Docker Swarm
Rolling Update & Rollback

Docker swarm updates

- Docker swarm supports various options to perform rolling updates and also rollback (if something goes bad)
- Let us create a simple Swarm using a sample voting webapp called instavote/vote

>docker swarm init --advertise-addr MY_IP

- Let us create 4 replicas of our web-app

>docker service create --name vote --replicas 4 --publish 5000:80 instavote/vote

Docker swarm Rolling update

- Check if the application is up & running fine

http://localhost:5000/

- Let us now update the service to a newer Image instavote/vote:indent

>docker service update --image instavote/vote:indent vote

- Check the pattern in which updates occur

Preparing > Ready > Starting > Running

Docker swarm Rolling update

- Each Node is updated one by one (Rolling Update)
- Can be customized using the following configuration

--update-parallelism: number of tasks to update at the same time

--update-delay: time to wait before updating the next batch of tasks

Docker swarm Rolling update

- Let us configure our Swarm update to update 2 nodes in parallel and let us set a 10s delay between such batch updates

>docker service update --update-parallelism 2 --update-delay 10s vote

- Let us perform a new update with this configuration

>docker service update --image instavote/vote:movies vote

- Check the node update pattern

Docker swarm Rollback

- Any service update to a swarm cluster can be easily rolled back since swarm knows about the previous deployment

>docker service rollback vote

- Uses default rolling update configuration for rollback and if needed, parallel and delays can be configured using

--rollback-parallelism

--rollback-delay

Docker swarm Auto Rollback

- Using effective health checks for new deployments, automatic rollbacks can be configured for bad deployments