Docker'da Dağıtım

Bu bölümde Wazuh'un Docker'a kurulum süreci ayrıntılı olarak anlatılmaktadır. Docker, yazılım kapsayıcıları içinde uygulama oluşturmak, sunmak ve çalıştırmak için açık bir platformdur. Docker kapsayıcıları, çalıştırmak için gereken her şeyi içeren yazılımları paketler: kod, çalışma zamanı, sistem araçları, sistem kitaplıkları ve ayarlar. Docker, uygulamaları altyapıdan ayırmayı sağlar. Bu, kapsayıcının çalıştığı ortamdan bağımsız olarak uygulamanın her zaman aynı şekilde çalışmasını garanti eder. Kapsayıcılar bulutta veya şirket içinde çalışır. wazuh/wazuh-managerWazuh'u , wazuh/wazuh-indexerve gibi oluşturduğumuz Docker görüntülerini kullanarak yükleyebilirsiniz . Tüm Wazuh Docker görüntülerini Docker hub'ındawazuh/wazuh-dashboard bulabilirsiniz . Docker kurulumu bölümünde , Docker'ın nasıl kurulacağını görebilirsiniz. Wazuh'un Docker'a nasıl kurulacağını Wazuh Docker dağıtımında bulabilirsiniz . Hizmetlere ve kapsayıcılara nasıl erişeceğinizi, veri birimlerini nasıl yöneteceğinizi ve bir kabuğu nasıl çalıştıracağınızı öğrenmek için Wazuh Docker yardımcı programları bölümünü okuyun. Son olarak, SSS bölümünde bazı sık sorulan soruların yanıtlarını bulabilirsiniz .

- Docker Kurulumu
- Wazuh Docker Dağıtımı
- Wazuh Docker Yardımcı Programları

Docker Kurulumu

Yapmanız gereken ilk şey, Docker ve Docker compose'u çalıştırmak için gereken gereksinimlere sahip bir sistem kurmaktır. Daha sonra, henüz yoksa Docker ve Docker compose'u yükleyin.

> Not: Aşağıda açıklanan tüm komutları çalıştırmak için root kullanıcı ayrıcalıklarına ihtiyacınız var.

Gereksinimler

Konteyner Belleği

Docker ana bilgisayarını en az 6 GB bellekle yapılandırmanızı öneririz. Dağıtım ve kullanıma bağlı olarak, Wazuh dizinleyici bellek tüketimi değişir. Bu nedenle, tam bir yığın dağıtımının düzgün çalışması için önerilen belleği ayırın.

Sunucunuzdaki max_map_count'u Artırın (Linux)

Wazuh indeksleyicisi birçok bellek eşlemeli alan oluşturur. Bu yüzden çekirdeği bir işleme en az 262.144 bellek eşlemeli alan verecek şekilde ayarlamanız gerekir.

1. `max_map_count`Docker hostunuzdaki artışı artırın:

   ```
   sysctl -w vm.max_map_count=262144
   ```

2. `vm.max_map_count`Bu değeri kalıcı olarak ayarlamak için ayarı güncelleyin `/etc/sysctl.conf`. Yeniden başlattıktan sonra doğrulamak için “`sysctl vm.max_map_count`” çalıştırın.

   > Uyarı: Eğer hostunuzda ayarlamazsanız `max_map_count`, Wazuh indeksleyicisi düzgün çalışmayacaktır.

Docker Motoru

Linux/Unix makineleri için Docker, çekirdek sürümü 3.10 veya üzeri olan bir amd64 mimari sistemine ihtiyaç duyar.

1. Bir terminal açın ve çekirdek sürümünüzü görüntülemek ve kontrol etmek için şunu kullanın: `uname -r`

   ```
   uname -r
   ```

   Output

   ```
   3.10.0-229.el7.x86_64
   ```

2. Docker kurulum betiğini çalıştırın:

   Ubuntu/Debian Makinelerinde

   ```
   curl -sSL https://get.docker.com/ | sh
   ```

   CentOS Makinelerinde

   ```
   yum install -y yum-utils
   yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
   yum install docker-ce docker-ce-cli containerd.io docker-compose-plugin
   ```

   Amazon Linux 2 Makinede

   ```
   yum update -y
   yum install docker
   ```

3. Docker servisini başlatın:

   Systemd

   ```
   systemctl start docker
   ```

   SysV Başlatma

   ```
   service docker start
   ```

> Not: Docker'ı kök olmayan bir kullanıcı olarak kullanmak istiyorsanız, kullanıcınızı `docker` aşağıdakine benzer bir komutla gruba eklemelisiniz: . Bunun etkili olması için oturumu kapatıp tekrar oturum açın.`usermod -aG docker your-user`

