



19 dicembre 2024



Giuliano Latini
«Artigiano d'infrastrutture IT»



Luigi Pandolfino
Azure Meetup Veneto



Andrea Marchi
Azure Meetup Veneto

Ogni docker è bello a
mamma sua! Dopo due lustri
di containerizzazione...



veneto@azuremeetup.it

Il gruppo nasce con l'obiettivo di creare una community per la condivisione di informazioni ed esperienze su **Microsoft Azure** e **365**.

Si rivolge a professionisti del settore IT ma anche ad appassionati di tecnologia, studenti e imprenditori interessati ad approfondire le **soluzioni cloud** e **ibride messe** a disposizione da **Microsoft**.



veneto@azuremeetup.it



Sessioni tecniche in **presenza** ed eventi dedicati al mondo **Microsoft**

23 settembre 2024

Luigi Pandolfino
Coordinatore Azure Meetup Veneto

Guido Imperatore
Microsoft Security MVP SIEM & XDR

Andrea Marchi
Moderatore

La sicurezza del Cloud ibrido con Defender for Cloud

AN MM
MA BG
AM +6

Webinar con esperti di tecnologie **Microsoft**

| Cosa facciamo |



veneto@azuremeetup.it

*Ci fermiamo per la pausa Natalizia,
ma gli eventi riprenderanno nel 2025
con tante sorprese.*

Scrivici se vuoi essere dei nostri!!

...e buone Feste!



| Call4Speaker |



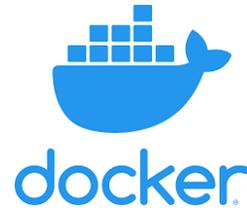
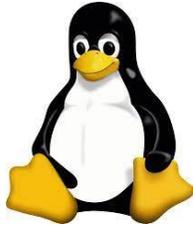
<proximo>



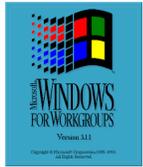
Work Path

IT OPERATIONS (ITOps)

Enter your sub headline here



Windows PowerShell



Community
Connect

Dev > Marche



giulianolatini



giulianolatini



giulianolatini

| About me |

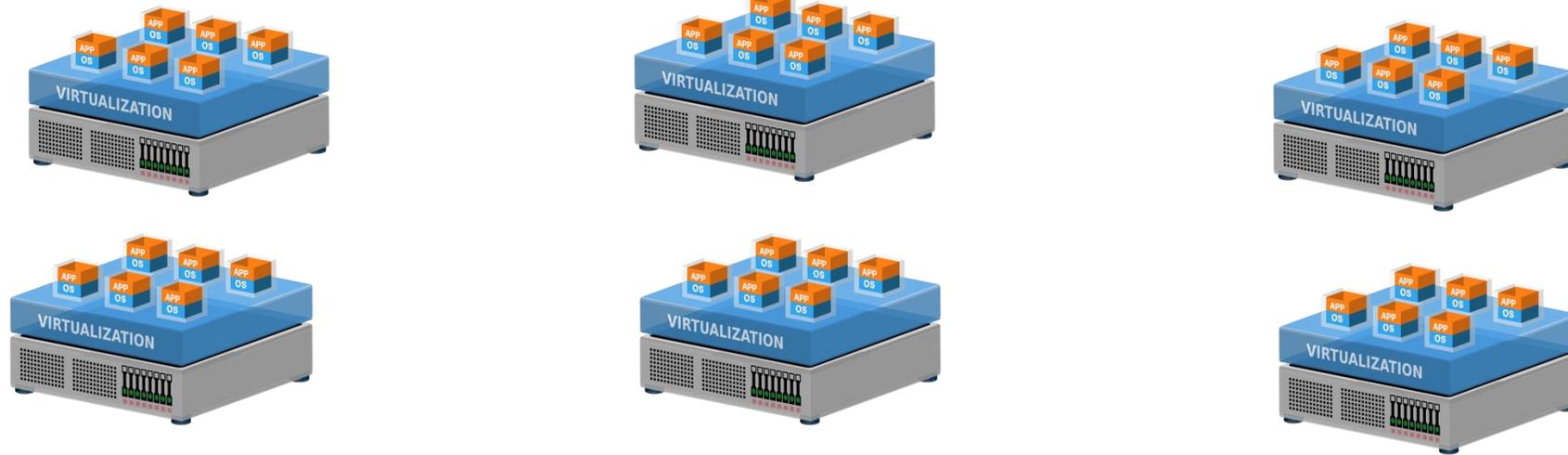


veneto@azuremeetup.it

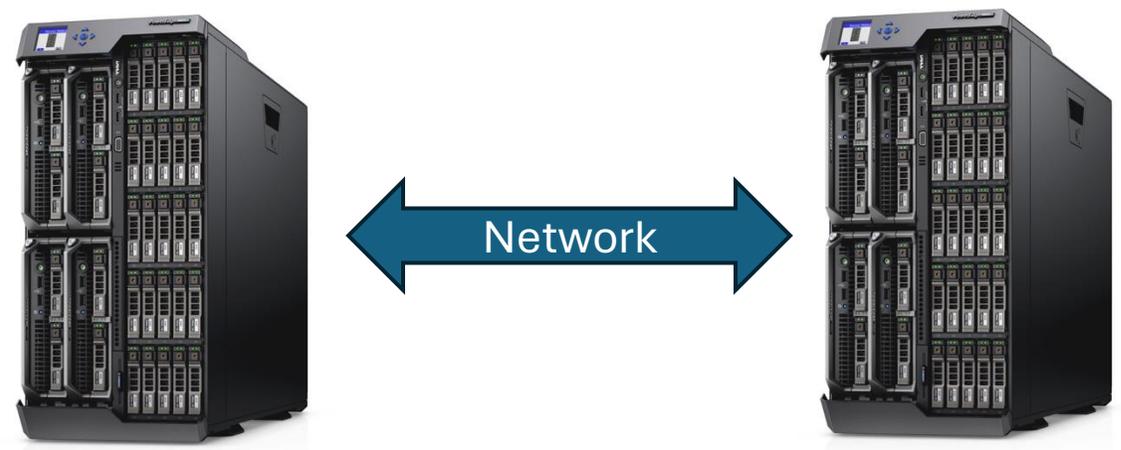
A long time ago
in a galaxy far,
far away....



Heavy Metal before 2000

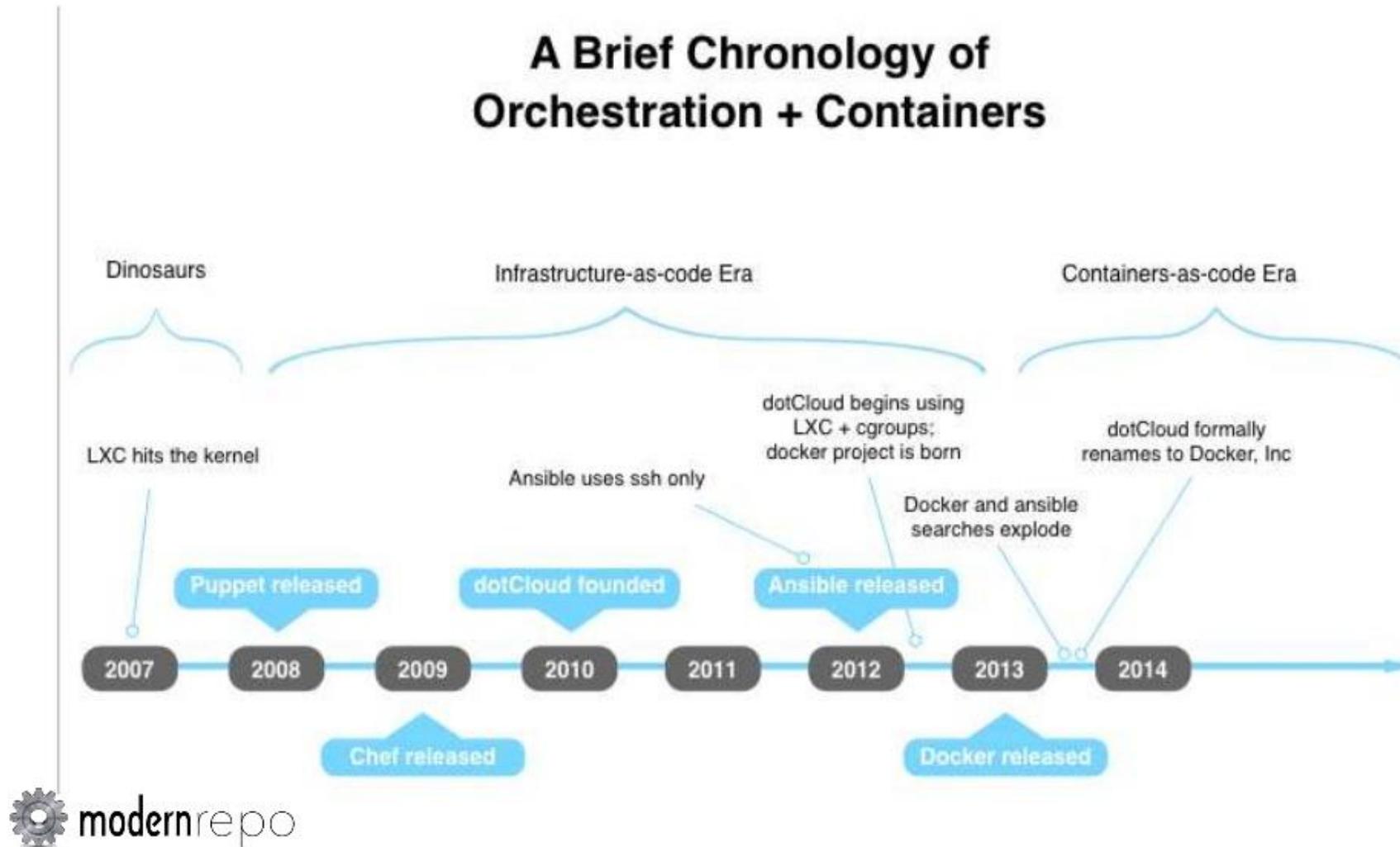


Virtual



After 2000 virtualizations' arrive and the consolidation wave began...

A Brief Chronology of Orchestration + Containers

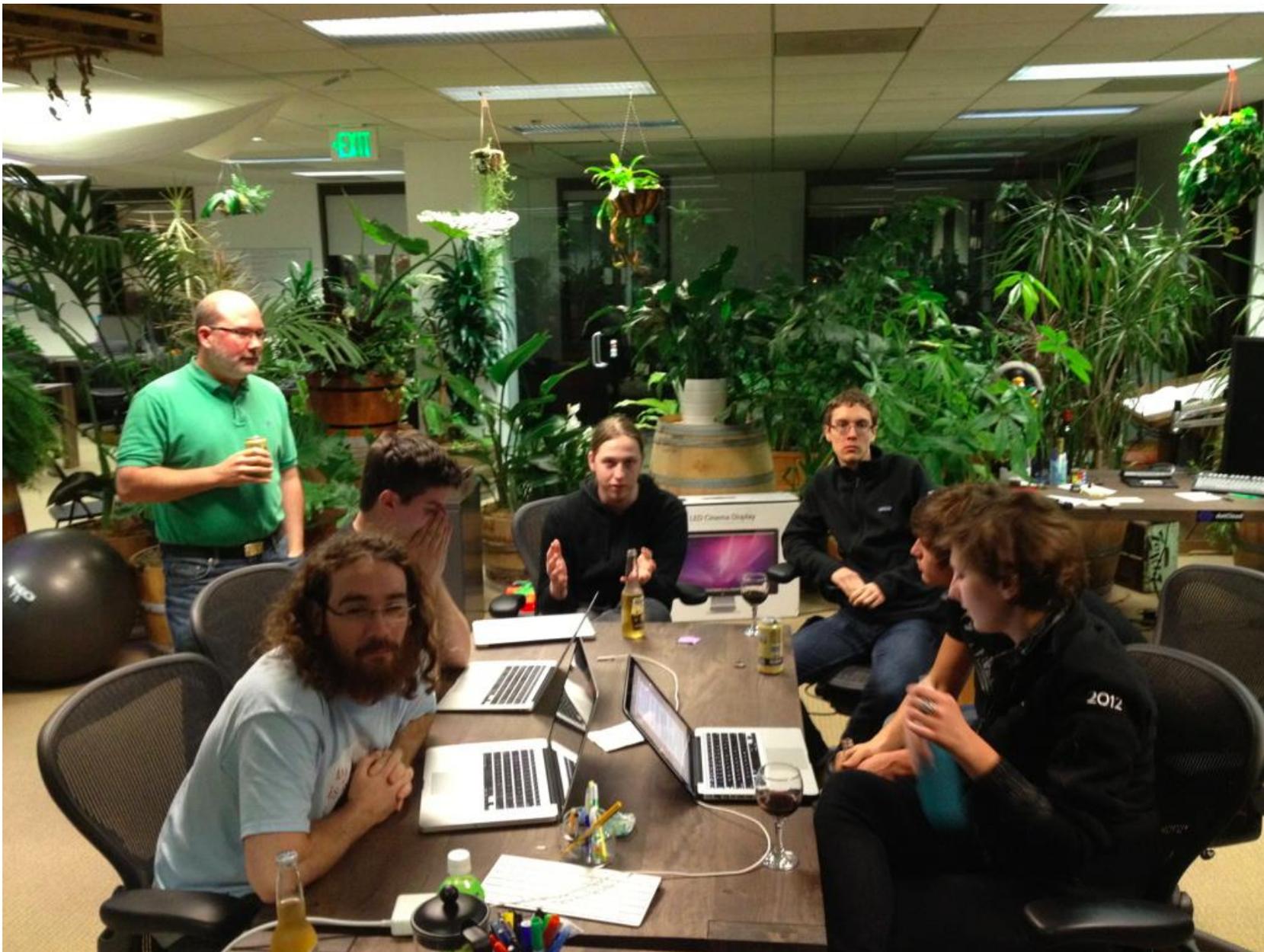


... but the VM deployment was too slow; the IT Ops, pushed from Agile methodologies, need a new paradigm.



veneto@azuremeetup.it

Meanwhile, in an
obscure
application cloud
provider...



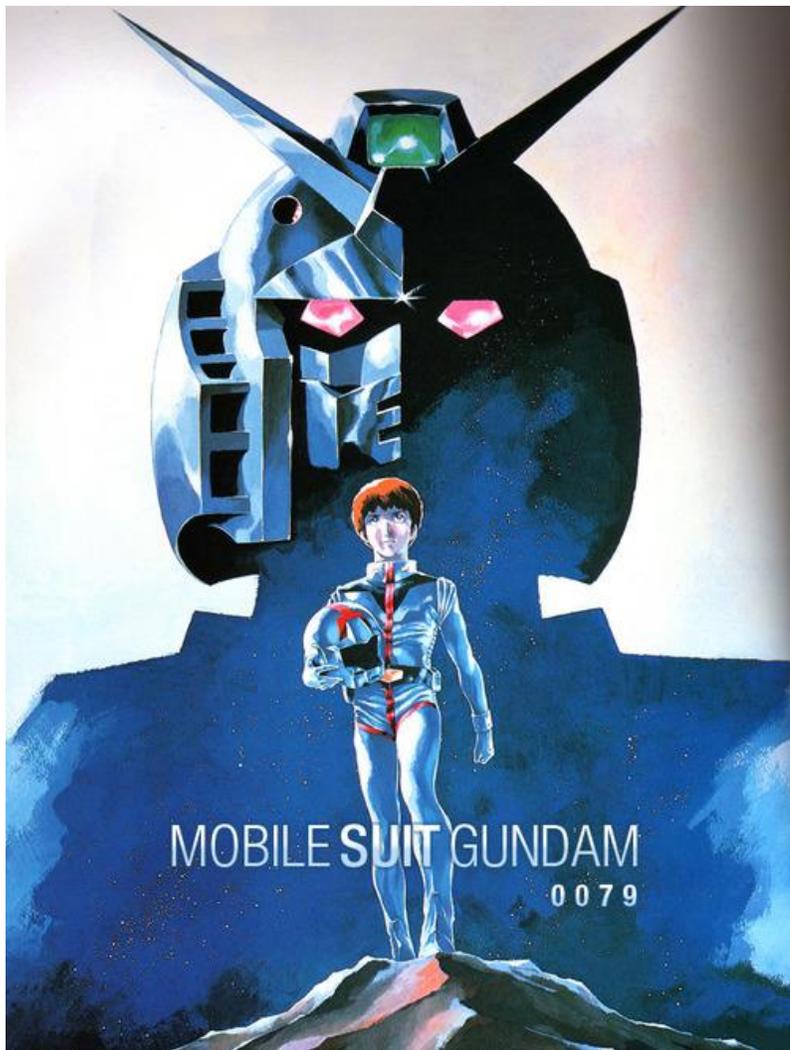
The dotCloud office before Docker's explosion. (Credit: [5 Years at Docker @KenCochrane.com](#))



The dotCloud office at Founder's Den, early 2011. (Credit: SFGate)

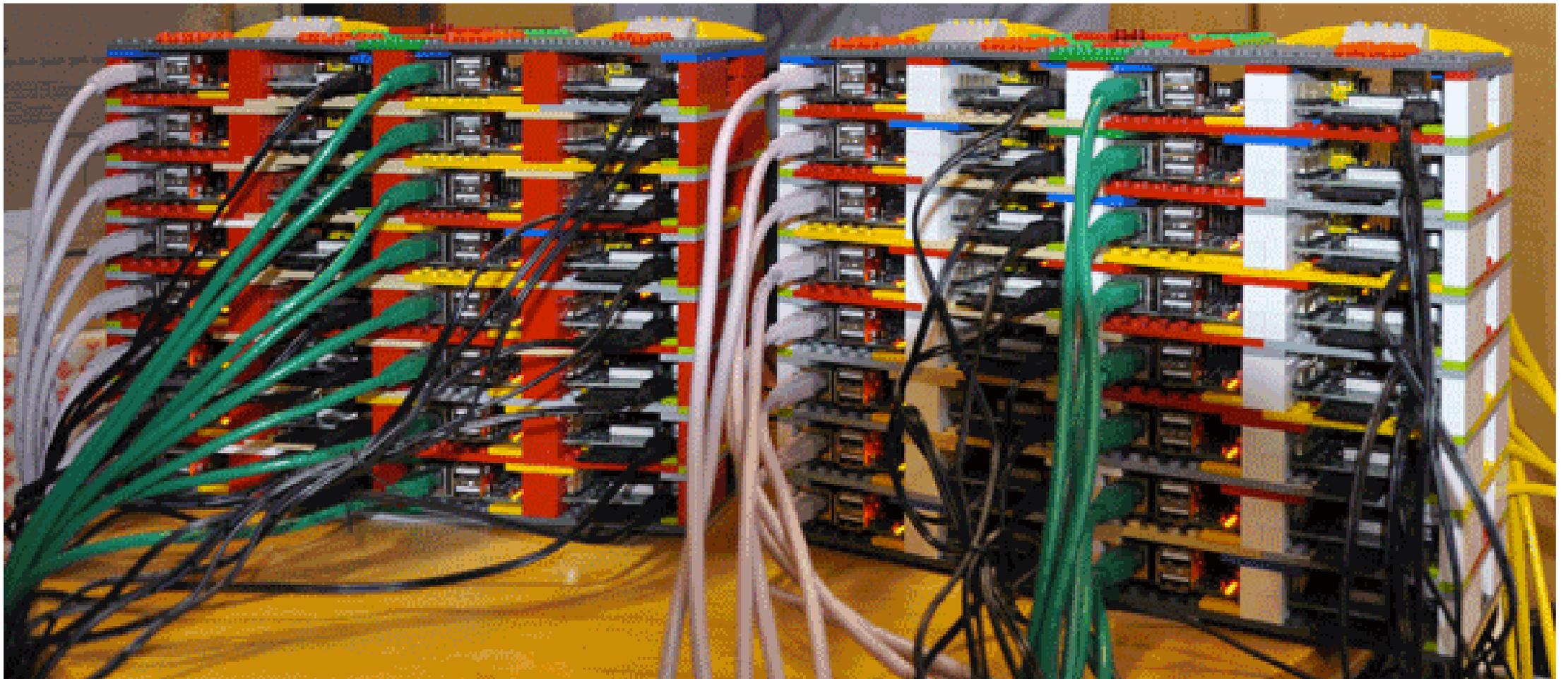


*The future of Linux Containers (by [dotcloudtv@Youtube](https://www.youtube.com/channel/UCdotcloudtv))- 22 mar 2013
At PyCon Solomon Hykes shows docker to the public for the first time*



