

**Integrated Network & Host Monitoring for Enterprise Web Infrastructure: A VM-based SOC Lab(Guide)**

**The steps implemented in this Guide:**

**1.Different OS setup in the VM (kali-linux,ubuntu,windows 10, windows server,linux-server,cloud server).**

**2.Making the Central SIEM Manger into Bridge Mode for Network logs(Ubuntu).**

**3.Seting Up the Network based IDS(snort) on Central SIEM Manger(Ubuntu).**

**4.In the Central SIEM Manger setup wazuh-Manger(Ubuntu).**

**5.Setup of wazuh in the different OS to forward the logs (kali-linux,window-10,windows-server,cloud serve,linux-server).**

**6. Make detection rules in the central SIEM .**

**7.Simluating the attacks to check the detection rules are working and making a ticket in the slack**

**Step-1:**

**Different OS setup in the VM (kali-linux,ubuntu,windows 10, windows server,linux-server,cloud server).**

* Kali-Linux os (<https://www.kali.org/get-kali/#kali-virtual-machines>)
* Ubuntu OS (<https://ubuntu.com/download/desktop>)
* Windows-10 os(<https://www.microsoft.com/en-in/software-download/windows10>)
* Windows-server 2022  
  (<https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2022>)
* Kali-linux server os (<https://www.kali.org/get-kali/#kali-virtual-machines>)

**Youtube Guide**

* **How to Create a Windows 11 Virtual Machine With VMware** [**YouTube**](https://www.youtube.com/watch?pp=0gcJCdgAo7VqN5tD&v=UB8CQC_lT5U&utm_source=chatgpt.com)
* **How to install Ubuntu Linux (Desktop) in VMware Workstation** [**YouTube**](https://www.youtube.com/watch?v=Ot1R1qPjRiU&utm_source=chatgpt.com)
* **Installing Kali on VMware** [**YouTube**](https://www.youtube.com/watch?v=FDSwC9WQV3s&utm_source=chatgpt.com)

**2.How To Enable Promiscuous Mode In Vmware Workstation(Ubuntu)**

Promiscuous mode is a network interface card (NIC) mode where the card passes all traffic it receives to the CPU rather than only frames addressed to it. This is commonly used in network analysis and monitoring applications, allowing tools to inspect all traffic on the network segment. In VMware Workstation, enabling promiscuous mode can be essential for certain tests, penetration testing practices, use of network sniffers, or learning environments involving network traffic analysis. This detailed guide provides step-by-step instructions on enabling promiscuous mode in VMware Workstation.

**Step-by-Step Guide: Enabling Promiscuous Mode**

**Step 1: Accessing VM Settings**

1. **Open VMware Workstation:** Launch the VMware Workstation application on your host machine.
2. **Select the Virtual Machine:** In the VMware Workstation interface, click on the virtual machine for which you want to enable promiscuous mode.
3. **Power Off the VM:** If the VM is currently running, you must power it off first. Promiscuous mode cannot be enabled while the VM is active.
4. **Open Settings:** Right-click on the VM and select “Settings,” or from the VM menu, click on “Edit Virtual Machine Settings.”

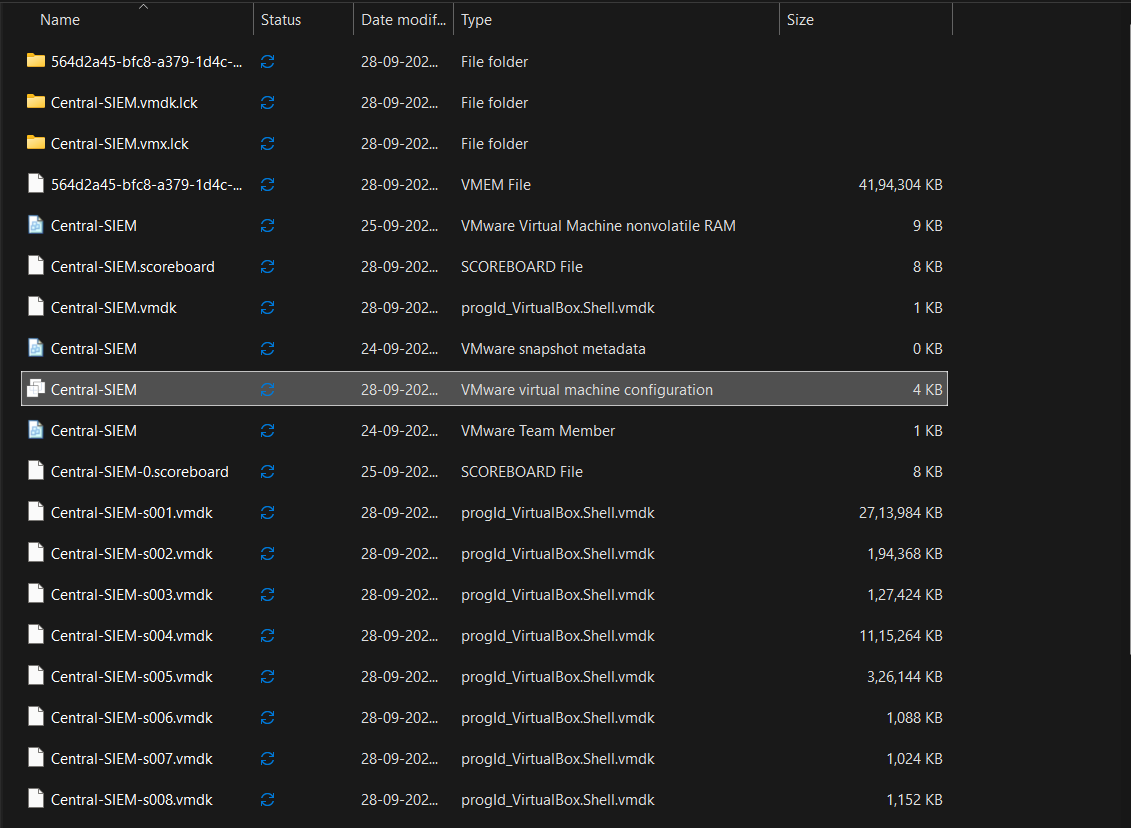
**Step 2: Configuring the Network Adapter**

1. **Select the Network Adapter:** In the virtual machine settings window, find and click on the “Network Adapter” option. Here, you will see settings related to the network for the virtual machine.
2. **Change Network Connection Type:** Ensure that the network connection type is set appropriately:
   * For promiscuous mode to take effect, the virtual network adapter should typically be set to "Bridged" or "Host-only" mode.
   * If set to "NAT," promiscuous mode will not work effectively since NAT configurations isolate VM traffic.
3. **Enable Promiscuous Mode:** Depending on your version of VMware Workstation, you may have an option labeled "Promiscuous Mode" within the settings of the network adapter:
   * If this option is available, slide the control to "Allow All" or "Allow VMs," depending on your needs:
     + **Allow All:** All traffic to and from the network adapter will be captured.
     + **Allow VMs:** Only traffic to and from VMs running on the same host will be captured.
   * If the option is not directly available, we will manually modify the configuration files in the next steps.

