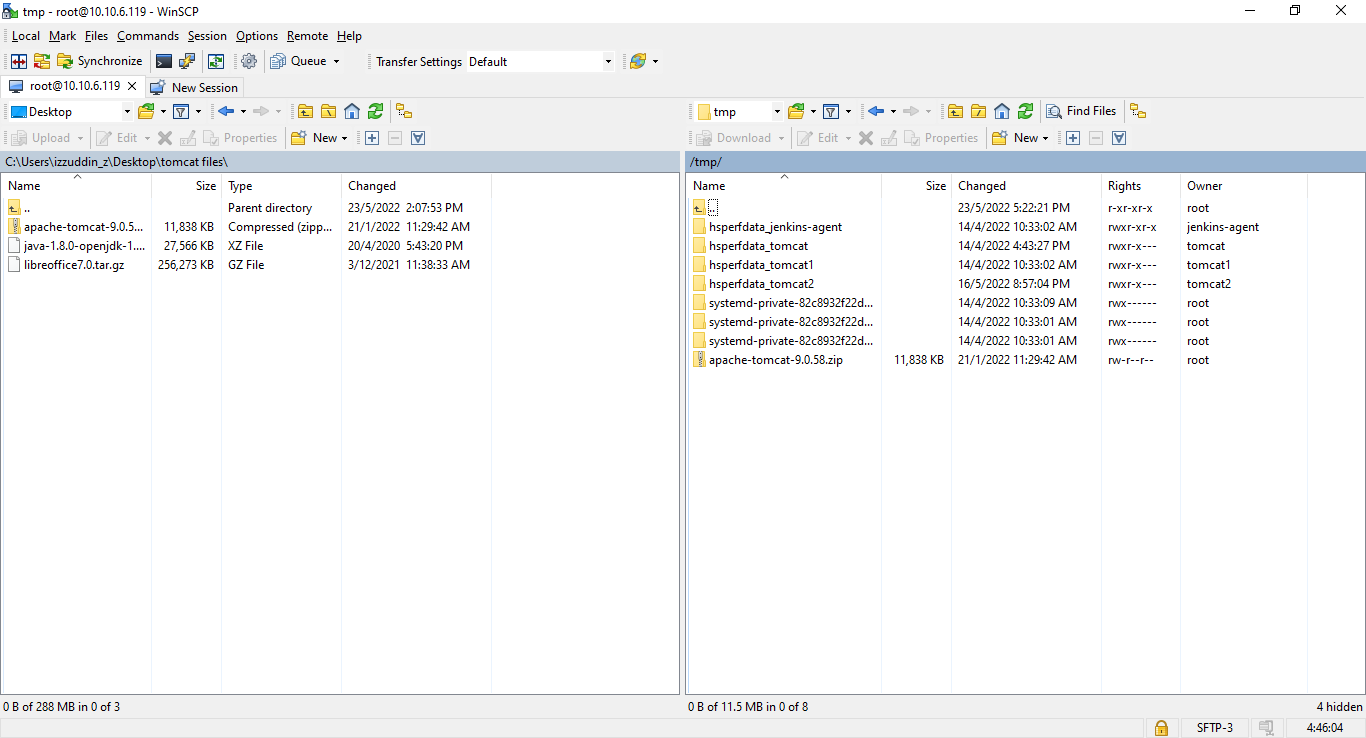
**-d /home/tomcat\_din: -** This option sets the home directory for the new user to **/home/tomcat\_din**.

**-s /bin/bash: -** This option specifies the login shell for the new user. In this case, the login shell will be set to **/bin/bash**, which is the Bash shell.

**tomcat\_din: -** This is the username for the new user account.

## Tomcat installation

Upload apache-tomcat-9.0.58.zip in /tmp directory. (Check 1.1 for the zip file)



Go to directory tmp:

|  |
| --- |
| cd /tmp |

**Example:**



And unzip the tomcat file to tomcat directory using this command

|  |
| --- |
| unzip /tmp/apache-tomcat-9.0.58.zip -d /home/tomcat\_din/ |

**Example:**



\*Please note that the tomcat name should be the one you have created for your tomcat

**unzip: -** Used to extract files from a ZIP archive.

**/tmp/apache-tomcat-9.0.58.zip: -** This is the path to the ZIP archive file that you want extract. In this case, the ZIP file is located at **/tmp/apache-tomcat-9.0.58.zip**. Adjust the path and filename accordingly if your ZIP file has a different name or location.

**-d /home/tomcat\_din/: -** This option specifies the destination directory where the contents of the ZIP file will be extracted. In this case, the destination directory is set to **/home/tomcat\_din/**. Adjust the path if you want to extract the files to a different directory.

**Note: The command above is often used to install or deploy software packages that are distributed in a ZIP format, such as the Apache Tomcat web server.**

If your apache-tomcat file is using format tar.gz, use the following command tu un-tar the gz file

|  |
| --- |
| tar xvzf apache-tomcat-9.0.58.tar.gz -C /home/tomcat\_din/ |

**tar: -** Used for manipulating tarball archive files.

**x: -** This option tells tar to extract the contents of the archive.

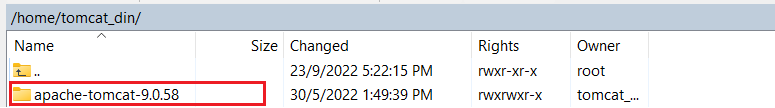
**v: -** This option enables verbose output, displaying the files as they are extracted.

**z: -** This option specifies that the archive file is in gzip compressed format.

**f apache-tomcat-9.0.58.tar.gz: -** This option specifies the name of the tarball archive file you want to extract. In this case, the archive file is named **apache-tomcat-9.0.58.tra.gz**. Adjust the filename if your tarball has a different name.

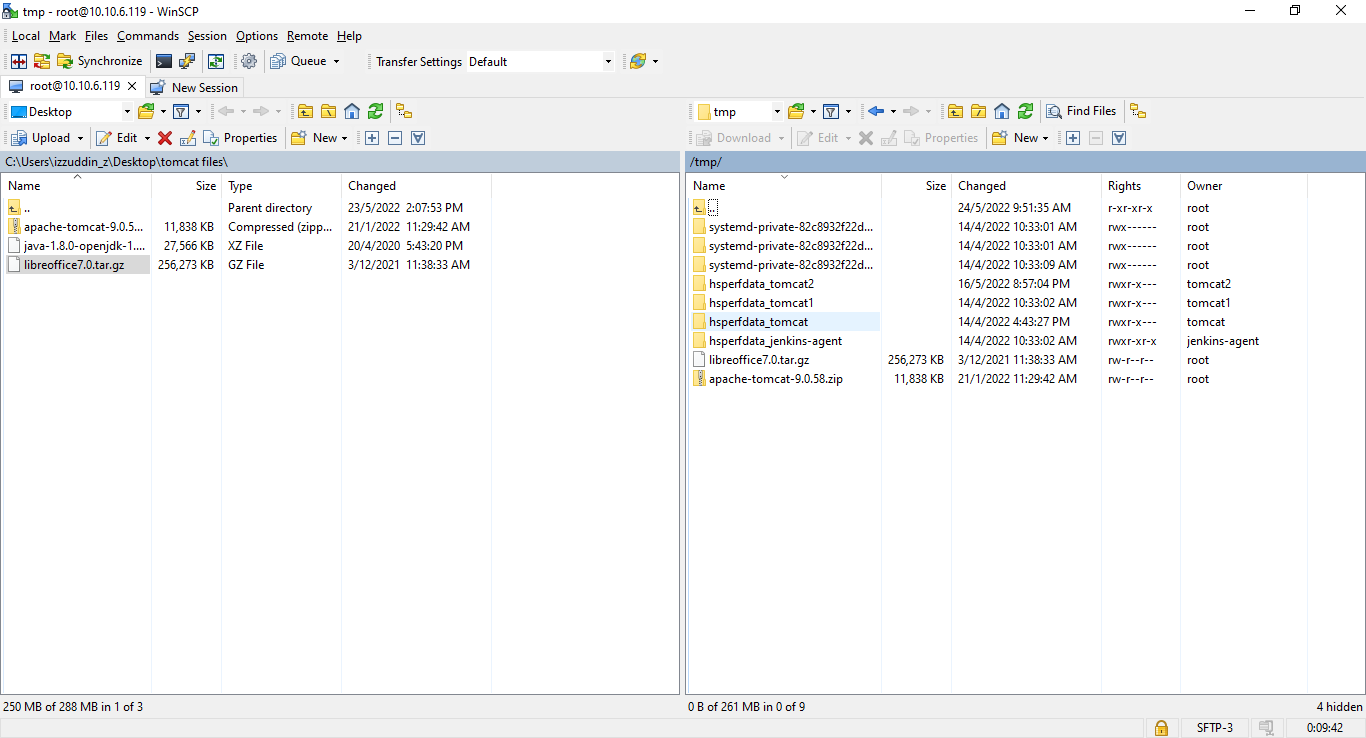
**-C /home/tomcat\_din/: -**This option specifies the destination directory where the contents of the tarball will be extracted. In this case, the destination directory is set to **/home/tomcat\_din**. Adjust the path if you want to extract the files to a different directory.

Once done, you will see the unzipped file in your tomcat directory.



## Libre Office Installation

Upload libre office in directory /tmp (Check 1.1 for the zip file)



Un-tar the libreoffice.tar to tomcat directory using this command.

|  |
| --- |
| tar xvzf /tmp/libreoffice7.0.tar.gz -C /home/tomcat\_din |

**Example:**



**tar: -** Used for manipulating tarball archive files.

**x: -** This option tells tar to extract the contents of the archive.

**v: -** This option enables verbose output, displaying the files as they are extracted.

**z: -** This option specifies that the archive file is in gzip compressed format.

**f /tmp/libreoffice7.0.tar.gz: -** This option specifies the name and location of the tarball archive file you want to extract. In this case, the archive file is located at **/tmp/libreoffice7.0.tar.gz**. Adjust the filename and path if your tarball has a different name or location.

**-C /home/tomcat\_din: -** This option specifies the destination directory where the contents of the tarball will be extracted. In this case, the destination directory is set to **/home/tomcat\_din**. Adjust the path if you want to extract the files to a different directory.

## Java installation

Navigate to **home/tomcat\_din**

|  |
| --- |
| cd /home/tomcat\_din |

**Example:**



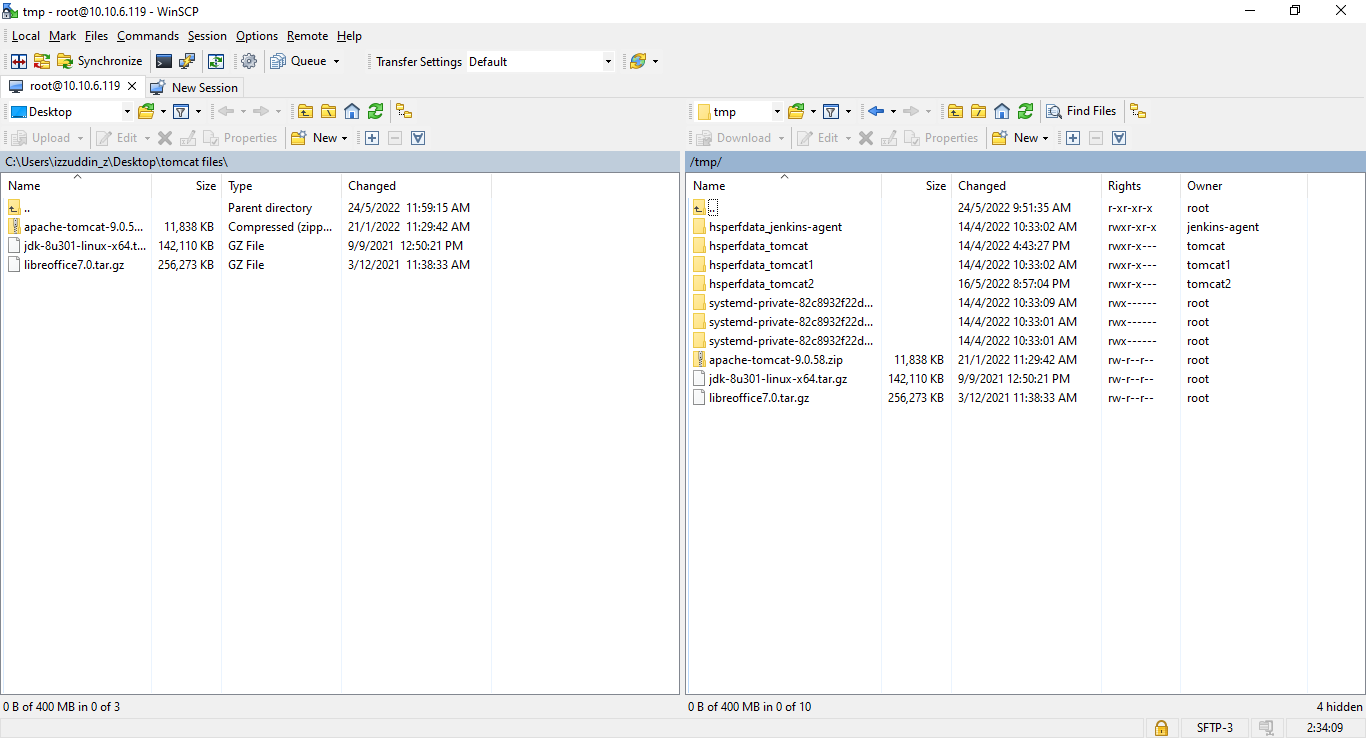
Make directory for java

|  |
| --- |
| mkdir java |

**Example:**



Upload JDK in /tmp directory (Check 1.1 for the zip file)



