Lesson-End Project

Creating a Docker Image and Replicated Service on a Swarm

Project agenda: To create a Docker image, push it to Docker Hub, and deploy a replicated service on a Swarm cluster to ensure high availability and simplifying the management of application updates

Description: Your company is experiencing a need for scalable and reliable deployment of its applications. To address this, you are undertaking a project to implement Docker containerization and orchestration using Docker Swarm, facilitating efficient management and deployment of services across multiple nodes.

Tools required: Docker and Ubuntu OS

Prerequisites: Docker Hub account

Expected deliverables: Dockerized application managed on a Swarm cluster

Steps to be followed:

1. Create a Dockerfile
2. Tag the image
3. Push the image to Docker Hub
4. Create a container
5. Check nodes in Swarm where the manager node is running
6. Create a service and check the default container

Step 1: Create a Dockerfile

1.1 Create a project folder and navigate to it using the following command:

mkdir gp3

cd gp3

```
sakshiguptasimp@ip-172-31-27-122:~$ mkdir gp3
sakshiguptasimp@ip-172-31-27-122:~$ cd gp3
```

1.2 Create a Dockerfile using the following command:

nano Dockerfile

```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ nano Dockerfile
```

1.3 Add the following configurations to the Dockerfile:

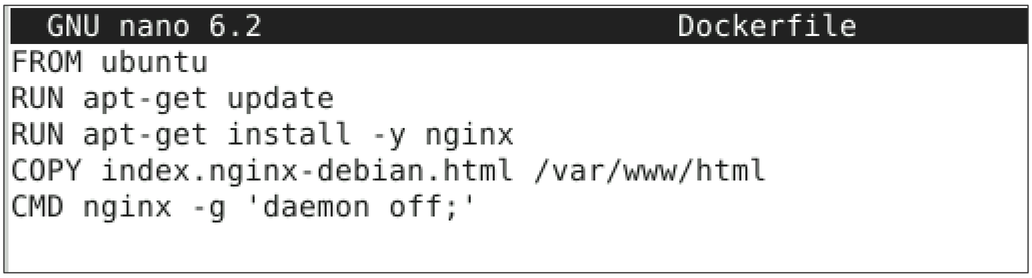
FROM ubuntu

RUN apt-get update

RUN apt-get install -y nginx

COPY index.nginx-debian.html /var/www/html

CMD nginx -g 'daemon off;'



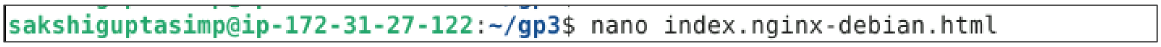
```
  GNU nano 6.2                                  Dockerfile
FROM ubuntu
RUN apt-get update
RUN apt-get install -y nginx
COPY index.nginx-debian.html /var/www/html
CMD nginx -g 'daemon off;'
```

This Dockerfile sequence sets up a container based on the Ubuntu image, updates the package list, installs Nginx, copies a custom HTML file to the web server directory, and starts Nginx in the foreground.

Note: To save the file press **Ctrl+X**, then **Y**, and finally **Enter**

1.4 Create an index file using the following command:

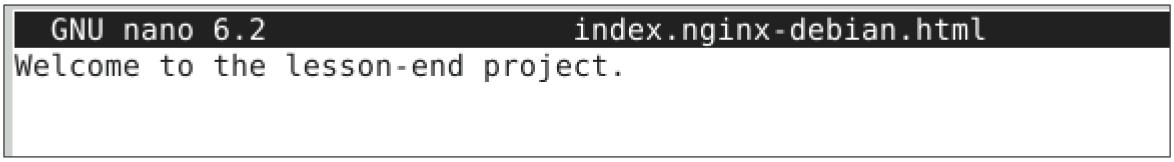
nano index.nginx-debian.html



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ nano index.nginx-debian.html
```

1.5 Add the following content and save the file:

Welcome to the lesson-end project.

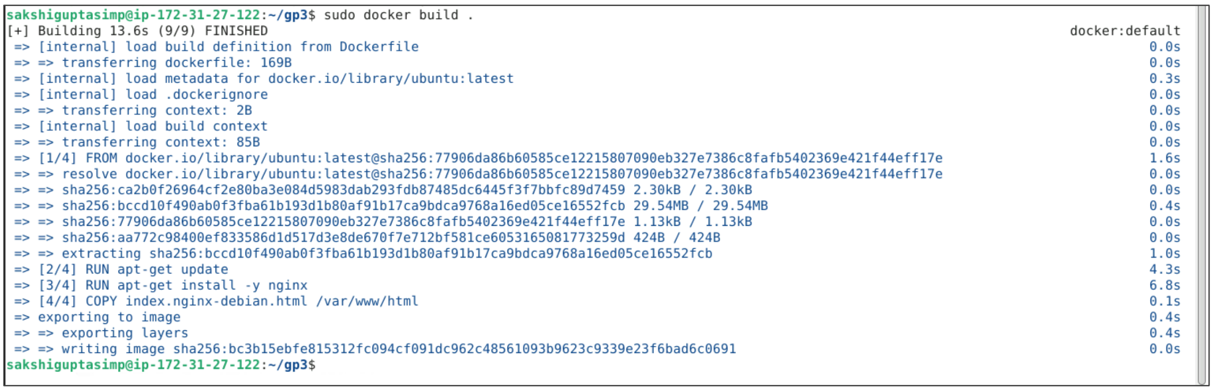


