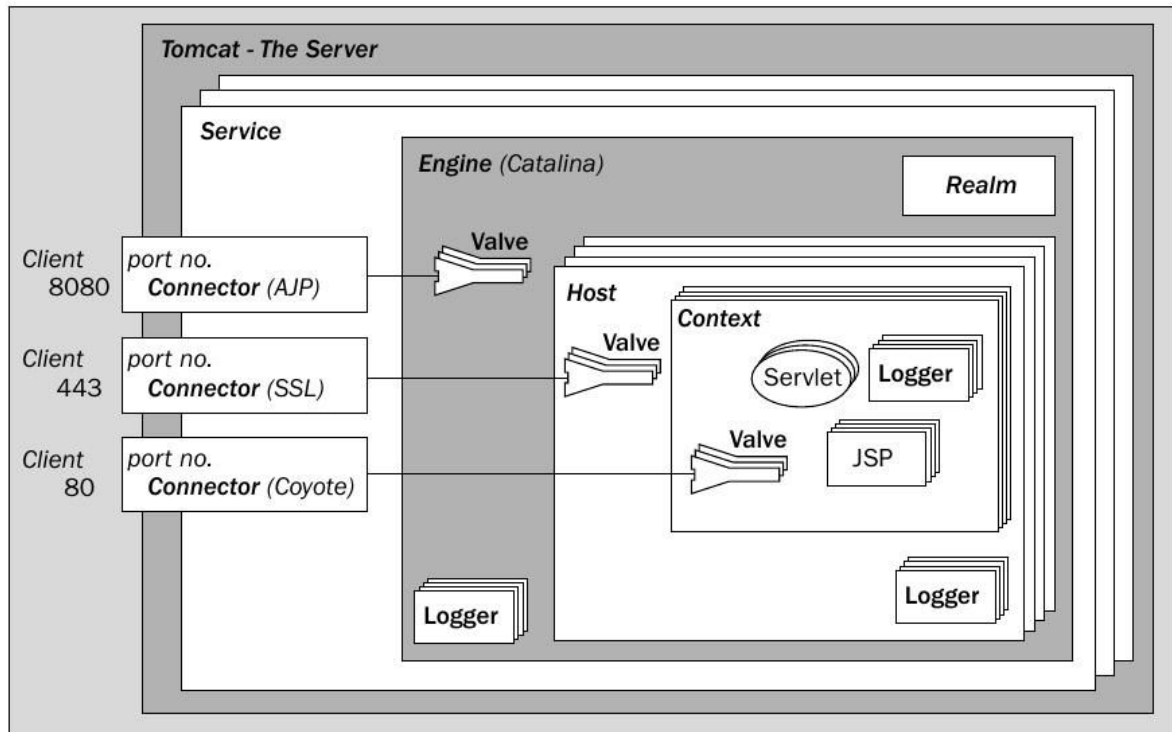


Homework, Apache_Tomcat_HTTPd

1. Apache Tomcat

Tomcat architecture



Tomcat's architecture.

Linux version (UBUNTU18.04)

1.1 Install Tomcat on Linux and carry out p.1.2-1.12 of the task

Install jdk

```
sudo apt-get install -y openjdk-8-jre
sudo apt-get install -y openjdk-8-jdk
whereis java
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
export PATH=$PATH:$JAVA_HOME
```

Install and start Tomcat server

```
sudo mkdir /home/tomcat
cd /home/tomcat
sudo wget
    http://ftp.byfly.by/pub/apache.org/tomcat/tomcat-9/v9.0.22/bin/apache-tomcat-9.0.22.tar.gz
sudo tar -xvf apache-tomcat-9.0.22.tar.gz
sudo rm -vf apache-tomcat-9.0.22.tar.gz
cd apache-tomcat-9.0.22
export CATALINA_HOME=$PWD
CATALINA_BASE=/home/web
sudo mkdir $CATALINA_BASE/
sudo mkdir $CATALINA_BASE/webapps
export CATALINA_BASE
export PATH=$PATH:$CATALINA_HOME:$CATALINA_BASE
```

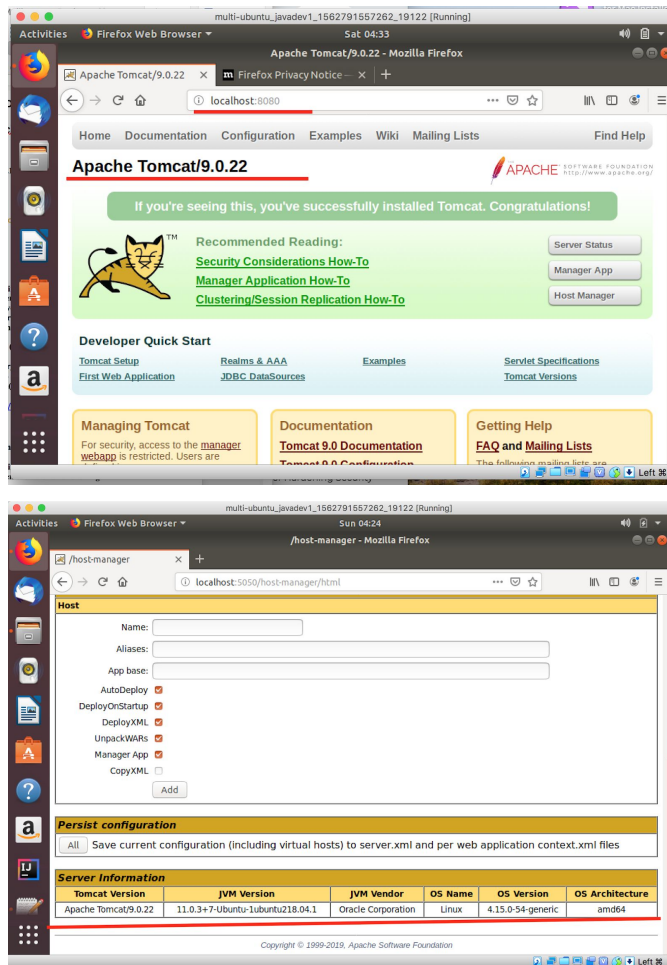
For access to manager and admin part of Tomcat it is needed to add appropriate roles. Insert into file `$CATALINA_HOME/conf/tomcat-users.xml`:

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="manager" password="qwerty1!"
    roles="manager-gui,manager-script"/>
<role rolename="admin-gui"/>
<user username="admin" password="qwerty1!" roles="admin-gui"/>
```

Start Tomcat server

```
sudo $CATALINA_HOME/bin/catalina.sh start
```

Screenshot#1.1. Tomcat is started



1.2. Create a Tomcat context

<https://github.com/dennis00010011b/simpleweb/blob/master/META-INF/context.xml>

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiResourceLocking="false" privileged="false" >
</Context>
```

