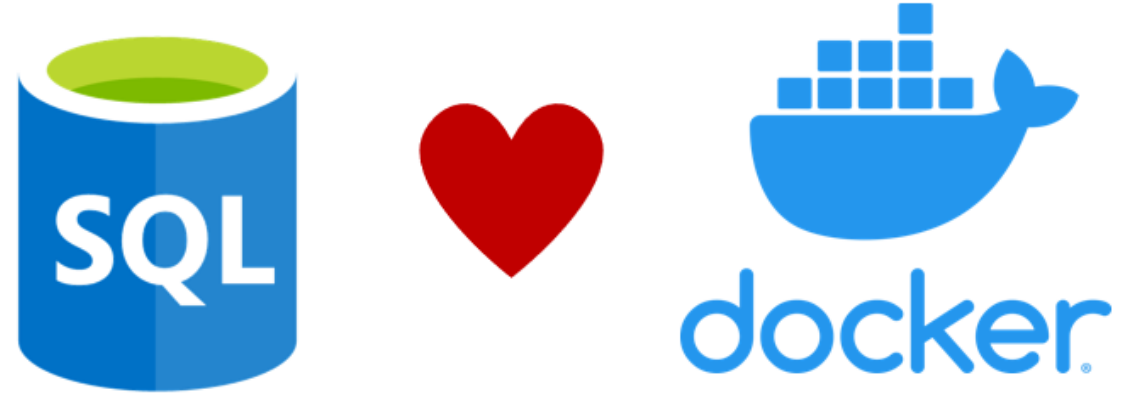


Docker

SQL Containers for Data
Professionals

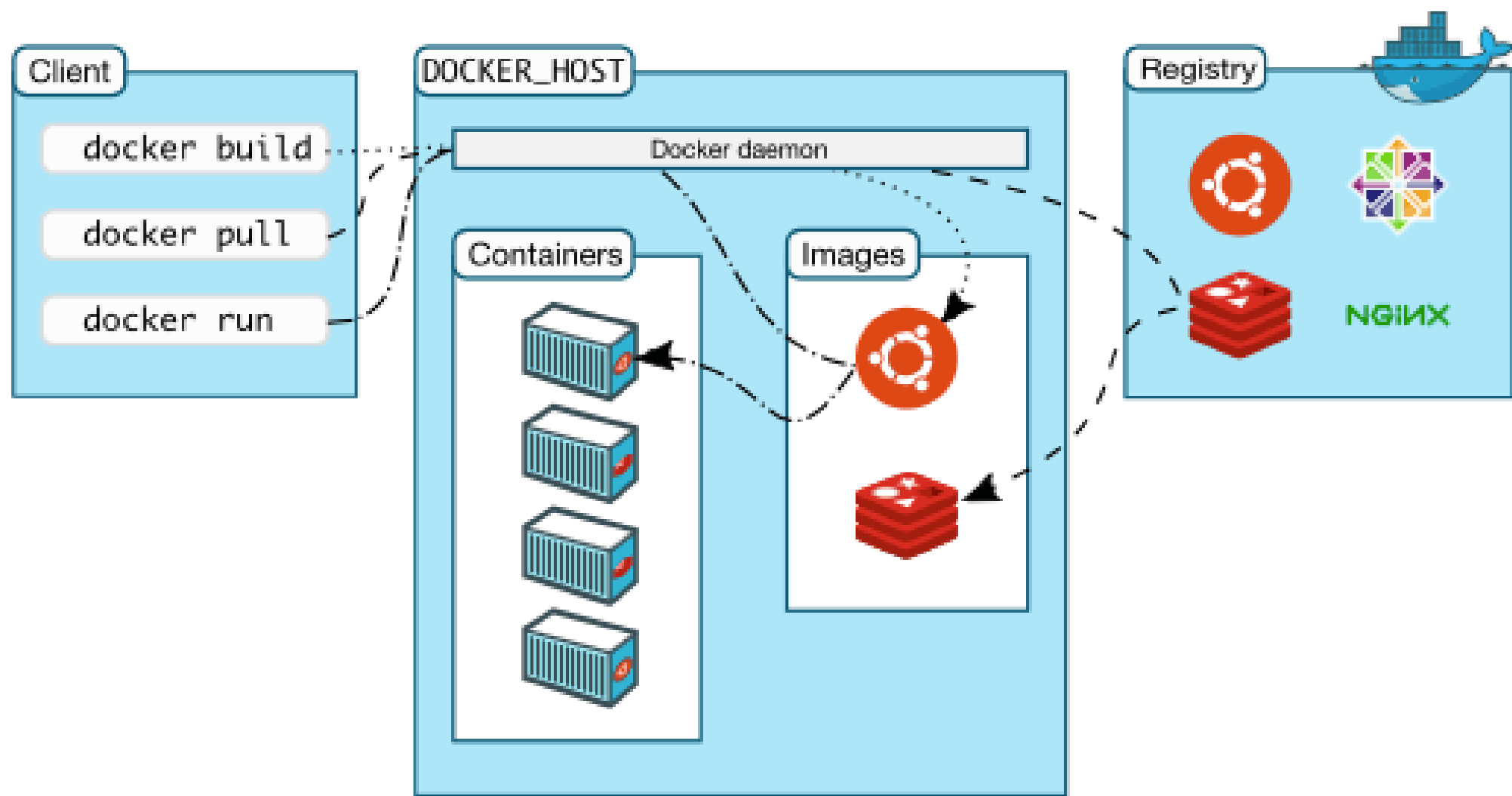


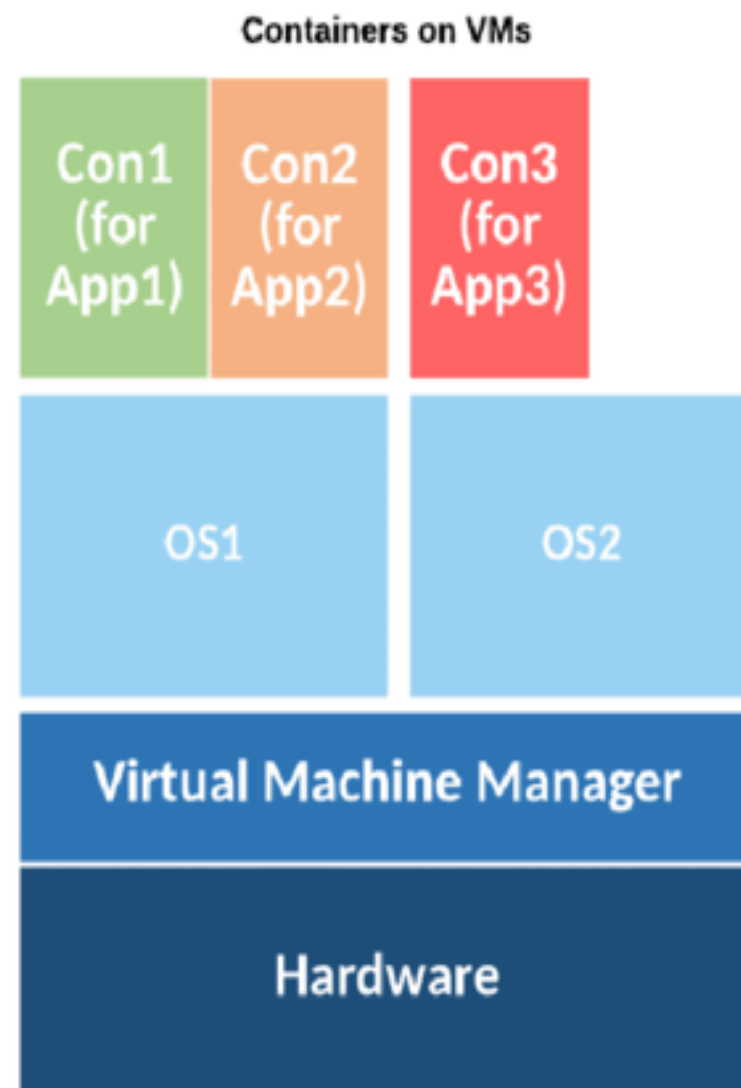
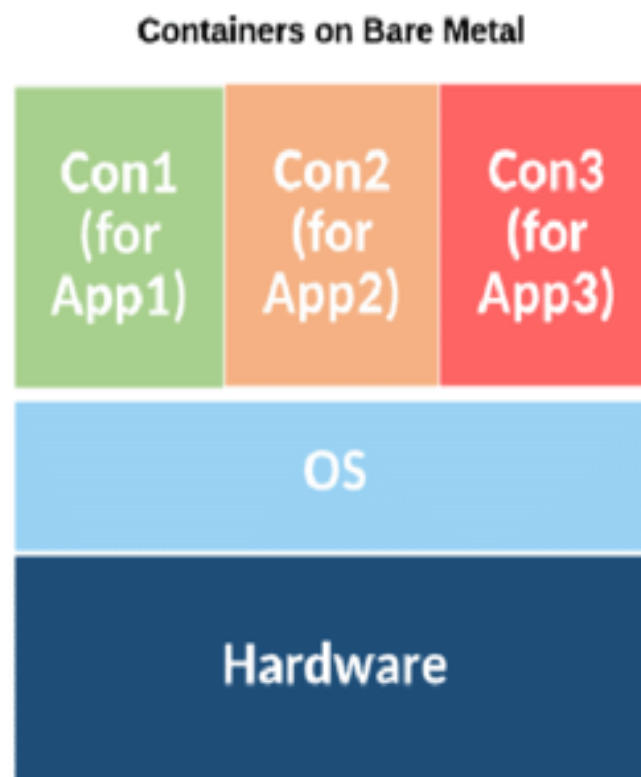
