Software Defined Networking

Dr. Nick Feamster
Professor

In this course, you will learn about software defined networking and how it is changing the way communications networks are managed, maintained, and secured.

Containers and Docker

- Overview of Containerization
  - What are containers?
  - Why are they useful?
- Examples of Containerization

Containers

- Operating System-level virtualization
  - Run multiple isolated Linux systems on a single host
  - Unprivileged containers allow users to run software on the host without accessing hardware
- Previous versions
  - OpenVZ
  - Linux vServer

Why Containers Are Useful

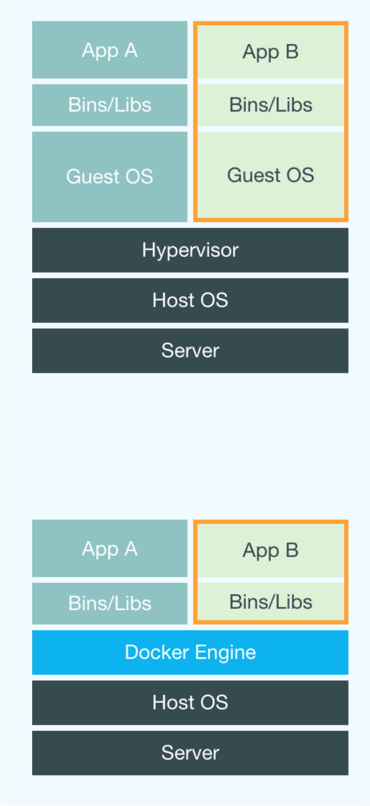
- **Portability:** Develop in a contained host environment, deploy on any host that is running a Docker host.

- **Isolation:** Application running in one container is isolated from others.

Differences from Virtual Machines

- Virtual machine not only has the application and binaries/ libraries, but also a guest OS
  - Many Gigabytes, slow to load

- Container has only the applications and its dependencies.



https://www.docker.com/whatisdocker/

Comparison to Virtual Machines

- Lower Overhead
  - Direct use of OS system calls
  - More efficient than emulation
- Less Flexibility
  - Guest OS different than host OS
- File-level Copy on Write
  - Easier backup, simpler caching, etc.

How is Docker Used?

- Distributed applications
- Continuous integration/delivery
- Platform as a Service
- Application deployment

Conclusion

- Containers are lightweight means of deploying applications
  - Faster to deploy, more lightweight than VMs

- A possible vehicle for deploying network functions, such as those we have heard about in a previous lecture.