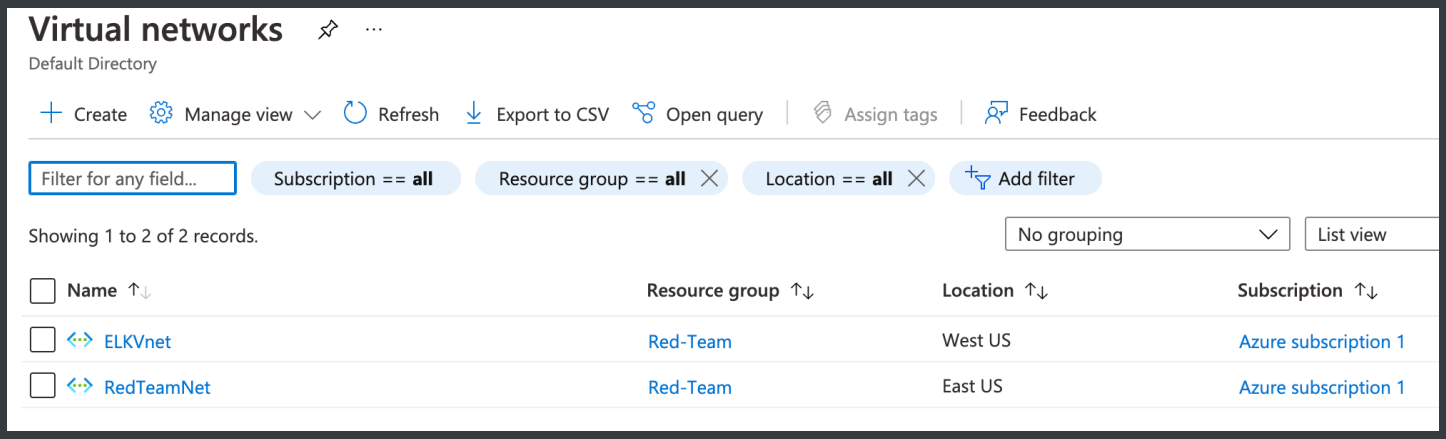
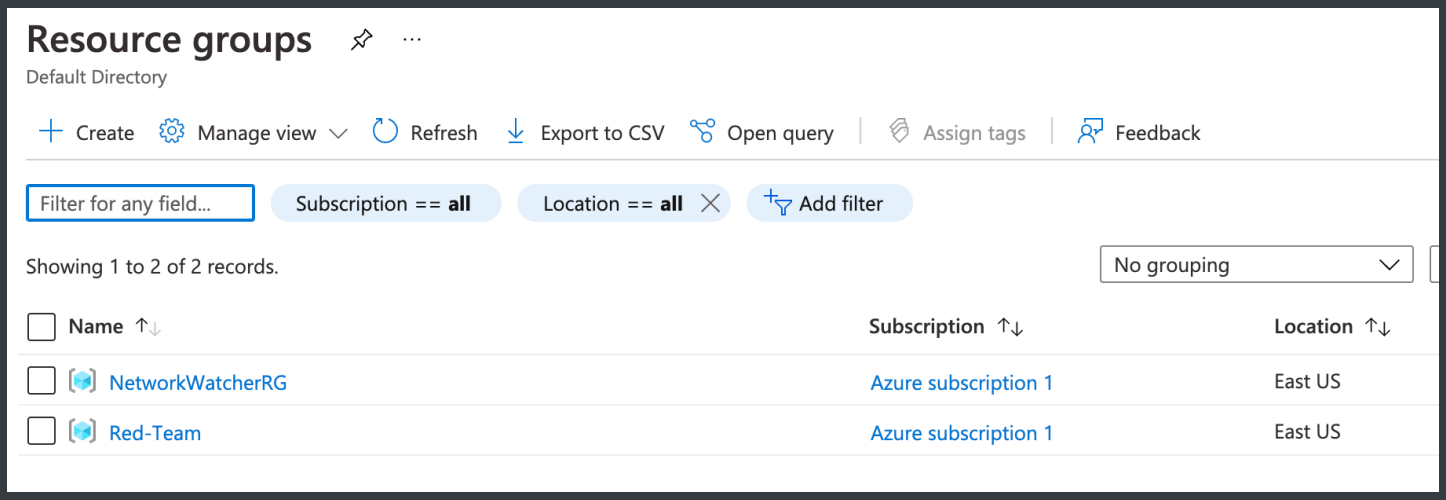
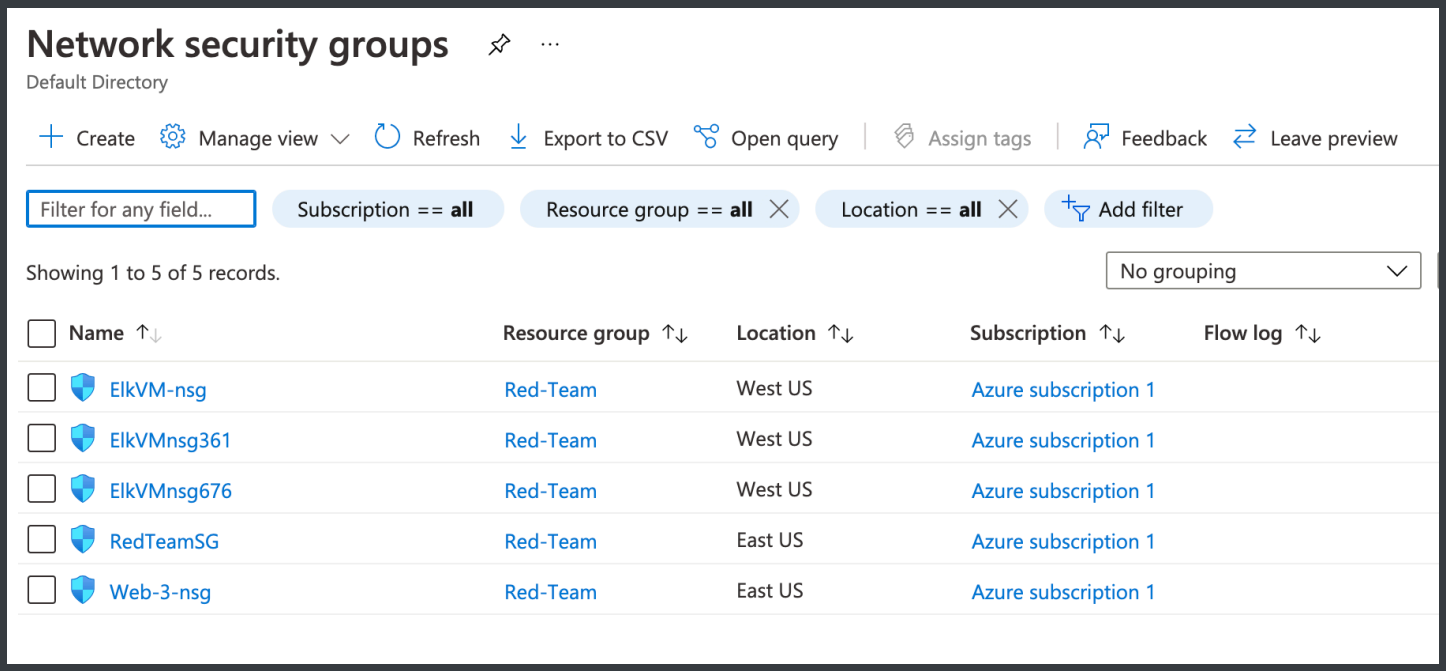
*AUTOMATED ELK STACK DEPLOYMENT