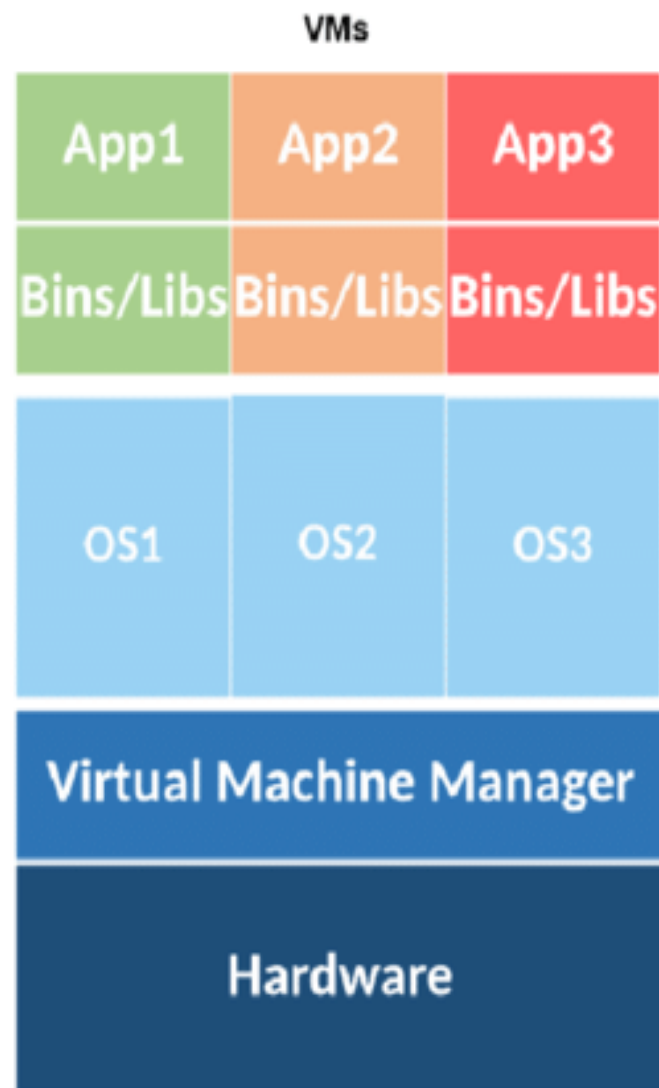
What is Docker Container?

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

Docker Architecture

Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon can run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface. Another Docker client is Docker Compose, that lets you work with applications consisting of a set of containers.

