# Kubernetes'te Dağıtım

Bu bölümde Wazuh'un Kubernetes'e kurulum, yükseltme ve temizleme süreci gösterilmektedir. Kubernetes açık kaynaklı bir konteyner düzenleme motorudur. Konteynerler, bağımlılıkları ve yapılandırmalarıyla paketlenmiş mikro hizmetlerdir. Kubernetes, konteynerleştirilmiş uygulamaların dağıtımını, ölçeklenmesini ve yönetimini otomatikleştirerek bir kümede çalışmak üzere tasarlanmıştır. Birden fazla sunucuya dağıtılan birden fazla konteyneri kapsayan uygulamaların çalışmasını basitleştirir. Kolay yönetim ve keşif için konteynerler, Kubernetes için temel operasyonel birim olan pod'lara gruplandırılır. Kubernetes pod'ları, yüksek kullanılabilirlik sağlamak için düğümler arasında dağıtılır. Kubernetes, konteynerlerinizi çalıştıran tüm Kubernetes düğümlerinde ağ oluşturma, yük dengeleme, güvenlik ve ölçekleme konusunda yardımcı olur. Belgelerin bu bölümünde, Wazuh Kubernetes deposunu nasıl klonlayacağınızı , sertifikaları nasıl ayarlayacağınızı, manifestoları nasıl uygulayacağınızı ve bulutta ve yerel ortamlarda Kubernetes'e Wazuh'u kurmak için gerekli pod'ları ve hizmetleri nasıl dağıtacağınızı göreceksiniz. Ayrıca, Kubernetes yapılandırma alt bölümünü bulacaksınız ve Kubernetes'e yüklenen Wazuh'u Yükselt alt bölümünde uygulamanızı nasıl yükselteceğinizi öğreneceksiniz . Son olarak, Temizleme alt bölümünde hem kümeleri hem de birimleri nasıl temizleyeceğinizi göreceksiniz .


- Kubernetes Yapılandırması
- Kubernetes'e Yüklenen Wazuh'u Yükseltin
- Dağıtım
- Temizlemek

# Kubernetes Yapılandırması

## Ön Koşullar

- Zaten dağıtılmış bir Kubernetes kümesi.
- Kubernetes 1.23 ve üzeri sürümlerini kullanan Amazon EKS dağıtımları için bir Amazon EBS CSI sürücüsü IAM rolü. CSI sürücüsünün düzgün çalışması için bir IAM rolü atamanız gerekir. Amazon EBS CSI sürücüsü IAM rolünü oluşturma talimatlarını bulmak için AWS belgelerini okuyun . Hem yeni hem de eski dağıtımlar için CSI sürücüsünü yüklemeniz gerekir. CSI sürücüsü, temel bir Kubernetes özelliğidir.

## Kaynak Gereksinimi

Wazuh'u Kubernetes'e dağıtmak için kümede en azından aşağıdaki kaynakların bulunması gerekir:

- 2 CPU ünitesi
- 3 Gi hafıza
- 2 Gi depolama

## Genel Bakış

### StatefulSet ve Dağıtım Denetleyicileri

*Bir Dağıtım* olarak , bir *StatefulSet,* aynı kapsayıcı spesifikasyonuna dayanan Pod'ları yönetir, ancak her bir pod'una bağlı bir kimliği korur. Bu pod'lar aynı spesifikasyondan oluşturulur, ancak birbirlerinin yerine kullanılamazlar: her birinin herhangi bir yeniden planlama boyunca korunan kalıcı bir tanımlayıcısı vardır.

Verileri kalıcı depolamaya kaydeden veritabanları gibi durumlu uygulamalar için kullanışlıdır. Her Wazuh yöneticisinin ve her Wazuh dizinleyicisinin durumları korunmalıdır, bu nedenle her başlatmada durumlarını koruduklarından emin olmak için bunları StatefulSet kullanarak bildiririz.

Dağıtımlar durumsuz kullanım için tasarlanmıştır ve oldukça hafiftir ve durumların bakımının gerekli olmadığı Wazuh panosu için uygun görünmektedir.

Kalıcı birimler (PV), sağlanan kümedeki depolama parçalarıdır. Bir düğümün küme kaynağı olması gibi kümedeki bir kaynaktır. Kalıcı birimler, Birimler gibi birim eklentileridir ancak PV'yi kullanan herhangi bir bireysel pod'dan bağımsız bir yaşam döngüsüne sahiptir. Bu API nesnesi, NFS, iSCSI veya bulut sağlayıcıya özgü bir depolama sistemi olsun, depolamanın uygulanmasının ayrıntılarını yakalar.

