

Prometheus ve Grafana ile Metrikleri Yapılandırma ve Görselleştirme

Tahmini Süre: 30 dakika

Prometheus ve Grafana ile Metrikleri Yapılandırma ve Görselleştirme laboratuvarına hoş geldiniz. Bu laboratuvar, node exporter'lar ile simüle edilmiş örnek sunucuları izlemek için Prometheus'u kullanma konusunda bilgi sahibi olmanızı sağlayacak. Node exporter'lar, metrikleri /metrics uç noktası aracılığıyla sunan sunucu uygulamalarını simüle eder. Prometheus'u, hedef node_exporter uygulamasını metrik uç noktalarını tarayarak izlemek için kullanacaksınız ve ayrıca Grafana kullanıcı arayüzüne anlayacaksınız.

Öğrenme Hedefleri

Bu alıştırmayı tamamladıktan sonra, aşağıdaki görevleri yerine getirebileceksiniz:

- Hedefle ilgili metrikleri almak için sorgular oluşturun
- Hedeflerin durumunu bulun
- Hedefler hakkında bazı bilgileri bulun ve bunları grafiklerle görselleştirin
- İzlenen hedeflerin metriklerini ve durumunu görselleştirmek için Prometheus'ta grafikler oluşturun ve özelleştirin.
- Prometheus'tan veri almak için Grafana'yı yapılandırın ve başlatın.

Ön Koşullar

Ön Koşullar

Bu laboratuvar, izleyebileceğiniz sunucular gibi davranışacak özel Node Exporters'ı çalıştırma için Docker kullanır. Ön koşul olarak, Docker Hub'dan bitnami/node-exporter imajını çekeceksiniz. Bu imajı, üç adet node exporter örneği oluşturmak için kullanacaksınız.

Göreviniz

1. Yeni bir terminal açarak başlayalım. Üst menüden **Terminal**'e gidin ve yeni bir terminal penceresi açmak için **Yeni Terminal'i seçin**.
2. Aşağıdaki docker pull komutunu kullanarak, 3 sunucunun izlenmesini simüle etmek için kullanacağınız bitnami/node-exporter imajını Docker Hub'dan indirin.

```
docker pull bitnami/node-exporter:latest
```

Artık laboratuvara başlamak için hazırınız.

Adım 1: İlk node exporter'ı başlat

Artık izleyeceğiniz bazı sunucu düğümlerine ihtiyacınız var. Sırasıyla 9101, 9102 ve 9103 portlarında dinleyen üç node exporter başlatacaksınız. Bu adımda sadece ilkini başlatacak ve doğru çalıştığını emin olacaksınız.

Göreviniz

1. Aşağıdaki komutu çalıştırarak tüm docker örneklerini çalıştıracağımız bir docker ağı oluşturun.

```
docker network create monitor
```

2. Aşağıdaki docker run komutunu çalıştırarak 9101 portunda dinleyen bir node exporter örneğini başlatın.

```
docker run -d -p 9101:9100 --name node-exporter1 --network monitor bitnami/node-exporter:latest
```

Bu, node-exporter'ın node_exporter1 adlı bir örneğini başlatacaktır. Çıktı şöyle görünmelidir:

3. Ardından, örneğin çalışıp çalışmadığını kontrol etmek için **Skills Network Toolbox**'a tıklayın ve **Diğerleri** altında **Uygulamayı Başlat** seçeneğini seçin, port numarasını **9101** girin ve URL'yi başlat butonuna tıklayın. Ya da aşağıdaki **[Uygulamayı Başlat]** butonuna basın:

Uygulamayı Başlat

4. Node exporter sayfası aşağıdaki gibi açılacak ve **Metrics** için bir hiper bağlantı içerecektir. Bu, Prometheus örneğinin izleyeceği metriklerdir.
5. Metriklere bir göz atmak için **Metrics** bağlantısına tıklayın.
6. Yukarıdaki görüntüde kırmızı ile işaretlenmiş URL'yi kopyalayın ve bir not defterine kaydedin. Bu URL'yi prometheus'u node exporter'ı izlemek için yapılandırmak amacıyla kullanacaksınız.

Adım 2: İki tane daha node exporter başlatın

Artık bir node exporter çalıştığını göre, toplamda Prometheus'un izleyebileceği üç düğüm olması için iki tane daha başlatalım. Bunu, ilk node exporter'ı başlattığınız gibi tam olarak aynı şekilde yapacaksınız, tek fark port numaralarını sırasıyla 9102 ve 9103 olarak değiştireceksiniz.

