

## Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.

---

1) Configure a sua aplicação e suba-a manualmente. Você pode se guiar pelo script que o instrutor seguiu durante a aula:

```
# Passos para configurar a app e subir manualmente
# Criando o arquivo .env (temporário)
cd /vagrant/jenkins-todo-list/to_do/
vi .env

[config]
# Secret configuration
SECRET_KEY = 'r*5ltfzw-61ksdm41fuul8+hxs$86yo9%k1%k=( !@=-wv4qtyv'

# conf
DEBUG=True

# Database
DB_NAME = "todo_dev"
DB_USER = "devops_dev"
DB_PASSWORD = "mestre"
DB_HOST = "localhost"
DB_PORT = "3306"

# Instalando o venv
sudo pip3 install virtualenv nose coverage noseexcover pylint
# Criando e ativando o venv (dev)
cd ../
virtualenv --always-copy venv-django-todolist
source venv-django-todolist/bin/activate
pip install -r requirements.txt
# Fazendo a migracao inicial dos dados
python manage.py makemigrations
python manage.py migrate
# Criando o superuser para acessar a app
python manage.py createsuperuser
# Repetir o processo de migracaoção para o ambiente de produção:
vi to_do/.env

[config]
# Secret configuration
SECRET_KEY = 'r*5ltfzw-61ksdm41fuul8+hxs$86yo9%k1%k=( !@=-wv4qtyv'

# conf
DEBUG=True

# Database
DB_NAME = "todo"
DB_USER = "devops"
DB_PASSWORD = "mestre"
DB_HOST = "localhost"
DB_PORT = "3306"

# Fazendo a migracao inicial dos dados
```

```

# Fazendo a migração inicial dos dados
python manage.py makemigrations
python manage.py migrate

# Criando o superuser para acessar a app
python manage.py createsuperuser

# Verificar o ip do servidor
ip addr

# Rodando a app
python manage.py runserver 0:8000
http://192.168.33.10:8000

```

2) Configure o *daemon* do Docker. Você pode se guiar pelo script que o instrutor seguiu durante a aula:

```

# Expor o deamon do docker
sudo mkdir -p /etc/systemd/system/docker.service.d/
sudo vi /etc/systemd/system/docker.service.d/override.conf
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:2376
sudo systemctl daemon-reload
sudo systemctl restart docker.service

```

3) Faça o *build* da sua imagem. Você pode se guiar pelo script que o instrutor seguiu durante a aula:

```

# Sugested Plugins
# Instalando os plugins
Gerenciar Jenkins -> Gerenciar Plugins -> Disponíveis
# docker
Install without restart -> Depois reiniciar o jenkins
Gerenciar Jenkins -> Configurar o sistema -> Nuvem
# Name: docker
# URI: tcp://127.0.0.1:2376
# Enabled
# This project is parameterized:
DOCKER_HOST
tcp://127.0.0.1:2376
# Voltar no job criado na aula anterior
# Manter a mesma configuracao do GIT para desenvolvimento
# Build step 1: Executar Shell
# Validando a sintaxe do Dockerfile
docker run --rm -i hadolint/hadolint < Dockerfile
# Build step 2: Build/Publish Docker Image
Directory for Dockerfile: ./
Cloud: docker
Image: rafaelvzago/django_todolist_image_build

```