



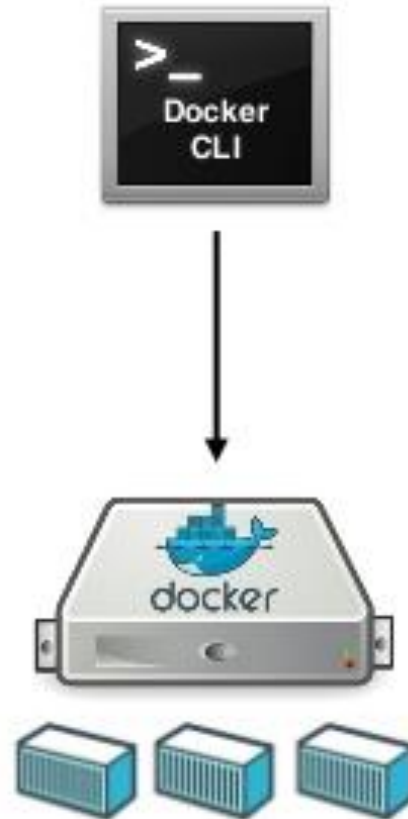
Persistent

Docker Orchestration: Introduction to Docker Swarm

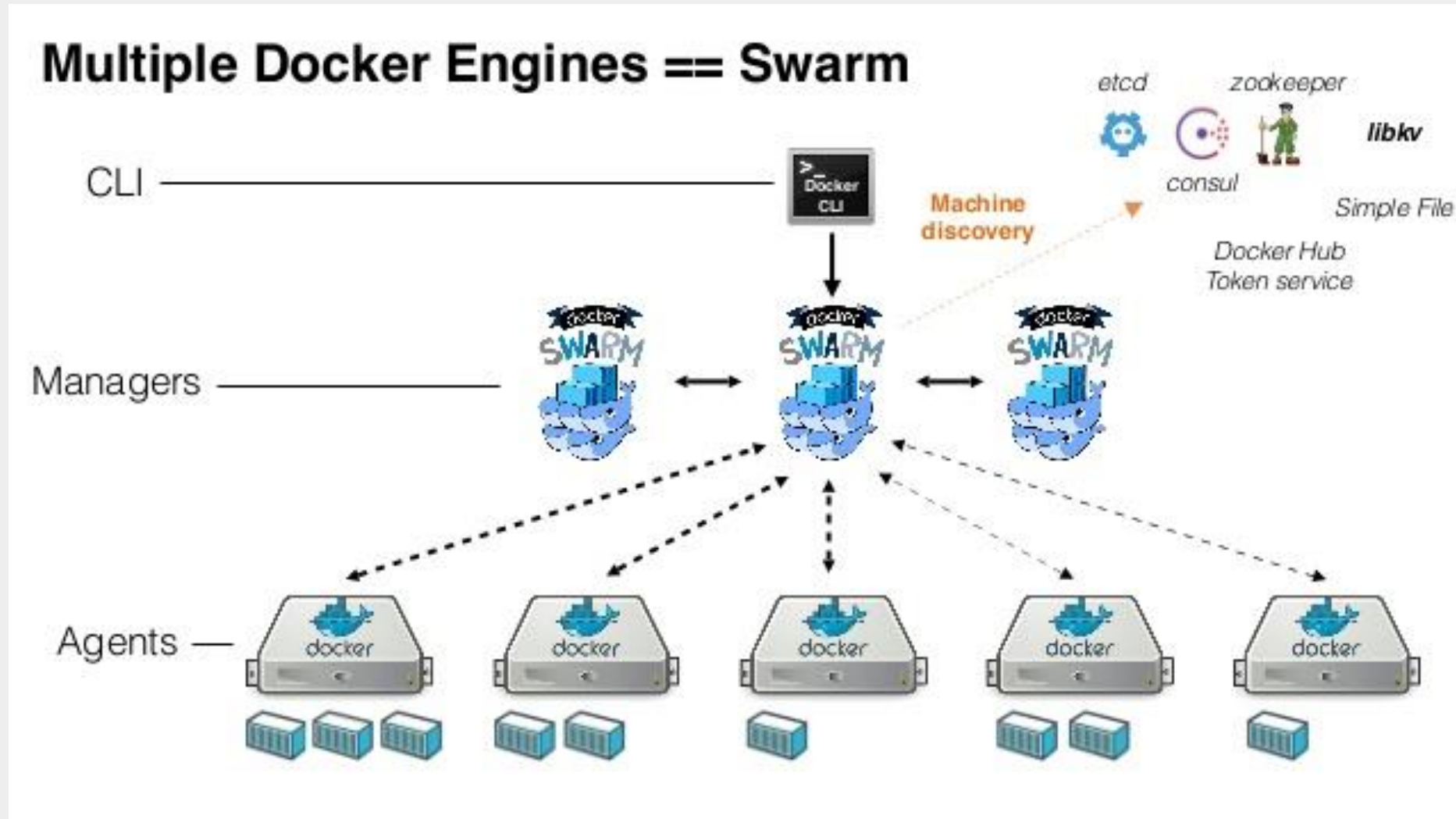


Single Docker engine

Single docker engine

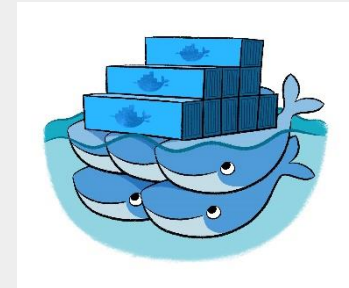


Multiple Docker engines using Swarm



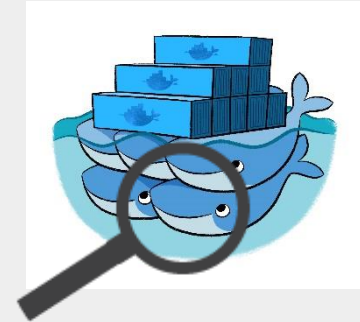
Docker Swarm Overview

- Docker Swarm is a native clustering for Docker.
- Docker Swarm is a tool that clusters Docker hosts and Schedules containers
- It turns a pool of Docker hosts into a single, virtual Docker host
- Docker swarm is part of Docker version 1.12



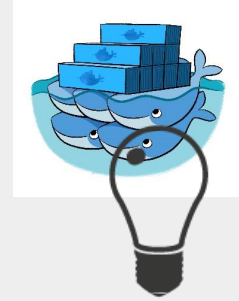
Docker Swarm Features

- Decentralize Cluster management integrated with Docker Engine
- Decentralized design - you can deploy both kinds of nodes, managers and workers, using the Docker Engine.
- Scaling – For each service For each service you can declare the number of tasks you want to run
- Service discovery, load balancing.



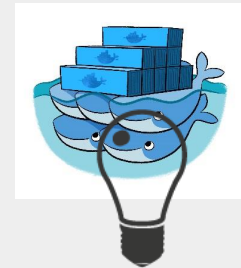
Docker Swarm Key concepts

- Swarm mode – Docker engines participating in a cluster are running in swarm mode. You enable swarm mode for the Engine by either initializing a swarm or joining an existing swarm.
- Node - A node is an instance of the Docker Engine participating in the swarm.
- Manager Node - The manager node dispatches units of work called tasks to worker nodes.
- Manager nodes also perform the orchestration and cluster management functions required to maintain the desired state of the swarm.
- Worker nodes receive and execute tasks dispatched from manager nodes.

