Glassfish setup

JAVA

Java was not installed initially

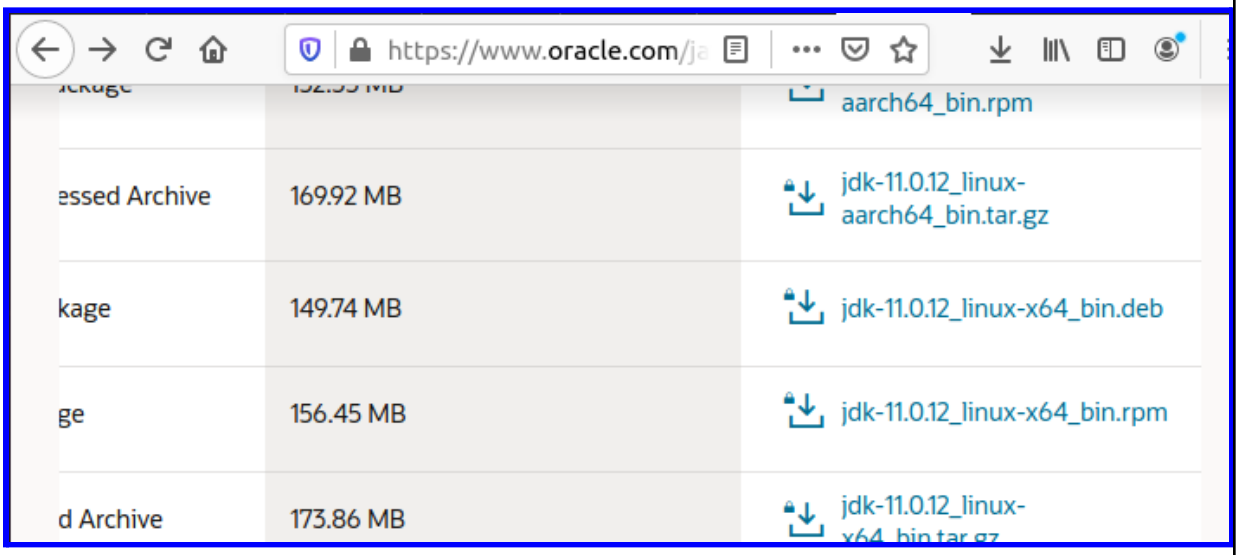
```
bibek@bibek-lfTechnology:~$ java --version

Command 'java' not found, but can be installed with:

sudo apt install openjdk-11-jre-headless  # version 11.0.11+9-0ubuntu2~20.04, o
r
sudo apt install default-jre              # version 2:1.11-72
sudo apt install openjdk-16-jre-headless  # version 16.0.1+9-1~20.04
sudo apt install openjdk-8-jre-headless   # version 8u292-b10-0ubuntu1~20.04
sudo apt install openjdk-13-jre-headless  # version 13.0.7+5-0ubuntu1~20.04
sudo apt install openjdk-17-jre-headless  # version 17+35-1~20.04
```

To install **oracle-jdk-11**

Download the debian package from oracle website



sudo dpkg -i <debian-package-name>

```
bibek@bibek-lfTechnology:~/Downloads$ sudo dpkg -i jdk-11.0.12_linux-x64_bin.de
b
[sudo] password for bibek:
Selecting previously unselected package jdk-11.0.12.
(Reading database ... 175544 files and directories currently installed.)
Preparing to unpack jdk-11.0.12_linux-x64_bin.deb ...
Unpacking jdk-11.0.12 (11.0.12-1) ...
Setting up jdk-11.0.12 (11.0.12-1) ...
```

After successful installation, few more steps for auto-mode and setting the java and javac path

sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-11.0.12/bin/java 1

```
bibek@bibek-lfTechnology:~/Downloads$ sudo update-alternatives --install /usr/b
in/java java /usr/lib/jvm/jdk-11.0.12/bin/java 1
update-alternatives: using /usr/lib/jvm/jdk-11.0.12/bin/java to provide /usr/bi
n/java (java) in auto mode
```

sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk-11.0.12/bin/javac 1

```
bibek@bibek-lfTechnology:~/Downloads$ sudo update-alternatives --install /usr/b
in/javac javac /usr/lib/jvm/jdk-11.0.12/bin/javac 1
update-alternatives: using /usr/lib/jvm/jdk-11.0.12/bin/javac to provide /usr/b
in/javac (javac) in auto mode
```

To view the java and javac version

java --version

javac --version

```
bibek@bibek-lfTechnology:~/Downloads$ java --version
java 11.0.12 2021-07-20 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.12+8-LTS-237)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.12+8-LTS-237, mixed mode)
bibek@bibek-lfTechnology:~/Downloads$ javac --version
javac 11.0.12
```

To set config of java

sudo update-alternatives --config java

To set environments

sudo gedit /etc/environments

And pate the path of JAVA_HOME to its path

```
1 PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/-
  games:/usr/local/games:/snap/bin"
2 JAVA_HOME="/usr/lib/jvm/jdk-11.0.12"
```

Save and exit

To refresh the environment

source /etc/environment

Now we can echo the home path of java

```
bibek@bibek-lfTechnology:~/Downloads$ echo $JAVA_HOME
/usr/lib/jvm/jdk-11.0.12
```

GlassFish

To install glassfish6

Download the zip file of glassfish6.2.2

wget https://github.com/eclipse-ee4j/glassfish/releases/download/6.2.2/glassfish-6.2.2.zip

We can view the zip file using "**ls" command**

And then unzip it

unzip glassfish-6.2.2.zip

```
bibek@bibek-lfTechnology:~$ wget https://github.com/eclipse-ee4j/glassfish/releas
--2021-11-13 21:35:38--  https://github.com/eclipse-ee4j/glassfish/releases/downl
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-releases.githubusercontent.com/148886237/ac9792e0-2716-4
X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20211113%2Fus-east-1%2Fs3%2Faws4_request&
nature=1a0ac327d191de249a98000d3914b50e6eaf3f8fa6e4994c0240da2464ac5490&X-Amz-Sig
esponse-content-disposition=attachment%3B%20filename%3Dglassfish-6.2.2.zip&respon
--2021-11-13 21:35:44--  https://github-releases.githubusercontent.com/148886237/
S4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20211113%2Fus-east-1%2Fs3%
s=300&X-Amz-Signature=1a0ac327d191de249a98000d3914b50e6eaf3f8fa6e4994c0240da2464a
_id=148886237&response-content-disposition=attachment%3B%20filename%3Dglassfish-6
am
Resolving github-releases.githubusercontent.com (github-releases.githubuserconten
.154, ...
Connecting to github-releases.githubusercontent.com (github-releases.githubuserco
HTTP request sent, awaiting response... 200 OK
Length: 118186239 (113M) [application/octet-stream]
Saving to: 'glassfish-6.2.2.zip'

glassfish-6.2.2.zip               100%[=========================================

2021-11-13 21:36:53 (1.64 MB/s) - 'glassfish-6.2.2.zip' saved [118186239/11818623

bibek@bibek-lfTechnology:~$ ls
Desktop  Documents  Downloads  glassfish-6.2.2.zip  Music  Pictures  Public  Temp
bibek@bibek-lfTechnology:~$ unzip glassfish-6.2.2.zip
Archive:  glassfish-6.2.2.zip
   creating: glassfish6/
   creating: glassfish6/glassfish/
```

