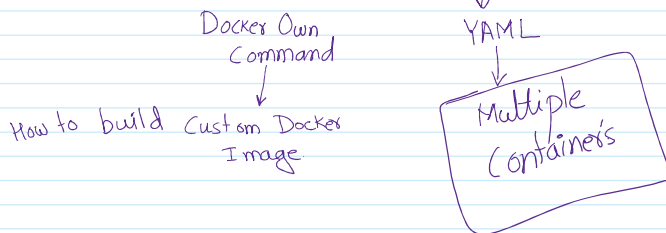Multiple Container

- Micro service
- Multi Tier based Application.
- Docker Compose. ⟶ YAML (Indentation based file)

Services
| C1 |
| C2 |

Volume

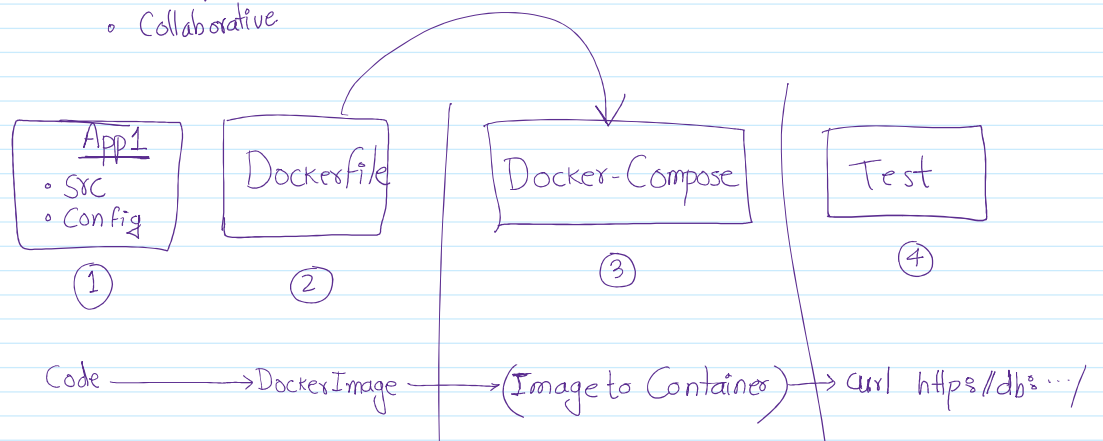Secret

Env.
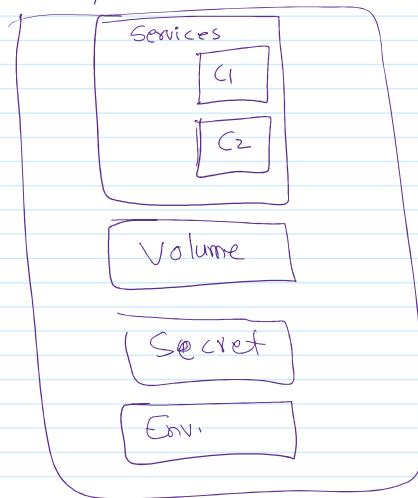
Dockerfile vs D. Compose   vs   Swarm.

Docker Own   YAML
Command

How to build Custom Docker   Multiple
          Image   Containers

docker build   (I) → (C)
docker run
docker exec.   - Gradually.
                      - Enterprise
                      - Collaborative

App1	Dockerfile	Docker-Compose	Test
- Src - Config			
①	②	③	④

Code ⟶ DockerImage ⟶ (Image to Container) → curl https//db:···/

Code ————→ Docker Image ————→ (Image to Container) →→ curl https//db:···/

build : .

