Red Hat OpenShift'e Giriş

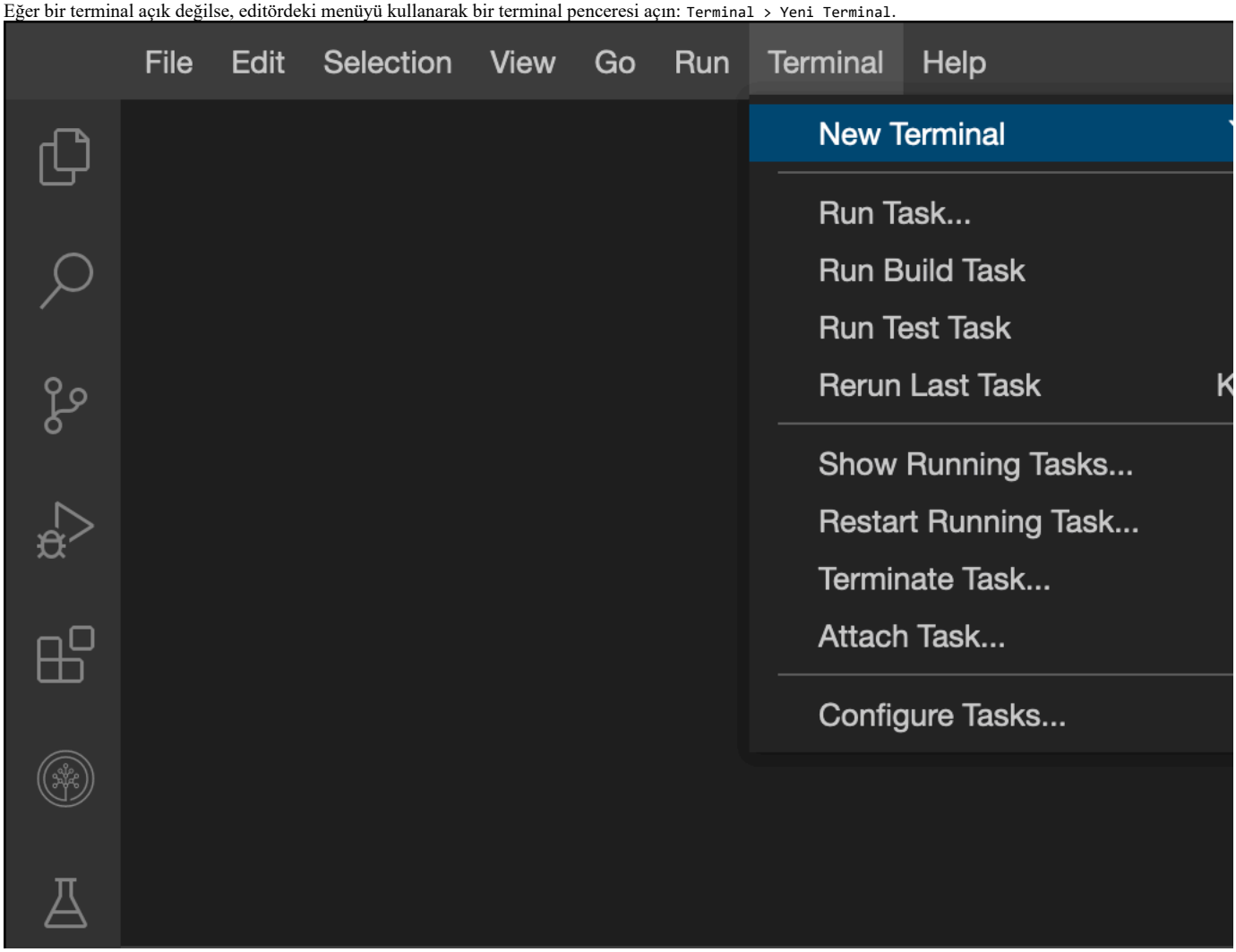
Amaçlar

Bu laboratuvarda şunları yapacaksınız:

- oc CLI'yi (OpenShift komut satırı arayüzü) kullanın
- OpenShift web konsolunu kullanın
- s2i ('Source-to-image' yapı stratejisi) kullanarak bir uygulama oluşturun ve dağıtın
- Bir BuildConfig ve bir ImageStream'i inceleyin
- Uygulamayı otomatik ölçeklendirin

Ortamı ve komut satırı araçlarını doğrulayın

1. Eğer bir terminal açık değilse, editördeki menüyü kullanarak bir terminal penceresi açın: `Terminal > Yeni Terminal`.



   Not: Terminal istemcisinin görünmesi için biraz bekleyin.

2. oc CLI'nin kurulu olduğunu doğrulayın.

   ```
   oc version
   ```

```
theia@theiaopenshift-[redacted]:/home/project$ oc version
Client Version: 4.13.7
Kustomize Version: v4.5.7
Kubernetes Version: v1.27.16+03a907c
```

Buna benzer bir çıktı görmelisiniz, ancak sürümler farklı olabilir.

3. Proje klasörünüze geçin.

   NOT: Eğer zaten `home/project` klasöründeyseniz bu adımı atlayabilirsiniz.

   ```
   cd /home/project
   ```

4. Bu laboratuvar için gerekli olan belgeleri içeren git deposunu, eğer zaten mevcut değilse, klonlayın.

   ```
   [ ! -d 'CC201' ] && git clone https://github.com/ibm-developer-skills-network/CC201.git
   ```

```
theia@theiaopenshift-[redacted]:/home/project$ [ ! -d 'CC201' ] && git clone https://github.com/ibm-developer-skills-network/CC201.gi
t
Cloning into 'CC201'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 20 (delta 6), reused 19 (delta 6), pack-reused 0
Unpacking objects: 100% (20/20), done.
```

oc CLI Kullanımı

OpenShift projeleri, ek yönetim işlevleri ile birlikte Kubernetes ad alanlarıdır. Bu nedenle, projeler aynı zamanda bir OpenShift kümesi içinde izolasyon sağlar. Bir OpenShift kümesinde bir projeye zaten erişiminiz var ve `oc` zaten o küme ve projeyi hedef alacak şekilde ayarlanmıştır.

Bazı temel `oc` komutlarına bakalım. `oc`'nun bir kopyası ile birlikte geldiğini unutmayın, bu nedenle tüm `kubectl` komutları `oc` ile çalıştırılabilir.

1. Bu ad alanındaki Pod'ları listeleyin.

   ```
   oc get pods
   ```

```
theia@theiaopenshift-[redacted]:/home/project$ oc get pods
NAME                                READY   STATUS    RESTARTS   AGE
openshift-web-console-995896df-vz2tp   2/2     Running   0          4h1m
```

Muhtemelen ortamın bir parçası olan birkaç Pod göreceksiniz. Bunlar hakkında endişelenmenize gerek yok.

2. Kubernetes nesnelerine ek olarak, OpenShift'e özgü nesneleri de alabilirsiniz.

   ```
   oc get buildconfigs
   ```

```
theia@theiaopenshift-[redacted]:/home/project$ oc get buildconfigs
No resources found in sn-labs-[redacted] namespace.
```

Henüz bir BuildConfig oluşturmadığınız için bu, herhangi bir kaynak döndürmeyecektir.

3. Şu anda kullanılan OpenShift projesini görüntüleyin.

```
oc project
```

```
theia@theiaopenshift-[redacted]:/home/project$ oc project
Using project "sn-labs-[redacted]" from context named "[redacted]-context" on server "https://c109-e.us-east.containers.cloud.ibm.com:30807".
theia@theiaopenshift-[redacted]:/home/project$
```

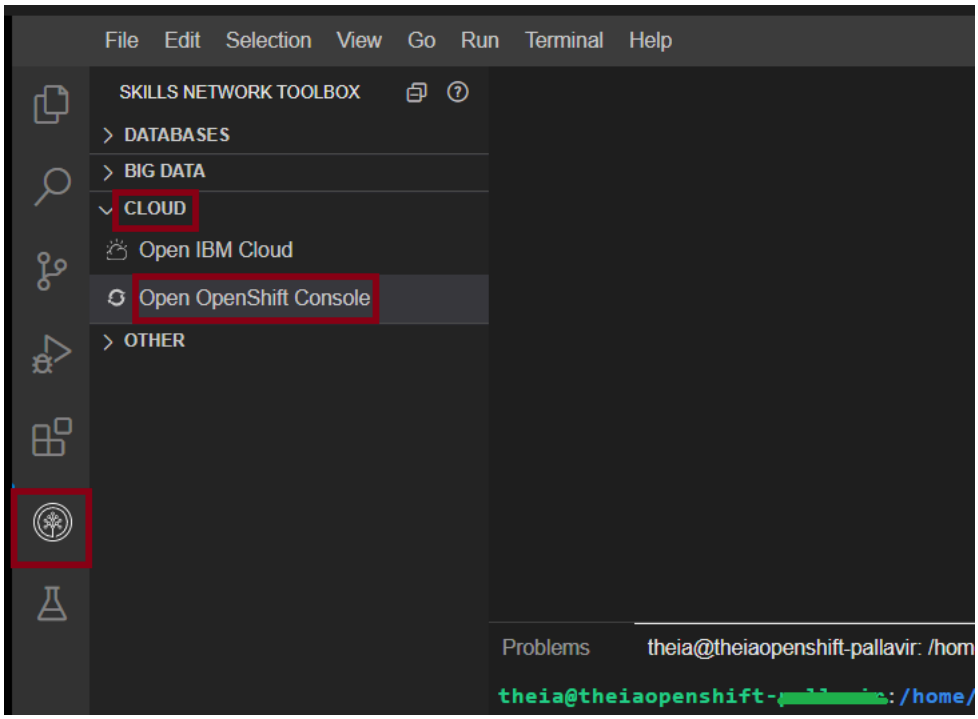
Bu proje size özeldir ve küme içinde izolasyon sağlar, böylece kendi uygulamalarınızı dağıtabilirsiniz.

OpenShift web konsolunu kullanın

CLI'ye ek olarak, OpenShift sezgisel bir web konsolu sağlar. Bu, uygulamaları dağıtmanıza, kaynakları görüntülemenize, uygulamaları izlemenize ve günlükleri görmenize olanak tanıdığı için faydalı ve güçlü bir özelliktir ve çok daha fazlasını konsolda yapmanızı sağlar.

Konsolu açalım ve etrafa bakalım.

1. OpenShift web konsolunu açmak için sağdaki Skills Network butonuna tıklayın, bu **Skills Network Toolbox**'u açacaktır. Ardından **Cloud**'a tıklayın ve ardından aşağıdaki resimde gösterildiği gibi **Open OpenShift console**'a tıklayın.

