Deploying & Managing Containerized Workloads in the Cloud

Janakiram MSV
Principal Analyst

Janakiram + Associates
janakiram.com

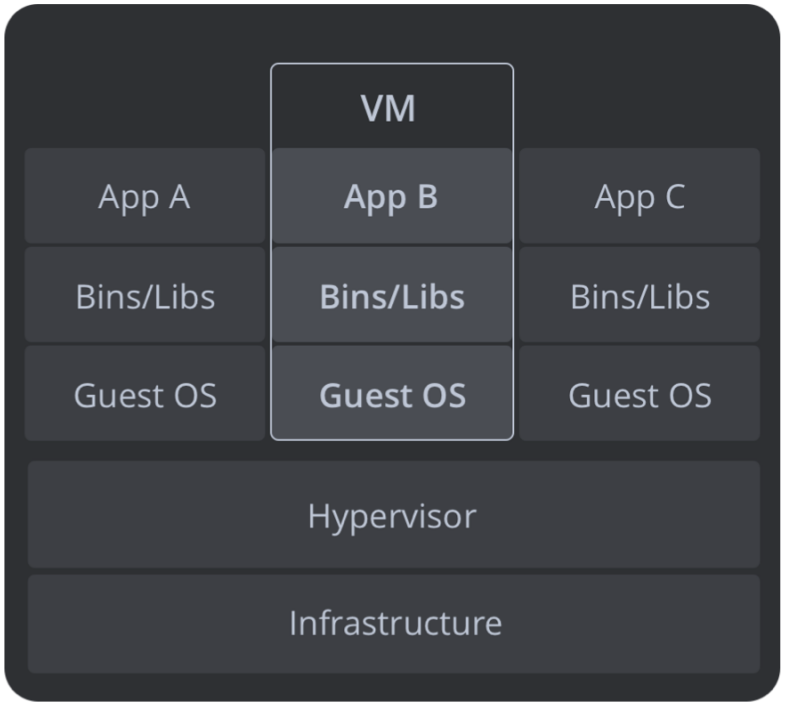
Session 1

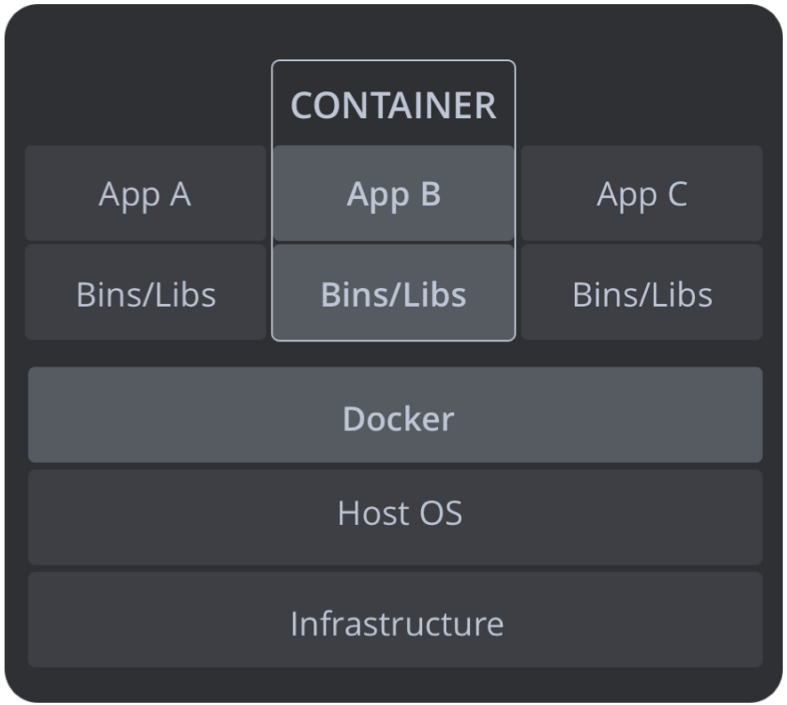
Installing & Configuring Docker

Agenda

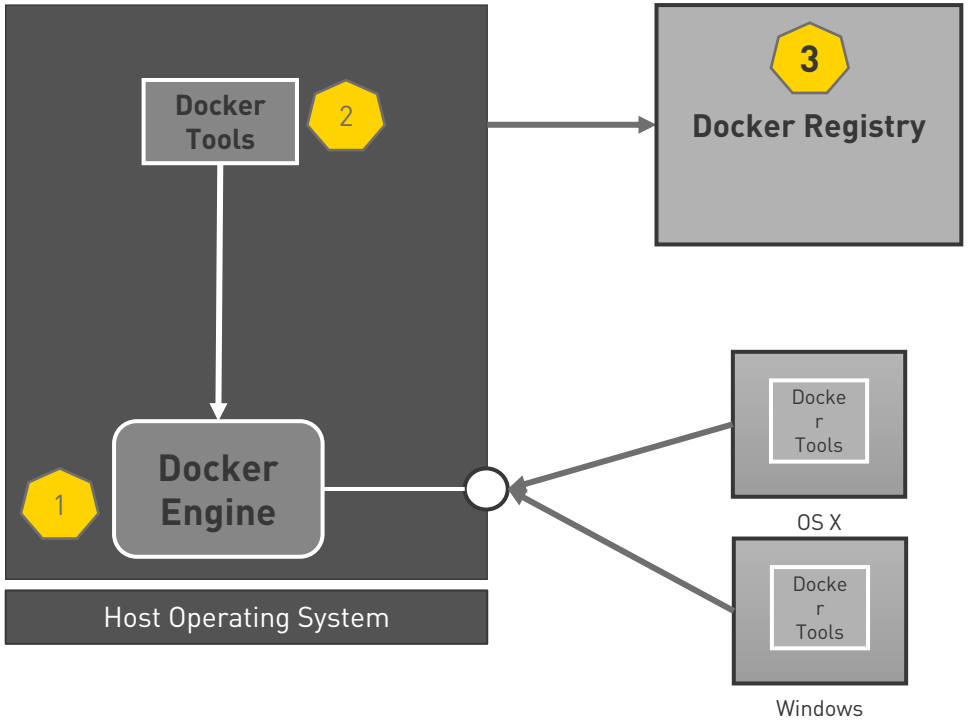
- Docker Architecture
- Launching Containers
- Managing Images
- Setting up Docker Registry