**Step 3: Editing The VMX File (if applicable)**

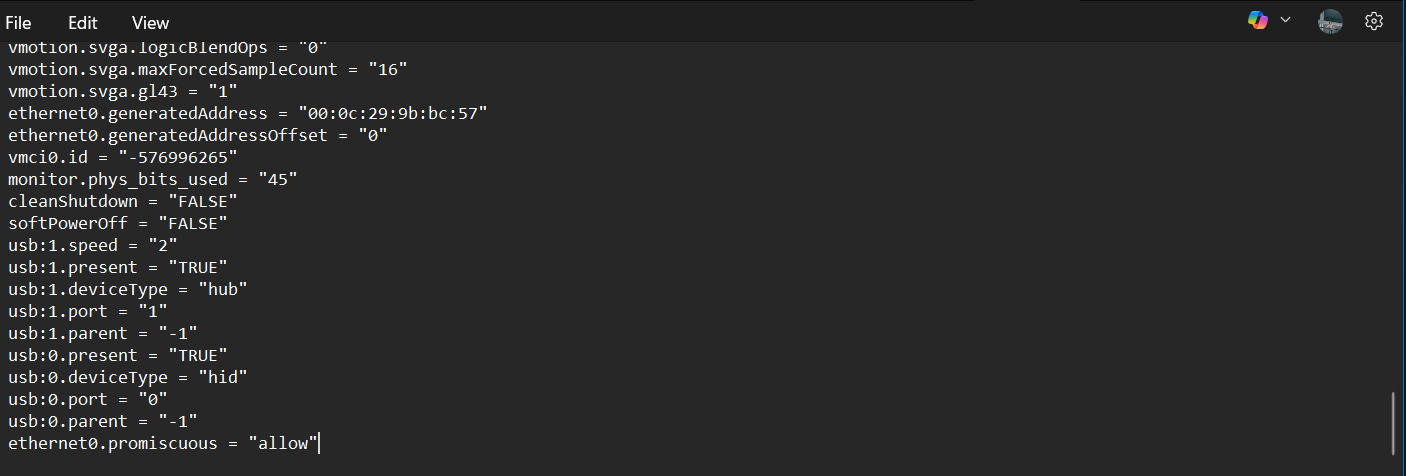
* 1. **Locate the VM’s .vmx File:**

The .vmx file contains configuration settings for your virtual machine. To locate it, open the file system, navigate to your virtual machine’s directory, and find the .vmx file corresponding to the VM you are configuring.

****

* 1. **Edit the .vmx File:**
     + Open the .vmx file in a text editor (you may need to run the editor as an administrator).
  2. **Add the Promiscuous Mode Setting:**
     + Add the following line to the bottom of the file:

**ethernet0.promiscuous = "allow"**

****

* + - This line instructs VMware to allow promiscuous mode for the first network adapter. If you have multiple network adapters, you may want to refer to them as ethernet1, ethernet2, etc.

**Optional Verification via Command Line**

You can also check if your interface is in promiscuous mode:

**ip link show eth0(here your NIC name)**

Look for the word PROMISC in the output. Example:

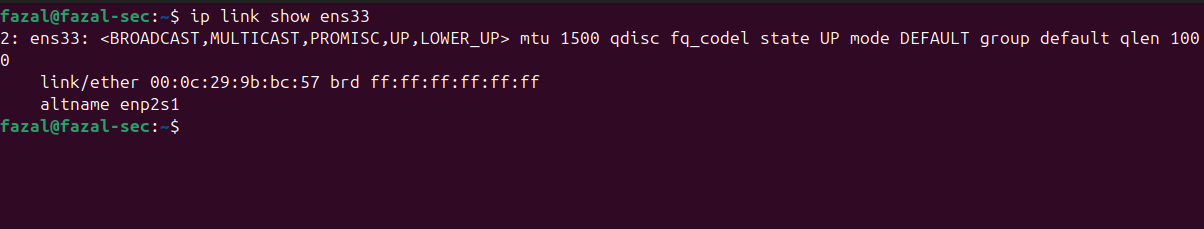
**2: eth0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER\_UP> ...**

* PROMISC → Promiscuous mode is active.
* Missing → It is not enabled.

**Force the interface into promiscuous mode**

Run:

**sudo ip link set eth0 promisc on**

****

**3.Seting Up the Network based IDS(snort) on Central SIEM Manger(Ubuntu).**

**1. Install Snort on Ubuntu**

Run these commands as root or with sudo:

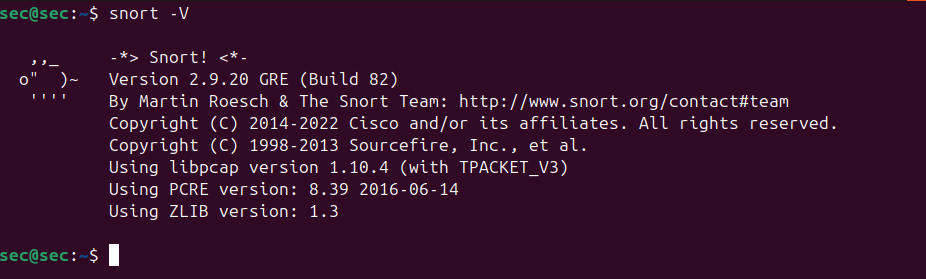
**sudo apt update && sudo apt upgrade -y**

**sudo apt install snort -y**

During installation, it will ask for **network interface** → choose the interface you want to monitor (e.g., eth0).