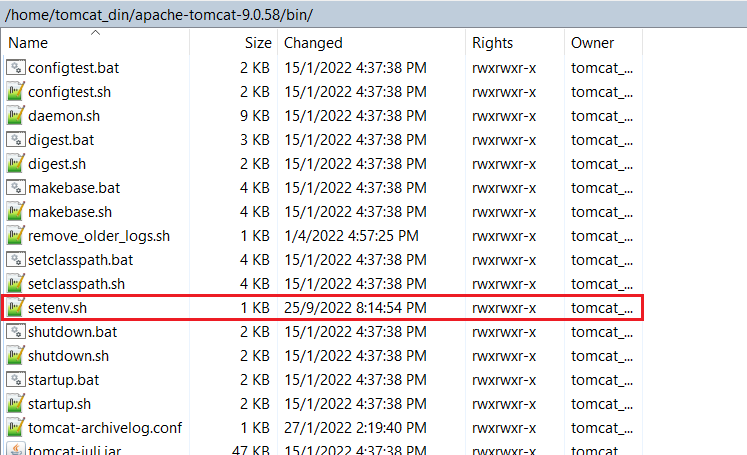
Un-tar the JDK in tomcat folder

|  |
| --- |
| tar xvzf /tmp/jdk-8u301-linux-x64.tar.gz -C home/tomcat\_din/java |

**Example:**



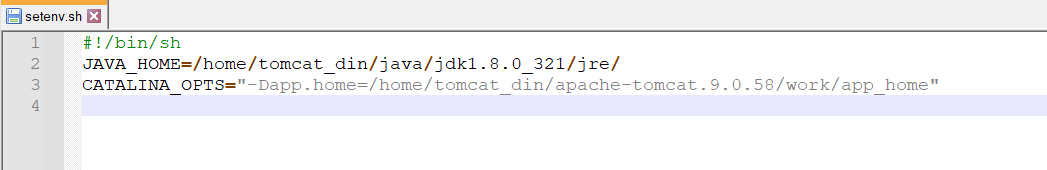
Go to **home/tomcat\_din/apache-tomcat-9.0.58/bin/** and create setenv.sh file.



Add this code in **setenv.sh**

|  |
| --- |
| #!/bin/sh  JAVA\_HOME=/home/tomcat\_din/java/jdk1.8.0\_321/jre/  CATALINA\_OPTS="-Dapp.home=/home/tomcat\_din/apache-tomcat-9.0.58/work/app\_home" |

**Example:**



**#!/bin/sh: -** This line is called a shebang and is used to specify the interpreter for the script. In this case, **#!/bin/sh** indicates that the script should be interpreted and executed using the Bourne shell (‘**sh**’).

**JAVA\_HOME=/home/tomcat\_din/java/jdk1.8.0\_321/jre/: -** This line assigns a value to the environment variable **‘JAVA\_HOME’**. The environment variable **‘JAVA\_HOME’** typically points to the installation directory of the Java Development Kit (JDK) or Java Runtime Environment (JRE). In this case, it is set to **/home/tomcat\_din/java/jdk1.8.0\_321/jre/**, which suggests that the script is configuring the Java environment for use within a specific directory.

**CATALINA\_OPTS="-Dapp.home=/home/tomcat\_din/apache-tomcat-9.0.58/work/app\_home": -** This line assigns a value to the environment variable **‘CATALINA\_OPTS’**. The variable **‘CATALINA\_OPTS’** is commonly used in the context of the Apache Tomcat web server to specify additional runtime options or system properties. In this case, it sets the **-Dapp.home** system property to **/home/tomcat\_din/apache-tomcat-9.0.58/work/app-home**.

# TOMCAT CONFIGURATIONS

## Change owner in directory tomcat.

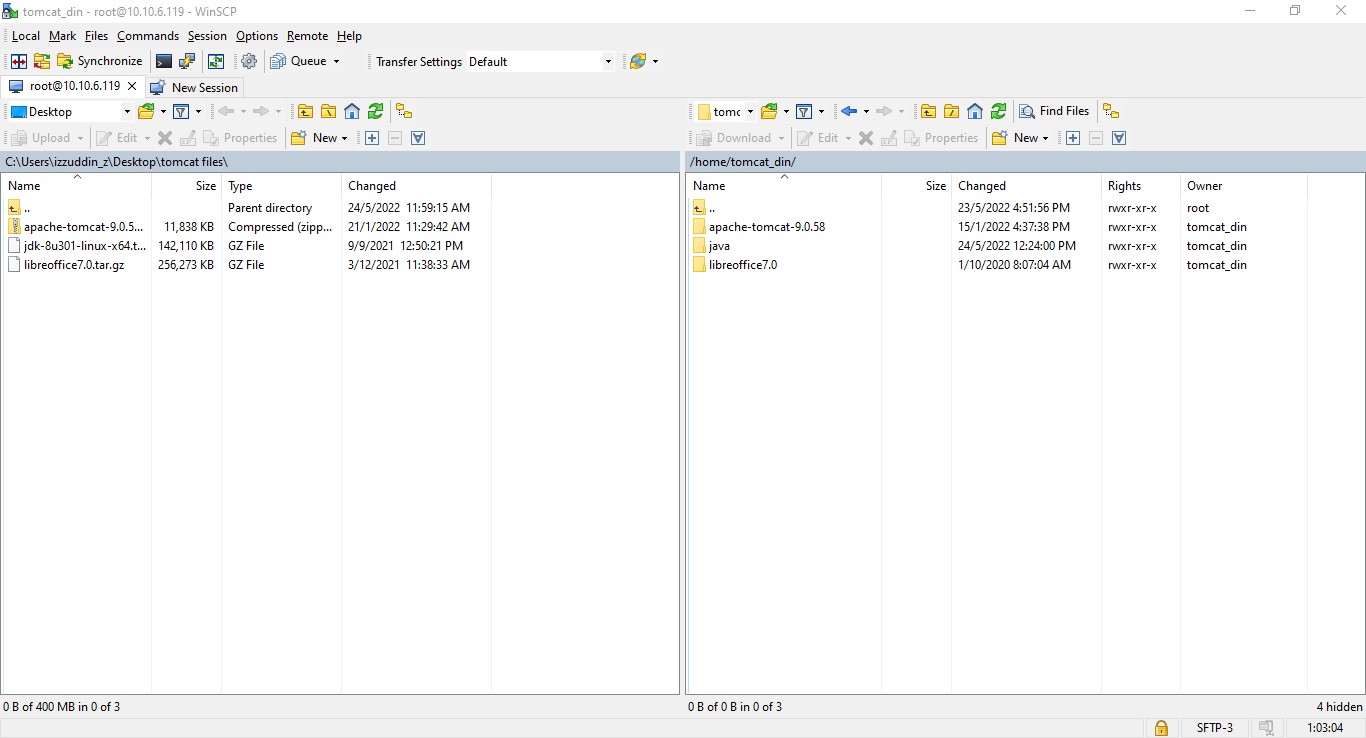
Change owner need to because only owner can access the directory. Run in putty.

|  |
| --- |
| chown -R tomcat\_din:tomcat\_din /home/tomcat\_din |

**Example:**



Then the owner will change to tomcat\_din.



## Tomcat Security Configurations

### 3.2.1 Change port (optional)

**\*Only apply if there are multiple tomcat in one server**

**\*Do not apply this in project. Only apply this for training.**

By default, the port will be 8080, to change it, open the server.xml file in **/conf**. In **server.xml**, find a statement like the following:

|  |
| --- |
| <Connector port="8080" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" /> |

Change **8080** to another number like **8281**:

|  |
| --- |
| <Connector port="8281" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" /> |

Then, save the file.

**port=8080 🡪** This attribute specifies the port number on which the web server will listen for incoming HTTP requests. In this case, it is set to port 8080.

**protocol=”HTTP/1.1” 🡪** This attribute specifies the protocol to be used for communication. Here, it indicates the HTTP version 11.

**connectionTimeout=”20000” 🡪** This attribute sets the maximum time, in milliseconds, that the server will wait for a client to send a complete request. If the client does not send the request within this timeout period, the server may close the connection.

**redirectPort=”8443” 🡪** This attribute specifies the port number to which the server should redirect clients if they attempt to access a secure (HTTPS) resource using an insecure (HTTP) connection. Here, it is set to port 8443, which is commonly used for secure information.

**Note: This configuration suggests that the web server is set up to handle HTTP requests on port 8080, using the HTTP/1.1 protocol. It has a connection timeout of 20000 milliseconds (20 seconds) and will redirect clients to port 8443 if they attempt to access a secure resource using an insecure connection.**

**For example:**

<Connector port="8185" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

### 3.2.2 Shutdown port (optional)

\*Only apply if there are **multiple tomcat in one server**

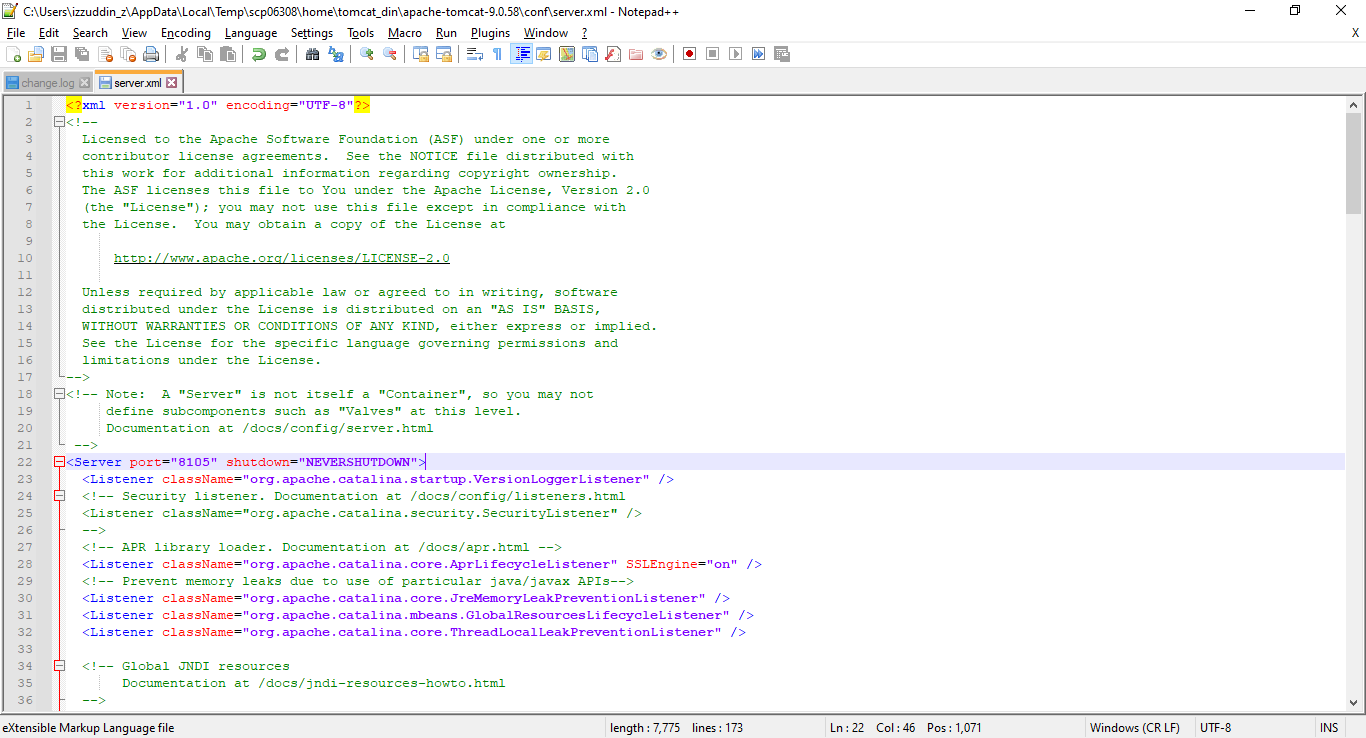
Shutdown port for each tomcat. Default port number will be 8005. Make sure every tomcat has different shutdown port. Go to **server.xml** file in **/conf** and change the configuration.