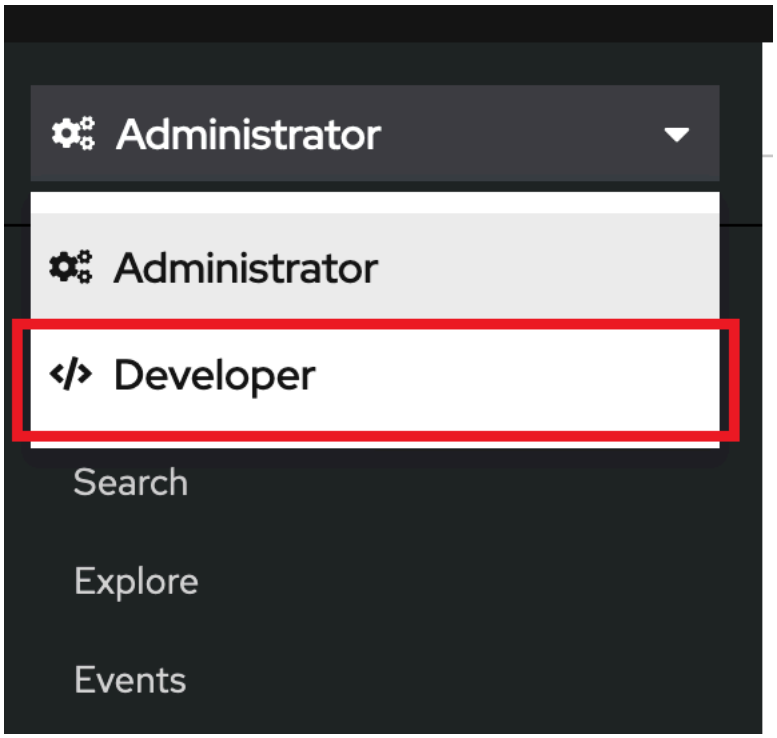


Laboratuvar ortamını açtıktan sonra erişilebilir hale gelmesi birkaç dakika alabilir, bu nedenle bir hata alırsanız bir dakika bekleyip tekrar deneyin.

2. Konsol, size atanmış proje için proje detaylarına açılmalıdır. OpenShift'in size sezgisel ve görsel bir şekilde sunduğu tüm bilgilere göz atın. Bu proje için Dashboard, Overview ve diğer sekmelere tıklayarak ek bilgilere ulaşabilirsiniz. Bu projede mevcut olan kaynakların envanterini, bu projeyi tanımlayan YAML'yi ve çok daha fazlasını görmelisiniz.

3. Sol navigasyon menüsündeki öğelere aşina olun. Operatörleri, birçok farklı Kubernetes nesnesini ve bu kursta bahsettiğimiz bazı OpenShift'e özgü nesneleri görebilirsiniz. Bu nesnelerin henüz çok fazla örneği olmayacak, ancak uygulamamızı dağıttığımızda dolacaktır.

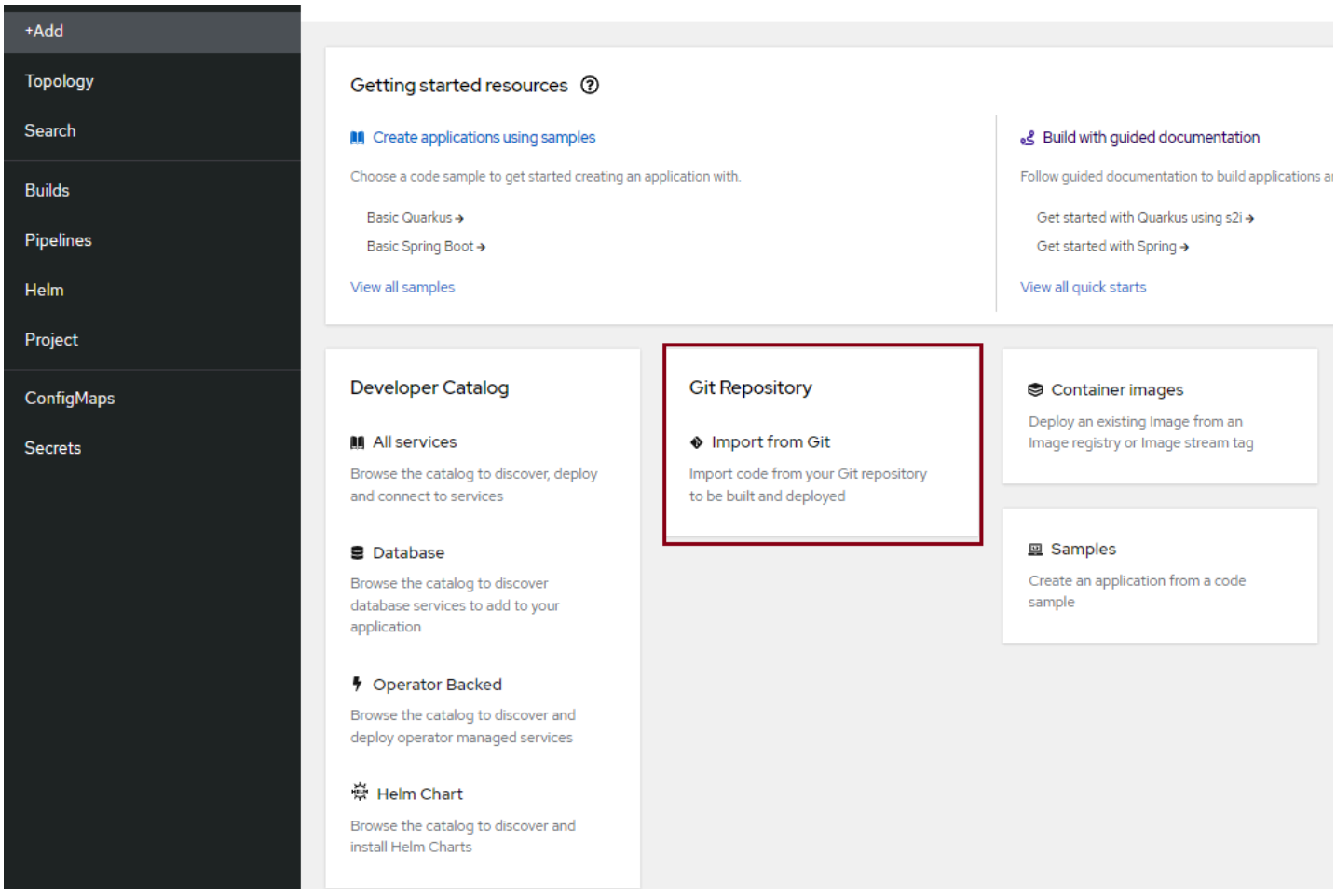
4. Sol üstteki **"Administrator"** kelimesine dikkat edin. Bu, Yönetici perspektifinde olduğunuzu gösterir. Ayrıca bir Geliştirici perspektifi de vardır. Her perspektif, o persona için özel iş akışları sunar. Bir uygulama dağıtmaya başlamak için **Geliştirici perspektifine geçin**. (Eğer zaten "Developer" yazıyorsa, değiştirmeyin.)



Bir uygulamayı web konsolunda dağıtmak

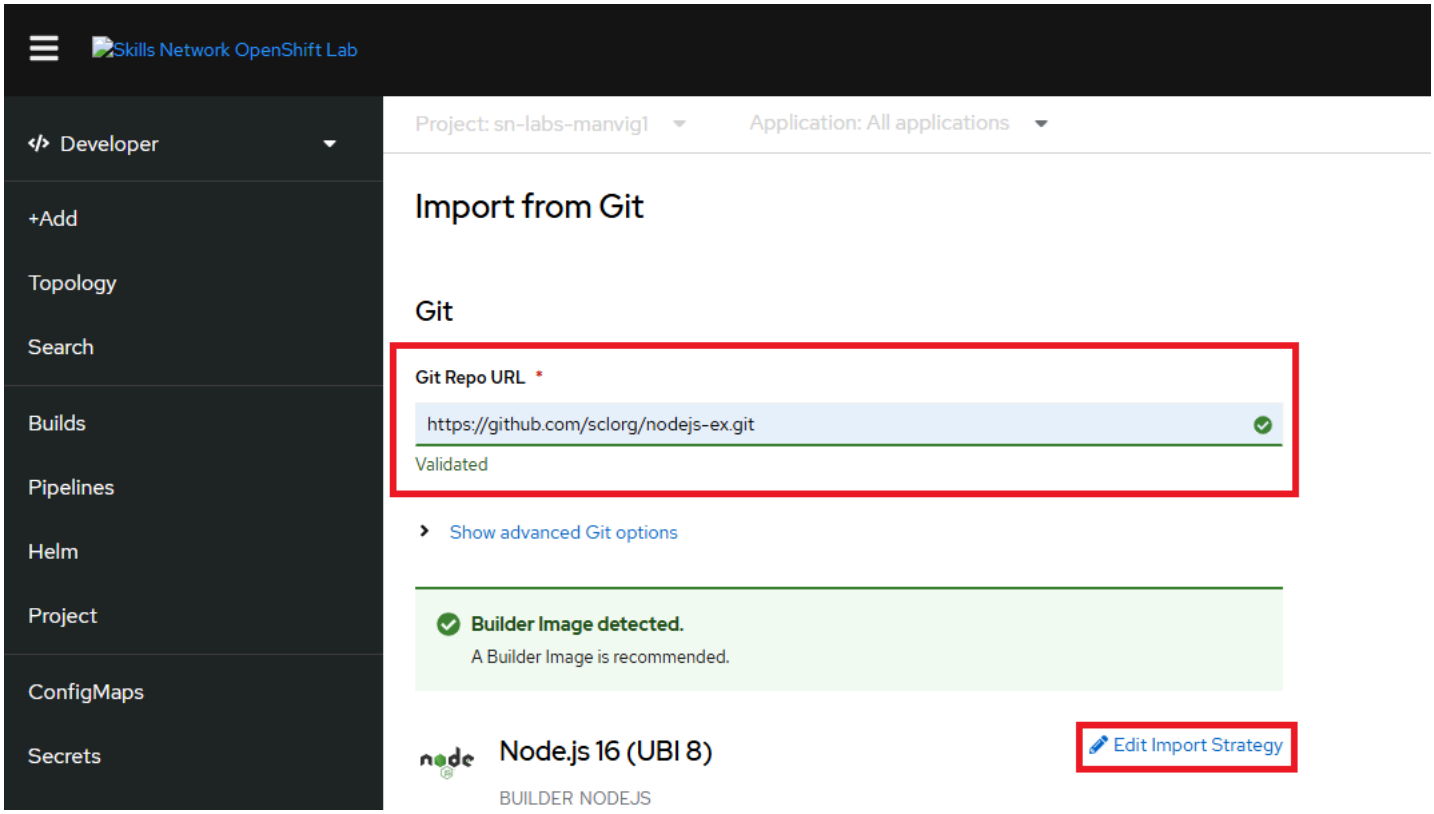
Geliştirici perspektifi, uygulama oluşturma ve dağıtma gibi geliştirici kullanım durumlarına özgü iş akışları sunar. Buradan başlayalım! Muhtemelen "Topoloji" görünümündesiniz, bu görünüm uygulamaların görsel temsilini sağlar. Değilse, göz atmak için bu görünüme geçin.

