

Sunucusuz Çerçeve ile Python'da Mikroservisleri Dağıtma, Güncelleme ve Ölçeklendirme



Tahmini Süre: 30 dakika

Bu laboratuvar çalışmasında, iki mikroservisten oluşan bir Birleşik Krallık Üniversitesi bilgi uygulaması oluşturmayı ve her bir mikroservisi diğerinden bağımsız olarak nasıl ölçeklendireceğinizi ve güncelleyeceğinizi öğreneceksiniz. IBM Cloud Code Engine, bu laboratuvar ortamı aracılığıyla sizin için erişilebilir hale getirilmiştir.

Bu uygulama, Birleşik Krallık'taki üniversiteleri filtreleyerek [bu bağlantıdan](#) elde edilen bir JSON dosyası kullanmaktadır.

Öğrenme Hedefleri:

Bu laboratuvarı tamamladıktan sonra şunları yapabileceksiniz:

1. Bir uygulama oluşturmak için Code Engine hizmetini başlatın.
2. Bir uygulamanın parçası olan iki mikroservisi dağıtmak için code engine hizmetini kullanın.
3. Mikroservislerden birini birden fazla örneğe ölçeklendirin.
4. Bir mikroservisi durdurmadan ve yeniden dağıtmadan güncelleyin.

Mikroservislerin Dağıtımı

1. Code engine CLI terminaline gidin. Eğer bir tane yoksa, aşağıya tıklayarak ayarlayın.

▼ Bir tane nasıl ayarlayacağınızı görmek için buraya tıklayın

1. Laboratuvar ortamınızdaki menüde, Cloud açılır menüsüne tıklayın ve Code Engine seçeneğini seçin. Code engine kurulum paneli açılır. Create Project butonuna tıklayın.
2. Code engine ortamının hazırlanması biraz zaman alır. Kurulum panelinde ilerleme durumu gösterilecektir.
3. Code engine kurulumu tamamlandığında, aktif olduğunu göreceksiniz. Aşağıdaki terminalde önceden yapılandırılmış CLI'yi başlatmak için Code Engine CLI butonuna tıklayın.
4. Önceden yapılandırılmış CLI'nin başladığını ve ana dizinin mevcut dizin olarak ayarlandığını göreceksiniz. Ön yapılandırma kapsamında proje ayarlandı ve Kubeconfig yapılandırıldı. Ayrıntılar terminalde gösterilmektedir.

2. Code Engine CLI'de, aşağıdaki komutu çalıştırarak https://github.com/ibm-developer-skills-network/gentu-microservices_practiceProj.git deposunu klonlayın.

```
git clone https://github.com/ibm-developer-skills-network/gentu-microservices_practiceProj.git
```

3. Klonladığımız projedeki universities dizinine geçin.

```
cd gentu-microservices_practiceProj/universities
```

4. İçinde dağıtacağımız mikro hizmetlerin bulunduğu dizinleri görmek için liste.

```
ls
```

İki dizinin listelendiğini göreceksiniz.

- listing
- websites

listing dizini, anahtar kelime aramasına dayalı olarak tüm üniversiteleri veya üniversite listesini listeleyecek mikroservisi içerir.

websites dizini, bir kolej adı verildiğinde web sitelerinin listesini sağlayacak mikroservisi içerir. Daha sonra uygulamayı, bir anahtar kelime aramasına uyan tüm kolejler için web sitelerini döndürmesi için güncelleyeceksiniz.

5. listing mikroservisini dağıtmak için aşağıdaki komutu çalıştırın.

```
ibmcloud ce app create --name listing --image us.icr.io/${SN_ICR_NAMESPACE}/listing --registry-secret icr-secret --port 5000 --build-context-dir listing --build-source .
```

Bu biraz zaman alır.

```
ibmcloud ce app create --name listing --image us.icr.io/${SN_ICR_NAMESPACE}/listing --registry-secret icr-secret --port 5000 --build-context-dir listing --build-source .
Creating application 'listing'...
Packaging files to upload from source path '.'...
Submitting build run 'listing-run-230217-01302675'...
Creating image 'us.icr.io/sn-labs-lavanyas/listing:230217-0630-nw5sj'...
Waiting for build run to complete...
Build run status: 'Running'
Build run completed successfully.
Run 'ibmcloud ce buildrun get -n listing-run-230217-01302675' to check the build run status.
Waiting for application 'listing' to become ready.
Configuration 'listing' is waiting for a Revision to become ready.
Ingress has not yet been reconciled.
Waiting for load balancer to be ready.
Run 'ibmcloud ce application get -n listing' to check the application status.
OK
```