Actually I used context by default

1.3 Create a simple WEB application consisting of an HTML document and a servlet

I used webapp (the Greeting) which was created in Task5.7

<https://github.com/dennis00010011b/simpleweb>

1.4 Deploy this application to Tomcat and demonstrate its functioning

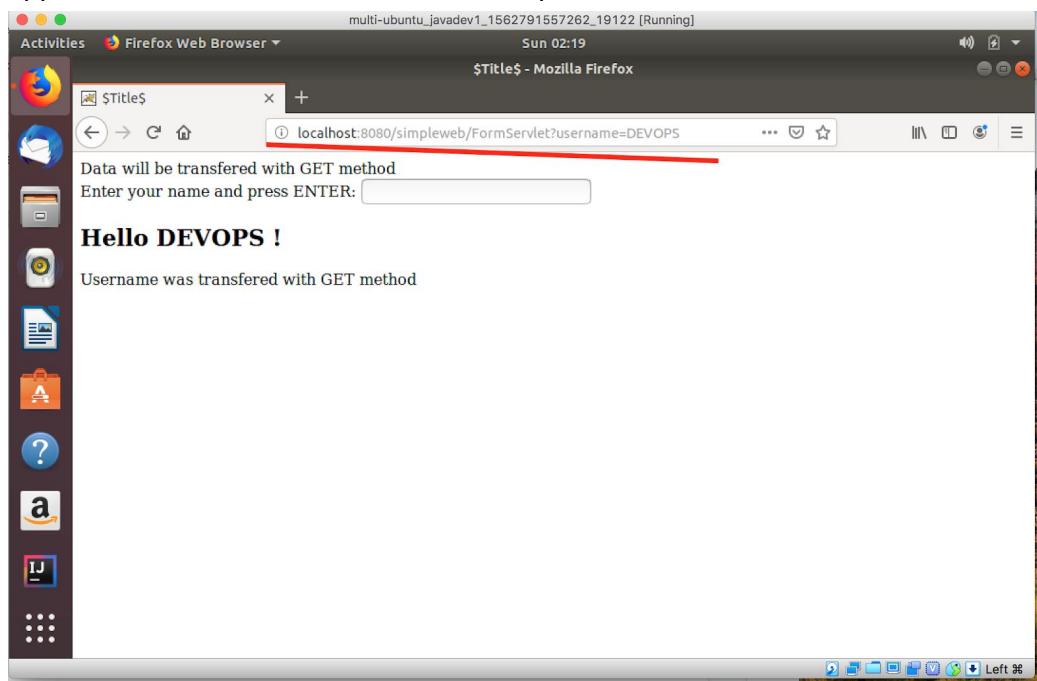
Deploying process described here

http://tomcat.apache.org/tomcat-8.0-doc/manager-howto.html#Deploy_A_New_Application_from_a_Local_Path

For deploying I created folder `$CATALINA_HOME/webapps/simpleweb` and clone app from gitHub repo:

```
sudo $CATALINA_HOME/bin/catalina.sh stop
cd $CATALINA_HOME/webapps
git clone https://github.com/dennis00010011b/simpleweb
sudo $CATALINA_HOME/bin/catalina.sh start
```

App available on URL `localhost:8080/simpleweb`



1.5 Pack this web application to a web archive & 1.6. Remove the deployed WEB application from Tomcat

Run from `./simpleweb`:

```
jar -cvf greeting.war *
mkdir ../greeting
mv greeting.war ../greeting
sudo rm -r ../simpleweb
```

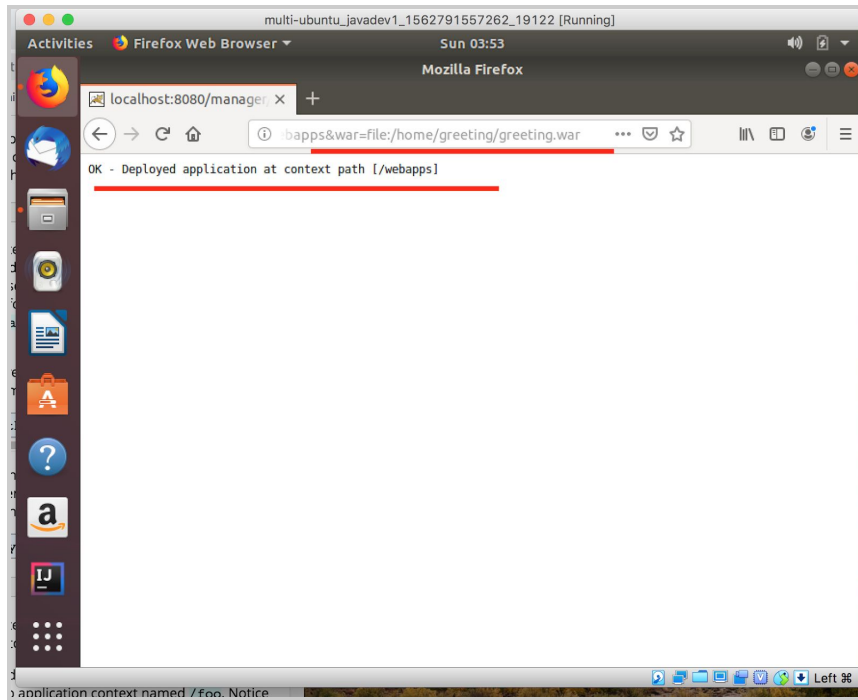
1.7. Deploy WEB application from WEB archive

Manager role manager-script MUST be allowed.

Deploying can be done with GET request to URL

```
localhost:8080/manager/text/deploy?path=/greeting&war=file:/home/greeting/greeting.war
```

I used GUI FireFox for make such GET request.