Göreviniz

1. Terminalde, iki tane daha node exporter örneğini başlatmak için aşağıdaki komutları çalıştırın.

```
docker run -d -p 9102:9100 --name node-exporter2 --network monitor bitnami/node-exporter:latest
```

ve

```
docker run -d -p 9103:9100 --name node-exporter3 --network monitor bitnami/node-exporter:latest
```

2. Şimdi, tüm node exporter örneklerinin çalışıp çalışmadığını kontrol etmek için docker ps komutunu kullanarak node-exporter için grep yapın.

```
docker ps | grep node-exporter
```

Sonuçlar

Her şey doğru bir şekilde başladıysa, docker komutlarından aşağıdaki gibi bir çıktı göremelisiniz:

Artık Prometheus'u kurmaya ve başlatmaya hazırlısanız.

Adım 3: Prometheus'u İndir, Yapılandırır ve Çalıştır

Artık izlenecek bazı düğümlerimiz olduğuna göre, düğümleri izlemek için Prometheus'u çalışma zamanı. Bu adımda, Prometheus'a hangi düğümleri izleyeceğini söylemek için özel bir yapılandırma dosyası da oluşturacaksınız. Ardından, kullanacağı yapılandırma dosyasını geçerek Prometheus'u başlatacaksınız.

Göreviniz

- Öncelikle, terminalinizde aşağıdaki docker pull komutunu çalıştırarak Prometheus'u yerel ortamınıza çekin.

```
docker pull bitnami/prometheus:latest
```

Bu, internet bağlantınıza bağlı olarak birkaç saniye sürebilir.

- Mevcut dizinde **prometheus.yml** adında bir dosya oluşturun. Bu, prometheus'u node exporter örneklerini izlemek için yapılandıracığınız dosyadır.

```
touch prometheus.yml
```

- Gezginden **prometheus.yml** dosyasına gidin ve dosyayı düzenleyin veya aşağıdaki [Open **prometheus.yml** in IDE] butonuna basın:

Open **prometheus.yml** in IDE

- Aşağıdaki içeriği dosyaya ekleyin. URL'leri uygun şekilde güncelleyin. node exporter 1 URL'si, önceki görevde bir not defterine kopyaladığınız URL olacaktır. node exporter 2, port numarası 9102 olarak değiştirilmiş aynı URL olacaktır. node exporter 3, port numarası 9103 olarak değiştirilmiş aynı URL olacaktır.

```
# benim genel yapılandırmam
global:
  scrape_interval: 15s # Tarama aralığını her 15 saniyede bir olarak ayarlayın. Varsayılan her 1 dakikadır.
scrape_configs:
  - job_name: 'node'
    static_configs:
      - targets: ['node-exporter1:9100']
        labels:
          group: 'monitoring_node_ex1'
      - targets: ['node-exporter2:9100']
        labels:
          group: 'monitoring_node_ex2'
      - targets: ['node-exporter2:9100']
        labels:
          group: 'monitoring_node_ex3'
```

Aşağıdaki resimde örnek bir yapılandırma görülebilir.

- Şimdi bu yaml dosyasını bir parametre olarak geçirerek prometheus izleyicisini başlatabilirsınız.

```
docker run --rm --name prometheus -p 9090:9090 --network monitor \
-v $(pwd)/prometheus.yml:/opt/bitnami/prometheus/conf/prometheus.yml \
bitnami/prometheus:latest
```

Sonuçlar

İzlemenin başladığını gösteren Prometheus günlüklerinin ekranda yukarı doğru kaydığını görmelisiniz.

Adım 4: Prometheus UI'yi Aç

Bu adımda Prometheus web UI'yi başlatacak ve sorguları çalıştırılmaya başlayabileceğiniz sayfaya gideceksiniz.

- Prometheus web UI'yi açmak için **Skills Network Toolbox**'a tıklayın ve **Diğerleri** altında **Uygulamayı Başlat** seçeneğini seçin, port numarasını **9090** girin ve başlatma URL'si butonuna tıklayın veya aşağıdaki "Prometheus'u Başlat" butonuna basarak harici bir tarayıcıda başlatın:

Prometheus'u Başlat

- Prometheus uygulama UI'si varsayılan olarak grafik uç noktasında açılır.
- Sonra, menüde **Durum**'a tıklayın ve izlenen hedefleri görmek için **Hedefler**'i seçin.
- Üç node exporter'ın durumunu görüntüleyin.
- Ana sayfaya dönmek için **Grafik**'e tıklayın.