We can see glassfish6 directory now using "**ls" command**

Go to the bin directory glassfish6/bin and start the domain to use glassfish

cd glassfish6/bin

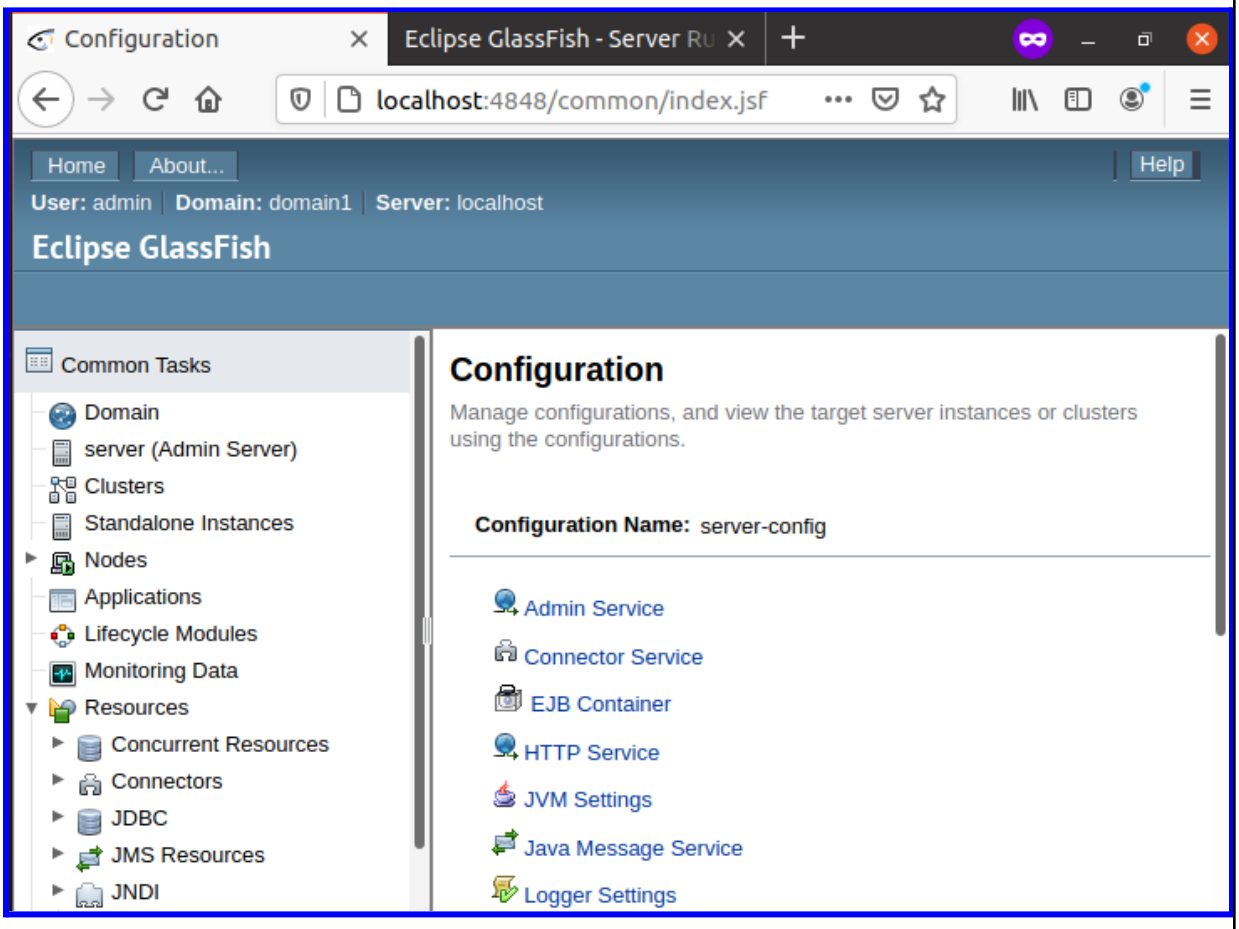
And run the asadmin script

./asadmin start-domain

```
bibek@bibek-lfTechnology:~$ ls
Desktop  Documents  Downloads  glassfish6  glassfish-6.2.2.zip  Music  Pictures
bibek@bibek-lfTechnology:~$ cd glassfish6/
bibek@bibek-lfTechnology:~/glassfish6$ ls
bin  glassfish  javadb  META-INF  mq  README.txt
bibek@bibek-lfTechnology:~/glassfish6$ cd bin/
bibek@bibek-lfTechnology:~/glassfish6/bin$ ls
asadmin  asadmin.bat  debug-asadmin  debug-asadmin.bat
bibek@bibek-lfTechnology:~/glassfish6/bin$ ./asadmin start-domain
Waiting for domain1 to start ................................
Successfully started the domain : domain1
domain  Location: /home/bibek/glassfish6/glassfish/domains/domain1
Log File: /home/bibek/glassfish6/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
```

After successful start message

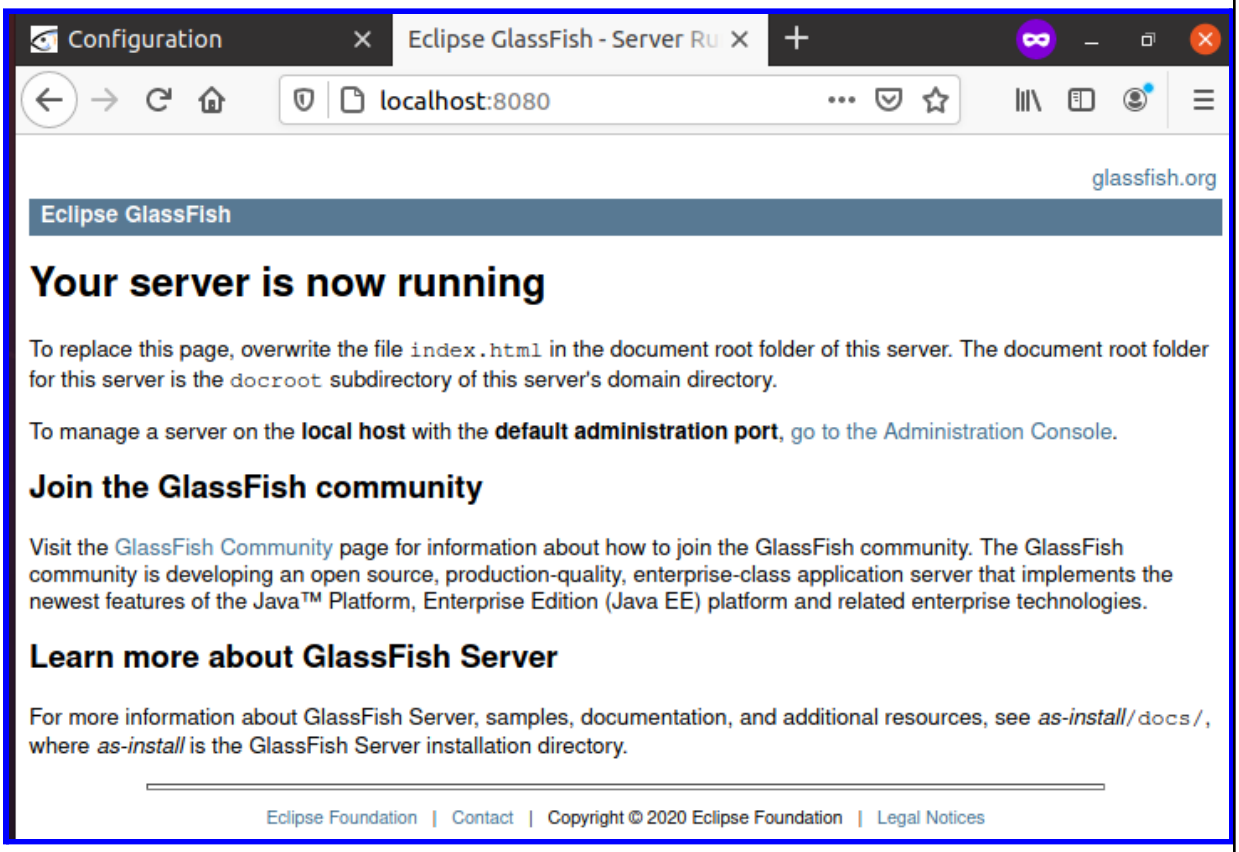
We can check the glassfish page on localhost at admin port (default:4848) in browser



Note: wait for sometime if it has not get started

To change http listener port to 8088

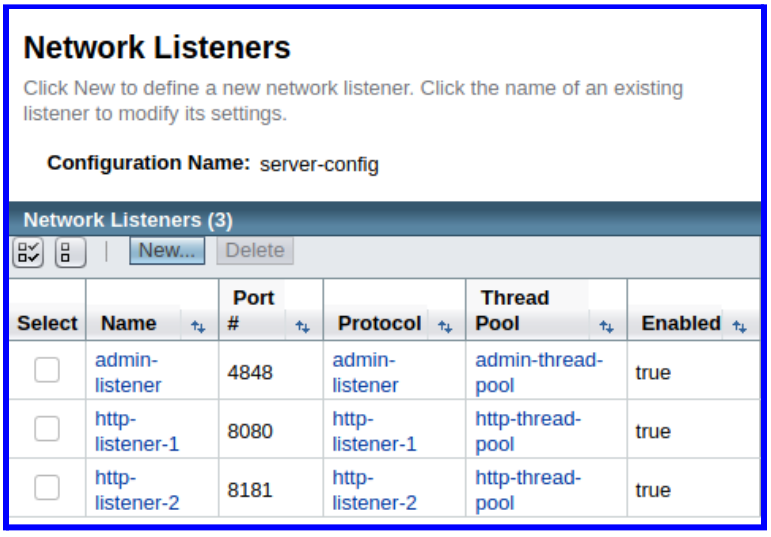
The default http listener port of glassfish is 8080



To change the listener port

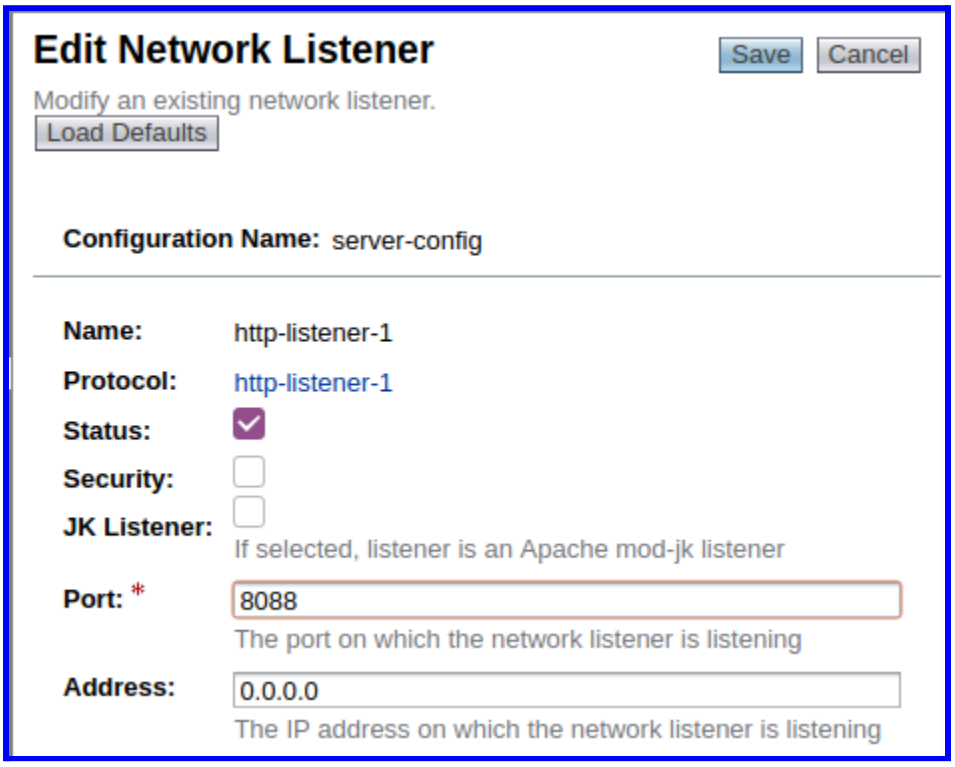
Go to admin panel

Click on server-config (at bottom) >> Network Listeners



We can see here the default http listener 1 port is 8080

Click on http-listener-1, change the port to 8088 and click on save (top-right-corner)