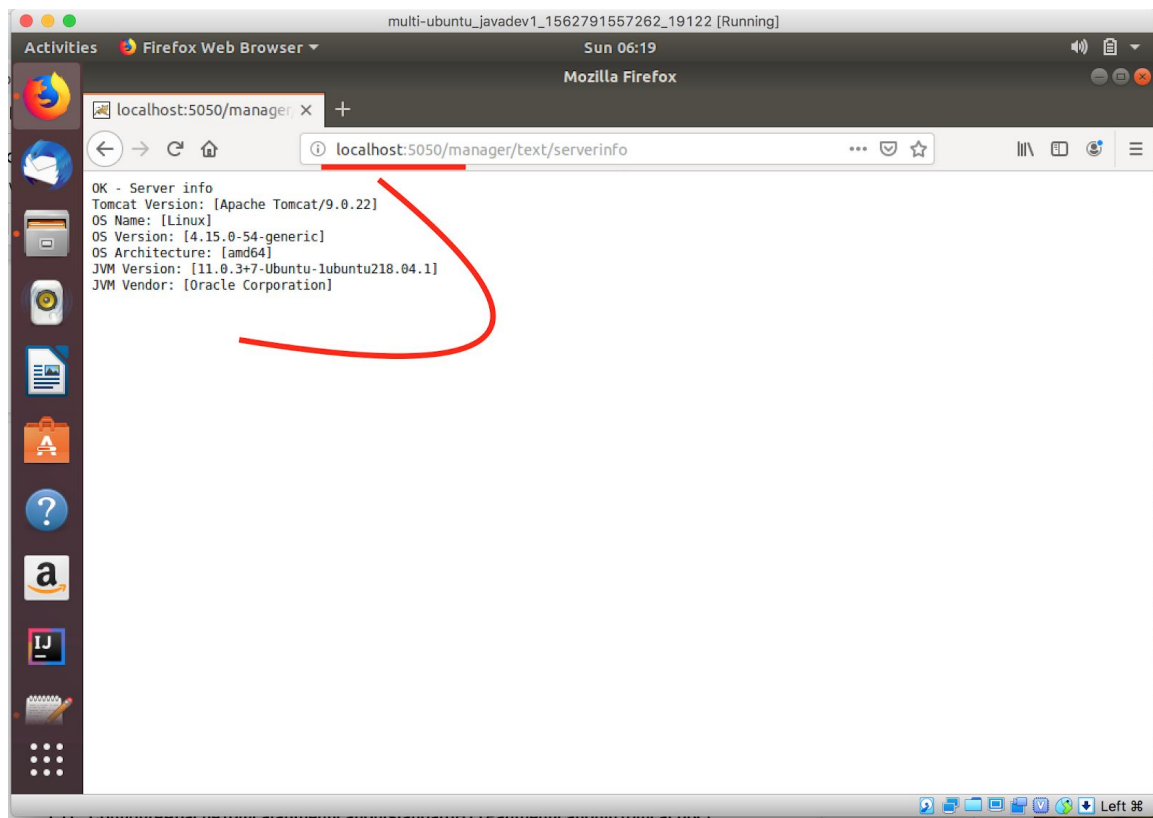
1.8. Change the Tomcat port to 5050

Add one more connector in `server.xml`:

```
<Connector port="5050" protocol="HTTP/1.1"
connectionTimeout="600000"
redirectPort="8443" />
```

Restart server

```
sudo $CATALINA_HOME/bin/catalina.sh stop
sudo $CATALINA_HOME/bin/catalina.sh start
```



1.9. Change the default application directory (webapps) to another and check that the directory has picked up (i.e. applications in this directory are accessible via http)

Need to change parameter `appBase` for the host in `server.xml` :

```
<Host name="localhost"  appBase="webapps1"
      unpackWARs="true" autoDeploy="true">
    <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" />
</Host>
```

And provide folder `webapps1` with webapps.

1.10. Write command on Linux OS to execute p.1.4-1.9

See commands above in p.1.4-1.9.

Also I tested this script for installing, starting Tomcat server and deploying webapps *Greeting*
<https://github.com/isurunix/tomcat-install-script>

1.11. Configure Apache Tomcat authentication(standard HTTP authentication in Tomcat.doc)

About HTTP auth framework

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication>

About Tomcat windows auth

<https://tomcat.apache.org/tomcat-9.0-doc/windows-auth-howto.html>

HTTP authentication is configured in `web.xml` for given webapp:

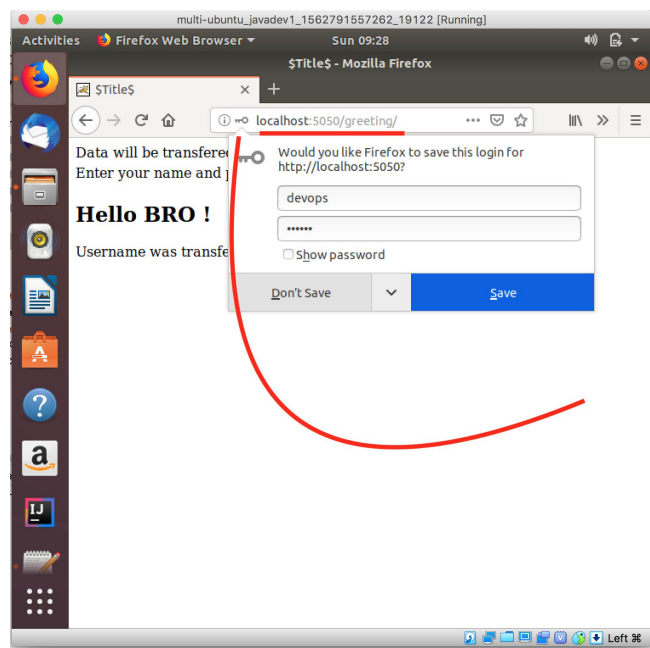
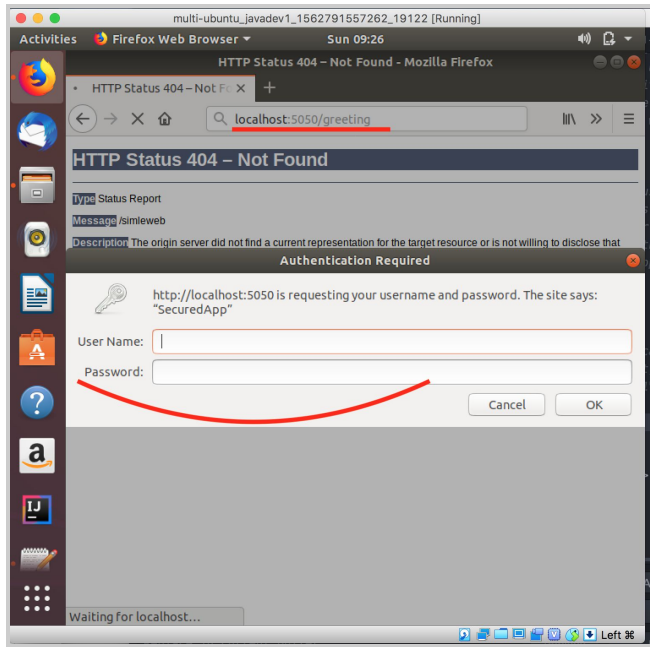
```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>ApplicationContent</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>devops</role-name>
  </auth-constraint>
</security-constraint>

<security-role>
  <role-name>devops</role-name>
</security-role>
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>SecuredApp</realm-name>
</login-config>
```

Also it needs to add role to tomcat-users.xml:

```
<role rolename="devops"/>
```

```
<user username="devops" password="qwerty" roles="devops"/>
```



1.12. Configure SSL for Apache Tomcat

1. Create a new **JKS** keystore from scratch, containing a single self-signed Certificate, execute the following from a terminal command line:

