WAZUH
DOCUMENTATION V 4.3

Table of content

What is Wazuh

Wazuh is a free, open source and enterprise-ready security monitoring solution for threat detection, integrity monitoring, incident response and compliance. Which provides protection for endpoints and cloud workloads. The solution is composed of a single universal agent and three central components: the Wazuh server, the Wazuh indexer, and the Wazuh dashboard.

Components

The Wazuh platform provides XDR and SIEM features to protect your cloud, container, and server workloads. These include log data analysis, intrusion and malware detection, file integrity monitoring, configuration assessment, vulnerability detection, and support for regulatory compliance.

The Wazuh solution is based on the Wazuh agent, which is deployed on the monitored endpoints, and on three central components: the Wazuh server, the Wazuh indexer, and the Wazuh dashboard.

- The **Wazuh indexer** is a highly scalable, full-text search and analytics engine. This central component indexes and stores alerts generated by the Wazuh server

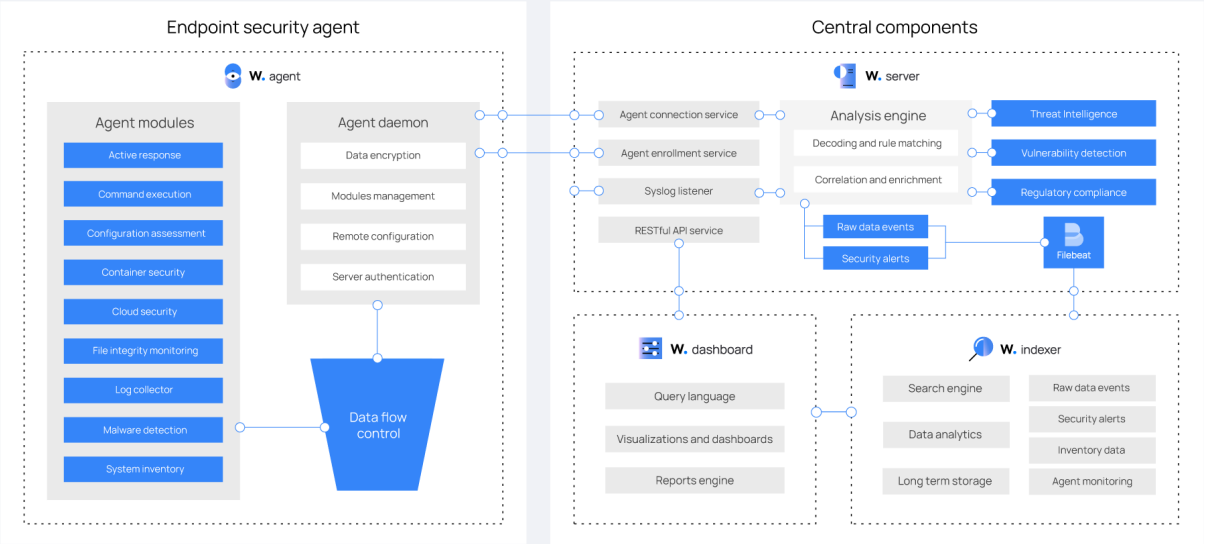
- The **Wazuh server** analyzes data received from the agents. It processes it through decoders and rules, using threat intelligence to look for well-known indicators of compromise (IOCs). A single server can analyze data from hundreds or thousands of agents, and scale horizontally when set up as a cluster. This central component is also used to manage the agents, configuring and upgrading them remotely when necessary

- The **Wazuh dashboard** is the web user interface for data visualization and analysis. It includes out-of-the-box dashboards for security events, regulatory compliance (e.g., PCI DSS, GDPR, CIS, HIPAA, NIST 800-53), detected vulnerable applications, file integrity monitoring data, configuration assessment results, cloud infrastructure monitoring events, and others. It is also used to manage Wazuh configuration and to monitor its status

- **Wazuh agents** are installed on endpoints such as laptops, desktops, servers, cloud instances, or virtual machines. They provide threat

prevention, detection, and response capabilities. They run on operating systems such as Linux, Windows, macOS, Solaris, AIX, and HP-UX

In addition to agent-based monitoring capabilities, the Wazuh platform can monitor agent-less devices such as **firewalls, switches, routers, or network IDS**, among others. For example, a system log data can be collected via Syslog, and its configuration can be monitored through periodic probing of its data, via SSH or through an API.



