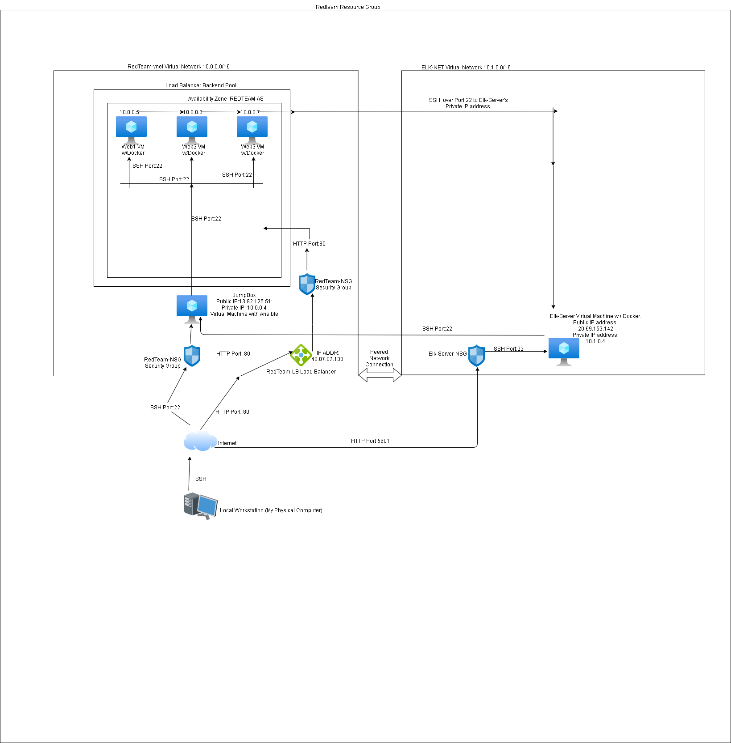
The purpose of this project was to create an ELK Stack resource in the Microsoft Azure cloud computing platform. An ELK stack is a virtual framework allowing for the deployment and integration of resources such as Kibana and Beats. Kibana that gives users the access and tools to analyze and see network/resource data. Below, see a network diagram of the entire setup. [[1]](#footnote-1)



The following .yml files were used to create and install the ELK Stack, and the Kibana and Filebeat resources:

1. ELK\_Server.yml
2. ansible.cfg
3. filebeat-config.yml
4. filebeat-playbook.yml
5. pentest.yml

Further, this description will contain the following:

1. An explanation of the of the network topology.
2. The configured access policies on all the resources.
3. The ELK parameters with the beats used and the machines being monitored.

The topology of this network was configured to use a load-balanced and monitored website. Said website is the Damn Vulnerable Web Application. The load-balancer will allow the web application to be available, even in the face of high network traffic. Also, given the rules assigned to said load balancer, certain traffic will be prohibited. From a security perspective, since load-balancers distribute network traffic, it can mitigate Denial of Service Attacks. Then, the Elk server provides diagnostics capabilities for the web1 to web3 virtual machines. The program Filebeat monitors and relays data about the file systems of the virtual machines. On the following page, is a table giving basic information about each of the web virtual machines, the JumpBox and the Elk server.[[2]](#footnote-2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Function | Availability Zone | IP Address | Operating System |
| JumpBox | Gateway | RedTeam | 13.82.125.51/ 10.0.0.4 | Linux |
| web1 VM w/docker | Web Machine | RedTeam | 10.0.0.5 | Linux |
| web2 VM w/docker | Web Machine | RedTeam | 10.0.0.6 | Linux |
| web3 VM w/docker | Web Machine | RedTeam | 10.0.0.7 | Linux |
| ELK-Server w/docker | Monitoring | RedTeam | 20.69.153.142/10.1.0.4 | Linux |

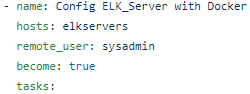
[[3]](#footnote-3)

Next will be a discussing of access policies. Each of the web virtual machines do not have internet access. Also, the web virtual machines can only be logged into via SSH over port 22 from the JumpBox virtual machine. For the Elk server, it can only be logged onto over a web browser for user’s workstation public IP address and/or the JumpBox virtual machine. [[4]](#footnote-4) On the following page will be a table layout of the access policies.