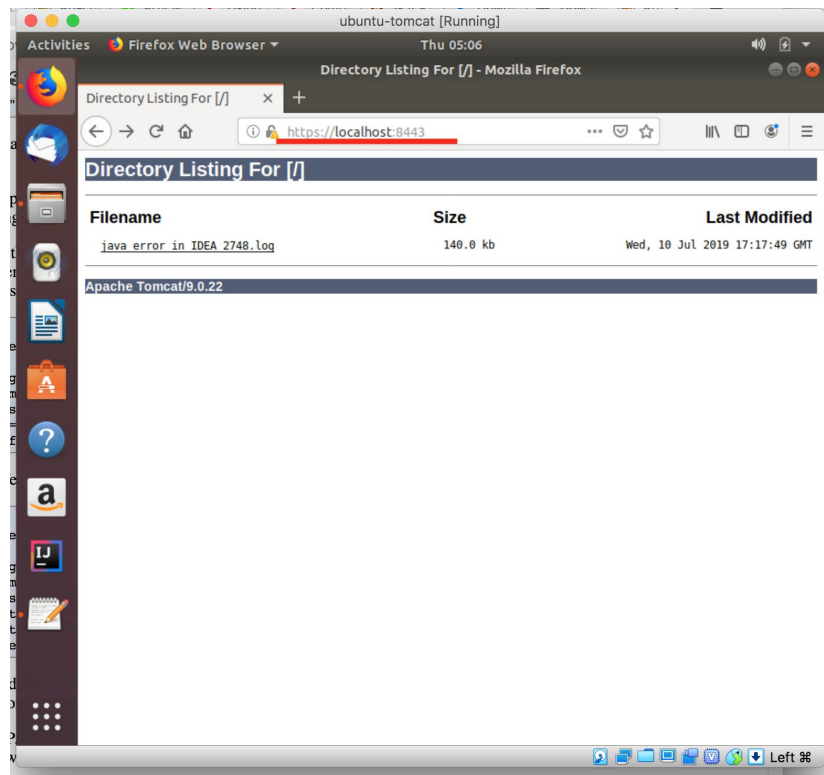
```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore  
$CATALINA_HOME
```

.*keystore* file was created, keys in JKS format, now we have a keystore file with a Certificate

2. Add connector to **server.xml** :

```
<Connector port="8443"  
protocol="org.apache.coyote.http11.Http11NioProtocol"  
maxThreads="150"  
scheme="https" secure="true" SSLEnabled="true"  
keystoreFile="$CATALINA_HOME/conf/.keystore"  
keystorePass="qwerty"  
clientAuth="false" sslProtocol="TLS" / >
```

3. Check SSL connection : <https://localhost:8443>



Windows version

1.1. Install Tomcat on Windows and carry out p.1.2-1.12 of the task

Config JAVA_HOME:

```
setx JAVA_HOME "C:\Program Files\Java\jdk1.8.0"  
setx PATH "%PATH%;%JAVA_HOME%\bin";
```

Install Tomcat:

```
mkdir C:\temp
```

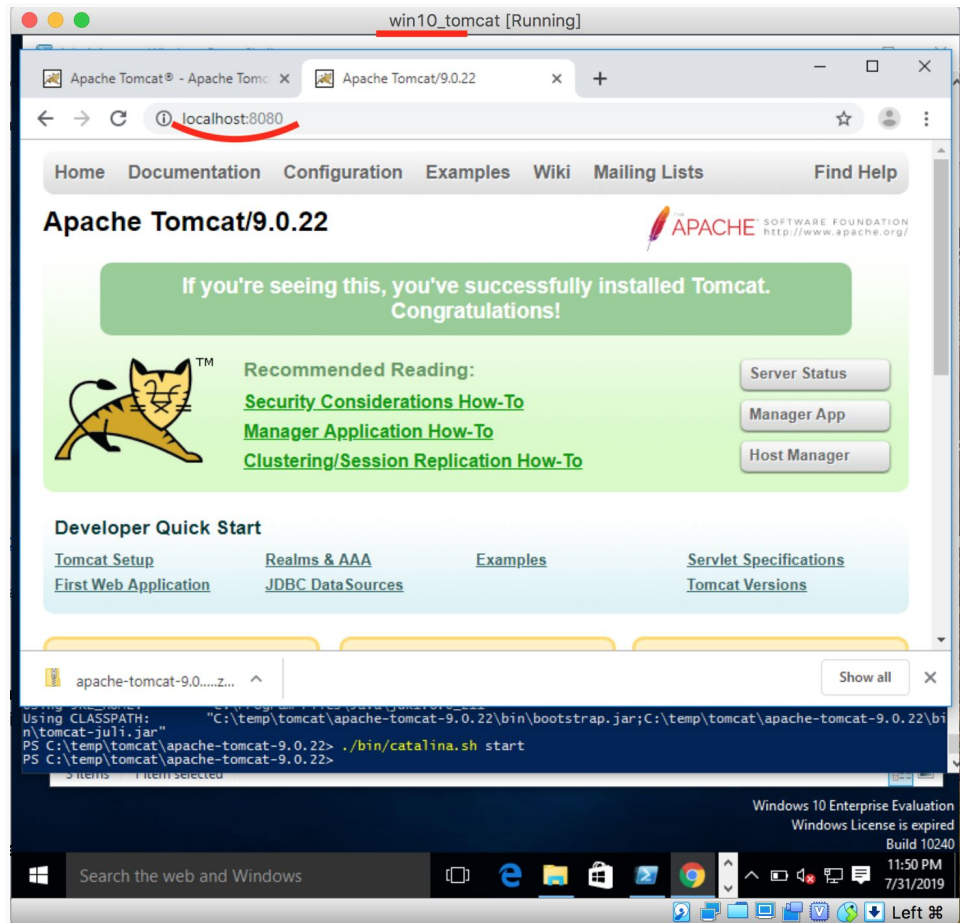
```
cd c:\temp
```

```
Invoke-WebRequest -URI
```

```
http://ftp.byfly.by/pub/apache.org/tomcat/tomcat-9/v9.0.22/bin/apache-tomcat-9.0.22-windows-x64.zip -OutFile
```

```
Expand-Archive apache-tomcat-9.0.22-windows-x64.zip tomcat
```

```
Setx CATALINA_HOME "C:\temp\tomcat\apache-tomcat-9.0.22"
```



1.2. Create a Tomcat context

<https://github.com/dennis00010011b/simpleweb/blob/master/META-INF/context.xml>

```
<?xml version="1.0" encoding="UTF-8"?>  
<Context antiResourceLocking="false" privileged="false" >
```

