Asure Cloud Infrastructure – Setup, Usage and Functionality

ALL ASURE VIRTUAL SERVERS ARE ON RESOURCE GROUP: RED_TEAM

1. Create two Virtual Networks (Red_Team_VN and Red_Team_VN2)

Set the following criteria and configurations to establish a Virtual Network:

-Search the Azure portal for 'virtual network', then click 'Add'

- Name: Red_Team_VN and Red_Team_VN2
- Subscription type: Azure Subscription 1
- Resource Group: Red_Team
- Region: West US 2 and East US 2
- Default network and subnet IP Addresses: 10.0.0.0/16 and 10.1.0.0/16
- Security Tab
  - -BastionHost: Disable
  - -DDoS Protection Standard: Disabled
  - -Firewall: Disable
- No tags are needed
- Click 'Review + Create'

2. Create two Network Security Groups (Red_Team_Security_Group and ELKServer-nsg)

-Search the Azure portal for 'network', choose Network security groups, then click 'Add'

- Subscription type: Azure Subscription 1
- Resource Group: Red_Team
- Name: Red_Team_Security_Group and ELKServer-nsg
- Region: West US 2 and East US 2
- Click 'Review + Create'

Restrict access to Network Security Groups

-On the Asure Website, Click on Network Security Group of Red_Team_Security_Group, click on 'Inbound Security Rules', then 'Add' 2 rules

1 - Restrict ALL Inbound Network Traffic

-Name: DenyAllInBound, Port: Any, Protocol: Any, Source: Any, Destination: Any, Action: Block /Deny

2 - Allow Inbound Network Traffic from Public IP Address

-Name: Public SSH SSH, Source: IP Address, Source IP Address: <Public IP Address(s) here>, Source Port Range: *, Destination: VirtualNetwork, Service: SSH, Destination Port Ranges: 22, Protocol: TCP, Action: Allow, Priority: 120, Description: SSH Inbound Traffic from Public IP Address

3. Jump-Box-Provisioner Server (1 CPU and 1 GB RAM)

Public IP: 20.83.225.130
Private IPs: 10.0.0.15

Description / Functionality

Our jump box is essentially a gateway router, which is exposed to the public internet utilizing it's SSH Port (#22). This server sits in front of other the other virtual machines (Load Balancer, Web Servers, ELK Server) which are not directly exposed to the public internet

It controls access to the other machines by allowing connections from specific IP addresses and forwarding network traffic to our other VMs

Virtual Network:
Red_Team_VM (Peered to Red_Team_VM2)

Network Security Group:
Red_Team_Security_Group

Location:
West US 2 (Zone 1)

Virtual Hardware Setup Process:
Create New JumpBox Virtual Machine

-Log into Asure website (https://portal.azure.com/) with Username and Password
-'Create New' VM in Asure with the following criteria:
  -Subscription: Azure subscription 1
  -Resource Group: Red_Team_VM
  -Virtual Machine Name: Jump-Box-Provisioner
  -Region: West US 2 (Zone 1)
  -Availability Options: Availability Zone
  -Availability Zone: 1
  -Image: Ubuntu Server 20.04 LTS - Gen 1
  -Size: Standard B1s (1 vcpus, 1 GiB memory)
  -Authentication Type: SSH Public Key
  -Username: azadmin
  -SSH public key source: Select 'Use existing public key'
  -Run GitBash command 'ssh-keygen' to generate SSH Public Key. Use for JumpBox and Web Servers

-Enter SSH Public Key:


```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCaCP2uNoG5LxdRHg74ZXiy4U5OFtUnts3JbqpPTA1ENKuyyuXNdSDHYg
vMyZXA5xCQY4G0EZrXtPzraxGt4R9mlTT3SOCO1AwRZX7IHa3gLqPtoU5cPVZDQlSeQg5N9ZVBDm2z9FiRHRMZ
724eoF7yoOeHQH6skBRv7jMcTdnetXvWYKvYD+6c9z1l8jeGYQJI8AHH7w4YK7xB5rBO+LQMB0TDe4Pu8Oaan0
5oJiDKcDEGCjYeconxfu8xI+DUvdAPgmlW+quwdYZyLN6jlIPbr2Rkstu83o4+lyZFWgXtzjkHsr7fj2GGxVZw
4vkh8dgqG3CQzDlmnccKy/IMlDWX root@3ca40a6a9d6b
```


  -Public inbound ports: Select 'Allow selected ports'
  -Select inbound ports: Select 'SSH (22)
  -Click 'Review and Create'

Create Docker / Ansible Container on JumpBox

-SSH into Jump Box
Command: ssh azadmin@20.83.225.130

-Update and Upgrade Jumpbox
Command: sudo apt-get update && apt-get upgrade

-Install docker.io container
-Command: sudo apt install docker.io

-Check Status of docker.io container
Command: sudo systemctl status docker

-Start to docker.io container
Command: sudo systemctl start docker

-Download specific docker container
Command: sudo docker pull cyberxsecurity/ansible

-Launch and Log into new docker container
Command: sudo docker run -ti cyberxsecurity/ansible:latest bash

-List all containers on JumpBox
Command: sudo docker container list -a
Command: sudo docker ps

-Start and attach to docker.io container
Command: sudo docker start <container name>
Command: sudo docker attach <container name>

-Navigate to ansible folder
Command: cd /etc/ansible/

Edit Ansible Config File

-Create an ansible configuration file (command: nano ansible.cfg) that installs an Ansible Docker container and configures it (file: ansible.cfg)

-Edit ansible.cfg file in the following section:
# default user to use for playbooks if user is not specified
# (/usr/bin/ansible will use current user as default)
remote_user = azadmin

-GitBash screen should look similar to:

```
root@3ca40a6a9d6b: ~
azadmin@Jump-Box-Provisioner:~$ sudo docker container list -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED       STATUS
             PORTS     NAMES
3ca40a6a9d6b   cyberxsecurity/ansible:latest   "bash"                   2 weeks ago   Exited (0) 3 ho
urs ago               cool_stonebraker
f1ca9b3e39f1   cyberxsecurity/ansible:latest   "bash"                   2 weeks ago   Exited (0) 2 we
eks ago               busy_keller
9f3fb4c6981c   cyberxsecurity/ansible:latest   "/bin/bash -o pipefa…"   2 weeks ago   Exited (0) 2 we
eks ago               relaxed_swirles
4450dc2e09e4   cyberxsecurity/ubuntu:bionic    "/bin/bash"              2 weeks ago   Exited (127) 2
weeks ago             trusting_swanson
azadmin@Jump-Box-Provisioner:~$ sudo docker start cool_stonebraker
cool_stonebraker
azadmin@Jump-Box-Provisioner:~$ sudo docker attach cool_stonebraker
root@3ca40a6a9d6b:~#
```

