## Automated ELK Stack Deployment

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

![TODO:](Images/Project\_1\_Network\_Diagram\_2.0.pdf)

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

![TODO:](Playbooks/Ansible\_config.yml)

![TODO:](Playbooks/Filebeat.yml)

![TODO:](Playbooks/Metricbeat.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

-Commands utilized throughout Project 1 within the Terminal

### 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 available by disbursing traffic to different web servers in the resource pool to ensure that no single server becomes overwhelmed or overworked, in addition to restricting access to the network. Load balancers will minimize the server response time and maximize the throughput. They play an important role in cloud security as they defend organizations/companies (private and public) from DDoS attacks. This is done by shifting attack traffic from corporate severs to public cloud provider. Load balancers route client requests across all the servers capable of fulfilling those requests in a manner that maximizes speed and capacity utilization, which if unkept or monitored could be detrimental to performance within the cloud.

The Jump Box is a secure computer within a virtual environment that all admins first connect to before launching any administrative task or use an origination point to connect other servers or untrusted environments. Jump Boxes must be configured so that they are less likely to be exploited and also must be restricted in what they are allowed to visit. Jump Boxes are also known as a controlled entry point into a virtual environment in the cloud and is a critical step in establishing a security zone that is reliable and effective.

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

Filebeat monitors log files or locations that you specify and collects log events and while Metricbeat collects metrics/statistical data from the OS and services running on the servers. Filebeat and Metricbeat then takes the information collected and outputs it to where it is specified to go.

The configuration details of each machine may be found below.

| Name | Function. | IP Address | Operating System|

|------------- |------------- |---------------|------------------------ |

| Jump Box | Gateway | 10.0.0.4 | Ubuntu |

| Web\_1 | DVWA Server| 10.0.0.5 | Ubuntu |

| Web\_2 | DVWA Server| 10.0.0.6 | Ubuntu |

| ELK Server| ELK Stack | 10.1.0.4 | Ubuntu |

### 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 my personal IP address and machines within the Azure Virtual Environment via SSH. The ELK Server is accessed through SSH via the Jump Box from my personal IP address.

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

| Name | Publicly Accessible | Allowed IP Addresses |

|-----------|---------------------|------------------------|

| Jump Box | Yes | my IP |

| Web-1 | No | 137.135.43.100 |

| Web-2 | No | 137.135.43.100 |

| ELK Server| Yes | my IP |

### Elk Configuration

Ansible was used to automate configuration of the ELK machine; no configuration was performed manually. This is advantageous because in automating the deployment we can set up numerous machines at one time in order to avoid having to write scripts on every machine where errors can occur.

The playbook implements the following tasks:

-Installation of a docker.io

-Installation of pip3

-Download, configure and launch the ELK docker container via the Virtual Machine in MS Azure

-Established and started the dvwa and published the ports

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](Additional\_Documents/Creation\_of\_ELK\_Server\_via\_Azure.pdf)

### Target Machines & Beats

This ELK server is configured to monitor the following machines:

-Web-1 - 10.0.0.5

-Web-2 - 10.0.0.6

Both Filebeat and Metricbeat have been installed on all the active machines. These Beats allow us to collect the following information from each machine:

-Filebeat will collect the log files on the hosts then transfer the logs to the ELK Server where they can be opened and inspected via Kibana.

-Metricbeat will collect the necessary metrics and statistical data requested then transfer the information to the ELK Server to be opened and inspected via Kibana.

### Using the Playbooks

In order to use the playbooks, 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 for Filebeat and Metricbeat playbooks:

-Copy the filebeat\_config.yml file to /etc/ansible/files.

-Update the config file to include the private IP of the ELK Server to the Elasticsearch and Kibana sections of the config file.

-Run the playbook and navigate to the ELK Servers IP:5601 to check that the installation worked as expected.

-Copy the metricbeat\_config.yml file to /etc/ansible/files.

-Update the config file to include the private IP of the ELK Server to the Elasticsearch and Kibana sections of the config file.

-Run the playbook and navigate to the ELK Servers IP:5601 to check that the installation worked as expected.

- The playbook is the Elk.yml and it goes to etc/ansible

- The etc/ansible/hosts is where you add the IP address of the machines you want it installed only this will allow you to put on as many machines as are needed.

You will navigate to http://20.55.105.58:5601 in order to check that the ELK server is running.

![TODO:](Images/Kibana\_HTTP\_Address)

As a bonus I have provided a list of all the commands utilized for Project 1 ELK Server:

![TODO:](Commands\_Utilied/Creation-Deployment\_for\_ELK\_Server.pdf)