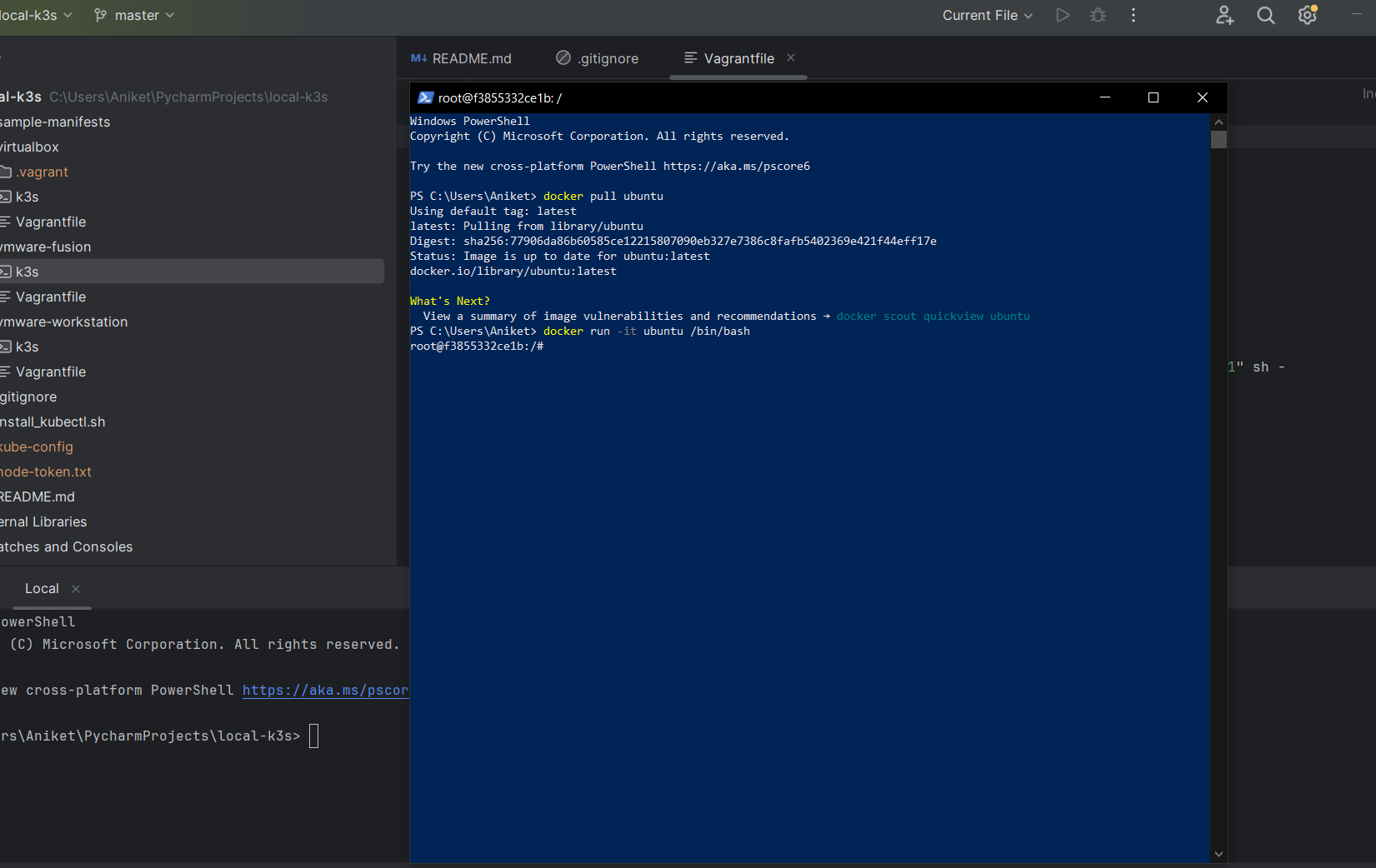
Part II: Practical Evaluation (2 hours)

1. (30 points) Using Docker, install a Ubuntu container and run a bash shell inside it. Provide the complete command sequence you used.
2. Use docker pull Ubuntu command to pull the image of Ubuntu.
3. Use docker run -it ubuntu /bin/bash to Run a container with a bash shell inside it.

