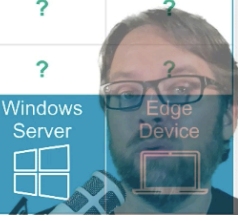
DOCKER

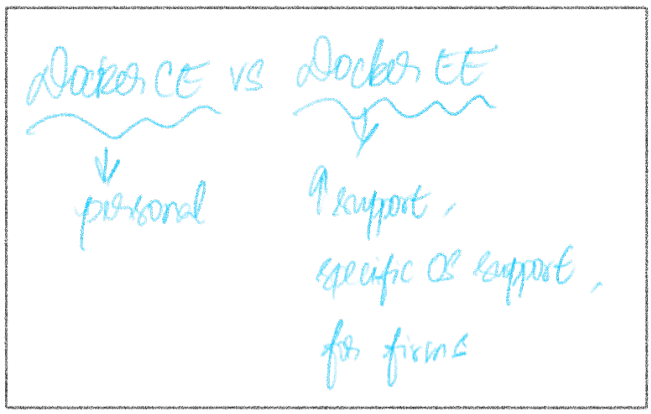
Docker Mastery

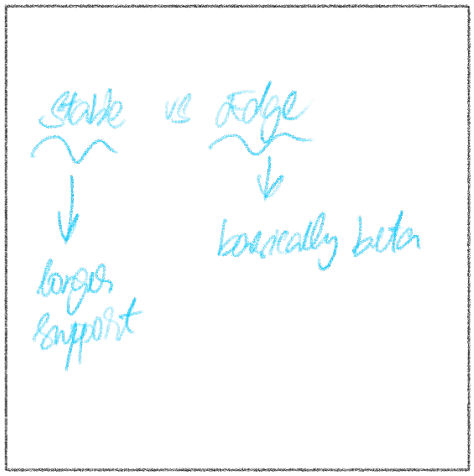
- Section 1: Openource project - Docker Inc - Containers - once in a decade shift
- point of Docker - Speed - full software life cycle gets much faster
- Matrix of Hell - Kotlin - docker/containers - run everywhere but build once -
- March celebrations

Static Website	?	?	?	?	?	?	?	?
Web Frontend	?	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?	?
	Desktop	Test/QA Cluster	Production Cluster	Public Cloud	Data Center	Mainframe	Windows Server	Edge Device



Section-1: Installation





Docker install link - https://hub.docker.com/editions/community/docker-ce-desktop-mac https://hub.docker.com/editions/community/docker-ce-desktop-mac

place example code snippets in permalled paths in file sharing

Section 2 - Containers

Image vs. Container

- An Image is the application we want to run
- A Container is an instance of that image running as a process
- You can have many containers running off the same image
- In this lecture our image will be the Nginx web server
- Docker's default image "registry" is called Docker Hub (hub.docker.com)

$ docker container run --publish 80:80 nginx (new way)

local port → host port

* Downloads nginx from Docker Hub (source for the img)

↳ link to Github for open source

* Start new container for the image
* left is client port, right host port
* routes traffic from host ip/port to container ip (client)

↳ client ip should be free.

$ docker container run --publish 80:80 --detach nginx

↳ run in background.

$ docker container stop (stop but don't remove)

$ docker container ls (→ list running containers)

$ docker container ls -a

↳ since no name is specified during create → auto generated

$ docker container run --publish 80:80 --detach --name anyName nginx

↳ for custom container names

$ docker container logs anyName

$ docker container top anyName

$ docker container rm b2f 690 ade

↳ each is a set of 1st 3 chars from the container's generated

↳ rm multiple containers (non-running stopped) — else use "-f" flag

What happens in 'docker container run'

1. Looks for that image locally in image cache, doesn't find anything
2. Then looks in remote image repository (defaults to Docker Hub)
3. Downloads the latest version (nginx:latest by default)
4. Creates new container based on that image and prepares to start
5. Gives it a virtual IP on a private network inside docker engine
6. Opens up port 80 on host and forwards to port 80 in container
7. Starts container by using the CMD in the image Dockerfile