Burada, hem Wazuh yöneticisinden hem de Wazuh dizinleyicisinden gelen verileri depolamak için kalıcı birimleri kullanıyoruz.

Daha fazla bilgi için kalıcı birimler sayfasına bakın .

# Baklalar

Wazuh docker konteynerlerinin nasıl oluşturulduğunu depolardan inceleyebilirsiniz .

**Wazuh master**

Bu pod, Wazuh kümesinin ana düğümünü içerir. Ana düğüm, çalışan düğümlerini merkezileştirir ve koordine eder, kritik ve gerekli verilerin tüm düğümler arasında tutarlı olmasını sağlar. Yönetim yalnızca bu düğümde gerçekleştirilir, bu nedenle aracı kayıt hizmeti (authd) buraya yerleştirilir.

| Resim | Kontroller |
|---|---|
| wazuh/wazuh-yöneticisi | Durumsal Küme |

**Wazuh worker 0 / 1**

Bu pod'lar Wazuh kümesinin bir işçi düğümünü içerir. Bunlar ajan olaylarını alacaktır.

| Resim | Kontroller |
|---|---|
| wazuh/wazuh-yöneticisi | Durumsal Küme |

**Wazuh indexer**

Wazuh dizinleyici pod'u Filebeat'ten alınan olayları alır.

| Resim | Kontroller |
|---|---|
| wazuh/wazuh-indeksleyici | Durumsal Küme |

**Wazuh dashboard**

Wazuh kontrol paneli, Wazuh dizinleyici verilerinizi, Wazuh aracı bilgileri ve sunucu yapılandırmasıyla birlikte görselleştirmenize olanak tanır.

| Resim | Kontroller |
| --- | --- |
| wazuh/wazuh-panosu | Dağıtım |

# Hizmetler

### Wazuh indeksleyici ve gösterge paneli

| İsim | Tanım |
| --- | --- |
| wazuh-indeksleyici | Wazuh indeksleyici düğümleri için iletişim. |
| dizinleyici | Bu, Wazuh panosunun uyarıları okumak/yazmak için kullandığı Wazuh dizinleyici API'sidir. |
| gösterge paneli | Wazuh kontrol paneli hizmeti. https://wazuh.your-domain.com:443 |

### Wazuh

| İsim | Tanım |
| --- | --- |
| vazuh | Wazuh API'si: wazuh-master.alan-adiniz.com:55000 |
| | Temsilci kayıt hizmeti (authd): wazuh-master.your-domain.com:1515 |
| wazuh-işçiler | Raporlama hizmeti: wazuh-manager.your-domain.com:1514 |
| wazuh-kümesi | Wazuh yönetici düğümleri için iletişim. |

# Kubernetes'e Yüklenen Wazuh'u Yükseltin

## Hangi Dosyaların Birime Aktarılacağının Kontrol Edilmesi

Kubernetes dağıtımımız Docker'dan Wazuh görüntülerimizi kullanır. Docker kullanarak Wazuh yapılandırmasından çıkarılan aşağıdaki koda bakarsak, yükseltmede hangi dizinlerin ve dosyaların kullanıldığını görebiliriz.

```
/var/ossec/api/configuration
/var/ossec/etc
/var/ossec/logs
/var/ossec/queue
/var/ossec/var/multigroups
/var/ossec/integrations
/var/ossec/active-response/bin
/var/ossec/agentless
/var/ossec/wodles
/etc/filebeat
/var/lib/filebeat
/usr/share/wazuh-dashboard/config/
/usr/share/wazuh-dashboard/certs/
/var/lib/wazuh-indexer
/usr/share/wazuh-indexer/certs/
/usr/share/wazuh-indexer/opensearch.yml
/usr/share/wazuh-indexer/opensearch-security/internal_users.yml
```

Bu dosyalarla ilgili herhangi bir değişiklik ilişkili birimde de yapılacaktır. Replika pod oluşturulduğunda, önceki değişiklikleri koruyarak bu dosyaları birimden alacaktır.

