



# DOCKER, DOCKER SWARM

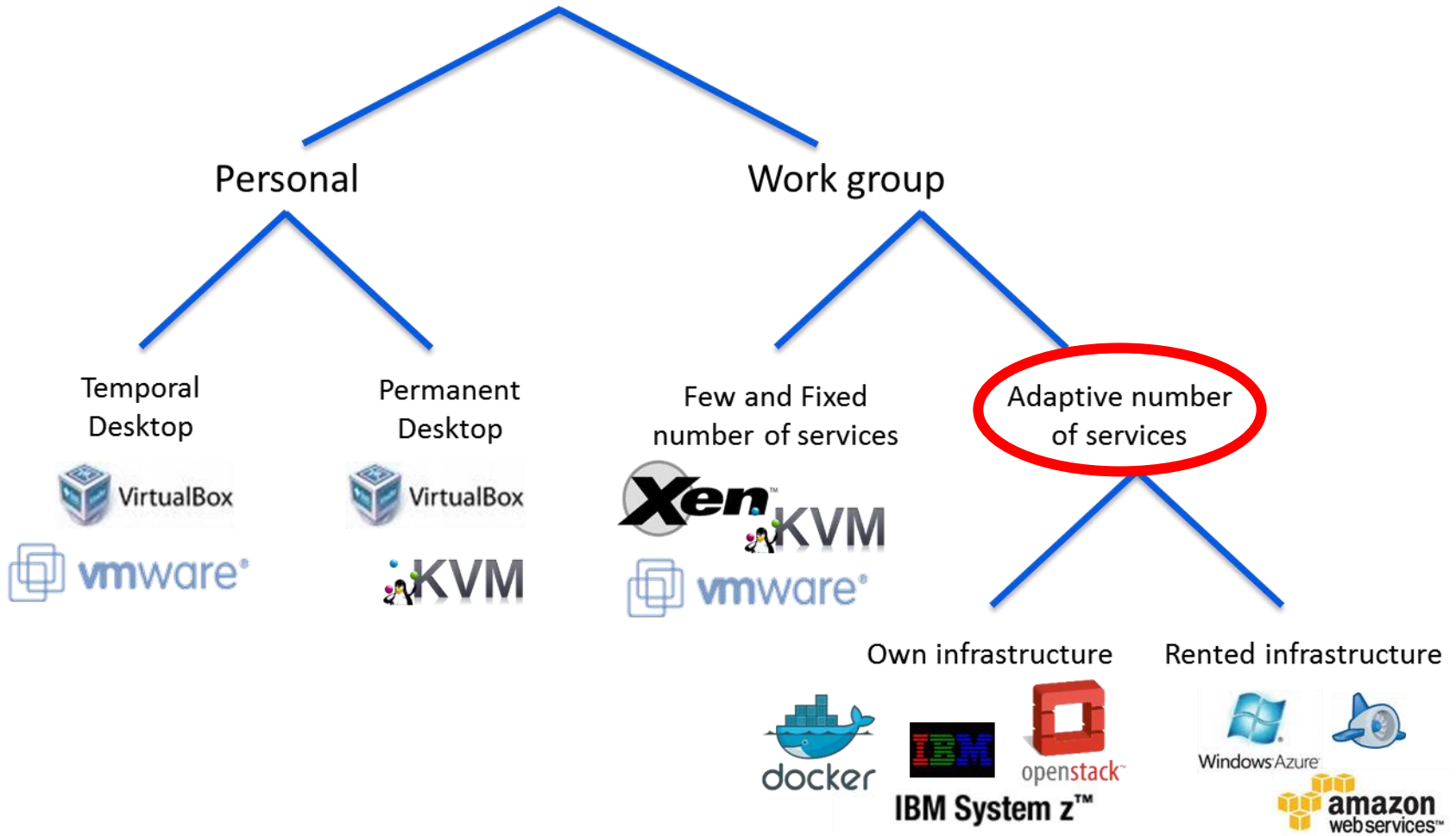
- Example of virtualization with dockers.
- Understanding how a cloud provider use to work.
- Virtualization in a cloud environment.



- Introduction
- Docker
- Docker Swarm

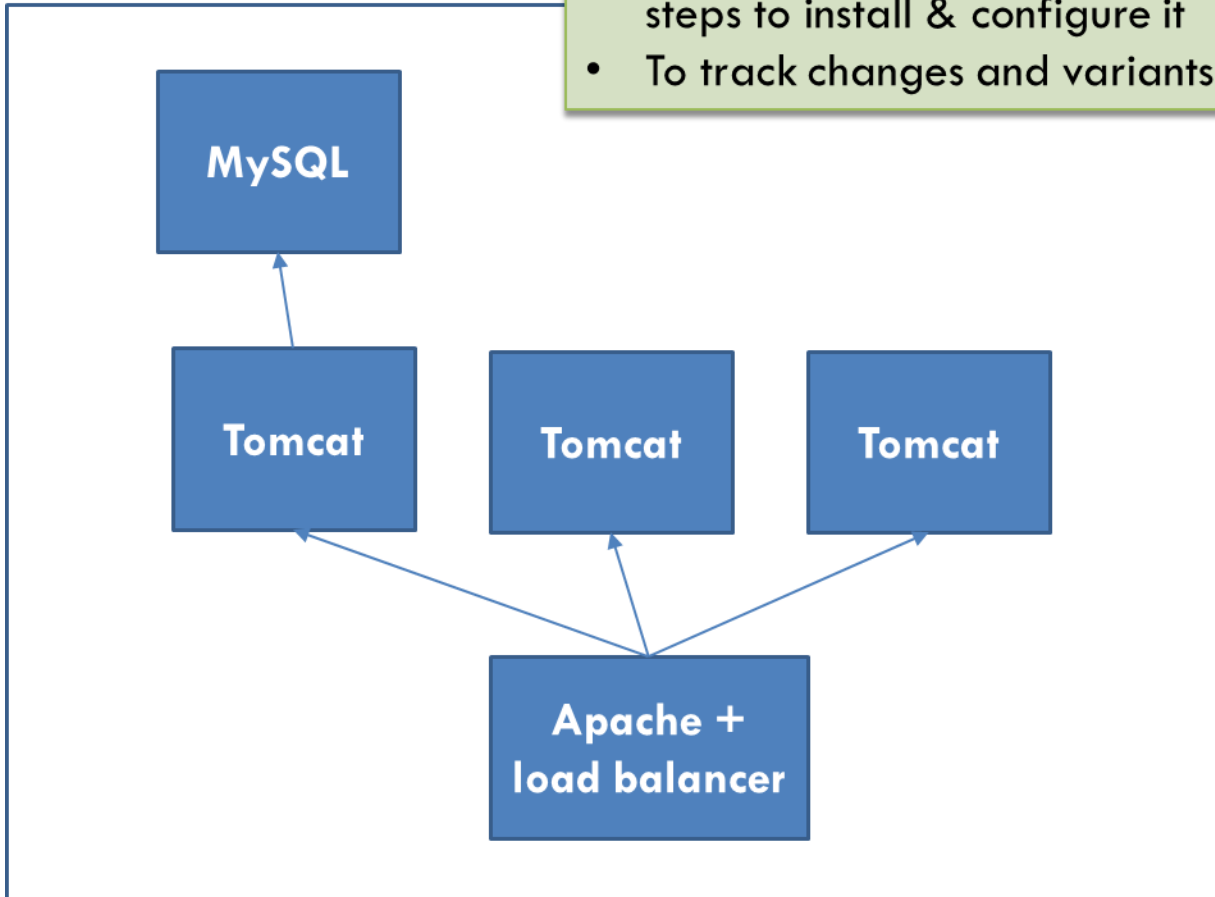
- **Introduction**
- **Docker**
- **Docker Swarm**

