**Summary:**

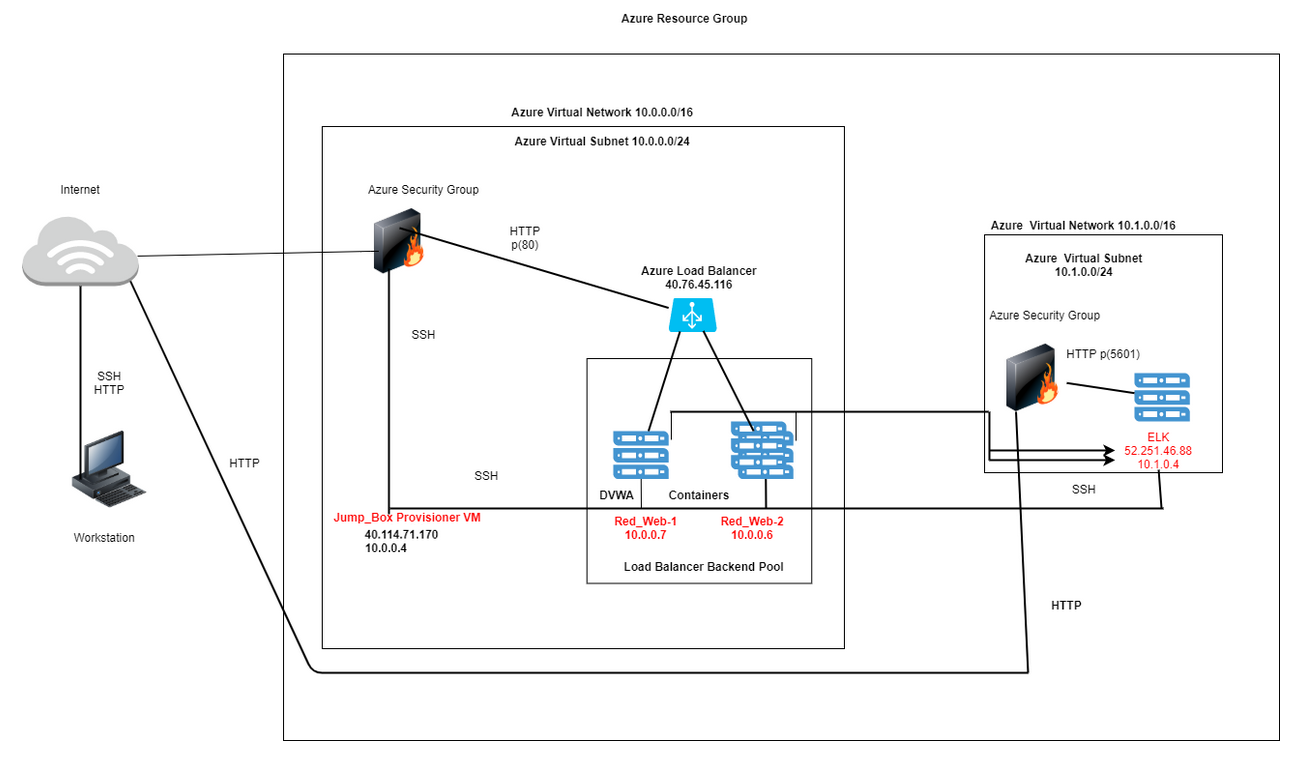
In this project; we were directed to create a load balanced web application with monitoring. The “public facing” web app was provided by Damn Vulnerable Web Application, and my monitoring solution was in the form of an ELK stack with Filebeat & Metricbeat (which I refer to as heartbeat services later). While using a web app that’s pre-built and intended to have some level of vulnerability goes against my instincts, I found this exercise a useful and quite fascinating experience with the Elastic/Logstash/Kibana stack monitoring solution. Incidentally, I also found a ton of value in exploring network topology and adminning a virtual load balancer for the first time.

As the DVWA server is open to traffic, it allows a mechanism for the heartbeat services to collect traffic and pass data to the ELK server. Setting up and configuring the ELK server was a great experience as well; it can collect so much data that overload is inevitable, or it can be so choked down it’s practically useless. Once I had it running, I immediately saw the value in the ELK stack as I added rules one by one. Even with a web app like DVWA that is practically begging to be exploited or pwned, ELK stacks and load balancing add an incredible value in how traffic is handled and recorded, and even adds levels of security in themselves and their intrinsic value.

