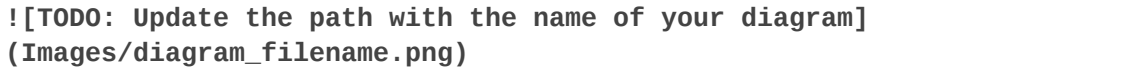
1.1 GITHUB REPO

Automated ELK Stack Deployment

The files in this repository were used to configure the network depicted below.



These files have been tested and used to generate a live ELK deployment on Azure. They can be used to either recreate the entire deployment pictured above. Alternatively, select portions of the __YAML___ file may be used to install only certain pieces of it, such as Filebeat.

  - _TODO: elk_playbook.yml

This document contains the following details:
- Description of the Topology
- Access Policies
- ELK Configuration
  - Beats in Use
  - Machines Being Monitored
- How to Use the Ansible Build

Description of the Topology

The main purpose of this network is to expose a load-balanced and monitored instance of DVWA, the D*mn Vulnerable Web Application.

Load balancing ensures that the application will be highly _effective___, in addition to restricting traffic __Access_ to the network.
- _TODO: What aspect of security do load balancers protect? They prevent unwanted or unauthorized traffic from reaching the application.

What is the advantage of a jump box?_ They add a security layer to the web servers preventing them from being exposed to the public.

Integrating an ELK server allows users to easily monitor the vulnerable VMs for changes to the __configuration files___ and system _files___.
- _TODO: What does Filebeat watch for?_They watch for Log files or log events
- _TODO: What does Metricbeat record?_They record Metrics from on going services on the server

The configuration details of each machine may be found below.
_Note: Use the [Markdown Table Generator]

(http://www.tablesgenerator.com/markdown_tables) to add/remove values from the table_.

Name	Function	IP Address	Operating System
Jump Box	Gateway	10.0.0.1	Linux
Web-1	Webserver	10.0.0.11	Linux
Web-2	Webserver	10.0.0.12	Linux
Web-3	Webserver	10.0.0.13	Linux
VM-ELK	Webserver	10.1.0.4	Linux

Access Policies

The machines on the internal network are not exposed to the public Internet.

Only the _Jump-box-Provisioner____ machine can accept connections from the Internet. Access to this machine is only allowed from the following IP addresses:
- _TODO: 172.56.11.249

Machines within the network can only be accessed by Jumpbox_____.
- _TODO: Which machine did you allow to access your ELK VM? My personal computer
 What was its IP address?_172.56.11.249

A summary of the access policies in place can be found in the table below.

Name	Publicly Accessible	Allowed IP Addresses
Jump Box	Yes	10.0.0.1 10.0.0.2
Web-1	No	172.56.11.249
Web-2	No	172.56.11.249
Web-3	No	,, ,, ,, ,,
VM_ELK	Yes(http)	,, ,, ,, ,,

Elk Configuration

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because...
- _TODO: What is the main advantage of automating configuration with Ansible?_ It is flexible because it allows changes to be made within any of the VMs associated with it.

The playbook implements the following tasks:
- _TODO: In 3-5 bullets, explain the steps of the ELK installation play. . 1) Install Docker.io 2)Install python3-pip 3)Install Docker Python Module 4) Download and launch a Docker web container 5)Download and launch a

docker web container.
- ...

The following screenshot displays the result of running `docker ps` after successfully configuring the ELK instance.

![TODO: Update the path with the name of your screenshot of docker ps output](Images/docker_ps.png)

Target Machines & Beats
This ELK server is configured to monitor the following machines:
- _TODO: List the IP addresses of the machines you are monitoring_
10.0.0.1   10.0.0.11   10.0.0.12   10.0.0.13 10.0.0.4

We have installed the following Beats on these machines:
- _TODO: Specify which Beats you successfully installed_
Filebeat and Metricbeat

These Beats allow us to collect the following information from each machine:
- _TODO: In 1-2 sentences, explain what kind of data each beat collects, and provide 1 example of what you expect to see. E.g., `Winlogbeat` collects Windows logs, which we use to track user logon events, etc._

Filebeat monitors log files and log events. Example: Inputs and harvesters. While Metricbeat looks out for any information in the file system which has been manipulated.

