MANAGEMENT OF MICROSERVICES BASED APPLICATIONS

Learning Report

Intern Name	:	Harsh Siddhapura
University Roll No.	:	*91800133026*
Intern Semester	:	6th Semester
Intern Department	:	Information & Communication Tech.
Intern University	:	Marwadi University
Research Guide	:	Prof. Julia Rubin
Guide Host University	:	University of British Columbia

Introduction

Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are

- ❖ Highly maintainable and testable
- ❖ Loosely coupled
- ❖ Independently deployable
- ❖ Organized around business capabilities
- ❖ Owned by a small team

The microservice architecture enables the rapid, frequent and reliable delivery of large, complex applications. It also enables an organization to evolve its technology stack. It has several drawbacks. Moreover, when using this architecture there are numerous issues that you must address. The microservice architecture pattern language is a collection of patterns for applying the microservice architecture. It has two goals:

- ❖ The pattern language enables you to decide whether microservices are a good fit for your application.
- ❖ The pattern language enables you to use the microservice architecture successfully.

Installing Docker for Windows

Link for Complete Guide : **https://docs.docker.com/docker-for-windows/install/**

System requirements

Your Windows machine must meet the following requirements to successfully install Docker Desktop.

Hyper-V backend and Windows containers

- ❖ Windows 10 64-bit: Pro, Enterprise, or Education (Build 17134 or higher).
- ❖ Hyper-V and Containers Windows features must be enabled.
- ❖ The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
    - ➢ 64 bit processor with Second Level Address Translation (SLAT)
    - ➢ 4GB system RAM

  - ➢ BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.

WSL 2 backend

- ❖ Windows 10 64-bit: Home, Pro, Enterprise, or Education, version 1903 (Build 18362 or higher).
- ❖ Enable the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
- ❖ The following hardware prerequisites are required to successfully run WSL 2 on Windows 10:

  - ➢ 64-bit processor with Second Level Address Translation (SLAT)
  - ➢ 4GB system RAM
  - ➢ BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.

- ❖ Download and install the Linux kernel update package.

What's included in the installer

The Docker Desktop installation includes Docker Engine, Docker CLI client, Docker Compose, Notary, Kubernetes, and Credential Helper.

Containers and images created with Docker Desktop are shared between all user accounts on machines where it is installed. This is because all Windows accounts use the same VM to build and run containers. Note that it is not possible to share containers and images between user accounts when using the Docker Desktop WSL 2 backend.

Nested virtualization scenarios, such as running Docker Desktop on a VMWare or Parallels instance might work, but there are no guarantees. For more information, see Running Docker Desktop in nested virtualization scenarios.

About Windows containers

Looking for information on using Windows containers?

- ❖ Switch between Windows and Linux containers describes how you can toggle between Linux and Windows containers in Docker Desktop and points you to the tutorial mentioned above.
- ❖ Getting Started with Windows Containers (Lab) provides a tutorial on how to set up and run Windows containers on Windows 10, Windows Server 2016 and

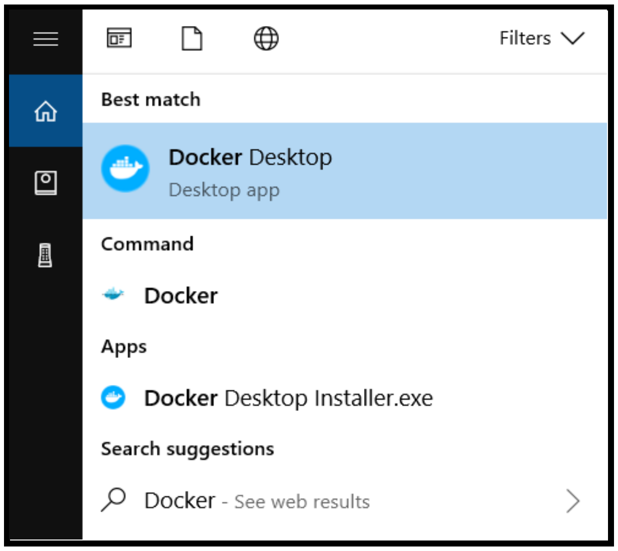
Windows Server 2019. It shows you how to use a MusicStore application with Windows containers.
- Docker Container Platform for Windows articles and blog posts on the Docker website.

Install Docker Desktop on Windows

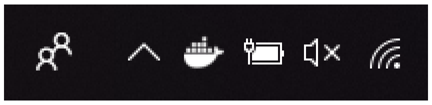
- Double-click Docker Desktop Installer.exe to run the installer.
- If you haven't already downloaded the installer (Docker Desktop Installer.exe), you can get it from Docker Hub. It typically downloads to your Downloads folder, or you can run it from the recent downloads bar at the bottom of your web browser.
- When prompted, ensure the Enable Hyper-V Windows Features or the Install required Windows components for WSL 2 option is selected on the Configuration page.
- Follow the instructions on the installation wizard to authorize the installer and proceed with the install.
- When the installation is successful, click Close to complete the installation process.
- If your admin account is different to your user account, you must add the user to the docker-users group. Run Computer Management as an administrator and navigate to Local Users and Groups > Groups > docker-users. Right-click to add the user to the group. Log out and log back in for the changes to take effect.

Start Docker Desktop

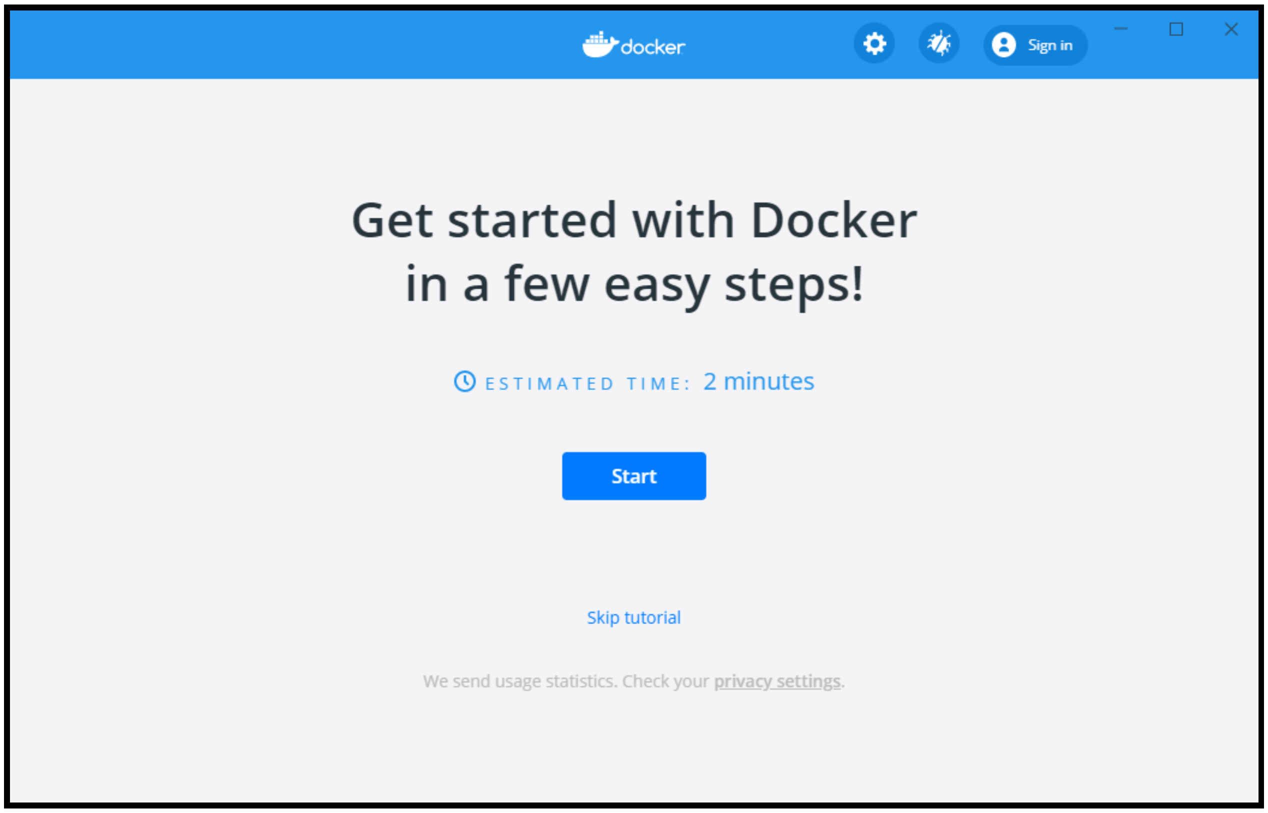
Docker Desktop does not start automatically after installation. To start Docker Desktop, search for Docker, and select Docker Desktop in the search results.



When the whale icon in the status bar stays steady, Docker Desktop is up-and-running, and is accessible from any terminal window.

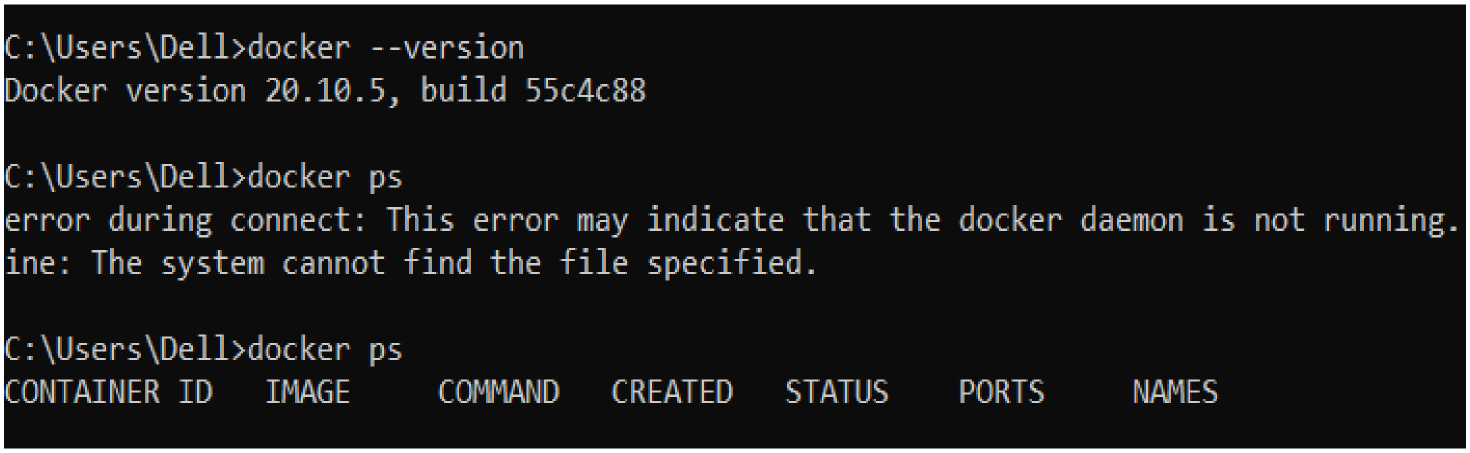


When the initialization is complete, Docker Desktop launches the onboarding tutorial. The tutorial includes a simple exercise to build an example Docker image, run it as a container, push and save the image to Docker Hub.

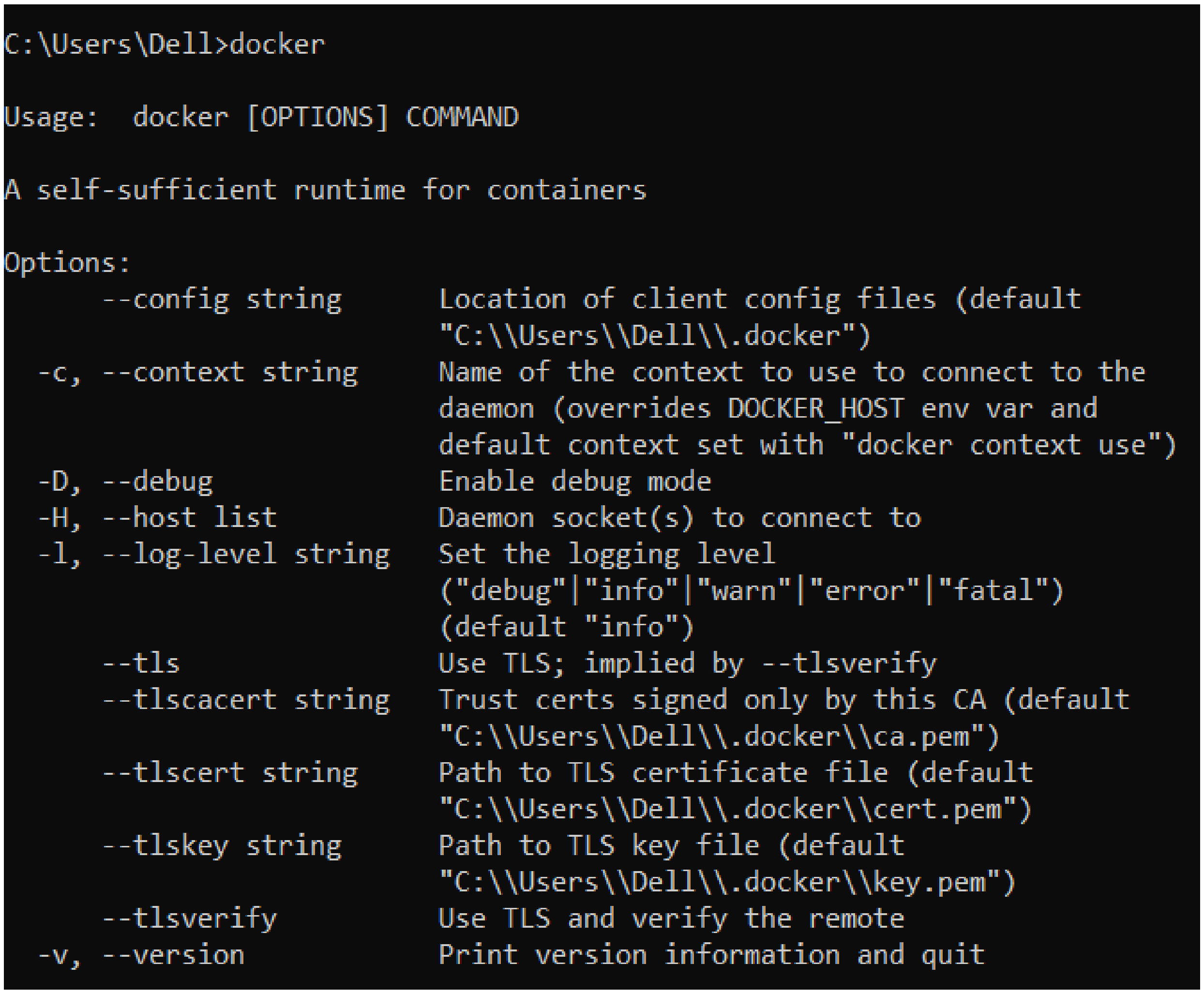

docker
Sign in
Get started with Docker in a few easy steps!
ESTIMATED TIME: 2 minutes
Start
Skip tutorial
We send usage statistics. Check your privacy settings.

