

Deploy Django Application In Background Using Jenkins

Requirements :

- Django 3.0.3
- asgiref==3.2.3
- django-cors-headers==3.2.1
- djangorestframework==3.11.0
- pytz==2019.3
- sqlparse==0.3.0

Above all mentioned requirements are available in “requirements.txt” file
After cloning project it contains “requirement.txt” file, we have to run -

pip install -r requirement.txt

Following are the steps to deploy Django Application through Jenkins -

Steps :

- 1) Create a Freestyle job in jenkins.
- 2) Clone your project repository from github



The screenshot shows the Jenkins configuration interface for a new job. The 'Repositories' section is expanded, showing a 'Repository URL' field with the value 'https://github.com/biswojit01/react_django_demo_app.git'. There are also 'Credentials' and 'Build Environment' sections visible below.

3) Mention from which Branch you have to clone your repository.

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

4) Under Build Step choose the “Execute Shell” option and write following commands.

Build Steps

≡ Execute shell ?

Command

See [the list of available environment variables](#)

```
export BUILD_ID=dontkillme
pip install -r requirements.txt
nohup python3 manage.py runserver &
```

export BUILD_ID=dontkillme : This tells the operating system not to kill the process even if the terminal session ends.

pip install -r requirements.txt : This command installs the Python packages listed in the requirements.txt file.

nohup python3 manage.py runserver & :

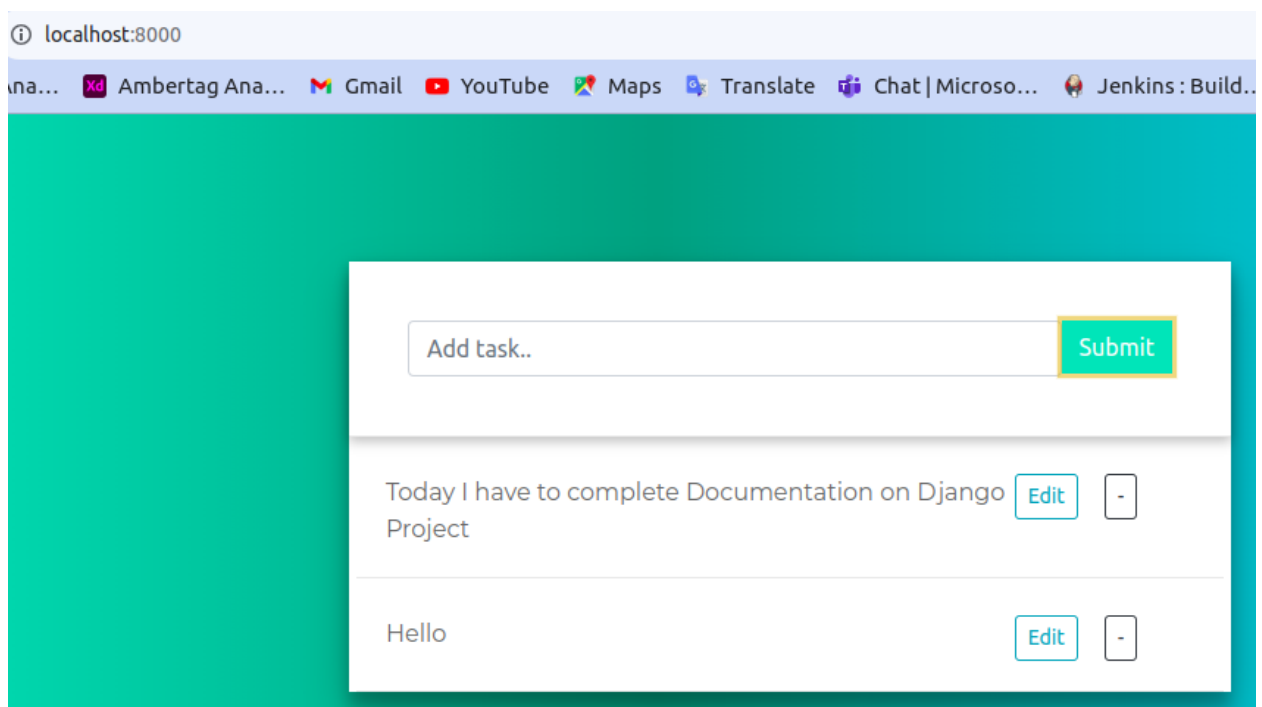
Nohup : A Unix command that stands for "no hang up." It's used to run a command in the background and prevent it from being terminated when the terminal session

Ends.

python3 manage.py runserver : This runs a development server for a Django web application (assuming you're using Django). It starts the server and listens for incoming requests.

& : This symbol runs the command in the background, allowing you to continue using the terminal for other tasks.

- 5) Save the job and After building the project we will get the following output.
Our application will run on localhost:8000



- 6) If we want to stop application then we need to kill process Id

```
sumit@sumit-ubuntu:/mnt/Django/New_Dj/react_django_demo_app$ ps aux | grep python3
root        609  0.0  0.1 48332 7116 ?        Ss   10:15   0:00 /usr/bin/python3 /usr/bin/networkd-d
root        771  0.0  0.2 126784 7888 ?        Ssl  10:15   0:00 /usr/bin/python3 /usr/share/unattenc
tdown --wait-for-signal
jenkins     11933 0.0  0.3 54552 15196 ?        S    15:02   0:00 python3 manage.py runserver
jenkins     11935 6.1  1.2 732260 48632 ?        RL   15:02   4:53 /usr/bin/python3 manage.py runserver
sumit       14150 0.0  0.0 17672  720 pts/1    S+   16:22   0:00 grep --color=auto python3
sumit@sumit-ubuntu:/mnt/Django/New_Dj/react_django_demo_app$ kill 11933
bash: kill: (11933) - Operation not permitted
sumit@sumit-ubuntu:/mnt/Django/New_Dj/react_django_demo_app$ sudo kill 11933
```