



Containers

Himanshu Neema

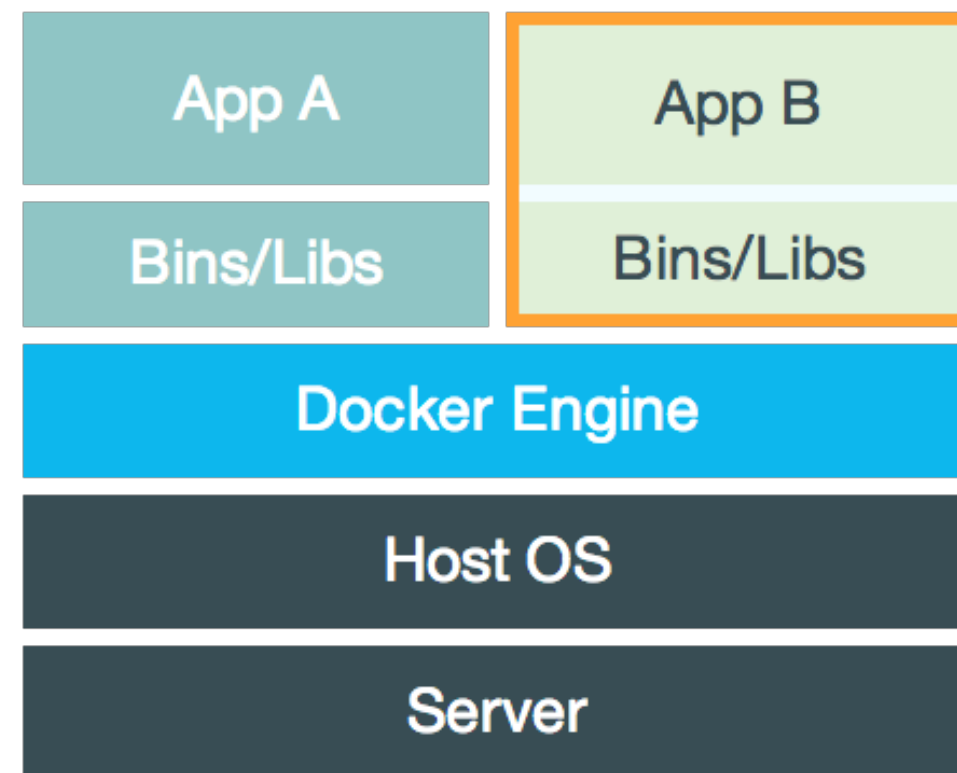
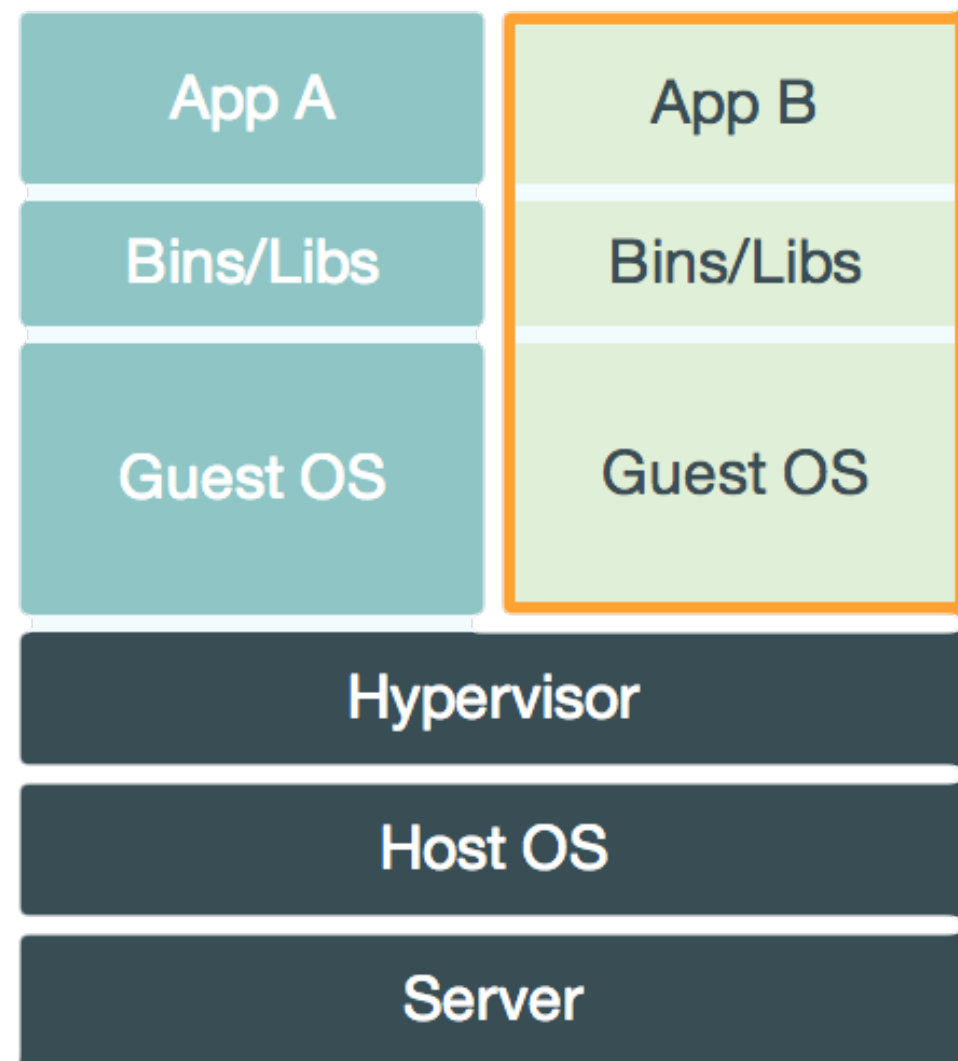


Shipping code: Linux Containers

- is a technology to contain application along with filesystem, libraries and other dependencies all into one package.
- lightweight process virtualization.

