IBM ProjectReport

On

Develop an application to facilitate IPR filing for the grassroots community

Developed By:
Rishabh Patel (19162171034)
Sahil Patel (20162172004)
Smit Patel (19162121031)

Guided By:
Prof.Umesh Lakhtariya (Internal)
Mr. Anoj Dixit (External guide)

Submitted to

Department of Computer Science & Engineering
Institute of Computer Technology

Year: 2023

CERTIFICATE

This is to certify that the **IBM** Project work entitled **"**Develop an application to facilitate IPR filing for the grassroots community**"** by Rishabh Patel (Enrolment No.19162171034), Sahil Patel (Enrolment No.20162172007) and Smit Patel (EnrolmentNo.19162121031) of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CS/BDA) Department at Elegant Microweb Pvt. Ltd. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree.

Name & Signature of Internal Guide

Name & Signature of Head

Place: ICT - GUNI

Date:

ACKNOWLEDGEMENT

IBM Internship project is a golden opportunity for learning and self-development. I considermyself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank Dr. Hemal Shah, Principal, ICT, and Prof. Dharmesh Darji, Head, ICT who gave us an opportunity to undertake this project. My grateful thanksto Prof. Umesh Lakhtariya & Mr. Anoj Dixit (Internal & External Guides) for their guidance in project work Develop an application to facilitate IPR filing for the grassroots community, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without his/her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Rishabh Patel (Enrollment No:19162171034)
Sahil Patel (Enrollment No: 20162172007)
Smit Patel (Enrollment No:19162121031)

ABSTRACT

The development of an application to facilitate IPR filing for the grassroots community aims to address the challenges faced by individuals and small organizations in navigating the complex and often costly process of obtaining intellectual property rights. By providing a user-friendly platform with step-by-step guides, document templates, and the ability to submit applications electronically, the application aims to make the IPR filing process more accessible and efficient for those with limited resources. Additionally, the application would provide information and resources on different types of IPR, such as patents, trademarks, and copyrights, to assist users in understanding the process and making informed decisions. Overall, the application aims to empower the grassroots community to protect their intellectual property and support their growth and development.

INDEX

CHAPTER: 1 INTRODUCTION

CHAPTER 1 INTRODUCTION

An application to facilitate IPR filing for the grassroots community would likely be a software tool or platform that makes it easier for individuals or small organizations with limited resources to file for intellectual property rights. This could include features such as step-by-step guides, document templates, and the ability to submit applications electronically. The application could also provide resources and information on different types of IPR, such as patents, trademarks, and copyrights. The goal of such an application would be to make the IPR filing process more accessible and user-friendly for those who may not have the knowledge or resources to navigate the process on their own.

CHAPTER: 2 PROJECT SCOPE

CHAPTER 2 PROJECT SCOPE

The scope of a project to develop an application to facilitate IPR filing for the grassroots community would include the following:

Research and analysis: This would involve researching the current IPR filing process, identifying the challenges faced by the grassroots community, and determining the features and functionality that would be most useful in addressing these challenges.

Design and development: This would involve creating the user interface and features of the application, such as step-by-step guides, document templates, and the ability to submit applications electronically.

Testing and quality assurance: This would involve testing the application to ensure it is user-friendly and functions as intended, and making any necessary adjustments before launch.

Deployment and maintenance: This would involve deploying the application and providing ongoing maintenance and support to ensure it continues to function effectively.

Documentations: This will include requirement analysis, design documents, user manuals and other required documents that will be shared with the end-users.

Training: This will include providing training to end-users on how to use the application and navigate the IPR filing process.

The project scope should be flexible enough to include any additional features or functionality that may be identified during the development process as needed.

CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS

CHAPTER 3 SOFTWARE AND HARDWARE REQUIREMENTS

Minimum Hardware Requirements

Processor	2.0 GHz
RAM	4GB
HDD	40GB

Table 3.1 Minimum Hardware Requirements

Minimum Software Requirements

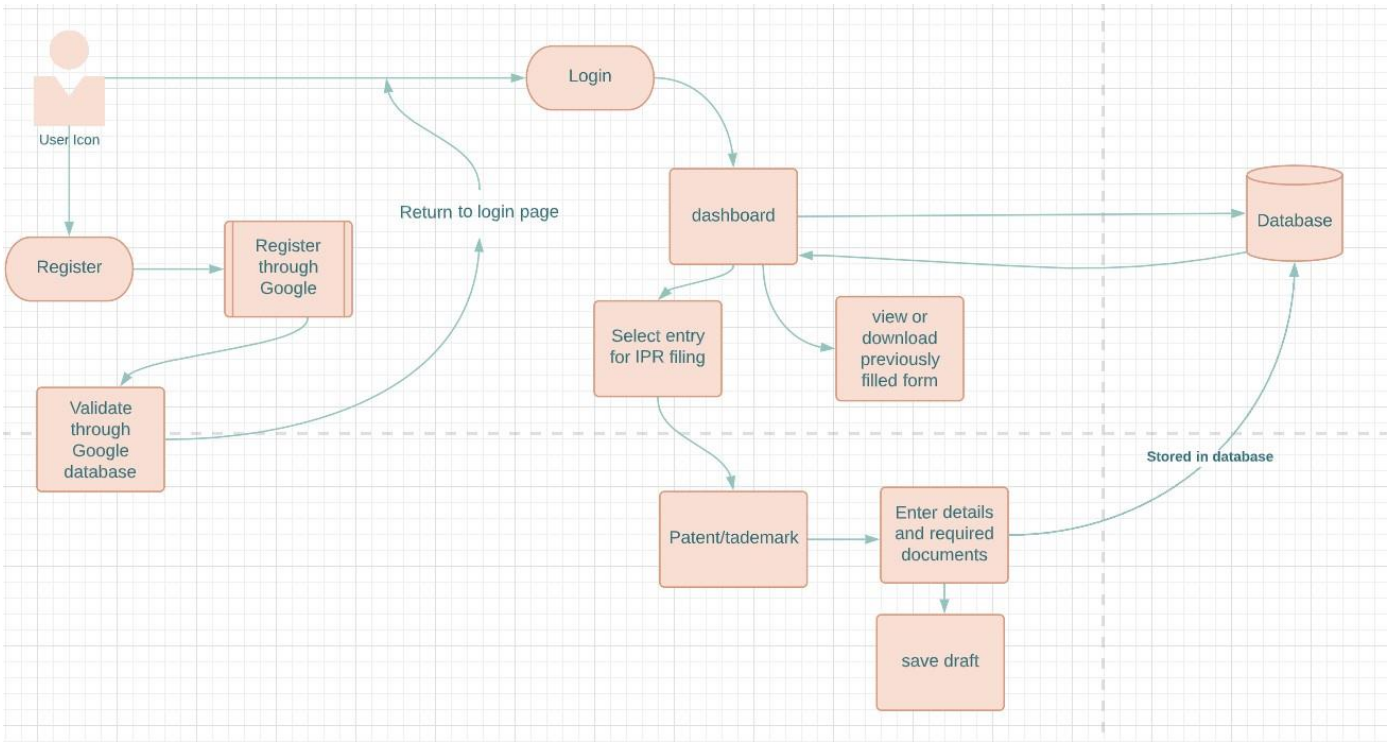
Operating System	Any operating system which can support an internet browser.
Programming language	-
Other tools & tech	Internet browser