Below is my explanation.

## ELK Stack Deployment in Azure Cloud

The files associated with this document 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, portions of the configuration file may be used to install only certain pieces of it, such as only Filebeat.

* ELK stack VM install script w/ Docker
* Filebeat installation playbook

**This document contains the following details:**

* Description of the Topology
* Access Policies
* ELK Configuration
  + Beats in Use
  + Machines Being Monitored

**Description of the Topology**

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 exposing the backend servers directly to the external internet network
* Load Balancers contribute to the Availability aspect of security in regards to the CIA Triad by support of redundancy, network performance.
* *JumpBox* used for System Administration and secured external network access.
* The advantage of a JumpBox is the origination point for launching Administrative Tasks. This ultimately sets the JumpBox as a Secure Admin Workstation). All Administrators when conducting any management Tasks will be required to connect to the JumpBox. The Jumpbox is configured for restrictive access providing a more secure environment.
* *Integrating an ELK server allows users to easily monitor the vulnerable VMs for changes to the filesystem and system resources.*
  + Filebeat watches for log files or locations that are configured, collecting log events and forwards them to Elasticsearch or Logstash for indexing
  + Metricbeat records metric and statistical data from the operating system and from the services running on the host servers.

The configuration details of each machine may be found below.

| **Name** | **Function** | **IP Address** | **Operating System** |
| --- | --- | --- | --- |
| Jump-Box-Provisioner | Access Point/DMZ | 40.114.71.170 10.0.0.4 | Linux Ubuntu 18.04 |
| Red-Web-1 | Docker-DVWA | 10.0.0.6 | Linux Ubuntu 18.04 |
| Red-Web-2 | Docker-DVWA | 10.0.0.7 | Linux Ubuntu 18.04 |
| ELK | ELK Stack | 52.251.46.88 10.1.0.4 | Linux Ubuntu 18.04 |

**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:

* whitelisted IP address: Admins personal IP address Machines within the network can only be accessed via SSH.
* Which machine did you allow to access your ELK VM? What was its IP address? Jump-Box-Provisioner 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** |
| --- | --- | --- |
| Jump-Box-Provisioner | Yes | Admin Personal IP |
| Red-Web-1 | No | 10.0.0.4 |
| Red-Web-2 | No | 10.0.0.4 |
| ELK-Stack | Yes | 40.114.71.170 10.0.0.4 |

**Elk Configuration**

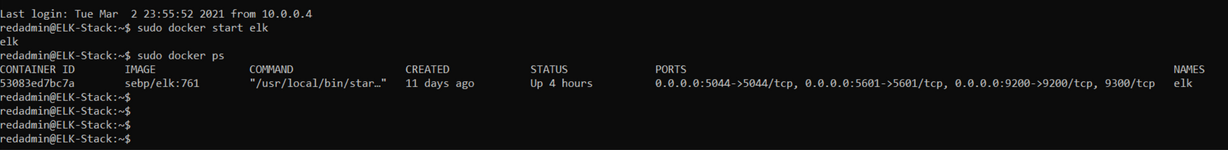
Docker was used to automate configuration of the ELK machine. No configuration was performed manually.

* The main advantage that Docker brings is automation of and simplification of repetitive, complex, and tedious operations a systems admin.

-The playbook implements the following tasks:

* Install: docker.io
* Install python-pip
* Install docker module
* Increase Memory Use: sysctl -w vm.max\_map\_count=262144
* Download and Launch a docker elk container

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:

| **VM Host** | **Private IP Addresses** | **Filebeat Installed** | **Metricbeat Installed** |
| --- | --- | --- | --- |
| Red-Web-1 | 10.0.0.7 | Yes | Yes |
| Red-Web-2 | 10.0.0.6 | Yes | Yes |

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

* Filebeat is a lightweight shipper for forwarding and centralizing log data. Filebeat monitors log files or locations you specify, collects log events, and forwards them either to Elasticsearch or Logstash for indexing.
* Metricbeat collects metrics from the operating system and from services running on the server. Metricbeat then takes the metrics and statistics that it collects and ships them to the output that you specify.