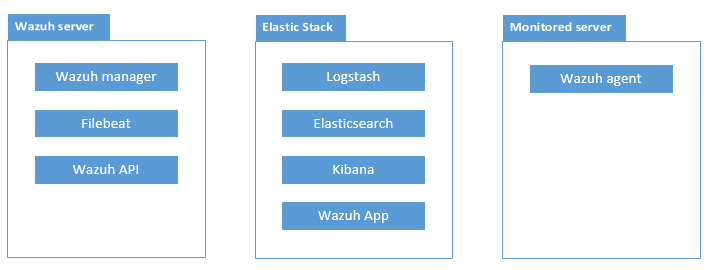
**Kiến trúc của Wazuh**

1. **Mô hình kiến trúc của Wazuh được chia thành 2 dạng :**

* Multi-node deployment
* Single node deployment

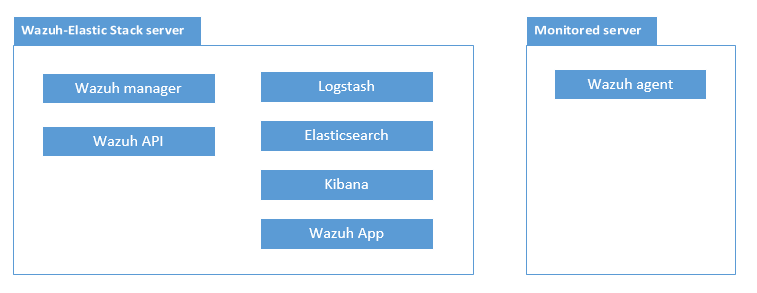
**Multi-node deployment**

Khi Wazuh server và Opensearch cluster chạy trên các host khác nhau, Filebeat được dùng để truyền một cách an toàn các cảnh báo, archived event tới Elasticsarch server sử dụng TLS. Chú ý rằng multi-node cluster sẽ là multiple Elastic stack sererver.

[](https://github.com/hocchudong/ghichep-SOC/blob/master/images/wazuh-03.png)

**Single-node deployment**

Wazuh và Elastic stack chạy với 1 single-node Opensearch cluster (số lượng agent < 50), có thể triển khai trên một single server. Ở triển khai này, Logstash sẽ đọc các cảnh báo, event từ Wazuh trực tiếp từ local file system và đẩy chúng tới local Opensearch instance.

[](https://github.com/hocchudong/ghichep-SOC/blob/master/images/wazuh-04.png)

