linkedin.com/in/heri-yono

OBTAIN PRACTICAL SKILLS
IN 30-DAYS
youtube/@mydfir

MYDFIR SOC ANALYST CHALLENGE

PART 1 (DAY 1 – DAY 10)

Workbook By Heri Yono

01	Create Logical Diagram	06	Elastic and Fleet Server Introduction
02	ELK Stack Introduction	07	Elastic and Fleet Server Setup
03	Elastisearch Setup	08	Sysmon Introduction
04	Kibana Setup	09	Sysmon Setup
05	Windows Server Installation	10	Elasticsearch Ingest Data

logstash → elasticsearch ← kibana

osTicket

MYTHIC C2

- Setup SIEM
- Spin Up C2
- Create Dashboards
- Create Alerts
- Setup Tikceting System

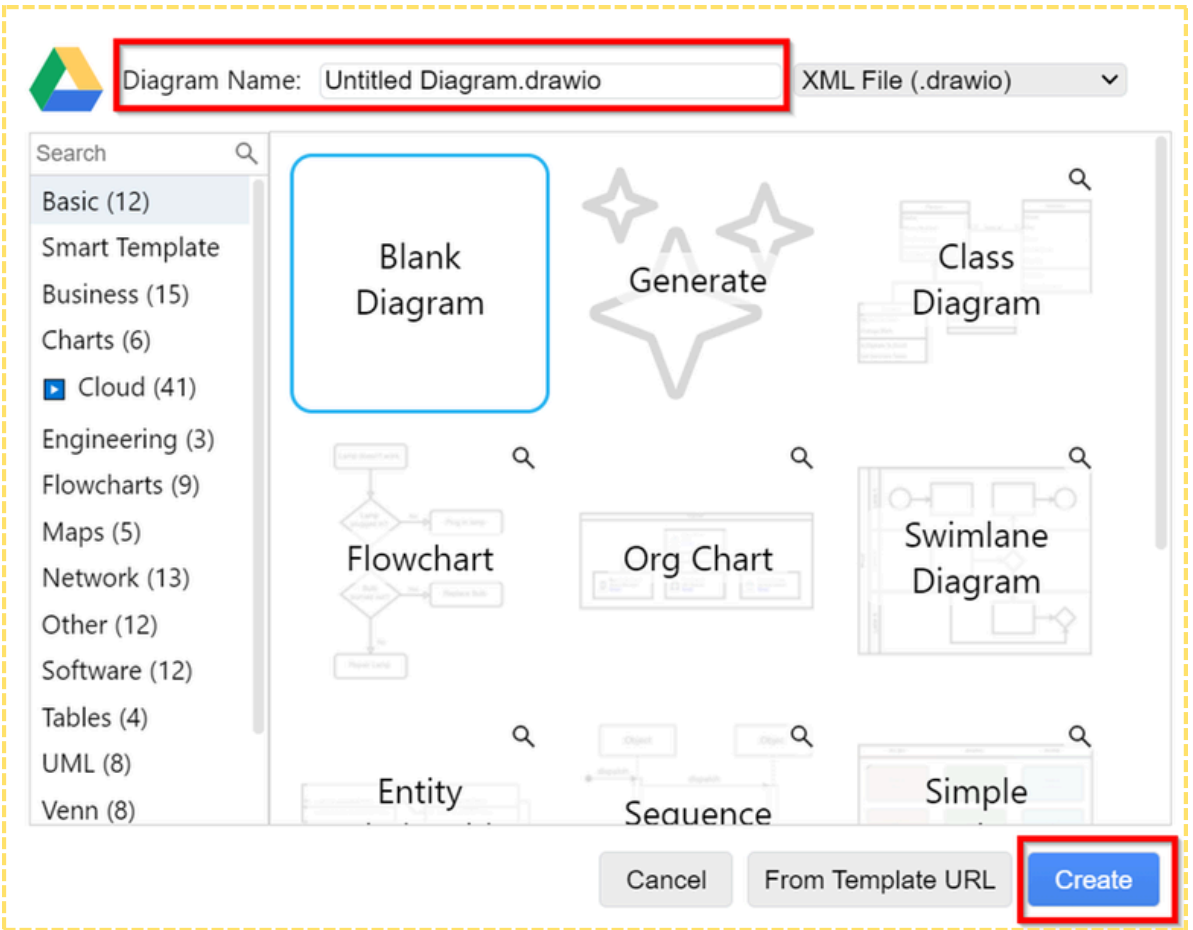
Day 1 - Create Logical Diagram using Draw io

This challnge provide the process of creating diagram using draw.io specifically for a Security Operations Center (SOC) infrastructure. This diagram will serve as a visual representation of the infrastructure we'll be working with during the 30-day SOC Analyst challenge. The aim is to give hopeful SOC analysts with a clear understanding of the environment they will be working with

Draw.io is a free diagramming tool that allows users to create a wide variety of diagrams, flowcharts, and visual representations. It's a web-based application, which means it can be accessed directly through a web browser without the need for installation. It also offer desktop versions for those who prefer offline use.

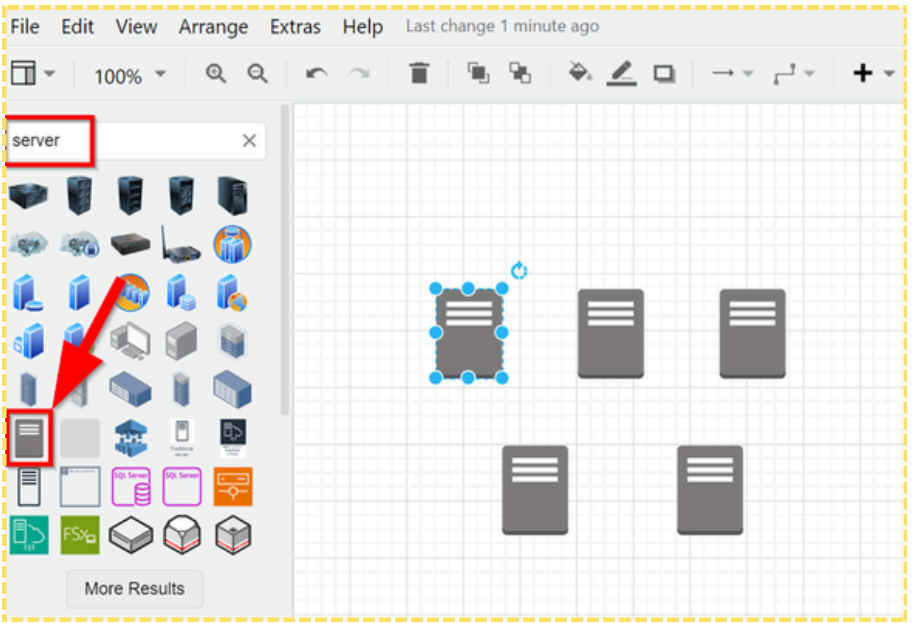
1. Set up Draw.io canvas

- Go to draw.io (https://app.diagrams.net/), you must log in first
- Click on "Create New Diagram"
- Choose "Blank Diagram",
- Rename diagram name to "30-day MyDFIR Diagram" and click "Create"



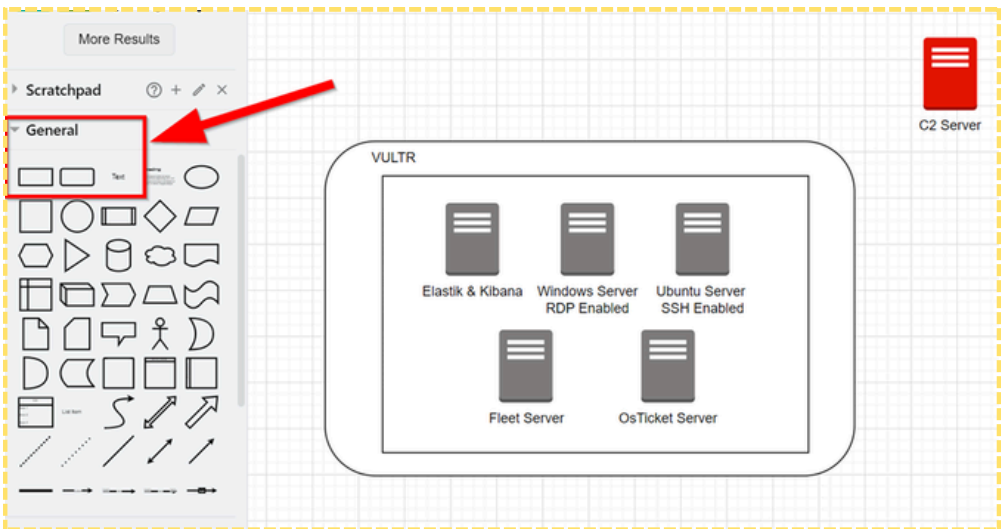
2. Add the server

- From the left sidebar, search for "server" in the shape library
- Drag and drop a server icon onto the canvas.
- Duplicate the server icon by selecting it, create a total of six servers
- Arrange the six server icons on the canvas



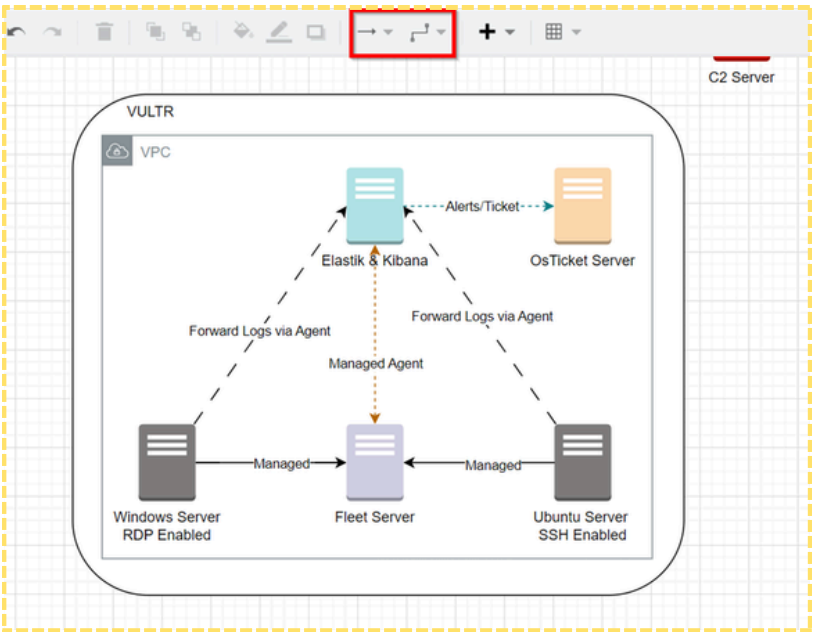
3. Create the Cloud Environment

- On the left-hand side, select the "Rounded Rectangle" shape.
- Draw a rectangle around the servers to represent the cloud provider.
- Select the "Text" tool from the left-hand side, type "VULTR" inside the rectangle.
- Right-click on the rectangle and select "To Back" to ensure the servers in front of the cloud provider rectangle.
- Label each server by selecting the "Text" tool and typing the server's name



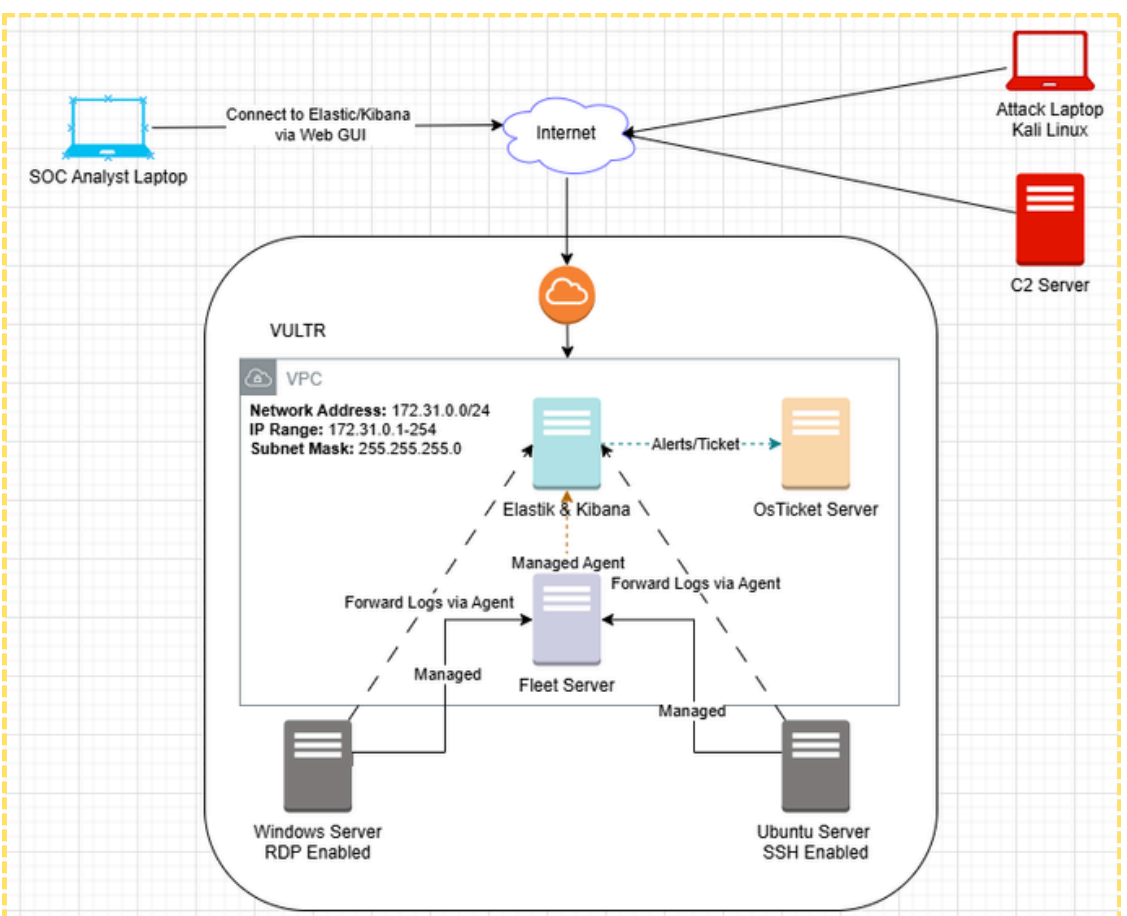
4. Connecting Components

- Use arrows to connect the Windows and Ubuntu servers to the Fleet Server.
- Connect the Fleet Server to Elastic and Kibana.
- Link OS Ticket to Elastic and Kibana.
- Connect Windows and Ubuntu servers directly to Elastic and Kibana for log forwarding.
- Adjust arrow styles for clarity
- Label connections ("managed," "forward logs via agent").



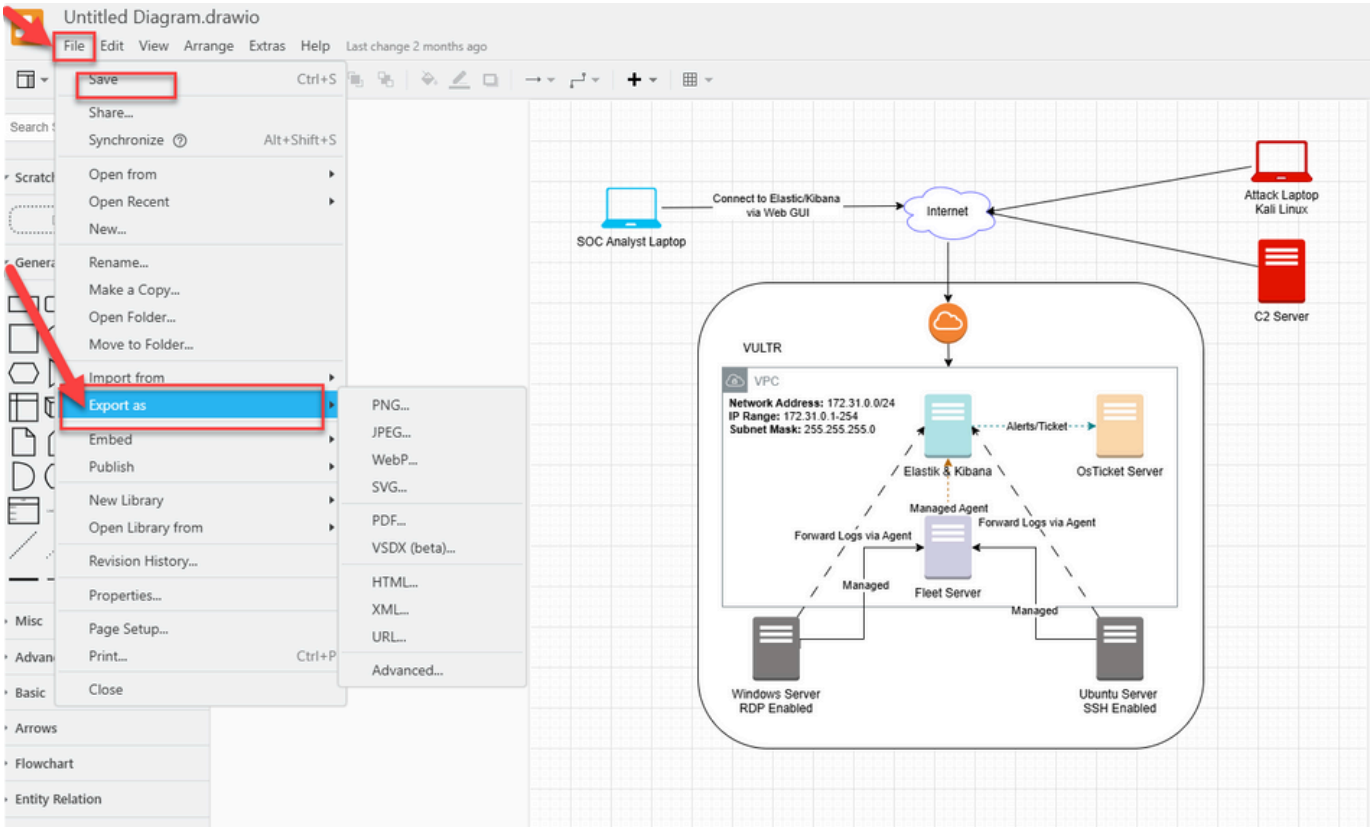
5. Adding the Analyst, Attacker Laptops and gateway internet

- Add a laptop icon for the SOC Analyst.
- Add another laptop icon for the Attacker, color the attacker laptop red.
- Search for "internet gateway" and drag the icon onto the canvas
- Connect both laptops to the internet cloud.
- Add a description for the SOC Analyst connection: "Connect to Elastic & Kibana via web GUI
- Add text to the arrow "connect to elastic cabana via web gui"



6. Saving and Exporting

- Click on "File" in the top left corner.
- Select "Save" to store diagram.
- Optionally, export the diagram in your preferred format (PNG, PDF,JPEG etc.) for sharing



7. Related Resources

- Video Tutorial: Refer back to the "How To Create a Logical Diagram | Day 1" video for a visual guide.
- draw.io Documentation: Check official draw.io documentation for more advanced features and troubleshooting.
- Community Forums: If you have any questions, consider posting them in related community forums where other participants might be able to assist.

Day 2 - ELK Stack Introduction

This challenge provide introduction to ELK Stack and designed to provide SOC analysts with **better understanding of what the ELK stack is and the benefits.** Its important to manage and analyze logs from different sources in real time for the health of system, security as well compliance requirements. its important to manage and check logs from different places in real time for the health of system, safety as well compliance needs. To meet this need, the ELK stack has become a very popular answer that gives a full set of tools for log management and analysis.

1. Understanding the ELK Stack

- The ELK Stack is a set of open source tools that work together.
- It has three main parts
  - Elasticsearch (E)
  - Logstash (L)
  - Kibana (K)
- The popularity of the ELK stack is based on its use as a central location for log data
- Used in IT operations (ITOps), security information and event management (SIEM) applications, business intelligence programs to extract informational nuggets from mountains of highly voluminous data.
- It helps to collect, process, store, search, and explore data from different sources.

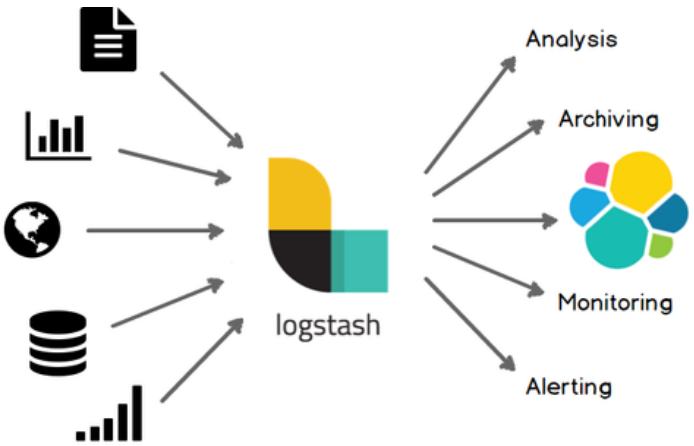
2. Components of the ELK Stack

2. 1 Elasticsearch

Elasticsearch is the core compnent of the ELK stack. Elasticsearch is an open source distributed, RESTful search and analytics engine, scalable data store, and vector database capable of addressing a growing number of use cases.

- Definition: Elasticsearch is a search and analytics engine that stores data.
- Function: It indexes data and allows you to search, analyze, and query it.
- Query Language: It uses ESQL (Elasticsearch Query Language) and a JSON-based language for searching.
- APIs: It uses RESTful APIs and JSON for communication.
- Access: It can access data through APIs, Kibana, or client libraries.

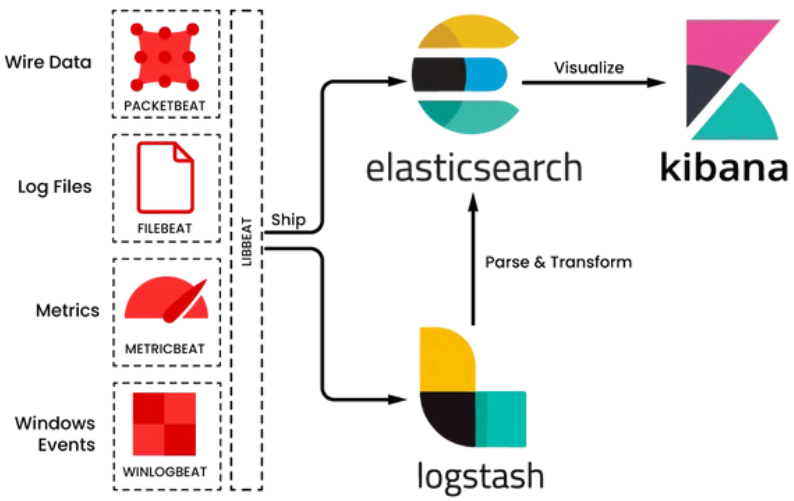
2. 2 Logstash



Logstash is data processing pipeline which ingests the logs then transforms it and lastly sends it over to centrally location. This data is merged from many places, transmuted and then pushed to Elasticsearch as a storage index. .

- Definition: Logstash is a data processing tool that collects, changes, and sends data to Elasticsearch.
- Function: It processes logs and other data from different sources.
- Data Collection: It collects data using agents like Beats or the Elastic Agent.
- Transformation: It changes and filters data using plugins.
- Output: It sends processed data to Elasticsearch.

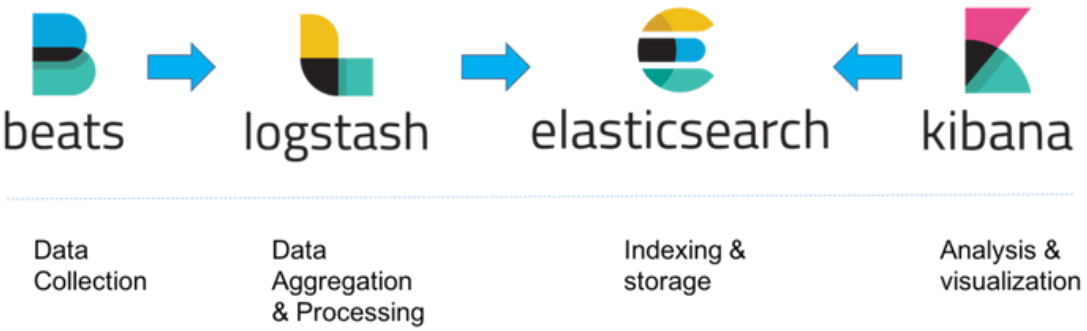
2. 3 Kibana



Run data analytics at speed and scale for observability, security, and search with Kibana. Powerful analysis on any data from any source, logs to application monitoring, and much more.

- Definition: Kibana is a tool for visualizing and exploring data in Elasticsearch.
- Function: It lets you query and visualize logs.
- Features: It includes tools like Kibana Lens, Discover, Machine Learning, Elastic Maps, Metrics Alerting, and Canvas.
- Purpose: It helps you search, visualize, and create reports from data.

3. Data Flow in the ELK Stack



3. 1 Data Collection: Data is collected using Beats or the Elastic Agent.

- Beats: light-weight data shippers that we can install on any system to collect some specific type of information, and then sends it to elasticsearch or Logstash assistant, there are 6 types
  - Filebeat: Collects log files.
  - Metricbeat: Collects system metrics.
  - Packetbeat: Captures network traffic.
  - Winlogbeat: Collects Windows event logs.
  - Auditbeat: Collects audit data from Linux systems.
  - Heartbeat: Monitors service availability.
- Elastic Agent: This is simpler because one agent can do many things and don't need to install many different beats.

3.2 Data Processing: Data is sent to Logstash, where it is changed and filtered.

3,3 Data Storage: Processed data is stored in Elasticsearch.

3.4 Data Visualization: Users can analyze data using Kibana.

4. Why Use ELK Stack

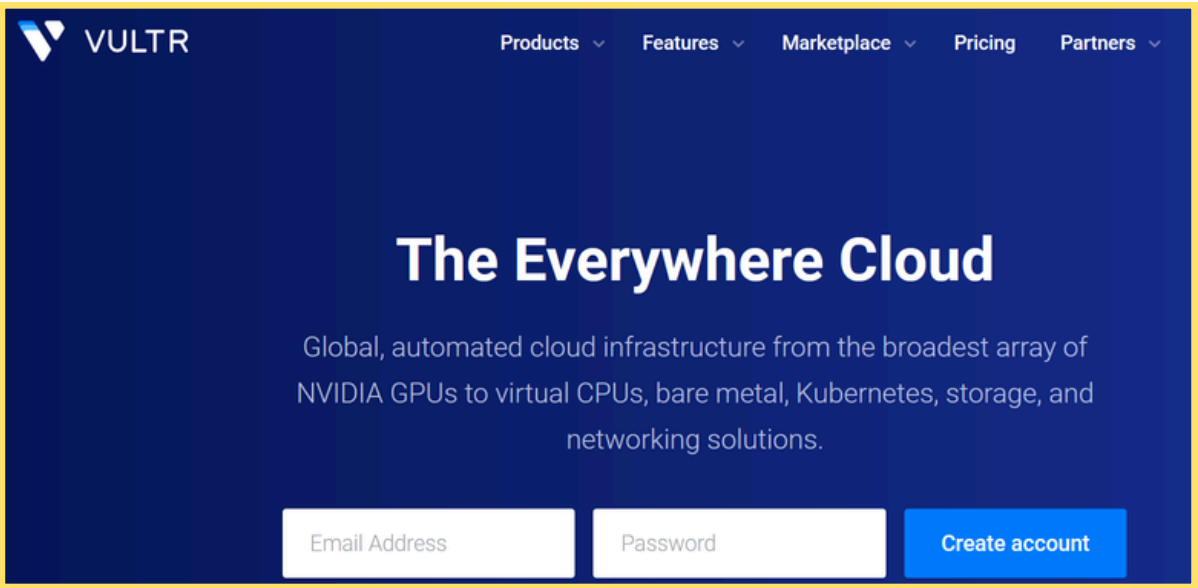
- All Logs in One Place
  - Helps with rules and with security when something happens
  - Centralized logging makes log management and analysis easier.
  - Good for following rules and regulations.
- Flexible
  - Can use Beats or the Elastic Agent
  - Can collect many types of data
  - Easy to change settings and works with many systems
- Visuals
  - Can make charts and dashboards that show important information
  - Easy to show and good for presentations
- Scalable
  - Can easily grow to handle more data by adding more nodes
  - Easy to add more storage and can handle lots of data
- Community
  - Is open-source and has a large community
  - Provides lots of resources, plugins, and support.
  - The large community provides a lot of help for users of the ELK Stack.