# Some common scenarios...



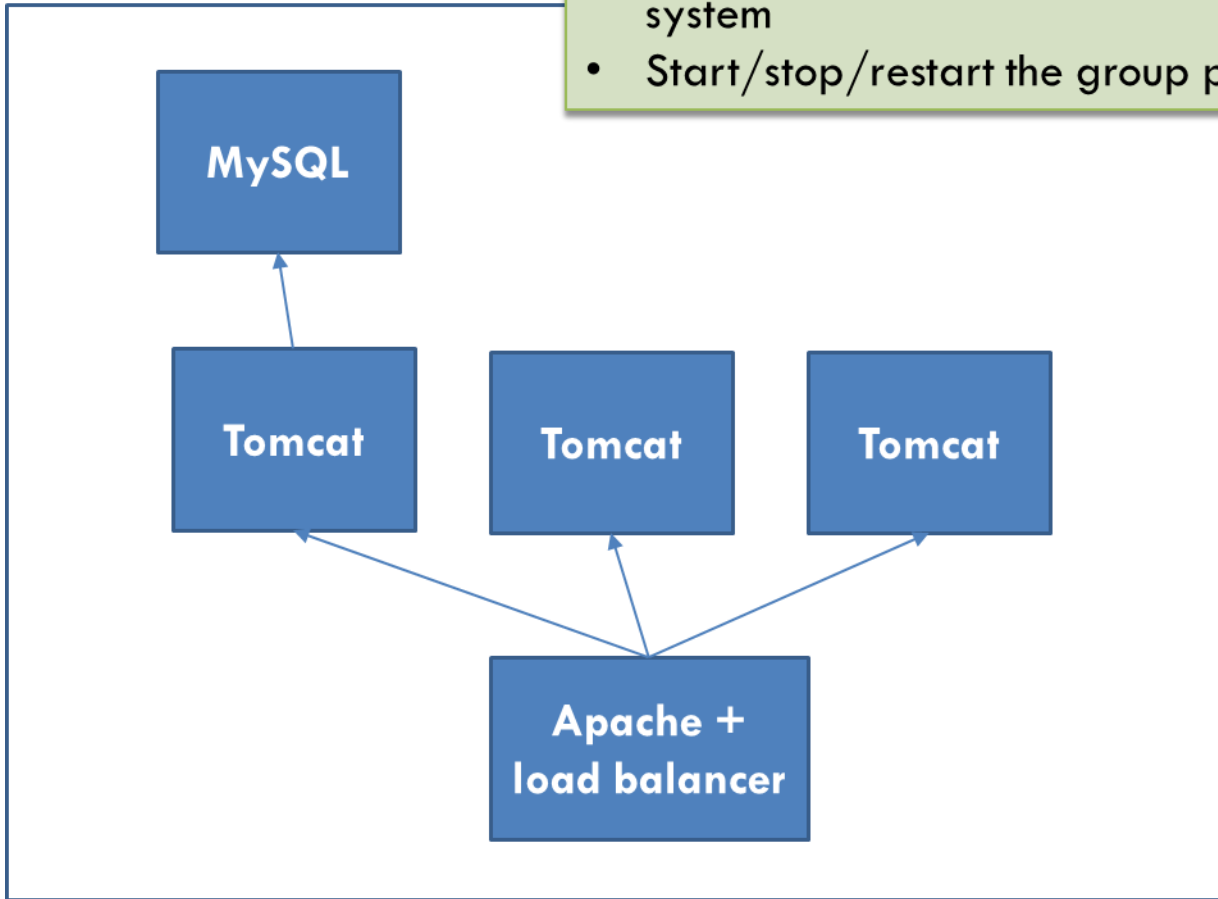
# Gentle Management Tools for dealing with a group of virtual machines

- To define a virtual machine as a list of steps to install & configure it
- To track changes and variants



# Gentle Management Tools for dealing with a group of virtual machines

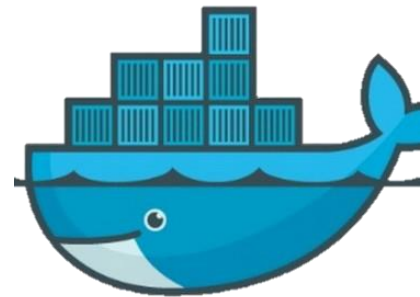
- Define a group of containers as a virtual system
- Start/stop/restart the group properly



- Introduction
- **Docker**
- Docker Swarm



- An **Open-Source** project started on **Marth 2013**
- Based on containers virtual machines
  - ▣ Initially for Linux:
    - `lxc/libcontainer`, namespaces and `cgroups`
  - ▣ Also available for Windows:
    - <https://docs.docker.com/installation/Windows/>
- Docker is a container system to transport applications



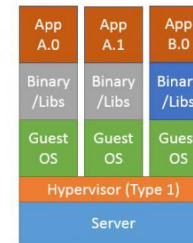
- An **Open-Source** project started on **Marth 2013**
- **Based on containers virtual machines**

- Initially for Linux:

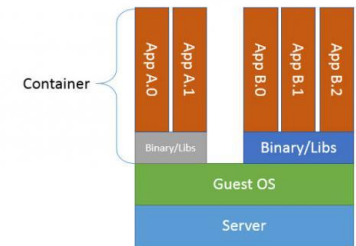
- lxc/libcontainer, namespaces and cgroups

