**Activity File: Interview Questions Domain: Cloud Security Question 3: Containers**

(1) It is a good idea to use containers when you are working on a larger or more complex project. For example, when you are trying to reduce the load of responsibility and resources on a machine, assigning each docker a specific task or application can solve this problem. Containers work by isolating the software from its environment and ensure standardization by separating application dependencies from infrastructure. Containers are designed to do one thing very well, and allow elasticity and scalability. “Rules” (or playbooks) run on the containers to do these specific things. Some benefits to this are that you are not paying for resources when they are not being used, and you can dynamically build resources as well. Containers also offer speed of use, as they spin up faster than servers (seconds compared to minutes), all while adding layers of security and accessibility controls. Docker containers are a good idea for use as version control on your application’s entire operating system. They are also ideal for team distribution and collaboration efforts, or when your app needs to go through multiple phases of development, for example.

(1) Since applications run independently in separate containers, each will have their own level of security. This feature of app isolation helps share additional features without any risk factor. In line with the Principle of Least Privilege, access to containers can be provided to internal or external users without granting access to any more than is necessary.

(2) In Project 1, I created an ELK stack that allows us to automate the process of monitoring the performance of multiple virtual machines in one database. Ansible was used to create one container on the Jump Box Provisioner. Three Ansible playbooks were created within that container. The first playbook created and mapped a new ELK container to the ELK server. The next two playbooks installed FileBeat and MetricBeat, which help collect file and system data on the web server VMs, which are sent to the ELK server. Based on the playbook instructions, as well as the NSG security rules, the ports and allowed IP is such that I can access the log data using Kibana through my internet browser.

(3) This virtual network setup, including containers, was an appropriate use of containers in this instance because it is being used to monitor a network with multiple web servers that has multiple layers of security. With regard to security in the Azure portal, there were NSG security rules set up to allow only certain traffic/ports. The machines were all set up to use public SSH Keys instead of passwords, removing the ability to brute force attack the passwords. The use of containers means app isolation, so potential bad actors do not have access to an entire machine(s).

(4) Getting containers set up within the Jump Box. Docker.io was installed, and Ansible was pulled in. The first container is then created on the Jump Box. The container's SSH Public Key was then added to VM’s within the Azure portal. Next, the container’s hosts file was updated to reflect the webservers IPs, and the config file was updated to include the webservers username remote user).

(4) The initial container had a randomly generated name and was set up within the Jump Box, however the second container was specific to the ELK server. A playbook was created to configure the ELK server. This playbook had similarities to the previous playbook, however one section of YAML code controlled the container source, name, and version (sebp/elk:761). To confirm that the correct container was installed, the playbook was run and the Play Recap showed no errors. Another way to check if things were working correctly would be if the commands to “start” and “attach” the container would change the command line to end up in the root command for that container.

(5) Without these automated actions, I would have run commands like syslog or auditd, which would have been a manual process that would have collected more data than I would want. Use of Elasticsearch, Logstash, FileBeat, and MetricBeat help streamline that process by collecting specific data and formatting it in a consistent manner. The data would also be collected within each virtual machine (versus being sent to a centralized location), which isn’t ideal because the tasks would have to run in each independent machine. This could also stress the machine, which has limited resources. Containers help carve out specific resources for specific tasks, making the process more efficient. Each machine and container have their own security rules, creating more layers of security if one item were compromised.

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

1. Restate the problem
2. Provide a Concrete Example Scenario
   * In Project 1, when did you use containers?

In this ELK Stack Project, we used a total of 2 different containers. We were able to install Ansible within our Jump Box Provisioner, which is an automation tool that is designed to automate playbooks used within the containers. We used the first container to install our second container, through the ELK Server as the host. The second container was then used to run two different YAML Ansible playbooks to install different logging data applications.

* + What did you use containers for?

We used containers to install an elk container on our ELK server, and then configured and deployed Ansible YAML playbooks for Linux Filebeat and Metricbeat data collection. This type of container allows for ease of data collection, logging and analysis via use of Elasticsearch, Logstash and Kibana.

Filebeat will allow us to collect file system data, and Metricbeat collects machine metrics from the system and services running on our server

1. Explain the Solution Requirements  
   * Why was this an appropriate use for containers?
   * What security benefits did you expect from using containers?
2. Explain the Solution Details  
   * In Project 1, how did you configure VMs to be able to run containers?
   * How did you select and install the correct container?

We were able to download docker.io onto our Jumpbox and determine that a container was installed by running the command, sudo systemctl status docker. We then ran sudo docker start <container name> to start the container and sudo docker attach <container name> to go into and operate in the container.

* + How did you verify that it was running correctly?

After running the command, sudo docker attach <container name>, the user then changed from azadmin@RedJumpbox-Provisioner to ‘root……’ This is a sign of success because it shows that you are now operating as the root user for that container.

1. Identify Advantages/Disadvantages of the Solution  
   * How would you have achieved the same thing without containers?

We would have been able to achieve the same result by creating our virtual network and having the different web servers on the network run the ansible playbooks.

* + What are the advantages to doing it without containers?

Set up of a machine would be faster, and debugging would be less difficult. Performance can also be better due to less limitations processing large amounts of data.

* + What are the disadvantages?

There are several disadvantages to running the ELK stack this way, versus within the containers. The first and most important disadvantage is this is a far less secure way to manage, process and organize data being evaluated. If any bad actors were trying to access the logged data, they would have direct access to the web servers and virtual network directly. Another disadvantage would be that the network and web servers themselves are not designed to house a lot of data and would quickly run out of room and be overwhelmed. Not all applications would benefit from running within a container, only those designed to run as microservices. Another disadvantage is that it is much more difficult to run complicated scripts, especially through automation.

There is a lot of wasted space in doing this type of project without using containers as only a few files on the entire disk would be relevant to running the ELK.