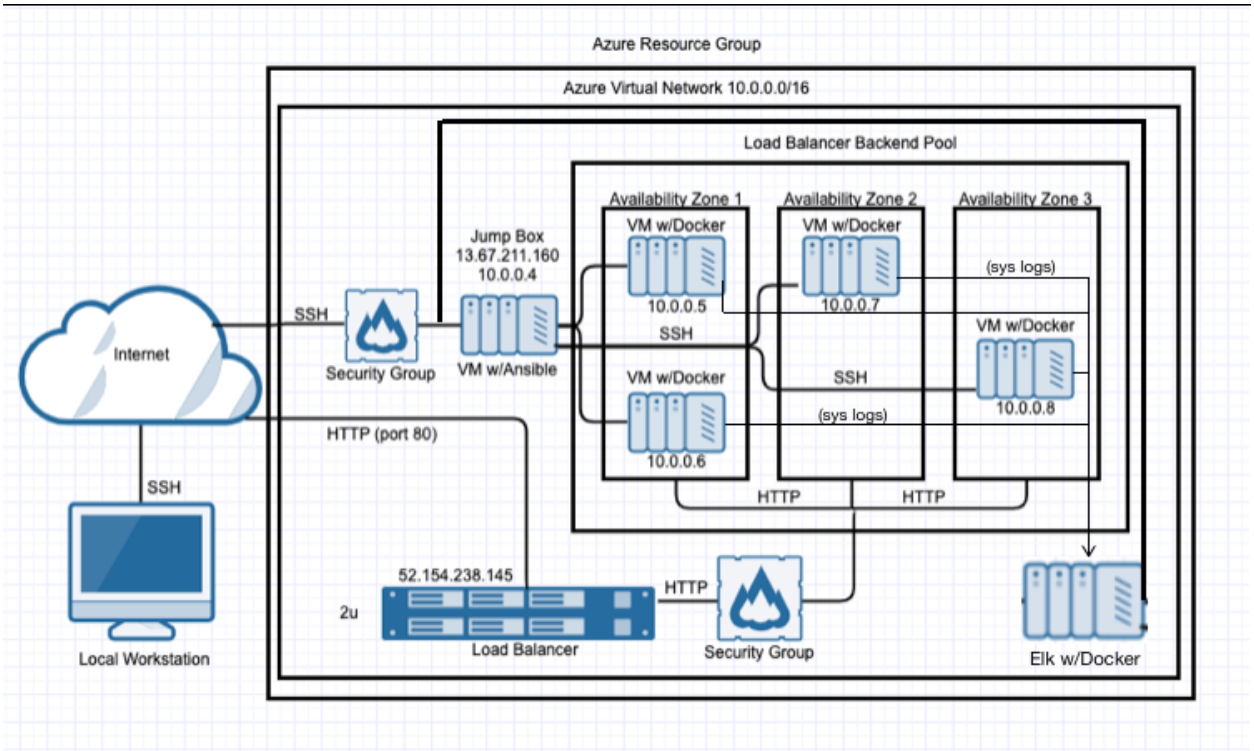
Automated ELK Stack Deployment

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




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

![ELK Stack Diagram](./Images/ELK-Stack-Diagram.png)

These files have been tested and verified and used to generate a live ELK Stack
deployment within Azure. They can also be used to either recreate the entire
deployment pictured above. And also maybe select portions of the yml files may
be used to install only certain pieces of it, such as Filebeat.

DVWA-Playbook.yml used to install DVWA Web Servers.

- [`DVWA-Playbook.yml` used to install DVWA
Webservers.](./DVWA/DVWA-Playbook.yml)

- [`install-elk.yml` is how to install ELK Stack Server.](./ELK/install-elk.yml)