Restrict access to the JumpBox Server

-On the Asure Website, Click on Network Security Group of Jump-Box-Provisioner-nsg, click on 'Inbound Security Rules', then 'Add' 3 rules

1 - Restrict ALL Inbound Network Traffic

-Name: DenyAllInBound, Port: Any, Protocol: Any, Source: Any, Destination: Any, Action: Deny

2 - Allow Inbound Network Traffic from Public IP Address

-Name: JumpBox SSH, Source: IP Address, Source IP Address: <Public IP Address(s) here>, Source Port Range: *, Destination: VirtualNetwork, Service: SSH, Destination Port Ranges: 22, Protocol: TCP, Action: Allow, Priority: 120, Description: SSH Inbound Traffic from Public IP Address to Jump-Box Server

3 - Allow Outbound Network Traffic from Jumpbox

-Name: SSHFromJumpBox, Source: IP Address, Source IP Address: 10.0.0.15, Source Port Range: *, Destination: VirtualNetwork, Service: Custom, Destination Port Ranges: *, Protocol: Any, Action: Allow, Priority: 110, Description: From Public IP Address to Elk VM, via Port 5601

JumpBox Installation Files (Located on Jumpbox Ansible Container):

JumpBox Host file (cd /etc/ansible/ansible.cfg)

GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/ansible.cfg

How to Access JumpBox Server / Usage (from Laptop or Desktop GitBash Terminal):

-SSH into Jump Box

Command: ssh azadmin@20.83.225.130

-Run Update and Upgrade

Command: sudo apt-get update && apt-get upgrade

---

4. Web Servers #1, #2 (1 CPU and 2 GB RAM)

Public IP: 13.66.246.231 (Load Balancer)
Private IPs: 10.0.0.10 and 10.0.0.11

Description / Functionality

A web server distributes web pages as they are requisitioned. It's objective is to store, process and deliver web pages to users using Hypertext Transfer Protocol (HTTP).

When a user requests for a website by adding the URL or web address on a web browser's address bar (i.e. www.cnn.com), the browser sends a request to the Internet for viewing the web page for that address. A Domain Name Server (DNS) converts this URL to an IP Address (For example 192.168.216.345), which in turn points to a Web Server.

The Web Server is requested to present the content website to the user's browser. All websites on the Internet have a unique identifier in terms of an IP address. This Internet Protocol address is used to communicate between different servers across the Internet. Apache server is the most common web server available in the market
[Source: https://economictimes.indiatimes.com/definition/web-server]

Virtual Network:
Red_Team_VM (Peered to Red_Team_VM2)

Network Security Group:
Red_Team_Security_Group

Location:
West US 2 (Zone 1)

Virtual Hardware Setup Process:
Create New Web Servers (web-1 and web-2)

-Log into Asure website (https://portal.azure.com/) with Username and Password
-'Create New' VM in Asure with the following criteria:
  -Subscription: Azure subscription 1
  -Resource Group: Red_Team_VM
  -Virtual Machine Name: web-1 / web-2
  -Region: West US 2 (Zone 1)
  -Availability Options: Availability Zone
  -Availability Zone: 1
  -Image: Ubuntu Server 20.04 LTS - Gen 1
  -Size: Standard B1s (1 vcpus, 2 GiB memory)
  -Authentication Type: SSH Public Key
  -Username: azadmin
  -SSH public key source: Select 'Use existing public key'

-Enter SSH Public Key:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCaCP2uNoG5LxdRHg74ZXiy4U5OFtUnts3JbqpPTA1ENKuyyuXNdSDHYg
vMyZXA5xCQY4G0EZrXtPzraxGt4R9mlTT3SOCO1AwRZX7IHa3gLqPtoU5cPVZDQlSeQg5N9ZVBDm2z9FiRHRMZ
724eoF7yoOeHQH6skBRv7jMcTdnetXvWYKvYD+6c9z1l8jeGYQJI8AHH7w4YK7xB5rBO+LQMB0TDe4Pu8Oaan0
5oJiDKcDEGCjYeconxfu8xI+DUvdAPgmlW+quwdYZyLN6jlIPbr2Rkstu83o4+lyZFWgXtzjkHsr7fj2GGxVZw
4vkh8dgqG3CQzDlmnccKy/IMlDWX root@3ca40a6a9d6b

-Public inbound ports: Select 'Allow selected ports'
-Select inbound ports: Select 'SSH (22)
-Click 'Review and Create'

Config Web VMs with Docker (1st Step)

-Create new Ansible host file (command: nano hosts). Then, edit the hosts.yml file by adding the following:

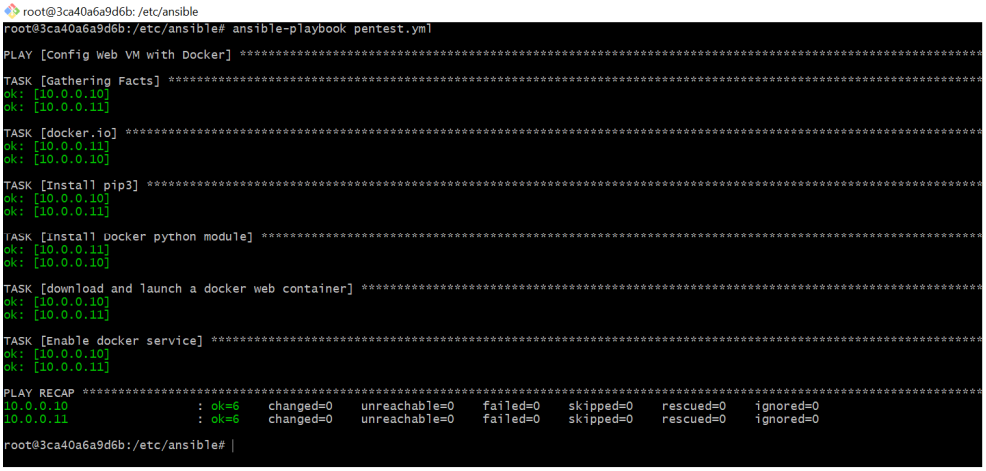
[webservers]
10.0.0.10 ansible_python_interpreter=/usr/bin/python3
10.0.0.11 ansible_python_interpreter=/usr/bin/python3