Search this port in server.xml

|  |
| --- |
| <Server port="8005" shutdown="SHUTDOWN"> |

Change 8005 to another number like 8105, please double check port offset before we execute. Below is example after change the port.

|  |
| --- |
| <Server port="8105" shutdown="NEVERSHUTDOWN"> |



Then, save the file.

### 3.2.3 Disable Firewall and SELinux

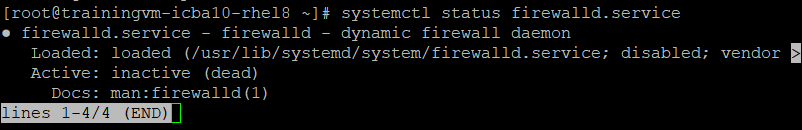
Before starting the server, make sure to **disable firewall and SELinux**.

To disable firewall, run the following command and make sure to run the command as “root” user.

|  |
| --- |
| systemctl stop firewalld.service |

To check if the firewall has been stopped, run the command below to check its status.

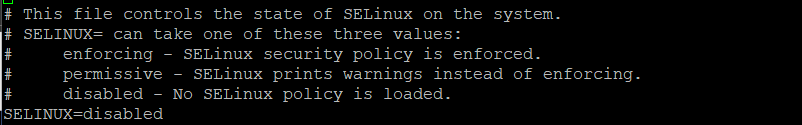
|  |
| --- |
| systemctl status firewalld.service |



Once done, go to **/etc/selinux** directory and edit /config file using vi editor. Run command below in Putty

|  |
| --- |
| vi /etc/selinux/config |

Press “**i**”to edit and change “**SELINUX=enforcing**” to “**SELINUX=disabled**”



Press “**esc**” button and enter “**:wq**” to save file.

### 3.2.4 Start Server and Access the Web Interface

To start the server, log in into your tomcat user. Use the command below

|  |
| --- |
| su – tomcat\_din |

Then, go into your bin directory

|  |
| --- |
| cd /home/tomcat\_din/apache-tomcat-9.0.58/bin |

Start your tomcat server as per below command

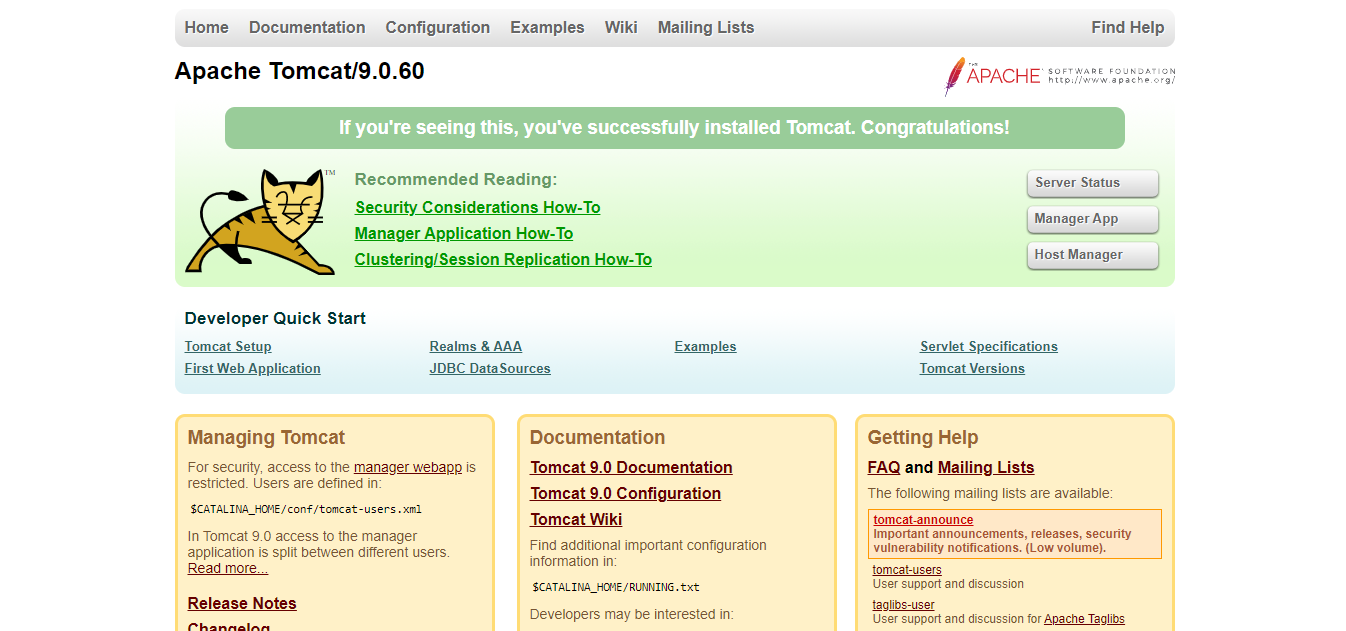
|  |
| --- |
| ./startup.sh |

Then, open your browser and enter your IP address and the port you have set up for your tomcat.

**Example**:

**http://10.10.6.119:8281/**

After completing this, open browser and you will see like this.



\*Note that all of the red-colored words may change based on your configuration. Please re-check before continuing

# RESTRICT ROOT USER FOR STARTUP AND SHUTDOWN

Go to home/tomcat/apache-tomcat-9.0.68/bin/ and search for **startup.sh**.

Add this code in **startup.sh** and put in the first lineto restrict startup to the tomcat user.

\*Note: The code is put in first line to ensure that the script does not modify or make changes in any files or directories.

|  |
| --- |
| if [ "$(id -u)" = "0" ]; then  echo "This script must not be run as root. Please run the script as the assigned tomcat user." 1>&2  exit 1  fi |

**if [ "$(id -u)" = "0" ]; then 🡪** This line starts an if statement that checks if the result of the command **‘id -u’** is equal to the string “0”. The **‘id -u’** command returns the user ID of the current user.

**echo "This script must not be run as root. Please run the script as the assigned tomcat user." 1>&2 🡪** If the condition in the if statement evaluates to true (i.e., the script is being run as the root user with user ID 0), this line will be executed. It prints the specified error message to the standard error stream using the **‘echo’** command. The **‘1>&2’** part redirects the output of **‘echo’** to the standard error stream.

**exit 1 🡪** This line terminates the script and exits with a non-zero status code (in this case, 1). The non-zero status code indicates an error condition to the calling process or script.

**Note: The code above checks if the script is being run as the root user, and if so, it prints an error message to the standard error stream and exits with a non-zero status code. This is used to ensure that the script is not executed with root privileges, but rather with a specific non-root user, such as assigned “tomcat” user.**

Do the same for **shutdown.sh**. Go to home/tomcat/apache-tomcat-9.0.68/bin/ and search for **shutdown.sh**.

Add this code in **shutdown.sh** and put in the first line.

|  |
| --- |
| if [ "$(id -u)" = "0" ]; then  echo "This script must not be run as root. Please run the script as the assigned tomcat user." 1>&2  exit 1  fi |

**🡪The code above is used to prevent the script from being executed with root privileges. If it detects that the script is being run as the root user, it displays an error message and exits with a non-zero status code to indicate the error. This helps ensure that the script is executed with the appropriate user privileges, such as the assigned “tomcat” user in this case.**

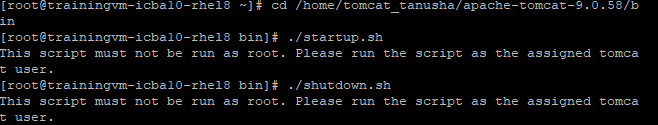
Once done, save the file. Run this command (in Putty) as root and go into your bin directory.

|  |
| --- |
| cd /home/tomcat/apache-tomcat-9.0.68/bin |

Restart your tomcat server as per below command.

|  |
| --- |
| ./startup.sh  ./shutdown.sh |

When running the user as root, make sure there will be message display that restricts the root user.



\*Note: The command will prompt an error message if it is run with root access. This is to ensure that the command is being run with the correct permission.

# SECURITY CONFIGURATION

## Remove localhost\_access\_log logging

Go to **server.xml** file in /home/tomcat\_din/apache-tomcat-9.0.58/conf and **comment out** the below configuration

|  |
| --- |
| <!--  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="localhost\_access\_log" suffix=".txt"  pattern="%h %l %u %t &quot;%r&quot; %s %b" />  --> |

**className="org.apache.catalina.valves.AccessLogValve" 🡪** This attribute specifies the Java class that implements the access log valve functionality. In this case, the **‘AccessLogValve’** class from the **‘org.apache.catalina.valves’** package is used.

**directory=”logs” 🡪** This attribute defines the directory where the access logs will be stored. In this case, the logs will be saved in the “logs” directory.

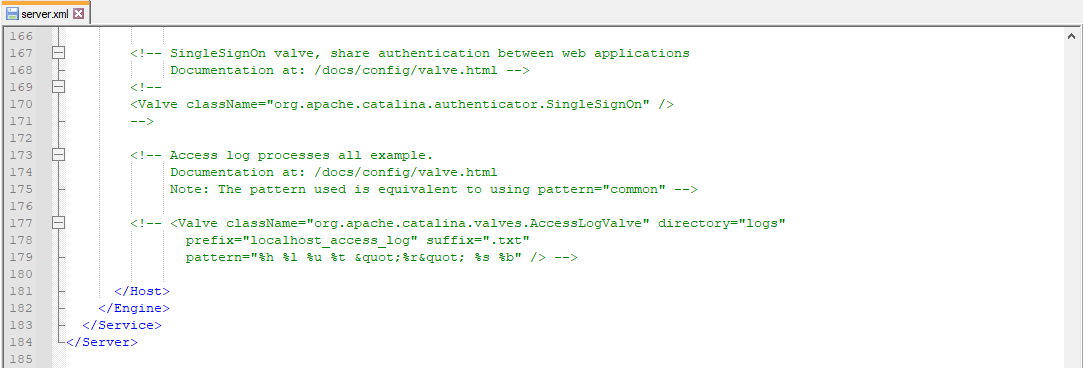
**prefix=”localhost\_access\_log” 🡪** This attribute sets the prefix for the access log file names. The log files will be named starting with “localhost\_access\_log” followed by a date and time stamp.

**suffix=”.txt” 🡪** This attribute specifies the suffix for the access log file names. In this case, the log files will have the “.txt” extension.

**pattern="%h %l %u %t &quot;%r&quot; %s %b" 🡪** This attribute sets the format pattern for the access log entries. The **‘%h’, ‘%l’, ‘%u’, ‘%t’, ‘%r’, ‘%s’,** and **‘%b’** are placeholders for various log entry components:

* **‘%h’:** Remote host name or IP address.
* **‘%l’:** Remote logical username (not implemented).
* **‘%u’:** Authenticated user (if any), or ‘-‘ if not authenticated.
* **‘%t’:** Date and time of the request is Common Log Format (CLF).
* **‘%r’:** First line of the request.
* **‘%s’:** HTTP status code of the response.
* **‘%b’:** Bytes sent, excluding HTTP headers.

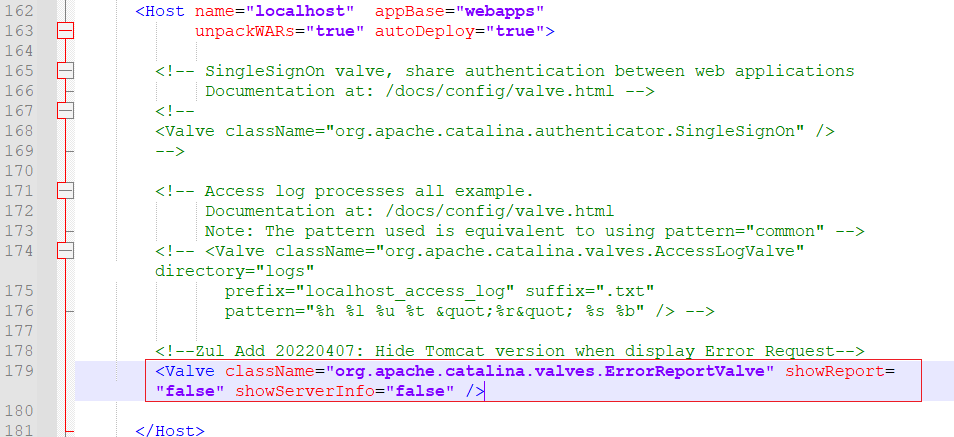
**Example:**



## Remove Software Version Numbers

Open **server.xml** in /home/tomcat\_din/apache-tomcat-9.0.58/ and add the following line before the end of Host tag **</Host>**

|  |
| --- |
| <Valve className="org.apache.catalina.valves.ErrorReportValve" showReport="false" showServerInfo="false" /> |



className="org.apache.catalina.valves.ErrorReportValve" 🡪 This attribute specifies the Java class that implements the error report valve functionality. In this case, the ‘ErrorReportValve’ class from the ‘org.apache.catalina.valves’ package is used.