Day 3 - Elasticsearch Setup Tutorial

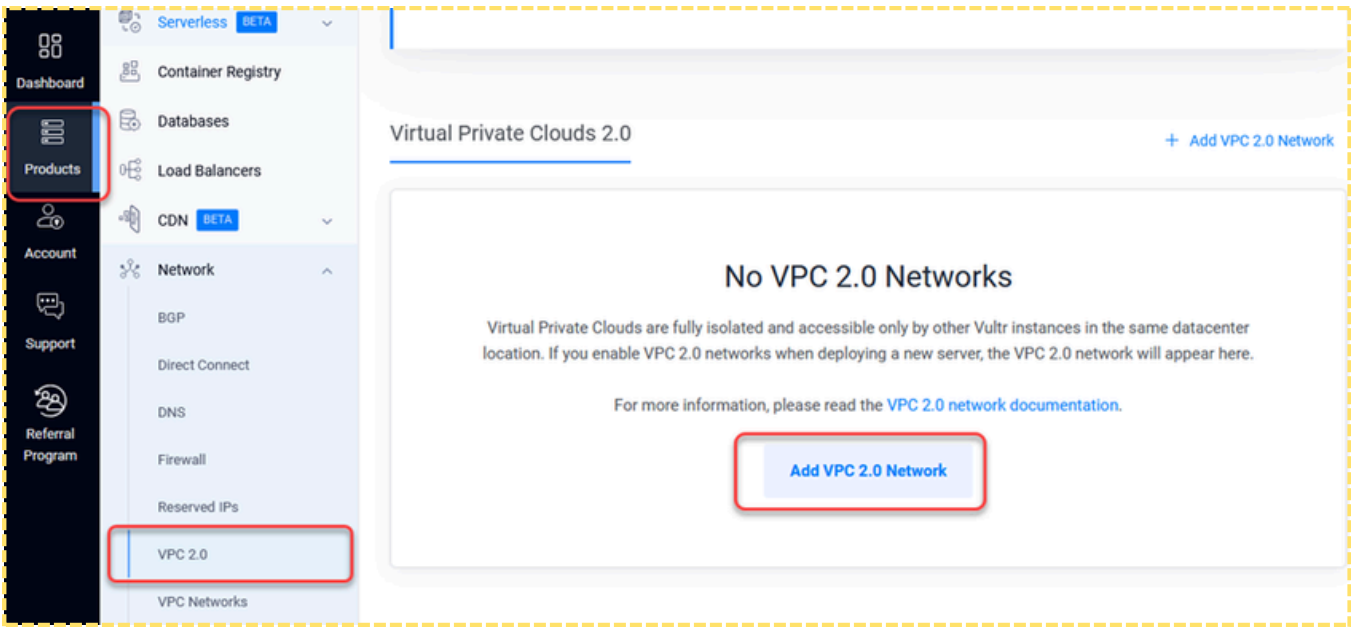
This challenge is designed to provide SOC analysts with **learn how to spin up Elasticsearch instance** using popular cloud hosting provider, **VULTR**. By the end, we will have a fully functional Elasticsearch instance ready for further exploration, data ingestion, and integration with other security tools and enabling you to simulate other challange.

1. Create Virtual Private Cloud (VPC)

- Visit VULTR page and sign up for an account (Vultr gives free credit $300)

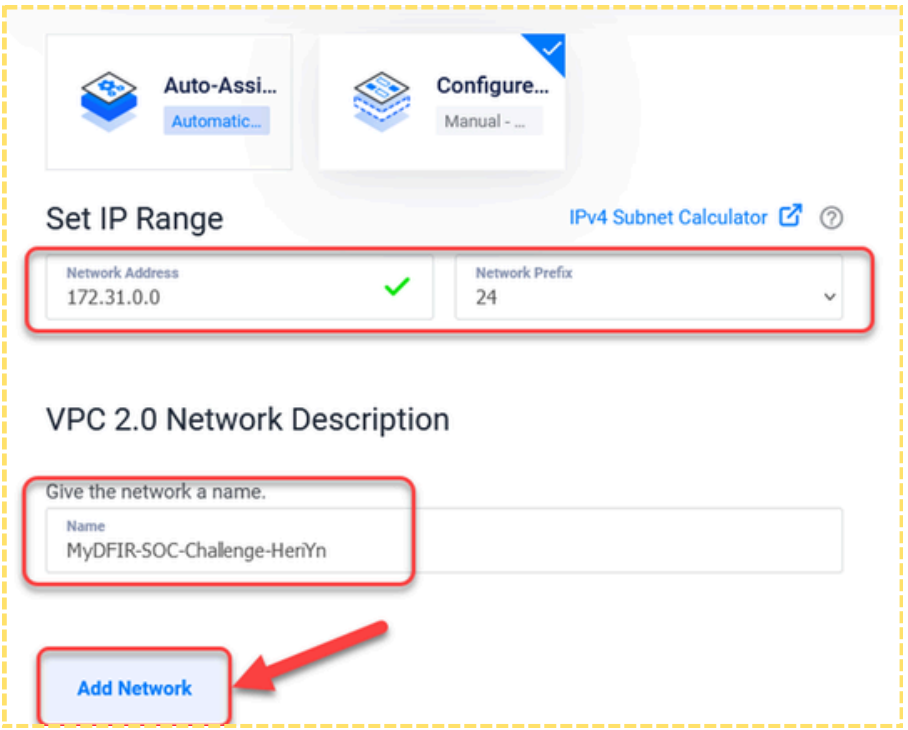


- After create account, Log in to Vultr
- Go to Products > Network > VPC 2.0
- Click on "Add VPC 2.0 Network", and choose the network location

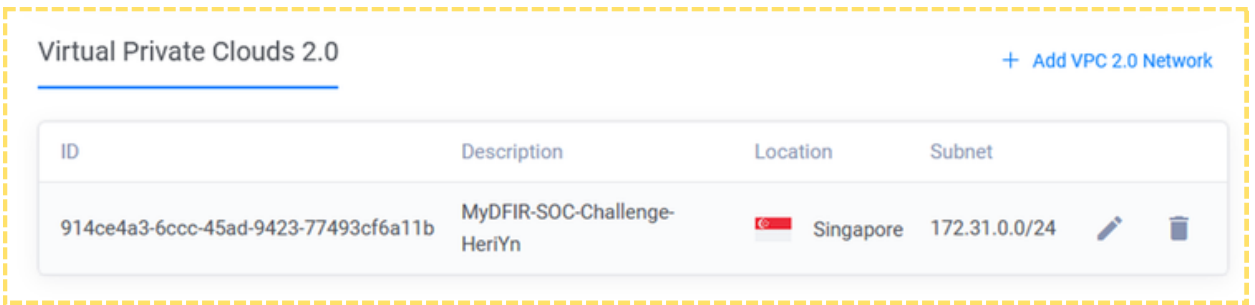


- Click Configure manual and Set the IPv4 range (172.31.0.0/24)
- Create network name ("MyDFIR SOC Challenge")

- To create VPC click on “Add Network”

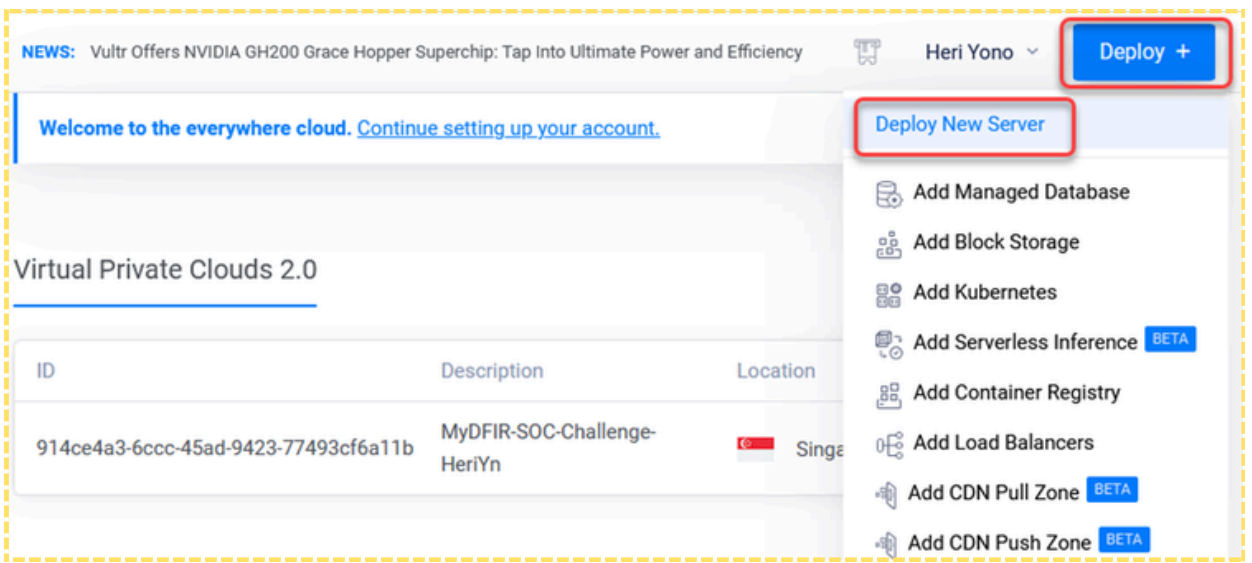


- VPC successfully created like this

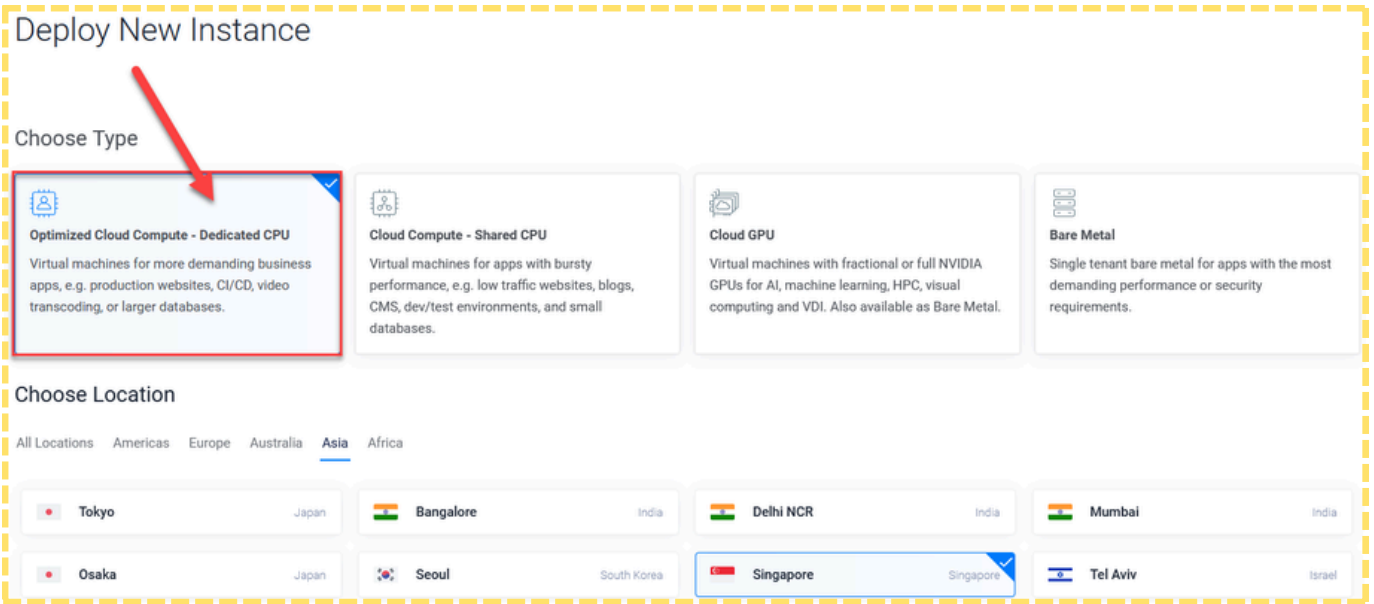


2. Deploying new Server

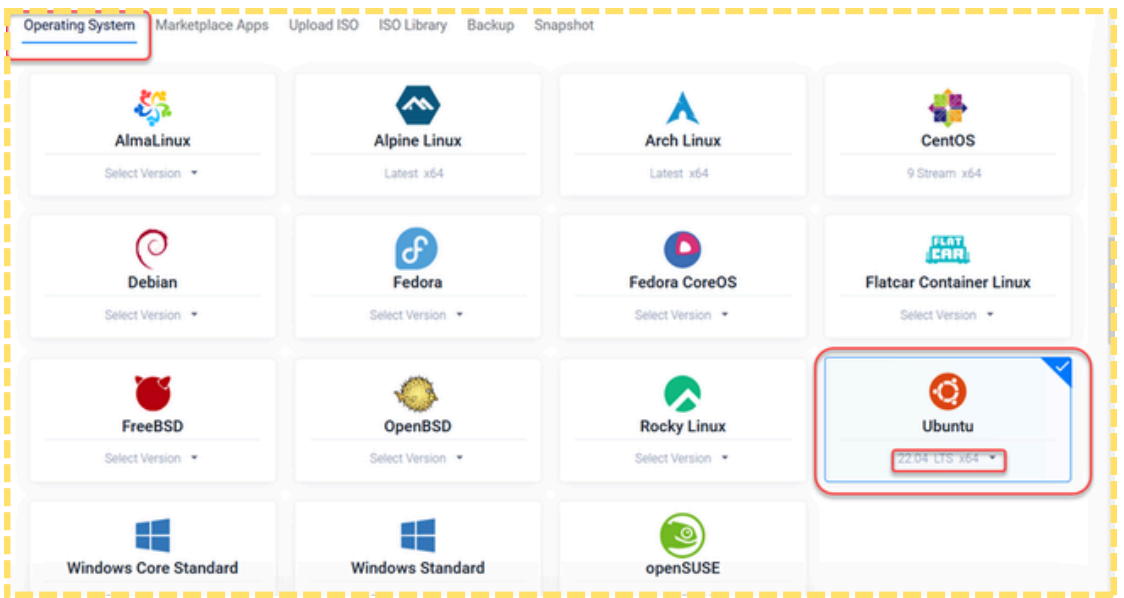
- Select "Deploy" at the top right corner and click "Deploy New Server."



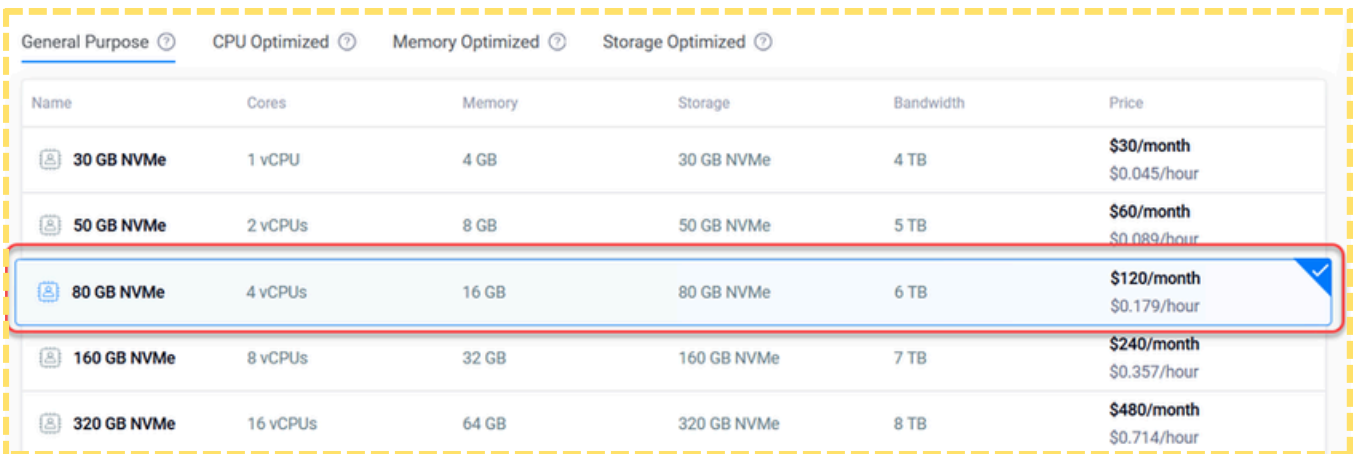
- Choose Type "Optimized Cloud Compute" and choose location (same as VPC location)



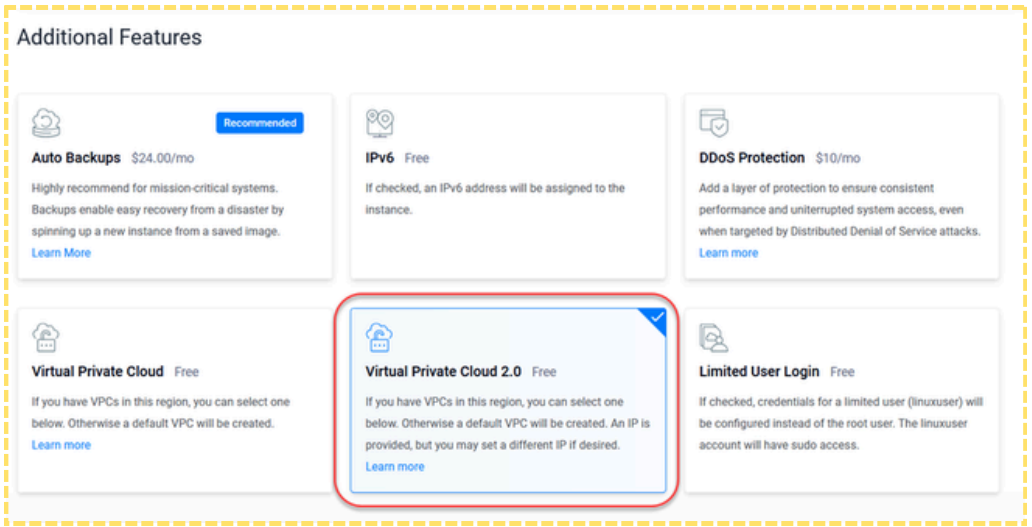
- Select Ubuntu 22.04 as the operating system



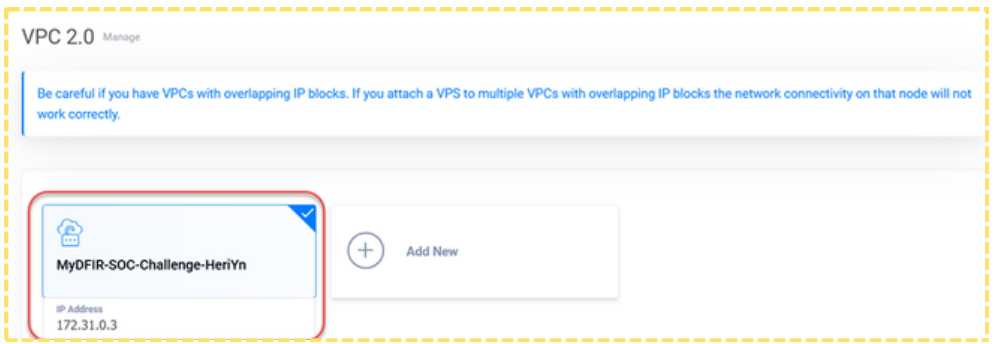
- Choose a plan with 80 GB NVMe, 4 vCPUs and 16GB RAM



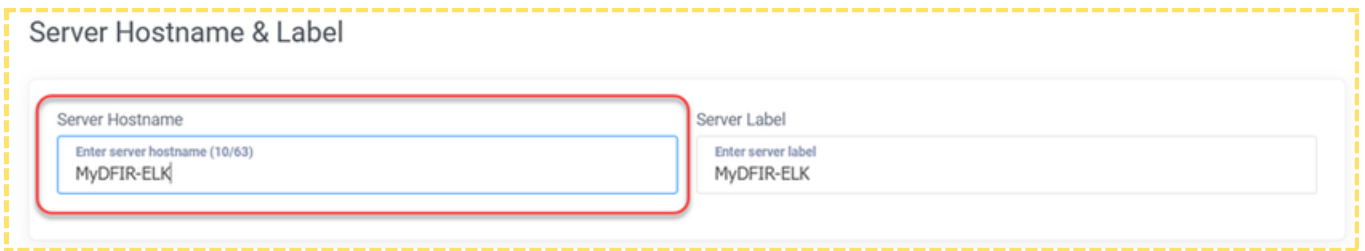
- Disable auto backups and IPv6, and select Virtual Private Cloud 2.0



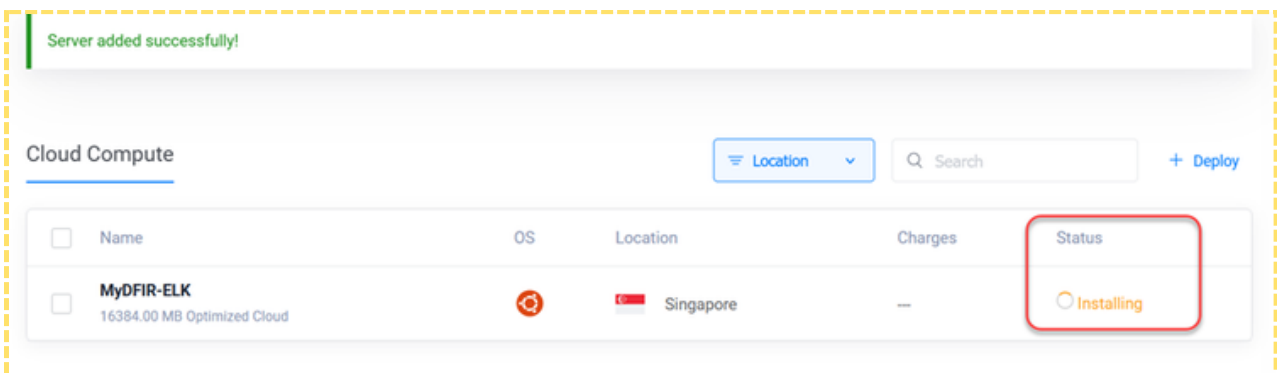
- Select your VPC 2.0 network



- Enter server hostname ("MyDFIR-ELK")



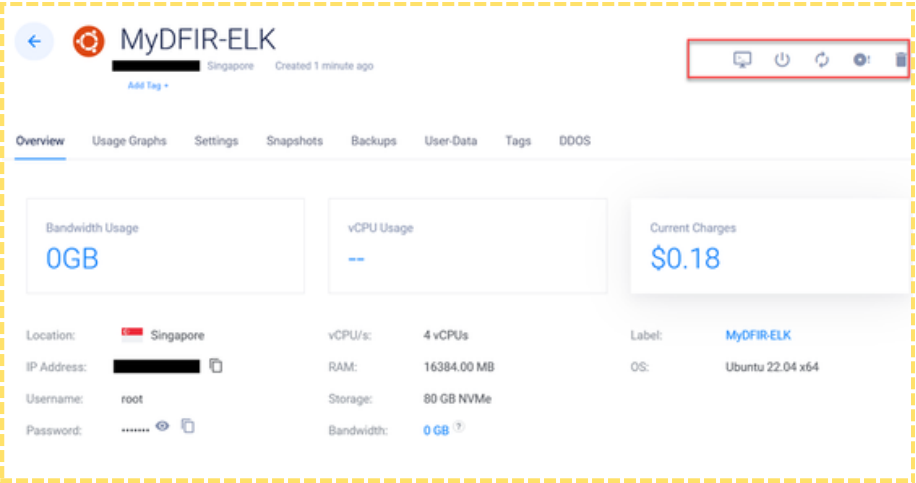
- Click "Deploy Now"



- Wait until the VM status changes to "running"

3. Accessing New Server ELK

- Click on the VM name ("MyDFIR-ELK") to see more details
- Note the server's public IP address.



- Go to terminal or powershell on local machine and SSH into the server ELK

```
ssh root@[your-server-ip]
```

```
Last login: Fri Sep  6 04:14:15 2024 from
root@MyDFIR-ELK:~#
```

4. Installing Elasticsearch

- Update Server Ubuntu `apt-get update && apt-get upgrade -y`

```
/usr/bin/xauth:  file /root/.Xauthority does not exist
root@MyDFIR-ELK:~# apt-get update && apt-get upgrade -y
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ubuntu.mirror.constant.com jammy InRelease
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://ubuntu.mirror.constant.com jammy-updates InRelease [128 kB]
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://archive.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,988 kB]
Hit:8 http://ubuntu.mirror.constant.com jammy-backports InRelease
Get:9 http://ubuntu.mirror.constant.com jammy-security InRelease [129 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.8 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,123 kB]
Get:12 http://ubuntu.mirror.constant.com jammy-updates/main amd64 Packages [1,988 kB]
```

- Open web browser and go to Elasticsearch download page "*elastic.co/downloads/elasticsearch*"
- Scroll down and choose platform **deb x86_64**
- Right-click on deb x86_64 download button and copy link address

- Paste and download Elasticsearch `"wget [Elasticsearch.deb package URL]"`

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@MyDFIR-ELK:~# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.15.0-amd64.deb
--2024-09-05 10:25:30--  https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.15.0-amd64.deb
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130, 2600:1901:0:1d7::
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 605981248 (578M) [application/vnd.debian.binary-package]
Saving to: 'elasticsearch-8.15.0-amd64.deb'

elasticsearch-8.15.0-amd64.deb    10%[=====>                                                    ]  61.92M  4.24
```

- Install Elasticsearch `"dpkg -i elasticsearch-[version].deb"`

```
root@MyDFIR-ELK:~# dpkg -i elasticsearch-8.15.0-amd64.deb
Selecting previously unselected package elasticsearch.
(Reading database ... 85487 files and directories currently installed.)
Preparing to unpack elasticsearch-8.15.0-amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (8.15.0) ...
Setting up elasticsearch (8.15.0) ...
--------------------------- Security autoconfiguration information ------------------------------

Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : [redacted]

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
after creating an enrollment token on your existing cluster.