Verifying Installation of Docker

Open the Windows Command Prompt and follow as shown in below...

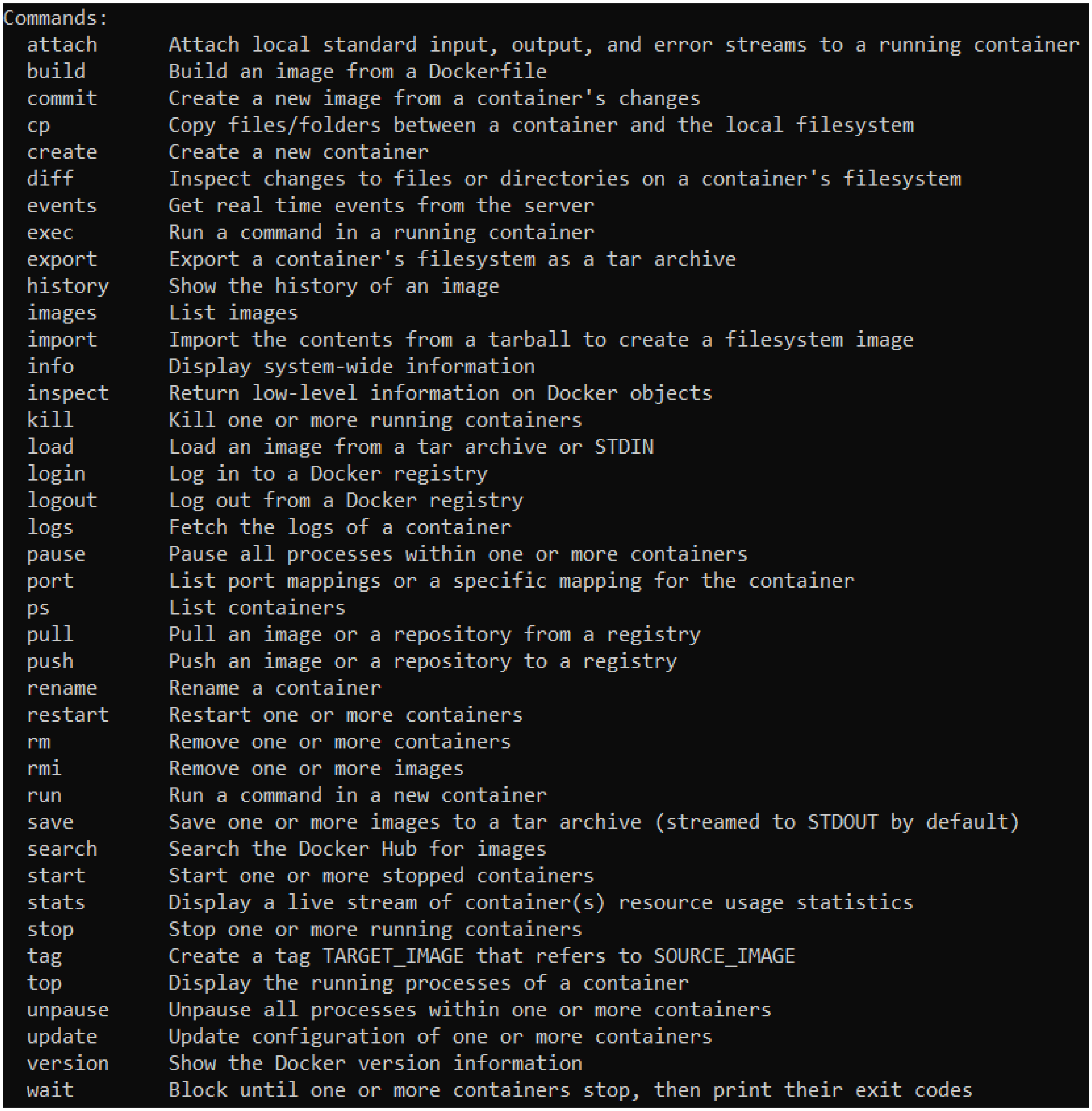


```
C:\Users\Dell>docker --version
Docker version 20.10.5, build 55c4c88

C:\Users\Dell>docker ps
error during connect: This error may indicate that the docker daemon is not running.
ine: The system cannot find the file specified.

C:\Users\Dell>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```



```
C:\Users\Dell>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "C:\\Users\\Dell\\.docker")
  -c, --context string     Name of the context to use to connect to the
                           daemon (overrides DOCKER_HOST env var and
                           default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level
                           ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default
                           "C:\\Users\\Dell\\.docker\\ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "C:\\Users\\Dell\\.docker\\cert.pem")
      --tlskey string      Path to TLS key file (default
                           "C:\\Users\\Dell\\.docker\\key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit
```

Commands :

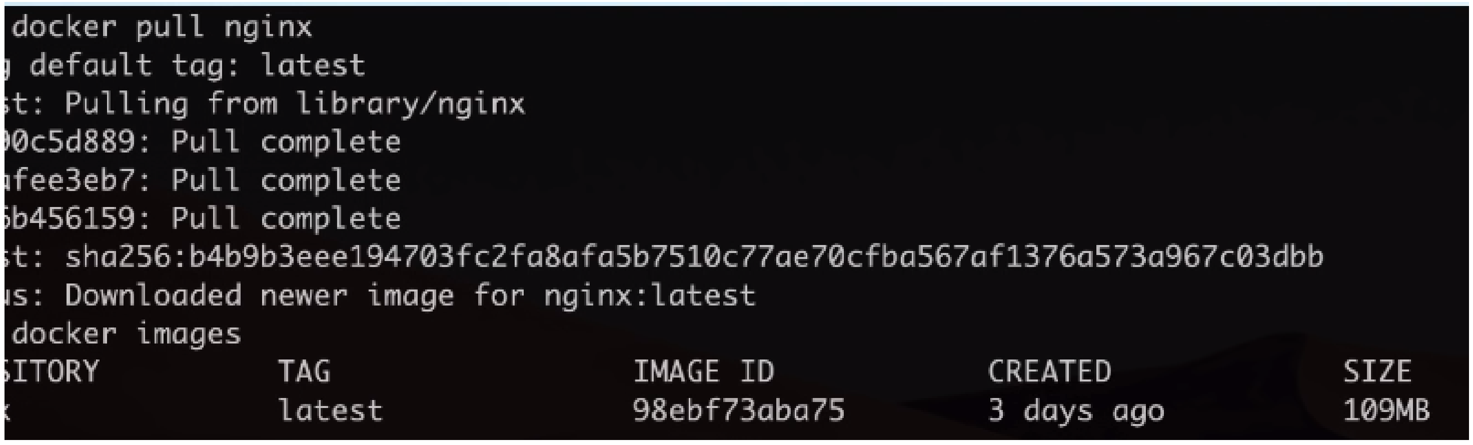


```
Commands:
  attach      Attach local standard input, output, and error streams to a running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit codes
```

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/

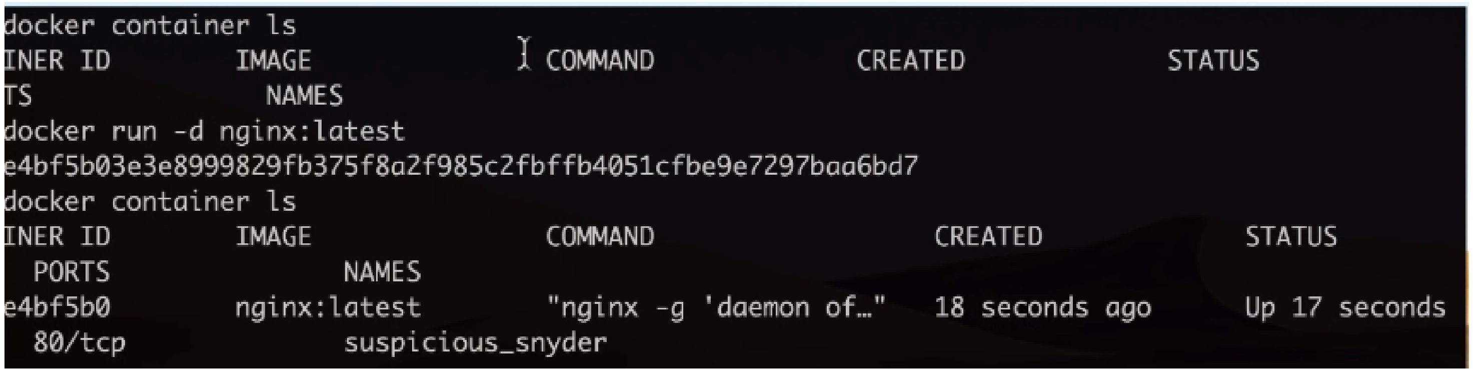
Images, Containers & Ports

Pulling Images from Docker Hub

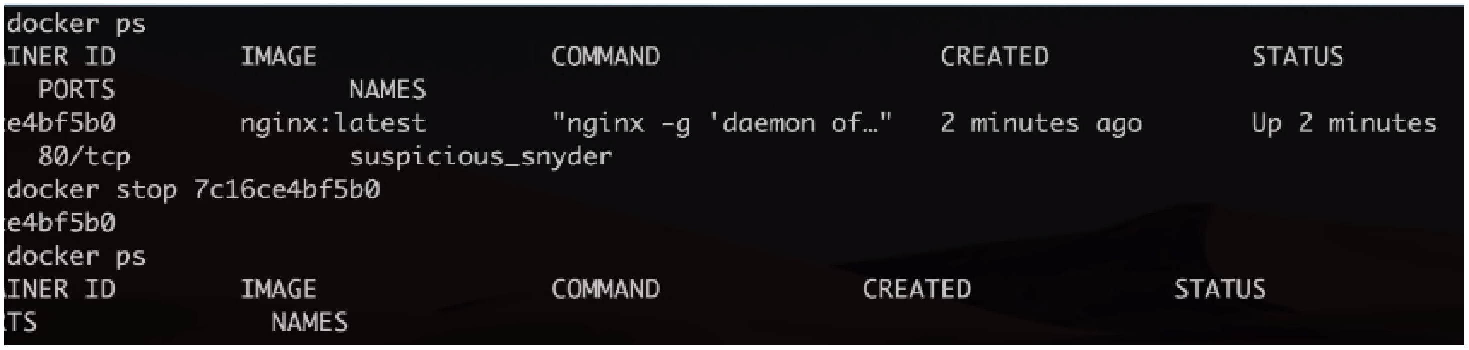


```
 docker pull nginx
g default tag: latest
st: Pulling from library/nginx
00c5d889: Pull complete
afee3eb7: Pull complete
6b456159: Pull complete
st: sha256:b4b9b3eee194703fc2fa8afa5b7510c77ae70cfba567af1376a573a967c03dbb
us: Downloaded newer image for nginx:latest
 docker images
SITORY          TAG                 IMAGE ID            CREATED             SIZE
x               latest              98ebf73aba75        3 days ago          109MB
```

Creating Container

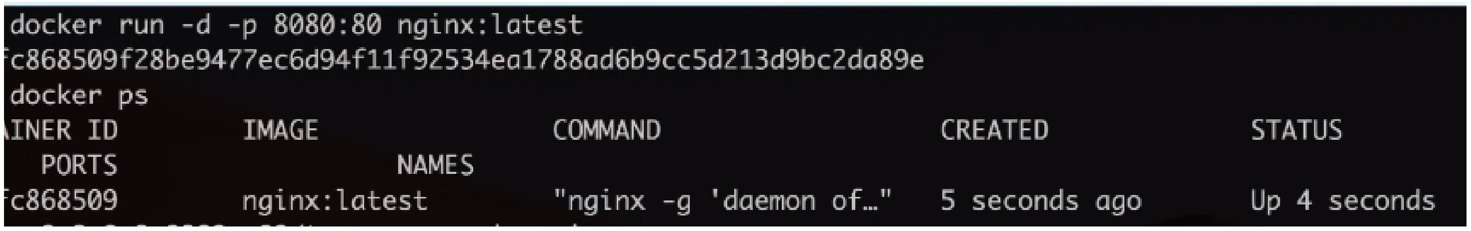


```
docker container ls
INER ID        IMAGE              COMMAND            CREATED            STATUS
TS                NAMES
docker run -d nginx:latest
e4bf5b03e3e8999829fb375f8a2f985c2fbffb4051cfbe9e7297baa6bd7
docker container ls
INER ID        IMAGE              COMMAND                  CREATED            STATUS
  PORTS              NAMES
e4bf5b0        nginx:latest       "nginx -g 'daemon of…"   18 seconds ago     Up 17 seconds
  80/tcp             suspicious_snyder
```

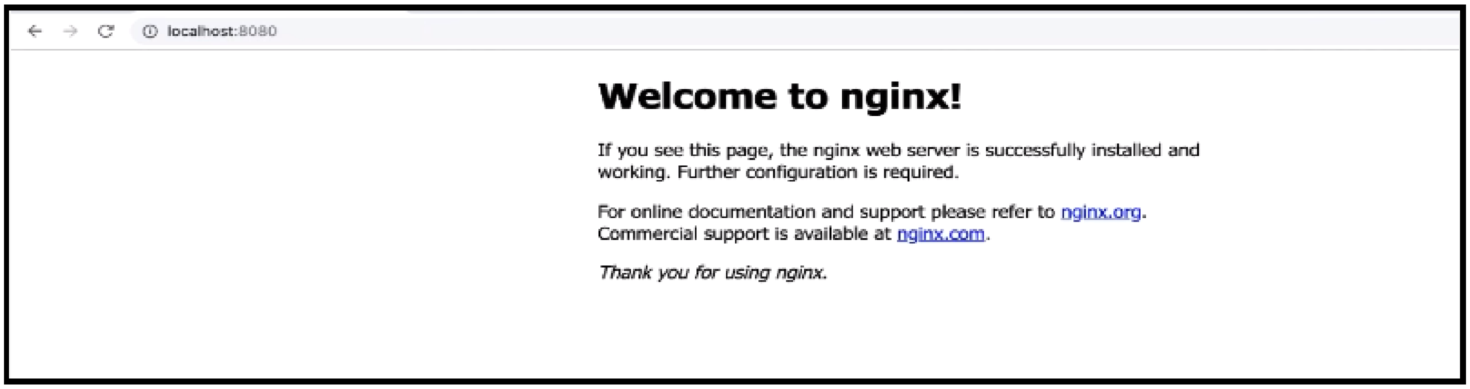


```
docker ps
INER ID        IMAGE              COMMAND                  CREATED            STATUS
  PORTS              NAMES
e4bf5b0        nginx:latest       "nginx -g 'daemon of…"   2 minutes ago      Up 2 minutes
  80/tcp             suspicious_snyder
docker stop 7c16ce4bf5b0
e4bf5b0
docker ps
INER ID        IMAGE              COMMAND            CREATED            STATUS
TS                NAMES
```

Exposing Single & Multiple Ports of Containers



