How can we transfer the docker images?

- Push the images to a Repository & Pull it on required machines (using Docker Push & Pull Commands)
  - ✓ Private Repo (Docker Trusted Registry DTR)
  - ✓ Public Repo (Docker Hub)

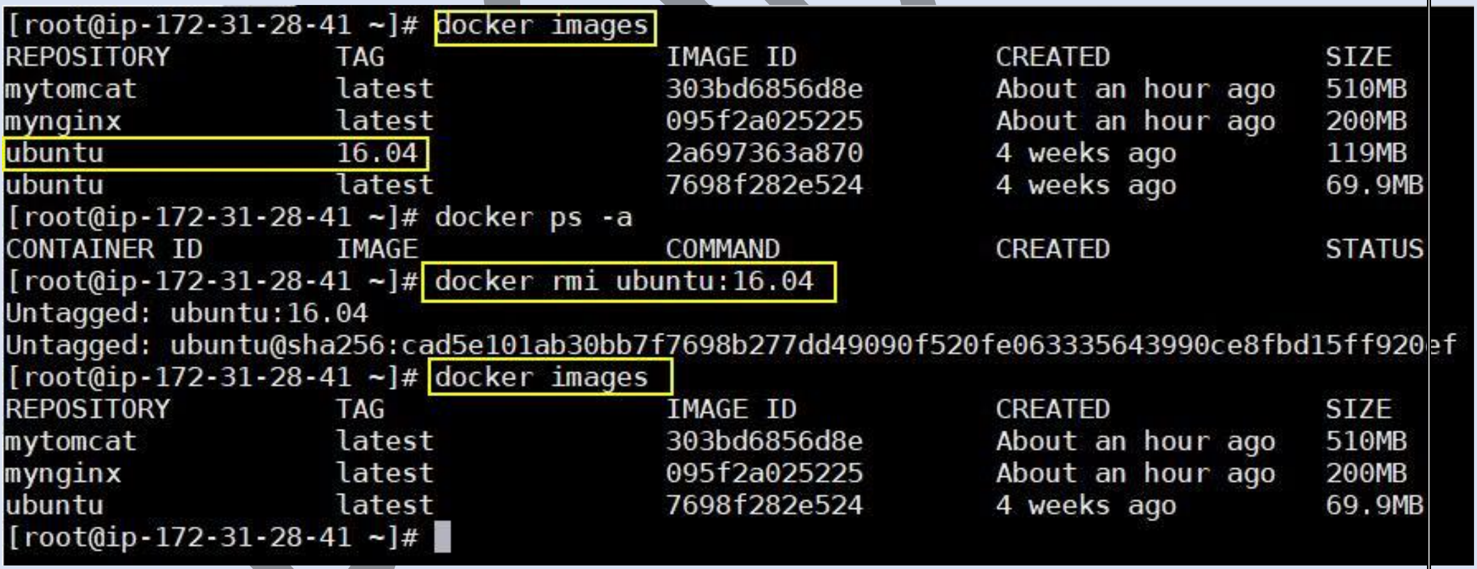
or

Save it as tar file & SCP to the required machines (not suggested)

To push the image to docker hub first we need to run “docker login” command from our terminal/client machine then we need to enter the username and password of the docker hub account (if you don’t have docker hub account then first signup for docker hub account).

Note: - We need to tag our image with our repo name to push them to our repo and the repo name we can get the dockerhub account with what id we have created the account.

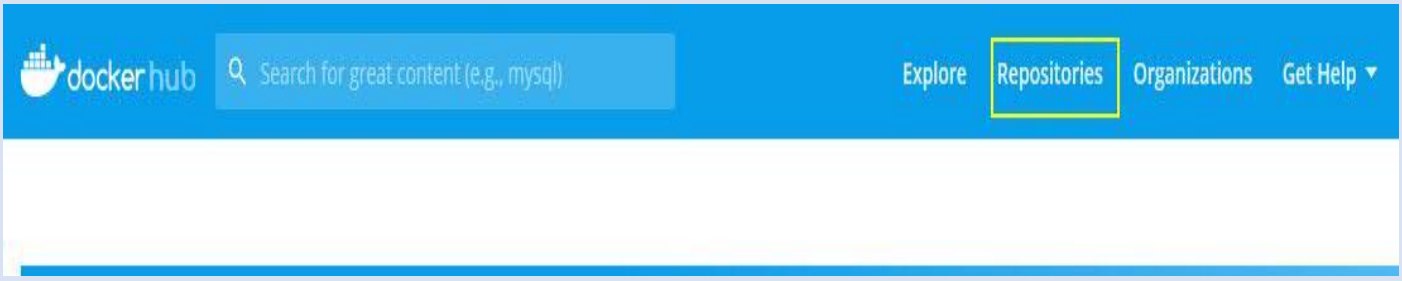
Importance of tag: see in below snap we do have two ubuntu image, one is latest version and other is 16.04, I had run the command docker rmi ubuntu:16.04, if I wouldn’t have given 16.04 version here then by default docker takes the latest tag id and ubuntu with latest image should have deleted.