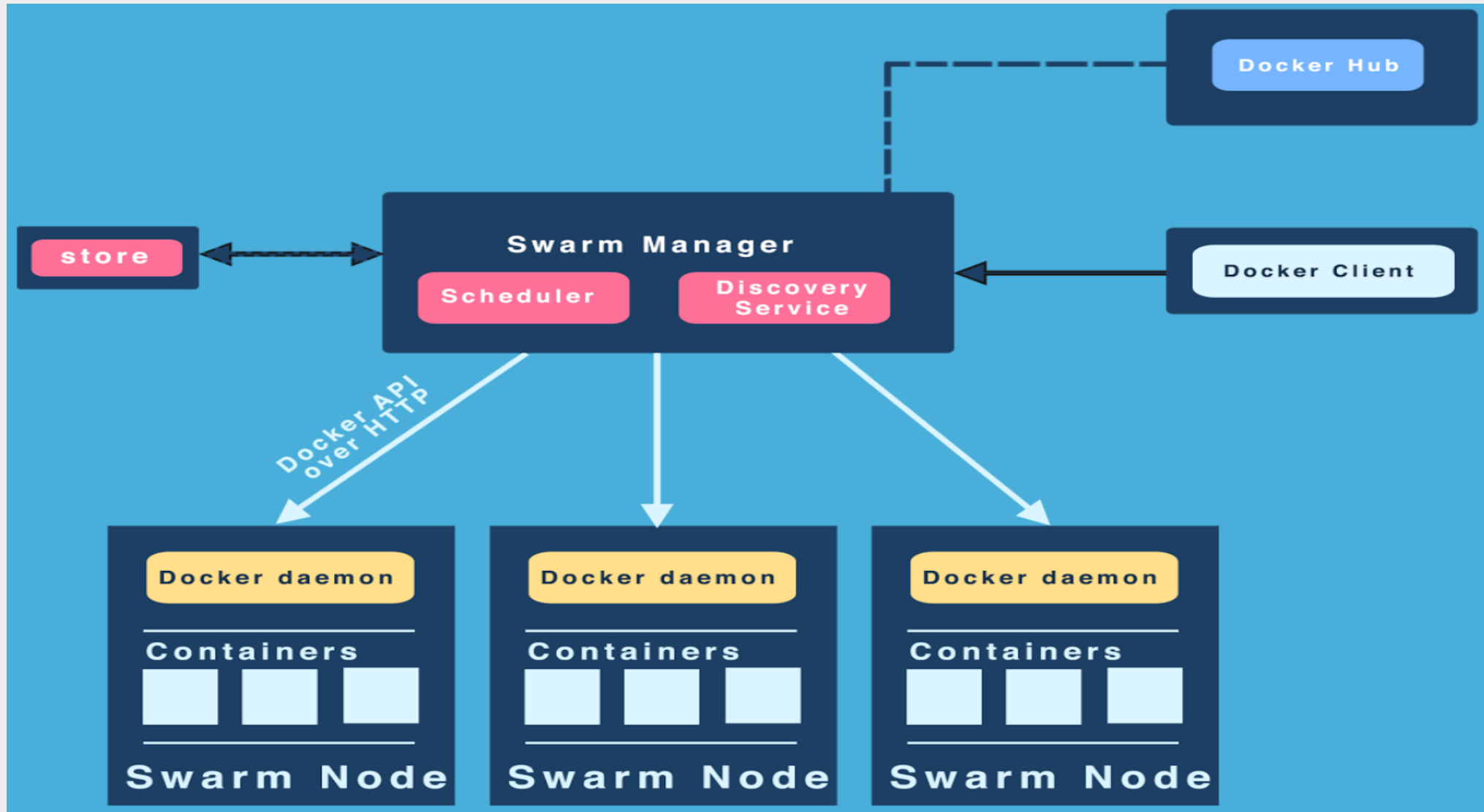


Docker Swarm Key concepts...

- A service is the definition of the tasks to execute on the worker nodes.
- Services specify the container image and commands to execute inside running containers.
- In the replicated services model, the swarm manager distributes a specific number of replica tasks among the nodes.
- A task carries a Docker container and the commands to run inside the container.
- Load balancing - The swarm manager uses ingress load balancing to expose the services you want to make available externally to the swarm.