```
 docker run -d -p 8080:80 nginx:latest
fc868509f28be9477ec6d94f11f92534ea1788ad6b9cc5d213d9bc2da89e
 docker ps
AINER ID        IMAGE               COMMAND                  CREATED             STATUS
   PORTS                  NAMES
fc868509        nginx:latest        "nginx -g 'daemon of…"   5 seconds ago       Up 4 seconds
```



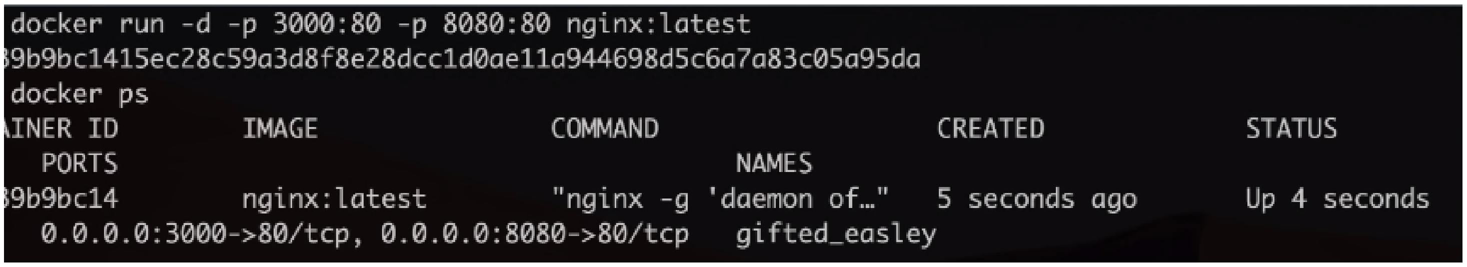
localhost:8080

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

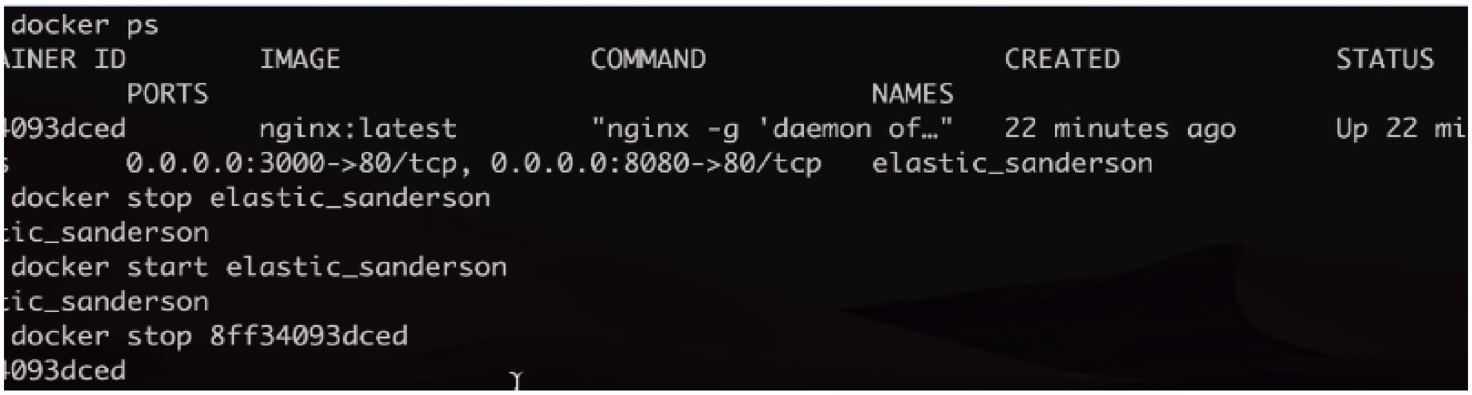
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

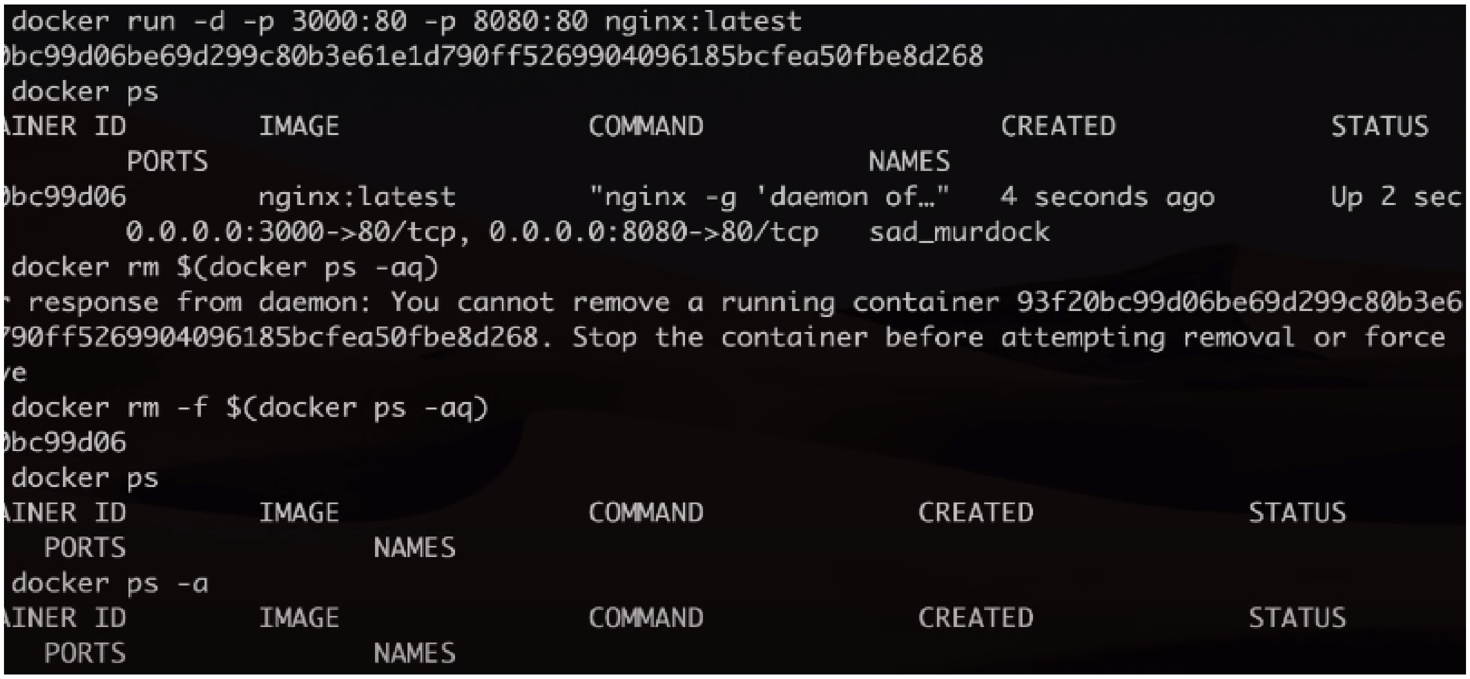


```
 docker run -d -p 3000:80 -p 8080:80 nginx:latest
39b9bc1415ec28c59a3d8f8e28dcc1d0ae11a944698d5c6a7a83c05a95da
 docker ps
AINER ID        IMAGE               COMMAND                  CREATED             STATUS
   PORTS                                         NAMES
39b9bc14        nginx:latest        "nginx -g 'daemon of…"   5 seconds ago       Up 4 seconds
   0.0.0.0:3000->80/tcp, 0.0.0.0:8080->80/tcp   gifted_easley
```

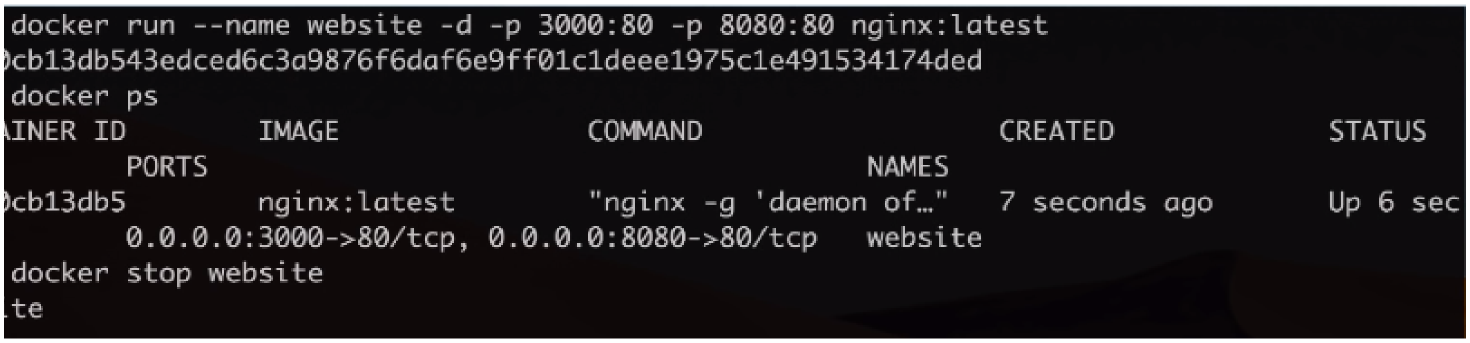
Managing Containers (Start, Stop, Remove & Naming)



```
 docker ps
AINER ID        IMAGE               COMMAND                  CREATED             STATUS
        PORTS                                         NAMES
4093dced        nginx:latest        "nginx -g 'daemon of…"   22 minutes ago      Up 22 mi
s       0.0.0.0:3000->80/tcp, 0.0.0.0:8080->80/tcp   elastic_sanderson
 docker stop elastic_sanderson
tic_sanderson
 docker start elastic_sanderson
tic_sanderson
 docker stop 8ff34093dced
4093dced
```



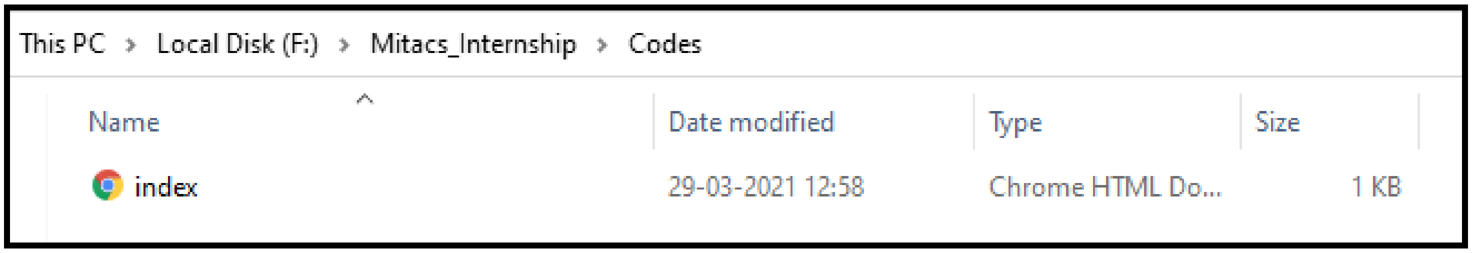
```
 docker run -d -p 3000:80 -p 8080:80 nginx:latest
)bc99d06be69d299c80b3e61e1d790ff5269904096185bcfea50fbe8d268
 docker ps
AINER ID        IMAGE                COMMAND                  CREATED             STATUS
       PORTS                                            NAMES
)bc99d06        nginx:latest         "nginx -g 'daemon of…"   4 seconds ago       Up 2 sec
       0.0.0.0:3000->80/tcp, 0.0.0.0:8080->80/tcp   sad_murdock
 docker rm $(docker ps -aq)
r response from daemon: You cannot remove a running container 93f20bc99d06be69d299c80b3e6
790ff5269904096185bcfea50fbe8d268. Stop the container before attempting removal or force
ve
 docker rm -f $(docker ps -aq)
)bc99d06
 docker ps
AINER ID        IMAGE                COMMAND             CREATED             STATUS
    PORTS               NAMES
 docker ps -a
AINER ID        IMAGE                COMMAND             CREATED             STATUS
    PORTS               NAMES
```



```
 docker run --name website -d -p 3000:80 -p 8080:80 nginx:latest
)cb13db543edced6c3a9876f6daf6e9ff01c1deee1975c1e491534174ded
 docker ps
AINER ID        IMAGE                COMMAND                  CREATED             STATUS
       PORTS                                            NAMES
)cb13db5        nginx:latest         "nginx -g 'daemon of…"   7 seconds ago       Up 6 sec
       0.0.0.0:3000->80/tcp, 0.0.0.0:8080->80/tcp   website
 docker stop website
te
```

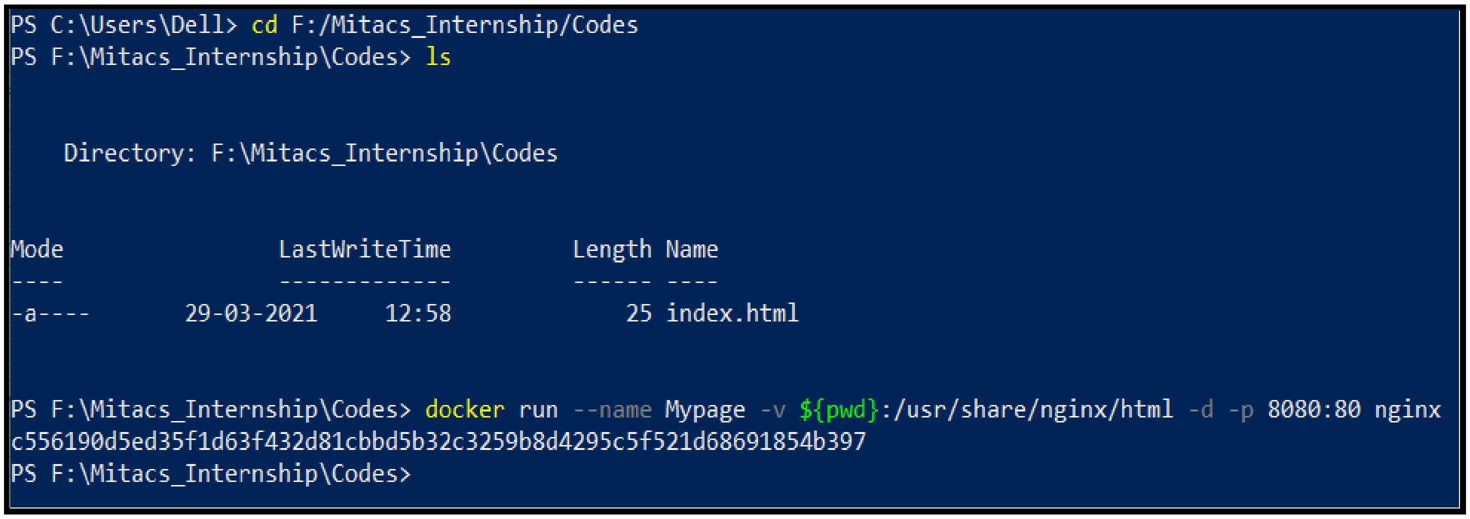
Volumes

Creating Volume between Host & Container

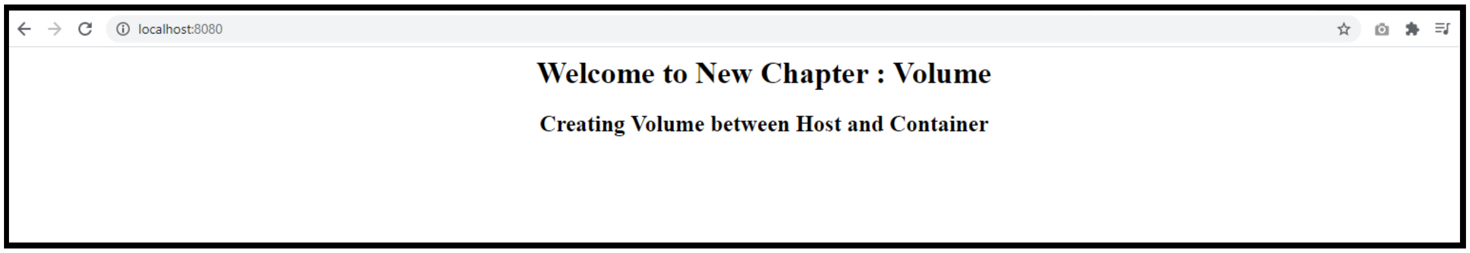


This PC > Local Disk (F:) > Mitacs_Internship > Codes

Name	Date modified	Type	Size
index	29-03-2021 12:58	Chrome HTML Do...	1 KB



