

Docker container-Container vs VM

Containers

- A lightweight and portable way to deploy applications
- Multiple isolated environments run on a single host OS instance
- Each container shares the same kernel as the host machine
- Containers are essentially running processes with their own file system, network stack, and memory space
- They are designed for fast deployment, scalable, and efficient resource utilization

Virtual Machines (VMs)

- A complete, self-contained operating environment that runs on top of a host OS instance
- Each VM has its own kernel, file system, network stack, and memory space
- VMs provide strong isolation between guest environments
- They are designed for security, legacy system support, and resource utilization similar to physical hardware

Example

Suppose you have an e-commerce application that needs a MySQL database server. With Docker containers:

1. You would create two separate container images: one for the web application (e-commerce app) and another for the database server (MySQL).
2. Each container image is built with its own dependencies, configuration, and software stack.
3. On your host machine, you can run both containers in isolation using Docker Engine, sharing the same kernel but having separate file systems, network stacks, and memory spaces.

Example commands:

```
docker build -t ecommerce-app .
docker build -t mysql-server .

docker run -d --name ecommerce-container ecommerce-app
docker run -d --name mysql-container mysql-server
```

In contrast, if you were using virtual machines (VMs):

1. You would create separate VM images for the web application and database server.
2. Each VM image would have its own complete operating environment, including kernel, file system, network stack, and memory space.
3. On your host machine, you can run both VMs in isolation, using a hypervisor like VirtualBox or VMware.

Example commands:

```
# Create separate VM images for web app and database server
virtualbox create ecommerce-vm --image ubuntu-x86_64
virtualbox create mysql-vm --image ubuntu-x86_64

# Run both VMs in isolation
virtualbox run ecommerce-vm
virtualbox run mysql-vm
```

In summary, containers (with Docker) offer a lightweight and portable way to deploy applications, while virtual machines provide strong isolation between guest environments. Choose containers for fast deployment, scalable resources, and efficient resource utilization, and choose VMs for security, legacy system support, and resource utilization similar to physical hardware.

Curated by Brajesh Kumar