

Statik Bir Hugo Web Sitesini AWS S3 üzerine TravisCI ile Yayınlamak

2018-03-24T20:12:05+03:00

Contents

Neden Hugo	1
1. Develop	2
2. Config	3
Deploy & Run	3

Bu yazımda Hugo ile oluşturulmuş bir web sitesini, CI/CD aracı olan TravisCI ile AWS S3 üzerine yayınlayacağız.

Neden Hugo

Şimdiye kadar onlarca araç denedim, kullandım. Sunucu tarafı olarak Drupal 7'yi uzun süre kullandım, siteyi statik şekilde arşivledim ve kenara aldım. Onlarca statik web sitesi aracı denedim, Jekyll, Octopress, Pelikan, Lektor ve bir aklıma gelen gelmeyen bir çok araç. Zilyon adet çözüm mevcut. Bu tür araçları kullanırken gördüğüm en büyük sorunlar:

Sunucu tarafı olanların canlı sistem olmaları, bakıma ve izlenmeye ihtiyaç duymaları, kaynak tüketmeleri, güvenlik açıkları, veri tabanı bağımlılıkları. . .

Şu anda aklıma gelen taslakları, yazılım geliştirirken karşılaştığım "pis kokan" kod parçalarını markdown ile rahatça yazabildiğim Heroku üzerinde çalışan bir Django uygulaması mevcut.

İşi profesyonel bloggerlık olmayan ve bakmakla yükümlü olduğu diğer "canlı" web uygulamaları olan bizler için statik web sitelerini çok daha sevimli bulanlardanım.

Hikaye kısmını burada bırakıp sürecimizi üç aşamaya ayıralım:

1. **Develop:** Hugo'nun kurulması ve lokal geliştirme ortamının oluşturulmasını anlatacağız.
2. **Config:** TravisCI ve AWS S3 ayarlarının yapılmasını burada anlatacağız.

3. **Deploy & Run:** TravisCI ile web sitemizi nasıl yayınlayacağımızı anlatacağız.

İhtiyacımız olanlar:

1. TravisCI hesabı
2. AWS hesabı

1. Develop

GoHugo > Installing adresinden işletim sisteminize uygun ve kolayınıza gelen yöntemlerden biri ile komut satırı aracını kurmanız gerekiyor.

Debian tabanlı bir işletim sisteminde kurulum çok basit:

```
wget https://github.com/gohugoio/hugo/releases/download/v0.37.1/hugo_0.37.1_Linux-64bit.deb
sudo dpkg -i hugo_0.37.1_Linux-64bit.deb
```

Quick Start kısmını takip ederek iki dakika içinde ilk içeriğinizi yazmaya başlayabilirsiniz.

Hugo ile henüz yayınlanmasını istemediğiniz içerikleri taslak olarak kaydedebilir, veya ileri bir yayınlanma tarihi verebilirsiniz. Böylece açıkça belirtmediğiniz takdirde taslaklarımız veya ileri tarihli yazılarımız yayınlanmamış olur.

Yazılarınızı yazarken taslak aşamasından olan yazılarınızı da görmek isteyebilirsiniz.

Bunu Hugo'ya belirtmemiz gerekiyor. **Makefile** kullanmayı seviyorum. **make** dediğinizde <http://127.0.0.1:1313> adresindeki web sunucusuyla sitenizin önizlemesini görebilirsiniz.

```
default: develop

DEVELOP := hugo \
  --watch serve \
  --destination /tmp/blog_dev \
  --buildDrafts \
  --buildFuture \
  --baseURL=127.0.0.1

develop:
  $(DEVELOP)

.PHONY: default develop
```

Artık Github'da web siteniz için yeni bir repo oluşturabilir ve değişikliklerinizi *push* edebilirsiniz.