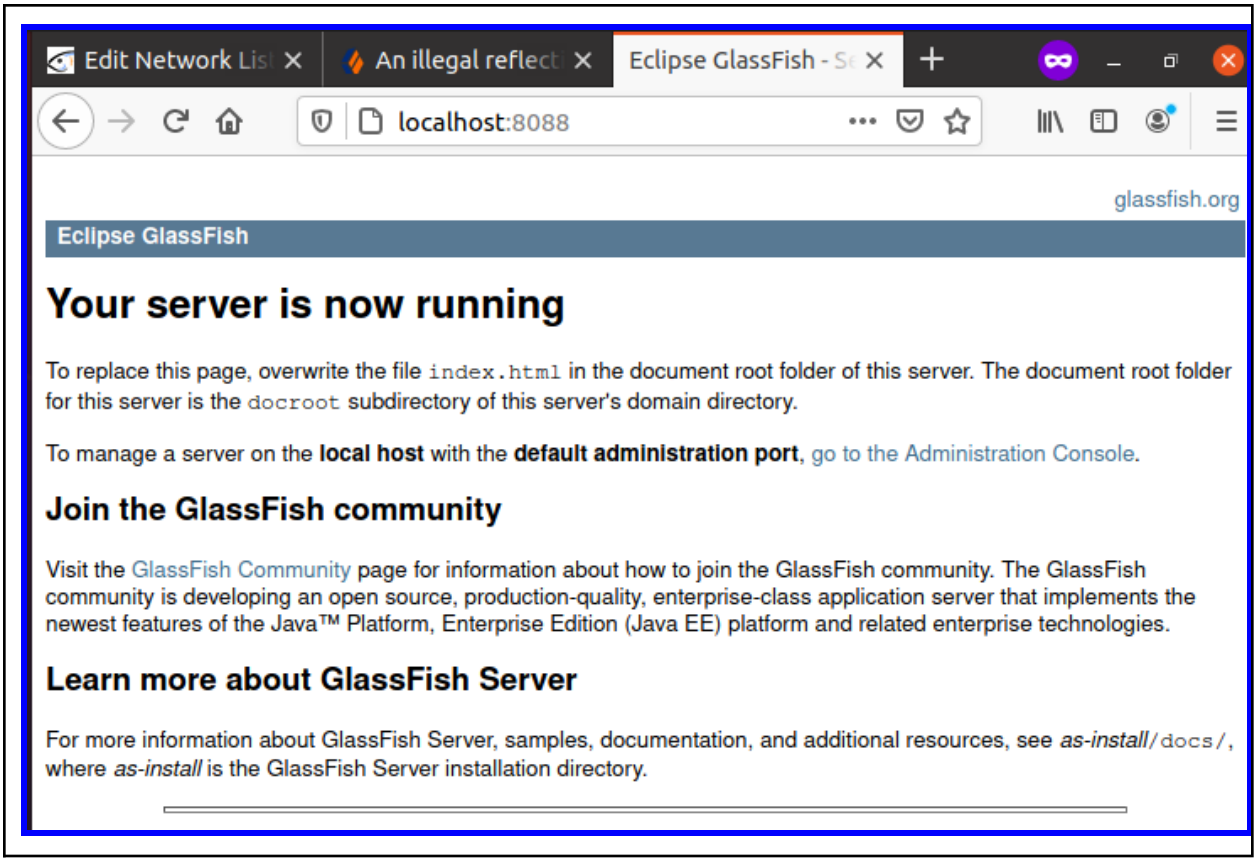
You can open the HTTP listener 1 on port 8088 of localhost.

- If it does not opens then stop the glassfish and again start it from the bin directory

./asadmin stop-domain

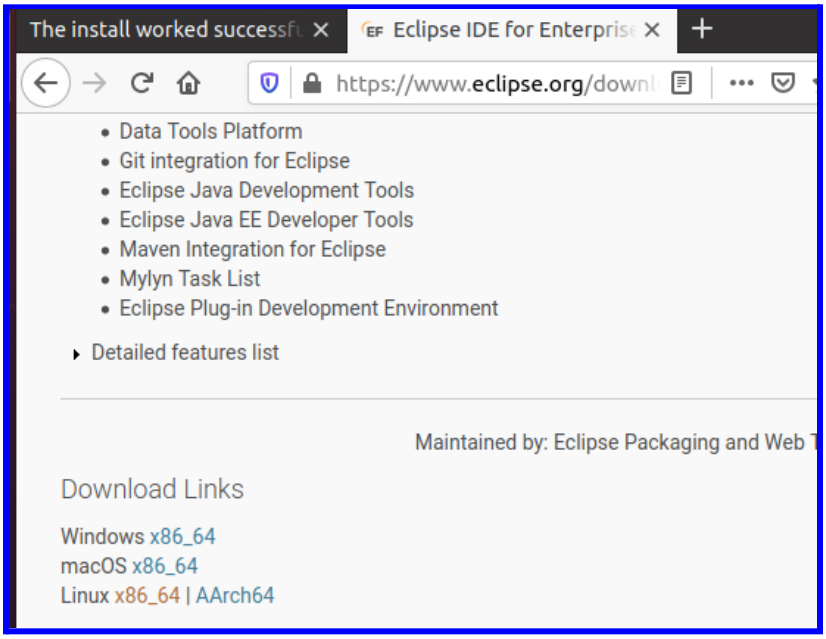
./asadmin start-domain

```
bibek@bibek-lfTechnology:~/glassfish6/bin$ ./asadmin stop-domain
Waiting for the domain to stop .....................................
Command stop-domain executed successfully.
bibek@bibek-lfTechnology:~/glassfish6/bin$ ./asadmin start-domain
Waiting for domain1 to start .....................................
Successfully started the domain : domain1
domain  Location: /home/bibek/glassfish6/glassfish/domains/domain1
Log File: /home/bibek/glassfish6/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
bibek@bibek-lfTechnology:~/glassfish6/bin$
```



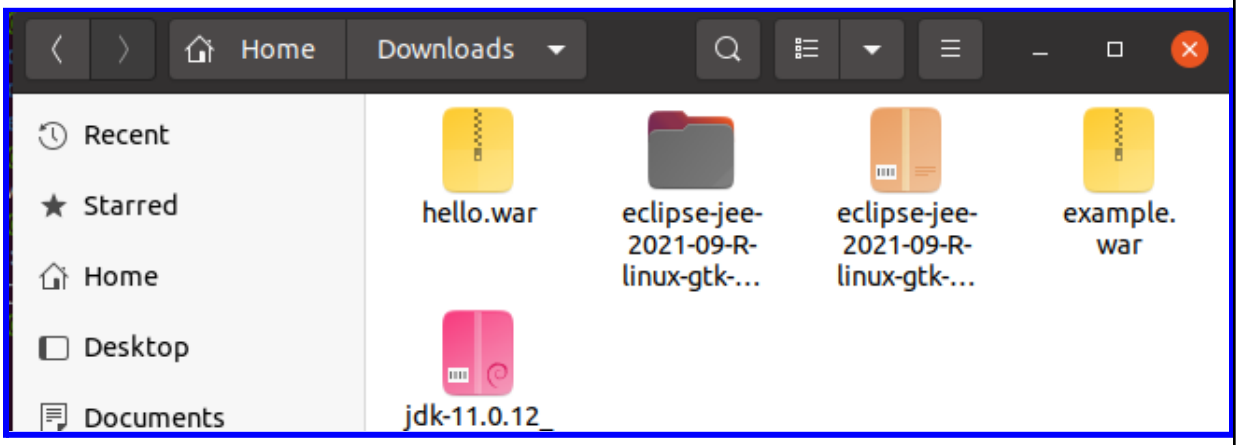
To create servlet install **Java EE - Eclipse** for using **maven project**

Downloaded the eclipse IDE from official page for linux

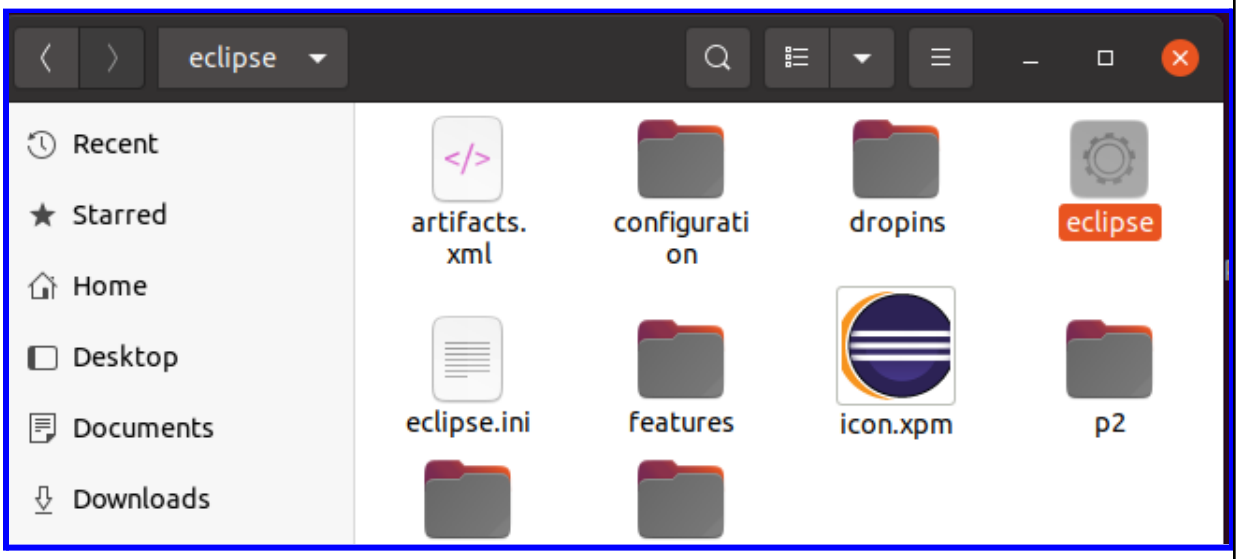


Extracted it and we can directly use it as I have done below

- We can do it either by gui directly or with the command
    - **sudo tar -xzvf <zip-filename>**