Config Web VMs with Docker (2nd Step)

-Create a playbook (command: nano pentest.yml) that installs an Ansible Docker container and configures it (file: pentest.yml)

-Run pentest.yml, with command: ansible-playbook pentest.yml

-GitBash screens should look similar to:



Restrict access to the Web Servers

-On the Asure Website, Click on Network Security Group of Web-1-nsg and Web-2-nsg, click on 'Inbound Security Rules', then 'Add' 3 rules

1 - Restrict ALL Inbound Network Traffic

-Name: DenyAllInBound, Port: Any, Protocol: Any, Source: Any, Destination: Any, Action: Block / Deny

2 - Allow Inbound Network Traffic from Public IP Address

-Name: Web1ServerInboundRule, Source: IP Address, Source IP Address: <Public IP Address(s) here>, Source Port Range: *, Destination: VirtualNetwork, Service: SSH, Destination Port Ranges: 22, Protocol: TCP, Action: Allow, Priority: 120, Description: SSH Inbound Traffic from Public IP Address to Web Server

3 - Allow Inbound SSH Network Traffic

-Name: SSH, Source: IP Address, Source IP Address: *, Source Port Range: *, Destination: VirtualNetwork, Service: SSH, Destination Port Ranges: 22, Protocol: TCP, Action: Allow, Priority: 140, Description: Allow Inbound SSH Network Traffic

Web Server Installation Files (Located on Jumpbox Ansible Container):
JumpBox Host file (cd /etc/ansible/ansible.cfg)
GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/ansible.cfg

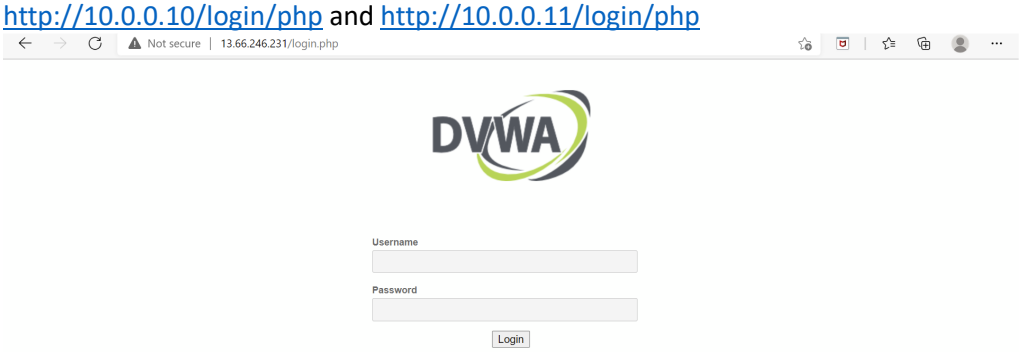
JumpBox Host file (cd /etc/ansible/hosts.yml)
GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/hosts.yml

JumpBoxr Install file (cd /etc/ansible/pentest.yml)
GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/pentest-yml.yml

How to Access JumpBox Server / Usage (from Laptop or Desktop GitBash Terminal):
-SSH into Jump Box
Command: ssh azadmin@20.83.225.130

-Run Update and Upgrade
Command: sudo apt-get update && apt-get upgrade

Verify that you can access your new ELK server by navigating to the Kibana website:
http://10.0.0.10/login/php and http://10.0.0.11/login/php



Load Balancer

Public IP: 13.66.246.231

Description / Functionality
A load balancer provides a website an external IP address that is accessed by the internet. This server receives network traffic that comes into the website and distributes it across multiple web servers

As the website receives more traffic, more servers can be added to the group ("pool") of servers that the load balancer has access to. This helps distribute traffic evenly among the servers and mitigates DoS attacks

A load balancer typically also has a **health probe** function to regularly check all of the web servers behind it, checking their status before sending traffic to them. Web VMs with issues are reported, and

the load balancer stops sending traffic to those servers. A load balanced configuration is much more resilient against a DDOS attack than if a single server was running the website.

Virtual Network:
Red_Team_VM (Peered to Red_Team_VM2)

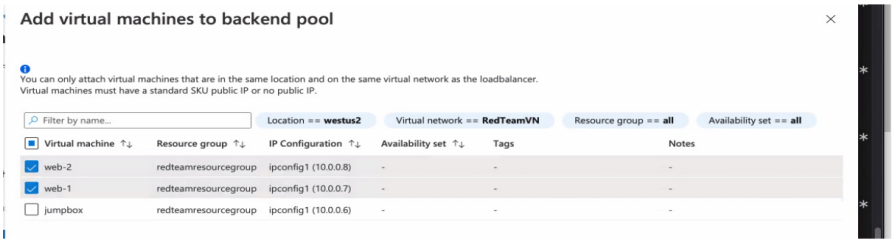
Network Security Group:
Red_Team_Security_Group

Location:
West US 2 (Zone 1)

Virtual Hardware Setup Process:
Create New Load Balancer Server
-Log into Asure website (https://portal.azure.com/) with Username and Password
-'Create New' Load Balancer in Asure with the following criteria:
  -Subscription: Azure subscription 1
  -Resource Group: Red_Team
  -Load Balancer Name: RedTeamLB
  -Region: West US 2 (Zone 1)
  -Type: Public
  -SKU: Standard
  -Tier: Regional
  -Public IP Address: Create New
  -Public IP Address Name: RedTeamLB
  -Public IP Address SKU: Standard
  -IP address assignment: Static
  -Availability Zone: 1
  -Add a public IPv6 address: No
  -Routing Preference: Microsoft network

Add a Health Probe
-Click Health Probes, then Add…
-Name: RedTeamHealthProbe
-Protocol: TCP
-Port: 80
-Interval: 5
-Unhealthy Threshold: 2
-Used by: RTLBRuleInbound
-Click on 'Save'

