

## Glassfish:

- Install Glassfish server and change HTTP port to 8088.
  - GlassFish Server Open Source Edition is an open-source application server built within the GlassFish community. Oracle GlassFish Server is based on GlassFish Server Open Source Edition. GlassFish Server users benefit from a vibrant community that offers self-support, contributes code and product features, product ideas and feedback, bug reports, and more.

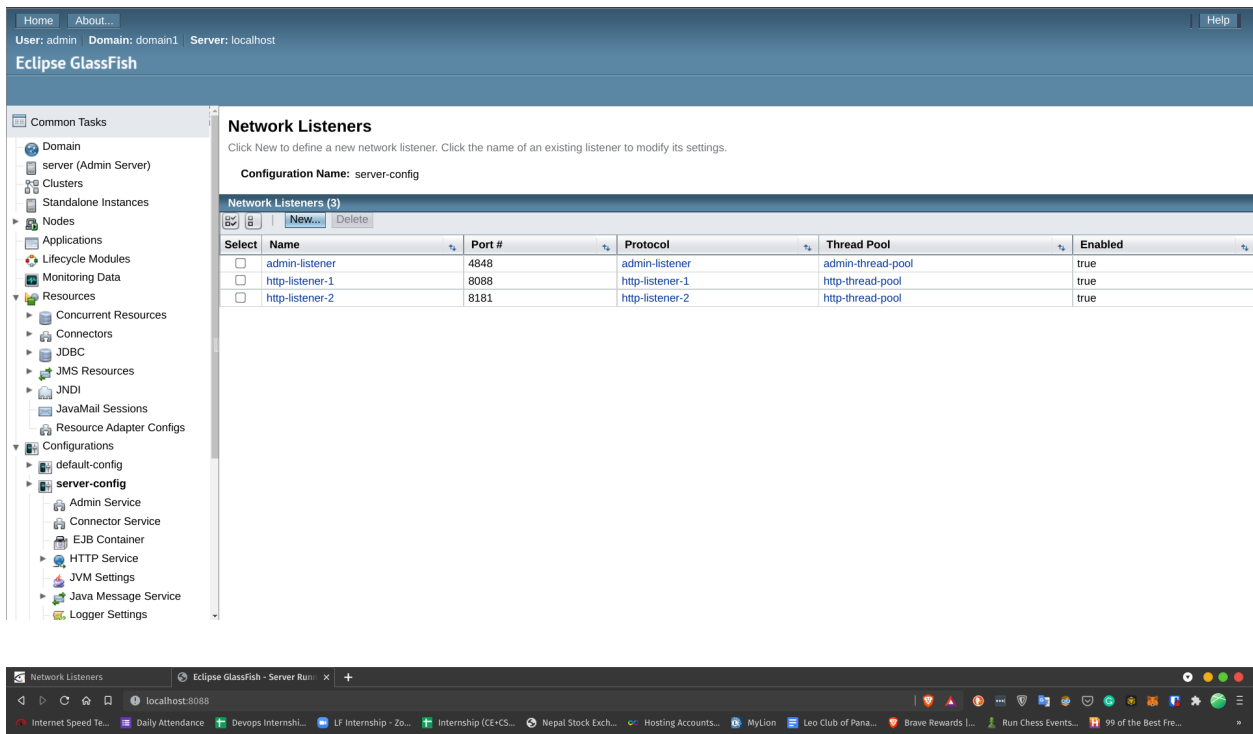
### Install GlassFish

- Install JDK at first using the command:
  - `sudo apt install openjdk-11-hre-headless`
- Download the zip file for GlassFish 6.2.2 from GitHub link using the command:
  - `wget https://github.com/eclipse-ee4j/glassfish/releases/download/6.2.2/glassfish-6.2.2.zip`
- Once it's downloaded, extract using the following command:
  - `unzip glassfish-6.2.2.zip`
- Now we get the directory called glassfish6 inside glassfish-6.2.2.
- Then we can navigate to glassfish4/bin and run the following command to start the server:
- `./asadmin start-domain`
- Our server has now been started.