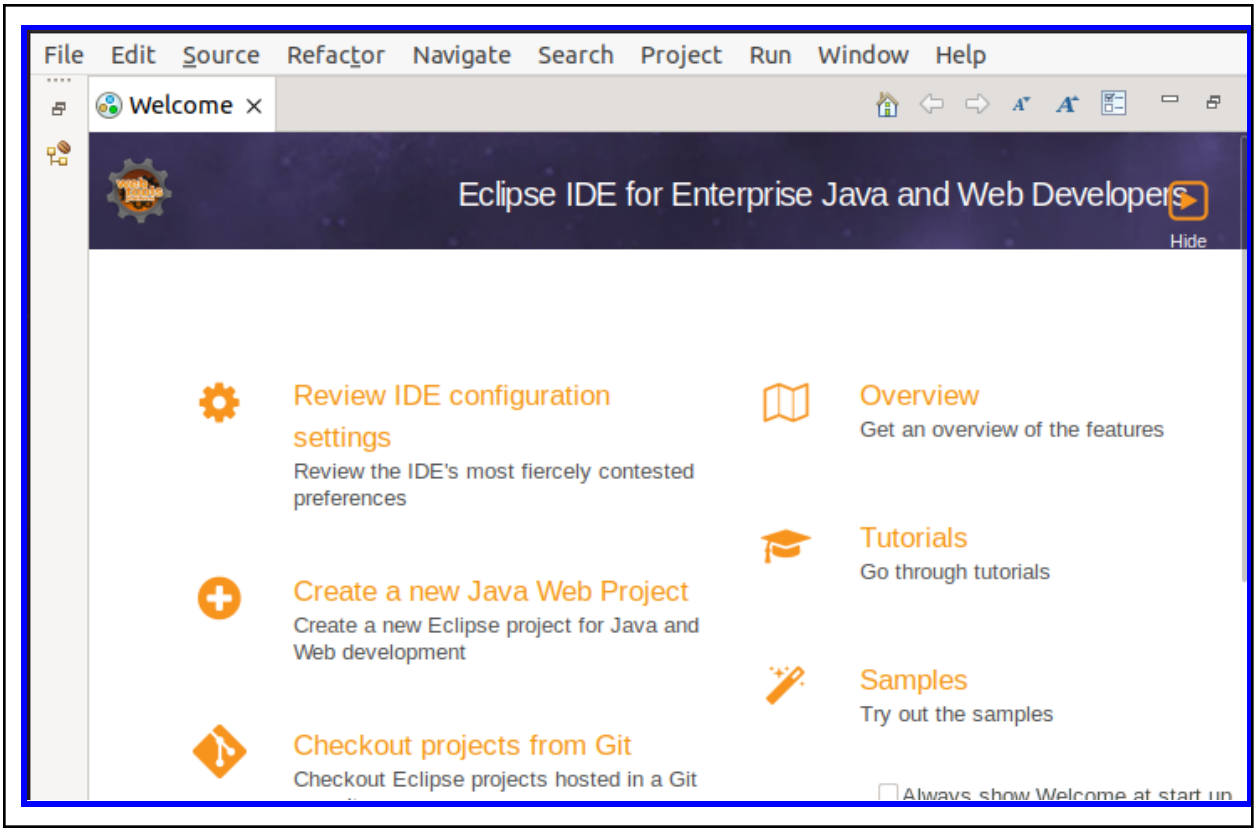


We can go inside this directory >> eclipse and open the eclipse file



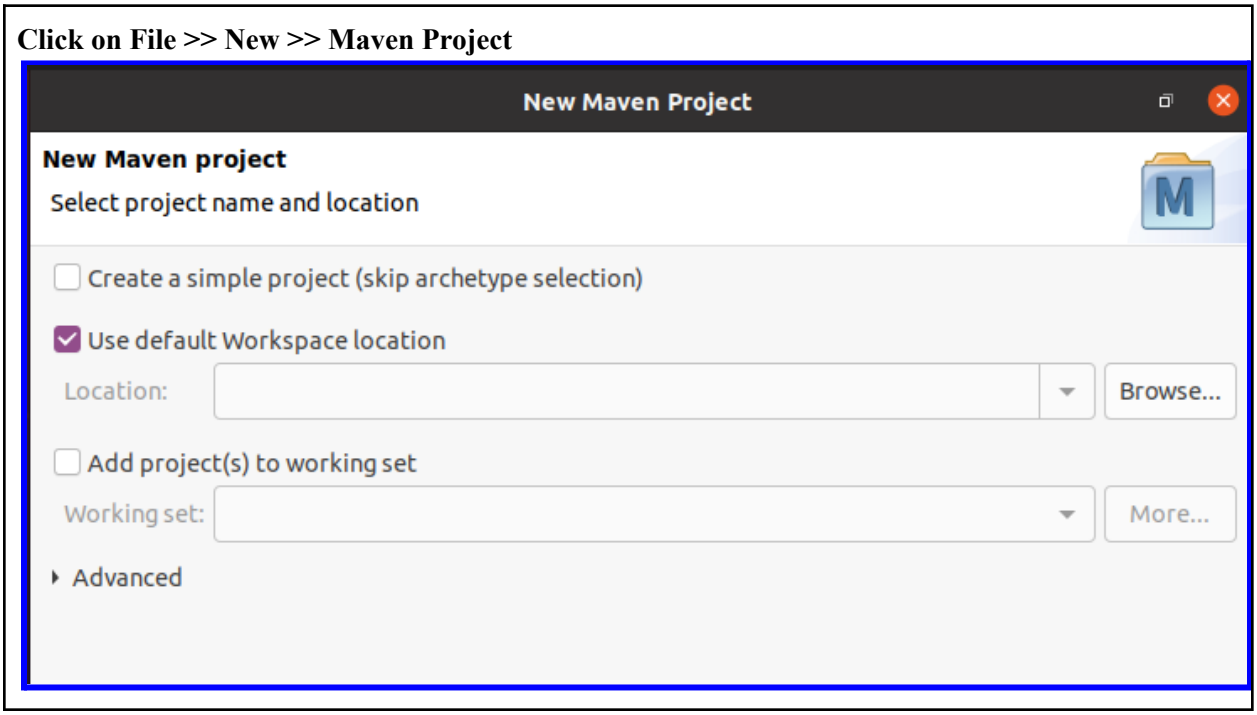
Now eclipse IDE will configure its workspace and we can continue working

At first we can get welcome page like this



Create new maven project

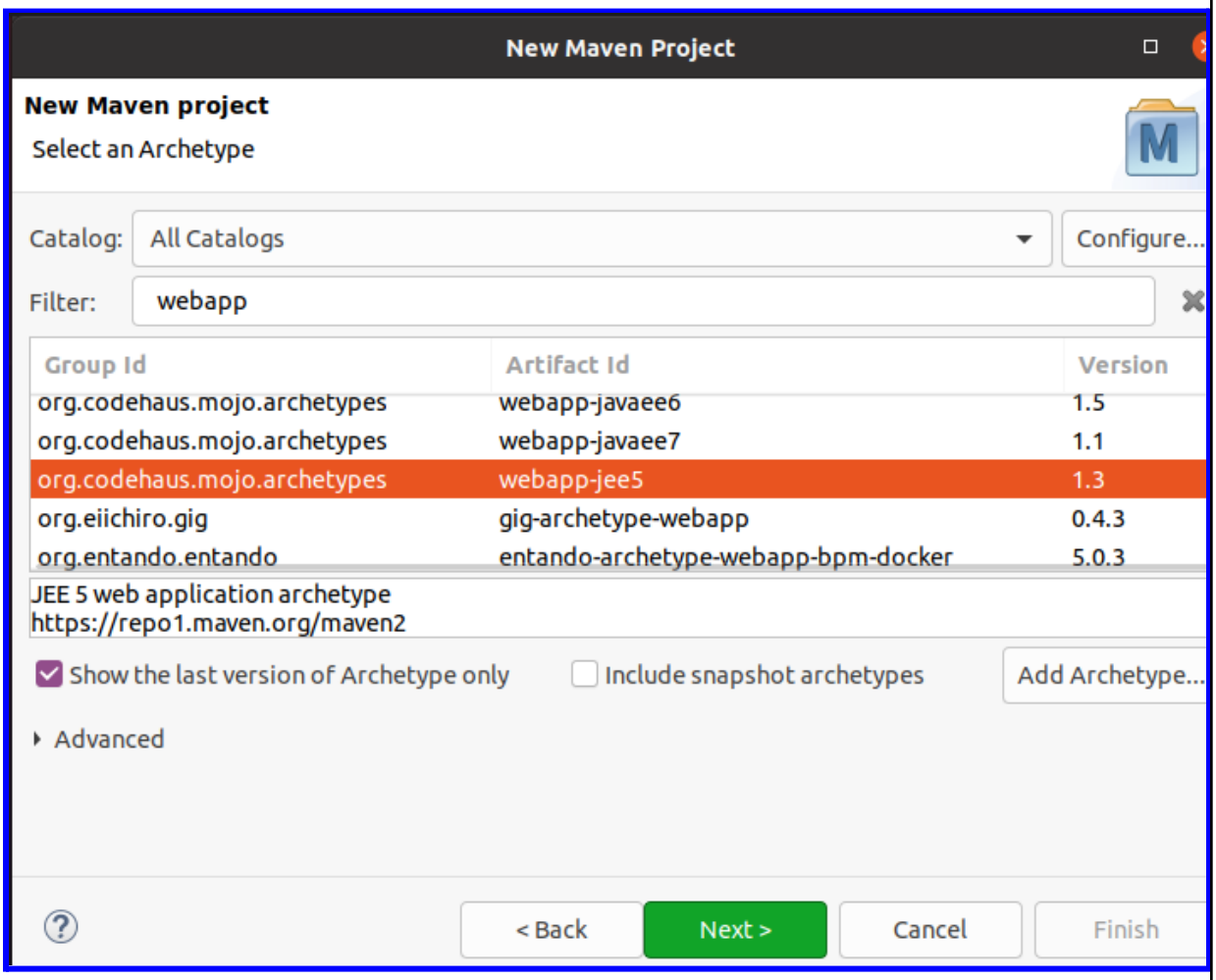
Click on File >> New >> Maven Project



optional -- we can change the default workspace if we want

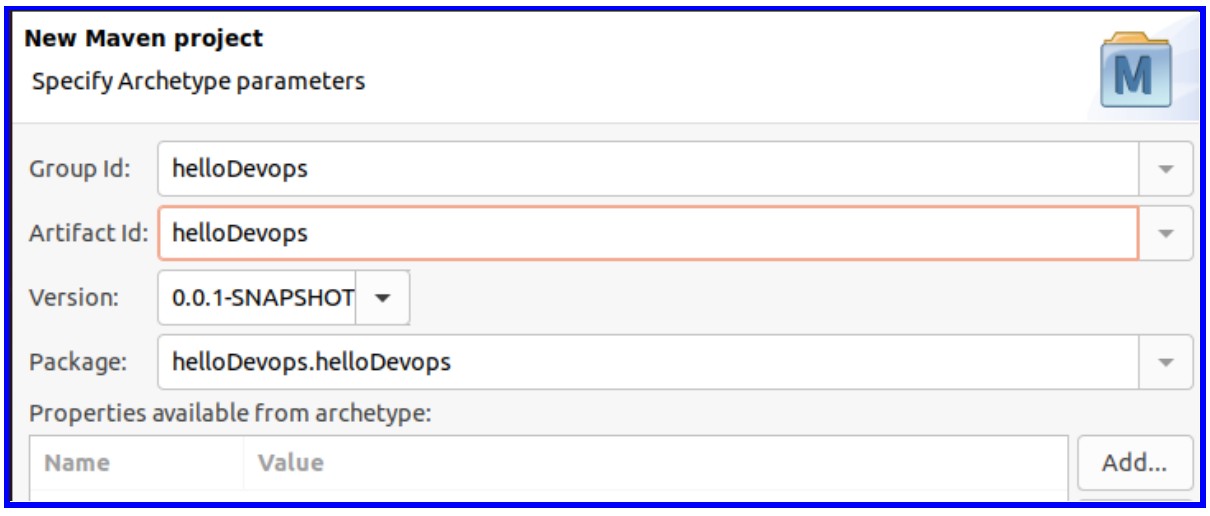
Click on Next (bottom)

