Docker Swarm

Ramit Surana
@ramitsurana
/in/ramitsurana

Agenda

- Introduction to Docker
- Introduction to Docker Swarm.
- Docker Swarm Architecture.
- Introduction of the Components of Docker Swarm :
  - Resource Management
  - Swarm Scheduler
  - Discovery Service
  - Swarm API's
  - Swarm Store
- Comparison between Kubernetes & Docker Swarm

What is Docker ?

- Docker is a powerful build for your Linux containers.
- Open platform for developers and sysadmins to build, ship, and run distributed applications.
- Docker enables apps to be quickly assembled from components.
- It eliminates the friction between development, QA, and production environments.

What is Docker Swarm ?

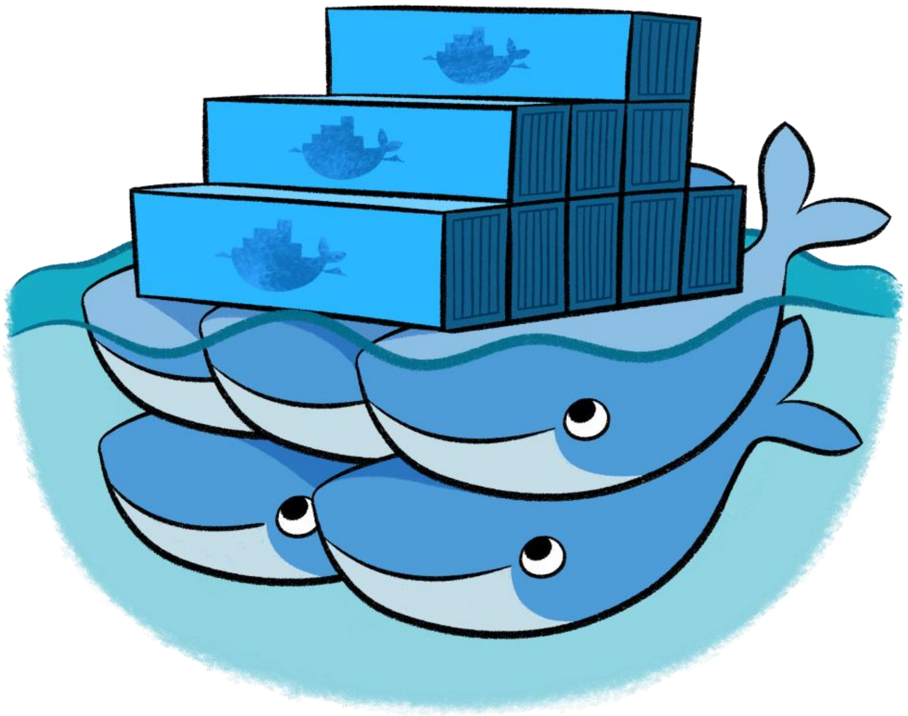
- Follows the "swap, plug, and play" principle.
- Native clustering for Docker.
- Allows you to create and access to a pool of Docker hosts.
- Support enabled for :

