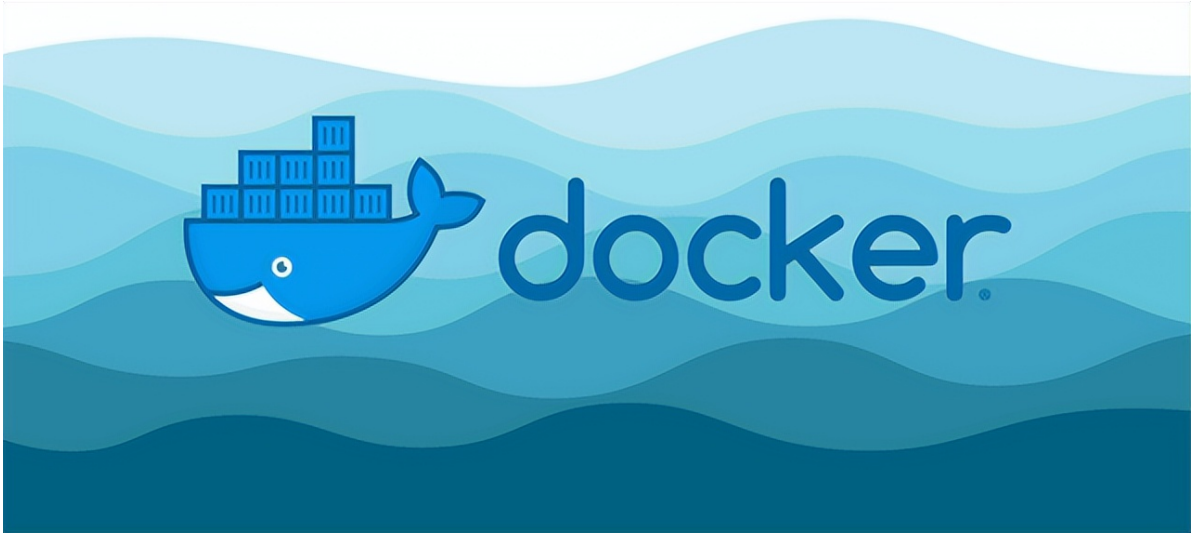
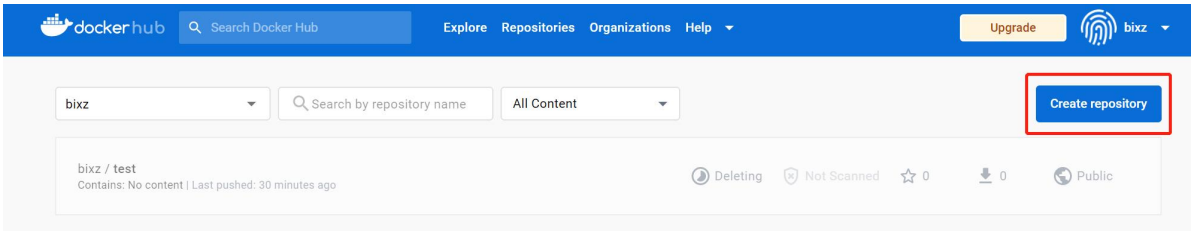
# 什么是仓库?