Containers vs VMs

- Containers are just restricted processes running inside the host OS.
- Running a db as a docker process: $docker run --name mongo -d mo
- docker ps
- docker top mongo
- docker stop mongo
- in linux → ps aux lists mongo as just a process nothing n
  in mac/win → since docker runs as small VMs it's not so direc
  check out this link:

https://www.bretfisher.com/docker-for-mac-commands-for-getting-into-local-docker-vm/

Assignment Question

Assignment: Manage Multiple Containers

- docs.docker.com and `--help` are your friend
- Run a `nginx`, a `mysql`, and a `httpd` (apache) server
- Run all of them `--detach` (or `-d`), name them with `--name`
- nginx should listen on `80:80`, httpd on `8080:80`, mysql on `3306:3306`
- When running `mysql`, use the `--env` option (or `-e`) to pass in `MYSQL_RANDOM_ROOT_PASSWORD=yes`
- Use `docker container logs` on mysql to find the random password it created on startup
- Clean it all up with `docker container stop` and `docker container rm` (both can accept multiple names or ID's)
- Use `docker container ls` to ensure everything is correct before and after cleanup


```
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS
       NAMES
4938ffddfb42   mysql     "docker-entrypoint.s…"   About a minute ago   Up About a minute   33060/tcp, 0.0.0.0:3307->330
6/tcp   mysql
c9fee66b9628   httpd     "httpd-foreground"       4 minutes ago        Up 4 minutes        0.0.0.0:8080->80/tcp
       httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   7 minutes ago        Up 7 minutes        0.0.0.0:80->80/tcp
       nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container stop 49 c9 64
49
c9
64
sanjitkumar@Sanjits-MacBook-Air ~ % docker container rm 49 c9 64
49
c9
64
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
sanjitkumar@Sanjits-MacBook-Air ~ %
```

```
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                   NAMES
c9fee66b9628   httpd     "httpd-foreground"       8 seconds ago   Up 7 seconds   0.0.0.0:8080->80/tcp    httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp      nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 3306:3306 --detach -e MYSQL_RANDOM_ROOT_PASSWORD=yes
 --name mysql mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
852e50cd189d: Already exists
29969ddb0ffb: Pull complete
a43f41a44c48: Pull complete
5cdd802543a3: Pull complete
b79b040de953: Pull complete
938c64119969: Pull complete
7689ec51a0d9: Pull complete
a880ba7c411f: Pull complete
984f656ec6ca: Pull complete
9f497bce458a: Pull complete
b9940f97694b: Pull complete
2f069358dc96: Pull complete
Digest: sha256:4bb2e81a40e9d0d59bd8e3dc2ba5e1f2197696f6de39a91e90798dd27299b093
Status: Downloaded newer image for mysql:latest
582b3d3db454f3ab697cde0d260080447cee65d470dd28f21f2ea83f38838c8d
docker: Error response from daemon: Ports are not available: listen tcp 0.0.0.0:3306: bind: address already in use.
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 3307:3306 --detach -e MYSQL_RANDOM_ROOT_PASSWORD=yes
 --name mysql mysql
docker: Error response from daemon: Conflict. The container name "/mysql" is already in use by container "582b3d3db454f
3ab697cde0d260080447cee65d470dd28f21f2ea83f38838c8d". You have to remove (or rename) that container to be able to reuse
 that name.
See 'docker run --help'.
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS         PORTS                   NAMES
582b3d3db454   mysql     "docker-entrypoint.s…"   54 seconds ago   Created                                mysql
c9fee66b9628   httpd     "httpd-foreground"       2 minutes ago    Up 2 minutes   0.0.0.0:8080->80/tcp    httpd
```

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'Serve
rName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'Serve
rName' directive globally to suppress this message
[Sat Dec 05 07:12:49.665318 2020] [mpm_event:notice] [pid 1:tid 140245473633408] AH00489: Apache/2.4.46 (Unix) configur
ed -- resuming normal operations
[Sat Dec 05 07:12:49.665767 2020] [core:notice] [pid 1:tid 140245473633408] AH00094: Command line: 'httpd -D FOREGROUND
'
^C[Sat Dec 05 07:13:31.628687 2020] [mpm_event:notice] [pid 1:tid 140245473633408] AH00491: caught SIGTERM, shutting do
wn
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                 NAMES
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp    nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS                      PORTS
 NAMES
73edfad11bd5   httpd     "httpd-foreground"       About a minute ago   Exited (0) 20 seconds ago
 httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago        Up 2 minutes                0.0.0.0:80->80/tcp
 nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container rm 73e -f
73e
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                 NAMES
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp    nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 8080:80 --detach --name httpd httpd
c9fee66b9628c4df18685f10051a21b16e9d556024b7e60935c44c88ae3cb6e8
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                   NAMES
c9fee66b9628   httpd     "httpd-foreground"       4 seconds ago   Up 2 seconds   0.0.0.0:8080->80/tcp    httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp      nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                   NAMES
```

```
The files /Users/sanjitkumar/app and /Users/sanjitkumar/safari do not exist.
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 80:80 --detach --name nginx nginx
64c69fba10c30734910e0515e1577afeb6f05f5c539cc7dff62b63f41d038354
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                NAMES
64c69fba10c3   nginx     "/docker-entrypoint.…"   8 seconds ago   Up 8 seconds   0.0.0.0:80->80/tcp   nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 8080:80 --name httpd httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
852e50cd189d: Already exists
67d51c33d390: Pull complete
b0ad2a3b9567: Pull complete
136f1f71f30c: Pull complete
01f8ace29294: Pull complete
Digest: sha256:fddc534b7f6bb6197855be559244adb11907d569aae1283db8e6ce8bb8f6f456
Status: Downloaded newer image for httpd:latest
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'Serve
rName' directive globally to suppress this message
```

```
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS
        NAMES
4938ffddfb42   mysql     "docker-entrypoint.s…"   About a minute ago   Up About a minute   33060/tcp, 0.0.0.0:3307->330
6/tcp   mysql
c9fee66b9628   httpd     "httpd-foreground"       4 minutes ago        Up 4 minutes        0.0.0.0:8080->80/tcp
        httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   7 minutes ago        Up 7 minutes        0.0.0.0:80->80/tcp
        nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container stop 49 c9 64
49
c9
64
sanjitkumar@Sanjits-MacBook-Air ~ % docker container rm 49 c9 64
49
c9
64
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
sanjitkumar@Sanjits-MacBook-Air ~ %
```

```
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                    NAMES
c9fee66b9628   httpd     "httpd-foreground"       8 seconds ago   Up 7 seconds   0.0.0.0:8080->80/tcp     httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp       nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 3306:3306 --detach -e MYSQL_RANDOM_ROOT_PASSWORD=yes
 --name mysql mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
852e50cd189d: Already exists
29969ddb0ffb: Pull complete
a43f41a44c48: Pull complete
5cdd802543a3: Pull complete
b79b040de953: Pull complete
938c64119969: Pull complete
7689ec51a0d9: Pull complete
a880ba7c411f: Pull complete
984f656ec6ca: Pull complete
9f497bce458a: Pull complete
b9940f97694b: Pull complete
2f069358dc96: Pull complete
Digest: sha256:4bb2e81a40e9d0d59bd8e3dc2ba5e1f2197696f6de39a91e90798dd27299b093
Status: Downloaded newer image for mysql:latest
582b3d3db454f3ab697cde0d260080447cee65d470dd28f21f2ea83f38838c8d
docker: Error response from daemon: Ports are not available: listen tcp 0.0.0.0:3306: bind: address already in use.
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 3307:3306 --detach -e MYSQL_RANDOM_ROOT_PASSWORD=yes
 --name mysql mysql
docker: Error response from daemon: Conflict. The container name "/mysql" is already in use by container "582b3d3db454f
3ab697cde0d260080447cee65d470dd28f21f2ea83f38838c8d". You have to remove (or rename) that container to be able to reuse
 that name.
See 'docker run --help'.
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS         PORTS                    NAMES
582b3d3db454   mysql     "docker-entrypoint.s…"   54 seconds ago   Created                                 mysql
c9fee66b9628   httpd     "httpd-foreground"       2 minutes ago    Up 2 minutes   0.0.0.0:8080->80/tcp     httpd
```

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'Serve
rName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'Serve
rName' directive globally to suppress this message
[Sat Dec 05 07:12:49.665318 2020] [mpm_event:notice] [pid 1:tid 140245473633408] AH00489: Apache/2.4.46 (Unix) configur
ed -- resuming normal operations
[Sat Dec 05 07:12:49.665767 2020] [core:notice] [pid 1:tid 140245473633408] AH00094: Command line: 'httpd -D FOREGROUND
'
^C[Sat Dec 05 07:13:31.628687 2020] [mpm_event:notice] [pid 1:tid 140245473633408] AH00491: caught SIGTERM, shutting do
wn
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                NAMES
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp   nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS                      PORTS
 NAMES
73edfad11bd5   httpd     "httpd-foreground"       About a minute ago   Exited (0) 20 seconds ago
 httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago        Up 2 minutes                0.0.0.0:80->80/tcp
 nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container rm 73e -f
73e
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                NAMES
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp   nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 8080:80 --detach --name httpd httpd
c9fee66b9628c4df18685f10051a21b16e9d556024b7e60935c44c88ae3cb6e8
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                    NAMES
c9fee66b9628   httpd     "httpd-foreground"       4 seconds ago   Up 2 seconds   0.0.0.0:8080->80/tcp     httpd
64c69fba10c3   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp       nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                    NAMES
```

```
The files /Users/sanjitkumar/app and /Users/sanjitkumar/safari do not exist.
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 80:80 --detach --name nginx nginx
64c69fba10c30734910e0515e1577afeb6f05f5c539cc7dff62b63f41d038354
sanjitkumar@Sanjits-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                NAMES
64c69fba10c3   nginx     "/docker-entrypoint.…"   8 seconds ago   Up 8 seconds   0.0.0.0:80->80/tcp   nginx
sanjitkumar@Sanjits-MacBook-Air ~ % docker container run --publish 8080:80 --name httpd httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
852e50cd189d: Already exists
67d51c33d390: Pull complete
b0ad2a3b9567: Pull complete
136f1f71f30c: Pull complete
01f8ace29294: Pull complete
Digest: sha256:fddc534b7f6bb6197855be559244adb11907d569aae1283db8e6ce8bb8f6f456
Status: Downloaded newer image for httpd:latest
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'Serve
rName' directive globally to suppress this message
```

What's going on inside containers?

What's Going On In Containers

- `docker container top` - process list in one container
- `docker container inspect` - details of one container config
- `docker container stats` - performance stats for all containers

returns a JSON array with all data about how the container was started

↓

basically metadata about container - configuration, volumes, networking

displays live stream of container's resource usage stats

eg: CPU, memory usage, network IO etc

→ Make sure no response over net.

Getting a shell inside Containers

Getting a Shell Inside Containers

- `docker container run -it` - start new container interactively
- `docker container exec -it` - run additional command in existing container
- Different Linux distros in containers

How to do stuff inside the container ⇝ NO need for SSH add'n inside.
⇝ Docker CLI enough

-it
- -i → --interactive → keep session open to run terminal ip
- -t → --tty → Allocate a pseudo-T (i.e - simulate real TO like what SSH does)

eg: $ docker container run -it --name proxy nginx bash

~> running 2nd nginx container server.

~> followed by bash as a command to exec after container creation
Here it means "with a bash terminal inside running container"

$ exit

exit the bash.

~> Note: this also stops the container on exit.
∴ the container runs only so long as the command which started it runs
↳ here we overwrote the default cmd command with "bash"
↳ so on exit it stopped.

$ docker container run -it --name ubuntu ubuntu

↳ full linux distro - ubuntu
↳ default cmd itself throws you into a bash
↳ have access to apt-get & other default ubuntu software ($ apt-get vim)

~> Note that a default ubuntu image VM is much bigger than its containers.