### Change HTTP port to 8088

- Run the GlassFish server and connect to the administrative interface using the default port (4848)
- In the left menu go to **Configurations**
- Then select **server-config**
- Now go to **Network Listeners**
- Select **http-listener-1** and change its port value to **8088**

- Click save and restart the server



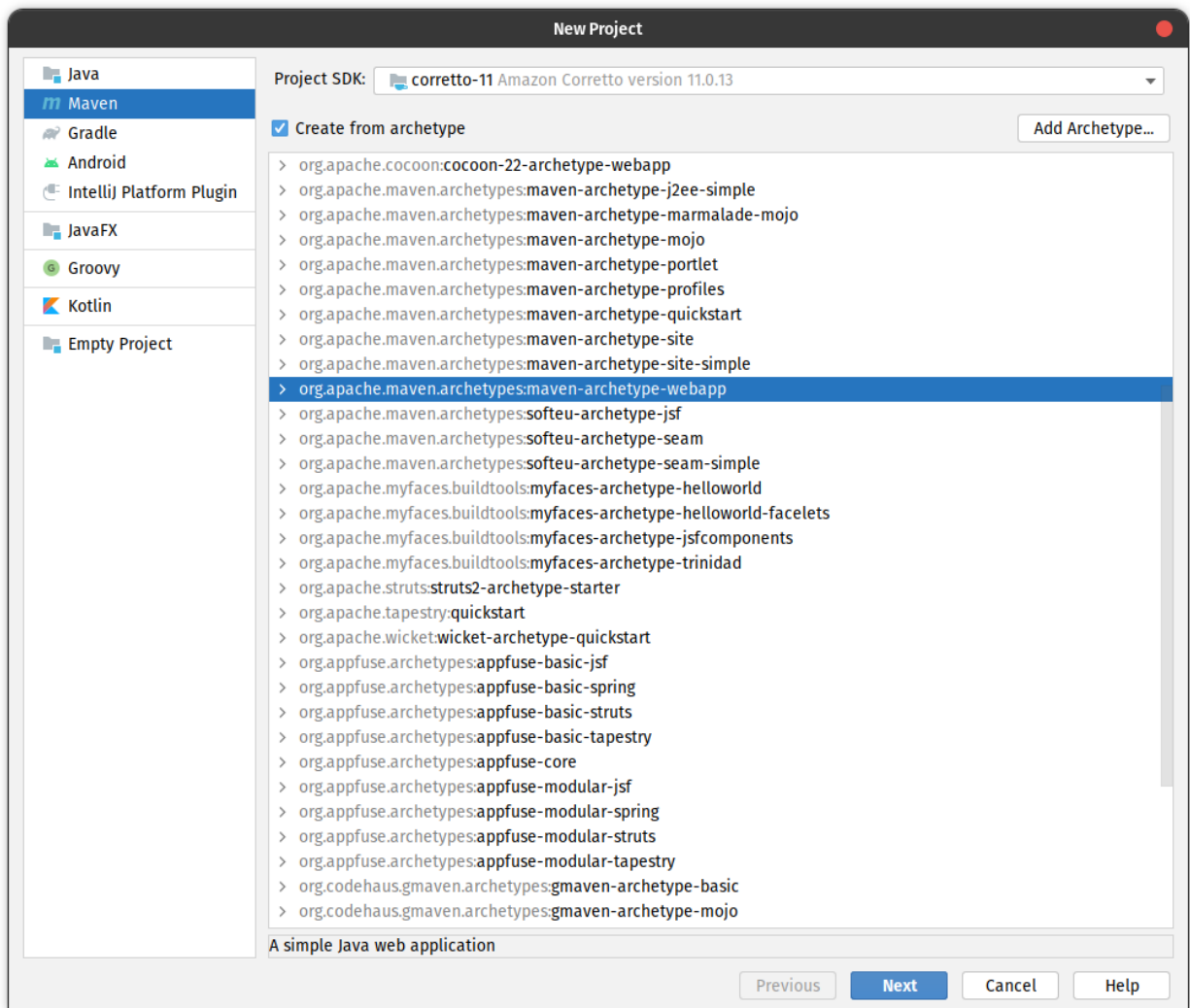
- Create a demo Java (11) servlet application with maven.

- A Servlet is a java class that is extended to handle the capabilities of a server. Servlets can be used to handle the requests and responses of a server.

- Apache Maven is a software project management and comprehension tool. Maven can help you manage a project's build, reporting, and documentation from a central piece of information.

