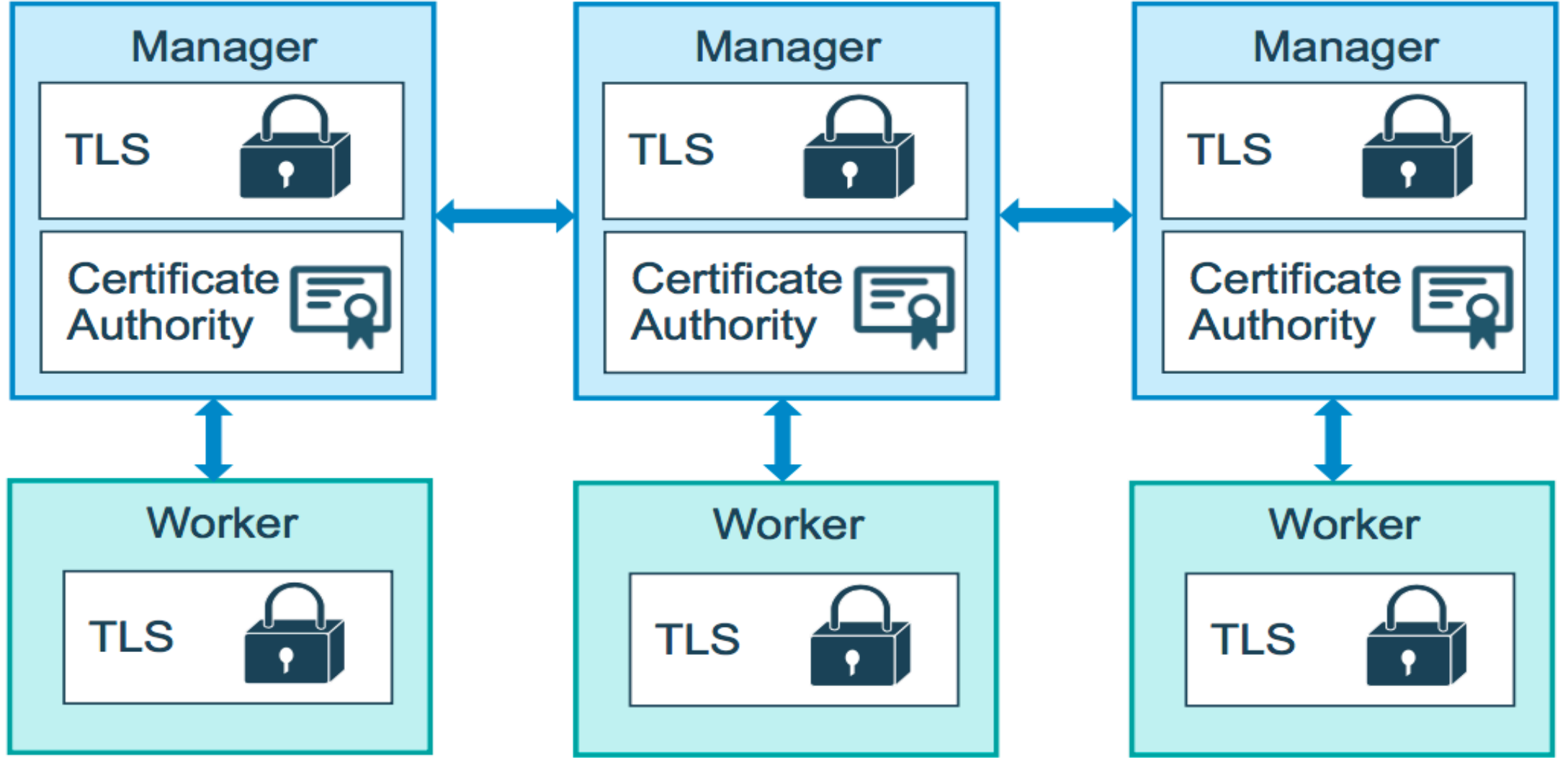


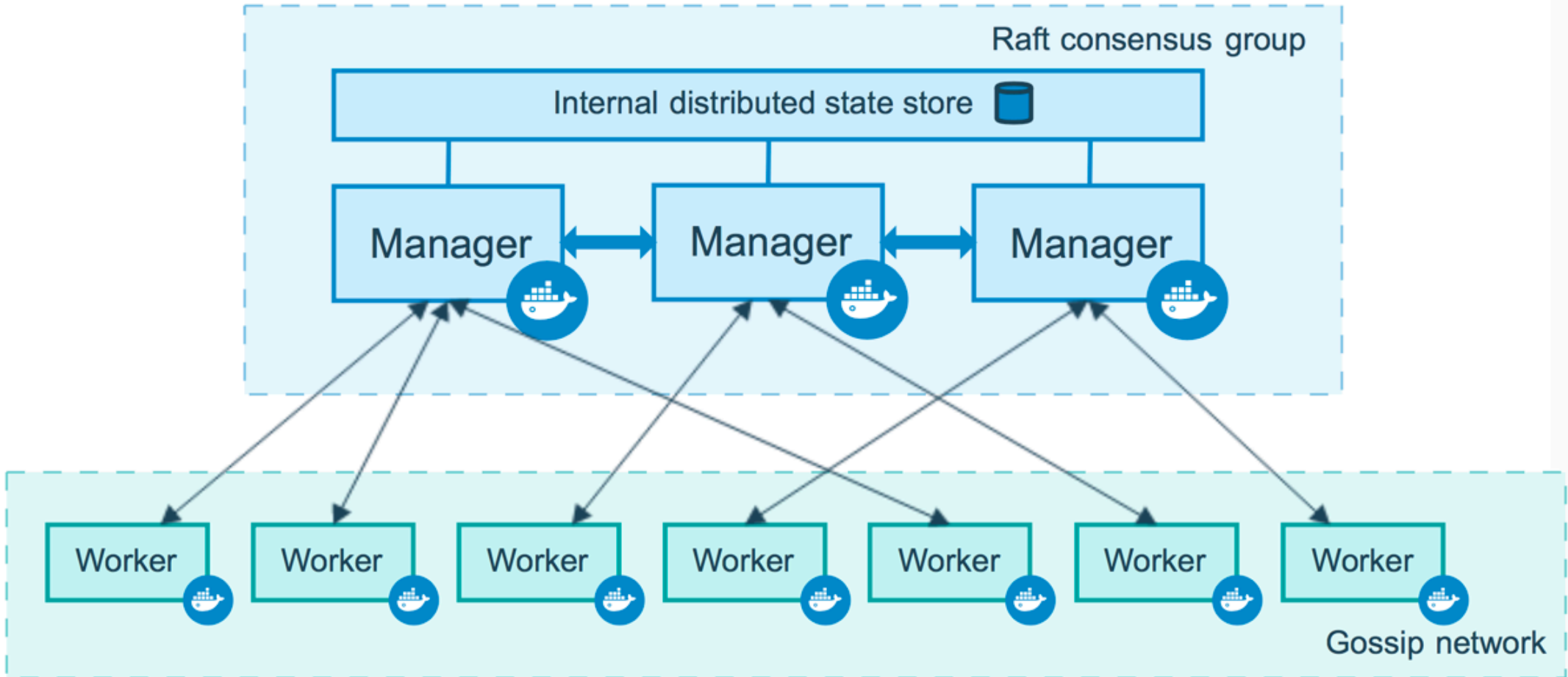
# Containers Everywhere = New Problems

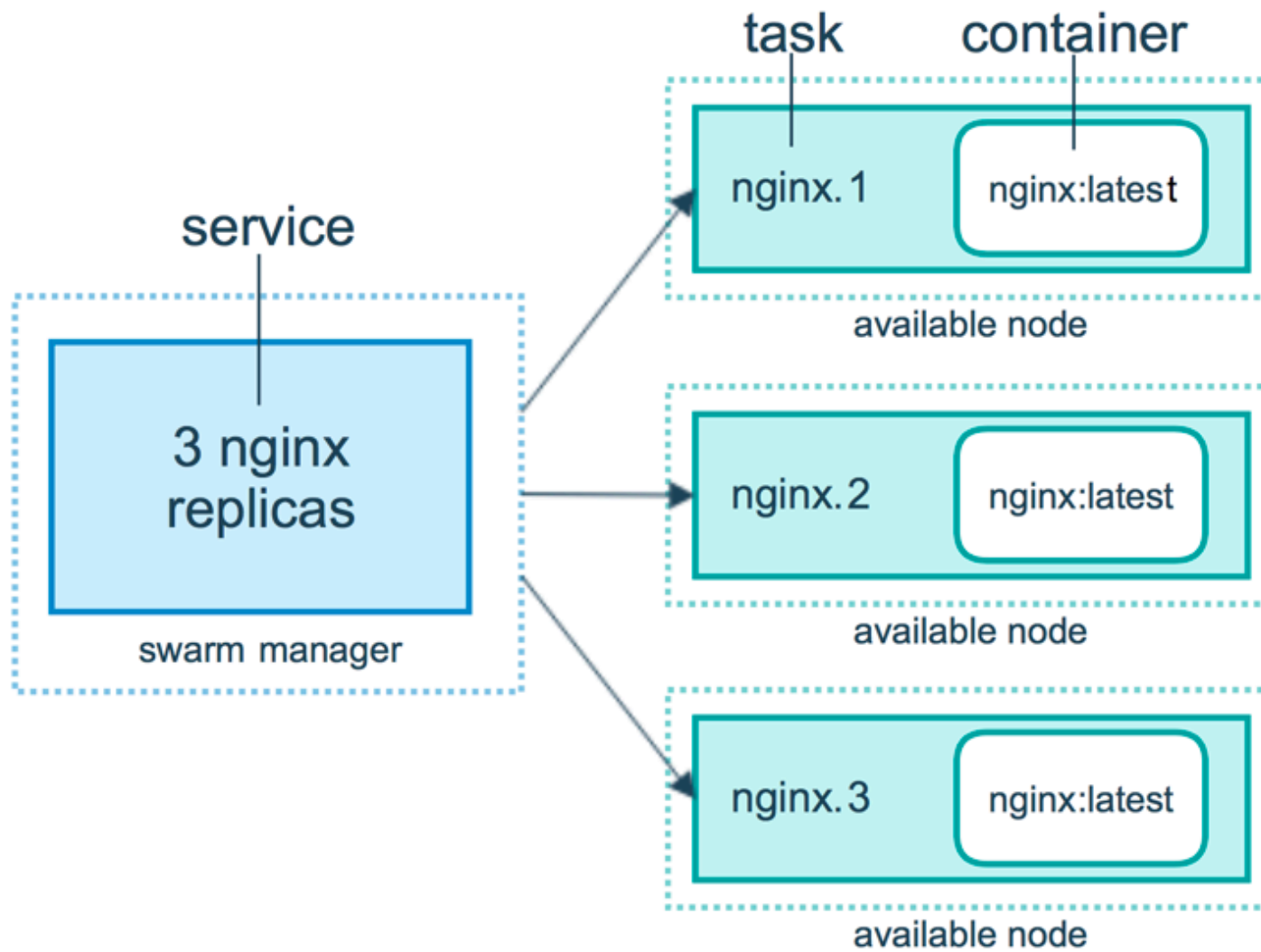
- How do we automate container lifecycle?
- How can we easily scale out/in/up/down?
- How can we ensure our containers are re-created if they fail?
- How can we replace containers without downtime (blue/green deploy)?
- How can we control/track where containers get started?
- How can we create cross-node virtual networks?
- How can we ensure only trusted servers run our containers?
- How can we store secrets, keys, passwords and get them to the right container (and only that container)?