1. Bu projeye yeni bir uygulama ekleyelim. Openshift'te yeni bir uygulama eklemenin birkaç yolu vardır.
2. Yeni bir uygulama eklemek için +**Ekle** butonuna tıklayın.
3. Seçenekler arasında **Git Deposu (Git'ten İçe Aktar)** seçeneğini seçin.

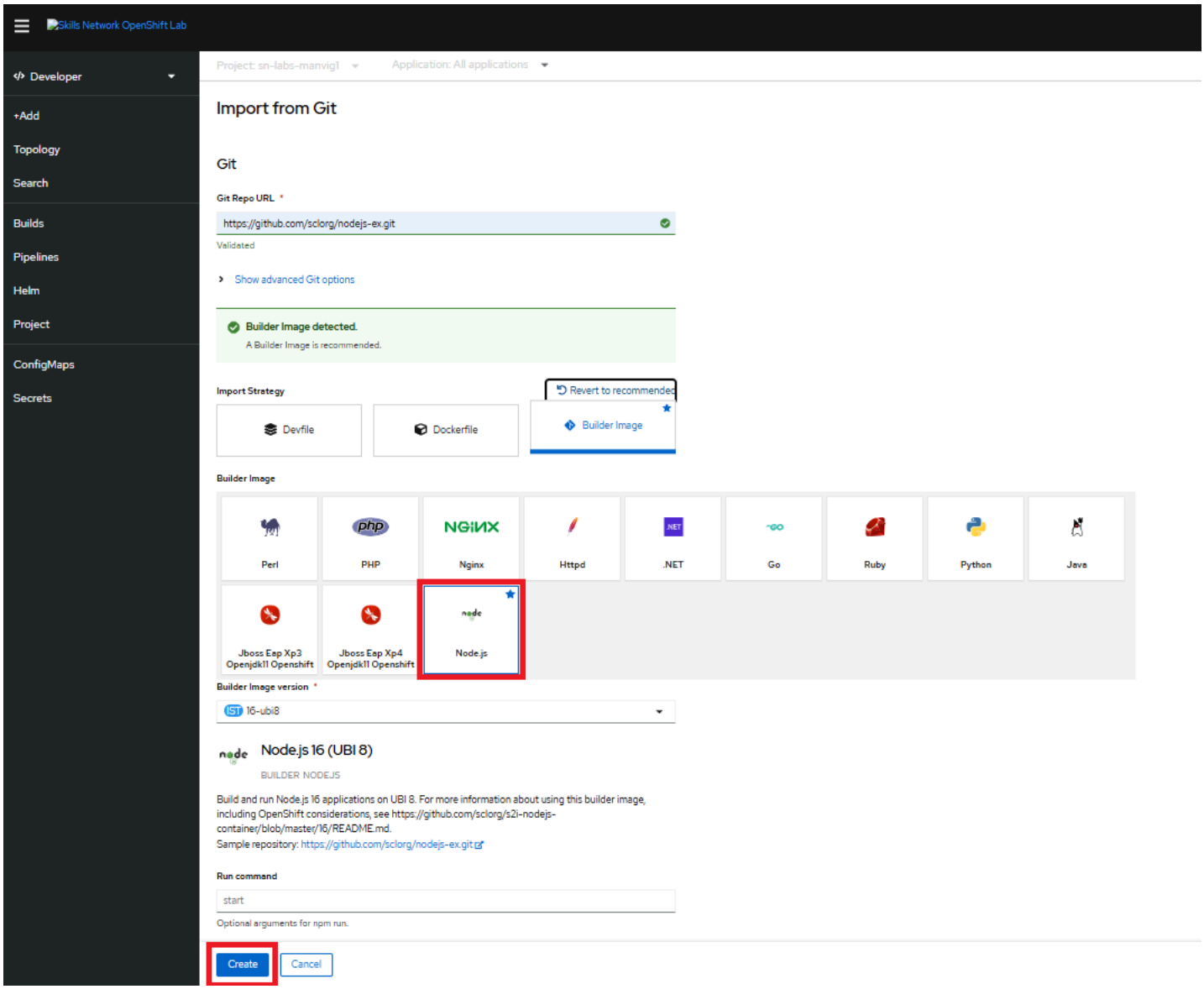


4. **Git'ten İçe Aktar** penceresine yönlendirileceksiniz. OpenShift, sizden yalnızca bir girdi alarak bir uygulama dağıtacak: uygulama kaynağı.
5. **Git Repo URL** kutusuna aşağıda belirtilen örnek URL'yi yapıştırın.

```
https://github.com/sclorg/nodejs-ex.git
```



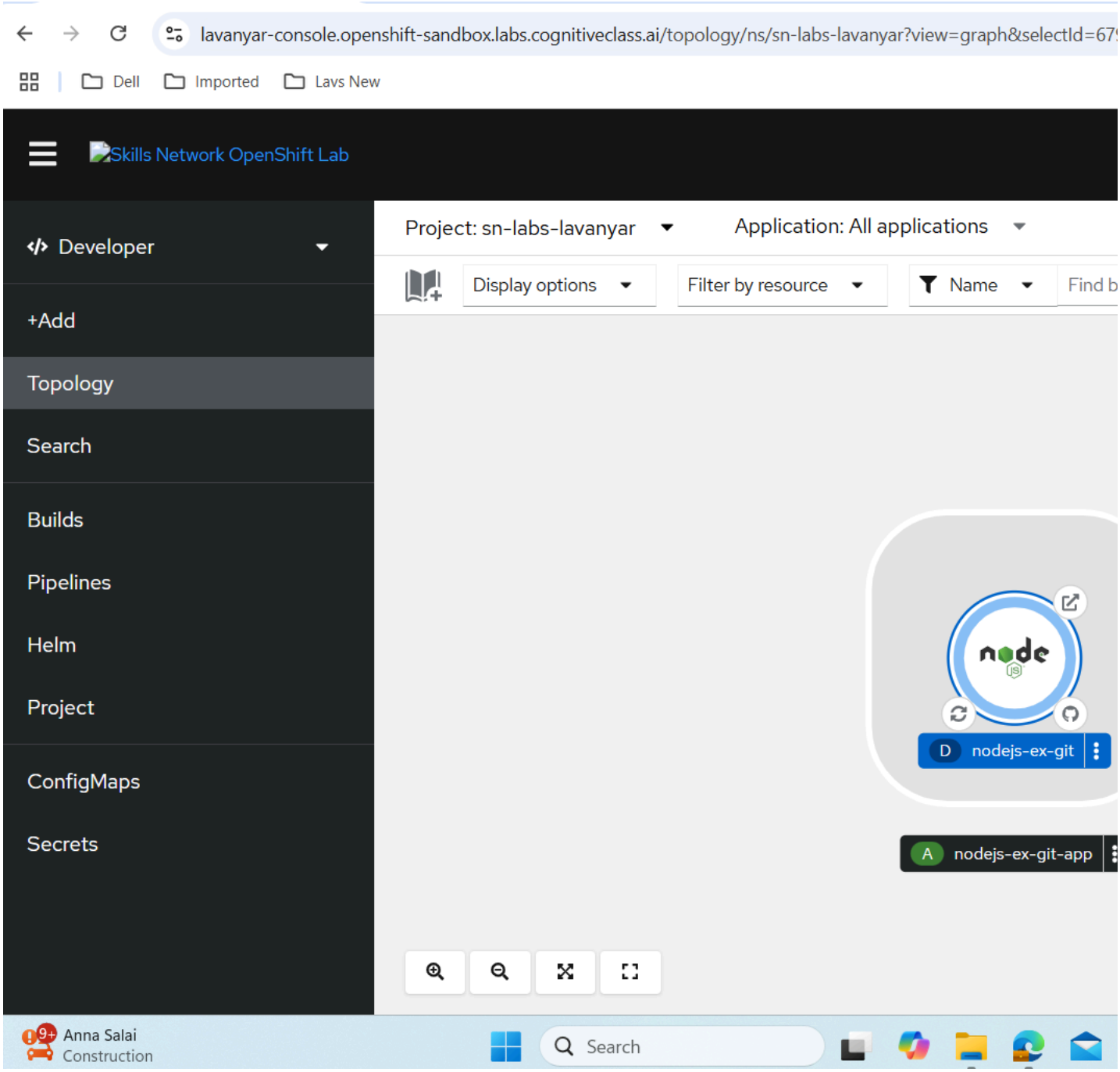
Not: Çeşitli builder görüntülerini görmek için İçe Aktarma Stratejisini Düzenle'ye tıklayın. Uygulamamız için Node.js görüntüsünü kullanacağız. Bu görüntünün seçili olduğundan emin olun.



6. Diğer varsayılan seçenekleri olduğu gibi bırakın. Ardından aşağı kaydırın ve **Oluştur**'a tıklayın.

Topoloji görünümünde, şimdi yeni oluşturduğunuz uygulamayı görmelisiniz.

NOT: Uygulamanın görünmesi birkaç dakika sürecektir. 3 dakika içinde herhangi bir uygulama görmüyorsanız, tarayıcıyı yenileyin.



Not: İlk derlemenin başarılı bir şekilde çalışmasını bekleyin. Beklerken **ImagePullBackOff** ve **ErrImagePull** hatalarını geçici olarak görebilirsiniz.