## Sertifikaların Yeniden Oluşturulması

v4.8.0'dan önceki bir sürümden yükseltme yapmak SSL sertifikalarını yeniden oluşturmanızı gerektirir. Bunun için SSL sertifikalarını kurun bölümündeki talimatları izleyin.

# Yükseltmeyi Yapılandırma

4.9 sürümüne yükseltmek için iki stratejiden birini izleyebilirsiniz.

- Varsayılan manifestoları kullanma : Bu strateji Wazuh 4.9 için varsayılan manifestoları kullanır. Güncel olmayan Wazuh sürümünüzün wazuh-kubernetes manifestolarını değiştirir.
- Özel bildirimleri tutma : Bu strateji, güncel olmayan Wazuh dağıtımınızın wazuh-kubernetes bildirimlerini korur. En son Wazuh sürümünün bildirimlerini yok sayar.

## Varsayılan Bildirimleri Kullanma

1. Wazuh-kubernetes'in güncel sürümü için etiketi inceleyin:

   ```
   git checkout v4.9.2
   ```

2. Yeni yapılandırmayı uygula

## Özel Beyannamelerin Tutulması

Wazuh 4.4'te bazı yollar önceki sürümlerdekilerden farklıdır. Özel bildirimlerinizi tutuyorsanız eski yolları yenileriyle güncellemelisiniz.

`old-path->new-path`

- `/usr/share/wazuh-dashboard/config/certs/->/usr/share/wazuh-dashboard/certs/`
- `/usr/share/wazuh-indexer/config/certs/->/usr/share/wazuh-indexer/certs/`
- `/usr/share/wazuh-indexer/plugins/opensearch-security/securityconfig/->/usr/share/wazuh-indexer/opensearch-security/`

Özel bildirimlerinizi koruyarak dağıtımınızı yükseltmek için aşağıdakileri yapın.

1. 4.3'ten güncelleme yapıyorsanız, aşağıdaki dosyaları düzenleyin ve 4.4'teki yeni yollarla güncelleyin. Aşağıdaki örneklerde her dosyanın yanında yeni yolları görebilirsiniz.

   - `wazuh/indexer_stack/wazuh-dashboard/dashboard-deploy.yaml`

     ```
     image: 'wazuh/wazuh-dashboard:4.9.2'
     mountPath: /usr/share/wazuh-dashboard/certs/cert.pem
     mountPath: /usr/share/wazuh-dashboard/certs/key.pem
     ```

```
mountPath: /usr/share/wazuh-dashboard/certs/root-ca.pem
value: /usr/share/wazuh-dashboard/certs/cert.pem
value: /usr/share/wazuh-dashboard/certs/key.pem
```

- `wazuh/indexer_stack/wazuh-dashboard/dashboard_conf/opensearch_dashboards.yml`

```
server.ssl.key: "/usr/share/wazuh-dashboard/certs/key.pem"
server.ssl.certificate: "/usr/share/wazuh-dashboard/certs/cert.pem"
opensearch.ssl.certificateAuthorities: ["/usr/share/wazuh-dashboard/certs/root-ca.pem"]
```

- `wazuh/indexer_stack/wazuh-indexer/cluster/indexer-sts.yaml`

```
image: 'wazuh/wazuh-indexer:4.9.2'
mountPath: /usr/share/wazuh-indexer/certs/node-key.pem
mountPath: /usr/share/wazuh-indexer/certs/node.pem
mountPath: /usr/share/wazuh-indexer/certs/root-ca.pem
mountPath: /usr/share/wazuh-indexer/certs/admin.pem
mountPath: /usr/share/wazuh-indexer/certs/admin-key.pem
mountPath: /usr/share/wazuh-indexer/opensearch.yml
mountPath: /usr/share/wazuh-indexer/opensearch-security/internal_users.yml
```

- `wazuh/indexer_stack/wazuh-indexer/indexer_conf/opensearch.yml`

```
plugins.security.ssl.http.pemcert_filepath: /usr/share/wazuh-indexer/certs/node.pem
plugins.security.ssl.http.pemkey_filepath: /usr/share/wazuh-indexer/certs/node-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /usr/share/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /usr/share/wazuh-indexer/certs/node.pem
plugins.security.ssl.transport.pemkey_filepath: /usr/share/wazuh-indexer/certs/node-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /usr/share/wazuh-indexer/certs/root-ca.pem
```

- `wazuh/wazuh_managers/wazuh-master-sts.yaml`

```
image: 'wazuh/wazuh-manager:4.9.2'
```

