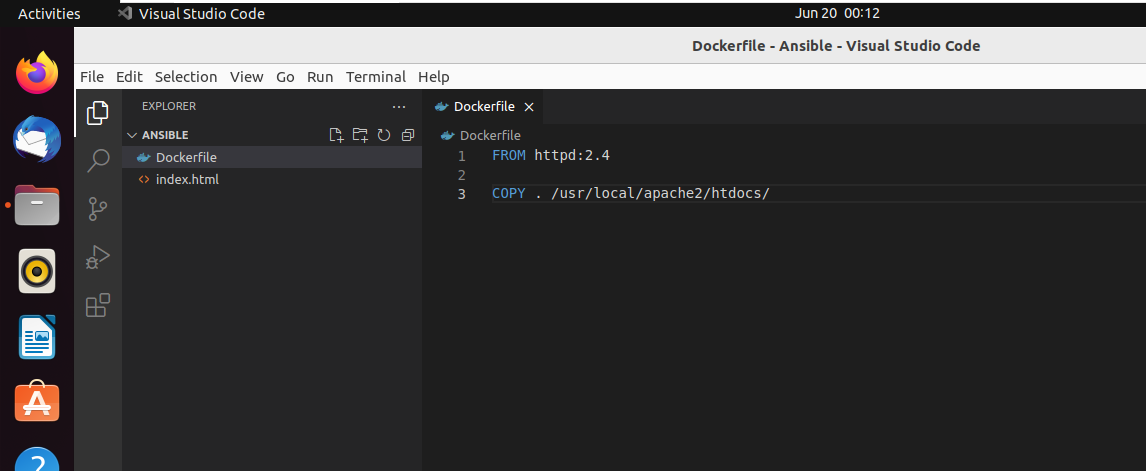
Ansible Documentation

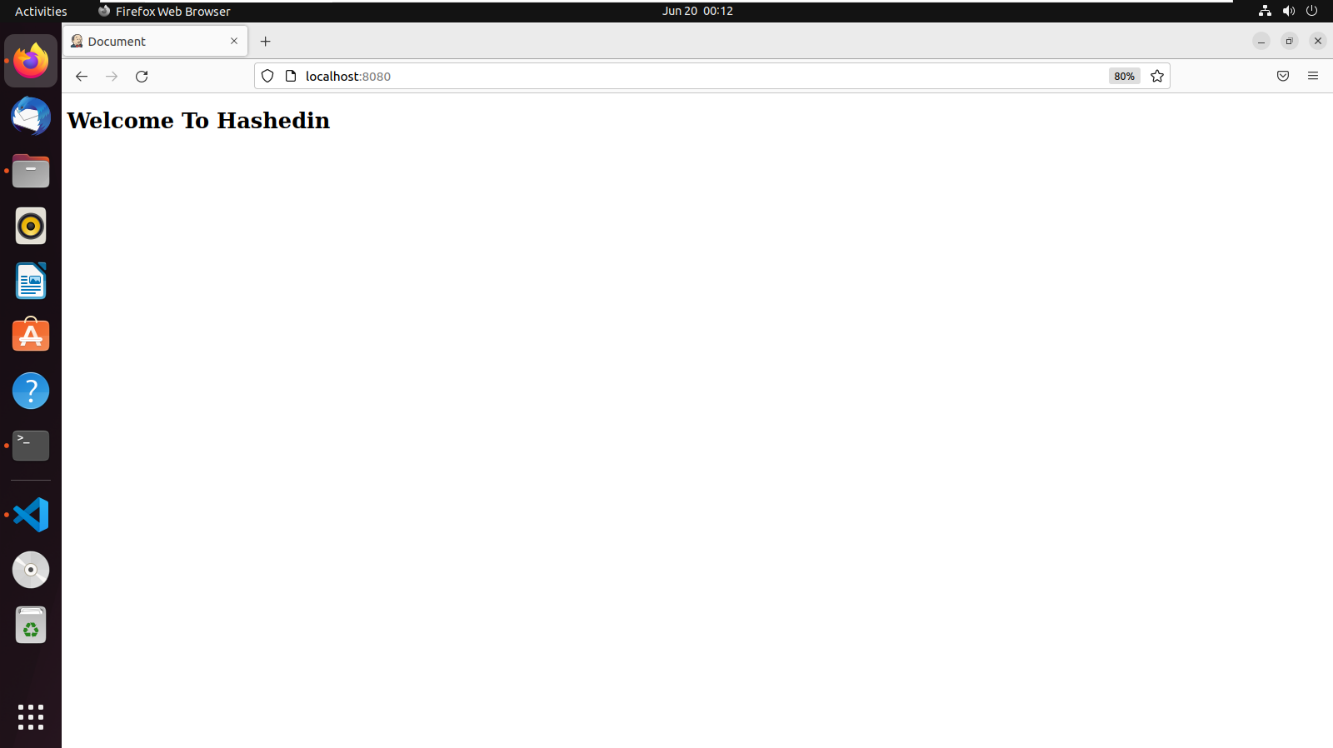
MySql

1. Create a MySql Database named **flask\_example\_app** inside the docker container named**mysql-container** running the image **mysql:latest**. Create a Table named **users**with attributes *username* and *password*of type *varchar*.
2. Given [here](https://amedeloitte-my.sharepoint.com/:u:/r/personal/sanksaxena_deloitte_com/Documents/flask-app.zip?csf=1&web=1&e=FNQY86) is a zipped codebase for a "simplest" Flask application written in a rush by a developer on last *Friday* evening. The app is a "webapp" that allows users to Register, Login, and Logout into the application portal. It makes use of MongoDB to store the data such as email, password, name, etc of the end-user.  
     
   After partying endlessly on *Friday* night, the developer has decided to escape the *Monday* blues, and hence took an unplanned leave. The developer forgot that he has to demo the application on the same day to the client. Sadly, you are the only remaining person in the team along with the manager, who is available on Monday and will be taking initiative to show case his work to the client.  
     
   While coding in a rush on Friday evening, the developer has missed some general configurations in the application, due to which it is unable to connect with the MongoDB running in the container. Now, you have to FIX IT, make the application ready for DEMO and SAVE the team!
3. Create a **dump** of the MySql Database **flask\_example\_app** (The one you created in the first step)
4. Run two docker containers using mysql:latest image and configure them as Master-Slave setup.