## Demo Java (11) servlet application with maven

- Install IntelliJ IDEA and Create a Maven Project



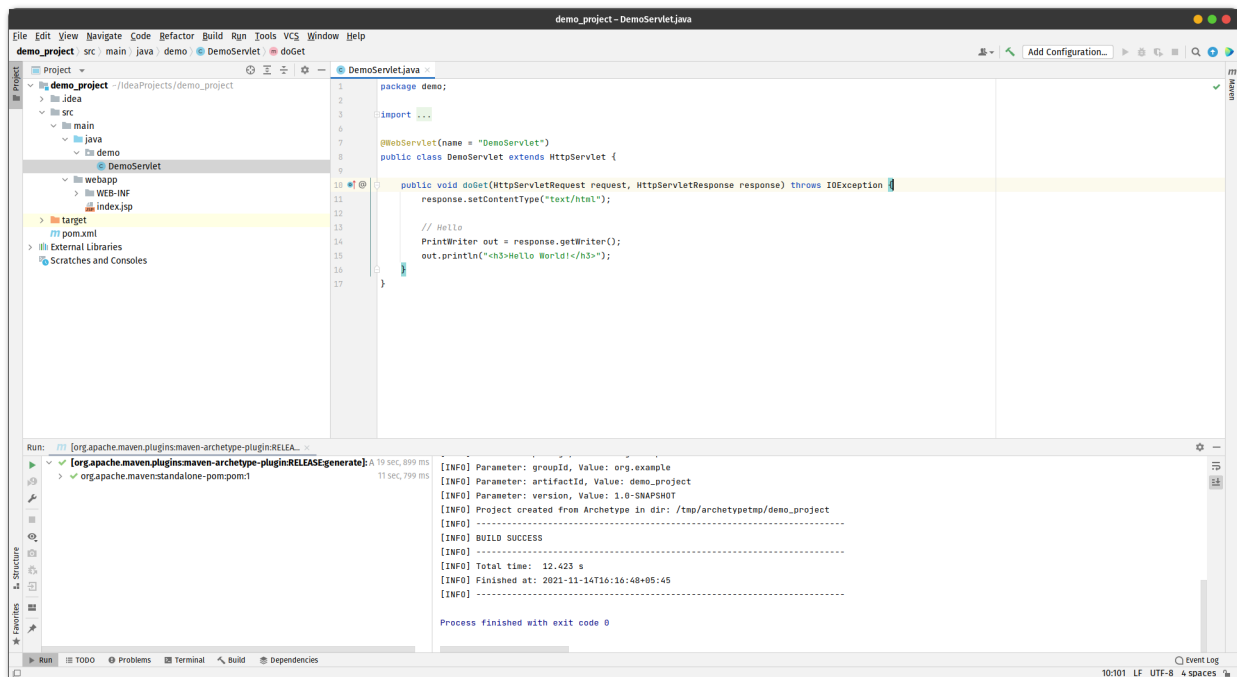
- Using pom.xml to manage the project dependencies
  - POM stands for "Project Object Model". It is an XML representation of a Maven project held in a file named pom.xml.
    - Here we need to add the servlet dependencies in order to work with it.

```

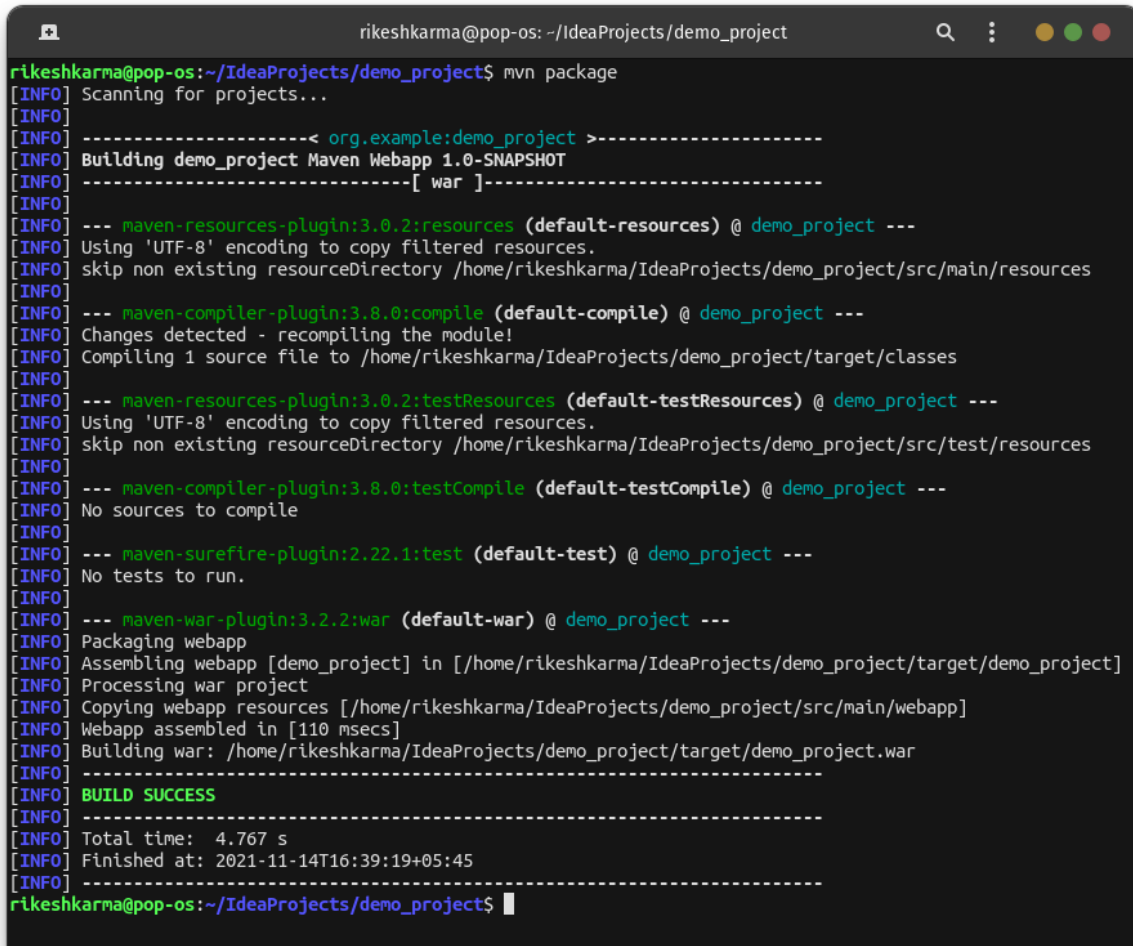
21
22 <dependencies>
23 <dependency>
24 <groupId>junit</groupId>
25 <artifactId>junit</artifactId>
26 <version>4.11</version>
27 <scope>test</scope>
28 </dependency>
29 <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
30 <dependency>
31 <groupId>javax.servlet</groupId>
32 <artifactId>javax.servlet-api</artifactId>
33 <version>4.0.1</version>
34 <scope>provided</scope>
35 </dependency>
36 </dependencies>
37

