

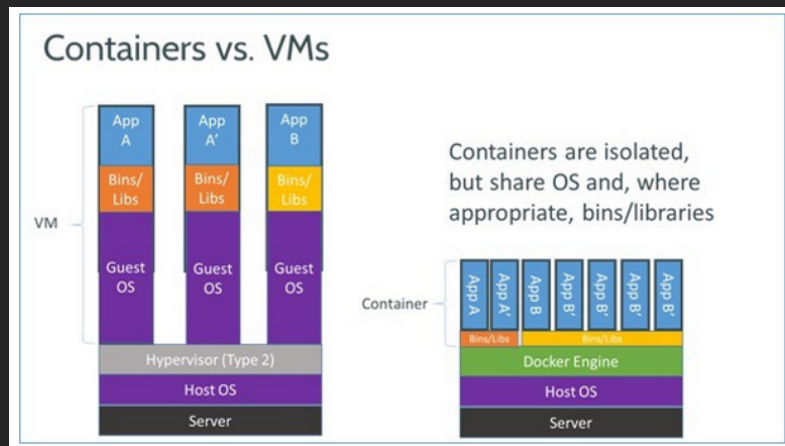


Docker Certified Associate

Docker Communication

Dockerized Application Communication

Essentially, Docker containerized applications communicate with all other applications and infrastructure exactly the way you would expect any other application running in your environment with a few exceptions. Consider the Docker diagram below:



Side by side, the architecture of a Docker containerized application externally does not differ from what you might see on more traditional physical or virtual deployments.

However, there are a couple of caveats to keep in mind so that all of your applications know how to find each other.



Communication Caveats on Docker Containers

A few items to consider with Dockerized applications when putting them in place in your IT ecosystem:

- **Routing**
 - Either you will have to expose the Docker application/service via port redirection when you launch them OR you will have to provide a routing mechanism (statically) to the container network on their host(s).
- **Port Redirection**
 - As mentioned above, managing ports either through passing them directly through or redirecting them to the underlying host on different ports, will affect how they behave in your environment.
- **Portability**
 - Making sure that data is external to the container application (on the host or via a network share) can have (sometimes significant) impacts on their performance.



Containers Should Be...

Abstract

- You containerize an application so that it remains removed from other components in the stack. That abstraction makes updates easier, but does introduce additional potential areas that communication can fail.

Portable

- Applications that are containerized can be picked up and put almost anywhere and will be consistent in their content and behavior. However, re-establishing communication with other components in the stack will depend on the new location and how you have planned access to that resource.

Flexible

- Containers give you almost limitless flexibility. Be sure to try and refrain from launching your container services tied too closely to external (and changeable) variables (hard coded related IP addresses or hostnames not easily changed).