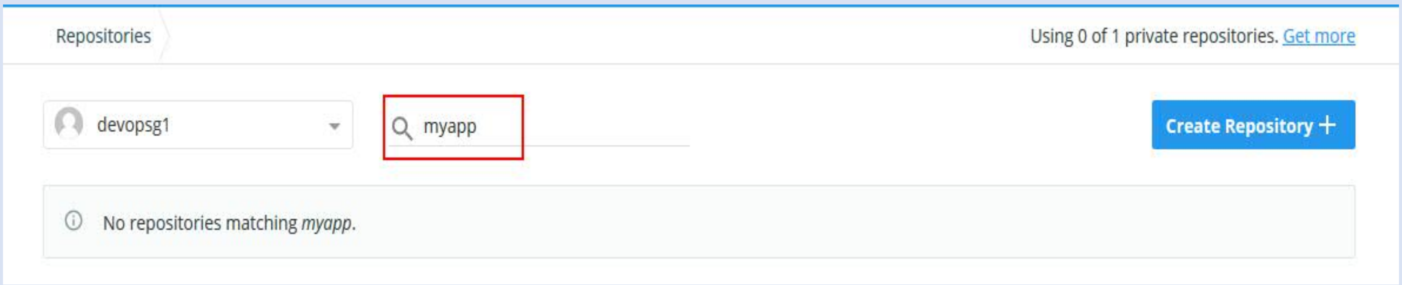
```
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mytomcat            latest              303bd6856d8e        About an hour ago   510MB
mynginx             latest              095f2a025225        About an hour ago   200MB
ubuntu              16.04               2a697363a870        4 weeks ago         119MB
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
[root@ip-172-31-28-41 ~]# docker rmi ubuntu:16.04
Untagged: ubuntu:16.04
Untagged: ubuntu@sha256:cad5e101ab30bb7f7698b277dd49090f520fe063335643990ce8fbd15ff920ef
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mytomcat            latest              303bd6856d8e        About an hour ago   510MB
mynginx             latest              095f2a025225        About an hour ago   200MB
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]#
```

First we need to go to docker hub and create a public repository with any name: -

Step1:-



Step 2:-



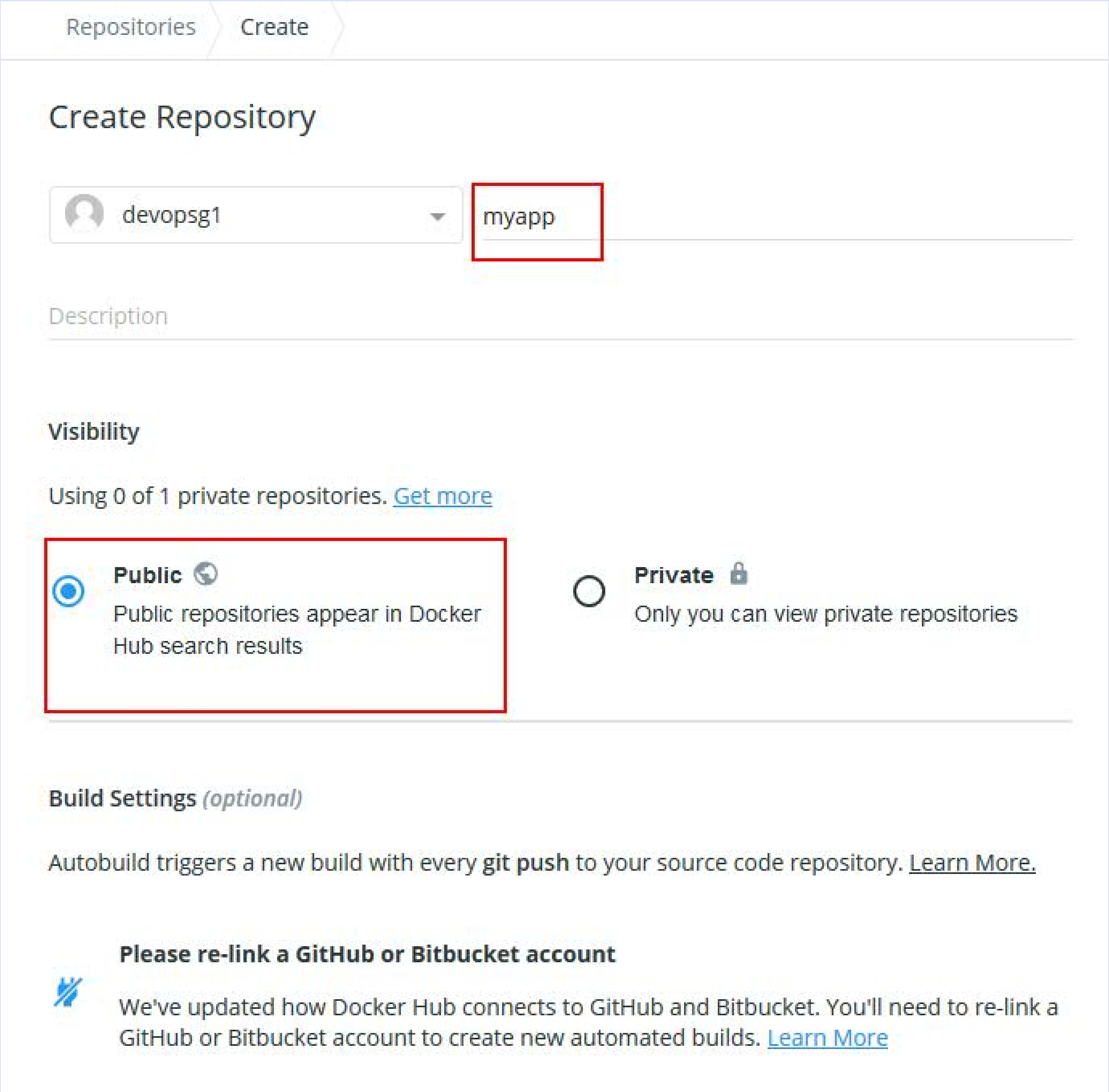
Using 0 of 1 private repositories. Get more