Artık sorguları çalıştırılmaya hazırlısınız.

Adım 5: İlk sorgunuzu çalıştırın

Artık ilk sorgunuzu çalıştırılmaya hazırlısınız. Aşağıdaki sorguyu yapıştırarak toplam CPU saniyesini düğümler için sorgulayın ve çalıştırın. Görselde verilen grafiği gösterecektir. Her bir örneğin üzerine fare ile gelerek detayları gözlemlayabilirsiniz.

Göreviniz

- Aşağıdaki sorguyu yapıştırarak çalıştırın. Görselde verilen grafiği gösterecektir. Her bir örneğin üzerine fare ile gelerek detayları gözlemlayabilirsiniz.

node_cpu_seconds_total

- Ardından **Tablo** sekmesine geçin ve tüm hedefler için CPU saniyesini tablo formatında sorgulayın.

- Şimdi, yalnızca bir örneğin detaylarını almak için sorguyu filtreleyin ve aşağıdaki sorguyu kullanın.

node_cpu_seconds_total{group="monitoring_node_ex2"}

- Son olarak, her düğümün sahip olduğu bağlantıları sorgulamak için bu sorguyu kullanın.

node_ipvs_connections_total

Adım 6: Durdur ve gözleme

Bu adımda, bir node exporter örneğini durduracağız ve bunun Prometheus konsolundaki yansımmasını göreceğiz.

Göreviniz

1. Yeni bir terminal açın.
2. Aşağıdaki docker stop komutunu çalıştırarak node-exporter1 örneğini durdurun ve ardından Prometheus'un çalıştığı eski terminale geri dönün.
`docker stop node-exporter1`

3. Şimdi tarayıcınızdaki Prometheus UI'ye geri dönün ve **Durum -> Hedefler** menü öğesini seçerek hedefleri kontrol edin.

Sonuçlar

Artık izlenen node exporter'ların birinin kapalı olduğunu görmelisiniz. *Not: Yeni durumu görmek için tarayınızı yenilemeniz gerekebilir.*

Adım 7: Grafana'yı Başlat

1. Grafana'yı docker'dan çekmek ve çalıştırmak için aşağıdaki komutu çalıştırın. Bunu node monitor ve prometheus ile aynı ağıda çalıştıracağız.

```
docker run --name=grafana -dp 3000:3000 --network monitor grafana/grafana
```

2. Şimdi aşağıdaki Launch Application butonuna basarak bağlantı ile Grafana tarayıcısını açın.

[Launch Grafana](#)

3. Kullanıcı adı olarak admin ve şifre olarak admin ile giriş yapın. Yeni bir şifre belirlemeyi seçebilirsiniz. Bu sizi Grafana ana sayfasına götürecektir. Eğer yeni bir şifre belirlerseniz, bu yalnızca o docker örneği için geçerli olacaktır. Yeniden başlattığınızda bunu sıfırlamanız gerekecek.
4. İlk veri kaynağını eklemek için Data Sources üzerine tıklayın.
5. Mevcut seçenekler listesinden Prometheus'u seçin.

Not: Bu laboratuvar, node exporter'ları izlemek ve metrikleri ve grafikleri görselleştirmek için Prometheus kullanmaya odaklanmaktadır. Adım 7'de Grafana'nın başlatılmasından bahsedilmesine rağmen, laboratuvar node exporter'ların izlenmesinde Prometheus'un yeteneklerini göstermek için tasarlanmıştır. Bu nedenle, bu özel laboratuvar için Grafana'yı yapılandırmaya devam etmiyoruz.

Sonuç

Tebrikler! Prometheus laboratuvarını tamamladınız. Artık uygulamalarınızı doğru bir şekilde çalıştırıldıklarından emin olmak için izlemeye hazırlısınız.

Sonraki Adımlar

Bir sonraki zorluğunuz, uygulamalarınızı izlemek için geliştirme ortamınızda Prometheus'u kurmaktır. Bu laboratuvar sırasında öğrendiğiniz bazı soruları kullanarak uygulamalarınızın sağlık ve performansını kontrol edebilirsiniz.

Author(s)

Lavanya T S

Other Contributor(s)

Pallavi Rai
[John J. Rofrano](#)

© IBM Corporation 2023. Tüm hakları saklıdır.