**Kong API Gateway and Konga dashboard installation on docker**

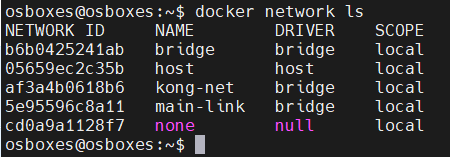
How to connect a Kong container to a Cassandra or PostgreSQL container.

Prerequisites:- Docker

Step1:- Create a Docker network

You will need to create a custom network to allow the containers to discover and communicate with each other. In this example kong-net is the network name, you can use any name.

$ docker network create kong-net



Step2: Start your database

If you wish to use a Cassandra container:

$ docker run -d --name kong-database \

--network**=**kong-net \

-p 9042:9042 \

cassandra:3

If you wish to use a PostgreSQL container:

$ docker run -d --name kong-database \

--network**=**kong-net \

-p 5432:5432 \

-e "POSTGRES\_USER=kong" \

-e "POSTGRES\_DB=kong" \

-e "POSTGRES\_PASSWORD=kong" \

postgres:9.6

Step3:- Prepare your database

Run the migrations with an ephemeral Kong container:

$ docker run --rm \

--network**=**kong-net \

-e "KONG\_DATABASE=postgres" \

-e "KONG\_PG\_HOST=kong-database" \

-e "KONG\_PG\_USER=kong" \

-e "KONG\_PG\_PASSWORD=kong" \

-e "KONG\_CASSANDRA\_CONTACT\_POINTS=kong-database" \

kong:latest kong migrations bootstrap

In the above example, both Cassandra and PostgreSQL are configured, but you should update the KONG\_DATABASE environment variable with either cassandra or postgres.

Note for Kong < 0.15: with Kong versions below 0.15 (up to 0.14), use the up sub-command instead of bootstrap. Also note that with Kong < 0.15, migrations should never be run concurrently; only one Kong node should be performing migrations at a time. This limitation is lifted for Kong 0.15, 1.0, and above.

Step 4:- Start Kong

When the migrations have run and your database is ready, start a Kong container that will connect to your database container, just like the ephemeral migrations container:

$ docker run -d --name kong \

--network**=**kong-net \

-e "KONG\_DATABASE=postgres" \

-e "KONG\_PG\_HOST=kong-database" \

-e "KONG\_PG\_USER=kong" \

-e "KONG\_PG\_PASSWORD=kong" \

-e "KONG\_CASSANDRA\_CONTACT\_POINTS=kong-database" \

-e "KONG\_PROXY\_ACCESS\_LOG=/dev/stdout" \

-e "KONG\_ADMIN\_ACCESS\_LOG=/dev/stdout" \

-e "KONG\_PROXY\_ERROR\_LOG=/dev/stderr" \

-e "KONG\_ADMIN\_ERROR\_LOG=/dev/stderr" \

-e "KONG\_ADMIN\_LISTEN=0.0.0.0:8001, 0.0.0.0:8444 ssl" \

-p 8000:8000 \

-p 8443:8443 \

-p 127.0.0.1:8001:8001 \

-p 127.0.0.1:8444:8444 \

kong:latest

Note:- We can user server’s IP instead of local host(127.0.0.1)

Step5 :- Use Kong

Kong is running:

$ curl -i http://localhost:8001/

How to setup Konga using Docker

Konga is an opensource powerful GUI that uses Kong Admin API, that makes managing of Kong quite literally…. a breeze.

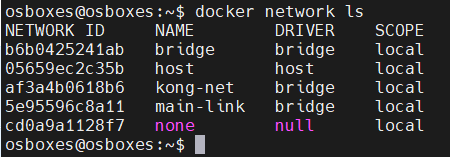
Today we are walking through setting up Konga using docker. We will be making use of PostgreSQL as our persistent database for Konga.

Step1:- First, we want to save PostgreSQL to a permanent folder, and to do that we will be creating a folder for this sole purpose

mkdir -p data

Step2:- Then the next step is to create a Docker network that will house both the PostgreSQL and Konga containers so that they can easily reach each other.  
P.S We could have them on the default network by default, but it is better to create network separation in case you have multiple docker images of the same type running. You do not want them clashing.

docker network create -d bridge main-link



Step3:- Then once done, we run the docker command to startup PostgreSQL using the alpine variant image and binding both the network and volume to the ones specified above. In addition to that, we set the default credentials for the database and set the name of the docker container

docker run --rm \

-e POSTGRES\_USER=konga \

-e POSTGRES\_DB=konga \

-e POSTGRES\_PASSWORD=konga \

-v data:/var/lib/postgresql/data \

--name postgres \

-d --network main-link \

postgres:9.6-alpine

Step4:- Then we need to seed the database with the Konga Schemas from the Konga image, we run the below before we start the Konga image proper.

docker run --network main-link --rm pantsel/konga:latest -c prepare -a postgres -u postgresql://konga:konga@postgres:5432/konga

Step5:- Once the Database has been seeded successfully, then you can start the docker image below.

docker run -p 1337:1337 \

--network main-link \

-e "TOKEN\_SECRET=ffffssf" \

-e "DB\_ADAPTER=postgres" \

-e "DB\_HOST=postgres" \

-e "DB\_PORT=5432" \

-e "DB\_USER=konga" \

-e "DB\_PASSWORD=konga" \

-e "DB\_DATABASE=konga" \

-e "NODE\_ENV=production" \

-d --name konga \

pantsel/konga

OR

docker run -p 1337:1337 --network=main-link -e "TOKEN\_SECRET=kong" -e "DB\_ADAPTER=postgres" -e "DB\_HOST=your-kong-database" -e "DB\_PORT=5432" -e "DB\_USER=kong" -e "DB\_PASSWORD=kong" -e "DB\_DATABASE=kong-database" -e "NODE\_ENV=production" -e "module=sails-postgresql" -e "ssl=true" --name konga pantsel/konga

