*DEPARTMENT OF INFORMATION TECHNOLOGY* Experiment No4

|  |  |  |
| --- | --- | --- |
| **Semester** | Semester VIII | |
| **Subject** | DevOps Lab | |
| **Subject Professor In- charge** | Prof. Yash Shah | |
| **Laboratory** | L11B | |
| **Student Name** | Ashwini Jadhav | |
| **Roll Number** | 17101B0038 | |
| **Grade and Subject Teacher’s Signature** |  |  |

|  |  |  |
| --- | --- | --- |
| **Experiment**  **Number** | 4 | |
| **Experiment**  **Title** | To use Docker Hub for saving container repositories | |
| **Resources / Apparatus Required** | Hardware:  Compatible Computer System | Kali Linux, Docker, Docker Hub |
| **Objectives** | Explore and implement DockerHub repositories | |
| **Theory** | **What is containerization?**  It involves encapsulating or packaging up software code and all its dependencies so that it can run uniformly and consistently on any infrastructure. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.  Container images become containers at runtime and in the case of Docker containers - images become containers when they run on [Docker Engine.](https://www.docker.com/products/container-runtime) Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.  **Need of containerization:**   * Containerization reduces wasted resources because each container only holds the application and related binaries or libraries. * By allowing more containers in the environment without the need for more servers, containerization increases scalability anywhere from 10 to 100 times that of traditional VM environments. | |

|  |  |
| --- | --- |
|  | * The ability to rapidly spin up new containers also increases the capacity to handle website traffic load seamlessly. * Using containerization helps your cloud environment efficiency; by deploying multiple containerized applications on to a single cloud instance, you get much closer to achieving 100% utilization. * Improved security by isolating applications from the host system and from each other. * Faster app start-up and easier scaling. * Flexibility to work on [virtualized](https://www.ibm.com/cloud/learn/virtualization-a-complete-guide) infrastructures or on bare metal servers * Easier management since install, upgrade, and rollback processes are built into the [Kubernetes](https://www.ibm.com/cloud/learn/kubernetes) platform.   **How to send repositories using Docker hub:**  To login into Docker Hub:  **docker login**  Push the repository:  **docker push *DockerHub\_ID/repository\_name***  Pull the repository:  **docker pull *DockerHub\_ID/repository\_name*** |
| **Output** | **Login to Docker Hub via terminal:**  Docker41  **Committing the container:**  Docker42  **List of images:**  Docker43  **Push repository to Docker Hub:**  Docker44 |

|  |  |
| --- | --- |
|  | **Pulling a repository from DockerHub:**  Docker45  **Creating container from pulled image:**  Docker46  **Port forwarding:**  Docker47  **Starting apache server within container:**  Docker49  **Result of the above:**  Docker48  Docker410 |

|  |  |
| --- | --- |
|  | **List of images and containers and Stopping and removing containers:**  Docker411 |
| **Conclusion** | Thus, we have implemented containerization using Docker and saved repository  to Docker Hub. |