- `wazuh/wazuh_managers/wazuh-worker-sts.yaml`

```
image: 'wazuh/wazuh-manager:4.9.2'
```

2. Yeni yapılandırmayı uygula

# Yeni Yapılandırmayı Uygula

Son adım yeni yapılandırmayı uygulamaktır:

- EKS kümesi

```
kubectl apply -k envs/eks/
```

- Diğer küme türleri

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

**Output**

```
statefulset.apps "wazuh-manager-master" configured
```

Bu işlem eski pod'u sonlandırırken aynı birime bağlı yeni bir sürümle yeni bir pod yaratacaktır. Pod'lar başlatıldığında güncelleme hazır olacak ve yüklenen yeni Wazuh sürümünü, kümeyi ve birimlerin kullanımıyla korunan değişiklikleri kontrol edebiliriz.

# Dağıtım

Gerekli servisleri ve pod'ları dağıtmak için bu deponun klonunu oluşturun.

```
git clone https://github.com/wazuh/wazuh-kubernetes.git -b v4.9.2 --depth=1
cd wazuh-kubernetes
```

# SSL Sertifikalarını Kurun

Wazuh dizinleyici kümesi için kendi kendine imzalı sertifikaları adresindeki betiği kullanarak üretebilir `wazuh/certs/indexer_cluster/generate_certs.sh` veya kendi betiğinizi sağlayabilirsiniz.

Wazuh panosu kümesi için kendi imzalı sertifikaları adresindeki betiği kullanarak üretebilir `wazuh/certs/dashboard_http/generate_certs.sh` veya kendi betiğinizi sağlayabilirsiniz.

Gerekli sertifikalar secretGenerator aracılığıyla şu dosyaya aktarılır `kustomization.yml`:

```
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
```

# Depolama Sınıfını Ayarlayın (EKS Olmayan Küme İçin İsteğe Bağlı)

Çalıştırdığınız kümenin türüne bağlı olarak, Depolama Sınıfı farklı bir sağlayıcıya sahip olabilir.

Sizinkini çalıştırarak kontrol edebilirsiniz . Şuna benzer bir şey göreceksiniz:`kubectl get sc`

```
kubectl get sc
NAME                      PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
elk-gp2                   microk8s.io/hostpath   Delete          Immediate          false                  67d
microk8s-hostpath (default)   microk8s.io/hostpath   Delete          Immediate          false                  54d
```

Provizyonlayıcı sütunu microk8s.io/hostpath'i gösteriyor, dosyayı düzenlemeli `envs/local-env/storage-class.yaml` ve bu provizyonlayıcıyı ayarlamalısınız.

# Tüm Bildirimleri Kustomize Kullanarak Uygulayın

Manifest'in iki çeşidi vardır: `eks`ve `local-env`. EKS kümesini kullanıyorsanız eks manifest'i kullanılmalıdır, diğer küme türleri için ise local-env manifest'i kullanılmalıdır.

Hangi bildirimi dağıtmak istediğinize bağlı olarak, yamaları düzenleyerek küme için kaynakları ayarlamak mümkündür. Her küme nesnesinin kalıcı birimleri için CPU, bellek ve depolamayı ayarlayabilirsiniz. Bu, bu yamaları kaldırarak `envs/eks/`veya yamaların kendilerini farklı değerlerle değiştirerek geri alınabilir.`envs/local-env/kustomization.yaml`

Özelleştirme dosyasını kullanarak kümeyi tek bir komutla dağıtabiliriz:

- EKS kümesi

  ```
  kubectl apply -k envs/eks/
  ```

- Diğer küme türleri

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

# Dağıtımın Doğrulanması

### Ad alanı

```
kubectl get namespaces | grep wazuh
```

### Output

```
wazuh        Active    12m
```

### Hizmetler

```
kubectl get services -n wazuh
```

### Output

```
NAME              TYPE          CLUSTER-IP       EXTERNAL-IP       PORT(S)                    AGE
indexer           ClusterIP     xxx.yy.zzz.24    <none>            9200/TCP                   12m
dashboard         ClusterIP     xxx.yy.zzz.76    <none>            5601/TCP                   11m
wazuh             LoadBalancer  xxx.yy.zzz.209   internal-a7a8...  1515:32623/TCP,55000:30283/TCP  9m
wazuh-cluster     ClusterIP     None             <none>            1516/TCP                   9m
Wazuh-indexer     ClusterIP     None             <none>            9300/TCP                   12m
wazuh-workers     LoadBalancer  xxx.yy.zzz.26    internal-a7f9...  1514:31593/TCP             9m
```