Docker Oluşturma

Wazuh Docker dağıtımı Docker Compose 1.29 veya üzerini gerektirir. Yüklemek için şu adımları izleyin:

1. Docker Compose ikili dosyasını indirin:

   ```
   curl -L "https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
   ```

2. Yürütme izinlerini verin:

   ```
   chmod +x /usr/local/bin/docker-compose
   ```

3. Her şeyin yolunda olduğundan emin olmak için kurulumu test edin:

   ```
   docker-compose --version
   ```

   Output

   ```
   Docker Compose version v2.12.2
   ```

   > Not: Kurulumdan sonra komut başarısız olursa . Yolunuzdaki herhangi bir dizine veya başka bir dizine `docker-compose`sembolik bir bağlantı oluşturun :`/usr/binln -s /usr/local/bin/docker-compose /usr/bin/docker-compose`

Wazuh Docker Dağıtımı

Kullanım

Wazuh'u tek düğümlü veya çok düğümlü bir yığın olarak dağıtabilirsiniz.

- **Tek düğümlü dağıtım** : Bir Wazuh yöneticisi, dizinleyici ve pano düğümü dağıtır.
- **Çoklu düğüm dağıtımı** : İki Wazuh yönetici düğümü (bir ana ve bir çalışan), üç Wazuh dizinleyici düğümü ve bir Wazuh gösterge paneli düğümü dağıtır.

Her iki dağıtım da kalıcılığı kullanır ve düğümler arasındaki iletişimleri güvence altına almak için sertifikaların yapılandırılmasına izin verir. Çok düğümlü yığın, yüksek kullanılabilirlik içeren tek dağıtımdır.

Tek Düğüm Dağıtımı

1. Wazuh deposunu sisteminize kopyalayın:

   ```
   git clone https://github.com/wazuh/wazuh-docker.git -b v4.9.2
   ```

   Daha sonra aşağıda anlatılan tüm komutları çalıştırmak için `single-node` dizine girin.

2. Yığındaki her düğüm için bir sertifika grubu sağlayarak düğümler arasındaki iletişimi güvence altına alın. Bu sertifikaları sağlamak için iki alternatifiniz var:
   - Her küme düğümü için kendi kendine imzalanmış sertifikalar oluşturun.
     Wazuh sertifika oluşturma aracını kullanarak sertifika oluşturmayı otomatikleştirmek için bir Docker imajı oluşturduk.
     Sisteminiz bir proxy kullanıyorsa, dosyaya aşağıdakileri ekleyin `generate-indexer-certs.yml`. Kullanmıyorsa, bu adımı atlayın:

     ```
     environment:
       - HTTP_PROXY=YOUR_PROXY_ADDRESS_OR_DNS
     ```

     Tamamlanmış bir örnek şu şekildedir:

     ```
     # Wazuh App Copyright (C) 2017 Wazuh Inc. (License GPLv2)
     version: '3'
     ```

```
services:
  generator:
    image: wazuh/wazuh-certs-generator:0.0.2
    hostname: wazuh-certs-generator
    volumes:
      - ./config/wazuh_indexer_ssl_certs/:/certificates/
      - ./config/certs.yml:/config/certs.yml
    environment:
      - HTTP_PROXY=YOUR_PROXY_ADDRESS_OR_DNS
```

İstenilen sertifikaları almak için aşağıdaki komutu çalıştırın:

```
docker-compose -f generate-indexer-certs.yml run --rm generator
```

Bu sertifikaları dizine kaydeder `config/wazuh_indexer_ssl_certs`.

