Software Containers

Why and how?

Brian, FOGGIE collaboration meeting, 4/22/2025

Questions I’m going to try to answer

- What is a container and how is it different than a virtual machine?
- Why are containers useful?
- How do I set up a container on my own computer or on a supercomputer, and how can I interact with it? (Demo/interactive part!)
- How do I set up a container to use with VSCode?
- What is Singularity, how is it different than Docker, and how do I move from one to the other?
- How do I use Singularity on a supercomputer?

What is a container?

- A tool that allows developers to package and execute an application in a software environment that is consistent and (mostly) isolated from the underlying system and from other processes/containers on that system.
- It has its own OS, software stack, supporting data, and internal directory structure, but still communicates directly with the system hardware.
- It has less overhead/requires fewer resources than a virtual machine, which typically emulates hardware as well as software and has much tighter constraints on its behavior (both are useful, but in different circumstances!)
- Docker is a widely-used open-source containerization platform, but there are several alternatives! (e.g., Podman, LXC, Buildah)

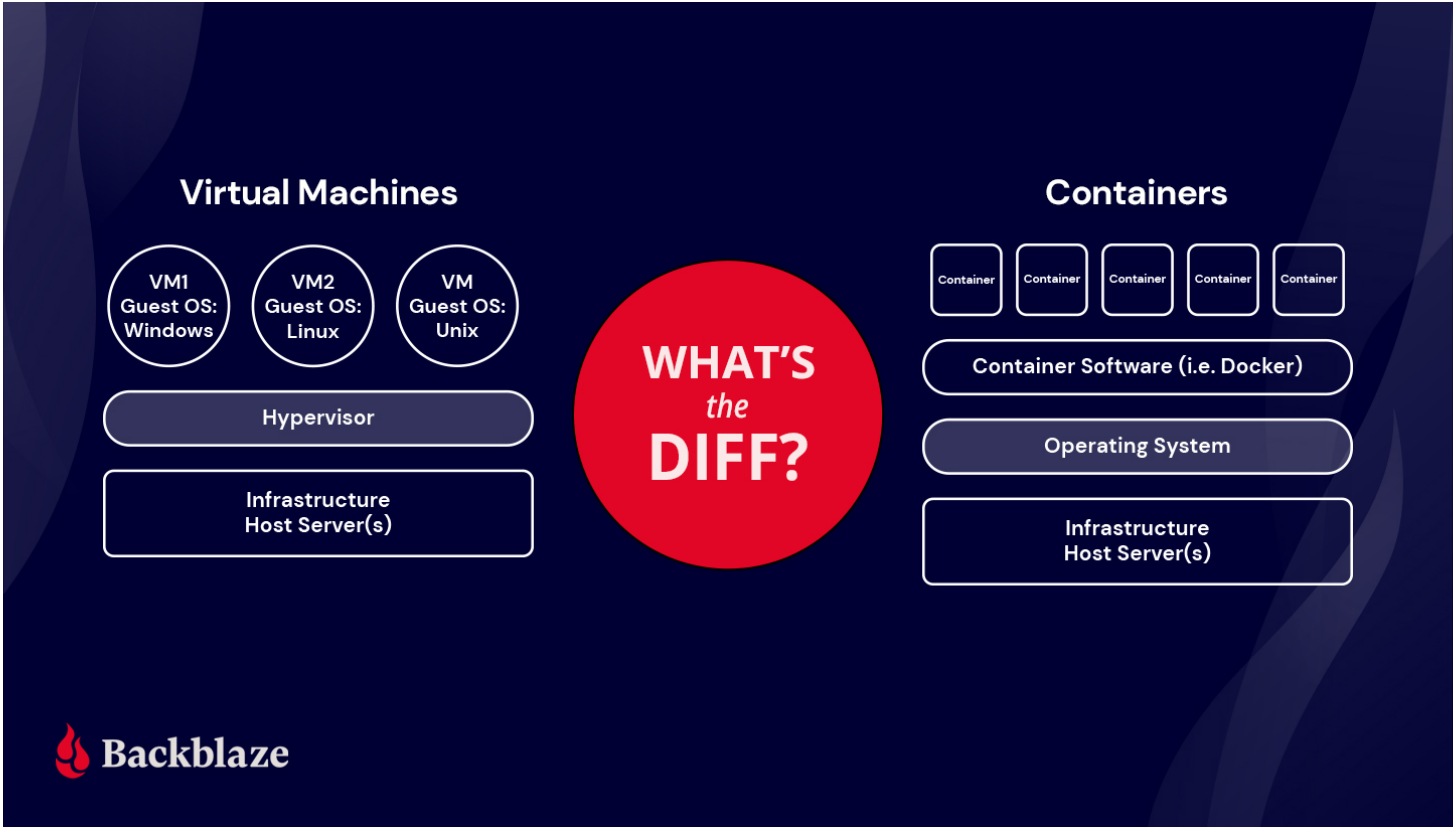

Image from a very useful blog post: https://www.backblaze.com/blog/vm-vs-containers/

When might you use a container?

- When your software has lots of very specific dependencies (i.e., needs specific versions of libraries)
- When your software is challenging to run on your operating system of choice (e.g., Charm++ on OS X)
- When you want to experiment with new Linux versions, compilers, libraries, applications, etc. without messing up your existing setup.
- When you have too many versions of some program/library/header file on your system and want to isolate your application so it can only use one.
- When you are annoyed with OS/XCode/MacPorts upgrades breaking your Enzo(-E)/Kokkos/etc. installation AGAIN
- When you want to run a bunch of different versions of something (e.g., the same code with different compilation options or using different compilers) for some reason and want them to be independent of each other.

How do I use Docker?

1. Download and install Docker (at docker.com)
2. Go into the FOGGIE repository (to foggie/containers_tutorial)
3. Look at the README.md file!
4. Follow along!

(This is the interactive part!)

Using Docker with VS Code

- It's really easy!
- Install Docker (or figure out where it's installed on a remote machine)
- Install the VS Code Docker Extension
- You can then do any of the following:
    - Use the VSCode Containers tutorial (https://code.visualstudio.com/docs/devcontainers/tutorial) and try an example development container
    - Open an existing folder/project on your machine or a remote machine in a container (using either a base Dev Container template or a custom Dockerfile)
    - Open a Git repository in an isolated container volume

See https://code.visualstudio.com/docs/devcontainers/containers

and https://www.youtube.com/watch?v=bUhjY2L1iFc for more info

Singularity and Rodman - Docker, but for supercomputers!

- Docker requires root access, which is not typically possible on supercomputers (for good reasons)
- Singularity (https://sylabs.io/) and Podman (https://podman.io/) are container environment that works on supercomputers. Singularity is supported by the DOE Exascale Computing Project (among others). It supports MPI across multiple cores+nodes and MPI+GPUs (which Docker does as well!). Podman supports most of this as well, but not on Pleiades!
- You can use Dockerfiles to create Singularity and Podman images very easily, and there are tons of Singularity and Rodman images available as well (see, e.g., https://e4s.io/ for Singularity).
- See DOCKER_NOTES.md and links to resources within that for examples of how to use these tools!