Filter for webapp-jee5



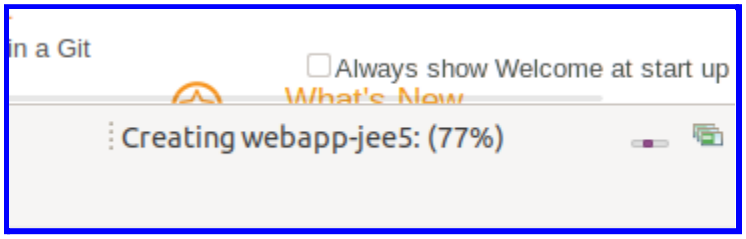
Select it and click next

Give the group Id and Artifact ID



Click on finish

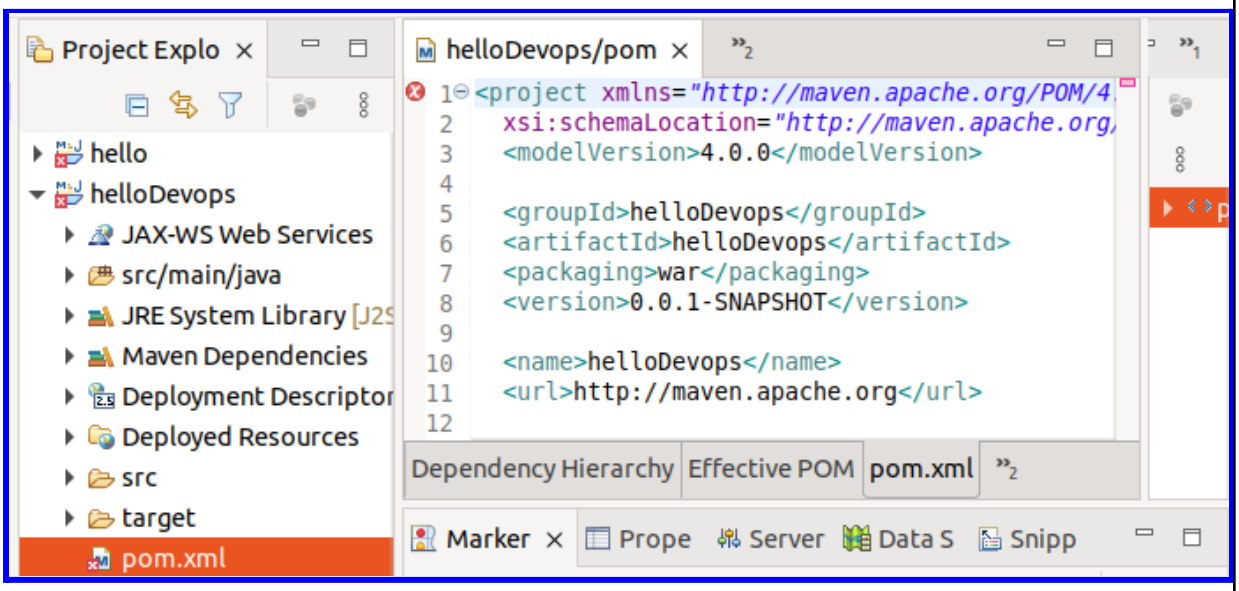
You can see something like this at bottom



When it finishes we can now proceed

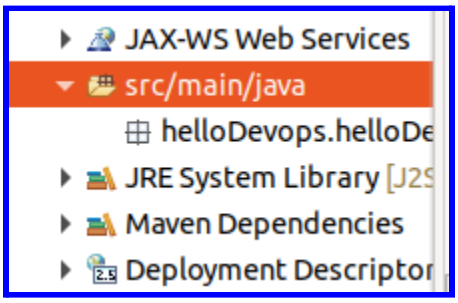
Clicking on pom.xml file

We can view the code

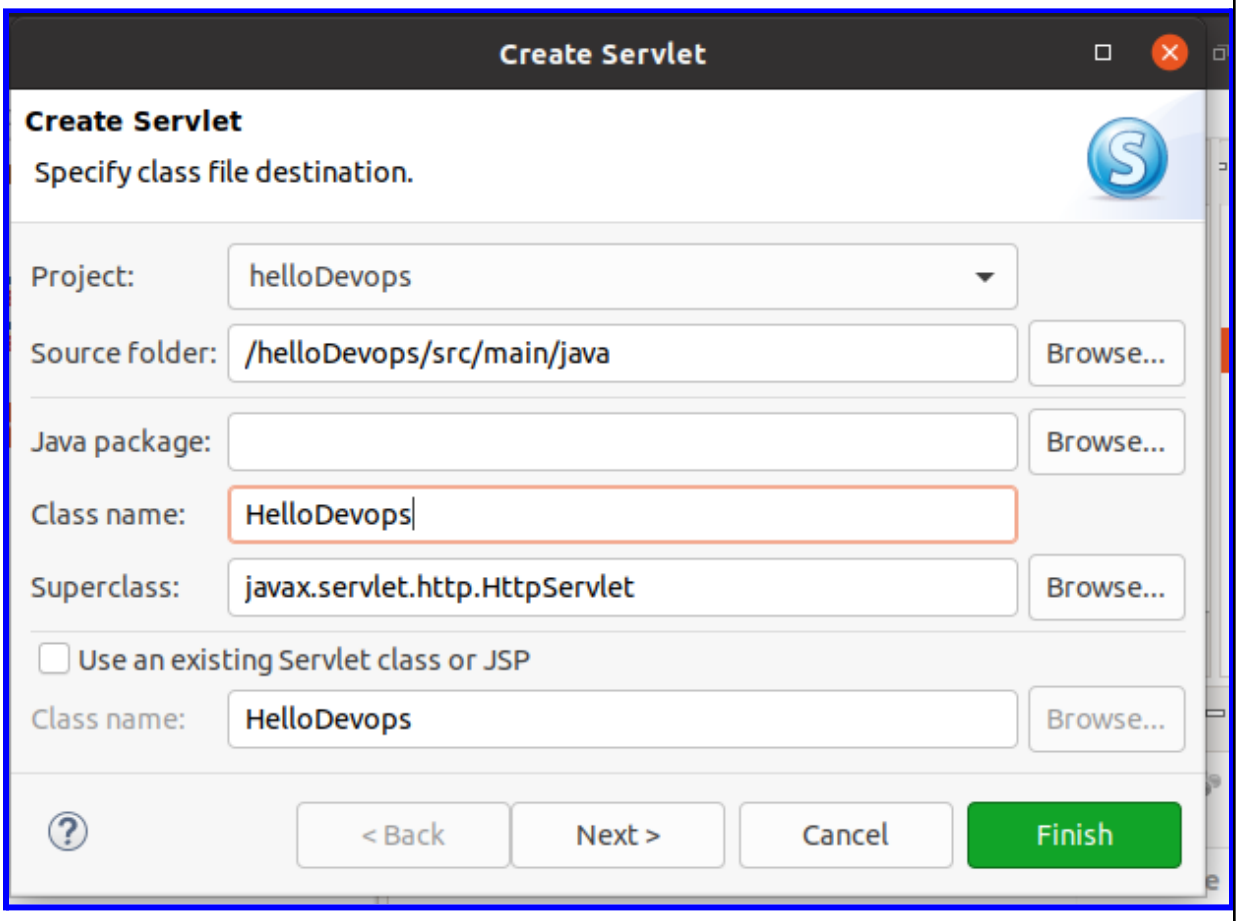


If we scroll down we can see javax.servlet dependencies too

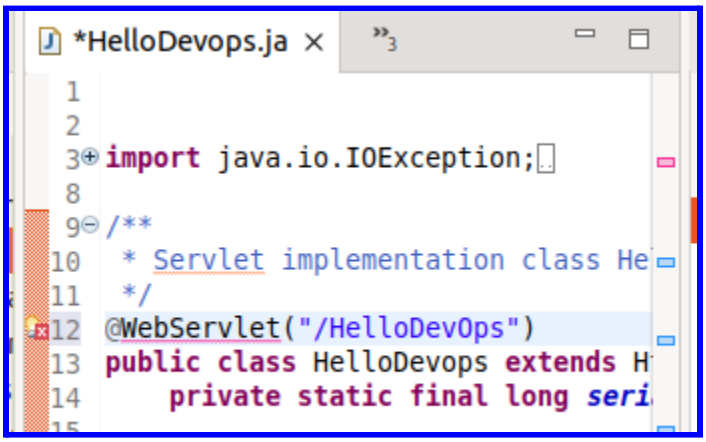
To add servlet class right click on src/main/java directory



