Wazuh Indexer Ayarı

Bu kılavuz, Wazuh dizinleyici performansını optimize etmek için ayarların nasıl değiştirileceğini gösterir. Wazuh dizinleyici parolasını değiştirmek için Parola yönetimi bölümüne bakın.

- Bellek kilitleme
- Parçalar ve kopyalar
- Parça tahsis farkındalığını veya zorunlu farkındalığı yapılandırın
- Bir kümedeki her düğüm için düğüm niteliklerini ayarlayın

Bellek Kilitleme

Sistem belleği takas ederken, Wazuh dizinleyicisi beklendiği gibi çalışmayabilir. Bu nedenle, Wazuh dizinleyici düğümünün sağlığı için Java Sanal Makinesi'nin (JVM) hiçbir zaman diske takas edilmemesi önemlidir. Herhangi bir Wazuh dizinleyici belleğinin takas edilmesini önlemek için, Wazuh dizinleyicisini işlem adres alanını RAM'e kilitlemek üzere aşağıdaki gibi yapılandırın.

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

1. `/etc/wazuh-indexer/opensearch.yml`Bellek kilitlemeyi etkinleştirmek için Wazuh indeksleyicisindeki yapılandırma dosyasına aşağıdaki satırı ekleyin :

   ```
   bootstrap.memory_lock: true
   ```

2. Sistem kaynaklarının sınırını değiştirin. Sistem ayarlarını yapılandırmak Wazuh dizinleyici kurulumunun işletim sistemine bağlıdır.

   Systemd

   1. Sistem sınırlarını belirten dosya için yeni bir dizin oluşturun:

      ```
      mkdir -p /etc/systemd/system/wazuh-indexer.service.d/
      ```

   2. Yeni sistem sınırı eklenerek yeni oluşturulan dizinde `wazuh-indexer.conf` dosyayı oluşturmak için aşağıdaki komutu çalıştırın :

      ```
      # cat > /etc/systemd/system/wazuh-indexer.service.d/wazuh-indexer.conf << EOF
      [Service]
      ```

```
LimitMEMLOCK=infinity
EOF
```

SysV Başlatma

1. Sistem sınırlarını belirten dosya için yeni bir dizin oluşturun:

```
mkdir -p /etc/init.d/wazuh-indexer.service.d/
```

2. Yeni sistem sınırı eklenerek yeni oluşturulan dizinde `wazuh-indexer.conf` dosyayı oluşturmak için aşağıdaki komutu çalıştırın :

```
# cat > /etc/init.d/wazuh-indexer.service.d/wazuh-indexer.conf << EOF
[Service]
LimitMEMLOCK=infinity
EOF
```

3. Dosyayı düzenleyin `/etc/wazuh-indexer/jvm.options`ve JVM bayraklarını değiştirin. Bellek kullanımını sınırlamak için bir Wazuh dizinleyici yığın boyutu değeri ayarlayın. JVM yığın sınırları, `OutOfMemory`Wazuh dizinleyicisi önceki adımdaki yapılandırma nedeniyle kullanılabilir olandan daha fazla bellek ayırmaya çalışırsa istisnayı önler. Önerilen değer sistem RAM'inin yarısıdır. Örneğin, 8 GB RAM'li bir sistem için boyutu aşağıdaki gibi ayarlayın.

```
-Xms4g
-Xmx4g
```

Toplam yığın alanı:
- `-Xms4g`- Başlangıç boyutu 4Gb RAM olarak ayarlandı.
- `-Xmx4g`- Maksimum boyut 4Gb RAM'dir.

> Uyarı: Çalışma zamanında JVM yığın yeniden boyutlandırması nedeniyle performans düşüşünü önlemek için, minimum (Xms) ve maksimum (Xmx) boyut değerlerinin aynı olması gerekir.

4. Wazuh dizinleyici hizmetini yeniden başlatın:

```
systemctl daemon-reload
systemctl restart wazuh-indexer
```

5. Ayarın başarıyla değiştirildiğini doğrulamak için aşağıdaki komutu çalıştırarak `mlockall` değerin şu şekilde ayarlandığını kontrol edin `true`:

```
curl -k -u <INDEXER_USERNAME>:<INDEXER_PASSWORD>
"https://<INDEXER_IP_ADDRESS>:9200/_nodes?filter_path=**.mlockall&pretty"
```

Output

```
{
  "nodes" : {
    "sRuGbIQRRfC54wzwIHjJWQ" : {
      "process" : {
        "mlockall" : true
      }
    }
  }
}
```

Çıktı ise `false`, istek başarısız olmuş ve dosyada aşağıdaki satır görünür `/var/log/wazuh-indexer/wazuh-indexer.log`:

```
Unable to lock JVM Memory
```

Parçalar ve Kopyalar

Wazuh dizinleyicisi, bir dizini shard adı verilen birden fazla parçaya bölme olanağı sunar. Her shard, Wazuh dizinleyici kümesindeki herhangi bir düğümde barındırılabilen tamamen işlevsel ve bağımsız bir "indekstir". Bölme iki ana nedenden dolayı önemlidir:

- Yatay ölçekleme.
- Parçalar arası dağıtım ve paralelleştirme işlemleri, performans ve verimi artırır.

