Swarm Cluster Setup

- The number of master nodes must be an odd number.
- Master nodes should be placed in different datacenters
- The master node should be running in drain mode (do not host any Docker containers).
- The worker nodes need to be distributed over multiple data centers.
- Connecting multiple independent Swarm cluster together behind a load balancer.

Service Availability

Deploying services or rolling updates for Docker Swarm services needs to be done with zero downtime.

Handle service logs

By default, Docker uses the `json-file` logging driver. This driver will store containers logs in the file system under the following path on the host node `/lib/docker/containers/${id}/${id}-json.log`.

- Use another log driver: Docker supports several logging drivers such as Syslog, Fluentd, or Gelf.
- Rotate log files

```
logging:
      driver: json-file
      options:
        "max-size": "10m"
        "max-file": "5"
```

Resource Usage

Running services in a Swarm cluster without limiting the attached resources (RAM and CPU) to these services may expose the whole Swarm cluster and the services running in it to a downtime.

According to the Docker Compose reference, both `cpu` and `memory` resource constraints can be set for the individual services in the following way:

```
version: "3.7"
services:
  redis:
    image: redis:alpine
    deploy:
      resources:
        limits:
          cpus: '0.50'
          memory: 50M
        reservations:
          cpus: '0.25'
          memory: 20M
```

Network Limits: several vs big network

- Every service can access every other service and this may introduce the risk of unwanted communication or unauthorized actions.

- Default overlay networks are sized with `/24` subnets, which means these networks can have only 265 IPs. The below command can be used to create a larger Docker overlay network.

```
docker network create --driver overlay --subnet=192.168.0.0/16 bignet
```

Cluster cleanup: Docker system prune

This task can be easily done by executing the below command.

```
docker image prune -a --filter "until=24h"
```

The above command will remove all the docker images created more than 24 hours ago. More regarding the prune command can be found here.