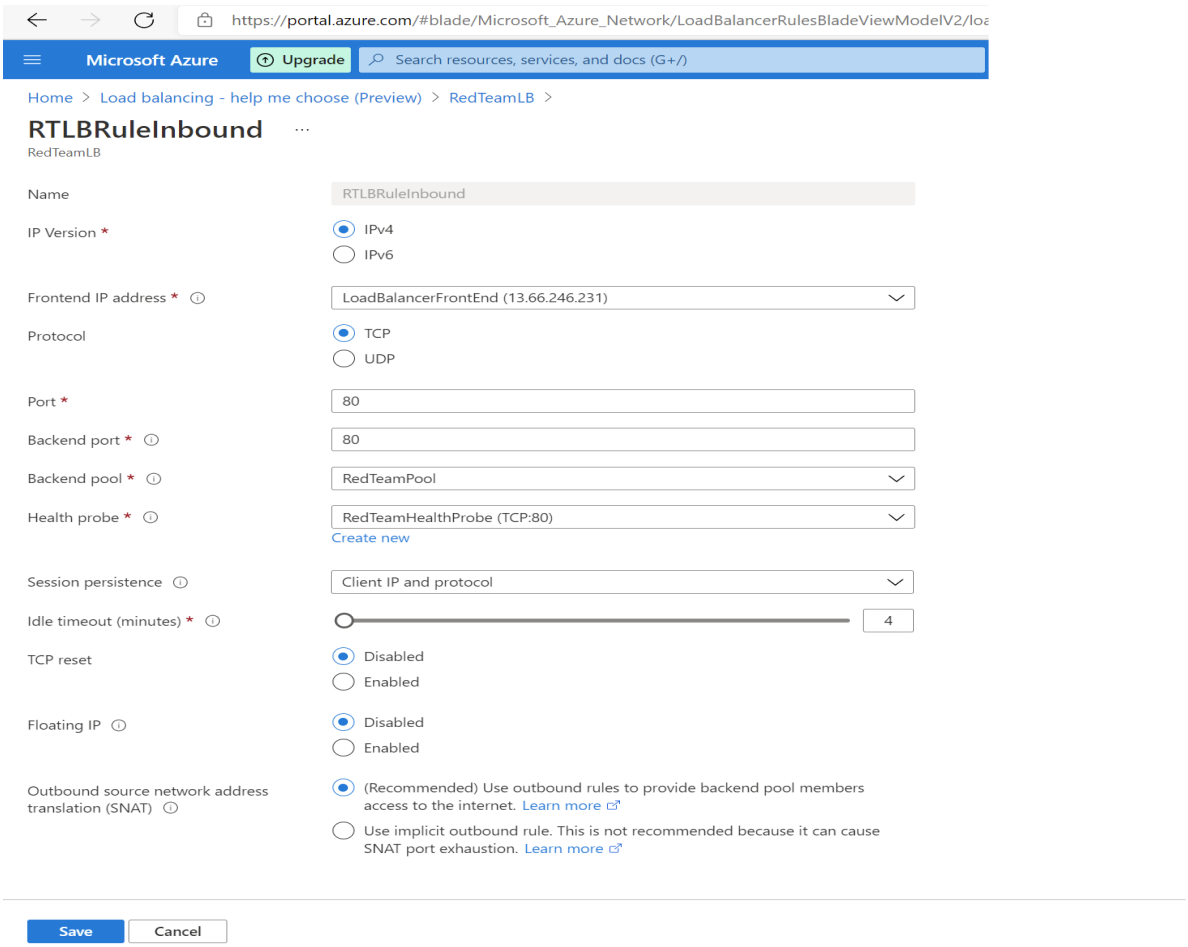
Add a Backend Pool
-Name: RedTeamPool
-Virtual Network: Red_Team_VN
-Backend Pool Configuration: NIC
-IP Version: IPv4
-Virtual Machines: Add, then select web-1 and web-2



Create Inbound and Outboud Network Rules

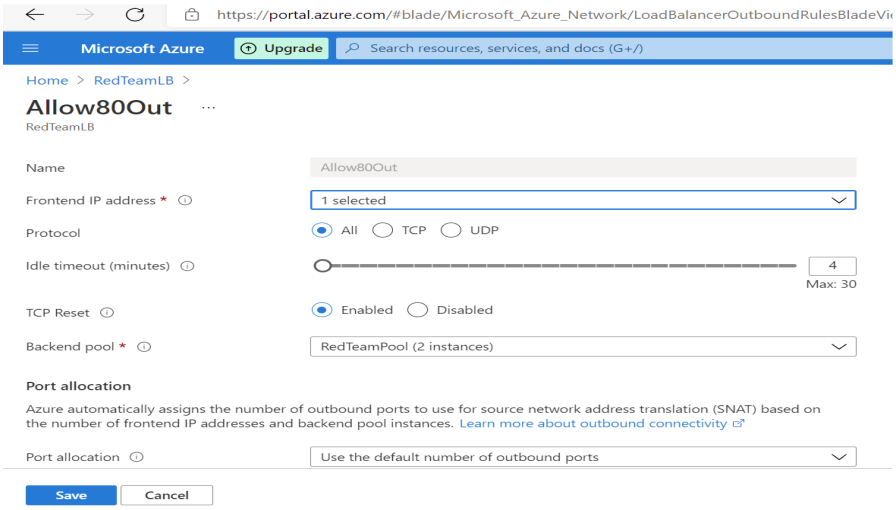
1 - Allow Inbound Network Traffic

-Name: RTLBRuleInbound, IP Version: IPv4, Frontend IP address: LoadBalancerFrontEnd (13.66.246.231), Protocol: TCP, Port: 80, Backend port: 80, Backend pool: RedTeamPool, Health probe: RedTeamHealthProbe (TCP:80), Session persistence: Client and protocol, Idle timeout (minutes): 4, TCP reset: Disabled, Floating IP: Disabled, Outbound source network address translation (SNAT): (Recommended) Use outbound rules to provide backend pool members access to the internet.



