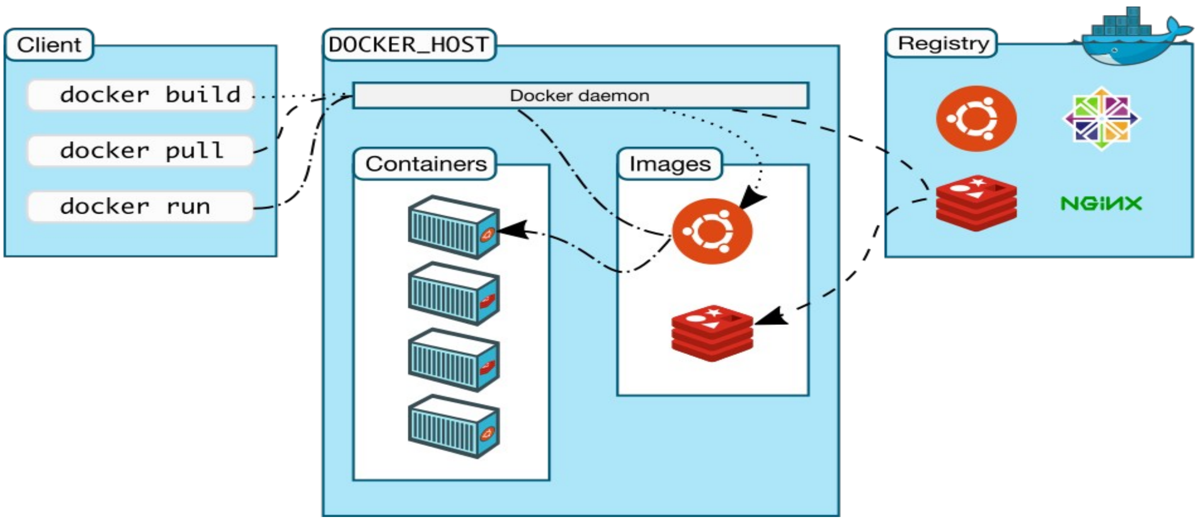
Experiment No. 9

Aim: To study and implement containerization using Docker

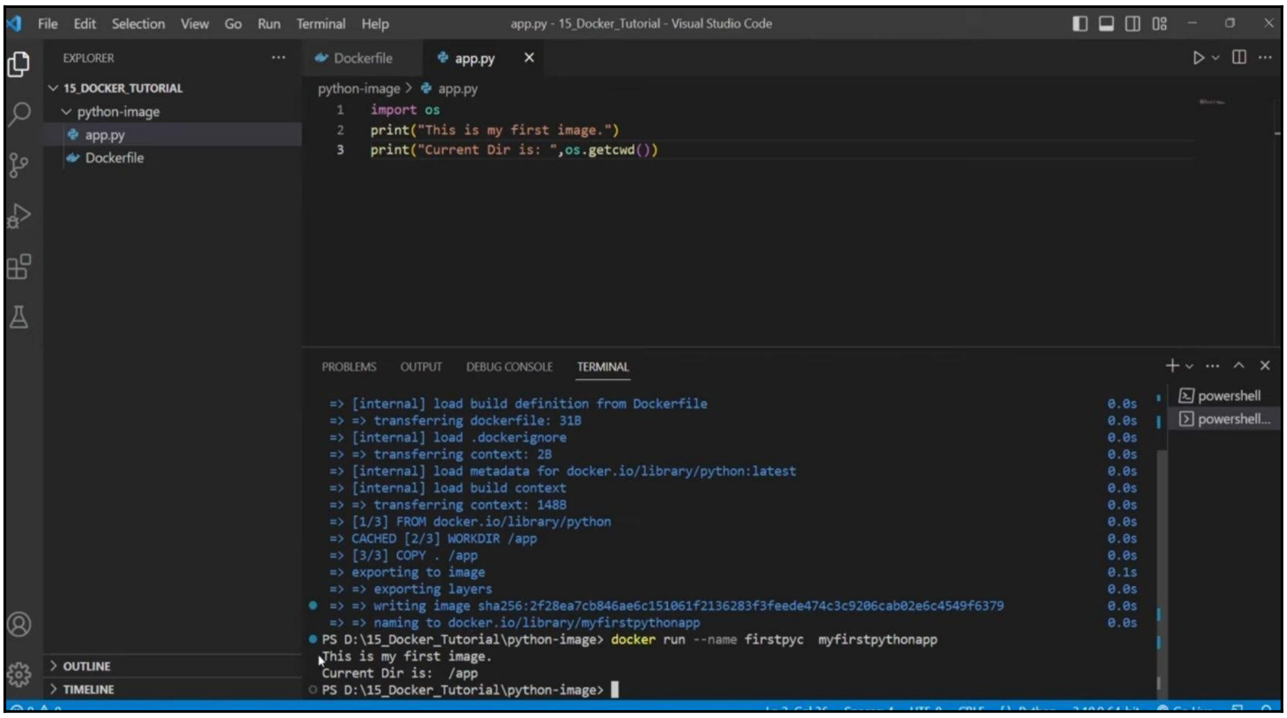
Theory:

- Docker is an open container management platform.
- It is a software platform for developing, shipping, and running applications based on containers --- small and lightweight execution environments that make shared use of the operating system kernel and run it in isolation from one another.
- Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.