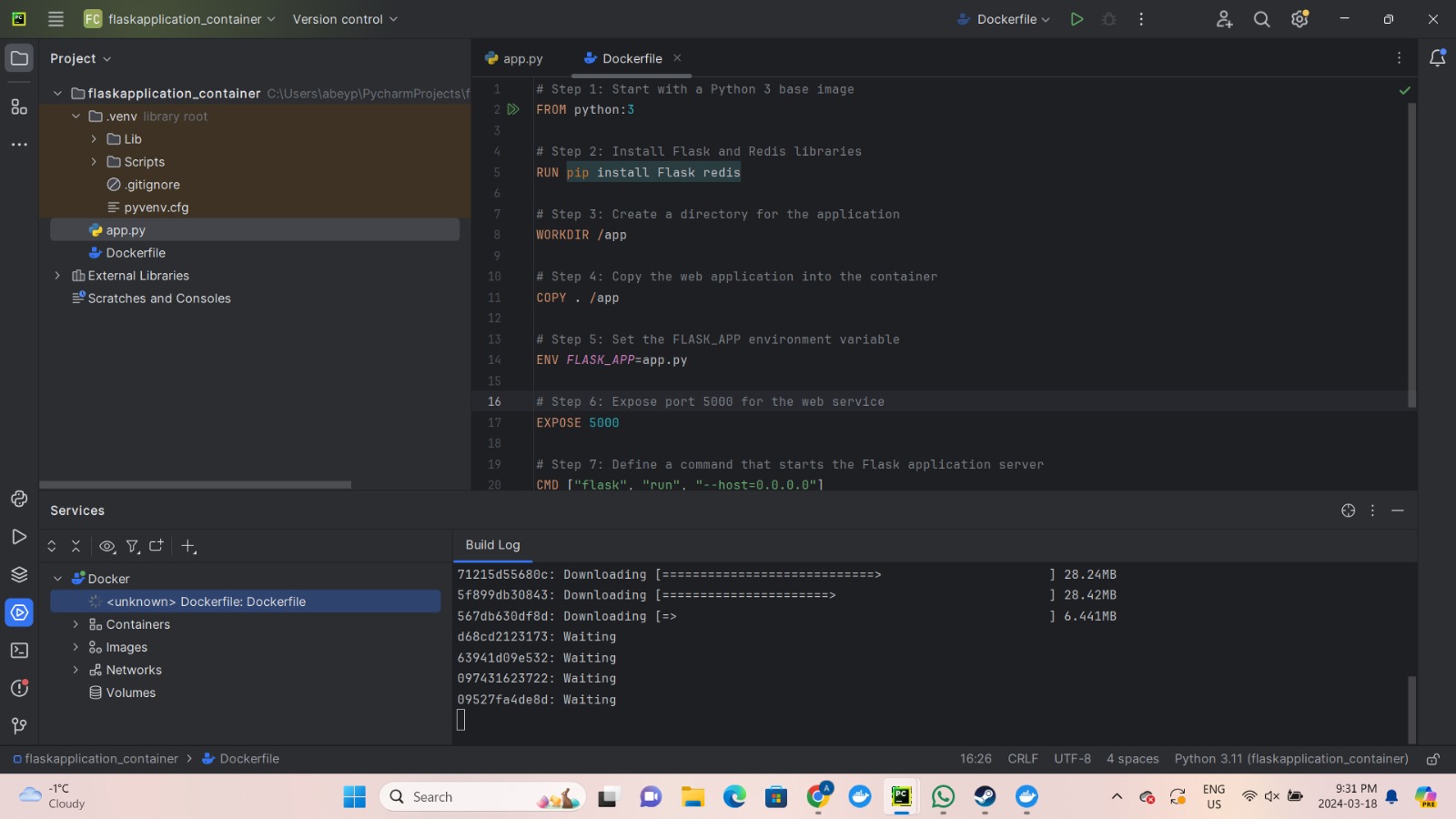


1. Create a Dockerfile that: a. Starts with a Python 3 base image b. Installs Flask and Redis libraries c. Copies a web application you've created into the container d. Sets an environment variable called "FLASK\_APP" to the name of your application file e. Exposes port 5000 for the web service f. Defines a command that starts the Flask application server