Using the Playbook
In order to use the playbook, you will need to have an Ansible control node already configured. Assuming you have such a control node provisioned:

SSH into the control node and follow the steps below:
- Copy the _ansible configuration____ file to __run playbooks___.

- Update the _ansible host____ file to include...

- Run the playbook, and navigate to _Jumpbox___ to check that the installation worked as expected.

TODO: Answer the following questions to fill in the blanks:
- _Which file is the playbook? The elk-playbook.yml

 Where do you copy it?_

- _Which file do you update to make Ansible run the playbook on a specific machine? elk-playbook.yml file

How do I specify which machine to install the ELK server on versus which to install Filebeat on?_ By using the IPs of the respective servers.

- _Which URL do you navigate to in order to check that the ELK server is running?
http://[ElkProject1VM-ip]:5601/app/kibana

As a **Bonus, provide the specific commands the user will need to run to download the playbook, update the files, etc.**
ansible-playbook elk-playbook.yml

1.2 Ansible Playbook and Configuration files

Elk-playbook.yml

```
- name: Config ELK with Docker
  hosts: elk
  become: true
  tasks:
  - name: Install docker.io
    apt:
      update_cache: yes
      name: docker.io
      state: present

  - name: Install pip3
    apt:
      force_apt_get: yes
      name: python3-pip
      state: present

  - name: Install Docker Python Module
    pip:
      name: docker
      state: present

  - name: Download and launch a Docker web container
    docker_container:
      name: elk
      image: sebp/elk:761
      state: started
      restart_policy: always
      published_ports:
        - 5601:5601
```

```
        - 9200:9200
        - 5044:5044

  - name: Configure elk VM to use more memory
    sysctl:
      name: vm.max_map_count
      value: "262144"
      state: present
      reload: yes

  - name: Enable Docker service
    systemd:
      name: docker
```

Ansible Hosts

```
[webservers]
10.0.0.11 ansible_python_interpreter=/usr/bin/python3
10.0.0.12 ansible_python_interpreter=/usr/bin/python3
10.0.0.13 ansible_python_interpreter=/usr/bin/python3

[elk]
10.1.0.4  ansible_python_interpreter=/usr/bin/python3
```

Filebeat-Configuration

```
# https://www.elastic.co/guide/en/beats/filebeat/index.html
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml

#==========================  Modules configuration
============================
filebeat.modules:
- module: elasticsearch
  # Server log
  server:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:

  gc:
    enabled: true
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:
```

```
  audit:
    enabled: true
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:

  slowlog:
    enabled: true
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:

  deprecation:
    enabled: true
- module: haproxy
  # All logs
  log:
    enabled: true

    # Set which input to use between syslog (default) or file.
    #var.input:
#------------------------------- Kafka Module
-------------------------------
- module: kafka
  log:
    enabled: true

    # Set custom paths for Kafka. If left empty,
    # Filebeat will look under /opt.
    #var.kafka_home:

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:

#------------------------------- Kibana Module
-------------------------------
- module: kibana
  # All logs
  log:
    enabled: true

- module: nats
  # All logs
  log:
    enabled: true
#---------------------------- Google Santa Module
----------------------------
- module: santa
```

```
  log:
    enabled: true
filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# Type of the files. Based on this the way the file is read is decided.
# The different types cannot be mixed in one input
#
# Possible options are:
# * log: Reads every line of the log file (default)
# * stdin: Reads the standard in

#------------------------------ Log input --------------------------------
- type: log

  # Change to true to enable this input configuration.
  enabled: false

  # Paths that should be crawled and fetched. Glob based paths.
  # To fetch all ".log" files from a specific level of subdirectories
  # /var/log/*/*.log can be used.
  # For each file found under this path, a harvester is started.
  # Make sure not file is defined twice as this can lead to unexpected
behaviour.
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*

  # Configure the file encoding
#------------------------- Elasticsearch output
------------------------------
output.elasticsearch:

  # Boolean flag to enable or disable the output module.
  #enabled: true

  # Array of hosts to connect to.
  # Scheme and port can be left out and will be set to the default (http
and 9200)
  # In case you specify and additional path, the scheme is required:
http://localhost:9200/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:9200
  hosts: ["10.1.0.4:9200"]
  username: "elastic"
  password: "changeme"
  setup.template.settings:
  setup.kibana:
```

```
  host: "10.1.0.4:5601"
# files.
logging.to_files: true
logging.files:
```

