REST API

"Analiz" bölümünde belirtildiği gibi, Cuckoo, Flask kullanılarak uygulanan basit ve hafif bir REST API sunucusu sağlar.

- API Server'ı Başlatma

API Server’ı Başlatma

API sunucusunu başlatmak için şu komutu kullanabilirsiniz:

```
$ cuckoo api
```

Varsayılan olarak, servisi localhost:8090 adresinde bağlar. Bu değerleri değiştirmek istiyorsanız, aşağıdaki sözdizimini kullanabilirsiniz:

```
$ cuckoo api --host 0.0.0.0 --port 1337
$ cuckoo api -H 0.0.0.0 -p 1337
```

API'ya yalnızca kimlik doğrulamış erişime izin vermek için cuckoo.conf'daki api_token değeri gizli bir değere ayarlanmalıdır. Yeni Cuckoo yüklemelerinde, sizin için otomatik olarak rastgele bir belirteç oluşturulur. API'ye erişmek için tüm isteklerinizi yapılandırmadaki belirteç kullanılarak Authorization: Bearer <token> başlığını göndermeniz gerekmektedir. API'ye güvensiz bir ağ üzerinden, örneğin İnternet üzerinden erişmek istiyorsanız, API sunucusunu aşağıdaki bölümde açıklanan nginx'in arkasında çalıştırmalı ve HTTPS'yi etkinleştirmelisiniz.

Web Dağıtımı

API sunucusunu başlatmanın varsayılan yöntemi birçok durum için iyi çalışsa da, bazı kullanıcılar sunucuyu güvenilir bir şekilde dağıtmak isteyebilir. Bu, API'yi bir web sunucusu aracılığıyla WSGI uygulaması olarak açığa çıkarmak suretiyle yapılabilir. Bu bölüm, API'nin uWSGI ve nginx üzerinden nasıl dağıtılacağını basit bir örneği göstermektedir. Bu talimatlar Ubuntu GNU/Linux göz önüne alınarak yazılmış olup, diğer platformlara uyarlanabilir.

Bu çözüm, uWSGI, uWSGI Python eklentisi ve nginx gerektirir. Bunlar paket olarak mevcuttur:

```
$ sudo apt-get install uwsgi uwsgi-plugin-python nginx
```

uWSGI Kurulumu

İlk olarak, API sunucusunu bir uygulama olarak çalıştırmak için uWSGI'yi kullanın.

Başlamak için, `cuckoo api --uwsgi` komutu tarafından rapor edilen gerçek yapılandırmayı içeren `/etc/uwsgi/apps-available/cuckoo-api.ini` adlı bir uWSGI yapılandırma dosyası oluşturun:

```
$ cuckoo api --uwsgi
[uwsgi]
plugins = python
virtualenv = /home/cuckoo/cuckoo
module = cuckoo.apps.api
callable = app
uid = cuckoo
gid = cuckoo
env = CUCKOO_APP=api
env = CUCKOO_CWD=/home/..somepath..
```

Bu yapılandırma, dağıtımın varsayılan uWSGI yapılandırmasından bir dizi ayar devralır ve gerçek işi yapmak için Cuckoo paketinden `cuckoo.apps.api`'yi içeri alır. Bu örnekte Cuckoo'yu `/home/cuckoo/cuckoo` konumundaki bir sanal ortamda yükledik. Cuckoo global olarak yüklendiyse, sanal ortam seçeneği gerekli değildir.

Uygulama yapılandırmasını etkinleştirin ve sunucuyu başlatın.

```
$ sudo ln -s /etc/uwsgi/apps-available/cuckoo-api.ini /etc/uwsgi/apps-enabled/
$ sudo service uwsgi start cuckoo-api     # or reload, if already running
```

> Uygulama için günlük dosyaları, dağıtım uygulama örnekleri için standart dizinde bulunabilir, yani `/var/log/uwsgi/app/cuckoo-api.log`. UNIX soketi de geleneksel bir konumda oluşturulur, yani `/run/uwsgi/app/cuckoo-api/socket`.

nginx Kurulumu

API sunucusunun uWSGI ile çalıştığı bir ortamda, nginx şimdi ona HTTP isteklerini yönlendirmek üzere bir web sunucusu/ters proxy olarak kurulabilir.

Başlamak için, `cuckoo api --nginx` komutu tarafından raporlanan gerçek yapılandırmayı içeren `/etc/nginx/sites-available/cuckoo-api` adlı bir nginx yapılandırma dosyası oluşturun:

```
$ cuckoo api --nginx
upstream _uwsgi_cuckoo_api {
    server unix:/run/uwsgi/app/cuckoo-api/socket;
```

```
}

server {
    listen localhost:8090;

    # REST API app
    location / {
        client_max_body_size 1G;
        uwsgi_pass  _uwsgi_cuckoo_api;
        include     uwsgi_params;
    }
}
```

Nginx'in uWSGI soketiyle iletişim kurabilmesi için, kullanıcısını cuckoo grubuna ekleyerek emin olun:

```
$ sudo adduser www-data cuckoo
```

Sunucu konfigürasyonunu etkinleştirin ve sunucuyu başlatın.

```
$ sudo ln -s /etc/nginx/sites-available/cuckoo-api /etc/nginx/sites-enabled/
$ sudo service nginx start    # or reload, if already running
```

Bu aşamada API sunucusu, sunucuda 8090 numaralı bağlantı noktasında kullanılabilir olmalıdır. Sunucu performansını ayarlamak, kimlik doğrulama eklemek veya HTTPS kullanarak iletişimi güvenli hale getirmek gibi bu yapıyı genişletmek için çeşitli konfigürasyonlar uygulanabilir.