SOC Automation Lab: Integrating Wazuh, TheHive, Shuffle, and VirusTotal

Lab Overview

This lab demonstrates how Security Operations Center (SOC) tasks can be automated by integrating multiple open-source tools. The goal is to reduce manual workload, improve detection and response speed, and provide security analysts with streamlined alerts.

Objectives

By the end of this lab, a user will be able to:

- Monitor endpoints and servers using **Wazuh SIEM**
- Create and customize **rules and indexes** in Wazuh
- Add and manage **Wazuh agents** for log collection
- Detect suspicious activities (e.g., **Mimikatz usage**)

- Extract and analyze **file hashes (SHA-256)** from Wazuh logs
- Automatically submit hashes to **VirusTotal** for reputation checks
- Trigger automated workflows in **Shuffle** for alert enrichment
- Forward critical alerts to **TheHive** for case management
- Send **email notifications** to administrators when incidents are detected

Prerequisites

Before you begin, ensure you have the following environment and tools ready:

Local Environment

- **Host OS**: Windows 10/11
- **Virtualization**: VMware Workstation or VMware Player
- **Windows 11 VM**: Used as the endpoint machine (Sysmon + Wazuh agent installed)

Cloud Environment (DigitalOcean)

- **Ubuntu VM #1**: Wazuh SIEM Server
- **Ubuntu VM #2**: TheHive (Incident Response Platform)

Tools & Services

- **Sysmon** (installed on Windows 11 VM for detailed event logging)
- **Wazuh Agent** (installed on Windows 11 VM for log forwarding)
- **Shuffle** (cloud-hosted, used for automation workflows)
- **VirusTotal** (Free API key for hash reputation checks)
- **Strangebee License** (Free Trial)
    - Registration on StrangeBee requires a valid **work** or **student** email address.

Learning Outcomes

During the development of this lab, I gained hands-on experience with:

- Deploying and configuring **Wazuh** for log analysis
- Creating custom **detection rules and indexes**
- Building **automation workflows** in Shuffle
- Ingesting alerts into **TheHive** for structured incident response
- Integrating third-party threat intelligence (**VirusTotal**) into the SOC pipeline

Setting Up Windows 11 Virtual Machine

In this section, you'll set up a Windows 11 virtual machine, which will act as your monitored endpoint in the SOC Automation Lab.

1. Download Windows 11 ISO

1. Go to the official Microsoft download page: 👉 Download Windows 11 Disk Image (ISO)
2. Under **"Download Windows 11 Disk Image (ISO) for x64 devices"**, select:
3. **Windows 11 (multi-edition ISO for x64 devices)** → click **Confirm**
4. Choose **Language** → *English (United States)* → click **Confirm**
5. Select **64-bit Download** → the ISO file download will begin.

2. Create Windows 11 VM in VMware

1. Open **VMware Workstation / Player**.
2. Click **Create a New Virtual Machine**.
3. Select **Typical (recommended)** → click **Next**.
4. Choose **Installer disc image file (ISO)** and browse to the Windows 11 ISO you downloaded → click **Next**.

5. Enter a username and password (make sure to remember it, you'll need it later). → click **Next**.

6. Specify the **disk size** (use the recommended value) → click **Next**.

7. (Optional) Customize hardware settings (e.g., RAM, CPU, or network adapter). If unsure, the default configuration works fine.

8. Check **Power on this virtual machine after creation**.

9. Click **Finish**.

3. Complete Windows 11 Installation

1. The VM will start and boot from the ISO.

2. Follow the on-screen installation steps (language, edition, setup options).

3. Once installation is complete, log in with the password you created earlier.

Now you have a working **Windows 11 VM** inside VMware. This will be used later for deploying and testing SOC automation tools (Wazuh, TheHive, Shuffle).

Installing Sysmon on Windows 11 VM

In this step, you'll install Sysmon on your Windows 11 VM to generate rich security logs that Wazuh will later monitor and analyze.

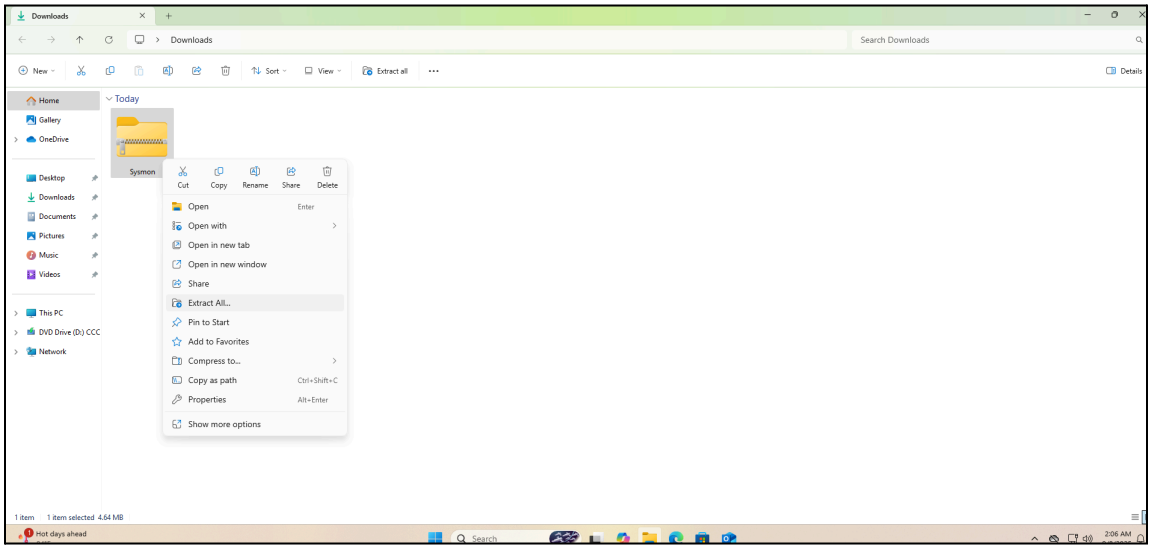
Why Sysmon?

Sysmon (System Monitor) is a Windows system service and device driver from the Sysinternals suite. It monitors and logs system activity, such as process creation, network connections, and file changes. In this project, you will use Sysmon to generate detailed security logs that can later be ingested by **Wazuh** for detection and automated response.

1. Download Sysmon

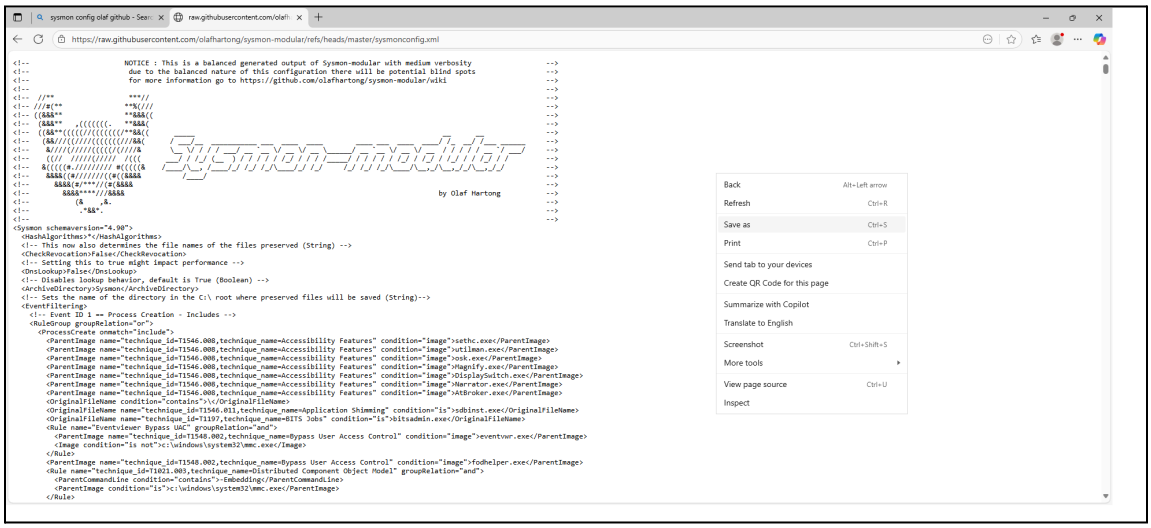
1. Inside your Windows 11 VM, open **Microsoft Edge**.
2. Go to the official Sysmon download page: 👉 Sysmon – Windows Sysinternals
3. Click **Download Sysmon**.
4. Once the download is complete, go to the download location.

5. Right-click the Sysmon archive and select **Extract All** to unzip it.



2. Download Sysmon Configuration

1. Go to the olafhartong Sysmon Modular configuration on GitHub:[Sysmon Modular Configuration (olafhartong)]
2. Right-click on the page → **Save As**.

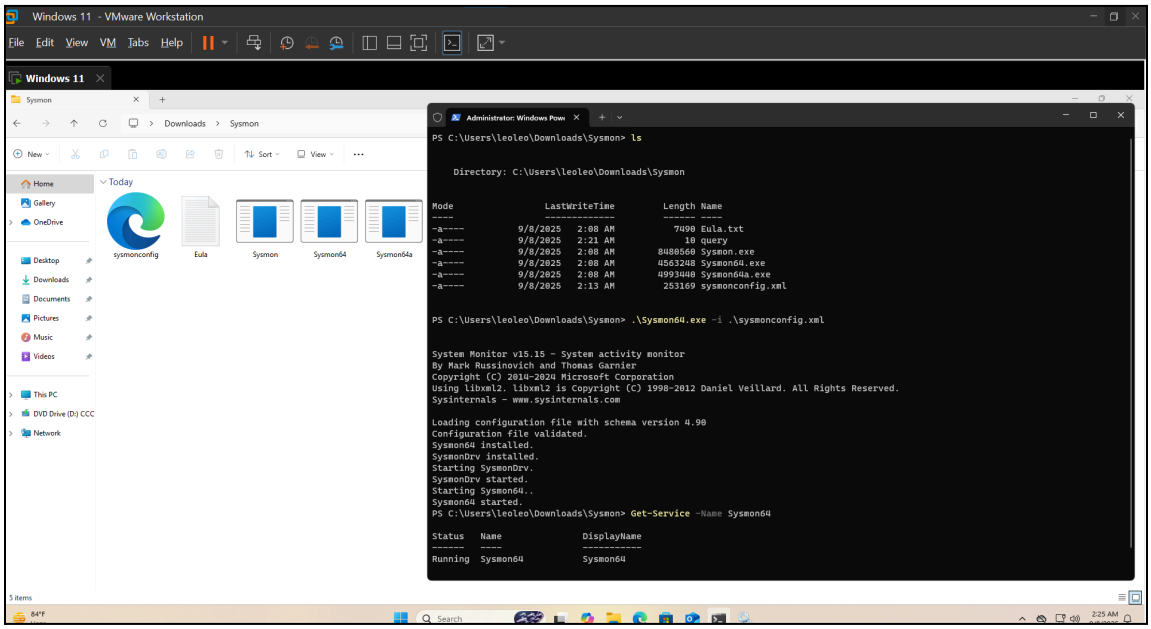


3. Name the file **`sysmonconfig.xml`**
4. Save it in the same Sysmon folder you extracted earlier.

3. Install Sysmon with Configuration

1. Open **PowerShell as Administrator**.
2. Navigate to the Sysmon directory: `cd "C:\path\to\Sysmon"`

3. Run the installation command: `.\Sysmon64.exe -i sysmonconfig.xml`

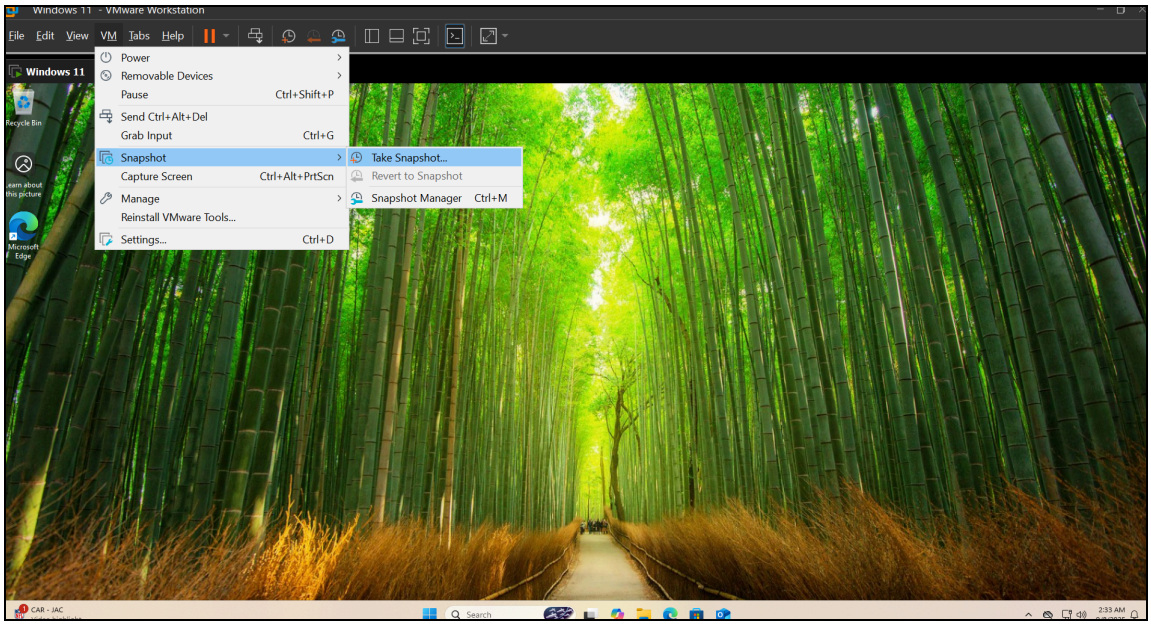


4. Accept the license agreement.
5. After installation, you should see the message: **`Sysmon64 started.`**
6. Run the following command to confirm Sysmon is running: `Get-Service -Name sysmon64`
7. If the **Status** shows **`Running`**, Sysmon has been successfully installed.
8. To verify logs are being generated, open Event Viewer → **Applications and Services Logs → Microsoft → Windows → Sysmon → Operational.**

4. Create a VMware Snapshot

It's a good practice to save your VM state after installing Sysmon.

1. In VMware, go to the top menu → **VM** → **Snapshot** → **Take Snapshot**.



2. Give your snapshot a descriptive name (e.g., *After Sysmon Installation*).
3. Click **OK**.

✅ You now have **Sysmon installed and running** with a modular configuration, and a **VM snapshot** saved for rollback if needed.

Setting Up Wazuh and TheHive on DigitalOcean

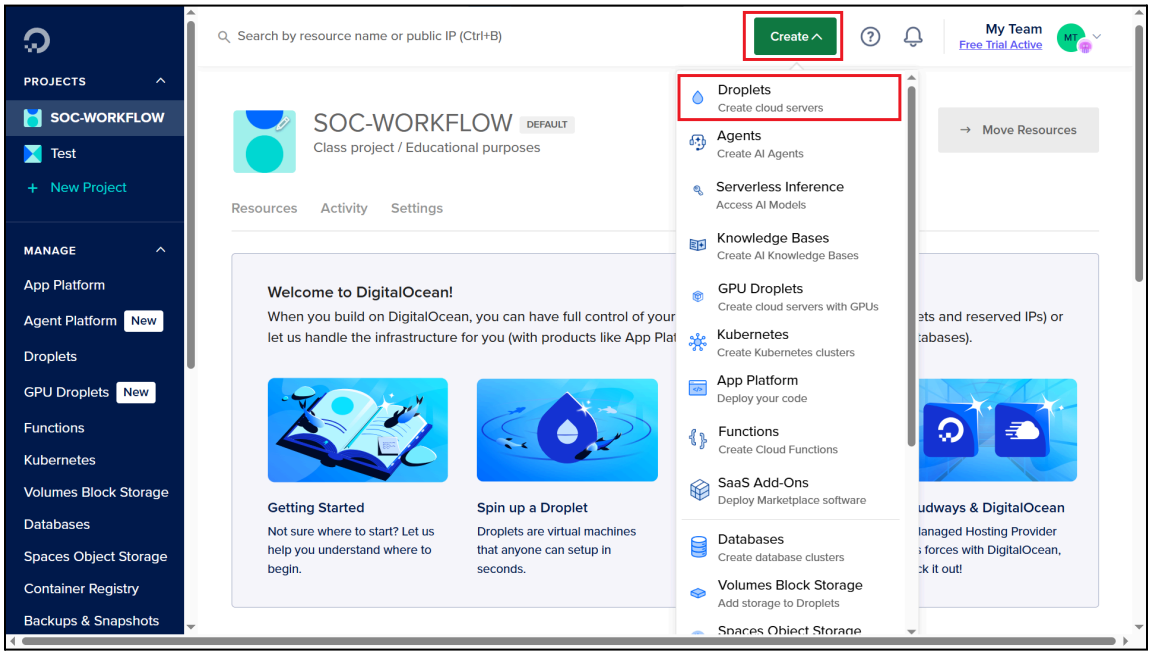
In this step, you'll deploy cloud-based Ubuntu servers for Wazuh and TheHive using DigitalOcean. These will act as your SOC's backend systems.

1. Sign Up for DigitalOcean

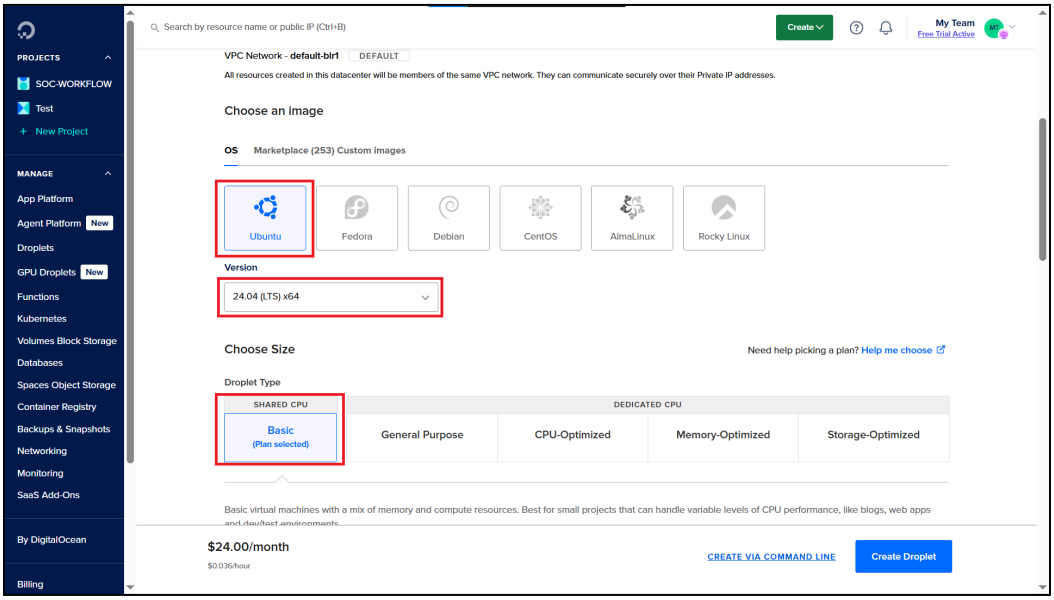
1. You can sign up here on DigitalOcean: https://m.do.co/c/28868f21cdd3
2. You can also use Vultr as an alternative: https://www.vultr.com/register/
3. Create an account using a **valid credit/debit card**.
    - DigitalOcean will temporarily charge **$1 for verification**, which is refunded after setup.
4. After logging in, go to **Projects** → click **New Project**.
5. Give your project a **name** (e.g., `SOC-Automation`) and select its **purpose**.
6. Click **Create Project**.

2. Create Wazuh Droplet

1. Inside your project, click **Create** → select **Droplets**.

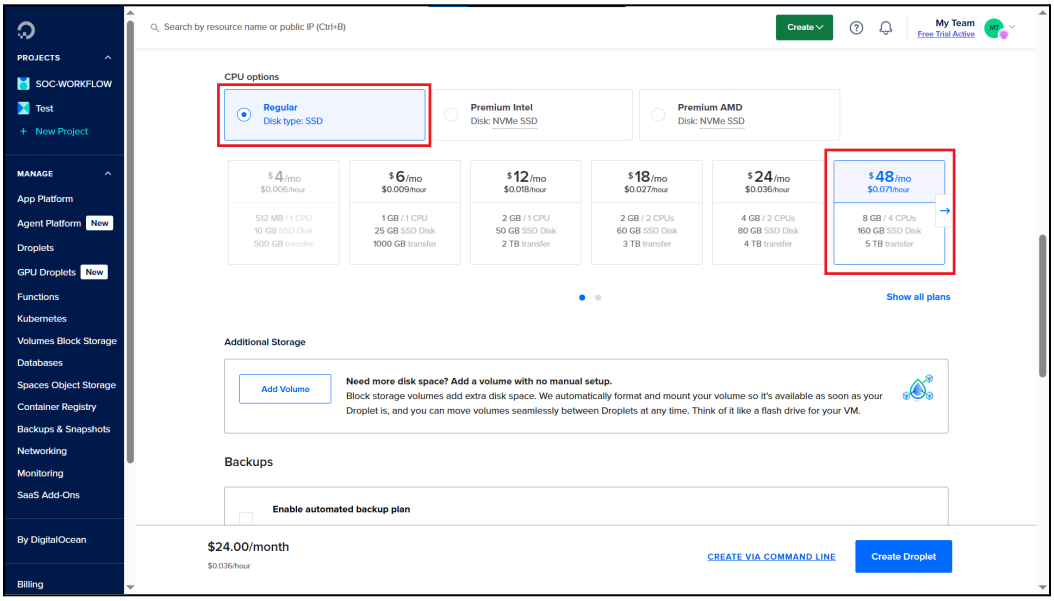


2. Configure your Droplet as follows:
   a. **Image**: Ubuntu 24.04 (LTS) x64

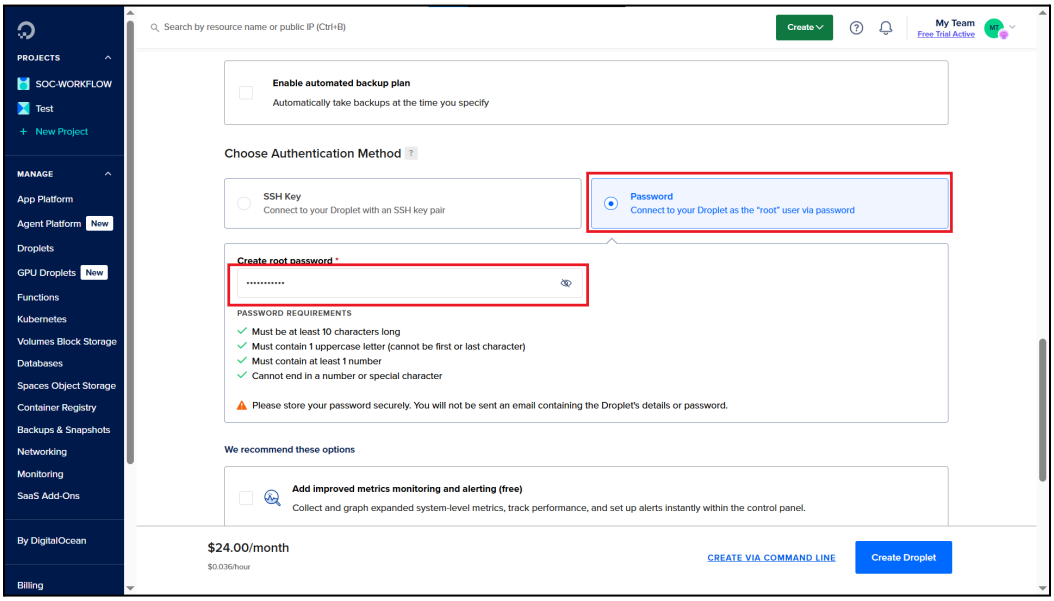


   b. **CPU Options**: Regular

c. **Size**: 8 GB RAM / 4 vCPUs / 160 GB SSD Disk / 5 TB Transfer



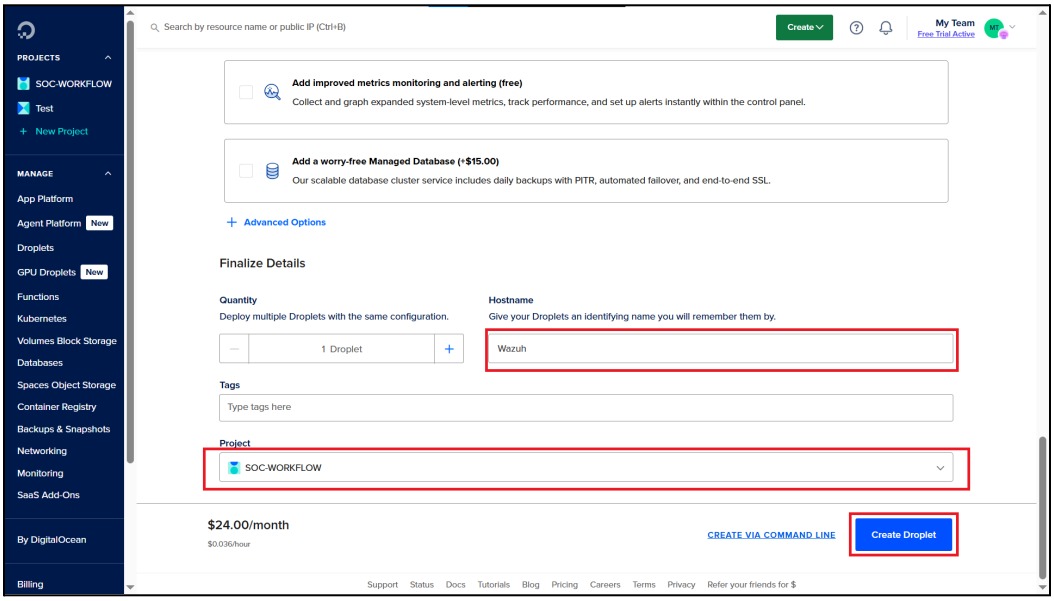
d. **Authentication**: Password (create a strong password and note it down)



e. **Hostname**: **Wazuh** (you can choose any name, but functional names keep things organized)

f. Select the project you created earlier.

g. Click **Create Droplet**.



h. To connect to your Droplet later, use: `ssh root@<droplet-ip>`

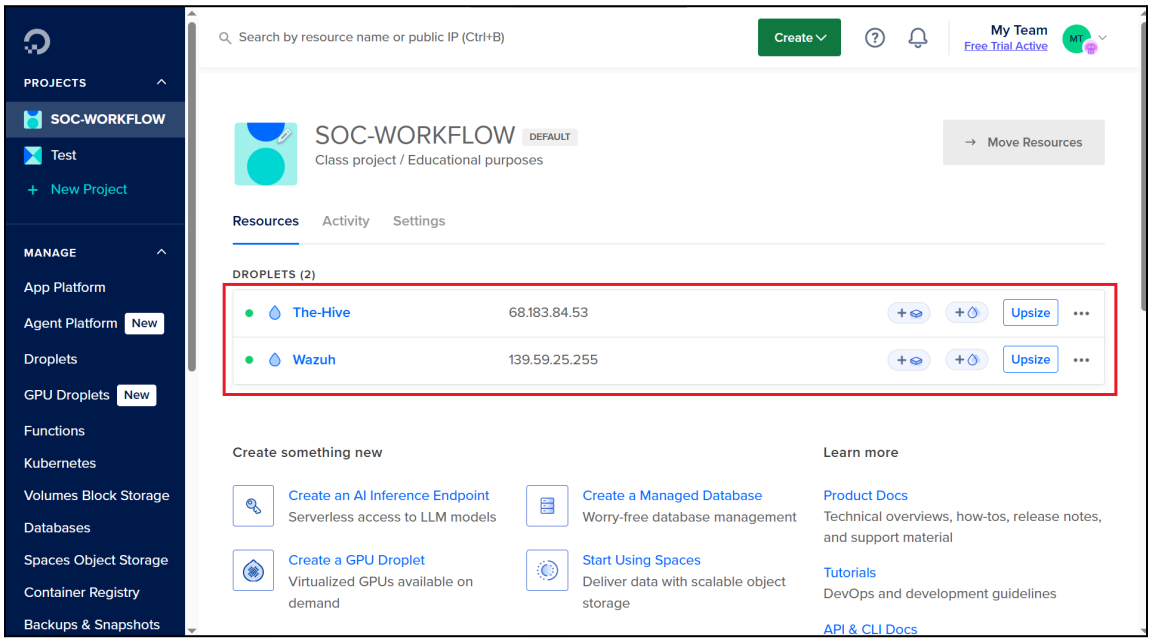
3. Create TheHive Droplet

Repeat the same steps as above, but this time:

- **Hostname**: **`TheHive`**
- **Recommended Size**: Ideally 16 GB RAM (TheHive is more resource-intensive than Wazuh).
    - If 16 GB isn't available in your plan, use the same specs as the Wazuh Droplet (8 GB RAM / 4 vCPUs).