- [`filebeat-config.yml` Filebeat configuration modified and copied to the
webservers as a `filebeat.yml`.](./Filebeat/filebeat-config.yml)
```

- [`filebeat-playbook.yml` I install Filebeat Syslog Service on the webservers.](./Filebeat/filebeat-playbook.yml)

- [`metricbeat-config.yml` Metricbeat configuration modified and copied to the webserver as a `metricbeat.yml`.](./Metricbeat/metricbeat-config.yml)

- [`metricbeat-playbook.yml` I install Metricbeat service on webservers](./Metricbeat/metricbeat-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 the web application is available across multiple web application servers restricting unwanted or unnecessary access to the Network.

A Load Balancer also may mitigate some DoS attacks as it can balance the load across many web application servers. Typically without the clients having to understand how many servers are in use or how they are configured.load balancers include a health probe to check all of the servers in its pool are functioning appropriately before sending traffic to them or it will stop sending traffic to missing or poor performing servers providing better uptime for the web application.

A Jump Box is similar to a gateway router as it becomes a single point of a protected network exposed to the public network as it sits in front of the other machines that are not exposed to the Internet. further explanation is that the jump box is a secure computer that all admins first connect to before launching any administrative task or use as an origination point to connect to other servers or untrusted environments.
To further control access only specified IP addresses and port 22 are allowed access to the Jump Box. To avoid the username and password weakness of SSH we used asynchronous encryption keys to ensure a higher degree of protection than usernames and passwords.

Integrating an ELK server allows users to easily monitor the vulnerable VMs for changes to the machine metrics and system logs.

Filebeat is used to capture system logs or file locations you specify then sending data to the ELK Server for indexing and review.

Metricbeat is used to capture machine metrics on Linux, Windows, and Mac hosts then forwarding to the ELK Server to track system level CPU usage, memory, file system, disk I/O, and network I/O metricbeat

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
Abdirahman	Worstation	24.118.22.58	WINDOWS 10
Jump Box	Gateway	10.0.0.4	Linux UBUNTU 18.4
WEB-1	DVWA	10.0.0.5	LINUX UBUNTU 18.4
WEB-2	DVWA	10.0.0.6	LINUX UBUNTU 18.4
WEB-3	DVWA	10.0.0.7	LUNUX UBUNTU 18.4
ELK-Server	ELK Stack	10.1.0.4	LINUX UBUNTU 18.4

Access Policies

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

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

- Machines within the network can only be accessed by Jump Box with the private IP address 10.0.0.4

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

Name	Publicly Accessible	Allowed IP Addresses
Abdirahman	YES	10.0.0.0/16 10.1.0.0/16
Jump Box	Yes	10.0.0.0/16 10.1.0.0/16
WEB-1	NO	10.0.0.0/16 10.1.0.0/16
WEB-2	NO	10.0.0.0/16 10.1.0.0/16
WEB-3	NO	10.0.0.0/16 10.1.0.0/16
ELK-Server	NO	10.0.0.0/16 10.1.0.0/16

Elk Configuration

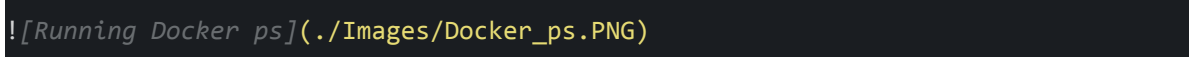
Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because...

its simplified deployment from the central location that can be used to expand or redeploy the ELK Stack just by running Ansible playbooks.

The playbook implements the following tasks:
In 3-5 bullets, explain the steps of the ELK installation play. E.g., install Docker; download image; etc._

- *The playbook implements the following tasks:*
- *First I Install docker.io, Python, and the Docker Module.*
- *Then I Configure the ELK Server memory and increased.*
- *After that I Download Docker ELK Container and configure it.*

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



Target Machines & Beats
This ELK server is configured to monitor the following machines:

- ***10.0.0.5***
- ***10.0.0.6***
- ***10.0.0.7***

I installed the following Beats on these machines:

- ***Filebeat***
- ***Metricbeat***

These Beats allow to collect the following information from each machine:

- *System log and application log details which include Web Traffic is gathered by Filebeat.*
- *CPU, Memory, Disk, Network, and other top-like statics are gathered with Metricbeat.*

Using the Playbook

In order to use the playbook, I will need to have an Ansible control node already configured, Ansible control node on the jump box

SSH into the control node and follow the steps below:

- *Copy the configuration file to the /etc/ansible/files/ to make configuration changes.*
- *Update the /etc/ansible/hosts file to include the targeted machine or machines.*

- *Create the* <role>-playbook.yml file with the required tasks to be run by ansible-playbook.
- Run the playbook then navigate to the ELK Server Kibana data installation page to check the Module status that the installation worked as expected.

Ansible playbook files were <role>-playbook.yml files located on the Jump Server within the Ansible Docker container in the /etc/ansible/roles/ folder.

In order update specific machines we edited the /etc/ansible/hosts by ensuring the [header] element is not commented out with a # or needs to be created then the hostname/IP of the machines are added to the file for Ansible to target groups of machines.

- Navigating to <*http://104.210.155.66/app/kibana*> successfully ensures the ELK Server is running and is ready for use.

Commands I used to install ELK Stack, Filebeat, and Metricbeat

- first I Install the ELK Stack on the elk server.
- then Connected to the Jump Box and attach to the Ansible container
- SSH into the Jump-Box by using ssh sysadmin@137.135.107.66

- Locate the container name:

```bash
Azureuser@Jump-Box-Provisioner:~$ sudo docker container list -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED
STATUS                   PORTS     NAMES
56b542ca508c  cyberxsecurity/ansible         "/bin/sh -c /bin/bas…"   2 weeks
ago   Exited (0) 4 hours ago             festive_wiles
```

- Start the container:

```bash
- Azureuser@Jump-Box-Provisioner:~$ sudo docker container start festive_wiles
```

```bash
- root@56b542ca508c:~#
```

- Attach (connect) to the Ansible container: festive_wiles

```bash
- Azureuser@Jump-Box-Provisioner:~$ sudo docker container attach festive_wiles
```

```bash
- root@56b542ca508c:~#
```

Update the Ansible hosts file and create yml playbook file

Add the ELK Server IP address to the Ansible /etc/ansible/hosts file creating an [elk] section with the IP address:

- Open hosts file:

```bash
root@56b542ca508c:~# nano /etc/ansible/hosts
```

Add the [elk] section followed by the ELK Server IP address:

[elk]

10.1.0.4

Create the Ansible playbook used to install and configure elk container on ELK Server virtual machine.

```bash
- root@56b542ca508c:~# nano /etc/ansible/roles/elk-playbook.yml
```

ELK install and the configuration tasks can be seen in the elk-playbook.yml playbook to automate ELK Stack deployment.

Exit nano and save the playbook.

Running the Playbook and testing the results

- Run the Ansible playbook:

```bash
root@56b542ca508c:/etc/ansible/roles# ansible-playbook elk-playbook.yml
```

```bash
root@56b542ca508c:/etc/ansible/roles# ansible-playbook elk-playbook.yml

PLAY [Configure Elk VM with Docker]
**********************************************
```

```
TASK [Gathering Facts]
*********************************************************************
ok: [10.1.0.4]

TASK [Install docker.io]
*********************************************************************
changed: [10.1.0.4]

TASK [Install python3-pip]
*********************************************************************
changed: [10.1.0.4]

TASK [Install Docker module]
*********************************************************************
changed: [10.1.0.4]

TASK [Increase virtual memory]
*********************************************************************
changed: [10.1.0.4]

TASK [Increase virtual memory on restart]
*********************************************
changed: [10.1.0.4]

TASK [download and launch a docker elk container]
******************************************
changed: [10.1.0.4]

TASK [Enable service docker on boot]
*********************************************************
changed: [10.1.0.4]

PLAY RECAP
*********************************************************************************
10.1.0.4                   : ok=1    changed=7    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

- ELK container is installed, SSH to your container and also double-check that my elk-docker container is running.

```bash
root@56b542ca508c:/etc/ansible/roles# ssh Azadmin@10.1.0.4
Azadmin@elk:~$ sudo docker ps
CONTAINER ID    IMAGE         COMMAND                  CREATED      STATUS
PORTS
NAMES
```

```
e16768c0f61   sebp/elk:761   "/usr/local/bin/star…"   7 days ago   Up 2 minutes
0.0.0.0:5044->5044/tcp, 0.0.0.0:5601->5601/tcp, 0.0.0.0:9200->9200/tcp, 9300/tcp
elk
```

```bash
Azadmin@elk-server:~$
```

- the ELK server container is up and running.

Navigate to *http://52.175.211.238:5601/app/kibana* to verify the ELK Stack is running
Use the public IP address of the ELK server from Azure.

Install the Filebeat on to the webserver virtual machines

First the Filebeat Configuration File was created
Stay attached to the Ansible container on the Jump box.
then Installed the ELK Stack on the elk server after that i Connected to the Jump Box and attached to the Ansible container for steps to attach to the Ansible container.
then Copied filebeat-config.yml to the Ansible container.

```bash
root@56b542ca508c:/etc/ansible# curl
https://gist.githubusercontent.com/slape/5cc350109583af6cbe577bbcc0710c93/raw/eca603b72586fbe148c11f9c87bf96a63cb25760/Filebeat >
/etc/ansible/files/filebeat-config.yml
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 73112  100 73112    0     0   964k      0 --:--:-- --:--:-- --:--:--...
```

Side note: I didn't know how to share this on my git hub so I downloaded it to google drive. Thank you.