**Project Submission Report**

**Cassandra installation on AWS:**

* [**https://dev9.com/blog-posts/2016/6/introduction-to-kong-api-gateway**](https://dev9.com/blog-posts/2016/6/introduction-to-kong-api-gateway)

**Step 1: Setup the JAVA environment**

##### sudo add-apt-repository ppa:webupd8team/java

##### sudo apt-get update

##### sudo apt-get install oracle-java8-installer

sudo apt install oracle-java8-set-default

**Step 2: Setup the KONG environment**

Installing the KONG:

<https://getkong.org/install/ubuntu/>

**Install the Package:**

**If you are downloading the** [**package**](https://getkong.org/install/ubuntu/#packages)**, execute:**

sudo apt-get update  
sudo apt-get install openssl libpcre3 procps perl

**Download Kong debian package at below link:**

<https://github.com/Mashape/kong/releases/download/0.10.1/kong-0.10.1.xenial_all.deb>

Command:

wget use\_the\_above\_link\_to\_download

sudo dpkg -i kong-0.10.1.\*.deb

**Configure your database**

[**Configure**](https://getkong.org/docs/0.10.x/configuration#database) **Kong so it can connect to your database. Kong supports both** [**PostgreSQL 9.4+**](http://www.postgresql.org/) **and** [**Cassandra 3.x.x**](http://cassandra.apache.org/) **as its datastore.**

**If you are using Postgres, please provision a database and a user before starting Kong, ie:**

CREATE USER kong; CREATE DATABASE kong OWNER kong;

**Step 3: Install the cassandra**

### Installation from Debian packages

* For tick-tock releases, the <release series> is the release number, without dot, and with an appended x, so 31x, 32x, …
* For older pre-tick-tock releases, the <release series> is the major version number, without dot, and with an appended x. So currently it can one of 21x, 22x or 30x.
* Add the Apache repository of Cassandra to /etc/apt/sources.list.d/cassandra.sources.list, for example for version 3.10:

echo "deb http://www.apache.org/dist/cassandra/debian 310x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list

* Add the Apache Cassandra repository keys:

curl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key add -

* Update the repositories:

sudo apt-get update

* If you encounter this error:

GPG error: http://www.apache.org 310x InRelease: The following signatures couldn't be verified because the public key is not available: NO\_PUBKEY A278B781FE4B2BDA

Then add the public key A278B781FE4B2BDA as follows:

sudo apt-key adv --keyserver pool.sks-keyservers.net --recv-key A278B781FE4B2BDA

and repeat sudo apt-get update. The actual key may be different, you get it from the error message itself. For a full list of Apache contributors public keys, you can refer to <https://www.apache.org/dist/cassandra/KEYS>.

* Install Cassandra:

sudo apt-get install cassandra

* You can start Cassandra with sudo service cassandra start and stop it with sudo service cassandra stop. However, normally the service will start automatically. For this reason be sure to stop it if you need to make any configuration changes.
* Verify that Cassandra is running by invoking nodetool status from the command line.
* The default location of configuration files is /etc/cassandra.
* The default location of log and data directories is /var/log/cassandra/ and /var/lib/cassandra.
* Start-up options (heap size, etc) can be configured in /etc/default/cassandra.

**Start Kong:**

$ kong start  
# Kong is running  
$ curl 127.0.0.1:8001

**If you get error while starting kong try this option and follow following steps.**

kong start --v

Error:

2017/04/10 22:24:18 [verbose] Kong: 0.10.1

2017/04/10 22:24:18 [verbose] no config file found at /etc/kong/kong.conf

2017/04/10 22:24:18 [verbose] no config file found at /etc/kong.conf

2017/04/10 22:24:18 [verbose] no config file, skipping loading

2017/04/10 22:24:18 [verbose] prefix in use: /usr/local/kong

2017/04/10 22:24:18 [verbose] preparing nginx prefix directory at /usr/local/kong

2017/04/10 22:24:18 [verbose] saving serf identifier to /usr/local/kong/serf/serf.id

2017/04/10 22:24:18 [verbose] saving serf shell script handler to /usr/local/kong/serf/serf\_event.sh

2017/04/10 22:24:18 [verbose] SSL enabled, no custom certificate set: using default certificate

2017/04/10 22:24:18 [verbose] default SSL certificate found at /usr/local/kong/ssl/kong-default.crt