2 - Allow Outbound Network Traffic

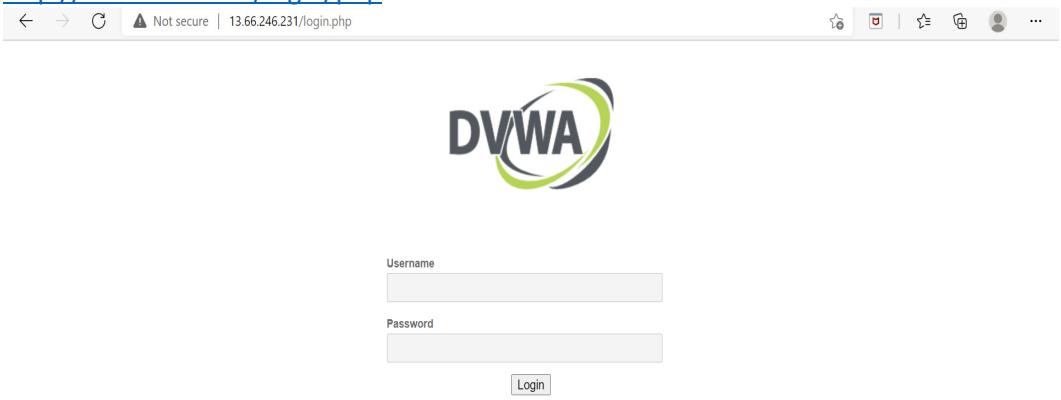
-Name: Allow80Out, Frontend IP Address: LoadBalancerFrontEnd (13.66.246.231), Protocol: All, Idle Timeout (minutes): 4, TCP Reset: Enabled, Backend pool: RedTeamPool (2 instances), Port allocation: Use the default number of outbound ports



Web Server Installation Files (Located on Jumpbox Ansible Container):
None

How to Access JumpBox Server / Usage (from Laptop or Desktop GitBash Terminal):
-SSH into Jump Box
Command: ssh azadmin@20.83.225.130

Verify that you can access your new ELK server by navigating to the Kibana website:
http://13.66.246.231/login/php



ELK Server (2 CPUs and 8 GB RAM)

Public IP: 20.62.88.5
Private IP: 10.1.0.4

Description / Functionality

ELK is an open-source technology comprised of 3 components

- Elasticsearch: Search and analytics engine, a special database for storing log data
- Logstash: Server-side data processing pipeline that sends data to Elasticsearch, a tool that makes it easy to collect logs
- Kibana: Tool for visualizing Elasticsearch data with charts and graphs
- 2 Beat Tools: Used to Monitor Web Server Traffic (Filebeats and Metricbeats)

Virtual Network:

Red_Team_VM2 (Peered to Red_Team_VM)

Network Security Group:

ELKServer-nsg

Location:

East US 2 (Zone 1)

Virtual Hardware Setup Process:

Create New ELK Server Virtual Machine

-Log into Asure website (https://portal.azure.com/) with Username and Password

-'Create New' VM in Asure with the following criteria:

-Subscription: Azure subscription 1
-Resource Group: NetworkWatcherRG
-Virtual Machine Name: ELKServer
-Region: East US 2 (Zone 1)
-Availability Options: Availability Zone
-Availability Zone: 1
-Image: Ubuntu Server 20.04 LTS - Gen 1
-Size: Standard D2s v3 (2 vcpus, 8 GiB memory)
-Authentication Type: SSH Public Key
-Username: azadmin
-SSH public key source: Select 'Use existing public key'
- Enter SSH Public Key:


```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCaCP2uNoG5LxdRHg74ZXiy4U5OFtUnts3JbqpPTA1ENKuyyuXNdSDHYg
vMyZXA5xCQY4G0EZrXtPzraxGt4R9mlTT3SOCO1AwRZX7IHa3gLqPtoU5cPVZDQlSeQg5N9ZVBDm2z9FiRHRMZ
724eoF7yoOeHQH6skBRv7jMcTdnetXvWYKvYD+6c9z1l8jeGYQJI8AHH7w4YK7xB5rBO+LQMB0TDe4Pu8Oaan0
5oJiDKcDEGCjYeconxfu8xI+DUvdAPgmlW+quwdYZyLN6jlIPbr2Rkstu83o4+lyZFWgXtzjkHsr7fj2GGxVZw
4vkh8dgqG3CQzDlmnccKy/IMlDWX root@3ca40a6a9d6b
```


-Public inbound ports: Select 'Allow selected ports'
-Select inbound ports: Select 'SSH (22)
-Click 'Review and Create'

Create Docker Container on ELK Server

-Edit Ansible host file (command: nano hosts). Make sure to edit the hosts.yml file by adding the following:

[elk]
10.1.0.4 ansible_python_interpreter=/usr/bin/python3

-Create a playbook (command: nano install-elk.yml) that installs a Docker container and configures it (file: install-elk.yml)