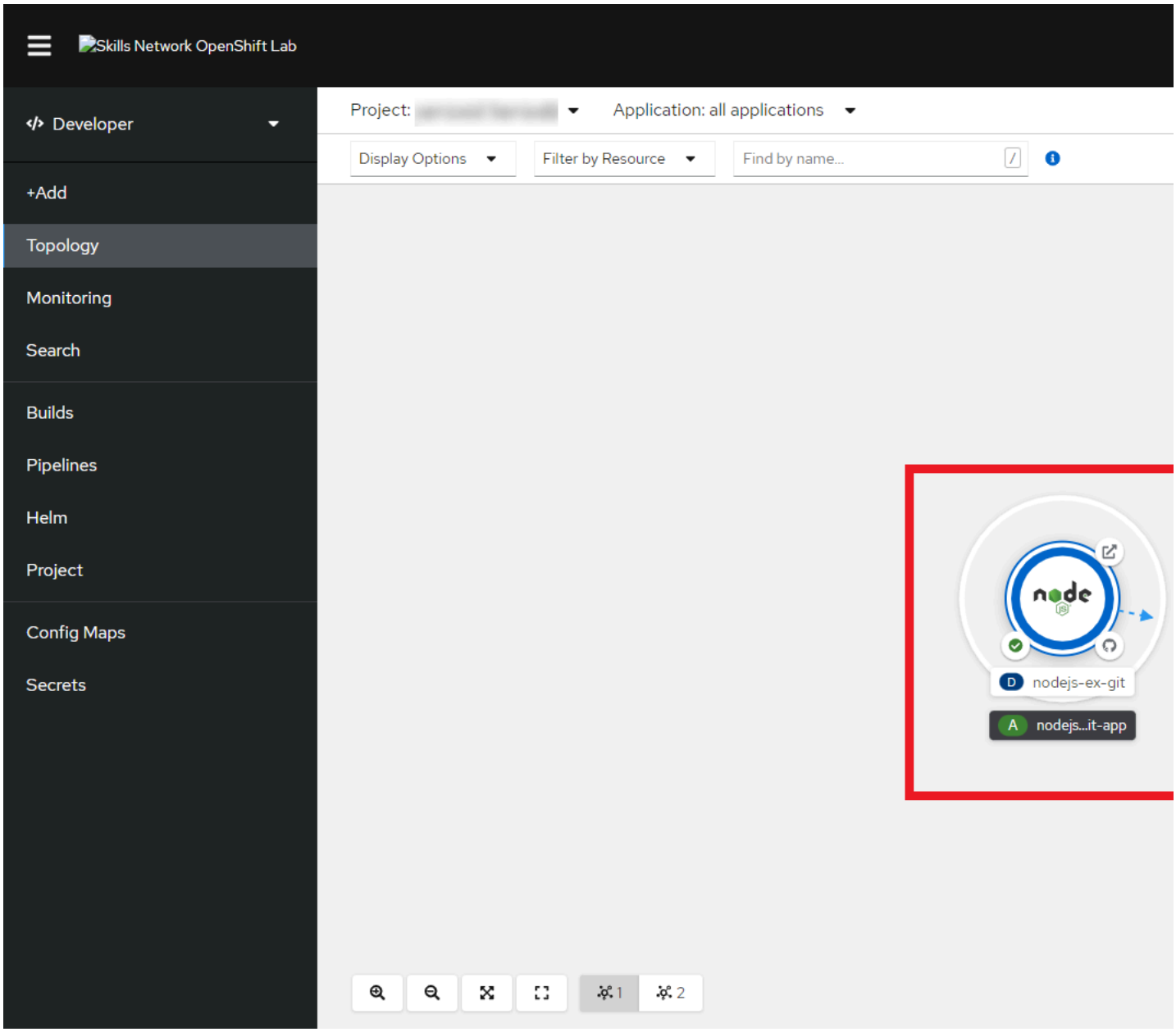
Uygulamayı web konsolunda görüntüle

Topoloji görünümü, bir uygulamanın birçok önemli kısmına hızlı bağlantılar sağlar:

- Dış daire, uygulama hakkında bilgi alır.
- Node.js logosu bulunan iç daire, Dağıtım hakkında bilgi verir.
- GitHub simgesi, kod deposuna erişim için kullanılır.
- Onay işareti, en son yapıyı gösterir (yapı devam ediyorsa dairesel oklar göreceksiniz).
- Bir kutudan çıkan ok, uygulama harici olarak mevcutsa tarayıcıda uygulamayı görüntülemek için kullanılabilir.

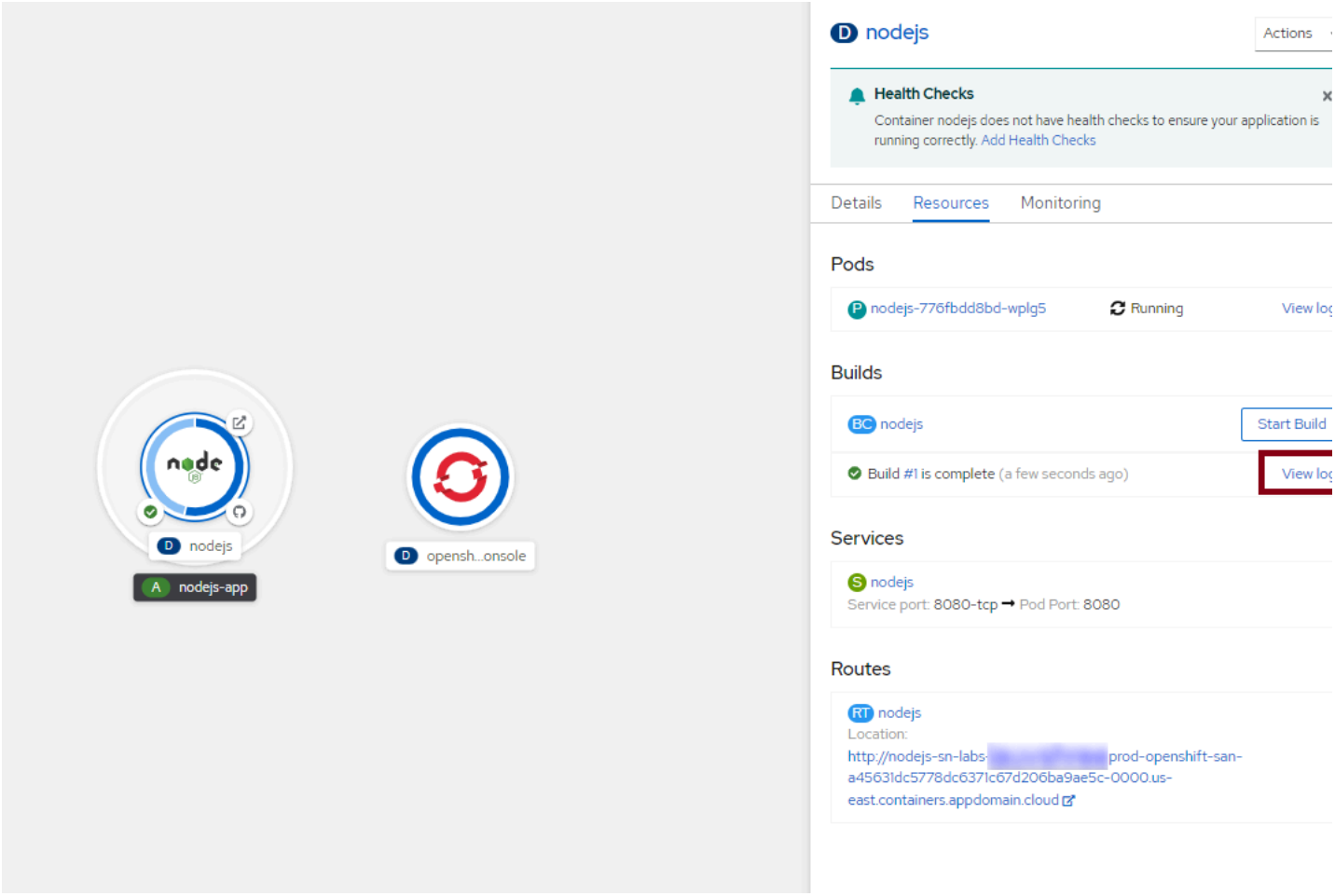
Belirli adımları deneyelim:

1. Node.js logosu bulunan iç daireye tıklayarak Dağıtım hakkında bilgi alabilir ve bu Dağıtım ile ilişkili dört kaynağı gözlemleyebilirsiniz: konteynerleştirilmiş uygulamayı çalıştıran bir Pod; uygulamayı bir konteyner görüntüsüne dönüştürmek için s2i stratejisini kullanan bir Build; uygulamayı bir ağ hizmeti olarak açığa çıkaran bir Service; ve dışarıdan erişilebilir bir ana bilgisayar adı sağlayan bir Route.

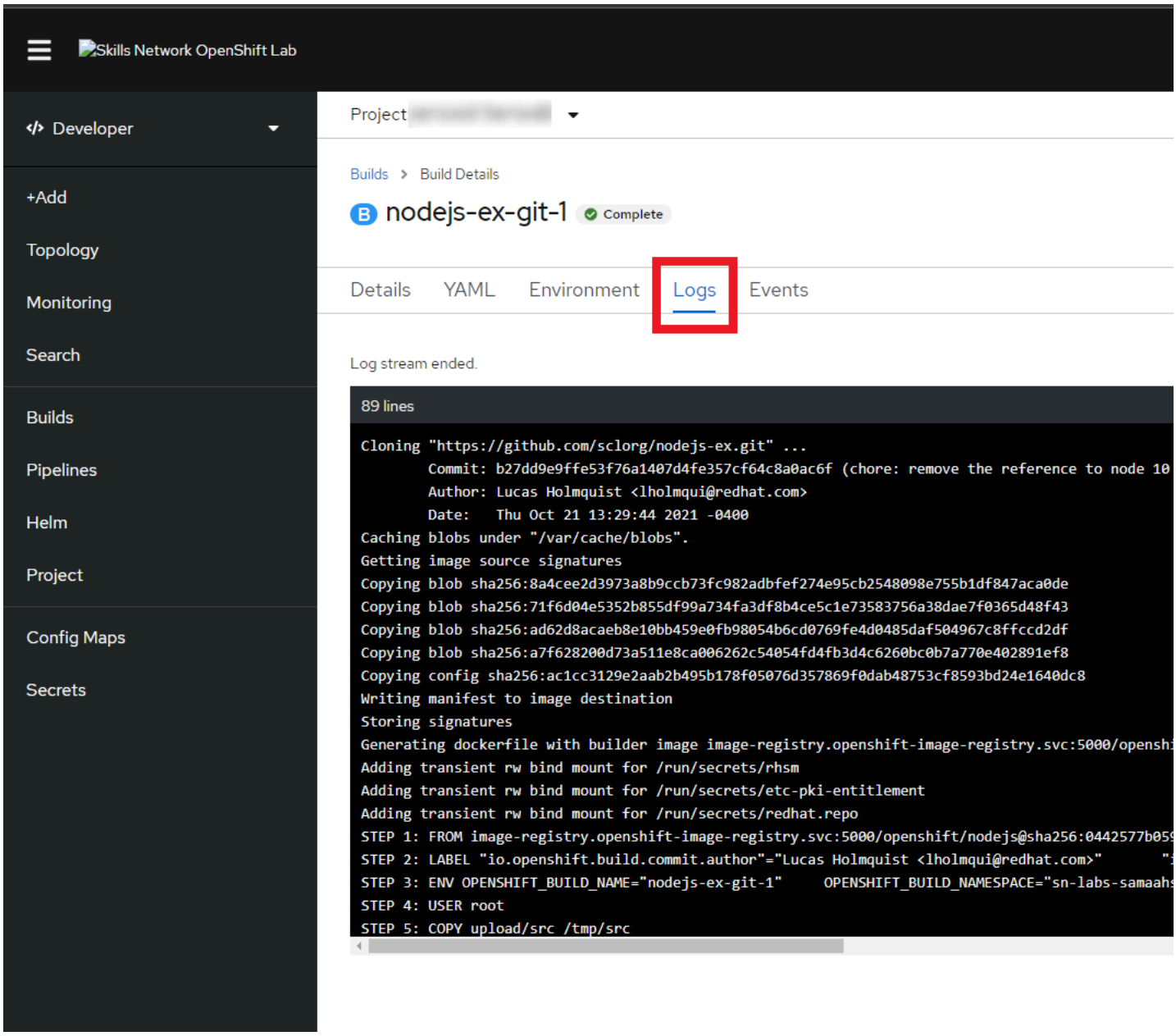


Not: Lütfen pod'un durumunun 'Çalışıyor' olarak değişmesini ve Yapının tamamlanmasını bekleyin.

2. **Build #1** yazılı satırda **Logları Görüntüle** seçeneğine tıklayın.

