Create your own NGINX docker image(not using official image)

- **Create a Dockerfile for NGINX image**
- **Edit file /usr/share/nginx/html/index.html and add a custom index.html through docker file**
- **Create a container from Nginx image and publish port 8080**
- **Exec bash on it and install nano**
- **Check on localhost:8080 changes**
- **Push image to docker hub in your docker registry, ECR, GCR**

Steps to create own nginx docker image

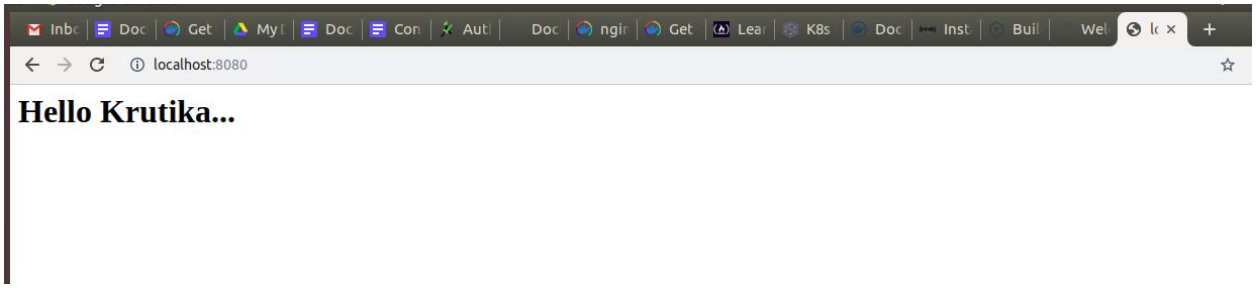
1. Create a Dockerfile and add the commands to install nginx
2. Create the image using the command
   docker build -t kruq2:latest .
3. After the image is created, build a container and map port 8080 to it
   docker run -dit -p 8080:80 kruq2:latest

Dockerfile

```
#Using UBUNTU OS
FROM ubuntu
MAINTAINER Krutika

#Update and install nginx
RUN apt-get update \
   && apt-get install nginx -y \
   && echo "daemon off;" >> /etc/nginx/nginx.conf
RUN mkdir -p /run/nginx

#Added index.html to the given path
ADD index.html /usr/share/nginx/html/

#the default file here changes the root to the path stated in the last command from /var/html
ADD default /etc/nginx/sites-available/

EXPOSE 80
CMD ["nginx"]
```

Default webpage:

Custom webpage:



Display running container

krutika@Quantiphi-930:~/Desktop/web$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
eb4c0b51fe48 kruq2:latest "nginx" About a minute ago Up About a minute 0.0.0.0:8080->80/tcp silly_hofstadter

Exec bash on container and installing nano

krutika@Quantiphi-930:~/Desktop/web$ sudo docker exec -it eb4c0b51fe48 bash
root@eb4c0b51fe48:/# apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 231 kB of archives.
After this operation, 778 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 nano amd64 2.9.3-2 [231 kB]
Fetched 231 kB in 0s (532 kB/s)

Pushing image to Docker hub

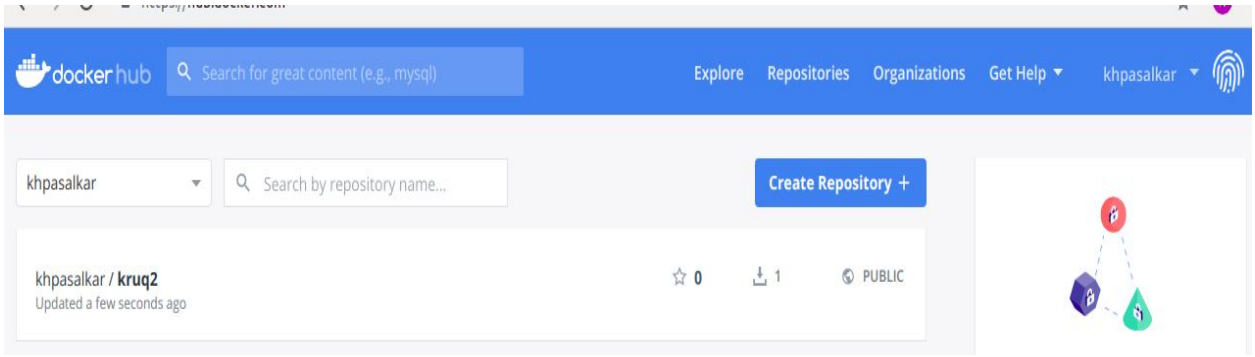
krutika@Quantiphi-930:~/Desktop/web$ **sudo docker login docker.io**

Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/krutika/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

Tagging image
krutika@Quantiphi-930:~/Desktop/web$ **sudo docker tag eafd3aca1748 khpasalkar/kruq2:practice1**

