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

Advantages of Containers

- same isolation and portability of VMs
- lightweight - less resources
- Faster than VMs
- smaller than VMs
- faster and simpler automation

Limitations of Containers

- less flex than VMs (No windows containers on linux machines . yet)
- new challenges with orchestrationa and automation

Docker

create manage and run containers

- This section of the course covers specific container and orchestration technologies.
- The focus of this lesson is on Docker, one of the most popular container runtimes.
- Docker is a container runtime, which enables the implementation and running of containers.
- Containers are a concept, and Docker is a tool that implements this concept.
- Docker provides tools for running, building, and managing containers and container images.
- The course does not provide technical details about using Docker but recommends exploring official Docker documentation for more information.
- There are alternative container runtimes available besides Docker, which will be briefly mentioned in the next lesson.

other container runtimes

- RKT "rocket" - security and composability
- containerd "container d" - simple

Kubernetes

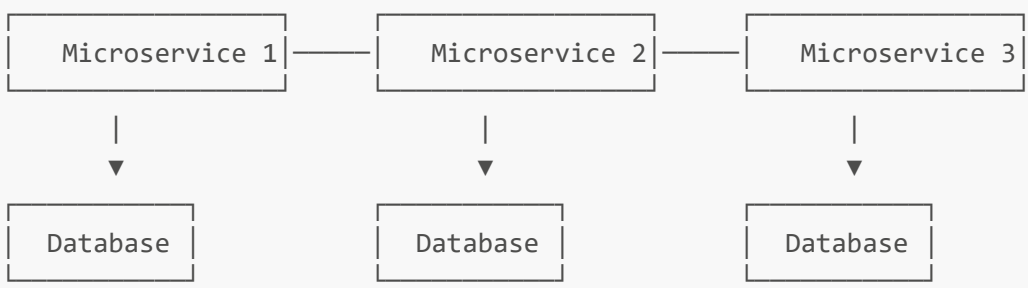
- This lesson introduces Kubernetes as a container orchestration tool.
- Kubernetes simplifies building and managing container infrastructure and automation.
- Orchestration tools like Kubernetes automate tasks such as deploying, scaling, and managing containers.
- Kubernetes enables self-healing applications, automated scaling, and easy automated deployments.
- It is the industry-leading container orchestration tool, known for its extensive features and power.
- You can explore more about Kubernetes in the official Kubernetes documentation at kubernetes.io.
- The next lesson will discuss alternative orchestration tools to consider.

other orchestration tools

- Docker Swarm - native to docker
- Marathon - tons of APIs
- nomad - simple and lightweight
- Amazon Elastic Container Service
- Amazon ECS for Kubernetes
- Azure Kubernetes Service
- IBM Cloud Kubernetes
- RedHat OpenShift
- Google's Kubernetes Engine

microservices

- This section discusses use cases of containers and orchestration, starting with microservices.
- Microservices involve splitting applications into small, independent services.
- Containers make it easier to implement and manage microservices.
- Microservices offer benefits like rapid development, reduced risk, and technology optimization.
- Containers are suited for managing a large number of small, independent microservices.
- Orchestration simplifies the deployment, scaling, and connection of microservice instances.
- Containers enable easy resource scaling for microservices, with orchestration automating the process.
- Orchestration can automatically detect increased usage and scale microservices without manual intervention, improving end-user experience.
- Containers and microservices provide business value by enhancing end-user experiences and reducing manual administrative work.



or

```
[Microservices]
[    App      ]
[product data]
[customer data]
[authentication]
[order history]
```