Use of Containers in Cloud Security

It is appropriate to use containers in cloud deployments to ensure a stable system architecture. This promotes an efficient operation where all virtual machines are running the same current version across the network. The security benefit of this includes modules that are relatively easy to update, with a redundancy making it more difficult for an intruder to target.

In the ELK Stack Project, containers were seeded in the Jump Box Provisioner, Web VMs 1 and 2, and the ELK server. A Docker container in the Provisioner was created to run Ansible, which automates software installations to other Virtual Machines in the network. Containers in Web-1 and 2 were used by Ansible to deploy the DVWA application, and the container in ELK received the configuration for Web server log monitoring and Kibana reports.

This method is useful when the same configuration needs to be installed on several different machines, such as Web servers in the same availability zone. Continued operation is assured in that, in the event of a Virtual Machine outage, one replicated unit can failover to another. Looking ahead, for machines in the same cohort, Ansible offers a platform for relatively easy redeployment, maintaining configurations or applying updates as well.

In the ELK Project, SSH keys were established to maintain connections with the appropriate Virtual Machines. For example, the Administrator Workstation with a public IP was paired with the Jump Box, and the SSH key within the Jump Box Docker container instance was paired with each VM in the deployment (Web servers and the ELK box). The Jump VM was first contacted to attach the initial Docker container, and Ansible was run within that instance to push DVWA software to the Web servers, and an ELK image to its corresponding VM.

Once completed, the container image could be verified by connecting to each machine via the Jump Box and inspecting it with ‘sudo docker ps’, or to verify live production, the http://(public IP)/setup.php of the available Web server, or the http://(public IP):5601/app/kibana#/home of ELK.

Without the Ansible automated deployment, the software configuration would have to be performed manually on each Virtual Machine. And while closer attention to the build might result in better accuracy and fault awareness on one machine, the lack of a single repeatable configuration would increase the likelihood of error when rolling out manual updates to larger networks.