```

- Create a new folder named *java* inside *src/main* folder, then set the *java* folder to sources root so as to show 'Class' on the IntelliJ IDEA when we right-click and select 'New'.
- Create a new class inside the *java* folder named *DemoServlet.java* inside the *demo* folder.



- Generate war package.
  - To generate a war package we need to run the following command inside the project directory:
    - *maven package*
  - The demo\_project.war file is generated inside ~/demo\_projects/target.



```

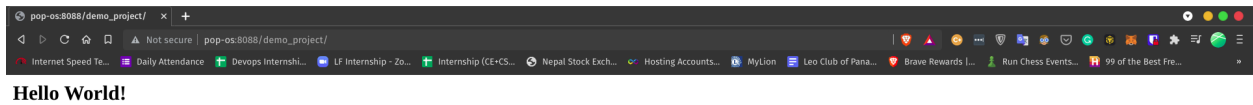
rireshkarma@pop-os: ~/IdeaProjects/demo_project
rireshkarma@pop-os:~/IdeaProjects/demo_project$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.example:demo_project >-----
[INFO] Building demo_project Maven Webapp 1.0-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ demo_project ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/rireshkarma/IdeaProjects/demo_project/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ demo_project ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/rireshkarma/IdeaProjects/demo_project/target/classes
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ demo_project ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/rireshkarma/IdeaProjects/demo_project/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ demo_project ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ demo_project ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ demo_project ---
[INFO] Packaging webapp
[INFO] Assembling webapp [demo_project] in [/home/rireshkarma/IdeaProjects/demo_project/target/demo_project]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/rireshkarma/IdeaProjects/demo_project/src/main/webapp]
[INFO] Webapp assembled in [110 msecs]
[INFO] Building war: /home/rireshkarma/IdeaProjects/demo_project/target/demo_project.war
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.767 s
[INFO] Finished at: 2021-11-14T16:39:19+05:45
[INFO]
rireshkarma@pop-os:~/IdeaProjects/demo_project$

