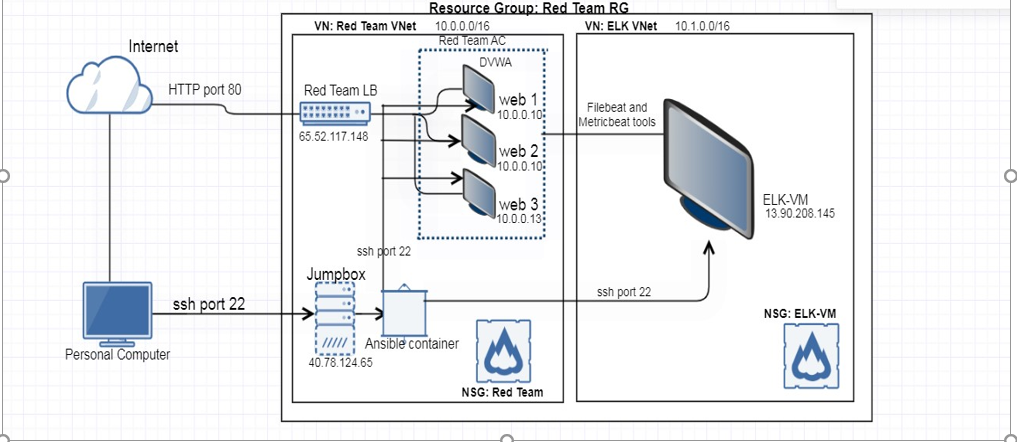
## Automated ELK Stack Deployment

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



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

* [My Playbook](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Ansible/my_playbook.yml)
* [Install ELK Stack Playbook](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Ansible/Install-ELK.yml)
* [Filebeat Playbook](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Ansible/Filebeat-playbook.yml)
* [Metricbeat Playbook](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Ansible/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 that the application will be highly available in addition to restricting access to the network. Load balancer are designed to secure your network by disbursing traffic to different web servers in the resource pool. It ensures that no single server becomes overworked and subsequently unreliable. Jump boxes are usually a single audit point for traffic where prospective administrators must log in in order to gain access to the DMZ assets and all access can be logged for a later audit. It is security hardened and treated as a single entryway and a "pivot server" to access other servers within your network.

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

The configuration details of each machine may be found below.

| **Name** | **Function** | **IP Address - Public** | **IP Address - Private** | **Operating System** |
| --- | --- | --- | --- | --- |
| Jump-Box Provisioner | Gateway | 20.102.121.98 | 10.0.0.4 | Linux Ubuntu 20.04 |
| Web-1 | VM Server | None | 10.0.0.7 | Linux Ubuntu 20.04 |
| Web-2 | VM Server | None | 10.0.0.8 | Linux Ubuntu 20.04 |
| Web-3 | VM Server | None | 10.0.0.9 | Linux Ubuntu 20.04 |
| ELK Server | ELK Stack | 52.162.176.157 | 10.1.0.4 | Linux Ubuntu 20.04 |

**Access Policies**

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

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

* Shauns’ Home Computer Public IP Address

Machines within the network can only be accessed by the Jump-Box-Provisioner.

* Shauns’ Home Computer Public IP Address

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

| **Name** | **Publicly Accessible** | **Allowed IP Address** |
| --- | --- | --- |
| Jump-Box Provisioner | Yes | Shauns’ Home Computer Public IP Address |
| Web-1 | No | 10.0.0.4 |
| Web-2 | No | 10.0.0.4 |
| Web-3 | No | 10.0.0.4 |
| ELK Server | Yes | Shauns’ Home Computer Public IP Address |

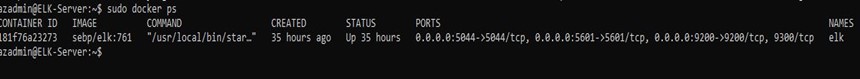
**Elk Configuration**

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because it allows streamlined, consistent, and identical configuration of a wide range of systems and devices such as databases, storage devices, networks, firewalls, etc., all at one time.

The playbook implements the following tasks:

* Install docker.io
* Install python3-pip
* Install docker module
* Increase and use more 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.

[](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Images/ELK.jpg)

**Target Machines & Beats**

This ELK server is configured to monitor the following machines:

| **Name** | **IP Address** |
| --- | --- |
| Web-1 | 10.0.0.7 |
| Web-2 | 10.0.0.8 |
| Web-3 | 10.0.0.9 |

We have installed the following Beats on these machines:

* Filebeat
* Metricbeat

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

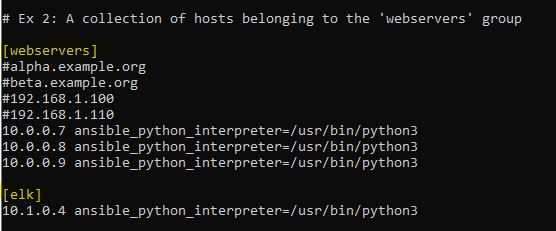
* Filebeat: It helps generate and organize log files to send to Logstash and Elasticsearch. Specifically, it logs information about the file system, including which files have changed and when. It is often used to collect log files from very specific files such as logs generated by Apache, Microsoft Azure tools, the Nginx web server, or MySQL databases.
* Metricbeat: It collects machine metrics such as CPU usage and uptime. It collects metrics from your system and services and makes it easy to monitor specific information about the machines in the network.

