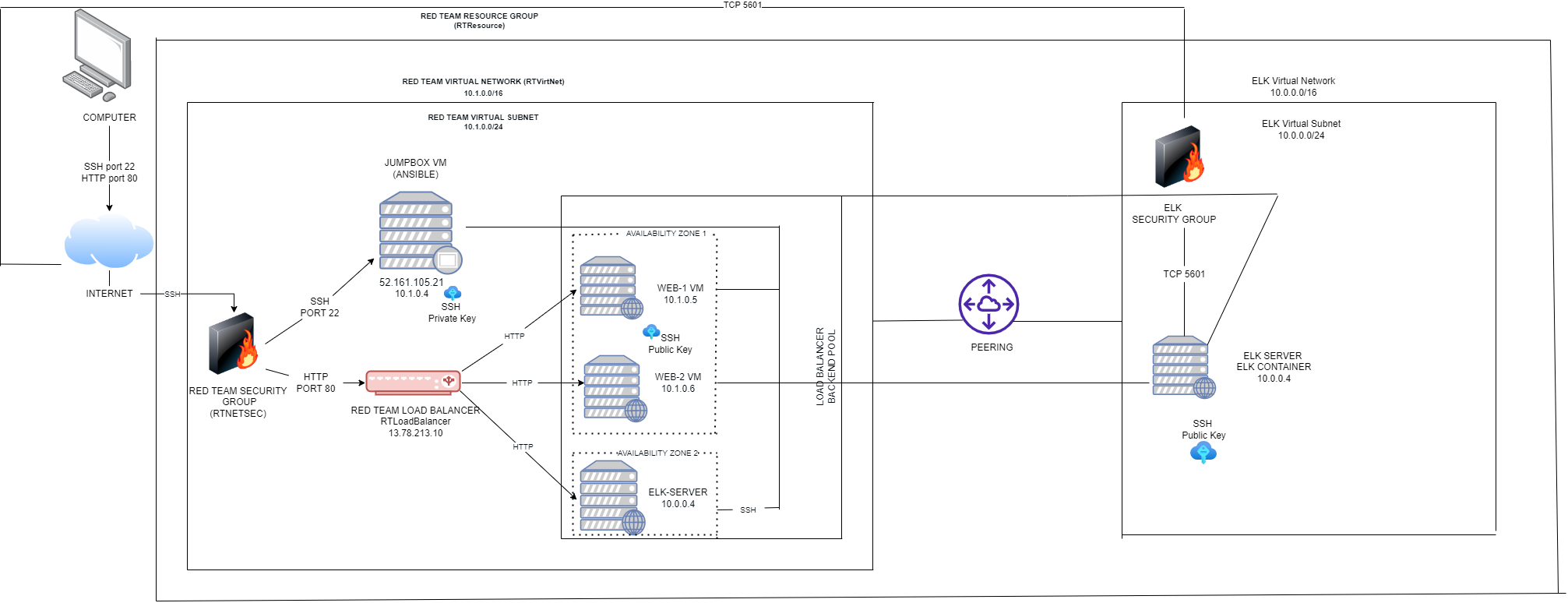
=====================================

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

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 functionable, in addition to restricting unwanted traffic to the network.

Load Balancers are able to reroute traffic from one server to another in the event that a server becomes unavailable due to a DDoS attack or other issues.

The advantage of a Jump Box Provisioner includes preventing other Azure virtual machines from exposure via public IP Address. In this Project, I restricted what IP Addresses are able to communicate with the Jump Box Provisioner.

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

- Filebeat watches for log files and locations which the user specifies, collects log events and forwards them to be indexed to Logstash or Elasticsearch.

- Metricbeat collects the metrics and statistics and sends them to Logstash or Elasticsearch.

The configuration details of each machine may be found below.

| NAME | FUNCTION | IP ADDRESS | OPERATING SYSTEM |
| --- | --- | --- | --- |
| Jump Box | Gateway | 10.1.0.4 52.161.105.21 | Linux |
| Web-1 | UbuntuServer | 10.1.0.5  13.78.213.10 | Linux |
| Web-2 | UbuntuServer | 10.1.0.6  13.78.213.10 | Linux |
| ELKServer | UbuntuServer | 10.0.0.4 | Linux |

### Access Policies

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

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

- My Public IP Address through TCP 5601.