Ayrıca, Wazuh dizinleyicisi kullanıcıların dizin parçacıklarının bir veya daha fazla kopyasını, kısaca replikalar veya replikalar olarak adlandırılan şekilde oluşturmasına olanak tanır. Replikasyon iki nedenden dolayı önemlidir:

- Bir parçanın veya düğümün arızalanması durumunda yüksek erişilebilirlik sağlar.
- Aramalar tüm replikalarda paralel olarak yürütülebildiğinden arama hacminin ve veriminin ölçeklenmesine olanak tanır.

Bir Index İçin Parça Sayısı

İlk dizini oluşturmadan önce, kaç tane parçaya ihtiyaç duyulacağını dikkatlice düşünün. Parça sayısını yeniden dizinlemeden değiştirmek mümkün değildir.

Optimum performans için gereken parça sayısı, Wazuh dizinleyici kümesindeki düğüm sayısına bağlıdır. Genel bir kural olarak, parça sayısı düğüm sayısıyla aynı olmalıdır. Örneğin, üç düğümü olan bir kümenin üç parçası olmalı, yalnızca bir düğümü olan bir kümenin ise yalnızca bir parçaya ihtiyacı olacaktır.

Bir Index İçin Kopya Sayısı

Kopyaların sayısı, dizinler için kullanılabilir depolama alanına bağlıdır. İşte üç düğüm ve üç parçadan oluşan bir Wazuh dizinleyici kümesinin nasıl kurulabileceğine dair bir örnek.

- **Kopya yok** : Her düğümün bir parçası vardır. Bir düğüm çökerse, yalnızca iki parçadan oluşan eksik bir dizin kullanılabilir.
- **Bir kopya** : Her düğümün bir parçası ve bir kopyası vardır. Bir düğüm çökerse, tam bir dizin hala kullanılabilir.
- **İki replika** : Her düğümün bir parça ve iki replika ile tam dizini vardır. Bu kurulumla, iki düğüm çökse bile küme çalışmaya devam eder. Bu en iyi çözüm gibi görünse de depolama gereksinimlerini artırır.

Aşağıdaki görüntü, her biri birincil parça ve iki kopya parça içeren üç düğümden oluşan bir Wazuh dizinleyici kümesini göstermektedir.



Parçalar ve kopyalar diyagramıyla Wazuh dizinleyici kümesi

Parça Sayısını Ayarlama

> Uyarı: Parça ve replika sayısı, dizin oluşturma sırasında dizin başına tanımlanır. Dizin oluşturulduktan sonra, replika sayısı dinamik olarak değiştirilebilse de, parça sayısı yeniden dizinleme yapılmadan değiştirilemez .

Wazuh dizinleyici düğümünün varsayılan kurulumu her dizini üç birincil parça ve hiçbir kopya olmadan oluşturur. Wazuh API'sini kullanarak yeni bir şablon yükleyerek birincil parça ve kopya sayısını değiştirebilirsiniz.

Aşağıdaki örnekte, tek düğümlü bir Wazuh dizinleyicisi için parçacık sayısını 1 olarak ayarladık. Wazuh API'sini kullanarak kimlik doğrulaması yapmasına izin verilen Wazuh dizinleyici düğümünde veya herhangi bir merkezi bileşende aşağıdaki adımları uygulayın.

1. Wazuh indeksleyici şablonunu indirin:

```
curl https://raw.githubusercontent.com/wazuh/wazuh/v4.9.2/extensions/elasticsearch/7.x/wazuh-template.json -o w-indexer-template.json
```

2. `index.number_of_shards` öğesini `1` olarak ayarlamak için `w-indexer-template.json` dosyasını düzenleyin. Filebeat'in mevcut şablonun üzerine yazmasını önlemek için sırayı 1 olarak ayarlayın. Aynı sırada birden fazla eşleşen şablon, deterministik olmayan bir birleştirme sırasına neden olur.

```
{
  "order": 1,
  "index_patterns": [
    "wazuh-alerts-4.x-*",
    "wazuh-archives-4.x-*"
  ],
  "settings": {
    "index.refresh_interval": "5s",
    "index.number_of_shards": "1",
    "index.number_of_replicas": "0",
    "index.auto_expand_replicas": "0-1",
    "index.mapping.total_fields.limit": 10000,
    ...
```