### Dağıtımlar

```
kubectl get deployments -n wazuh
```

### Output

```
NAME           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
wazuh-dashboard  1       1         1            1           11m
```

### Durum Kümesi

```
kubectl get statefulsets -n wazuh
```

### Output

```
NAME              READY   AGE
wazuh-indexer     3/3     15m
```

```
wazuh-manager-master   1/1     15m
wazuh-manager-worker   2/2     15m
```

**Baklalar**

```
kubectl get pods -n wazuh
```

**Output**

```
NAME                          READY     STATUS    RESTARTS   AGE
wazuh-indexer-0               1/1       Running   0          15m
wazuh-dashboard-f4d9c7944-httsd   1/1       Running   0          14m
wazuh-manager-master-0          1/1       Running   0          12m
wazuh-manager-worker-0-0        1/1       Running   0          11m
wazuh-manager-worker-1-0        1/1       Running   0          11m
```

## Wazuh panosuna erişim

Hizmetler için alan adları oluşturduysanız, önerilen alan adını kullanarak panoya erişebilmelisiniz: `https://wazuh.your-domain.com`. Bulut sağlayıcıları genellikle panoya doğrudan erişim için harici bir IP adresi veya ana bilgisayar adı sağlar. Bu, hizmetleri kontrol ederek görüntülenebilir:

```
kubectl get services -o wide -n wazuh
```

**Output**

```
 NAME              TYPE          CLUSTER-IP       EXTERNAL-IP                PORT(S)                      AGE
SELECTOR
 dashboard         LoadBalancer   xxx.xx.xxx.xxx   xxx.xx.xxx.xxx             80:31831/TCP,443:30974/TCP
15m       app=wazuh-dashboard
```

**İsteğe bağlı** : Harici IP adresine erişilemeyen yerel bir küme dağıtımında şunları kullanabilirsiniz `port-forward`:

```
kubectl -n wazuh port-forward --address <INTERFACE_IP_ADDRESS> service/dashboard 8443:443
```

`<INTERFACE_IP_ADDRESS>`Kubernetes ana bilgisayarının IP adresi nerede ?

Wazuh kontrol paneline şu adresten ulaşabilirsiniz `https://<INTERFACE_IP_ADDRESS>:8443`: .

Varsayılan kimlik bilgileri şunlardır `admin:SecretPassword`: .

# Wazuh Kullanıcılarının Şifresini Değiştirin

Güvenliği artırmak için Wazuh kullanıcılarının varsayılan şifresini değiştirebilirsiniz. İki tür Wazuh kullanıcısı vardır:

- Wazuh dizinleyici kullanıcıları
- Wazuh API kullanıcıları

## Wazuh Dizinleyici Kullanıcıları

Varsayılan `admin` ve `kibanaserver` kullanıcıların şifrelerini değiştirmek için aşağıdakileri yapın.

> Uyarı: Özel kullanıcılarınız varsa, bunları `internal_users.yml` dosyaya ekleyin. Aksi takdirde, bu prosedürü yürütmek onları siler.

## Wazuh Dashboard Oturumunuzu Kapatma

Şifre değiştirme işlemine başlamadan önce Wazuh kontrol paneli oturumunuzdan çıkış yapmanızı öneririz.

Çıkış yapmadığınız takdirde, kalıcı oturum çerezleri kullanıcı şifrelerini değiştirdikten sonra Wazuh'a erişirken hatalara neden olabilir.

## Yeni Bir Hash Ayarlama

1. `wazuh-indexer-0`'de bir Bash kabuğu başlatın.

   ```
   kubectl exec -it wazuh-indexer-0 -n wazuh -- /bin/bash
   ```

2. Yeni parolanızın karmasını oluşturmak için bu komutları çalıştırın. İstendiğinde, yeni parolayı girin ve **Enter'a** basın.

   ```
   wazuh-indexer@wazuh-indexer-0:~$ export JAVA_HOME=/usr/share/wazuh-indexer/jdk
   wazuh-indexer@wazuh-indexer-0:~$ bash /usr/share/wazuh-indexer/plugins/opensearch-security/tools/hash.sh
   ```

3. Oluşturulan hash'i kopyalayın ve Bash kabuğundan çıkın.
4. Dosyayı açın `wazuh/indexer_stack/wazuh-indexer/indexer_conf/internal_users.yml`. Şifresini değiştirdiğiniz kullanıcıya ait bloğu bulun.

