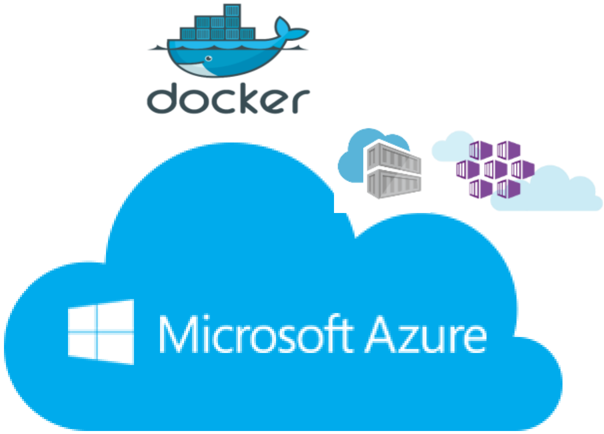
Containers on Azure

Ehsan Eskandari
IT Consultant
@EhsanEskandarim
http://passionatecoder.ca

#Azure, #Container, #Docker
Toronto Enterprise Devops User Group
June 8, 2017



Agenda

- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo

Agenda

- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo

What is Docker?

- Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment.

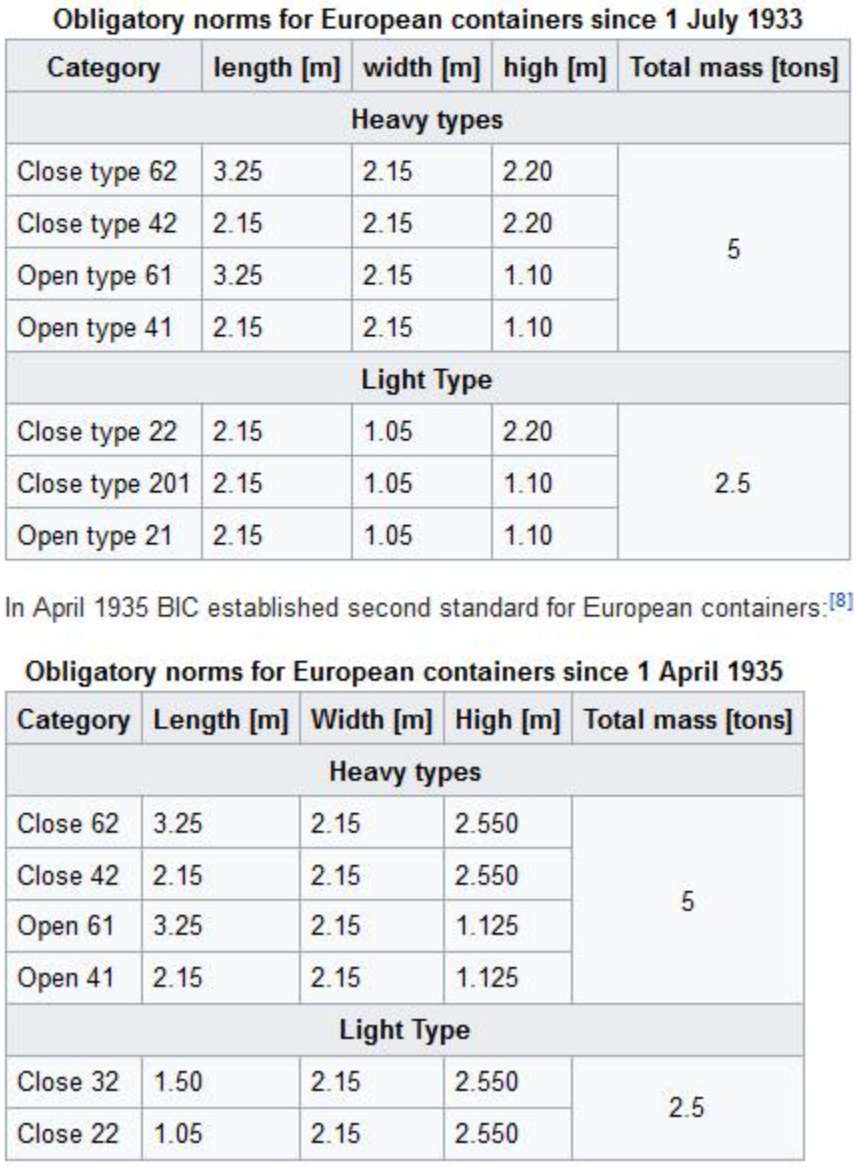
What is Docker?

- Lightweight, open source, secure platform
- Simplifies building, shipping, running apps
- Shipping container system for delivering code
- Runs natively on Linux and Windows Server 2016 or higher
- Relies on Images and Containers

What is Docker?

Loading cargo the old way

What is Docker?



Obligatory norms for European containers since 1 July 1933

Category	length [m]	width [m]	high [m]	Total mass [tons]
Heavy types				
Close type 62	3.25	2.15	2.20	5
Close type 42	2.15	2.15	2.20	
Open type 61	3.25	2.15	1.10	
Open type 41	2.15	2.15	1.10	
Light Type				
Close type 22	2.15	1.05	2.20	2.5
Close type 201	2.15	1.05	1.10	
Open type 21	2.15	1.05	1.10	

In April 1935 BIC established second standard for European containers:[8]

Obligatory norms for European containers since 1 April 1935

Category	Length [m]	Width [m]	High [m]	Total mass [tons]
Heavy types				
Close 62	3.25	2.15	2.550	5
Close 42	2.15	2.15	2.550	
Open 61	3.25	2.15	1.125	
Open 41	2.15	2.15	1.125	
Light Type				
Close 32	1.50	2.15	2.550	2.5
Close 22	1.05	2.15	2.550	