</Context>

1.3 Create a simple WEB application consisting of an HTML document and a servlet

I used webapp (the Greeting) which was created in Task5.7

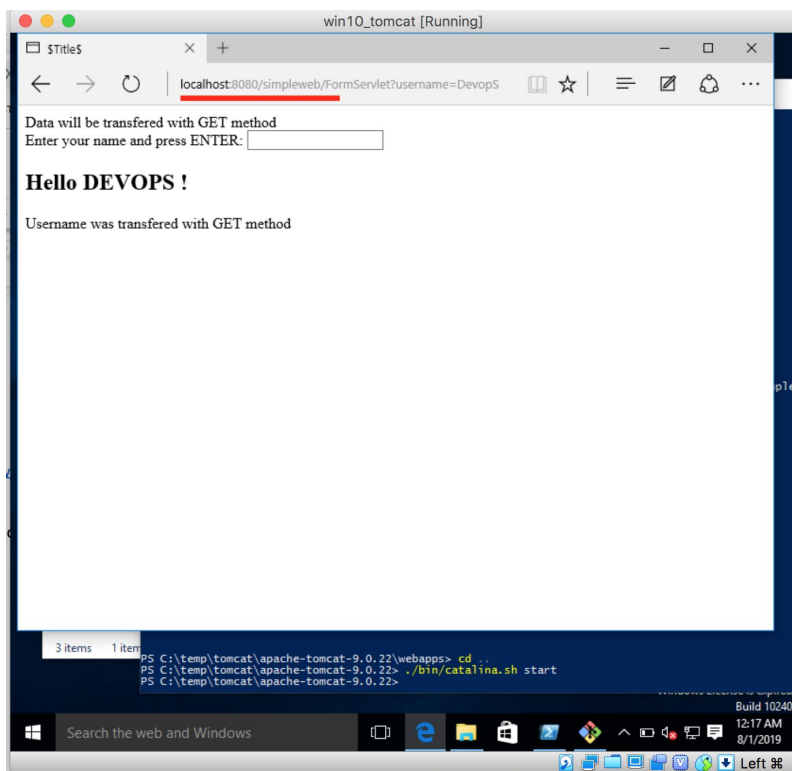
<https://github.com/dennis00010011b/simpleweb>

1.4. Deploy this application to Tomcat and demonstrate its functioning

For deploying I cloned app from gitHub repo:

```
sudo $CATALINA_HOME./bin/catalina.sh stop
cd $CATALINA_HOME/webapps
git clone https://github.com/dennis00010011b/simpleweb
sudo $CATALINA_HOME./bin/catalina.sh start
```

App available on URL localhost:8080/simpleweb



1.5 Pack this web application to a web archive & 1.6. Remove the deployed WEB application from Tomcat

Run from `./simpleweb`:

```
jar -cvf greeting.war *
```

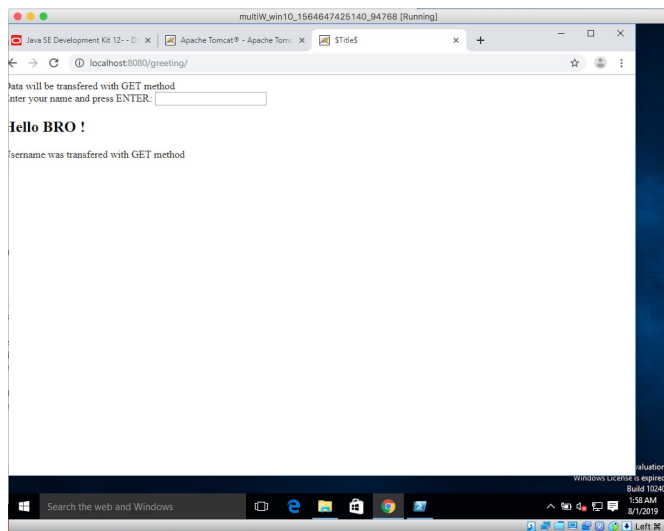
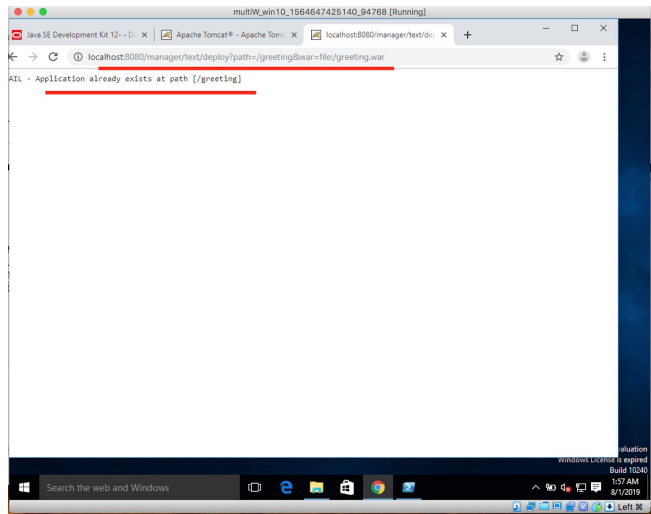
1.7. Deploy WEB application from WEB archive

Manager role `manager-script` MUST be allowed.

Deploying can be done with GET request to URL

```
localhost:8080/manager/text/deploy?path=/greeting&war=file:///greeting.war
```

I used GUI Chrome for make such GET request.



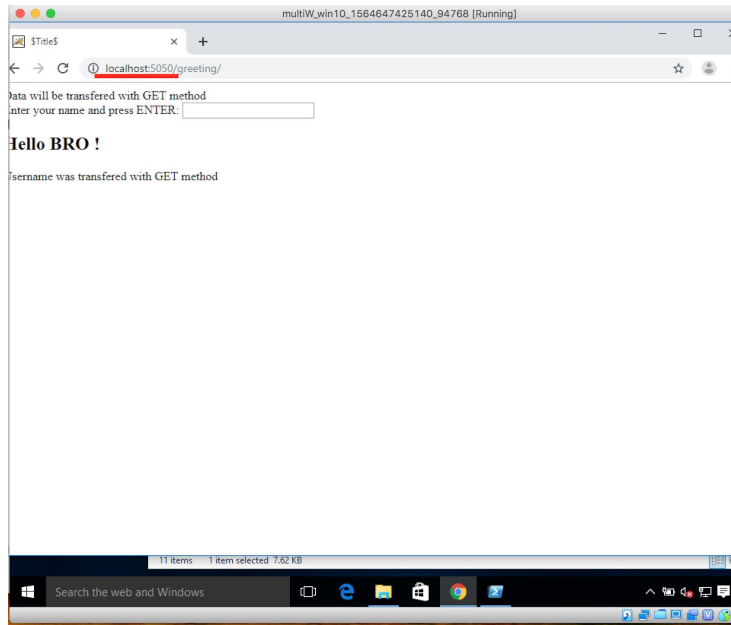
1.8. Change the Tomcat port to 5050

Add one more connector in `server.xml`:

```
<Connector port="5050" protocol="HTTP/1.1"
connectionTimeout="600000"
```

```
redirectPort="8443" />
```

Restart server



1.9. Change the default application directory (webapps) to another and check that the directory

has picked up (i.e. applications in this directory are accessible via http)

Need to change parameter `appBase` for the host in `server.xml` :

```
<Host name="localhost"  appBase="webapps1"
      unpackWARs="true" autoDeploy="true">
    <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" />
  </Host>
```

And provide folder `webapps1` with webapps.