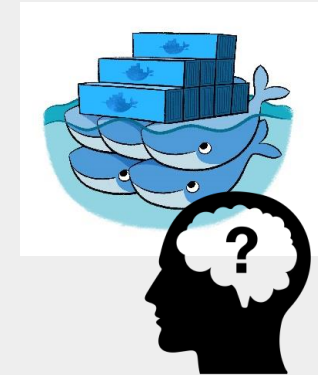


Docker swarm Architecture



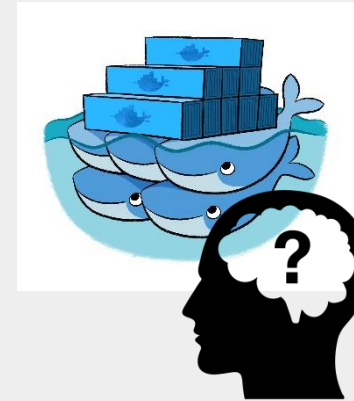
Understand Swarm cluster creation

- The first step to creating a Swarm cluster on your network is to pull the Docker Swarm image.
- Using Docker, you configure the Swarm manager and all the nodes to run Docker Swarm.
- Using Docker Machine, you can quickly install a Docker Swarm on cloud providers or inside your own data center.
- Here is the link for creating Docker Swarm
 - <https://docs.docker.com/swarm/install-w-machine/>



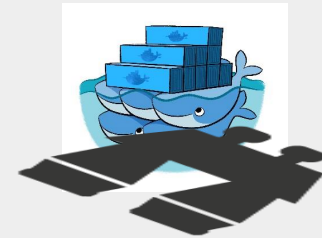
Understand Swarm cluster creation...

- Here are the high level steps for creating Docker Swarm cluster.
- Install Docker Toolbox , the toolbox installs a handful of tools on your local Windows or Mac OS X computer.
- Create three VMs running Docker Engine
- Create a Swarm discovery token
- Create the Swarm manager and nodes
- Manage your Swarm



Docker Swarm discovery

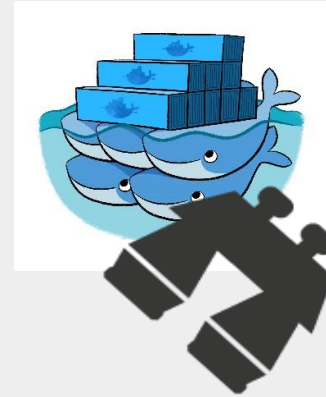
- Helps Swarm manager discover nodes.
- Three main functions –
 - Register a new node
 - Watch – callback a method for a swarm manager when a new node is added.
 - Fetch list of entries.



Docker Swarm discovery...

Swarm discovery can be based on following approaches :-

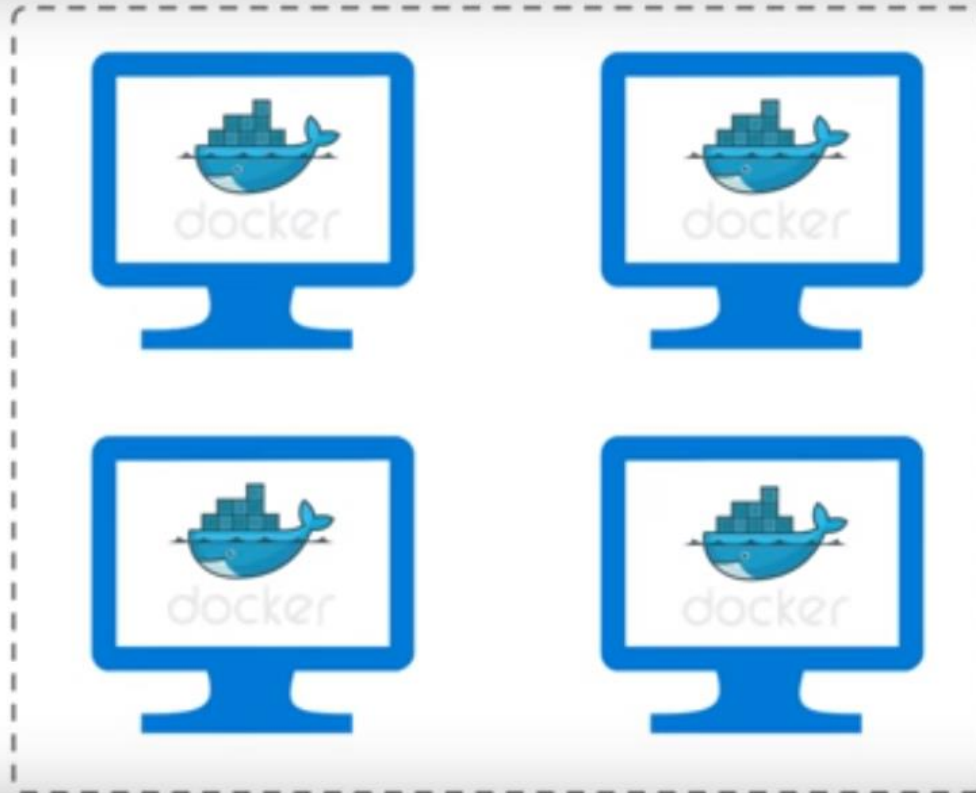
- Token based.
- etcd based.
- Zookeeper based.
- File based.
- Consul based



