# Docker - Compose Configuration Tutorial

## **Tutorial**

In this tutorial we are going to create a simple app with nginx and use docker compose to build them separately and allow them to interact.

First make a file structure that is the same as the one below.

docker-compose-tutorial

├──nginx

| └──nginx.conf

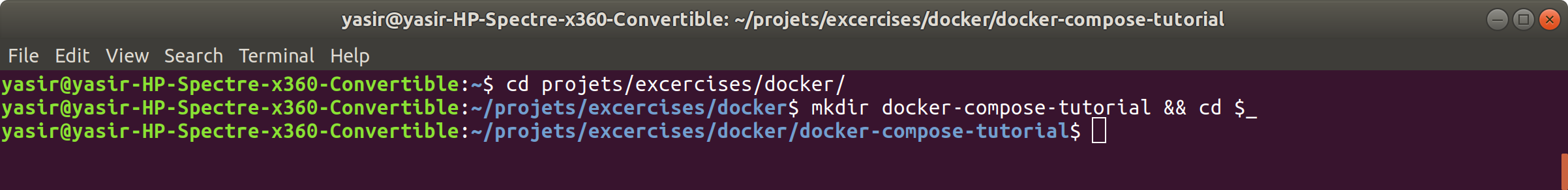
|

├──flask-app

| ├──app.py

| └──Dockerfile

└──docker-compose.yaml





Now let's configure the Nginx service, in your nginx.conf:

events {}

http {

server {

listen 80;

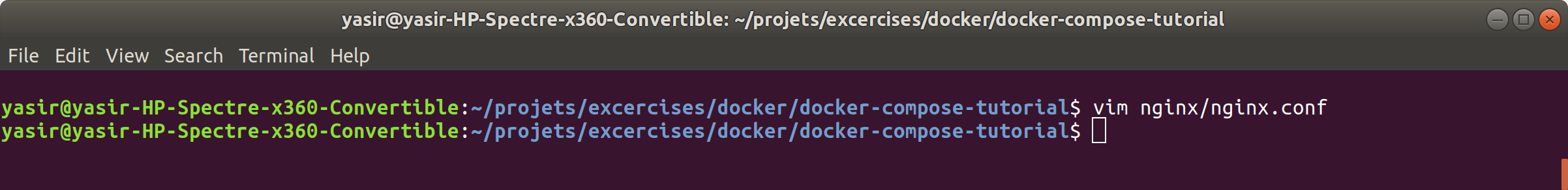
location / {

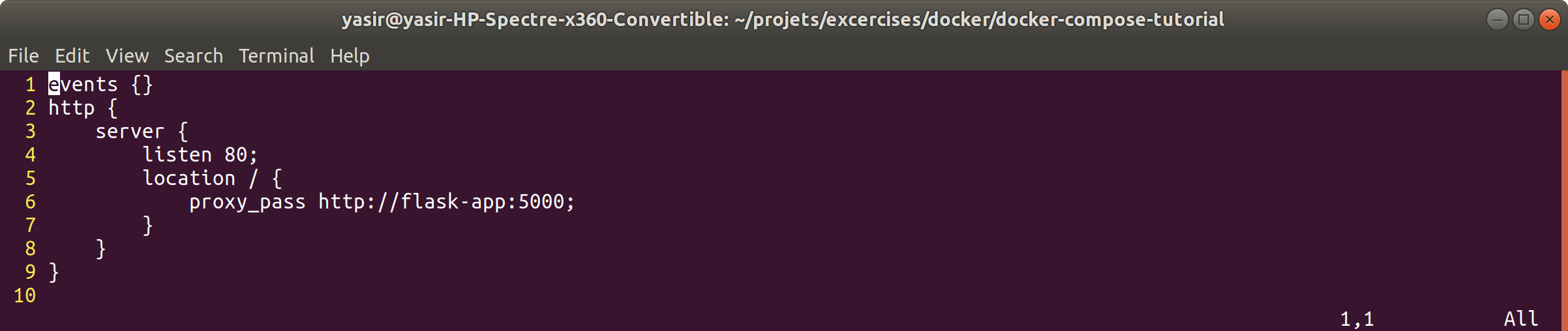
proxy\_pass http://flask-app:5000;