*la modularità
Vince Sempre*

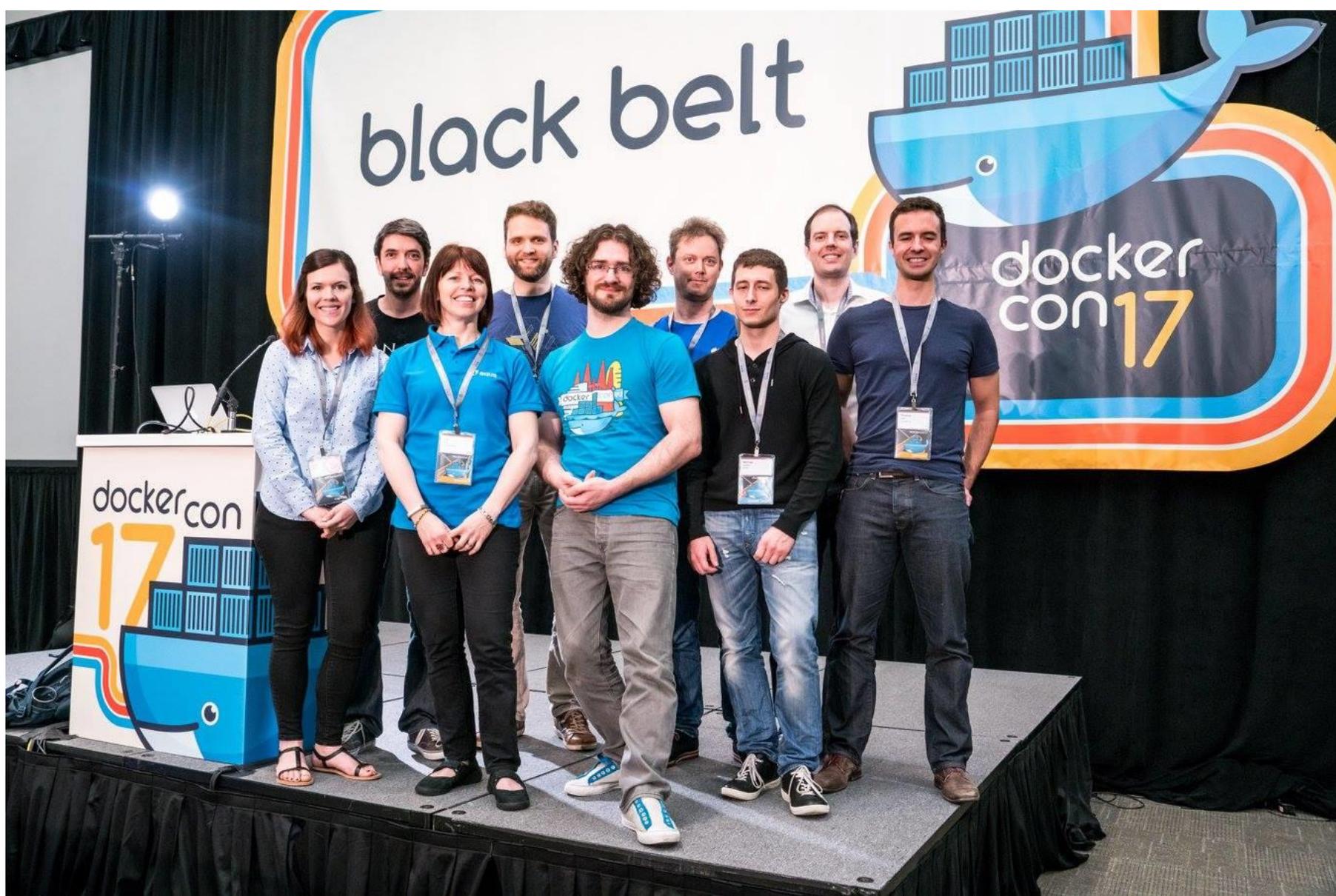




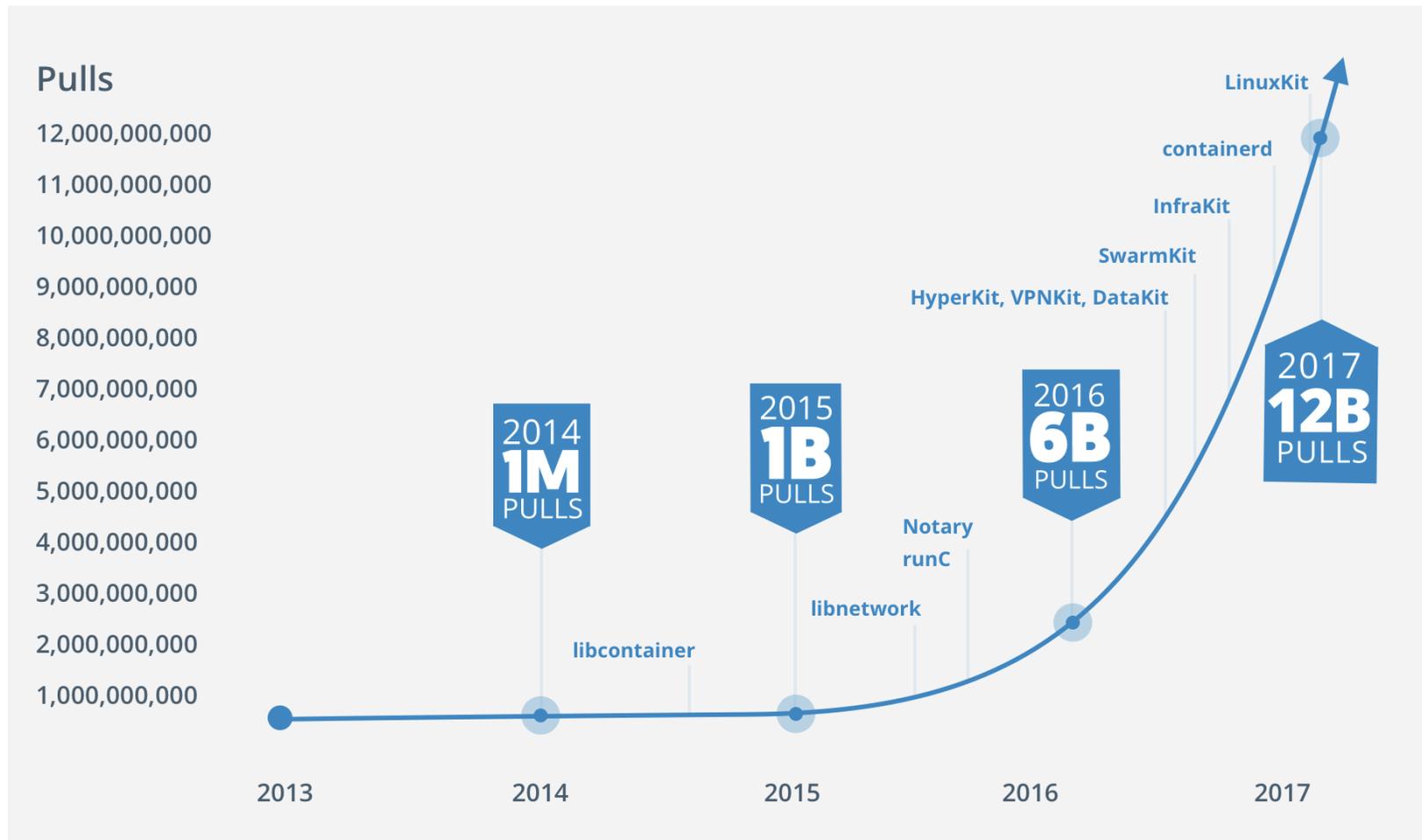
Progetto interno a Google.

Datacenter progettato dalle prime classi (elementari) della scuola aziendale. I componenti base sono: 3 scatole di mattoncini lego; 32 Raspberry Pi B; 2 switch 24 porte, cavi.

L'infrastruttura è un private cloud per sostenere i 48 blog Wordpress degli alunni che l'hanno costruita.



Black Belt track speakers from DockerCon 2017 in Austin.



Docker pulls are through the roof!

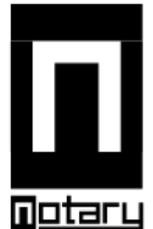


veneto@azuremeetup.it

The Good, the Bad and the Ugly

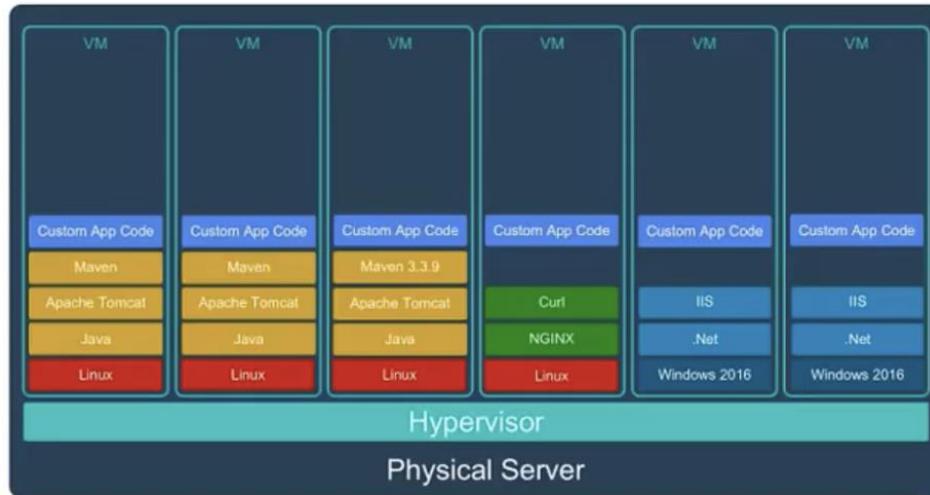
Docker Features

- Multiplatforma
- Sviluppo, testing, frontend infrastrutturale, Infrastruttura per usare DockerApp
- Securable Supply Chain
- Integrato con qualsiasi prodotto DevOps
- Integrato nei principali Cloud Pubblici
- Infrastruttura per lo sviluppo integrata con VSCode
- SoftwareBOM, Notary, Synk



Our Current State: Virtual Machines

- One app per VM
- Separate OS copy
- Separate app framework
- "Black box" to hypervisor
- App is tracked elsewhere



Consolidamento ed isolamento in salsa VM.

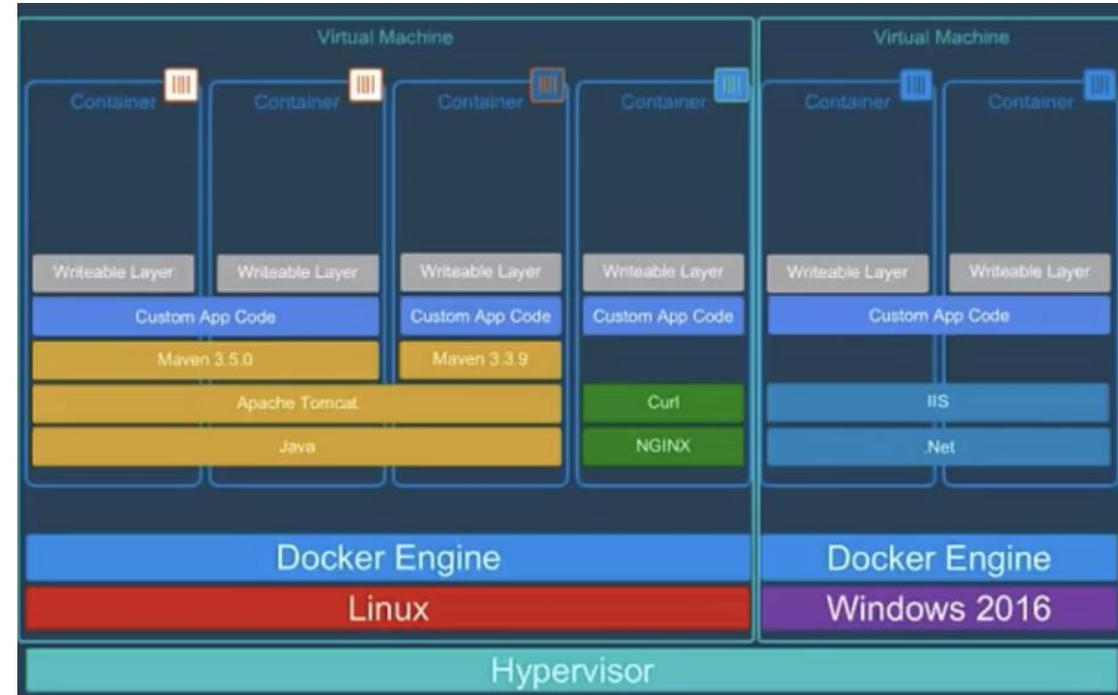
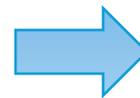
Hypervisor di tipo I

← L'HW è condiviso da più VM ciascuna con il proprio HW virtuale, SO, Stack Applicativo

Confronto Docker vs VMs.

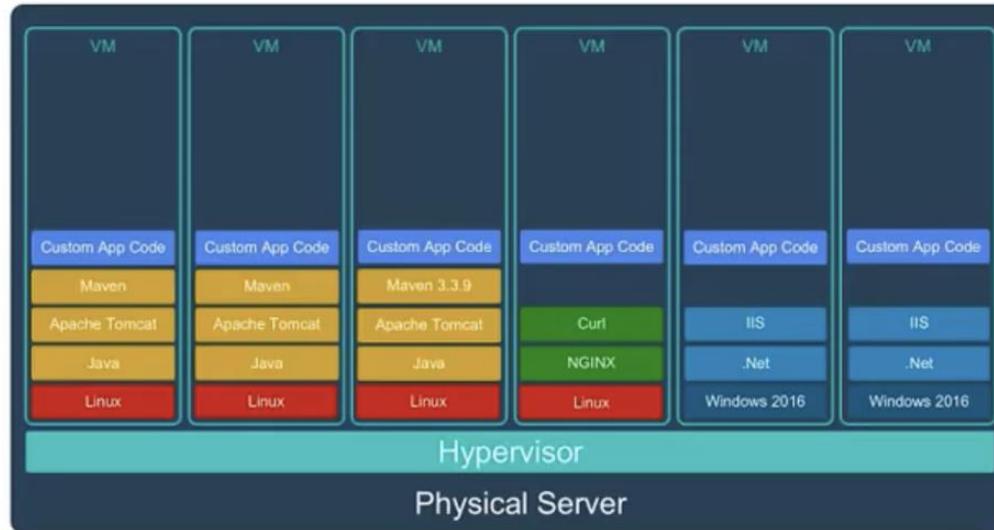
Hypervisor di tipo II (OS)

l'OS è condiviso tra più stack applicativi isolati tra di loro. l'Engine consente di estendere il networking dei container verso altri nodi docker



Our Current State: Virtual Machines

- One app per VM
- Separate OS copy
- Separate app framework
- "Black box" to hypervisor
- App is tracked elsewhere



Consolidamento ed isolamento in salsa VM.

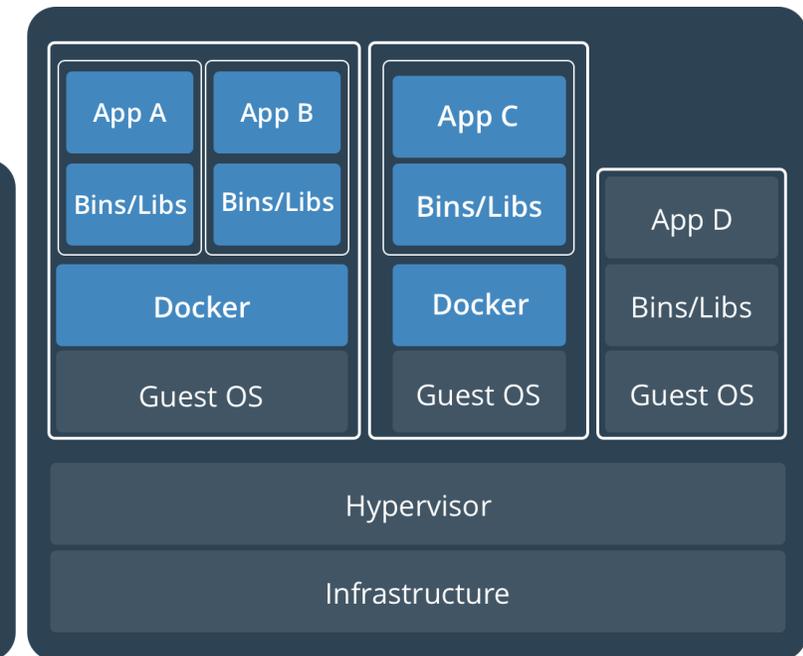
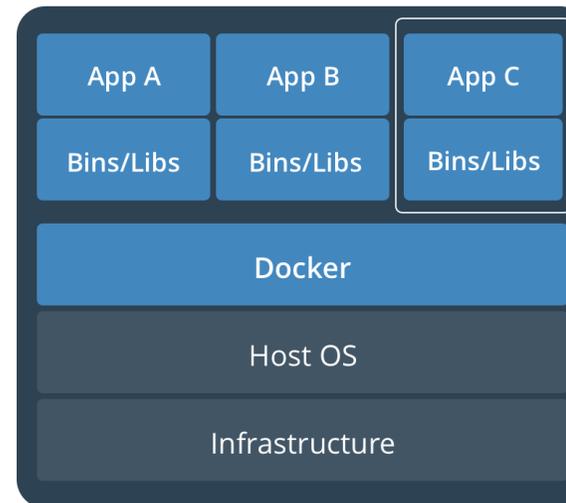
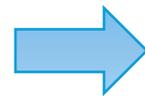
Hypervisor di tipo I

← L'HW è condiviso da più VM ciascuna con il proprio HW virtuale, SO, Stack Applicativo

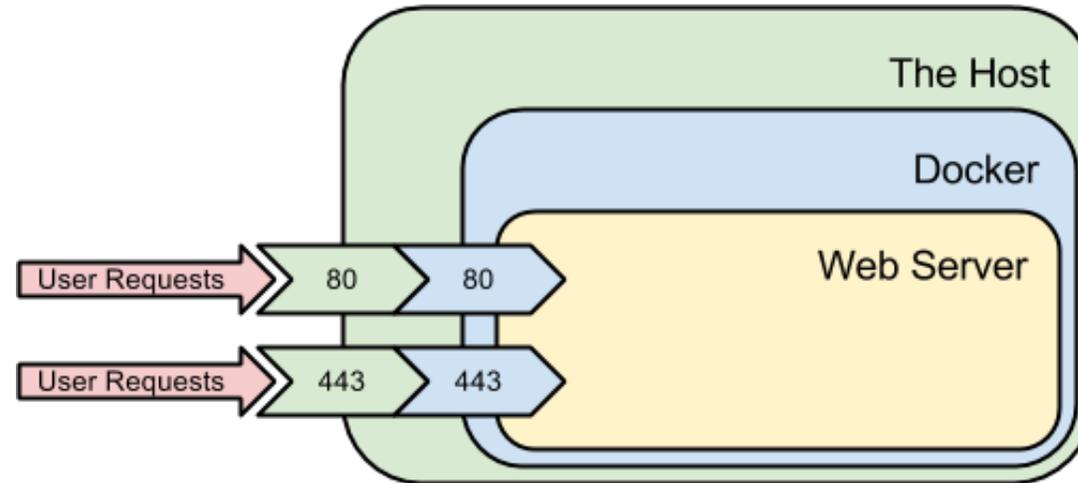
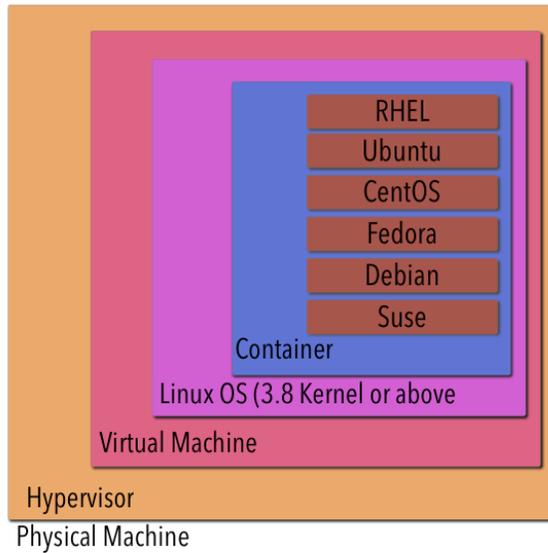
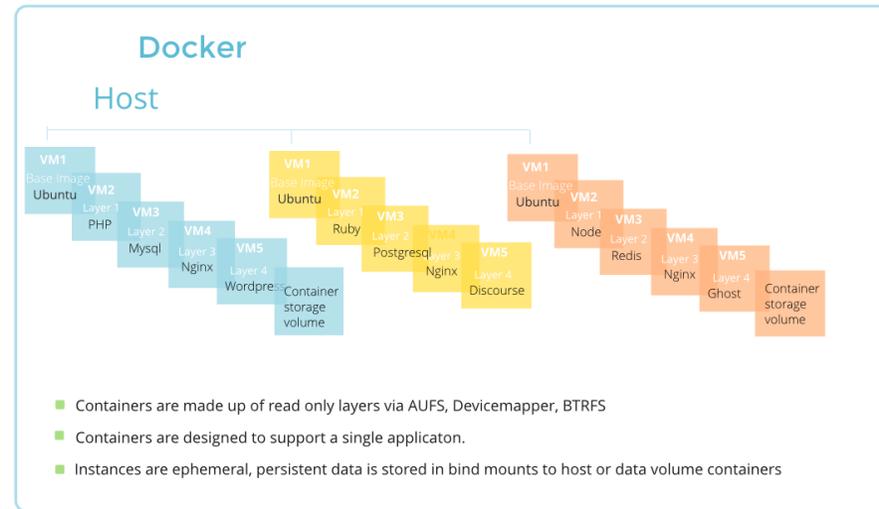
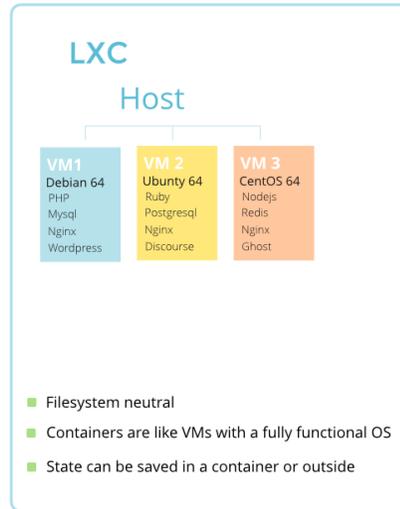
Docker on HW vs Virtual Env.