👉 Check Snort version:

**snort -V**

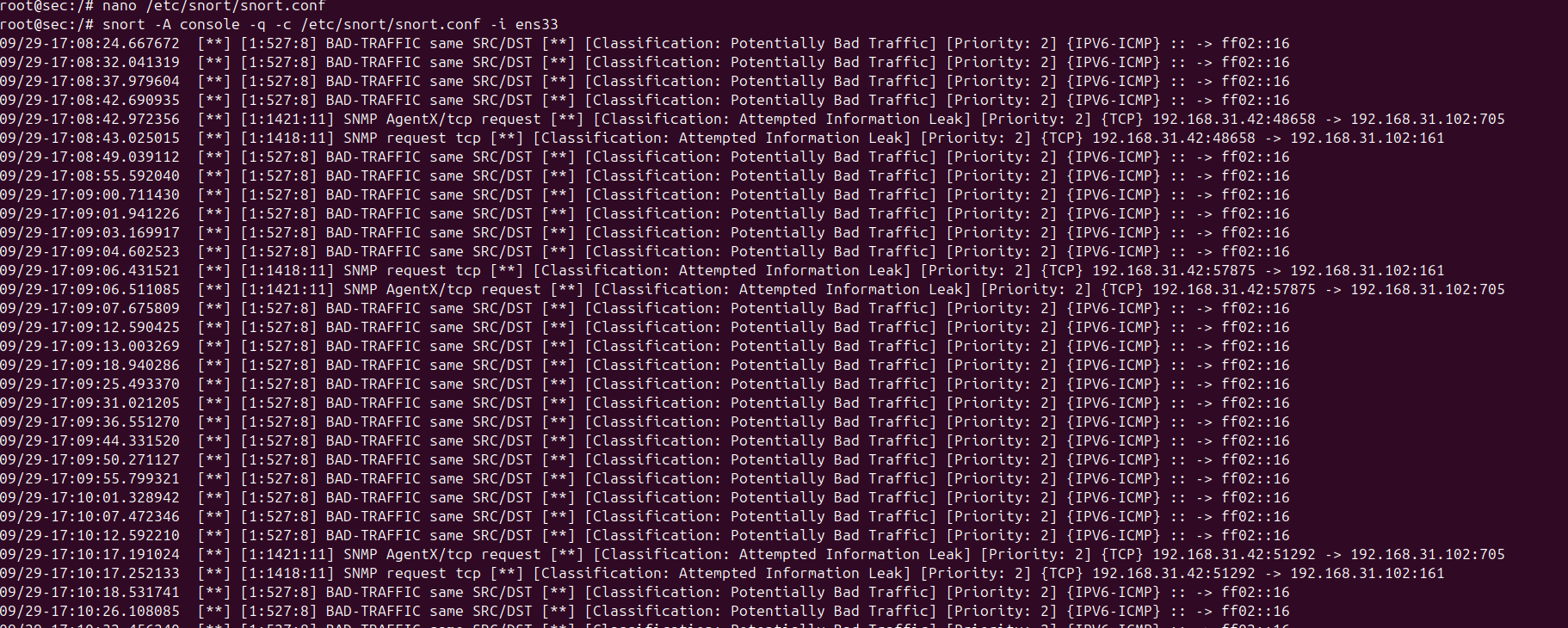
****

**Test Snort in IDS Mode**

Run with default rules:

sudo snort -A console -q -c /etc/snort/snort.conf -i eth0

* -A console → shows alerts on screen
* -q → quiet mode
* -i eth0 → interface



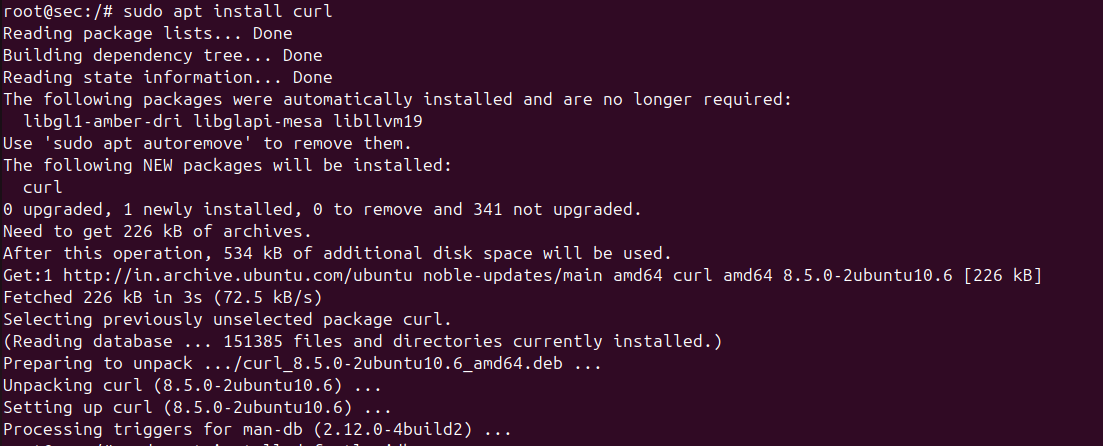
Check for video tutorial

[**snort-setup**](https://www.youtube.com/watch?v=cD-DoKLzq2s)

