Docker Simplified
Part -1

1.Installing Docker

```
# wget -qO- https://get.docker.com/ | sh
# systemctl enable docker
# service start docker
# docker --version
# docker info
```

1.1 To make a service account to control Docker without root access

```
$ sudo usermod -aG docker docadm
$ cat /etc/group | grep docadm
docker:x:982:docadm
```

#Logout and login to make the user settings to get updated.

2. Searching Docker registry/repository for specific image versions

```
$ docker search alpine
```

2.1 Search using filters

```
$ docker search --filter=stars=30 --filter=is-automated=true alpine
```
(or)
```
$ docker search -f=stars=30 -f=is-automated=true alpine
```

Complied By :- Barani Kannan
https://github.com/baranikannan/cheatsheats.git
linkedin.com/in/baranikannan-subramanian-19593922

```
$ docker search --filter=stars=5 nginx
$ docker search --filter=is-official=true ubuntu
$ docker search --filter=stars=5 --filter=is-automated=true mysql
```

#AUTOMATED : Tells whether the images is built automatically with a push in GitHub or Bitbucket repositories.

#Get script from below git URL and get list of versions available in repository which requires python3 and requests module

https://github.com/al4/docker-registry-list

```
$ ./docker-registry-list.py alpine
```

3. Docker Image management

#With no version mentioned the latest version will be pulled by default.
```
$ docker pull alpine
```

#Pulls specific version.
```
$ docker pull alpine:2.7
```

#Pulls all versions.
```
$ docker pull -a alpine
```

#To pull from specific private repository
```
$ docker pull registry.mydomain.com:5000/test-image
$ docker pull registry.mydomain.com:5000/projects/test-image:0.2
$ docker pull -a registry.mydomain.com:5000/projects/test-image
```

#To list the images in local registry/repository
```
$ docker images
```
(or)
```
$ docker image ls
```

#To list all versions of a specific image in local registry/repository.
```
$ docker images alpine
$ docker image inspect alpine:2.7
```

(or)
```
$ docker image inspect e738dfbe7a10
```

#To find the information of the specific image.
```
$ $ docker image inspect --format "{{ .Architecture }}" alpine
$ docker image inspect --format "{{ .Metadata }}" alpine
$ docker image inspect --format "{{ .Size }}" alpine
$ docker image inspect --format "{{ .Os }}" alpine
$ docker image inspect --format "{{ .DockerVersion }}" alpine
$ docker image inspect --format "{{ .RepoTags }}" alpine
$ docker image inspect --format "{{ .Created }}" alpine
```

Difference between Save and Export of a image

Docker save will indeed produce a tarball preserving the history with all parent layers and all tags + versions.
docker export does also produce a tarball, but without any layer/history.
A Docker image can be saved to a tarball and loaded back again. This will preserve the history of the image.

#To export the image as tar file.

```
$ docker image save 055936d39205 -o alpine_3.9.4.tar
```
(or)
```
$ docker image save alpine:3.9.4 -o alpine_3.9.4.tar
```
(or)
```
$ docker export a6fc1dbfa81a > latest_alpine.tar
```
(or)
```
$ docker export --output="latest_alpine.tar" a6fc1dbfa81a
```

#To import the image from tar file.

```
$ docker import alpine.2.6.tar alpine:2.6
```
(or)
```
$ docker image import http://image.com/alpineimage.tgz alpine:version
```
(or)
```
$ docker import alpine.2.6.tar alpine/imported:2.6
```
(or)

$ cat alpineimage.tgz | docker image import --message "Imported from tarball" - alpine:new
(or)
$ cat alpineimage.tgz | docker image import - alpine/imagelocal
(or)
$ docker load -i test.tar

Load option will import the image once only whereas import can be done multiple times but with different tags.

#Rename or tag a image. If no version mentioned tag will be added as latest.
$ docker tag b164ab142922 alpine:3.9

#To see the tree structure of images run the below docker instance
$ docker run --rm -v /var/run/docker.sock:/var/run/docker.sock nate/dockviz images -t

(or)

# if docker client using local unix socket
alias dockviz="docker run -it --rm -v /var/run/docker.sock:/var/run/docker.sock nate/dockviz"

# if docker client using tcp
alias dockviz="docker run -it --rm -e DOCKER_HOST='tcp://127.0.0.1:2375' nate/dockviz"

#see the tree structure of images by running the alias as below
$ dockviz images -t

#To see the build history by image ID or name
$ docker image history aa7889871a6d
(or)
$ docker image history alpine:latest

#To remove the image by IMAGE ID or name and version
$ docker rmi e738dfbe7a10
(or)

$ docker rmi alpine:2.7
(or)
$ docker image rm alpine:2.7
(or)
$ docker image rm e738dfbe7a10

#To remove all images
$ docker rmi -f $ (docker images -q)
(or)
$ docker images -q | xargs docker rmi -f

# This will delete all unused images and volumes after confirmation.
$ docker image prune