-Run install-elk.yml, with command: ansible-playbook install-elk.yml

-GitBash Screen should look similar to:

```
root@3ca40a6a9d6b:/etc/ansible# ls
ansible.cfg           files            metricbeat-config.yml    pentest.yml
filebeat-config.yml   hosts            metricbeat-playbook.yml  roles
filebeat-playbook.yml install-elk.yml  my-playbook.yml
root@3ca40a6a9d6b:/etc/ansible# ansible-playbook install-elk.yml

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

TASK [Gathering Facts] *********************************************************************************
ok: [10.1.0.4]

TASK [Install docker.io] *******************************************************************************
ok: [10.1.0.4]

TASK [Install python3-pip] *****************************************************************************
ok: [10.1.0.4]

TASK [Install Docker module] ***************************************************************************
ok: [10.1.0.4]

TASK [Increase virtual memory] *************************************************************************
changed: [10.1.0.4]

TASK [Use more memory] *********************************************************************************
ok: [10.1.0.4]

TASK [download and launch a docker elk container] ******************************************************
ok: [10.1.0.4]

TASK [Enable service docker on boot] *******************************************************************
ok: [10.1.0.4]

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

Restrict access to the ELK VM

-On the Asure Website, Click on Network Security Group of ELKServer-nsg, click on 'Inbound Security Rules', then 'Add' 3 rules

1 - Restrict ALL Inbound Network Traffic

-Name: DenyAllInBound, Port: Any, Protocol: Any, Source: Any, Destination: Any, Action: Block / Deny

2 - Allow Inbound Network Traffic from Public IP Address

-Name: ElkInBound, Source: IP Address, Source IP Address: <Public IP Address(s) here>, Source Port Range: *, Destination: VirtualNetwork, Service: Custom, Destination Port Ranges: 5601, Protocol: Any, Action: Allow, Description: From Jumpbox Private IP Address to Elk VM, via Port 22 (SSH)

3 - Allow Inbound Network Traffic from Jumpbox

-Name: ElkInBound, Source: IP Address, Source IP Address: 10.0.0.4, Source Port Range: *, Destination: VirtualNetwork, Service: Custom, Destination Port Ranges: 5601, Protocol: TCP, Action: Allow, Description: From Public IP Address to Elk VM, via Port 5601

Elk Server Installation Files (Located on Jumpbox Ansible Container):

Elk Server Host file (cd /etc/ansible/hosts.yml)

GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/hosts.yml

Elk Server Install file (cd /etc/ansible/install-elk.yml)

GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/install-elk.yml

How to Access ELK Server / Usage (from Laptop or Desktop GitBash Terminal):

-SSH into Jump Box

Command: ssh azadmin@20.83.225.130

-Run Update and Upgrade (weekly)

Command: sudo apt-get update && apt-get upgrade

