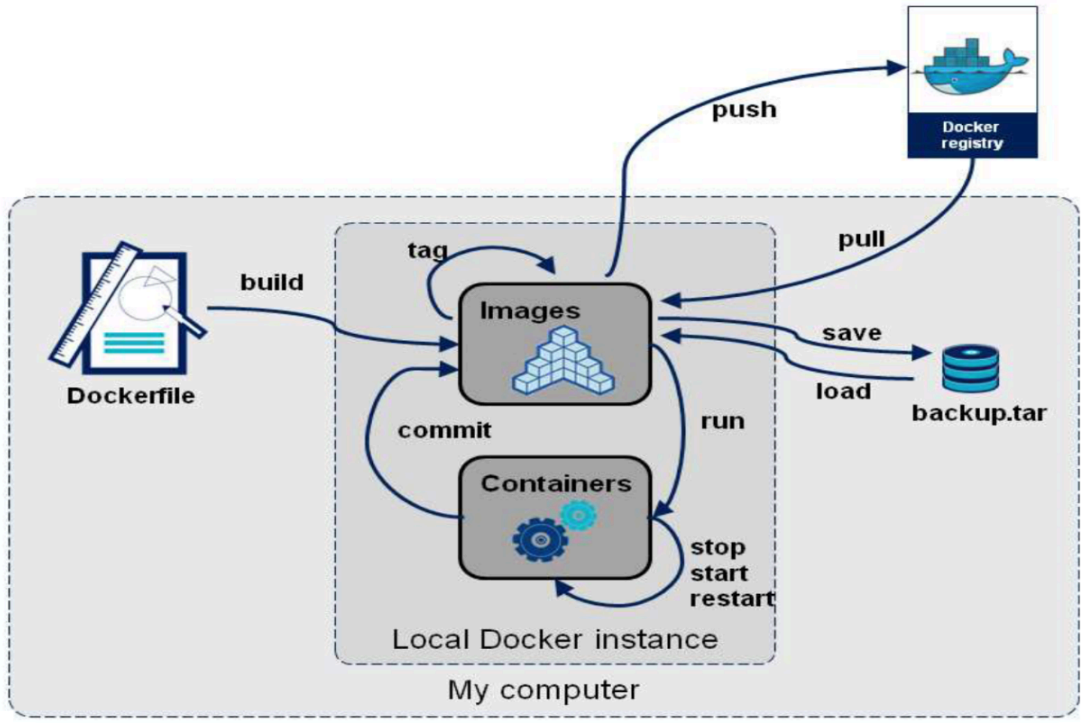
Saving Images and Containers as Tar Files for Sharing



Four basic Docker CLI comes into action:

- The `docker export` - Export a container's filesystem as a tar archive
- The `docker import` - Import the contents from a tarball to create a filesystem image
- The `docker save` - Save one or more images to a tar archive (streamed to STDOUT by default)
- The `docker load` - Load an image from a tar archive or STDIN

Create Nginx Container

```
docker run -d -p 80:80 nginx
```

```
##############################################################
[node1] (local) root@192.168.0.8 ~
$ docker run -d -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
e4fff0779e6d: Pull complete
2a0cb278fd9f: Pull complete
7045d6c32ae2: Pull complete
03de31afb035: Pull complete
0f17be8dcff2: Pull complete
14b7e5e8f394: Pull complete
23fa5a7b99a6: Pull complete
Digest: sha256:447a8665cc1dab95b1ca778e162215839ccbb9189104c79d7ec3a81e14577add
Status: Downloaded newer image for nginx:latest
cbc263d65f81e380e75ef6baf8324cf6ba13636c583bcea3df3ac65225b0ae2f
[node1] (local) root@192.168.0.8 ~
```

Displaying Running Container

`docker ps -a`

```
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                  NAMES
cbc263d65f81   nginx     "/docker-entrypoint.…"   35 seconds ago   Up 34 seconds   0.0.0.0:80->80/tcp   stoic_meitner
[node1] (local) root@192.168.0.8 ~
$
```

```
docker export cbc > nginx.tar
```

```
$ docker export cbc > nginx.tar
[node1] (local) root@192.168.0.8 ~
```

```
$ ls -l
total 185688
-rw-r--r--    1 root     root     190141952 Aug 18 15:56 nginx.tar
[node1] (local) root@192.168.0.8 ~
```

You could commit this container as a new image locally, but you could also use the Docker import command:

`docker import - mynginx < nginx.tar`

```
$ docker import - mynginx < nginx.tar
sha256:63c3c3eede0949009759773bde0910822f5720e43aa53ff56be89bd8d409c281
[node1] (local) root@192.168.0.8 ~
```

```
docker images
```

```
$ docker images
REPOSITORY    TAG        IMAGE ID       CREATED          SIZE
mynginx       latest     63c3c3eede09   24 seconds ago   186MB
nginx         latest     5ef79149e0ec   3 days ago       188MB
[node1] (local) root@192.168.0.8 ~
```

If you wanted to share this image with one of your collaborators, you could upload the tar file on a web server and let your collaborator download it and use the import command on his Docker host.

If you would rather deal with images that you have already committed, you can use the load and save commands:

```
docker save -o mynginx1.tar nginx
```


```
ls -l
```

```
$ ls -l
total 373048
-rw-------    1 root     root     191853568 Aug 18 15:58 mynginx1.tar
-rw-r--r--    1 root     root     190141952 Aug 18 15:56 nginx.tar
[node1] (local) root@192.168.0.8 ~
$
```

```
docker rmi mynginx
```

```
[node1] (local) root@192.168.0.8 ~
$ docker rmi mynginx
Untagged: mynginx:latest
Deleted: sha256:63c3c3eede0949009759773bde0910822f5720e43aa53ff56be89bd8d409c281
Deleted: sha256:8da77e26a7d6bf8c8c6889ebe0de64dfaa3c0ae7353209968eaf3be338ad77c5
[node1] (local) root@192.168.0.8 ~
$
```

Now deleted mynginx image

```
docker images
```

```
$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
nginx        latest    5ef79149e0ec   3 days ago   188MB
[node1] (local) root@192.168.0.8 ~
$
```

```
docker rmi -f nginx:latest
```

```
$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
[node1] (local) root@192.168.0.8 ~
$
```

```
docker load < mynginx1.tar
```

```
$ docker load < mynginx1.tar
Loaded image: nginx:latest
[node1] (local) root@192.168.0.8 ~
$
```

```
docker images
```

```
$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
nginx        latest    5ef79149e0ec   3 days ago   188MB
[node1] (local) root@192.168.0.8 ~
$
```