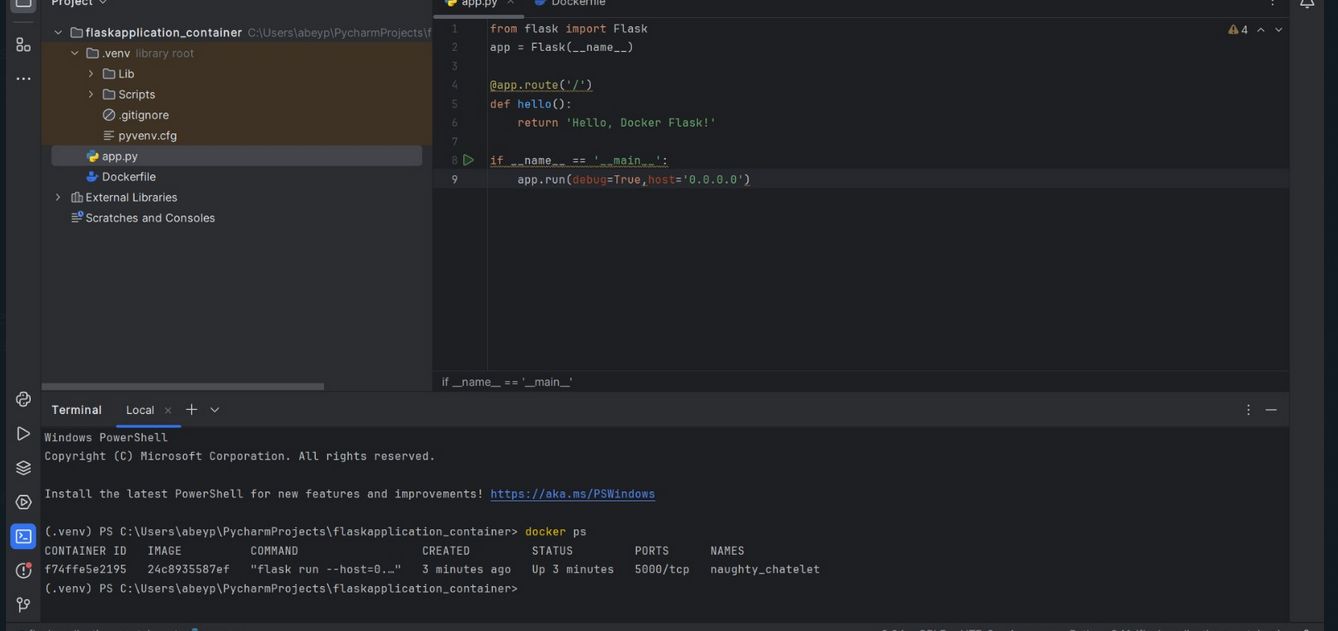
**DockerFile:**

# Step 1: Start with a Python 3 base image  
FROM python:3  
  
# Step 2: Install Flask and Redis libraries  
RUN pip install Flask redis  
  
# Step 3: Create a directory for the application  
WORKDIR /app  
  
# Step 4: Copy the web application into the container  
COPY . /app  
  
# Step 5: Set the FLASK\_APP environment variable  
ENV *FLASK\_APP*=app.py  
  
# Step 6: Expose port 5000 for the web service  
EXPOSE 5000  
  
# Step 7: Define a command that starts the Flask application server  
CMD ["flask", "run", "--host=0.0.0.0"]