Ansible-playbook.yml

```
  - name: Config Web VM with Docker
    hosts: webservers
    become: true
    tasks:

    - name: docker.io
      apt:
        update_cache: yes
        name: docker.io
        state: present

    - name: Install pip3
      apt:
        name: python3-pip
        state: present

    - name: Install Python Docker Module
      pip:
        name: docker
        state: present

    - name: Download and launch a docker web container
      docker_container:
        name: dvwa
        image: cyberxsecurity/dvwa
        state: started
        restart_policy: always
        published_ports: 80:80
```

Filebeat-playbook.yml

```
 - name: Installing and Launching Filebeat
    hosts: webservers
    become: yes
    tasks:

    - name: Download filebeat .deb file
      command: curl -L -O
https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.0-
amd64.deb
```

```
    - name: Install filebeat.deb
      command: sudo dpkg -i filebeat-7.4.0-amd64.deb


    - name: Drop in filebeat.yml
      copy:
        src: /etc/ansible/files/filebeat-config.yml
        dest: /etc/filebeat/filebeat.yml


    - name: Enable and configure system module
      command: filebeat modules enable system


    - name: Setup filebeat
      command: filebeat setup


    - name: Start filebeat service
      command: service filebeat start


    - name: Enable service filebeat on boot
      systemd:
        name: filebeat
        enabled: yes
```

Metricbeat-configuration

```
# You can find the full configuration reference here:
# https://www.elastic.co/guide/en/beats/metricbeat/index.html
#========================  Modules configuration
============================
metricbeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml
  # Set to true to enable config reloading
  reload.enabled: false
  # Period on which files under path should be checked for changes
  #reload.period: 10s
#==================== Elasticsearch template setting
==========================
setup.template.settings:
  index.number_of_shards: 1
  index.codec: best_compression
  #_source.enabled: false
#================================ General
```

```
====================================
# The name of the shipper that publishes the network data. It can be used
to group
# all the transactions sent by a single shipper in the web interface.
#name:
# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]
# Optional fields that you can specify to add additional information to
the
# output.
#fields:
#  env: staging
#============================== Dashboards
=====================================
# These settings control loading the sample dashboards to the Kibana
index. Loading
# the dashboards is disabled by default and can be enabled either by
setting the
# options here or by using the `setup` command.
#setup.dashboards.enabled: false
# The URL from where to download the dashboards archive. By default this
URL
# has a value which is computed based on the Beat name and version. For
released
# versions, this URL points to the dashboard archive on the
artifacts.elastic.co
# website.
#setup.dashboards.url:
#============================== Kibana
=====================================
# Starting with Beats version 6.0.0, the dashboards are loaded via the
Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:
  host: "10.1.0.4:5601"
  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http
and 5601)
  # In case you specify and additional path, the scheme is required:
http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  #host: "localhost:5601"
  # Kibana Space ID
  # ID of the Kibana Space into which the dashboards should be loaded. By
default,
  # the Default Space will be used.
  #space.id:
#============================== Elastic Cloud
```

```
==================================
# These settings simplify using Metricbeat with the Elastic Cloud
(https://cloud.elastic.co/).
# The cloud.id setting overwrites the `output.elasticsearch.hosts` and
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:
# The cloud.auth setting overwrites the `output.elasticsearch.username`
and
# `output.elasticsearch.password` settings. The format is `<user>:<pass>`.
#cloud.auth:
#================================ Outputs
===================================
# Configure what output to use when sending the data collected by the
beat.
#------------------------- Elasticsearch output
------------------------------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["10.1.0.4:9200"]
  username: "elastic"
  password: "changeme"
  # Optional protocol and basic auth credentials.
  #protocol: "https"
  #username: "elastic"
  #password: "changeme"
#---------------------------- Logstash output
-------------------------------
#output.logstash:
  # The Logstash hosts
  #hosts: ["localhost:5044"]
  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]
  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"
  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"
#================================ Processors
===================================
# Configure processors to enhance or manipulate events generated by the
beat.
processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~
#================================ Logging
===================================
# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
```

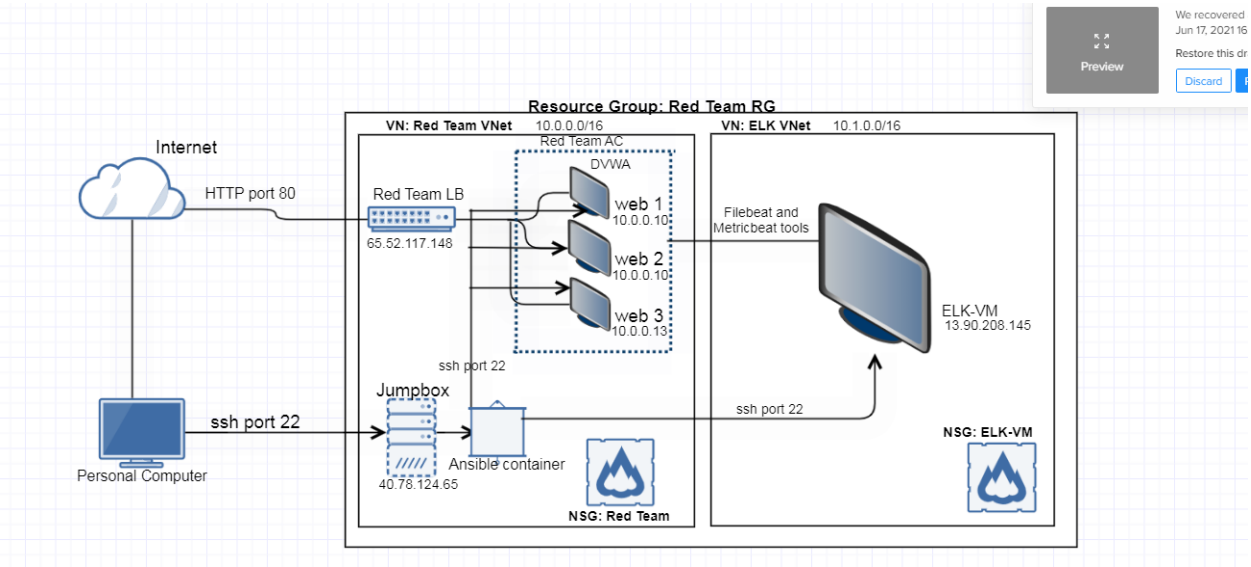
```
#logging.level: debug
# At debug level, you can selectively enable logging only for some
components.
# To enable all selectors use ["*"]. Examples of other selectors are
"beat",
# "publish", "service".
#logging.selectors: ["*"]
#============================== X-Pack Monitoring
===============================
# metricbeat can export internal metrics to a central Elasticsearch
monitoring
# cluster.  This requires xpack monitoring to be enabled in Elasticsearch.
The
# reporting is disabled by default.
# Set to true to enable the monitoring reporter.
#monitoring.enabled: false
# Sets the UUID of the Elasticsearch cluster under which monitoring data
for this
# Metricbeat instances will appear in the Stack Monitoring UI. If
output.elasticsearch
# is enabled, the UUID is derived from the Elasticsearch cluster
referenced by output.elasticsearch.
```

Metricbeat-playbook.yml

```
 - name: Installing and Launching metricbeat
    hosts: webservers
    become: yes
    tasks:
    - name: Download metricbeat .deb file
      command: curl -L -O
https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.4>


    - name: Install metricbeat .deb
      command: dpkg -i metricbeat-7.4.0-amd64.deb


    - name: Drop in metricbeat.yml
      copy:
        src: /etc/ansible/files/metricbeat-config.yml
        dest: /etc/metricbeat/metricbeat.yml


    - name: Enable and configure system module
      command: metricbeat modules enable docker


    - name: Metricbeat setup
      command: metricbeat setup
```

```
- name: Start metricbeat service
  command: service metricbeat start


- name: Enable service metricbeat on boot
  systemd:
    name: metricbeat
    enabled: yes
```

1.3 Network Diagram



2.1 Project 1 Interview Activity

Domain: Cloud Security

Question 1: Cloud Access Control

How would you control access to a cloud network?

1. Restate the Problem

- How would I restrict particular areas in a cloud network to certain sets of rules?

2. Provide a Concrete Example Scenario

- In Project 1, did you deploy an on-premises or cloud network?
  I deployed a cloud network
- Did you have to configure access controls to this network?
  Yes, I did.
- What kinds of access controls did you configure, and why were they necessary?
  I created a security group and assigned an inbound security rule allowing ssh to port 22 from the Red Team virtual network. It was necessary for the DVWA VMs to gain access to ELK through port 22.
- How do these details relate to the interview question?
  They relate to the question because they show how access controls will help an individual to protect his/her cloud-based infrastructure by assuring that only authorized persons can work with critical files and no one else.

3. Explain the Solution Requirements

- In Project 1, what kinds of access controls did you have to implement? Consider:
  I deployed NSG's around the ELK VNet and VMs. The local firewalls deployed around the VMs were embedded in the NSG for the VNet.
  I also allowed TCP protocols as part of my security rules.

- What did each access control achieve, and why was this restriction necessary for the project?
  - The NSGs created rules that allow or deny inbound network traffic. It was necessary for the projects because it helped to specify the source, destination, port, and protocols according to my preference.
  - Local firewalls which were embedded in the NSGs helped to eliminate the occurrence of unwanted network communications and this allowed all legitimate communication for the project to flow freely.
  - The TCP was used to verify the integrity of data traveling between two endpoints which was necessary for access from the jump box to the DVWAs

4. Explain the Solution Details

- Which rules do you set for each NSG in the network?

I allowed inbound traffic to the load balancers by setting the source IP address to 10.0.0.10 and the destination to the virtual Network. The port was 80 and protocol, TCP.

I allowed ssh from my personal IP to the virtual networks by setting the source IP address to 172.58.120.195 and the destination to virtual networks. The port was set to 22 and protocol set to any.

I allowed internal ssh from the jumpbox by setting the source IP address to 10.0.0.10 and the destination to the virtual Network. The port was 22 and protocol set
to any.

I allowed Elk to be accessible by my PC by setting the source IP address to 172.58.121.188, the destination to the virtual network. The port was 5601 and protocol was set to any.

- How does access to the jump box work?

I accessd the jump box by using ssh and a private key with the jumpbox public IP address.

- How does access from the jump box to the web servers work?

It works by using ssh and a private key with the webserver's private address.

5. Identify Advantages/Disadvantages of the Solution

- Does your solution scale?

Yes it does

- Is there a better solution than a jump box?

Jump box is very good but this approach can also expose organizations to enormous risks. If a malicious user breaches the perimeter, they could traverse through the organization's networks and resources with relative ease. Switching to more dynamic methods such as zero trust security, in which all network traffic is untrusted by default can be a better solution.

- What are the disadvantages of implementing a VPN that kept you from doing it this time?

- They require an in-depth understanding of public network security issues and proper deployment options.
- Availability and performance depends on factors largely outside of their control.
- VPNs require accommodation of protocols other than IP and existing internal network technology.

- What are the advantages of a VPN?

- A VPN protects private data
- They potentially give remote and international locations in particular better reach and service quality.
- VPNs can help avoid data-throttling

- When is it appropriate to use a VPN?

It is appropriate to use a VPN when protecting data to make sure it can be accessed securely through various devices.

Question 2: Corporate VPN

What are the advantages and disadvantages of using a corporate VPN, and under what circumstances is using one appropriate?

1. Restate the Problem

What are the pros and cons of using a corporate vpn and in what conditions is it best to use one?

2. Provide a Concrete Example Scenario

- In Project 1, which VMs did you have on the network?

I had the ELK-VM in the ELK-VNet. I also had Web-1,Web-2, Web-3 and the Jumpbox VMs in the RedTeam-VNet

- Which tools did you use to control access to and from the network?

I used the Network Security Group

- If you didn't use a VPN, what did you use?

I used the Network Security Group

- What disadvantage(s) did your non-VPN solution have?

With just firewalls, the network is more easily compromised but with a VPN, there is improved communication security through encryption and your real online identity is masked thereby providing an extra layer of privacy security.

- What advantage(s) did your non-VPN solution have?

The major advantage is that whenever there is a change in trust level between one network segment and another, using just a firewall (non-VPN solution) is always the best.

3. Explain the Solution Requirements

- Would a VPN meet the access control requirements you had for Project 1?

Yes It would because it can create a direct connection between the RedTeam-VNet and the ELK-VNet

- How would a VPN protect the network just as well, or better, than your current solution?

It would by providing an extra layer of security through communication encryption.

4. Explain the Solution Details

- Which Azure tools would you use to implement a VPN to your Project 1 network?

I would use the Virtual Network tool then create a Gateway Subnet and Local Network Gateway which the IP will be configured to then create a connection.

5. Identify Advantages and Disadvantages of the Solution

- In Project 1, would a VPN have been an appropriate access control solution?

Yes it would have.

- Under what circumstances is a VPN a good solution?

It's a good solution when it makes it possible for remote employees can access a company's resources within the company's LAN securely.
Also when it creates a Tunnel across the less secure networks, that only authorized users can access.

- When, if ever, is a VPN "overkill"?

It is an overkill when used together with another VPN or a software or application that directs internet traffic in such a manner that conceals a user's location and usage.

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

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

1. Restate the Problem

When is it best to use containers in cloud environments and how can they keep the network secure.

2. Provide a Concrete Example Scenario

- In Project 1, when did you use containers?

I started and attached the ansible container from the Jumpbox provisioner, then within the ansible host file, I added ELK to the group of host web servers.

- What did you use containers for?

The containers were used to configure the ELK VM

3. Explain the Solution Requirements

- Why was this an appropriate use for containers?

It was because containers are far more lightweight, start much faster, and use a fraction of the memory compared to a VM.

- What security benefits did you expect from using containers?

Containers running in the cloud do not interact with each other and this isolation brings a decrease in security risks.

4. Explain the Solution Details

- In Project 1, how did you configure VMs to be able to run containers?
  I did so by adding their web servers to the container host files, then created a playbook which enabled them to run containers.
- How did you select and install the correct container?
  I did so by running the YAML file

- How did you verify that it was running correctly?
  I did so by running the ansible playbook in respect to that particular VM

5. Identify Advantages/Disadvantages of the Solution

- How would you have achieved the same thing without containers?
  By using only Virtual Machines

- What are the advantages to doing it without containers?
  When using just VMs, there is better guest compatibility because VMs can run just about any operating system unlike containers that run same operating system version as the host
- What are the disadvantages?
  Since it runs a complete operating system, it requires more system resources such as CPU,memory and storage.

Question 4: Cloud Infrastructure as Code

What are the security benefits of defining cloud infrastructure as code?

1. Restate the Problem
What are the security pros of deploying cloud infrastructure as code

2. Provide a Concrete Example Scenario

- In Project 1, when did you use infrastructure as code (IaC)?
  I used it when creating a playbook file

- What tool did you use?
  I used the docker

- What did you use it to do
  I used it to configure ELK

3. Explain the Solution Requirements

- Were there any alternatives to IaC?

  Yes there are. Service provisioning can be used.

- What benefits does IaC have over alternative approaches?

  IaC makes it possible to automate your infrastructure.

4. Explain the Solution Details

- In Project 1, which specific configurations did your IaC set up?

```
- name: Config ELK with Docker
  hosts: elk
  become: true
  tasks:
  - name: Install docker.io
    apt:
      update_cache: yes
      name: docker.io
      state: present

  - name: Install pip3
    apt:
      force_apt_get: yes
      name: python3-pip
      state: present

  - name: Install Docker Python Module
    pip:
      name: docker
      state: present

  - name: Download and launch a Docker web container
    docker_container:
      name: elk
      image: sebp/elk:761
      state: started
      restart_policy: always
      published_ports:
        - 5601:5601
        - 9200:9200
        - 5044:5044

  - name: Configure elk VM to use more memory
    sysctl:
      name: vm.max_map_count
      value: "262144"
      state: present
```

```
      reload: yes

  - name: Enable Docker service
    systemd:
      name: docker
```

- How did you run and test these configurations?
I ran the playbook with the following command: ansible-playbook elk-playbook

5. Identify Advantages/Disadvantages of the Solution

- Are there any disadvantages to using IaC over the "traditional" approach?

- They require a lot of planning before the configuration - such as choosing the right tools.
- Bad configurations could get duplicated on all the servers