Table 3.2 Minimum Software Requirements

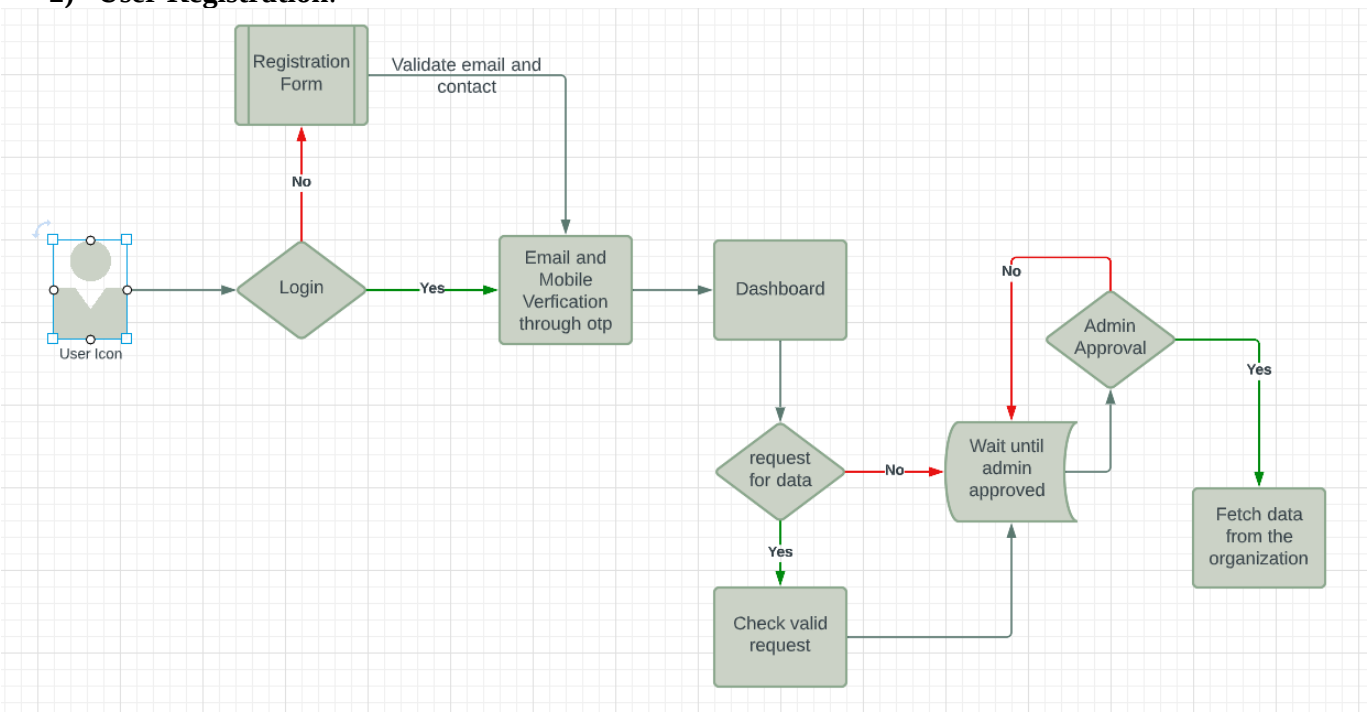
CHAPTER: 4 PROCESS MODEL

CHAPTER 4 PROCESS MODEL

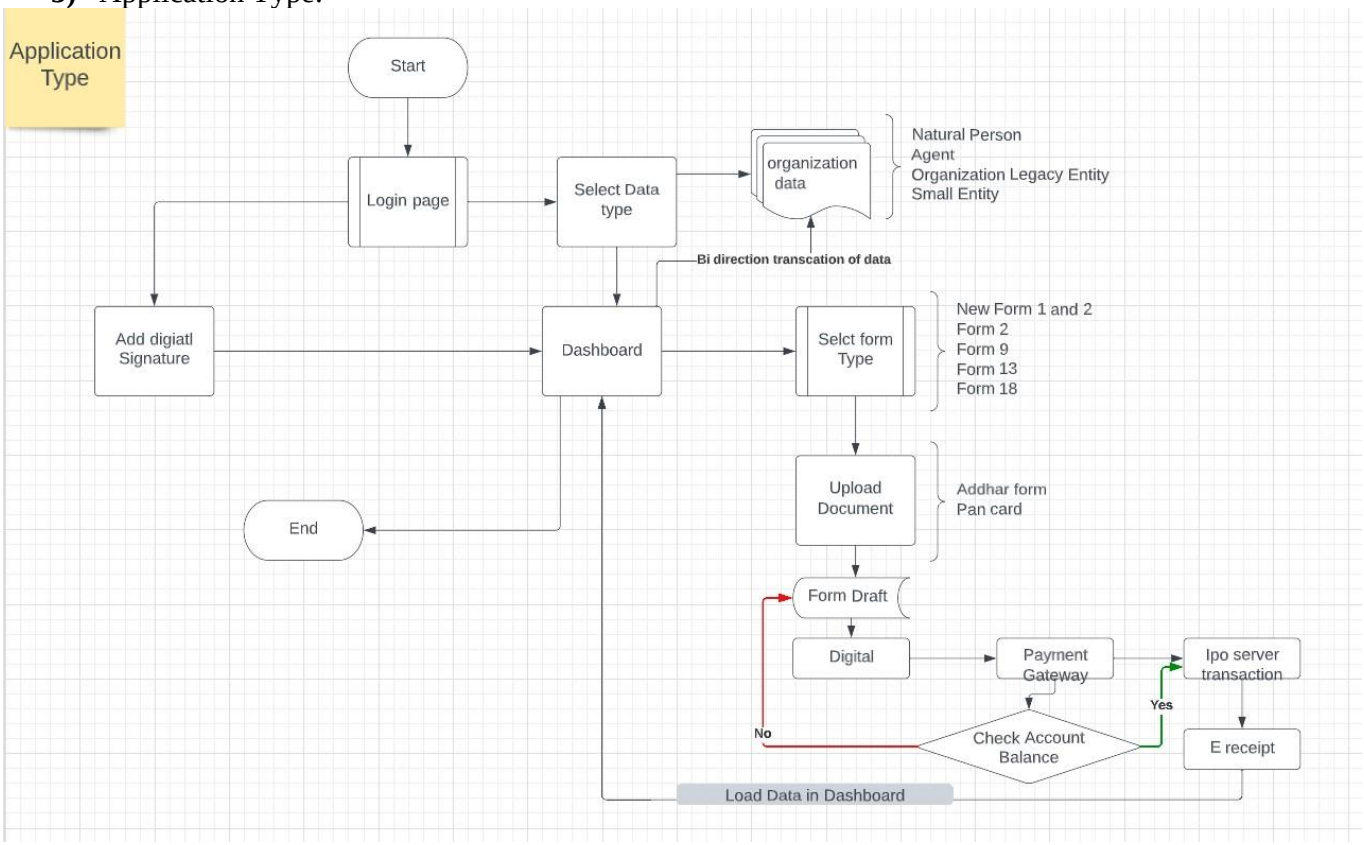
1) User Login: -



2) User Registration: -



3) Application Type: -



CHAPTER: 5 PROJECT PLAN

CHAPTER 5 PROJECT PLAN

A project plan for developing an application to facilitate IPR filing for the grassroots community could include the following steps:

Kick-off meeting: Hold a meeting with the project team to discuss the project scope, goals, and timelines.

Research and analysis: Conduct research on the current IPR filing process and identify the challenges faced by the grassroots community. Determine the features and functionality that would be most useful in addressing these challenges.

Design and development: Create the user interface and features of the application, such as step-by-step guides, document templates, and the ability to submit applications electronically.

Testing and quality assurance: Test the application to ensure it is user-friendly and functions as intended, and make any necessary adjustments before launch.

Deployment and maintenance: Deploy the application and provide ongoing maintenance and support to ensure it continues to function effectively.

Training: Provide training to end-users on how to use the application and navigate the IPR filing process.

Monitoring and evaluation: Monitor the usage of the application and gather feedback from the end-users to evaluate the effectiveness of the application and identify areas for improvement.

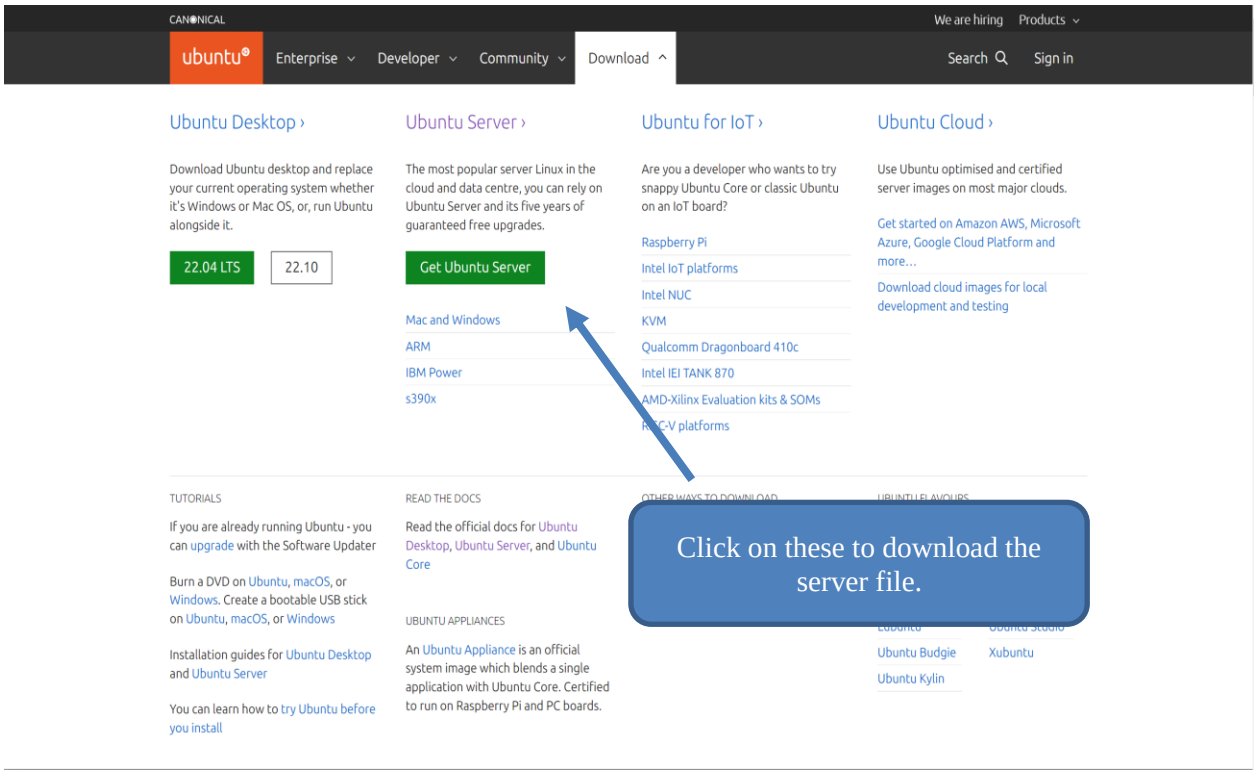
Close project: Close the project and document the lessons learned.

It is important to establish clear milestones and deadlines for each step, and to regularly review the progress of the project to ensure it stays on track. Also, this plan should be flexible enough to accommodate any changes or unforeseen challenges that may arise during the course of the project.

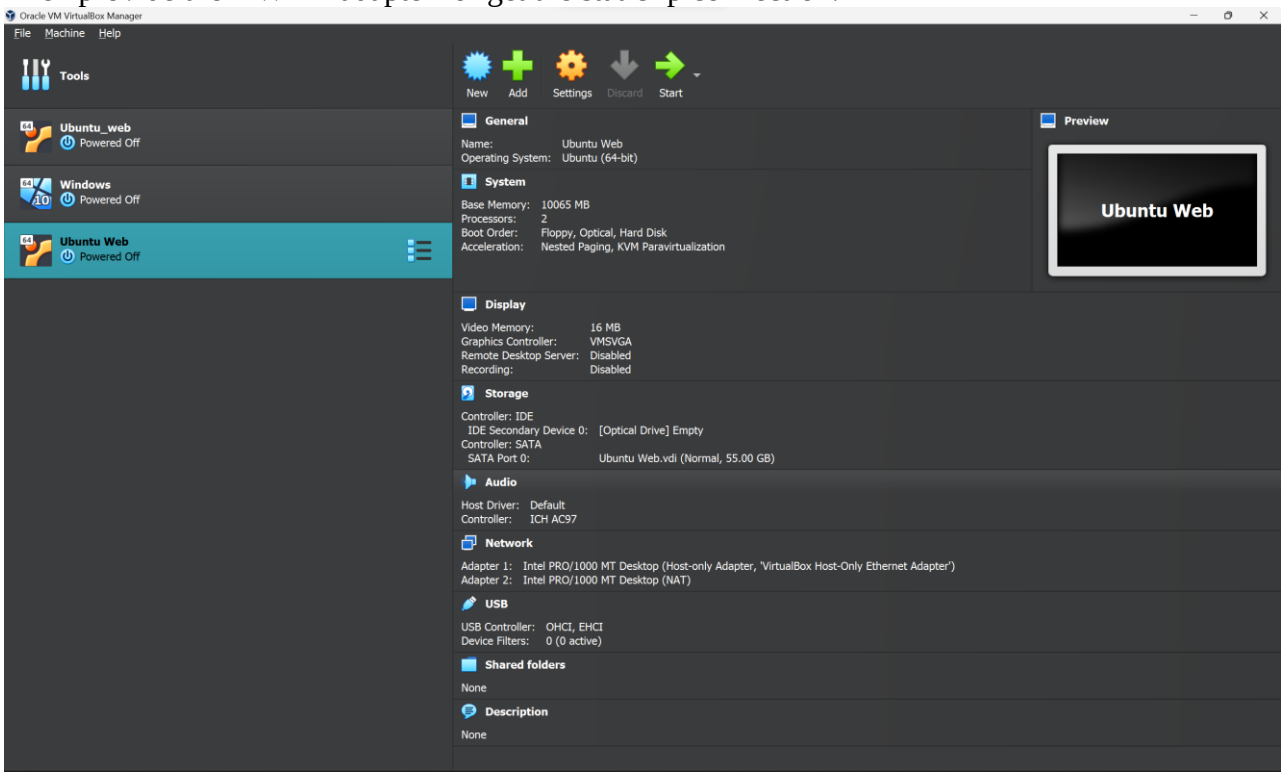
CHAPTER: 6 IMPLEMENTATION DETAILS

4.1. Ubuntu server installation.

Move to Ubuntu official web side and download the server file form the download tab.

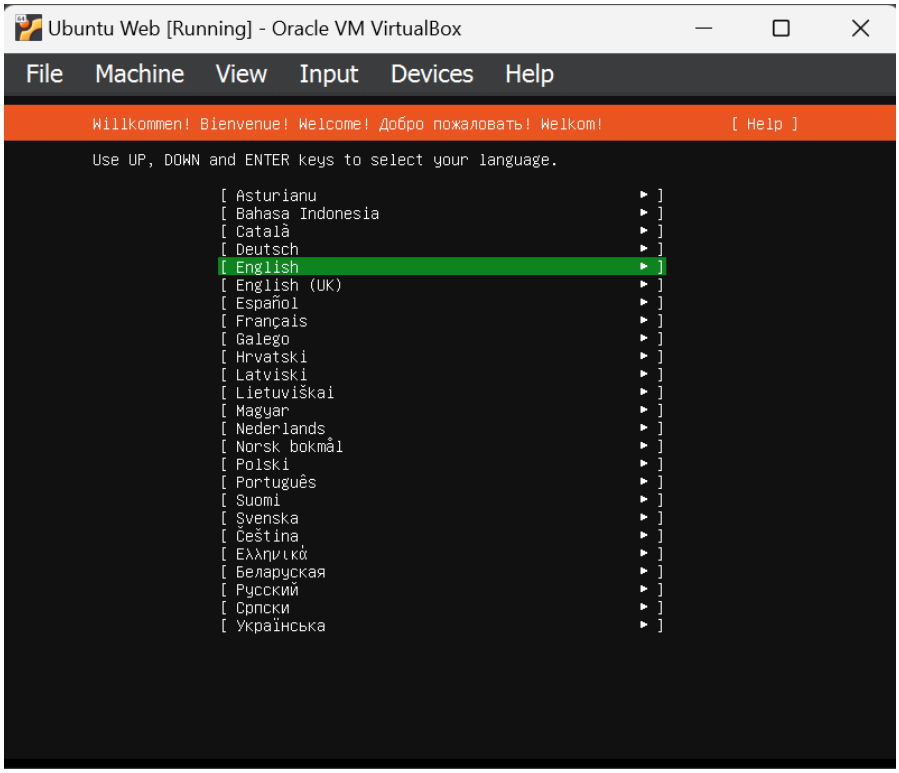


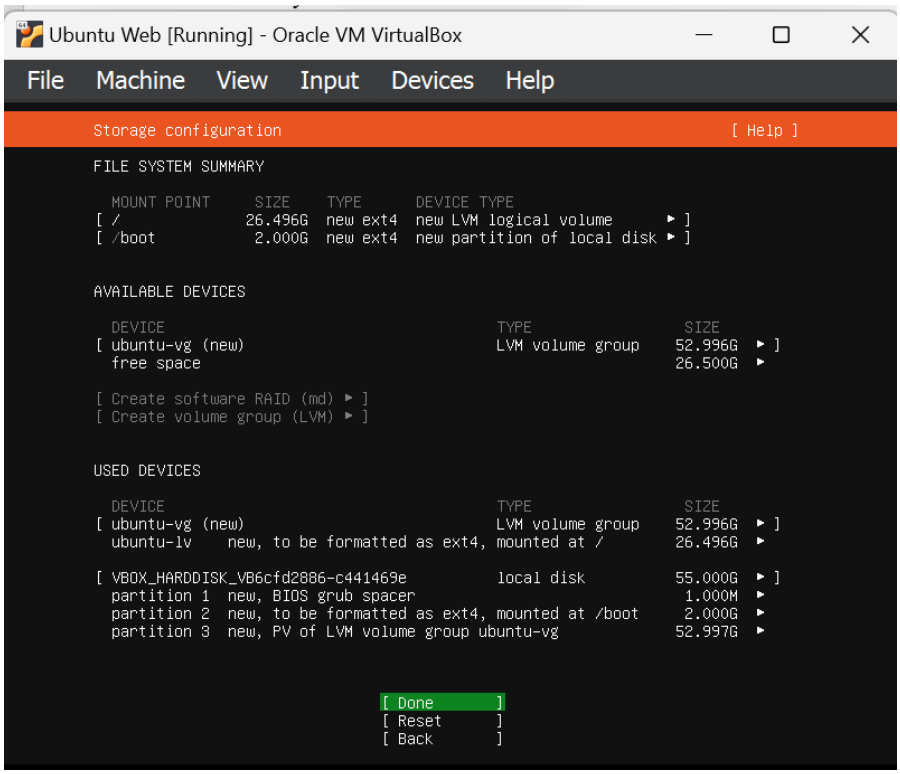
Now open the Virtual Box and create a new machine with name web and password 1212
Then provide the 2 Wi-Fi adapter for get the static Ip connection.