**仓库就是存储物品的地方，docker的仓库就是存放docker镜像的仓库。**分为在线仓库，类似于github，可以设置公有和私有，本节我们主要介绍Docker hub。还有私有仓库，自己搭建服务器存储docker镜像。

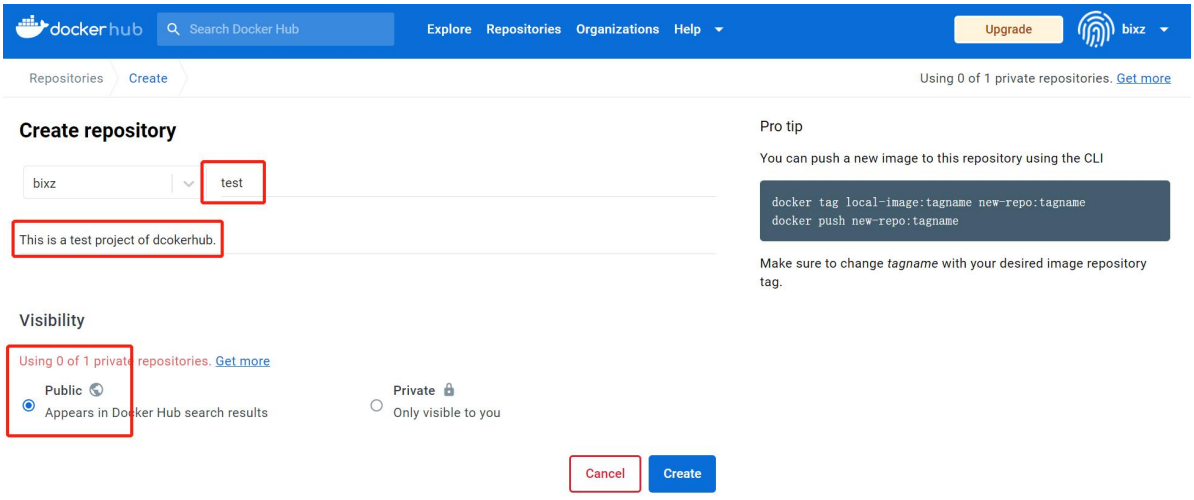



# Docker hub

首先，登录到Docker hub，点击创建仓库：




填写项目名，描述可选，仓库一般选择共有，私有仓库数量有限，需要更多私有仓库需要升级（也就是需要money）：




本地登录：docker login，然后输入用户名和密码

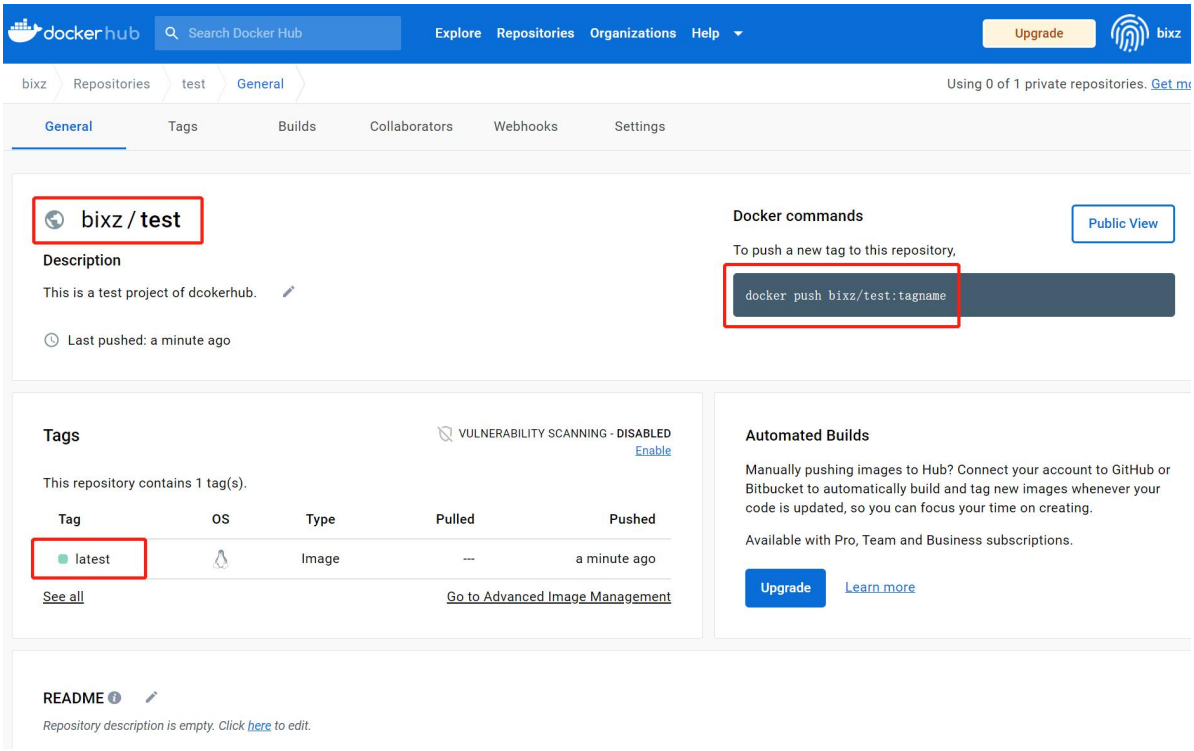
```
root@pc:~# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: bixz
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

tag标签将原有镜像改名：用户名/仓库名:版本

```
root@pc:~# docker tag ubuntu:2.0 bixz/test
root@pc:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
bixz/test     latest    3a25356027d6   4 hours ago    119MB
bixz/ubuntu   2.0       3a25356027d6   4 hours ago    119MB
ubuntu        2.0       3a25356027d6   4 hours ago    119MB
ubuntu        latest    6b7dfa7e8fdb   7 weeks ago    77.8MB
```

没有写版本号，默认是latest：




推送成功，在其他电脑上可以拉取共有test仓库镜像：

```
root@yunpc:~# docker pull bixz/test
Using default tag: latest
latest: Pulling from bixz/test
3e6068eb22e7: Pull complete
Digest: sha256:a31b7fec9065b9d9c6505fd2d3c6e4231174e592c7903
Status: Downloaded newer image for bixz/test:latest
docker.io/bixz/test:latest
root@yunpc:~# docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
bixz/test    latest    3a25356027d6   4 hours ago    119MB
```

# 私有仓库部署

下载registry镜像：`docker pull registry`

配置私有仓库：`vim /etc/docker/daemon.json`

```
{"registry-mirrors": ["http://f1361db2.m.daocloud.io"],"insecure-registries":["192.168.92.134:5000"]}
```

重启服务：`systemctl restart docker`

运行私有仓库：