$ exit

$ docker container start -ai ubuntu (to still have vim)

$ docker container exec -it mysql bash

↝ perform admin actions

↝ on exit the default daemon doesn't close (see exec cmd)

$ docker container run -it alpine bash

↝ won't work w/ alpine linux distro (v small size) — no bash

↝ can only run programs/processes already existing in the image

↝ alpine has "sh" instead & "apk" package manager

↳ can use it to install bash

Docker Networks

Docker Networks: Concepts

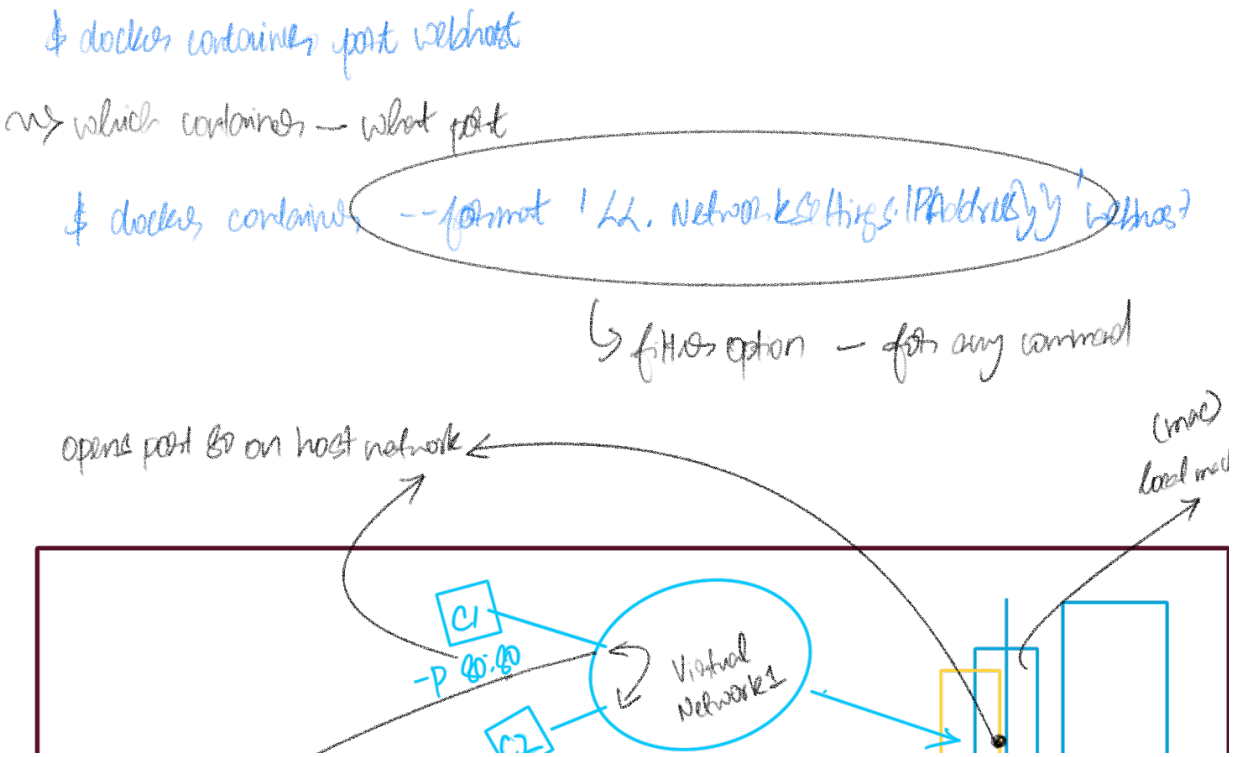
- Review of `docker container run -p`
- For local dev/testing, networks usually "just work"
- Quick port check with `docker container port <container>`
- Learn concepts of Docker Networking
- Understand how network packets move around Docker