```

- Deploy the war using the glassfish app server.
  - To deploy we need to start the GlassFish server first by navigating inside glassfish4/bin like before and run the following command to start the server:
    - *./asadmin start-domain*
    - Then we need to run the following command to deploy the war file we just generated:

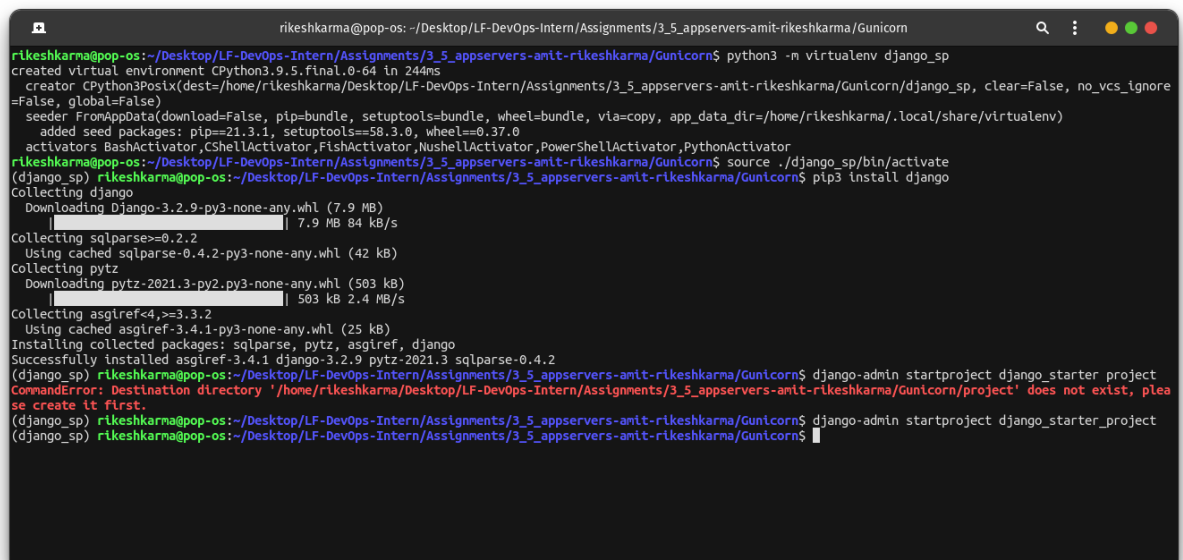
- `./asadmin deploy`  
`/home/rikeshkarma/IdeaProjects/demo_project/target/demo_project.war`

```
rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Pract/glassfish6/bin
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Pract/glassfish6/bin$ ./asadmin start-domain
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: /home/rikeshkarma/Desktop/LF-DevOps-Intern/Pract/glassfish6/glassfish/domains/domain1
Log File: /home/rikeshkarma/Desktop/LF-DevOps-Intern/Pract/glassfish6/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Pract/glassfish6/bin$ ./asadmin deploy /home/rikeshkarma/IdeaProjects/demo_project/target/demo_project.war
Application deployed with name demo_project.
Command deploy executed successfully.
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Pract/glassfish6/bin$
```