6. Şimdi uygulama hakkında detayları almak için aşağıdaki komutu çalıştırın. Bu, uç noktalarınıza erişebileceğiniz URL'leri de listeleyecektir. URL'yi kopyalayın.

```
ibmcloud ce app get -n listing
```

```
theia@theiadocker-lavanyas:/home/project/gentu-microservices_practiceProj/universities$ ibmcloud ce
application get -n listing
For troubleshooting information visit: https://cloud.ibm.com/docs/codeengine?topic=codeengine-troubleshooting-apps.
Run 'ibmcloud ce application events -n listing' to get the system events of the application instances.
Run 'ibmcloud ce application logs -f -n listing' to follow the logs of the application instances.
OK

Name:          listing
ID:            f2bf0509-d89f-4968-8761-dce35ab2e3ba
Project Name:  Code Engine - sn-labs-lavanyas
Project ID:    1b755f2b-32d2-4aa2-93f5-ce505b849674
Age:          3m37s
Created:       2023-02-17T01:30:30-05:00
URL:           https://listing.z02zthu24x2.us-south.codeengine.appdomain.cloud
Cluster Local URL: http://listing.z02zthu24x2.svc.cluster.local
Console URL:   https://cloud.ibm.com/codeengine/project/us-south/1b755f2b-32d2-4aa2-93f5-ce505b849674/application/listing/configuration
Status Summary: Application deployed successfully
```

7. Şimdi, daha önce kopyaladığımız uç noktaya erişmeyi deneyin ve aşağıdaki komutla değiştirin. Bu komut, adında Trinity geçen tüm kolejleri alacaktır.

```
curl <your deploymenturl>/colleges/Trinity
```

Bu biraz zaman alacak çünkü hiçbir örnek çalışmıyor.

```
theia@theiadocker-lavanyas:/home/project/gentu-microservices_practiceProj/universities$ curl
https://listing.z02zthu24x2.us-south.codeengine.appdomain.cloud/colleges/Trinity
{
  "Colleges": [
    "Trinity College of Music",
    "Trinity College Carmarthen",
    "Leeds Trinity University",
    "Trinity College of Music",
    "Trinity College Carmarthen",
    "Leeds Trinity University"
  ]
}
```

8. Şimdi dağıtımın ayrıntılarını tekrar alın.

```
ibmcloud ce app get --name listing -q
```

Artık örneklerin, uç noktayı eriştiğiniz için listelendiğini görebilirsiniz. Ancak bu örnek, artık talep olmadığında tekrar sonlandırılacaktır. Bunu, 5 dakikalık bir aradan sonra aynı komutu çalıştırarak doğrulayabilirsiniz.

9. Aşağıdaki komutu çalıştırarak `websites`'i bir mikro hizmet olarak dağıtın.

```
ibmcloud ce app create --name websites --image us.icr.io/${SN_ICR_NAMESPACE}/websites --registry-secret icr-secret --port 5000 --build-context-dir websites --build-source .
```

Bu biraz zaman alır ve başarılı bir şekilde dağıtıldığında, uç noktalar aracılığıyla erişebileceğiniz bir URL alacaksınız. URL'yi kopyalayın.

10. Şimdi, daha önce kopyaladığınız uç noktaya erişmeyi deneyin, bunu aşağıdaki komutta değiştirerek yapın. Bu komut, *Trinity College of Music* için tüm web sitelerini alacaktır.

```
curl <your_deployurl>/websites/Trinity%20College%20of%20Music
```

%20, boşluk için URL kodlamasıdır.

11. Web siteleri için anahtar kelime araması yoktur. Sadece kolej adı, geçirilen değerle tam olarak eşleşirse, web siteleri alınır. Daha önce kopyaladığınız uç noktaya, aşağıdaki komutta değiştirilmiş haliyle erişmeyi deneyin. Bu komut hiçbir web sitesi almayacaktır.

```
curl <your_deployurl>/websites/Trinity
```

Hiçbir şey döndürmeyecek.

Uygulamadaki mikro hizmetleri ölçeklendirme ve güncelleme

Monolitik bir uygulamanın aksine, uygulamanın birden fazla mikro hizmetten oluşması çok avantajlıdır. Bu, mikro hizmetleri ölçeklendirmemize ve/veya güncellemelerimizi bağımsız olarak yapmamıza olanak tanır; bu da tüm uygulamayı etkilemeden, yeniden dağıtmak veya yeniden başlatmak zorunda kalmadan mümkündür. Örneğin, bir hizmet çok fazla talep alıyorsa, sadece o hizmeti ölçeklendirebiliriz.