You can complete the following actions at any time:
```

- Copy and save the security auto-configuration information displayed during installation
- If forget to copy the password, reset it using this command

`/usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic`

5. Configuring Elasticsearch

- Go to Elasticsearch configuration directory `cd /etc/elasticsearch`
- Edit elasticsearch configuration file `vi elasticsearch.yml`

```
# Use a descriptive name for your cluster:
#
#cluster.name: my-application
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
#node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ---------------------------------- Paths ----------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
"elasticsearch.yml" 119L, 4055B
```

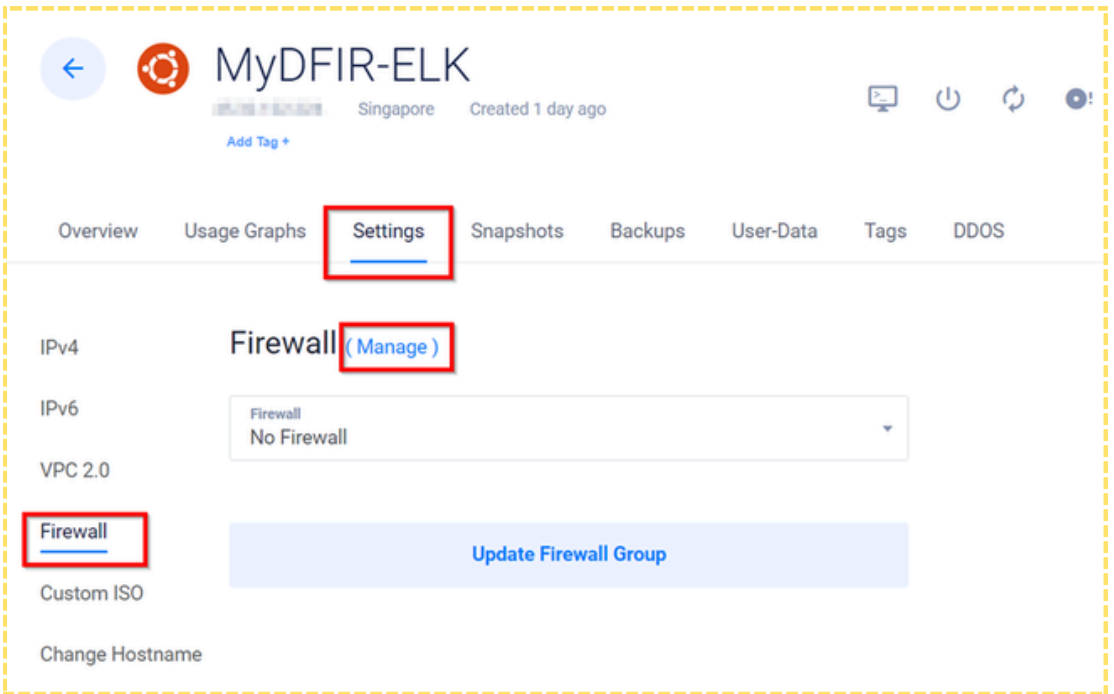
- Modify network host and replace the default IP
- Modify http port
- Save configuration file

```
network.host: [your-server-public-ip]
http.port: 9200
```

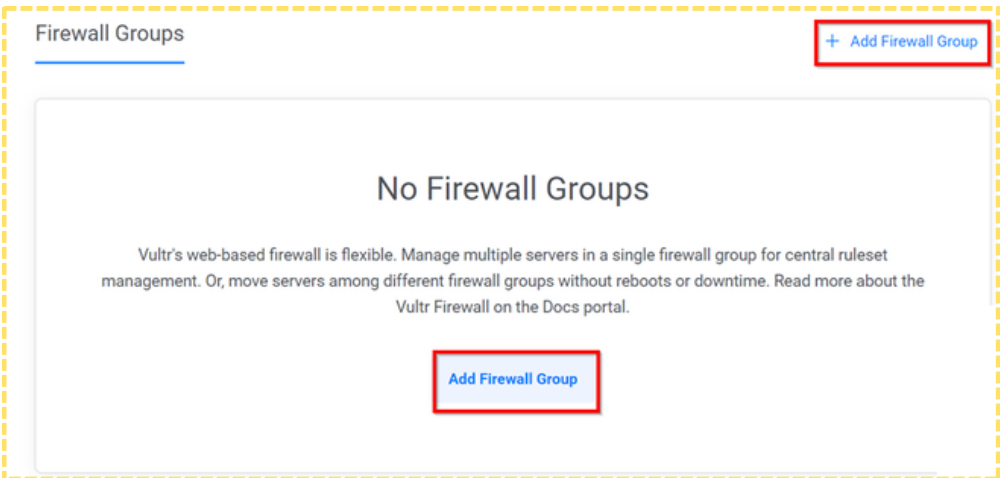
```
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: [redacted]
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
```

6. Firewall Settings Vultr

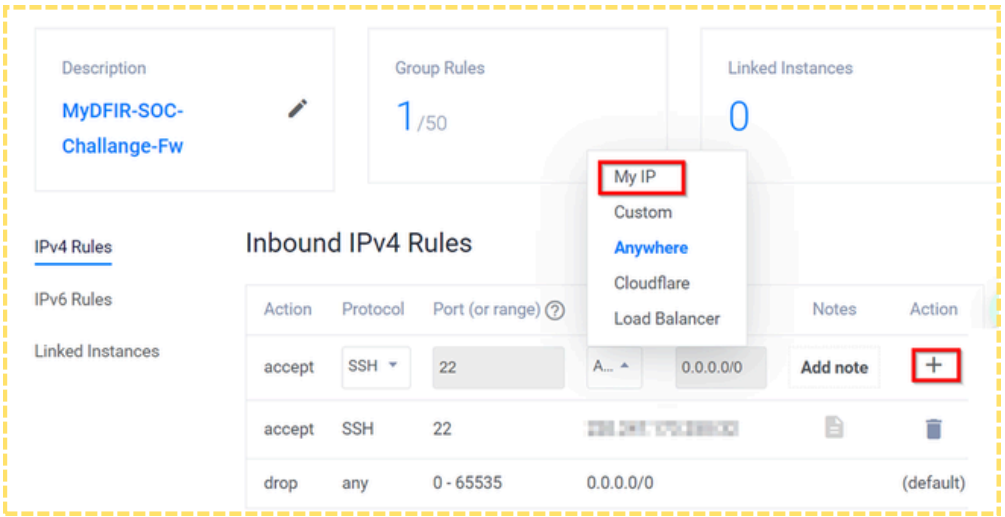
- Go back to Vultr Dashboard
- Navigate to VM settings, select Firewall and click (Manage) Firewall



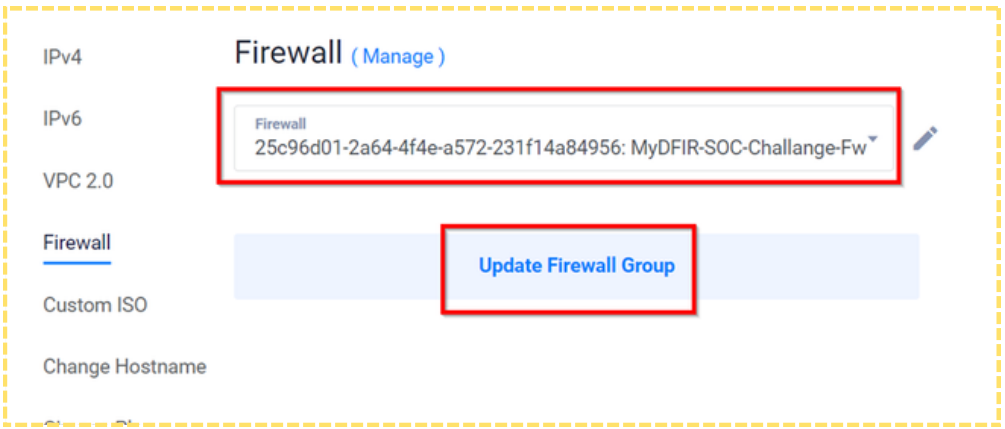
- Click Add Firewall Group and type description



- Modify the SSH rule to restrict access



- Go back to Server Setting and click Firewall
- Select the firewall group that was just created and click “Update Firewall Group”



7. Start Elasticsearch Service

- Before start, reload system daemon `systemctl daemon-reload`
- Enable Elasticsearch service `systemctl enable elasticsearch.service`

```
t@MyDFIR-ELK:/etc/elasticsearch# systemctl daemon-reload
t@MyDFIR-ELK:/etc/elasticsearch# systemctl enable elasticsearch.service
ated symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service
t@MyDFIR-ELK:/etc/elasticsearch#
```

- Start Elasticsearch service `systemctl start elasticsearch.service`
- Verify the service status `systemctl status elasticsearch.service`

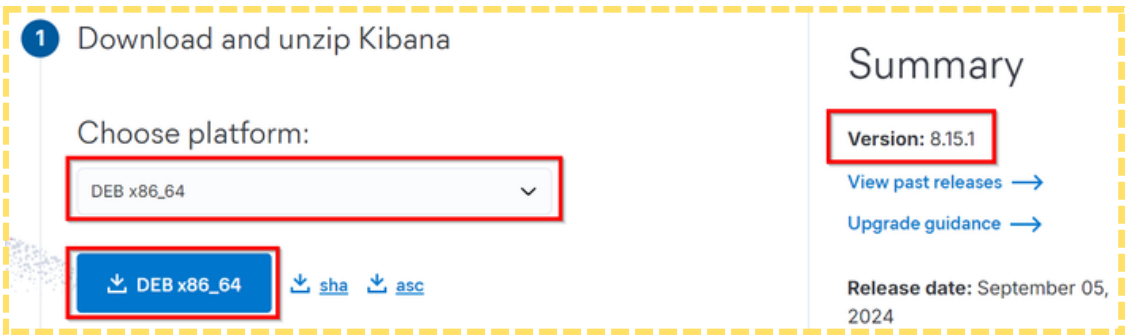
```
MyDFIR-ELK:/etc/elasticsearch# systemctl start elasticsearch.service
MyDFIR-ELK:/etc/elasticsearch# systemctl status elasticsearch.service
sticsearch.service - Elasticsearch
Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
Active: active (running) since Fri 2024-09-06 04:42:56 UTC; 22s ago
  Docs: https://www.elastic.co
in PID: 10869 (java)
 Tasks: 90 (limit: 19042)
Memory: 8.4G
   CPU: 41.747s
CGroup: /system.slice/elasticsearch.service
        ├─10869 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/u
        ├─10928 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negati
        └─10951 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

6 04:42:43 MyDFIR-ELK systemd[1]: Starting Elasticsearch...
```

Day 4 - Kibana Setup Tutorial

This challenge is designed to provide SOC analysts with **learn how to install Kibana instance** using popular cloud hosting provider, **VULTR**. This challenge will cover downloading, installing, configuring, and troubleshooting common issues encountered during the setup process. By the end, Kibana instance connected to Elasticsearch deployment and will be ready to use.

1. Download and Install Kibana

- Get Kibana by downloading it from the official Elastic page website
- Choose DEB x86_64, right click the blue button and copy link address.



- Paste and download the Kibana Debian package `wget [Kibana.deb package URL]`

```
root@MyDFIR-ELK:~# wget https://artifacts.elastic.co/downloads/kibana/kibana-8.
--2024-09-06 09:49:12--  https://artifacts.elastic.co/downloads/kibana/kibana-8
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130, 2600:1
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443..
HTTP request sent, awaiting response... 200 OK
Length: 340199408 (324M) [application/vnd.debian.binary-package]
Saving to: 'kibana-8.15.1-amd64.deb'

kibana-8.15.1-amd64  24%[===>                 ]  78.84M  3.43MB/s    eta 73s
```

- Replace <Kibana package URL> with the actual link copied from the Elastic website

2. Installing Kibana Package

- Once the download is complete, use ***ls*** to verify kibana package
- Install kibana package using `dpkg -i kibana-[version].deb`

```
root@MyDFIR-ELK:~# ls
elasticsearch-8.15.0-amd64.deb  kibana-8.15.1-amd64.deb  snap
root@MyDFIR-ELK:~# dpkg -i kibana-8.15.1-amd64.deb
Selecting previously unselected package kibana.
(Reading database ... 86887 files and directories currently installed.)
Preparing to unpack kibana-8.15.1-amd64.deb ...
Unpacking kibana (8.15.1) ...
```

- After a few minute, Kibana should be installed

```
root@MyDFIR-ELK:~# ls -l /etc/kibana/
total 16
-rw-rw---- 1 root kibana  130 Sep  6 10:08 kibana.keystore
-rw-rw---- 1 root kibana 7822 Sep  2 12:26 kibana.yml
-rw-rw-r-- 1 root kibana  447 Sep  2 12:26 node.options
root@MyDFIR-ELK:~# systemctl status kibana.service
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; disabled; vendor prese>
     Active: active (running) since Fri 2024-09-06 14:00:10 UTC; 2 days ago
       Docs: https://www.elastic.co
   Main PID: 12063 (node)
```

3. Configuring Kibana

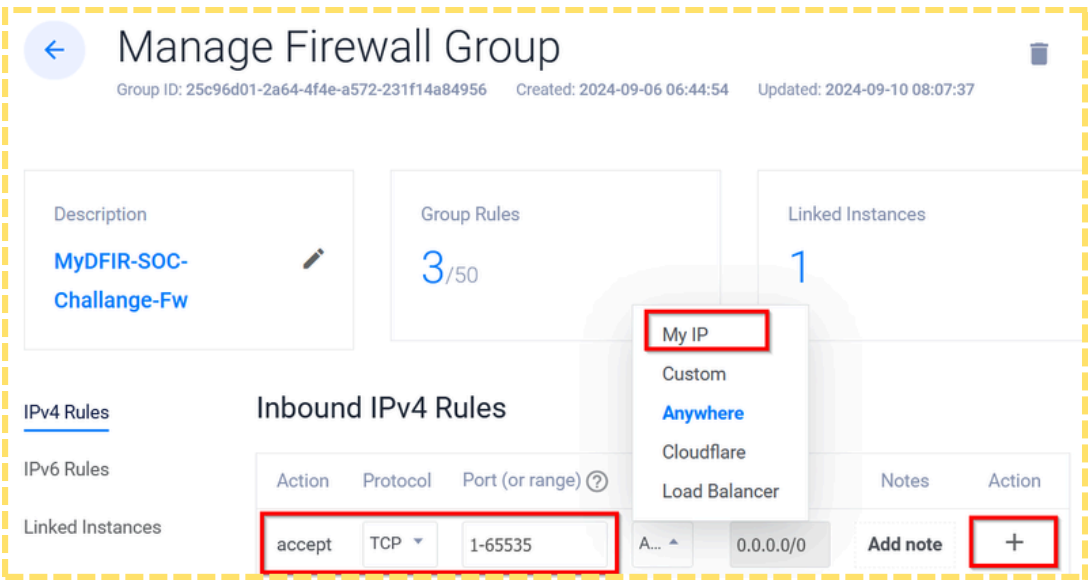
- Edit the Kibana configuration file `vi /etc/kibana/kibana.yml`
- Remove comment (#) from the server.port line and server.host line
- Set server.port line to 5601
- Replace server.host line with the public IP address server

```
# ================== System: Kibana Server ==================
# Kibana is served by a back end server. This setting specifies the port to use

server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and
host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be a
ble to connect.
# To allow connections from remote users, set this parameter to a non-loopback
address.
server.host: "ip public elk server"

# Enables you to specify a path to mount Kibana at if you are running behind a
proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove t
he basePath
```

- Save and exit kibana.yml file

4. Firewall Settings

- Go to Vultr instance firewall settings
- Add a new rule to allow TCP protocol on port 1-65535 from inbound public IP address



- On ubuntu VM server, allow connections on port 5601

```
root@MyDFIR-ELK:~# ufw allow 5601
Rule added
Rule added (v6)
root@MyDFIR-ELK:~#
```

- If ufw is not enabled, try to enable it with ufw enable

5. Starting Kibana Service

- Reload the system daemon `systemctl daemon-reload`
- Enable the Kibana service `systemctl enable kibana.service`
- Start the Kibana service `systemctl start kibana.service`

```
root@MyDFIR-ELK:~# systemctl daemon-reload
root@MyDFIR-ELK:~# systemctl enable kibana.service
Created symlink /etc/systemd/system/multi-user.target.wa
ib/systemd/system/kibana.service.
root@MyDFIR-ELK:~# systemctl start kibana.service
```

- Verify status Kibana is active (running)

```
root@MyDFIR-ELK:~# systemctl status kibana.service
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; vendor prese>
     Active: active (running) since Fri 2024-09-06 14:00:10 UTC; 2 days ago
       Docs: https://www.elastic.co
   Main PID: 12063 (node)
      Tasks: 11 (limit: 19042)
     Memory: 240.2M
        CPU: 2min 29.891s
     CGroup: /system.slice/kibana.service
             └─12063 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/sha>
```

6. Elasticsearch Enrollment Token

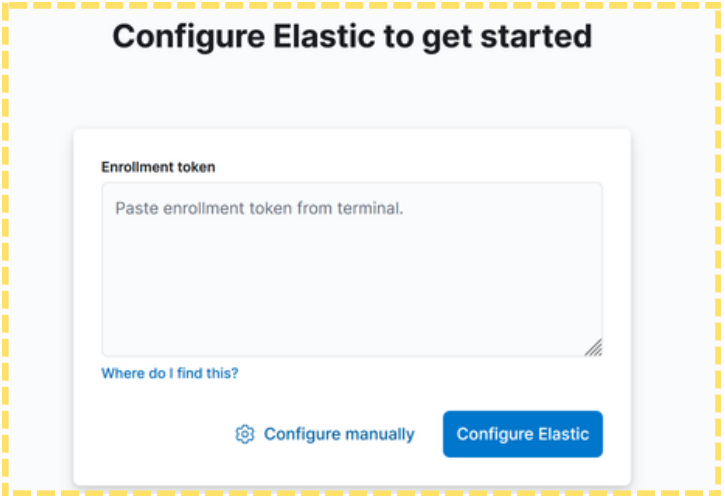
- To connect Kibana with Elasticsearch, we need an enrollment token
- Change directory to Elasticsearch bin folder `cd /usr/share/elasticsearch/bin`
- Generate enrollment token `./elasticsearch-create-enrollment-token --scope kibana`

```
root@MyDFIR-ELK:/# cd /usr/share/elasticsearch/bin/
root@MyDFIR-ELK:/usr/share/elasticsearch/bin# ls elasticsearch-create-enrollment-token -l
-rwxr-xr-x 1 root root 353 Aug  5 10:08 elasticsearch-create-enrollment-token
root@MyDFIR-ELK:/usr/share/elasticsearch/bin# ./elasticsearch-create-enrollment-token --sco
[redacted]
root@MyDFIR-ELK:/usr/share/elasticsearch/bin#
```

- Save result generate enrollment token in notepad

7. Accessing Kibana

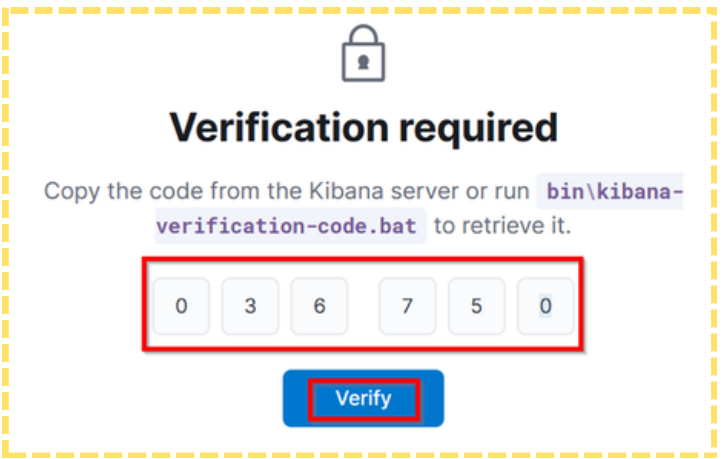
- Open kibana in browser, enter **http://[IP Server ELK]:5601**
- Paste the Elasticsearch enrollment token that saved earlier and click Configure Elastic



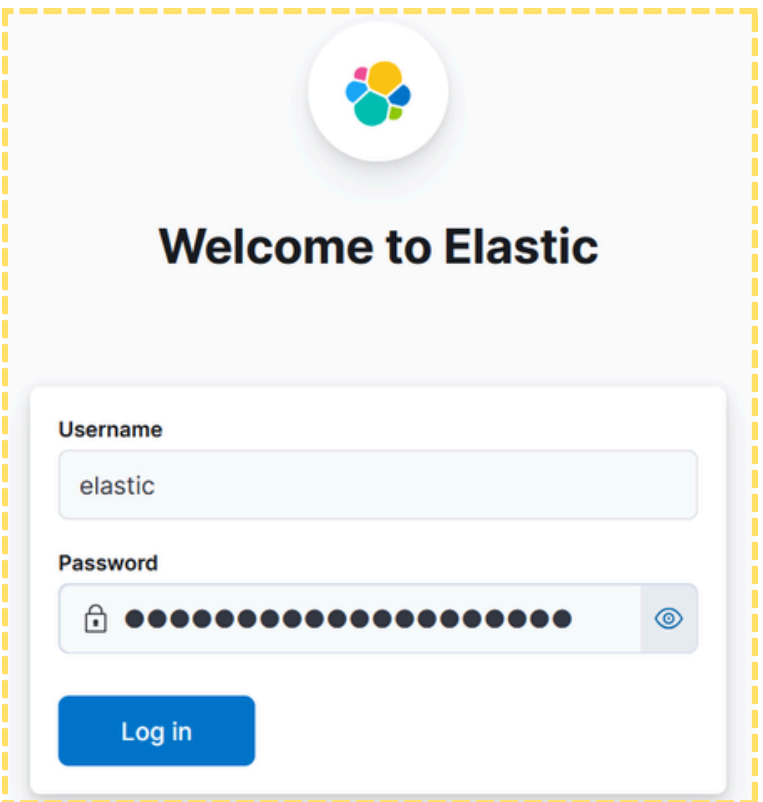
- Verification code will be prompted and next go back to SSH session
- Navigate to folder kibana bin using `cd /usr/share/kibana/bin/`
- Run verification code command on server vm ELK `./kibana-verification-code`
- Copy verification code

```
root@MyDFIR-ELK:~# cd /usr/share/kibana/bin/
root@MyDFIR-ELK:/usr/share/kibana/bin# ls
kibana  kibana-encryption-keys  kibana-health-gateway  kibana-keystore  kibana-plug
root@MyDFIR-ELK:/usr/share/kibana/bin# ./kibana-verification-code
Your verification code is:  036 750
root@MyDFIR-ELK:/usr/share/kibana/bin#
```

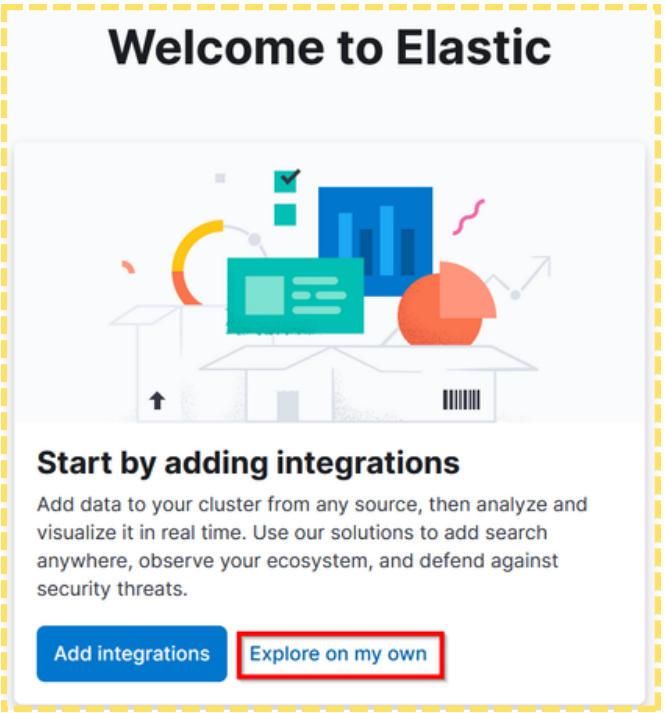
- Paste and enter the verification code into the Kibana web GUI



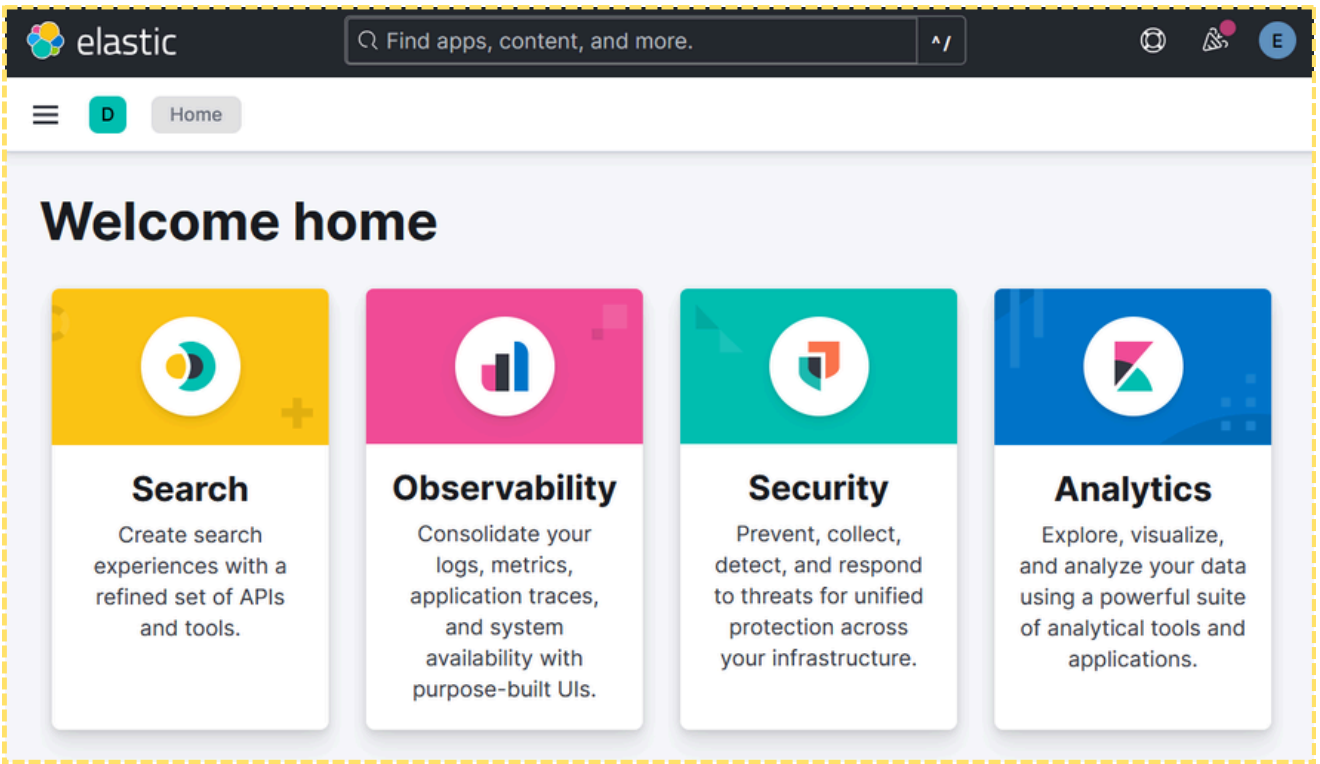
- Kibana will redirect to the login page after process was success
- Use default username and password generated during installation



- On elastic welcome page click **Explore on my own**

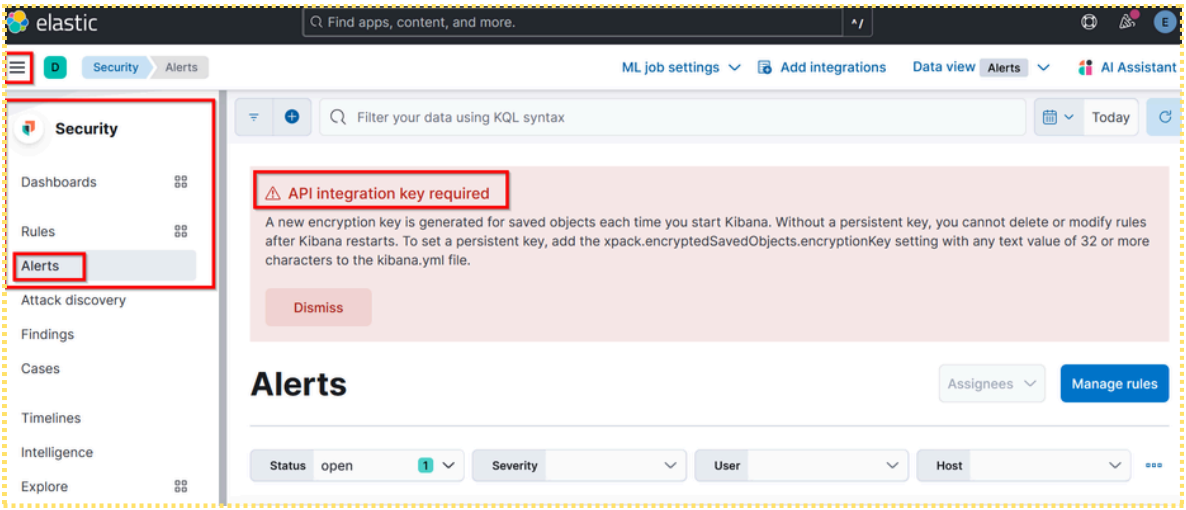


- Will appear the Elastic welcome home page.



5. Setting Up Integration Keys

- Select Alert section under Security menu, there's warning message "API integration key required"



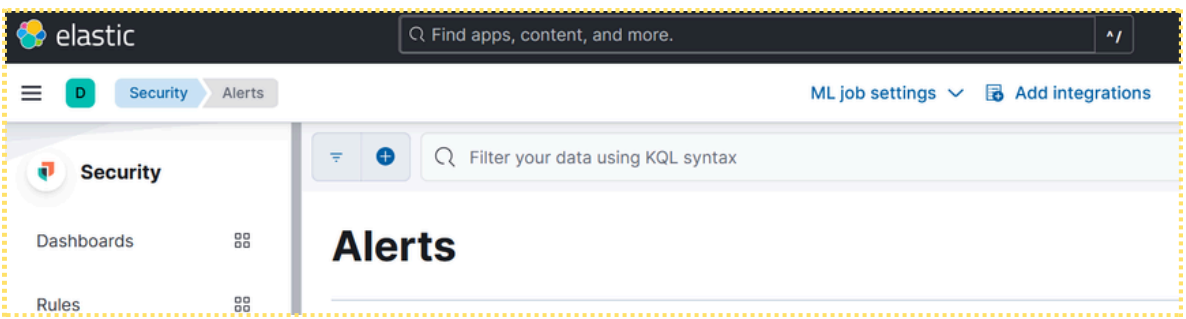
- Change to directory kibana bin
- Generate encryption key `./kibana-encryption-keys generate`

```
root@MyDFIR-ELK:/usr/share/kibana/bin# ls
kibana                   kibana-keystore  kibana-verification-code
kibana-encryption-keys   kibana-plugin
kibana-health-gateway    kibana-setup
root@MyDFIR-ELK:/usr/share/kibana/bin# ./kibana-encryption-keys generate
## Kibana Encryption Key Generation Utility

The 'generate' command guides you through the process of setting encryption key
s for:

xpack.encryptedSavedObjects.encryptionKey
    Used to encrypt stored objects such as dashboards and visualizations
    https://www.elastic.co/guide/en/kibana/current/xpack-security-secure-saved-
objects.html#xpack-security-secure-saved-objects
```

- Copy the generated keys to text editor
- Add the keys to Kibana keystore

