



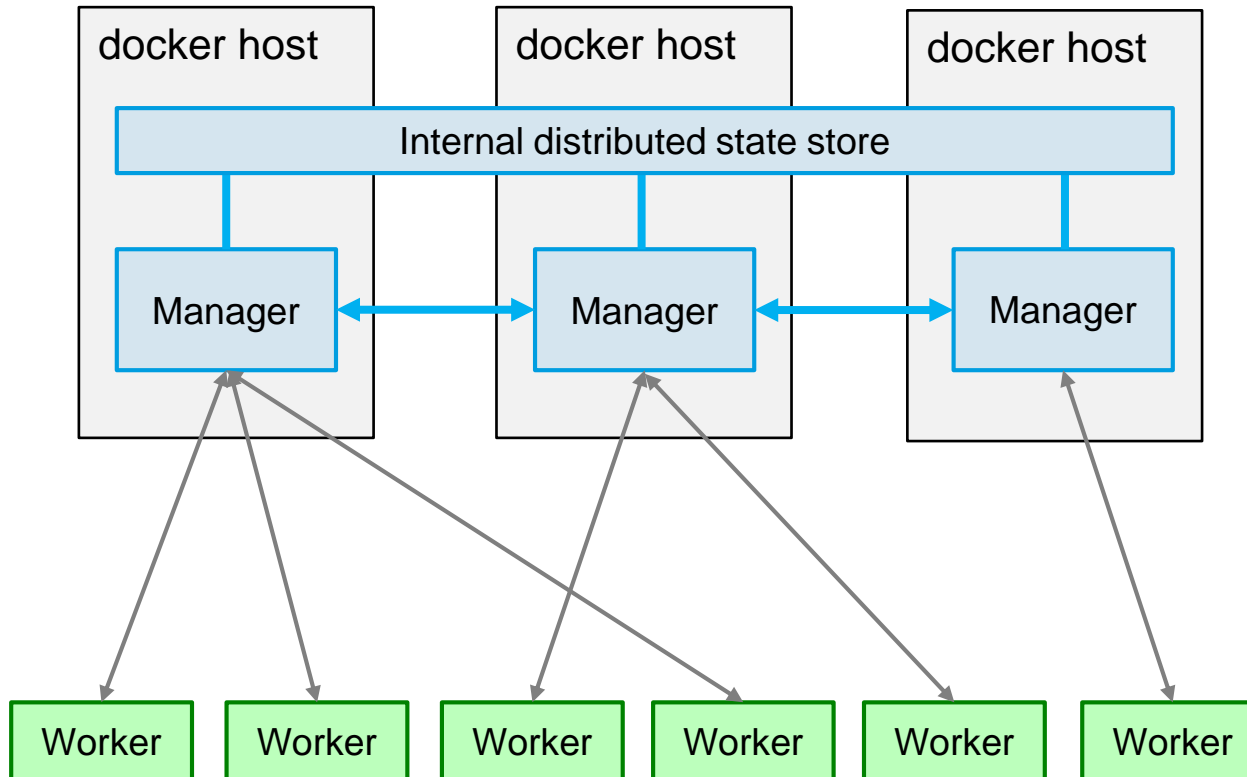
## Docker and Containers

### Module 7: Docker Swarm

- 1. Short introduction.**
- 2. Demo: The docklab.de swarm.**
- 3. Demo: The Raft Consensus Algorithm.**



## Swarm Mode Architecture



compare: <https://blog.docker.com/2016/07/docker-built-in-orchestration-ready-for-production-docker-1-12-goes-ga/>



## Docker Swarm Feature Highlights

- **Cluster management integrated with Docker Engine.** No additional orchestration software to create or manage a swarm.
- **Decentralized design:** You can deploy both kinds of nodes, managers and workers, using the Docker Engine.
- **Declarative service model:** Docker Engine uses a declarative approach to let you define the desired state of the various services in your application stack.
- **Scaling:** For each service, you can declare the number of tasks you want to run. When you **scale up or down**, the swarm manager automatically adapts by adding or removing tasks to maintain the desired state.
- **Desired state reconciliation:** The swarm manager node **constantly monitors the cluster state** and reconciles any differences between the actual state and your expressed desired state.
- **Multi-host networking:** You can specify an **secure overlay network** for your services.

From: docker docs: <https://docs.docker.com/engine/swarm/>



## Docker Swarm Feature Highlights

- **Service discovery:** Swarm manager nodes assign each service in the swarm a unique DNS name and **load balances running containers.**
- **Load balancing:** You can expose the ports for services to an external load balancer. Internally, the swarm lets you specify how to distribute service containers between nodes.
- **Secure by default:** Each node in the swarm enforces **TLS mutual authentication and encryption** to secure communications between itself and all other nodes.
- **Rolling updates:** At rollout time you can **apply service updates to nodes incrementally.** The swarm manager lets you control the delay between service deployment to different sets of nodes. If anything goes wrong, you can roll-back a task to a previous version of the service.

From: docker docs: <https://docs.docker.com/engine/swarm/>

