Cloud Computing - Virtual Servers

Task 1 – Installing/Configuring a Tomcat Server

1.1 If you have managed to SSH into the VM server, you can start to install software and configure the server any way you wish. You can now install Tomcat server on the VM so that you can use this server to host java webapps.

All interaction with the server will be done through the command line interface (CLI).

First thing to do is to update the server software repositories with the following command:

```
sudo apt update
```

1.2 Next you need to install the Java Development Kit (JDK) on the server:

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

1.3 Create the tomcat installation folders:

```
sudo mkdir /opt
```

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

1.4 Create a tomcat user with the following command:

```
sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat
```

1.5 Next install tomcat. Start by downloading the tomcat installation files to the VM server, at the time of writing this tomcat 9.0.80 is the latest version available (remember you need the tar.gz link). The following command will download tomcat to a temp folder:

```
wget LINK_FOR_LATEST_TOMCAT9_VERSION -P /tmp
```

1.6 Next, extract the downloaded archive and move its contents to the tomcat folder:

```
sudo tar xf /tmp/apache-tomcat-9*.tar.gz -C /opt/tomcat
```

1.7 To have more control over Tomcat versions and updates, create symbolic link called latest that points to the Tomcat installation directory, this step will save you typing “apache-tomcat-9.0.30” every time you need to configure tomcat:

```
sudo ln -s /opt/tomcat/apache-tomcat-9* /opt/tomcat/latest
```

1.8 The following command changes the directory ownership to the tomcat user created in step 1.22:

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

1.9 The following command makes the scripts that come with tomcat “executable” so that they can be run:

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

1.10 Next you need to create service file that will allow tomcat to run as a service on the server. The following command will create a new file called “tomcat.service” in the “etc/systemd/system” folder:

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

The above command will open a CLI text editor, use the editor to enter the following configuration settings:

```
[Unit]
Description=Tomcat 9 servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

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

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

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

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

1.11 Then press **ctrl+s** to save the file and **ctrl+x** to exit the text editor.

1.12 Next you need to make the server aware of the newly created service file:

```
sudo systemctl daemon-reload
```

1.13 Its finally time to start tomcat, start the tomcat service with the following command:

```
sudo systemctl start tomcat
```

You can verify if tomcat is running properly with the following command:

```
sudo systemctl status tomcat
```

You should see the following message:

```
kaleem@ubuntu-vm-demo:/$ sudo systemctl status tomcat
● tomcat.service - Tomcat 9 servlet container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-01-01 19:06:23 UTC; 18s ago
  Process: 16456 ExecStart=/opt/tomcat/latest/bin/startup.sh (code=exited, status=0/SUCCESS)
 Main PID: 16470 (java)
    Tasks: 43 (limit: 2318)
   CGroup: /system.slice/tomcat.service
           └─16470 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/opt/tomcat/
```

Press **ctrl+c** to exit the status message once you are finished)

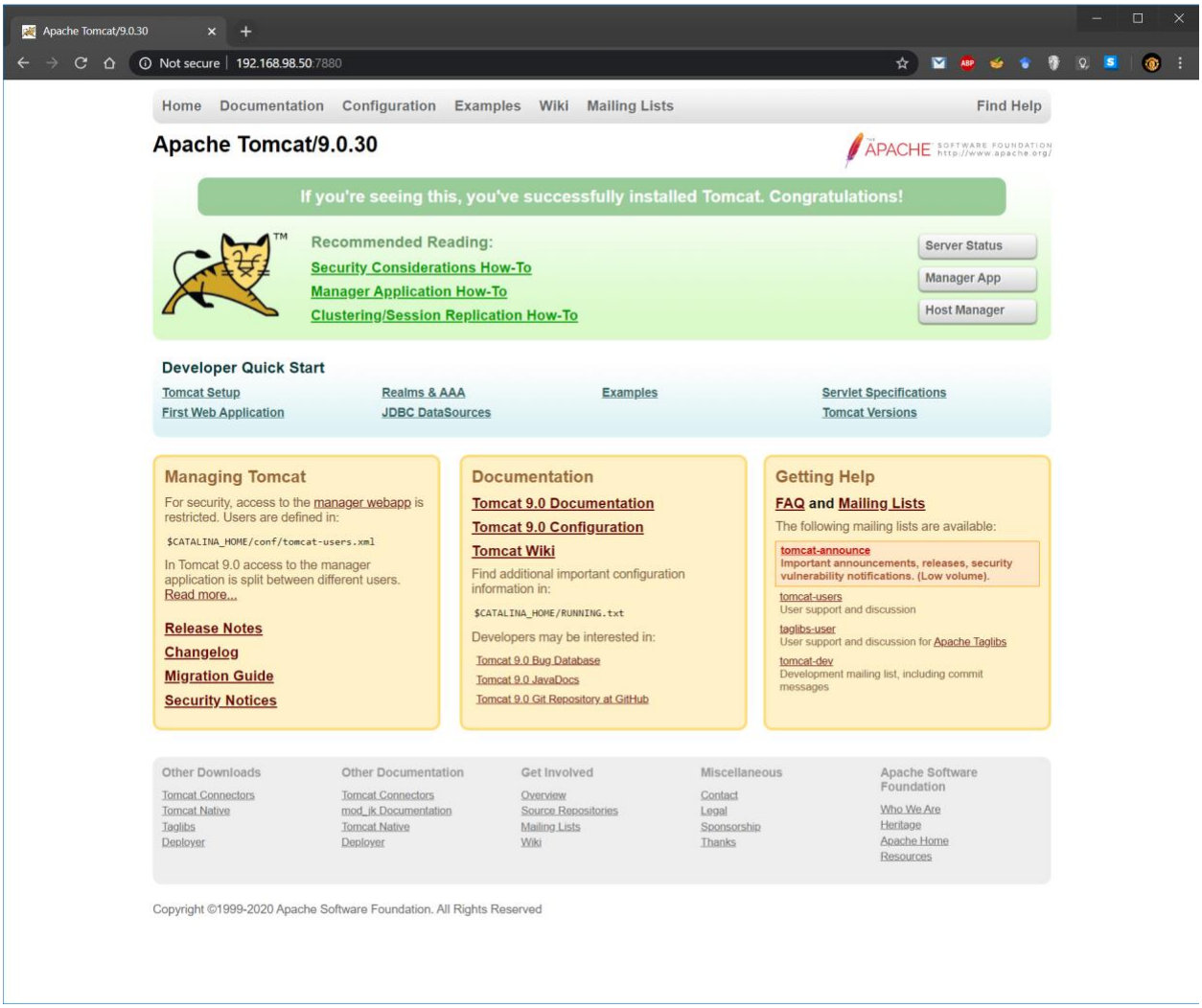
1.14 If there are no errors, enable the Tomcat service to be automatically started when the server boots with the following command:

```
sudo systemctl enable tomcat
```

1.15 Tomcat runs on port 8080 by default, since this is a non-standard port, you must open this port on the server for communication with the outside world. To allow traffic on port 8080 type the following command:

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

1.16 You have now successfully installed Tomcat on your virtual server, you can test this by connecting to the VM server through a browser on the host machine. Open a browser and type the IP address you used to SSH into the server followed by port 7880 (illustrated below). You should see the tomcat welcome screen:



1.17 Now that Tomcat is installed and running, the next step is to create a user with access the web management interface.

Tomcat users and roles are defined in the tomcat-users.xml file. This file is a template with comments and examples describing how to configure the create a user or role. Open this file for editing with the following command:

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

1.18 Add the following lines to the tomcat-users.xml file to create a user with access to the web management interface (feel free to change the username and password values to your liking):

```
  <role rolename="admin-gui"/>
  <role rolename="manager-gui"/>
  <user username="admin" password="password" roles="admin-gui,manager-gui"/>
```

1.19 Then press **ctrl+s** to save the file and **ctrl+x** to exit the text editor.

1.20 The next step is to enable access to the management interface from any IP. To enable access to the web interface from any source IP, open the following two files and comment or remove the lines highlighted in yellow below.

For the Manager app, open the following file:

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

For the Host Manager app, open the following file:

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

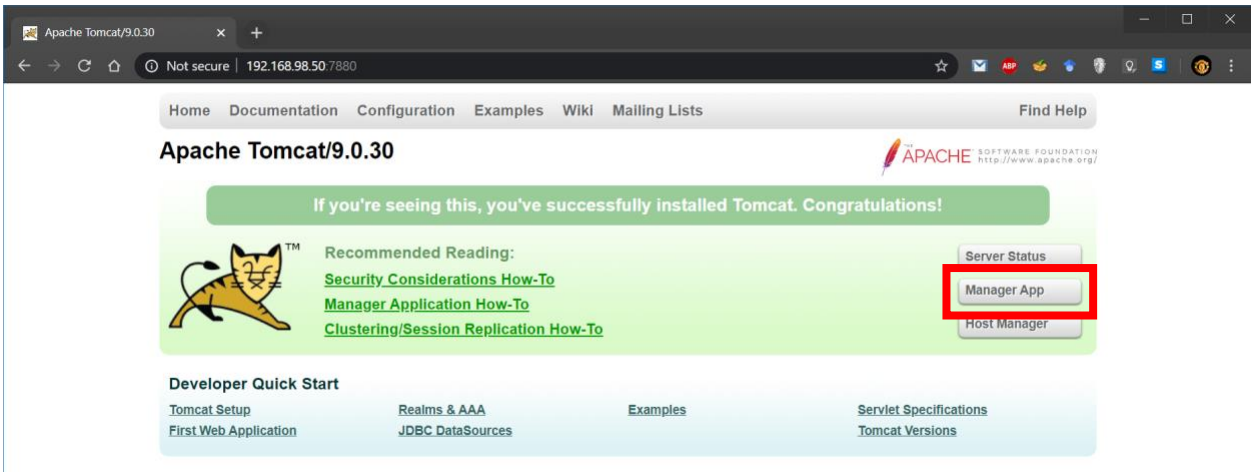
```
<Context antiResourceLocking="false" privileged="true" >
<!--
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
-->
</Context>
```

Remember to save the changes in **both** files by pressing **ctrl+o** to save the file and **ctrl+x** to exit the text editor.

1.21 Finally, just restart tomcat to update the changes made:

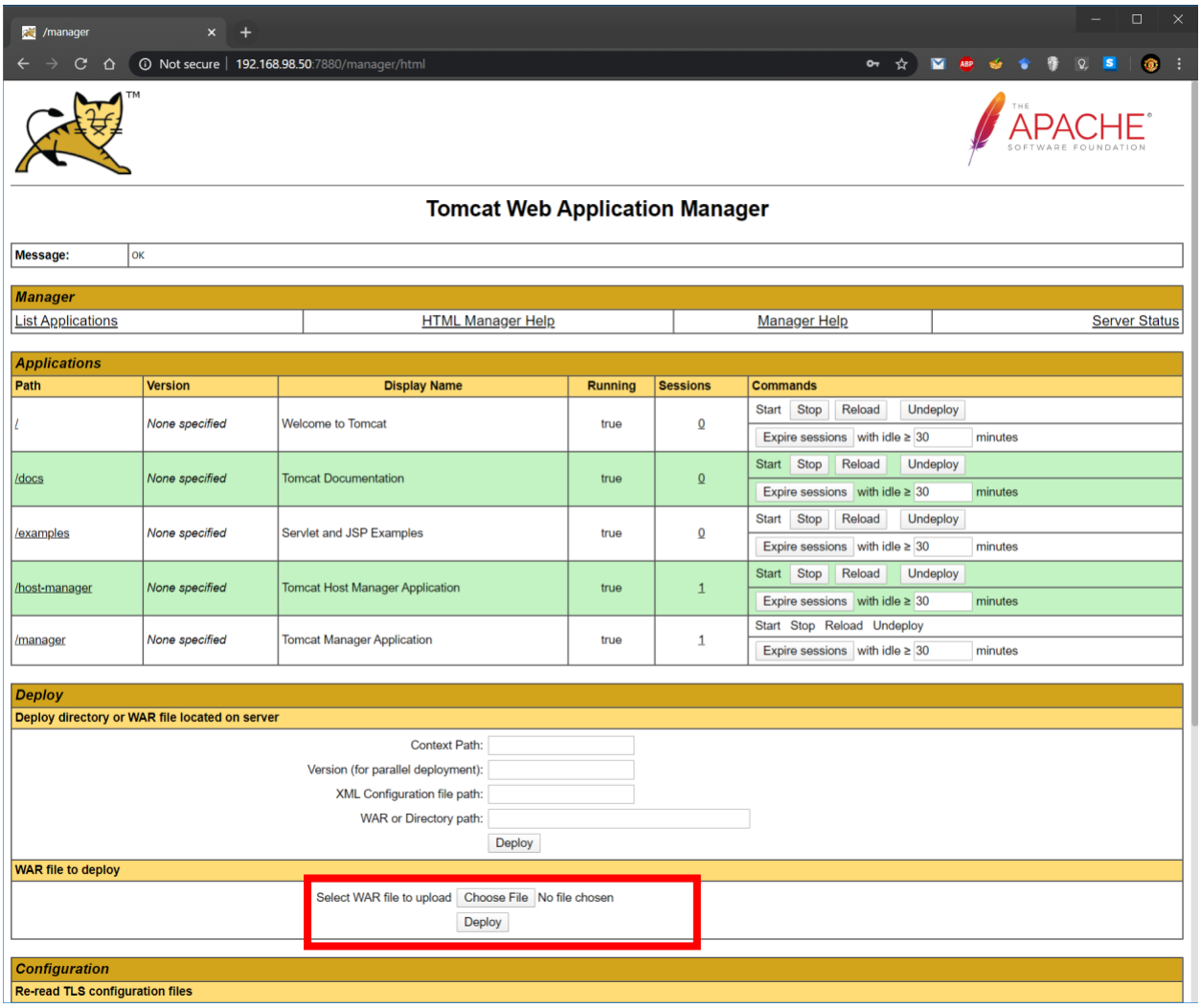
```
sudo systemctl restart tomcat
```

1.22 You should now have full access to the tomcat web management interface to deploy java webapps. Click on the “Manager App” button, you will be prompted for the username and password you set during step 1.36.

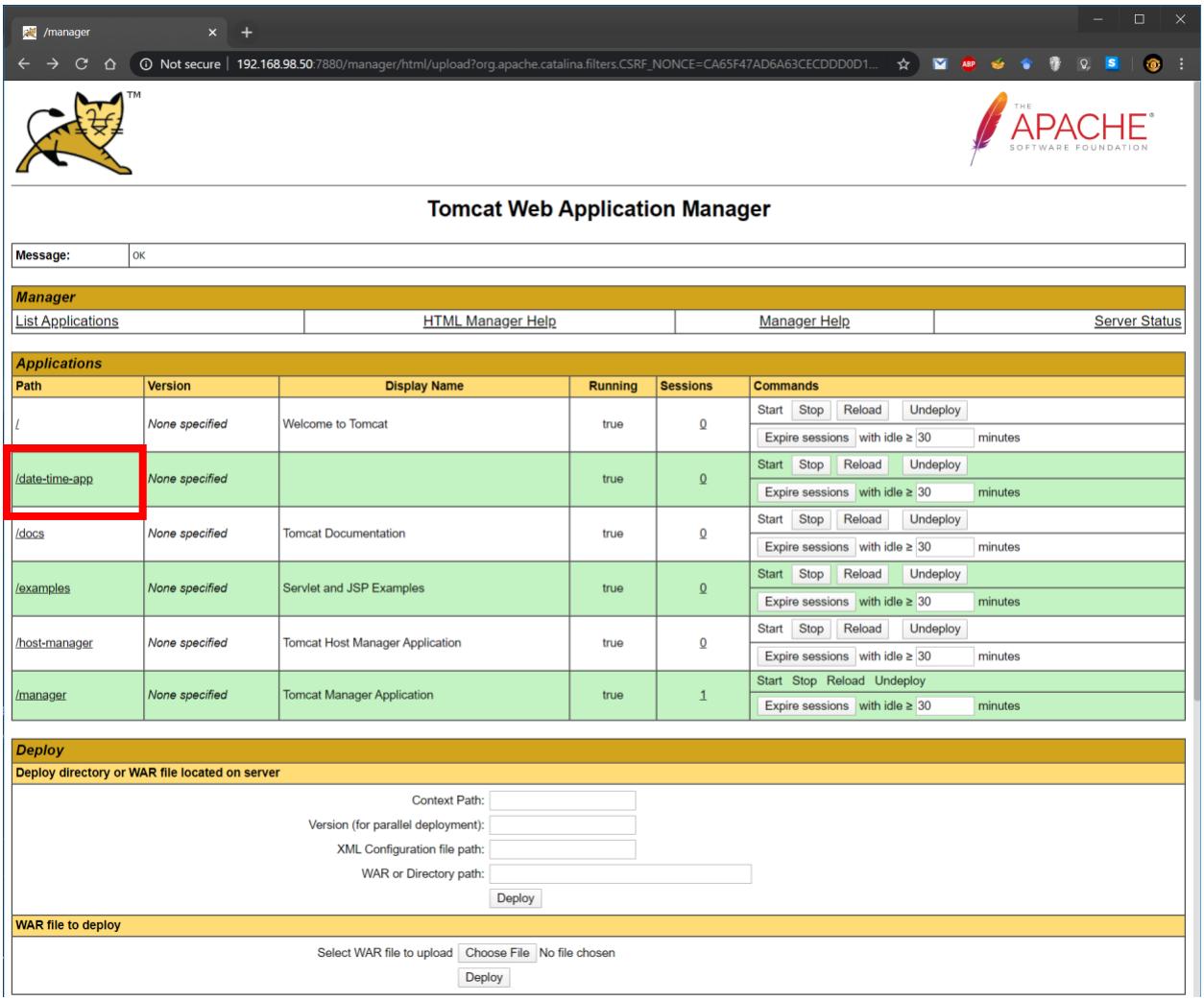


1.23 From this interface you can deploy java web apps (WAR files). There a simple date-time web app WAR file on Moodle which you can download and deploy to the you server.

Download the example WAR file and deploy it on your server through the web app manager. Click on the **Choose File** button, select the WAR file and then click the **Deploy** button.



1.24 Once you have deployed the WAR file you will see it listed in the applications table, from here you can click on the application name and see your app working in the browser:



That's it, you've successfully deployed a server from scratch, installed and configured tomcat and deployed a java web app. Well done!

Now moving forward, you can install and configure MySQL database on this virtual server so that you can deploy database driven java web apps. Or you can install the full LAMP stack so that you can deploy PHP web apps.