v-Soc installation and configuration manual

Montserrat Flores Castillo

05/09/2025

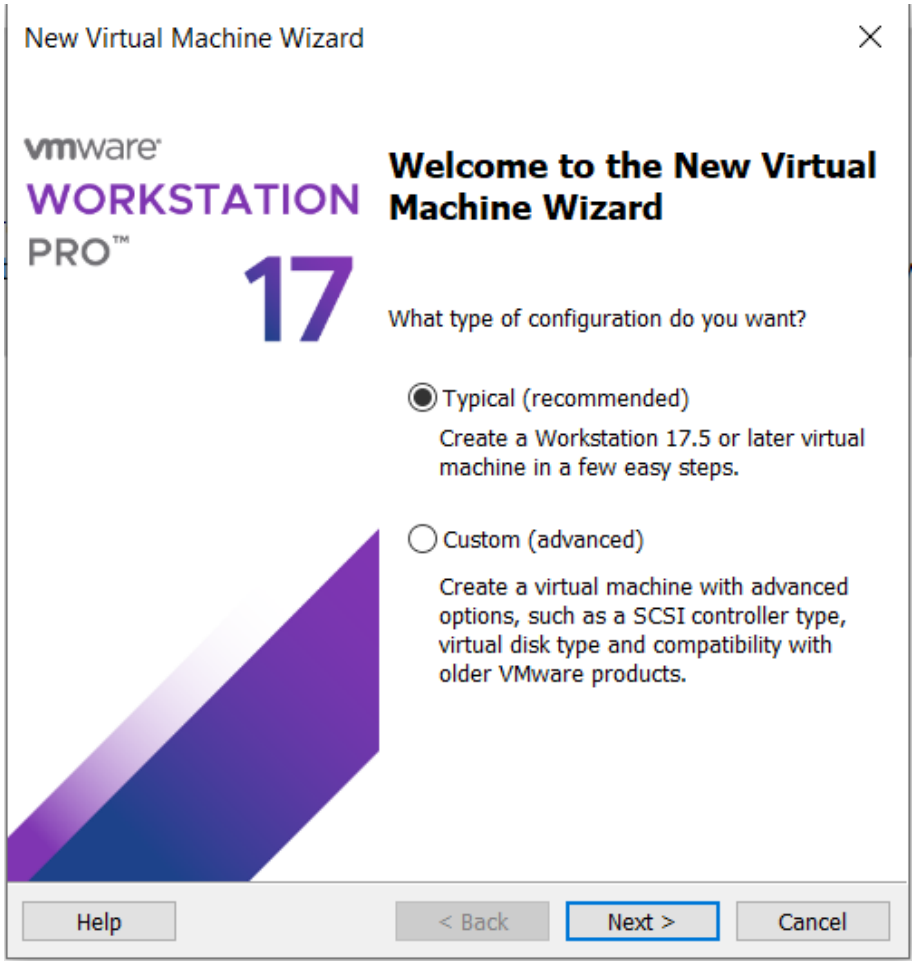
Contents

1. Virtual Machine creation and configuration in VMware

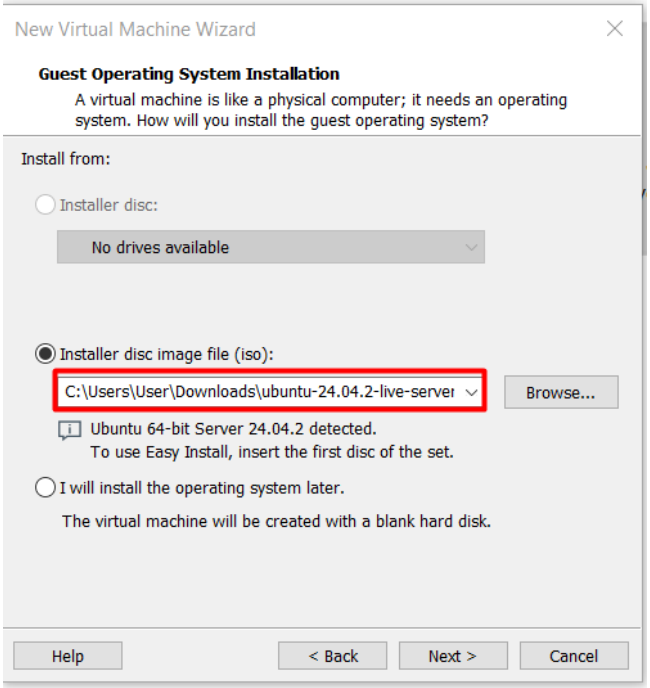
Link for download the software: https://blogs.vmware.com/workstation/2024/05/vmware-workstation-pro-now-available-free-for-personal-use.html

For be available to download the VMware pro workstation you will need to create account, and then you will be able to download it.

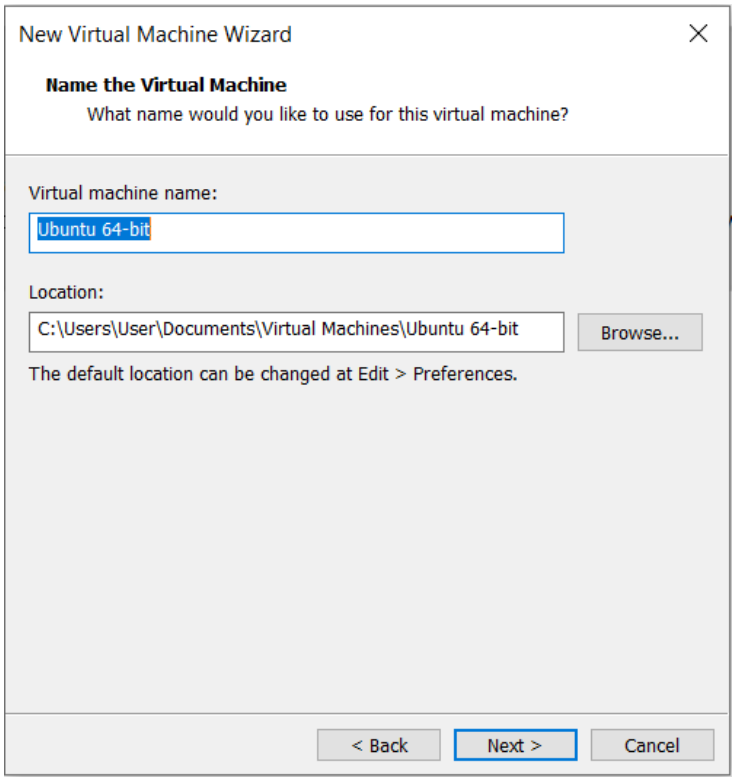
This is the wizard for creating virutal machines in VMware, normally you will use the typical recommended configuration.



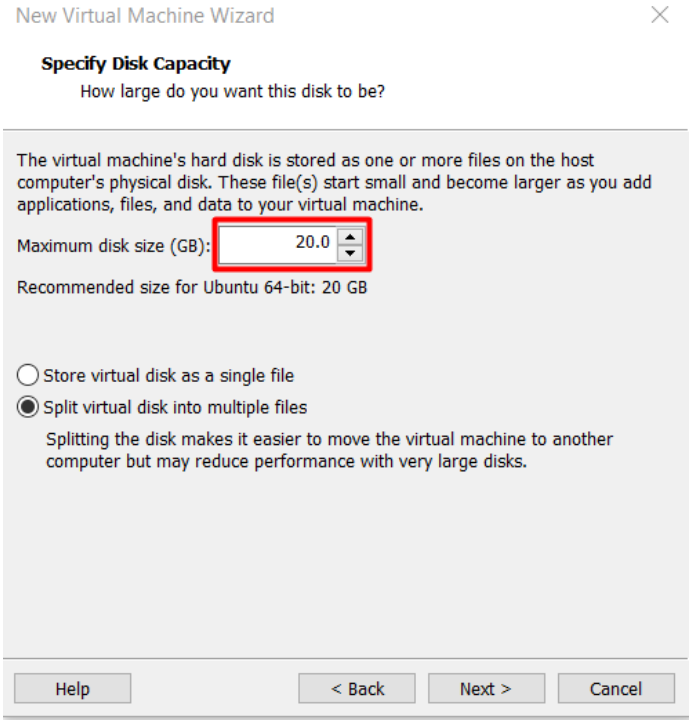
Choose the directory of your ISO image with the Ubuntu OS for example.



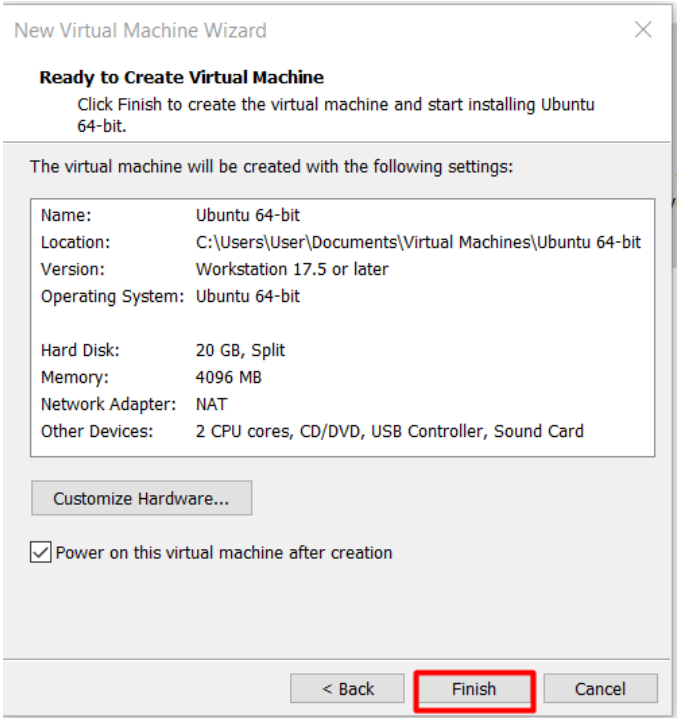
Add a name to your VM and choose the location where is going to be located.



Change the size of your disk capacity to the needed one, for example: 120 GB



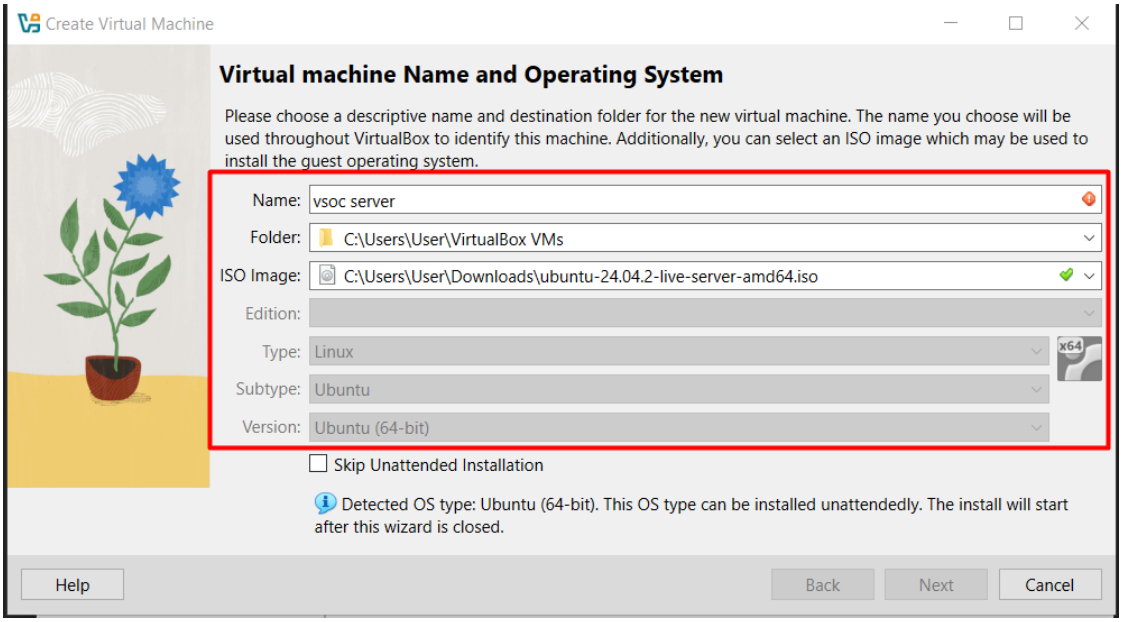
Now you are in the resume page, where you can change any configuration before finishing creating the VM.