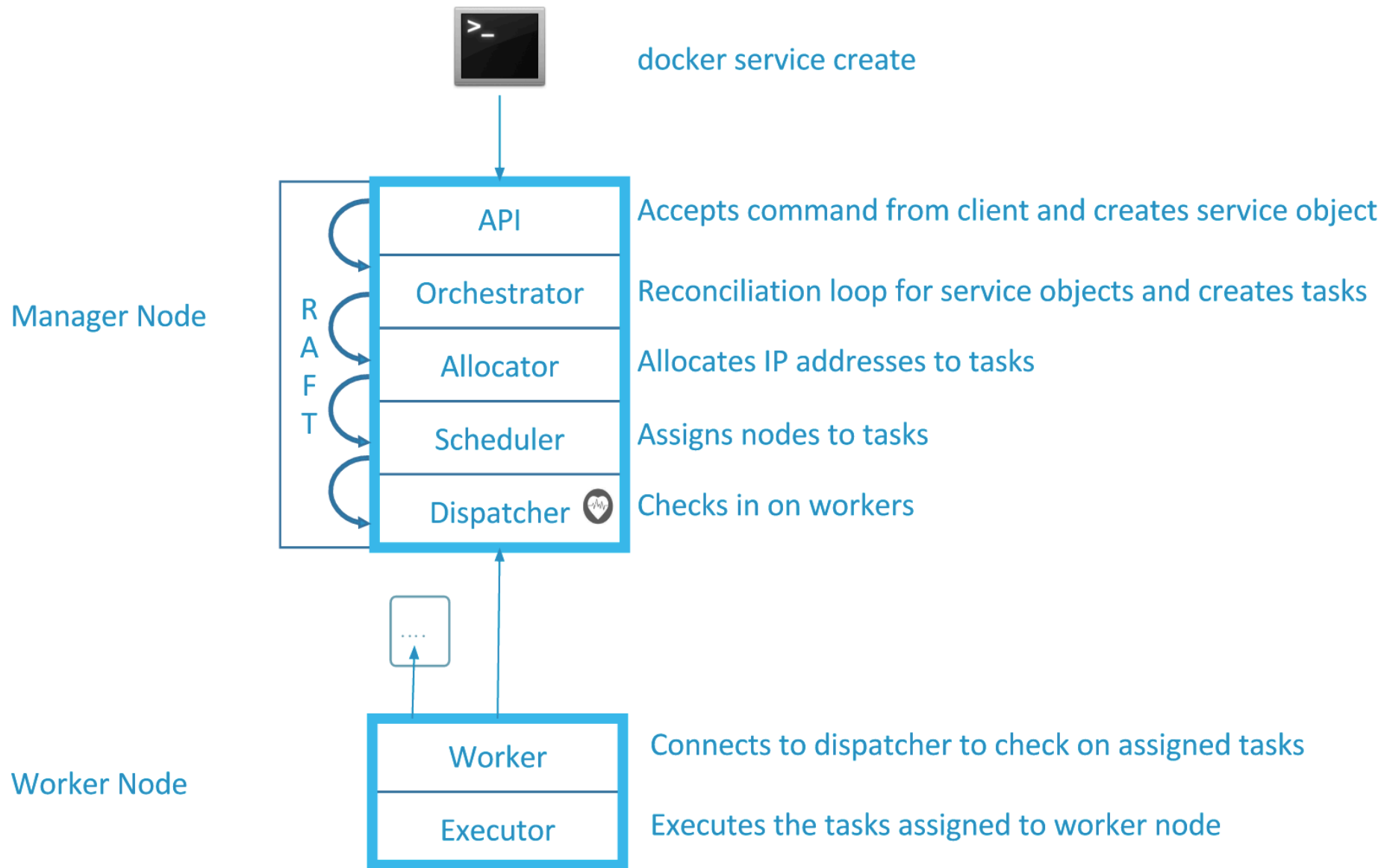
# Swarm Mode: Built-In Orchestration

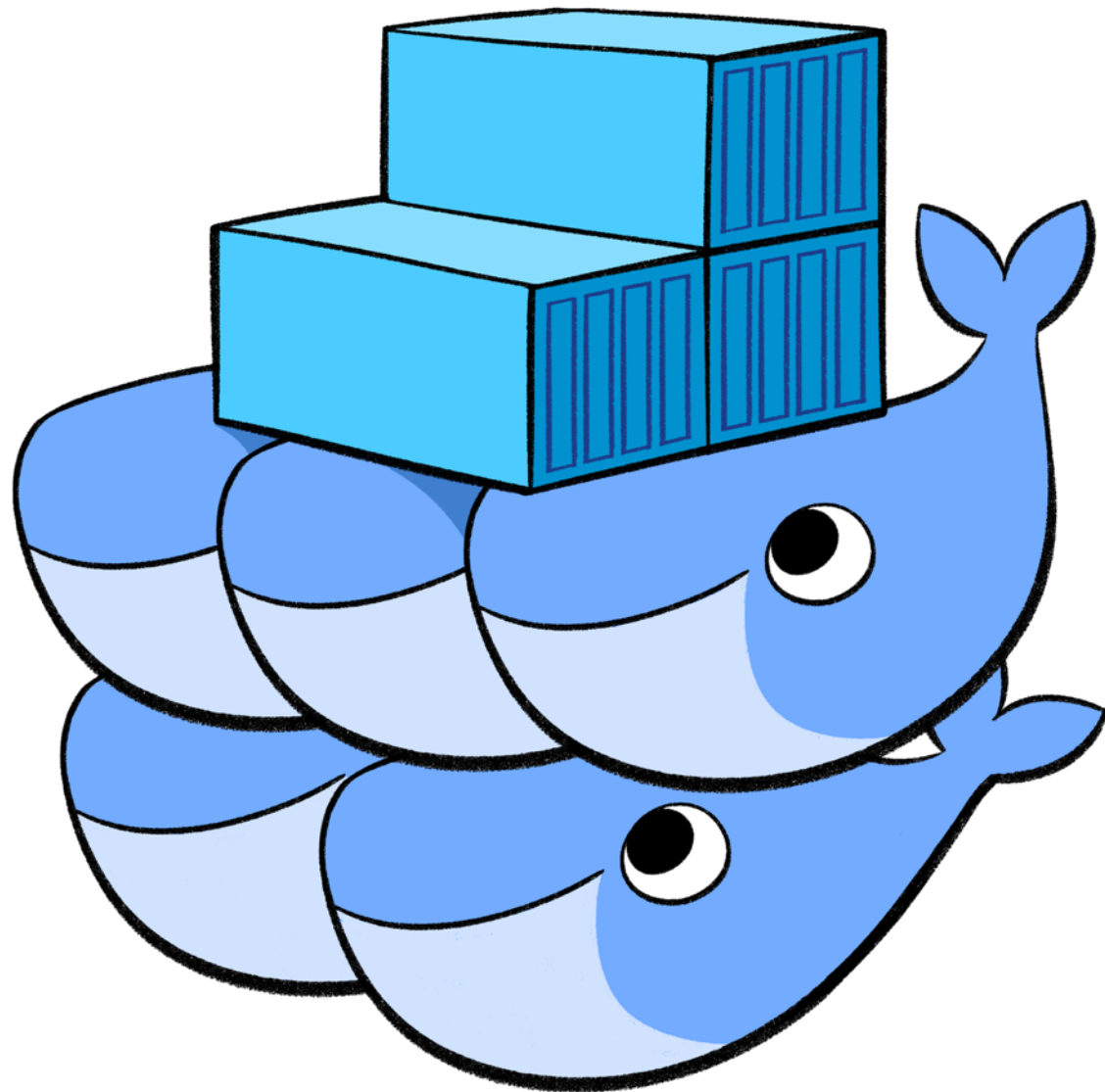
- Swarm Mode is a clustering solution built inside Docker
- Not related to Swarm "classic" for pre-1.12 versions
- Added in 1.12 (Summer 2016) via SwarmKit toolkit
- Enhanced in 1.13 (January 2017) via Stacks and Secrets
- Not enabled by default, new commands once enabled
  - docker swarm
  - docker node
  - docker service
  - docker stack
  - docker secret











# docker swarm init: What Just Happened?

- Lots of PKI and security automation
  - Root Signing Certificate created for our Swarm
  - Certificate is issued for first Manager node
  - Join tokens are created
- Raft database created to store root CA, configs and secrets
  - Encrypted by default on disk (1.13+)
  - No need for another key/value system to hold orchestration/secrets
  - Replicates logs amongst Managers via mutual TLS in "control plane"



# Creating 3-Node Swarm: Host Options

- A. [play-with-docker.com](https://play-with-docker.com)
  - Only needs a browser, but resets after 4 hours
- B. [docker-machine](#) + [VirtualBox](#)
  - Free and runs locally, but requires a machine with 8GB memory
- C. [Digital Ocean](#) + [Docker](#) install
  - Most like a production setup, but costs \$5-10/node/month while learning
  - Use my referral code in section resources to get \$10 free
- D. Roll your own
  - [docker-machine](#) can provision machines for Amazon, Azure, DO, Google, etc.
  - Install docker anywhere with [get.docker.com](https://get.docker.com)