```
PS C:\Users\Dell> cd F:/Mitacs_Internship/Codes
PS F:\Mitacs_Internship\Codes> ls


    Directory: F:\Mitacs_Internship\Codes


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        29-03-2021     12:58             25 index.html


PS F:\Mitacs_Internship\Codes> docker run --name Mypage -v ${pwd}:/usr/share/nginx/html -d -p 8080:80 nginx
c556190d5ed35f1d63f432d81cbbd5b32c3259b8d4295c5f521d68691854b397
PS F:\Mitacs_Internship\Codes>
```

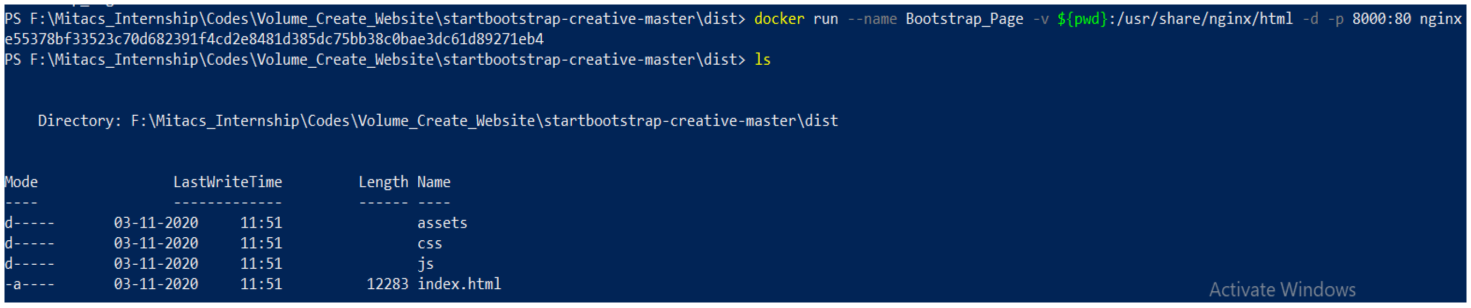


Welcome to New Chapter : Volume

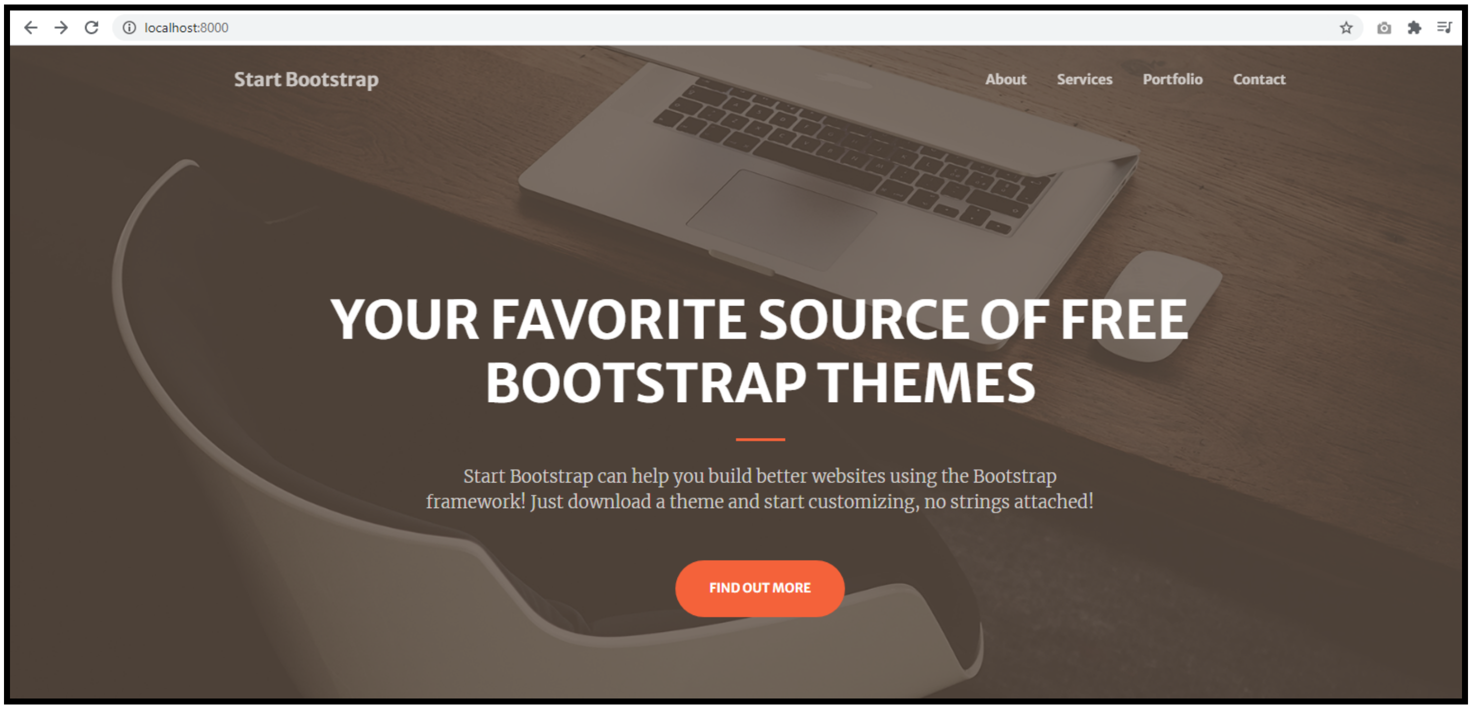
Creating Volume between Host and Container

Customizing Website using Volumes

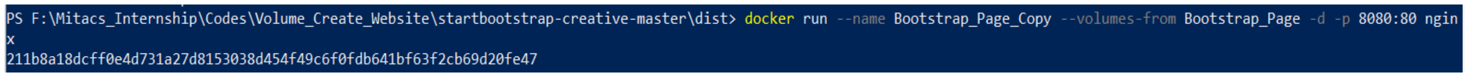
For simplicity, take any bootstrap page and download it from any website or github repository. Unzip the file and place it in our working folder...



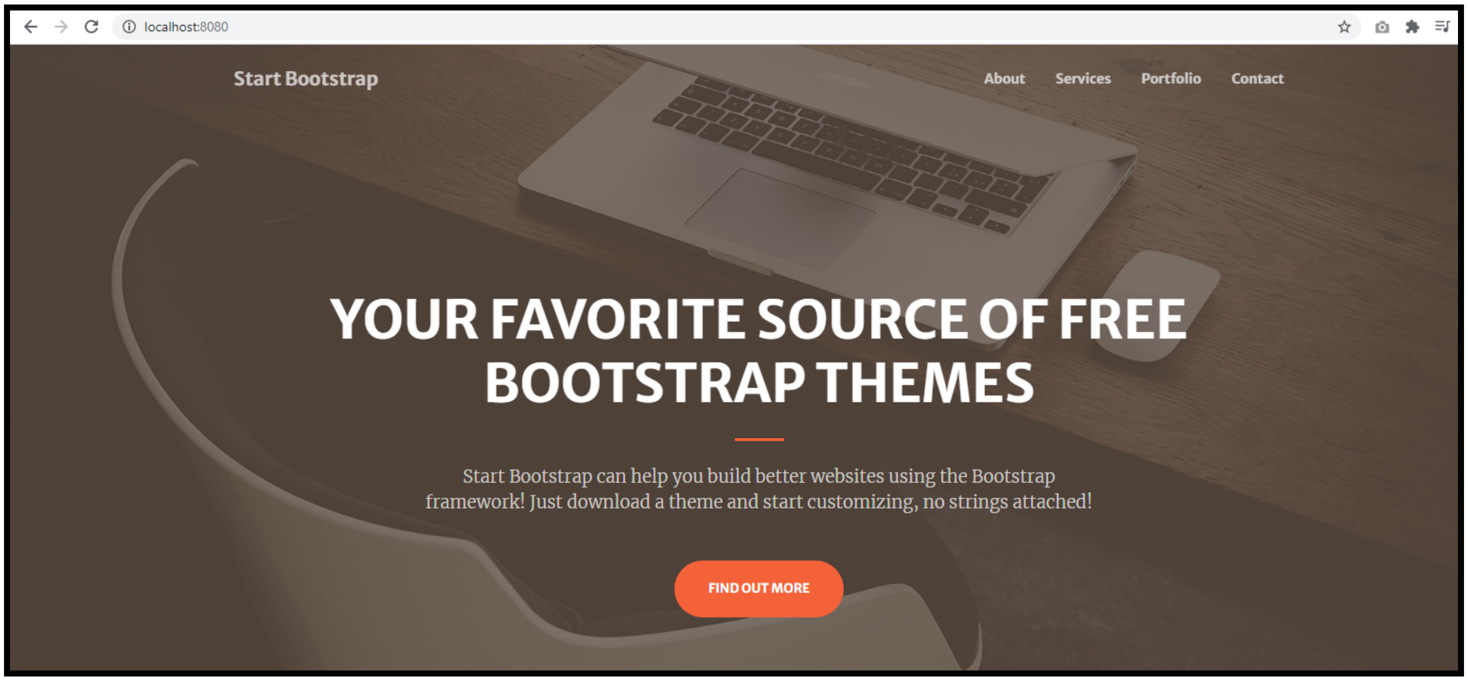
```
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-creative-master\dist> docker run --name Bootstrap_Page -v ${pwd}:/usr/share/nginx/html -d -p 8000:80 nginx
e55378bf33523c70d682391f4cd2e8481d385dc75bb38c0bae3dc61d89271eb4
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-creative-master\dist> ls


    Directory: F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-creative-master\dist


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        03-11-2020     11:51                assets
d-----        03-11-2020     11:51                css
d-----        03-11-2020     11:51                js
-a----        03-11-2020     11:51          12283 index.html
```

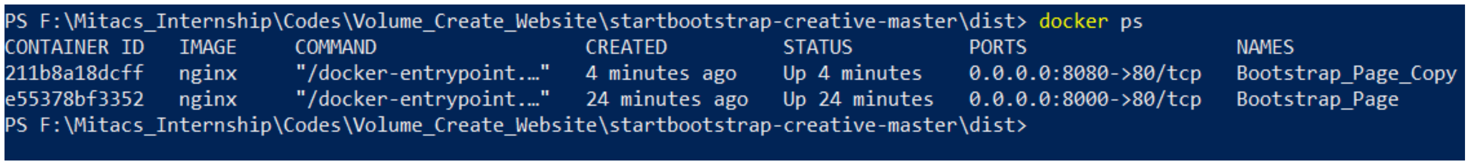

localhost:8000
Start Bootstrap
About
Services
Portfolio
Contact
YOUR FAVORITE SOURCE OF FREE BOOTSTRAP THEMES
Start Bootstrap can help you build better websites using the Bootstrap framework! Just download a theme and start customizing, no strings attached!
FIND OUT MORE

Volumes between Containers



```
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-creative-master\dist> docker run --name Bootstrap_Page_Copy --volumes-from Bootstrap_Page -d -p 8080:80 ngin
x
211b8a18dcff0e4d731a27d8153038d454f49c6f0fdb641bf63f2cb69d20fe47
```


localhost:8080
Start Bootstrap
About
Services
Portfolio
Contact
YOUR FAVORITE SOURCE OF FREE BOOTSTRAP THEMES
Start Bootstrap can help you build better websites using the Bootstrap framework! Just download a theme and start customizing, no strings attached!
FIND OUT MORE