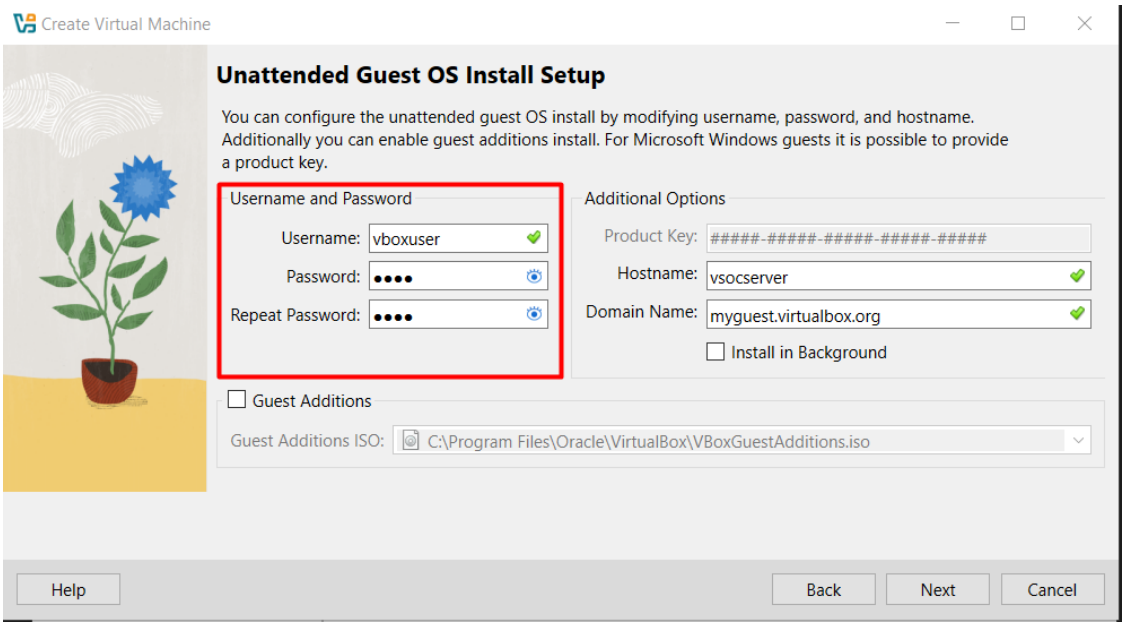
2. Virtual Machine creation and configuration in VirtualBox

Link for download VirtualBox: https://www.virtualbox.org/wiki/Downloads

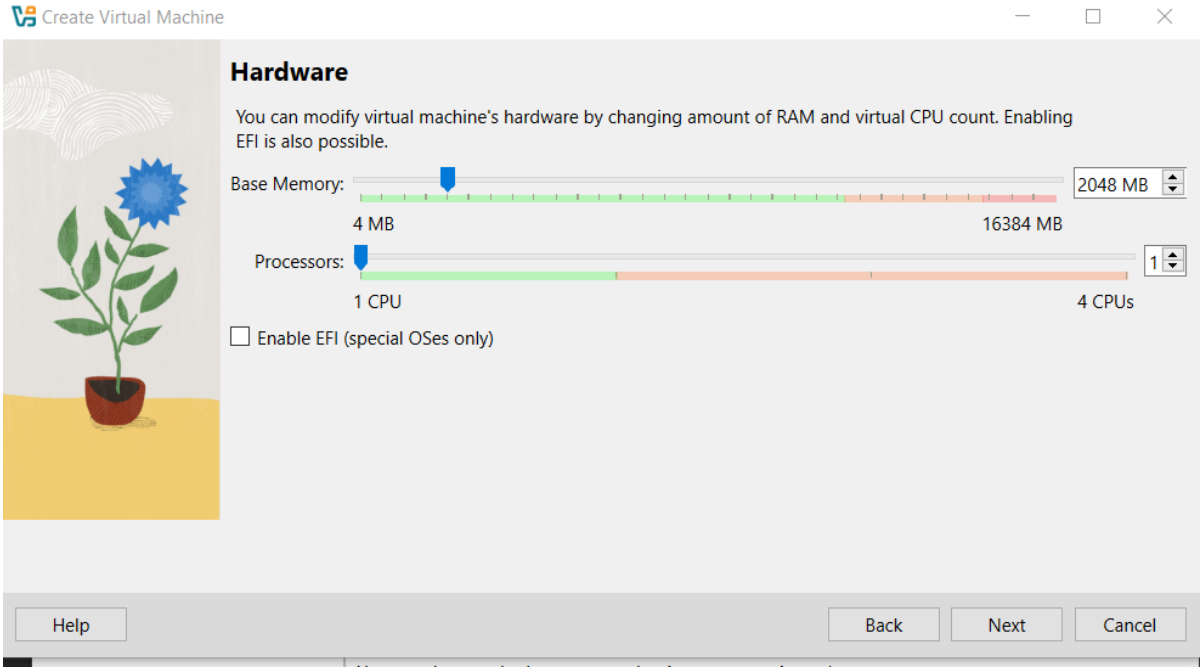
In the vm creation process here we must indicate the name of the machine, where we want to locate it and the ISO of the OS.



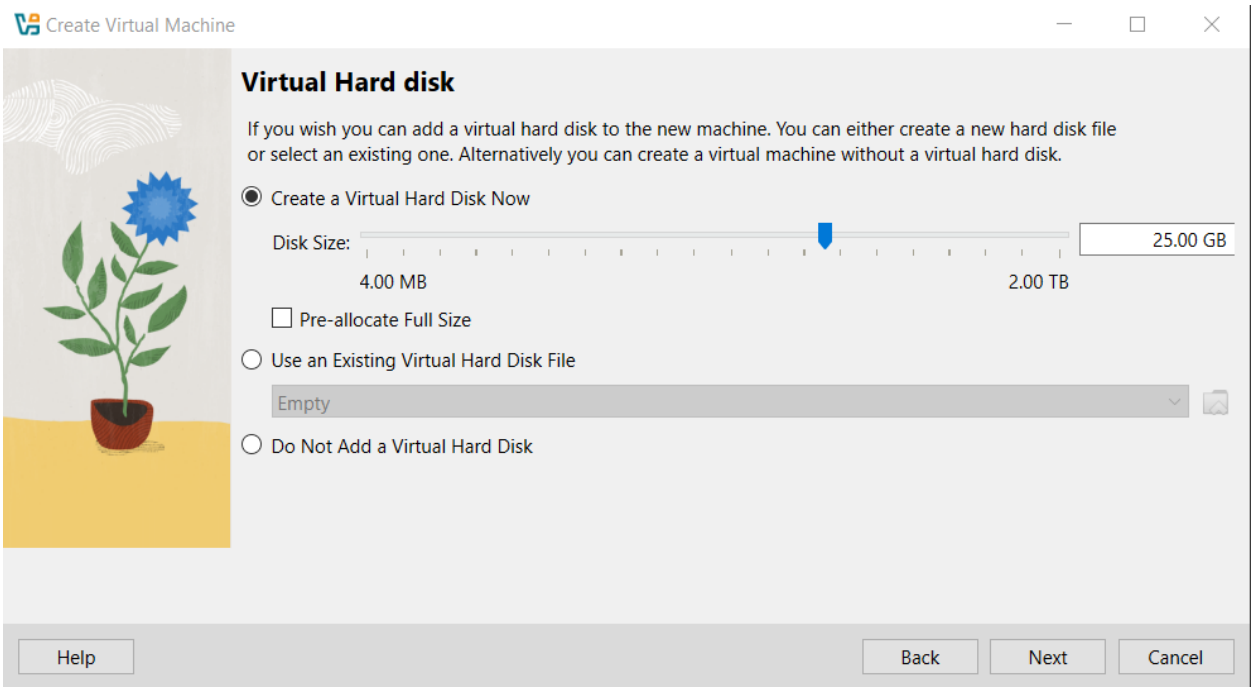
Indicate the username for the root user and the password.



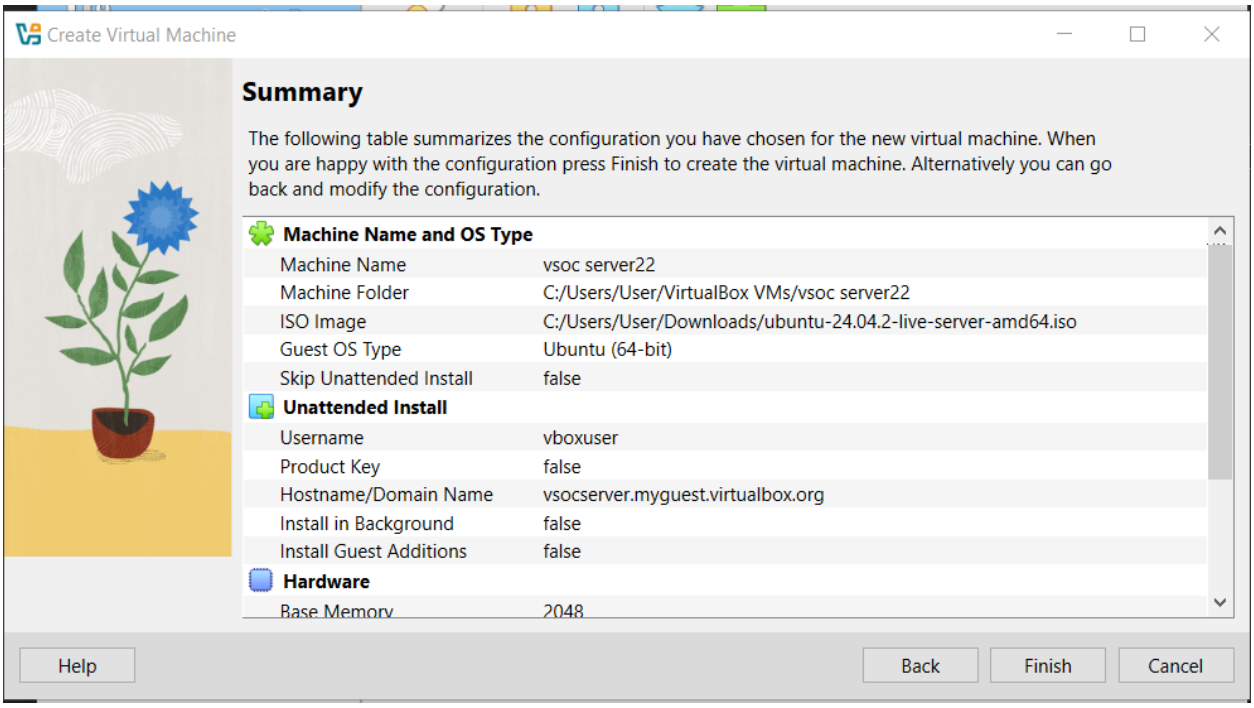
Here we need to indicate the RAM and CPU we want to use in the VM



Indicate the size of the disk for the VM.

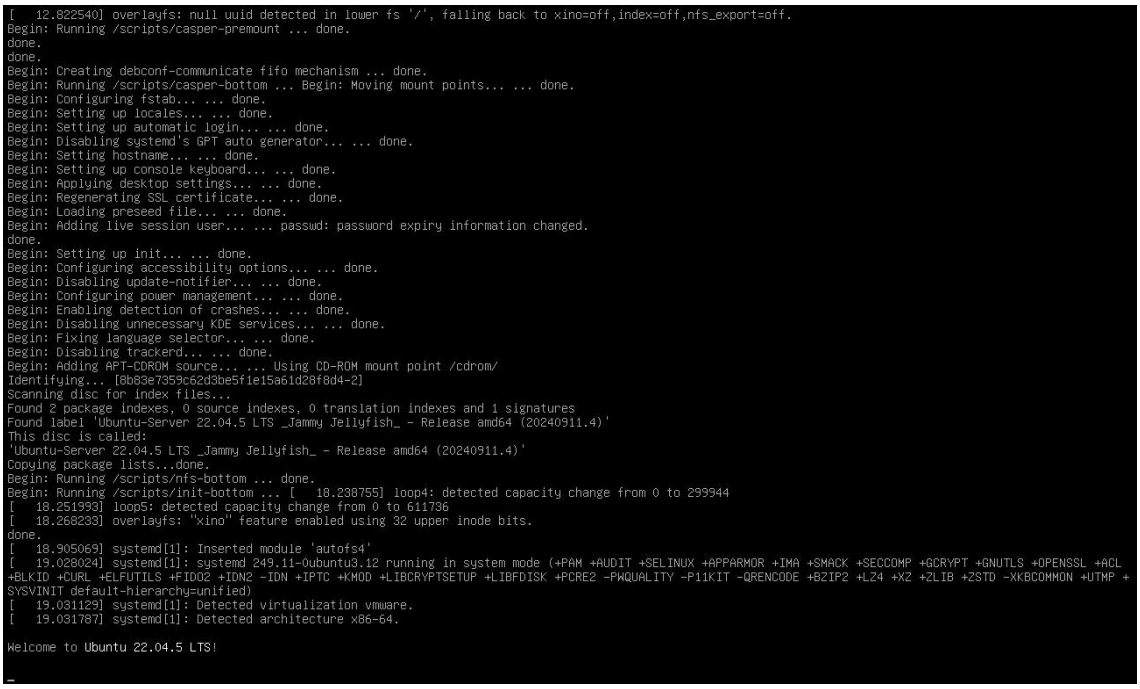


Summary of your VM creation in VirtualBox.



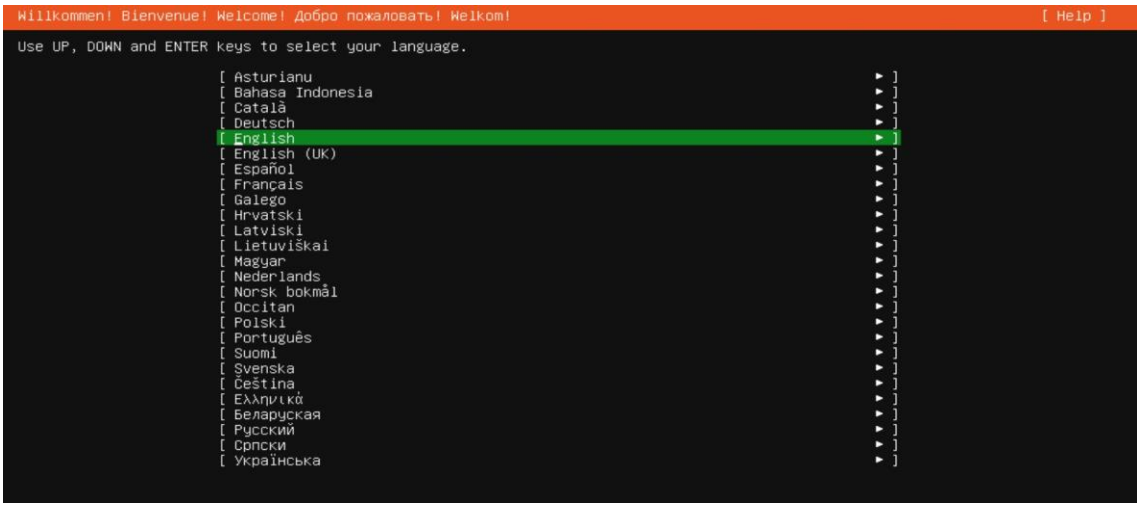
3. Installation of the OS (Ubuntu Server)

At the beginning of the installation we will see this prompt which can take couple minutes until it pop you up the installation of the operating system.

