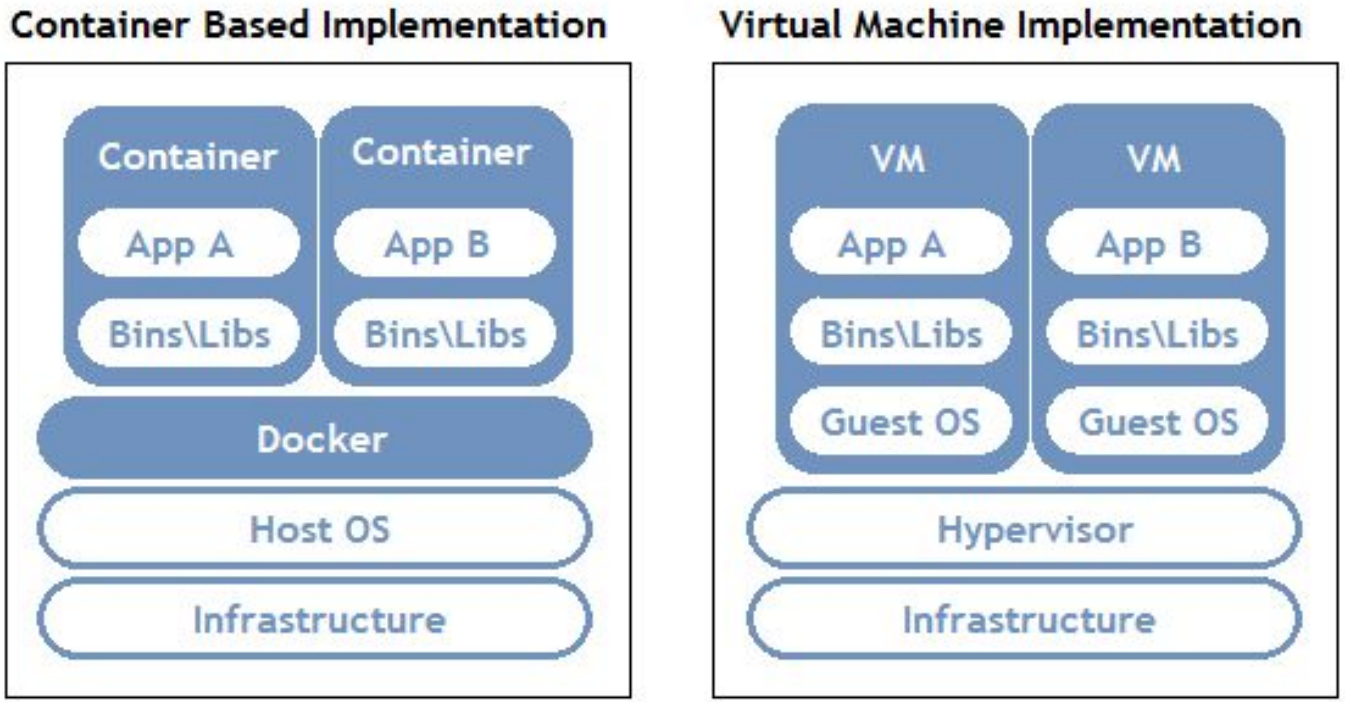
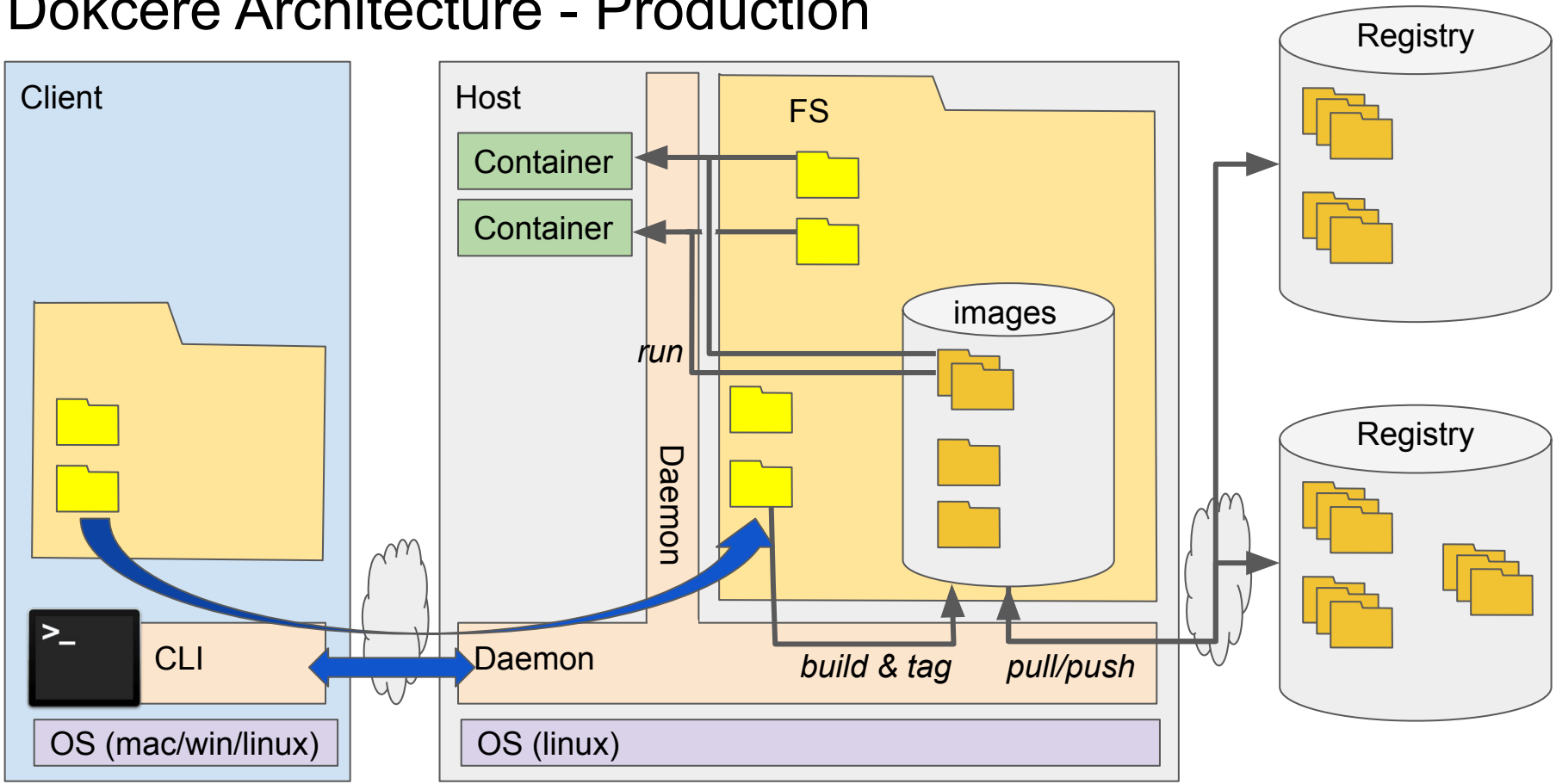