docker run -p 1337:1337 \

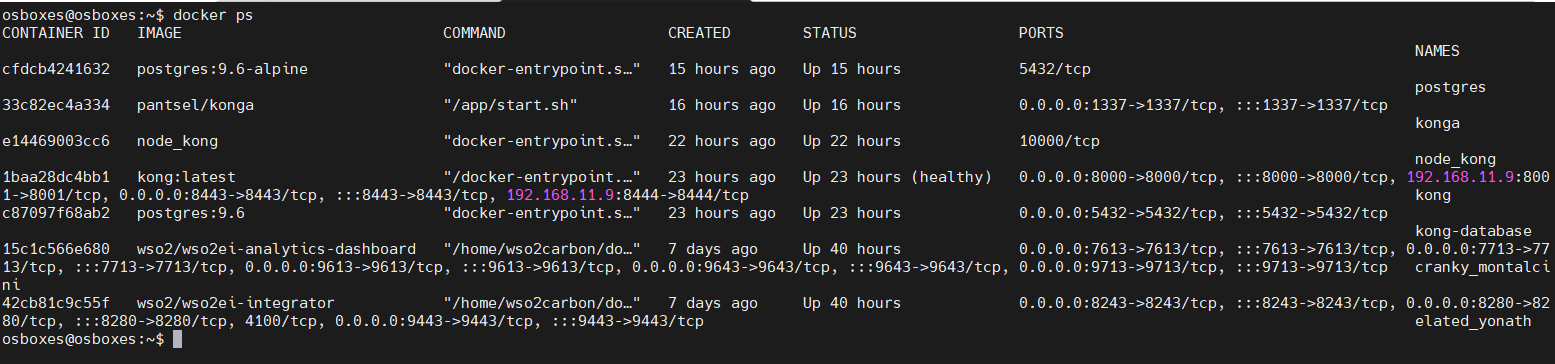
--link kong:kong \

--name konga \

--network main-link \

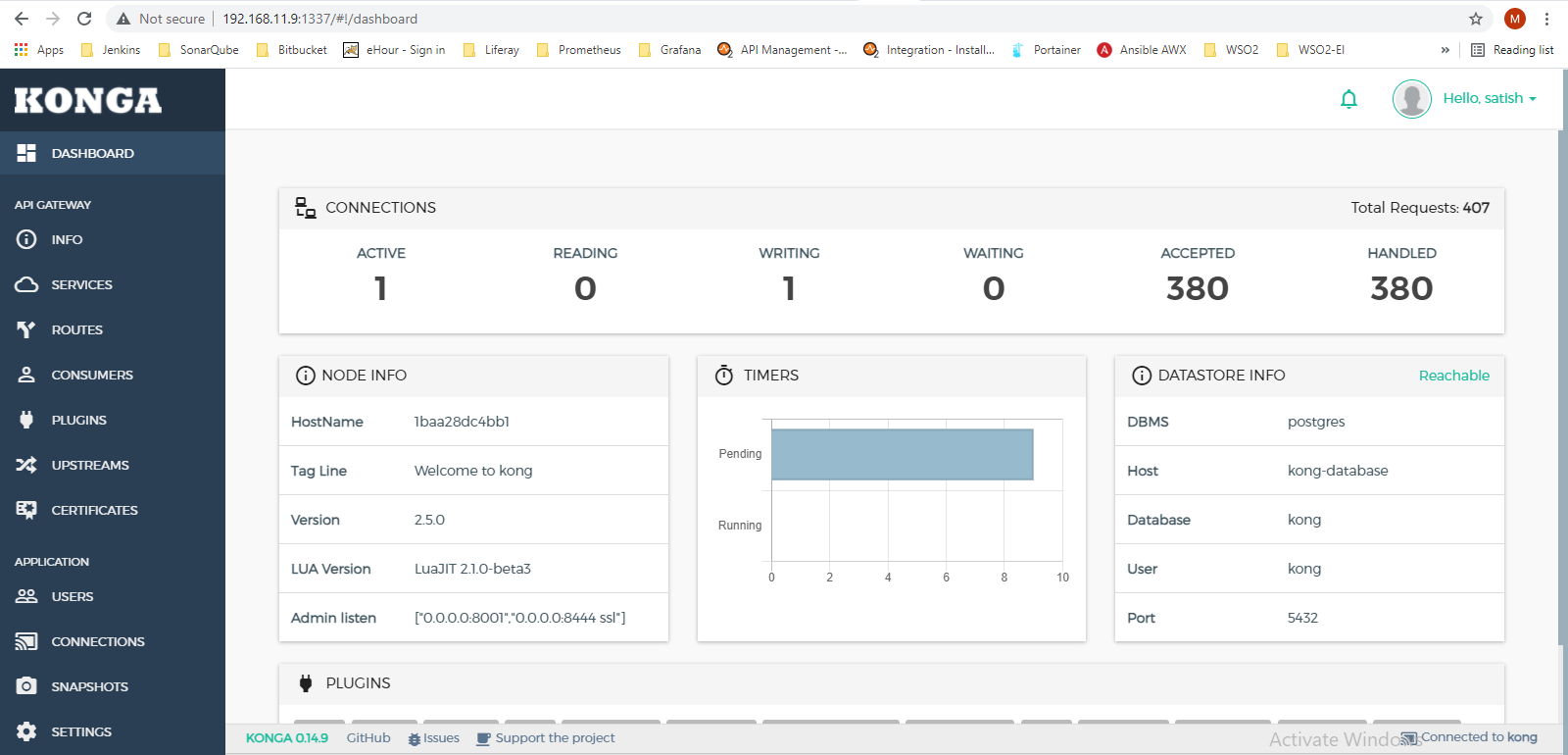
-e "NODE\_ENV=production" \

pantsel/konga



Step6:- Start using Konga dashboard in your browser with below IP

<http://192.168.11.9:1337/>



Reference URLs:-

1. For Kong API Manager installation

Link:- <https://docs.konghq.com/install/docker/>

1. For Konga dashboard installation

Link:- <https://www.dennisotugo.com/how-to-setup-konga-using-docker/>

OR

<https://frameworks.readthedocs.io/en/latest/framework/api/kongaDocker.html>

1. For Setting up Kong API Gateway

Link:- <https://mrkaran.dev/posts/setting-up-kong-part-1/>

<https://mrkaran.dev/posts/setting-up-kong-part-2/>

1. For Free fake API for testing and prototyping

Link:- <https://jsonplaceholder.typicode.com/>