



Docker Cheatsheet



Docker Cheatsheet



Docker Setup on CentOS

Install dependent packages and Configure Stable Docker Repo on Supported Centos OS Version

```
$ sudo yum install -y yum-utils
```

```
$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

Reference: <https://docs.docker.com/>

Install Latest Docker Engine, Docker CLI and containerd

```
$ sudo yum install -y docker-ce docker-ce-cli containerd.io
```

Start & Enable Docker Service

```
$ sudo systemctl start docker
```

```
$ sudo systemctl enable docker
```

Docker Host Related Commands

Docker CLI and Docker Daemon version details

```
$ sudo docker --version # To check Docker Command Line Version
```

```
$ sudo docker version # To check Docker CLI version and Docker Daemon version (Docker Engine Details)
```

Docker Host Commands

```
$ sudo docker system info # To display System Wide Info like, Memory, CPU, Kernel and Docker Version, Number of Containers and Images, Storage Driver etc.
```

```
$ sudo docker system df # To check storage/disk details used by Docker objects like Containers, Images, Volumes and Build Cache
```

```
$ sudo docker system prune # Release all unused objects like Stopped Containers, Dangling Images, unused Networks etc.
```



Docker - Containers



Start a container in Detached Mode

\$ sudo docker container run -d httpd:latest # To Start a Container in detached mode

Start a container in Attached Mode

\$ sudo docker container run -i -t centos:8 # To Start & get inside a container using “-t” flag to attach pseudo terminal with it for interaction (i.e -t is for Terminal and -i is for Interactive)

List Containers

\$ sudo docker container ps # List high level details for all running container
\$ sudo docker container ps -a # List high level details for all (Stopped and running) containers irrespective of their states
\$ sudo docker container ps -l # List high level details for last created container
\$ sudo docker container ps -q # List “short container ID’s” for running containers only
\$ sudo docker container ps -aq # List “short container ID’s for all stopped and running containers

OR

List Containers

\$ sudo docker container ls # List high level details for all running container
\$ sudo docker container ls -a # List high level details for all (Stopped and running) containers irrespective of their states
\$ sudo docker container ls -l # List high level details for last created container
\$ sudo docker container ls -q # List “short container ID’s” for running containers only
\$ sudo docker container ls -aq # List “short container ID’s for all stopped and running containers



Docker - Containers



Stop a Containers

\$ sudo docker container stop <container-id> or <container-name> **# Stop a container using SIGTERM**

\$ sudo docker container kill <container-id> or <container-name> **# Stop a container using SIGKILL**

Start a Containers

\$ sudo docker container start <container-id> or <container-name> **# Start a container**

Delete Containers

\$ sudo docker container rm <container-id> **# Delete a container (Make sure to stop a container before you remove a container)**

\$ sudo docker container prune **# Delete all stopped containers**

\$ sudo docker container rm \$(docker container ps -aq) **# Delete all containers**

OR

\$ sudo docker container rm \$(docker container ps -aq) **# Delete all containers**

Running a new process inside a Container

\$ sudo docker container exec -it <container-id> /bin/bash **# To Enter inside in the running container with a new “/bin/bash” process**

\$ sudo docker container exec -it <container-id> uname **# To run a command like “uname” inside a container and display the output on the terminal**

Container Troubleshooting Tips

\$ sudo docker container logs <container-id> **# To Check application logs**

\$ sudo docker container stats **# To display Memory, CPU, Network I/O and Disk I/O details for containers**

\$ sudo docker container top <container-id> **# To check the real process ID's running inside a particular container**



Docker - Images



List Local Images inside Docker Host

\$ sudo docker image ls **# To List Images**