```
root@MyDFIR-ELK:/usr/share/kibana/bin# ls
kibana                   kibana-keystore  kibana-verification-code
kibana-encryption-keys   kibana-plugin
kibana-health-gateway    kibana-setup
root@MyDFIR-ELK:/usr/share/kibana/bin# ./kibana-keystore add xpack.encryptedSavedObjects.encryptionKey
Enter value for xpack.encryptedSavedObjects.encryptionKey: ********************************
root@MyDFIR-ELK:/usr/share/kibana/bin# ./kibana-keystore add xpack.reporting.encryptionKey
Enter value for xpack.reporting.encryptionKey: ********************************
root@MyDFIR-ELK:/usr/share/kibana/bin# ./kibana-keystore add xpack.security.encryptionKey
Enter value for xpack.security.encryptionKey: ********************************
root@MyDFIR-ELK:/usr/share/kibana/bin#
```

- Refresh web page and login again to elastic. Currently no warning messsage displayed.



6. Troubleshooting

- Connection Timed Out:
  - Check cloud provider or VM firewall rules.
  - Check Ubuntu firewall rules using ufw status.
  - Check that are using correct IP and port
  - Verify Elasticsearch services are running.
- Kibana Not Starting:
  - Check Kibana logs errors.

    ```
    journalctl -u kibana.service
    ```
  - Verify the configuration file kibana is correct.

    ```
    /etc/kibana/kibana.yml
    ```
  - Ensure kibana status is active running.

    ```
    systemctl status kibana.service
    ```
- Enrollment Token Issues:
  - Ensure the correct token is being used.
  - Regenerate the token if necessary.

    ```
    ./elasticsearch/bin/elasticsearch-create-enrollment-token --scope kibana
    ```
- API Integration Key Error:
  - Ensure the encryption keys are correctly added to key store.
  - Restart Kibana after adding the keys.

    ```
    systemctl restart kibana.service
    ```
- Login Issues:
  - Verify the username and password are correct.
  - If forgot the password, reset password using this command

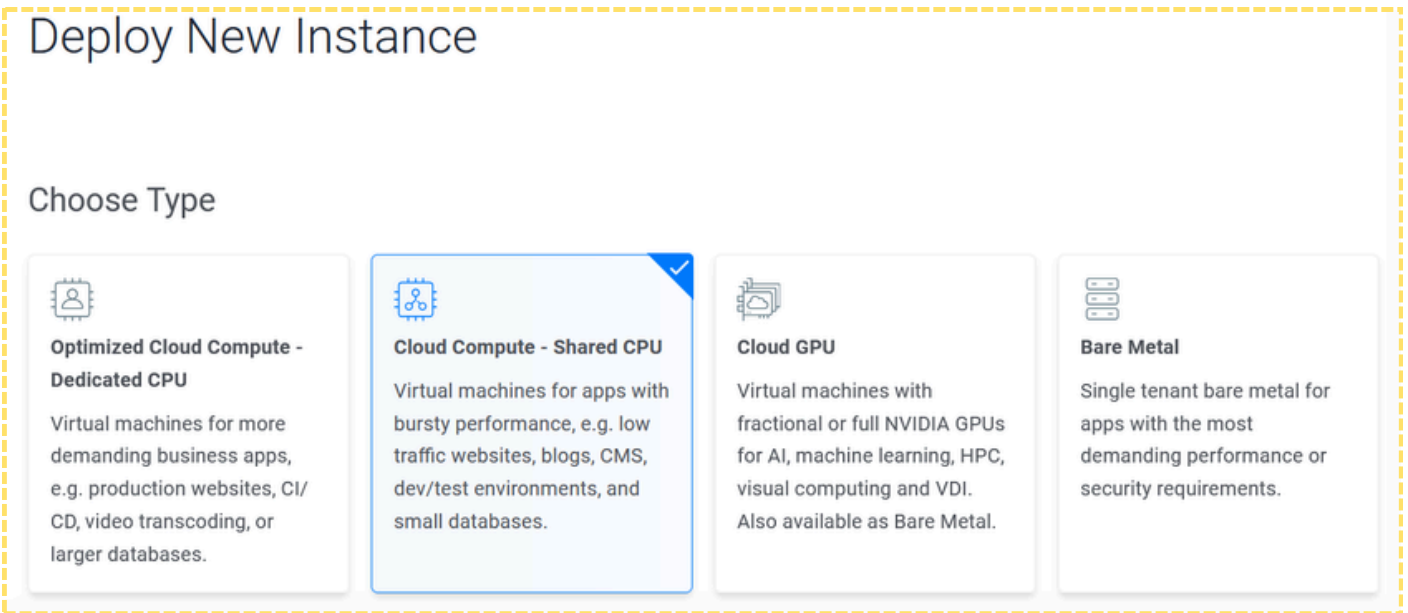
    ```
    /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic
    ```

Day 5 - Windows Server Installation

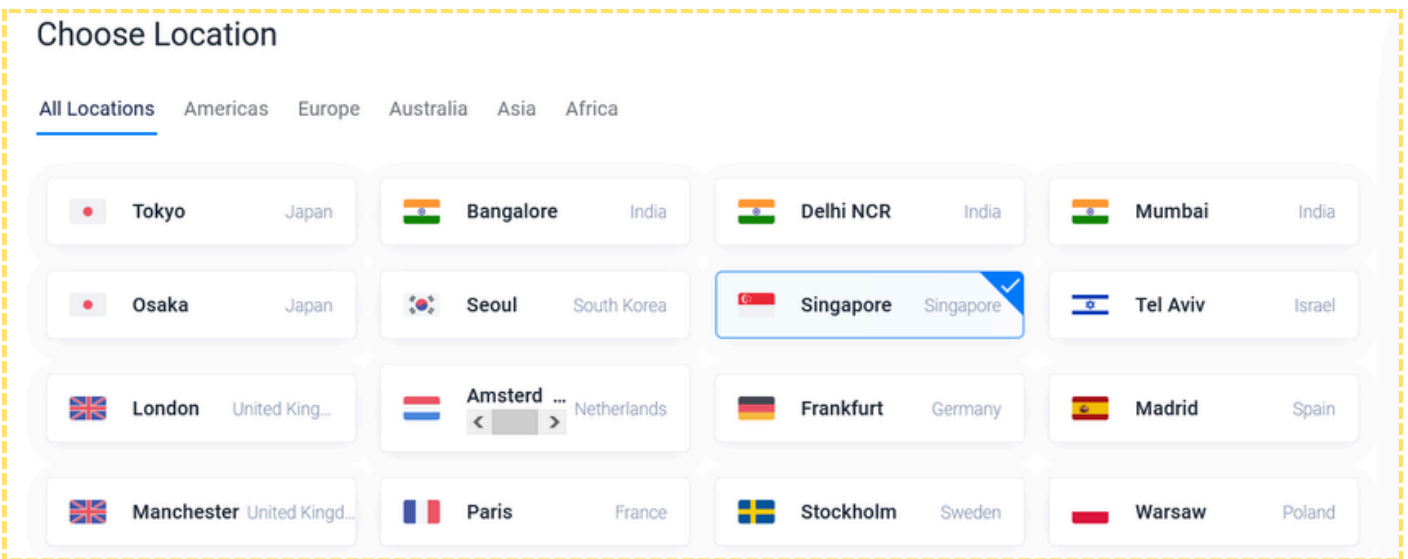
This challenge is designed to provide SOC analysts with **learn how to install Windows Server 2022** in the cloud hosting provider, **VULTR with RDP exposed to the internet.** This challenge will cover downloading, installing, configuring, and troubleshooting common issues encountered during the setup process.
By the end, Windows Server 2022 instance can accessible via Remote Desktop Protocol (RDP) over the internet for getting RDP login attempts for seeing real-world attack. Monitor login attempts from different IP addresses will help understand security threats and will give real data to analyze. For example a bunch of failed login attempts from one IP which could be brute-force attack.

1. Deploy New Server

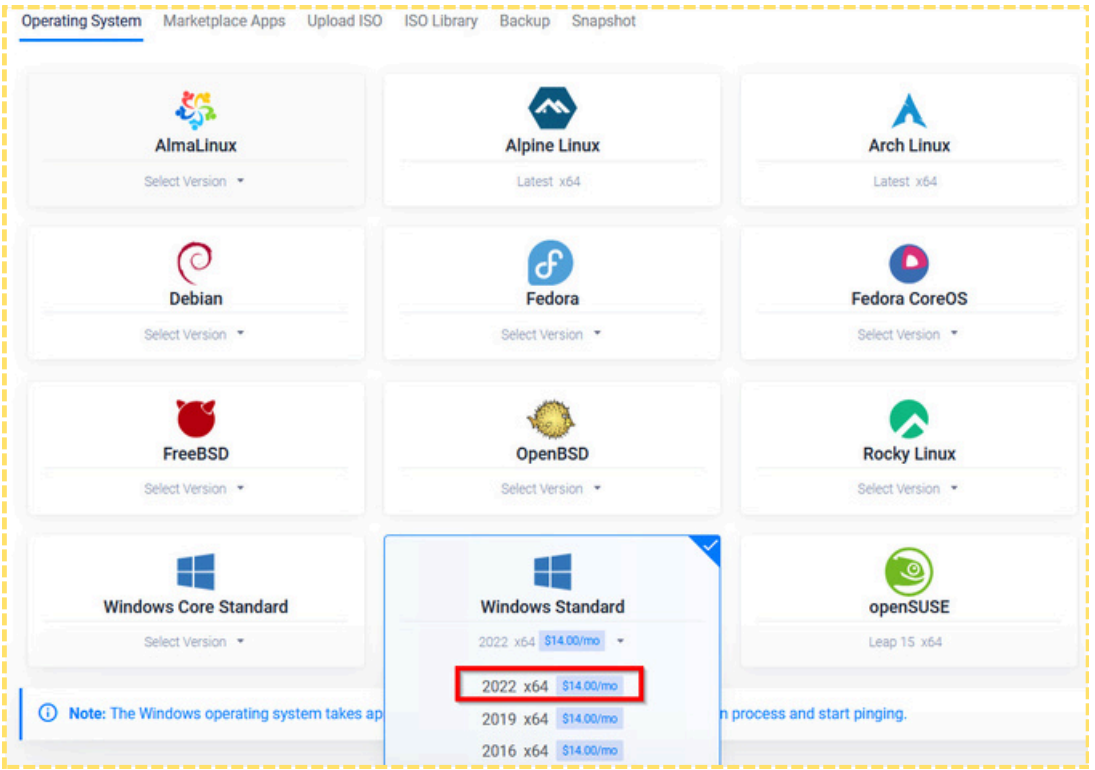
- Login to Vultr and select Deploy New Server
- For Type Instance, choose **Cloud Compute-Shared CPU**



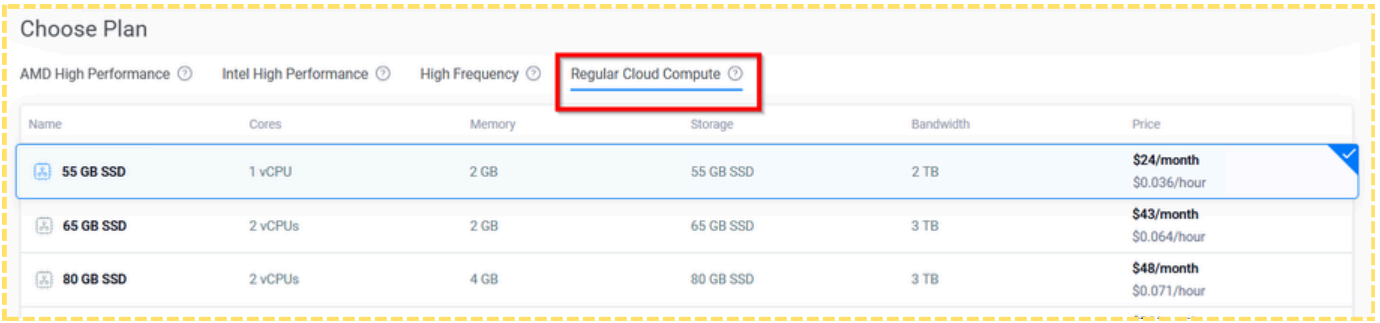
- Select preferred location



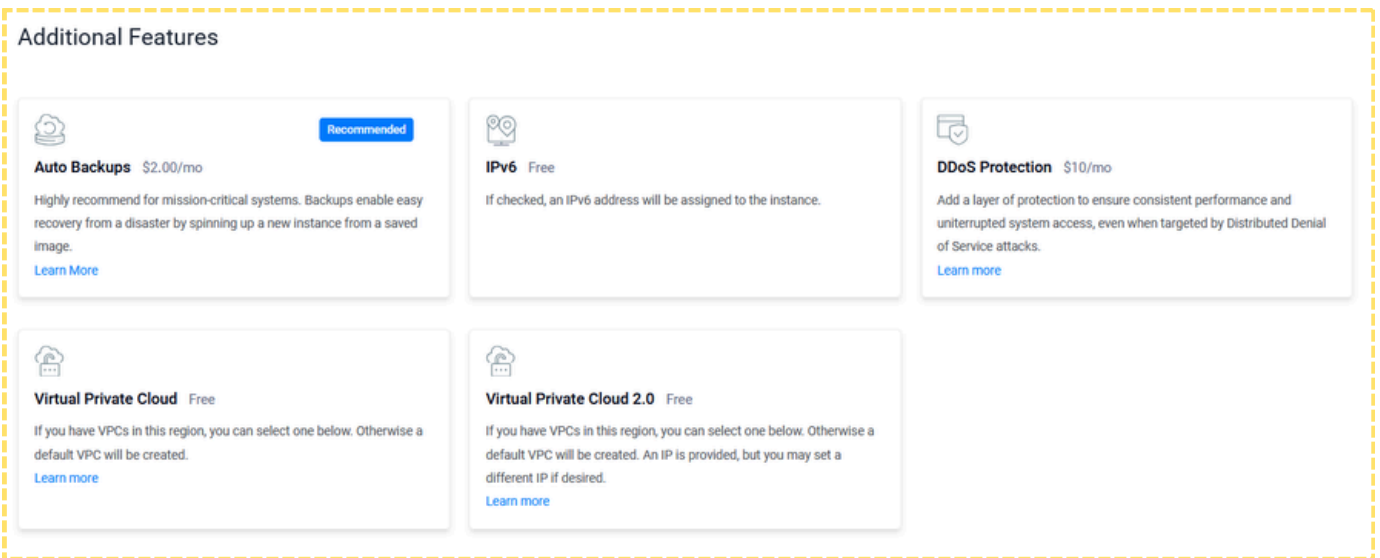
- Under Image choose "**Windows Standard**" and select "**2022 x64**"



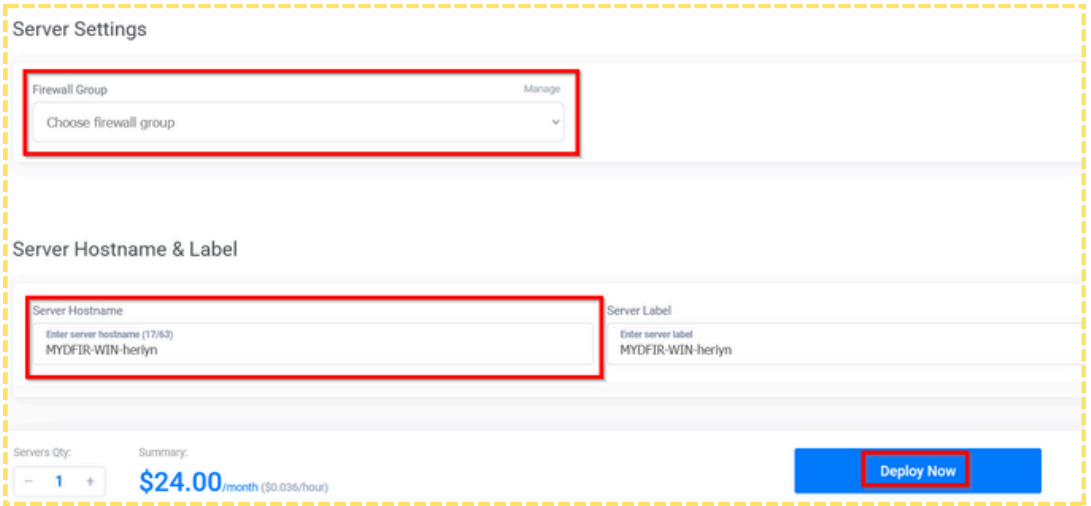
- Click tab “Regular Cloud Compute” and choose plan the cheapest option



- Turn off Auto Backups and IPv6 to save credits for learning

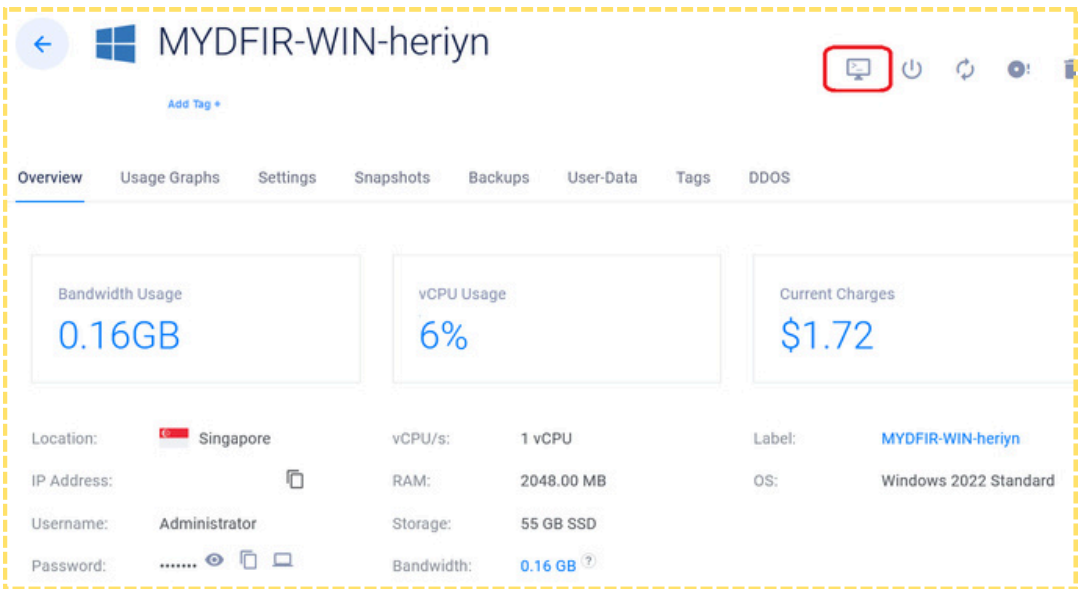


- Leave firewall group blank for now
- Type for server hostname & Label and Click "**Deploy Now**"



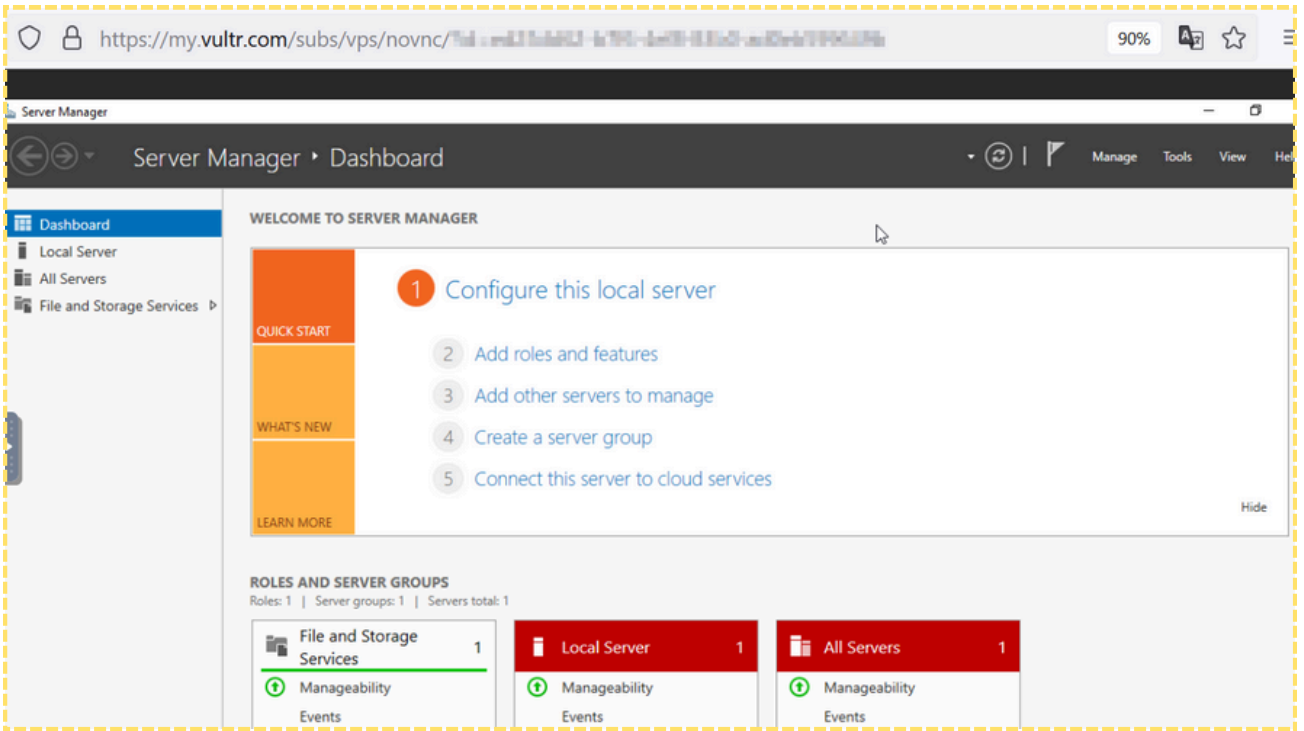
2. Accessing Windows Server Instance

- After take a few minutes, server status changes to "Running"
- Click on the server instance and select "View Console”



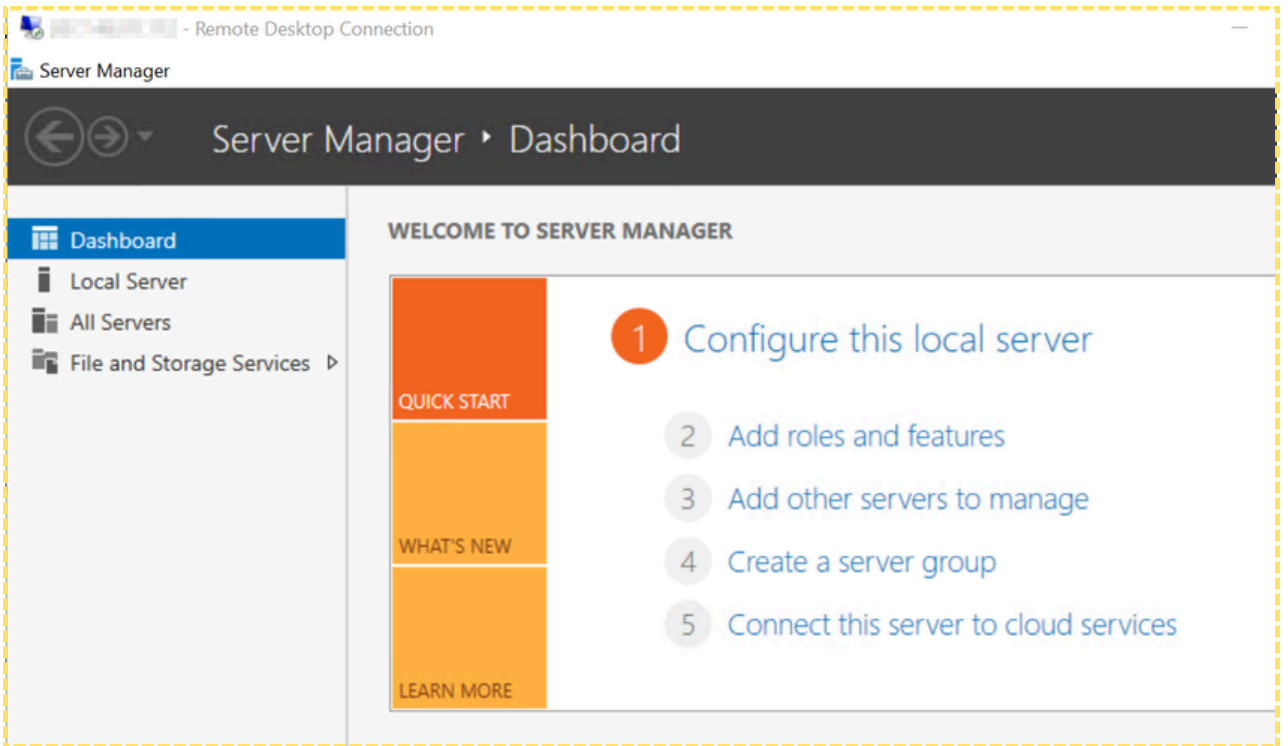
- Click **Show Extra Keys** and click "**Send Ctrl+Alt+Delete**"

- Copy password from the Vultr server details
- Paste password into the console and login to the server



2. RDP Access

- Copy server IP address from the Vultr dashboard
- Open Remote Desktop Connection on local machine
- Paste IP address and click Connect
- Login using the same credentials as the console
- If connect it will display the Windows Server desktop, indicating correctly and ready for use



Day 6 - Elastic Agent and Fleet Server Introduction

This challenge is designed to provide SOC analysts with **Learn what a Fleet Server is and the Elastic Agent**. This challenge will cover introduce to Elastic Agent and Fleet Server, important components for managing data collection in Elastic Stack environment. This is especially useful for those working in Security Operations Centers or anyone who needs to manage multiple endpoints.

1. Managing Agents at Scale

Imagine having to manually configure and installed agents on 100 Windows machines. Only to realize that they were not configured to forward PowerShell logs. Several options are available is :

- Manual Configuration. Logging into each machine and configuring the agent individually. This is time-consuming and impractical for a large number of machines.
- Group Policy. Using a group policy to update all endpoints. This is better than manual configuration, but can still be complex to manage.
- Fleet Server. This is the best solution. Using a Fleet Server to manage all agents from a central location.

2. Elastic Agent Introduction

Definition

- The Elastic agent is a unified agent that allows to collect various types of data, including logs, metrics, and more, from your endpoints.
- It acts as a single point of data collection, simplifying the process of gathering information from different sources

Key Features

- Unified Data Collection: Collects various types of data (logs, metrics, etc.) using a single agent.
- Policy-Based Management: Agents are managed through policies that define what data to collect and where to send it.
- Integration Support: Easily add new integrations and protections through policy updates.
- Data Forwarding: Sends collected data to either Elasticsearch or Logstash.

Deployed in two modes

- Standalone mode — All policies are applied to Elastic agent manually as a YAML file.
- Managed by Fleet — The Elastic Agent policies and lifecycle are centrally managed by the Fleet app in Kibana. The Integrations app also lets you centrally add integrations with other popular services and systems. This is the recommended option for most users.

Comparison with Beats

- Beats
  - Different types for specific data collection needs (File beat, Metric beat, Packet beat, etc.)
  - Each Beat needs to be installed and configured separately
  - May be necessary for specific, specialized data collection needs
- Elastic Agent
  - One agent collects various types of logs and metrics
  - Simplified management and configuration
  - Managed through policies via Fleet Server
  - Generally suitable for most use cases due to its unified approach and centralized management

3. Fleet Server

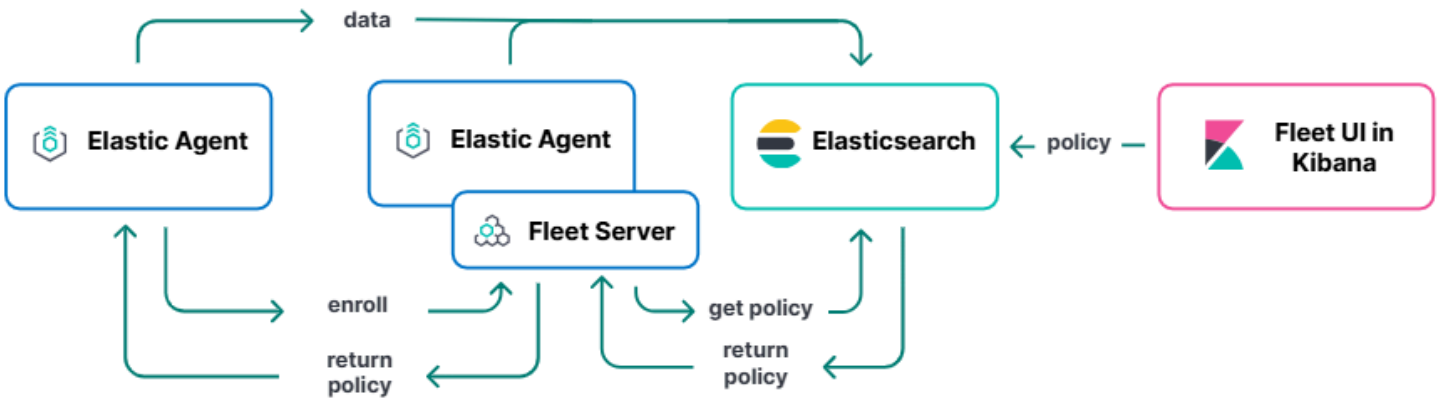
Definition

- Fleet Server is a component of the Elastic Stack used to centrally manage Elastic Agents.
- Fleet Server is like the hub that links all your Elastic Agents together in a "fleet"
- It makes it easy to manage a bunch of agents from one place, so there's no need to fiddle with each one individually.
- It’s launched as part of an Elastic Agent on a host intended to act as a server.
- Fleet Server deployment steps are similar to any Elastic Agent, except that you enroll the agent in a special Fleet Server polic

Key Benefits

- Centralized Management: Manage all agents from one location.
- Policy Updates: Easily update agent policies to add new integrations or change data forwarding destinations (Elasticsearch or Logstash).
- Agent Management: Update agent versions and unenroll agents easily.
- Avoid Manual Updates: Without a Fleet Server, updating agents can be painful, especially if you have to do it manually..

Elastic Agents communicate with Fleet Server



- Elastic Agents connect to Fleet Server to get and apply policies.
- New policies are created in the Fleet UI and saved in Elasticsearch.
- Elastic Agents enroll in a policy by sending a request to Fleet Server with an enrollment key.
- Fleet Server retrieves the policy from Elasticsearch and sends it to the enrolled agents.
- Agents use the policy to configure how they collect and send data to Elasticsearch.
- Agents stay connected to Fleet Server to check for updates.
- When a policy is updated, Fleet Server fetches the changes from Elasticsearch and sends them to the agents.
- Fleet Server writes updates to Elasticsearch to track agent status and policy rollouts.

4. Summary

- Elastic Agent is a unified way to collect data, replacing the need for multiple Beats in many cases.
- Fleet Server provides a centralized location to manage your Elastic Agents, making updates and policy changes much easie
- Elastic Agents and Fleet Servers are essential tools for managing data collection in the Elastic Stack

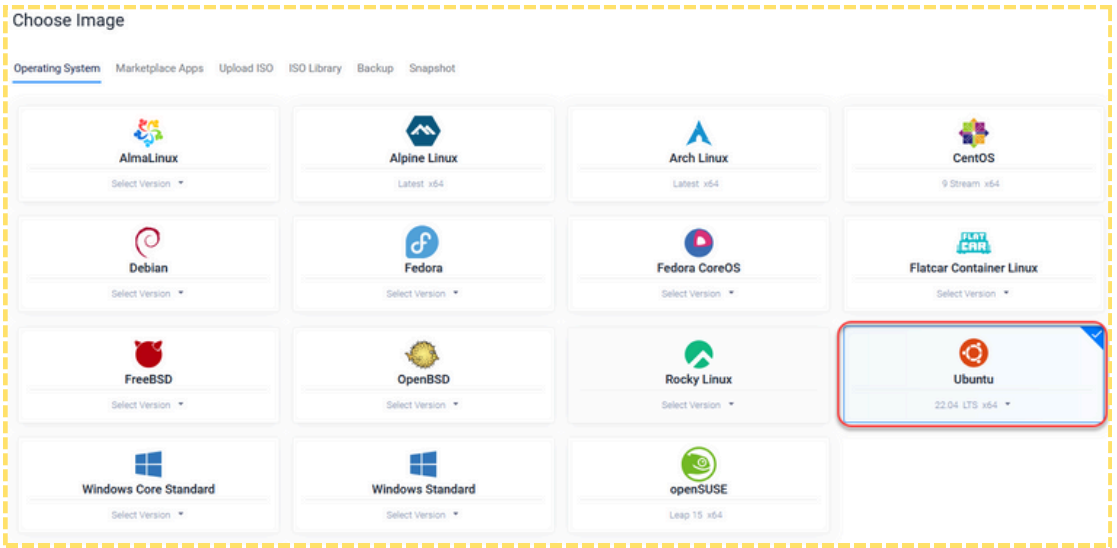
Day 7 - Elastic Agent and Fleet Server Setup

This challenge is designed to provide SOC analysts with **Install Elastic Agent on Windows Server and Enroll the windows Server into a Fleet**. This challenge will cover process of setting up an Elastic Agent on a Windows server and configuring a Fleet Server for centralized management using Elastic Stack. The Elastic Agent is a lightweight data shipper that collects logs and metrics, while the Fleet Server provides a central point for managing these agents. This setup allows for efficient monitoring and data collection from various hosts.

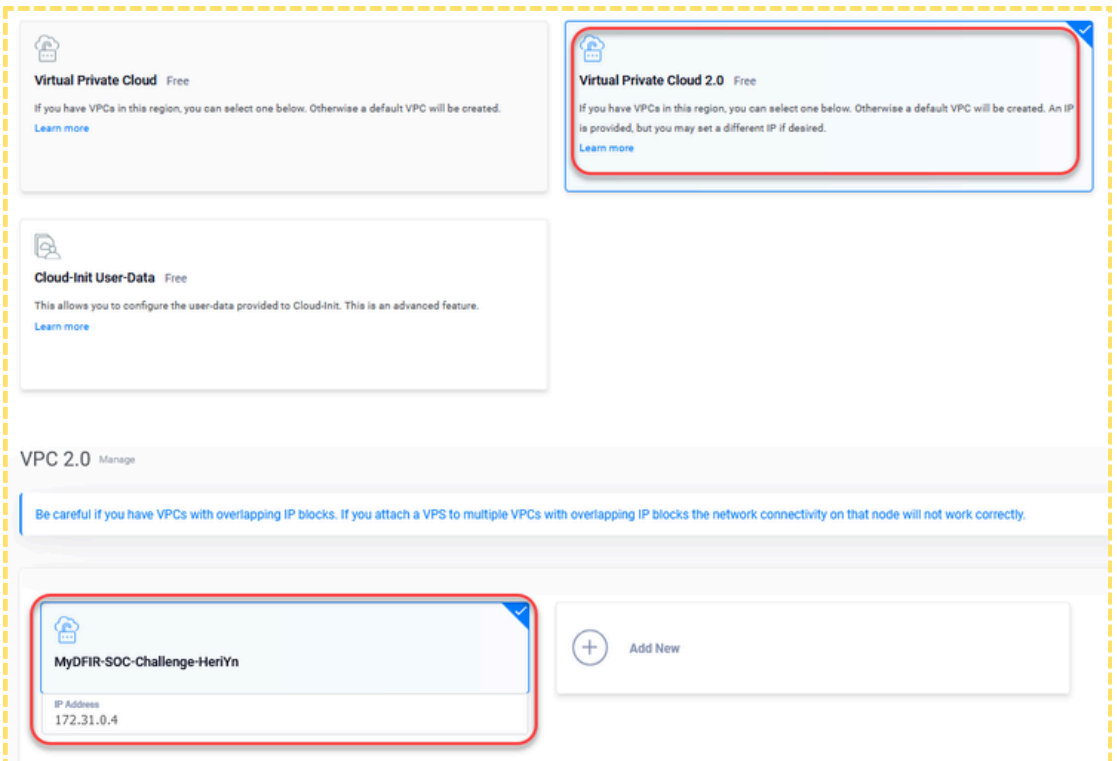
1. Deploy New Fleet Server

To begin, needs to set up a new server to function as the Fleet Server

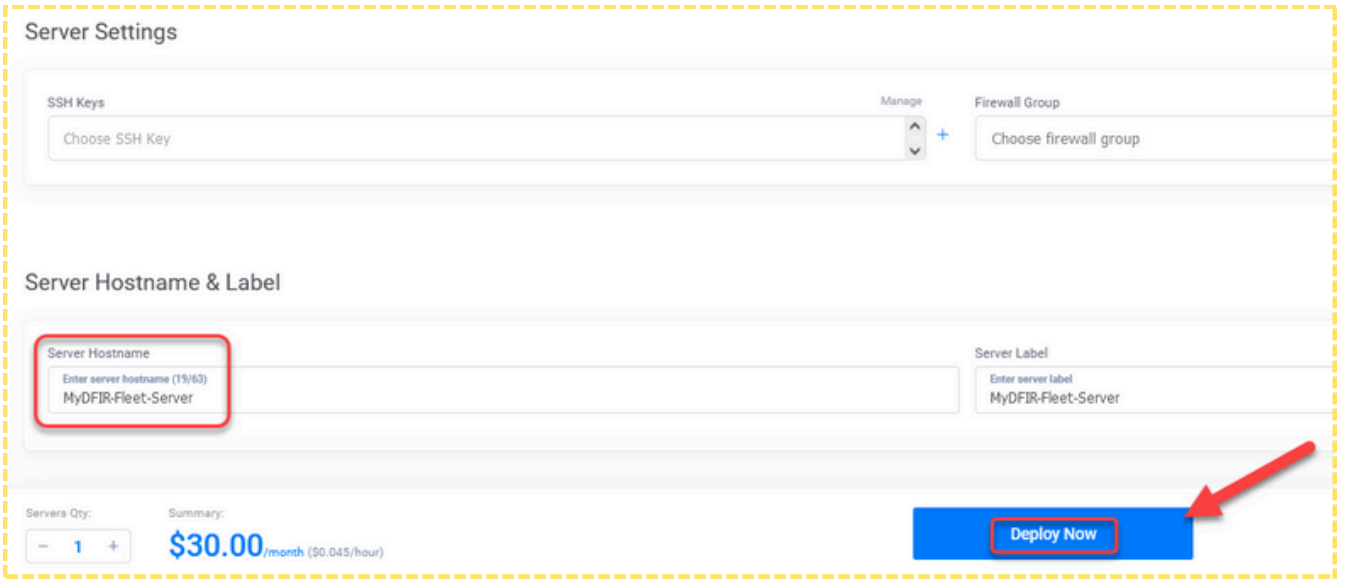
- In Vultr, select deploy a new server.
- Select location and Ubuntu 22.04 as the operating system



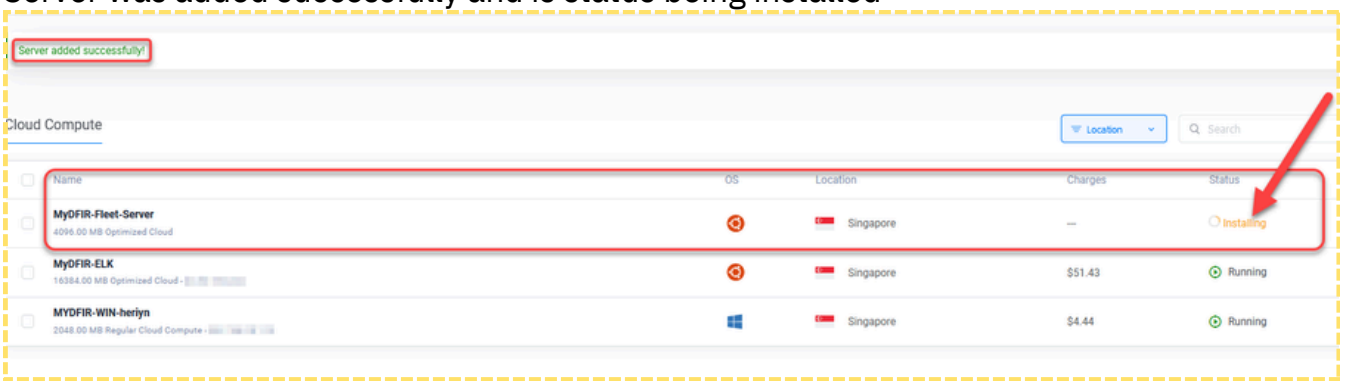
- Choose 1 CPU and 4 GB RAM, disable auto backups and IPv6 and select Virtual Private Cloud 2.0



- Ensure your network is selected. For the firewall group, we can leave it as default for now
- Name it "MyDFIR-Fleet-Server" and click "Deploy."

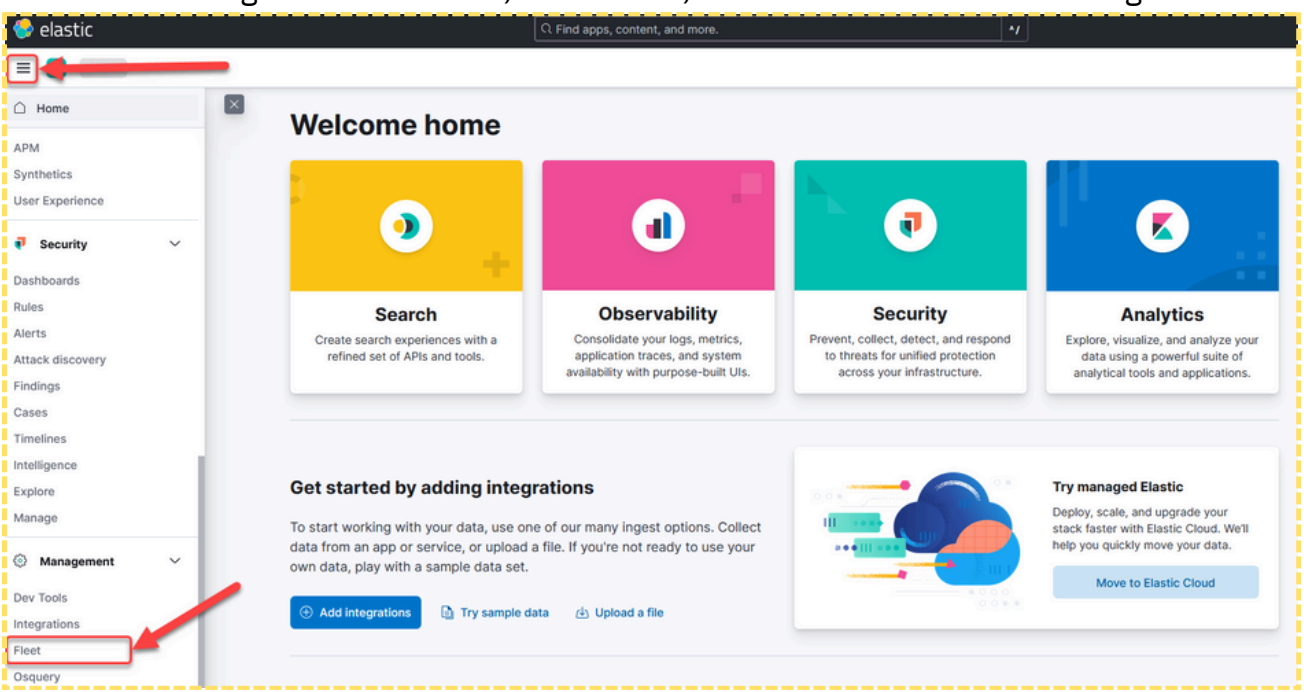


- Server was added successfully and is status being installed

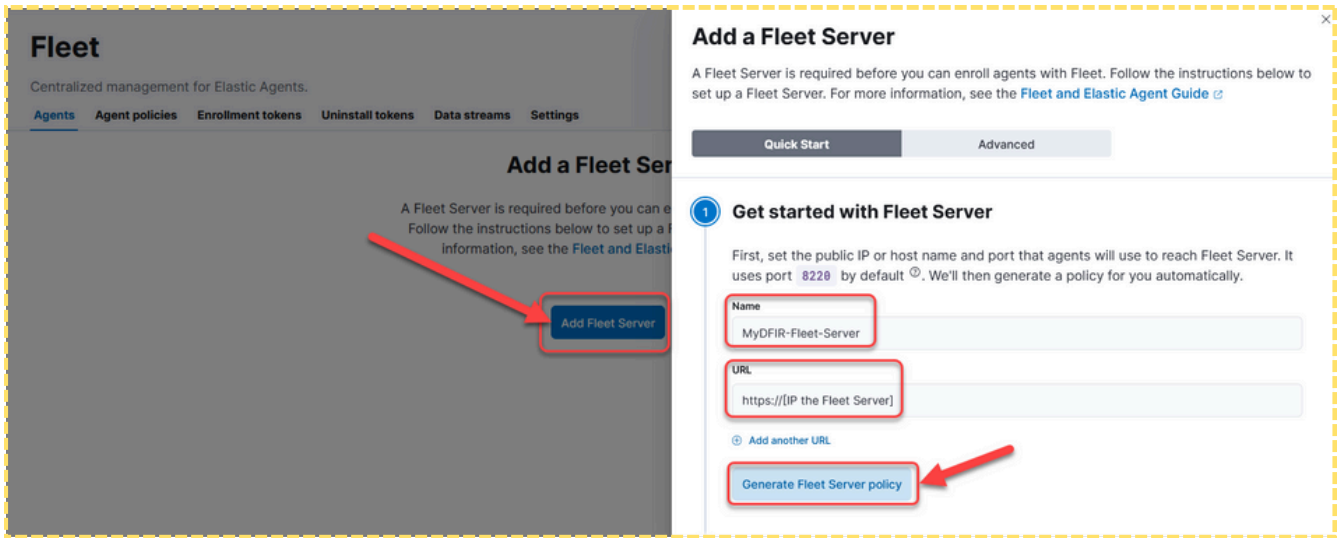


2. Add a Fleet Server from ELK GUI

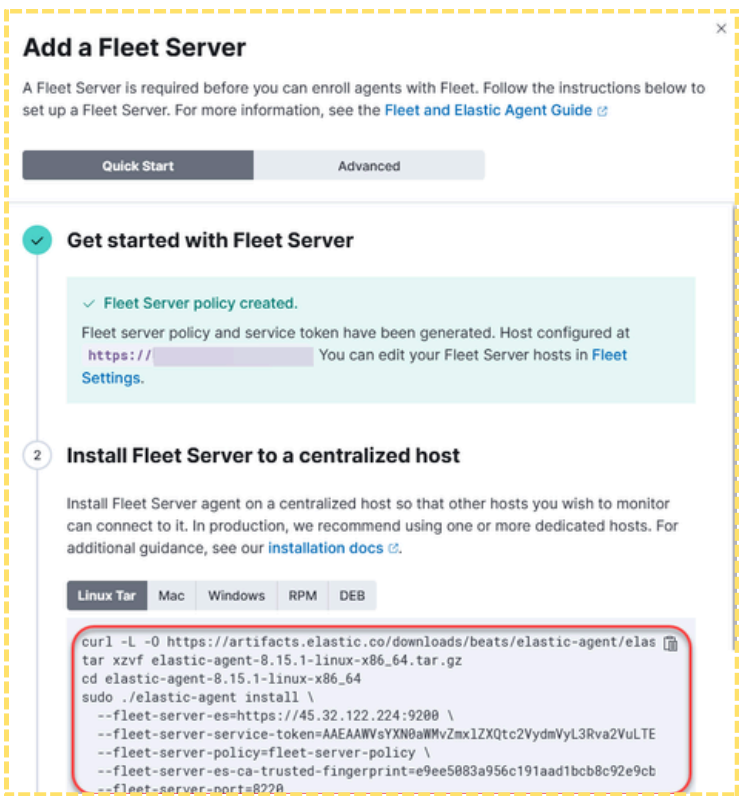
- Go back to access Elastic Web GUI
- Click the hamburger icon on the left, scroll down, and select "Fleet" under Management



- Click the blue "Add Fleet Server" button and choose "Quick Start"
- Name it "MyDFIR-Fleet-Server" for the fleet name
- Enter IP address with https:// and followed by port **8220**
- Click "Generate Fleet Server Policy"



- Once the policy is created, copy the installation command



3. Installing Ubuntu As a Fleet server

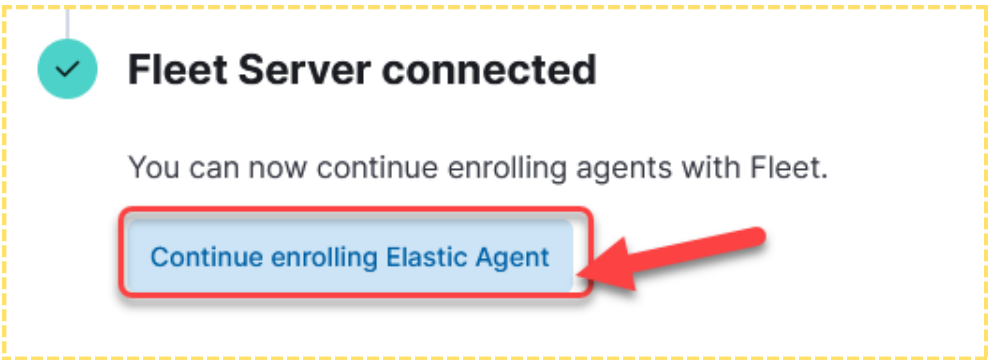
- Connect to ubuntu Server via SSH using the root user
- Update the package repositories with run `apt-get update && apt-get upgrade -y`
- Ensure Fleet Server can communicate with Elasticsearch by allowing port 9200 using ufw

