

Installing Docker Compose .

built on docker engine .

This is pre installed

Verify

docker -compose --Version

Remove all Containers and images.

Remove all
Containers

docker Container rm
-f \$(docker
Container ls -aq)
→ a is for stopped Container
→ q is for just container
ids.

Remove all
images

docker image rm
-f \$(docker
image ls -q)

Alternative

Can click on docker
icon & it has an
option to clean/purge
data.

Sample Web application

Frontend
backend

- react app
- node project

docker - Compose

Reduces the commands required to run all projects with a single command.

Command

docker - Compose up
→ It will install db, front end, backend and runs project.
Also installs migration script.

Creating a Compose file

Version

Refer in google which version for docker compose version.

Version : "3.8"

services :

web :

api :

db :

full file
in next
page -

Version: "3.8"

Services:

Web:

build: ./frontend

Ports:

- 3000:3000

api:

build: ./backend

Ports:

- 3001:3001

Environment:

DB_URL=mongodb://db/vidly

db:

image: mongo:4.0-xenial

Ports:

- 27017:27017

Volumes:

- vidly:/db/data

Volumes:

vidly:

Service has app names

→ where docker file is located

mongodb://[host]/[db]
Connection string

image here is pulled from docker hub
- 27017 is default mongodb port

/db/data - Default storage for mongodb

This is to define volume

Building images

docker-compose

lists all sub commands.

docker-compose build
-- help.

→ lists options.

1. --no-cache

If weird issues encountered
then could be because of
cache, so use this

2. -- pull

Always pulls newer
version of image

build

docker-compose build

docker images.

Vidly-web latut
Vidly-api latut
db

docker compose automatically
prefixes with name of
directory - here vidly.

Starting and stopping applications

Command

docker-compose up

Useful options

1. --build
forces rebuild & up
combined.

2. -d
detached mode.

See all
running
docker compose
Containers

docker-compose ps

vidly-api-1 } 1 is
vidly-db-1 } appended
vidly-web-1 } to represent
diff containers
using same image

If we go to local host : 3000
app is up

bring entire
app down

docker-compose down.

Containers are removed
but images are still
available.

Docker networking

What is it?

docker Compose creates
a network under the
hood connecting all

how to see

docker-compose up -d
→ It will print below
Creating network
"vidly-default" with
default driver

Command

docker network ls

bridge
host
none

vidly-default

} not same
ignore

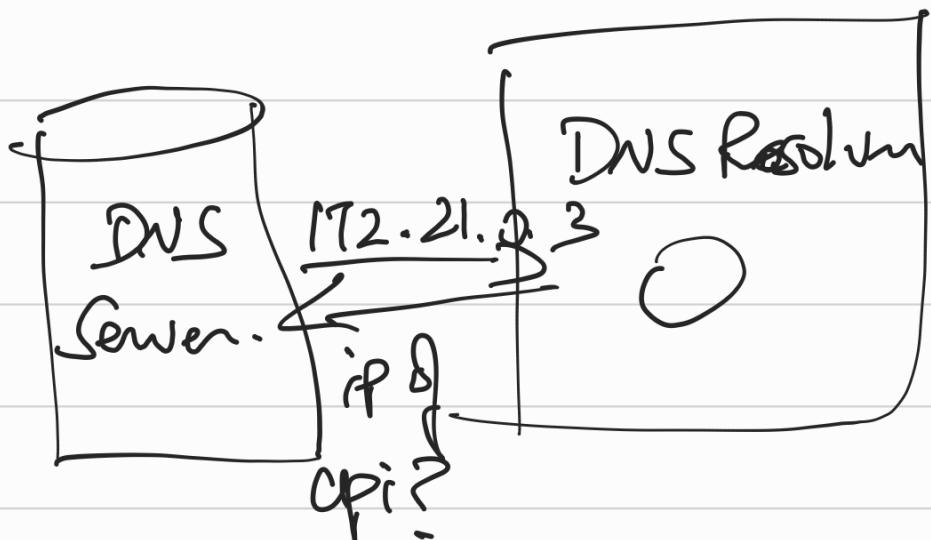
vidly - default

This contains 3 host /
Containers
web, api, db.

To check
Start a shell
session and
ping a container

docker exec -it -u root
fcb sh
/app # ping api
→ It's able to ping.

Container



ifconfig

eth0 inet 172.21.0.2
lo

Coming to
this new
variable we
defined in
docker Compose
file

DB-URL : mongodb://
db Pividly

db is able to be
resolved because it
gets IP address from
DNS server within
docker.

This is how
api service is able
to access db.

Viewing logs

docker-compose logs.

- Lists all logs.

options

-f - follow
-t - timestamps.
--tail

If you want
logs from
single service
only. like
api/web

docker logs sc6 -f

Publishing changes

docker-compose api:
file

⋮
⋮
⋮

Volumes :

- . /backend : /app

Now running docker
build gives below error
Sh: nodemon not found

What is node
mon.

It is responsible for
detecting change & restart
node server.

But it's available
in package-lock
json.

npm install runs inside
Container but not available
in host so error

Run npm install in local -

Error goes away.

Now index.js file modified in api project -

Reflects immediately .

localhost:3000/api

Migrating database -

We are using a database migration tool called `migrate-mongo.js`

It is under package.json dependencies.

Command

`migrate-mongo up`

or

`npm run db:up-`

`docker-compose`

api:

-
-
-

Command =
migrate-mongo up
~~&~~ npm start.

docker wait
for service
[Same wait
for script]

Command :
. /wait-for
db:27017 ~~&~~
migrate-mongo up
~~&~~ npm start.

Running automated test

Normal run
test -

It can be
run inside
Container too
(but it is slow)

npm test .

Web-test :

image : vidly-web

volumes :

- . /frontend : /app

Command : npm test

docker-compose up.

It runs tests now.