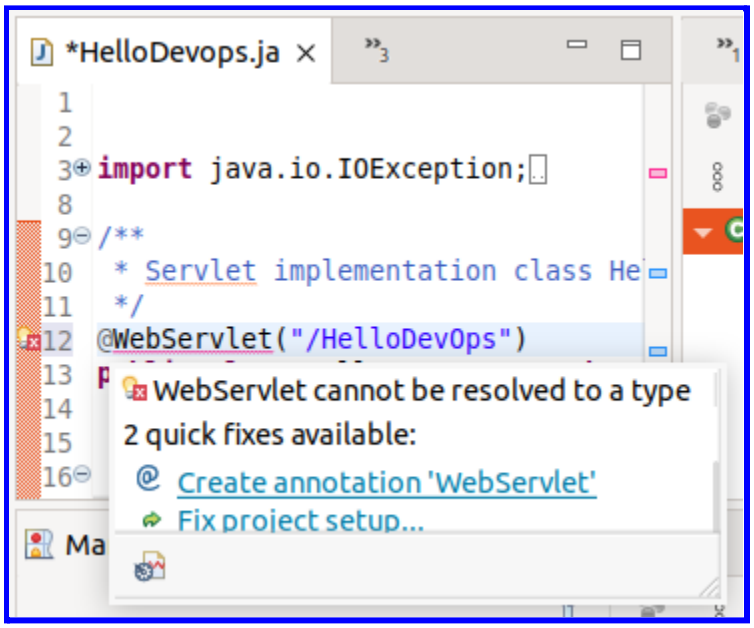
Folow for New >> Servlet and give the class name



