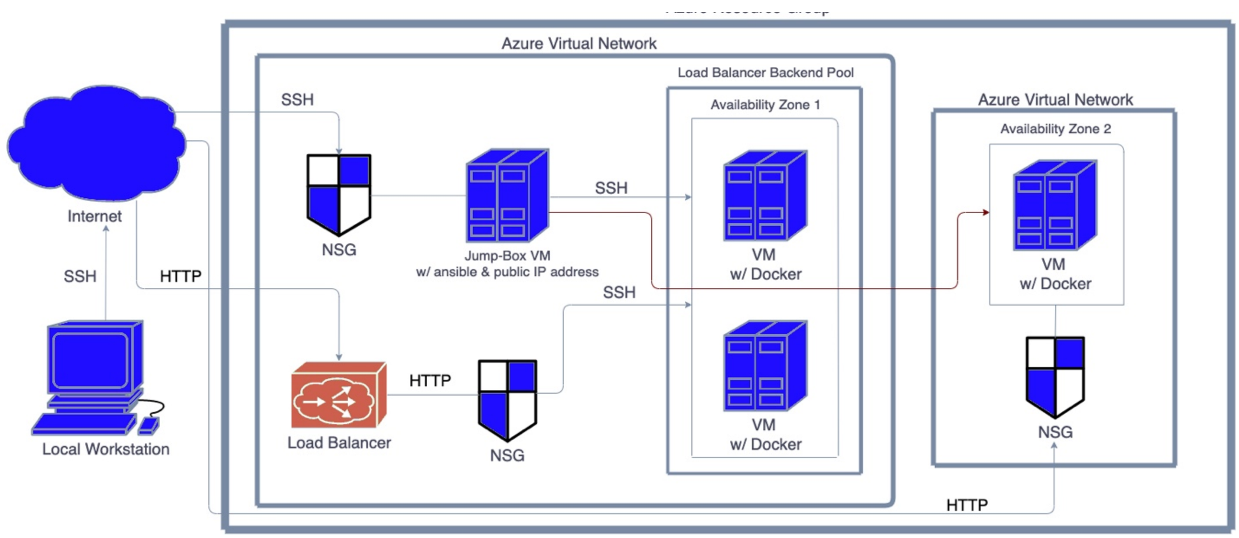
Karl Walz

Project #1: Creating and configuring an ELK server

Azure Virtual Network using a JumpBox VM with Ansible, load balancer, Network Security Group, and 3 VMs using Docker:



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

The main purpose of this network is to expose a load-balanced and monitored instance of DVWA.

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

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

The configuration details of each machine are found in the table below:

Name	Function	IP Address	Operating System
Jump Box	Gateway	10.0.0.4	Linux

Name	Function	IP Address	Operating System
Jump Box	Gateway	Public IP 68.2.91.221	Linux
Web-1	DVWA 1	10.0.0.7	Linux
Web-2	DVWA 2	10.0.0.8	Linux
ELK	Security	10.1.1.4	Linux

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: 68.2.91.221

Machines within the network can only be accessed by the Jump-Box-Provisioner machine with Ansible container. IP address: 137.135.63.210

Commands:

```
‘ssh azadmin@137.135.63.210’

‘sudo docker start cf331a2fb986’

‘sudo docker attach cf331a2fb986’
```

Access Policy Table:

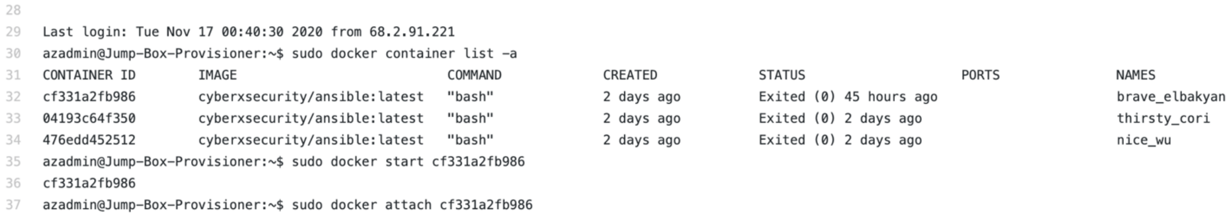
Name	Publicly Accessible	Allowed IP Addresses
Jump Box	Yes	workstation public IP
Web-1	No	10.0.0.4, 10.0.0.8
Web-2	No	10.0.0.4, 10.0.0.7
Load Balancer	No	workstation public IP
ELK Server	No	68.2.91.221
Kibana	Yes	workstation public IP

The ansible configuration file (ansible.cfg) and install-elk playbook (install-elk.yml) allows for automation of the Azure Virtual Network startup. The configuration file and ELK playbook are available in this GitHub repository for examination.

The playbook implements the following tasks:

Install docker.io
Install python3-pip3
Install Docker module
Increase virtual memory
Use more memory
Download and launch a Docker ELK container

Terminal screenshot listing and attaching the docker container:



```

Last login: Tue Nov 17 00:40:30 2020 from 68.2.91.221
azadmin@Jump-Box-Provisioner:~$ sudo docker container list -a
CONTAINER ID        IMAGE                             COMMAND             CREATED             STATUS                      PORTS               NAMES
cf331a2fb986        cyberxsecurity/ansible:latest     "bash"              2 days ago          Exited (0) 45 hours ago                         brave_elbakyan
04193c64f350        cyberxsecurity/ansible:latest     "bash"              2 days ago          Exited (0) 2 days ago                           thirsty_cori
476edd452512        cyberxsecurity/ansible:latest     "bash"              2 days ago          Exited (0) 2 days ago                           nice_wu
azadmin@Jump-Box-Provisioner:~$ sudo docker start cf331a2fb986
cf331a2fb986
azadmin@Jump-Box-Provisioner:~$ sudo docker attach cf331a2fb986
```

The ELK server created is configured to monitor the following machines:

10.0.0.7

10.0.0.8

Two different Beats were installed on the Azure Virtual Network:

1. Filebeat is the logging agent installed on the machine generating log files, tailing them, and forwarding the data to Logstash for advanced processing or Elasticsearch for indexing. We can view these logs in Kibana. For example, Filebeat allows us to view how many times we used SSH to connect to a VM. We will be able to see the time and hostname of the source.
2. Metricbeat collects metrics from the operating system and services running on the server. It takes the metrics and statistics that it collects and ships them to Elasticsearch and Logstash. For example, using Metricbeat, we can see memory usage, inbound and outbound traffic speeds etc.

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 install-elk.yml file to path /etc/ansible inside the Ansible container.

Update the hosts file to include the IP of the ELK server.

Run the playbook, and navigate to https://23.96.99.53:5601/app/kibana to check that the installation worked as expected.

The ansible.cfg and install-elk.yml files are also included in this repository.

A full Terminal input/output log is additionally included in this repository: https://github.com/kwalz5504/ELK/blob/main/11-18-2020.txt