Machines within the network can only be accessed by my Public IP Address and JumpBox SSH.

- In this project, I allowed JumpBoxProvisioner IP Address 10.1.0.4.

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

| NAME | PUBLICLY ACCESSIBLE | ALLOWED IP ADDRESSES |
| --- | --- | --- |
| Jump Box | Yes | 73.170.31.143 via SSH 22 |
| Web-1 | No | 10.1.0.4 via SSH 22 |
| Web-2 | No | 10.1.0.4 via SSH 22 |
| ELKServer | No | My Public IP Address via TCP 5601 |

### Elk Configuration

Ansible was used to automate configuration of the ELK machine. No configuration was performed manually, which is advantageous because...

- Ansible allows users to deploy multiple applications utilizing a YML Playbook.

- Ansible’s playbooks are very user friendly and there is no requirement to write code.

The playbook implements the following tasks:

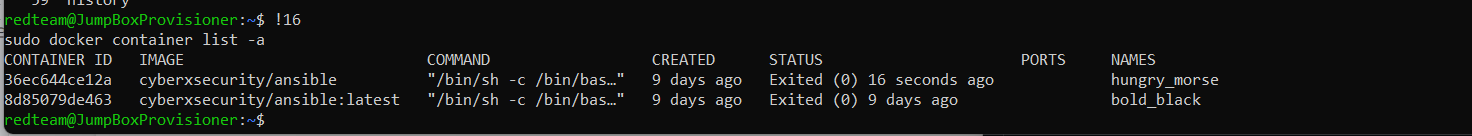
- Install docker.io

- Install python3-pip

- Download and launch ELK Docker Container

- Made Ports 5044, 5601, and 9200 available

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



### Target Machines & Beats

This ELK server is configured to monitor the following machines:

- Web-1: 10.1.0.5

- Web-2: 10.1.0.6

We have installed the following Beats on these machines:

- Filebeat

- Metricbeat

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

- Filebeat is used to collect log files from specific places such as: Microsoft Azure, web servers, etc.

-Metricbeat is used to collect the statistics and metrics of virtual machines and operating systems, such as memory and data from services running on the server.

### 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 YML file to the ansible folder.

- Update the config file to include the remote users and ports you wish to attach.

- Run the playbook by using the following:

ansible-playbook filebeat-playbook.yml

ansible-playbook metricbeat-playbook.yml

and then navigate to https:/[your.VM.IP]:5601/app/kibana to check that the installation worked as expected.

- The file for the Ansible playbook is ansible-playbook.yml. The file for the Filebeat playbook is filebeat-playbook.yml and the file for the Metricbeat playbook is metricbeat-playbook.yml. I copied these playbooks into /etc/ansible/.

- In order to update Ansible and allow it to run the playbook on a specific machine, you must update the /etc/ansible/hosts file and include the IP Address of the Virtual Machine you wish to include.

- In order to specify which machine to install the ELK server on versus which machine to install Filebeat on, the IP Addresses of the webservers/Virtual Machines will be used.

- In order to ensure that the ELK Server is running, I used

https:/http://104.42.254.86:5601/app/kibana

The following shows the specific commands a user will run in order to download the playbook and configuration files:

ssh-keygen

sudo cat .ssh/id\_rsa.pub

ssh [your username]@[jumpboxprovisioner IP address]

sudo docker container list -a

sudo docker start [container name]

sudo docker attach [container name]

cd /etc/ansible

nano /etc/ansible/hosts

nano ansible.config

nano pentest.yml

ansible-playbook [file name]

sudo apt install docker.io

sudo service docker start

sudo systemctl status docker

sudo systemctl start docker

curl -L -O [web location of the file]

dpkg -i [file name]

http:/[IP address]:5601/app/kibana

nano filebeat-config.yml

nano filebeat-playbook.yml

nano metricbeat-config.yml

nano metricbeat-playbook.yml