**Creating an image**





****

**docker-Compose.yml**

**Build using the same Image**

**version: '3'**

**services:**

**webapp:**

**image: 24c8935587ef729d7a268c7c34ed59be5b9898804e00832baf5603919e0fb2d4**

**build: .**

**ports:**

**- "8000:5000"**

**environment:**

**- REDIS\_HOST=redis**

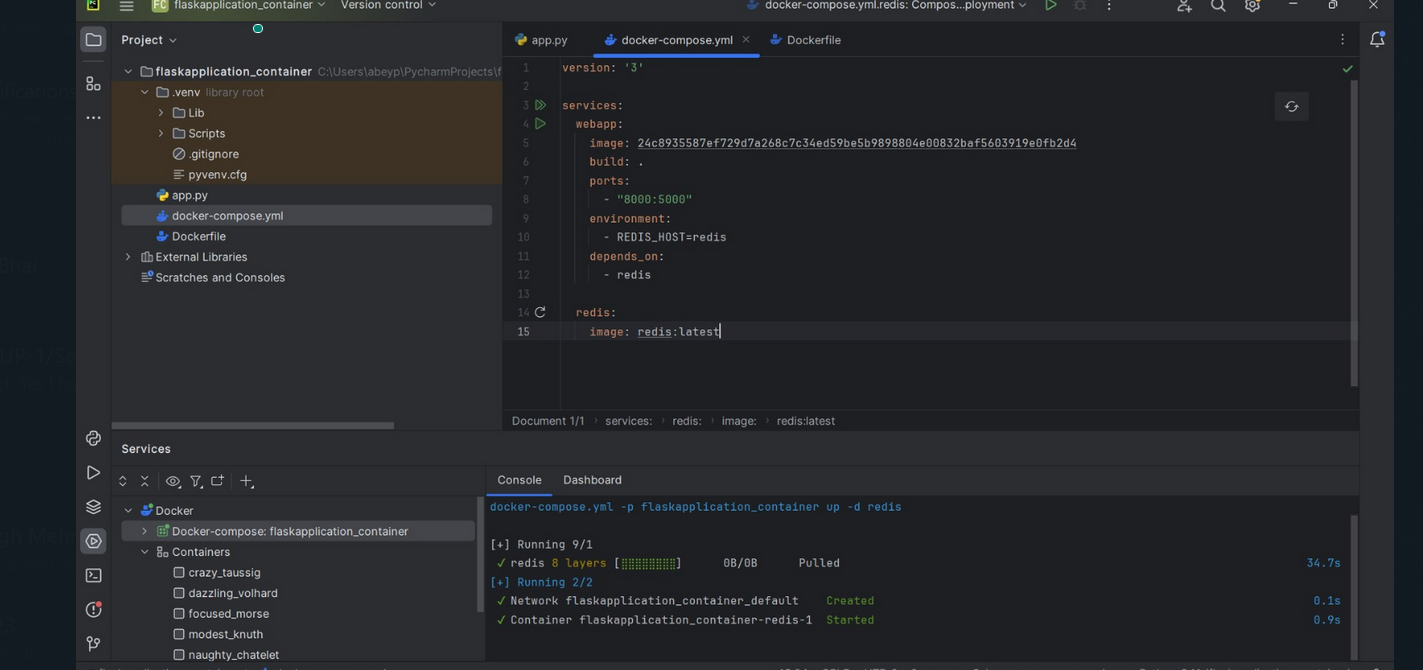
**depends\_on:**

**- redis**

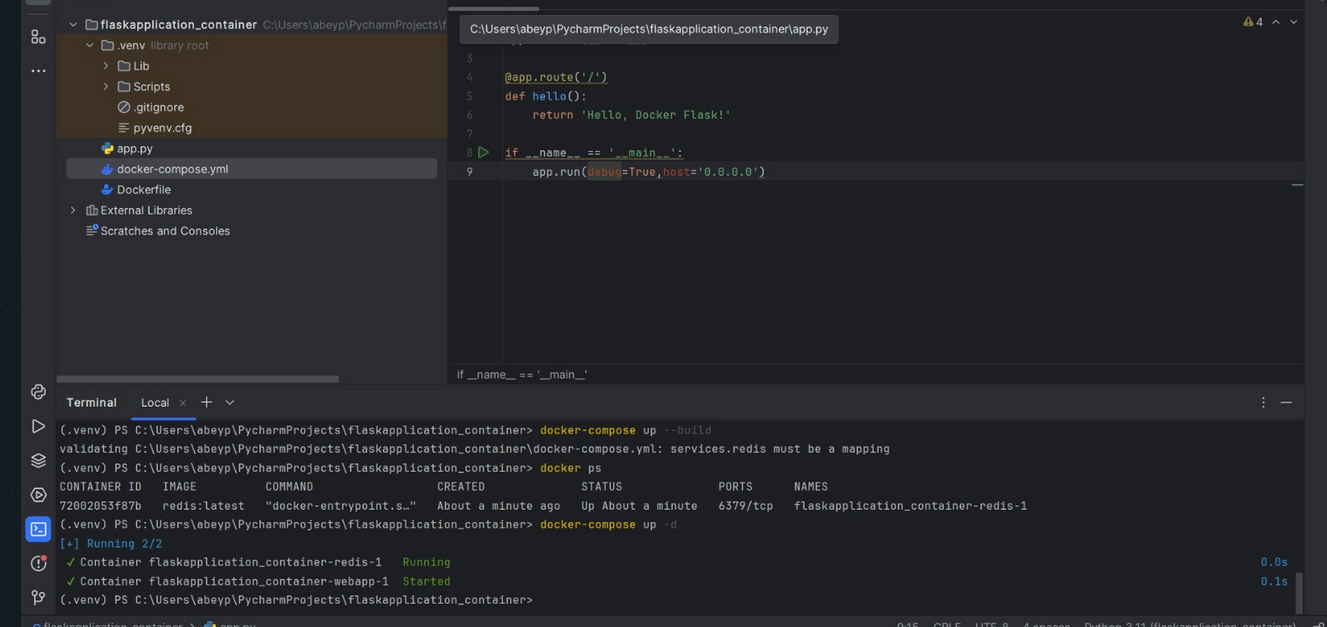
**redis:**

**image: redis:latest**

**docker-compose up -–build to build the container**

****

**User docker-compose up –d to run the container**

****

****

**Thank You**