  - Jenkins

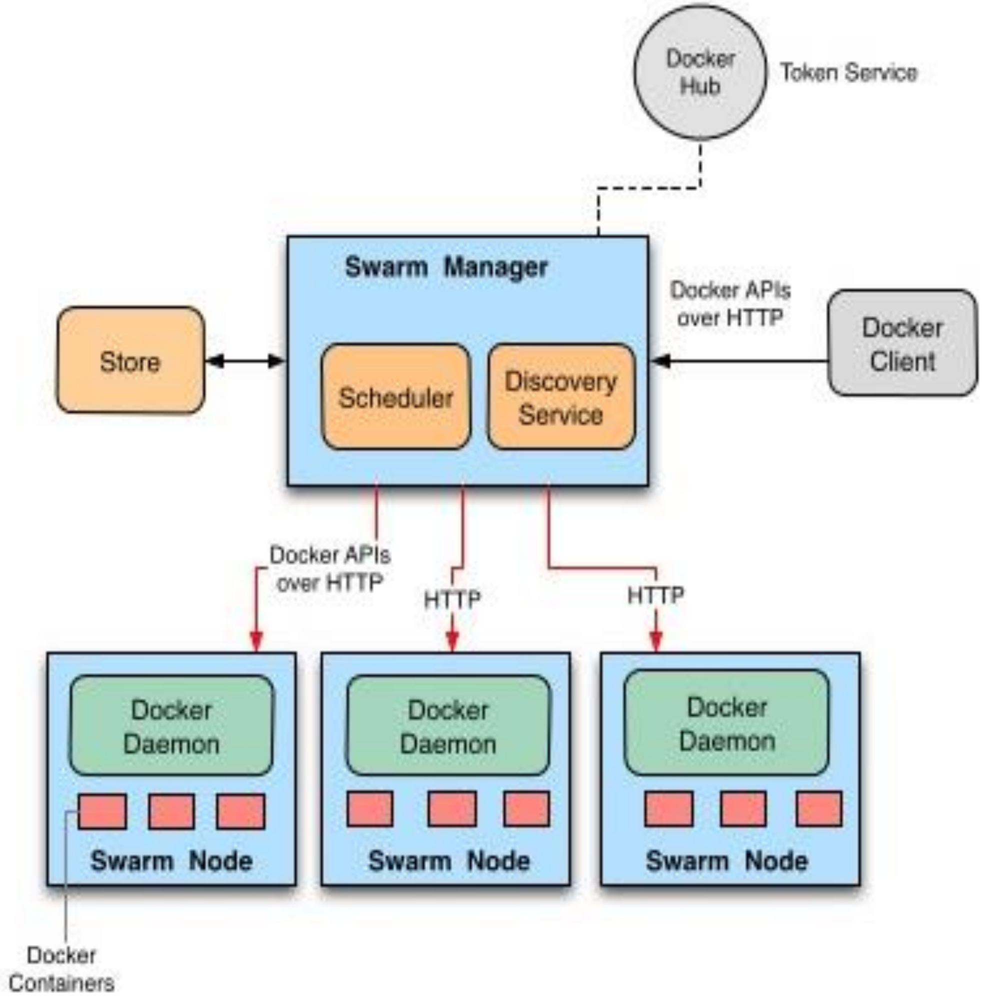
  - Docker Compose

  - Krane

  - Dokku
- Built by **Victor Vieux** and **Andrea Luzzardi.**



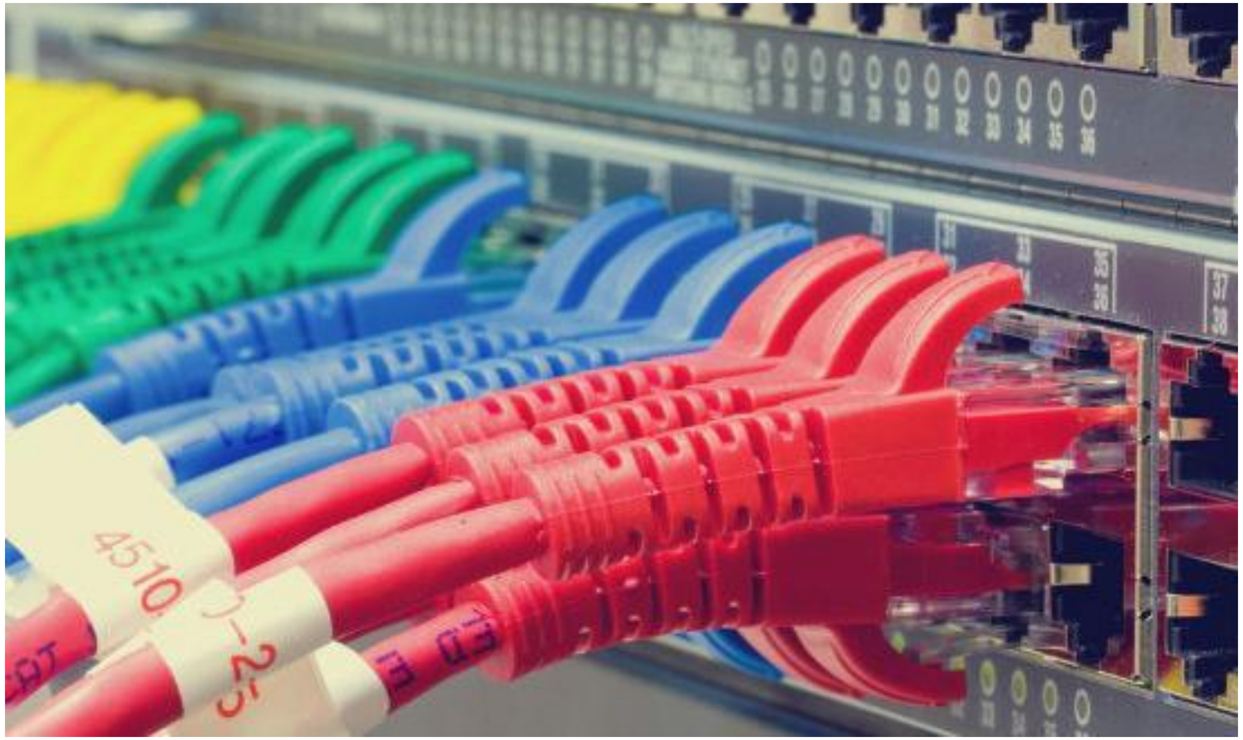
Architecture



Nodes

- Shows Runtime Instance
- Install Docker on each node.
- Elements of Swarm Node :

  - ID
  - Map of Containers & Images
  - CPU's Info
  - Memory Info



Resource management

- It is aware of the resources available in the cluster and will place containers accordingly.
- It takes into account the resource requirements of the container and the available resources of the hosts composing the cluster to optimize the placement using a bin packing algorithm.

Swarm Scheduler

- It is proposed to schedule containers on the nodes.
- It come with two strategies,namely:
- BinPacking Strategy:

It will rank the nodes according to there CPU and RAM available.

- Random Strategy:

It is majorly used for debugging,it selects a random node.

Discovery Service

- It is a hosted discovery service with Docker Swarm.
- The service maintains a list of IPs in your swarm. There are several available services, which can connect to docker swarm such as `etcd`, `consul` and `zookeeper`.

Swarm API's

- Containers:

GET

POST

DELETE

- `GET "/containers/json"`: Node's name prepended to the container name.
- `GET "/images/json"` : Use '--filter node=<Node name>' to show images of the specific node.

Swarm Store