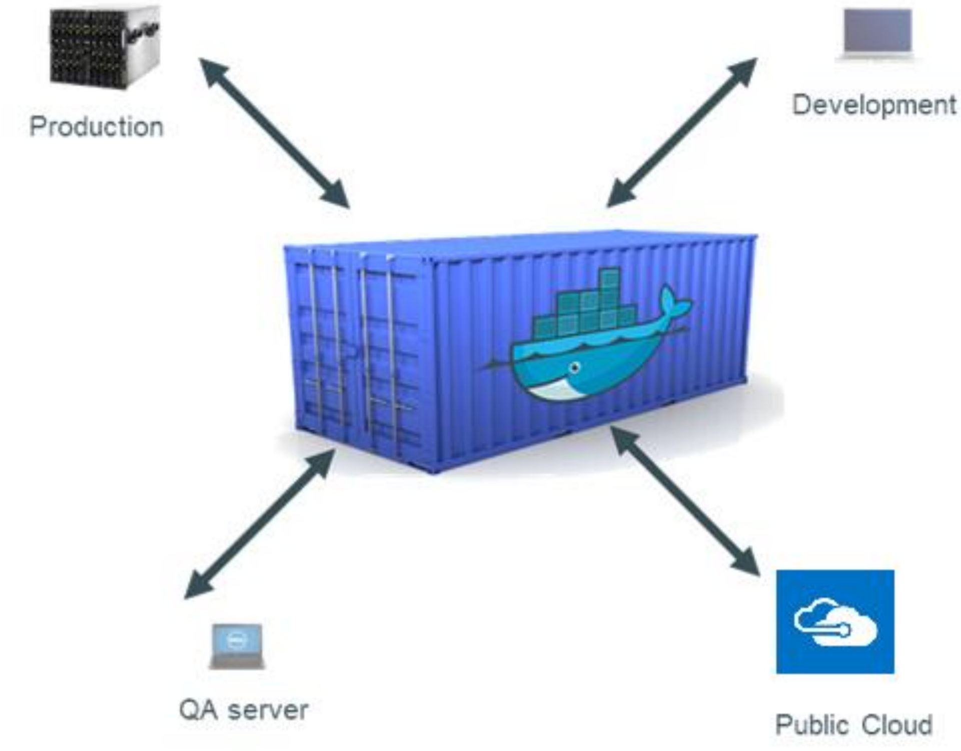
LIMAX
CASER
MAERSK SEALAND
MAERSK
53'
48'
45'
40'
20'

What is Docker?

Efficiency in

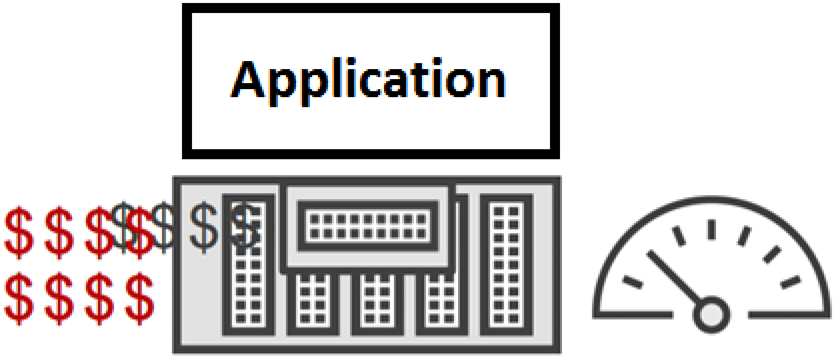
- Shipping Speed
- Shipping Costs

What is Docker?



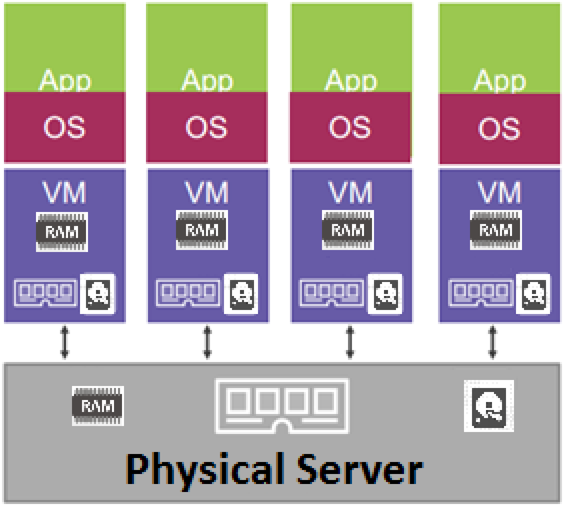
Servers in Old days

– One application on one physical server.
  - Procurement
  - Capex and Opex costs
  - Costs: Hardware, Power, Administration, etc.
  - Massively Over-Powered Servers with 10% CPU consumption



Virtualization and Hypervisor

- – Technology that enables us having Same Physical Server and squeeze so much more
- – Instead of one application on one physical server, now multiple application on one physical server.



