Ethan Willett
Assignment 7
DevOps Software Development
7/14/2023

Vagrant Vs. Docker

I have not had much personal experience with Docker outside of this class and have never used Vagrant before. To that extent comparing the two tools will be mostly restricted to what I have experienced thus far this semester, with some of my personal Docker experience to be considered.

Vagrant provides a portable development environment, enabling developers to configure and share environments across different machines. This differs from Docker in the way Vagrant provides the development environments. Vagrant provides more portability by allowing any system, capable of running a VM application (such as VirtualBox or VMWare), to deploy Vagrant “Boxes”. Docker, on the other hand, only allows host systems to run docker containers that the container was specifically created for. This means a Docker container might not be able to run on a Windows, Linux, or MacOS host depending on that particular host system’s setup.

The way Containers and Boxes are created highlight their specific intents. Vagrant is designed to provide portable development environments while Docker packages applications and dependencies into lightweight and portable containers. This provides specific pros and cons for each tool and while they can be considered interchangeable, their benefits are focused on specific use cases.

Vagrant utilizes full virtual machines, consuming more system resources, requiring booting the VMs, meaning more system resource requirements and longer boot times. Docker provides lightweight and minimal overhead containers, which boot quickly and perform better than VMs, especially when running multiple. Docker also provides tools and systems such as Compose and Swarm for configuring and running multiple containers in an orchestrated deployment. Vagrant does not have easy or specific use solutions for easy orchestration and management of multiple boxes.

Given the expected use cases for each tool, I would say that I am happy to have learned about Vagrant. I think that a lot of people expect Docker containers to work in the way Vagrant does, a simplified VM, instead of an ultra lightweight and genericised application deployment system. I would expect to use Vagrant to try out different development environments if one was provided. But due to my history with Docker and the complexity around creating Vagrant Boxes I do not think I would choose creating a Box over a Container for application deployment.