1. Host-only
2. NAT

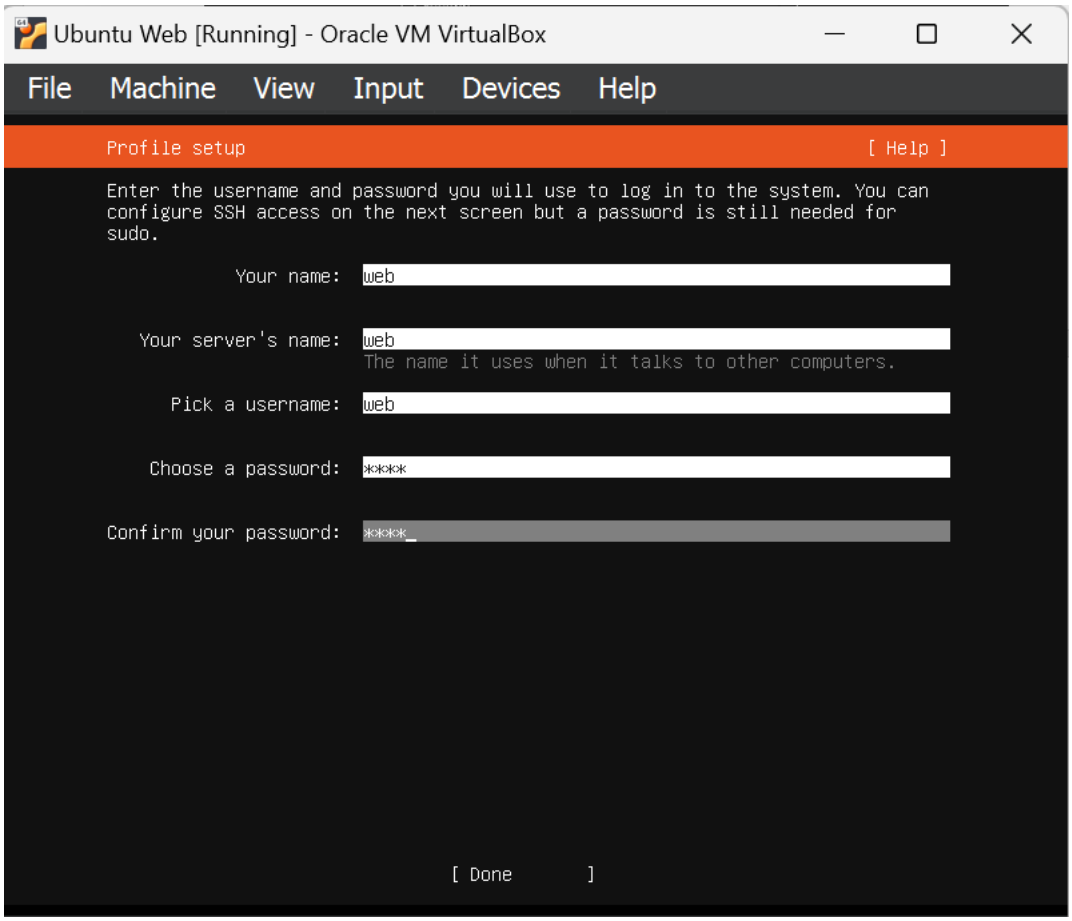
Now Run the Virtual Machine of Ubuntu Server.



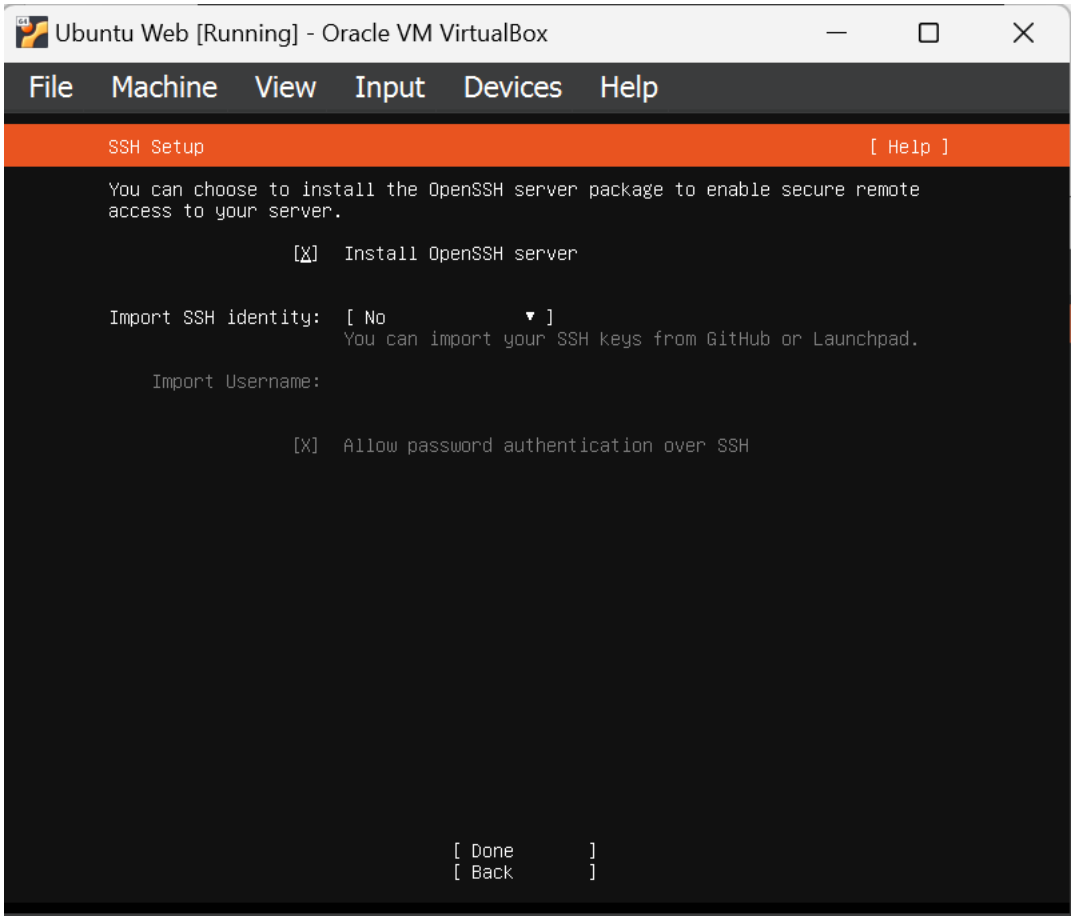


Hit enter – enter until you get these screen.

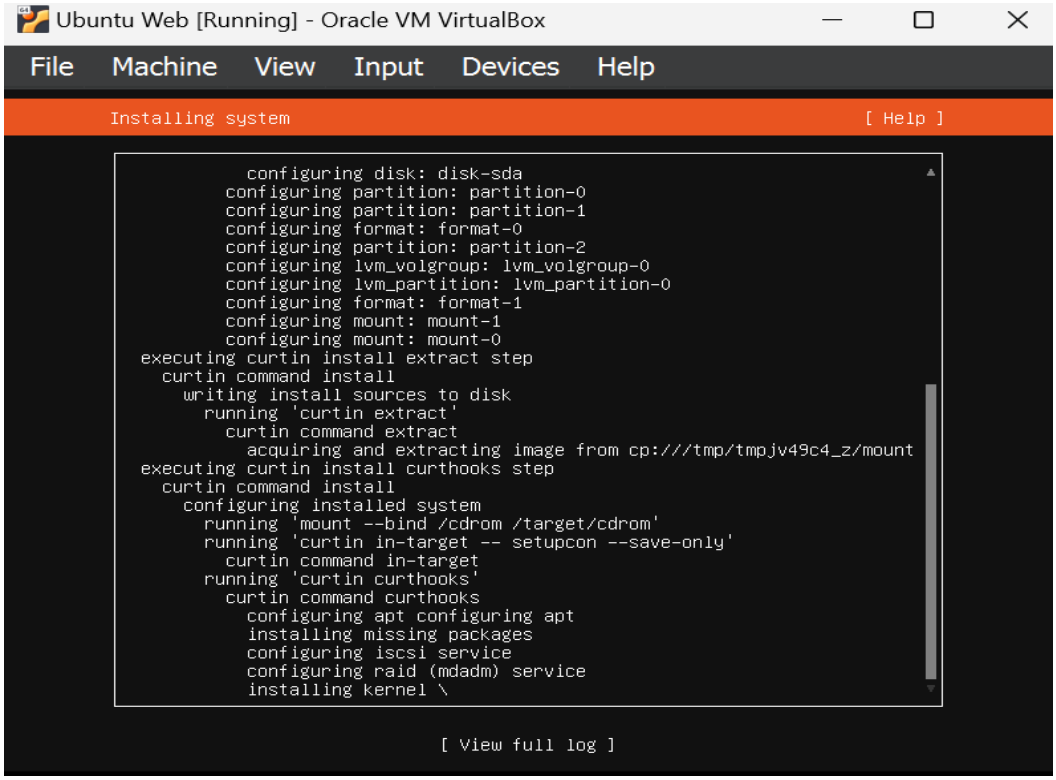
Enter the Credentials as shown in figure.



Now hit enter and move to SSH installation page.



Then just hit enter and wait till installation has been completed.



4.2. Wazuh Installation through CLI.

Make 2 directory in server

1) Wazuh – primary directory
2) Backup – backup directory

```
webserver@webserver:~$ mkdir wazuh
webserver@webserver:~$ mkdir backup
```

Installing Wazuh in primary directory.

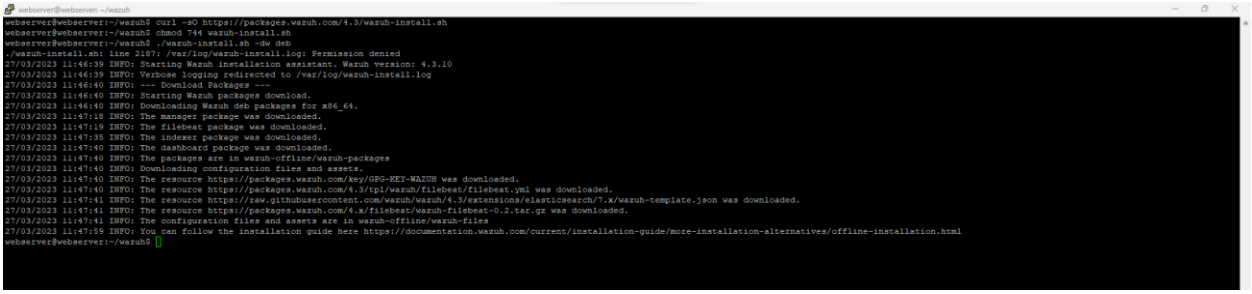
"curl -sO https://packages.wazuh.com/4.3/wazuh-install.sh" download the Wazuh package.

```
webserver@webserver:~/wazuh$ curl -sO https://packages.wazuh.com/4.3/wazuh-install.sh
webserver@webserver:~/wazuh$
```

Change the permission of Wazuh-install.sh file and after run the install command for installing the file.