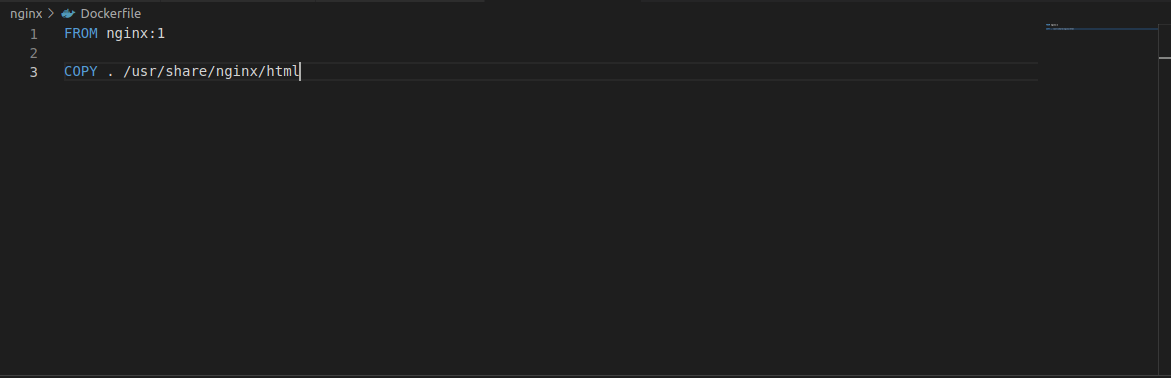
Webserver

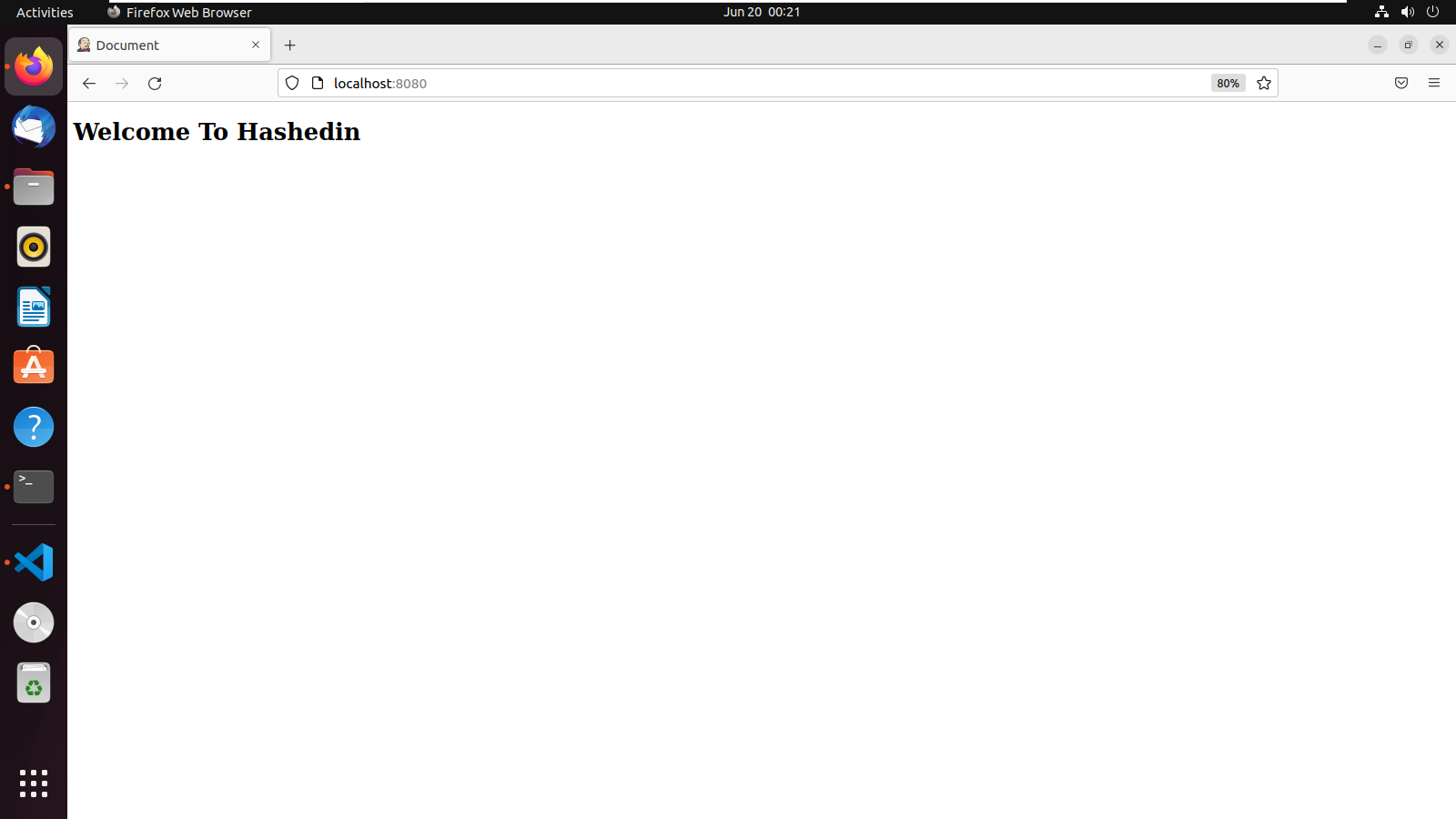
1. Run the Apache server on the docker container with the name **apache-welcome-container** and change the home page to “**Welcome to HashedIn**”.





1. Run the *Nginx server* on the docker container with the name **nginx-welcome-container** and change the home page to “**Welcome to HashedIn**".