Hypervisor di tipo II (OS)

l'OS è condiviso tra più stack applicativi isolati tra di loro.



Key differences between LXC and Docker

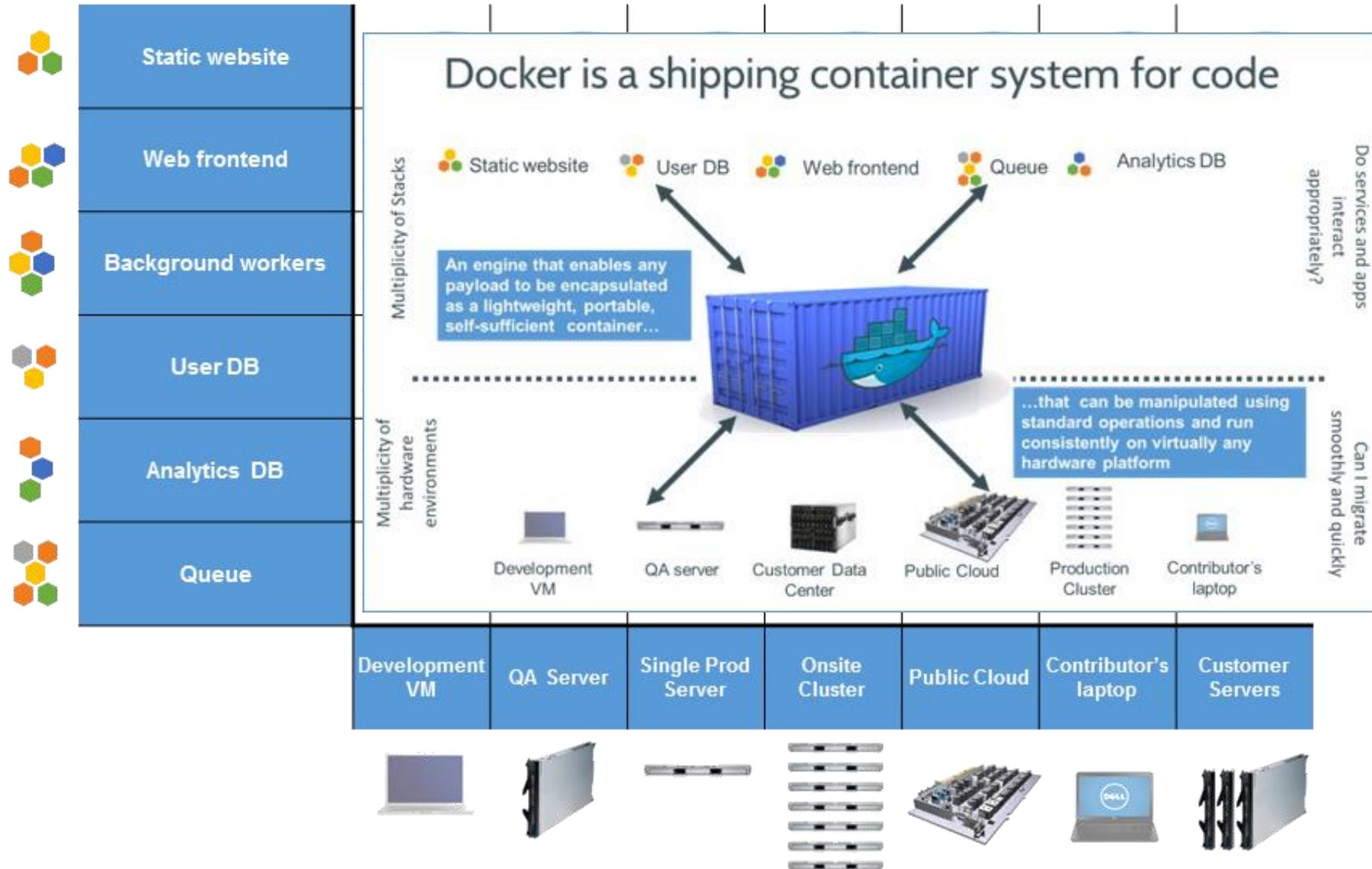


Architectural Schema

Matrice d'usabilità

	Static website							
	Web frontend							
	Background workers							
	User DB							
	Analytics DB							
	Queue							
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
								

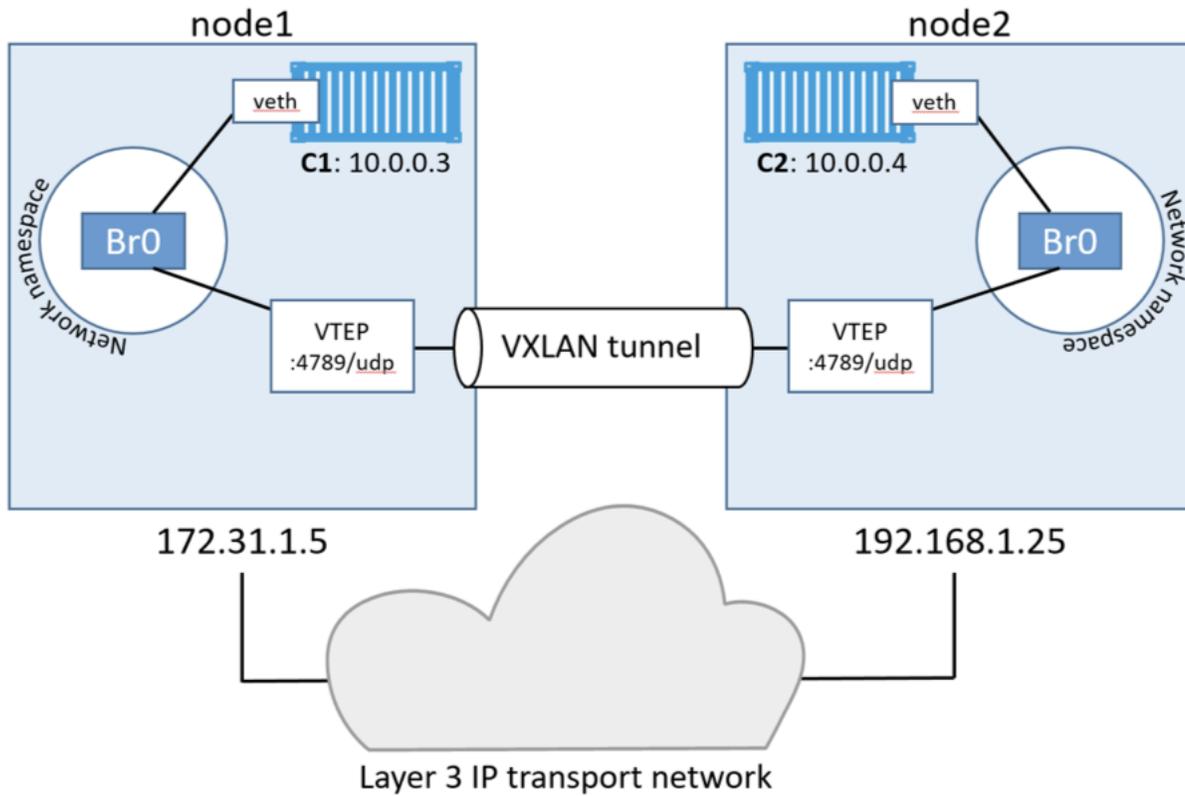
Matrice d'usabilità





veneto@azuremeetup.it

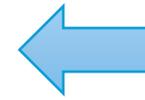
Grey's Anatomy *VS* Doctor House



Networking internodo

Livello fisico

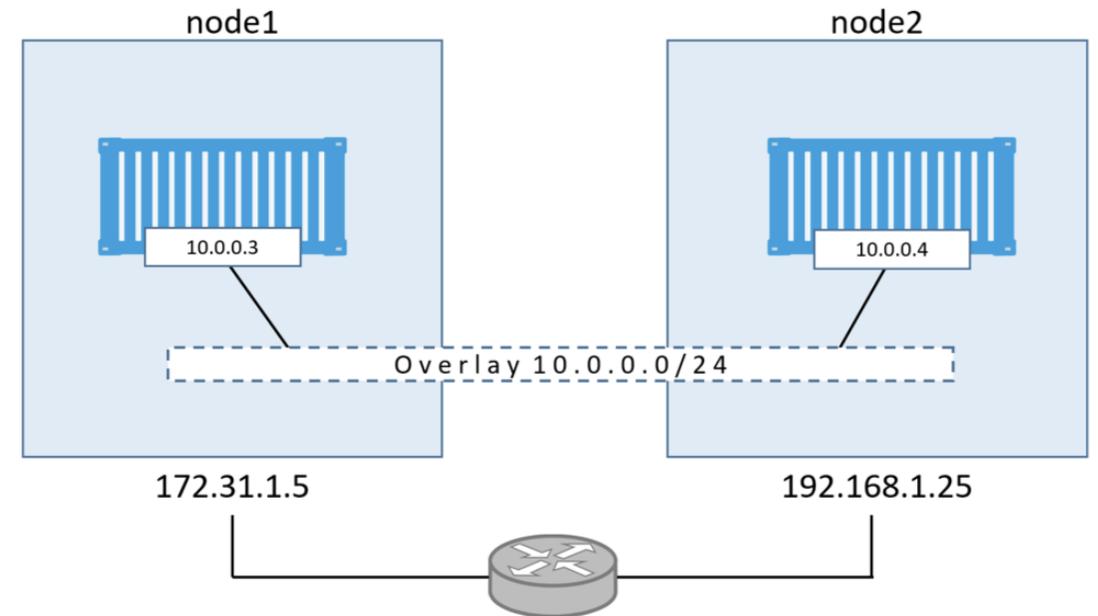
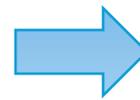
Le Software Define Network gestite dalla docker engine sono estendibili tra nodi attraverso un tunnel VXLAN tra nodi separati da routing.

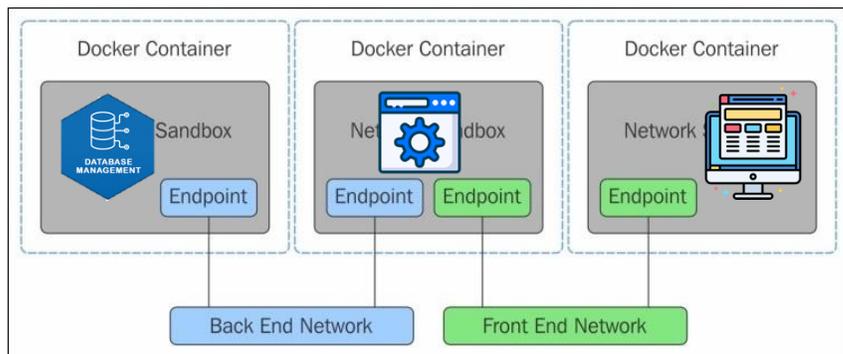
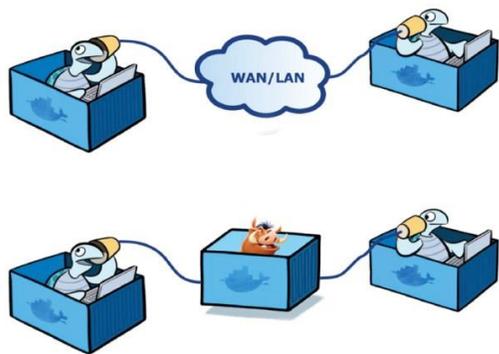


Networking internodo

Livello logico

Due o piú container sulla stessa net SDN comunicano attraverso un canale broadcast esteso con un tunnel crypted tra i nodi connessi.



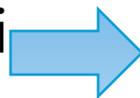


Bridged vs Overlay

Default SDN

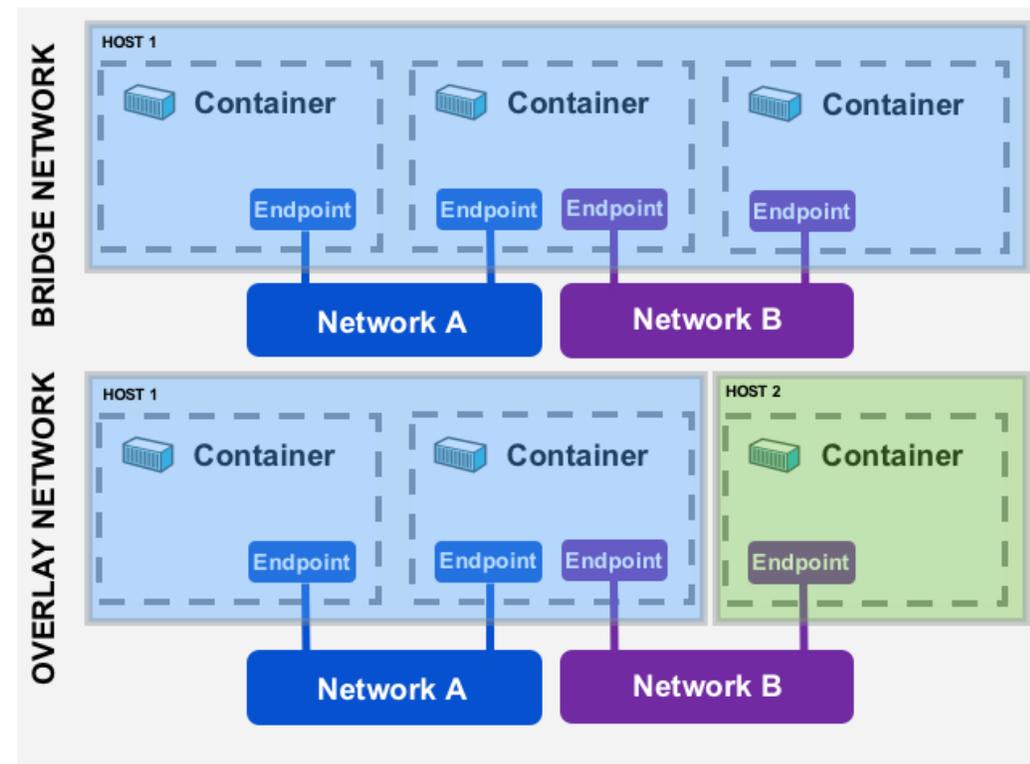
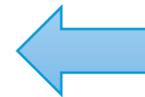
Il Bridge driver definisce all'avvio dell'engine una network per connettere i container nel nodo.

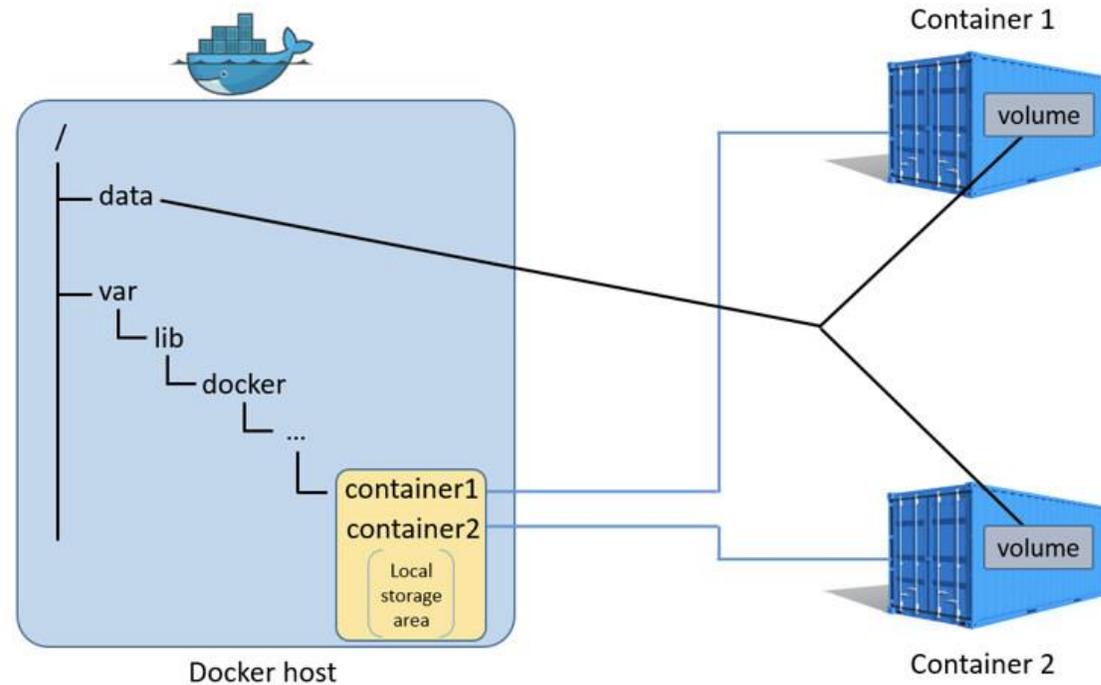
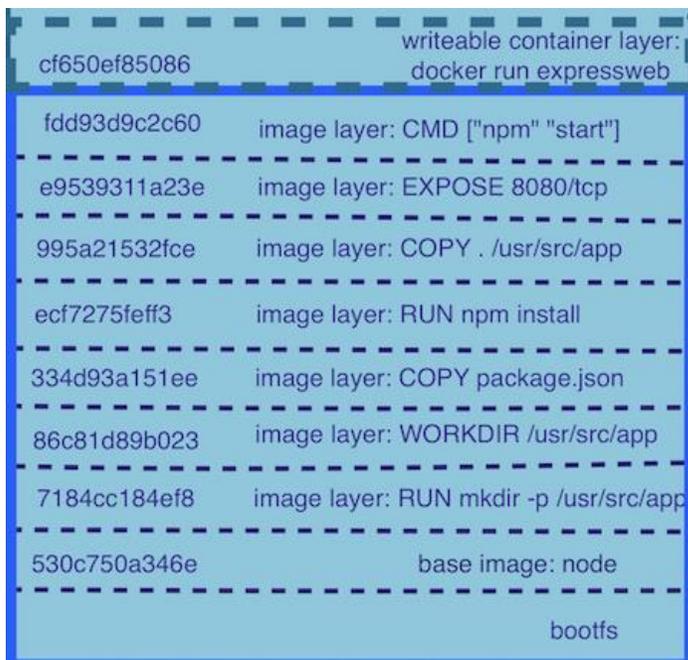
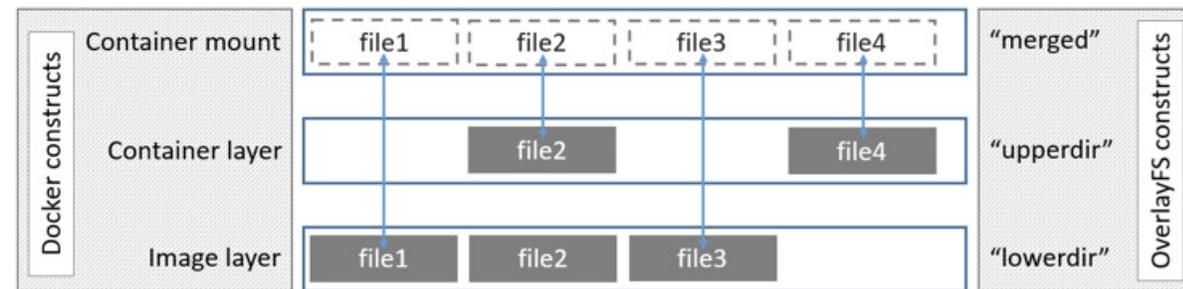
L'Overlay driver definisce una rete estesa multinodo sopra le SDN definite localmente. È alla base del networking nel Docker Swarm.