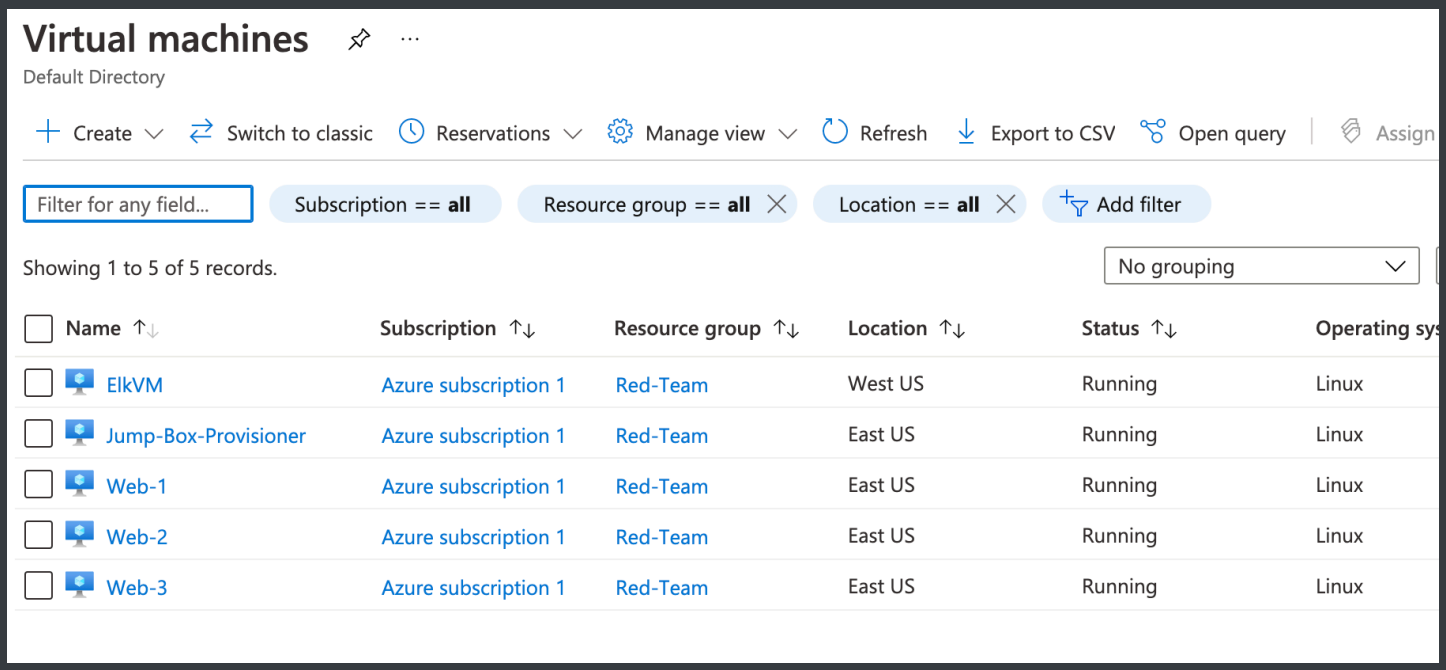
For this cloud environment project I was responsible for creating virtual networks, resource groups, network security groups, and virtual machines (three web VMs, a jumpbox, and an Elk server) that were distributed along with a load balancer on Microsoft Azure. I deployed an Elk Stack VM with Kibana using an ansible container which I also developed then installed Filebeat and Metricbeat to monitor vulnerable web servers that my Red Team was using.

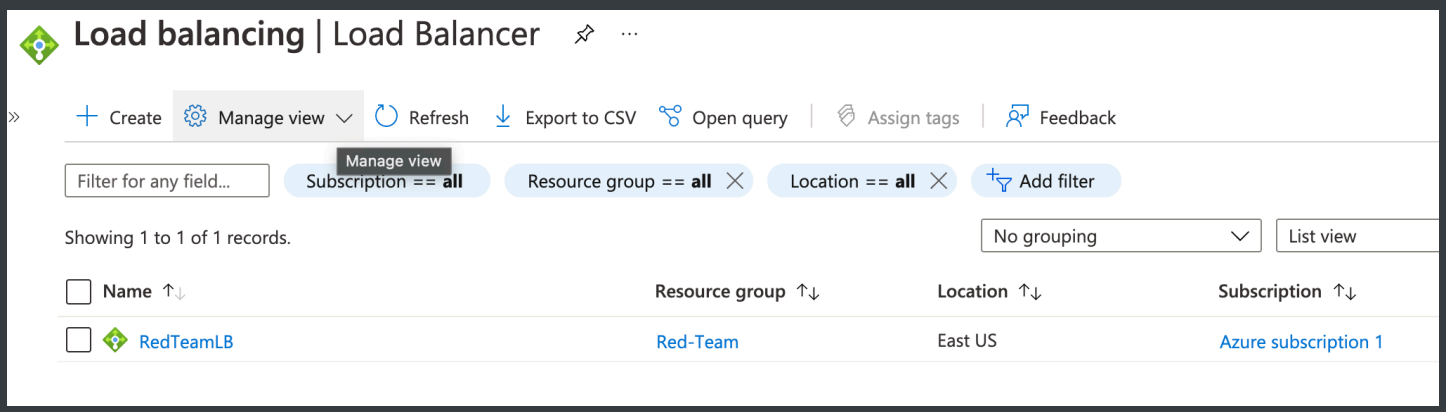
These files have been tested and used to generate a live ELK deployment on Azure. Thy can be used to recreate the entire deployment pictured.