```
root@MyDFIR-ELK:~# ufw allow 9200
Rule added
Rule added (v6)
root@MyDFIR-ELK:~#
```

- Paste the copied installation command into the terminal and execute it

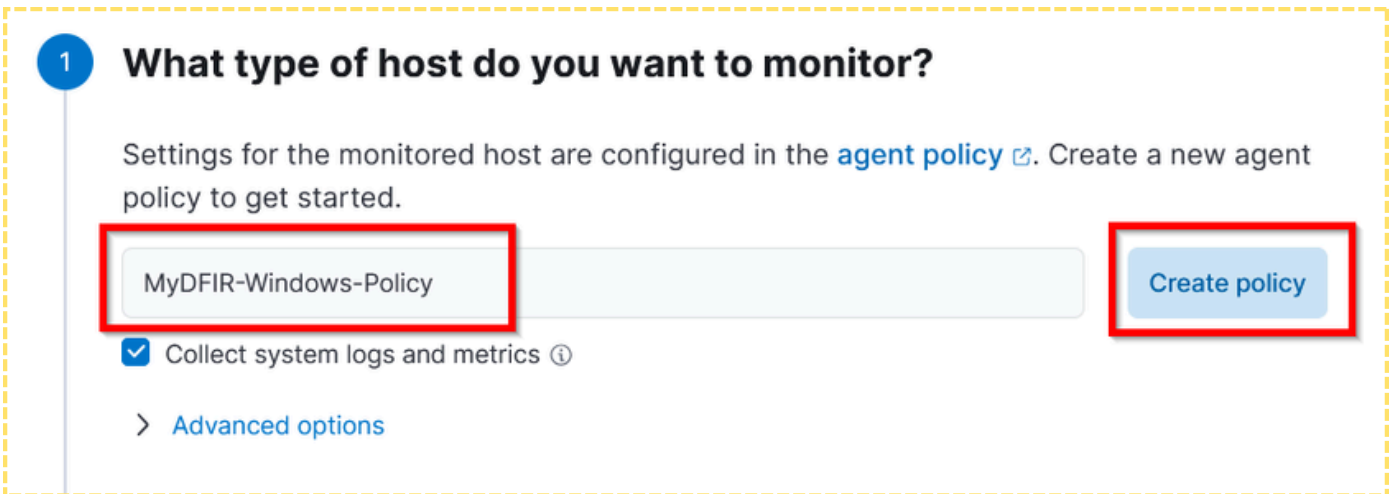
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@MyDFIR-Fleet-Server:~# curl -L -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elast
ic-agent-8.15.1-linux-x86_64.tar.gz
tar xzvf elastic-agent-8.15.1-linux-x86_64.tar.gz
cd elastic-agent-8.15.1-linux-x86_64
sudo ./elastic-agent install \
  --fleet-server-es=https://[redacted]:9200 \
  --fleet-server-service-token=AAEAAWVsYXN0aWMvZmxlZXQtc2VydmVyL3Rva2VuLTE3MjY0OTcwNzQxMDk6NTlJYVU4RTNT
NXFYcFBMd3poQkVVQQ \
  --fleet-server-policy=fleet-server-policy \
  --fleet-server-es-ca-trusted-fingerprint=e9ee5083a956c191aad1bcb8c92e9cb972da302dd3ea62e62fb1907a7a4c
66ba \
  --fleet-server-port=8220
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  294M  100  294M    0     0  21.6M      0  0:00:13  0:00:13 --:--:-- 31.6M
elastic-agent-8.15.1-linux-x86_64/manifest.yaml
elastic-agent-8.15.1-linux-x86_64/.elastic-agent.active.commit
```

- This terminal session demonstrates process of downloading, extracting, and installing Elastic agent as a Fleet server
- Go back to the Elastic web UI and verify that the Fleet Server is connected

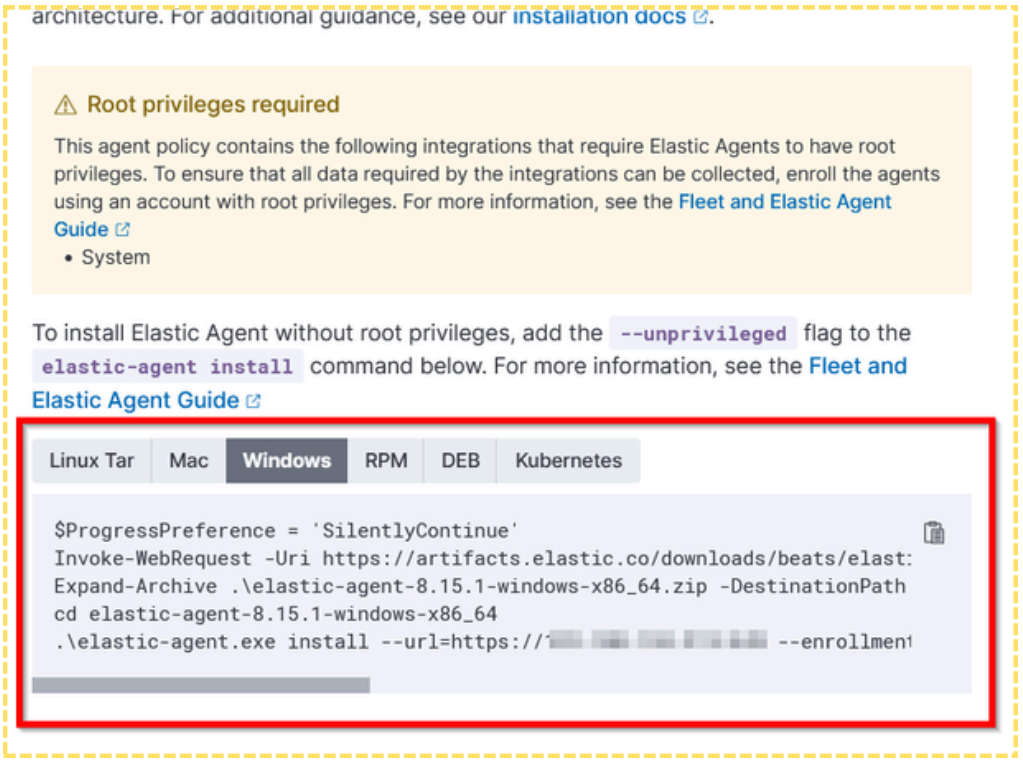


4. Install Elastic Agent on Windows

- In the Elastic web UI, go to "Agents" and create a new policy for Windows.
- Name it policiy "MyDFIR-Windows-Policy"

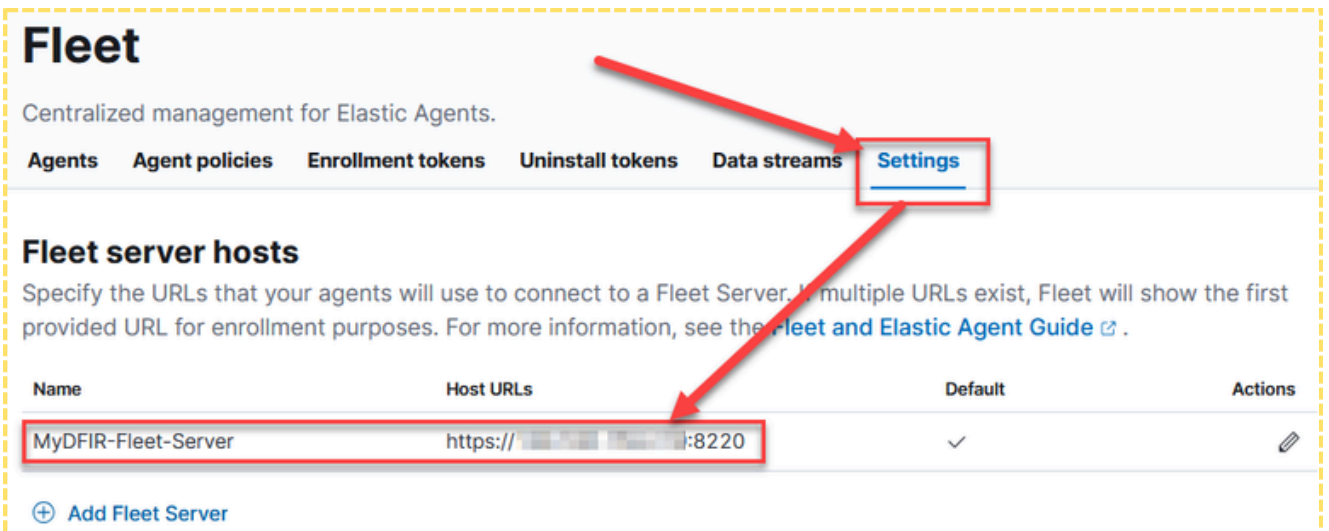


- Copy the powershell command provided for installing the agent



- On the Fleet Server, use ufw to allow connections port 8220

- Make sure the host URL settings on the Fleet Server using the correct IP:port

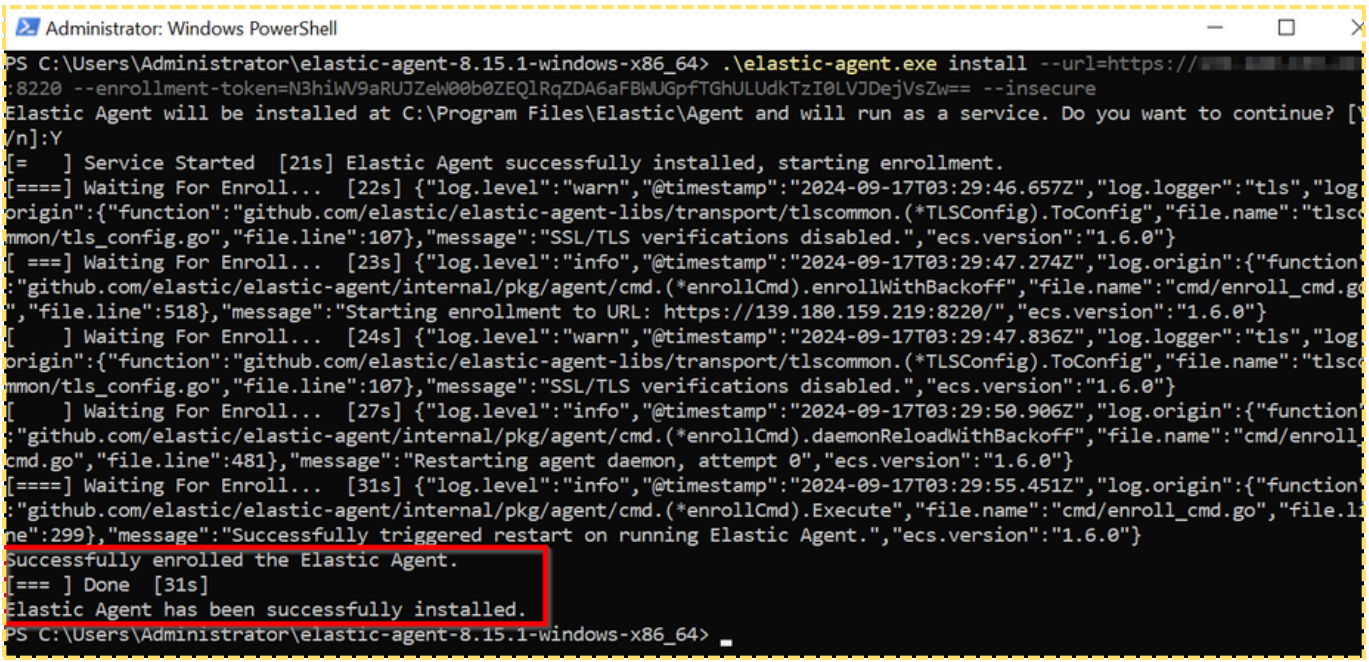


- Login back to the Windows Server as an administrator
- Open terminal as an administrator on Windows server

