Kubernetes:

Docker is a native of Linux.

Kubernetes is native of google.

Uses: Netflix, Zee5, IBM, & Google.

Docker Swarm is only used with Docker. For others, we must go for Kubernetes.

Monolithic Architecture: (Mono = Single, Lithic = Layer) Single Codebase used for software development in early days. In this architecture, if there is any bug you must redeploy it on the pod server which is 90% sure in monolithic architecture. It takes time, costs a lot, & no CI/CD. Every feature in monolithic architecture is tightly coupled as all are dependent of one another. DevOps can't be on Monolithic Architecture.

Microservices: Improved version of Monolithic architecture. E.g., Java Spring Boot. Every Feature in this architecture is loosely coupled. Multiple Codebases & multiple languages also worked in this. Like Java & Python. API Gateways is used for intercommunications between different teams in microservices. Microservices bring the use of API's. Microservices has pre-defined allocation of resources, which is a problem. Then the concepts of containers will come into the market which will manage resources efficiently.

Jenkins -> Maven -> Test -> Unit Test -> …… -> Deployment

Containers: Containers are based on dynamic allocation. When traffic increases or decreases, containers will not increase or decrease appropriately according to the traffic. At that Kubernetes solved this by automating this containers problem.

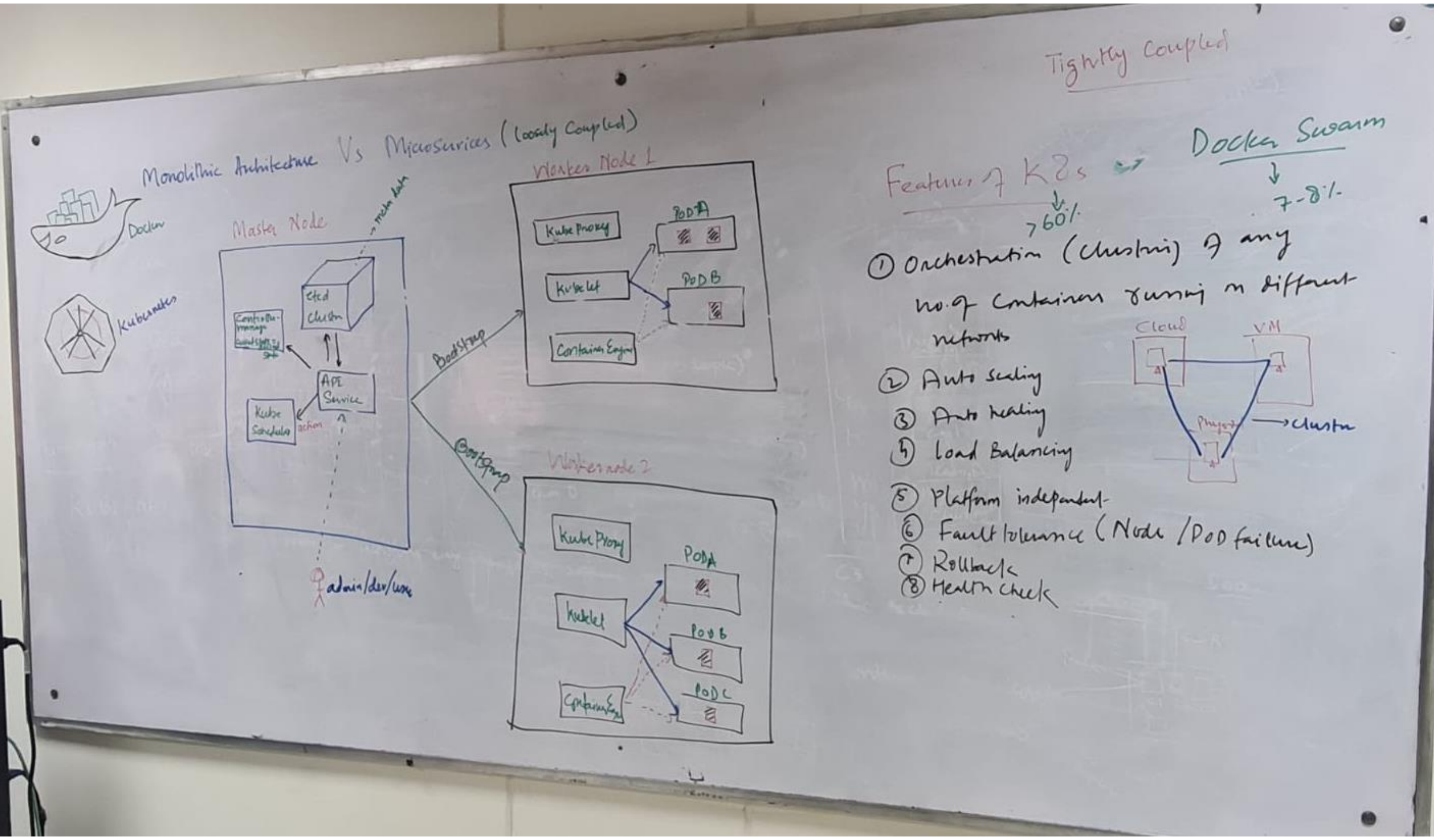
Kubernetes: (Greek Word: Captain of ships) Orchestration & container management tool. Also called K8S. Kubernetes not directly interact with containers; it has to deal with pods & inside the pods there is containers.