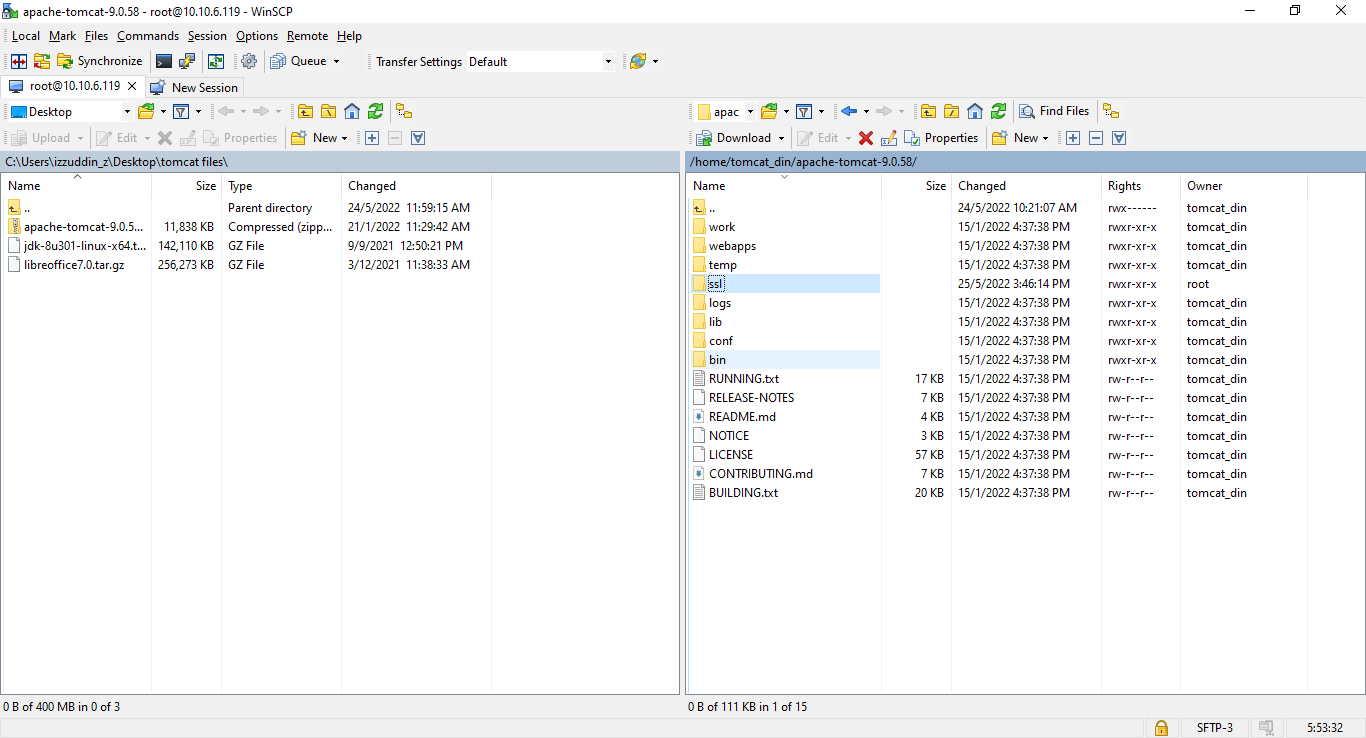
showReport="false" 🡪 This attribute controls whether detailed error reports should be displayed to the client. By setting it to “false”, detailed error reports will not be shown to the client.

showServerInfo="false" 🡪 This attribute determines whether server information, such as the server version and name, should be included in error reports. By setting it to “false”, server information will not be shown in error reports.

Note: The Error Report Valve is responsible for handling and processing error reports in Apache Tomcat. By configuring it with these attributes, the detailed error reports and server information are disabled, providing a more secure and controlled error handling behavior.

## HTTPS setup

Create **ssl** folder into your apache-tomcat directory, /home/tomcat\_din/apache-tomcat-9.0.58/



Generate keystore by using the command below:

|  |
| --- |
| keytool -genkey -alias tomcat -keyalg RSA |

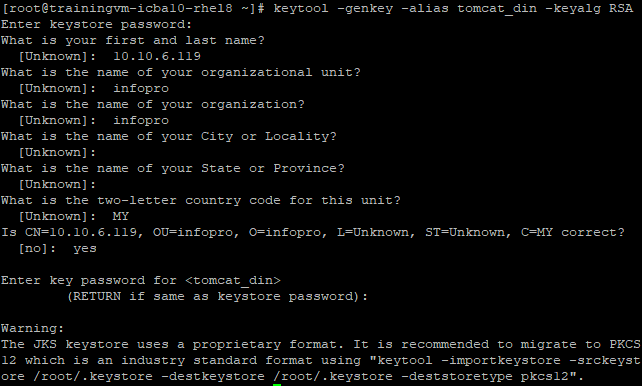
Enter ‘infopro’ for the keystore password

**keytool 🡪** This is command-line tool provided by Java for managing certificates and keystores.

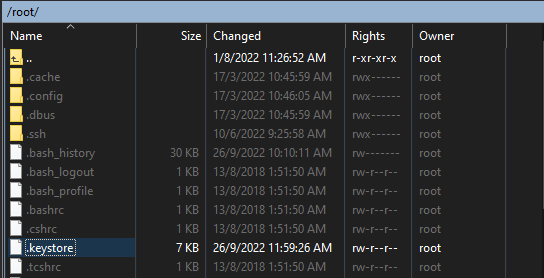
**-genkey 🡪** This option instructs keytool to generate a new key pair.

**-alias tomcat 🡪** This specifies the alias to be used for the generated key pair. In this case, the alias is set to “tomcat”.

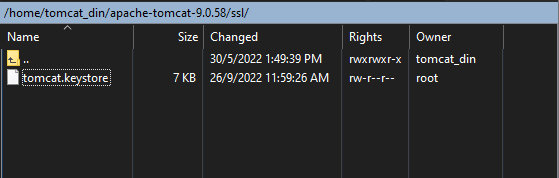
**-keyalg RSA 🡪** The specifies the key algorithm to be used for generating the key pair. RSA (Rivest-Shamir-Adleman) is a widely used asymmetric encryption algorithm. By specifying **‘-keyalg RSA’**, keytool will generate an RSA key pair.



You can find the generated **keystore** file in /root.



Copy the keystore file and paste it into the ssl folder you have just created in your tomcat directory. Rename the file from **‘.keystore’** to **‘tomcat.keystore’**



And then, go to **server.xml** file in /conf directory and add the below configuration.

|  |
| --- |
| <Connector port="8543" protocol="HTTP/1.1"  connectionTimeout="20000"  Server =" " compression="on"  allowTrace="false"  SSLEnabled="true"  scheme="https" keystoreFile="/home/tomcat\_din/apache-tomcat-9.0.58/ssl/tomcat.keystore" keystorePass="infopro"  clientAuth="false" sslProtocols ="TLSv1,TLSv1.1,TLSv1.2" ciphers="TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384, TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384, TLS\_ECDH\_RSA\_WITH\_AES\_256\_GCM\_SHA384, TLS\_ECDH\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDH\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDH\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA,TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA,TLS\_ECDH\_RSA\_WITH\_AES\_256\_CBC\_SHA384,TLS\_ECDH\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384,TLS\_ECDH\_RSA\_WITH\_AES\_256\_CBC\_SHA, TLS\_ECDH\_ECDSA\_WITH\_AES\_256\_CBC\_SHA, TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256, TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256, TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA, TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA, TLS\_ECDH\_RSA\_WITH\_AES\_128\_CBC\_SHA256, TLS\_ECDH\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256, TLS\_ECDH\_RSA\_WITH\_AES\_128\_CBC\_SHA, TLS\_ECDH\_ECDSA\_WITH\_AES\_128\_CBC\_SHA" /> |

\*Note that you can set different port if there are multiple tomcats in the server

**port=”8543” 🡪** Specifies the port number on which the connector will listen for incoming requests.

**protocol=”HTTP/1.1” 🡪** Defines the protocol used for the connection. In this case, it is the HTTP 1.1 protocol.

**connectionTimeout=”20000” 🡪** Sets the maximum time, in milliseconds, that a connection can remain idle before it is closed.

**Server=” “ 🡪** Represents the server’s name. It appears to be an empty string in the provided snippet.

**compression=”on” 🡪** Enables compression of the HTTP responses to reduce the size of the data being transferred.

**allowTrace=”false” 🡪** Disallows the TRACE HTTP method, which can be used for cross-site scripting (XSS) attacks.

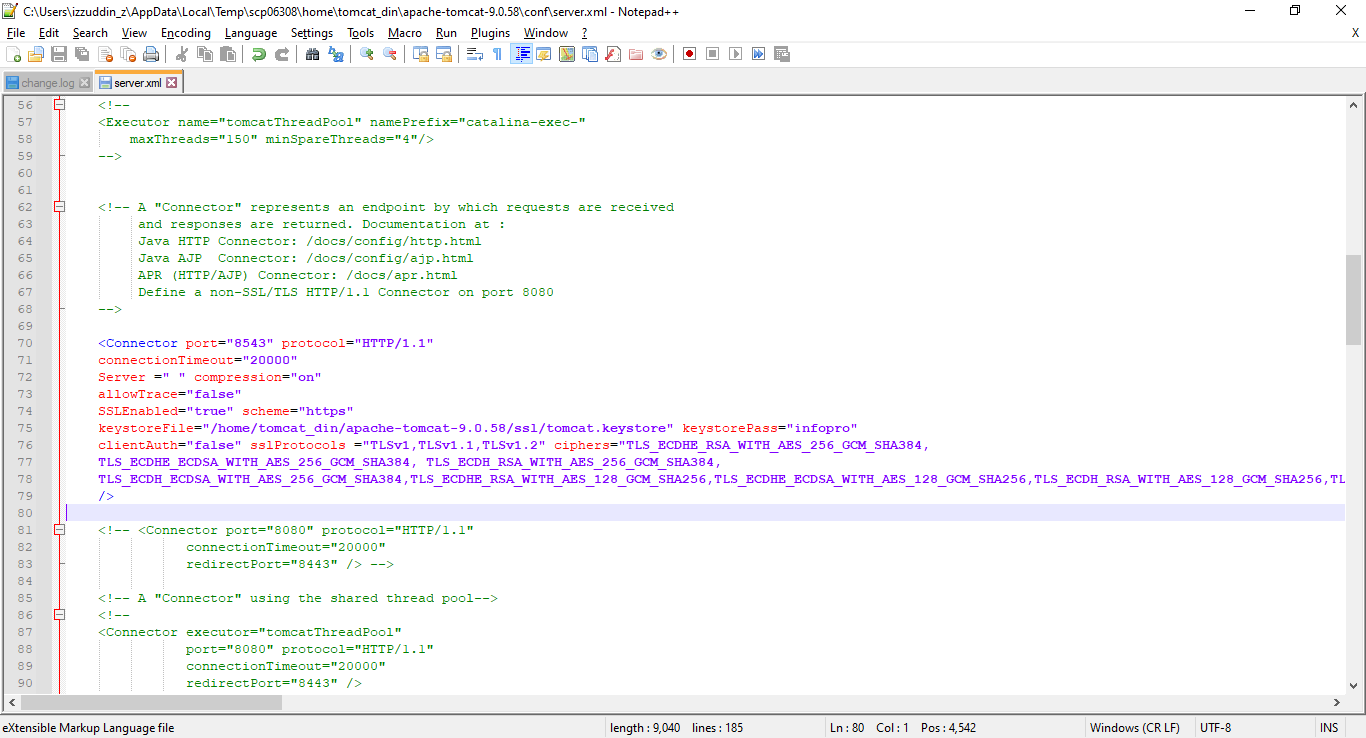
**SSLEnabled=”true” 🡪** Indicates that SSL/TLS is enabled for this connector, allowing HTTPS connections.

**scheme=”https” 🡪** Specifies the scheme used for the connector, which is “https” in this case.

**keystoreFile=”/home/tomcat\_din/apache-tomcat-9.0.58/ssl/tomcat.keystore” 🡪** Defines the path to the keystore file that holds the server’s SSL certificate and private key.