3. Yeni ayarları yükleyin.

```
curl -X PUT "https://<INDEXER_IP_ADDRESS>:9200/_template/wazuh-custom" -H 'Content-Type: application/json' -d @w-indexer-template.json -k -u <INDEXER_USERNAME>:<INDEXER_PASSWORD>
```

Output

```
{"acknowledged":true}
```

4. Yapılandırmanın başarıyla güncellendiğini onaylayın.

```
curl "https://<INDEXER_IP_ADDRESS>:9200/_template/wazuh-custom?pretty&filter_path=wazuh-custom.settings" -k -u <INDEXER_USERNAME>:<INDEXER_PASSWORD>
```

Output

```
{
  "wazuh-custom" : {
    "settings" : {
      "index" : {
        "mapping" : {
          "total_fields" : {
            "limit" : "10000"
```

```
      }
    },
    "refresh_interval" : "5s",
    "number_of_shards" : "1",
    "auto_expand_replicas" : "0-1",
    "number_of_replicas" : "0",
    ...
```

Eğer indeks daha önceden oluşturulmuşsa yeniden indekslenmesi gerekir .

Kopyaların Sayısını Ayarlama

Kopya sayısı, Wazuh dizinleyici API'si kullanılarak dinamik olarak değiştirilebilir. Tek düğümlü bir kümede, kopya sayısı sıfıra ayarlanmalıdır. Bu, Wazuh dizinleyici düğümünde veya Wazuh API'si kullanılarak kimlik doğrulaması yapılmasına izin verilen herhangi bir merkezi bileşende aşağıdaki komutu çalıştırarak gerçekleştirilir:

```
curl -k -u "<INDEXER_USERNAME>:<INDEXER_PASSWORD>" -XPUT "https://<INDEXER_IP_ADDRESS>:9200/wazuh
{
  "settings": {
    "index": {
      "number_of_replicas": 0
    }
  }
}'
```

Parça Tahsis Farkındalığını veya Zorunlu Farkındalığı Yapılandırın

Bu, Wazuh indeksleyici düğümlerinin coğrafi olarak dağınık bölgelere yayıldığı durumlarda en çok uygulanabilir.

Farkındalığı yapılandırmak için, `/etc/wazuh-indexer/opensearch.yml`farklı bölgeler için Wazuh dizinleyici düğümlerindeki dosyaya bölge niteliklerini ekleyin.

`/etc/wazuh-indexer/opensearch.yml`Örneğin: A ve B bölgesi adında iki bölgeniz var. Aşağıdaki yapılandırmayı sırasıyla A ve B bölgesindeki her Wazuh dizinleyici düğümüne dosyaya ekleyeceksiniz :

```
node.attr.zone: zoneA
```

```
node.attr.zone: zoneB
```

Tahsis farkındalığı, A ve B bölgesindeki Wazuh dizinleyici düğümlerindeki depolama %50'den az kullanılıyorsa en iyi şekilde kullanılır. Bu, bölgedeki replikaları tahsis etmek için yeterli depolama kapasitesi sağlar.

Hem A hem de B bölgesindeki Wazuh dizinleyici düğümlerinin tüm birincil ve kopya parçacıklarını depolamak için yeterli kapasitesi yoksa, zorunlu farkındalık bir seçenektir. Bu, bir bölge arızası olması durumunda Wazuh dizinleyicisinin kalan bölgenizi aşırı yüklememesini ve kümenizin depolama yetersizliği nedeniyle kilitlenmesini önler.

Tahsis farkındalığı veya zorunlu farkındalığı seçmek, birincil ve kopya parçalarınızı dengelemek için her bölgede ne kadar alanınız olduğuna bağlıdır.

Parça Tahsisi Farkındalığı

Parça tahsisi farkındalığı, birincil ve replika parçaları birden fazla bölgeye yaymaya çalışır. Bir replika parçayı birincil bölgesinden farklı bir bölgeye tahsis etmek için kullanılır.

Bir bölge içinde düğüm arızası durumunda, replika parçalarınızın kalan bölgeleriniz arasında dağıtıldığından emin olabilirsiniz. Bu, hata toleransını artırarak verilerinizi bölge arızalarına ve bireysel düğüm arızalarına karşı korur.