1.10. Write command scripts in Windows OS and Linux OS to execute p.1.4-1.9

See above in p.1.4-1.9

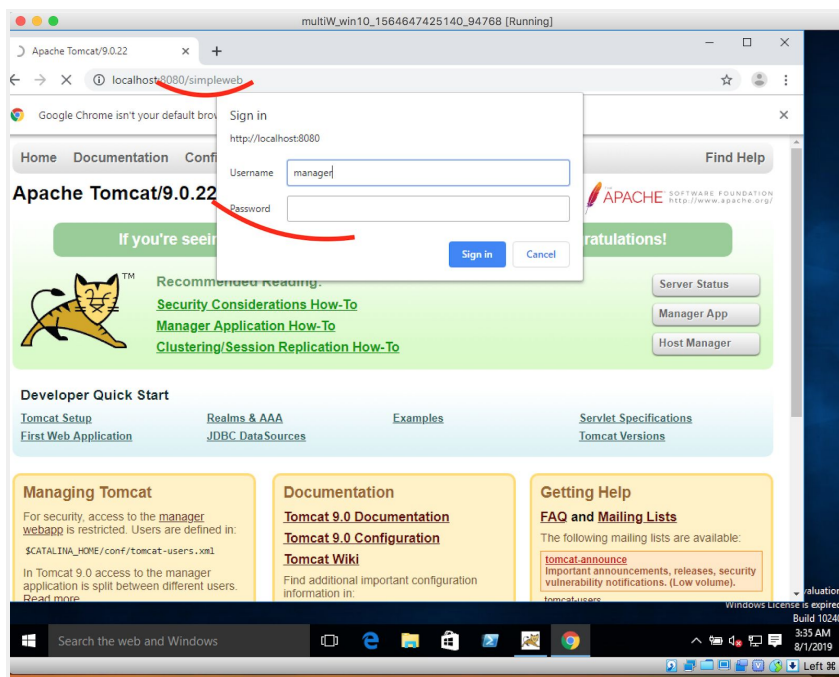
1.11. Configure Apache Tomcat authentication (standard HTTP authentication in Tomcat.doc)

HTTP authentication is configured in *web.xml* for given webapp:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>ApplicationContent</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>manager</role-name>
  </auth-constraint>
</security-constraint>

<security-role>
  <role-name>manager</role-name>
</security-role>

<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>SecuredApp</realm-name>
</login-config>
```



1.12. Configure SSL for Apache Tomcat

1. Create a new *JKS* keystore from scratch, containing a single self-signed Certificate, execute the following from a terminal command line:

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore  
$CATALINA_HOME
```

.keystore file was created, keys in JKS format, now we have a keystore file with a Certificate

2. Add connector to *server.xml* :

```
<Connector port="8443"  
protocol="org.apache.coyote.http11.Http11NioProtocol"  
    maxThreads="150"  
    scheme="https" secure="true" SSLEnabled="true"  
    keystoreFile="$CATALINA_HOME/conf/.keystore"  
    keystorePass="qwerty"  
    clientAuth="false" sslProtocol="TLS" />
```

3. Check SSL connection : <https://localhost:8443>

2. Apache HTTPd

Configuration files in `/etc/apache2/`

Webapp root directory is `/var/www/html/`

Control Apache : `/etc/init.d/apache2` or `/usr/sbin/apache2ctl`

LINUX VERSION (UBUNTU18.04)

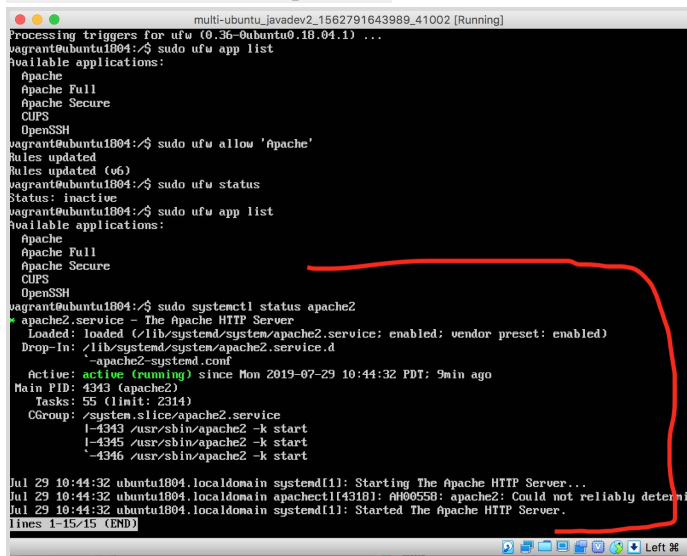
2.1. Install Apache HTTPd on Linux and carry out p.2.2-2.7 of the task

There is a nice aApache installation tutorial

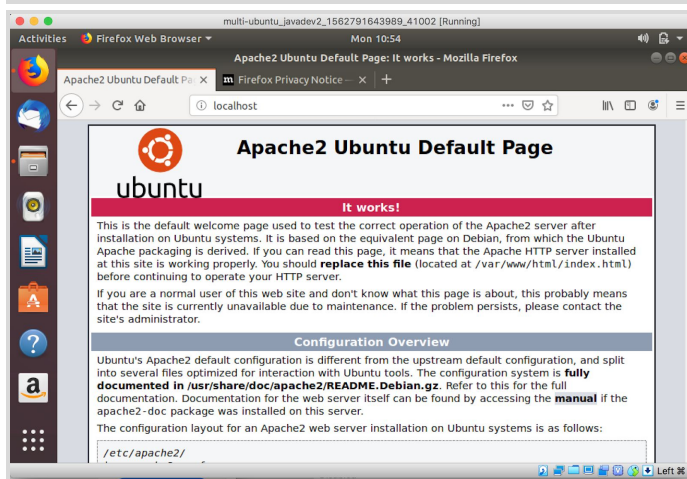
<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04-quickstart>

```
sudo apt-get install apache2
```

```
sudo ufw allow 'Apache'
```