```
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-creative-master\dist> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                  NAMES
211b8a18dcff   nginx     "/docker-entrypoint.…"   4 minutes ago    Up 4 minutes    0.0.0.0:8080->80/tcp   Bootstrap_Page_Copy
e55378bf3352   nginx     "/docker-entrypoint.…"   24 minutes ago   Up 24 minutes   0.0.0.0:8000->80/tcp   Bootstrap_Page
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-creative-master\dist>
```

Building Images : Dockerfiles

Dockerfile Introduction

Dockerfile helps us to build our own images of a website. So, we will create the image (using dockerfile) of our own website.

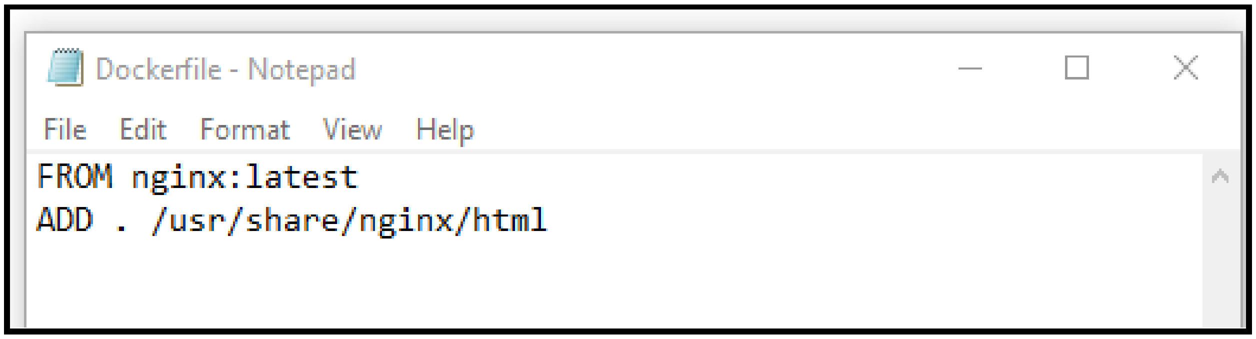
Link : **https://docs.docker.com/engine/reference/builder/**

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

Creating Dockerfile or Custom Image

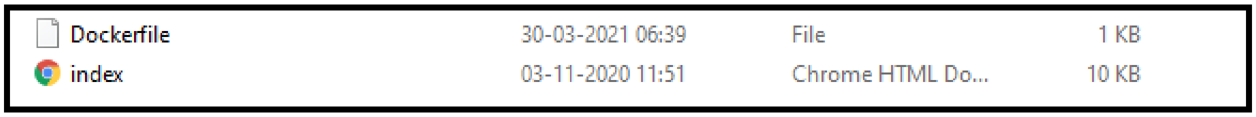
Follow below instructions strictly :

- ❖ Open "Notepad" in windows.
- ❖ Type in the information you would like to save without an extension.



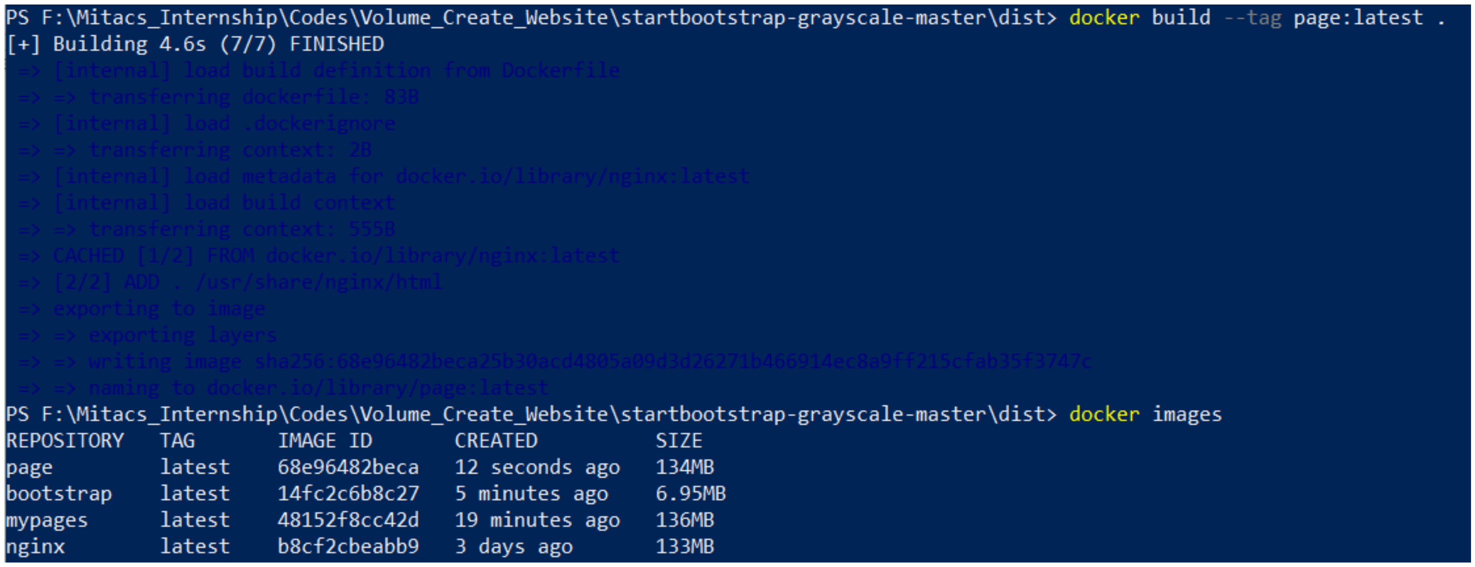
```
FROM nginx:latest
ADD . /usr/share/nginx/html
```

- ❖ Click "File" and then "Save" and the "Save As" dialog box is displayed.
- ❖ Type an opening quotation mark, the file name and then the closing quotation mark in the "File name" section. For example, type "Dockerfile" to create a file called noextension.
- ❖ Click the "Save" button.



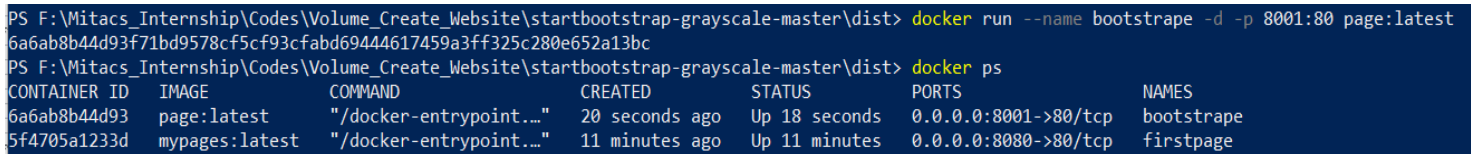
```
Dockerfile    30-03-2021 06:39    File                 1 KB
index         03-11-2020 11:51    Chrome HTML Do...    10 KB
```

Building Image :



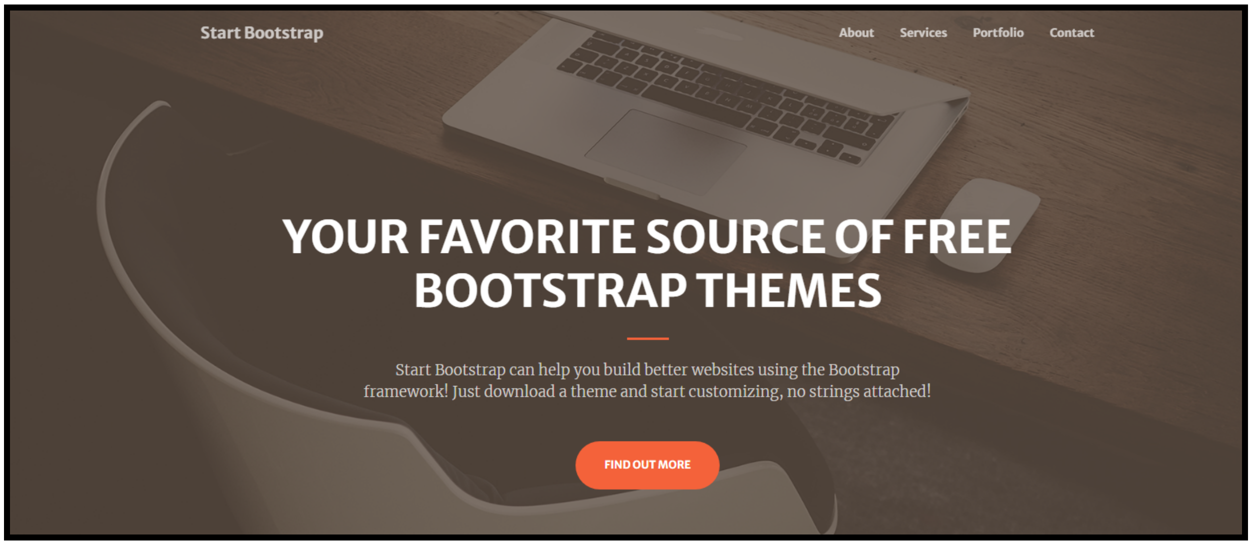
```
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-grayscale-master\dist> docker build --tag page:latest .
[+] Building 4.6s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 83B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/nginx:latest
 => [internal] load build context
 => => transferring context: 555B
 => CACHED [1/2] FROM docker.io/library/nginx:latest
 => [2/2] ADD . /usr/share/nginx/html
 => exporting to image
 => => exporting layers
 => => writing image sha256:68e96482beca25b30acd4805a09d3d26271b466914ec8a9ff215cfab35f3747c
 => => naming to docker.io/library/page:latest
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-grayscale-master\dist> docker images
REPOSITORY   TAG      IMAGE ID       CREATED          SIZE
page         latest   68e96482beca   12 seconds ago   134MB
bootstrap    latest   14fc2c6b8c27   5 minutes ago    6.95MB
mypages      latest   48152f8cc42d   19 minutes ago   136MB
nginx        latest   b8cf2cbeabb9   3 days ago       133MB
```

Running Container of our Custom Image :



```
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-grayscale-master\dist> docker run --name bootstrape -d -p 8001:80 page:latest
6a6ab8b44d93f71bd9578cf5cf93cfabd69444617459a3ff325c280e652a13bc
PS F:\Mitacs_Internship\Codes\Volume_Create_Website\startbootstrap-grayscale-master\dist> docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED          STATUS          PORTS                  NAMES
6a6ab8b44d93   page:latest      "/docker-entrypoint.…"   20 seconds ago   Up 18 seconds   0.0.0.0:8001->80/tcp   bootstrape
5f4705a1233d   mypages:latest   "/docker-entrypoint.…"   11 minutes ago   Up 11 minutes   0.0.0.0:8080->80/tcp   firstpage
```

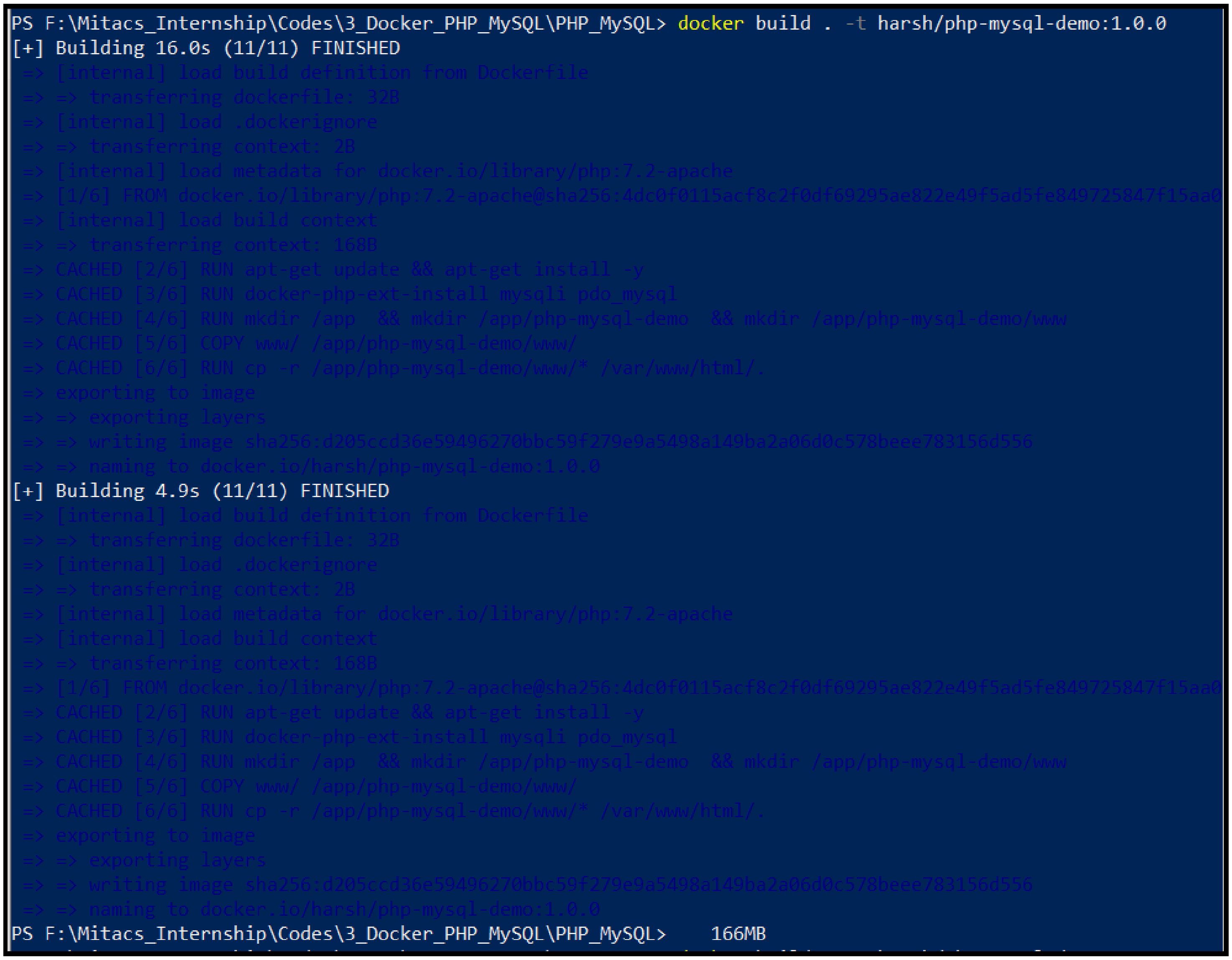
Open Chrome and write Localhost:8001



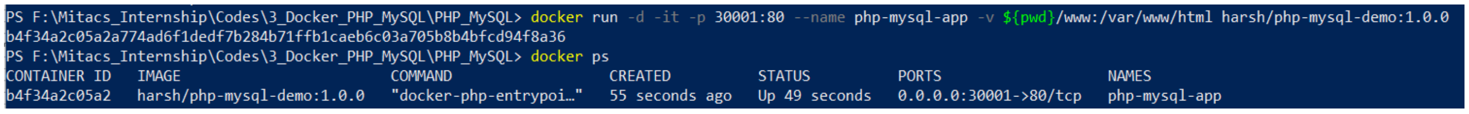
YOUR FAVORITE SOURCE OF FREE BOOTSTRAP THEMES

Start Bootstrap can help you build better websites using the Bootstrap framework! Just download a theme and start customizing, no strings attached!

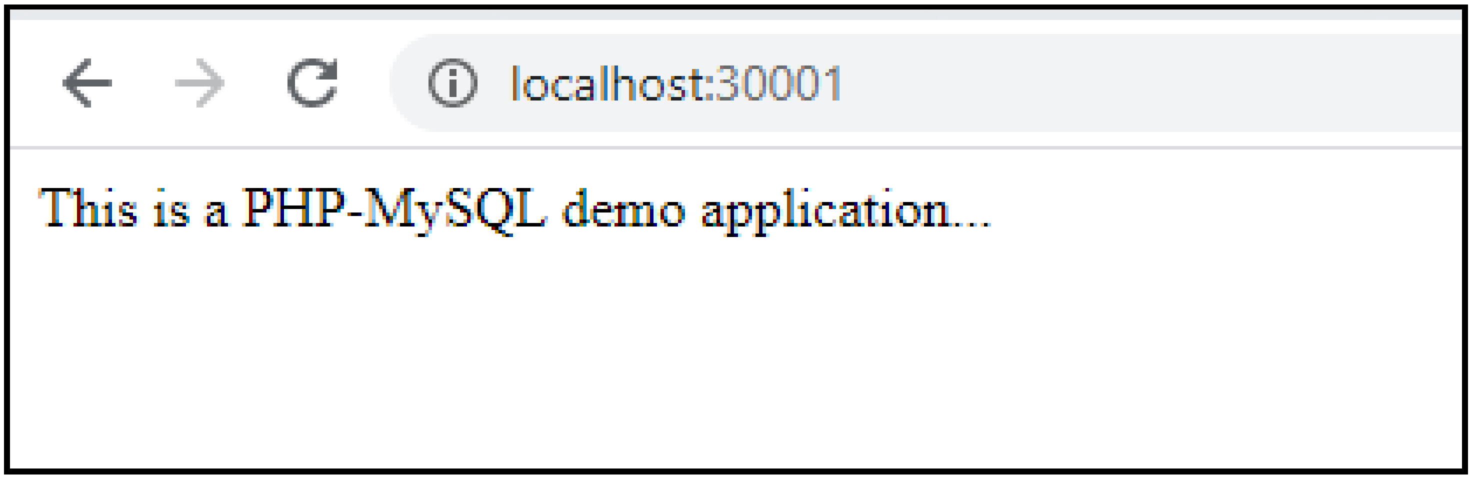
Docker Container for PHP 7 and MySQL Based Application