Bu örnekte, iki senaryo hayal edeceğiz:

1. `listings` API uç noktası çok fazla talep alıyor ve ölçeklendirilmesi gerekiyor.
2. Kullanıcı deneyimine dayanarak, `websites` API uç noktasını oluşturduktan sonra, tam kelime araması yerine bir anahtar kelime araması sunmaya karar veriyorsunuz.
3. `listing` uygulamasını üç örneğe ölçeklendirmek için aşağıdaki komutu çalıştırın.

```
ibmcloud ce app update --name listing --min 2
```

Bu, minimum örnek sayısını 2'ye ölçeklendirecektir. Herhangi bir anda, uygulamanın 2 örneği çalışıyor olacak.

2. `listing` uygulamasının ölçeklenip ölçeklenmediğini görmek için aşağıdaki komutu çalıştırın.

```
ibmcloud ce app get --name listing -q
```

Uygulamanın ölçeklenmesini ve tüm örneklerin çalışmasını bekleyin.

3. Artık `curl` komutunu çalıştırdığımızda, uç noktaya erişim çok daha hızlı olacaktır çünkü örnek zaten 2 örnek çalıştırıyor.

Mikroservisi Güncelleme

Artık `websites` API Uç Noktasını, tüm üniversite adını vermek yerine anahtar kelime araması parametresi alacak şekilde güncelleyeceksiniz.

1. Aşağıdaki düğmeye tıklayarak `websites` dizinindeki `app.py` dosyasını açın.

Open **app.py** in IDE

2. API uç noktasının uygulamasını aşağıdaki kod ile değiştirin. Bu, mikroservisin web siteleri için anahtar kelime aramaları sağlamasını sağlayacaktır.

Güncellenmiş kod aşağıdaki gibi olmalıdır.

```
from flask import Flask
from flask_cors import CORS
import json
app = Flask("University Websites")
CORS(app)
with open("UK_Universities.json", "r") as unfile:
    data = json.load(unfile)
@app.route("/websites/<name>")
def getCollegesWebsites(name):
    webpages = []
    for college in data:
        if college["name"].__contains__(name):
            for website in college["web_pages"]:
                webpages.append(website)
    return json.dumps({"webpages": webpages},indent=4)
if __name__ == "__main__":
    app.run(debug=True, port=5000)
```

3. Uygulamayı güncellemek için aşağıdaki komutu çalıştırın.

```
ibmcloud ce app update --name websites --image us.icr.io/${SN_ICR_NAMESPACE}/websites --registry-secret icr-secret --port 5000 --build-context-dir websites --build-source .
```

4. Bu, mevcut uygulamayı güncelleyecektir. Bunu doğrulamak için aşağıdaki komutu çalıştırın. Önceki alıştırmadan hatırlayabilirsiniz ki, bu herhangi bir web sitesi döndürmedi çünkü önceki kod anahtar kelime aramasını desteklemiyordu.

```
curl <your_deployementurl>/websites/Trinity
```

Tüm adında Trinity geçen kolejlerin web sitelerini göreceksiniz.

Uygulama Egzersizi:

1. web siteleri mikroservisini en az 2 örneğe ölçeklendirin.

NOT: Web sitesi hizmetini ölçeklendirmeden önce, hem listeleme hem de web sitesi hizmetleri için mevcut çalışan örnek sayısını kontrol edin. Eğer 3–4 örnek çalışıyorsa, aktif istek olmadığına bu örneklerin sona ermesi için yaklaşık 5 dakika bekleyin. 5 dakikalık beklemenin ardından aşağıdaki komutu çalıştırarak bunu doğrulayabilirsiniz. Yeni örnekler için yeterli bellek sağlamak amacıyla, yalnızca 2 veya daha az örnek olduğunda ölçeklendirmeye başlayın.

```
ibmcloud ce app get --name websites -q
```

2. listings mikro hizmetlerini, üniversitelerin listesini büyük/küçük harf duyarlı bir şekilde almak için güncelleyin. Aynı zamanda “TriniTY” parametresi ile doğrulayın. Başarıyla güncellenirse, uç nokta ‘Trinity’ kelimesini içeren tüm üniversiteleri döndürmelidir.

Tebrikler! Bu laboratuvarı başarıyla tamamladınız ve iki mikro hizmetten oluşan uygulamanızı Code Engine üzerinde dağıttınız, bir mikro hizmeti ölçeklendirdiniz ve diğer mikro hizmeti güncellediniz.

Yazar(lar)

Lavanya T S

© IBM Corporation. Tüm hakları saklıdır.