devopsg1

myapp

No repositories matching *myapp*.

Step 3:-



Create Repository

devopsg1

myapp

Description

Visibility

Using 0 of 1 private repositories. Get more

● **Public**
Public repositories appear in Docker Hub search results

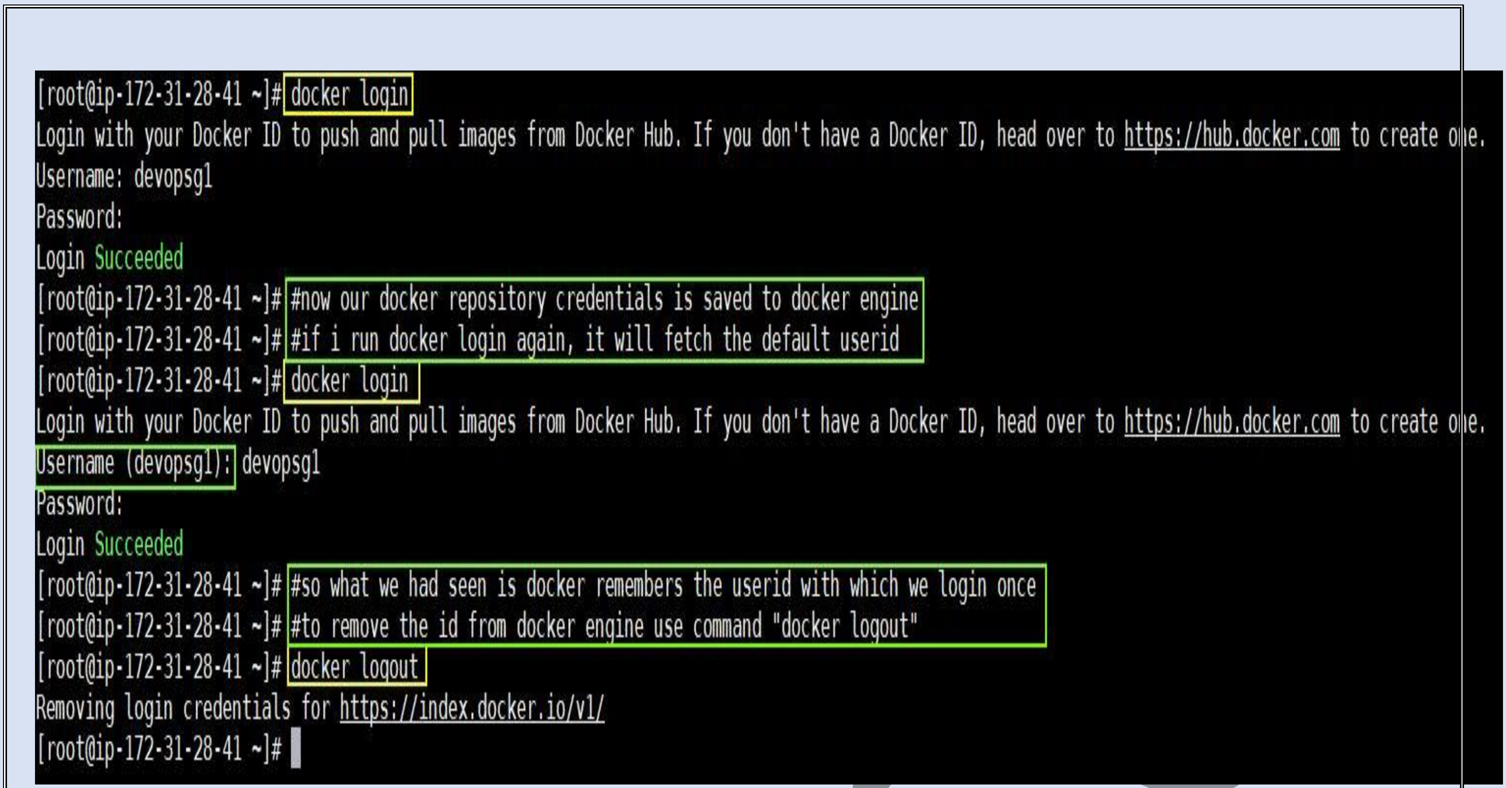
○ **Private**
Only you can view private repositories