4. Verify Droplets

1. After creation, you will see both VMs listed under your project.



2. Note down the **public IP addresses** for **`Wazuh`** and **`TheHive`**.
3. You will use these IPs later to connect via SSH and install the respective tools.

✅ At this point, you have two Ubuntu VMs running on DigitalOcean:

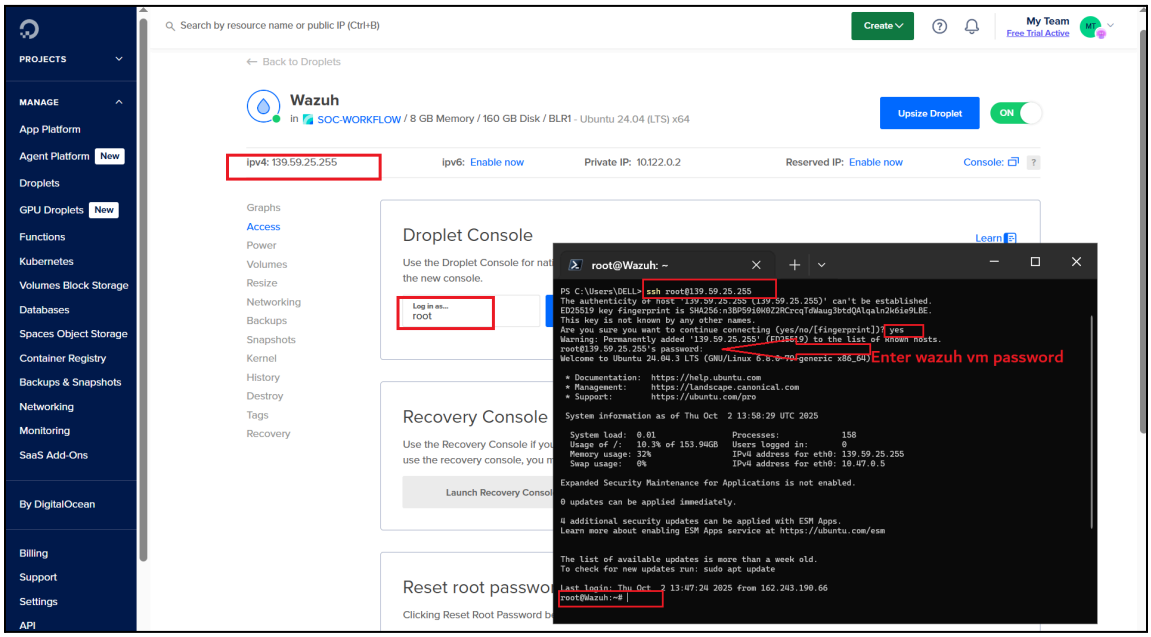
- **`Wazuh`** → for SIEM setup
- **`TheHive`** → for incident response and case management

Droplet	Purpose	Recommended Specs	Hostname
Wazuh	SIEM & log analysis	8 GB RAM / 4 vCPUs	Wazuh
The-Hive	Incident response	16 GB RAM / 4 vCPUs	The-Hive

Connect to the Wazuh VM

1. Open **PowerShell** on your local machine.
2. Run the following command (replace with your Wazuh VM IP): `ssh root@{your-wazuh-vm-ip}`
3. Type **`yes`** and press **Enter**.

4. Enter the VM password you created during set



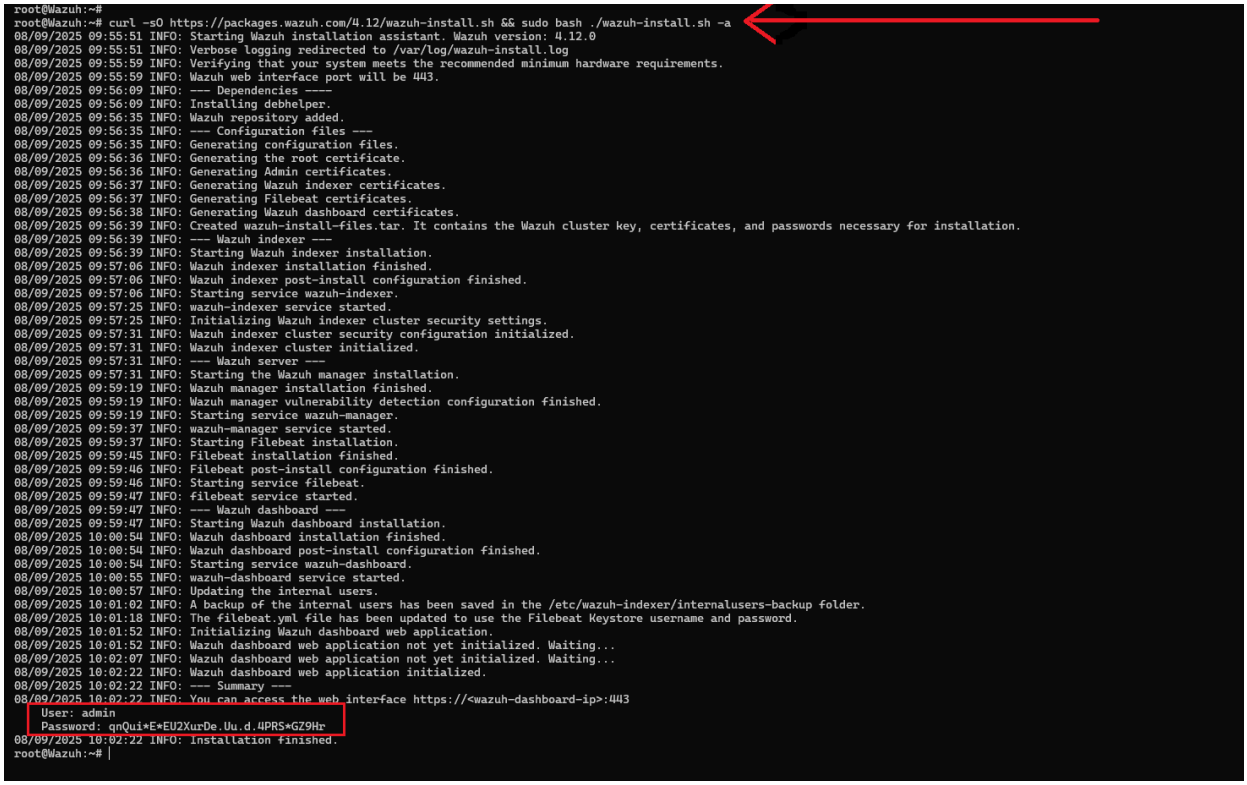
You are now inside your Wazuh VM.

Run the following to update your packages:

```
apt-get update && apt-get upgrade -y
```

Install Wazuh

1. Go to the official Wazuh installation guide: [Wazuh Quickstart]
2. Copy the **latest installation command** shown there.
3. Paste it into your Wazuh VM and press **Enter**.



If everything is successful, you will see:

- A generated **user and password** → Write these down! You will need them for the Wazuh web dashboard.
- **Installation finished message**

Verify Wazuh Service

Check the status of the Wazuh manager service: `systemctl status wazuh-manager.service`

- If it's **running** → you're good to go.
- If not, restart the service and check again:

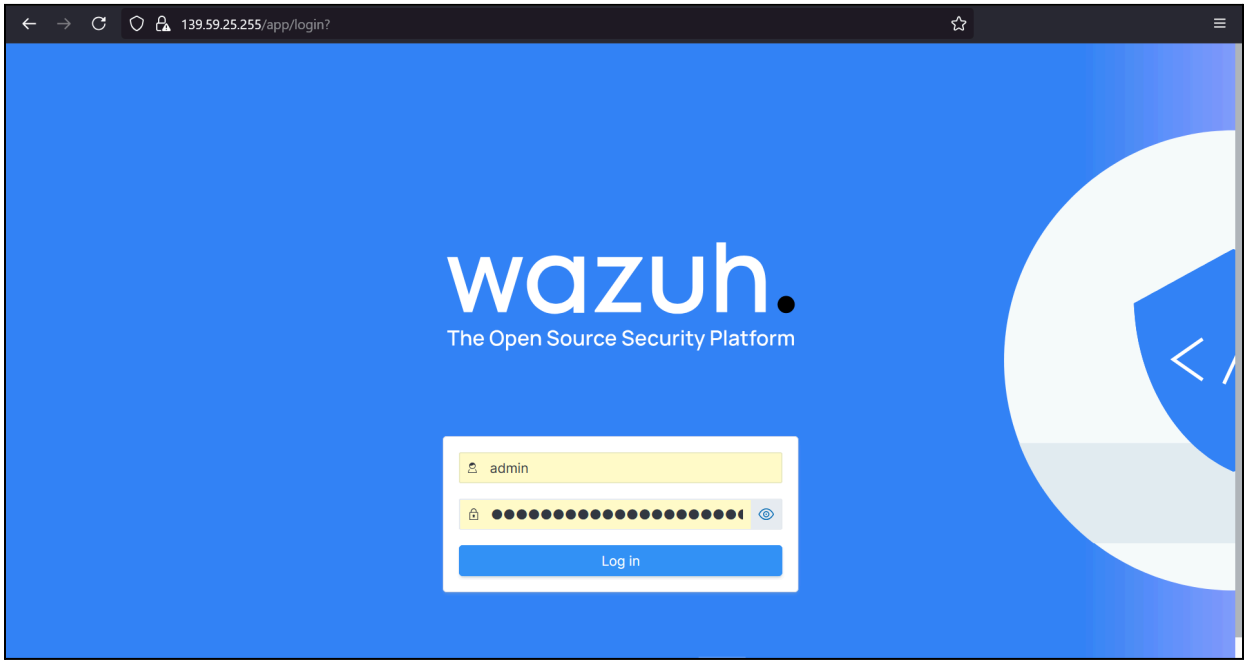
```
systemctl restart wazuh-manager.service
systemctl status wazuh-manager.service
```

```
root@Wazuh:~# systemctl status wazuh-manager.service
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-10-03 02:50:33 UTC; 1h 33min ago
    Process: 832 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 176 (limit: 9486)
     Memory: 2.5G (peak: 2.7G)
        CPU: 1min 58.231s
     CGroup: /system.slice/wazuh-manager.service
             ├─1235 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─1236 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─1237 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─1240 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─1243 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─1266 /var/ossec/bin/wazuh-integratord
             ├─1287 /var/ossec/bin/wazuh-authd
             ├─1300 /var/ossec/bin/wazuh-db
             ├─1326 /var/ossec/bin/wazuh-execd
             ├─1341 /var/ossec/bin/wazuh-analysisd
             ├─1353 /var/ossec/bin/wazuh-syscheckd
             ├─1367 /var/ossec/bin/wazuh-remoted
             ├─1449 /var/ossec/bin/wazuh-logcollector
             ├─1466 /var/ossec/bin/wazuh-monitord
             └─1479 /var/ossec/bin/wazuh-modulesd

Oct 03 02:50:28 Wazuh env[832]: Started wazuh-analysisd...
Oct 03 02:50:28 Wazuh env[832]: Started wazuh-syscheckd...
Oct 03 02:50:29 Wazuh env[832]: Started wazuh-remoted...
Oct 03 02:50:29 Wazuh env[832]: Started wazuh-logcollector...
Oct 03 02:50:30 Wazuh env[832]: Started wazuh-monitord...
Oct 03 02:50:30 Wazuh env[1475]: 2025/10/03 02:50:30 wazuh-modulesd:router: INFO: Loaded router module.
Oct 03 02:50:30 Wazuh env[1475]: 2025/10/03 02:50:30 wazuh-modulesd:content_manager: INFO: Loaded content_manager module.
Oct 03 02:50:31 Wazuh env[832]: Started wazuh-modulesd...
Oct 03 02:50:33 Wazuh env[832]: Completed.
Oct 03 02:50:33 Wazuh systemd[1]: Started wazuh-manager.service - Wazuh manager.
root@Wazuh:~#
```

Access the Wazuh Web Interface

1. Open your browser and go to: **https://{your-wazuh-vm-ip}**
2. If you cannot access it, allow HTTPS traffic (port 443): `sudo ufw allow 443`
3. Try again — you should now see the Wazuh dashboard.
4. Log in using the **username and password** you saved earlier.

Once logged in, you should see the Wazuh Dashboard Overview showing system status, agent summary, and alert statistics. This confirms that your Wazuh installation is active and operational.✅



👉 If it still doesn't work, check your **cloud provider firewall settings** (DigitalOcean, AWS, Azure, Vultr, etc.) and allow port **443**.

Installing TheHive

Register for TheHive

Go to the official StrangeBee portal and register for an account: 👉 Register here

📌 **Note:** TheHive requires a work or student email. They also provide a free trial.

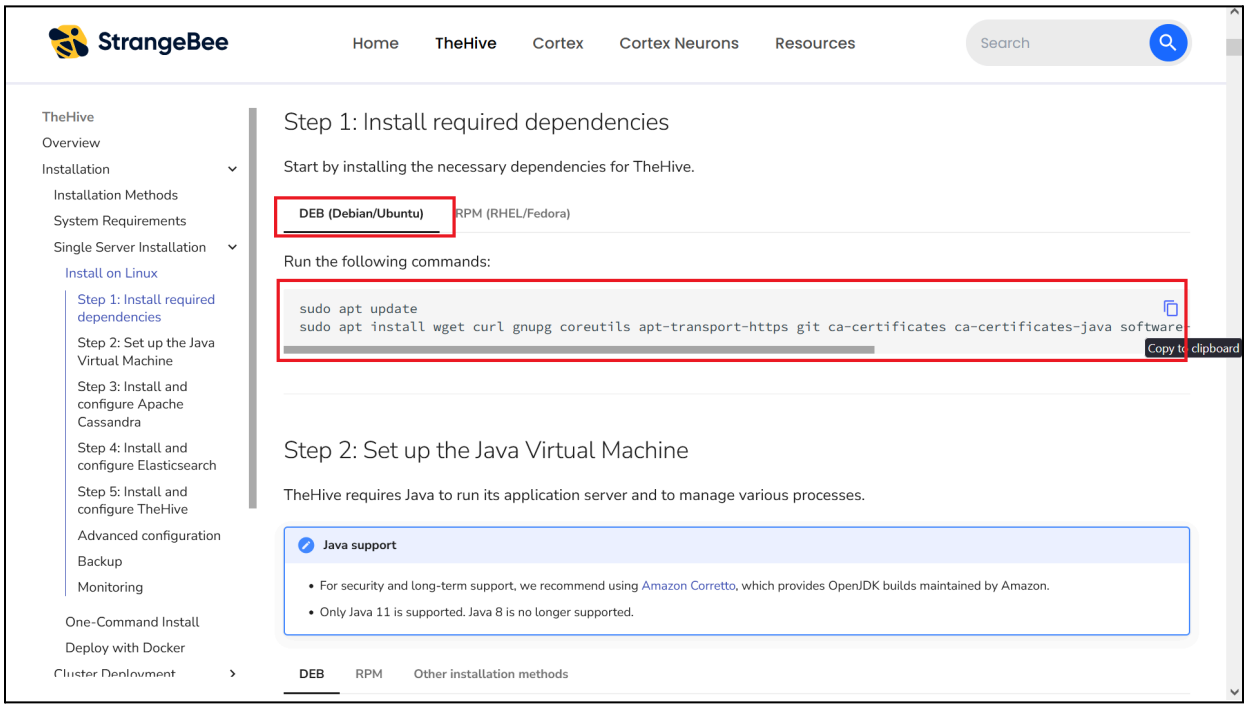
Connect to TheHive VM

- Open PowerShell on your local machine and connect via SSH (replace with your TheHive VM IP): `ssh root@{your-thehive-vm-ip}`
- Type **yes** when prompted, then enter your VM password.
- Update your system packages: `sudo apt update && apt upgrade -y`

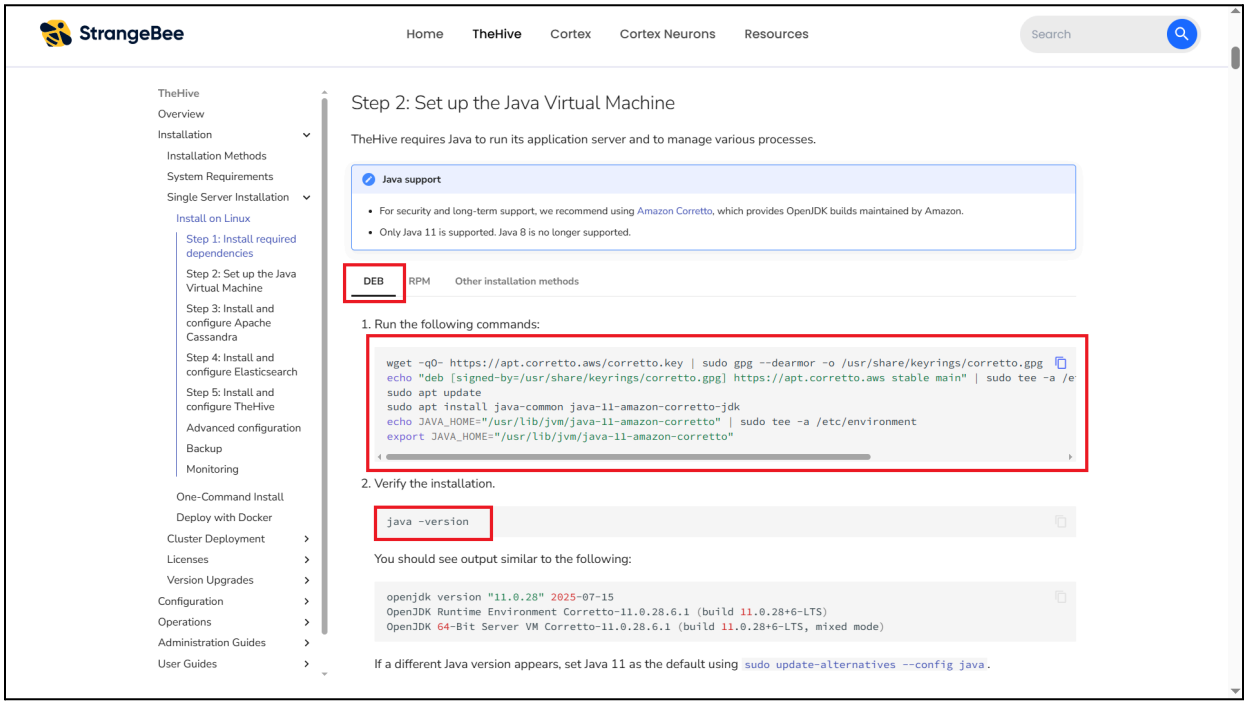
Install Dependencies

Follow the official StrangeBee installation guide:
👉 TheHive Installation Guide (Linux Standalone Server)

Copy and run the commands exactly as shown in the documentation.




```
root@The-Hive:~# sudo apt update
sudo apt install wget curl gnupg coreutils apt-transport-https git ca-certificates ca-certificates-java software-properties-common python3-pip l
sb-release unzip
Hit:1 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu noble InRelease
Hit:3 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
coreutils is already the newest version (9.4-3ubuntu6.1).
coreutils set to manually installed.
apt-transport-https is already the newest version (2.8.3).
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
ca-certificates is already the newest version (20240203).
software-properties-common is already the newest version (0.99.49.3).
software-properties-common set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu dpkg-dev
  fakeroot g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu
  javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libatomic1 libbinutils libcc1-0
  libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-13-dev libgomp1 libgprofng0 libhwasan0 libisl23
  libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libpython3-dev libpython3.12-dev libquadmath0 libsframe1
  libstdc++-13-dev libtsan2 libubsan1 lto-disabled-list make python3-dev python3-wheel python3.12-dev zlib1g-dev
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc cpp-doc gcc-13-locales cpp-13-doc debian-keyring g++-multilib g++-13-multilib gcc-13-doc gcc-multilib
```




```
root@The-Hive:~# wget -qO- https://apt.corretto.aws/corretto.key | sudo gpg --dearmor -o /usr/share/keyrings/corretto.gpg
echo "deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main" | sudo tee -a /etc/apt/sources.list.d/co
rretto.sources.list
sudo apt update
sudo apt install java-common java-11-amazon-corretto-jdk
echo JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto" | sudo tee -a /etc/environment
export JAVA_HOME="/usr/lib/jvm/java-11-amazon-corretto"
deb [signed-by=/usr/share/keyrings/corretto.gpg] https://apt.corretto.aws stable main
Hit:1 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Get:2 https://apt.corretto.aws stable InRelease [10.7 kB]
Hit:3 http://mirrors.digitalocean.com/ubuntu noble InRelease
Get:4 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease [126 kB]
Get:5 https://apt.corretto.aws stable/main amd64 Packages [22.5 kB]
Hit:6 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
0% [Waiting for headers] [7 InRelease 14.2 kB/126 kB 11%]
```

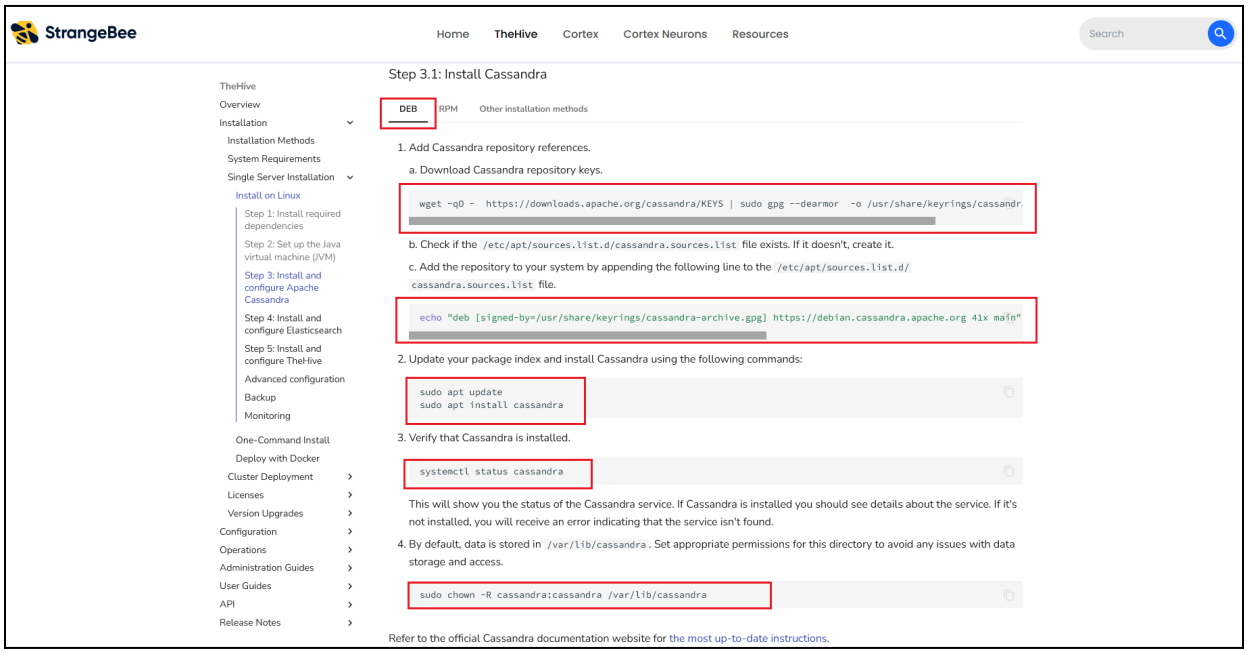
After installing Java, verify the version: `java -version`