"chmod 744 wazuh-install.sh"

"./wazuh-install.sh -dw deb"



“curl -sO https://packages.wazuh.com/4.3/config.yml” Download the certificates configuration file.

```
webserver@webserver:~/wazuh$ curl -sO https://packages.wazuh.com/4.3/config.yml
webserver@webserver:~/wazuh$ ls
config.yml  wazuh-install.sh  wazuh-offline.tar.gz
webserver@webserver:~/wazuh$
```

Edit config.yml to prepare the certificates creation. Give the ip address of ubuntu server in indexer, server and dashboard.

```
root@webserver: /home/webserver/wazuh
  GNU nano 6.2
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: 192.168.1.10
    #- name: node-2
    #  ip: <indexer-node-ip>
    #- name: node-3
    #  ip: <indexer-node-ip>

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: wazuh-1
      ip: 192.168.1.10
    #  node_type: master
    #- name: wazuh-2
    #  ip: <wazuh-manager-ip>
    #  node_type: worker
    #- name: wazuh-3
    #  ip: <wazuh-manager-ip>
    #  node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: dashboard
      ip: 192.168.1.10
```

After press ctrl+o and ctrl+x for save and exit the file.
For recheck the enter value use cat command: “cat config.yml”

```
root@webserver:/home/webserver/wazuh# cat config.yml
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: 192.168.1.10
    #- name: node-2
    #  ip: <indexer-node-ip>
    #- name: node-3
    #  ip: <indexer-node-ip>

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: wazuh-1
      ip: 192.168.1.10
    #  node_type: master
    #- name: wazuh-2
    #  ip: <wazuh-manager-ip>
    #  node_type: worker
    #- name: wazuh-3
    #  ip: <wazuh-manager-ip>
    #  node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: dashboard
      ip: 192.168.1.10
```

Run the ./wazuh-certs-tool.sh to create the certificates. For a multi-node cluster, these certificates need to be later deployed to all Wazuh instances in your cluster.

"curl -sO https://packages.wazuh.com/4.3/wazuh-certs-tool.sh"
"chmod 744 wazuh-certs-tool.sh"
"./wazuh-certs-tool.sh –all"

Copy or move wazuh-offline.tar.gz file and ./wazuh-certificates/ folder to a folder accessible to the host(s) from where the offline installation will be carried out

```
root@webserver:/home/webserver/wazuh# ls
config.yml  wazuh-certificates  wazuh-certs-tool.sh  wazuh-install.sh  wazuh-offline.tar.gz
root@webserver:/home/webserver/wazuh#
```

Now copy file to backup directory if you want otherwise it`s fine to continue the work.

```
root@webserver:/home/webserver/wazuh# scp -r wazuh-offline.tar.gz webserver@192.168.1.10:/home/webserver/
The authenticity of host '192.168.1.10 (192.168.1.10)' can't be established.
ED25519 key fingerprint is SHA256:owokQm7byTnSOetXTkwSI0oWZcLxoLP+BNMjoqMFyr8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.10' (ED25519) to the list of known hosts.
webserver@192.168.1.10's password:
wazuh-offline.tar.gz                                                                    100%  599MB 384.7MB/s   00:01
root@webserver:/home/webserver/wazuh# scp -r wazuh-certificates webserver@192.168.1.10:/home/webserver/
webserver@192.168.1.10's password:
root-ca.pem                                                                             100% 1204     1.3MB/s   00:00
wazuh-1.pem                                                                             100% 1277     1.4MB/s   00:00
dashboard-key.pem                                                                       100% 1704     1.5MB/s   00:00
node-1-key.pem                                                                          100% 1700     1.8MB/s   00:00
admin-key.pem                                                                           100% 1704     2.5MB/s   00:00
root-ca.key                                                                             100% 1704     2.3MB/s   00:00
wazuh-1-key.pem                                                                         100% 1704     1.5MB/s   00:00
node-1.pem                                                                              100% 1277     1.8MB/s   00:00
admin.pem                                                                               100% 1119   764.0KB/s   00:00
dashboard.pem                                                                           100% 1281     1.9MB/s   00:00
root@webserver:/home/webserver/wazuh#
```

```
webserver@webserver:~$ ls
backup  wazuh  wazuh-certificates  wazuh-offline.tar.gz
webserver@webserver:~$ sudo su
[sudo] password for webserver:
root@webserver:/home/webserver# mv wazuh-certificates /home/webserver/backup/
root@webserver:/home/webserver# ls
backup  wazuh  wazuh-offline.tar.gz
root@webserver:/home/webserver# mv wazuh-offline.tar.gz /home/webserver/backup/
root@webserver:/home/webserver# ls
backup  wazuh
root@webserver:/home/webserver# _
```

Backup Directory.

```
root@webserver:/home/webserver# cd backup/
root@webserver:/home/webserver/backup# ls
wazuh-certificates  wazuh-offline.tar.gz
root@webserver:/home/webserver/backup# ls -l
total 613844
drwxr--r-- 2 webserver webserver      4096 Mar 27 12:00 wazuh-certificates
-rw------- 1 webserver webserver 628567785 Mar 27 11:59 wazuh-offline.tar.gz
root@webserver:/home/webserver/backup#
```

Now move back to primary directory and continue the installation.

"tar xf wazuh-offline.tar.gz"

```
root@webserver:/home/webserver/wazuh# tar xf wazuh-offline.tar.gz
root@webserver:/home/webserver/wazuh#
```

"Installing the Wazuh indexer"

"dpkg -i ./wazuh-offline/wazuh-packages/wazuh-indexer*.deb"

```
root@webserver: /home/webserver/wazuh
root@webserver:/home/webserver/wazuh# dpkg -i ./wazuh-offline/wazuh-packages/wazuh-indexer*.deb
Selecting previously unselected package wazuh-indexer.
(Reading database ... 73929 files and directories currently installed.)
Preparing to unpack .../wazuh-indexer_4.3.10-1_amd64.deb ...
Creating wazuh-indexer group... OK
Creating wazuh-indexer user... OK
Unpacking wazuh-indexer (4.3.10-1) ...
Setting up wazuh-indexer (4.3.10-1) ...
Created opensearch keystore in /etc/wazuh-indexer/opensearch.keystore
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
root@webserver:/home/webserver/wazuh#
```

Run the following commands replacing <indexer-node-name> with the name of the Wazuh indexer node you are configuring as defined in config.yml. For example, node-1. This deploys the SSL certificates to encrypt communications between the Wazuh central components.

"NODE_NAME=node-1"

"mkdir /etc/wazuh-indexer/certs"

"mv -n wazuh-certificates/$NODE_NAME.pem /etc/wazuh-indexer/certs/indexer.pem"

"mv -n wazuh-certificates/$NODE_NAME-key.pem /etc/wazuh-indexer/certs/indexer-key.pem"

"mv wazuh-certificates/admin-key.pem /etc/wazuh-indexer/certs/"

“mv wazuh-certificates/admin.pem /etc/wazuh-indexer/certs/”
“cp wazuh-certificates/root-ca.pem /etc/wazuh-indexer/certs/”
“chmod 500 /etc/wazuh-indexer/certs”
“chmod 400 /etc/wazuh-indexer/certs/*”
“chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs”

```
root@webserver:/home/webserver/wazuh# NODE_NAME=node-1
root@webserver:/home/webserver/wazuh# mkdir /etc/wazuh-indexer/certs
root@webserver:/home/webserver/wazuh# mv -n wazuh-certificates/$NODE_NAME.pem /etc/wazuh-indexer/certs/indexer.pem
root@webserver:/home/webserver/wazuh# mv -n wazuh-certificates/$NODE_NAME-key.pem /etc/wazuh-indexer/certs/indexer-key.pem
root@webserver:/home/webserver/wazuh# mv wazuh-certificates/admin-key.pem /etc/wazuh-indexer/certs/
root@webserver:/home/webserver/wazuh# mv wazuh-certificates/admin.pem /etc/wazuh-indexer/certs/
root@webserver:/home/webserver/wazuh# cp wazuh-certificates/root-ca.pem /etc/wazuh-indexer/certs/
root@webserver:/home/webserver/wazuh# chmod 500 /etc/wazuh-indexer/certs
root@webserver:/home/webserver/wazuh# chmod 400 /etc/wazuh-indexer/certs/*
root@webserver:/home/webserver/wazuh# chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
root@webserver:/home/webserver/wazuh#
```

Here you move the node certificate and key files, such as node-1.pem and node-1-key.pem, to their corresponding certs folder.
Edit /etc/wazuh-indexer/opensearch.yml and replace the following values:
network.host: Sets the address of this node for both HTTP and transport traffic. The node will bind to this address and will also use it as its publish address. Accepts an IP address or a hostname.
Use the same node address set in config.yml to create the SSL certificates.
node.name: Name of the Wazuh indexer node as defined in the config.yml file. For example, node-1.
cluster.initial_master_nodes: List of the names of the master-eligible nodes. These names are defined in the config.yml file. Uncomment the node-2 and node-3 lines, change the names, or add more lines, according to your config.yml definitions.

```
network.host: "192.168.1.10"
node.name: "node-1"
cluster.initial_master_nodes:
- "node-1"
#- "node-2"
#- "node-3"
cluster.name: "wazuh-cluster"
#discovery.seed_hosts:
#  - "node-1-ip"
#  - "node-2-ip"
#  - "node-3-ip"
node.max_local_storage_nodes: "3"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer

plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.transport.resolve_hostname: false

plugins.security.authcz.admin_dn:
- "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"
plugins.security.check_snapshot_restore_write_privileges: true
plugins.security.enable_snapshot_restore_privilege: true
plugins.security.nodes_dn:
- "CN=node-1,OU=Wazuh,O=Wazuh,L=California,C=US"
#- "CN=node-2,OU=Wazuh,O=Wazuh,L=California,C=US"
#- "CN=node-3,OU=Wazuh,O=Wazuh,L=California,C=US"
plugins.security.restapi.roles_enabled:
- "all_access"
- "security_rest_api_access"

plugins.security.system_indices.enabled: true
plugins.security.system_indices.indices: [".opendistro-alerting-config", ".opendistro-alerting-

### Option to allow Filebeat-oss 7.10.2 to work ###
compatibility.override_main_response_version: true
```

Ctrl+o to save and ctrl+x to exit the editor.

Enable and start the Wazuh indexer service.
“systemctl daemon-reload”
“systemctl enable wazuh-indexer”
“systemctl start wazuh-indexer”

```
root@webserver:/home/webserver/wazuh# systemctl daemon-reload
root@webserver:/home/webserver/wazuh# systemctl enable wazuh-indexer
Synchronizing state of wazuh-indexer.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-indexer
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-indexer.service → /lib/systemd/system/wazuh-indexer.service.
root@webserver:/home/webserver/wazuh# systemctl start wazuh-indexer
root@webserver:/home/webserver/wazuh#
```

When all Wazuh indexer nodes are running, run the Wazuh indexer indexer-security-init.sh script on any Wazuh indexer node to load the new certificates information and start the cluster.
“/usr/share/wazuh-indexer/bin/indexer-security-init.sh”

```
root@webserver:/home/webserver/wazuh# /usr/share/wazuh-indexer/bin/indexer-security-init.sh
Security Admin v7
Will connect to 192.168.1.10:9300 ... done
Connected as CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US
OpenSearch Version: 1.2.4
OpenSearch Security Version: 1.2.4.0
Contacting opensearch cluster 'opensearch' and wait for YELLOW clusterstate ...
Clustername: wazuh-cluster
Clusterstate: GREEN
Number of nodes: 1
Number of data nodes: 1
.opendistro_security index does not exists, attempt to create it ... done (0-all replicas)
Populate config from /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/
Will update '_doc/config' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/config.yml
   SUCC: Configuration for 'config' created or updated
Will update '_doc/roles' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/roles.yml
   SUCC: Configuration for 'roles' created or updated
Will update '_doc/rolesmapping' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/roles_mapping.yml
   SUCC: Configuration for 'rolesmapping' created or updated
Will update '_doc/internalusers' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/internal_users.yml
   SUCC: Configuration for 'internalusers' created or updated
Will update '_doc/actiongroups' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/action_groups.yml
   SUCC: Configuration for 'actiongroups' created or updated
Will update '_doc/tenants' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/tenants.yml
   SUCC: Configuration for 'tenants' created or updated
Will update '_doc/nodesdn' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/nodes_dn.yml
   SUCC: Configuration for 'nodesdn' created or updated
Will update '_doc/whitelist' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/whitelist.yml
   SUCC: Configuration for 'whitelist' created or updated
Will update '_doc/audit' with /usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/audit.yml
   SUCC: Configuration for 'audit' created or updated
Done with success
root@webserver:/home/webserver/wazuh#
```

Run the following command to check that the installation is successful. Note that this command uses localhost, set your Wazuh indexer address if necessary.