```
  GNU nano 6.2                         index.nginx-debian.html
Welcome to the lesson-end project.
```

Note: To save the file press **Ctrl+X**, then **Y**, and finally **Enter**

1.6 Build the Docker image using the following command:

sudo docker build .

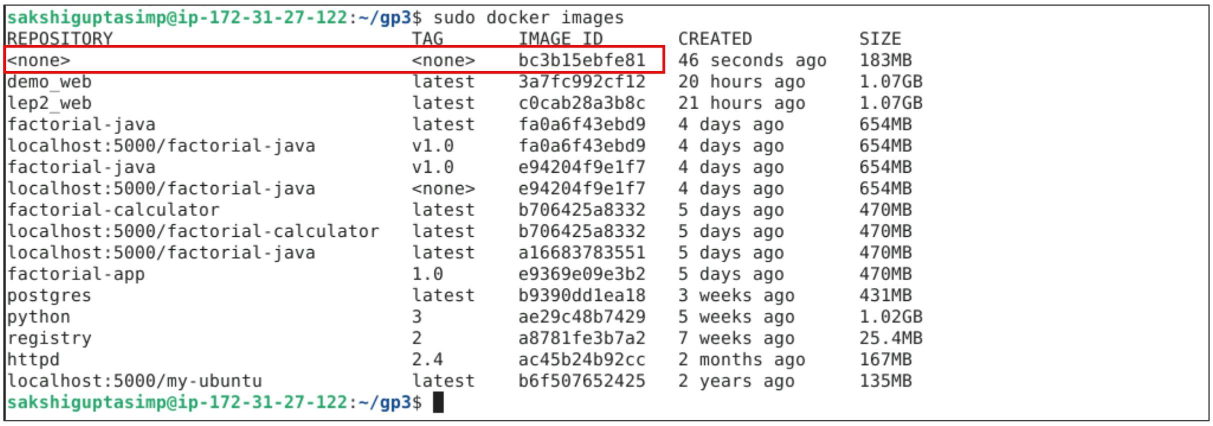


```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker build .
[+] Building 13.6s (9/9) FINISHED                                                                                                     docker:default
 => [internal] load build definition from Dockerfile                                                                                            0.0s
 => => transferring dockerfile: 169B                                                                                                            0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                                                0.3s
 => [internal] load .dockerignore                                                                                                               0.0s
 => => transferring context: 2B                                                                                                                 0.0s
 => [internal] load build context                                                                                                               0.0s
 => => transferring context: 85B                                                                                                                0.0s
 => [1/4] FROM docker.io/library/ubuntu:latest@sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e                          1.6s
 => => resolve docker.io/library/ubuntu:latest@sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e                          0.0s
 => => sha256:ca2b0f26964cf2e80ba3e084d5983dab293fdb87485dc6445f3f7bbfc89d7459 2.30kB / 2.30kB                                                  0.0s
 => => sha256:bccd10f490ab0f3fba61b193d1b80af91b17ca9bdca9768a16ed05ce16552fcb 29.54MB / 29.54MB                                                0.4s
 => => sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e 1.13kB / 1.13kB                                                  0.0s
 => => sha256:aa772c98400ef833586d1d517d3e8de670f7e712bf581ce6053165081773259d 424B / 424B                                                      0.0s
 => => extracting sha256:bccd10f490ab0f3fba61b193d1b80af91b17ca9bdca9768a16ed05ce16552fcb                                                       1.0s
 => [2/4] RUN apt-get update                                                                                                                    4.3s
 => [3/4] RUN apt-get install -y nginx                                                                                                          6.8s
 => [4/4] COPY index.nginx-debian.html /var/www/html                                                                                            0.1s
 => exporting to image                                                                                                                          0.4s
 => => exporting layers                                                                                                                         0.4s
 => => writing image sha256:bc3b15ebfe815312fc094cf091dc962c48561093b9623c9339e23f6bad6c0691                                                    0.0s
sakshiguptasimp@ip-172-31-27-122:~/gp3$
```

Step 2: Tag the image

2.1 Identify the image ID using the following command:

sudo docker images

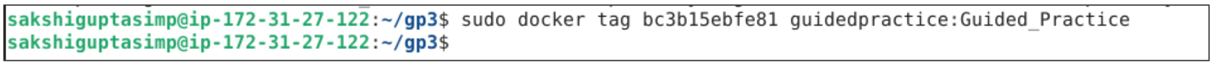