Build Settings *(optional)*

Autobuild triggers a new build with every **git push** to your source code repository. Learn More.

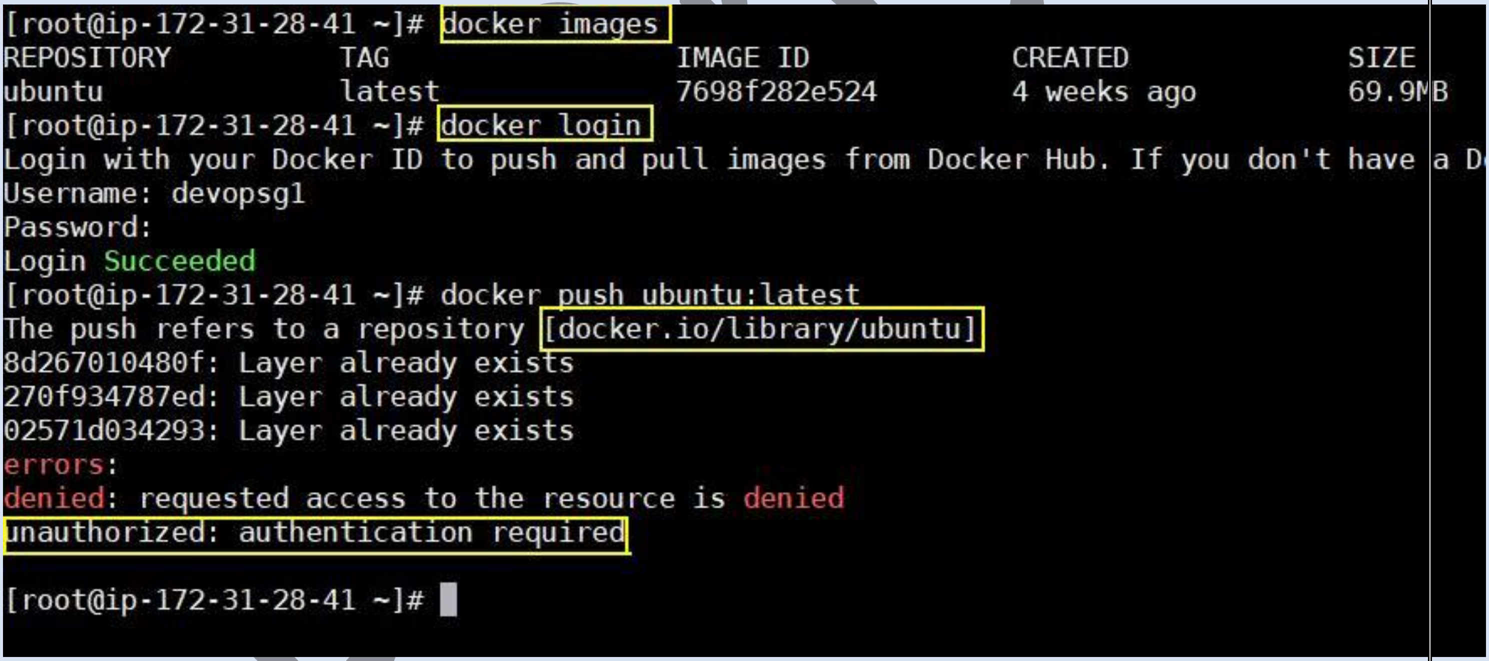
Please re-link a GitHub or Bitbucket account

We've updated how Docker Hub connects to GitHub and Bitbucket. You'll need to re-link a GitHub or Bitbucket account to create new automated builds. Learn More



