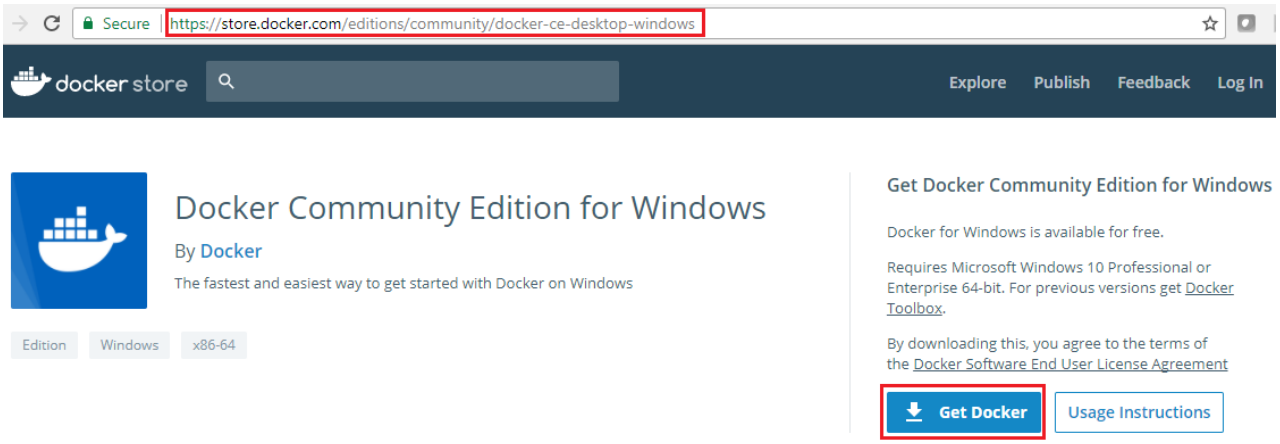
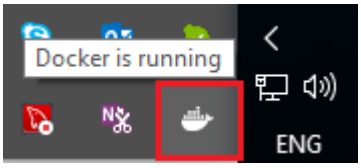
How to deploy Java Web App in Docker container with Ubuntu+Tomcat+MySQL

Install Docker

1) Download Docker installation package from official site. For example, for Windows it will be next link https://store.docker.com/editions/community/docker-ce-desktop-windows



2) Install Docker using downloaded **Docker for Windows Installer.exe**
3) Run Docker. As a result you should see next ico



Create Docker container with Ubuntu

1) Open PowerShell

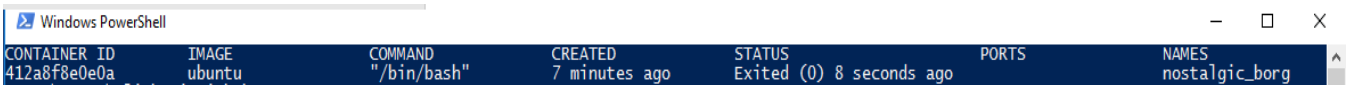
2) Setup Ubuntu image by using next command:

```
docker run -it ubuntu
```

As a result Docker will download Ubuntu, create container with Ubuntu and random name and open command line. To stop this container you should type **exit** and press **Enter**.

To see this new container you can use command which shows you all created containers:

`docker ps –a`

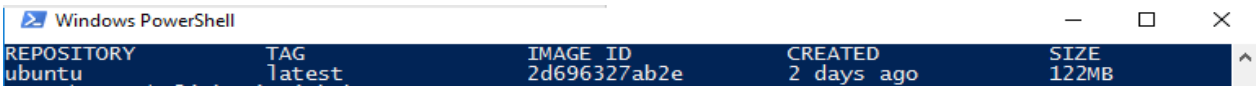


You can remove this default Ubuntu container by using **CONTAINER_ID** and next command:

`docker rm –f 412a8f8e0e0a`

Also you can check that now we have Ubuntu image by using next command:

`docker images`



And now we can create docker containers with Ubuntu based on this image.

3) Create Ubuntu container with name (ex.: **c1**) and connection between localhost's port 80 and container's port 8080 (tomcat port) by using next command:

`docker run --name c1 -it -p 80:8080 ubuntu`

As a result Docker creates new Ubuntu container with name **c1**, run it and open command line.