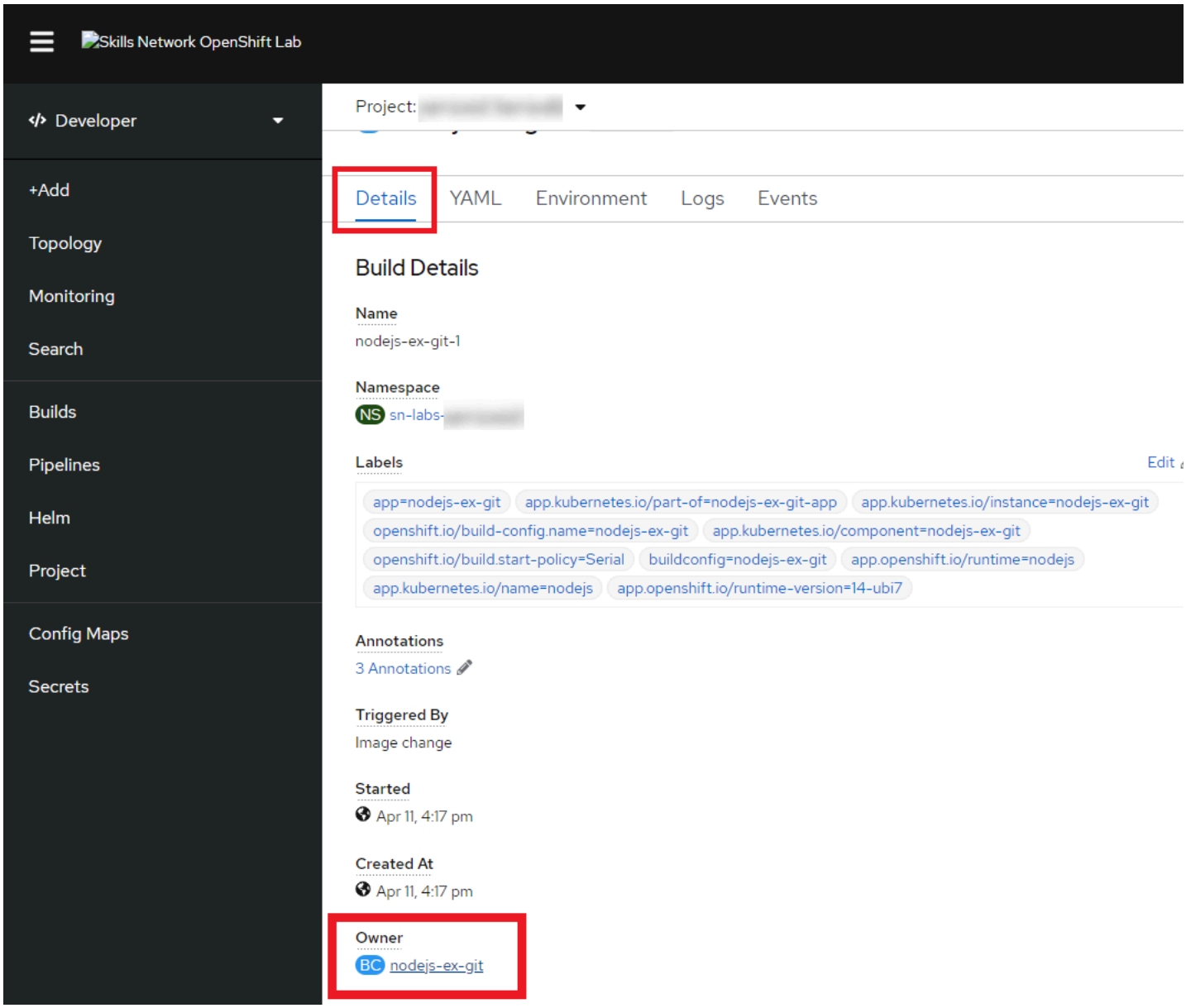


3. Günlükleri okuyarak birkaç önemli tamamlanmış adımı görün. Depo klonlanır, bir Dockerfile oluşturulur, bir görüntü oluşturulur ve görüntü iç kayıt defterine itilir.

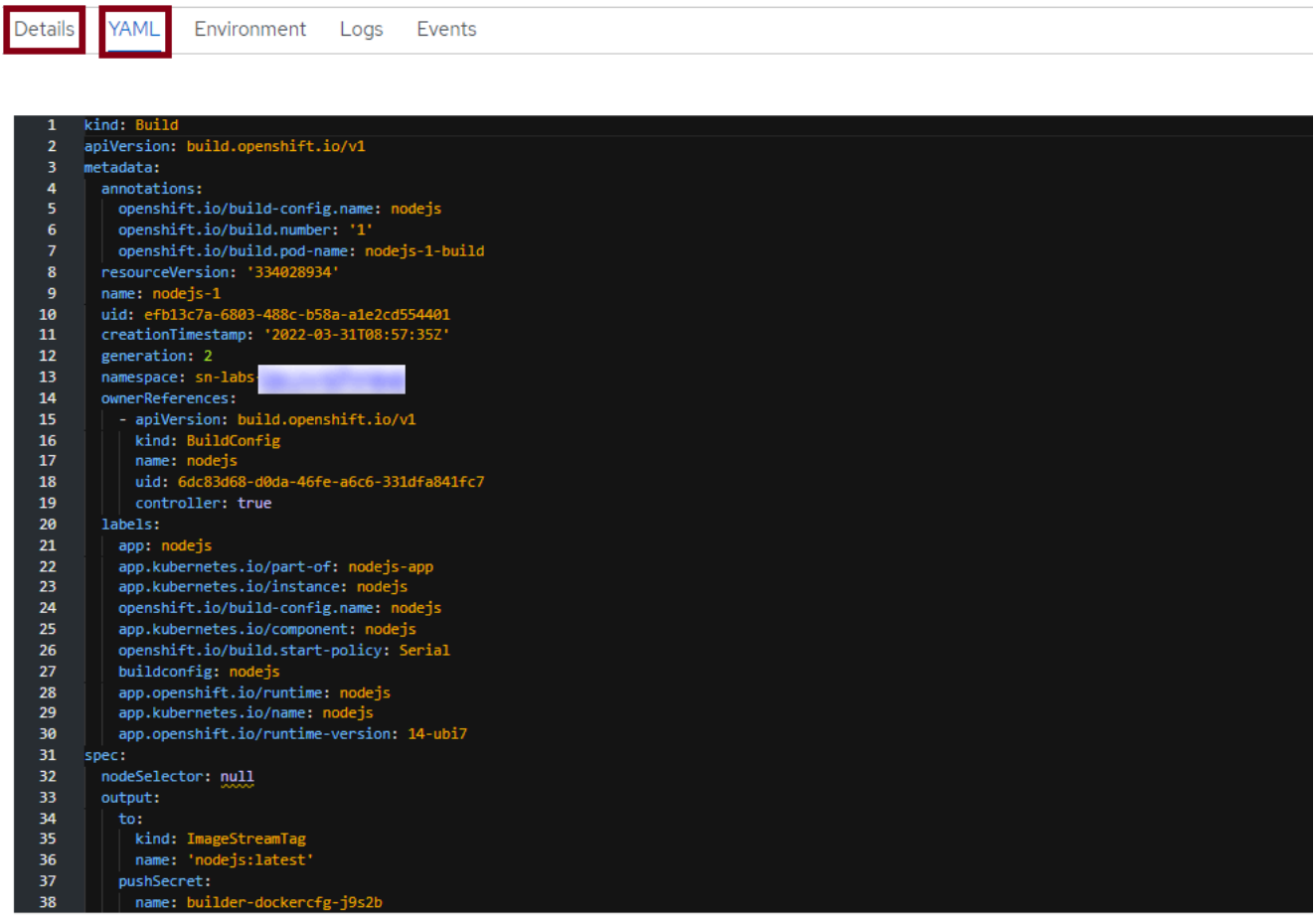


4. Bu Yapının **Ayrıntılar** sekmesine tıklayın.

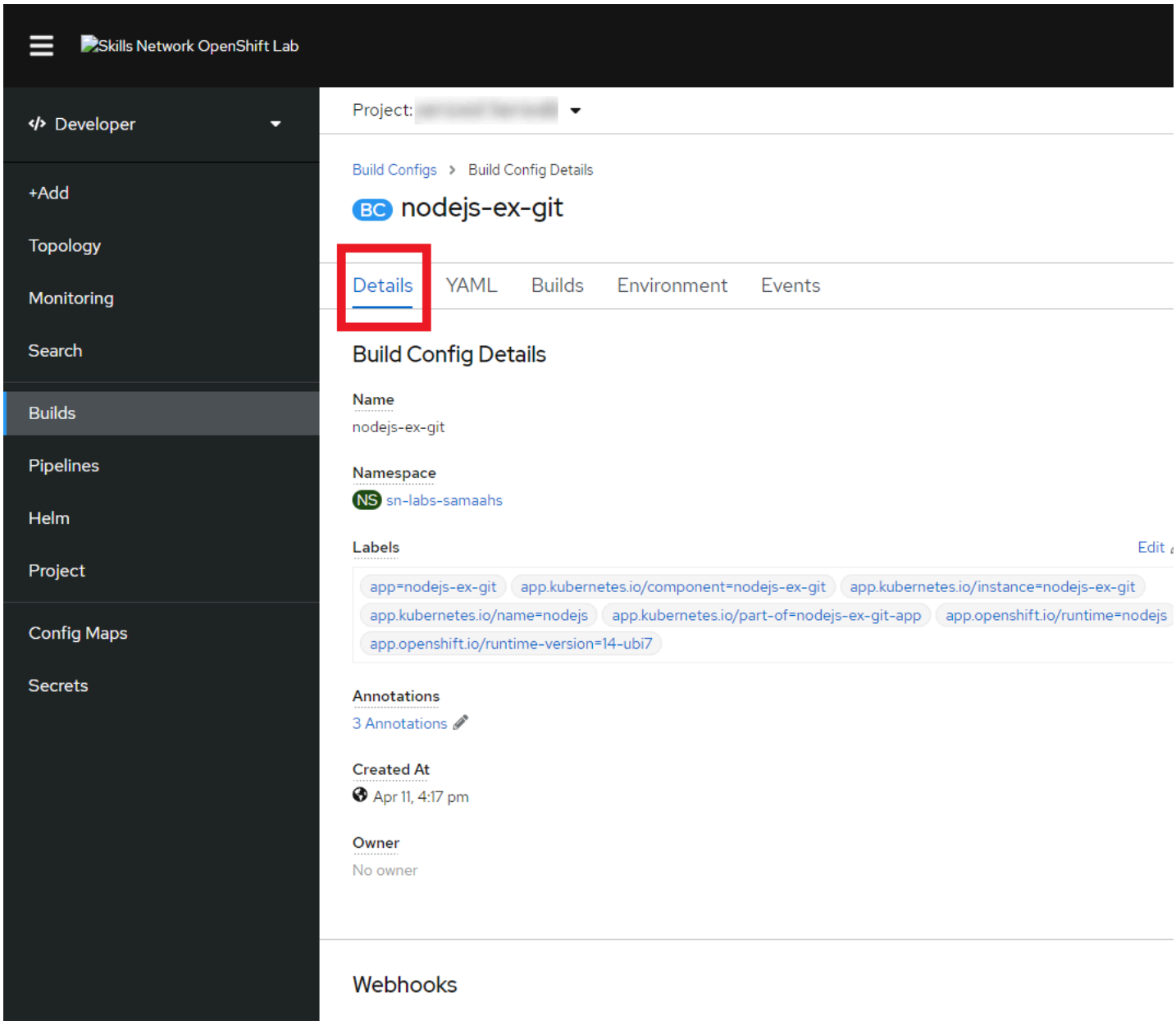
5. Ardından, en altta bulunan **Sahibi** altındaki BC (Build Config) yazılı bağlantıya tıklayın.