2017/04/10 22:24:18 [verbose] Admin SSL enabled, no custom certificate set: using default certificate

2017/04/10 22:24:18 [verbose] admin SSL certificate found at /usr/local/kong/ssl/admin-kong-default.crt

2017/04/10 22:24:18 [warn] ulimit is currently set to "1024". For better performance set it to at least "4096" using "ulimit -n"

2017/04/10 22:24:18 [verbose] running datastore migrations

2017/04/10 22:24:18 [verbose] could not start Kong, stopping services

2017/04/10 22:24:18 [verbose] leaving serf cluster

2017/04/10 22:24:18 [verbose] stopped services

Error:

/usr/local/share/lua/5.1/kong/cmd/start.lua:34: /usr/local/share/lua/5.1/kong/cmd/start.lua:21: [postgres error] could not get current migrations: [postgres error] connection refused

stack traceback:

[C]: in function 'error'

/usr/local/share/lua/5.1/kong/cmd/start.lua:34: in function 'cmd\_exec'

/usr/local/share/lua/5.1/kong/cmd/init.lua:88: in function </usr/local/share/lua/5.1/kong/cmd/init.lua:88>

[C]: in function 'xpcall'

/usr/local/share/lua/5.1/kong/cmd/init.lua:88: in function </usr/local/share/lua/5.1/kong/cmd/init.lua:45>

/usr/local/bin/kong:5: in function 'file\_gen'

init\_worker\_by\_lua:38: in function <init\_worker\_by\_lua:36>

[C]: in function 'pcall'

init\_worker\_by\_lua:45: in function <init\_worker\_by\_lua:43>

**Use the below link to resolve the above error:**

<https://getkong.org/docs/0.9.x/configuration/>

Sudo cp /etc/kong/kong.conf.default /etc/kong/kong.conf

Sudo cd /etc/kong

Sudo vi kong.conf

**Now make the following changes in the setting (kong.conf file)**

**Change the database = cassandra from database = postgresql**

**And uncomment all thr properties related to the cassandra**

**Under the**

**#------------------------------------------------------------------------------**

**# DATASTORE**

**#------------------------------------------------------------------------------**

**# Kong will store all of its data (such as APIs, consumers and plugins) in**

**# either Cassandra or PostgreSQL.**

**#**

**# All Kong nodes belonging to the same cluster must connect themselves to the**

**# same database.**

**#database = cassandra # Determines which of PostgreSQL or Cassandra**

**# this node will use as its datastore.**

**# Accepted values are `postgres` and**

**# `cassandra`.**

**Use Kong:**

Quickly learn how to use Kong with the [5-minute Quickstart](https://getkong.org/docs/latest/getting-started/quickstart).

**Start Kong:**

$ kong start  
# Kong is running  
$ curl 127.0.0.1:8001

**Now verify it by logging in th cassandra command line.**

Cqlsh <private ip of the instance>

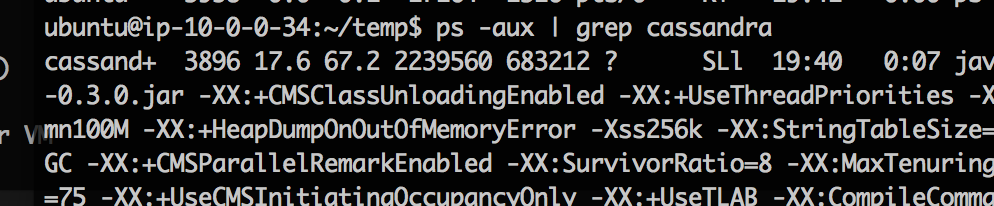
cqlsh> DESCRIBE KEYSPACES;

It should show ‘kong’ as one of the the key spaces.

