Postgres in a world of Docker Containers

Build, Ship and Deploy

Shaktisikha Sahoo
Lead QA Engineer

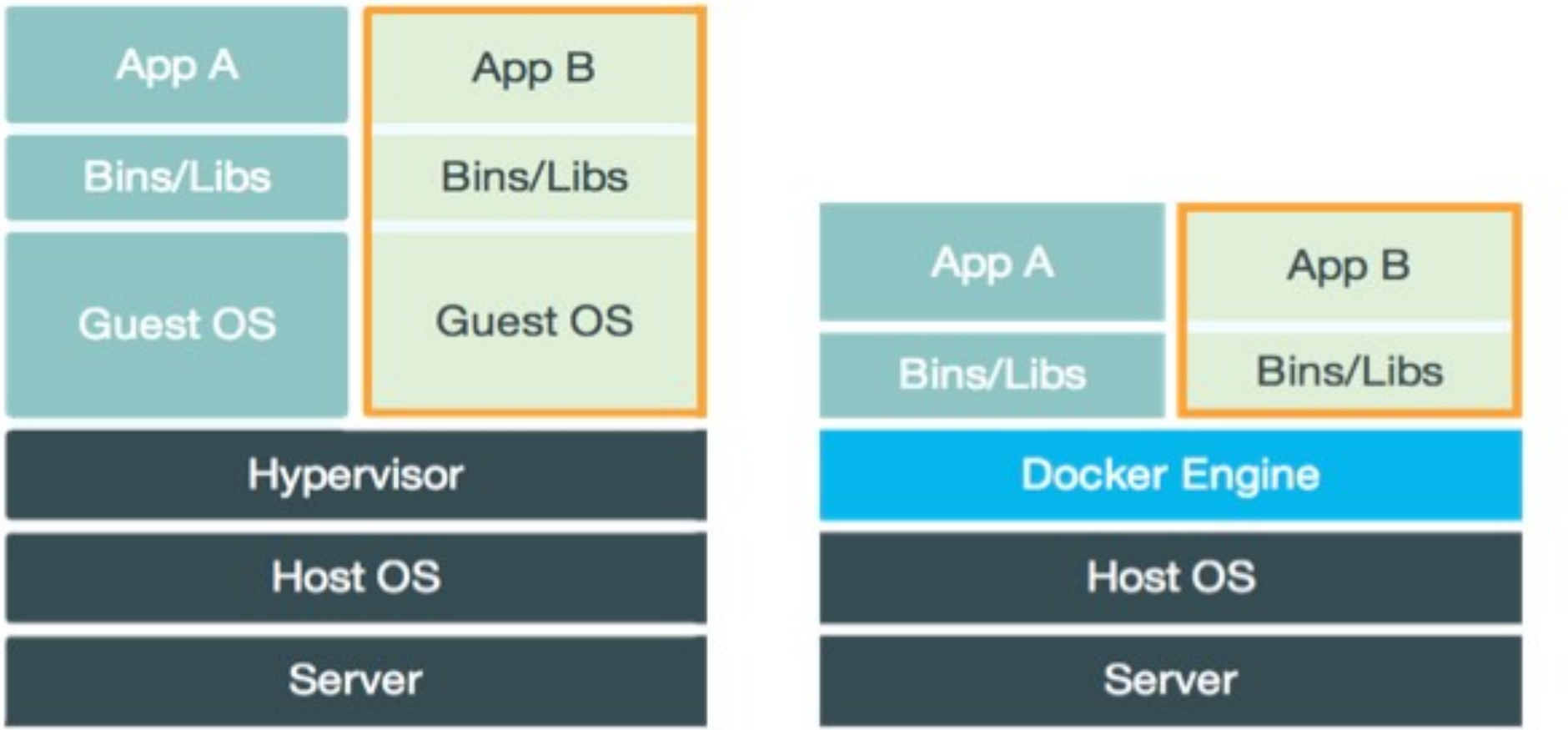
Outline

- A brief introduction to Docker.
- Docker over Virtual machines.
- How Docker works?
- Docker + PostgreSQL
- Live Demo

What is Docker??

- Open platform for developers and sysadmins to build, ship and run distributed applications.
- Can run on popular 64-bit Linux distributions with kernel 3.8 or later.
- Supported by several cloud platforms including Amazon EC2, Google Compute Engine, and Rackspace.

Virtual Machine Versus Container……



Key Benefits of Docker Containers

Speed

- No OS to boot = applications online in seconds

Portability

- Less dependencies between process layers = ability to move between infrastructure

Efficiency

- Less OS overhead
- Improved VM density

How does Docker work ?

- You can build Docker images that hold your applications.
- You can create Docker containers from those Docker images to run your applications.
- You can share those Docker images via Docker Hub or your own registry.

Docker Container Lifecycle ...

The Life of a Container :

— Conception
Build an Image from a Docker file
— Birth
RUN (create+start) a container
— Reproduction
COMMIT (persist) a container to a new image
RUN a new container from an image
— Sleep
KILL a running container
—Wake
START a stopped container
— Death
RM(delete) a stopped container

Extinction

— RMI a container image (delete image)

Containers & PostgreSQL

Containers provide several advantages to running PostgreSQL:

- Setup & distribution for developer environments
- Ease of packaging extensions & minor upgrades
- Separate out secondary applications (monitoring, administration)

Commands for Demo:

- Docker Installation:
  - Yum Install Docker
  - Systemctl docker start

```
mkdir -p $DATA_VOLUME
chmod -R 777 $DATA_VOLUME
chcon -Rt svirt_sandbox_file_t $DATA_VOLUME
```

- Create and Run Docker Container:
  - docker run --name "demo" -h "hostname" -e DATABASE_NAME="test" -e PGPORT=5444 -e DATABASE_USER="postgres" -e ENTERPRISEDB_PASSWORD="postgres" -e LOCALEPARAMETER="en_US.UTF-8" -e ACCEPT_EULA=Yes -e REPL_USER="repl" -e REPL_PASSWORD="postgres" -v /volumes/edbas:/edbvolume -p 1234:5444 -d <Image name>

- List all Docker images : Docker images
- List all running containers : Docker ps
- Login to the container : Docker exec -it <container> bash

Live Demo

Summary

- Easy to build, run & share containers
- Reduces complexity of system builds

- Docker containers + PostgreSQL gives you :
  - Easy-to-setup development environments
  - Your own production database-as-a-service
  - Tools to automate management of over 1000s of instances in short-order


EDB
POSTGRES
THANK YOU