**keystorePass=”infopro” 🡪** Provides the password for the keystore.

**Example:**



Comment out the below configuration, in 2.7.1 where you have specified the port for **8281**

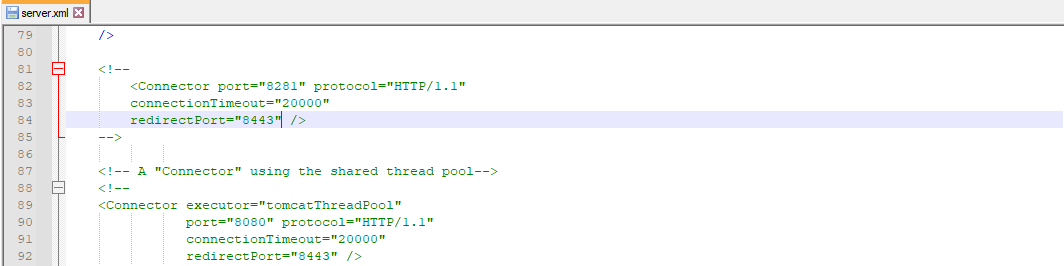
|  |
| --- |
| <!--  <Connector port="8281" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" />  --> |

**port=”8281” 🡪** Specifies the port number on which the connector will listen for incoming HTTP requests.

**protocol=”HTTP/1.1” 🡪** Defines the protocol used by the connector, indicating HTTP 1.1 in this case.

**connectionTimeout=”20000” 🡪** Sets the maximum time, in milliseconds, that a connection can remain idle before it is closed.

**redirectPort=”8443” 🡪** Specifies the port number to which the server should redirect the client if it attempts to access a secure resource using HTTP instead of HTTPS.



And then, change the owner, after implementing this line please refresh.

|  |
| --- |
| chown -R tomcat\_din:tomcat\_din /home/tomcat\_din |

**Example:**



## Logging setup

Go to **/conf** and open **logging.properties** file. Comment out 2localhost, 3manager, 4host-manager. Remove the dot “**.**” of the second handlers below the commented one.

|  |
| --- |
| #handlers = 1catalina.org.apache.juli.AsyncFileHandler, 2localhost.org.apache.juli.AsyncFileHandler, 3manager.org.apache.juli.AsyncFileHandler, 4host-manager.org.apache.juli.AsyncFileHandler, java.util.logging.ConsoleHandler  handlers = 1catalina.org.apache.juli.AsyncFileHandler, java.util.logging.ConsoleHandler |

**#handler 🡪** The pound sign (‘#’) at the beginning indicates that the line is commented out, meaning it is currently inactive. It is likely the original configuration line that has been commented out.

**handlers 🡪** This line specifies the active logging handlers to use, separated by commas.

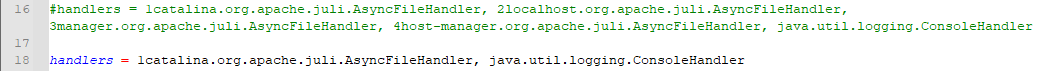
**In the commented-out line, there are multiple handlers specified:**

* **1catalina.org.apache.juli.AsyncFileHandler :-** It represents an asynchronous file logging handler for the “catalina” component in Apache Tomcat.
* **2localhost.org.apache.juli.AsyncFileHandler :-** It represents an asynchronous file logging handler for the “localhost” component in Apache Tomcat.
* **3manager.org.apache.juli.AsyncFileHandler :-** It represents an asynchronous file logging handler for the “manager” component in Apache Tomcat.
* **4host-manager.org.apache.juli.AsyncFileHandler :-** It represents an asynchronous file logging handler for the “host-manager” component in Apache Tomcat.
* **java.util.logging.ConsoleHandler :-** It represents a console logging handler provided by the Java standard library.

**In the active line, only two handlers are specified:**

* **1catalina.org.apache.juli.AsyncFileHandler :-** It is the asynchronous file logging handler for the “catalina” component.
* **java.util.logging.ConsoleHandler :-** It is console logging handler.

**Example:**



## Secure and HttpOnly Flag to Cookie

Prevent from steal or manipulate web applications session and cookies which is flag that injected in the response header.

* Apply code below by adding inside **<session-config>** tag in **web.xml** file
* Add following before **</web-app>** syntax for SSL and HTTP Method Allow

|  |
| --- |
| <session-config>  <cookie-config>  Session Configuration  <http-only>true</http-only>  <secure>true</secure>  </cookie-config>  <session-timeout>30</session-timeout>  </session-config>  <security-constraint>  <web-resource-collection>  Security Constraints  <web-resource-name>Protected Context</web-resource-name>  <url-pattern>/\*</url-pattern>  </web-resource-collection>  <user-data-constraint>  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  </user-data-constraint>  </security-constraint>  <security-constraint>  <web-resource-collection>  <web-resource-name>All Access</web-resource-name>  Additional Security Constraint  <url-pattern>/\*</url-pattern>  <http-method>OPTIONS</http-method>  <http-method>TRACE</http-method>  <http-method>TRACK</http-method>  </web-resource-collection>  <auth-constraint />  </security-constraint> |

**Session Configuration:**

* **<session-config> :-** Configures the session settings.
* **<cookie-config> :-** Configures the session cookie settings.
* **<http-only>true</http-only>:-** This element indicates that the session cookies should be accessible only through HTTP requests and not through client-side scripts.
* **<secure>true</secure>: -** Sets the ‘Secure’ attribute on the session cookie, ensuring that it is only transmitted over a secure (HTTPS) connection.
* **<session-timeout>30</session-timeout> :-** Specifies the session timeout duration in minutes. In this case, it is set to 30 minutes.

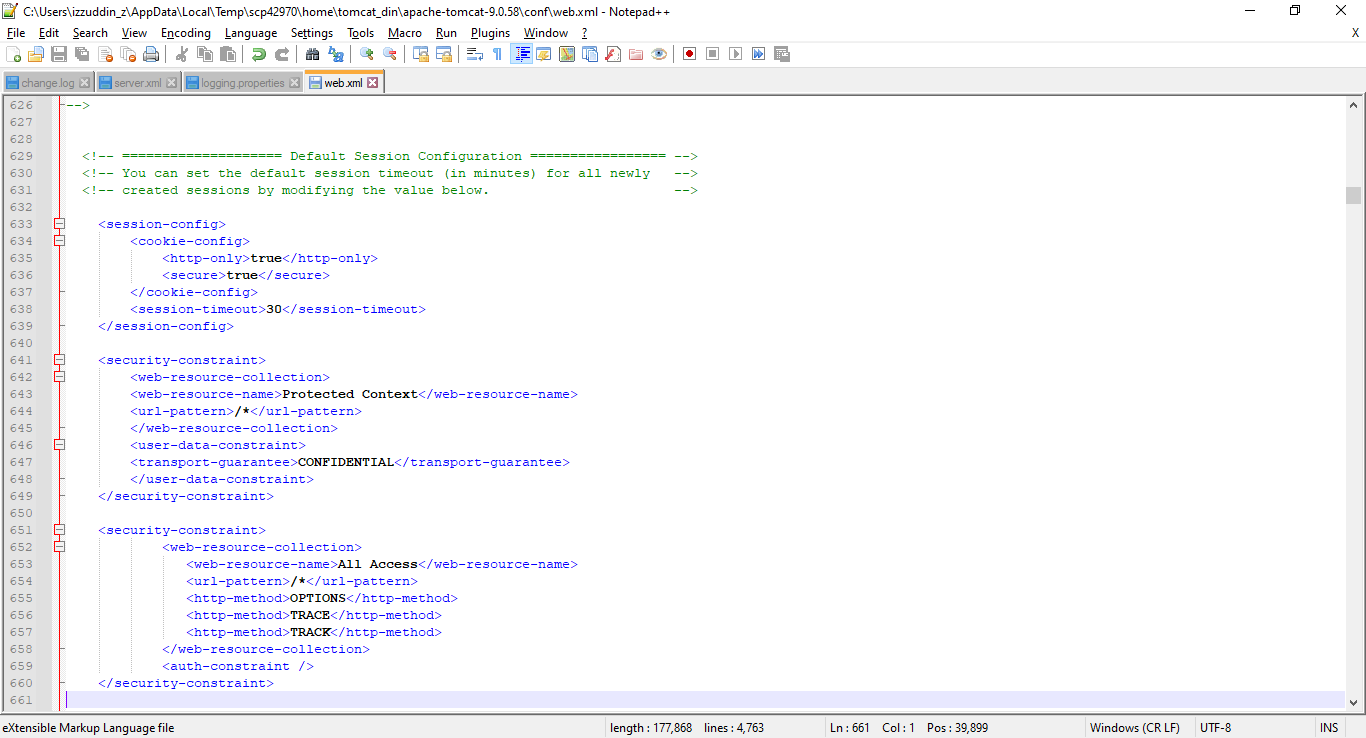
**Security Constraints:**

* **<security-constraint> :-** This element defines a security constraint for a protected context.
* **<web-resource-collection> :-** This element specifies the resources to which the security constraint applies.
* **<web-resource-name>Protected Context</web-resource-name> :-** This element provides a name for the protected resources.
* **<url-pattern>/\*</url-pattern> :-** This element specifies that the security constraint applies to all URLs within the protected context.
* **<user-data-constraint> :-** This element specifies the constraint for user data.
* **<transport-guarantee>CONFIDENTIAL</transport-guarantee> :-** This element indicates that the user data should be transported using a secure (HTTPS) connection.

**Additional Security Constraints:**

* This security constraint allows all access (no authentication constraint) to the specified URL pattern (/\*) but restricts certain HTTP methods.
* **<web-resource-name>All Access</web-resource-name> :-** This element provides a name for the resource with unrestricted access.
* **<url-pattern>/\*</url-pattern> :-** This element specifies that the security constraint applies to all URLs.
* **<http-method>OPTIONS</http-method>, <http-method>TRACE</http-method>, <http-method>TRACK</http-method> :-** These elements specify the HTTP methods (OPTIONS, TRACE, and TRACK) that are restricted for this URL pattern.
* **<auth-constraint /> :-** This element specifies that there is no authentication constraint for the resources with unrestricted access.

**Example:**



HTTP commands like PUT and DELETE are rejected – Disable

Then, apply code below by adding inside **<servlet>** tag in **web.xml** file:

|  |
| --- |
| <servlet>  <servlet-name>default</servlet-name>  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>  <init-param>  <param-name>debug</param-name>  <param-value>0</param-value>  </init-param>  <init-param>  <param-name>listings</param-name>  <param-value>false</param-value>  </init-param>  <init-param>  <param-name>readonly</param-name>  <param-value>true</param-value>  </init-param>  <load-on-startup>1</load-on-startup>  </servlet> |

**Explanation:**

**<servlet> 🡪** This element defines a servlet configuration.

**<servlet-name>default</default-name> 🡪** This element specifies the name of the servlet, in this case, “default”.

**<servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class> 🡪** This element specifies the Java class that implements the servlet, in this case, ” org.apache.catalina.servlets.DefaultServlet”.

**<init-param> 🡪** This element is used to provide initialization parameters (or configuration options) for the servlet.

**<param-name>debug</param-name> 🡪** This element specifies the name of the initialization parameter “debug”.

**<param-value>0</param-value> 🡪** This element specifies the value of the “debug” parameter, in this case, “0”.

**<param-name>listings</param-name> 🡪** This element specifies the name of the initialization parameters “listings”.

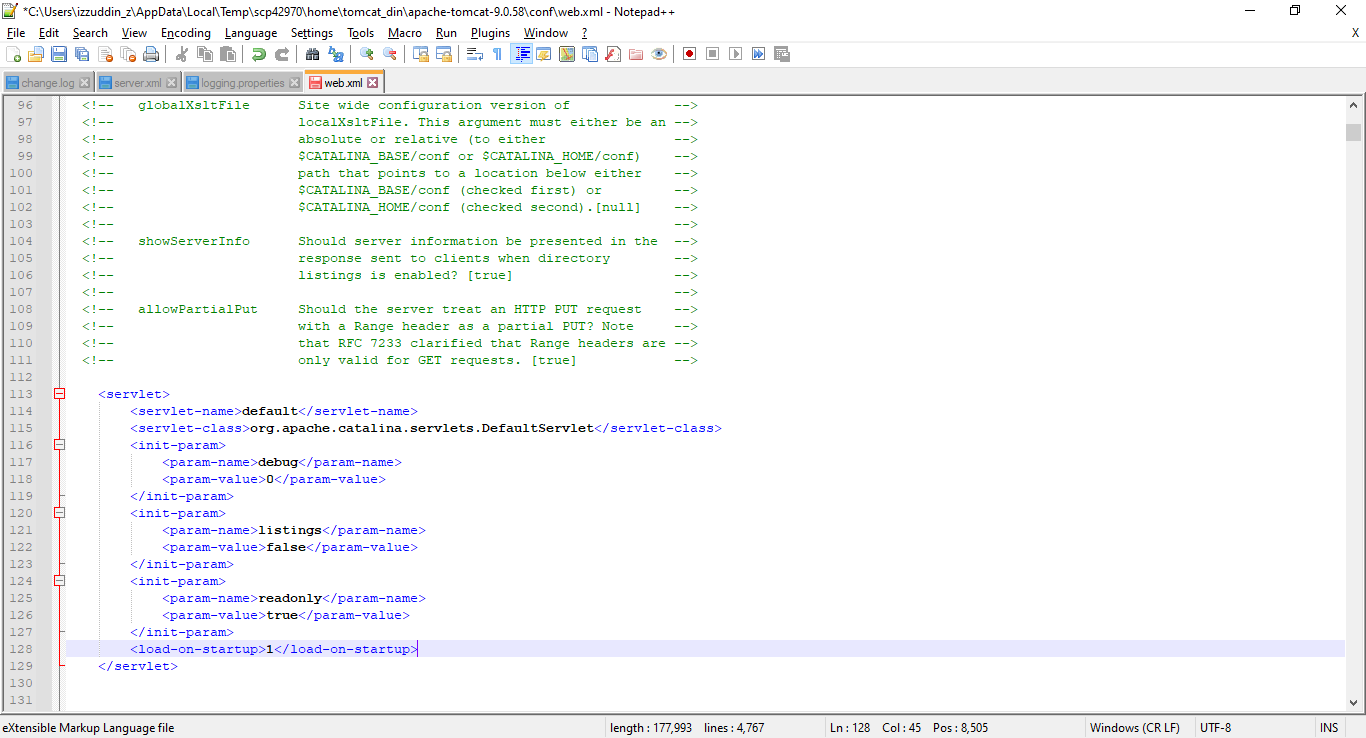
**<param-value>false</param-value> 🡪** This element specifies the value of the “listings” parameter, in this case, “false”.

