# Docker Tutorial

In this tutorial we will discuss about the Docker and implement the following steps.

## 1) Installing the Docker and running Hello world image.

We will use the windows platform for this tutorial. Download the Docker stable installation file for windows. Before installing the Docker make sure 'Hyper V' featured is turned on windows. You can turn it on by going (search) to "Turn windows features on and off" and clicking on "Hyper-V". By activating the Hyper-V, Virtual box will not work.

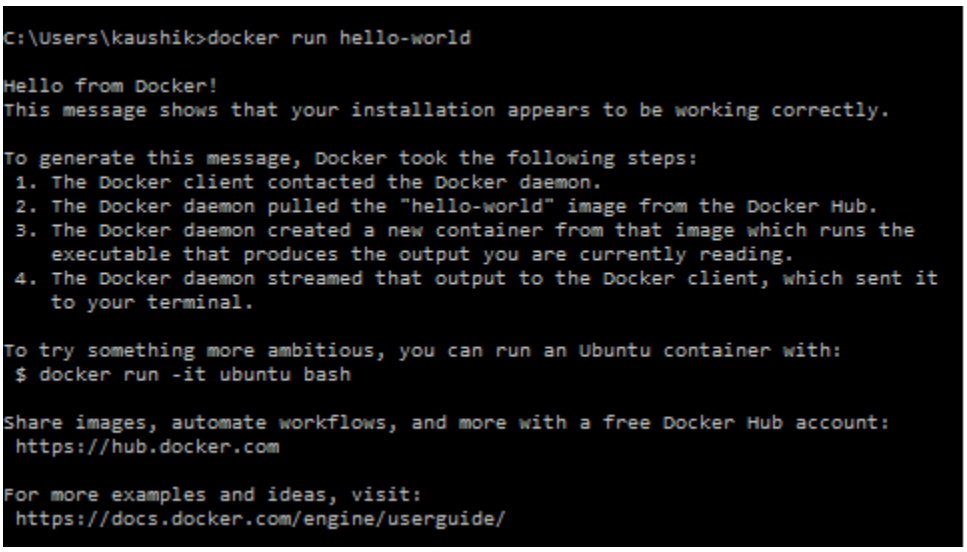
https://download.docker.com/win/stable/InstallDocker.msi

You may have to restart your computer after installation. Open "CMD" and type "docker version" to verify the installation.

Run the hello-world image by using the following in CMD.

"docker run hello-world".

If hello-world image doesn't exist in your system, it will be downloaded.




## 2)Running an image in Docker container.

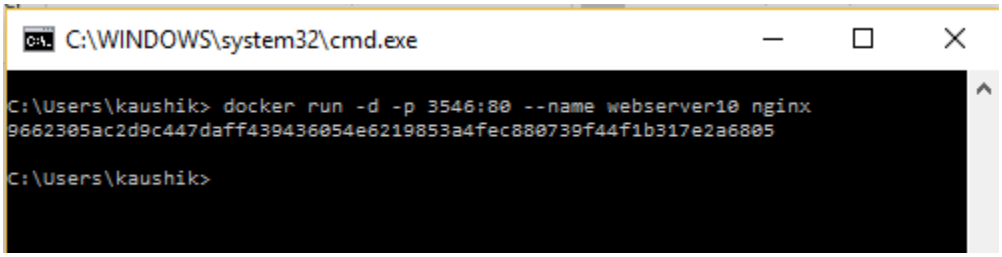
As the next step we will run the nginx web server in Docker container. Nginx image will be downloaded automatically. Type the following command.

"docker **run** -d -p 3546:80 --name webserver nginx"

3546:80 maps the local 3546 port to Docker container port.

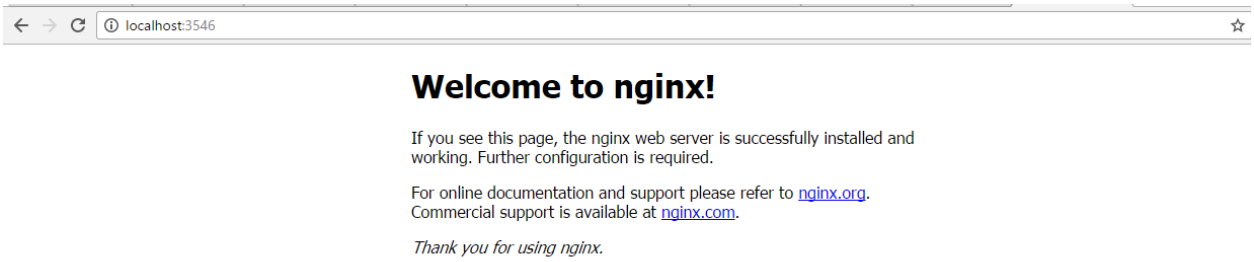
The first argument 'webserver' specifies the name of the container.

The second argument ‘nginx’ specifies the name of the image.




Your nginx webserver will be started. Open the browser and type “localhost:3546”.

You should see the welcome message from nginx server.




3)Build your own image and run it in Docker.

In this step we will create our own image of mongoDB and run it.

Open “CMD” and type “notepad Dockerfile”. Don’t change the name of file and make sure the file name is “Dockerfile” it should not be “Dockerfile.txt”.