If no version is displayed, rerun the Java installation command (the one starting with **`wget`** `...`),

then confirm installation again with: `java -version`

Install Cassandra

Follow the official Apache Cassandra installation guide

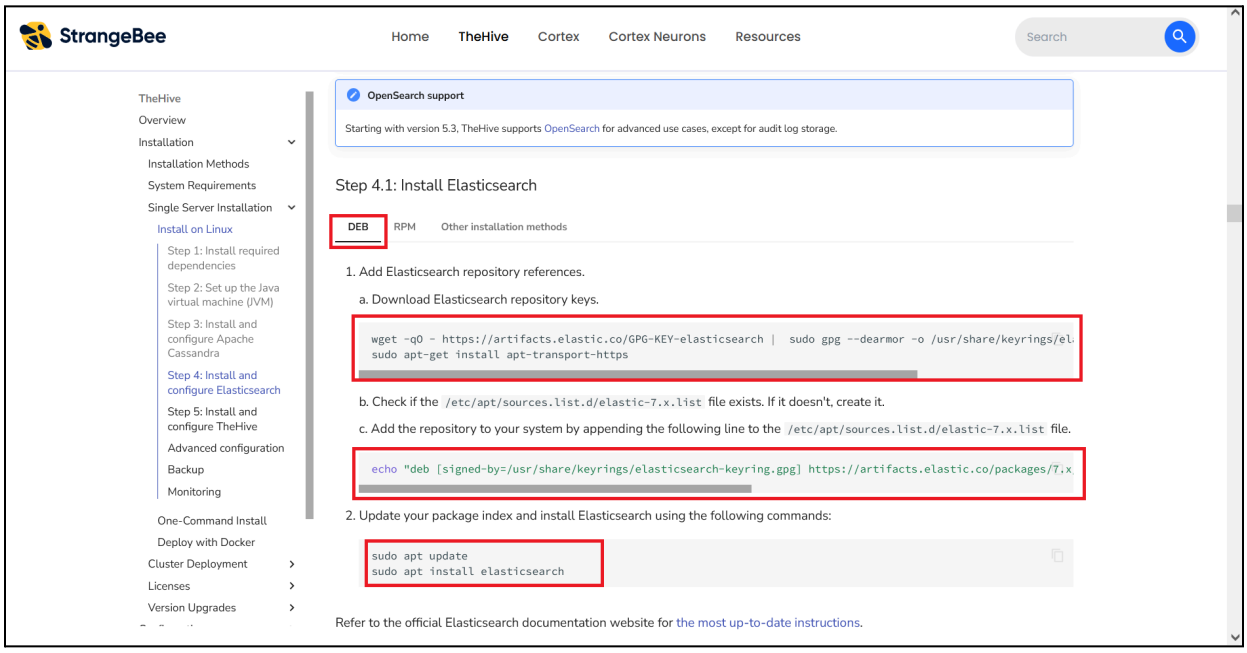



```
root@The-Hive:~# wget -qO -  https://downloads.apache.org/cassandra/KEYS | sudo gpg --dearmor  -o /usr/share/keyrings/cassandra-archive.gpg
root@The-Hive:~# echo "deb [signed-by=/usr/share/keyrings/cassandra-archive.gpg] https://debian.cassandra.apache.org 41x main" |  sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list
deb [signed-by=/usr/share/keyrings/cassandra-archive.gpg] https://debian.cassandra.apache.org 41x main
root@The-Hive:~# sudo apt update
sudo apt install cassandra
Hit:1 https://apt.corretto.aws stable InRelease
Hit:2 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:3 http://mirrors.digitalocean.com/ubuntu noble InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease
Hit:5 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease
Get:6 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:7 https://apache.jfrog.io/artifactory/cassandra-deb 41x InRelease [3902 B]
Get:8 https://apache.jfrog.io/artifactory/cassandra-deb 41x/main amd64 Packages [696 B]
Fetched 131 kB in 3s (50.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  cassandra-tools
The following NEW packages will be installed:
  cassandra
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 50.7 MB of archives.
After this operation, 61.3 MB of additional disk space will be used.
Get:1 https://apache.jfrog.io/artifactory/cassandra-deb 41x/main amd64 cassandra all 4.1.10 [50.7 MB]
Fetched 50.7 MB in 8s (6326 kB/s)
Selecting previously unselected package cassandra.
(Reading database ... 110320 files and directories currently installed.)
Preparing to unpack .../cassandra_4.1.10_all.deb ...
Unpacking cassandra (4.1.10) ...
Setting up cassandra (4.1.10) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group `cassandra' (GID 112) ...
vm.max_map_count = 1048575
net.ipv4.tcp_keepalive_time = 300
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@The-Hive:~# systemctl status cassandra
● cassandra.service - LSB: distributed storage system for structured data
     Loaded: loaded (/etc/init.d/cassandra; generated)
     Active: active (running) since Fri 2025-09-12 09:28:58 UTC; 9s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 5884 ExecStart=/etc/init.d/cassandra start (code=exited, status=0/SUCCESS)
      Tasks: 26 (limit: 9486)
     Memory: 2.1G (peak: 2.1G)
        CPU: 13.125s
     CGroup: /system.slice/cassandra.service
             └─6046 /usr/bin/java -ea -da:net.openhft... -XX:+UseThreadPriorities -XX:+HeapDumpOnOutOfMemoryError -Xss256k -XX:+Alwa>

Sep 12 09:28:58 The-Hive systemd[1]: Starting cassandra.service - LSB: distributed storage system for structured data...
Sep 12 09:28:58 The-Hive systemd[1]: Started cassandra.service - LSB: distributed storage system for structured data.
root@The-Hive:~# sudo chown -R cassandra:cassandra /var/lib/cassandra
root@The-Hive:~#
```

Once installed, configure it properly before starting TheHive.

Install Elasticsearch

Next, install Elasticsearch.

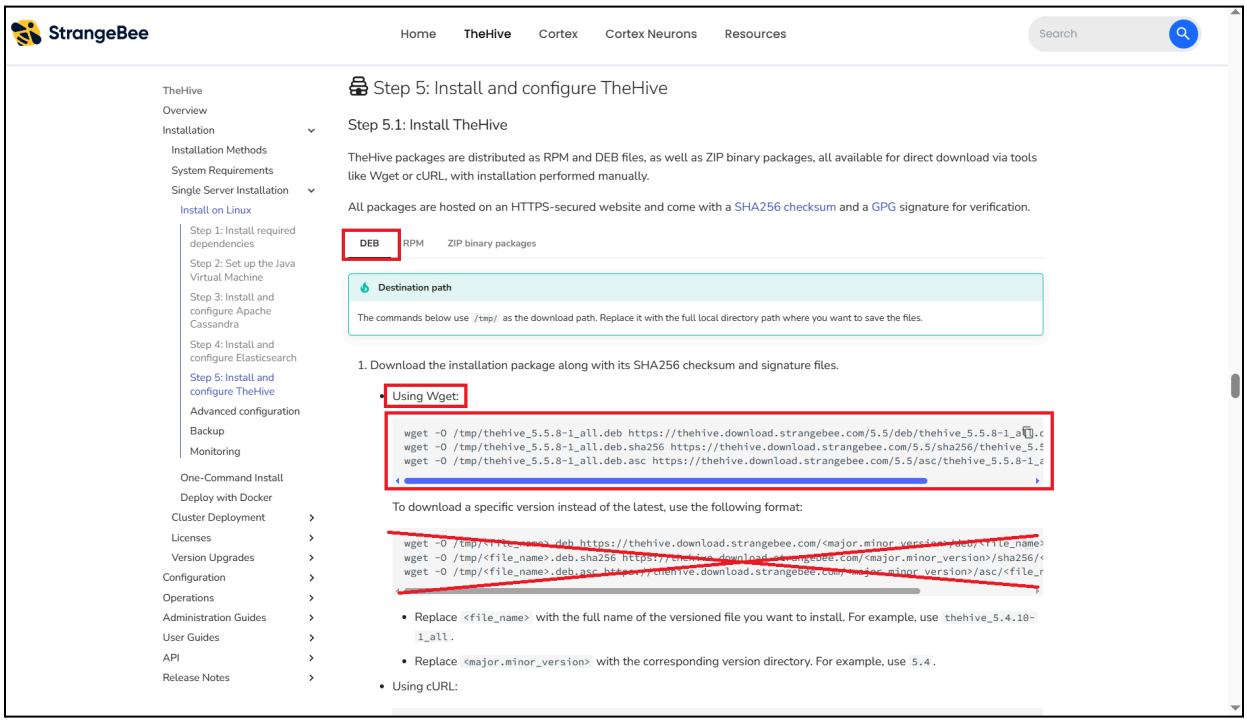



```
root@The-Hive:~#
root@The-Hive:~# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |  sudo gpg --dearmor -o /usr/share/keyrings/elasticsearch-keyring.gpg
sudo apt-get install apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.8.3).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@The-Hive:~# echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" |  sudo tee /etc/apt/sources.list.d/elastic-7.x.list
deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main
root@The-Hive:~# sudo apt update
sudo apt install elasticsearch
Hit:1 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:2 https://apt.corretto.aws stable InRelease
Get:3 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Hit:4 http://mirrors.digitalocean.com/ubuntu noble InRelease
Hit:5 http://mirrors.digitalocean.com/ubuntu noble-updates InRelease
Get:6 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 Packages [145 kB]
Hit:7 http://mirrors.digitalocean.com/ubuntu noble-backports InRelease
Hit:8 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:9 https://apache.jfrog.io/artifactory/cassandra-deb 41x InRelease
Fetched 158 kB in 9s (18.5 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 325 MB of archives.
After this operation, 542 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 elasticsearch amd64 7.17.29 [325 MB]
Fetched 325 MB in 5s (66.9 MB/s)
Selecting previously unselected package elasticsearch.
(Reading database ... 110493 files and directories currently installed.)
Preparing to unpack .../elasticsearch_7.17.29_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (7.17.29) ...
Setting up elasticsearch (7.17.29) ...
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
 sudo systemctl daemon-reload
 sudo systemctl enable elasticsearch.service
### You can start elasticsearch service by executing
 sudo systemctl start elasticsearch.service
warning: usage of JAVA_HOME is deprecated, use ES_JAVA_HOME
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@The-Hive:~#
```

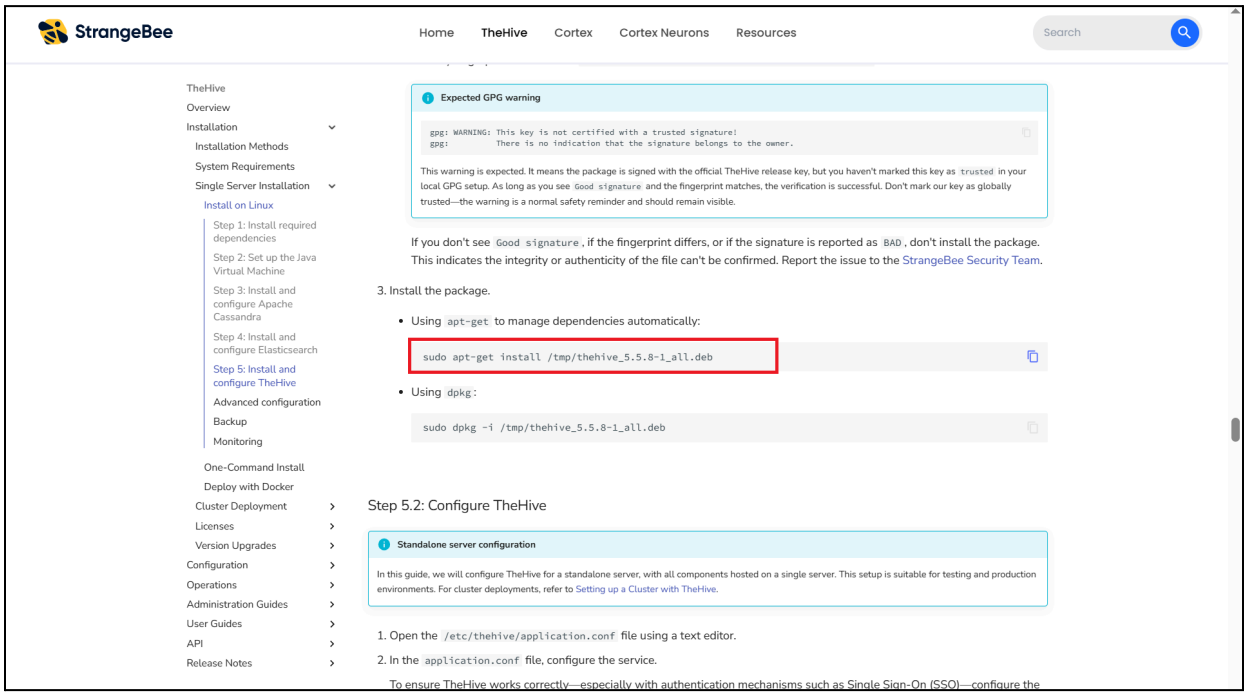
Install TheHive

Use **`wget`** (recommended) to download and install TheHive.

👉 If you prefer **`curl`** or another method, scroll down in the StrangeBee docs and copy the relevant command:




```
root@The-Hive:~# wget -O /tmp/thehive_5.5.8-1_all.deb https://thehive.download.strangebee.com/5.5/deb/thehive_5.5.8-1_all.deb
wget -O /tmp/thehive_5.5.8-1_all.deb.sha256 https://thehive.download.strangebee.com/5.5/sha256/thehive_5.5.8-1_all.deb.sha256
wget -O /tmp/thehive_5.5.8-1_all.deb.asc https://thehive.download.strangebee.com/5.5/asc/thehive_5.5.8-1_all.deb.asc
--2025-09-12 09:47:32--  https://thehive.download.strangebee.com/5.5/deb/thehive_5.5.8-1_all.deb
Resolving thehive.download.strangebee.com (thehive.download.strangebee.com)... 3.164.68.11, 3.164.68.103, 3.164.68.79, ...
Connecting to thehive.download.strangebee.com (thehive.download.strangebee.com)|3.164.68.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 472391872 (451M) [application/vnd.debian.binary-package]
Saving to: '/tmp/thehive_5.5.8-1_all.deb'

/tmp/thehive_5.5.8-1_all.deb          100%[=====================================================================>] 450.51M  6.88MB/s    in 82s

2025-09-12 09:48:56 (5.51 MB/s) - '/tmp/thehive_5.5.8-1_all.deb' saved [472391872/472391872]

--2025-09-12 09:48:56--  https://thehive.download.strangebee.com/5.5/sha256/thehive_5.5.8-1_all.deb.sha256
Resolving thehive.download.strangebee.com (thehive.download.strangebee.com)... 3.164.68.11, 3.164.68.124, 3.164.68.103, ...
Connecting to thehive.download.strangebee.com (thehive.download.strangebee.com)|3.164.68.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90 [binary/octet-stream]
Saving to: '/tmp/thehive_5.5.8-1_all.deb.sha256'

/tmp/thehive_5.5.8-1_all.deb.sha256    100%[=====================================================================>]      90  --.-KB/s    in 0s

2025-09-12 09:48:57 (49.2 MB/s) - '/tmp/thehive_5.5.8-1_all.deb.sha256' saved [90/90]

--2025-09-12 09:48:57--  https://thehive.download.strangebee.com/5.5/asc/thehive_5.5.8-1_all.deb.asc
Resolving thehive.download.strangebee.com (thehive.download.strangebee.com)... 3.164.68.103, 3.164.68.124, 3.164.68.11, ...
Connecting to thehive.download.strangebee.com (thehive.download.strangebee.com)|3.164.68.103|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 833 [application/pgp-keys]
Saving to: '/tmp/thehive_5.5.8-1_all.deb.asc'

/tmp/thehive_5.5.8-1_all.deb.asc       100%[=====================================================================>]     833  --.-KB/s    in 0s

2025-09-12 09:48:58 (38.2 MB/s) - '/tmp/thehive_5.5.8-1_all.deb.asc' saved [833/833]

root@The-Hive:~#
```




```
root@The-Hive:~# sudo apt-get install /tmp/thehive_5.5.8-1_all.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'thehive' instead of '/tmp/thehive_5.5.8-1_all.deb'
The following NEW packages will be installed:
  thehive
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 0 B/472 MB of archives.
After this operation, 472 MB of additional disk space will be used.
Get:1 /tmp/thehive_5.5.8-1_all.deb thehive all 5.5.8-1 [472 MB]
Selecting previously unselected package thehive.
(Reading database ... 111591 files and directories currently installed.)
Preparing to unpack /tmp/thehive_5.5.8-1_all.deb ...
Unpacking thehive (5.5.8-1) ...
Setting up thehive (5.5.8-1) ...
Creating system group: thehive
Creating system user: thehive in thehive with thehive daemon-user and shell /bin/false
1+0 records in
1+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 5.9664e-05 s, 17.2 MB/s
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@The-Hive:~#
```

Configure Cassandra

Open the Cassandra configuration file: `nano /etc/cassandra/cassandra.yaml`

If `cassandra.yaml` is missing, Cassandra likely didn't install correctly. Reinstall it before proceeding.

Edit the following keys (use **CTRL+W** in nano to search each key):

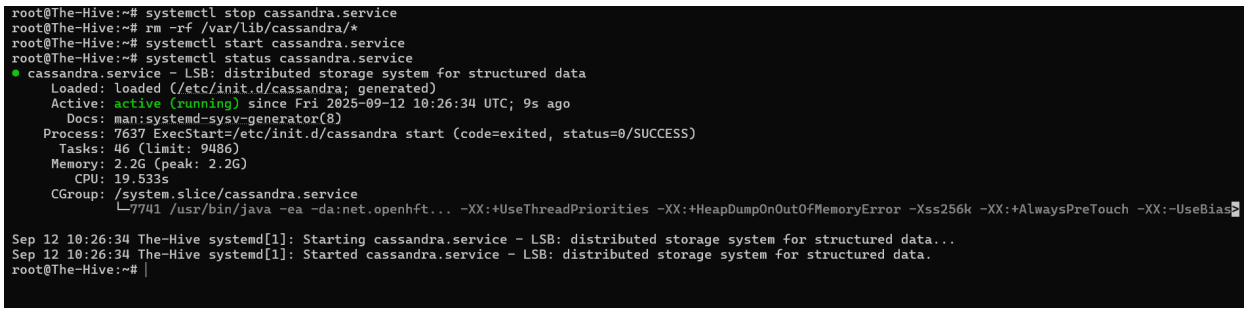
```
cluster_name: 'soc-workflow'
listen_address: {thehive-vm-ip}
rpc_address: {thehive-vm-ip}
- seeds: "{thehive-vm-ip}:7000"
```

Save and exit: **CTRL+X** → **Y** → **Enter**

Restart Cassandra cleanly:

```
sudo systemctl stop cassandra.service
sudo rm -rf /var/lib/cassandra/*
sudo systemctl start cassandra.service
sudo systemctl status cassandra.service
```

✅ If the status shows **active (running)**, Cassandra is ready.



Configure Elasticsearch

Open the configuration file: `nano /etc/elasticsearch/elasticsearch.yml`

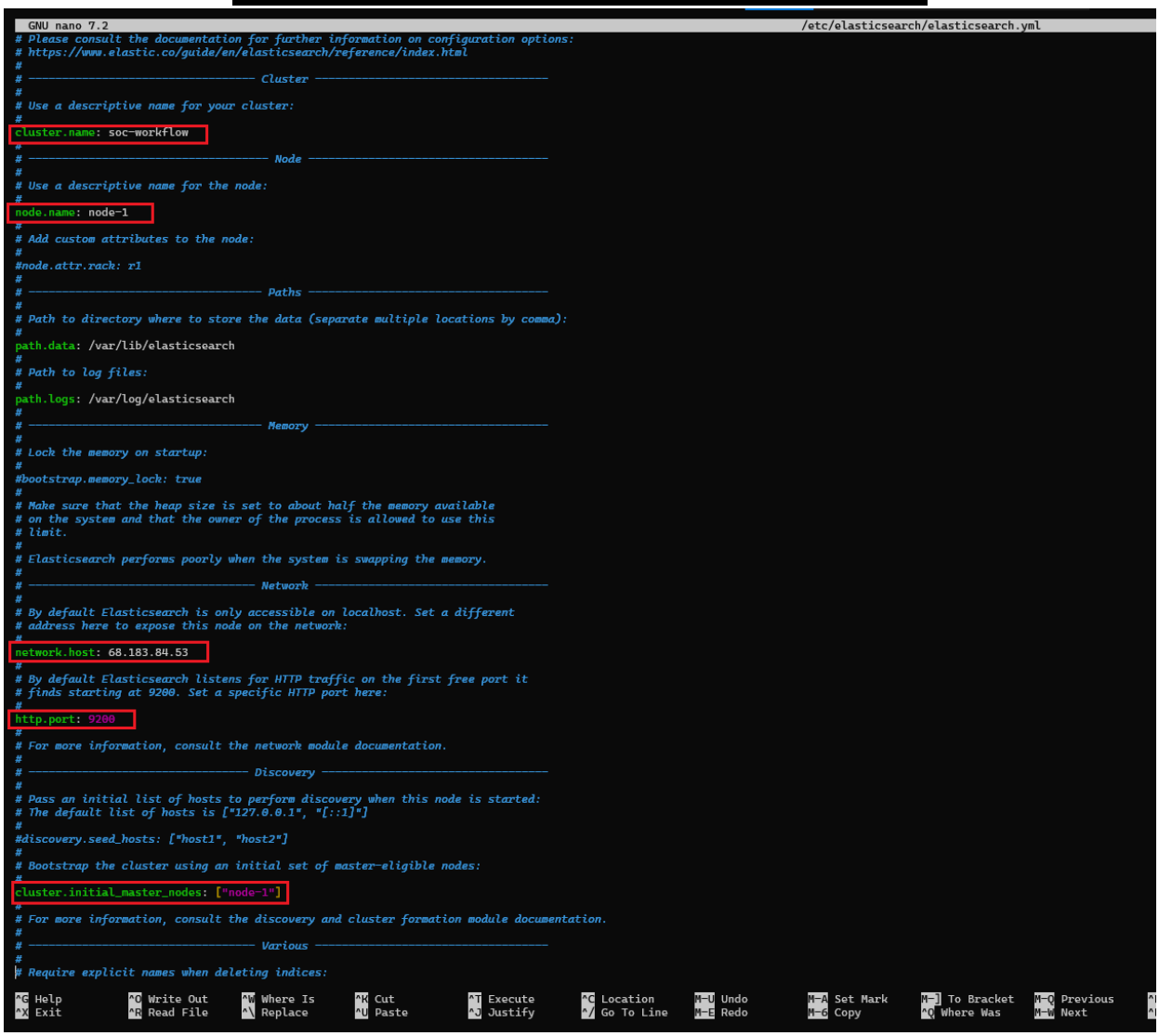
⚠️ If `elasticsearch.yml` is missing, Elasticsearch likely didn't install correctly. Reinstall it using the previous steps.

Then open the configuration file again: `nano /etc/elasticsearch/elasticsearch.yml`

Update the following keys (use **CTRL+W** in nano to search each key):

```
cluster.name: soc-workflow
```

```
node.name: node-1

network.host: {hive-vm-ip}

http.port: 9200

cluster.initial_master_nodes: ["node-1"]
```



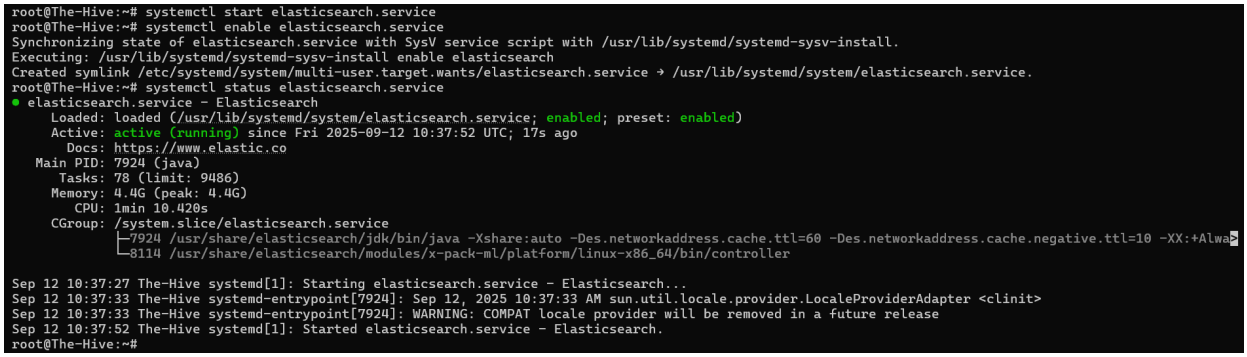
Save and exit: **CTRL+X → Y → Enter**

Restart and enable Elasticsearch:

```
sudo systemctl start elasticsearch.service
```

```
sudo systemctl enable elasticsearch.service
sudo systemctl status elasticsearch.service
```

✅ If the status shows **active (running)**, Elasticsearch is ready.

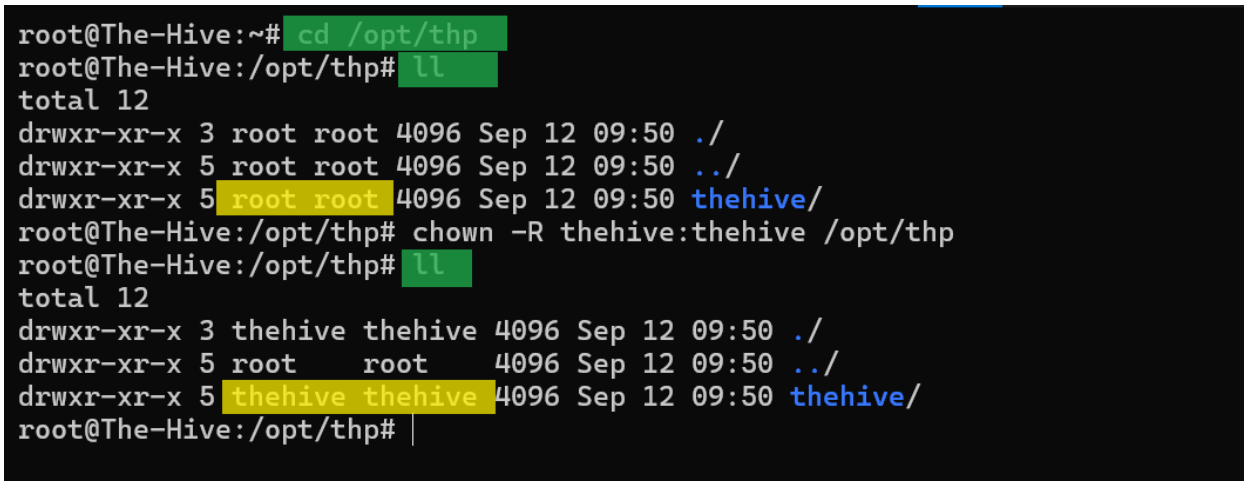


Configure TheHive

Before configuring TheHive, update file permissions for its directory:

```
cd /opt/thp
ls -l
sudo chown -R thehive:thehive /opt/thp
ls -l
```

✅ This ensures TheHive has proper access to its files.