```
PS F:\Mitacs_Internship\Codes\3_Docker_PHP_MySQL\PHP_MySQL> docker build . -t harsh/php-mysql-demo:1.0.0
[+] Building 16.0s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 32B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/php:7.2-apache
 => [1/6] FROM docker.io/library/php:7.2-apache@sha256:4dc0f0115acf8c2f0df69295ae822e49f5ad5fe849725847f15aa0
 => [internal] load build context
 => => transferring context: 168B
 => CACHED [2/6] RUN apt-get update && apt-get install -y
 => CACHED [3/6] RUN docker-php-ext-install mysqli pdo_mysql
 => CACHED [4/6] RUN mkdir /app  && mkdir /app/php-mysql-demo  && mkdir /app/php-mysql-demo/www
 => CACHED [5/6] COPY www/ /app/php-mysql-demo/www/
 => CACHED [6/6] RUN cp -r /app/php-mysql-demo/www/* /var/www/html/.
 => exporting to image
 => => exporting layers
 => => writing image sha256:d205ccd36e59496270bbc59f279e9a5498a149ba2a06d0c578beee783156d556
 => => naming to docker.io/harsh/php-mysql-demo:1.0.0
[+] Building 4.9s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 32B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/php:7.2-apache
 => [internal] load build context
 => => transferring context: 168B
 => [1/6] FROM docker.io/library/php:7.2-apache@sha256:4dc0f0115acf8c2f0df69295ae822e49f5ad5fe849725847f15aa0
 => CACHED [2/6] RUN apt-get update && apt-get install -y
 => CACHED [3/6] RUN docker-php-ext-install mysqli pdo_mysql
 => CACHED [4/6] RUN mkdir /app  && mkdir /app/php-mysql-demo  && mkdir /app/php-mysql-demo/www
 => CACHED [5/6] COPY www/ /app/php-mysql-demo/www/
 => CACHED [6/6] RUN cp -r /app/php-mysql-demo/www/* /var/www/html/.
 => exporting to image
 => => exporting layers
 => => writing image sha256:d205ccd36e59496270bbc59f279e9a5498a149ba2a06d0c578beee783156d556
 => => naming to docker.io/harsh/php-mysql-demo:1.0.0
PS F:\Mitacs_Internship\Codes\3_Docker_PHP_MySQL\PHP_MySQL>    166MB
```

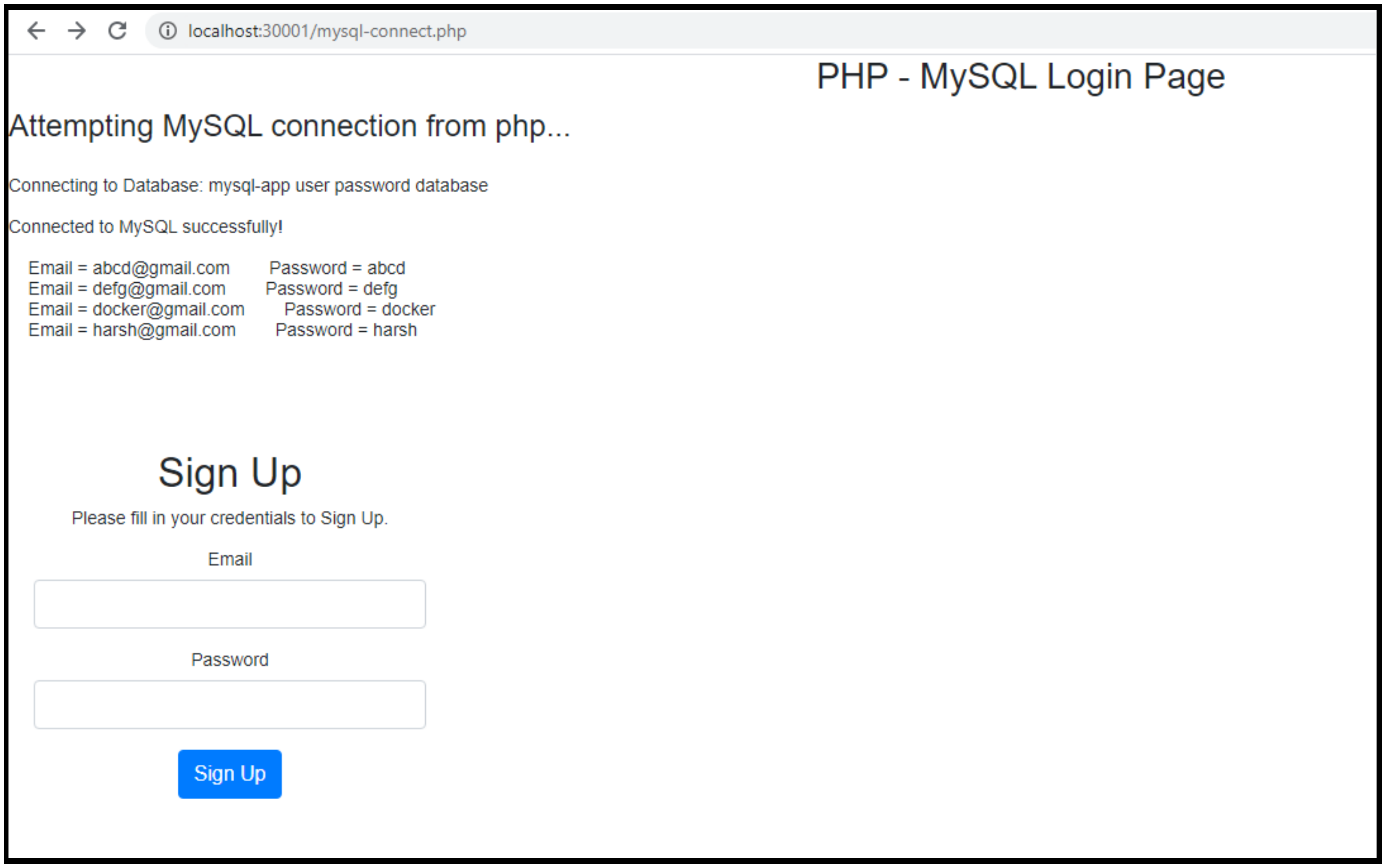


```
PS F:\Mitacs_Internship\Codes\3_Docker_PHP_MySQL\PHP_MySQL> docker run -d -it -p 30001:80 --name php-mysql-app -v ${pwd}/www:/var/www/html harsh/php-mysql-demo:1.0.0
b4f34a2c05a2a774ad6f1dedf7b284b71ffb1caeb6c03a705b8b4bfcd94f8a36
PS F:\Mitacs_Internship\Codes\3_Docker_PHP_MySQL\PHP_MySQL> docker ps
CONTAINER ID   IMAGE                        COMMAND                  CREATED          STATUS          PORTS                   NAMES
b4f34a2c05a2   harsh/php-mysql-demo:1.0.0   "docker-php-entrypoi…"   55 seconds ago   Up 49 seconds   0.0.0.0:30001->80/tcp   php-mysql-app
```



localhost:30001

This is a PHP-MySQL demo application...



localhost:30001/mysql-connect.php

PHP - MySQL Login Page

Attempting MySQL connection from php...

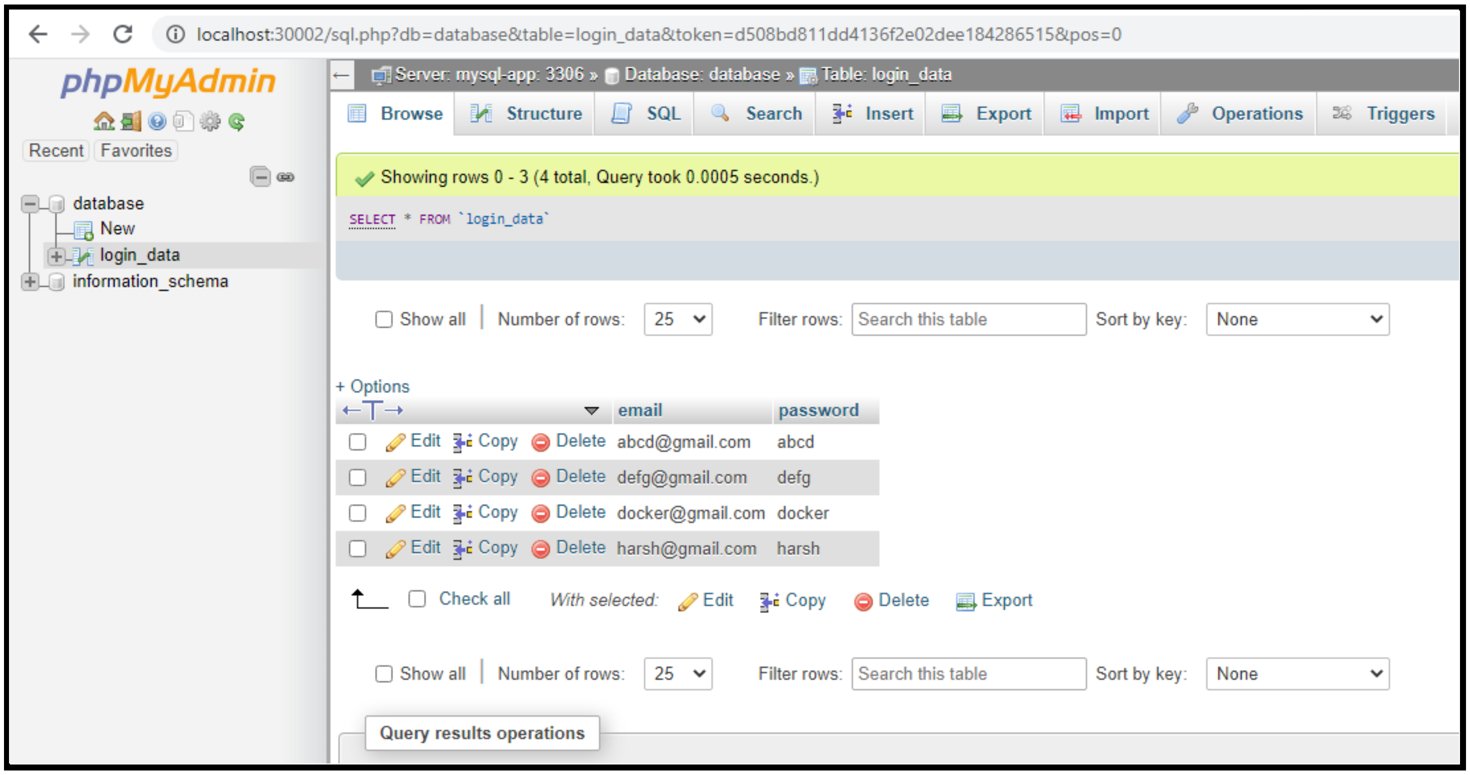
Connecting to Database: mysql-app user password database

Connected to MySQL successfully!

```
Email = abcd@gmail.com      Password = abcd
Email = defg@gmail.com      Password = defg
Email = docker@gmail.com      Password = docker
Email = harsh@gmail.com      Password = harsh
```

Sign Up

Please fill in your credentials to Sign Up.



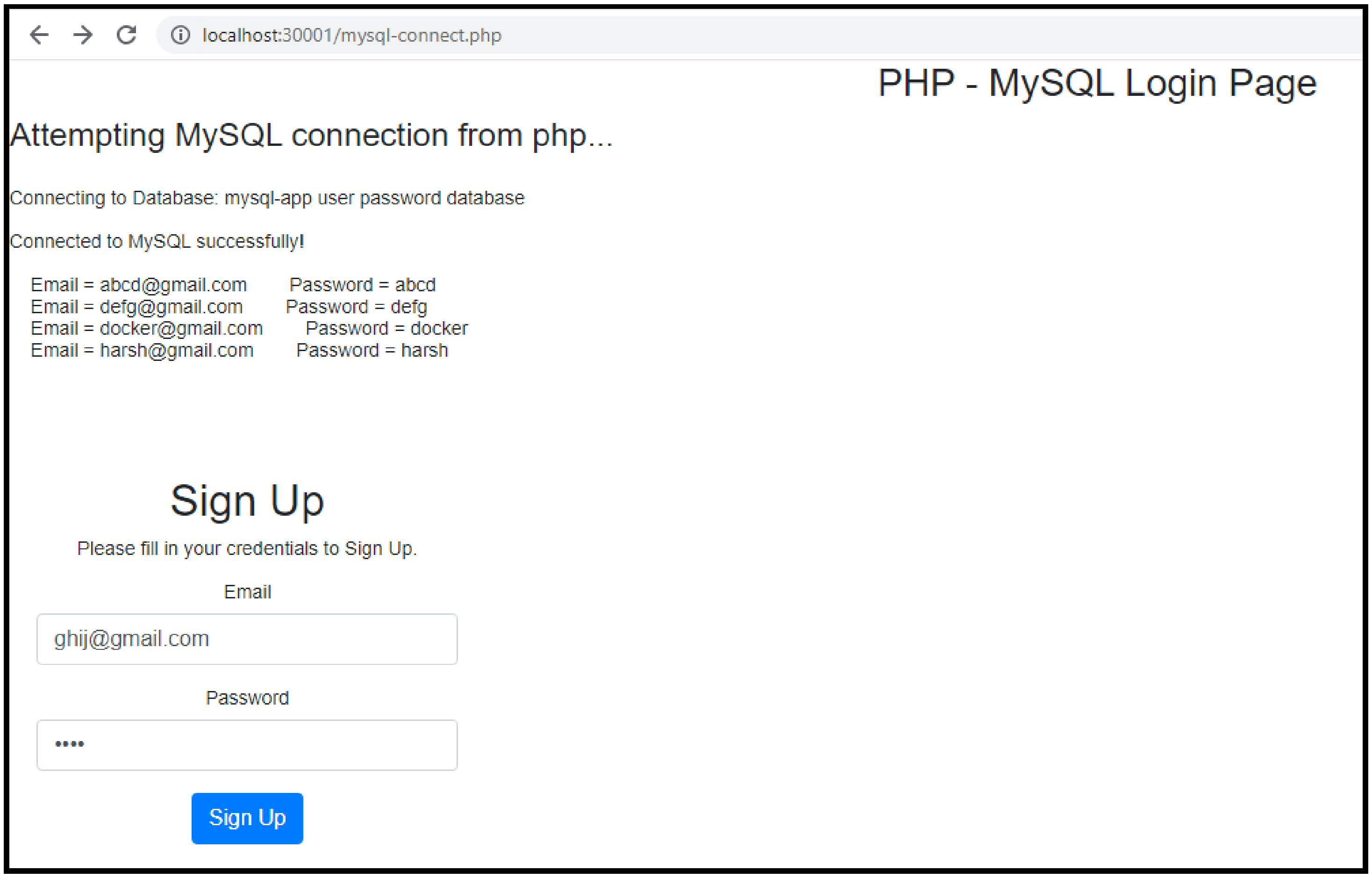
localhost:30002/sql.php?db=database&table=login_data&token=d508bd811dd4136f2e02dee184286515&pos=0

Server: mysql-app: 3306 » Database: database » Table: login_data

Showing rows 0 - 3 (4 total, Query took 0.0005 seconds.)

