*DEPARTMENT OF INFORMATION TECHNOLOGY* Experiment No3

|  |  |  |
| --- | --- | --- |
| **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** | 3 | |
| **Experiment Title** | To install and configure Docker for creating Containers | |
| **Resources / Apparatus Required** | Hardware:  Compatible Computer System | Kali Linux, Docker |
| **Objectives** | Explore and implement containerization. | |
| **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 carry out containerization using Docker:**  Create a docker image by pulling from docker:  **docker pull ubuntu** (different OS have their own image)  Build a container by running an image:  **docker run -it -d *image\_name***  (-it, makes container interactive,  -d, stands for daemon, container will work in background)  Developer has to enter container to put project files. In order to do so, we execute the container:  **docker exec -it *container\_id* bash**  Container id can be obtained from:  **docker ps**  Add project files within container. To exit container:  **exit**  Developer can send only image to tester. To create an image:  **docker commit *container\_id name*** |
| **Output** | **Creating docker image:**  Docker1  **Building a container:**  Docker2 |

|  |  |
| --- | --- |
|  | **Obtaining container ID and entering container:**  Docker3  **Only Basic OS in container:**  Docker4  **Updating apt within container:**  Docker5  **Installing text editor ‘nano’ within container:**  Docker6 |

|  |  |
| --- | --- |
|  | **Installing apache within container:**  Docker7  **Creating ‘sample project’ within container:**  Docker8  **Sample Index file:**  Docker9  **To go out of the container:**  Docker10  **Creating an image of the container:**  Docker11 |
| **Conclusion** | Thus, we have implemented containerization using Docker. |