Installation Guide

Requirements

Make sure that your system environment meets all requirements and that you have root user privilege.

Recommended Operating System

Wazuh can be installed on a 64-bit Linux operating system. Wazuh supports the following operating system versions

- Amazon Linux 2
- CentOS 7, 8
- Red Hat Enterprise Linux 7, 8, 9
- Ubuntu 16.04, 18.04, 20.04, 22.04

Hardware recommendations

Component	RAM (min)	CPU (min)	RAM (max)	CPU (max)
Wazuh indexer	4	2	16	8

However we have to install **Three** things for **Wazuh platform**

1. **Wazuh Indexer**
2. **Wazuh Server**
3. **Wazuh Dashboard**

Installation of Wazuh Indexer Step by Step

Wazuh indexer is Highly scalable full-text search engine

It helps in-

- Altering
- Index management
- Deep performance analysis and more.

Installation Process is divided into Three Stages

1. Certificate Creation
2. Node Installation
3. Cluster Initialization

1. Certificate Creation

Generating SSL Certificates

STEP 1 - Download the `wazuh-certs-tool.sh` script and the `config.yml` configuration file. This creates the certificates that encrypt communications between the Wazuh central components.

```
# curl -sO https://packages.wazuh.com/4.3/wazuh-certs-tool.sh
# curl -sO https://packages.wazuh.com/4.3/config.yml
```

```
ubuntu@ubuntu2004:~/ajay$ curl -sO https://packages.wazuh.com/4.3/wazuh-certs-to
ol.sh
ubuntu@ubuntu2004:~/ajay$ curl -sO https://packages.wazuh.com/4.3/config.yml
ubuntu@ubuntu2004:~/ajay$ ls
config.yml  wazuh-certs-tool.sh
```

STEP 2- Edit the `config.yml` file with Text editor. Change the node name and Assign IP for the Node where you want to host Wazuh Manager. In My case im Assigning localhost IP.

STEP 3- Do the same thing for SERVER and DASHBOARD change name and Assign IP. In My case I left NAMES Default.

```
1 nodes:
2   # Wazuh indexer nodes
3   indexer:
4     - name: node-1
5       ip: 192.168.20.15
6     #- name: node-2
7     #  ip: <indexer-node-ip>
8     #- name: node-3
9     #  ip: <indexer-node-ip>
10
11  # Wazuh server nodes
12  # If there is more than one Wazuh server
13  # node, each one must have a node_type
14  server:
15    - name: wazuh-1
16      ip: 192.168.20.15
17    #  node_type: master
18    #- name: wazuh-2
19    #  ip: <wazuh-manager-ip>
20    #  node_type: worker
21    #- name: wazuh-3
22    #  ip: <wazuh-manager-ip>
23    #  node_type: worker
24
25  # Wazuh dashboard nodes
26  dashboard:
27    - name: dashboard
28      ip: 192.168.20.15
```

STEP 4 - Run `./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.

```
ubuntu@ubuntu2004:~/ajay$ bash ./wazuh-certs-tool.sh -A
27/01/2023 01:42:18 INFO: Admin certificates created.
l27/01/2023 01:42:18 INFO: Wazuh indexer certificates created.
27/01/2023 01:42:18 INFO: Wazuh server certificates created.
s27/01/2023 01:42:18 INFO: Wazuh dashboard certificates created.
ubuntu@ubuntu2004:~/ajay$ ls
config.yml  wazuh-certificates  wazuh-certs-tool.sh
```