- Also available for Windows:

- <https://docs.docker.com/installation/Windows/>



Traditional Virtualization



Docker

- Docker is a container system to transport applications

□ An **Open-Source** project started on **Marth 2013**

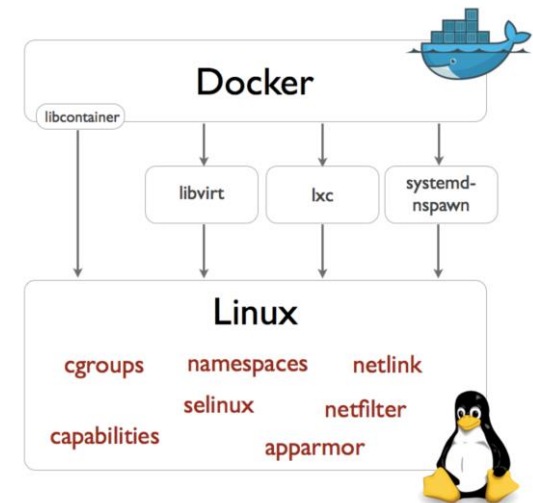
□ **Based on containers virtual machines**

▣ Initially for Linux:

■ lxc/libcontainer, namespaces and cgroups

▣ Also available for Windows:

■ <https://docs.docker.com/installation/Windows/>



□ Docker is a container system to transport applications

□ An **Open-Source** project started on **Marth 2013**

□ **Based on containers virtual machines**

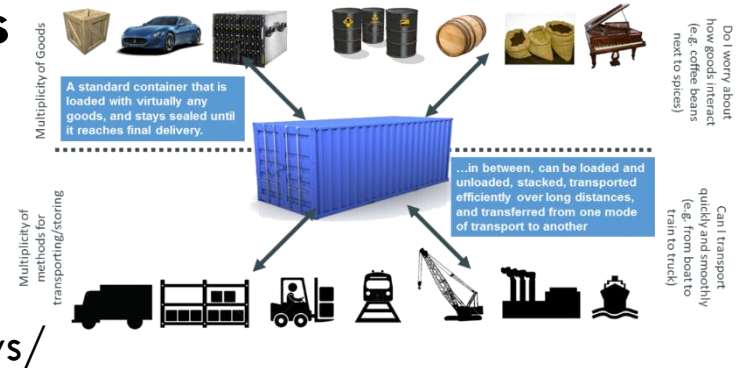
▣ Initially for Linux:

■ `lxc/libcontainer`, namespaces and `cgroups`

▣ Also available for Windows:

■ <https://docs.docker.com/installation/Windows/>

□ **Docker is a container system to transport applications**



## 1. To define a container

```
acaldero@lab01:~# cat > tmp/dockerfile-sshd
FROM debian:latest
MAINTAINER "Kirill Müller" <krmlr+docker@mailbox.org>

RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get -y install openssh-server sudo
RUN mkdir -p /var/run/sshd && sed -i "s/UsePrivilegeSeparation.* /UsePrivilegeSeparation no/g"
/etc/ssh/sshd_config \
&& sed -i 's/PermitRootLogin without-password/PermitRootLogin yes/' /etc/ssh/sshd_config \
&& touch /root/.Xauthority && true

RUN useradd docker && passwd -d docker \
&& mkdir /home/docker && chown docker:docker /home/docker \
&& addgroup docker staff && addgroup docker sudo && true

EXPOSE 22
CMD ["/usr/sbin/sshd","-D"]
<EOF>
```

<https://github.com/krmlr/debian-ssh>

## 2. To pack the associated container image

```
acaldero@lab01:~# docker build -t lab_sshd -f tmp/dockerfile-sshd tmp
```

```
Sending build context to Docker daemon 46.5 MB
```

```
Step 1 : FROM debian:latest
```

```
---> a604b236bcde
```

```
Step 2 : MAINTAINER "Kirill Müller" <krmlr+docker@mailbox.org>
```

```
---> Running in 2a2f0b40f2da
```

```
---> 1099448637af
```

```
Removing intermediate container 2a2f0b40f2da
```

```
...
```

## 3. To execute the container from the image

```
acaldero@lab01:~# docker run -d -p 2222:22 --name test_sshd lab_sshd
```

```
acaldero@lab01:~# ssh docker@localhost -p 2222
```

```
# The password is ``docker``.
```

```
$
```

## □ To stop a container

```
acaldero@lab01:~# docker stop <docker id>
```

## □ To start a container

```
acaldero@lab01:~# docker start <docker id>
```

## □ To execute a bash shell within a container

```
acaldero@lab01:~# docker exec -it <docker id> bash  
acaldero@<docker id>:~/# exit
```