dc UP
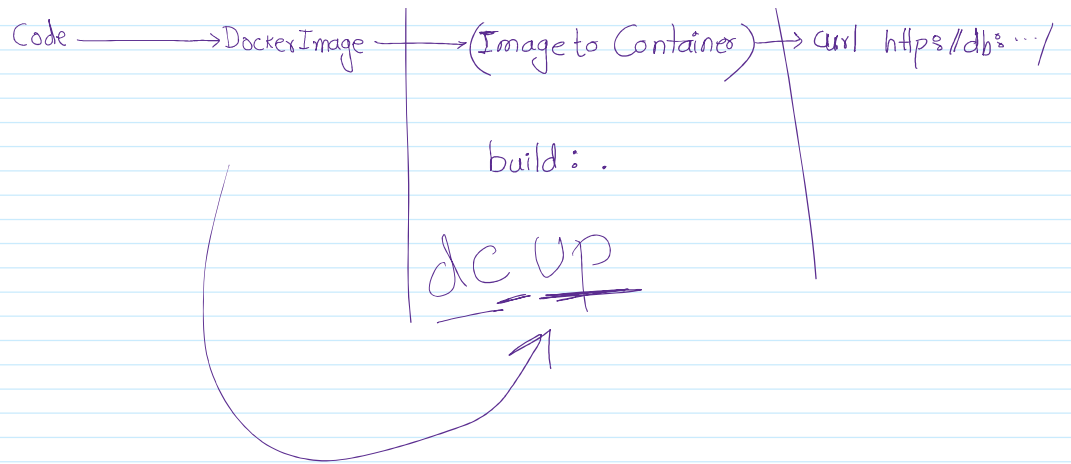
Swarm :—
Docker   Mode   ◦ active / inactive.
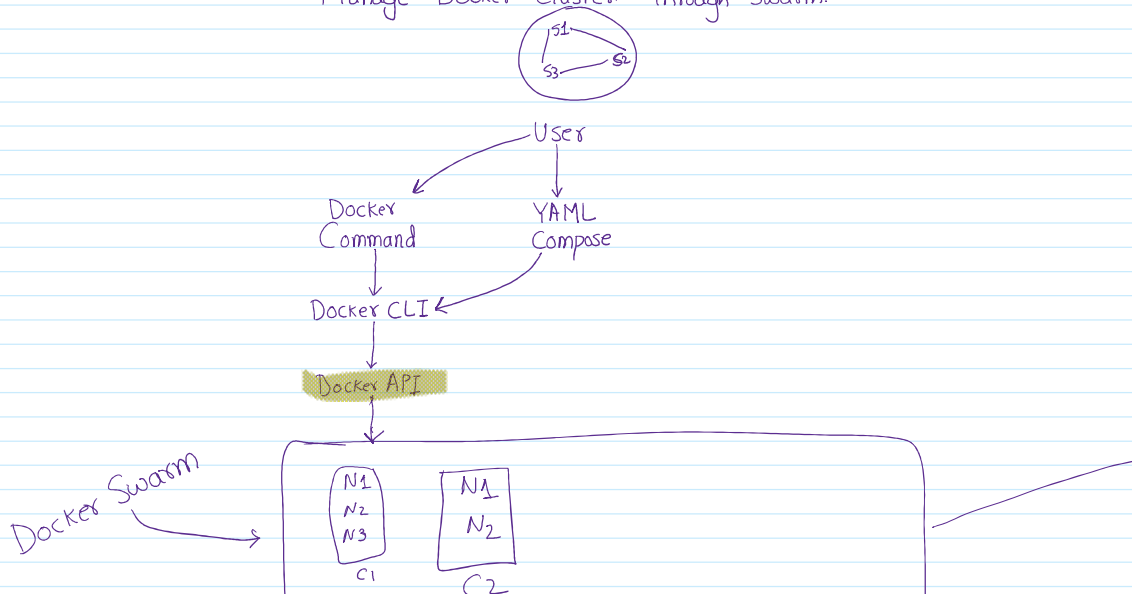
◦ Container   orchestration tool
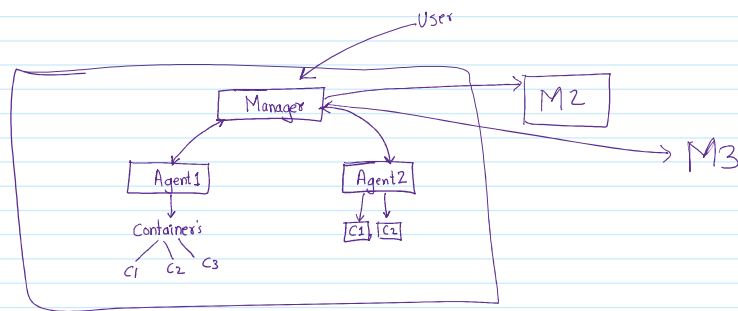
Feature's
  1] Scaling.
  2] Multi host networking.
  3] Service   Discovery.
  4] Decentralized design
  5] Load Balancer
  6] Secure by default (Token)
  7] Rolling updates
  8] All types   instruction
        ◦ Command
        ◦ Dockerfile
        ◦ Docker- Compose.