STEP 5 - Compress all the necessary file. Make .tar File of Generated Certs and you have to Copy this .tar file to all nodes including INDEX, SERVER, DASHBOARD. This can be done using SCP utility.

```
ubuntu@ubuntu2004:~/ajay$ tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates/ .
./
./node-1-key.pem
./root-ca.pem
./admin-key.pem
./root-ca.key
./wazuh-1-key.pem
./admin.pem
./dashboard.pem
./node-1.pem
./wazuh-1.pem
./dashboard-key.pem
ubuntu@ubuntu2004:~/ajay$ rm -rf ./wazuh-certificates
ubuntu@ubuntu2004:~/ajay$ ls
config.yml  wazuh-certificates.tar  wazuh-certs-tool.sh
```

2. Node Installation

Installing package Dependencies

STEP 1 - Install the following packages if missing

```
# apt-get install debconf adduser procps
```

```
root@ubuntu2004:/home/ubuntu/ajay# apt-get install debconf adduser procps
Reading package lists... Done
Building dependency tree
Reading state information... Done
adduser is already the newest version (3.118ubuntu2).
adduser set to manually installed.
debconf is already the newest version (1.5.73).
debconf set to manually installed.
procps is already the newest version (2:3.3.16-1ubuntu2.3).
procps set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

Adding the Wazuh repository

STEP 2 - Install the following packages if missing.

```
# apt-get install gnupg apt-transport-https
```

```
root@ubuntu2004:/home/ubuntu/ajay# apt-get install gnupg apt-transport-https
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.2).
apt-transport-https is already the newest version (2.0.9).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

STEP 3 - 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@ubuntu2004:/home/ubuntu/ajay# 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
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>" impo
rted
gpg: Total number processed: 1
gpg:               imported: 1
```

STEP 4 - Adding repository & Update the Packages

```
# 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
```

```
# apt-get update
```

```
root@ubuntu2004:/home/ubuntu/ajay# 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
```

Installing the Wazuh Indexer

STEP 1 - Install the Wazuh indexer package

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

```
root@ubuntu2004:/home/ubuntu/ajay# apt-get -y install wazuh-indexer
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  wazuh-indexer
0 upgraded, 1 newly installed, 0 to remove and 20 not upgraded.
Need to get 357 MB of archives.
After this operation, 639 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-indexer amd64 4.3.10-1 [357 M
B]
Fetched 357 MB in 42s (8,434 kB/s)
Selecting previously unselected package wazuh-indexer.
(Reading database ... 204299 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 systemd (245.4-4ubuntu3.19) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
```

Configuring the Wazuh indexer

STEP 1 - Edit the `/etc/wazuh-indexer/opensearch.yml`
STEP 2 - Set the `Network.host` IP same as `config.yml`
STEP 3 - change the **`node.name`** same as it is set in **`Config.yml`**
STEP 4 - `cluster.initial_master_nodes` same as it is in `config.yml`
`(as im installing only one node so no need to configure)`
STEP 5 - **`discovery.seed_hosts`** same as it is set in **`config.yml`**
`(as im installing only one node so no need to configure)`

```
network.host: "192.168.20.15"
node.name: "node-1"
cluster.initial_master_nodes:
- "node-1"
#- "wazuh-1"
#- "dashboard"
cluster.name: "wazuh-cluster"
discovery.seed_hosts:
 # - "192.168.20.15"
 #- "192.168.20.15"
 #- "192.168.20.15"
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=wazuh-1,OU=Wazuh,O=Wazuh,L=California,C=US"
- "CN=dashboard-3,OU=Wazuh,O=Wazuh,L=California,C=US"
plugins.security.restapi.roles_enabled:
- "all_access"
- "security_rest_api_access"
```

3. Deploying Certificates

STEP 1 - Run the following command replace the `<indexer-node-name>`