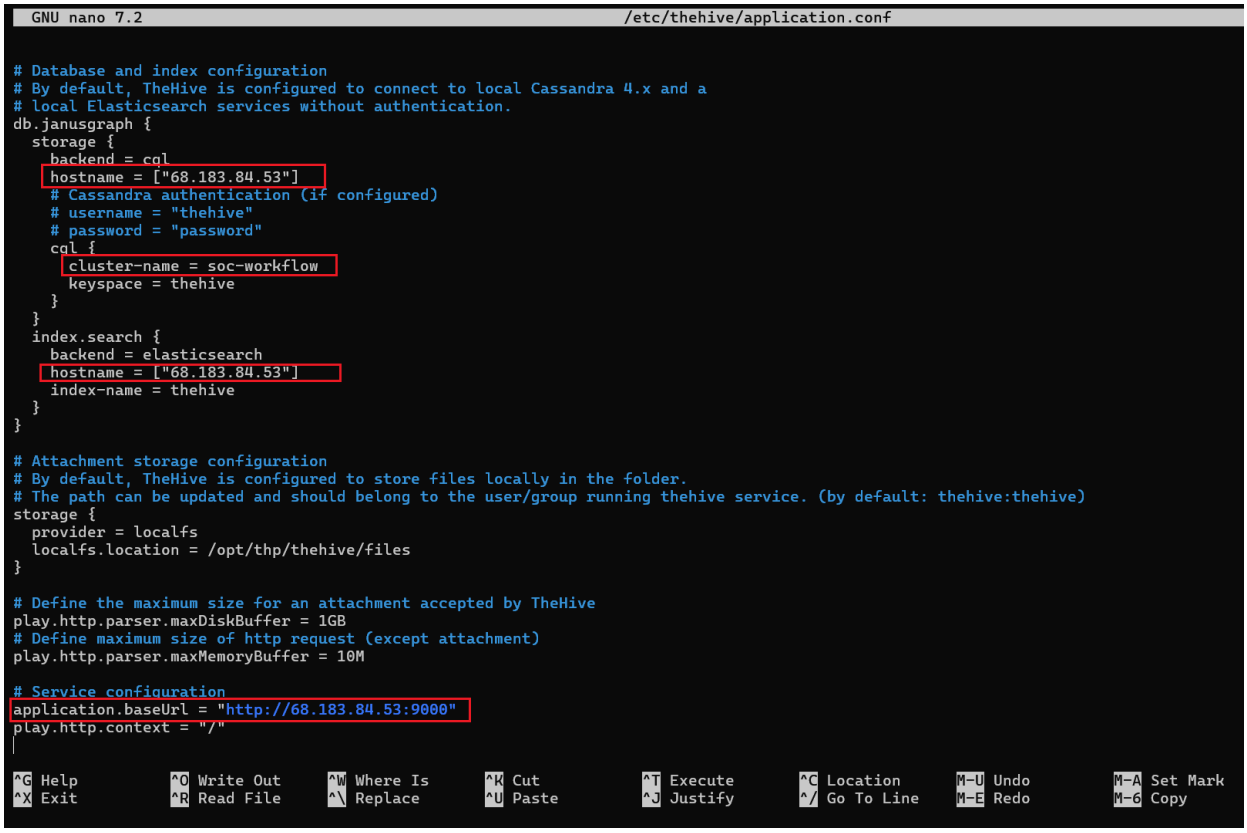


Open the configuration file: `sudo nano /etc/thehive/application.conf`

⚠️ If `application.conf` is missing, TheHive likely didn't install correctly. Reinstall it before proceeding.

Edit the following keys (use **CTRL+W** in nano to search each one):

```
hostname = ["{thehive-vm-ip}"]

cluster-name = soc-workflow

backend = elasticsearch
    hostname = ["{thehive-vm-ip}"]

application.baseUrl = "http://{thehive-vm-ip}:9000"
```

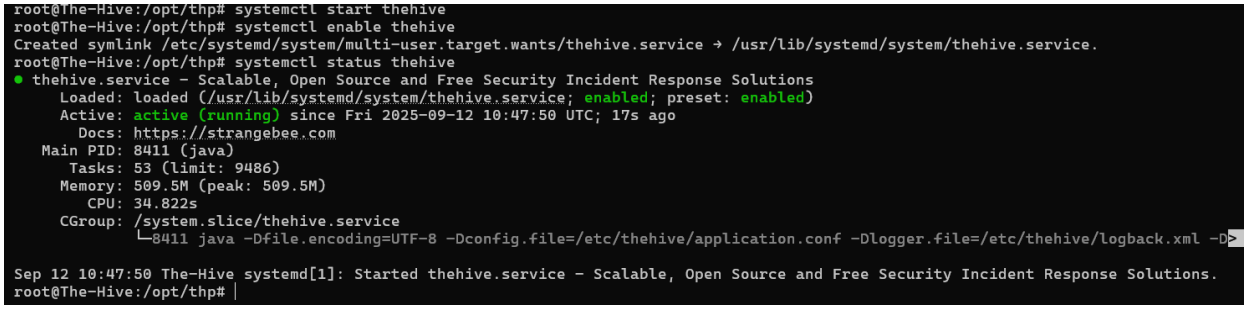


Save and exit: **CTRL+X → Y → Enter**

Restart and enable TheHive

```
sudo systemctl start thehive.service
sudo systemctl enable thehive.service
sudo systemctl status thehive.service
```

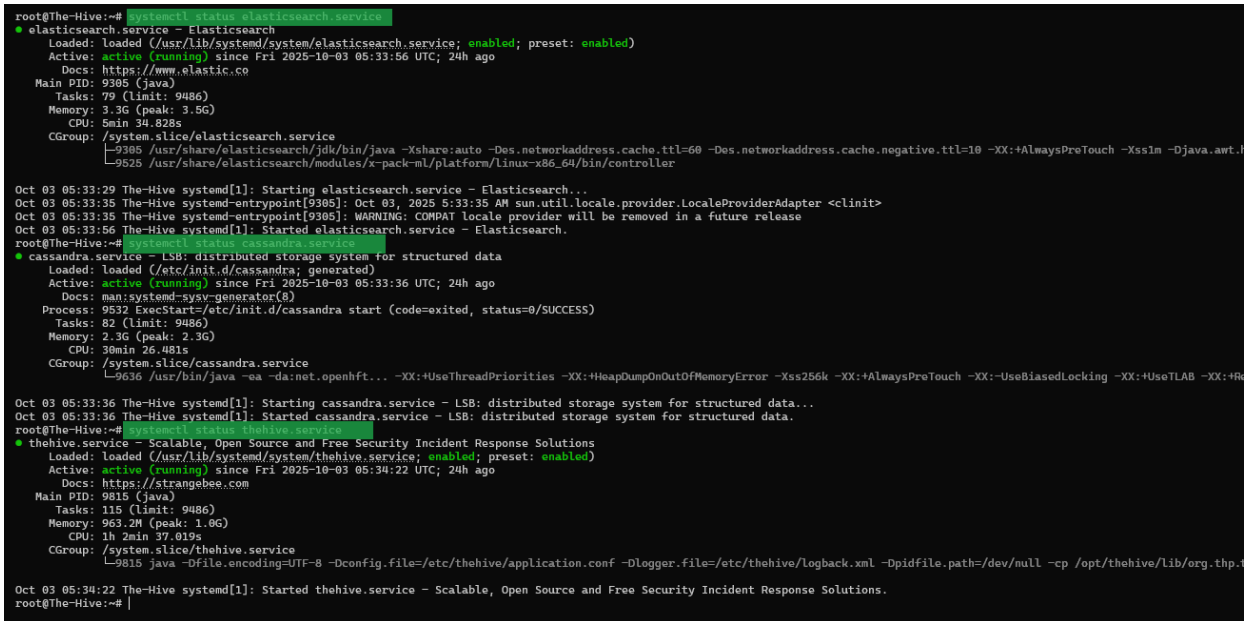
✅ If the status shows **active (running)**, TheHive is ready.



Verify all required services are running

```
sudo systemctl status cassandra.service
sudo systemctl status elasticsearch.service
sudo systemctl status thehive.service
```

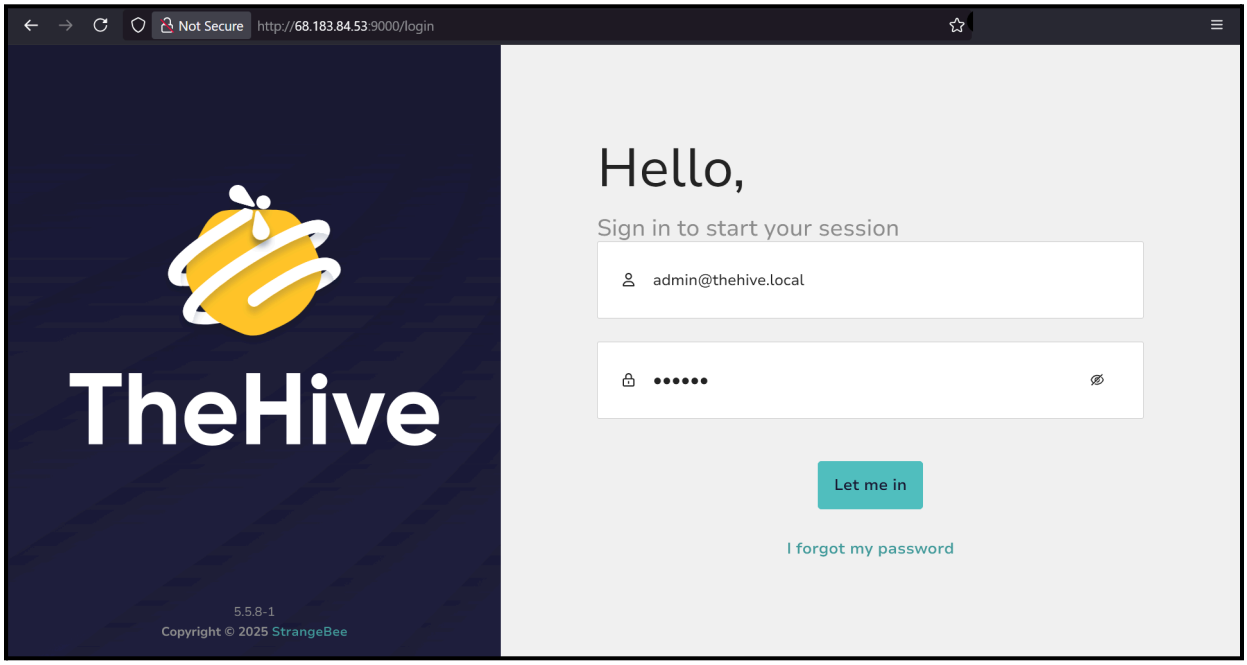
All three must show **active (running)** before you proceed.



Once all services are running, open your browser and go to:

http://{your-hive-vm-ip}:9000

You should now see **TheHive login page** 🎉



If the page doesn't load, your firewall might be blocking port 9000.
Allow it using: `sudo ufw allow 9000`

To find the **latest default credentials** or learn how to perform the initial admin setup, refer to the official StrangeBee documentation:

👉 Perform Initial Setup as Admin — TheHive Docs

This ensures you always use the **most up-to-date credentials and security setup** recommended by TheHive team.

Integrate Windows 11 VM with Wazuh

Now that both **Wazuh** and **TheHive** servers are running on your DigitalOcean VMs, you can connect your **Windows 11 VM (in VMware)** as a Wazuh agent.

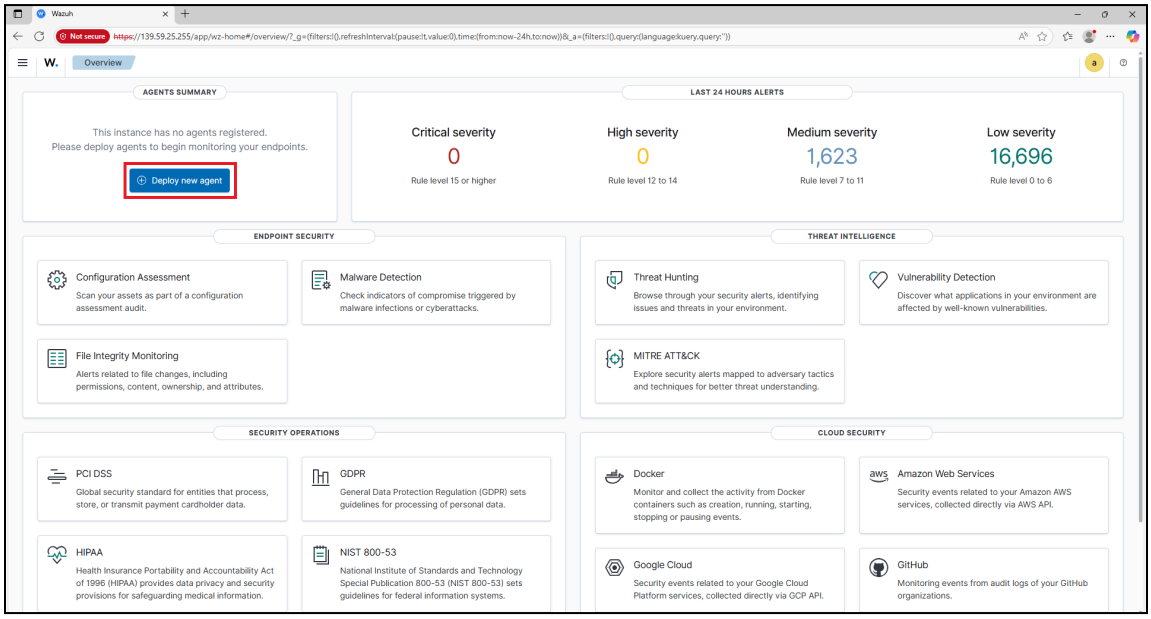
Access the Wazuh Dashboard

Inside your Windows 11 VM:

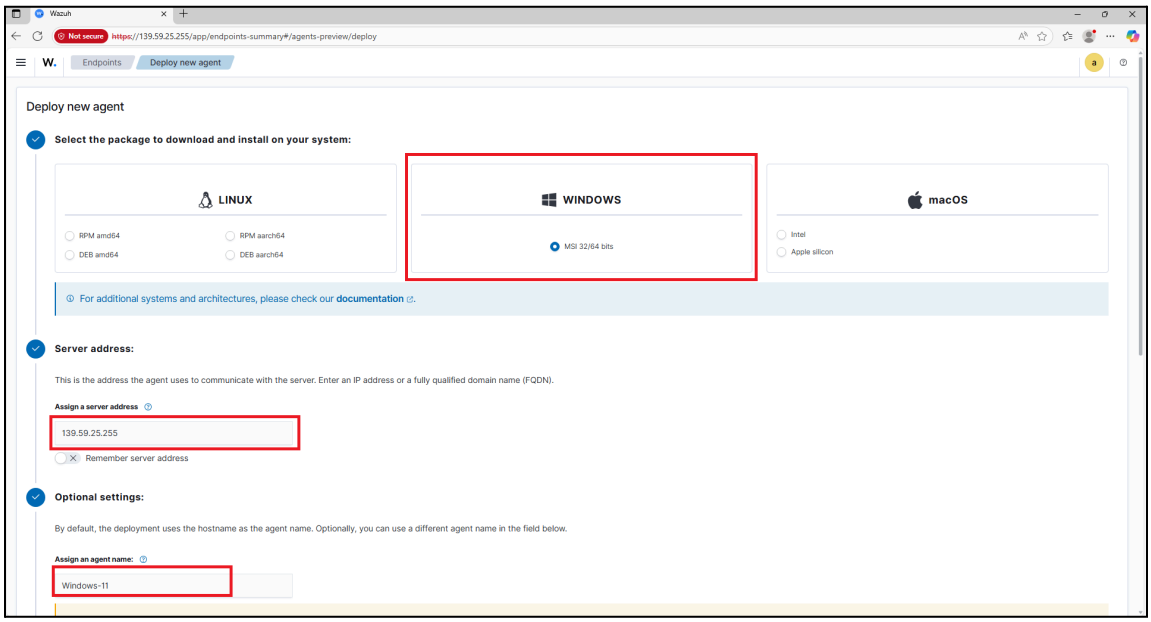
1. Open your browser and go to **https://{your-wazuh-vm-ip}**
2. Log in using your Wazuh dashboard credentials.

Add a New Agent

1. In the Wazuh dashboard, click **“Deploy new agent.”**



2. Select **Windows** as the agent type.
3. In **Server address**, enter your Wazuh server IP.
4. Assign a friendly **Agent name** such as **`Windows-11`.**

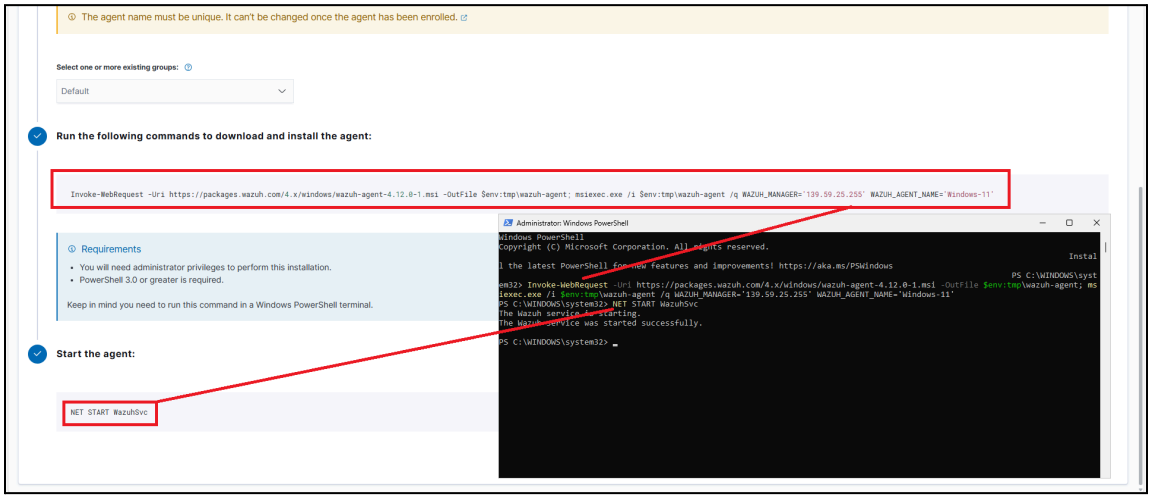


5. Copy the **PowerShell commands** shown on the screen.

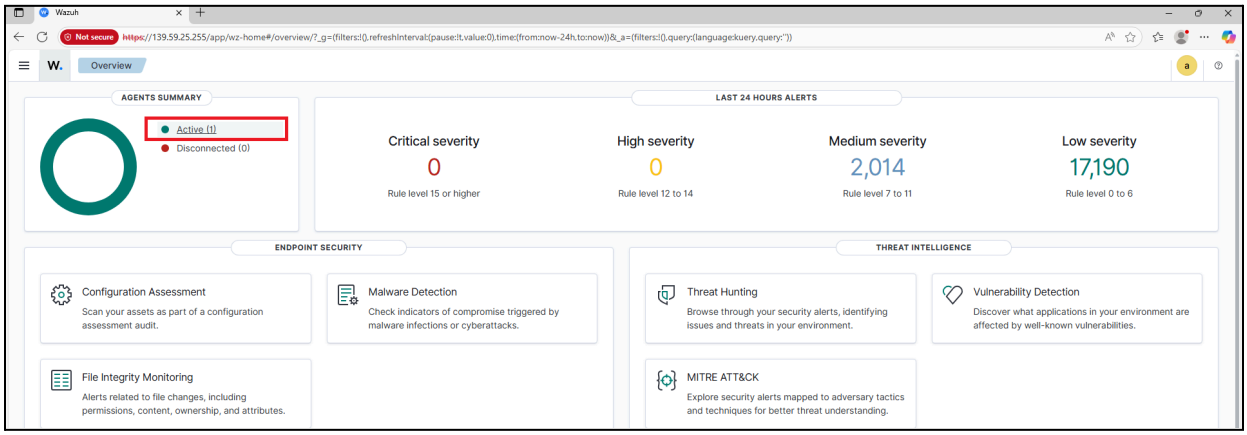
Install the Agent on Windows

1. Open **PowerShell as Administrator** in your Windows VM.
2. Paste and run the copied commands.

3. Return to the Wazuh dashboard and refresh the **Agents Summary** page.



You should now see **`Active (1)`** under agents.



If the agent does not appear active:

On your Wazuh VM, allow the following ports:

```
sudo ufw allow 1514
sudo ufw allow 1515
```

On your Windows VM, restart the Wazuh service: `Restart-Service -Name "wazuh"`
Refresh the dashboard again. The agent should now appear as active.

Configure Sysmon Telemetry in Wazuh

To enhance visibility, integrate **Sysmon** logs with your Windows Wazuh agent.

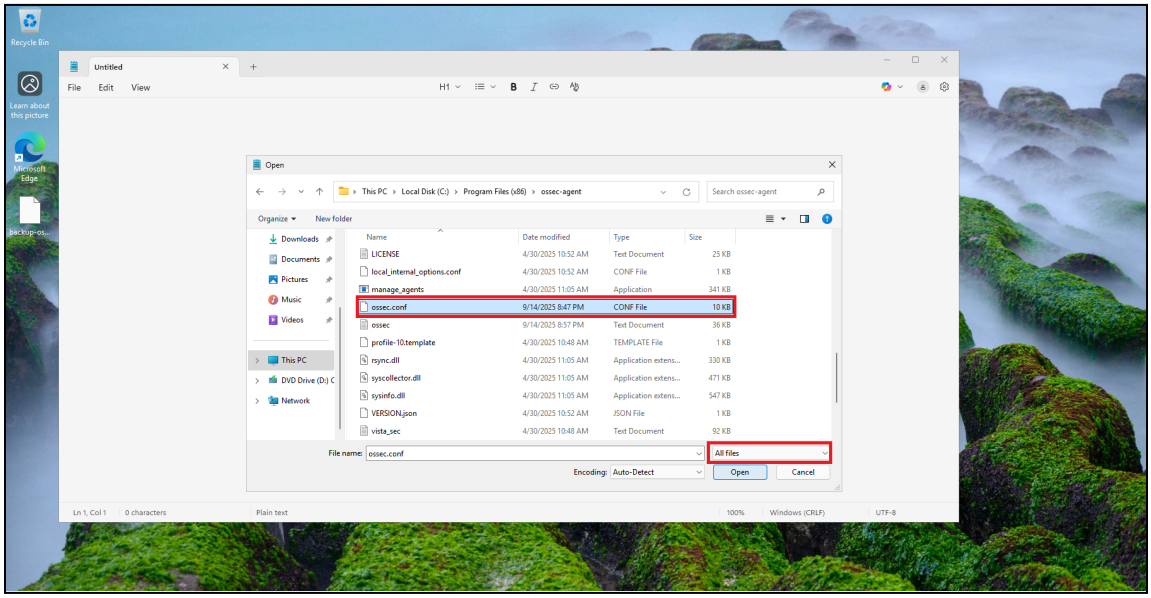
Backup Configuration

Make a backup of your current Wazuh agent config file: **C:\Program Files (x86)\ossec-agent\ossec.conf**

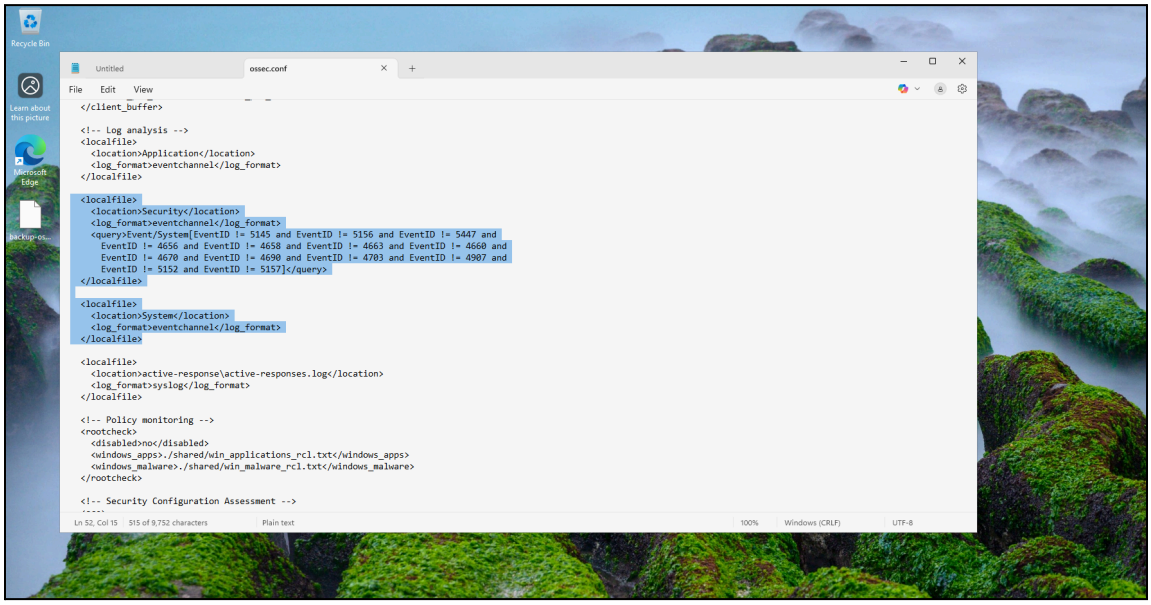
Copy it to another location (for example, your Desktop).

Edit Configuration

1. Open **Notepad as Administrator**.
2. Press **`Ctrl + O`** and navigate to: **C:\Program Files (x86)\ossec-agent**
3. Open **`ossec.conf`**.