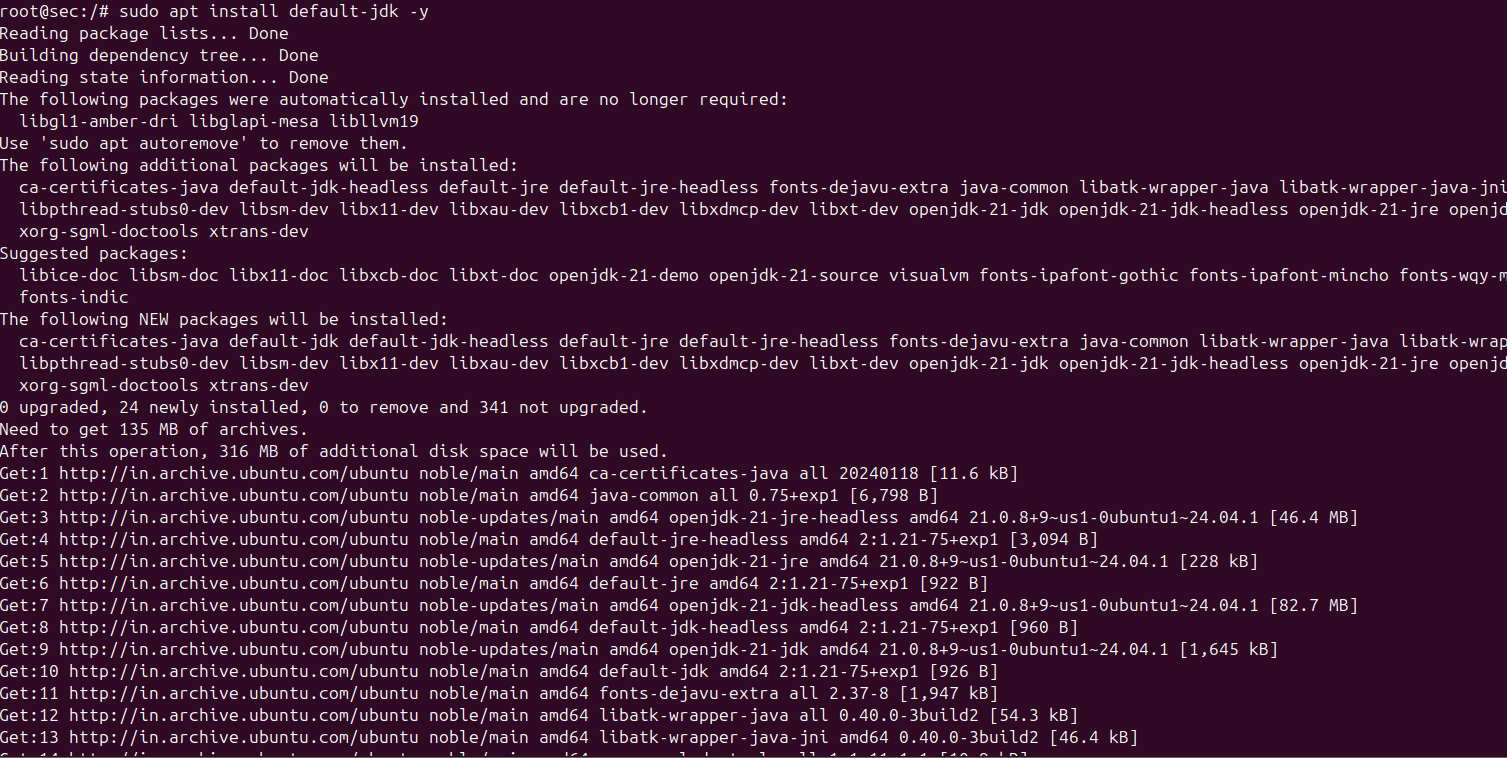
**4.In the Central SIEM Manger setup the ELK stack and wazuh-Manger(Ubuntu).**

1.Wazuh manger setup (Do paste this steps according order)

