

Eknaprasath's Blog For Linux Technologies

A Technology Guide

[stay updated via rss](#)

Docker Tutorial

Posted: July 14, 2018 in [Devops](#)

0

Docker Have Two editions

Docker EE (Enterprise Edition)

Docker CE (Community Edition)

Example to run a container

`docker container run -it --name=name_to_this_container image_name`

`docker container run -it --name=name_to_this_container -d image_name` -> used to start container in background

`docker container run -it --name=bb busybox`

`docker container run -it --name=bb1 -d busybox`

To exit from a container

exit (or) `ctrl+p ctrl+q`

Check running and exited containers

`docker ps`

`docker ps -a`

`docker container ls`

`docker container ls -a`

To login into a container

`docker exec -it container-name sh (or) bash`

`docker container exec -it container_name sh (or) bash`

`docker attach container-name`

`docker exec -it bb sh (or) bash`

`docker container exec -it bb sh (or) bash`

`docker attach bb` → if we used this command to login we need to use “`ctrl+p ctrl+q`” to exit, if we use `exit` command container will be stopped.

To stop container

`docker stop container-name`

`docker container stop container-name`

`docker stop bb`

`docker container stop bb1`

To start a stopped container

`docker start container-name`

`docker container start container_name`

`docker start bb`

`docker container start bb`

To remove docker

`docker rm bb`

`docker container rm bb`

Default Network

Following Networks exists in all docker servers

- * none
- * host
- * bridge (default)

The “Bridge” Network in more Detail

- * Traffic to each host goes through a NAT gateway
- * container IP addresses are randomly assigned from a private pool
- * containers on the same bridge can connect to each other by IP

The “Host” Network in more Detail

- * For standalone containers,
- * remove network isolation between the container and the Docker host, and use the host's networking directly.
- * host is only available for swarm services on Docker 17.06 and higher

The “None” Network in more Detail

- * For this container, disable all networking.
- * Usually used in conjunction with a custom network driver.
- * none is not available for swarm services

DNS in Containers

- * Docker runs an internal DNS server for containers
- * The DNS server is accessible at 127.0.0.11 from within the container

```
docker container run -it --dns=192.168.1.1 --dns-search="example.com" --name=bb busybox
```

it will made entry in resolve.conf file inside container

User defined Bridge Networks

- * Preferred way to connect containers
- * Containers on separate networks are isolated from each other
- * Can connect to many networks as needed
- * Docker provides DNS service discovery based the container's name

```
docker network create --driver bridge frontend
```

```
docker network create --driver bridge backend
```

```
docker network ls
```

Now create containers on each networks

```
docker container run -it --network=frontend -d --name=front busybox
```

```
docker container run -it --network=backend -d --name=back busybox
```

lauch a test container to ping test

```
docker container run -it --network=frontend --name=test busybox
```

```
ping -c 2 front
```

```
ping -c 2 back
```

To connect to different network

```
docker network connect backend test
```

```
ping back
```

Docker Image

Image Layers

- * Images are made of layers
- * Each instructions in the Dockerfile creates a new layers
- * Each layer is the set of differences
- * Running containers have thier own Read write layer

```
docker history --no-trunc nginx
```

```
root@ip-172-31-87-86:~# docker history ubuntu:15.04
```

```
IMAGE CREATED CREATED BY SIZE COMMENT
```

```
This Read Write layer --> Container Layers
```

```
d1b55fd07600 2 years ago /bin/sh -c #(nop) CMD ["/bin/bash"] 0B --> Read only
```

```
<missing> 2 years ago /bin/sh -c sed -i 's/^#\s*\s*(deb.*universe\)$... 1.88kB --> Read only
```

```
<missing> 2 years ago /bin/sh -c echo '#!/bin/sh' > /usr/sbin/poli... 701B --> Read only
```

```
<missing> 2 years ago /bin/sh -c #(nop) ADD file:3f4708cf445dc1b53... 131MB --> Read only
```

- * Layers are shared between images

- * Existing layers do not have downloaded again

- * Saves time and disk space

Sharing Images

- * Each container gets its own "Container Layer"

