\*\*Question 3: Containers\*\*

When is it appropriate to use containers in cloud deployments, and what are the security benefits of doing so?

Often, when code is developed in a certain environment and then transferred to where it needs to be, bugs and errors occur. It is the containerization that can/will eliminate this problem. It can be achieved by modernizing application in a cloud deployment, making it fast as well as securely performed. Containers are basically lightweight VM, using fewer resources, accomplished by sharing resources in common with other containers.

There are important benefits of using containers. One of the greatest benefits is the portability, which is achieved through speeds development (so called “written once and run anywhere”). They also do not need much start up time because their capacity is small (lightweight). Further, they reduce licensing and server costs through their higher server efficiencies. Another advantage is the fact the container is isolated and its operation is not dependant on any other container. Containers work equally well on Windows and Linux OS. They are easy to manage because the container platform can be automated (including installation, scaling, and management of workloads and services). The security benefit is due to isolation of application, as this prevents the invasion of malicious code from affecting other containers and host computer.

In my project I installed the ansible container in the provisioner (the gateway jump box) so that I could connect without manual logging and issue commands to all VMs within the networks. I created a stateless container so the data is moved to a central location (as opposite a stateful container that cannot be safely destroyed and replaced if no other container contains that particular data). I have also installed a docker containers to configure Web-1, Web-2 and Web-3, to accommodate testing and training. Subsequently I installed a container in elk machine to collect and process data from 3 web VMs and store them in a centralized location. These data then can be analyzed.

This greatly reduces the probability of human error. Because it is automatic, a large number of identical machines can be configured at the same time. The main security benefit ensures that if one container is attacked, it can be killed and regenerated without the entire website going down.

The container installation process in jump box VM start by installing a Docker container by running apt commands, ‘sudo apt update’, and ‘sudo apt install docker.io’. To verify, I ran ‘sudo systemctl status docker’. After that I used the Docker command to install the ansible containers. By running ‘sudo docker pull cybersecurity/ansible’. To select the name of the container I ran the ‘sudo docker container list – a’. Then I ran ‘sudo docker start “name”’ with the correct container name; lastly, I ran ‘sudo docker attach “name”’ to run the container (to verify).

In the absence of containers, the other option is using the Virtual Machine. VM is an operating system that shares the physical resources of one server. VM is composed of several layers. It has its own infrastructure and is self contained. VM are definitely cheaper to run than multiple physical machines, because of utilizing one physical resource. Also, because of that, they manage efficiently all the virtual environments. There are also disadvantages to using VM. They take up a lot of system resources (lot of Ram and CPU cycles, “heavy”, long time to download and deploy, lot of “wasted space”). Relocating an app is complicated as app must be migrated together with the OS. Planning and distribution can be difficult.