

Stand-alone vs Orchestrated

After completing this episode, you should be able to:

- Discuss stand-alone vs orchestrated container implementations

Description: In this episode, you will learn about the different methods of implementing containers in your cloud environment. This discussion includes a comparison of stand-alone versus orchestrated cloud container infrastructures.

Stand-alone vs Orchestrated

In a standalone container infrastructure, each application or service runs within its isolated container, managed and operated individually. This means that each container must be manually configured, deployed, and scaled, typically without coordination with other containers. While standalone containers offer simplicity and flexibility, they require more manual effort to manage and lack advanced features like automated scaling and load balancing.

On the other hand, an orchestrated container infrastructure utilizes a container orchestration platform such as Kubernetes to automate the management, deployment, scaling, and networking of containers across a cluster of machines. With container orchestration, administrators can define desired states for the application, such as the number of instances, resource requirements, and networking rules. The orchestration platform then handles the deployment and scaling of containers to meet these requirements automatically. This approach simplifies operations, improves resource utilization, and enhances fault tolerance by automatically redistributing workloads and restarting containers in case of failures.

While standalone containers offer simplicity and are suitable for smaller deployments or environments where manual management is acceptable, orchestrated container infrastructures are preferred for larger-scale deployments, where automation, scalability, and reliability are critical requirements. They provide more advanced features and capabilities to manage complex distributed applications efficiently.

Additional resources

- What is a Container: <https://www.docker.com/resources/what-container/> (<https://www.docker.com/resources/what-container/>)