2. Config

S3 Bucket Oluşturma ve İzinleri Ayarlama S3 Console adresinden yeni bir bucket oluşturalım. *Web sitesi olarak kullanacak ve özel bir alan adı üzerinden yayın yapmayı düşünüyorsanız, foo.example.com isminde bucket açmanız gerekiyor.* Bkz. S3 VirtualHosting

IAM servisi ile travisci için bir kullanıcı oluşturalım ve oluşturulan *Access Key* ve *Access Secret* değerlerini kaydedelim. Kaydetmediğiniz takdirde her ihtiyacınız olduğunda yeni bir *Security Credential* oluşturmanız gerekecek.

Bucket ve kullanıcıyı oluşturduk. TravisCI üzerinden S3'e dosyalarımızı yükleyecek bu kullanıcıya gerekli izinleri vermek için

Add Permission > Attach existing policies directly > Create Policy > JSON yolunu takip ederek

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObjectAcl",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "s3:DeleteObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::foo.example.com/*"
      ]
    }
  ]
}
```

gerekli izinleri veriyoruz.

Deploy & Run

Hugo web sitemizi çalışır hale getirdik, S3 bucket oluşturup izinleri ayarladık. TravisCI ile Hugo web sitemizi derleyip, S3 üzerine yayımlayabiliriz.

Öncelikle reponun ana dizinine `.travis.yml` adında bir dosya eklememiz gerekiyor.

```
{{< highlight yaml "linenos=table,hl_lines=,linenostart=1" >}} sudo: required
dist: trusty
```

```
before_install: - wget https://github.com/gohugoio/hugo/releases/download/v0.37.1/hugo_0.37.1_Linux-64bit.deb && sudo dpkg -i hugo_0.37.1_Linux-64bit.deb
```

```
script: - hugo --theme=paperback
```

```
deploy: provider: s3 on: master skip_cleanup: true access_key_id: $S3_KEY secret_access_key: $S3_SECRET bucket: $S3_BUCKET region: $S3_REGION acl: public-read local_dir: public {{< / highlight >}}
```

Yirmi satırdan az bir konfigürasyon dosyası ile sitemizi yayımlayabiliyoruz.

İlk beş satırda hugonun kurulumunu yapıyoruz.

```
{{< highlight yaml "linenos=table,hl_lines=,linenostart=1" >}} sudo: required
dist: trusty
```

```
before_install: - wget https://github.com/gohugoio/hugo/releases/download/v0.37.1/hugo_0.37.1_Linux-64bit.deb && sudo dpkg -i hugo_0.37.1_Linux-64bit.deb
```

```
{{< / highlight >}}
```

Hugonun `public` dizini altına `paperback` dizini altına web sitemizi oluşturmasını sağlıyoruz. Temayı buradan vermek zorunda değilsiniz. `config.toml` dosyanızda `theme = "paperback"` ile temayı tanımlayıp `hugo` komutunu parametresiz çalıştırmanız yeterli.

```
{{< highlight yaml "linenos=table,hl_lines=,linenostart=7" >}} script: - hugo --theme=paperback {{< / highlight >}}
```

Bu satırları tek tek açıklayalım.

```
{{< highlight yaml "linenos=table,hl_lines=,linenostart=10" >}} deploy: provider: s3 on: master skip_cleanup: true access_key_id: $S3_KEY secret_access_key: $S3_SECRET bucket: $S3_BUCKET region: $S3_REGION acl: public-read local_dir: public {{< / highlight >}}
```

satır 12, sadece master branch için deployment yapılacağını bildiriyor. Başka ifadeyle `develop` üzerinden derleme tetiği verdiğinizde bu kısım dikkate alınmayacaktır.

satır 14-17, başlarında `$` olan ifadeler, bu ifadelerin TravisCI üzerinden tanımlanan *Environment Variable* olduğunu bildiriyor.

satır 18, yüklenecek yeni dosyaların herkes tarafından görülebilmesi için S3 üzerine yüklenen dosyaların izinleri *anonim* olarak ayarlanıyor.

satır 19, oluşturulan dosyaların hangi klasörde olduğunu ve deployment'ın bu klasörden yapılmasını gerektiğini bildiriyor.