```
SELECT * FROM `login_data`
```

email	password
abcd@gmail.com	abcd
defg@gmail.com	defg
docker@gmail.com	docker
harsh@gmail.com	harsh



localhost:30001/mysql-connect.php

PHP - MySQL Login Page

Attempting MySQL connection from php...

Connecting to Database: mysql-app user password database

Connected to MySQL successfully!

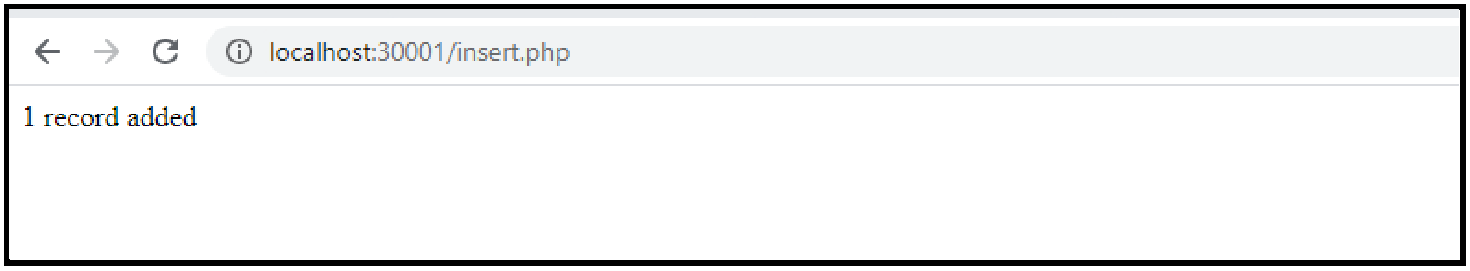
```
Email = abcd@gmail.com      Password = abcd
Email = defg@gmail.com      Password = defg
Email = docker@gmail.com      Password = docker
Email = harsh@gmail.com      Password = harsh
```

Sign Up

Please fill in your credentials to Sign Up.

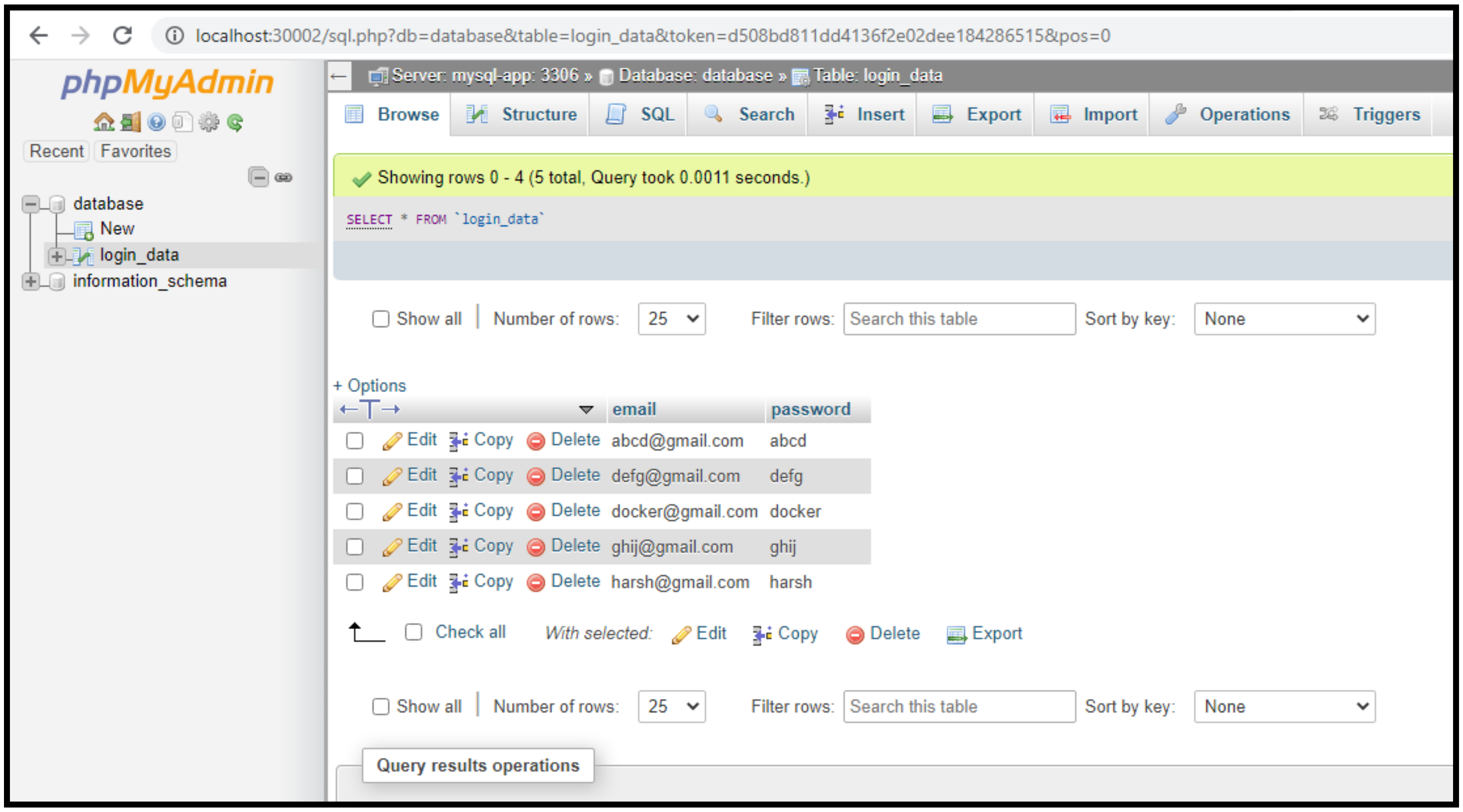
Email: ghij@gmail.com

Password: ••••



localhost:30001/insert.php

```
1 record added
```



localhost:30002/sql.php?db=database&table=login_data&token=d508bd811dd4136f2e02dee184286515&pos=0

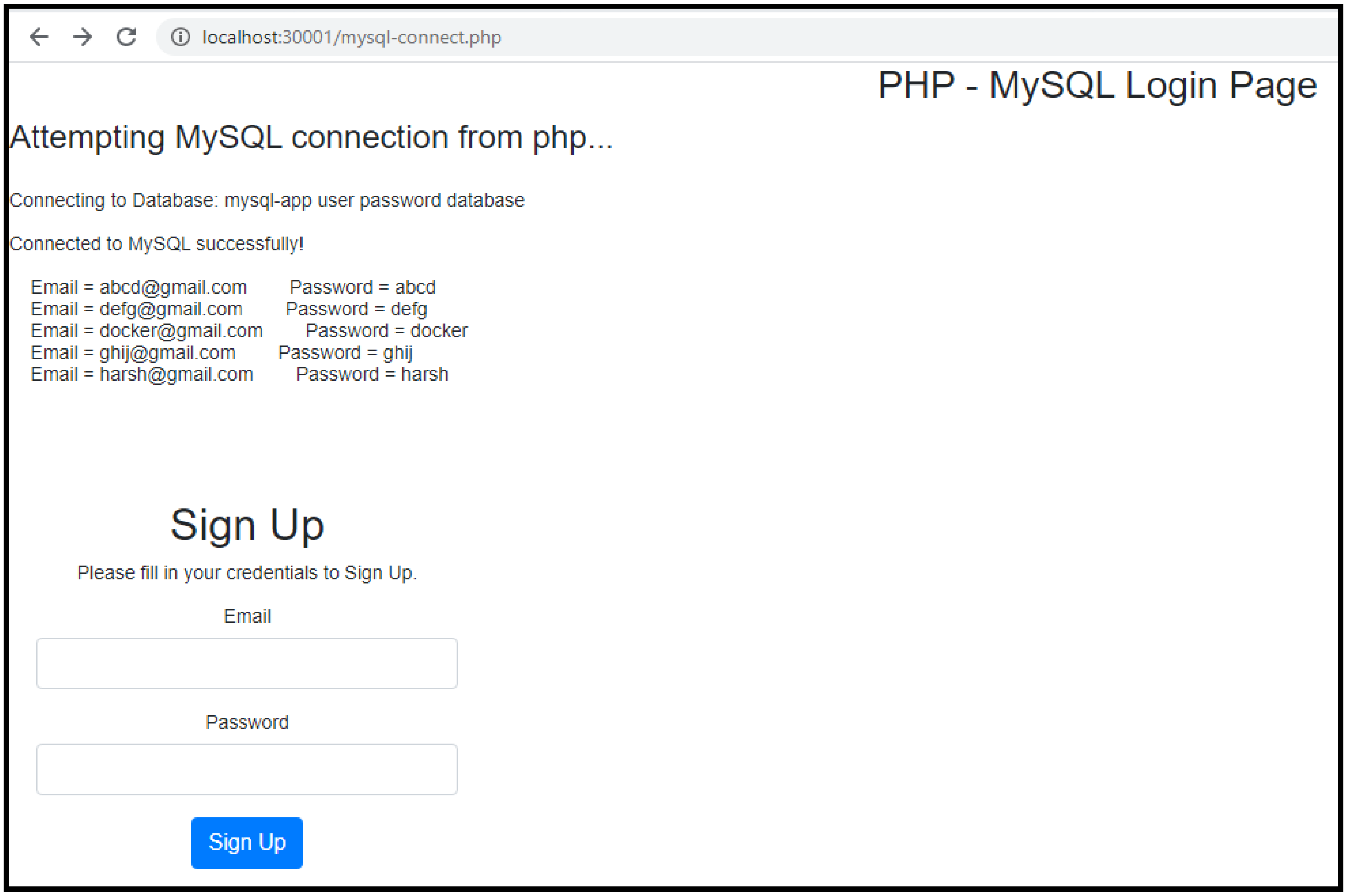
Server: mysql-app: 3306 » Database: database » Table: login_data

Showing rows 0 - 4 (5 total, Query took 0.0011 seconds.)