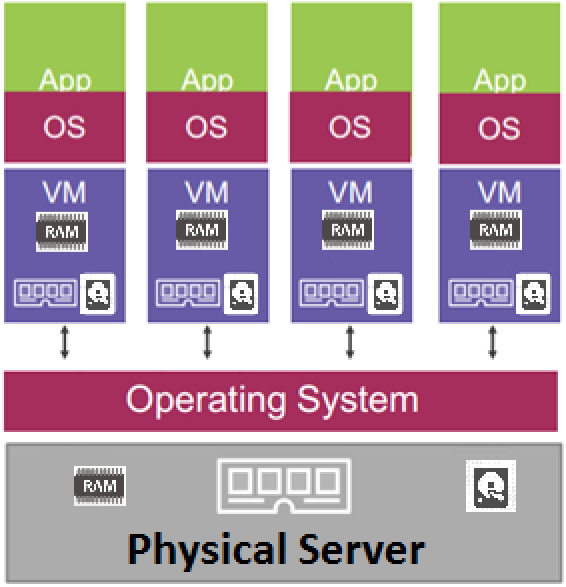
Virtualization Issues

- One physical server
- n virtual machines for n applications
- These vm are slices of physical hardware, e.g. allocating 25% of cpu power, 25% memory, 25% disk space
- These vm are slices of real resources in the physical server
- Each of these vms needs a dedicated OS and each of which uses cpu, memory, and disk
- So far it is just OS and there is no application yet
- Each OS needs license $$$$
- Each OS needs admin e.g. patching, anti-virus

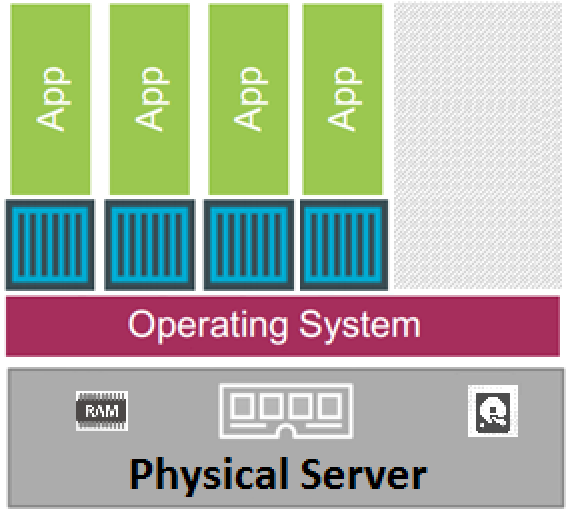
Docker containers

- **Instead of installing Hypervisor and virtual machines, we install only one OS and on top that we create 4 containers. Then applications are running inside each containers.**
- **Containers are much smaller and more efficient than VMs**