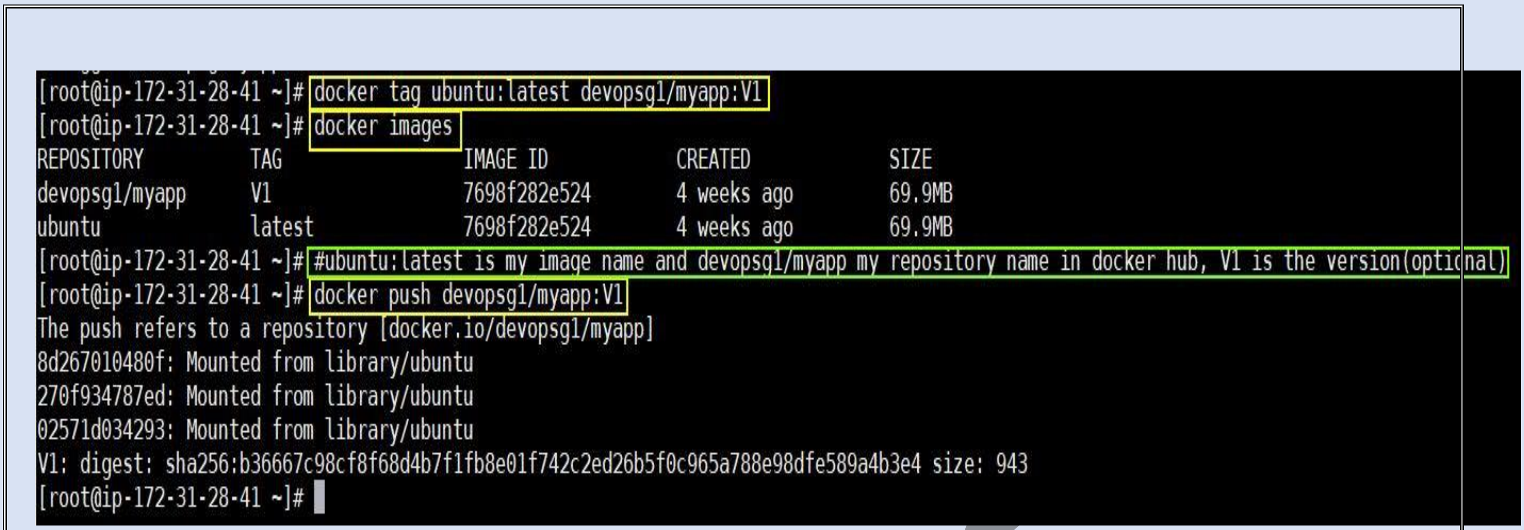
```
[root@ip-172-31-28-41 ~]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: devopsg1
Password:
Login Succeeded
[root@ip-172-31-28-41 ~]# #now our docker repository credentials is saved to docker engine
[root@ip-172-31-28-41 ~]# #if i run docker login again, it will fetch the default userid
[root@ip-172-31-28-41 ~]# docker login
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 (devopsg1): devopsg1
Password:
Login Succeeded
[root@ip-172-31-28-41 ~]# #so what we had seen is docker remembers the userid with which we login once
[root@ip-172-31-28-41 ~]# #to remove the id from docker engine use command "docker logout"
[root@ip-172-31-28-41 ~]# docker logout
Removing login credentials for https://index.docker.io/v1/
[root@ip-172-31-28-41 ~]#
```

I can't push any image directly to docker hub without tagging with my repository name because default repository is always docker.io and it will give us error when we will try to push any image directly: -