Once the kong has started now we need to setup the cassandra cluster but for that we need to first stop all the running cassandra clusters and clean up the directories.

ps -aux | grep cassandra

and you should see something like:



Notice the PID is 3896 in my example.With whatever your PIDis,run:

sudo kill 3896

And let's delete the data:

sudo rm -rf /var/lib/cassandra/\*

**Step 4: Create a cassandra cluster**

<http://ealfonso.com/setting-up-a-cassandra-cluster-on-awsubuntu14-04/>

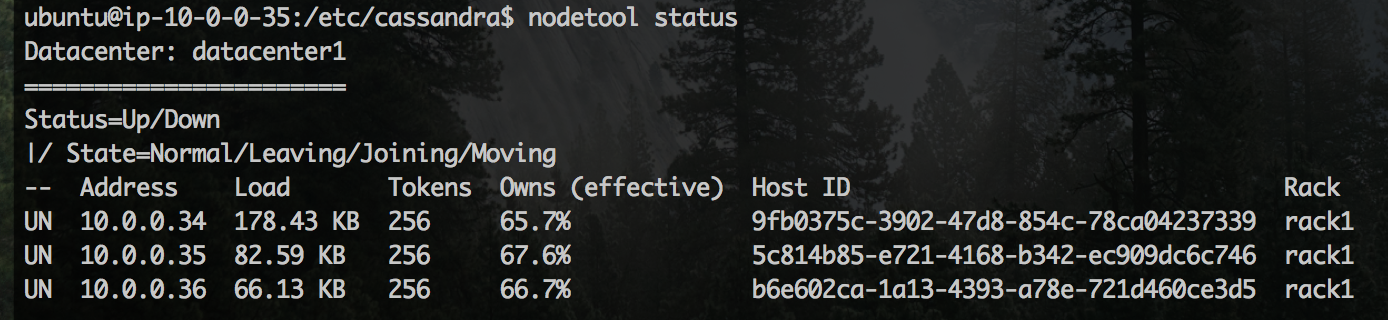
* Since we're launching a three node cluster we really only need to have one seed node that the other two nodes rely on. For now, ssh into whichever server you want the seed server to be and let's edit **/etc/cassandra/cassandra.yaml**
* **sudo nano /etc/cassandra/cassandra.yaml**
* Find the follow configuration settings: (This setting will work only if all instances are created in the same VPC in AWS)
* **cluster\_name: "<name of your cluster>"  
  ...  
  seed\_provider:   
   - class\_name: org.apache.cassandra.locator.SimpleSeedProvider  
   parameters:  
   - seeds: "<private ip of seed server>" (Put the list ip’s of all three nodes Ex. “<ip1>,<ip2>,<ip3>”)  
  ...  
  listen\_address: <private ip of current server>  
  ...  
  rpc\_address: <private ip of current server>**
* When completed, let's run Cassandra:

**sudo cassandra &**

* The & is so we run Cassandra in the background.
* Now, repeat these steps with the remaining nodes. When you are finished and you've launched Cassandra on all of the nodes (without errors!) you can run:

**nodetool status**

* This outputs the status of the cluster, and should look like this:



This means all your nodes are connected to form a cluster.

**Step 5: Migrating the Kong Schema**

As we have already setup the kong and cassandra our priviously created kong schema will be washed off so we need to migrate it again. You shoud do this on only once instance and should be update d on all other schema automatically.

Use the following command for doing that.

sudo kong migrations up

**Now verify it by logging in th cassandra command line.**

Cqlsh <private ip of the instance>

cqlsh> DESCRIBE KEYSPACES;

It should show ‘kong’ as one of the the key spaces.

**Step 6: Migrating the Kong Schema**

Sudo apt-get install npm

**Step 7: Installing NodeJS-Legacy**

Sudo apt-get install nodejs-legacy

**Step 8: Migrating the Kong Schema**

### **With Npm**

# Install Kong Dashboard  
npm install -g kong-dashboard  
  
# Start Kong Dashboard  
kong-dashboard start  
  
# To start Kong Dashboard on a custom port  
kong-dashboard start -p [port]  
  
# To start Kong Dashboard with basic auth  
kong-dashboard start -a user=password  
  
# You can set basic auth user with environment variables  
# Do not set -a parameter or this will be overwritten  
set kong-dashboard-name=admin && set kong-dashboard-pass=password && kong-dashboard start

**Step 8: Open following port on your AWS instance**

* **HTTP - 80**
* **HTTPS - 443**
* **Custom TCP Rule - 9024**
* **Custom TCP Rule - 8000**
* **Custom TCP Rule - 8001**
* **Custom TCP Rule - 7199**
* **SSH - 22**
* **Custom TCP Rule - 7000 - 7001**
* **Custom TCP Rule - 9160**