VMs vs Containers

- Hardware virtualization vs Operating System level virtualization



How does it work?

- Using Linux kernel features as building blocks:
mainly namespaces and cgroups

namespaces

- A namespace wraps a global system resource in an abstraction that makes it appear to the processes within the namespace that they have their own isolated instance of the global resource.
- Linux provides following namespaces:
 - Cgroup Cgroup root directory
 - IPC System V IPC, POSIX message queues
 - Network Network devices, stacks, ports, etc.
 - Mount Mount points
 - PID Process IDs
 - User User and group IDs
 - UTS Hostname and NIS domain name

cgroups

- abbreviated from control groups, where group is collection of process.
- **Resource limiting** – limits group's resource usage: CPU, memory, disk I/O, network, etc.
- **Prioritization** – some groups may get a larger share of CPU utilization or disk I/O throughput
- **Accounting** – measures a group's resource usage, which may be used, for example, for billing purposes
- **Control** – freezing groups of processes, their checkpointing and restarting

Container runtimes

- Docker
- LXC (LDX tool for user experience)
- rkt from CoreOS

Lets see container in action -
<https://linuxcontainers.org/lxd/try-it/>

Docker Vs LXC

- **Single Process Vs Multiprocess:**
 - LXD aims for full system containers: run multiple processes
 - Docker aims for single process container, so if the app needs app, nginx, worker processes, etc. Run them in separate containers.
- **Stateless vs Stateful:**
 - Docker is designed to be stateless more than LXC, so if you created an Docker image its meant to run unmodified, any change is a new image.

Lets see Docker in
action

containers @ web scale

- How to deploy and manage containers to thousands of machines ? enter: Container orchestration tools:
 - Kubernetes
 - Docker Swarm
 - Apache Mesos

Secure <https://aws.amazon.com/ecs/>

amazon web services

Products Solutions Pricing Software Support Customers More English My Account

Amazon EC2 Container Service

Run containerized applications in production

Get Started

Product Details Getting Started Pricing FAQs Recruiting Release Notes

Amazon EC2 Container Service (ECS) is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances. Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure. With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your cluster, and access many familiar features like security groups, Elastic Load Balancing, EBS volumes, and IAM roles. You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs and availability requirements. You can also integrate your own scheduler or third-party schedulers to meet business or application specific requirements.

Introducing Amazon EC2 Container Service

aws.amazon.com/gp/aws/developer/registration/index.html

Microsoft Corporation [US] <https://azure.microsoft.com/en-us/services/container-service/>

Microsoft Azure

SALES 1-800-867-1389

Why Azure Solutions Products Documentation Pricing Partners Blog Resources Support

Azure Container Service

Deploy and manage containers using the tools you choose

- ✓ Create a container hosting solution optimized for Azure
- ✓ Use popular open-source, client-side tool
- ✓ Scale and orchestrate applications using DC/OS, Docker Swarm, or Kubernetes
- ✓ Migrate container workloads to and from changes

Free account >

Already using Azure? Try Container Service now >

Explore Container Service: [Pricing details](#) [Documentation](#)

Secure <https://cloud.google.com/container-engine/>

Google Cloud Platform

Why Google Products Solutions Launcher Pricing Customers Documentation Support

CONTAINER ENGINE

One-click Kubernetes clusters, managed by Google

[VIEW CONTAINER ENGINE DOCS](#) [VIEW MY CONSOLE](#)

Automated Container Management

Google Container Engine is a **powerful cluster manager and orchestration system** for running your Docker containers. Container Engine **schedules your containers** into the cluster and manages them automatically based on requirements you define (such as CPU and memory). It's **built on the open source Kubernetes** system, giving you the flexibility to take advantage of **on-premises, hybrid, or public cloud infrastructure**.

https://console.ng.bluemix.net/docs/containers/container_index.html

Bluemix Docs

services / Compute

IBM Bluemix Container Service

Documentation Resources

Getting started with IBM Bluemix Container Service

Updated: 24 February 2017

Use IBM Bluemix® Container Service to run Docker containers in a hosted cloud environment. Containers are virtual software objects that include all of the elements that an application needs to run. A container has the benefits of resource isolation and allocation, but is more efficient than, for example, a virtual machine. Containers help you build high-quality applications.

Getting started with container groups

From the catalog, select **Containers** and choose an image. Some images are provided by IBM and images can be added by the people in your Bluemix organization. Let's get started with an IBM provided image.

Getting started

- > About
- > Planning
- > Security
- > Setting up the CLI
- > Adding Docker images
- > Running short-term

Whats Next?

- Build simple RESTful web app.
- Build Docker container for it.
- Deploy on GCP, AWS, etc.
- Scale to thousands of requests/sec.

Thank You



all images from internet