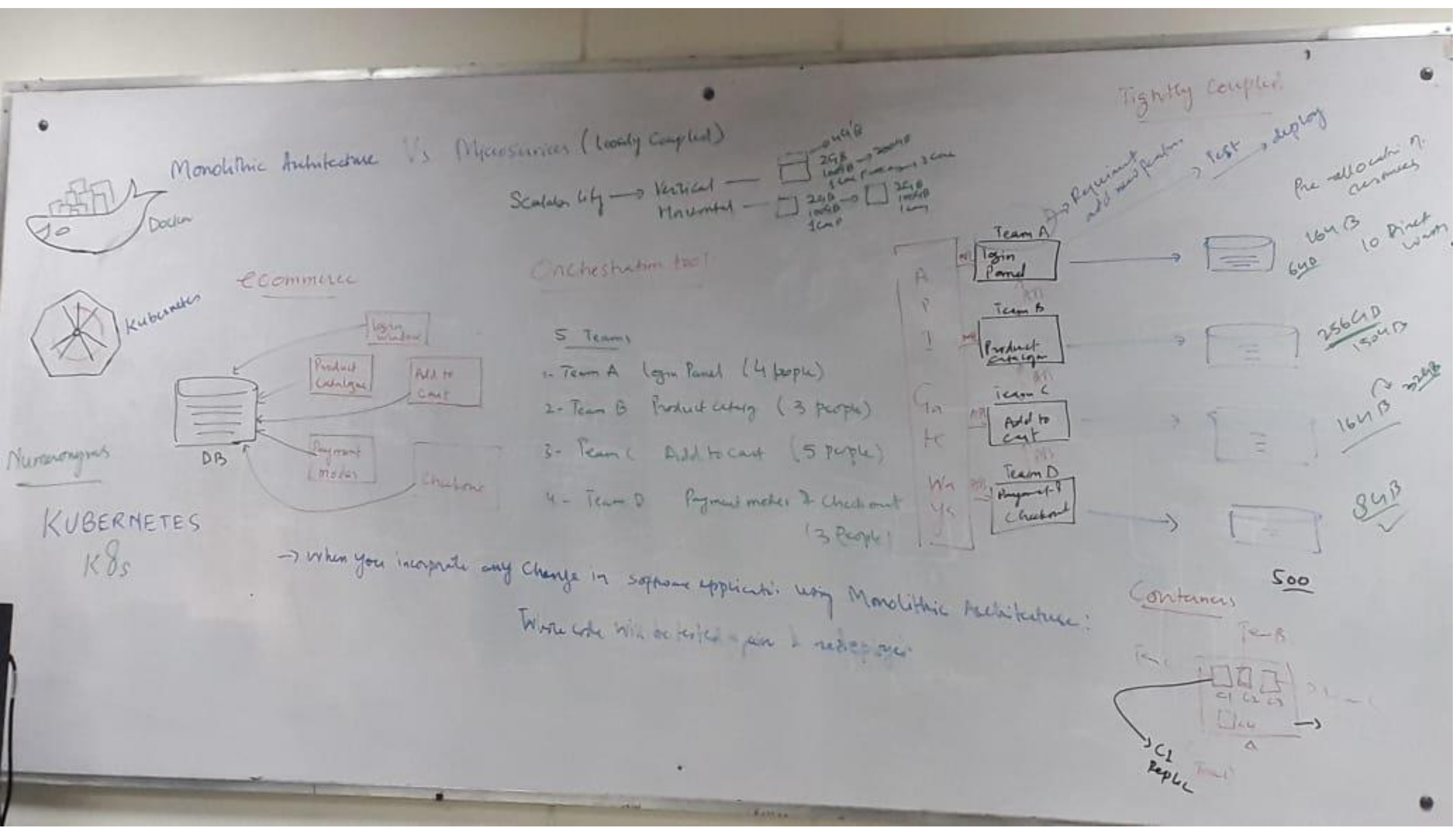
Cluster is combination of master & working node.

Features:

1. Claustration: Clustering of any number of combinations running in different network.
2. Auto Scaling: Increase or decrease of containers automatically.
3. Auto Healing: Look & diagnose for any faults automatically.
4. Load Balancing: Evenly distribution of requests, no burdened, no overload.
5. Platform Independent:
6. Fault Tolerance: By providing highly available systems like backup.
7. Rollback: Versioning of system.
8. Health Check: Check for containers health.
9. Powerful:
10. Flexible:

Bootstrapping: Connection between master & working node. Single Container in single pod as its easy to identify single container.


Tightly Coupled
Monolithic Architecture Vs Microservices (Loosely Coupled)
Docker
Kubernetes
Master Node
etcd cluster
API Service
Kube Scheduler
action
admin/dev/user
Bootstrap
Worker Node 1
Kube Proxy
Kubelet
Container Engine
POD A
POD B
Worker node 2
POD A
POD B
POD C
Features of K8s
>60%
Docker Swarm
7-8%
① Orchestration (clustering) of any no. of Containers running in different networks
② Auto scaling
③ Auto healing
④ Load Balancing
⑤ Platform independent
⑥ Fault tolerance (Node / POD failure)
⑦ Rollback
⑧ Health check
Cloud
VM
cluster


Tightly Coupled
Monolithic Architecture Vs Microservices (Loosely Coupled)
Docker
Kubernetes
ecommerce
DB
KUBERNETES
K8s
Scalability → Vertical
Horizontal
Orchestration tool
5 Teams
1- Team A Login Panel (4 people)
2- Team B Product catalog (3 people)
3- Team C Add to Cart (5 people)
4- Team D Payment modes & Checkout (3 People)
Team A
Login Panel
Team B
Product Catalogue
Team C
Add to cart
Team D
Checkout
→ test → deploy
→ When you incorporate any Change in software application using Monolithic Architecture:
500
Containers
C1
Replica