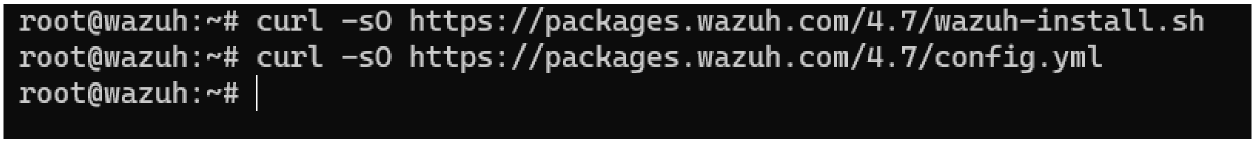
Wazuh Installation

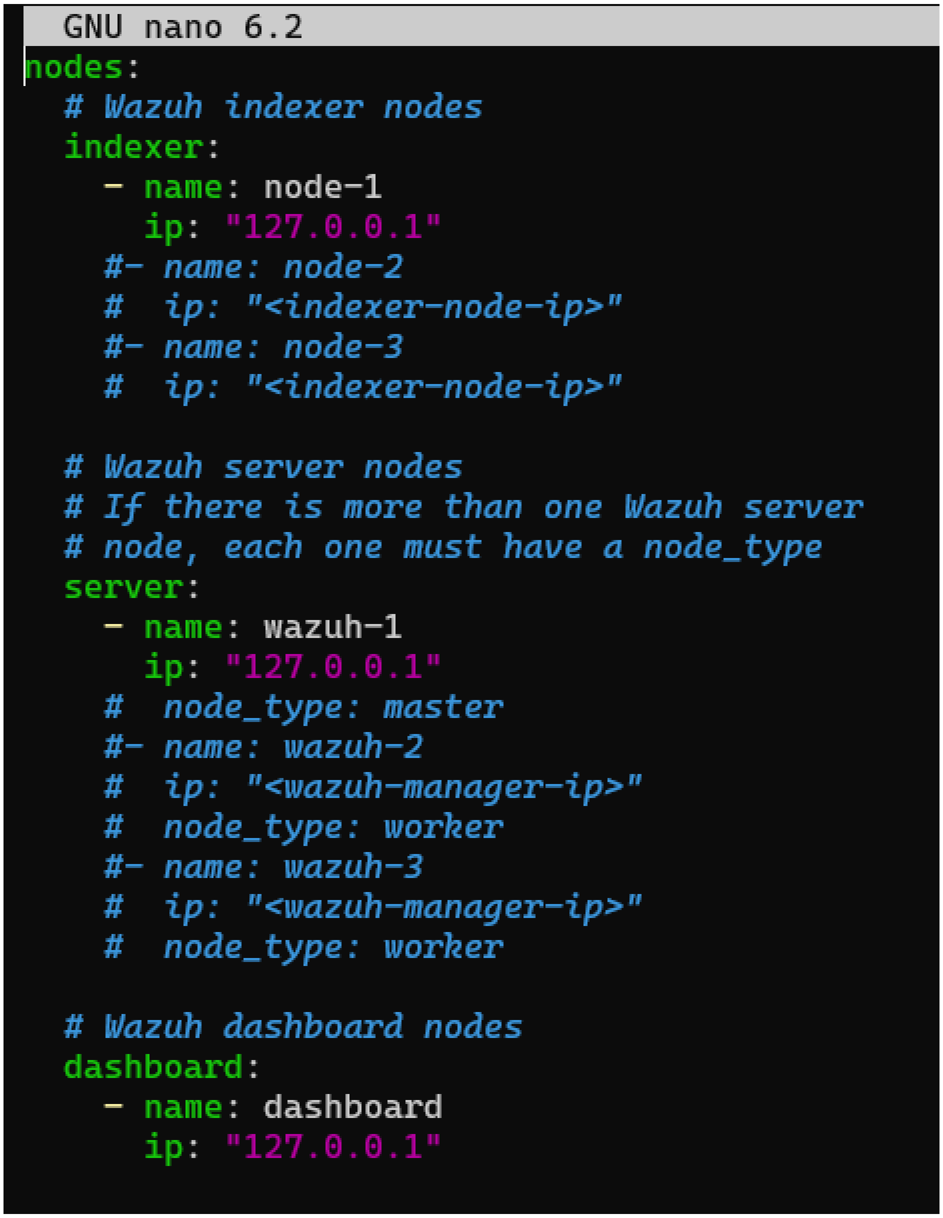
## Download the Wazuh installation assistant and the configuration file.

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh

curl -sO https://packages.wazuh.com/4.7/config.yml
```



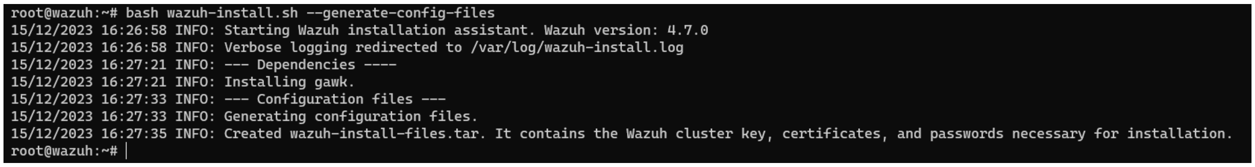
```
root@wazuh:~# curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
root@wazuh:~# curl -sO https://packages.wazuh.com/4.7/config.yml
root@wazuh:~#
```

## Edit config.yml file mention ip like below for all



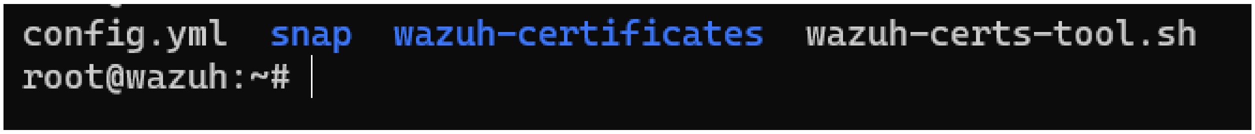
```
  GNU nano 6.2
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: "127.0.0.1"
    #- name: node-2
    #  ip: "<indexer-node-ip>"
    #- name: node-3
    #  ip: "<indexer-node-ip>"

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: wazuh-1
      ip: "127.0.0.1"
    #  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: "127.0.0.1"
```

## Run the assistant with the option `--generate-config-files` to generate the Wazuh cluster key, certificates, and passwords necessary for installation.
→bash wazuh-install.sh --generate-config-files



```
root@wazuh:~# bash wazuh-install.sh --generate-config-files
15/12/2023 16:26:58 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.0
15/12/2023 16:26:58 INFO: Verbose logging redirected to /var/log/wazuh-install.log
15/12/2023 16:27:21 INFO: --- Dependencies ----
15/12/2023 16:27:21 INFO: Installing gawk.
15/12/2023 16:27:33 INFO: --- Configuration files ---
15/12/2023 16:27:33 INFO: Generating configuration files.
15/12/2023 16:27:35 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
root@wazuh:~# 
```

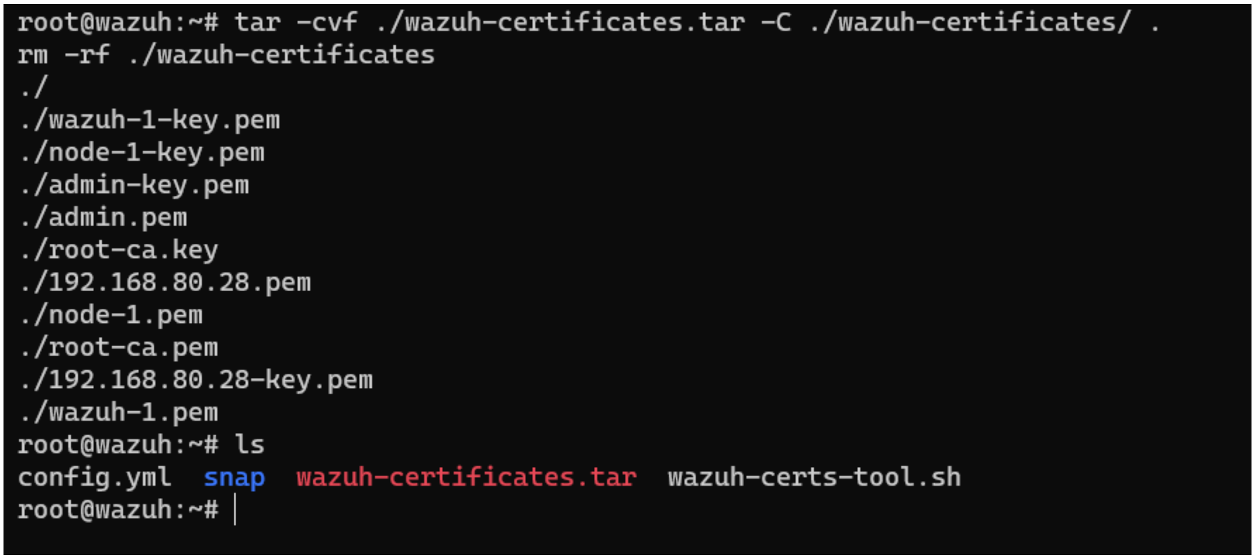
Here will create a tar file after run the script



```
config.yml  snap  wazuh-certificates  wazuh-certs-tool.sh
root@wazuh:~# 
```

## Compress the necessary file after that delete the directory

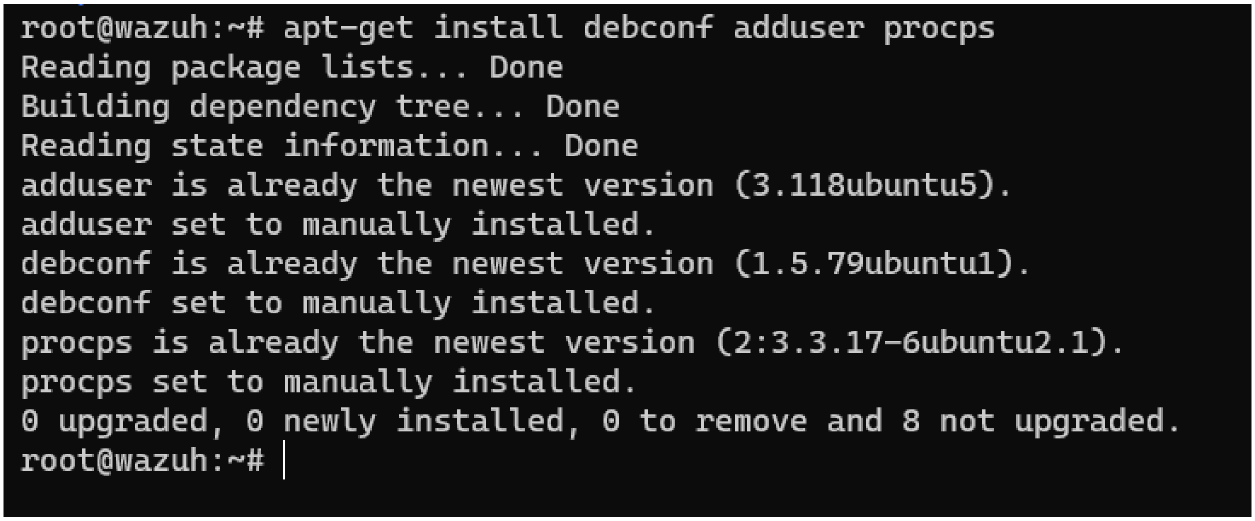
tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates/ .
rm -rf ./wazuh-certificates