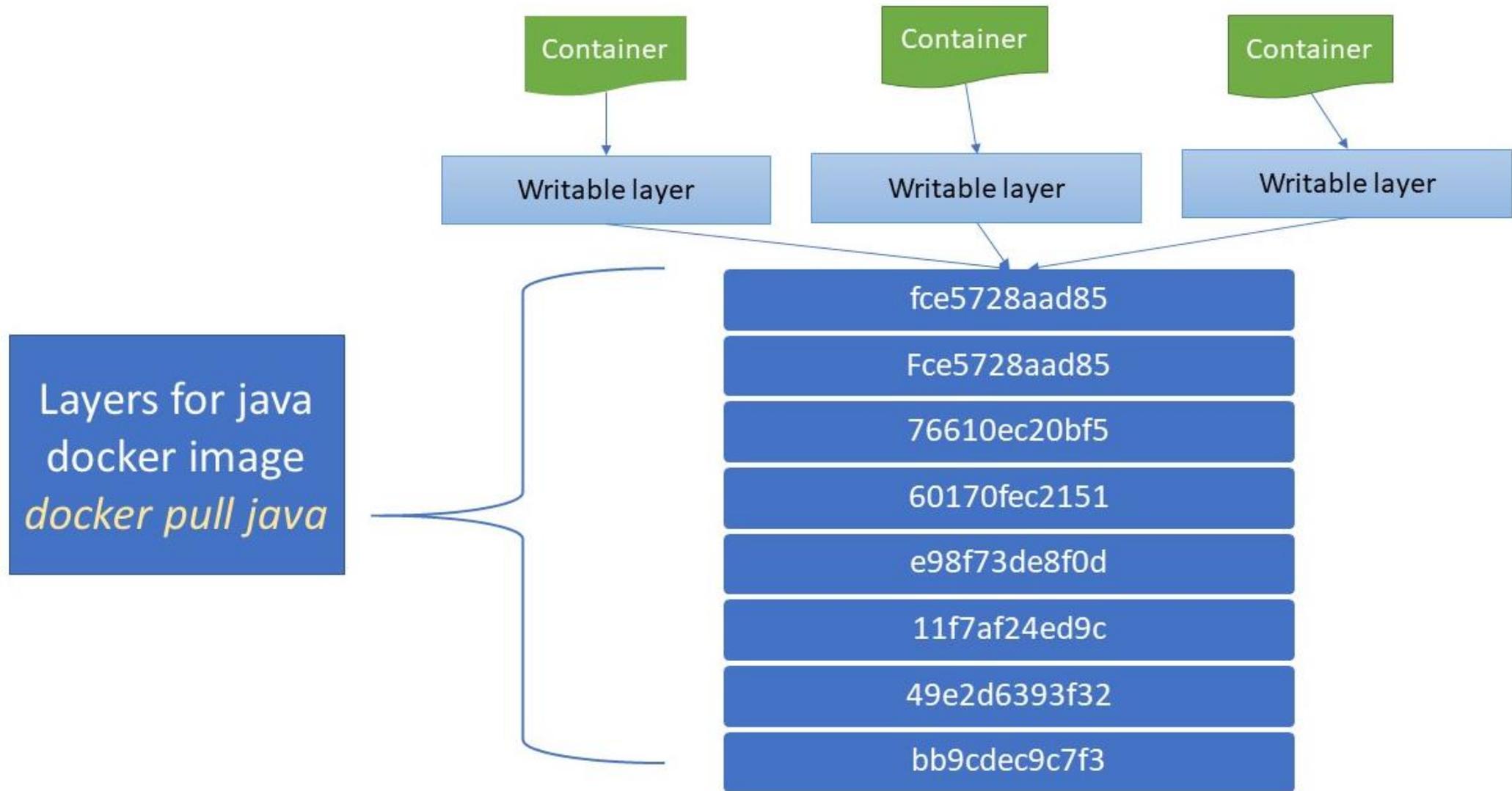


Architettura di Networking Architettura 3-Tier

Separazione Presentation, Business Logic, DBMS implementata tramite SDN docker tra containers







Storage optimization by design

```
docker-user@ubuntu1510:~$
```

```
docker-user@ubuntu1510-2:~$
```

```
[demo] 0:docker-machine*
```

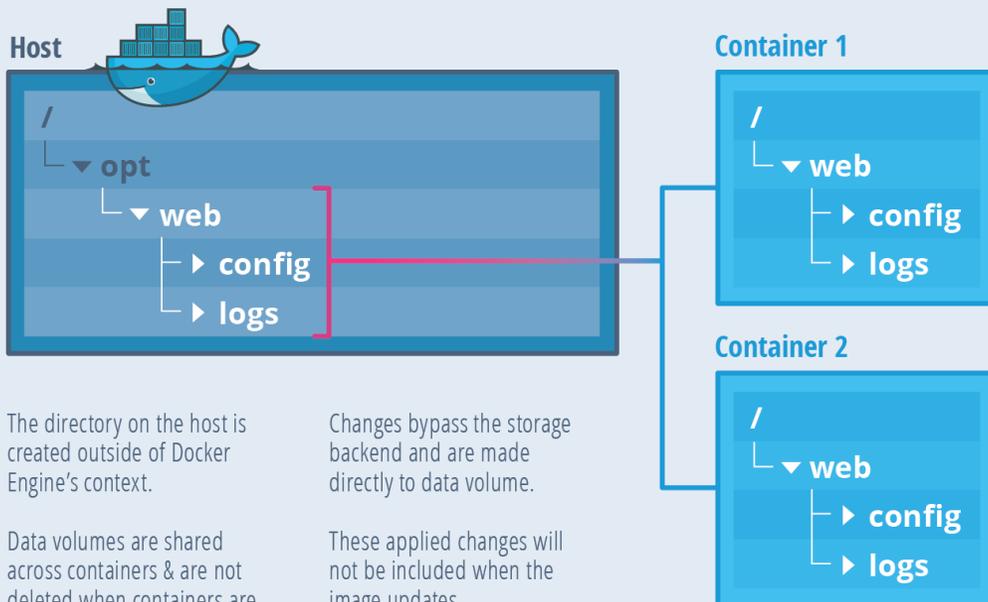
```
"macbook.local" 15:33 11-Mar-16
```

Storage optimization by design

GESTIONE DELLO STORAGE

La persistenza in container tramite la creazione di volumi è preferibile alla persistenza su Host perché i *volume plugins* permettono l'uso di SAN o filesystem condivisi.

Host-Based Persistence Shared Among Containers



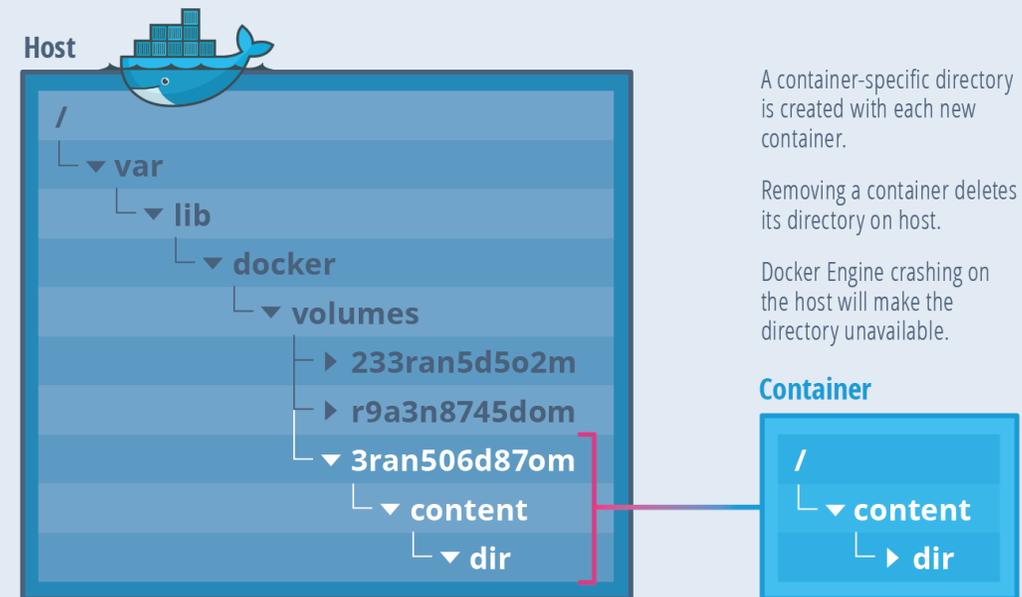
The directory on the host is created outside of Docker Engine's context.

Data volumes are shared across containers & are not deleted when containers are removed.

Changes bypass the storage backend and are made directly to data volume.

These applied changes will not be included when the image updates.

Host-Based Persistence Per Container



A container-specific directory is created with each new container.

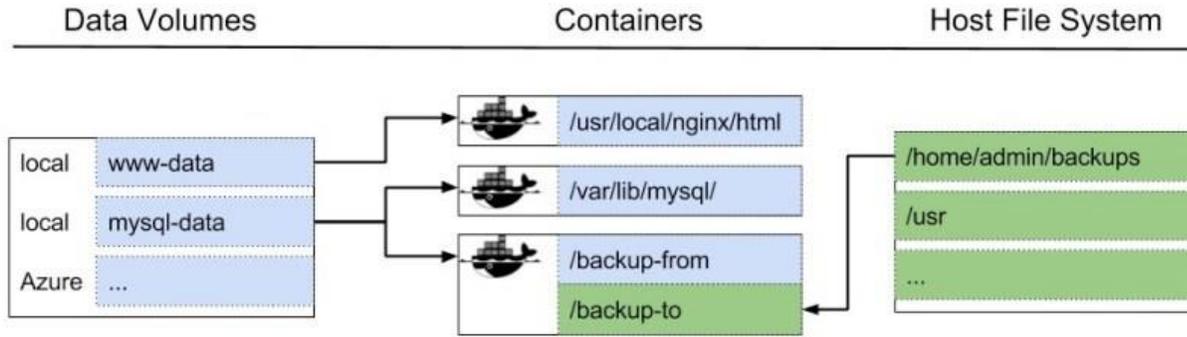
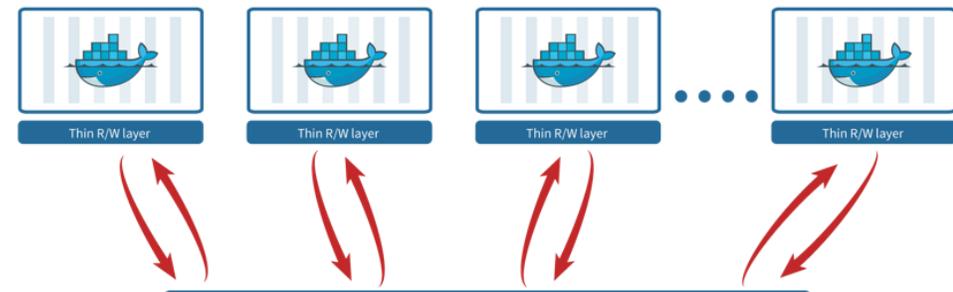
Removing a container deletes its directory on host.

Docker Engine crashing on the host will make the directory unavailable.

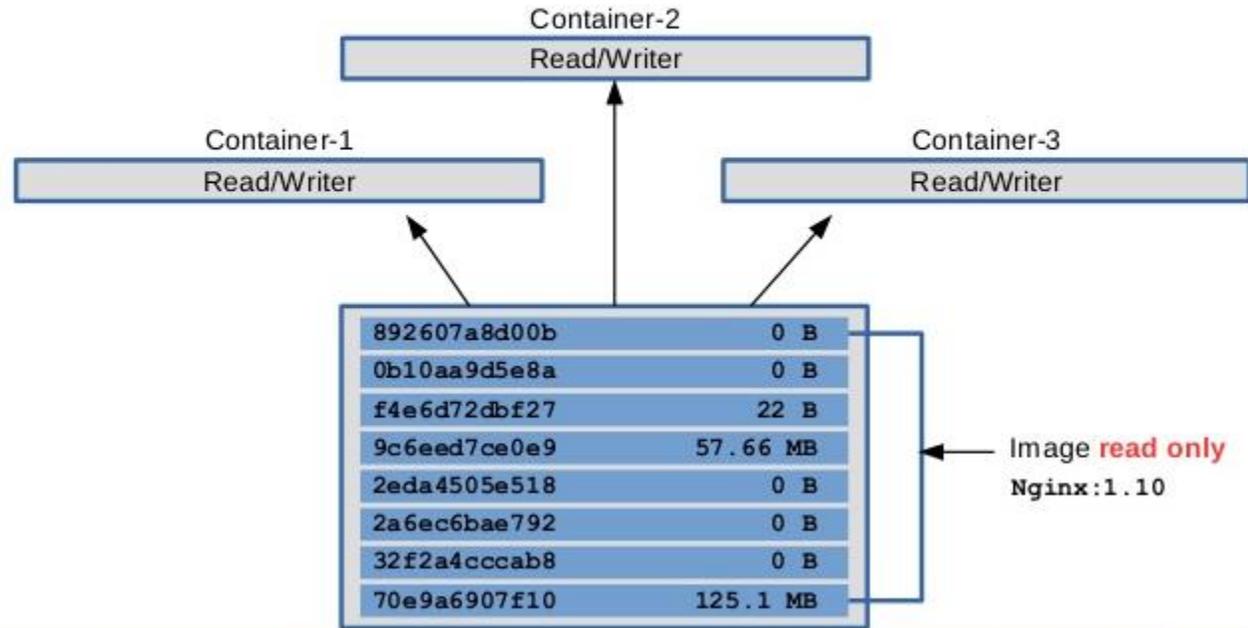
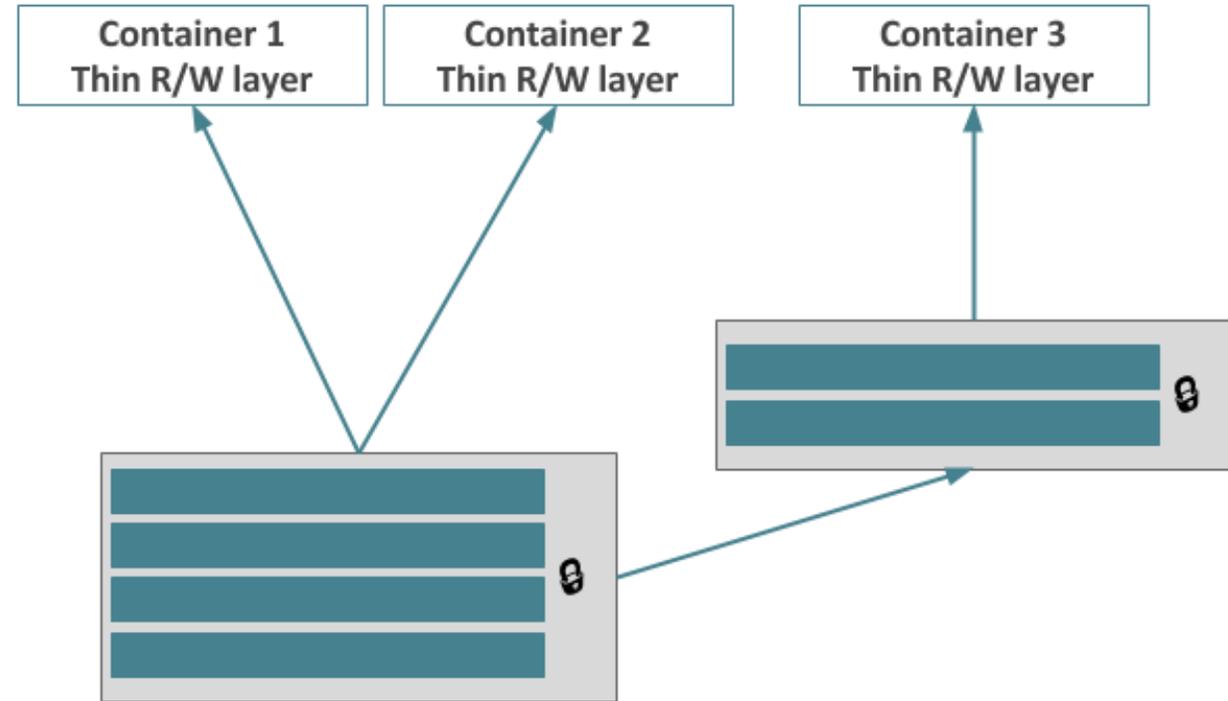
Container



TECNOLOGIA COW

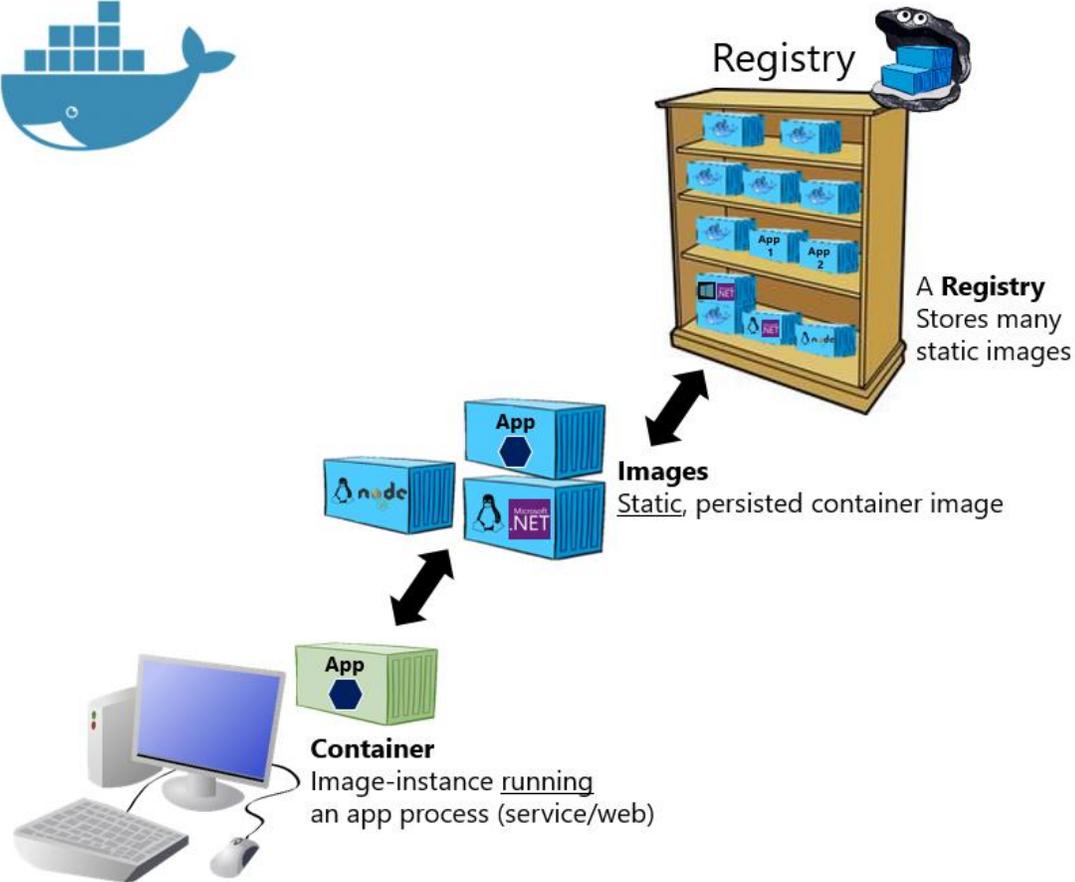


copy-on-write strategy



CONCETTI E GESTIONE DEL REGISTRY

Basic taxonomy in Docker



Hosted Docker Registry

Docker Trusted Registry on-prem.