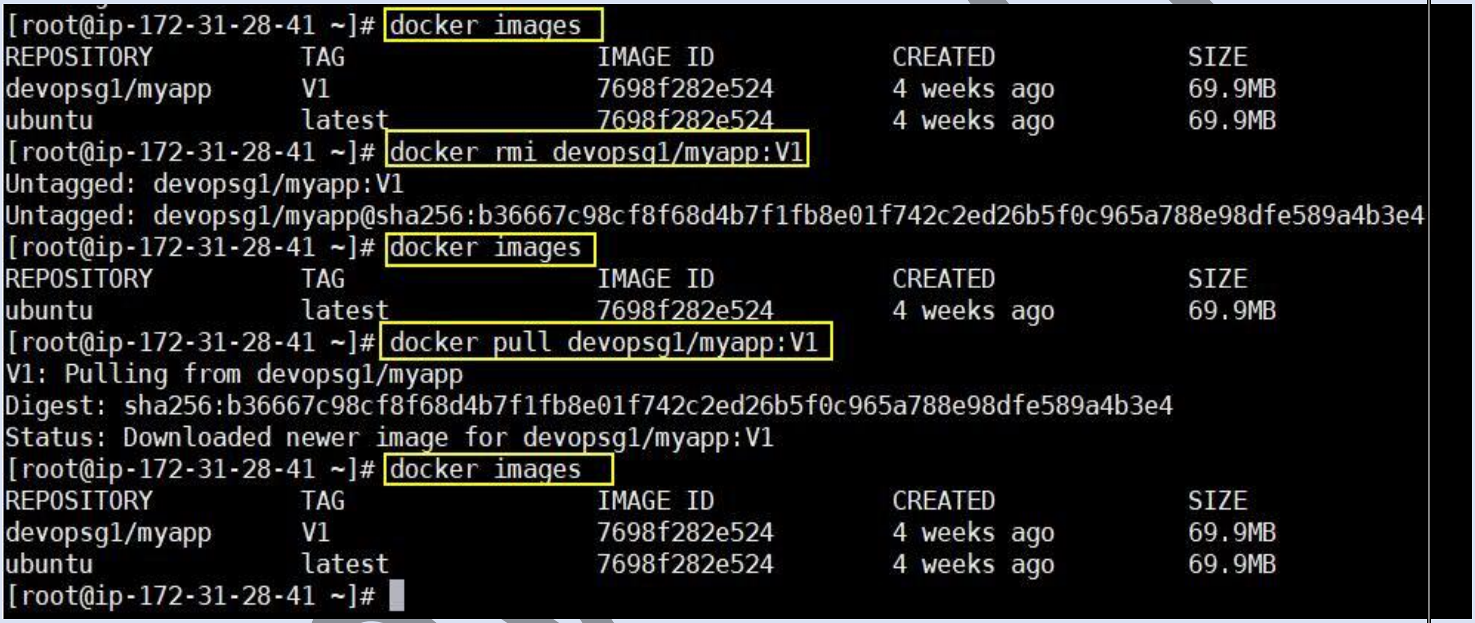
```
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a D
Username: devopsg1
Password:
Login Succeeded
[root@ip-172-31-28-41 ~]# docker push ubuntu:latest
The push refers to a repository [docker.io/library/ubuntu]
8d267010480f: Layer already exists
270f934787ed: Layer already exists
02571d034293: Layer already exists
errors:
denied: requested access to the resource is denied
unauthorized: authentication required

[root@ip-172-31-28-41 ~]#
```

So, in below snap we have pushed the image to our docker hub account: -



```
[root@ip-172-31-28-41 ~]# docker tag ubuntu:latest devopsg1/myapp:V1
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
devopsg1/myapp      V1                  7698f282e524        4 weeks ago         69.9MB
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]# #ubuntu:latest is my image name and devopsg1/myapp my repository name in docker hub, V1 is the version(optional)
[root@ip-172-31-28-41 ~]# docker push devopsg1/myapp:V1
The push refers to a repository [docker.io/devopsg1/myapp]
8d267010480f: Mounted from library/ubuntu
270f934787ed: Mounted from library/ubuntu
02571d034293: Mounted from library/ubuntu
V1: digest: sha256:b36667c98cf8f68d4b7f1fb8e01f742c2ed26b5f0c965a788e98dfe589a4b3e4 size: 943
[root@ip-172-31-28-41 ~]#
```

In below snap I had pushed the image to docker hub, and then removed that image from my docker engine, after that again I pulled the same image from my docker hub to my docker engine: -



```
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
devopsg1/myapp      V1                  7698f282e524        4 weeks ago         69.9MB
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]# docker rmi devopsg1/myapp:V1
Untagged: devopsg1/myapp:V1
Untagged: devopsg1/myapp@sha256:b36667c98cf8f68d4b7f1fb8e01f742c2ed26b5f0c965a788e98dfe589a4b3e4
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]# docker pull devopsg1/myapp:V1
V1: Pulling from devopsg1/myapp
Digest: sha256:b36667c98cf8f68d4b7f1fb8e01f742c2ed26b5f0c965a788e98dfe589a4b3e4
Status: Downloaded newer image for devopsg1/myapp:V1
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
devopsg1/myapp      V1                  7698f282e524        4 weeks ago         69.9MB
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]#
```

DTR (Docker Trusted Registry):-

DTR is a Private repository just like docker-hub but it lives in your Data Center, so that images in DTR can be accessed by only your company.

