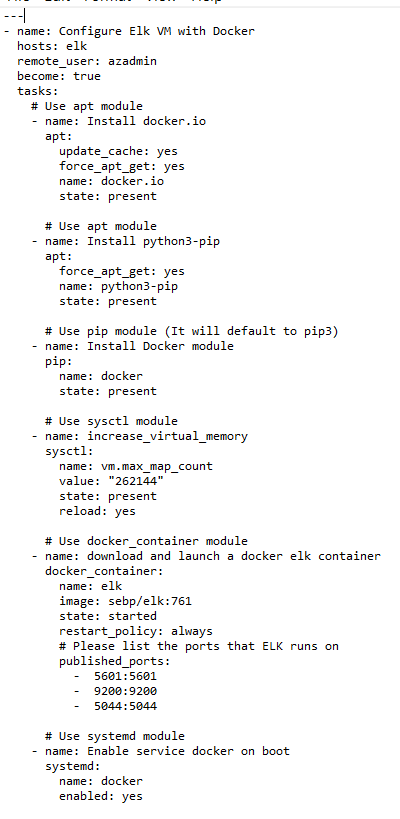
## Automated ELK Stack Deployment

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

![TODO: Update the path with the name of your diagram](Images/diagram\_filename.png)DRAW.IO

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 playbook file may be used to install only certain pieces of it, such as Filebeat.

- \_TODO: Enter the playbook file.\_



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 **available**, in addition to restricting **access** to the network.

- \_TODO: What aspect of security do load balancers protect? What is the advantage of a jump box?\_

**Load Balancers protect the availability and make sure the functionality of the server remains good. The Jumpbox is able to run smoothly because of the load balancers.**

Integrating an ELK server allows users to easily monitor the vulnerable VMs for changes to the **log files** and system **resources**.

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

**Filebeat collects logs events and forwards any changes to the Elasticsearch Host**

- \_TODO: What does Metricbeat record?\_

**Metricbeat is used to record metrics within a target server and then displays in Elasticsearch**

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 | Ubuntu |**

**| Web 1 | web server | 10.0.0.5 | Ubuntu |**

**| Web 2 | web server | 10.0.0.6 | Ubuntu |**

**| Elk | ElasticSearch Stack | 10.1.0.4 | Ubuntu |**

### Access Policies

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

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

- \_TODO: Add whitelisted IP addresses\_

Machines within the network can only be accessed by **ssh**.

- \_TODO: Which machine did you allow to access your ELK VM? What was its IP address?\_

**The elk vm is only accessible via ssh from the Jumpbox provisioner and through web access 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 | No | Personal IP |**

**| web 1 | yes load balancer | 10.0.0.5 jumpbox private 52.183.9577 LB public ip |**

**| web 2 | yes load balancer | 10.0.0.6 jumpbox private 52.183.9577 LB public ip |**

**| elkvm | No | ssh 10.1.0.4 Jumpbox http port 5601 personal ip |**

### 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?\_

**An advantage of Ansible is easy deployment of multiple servers without touching each server and it is a free open-source tool.**

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**

**Increase VM Memory**

**Install Python**

**Download and launch Elk docker**

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\_output.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\_

**Web 1 10.0.0.4**

**Web 2 10.0.0.5**

We have installed the following Beats on these machines:

- \_TODO: Specify which Beats you successfully installed\_

**Filebeat**

**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.\_

### 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 **ssh-keygen** file to **ssh public key.**

- Update the **security rule** file to include...

- Run the playbook, and navigate to **ELK VM Public IP/app/kibana** to check that the installation worked as expected.

\_TODO: Answer the following questions to fill in the blanks:\_

- \_Which file is the playbook? Where do you copy it?\_

- \_Which file do you update to make Ansible run the playbook on a specific machine? **Ansible-playbook install-elk-playbook.yml** How do I specify which machine to install the ELK server on versus which to install Filebeat on?\_

- \_Which URL do you navigate to in order to check that the ELK server is running?

\_As a \*\*Bonus\*\*, provide the specific commands the user will need to run to download the playbook, update the files, etc.\_