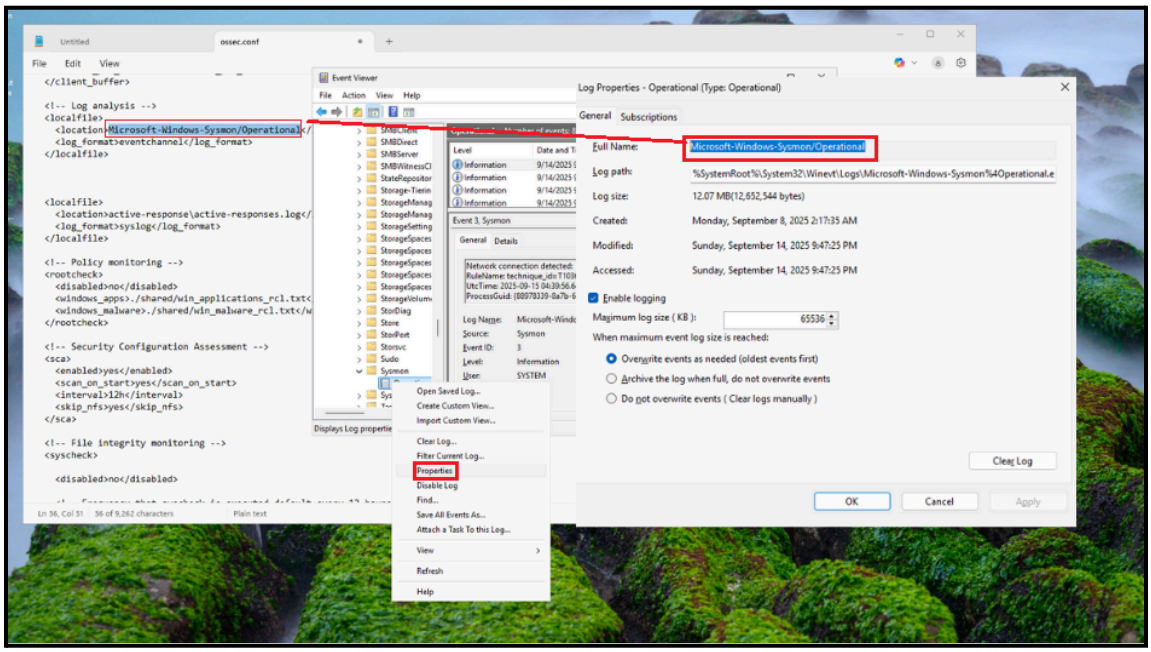


4. Locate and remove the highlighted default log entries.

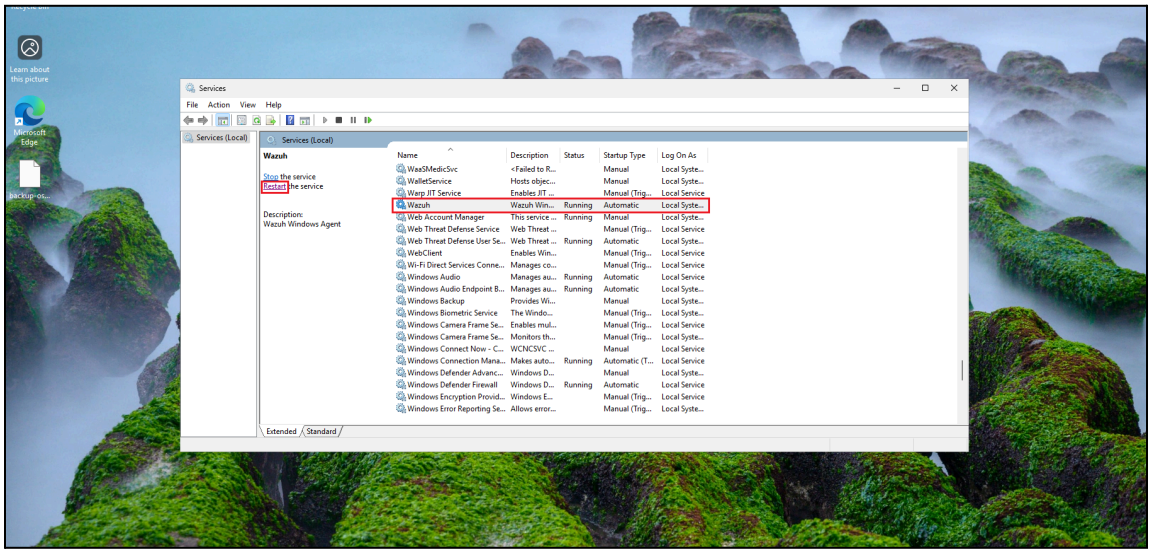


Add Sysmon Logs

1. Open **Event Viewer** → **`Applications and Services Logs`** → **`Microsoft`** → **`Windows`** → **Sysmon** → **`Operational`**.
2. Right-click **Operational** → **Properties** → copy the **Full Name**.
3. Paste this log path inside your **`ossec.conf`** file under the **`<localfile>`** section.

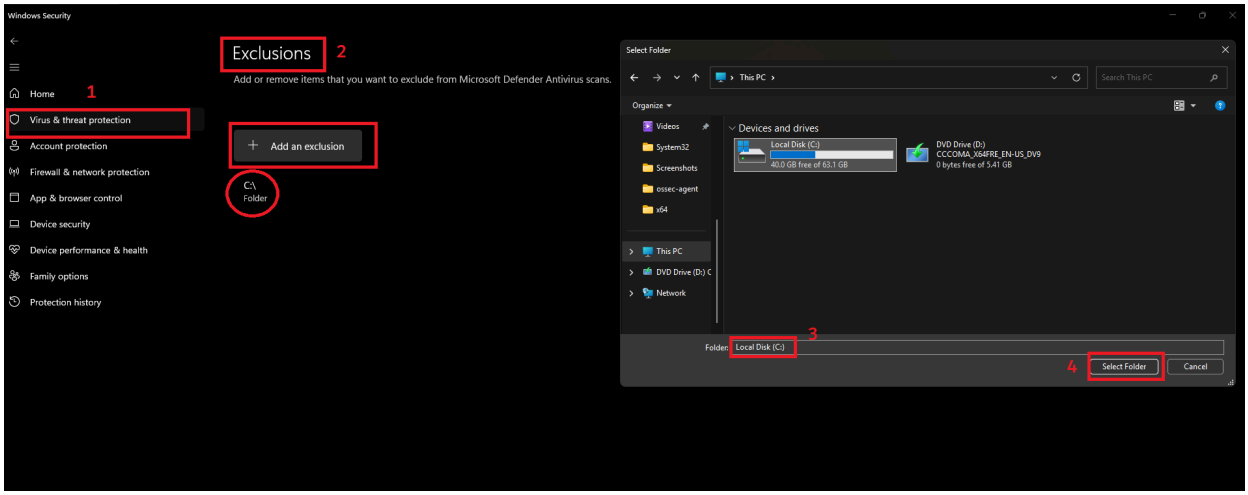


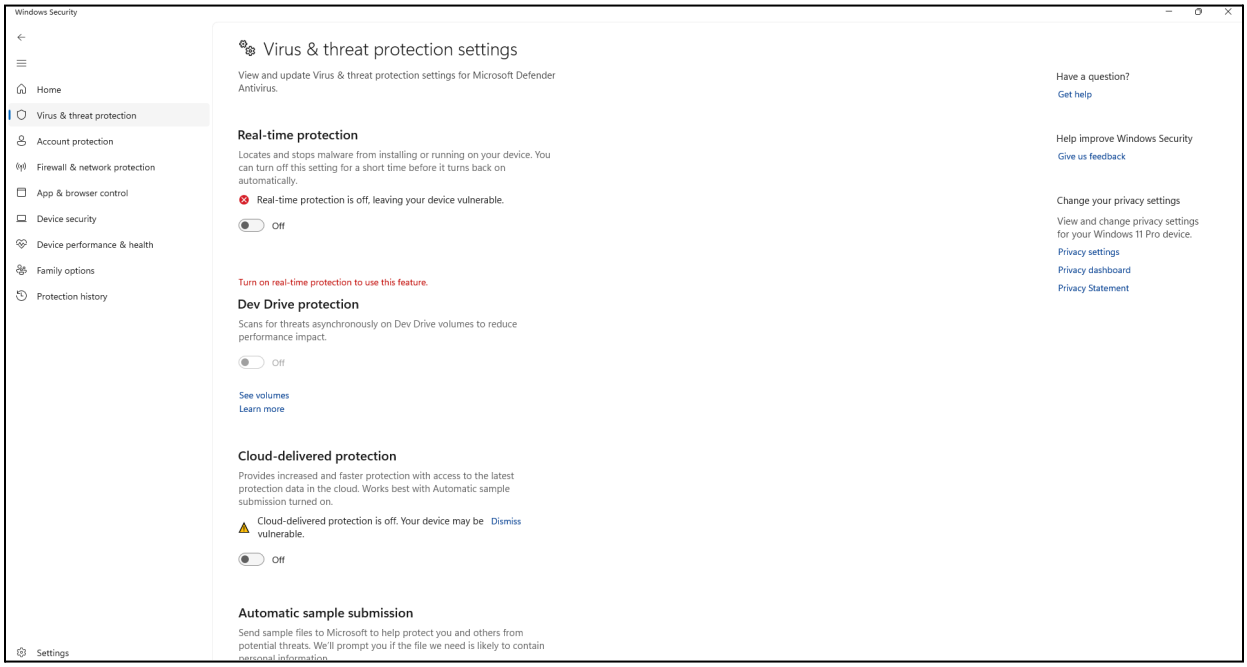
4. Save the file and restart the Wazuh service



Disable Windows Defender (for Lab Purposes Only)

1. Open **Windows Security** → **Virus & threat protection**.
2. Scroll down to **Exclusions** → **Add or remove exclusions**.
3. Click **Add an exclusion** → **Folder** → select **C:**



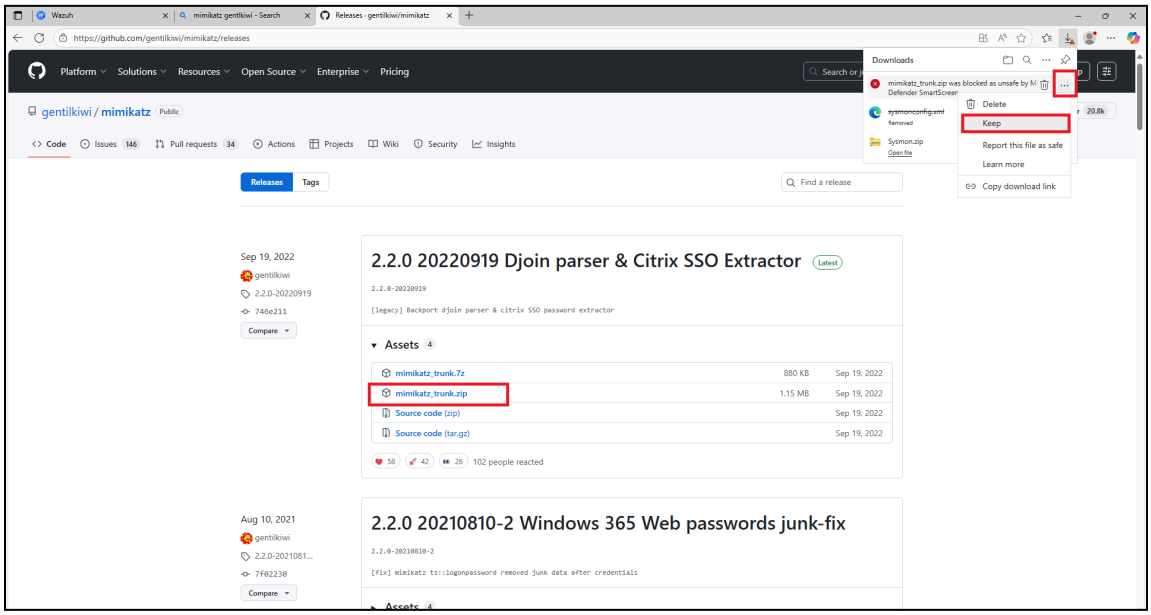


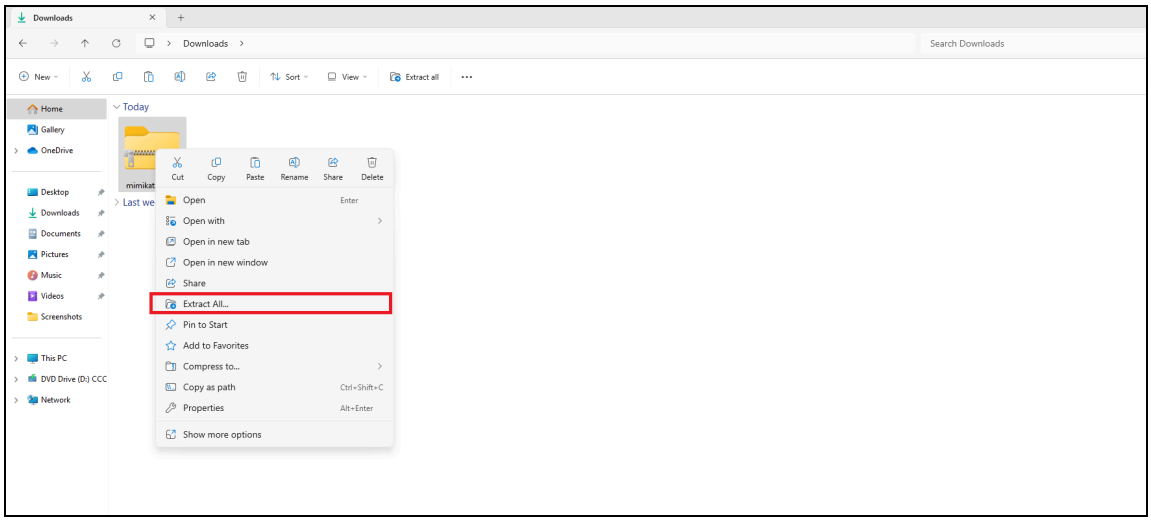
You have now successfully connected your **Windows 11 VM** to **Wazuh** and enabled **Sysmon telemetry**.

Install Mimikatz

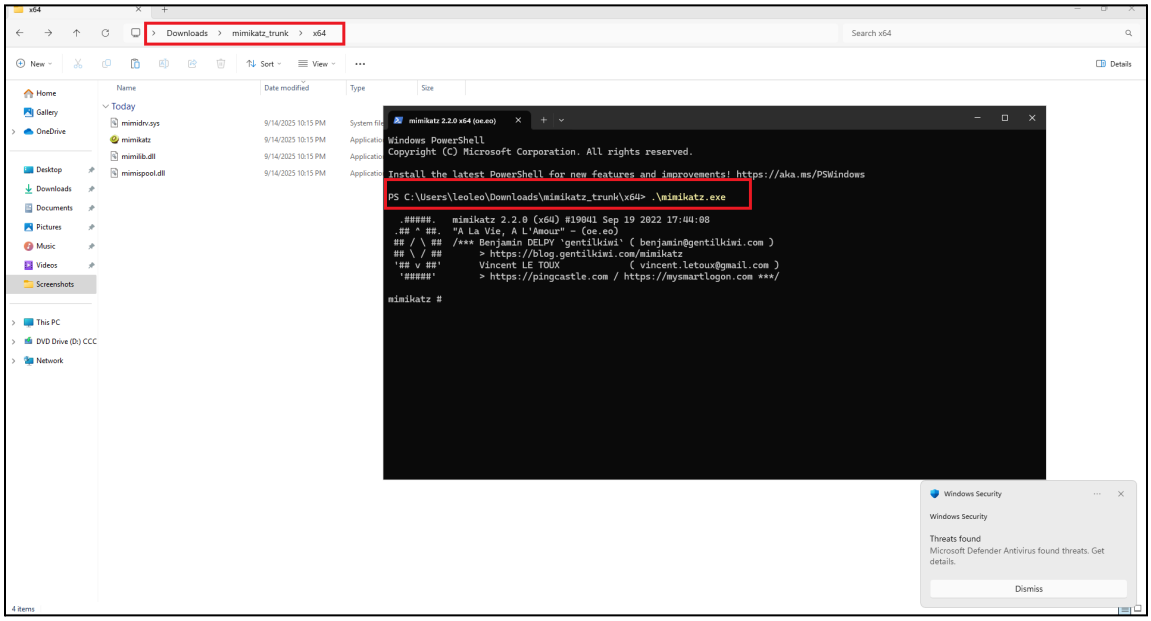
1. Visit the official Mimikatz GitHub:
    a. 👉 https://github.com/gentilkiwi/mimikatz/releases

2. Download and extract the latest release.





3. Open **Command Prompt or PowerShell** inside the **x64** folder.



Now Mimikatz is ready for use in your test environment.

Configure Wazuh to Detect Mimikatz Activity

In your **Wazuh VM**, open and verify the logging configuration: `nano /var/ossec/etc/ossec.conf`

Ensure these lines are set to:

```
<logall>yes</logall>
<logall_json>yes</logall_json>
```

```
  GNU nano 7.2                                    /var/ossec/etc/ossec.conf *
<!--
  Wazuh - Manager - Default configuration for ubuntu 24.04
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
  <logging>
    <log_format>plain</log_format>

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy
```

Save the file and restart Wazuh: `systemctl restart wazuh-manager.service`

Check **`/var/ossec/logs/archives/`** logs should now be generated there.

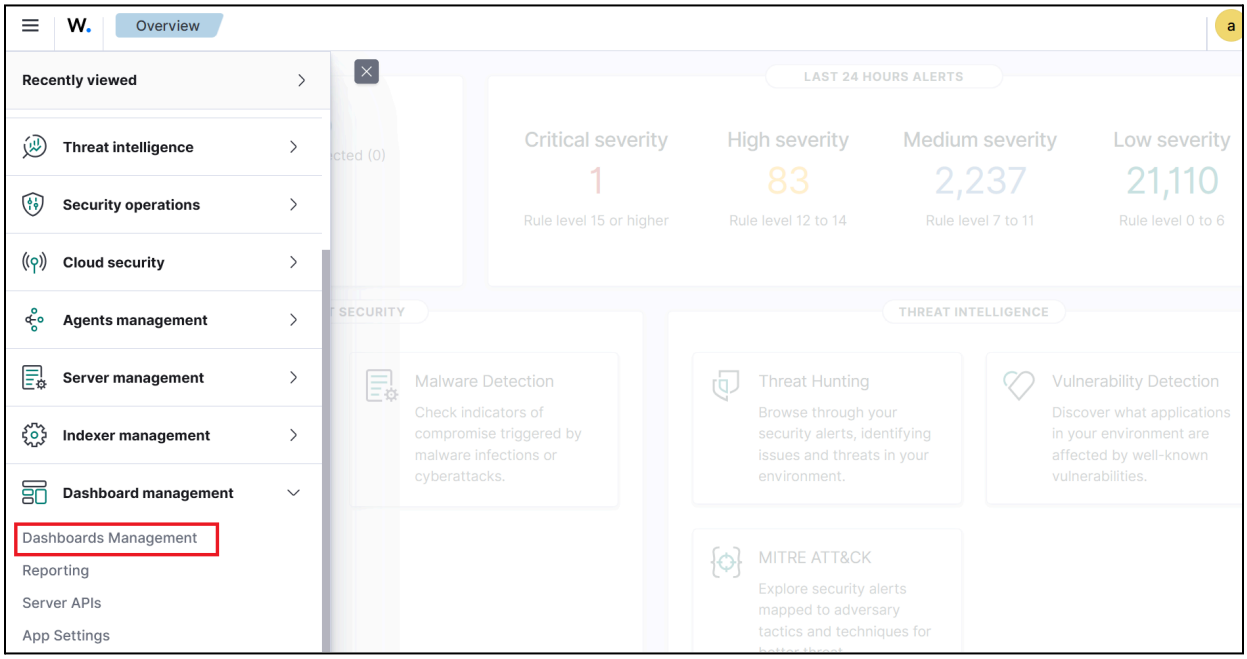
```
root@Wazuh:~# systemctl restart wazuh-manager.service
root@Wazuh:~# cd /var/ossec/logs/archives/
root@Wazuh:/var/ossec/logs/archives# ls -la
total 908
drwxr-x--- 3 wazuh wazuh   4096 Sep 15 05:50 .
drwxrwx--- 8 wazuh wazuh   4096 Sep 15 00:00 ..
drwxr-x--- 3 wazuh wazuh   4096 Sep  8 09:59 2025
-rw-r----- 2 wazuh wazuh 644000 Sep 15 05:51 archives.json
-rw-r----- 2 wazuh wazuh 269751 Sep 15 05:51 archives.log
root@Wazuh:/var/ossec/logs/archives# cat archives.log | head
2025 Sep 15 05:50:57 Wazuh->rootcheck Starting rootcheck scan.
2025 Sep 15 05:51:00 (Windows-11) any->EventChannel {"win":{"system":{"providerName":"Microsoft-Windows-Sysmon","providerGuid":"{5770
385f-c22a-43e0-bf4c-06f5698ffbd9}","eventID":"11","version":"2","level":"4","task":"11","opcode":"0","keywords":"0x8000000000000000",
"systemTime":"2025-09-15T05:50:38.2936770Z","eventRecordID":"34842","processID":"4080","threadID":"4332","channel":"Microsoft-Windows
-Sysmon/Operational","computer":"leo","severityValue":"INFORMATION","message":"\"File created:\r\nRuleName: technique_id=T1574.010,te
chnique_name=Services File Permissions Weakness\r\nUtcTime: 2025-09-15 05:50:38.292\r\nProcessGuid: {88978339-a92a-68c7-2405-00000000
0800}\r\nProcessId: 6660\r\nImage: C:\\Windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe\r\nTargetFilename: C:\\Windows\\
assembly\\NativeImages_v4.0.30319_64\\Temp\\1a04-0\\Microsoft.PowerShell.Management.Activities.dll\r\nCreationUtcTime: 2025-09-15 05:
50:38.292\r\nUser: NT AUTHORITY\\SYSTEM\""},"eventdata":{"ruleName":"technique_id=T1574.010,technique_name=Services File Permissions
Weakness","utcTime":"2025-09-15 05:50:38.292","processGuid":"{88978339-a92a-68c7-2405-000000000800}","processId":"6660","image":"C:\\
\\Windows\\\\Microsoft.NET\\\\Framework64\\\\v4.0.30319\\\\mscorsvw.exe","targetFilename":"C:\\\\Windows\\\\assembly\\\\NativeImages_
v4.0.30319_64\\\\Temp\\\\1a04-0\\\\Microsoft.PowerShell.Management.Activities.dll","creationUtcTime":"2025-09-15 05:50:38.292","user"
:"NT AUTHORITY\\\\SYSTEM"}}}
2025 Sep 15 05:51:00 (Windows-11) any->EventChannel {"win":{"system":{"providerName":"Microsoft-Windows-Sysmon","providerGuid":"{5770
385f-c22a-43e0-bf4c-06f5698ffbd9}","eventID":"11","version":"2","level":"4","task":"11","opcode":"0","keywords":"0x8000000000000000",
"systemTime":"2025-09-15T05:50:38.3345454Z","eventRecordID":"34843","processID":"4080","threadID":"4332","channel":"Microsoft-Windows
-Sysmon/Operational","computer":"leo","severityValue":"INFORMATION","message":"\"File created:\r\nRuleName: technique_id=T1574.010,te
chnique_name=Services File Permissions Weakness\r\nUtcTime: 2025-09-15 05:50:38.332\r\nProcessGuid: {88978339-a92a-68c7-2405-00000000
0800}\r\nProcessId: 6660\r\nImage: C:\\Windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe\r\nTargetFilename: C:\\Windows\\
assembly\\NativeImages_v4.0.30319_64\\ngenlock.dat\r\nCreationUtcTime: 2025-09-15 05:50:24.930\r\nUser: NT AUTHORITY\\SYSTEM\""},"eve
ntdata":{"ruleName":"technique_id=T1574.010,technique_name=Services File Permissions Weakness","utcTime":"2025-09-15 05:50:38.332","p
rocessGuid":"{88978339-a92a-68c7-2405-000000000800}","processId":"6660","image":"C:\\\\Windows\\\\Microsoft.NET\\\\Framework64\\\\v4.
0.30319\\\\mscorsvw.exe","targetFilename":"C:\\\\Windows\\\\assembly\\\\NativeImages_v4.0.30319_64\\\\ngenlock.dat","creationUtcTime"
```

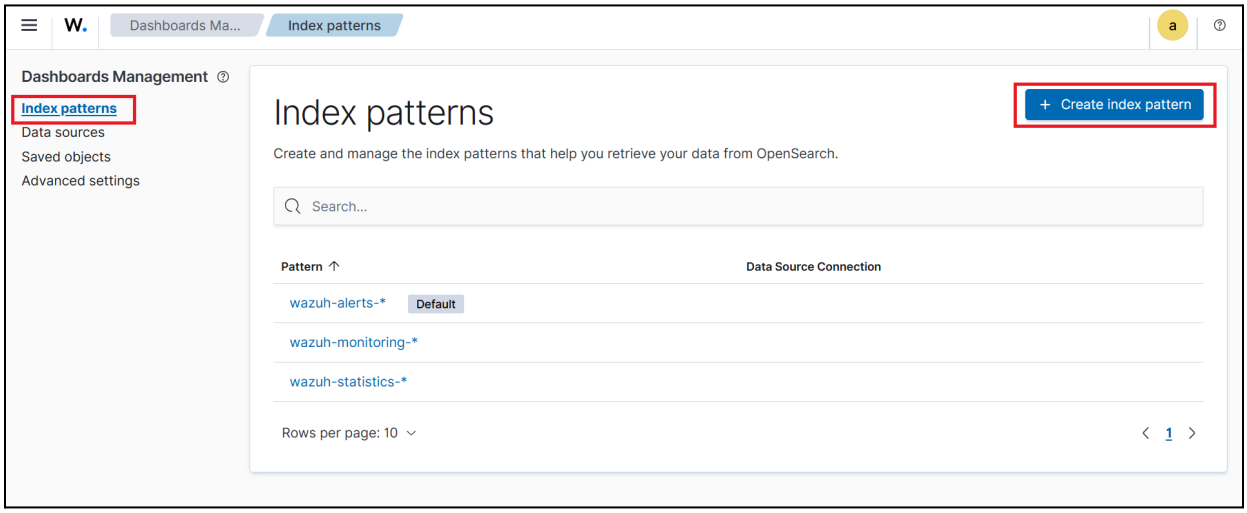
Adjust Filebeat Configuration: `nano /etc/filebeat/filebeat.yml`