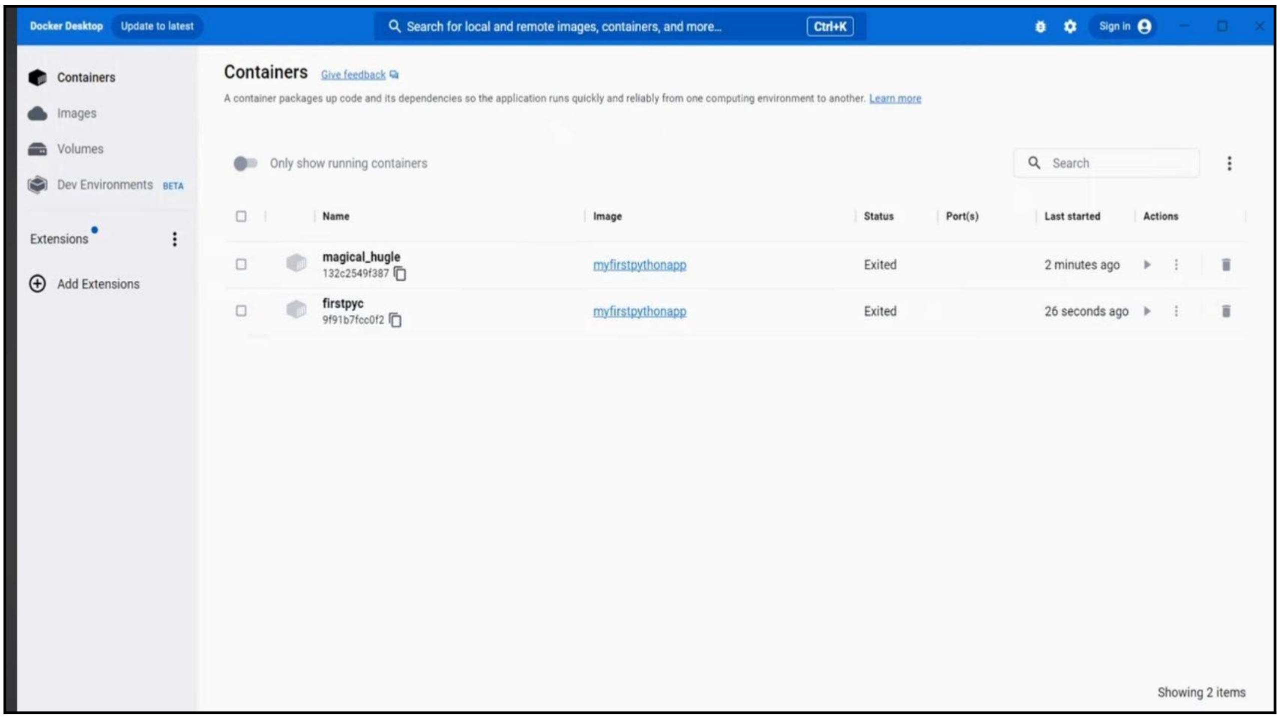
Containers isolate application environments from one another, and only share the underlying OS kernel. Containers are an abstraction at the app layer that packages code and dependencies together. By default, a container is relatively well isolated from other containers and its host machine.
You can control how isolated a container's network, storage, or other underlying subsystems are from other containers or from the host machine.
Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.

Output:


File Edit Selection View Go Run Terminal Help
app.py - 15_Docker_Tutorial - Visual Studio Code
EXPLORER
15_DOCKER_TUTORIAL
python-image
app.py
Dockerfile
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
powershell...

```python
import os
print("This is my first image.")
print("Current Dir is: ",os.getcwd())
```

```
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 31B                                                     0.0s
 => [internal] load .dockerignore                                                       0.0s
 => => transferring context: 2B                                                         0.0s
 => [internal] load metadata for docker.io/library/python:latest                        0.0s
 => [internal] load build context                                                       0.0s
 => => transferring context: 148B                                                       0.0s
 => [1/3] FROM docker.io/library/python                                                 0.0s
 => CACHED [2/3] WORKDIR /app                                                           0.0s
 => [3/3] COPY . /app                                                                   0.0s
 => exporting to image                                                                  0.1s
 => => exporting layers                                                                 0.0s
 => => writing image sha256:2f28ea7cb846ae6c151061f2136283f3feede474c3c9206cab02e6c4549f6379  0.0s
 => => naming to docker.io/library/myfirstpythonapp                                     0.0s
PS D:\15_Docker_Tutorial\python-image> docker run --name firstpyc  myfirstpythonapp
This is my first image.
Current Dir is:  /app
PS D:\15_Docker_Tutorial\python-image>
```