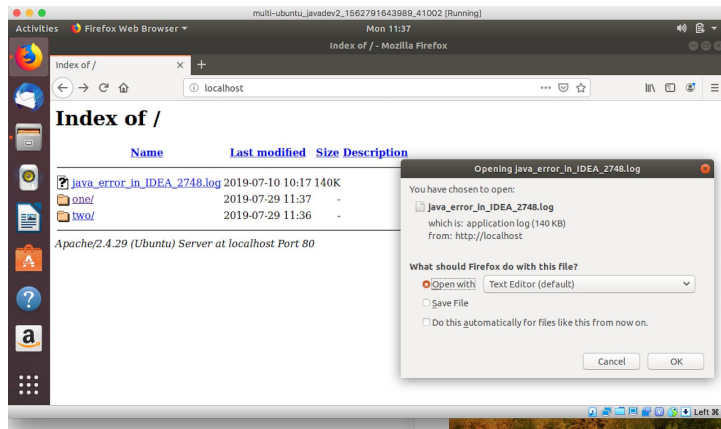


```
multi-ubuntu_javadev2_1562791643989_41002 [Running]
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
vagrant@ubuntu1804:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
vagrant@ubuntu1804:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
vagrant@ubuntu1804:~$ sudo ufw status
Status: inactive
vagrant@ubuntu1804:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
vagrant@ubuntu1804:~$ sudo systemctl status apache2
apache2.service - The Apache HTTP Server
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
Drop-In: /lib/systemd/system/apache2.service.d
         *-apache2-systemd.conf
Active: active (running) since Mon 2019-07-29 10:44:32 PDT; 9min ago
Main PID: 4343 (apache2)
Tasks: 55 (limit: 2314)
CGroup: /system.slice/apache2.service
         └─4343 /usr/sbin/apache2 -k start
           └─4345 /usr/sbin/apache2 -k start
             └─4346 /usr/sbin/apache2 -k start
Jul 29 10:44:32 ubuntu1804.localdomain systemd[1]: Starting The Apache HTTP Server...
Jul 29 10:44:32 ubuntu1804.localdomain apache2[4318]: AH00558: apache2: Could not reliably determine
Jul 29 10:44:32 ubuntu1804.localdomain systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```



2.2. Make the file available via http

Just copy files to `/var/www/html/`. May need to change access to this folder with `chmod` command



2.3. Change root directory

It suppose to change document root?

If yes, then modify path in `/etc/apache2/sites-available/000-default.conf`:

DocumentRoot `/var/www/www1`

and in file `/etc/apache2/apache2.conf` to rewrite this piece of code:

```
<Directory /usr/www/www1/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

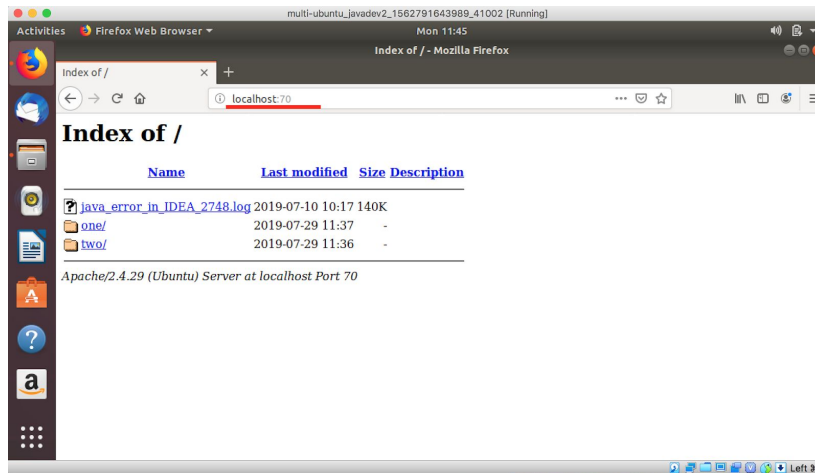
2.4. Change apache port to 70

`/etc/apache2/ports.conf` used to determine the listening port. For change port number just rewrite and restart server :

Listen `70`

and restart server :

```
sudo systemctl restart apache2
```



2.5. Make not root directory accessible via http (Alias)

Aliases defined in `/etc/apache2/mods-available/alias.conf`:

```
Alias "/somewhere" "/home/notRoot"
<Directory"/home/notRoot">
    Require all granted
</Directory>
```

2.6. Configure Apache HTTP authentication (Apache basic authentication setup.doc)

First of all create pair user:password :

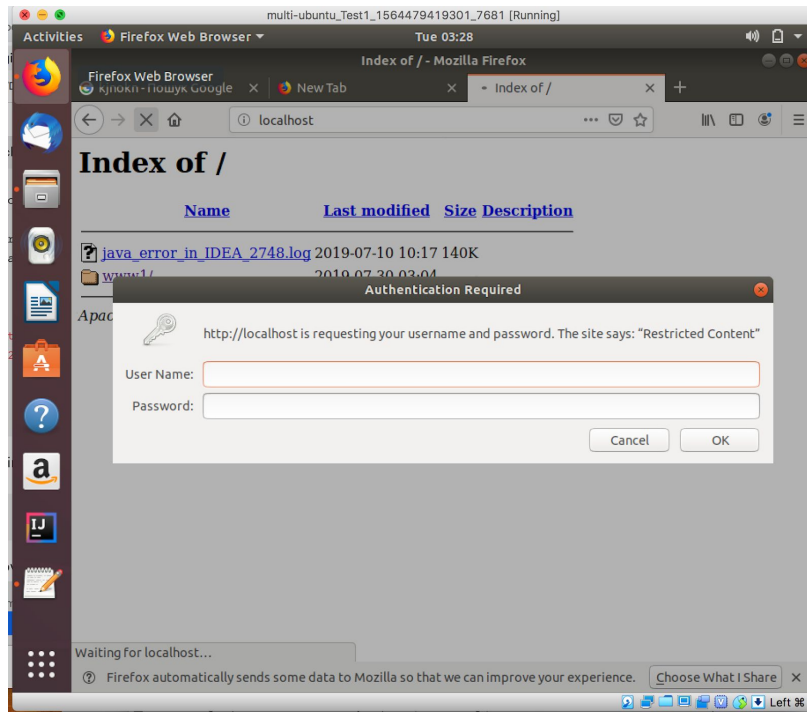
```
sudo apt-get install apache2 apache2-utils #need utils htpasswd
sudo htpasswd -c /etc/apache2/.htpasswd user1
```

Next to add desirable directory in host file `/etc/apache2/sites-enabled/000-default.conf`:

```
<Directory "/var/www/html">
    AuthType Basic
    AuthName "Restricted Content"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>
```

This configuration allows access for all users. For restrict access it needs to specify group of user with AuthGroupFile directive

https://httpd.apache.org/docs/2.4/mod/mod_authz_groupfile.html#authgroupfile



2.7. Configure SSL for Apache HTTP (Apache and SSL.doc)

Steps to config SSL described here <https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html>

1. Create CA:

```
openssl req -x509 -newkey rsa:4096 -keyout /etc/apache2/key.pem -out
/etc/apache2/cert.pem -days 365
```