- Her düğüm için kendi sertifikalarınızı sağlayın.
  Eğer kendi sertifikalarınız varsa, bunları dizinde aşağıdaki şekilde sağlayın `config/wazuh_indexer_ssl_certs`:

  Wazuh indexer:

  ```
  config/wazuh_indexer_ssl_certs/root-ca.pem
  config/wazuh_indexer_ssl_certs/wazuh.indexer-key.pem
  config/wazuh_indexer_ssl_certs/wazuh.indexer.pem
  config/wazuh_indexer_ssl_certs/admin.pem
  config/wazuh_indexer_ssl_certs/admin-key.pem
  ```

  Wazuh manager:

  ```
  config/wazuh_indexer_ssl_certs/root-ca-manager.pem
  config/wazuh_indexer_ssl_certs/wazuh.manager.pem
  config/wazuh_indexer_ssl_certs/wazuh.manager-key.pem
  ```

  Wazuh dashboard:

  ```
  config/wazuh_indexer_ssl_certs/wazuh.dashboard.pem
  config/wazuh_indexer_ssl_certs/wazuh.dashboard-key.pem
  config/wazuh_indexer_ssl_certs/root-ca.pem
  ```

3. docker-compose kullanarak Wazuh tek düğümlü dağıtımını başlatın:

   - **Ön Plan** :

     ```
     docker-compose up
     ```

   - **Arka plan** :

     ```
     docker-compose up -d
     ```

Wazuh panosu için varsayılan kullanıcı adı ve parola `admin` ve `SecretPassword`'dir . Ek güvenlik *için, Wazuh dizinleyici yönetici* kullanıcısı için varsayılan parolayı değiştirebilirsiniz .

> Not: Wazuh dizinleyicisinin ne zaman çalıştığını bilmek için Wazuh gösterge paneli kapsayıcısı `curl`Wazuh dizinleyici API'sine birden fazla sorgu çalıştırmak için kullanılır. Wazuh dizinleyicisi başlatılana kadar birkaç günlük `Failed to connect to Wazuh indexer port 9200` mesajı veya " *Wazuh gösterge paneli sunucusu henüz hazır değil* " mesajı görmeyi bekleyebilirsiniz. Ardından kurulum süreci normal şekilde devam eder. Wazuh dizinleyicisinin başlatılması yaklaşık 1 dakika sürer. Varsayılan Wazuh dizinleyici kimlik bilgilerini `docker-compose.yml` dosyada bulabilirsiniz .

Çoklu Düğüm Dağıtımı

1. Wazuh deposunu sisteminize kopyalayın:

   ```
   git clone https://github.com/wazuh/wazuh-docker.git -b v4.9.2
   ```

   `multi-node`Daha sonra aşağıda anlatılan tüm komutları çalıştırmak için dizine girin.

2. Yığındaki her düğüm için bir sertifika grubu sağlayarak düğümler arasındaki iletişimleri güvence altına alın. Bu sertifikaları sağlamak için iki alternatifiniz var:
   - Her küme düğümü için kendi kendine imzalanmış sertifikalar oluşturun.
     Wazuh sertifika oluşturma aracını kullanarak sertifika oluşturmayı otomatikleştirmek için bir Docker imajı oluşturduk.
     Sisteminiz bir proxy kullanıyorsa, dosyaya aşağıdakileri ekleyin `generate-indexer-certs.yml`. Kullanmıyorsa, bu adımı atlayın:

     ```
     environment:
       - HTTP_PROXY=YOUR_PROXY_ADDRESS_OR_DNS
     ```

     Tamamlanmış bir örnek şu şekildedir:

     ```
     # Wazuh App Copyright (C) 2017 Wazuh Inc. (License GPLv2)
     version: '3'

     services:
       generator:
         image: wazuh/wazuh-certs-generator:0.0.2
         hostname: wazuh-certs-generator
         volumes:
           - ./config/wazuh_indexer_ssl_certs/:/certificates/
           - ./config/certs.yml:/config/certs.yml
     ```

```
environment:
  - HTTP_PROXY=YOUR_PROXY_ADDRESS_OR_DNS
```

İstenilen sertifikaları almak için aşağıdaki komutu çalıştırın:

```
docker-compose -f generate-indexer-certs.yml run --rm generator
```

Bu sertifikaları dizine kaydeder `config/wazuh_indexer_ssl_certs`.

- Her düğüm için kendi sertifikalarınızı sağlayın.
  Eğer kendi sertifikalarınız varsa, bunları aşağıdaki şekilde temin edin:

  Wazuh indeksleyicisi :

  ```
  config/wazuh_indexer_ssl_certs/root-ca.pem
  config/wazuh_indexer_ssl_certs/wazuh1.indexer-key.pem
  config/wazuh_indexer_ssl_certs/wazuh1.indexer.pem
  config/wazuh_indexer_ssl_certs/wazuh2.indexer-key.pem
  config/wazuh_indexer_ssl_certs/wazuh2.indexer.pem
  config/wazuh_indexer_ssl_certs/wazuh3.indexer-key.pem
  config/wazuh_indexer_ssl_certs/wazuh3.indexer.pem
  config/wazuh_indexer_ssl_certs/admin.pem
  config/wazuh_indexer_ssl_certs/admin-key.pem
  ```