```
root@wazuh:~# tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates/ .
rm -rf ./wazuh-certificates
./
./wazuh-1-key.pem
./node-1-key.pem
./admin-key.pem
./admin.pem
./root-ca.key
./192.168.80.28.pem
./node-1.pem
./root-ca.pem
./192.168.80.28-key.pem
./wazuh-1.pem
root@wazuh:~# ls
config.yml  snap  wazuh-certificates.tar  wazuh-certs-tool.sh
root@wazuh:~# 
```

## Install dependencies

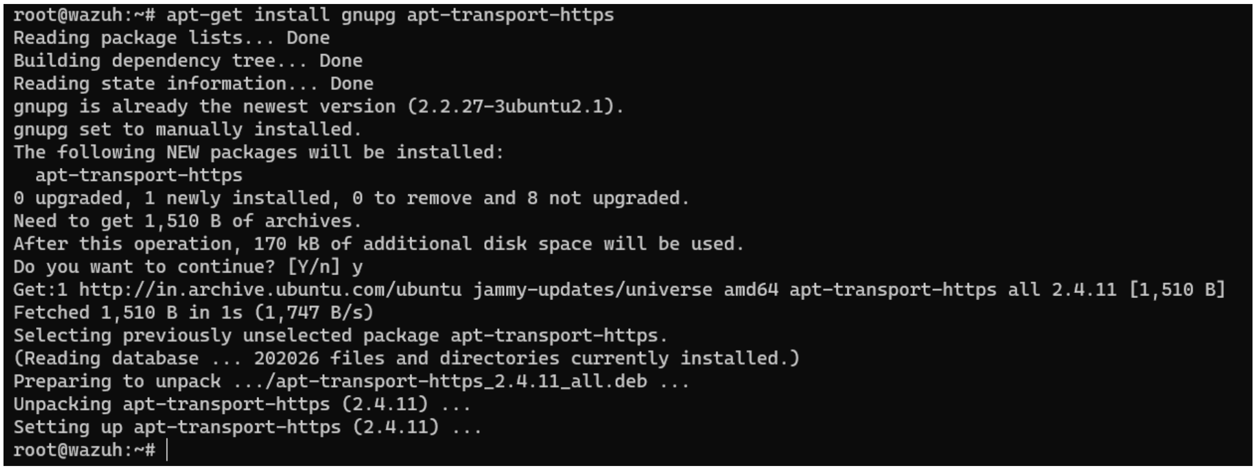
apt-get install debconf adduser procps



```
root@wazuh:~# apt-get install debconf adduser procps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
adduser is already the newest version (3.118ubuntu5).
adduser set to manually installed.
debconf is already the newest version (1.5.79ubuntu1).
debconf set to manually installed.
procps is already the newest version (2:3.3.17-6ubuntu2.1).
procps set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
root@wazuh:~# 
```

## Install dependency for create repository

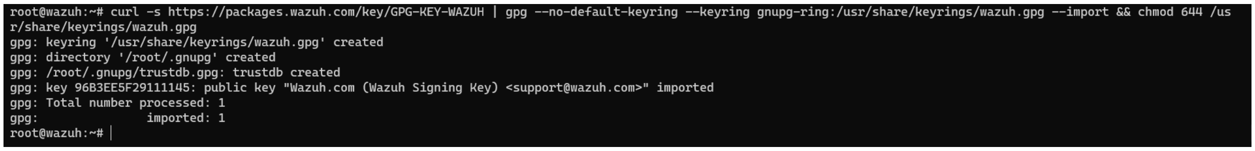
apt-get install gnupg apt-transport-https



```
root@wazuh:~# apt-get install gnupg apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 1,510 B of archives.
After this operation, 170 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.11 [1,510 B]
Fetched 1,510 B in 1s (1,747 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 202026 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.11_all.deb ...
Unpacking apt-transport-https (2.4.11) ...
Setting up apt-transport-https (2.4.11) ...
root@wazuh:~#
```

## Install the GPG key

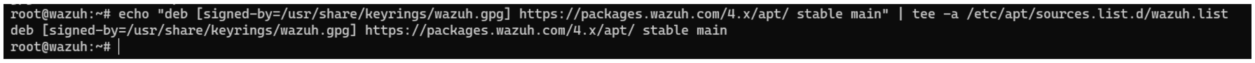
```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg
--no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import
&& chmod 644 /usr/share/keyrings/wazuh.gpg
```



```
root@wazuh:~# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /us
r/share/keyrings/wazuh.gpg
gpg: keyring '/usr/share/keyrings/wazuh.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 96B3EE5F29111145: public key "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
root@wazuh:~#
```

## Add repository

```
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg]
https://packages.wazuh.com/4.x/apt/ stable main" | tee -a
/etc/apt/sources.list.d/wazuh.list
```



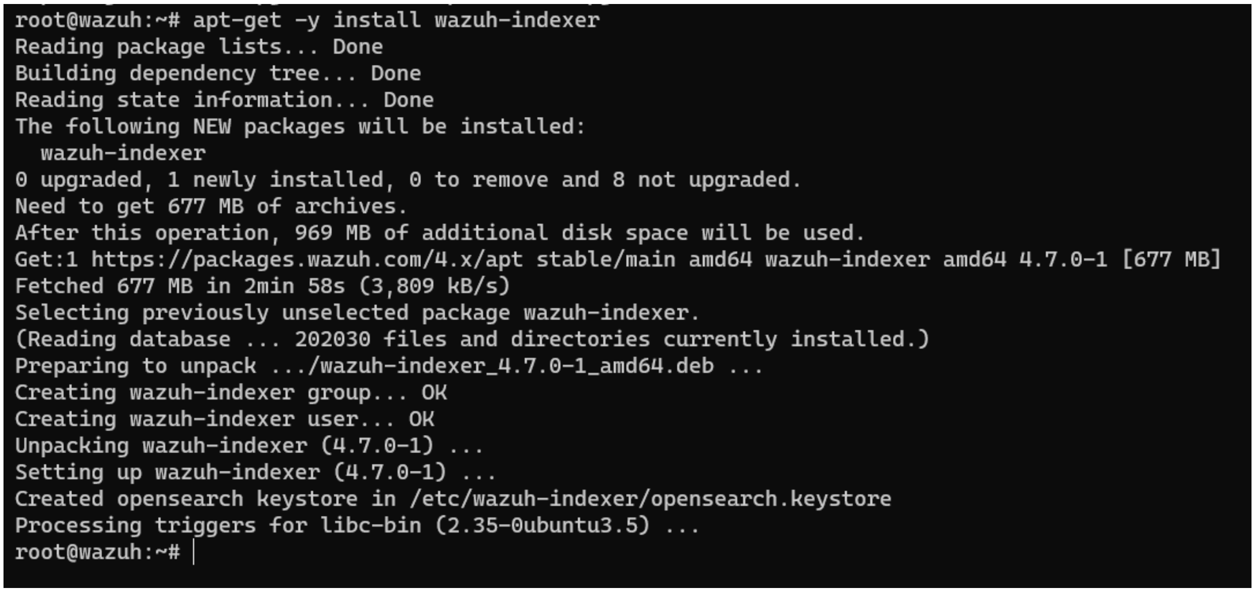
```
root@wazuh:~# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
root@wazuh:~#
```

## Now update your system using this command

```
apt update
```

## Now install wazuh indexer

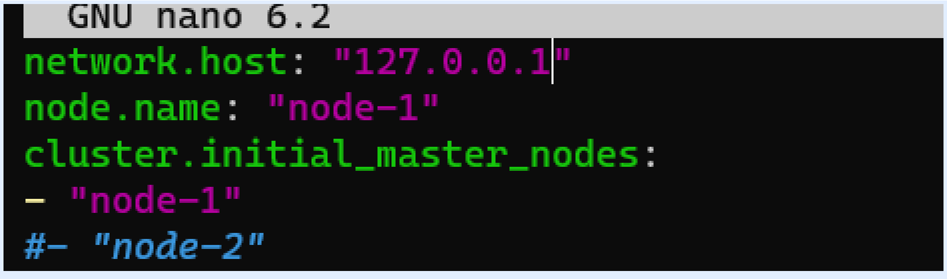
```
apt-get -y install wazuh-indexer
```



```
root@wazuh:~# apt-get -y install wazuh-indexer
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  wazuh-indexer
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 677 MB of archives.
After this operation, 969 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-indexer amd64 4.7.0-1 [677 MB]
Fetched 677 MB in 2min 58s (3,809 kB/s)
Selecting previously unselected package wazuh-indexer.
(Reading database ... 202030 files and directories currently installed.)
Preparing to unpack .../wazuh-indexer_4.7.0-1_amd64.deb ...
Creating wazuh-indexer group... OK
Creating wazuh-indexer user... OK
Unpacking wazuh-indexer (4.7.0-1) ...
Setting up wazuh-indexer (4.7.0-1) ...
Created opensearch keystore in /etc/wazuh-indexer/opensearch.keystore
Processing triggers for libc-bin (2.35-0ubuntu3.5) ...
root@wazuh:~# 
```

## Edit this file and set network host

```
nano /etc/wazuh-indexer/opensearch.yml
```



```
  GNU nano 6.2
network.host: "127.0.0.1"
node.name: "node-1"
cluster.initial_master_nodes:
- "node-1"
#- "node-2"
```

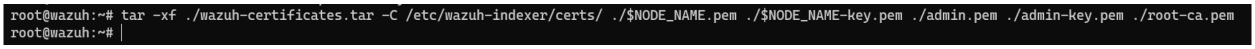
## set node name using this command
NODE_NAME=node-1

## create a directory

```
mkdir /etc/wazuh-indexer/certs
```

## Untar the file

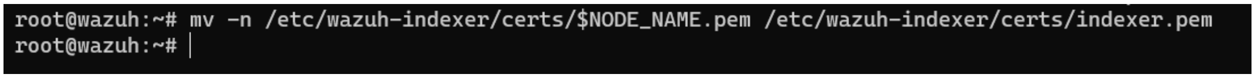
```
tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/
./$NODE_NAME.pem ./$NODE_NAME-key.pem ./admin.pem ./admin-key.pem
./root-ca.pem
```