```
# NODE_NAME=<indexer-node-name>

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

```
# mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem
/etc/wazuh-indexer/certs/indexer.pem
# mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem
/etc/wazuh-indexer/certs/indexer-key.pem
# chmod 500 /etc/wazuh-indexer/certs
# chmod 400 /etc/wazuh-indexer/certs/*
# chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
```

```
root@ubuntu2004:/home/ubuntu/ajay# NODE_NAME=node-1
root@ubuntu2004:/home/ubuntu/ajay# tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/cert
s/ ./node-1.pem ./node-1-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem
root@ubuntu2004:/home/ubuntu/ajay# mv -n /etc/wazuh-indexer/certs/node-1.pem /etc/wazuh-indexe
r/certs/indexer.pem
root@ubuntu2004:/home/ubuntu/ajay# mv -n /etc/wazuh-indexer/certs/node-1-key.pem /etc/wazuh-in
dexer/certs/indexer-key.pem
root@ubuntu2004:/home/ubuntu/ajay# chmod 500 /etc/wazuh-indexer/certs
root@ubuntu2004:/home/ubuntu/ajay#
root@ubuntu2004:/home/ubuntu/ajay# chmod 400 /etc/wazuh-indexer/certs/*
root@ubuntu2004:/home/ubuntu/ajay#
root@ubuntu2004:/home/ubuntu/ajay# chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/cer
ts
root@ubuntu2004:/home/ubuntu/ajay# ls
config.yml  wazuh-certificates.tar  wazuh-certs-tool.sh
root@ubuntu2004:/home/ubuntu/ajay# cd /etc/wazuh-indexer/
root@ubuntu2004:/etc/wazuh-indexer# ls
certs        jvm.options.d      opensearch.keystore       opensearch-reports-scheduler
jvm.options  log4j2.properties  opensearch-observability  opensearch.yml
root@ubuntu2004:/etc/wazuh-indexer# cd certs/
root@ubuntu2004:/etc/wazuh-indexer/certs# ls
admin-key.pem  admin.pem  indexer-key.pem  indexer.pem  root-ca.pem
```

Starting the Service

> Enable and start the Wazuh indexer service.

```
# systemctl daemon-reload
# systemctl enable wazuh-indexer
# systemctl start wazuh-indexer
```

```
root@ubuntu2004:/etc/wazuh-indexer/certs# systemctl status wazuh-indexer
● wazuh-indexer.service - Wazuh-indexer
     Loaded: loaded (/lib/systemd/system/wazuh-indexer.service; enabled; vendor preset: enabl>
     Active: active (running) since Fri 2023-01-27 04:26:05 EST; 9s ago
       Docs: https://documentation.wazuh.com
   Main PID: 526583 (java)
      Tasks: 43 (limit: 4613)
     Memory: 1.3G
     CGroup: /system.slice/wazuh-indexer.service
             └─526583 /usr/share/wazuh-indexer/jdk/bin/java -Xshare:auto -Dopensearch.network>

Jan 27 04:24:51 ubuntu2004 systemd[1]: Starting Wazuh-indexer...
Jan 27 04:25:31 ubuntu2004 systemd-entrypoint[526583]: WARNING: An illegal reflective access >
Jan 27 04:25:31 ubuntu2004 systemd-entrypoint[526583]: WARNING: Illegal reflective access by >
Jan 27 04:25:31 ubuntu2004 systemd-entrypoint[526583]: WARNING: Please consider reporting thi>
Jan 27 04:25:31 ubuntu2004 systemd-entrypoint[526583]: WARNING: Use --illegal-access=warn to >
Jan 27 04:25:31 ubuntu2004 systemd-entrypoint[526583]: WARNING: All illegal access operations>
Jan 27 04:26:05 ubuntu2004 systemd[1]: Started Wazuh-indexer.
lines 1-17/17 (END)
```

Cluster initialization

STEP 1 - Run the Wazuh indexer `indexer-security-init.sh` Script.

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

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# /usr/share/wazuh-indexer/bin/indexer-security-init.sh
Security Admin v7
Will connect to 192.168.92.128:9300 ... 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
```

STEP 2 - Testing the Cluster Installation

```
# curl -k -u admin:admin https://<WAZUH_INDEXER_IP>:9200
```

```
ubuntu@ubuntu-virtual-machine:~$ curl -k -u admin:admin https://192.168.92.128:9
200
{
  "name" : "node-1",
  "cluster_name" : "wazuh-cluster",
  "cluster_uuid" : "4LyvEgkxQA66zg2vcVfN8A",
  "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/"
```

STEP 3 - To check the single node or multiple node is working

```
ubuntu@ubuntu-virtual-machine:~$ curl -k -u admin:admin https://192.168.92.128:9200/
_cat/nodes?v
ip             heap.percent ram.percent cpu load_1m load_5m load_15m node.role maste
r name
192.168.92.128           36          95  -1    0.27    0.62     0.73 dimr      *
  node-1
```

Installation of Wazuh Server Step by Step

The Wazuh server is a central component that includes the Wazuh manager and Filebeat. The Wazuh manager collects and analyses data from the deployed Wazuh agents. It triggers alerts when threats or anomalies are detected. Filebeat securely forwards alerts and archived events to the Wazuh indexer.

The installation process is divided into two stages

1. Wazuh server node installation
2. Cluster configuration for multi-node deployment

1. Wazuh server node installation

a. Install the following packages if missing

```
# apt-get install gnupg apt-transport-https
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get install gnupg apt-transpo
rt-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
apt-transport-https is already the newest version (2.4.8).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 266 not upgraded.
```

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@ubuntu-virtual-machine:/home/ubuntu/ajay# curl -s https://packages.wazuh.co
m/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/k
eyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: key 96B3EE5F29111145: "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" n
ot changed
gpg: Total number processed: 1
gpg:              unchanged: 1
```

Adding repository & then Update packages

```
# 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

# apt-get update
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# echo "deb [signed-by=/usr/share/k
eyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /et
c/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt
/ stable main
```

b. Installing wazuh manager

Install the Wazuh manager package.

```
# apt-get -y install wazuh-manager
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get -y install wazuh-manager
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
Suggested packages:
  expect
The following NEW packages will be installed:
  wazuh-manager
0 upgraded, 1 newly installed, 0 to remove and 266 not upgraded.
Need to get 120 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-manager amd64 4.3.1
0-1 [120 MB]
Fetched 120 MB in 10s (11.5 MB/s)
Selecting previously unselected package wazuh-manager.
(Reading database ... 196586 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) ...
```

Enable and start the Wazuh manager service.

```
# systemctl daemon-reload
# systemctl enable wazuh-manager
# systemctl start wazuh-manager
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# systemctl daemon-reload
root@ubuntu-virtual-machine:/home/ubuntu/ajay# systemctl enable wazuh-manager
Synchronizing state of wazuh-manager.service with SysV service script with /lib/syst
emd/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@ubuntu-virtual-machine:/home/ubuntu/ajay# systemctl start wazuh-manager
```

Run the following command to verify the Wazuh manager status

```
# systemctl status wazuh-manager
```

```
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor pre>
     Active: active (running) since Sat 2023-01-28 13:38:22 IST; 17s ago
    Process: 46489 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=>
      Tasks: 113 (limit: 2247)
     Memory: 516.0M
        CPU: 42.912s
     CGroup: /system.slice/wazuh-manager.service
             ├─46542 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts>
             ├─46582 /var/ossec/bin/wazuh-authd
             ├─46598 /var/ossec/bin/wazuh-db
             ├─46621 /var/ossec/bin/wazuh-execd
             ├─46624 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts>
             ├─46627 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts>
             ├─46641 /var/ossec/bin/wazuh-analysisd
             ├─46684 /var/ossec/bin/wazuh-syscheckd
             ├─46700 /var/ossec/bin/wazuh-remoted
             ├─46732 /var/ossec/bin/wazuh-logcollector
             ├─46752 /var/ossec/bin/wazuh-monitord
             └─46773 /var/ossec/bin/wazuh-modulesd

Jan 28 13:38:12 ubuntu-virtual-machine env[46489]: Started wazuh-db...
Jan 28 13:38:13 ubuntu-virtual-machine env[46489]: Started wazuh-execd...
```

c. Installing filebeat

Install the Filebeat package

apt-get -y install filebeat

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get -y install filebeat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 266 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 2s (10.5 MB/s)
Selecting previously unselected package filebeat.
(Reading database ... 215277 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) ...
```

Configuring filebeat

Step 1 - Download the preconfigured Filebeat configuration file.