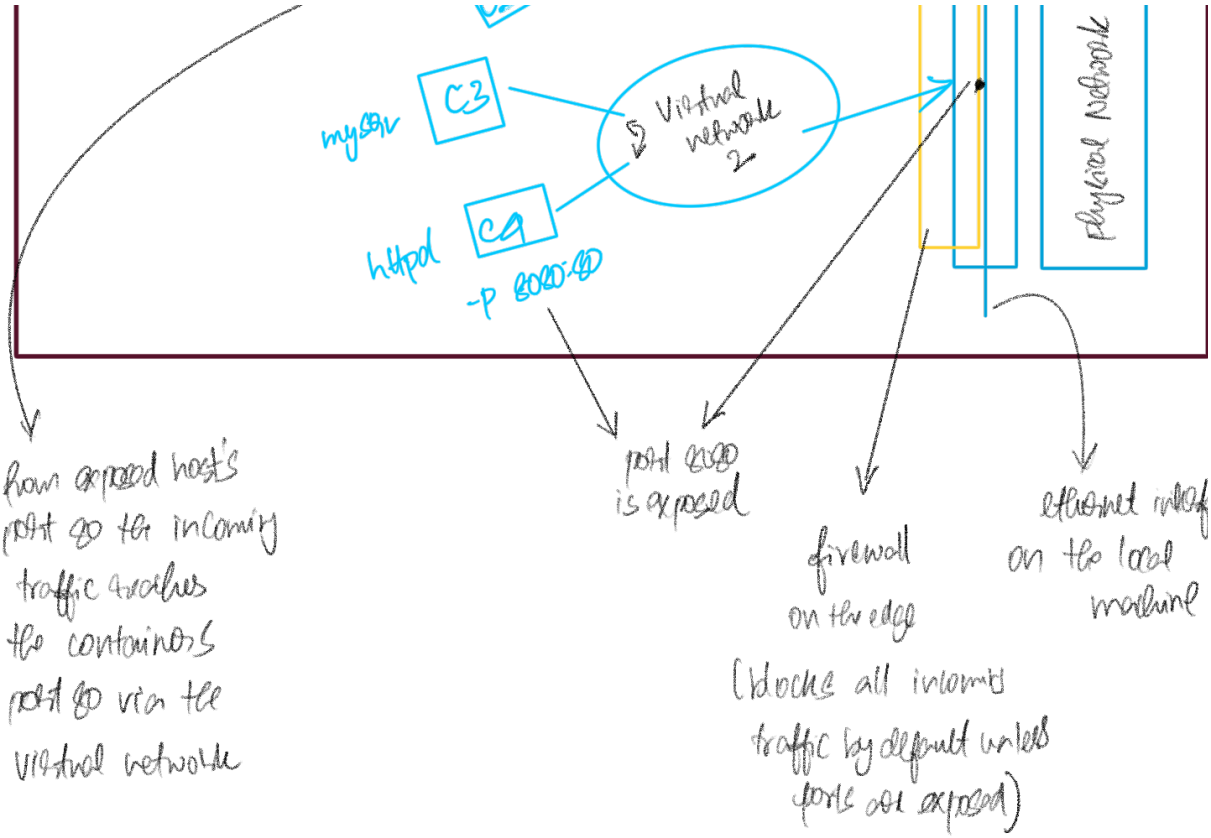
Docker Networks Defaults

- Each container connected to a private virtual network "bridge"
- Each virtual network routes through NAT firewall on host IP
- All containers on a virtual network can talk to each other without -p
- Best practice is to create a new virtual network for each app:
  - network "my_web_app" for mysql and php/apache containers
  - network "my_api" for mongo and nodejs containers

Docker Networks Cont.

- "Batteries Included, But Removable"
  - Defaults work well in many cases, but easy to swap out parts to customize it
- Make new virtual networks
- Attach containers to more then one virtual network (or none)
- Skip virtual networks and use host IP (--net=host)
- Use different Docker network drivers to gain new abilities
- and much more...





=> C1 & C2 on the same virtual pvt network can talk to each other.

=> incoming 8080 traffic reaches apache server (80) in C4 via the virtual network

↳ Node apache & mySQL can freely talk to each other

=> if C1 & C3 had to talk - has to go via published ports in the ethernet interface.

So while thinking abt Virtual Networks in Docker => think about proximity of containers

↳ you'll want node & mongo to talk freely (same VN?)

with each [illegible]

↳ what about hyperbolic maps? (probably not)