# Hızlı Kullanım

Pratik kullanım için aşağıdaki birkaç komut işinizi görecektir.

Bir cuckoo node ve aynı makinede çalışan bir Cuckoo API'si kaydedin:

```
$ curl http://localhost:9003/api/node -F name=localhost -F ip=127.0.0.1
```

Cuckoo node devre dışı bırakmak için:

```
$ curl -XDELETE http://localhost:9003/api/node/localhost
```

Herhangi özel gereksinim olmadan yeni bir analiz görevi gönderin (örneğin, Cuckoo etiketleri, belirli bir makine kullanma, vb.).

```
$ curl http://localhost:9003/api/task -F file=@/path/to/sample.exe
```

Tamamlanmış bir görevin raporunu alın (eğer tamamlanmamışsa, 420 kodlu bir hata alırsınız). Aşağıdaki örnek, varsayılan olarak JSON raporunu kullanacaktır:

```
$ curl http://localhost:9003/api/report/1
```

Bir Cuckoo node takılırsa ve sıfırlanması gerekiyorsa, aşağıdaki adımlar temiz bir şekilde yeniden başlatmak için gerçekleştirilebilir. Bu, SaltStack yapılandırmasının kullanımını ve bazı manuel SQL komutlarını gerektirir (ve tercihen Distibuted Cuckoo Worker geçici olarak devre dışı bırakılmış olmalıdır, yani `supervisorctl stop distributed`):

```
$ psql -c "UPDATE task SET status = 'pending' WHERE status = 'processing' AND
node_id = 123"
$ salt cuckoo1 state.apply cuckoo.clean
$ salt cuckoo1 state.apply cuckoo.start
```

Eğer tüm Cuckoo kümesi bir şekilde kilitlendi, yani tüm görevler 'atanmış', 'işleniyor' veya 'tamamlandı' durumundayken hiçbir Cuckoo node söz konusu analizleri şu anda çalıştırmıyorsa (örneğin, birçok sıfırlama nedeniyle), o zaman tüm durumu sıfırlamak için aşağıdaki adımlar kullanılabilir:

```
$ supervisorctl -c ~/.cuckoo/supervisord.conf stop distributed
$ salt '*' state.apply cuckoo.stop
$ salt '*' state.apply cuckoo.clean
$ psql -c "UPDATE task SET status = 'pending', node_id = null WHERE status IN
('assigned', 'processing', 'finished')"
$ salt '*' state.apply cuckoo.start
$ supervisorctl -c ~/.cuckoo/supervisord.conf start distributed
```

Eğer bir Cuckoo node üzerinde işlenemeyen bir dizi görev varsa ve bu nedenle Cuckoo node tamamen kilitlemişse, o zaman Cuckoo örneklerini hata düzeltilmiş bir sürümle güncelleyip tüm analizleri yeniden işlemek işe yarayabilir:

```
$ salt cuckoo1 state.apply cuckoo.update  # Upgrade Cuckoo.
# To make sure there are failed analyses in the first place.
$ salt cuckoo1 cmd.run "sudo -u cuckoo psql -c \"SELECT * FROM tasks WHERE
status = 'failed_processing'\"
# Reset each analyses to be re-processed.
$ salt cuckoo1 cmd.run "sudo -u cuckoo psql -c \"UPDATE tasks SET status =
'completed', processing = null WHERE status = 'failed_processing'\""
```

Distributed Cuckoo ana bilgisayarını yükseltmek için aşağıdaki adımları gerçekleştirmek isteyebilirsiniz:

```
$ /etc/init.d/uwsgi stop
$ supervisorctl -c ~/.cuckoo/supervisord.conf stop distributed
$ pip uninstall -y cuckoo
$ pip install cuckoo==2.0.0         # Specify your version here.
$ pip install Cuckoo-2.0.0.tar.gz   # Or use a locally archived build.
$ cuckoo distributed migrate
$ supervisorctl -c ~/.cuckoo/supervisord.conf start distributed
$ /etc/init.d/uwsgi start
$ /etc/init.d/nginx restart
```

Tüm Cuckoo kümenizi, yani her Cuckoo nodedaki her makineyi test etmek için, örnek alınabilecek bir `stuff/distributed/cluster-test.py` betiği bulunmaktadır. Mevcut haliyle, kümedeki her yapılandırılmış

makinede aktif bir internet bağlantısını kontrol etmenize olanak tanır. Bu betik, hatalı veya bir şekilde bozulmuş makineleri belirlemek için kullanılabilir. Örnek kullanım aşağıdaki gibi olabilir:

```
# Assuming Distributed Cuckoo listens on localhost and that you want to
# run the 'internet' script (see also the source of cluster-test.py).
$ python stuff/distributed/cluster-test.py localhost -s internet
```

---

Revision #1
Created 18 January 2024 21:30:51 by Ertan Sözer
Updated 18 January 2024 21:33:51 by Ertan Sözer