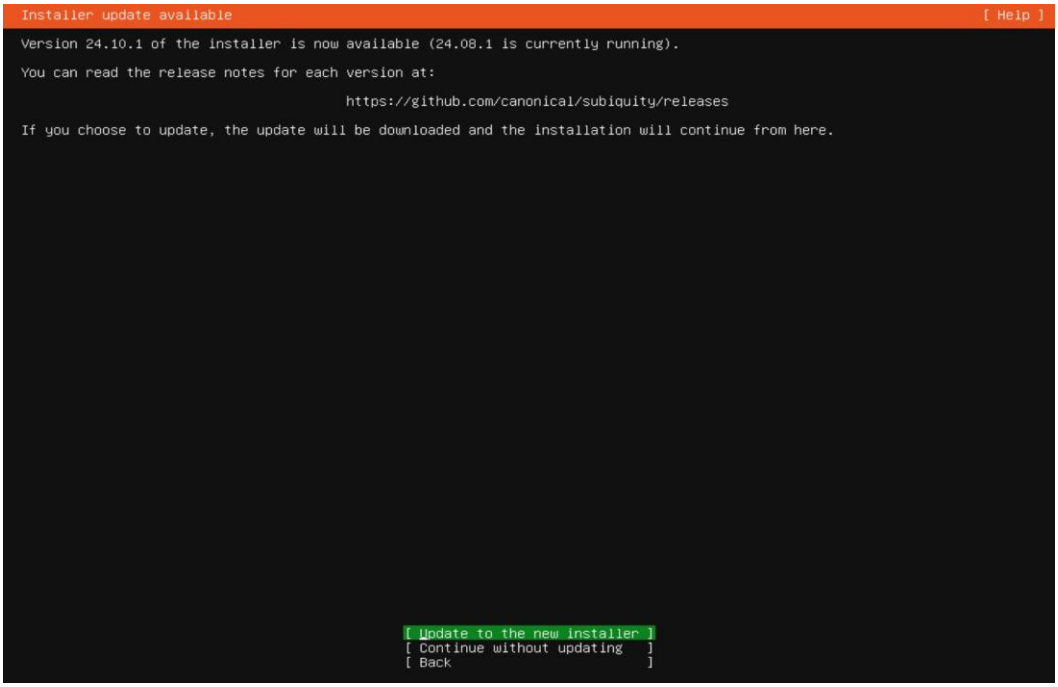


```
[   12.822540] overlayfs: null uuid detected in lower fs '/', falling back to xino=off,index=off,nfs_export=off.
Begin: Running /scripts/casper-premount ... done.
done.
done.
Begin: Creating debconf-communicate fifo mechanism ... done.
Begin: Running /scripts/casper-bottom ... Begin: Moving mount points... ... done.
Begin: Configuring fstab... ... done.
Begin: Setting up locales... ... done.
Begin: Setting up automatic login... ... done.
Begin: Disabling systemd's GPT auto generator... ... done.
Begin: Setting hostname... ... done.
Begin: Setting up console keyboard... ... done.
Begin: Applying desktop settings... ... done.
Begin: Regenerating SSL certificate... ... done.
Begin: Loading preseed file... ... done.
Begin: Adding live session user... ... passwd: password expiry information changed.
done.
Begin: Setting up init... ... done.
Begin: Configuring accessibility options... ... done.
Begin: Disabling update-notifier... ... done.
Begin: Configuring power management... ... done.
Begin: Enabling detection of crashes... ... done.
Begin: Disabling unnecessary KDE services... ... done.
Begin: Fixing language selector... ... done.
Begin: Disabling trackerd... ... done.
Begin: Adding APT-CDROM source... ... Using CD-ROM mount point /cdrom/
Identifying... [8b83e7359c62d3be5f1e15a61d28f8d4-2]
Scanning disc for index files...
Found 2 package indexes, 0 source indexes, 0 translation indexes and 1 signatures
Found label 'Ubuntu-Server 22.04.5 LTS _Jammy Jellyfish_ - Release amd64 (20240911.4)'
This disc is called:
'Ubuntu-Server 22.04.5 LTS _Jammy Jellyfish_ - Release amd64 (20240911.4)'
Copying package lists...done.
Begin: Running /scripts/nfs-bottom ... done.
Begin: Running /scripts/init-bottom ... [   18.238755] loop4: detected capacity change from 0 to 299944
[   18.251993] loop5: detected capacity change from 0 to 611736
[   18.268233] overlayfs: ''xino'' feature enabled using 32 upper inode bits.
done.
[   18.905069] systemd[1]: Inserted module 'autofs4'
[   19.028024] systemd[1]: systemd 249.11-0ubuntu3.12 running in system mode (+PAM +AUDIT +SELINUX +APPARMOR +IMA +SMACK +SECCOMP +GCRYPT +GNUTLS +OPENSSL +ACL
+BLKID +CURL +ELFUTILS +FIDO2 +IDN2 -IDN +IPTC +KMOD +LIBCRYPTSETUP +LIBFDISK +PCRE2 -PWQUALITY -P11KIT -QRENCODE +BZIP2 +LZ4 +XZ +ZLIB +ZSTD -XKBCOMMON +UTMP +
SYSVINIT default-hierarchy=unified)
[   19.031129] systemd[1]: Detected virtualization vmware.
[   19.031787] systemd[1]: Detected architecture x86-64.

Welcome to Ubuntu 22.04.5 LTS!

_
```

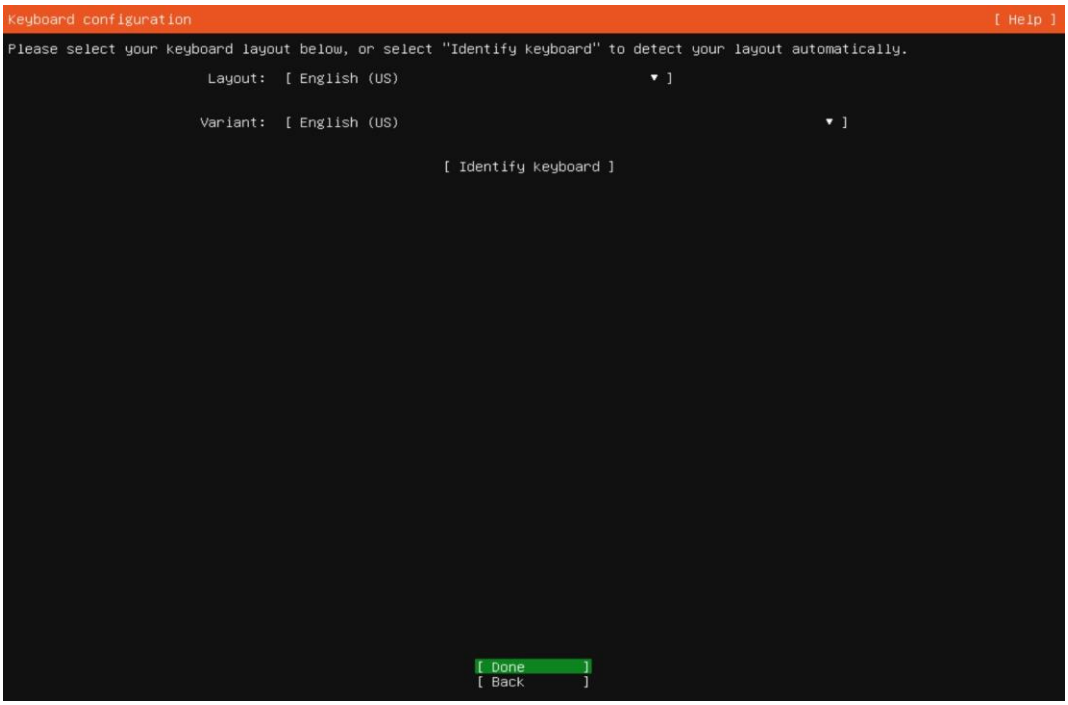
In this section we will need to choose the language of the system:



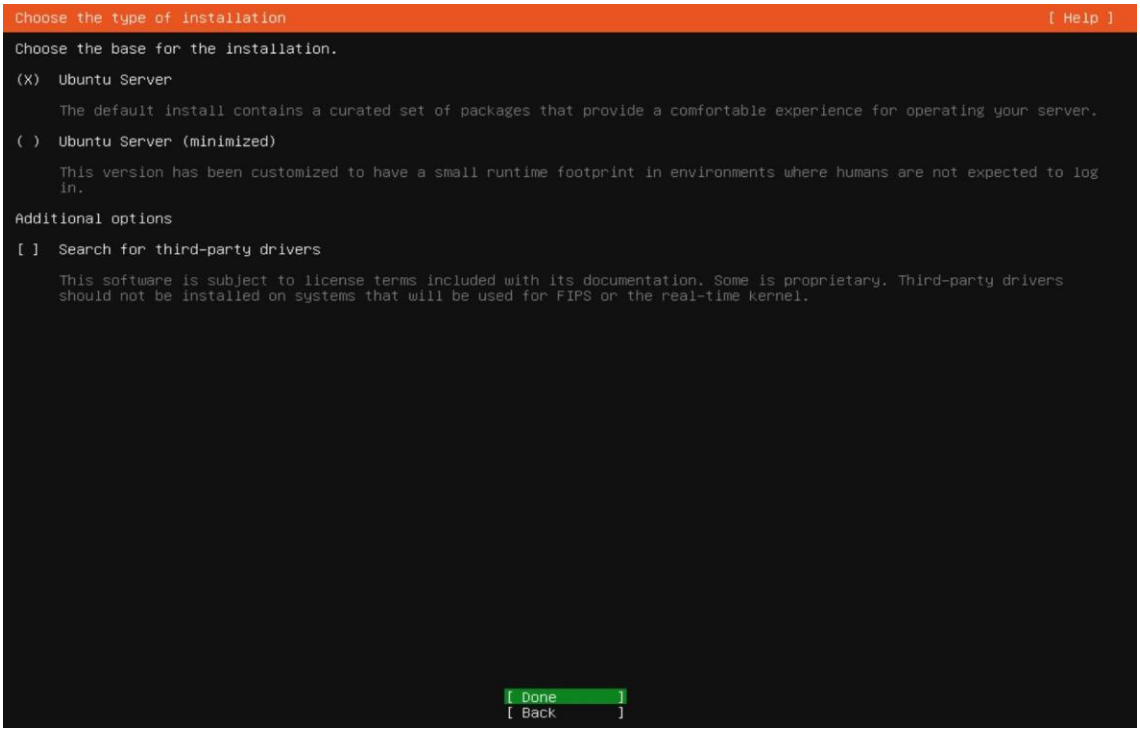
Indicates we have a new update available so is recommended to install the operating system with the newest version:



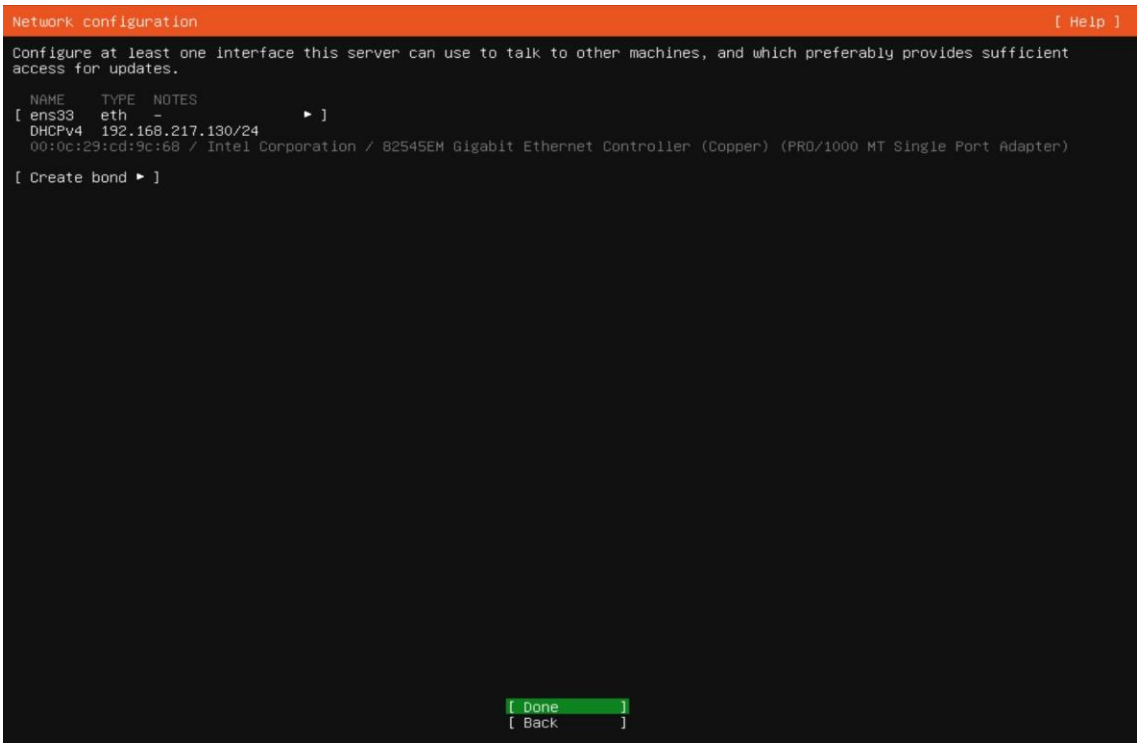
Choose the language of the keyboard and layout of the keyboard:



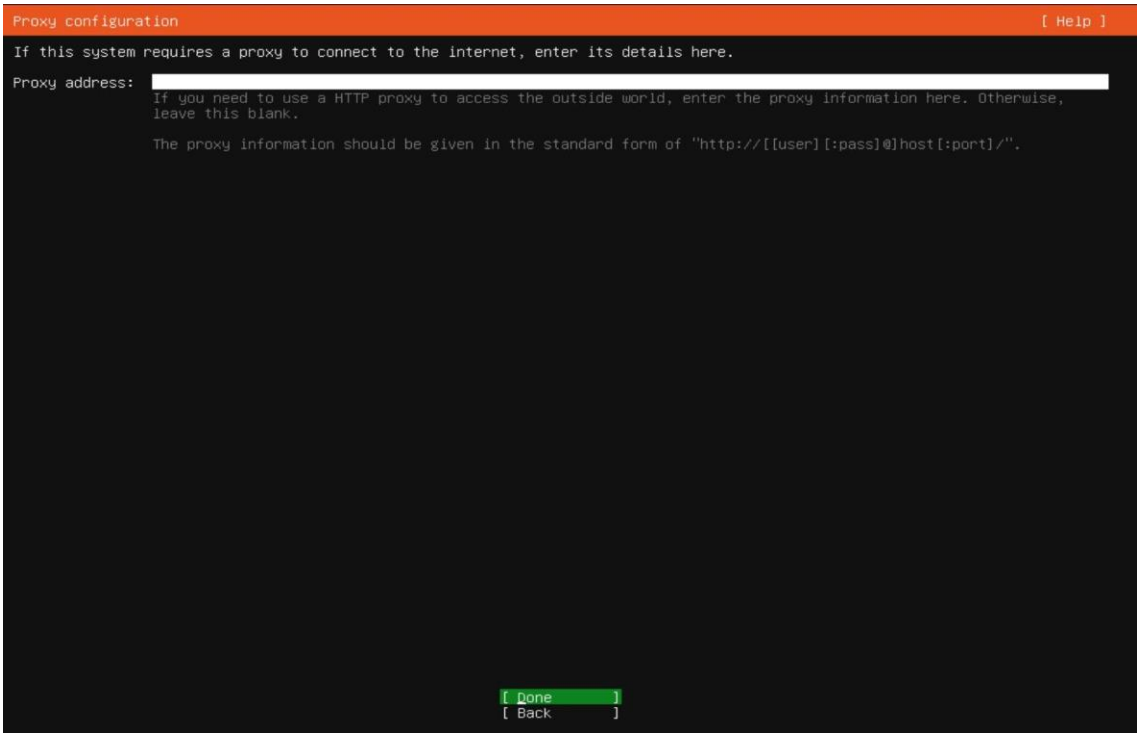
Here we indicate which kind of installation of Ubuntu Server we want, in our case the normal installation of Ubuntu Server:



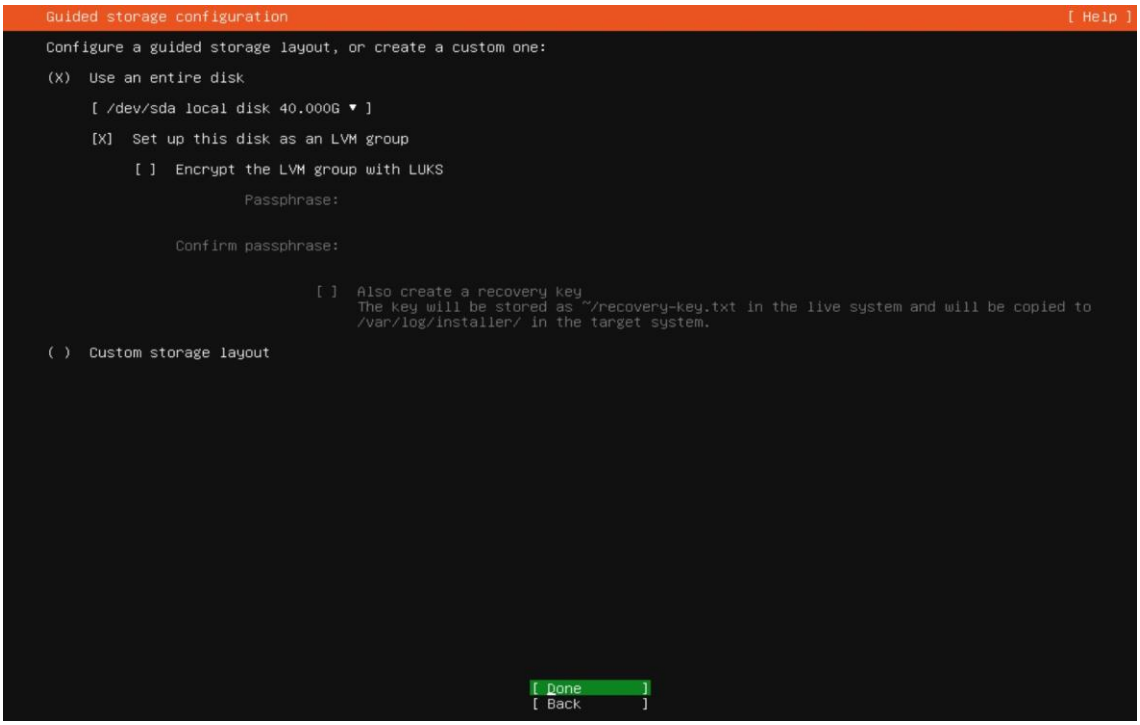
The network configuration we leaved in the default configuration until we need to configure it in the future:



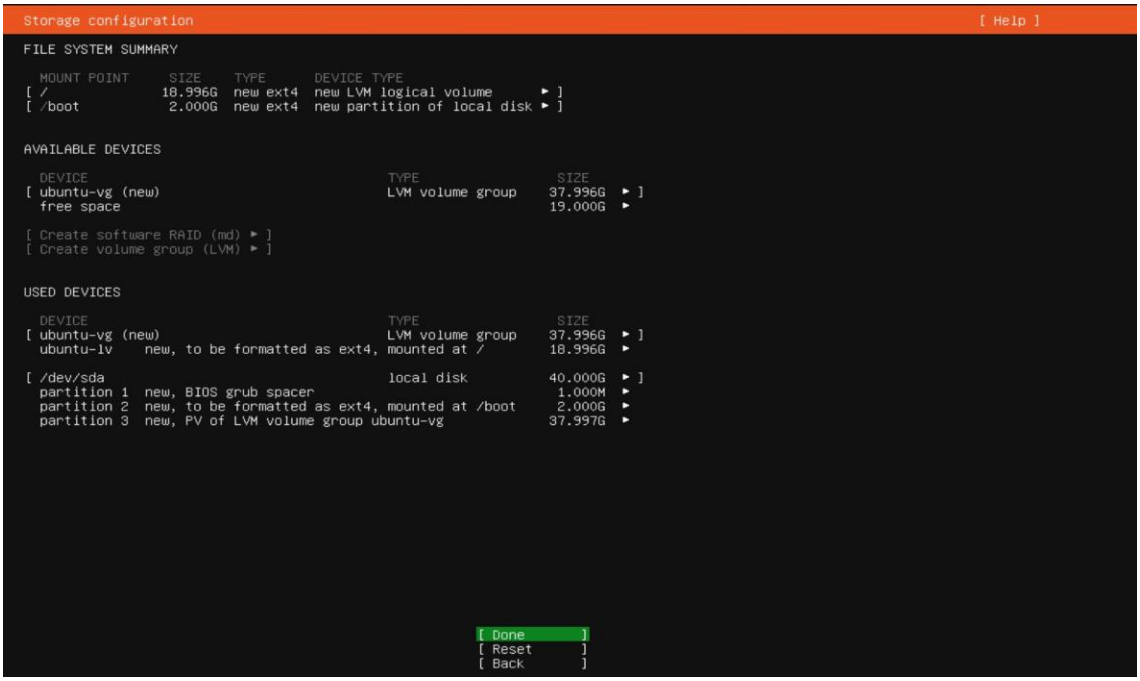
We left in the default configuration empty until we configure our own proxy:



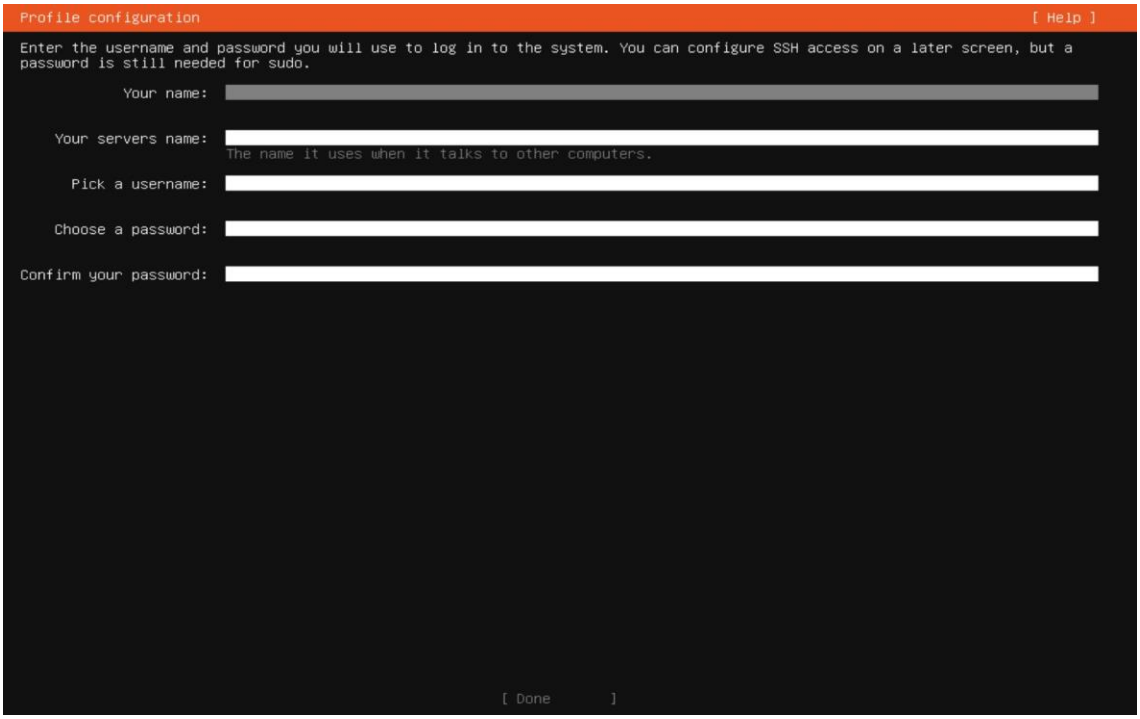
The storage configuration we leave by default here we can change where we want to locate the installation of the ubuntu server.



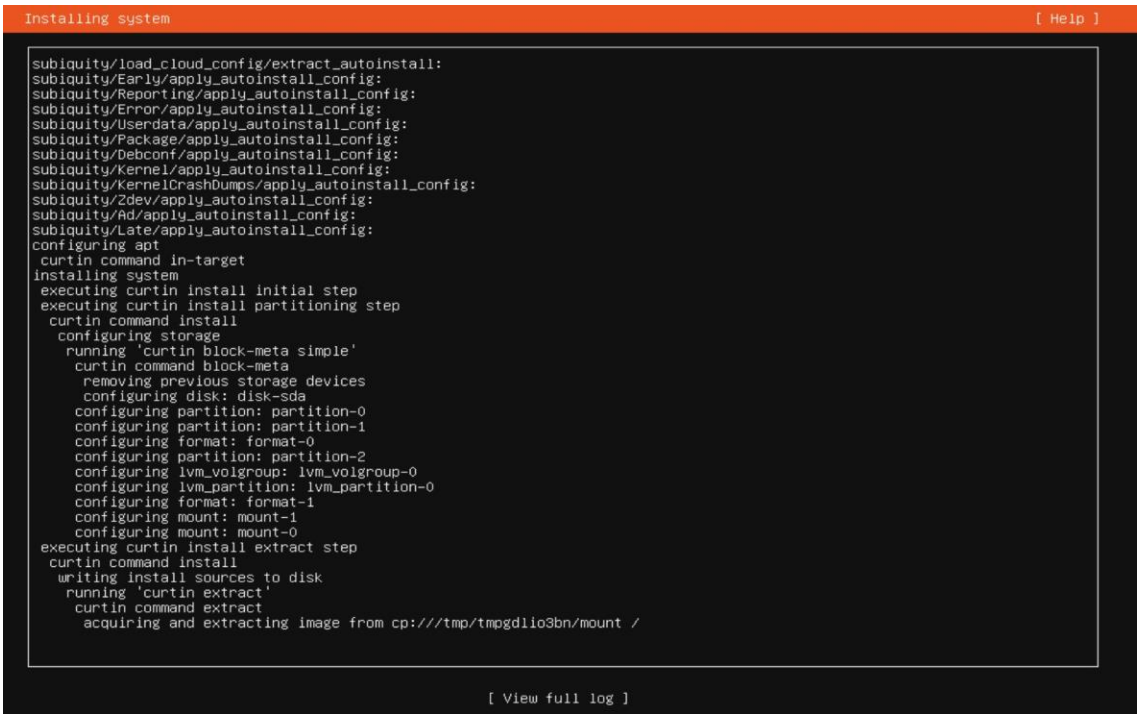
More extra configuration we can leave in default, like where is locate the /boot or the root system /:



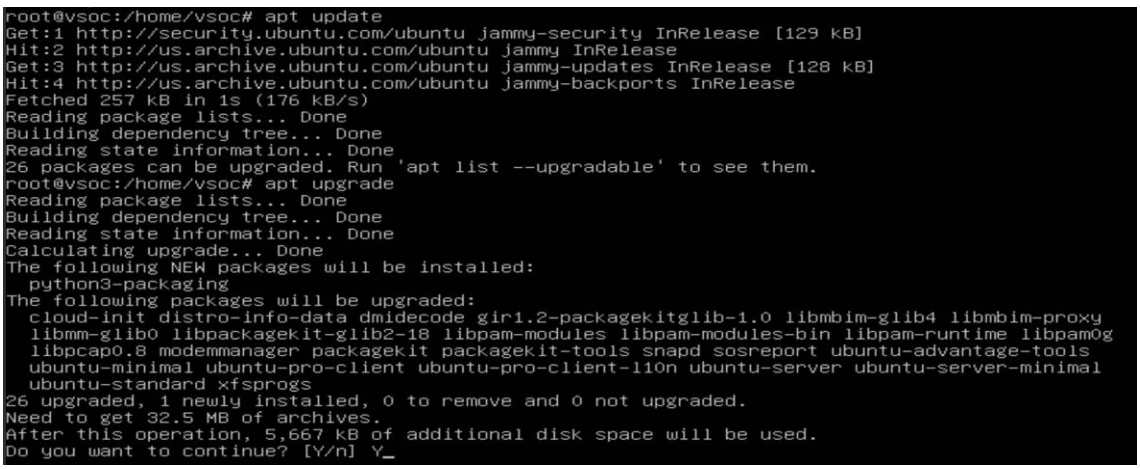
Creation of superuser and user of your ubuntu server.