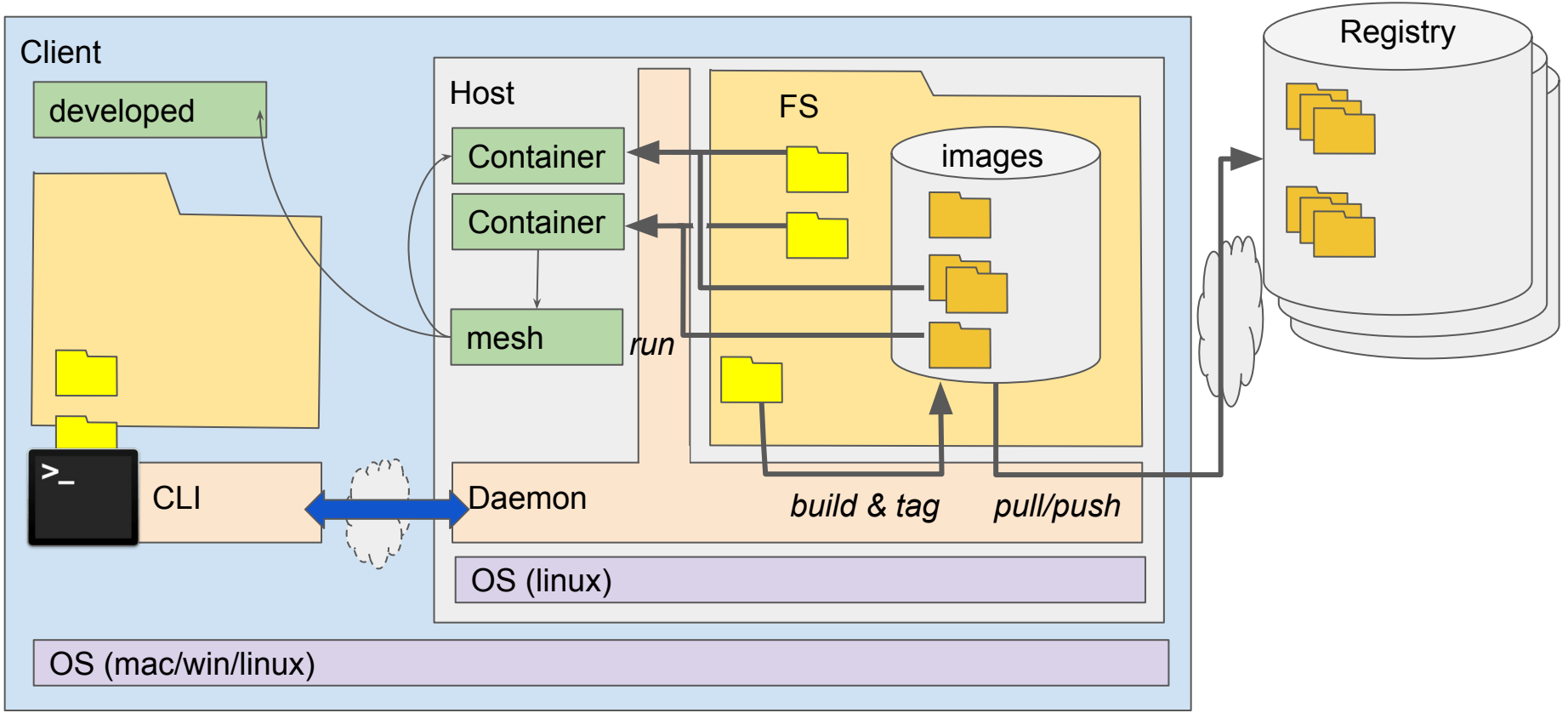
Virtualization vs Containerization