4) Also we need update and install several tools which were needed in next steps:

4.1 – Update apt-get tool (package installer):

`apt-get update`

4.2 – Install nano (linux text editor):

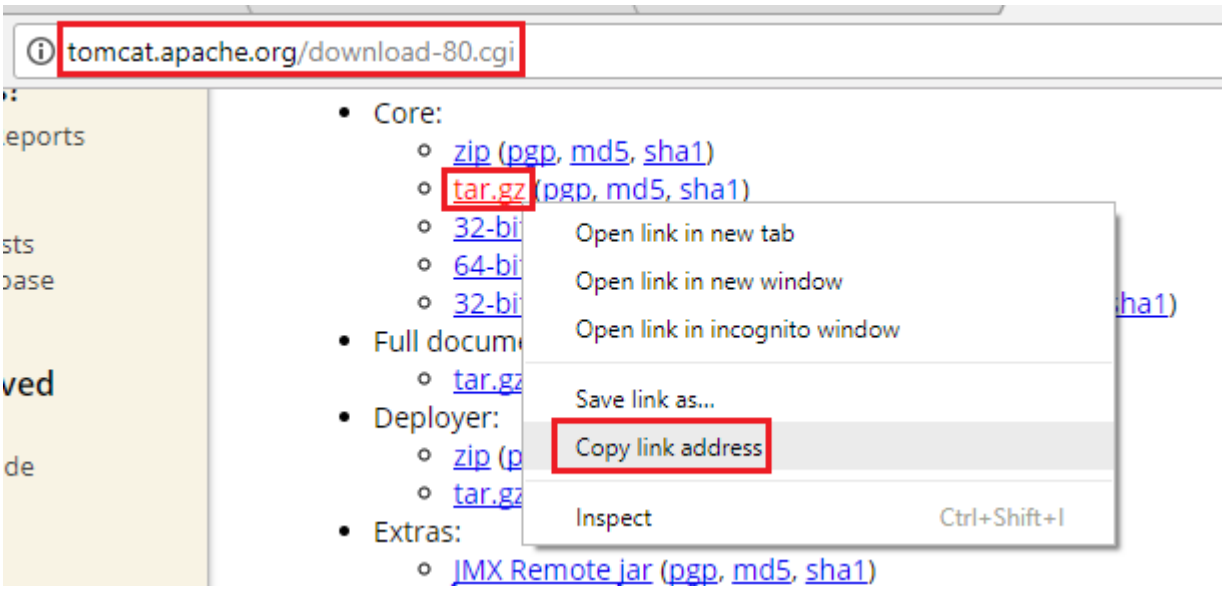
`apt-get install nano`

4.3 – Install wget (downloader)

`apt-get install wget`

Install and configure Apache Tomcat

1) We should copy link to **tar.gz** archive with latest tomcat version. For this purpose we should use next link http://tomcat.apache.org/download-80.cgi



2) Use copied link to download this archive via wget tool by using next command:

`wget <copied link>`

After this you can use **ls** command to that *.tar.gz archive with tomcat was downloaded and placed in current directory:

3) Extract tomcat files from downloaded archive by using command:

`tar xvzf <archive file name>`

For example, in this case:
`tar xvzf apache-tomcat-8.5.20.tar.gz`

Now we can see folder apache-tomcat-8.5.20 with extracted files by using **ls** command:

4) Move files from **apache-tomcat-8.5.20** folder to **/opt/tomcat** folder by using next command:

```
mv apache-tomcat-8.5.20 /opt/tomcat
```

5) Install Java by using command:

```
apt-get install openjdk-8-jdk
```

6) Set **JAVA_HOME** and **CATALINA_HOME** environmental variables:

6.1 – Open file **.bashrc** in nano text editor by using command:

```
nano ~/.bashrc
```

6.2 – Enter next 2 lines to the end of this file:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export CATALINA_HOME=/opt/tomcat
```

6.3 – Press **Ctrl + X** and next **Y** to save and close this file

6.4 – Apply this changes by using command

```
. ~/.bashrc
```

6.5 – Try to start and stop tomcat by using next commands:

```
$CATALINA_HOME/bin/startup.sh
$CATALINA_HOME/bin/shutdown.sh
```

7) Configure tomcat to be able to use **Tomcat App Manager** and **Tomcat Host Manager**

7.1 – Open **tomcat-users.xml** in nano text editor by using command:

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

7.2 – Add user (ex.: tomcat / tomcat) with 2 necessary roles as listed below:

```
<tomcat-users . . .>
      <role rolename="manager-gui"/>
      <role rolename="admin-gui"/>
```

