notes

links

dashboard

Containers - Learning Path

Containers and Orchestration

Containers

- Lightweight virtualization technology.
- Packages applications and dependencies in a consistent environment.
- Ensures consistent application execution across different systems.
- Ideal for development, testing, and deployment.
- Offers portability and efficient resource utilization.

Kubernetes (K8s)

- Open-source container orchestration platform.
- Automates deployment, scaling, and management of containerized applications.
- Provides features like load balancing, auto-scaling, self-healing, and rolling updates.
- Abstracts infrastructure complexities, focusing on application management.
- Enables efficient deployment and scaling of container workloads across clusters.

container diagram

```
[ Host Server (with OS)]
[Container0]
[Container1]
[Container2]
[Container3]
```

Orchestration

- Container orchestration automates and manages containers.
- Orchestration tools like Kubernetes automate container deployment and scaling.
- Orchestration ensures containers are distributed across hosts for redundancy.
- It simplifies complex requirements for managing containers.
- Zero-downtime deployments ensure uninterrupted service during updates.
- Containers running old and new code coexist during a deployment.
- Orchestration tools coordinate container deployment steps efficiently.

- Orchestration automates tasks like spinning up containers, traffic switching, and cleanup.
- Container orchestration enables quick, reliable, and efficient management of containers.

Zero-Downtime deployment

1. Spin up conatiners running new Code
2. When they are fully up, direct user traffic to the new containers
3. Remove the old containers running the old code (no downtime for users)

what containers can be used for ...

1. Software Portability
2. Isolation
3. Scaling
4. Automation
5. Efficient Resource Usage