- To list running containers

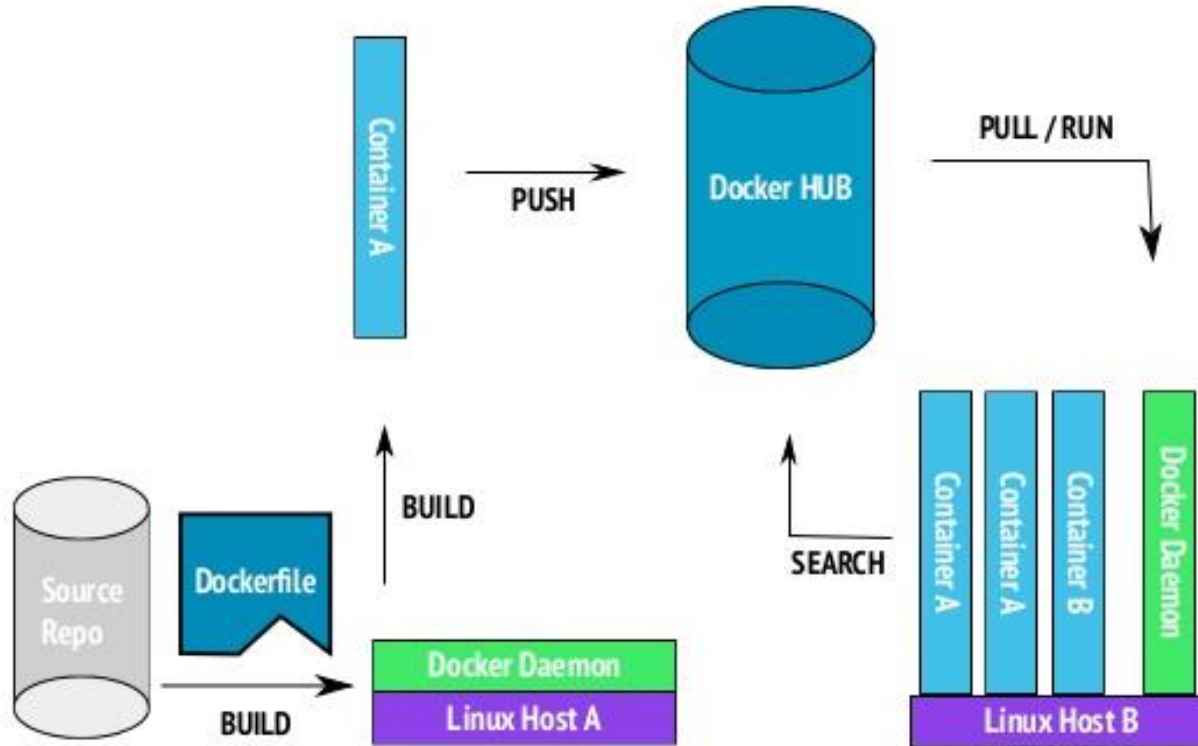
```
acaldero@lab01:~# docker ps
```

- To list all containers

```
acaldero@lab01:~# docker ps -a
```



# Dockers' Build+Ship+Run...



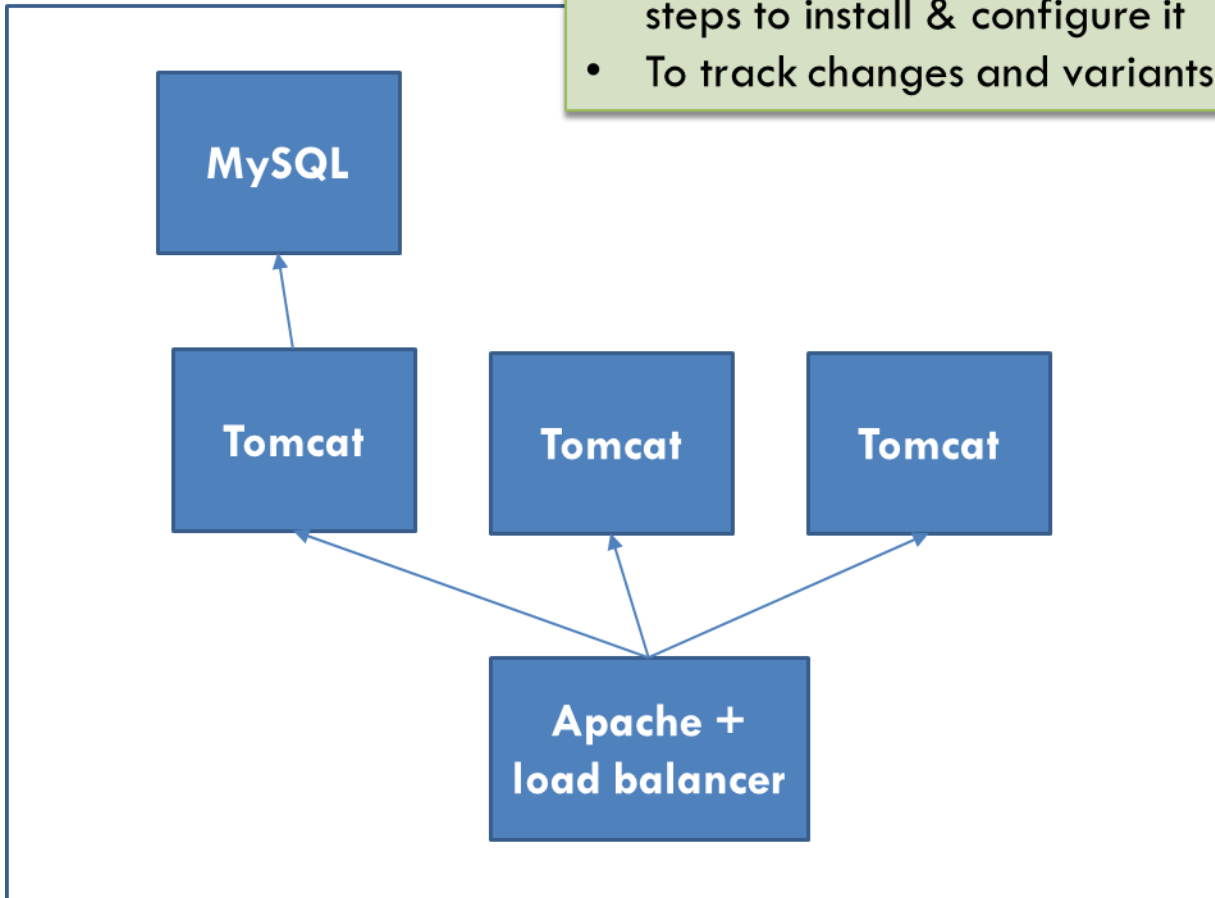
- Download the most recent image

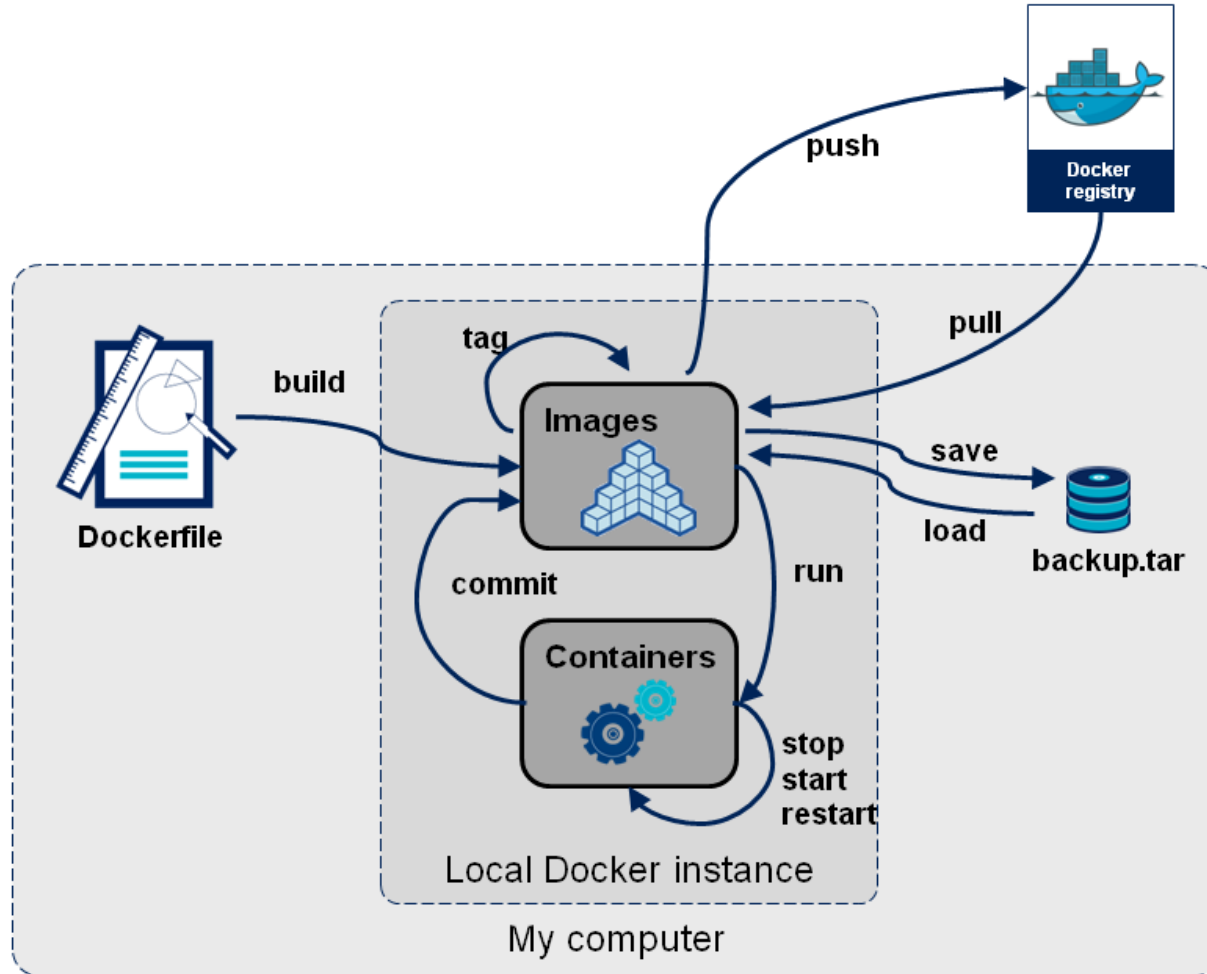
```
acaldero@lab01:~# docker pull sameersbn/bind:latest
```

- To create a container from an image

```
acaldero@lab01:~# mkdir -p /home/docker/bind  
acaldero@lab01:~# docker run --name bind -d \  
    --env ROOT_PASSWORD=xxxyyyzzz \  
    --publish 53:53/udp \  
    --volume /home/docker/bind:/data \  
    sameersbn/bind:latest
```

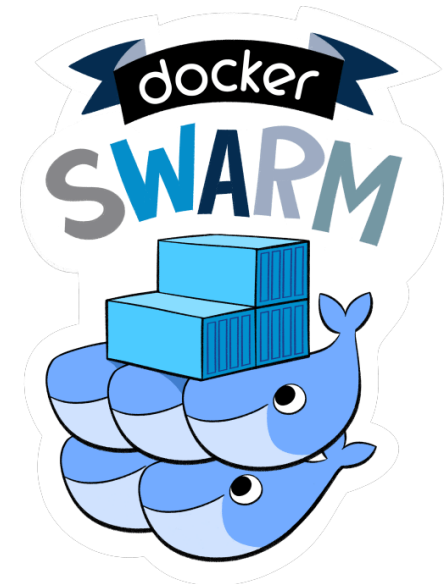
- To define a virtual machine as a list of steps to install & configure it
- To track changes and variants