Virtual Machines vs. Containers





Docker Architecture



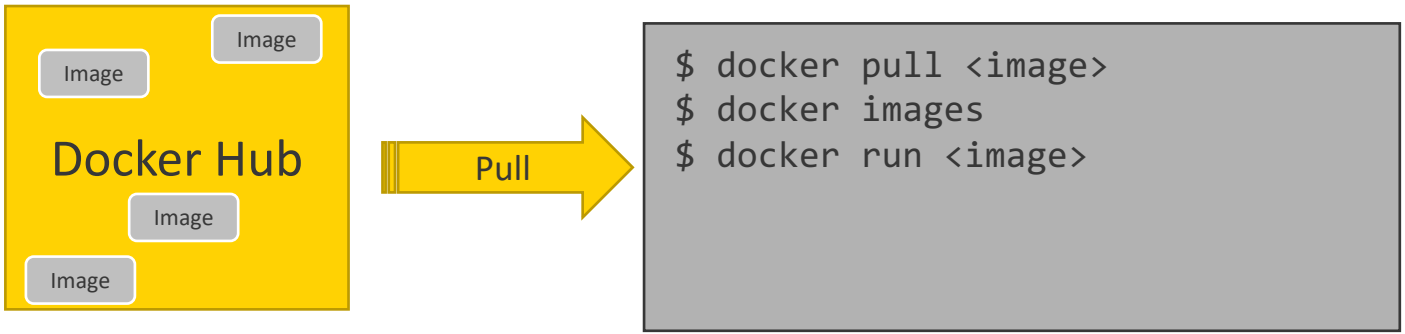
1. **Docker Engine**
   - Runs in Linux as a background process
   - Exposes REST API
   - Interfaces with the kernel
2. **Docker Tools**
   - Client to the Docker Engine
   - Manages interactions
   - Can run on Windows and OS X
3. **Docker Registry**
   - Stores and distributes Docker images
   - Can be private or public
   - Runs on a variety of storage backends

Demo 1

Installing Docker

Launching Your First Container

- Pull the required image from the Docker Hub
- Launch the container locally
  - Add optional parameters to *docker run* command



Demo 2

Launching Containers

Adding Storage to Containers

- Containers have ephemeral storage
- Data stored within the container is not accessible after it is terminated
- Volumes provide persistence to containers
- Each volume is implicitly or explicitly mapped to a host directory

Container

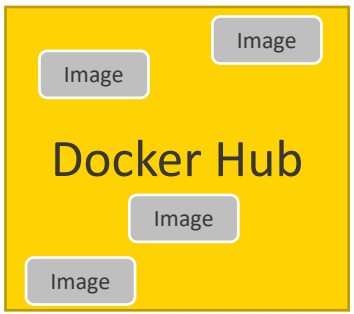
```
# Method 1
$ docker volume create data
$ docker run -v data:/data <image>

# Method 2
$ docker run –d –v /opt/data:/data <image>
$ docker volume ls
```

Demo 3

Dealing with Persistence

Creating Custom Images

- New Images can be created from running containers
- Stop the container and commit the container to an image
- Tag the image and push it to Docker Hub




```
$ docker pull <image>
$ docker images
$ docker run <image>
$ docker stop <container>
$ docker commit <container>
$ docker tag <user>/<image>:<tag>
$ docker login
$ docker push
```

Demo 4

Creating Container Images

Configuring Private Registry

- Registry holds container multiple tagged versions of images
- Private registry keeps the images in a private environment
- Images are kept close to the Docker Engine
- Docker Registry is available as an image in Docker Hub
- Registry container storage should be based on a persistent volume

Demo 5

Launching a Private Registry

Summary

- Docker makes it easy to deal with containers
- Docker has an engine, tools, and registry
- Docker Hub is the public registry to store images
- Images are pulled from Docker Hub to run on the host
- Volumes bring persistence to containers
- Custom images can be built from base Docker images
- Private registry keeps the images secure and closer to the engine
- Leverage DigitalOcean Block Storage for persistence
- Configure DigitalOcean Firewall for securing the ports

Key Docker Commands to Learn

1. docker version
2. docker images
3. docker pull
4. docker run
5. docker exec
6. docker ps
7. docker volume
8. docker inspect
9. docker stop
10. docker rm
11. docker commit
12. docker tag
13. docker push
14. docker save
15. docker load

Thank You!

Janakiram + Associates

janakiram.com