```
# curl -so /etc/filebeat/filebeat.yml
https://packages.wazuh.com/4.3/tpl/wazuh/filebeat/filebeat.yml
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# curl -so /etc/filebeat/filebeat.yml h
ttps://packages.wazuh.com/4.3/tpl/wazuh/filebeat/filebeat.yml
```

Step 2 - Edit the filebeat.yml file

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.

```
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["192.168.92.128:9200"]
  protocol: https
  username: ${username}
  password: ${password}
```

Step 3 - Create a filebeat keystore

Create a Filebeat keystore to securely store authentication credentials.

```
# filebeat keystore create
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# filebeat keystore create
Created filebeat keystore
```

Step 4 - add user & Password

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
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# echo admin | filebeat keystore add us
ername --stdin --force
Successfully updated the keystore
root@ubuntu-virtual-machine:/home/ubuntu/ajay# echo admin | filebeat keystore add pa
ssword --stdin --force
Successfully updated the keystore
```

Step 5 - alert template

Download the alerts template for the Wazuh indexer.

```
# curl -so /etc/filebeat/wazuh-template.json
https://raw.githubusercontent.com/wazuh/wazuh/4.3/extensions/elast
icsearch/7.x/wazuh-template.json
# chmod go+r /etc/filebeat/wazuh-template.json
```

Step 6 - wazuh module

Install the Wazuh module for Filebeat.

```
# curl -s
https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.2.tar.gz
| tar -xvz -C /usr/share/filebeat/module
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# chmod go+r /etc/filebeat/wazuh-templa
te.json
root@ubuntu-virtual-machine:/home/ubuntu/ajay# curl -s https://packages.wazuh.com/4.
x/filebeat/wazuh-filebeat-0.2.tar.gz | tar -xvz -C /usr/share/filebeat/module
wazuh/alerts/
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/manifest.yml
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/archives/
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/manifest.yml
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/module.yml
```

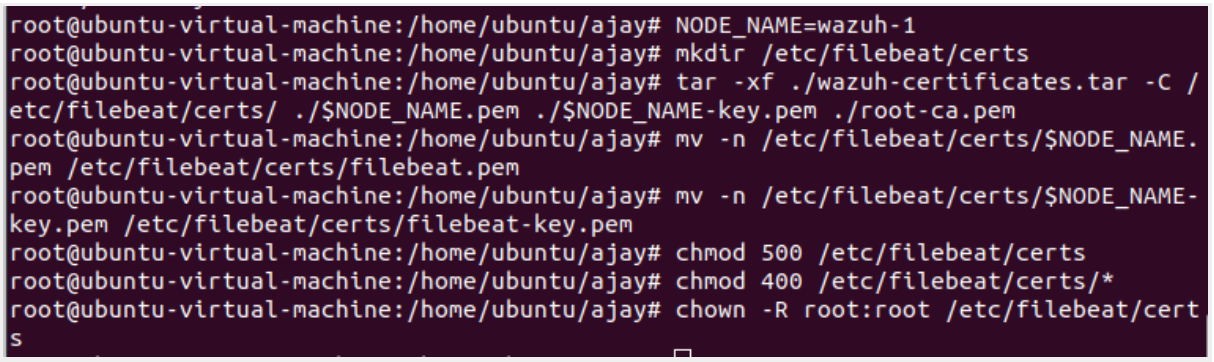
Deploying certificates

Step 1

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=<server-node-name>

# 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
```



Starting filebeat service

Step 2 - start service

Enable and start the Filebeat service.