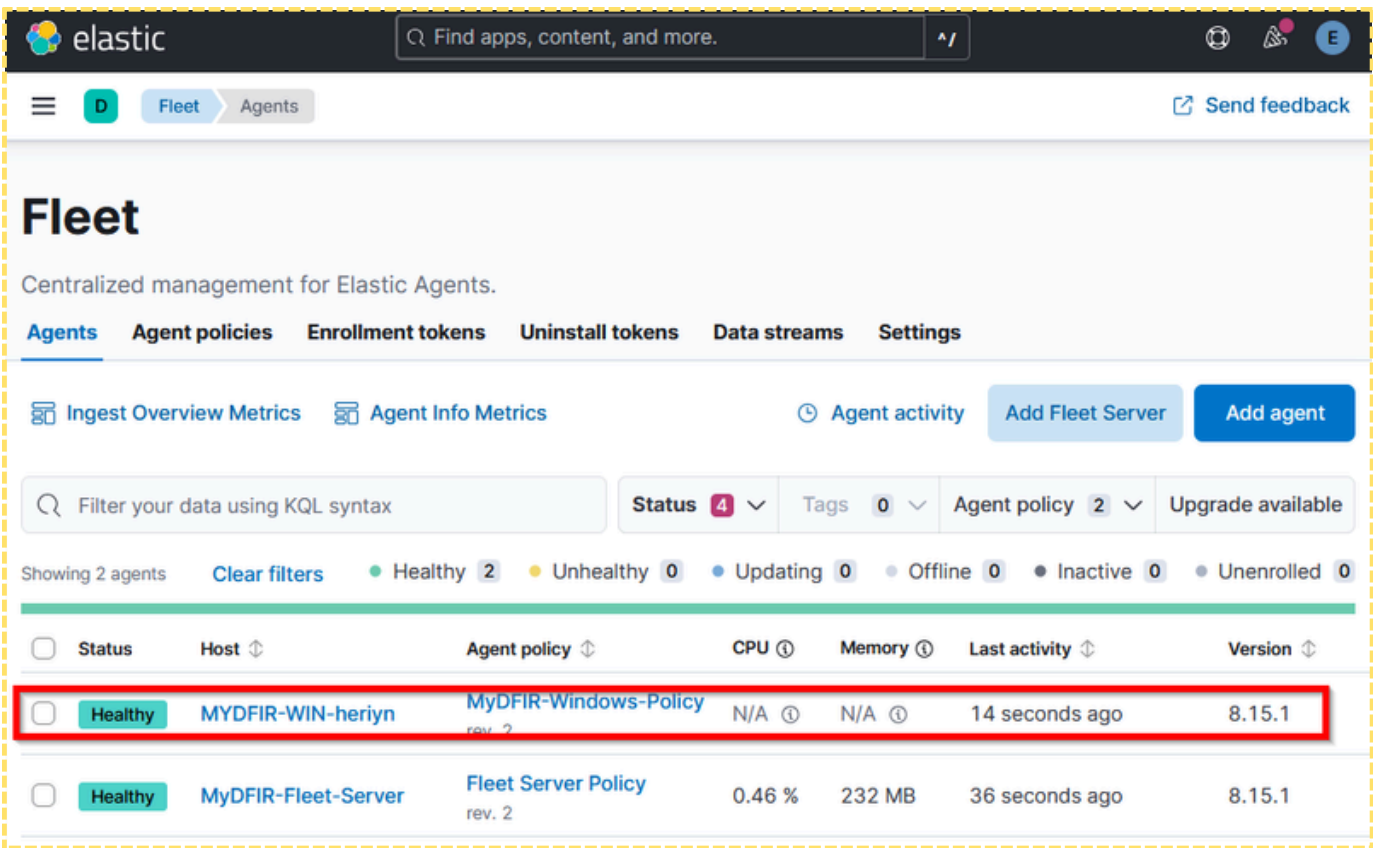
- Pastes the installation command into terminal and execute it install and enroll
- Add --insecure to bypass the self-signed certificate check (SSL/TLS verification disabled)

```
.\elastic-agent.exe install --url=https://<Fleet Server IP>:8220
--enrollment-token=<token> --insecure
```

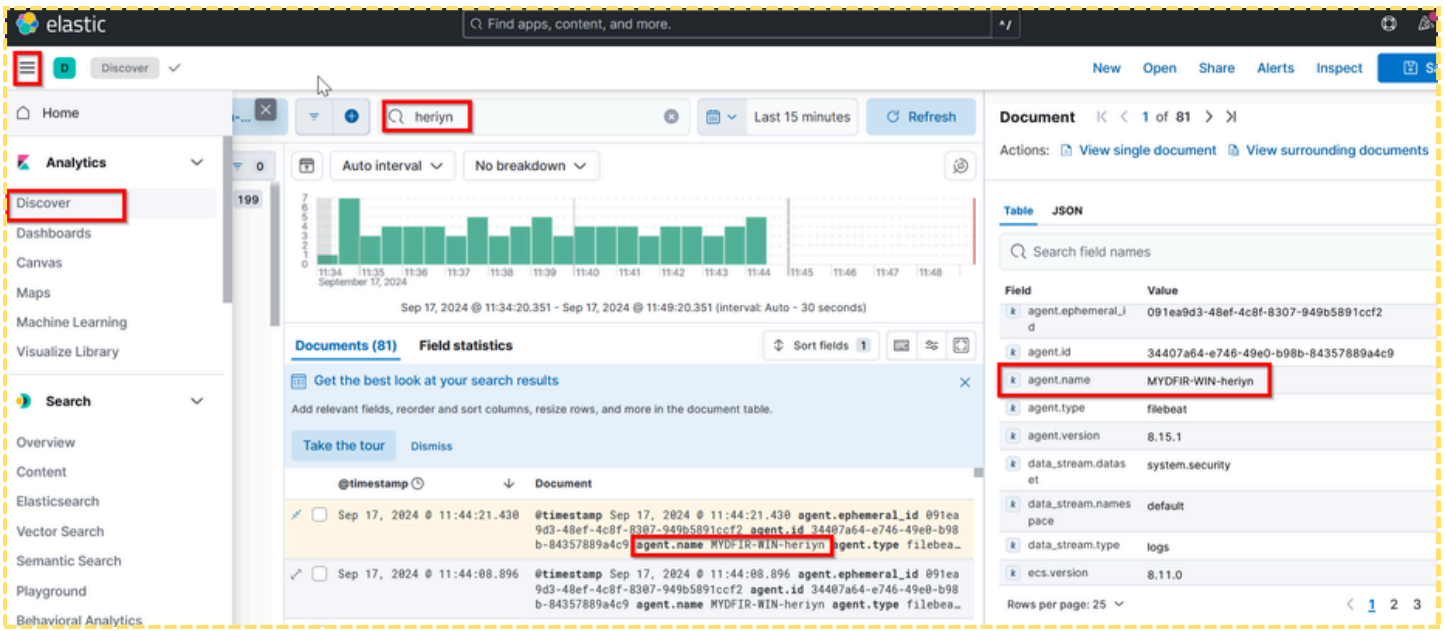
- Log the Elastic agent was successfully installed and enrolled with the Fleet Server



- Go back to the Elastic web UI and checks the **Agents** section in the Fleet Mgmt interface
- The agent is now visible in Fleet centralize management, showing a healthy status and assigned to “MyDFIR-Windows-Policy”



- Verify that Elastic agent installed on the Windows server is successfully sending logs to Elasticsearch
- Clicks hamburger menu, select “**Discover**” option under the "**Analytic**s" section
- Use the search bar at the top of the Discover page to filter logs by agent name
- Search term **heriyn** is used to filter data related to agent **MYDFIR-WIN-heriyn**



5. Troubleshooting Tips

- Check network connectivity between all components using ping and telnet
- Verify firewall ufw rules on ELK server allow connections from Fleet Server on port 9200
- By default Fleet Server uses port 8220. Allow traffic on port 8220 which the Fleet Server uses to communicate with agents
- Ensure Fleet Server URL is correctly configured in the Fleet settings ELK GUI using port 8220
- To bypass certificate error issue, add the --insecure flag to the enrollment command when running it on the agent
- Errors like "connection refused" or "failed to execute request to Fleet server" indicate network or port issues
- If the agent fails to enroll, try restarting the agent service on the host machine and try re enrolling the agent using installation enrollment command
- Review logs for any error messages on all components (Elasticsearch, Fleet Server, Elastic Agent)
- Restart services on the respective servers to ensure changes are applied

Day 8 - What is Sysmon

This challenge is designed to provide SOC analysts to **learn about sysmon and what it can do**. Sysmon (System Monitor) is a Microsoft Sysinternals tool that provides detailed monitoring of a Windows computer, and this tool is essential for security monitoring. This guide will explains what Sysmon is, its capabilities, and how it can be used to enhance endpoint visibility for security investigations. By implementing Sysmon and following best practices, SOC analysts can improve their ability to detect and respond threats on Windows endpoints.

1. Endpoint Visibility

- Endpoints are often the first targets in cyberattacks, as they serve as entry points for attackers to infiltrate an organization's network
- Endpoint visibility refers to the ability to monitor activities occurring on endpoints
- Endpoint visibility is critical for detecting and investigating potential compromises
  - For example, if an attacker uses a phishing email to deliver malware to an endpoint, visibility into the endpoint's processes, file changes, and network connections can help detect the compromise early
- When a potential compromise is detected, having detailed logs and data from endpoints is essential for understanding what happened

2. Limitations Default Windows Logging

- Default Windows logging provides only basic information about system activities
  - Default Windows logging lacks granular details about system activities
  - Log network connection not specify which process initiated it, such as execution of suspicious script
- Default Windows logging does not track many important events that are critical for security investigations, such as:
  - Process creation and termination
  - File creation, modification, or deletion
  - Registry changes
  - Network connections initiated by specific processes

3. Introduction to Sysmon

- **What is Sysmon?** Sysmon is a free Microsoft tool, part of the Sysinternals suite, that provides detailed logging of endpoint activity. It is designed to enhance visibility and aid in security investigations
- **Why is it important?** Default Windows logging is often insufficient for in-depth security analysis. Sysmon captures important events, such as process creations, network connections, and file modifications, that are not logged by default
- **Key Features**:
  - Process Creation logging with command-line details
  - Process Hashing for threat intelligence lookups
  - Process GUID for event correlation
  - Network connection logging (source/destination IPs, ports, process)
  - Customizable logging via configuration files

4. Sysmon Event IDs

- Process Creation (Event ID 1):
  - Tracks new processes along with their command line.
  - Includes file hashes for threat intelligence lookups.
  - Identifying malicious process execution.
- Network Connections (Event ID 3):
  - Tracks network connections initiated by processes.
  - Logs source and destination IPs and ports.
  - Disabled by default, must be enabled via configuration file.
  - Essential for detecting command and control (C2) communications.
- Driver/Image Load and Create Remote Thread (Event IDs 6, 7, 8):
  - Identifies potential defense evasion techniques like process injection.
  - Can be noisy with false positives, require careful analysis.
  - Event ID 7 (image load) is disabled by default.
- Process Access (Event ID 10):
  - Tracks when a process accesses another process.
  - Useful for detecting credential access attempts, especially against the lsass.exe process.
- DNS Query (Event ID 22):
  - Logs DNS queries made by the endpoint
  - Identifying communication with malicious domains, including generated by Domain Generation Algorithms (DGAs)

5. Sysmon Use Cases

- Detecting Malicious Process Execution
  - Scenario: An attacker executes a malicious executable on an endpoint.
  - Sysmon Log: Event ID 1 will be generated, logging the malicious process creation with its command line and file hash
- Detecting C2 Communication
  - Scenario: A compromised host establishes an outbound connection to a C2 server
  - Sysmon Log: Event ID 3 (if enabled) will log the network connection, including source/destination IPs, ports, and the process involved
- Detecting Credential Access
  - Scenario: An attacker attempts to access the lsass.exe process to steal credentials
  - Sysmon Log: Event ID 10 will log the process access attempt to the lsass.exe process
- Detecting Domain Communication
  - Scenario: A compromised host attempts to communicate with a domain generated by a DGA
  - Sysmon Log: Event ID 22 will log the DNS query for the DGA domain

6. Related Resources

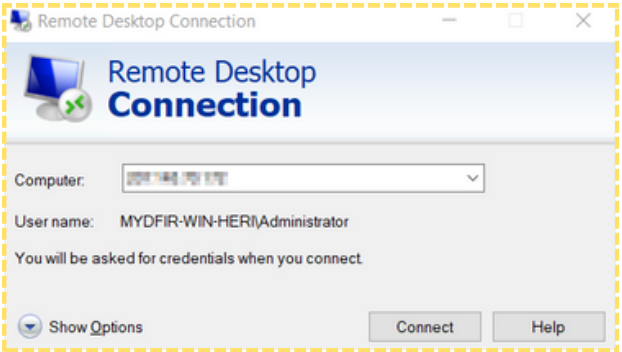
- Sysmon Documentation: learn.microsoft.com/sysinternals/downloads/sysmon
- 30-Day SOC Analyst Challenge: youtube.com/@MyDFIR
- Community Forums and GitHub Repositories, here some references :
  - github.com/microsoft/MSTIC-Sysmon
  - github.com/jymcheong/SysmonResources
  - mhaggis.github.io/sysmon-dfir/

Day 9 - Sysmon Setup Tutorial

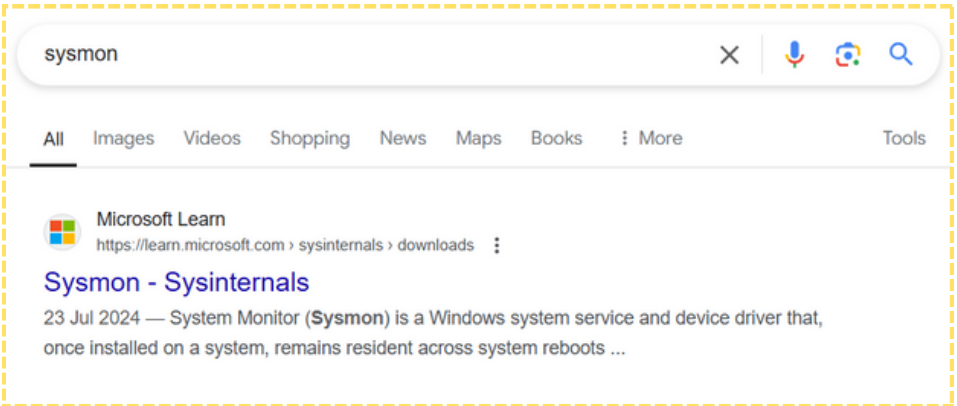
This challenge is designed to provide SOC analysts to **install Sysmon onto Windows server confirm telemetry**. This guide will demonstrates how to install and configure Sysmon to help SOC analysts gain practical experience in setting up Sysmon, understanding its logs, and preparing for real-world incident response scenarios. By implementing Sysmon and following best practices, SOC analysts can improve their ability to detect and respond threats on Windows endpoints.

1. Downloading Sysmon

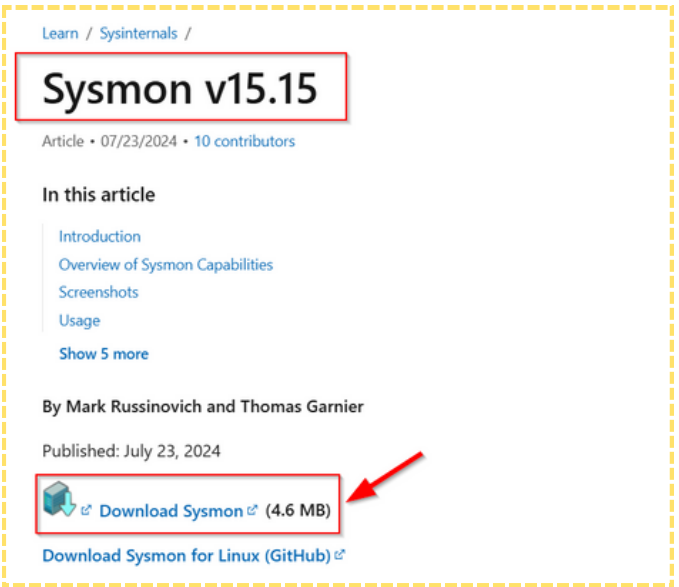
- To begin, retrieves the public IP address Windows server from VULTR and connect using RDP
- Login using username and the server password provided during setup



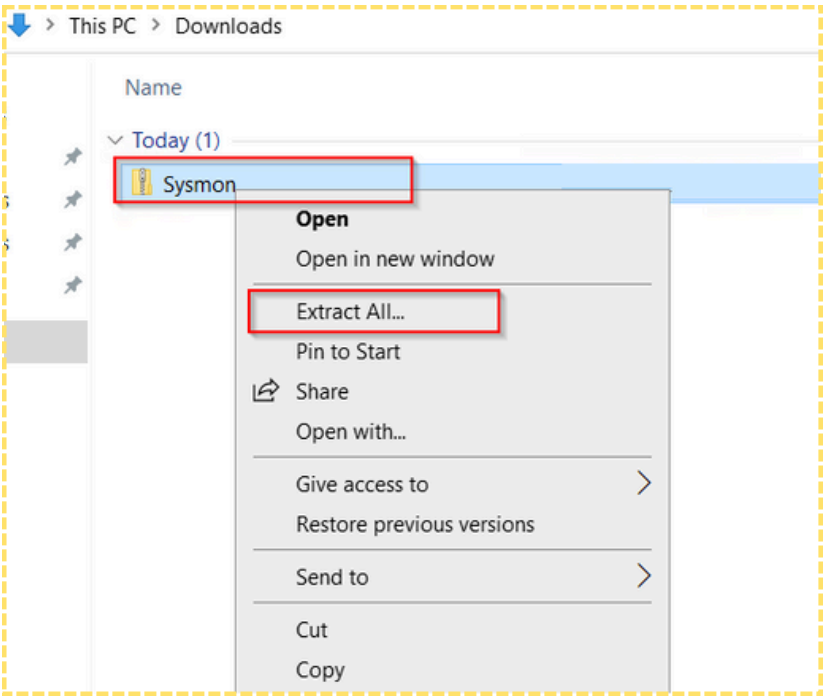
- Open a web browser on Windows server and search for "sysmon"



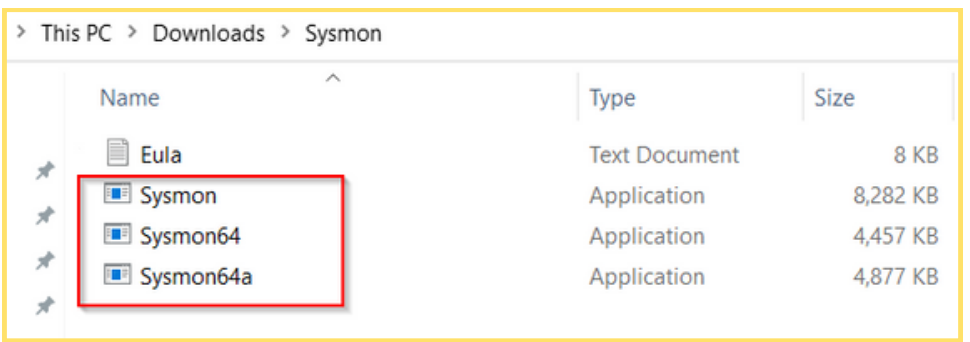
- Selects the official **Microsoft Learn Sysinternals page** and downloads the latest version



- Once downloaded, go to the downloads folder and extract the contents of the Sysmon ZIP file

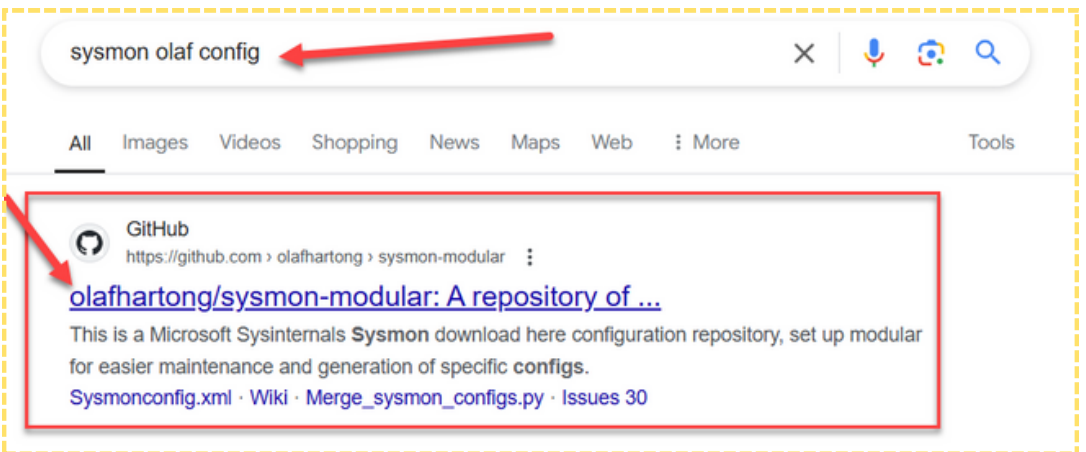


- Verify that three binaries sysmon are extracted successfully

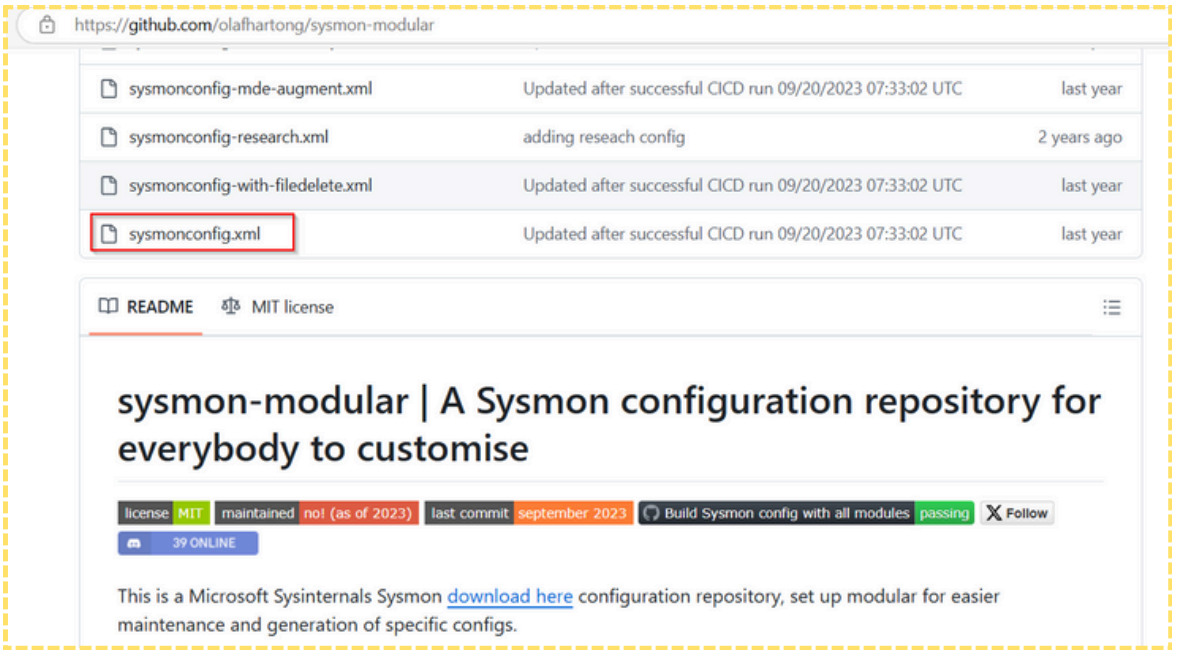


2. Sysmon Configuration File

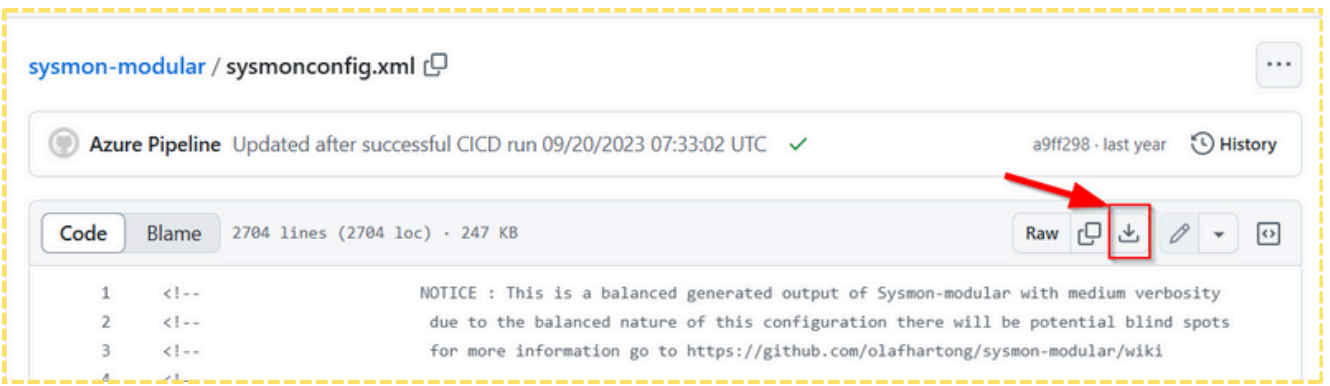
- Sysmon configuration file which determines what events Sysmon will log
- Recommends using a popular configuration file created by **Olaf Hartong**
- Open a web browser and search for "sysmon olaf configuration"
- Click on the GitHub link for Olafhartong Sysmon configuration



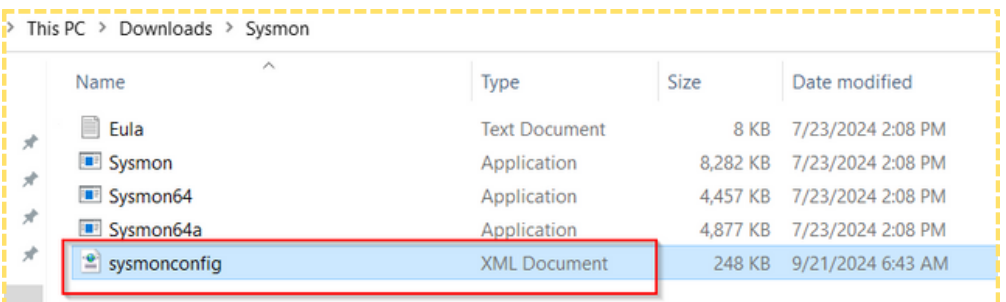
- On the GitHub page, scroll down and select “sysmonconfig.xml”



- Click "Download raw file"

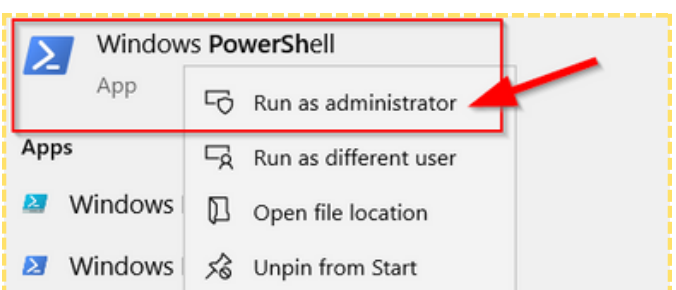


- Save the file as sysmonconfig.xml in the same directory as Sysmon binaries extracted