```
root@wazuh:~# tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem
root@wazuh:~# 
```

## change the key name

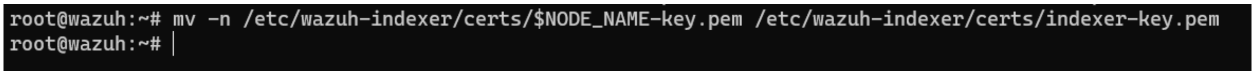
```
mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem
/etc/wazuh-indexer/certs/indexer.pem
```



```
root@wazuh:~# mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem /etc/wazuh-indexer/certs/indexer.pem
root@wazuh:~#
```

## change the key name

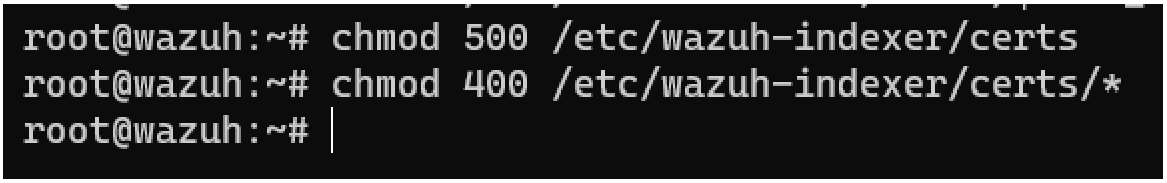
```
mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem
/etc/wazuh-indexer/certs/indexer-key.pem
```



```
root@wazuh:~# mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem /etc/wazuh-indexer/certs/indexer-key.pem
root@wazuh:~#
```

## change the permission

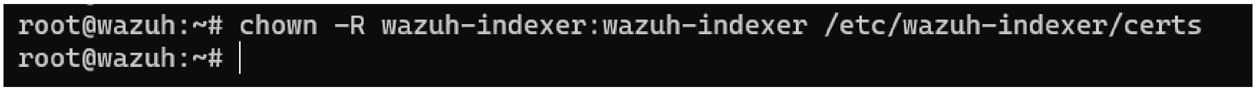
```
chmod 500 /etc/wazuh-indexer/certs

chmod 400 /etc/wazuh-indexer/certs/*
```



```
root@wazuh:~# chmod 500 /etc/wazuh-indexer/certs
root@wazuh:~# chmod 400 /etc/wazuh-indexer/certs/*
root@wazuh:~#
```

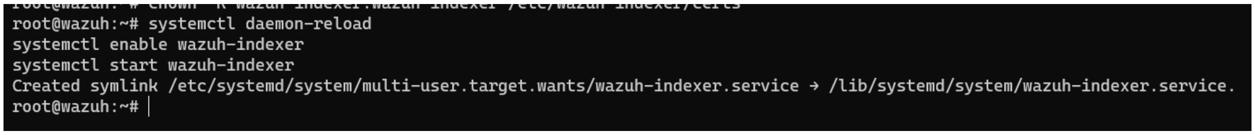
## change the ownership

```
chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```



```
root@wazuh:~# chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
root@wazuh:~#
```

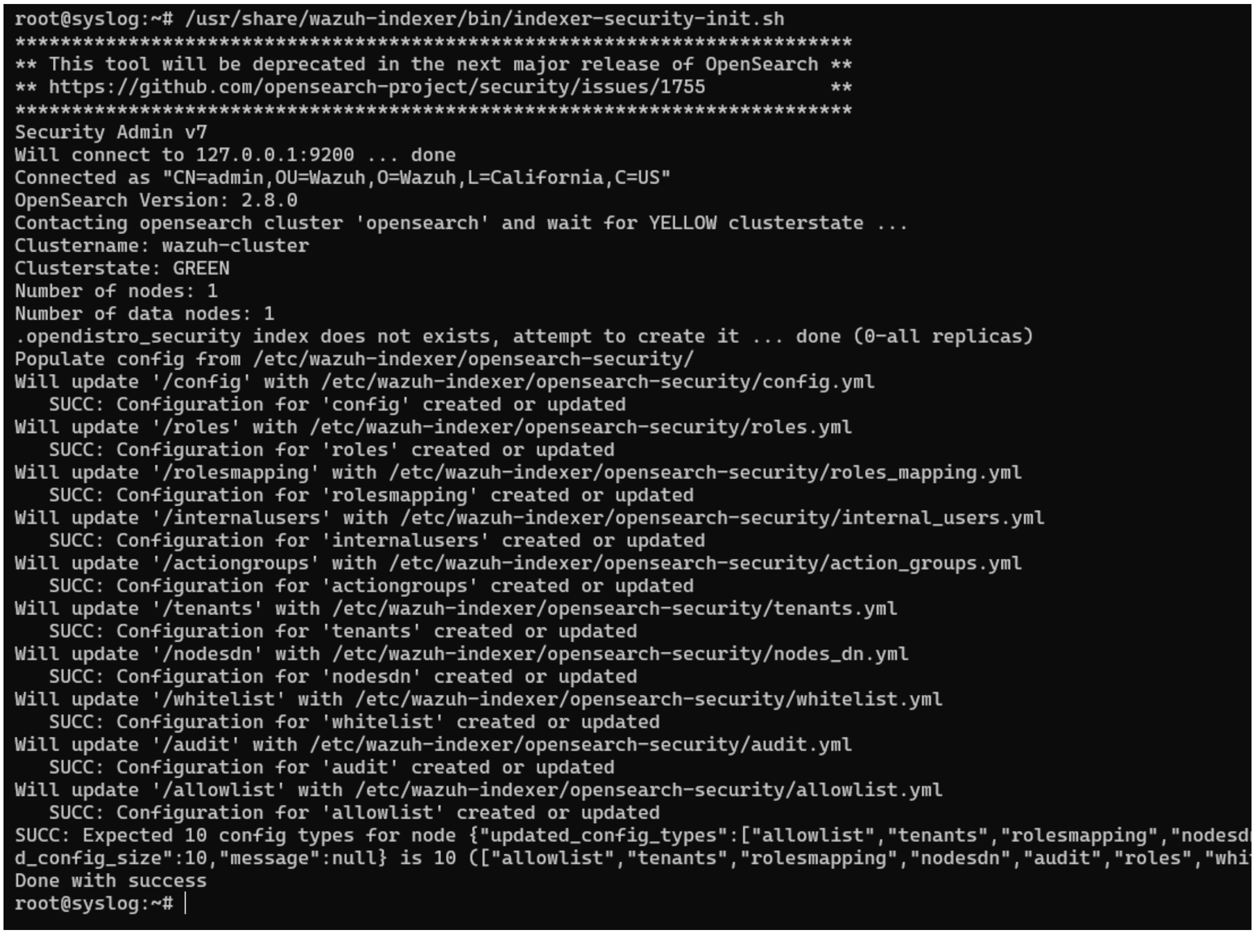
```
## reload, enable, & start the indexer service
systemctl daemon-reload
systemctl enable wazuh-indexer
systemctl start wazuh-indexer
```



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

## **Run the Wazuh indexer script to load the new certificates information and start the single-node or multi-node cluster.**

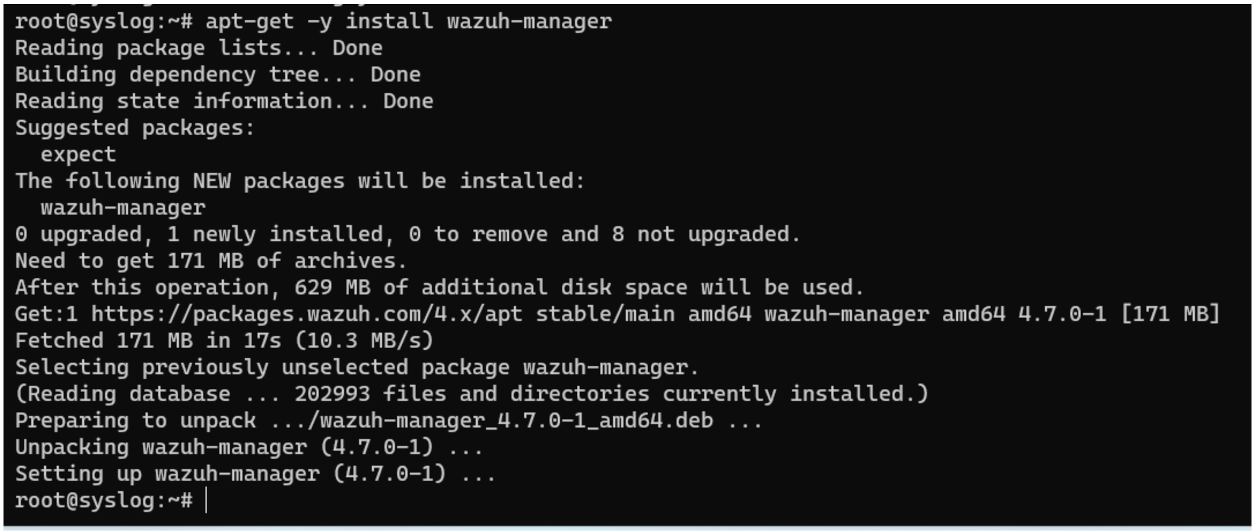
```
/usr/share/wazuh-indexer/bin/indexer-security-init.sh
```



```
root@syslog:~# /usr/share/wazuh-indexer/bin/indexer-security-init.sh
**************************************************************************
** This tool will be deprecated in the next major release of OpenSearch **
** https://github.com/opensearch-project/security/issues/1755           **
**************************************************************************
Security Admin v7
Will connect to 127.0.0.1:9200 ... done
Connected as "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"
OpenSearch Version: 2.8.0
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 /etc/wazuh-indexer/opensearch-security/
Will update '/config' with /etc/wazuh-indexer/opensearch-security/config.yml
   SUCC: Configuration for 'config' created or updated
Will update '/roles' with /etc/wazuh-indexer/opensearch-security/roles.yml
   SUCC: Configuration for 'roles' created or updated
Will update '/rolesmapping' with /etc/wazuh-indexer/opensearch-security/roles_mapping.yml
   SUCC: Configuration for 'rolesmapping' created or updated
Will update '/internalusers' with /etc/wazuh-indexer/opensearch-security/internal_users.yml
   SUCC: Configuration for 'internalusers' created or updated
Will update '/actiongroups' with /etc/wazuh-indexer/opensearch-security/action_groups.yml
   SUCC: Configuration for 'actiongroups' created or updated
Will update '/tenants' with /etc/wazuh-indexer/opensearch-security/tenants.yml
   SUCC: Configuration for 'tenants' created or updated
Will update '/nodesdn' with /etc/wazuh-indexer/opensearch-security/nodes_dn.yml
   SUCC: Configuration for 'nodesdn' created or updated
Will update '/whitelist' with /etc/wazuh-indexer/opensearch-security/whitelist.yml
   SUCC: Configuration for 'whitelist' created or updated
Will update '/audit' with /etc/wazuh-indexer/opensearch-security/audit.yml
   SUCC: Configuration for 'audit' created or updated
Will update '/allowlist' with /etc/wazuh-indexer/opensearch-security/allowlist.yml
   SUCC: Configuration for 'allowlist' created or updated
SUCC: Expected 10 config types for node {"updated_config_types":["allowlist","tenants","rolesmapping","nodesd
d_config_size":10,"message":null} is 10 (["allowlist","tenants","rolesmapping","nodesdn","audit","roles","whi
Done with success
root@syslog:~#
```

