Why Containerization/Dockerization applications is essential

Background and Introduction

Background

Before Containerization, most companies and tech enthusiast alike used and still use Virtual Machines. A Software like oracle virtual box would allow the use to create an application or a database by downloading an OS image, it can be different flavors of Linux (ubuntu, centos etc.), downloading relevant application related files, it can be an oracle database software for database creation or framework related files and spin up a "virtual computer" using the downloaded OS image and relevant application files via oracle virtual box, CPU and other hardware is shared with the computer the VM is running on. This process made software development and infrastructure manage less hassle free as it allowed users to speed up development process without needed to set up an entire infrastructure. I remember using VM's when I worked in India for a period of 1 year. Since our team was in multiple parts of India and our clients were both in India and UK, we very given access to VMs for our work. While the process was smooth sailing for the most part, OS failures and lag in virtual box response time made the experience unpleasant for users, especially when the project or application had to scale up. And downloading the OS image along with software was sub-optimal for storage because an Ubuntu Linux image was 10 GB or above in Size and software like oracle database 12c or 19c was roughly 20 GB. Installing a simple OS with a database software would cost 30 GB in storage. A more optimal solution was need for quick software deployment beyond the VMs and the standard process of local development and git. The solution to his problem came with Containerization be it using Docker or Kubernetes. It gave users the ability to quick spin up a packaged application using Docker file or Docker Compose. I would be using docker compose and dockerfile for data storage for my IWS project.

Introduction

Given the rapidly changing nature of Application and Software Development, it has become increasingly important to have flexible software infrastructure that can function even during times of failure. "High Availability" is highly desired in the Technology world today. Containerization or Dockerization does exactly that. Dockerization would involve combining all the necessary functionality for your application into a docker container, and quickly spin up application. I need to use Docker at least partially for creating the data storage infrastructure for the IWS 1 project.

Importance

Importance of Containerization in my view can be divided into 2 aspects 1. Size 2. Availability

1. **Size**
   The overall size of a docker image is only a few mb, and even after adding additional software layers, it would not for the most part cross 1 Gb, the limit for a docker container, if needed is 10 GB. Compared to that a VM with linux OS and some software like oracle

12c would be at least 30 GB. We can add more layers to a docker container while consuming less space when compared to a VM.

2. **Availability**

   If the operating system or the environment the docker container is running on has been comprised or is inaccessible, the container can simply be connected to another network using add-network, and this high availability means your application is protected against system failures. Connecting to a container build on linux to another linux system is easy but connecting a linux to windows can only happen via windows subsystem for linux.

Objectives

Given the advantages posed by Docker, I have 2 objectives with my analysis

1. Use Docker Compose for data storage for the IWS Project, analyse its performance using the 2 factors mentioned above i.e. Size and Availability.
2. Test Docker's High Availability in its entirety by using Docker Swarm, which is docker container management tool, that let's users build multiple containers, each of which is a part of the application (database, server code, app side), on different servers with one manager node and multiple worker nodes. And all the nodes/containers communicate with each other to run the application. Since I only aim to demo the functionality of docker swarm, I will be using a sample application (something like a simple one-page node js app)

References

1. **Author** - Srinath Reddy Meadusani, **Research Paper** - Virtualization Using Docker Containers: For Reproducible Environments and Containerized, Paper Link
2. Docker Hub Documentation Website - https://hub.docker.com/
3. **Author** - Vincenzo Ferme, **Research Paper** - Using Docker Containers to Improve Reproducibility in Software and Web Engineering Research, Paper Link