Parça tahsis farkındalığını yapılandırmak için küme ayarlarını güncelleyin:

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.awareness.attributes": "zone"
  }
}
```

`persistent`ya da ayarlarını kullanabilirsiniz `transient`. Ayarı kullanmanızı öneririz `persistent`çünkü küme yeniden başlatma sırasında kalıcıdır. `transient`Ayar küme yeniden başlatma sırasında kalıcı değildir.

> Not: Yalnızca bir bölge mevcutsa (örneğin bölge arızalarından sonra), Wazuh dizinleyicisi çoğaltma parçalarını yalnızca kalan bölgeye tahsis eder.

Zorla Farkındalık

Zorunlu farkındalığın kullanılması, birincil ve kopya parçaların asla aynı bölgeye tahsis edilmediği anlamına gelir.

Zorunlu farkındalığı yapılandırmak için bölge nitelikleriniz için tüm olası değerleri belirtin:

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.awareness.attributes": "zone",
    "cluster.routing.allocation.awareness.force.zone.values":["zoneA", "zoneB"]
  }
}
```

Başka bölgeler varsa, diğer bölgeleri `cluster.routing.allocation.awareness.force.zone.values` alanına ekleyin .

> Uyarı: Bir düğüm başarısız olursa, zorunlu farkındalık replikaları aynı bölgedeki başka bir düğüme tahsis etmez. Bunun yerine, küme sarı bir duruma girer ve yalnızca diğer bölgedeki(bölgelerdeki) düğümler çevrimiçi olduğunda replikaları tahsis eder.

Tahsis Filtreleme

Bu, bir düğümün parça tahsisinden hariç tutulmasına olanak tanır. Yaygın bir kullanım durumu, bir bölge içindeki bir düğümü devre dışı bırakmak istediğiniz zamandır.

Bir düğümü devre dışı bırakmadan önce parçaları taşımak için, düğümü IP adresini kullanarak hariç tutan bir filtre oluşturun. Bu, kapatılmadan önce o düğüme tahsis edilen tüm parçaları taşıyacaktır. Ayrıca, `*`bir IP aralığında devre dışı bırakılacak birden fazla düğümün olduğu bir durumda joker karakter kullanabilirsiniz.

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.exclude._ip": "192.168.0.*"
  }
}
```

Bir Kümedeki Her Düğüm İçin Düğüm Niteliklerini Ayarlayın

Varsayılan olarak, her Wazuh dizinleyici düğümü bir ana uygun, veri, alım ve koordinasyon düğümüdür. Düğüm sayısına karar vermek, düğüm türlerini atamak ve her düğüm türü için donanımı seçmek kullanım durumunuza bağlıdır.

Küme Yöneticisi Düğümleri

Küme yöneticisi düğümleri, düğümlere parça ekleme, kaldırma ve tahsis etme, ayrıca dizin ve alan oluşturma ve silme dahil olmak üzere küme genelindeki tüm yapılandırmaları ve değişiklikleri yönetir.

Dağıtılmış bir fikir birliği tekniği, küme yöneticisi uygun düğümleri arasından tek bir küme yöneticisi düğümü seçmek için kullanılır. Bu küme yöneticisi düğümü, mevcut düğüm başarısız olursa yeniden seçilir.

Varsayılan olarak zaten yapılmış olsa da, bir Wazuh dizinleyici düğümünün küme yöneticisi düğümü olduğunu belirtebilirsiniz.

`cluster_manager`Aşağıdaki yapılandırmayı dosyaya ekleyerek bir Wazuh dizinleyici düğümü rolü ayarlayın `/etc/wazuh-indexer/opensearch.yml`:

```
node.roles: [ cluster_manager ]
```

Veri Düğümleri

Veri düğümü, verileri depolamak ve aramaktan sorumludur. Yerel parçalarda tüm veriyle ilgili işlemleri (indeksleme, arama, toplama) gerçekleştirir. Bunlar Wazuh dizinleyici kümenizin çalışan düğümleridir ve diğer tüm düğüm türlerinden daha fazla disk alanına ihtiyaç duyarlar.

Aşağıdaki yapılandırmayı dosyaya ekleyerek bir Wazuh dizinleyici düğümü rolünü veri düğümü olarak ayarlayın `/etc/wazuh-indexer/opensearch.yml`:

```
node.roles: [ data, ingest ]
```

Veri düğümleri eklerken bunları bölgeler arasında dengeli tutmak önemlidir. Örneğin, üç bölgeniz varsa, her bölge için bir veri düğümü ekleyin. Depolama ve RAM ağırlıklı düğümler kullanmanızı öneririz.

Koordinasyon Düğümleri

Koordinasyon düğümü, istemci isteklerini veri düğümlerindeki parçalara devreder, sonuçları toplar ve tek bir nihai sonuçta birleştirir ve bunu Wazuh panosuna geri gönderir.

Her düğüm varsayılan olarak bir koordinasyon düğümüdür, ancak bir düğümü özel bir koordinasyon düğümü yapmak için `node.roles`boş bir liste ayarlayın:

```
node.roles: []
```

Revision #8
Created 31 December 2024 18:03:10 by Ayşegül Sarıkaya
Updated 31 December 2024 21:13:56 by Ayşegül Sarıkaya