**Also open all those ports as well on which you api’s are running such as**

**8080**

**Step 9: Create Admin API**

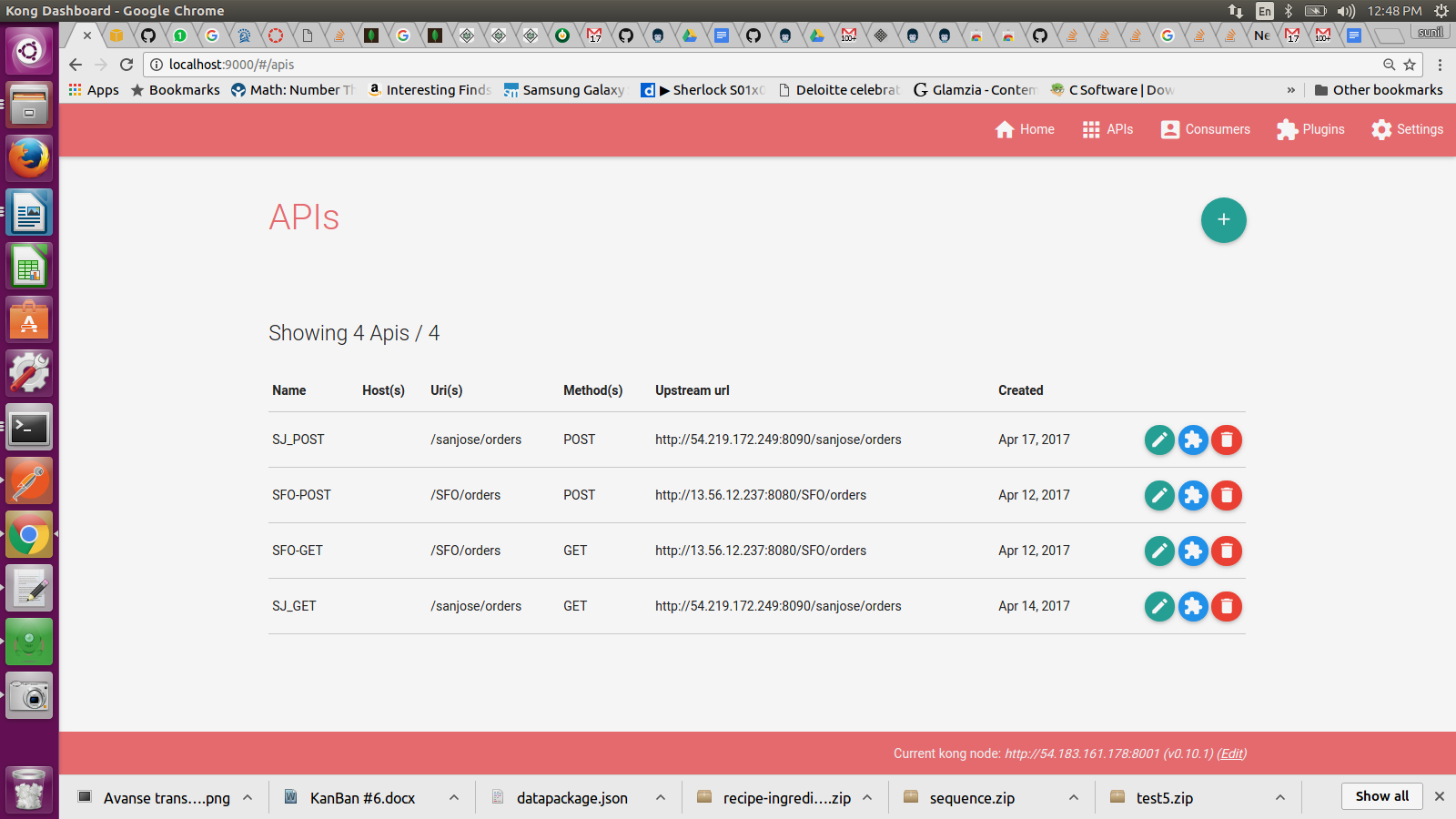
curl -i -X POST \

--url http://54.183.161.178:8001/apis/ \

--data 'name=node-api' \

--data 'hosts=ec2-54-193-32-159.us-west-1.compute.amazonaws.com' \

--data 'upstream\_url=http://54.193.32.159:8080/SFO/orders'



