**Automated ELK Stack Deployment**

The network pictured below was configured using the files in this repository.

Diagram

Description automatically generated

These files have been thoroughly tested and used to create a real ELK deployment on Azure. They may be used to reproduce the whole deployment seen in the image above. Select elements of the yml and config files, such as Filebeat, may also be used to install only particular parts of it.

Ansible Configuration - ansible.cfg

My Playbook - playbook.yml

Filebeat Playbook - filebeat-playbook.yml

Filebeat Configuration - filebeat-config.yml

Metricbeat Playbook - metricbeat-playbook.yml

Metricbeat Playbook - metricbeat-config.yml

Elk - elk.yml

Hosts

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

The following information is included in this document:

• Topology Description

• Access Policies

• ELK Configuration

o Beats in Use

o Machines Monitored

• How to Use the Ansible Build

**The Topology is described.**

The primary goal of this network is to offer a load-balanced and monitored instance of the D\*mn Vulnerable Web Application, or DVWA.

In addition to limiting network access, load balancing guarantees that the application is highly accessible.

• The load balancer protects the network's availability by spreading traffic equally so that no one server is overburdened, preventing DoS attacks.

• The Jump Box Provisioner automates computer configuration changes and limits access to the virtual network in Infrastructure as a Code (IaC) configurations.

Users may simply monitor changes to data and system logs on susceptible VMs by integrating an ELK server.

• Filebeat keeps an eye on the log files or locations you designate, collects log events, and sends them to Elasticsearch or Logstash to be indexed.

• Metricbeat is a server monitoring tool that collects metrics from the system and services operating on the server, such as Apache, MySQL, HAProxy, MongoDB, and so on.

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** |
| JumpBox-Provisioner | Gateway | 10.0.0.5 | Linux |
| Web-2 | Host DVWA 1 | 10.0.0.4 | Linux |
| Web-3 | Host DVWA 2 | 10.0.0.6 | Linux |
| Web-4 | Host DVWA | 10.0.0.7 | Linux |
| ELK | Metric/Log Collection | 10.1.0.4 | Linux |

### **Policy on Access**

### The internal network's machines aren't connected to the internet.

### Internet connections can only be accepted by the Jumpbox machine. Only the following IP addresses have permission to access this machine:

### • SSH port 22 on my Workstation with my Public IP.

### Only 10.0.0.5 has access to the network's machines (ELK Server).

### • How did you grant access to your ELK VM to which machine? TCP port 5601 to my public IP address

### In the table below, you'll find an overview of the access policies in place.

|  |  |  |
| --- | --- | --- |
| **Name** | **Publicly Accessible** | **Allowed IP Addresses** |
| Jump Box | No | My Workstation Public IP via SSH on Port 22 |
| Web 2 | No | 10.0.0.4 via SSH on Port 22 |
| Web 3 | No | 10.0.0.6 via SSH on Port 22 |
| Web 4 | No | 10.0.0.7 via SSH on Port 22 |
| ELK | Yes | My Workstaion Public IP via HHTP on Port 80 |

### **Elk Configuration**

Ansible was used to automate the ELK machine's configuration. There was no manual configuration, which is advantageous because...

•

What are the primary benefits of using Ansible to automate configuration? It's adaptable, making it simple to reuse changes on other associated VMs.

The following tasks are carried out by the playbook:

• Set up Docker •

Set up Python3-pip

• Enable Docker on Boot •

Increase and Use Virtual Memory

• Install and Launch Docker Elk Container

The output of running docker ps after successfully configuring the ELK instance is shown in the following screenshot.

Text

Description automatically generated

### Target Machines & Beats

This ELK server is configured to monitor the following machines:

* Web-2: 10.0.0.4
* Web-3 : 10.0.0.6
* Web-4 : 10.0.0.7

We have installed the following Beats on these machines:

* Filebeat
* Metricbeat

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

* Filebeat monitors data logs such as Unique vistors, operating systems, heatmap etc.
* Metricbeats collects metrics and system statistics such as CPU usage, RAM 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 [ansible.cfg](https://github.com/Wambyi/ELK-Stack/blob/main/Ansible/ansible.cfg) file to /etc/ansible.
* Update the /etc/ansible/hosts file to include the virtual elk server as seen below.
* Run the playbook using the command: \*ansible-playbook <yml\_file>.yml and navigate to <public\_ip\_of\_elk>:5601/app/kibana to check that the installation worked as expected.

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

* Which file is the playbook? Where do you copy it?\_

The. yml are playbook files which can be placed in the ansible container to be deployed.

* Which file do you update to make Ansible run the playbook on a specific machine? How do I specify which machine to install the ELK server on versus which to install Filebeat on?

The host file can be updated to make Ansible run on a specific machine.

The file where the ELk server can be installed can be seen on the [elk.yml](https://github.com/Wambyi/ELK-Stack/blob/main/Ansible/elk.yml) and the (filebeat.yml)[Ansible/filebeat-playbook.yml] contains the file that installs the filebeat.

* Which URL do you navigate to in order to check that the ELK server is running? http://(elk-server-public-ip-address):5601/app/kibana

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

**Commands to install ansible, write and run playbook:**

* First ssh into Jumpbox-Provisioner: ssh admin@jump-box-ip
* Then update Jumpbox: sudo apt-get update
* Install Docker.io: sudo apt install docker.io
* Verify the status of docker: sudo systemctl status docker
* Start Docker if it is not running: sudo systemctl start docker
* Pull Ansible Docker Container : sudo docker pull cyberxsecurity/ansible
* Launch Ansible Container: docker run -ti cyberxsecurity/ansible:latest bash
* To View docker container list created: sudo docker ps
* Start the Docker container created: sudo docker start container\_name
* Activate shell on container: sudo docker attach container\_name
* To view ansible created: cd /etc/ansible
* Create yml file by using the extenton: my\_yml\_file.yml
* Run yml file: ansible-playbook my\_yml\_file.yml