Building our own NGINX Image

**Step 1**: Create a dockerfile (in any of the directory) and name it dockerfile using any text editor like vi/nano etc… .

**Step 2**: In the dockerfile specify your nginx configurations, ie. what are all the dependencies. Your dockerfile should look like🡪

############################################################

# Dockerfile to build Nginx Installed Containers

# Based on Ubuntu

############################################################

# Set the base image to Ubuntu

FROM ubuntu

# File Author / Maintainer

MAINTAINER Maintaner Name

# Install Nginx

# Add application repository URL to the default sources

RUN echo "deb http://archive.ubuntu.com/ubuntu/ raring main universe" >> /etc/apt/sources.list

# Update the repository

RUN apt-get update

# Install necessary tools

RUN apt-get install -y nano wget dialog net-tools

# Download and Install Nginx

RUN apt-get install -y nginx

# Remove the default Nginx configuration file

RUN rm -v /etc/nginx/nginx.conf

# Copy a configuration file from the current directory

ADD nginx.conf /etc/nginx/

# Append "daemon off;" to the beginning of the configuration

RUN echo "daemon off;" >> /etc/nginx/nginx.conf

# Expose ports

EXPOSE 80

# Set the default command to execute

# when creating a new container

CMD service nginx start

**Step 3**: After creating the dockerfile we should create an image of it using **docker build** command.

**Step 4**: Before that create an configuration file for nginx (to refer the ips nginx for load balancing). Using any text editor create **nginx.conf** file under the directory **/etc/nginx/nginx.cong.**

Your nginx.conf file should look like 🡪

worker\_processes 1;

events { worker\_connections 1024; }

http {

upstream servers {

server 172.18.0.2:8080; #your ip

server 172.18.0.3:8080; #your ip

server 172.18.0.4:8080; #your ip

}

server {

listen 80;

location / {

proxy\_pass <http://servers>;

}

}}

**Step 5**: Under the dockerfile directory execute the **docker build** command.(Note: make sure your dockerfile is in the same folder as the nginx.cong file)

sudo docker build -t nginx\_img\_1 .

**Step 6**: Check your docker image created using the command

docker images

**Step 7**: Run the image you create in a new container by running the command.

sudo docker run -name nginx\_cont\_1 -p 80:80 -i -t nginx\_img\_1

Use this image as your nginx image to configure in the docker-compose.yml file. Hence, just by starting the docker-compose, the the images configured under the docker-compose get started and also the nginx. The images will be running in some default virtual ips. These virtual ips should be configured in the nginx.conf file.

The docker-compose.yml file should look like🡪

version: '2'

services:

nginx:

image: 18f6b3240253(our nginx image we created)

ports:

- "81:81"

# - "9999/tcp"

depends\_on:

- weba

- webb

- webc

weba:

image: 65c3baae63a6

webb:

image: 65c3baae63a6

webc:

image: 65c3baae63a6

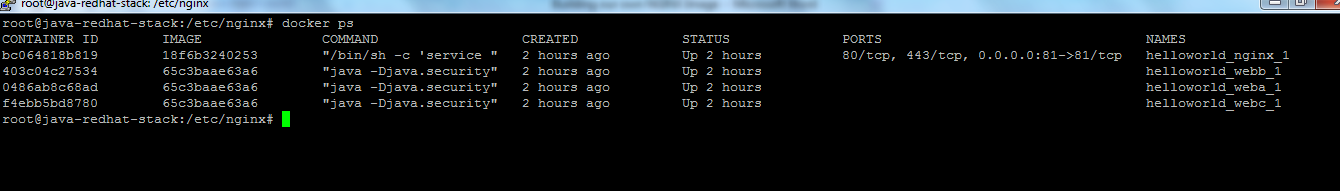
Step 8: Start the docker-compose by running the command

docker-compose up

This will automatically start three running images(weba, webb, webc) and also the nginx image. You can check it by running the command

docker ps

This will list all the running docker containers🡪



**Step 9**: So, everything works fine. Run a curl command to access nginx(load balancer).

curl http:// 172.18.0.5 (change your own ip)

This will automatically connect to the running images and will give the output.

NOTE: For any doubt refer the url below.

<https://www.digitalocean.com/community/tutorials/docker-explained-how-to-containerize-and-use-nginx-as-a-proxy>