-List all the containers created on the system:

Command: sudo docker container list -a

-Start and Attach the JumpBox Container

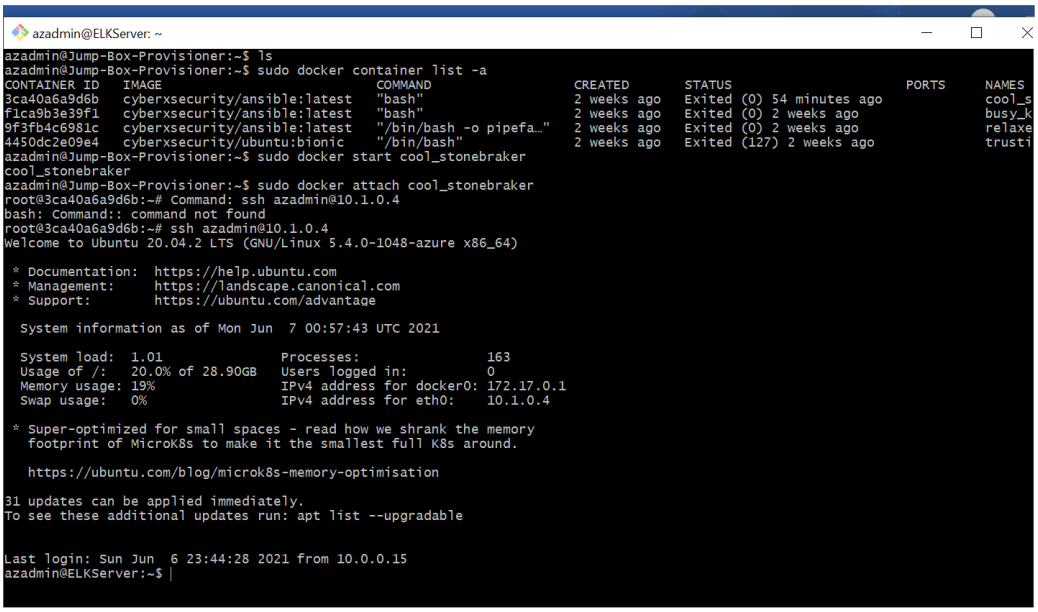
Command: sudo docker start <Container Name>

Command: sudo docker attach <Container Name>

-SSH Connect from Jumpbox Container to New ELK Server

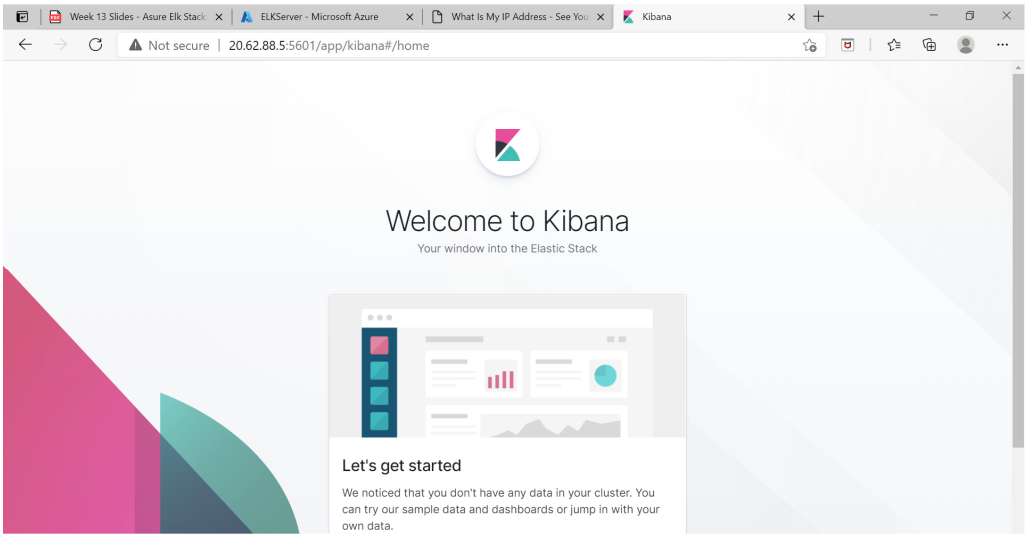
Command: ssh azadmin@10.1.0.4

-GitBash Screen should look similar to:



Verify that you can access your new ELK server by navigating to the Kibana website:

http://20.62.88.5:5601/app/kibana



Elk Server Data Collection Tool - Filebeats

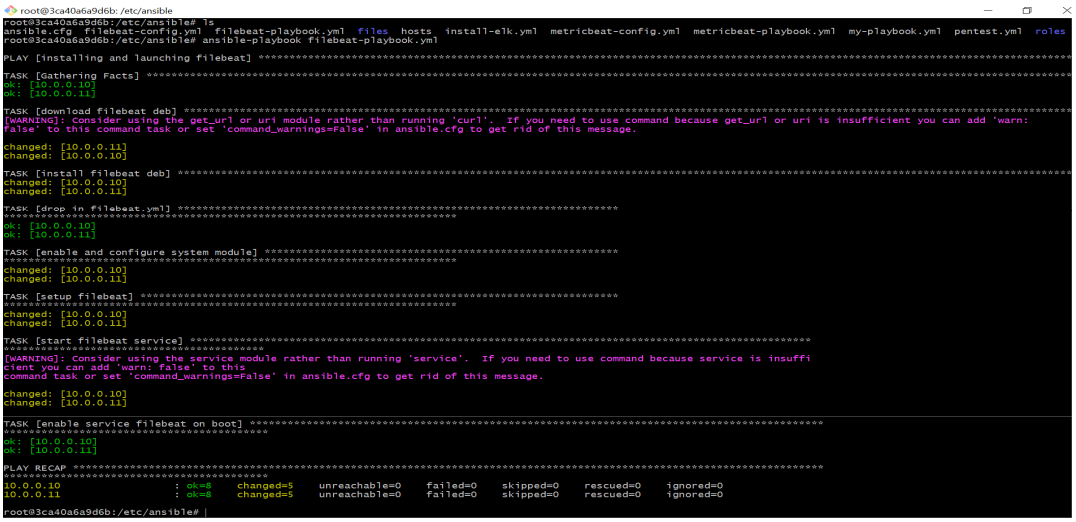
Description / Functionality:

Collects data about the file system, enables analysts to monitor files for suspicious changes.
Use Filebeat to collect, parse, and visualize ELK logs in a single command. This will help us better track our organizational goals. Specifically, we will monitor the Apache server and MySQL database logs generated by the web servers

Install and Run New Filebeats Tool to Monitor Web Server Traffic Logs:

-Create new Filebeats config and playbook files (files: filebeat-config.yml and filebeat-playbook.yml)
command: nano filebeat-config,yml
command: nano filebeat-playbook,yml
command: ansible-playbook filebeat-playbook,yml

GitBash Screen should look similar to:



Filebeat Installation Files (Located on Jumpbox Ansible Container):

Filebeat Install files (cd /etc/ansible/filebeat-config.yml and filebeat-playbook.yml)
GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/filebeat-config.yml
GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/filebeat-playbook.yml

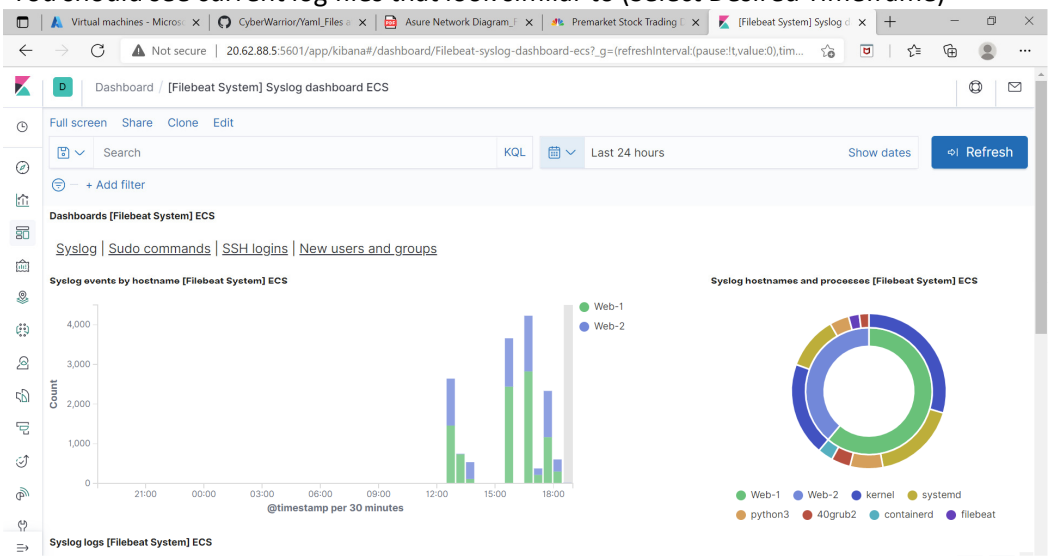
Usage / How to Access:

Confirm that the ELK Stack is receiving logs

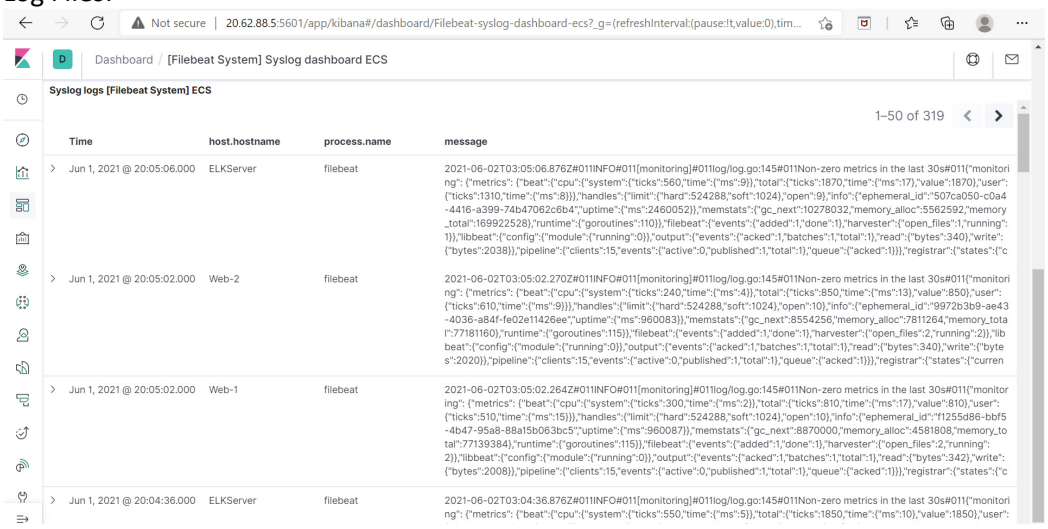
-Log into Kibana website by using the browser link: http://20.62.88.5:5601/app/kibana

-Click on 'Add Log Data' then 'System Logs' then Getting Started 'DEB' Tab, then 'Systems Log Dashboard' to view Web Server Log Files

-You should see current log files that look similar to (Select Desired Timeframe)



Log Files:



Elk Server Data Collection Tool - Metricbeats

Description / Functionality

Collects machine metrics, such as uptime and CPU usage, displays data in graph / charts

-CPU usage: The heavier the load on a machine's CPU, the more likely it is to fail. Analysts often receive alerts when usage is too high

- Uptime: Measures how long a machine has been on. Servers are generally expected to be available for a certain percentage of the time. Ensure the web servers meet service-level agreements (SLAs).

Install and Run New Metricbeats Tool to Collect and Display Web Server Metrics

-Create new Metricbeat config and playbook files (files: metricbeat-config.yml and metricbeat-playbook.yml)

command: nano metricbeat-config,ym

command: nano metricbeat-playbook,yml

command: ansible-playbook metricbeat-playbook,yml

GitBash Screen should look similar to:

```
root@3ca40a6a9d6b: /etc/ansible
ansible.cfg  filebeat-config.yml  filebeat-playbook.yml  files  hosts  install-elk.yml  metricbeat-config.yml  metricbeat-playbook.yml  my-playbook.yml  pentest.yml  roles
root@3ca40a6a9d6b:/etc/ansible#
root@3ca40a6a9d6b:/etc/ansible# ansible-playbook metricbeat-playbook.yml

PLAY [Install metric beat] ******************************************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************************************
ok: [10.0.0.11]
ok: [10.0.0.10]

TASK [Download metricbeat] ******************************************************************************************************************************************
[WARNING]: Consider using the get_url or uri module rather than running 'curl'.  If you need to use command because get_url or uri is insufficient you can add 'warn:
false' to this command task or set 'command_warnings=False' in ansible.cfg to get rid of this message.

changed: [10.0.0.11]
changed: [10.0.0.10]

TASK [install metricbeat] *******************************************************************************************************************************************
changed: [10.0.0.10]
changed: [10.0.0.11]

TASK [drop in metricbeat config] ************************************************************************************************************************************
ok: [10.0.0.10]
ok: [10.0.0.11]

TASK [enable and configure docker module for metric beat] ***********************************************************************************************************
changed: [10.0.0.10]
changed: [10.0.0.11]

TASK [setup metric beat] ********************************************************************************************************************************************
changed: [10.0.0.10]
changed: [10.0.0.11]

TASK [start metric beat] ********************************************************************************************************************************************
[WARNING]: Consider using the service module rather than running 'service'.  If you need to use command because service is insufficient you can add 'warn: false' to this
command task or set 'command_warnings=False' in ansible.cfg to get rid of this message.

changed: [10.0.0.10]
changed: [10.0.0.11]

TASK [enable service metricbeat on boot] ****************************************************************************************************************************
ok: [10.0.0.11]
ok: [10.0.0.10]

PLAY RECAP **********************************************************************************************************************************************************
10.0.0.10                  : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.0.0.11                  : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@3ca40a6a9d6b:/etc/ansible#
```

Metricbeat Installation Files (Located on Jumpbox Ansible Container):

Metricbeat Install files (cd /etc/ansible/metricbeat-config.yml and metricbeat-playbook.yml)

GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/metricbeat-config.yml

GitHub url: https://github.com/Jbrowne81/CyberWarrior/blob/main/Yaml_Files/metricbeat-playbook.yml

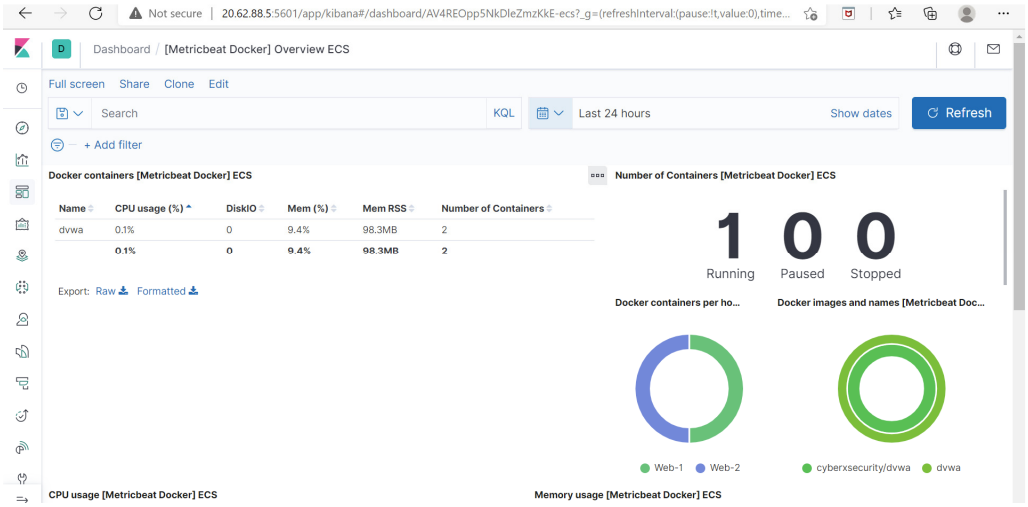
Usage / How to Access:

Confirm that the ELK Stack is receiving Metric Data

-Log into Kibana website by using the browser link: http://20.62.88.5:5601/app/kibana

-Click on 'Add Metric Data' then 'Docker Metrics' then Getting Started 'DEB' Tab, then 'Docker Metrics Dashboard' to view Web Server Metrics

-You should see current log files that look similar to (Select Desired Timeframe)


Not secure | 20.62.88.5:5601/app/kibana#/dashboard/AV4REOpp5NkDleZmzKkE-ecs?_g=(refreshInterval:(pause:!t,value:0),time...
Dashboard / [Metricbeat Docker] Overview ECS
Full screen Share Clone Edit
Search
KQL
Last 24 hours
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% 98.3MB 2
0.1% 0 9.4% 98.3MB 2
Export: Raw Formatted
Number of Containers [Metricbeat Docker] ECS
1 Running
0 Paused
0 Stopped
Docker containers per ho...
Docker images and names [Metricbeat Doc...
Web-1 Web-2
cyberxsecurity/dvwa dvwa
CPU usage [Metricbeat Docker] ECS
Memory usage [Metricbeat Docker] ECS


Not secure | 20.62.88.5:5601/app/kibana#/dashboard/AV4REOpp5NkDleZmzKkE-ecs?_g=(refreshInterval:(pause:!t,value:0),time...
Dashboard / [Metricbeat Docker] Overview ECS
Web-1 Web-2
cyberxsecurity/dvwa dvwa
CPU usage [Metricbeat Docker] ECS
dvwa
Count
0.1%
0.08%
0.06%
0.04%
0.02%
0%
21:00 00:00 03:00 06:00 09:00 12:00 15:00 18:00
@timestamp per 30 minutes
Memory usage [Metricbeat Docker] ECS
dvwa
190.7MB
143.1MB
95.4MB
0B
@timestamp per 30 minutes
Network IO [Metricbeat Docker] ECS
dvwa: IN bytes
dvwa: OUT bytes
100B
80B
60B
40B
20B
0B
@timestamp per 30 minutes