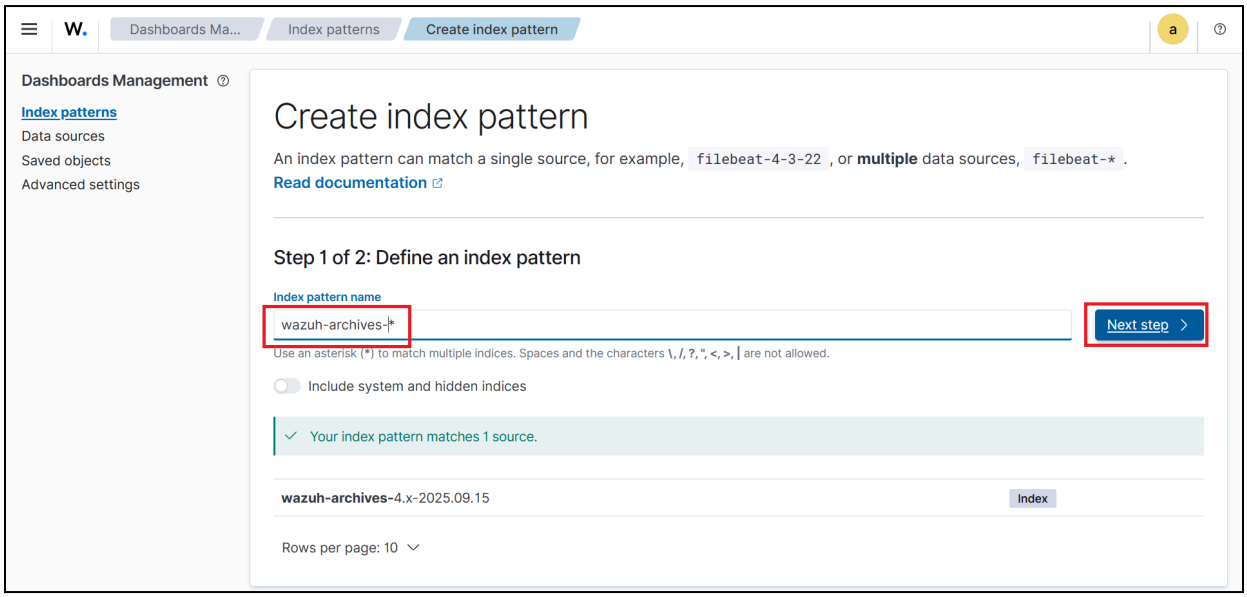
```
 GNU nano 7.2                                  /etc/filebeat/filebeat.yml *
# Wazuh - Filebeat configuration file
output.elasticsearch.hosts:
        - 127.0.0.1:9200
#        - <elasticsearch_ip_node_2>:9200
#        - <elasticsearch_ip_node_3>:9200

output.elasticsearch:
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/wazuh-server.pem"
  ssl.key: "/etc/filebeat/certs/wazuh-server-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false

filebeat.modules:
  - module: wazuh
    alerts:
      enabled: true
    archives:
      enabled: true

logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat

^G Help        ^O Write Out   ^W Where Is   ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
^X Exit        ^R Read File   ^\ Replace    ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy
```

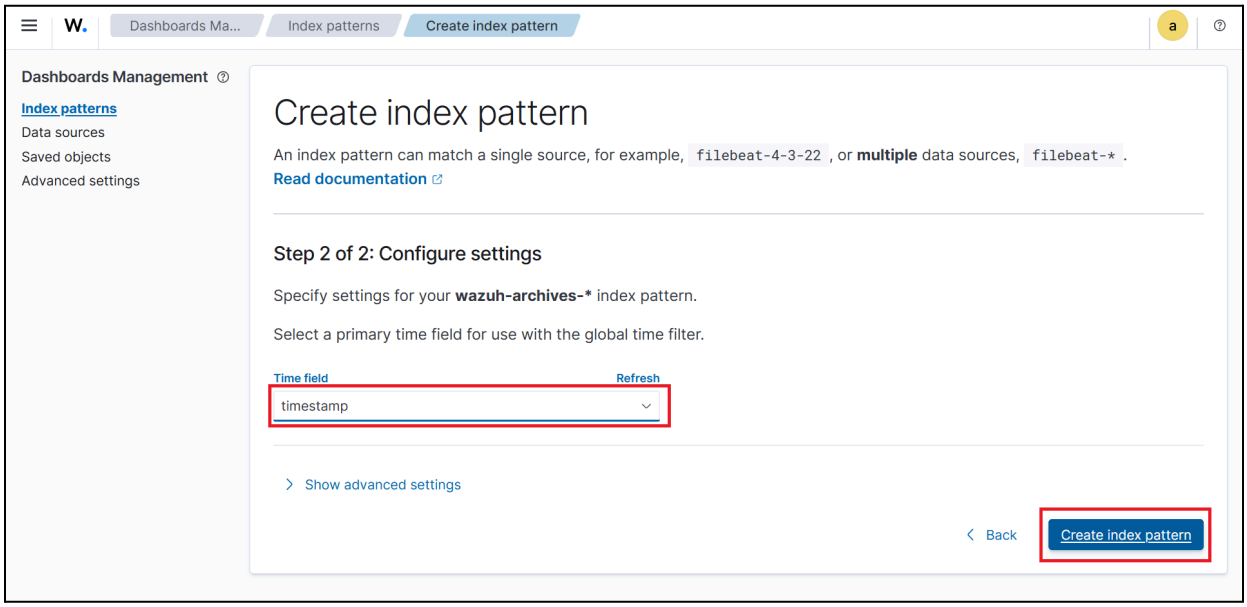
save (**`Ctrl + X`** → **`Y`** → **`Enter`**), and restart: `systemctl restart filebeat.service`

Now return to your **Wazuh dashboard** and create a **new index**.

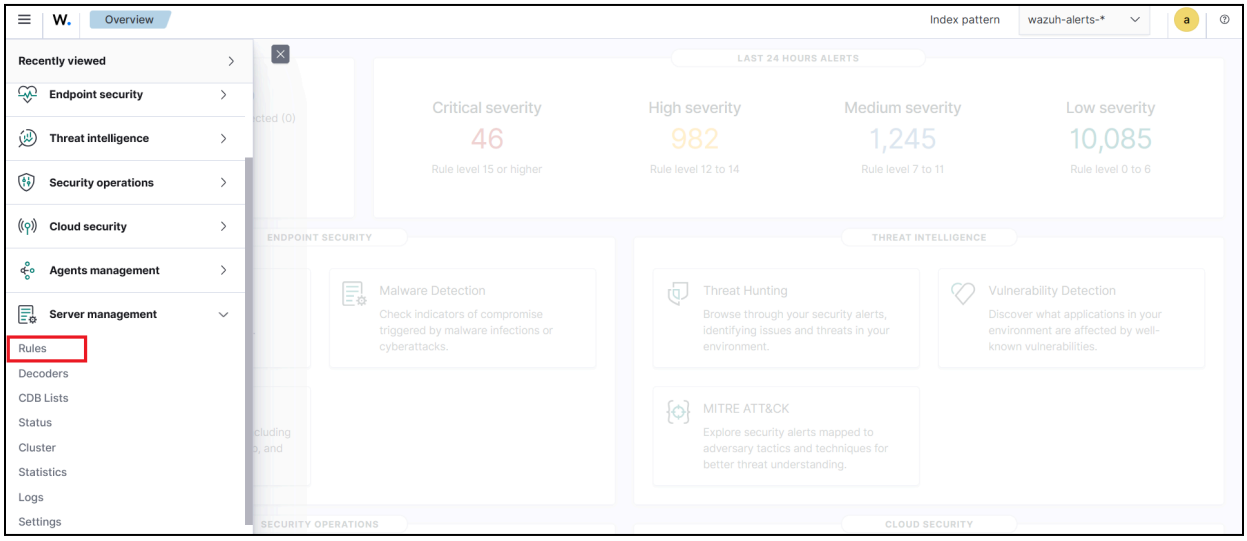


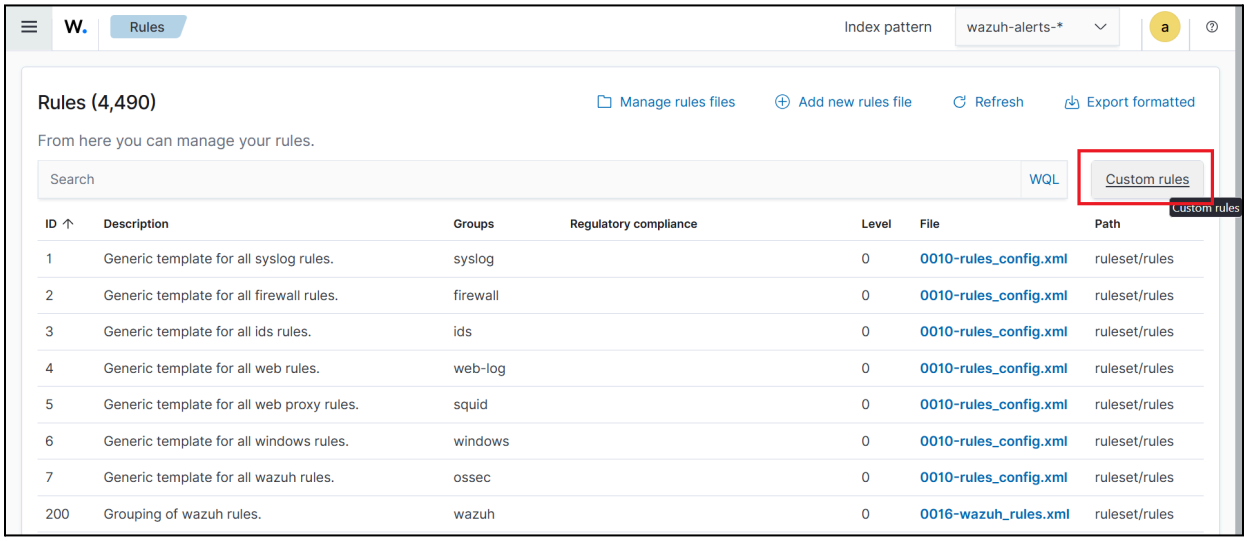

Dashboards Ma...
Index patterns
a
Dashboards Management
Index patterns
Data sources
Saved objects
Advanced settings
Index patterns
+ Create index pattern
Create and manage the index patterns that help you retrieve your data from OpenSearch.
Search...
Pattern
Data Source Connection
wazuh-alerts-*
Default
wazuh-monitoring-*
wazuh-statistics-*
Rows per page: 10
1


Dashboards Ma...
Index patterns
Create index pattern
a
Dashboards Management
Index patterns
Data sources
Saved objects
Advanced settings
Create index pattern
An index pattern can match a single source, for example, filebeat-4-3-22 , or multiple data sources, filebeat-* .
Read documentation
Step 1 of 2: Define an index pattern
Index pattern name
wazuh-archives-*
Next step
Use an asterisk (*) to match multiple indices. Spaces and the characters \, /, ?, ", <, >, | are not allowed.
Include system and hidden indices
Your index pattern matches 1 source.
wazuh-archives-4.x-2025.09.15
Index
Rows per page: 10



Create a Detection Rule for Mimikatz





Add the following custom rule to your ruleset:

```
<group name="local,syslog,sshd,">

  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed, pci_dss_10.2.4, pci_dss_10.2.5</group>
  </rule>

  <rule id="100002" level="15">
    <if_group>sysmon_event1</if_group>
    <field name="win.eventdata.originalFileName"
```

```
type="pcre2">(?i)mimikatz\.exe</field>
    <description>Suspicious Mimikatz activity detected on
system</description>
    <mitre>
      <id>T1003</id>
    </mitre>
  </rule>

</group>
```

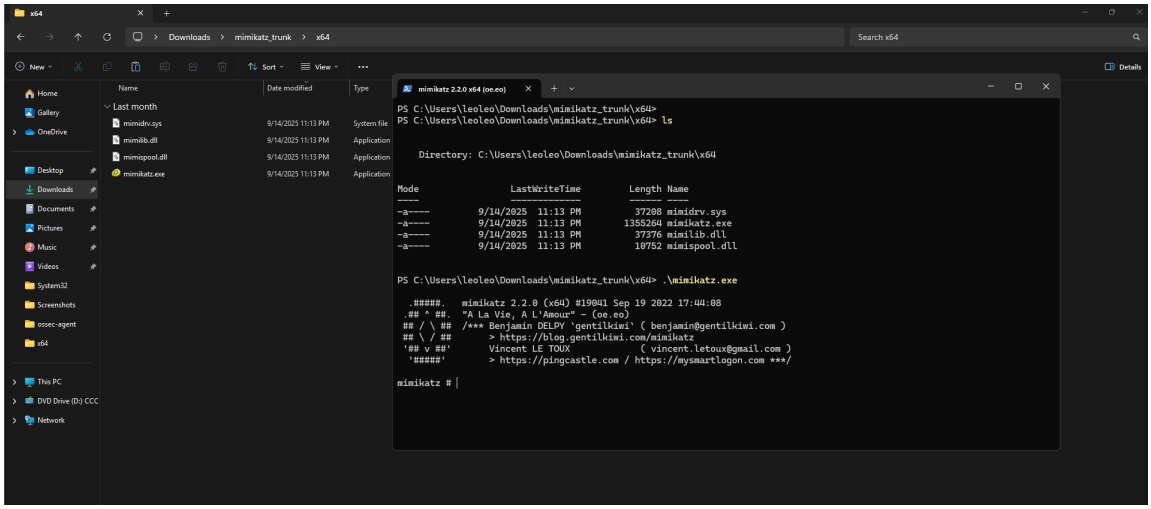
Rule Explanation (In Simple Terms)

- **Rule 1:** Detects repeated SSH authentication failures from IP **`1.1.1.1`**.

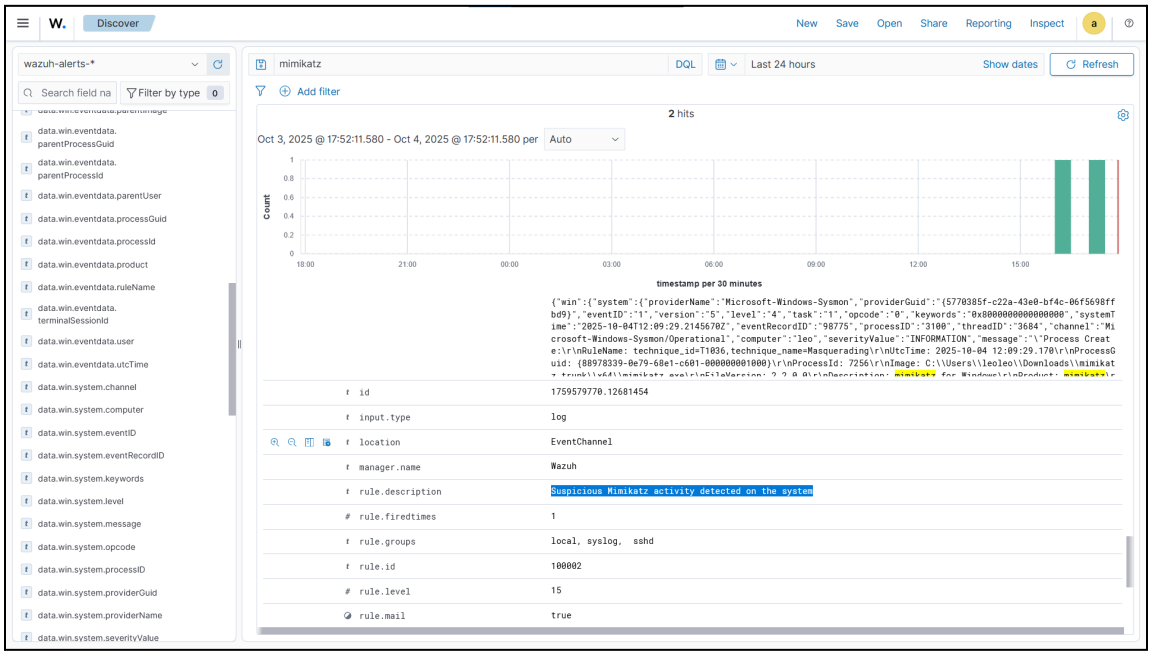
- **Rule 2:** Detects Sysmon Event ID 1 (process creation) if the executable name is **`mimikatz.exe`**.
  → This triggers a **level 15 alert** with MITRE technique **T1003 (Credential Dumping)**.

Validate the Rule

1. Run **Mimikatz** again in your Windows VM.



2. Go to your **Explore** → **Discover** → select **wazuh-alerts** → search **mimikatz**.You should see an alert labeled **“Suspicious Mimikatz activity detected on system”**



Automating SOC Alerts with Shuffle

Now that detection is working, let's automate the response workflow using **Shuffle**.

Create a Shuffle Account

Go to https://shuffler.io → **Sign up / Log in** → In the sidebar, click **Workflows** → **+ Create Workflow** → name it **`soc-workflow`**.

Add Webhook Integration

- From the **left panel**, drag **Webhook** onto the canvas.
- Copy its **Webhook URI**.

Now, go back to your **Wazuh VM** and open the config file: `sudo nano /var/ossec/etc/ossec.conf`

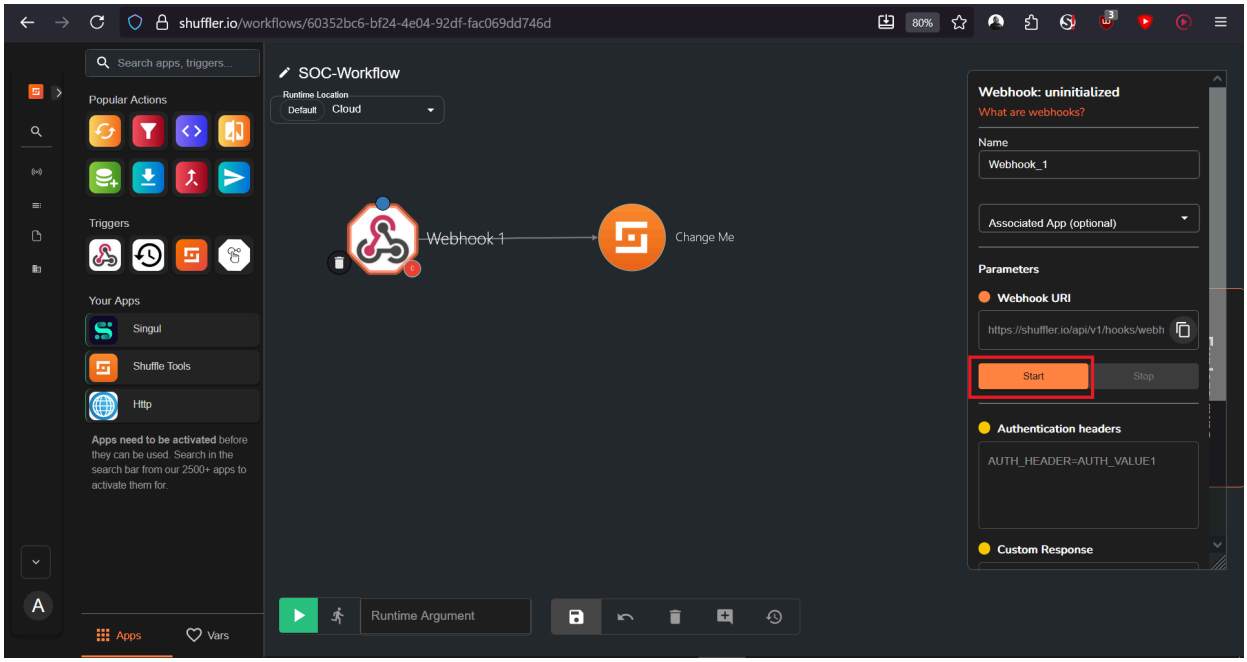
Add this under the **`</global>`** tag:

```
<integration>
  <name>shuffle</name>
  <hook_url>{your-webhook-URI}</hook_url>
  <rule_id>100002</rule_id>
  <alert_format>json</alert_format>
</integration>
```

```
 GNU nano 7.2                                        /var/ossec/etc/ossec.conf *
  Wazuh - Manager - Default configuration for ubuntu 24.04
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <integration>
    <name>shuffle</name>
    <hook_url>https://shuffler.io/api/v1/hooks/webhook_c569ced1-80f5-4d2d-875e-2ecf85e14abb</hook_url>
    <rule_id>100002</rule_id>
    <alert_format>json</alert_format>
  </integration>

  <alerts>
    <log_alert_level>3</log_alert_level>
File Name to Write: /var/ossec/etc/ossec.conf
^G Help          M-D DOS Format     M-A Append         M-B Backup File
^C Cancel        M-M Mac Format     M-P Prepend        ^T Browse
```

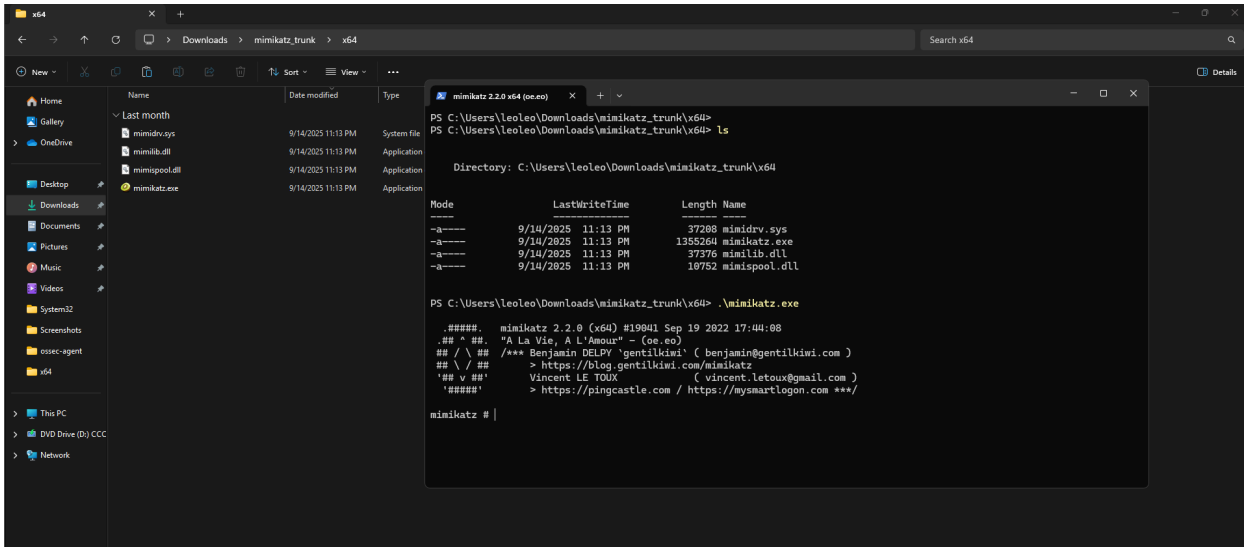
Save and restart the service: `sudo systemctl restart wazuh-manager.service`

In Shuffle, click **Start Webhook** to begin listening.

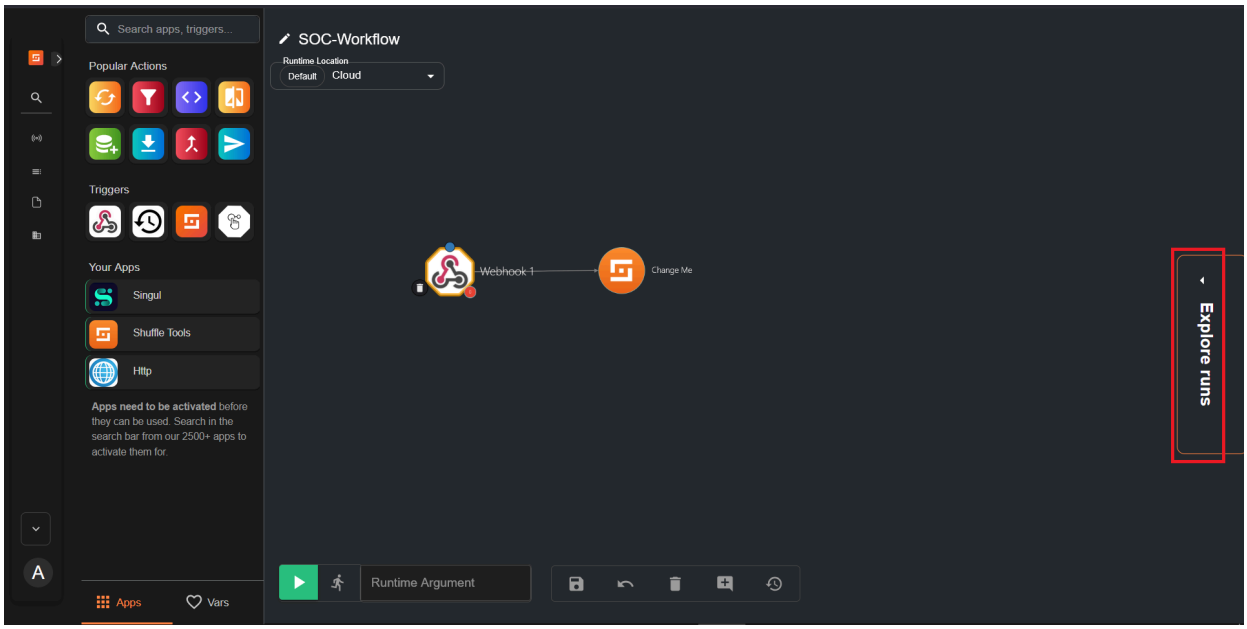


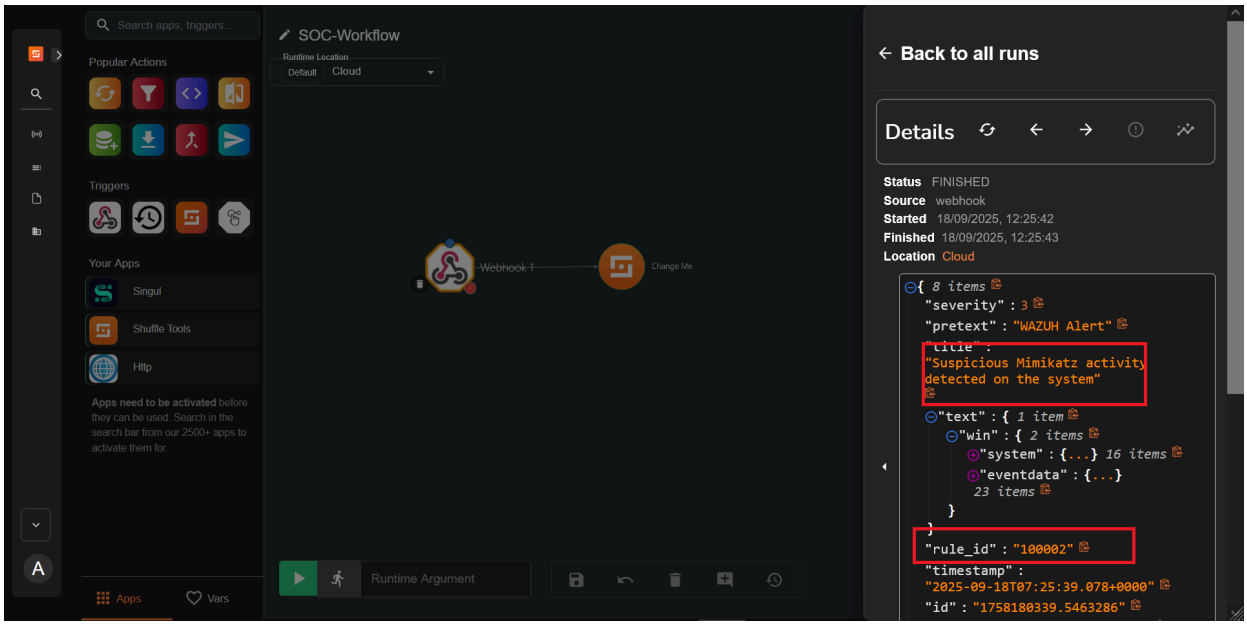
Test the Webhook

Run **Mimikatz** again in your Windows VM.