Pushing image to docker hub
krutika@Quantiphi-930:~/Desktop/web$ s**udo docker push khpasalkar/kruq2**
The push refers to repository [docker.io/khpasalkar/kruq2]
a5c373908fdc: Pushed
c0352faebfa5: Pushed
5f5344ba7061: Pushed
d2ff4f2b3997: Pushed
b079b3fa8d1b: Mounted from library/ubuntu
a31dbd3063d7: Mounted from library/ubuntu
c56e09e1bd18: Mounted from library/ubuntu
543791078bdb: Mounted from library/ubuntu
practice1: digest:
sha256:099ee20f8ea7d20b838712ef7b0fa0e8e6b5deb030dd578c6c2fa8594d46ed11 size:
1986



<u>Pushing image to ECR</u>
1. Create an instance and ssh inside that instance

krutika@Quantiphi-930:~/Downloads$ ssh -i "k-terra.pem"
ec2-user@ec2-100-24-4-217.compute-1.amazonaws.com

2.Update the installed packages and package cache on instance

[ec2-user@ip-172-31-94-228 ~]$ sudo yum update -y

3. Install the most recent Docker Community Edition package.

[ec2-user@ip-172-31-94-228 ~]$ sudo amazon-linux-extras install docker
Installing docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker
10 metadata files removed
4 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpac

4. Start the Docker service.

[ec2-user@ip-172-31-94-228 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service

5. Add the ec2-user to the docker group so you can execute Docker commands without using sudo.

[ec2-user@ip-172-31-94-228 ~]$ sudo usermod -a -G docker ec2-user

6. Logging into dockerhub

[ec2-user@ip-172-31-94-228 ~]$ sudo docker login docker.io
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: khpasalkar
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

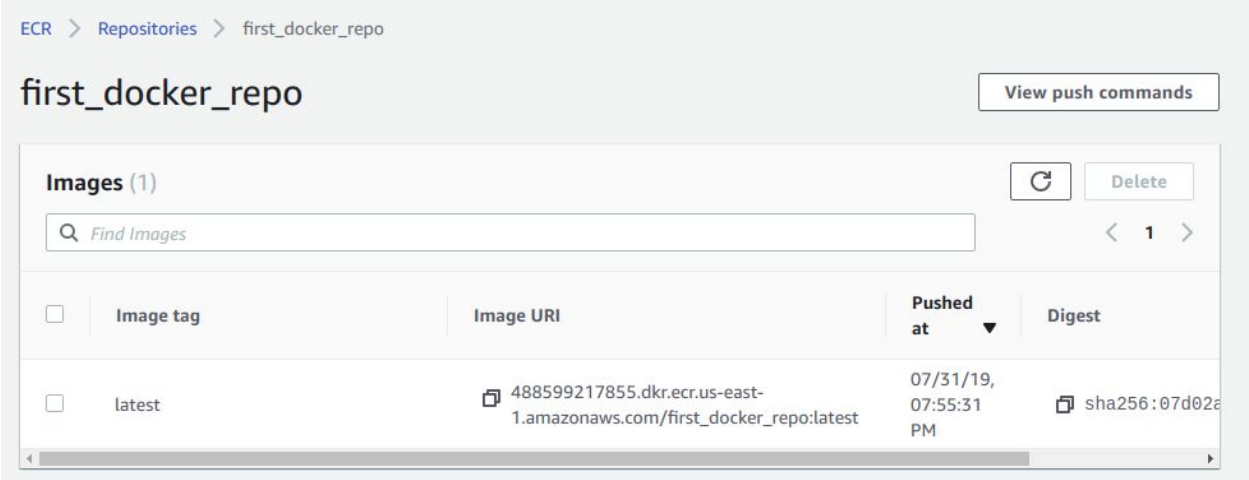
7. Pull the docker image from dockerhub into the ec2 instance

[ec2-user@ip-172-31-94-228 ~]$ sudo docker pull khpasalkar/kruq2:practice1
practice1: Pulling from khpasalkar/kruq2
7413c47ba209: Pull complete
0fe7e7cbb2e8: Pull complete
1d425c982345: Pull complete
344da5c95cec: Pull complete
0bb049f9b147: Pull complete
19eaee407d0c: Pull complete
9e0deef4428e: Pull complete
880a6f1340b5: Pull complete
Digest: sha256:099ee20f8ea7d20b838712ef7b0fa0e8e6b5deb030dd578c6c2fa8594d46ed11
Status: Downloaded newer image for khpasalkar/kruq2:practice1

8. Retrieve the login command to use to authenticate your Docker client to your registry.
[ec2-user@ip-172-31-94-228 ~]$ (aws ecr get-login --no-include-email --region us-east-1)

9. tag your image so you can push the image to the repository
[ec2-user@ip-172-31-94-228 ~]$ sudo docker tag khpasalkar/kruq2:practice1
488599217855.dkr.ecr.us-east-1.amazonaws.com/first_docker_repo:latest