To create a new image

\$ sudo docker image build -t <image-repo-name>:<image-tag> -f <Dockerfile-name> <context-path-location> **# To create an Image from Dockerfile Method**

Example: docker image build -t thinknyximage:v1 -f <directory path where docker file is present>

\$ sudo docker container commit -a "Kulbhushan Mayer <kulbhushan.mayer@thinknyx.com>" -m "added App Dependencies" <container-ID> <image-repo-name>:<image-tag>

Create an image by commit the changes done in a container # This is known as Docker Commit Method

\$ sudo docker image tag <image-repo-name>:<image-tag> <new-image-repo-name>:<new-image-tag> **# Tag an existing image with new repo name and tag name**

Ship an Image without a Registry

\$ sudo docker image save <image-repo-name>:<image-tag> -o demo.tar **# To save image as a tar ball with the name demo.tar on Docker Host to share with others**

\$ sudo docker image load -i demo.tar **# To Load an image from the given tar ball on any Docker Host**

Managing an Image with Docker Registry

\$ sudo docker login --u <username> --p <password> **# Login to Default Global Public Registry <https://hub.docker.com>**

\$ sudo docker login registry.thinknyx.com:5000 -u thinknyx -p ***** **# Login to Private Docker Registry registry.thinknyx.com:5000**

\$ sudo docker pull <image-repo-name>:<image-tag> **# Pull an image from global Registry**

\$ sudo docker push <image-repo-name>:<image-tag> **# Push an image to a global Registry (Naming convention of the Image should be <REGISTRYNAME/REPOSITORYNAME/IMAGENAME:TAGNAME>)**

\$ sudo docker pull registry.thinknyx.com:5000/<image-repo-name>:<image-tag> **# Pull an image from private Registry**

\$ sudo docker push registry.thinknyx.com:5000/<image-repo-name>:<image-tag> **# Push and image from private Registry**

Delete Images

\$ sudo docker image prune **# Delete all unused & Dangling Images**

\$ sudo docker image rm <image-repo-name>:<image-tag> **# Delete specific image or images (Make sure the image is not in use by the containers before removal)**



Docker – Networks & Volumes



List Docker Networks

\$ sudo docker network ls **# To List local Docker networks**

Create User Defined Docker Network

\$ sudo docker network create <network-name> --driver <driver-name> --subnet 192.168.0.0/16 --gateway 192.168.0.1 **# To create a new network with defined Subnet and Gateway**

Add or Remove Network to containers

\$ sudo docker network connect <network-name> <container-id> **# To Connect a particular container to a new network**

\$ sudo docker network disconnect <network-name> <container-id> **# To disconnect a particular container to a new network**

Delete a Network

\$ sudo docker network prune **# Delete All unused docker networks**

\$ sudo docker network rm <network-id> **# Delete specific network or networks**

List Docker Volumes

\$ sudo docker volume ls **# To List Volumes**

Create User Defined Volume

\$ sudo docker volume create <volume-name> **# To create a new volume, default volume location is “/var/lib/docker/volumes/”**

Delete a Volume

\$ sudo docker volume prune **# Delete All unused volumes**

\$ sudo docker volume rm <volume-id> **# Delete specific volume or volumes**



Docker – Sample Dockerfile



Sample Dockerfile to Create a HTTPD Server Image based centos:7

```
FROM      centos:7
RUN       yum install -y httpd
WORKDIR   /var/www/html
COPY      ./index.html .
EXPOSE    80/tcp
CMD       ["httpd","-D","FOREGROUND"]
```

Docker command to Build an Image using above Dockerfile

```
$ sudo docker image build -t thinknyx/httpd:v1 .
```

Here . (dot) stands for current working directory where Dockerfile is present

Docker command to create and start a container using newly created image

```
$ sudo docker container run -itd -p 80:80 --name=webcontainer thinknyx/httpd:v1
```

```
[thinknyx@devops-in-action docker]# sudo docker ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7984adb04634	thinknyx/httpd:v1	"httpd -D FOREGROUND"	6 seconds ago	Up 5 seconds	0.0.0.0:80->80/tcp	webcontainer



Thinknyx[®]

YOU TRUST, WE DELIVER



www.thinknyx.com



support@thinknyx.com



+91 9810344919/9717917973

Reach out to us at: