DOCKER

Docker Swarm : Terminology

Docker : Docker Swarm

- Docker Swarm : The cluster management and orchestration features embedded in the Docker Engine are built using swarmkit.
- A swarm consists of multiple Docker hosts which run in **swarm mode** and act as managers (to manage membership and delegation) and workers (which run swarm services).
- Host : Docker host can be a manager, a worker, or perform both roles.
- Service : When you create a service, you define its optimal state (number of replicas, network and storage resources available to it, ports the service exposes to the outside world, and more).

Docker : Docker Swarm

- Docker Swarm : Docker Swarm maintains the Service Desired State. For instance, if a worker node becomes unavailable, Docker schedules that node’s tasks on other nodes.
- Task : T*ask* is a running container which is part of a swarm service and managed by a swarm manager.

Docker : Docker Swarm

- Nodes : A **node** is an instance of the Docker engine participating in the swarm.
- You can run one or more nodes on a single physical computer or cloud server, but production swarm deployments typically include Docker nodes distributed across multiple physical and cloud machines.
- To deploy your application to a swarm, you submit a service definition to a **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. Manager nodes elect a single leader to conduct orchestration tasks.

Docker : Docker Swarm

- **Worker nodes** receive and execute tasks dispatched from manager nodes.
- Service : A **service** is the definition of the tasks to execute on the manager or worker nodes.
- When you create a service, you specify which container image to use and which commands to execute inside running containers.
- Task : A **task** carries a Docker container and the commands to run inside the container.
- Once a task is assigned to a node, it cannot move to another node. It can only run on the assigned node or fail.

Docker : Docker Swarm

- Load Balancing : Swarm manager uses **ingress load balancing** to expose the services you want to make available externally to the swarm.
- External components, such as cloud load balancers, can access the service on the PublishedPort of any node in the cluster whether or not the node is currently running the task for the service. All nodes in the swarm route ingress connections to a running task instance.

Will see you in Next Lecture…

See you in next lecture …