# Gunicorn

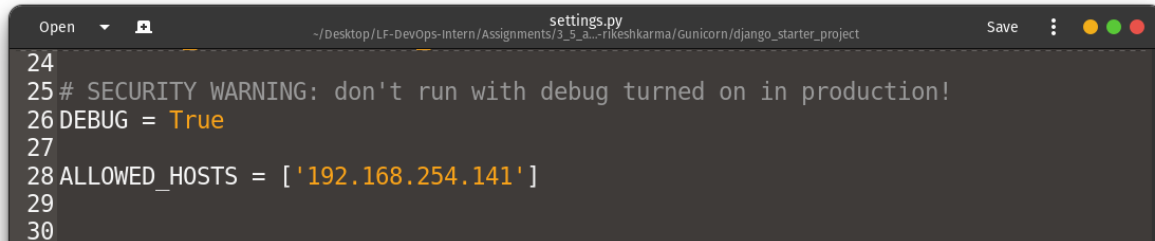
- Create Django starter project in a separate virtual environment.
  - Firstly we install virtualenv using the following commands:
    - `pip3 install virtualenv`
  - Then create a virtual environment using the command:
    - `python3 -m virtualenv django_sp_env`
  - Activate the virtual environment using the command:
    - `source ./django_sp_env/bin/activate`
  - Now we install Django in the virtual environment:
    - `pip3 install django`
  - Lastly, we create a Django Project, let's call our project **django\_starter\_project** using the command:
    - `django-admin startproject django_starter_project`



```
rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ python3 -m virtualenv django_sp
created virtual environment CPython3.9.5.final.0-64 in 244ms
creator CPython3Posix(dest=/home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn/django_sp, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/rikeshkarma/.local/share/virtualenv)
added seed packages: pip=21.3.1, setuptools=58.3.0, wheel=0.37.0
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ source ./django_sp/bin/activate
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ pip3 install django
Collecting django
  Downloading Django-3.2.9-py3-none-any.whl (7.9 MB)
    | 7.9 MB 84 kB/s
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Collecting pytz
  Downloading pytz-2021.3-py2.py3-none-any.whl (503 kB)
    | 503 kB 2.4 MB/s
Collecting asgiref<4,>=3.3.2
  Using cached asgiref-3.4.1-py3-none-any.whl (25 kB)
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.4.1 django-3.2.9 pytz-2021.3 sqlparse-0.4.2
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ django-admin startproject django_starter_project
CommandError: Destination directory '/home/rikeshkarma/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn/project' does not exist, please create it first.
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ django-admin startproject django_starter_project
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$
```

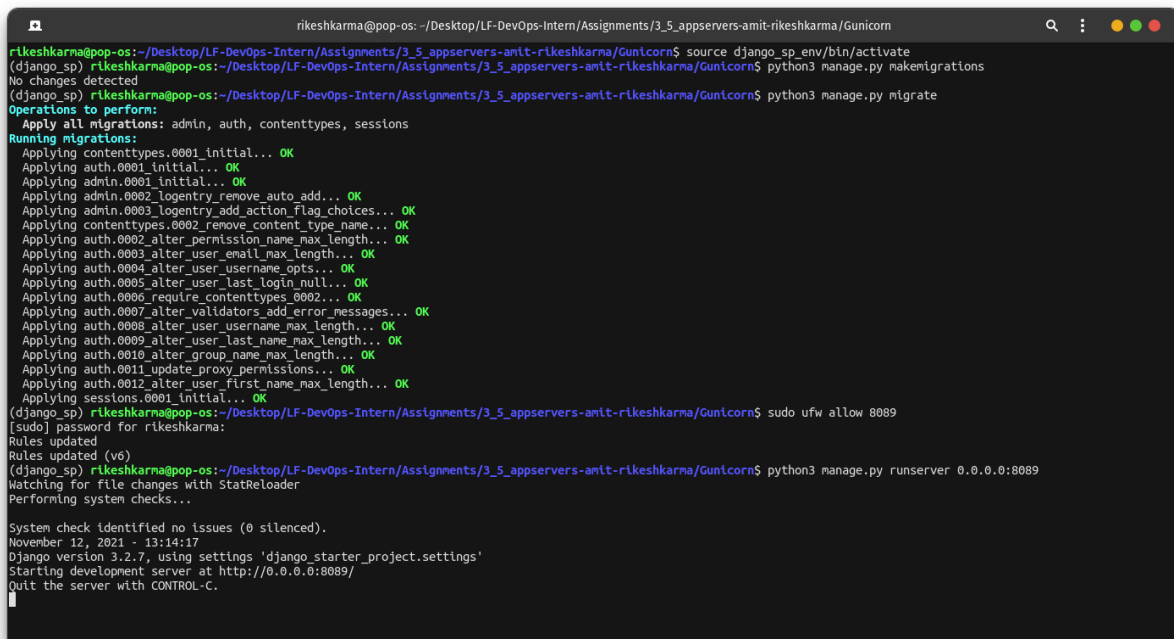
We have created a Django starter project in a virtual environment.