```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker images
REPOSITORY                            TAG       IMAGE ID       CREATED          SIZE
<none>                                <none>    bc3b15ebfe81   46 seconds ago   183MB
demo_web                              latest    3a7fc992cf12   20 hours ago     1.07GB
lep2_web                              latest    c0cab28a3b8c   21 hours ago     1.07GB
factorial-java                        latest    fa0a6f43ebd9   4 days ago       654MB
localhost:5000/factorial-java         v1.0      fa0a6f43ebd9   4 days ago       654MB
factorial-java                        v1.0      e94204f9e1f7   4 days ago       654MB
localhost:5000/factorial-java         <none>    e94204f9e1f7   4 days ago       654MB
factorial-calculator                  latest    b706425a8332   5 days ago       470MB
localhost:5000/factorial-calculator   latest    b706425a8332   5 days ago       470MB
localhost:5000/factorial-java         latest    a16683783551   5 days ago       470MB
factorial-app                         1.0       e9369e09e3b2   5 days ago       470MB
postgres                              latest    b9390dd1ea18   3 weeks ago      431MB
python                                3         ae29c48b7429   5 weeks ago      1.02GB
registry                              2         a8781fe3b7a2   7 weeks ago      25.4MB
httpd                                 2.4       ac45b24b92cc   2 months ago     167MB
localhost:5000/my-ubuntu              latest    b6f507652425   2 years ago      135MB
sakshiguptasimp@ip-172-31-27-122:~/gp3$ █
```

Note: Notice the newly created image with <none> tag. Copy the Image ID of this new image.

2.2 Tag the image with a name using the following command:

sudo docker tag IMAGE_ID guidedpractice:Guided_Practice

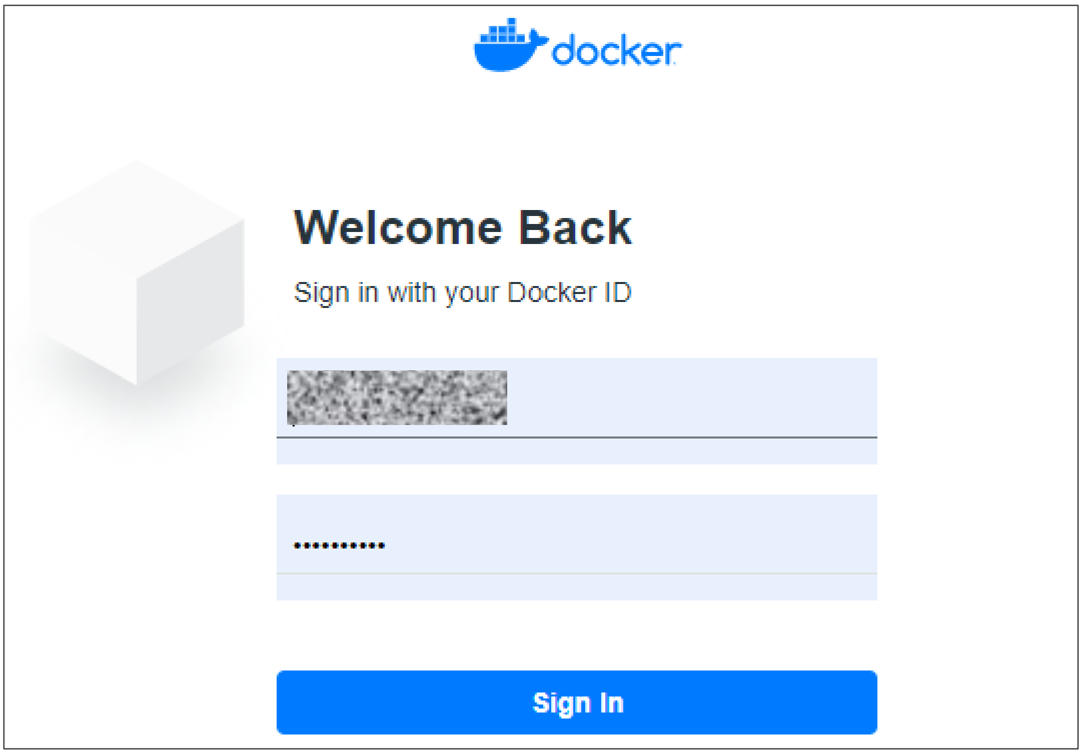