On-premises
(‘n’ private organizations)

Docker Hub Registry

Docker Trusted Registry on-cloud

Azure Container Registry

AWS Container Registry

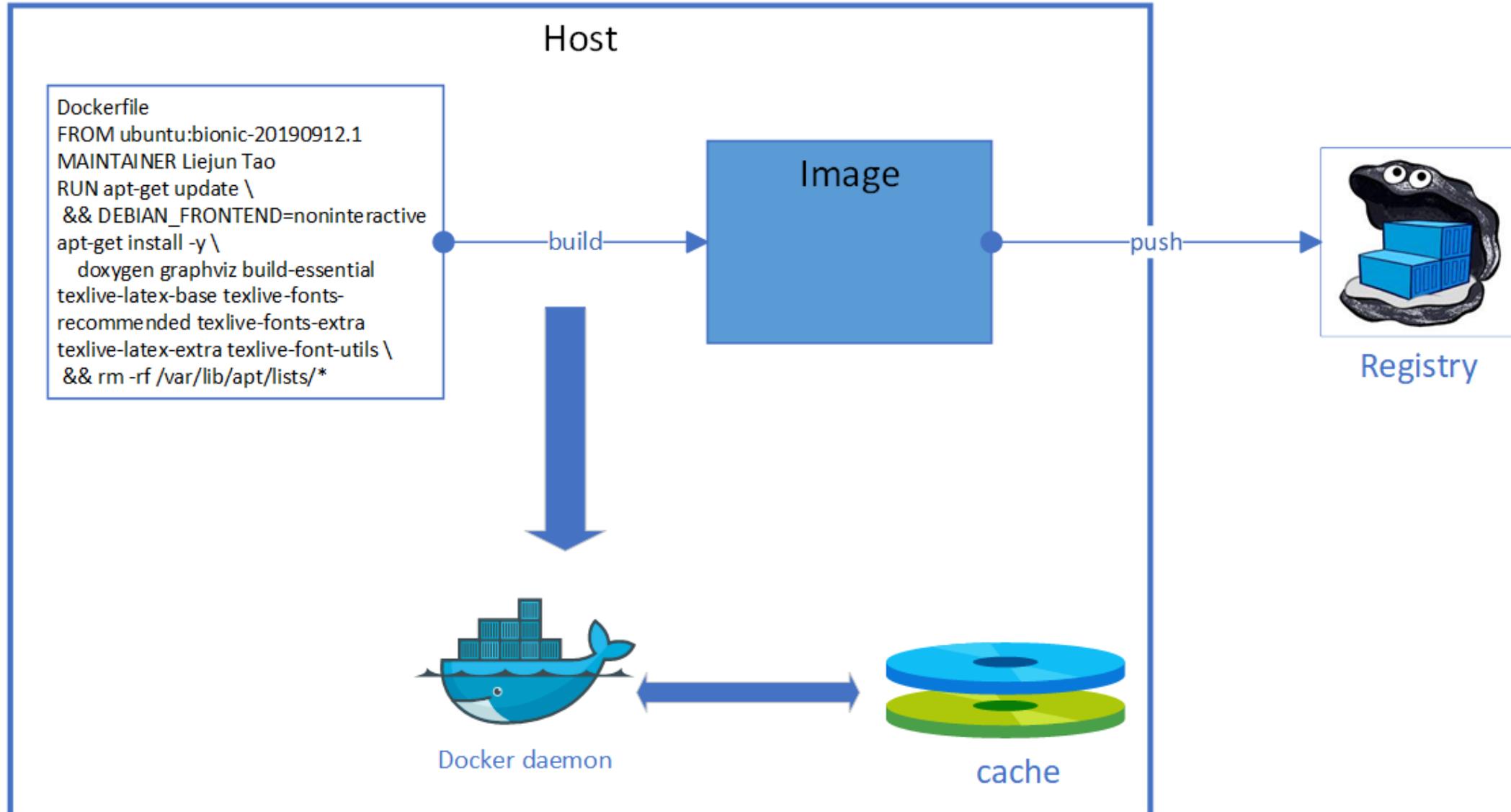
Google Container Registry

Quay Registry

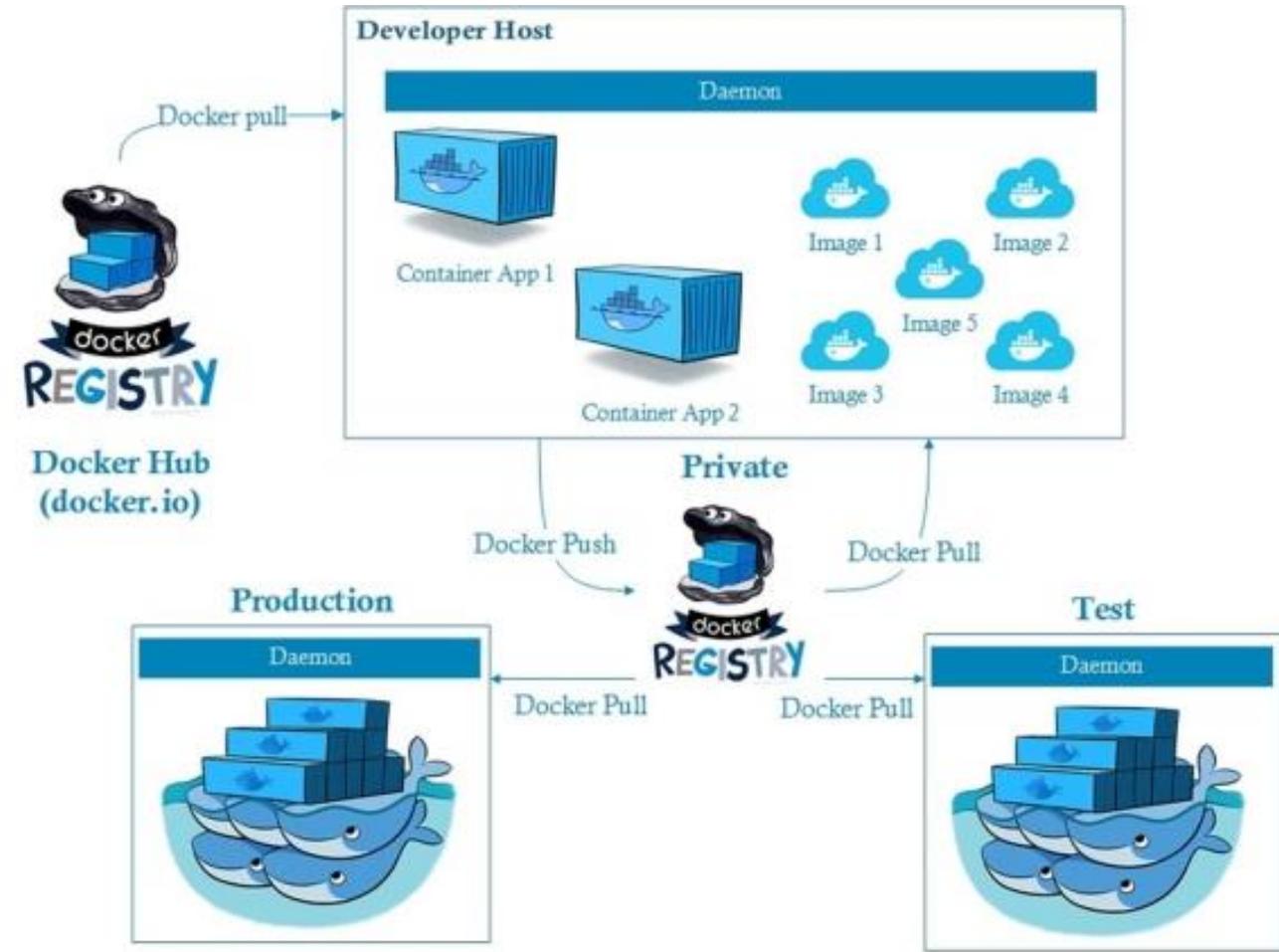
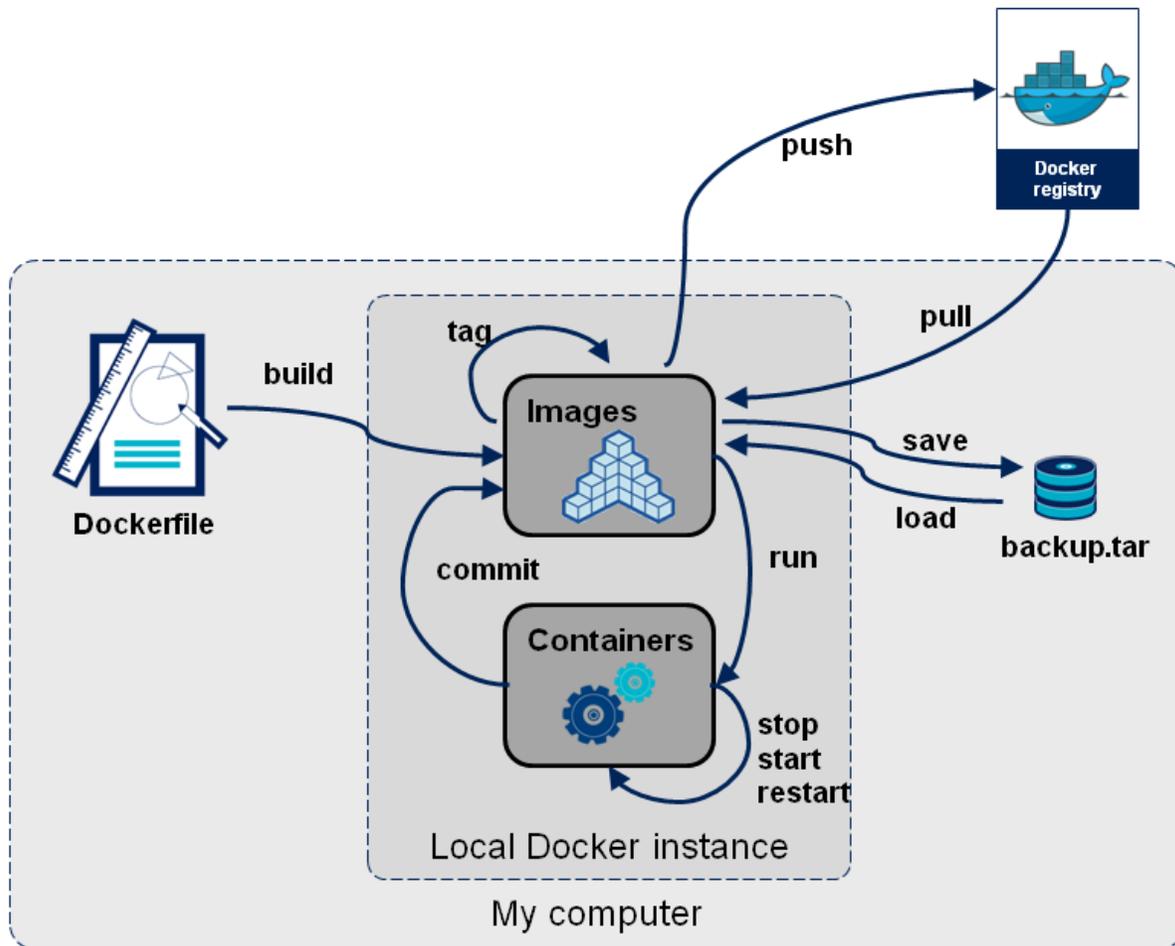
Other Cloud

Public Cloud
(specific vendors)

DOCKER *BUILD*

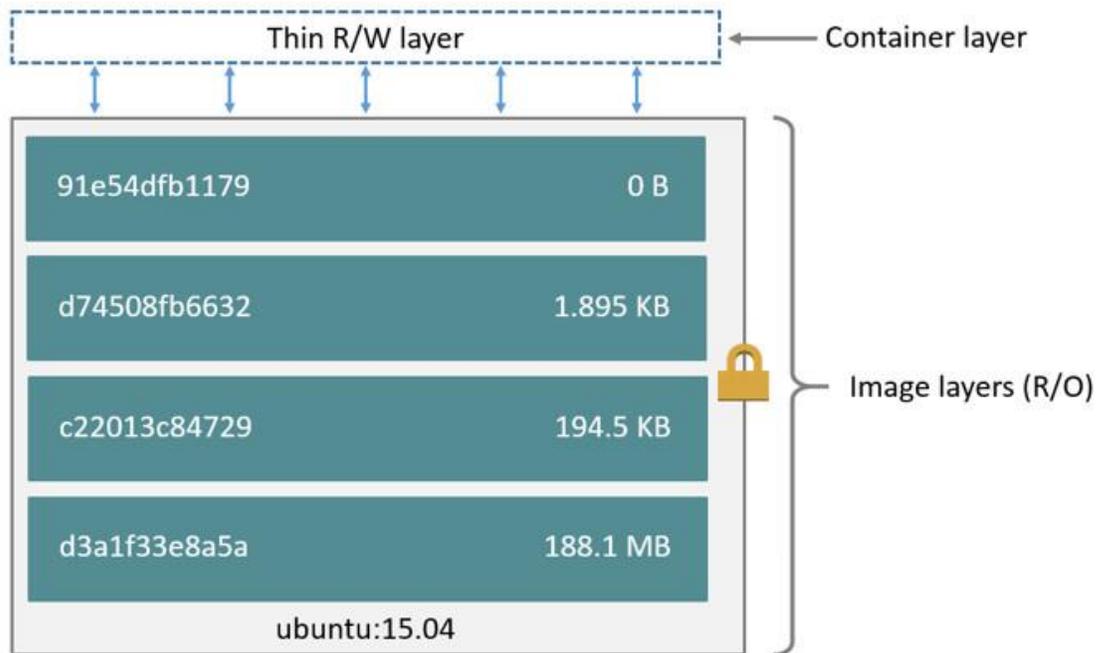


CONCETTI E GESTIONE DEL REGISTRY



DOCKERFILE E SUE BEST PRACTICES

L'esecuzione di ogni verbo presente nel *Dockerfile* produce uno strato nell'immagine risultante dalla *build*.



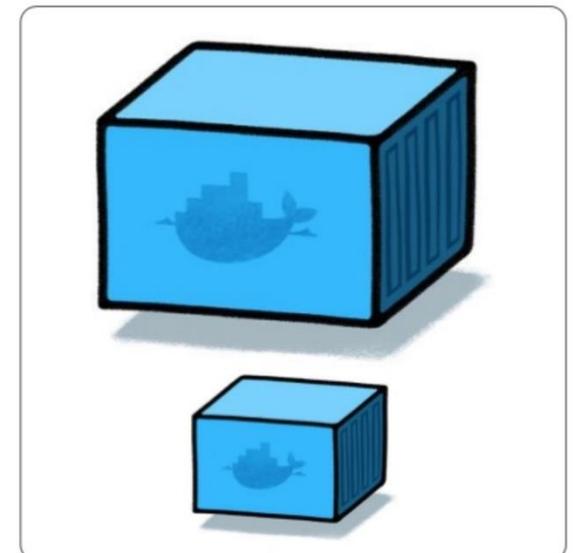
Container
(based on ubuntu:15.04 image)

```
# Dockerfile
# build stage
FROM buildbase as build
...
...
...

# production ready stage
FROM runbase
...
COPY --from=build
/artifact /app
```