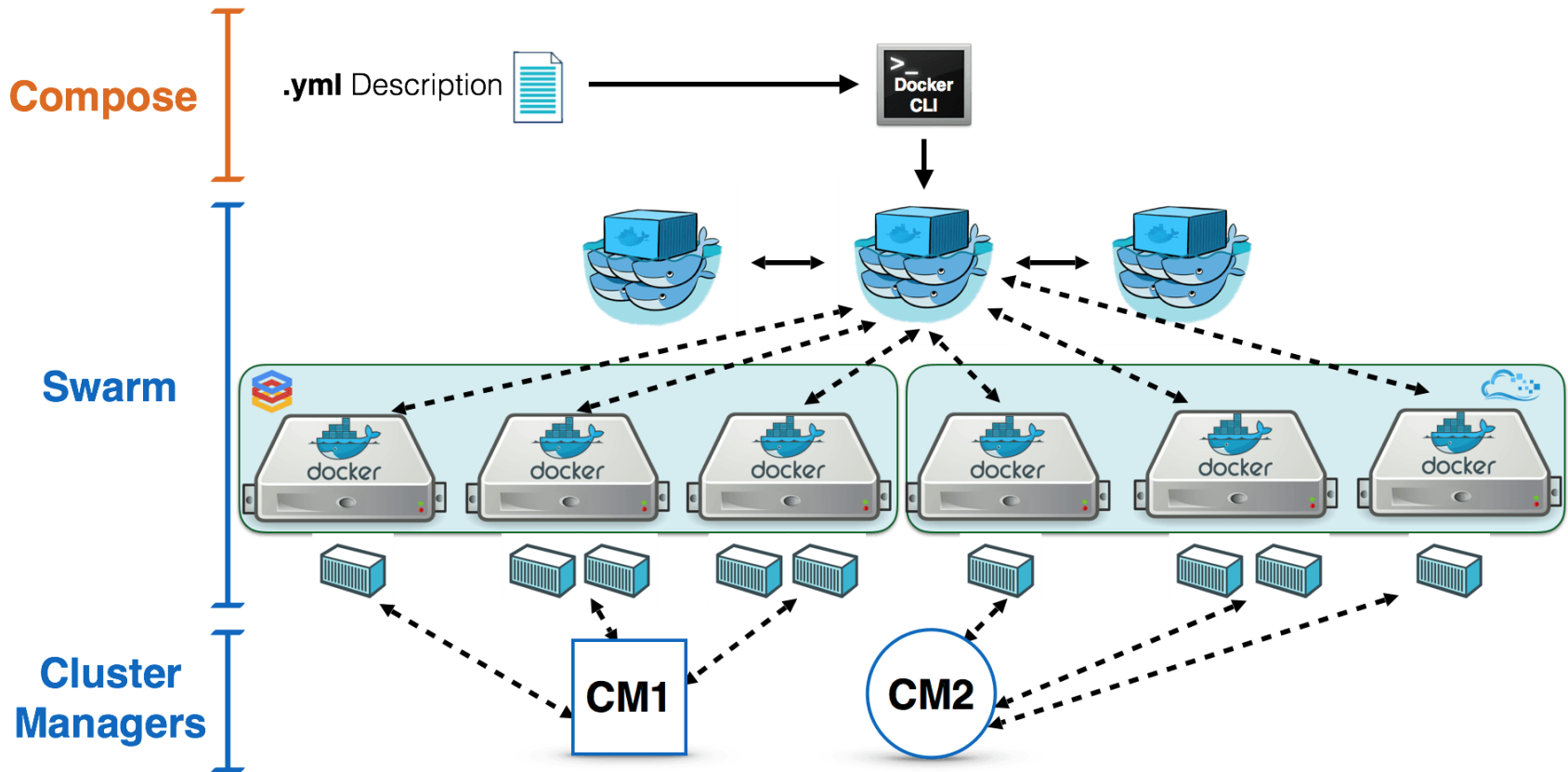


- Introduction
- Docker
- **Docker Swarm**

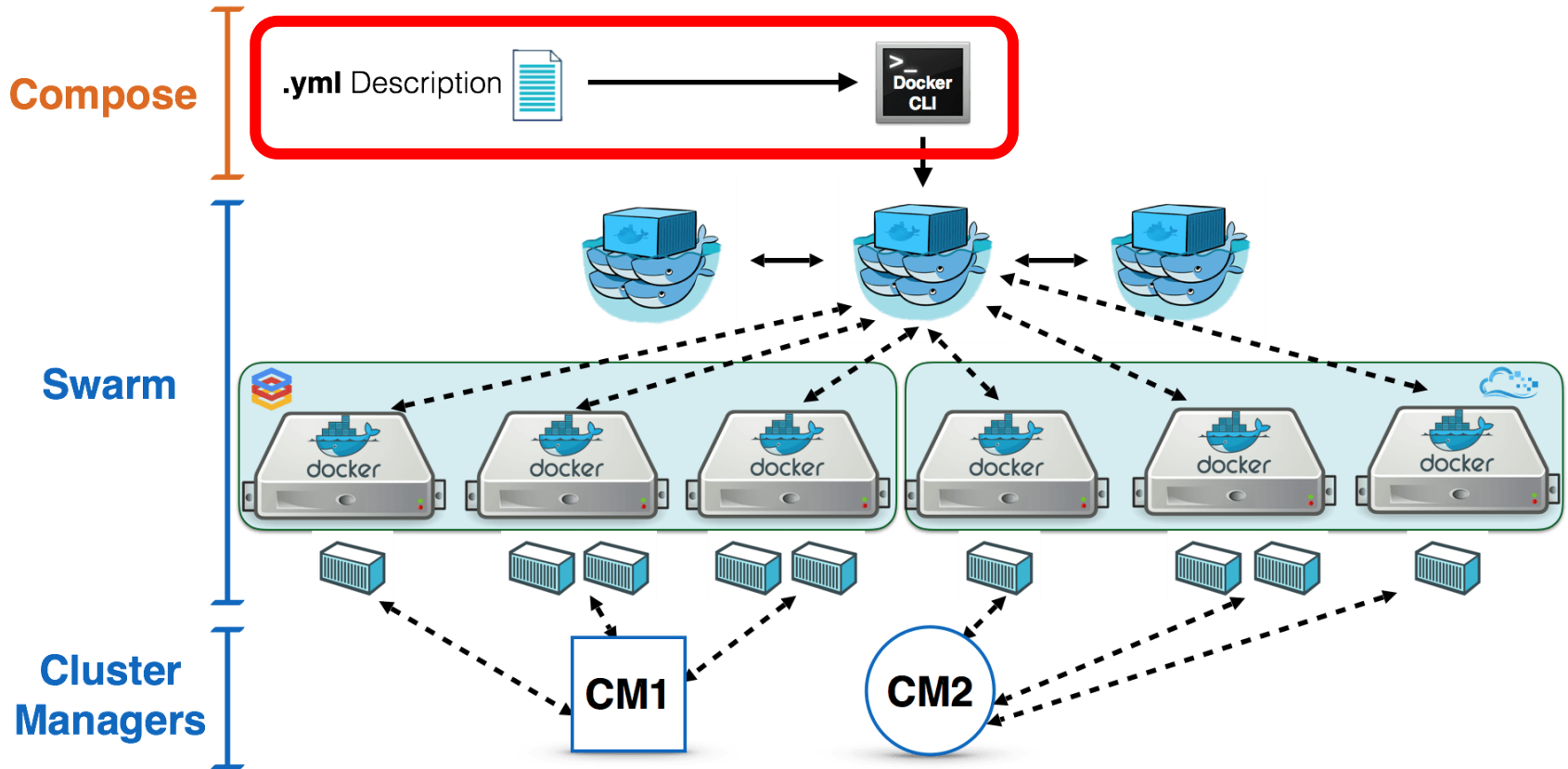
- Docker Swarm is a **clustering + scheduling tool for Docker** containers.
- With Swarm, IT administrators and developers can establish and **manage a cluster of Docker nodes as a single virtual system.**



<http://searchitoperations.techtarget.com/definition/Docker-Swarm>



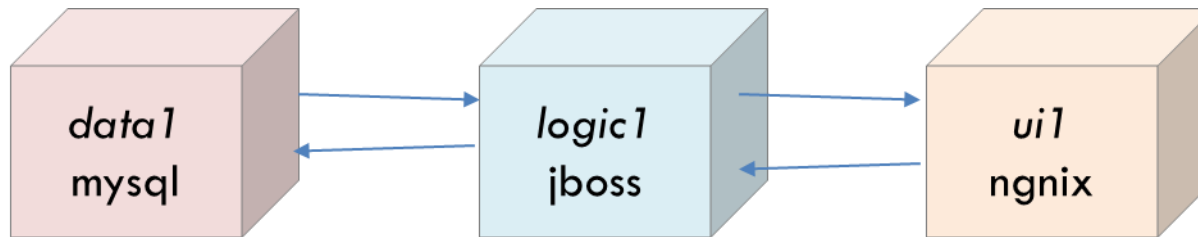
<https://blog.docker.com/2015/11/deploy-manage-cluster-docker-swarm/>



