This document describes the deployment of an automated ELK Stack and contains the following sections:

* Deployment
* Description of the Topology
* Access Policies
* ELK Configuration
  + Machines being Monitored
  + Beats in Use
* How to Use the Ansible Build
* Answers to Questions
* Miscellaneous
  + Filebeat Configuration
  + Metricbeat Configuration
  + Steps to Configure an Azure Virtual Network

**Important NOTE:**

This ELK server was created within an Azure virtual network. It is assumed at this point that a virtual network has been created with Ansible installed.

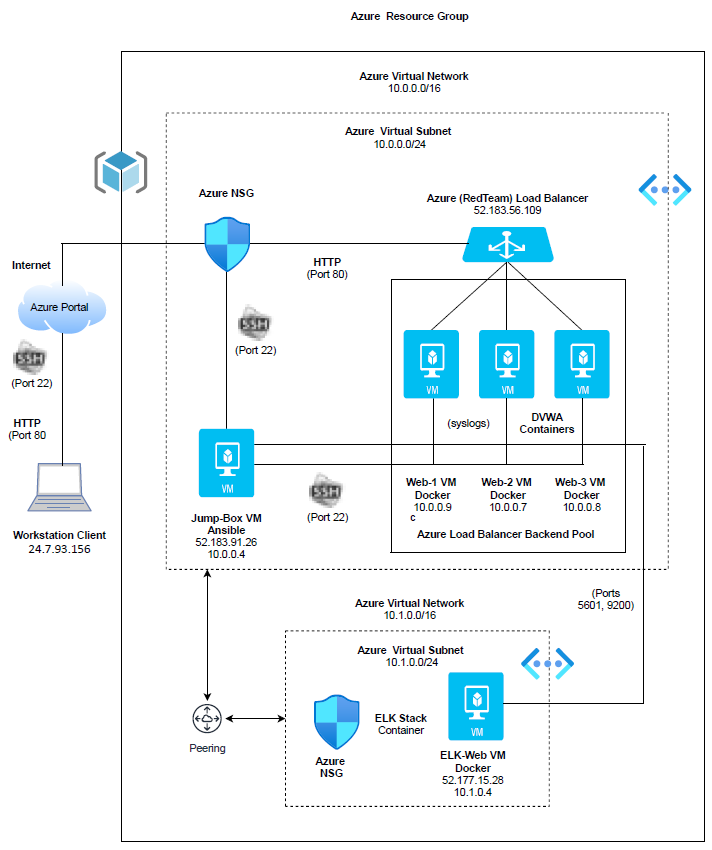
A more detailed description on how to configure and set up this ELK server along with an Azure virtual network is in the following document: **Documents/Azure Virtual Network Configuration**

**Deployment**

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

This diagram is located in:

**Diagrams/Azure\_Virtual\_Network - ELK\_Stack\_Diagram**



**Deployment (con’t)**

Playbooks and files used to create the ELK server:

elkplaybook.yml

filebeat-playbook.yml

metricbeat-playbook.yml

hosts

filebeat-config.yml

metricbeat-config.yml

These files have been tested and used to generate a live ELK deployment on Azure.

The Playbooks and files can be copied over by cloning them from GitHub.

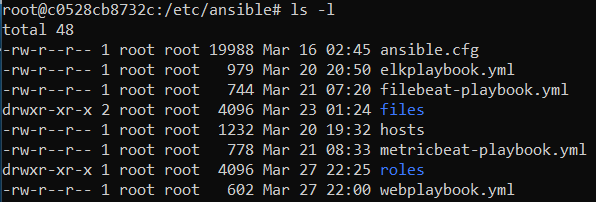
NOTE: Make sure you are in your Ansible container on the Jump Box.

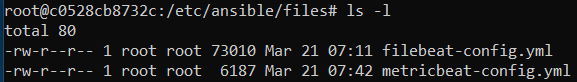
To clone from Git, follow the directions below:

* Change to a directory where you want to clone from Git.
  + Example: /root
* Clone repository from Git.
  + git clone https://github.com/ttambio/elk-stack-project.git
* Change directory to your Git repository.
  + Example: cd elk-stack-project
* Copy Playbooks and files to the appropriate folders in ‘/etc/ansible’.
  + cp elk-stack-project/Ansible/hosts /etc/ansible
  + cp elk-stack-project/Ansible/\*.cfg /etc/ansible
  + cp elk-stack-project/Ansible/\*.yml /etc/ansible
  + cp elk-stack-project/Ansible/\*config.yml /etc/ansible/files

**Deployment (con’t)**

The Playbooks and files should be located in the ‘/etc/ansible’ directory in the Ansible container of the Jump Box.