- Stores the state of the Cluster.
- State is loaded in the memory when cluster starts.
- Lifecycle events :
  - Delete the state
  - Replace the state of the key with new state
  - Load all data stored

Constraints and Affinity

- In order to meet the specific requirements of each container, their placement can be fine-tuned using constraints.
- Constraints operate on Docker daemon labels.
- The placement of a container must be relative to other containers. Swarm lets you define those relationships through affinities.
- Affinities are automatically generated when the relationship between containers is implied.

Setup

- Creating a Cluster :

```
$ swarm create
```

- Add nodes to the cluster :

```
$ swarm join --token=<token> --addr=<node_ip>
```

- $ swarm list --token=<token>
- $ swarm manage --token=<token> --addr=<swarm_ip>

Kuberntes vs Swarm

- Built by Highly Experienced Google Team.
- It used flannel to create networking between containers, it has load balancer integrated, it uses etcd for service discovery, and much more.
- You cannot use Docker CLI nor you can use Docker Compose to define containers.

- Built by Docker.
- Native clustering for Docker.
- It exposes standard Docker API meaning that any tool that you used to communicate with Docker can work equally well with Docker Swarm.

Questions ?

Connect with Me:

/in/ramitsurana

@ramitsurana

github.com/ramitsurana

Sharing is caring !