## Install Wazuh manager

apt-get -y install wazuh-manager

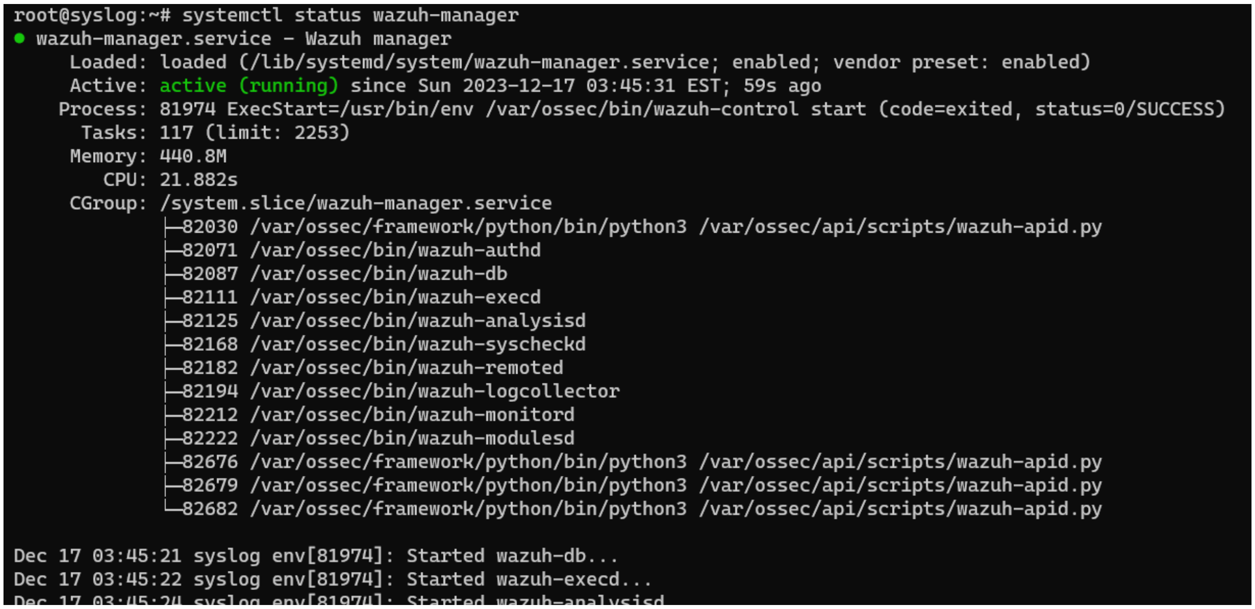


```
root@syslog:~# apt-get -y install wazuh-manager
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  expect
The following NEW packages will be installed:
  wazuh-manager
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 171 MB of archives.
After this operation, 629 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-manager amd64 4.7.0-1 [171 MB]
Fetched 171 MB in 17s (10.3 MB/s)
Selecting previously unselected package wazuh-manager.
(Reading database ... 202993 files and directories currently installed.)
Preparing to unpack .../wazuh-manager_4.7.0-1_amd64.deb ...
Unpacking wazuh-manager (4.7.0-1) ...
Setting up wazuh-manager (4.7.0-1) ...
root@syslog:~#
```

## Reload, enable and start the wazuh manager system

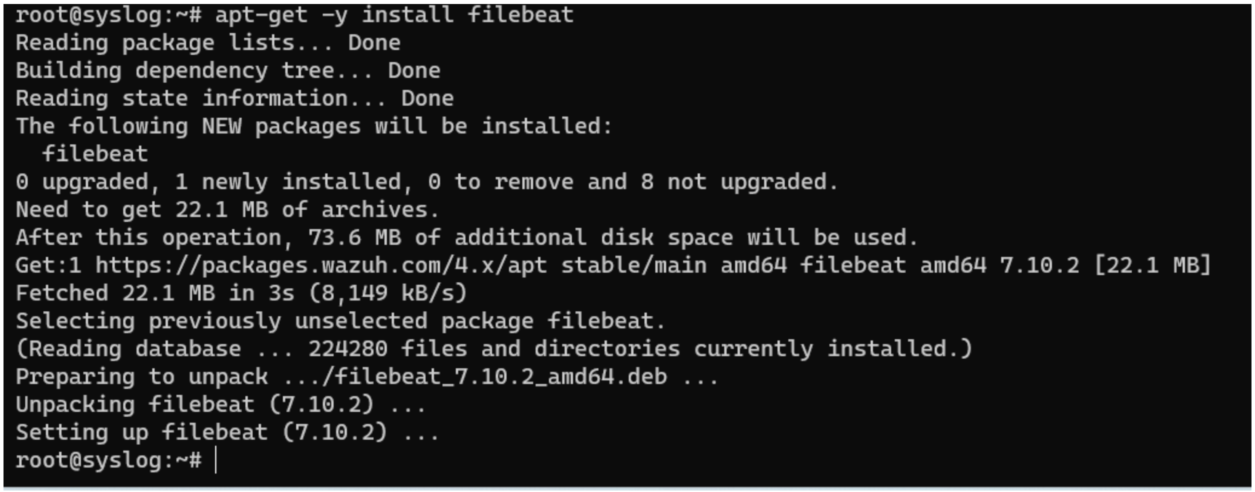
systemctl daemon-reload

```
systemctl enable wazuh-manager
systemctl start wazuh-manager
systemctl status wazuh-manager
```



```
root@syslog:~# 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 Sun 2023-12-17 03:45:31 EST; 59s ago
    Process: 81974 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 117 (limit: 2253)
     Memory: 440.8M
        CPU: 21.882s
     CGroup: /system.slice/wazuh-manager.service
             ├─82030 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─82071 /var/ossec/bin/wazuh-authd
             ├─82087 /var/ossec/bin/wazuh-db
             ├─82111 /var/ossec/bin/wazuh-execd
             ├─82125 /var/ossec/bin/wazuh-analysisd
             ├─82168 /var/ossec/bin/wazuh-syscheckd
             ├─82182 /var/ossec/bin/wazuh-remoted
             ├─82194 /var/ossec/bin/wazuh-logcollector
             ├─82212 /var/ossec/bin/wazuh-monitord
             ├─82222 /var/ossec/bin/wazuh-modulesd
             ├─82676 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─82679 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             └─82682 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py

Dec 17 03:45:21 syslog env[81974]: Started wazuh-db...
Dec 17 03:45:22 syslog env[81974]: Started wazuh-execd...
Dec 17 03:45:24 syslog env[81974]: Started wazuh-analysisd
```

```
## Install filebeat
apt-get -y install filebeat
```

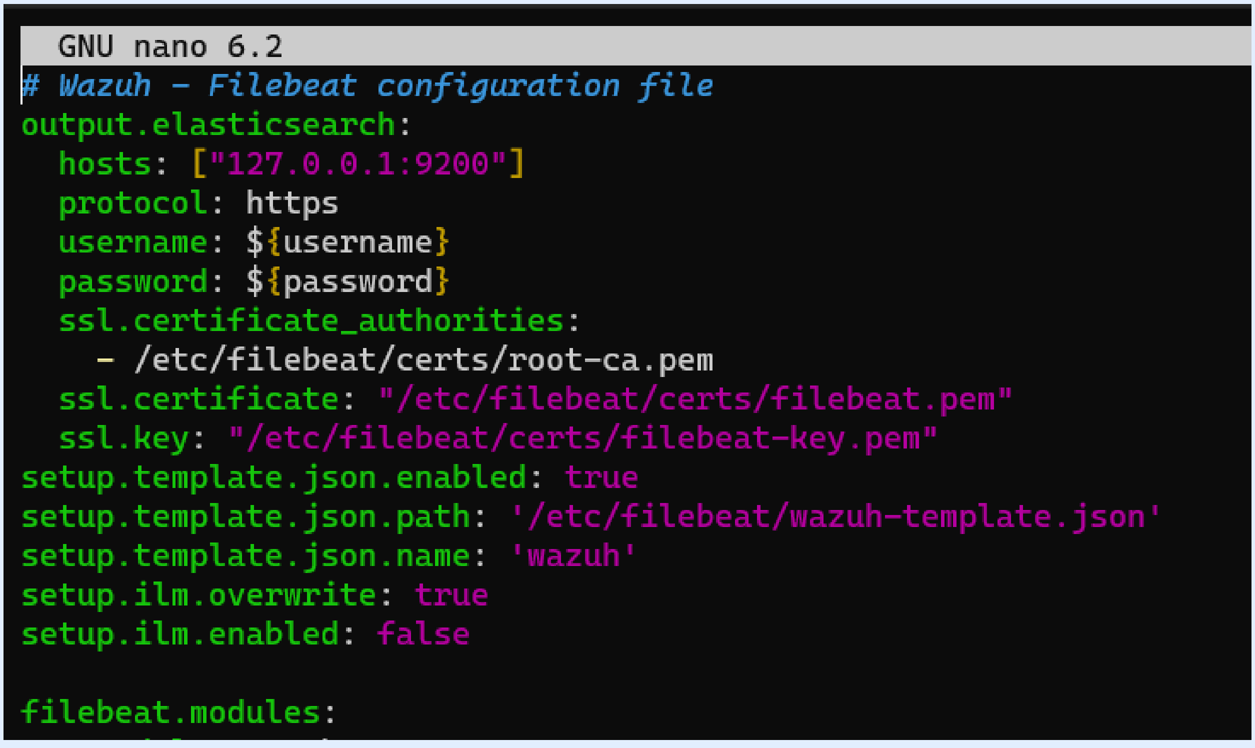


```
root@syslog:~# apt-get -y install filebeat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 22.1 MB of archives.
After this operation, 73.6 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 filebeat amd64 7.10.2 [22.1 MB]
Fetched 22.1 MB in 3s (8,149 kB/s)
Selecting previously unselected package filebeat.
(Reading database ... 224280 files and directories currently installed.)
Preparing to unpack .../filebeat_7.10.2_amd64.deb ...
Unpacking filebeat (7.10.2) ...
Setting up filebeat (7.10.2) ...
root@syslog:~# 
```