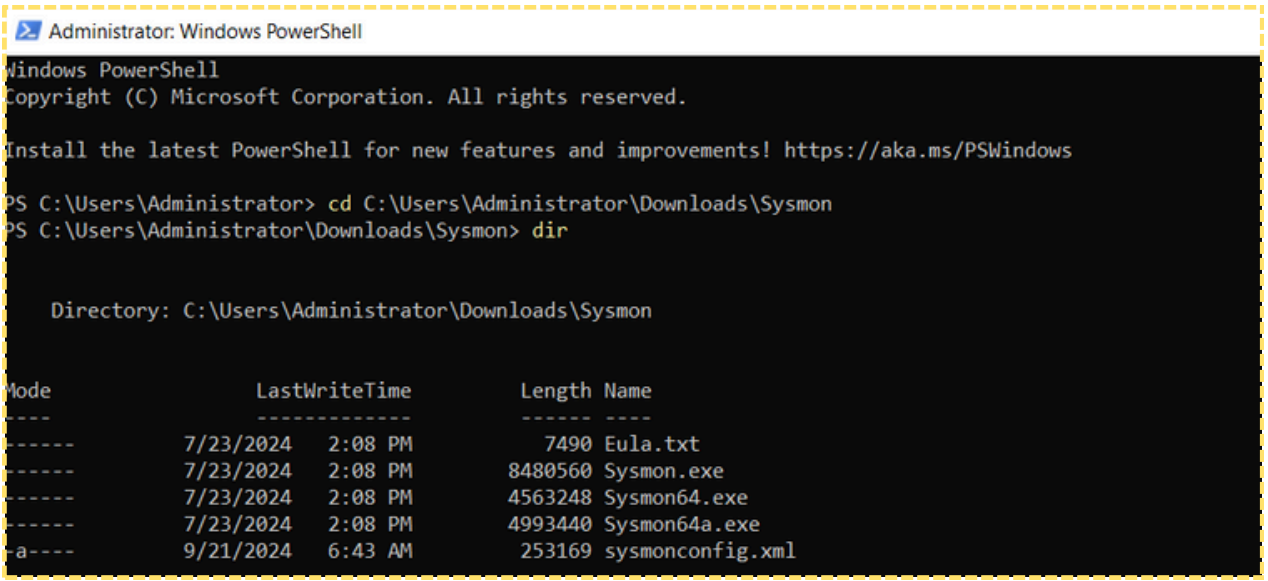


3. Installing Sysmon

- Open PowerShell as an administrator



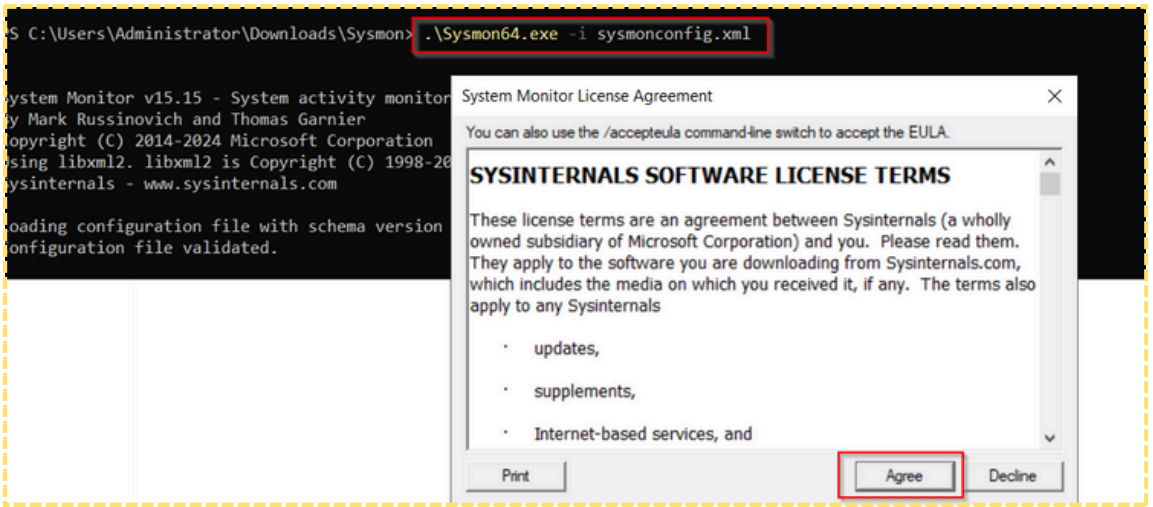
- Navigates to the Sysmon directory using the cd command
- Type dir to verify that you are in the correct directory



- Install Sysmon using the command

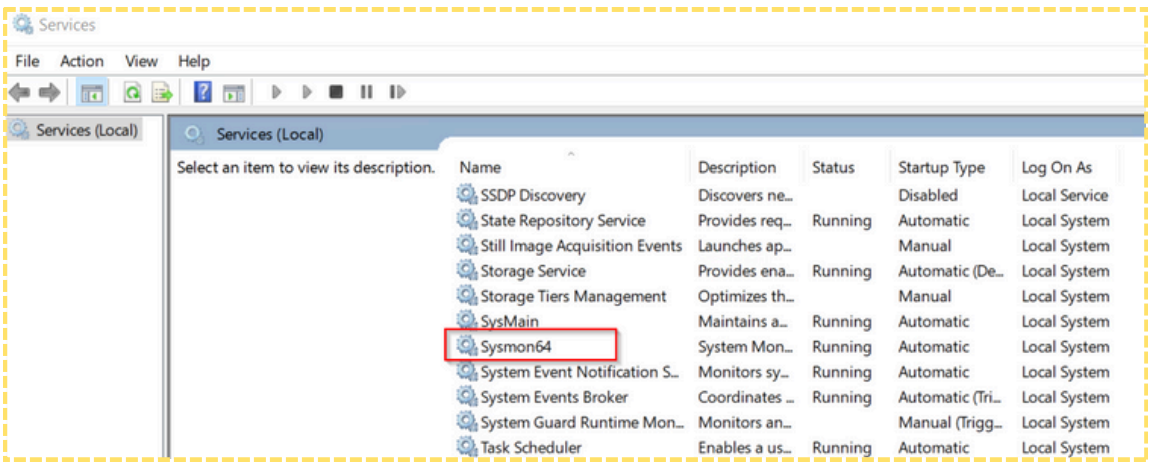
```
.\sysmon64.exe -i sysmonconfig.xml
```

- Click "Agree" to the license agreement

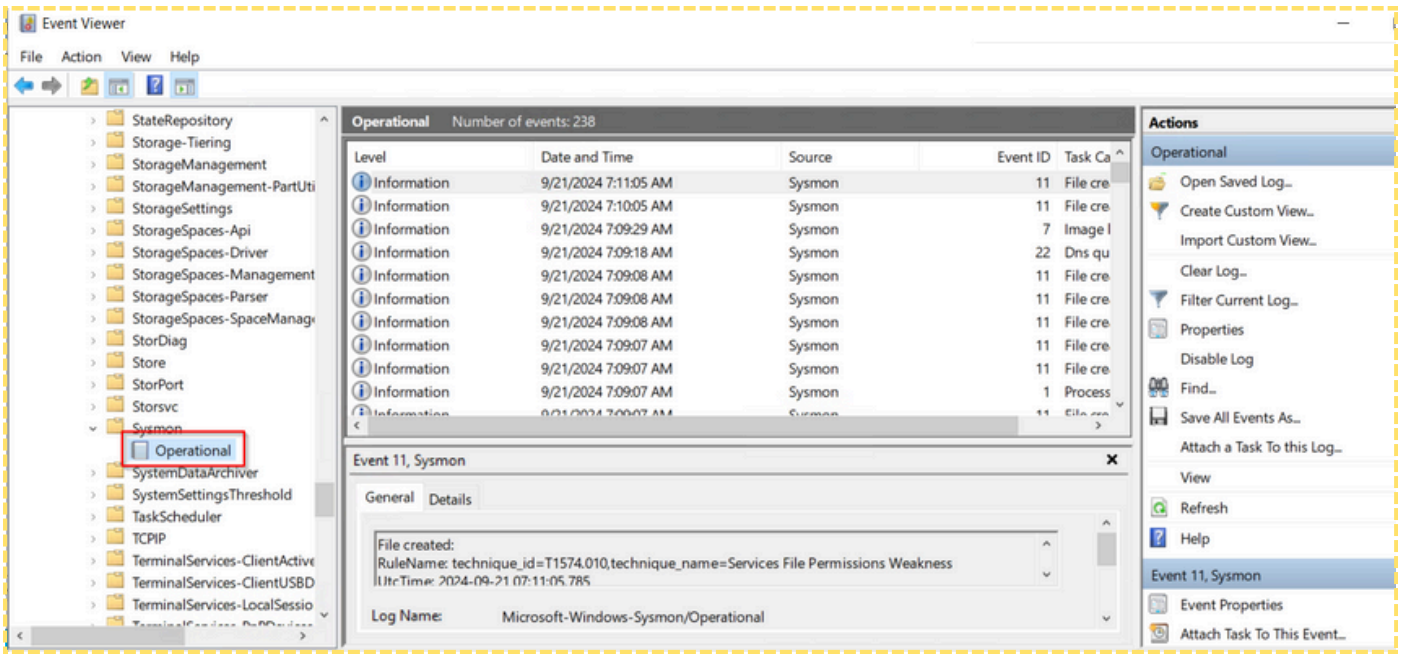


4. Verify Sysmon Installation

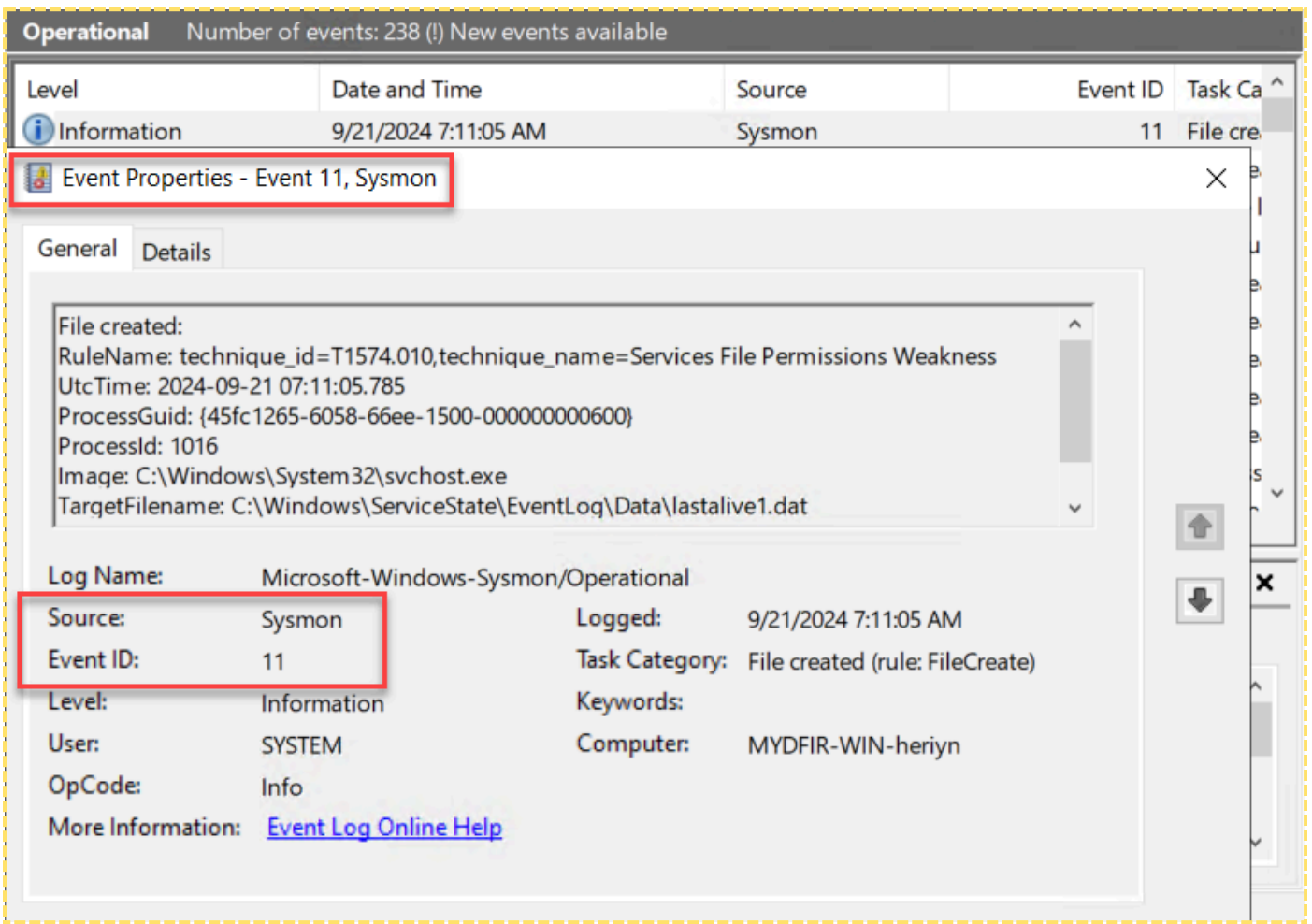
- Open Windows Services application and confirm that "Sysmon" is listed and running.



- Open Event Viewer and navigates to "Applications and Services Logs > Microsoft > Windows > Sysmon > Operational”



- Confirms that Sysmon is generating logs such as event id 11 associated with "FileCreate" events



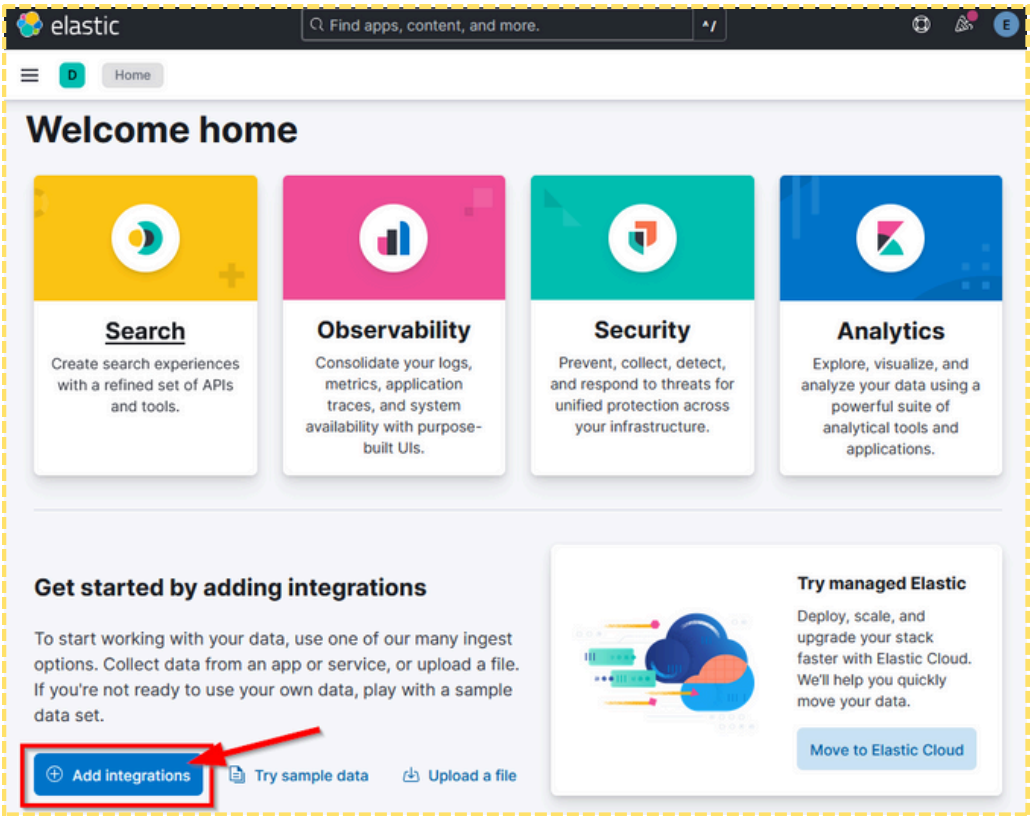
- Sysmon has been successfully installed and is generating logs on the Windows Server

Day 10 - Elasticsearch Ingest Data Tutorial

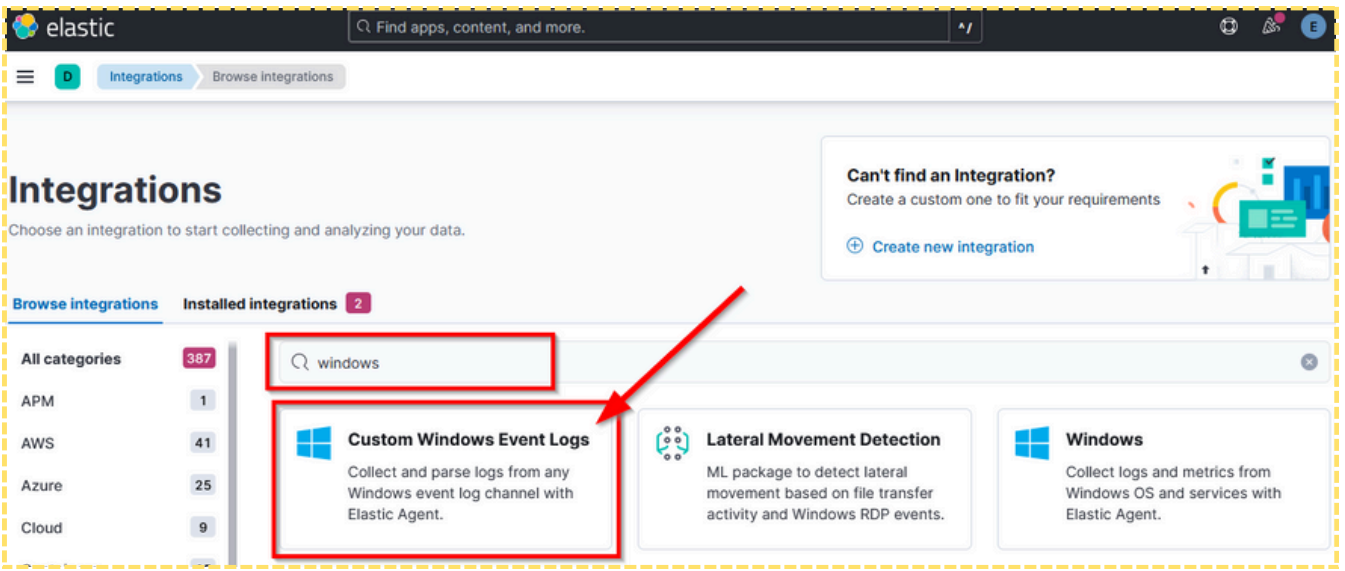
This challenge is designed to provide SOC analysts to **Learn how to ingest Sysmon and Windows Defender logs into Elasticsearch**. This guide will demonstrates how to ingest Sysmon and Microsoft Defender logs from a Windows Server into an Elasticsearch instance. It covers configuring custom Windows Event Log integrations, filtering relevant event IDs, troubleshooting connectivity issues, and verifying log ingestion.

1. Sysmon Log Ingestion

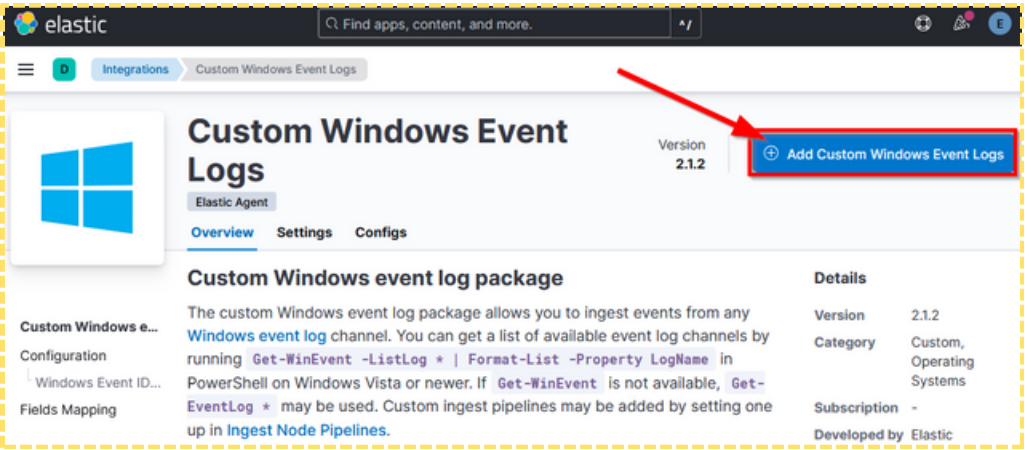
- Login to ELK and click on the "Add Integrations" button on the homepage



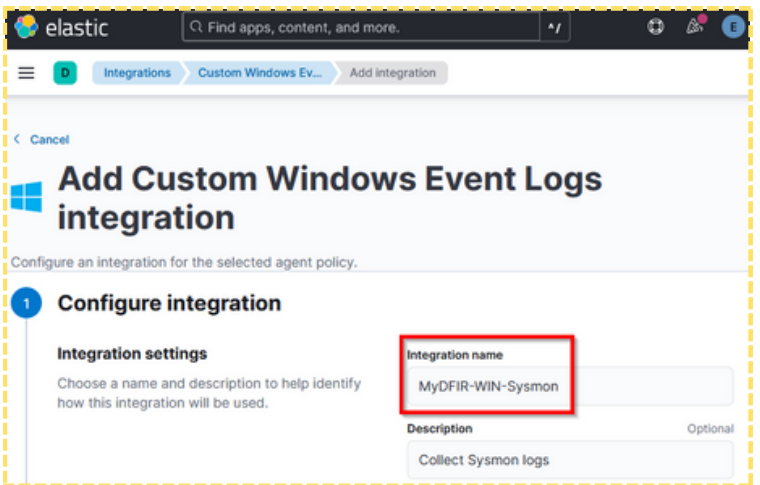
- Search "Windows Event" in the integrations search bar
- Select “Custom Windows Event Log”



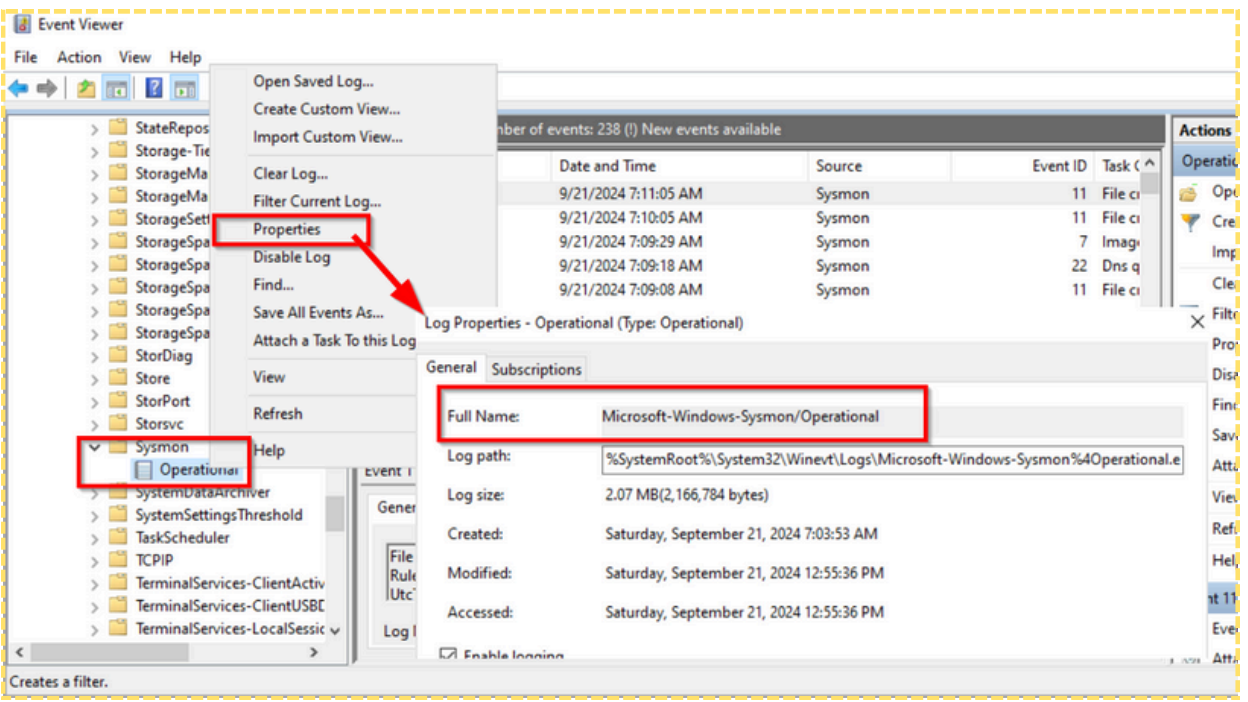
- Click "Add Custom Windows Event Logs" to begin the configuration



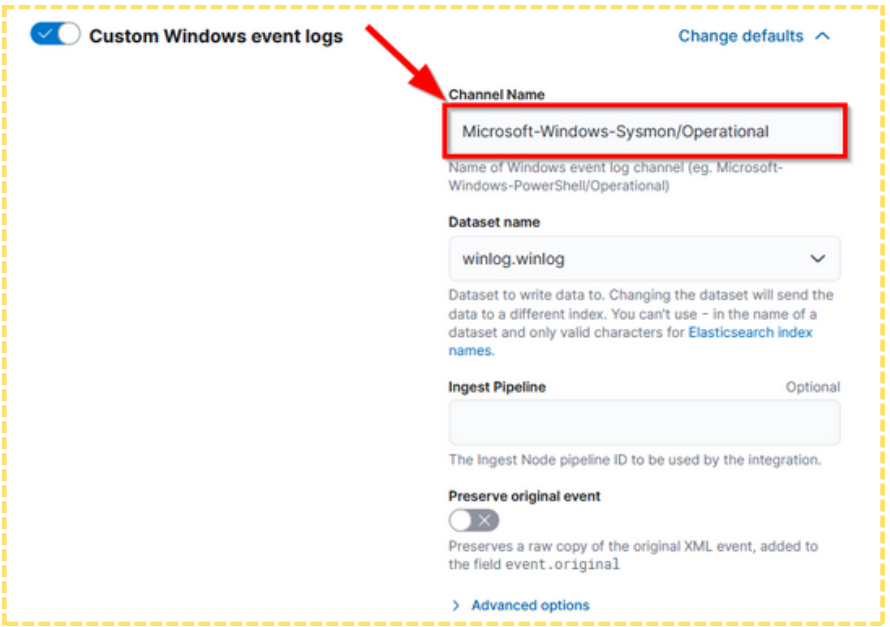
- Type "MyDFIR-Win-Sysmon" and provide a description, such as "Collect Sysmon logs"



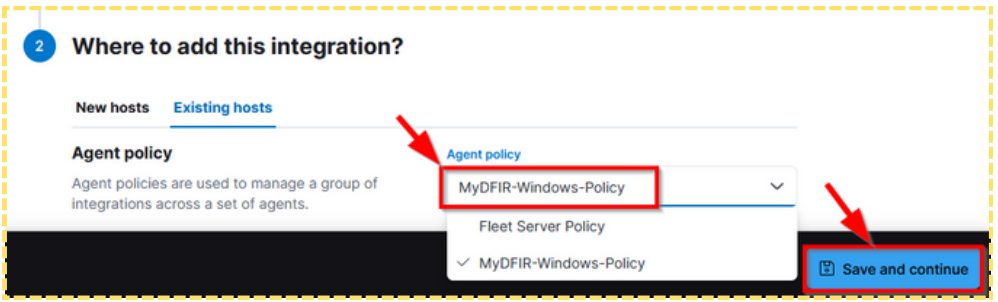
- To find the correct channel name for Sysmon logs, go to Windows Server
- Open Event Viewer and navigate to "Applications and Services Logs" > "Microsoft" > "Windows" > "Sysmon" > "Operational"
- Right click "Operational," select "Properties," and copy full name



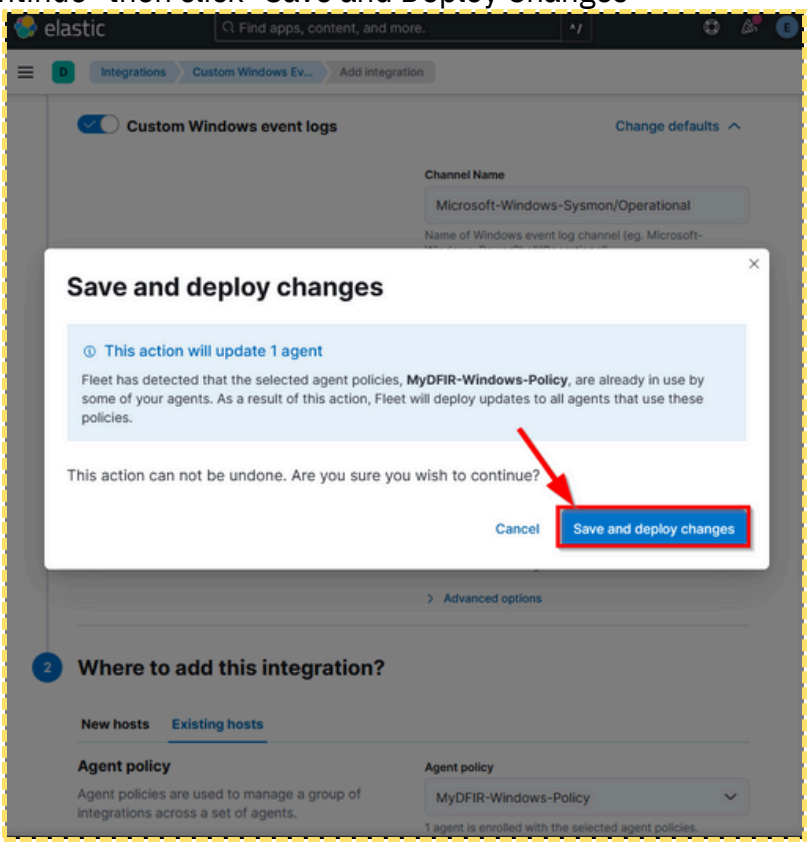
- Paste the full name into the Channel Name on Elasticsearch costom integration



- Select the existing host and agent policy ("MyDFIR-Windows-Policy") for the integration