**<param-name>readonly</param-name> 🡪** This element specifies the name of the initialization parameter “readonly”.

**<param-value>true</param-value> 🡪** This element specifies the value of the “readonly” parameter, in this case, “true”.

**<load-on-startup>1</load-on-startup> 🡪** This element specifies the order in which the servlet should be loaded during application startup. In this case, it has a value of 1, indicating that is should be loaded first.

**Example:**



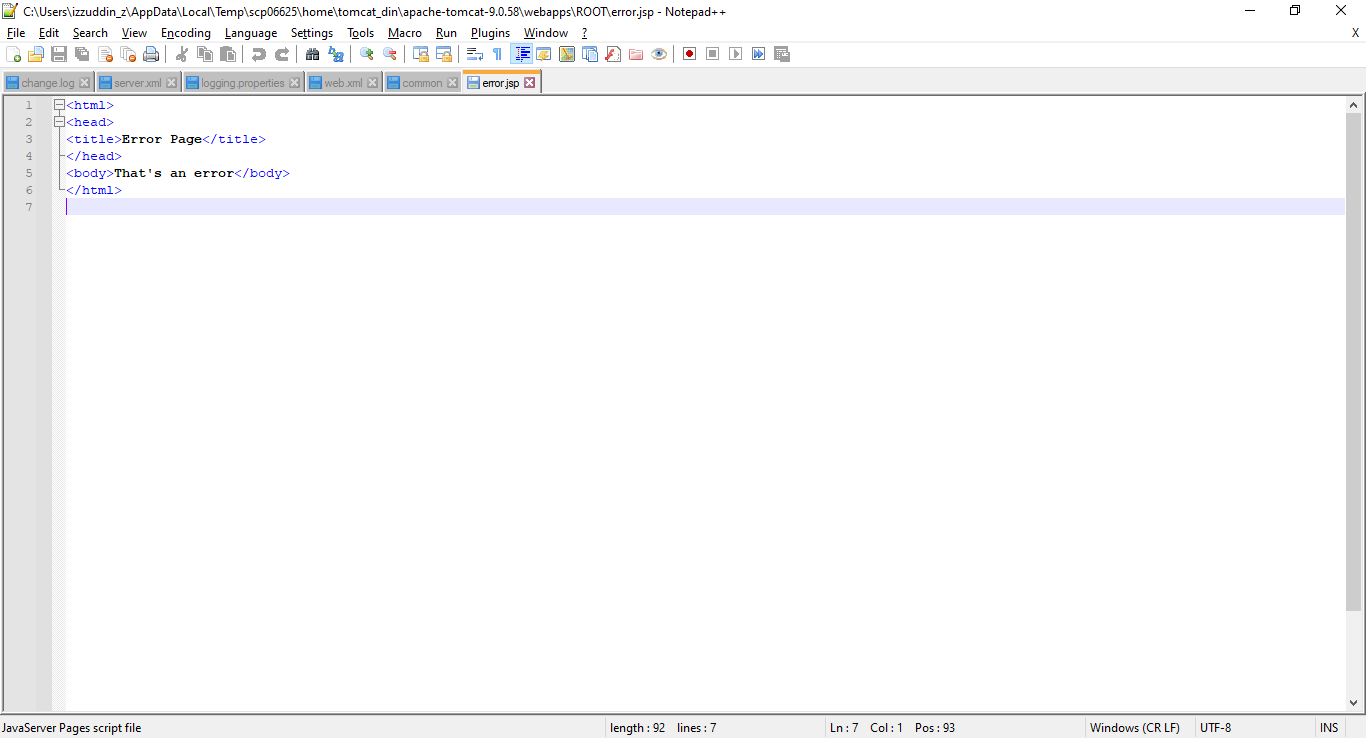
## Replace Default 404,403,500 page

To mitigate, you can first create a general error page and configure **/web.xml** to redirect to a general error page.

Go to **/home/tomcat\_din/apache-tomcat-9.0.5/webapps/ROOT/** directory Create an **error.jsp** file and put this code into this file.

|  |
| --- |
| <html>  <head>  <title>Error Page</title>  </head>  <body>That's an error</body>  </html> |

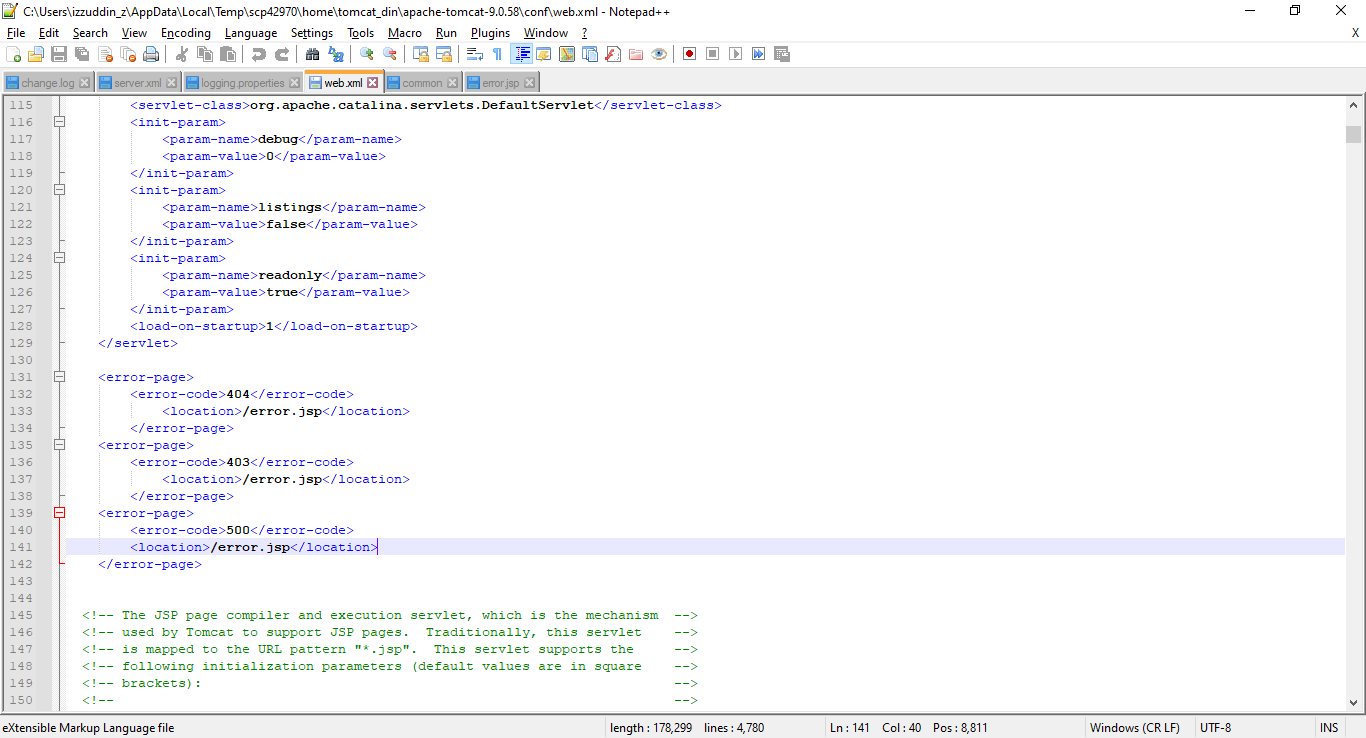
**Example:**



Go to **/conf** folder and add the following in the **web.xml** file. Ensure you add before </web-app> syntax.

|  |
| --- |
| <error-page>  <error-code>404</error-code>  <location>/error.jsp</location>  </error-page>  <error-page>  <error-code>403</error-code>  <location>/error.jsp</location>  </error-page>  <error-page>  <error-code>500</error-code>  <location>/error.jsp</location>  </error-page> |

**Example:**



## HTTP Security Configuration

Apply code below by adding below tags in **/conf** **web.xml** file:

|  |
| --- |
| <filter>  <filter-name>httpHeaderSecurity</filter-name>  <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</filter-class>  <async-supported>true</async-supported>  <init-param>  <param-name>hstsEnabled</param-name>  <param-value>true</param-value>  </init-param>  <init-param>  <param-name>hstsMaxAgeSeconds</param-name>  <param-value>31556927</param-value>  </init-param>  <init-param>  <param-name>hstsIncludeSubDomains</param-name>  <param-value>true</param-value>  </init-param>  <init-param>  <param-name>antiClickJackingEnabled</param-name>  <param-value>true</param-value>  </init-param>  <init-param>  <param-name>antiClickJackingOption</param-name>  <param-value>SAMEORIGIN</param-value>  </init-param>  <init-param>  <param-name>blockContentTypeSniffingEnabled</param-name>  <param-value>true</param-value>  </init-param>  </filter>  <!-- The mapping for the HTTP header security Filter -->  <filter-mapping>  <filter-name>httpHeaderSecurity</filter-name>  <url-pattern>/\*</url-pattern>  <dispatcher>REQUEST</dispatcher>  </filter-mapping> |

* **<filter> 🡪** This element defines the configuration for a filter.
* **<filter-name>httpHeaderSecurity</filter-name> 🡪** This element specifies the name of the filter, in this case, “httpHeaderSecurity”.
* **<filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</filter-class> 🡪** This element specifies the Java class that implements the filter, in this case, “org.apache.catalina.filters.HttpHeaderSecurityFilter”.
* **<async-supported>true</async-supported> 🡪** This element indicates that the filter supports asynchronous processing.