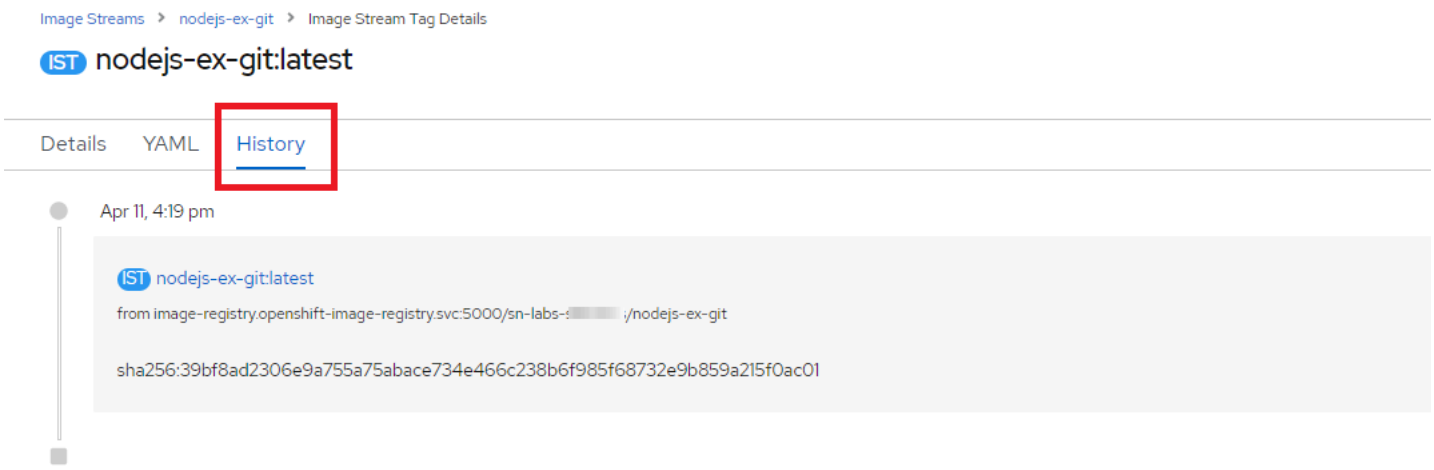
6. **Ayrıntılar** ve **YAML** sekmelerine baktığınızda, bu modülde konuştuğumuz birçok kavramı göreceksiniz: tetikleyiciler, yapı stratejisi, web kancaları ve daha fazlası.



7. **Ayrıntılar** sekmesinde, **Çıktı** altındaki IST (ImageStreamTag) yazılı bağlantıya tıklayın.

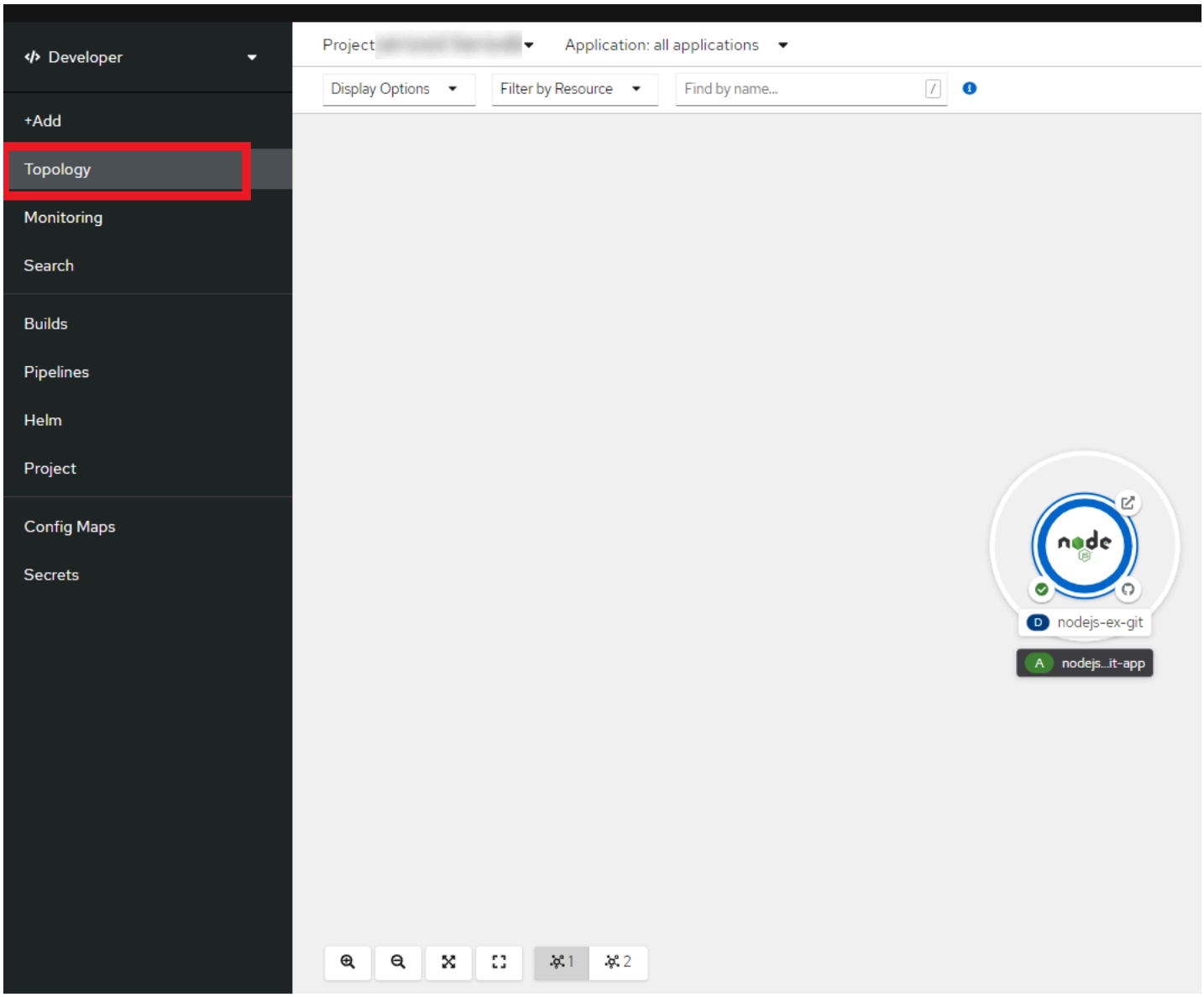


8. Artık yapının çıktısı olarak oluşturulan ImageStreamTag'yi görebilirsiniz. Bu ImageStreamTag'nin işaret ettiği iç kayıt defterindeki görüntüyü görmek için **Geçmiş** sekmesine tıklayın.



9. Topoloji görünümüne geri dönün ve Dağıtım bilgilerinize tıklayın. OpenShift'in sizin için otomatik olarak oluşturduğu Route'a tıklayın. Bu, uygulamayı tarayıcıda açacaktır.

   Not: Lütfen bu URL'yi not edin, çünkü bir sonraki bölümde kullanılacaktır.



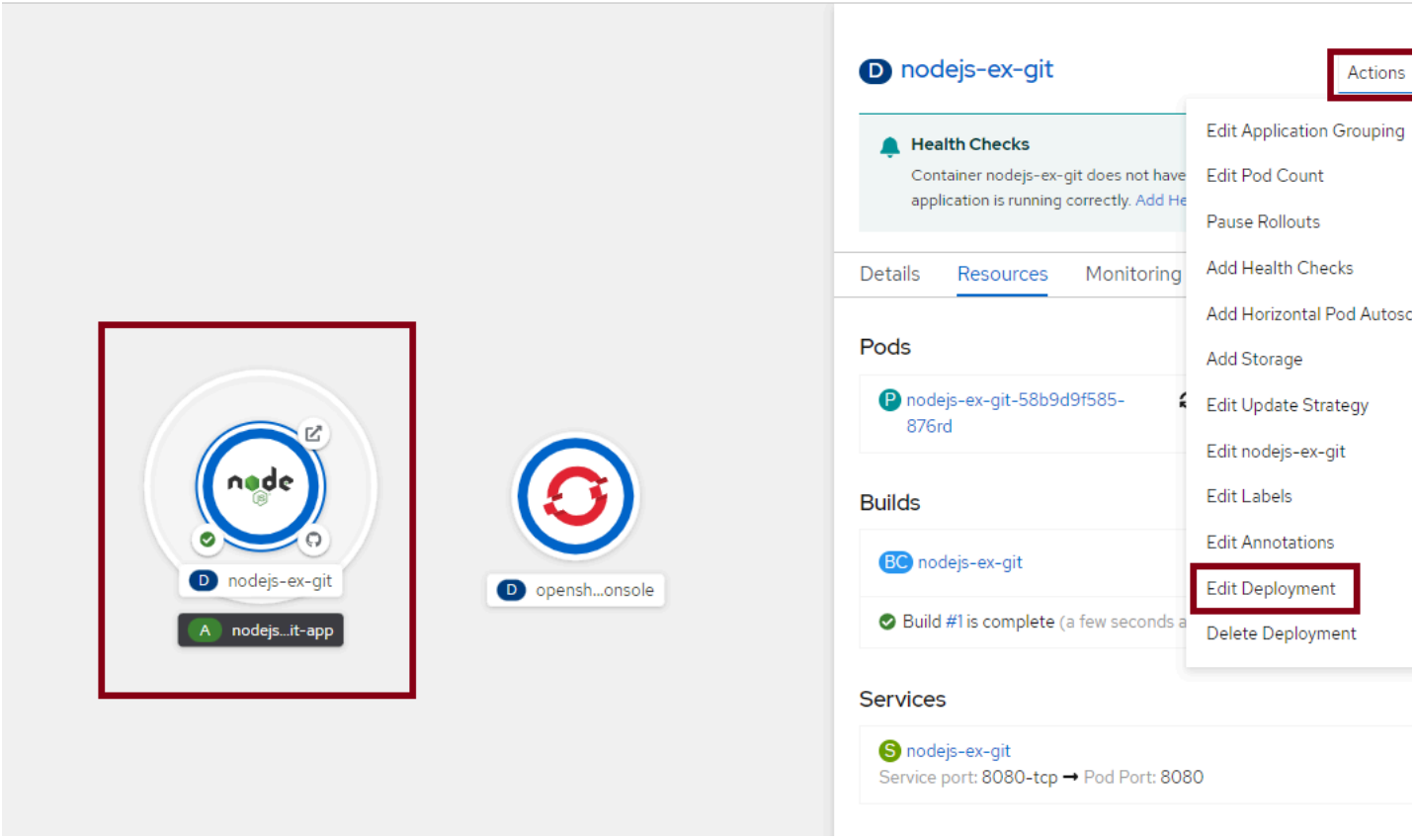
Otomatik Ölçeklendirme `nodejs-ex-git` Uygulaması

Artık `nodejs-ex-git` uygulaması başarıyla çalıştığına göre, gelen yükü karşılayabilmesi için yatay pod otomatik ölçekleyici (HPA) ayarlayalım. `nodejs-ex-git` uygulamasını bir tarayıcı sekmesinde açık tutmayı unutmayın, böylece talepleri yapmaya ve kaynak tüketmeye devam edebilir ve başarıyla otomatik ölçeklendirilir.