}

}

}





Now let's configure the app itself. Add this to your app.py.

from flask import Flask

app = Flask(\_\_name\_\_)

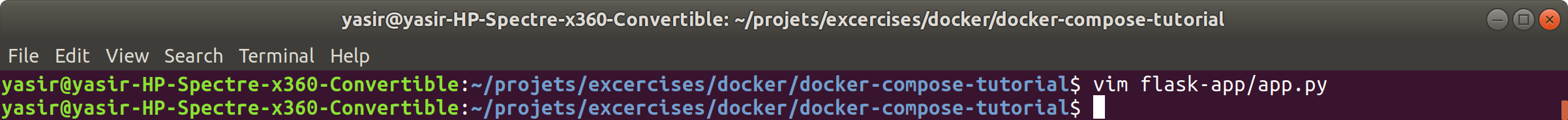
@app.route('/')

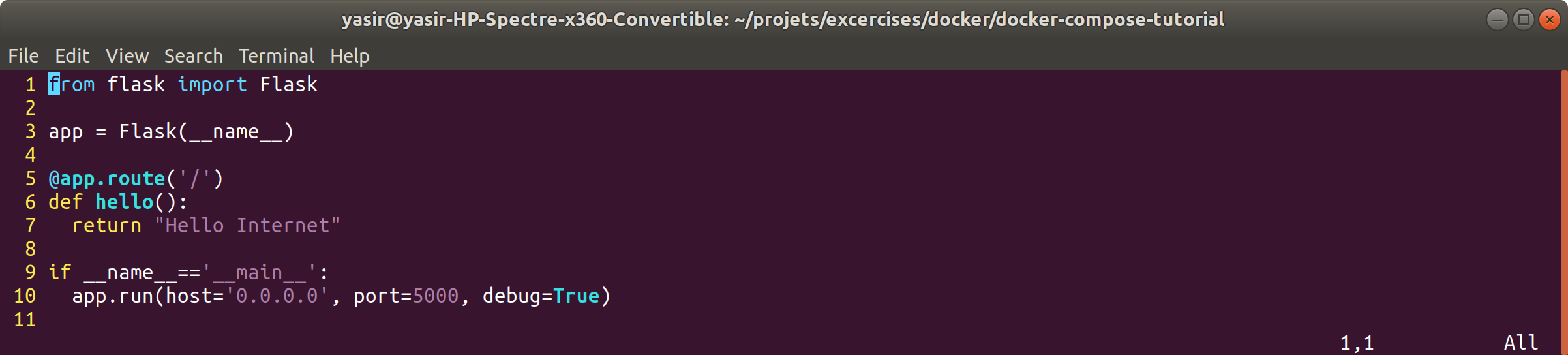
def hello():

return "Hello Internet"

if \_\_name\_\_=='\_\_main\_\_':

app.run(host='0.0.0.0', port=5000, debug=True)





And then in your Dockerfile.

FROM python:3.5

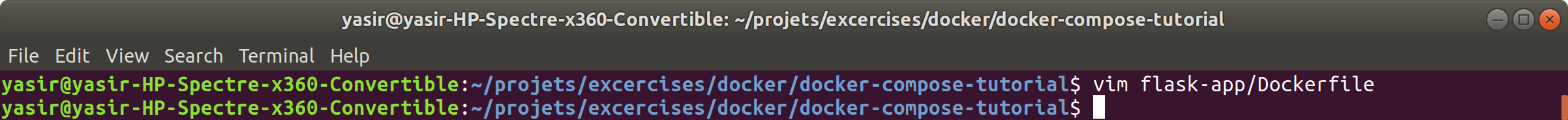
RUN apt-get update && apt-get install -y python3-pip