10. Run the following command to push this image to created AWS repository
[ec2-user@ip-172-31-94-228 ~]$ sudo docker push
488599217855.dkr.ecr.us-east-1.amazonaws.com/first_docker_repo:latest



Pushing image to GCR

1. Configured Docker to use gcloud as a credential helper

krutika@Quantiphi-930:~$ gcloud auth configure-docker
The following settings will be added to your Docker config file

located at [/home/krutika/.docker/config.json]:
```
 {
  "credHelpers": {
    "gcr.io": "gcloud",
    "us.gcr.io": "gcloud",
    "eu.gcr.io": "gcloud",
    "asia.gcr.io": "gcloud",
    "staging-k8s.gcr.io": "gcloud",
    "marketplace.gcr.io": "gcloud"
  }
}
```

Do you want to continue (Y/n)? y

Docker configuration file updated.

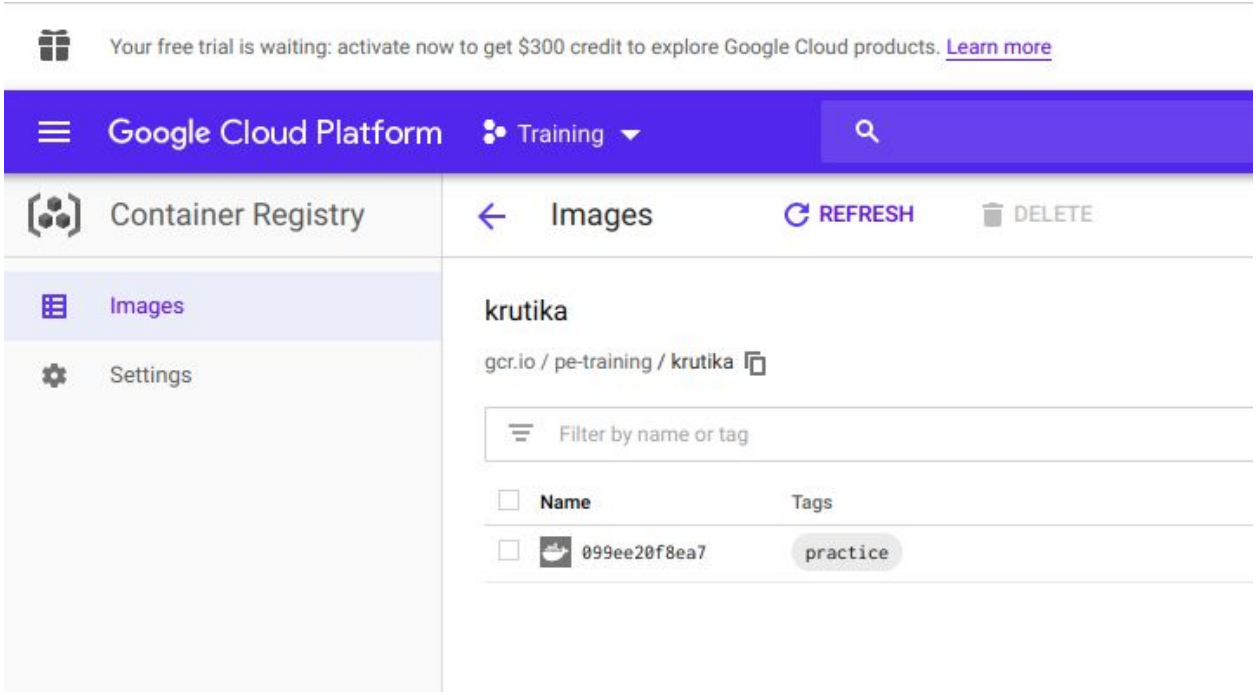
```
krutika@Quantiphi-930:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS                    NAMES
eb4c0b51fe48        kruq2:latest        "nginx"             5 hours ago         Up 5 hours
0.0.0.0:8080->80/tcp   silly_hofstadter
```

2. Tag the local image with the registry name by using the below command

```
krutika@Quantiphi-930:~$ docker tag kruq2:latest gcr.io/pe-training/krutika:practice
```

3. Push the tagged image to Container Registry

```
krutika@Quantiphi-930:~$ docker push gcr.io/pe-training/krutika:practice
The push refers to repository [gcr.io/pe-training/krutika]
a5c373908fdc: Pushed
c0352faebfa5: Pushed
5f5344ba7061: Pushed
d2ff4f2b3997: Pushed
b079b3fa8d1b: Layer already exists
a31dbd3063d7: Layer already exists
c56e09e1bd18: Layer already exists
543791078bdb: Layer already exists
practice: digest:
sha256:099ee20f8ea7d20b838712ef7b0fa0e8e6b5deb030dd578c6c2fa8594d46ed11 size:
1986
```


Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more
Google Cloud Platform
Training
Container Registry
Images
Settings
Images
REFRESH
DELETE
krutika
gcr.io / pe-training / krutika
Filter by name or tag
Name
Tags
099ee20f8ea7
practice