5. Karmayı değiştirin.
   - `admin`kullanıcı

     ```
     ...
     admin:
       hash: "$2y$12$K/SpwjtB.wOHJ/Nc6GVRDuc1h0rM1DfvziFRNPtk27P.c4yDr9njO"
       reserved: true
       backend_roles:
       - "admin"
       description: "Demo admin user"

     ...
     ```

   - `kibanaserver`kullanıcı

     ```
     ...
     kibanaserver:
       hash: "$2a$12$4AcgAt3xwOWadA5s5blL6ev39OXDNhmOesEoo33eZtrq2N0YrU3H."
       reserved: true
       description: "Demo kibanaserver user"

     ...
     ```

# Yeni Şifreyi Ayarlama

> Uyarı: Yeni şifrenizde `$`veya `&` karakterlerini kullanmayın . Bu karakterler dağıtım sırasında hatalara neden olabilir.

1. Yeni parolanızı base64 biçiminde kodlayın. Karma değerini korumak için son satır karakteri eklemekten kaçının. Örneğin, seçeneği `-n`komutla `echo`aşağıdaki gibi kullanın.

   ```
   echo -n "NewPassword" | base64
   ```

2. Dizinleyici veya pano sırları yapılandırma dosyasını aşağıdaki gibi düzenleyin. Alanın değerini `password`yeni kodlanmış parolanızla değiştirin.
   - Kullanıcı şifresini değiştirmek için dosyayı `admin`düzenleyin `wazuh/secrets/indexer-cred-secret.yaml`.

     ```
     ...
     apiVersion: v1
     kind: Secret
     metadata:
       name: indexer-cred
     data:
       username: YWRtaW4=              # string "admin" base64 encoded
       password: U2VjcmV0UGFzc3dvcmQ=  # string "SecretPassword" base64 encoded
     ```

```
...
```

- Kullanıcı şifresini değiştirmek için dosyayı `kibanaserver`düzenleyin `wazuh/secrets/dashboard-cred-secret.yaml`.

```
...
apiVersion: v1
kind: Secret
metadata:
    name: dashboard-cred
data:
    username: a2liYW5hc2VydmVy  # string "kibanaserver" base64 encoded
    password: a2liYW5hc2VydmVy  # string "kibanaserver" base64 encoded
...
```

# Değişiklikleri Uygulama

1. Bildirim değişikliklerini uygulayın

   - EKS kümesi

     ```
     kubectl apply -k envs/eks/
     ```

   - Diğer küme türleri

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

2. Bir kez daha bash kabuğunu başlatın `wazuh-indexer-0`.

   ```
   kubectl exec -it wazuh-indexer-0 -n wazuh -- /bin/bash
   ```

3. Aşağıdaki değişkenleri ayarlayın:

   ```
   export INSTALLATION_DIR=/usr/share/wazuh-indexer
   CACERT=$INSTALLATION_DIR/certs/root-ca.pem
   KEY=$INSTALLATION_DIR/certs/admin-key.pem
   CERT=$INSTALLATION_DIR/certs/admin.pem
   export JAVA_HOME=/usr/share/wazuh-indexer/jdk
   ```

4. Wazuh dizinleyicisinin düzgün bir şekilde başlatılmasını bekleyin. Bekleme süresi iki ila beş dakika arasında değişebilir. Kümenin boyutuna, atanan kaynaklara ve ağın hızına bağlıdır. Ardından, `securityadmin.sh`tüm değişiklikleri uygulamak için scripti çalıştırın.

```
bash /usr/share/wazuh-indexer/plugins/opensearch-security/tools/securityadmin.sh -cd /usr/share/wazuh-indexer/opensearch-security/ -nhnv -cacert  $CACERT -cert $CERT -key $KEY -p 9200 -icl -h $NODE_NAME
```

5. Bileşen kimlik bilgilerini güncellemek için tüm Wazuh yönetici bölmelerini silin.

```
kubectl delete -n wazuh pod/wazuh-manager-master-0 pod/wazuh-manager-worker-0 pod/wazuh-manager-worker-1
```

6. Wazuh kontrol panelinde yeni kimlik bilgilerinizle giriş yapın.

## Wazuh API Kullanıcıları

Kullanıcı `wazuh-wui`, varsayılan olarak Wazuh API'sine bağlanacak kullanıcıdır. Parolayı değiştirmek için şu adımları izleyin.