**Step 10: Route your request through Admin API**

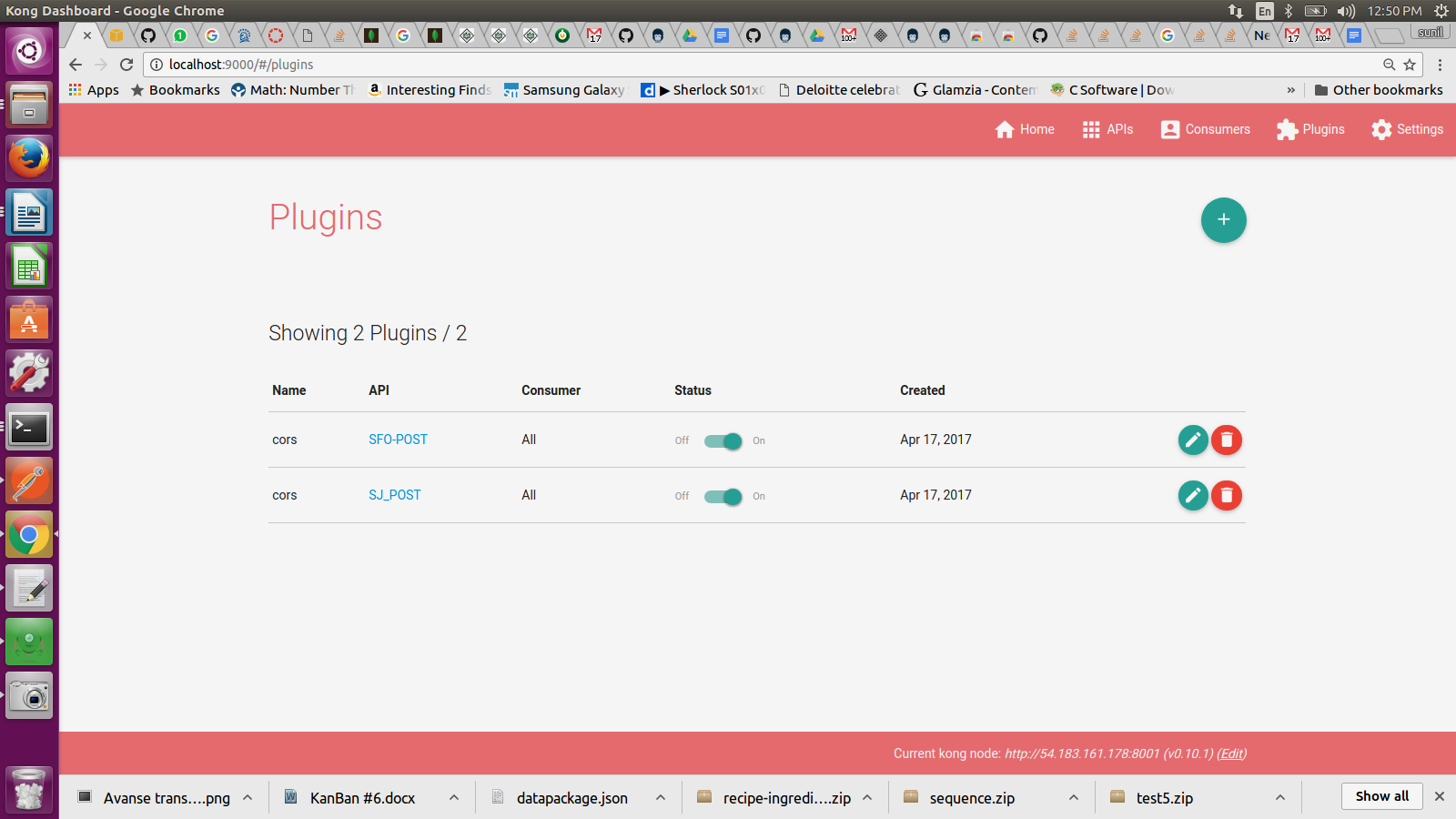
curl -i -X GET \

--url http://54.183.161.178:8000/ \

--header 'Host: ec2-54-193-32-159.us-west-1.compute.amazonaws.com'

**Step 11: Adding the API in KONG Dashboard**

CORS Plugin:



**Which API(s) should this plugin apply to?**

SFO-POST

**Plugin:** Cors

**Origins:** \*

**Exposed headers:** X-Auth-Token

**Methods:** GET,HEAD,OPTIONS,POST,PUT,DELETE

**Max age:** 3600

**Headers:**

Access-Control-Allow-Headers,Origin,Accept,X-Requested-With,Content-Type,Access-Control-Request-Method,Access-Control-Request-Headers,Accept-Version,Content-Length,Content-MD5,Date,X-Auth-Token

**Credentials:** True

**Preflight continue:** True

**Step 12: Installing Heroku CLI (Ubuntu)**

sudo add-apt-repository "deb https://cli-assets.heroku.com/branches/stable/apt ./"

curl -L https://cli-assets.heroku.com/apt/release.key | sudo apt-key add -

sudo apt-get update

sudo apt-get install heroku

**Sample:**

ubuntu@ip-10-0-0-23:~$ curl -i -X POST --url http://54.183.161.178:8001/apis/ --data 'name=SFO\_POST' --data 'hosts=ec2-54-193-32-159.us-west-1.compute.amazonaws.com' --data 'upstream\_url=http://54.193.32.159:8080/SFO/orders'

HTTP/1.1 201 Created

Date: Thu, 13 Apr 2017 01:36:59 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Credentials: false

Server: kong/0.10.1

{"http\_if\_terminated":true,"id":"a97845f5-d6c8-43bd-aad0-cfc064915e43","retries":5,"preserve\_host":false,"created\_at":1492047418220,"upstream\_connect\_timeout":60000,"upstream\_url":"http:\/\/54.193.32.159:8080\/SFO\/orders","upstream\_send\_timeout":60000,"https\_only":false,"upstream\_read\_timeout":60000,"strip\_uri":true,"name":"SFO\_POST","hosts":["ec2-54-193-32-159.us-west-1.compute.amazonaws.com"]}

ubuntu@ip-10-0-0-23:~$ curl -i -X GET --url http://54.183.161.178:8001/apis/ --data 'name=SFO\_GET' --data 'hosts=ec2-54-193-32-159.us-west-1.compute.amazonaws.com' --data 'upstream\_url=http://54.193.32.159:8080/SFO/orders'

HTTP/1.1 200 OK

Date: Thu, 13 Apr 2017 01:37:30 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Credentials: false

Server: kong/0.10.1

{"data":[],"total":0}

ubuntu@ip-10-0-0-23:~$ curl -i -X POST --url http://54.183.161.178:8001/apis/ --data 'name=SFO\_GET' --data 'hosts=ec2-54-193-32-159.us-west-1.compute.amazonaws.com' --data 'upstream\_url=http://54.193.32.159:8080/SFO/orders'

HTTP/1.1 201 Created

Date: Thu, 13 Apr 2017 01:38:17 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

Access-Control-Allow-Origin: \*

Access-Control-Allow-Credentials: false

Server: kong/0.10.1

{"http\_if\_terminated":true,"id":"0ff8e41e-41b3-4881-9f7d-2cff9108865b","retries":5,"preserve\_host":false,"created\_at":1492047497210,"upstream\_connect\_timeout":60000,"upstream\_url":"http:\/\/54.193.32.159:8080\/SFO\/orders","upstream\_send\_timeout":60000,"https\_only":false,"upstream\_read\_timeout":60000,"strip\_uri":true,"name":"SFO\_GET","hosts":["ec2-54-193-32-159.us-west-1.compute.amazonaws.com"]}

**Important links for reference:**

<https://getkong.org/install/ubuntu/>

<http://datascale.io/how-to-create-a-cassandra-cluster-in-aws/>

<http://datascale.io/how-to-create-a-cassandra-cluster-in-aws-part-2/>

<http://docs.datastax.com/en/cassandra/3.0/cassandra/install/installRemove.html>

<http://ealfonso.com/setting-up-a-cassandra-cluster-on-awsubuntu14-04/>

<http://docs.datastax.com/en/archived/cql/3.0/cql/cql_reference/cqlsh.html>

<http://cassandra.apache.org/download/>

<https://getkong.org/docs/0.9.x/cli/#kong-reload>

<https://getkong.org/docs/0.2.x/cli/>

<https://github.com/Mashape/kong/issues/2174>

<https://dev9.com/blog-posts/2016/6/introduction-to-kong-api-gateway>