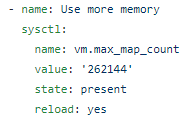
|  |  |  |
| --- | --- | --- |
| Name | Publicly Accessible | Allowed IP Addresses |
| JumpBox | No | Workstation’s Public IP Address |
| web1 VM w/docker | Yes (Through the Load Balancer) | 40.87.87.133 (RedTeam-LB-IP) |
| web2 VM w/docker | Yes (Through the Load Balancer) | 40.87.87.133 (RedTeam-LB-IP) |
| web3 VM w/docker | Yes (Through the Load Balancer) | 40.87.87.133 (RedTeam-LB-IP) |
| ELK-Server w/docker | No | 40.87.87.133 (RedTeam-LB-IP) |

[[5]](#footnote-5)

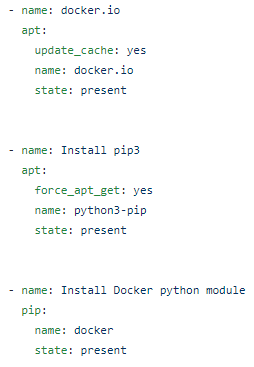
Next, is how the Elk server itself was configured. Using the Ansible program and a .yml configuration file, the tasks associated with creating the Elk virtual machine were automated. Additionally, it was faster to use this method. The figure below shows a part of the .yml code that specifies the virtual machines to use and an alternate remote user, if applicable.



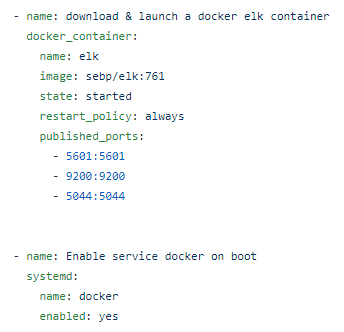
The next figure below specifies the machine to increase its disk space.



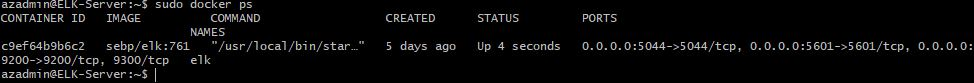
The figure on the following page shows the code for installing docker.io, python-pip and docker.



The last segment of code, below, downloads and launches the Elk container with all of its needed configuration parameters.

[[6]](#footnote-6)

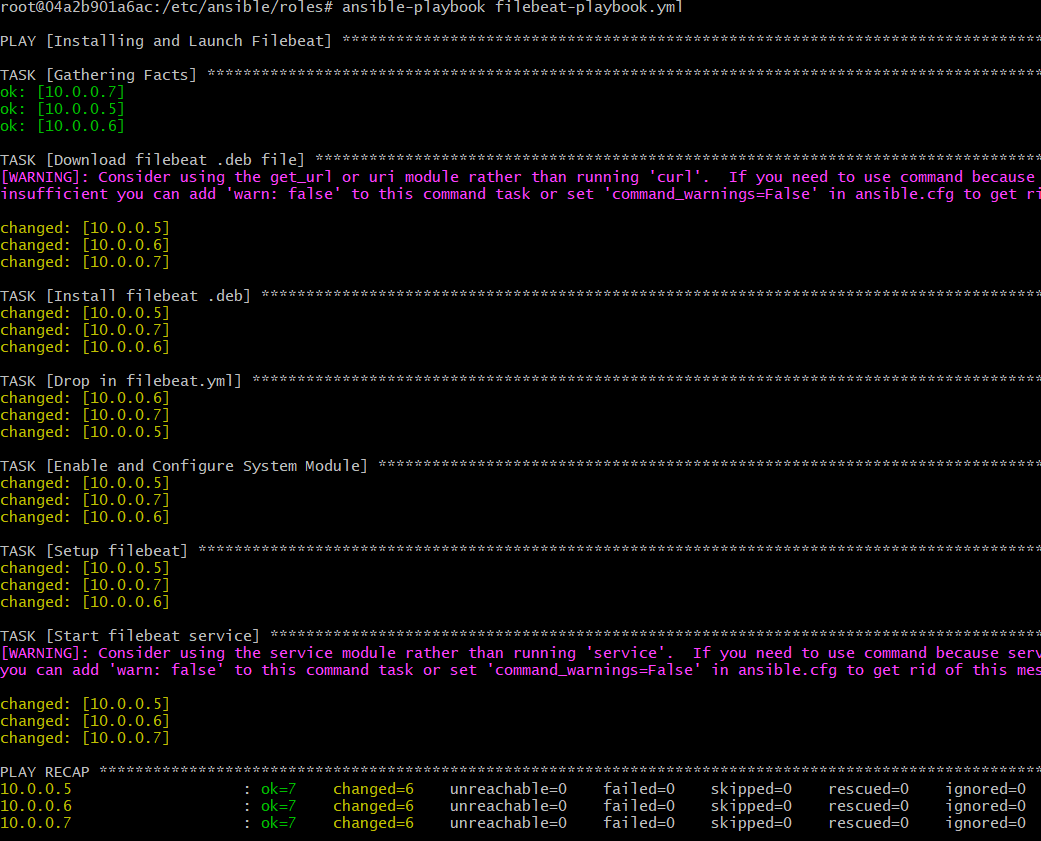
Below is the result of executing the above code.



