**UDT Tools**

what is **Devonfw**

Devonfw is an open-source software development platform designed to help organizations build and manage applications efficiently.

what is **Tabnine**

Tabnine is an AI-powered code completion tool designed to enhance the productivity of software developers.

what is **Sonarlint**

SonarLint is a linting tool for software development that helps developers identify and fix quality and security issues in their code

what is **Smart UI Builder**

Smart UI Builder is a tool or platform that allows developers to build user interfaces (UI) for applications quickly and easily.

.

what is **Diffblue**

Diffblue is that provides AI-powered code analysis and test generation tools for software developers.

what is **Ponicode**

Ponicode is an AI-powered code completion tool for software developers. It uses machine learning algorithms to suggest code snippets and **complete lines of code in real-time, helping developers write code faster and more accurately**

what is **Jacoco**

JaCoCo is a free code coverage library for Java, which provides measurements of how much of the code is executed during testing

what is **Snyk**

Snyk is a security platform for developers. It provides tools for identifying and fixing vulnerabilities in the open-source components and dependencies used in software development.

what is **Jet Brains Space**

JetBrains Space is a team collaboration and productivity platform developed by JetBrains. It provides a central space for teams to store and manage their code, track tasks and issues, and collaborate on projects.

what is **Jhipster**

JHipster is an open-source application generator for modern web applications. It allows developers to generate a complete, customizable, and production-ready web application in a matter of minutes.

What is **TestIM**

TestIM is a cloud-based testing platform that provides a comprehensive solution for automated testing, test management, and continuous integration. It helps software teams improve the quality of their applications by automating the testing process and integrating it into their development workflow.

**Docker:**

**Docker** is a container management service. The keywords of Docker are **develop, ship** and **run** anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

**Docker Hub** is a registry service on the cloud that allows you to download Docker images that are built by other communities. You can also upload your own Docker built images to Docker hub.we will see how to download and the use the Jenkins Docker image from Docker hub.

This will be used to download the Jenkins image onto the local Ubuntu server.

$ docker pull jenkins

**Image:** In Docker, everything is based on **Images**. An image is a combination of a file system and parameters. Let’s take an example of the following command in Docker.

$Docker build ImageName ---- to create a image

$docker run hello-world

* The Docker command is specific and tells the Docker program on the Operating System that something needs to be done.
* The **run** command is used to mention that we want to create an instance of an image, which is then called a **container**.
* Finally, "hello-world" represents the image from which the container is made.

## Displaying Docker Images

To see the list of Docker images on the system, you can issue the following command.

docker images

## Removing Docker Images

The Docker images on the system can be removed via the **docker rmi** command. Let’s look at this command in more detail.

$ docker rmi ImageID

🡪The output will show only the Image ID’s of the images on the Docker host.

sudo docker images -q

🡪This command is used see the details of an image or container.

docker inspect Repository

**Containers:** are instances of Docker images that can be run using the Docker run command. The basic purpose of Docker is to run containers. Let’s discuss how to work with containers.

Docker ps ------ it shows the all the running containers available in system

Docker ps -a ------- it shows the all the containers available in ur system.

## docker history

With this command, you can see all the commands that were run with an image via a container.

### **Syntax**

$docker history ImageID

* **ImageID** − This is the Image ID for which you want to see all the commands that were run against it.

## docker top

With this command, you can see the top processes within a container.

### **Syntax**

$docker top ContainerID

* **ContainerID** − This is the Container ID for which you want to see the top processes.

## docker stop

This command is used to stop a running container.

### **Syntax**

docker stop ContainerID

* **ContainerID** − This is the Container ID which needs to be stopped.

## docker rm

This command is used to delete a container.

### **Syntax**

docker rm ContainerID

## docker kill

This command is used to kill the processes in a running container.

### **Syntax**

docker kill ContainerID

Docker – Container Lifecycle

The following illustration explains the entire lifecycle of a Docker container.