**Using the Playbook**

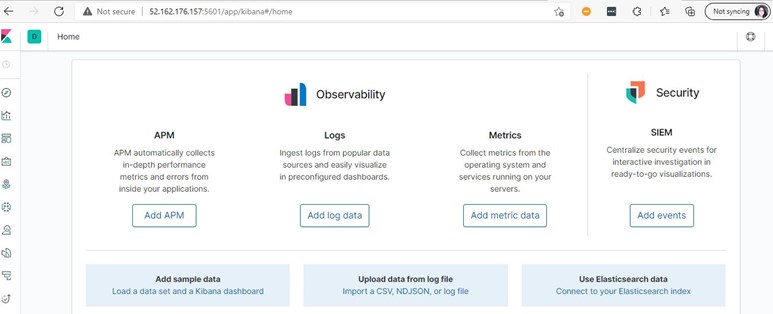
In order to use the playbook, you will need to have an Ansible control node already configured. Assuming you have such a control node provisioned:

SSH into the control node and follow the steps below:

* Copy the 'ansible.cfg' file to the '/etc/ansible' directory inside the ansible container. Update this configuration file by entering the 'remote\_user\_name' of your choice.
* Update the hosts file to include the 'webservers' groups with Web-1, Web-2, and Web-3 private IP addresses and also the 'elk' group making sure to include the Elk Server's private IP address to Ansible's inventory. Be sure to include te python3 interpreter next to each IP address.

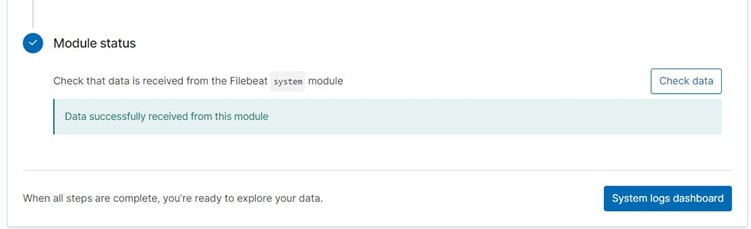
[](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Images/Webserver_Elk_Hostgroups.JPG)

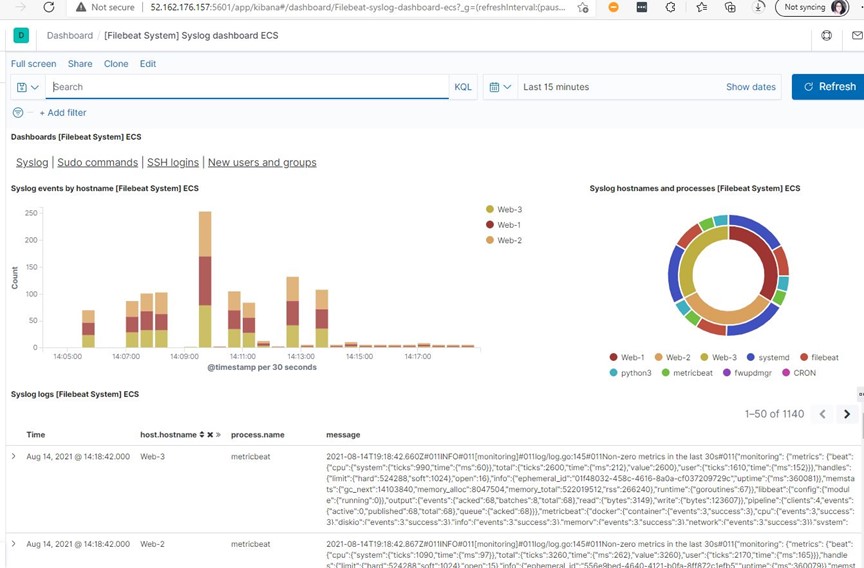
* Once the groups have been set, create a playbook. The playbook is named "Install-ELK.yml". This file should be located within the '/etc/ansible' directory.
* Run the playbook, and navigate to the URL <http://52.162.176.157:5601/app/kibana> to verify that the installation worked.

[](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Images/Kibana_Snapshot.jpg)

**Using Filebeat**

* In your '/etc/ansible' directory, locate the configuration file for filebeat (filebeat.config).
* Nano into the filebeat.config file located in the '/etc/ansible/' directory to make the necessary changes to your configuration.
* Scroll to line #1106 and replace the IP address with the internal IP address of the ELK server. Next, scroll to line #1806 and replace the IP address with the internal IP address of the ELK server.
* Run the playbook, using the command 'ansible-playbook filebeat-playbook.yml'. This playbook will run the tasks you have specified in the playbook by downloading the filebeat software from a web repository to each webserver. It will copy the filebeat-config.yml from the JumpBox to every webserver designated in the hosts file. Lastly, it will build and start the filebeat container on each webserver.
* Once completed, you can verify the installation and confirm that the playbooks worked by verifying that the ELK stack is receiving logs. You can of course view this through Kibana and are able to analyze and search through the log data for all your security issues.

[](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Images/Filebeat_System_Snapshot.jpg)

[](https://github.com/TiffanyLG/TG-Cybersecurity--Repository/blob/main/TG-Project1/Images/Filebeat_Dashboard.jpg)