## Download the preconfigured Filebeat configuration file.
curl -so /etc/filebeat/filebeat.yml
https://packages.wazuh.com/4.7/tpl/wazuh/filebeat/filebeat.yml

## Edit ip and port in this file

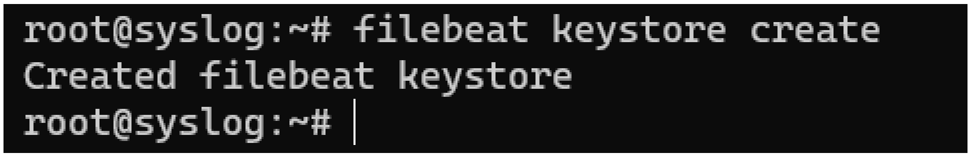
```
nano /etc/filebeat/filebeat.yml
```



```
  GNU nano 6.2
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["127.0.0.1:9200"]
  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

filebeat.modules:
```

## Create a Filebeat keystore to securely store authentication credentials.

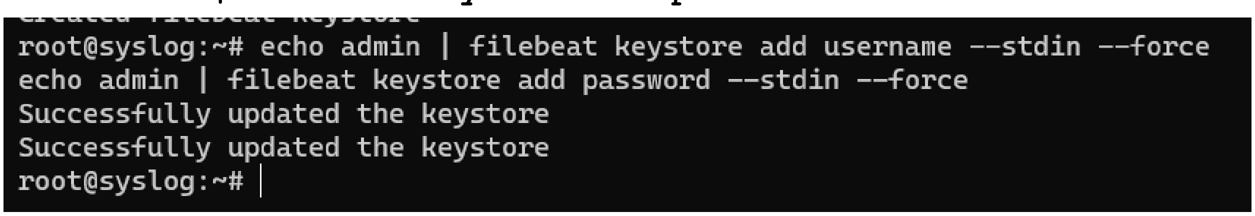
filebeat keystore create



```
root@syslog:~# filebeat keystore create
Created filebeat keystore
root@syslog:~#
```

## Add the default 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
```

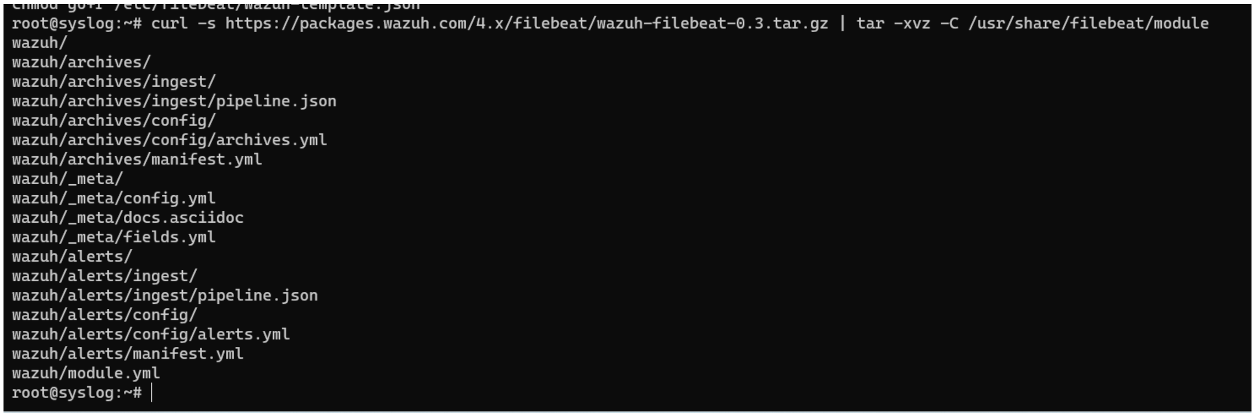


```
Created filebeat keystore
root@syslog:~# echo admin | filebeat keystore add username --stdin --force
echo admin | filebeat keystore add password --stdin --force
Successfully updated the keystore
Successfully updated the keystore
root@syslog:~#
```

## Download the alerts template for the Wazuh indexer.

curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/v4.7.0/extensions/elasticsearch/7.x/wazuh-template.json
chmod go+r /etc/filebeat/wazuh-template.json

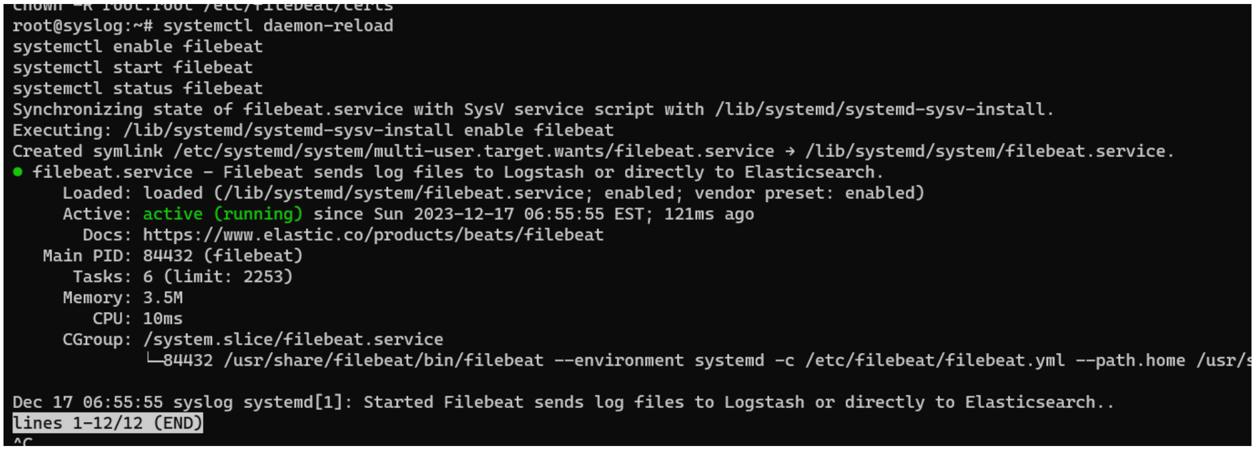
## Install the Wazuh module for Filebeat.
curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.3.tar.gz | tar -xvz -C /usr/share/filebeat/module



```
chmod go+r /etc/filebeat/wazuh-template.json
root@syslog:~# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.3.tar.gz | tar -xvz -C /usr/share/filebeat/module
wazuh/
wazuh/archives/
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/manifest.yml
wazuh/_meta/
wazuh/_meta/config.yml
wazuh/_meta/docs.asciidoc
wazuh/_meta/fields.yml
wazuh/alerts/
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/manifest.yml
wazuh/module.yml
root@syslog:~# |
```

## Replace `<server-node-name>` with your Wazuh server node certificate name, the same one used in `config.yml` when creating the certificates. Then, move the certificates to their corresponding location.

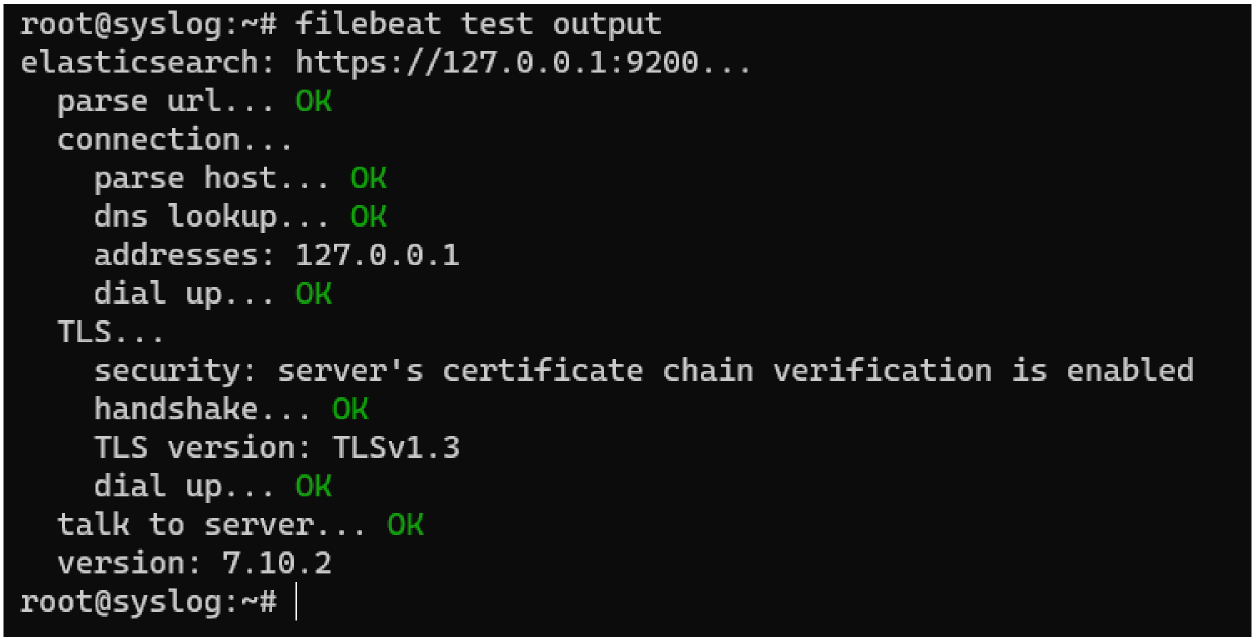
```
NODE_NAME=wazuh-1

mkdir /etc/filebeat/certs

tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ ./$NODE_NAME.pem

./$NODE_NAME-key.pem ./root-ca.pem

mv -n /etc/filebeat/certs/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem

mv -n /etc/filebeat/certs/$NODE_NAME-key.pem

/etc/filebeat/certs/filebeat-key.pem

chmod 500 /etc/filebeat/certs

chmod 400 /etc/filebeat/certs/*

chown -R root:root /etc/filebeat/certs


## reload, enable, start & check status of manager

systemctl daemon-reload

systemctl enable filebeat

systemctl start filebeat

systemctl status filebeat
```



```
chown -R root:root /etc/filebeat/certs
root@syslog:~# systemctl daemon-reload
systemctl enable filebeat
systemctl start filebeat
systemctl status 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.
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
     Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2023-12-17 06:55:55 EST; 121ms ago
       Docs: https://www.elastic.co/products/beats/filebeat
   Main PID: 84432 (filebeat)
      Tasks: 6 (limit: 2253)
     Memory: 3.5M
        CPU: 10ms
     CGroup: /system.slice/filebeat.service
             └─84432 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/s