1

2

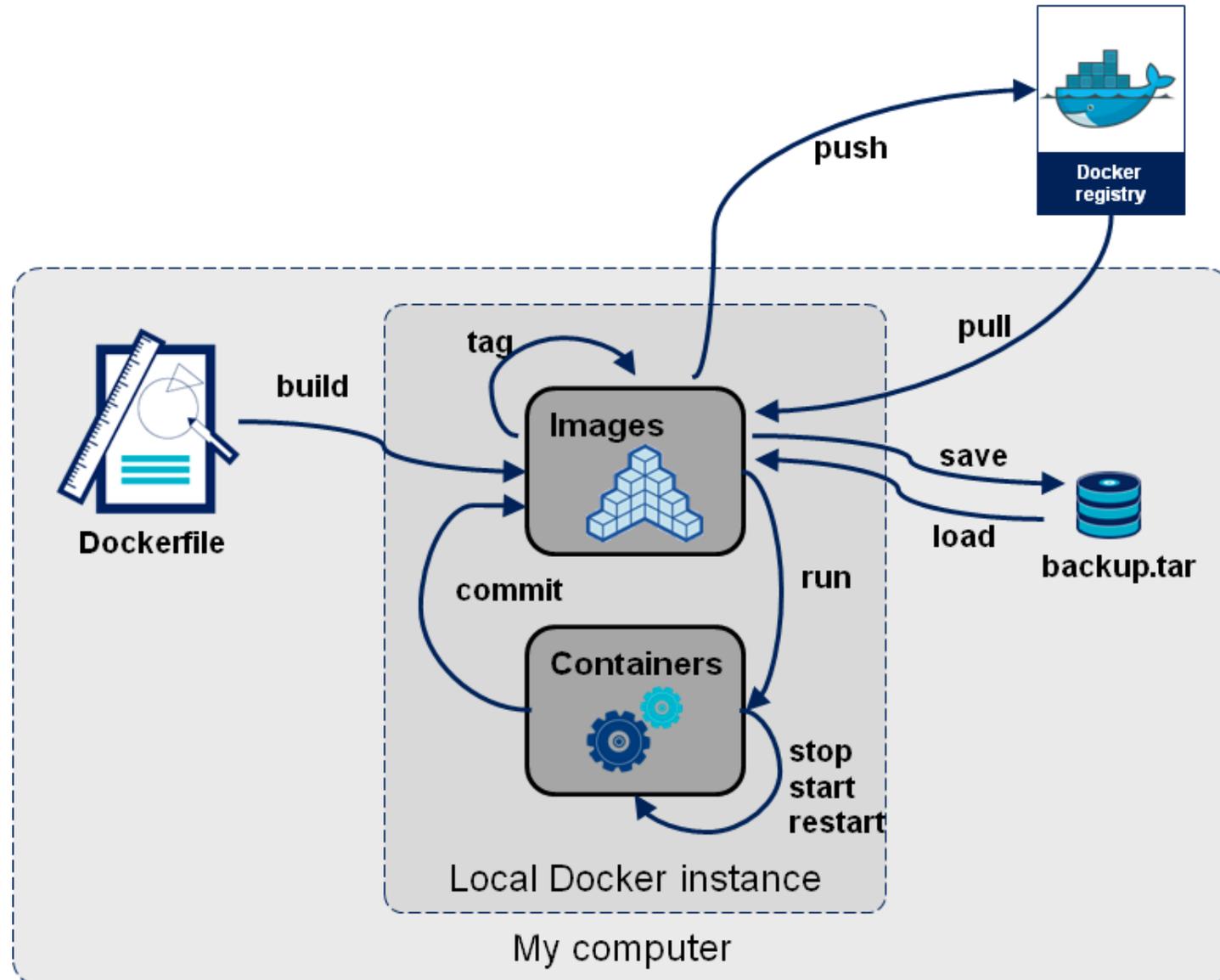


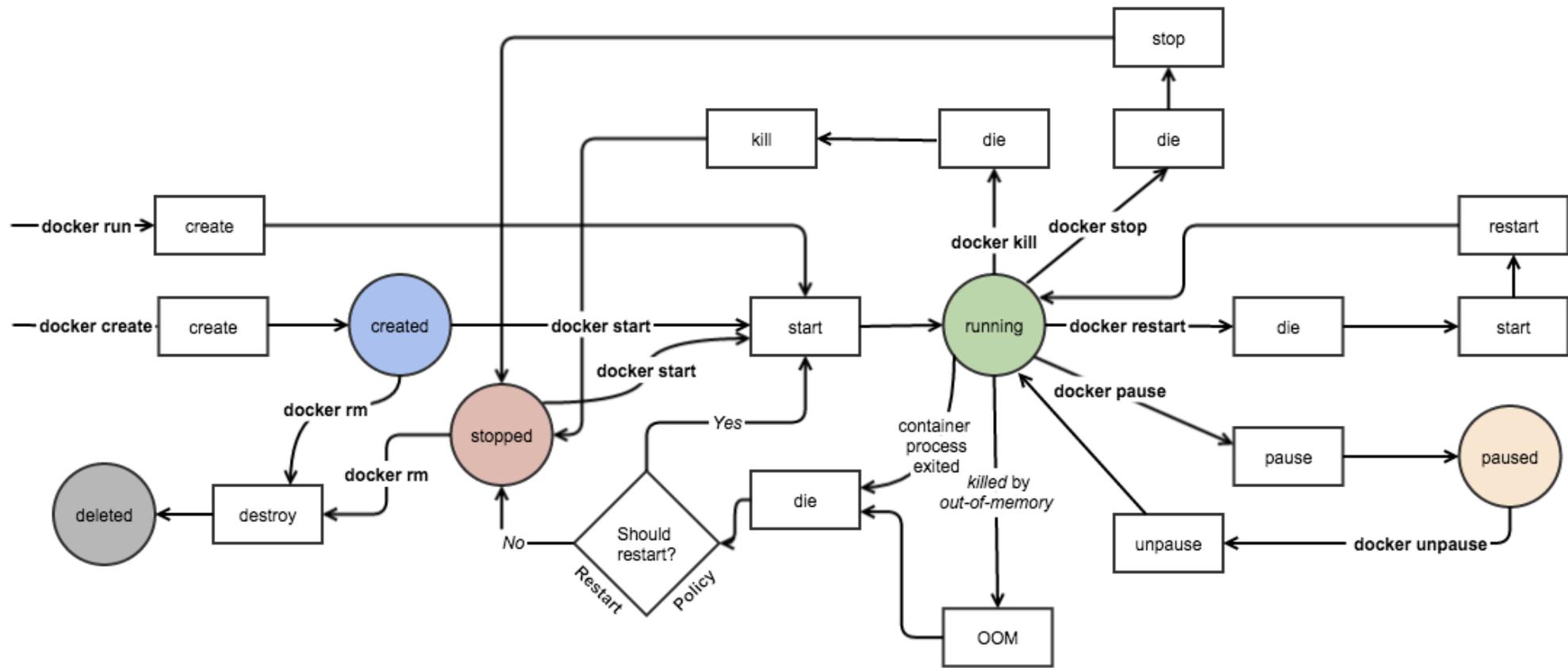


veneto@azuremeetup.it

IT Crowd *vs* Silicon Valley

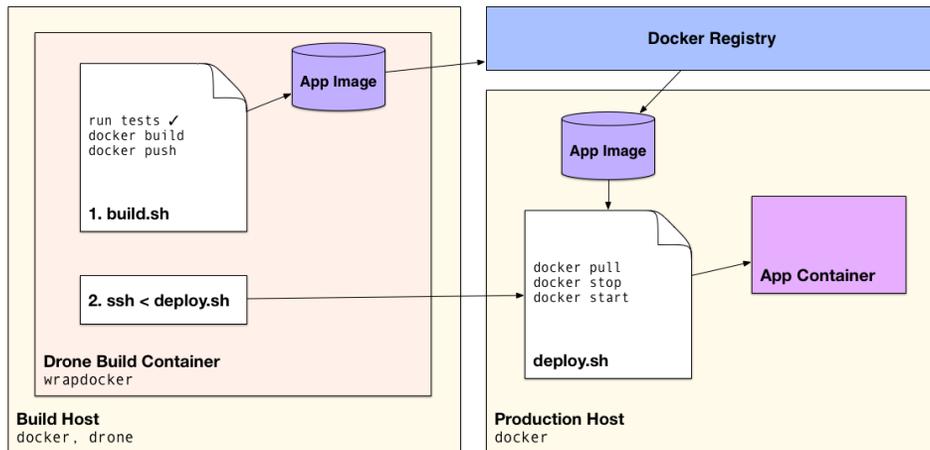
Workflow Docker



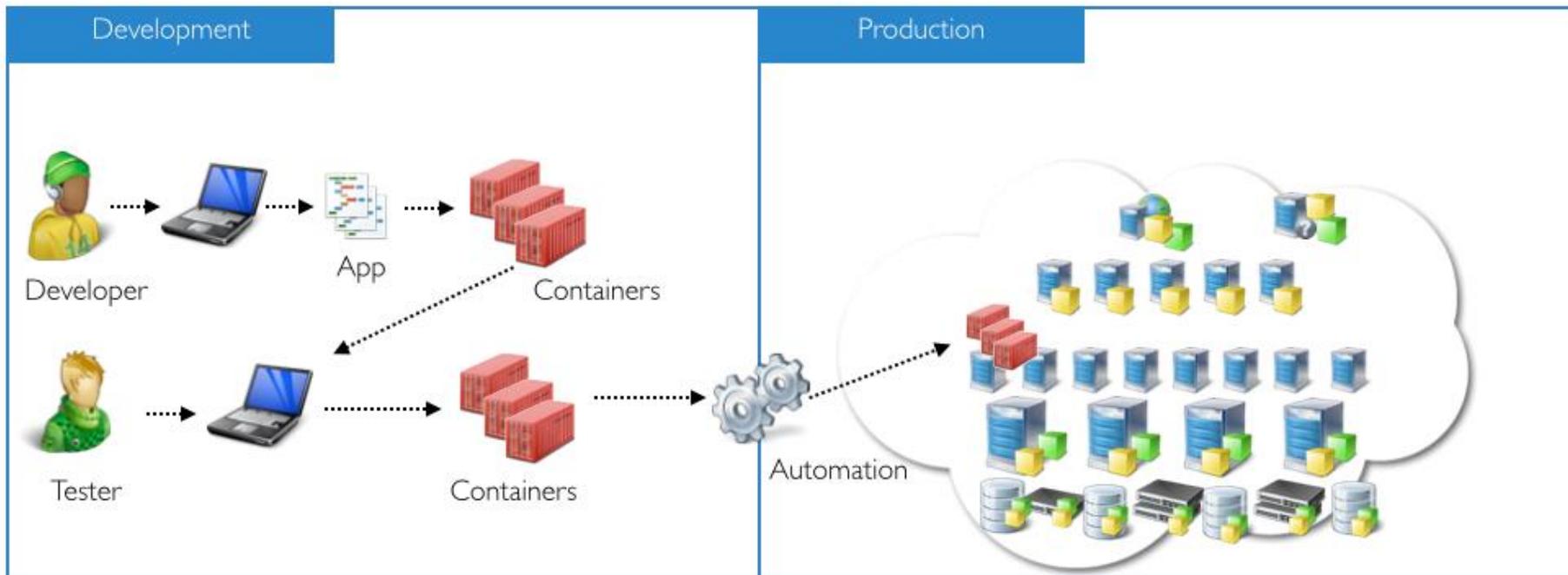
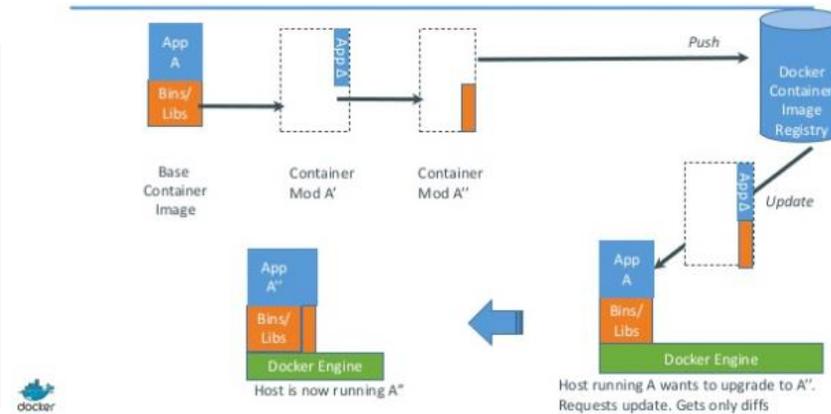


Workflow docker verbs

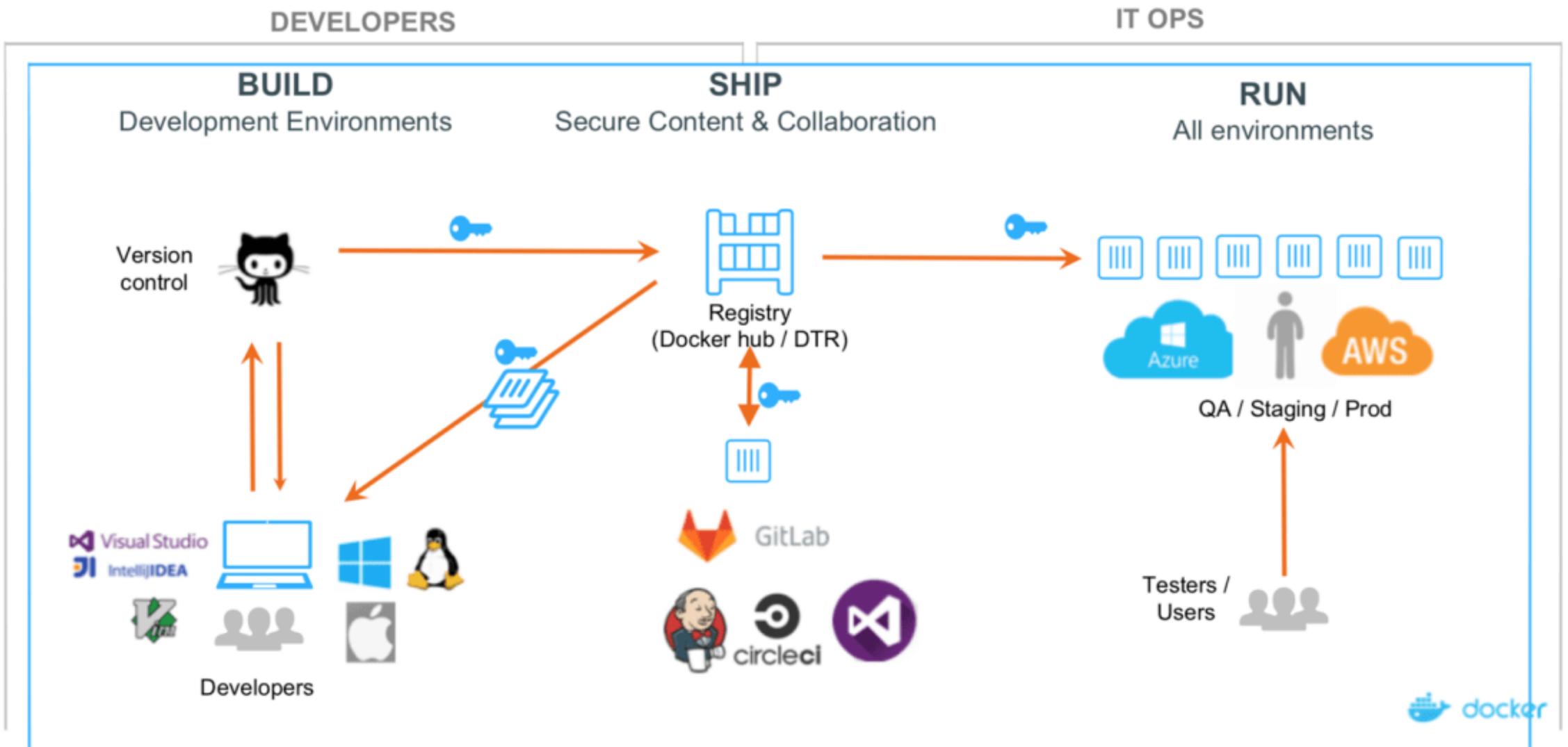
Building & Deploying Docker Containers with Drone



Changes and Updates



Workflow Sviluppo/Produzione



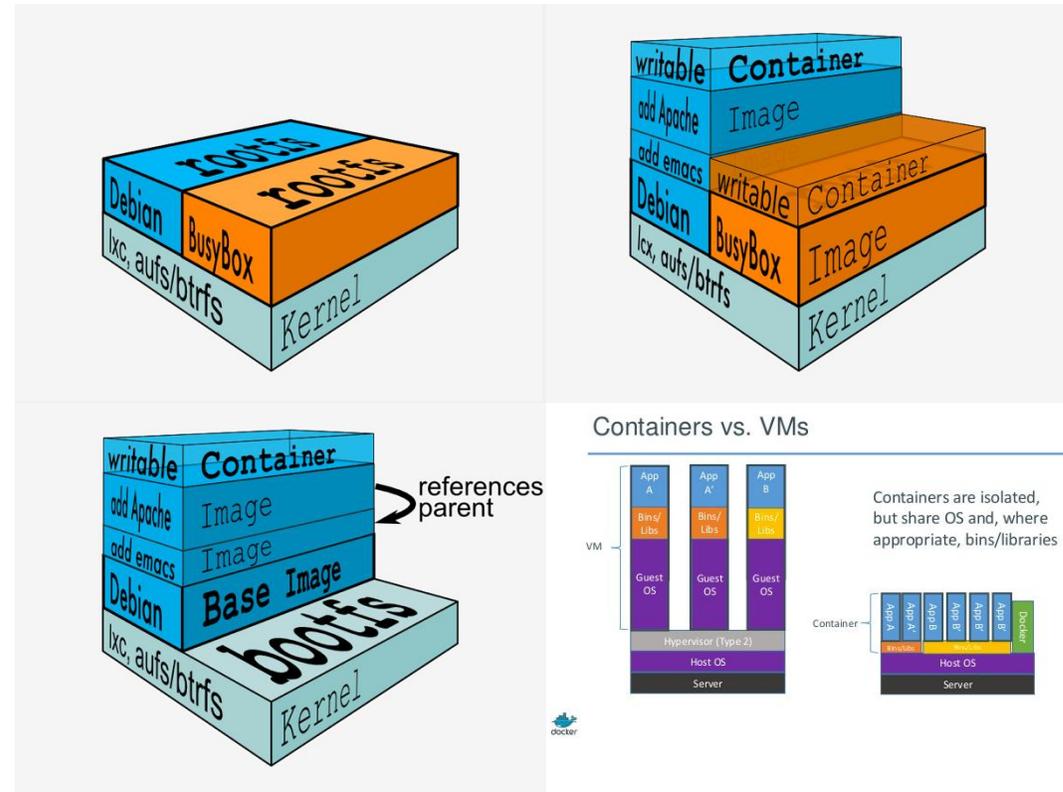
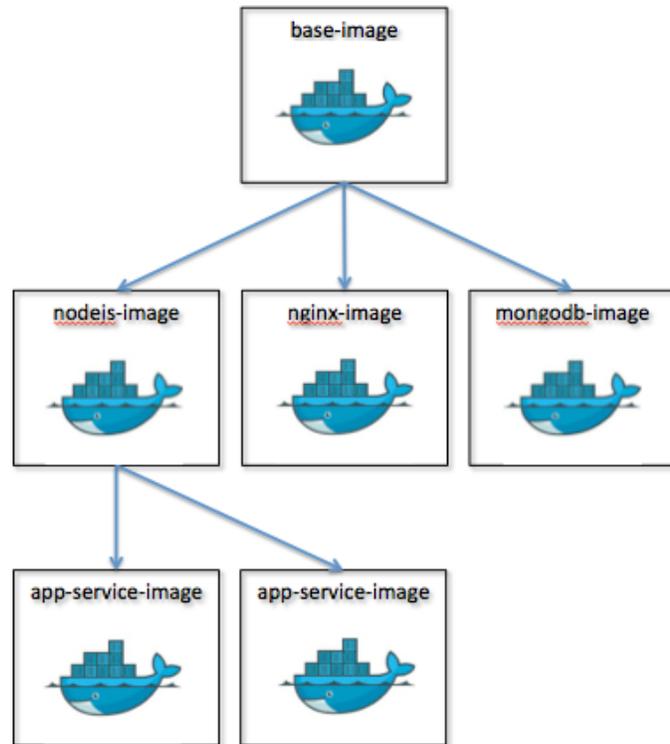
Secure deployment workflow



veneto@azuremeetup.it

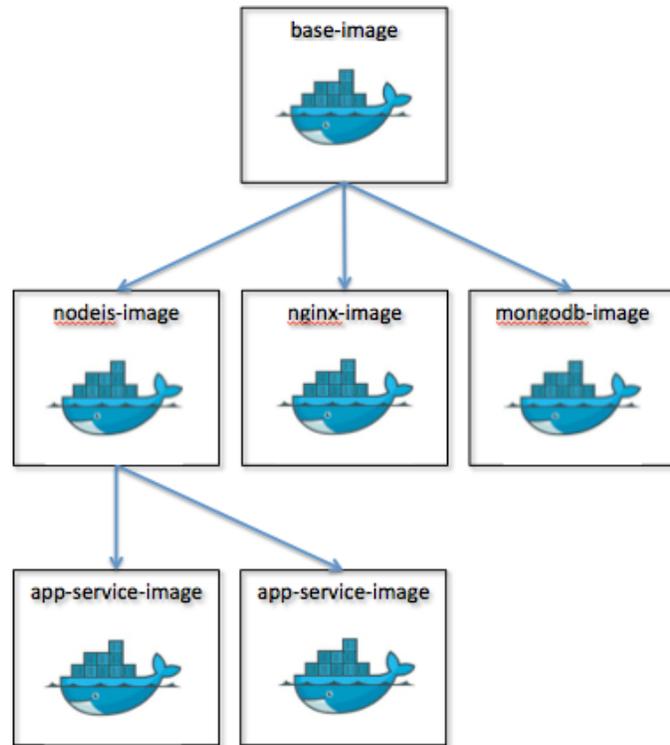
Impossible Engineering *VS* Pimp My Wheels

Container über alles

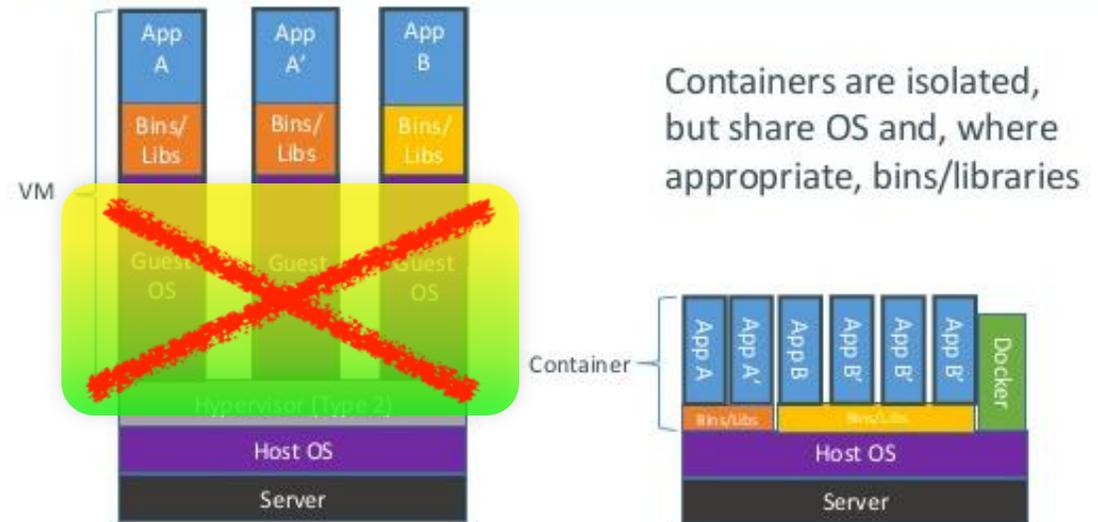


Container: usiamo docker come i mattoncini Lego, per costruire un'infrastruttura slim e modulare.

Container über alles



Containers vs. VMs



Container: usiamo docker come i mattoncini Lego, per costruire un'infrastruttura slim e modulare.

docker build *dockerfile*

```
# This will build the handson-namer image
FROM ubuntu:12.10
MAINTAINER Your Name, your@email.com

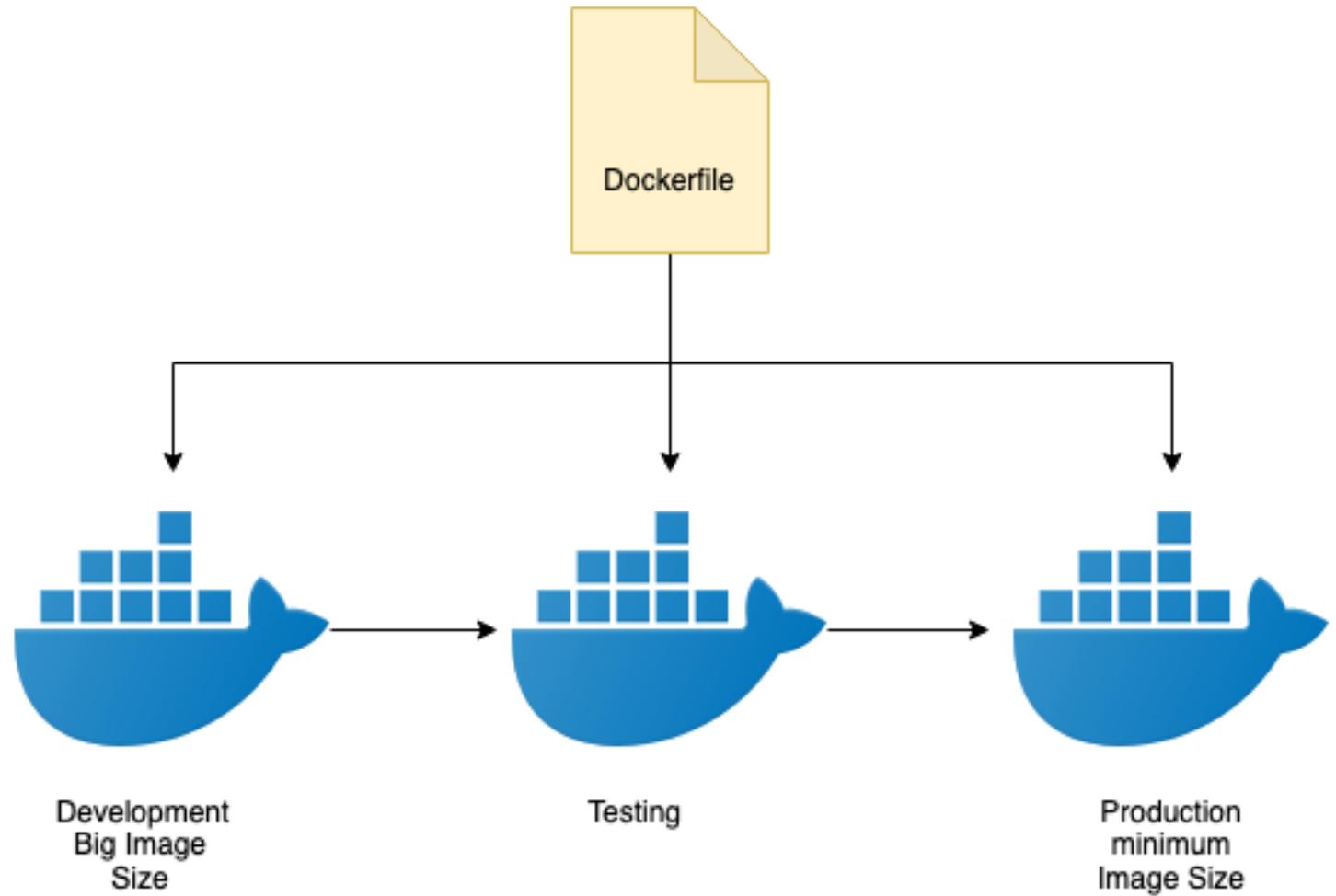
# Install dependencies
RUN apt-get update
RUN apt-get install -y curl git \
    build-essential ruby1.9.3
RUN gem install rubygems-update
RUN update_rubygems
RUN gem install bundler sinatra --no-ri --no-rdoc

# Sinatra uses this port
EXPOSE 9292

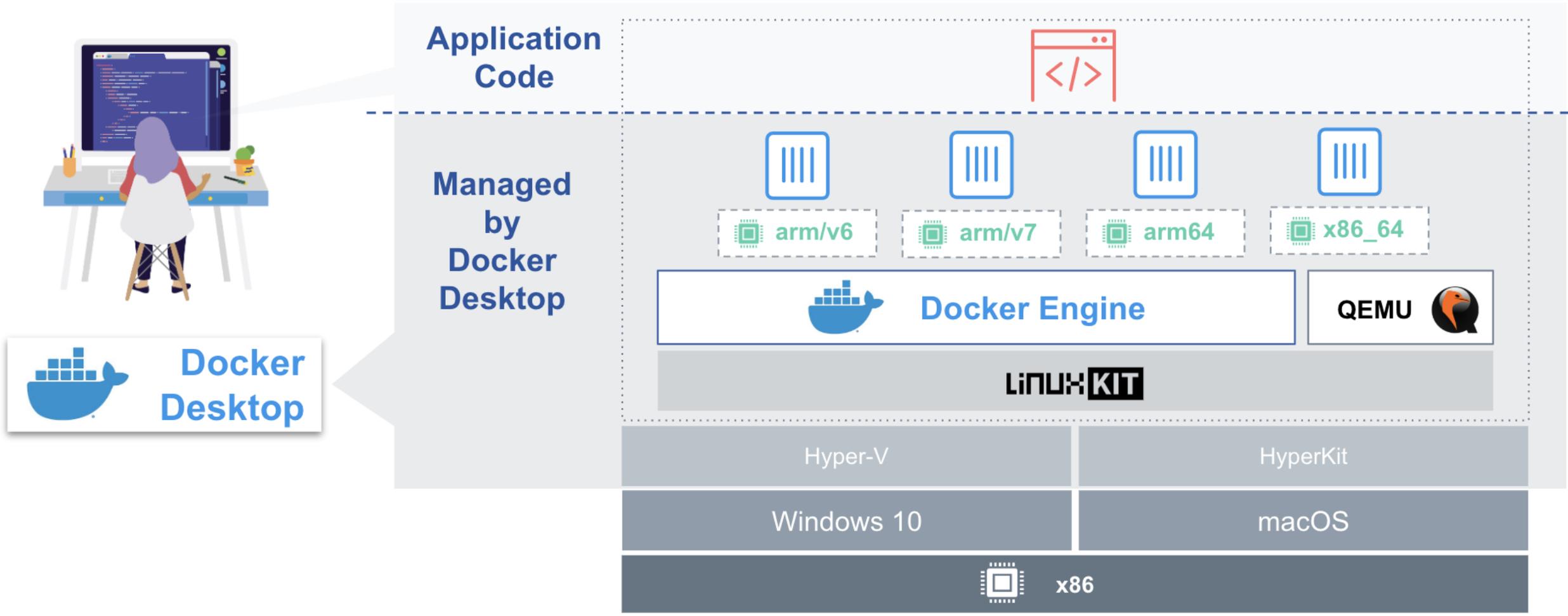
# Copy the app into the container at build time
ADD namer /opt/namer
WORKDIR /opt/namer
RUN bundle install

# Set the startup command
CMD [ "rackup" ]
```

```
1 # Stage - Development
2 FROM Dev-Image as dev
3 ~ . . .
4 ~ . . .
5
6 # Stage - Testing
7 FROM Test-Image as test
8 COPY --from=dev /src/app .
9 ~ . . .
10 ~ . . .
11
12 # Stage - Production
13 FROM Minimum-Image as prod
14 COPY --from=dev /src/app .
15 ~ . . .
16 ~ . . .
```

