INTRODUCTION:

**Docker** is one of the revolutionary technologies based on the containerization technology created by **Solomon Hykes** officially released on **7th March 2013.**

It is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package. By doing so, thanks to the container, the developer can rest assured that the application will run on any other Linux machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code.

Docker is designed to benefit both developers and system administrators, making it a part of many DevOps (developers + operations) toolchains. For developers, it means that they can focus on writing code without worrying about the system that it will ultimately be running on. It also allows them to get a head start by using one of thousands of programs already designed to run in a Docker container as a part of their application. For operations staff, Docker gives flexibility and potentially reduces the number of systems needed because of its small footprint and lower overhead.

PROJECT DESCRIPTION:

This is the final Docker project made with Docker Compose based on Red hat Linux-8 operating system as host. My project uses “Infrastructure as a Code” i.e. “Docker Compose” to launch a full working Joomla setup in one go.

What is Joomla?

Joomla is a free and open-source content management system (CMS) for publishing web content. It is built on a model–view–controller web application framework that can be used independently of the CMS. Joomla is written in PHP, uses object-oriented programming (OOP) techniques and software design patterns, stores data in a MySQL, MS SQL, or PostgreSQL database, and includes features such as page caching, RSS feeds, printable versions of pages, news flashes, blogs, search, and support for language internationalization.



INITIAL SETUP REQUIREMENTS:

Following are the steps to setup Docker in your host Red hat Linux system:

Pre-requisites:

1. First of all, configure your “yum” with your custom repo which allow you to install docker in your host system.
2. Install Docker Community Edition with the command “yum install docker-ce –nobest”. This will install Docker in your base Red Hat system.
3. Install Docker Compose by running this command:

“curl -L "https://github.com/docker/compose/releases/download/1.25.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose”

1. Now, run “chmod +x /usr/local/bin/docker-compose” to get the “docker-compose” command.

Pulling docker images from Docker Hub:

1. Pulling MySQL Image:

Use “docker pull mysql:5.7” command to install MySQL in your system. This will create MySQL database in your system to host Joomla Web App Environment in your base Red Hat system.

1. Pulling Joomla Image:

Use “docker pull joomla:latest” command to install Joomla in your system. This will install Joomla in your base Red hat system.

CREATING THE ENTIRE INFRASTRUCTURE:

1. First of all with “mkdir”, create a directory with any name, for example: “mkdir/myjoomlacompose” where we create our workspace to store our .yaml file.
2. Now go to the directory (cd /myjoomlacompose) and create .yaml file with the command “vim docker-compose.yml”
3. Write the yaml code to lunch your entire Joomla setup in one go.
4. Save the .yml file and exit with (: wq) command.
5. Now, go to your workspace directory by (cd /myjoomlacompose) command.
6. Run “docker-compose up” to launch your entire setup.
7. Your Joomla web environment is now successfully created hosted on your base Red Hat system.
8. To Stop the service run “docker-compose stop”.