Öncelikle, çalışacak konteynerler için kaynak talepleri ve limitleri ayarlamamız gerekiyor. Bir konteyner CPU veya bellek gibi bir kaynak talep ettiğinde, Kubernetes bunu yalnızca o kaynağı verebilecek bir düğümde planlar. Öte yandan, limitler bir konteynerin belirli bir miktardan fazla kaynak tüketmesini engeller.

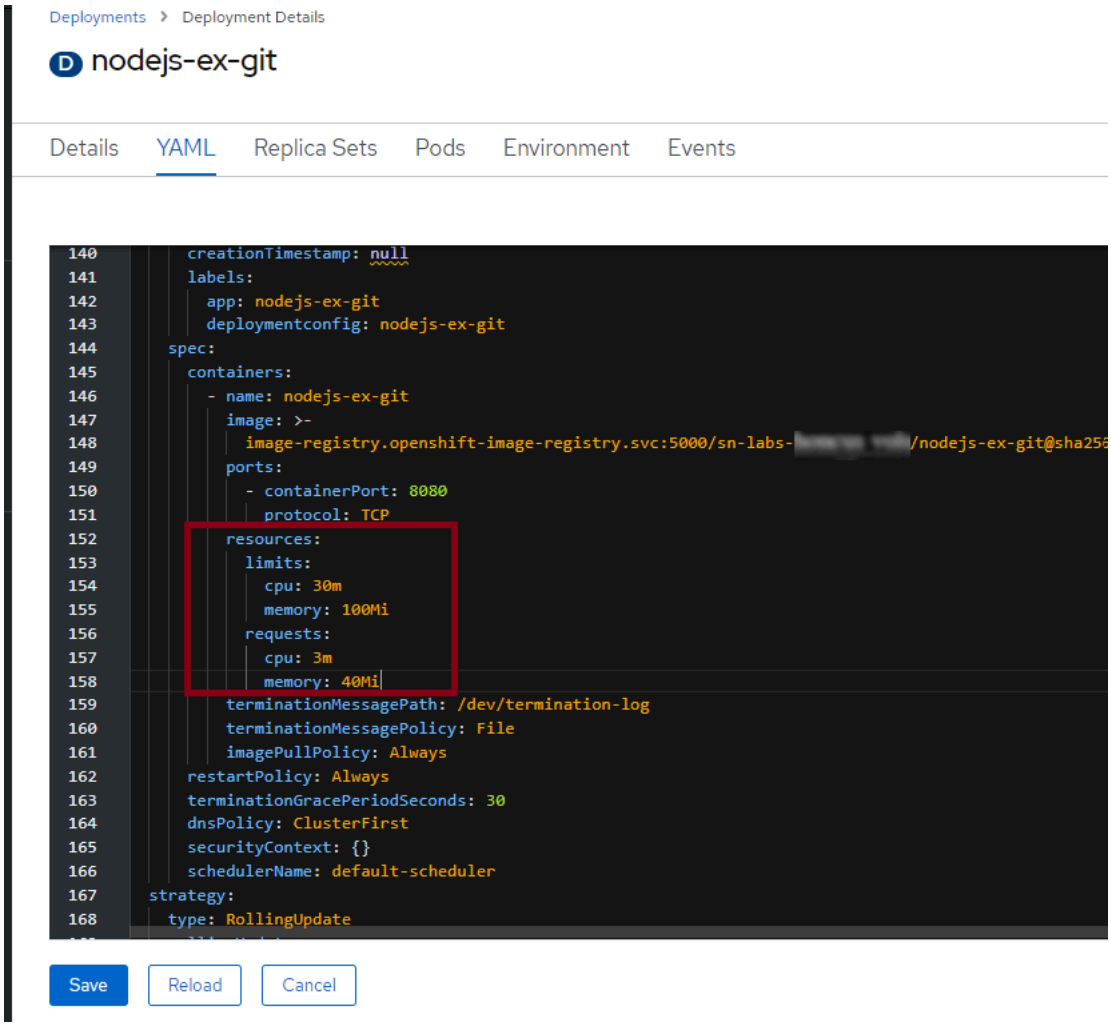
Bu durumda, 3 milikoro CPU ve 40 MB RAM talep edeceğiz. Konteynerleri 30 milikoro ve 100 MB ile sınırlayacağız. Bu sayılar, uygulamanın ölçeklenmesini sağlamak için uydurulmuştur.

1. Topoloji görünümünden, `nodejs-ex-git` Dağıtımına tıklayın. Ardından, İşlemler > Dağıtımı Düzenle'ye tıklayın.

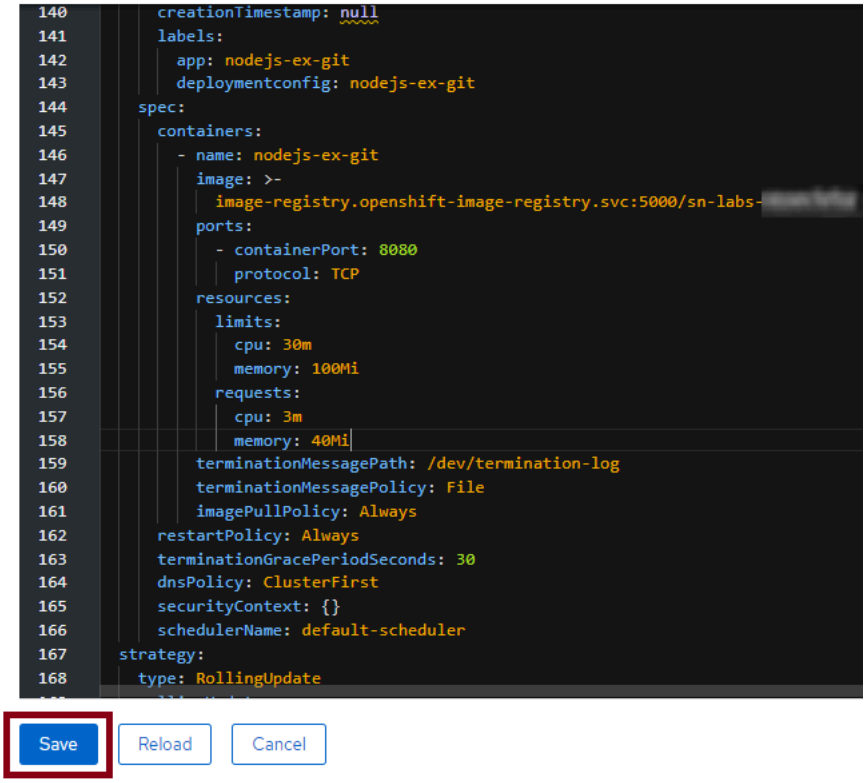


2. template.spec.containers bölümünde, resources: {} kısmını bulun. Bunu aşağıdaki metinle değiştirin. YAML'nin katı girinti kullandığını unutmayın, bu nedenle boşlukların doğru olduğundan emin olun.

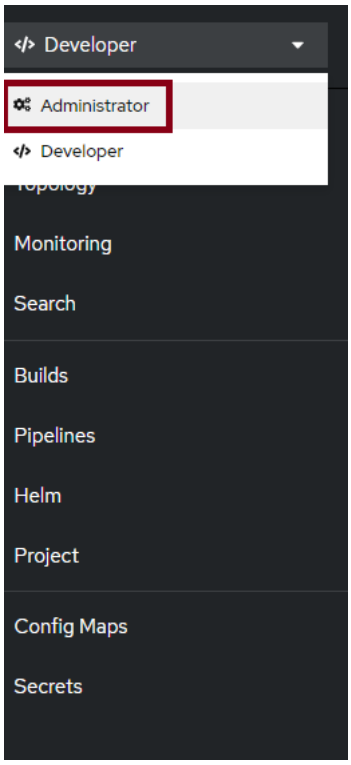
```
resources:
  limits:
    cpu: 30m
    memory: 100Mi
  requests:
    cpu: 3m
    memory: 40Mi
```



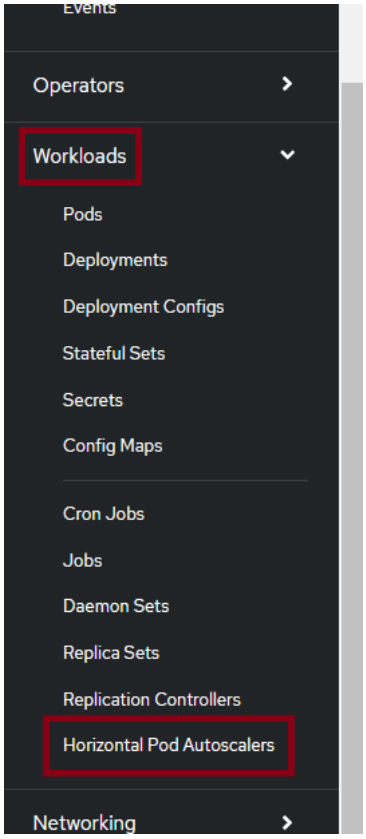
3. Kaydet'e tıklayın.



4. Yöneticilik perspektifine geçin.



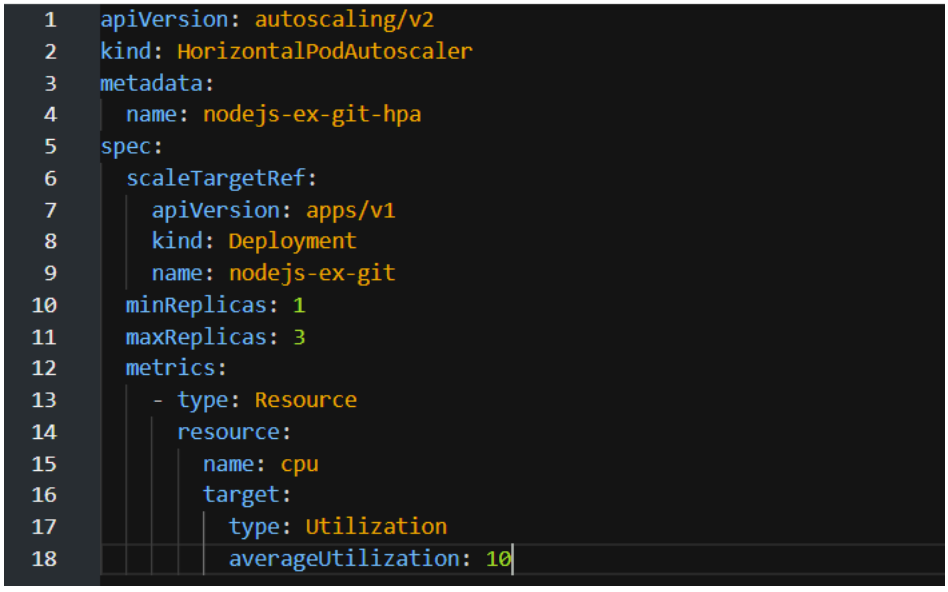
5. İş Yükleri > Yatay Pod Otomatik Ölçekleyicileri'ni seçin.