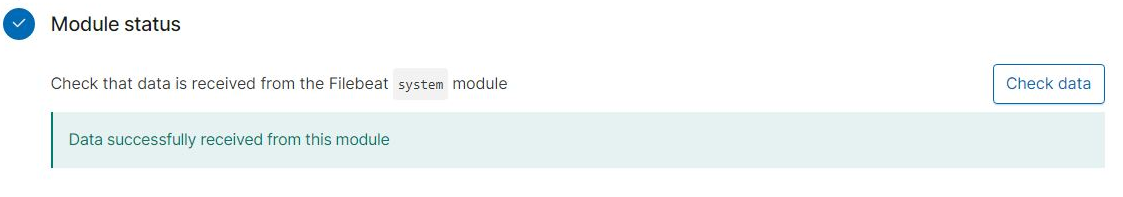
Next, the Elk machine was setup to collect data for the below machines:

1. Web1 w/docker @10.0.0.5
2. Web2 w/docker @10.0.0.6
3. Web3 w/docker @10.0.0.7

Next, was to setup and install Filebeat, the monitoring tool. To do this, the Jumpbox virtual machine was used to download the Filebeat .deb file. Then, using the “scp” Linux command, the installation file was copied to each of the web virtual machines. Finally, using the filebeat-playbook.yml in conjunction with ansible, the program was installed and enabled. The figure below will show the successful completion on this process.



With the above complete it is now possible to access the monitoring tools on kibana using the link <http://ELK_IP_ADDR:5601/app/kibana>. From the home page navigate to Add log data>System logs>DEB. The click on “Check data”. Below is what happens if everything was configured properly.

[[7]](#footnote-7)

1. <https://logz.io/learn/complete-guide-elk-stack/>

   https://github.com/checiches81/Cybersecurity/tree/master/Cloud\_Security [↑](#footnote-ref-1)
2. https://github.com/checiches81/Cybersecurity/tree/master/Cloud\_Security [↑](#footnote-ref-2)
3. https://github.com/checiches81/Cybersecurity/tree/master/Cloud\_Security [↑](#footnote-ref-3)
4. https://github.com/checiches81/Cybersecurity/tree/master/Cloud\_Security [↑](#footnote-ref-4)
5. https://github.com/checiches81/Cybersecurity/tree/master/Cloud\_Security [↑](#footnote-ref-5)
6. https://github.com/checiches81/Cybersecurity/tree/master/Cloud\_Security [↑](#footnote-ref-6)
7. https://github.com/checiches81/Cybersecurity/tree/master/Cloud\_Security [↑](#footnote-ref-7)