Dec 17 06:55:55 syslog systemd[1]: Started Filebeat sends log files to Logstash or directly to Elasticsearch..
lines 1-12/12 (END)
^C
```

## Run the following command to verify that Filebeat is successfully installed.

`filebeat test output`



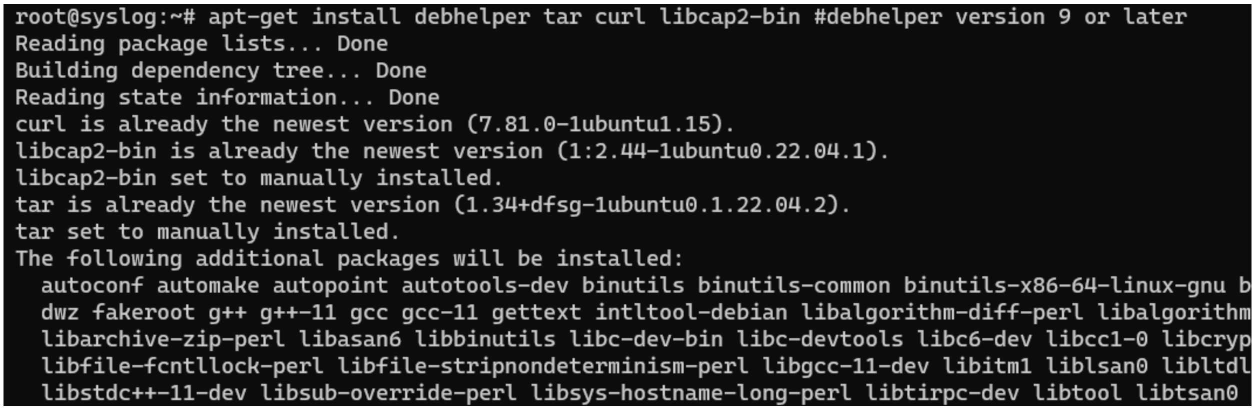
```
root@syslog:~# filebeat test output
elasticsearch: https://127.0.0.1:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 127.0.0.1
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.10.2
root@syslog:~#
```

⇒ `Dashboard Installation`

## Install the following packages for dashboard installation

apt-get install debhelper tar curl libcap2-bin #debhelper version 9 or later

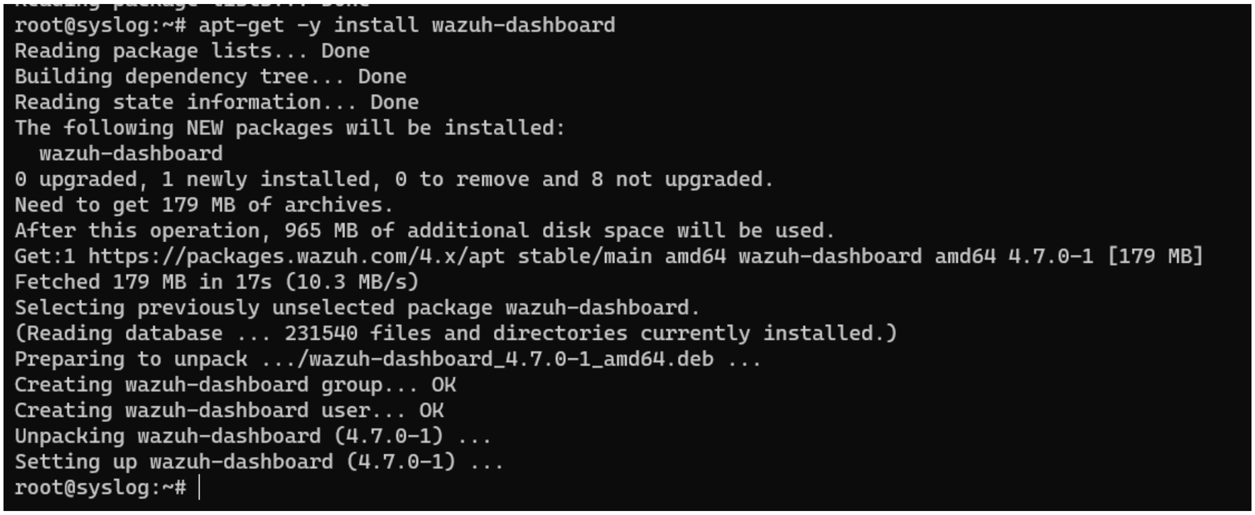
apt-get install gnupg apt-transport-https



```
root@syslog:~# apt-get install debhelper tar curl libcap2-bin #debhelper version 9 or later
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-1ubuntu1.15).
libcap2-bin is already the newest version (1:2.44-1ubuntu0.22.04.1).
libcap2-bin set to manually installed.
tar is already the newest version (1.34+dfsg-1ubuntu0.1.22.04.2).
tar set to manually installed.
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev binutils binutils-common binutils-x86-64-linux-gnu b
  dwz fakeroot g++ g++-11 gcc gcc-11 gettext intltool-debian libalgorithm-diff-perl libalgorithm
  libarchive-zip-perl libasan6 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcryp
  libfile-fcntllock-perl libfile-stripnondeterminism-perl libgcc-11-dev libitm1 liblsan0 libltdl
  libstdc++-11-dev libsub-override-perl libsys-hostname-long-perl libtirpc-dev libtool libtsan0
```

## Download Dashboard

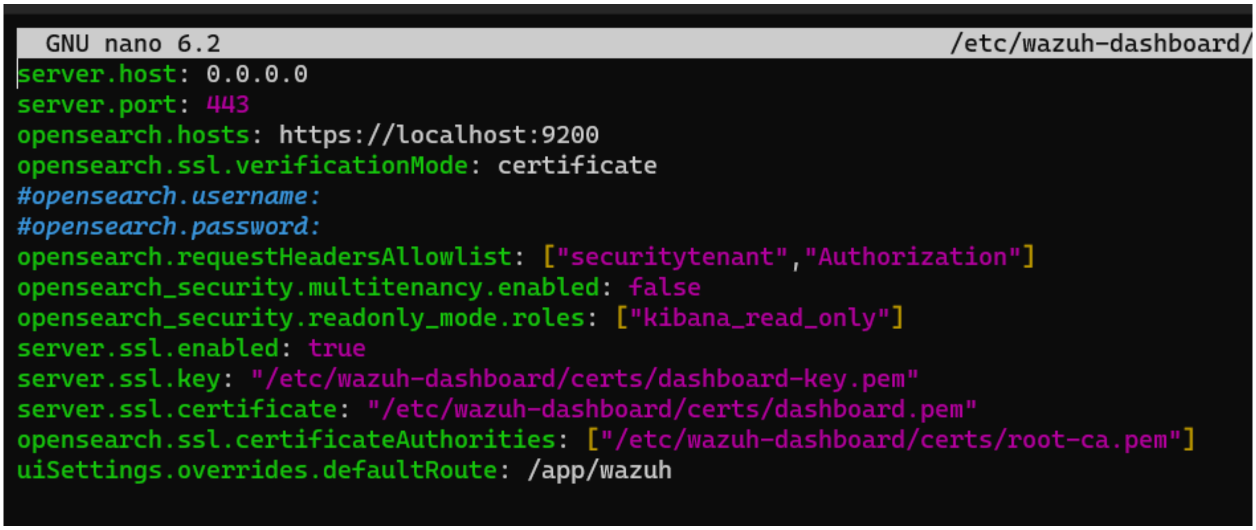
apt-get -y install wazuh-dashboard



```
root@syslog:~# apt-get -y install wazuh-dashboard
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  wazuh-dashboard
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 179 MB of archives.
After this operation, 965 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-dashboard amd64 4.7.0-1 [179 MB]
Fetched 179 MB in 17s (10.3 MB/s)
Selecting previously unselected package wazuh-dashboard.
(Reading database ... 231540 files and directories currently installed.)
Preparing to unpack .../wazuh-dashboard_4.7.0-1_amd64.deb ...
Creating wazuh-dashboard group... OK
Creating wazuh-dashboard user... OK
Unpacking wazuh-dashboard (4.7.0-1) ...
Setting up wazuh-dashboard (4.7.0-1) ...
root@syslog:~#
```

## Edit this file and configure like this

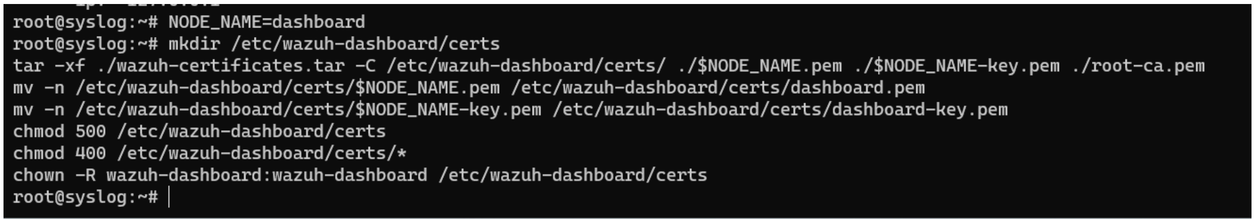
nano /etc/wazuh-dashboard/opensearch_dashboards.yml



```
  GNU nano 6.2                                                    /etc/wazuh-dashboard/
server.host: 0.0.0.0
server.port: 443
opensearch.hosts: https://localhost:9200
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersAllowlist: ["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
```

## Replace `<dashboard-node-name>` with your Wazuh dashboard node name, the same one used in `config.yml` to create the certificates, and move the certificates to their corresponding location.

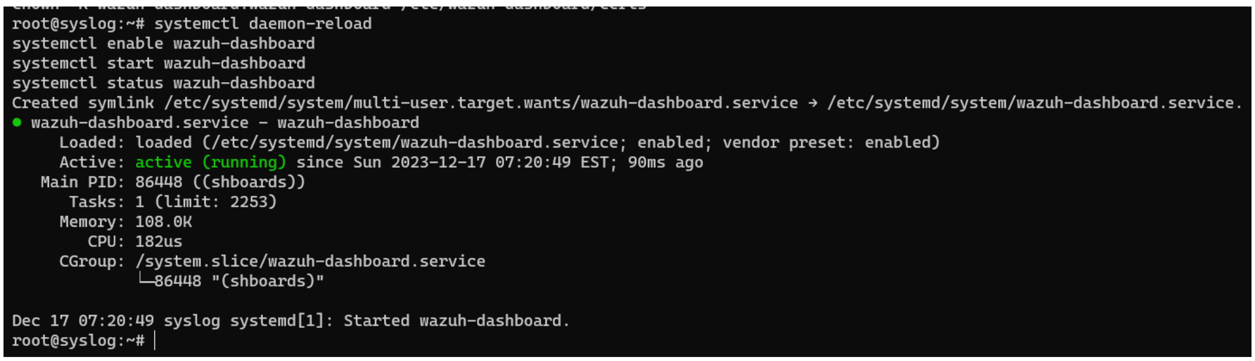
NODE_NAME=dashboard
mkdir /etc/wazuh-dashboard/certs
tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem
mv -n /etc/wazuh-dashboard/certs/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem
mv -n /etc/wazuh-dashboard/certs/$NODE_NAME-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem
chmod 500 /etc/wazuh-dashboard/certs
chmod 400 /etc/wazuh-dashboard/certs/*
chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs



```
root@syslog:~# NODE_NAME=dashboard
root@syslog:~# mkdir /etc/wazuh-dashboard/certs
tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem
mv -n /etc/wazuh-dashboard/certs/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem
mv -n /etc/wazuh-dashboard/certs/$NODE_NAME-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem
chmod 500 /etc/wazuh-dashboard/certs
chmod 400 /etc/wazuh-dashboard/certs/*
chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
root@syslog:~# 
```

## `reload, enable, start & check status of Dshboard`

systemctl daemon-reload
systemctl enable wazuh-dashboard
systemctl start wazuh-dashboard
systemctl status wazuh-dashboard



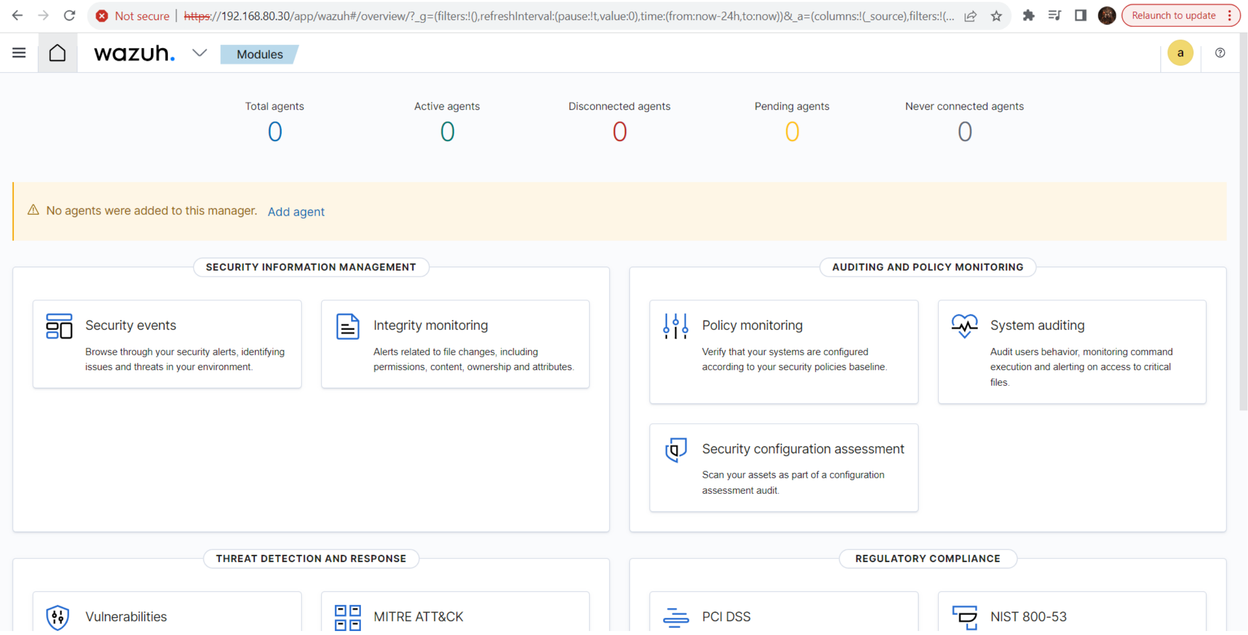
```
chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
root@syslog:~# systemctl daemon-reload
systemctl enable wazuh-dashboard
systemctl start wazuh-dashboard
systemctl status wazuh-dashboard
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-dashboard.service → /etc/systemd/system/wazuh-dashboard.service.
● wazuh-dashboard.service - wazuh-dashboard
     Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2023-12-17 07:20:49 EST; 90ms ago
   Main PID: 86448 ((shboards))
      Tasks: 1 (limit: 2253)
     Memory: 108.0K
        CPU: 182us
     CGroup: /system.slice/wazuh-dashboard.service
             └─86448 "(shboards)"

Dec 17 07:20:49 syslog systemd[1]: Started wazuh-dashboard.
root@syslog:~# 
```

## for login on browser type server-ip with https

https://192.168.80.30

Default id →admin

Password→admin



Not secure | ~~https~~://192.168.80.30/app/wazuh#/overview/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-24h,to:now))&_a=(columns:!(_source),filters:!(...

Total agents	Active agents	Disconnected agents	Pending agents	Never connected agents
0	0	0	0	0

⚠ No agents were added to this manager. Add agent

SECURITY INFORMATION MANAGEMENT

Security events
Browse through your security alerts, identifying issues and threats in your environment.

Integrity monitoring
Alerts related to file changes, including permissions, content, ownership and attributes.

AUDITING AND POLICY MONITORING

Policy monitoring
Verify that your systems are configured according to your security policies baseline.

System auditing
Audit users behavior, monitoring command execution and alerting on access to critical files.

Security configuration assessment
Scan your assets as part of a configuration assessment audit.

THREAT DETECTION AND RESPONSE

Vulnerabilities

MITRE ATT&CK

REGULATORY COMPLIANCE

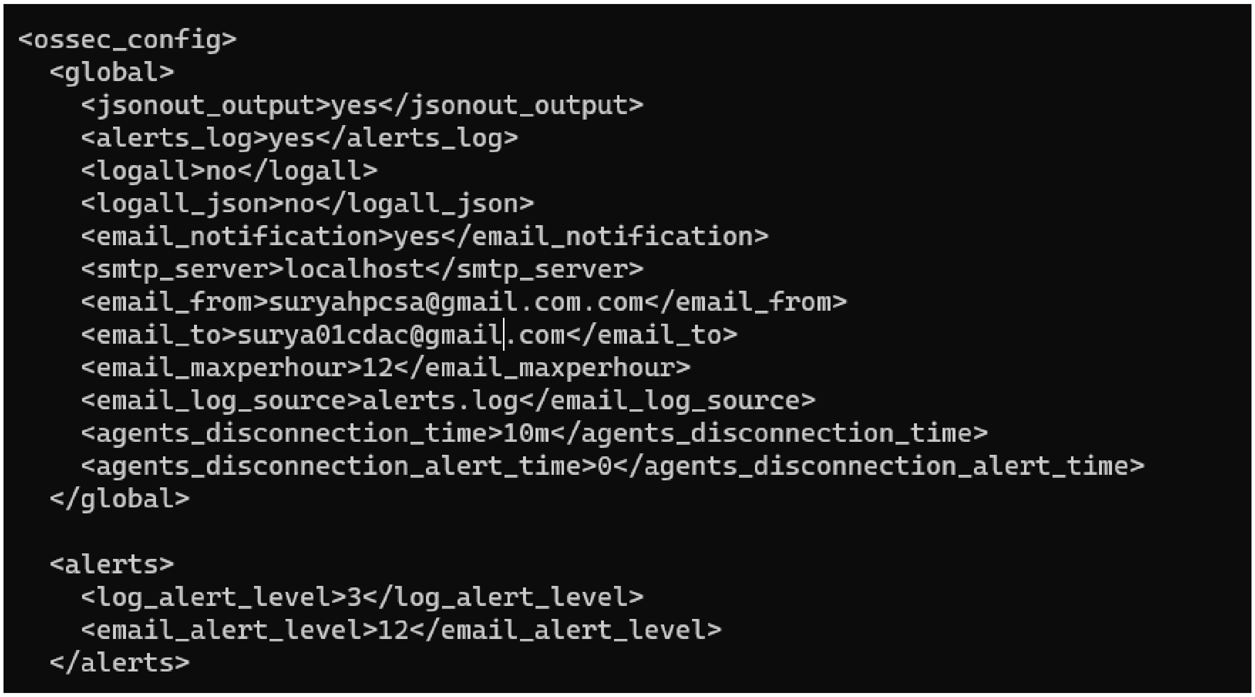
PCI DSS

NIST 800-53

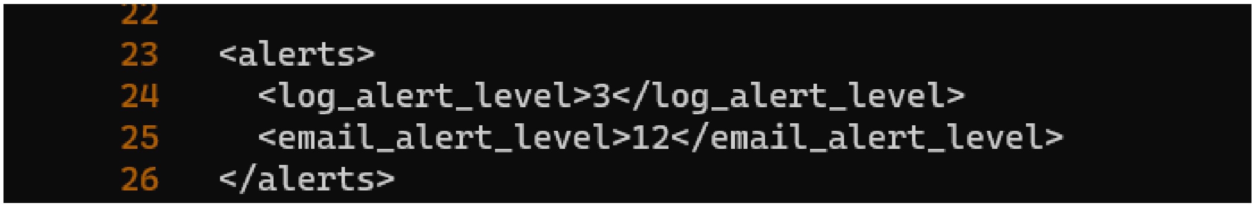
## Configuration file

vi /var/ossec/etc/ossec.conf

## we can set alert level



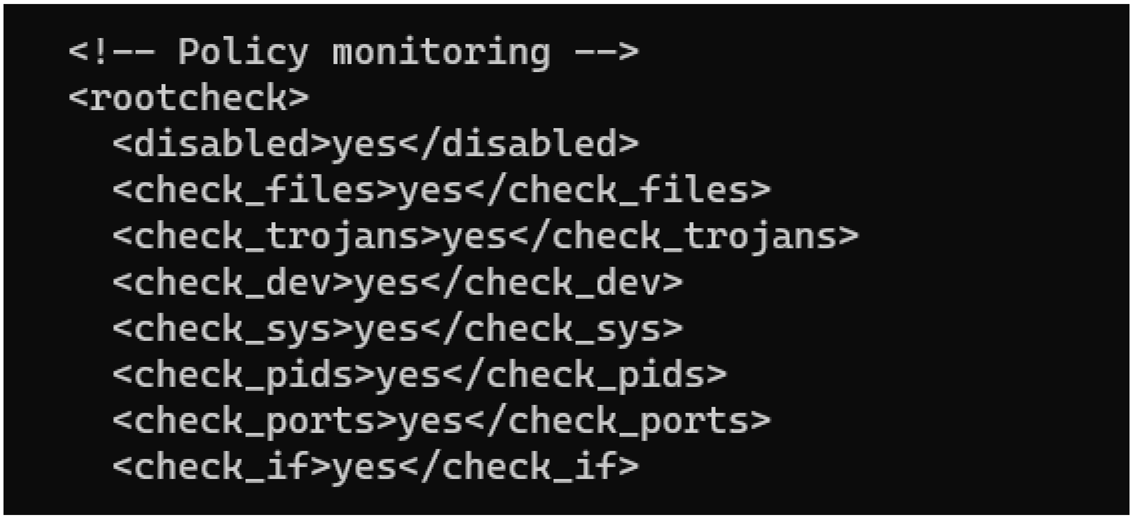
```
<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>yes</email_notification>
    <smtp_server>localhost</smtp_server>
    <email_from>suryahpcsa@gmail.com.com</email_from>
    <email_to>surya01cdac@gmail.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
  </global>

