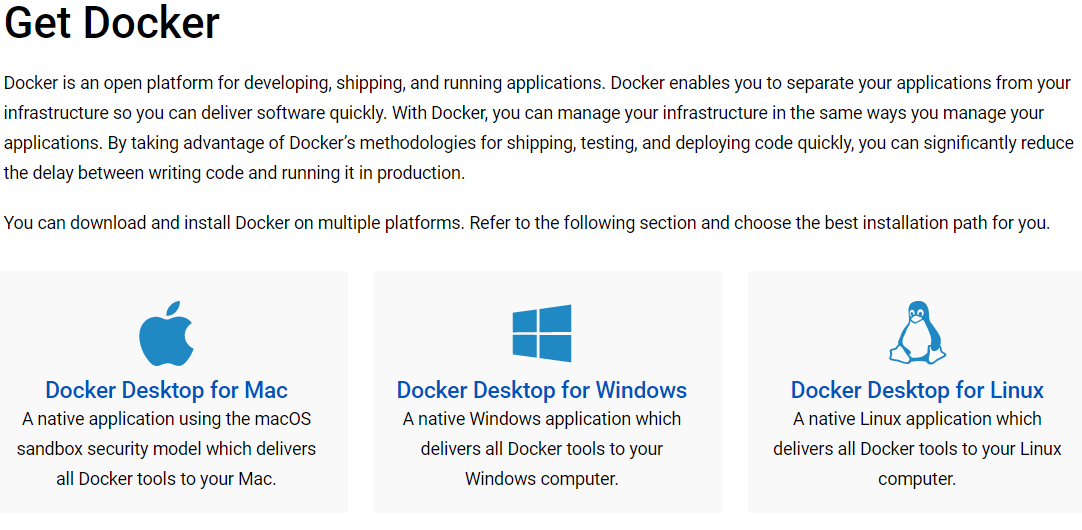
**Docker commands Documentation**

1. Create login in Docker hub
2. Install docker desktop for windows



**Docker Commands**

C:\Users\srila>docker

* It will display weather the docker is installed or not

C:\Users\srila>docker – version

* It will display the version of the docker

C:\Users\srila>docker help

* It will display all the docker commands list

**Nginx**

C:\Users\srila>docker pull nginx

[syntax to pull image: docker pull <image name>

* It will pull all the layers included in that image.

C:\Users\srila>docker run -d name=nginx1 -p 8080:80 nginx

98e9c120cf3e4828a7649094101c72d90c4924358a0d3f5b806e99f1c8183846

Here, -d is for detach…to run the process in background

Name tag= give a name to container

-p tag=to attach port number to the container

Here, 8080 is the default port number for nginx web server

Last, nginx is the image name.

* Localhost:8080 in chrome…will display nginx home page.

**Mongo db. and Mongo Express**

C:\Users\srila>docker network create mongonet

* To connect two containers with networks.

C:\Users\srila>docker run -d --name samplemongo -e MONGO\_INITDB\_ROOT\_USERNAME=test -e MONGO\_INITDB\_ROOT\_PASSWORD=bluespire -p 27017:27017 --net mongonet mongo

218ed29263c92dd443aa306c8d8d651219cdb0ce970cc358faa9a1ae9e31b7dd

C:\Users\srila>docker -d --name mongoexpui -e ME\_CONFIG\_MONGODB\_ADMINUSERNAME=test -e ME\_CONFIG\_MONGODB\_ADMINPASSWORD=bluespire -e ME\_CONFIG\_MONGODB\_SERVER=samplemongo -p 8081:8081 --net mongonet mongo-express

8a26f257dcce7e9b7f5f46d41afc396596b3766f9e99ee8093e24d301f9f5099

* -e tag for mapping environment variables like …. username, password, network.
* Check localhost:8081 in chrome…will display mongo UI.

C:\Users\srila>docker start <container name>

* Will starts the previously stopped container.

C:\Users\srila>docker stop <container name>

* Will stops the particular container.