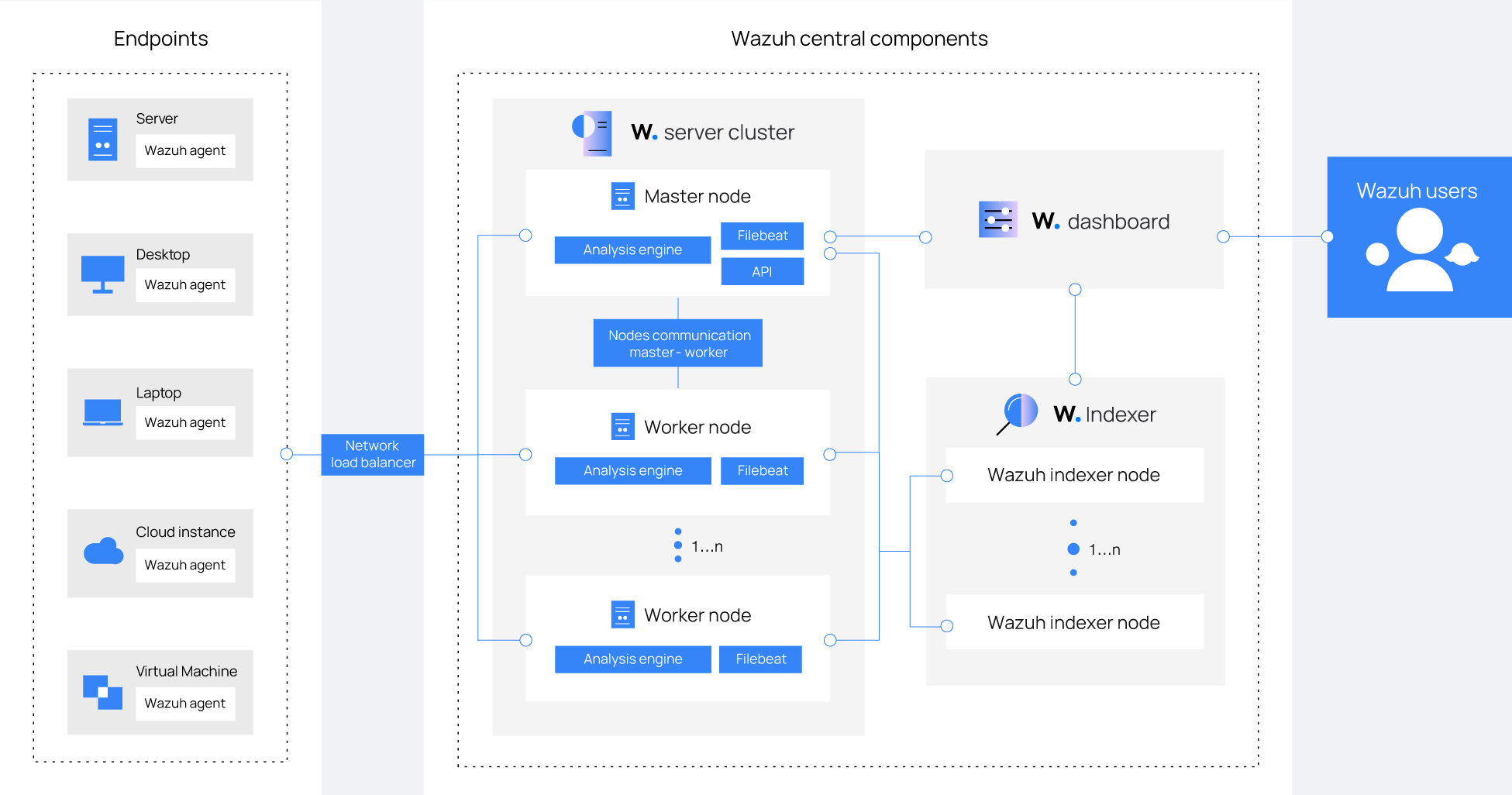
**Multi-node deployment**

Kiến trúc Wazuh dựa trên các tác nhân, chạy trên các điểm cuối được giám sát, chuyển tiếp dữ liệu bảo mật đến một máy chủ trung tâm. Các thiết bị không cần tác nhân như tường lửa, switches, routers, và điểm truy cập được hỗ trợ và có thể chủ động gửi dữ liệu nhật ký thông qua Syslog, SSH, hoặc sử dụng API của chúng. Máy chủ trung tâm giải mã và phân tích thông tin nhận được và chuyển kết quả tới cụm chỉ mục Wazuh để lập chỉ mục và lưu trữ.

Cụm chỉ mục Wazuh là một tập hợp của một hoặc nhiều nút giao tiếp với nhau để thực hiện các thao tác đọc và ghi trên các chỉ mục. Các triển khai Wazuh nhỏ, không yêu cầu xử lý lượng dữ liệu lớn, có thể dễ dàng được xử lý bởi một cụm chỉ mục đơn lẻ. Các cụm nhiều nút được khuyến nghị khi có nhiều điểm cuối được giám sát, khi dự đoán khối lượng dữ liệu lớn, hoặc khi yêu cầu khả năng sẵn có cao.

Đối với môi trường sản xuất, nên triển khai máy chủ Wazuh và cụm chỉ mục Wazuh trên các máy chủ khác nhau. Trong trường hợp này, Filebeat được sử dụng để chuyển tiếp các cảnh báo và sự kiện đã được lưu trữ của Wazuh đến cụm chỉ mục Wazuh (cụm đơn lẻ hoặc nhiều nút) một cách an toàn bằng cách sử dụng mã hóa TLS.