We need to setup the DTR in your local to push pull the images, once we setup DTR we will have similar UI to login and create your own repo as we did in Docker Hub

We can use the same commands which we used for docker hub, the difference is you need to add your repo addr to login & push

example:

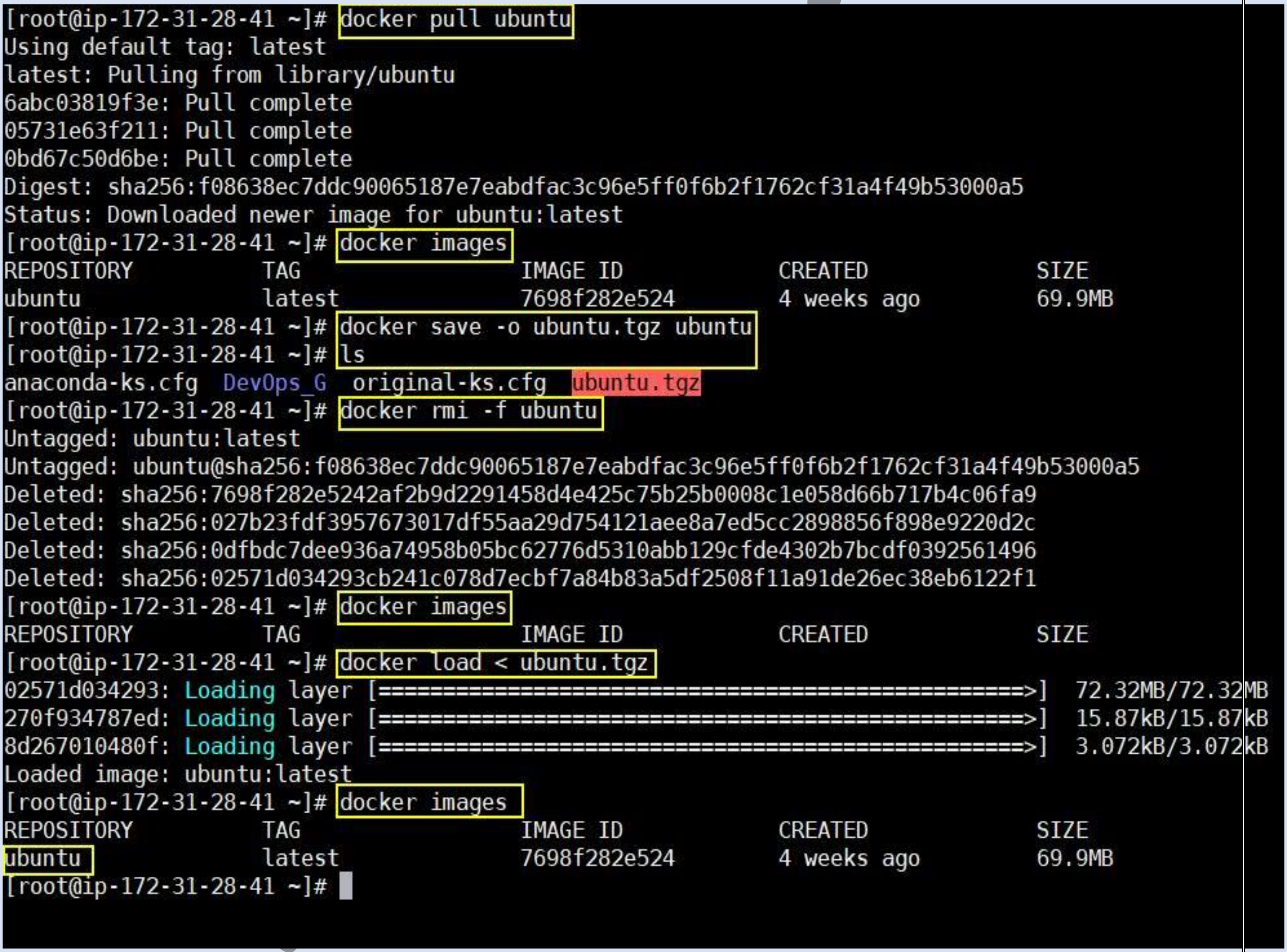
docker login **<repo addr>**

docker tag <image id/name> **repoaddr**/loginid/reponame:tag

docker push **repoaddr**/loginid/reponame:tag

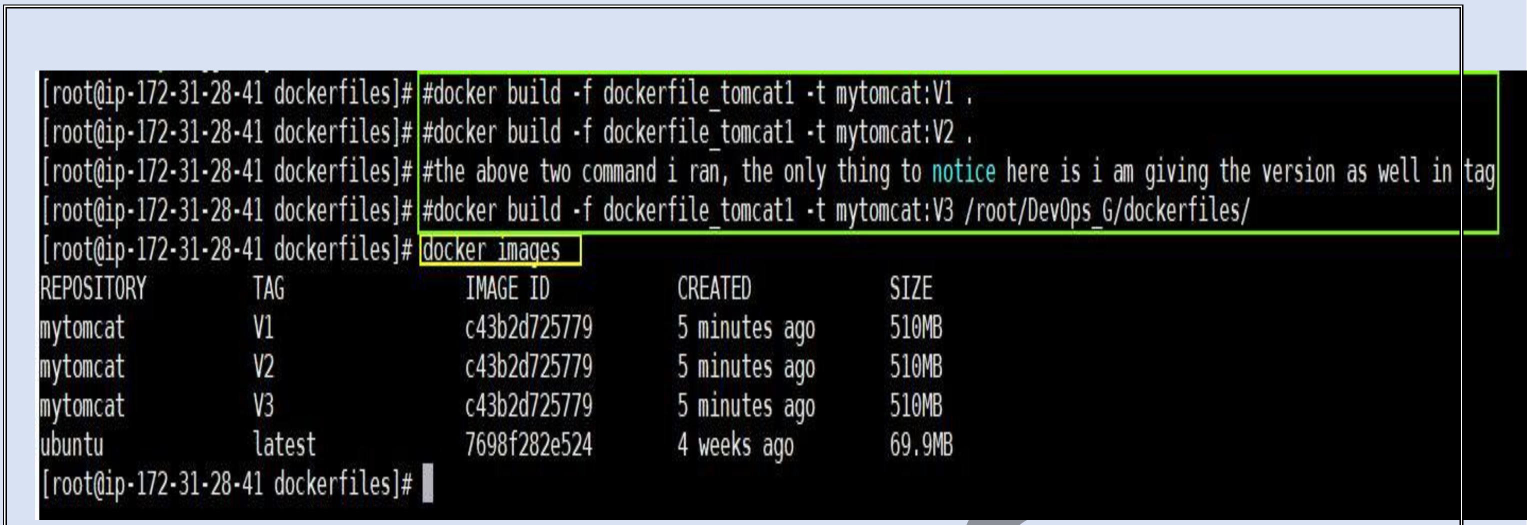
(If we have noticed by default always docker repository is docker.io, so in case of our private DTR we need to specify our repository as well)

We can also save any images in docker in .tgz format using save command and can run the tgz file back to images using load command as shown in below pic:-



```
[root@ip-172-31-28-41 ~]# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
6abc03819f3e: Pull complete
05731e63f211: Pull complete
0bd67c50d6be: Pull complete
Digest: sha256:f08638ec7ddc90065187e7eabdfac3c96e5ff0f6b2f1762cf31a4f49b53000a5
Status: Downloaded newer image for ubuntu:latest
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]# docker save -o ubuntu.tgz ubuntu
[root@ip-172-31-28-41 ~]# ls
anaconda-ks.cfg  DevOps_G  original-ks.cfg  ubuntu.tgz
[root@ip-172-31-28-41 ~]# docker rmi -f ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:f08638ec7ddc90065187e7eabdfac3c96e5ff0f6b2f1762cf31a4f49b53000a5
Deleted: sha256:7698f282e5242af2b9d2291458d4e425c75b25b0008c1e058d66b717b4c06fa9
Deleted: sha256:027b23fdf3957673017df55aa29d754121aee8a7ed5cc2898856f898e9220d2c
Deleted: sha256:0dfbdc7dee936a74958b05bc62776d5310abb129cfde4302b7bcdf0392561496