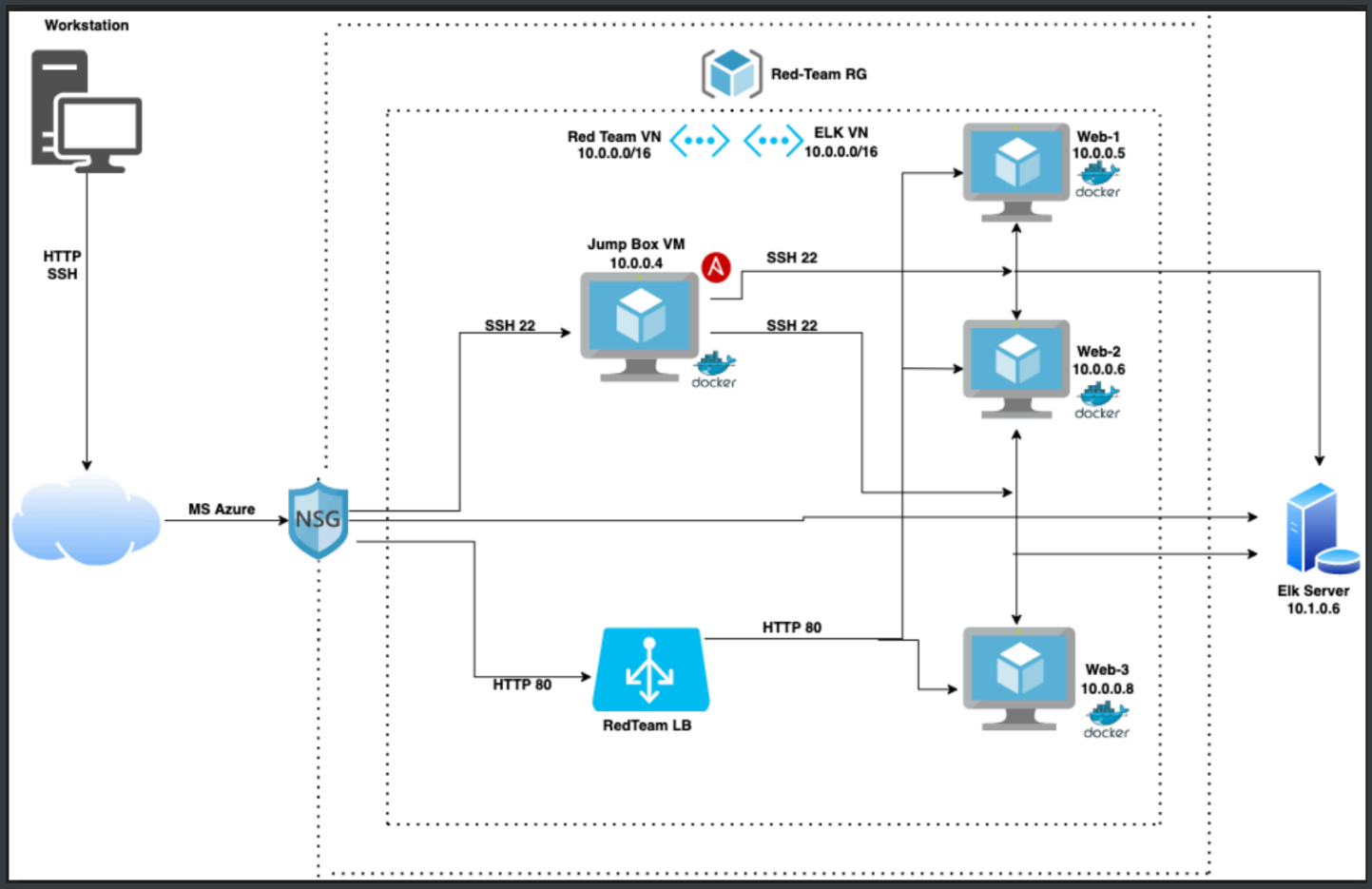



Network security groups
Default Directory
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Feedback
Leave preview
Filter for any field...
Subscription == all
Resource group == all
Location == all
Add filter
Showing 1 to 5 of 5 records.
No grouping
Name
Resource group
Location
Subscription
Flow log
ElkVM-nsg Red-Team West US Azure subscription 1
ElkVMnsg361 Red-Team West US Azure subscription 1
ElkVMnsg676 Red-Team West US Azure subscription 1
RedTeamSG Red-Team East US Azure subscription 1
Web-3-nsg Red-Team East US Azure subscription 1


Virtual machines
Default Directory
Create
Switch to classic
Reservations
Manage view
Refresh
Export to CSV
Open query
Filter for any field...
Subscription == all
Resource group == all
Location == all
Add filter
Showing 1 to 5 of 5 records.
No grouping
Name
Subscription
Resource group
Location
Status
Operating sys
ElkVM Azure subscription 1 Red-Team West US Running Linux
Jump-Box-Provisioner Azure subscription 1 Red-Team East US Running Linux
Web-1 Azure subscription 1 Red-Team East US Running Linux
Web-2 Azure subscription 1 Red-Team East US Running Linux
Web-3 Azure subscription 1 Red-Team East US Running Linux


Load balancing | Load Balancer
Create
Manage view
Refresh
Export to CSV
Open query
Assign tags
Feedback
Manage view
Filter for any field...
Subscription == all
Resource group == all
Location == all
Add filter
Showing 1 to 1 of 1 records.
No grouping
List view
Name
Resource group
Location
Subscription
RedTeamLB Red-Team East US Azure subscription 1

CLOUD SECURITY VIRTUALIZATION NETWORK DIAGRAM



DAY 1: ELK INSTALLATION

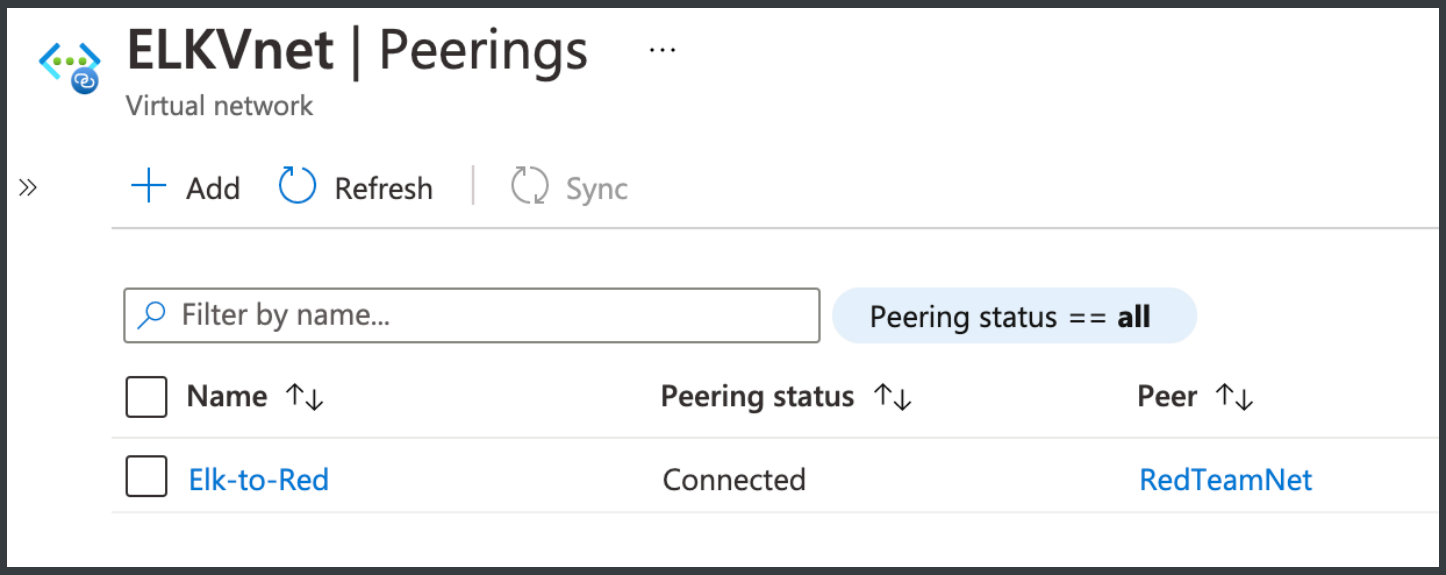
On day 1 I configured an ELK server within my virtual network. Specifically, I:

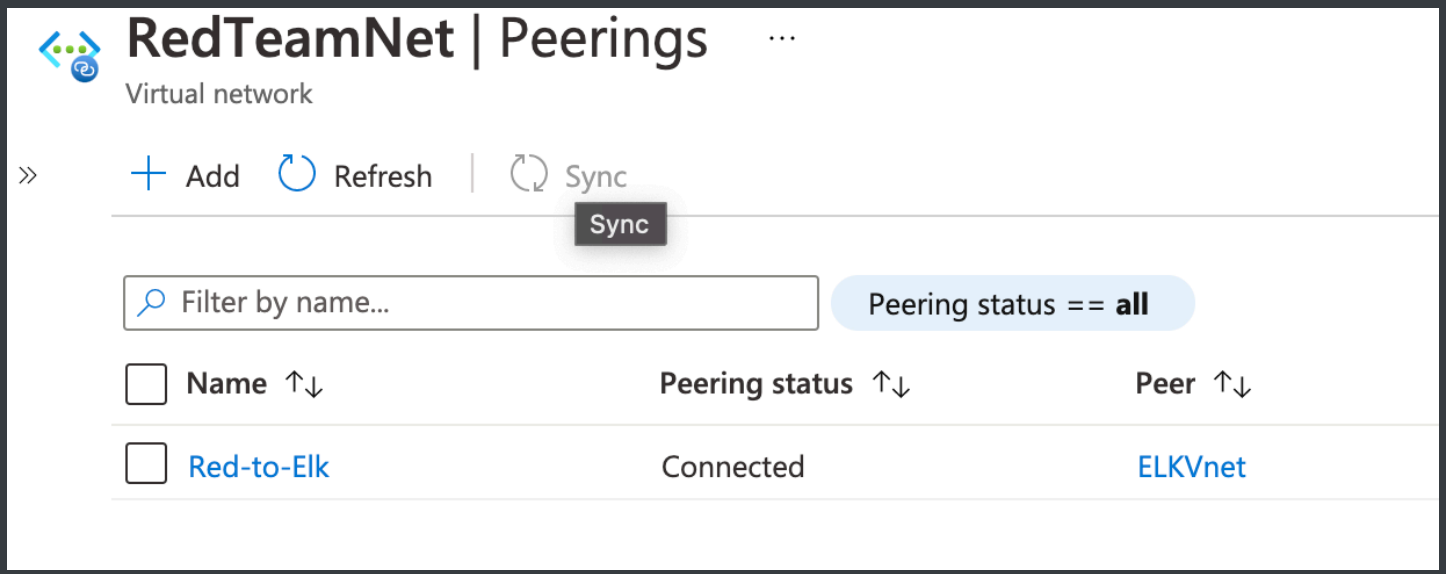
1. Created a new virtual network called ELKVnet in a new region (West US) within my Red-Team resource group.
2. Created a Peer Network Connection between my two vNets - ELKVnet and RedTeamNet.
3. Created a new VM called ElkVM and deployed this VM into the ELKVnet with its own

Security Group. This VM was host to the ELK server.

4. Downloaded and configured the elk-docker container onto the ElkVM.
5. Launched and exposed the elk-docker container to start the ELK server.
6. Implemented indentiy and access management by configuring the new security group so that I can connect ELK via HTTP and view it through the browser.

PEERING THE vNETS



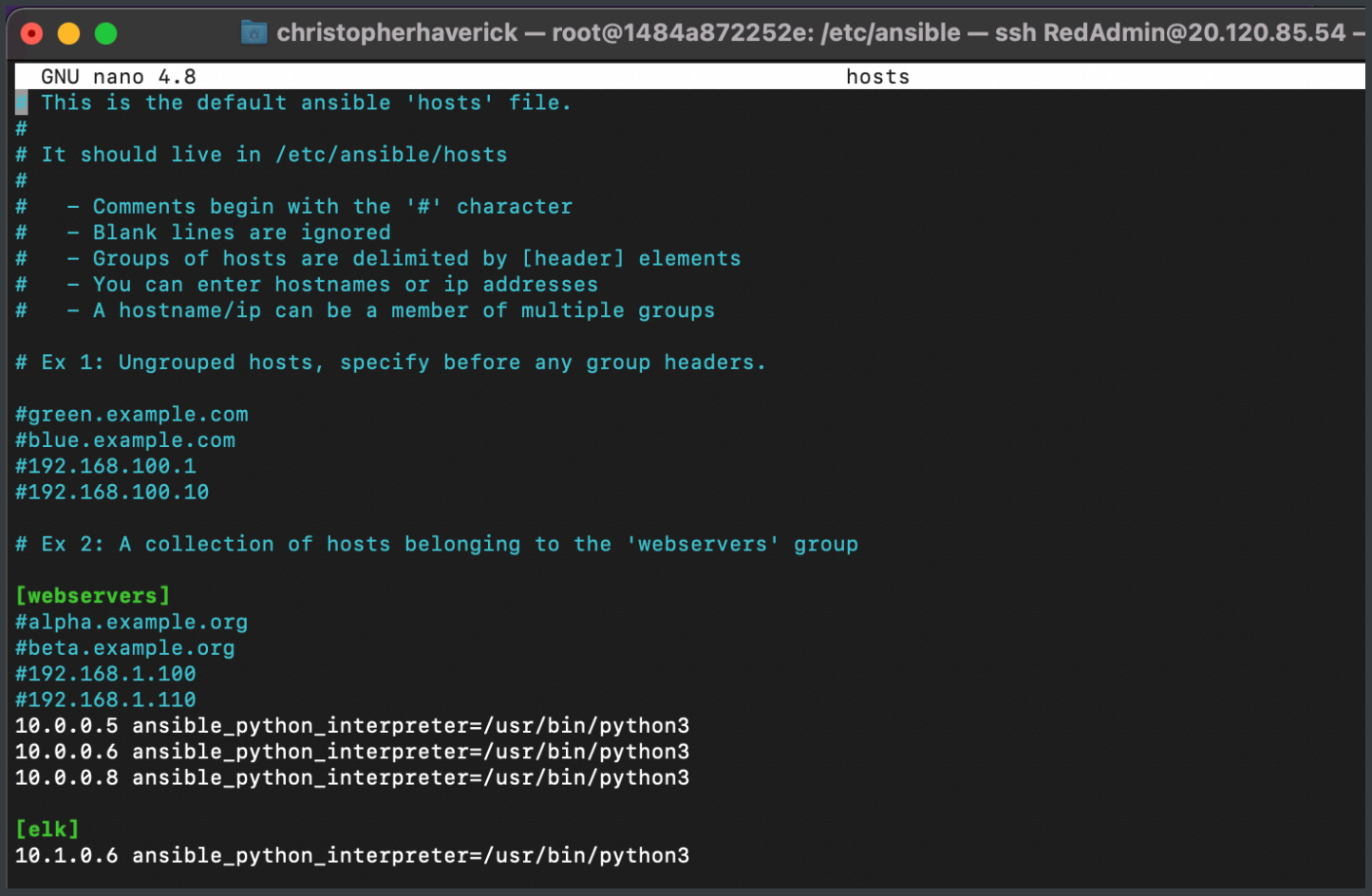


DOWNLOADING AND CONFIGURING THE CONTAINER

1. Using Ansible I configured the ElkVM.
2. From the Ansible container I added the ElkVM to Ansibles /etc/ansible/hosts file.
3. Created a playbook that installs docker and configures the container
4. Ran the playbook to launch the container
5. The ElkVM was created to run my ELK stack. In order to use Ansible to configure this VM I had to add it to the list of machines Ansible can discover and connect to.