```
# systemctl daemon-reload
# systemctl enable filebeat
# systemctl start filebeat
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get install debhelper tar curl li
bcap2-bin #debhelper version 9 or later
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libcap2-bin is already the newest version (1:2.44-1build3).
libcap2-bin set to manually installed.
tar is already the newest version (1.34+dfsg-1build3).
tar set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.7).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev binutils binutils-common
  binutils-x86-64-linux-gnu build-essential cpp-11 debugedit dh-autoreconf
  dh-strip-nondeterminism dpkg-dev dwz fakeroot g++ g++-11 gcc gcc-11 gcc-11-base
  gcc-12-base gettext intltool-debian libalgorithm-diff-perl
```

Step 3 - verify

Run the following command to verify that Filebeat is successfully installed

```
# filebeat test output
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get install gnupg apt-transport-h
ttps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
apt-transport-https is already the newest version (2.4.8).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 266 not upgraded.
```

Additional info: Cluster configuration is for multi-node. As for now we are only installing a single node so we don't need cluster configuration.

Installing wazuh Dash board

Installing package dependencies

Install the following packages if missing.

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

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get install debhelper tar curl li
bcap2-bin #debhelper version 9 or later
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libcap2-bin is already the newest version (1:2.44-1build3).
libcap2-bin set to manually installed.
tar is already the newest version (1.34+dfsg-1build3).
tar set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.7).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev binutils binutils-common
  binutils-x86-64-linux-gnu build-essential cpp-11 debugedit dh-autoreconf
  dh-strip-nondeterminism dpkg-dev dwz fakeroot g++ g++-11 gcc gcc-11 gcc-11-base
  gcc-12-base gettext intltool-debian libalgorithm-diff-perl
```

Adding Wazuh repository

Install the following packages if missing.

```
# apt-get install gnupg apt-transport-https
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get install gnupg apt-transport-h
ttps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
apt-transport-https is already the newest version (2.4.8).
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 266 not upgraded.
```

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@ubuntu-virtual-machine:/home/ubuntu/ajay# curl -s https://packages.wazuh.com/ke
y/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/
wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: key 96B3EE5F29111145: "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" not c
hanged
gpg: Total number processed: 1
gpg:               unchanged: 1
```

Add repository & Update

```
# 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@ubuntu-virtual-machine:/home/ubuntu/ajay# echo "deb [signed-by=/usr/share/keyri
ngs/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/so
urces.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ st
able main
```

Installing the wazuh Dashboard

Install the Wazuh dashboard package.

```
# apt-get -y install wazuh-dashboard
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# apt-get -y install wazuh-dashboard
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwe-hwdb
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  wazuh-dashboard
0 upgraded, 1 newly installed, 0 to remove and 266 not upgraded.
Need to get 130 MB of archives.
After this operation, 635 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-dashboard amd64 4.3
.10-1 [130 MB]
Fetched 130 MB in 14s (9,559 kB/s)
Selecting previously unselected package wazuh-dashboard.
(Reading database ... 215596 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) ...
```

Configuration of Wazuh documentation

Edit the `/etc/wazuh-dashboard/opensearch_dashboards.yml` file and replace the following values:

- `server.host`: This setting specifies the host of the Wazuh dashboard server. To allow remote users to connect, set the value to the IP address or DNS name of the Wazuh dashboard server. 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.

```
server.host: 0.0.0.0
server.port: 443
opensearch.hosts: https://localhost:9200
opensearch.ssl.verificationMode: certificate
```

Deploying Certificates

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-node-name>

# 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@ubuntu-virtual-machine:/home/ubuntu/ajay# NODE_NAME=dashboard
root@ubuntu-virtual-machine:/home/ubuntu/ajay# mkdir /etc/wazuh-dashboard/certs
root@ubuntu-virtual-machine:/home/ubuntu/ajay# tar -xf ./wazuh-certificates.tar -C /
etc/wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem
root@ubuntu-virtual-machine:/home/ubuntu/ajay# mv -n /etc/wazuh-dashboard/certs/$NOD
E_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem
root@ubuntu-virtual-machine:/home/ubuntu/ajay# mv -n /etc/wazuh-dashboard/certs/$NOD
E_NAME-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem
root@ubuntu-virtual-machine:/home/ubuntu/ajay# chmod 500 /etc/wazuh-dashboard/certs
root@ubuntu-virtual-machine:/home/ubuntu/ajay# chmod 400 /etc/wazuh-dashboard/certs/
*
root@ubuntu-virtual-machine:/home/ubuntu/ajay# chown -R wazuh-dashboard:wazuh-dashbo
ard /etc/wazuh-dashboard/certs
```