- Deploy the 3 instances of the application using Gunicorn in 8089 port.
  - Activate the virtual environment and install gunicorn:
    - `pip3 install gunicorn`
  - Let's add our IP address to the ALLOWED\_HOSTS variable in path *django\_starter\_project/settings.py*



```
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = ['192.168.254.141']
29
30
```

- Now we run migrations:
  - python3 manage.py makemigrations
  - python3 manage.py migrate`
- Then let's test the sample project by running the following command:
  - sudo ufw allow 8089
  - This opens port:8089 by allowing it over the firewall. Let's check our Django server to test the setup so far using the command:
    - python3 manage.py runserver 0.0.0.0:8089



```
rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ source django_sp_env/bin/activate
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ python3 manage.py makemigrations
No changes detected
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ sudo ufw allow 8089
[sudo] password for rikeshkarma:
Rules updated
Rules reloaded
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ python3 manage.py runserver 0.0.0.0:8089
Watching for file changes with StatReloader
Performing system checks...

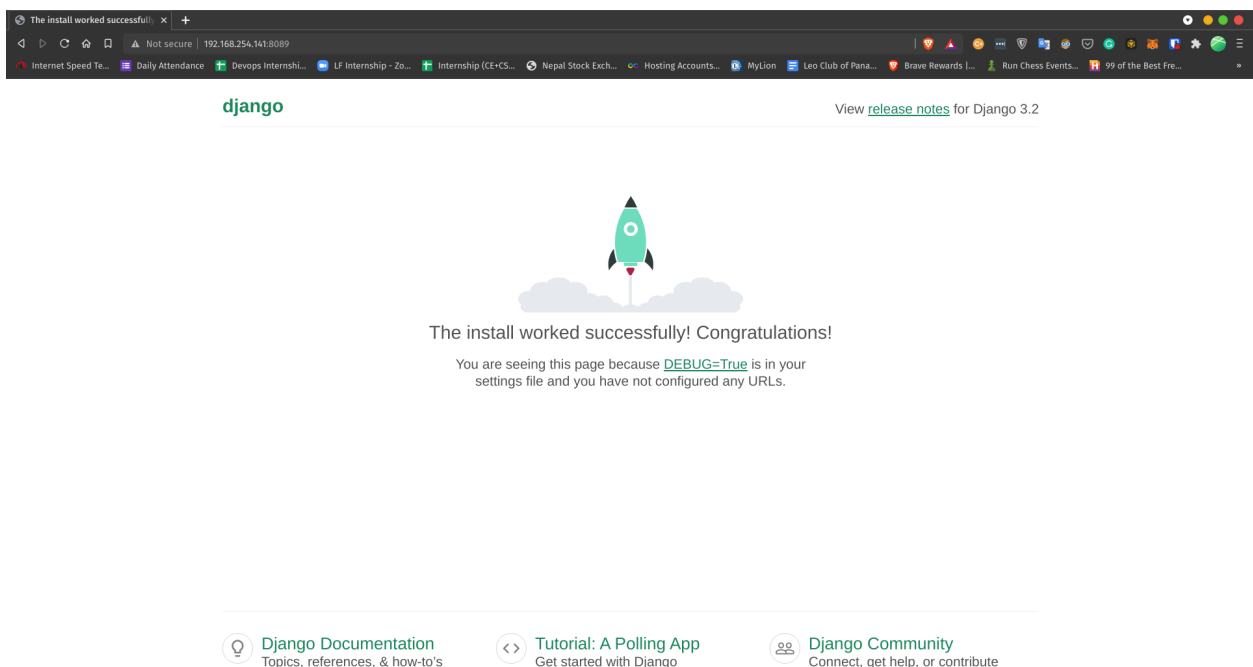
System check identified no issues (0 silenced).
November 12, 2021 - 13:14:17
Django version 3.2.7, using settings 'django_starter_project.settings'
Starting development server at http://0.0.0.0:8089/
Quit the server with CONTROL-C.
```

- Now we configure gunicorn
  - Firstly, let's make a configuration file for gunicorn server using the following commands:
    - mkdir gunicorn\_conf
    - nano config.py