Logging into /etc/ansible/hosts

```
RedAdmin@Jump-Box-Provisioner:~$ sudo docker container list -a
CONTAINER ID   IMAGE                     COMMAND                  CREATED       STATUS                     PORTS     NAMES
1484a872252e   cyberxsecurity/ansible    "/bin/sh -c /bin/bas…"   2 weeks ago   Exited (137) 7 days ago              elegant_haibt
RedAdmin@Jump-Box-Provisioner:~$ sudo docker start elegant_haibt
elegant_haibt
RedAdmin@Jump-Box-Provisioner:~$ sudo docker attach elegant_haibt
root@1484a872252e:~# ls
root@1484a872252e:~# cd etc
bash: cd: etc: No such file or directory
root@1484a872252e:~# cd /etc
root@1484a872252e:/etc# ls
adduser.conf            fstab         ldap           pam.conf     selinux
alternatives            gai.conf      legal          pam.d        shadow
ansible                 group         libaudit.conf  passwd       shells
apt                     group-        login.defs     profile      skel
bash.bashrc             gshadow       logrotate.d    profile.d    ssh
bindresvport.blacklist  gshadow-      lsb-release    python3      ssl
ca-certificates         gss           machine-id     python3.8    subgid
ca-certificates.conf    host.conf     mailcap        rc0.d        subuid
cron.d                  hostname      mailcap.order  rc1.d        sysctl.conf
cron.daily              hosts         metricbeat     rc2.d        sysctl.d
debconf.conf            init.d        mime.types     rc3.d        systemd
debian_version          inputrc       mke2fs.conf    rc4.d        terminfo
default                 issue         mtab           rc5.d        update-motd.d
deluser.conf            issue.net     nanorc         rc6.d        vim
dpkg                    kernel        networks       rcS.d        xattr.conf
e2scrub.conf            ld.so.cache   nsswitch.conf  resolv.conf
environment             ld.so.conf    opt            rmt
filebeat                ld.so.conf.d  os-release     security
root@1484a872252e:/etc# X
```



6. Once I created the [elk] group, I created a playbook to configure it called install-elk.yml

```
---
- name: Configure Elk VM with Docker
  hosts: elk
  remote_user: sysadmin
  become: true
  tasks:
    # Use apt module
    - name: Install docker.io
      apt:
        update_cache: yes
        force_apt_get: yes
        name: docker.io
        state: present
```

```
  # Use apt module
- name: Install python3-pip
  apt:
    force_apt_get: yes
    name: python3-pip
    state: present

  # Use pip module (It will default to pip3)
- name: Install python Docker module
  pip:
    name: docker
    state: present

  # Use command module
- name: Increase virtual memory
  command: sysctl -w vm.max_map_count=262144

  # Use sysctl module
- name: Use more memory
  sysctl:
    name: vm.max_map_count
    value: "262144"
    state: present
    reload: yes

  # Use docker_container module
- name: download and launch a docker elk container
  docker_container:
    name: elk
    image: sebp/elk:761
    state: started
    restart_policy: always
```

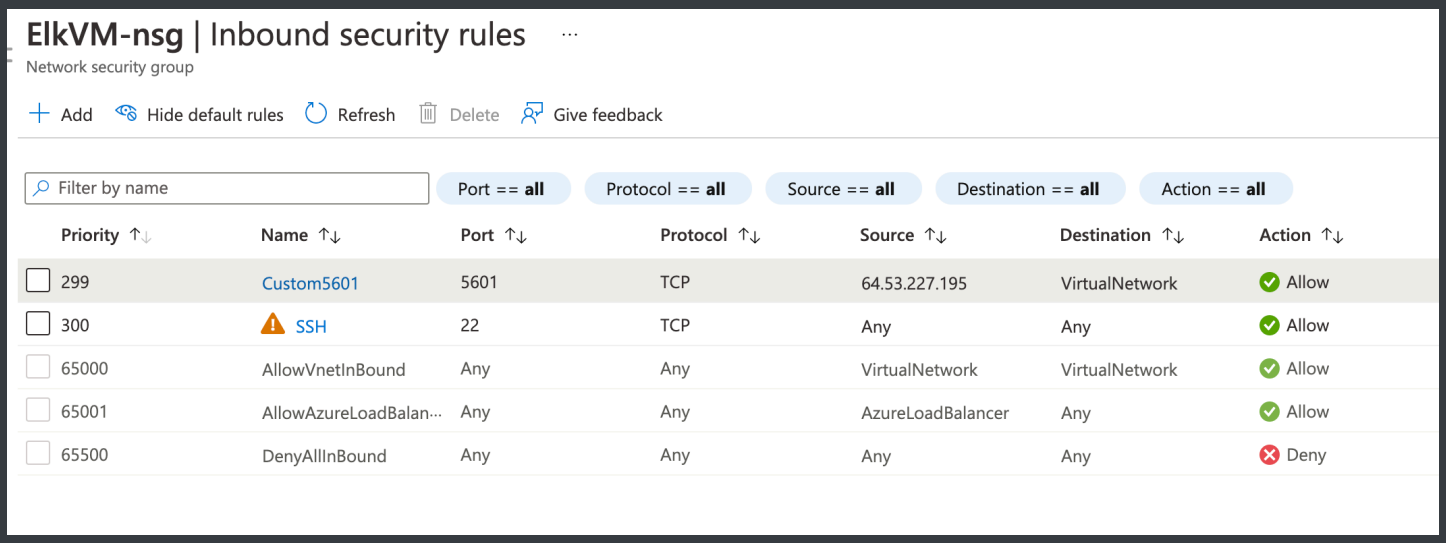
```
    # Please list the ports that ELK runs on
    published_ports:
      -  5601:5601
      -  9200:9200
      -  5044:5044

  # Use systemd module
- name: Enable service docker on boot
  systemd:
    name: docker
    enabled: yes
```

7. Next I ran the playbook using the command: ansible-playbook install-elk.yml. This will enable the docker service on boot, so that when I restart the ElkVM the docker service starts up automatically.