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



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

It means whenever get attack on our system then send email and take action automatically

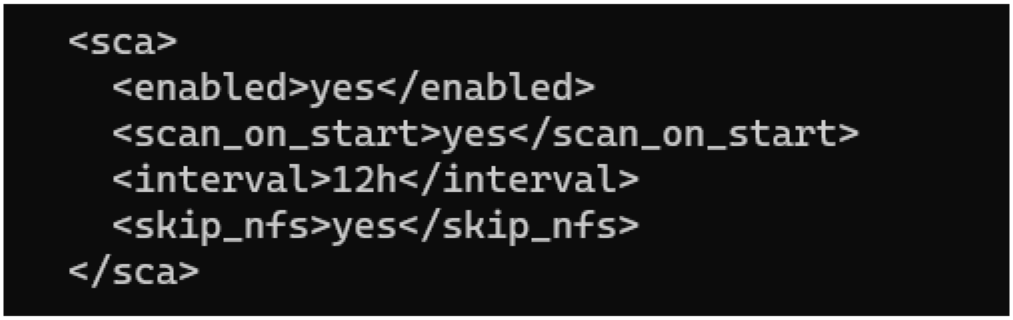
## we can set root action here



```
  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>yes</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>
```

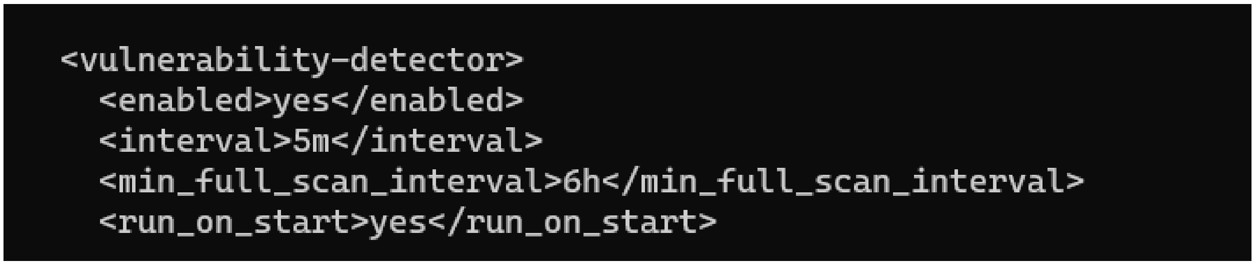
By set ‘no’ to ‘yes’

## for monitor ‘SCA’ enable here



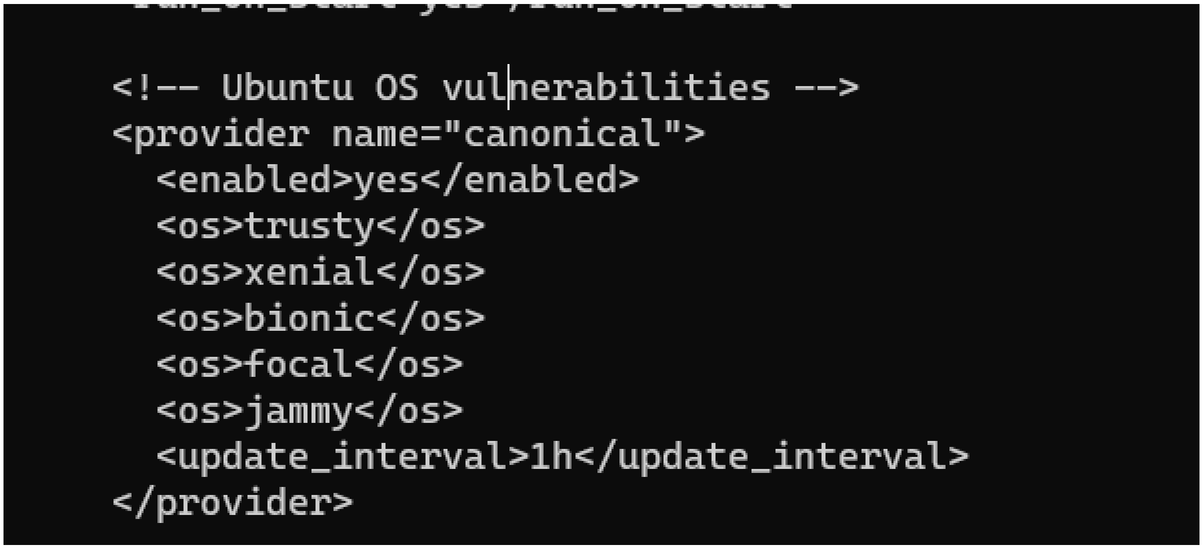
```
<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>
```

## For monitor vulnerability enable here



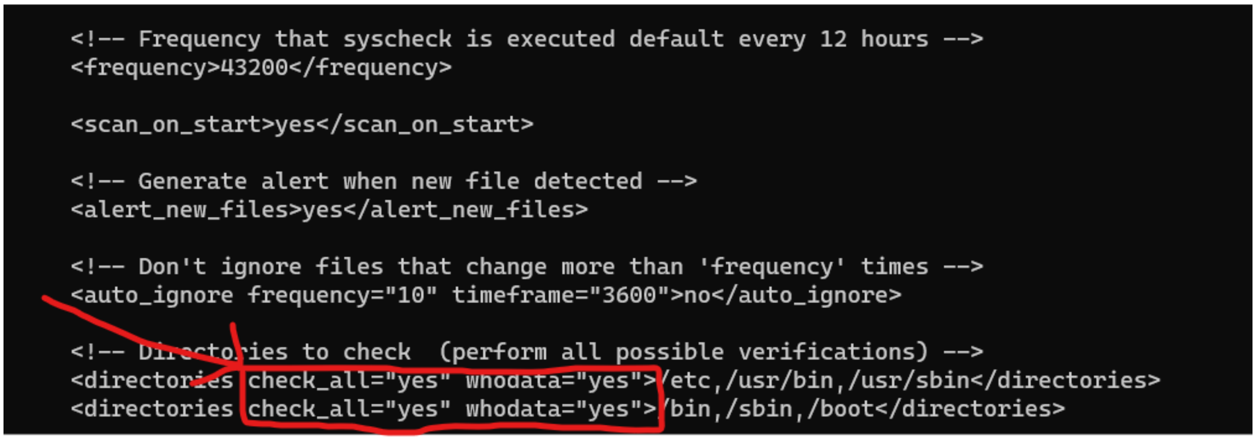
```
<vulnerability-detector>
  <enabled>yes</enabled>
  <interval>5m</interval>
  <min_full_scan_interval>6h</min_full_scan_interval>
  <run_on_start>yes</run_on_start>
```

## For monitor ‘OS’ here enable here



```
  <run_on_start>yes</run_on_start>

  <!-- Ubuntu OS vulnerabilities -->
  <provider name="canonical">
    <enabled>yes</enabled>
    <os>trusty</os>
    <os>xenial</os>
    <os>bionic</os>
    <os>focal</os>
    <os>jammy</os>
    <update_interval>1h</update_interval>
  </provider>
```

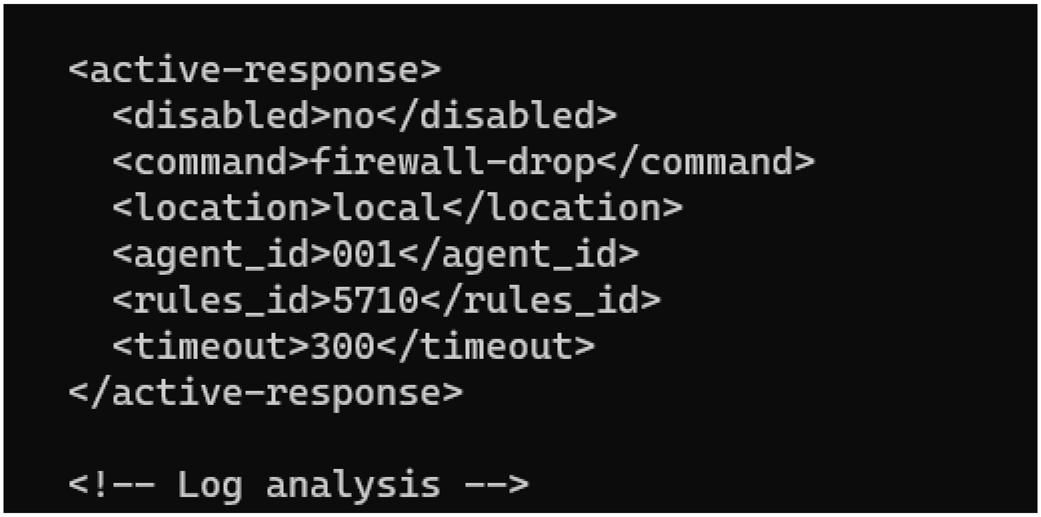
## for monitor directories configure here



```
<!-- Frequency that syscheck is executed default every 12 hours -->
<frequency>43200</frequency>

<scan_on_start>yes</scan_on_start>

<!-- Generate alert when new file detected -->
<alert_new_files>yes</alert_new_files>

<!-- Don't ignore files that change more than 'frequency' times -->
<auto_ignore frequency="10" timeframe="3600">no</auto_ignore>

<!-- Directories to check  (perform all possible verifications) -->
<directories check_all="yes" whodata="yes">/etc,/usr/bin,/usr/sbin</directories>
<directories check_all="yes" whodata="yes">/bin,/sbin,/boot</directories>
```

Add this sentence like above image
check_all="yes" whodata="yes"
check_all="yes" whodata="yes"

## Configure active response like this



```
<active-response>
  <disabled>no</disabled>
  <command>firewall-drop</command>
  <location>local</location>
  <agent_id>001</agent_id>
  <rules_id>5710</rules_id>
  <timeout>300</timeout>
</active-response>

<!-- Log analysis -->
```

Agent id-000 is server
Server monitor itself first