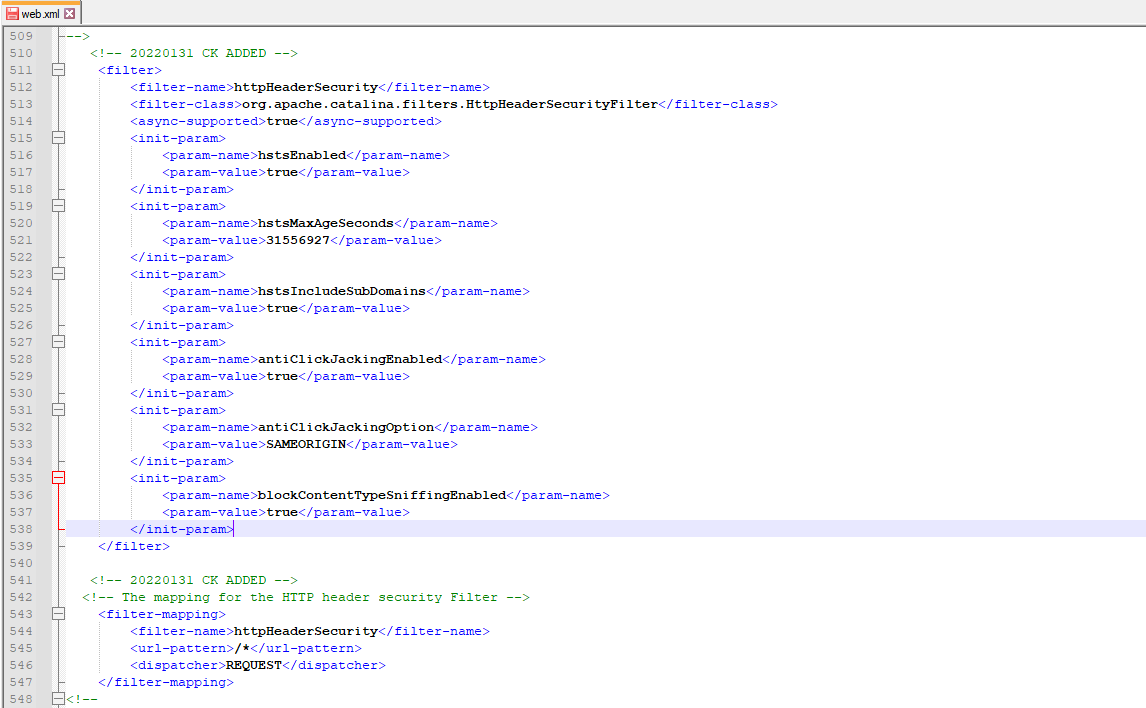
The **<init-param>** elements specify the initialization parameters (or configuration options) for the **‘HttpHeaderSecurityFilter’:**

* **<param-name>hstsEnabled</param-name> 🡪** This element specifies the name of the initialization parameter “hstsEnabled”.
* **<param-value>true</param-value> 🡪** This element specifies the value of the “hstsEnabled” parameter, in this case, “true”. It enables HTTP Strict Transport Security (HSTS).
* **<param-name>hstsMaxAgeSeconds</param-name> 🡪** This element specifies the name of the initialization parameter “hstsMaxAgeSeconds”.
* **<param-value>31556927</param-value> 🡪** This element specifies the value of the “hstsMaxAgeSeconds” parameter, in this case, “31556927” (1 year). It sets the maximum age for which the HSTS policy should be cached by the browser.
* **<param-name>hstsIncludeSubDomains</param-name> 🡪** This element specifies the name of the initialization parameter “hstsIncludeSubDomains”.
* **<param-value>true</param-value> 🡪** This element specifies the value of the “hstsIncludeSubDomains” parameter, in this case, “true”. It includes all subdomains in the HSTS policy.
* **<param-name>antiClickJackingEnabled</param-name> 🡪** This element specifies the name of the initialization parameter “antiClickJackingEnabled”.
* **<param-value>true</param-value> 🡪** This element specifies the value of the “antiClickJackingEnabled” parameter, in this case, “true”. It enables protection against clickjacking attacks.
* **<param-name>antiClickJackingOption</param-name> 🡪** This element specifies the name of the initialization parameter “antiClickJackingOption”.
* **<param-value>SAMEORIGIN</param-value> 🡪** This element specifies the value of the “antiClickJackingOption” parameter, in this case,“SAMEORIGIN”. It configures the anti-clickjacking option to allow the resource to be displayed in a frame only if the frame is on the same origin.
* **<param-name>blockContentTypeSniffingEnabled</param-name> 🡪** This element specifies the name of the initialization parameter “blockContentTypeSniffingEnabled”.
* **<param-value>true</param-value> 🡪** This element specifies the value of the “blockContentTypeSniffingEnabled” parameter, in this case, “true”. It enables blocking content type sniffing in the browser.

The **<filter-mapping>** element specifies the mapping for the **‘HttpHeaderSecurityFilter’:**

* **<filter-mapping> 🡪** This element defines the mapping for a filter.
* **<filter-name>httpHeaderSecurity</filter-name> 🡪** This element specifies the name of the filter to be mapped, in this case, “httpHeaderSecurity”.
* **<url-pattern>/\*</url-pattern> 🡪** This element specifies the URL pattern to which the filter should be applied. In this case, “/\*” means all URLs.
* **<dispatcher>REQUEST</dispatcher> 🡪** This element specifies that the filter should be applied to requests.

**Example:**



## Set SameSite Cookie Flag

Open **context.xml** in **/home/tomcat\_din/apache-tomcat-9.0.58/conf** and add the below line.

|  |
| --- |
| <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor" sameSiteCookies="strict" /> |

**<CookieProcessor> 🡪** This element defines the configuration for the cookies processing in Tomcat.

**className=”org.apache.tomcat.util.http.Rfc6265CookieProcessor” 🡪** This attribute specifies the fully qualified class name of the cookie processor implementation to be used. In this case, it is set to ‘org.apache.tomcat.util.http.Rfc6265CookieProcessor’, which is a cookie processor implementation based on the RFC 6265 specification.

**sameSiteCookies=”strict” 🡪** This attribute specifies the SameSite attribute value for cookies. The SameSite attribute is used to restrict the sending of cookies in cross-site requests. Setting it to “strict” means that cookies should only be sent in first-party contexts and not in third-party contexts.



Once done, save file and run again this command to make sure the tomcat owner has all permission to the tomcat server.

|  |
| --- |
| chown -R tomcat\_din:tomcat\_din /home/tomcat\_din |

Then, restart tomcat. Open browser and enter your IP address and the port you have set up for your https in [HTTPS setup](#_3.2_HTTPS_setup).

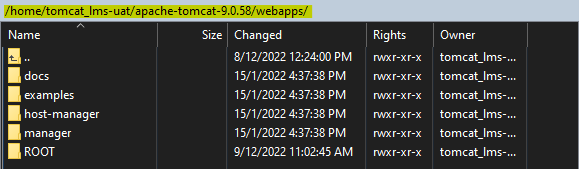
**https://10.10.6.119:8543/**

Make sure that you can still access the tomcat page as shown in [2.7.4](#tomcatpage).

## Remove Default/Unwanted Applications

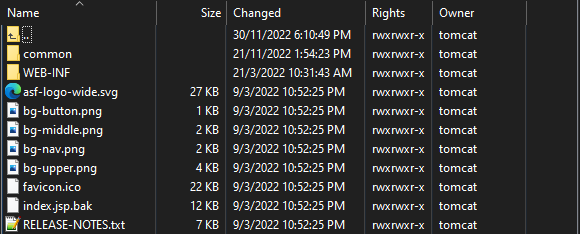
**Delete** Tomcat default application to keep it clean and avoid any known security risk with it in **{CATALINA\_HOME}/webapps** directory.

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



They are available under **/webapps** folder in your tomcat directory

* ROOT – Default welcome page. Open the ROOT folder, **rename** **index.jsp** to **index.jsp.bak**

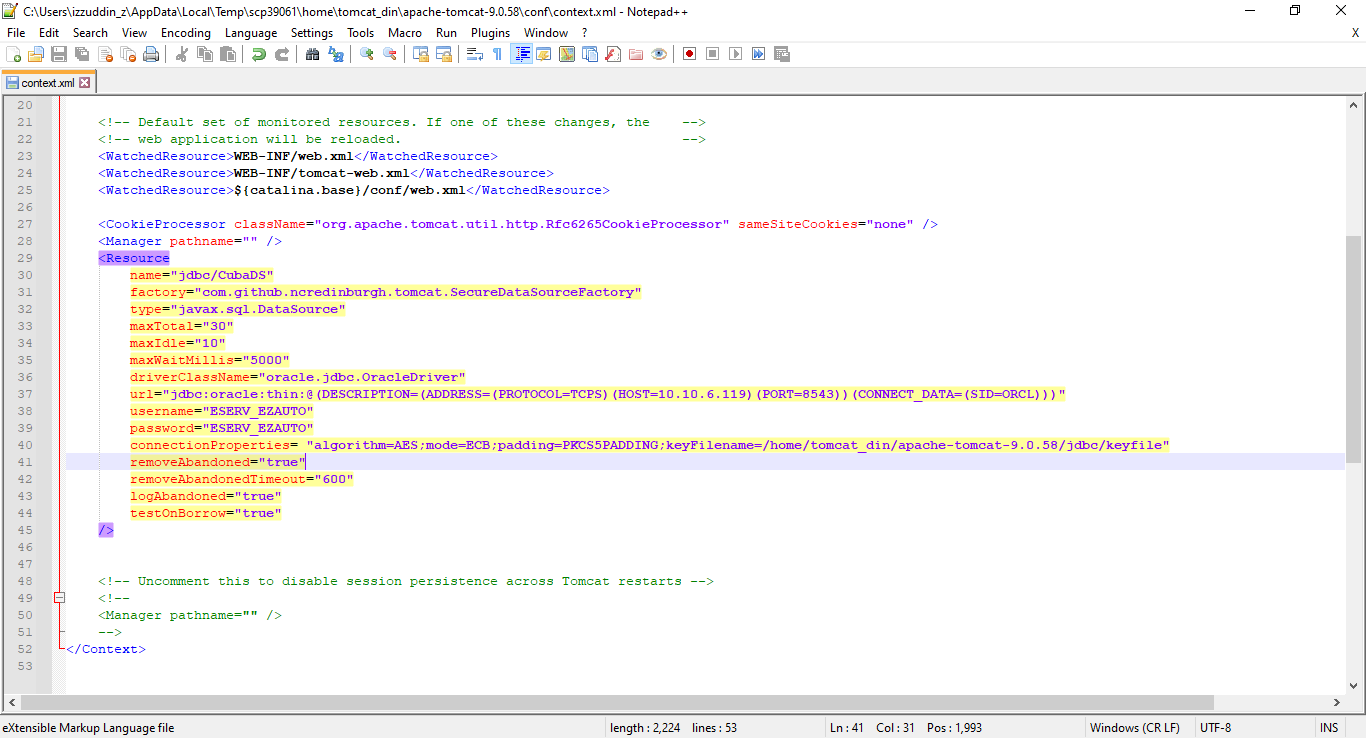


## JDBC configuration

Upload JDBC folder in directory **/home/tomcat\_din/apache-tomcat-9.0.58**. Go to **context.xml** in /conf folder. Replace this and make sure the database connection is correct.

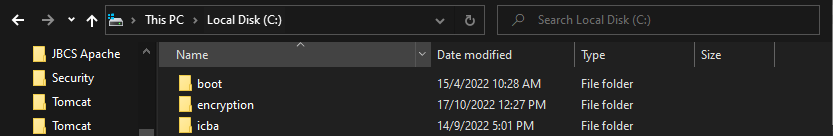
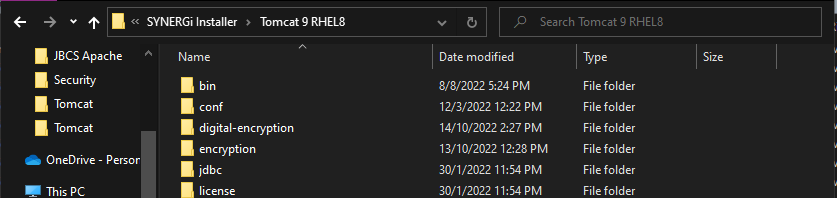
|  |
| --- |
| <Manager pathname="" />  <Resource  name="jdbc/CubaDS"  factory="com.github.ncredinburgh.tomcat.SecureDataSourceFactory"  type="javax.sql.DataSource"  maxTotal="30"  maxIdle="10"  maxWaitMillis="5000"  driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=10.10.6.198)(PORT=1528))(CONNECT\_DATA=(SID=ORCL)))"  username="ESERV\_EZAUTO"  password="ESERV\_EZAUTO”  connectionProperties= "algorithm=AES;mode=ECB;padding=PKCS5PADDING;keyFilename=/home/tomcat\_din/apache-tomcat-9.0.58/jdbc/keyfile"  removeAbandoned="true"  removeAbandonedTimeout="600"  logAbandoned="true"  testOnBorrow="true"  /> |

**Example:**



## Password Encryption

Copy the **encryption** ([check 1.1](#_File_needed) for the folder location) folder on **Your local PC C Drive**



Open command prompt and point to the encryption folder.



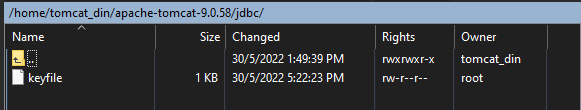
Run the following command to generate keyfile (or you can take the generated keyfile from the [jdbc folder](#_File_needed)):

|  |
| --- |
| java -jar secure-tomcat-datasourcefactory-0.3-infopro.jar generateKey AES 128 keyfile |

Example:



keyfile was generated at same folder – please upload keyfile to jdbc folder in Tomcat Catalina home IF you have re-generated a new keyfile



Then, run the following command to generate encrypted password in command prompt.

|  |
| --- |
| java -cp "secure-tomcat-datasourcefactory-0.3-infopro.jar;lib/\*" com.github.ncredinburgh.tomcat.Main encryptPassword <CLEAR\_PASSWORD> AES ECB PKCS5PADDING keyfile |

**Example:**



\*Note: Replace <CLEAR\_PASSWORD> with the password of the schema to encrypt.