```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker tag bc3b15ebfe81 guidedpractice:Guided_Practice
sakshiguptasimp@ip-172-31-27-122:~/gp3$
```

Note: Replace IMAGE_ID with the ID of the image copied previously

Step 3: Push the image to Docker Hub

3.1 Go to **hub.docker.com** and login to your account



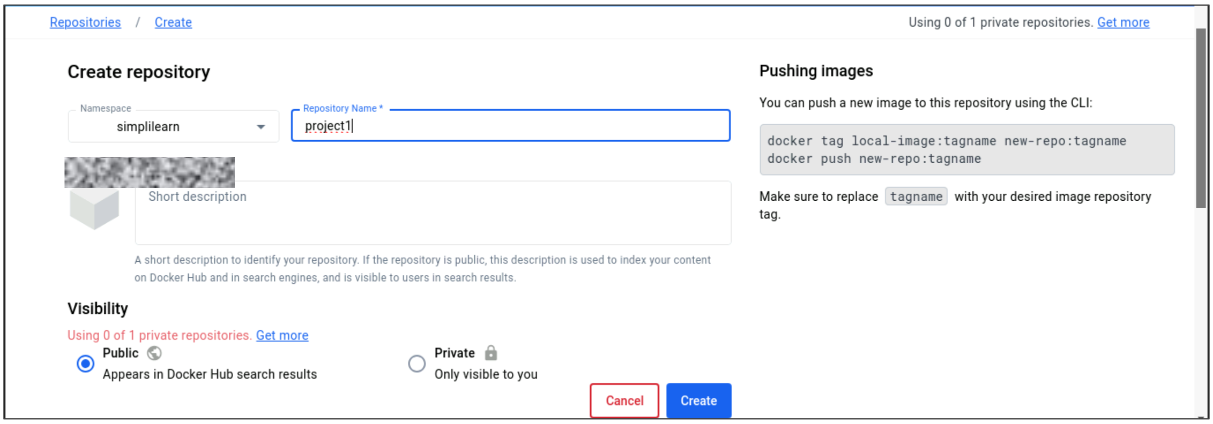
Welcome Back

Sign in with your Docker ID

[redacted]

••••••••••

3.2 Click on the **Create Repository** button to create a new repository called **project1**



Repositories / Create

Using 0 of 1 private repositories. Get more

Create repository

Namespace: simplilearn

Repository Name *: project1

[redacted]

Short description

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. Get more

● Public — Appears in Docker Hub search results

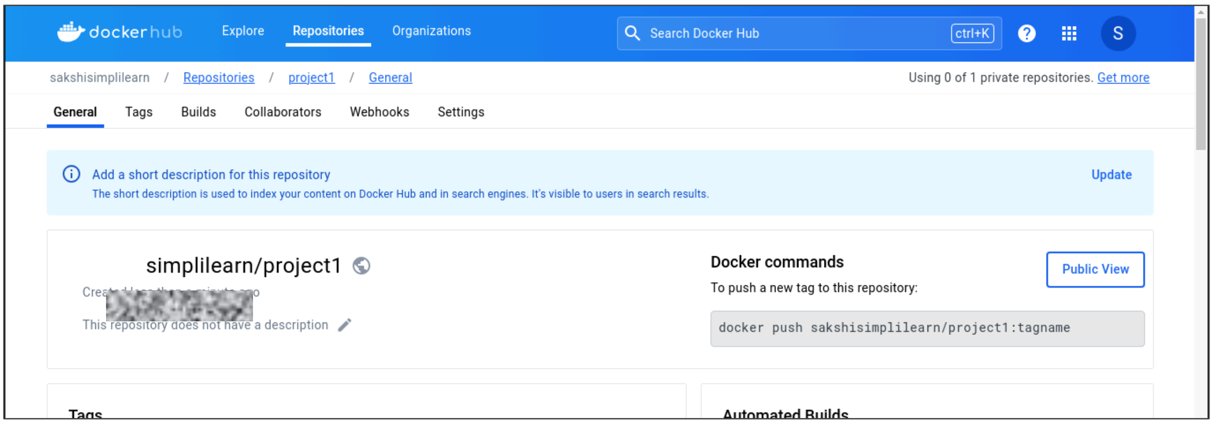
○ Private — Only visible to you

Pushing images

You can push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to replace `tagname` with your desired image repository tag.



Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

simplilearn/project1

[redacted]

This repository does not have a description

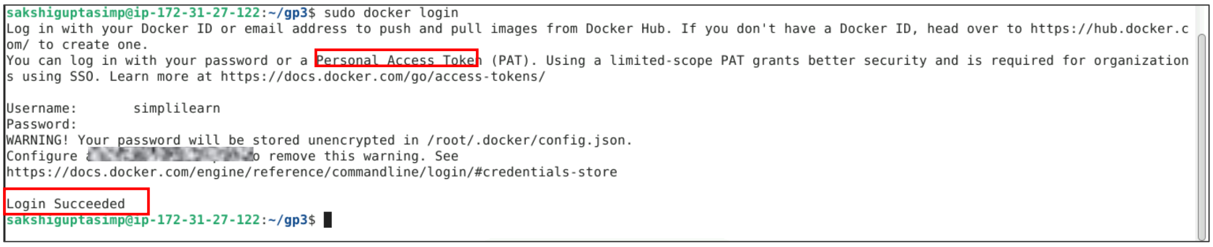
Docker commands

To push a new tag to this repository:

```
docker push sakshisimplilearn/project1:tagname
```

3.3 Navigate back to the terminal and log in to the Docker Hub account using the following command:

sudo docker login

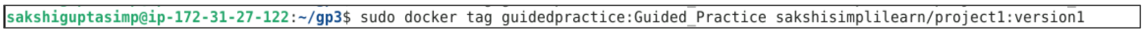


```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.c
om/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organization
s using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username:       simplilearn
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure [redacted]o remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
sakshiguptasimp@ip-172-31-27-122:~/gp3$ 
```

Note: Enter your Docker Hub Username and Password to login

3.4 Retag the image to align with Docker Hub using the following command:

sudo docker tag guidedpractice:Guided_Practice DOCKER_HUB_USERNAME/project1:version1

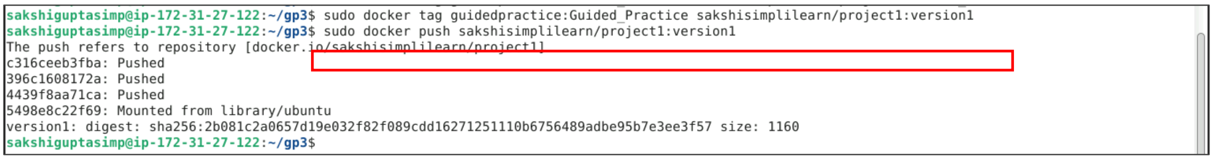


```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker tag guidedpractice:Guided_Practice sakshisimplilearn/project1:version1
```

Note: Replace DOCKER_HUB_USERNAME with the Username of your Docker Hub account

3.5 Push the retagged image to Docker Hub using the following command:

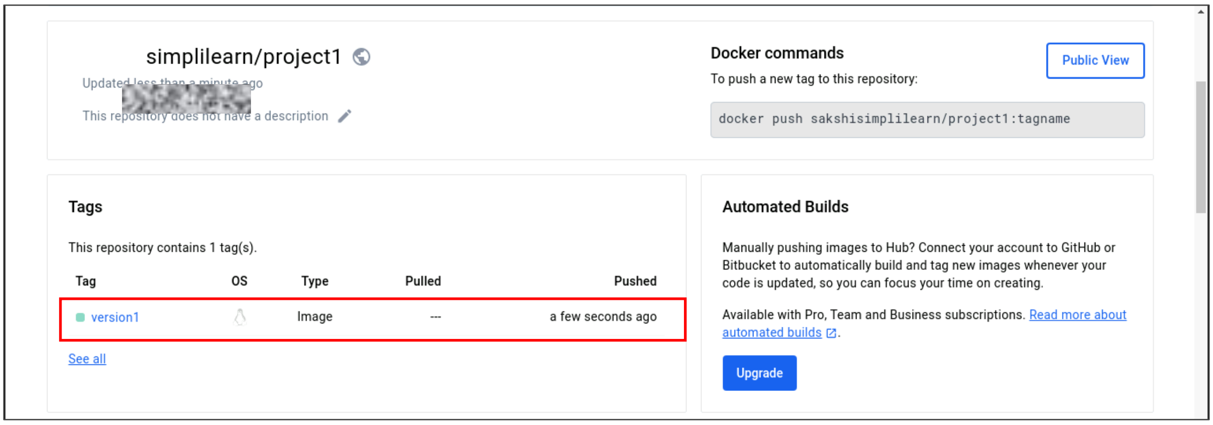
sudo docker push DOCKER_HUB_USERNAME/project1:version1



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker tag guidedpractice:Guided_Practice sakshisimplilearn/project1:version1
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker push sakshisimplilearn/project1:version1
The push refers to repository [docker.io/sakshisimplilearn/project1]
c316ceeb3fba: Pushed
396c1608172a: Pushed
4439f8aa71ca: Pushed
5498e8c22f69: Mounted from library/ubuntu
version1: digest: sha256:2b081c2a0657d19e032f82f089cdd16271251110b6756489adbe95b7e3ee3f57 size: 1160
sakshiguptasimp@ip-172-31-27-122:~/gp3$
```

Note: Replace DOCKER_HUB_USERNAME with the username of your Docker Hub account

3.6 Go to your Docker Hub account and navigate to Repositories to see the recently pushed image



simplilearn/project1

Updated less than a minute ago [redacted]