Installation process:



After installing the operating system is always recommended to check for last updates and upgrades. Using the command: (sudo apt update, sudo apt upgrade):



Now we are going to installing the GUI version in your ubuntu server.

Command: *sudo apt install tasksel dialog*

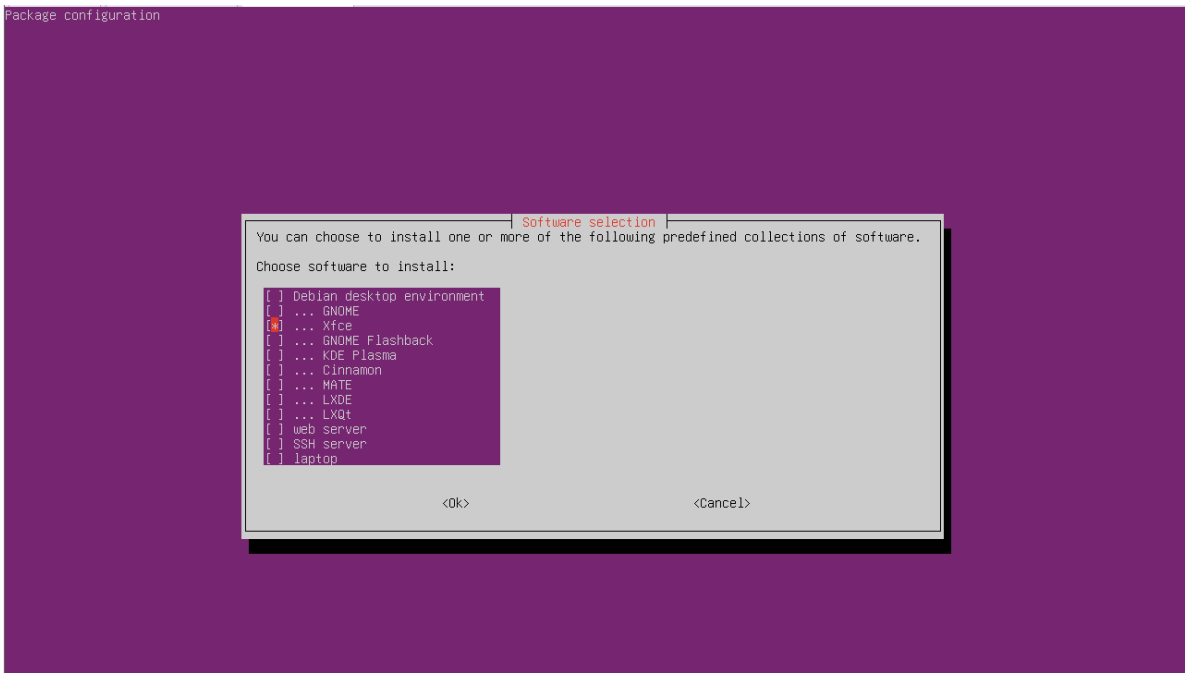
After installing the taksel dialog which is a program for launch an interface version inside of your CLI. Now we will execute this program.

sudo tasksel

Command:

In the menu of options of desktop environments, we will choose XFCE, because is lightweight and less resource consuming.

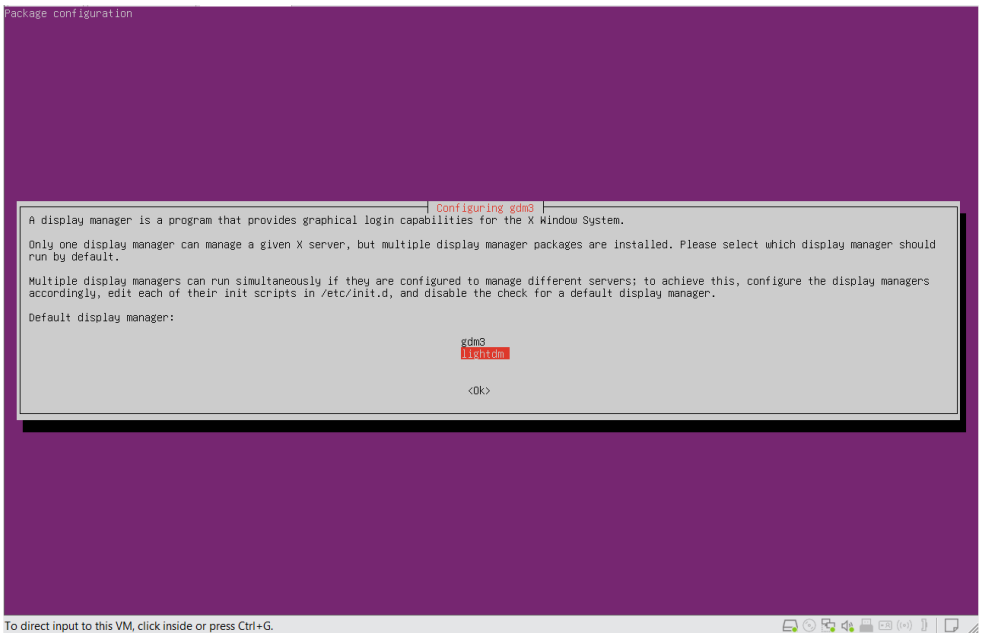
Tip: For move in the menu use your arrows in the keyboard and press space for mark your option. For move to the OK use the TAB and press Enter.



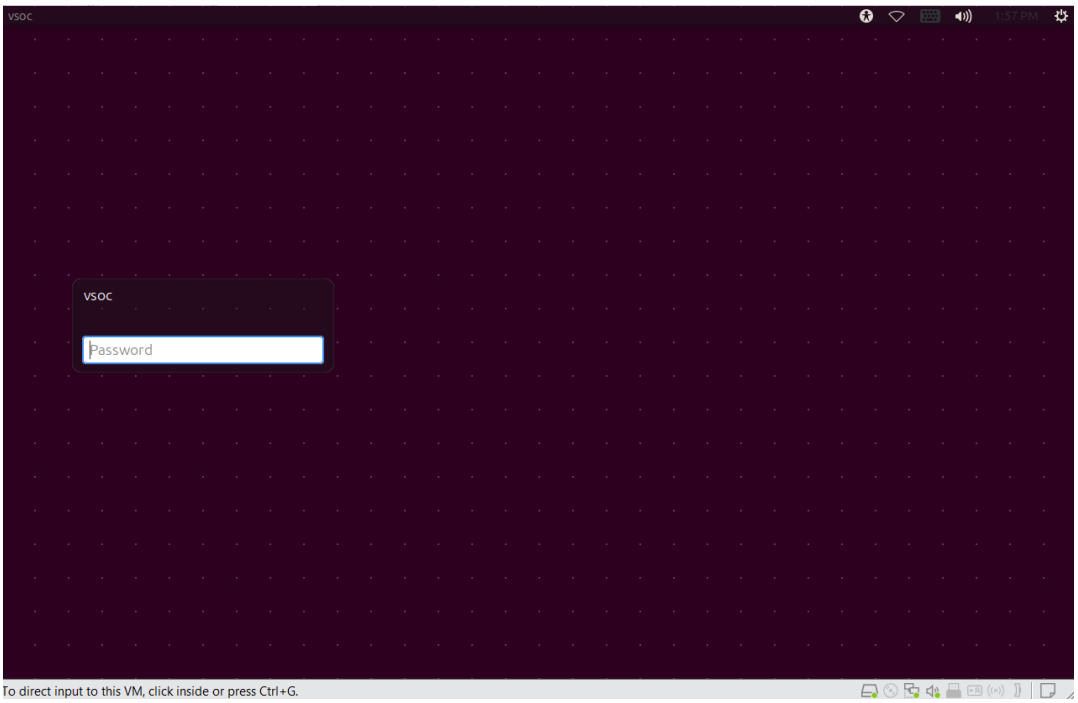
Now after installing the xfce desktop , we need to execute this command for install all the necessaries requeriments for ubuntu server have a GUI.

Command: *sudo apt-get install ubuntu-desktop*

Choose the lightdm option for graphical option.



After installing the desktop **RESTART** your virtual machine to activate the GUI.



4. Ansible installation step-by-step

For install Ansible we need first to install the common software properties and add the repository for install the Ansible package.

Command

sudo apt install software-properties-common

sudo add-apt-repository –yes –update ppa:ansible/ansible

sudo apt install ansible

```
vsoc@vsoc:~$ sudo apt install software-properties-common
[sudo] password for vsoc:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.49.2).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
vsoc@vsoc:~$
```

```
vsoc@vsoc:~$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
'
Description:
Ansible is a radically simple IT automation platform that makes your application
s and systems easier to deploy. Avoid writing scripts or custom code to deploy a
nd update your applications— automate in a language that approaches plain Englis
h, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu noble/main Icons (48x48) [106 kB]
```

```
vsoc@vsoc:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-kerberos python3-ntlm-auth python3-requests-ntlm
  python3-resolvelib python3-winrm python3-xmltodict sshpass
The following NEW packages will be installed:
  ansible ansible-core python3-kerberos python3-ntlm-auth
  python3-requests-ntlm python3-resolvelib python3-winrm python3-xmltodict
  sshpass
0 upgraded, 9 newly installed, 0 to remove and 1 not upgraded.
Need to get 20.1 MB of archives.
After this operation, 228 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Now we need to get the last version of the file configuration of Ansible:

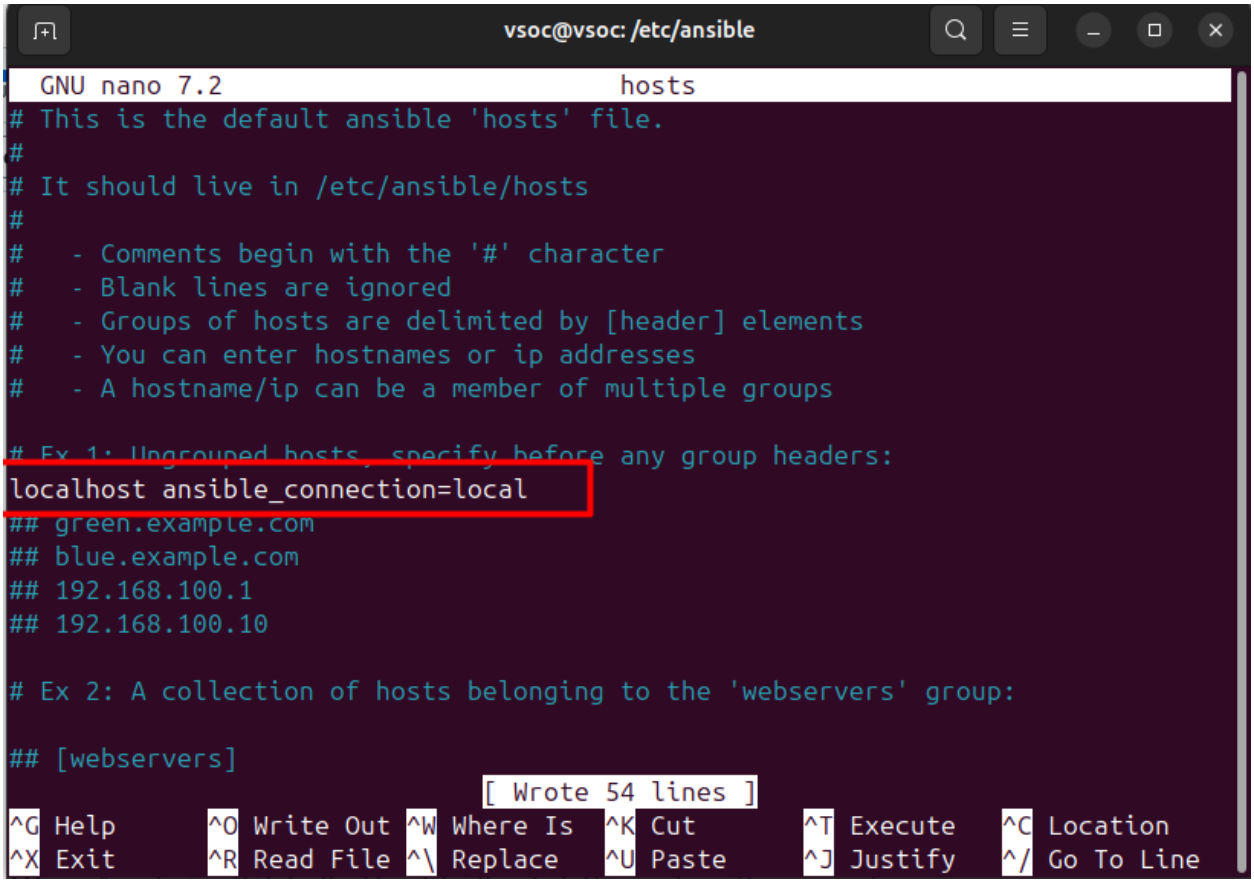
Command: *ansible-config init –disabled > ansible.cfg*

```
vsoc@vsoc:~$ ansible-config init --disabled > ansible.cfg
vsoc@vsoc:~$
```

Add this lines of code, for configure the ansible configuration file located in /etc/ansible/ansible.cfg

```
vsoc@vsoc: /etc/ansible
  GNU nano 7.2                    ansible.cfg *
# Also you can now have a more complete file by including existing plugins:
# ansible-config init --disabled -t all > ansible.cfg

# For previous versions of Ansible you can check for examples in the 'stable' b>
# Note that this file was always incomplete  and lagging changes to configurati>

# for example, for 2.9: https://github.com/ansible/ansible/blob/stable-2.9/exam>
[default]
inventory = /etc/ansible/hosts
remote_user = vsoc
host_key_cheking = False
retry_files_enabled = False
interpreter_python = auto_silent
[privilege_escalation]
become = True
become_method = sudo
become_user = root
become_ask_pass = False
[connection]
type = local

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

Now we need to indicate inside of the file hosts inside of the Ansible directory that we want to do our installations locally. Add the line inside of the /etc/ansible/hosts file.



5. Firewall UFW installation and configuration

The UFW is already installed by default in your Linux OS, just need to be configured and enabled.

Command:

sudo ufw default deny incoming

sudo ufw default allow outgoing

```
vsoc@vsoc:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
vsoc@vsoc:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Now our UFW Firewall can be activated and add the rules.

Command:

sudo ufw enable

sudo ufw allow 9200/tcp | sudo ufw allow 9300/tcp | sudo ufw allow 5000/tcp | sudo ufw allow 5044/tcp | sudo ufw allow 5061/tcp | sudo ufw allow 80/tcp | sudo ufw allow 443/tcp | sudo ufw allow 9600/tcp | sudo ufw allow 22/tcp | sudo ufw allow 5040/tcp

```
vsoc@vsoc:~$ sudo ufw enable
Firewall is active and enabled on system startup
vsoc@vsoc:~$ sudo ufw status
Status: active
vsoc@vsoc:~$ sudo ufw allow 9200,9300,5000,5044,5601,80,443,22,514,9600/tcp
Rule added
Rule added (v6)
```

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

To                         Action      From
--                         ------      ----
22,80,443,514,5000,5044,5601,9200,9300,9600/tcp ALLOW       Anywhere

22,80,443,514,5000,5044,5601,9200,9300,9600/tcp (v6) ALLOW       Anywhere (v6)
```

Command:

sudo ufw reload

sudo ufw logging high

6. Snort installation and configuration

For install snort we are going to use the ansible playbook named as "snort.yml". This file must be located inside of /etc/ansible/ and you need to be in that path.

Command: *sudo ansible-playbook -c local snort.yml*

Now you need to configure snort with the alert's necessaries, and functions for detecting traffic in the network.

Command: *sudo nano /etc/snort/rules/local.rules*

```
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.
drop tcp any any -> any any (msg:"Blocking known malicious IP"; sid:1000002; rev:1;)
drop icmp any any -> any any (msg:"Blocking ICMP (Ping)"; sid:1000003; rev:1;)
drop tcp any any -> any 80 (msg:"Fake AV Domain Access"; content:"Host: fakeav.example.c
eader; sid:1000006; rev:1;)
drop tcp any any -> any 23 (msg:"Telnet Attempt Blocked"; sid:1000008; rev:1;)
#alert icmp any any -> any any (msg:"ICMP Packet Detected"; sid:1000001; rev:1;)
# Alert on any TCP traffic (for testing purposes)
alert tcp any any -> any any (msg:"Alert: TCP traffic detected"; sid:1000002; rev:1;)

# Alert on any ICMP traffic (e.g., ping)
alert icmp any any -> any any (msg:"Alert: ICMP packet detected"; sid:1000003; rev:1;)

# Alert on SSH connections
alert tcp any any -> any 22 (msg:"Alert: SSH connection attempt"; sid:1000004; rev:1;)

# Alert on HTTP traffic (port 80)
alert tcp any any -> any 80 (msg:"Alert: HTTP traffic detected"; sid:1000005; rev:1;)

# Alert on suspicious DNS queries (UDP port 53)
#alert udp any any
```

Att: Also file local.rules is provided for download and substitute directly.

Save the file and restart the services of snort for applying the rules and check if it works properly.

Command:

sudo systemctl enable snort

sudo systemctl restart snort

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

```
Commencing packet processing (pid=12762)
05/09-09:26:54.849941  [**] [1:1000002:1] Alert: TCP traffic detected [**] [Priority: 0]
20.177.193:443 -> 192.168.159.133:35486
05/09-09:26:56.845518  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.1 -> 192.168.159.133
05/09-09:26:56.845565  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.133 -> 192.168.159.1
05/09-09:26:57.856870  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.1 -> 192.168.159.133
05/09-09:26:57.856907  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.133 -> 192.168.159.1
05/09-09:26:58.878256  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.1 -> 192.168.159.133
05/09-09:26:58.878460  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.133 -> 192.168.159.1
05/09-09:26:59.895903  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.1 -> 192.168.159.133
05/09-09:26:59.895950  [**] [1:1000003:1] Alert: ICMP packet detected [**] [Priority: 0]
.168.159.133 -> 192.168.159.1
^C*** Caught Int-Signal
^Z
[1]+  Stopped                 sudo snort -A console -c /etc/snort/snort.conf -i ens33
```

7. Syslog-ng installation and configuration

For install syslog-ng we are going to use the ansible playbook named as "syslog-ng.yml". This file must be located inside of /etc/ansible/ and you need to be in that path.

Command: *sudo ansible-playbook -c local syslog-ng.yml*

Now you need to configure syslog-ng for detect logs from UFW Firewall and Snort.

Command: *sudo nano /etc/syslog-ng/syslog-ng.conf*

Att: Also, file syslog-ng.conf is provided for download and substitute directly.

```
# Send UFW logs to Fluentd
log {
    source(s_ufw_log);
    filter(f_ufw);
    destination(d_fluentd_ufw);
};
# Source: Snort alert_fast file
source s_snort {
    file("/var/log/snort/snort.alert.fast" follow_freq(1) flags(no-parse));
};
# Log Path: Snort to Fluentd
log {
    source(s_snort);
    destination(d_fluentd_snort);
};
source s_nagios_log {
    file("/usr/local/nagios/var/nagios.log" follow_freq(1) flags(no-parse));
};
filter f_nagios {
    message("nagios");
};
log {
    source(s_nagios_log);
    filter(f_nagios);
    destination(d_fluentd_nagios);
};
@include "/etc/syslog-ng/conf.d/*.conf"
```

Command: *sudo systemctl enable syslog-ng*

sudo systemctl restart syslog-ng

8. Fluentd installation and configuration

For install Fluentd we are going to use the ansible playbook named as "fluentd.yml". This file must be located inside of /etc/ansible/ and you need to be in that path.

Command: *sudo ansible-playbook -c local fluentd.yml*

Now you need to configure Fluentd, to get the logs are collected and organized by syslog-ng.

Command: *sudo nano /etc/fluent/fluentd.conf*

Att: Also, file fluentd.conf is provided for download and substitute directly.

```
vsoc@vsoc: ~
  <store>
    @type stdout
  </store>
  <store>
    @type elasticsearch
    host localhost
    port 9200
    <buffer>
      flush_interval 5s
    </buffer>
    index_name snort-test-logs
    logstash_format false
  </store>
</match>

# Nagios Logs
<match nagios.log>
  @type stdout
#  @type elasticsearch
  <buffer>
    flush_interval 5s
  </buffer>
</match>
<match *.log>
  @type elasticsearch
  host localhost
  port 9200
  logstash_format true
  index_name fluentd
</match>
```

Command: *sudo systemctl enable fluentd*

sudo systemctl restart fluentd

9. Elasticsearch and Kibana installation and configuration

For install Elasticsearch and Kibana we are going to use the ansible playbook named as “elasticsearch-installation.yml”. This file must be located inside of /etc/ansible/ and you need to be in that path.

Command: *sudo ansible-playbook -c local elasticsearch.yml*

Now you need to configure Elasticsearch and Kibana for get the logs from Fluentd, and be available to visualize this logs.

Command: *sudo nano /etc/elasticsearc/elasticsearch.yml*

Att: Also, file elasticsearch.yml is provided for download and substitute directly.

```
# ======================== Elasticsearch Configuration =========================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: elk-vsoc
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
#node.name: node-1
#
# Add custom attributes to the node:
#
```

Command: *sudo nano /etc/kibana/kibana.yml*

Att: Also, file kibana.yml is provided for download and substitute directly.

```
# The number of times to retry temporary migration failures. Increase the setting
# if migrations fail frequently with a message such as `Unable to complete the [...] ste
# 15 attempts, terminating`. Defaults to 15
#migrations.retryAttempts: 15

# ==================== Search Autocomplete ====================
# Time in milliseconds to wait for autocomplete suggestions from Elasticsearch.
# This value must be a whole number greater than zero. Defaults to 1000ms
#unifiedSearch.autocomplete.valueSuggestions.timeout: 1000

# Maximum number of documents loaded by each shard to generate autocomplete suggestions.
# This value must be a whole number greater than zero. Defaults to 100_000
#unifiedSearch.autocomplete.valueSuggestions.terminateAfter: 100000


# This section was automatically generated during setup.
elasticsearch.hosts: [http://localhost:9200]
elasticsearch.username: kibana_system
elasticsearch.password: '-dyVud-yg3X+vDIFxrDH'
elasticsearch.ssl.certificateAuthorities: [/var/lib/kibana/ca_1745590482890.crt]
xpack.fleet.outputs: [{id: fleet-default-output, name: default, is_default: true, is_def
ring: true, type: elasticsearch, hosts: [http://localhost:9200], ca_trusted_fingerprint:
4b6c44d4df13dd8ce050d641a1be8c0ed7f2833fe2d5b158e133ff}]
```

Command:

sudo systemctl enable elasticsearch

sudo systemctl restart elasticsearch

sudo systemctl enable kibana

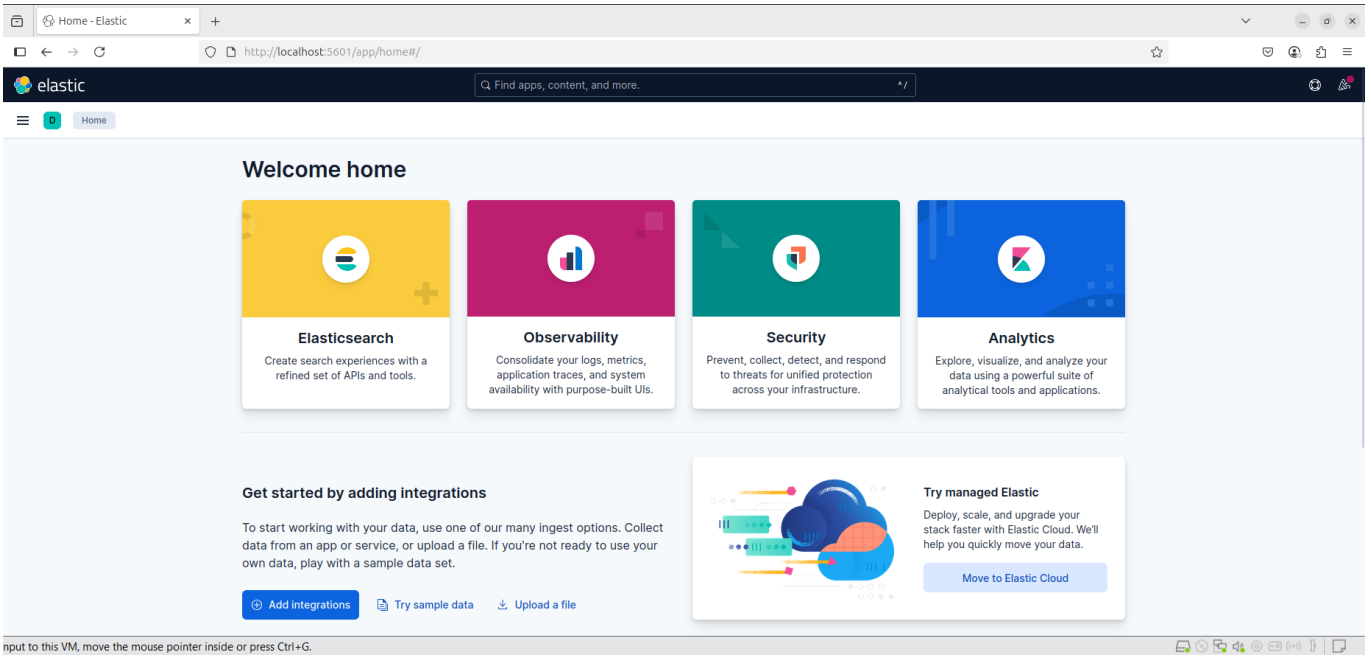
sudo systemctl restart kibana

Now we can access to Firefox and go to http://localhost:5601 , the username is elastic and the password will need to be reset.

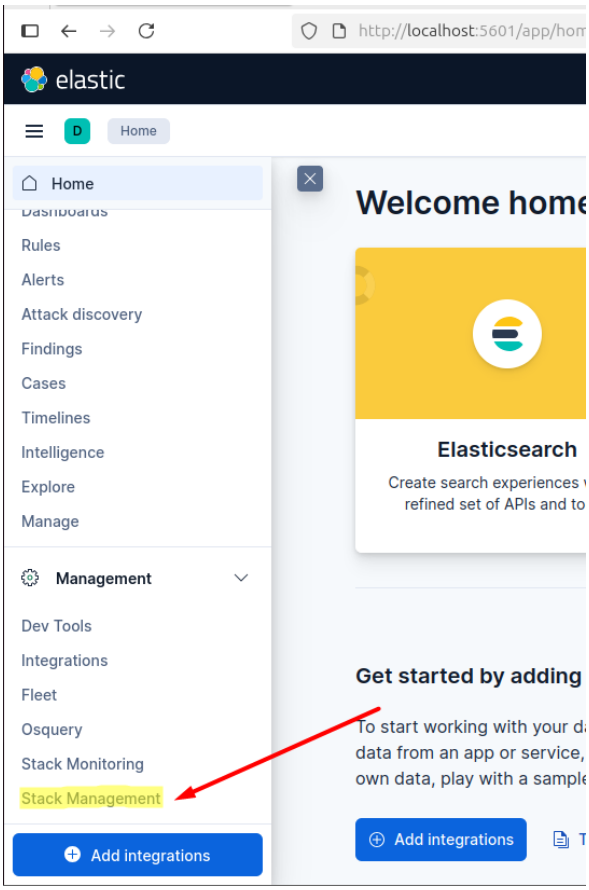
Command: *sudo /usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic*

This command will prompt you the password for the user, you will need to use. After that you will have a Access Token from Kibana.

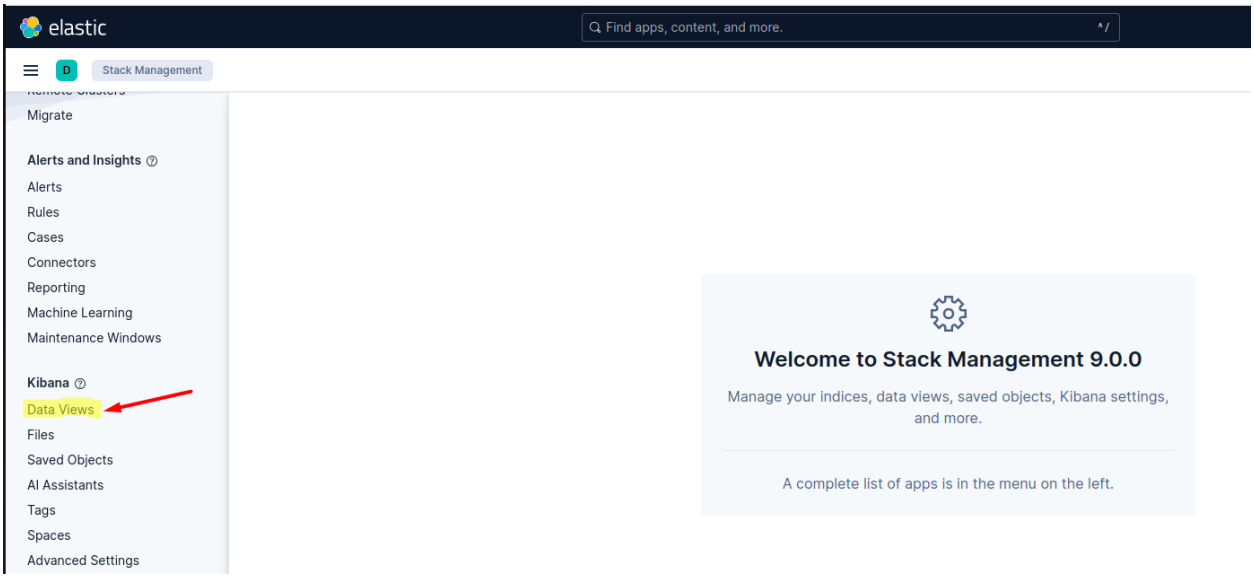
Command: *sudo /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana*



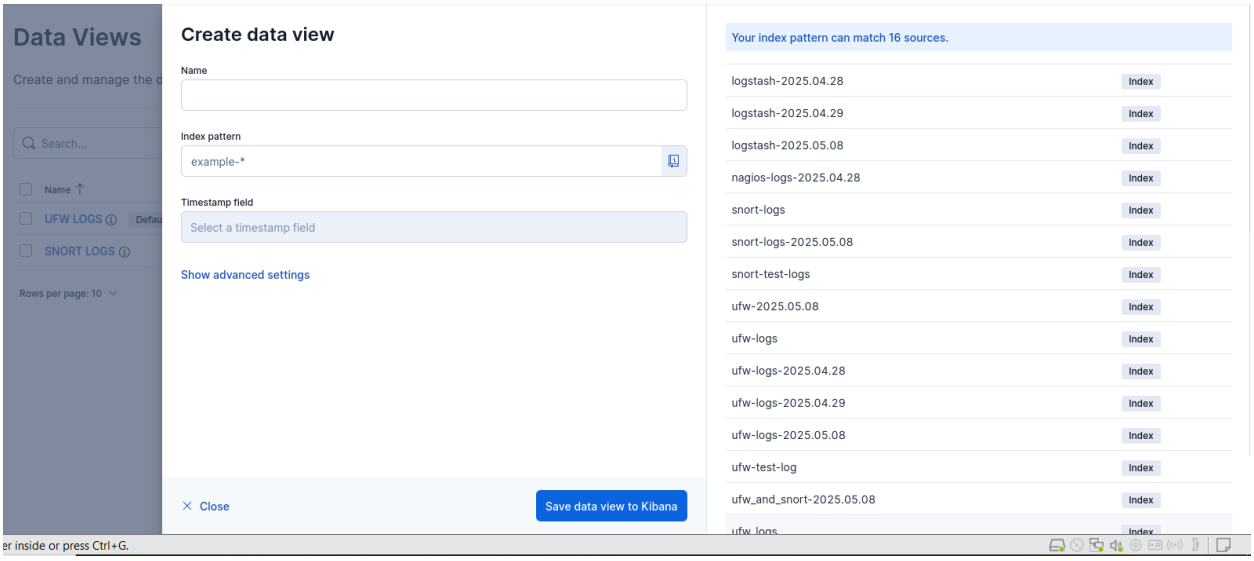
Now we can configure the dashboard in ELK, we need to go to Management→ Stack Management.



Then Kibana → Data Views.



Now you can create a new Data View for Kibana, on the right side will show the options of logs is detecting your Kibana are being sent to Elasticsearch. So, you must indicate the index pattern for example: snort-* (This will give you all the snort logs).



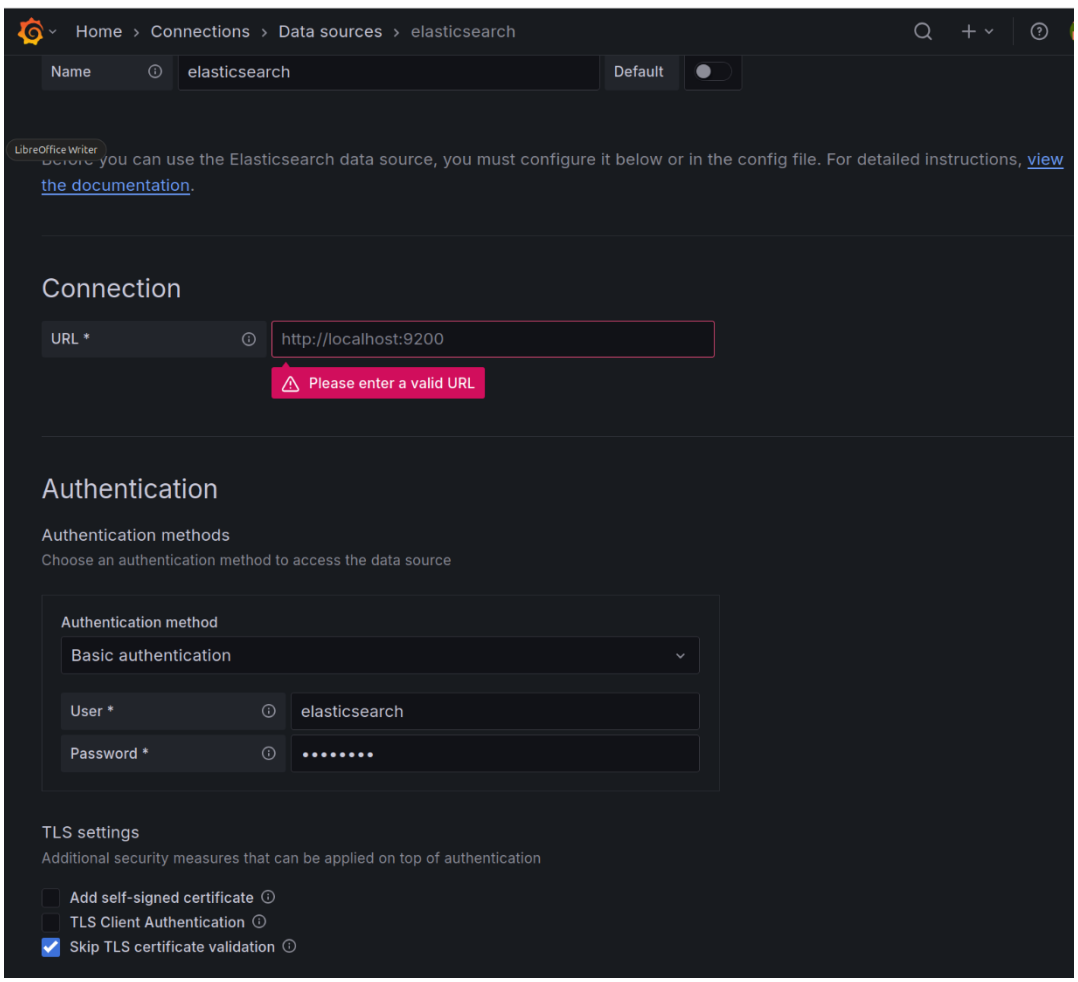
10. Grafana installation and configuration

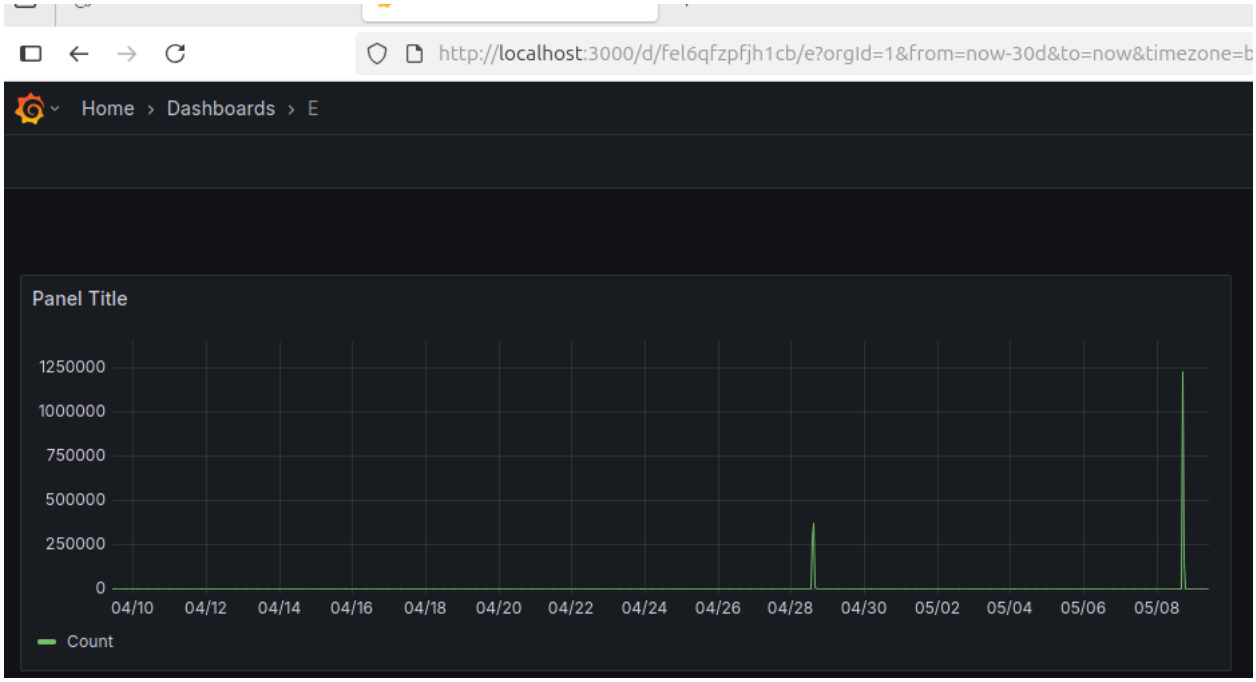
For install Grafana we are going to use the ansible playbook named as "grafana.yml". This file must be located inside of /etc/ansible/ and you need to be in that path.

Command: *sudo ansible-playbook -c local grafana.yml*

After you finish your Grafana installation, we need to access to: http://localhost:3000 , user: admin and password: admin. Will ask to you of changing the password put for example: vsoc2025.

Now go to Home→Connections → Add new connection → Search for elasticsearch



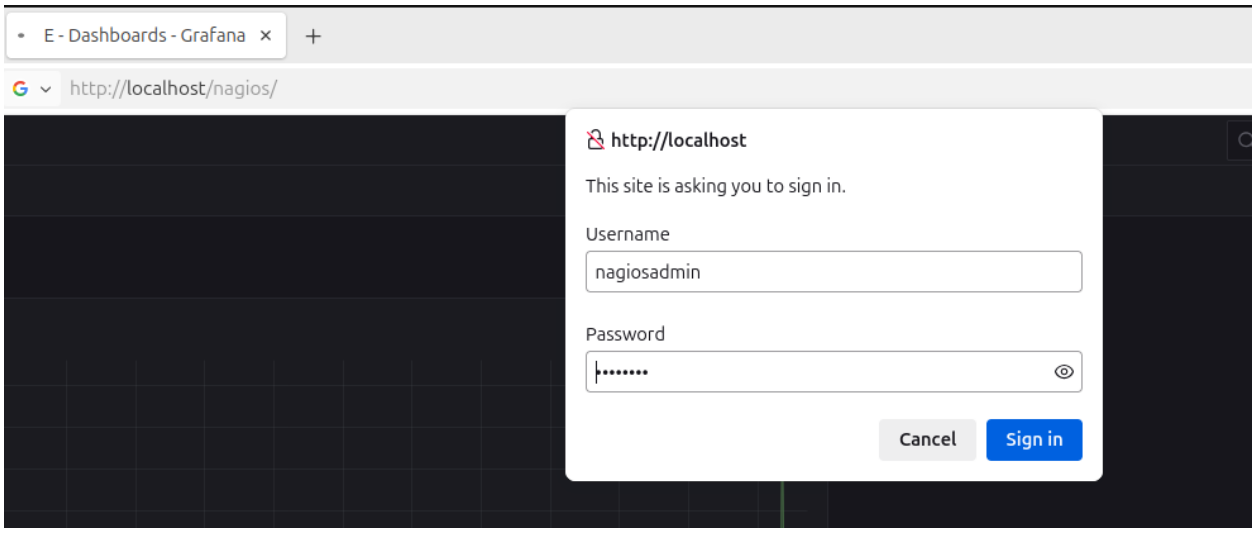

http://localhost:3000/d/fel6qfzpfjh1cb/e?orgId=1&from=now-30d&to=now&timezone=b
Home › Dashboards › E
Panel Title
1250000
1000000
750000
500000
250000
0
04/10
04/12
04/14
04/16
04/18
04/20
04/22
04/24
04/26
04/28
04/30
05/02
05/04
05/06
05/08
Count

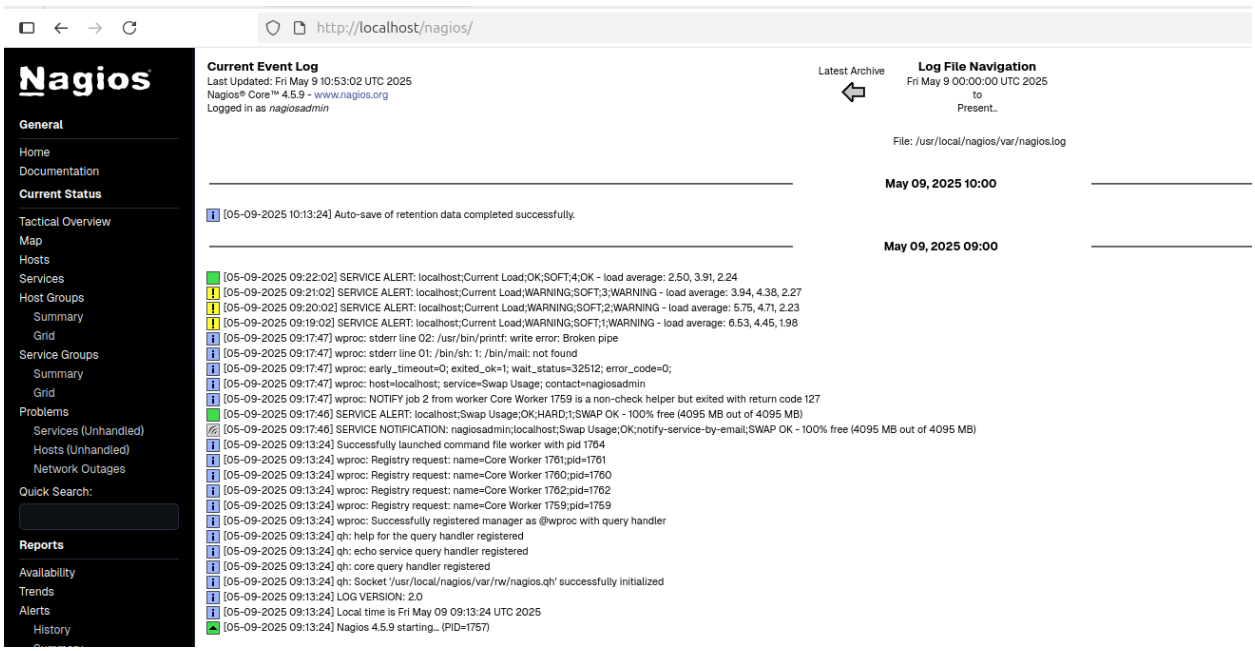
11. Nagios installation and configuration

For install Nagios we are going to use the ansible playbook named as "nagios.yml". This file must be located inside of /etc/ansible/ and you need to be in that path.

Command: *sudo ansible-playbook -c local nagios.yml*

After you finish your Nagios installation you can access to http://localhost/nagios/ there will prompt you for indicate username: nagiosadmin, and password: nagiosadmin123 (Change it after log-in).





12. MITRE CALDERA installation and configuration

For install MITRE CALDERA will be doing it without ansible due the python environment for installation can have some issues for do the job.

First is needed to install the pre-needed packages for the installation.

Command:

```
sudo apt install git python3 python3-pip python3-venv -
```

Then,

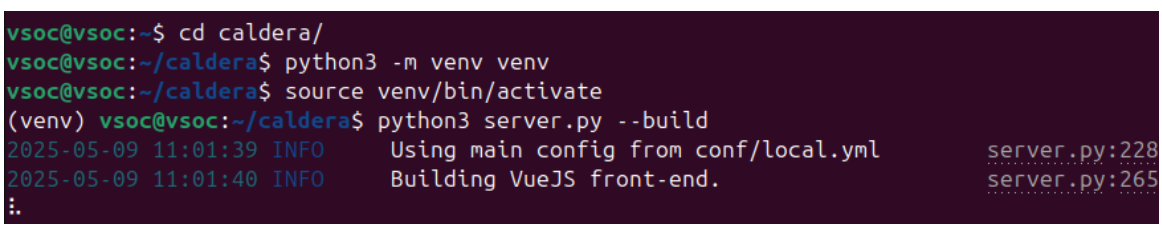
Command:

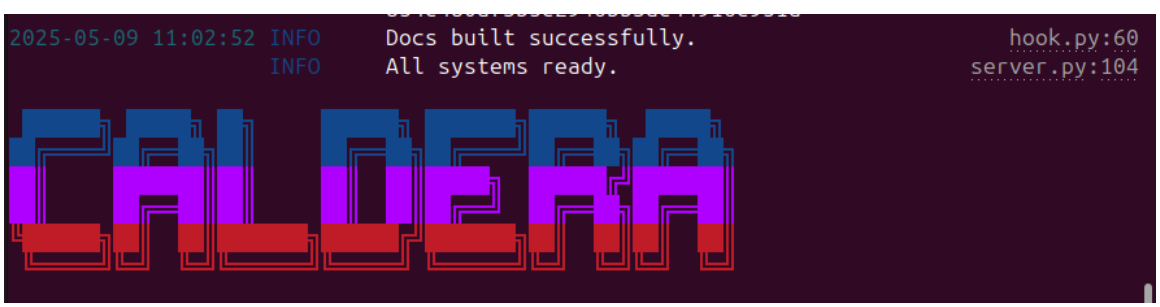
```
cd /opt
sudo git clone https://github.com/mitre/caldera.git --recursive
cd caldera
```

Now we can activate the Python environment for make the installation of the software.

Command:

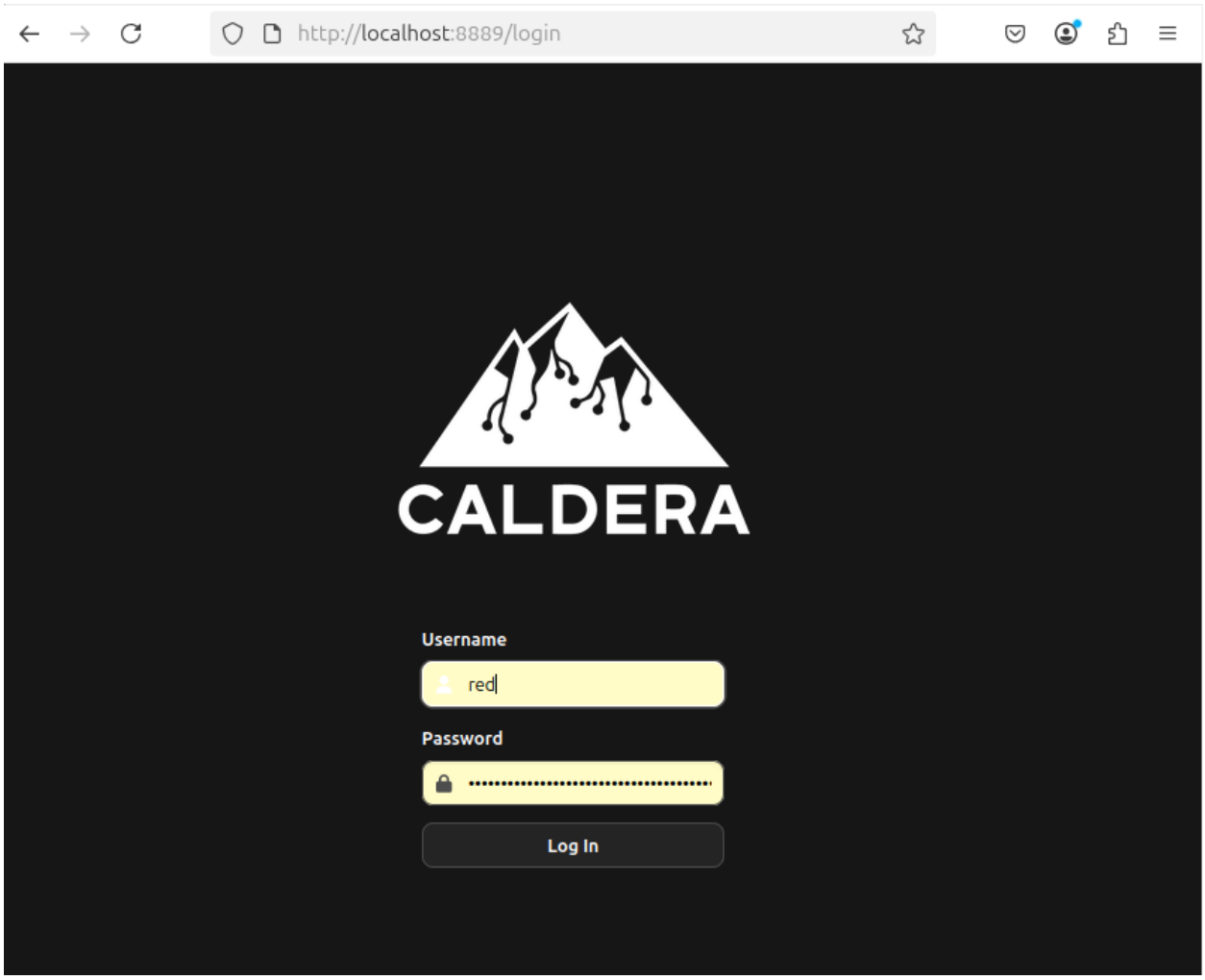
```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
python3 server.py --build
```





Now you can access to http://localhost:8888 or http://localhost:8889. There are two different log-in red team or blue team, you can find the credentials of these into /caldera/conf/local.yml

```
users:
  blue:
    blue: yE3-kuoTVyztCSz9S10eREuH-EdXN6bPFFKZsm9rU4g
  red:
    red: uz7m6F33BaLq3jc4f085u4UHfyqHD6Q-VGtW2x43gck
vsoc@vsoc:~/caldera/conf$
```

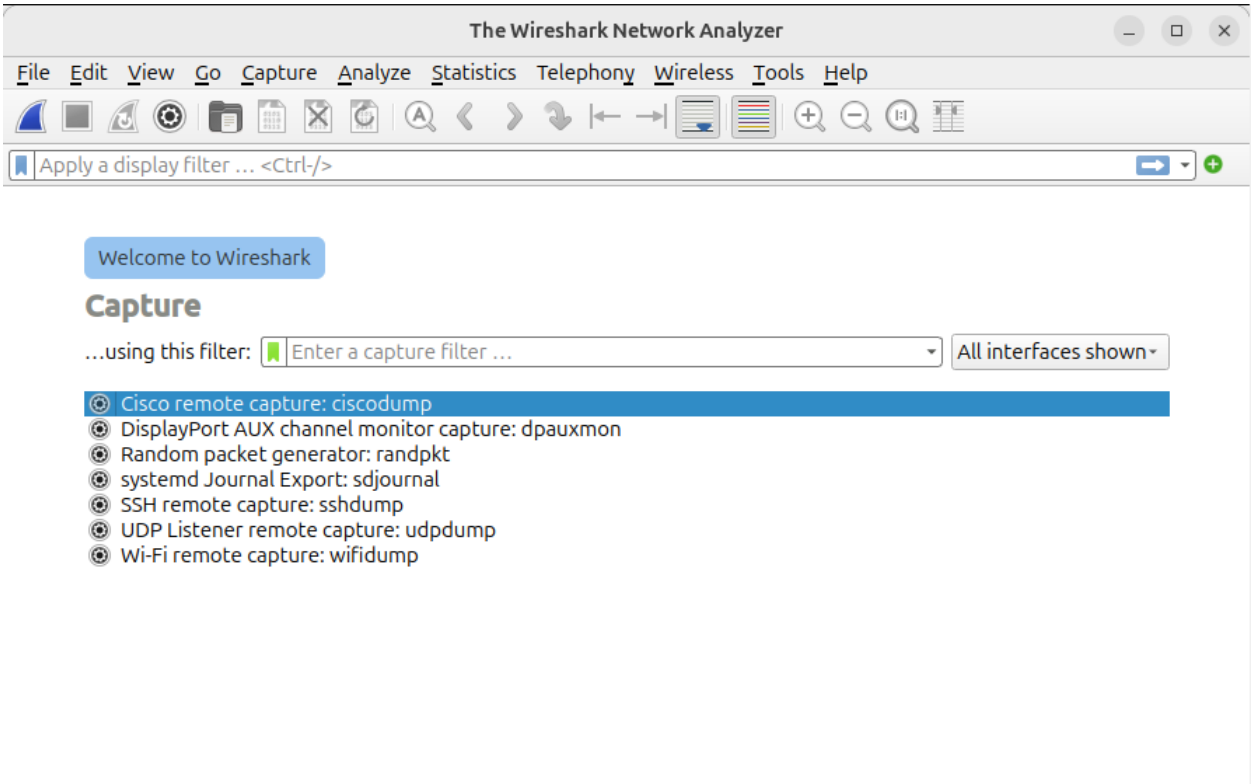


13. Wireshark installation

Command: *sudo apt install wireshark*

Then execute the next command for opening the application of Wireshark.

Command: *wireshark*



14. Links and Resources

1. https://www.hostmycode.in/tutorials/install-and-configure-nagios-on-ubuntu
2. https://www.manageengine.com/products/eventlog/sem/glp/syslog-server.html?camid=19529450106&adgid=175568250399&kwd=syslog%20server%20linux&matchtype=p&adid=738673661453&network=g&adposition=&loc=9008042&placement=&target=&device=c&gad_source=1&gad_campaignid=19529450106&gbraid=0AAAAAChAr7Z_crrrCCamor6xseCCh_2vF&gclid=CjwKCAjwz_bABhAGEiwAm-P8YRoprj1afTVobUEpFnVrPAzekmCQQv7GjT0uI1zR6qqPFtEosivgDRoCH4kQAvD_BwE
3. https://www.syslog-ng.com/community/b/blog/posts/installing-the-latest-syslog-ng-on-ubuntu-and-other-deb-distributions
4. https://www.zenarmor.com/docs/linux-tutorials/how-to-install-and-configure-snort-on-ubuntu-linux#5-running-snort-as-a-service
5. https://docs.netgate.com/pfsense/en/latest/packages/snort/setup.html
6. https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-elasticsearch-on-ubuntu-22-04
7. https://grafana.com/docs/grafana/latest/setup-grafana/installation/debian/
8. https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu
9. https://docs.fluentd.org/installation
10. https://medium.com/@salim.y.salimov/installing-mitre-caldera-4-on-ubuntu-vm-fac970825352
11. https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html