Starting the wazuh Dashboard service

Enable and start the Wazuh dashboard service

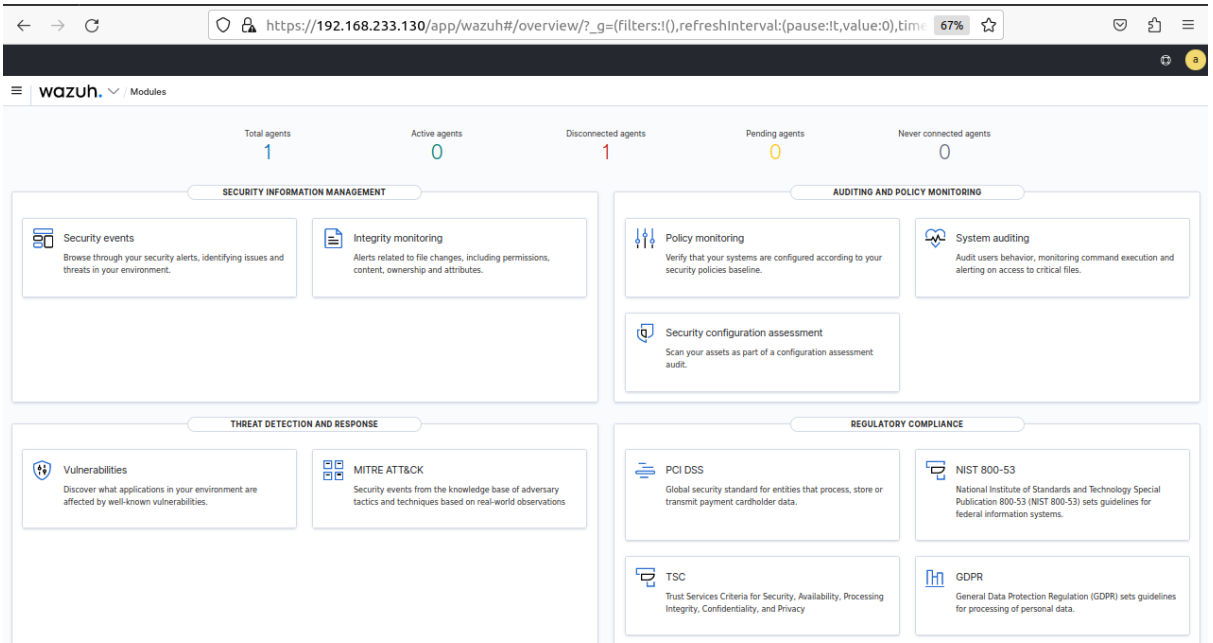
```
# systemctl daemon-reload
# systemctl enable wazuh-dashboard
# systemctl start wazuh-dashboard
```

```
root@ubuntu-virtual-machine:/home/ubuntu/ajay# systemctl daemon-reload
root@ubuntu-virtual-machine:/home/ubuntu/ajay# systemctl enable wazuh-dashboard
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-dashboard.service
→ /etc/systemd/system/wazuh-dashboard.service.
root@ubuntu-virtual-machine:/home/ubuntu/ajay# systemctl start wazuh-dashboard
root@ubuntu-virtual-machine:/home/ubuntu/ajay# systemctl status wazuh-dashboard
● wazuh-dashboard.service - wazuh-dashboard
     Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; vendor p>
     Active: active (running) since Sat 2023-01-28 14:32:48 IST; 9s ago
   Main PID: 53377 (node)
      Tasks: 11 (limit: 2247)
     Memory: 183.2M
        CPU: 5.475s
     CGroup: /system.slice/wazuh-dashboard.service
             └─53377 /usr/share/wazuh-dashboard/bin/../node/bin/node --no-warnings >

Jan 28 14:32:48 ubuntu-virtual-machine systemd[1]: Started wazuh-dashboard.
```

Successfully implemented Wazuh Server

Here is an Dashboard running on local system



—----------EOD—----------