6. Yatay Pod Otomatik Ölçekleyici Oluştur'a tıklayın.

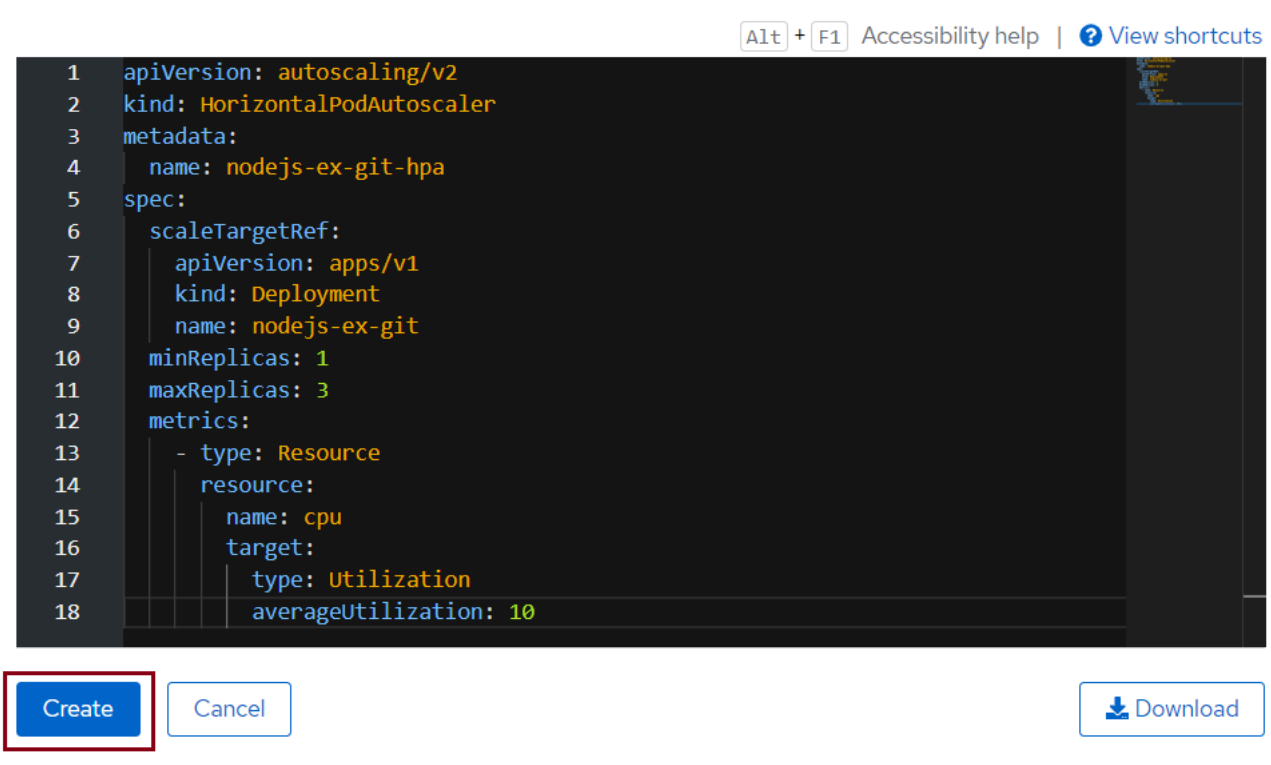
7. Aşağıdaki YAML'ı editöre yapıştırın.

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: nodejs-ex-git-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nodejs-ex-git
  minReplicas: 1
  maxReplicas: 3
  metrics:
    - type: Resource
      resource:
        name: cpu
        target:
          type: Utilization
          averageUtilization: 10
```



Bu HPA, CPU kullanımına dayalı ölçeklendirme yapacağımızı gösteriyor. Genel olarak, CPU kullanımınız %50-90 aralığında olduğunda ölçeklendirme yapmak istersiniz. Bu örnek için, uygulamanın ölçeklendirme gerektirme olasılığını artırmak amacıyla %10 kullanacağız. minReplicas ve maxReplicas alanları, yük durumuna bağlı olarak Dağıtımın her zaman bir ila üç kopya arasında olmasını belirtir.

8. Oluştur'a tıklayın



9. `nodejs-ex-git` üzerindeki yükü artırmak ve Otomatik Ölçeklendirmeyi görüntülemek için Theia'daki terminalde aşağıdaki komutu çalıştırın:

```
for i in $(seq 1000); do curl -s -L <your app URL> & done
```

Not: `<your app URL>` ile önceki bölümdeki 9. Adımda elde ettiğiniz URL'yi değiştirin.

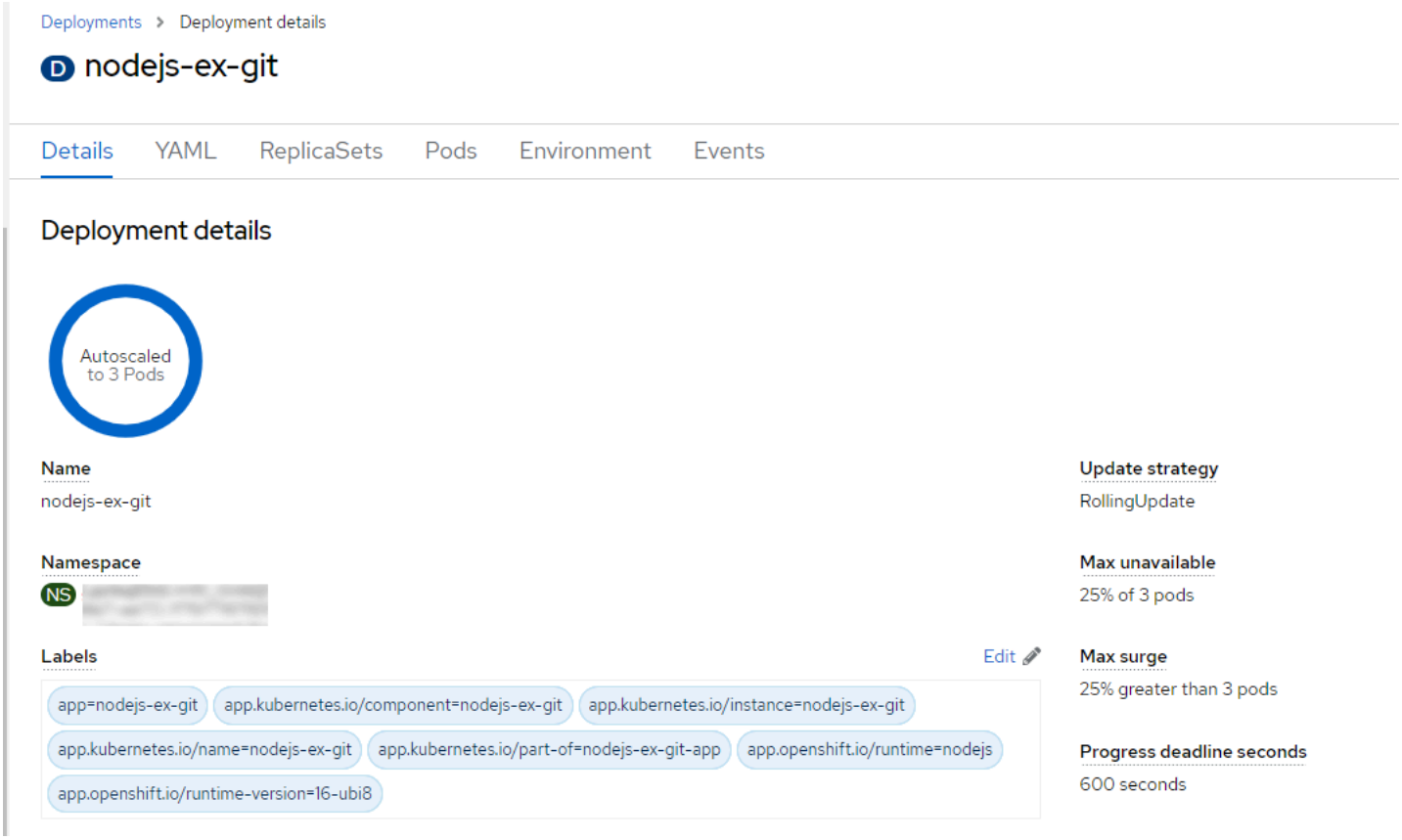
```
theia@theiaopenshift-nikeshkr:/home/project$ for i in $(seq 1000); do curl
labs-nikeshkr.labs-prod-openshift-san-a45631dc5778dc6371c67d206ba9ae5c-0000
loud/ & done
```

Komut, başarılı yük oluşturma işlemini gösteren aşağıdaki çıktıyı vermeye devam edecektir:

Not: Autoscaling'i, tarayıcınızda doğrudan uygulama URL'nizi çalıştırarak da doğrulayabilirsiniz. Uygulama arayüzüne bazı meyve isimleri ve miktarlarını ekleyin ve 'Kaydet' butonuna tıklayın. Eklenen meyveler kullanıcı arayüzünde görünmese de, bu işlem bir yük değişikliği tetikleyecek ve zamanla pod'ların 3'e autoscale olmasına neden olacaktır.

10. `Scale Target` altında `nodejs-ex-git`'e tıklayın.

11. Beklerseniz, Hem Mevcut Replicalar hem de İstenen Replicalar üçe çıkacaktır. Bu, HPA'nın pod'ların maksimum sayısı olan üçe ölçeklenmesi için yeterli yük tespit etmesindendir. Ayrıca Son Ölçekleme Zamanı ile mevcut ve hedef CPU kullanımını da görebilirsiniz. Hedef, belirlediğimiz gibi açıkça %1'dir. Ölçeklenmenin tetiklenmesinin birkaç dakika sürebileceğini unutmayın.



Vay! OpenShift sizin adınıza gerçekten inanılmaz bir iş yaptı. Tek ihtiyacı olan bir kod deposuydu ve bu kodu bir konteyner imajına dönüştürmeyi, o imajı bir kayıt defterine itmeyi, o imajı referans alan bir Deployment oluşturmayı ve uygulamayı bir hostname ile internete açmayı başardı.

Tebrikler! Bu kursun dördüncü modülündeki laboratuvarı tamamladınız.