  Wazuh manager:

  ```
  config/wazuh_indexer_ssl_certs/root-ca-manager.pem
  config/wazuh_indexer_ssl_certs/wazuh.master.pem
  config/wazuh_indexer_ssl_certs/wazuh.master-key.pem
  config/wazuh_indexer_ssl_certs/wazuh.worker.pem
  config/wazuh_indexer_ssl_certs/wazuh.worker-key.pem
  ```

  Wazuh dashboard:

  ```
  config/wazuh_indexer_ssl_certs/wazuh.dashboard.pem
  config/wazuh_indexer_ssl_certs/wazuh.dashboard-key.pem
  config/wazuh_indexer_ssl_certs/root-ca.pem
  ```

3. Wazuh çoklu düğüm dağıtımını şunu kullanarak başlatın `docker-compose`:

   - **Ön Plan** :

     ```
     docker-compose up
     ```

   - **Arka plan** :

     ```
     docker-compose up -d
     ```

   Wazuh panosu için varsayılan kullanıcı adı ve parola ve'dir `admin`. `SecretPassword`Ek güvenlik *için, Wazuh dizinleyici yönetici* kullanıcısı için varsayılan parolayı değiştirebilirsiniz .

> Not: Wazuh dizinleyicisinin ne zaman çalıştığını bilmek için Wazuh gösterge paneli kapsayıcısı `curl`Wazuh dizinleyici API'sine birden fazla sorgu çalıştırmak için kullanılır. Wazuh dizinleyicisi başlatılana kadar birkaç günlük `Failed to connect to Wazuh indexer port 9200` mesajı veya "Wazuh gösterge paneli sunucusu henüz hazır değil" mesajı görmeyi bekleyebilirsiniz. Ardından kurulum süreci normal şekilde devam eder. Wazuh dizinleyicisinin başlatılması yaklaşık 1 dakika sürer. Varsayılan Wazuh dizinleyici kimlik bilgilerini `docker-compose.yml` dosyada bulabilirsiniz .

Docker Görüntülerini Yerel Olarak Oluşturun

Wazuh yöneticisini, dizinleyiciyi ve gösterge paneli görüntülerini yerel olarak değiştirebilir ve oluşturabilirsiniz.

1. Wazuh deposunu sisteminize kopyalayın:

   ```
   git clone https://github.com/wazuh/wazuh-docker.git -b v4.9.2
   ```

2. 4.3.4'e kadar olan sürümler için dizine girin `build-docker-images`ve Wazuh yöneticisini, dizinleyiciyi ve gösterge paneli görüntülerini oluşturun:

   ```
   docker-compose build
   ```

   4.3.5 ve üzeri sürümler için görüntü oluşturma betiğini çalıştırın:

   ```
   build-docker-images/build-images.sh
   ```

Wazuh Kullanıcılarının Şifresini Değiştirin

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

- Wazuh dizinleyici kullanıcıları
- Wazuh API kullanıcıları

Bu Wazuh kullanıcılarının parolasını değiştirmek için aşağıdaki adımları uygulayın. Wazuh on Docker dağıtımınıza bağlı olarak komutları `single-node/`veya `multi-node/` dizininizden çalıştırmalısınız .

Wazuh Indexer Kullanıcıları

Varsayılan `admin` ve `kibanaserver` kullanıcıların şifresini değiştirmek için aşağıdakileri yapın. Bir seferde yalnızca birini değiştirebilirsiniz.

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

Wazuh Panosu 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 Karma Ayarlama

1. Çalışıyorsa dağıtım yığınını durdurun:

   ```
   docker-compose down
   ```

2. Yeni parolanızın karmasını oluşturmak için bu komutu çalıştırın. Konteyner başlatıldığında, yeni parolayı girin ve **Enter'a** basın.

