1.What is meant by Docker

Docker image

Docker Container

Docker file

**Solution:-**

Docker is an open-source platform that allows you to automate the deployment and management of applications within lightweight, isolated containers. It provides a standardized way to package applications and their dependencies into a portable and consistent environment.

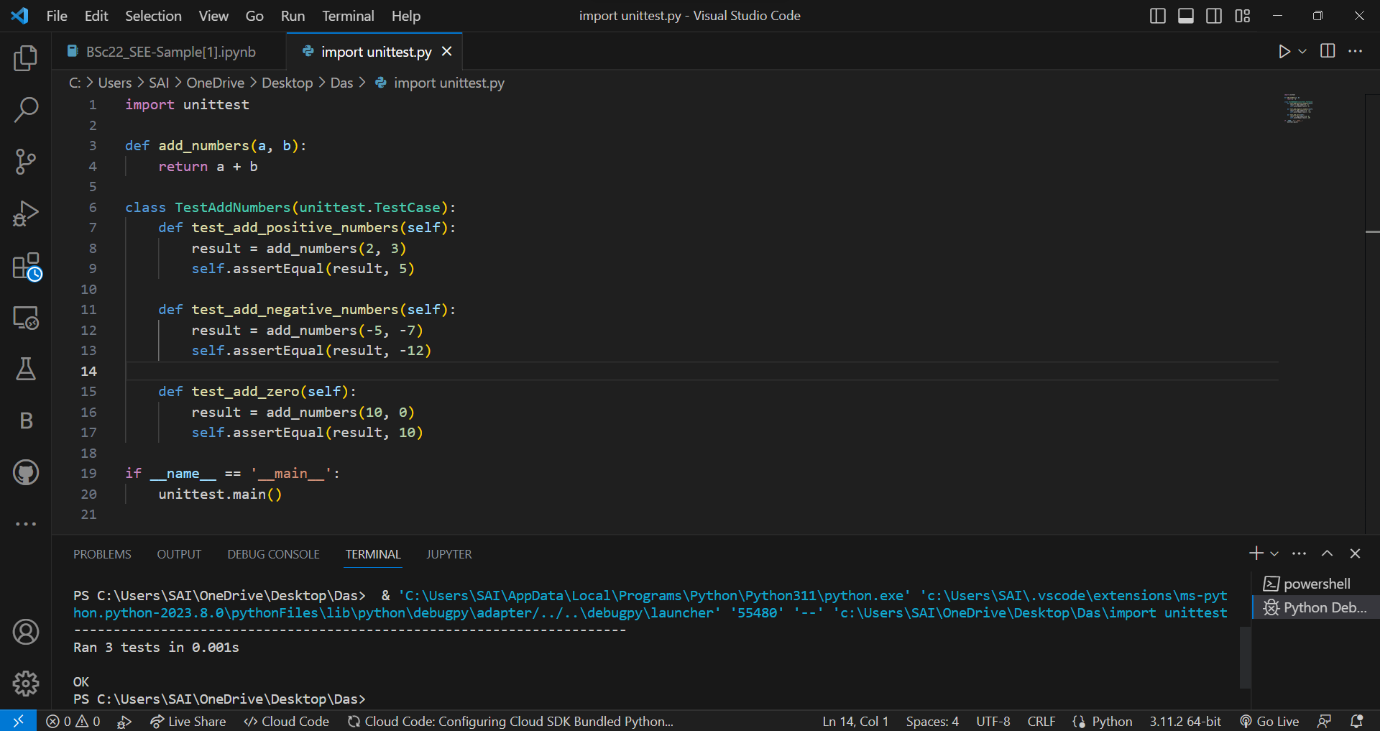
Docker Image: A Docker image is a lightweight, standalone, and executable package that includes everything needed to run an application, including the code, runtime, libraries, and system tools. It is built based on a set of instructions defined in a Dockerfile.

Docker Container: A Docker container is a running instance of a Docker image. It provides an isolated environment in which an application can run consistently, regardless of the underlying host system. Each container shares the host system's kernel but has its own isolated file system, processes, and network interfaces.

Dockerfile: A Dockerfile is a text file that contains a set of instructions used to build a Docker image. It defines the base image, the environment, the dependencies, and the steps to set up and configure the application within the image. Dockerfiles follow a declarative syntax, allowing developers to define the desired state of the image.

In summary, Docker simplifies application deployment by packaging applications and their dependencies into portable containers. Docker images provide the building blocks for containers, while Docker containers offer an isolated runtime environment. Dockerfiles define the instructions for building Docker images, ensuring consistent and reproducible deployments across different environments.

1.Write a simple Python program , Go through the link, and identify and study how it works



In this example, we define a simple function **add\_numbers** that takes two numbers as input and returns their sum.

Then, we create a test case class **TestAddNumbers** that inherits from **unittest.TestCase**. Inside this class, we define multiple test methods, each starting with the prefix **test\_**. These methods are responsible for testing different aspects of the **add\_numbers** function.

In the **test\_add\_positive\_numbers** method, we call **add\_numbers(2, 3)** and use the **self.assertEqual** method to assert that the result should be equal to **5**. Similarly, the other test methods check different scenarios.

Finally, we use the **unittest.main()** function to run the tests. This will discover all the test methods in the **TestAddNumbers** class and execute them.

By running this program, you can observe how the **unittest** module executes the test methods and reports the results. It will indicate whether each test passed or failed, along with additional information if a test fails.

Note: To run this program, you need to have the **unittest** module installed. You can install it by running **pip install unittest** in your terminal or command prompt. However, in most Python installations, the **unittest** module comes bundled with the standard library, so you don't need to install it separately.

2. Write Python Program and store it in GIT/GitHub.

Solution:- [saigawand.github.io/python-/python.py](https://saigawand.github.io/python-/python.py)

3. Explain how tool maven works.

**Solution: -** Maven is a build automation and project management tool for Java projects. It uses an XML file called `pom.xml` to configure the project and manage dependencies. Maven handles tasks like downloading dependencies, compiling code, running tests, generating documentation, and more. It follows a convention-over-configuration approach and has a standard build lifecycle with phases like `compile` and `test`. Maven relies on repositories to store and retrieve artifacts. It provides a command-line interface and generates build output in the `target` directory.

4. Explain how Jenkins can be used as an orchestration tool.

**Solution:-** Jenkins is an automation server used for orchestration. It supports continuous integration, pipelines, and distributed builds. With its plugin ecosystem, Jenkins integrates with various tools and services. It provides monitoring, notifications, and scalability. Jenkins helps automate development, testing, and deployment processes, enabling efficient CI/CD pipelines and fostering collaboration**.**