Docker swarm mode example

<http://stackoverflow.com/questions/38019491/docker-upgrade-from-v1-11-to-v1-12-using-apt-get-on-ubuntu-14-04-4-lts>

- Managers:
 - swarm-00
- Workers:
 - swarm-01
 - swarm-02
 - swarm-03



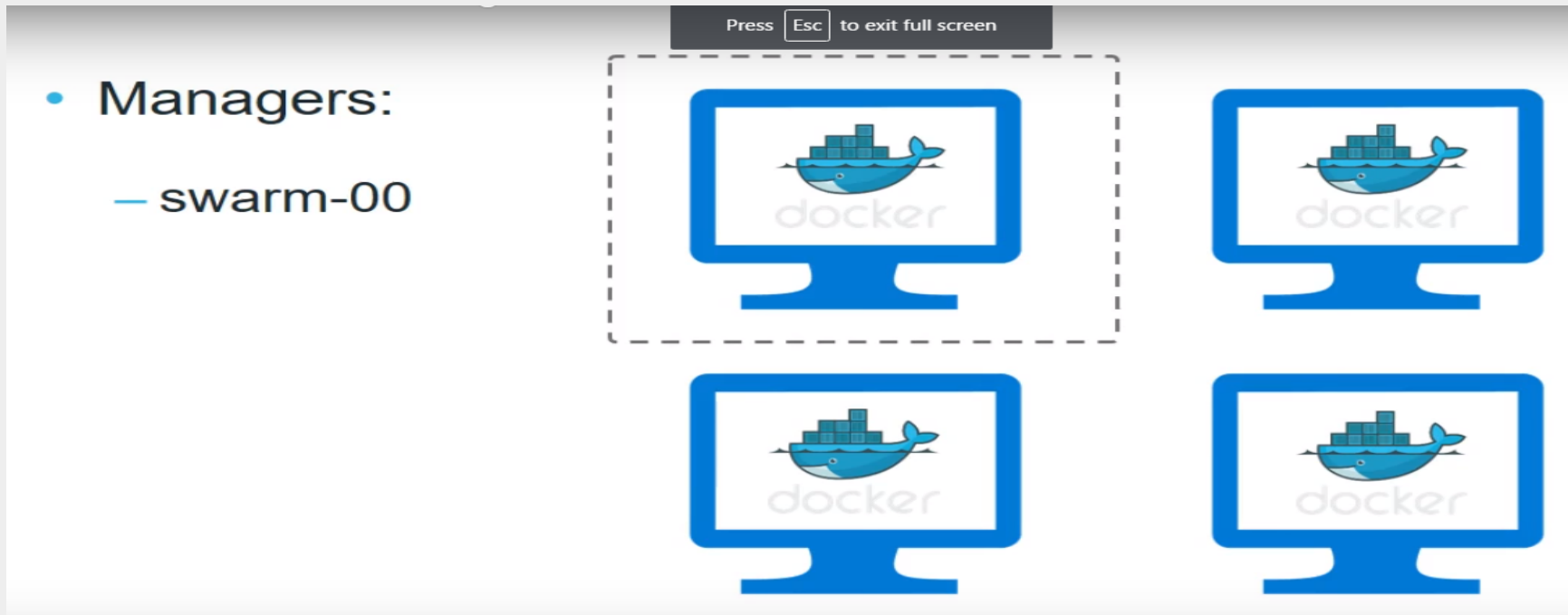


SWARM MODE PREREQUISITES

- Some machines running Docker 1.12
- Which are in the same subnet
- With ports **2377**, **4789** and **7946** open