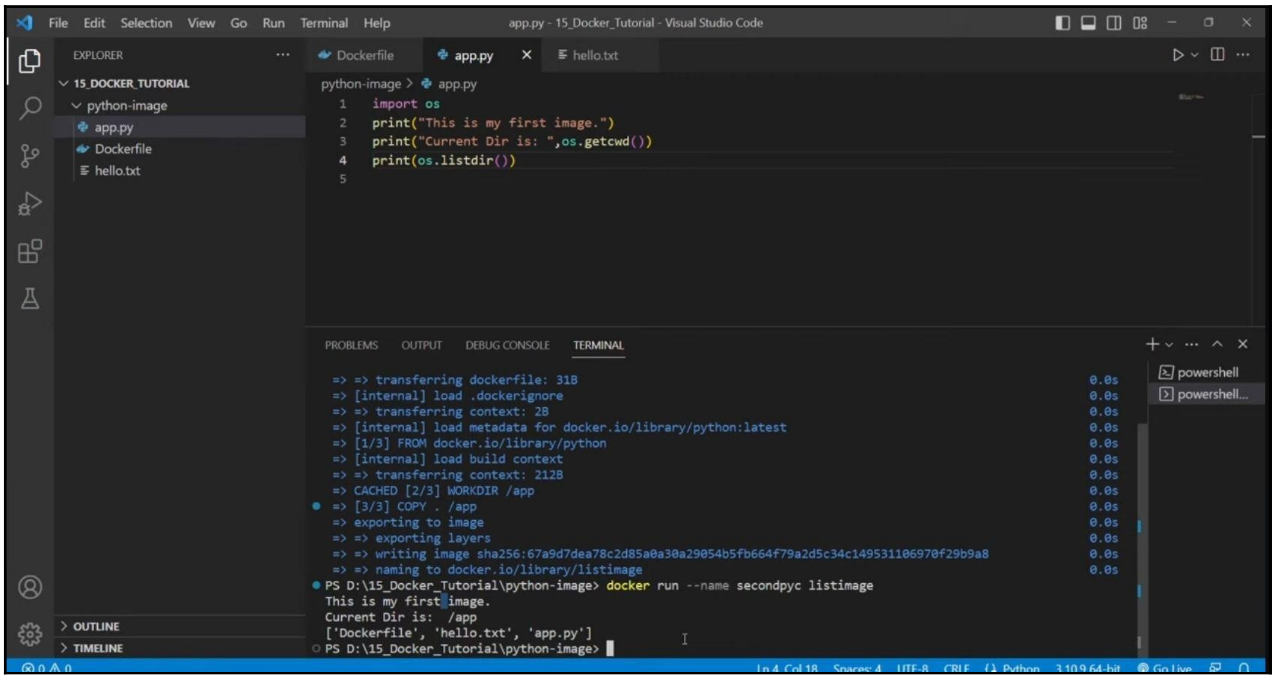


Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another.

Name	Image	Status	Port(s)	Last started	Actions
magical_hugle 132c2549f387	myfirstpythonapp	Exited		2 minutes ago	
firstpyc 9f91b7fcc0f2	myfirstpythonapp	Exited		26 seconds ago	

Showing 2 items



```python
import os
print("This is my first image.")
print("Current Dir is: ",os.getcwd())
print(os.listdir())
```

```
 => => transferring dockerfile: 31B                                                      0.0s
 => [internal] load .dockerignore                                                        0.0s
 => => transferring context: 2B                                                          0.0s
 => [internal] load metadata for docker.io/library/python:latest                         0.0s
 => [1/3] FROM docker.io/library/python                                                  0.0s
 => [internal] load build context                                                        0.0s
 => => transferring context: 212B                                                        0.0s
 => CACHED [2/3] WORKDIR /app                                                            0.0s
 => [3/3] COPY . /app                                                                    0.0s
 => exporting to image                                                                   0.0s
 => => exporting layers                                                                  0.0s
 => => writing image sha256:67a9d7dea78c2d85a0a30a29054b5fb664f79a2d5c34c149531106970f29b9a8  0.0s
 => => naming to docker.io/library/listimage                                             0.0s
PS D:\15_Docker_Tutorial\python-image> docker run --name secondpyc listimage
This is my first image.
Current Dir is:  /app
['Dockerfile', 'hello.txt', 'app.py']
PS D:\15_Docker_Tutorial\python-image>
```

Conclusion: Docker is a powerful tool for packaging, distributing, and running applications within lightweight, isolated containers. Its applications span across various domains, including software development, deployment, and operations. In software development, Docker facilitates consistency across different environments, streamlining the development process and enabling developers to work in isolated environments without worrying about dependencies. For deployment, Docker simplifies the process of deploying applications across different infrastructure environments, from local development machines to cloud platforms, ensuring consistency and reproducibility. In operations, Docker enables efficient resource utilization through containerization, allowing for scalability and flexibility in managing infrastructure resources. Overall, Docker revolutionizes the way applications are built, shipped, and run, offering benefits such as portability, efficiency, and consistency throughout the software development lifecycle.