```
    <user username="tomcat" password="tomcat" roles="manager-gui,admin-gui"/>
</tomcat-users>
```

7.3 – Press **Ctrl + X** and next **Y** to save and close this file

7.4 – Open **context.xml** for **Tomcat App Manager** in nano text editor by using command:

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

7.5 – Delete all content between **context** tags:

```
<Context antiResourceLocking="false" privileged="true">
    // Should be empty
</Context>
```

7.6 – Press **Ctrl + X** and next **Y** to save and close this file

7.7 – Open **context.xml** for **Tomcat Host Manager** in nano text editor by using command:

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

7.8 – Delete all content between **context** tags:

```
<Context antiResourceLocking="false" privileged="true">
    // Should be empty
</Context>
```

7.9 – Press **Ctrl + X** and next **Y** to save and close this file

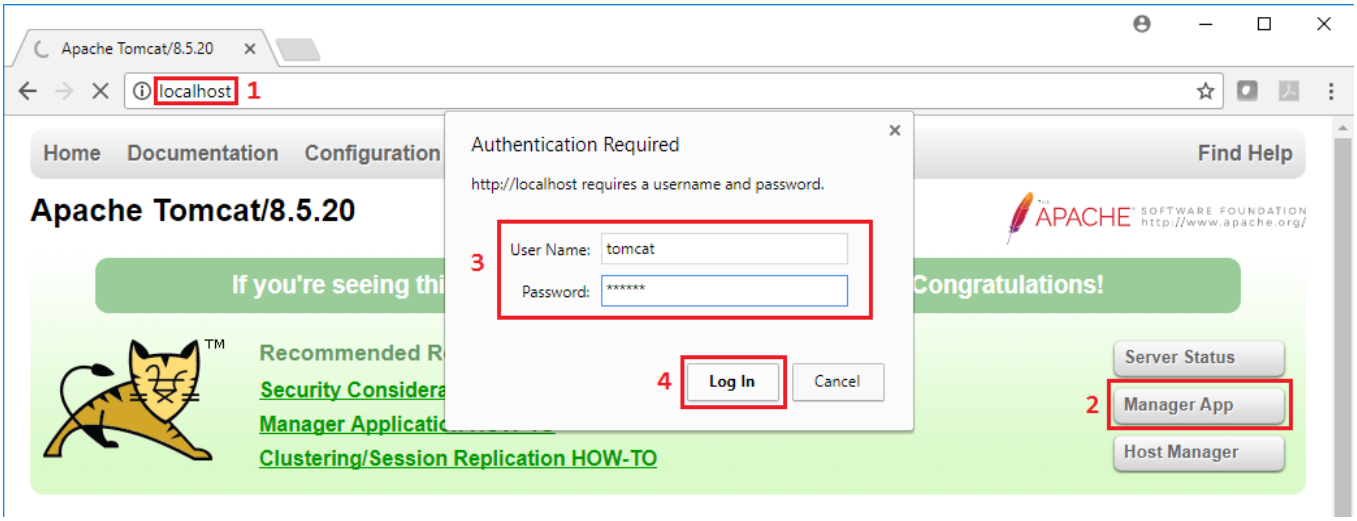
8) Start tomcat and try to open Tomcat App Manager

8.1 – Start tomcat by using next command:

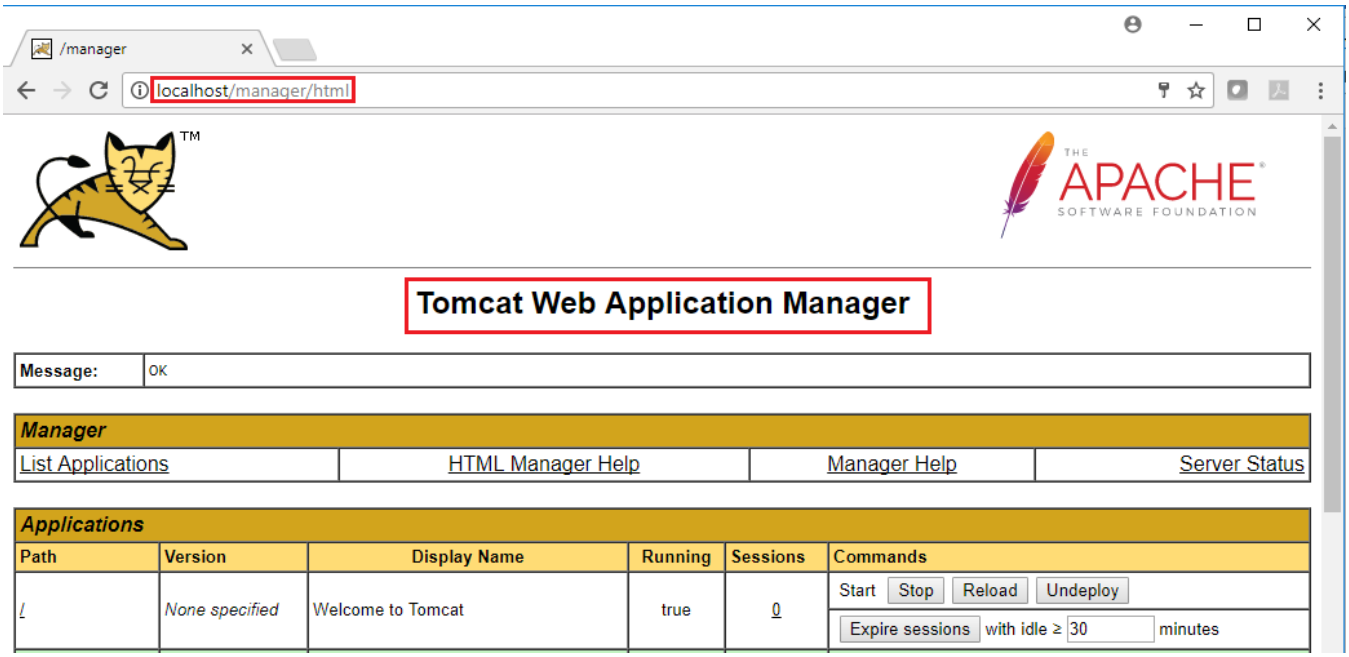
```
$CATALINA_HOME/bin/startup.sh
```

8.2 – Open tomcat start page on host machine by querying **localhost** via web-browser

8.3 – Click **Manager App** button and enter manager credentials (**tomcat / tomcat**) which we have already set on **step 7.2**



As a result we should see next page



Install MySQL Server

1) Install MySQL server by using command:

```
apt-get install mysql-server
```

During installation you should enter password for root user.
You can change it in the future.

2) Now you can start and stop MySQL Server by using next commands:

```
service mysql start
service mysql stop
```

Also you can check status (started or stoped) by using next command:

```
service mysql status
```

3) Try to make queries to MySQL Server to be sure that it works fine

3.1 – Start MySQL Server by using command:

```
service mysql start
```

3.2 – Open MySQL command line by using next command:

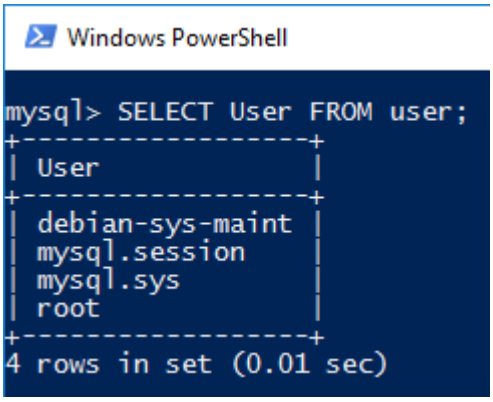
```
mysql -p -u root
```

And next enter password for root user which you already entered during installation of MySQL Server

3.3 – Enter next SQL commands to see password for root user

```
use mysql;
SELECT User FROM user;
```

As a result you should see next output:



3.4 – To exit MySQL command line you should use **exit** command.

Deploy and run Java Web Application on Tomcat+MySQL

For example, we have a Java Web Application builded as .war file (ex.: named as **app.war**). And this app has configured connection to DB on MySQL server.

1) Create necessary DB on MySQL server.

1.1 – We will create this DB by running ***.sql** script (ex.: **app-db.sql**) which contains all necessary sql-queries to create DB and Tables. So we need to upload our app-db.sql script from host machine to our Ubuntu container by using tools like WinSCP and etc. or we can create it manually.

1.2 – To create this script manually we should create file with name **app-db.sql** by using command:

```
touch app-db.sql
```

1.3 – Open this file via nano text editor by using command

```
nano app-db.sql
```

1.4 – Copy and paste all necessary sql-queries from **.sql** script file on host machine into this file.

1.5 – Press **Ctrl + X** and next **Y** to save and close this file.

1.6 – Start MySQL Server by using next command

```
service mysql start
```

1.7 – Run **app-db.sql** script which we have already created to create necessary DB and Tables by using command:

```
mysql –p –u root < app-db.sql
```

1.8 – Try to make queries to created DB to be sure that all created fine.

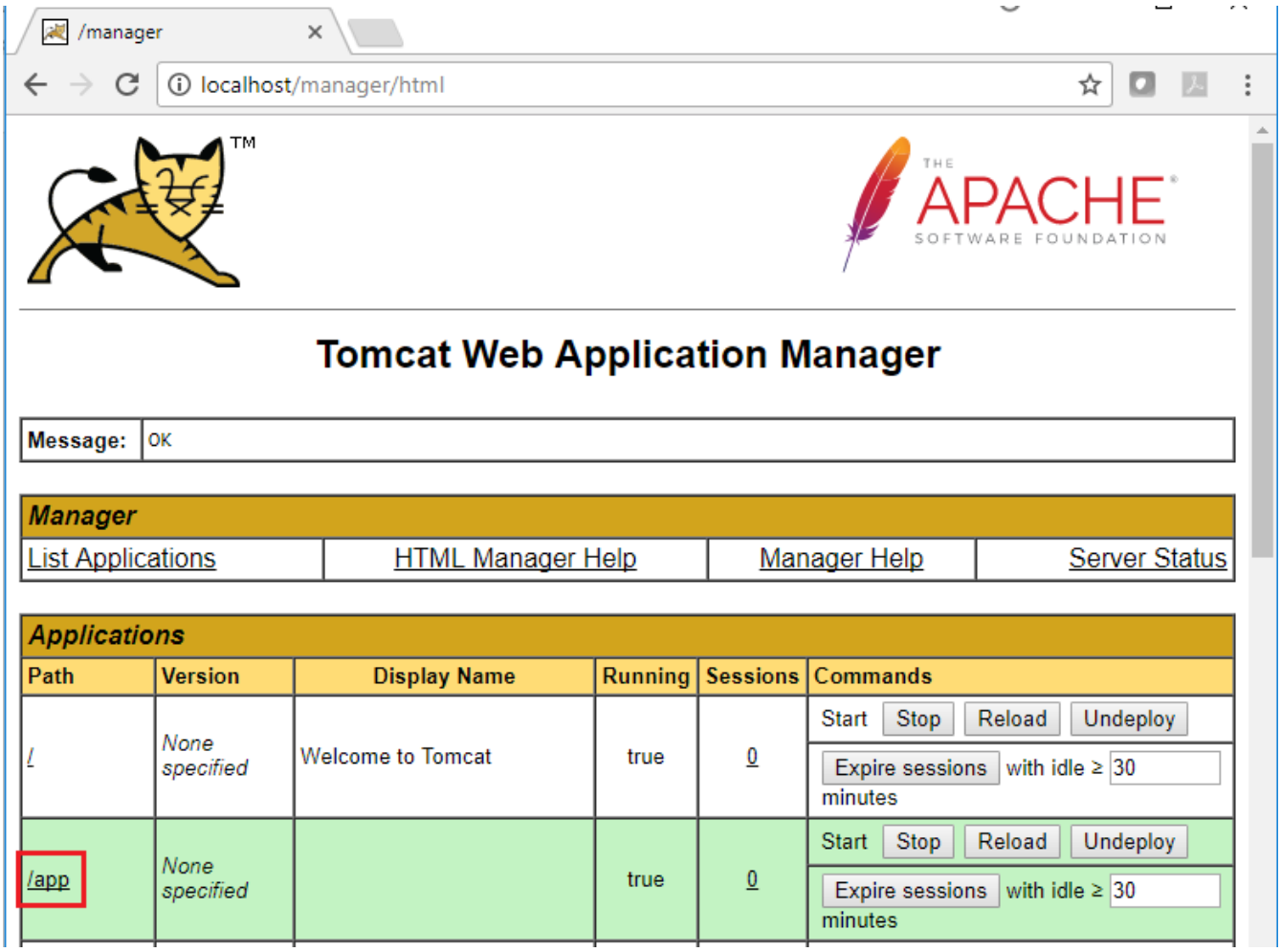
1.9 – Leave MySQL command line by using **exit** command.

2) Deploy your Java Web App by using Tomcat App Manager

2.1 – Open **Manager App** by using step 8 of **Install and configure Apache Tomcat** section

2.2 – Click **Choose File** button in **Deploy > WAR file to deploy** section, choose necessary .war (app.war in this example) file and click **Deploy** button. Next Tomcat will put this file into webapps directory and deploy it.

As a result we will see our app in Applications section:



And it means that we can open our Java Web Application via web-browser by using next address:

http://localhost/app

That's all.