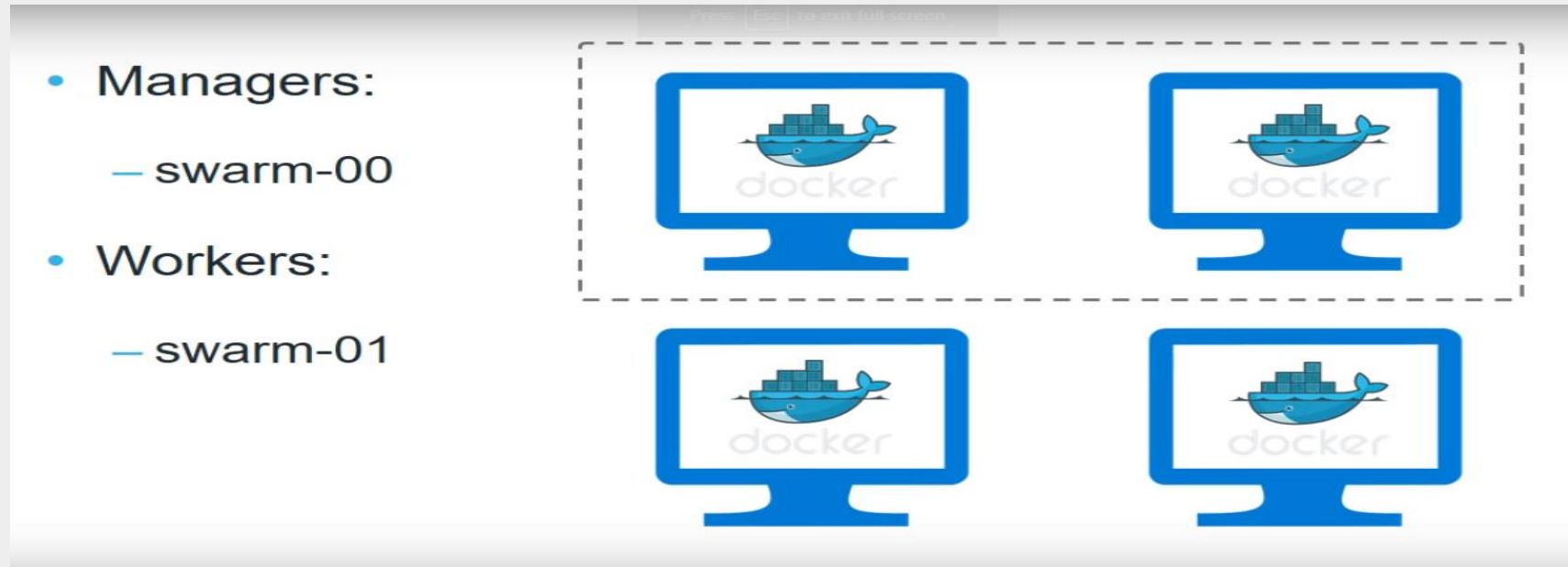
Docker Swarm creation

- Run the below command to init the cluster, on swarm-00 instance.
 - `docker swarm init --listen-addr 172.17.0.1:2377`
- Run the below command to list the swarm nodes,
 - `sudo docker node list`