This repository does not have a description

Docker commands

To push a new tag to this repository:

```
docker push sakshisimplilearn/project1:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
version1		Image	---	a few seconds ago

Automated Builds

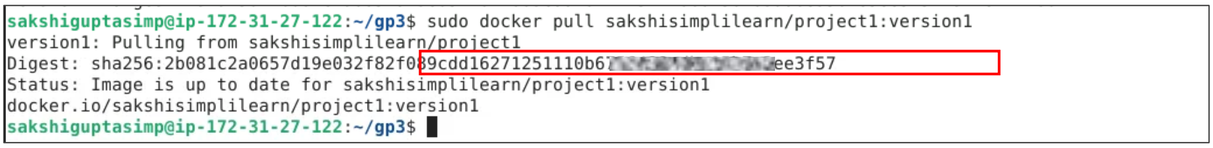
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

Step 4: Create a container

4.1 Pull the image from Docker Hub using the following command:

sudo docker pull DOCKER_HUB_USERNAME/project1:version1

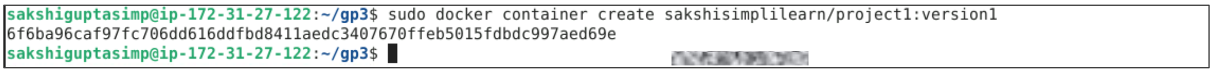


```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker pull sakshisimplilearn/project1:version1
version1: Pulling from sakshisimplilearn/project1
Digest: sha256:2b081c2a0657d19e032f82f089cdd16271251110b6[redacted]ee3f57
Status: Image is up to date for sakshisimplilearn/project1:version1
docker.io/sakshisimplilearn/project1:version1
sakshiguptasimp@ip-172-31-27-122:~/gp3$
```

Note: Replace DOCKER_HUB_USERNAME with the username of your Docker Hub account

4.2 Create a container using the following command:

sudo docker container create DOCKER_HUB_USERNAME/project1:version1



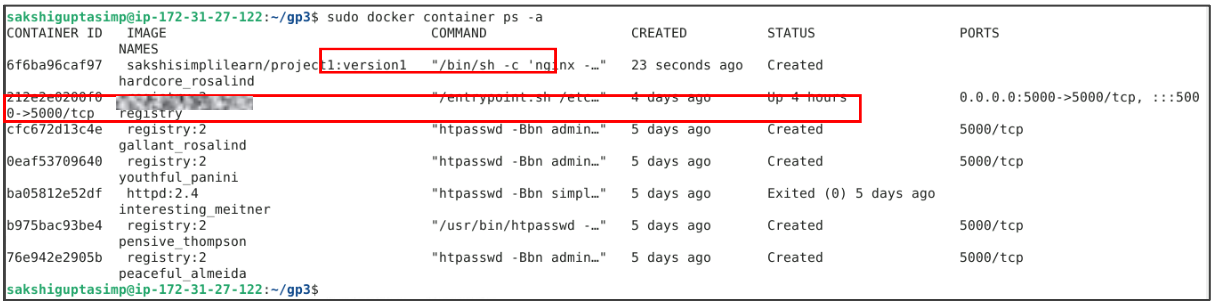
```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker container create sakshisimplilearn/project1:version1
6f6ba96caf97fc706dd616ddfbd8411aedc3407670ffeb5015fdbdc997aed69e
sakshiguptasimp@ip-172-31-27-122:~/gp3$            [redacted]
```

Note: Replace DOCKER_HUB_USERNAME with the username of your Docker Hub account

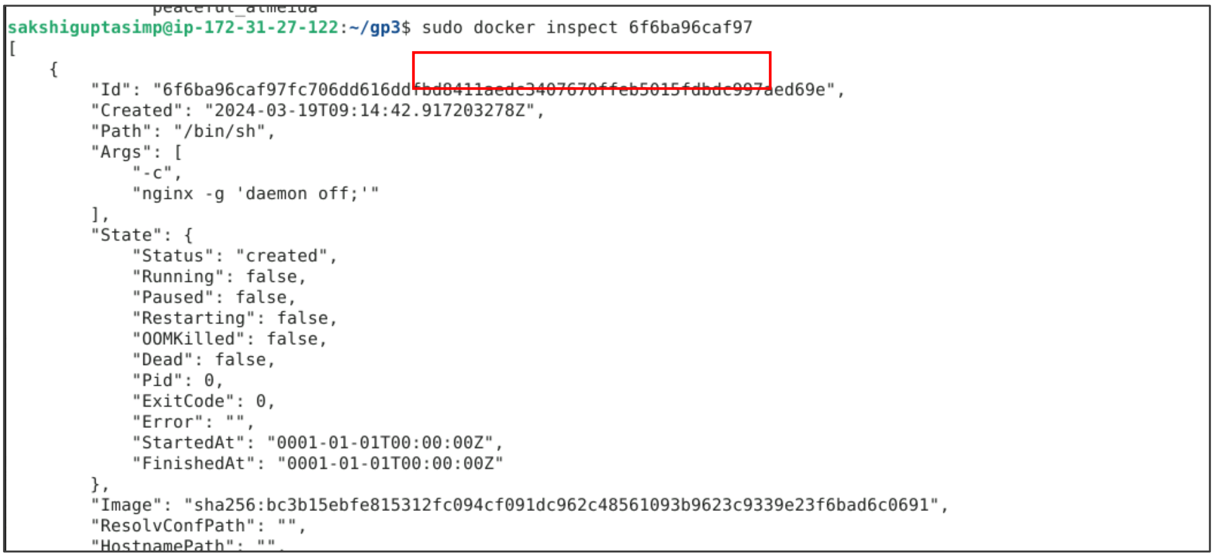
4.3 Check the newly created container and inspect it using the following commands:

sudo docker container ps -a

sudo docker inspect CONTAINER_ID



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker container ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS                   PORTS
               NAMES
6f6ba96caf97   sakshisimplilearn/project1:version1   "/bin/sh -c 'nginx -…"   23 seconds ago   Created
               hardcore_rosalind
212e2e0200f0   [redacted]                           "/entrypoint.sh /etc…"   4 days ago       Up 4 hours               0.0.0.0:5000->5000/tcp, :::500
0->5000/tcp    registry
cfc672d13c4e   registry:2                           "htpasswd -Bbn admin…"   5 days ago       Created                  5000/tcp
               gallant_rosalind
0eaf53709640   registry:2                           "htpasswd -Bbn admin…"   5 days ago       Created                  5000/tcp
               youthful_panini
ba05812e52df   httpd:2.4                            "htpasswd -Bbn simpl…"   5 days ago       Exited (0) 5 days ago
               interesting_meitner
b975bac93be4   registry:2                           "/usr/bin/htpasswd -…"   5 days ago       Created                  5000/tcp
               pensive_thompson
76e942e2905b   registry:2                           "htpasswd -Bbn admin…"   5 days ago       Created                  5000/tcp
               peaceful_almeida
sakshiguptasimp@ip-172-31-27-122:~/gp3$
```



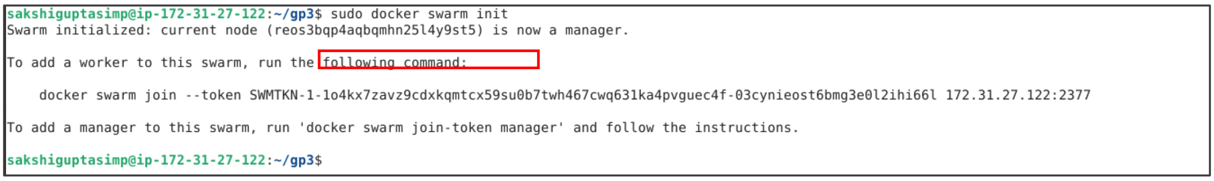
```
               peaceful_almeida
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker inspect 6f6ba96caf97
[
    {
        "Id": "6f6ba96caf97fc706dd616ddfbd8411aedc3407670ffeb5015fdbdc997aed69e",
        "Created": "2024-03-19T09:14:42.917203278Z",
        "Path": "/bin/sh",
        "Args": [
            "-c",
            "nginx -g 'daemon off;'"
        ],
        "State": {
            "Status": "created",
            "Running": false,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 0,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "0001-01-01T00:00:00Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:bc3b15ebfe815312fc094cf091dc962c48561093b9623c9339e23f6bad6c0691",
        "ResolvConfPath": "",
        "HostnamePath": "",
```

Note: Replace CONTAINER_ID with the ID of the newly created container

Step 5: Check nodes in Swarm where the manager node is running

5.1 Initialize Docker Swarm on the current node using the following command:

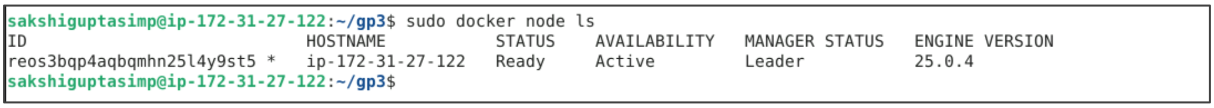
sudo docker swarm init



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker swarm init
Swarm initialized: current node (reos3bqp4aqbqmhn25l4y9st5) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-1o4kx7zavz9cdxkqmtcx59su0b7twh467cwq631ka4pvguec4f-03cynieost6bmg3e0l2ihi66l 172.31.27.122:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

sakshiguptasimp@ip-172-31-27-122:~/gp3$
```

5.2 Run the following command to check nodes in the Swarm using the following command:

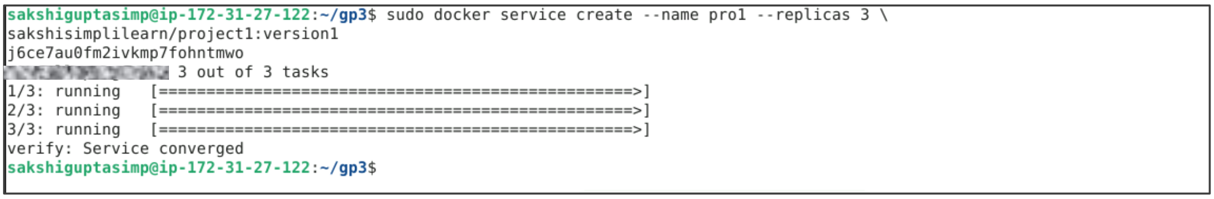
sudo docker node ls



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker node ls
ID                            HOSTNAME           STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
reos3bqp4aqbqmhn25l4y9st5 *   ip-172-31-27-122   Ready     Active         Leader           25.0.4
sakshiguptasimp@ip-172-31-27-122:~/gp3$
```