In this file we have to write the installation and configuration steps of mongoDB. To ease the process, you can download the file from following link https://drive.google.com/open?id=0B5rpsxEC4Z6NOFFKRXVKTm9zMFE

Type the following command to build the image.

“docker build --tag docker-mongodb .”

Docker-mongodb is the name of your image.

Dot after the Docker-mongodb specifies that Dockerfile is in the current directly.

The output will be as follows.

```
Adding user `mongodb' to group `mongodb' ...
Adding user mongodb to group mongodb
Done.
Setting up mongodb-enterprise-mongos (3.2.10) ...
Setting up mongodb-enterprise-tools (3.2.10) ...
Setting up mongodb-enterprise (3.2.10) ...
Setting up rename (0.20-4) ...
update-alternatives: using /usr/bin/file-rename to provide /usr/bin/rename (rename) in auto mode
Setting up tcpd (7.6.q-25) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Processing triggers for systemd (229-4ubuntu10) ...
 ---> 21274efc9199
Removing intermediate container 6781b2557e11
Step 6 : RUN mkdir -p /data/db
 ---> Running in e415a4e8c39e
 ---> 0e7f3d7ca736
Removing intermediate container e415a4e8c39e
Step 7 : EXPOSE 27017
 ---> Running in 4de0f387bb9c
 ---> 8173ce25ebdb
Removing intermediate container 4de0f387bb9c
Step 8 : ENTRYPOINT /usr/bin/mongod
 ---> Running in e69aa96d8b51
 ---> 3bd0b9af710d
Removing intermediate container e69aa96d8b51
Successfully built 3bd0b9af710d
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files an
d directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and
 reset permissions for sensitive files and directories.

C:\Users\kaushik\DockerBuild>
```

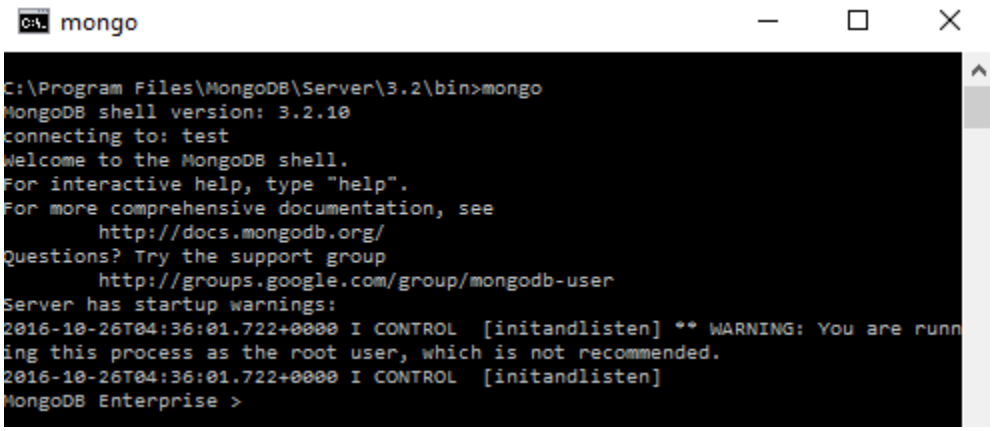
Run the newly created mongoDB image by using the following command.

“docker run -p 27017:27017 --name mongo_instance_001 -d docker-mongodb”

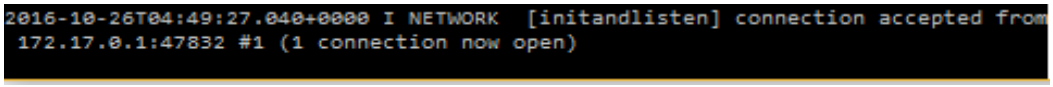
The below figure shows the output of that command. We can see the mongoDB running.

```
2016-10-26T04:35:59.926+0000 I CONTROL  [initandlisten]     distarch: x86_64
2016-10-26T04:35:59.926+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2016-10-26T04:35:59.926+0000 I CONTROL  [initandlisten] options: {}
2016-10-26T04:35:59.934+0000 I STORAGE  [initandlisten] wiredtiger_open config:
create,cache_size=1G,session_max=20000,eviction=(threads_max=4),config_base=fals
e,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snapp
y),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),stati
stics_log=(wait=0),
2016-10-26T04:36:01.722+0000 I CONTROL  [initandlisten] ** WARNING: You are runn
ing this process as the root user, which is not recommended.
2016-10-26T04:36:01.722+0000 I CONTROL  [initandlisten]
2016-10-26T04:36:01.723+0000 I FTDC     [initandlisten] Initializing full-time d
iagnostic data capture with directory '/data/db/diagnostic.data'
2016-10-26T04:36:01.724+0000 I NETWORK  [HostnameCanonicalizationWorker] Startin
g hostname canonicalization worker
2016-10-26T04:36:07.605+0000 I NETWORK  [initandlisten] waiting for connections
on port 27017
```

To verify that the image is running exactly download the mongoDB (Server edition will also work). Navigate through installation path in CMD and “mongo”. Your mongo client will be connected to mongoDB in Docker.




Output of mongo client




Output of mongoDB in Docker