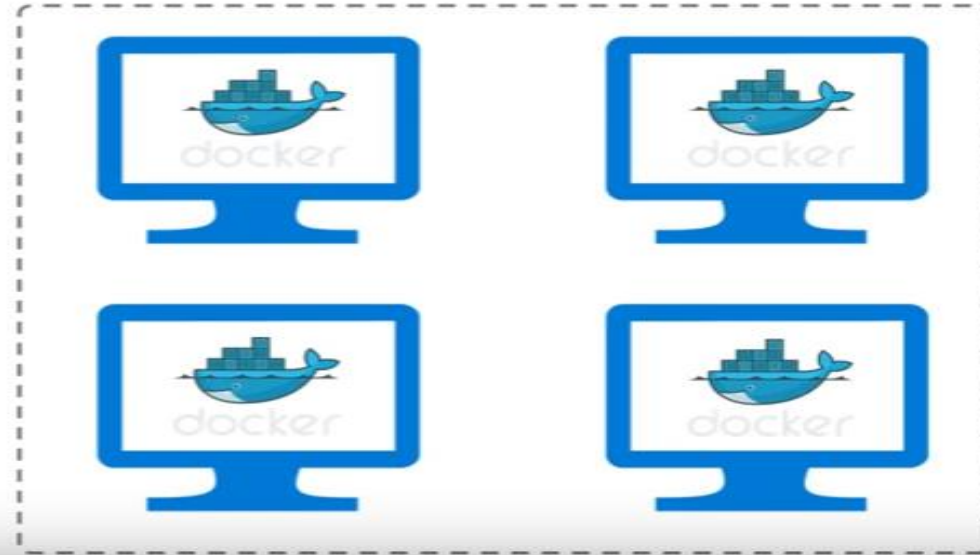
Docker Swarm workers creation

- Run the below command to run a Swarm worker.
 - `sudo docker swarm join 172.31.16.130:2377`
- Let's list the swarm nodes.
 - `sudo docker node list`
- Now you will see two nodes, one manager and one worker.



Docker Swarm managers and workers

- Managers:
 - swarm-00
- Workers:
 - swarm-01
 - swarm-02
 - swarm-03

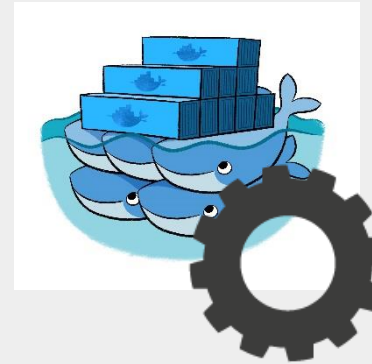


```
ubuntu@ip-172-31-16-130:~$ sudo docker node list
```

ID	HOSTNAME	MEMBERSHIP	STATUS	AVAILABILITY	MANAGER
2hr957awc0fbejy5pm7wuwi8k	ip-172-31-23-161	Accepted	Ready	Active	
39cgbxrt60ljhtxu69kxp940u *	ip-172-31-16-130	Accepted	Ready	Active	Leader
4sndiijscn6ehez2mt4mzh6en	ip-172-31-26-54	Accepted	Ready	Active	
93w8yfgkvu2egs3881i4mfj4l	ip-172-31-17-186	Accepted	Ready	Active	

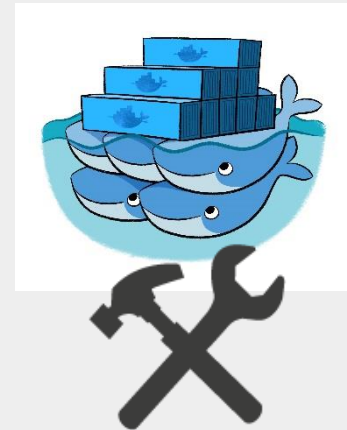
Running Docker swarm service

- Let's run the Nginx service on the Docker container.
 - `sudo docker service create 80:80 sixeyed/docker-swarm-walkthroughsixeyed/docker-swarm-walkthrough`
- List the docker service running
 - `sudo docker service ls`
- Let's list the tasks
 - `sudo docker service tasks pensive_poitras`
- Create replicas on multiple nodes.
 - `sudo docker service update --replicas 10 trusting_wilson`