Containers vs Virtualization

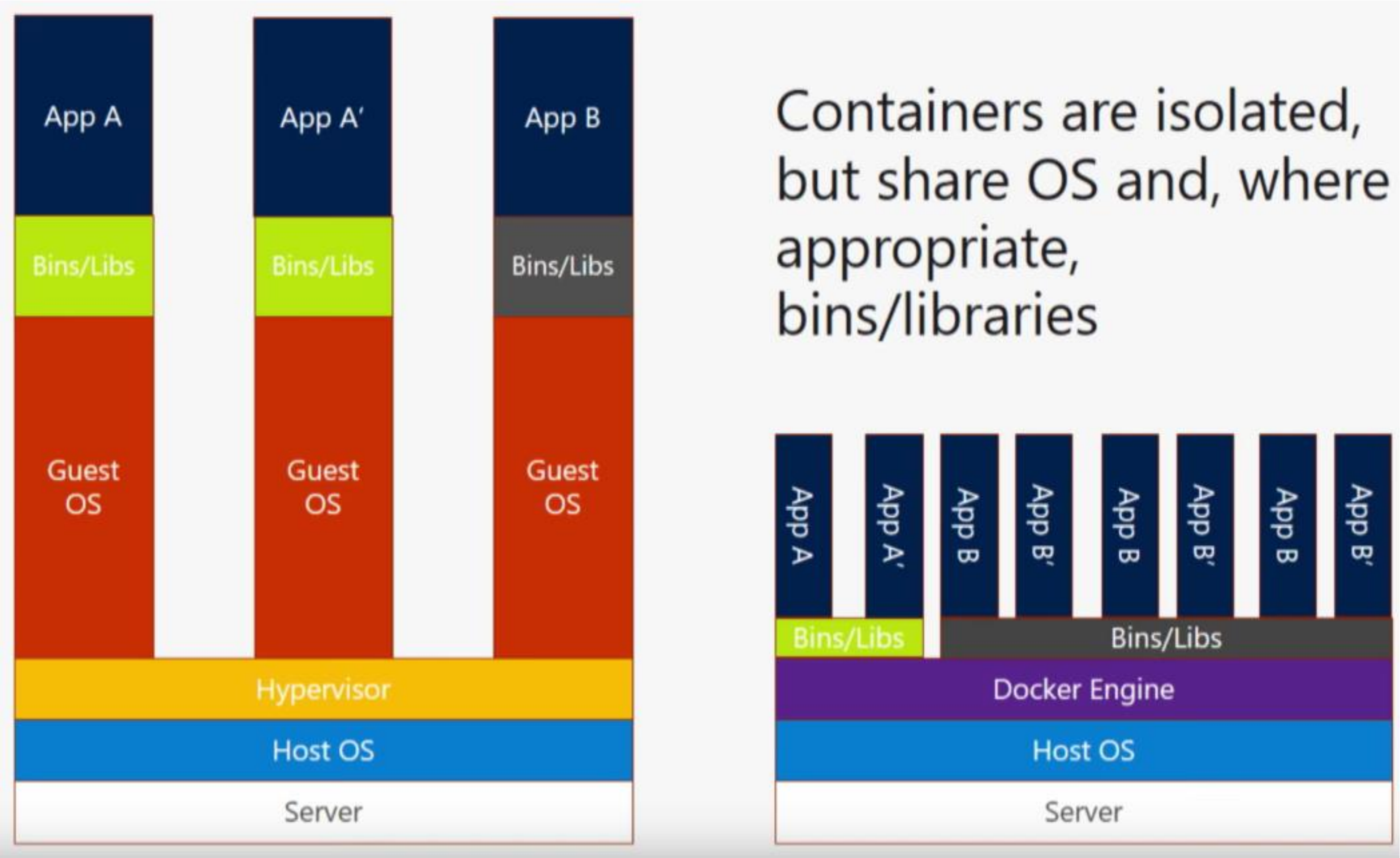


Hypervisor
Architecture



Container
Architecture

Containers vs Virtualization



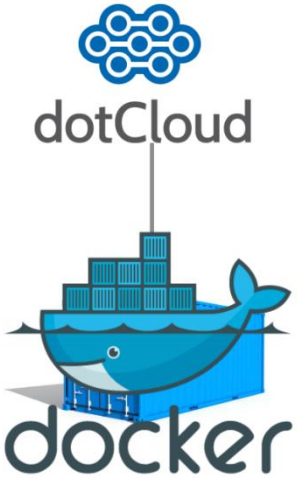
Containers are isolated, but share OS and, where appropriate, bins/libraries

Docker

- an open-source project that automates the deployment of applications inside software containers
- Docker is the containers What Vmwares is to the Hypervisor
- Docker Inc.(the company)
- Docker project: container engine plus Docker tools e.g. Docker client, Docker compose…
- github Repository

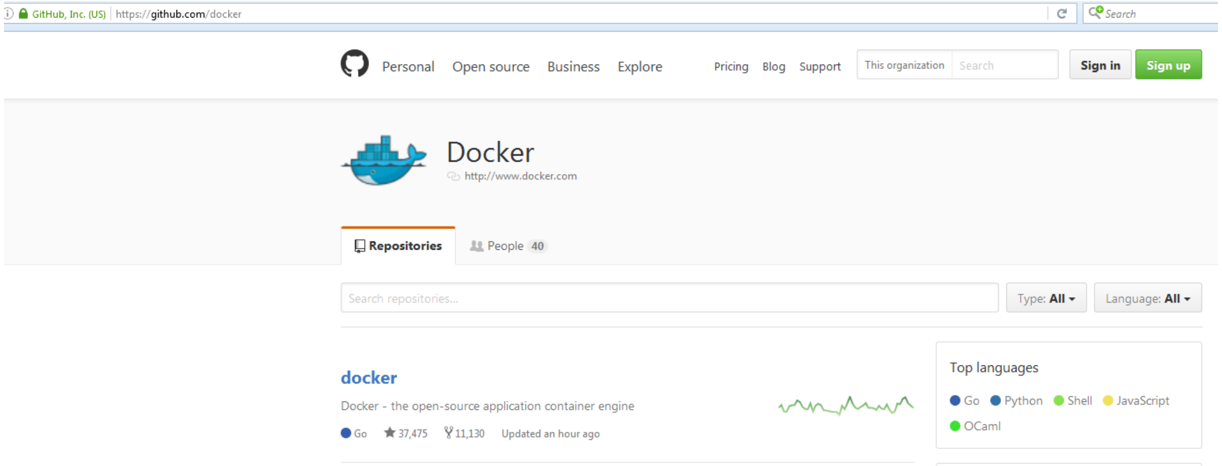
Docker Inc.

- The original author and primary sponsor of the Docker open source project.
- Tech Starter based on SF
- Originally it was dotCloud (platform as a service) and the idea behind that business was to offer the developer platform on top of amazon web service
- Docker Inc. Website
- Docker Inc has over 120 employees



Docker Project

- The Docker project is not absolutely the Docker Inc.
- It belongs to the community.
- Open source under Apache License 2.0
- Core Docker component written in GO language
- github Repository



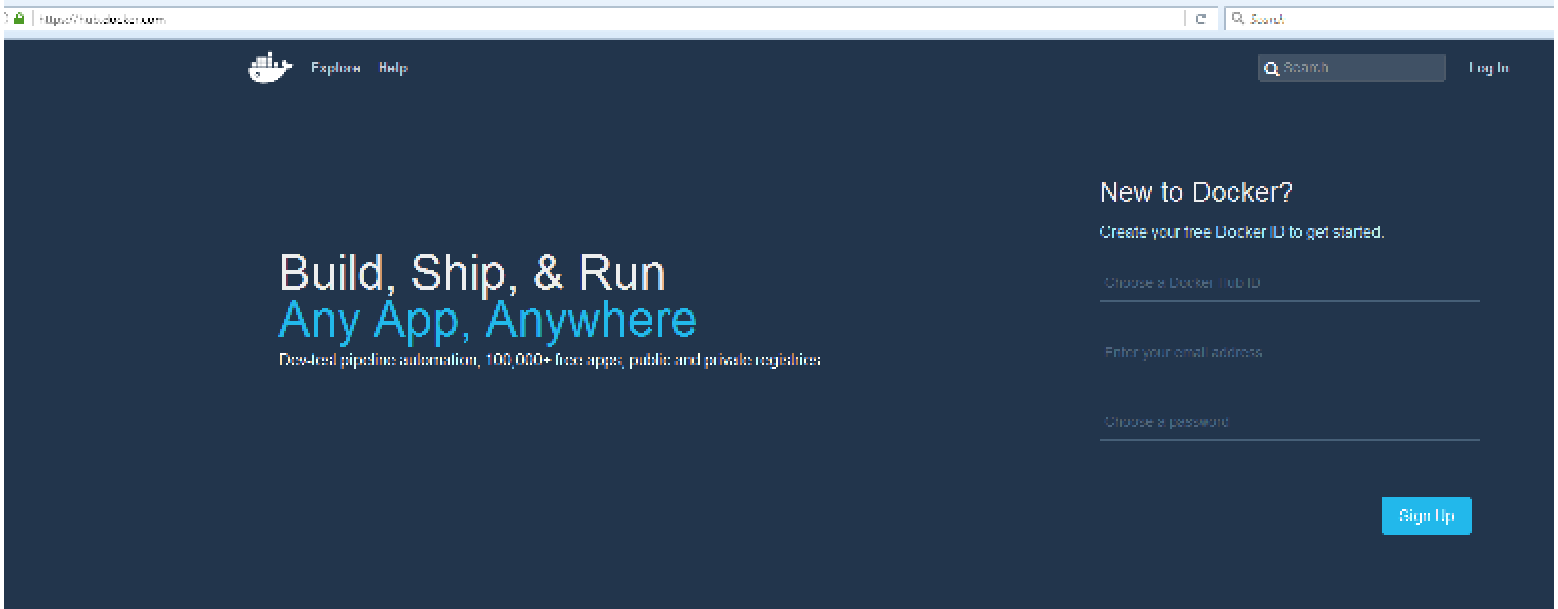
Docker
http://www.docker.com

docker
Docker - the open-source application container engine
● Go ★ 37,475 11,130 Updated an hour ago

Top languages
● Go ● Python ● Shell ● JavaScript ● OCaml

Docker Hub

- a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images.
- The public Docker registry(Hub.Docker.com).
- Searchable.
- Private Registries: can be hosted in Docker, Azure container registry, AWS, Google



Build, Ship, & Run
Any App, Anywhere

Dev-test pipeline automation, 100,000+ free apps, public and private registries

New to Docker?

Create your free Docker ID to get started.

Choose a Docker Hub ID

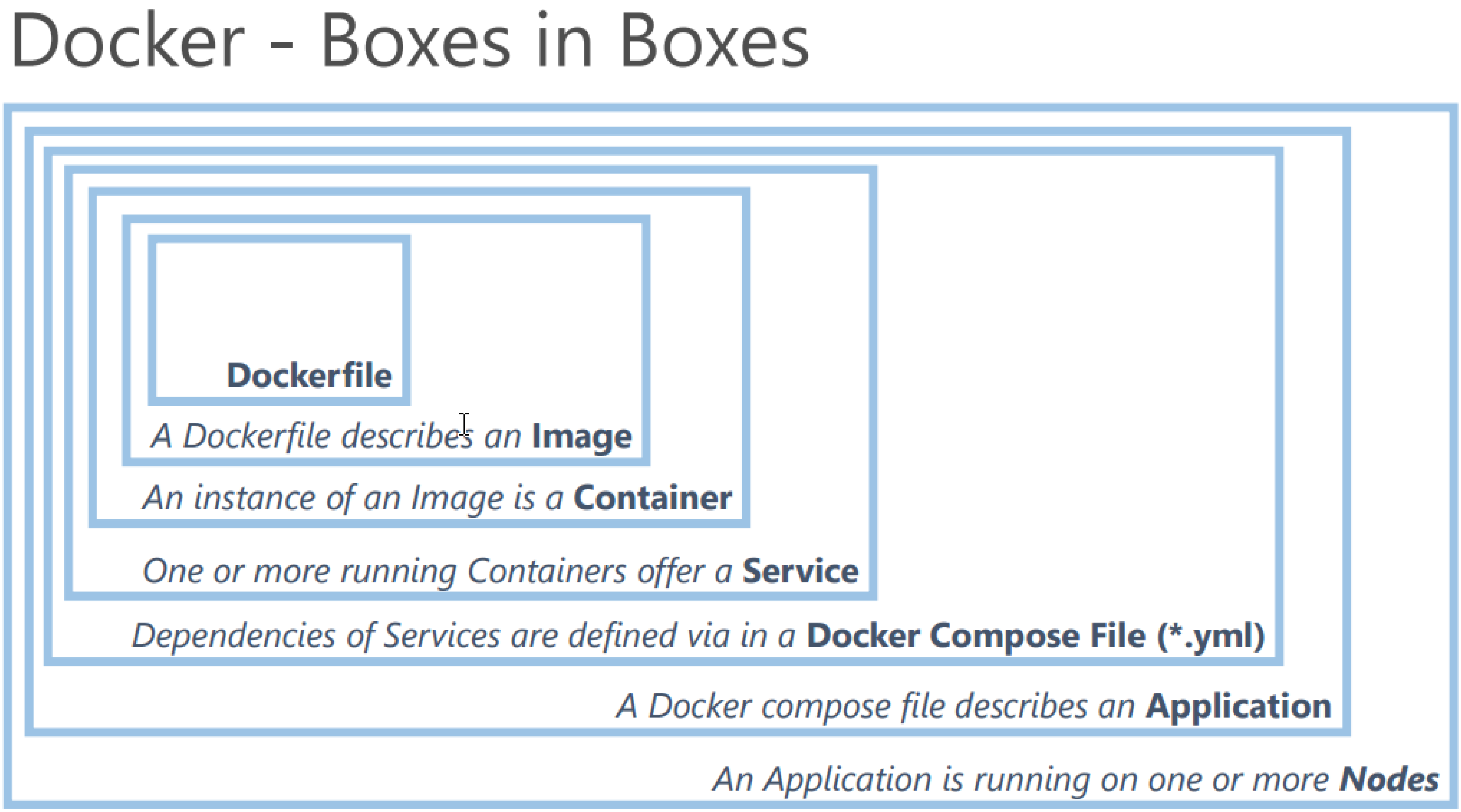
Enter your email address

Choose a password

Docker Terms

- Application: a combination of services
- Service: provided by 1 or more containers
- Container: a running instance of an image
- Image: basis for a container, generated from docker file
- Docker file: describe an image
- Container registry: online portal for images