> Not:  Wazuh API kullanıcıları için parola 8 ila 64 karakter uzunluğunda olmalıdır. En az bir büyük harf ve bir küçük harf, bir sayı ve bir sembol içermelidir.

1. Yeni parolanızı base64 biçiminde kodlayın. Karma değerini korumak için son satır karakteri eklemekten kaçının. Örneğin, seçeneği `-n`komutla `echo`aşağıdaki gibi kullanın.

```
echo -n "NewPassword" | base64
```

2. `wazuh/secrets/wazuh-api-cred-secret.yaml` dosyayı düzenleyin ve `password` alanın değerini değiştirin .

```
apiVersion: v1
kind: Secret
metadata:
  name: wazuh-api-cred
  namespace: wazuh
data:
  username: d2F6dWgtd3Vp          # string "wazuh-wui" base64 encoded
  password: UGFzc3dvcmQxMjM0LmE=  # string "MyS3cr37P450r.*-" base64 encoded
```

3. Manifest değişikliklerini uygulayın.

```
kubectl apply -k envs/eks/
```

4. Wazuh panosu ve Wazuh yöneticisi ana bilgisayarı için pod'ları yeniden başlatın.

# Agentlar

Wazuh ajanları ana bilgisayarları izlemek için tasarlanmıştır. Bunları kullanmaya başlamak için:

1. Ajanı yükleyin .
2. Dosyayı değiştirerek aracı kaydedin `/var/ossec/etc/ossec.conf`. "Taşıma protokolünü" TCP olarak değiştirin ve 'yi `MANAGER_IP`1514 numaralı bağlantı noktasına işaret eden hizmetin harici IP adresiyle veya bulut sağlayıcısı tarafından sağlanan ana bilgisayar adıyla değiştirin

Acentelerin kaydedilmesi hakkında daha fazla bilgi edinmek için dokümantasyonun Wazuh acente kaydı bölümüne bakın.

# Temizlemek

Tüm dağıtımların, hizmetlerin ve birimlerin temizlenmesine yönelik adımlar.

1. Tüm kümeyi kaldır

Wazuh yönetici kümesinin dağıtımı, farklı StatefulSet öğelerinin yanı sıra yapılandırma haritaları ve hizmetlerinin kullanımını içerir.

Wazuh kümenizi silmek için bu depo dizininden aşağıdaki komutu çalıştırmanız yeterlidir.

- EKS kümesi

```
kubectl delete -k envs/eks/
```

- Diğer küme türleri

```
kubectl delete -k envs/local-env/
```

Bu, dosyada tanımlanan tüm kaynakları kaldıracaktır `kustomization.yml`.

2. Kalıcı birimleri kaldırın.

```
kubectl get persistentvolume
```

**Output**

```
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
CLAIM                                                 STORAGECLASS           REASON   AGE
pvc-024466da-f7c5-11e8-b9b8-022ada63b4ac   10Gi       RWO            Retain           Released
wazuh/wazuh-manager-worker-wazuh-manager-worker-1-0          gp2-encrypted-retained          6d
pvc-b3226ad3-f7c4-11e8-b9b8-022ada63b4ac   30Gi       RWO            Retain           Bound
wazuh/wazuh-indexer-wazuh-indexer-0                  gp2-encrypted-retained          6d
pvc-fb821971-f7c4-11e8-b9b8-022ada63b4ac   10Gi       RWO            Retain           Released
wazuh/wazuh-manager-master-wazuh-manager-master-0          gp2-encrypted-retained          6d
pvc-ffe7bf66-f7c4-11e8-b9b8-022ada63b4ac   10Gi       RWO            Retain           Released
wazuh/wazuh-manager-worker-wazuh-manager-worker-0-0          gp2-encrypted-retained          6d
```

```
kubectl delete persistentvolume pvc-b3226ad3-f7c4-11e8-b9b8-022ada63b4ac
```

Tüm Wazuh ile ilgili kalıcı birimleri silmek için kubectl delete komutunu tekrarlayın.

> Uyarı:  Gerektiğinde birimleri manuel olarak silmeyi unutmayın.