Dokcere Architecture - Production



Dokcere Architecture - Docker Desktop



run a container, named getting-started, bound to port 80

```
docker run -d -p 80:80 --name getting-started docker/getting-started
```

Check what is running - expect the getting-started

```
docker ps
```

Check the logs of the getting-started container:

```
docker logs getting-started
```

Stop the container:

```
docker stop getting-started
```

Check what is running - expect nothing!

```
docker ps
```

Check what containers are found - regardless to running or not

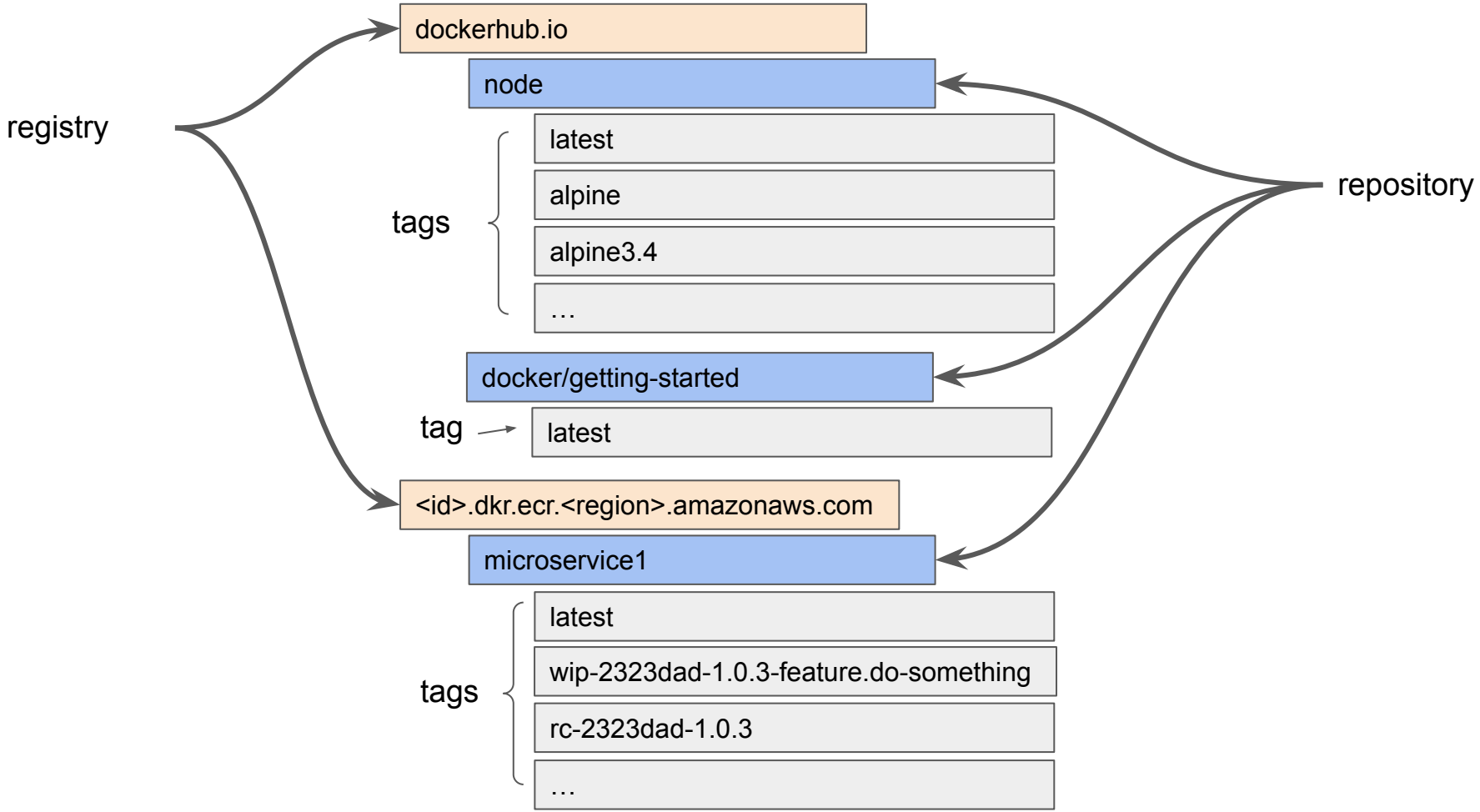
```
docker ps --all
```

Remove the getting started

```
docker rm getting-started
```

Check what is running and what is present - expect nothing

```
docker ps
docker ps -a
```


registry
dockerhub.io
node
tags
latest
alpine
alpine3.4
…
repository
docker/getting-started
tag
latest
<id>.dkr.ecr.<region>.amazonaws.com
microservice1
tags
latest
wip-2323dad-1.0.3-feature.do-something
rc-2323dad-1.0.3
…

Consuming images

https://docs.docker.com/engine/reference/commandline/ps/

https://docs.docker.com/engine/reference/commandline/run/

https://docs.docker.com/engine/reference/commandline/exec/

https://docs.docker.com/engine/reference/commandline/logs/

https://docs.docker.com/engine/reference/commandline/stop/

https://docs.docker.com/engine/reference/commandline/rm/

https://docs.docker.com/engine/reference/commandline/rm/

Customized run

docker run [options] <image-URI> [command + args]

Important switches:

--interactive
--detach
