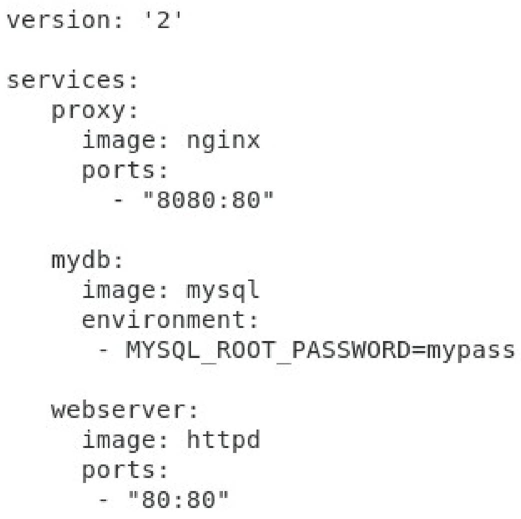
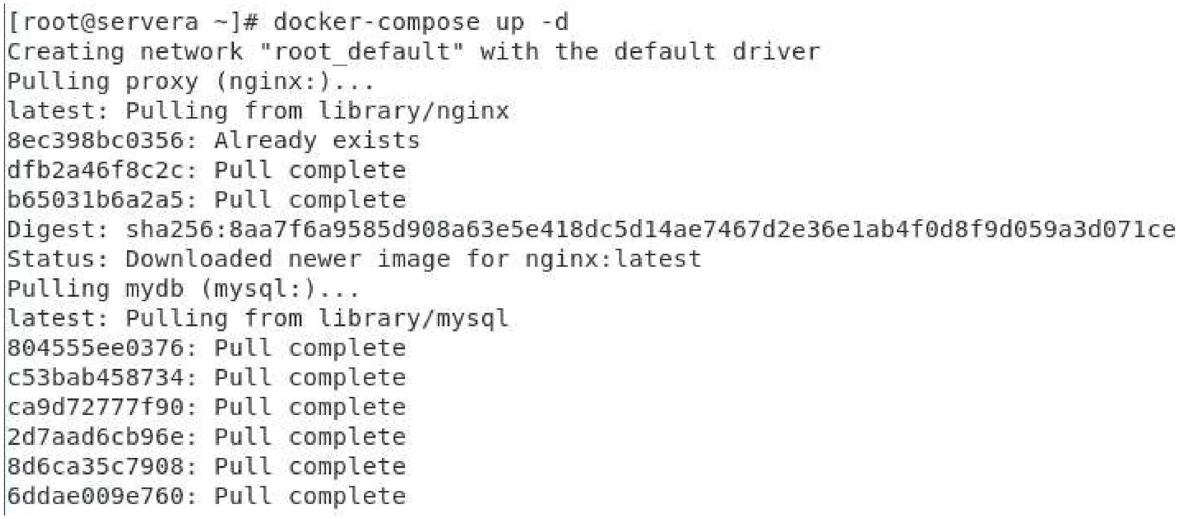
Practice F : Using docker-compose to create multi-container environment – nginx-httpd-mysql

- Create a docker-compose.yml file that will bring up 3 containers
- 3 containers are:
  - Nginx that listens to port 8080 named as proxy
  - Mysql that uses ‘mypass’ as the password and named as mydb
  - Httpd service that uses port 80 and named as webserver
- All 3 containers are not connected to each other, this practice is about docker-compose.yml and understanding of how to bring up multiple container up in single command

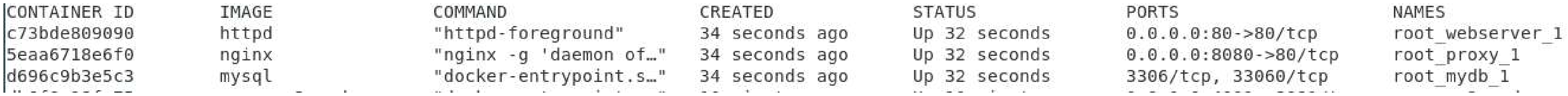
Practice F: Solution



```
version: '2'

services:
   proxy:
     image: nginx
     ports:
       - "8080:80"

   mydb:
     image: mysql
     environment:
      - MYSQL_ROOT_PASSWORD=mypass

   webserver:
     image: httpd
     ports:
      - "80:80"
```



```
[root@servera ~]# docker-compose up -d
Creating network "root_default" with the default driver
Pulling proxy (nginx:)...
latest: Pulling from library/nginx
8ec398bc0356: Already exists
dfb2a46f8c2c: Pull complete
b65031b6a2a5: Pull complete
Digest: sha256:8aa7f6a9585d908a63e5e418dc5d14ae7467d2e36e1ab4f0d8f9d059a3d071ce
Status: Downloaded newer image for nginx:latest
Pulling mydb (mysql:)...
latest: Pulling from library/mysql
804555ee0376: Pull complete
c53bab458734: Pull complete
ca9d72777f90: Pull complete
2d7aad6cb96e: Pull complete
8d6ca35c7908: Pull complete
6ddae009e760: Pull complete
```



```
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
c73bde809090        httpd               "httpd-foreground"       34 seconds ago      Up 32 seconds       0.0.0.0:80->80/tcp       root_webserver_1
5eaa6718e6f0        nginx               "nginx -g 'daemon of…"   34 seconds ago      Up 32 seconds       0.0.0.0:8080->80/tcp     root_proxy_1
d696c9b3e5c3        mysql               "docker-entrypoint.s…"   34 seconds ago      Up 32 seconds       3306/tcp, 33060/tcp      root_mydb_1
```