## Automated ELK Stack Deployment

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

**My files reside in ~/Downloads/README/README**

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.

**The playbook resides in “/etc/ansible/roles/”**

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 Availability, in addition to restricting in-bound to the network.

**The off-loading function of a load balancer defends an organization against distributed denial-of-service (DDoS) attacks. It does this by shifting attack traffic from the corporate server to a public cloud provider.**

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

**Changes to file changes on the machine**

**Collect metrics from the operating system and from services running 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 | 10.1.0.4 | Linux |**

**| Web VM 1 | Webserver | 10.1.0.5 | Linux |**

**| Web VM 2 | Webserver | 10.1.0.6 | Linux |**

**| Elk-Server | Monitor Traffic | 10.0.0.4 | Linux |**

### Access Policies

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

Only the Jumpbox Provisioner machine can accept connections from the Internet. Access to this machine is only allowed from the following IP addresses:

**5601 Kibana**

Machines within the network can only be accessed by Jumpbox Provisioner.

**My ansible container inside my Jumpbox “68.52.23.138”**

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

**| Name | Publicly Accessible | Allowed IP Addresses |**

**|----------|---------------------|----------------------|**

**| Jump Box | no | |68.52.23.138**

**| Web VM 1 | no | 10.1.0.4 |**

**| Elk-Server | no | 10.1.0.4 |**

### Elk Configuration

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

**It is flexible you can make changes to the entire deployment.**

The playbook implements the following tasks:

**Install docker.io**

**Install pip3**

**Install Docker python module**

**Increase virtual memory**

**Download and launch a docker**

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:

**It is set to monitor the Web VM 1 “10.1.0.5” && “10.1.0.6”**

We have installed the following Beats on these machines:

**Filebeat**

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

**File beat is only pulling files from the file system. Metricbeat is collecting machines’ metrics while it is in uptime.**

### 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 \_\_\_\_\_ file to \_\_\_\_\_.

- Update the \_\_\_\_\_ file to include

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

**Ansible playbook file to Ansible control node**

**Hosts**

**Kibana**