--publish <hostport>:<container port>
--network <network name>
--volume <host path>:<container path>
--env <ENV_NAME>=<VALUE>
--workdir <container path>
--entrypoint <container file path>

--ttl
--rm

--voume <named-volume>
--env-file <host path>

Creating containers - the build command

docker build [options] <dirpath>
https://docs.docker.com/engine/reference/commandline/build/

Common options:

--tag <image-URI>
--file <filepath> ← *use <filepath> instead of <dirpath>/Dockerfile)*
--build-arg <ENV_NAME>=<value>
--rm - remove intermediate layers upon success
--fore-rm - remove intermediate layers, always
--no-cache

Creating containers - Dockerfile - bring things in

Full reference:
https://docs.docker.com/engine/reference/builder/

Must know:
https://docs.docker.com/engine/reference/builder/#from
https://docs.docker.com/engine/reference/builder/#workdir
https://docs.docker.com/engine/reference/builder/#copy
https://docs.docker.com/engine/reference/builder/#run
https://docs.docker.com/engine/reference/builder/#arg
https://docs.docker.com/engine/reference/builder/#env

Creating containers - Dockerfile - what runs?

https://docs.docker.com/engine/reference/builder/#understand-how-cmd-and-entrypoint-interact

TL;DR: together they comprise the shell command that executes the main process:

```
shellCmd = entrypoint.contact(cmd)
```

If the entrypoint has a shebang - it will be “`.shift()`ed”, and that’s what runs.

https://docs.docker.com/engine/reference/builder/#entrypoint
https://docs.docker.com/engine/reference/builder/#cmd