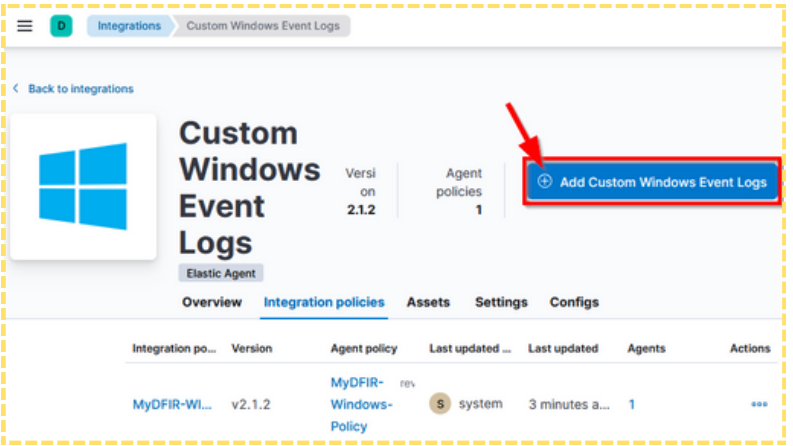


- Click "Save and Continue" then click "Save and Deploy Changes"

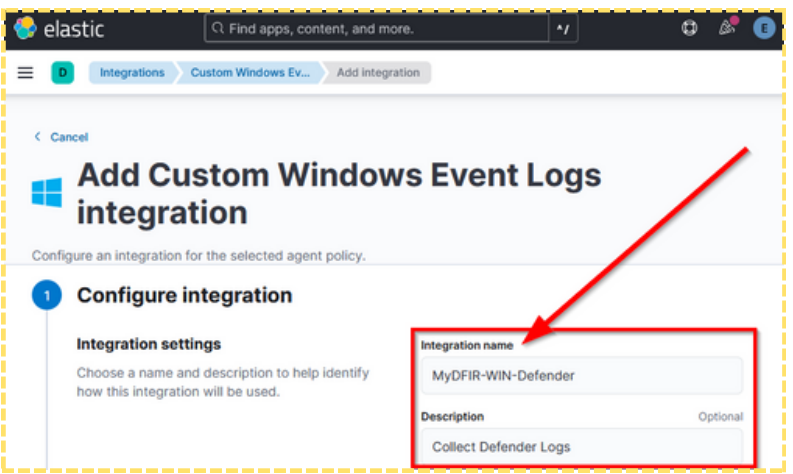


2. Microsoft Defender Log Ingestion

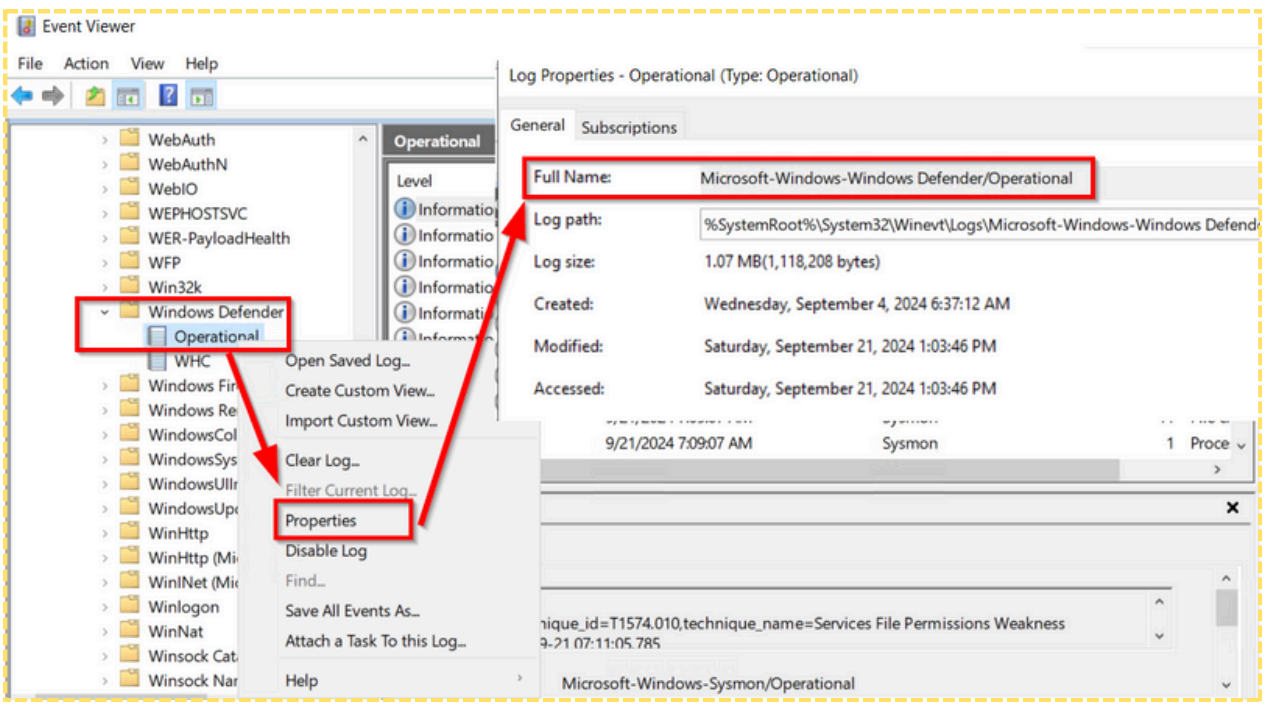
- Repeats the process for adding another Custom Windows Event Log integration
- Click "Add Custom Windows Event Logs" again



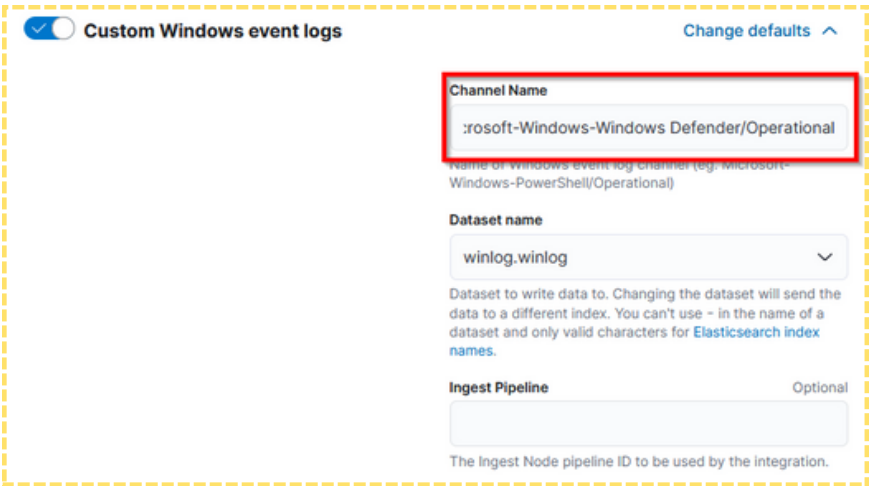
- Type name is "MyDFIR-Win-Defender" and provide a description, such as "Collect Defender logs"



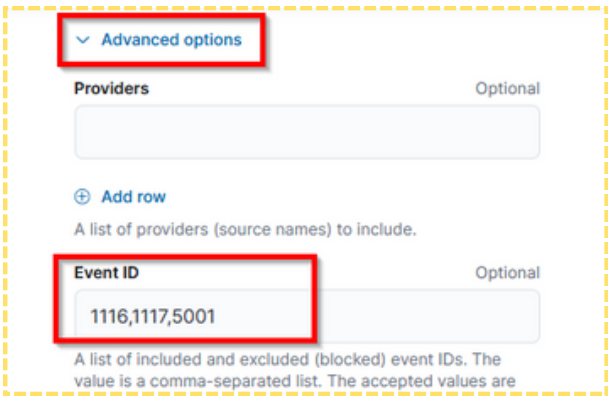
- Finding channel Name in Windows Event Viewer, navigate to "Windows Defender" > "Operational"
- Right click "Operational," select "Properties," and copy the full nam



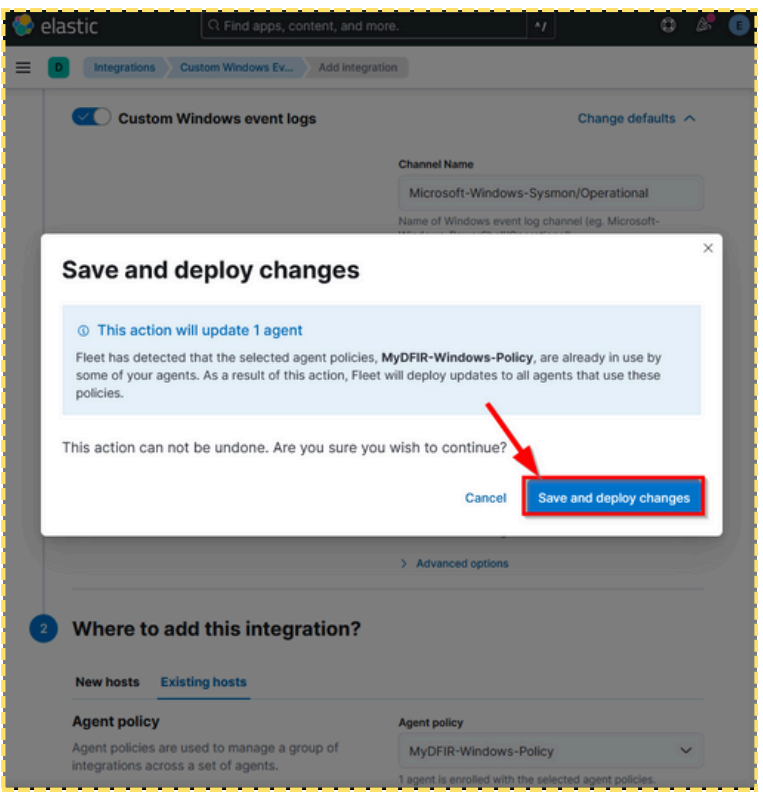
- Paste the full name into the Channel Name on Elasticsearch costom integration



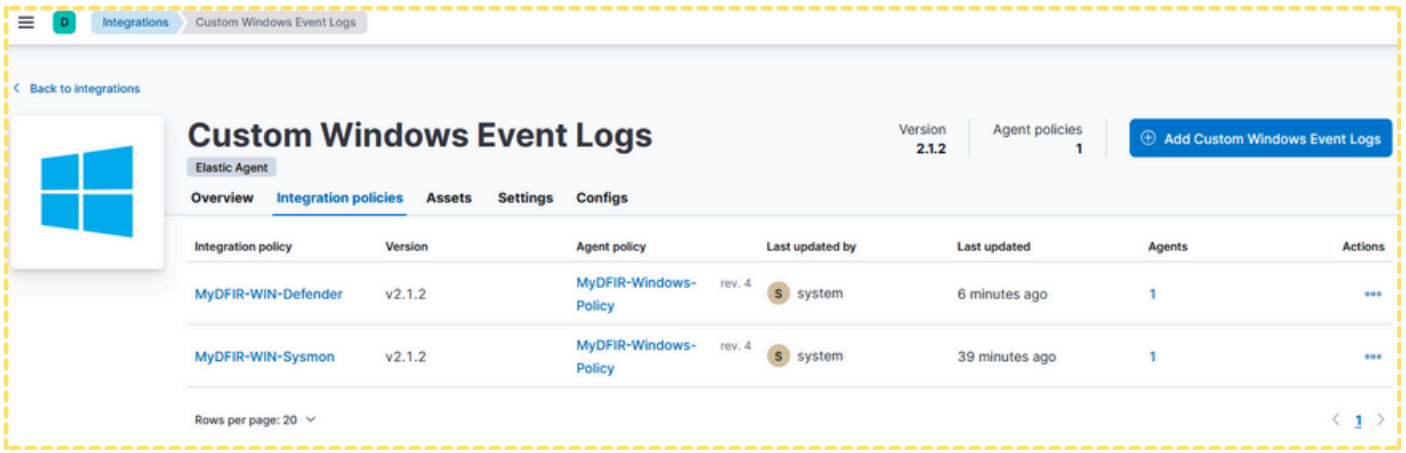
- Under "Advanced Options," specify the event IDs. Filter on specific event IDs like 1116 (malware detected), 1117 (protection action taken), and 5001 (real-time protection disabled)



- Select the existing host and agent policy ("MyDFIR-Windows-Policy") for the integration
- Click "Save and Continue," then click "Save and Deploy Changes”

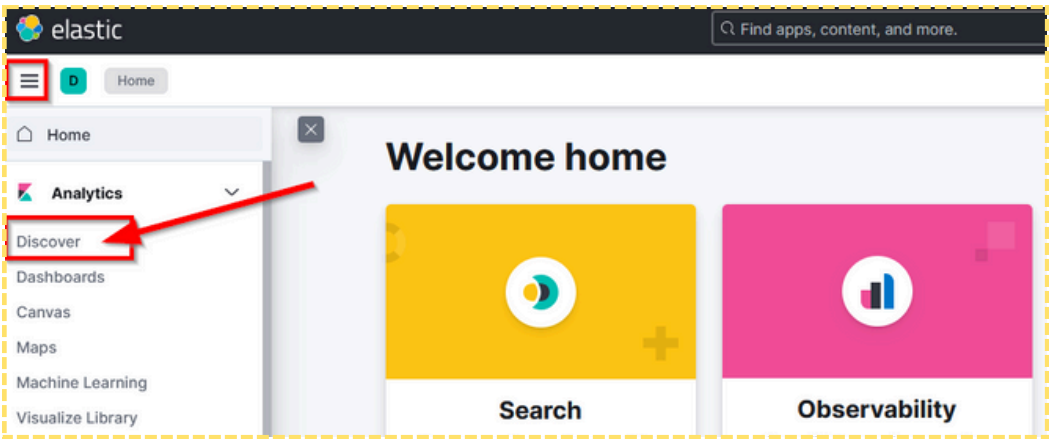


- Once both Sysmon and Microsoft Defender log integrations are configured, integration policies should now be ready

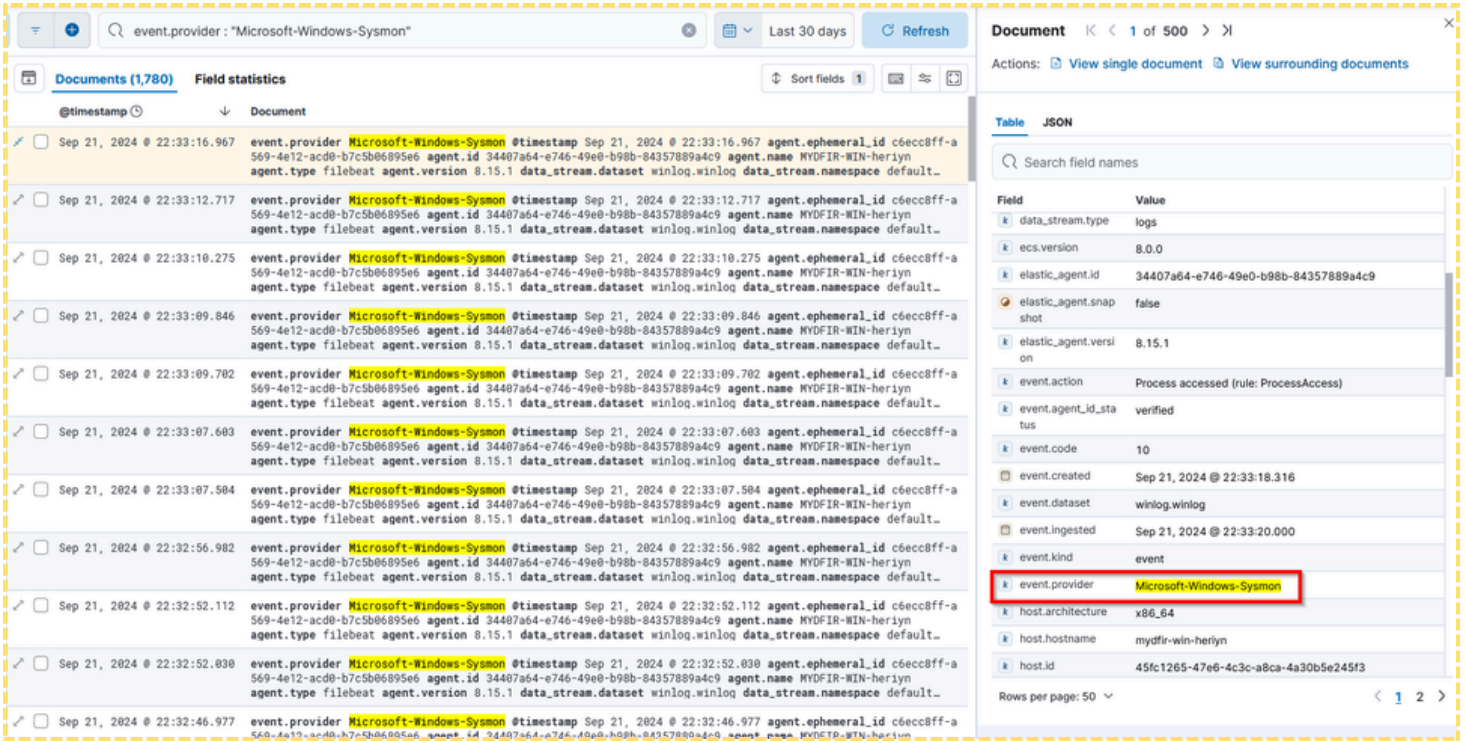


3. Verifying Log Ingestion

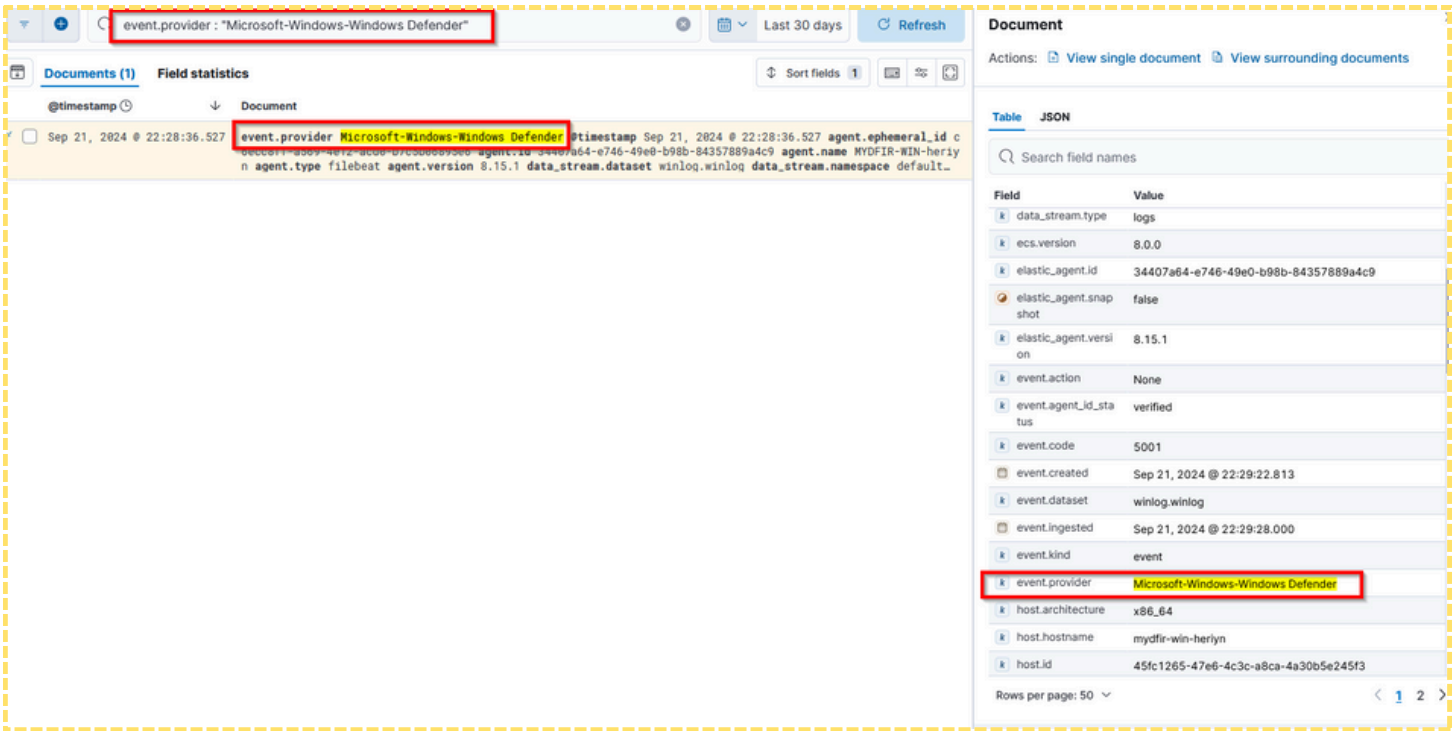
- Go to ELK Menu and select "Discover"



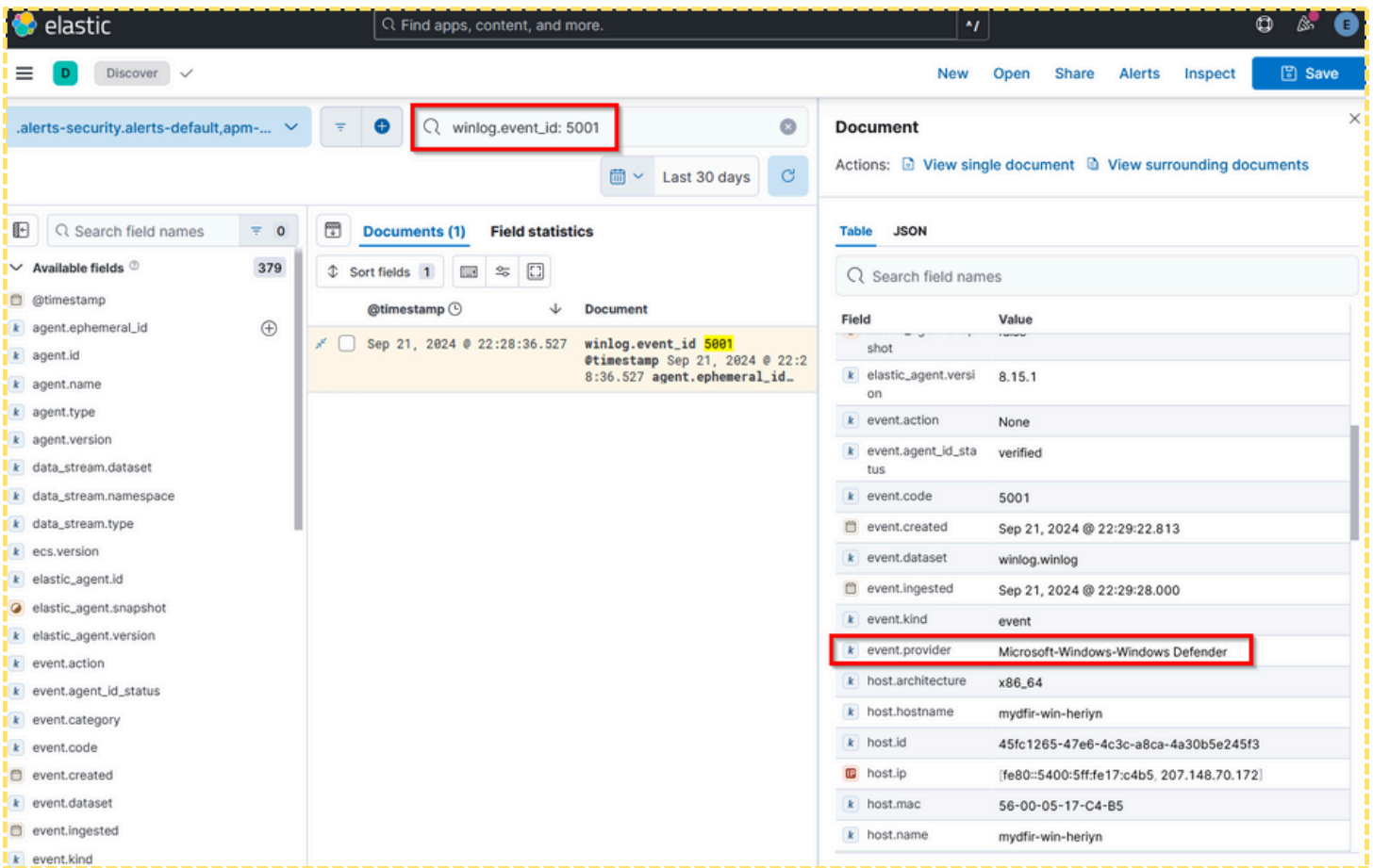
- Search for Sysmon logs, enter the query: winlog.provider_name: "Microsoft-Windows-Sysmon"



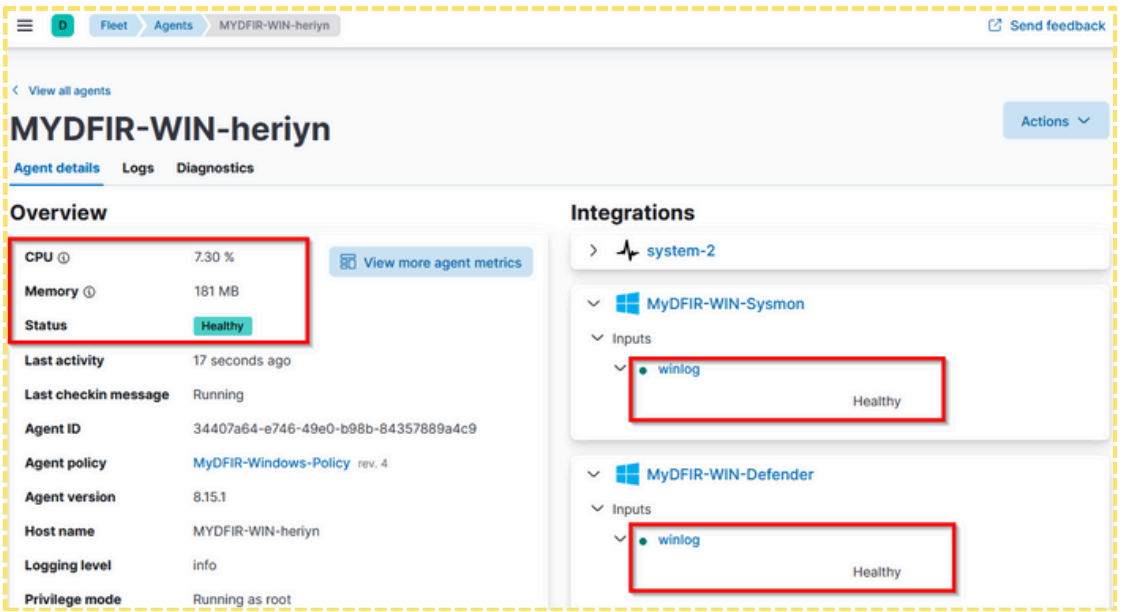
- Search for Microsoft Defender logs, Use the query: winlog.provider_name: "Microsoft-Windows-Windows Defender"



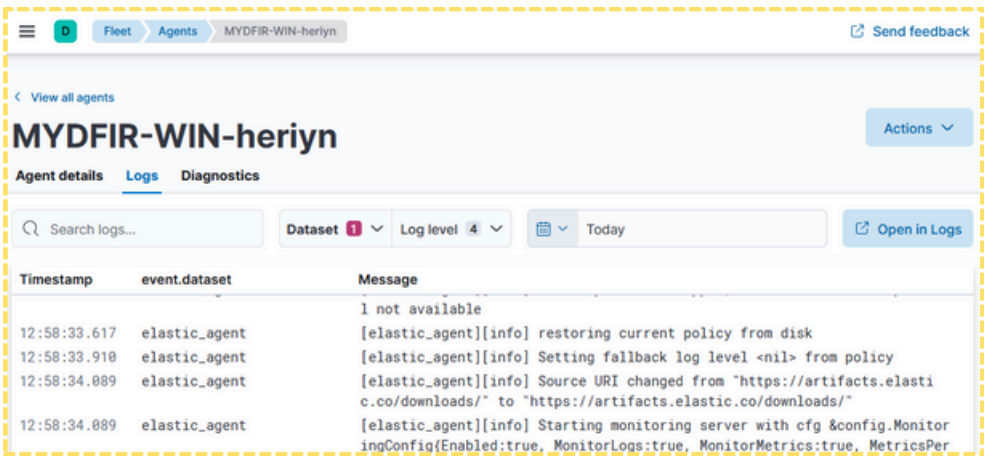
- Sysmon process creation events: winlog.event_id: 1



- Navigates to the Fleet section and click on the host to view agent details, expand the "winlog" section and verify that the status is "Healthy



- Click inside the "winlog" section to view log messages



4. Troubleshooting Tips

- Check Agent Status: In Elasticsearch Fleet, verify that the agent is healthy and has a recent activity timestamp. If the CPU and memory values for the agent show "N/A," it indicates that the agent cannot communicate with Elasticsearch
- Review Agent Logs: Check the agent logs for any error messages.
- Restart Agent Service: Restart the Elastic Agent service on the Windows server.
- Verify Firewall Rules: Ensure that the firewall rules are correctly configured to allow connections to port 9200.
- Test Connectivity: Use tools like telnet or nc to test connectivity to port 9200 on the Elasticsearch server from the Windows server.
- Search for Specific Events: Use the Discover interface to search for specific event IDs and verify that they are being ingested.