“curl -XGET https://localhost:9200 -u admin:admin -k”

```
root@webserver:/home/webserver/wazuh# curl -XGET https://192.168.1.10:9200 -u admin:admin -k
{
  "name" : "node-1",
  "cluster_name" : "wazuh-cluster",
  "cluster_uuid" : "llgadxl3Q36wuUeNYOXPww",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "rpm",
    "build_hash" : "e505b10357c03ae8d26d675172402f2f2144ef0f",
    "build_date" : "2022-01-14T03:38:06.881862Z",
    "build_snapshot" : false,
    "lucene_version" : "8.10.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
```

“Installing the Wazuh server”

Run the following commands to import the Wazuh key and install the Wazuh manager.

“dpkg -i ./wazuh-offline/wazuh-packages/wazuh-manager*.deb”

```
root@webserver:/home/webserver/wazuh# dpkg -i ./wazuh-offline/wazuh-packages/wazuh-manager*.deb
Selecting previously unselected package wazuh-manager.
(Reading database ... 74874 files and directories currently installed.)
Preparing to unpack .../wazuh-manager_4.3.10-1_amd64.deb ...
Unpacking wazuh-manager (4.3.10-1) ...
Setting up wazuh-manager (4.3.10-1) ...
root@webserver:/home/webserver/wazuh#
```

Enable and start the Wazuh manager service

“systemctl daemon-reload”
“systemctl enable wazuh-manager”
“systemctl start wazuh-manager”

```
root@webserver: /home/webserver/wazuh
root@webserver:/home/webserver/wazuh# systemctl daemon-reload
root@webserver:/home/webserver/wazuh# systemctl enable wazuh-manager
Synchronizing state of wazuh-manager.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-manager
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-manager.service → /lib/systemd/system/wazuh-manager.service.
root@webserver:/home/webserver/wazuh# systemctl start wazuh-manager
root@webserver:/home/webserver/wazuh#
```

Run the following command to verify that the Wazuh manager status is active.
“systemctl status wazuh-manager”

```
root@webserver:/home/webserver/wazuh# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-03-27 12:18:49 UTC; 23s ago
    Process: 40542 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 133 (limit: 20362)
     Memory: 580.2M
        CPU: 20.715s
     CGroup: /system.slice/wazuh-manager.service
             ├─40595 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─40634 /var/ossec/bin/wazuh-authd
             ├─40650 /var/ossec/bin/wazuh-db
             ├─40664 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─40667 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─40679 /var/ossec/bin/wazuh-execd
             ├─40693 /var/ossec/bin/wazuh-analysisd
             ├─40754 /var/ossec/bin/wazuh-syscheckd
             ├─40773 /var/ossec/bin/wazuh-remoted
             ├─40805 /var/ossec/bin/wazuh-logcollector
             ├─40827 /var/ossec/bin/wazuh-monitord
             └─40849 /var/ossec/bin/wazuh-modulesd

Mar 27 12:18:40 webserver env[40542]: Started wazuh-db...
Mar 27 12:18:41 webserver env[40542]: Started wazuh-execd...
Mar 27 12:18:42 webserver env[40542]: Started wazuh-analysisd...
Mar 27 12:18:43 webserver env[40542]: Started wazuh-syscheckd...
Mar 27 12:18:44 webserver env[40542]: Started wazuh-remoted...
Mar 27 12:18:46 webserver env[40542]: Started wazuh-logcollector...
Mar 27 12:18:47 webserver env[40542]: Started wazuh-monitord...
Mar 27 12:18:47 webserver env[40542]: Started wazuh-modulesd...
Mar 27 12:18:49 webserver env[40542]: Completed.
Mar 27 12:18:49 webserver systemd[1]: Started Wazuh manager.
root@webserver:/home/webserver/wazuh#
```

“Installing Filebeat”
Run the following command to install Filebeat
“dpkg -i ./wazuh-offline/wazuh-packages/filebeat*.deb”

```
root@webserver:/home/webserver/wazuh# dpkg -i ./wazuh-offline/wazuh-packages/filebeat*.deb
Selecting previously unselected package filebeat.
(Reading database ... 93565 files and directories currently installed.)
Preparing to unpack .../filebeat-oss-7.10.2-amd64.deb ...
Unpacking filebeat (7.10.2) ...
Setting up filebeat (7.10.2) ...
root@webserver:/home/webserver/wazuh#
```

Move a copy of the configuration files to the appropriate location. Ensure to type “yes” at the prompt to overwrite /etc/filebeat/filebeat.yml

“cp ./wazuh-offline/wazuh-files/filebeat.yml /etc/filebeat/ &&\ cp ./wazuh-offline/wazuh-files/wazuh-template.json /etc/filebeat/ &&\ chmod go+r /etc/filebeat/wazuh-template.json”

```
root@webserver:/home/webserver/wazuh# cp ./wazuh-offline/wazuh-files/filebeat.yml /etc/filebeat/ &&\
cp ./wazuh-offline/wazuh-files/wazuh-template.json /etc/filebeat/ &&\
chmod go+r /etc/filebeat/wazuh-template.json
root@webserver:/home/webserver/wazuh#
```

Edit /etc/filebeat/wazuh-template.json and change to "1" the value for "index.number_of_shards" for a single-node installation. This value can be changed based on the user requirement when performing a distributed installation.

“nano /etc/filebeat/wazuh-template.json”

```
root@webserver: /home/webserver/wazuh
  GNU nano 6.2
{
  "order": 0,
  "index_patterns": [
    "wazuh-alerts-4.x-*",
    "wazuh-archives-4.x-*"
  ],
  "settings": {
    "index.refresh_interval": "5s",
    "index.number_of_shards": "1",
    "index.number_of_replicas": "0",
    "index.auto_expand_replicas": "0-1",
    "index.mapping.total_fields.limit": 10000,
    "index.query.default_field": [
```

Edit the /etc/filebeat/filebeat.yml configuration file and replace the following value:

hosts: The list of Wazuh indexer nodes to connect to. You can use either IP addresses or hostnames. By default, the host is set to localhost hosts: ["127.0.0.1:9200"]. Replace it with your Wazuh indexer address accordingly.

If you have more than one Wazuh indexer node, you can separate the addresses using commas. For example, hosts: ["10.0.0.1:9200", "10.0.0.2:9200", "10.0.0.3:9200"]

“nano /etc/filebeat/filebeat.yml”

```
root@webserver: /home/webserver/wazuh
  GNU nano 6.2
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["192.168.1.10:9200"]
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/filebeat.pem"
  ssl.key: "/etc/filebeat/certs/filebeat-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false
```

Create a Filebeat keystore to securely store authentication credentials.
"filebeat keystore create"

```
root@webserver:/home/webserver/wazuh# filebeat keystore create
Created filebeat keystore
root@webserver:/home/webserver/wazuh#
```

Add the username and password admin:admin to the secrets keystore.
"echo admin | filebeat keystore add username --stdin --force"
"echo admin | filebeat keystore add password --stdin --force"

```
root@webserver:/home/webserver/wazuh# echo admin | filebeat keystore add username --stdin --force
Successfully updated the keystore
root@webserver:/home/webserver/wazuh# echo admin | filebeat keystore add password --stdin --force
Successfully updated the keystore
root@webserver:/home/webserver/wazuh#
```

Install the Wazuh module for Filebeat.
"tar -xzf ./wazuh-offline/wazuh-files/wazuh-filebeat-0.2.tar.gz -C /usr/share/filebeat/module"

```
root@webserver:/home/webserver/wazuh# tar -xzf ./wazuh-offline/wazuh-files/wazuh-filebeat-0.2.tar.gz -C /usr/share/filebeat/module
root@webserver:/home/webserver/wazuh#
```

Replace <server-node-name> with your Wazuh server node certificate name, the same used in config.yml when creating the certificates. For example, wazuh-1. Then, move the certificates to their corresponding location.
"NODE_NAME=Wazuh-1"
"mkdir /etc/filebeat/certs"
"mv -n wazuh-certificates/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem"
"mv -n wazuh-certificates/$NODE_NAME-key.pem /etc/filebeat/certs/filebeat-key.pem"
"cp wazuh-certificates/root-ca.pem /etc/filebeat/certs/"
"chmod 500 /etc/filebeat/certs"
"chmod 400 /etc/filebeat/certs/*"
"chown -R root:root /etc/filebeat/certs"

```
root@webserver:/home/webserver/wazuh# NODE_NAME=wazuh-1
root@webserver:/home/webserver/wazuh# mkdir /etc/filebeat/certs
root@webserver:/home/webserver/wazuh# mv -n wazuh-certificates/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem
root@webserver:/home/webserver/wazuh# mv -n wazuh-certificates/$NODE_NAME-key.pem /etc/filebeat/certs/filebeat-key.pem
root@webserver:/home/webserver/wazuh# cp wazuh-certificates/root-ca.pem /etc/filebeat/certs/
root@webserver:/home/webserver/wazuh# chmod 500 /etc/filebeat/certs
root@webserver:/home/webserver/wazuh# chmod 400 /etc/filebeat/certs/*
root@webserver:/home/webserver/wazuh# chown -R root:root /etc/filebeat/certs
root@webserver:/home/webserver/wazuh#
```

Enable and start the Filebeat service.
“systemctl daemon-reload”
“systemctl enable filebeat”
“systemctl start filebeat”

```
root@webserver: /home/webserver/wazuh
root@webserver:/home/webserver/wazuh# systemctl daemon-reload
root@webserver:/home/webserver/wazuh# systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
root@webserver:/home/webserver/wazuh# systemctl start filebeat
root@webserver:/home/webserver/wazuh#
```

Run the following command to make sure Filebeat is successfully installed
“filebeat test output”

```
root@webserver:/home/webserver/wazuh# filebeat test output
elasticsearch: https://192.168.1.10:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 192.168.1.10
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.10.2
root@webserver:/home/webserver/wazuh#
```

To check the number of shards that have been configured, you can run the following command. Note that this command uses localhost, set your Wazuh indexer address if necessary.
“curl -k -u admin:admin "https://192.168.1.10:9200/_template/wazuh?pretty&filter_path=wazuh.settings.index.number_of_shards"”

```
root@webserver:/home/webserver/wazuh# curl -k -u admin:admin "https://192.168.1.10:9200/_template/wazuh?pretty&filter_path=wazuh.settings.index.number_of_shards"
{
  "wazuh" : {
    "settings" : {
      "index" : {
        "number_of_shards" : "1"
      }
    }
  }
}
root@webserver:/home/webserver/wazuh#
```

“Installing the Wazuh dashboard”
Run the following commands to install the Wazuh dashboard.
“dpkg -i ./wazuh-offline/wazuh-packages/wazuh-dashboard*.deb”

```
root@webserver:/home/webserver/wazuh# dpkg -i ./wazuh-offline/wazuh-packages/wazuh-dashboard*.deb
Selecting previously unselected package wazuh-dashboard.
(Reading database ... 93884 files and directories currently installed.)
Preparing to unpack .../wazuh-dashboard_4.3.10-1_amd64.deb ...
Creating wazuh-dashboard group... OK
Creating wazuh-dashboard user... OK
Unpacking wazuh-dashboard (4.3.10-1) ...
Setting up wazuh-dashboard (4.3.10-1) ...
root@webserver:/home/webserver/wazuh#
```

