

Lab 2 - QUIZ by Athika Fatima(101502209)

1) Understand what kubernetes doesn't do?

Answer: While Kubernetes is capable of so many things like orchestrating containers and making our lives easier, it is also very important to understand and know what it is not capable of doing. Here are some things that Kubernetes doesn't handle explicitly:

- **Application Code:** Kubernetes is focused on orchestrating and managing containerized applications but developers are responsible for the actual implementation of the application logic. So if a component is not giving you the right output, it's your code logic - NOT KUBERNETES!
- **Image Building:** Kubernetes doesn't provide tools for building container images. We need to use Docker to create container images that can be deployed on a Kubernetes cluster.
- **Storage and Databases:** Although Kubernetes supports storage volumes, it doesn't manage databases or handle data persistence for apps. We need to integrate external solutions or cloud services for database management and storage.
- **Networking Solution:** Kubernetes manages the networking between containers and services within the cluster, but it doesn't replace the need for external networking solutions. We will still need to use other platforms to handle Networking policies, firewalls, and external load balancing.
- **Monitoring and Logging:** Kubernetes doesn't include built-in tools for monitoring or logging. We need to integrate third-party solutions or use other monitoring systems to keep track of the health and performance of their applications running on Kubernetes.
- **Security Policies:** While Kubernetes provides some security features, it doesn't enforce application-specific security policies. We need to implement security best practices for our containerized apps.
- **Application-Specific Configuration:** Kubernetes manages the deployment, scaling, and updating of applications, but it doesn't define or manage the specific configuration parameters of our application. Configurations such as environment variables or application-specific settings must be handled by the application itself.

- **Continuous Integration/Continuous Deployment (CI/CD):** Kubernetes does not handle the entire CI/CD pipeline. CI/CD tools are used separately to automate the build, test, and deployment processes. Kubernetes is often integrated into these pipelines for deploying and managing applications.

2) What other Orchestration tools are available other than Kubernetes?

Answer: There are many container orchestration tools, each with its own features and benefits. Here are some:

- **Docker Swarm:** Docker Swarm is a native clustering and orchestration solution for Docker. It's relatively lightweight and easy to set up, making it suitable for smaller-scale deployments.
Key Features: Simplicity, integrated with Docker, supports Docker Compose files.
- **Apache Mesos:** Apache Mesos is a general-purpose cluster manager that can handle various workloads, including containers. It provides resource isolation and sharing across distributed applications.
Key Features: Support for diverse workloads, scalability, fault tolerance.
- **Amazon ECS (Elastic Container Service):** ECS is a fully managed container orchestration service provided by AWS. It simplifies the deployment and management of containers on AWS infrastructure.
Key Features: Integration with AWS services, ease of use, fully managed service.
- **OpenShift:** OpenShift, developed by Red Hat, is an enterprise Kubernetes platform with additional features for developer and operational tooling. It provides a comprehensive solution for building, deploying, and managing containerized applications.
Key Features: Integrated developer tools, security features, extensive enterprise support.
- **Nomad:** Nomad is an orchestration platform developed by HashiCorp. It supports the deployment and management of applications, including containers, VMs, and standalone executables.
Key Features: Simplicity, flexibility, supports multiple workload types.
- **Rancher:** Rancher is an open-source container management platform that supports multiple orchestrators, including Kubernetes, Docker Swarm, and Apache Mesos. It provides a user-friendly interface for managing containerized applications.

Key Features: Multi-cluster management, ease of use, and support for various orchestrators.