**Wazuh**

**Kubernetes configuration**

**Pre-requisites**

1. A Kubernetes cluster already deployed.
2. Resource Requirement

To deploy Wazuh on Kubernetes, the cluster should have at least the following resources available:

* 2 CPU units
* 3 Gi of memory
* 2 Gi of storage

**StatefulSet:** Manages Pods with unique identities. Ideal for stateful applications like databases, as each Pod keeps its identity and state, even if rescheduled. Used for Wazuh manager and Wazuh indexer to maintain their states across restarts.

**Deployment:** Manages stateless Pods. Suitable for applications where state is not maintained, like the Wazuh dashboard.

**Persistent Volumes:** Provide storage in a cluster, independent of the lifecycle of Pods. Used to store data for both Wazuh manager and Wazuh indexer.

This way, StatefulSets ensure that Wazuh components maintain their state, while Deployments keep the Wazuh dashboard lightweight and stateless. Persistent Volumes provide reliable storage for stateful data.

[https://kubernetes.io/docs/concepts/storage/persistent-volumes/](file:///C:\Users\Swapnil.Gangurde\Documents\Custom%20Office%20Templates)

**Wazuh Components**

1. **Wazuh Master**

**Purpose:** Central node for coordinating worker nodes and ensuring data consistency.

**Image:** wazuh/wazuh-manager

**Controller:** StatefulSet

**Services:**

Wazuh API: wazuh-master.your-domain.com:55000

Agent registration service (authd): wazuh-master.your-domain.com:1515

1. **Wazuh Workers (0 / 1)**

Purpose: Worker nodes that receive agent events.

Image: wazuh/wazuh-manager

Controller: StatefulSet

Service: Reporting service: wazuh-manager.your-domain.com:1514

1. **Wazuh Indexer**

Purpose: Ingests events received from Filebeat, Communication for Wazuh indexer nodes, API used by the Wazuh dashboard to read/write alerts

Image: wazuh/wazuh-indexer

Controller: StatefulSet

Services:

Communication for indexer nodes: wazuh-indexer

Indexer API: indexer

1. **Wazuh Dashboard**

Purpose: Visualizes Wazuh indexer data and provides Wazuh app features, Access the Wazuh dashboard

Image: wazuh/wazuh-dashboard

Controller: Deployment

Service: Dashboard access: https://wazuh.your-domain.com:443

**Deployment**

Clone this repository to deploy the necessary services and pods.

>> git clone https://github.com/wazuh/wazuh-kubernetes.git -b v4.7.5 --depth=1

>> cd wazuh-kubernetes

**Setup SSL certificates**

**Generating and Using Certificates for Wazuh**

1. **Wazuh Indexer:** Use the script wazuh/certs/indexer\_cluster/generate\_certs.sh or provide your own certificates.
2. **Wazuh Dashboard:** Use the script wazuh/certs/dashboard\_http/generate\_certs.sh or provide your own certificates.

**Importing Certificates**

The certificates are imported using secretGenerator in the kustomization.yml file:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*kustomization.yaml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

secretGenerator:

- name: indexer-certs

files:

- certs/indexer\_cluster/root-ca.pem

- certs/indexer\_cluster/node.pem

- certs/indexer\_cluster/node-key.pem

- certs/indexer\_cluster/dashboard.pem

- certs/indexer\_cluster/dashboard-key.pem

- certs/indexer\_cluster/admin.pem

- certs/indexer\_cluster/admin-key.pem

- certs/indexer\_cluster/filebeat.pem

- certs/indexer\_cluster/filebeat-key.pem

- name: dashboard-certs

files:

- certs/dashboard\_http/cert.pem

- certs/dashboard\_http/key.pem

- certs/indexer\_cluster/root-ca.pem

**Applying Manifests using Kustomize**

1. **Check Cluster Type:** Determine whether you're using an EKS cluster or another type.
2. **Adjust Resources:** If needed, you can edit resources like CPU, memory, and storage in the patches located in envs/eks/ or envs/local-env/.
3. **Apply Manifests:**

>> kubectl apply -k envs/local-env/

**Verifying Deployment**

1. **Namespace:**

>> kubectl get namespaces | grep wazuh

Output:

>> wazuh Active 12m >>

1. **Services:**

>> kubectl get services -n wazuh

1. **Deployments:**

>> kubectl get deployments -n wazuh

1. **StatefulSets:**

>> kubectl get statefulsets -n wazuh

1. **Pods:**

>> kubectl get pods -n wazuh

**Accessing Wazuh Dashboard**

1. **Using Domain Name (If Available):**

>> https://wazuh.your-domain.com

1. **Using External IP Address:**

>> kubectl get services -o wide -n wazuh

Look for the **EXTERNAL-IP** field.

1. **Using Port Forward (For Local Cluster):**

>> kubectl -n wazuh port-forward --address <INTERFACE\_IP\_ADDRESS> service/dashboard 8443:443

Access the dashboard at **https://<INTERFACE\_IP\_ADDRESS>:8443.**

**Changing Passwords**

1. **Wazuh Indexer Users:**

* Change passwords for admin and kibanaserver users in internal\_users.yml.
* Encode the new password in base64 format.
* Update the password field in indexer-cred-secret.yaml or dashboard-cred-secret.yaml.
* Apply changes.

1. **Wazuh API Users:**

* Change the password for the wazuh-wui user in wazuh-api-cred-secret.yaml.
* Encode the new password in base64 format.
* Apply changes.

1. **Restart Pods (If Needed)**

* Restart pods for Wazuh dashboard and Wazuh manager master.

1. **Enrolling Agents**

* Install the Wazuh agent on hosts.
* Modify /var/ossec/etc/ossec.conf to set the transport protocol to TCP and specify the manager IP address or hostname.