Assignment – 4

Study best practices of docker push and docker pull.

Docker push and pull are essential commands used in the Docker platform to upload and download Docker images from a registry. Here are some best practices for using these commands:

1. **Use secure authentication:** Always authenticate your Docker client to the registry using secure authentication methods such as OAuth or token-based authentication. This ensures that your Docker images are secure and are only accessible by authorized users.
2. **Use tagged images:** Always tag your Docker images with a unique identifier, such as a version number or a commit ID. This helps to ensure that you are using the correct image, and it also makes it easier to manage and update your images.
3. **Use a private registry:** If you're working with sensitive data, it's best to use a private registry to store your Docker images. Private registries are more secure than public registries because you can control who has access to your images.
4. **Use caching:** Docker has a built-in caching mechanism that can speed up the process of pushing and pulling Docker images. By enabling caching, Docker will check if an image has already been downloaded or uploaded and will use the cached image if it's available.
5. **Optimize image size:** It's best to optimize your Docker images to reduce their size. This can help to speed up the push and pull process, as well as reduce storage costs.
6. Use version control: Always use version control to manage your Docker images. This makes it easier to track changes and roll back to previous versions if necessary.
7. **Monitor and update images:** It's important to monitor your Docker images for vulnerabilities and updates. Make sure to regularly update your images and replace any outdated images with newer versions. This will help to ensure that your Docker environment is secure and up to date.

1] How to Push a Image to a Docker Repository?

Step 1: Login to your docker account. Provide required username and password to login.

```
docker login
```

Step 2: Select an image that we want to push for that use following command to check all images.

```
docker images
```

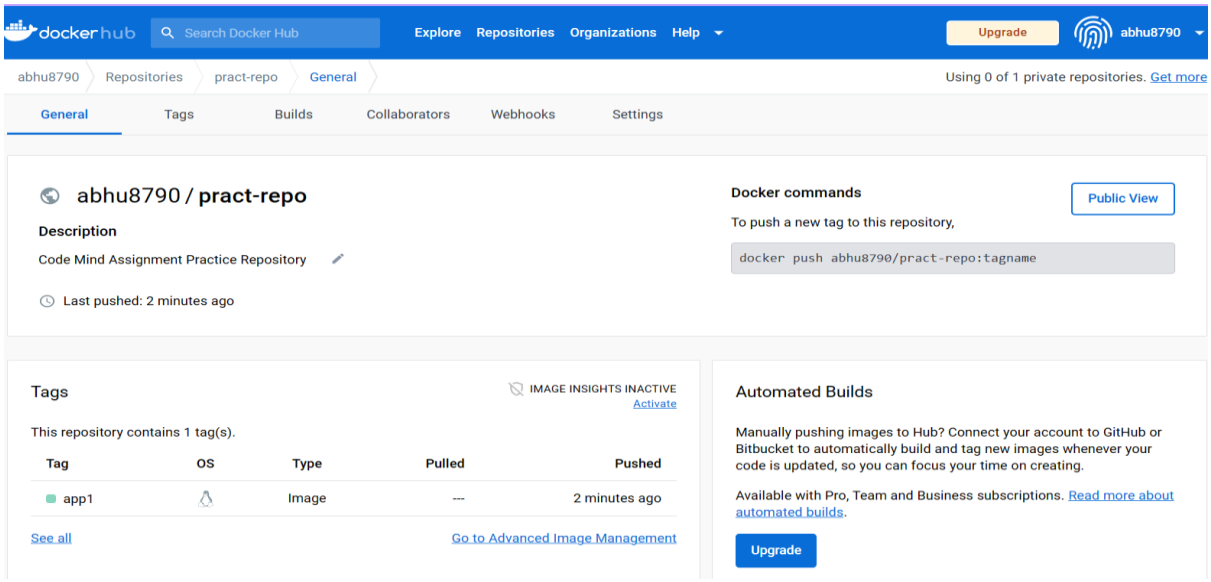
Step 3: Change the tag name if required using following command:

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

Step 4: Now Push the image using following command.

```
docker push new-repo:tagname
```

```
bhushan@ubuntu:~$ docker images
REPOSITORY                          TAG       IMAGE ID       CREATED       SIZE
demo                                latest    7943ea59c5fb   3 hours ago   7.05MB
pract-repo/demoimage1               l1        7943ea59c5fb   3 hours ago   7.05MB
abhu8790/pract-repo/demo            latest    7943ea59c5fb   3 hours ago   7.05MB
codemind_docker_assignment_image    v1        7943ea59c5fb   3 hours ago   7.05MB
alpine                              latest    b2aa39c304c2   6 weeks ago   7.05MB
bhushan@ubuntu:~$ docker tag codemind_docker_assignment_image:v1 abhu8790/pract-repo:app1
bhushan@ubuntu:~$ docker images
REPOSITORY                          TAG       IMAGE ID       CREATED       SIZE
abhu8790/pract-repo                 app1      7943ea59c5fb   3 hours ago   7.05MB
codemind_docker_assignment_image    v1        7943ea59c5fb   3 hours ago   7.05MB
demo                                latest    7943ea59c5fb   3 hours ago   7.05MB
pract-repo/demoimage1               l1        7943ea59c5fb   3 hours ago   7.05MB
abhu8790/pract-repo/demo            latest    7943ea59c5fb   3 hours ago   7.05MB
alpine                              latest    b2aa39c304c2   6 weeks ago   7.05MB
bhushan@ubuntu:~$ docker push abhu8790/pract-repo:app1
The push refers to repository [docker.io/abhu8790/pract-repo]
7cd52847ad77: Pushed
app1: digest: sha256:34adcf6e16295750edc9c36794f2cc3a37992c47ad78a7725378cd1c6b9e856f size: 528
bhushan@ubuntu:~$
```



2] How to pull the docker images from docker hub?

To pull Docker Images with specific tags or versions, you can use the following command.

sudo docker pull <image-name>:<tag-name>

```
bhushan@ubuntu:~$ docker pull alpine:3.17.2
3.17.2: Pulling from library/alpine
Digest: sha256:ff6bdca1701f3a8a67e328815ff2346b0e4067d32ec36b7992c1fdc001dc8517
Status: Downloaded newer image for alpine:3.17.2
docker.io/library/alpine:3.17.2
bhushan@ubuntu:~$ docker images
REPOSITORY                          TAG       IMAGE ID       CREATED       SIZE
abhu8790/pract-repo                 app1      7943ea59c5fb   3 hours ago   7.05MB
codemind_docker_assignment_image    v1        7943ea59c5fb   3 hours ago   7.05MB
demo                                latest    7943ea59c5fb   3 hours ago   7.05MB
pract-repo/demoimage1               l1        7943ea59c5fb   3 hours ago   7.05MB
abhu8790/pract-repo/demo            latest    7943ea59c5fb   3 hours ago   7.05MB
alpine                              3.17.2    b2aa39c304c2   6 weeks ago   7.05MB
alpine                              latest    b2aa39c304c2   6 weeks ago   7.05MB
bhushan@ubuntu:~$ docker run -it --name myapp alpine:3.17.2
/ # ls
bin   dev   etc   home  lib   media mnt   opt   proc  root  run   sbin  srv   sys   tmp   usr   var
/ # exit
bhushan@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS                      PORTS     NAMES
d54770d08d90   alpine:3.17.2                        "/bin/sh"                18 seconds ago   Exited (0) 5 seconds ago              myapp
2ecd0dfb9174   codemind_docker_assignment_image:v1  "/bin/sh -c 'date '+…"   3 hours ago      Exited (0) 3 hours ago                cadc
b0b1073d819c   codemind_docker_assignment_image:v1  "/bin/sh -c 'date '+…"   3 hours ago      Exited (0) 3 hours ago                codemind_docker_assignment_container
bhushan@ubuntu:~$
```