   ```
   docker run --rm -ti wazuh/wazuh-indexer:4.9.2 bash /usr/share/wazuh-indexer/plugins/opensearch-security/tools/hash.sh
   ```

3. Oluşturulan hash'i kopyalayın.
4. Dosyayı açın `config/wazuh_indexer/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. Dosyayı açın `docker-compose.yml`. Eski parolanın tüm oluşumlarını yenisiyle değiştirin. Örneğin, tek düğümlü bir dağıtım için:

   - `admin`kullanıcı

```
...
services:
  wazuh.manager:
    ...
    environment:
      - INDEXER_URL=https://wazuh.indexer:9200
      - INDEXER_USERNAME=admin
      - INDEXER_PASSWORD=SecretPassword
      - FILEBEAT_SSL_VERIFICATION_MODE=full
      - SSL_CERTIFICATE_AUTHORITIES=/etc/ssl/root-ca.pem
      - SSL_CERTIFICATE=/etc/ssl/filebeat.pem
      - SSL_KEY=/etc/ssl/filebeat.key
      - API_USERNAME=wazuh-wui
      - API_PASSWORD=MyS3cr37P450r.*-
  ...
  wazuh.indexer:
    ...
    environment:
      - "OPENSEARCH_JAVA_OPTS=-Xms1024m -Xmx1024m"
  ...
  wazuh.dashboard:
    ...
    environment:
      - INDEXER_USERNAME=admin
      - INDEXER_PASSWORD=SecretPassword
      - WAZUH_API_URL=https://wazuh.manager
```

```
      - DASHBOARD_USERNAME=kibanaserver
      - DASHBOARD_PASSWORD=kibanaserver
      - API_USERNAME=wazuh-wui
      - API_PASSWORD=MyS3cr37P450r.*-
  ...
```

- `kibanaserver` kullanıcı

```
...
services:
  wazuh.dashboard:
    ...
    environment:
      - INDEXER_USERNAME=admin
      - INDEXER_PASSWORD=SecretPassword
      - WAZUH_API_URL=https://wazuh.manager
      - DASHBOARD_USERNAME=kibanaserver
      - DASHBOARD_PASSWORD=kibanaserver
      - API_USERNAME=wazuh-wui
      - API_PASSWORD=MyS3cr37P450r.*-
  ...
```

Değişiklikleri Uygulama

1. Dağıtım yığınını başlatın.

   ```
   docker-compose up -d
   ```

2. Çalıştırın ve ilk Wazuh dizinleyici kabının adını not edin. Örneğin, , veya .`docker ps` `single-node-wazuh.indexer-1` `multi-node-wazuh1.indexer-1`
3. Konteynere girmek için koşun . Örneğin:`docker exec -it <WAZUH_INDEXER_CONTAINER_NAME> bash`

   ```
   docker exec -it single-node-wazuh.indexer-1 bash
   ```

4. 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
   ```

5. 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 betiği çalıştırın.

Tek Düğümlü Küme

```
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
```

Çok Düğümlü Küme

```
HOST=$(grep node.name $INSTALLATION_DIR/opensearch.yml | awk '{printf $2}')
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 $HOST
```

6. Wazuh dizinleyici konteynerinden çıkın ve Wazuh panosunda yeni kimlik bilgilerinizle oturum açın.

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. Dosyayı açın `config/wazuh_dashboard/wazuh.yml`ve parametrenin değerini değiştirin `password`.

```
...
hosts:
  - 1513629884013:
      url: "https://wazuh.manager"
      port: 55000
      username: wazuh-wui
      password: "MyS3cr37P450r.*-"
      run_as: false
...
```

2. Dosyayı açın `docker-compose.yml`. Eski şifrenin tüm tekrarlarını yenisiyle değiştirin.

```
...
services:
  wazuh.manager:
    ...
    environment:
      - INDEXER_URL=https://wazuh.indexer:9200
      - INDEXER_USERNAME=admin
      - INDEXER_PASSWORD=SecretPassword
      - FILEBEAT_SSL_VERIFICATION_MODE=full
      - SSL_CERTIFICATE_AUTHORITIES=/etc/ssl/root-ca.pem
      - SSL_CERTIFICATE=/etc/ssl/filebeat.pem
      - SSL_KEY=/etc/ssl/filebeat.key
      - API_USERNAME=wazuh-wui
      - API_PASSWORD=MyS3cr37P450r.*-
  ...
  wazuh.dashboard:
    ...
    environment:
      - INDEXER_USERNAME=admin
      - INDEXER_PASSWORD=SecretPassword
      - WAZUH_API_URL=https://wazuh.manager
      - DASHBOARD_USERNAME=kibanaserver
      - DASHBOARD_PASSWORD=kibanaserver
      - API_USERNAME=wazuh-wui
      - API_PASSWORD=MyS3cr37P450r.*-
  ...
```

3. Wazuh konteynerlerini yeniden oluşturun:

```
docker-compose down
docker-compose up -d
```

Açıkta Kalan Portlar

Varsayılan olarak, yığın aşağıdaki portları açığa çıkarır:

1514	Wazuh TCP
1515	Wazuh TCP
514	Wazuh UDP
55000	Wazuh API
9200	Wazuh dizinleyici HTTPS
443	Wazuh panosu HTTPS

> Not: Docker yapılandırmayı dinamik olarak yeniden yüklemez. Bir bileşenin yapılandırmasını değiştirdikten sonra yığını yeniden başlatmanız gerekir.

Wazuh Docker Yardımcı Programları

Wazuh-Docker konteynerlerini kurduktan sonra, Wazuh kurulumunuzdan en iyi şekilde yararlanmak için yapabileceğiniz birkaç görev vardır.

Hizmetlere ve Konteynerlere Erişim

1. Docker ana bilgisayar IP adresini kullanarak Wazuh panosuna erişin. Örneğin, `https://localhost`Docker ana bilgisayarındaysanız.

   > Not: Kendinden imzalı bir sertifika kullanmanız durumunda tarayıcınız sertifikanın gerçekliğini doğrulayamadığına dair bir uyarı verecektir.

2. Standart kayıt sürecini izleyerek ve Docker ana bilgisayar adresini yönetici adresi olarak kullanarak aracıları kaydedin. Daha fazla bilgi için Wazuh aracı kayıt belgelerine bakın.
3. Wazuh dosyasının bulunduğu dizindeki kapsayıcıları listeleyin `docker-compose.yml`:

   ```
   docker-compose ps
   ```

   Output

   ```
   NAME                        COMMAND                  SERVICE           STATUS          PORTS
   single-node-wazuh.dashboard-1   "/entrypoint.sh"         wazuh.dashboard   running         443/tcp, 0.0.0.0:443->5601/tcp
   single-node-wazuh.indexer-1     "/entrypoint.sh open…"   wazuh.indexer     running         0.0.0.0:9200->9200/tcp
   single-node-wazuh.manager-1     "/init"                  wazuh.manager     running         0.0.0.0:1514-1515->1514-1515/tcp, 0.0.0.0:514->514/udp, 0.0.0.0:55000->55000/tcp, 1516/tcp
   ```

4. `docker-compose.yml`Her bir konteynerin komut satırına erişmek için dosyanın bulunduğu dizinden aşağıdaki komutu çalıştırın:

   ```
   docker-compose exec <SERVICE> bash
   ```

Wazuh Servis Veri Hacimleri

Wazuh yapılandırma ve günlük dosyalarının kapsayıcılarının dışında var olmasını ayarlayabilirsiniz. Bu, dosyaların kapsayıcıları kaldırdıktan sonra da kalıcı olmasını sağlar ve kapsayıcılarınıza özel yapılandırma dosyaları sağlayabilirsiniz.

Bir Wazuh konteynerinde kalıcılığı garantilemek için birden fazla birime ihtiyacınız var. Aşağıda `docker-compose.yml`kalıcı birimlere sahip bir örnek verilmiştir:

```
services:
  wazuh:
    . . .
    volumes:
      - wazuh_api_configuration:/var/ossec/api/configuration


volumes:
  wazuh_api_configuration:
```

Kalıcı birimleri şu şekilde listeleyebilirsiniz : `docker volume ls`

Output

```
DRIVER              VOLUME NAME
local               single-node_wazuh_api_configuration
```

Wazuh Indexer ve Dashboard İçin Depolama Hacmi

Wazuh dizinleyici verilerinin depolanması için bir birim eklemek de mümkündür. Varsayılan olarak, tek düğümlü ve çok düğümlü dağıtımlar zaten yapılandırılmış birimlere sahiptir. Tek düğümlü bir wazuh dizinleyici biriminin bir örneği aşağıda gösterilmiştir `docker-compose.yml`:

```
wazuh.indexer:
    . . .
     volumes:
       - wazuh-indexer-data:/var/lib/wazuh-indexer


    . . .
```

```
volumes:
  wazuh-indexer-data
```

Özel Komutlar ve Scriptler

Wazuh yönetici kabında komutları çalıştırmak için bir kabuk çalıştırabilirsiniz:

```
docker exec -it single-node-wazuh.manager-1 bash
```

Bu kabukta yapılan her değişiklik, veri birimleri doğru şekilde yapılandırıldığı sürece kalıcı olur.