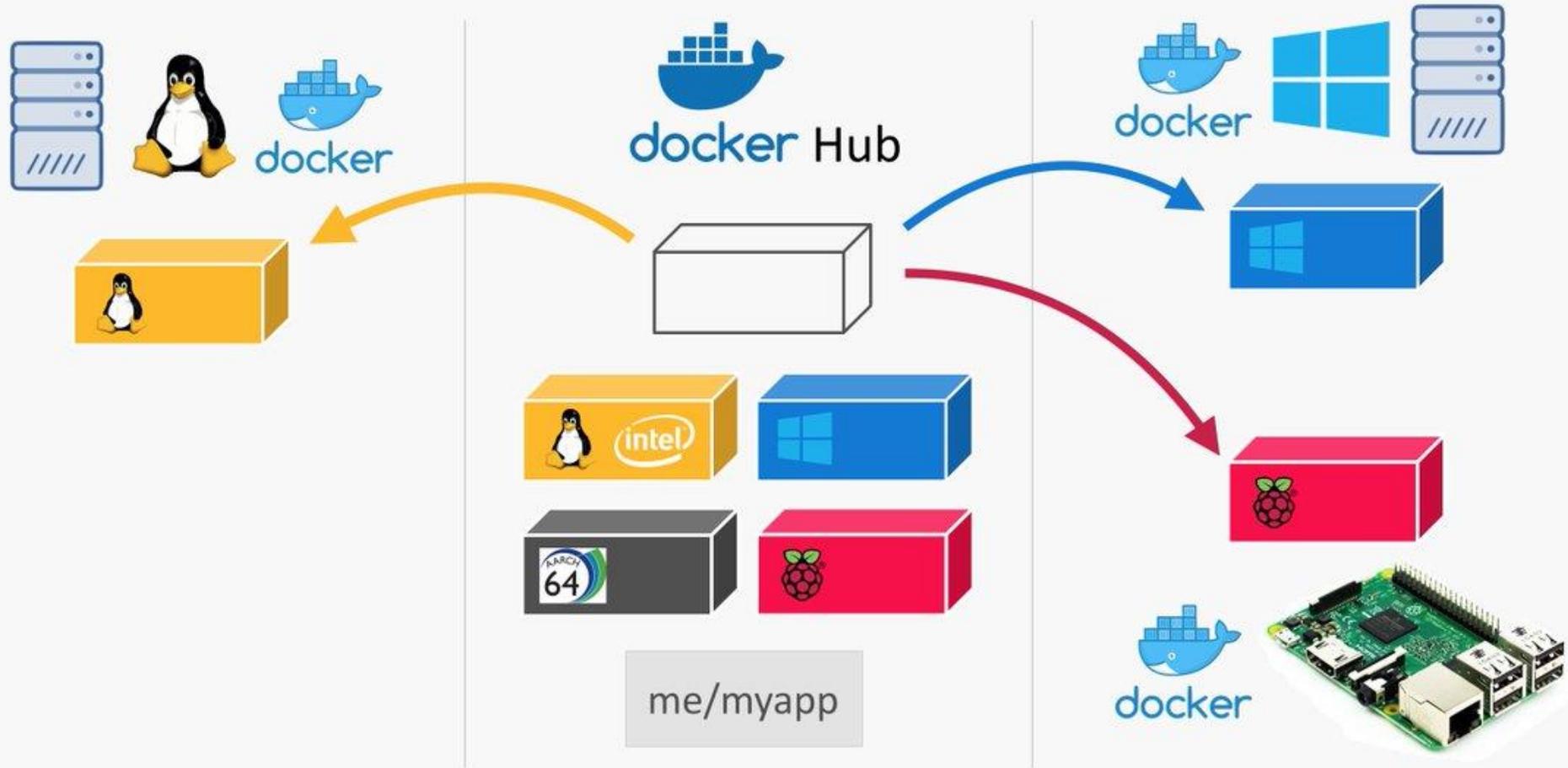


Docker build multi-stage feature



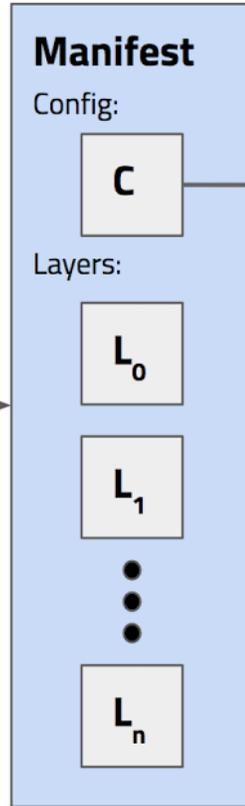
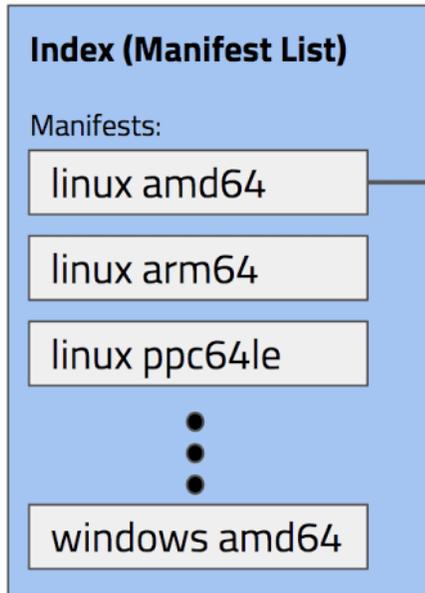
Docker Desktop buildx multi-arch feature

Manifest list: docker container run me/myapp

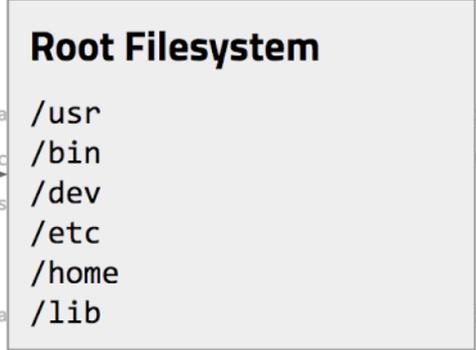
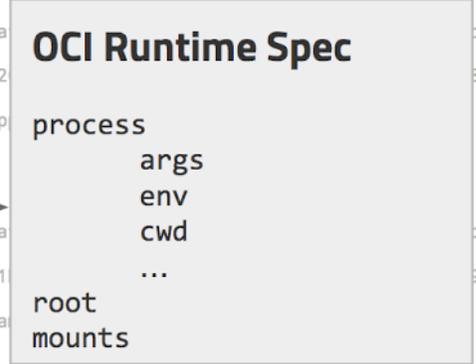


Docker Desktop buildx multi-arch feature

Image Manifests



```
{  
  "schemaVersion": 2,  
  "mediaType": "application/vnd.docker.distribution.manifest.list.v2+json",  
  "manifests": [  
    {  
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",  
      "size": 2094,  
      "digest": "sha256:782...",  
      "platform": {  
        "architecture": "ppc64le",  
        "os": "linux"  
      }  
    },  
    {  
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",  
      "size": 1922,  
      "digest": "sha256:ae1...",  
      "platform": {  
        "architecture": "amd64",  
        "os": "linux",  
        "features": [  
          "sse"  
        ]  
      }  
    },  
    {  
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",  
      "size": 2084,  
      "digest": "sha256:e4c...",  
      "platform": {  
        "architecture": "s390x",  
        "os": "linux"  
      }  
    },  
    {  
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",  
      "size": 2084,  
      "digest": "sha256:07ebe243465ef4a667b78154ae6c3ea46fdb1582936aac3ac899ea311a701b40",  
      "platform": {  
        "architecture": "arm",  
        "os": "linux",  
        "variant": "armv7"  
      }  
    },  
    {  
      "mediaType": "application/vnd.docker.distribution.manifest.v1+json",  
      "size": 2090,  
      "digest": "sha256:fb2fc0707b86dafa9959fe3d29e66af8787aee4d9a23581714be65db4265ad8a",  
      "platform": {  
        "architecture": "arm64",  
        "os": "linux",  
        "variant": "armv8"  
      }  
    }  
  ]  
}
```



Docker Desktop buildx multi-arch feature

docker run -i

```
local ~/workspaces/sqldump/docker-dev (±docker-dev/master)
↳ docker run -i -t --rm \
>     -p 3000:3000 \
>     -v `pwd`:/src \
>     sqldump/docker-dev:0.1 \
>     /bin/bash
[ root@5cf25bf83d1e:/src ]$
```

apriamo una sessione shell per interagire con un'istanza docker



veneto@azuremeetup.it

Chuck *vs* Blacklist

```
docker sbom nginx:latest
```

```
root@master:~# docker sbom nginx:latest
```

```
Syft v0.46.3
```

```
✓ Pulled image
```

```
✓ Loaded image
```

```
✓ Parsed image
```

```
✓ Cataloged packages [151 packages]
```

NAME	VERSION	TYPE
adduser	3.134	deb
apt	2.6.1	deb
base-files	12.4+deb12u1	deb
base-passwd	3.6.1	deb
bash	5.2.15-2+b2	deb
bsdutils	1:2.38.1-5+b1	deb
ca-certificates	20230311	deb
coreutils	9.1-1	deb
curl	7.88.1-10+deb12u1	deb
dash	0.5.12-2	deb
debconf	1.5.82	deb
debian-archive-keyring	2023.3	deb
debianutils	5.7-0.4	deb
diffutils	1:3.8-4	deb

Docker sbom from shell

The screenshot displays the Docker Desktop interface with the 'Images' section selected. The 'nginx:latest' image is highlighted, showing its ID 'fb197595ebe7'. The 'Image hierarchy' and 'Layers (17)' are visible on the left. The main panel shows the 'Vulnerabilities (23)' section, which is expanded to show details for 'CVE-2023-31486'. A red arrow points to this specific vulnerability. The interface also shows a table of other vulnerabilities and their associated packages.

Package	Vulnerabilities
debian/glibc 2.36-9+deb12u9	0 0 0 7 0
debian/systemd 252.31-1~deb12u1	0 0 0 4 0
debian/perl 5.36.0-7+deb12u1	0 0 0 2 0
CVE-2023-31486	N/A
CVE-2011-4116	N/A
debian/gcc-12 12.2.0-14	0 0 0 2 0
debian/util-linux 2.38.1-5+deb12u2	0 0 0 1 0
debian/tar 1.34+dfsg-1.2+deb12u1	0 0 0 1 0
debian/shadow 1:4.13+dfsg1-1	0 0 0 1 0
debian/libcrypt20 1.10.1-3	0 0 0 1 0
debian/gnutls28 3.7.9-2+deb12u3	0 0 0 1 0
debian/gnupg2 2.2.40-1.1	0 0 0 1 0

Docker Desktop image vulnerabilities inspection (*docker scout*)

docker desktop PRO

More Docker. Easy Access. New Streamlined Plans. [Learn more](#)

Containers

Images

Volumes

Builds

Docker Scout

Extensions

Manage

Disk usage

Snyk

Images / jarviselastic:latest

jarviselastic:latest
e10c4bac68d7

CREATED 5 months ago SIZE 1 GB

Recommended fixes

Run

Analyzed by Docker Scout

Give feedback

Image hierarchy

FROM ubuntu:20.04, focal, focal-20240530

ALL jarviselastic:latest

Layers (24)

0	ARG RELEASE	0 B	✓
1	ARG LAUNCHPAD_BUILD_ARCH	0 B	✓
2	LABEL org.opencontainers.image.ref.name=ubuntu	0 B	✓
3	LABEL org.opencontainers.image.version=20.04	0 B	✓
4	ADD file:6d8cc056ee741f09a6c7d965d8e2027d80ed2eccbf0312593ce52d9256db437 in /	74.64 MB	⚠
5	CMD ["/bin/bash"]	0 B	✓
6	RUN /bin/sh -c yes no dpkg-reconfigure dash && for iter in 1 2 3 4 5 6 7 8 9 10; do export DEBIAN_FRONTEND=noninteractive &...	19.62 MB	⚠
7	RUN /bin/sh -c groupadd -g 1000 elasticsearch && adduser --uid 1000 --gid 1000 --home /usr/share/elasticsearch elasticsearch...	413.7 KB	✓
8	ENV ELASTIC_CONTAINER=true	0 B	✓
9	WORKDIR /usr/share/elasticsearch	4.1 KB	✓
10	COPY /usr/share/elasticsearch /usr/share/elasticsearch # buildkit	541.57 MB	⚠
11	COPY /bin/tini /bin/tini # buildkit	36.86 KB	✓
12	ENV PATH=/usr/share/elasticsearch/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin	0 B	✓
13	COPY bin/docker-entrypoint.sh /usr/local/bin/docker-entrypoint.sh # buildkit	24.58 KB	✓
14	RUN /bin/sh -c chmod g=u /etc/passwd && chmod 0555 /usr/local/bin/docker-entrypoint.sh && find / -xdev -perm -4000 -exec ...	528.38 KB	✓
15	COPY bin/docker-openjdk /etc/ca-certificates/update.d/docker-openjdk # buildkit	20.48 KB	✓
16	RUN /bin/sh -c /etc/ca-certificates/update.d/docker-openjdk # buildkit	184.32 KB	✓
17	EXPOSE map[9200/tcp:{}] 9300/tcp:[]	0 B	✓
18	LABEL org.label-schema.build-date=2024-07-25T14:37:42.448799567Z org.label-schema.license=Elastic-License-2.0 org.label-...	0 B	✓
19	ENTRYPOINT ["/bin/tini" "--" /usr/local/bin/docker-entrypoint.sh]	0 B	✓
20	CMD [eswrapper]	0 B	✓
21	ENV discovery.type=single-node	0 B	✓

Images (2) Vulnerabilities (36) Packages (381)

FROM ubuntu:20.04 0 0 2 12 0

Docker Official Image

Packages: 128 created 6 months ago

Newer image available

jarviselastic:latest 0 5 13 18 0

Packages: 381

Engine running

RAM 2.20 GB CPU 0.00% Disk 51.75 GB avail. of 62.67 GB

Terminal v4.36.0

Docker Desktop image inspection (*docker scout*)

The screenshot shows the Docker Desktop interface for the 'jarviselastic:latest' image. The left sidebar contains navigation options: Containers, Images, Volumes, Builds, Docker Scout, Extensions, and Manage. The main area displays the image hierarchy, layers (24), and a list of installed packages (381) analyzed by Docker Scout. The status bar at the bottom shows 'Engine running' and system resources like RAM (2.14 GB) and CPU (0.12%).

Image hierarchy

- FROM ubuntu:20.04, focal, focal-20240530
- ALL jarviselastic:latest

Layers (24)

Layer ID	Command	Size	Status
0	ARG RELEASE	0 B	✓
1	ARG LAUNCHPAD_BUILD_ARCH	0 B	✓
2	LABEL org.opencontainers.image.ref.name=ubuntu	0 B	✓
3	LABEL org.opencontainers.image.version=20.04	0 B	✓
4	ADD file:6d8cc056ee741f09a6c7d965d8e2027d80ed2eccbf0312593ce52d9256db437 in /	74.64 MB	⚠
5	CMD ["/bin/bash"]	0 B	✓
6	RUN /bin/sh -c yes no dpkg-reconfigure dash && for iter in 1 2 3 4 5 6 7 8 9 10; do export DEBIAN_FRONTEND=noninteractive && ...	19.62 MB	⚠
7	RUN /bin/sh -c groupadd -g 1000 elasticsearch && adduser --uid 1000 --gid 1000 --home /usr/share/elasticsearch elasticsearch...	413.7 KB	✓
8	ENV ELASTIC_CONTAINER=true	0 B	✓
9	WORKDIR /usr/share/elasticsearch	4.1 KB	✓
10	COPY /usr/share/elasticsearch /usr/share/elasticsearch # buildkit	541.57 MB	⚠
11	COPY /bin/tini /bin/tini # buildkit	36.86 KB	✓
12	ENV PATH=/usr/share/elasticsearch/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin	0 B	✓
13	COPY bin/docker-entrypoint.sh /usr/local/bin/docker-entrypoint.sh # buildkit	24.58 KB	✓
14	RUN /bin/sh -c chmod g=u /etc/passwd && chmod 0555 /usr/local/bin/docker-entrypoint.sh && find / -xdev -perm -4000 -exec ...	528.38 KB	✓
15	COPY bin/docker-openjdk /etc/ca-certificates/update.d/docker-openjdk # buildkit	20.48 KB	✓
16	RUN /bin/sh -c /etc/ca-certificates/update.d/docker-openjdk # buildkit	184.32 KB	✓
17	EXPOSE map[9200/tcp:{} 9300/tcp:{}]	0 B	✓
18	LABEL org.label-schema.build-date=2024-07-25T14:37:42.448799567Z org.label-schema.license=Elastic-License-2.0 org.label-...	0 B	✓
19	ENTRYPOINT ["/bin/tini" "--" "/usr/local/bin/docker-entrypoint.sh"]	0 B	✓
20	CMD ["eswrapper"]	0 B	✓
21	ENV discovery.type=single-node	0 B	✓

Installed Packages (381)

- ubuntu/acl 2.2.53-6
- ubuntu/adduser 3.118ubuntu2
- ubuntu/apt 2.0.10
- ubuntu/attr 1.2.4.48-5
- ubuntu/audit 1:2.8.5-2ubuntu6
- ubuntu/base-files 11ubuntu5.8
- ubuntu/base-passwd 3.5.47
- ubuntu/bash 5.0-6ubuntu1.2
- ubuntu/brotli 1.0.7-6ubuntu0.1
- ubuntu/bsdutils 1:2.34-0.1ubuntu9.6

Docker Desktop image package installed (*docker scout*)