- * Changes only happen to the container layer

- * The container layer is deleted when the container is deleted

Building an IMAGE

- * Docker build

- * Looks for "Dockerfile" in the base directory

- * use -t option to name your image_name

- * Image is only available if the build completed successfully

- * Previous build layers will be reused

- * the -f flag can be used to specify a different Dockerfile

```
docker build -t my-image_name .
```

```
docker build -t my-image_name -f Dockerfile-bleed .
```

Create a file called Dockerfile-copy-file and update the following

```
FROM httpd:latest
```

```
COPY index.html /usr/local/apache2/htdocs
```

create a index.html file in present working directory and update the following

```
<html><body><h1>This file deployed using Dockerfile</h1></body></html>
```

Run the below command to create a image using docker file

```
docker build -t demo -f Dockerfile-copy-file .
```

Create a container using image which created

```
docker run -it -p 80:80 -d demo
```

```
docker run -rm -it demo ls -R /usr/local/apache2/htdocs/dir/ --> To delete container after accessed
```

Demo Copy Directory

```
FROM httpd:latest
```

```
COPY folder /usr/local/apache2/htdocs/dir/
```

Distination Paths

- * Destination data can be relative or absolute
- * Paths that do not end in "/" are treated as a regular file
- * Paths that end in "/" is treated as a directory

Download a file

ADD src ... dest

- * ADD <https://foo/example> (<https://foo/example>) /dest/example/
- * If neither the source URI or the destination end with a slash, the file is downloaded and copied to the destination
- * If both the source url and the destination end with a slash, the downloaded file is copied in to the destination directory.

Unpack Tar files

- * Unpack local tar files
- * Can be compressed with gzip, bzip2 and xz
- * Does not work with URLs

Create a Dockerfile called "Dockerfile-unzip-files

```
FROM debian:stretch
```

```
ADD test.tar.gz /data/
```

```
tar -ztf test.tar.gz
```

```
docker build -t imagename -f Dockerfile-unzip-files .
```

```
docker run --rm -it imagename ls -R /data/
```

Source Pattern Matching.

- * Matches zero or more characters
- ? Matches exactly one character
- \\ Escape character
- [,.[^] Character class

COPY test/[ac].txt /data/ -> This will copies a.txt & c.txt to /data/

COPY test/[!a].txt /data/ -> This will copies every .txt files except a.txt

Best Practies

- * Use a separate COPY or ADD instructions for each file or directory
- * Use COPY rather that ADD
- * Do not use ADD to download files which will be deleted later on build process which will create a larger image
- * Add one-off administrative scripts

Running Commands to customize an Image

Shell form

* Command is passed to the default shell

/bin/sh -C -> for linux container
cmd /S /C -> for windows container

* Works like the shell

* Succeeds if the command returns a valid success code

FROM debian:stretch

RUN touch /tmp/test

FROM debian:stretch

RUN apt-get update && \

apt-get install -y \

build-essential \

git \

golang \

python-pygments \

rsync \

ruby-dev \

rubygems \

ssh-client \

wget

Pipe Problems

We need to use the below format if we need to use pipe

FROM buildpack-deps:stretch-curl

RUN wget -O - <https://google/>(<https://google/>) | wc -l

If we use above format to create image it will success even if command failes
so we need to use alternate method

FROM buildpack-deps:stretch-curl

RUN /bin/bash -C "set -O pipefail && wget 0O - <https://google/>(<https://google/>) | wc -l

Changing the Default Shell

* SHELL instruction changes the default shell

* Specified in JSON format

* Later RUN instructions will use the new shell.

FROM microsoft/windowsservercore

SHELL ["powershell", "-NoProfile", "-Command"]

EXEC from

- * Does not require a shell
- * No Variable replacement
- * JSON format
- * Enclose entries in double quotes ("")
- * Escape characters with backslash (\)

FROM debian:stretch

RUN touch /this-is-shell-form-\${HOSTNAME}

RUN ["touch" , "/this-is-exec-form-\${HOSTNAME}"]

[Create a free website or blog at WordPress.com.](#)