Replace <dashboard-node-name> with your Wazuh dashboard node name, the same used in config.yml to create the certificates. For example, dashboard. Then, move the certificates to their corresponding location.
“NODE_NAME=dashboard”
“mkdir /etc/wazuh-dashboard/certs”
“mv -n wazuh-certificates/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem”
“mv -n wazuh-certificates/$NODE_NAME-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem”
“cp wazuh-certificates/root-ca.pem /etc/wazuh-dashboard/certs/”
“chmod 500 /etc/wazuh-dashboard/certs”
“chmod 400 /etc/wazuh-dashboard/certs/*”
“chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs”

```
root@webserver:/home/webserver/wazuh# NODE_NAME=dashboard
root@webserver:/home/webserver/wazuh# mkdir /etc/wazuh-dashboard/certs
root@webserver:/home/webserver/wazuh# mv -n wazuh-certificates/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem
root@webserver:/home/webserver/wazuh# mv -n wazuh-certificates/$NODE_NAME-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem
root@webserver:/home/webserver/wazuh# cp wazuh-certificates/root-ca.pem /etc/wazuh-dashboard/certs/
root@webserver:/home/webserver/wazuh# chmod 500 /etc/wazuh-dashboard/certs
root@webserver:/home/webserver/wazuh# chmod 400 /etc/wazuh-dashboard/certs/*
root@webserver:/home/webserver/wazuh# chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
root@webserver:/home/webserver/wazuh#
```

Edit the /etc/wazuh-dashboard/opensearch_dashboards.yml file and replace the following values:
server.host: This setting specifies the host of the back end server. To allow remote users to connect, set the value to the IP address or DNS name of the Wazuh dashboard. The value 0.0.0.0 will accept all the available IP addresses of the host.
opensearch.hosts: The URLs of the Wazuh indexer instances to use for all your queries. The Wazuh dashboard can be configured to connect to multiple Wazuh indexer nodes in the same cluster. The addresses of the nodes can be separated by commas. For example, ["https://10.0.0.2:9200", "https://10.0.0.3:9200","https://10.0.0.4:9200"]
“nano /etc/wazuh-dashboard/opensearch_dashboards.yml**”**

```
root@webserver: /home/webserver/wazuh
  GNU nano 6.2
server.host: 0.0.0.0
server.port: 443
opensearch.hosts: https://192.168.1.10:9200
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersWhitelist: ["securitytenant","Authorization"]
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]
server.ssl.enabled: true
server.ssl.key: "/etc/wazuh-dashboard/certs/dashboard-key.pem"
server.ssl.certificate: "/etc/wazuh-dashboard/certs/dashboard.pem"
opensearch.ssl.certificateAuthorities: ["/etc/wazuh-dashboard/certs/root-ca.pem"]
uiSettings.overrides.defaultRoute: /app/wazuh
```

Enable and start the Wazuh dashboard.
"systemctl daemon-reload"
"systemctl enable wazuh-dashboard"
"systemctl start wazuh-dashboard"

```
root@webserver:/home/webserver/wazuh# systemctl daemon-reload
root@webserver:/home/webserver/wazuh# systemctl enable wazuh-dashboard
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-dashboard.service → /etc/systemd/system/wazuh-dashboard.service.
root@webserver:/home/webserver/wazuh# systemctl start wazuh-dashboard
root@webserver:/home/webserver/wazuh#
```

Only for distributed deployments:
Edit the file **/usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml** and replace the url value with the IP address or hostname of the Wazuh server master node.

"nano /usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml "

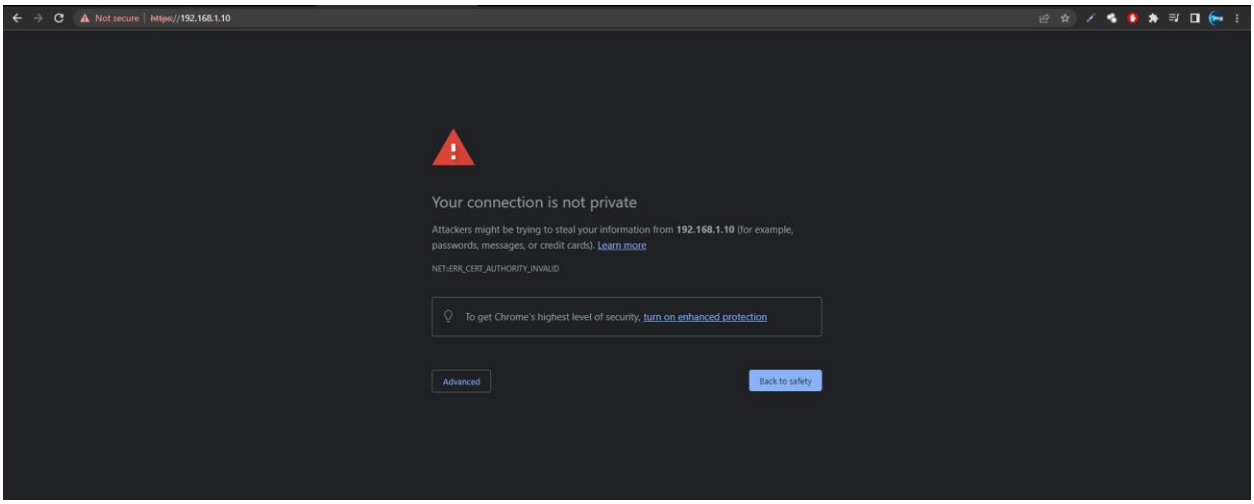
```
#
#---------------------------------- Wazuh hosts ----------------------------------
#
# The following configuration is the default structure to define a host.
#
# hosts:
#   # Host ID / name,
#   - env-1:
#       # Host URL
#       url: https://env-1.example
#       # Host / API port
#       port: 55000
#       # Host / API username
#       username: wazuh-wui
#       # Host / API password
#       password: wazuh-wui
#       # Use RBAC or not. If set to true, the username must be "wazuh-wui".
#       run_as: true
#   - env-2:
#       url: https://env-2.example
#       port: 55000
#       username: wazuh-wui
#       password: wazuh-wui
#       run_as: true

hosts:
  - default:
      url: https://192.168.1.10
      port: 55000
      username: wazuh-wui
      password: wazuh-wui
      run_as: false
```

Run the following command to verify the Wazuh dashboard service is active
"systemctl status wazuh-dashboard"

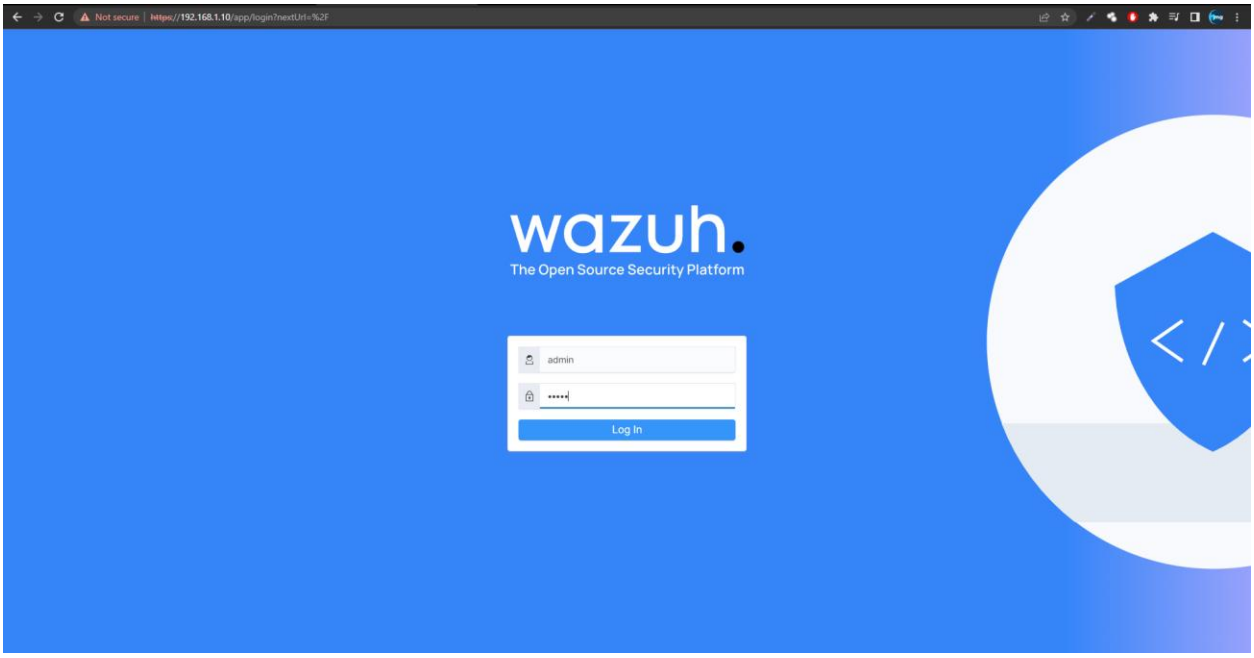
```
root@webserver:/home/webserver/wazuh# systemctl status wazuh-dashboard
● wazuh-dashboard.service - wazuh-dashboard
     Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-03-27 12:45:06 UTC; 2min 44s ago
   Main PID: 42716 (node)
      Tasks: 11 (limit: 20362)
     Memory: 119.8M
        CPU: 4.897s
     CGroup: /system.slice/wazuh-dashboard.service
             └─42716 /usr/share/wazuh-dashboard/bin/../node/bin/node --no-warnings --max-http-header-size=65536 --unhandled-rejections=warn /usr/share/wazuh-dashboard/bin/../src/cli/dist -c /etc/wazuh-dashboard/opensearch_dashboards.yml

Mar 27 12:45:09 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:09Z","tags":["info","savedobjects-service"],"pid":42716,"message":"Waiting until all OpenSearch nodes are compatible with OpenSearch Da
Mar 27 12:45:09 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:09Z","tags":["info","savedobjects-service"],"pid":42716,"message":"Starting saved objects migrations"}
Mar 27 12:45:09 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:09Z","tags":["info","savedobjects-service"],"pid":42716,"message":"Creating index .kibana_1."}
Mar 27 12:45:09 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:09Z","tags":["info","savedobjects-service"],"pid":42716,"message":"Pointing alias .kibana to .kibana_1."}
Mar 27 12:45:09 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:09Z","tags":["info","savedobjects-service"],"pid":42716,"message":"Finished in 235ms."}
Mar 27 12:45:09 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:09Z","tags":["info","plugins-system"],"pid":42716,"message":"Starting [42] plugins: [alertingDashboards,usageCollection,opensearchDashb
Mar 27 12:45:10 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:10Z","tags":["listening","info"],"pid":42716,"message":"Server running at https://0.0.0.0:443"}
Mar 27 12:45:10 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:10Z","tags":["info","http","server","OpenSearchDashboards"],"pid":42716,"message":"http server running at https://0.0.0.0:443"}
Mar 27 12:45:10 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:10Z","tags":["error","opensearch","data"],"pid":42716,"message":"[ResponseError]: Response Error"}
Mar 27 12:45:10 webserver opensearch-dashboards[42716]: {"type":"log","@timestamp":"2023-03-27T12:45:10Z","tags":["error","opensearch","data"],"pid":42716,"message":"[ResponseError]: Response Error"}
lines 1-20/20 (END)
```

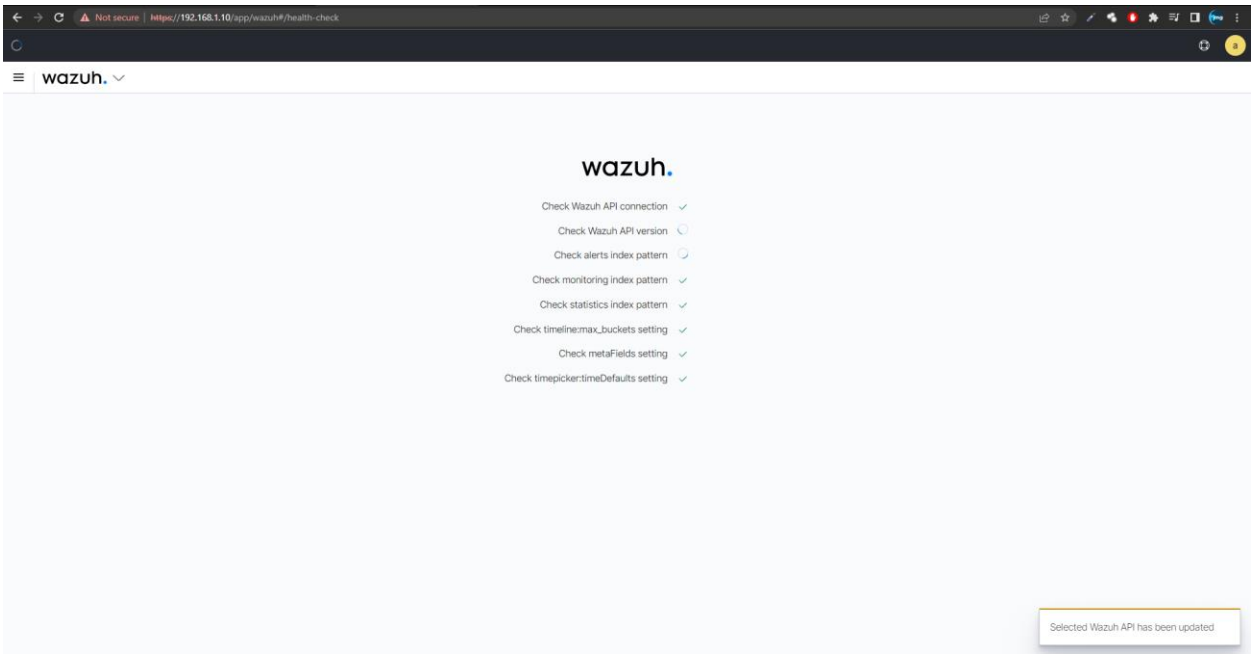
Press q for exit the loop file.

Now open the web-browser and enter the following url
https://192.168.125.104/

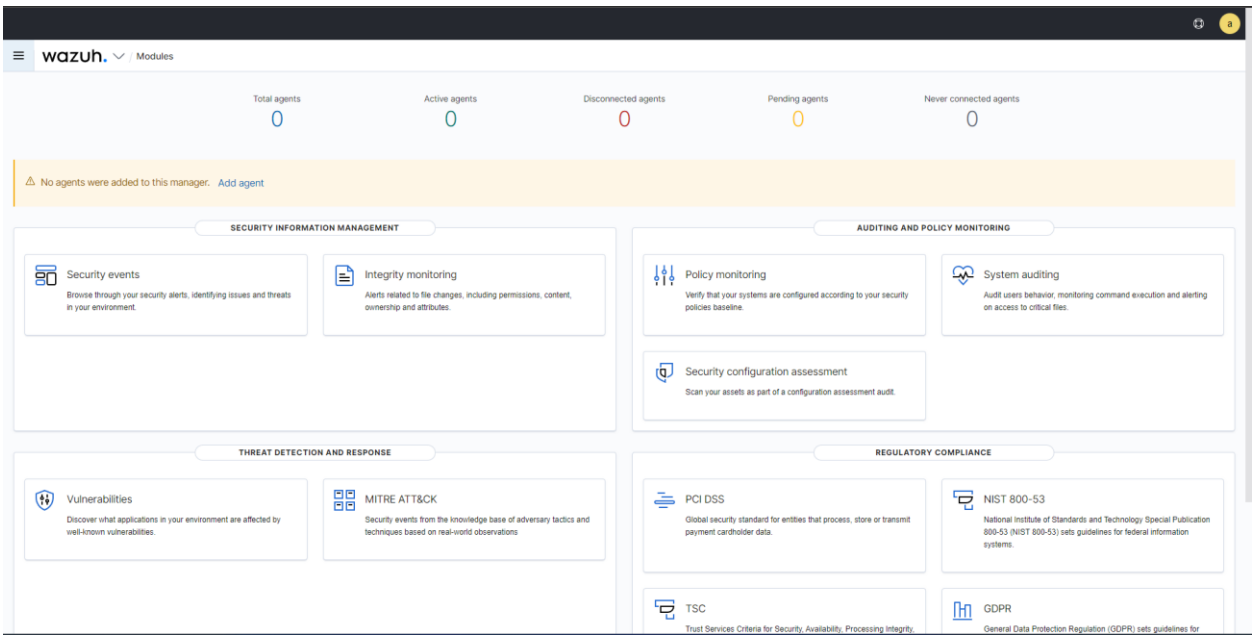


Click on advance and then proceed for unsafe.





Wazuh-Dashboard



4.3. Wazuh Agent Configuration

Click on windows installer and run it after installing. It will found at following path.
"C:\Program Files (x86)\ossec-agent\wazuh-agent.exe"
Now move back to Ubuntu server and apply following command.
"cd /var/ossec/bin/"

```
root@webserver:/home/webserver/wazuh# cd /var/ossec/bin/
root@webserver:/var/ossec/bin# ls
agent_control  agent_upgrade  cluster_control  verify-agent-conf  wazuh-analysisd  wazuh-authd     wazuh-control   wazuh-db   wazuh-execd       wazuh-logcollector  wazuh-logtest-legacy  wazuh
agent_groups   clear_stats    manage_agents    wazuh-agentlessd   wazuh-apid       wazuh-clusterd  wazuh-csyslogd  wazuh-dbd  wazuh-integratord  wazuh-logtest       wazuh-maild           wazuh
```

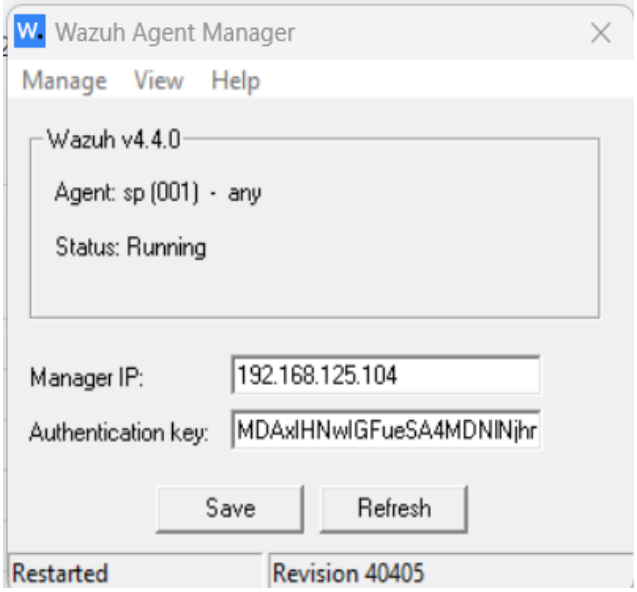
"./manage_agents"

```
   (Q)uit.
Choose your action: A,E,L,R or Q: A

- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
   * A name for the new agent: sahil
   * The IP Address of the new agent: any
Confirm adding it?(y/n): y
2023/04/06 14:00:56 manage_agents: WARNING: 9008: Duplicate name


****************************************
* Wazuh v4.4.0 Agent manager.          *
* The following options are available: *
****************************************
   (A)dd an agent (A).
   (E)xtract key for an agent (E).
   (L)ist already added agents (L).
   (R)emove an agent (R).
   (Q)uit.
Choose your action: A,E,L,R or Q:
```

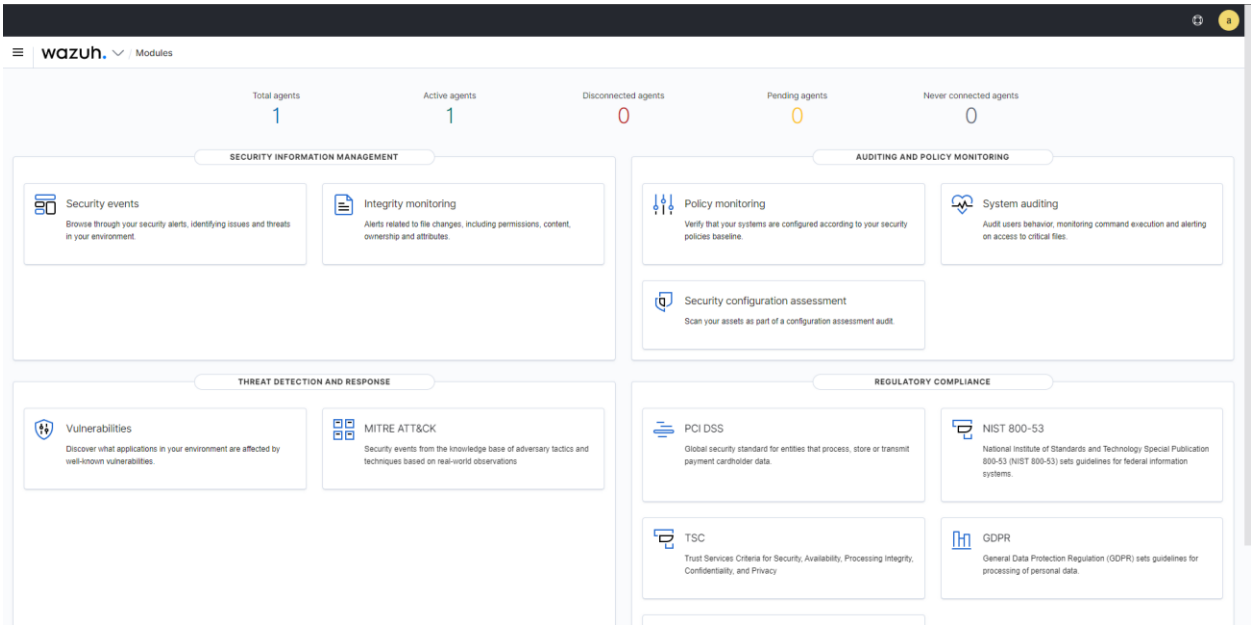
After adding the agent we have to extract the key value for particular agents.
For that press E
Select the agent by entering the index value.

Now move back to agent installer.

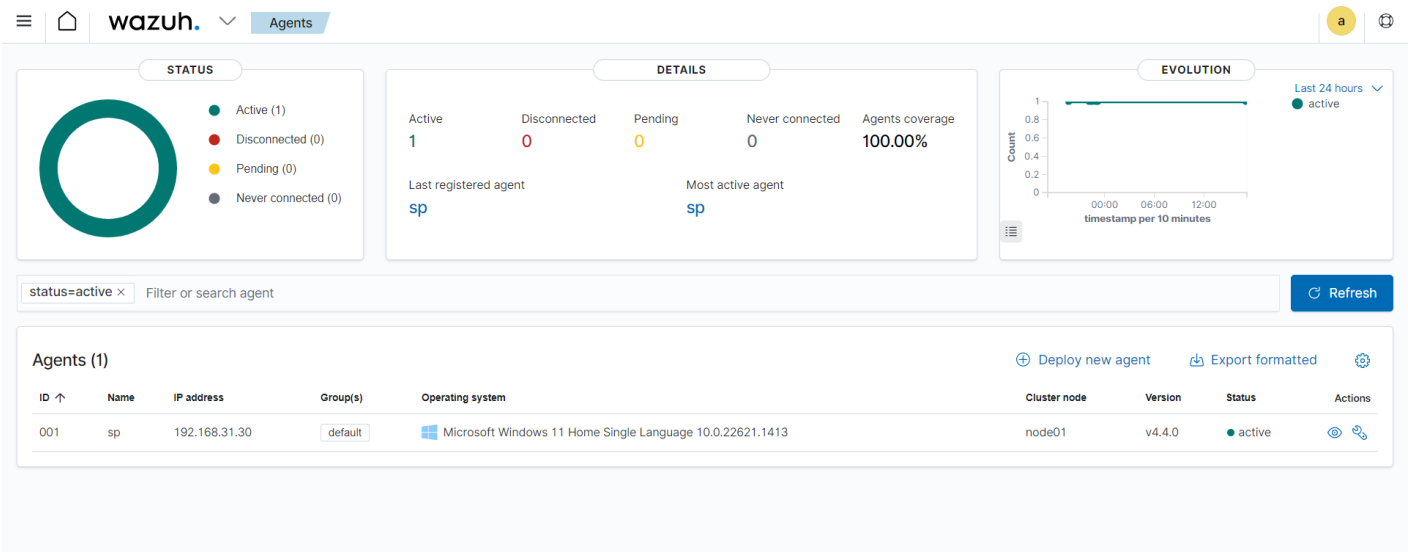
Enter the Ip Address and key vale and save it.



Now move to web-browser and reload the Wazuh-server page.