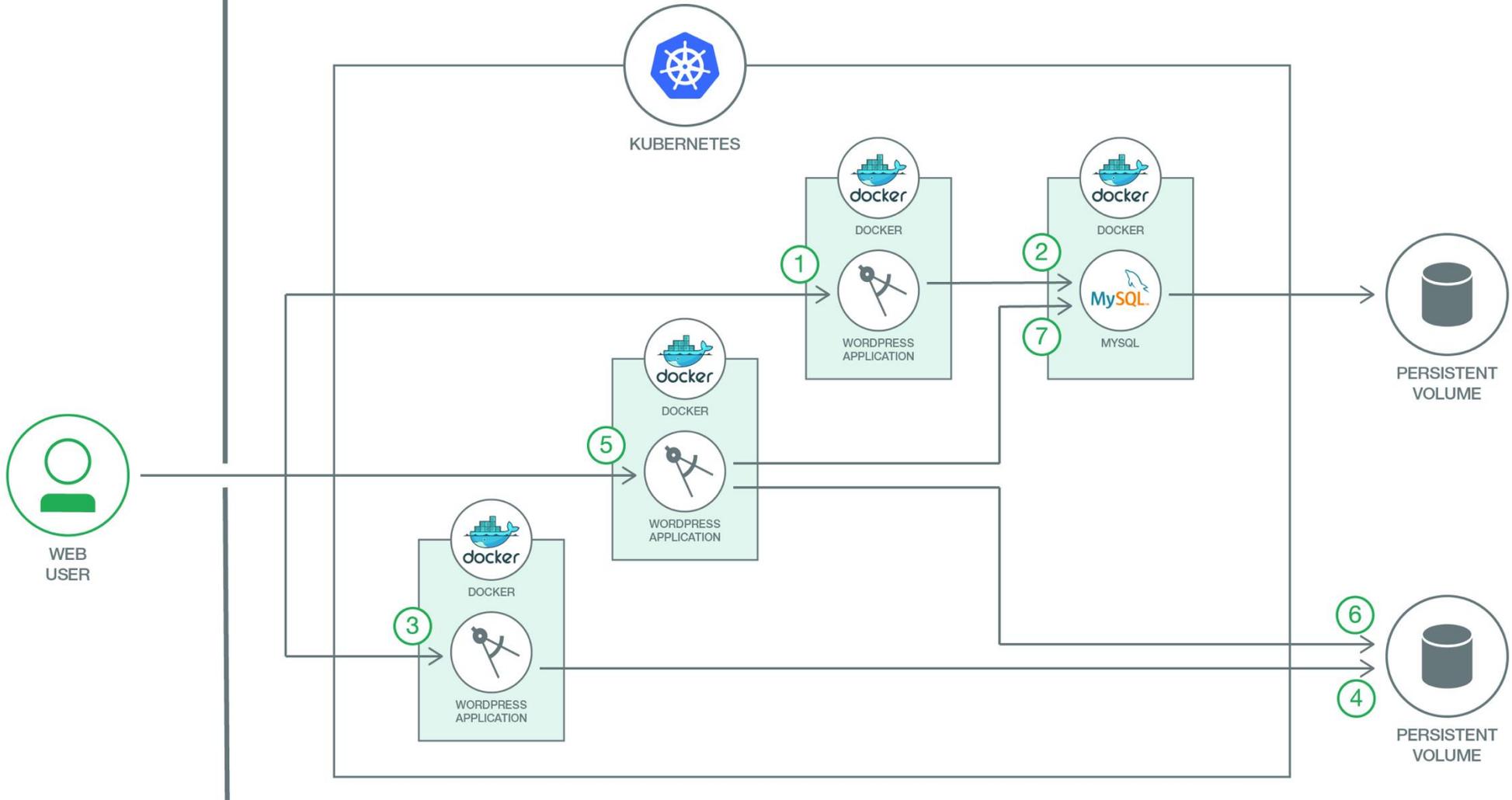


veneto@azuremeetup.it

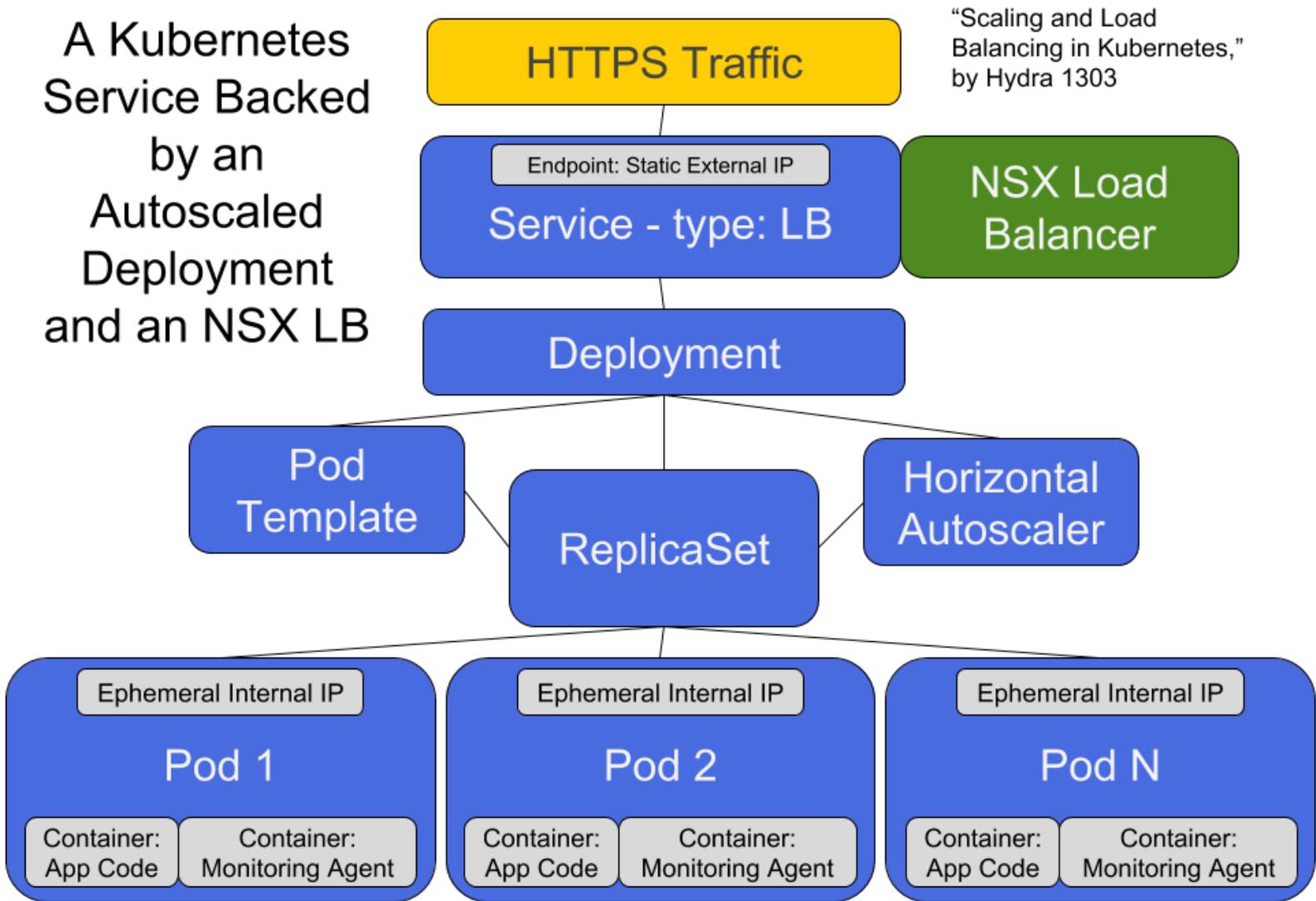
Battle Royale *vs* Code Monkeys

CLIENT DEVICE

PROVIDER CLOUD



Schema concettuale Kubernetes POD



Schema Kubernetes

K8 Deployment YAML + DEMO

```
! nginx-deployment-withrolling.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    labels:
5      environment: test
6    name: testdeploy
7  spec:
8    replicas: 3
9    selector:
10   matchLabels:
11     environment: test
12   minReadySeconds: 10
13   strategy:
14     rollingUpdate:
15       maxSurge: 1
16       maxUnavailable: 0
17     type: RollingUpdate
18   template:
19     metadata:
20       labels:
21         environment: test
22     spec:
23       containers:
24       - image: nginx:1.16
25         name: nginx
```



Manages pods with label `environment: test` and manages replicaset defined within



Manages pods with label `environment: test`



Nginx:1.16

`environment: test`



Nginx:1.16

`environment: test`

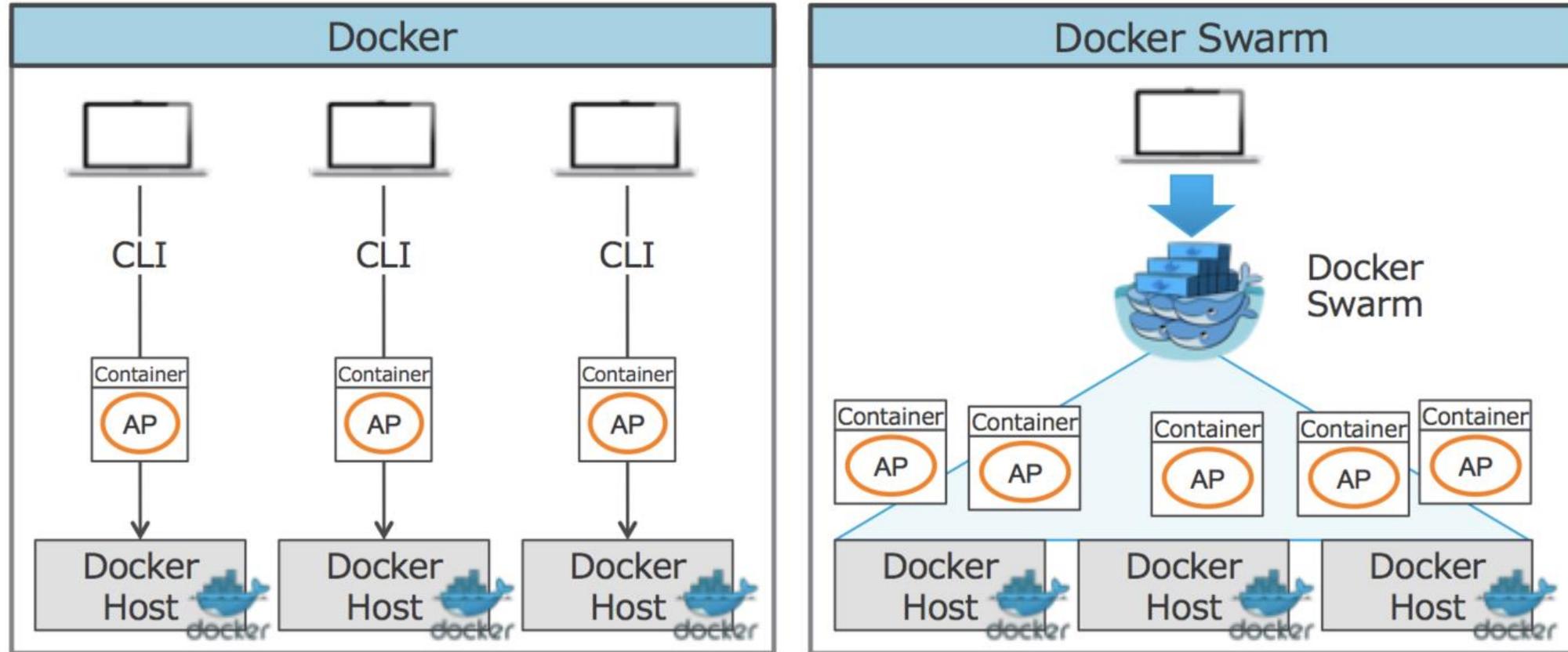


Nginx:1.16

`environment: test`

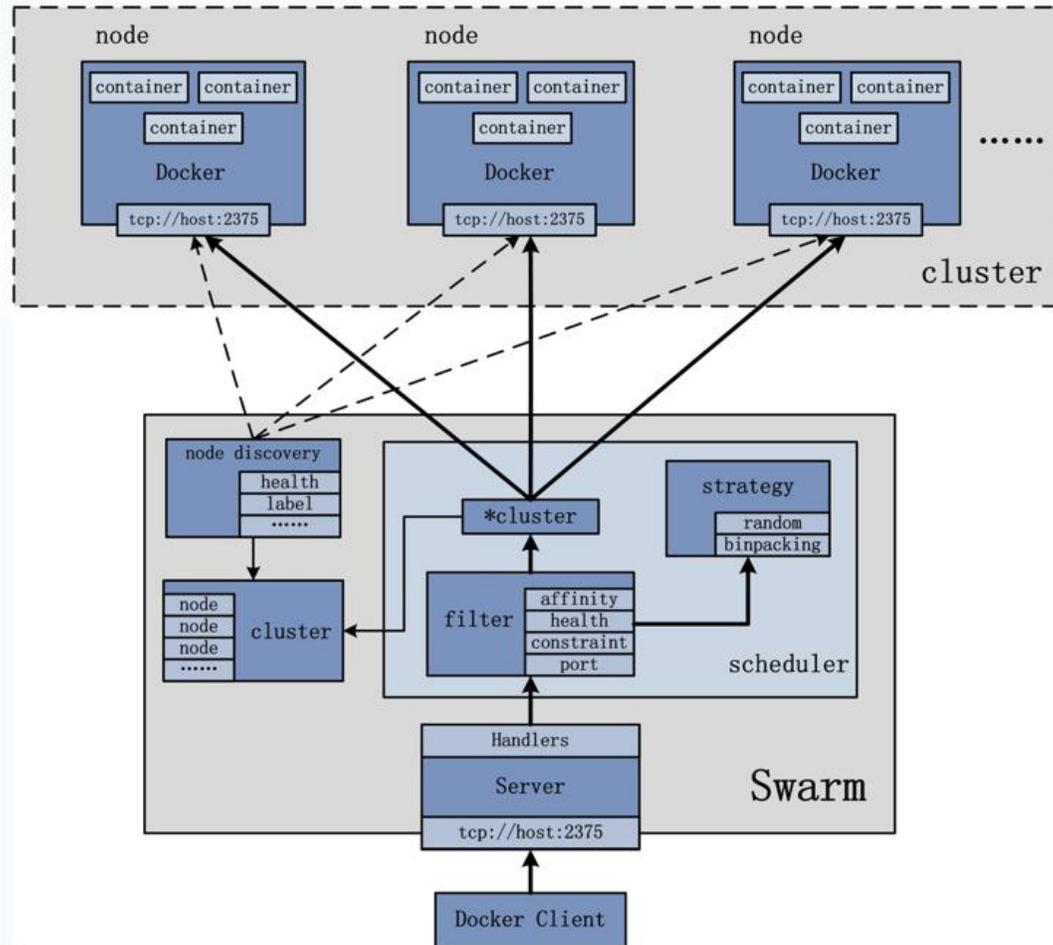
Kubernetes Deployment

Docker Engine vs Docker Swarm



Docker Swarm

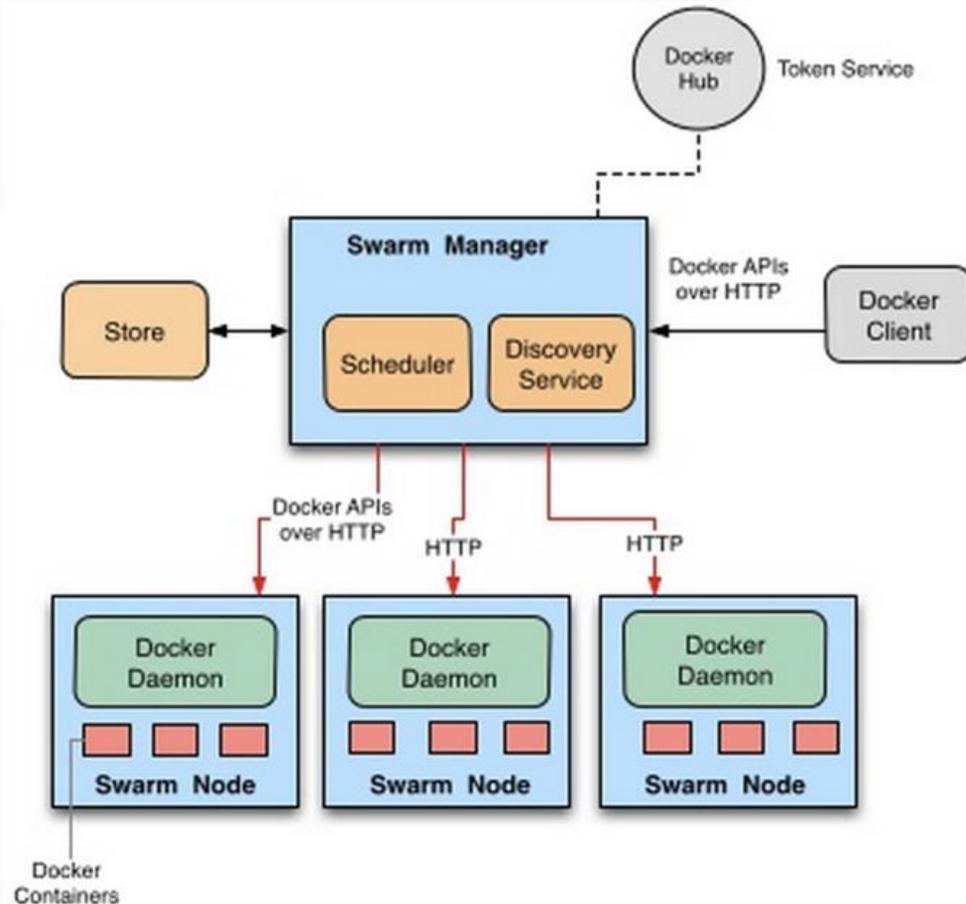
ARCHITETTURA FISICA DOCKER SWARM



Swarm master organizza una rete di docker-host garantendo raggiungibilità ed accessibilità così da disporre tramite il docker client di un punto di accesso unico a tutte le risorse.

Docker Swarm

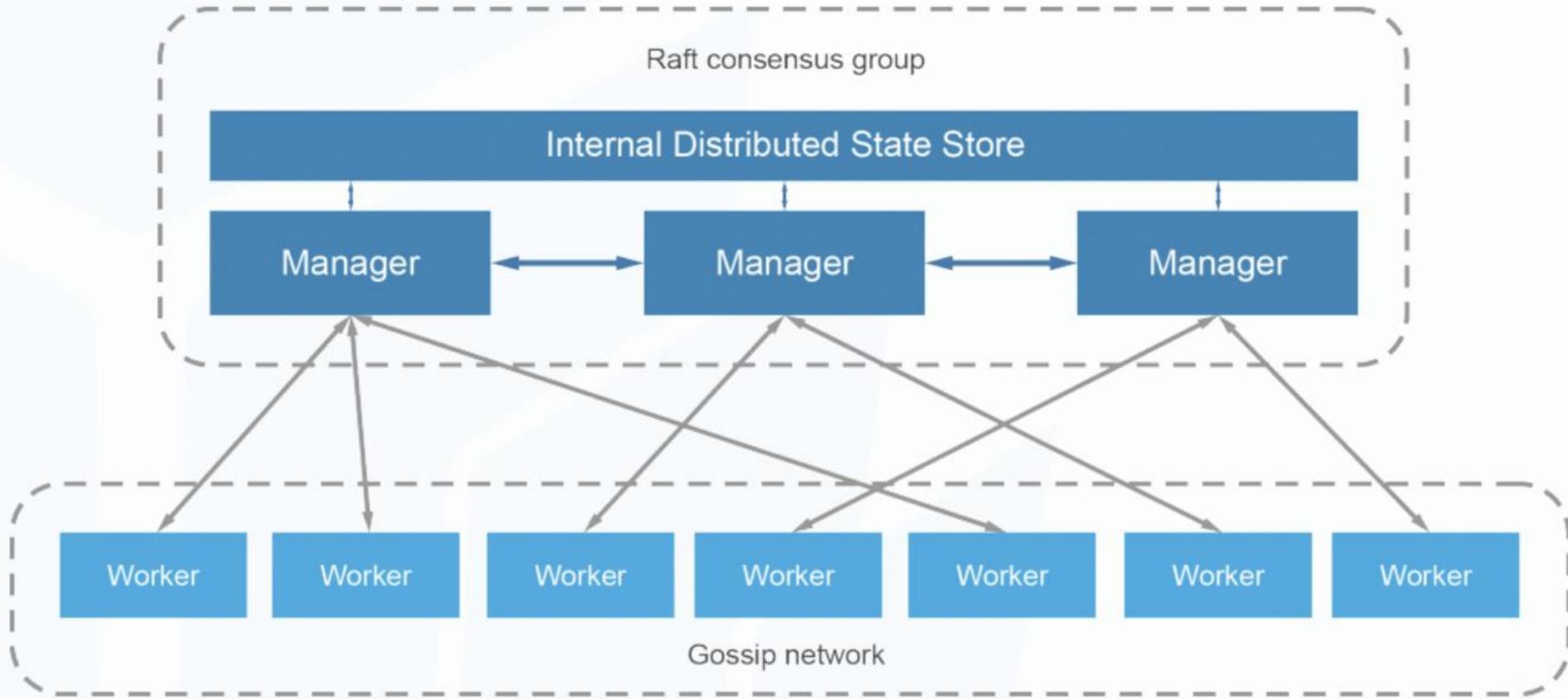
ARCHITETTURA FUNZIONALE DI UN CLUSTER



- Docker Client facilita l'accesso alle risorse del cluster.
- Docker Hub è il repository da cui attingere per le immagini da eseguire come contaniers dal Docker Daemon.

Docker Swarm

ARCHITETTURA FUNZIONALE DI UN CLUSTER



Docker Swarm



veneto@azuremeetup.it

*Ci fermiamo per la pausa Natalizia,
ma gli eventi riprenderanno nel 2025
con tante sorprese.*

Scrivici se vuoi essere dei nostri!!

...e buone Feste!



| [Call4Speaker](#) |