2. Create new VirtualHost with SSL configuration (I used default `/etc/apache2/sites-available/default-ssl.conf`) and place it to folder `/etc/apache2/sites-available`:

```
<VirtualHost *:443>
    DocumentRoot /var/www/html
    SSLEngine on
    SSLCertificateFile /etc/apache2/cert.pem
    SSLCertificateKeyFile /etc/apache2/key.pem
</VirtualHost>
```

3. Enable this site:

```
sudo a2ensite default-ssl.conf
```

File `/etc/apache2/sites-available/default-ssl.conf` will be copied to `/etc/apache2/sites-enabled/default-ssl.conf`

4. Enable apache ssl mode:

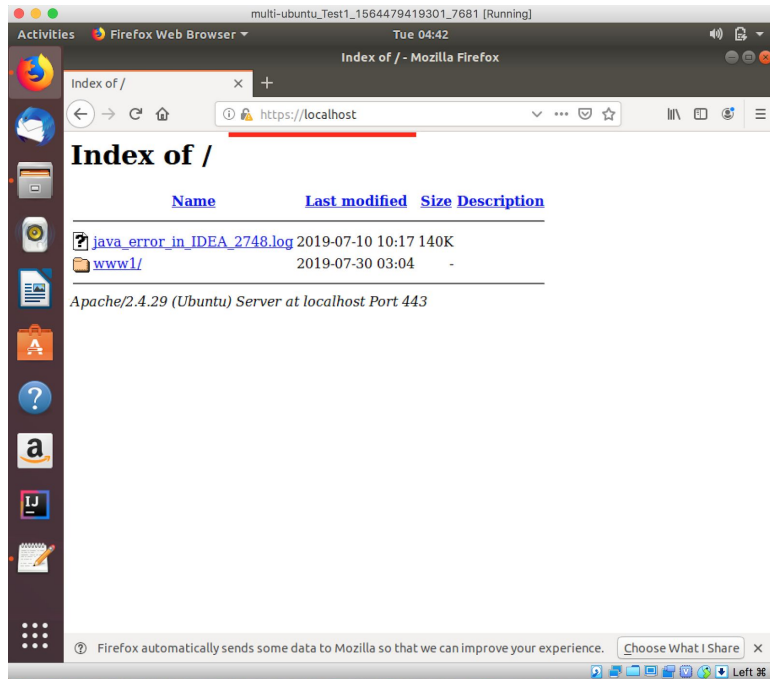
```
sudo a2enmod ssl
```

5.Reload apache2:

```
sudo systemctl reload apache2
```

6. Check secure connection in browser:

```
https://localhost
```



2.8.2. On the second computer, install only Apache Tomcat.



```
apt-get install libapache2-mod-jk
```

JK log file `/var/log/apache2/mod_jk.log`

Server	IP address	Port	AJP port	VM name
Apache2 httpd	192.168.50.12	70	n/a	ubuntu_httpd_to_mcat
Tomcat 9	192.168.50.12	8080	8009	ubuntu_httpd_to_mcat
Tomcat 9	192.168.50.2	8080	8009	ubuntu_tomcat

Config `/etc/libapache2-mod-jk/workers.properties`:

```
workers.tomcat_home=/etc/tomcat/apache-tomcat-9.0.22
workers.java_home=/usr/lib/jvm/java-8-openjdk-amd64

worker.list=loadbalancer,status

# Define Node1
worker.tomcat.port=8009
worker.tomcat2.host=192.168.50.2
worker.tomcat2.type=ajp13
worker.tomcat2.lbfactor=1

# Define Node2
worker.tomcat12.port=8009
worker.tomcat12.host=192.168.50.12
worker.tomcat12.type=ajp13
worker.tomcat12.lbfactor=1

# Load-balancing behavior
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=tomcat2,tomcat12

# Status worker for managing load balancer
worker.status.type=status
```

Modify redirection in ApacheHTTPd VirtualHost file

`/etc/apache2/sites-available/000-default.conf`:

```
JkMount /greeting/* loadbalancer
JkMount /greeting lloadbalancer
#mod_jk status page
JkMount /status status
```

2.8.4. Deploy a web application from section 1.5

Did it. See next step.

2.8.5. Demonstrate logs of query execution taking into account balancing

Load balancer 1/2

JK Status Manager for localhost:70

Server Version: Apache/2.4.29 (Ubuntu) mod_jk/1.2.43 OpenSSL/1.1.1 Server Time: 2019-07-31 03:38:51 -0700
JK Version: mod_jk/1.2.43 Unix Seconds: 1564569531

Start auto refresh (every 10 seconds) | Change format XML

[Read Only] [Dump] [S=Show only this worker, E=Edit worker, R=Reset worker state, T=Try worker recovery]

Listing Load Balancing Worker (1 Worker) [Hide]

[S][E][R] Worker Status for loadbalancer

Sticky	Force Sticky	LB	Recover	Error	Max Reply
Type Sessions	Sessions	Retries Method	Locking Wait Time	Escalation Time	Timeouts
lb	True	False	2	Request Optimistic:60	30
					0

Good Degraded Bad/Stopped Busy Max Busy Next Maintenance Last Reset [Hide]

2	0	0	0	1	48/108	135
---	---	---	---	---	--------	-----

Balancer Members [Hide]

Name	Type	Hostname	Address:Port	Source	Pool	Connection Timeout	Connect Prepost	Reply	Recovery	Busy	Max Packet
tomcat2	ajp13	192.168.50.2	192.168.50.2:8009	undefined	0	0	0	0	2	0	8192
tomcat12	ajp13	192.168.50.12	192.168.50.12:8009	undefined	0	0	0	2	0	0	8192

Name	Act	State	DFM	V	Acc	Sess	Err	CERE	Wr	Rd	Busy	Max	Busy	Con	Max	Con	Route	RR	Cd	Rs	LR	IE	
[S][E][R] tomcat2	ACT	OK	0	2	1	12	12	(0/sec)	0	0	0	8.1K	(61/sec)	5.7K	(43/sec)	0	1	1	1	1	1	0/0	135
[S][E][R] tomcat12	ACT	OK	0	1	2	12	6	(0/sec)	0	0	0	4.1K	(30/sec)	3.0K	(22/sec)	0	1	1	1	1	1	0/0	135

Edit this attribute for all members: Activation Go

URI Mappings for loadbalancer (2 maps) [Hide]

Server	Match	URI	Type	Source	Reply	Sticky	Fail on	Use Server														
ubuntu1804.localdomain	[*?70]/greeting/*	Wildcard	jkMount	-1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Not Secure | 192.168.50.12:70/greeting/

IT IS TOMCAT IP 192.168.50.2 !!(?!(*^(*&@)(&
Data will be transferred with GET method
Enter your name and press ENTER:

Hello BRO !

Username was transferred with GET method