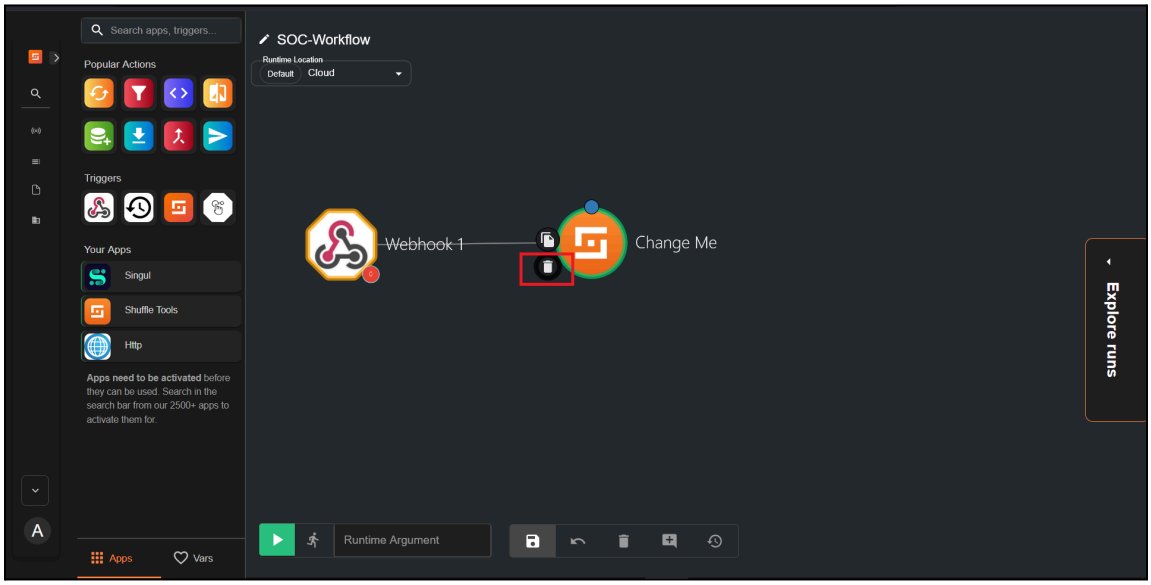
Then go to **Explore Runs** in Shuffle → you should see data received successfully.





Add SHA256 Extraction

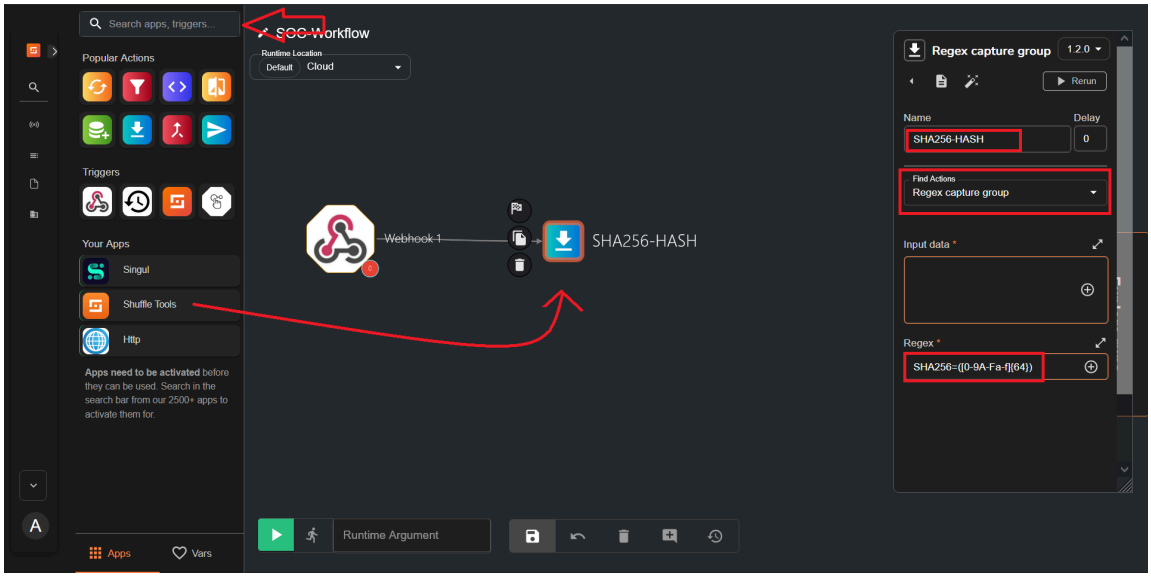
- Delete the “Change Me” icon.



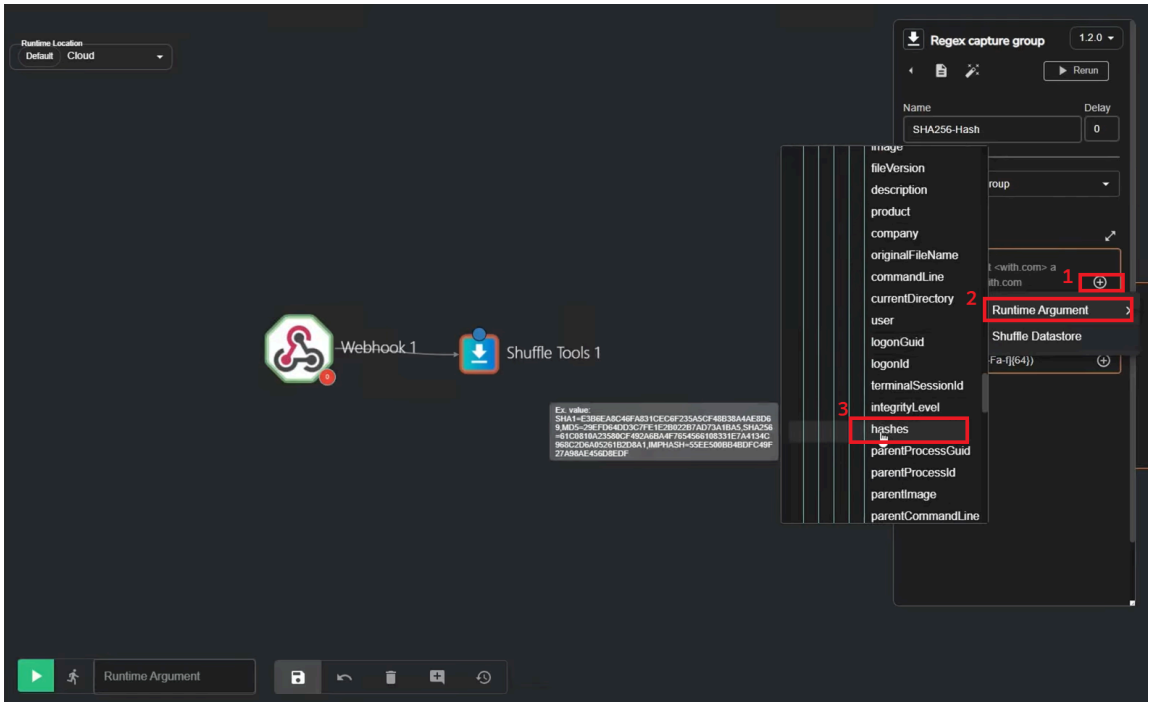
- Drag **Shuffle Tools** → rename it → under **Find Actions**, choose **`Regex Capture Group`**.

- Use this Regex:

SHA256=([0-9A-Fa-f]{64})

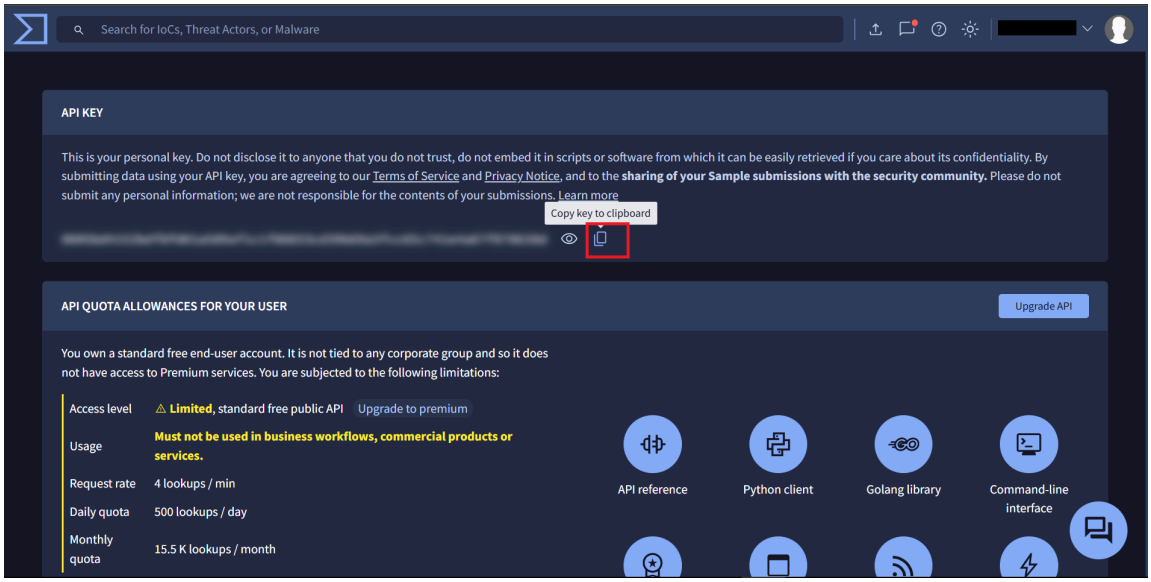


- For **Input data**, click the ⊕ icon → select **Runtime Argument** → **hashes**.

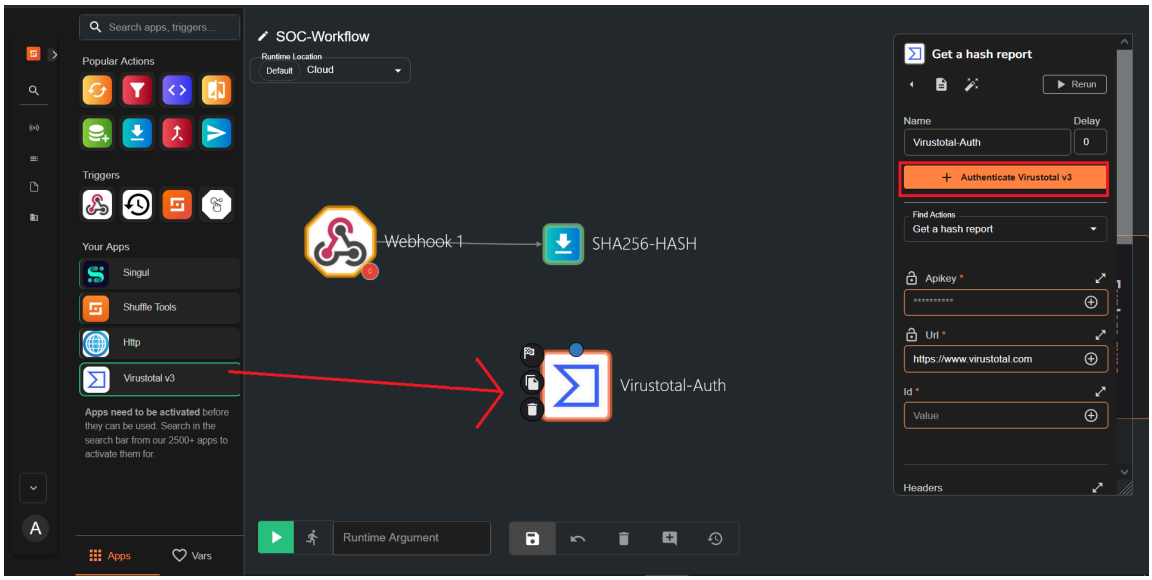


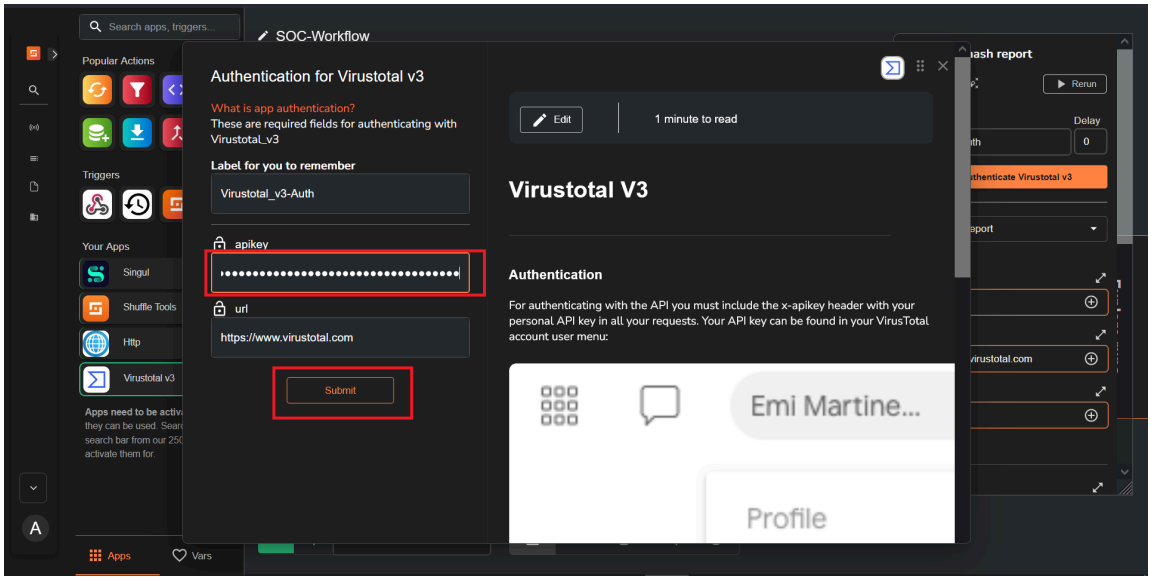
Add VirusTotal Integration

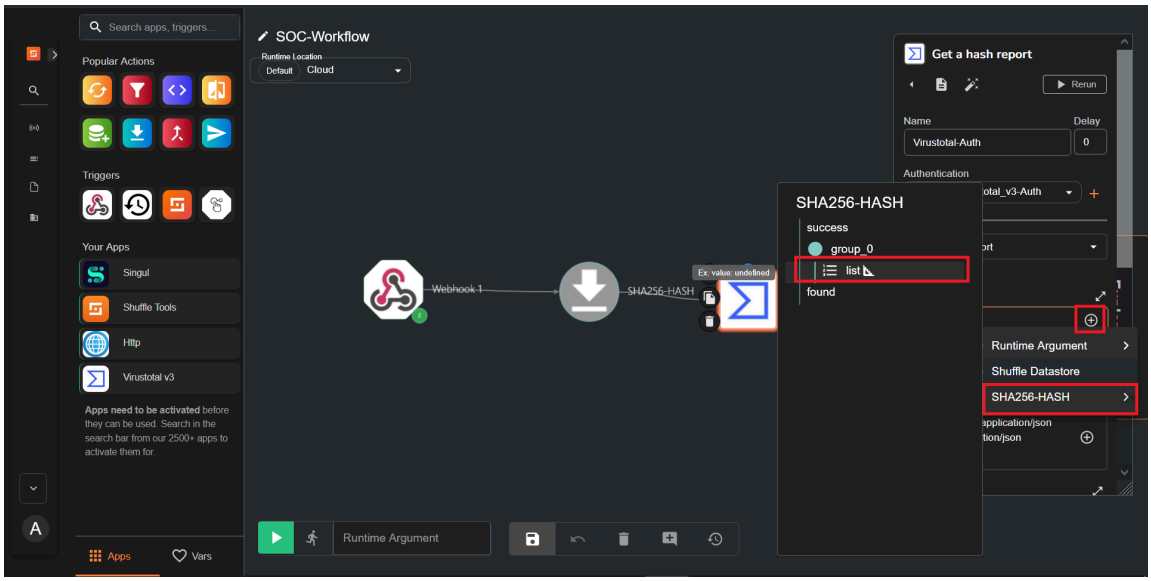
- Go to https://www.virustotal.com→ Sign in → click your profile → **API Key** → copy it.



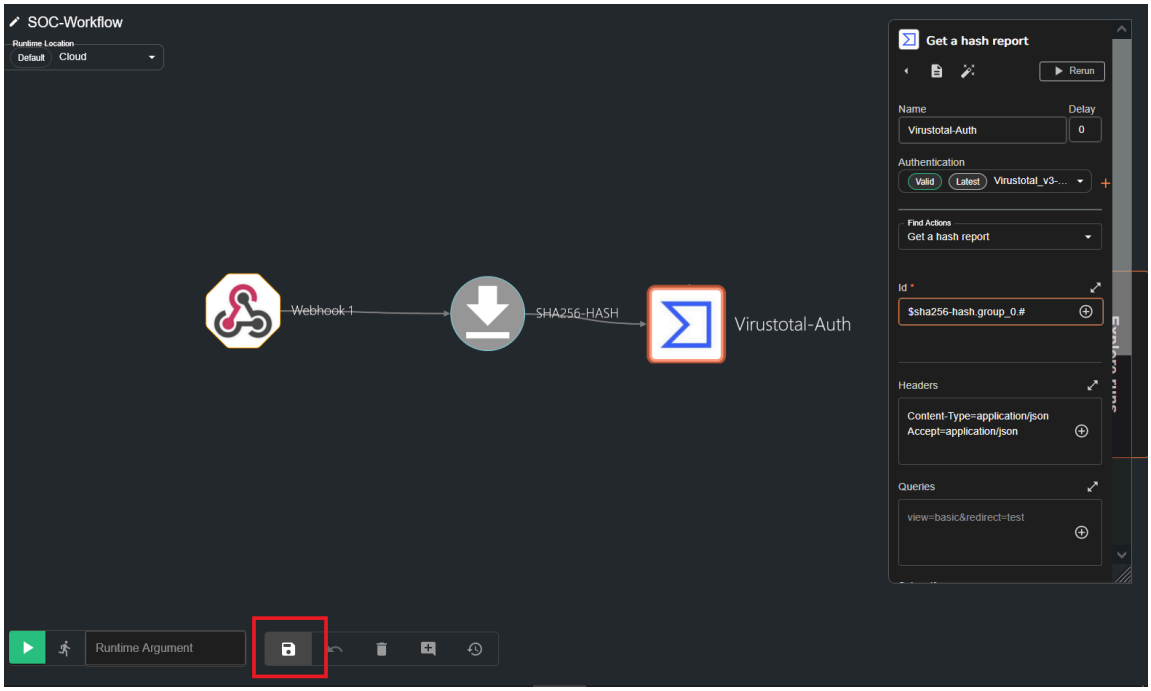
- In Shuffle, drag the **VirusTotal** app → click **Authenticate** → paste your API key.




Search apps, triggers...
Popular Actions
Triggers
Your Apps
Singul
Shuffle Tools
Http
Virustotal v3
Apps need to be activ
they can be used. Sear
search bar from our 25
activate them for.
Apps
Vars
SOC-Workflow
Authentication for Virustotal v3
What is app authentication?
These are required fields for authenticating with Virustotal_v3
Label for you to remember
Virustotal_v3-Auth
apikey
url
https://www.virustotal.com
Submit
Edit
1 minute to read
Virustotal V3
Authentication
For authenticating with the API you must include the x-apikey header with your personal API key in all your requests. Your API key can be found in your VirusTotal account user menu:
Emi Martine...
Profile
Rerun
Delay
0
Authenticate Virustotal v3
virustotal.com

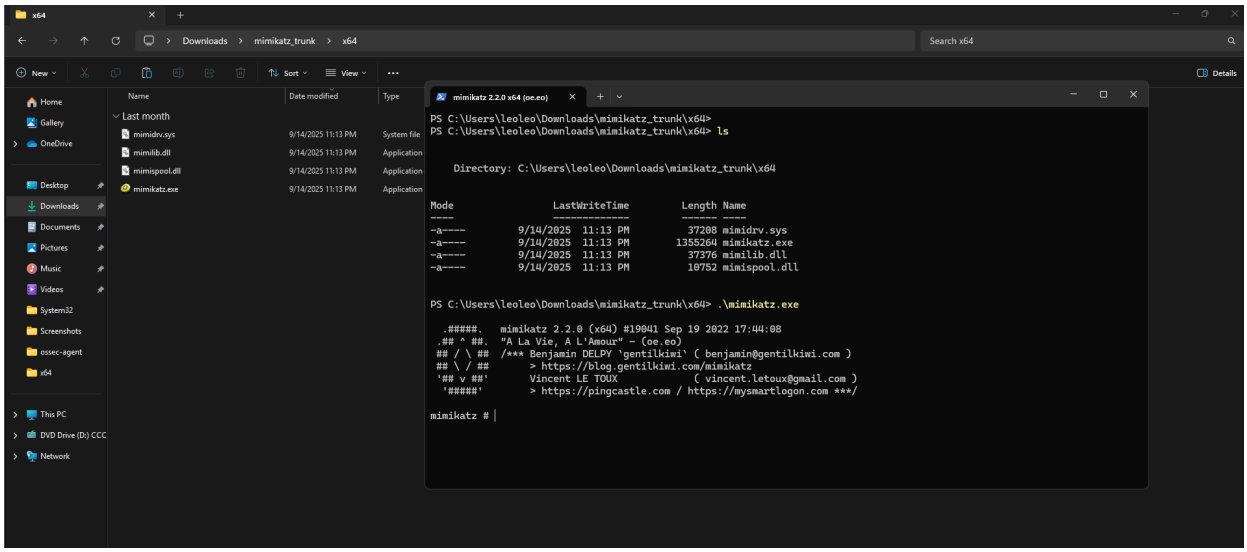

Search apps, triggers...
Popular Actions
Triggers
Your Apps
Singul
Shuffle Tools
Http
Virustotal v3
Apps need to be activated before they can be used. Search in the search bar from our 2500+ apps to activate them for.
Apps
Vars
SOC-Workflow
Runtime Location
Default
Cloud
Webhook 1
SHA256-HASH
Ex. value: undefined
Runtime Argument
Get a hash report
Rerun
Name
Virustotal-Auth
Delay
0
Authentication
SHA256-HASH
success
group_0
list
found
Runtime Argument
Shuffle Datastore
SHA256-HASH

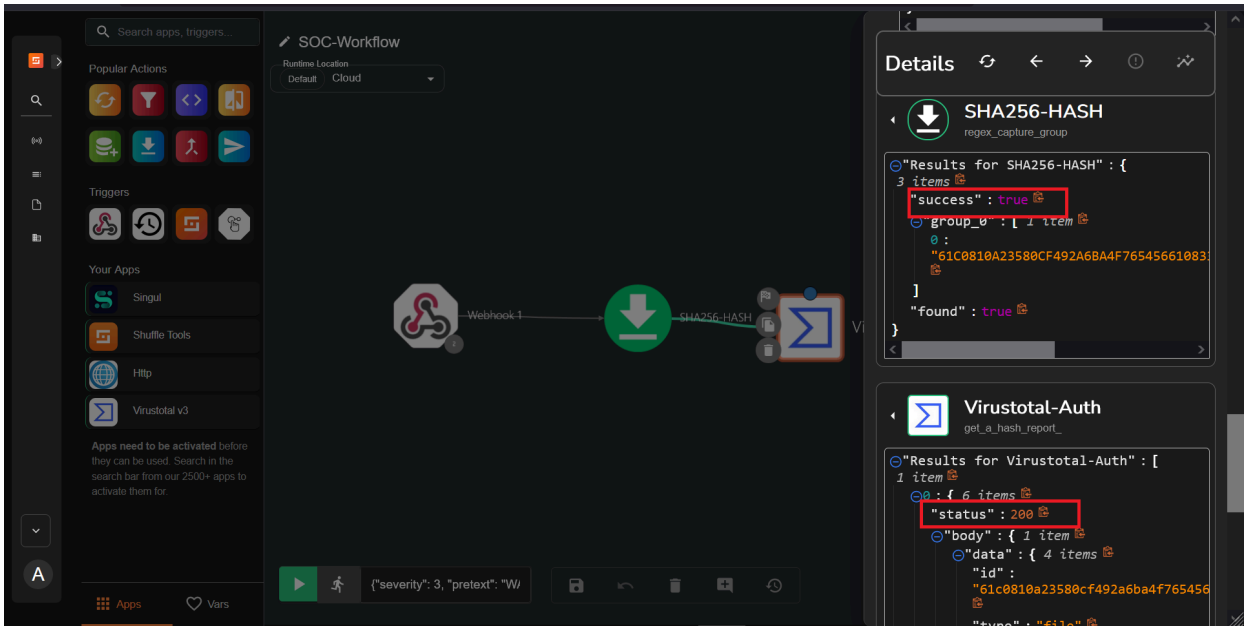
- Connect it with the SHA256 output.