Copy the generated password then update apache-tomcat/conf/context.xml

|  |
| --- |
| <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor" sameSiteCookies="none" />  <Manager pathname="" />  <Resource  name="jdbc/CubaDS"  factory="com.github.ncredinburgh.tomcat.SecureDataSourceFactory"  type="javax.sql.DataSource"  maxTotal="30"  maxIdle="10"  maxWaitMillis="5000"  driverClassName="oracle.jdbc.OracleDriver" url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=10.10.6.198)(PORT=1528))(CONNECT\_DATA=(SID=ORCL)))"  username="ESERV\_EZAUTO"  password="GEPvDR4PpDg6ieSwpBg2XQ=="  connectionProperties= "algorithm=AES;mode=ECB;padding=PKCS5PADDING;keyFilename=/home/tomcat\_din/apache-tomcat-9.0.58/jdbc/keyfile"  removeAbandoned="true"  removeAbandonedTimeout="600"  logAbandoned="true"  testOnBorrow="true"  /> |

Copy the required jar “**secure-tomcat-datasourcefactory-0.3-infopro.jar**” and “**ojdbc8-19.9.0.0.jar**” into **$CATALINA\_HOME\lib** folder.

\*Note: Only need to do for the first time

## Archive and Cron Job configuration

### **Add reboot Startup (OPTIONAL)**

***Note: This is optional. Please refer to your mentor if this is needed or not.***

1. Go to putty and change user to your tomcat user using this command

su – tomcat\_kahchong

1. Run this command to edit crontab

crontab -e

1. Click ‘**I**’ to edit, then add this line to reboot tomcat

@reboot /home/tomcat\_kahchong/apache-tomcat-9.0.58/bin/startup.sh

1. Press “**esc**” button and enter “**:wq**” to save file.

### **Compress catalina.out file**

1. Create new file in ***/etc/logrotate.d/***. Name the file as your tomcat username. Make sure the owner for this file is ***root*** following the rest of the other files in the directory.
2. Add this line into the file. Then, save the file.

/home/tomcat\_kahchong/apache-tomcat-9.0.58/logs/catalina.out {

su tomcat\_kahchong tomcat\_kahchong

copytruncate

daily

rotate 7

compress

missingok

size 5M

}

### **Archive application los in work/app\_home/logs**

1. Inside putty, run this command

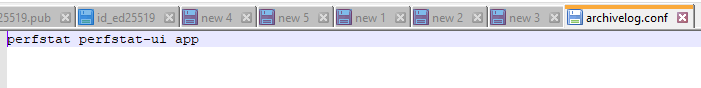
crontab –e

1. Press ‘**I**’, and then copy this command into crontab.

59 3 \* \* \* /home/tomcat\_kahchong/apache-tomcat-9.0.58/bin/archivelog.sh home/tomcat\_kahchong/apache-tomcat-9.0.58 work/app\_home/logs archivelog.conf > /dev/null 2>&1

1. Press “**esc**” button and enter “**:wq**” to save file.
2. Copy ***archivelog.sh*** file into $CATALINA\_HOME/bin. (Check [1.1](#_FILE_REQUIREMENT) for the file)
3. Inside $CATALINA\_HOME/conf, create a new file named ***archivelog.conf***, add these three files name Then, save.

**(*This may change following the application log of the war file*. *The application log can be check inside work/app\_home/logs.)***



### **Remove older logs in archive and temp folder**

1. Inside putty, run this command

crontab –e

1. Press ‘**I**’, and then copy this command into crontab.

0 4 \* \* \* sh /home/tomcat\_kahchong/apache-tomcat-9.0.58/bin/remove\_older\_logs.sh home/tomcat\_kahchong/apache-tomcat-9.0.58 30 archive > /dev/null 2>&1

0 4 \* \* \* sh /home/tomcat\_kahchong/apache-tomcat-9.0.58/bin/remove\_older\_logs.sh home/tomcat\_kahchong/apache-tomcat-9.0.58 1 temp > /dev/null 2>&1

1. Based on above, cron will run remove\_older\_logs.sh file and use the parameter configured to remove older logs in folder mentioned. Example in first line it will remove older logs in **archive** folder after 30 days. The following line will remove older logs in **temp** folder after 1 day.
2. Press “**esc**” button and enter “**:wq**” to save file.
3. Copy ***remove\_older\_logs.sh*** file into $CATALINA\_HOME/bin. (Check [1.1](#_FILE_REQUIREMENT) for the file)

## CORS Security Configuration (Optional)

\*\*Only apply if needed

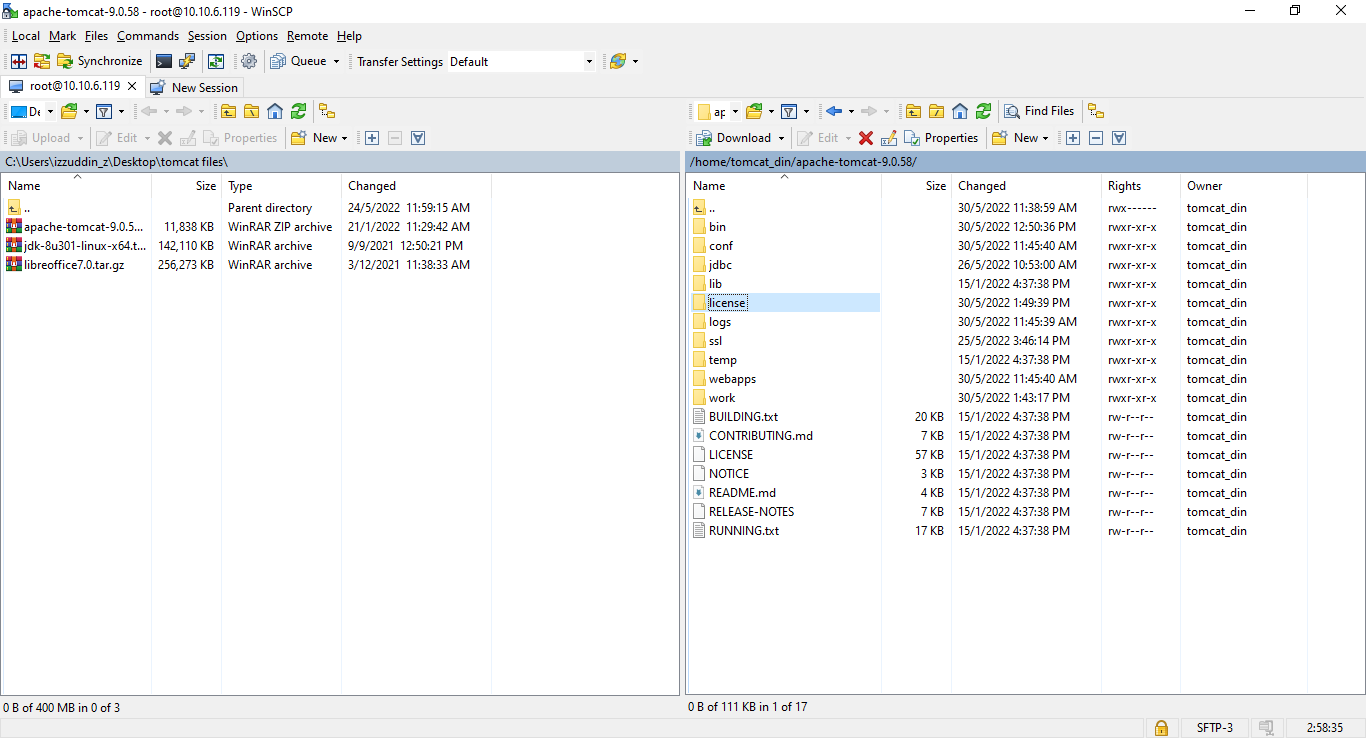
Add the code below by adding them below the **<filter>** tag in **/home/tomcat\_din/apache-tomcat-9.0.58/conf/web.xml**

|  |
| --- |
| <filter>  <filter-name>CorsFilter</filter-name>  <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>  <init-param>  <param-name>cors.allowed.origins</param-name>  <param-value>\*</param-value>  </init-param>  <init-param>  <param-name>cors.allowed.methods</param-name>  <param-value>GET,POST,HEAD</param-value>  </init-param>  <init-param>  <param-name>cors.allowed.headers</param-name>  <param-value>Content-Type,X-Requested-With,accept,Origin,Access-Control-Request-Method,Access-Control-Request-Headers</param-value>  </init-param>  <init-param>  <param-name>cors.exposed.headers</param-name>  <param-value>Access-Control-Allow-Origin,Access-Control-Allow-Credentials</param-value>  </init-param>  <init-param>  <param-name>cors.preflight.maxage</param-name>  <param-value>10</param-value>  </init-param>  </filter>  <!-- The mapping for the CORS security Filter -->  <filter-mapping>  <filter-name>CorsFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> |

## Synergy Configuration (Optional)

\*\*Only apply if needed

Create license folder in **/home/tomcat\_din/apache-tomcat-9.0.58/**,and copy and paste the **license.lic** in the folder



Then, go to **/home/tomcat\_din/apache-tomcat-9.0.58/work/app\_home/** then create **local.app.properties**. Add below configuration.

|  |
| --- |
| #GENERAL  reporting.openoffice.path=/home/tomcat\_din/libreoffice7.0/program  whatsapp.enabled=false  cuba.httpSessionExpirationTimeoutSec=3600  cuba.web.closeIdleHttpSessions=true  cuba.web.uiHeartbeatIntervalSec=3660  cuba.userSessionExpirationTimeoutSec=3600  cuba.fileStorageDir=/filerepo/filestorage/  cuba.web.productionMode=true  cuba.web.infopro.webMenuShowPOCItem=false  cuba.infopro.envLabel=PROD  cuba.infopro.samlCode=synergi  #EMAIL  cuba.email.smtpHost = mail.rfc.com.ph  #SAML  cuba.addon.saml.samlIntegrationEnabled=true  cuba.addon.saml.proxy.enabled=true  cuba.addon.saml.proxy.serverUrl=https://10.10.6.119:8843/eservice  cuba.addon.saml.logAllSamlMessages=false  #DB  cuba.dataSourceProvider=jndi  cuba.dataSourceJndiName=jdbc/CubaDS  #for server name in SYS\_SERVER  cuba.webHostName=10.10.6.119  cuba.webPort=8845  cuba.schedulingActive=true  # put here for notes, CUBA scheduling framework tries to call API every 1 second - default.  # set to higher to reduce the FOR UPDATE interval  cuba.schedulingInterval=15000  #License  #license.allowMultipleSession = false  license.publicKey = MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC30AK5k8QGadIUHrX84f9Jya4+F9gtSOu6lcoYgyfq0KsNrj8CbFPg9fuWook3G3guHzJBrxDPDnaPyG26a09rDXWddSmKFuSAN7fIkeynKReAtTfJVF9k2MIPhACM14dlYaMM9Oe1lNJQkgSVk+6TkfB0dht3Pe2boEIQwh4nAwIDAQAB  license.licenseFile = /home/tomcat\_din/apache-tomcat-9.0.58/license/license.lic  license.licenseKey = gXMHaTfAnLk8LrHU8EaKg64GaoYScxegI4+YQCtihCfMJaHuV4TL0Vttup5NkM2hSXNDraDzh4NGkVXWP2EeVmt1D8T3zZnz4WpV+ZVlAur2wLtWcP9ggqV6TGqt8SI8E9KhS8icx15Hs3/8UHdqPgwr4MjAcqqwK+hwZOWVMXc=.g97Mmgy984tMOoj+tqVN0A==.ZST/q6I8/y3a8uCrAgy9RkCNX8uhBj9ZweDeZHD+tD4V64Cy4VsW0lQZOSbEe1lijy2C5qWrmu89sWJ85i68h9jKfb/8PH5NZFwaVEhXYan/3QPAk211neexWqyJm7ENjOqWpN1xeJX7uCefCaRdCbd0Zr+7fQ3e8iS8LNMvcXI= |

Save file.

*\*\*Please note, above highlighted might change. Please double check before we execute the command.*

# POST INSTALLATION

## Change owner and permission for your Tomcat user

Login as “**root**” via putty.

|  |
| --- |
| su root |

Example:



Execute below commands for OS User

|  |
| --- |
| chgrp -R tomcat\_din /home/tomcat\_din/  chown -R tomcat\_din /home/tomcat\_din/  chmod -R 775 /home/tomcat\_din/ |

Example:



## Restart Tomcat

To restart start the server, log in into your tomcat user. Use the command below

|  |
| --- |
| su – tomcat\_din |

Then, go into your bin directory

|  |
| --- |
| cd /home/tomcat\_din/apache-tomcat-9.0.58/bin |

Restart your tomcat server as per below command

|  |
| --- |
| ./shutdown.sh  ./startup.sh |

Then, open your browser and enter your IP address and the port you have set up for your tomcat.