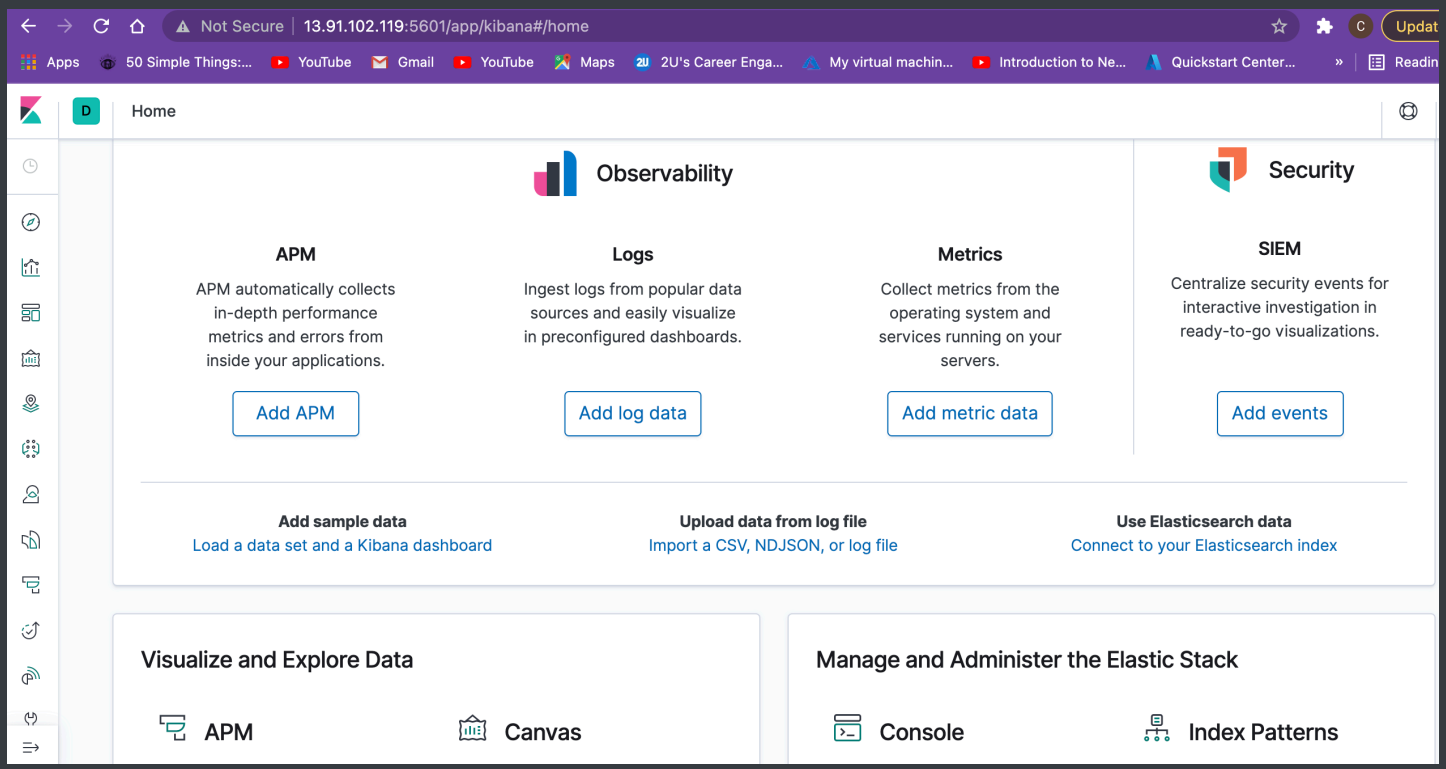
```
root@1484a872252e:/etc/ansible# ansible-playbook install-elk.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths
that can cause Display to print incorrect line lengths

PLAY [Configure Elk VM with Docker] ********************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [10.1.0.6]

TASK [Install docker.io] *******************************************************************************************************
ok: [10.1.0.6]

TASK [Install python3-pip] *****************************************************************************************************
ok: [10.1.0.6]

TASK [Install python Docker module] ********************************************************************************************
ok: [10.1.0.6]

TASK [Increase virtual memory] *************************************************************************************************
changed: [10.1.0.6]

TASK [Use more memory] *********************************************************************************************************
ok: [10.1.0.6]

TASK [download and launch a docker elk container] ******************************************************************************
[DEPRECATION WARNING]: The container_default_behavior option will change its default value from "compatibility" to
"no_defaults" in community.docker 2.0.0. To remove this warning, please specify an explicit value for it now. This feature will
 be removed from community.docker in version 2.0.0. Deprecation warnings can be disabled by setting deprecation_warnings=False
in ansible.cfg.
ok: [10.1.0.6]

TASK [Enable service docker on boot] *******************************************************************************************
ok: [10.1.0.6]

PLAY RECAP *********************************************************************************************************************
10.1.0.6                   : ok=8    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

8. Next I restricted access to the ElkVM using Azure's network security groups. The ELK

stack's web server runs on port 5601. I opened my virutal network's existing NSG and created an incoming rule for my security group that allows TCP traffic over port 5601 from my public IP address.



9. Finally, I verified that i could access my server by navigating to {ElkVM public IP}:5601/app/kibana



*DAY 2: FILEBEAT AND METRICBEAT INSTALLATION**

The ELK monitoring server is installed and configured I added Filebeat and Metricbeat. Filebeat is used to collect, parse, and visualize ELK logs which helps to better keep track of organizational goals. I completed the following:

1. Install Filebeat/Metricbeat on the Web VM's
2. Created a Filebeat/Metricbeat configuration file for my DVWA VMs
3. Created a Filebeat/Metricbeat installation play by creating an Ansible playbook that accomplishes the tasks required to install Filebeat/Metricbeat
4. Verified the installation and playbooks by confirming that both the installation and playbooks worked by verifying that the ELK stack is receiving logs.

5. In the Filebeat config file I replaced the IP address with the private IP address of my ELK machine in two places

```
#-------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  # Boolean flag to enable or disable the output module.
  #enabled: true

  # Array of hosts to connect to.
  # Scheme and port can be left out and will be set to the default (http and 9200)
  # In case you specify and additional path, the scheme is required: http://localhost:9200/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:9200
  hosts: ["10.1.0.6:9200"]
  username: "elastic"
  password: "changeme" # TODO: Change this to the password you set
```