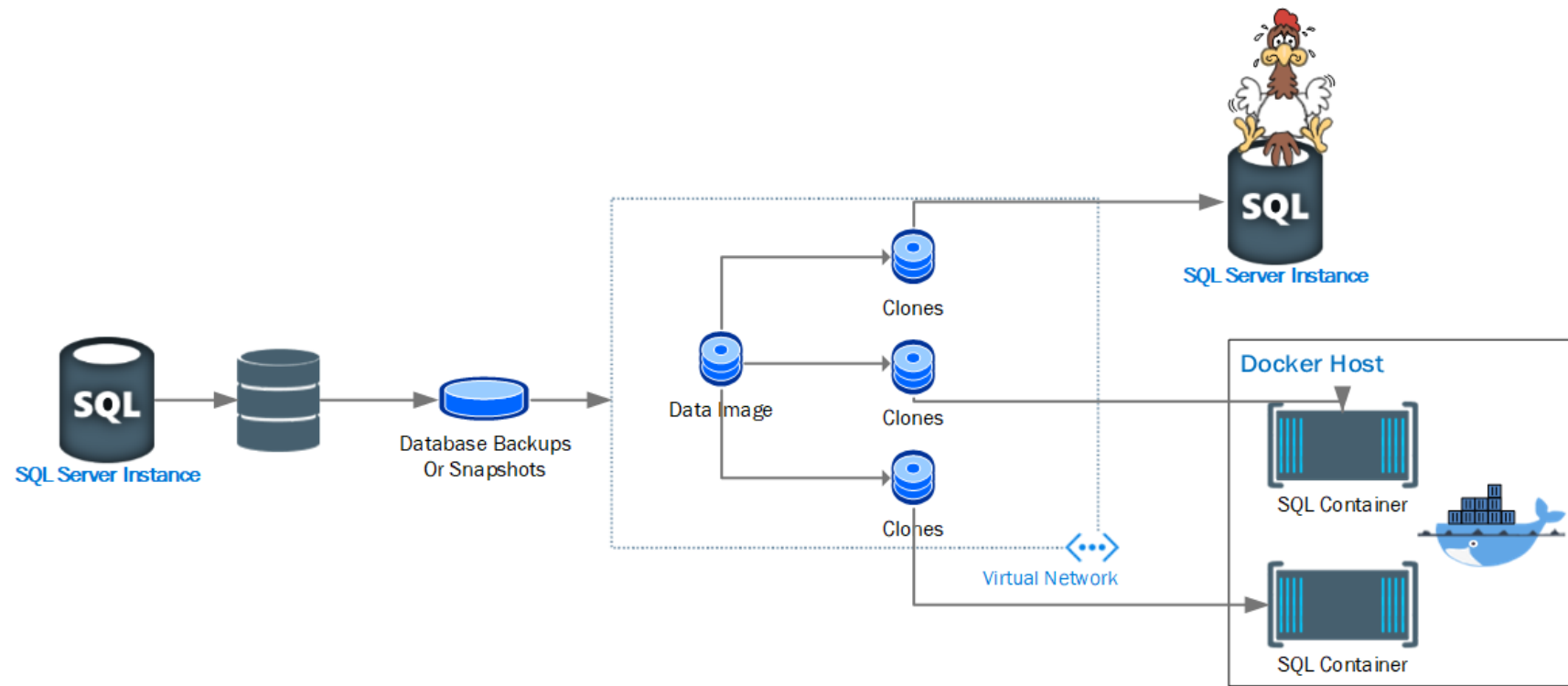




Container for SQL Developers

Probably right now you are asking, why we need to use SQL containers? If we have LAB or Dev environments, or local SQL Server instances.

1. **Savings:** SQL developers adopting containers can help to reduce between 8x to 10x the number of SQL Servers hosted in our networks, reducing the licenses costs for the company.
2. **Speed:** This is one of the most important features of the containers. We can deploy or mount a 1TB database server in a few minutes, in comparison to installing, setup, and restore a VM or Physical server.
3. **New code or versions of the databases:** We can incorporate into the images changes not affecting other environments, and each developer can work on the version that they need.
4. **Fast delivery of Code:** Using containers to test code in isolated mode will reduce the errors on the Dev or LAB env, providing safe code to the CI/CD process.



Ingredients

1. Docker Desktop Application.
2. Azure Data Studio or Visual Studio Code.
3. Scripting Tool like PowerShell or Programming Language like Python.
4. Docker File (Not 100% Required).
5. Putty or SSH Tool can be useful to debug the Linux OS.

Docker Main Commands

```
docker pull mcr.microsoft.com/mssql/server:2019-latest
```

```
docker run -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=C0ntainers2021' -p 11433:1433 --name staging -h staging `
-d mcr.microsoft.com/mssql/server:2019-latest
```

```
# Check docker images
docker ps -a
```

```
# Start a Docker container.
docker start <docker id or name>
docker run <docker id or name>
```

```
# Stop and unload a docker container
docker stop <docker id or name>
docker rm <docker id or name>
```

Demo