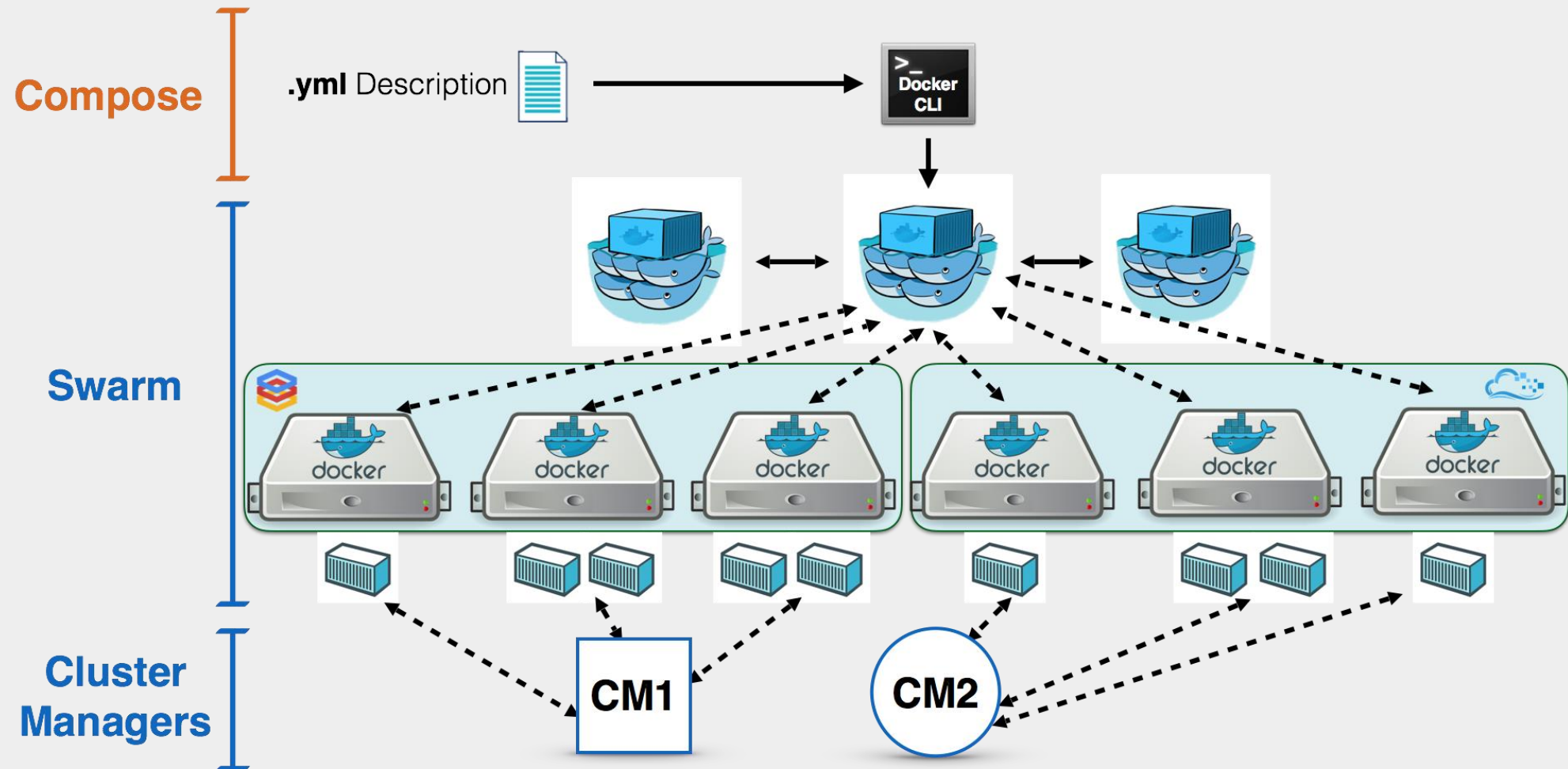


Planning Docker Swarm in Production.

- Here is the link which provides details about Docker Swarm in production
 - <https://docs.docker.com/swarm/plan-for-production/>
- Here are the main aspects for running Docker swarm in production –
- Security
 - Configure Swarm for TLS
 - Network access control
- High Availability (HA)
- Performance



Docker swarm, machine and Compose



Reference Material : Websites & Blogs

- <https://www.youtube.com/watch?v=KC4Ad1DS8xU>
- <https://docs.docker.com/compose/wordpress/>
- <https://www.docker.com/>
- <https://training.docker.com/self-paced-training>
- <https://www.youtube.com/watch?v=Q5POuMHxW-0>
- <https://www.digitalocean.com/community/tutorials/how-to-run-nginx-in-a-docker-container-on-ubuntu-14-04>

Docker up and Running by Karl Matthias and Sean kane

The Docker Book by James Turnball

Key Contacts

Docker Interactive

Dattatray Kulkarni

dattatray_kulkarni@persistent.co.in

Asif Immanad

asif_immanad@persistent.co.in

Vice President

Shubhangi Kelkar

shubhangi_kelkar@persistent.co.in



Persistent

Thank you!

Persistent University