```
#============================== Kibana =====================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:
  host: "10.1.0.6:5601" # TODO: Change this to the IP address of your ELK server
  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  #host: "localhost:5601"
```

6. Next I created a new playbook that installs Filebeat and copied the configuration file to the correct location.

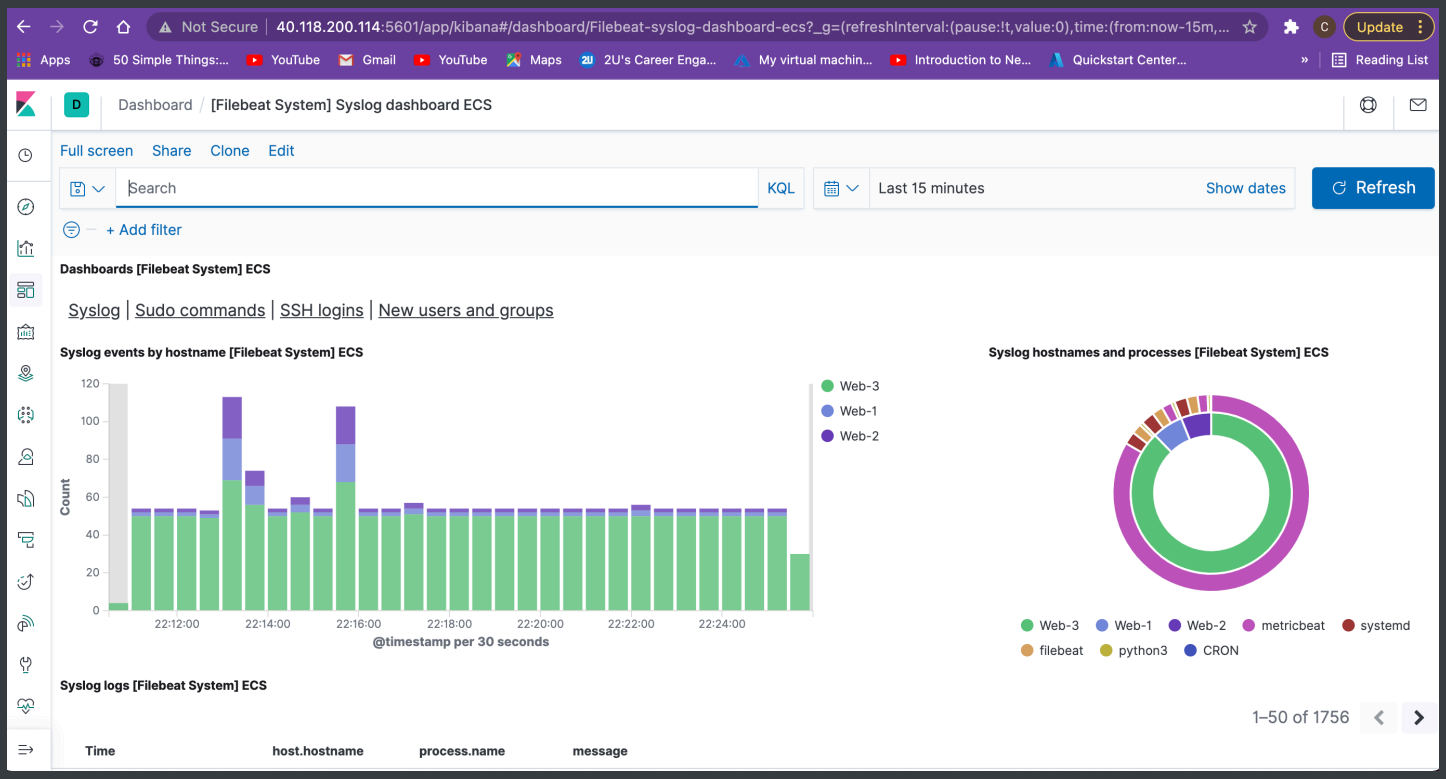
```
---
- name: installing and launching filebeat
  hosts: webservers
  become: yes
  tasks:

  - name: download filebeat deb
    command: curl -L -O
https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.6.2-
amd64.deb


  - name: install filebeat deb
    command: dpkg -i filebeat-7.6.2-amd64.deb

  - name: drop in filebeat.yml
    copy:
      src: /etc/ansible/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
```

7. After creating and saving this playbook I ran it with the command ansible-playbook filbert-playbook.yml

```
root@1484a872252e:/etc/ansible/roles# ansible-playbook filebeat-playbook.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths
that can cause Display to print incorrect line lengths

PLAY [installing and launching filebeat] ***************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [10.0.0.8]
ok: [10.0.0.5]
ok: [10.0.0.6]

TASK [download filebeat deb] ***************************************************************************************************
changed: [10.0.0.8]
changed: [10.0.0.6]
changed: [10.0.0.5]

TASK [install filebeat deb] ****************************************************************************************************
changed: [10.0.0.8]
changed: [10.0.0.5]
changed: [10.0.0.6]

TASK [drop in filebeat.yml] ****************************************************************************************************
ok: [10.0.0.8]
ok: [10.0.0.5]
ok: [10.0.0.6]

TASK [enable and configure system module] **************************************************************************************
changed: [10.0.0.5]
changed: [10.0.0.6]
changed: [10.0.0.8]

TASK [setup filebeat] **********************************************************************************************************
changed: [10.0.0.5]
changed: [10.0.0.8]
changed: [10.0.0.6]

TASK [start filebeat service] **************************************************************************************************
changed: [10.0.0.5]
changed: [10.0.0.8]
changed: [10.0.0.6]

TASK [enable service filebeat on boot] *****************************************************************************************
ok: [10.0.0.8]
ok: [10.0.0.5]
ok: [10.0.0.6]

PLAY RECAP *********************************************************************************************************************
10.0.0.5                   : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.0.0.6                   : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.0.0.8                   : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

8. Then I confirmed that the ELK stack was receiving logs from my DVWA machines on Kibana by navigating to 'Add log data' - 'System logs' - 'System logs dashboard'