```
SELECT * FROM `login_data`
```

email	password
abcd@gmail.com	abcd
defg@gmail.com	defg
docker@gmail.com	docker
ghij@gmail.com	ghij
harsh@gmail.com	harsh

phpMyAdmin
Recent Favorites
database
New
login_data
information_schema
Browse Structure SQL Search Insert Export Import Operations Triggers
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
Edit Copy Delete
Check all With selected: Edit Copy Delete Export
Query results operations



localhost:30001/mysql-connect.php

PHP - MySQL Login Page

Attempting MySQL connection from php...

Connecting to Database: mysql-app user password database

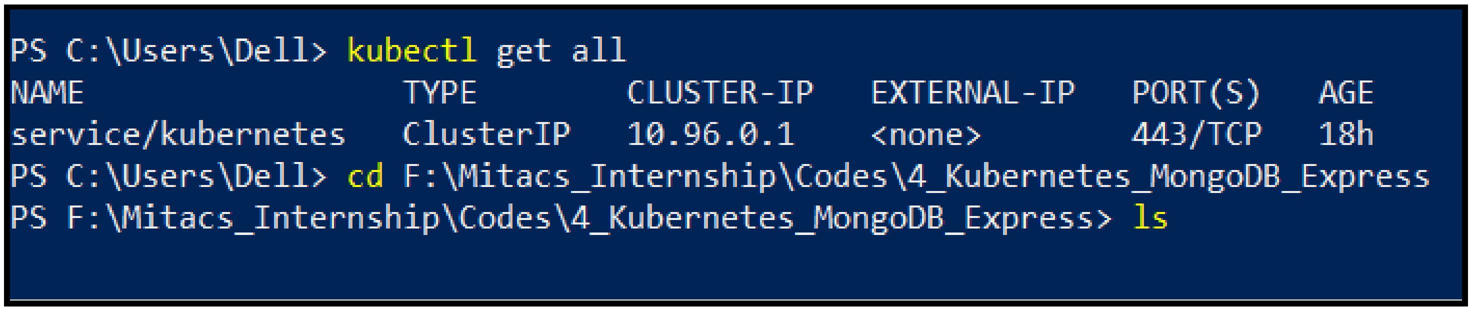
Connected to MySQL successfully!

Email = abcd@gmail.com	Password = abcd
Email = defg@gmail.com	Password = defg
Email = docker@gmail.com	Password = docker
Email = ghij@gmail.com	Password = ghij
Email = harsh@gmail.com	Password = harsh

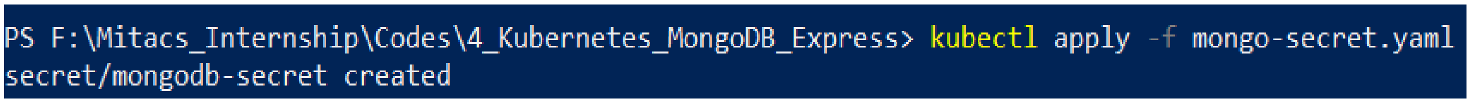
Sign Up

Please fill in your credentials to Sign Up.

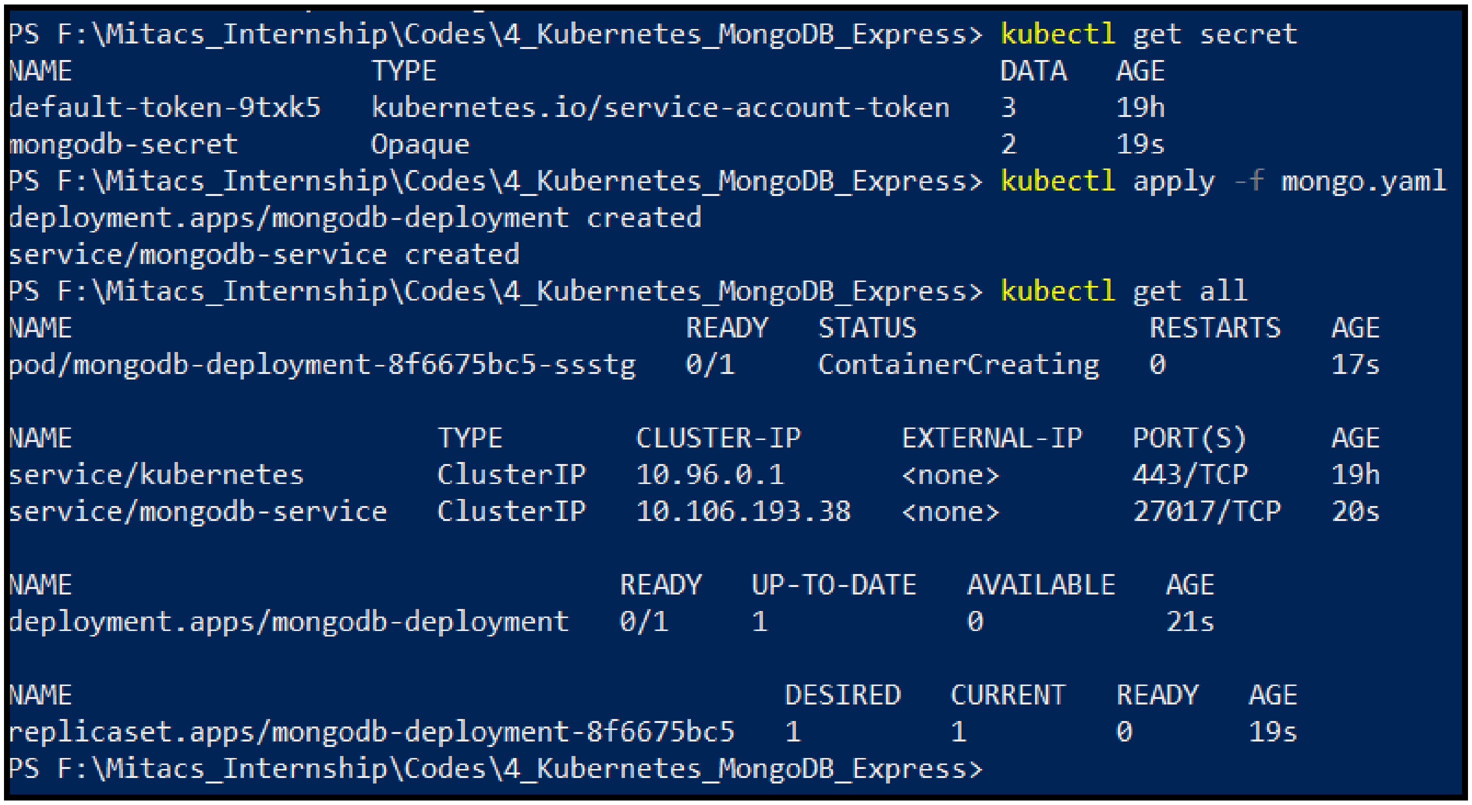
Kubernetes Application



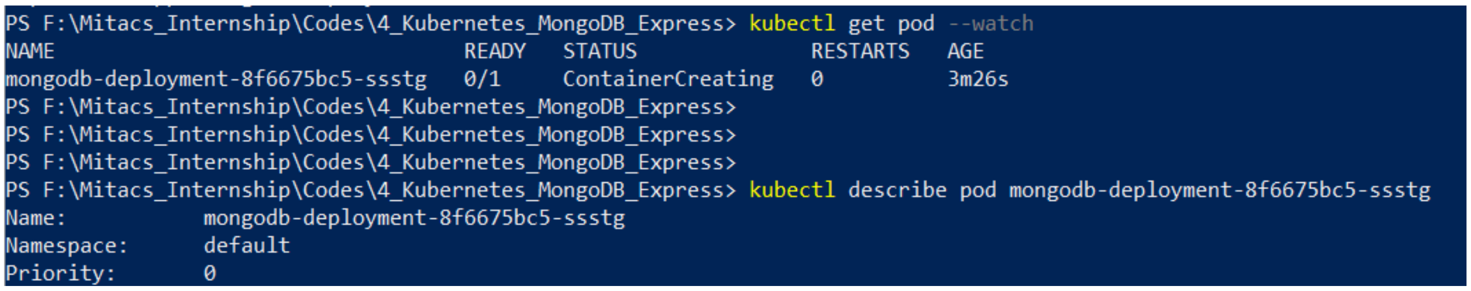
```
PS C:\Users\Dell> kubectl get all
NAME                     TYPE         CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
service/kubernetes       ClusterIP    10.96.0.1     <none>        443/TCP   18h
PS C:\Users\Dell> cd F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> ls
```



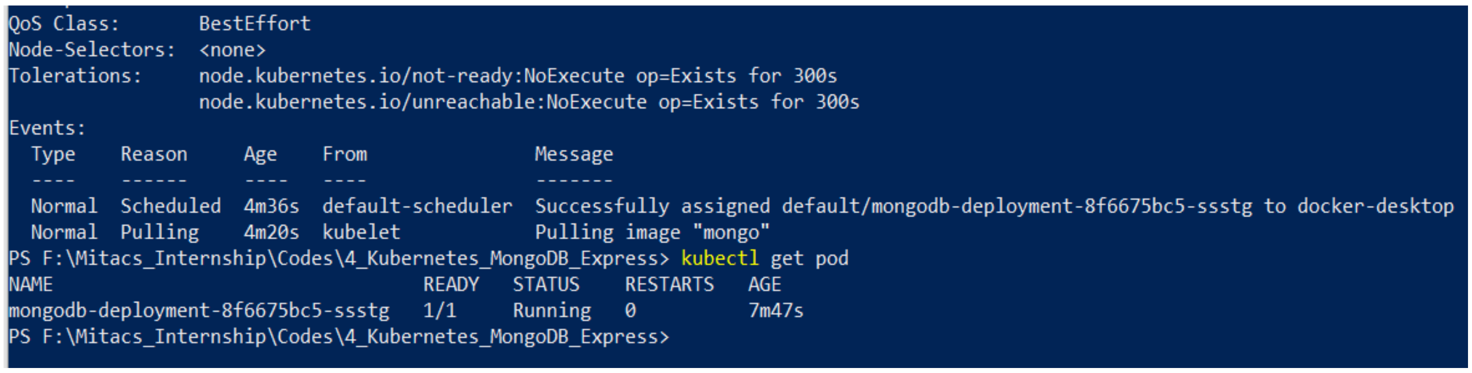
```
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl apply -f mongo-secret.yaml
secret/mongodb-secret created
```



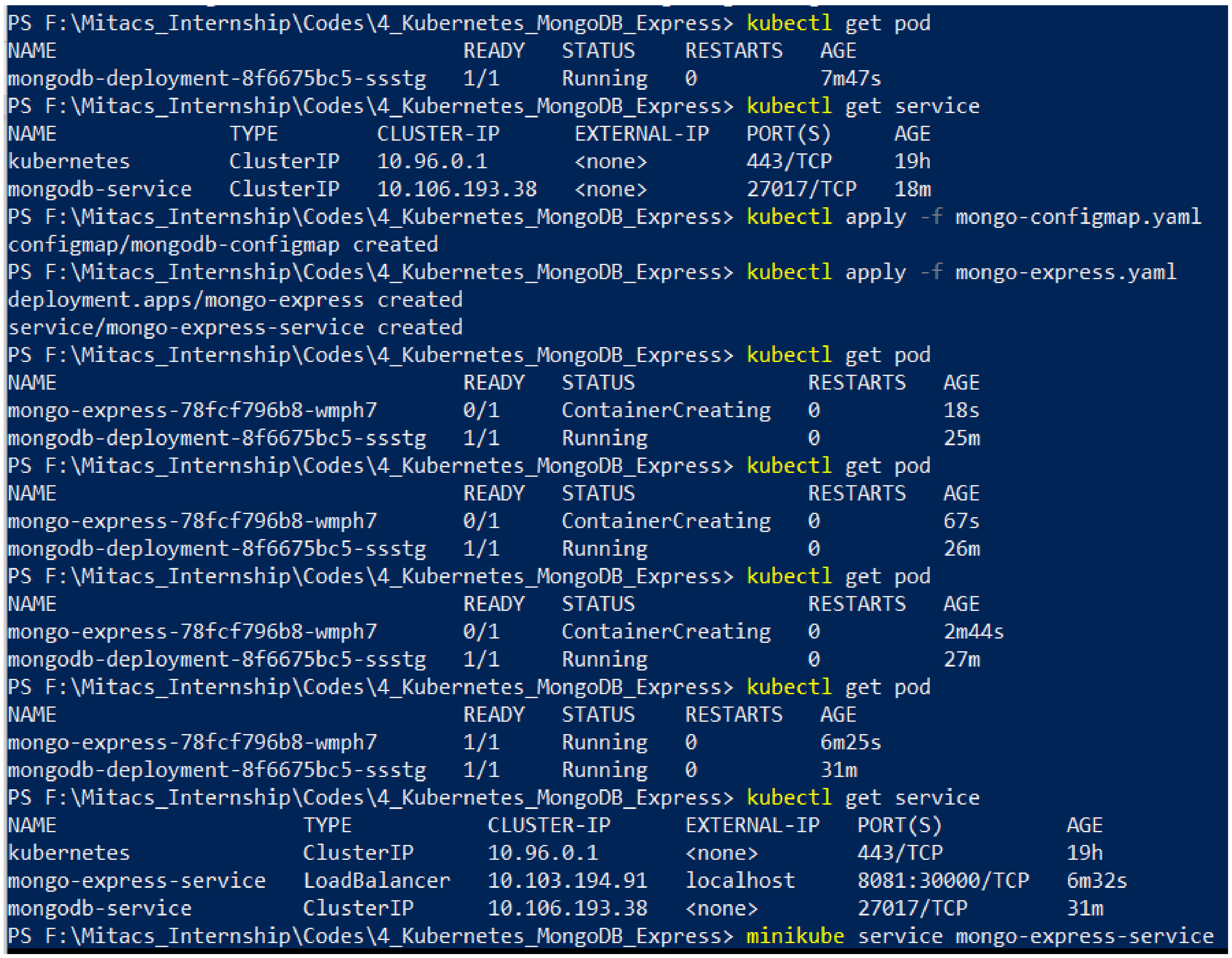
```
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-9txk5   kubernetes.io/service-account-token   3      19h
mongodb-secret        Opaque                                2      19s
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl apply -f mongo.yaml
deployment.apps/mongodb-deployment created
service/mongodb-service created
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get all
NAME                                      READY   STATUS              RESTARTS   AGE
pod/mongodb-deployment-8f6675bc5-ssstg   0/1     ContainerCreating   0          17s

NAME                      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
service/kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP     19h
service/mongodb-service   ClusterIP   10.106.193.38   <none>        27017/TCP   20s

NAME                                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mongodb-deployment   0/1     1            0           21s

NAME                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/mongodb-deployment-8f6675bc5   1         1         0       19s
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express>
```



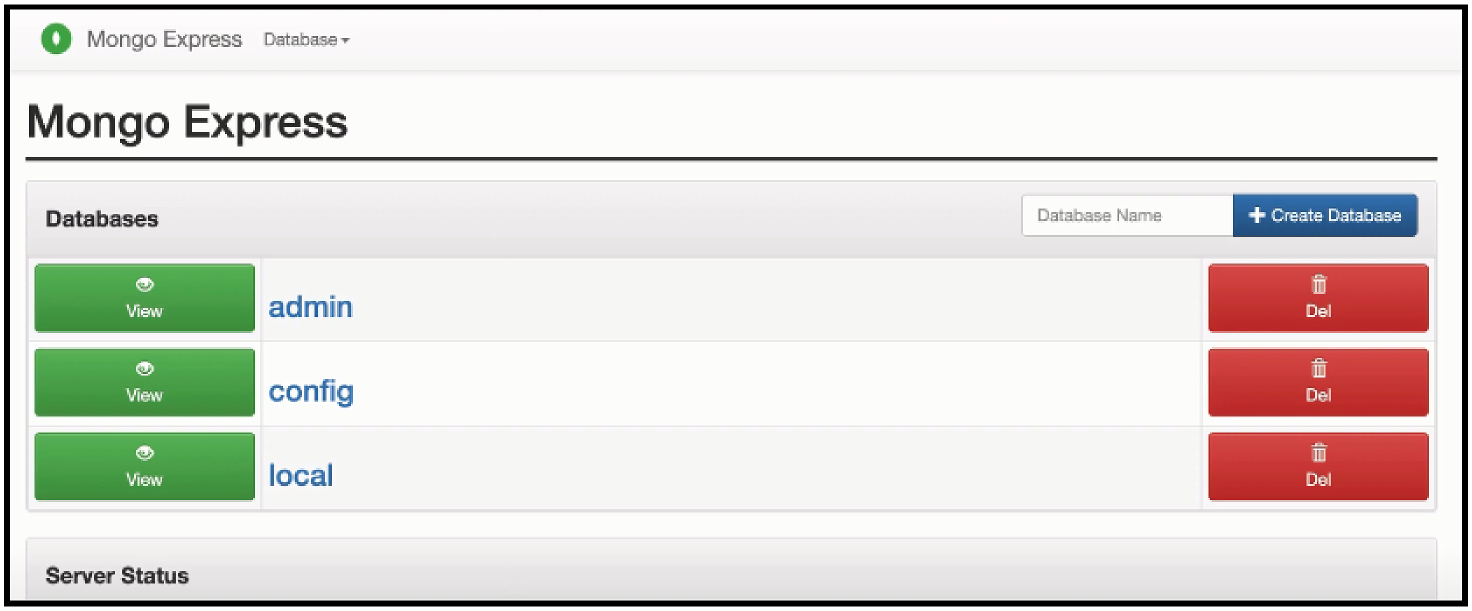
```
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get pod --watch
NAME                                 READY   STATUS              RESTARTS   AGE
mongodb-deployment-8f6675bc5-ssstg   0/1     ContainerCreating   0          3m26s
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express>
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express>
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express>
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl describe pod mongodb-deployment-8f6675bc5-ssstg
Name:         mongodb-deployment-8f6675bc5-ssstg
Namespace:    default
Priority:     0
```



```
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  4m36s  default-scheduler  Successfully assigned default/mongodb-deployment-8f6675bc5-ssstg to docker-desktop
  Normal  Pulling    4m20s  kubelet            Pulling image "mongo"
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get pod
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-deployment-8f6675bc5-ssstg   1/1     Running   0          7m47s
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express>
```



```
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get pod
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-deployment-8f6675bc5-ssstg   1/1     Running   0          7m47s
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get service
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP     19h
mongodb-service   ClusterIP   10.106.193.38   <none>        27017/TCP   18m
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl apply -f mongo-configmap.yaml
configmap/mongodb-configmap created
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl apply -f mongo-express.yaml
deployment.apps/mongo-express created
service/mongo-express-service created
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get pod
NAME                                 READY   STATUS              RESTARTS   AGE
mongo-express-78fcf796b8-wmph7       0/1     ContainerCreating   0          18s
mongodb-deployment-8f6675bc5-ssstg   1/1     Running             0          25m
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get pod
NAME                                 READY   STATUS              RESTARTS   AGE
mongo-express-78fcf796b8-wmph7       0/1     ContainerCreating   0          67s
mongodb-deployment-8f6675bc5-ssstg   1/1     Running             0          26m
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get pod
NAME                                 READY   STATUS              RESTARTS   AGE
mongo-express-78fcf796b8-wmph7       0/1     ContainerCreating   0          2m44s
mongodb-deployment-8f6675bc5-ssstg   1/1     Running             0          27m
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get pod
NAME                                 READY   STATUS    RESTARTS   AGE
mongo-express-78fcf796b8-wmph7       1/1     Running   0          6m25s
mongodb-deployment-8f6675bc5-ssstg   1/1     Running   0          31m
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> kubectl get service
NAME                    TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes              ClusterIP      10.96.0.1       <none>        443/TCP          19h
mongo-express-service   LoadBalancer   10.103.194.91   localhost     8081:30000/TCP   6m32s
mongodb-service         ClusterIP      10.106.193.38   <none>        27017/TCP        31m
PS F:\Mitacs_Internship\Codes\4_Kubernetes_MongoDB_Express> minikube service mongo-express-service
```


Mongo Express
Database
Mongo Express
Databases
Database Name
Create Database
View
admin
Del
View
config
Del
View
local
Del
Server Status