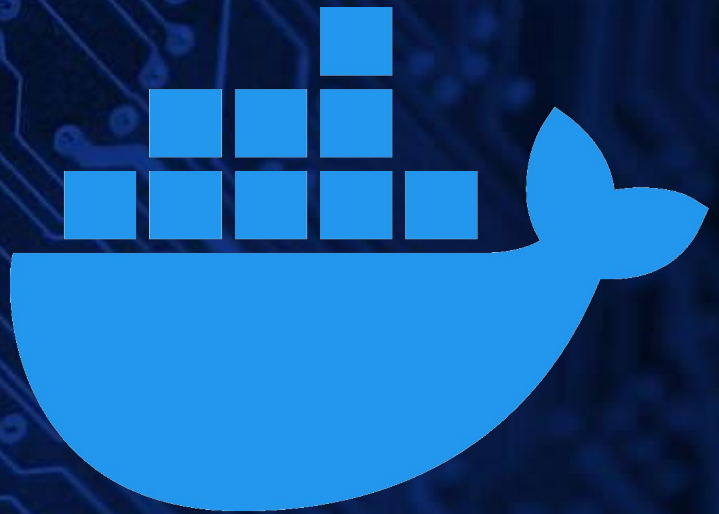
Docker : Docker MasterClass for DevOps

Docker Swarm Introduction



Docker : Service Containers

- Problem Statement :
  - How to Scale Containers?
  - How to manage Containers or re-create if they Fails/Crash?
  - How to Upgrade the Service with Zero DownTime?
  - How to Manage Containers on VMs, Nodes?

Docker : Service Containers

- **Docker Swarm :**
- Docker Swarm is a clustering and scheduling tool for **Docker containers**.
- Swarm is Docker's native support for orchestrating clusters of Docker engines.
- **Orchestration** : Define nodes. Define services. Set how many nodes you want to run and where, and you're done.
- At a high level, Swarm takes multiple Docker Engines running on different hosts and lets you use them together.

Docker : Service Containers

- Docker Swarm : Docker Swarm have Two Type of Nodes Master(Manager) and Worker.

- Every swarm starts out with one manager node designated as the leader.

- Swarm is highly available thanks to its implementation of the **Raft algorithm**.

- **Raft Algo** : Raft is a consensus algorithm designed to achieve fault tolerance in distributed systems.
  The leader node is constantly checking in with its fellow manager nodes and syncing their states.

Docker : Service Containers

- **Nodes and Roles**: In Raft, a cluster consists of multiple nodes, and each node can have one of three roles: leader, follower, or candidate.
  There is one leader at a time, and followers replicate the leader's actions.

- **Leader Election**: Algorithm starts with all nodes in the follower state.
  - If a follower doesn't receive communication from a leader for a certain period (election timeout), it transitions to the candidate state and requests votes from other nodes to become the leader.
  - If a candidate receives votes from the majority of the nodes, it becomes the leader.

Docker : Service Containers

- **Handling Failures**:
  - If a leader fails, a new leader is elected through the election process.
  - Nodes can detect inconsistencies and missing entries in their logs and update them by replicating the logs from the leader.

Docker : Service Containers

- Task Scheduling
- Load Balancing
- Rolling Updates
- Security

Thank You...

Don’t be the Same! Be Better!!!