<https://blog.docker.com/2015/11/deploy-manage-cluster-docker-swarm/>



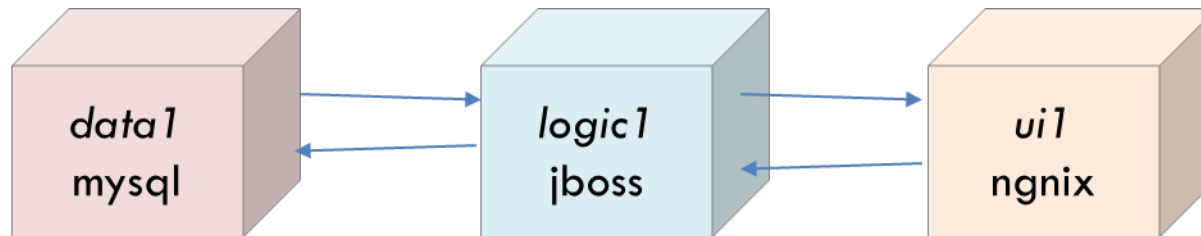
## □ Example of docker deployment:



```
acaldero@lab01:~# docker run -d --name data1 -p 3306:3306 mysql
acaldero@lab01:~# docker run -d --name logic1 -p 8080:8080 -l data1:bbdd jboss/wildfly
acaldero@lab01:~# docker run -d --name ui1 -p 80:80 -l logic1:app nginx
```

<https://www.adictosaltrabajo.com/tutoriales/docker-compose-machine-y-swarm/>

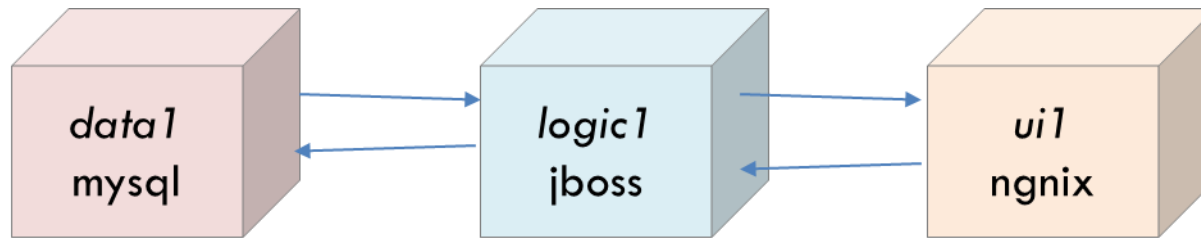
## □ Example of docker deployment:



```
acaldero@lab01:~# cat > docker-composer.yml
data1:
  image: mysql
  expose:
    - 3306
logic1:
  image: jboss/wildfly
  links:
    - data1:redis
  expose:
    - 8080
  volumes:
    - ./webapps:/opt/jboss/wildfly/standalone/deployments/
web:
  build: ./
  links:
    - logic1:app1
  ports:
    - 80:80
```

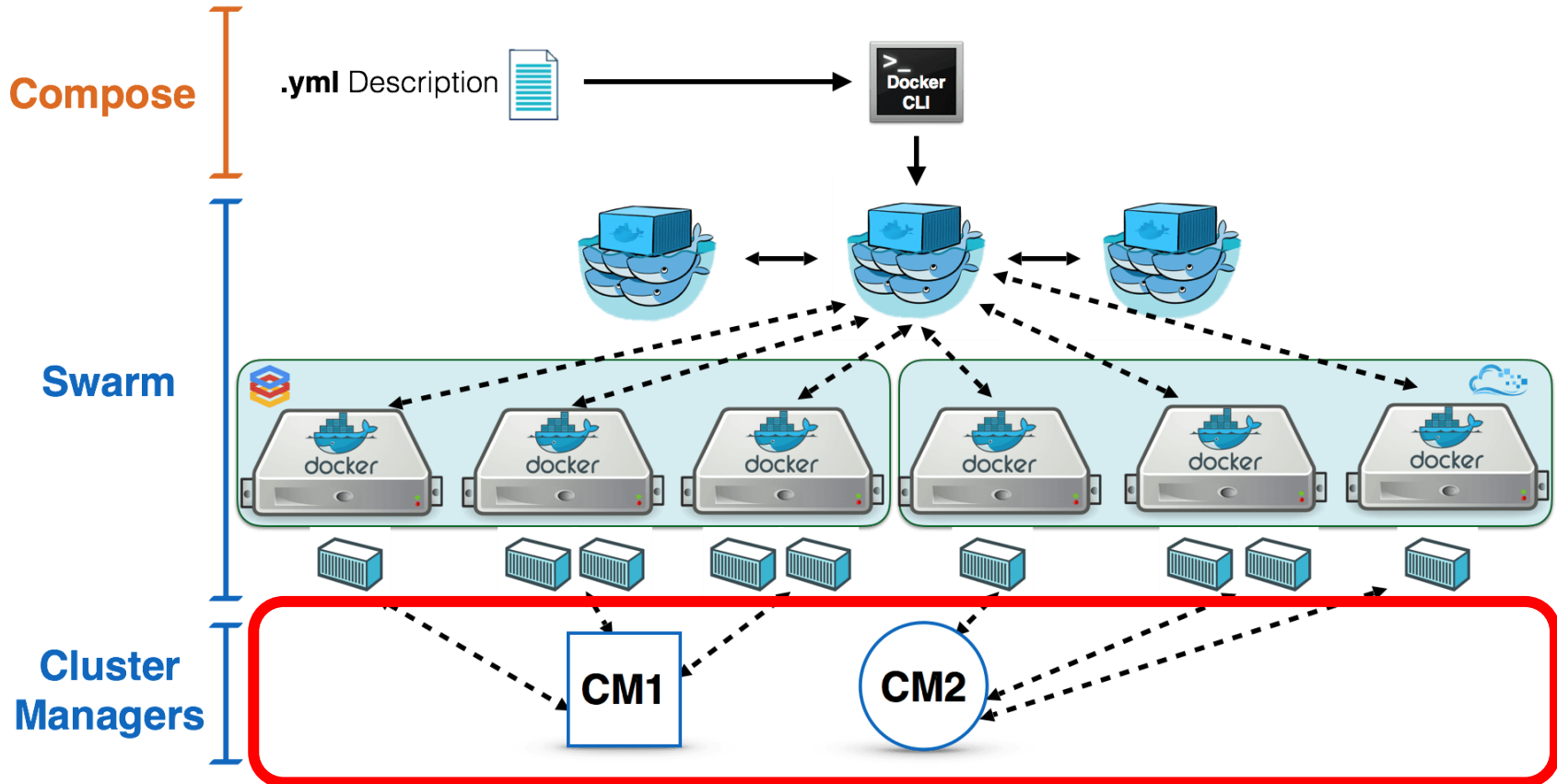
<https://www.adictosaltrabajo.com/tutoriales/docker-compose-machine-y-swarm/>