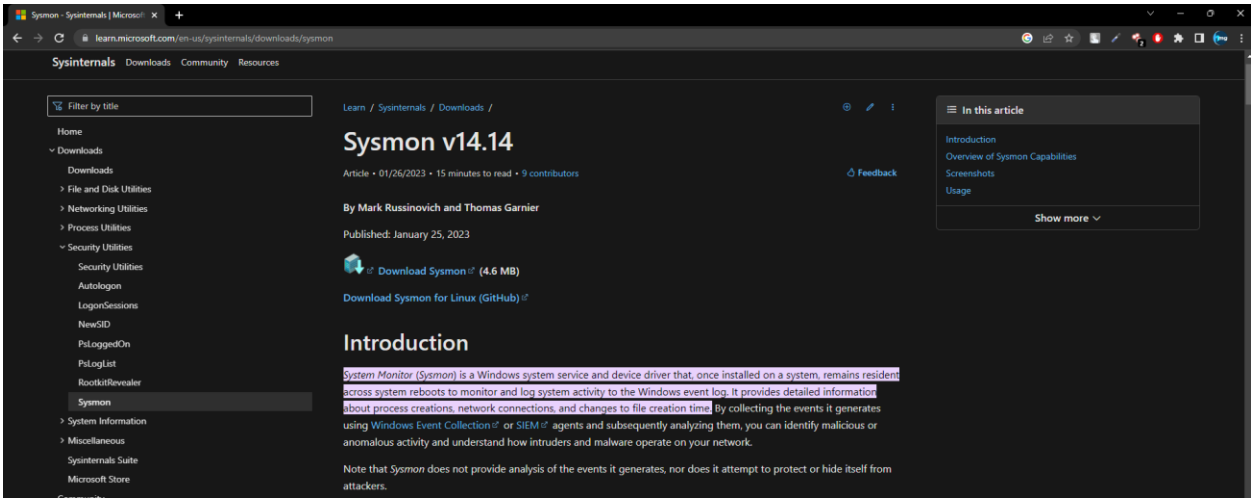


Agent being install and in running state.

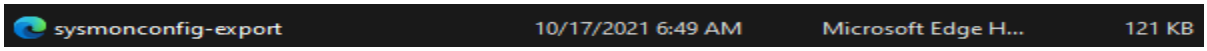

wazuh.
Agents
STATUS
Active (1)
Disconnected (0)
Pending (0)
Never connected (0)
DETAILS
Active
1
Disconnected
0
Pending
0
Never connected
0
Agents coverage
100.00%
Last registered agent
sp
Most active agent
sp
EVOLUTION
Last 24 hours
active
Count
timestamp per 10 minutes
status=active
Filter or search agent
Refresh
Agents (1)
Deploy new agent
Export formatted
ID
Name
IP address
Group(s)
Operating system
Cluster node
Version
Status
Actions
001
sp
192.168.31.30
default
Microsoft Windows 11 Home Single Language 10.0.22621.1413
node01
v4.4.0
active

4.4. Sysmon setup for windows machine.

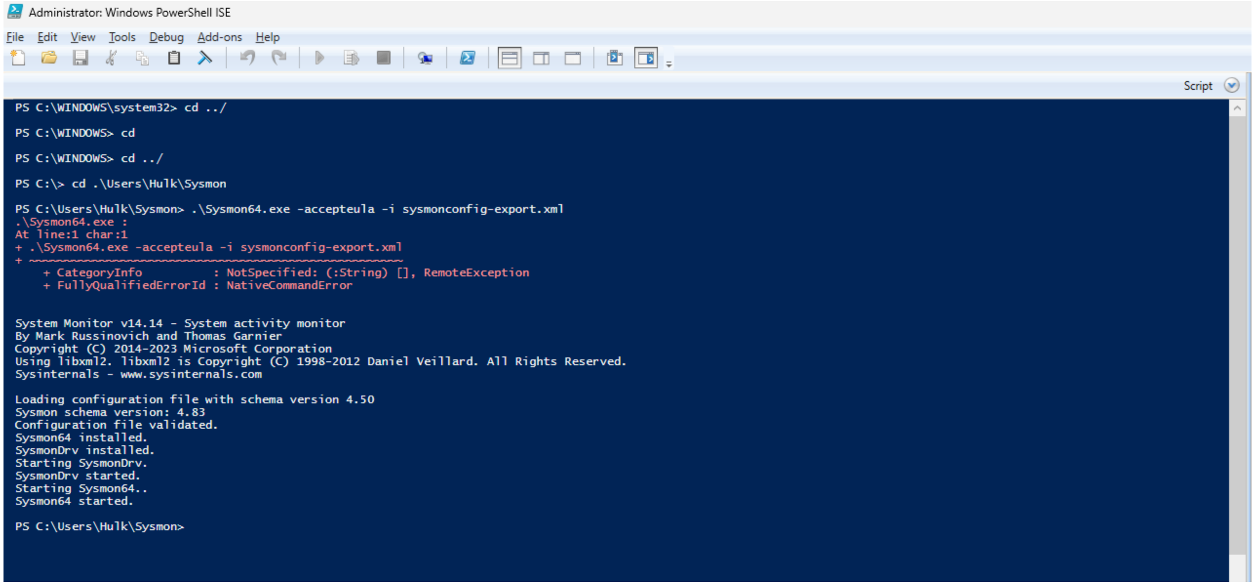
First we have to download the sysinternal and configure in the c drive from Microsoft official website.



After Unizping the file we have to download one more file for configuration for that open the followingurl **https://github.com/SwiftOnSecurity/sysmon-config.**
Now unzip the file and copy only **"sysmonconfig-export.xml"** to c drive wher we had place the Sysmon file.

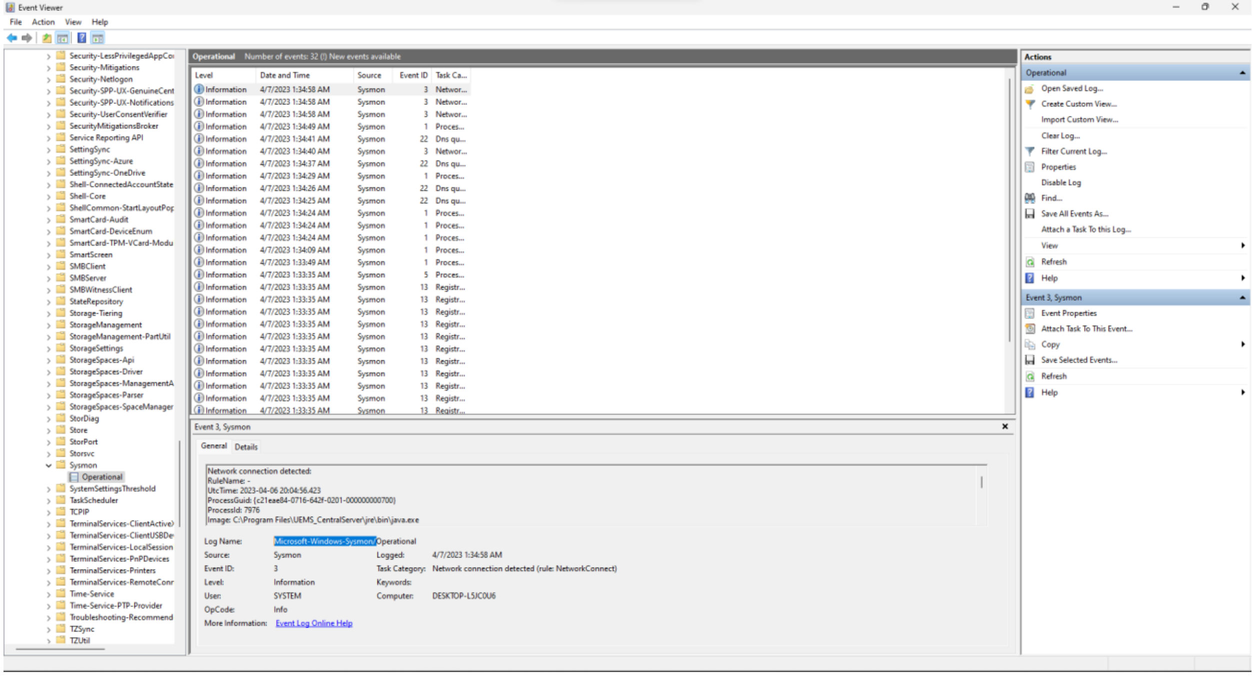


Now open power-shell and apply the following command.
".\Sysmon64.exe -accepteula -i sysmonconfig-export.xml"



```
PS C:\WINDOWS\system32> cd ../

PS C:\WINDOWS> cd

PS C:\WINDOWS> cd ../

PS C:\> cd .\Users\Hulk\Sysmon

PS C:\Users\Hulk\Sysmon> .\Sysmon64.exe -accepteula -i sysmonconfig-export.xml
.\Sysmon64.exe :
At line:1 char:1
+ .\Sysmon64.exe -accepteula -i sysmonconfig-export.xml
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError


System Monitor v14.14 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2023 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.50
Sysmon schema version: 4.83
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.

PS C:\Users\Hulk\Sysmon>
```

For verification we have to open event viewer and move to following location.
Application and service logs → Microsoft → Windows → Sysmon → Operational.
If you get logs here means your configurations has been done successfully.



Operational Number of events: 32 (!) New events available

Level	Date and Time	Source	Event ID	Task Ca...
Information	4/7/2023 1:34:58 AM	Sysmon	3	Networ...
Information	4/7/2023 1:34:58 AM	Sysmon	3	Networ...
Information	4/7/2023 1:34:58 AM	Sysmon	3	Networ...
Information	4/7/2023 1:34:49 AM	Sysmon	1	Proces...
Information	4/7/2023 1:34:41 AM	Sysmon	22	Dns qu...
Information	4/7/2023 1:34:40 AM	Sysmon	3	Networ...
Information	4/7/2023 1:34:37 AM	Sysmon	22	Dns qu...
Information	4/7/2023 1:34:29 AM	Sysmon	1	Proces...
Information	4/7/2023 1:34:26 AM	Sysmon	22	Dns qu...
Information	4/7/2023 1:34:25 AM	Sysmon	22	Dns qu...
Information	4/7/2023 1:34:24 AM	Sysmon	1	Proces...
Information	4/7/2023 1:34:24 AM	Sysmon	1	Proces...
Information	4/7/2023 1:34:24 AM	Sysmon	1	Proces...
Information	4/7/2023 1:34:09 AM	Sysmon	1	Proces...
Information	4/7/2023 1:33:49 AM	Sysmon	1	Proces...
Information	4/7/2023 1:33:35 AM	Sysmon	5	Proces...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...
Information	4/7/2023 1:33:35 AM	Sysmon	13	Registr...

Event 3, Sysmon

General Details

```
Network connection detected:
RuleName: -
UtcTime: 2023-04-06 20:04:56.423
ProcessGuid: {c21eae84-0716-642f-0201-000000000700}
ProcessId: 7976
Image: C:\Program Files\UEMS_CentralServer\jre\bin\java.exe
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	4/7/2023 1:34:58 AM
Event ID:	3	Task Category:	Network connection detected (rule: NetworkConnect)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	DESKTOP-L5JC0U6
OpCode:	Info		
More Information:	Event Log Online Help		

4.5. Sysmon rule on Wazuh dashboard.

Adding the Rule in Wazuh dashboard.

Wazuh>Management>Groups>default>files>config.yml

Add the following rule in file.

```
"<agent_config>
        <!-- Shared agent configuration here -->
        <client_buffer>
                <!-- Agent buffer options -->
                <disable>no</disable>
                <queue_size>100000</queue_size>
                <events_per_second>1000</events_per_second>
        </client_buffer>
        <localfile>
                <location>Microsoft-Windows-Windows Defender/Operational</location>
                <log_format>eventchannel</log_format>
                <location>Security</location>
                <log_format>eventlog</log_format>
                <location>Microsoft-Windows-Sysmon/Operational</location>
                <log_format>eventchannel</log_format>
        </localfile>
</agent_config>"
```

wazuh. / Management / Groups

agent.conf of default group

Save

```
<agent_config>
    <!-- Shared agent configuration here -->
    <client_buffer>
        <!-- Agent buffer options -->
        <disable>no</disable>
        <queue_size>100000</queue_size>
        <events_per_second>1000</events_per_second>
    </client_buffer>
    <localfile>
        <location>Microsoft-Windows-Windows Defender/Operational</location>
        <log_format>eventchannel</log_format>
        <location>Security</location>
        <log_format>eventlog</log_format>
        <location>Microsoft-Windows-Sysmon/Operational</location>
        <log_format>eventchannel</log_format>
    </localfile>
</agent_config>
```

CHAPTER: 8 REFERENCES

CHAPTER 8 REFERENCES

- https://www.ipindia.gov.in/
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3217699/
- https://www.ies.gov.in/pdfs/why-India-needs-to-urgently-invest-in-its-IPR-ecosystem-16th-Aug-2022.pdf
- http://www.sric.iitkgp.ac.in/docss/iitkgpipguide.pdf

IBM

ORIGINALITY REPORT

11%	4%	0%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Ganpat University Student Paper	9%
2	www.nerc.com Internet Source	1%
3	www.slideshare.net Internet Source	1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		