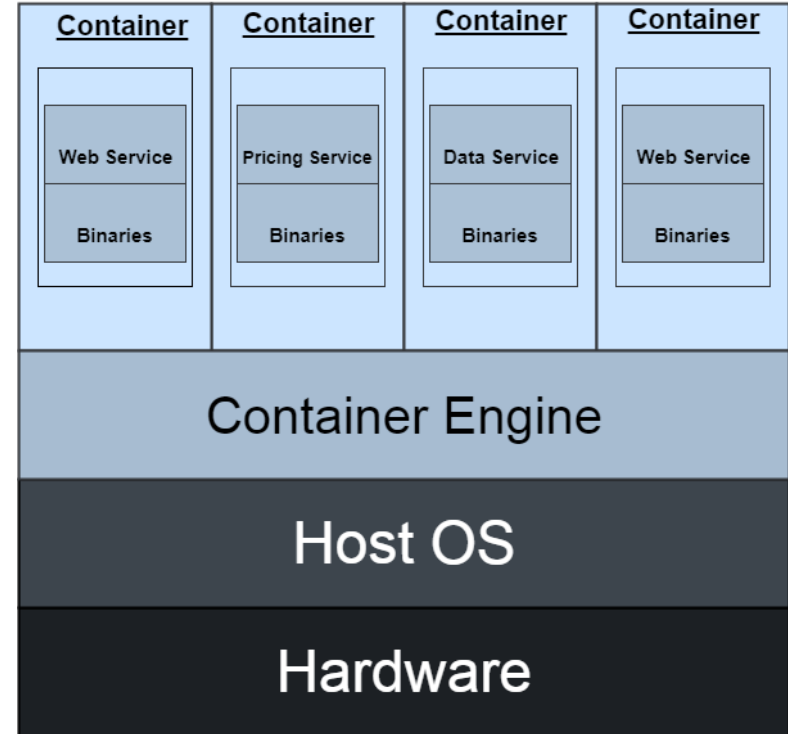


Cloud Containers

Concept Visualization



Review of Virtualization Technology

- Comparing the illustrations
- What is distinct
- Motive
- Approaches and technologies
- Let's delve deeper

What are Containers?

- Basically a means to bundle or package software in a portable and detached environment.
- It includes:
 - all the relevant software
 - dependencies to run application
 - Application code, libraries, runtime
- Though containers are similar to virtual machines, they are
 - lightweight
 - efficient
 - portable

Containers

- Containers provide a consistent and reproducible environment across different computing environments.
- This makes them ideal for deploying and scaling applications in cloud environments

Containerization Technology

- Containers use containerization technology.
- In this technology, the application and its dependencies are packaged together.
- Some popular containerization technologies include
 - Docker
 - Containerd
 - Podman

Isolation in Containers

- Containers provide process and file system isolation
- This enables applications to run independently of the underlying infrastructure.
- Each container has its own filesystem. This ensures that dependencies do not conflict.

Portable

- Containers are highly portable across different environments, such as development, testing, and production.
- They encapsulate everything needed to run an application, making it easy to move applications between different cloud providers or on-premises environments.

Orchestration

- Container orchestration tools, such as Kubernetes and Docker Swarm, manage the deployment, scaling, and operation of containers in a cluster.
- These tools automate the deployment and management of containerized applications.

Microservices Architecture

- Containers are often used in a microservices architecture, where an application is decomposed into smaller, loosely coupled services.
- Each microservice runs in its own container. This enables
 - scalability and
 - ease of management.

Resource Efficiency

- Containers share the host operating system's kernel, reducing overhead compared to virtual machines.
- They start quickly and consume fewer resources, making them efficient in terms of both startup time and resource utilization.

Sample Containerization platform

- **Docker**
- A widely used platform for containerization.
- It provides tools to build, ship, and run containers.
- Docker images are used to package applications and their dependencies.
- Docker containers are instances of those images.

Container Registries

- Container images are stored in container registries, which are repositories for sharing and distributing container images.
- Examples of container registries include:
 - Docker Hub,
 - Google Container Registry,
 - and Amazon Elastic Container Registry (ECR)

Immutable Infrastructure

- Containers promote the concept of **immutable infrastructure**.
- This implies infrastructure components, including containers, are never modified after they are deployed.
- Instead, new instances are created with the necessary updates.

DevOps Integration

- Containers play a crucial role in DevOps practices.
- They enable continuous integration and continuous deployment (CI/CD).
- Containers facilitate consistent testing and deployment pipelines, reducing discrepancies between development and production environments.

Real life Applications

- Analytics in real-time
- Real-time streaming
- Real-time Communication
- Real-time gaming

Conclusion

- Cloud containers provide a flexible and scalable approach to application deployment.
- This allows developers to build, package, and distribute applications efficiently.
- Container orchestration tools further enhance their manageability and scalability in cloud environments.
- Containers are expected to play very useful roles as cloud computing evolves

Questions?