Save your workflow and refresh the page.

Test again by running Mimikatz in your Windows VM → you should see **`"success": true`** and **`"status": 200`** in Shuffle.

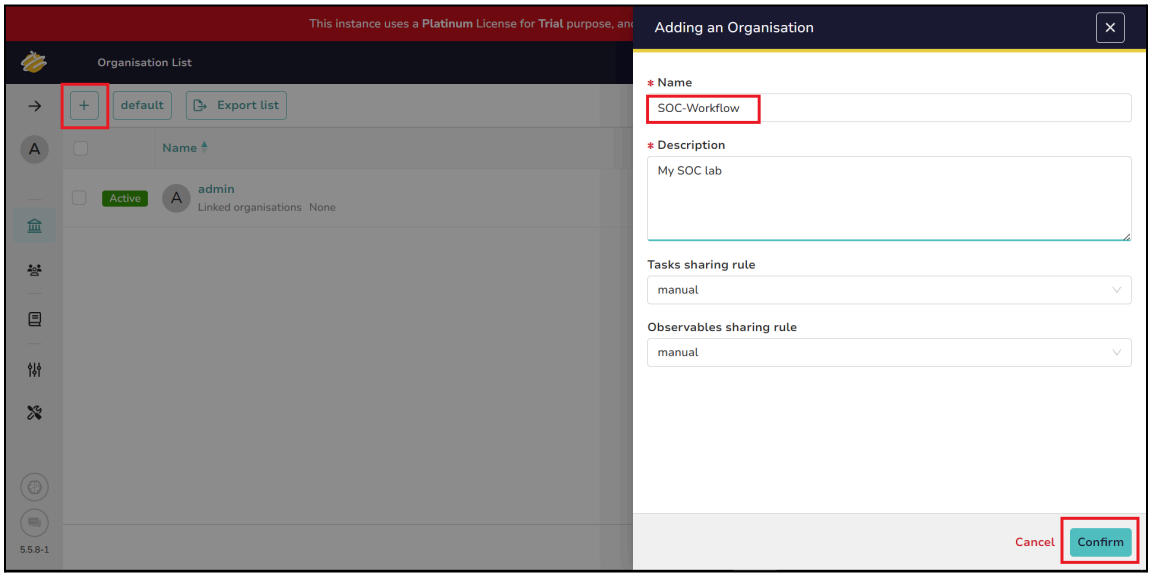




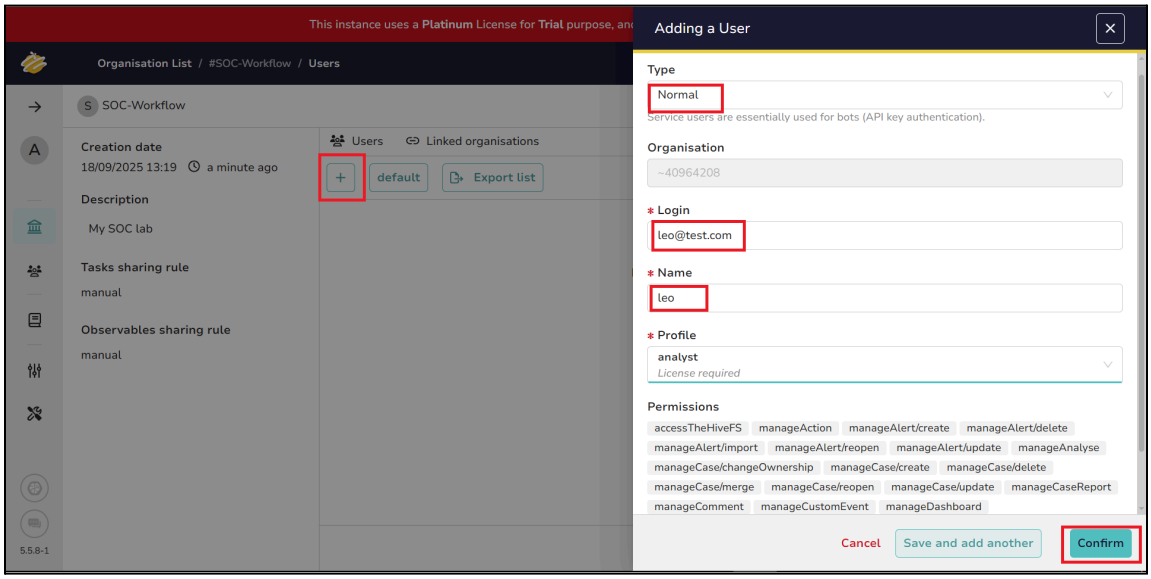
Add TheHive Integration

Go to your TheHive dashboard: http://{your-thehive-vm-ip}:9000

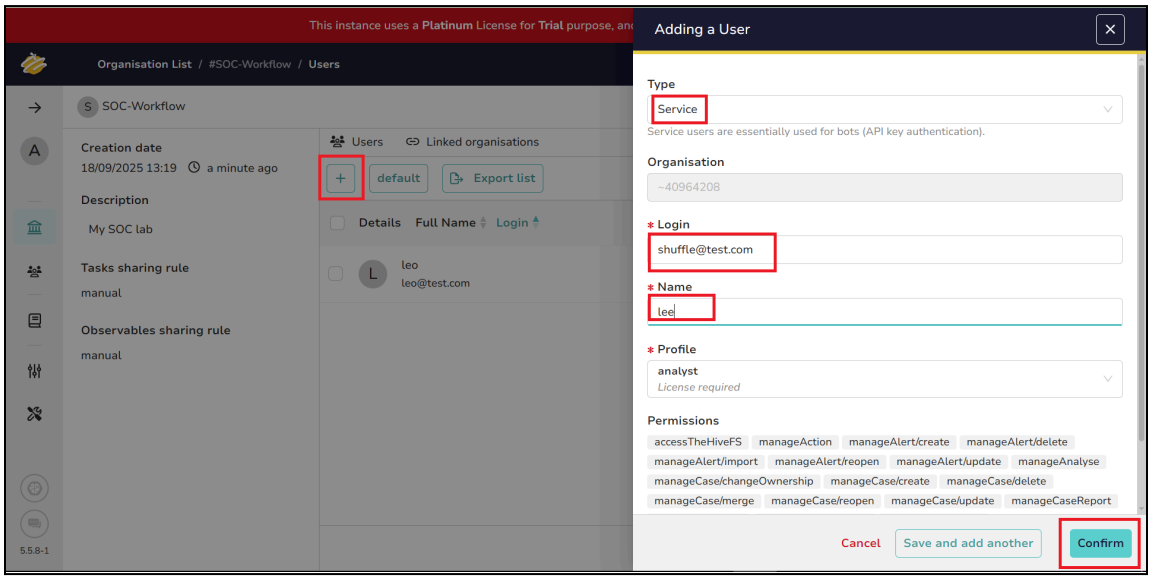
1. Create a new **Organization** → click **+**.



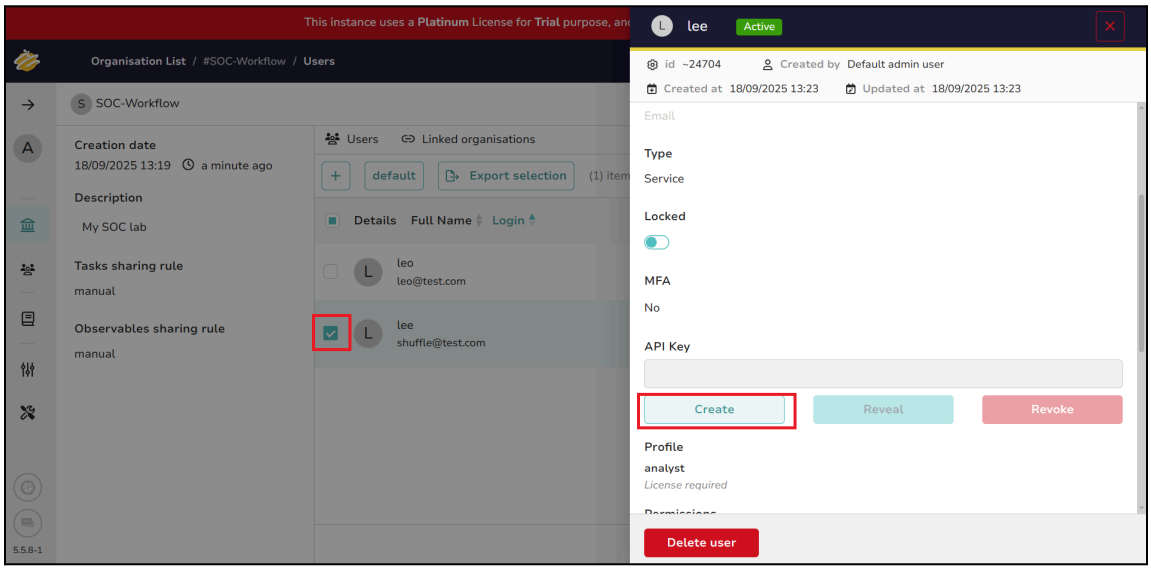
2. Inside it, create a **User Account** → click **+** → fill in details → set password → Confirm.

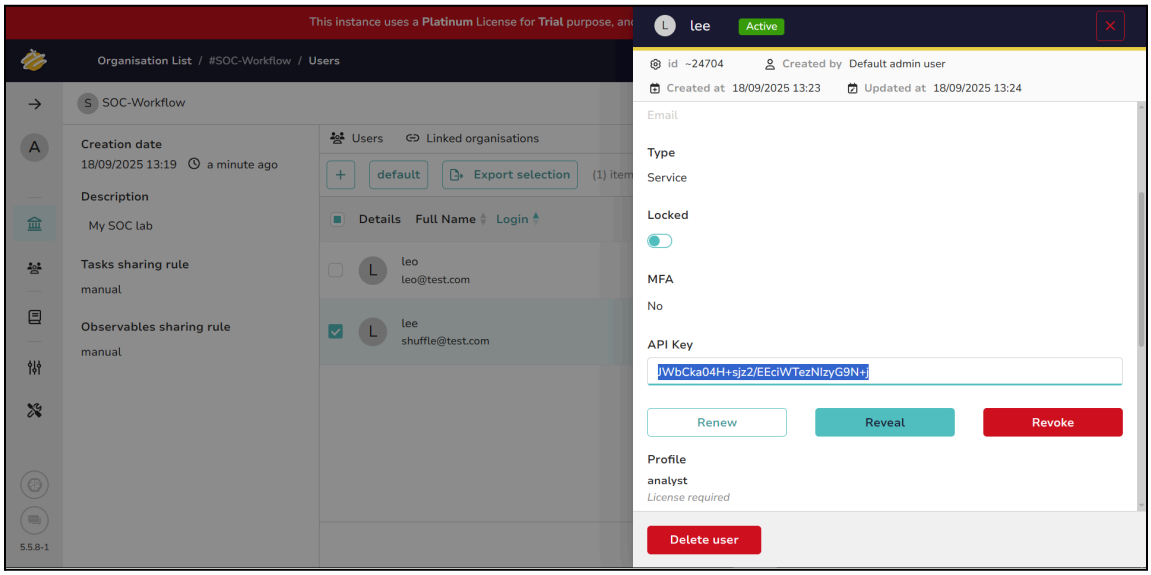


3. Inside it, also create a **Service Account** → click **+** → fill in details → set password → Confirm.



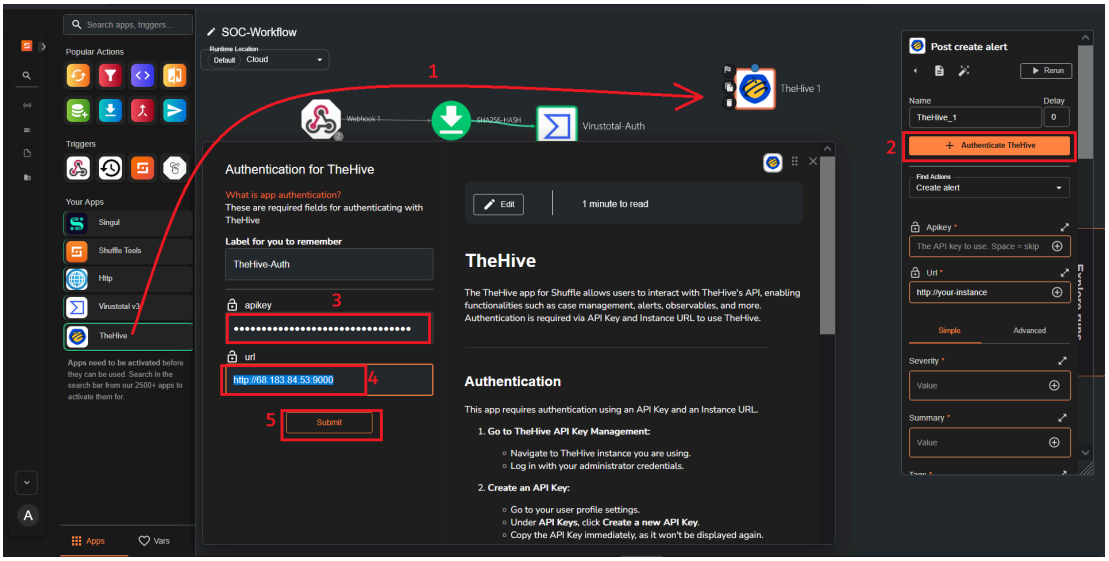
4. Go to **Service Accounts** → create → copy its **API key**





Now in Shuffle:

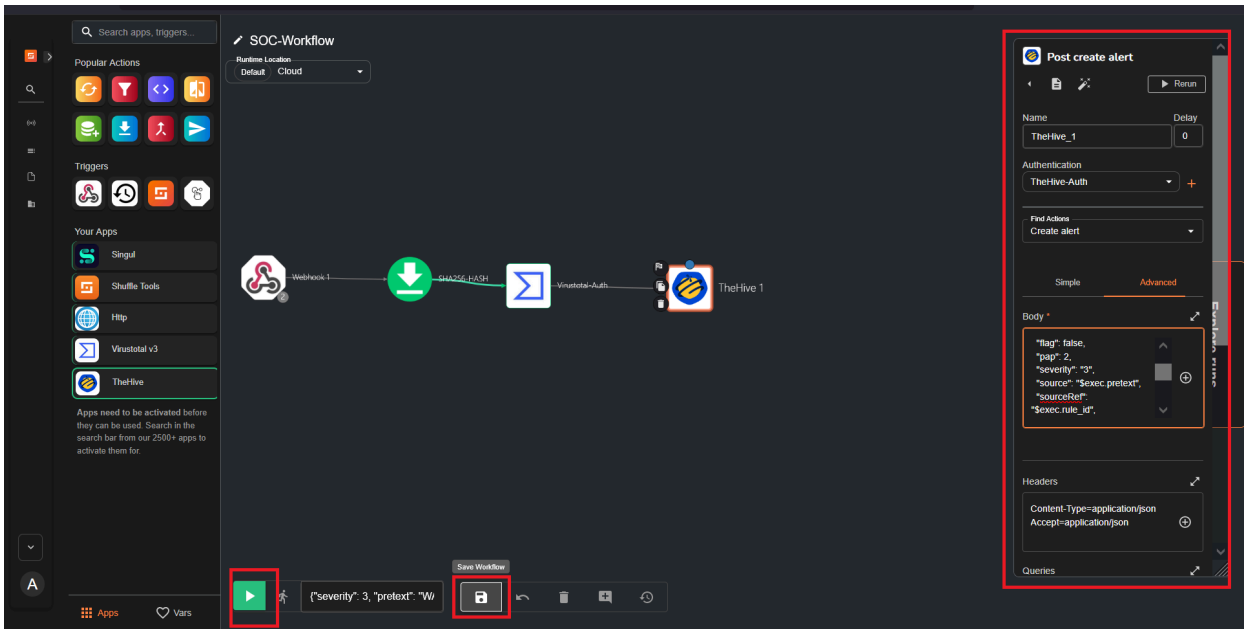
- Drag **TheHive** app → click **Authenticate** → use the service account API key.



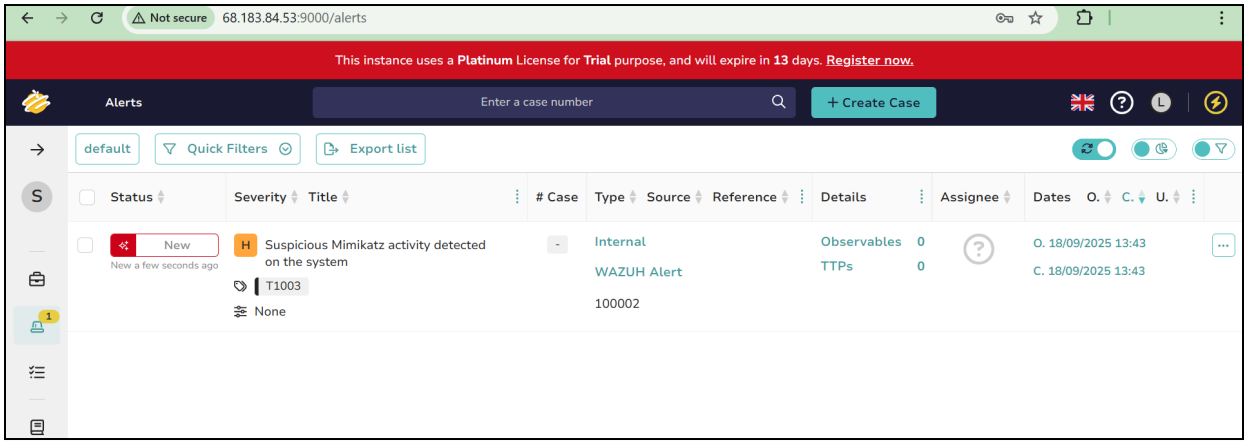
- Connect it after VirusTotal.

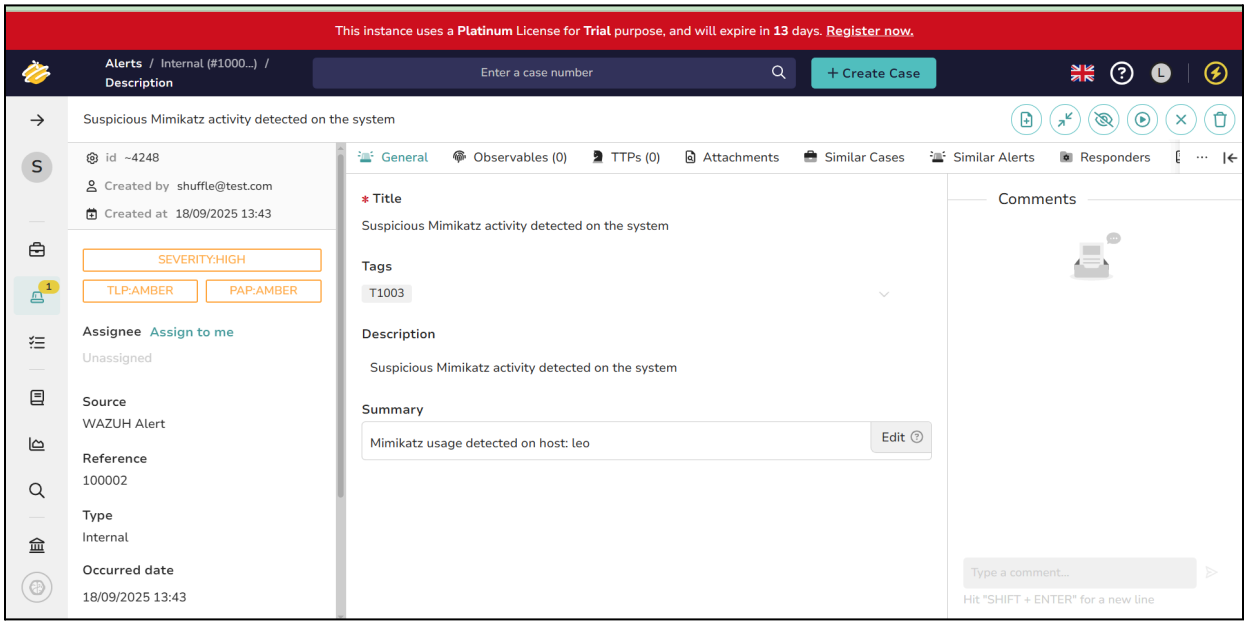
Paste this body:

```
    {
  "description": "$exec.title",
  "externallink": "${externallink}",
  "flag": false,
  "pap": 2,
  "severity": "3",
  "source": "$exec.pretext",
  "sourceRef": "$exec.rule_id",
  "status": "New",
  "summary": "Mimikatz usage detected on host: $exec.text.win.system.computer",
  "tags": ["T1003"],
  "title": "$exec.title",
  "tlp": "2",
  "type": "Internal"
}
```



Save and run the workflow → verify on TheHive that a **new alert** appears.



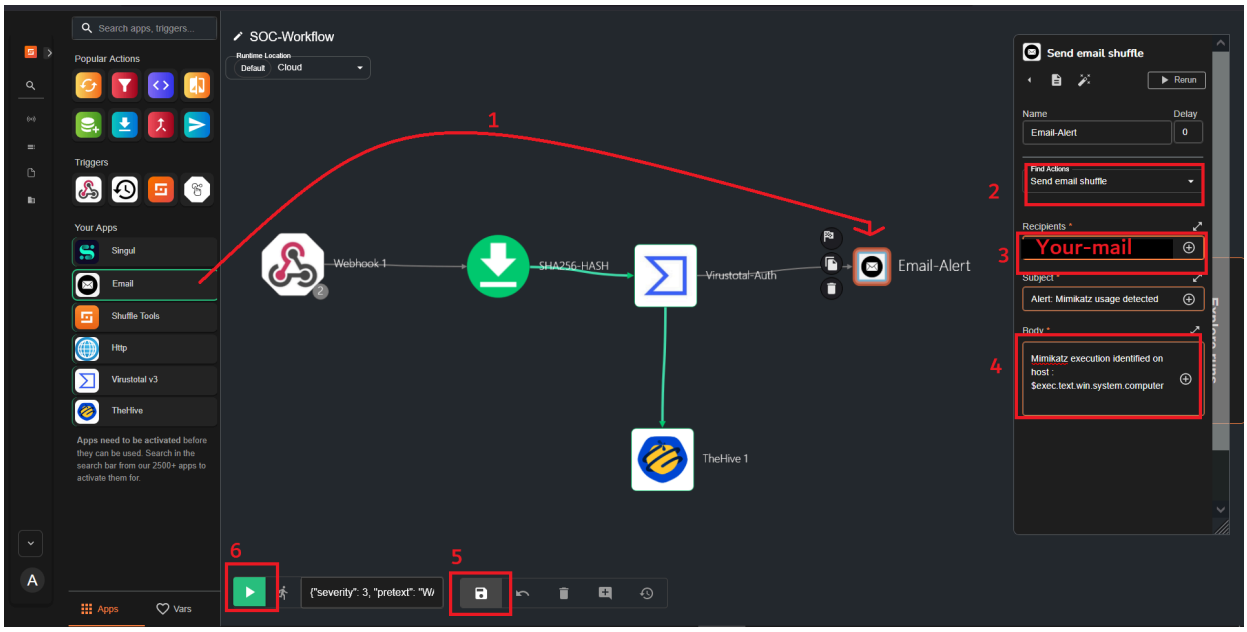


Add Email Notification

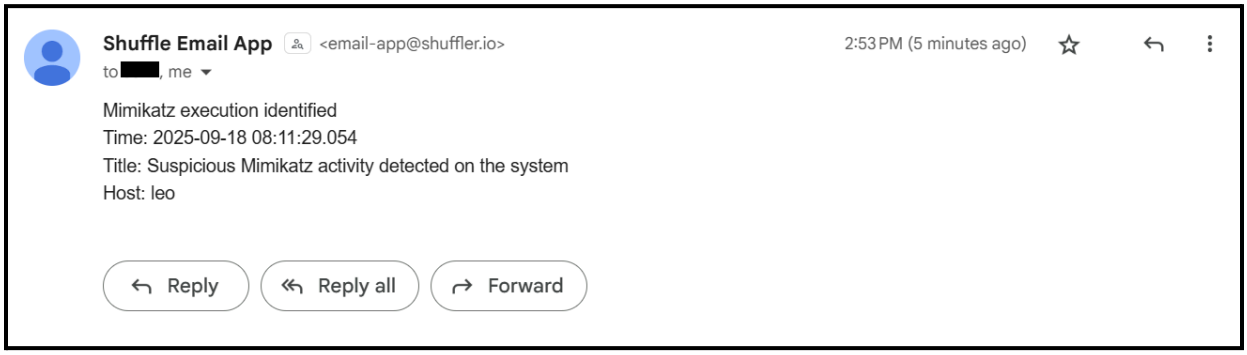
- Drag the **Email** app onto the board.
- Connect it after VirusTotal or TheHive.
- Configure the recipient email address.
- Paste this into the Body field:

```
Mimikatz execution identified
Time: $exec.all_fields.full_log.win.eventdata.utcTime
Title: $exec.title
Host: $exec.text.win.system.computer
```

You can also include other dynamic elements by clicking the ⊕ icon → **Runtime Argument** → and selecting the desired field (for example, *rule_id*, *hashes*, or *description*).



Save and run the workflow again you should now receive an email alert.



You've successfully built and automated an end-to-end SOC response pipeline.
Here's what happens when **Mimikatz** is executed on your Windows VM:

1. **Wazuh** detects the activity using custom rule **`100002`**.
2. **Shuffle** receives the alert through the Wazuh webhook.
3. The workflow automatically **extracts the SHA256 hash** of the detected file.
4. **VirusTotal** is queried to check the file's reputation.
5. If confirmed suspicious, a **case is created in TheHive** for investigation.

6. **Email notifications** are sent to inform the analyst of the incident.

Conclusion

By completing this lab, you have successfully built a functional SOC automation pipeline that integrates Wazuh, TheHive, Shuffle, and VirusTotal.
You now understand how alerts can be automatically detected, analyzed, enriched, and escalated without manual effort.

You can extend this lab by:

- Adding more detection rules, such as PowerShell abuse or persistence techniques
- Integrating additional threat intelligence sources like AbuseIPDB or AlienVault OTX
- Creating advanced workflows in Shuffle to automate alert enrichment and case prioritization

This lab serves as a foundation for building enterprise-grade SOC automation using open-source tools. Continue exploring, testing, and improving to enhance your incident response capabilities.