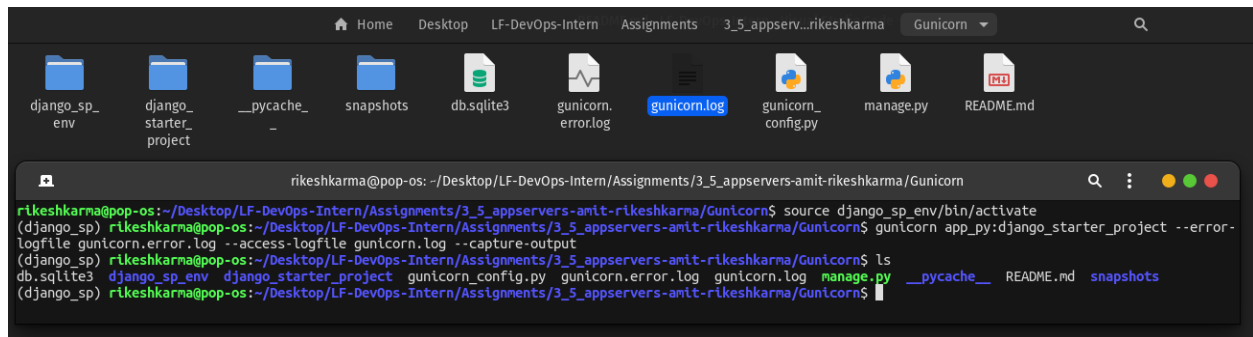


- Then add the following in the file:
  - `command='./django_sp_env/bin/gunicorn'`
  - `pythonPath='./'`
  - `bind = '192.168.254.141:8089'`
  - `workers = 3`
- After we configure the configuration file we need to run the following command which will deploy 3 instances of the application using Gunicorn in 8089 port:
  - `gunicorn -c gunicorn_config.py django_starter_project.wsgi`
- Now we can test the server in the browser by using `192.168.254.141:8089`

```
rireshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rireshkarma/Gunicorn
(django_sp) rakeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rireshkarma/Gunicorn$ gunicorn -c gunicorn_config.py django_starter_project.wsgi
[2021-11-14 15:47:18 +0545] [418172] [INFO] Starting gunicorn 20.1.0
[2021-11-14 15:47:18 +0545] [418172] [INFO] Listening at: http://192.168.254.141:8089 (418172)
[2021-11-14 15:47:18 +0545] [418172] [INFO] Using worker: sync
[2021-11-14 15:47:18 +0545] [418173] [INFO] Booting worker with pid: 418173
[2021-11-14 15:47:18 +0545] [418174] [INFO] Booting worker with pid: 418174
[2021-11-14 15:47:18 +0545] [418175] [INFO] Booting worker with pid: 418175
```



- Dump access logs in a file in the non-default pattern.
- Dump error logs in a file.
  - We can use following command to dump the log files:
    - `gunicorn app_py:django_starter_project --error-logfile gunicorn.error.log --access-logfile gunicorn.log --capture-output`
    - And we can ls to check the files inside
      - We can see 'gunicorn.error.log' and 'gunicorn.log' files now.



The screenshot shows a file manager window at the top with a dark theme. The breadcrumb path is 'Home > Desktop > LF-DevOps-Intern > Assignments > 3\_5\_appserv...rikeshkarma > Gunicorn'. The file list includes folders like 'django\_sp\_env', 'django\_starter\_project', and 'snapshots', and files like 'db.sqlite3', 'gunicorn.error.log', 'gunicorn.log' (highlighted in blue), 'gunicorn\_config.py', 'manage.py', and 'README.md'. Below the file manager is a terminal window with the title 'rikeshkarma@pop-os: ~/Desktop/LF-DevOps-Intern/Assignments/3\_5\_appservers-amit-rikeshkarma/Gunicorn'. The terminal shows the following commands and output:

```
rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ source django_sp_env/bin/activate
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ gunicorn app_py:django_starter_project --error-logfile gunicorn.error.log --access-logfile gunicorn.log --capture-output
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$ ls
db.sqlite3  django_sp_env  django_starter_project  gunicorn_config.py  gunicorn.error.log  gunicorn.log  manage.py  __pycache__  README.md  snapshots
(django_sp) rikeshkarma@pop-os:~/Desktop/LF-DevOps-Intern/Assignments/3_5_appservers-amit-rikeshkarma/Gunicorn$
```