In HelloDevops.java add @WebServlet line above public class

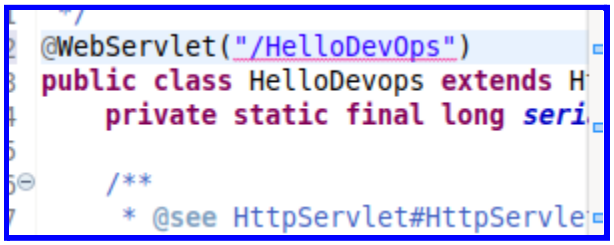


Hover over WebServlet and **create annotation**

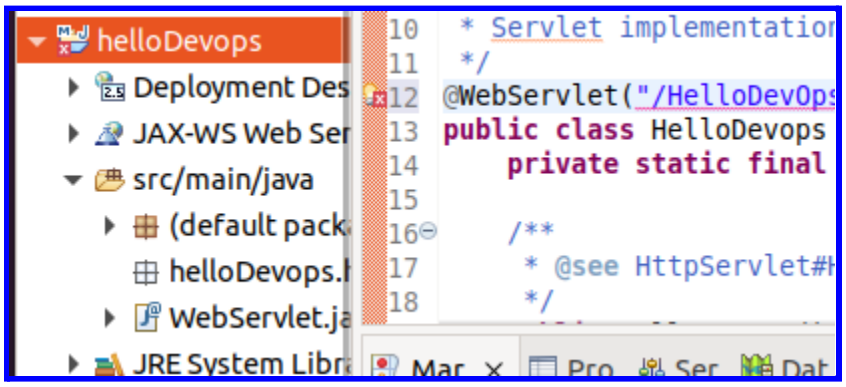


Click on finish when you get another page

Now we can see the error has gone

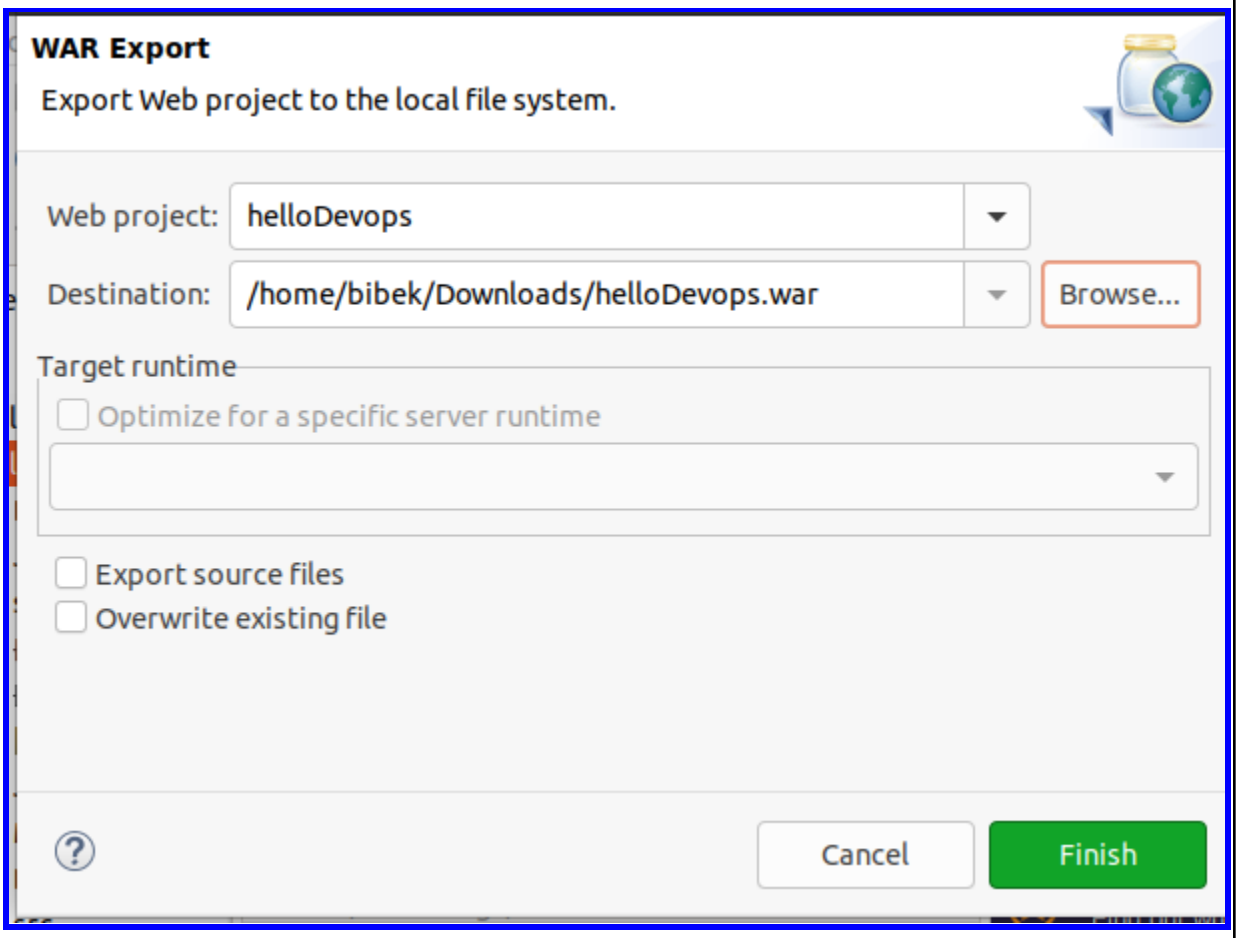


Now we can generate the war file from the main project



Right click on it >> export >> war file

Browse the destination

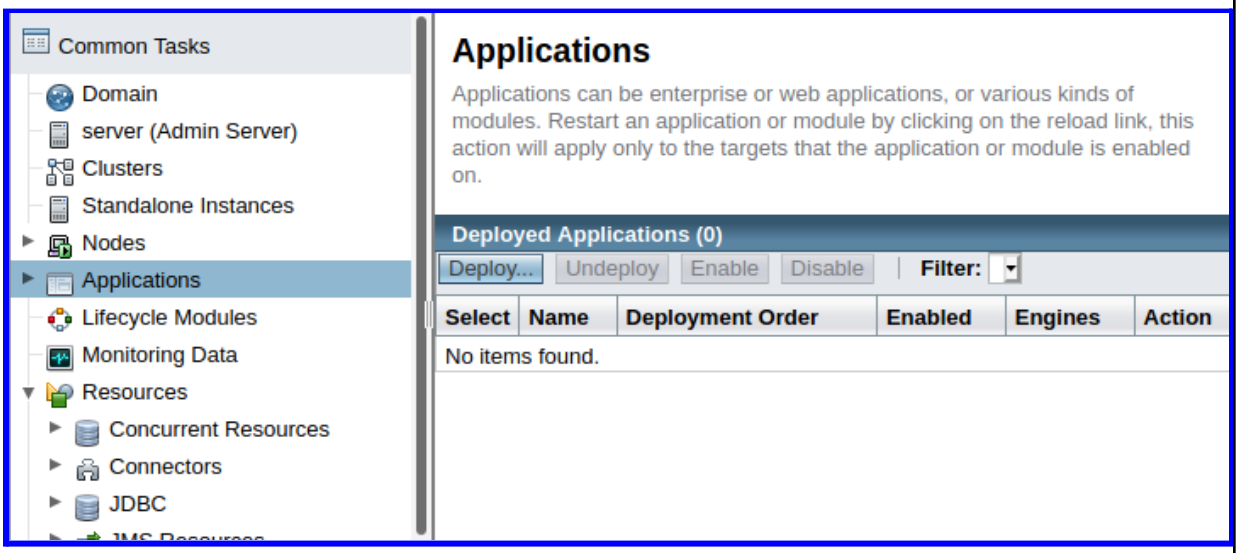


And finish

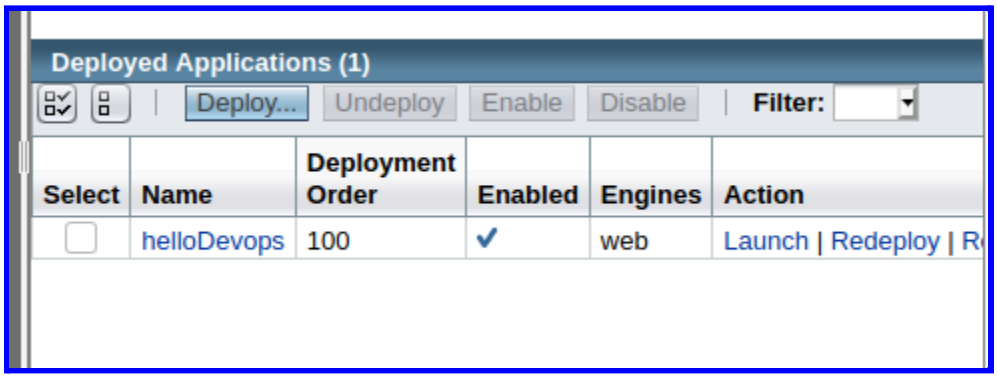
Deploy WAR file to GlassFish Server

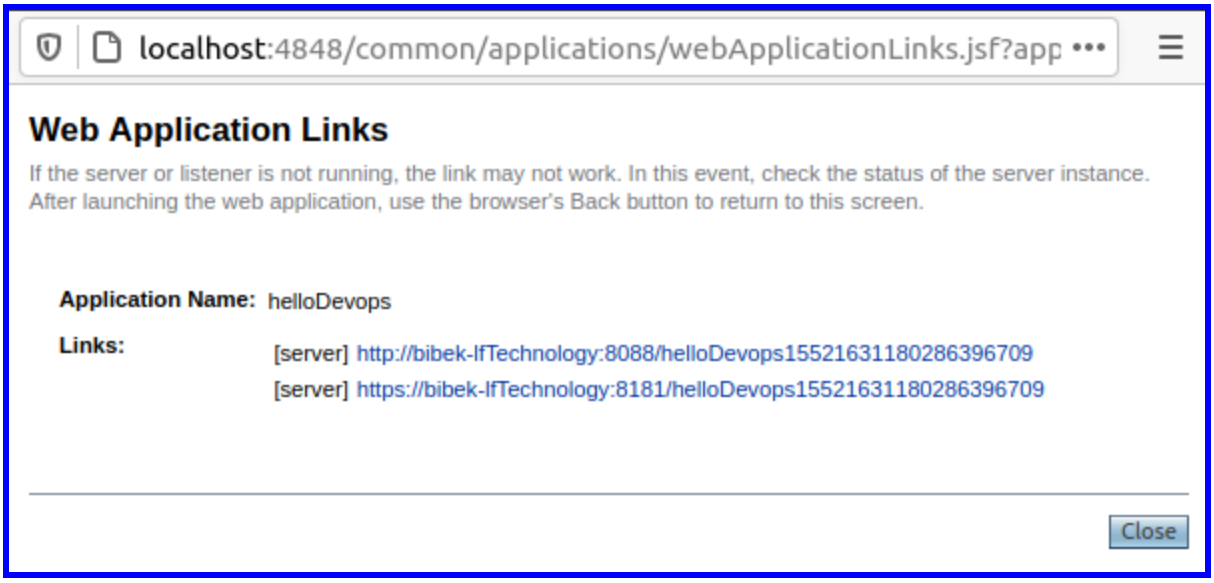
Browse the admin panel of glassfish to deploy the application

Go to applications >> deploy >> packaged filed to be uploaded >> browse the file from the destination >> ok



Then click on launch





Clicking on the first server we can get the deployed web page

In this way a maven servlet application can be deployed to the glassfish

Gunicorn

We need python3-pip, python3-venv and nginx to be installed on our system

First setup a virtual environment with python and activate it

I have made a different directory to go with this django project

mkdir django

cd django

python3 -m venv django1

source django1/bin/activate

Install django and gunicorn

pip install django gunicorn

Start django demo project in the same directory so that we can get manage.py here

django-admin startproject demo ~/django

To migrate the database if needed (optional)

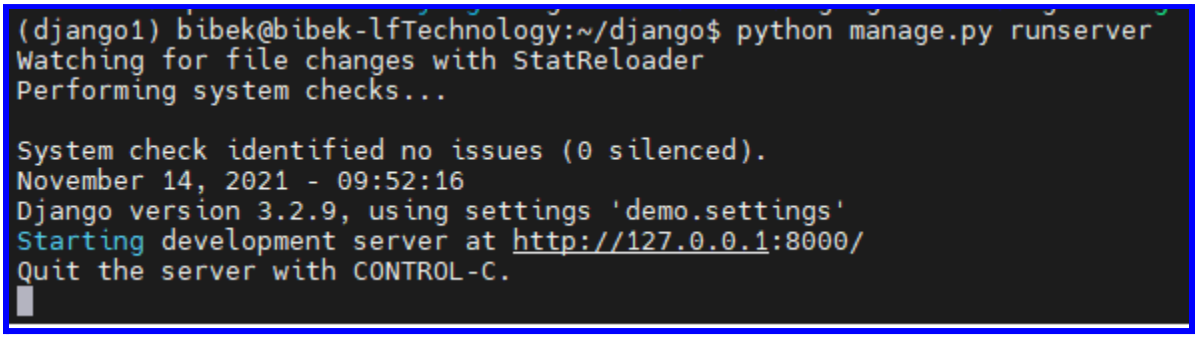
python manage.py migrate

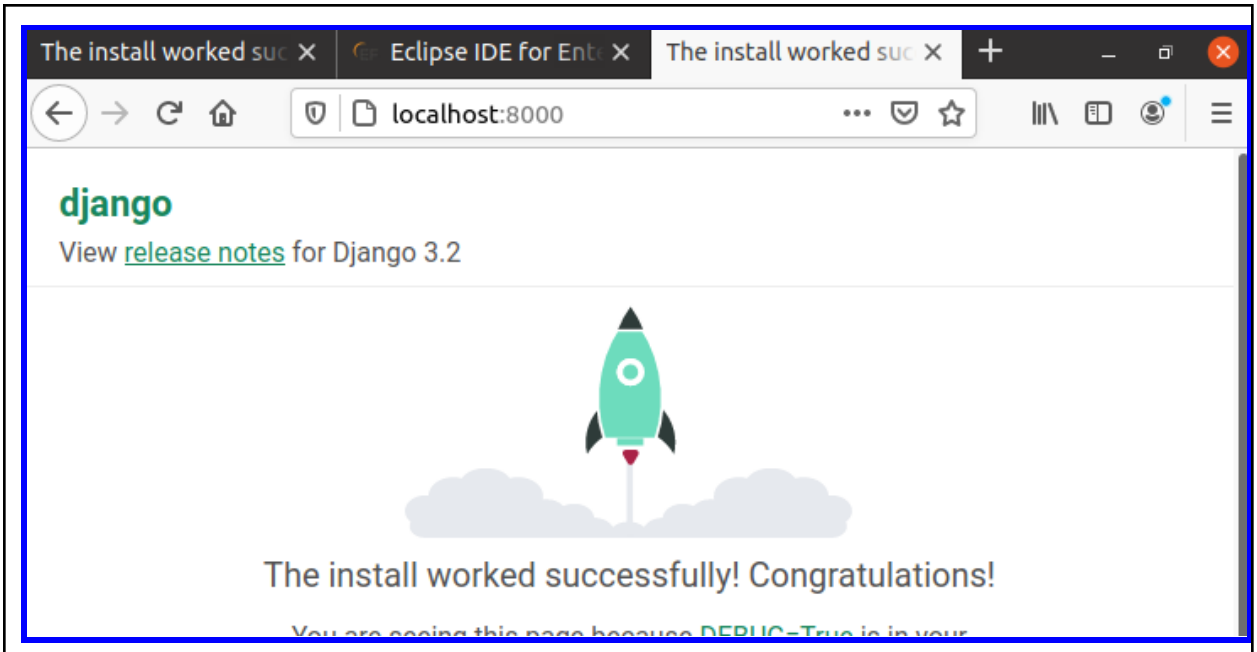
And we can directly test if this is working or not by running

python manage.py runserver

Or we can define the port to run on by

python manage.py runserver 0..0.0.0:8000





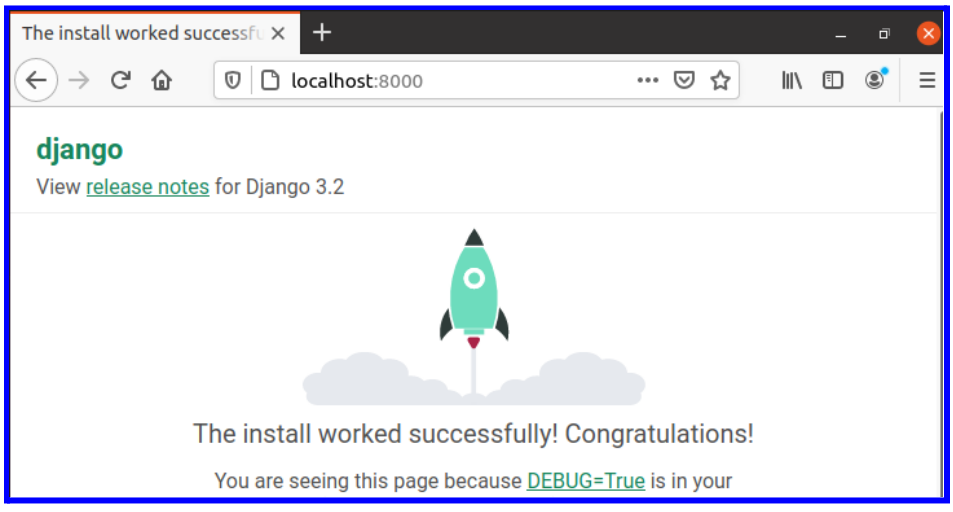
So it confirms that we can proceed now

To run this demo project through gunicorn

gunicorn --bind 0.0.0.0:8000 demo.wsgi

```
^C(django1) bibek@bibek-lfTechnology:~/django$ gunicorn --bind 0.0.0.0:8000 de
.wsgi
[2021-11-14 15:41:45 +0545] [10210] [INFO] Starting gunicorn 20.1.0
[2021-11-14 15:41:45 +0545] [10210] [INFO] Listening at: http://0.0.0.0:8000 (
10210)
[2021-11-14 15:41:45 +0545] [10210] [INFO] Using worker: sync
[2021-11-14 15:41:45 +0545] [10212] [INFO] Booting worker with pid: 10212
```

And again we can view the starter page is running



Now deactivate the environment and setup the reverse proxy through nginx

- **To secure our application**
- **And we don't need to execute the app server command gunicorn to host our app**

Make gunicorn.socket file inside system directory

sudo vim /etc/systemd/system/gunicorn.socket

And paste the following

```
[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock

[Install]
WantedBy=sockets.target
~
```