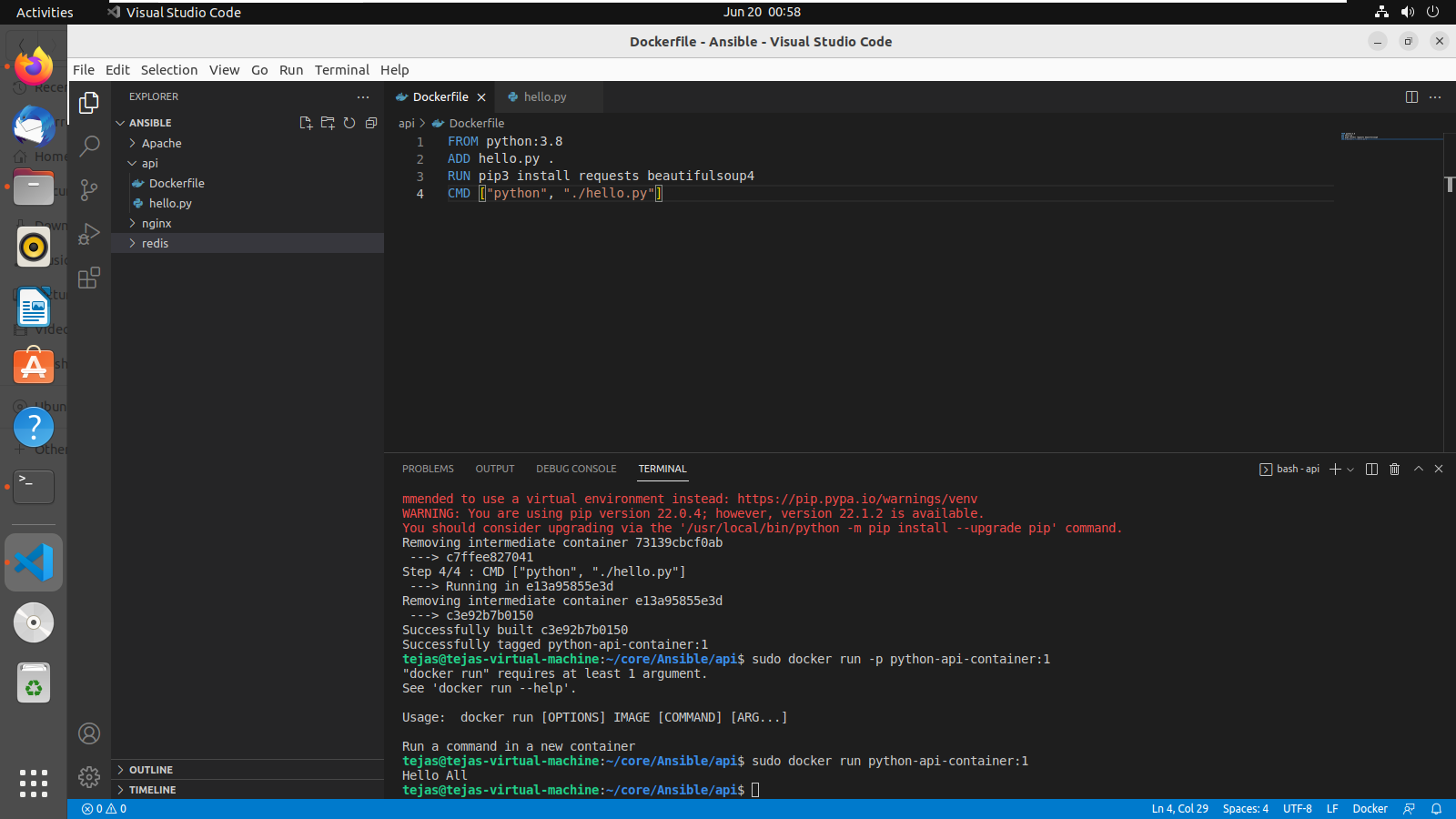




API:

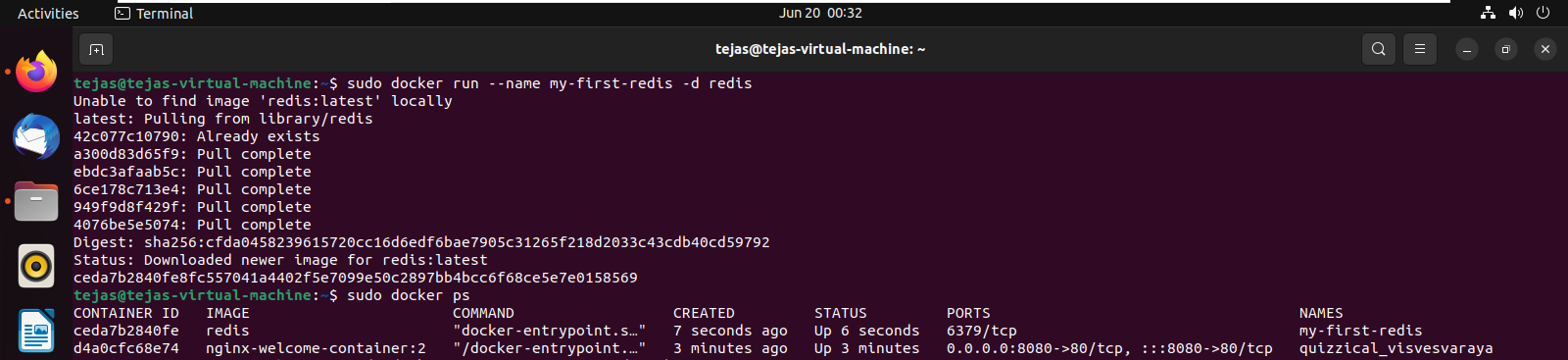
1. Write a python program and run it inside the docker container with the name python-api-container that fetches details of a movie provided as an input by the user. Use OmdbApi to fetch the movie. Display Year, Director, Actors, Plot, and ImdbRating as the output.