Step 6: Create a service and check the default container

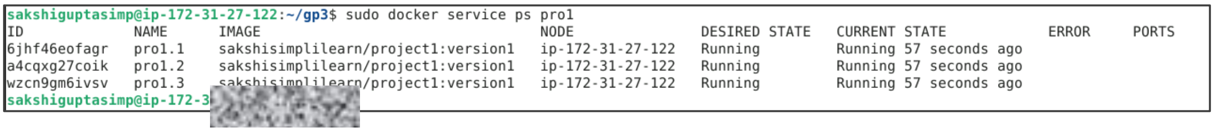
6.1 Create a service named pro1 with 3 replicas to run them on three nodes using the following command:
sudo docker service create --name pro1 --replicas 3 DOCKER_HUB_USERNAME/project1:version1



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker service create --name pro1 --replicas 3 \
sakshisimplilearn/project1:version1
j6ce7au0fm2ivkmp7fohntmwo
[redacted] 3 out of 3 tasks
1/3: running   [==================================================>]
2/3: running   [==================================================>]
3/3: running   [==================================================>]
verify: Service converged
sakshiguptasimp@ip-172-31-27-122:~/gp3$
```

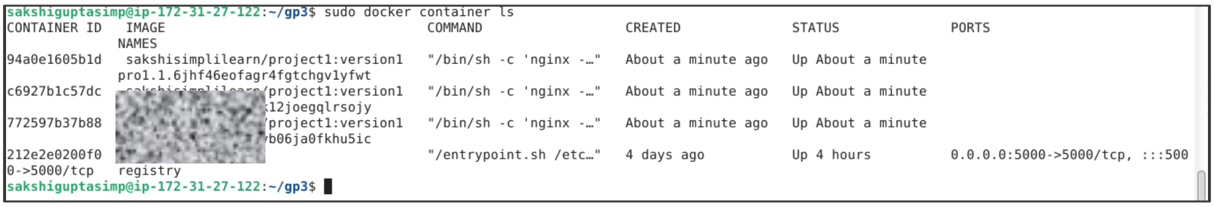
Note: Replace DOCKER_HUB_USERNAME with the username of your Docker Hub account

6.2 Check the nodes and task details of service pro1 using the following command:
sudo docker service ps pro1



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker service ps pro1
ID             NAME       IMAGE                                 NODE               DESIRED STATE   CURRENT STATE            ERROR     PORTS
6jhf46eofagr   pro1.1     sakshisimplilearn/project1:version1   ip-172-31-27-122   Running         Running 57 seconds ago
a4cqxg27coik   pro1.2     sakshisimplilearn/project1:version1   ip-172-31-27-122   Running         Running 57 seconds ago
wzcn9gm6ivsv   pro1.3     [redacted]/project1:version1   ip-172-31-27-122   Running         Running 57 seconds ago
sakshiguptasimp@ip-172-3[redacted]
```

6.3 Check the containers that are created by default using the following command:
sudo docker container ls



```
sakshiguptasimp@ip-172-31-27-122:~/gp3$ sudo docker container ls
CONTAINER ID   IMAGE                                 COMMAND                  CREATED              STATUS              PORTS
              NAMES
94a0e1605b1d    sakshisimplilearn/project1:version1   "/bin/sh -c 'nginx -…"   About a minute ago   Up About a minute
              pro1.1.6jhf46eofagr4fgtchgv1yfwt
c6927b1c57dc   [redacted]/project1:version1   "/bin/sh -c 'nginx -…"   About a minute ago   Up About a minute
              [redacted]12joegqlrsojy
772597b37b88   [redacted]/project1:version1   "/bin/sh -c 'nginx -…"   About a minute ago   Up About a minute
              [redacted]b06ja0fkhu5ic
212e2e0200f0   [redacted]                       "/entrypoint.sh /etc…"   4 days ago           Up 4 hours          0.0.0.0:5000->5000/tcp, :::500
0->5000/tcp   registry
sakshiguptasimp@ip-172-31-27-122:~/gp3$ █
```

By following these steps, you have successfully created a Docker image, pushed it to Docker Hub, and deployed a replicated service on a Swarm cluster to ensure high availability and simplify application update management.