Docker Terms



Docker - Boxes in Boxes

Dockerfile

A Dockerfile describes an **Image**

An instance of an Image is a **Container**

One or more running Containers offer a **Service**

Dependencies of Services are defined via in a **Docker Compose File (*.yml)**

A Docker compose file describes an **Application**

An Application is running on one or more ***Nodes***

Docker Containers

- Linux Containers
- Windows Containers

Docker Containers

- Linux Containers on Azure on Ubuntu Machine

Windows Containers

-on June 8, 2016, Microsoft announced that Docker could be used natively on Windows 10 with Hyper-V Containers, to build, ship and run containers utilizing the Windows Server 2016 Technical Preview 5 Nano Server container OS image.

-richly integrated with docker and as a result providing seamless application portability and delivery.

-don’t run Linux containers.

-Windows containers help application development by providing two compatible deployment options:

- **Windows Server Containers:**

Windows Server Containers, they use isolation just like a Linux container. It's all isolation based on process isolation. Windows Server Containers maximize density and performance for a variety of development scenarios.

- **Hyper-V Containers:**

Hyper-V containers provide great isolation and control which might be imported in limited trusted environments such as running containerized applications on shared multi-tenant infrastructure. Also sometimes there are rules and regulation that requires strict isolation.

Hyper-V Containers encapsulates each container in a light weight virtual machine. Hyper-V containers are the default on Windows 10. In order to run a Hyper-V container we need to define the isolation as below:

docker run –isolation=hyperv

Windows Containers

	Windows Server Containers	**Hyper-V Containers**
Isolation	Process	VM
Windows Server 2016	Supports	Supports
Windows 10 Pro+	Doesn’t support	Supports
Application	maximize density and performance	Multi-tenant infrastructure and untrusted code. Rules and regulation that requires strict isolation
	docker run	*docker run –isolation=hyperv*

Windows Containers

Docker on windows development ENV

1.Install windows 10 Pro or Enterprise

2.Enale virtualization in bios level

3.Install hyper-v

4.Install docker for windows

Docker Containers

- Windows Containers on Azure

Agenda

- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo

Docker Benefits

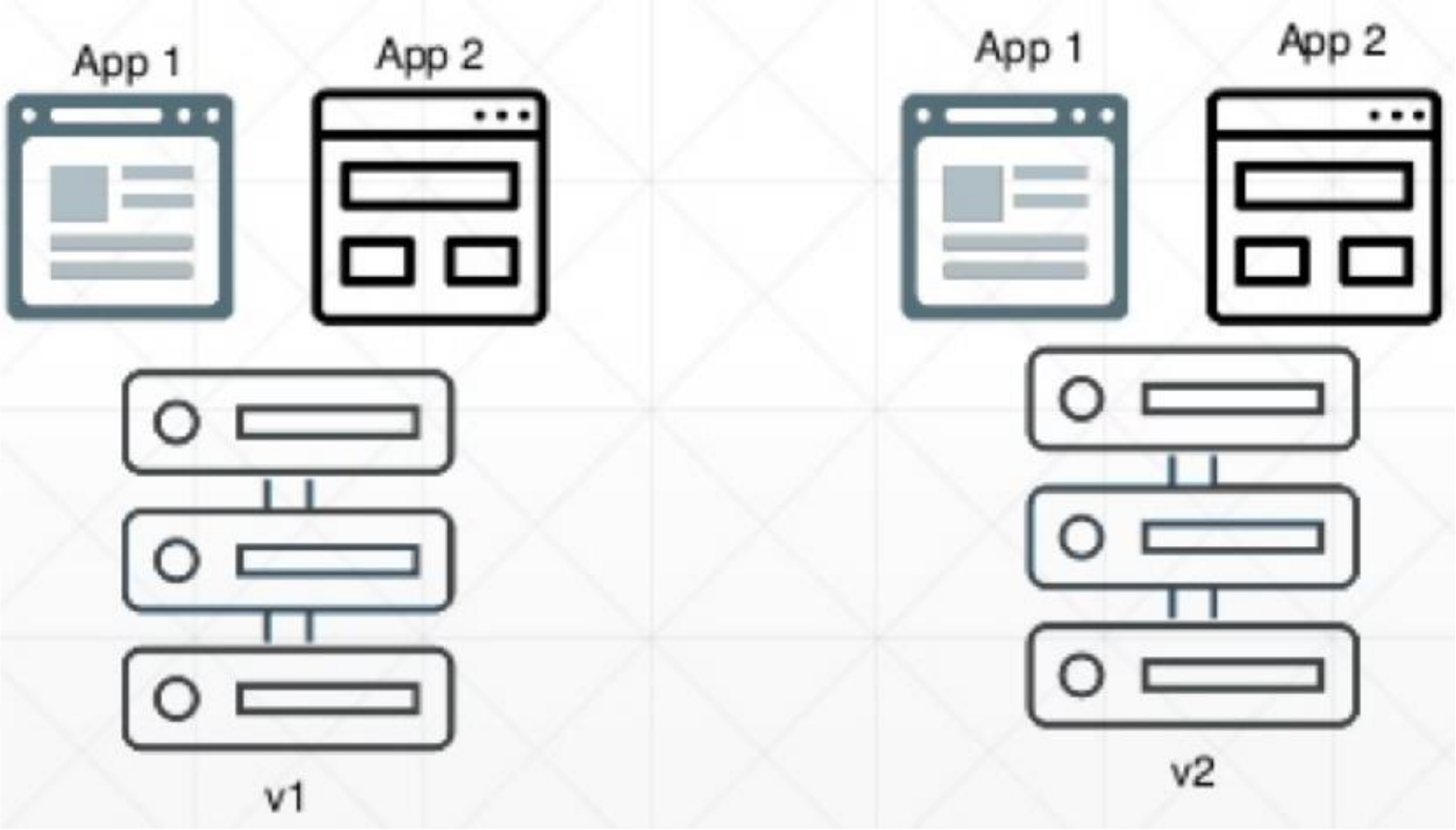
- Set up DEV environment very quickly
- No More App Conflict
- Consistent Environment
- Faster Software Shipment