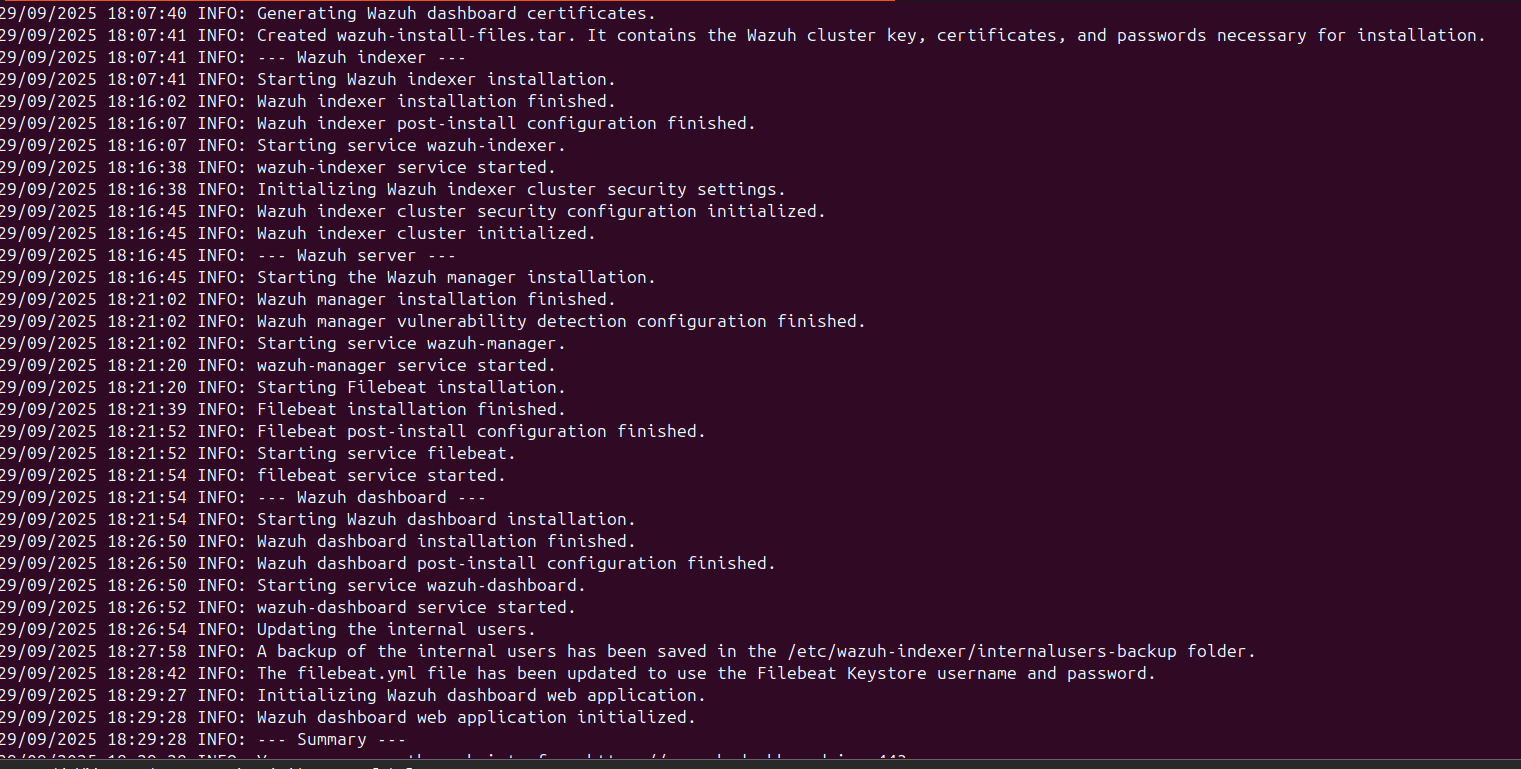
* Sudo apt install curl



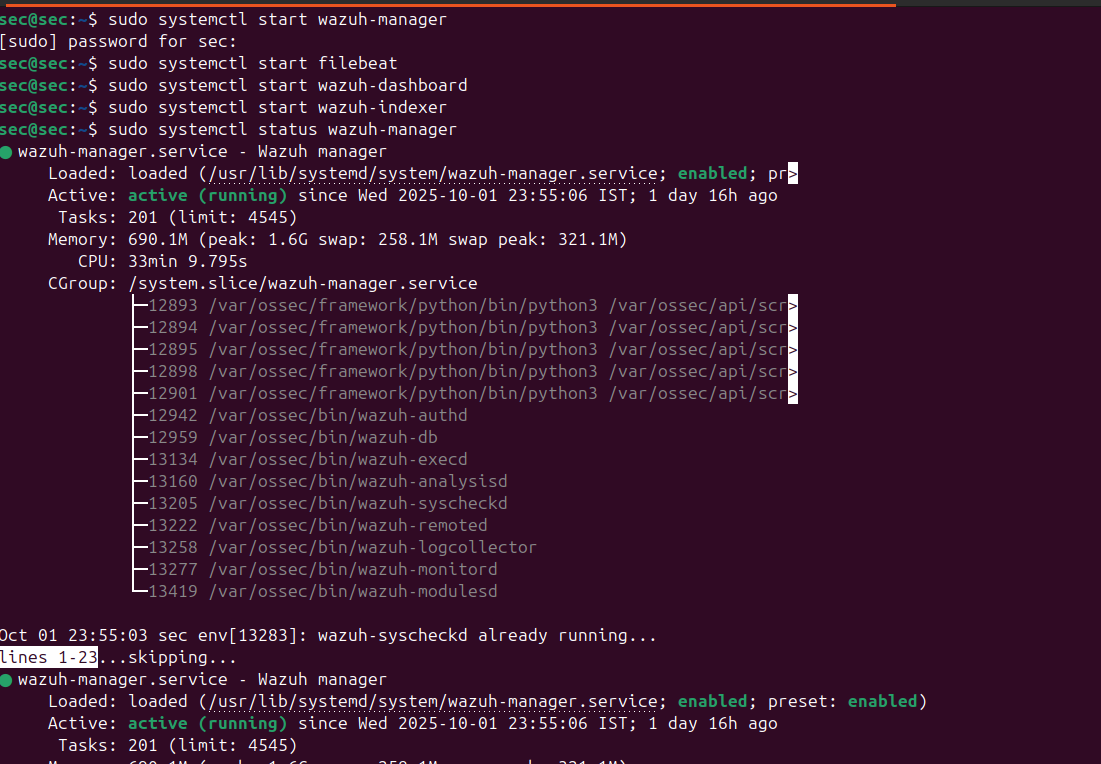
* Sudo apt install default-jdk -y



* (curl -sO <https://packages.wazuh.com/4.13/wazuh-install.sh>)
* bash wazuh-install.sh -a

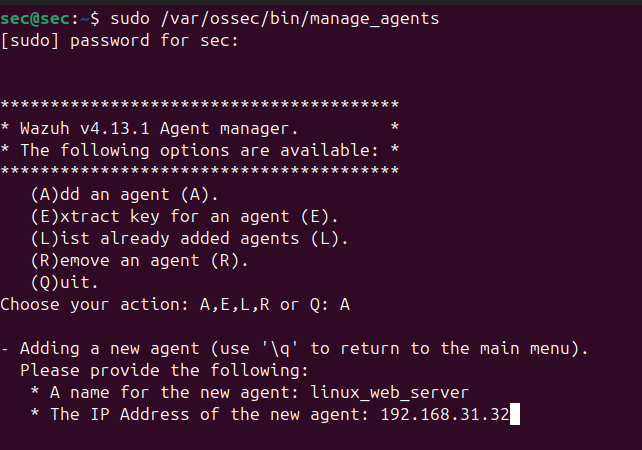


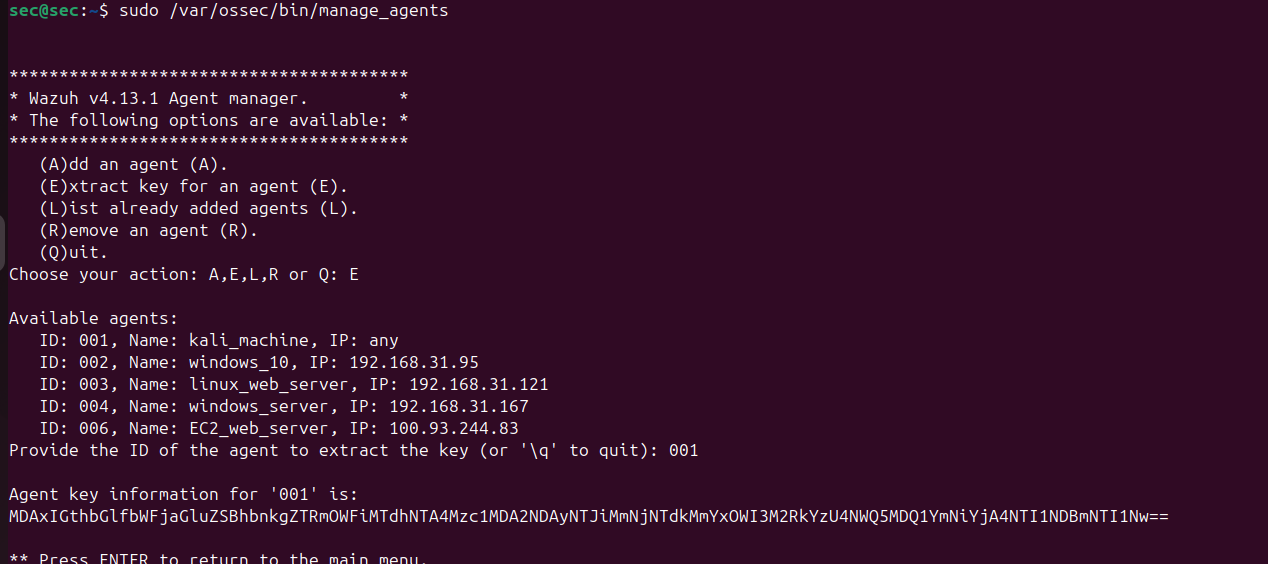
**Starting of the Wazuh-magner**

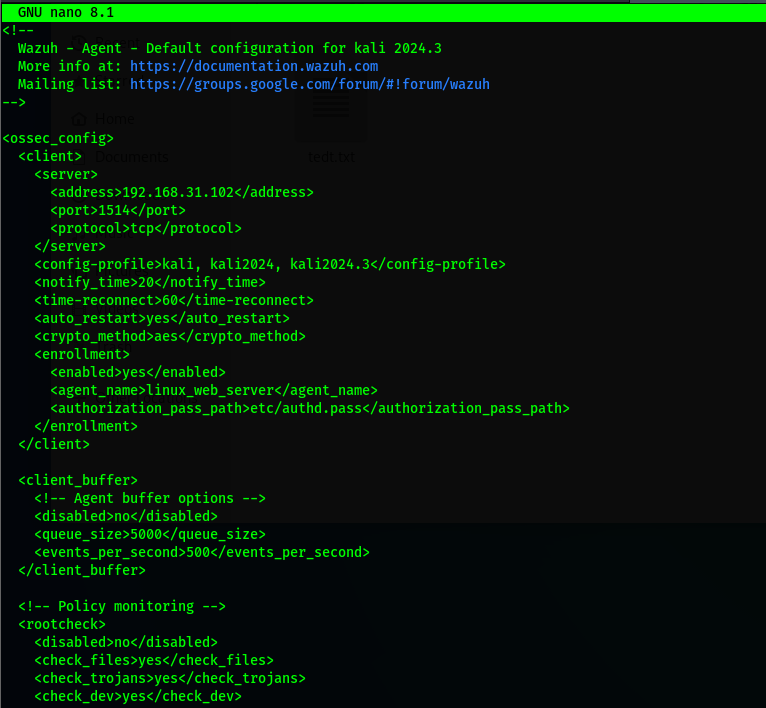
* **sudo systemctl start wazuh-manager**
* **sudo systemctl start filebeat**
* **sudo systemctl start wazuh-dashboard**
* **sudo systemctl start wazuh-indexer**

**Setup of the Wazuh Agent in different OS**

* In Wazuh manger(Central SIEM) in the terminal add this command (sudo /var/ossec/bin/manage\_agents )



* Next in the manger of copy the agent key
* In the OS want to add the wazuh agent (**Configure the Wazuh Agent as Edit the agent config file**
* **sudo nano /var/ossec/etc/ossec.conf**

****

**Replace MANAGER\_IP with your manager’s IP**  
Example: if your Wazuh Manager is at 192.168.1.100

**Save and exit**

* In nano: CTRL+O, then Enter, then CTRL+X

**Restart the agent**

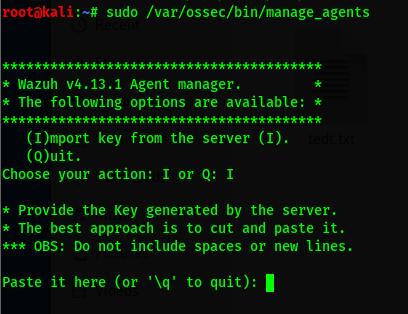
sudo systemctl daemon-reexec

sudo systemctl restart wazuh-agent

sudo systemctl status wazuh-agent

On your **Kali (where the agent is installed)**, the logs are located in:

/var/ossec/logs/ossec.log

now the agent machine paste this command and import the agent key which (**sudo /var/ossec/bin/manage\_agents)** ****

for other type of os refer this ([wazuh-guide-endpoints](https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-windows.html))

**Launching an EC2 Instance with Amazon Linux**

**Prerequisites**

1. **AWS Account** — you need an AWS account with permission to create EC2 instances.
2. **IAM Permissions** — ensure your IAM user has the necessary EC2, VPC, and KeyPair permissions.
3. **Region Selection** — choose an AWS region close to your user base to reduce latency.

**Steps to Launch**

1. **Open EC2 Console**  
   Go to the AWS Management Console → Services → EC2. [Amazon Web Services, Inc.+1](https://aws.amazon.com/ec2/getting-started/?utm_source=chatgpt.com)
2. **Launch Instance Wizard**  
   In the EC2 dashboard, click **“Launch instance”**. [AWS Documentation+2AWS Documentation+2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-wizard.html?utm_source=chatgpt.com)
3. **Name & Tags**  
   Give your instance a descriptive name tag (e.g. web-server, dev-instance). [AWS Documentation+2TechTarget+2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-wizard.html?utm_source=chatgpt.com)
4. **Choose AMI (Amazon Machine Image)**  
   Select Amazon Linux (Amazon Linux 2 or Amazon Linux 2023) as your OS. [AWS Documentation+4AWS Documentation+4AWS Documentation+4](https://docs.aws.amazon.com/linux/al2/ug/ec2.html?utm_source=chatgpt.com)
   * To always get the latest Amazon Linux image in scripts or CLI, you can use the SSM public parameter like /aws/service/ami-amazon-linux-latest/... [AWS Documentation](https://docs.aws.amazon.com/linux/al2/ug/ec2.html?utm_source=chatgpt.com)
5. **Choose Instance Type**  
   Pick a size (for example t2.micro or t3.micro if under the free tier) depending on your workload. [AWS Documentation+2AWS Documentation+2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/LaunchingAndUsingInstances.html?utm_source=chatgpt.com)
6. **Configure Instance Details**
   * Network and Subnet (usually default VPC)
   * IAM role (if your instance needs AWS access)
   * Auto-assign Public IP (if you want it to be reachable from the Internet)
   * Advanced options like user data scripts, shutdown behavior [AWS Documentation+2AWS Documentation+2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-wizard.html?utm_source=chatgpt.com)
7. **Add Storage**  
   By default, Amazon Linux instances come with an EBS volume (often 8 GB). Adjust size or storage type if needed. [AWS Documentation+1](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-wizard.html?utm_source=chatgpt.com)
8. **Configure Security Group**  
   Set inbound rules (for example, allow SSH on port 22, HTTP on port 80, etc.). It’s a firewall layer — be restrictive by default. [AWS Documentation+3Jenkins+3AWS Documentation+3](https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/?utm_source=chatgpt.com)
9. **Key Pair for SSH Access**  
   Create a new key pair (or use an existing one) — you’ll download a .pem file. This is needed to SSH into your instance. [AWS Documentation+2AWS Documentation+2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-wizard.html?utm_source=chatgpt.com)  
   Make sure to keep your private key safe and set proper permissions (chmod 400 key.pem on Linux/Mac). [AWS Documentation+1](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html?utm_source=chatgpt.com)
10. **Review & Launch**  
    Review all configurations. If everything is acceptable, click **Launch instance** to start it. [AWS Documentation+2AWS Documentation+2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance-wizard.html?utm_source=chatgpt.com)
11. **Connect to the Instance via SSH**
    * Go to the EC2 console → Instances → select your instance → **Connect**. [AWS Documentation+1](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html?utm_source=chatgpt.com)
    * Use the SSH command (on Linux/macOS) such as:
    * ssh -i /path/to/key.pem ec2-user@your-instance-public-dns

(ec2-user is the default for Amazon Linux) [AWS Documentation+2AWS Documentation+2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html?utm_source=chatgpt.com)

* + On Windows, you can use tools like PuTTY (convert the .pem to .ppk first) to SSH in.
  + Before connecting, set private key permissions: chmod 400 key.pem (Linux/macOS). [AWS Documentation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html?utm_source=chatgpt.com)

1. **Terminate or Stop Instance (when done)**  
   When you finish your work, stop or terminate the instance to avoid unnecessary charges

Reference guid youtube video([EC2\_ creation](https://youtu.be/fy2x8DgjwX8))

Now need to connect cloud host with manger,need to setup the VPN service in the both

**Install Tailscale(VPN)**

**curl -fsSL https://tailscale.com/install.sh | sh**

Enable and start the service:

**sudo systemctl enable --now tailscaled**

**🔹 Step 3: Generate an Auth Key (on your laptop/desktop browser)**

1. Go to <https://login.tailscale.com/admin/settings/keys>
2. Click **Generate auth key** (choose *Reusable* if you’ll need it often).
3. Copy the key (it looks like tskey-xxxxxxxxxxxxxxxx).

**🔹 Step 4: Authenticate EC2 instance with the key**

On EC2 terminal:

**sudo tailscale up --authkey tskey-xxxxxxxxxxxxxxxx**

This bypasses the browser login and directly registers your EC2 instance into your Tailscale network.

**🔹 Step 5: Verify Tailscale is working**

**tailscale status**

**tailscale ip -4**

You should see your EC2 listed with a **100.x.x.x IP**.

**🔹 Step 6: Connect Manager (Private Machine) also to Tailscale**

On your private machine (Wazuh Manager host):

sudo tailscale up

(or use --authkey again).

Now both the **EC2 agent** and the **Wazuh Manager** are in the same VPN, reachable by their **Tailscale IPs**.

**🔹 Step 7: Configure Wazuh Agent (on EC2) to talk to Manager**

Edit the agent config:

**sudo nano /var/ossec/etc/ossec.conf**

Find:

<client>

<server>

<address>MANAGER\_IP</address>

<port>1514</port>

<protocol>tcp</protocol>

</server>

</client>

👉 Replace MANAGER\_IP with your **Wazuh Manager’s Tailscale IP (100.x.x.x)**.

Restart agent:

sudo systemctl restart wazuh-agent

**🔹 Step 8: Approve the Agent on Manager**

On your manager node:

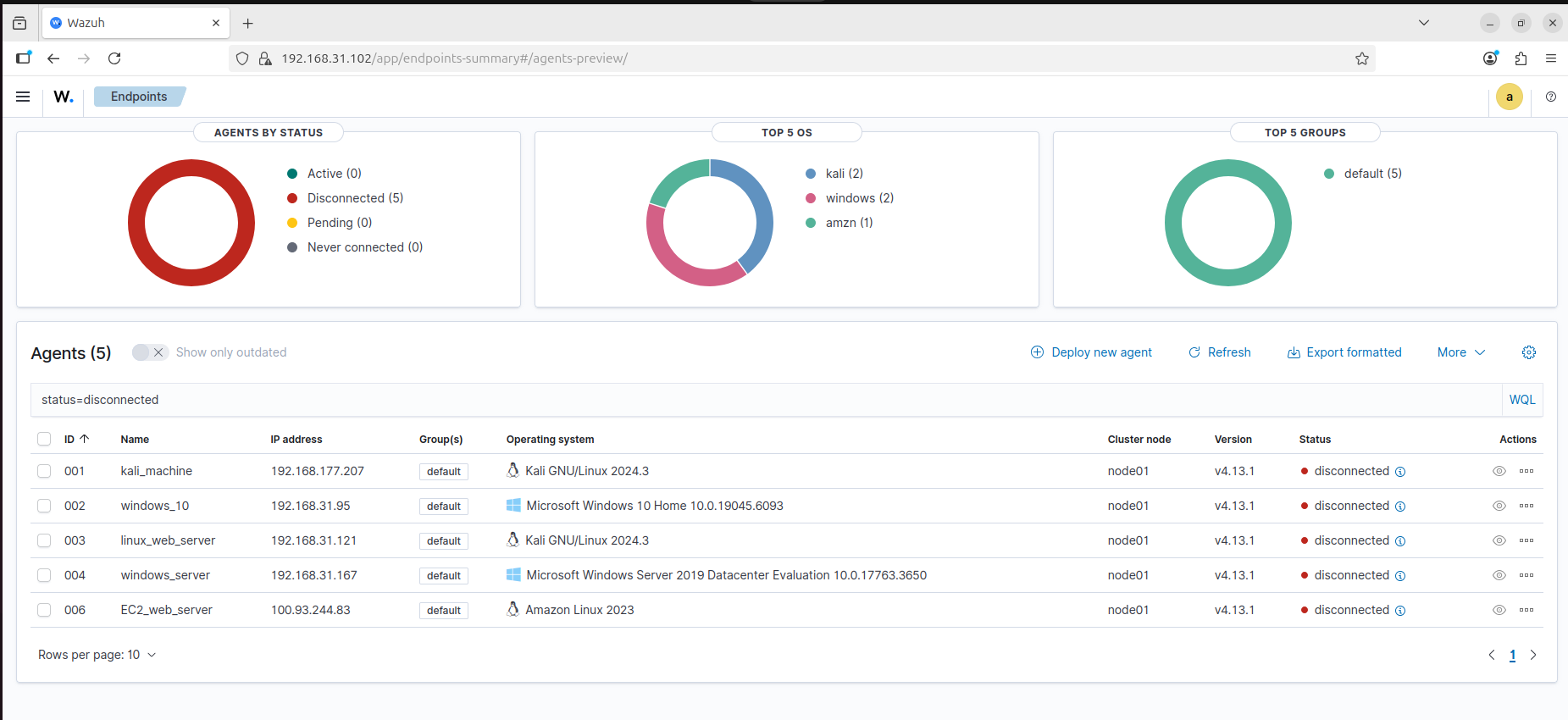
**sudo /var/ossec/bin/manage\_agents**

* Add the new agent, copy the key.
* On EC2, import the key with:
* **sudo /var/ossec/bin/manage\_agents**

→ Choose **Import key** and paste it.

Restart services on both sides.

This how wazuh dashboard looks like

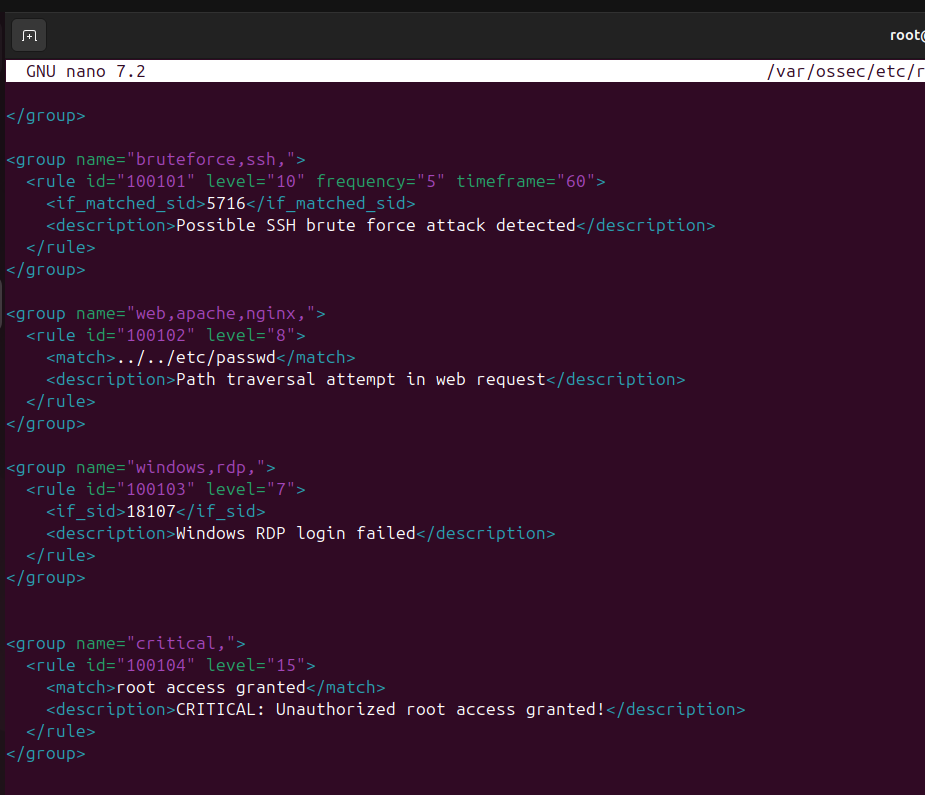


**6.Make detection rules in the central SIEM .**

**Steps to add custom alerts for all agents**

1. SSH into your **Wazuh Manager**.
2. Open the local rules file:
3. **sudo nano /var/ossec/etc/rules/local\_rules.xml**
4. Paste your custom rules inside <group name="local,"> ... </group> block.  
   Example:
5. **<group name="local,">**
6. **<rule id="100001" level="7">**
7. **<if\_sid>5712</if\_sid>**
8. **<description>Multiple SSH authentication failures (possible brute force)</description>**
9. **<group>authentication, sshd, brute\_force</group>**
10. **</rule>**
11. **</group>**
12. Save and exit.
13. Restart the Wazuh manager to apply changes:

**Sudo systemctl restart wazuh-manager**

****

**7.Simluating the attacks to check the detection rules are working and making a ticket in the slack**

**webhook setup of the Slack for wazuh**

1.Go to the slack and create a channel([slack](https://app.slack.com/))

2.Now go the slack api setup ([slack-api](https://api.slack.com/))

3.create new app and add that channel to receive the alerts

4.In features field in left hand side bar go to incoming webhooks([slack-webhook](https://api.slack.com/apps/A09JQ2T2FMJ/incoming-webhooks?))

5. On toggle on the right hand side ,now scroll down and click on the **add new webhook** button.

6. In the wazuh-manager go this file (/var/ossec/etc/ossec.conf)

Then paste this lines

**<integration>**

**<name>slack</name>**

**<hook\_url>https://hooks.slack.com/services/YOUR/SLACK/WEBHOOK</hook\_url>**

**<level>10</level>**

**<alert\_format>json</alert\_format>**

**</integration>**

At hookurl paste your webhook url