API URL: <http://www.omdbapi.com/>

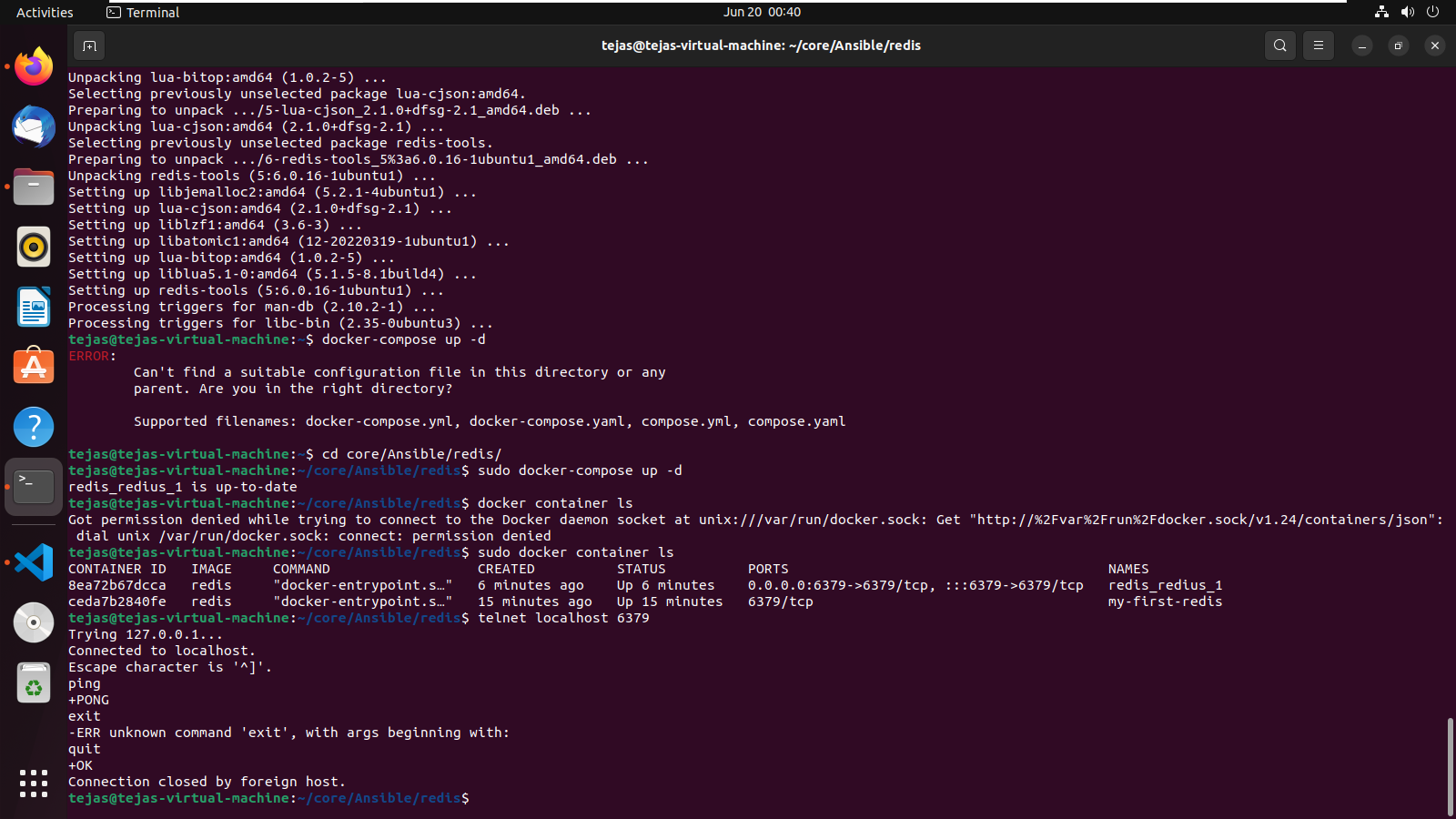


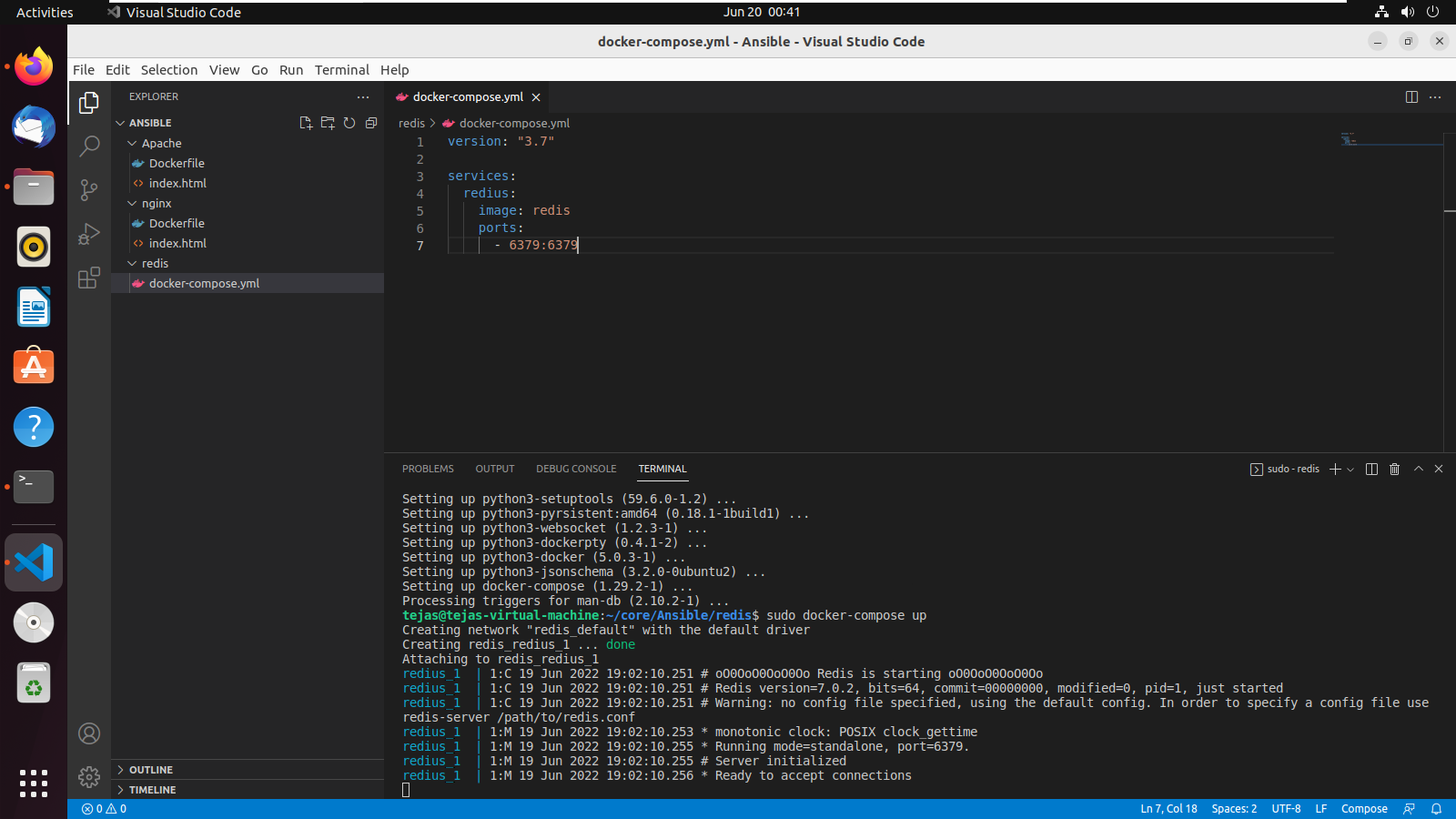
Redis

1. Configure Redis by running it inside a docker container and test the same using **redis-cli**.









1. Configure master-slave architecture of Redis using docker containers.