Docker Benefits

- Set up DEV environment very quickly

Docker Benefits

- No More App Conflict

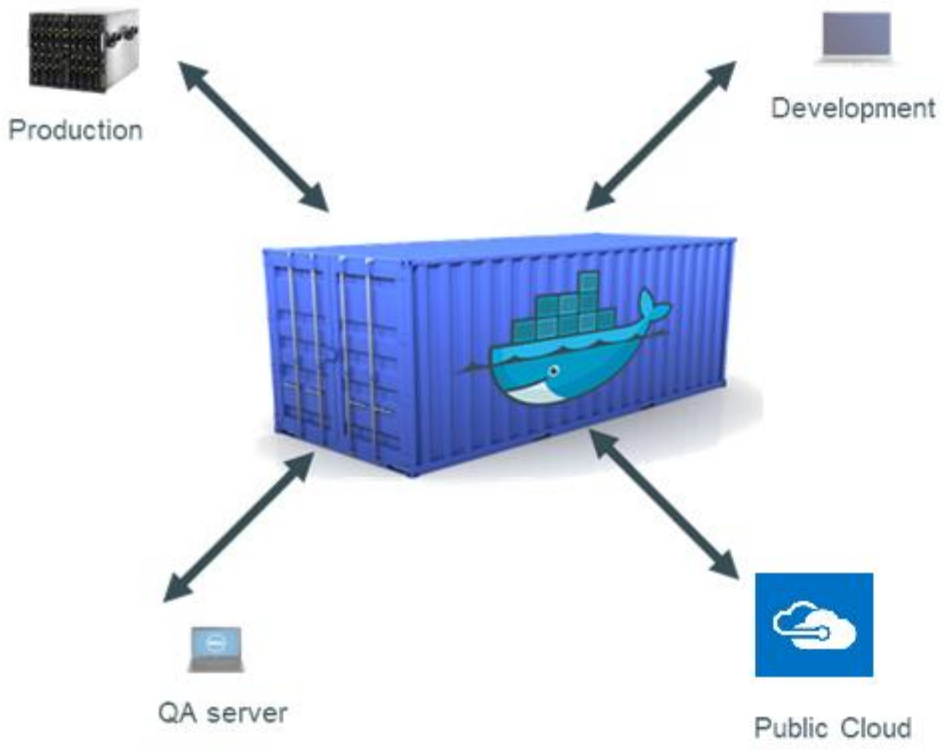


Docker Benefits

- Consistent Environment

Docker Benefits