Sơ đồ dưới đây đại diện cho kiến trúc triển khai Wazuh. Nó cho thấy các thành phần của giải pháp và cách mà máy chủ Wazuh cùng các nút lập chỉ mục Wazuh có thể được cấu hình thành các cụm, cung cấp cân bằng tải và tính sẵn sàng cao.



1. **Agent-server communication**

Wazuh agent sử dụng OESSEC message protocol để gửi các event thu thập được tới Wazhu server thông qua port 1514 (UDP hoặc TCP). Wazuh server giải mã và thực hiện rule-check với các event nhận được với công cụ phân tích. Các event ứng với các rule được bổ sung dữ liệu cảnh báo như rule-id và rule-name. Các event có thể được đẩy tới 1 hoặc cả 2 file sau, dựa vào việc có được event có tương ứng với rule hay không :

* File /var/ossec/logs/archives/archives.json chứa tất cả các event dù có tương ứng với rule hay không.
* File /var/ossec/logs/alerts/alerts.json chỉ chứa các event tương ứng với rule đã đặt ra.

OSSEC message protocol dùng 1 mã hóa 192-bit Blowfish với full 16-round implemetation.

**Wazuh-elastic communication**

Trên mô hình triển khai diện rộng, Wazuh server sử dụng Filebeat để chuyển dữ liệu về cảnh báo và event tới Logstash (5000/TCP) trên Elastic Stack server, sử dụng TLS. Với kiến trúc single-host, Logstash đọc trực tiếp từ local file system mà không cần dùng Filebeat.

Logstash định hình incoming data, và có thể là GeoIP, trước gửi tới Opensearch (port 9200/TCP). Một khi dữ liệu được index tới Opensearch, Kibana (port 5601/TCP) được dùng để khai thác và hiển thị thông tin.

Wazuh App chạy bên trong Kibana liên tục truy vấn tới RESTful API (port 55000/TCP trên Wazuh manager) để hiển thị cấu hình và thông tin trạng thái liên quan của server và agent, cũng như restart agent theo yêu cầu. Liên lạc này được mã hóa với TLS và được xác thực với username và password.

**Lưu trữ dữ liệu**

Các event về alert và non-alert được lưu trữ cùng nhau trong file trên Wazuh server và sau để gửi tới Opensearch. Các file này được viết với dạng JSON hoặc với plain text format. Các file này được nén hàng ngày và được đánh dấu với MD5 và SHA1 checksums. Cấu trúc thư mục và tên file như sau :

A screen shot of a computer

Description automatically generated

**DEMO**

**Cài đặt mô hình cluster Multi-node deployment gồm 3 node:**

A black background with white text

Description automatically generated

**File cấu hình của cụm Wazuh sử dụng Ansible**

all:

hosts:

############# Wazuh indexer #############

wi1:

ansible\_host: 192.168.100.50

private\_ip: 192.168.100.50

indexer\_node\_name: node-1

wi2:

ansible\_host: 192.168.100.179

private\_ip: 192.168.100.179

indexer\_node\_name: node-2

wi3:

ansible\_host: 192.168.100.180

private\_ip: 192.168.100.180

indexer\_node\_name: node-3

############# Wazuh servers #############

manager:

ansible\_host: 192.168.100.50

private\_ip: 192.168.100.50

dashboard:

ansible\_host: 192.168.100.50

private\_ip: 192.168.100.50

worker1:

ansible\_host: 192.168.100.179

private\_ip: 192.168.100.179

worker2:

ansible\_host: 192.168.100.180

private\_ip: 192.168.100.180

# Group of above hosts

children:

wi\_cluster:

hosts:

wi1:

wi2:

wi3:

wazuh\_cluster:

hosts:

manager:

worker1:

worker2:

**File Playbook Ansible**

---

# Certificates generation

- hosts: wi1

roles:

- role: ../roles/wazuh/wazuh-indexer

indexer\_network\_host: "{{ private\_ip }}"

indexer\_cluster\_nodes:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

indexer\_discovery\_nodes:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

perform\_installation: false

become: yes

vars:

indexer\_node\_master: true

instances:

node1:

name: node-1 # Important: must be equal to indexer\_node\_name.