Save and exit

Make gunicorn.service file and set the configurations

sudo vim /etc/systemd/system/gunicorn.service

And set the configuration according to your system

```
[Unit]
Description=gunicorn daemon
Requires=gunicorn.socket
After=network.target

[Service]
User=bibek
Group=www-data
WorkingDirectory=/home/bibek/django
ExecStart=/home/bibek/django/django1/bin/gunicorn \
          --access-logfile gunicorn.log \
          --error-logfile gunicorn.error.log \
          --workers 3 \
          --bind unix:/run/gunicorn.sock \
          demo.wsgi:application

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

Variables to set

- **User**: your user name

- **Working Directory**: the directory where we initialize the virtual environment and installed all the requirements of gunicorn
- **--access-log file**: to get the log of gunicorn
- **--error-logfile**: to get the error message of gunicorn
- **--workers** to set the number of worker process to serve the requests
- **--bind**: to bind the unix socket of gunicorn

Save and exit

Start the gunicorn service and enable the service

sudo systemctl start gunicorn.socket

sudo systemctl enable gunicorn.socket

Now configure the nginx to serve the gunicorn appserver

Make a file in sites-available of nginx

sudo vim /etc/nginx/sites-available/demo

I have given the same name as of starterproject of django

And and the configuration as below

```
server {
    listen 80;
    server_name localhost;

    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root /home/bibek/django;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/run/gunicorn.sock;
    }
}
```

Adjust the parameter

- **Server name**: as per our host
- **Root directory**: where your project is located
- **proxy pass**: to map to the gunicorn.sock

Save And exit

Create soft link pf demo from sites-available to sites-enable of nginx to enable demo application

sudo ln -s /etc/nginx/sites-available/demo /etc/nginx/sites-enabled/

You can delete the default page rested in site-enabled directory if default page comes browsing localhost to disable default page

nginx -t

To check errors in the configuration file of nginx.

If it gives ok

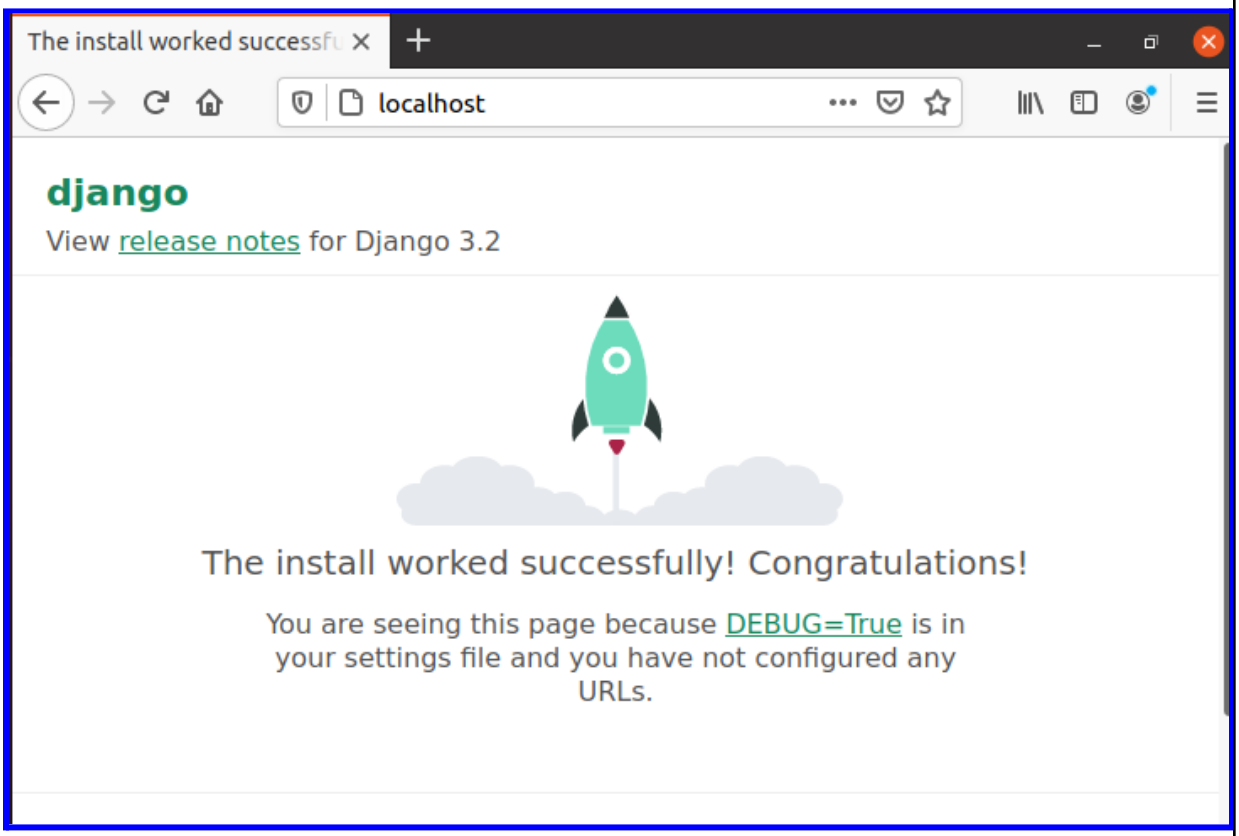
```
bibek@bibek-lfTechnology:~/django$ sudo !!
sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
bibek@bibek-lfTechnology:~/django$
```

Then we can restart the nginx service

sudo systemctl restart nginx

As nginx is serving on port 80

We can directly browse the starter web page on localhost



We can see the log file and error file of gunicorn in main directory of virtual environment

```
bibek@bibek-lfTechnology:~/django$ ls
conf  db.sqlite3  demo  django1  gunicorn.error.log  gunicorn.log  manage.py
bibek@bibek-lfTechnology:~/django$
```

And if we check the status of gunicorn we can see the three worker processes are running

```
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
```

To change the default port of gunicorn to 8089

Make a conf directory where our demo project is created

mkdir conf

Make a file gunicorn_config.py inside conf directory to set the port and other variables

vim conf/gunicorn_config.py

```
command ='/home/bibek/django/django1/bin/gunicorn'
pythonpath = '/home/bibek/django/demo'
bind = 'localhost:8089'
workers = 3
~
```

Parameters to be adjusted

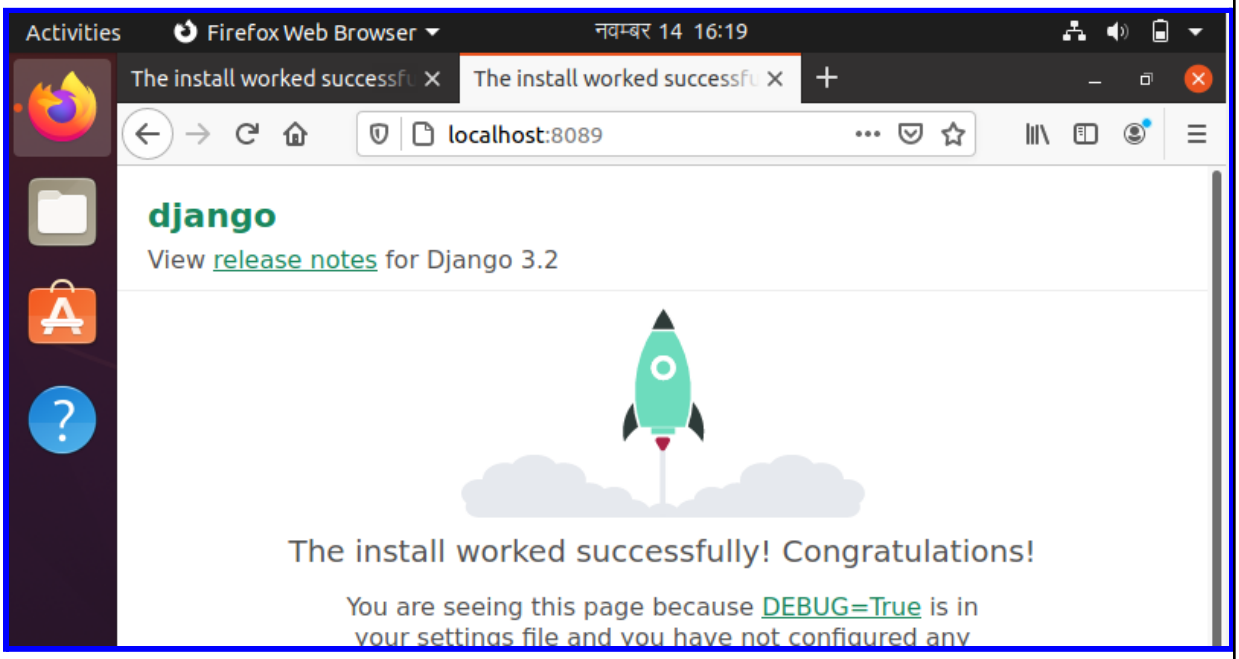
- **command**: the path to execute the gunicorn inside virtual env
- **pythonpath**: path to the starter page inside virtual env
- **bind**: ‘server:port’
- **workers** : number of worker node process

Activate the virtual environment and run gunicorn giving the path of conf file to serve

gunicorn -c conf/gunicorn_config.py demo.wsgi

```
bibek@bibek-lfTechnology:~/django$ ls
conf  db.sqlite3  demo  django1  gunicorn.error.log  gunicorn.log  manage.py
bibek@bibek-lfTechnology:~/django$ source django1/bin/activate
(django1) bibek@bibek-lfTechnology:~/django$ gunicorn -c conf/gunicorn_config.
py demo.wsgi
[2021-11-14 16:16:25 +0545] [10507] [INFO] Starting gunicorn 20.1.0
[2021-11-14 16:16:25 +0545] [10507] [INFO] Listening at: http://127.0.0.1:8089
 (10507)
[2021-11-14 16:16:25 +0545] [10507] [INFO] Using worker: sync
[2021-11-14 16:16:25 +0545] [10509] [INFO] Booting worker with pid: 10509
[2021-11-14 16:16:25 +0545] [10510] [INFO] Booting worker with pid: 10510
[2021-11-14 16:16:25 +0545] [10511] [INFO] Booting worker with pid: 10511
Not Found: /static/admin/css/fonts.css
^C[2021-11-14 16:18:44 +0545] [10507] [INFO] Handling signal: int
[2021-11-14 10:33:44 +0000] [10509] [INFO] Worker exiting (pid: 10509)
[2021-11-14 10:33:44 +0000] [10511] [INFO] Worker exiting (pid: 10511)
[2021-11-14 10:33:44 +0000] [10510] [INFO] Worker exiting (pid: 10510)
[2021-11-14 16:18:44 +0545] [10507] [INFO] Shutting down: Master
(django1) bibek@bibek-lfTechnology:~/django$
```

Now if we browse the localhost at 8089 port we can see the starter page



In this way we can set the default port for gunicorn to serve he application