  ◦ Embedded Feature of docker, swarmkit
  ◦ Manage Docker Cluster through Swarm.

            S1
         S3   S2

            User

  Docker   YAML
  Command   Compose

  Docker CLI

  Docker API

Docker Swarm →
            N1   N1
            N2   N2
            N3
              C1   C2

Docker →

```
┌─────────┐  ┌──────┐
│ N2      │  │ N2   │
│ N3      │  │      │
└─────────┘  └──────┘
   C1          C2
```

VM/cloud/
on-prem

```
┌──────────────────────────┐
│        Linux             │
│          ↓               │
│       Hardware.          │
└──────────────────────────┘
```

User

```
┌──────────────────────────────────┐
│          ┌─────────┐      ┌──────┐│
│  ┌──────→│ Manager │─────→│ M2   ││
│          └─────────┘      └──────┘│
│           ↙        ↘              │
│  ┌────────┐      ┌────────┐  → M3 │
│  │ Agent1 │      │ Agent2 │       │
│  └────────┘      └────────┘       │
│      ↓              ↓             │
│  Container's      ┌──┐┌──┐        │
│   ╱  │  ╲         │C1││C2│        │
│  C1  C2  C3       └──┘└──┘        │
└──────────────────────────────────┘
```
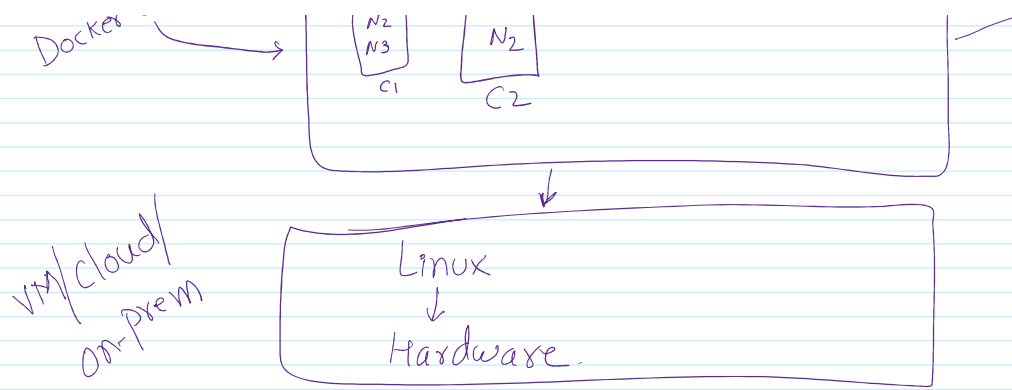
1] At least 2 Machines

2] Same network (VPC should be same)

3] Sequence of IP address should be :

4] master $ initialize. Swarm ⟶

                              Token, MasterIP, 2377

5] worker1 $ Join Swarm ←

6] Check nodes (Master, Worker's)

```
$ docker service create \
  --name=viz \
  --publish=8080:8080/tcp \
  --constraint=node.role==manager \
  --mount=type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \
  dockersamples/visualizer
```

master# docker swarm init --advertise-addr 192.168.0.23

worker1# docker swarm join --token SWMTKN-1-1483ufrinaxi2wwtq5vbpginnoekafectlqqul4zxlg219pull-7slsavk4u8gvy9wod170dg3g9 192.168.0.23:2377

docker swarm join --token SWMTKN-1-... MasterIP:2377

master $ docker node ls

master $ docker info