**Demo app - developing with Docker**

|  |
| --- |
| This demo app shows a simple user profile app set up using   * index.html with pure js and css styles * nodejs backend with express module * mongodb for data storage   All components are docker-based  **With Docker**  **To start the application**  Step 1: Create docker network  docker network create mongo-network  Step 2: start mongodb  docker run -d -p 27017:27017 -e MONGO\_INITDB\_ROOT\_USERNAME=admin -e  MONGO\_INITDB\_ROOT\_PASSWORD=password --name mongodb --net mongo-network mongo  Step 3: start mongo-express  docker run -d -p 8081:8081 -e ME\_CONFIG\_MONGODB\_ADMINUSERNAME=admin  -e ME\_CONFIG\_MONGODB\_ADMINPASSWORD=password --net mongo-network --name mongo-express  -e ME\_CONFIG\_MONGODB\_SERVER=mongodb mongo-express  *NOTE: creating docker-network in optional. You can start both containers in a default network.*  *In this case, just emit --net flag in docker run command*  Step 4: open mongo-express from browser  [http://localhost:8081](http://localhost:8081/)  Step 5: create user-account *db* and users *collection* in mongo-express  Step 6: Start your nodejs application locally - go to app directory of project  npm install  node server.js  Step 7: Access you nodejs application UI from browser  [http://localhost:3000](http://localhost:3000/)  **With Docker Compose**  **To start the application**  Step 1: start mongodb and mongo-express  docker-compose -f docker-compose.yaml up  *You can access the mongo-express under localhost:8080 from your browser*  Step 2: in mongo-express UI - create a new database "my-db"  Step 3: in mongo-express UI - create a new collection "users" in the database "my-db"  Step 4: start node server  npm install  node server.js  Step 5: access the nodejs application from browser  [http://localhost:3000](http://localhost:3000/)  **To build a docker image from the application**  docker build -t my-app:1.0 .  The dot "." at the end of the command denotes location of the Dockerfile.  **Adding html page to nginx**  C:\Users\srila>docker volume create sitedata.   * Volumes will help to data persistency. * Root path for nginx webserver volume…..---/usr/share/nginx/html   C:\Users\srila>docker run -d –name:testnginx -p 8080:8080 -v sitedata:/usr/share/nginx/html nginx  C:\Users\srila>docker exec -it nginxtest bash   * It will change to root. * rootpath@.......:/usr/share/nginx/html#   ..  ..:/usr/share/nginx/html#apt update  ..:/usr/share/nginx/html#apt install vim  ..:/usr/share/nginx/html#vim index.html   * -- here in index.html you can edit the heading of the website as you want. * -- for this press escape+I [for insert] * -- to exit from html page-🡪 escape:shift+:-----wq for write and quit.   **Docker compose**  version: "2.10.2"  services:    mongodb:      image: mongo:6      environment:       - MONGO\_INITDB\_ROOT\_USERNAME=admin       - MONGO\_INITDB\_DATABASE=auth       - MONGO\_INITDB\_ROOT\_PASSWORD=pass      ports:       - '27017:27017'    mongo-express:      container\_name: mongo-express      image: mongo-express:0.54.0      depends\_on:        - mongodb      environment:       - ME\_CONFIG\_MONGODB\_ADMINUSERNAME=admin       - ME\_CONFIG\_MONGODB\_ADMINPASSWORD=pass       - ME\_CONFIG\_MONGODB\_SERVER=mongodb       - ME\_CONFIG\_BASICAUTH\_USERNAME=admin       - ME\_CONFIG\_BASICAUTH\_PASSWORD=ihavealongpassword      ports:       - '8081:8081'   * C:\users\srila>cd downloads (to path where the yaml file is located) * C:\downloads>docker compose -f <yamlfile name> up. * Execute the above yaml file to execute multi container mongodb. * Localhost:8081 in chrome to view the webpage.   **Jenkins** |
| C:\Users\srila>docker create volume sample  To know the OS name  C:\Users\srila>cat /etc/issue  # for Jenkins and nodejs installation we have to check the Os name.  C:\Users\srila>docker run -d –name=jenkinstest -p 8080:8080 -v sample:/var/Jenkins\_home Jenkins/Jenkins:lts |
|  |