```
docker run -d --network=host registry
```

更改镜像名称：

```
docker tag ubuntu:latest 192.168.92.134:5000/ubuntu:1.0
```

```
root@pc:~# docker run -d -p 5000:5000 registry
7071bee77ade290ffd173b3ed1c7a2ac32677a06d4a6f662df289c10ee9d79ea
root@pc:~#
root@pc:~# curl 127.0.0.1:5000/v2/_catalog
{"repositories":[]}
```

推送到本地仓库：

```
docker push 192.168.92.134:5000/ubuntu:1.0
curl 192.168.92.134:5000/v2/_catalog
```

```
root@pc:~# docker tag ubuntu:latest 192.168.92.134:5000/ubuntu:1.0
root@pc:~#
root@pc:~# docker images
REPOSITORY                    TAG       IMAGE ID       CREATED        SIZE
192.168.92.134:5000/ubuntu    1.0       6b7dfa7e8fdb   7 weeks ago    77.8MB
ubuntu                        latest    6b7dfa7e8fdb   7 weeks ago    77.8MB
registry                      latest    81c944c2288b   2 months ago   24.1MB
root@pc:~#
root@pc:~# docker push 192.168.92.134:5000/ubuntu
Using default tag: latest
The push refers to repository [192.168.92.134:5000/ubuntu]
tag does not exist: 192.168.92.134:5000/ubuntu:latest
root@pc:~# docker push 192.168.92.134:5000/ubuntu:1.0
The push refers to repository [192.168.92.134:5000/ubuntu]
6515074984c6: Pushed
1.0: digest: sha256:965fbcae990b0467ed5657caceaec165018ef44a4d2d46c7cdea80a9dff0d1ea size: 529
root@pc:~#
root@pc:~# curl 192.168.92.134:5000/v2/_catalog
{"repositories":["ubuntu"]}
root@pc:~#
```

拉取镜像：

```
root@pc:~# docker pull 192.168.92.134:5000/ubuntu:1.0
1.0: Pulling from ubuntu
Digest: sha256:965fbcae990b0467ed5657caceaec165018ef44a4d2d46c7cdea80a9dff0d1ea
Status: Downloaded newer image for 192.168.92.134:5000/ubuntu:1.0
192.168.92.134:5000/ubuntu:1.0
root@pc:~# docker images
REPOSITORY                        TAG       IMAGE ID       CREATED        SIZE
192.168.92.134:5000/ubuntu        1.0       6b7dfa7e8fdb   7 weeks ago    77.8MB
ubuntu                            latest    6b7dfa7e8fdb   7 weeks ago    77.8MB
registry                          latest    81c944c2288b   2 months ago   24.1MB
root@pc:~#
```

# 数据卷和数据卷容器

**数据卷就是将宿主机的某个目录，映射到容器中，作为数据存储的目录，我们就可以在宿主机对数据进行存储。**

首先在宿主机目录下新建一个文件夹，然后进行数据卷映射：

```
docker run -it --name test3 -v ~/data:/root/data ubuntu /bin/bash
```

可以让多个容器映射同一个宿主机目录。

那么如果多个容器有一个相同的共享目录，单个创建目录和映射过程比较繁琐。**所以，使用数据卷容器可以做一个模板映射容器，通过这个模板容器创建的容器就会拥有相同的共用目录。**

创建模板容器：

```
docker create --name myrq -v ~/data ubuntu
```

```
root@pc:~# docker create --name mbrq -v ~/data ubuntu
08784b785fd47f1ad9227070a93d2f0649ea18966fe9b0f3b8c32ca21eedeba4
root@pc:~# docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED             STATUS                       PORTS     NAMES
08784b785fd4   ubuntu     "bash"                   4 seconds ago       Created                              mbrq
77d3b1ca355e   ubuntu     "/bin/bash"              4 minutes ago       Up 4 minutes                         test2
e51176253fb9   ubuntu     "/bin/bash"              5 minutes ago       Up 5 minutes                         test1
6ffaeabd4c5f   registry   "/entrypoint.sh /etc…"   About an hour ago   Exited (2) 17 minutes ago            eloquent_williams
```

通过模板容器创建新的容器：

```
docker run -it --volumes-from myrq --name test3 ubuntu /bin/bash
```

**通过模板容器创建的所有容器都有同样的共享目录。**

查看所有数据卷：

```
root@pc:~# docker volume ls
DRIVER    VOLUME NAME
local     2eaa7b40844fafc7d74ad7c52cfc8011801216649bfbcd6a2bb50d4642e0e35a
local     921d3ede5c0344f83df5229d6542e6c34fba2a0f74b93d669f21c9c50b875730
local     7808effda1496f6ece8f094bfee9a05a447e0a4999ffb88c2ea29ecaff280a76
local     213593e7780640f90a222dddff31777a7a9725d56de4cdcd5fa1a87f9162dcbd
local     db3af9501de08ee83b8dd5296cf215bd02810efc8d56d9d7cd9d4c519de7806c
```

删除所有未使用的数据卷：

```
docker volume prune
```

删除指定编号数据卷：

```
docker volume rm 数据卷编号
```