ip: "{{ hostvars.wi1.private\_ip }}" # When unzipping, the node will search for its node name folder to get the cert.

role: indexer

node2:

name: node-2

ip: "{{ hostvars.wi2.private\_ip }}"

role: indexer

node3:

name: node-3

ip: "{{ hostvars.wi3.private\_ip }}"

role: indexer

node4:

name: node-4

ip: "{{ hostvars.manager.private\_ip }}"

role: wazuh

node\_type: master

node5:

name: node-5

ip: "{{ hostvars.worker1.private\_ip }}"

role: wazuh

node\_type: worker

node6:

name: node-6

ip: "{{ hostvars.dashboard.private\_ip }}"

role: dashboard

node7:

name: node-7

ip: "{{ hostvars.worker2.private\_ip }}"

role: wazuh

node\_type: worker

tags:

- generate-certs

# Wazuh indexer cluster

- hosts: wi\_cluster

strategy: free

roles:

- role: ../roles/wazuh/wazuh-indexer

indexer\_network\_host: "{{ private\_ip }}"

become: yes

become\_user: root

vars:

indexer\_cluster\_nodes:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

indexer\_discovery\_nodes:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

indexer\_node\_master: true

instances:

node1:

name: node-1 # Important: must be equal to indexer\_node\_name.

ip: "{{ hostvars.wi1.private\_ip }}" # When unzipping, the node will search for its node name folder to get the cert.

role: indexer

node2:

name: node-2

ip: "{{ hostvars.wi2.private\_ip }}"

role: indexer

node3:

name: node-3

ip: "{{ hostvars.wi3.private\_ip }}"

role: indexer

node4:

name: node-4

ip: "{{ hostvars.manager.private\_ip }}"

role: wazuh

node\_type: master

node5:

name: node-5

ip: "{{ hostvars.worker1.private\_ip }}"

role: wazuh

node\_type: worker

node6:

name: node-6

ip: "{{ hostvars.dashboard.private\_ip }}"

role: dashboard

node7:

name: node-7

ip: "{{ hostvars.worker2.private\_ip }}"

role: wazuh

node\_type: worker

# Wazuh cluster

- hosts: manager

roles:

- role: "../roles/wazuh/ansible-wazuh-manager"

- role: "../roles/wazuh/ansible-filebeat-oss"

filebeat\_node\_name: node-4

become: yes

become\_user: root

vars:

wazuh\_manager\_config:

connection:

- type: 'secure'

port: '1514'

protocol: 'tcp'

queue\_size: 131072

api:

https: 'yes'

cluster:

disable: 'no'

node\_name: 'master'

node\_type: 'master'

key: 'c98b62a9b6169ac5f67dae55ae4a9088'

nodes:

- "{{ hostvars.manager.private\_ip }}"

hidden: 'no'

wazuh\_api\_users:

- username: "{{ api\_username }}"

password: "{{ api\_password }}"

filebeat\_output\_indexer\_hosts:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

- hosts: worker1

roles:

- role: "../roles/wazuh/ansible-wazuh-manager"

- role: "../roles/wazuh/ansible-filebeat-oss"

filebeat\_node\_name: node-5

become: yes

become\_user: root

vars:

wazuh\_manager\_config:

connection:

- type: 'secure'

port: '1514'

protocol: 'tcp'

queue\_size: 131072

connection\_syslog:

- type: 'syslog'

port: '514'

protocol: 'tcp'

allowed\_ips: ['0.0.0.0/0']

local\_ip: "{{ hostvars.worker1.private\_ip }}"

api:

https: 'yes'

cluster:

disable: 'no'

node\_name: 'worker\_01'

node\_type: 'worker'

key: 'c98b62a9b6169ac5f67dae55ae4a9088'

nodes:

- "{{ hostvars.manager.private\_ip }}"

hidden: 'no'

filebeat\_output\_indexer\_hosts:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

