**## Automated ELK Stack Deployment**

​

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

* <https://github.com/mzainab8/Automated-ELK-Stack-Deployment/blob/main/Images/Project1-Diagram.png> ​

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.*

* [Elk Stack Playbook](https://github.com/mzainab8/Automated-ELK-Stack-Deployment/blob/main/Ansible/ElkstackPlaybook)

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 monitor instance of DVWA, the D*\*mn Vulnerable Web Application.*

​

Load balancing ensures that the application will be highly ***available****, in addition to restricting* ***traffic*** *to the network.*

* Load balancers can distribute the workload between multiple machines so that if a denial-of-service (DDos) attack occurs the system can still maintain availability. The jump box serves as a gateway to our other virtual machines within our network. It allows us the ability to set security control settings to a single machine so that entire virtual machine network isn’t exposed to the public.

​

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

* Filebeat monitors the log files or locations that you specify.
* Metricbeat records the metrics and statistics from the operation system and from services running on the server.

The configuration details of each machine may be found below.

​

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Function** | **IP Address** | **Operating System** |
| JumpBoxProvisioner | Gateway | 10.0.0.4 | Linux |
| Web-1 | Webserver 1 | 10.0.0.5 | Linux |
| Web-2 | Webserver 2 | 10.0.0.6 | Linux |
| Web-3 | Webserver 3 | 10.0.0.7 | 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:*

* 68.100.93.11

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

* Only the Jump Box VM has access to the ELK VM. The IP address of the Jump Box VM is 10.0.0.4

​

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

​

|  |  |  |
| --- | --- | --- |
| **Name** | **Publicly Accessible** | **Allowed IP Addresses** |
| JumpBoxProvisioner | No | 68.100.93.11 |
| Web-1 | No | 10.0.0.4 |
| Web-2 | No | 10.0.0.4 |
| Web-3 | No | 10.0.0.4 |
| ElkVM | No | 10.0.0.4 |

​

**### Elk Configuration**

​

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because it is easy to set up and use. There is no need to install any other software and you can customize to your needs.

​

The playbook implements the following tasks:

* Install Docker
* Install python3-pip
* Install Docker python module
* Set the vm.max\_map\_count to 262144
* 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/mzainab8/Automated-ELK-Stack-Deployment/blob/main/Images/DockerPS.png>

**### 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 |
| Web-3 | 10.0.0.7 |

We have installed the following Beats on these machines:

* Filbeat
* Metricbeat

​

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

* Filebeat monitors specified log files and sends to Elasticsearch or Logstash to process. Metricbeat collects records metrics the our machine’s operation system and from services running on the server from time to time and sends to Elasticsearch or Logstash to process.

**### 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 /etc/ansible
* add the machine, its IP, and ansible\_python\_interpreter=/usr/bin/python3 to the hosts in the ansible.cfg
* nano /etc/ansible/hosts
* under the [webservers] header add:
* 10.0.0.4 ansible\_python\_interpreter=/usr/bin/python3
* 10.0.0.5 ansible\_python\_interpreter=/usr/bin/python3
* 10.0.0.6 ansible\_python\_interpreter=/usr/bin/python3
* under the [elk] header add:
* 10.1.0.4 ansible\_python\_interpreter=/usr/bin/python3
* Copy the install-elk.yml and filebeat-playbook.yml file to /etc/ansible.
* Update the install-elk.yml and filebeat-playbook.yml file to include the machine you want use the playbooks on by changing the hosts name on the 3rd line.
* Run the playbook and navigate to http://40.112.213.52:5601/app/kibana to check that the installation worked as expected.

*Bonus:*

* nano ansible.cfg
* add the machine, its IP, and ansible\_python\_interpreter=/usr/bin/python3 to the hosts
* ctrl + s to save file
* ctrl + x to exit file
* in the folder that install-elk.yml is in, run: cp install-elk.yml /etc/ansible
* nano install-elk.yml /etc/ansible
* name: installing elk hosts: [your\_machine]
* ctrl + s to save file
* Ctrl + x to exit file
* ansible-playbook install-elk.yml