## □ Example of docker deployment:



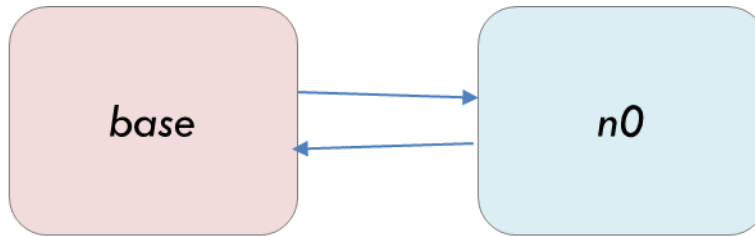
```
acaldero@lab01:~# docker-compose up docker-composer.yml
```

<https://www.adictosaltrabajo.com/tutoriales/docker-compose-machine-y-swarm/>



<https://blog.docker.com/2015/11/deploy-manage-cluster-docker-swarm/>

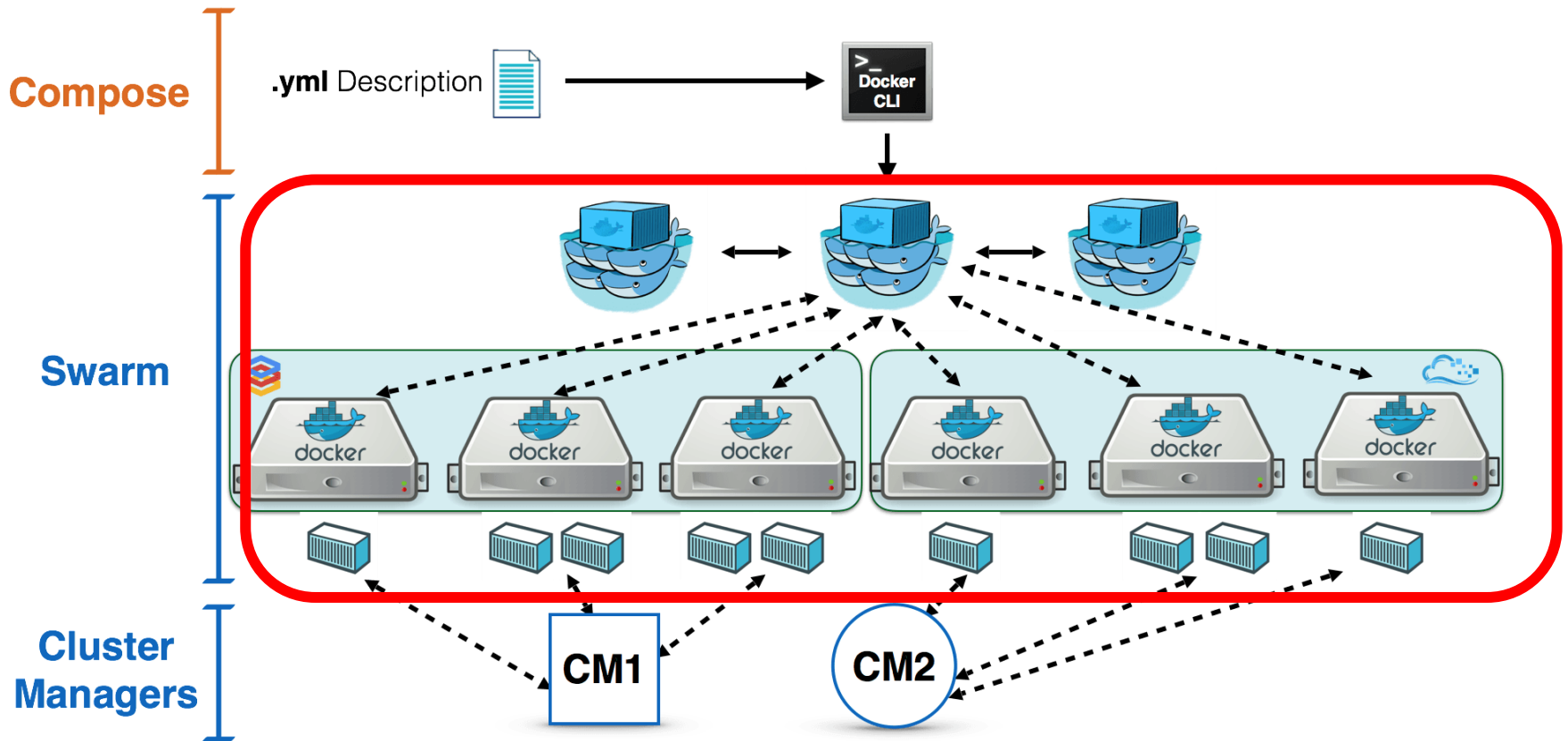
## □ Example docker machine:



Amazon Web Services  
Microsoft Azure  
Digital Ocean  
Exoscale  
Google Compute Engine  
Generic  
Microsoft Hyper-V  
OpenStack  
Rackspace  
IBM Softlayer  
Oracle VirtualBox  
VMware vCloud Air  
VMware Fusion  
VMware vSphere

```
acaldero@lab01:~# docker-machine create --driver virtualbox n0
acaldero@lab01:~# docker-machine ls
acaldero@lab01:~# eval $(docker-machine env dev)
```

<https://www.adictosaltrabajo.com/tutoriales/docker-compose-machine-y-swarm/>



<https://blog.docker.com/2015/11/deploy-manage-cluster-docker-swarm/>

- To start swarm at manager, and get token id.

```
acaldero@lab01:~# sid=$(docker run --rm swarm create)
acaldero@lab01:~# echo $sid
```

- To create master and nodes using token id.

```
acaldero@lab01:~# docker-machine create -d virtualbox --swarm --swarm-master \
--swarm-discovery token://$sid master
acaldero@lab01:~# docker-machine create -d virtualbox --swarm --swarm-discovery token://$sid n1
acaldero@lab01:~# docker-machine create -d virtualbox --swarm --swarm-discovery token://$sid n2
```

- To introduce the master to the manager

```
acaldero@lab01:~# eval "$(docker-machine env --swarm master)"
acaldero@lab01:~# docker info
```



# DOCKER, DOCKER SWARM