## Automated ELK Stack Deployment

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

(Images/Tim\_Project1\_Diagram.drawio)

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 Ansible files may be used to install only certain pieces of it, such as Filebeat.

-INSTALL\_ELK.yml

-Elk-Config.yml

-Filebeat-playbook.yml

-Filebeat-config.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 accessible, in addition to restricting access to the network.

The Load Balancer ensures that if one of the Web Vm’s is unavailable, the network is still accessible to the users. There are several advantages to using the JumpBox configuration, the first is the ease of configuring the other VM’s on the network via Docker and Ansible. The second, and from a security perspective, more important, is that the Jumpbox is the only way to access the network thereby limiting the surface area available to attack.

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

- \_TODO: What does Filebeat watch for?\_

- \_TODO: What does Metricbeat record?\_

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 | 10.0.0.4 | Linux |

| My Load Balancer | Load Balancer | 20.98.81.100 | n/a |

| Web-1 | Server | 10.0.0.10 | Linux |

| Web-2 | Server | 10.0.0.11 | Linux |

| Web-3 | Server | 10.0.0.12 | Linux |

| ELK-VM | ELK Server | 10.1.0.4 | Linux |

### 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:

- All Public IP’s are white listed.

Machines within the network can only be accessed by the Jump Box after attaching the great-swirles docker container.

-The ELK-VM can be only directly accessed via SSH by the Jump Box after attaching the great\_swirles docker container. The Jump Box private IP is 10.0.0.4\_

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

| Name | Publically Accessible | Allowed IP Addresses |

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

| Jump Box | Yes | Any |

| My Load Balancer | No | None |

| Web-1 | No | 10.0.0.4 |

| Web-2 | No | 10.0.0.4 |

| Web-3 | No | 10.0.0.4 |

| ELK-VM | Yes | Any |

### Elk Configuration

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because it avoids the potential human error of setting up each machine manually and individually. The Ansible configuration is a faster process then configuring each machine individually as a manual configuration would require.

The playbook implements the following tasks:

- \_TODO: In 3-5 bullets, explain the steps of the ELK installation play. E.g., install Docker; download image; etc.\_

- Install docker.io

- Install python3-pip

- Install Docker module

- Increase virtual memory

- download and launch a docker elk container

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

(Images/elk-VM\_sebp-elk-761)

### Target Machines & Beats

This ELK server is configured to monitor the following machines:

- Web-1:10.0.0.10

- Web-2:10.0.0.11

- Web-3:10.0.0.12

We have installed the following Beats on these machines:

- FileBeats

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

- Filebeat monitors the log files or other specified locations, collects log events, and forwards the collected data to either Elasticsearch or Logstash for indexing.

### 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:

**TO Install the ELK server: Note: target VM must be active**

- Copy the Install-ELK.yml file to /etc/ansible.

-‘cp <yourdownloadlocation/Install-ELK.yml> /etc/ansible/Install-ELK.yml’

-‘cd /etc/ansible’ and verify ‘cp’ success

-NOTE: ‘cp’ used instead of ‘mv’ to ensure a clean file is immediately available if needed.

- Update the ansible.cfg file to include the ELK Server Private IP and the private IP’s of servers to be monitored and include the location in Install-ELK.yml file.

-‘nano ansible.cfg’ modify the elk server section, modify ‘webservers’ with servers private IP’s

-‘nano Install-ELK.yml’ modify install location to reflect ansible.cfg Elk server location

- Run the playbook, ‘ansible-playbook Install-ELK.yml’

- To verify the installation was successful: navigate to http:<ELK-VMPublicIP>:5601/app.kibana on your browser to check that the installation worked as expected. The webpage should show the Kibana homescreen.

**To Install Filebeats: Note: target VM’s must be active**

-Copy the Filebeat-playbook.yml and Filebeat-config.yml to /etc/ansible

-‘cp <yourdownloadlocation/Filebeat-config.yml> /etc/ansible/Install-ELK.yml’

-‘cp <yourdownloadlocation/Filebeat-playbook.yml> /etc/ansible/Filebeat-playbook.yml’

-‘cd /etc/ansible’ to verify ‘cp’ success

-NOTE: My personal preference is to copy important files one at a time and the commands above reflect that.

-Modify Filebeat-config.yml to include the ELK Server IP.

-‘nano Filebeat-config.yml’ add in ELK Server IP

-Modify the Filebeat-playbook.yml to include the IP’s of the servers to be monitored.

- nano Filebeat-playbook.yml’ verify that drop in location is ‘webservers’, which was modified previously in Ansible.cfg.

-Run the playbook with ‘ansible-playbook Filebeat-playbook.yml’

-To verify installation, open a browser window and navigate to http:<ELK-VMPublicIP>:5601/app.kibana, add data, select logs, select filebeats, select receiving data to verify filebeats is reporting to the ELK Server.