Implementing Containers Using Docker

Andrew Mallett

Linux Author and Trainer

@theurbanpenguin   www.theurbanpenguin.com

Overview

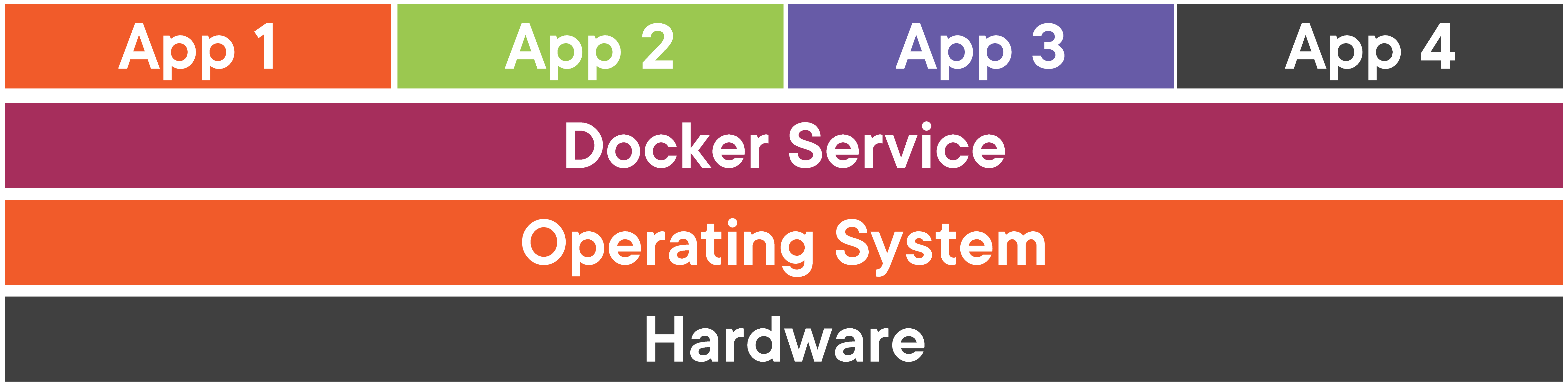
Containers and Docker

- Containers vs Virtual Machines
- Docker
  - Images
  - Dockerfile
  - Containers
  - Volumes
  - Ports

Containers

Containers are a way to isolate applications from the rest of the OS and other applications. A container image only needs the resources required to run the application.

Isolated Container Applications with Independent Dependencies



```
$ sudo -i
# ip link ls
# apt update
# apt install -y docker.io
# systemctl status docker
# ip link ls
# docker --version
# docker info
# docker run hello-world
```

Installing Docker on Ubuntu 20.04

If you haven't already installed Ubuntu on your Ubuntu system from the last module, you can install it now. The hello-world image can be executed as root to prove Docker is working.

Demo

Working on the Ubuntu 20.04 System:

- Ensure Docker is installed
- Use the hello-world image

Security

There is a Docker group created with full access to Docker. It may seem ideal to add users to the group to manage Docker. This does work but bypasses security logging, which may be an issue.

Sudo / Docker Group

Sudo

All operations are logged using sudo, malicious actions can be detected

Docker Group

If a user is a member of the group Docker, they can perform all operations with containers and they actions are not logged

```
$ sudo docker run -ti --name bad --privileged -v /:/host ubuntu chroot /host
$ sudo docker rm bad
```

Using Sudo Actions are Logged

The details of the command we will see later but we are running a Linux Ubuntu container, mapping the root volume of the host to the /host directory in the container. The container starts by executing the the command to chroot to the host directory. We now have full control over the host system. Without using sudo, this would not be logged.

Demo

Warning Against Using the Docker Group

- Although the most obvious, adding users the the Docker group may not be the most secure
- The demonstration is to make you aware of what can be done with Docker containers

```
$ sudo docker image --help
$ sudo docker image ls
$ sudo docker image inspect hello-world -f '{{ .RepoTags }}'
$ sudo docker image tag hello-world my-hello
$ sudo docker image ls
$ sudo find /var/lib/docker -name "<image id>*"
$ docker search ubuntu -f is-official=true
```

Docker Images

Images come from the Docker Image registry at hub.docker.com. The image is cloned to running containers as needed. Images are files in the file system based on the image id, but they are located by the image tag and additional tags can be assigned to the same image.

Demo

Working with Images

```
$ sudo docker run ubuntu
$ sudo docker ps ; sudo docker ps -a
$ sudo docker run -it --name u1 ubuntu
exit
$ sudo docker ps ; sudo docker ps -a
$ sudo docker run -it --name u2 ubuntu
CNTRL pq
$ sudo docker ps
```

Container Management

Containers are cloned from images with the docker run command. Assign a name to the container to avoid auto-generated name. The Ubuntu image runs bash by default and exit will stop the container. To keep the container running use CNTRL+pq

Demo

Managing Containers

```
$ sudo docker run -d --name web nginx
$ sudo docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' web
$ curl <ip address>
$ mkdir web; echo hello > web/index.html
$ sudo docker run -d --name web \
  -p 8000:80 -v /home/tux/web/:/usr/share/nginx/html nginx
$ curl localhost:8000
```

Container Services

Often, we need containers to host services. We can make life easier for use by mapping ports to the container and volumes.

Demo

Managing Container Based Services

- Port mapping
- Volume mapping

```
$ cd web ; vim Dockerfile

FROM ubuntu
RUN apt update && apt install -y nginx
EXPOSE 80/tcp
ADD index.html /var/www/html/
CMD ["nginx","-g", "daemon off;" ]

$ sudo docker build -t myweb .
```

Dockerfile

A Dockerfile (with uppercase D) can be used to specify the build of a new image. Don't forget the . at the end of the build command indicating the current directory. Here, we add the web content to the image, eliminating the need to map a volume but affording less flexibility.

Demo

Building Images

- Dockerfile
- docker build

Summary

Working with Docker Containers:

- **images** : Golden images for containers
- **containers** : Runtime instances of images
- **Dockerfile** : Build custom images
- **port maps** : Map host ports to container ports
- **volume maps** : Map host directories to container directories
- **security and the group, docker** : Can be a security hole

Managing Container Micro-Services Using Kubernetes