Deleted: sha256:02571d034293cb241c078d7ecbf7a84b83a5df2508f11a91de26ec38eb6122f1
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
[root@ip-172-31-28-41 ~]# docker load < ubuntu.tgz
02571d034293: Loading layer [==================================================>]  72.32MB/72.32MB
270f934787ed: Loading layer [==================================================>]  15.87kB/15.87kB
8d267010480f: Loading layer [==================================================>]  3.072kB/3.072kB
Loaded image: ubuntu:latest
[root@ip-172-31-28-41 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 ~]# 
```

In last video of docker file I missed the version of tagging, we can give the name of tag along with version as well, here I had given V1, V2 and V3 as tag version which can be seen in snap: -



```
[root@ip-172-31-28-41 dockerfiles]# #docker build -f dockerfile_tomcat1 -t mytomcat:V1 .
[root@ip-172-31-28-41 dockerfiles]# #docker build -f dockerfile_tomcat1 -t mytomcat:V2 .
[root@ip-172-31-28-41 dockerfiles]# #the above two command i ran, the only thing to notice here is i am giving the version as well in tag
[root@ip-172-31-28-41 dockerfiles]# #docker build -f dockerfile_tomcat1 -t mytomcat:V3 /root/DevOps_G/dockerfiles/
[root@ip-172-31-28-41 dockerfiles]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mytomcat            V1                  c43b2d725779        5 minutes ago       510MB
mytomcat            V2                  c43b2d725779        5 minutes ago       510MB
mytomcat            V3                  c43b2d725779        5 minutes ago       510MB
ubuntu              latest              7698f282e524        4 weeks ago         69.9MB
[root@ip-172-31-28-41 dockerfiles]#
```