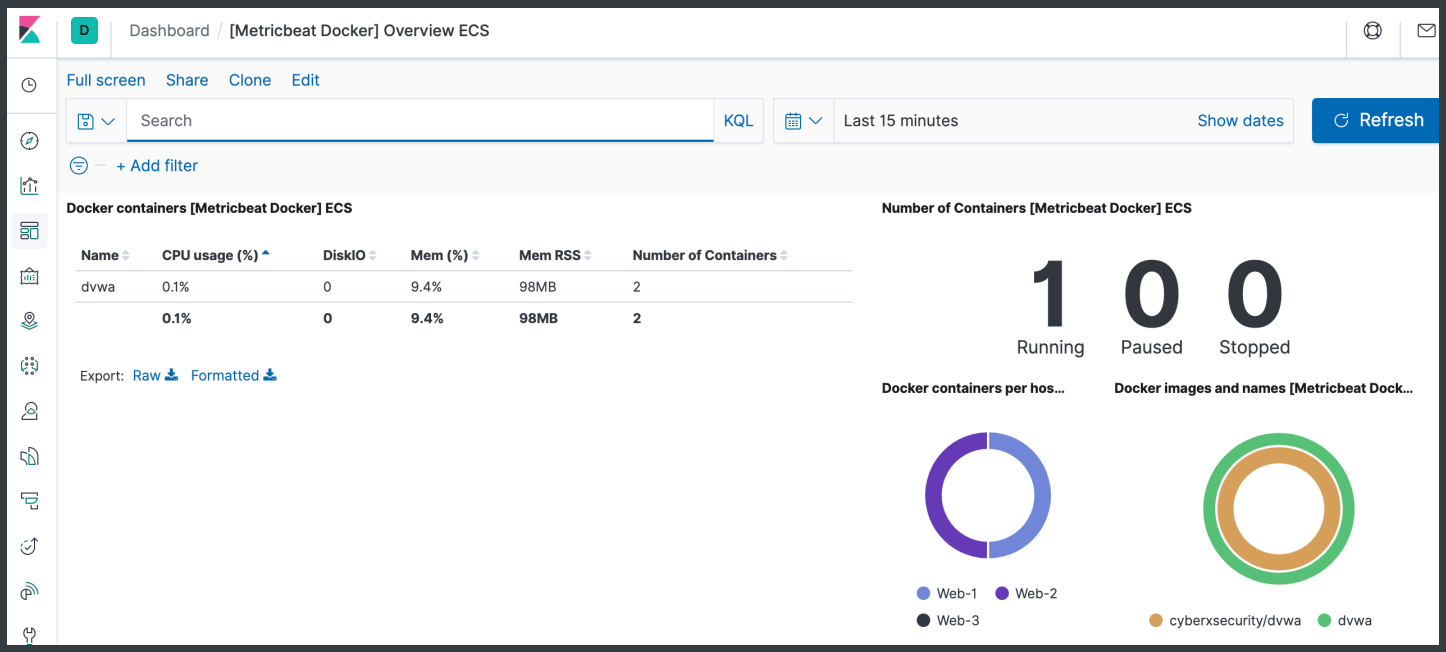


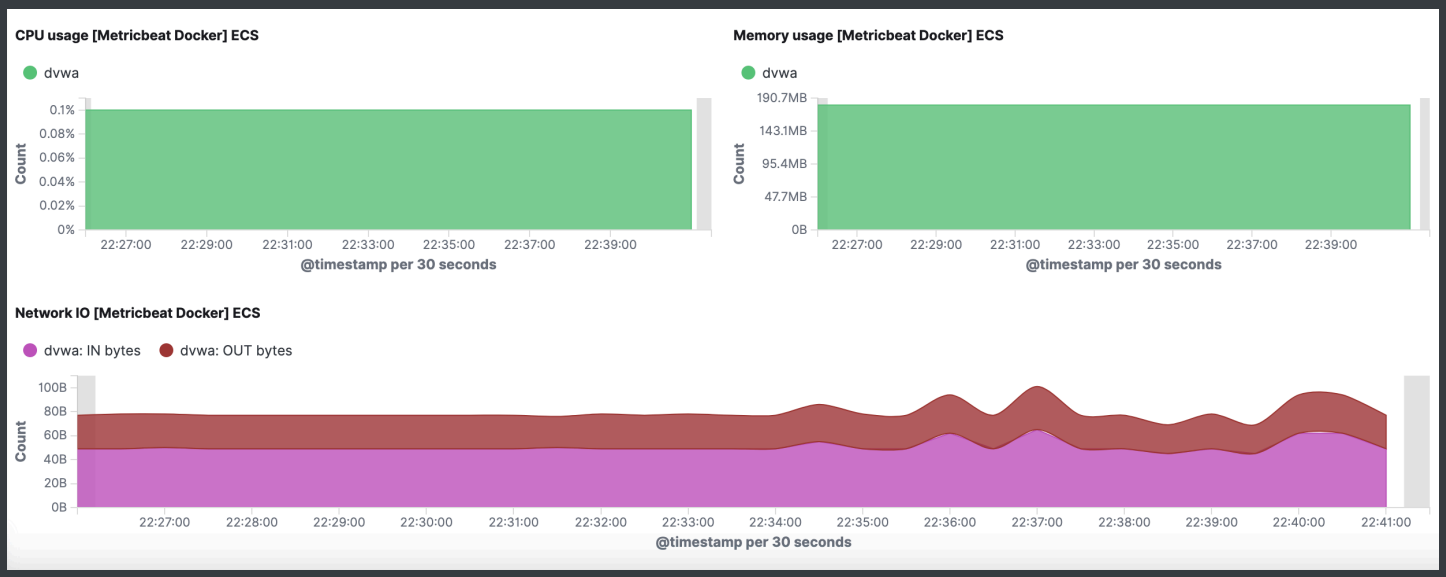
9. Next I installed Metricbeat in the same method of installing Filebeat. I went into the metricbeat.config.yml file and changed the IP adress in the output.elasticsearch and setup.kibana parts of the file to the private IP address of the ElkVM. Then I created a metricbeat-playbook.yml file and ran the playbook.

```
---
- name: Install metric beat
  hosts: webservers
  become: true
  tasks:
    # Use command module
  - name: Download metricbeat
    command: curl -L -O
https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-7.6.1-
amd64.deb

    # Use command module
  - name: install metricbeat
```

```
    command: dpkg -i metricbeat-7.6.1-amd64.deb

    # Use copy module
  - name: drop in metricbeat config
    copy:
      src: /etc/ansible/files/metricbeat-config.yml
      dest: /etc/metricbeat/metricbeat.yml

    # Use command module
  - name: enable and configure docker module for metric beat
    command: metricbeat modules enable docker

    # Use command module
  - name: setup metric beat
    command: metricbeat setup

    # Use command module
  - name: start metric beat
    command: service metricbeat start

    # Use systemd module
  - name: enable service metricbeat on boot
    systemd:
      name: metricbeat
      enabled: yes
```

10. Finally I made sure metric data was being received by the ELK server by navigating Kibana by going to 'Add Metric Data' - 'Docker Metrics' - 'Docker Metrics Dashboard'


D
Dashboard / [Metricbeat Docker] Overview ECS
Full screen
Share
Clone
Edit
Search
KQL
Last 15 minutes
Show dates
Refresh
+ Add filter
Docker containers [Metricbeat Docker] ECS
Name
CPU usage (%)
DiskIO
Mem (%)
Mem RSS
Number of Containers
dvwa 0.1% 0 9.4% 98MB 2
0.1% 0 9.4% 98MB 2
Export: Raw Formatted
Number of Containers [Metricbeat Docker] ECS
1
Running
0
Paused
0
Stopped
Docker containers per hos...
Web-1
Web-2
Web-3
Docker images and names [Metricbeat Dock...
cyberxsecurity/dvwa
dvwa


CPU usage [Metricbeat Docker] ECS
dvwa
0.1%
0.08%
0.06%
0.04%
0.02%
0%
Count
22:27:00
22:29:00
22:31:00
22:33:00
22:35:00
22:37:00
22:39:00
@timestamp per 30 seconds
Memory usage [Metricbeat Docker] ECS
dvwa
190.7MB
143.1MB
95.4MB
47.7MB
0B
Count
22:27:00
22:29:00
22:31:00
22:33:00
22:35:00
22:37:00
22:39:00
@timestamp per 30 seconds
Network IO [Metricbeat Docker] ECS
dvwa: IN bytes
dvwa: OUT bytes
100B
80B
60B
40B
20B
0B
Count
22:27:00
22:28:00
22:29:00
22:30:00
22:31:00
22:32:00
22:33:00
22:34:00
22:35:00
22:36:00
22:37:00
22:38:00
22:39:00
22:40:00
22:41:00
@timestamp per 30 seconds