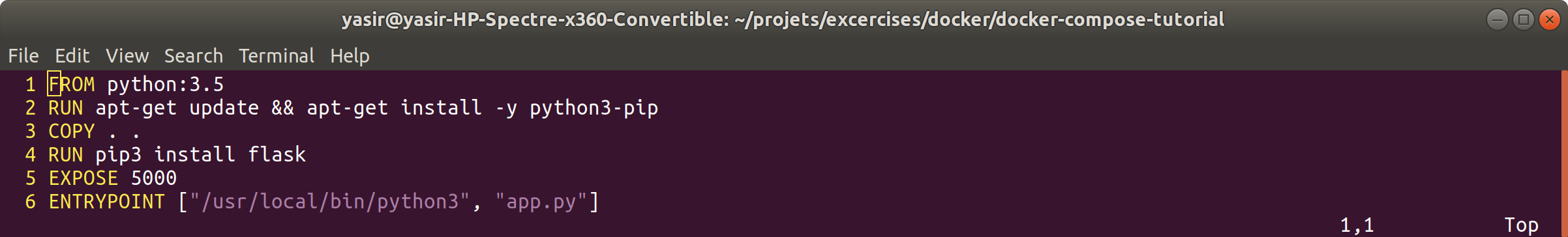
COPY . .

RUN pip3 install flask

EXPOSE 5000

ENTRYPOINT ["/usr/local/bin/python3", "app.py"]





And finally in your docker-compose.yaml.

version: '3.7'

services:

flask-app:

build: ./flask-app

ports:

- target: 5000

published: 5000

protocol: tcp

nginx:

image: nginx:latest

ports:

- target: 80

published: 80

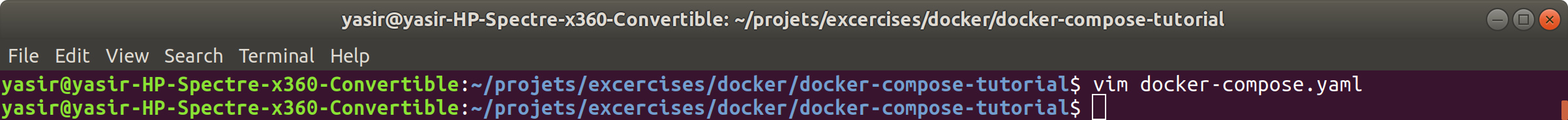
protocol: tcp

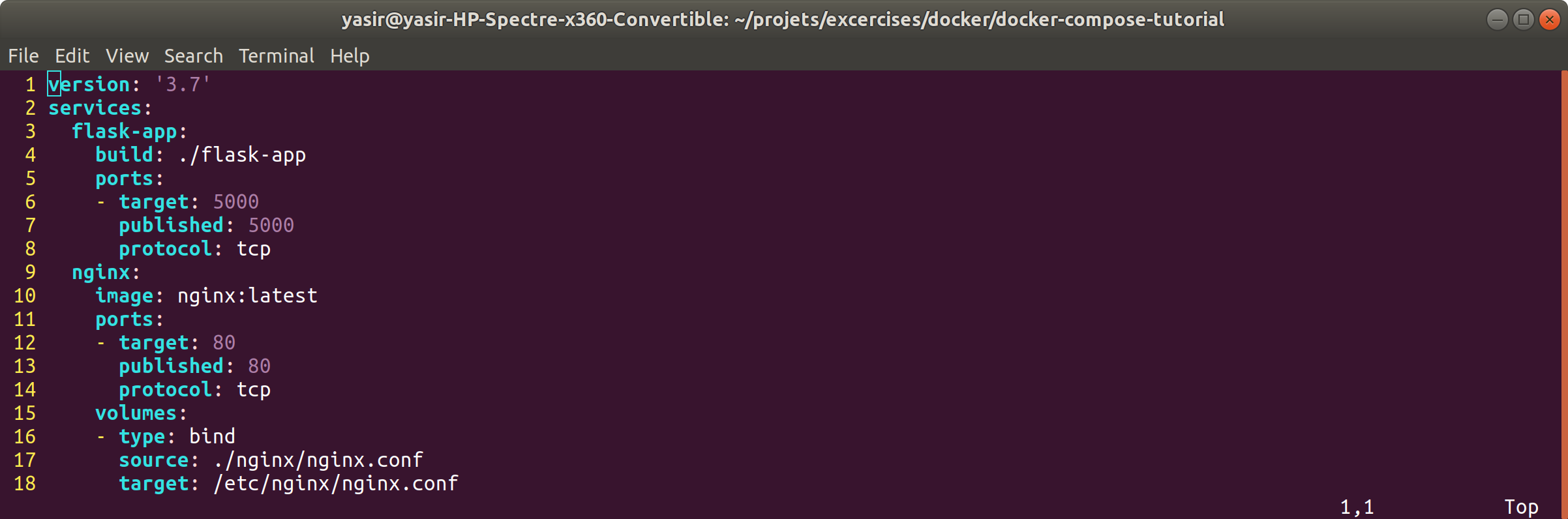
volumes:

- type: bind

source: ./nginx/nginx.conf

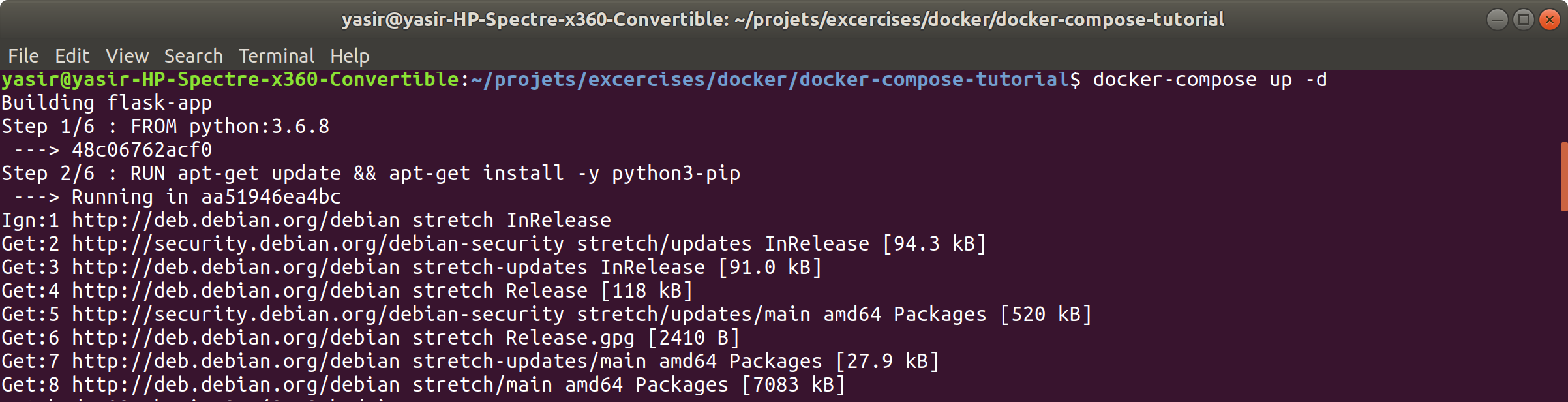
target: /etc/nginx/nginx.conf

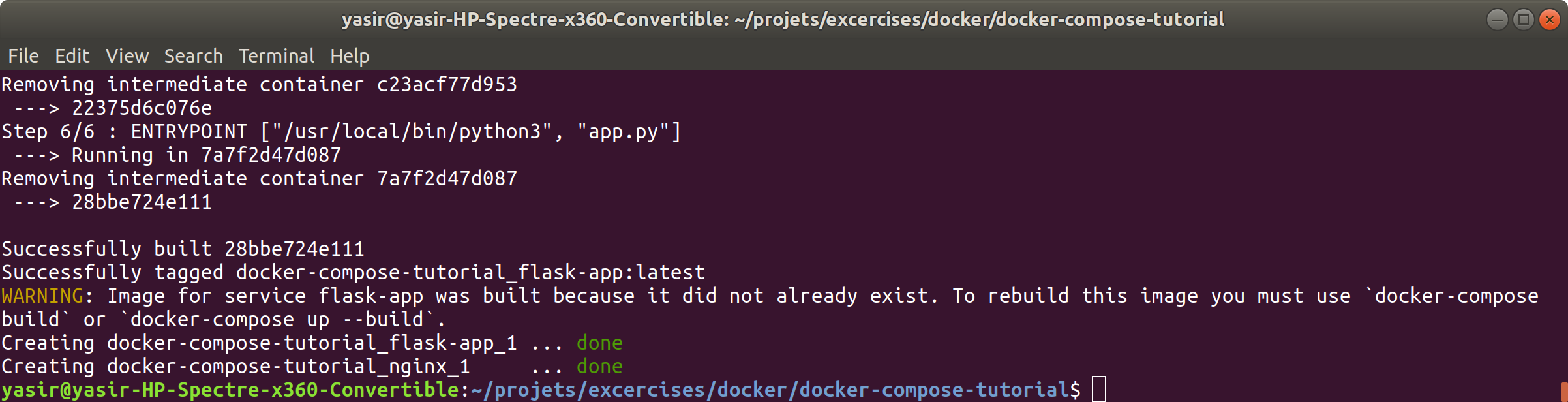




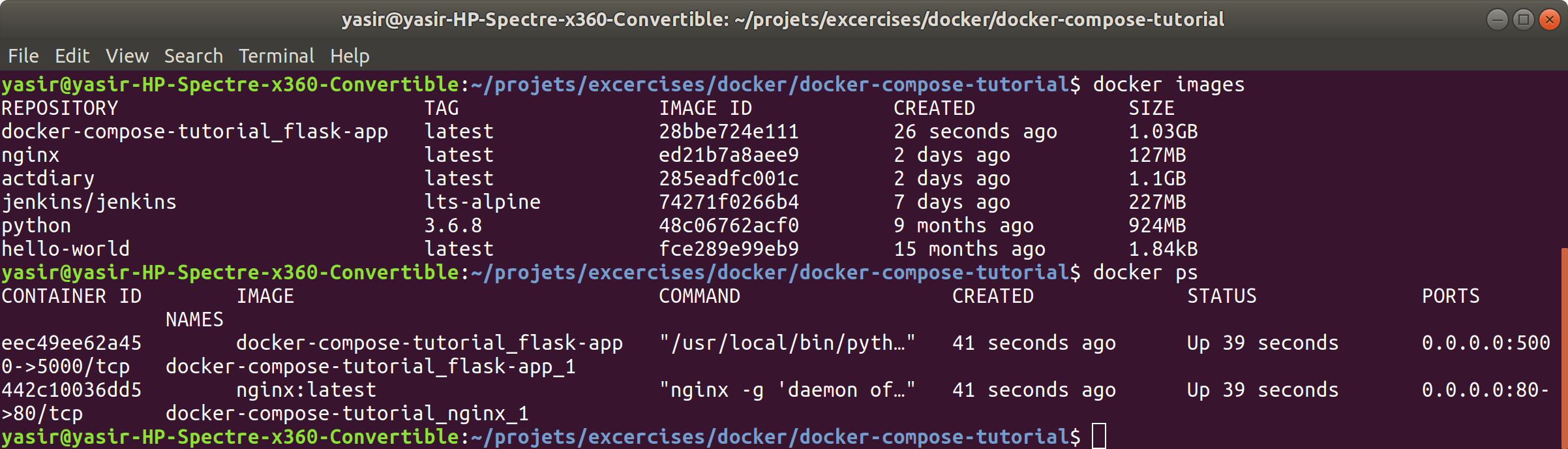
Now run the command:

**$ docker-compose up -d**





check the images and running containers



And visit localhost or your vms public IP address on port 80 and you will find your app.