**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 and scalable, in addition to restricting inbound access to the network.

Adding a load balancer to our network will provide securities such as authentication and protect against distributed denial-of-service (DDoS) attacks.

The advantage of having a Jump Box is that adds an additional layer of protection, it acts like a gateway router to access our virtual machines. We can add hardening steps to our Jump Box such as adding a firewall or use a VPN to limit network access.

Integrating an ELK server allows users to easily monitor the vulnerable virtual machines for changes to the virtual machine file systems and system applications. It also allows us to watch system metrics such as CPU usage, SSH logins, etc.

The configuration details of each machine may be found below.

| Name | Function | IP Address | Operating System |

|:--------------:|:----------:|:----------:|:----------------:|

| Jump Box | Gateway | 10.0.0.4 | Linux |

| Web-1 | VM | 10.0.0.9 | Linux |

| Web-2 | VM | 10.0.0.7 | Linux |

| Web-3 | VM | 10.0.0.8 | Linux |

| Elk-Web-Server | Monitoring | 10.1.0.4 | Linux |

NOTE: We have a load balancer in front of our three Web VM servers.

**Access Policies**

The virtual 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 address: 52.183.91.26.

The Jump Box has access to the ELK server using IP address of 10.0.0.4.

Machines within the network can only be accessed by each other. The DVWA 1, DVWA 2 and DVWA 3 virtual machines send traffic to the ELK server.

Summary of access policies are listed below:

| Name | Publicly Accessible | Allowed IP Addresses |

|:--------------:|:---------------------:|:----------------------:|

| Jump Box | Yes | 52.183.91.26 |

| Web-1 | No | 10.0.0.4 |

| Web-2 | No | 10.0.0.4 |

| Web-3 | No | 10.0.0.4 |

| Elk-Web-Server | 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 we can automate our configurations by using an Ansible Playbook to execute on different systems with the same specifications.

The ELK playbook implements the following tasks:

* Installs docker, python3, docker module.
* Increases virtual memory; uses increased memory.
* Downloads and installs docker ELK container.
* Enables docker service on boot.

Below is the Playbook for the setup of the ELK server.

**Elk Configuration (con’t)**

---

- name: Config Web VM with Docker

hosts: elk

remote\_user: sysadmin

become: true

tasks:

- name: Install docker.io

apt:

update\_cache: yes

force\_apt\_get: yes

name: docker.io

state: present

- name: Install python3-pip

apt:

force\_apt\_get: yes

name: python3-pip

state: present

- name: Install docker module

pip:

name: docker

state: present

- name: Increase virtual memory

command: sysctl -w vm.max\_map\_count=262144

- name: Use increase memory

sysctl:

name: vm.max\_map\_count

value: '262144'

state: present

reload: yes

- name: Download and launch a docker elk container

docker\_container:

name: elk

image: sebp/elk:761

state: started

restart\_policy: always

published\_ports:

- 5601:5601

- 9200:9200

- 5044:5044

- name: Enable docker service on boot

systemd:

name: docker

enabled: yes

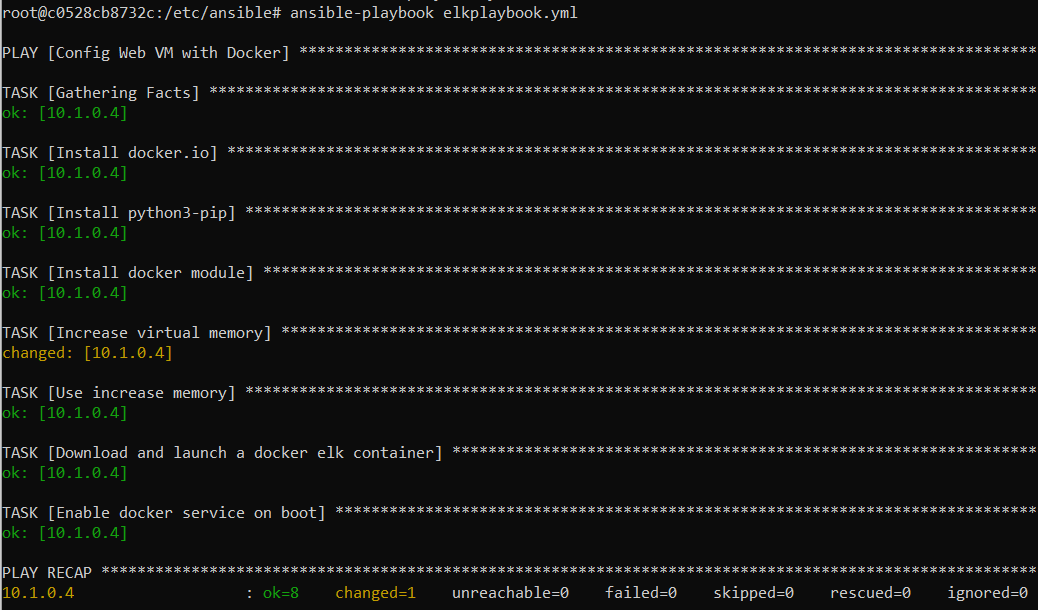
**Elk Configuration (con’t)**

To execute this ELK Playbook:

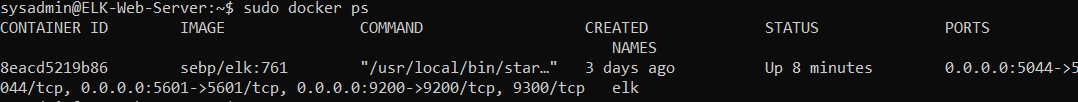
* Log into the Jump Box.
* Start and attach to a container.
* From /etc/ansible, execute the following command:

ansible-playbook elkplaybook.yml

Example of results:



The following screenshot displays the result of running `docker ps` after successfully configuring the ELK instance.



**Elk Configuration (con’t)**

**Machines being Monitored:**

This ELK server is configured to monitor the following machines:

| Name | IP Address |

|:-----:|:----------:|

| Web-1 | 10.0.0.9 |

| Web-2 | 10.0.0.7 |

| Web-3 | 10.0.0.8 |

**Beats in Use:**

We have installed the following Beats on these machines:

* Filebeat
* Metricbeat

Filebeat monitors, collects and forwards data log files and we can use it to collect logs from Docker containers.

Metricbeat collects and forwards metrics such as the operating system and services on a server and we can use it to analyze system CPU and memory or container performance.

**Elk Configuration (con’t)**

**Beats in Use: (con’t)**

The Playbook below is for the installation of Filebeat:

---

- name: Installing and launching Filebeat

hosts: webservers

become: yes

tasks:

- name: Download Filebeat deb

command: curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.0-amd64.deb

- name: Install Filebeat deb

command: dpkg -i filebeat-7.4.0-amd64.deb

- name: Drop in filebeat.yml

copy:

src: /etc/ansible/files/filebeat-config.yml

dest: /etc/filebeat/filebeat.yml

- name: Enable and configure system module

command: filebeat modules enable system

- name: Setup Filebeat

command: filebeat setup

- name: Start Filebeat service

command: service filebeat start

- name: Enable service Filebeat on boot

systemd:

name: filebeat

enabled: yes

**Elk Configuration (con’t)**

**Beats in Use: (con’t)**

The Playbook below is for the installation of Metricbeat:

---

- name: Installing Metricbeat

hosts: webservers

become: yes

tasks:

- name: Download Metricbeat

command: curl -L -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.4.0-amd64.deb

- name: Install Metricbeat deb

command: dpkg -i metricbeat-7.4.0-amd64.deb

- name: Drop in metricbeat config

copy:

src: /etc/ansible/files/metricbeat-config.yml

dest: /etc/metricbeat/metricbeat.yml

- name: Enable and configure docker module for Metricbeat

command: metricbeat modules enable docker

- name: Setup Metricbeat

command: metricbeat setup

- name: Start Metricbeat service

command: service metricbeat start

- name: Enable service Metricbeat on boot

systemd:

name: metricbeat

enabled: yes

**How to Use the Ansible Build**

**NOTE:**  A more detailed description on how to configure and set up this ELK server along with an Azure virtual network is in the following document: **Documents/Azure Virtual Network Configuration**

In order to use the ELK playbook, you will need to have an Ansible control node already configured.

For our deployment, we used Git Bash to access the Jump Box.

From Git Bash, SSH into the Jump Box and follow the steps below:

1. Log into the Jump Box.
2. Start and attach to an Ansible container.
3. Change directory to ‘/etc/ansible’.
4. Make sure the ‘hosts’ file is updated to include the ELK server.

$ cat hosts

.

.

[webservers]

## alpha.example.org

## beta.example.org

## 192.168.1.100

## 192.168.1.110

10.0.0.9 ansible\_python\_interpreter=/usr/bin/python3

10.0.0.7 ansible\_python\_interpreter=/usr/bin/python3

10.0.0.8 ansible\_python\_interpreter=/usr/bin/python3

[elk]

10.1.0.4 ansible\_python\_interpreter=/usr/bin/python3

**NOTE:** We have included the latest python version.

1. Run the Playbooks.
   * Change directory back to ‘/etc/ansible’.
   * Execute the Playbooks with the following commands:

ansible-playbook elkplaybook.yml

ansible-playbook filebeat-playbook.yml

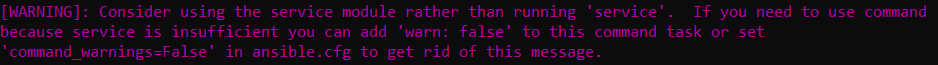
ansible-playbook metricbeat-playbook.yml

See below for warning messages that might appear when executing the Playbooks for Filebeat and Metricbeat.

**How to Use the Ansible Build (con’t)**

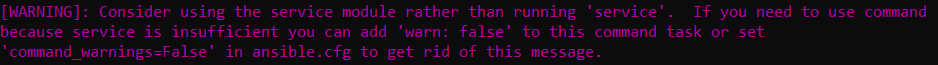
From Filebeat:

You will receive the following valid warning for both the ‘Download Filebeat deb’ and ‘Start Filebeat service’ tasks.



From Metricbeat:

You will receive the following valid warning for both the ‘Download Metricbeat’ and ‘Start Metricbeat service’ tasks.

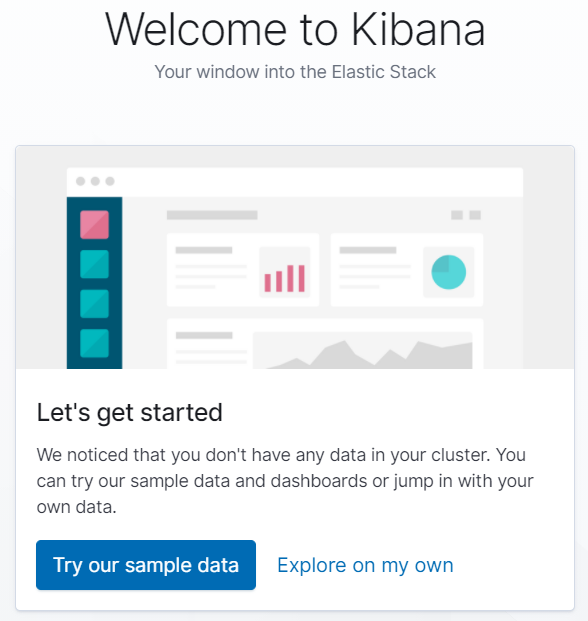


1. Verify that the ELK Server is up.
   * From a browser, navigate to:

http:/13.77.65.19:5601/app/kibana

NOTE: The IP address is the public IP address of your ELK virtual machine.

The website should come up with the Kibana home page.



**Answers to Questions**

1. Which file is the ELK playbooks? Where do you copy it?

There are three playbooks that have been copied from Git into the directory ‘/etc/ansible’ of the Ansible container.

elkplaybook.yml

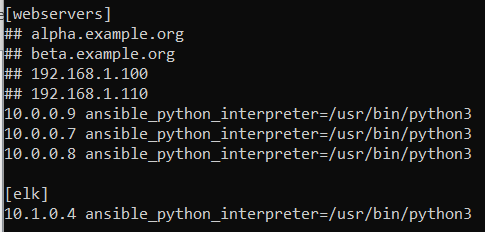
filebeat-playbook.yml

metricbeat-playbook.yml

1. 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?

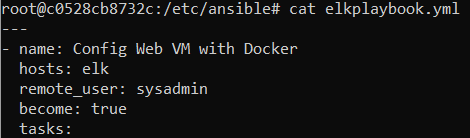
Update the ‘hosts’ and the file from ‘/etc/ansible’.

Example:



Within each Playbook, update the group, which is the ‘hosts’ name, to reflect which server to execute on.

Example from the ELK Playbook:



**Answers to Questions (con’t)**

1. Which URL do you navigate to in order to check that the ELK server is running?

http:/13.77.65.19:5601/app/kibana

NOTE: The IP address is the public IP address of your ELK virtual machine.

**Miscellaneous**

**NOTE:**  A more detailed description on how to configure and set up this ELK server along with an Azure virtual network is in the following document: **Documents/Azure Virtual Network Configuration**

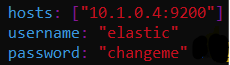
**Filebeat Configuration:**

To copy and update a Filebeat configuration file:

* Change directory to ‘/etc/ansible/files’.
  + NOTE: Create a ‘files’ subdirectory if it does not exist.
* Copy the Filebeat configuration file from the following site into a file called ‘filebeat-config.yml’ using ‘curl’ into your Ansible container.

https://gist.githubusercontent.com/slape/5cc350109583af6cbe577bbcc0710c93/raw/eca603b72586fbe148c11f9c87bf96a63cb25760/Filebeat

* Edit the Filebeat configuration file accordingly.
  + - On line #1106, replace the IP address with the ELK VM private IP address.



* + - Leave the default port as ‘9200’
    - Leave the username/password defaulted to ‘elastic’/’changeme’.
    - On line #1806, replace the IP address with the ELK VM private IP address.



* + - * Leave the default port as ‘5601’.

**Miscellaneous (con’t)**

**Metricbeat Configuration:**

To copy and update a Metricbeat configuration file:

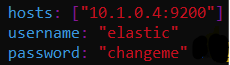
* Change directory to ‘/etc/ansible/files’.
  + NOTE: Create a ‘files’ subdirectory if it does not exist.
* Copy the Metricbeat configuration file from the following site into a file called ‘metricbeat-config.yml’ using ‘curl’ into your Ansible container.

https://gist.githubusercontent.com/slape/58541585cc1886d2e26cd8be557ce04c/raw/0ce2c7e744c54513616966affb5e9d96f5e12f73/metricbeat

* + Edit the Metricbeat configuration file accordingly.
    - On line #62, replace the IP address with the ELK VM private IP address.



* + - * Leave the default port as ‘5601’.
    - On #95, replace the IP address with the ELK VM private IP address,



* + - Leave the default port as ‘9200’
    - Leave the username/password defaulted to ‘elastic’/’changeme’.

**Miscellaneous (con’t)**

**Steps to Configure an Azure Virtual Network:**

The following is an outline of the steps that need to be taken in order to create an Azure Virtual Network with five virtual machines (Jump Box, four Web (DVWA/ELK) servers and Ansible installed.

A more detailed step-by-step instructions can be found in:

**Documents/Azure Virtual Network Configuration**

The steps are broken down into the following sections:

**Azure Virtual Network Configuration**

* Azure Virtual Network Diagram
* Set-up Azure Account
* Create Resource Group
* Create Azure Virtual Network
* Create Security Group
  + Create Network Security Group (NSG)
  + Create Inbound Security Rule (Block)
* Create Jump Box Virtual Machine
  + Create SSH key
  + Create Virtual Machine
* Create Three Web Virtual Machines
* Create Inbound Security Rule (Client Port 22)
  + Find Client Public IP
  + Create Inbound Rule
  + Connect to Jump Box
* Create Containers
  + Install Docker
  + Create Inbound Security Rule (Jump Box Port 22)
* Create Container SSH Key
  + Start/Connect Container
  + Create SSH Key
  + Reset Web Virtual Machine Password
  + Connect to Web Containers
* Update Configuration Files
  + Hosts File
  + Configuration File
  + Test Group Connectivity
* Create Ansible Playbook
  + Create Playbook
  + Execute Playbook
  + Test DVWA Container

**Miscellaneous (con’t)**

**Steps to Configure an Azure Virtual Network: (con’t)**

This documentation is broken down into the following sections: (con’t)

**Azure Virtual Network Configuration (con’t)**

* Configure Load Balancer
  + Create Load Balancer
  + Install Health Probe
  + Update Load Balancer Settings
* Set-up Security Configuration
  + Create Load Balancing Rule (Port 80)
  + Create Inbound Security Rule (Client Port80)
  + Remove Inbound Security Rule (Block)
  + Connect to DVWA Website

**ELK Stack Configuration**

* Create Second Virtual Network
  + Create Virtual Network
  + Create Peer Network Connection
* Create Fourth Web (ELK) Virtual Machine
  + Copy SSH Key
  + Create Virtual Machine
  + Connect to Virtual Machine
* Update Hosts File
* Create ELK Stack Playbook
  + Create Playbook
  + Execute Playbook
  + Launch Container
* Create Inbound Security Rule (Client Port 5601)
  + Create Inbound Security Rule
  + Connect to Kibana
* Filebeat
  + Create Filebeat Configuration File
  + Create Filebeat Playbook
  + Execute Filebeat Playbook
* Metricbeat
  + Create Metricbeat Configuration File
  + Create Metricbeat Playbook
* Kibana – Filebeat
* Kibana - Metricbeat

**Miscellaneous (con’t)**

**Steps to Configure an Azure Virtual Network:**

This documentation is broken down into the following sections: (con’t)

**Miscellaneous**

* GitHub Repository
* Locate IP Addresses
* SSH keys
* Machine Login
* Azure Portal - Screenshots
* Explore Kibana
* Pen Testing