- hosts: worker2

roles:

- role: "../roles/wazuh/ansible-wazuh-manager"

- role: "../roles/wazuh/ansible-filebeat-oss"

filebeat\_node\_name: node-7

become: yes

become\_user: root

vars:

wazuh\_manager\_config:

connection:

- type: 'secure'

port: '1514'

protocol: 'tcp'

queue\_size: 131072

connection\_syslog:

- type: 'syslog'

port: '514'

protocol: 'tcp'

allowed\_ips: ['0.0.0.0/0']

local\_ip: "{{ hostvars.worker2.private\_ip }}"

api:

https: 'yes'

cluster:

disable: 'no'

node\_name: 'worker\_02'

node\_type: 'worker'

key: 'c98b62a9b6169ac5f67dae55ae4a9088'

nodes:

- "{{ hostvars.manager.private\_ip }}"

hidden: 'no'

filebeat\_output\_indexer\_hosts:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

# Wazuh dashboard node

- hosts: dashboard

roles:

- role: "../roles/wazuh/wazuh-dashboard"

become: yes

become\_user: root

vars:

indexer\_network\_host: "{{ hostvars.wi1.private\_ip }}"

indexer\_cluster\_nodes:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

dashboard\_node\_name: node-6

wazuh\_api\_credentials:

- id: default

url: https://{{ hostvars.manager.private\_ip }}

port: 55000

username: "{{ api\_username }}"

password: "{{ api\_password }}"

ansible\_shell\_allow\_world\_readable\_temp: true

# Indexer + dashboard node

- hosts: dashboard

roles:

- role: "../roles/wazuh/wazuh-indexer"

- role: "../roles/wazuh/wazuh-dashboard"

become: yes

become\_user: root

vars:

indexer\_network\_host: "{{ hostvars.dashboard.private\_ip }}"

indexer\_node\_name: node-6

indexer\_node\_master: false

indexer\_node\_ingest: false

indexer\_node\_data: false

indexer\_cluster\_nodes:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

indexer\_discovery\_nodes:

- "{{ hostvars.wi1.private\_ip }}"

- "{{ hostvars.wi2.private\_ip }}"

- "{{ hostvars.wi3.private\_ip }}"

dashboard\_node\_name: node-6

wazuh\_api\_credentials:

- id: default

url: https://{{ hostvars.manager.private\_ip }}

port: 55000

username: "{{ api\_username }}"

password: "{{ api\_password }}"

instances:

node1:

name: node-1

ip: "{{ hostvars.wi1.private\_ip }}" # When unzipping, the node will search for its node name folder to get the cert.

role: indexer

node2:

name: node-2

ip: "{{ hostvars.wi2.private\_ip }}"

role: indexer

node3:

name: node-3

ip: "{{ hostvars.wi3.private\_ip }}"

role: indexer

node4:

name: node-4

ip: "{{ hostvars.manager.private\_ip }}"

role: wazuh

node\_type: master

node5:

name: node-5

ip: "{{ hostvars.worker1.private\_ip }}"

role: wazuh

node\_type: worker

node6:

name: node-6

ip: "{{ hostvars.dashboard.private\_ip }}"

role: dashboard

node7:

name: node-7

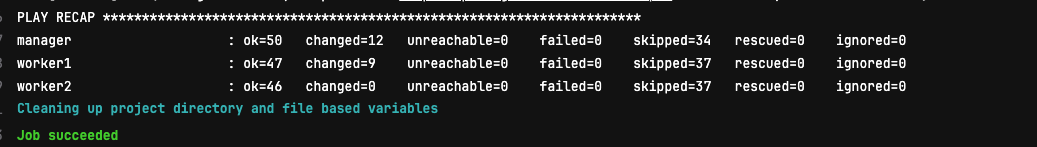
ip: "{{ hostvars.worker2.private\_ip }}"

role: wazuh

node\_type: worker

ansible\_shell\_allow\_world\_readable\_temp: true

**Sau khi chạy xong tiến trình cài đặt cluster**

****

**A black screen with white text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**