|  |
| --- |
| **Name:** WASIMA QAYYUMUDDIN SHAIKH  **RollNo:**6220071  **Class:** T.E.I.T  **Sem:** V  **Subject:** ADVACE DEVOPS LAB **(Addevops**)  **EXPERIMENT NO: 10**  **Q1. What is hub.docker.com?**    Docker Hub (link resides outside IBM) is the public repository of Docker images that calls itself the “world’s largest library and community for container images.” It holds over 100,000 container images sourced from commercial software vendors, open-source projects, and individual developers. It includes images that have been produced by Docker, Inc., certified images belonging to the Docker Trusted Registry, and many thousands of other images.  All Docker Hub users can share their images at will. They can also download predefined base images from the Docker filesystem to use as a starting point for any containerization project.  Docker Hub is a service provided by Docker for finding and sharing container images with your team. It is the world’s largest repository of container images with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers.  Users get access to free public repositories for storing and sharing images or can choose a subscription plan for private repositories.  Docker Hub provides the following major features:   * Repositories: Push and pull container images. * Teams & Organizations: Manage access to private repositories of container images. * Docker Official Images: Pull and use high-quality container images provided by Docker. * Docker Verified Publisher Images: Pull and use high- quality container images provided by external vendors. * Builds: Automatically build container images from GitHub and Bitbucket and push them to Docker Hub. * Webhooks: Trigger actions after a successful push to a repository to integrate Docker Hub with other services.   Docker provides a Docker Hub CLI tool (currently experimental) and an API that allows you to interact with Docker Hub. Browse through the Docker Hub API documentation to explore the supported endpoints.  Why use Docker Hub   * **A large library of trusted images-**Docker Certified images, Verified Publisher images (which are Docker Certified and verified by the publisher), and Official Images published by Docker add a layer of trust for users. With millions — or in some cases billions — of downloads for many commonly used images, you can count on a reliable base image when you use Docker hub. While that’s great from the user perspective, it also benefits publishers as hosting an image in Docker Hub can give your project more exposure. * **A free tier-**Currently, Docker’s free plan offers unlimited public repositories and 1 private repository with up to 3 collaborators. This is useful for basic testing to get you familiar with the platform. However,[recent changes to the terms of service](https://dev.to/jfrog/how-everyone-can-win-in-the-new-era-of-docker-hub-limits-3knd) make it unsuitable for serious development. * **Built-in security features-**All accounts can benefit from local image vulnerability scans. “Team” accounts also gain access to audit-logs and multifactor authentication (MFA) to further secure repositories. * **Integrations & features that enable CI/CD**– Docker Hub also supports GitHub & Bitbucket integrations, automated tests, build triggers, and webhooks to help automate development pipelines and enable CI/CD (continuous integration/continuous delivery).   **Q2. What Is Docker Hub Used For?**  Docker Hub is used for:   1. Docker Image Repositories - A Docker Image repository is a place where Docker Images are actually stored, compared to the image registry which is a collection of pointers to this images. 2. Working With Dockerfiles - The Dockerfile is essentially the build instructions to build the Docker image. The advantage of a Dockerfile over just storing the binary image is that the automatic builds will ensure you have the latest version available. 3. Running Docker Containers - All docker containers run one main process. After that process is complete the container stops running. 4. Working With Docker Hub - Docker Hub is a cloud-based repository in which Docker users and partners create, test, store and distribute container images. Through Docker Hub, a user can access public, open source image repositories, as well as use a space to create their own private repositories, automated build functions, and work groups. 5. Docker Container Management - The true power of Docker container technology lies in its ability to perform complex tasks with minimal resources. If not managed properly they will bloat, bogging down the environment and reducing the capabilities they were designed to deliver. 6. Storing Data Within Containers – It is possible to store data within the writable layer of a container. Docker offers three different ways to mount data into a container from the Docker host: volumes, bind mounts, or tmpfs volumes.   **Q3. Install docker on AWS EC2 –Ubuntu by using curl (curl -fsSLhttps://get.docker.com -o get-docker.sh; sh get-docker.sh)**   1. Create EC2 instance Service → EC2 →Launch Instance        1. Then choose AMI here we are choosing Ubuntu Server 20.4      1. Then in Configure Security Group → Add Rule → (HTTP)        1. Then review and launch by downloading key-value pair      1. After launching the instance, we need to connect to our instance by clicking on connect. Then we need to select SSH client.      1. After that open Termux qpp -> locate private key file by using cd command 🡪 Execute chmod 400 wasima12.pem for publically viewable to connect instance execute command ssh -I “wasima12.pem” and it’s Public DNS.      1. Then type ‘sudo su’ for entering root user. Then type curl command for installing docker To install docker write the command. #curl –fsSL https://get.docker.com -o get-docker.sh.     #sh get-docker.sh    **Q4. Run Hello-World From Docker Hub And Explain The Steps**  **In this we don’t have hello-world container so it will connect to global**  **repository and then download it and then print Hello world**  **Q5. Pull 3 Or 4 Images One Of The Python Run “Hello World” Inside Container.**  **Python image**  **MySQL image**  **Openjdk image**  **Jruby image**  **Ubuntu image**  **Run hello world**  **Q6. Demonstrate Any 15 Docker Command And Explain Its Uses**   * 1. Docker Version: Show the Docker version information   2. Docker images: List images      * 1. Docker ps : List containers   2. Docker ps -a : To list the containers which was previously running   3. Docker rm: Remove one or more containers      * 1. Docker rmi: Remove one or more images      1. Docker pull : Pull an image or a repository from a registry      1. Docker run : Run a command in a new container      1. Docker info: Display system-wide information 2. Docker history: Show the history of an image 3. Docker events: Get real time events from the server      1. Docker create: Create a new container      1. Docker context: Manage contexts 2. Docker config: Manage Docker configs      1. Docker Search: Search the Docker Hub for images     **Terminate Resource**  **Select the instance🡪Click on Terminate instance.** |