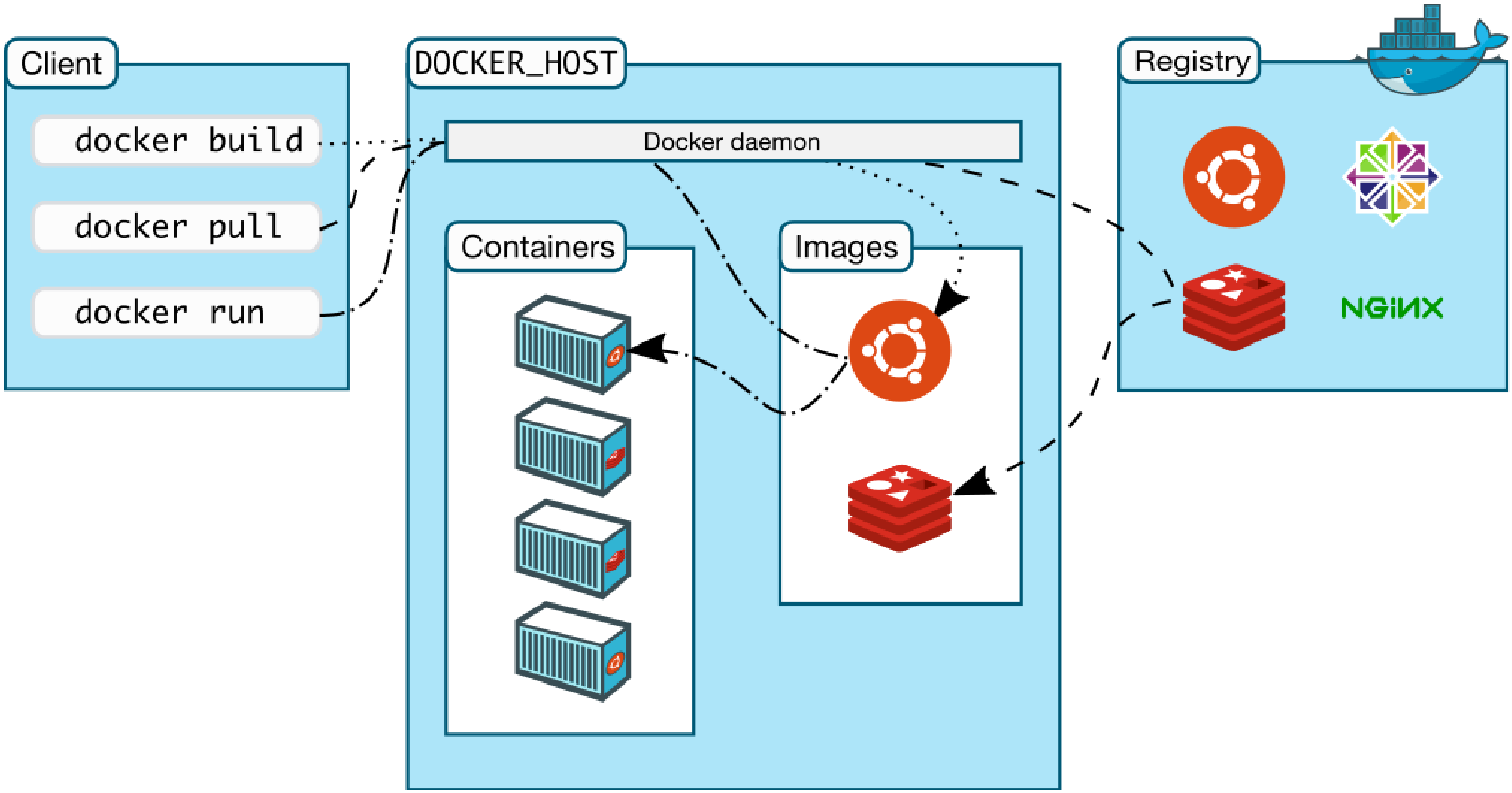
- Faster Software Shipment



Agenda

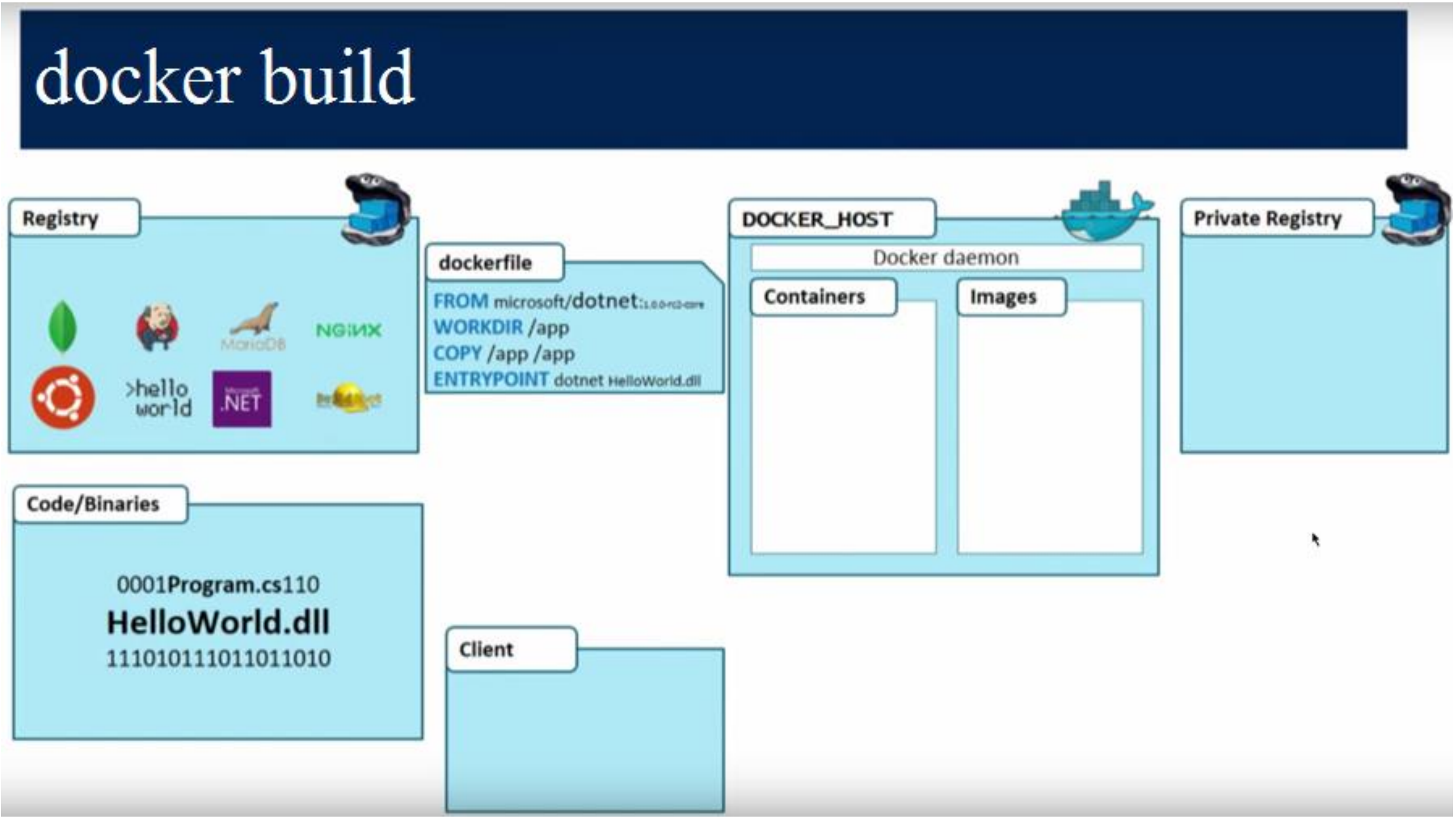
- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo

Docker Architecture



https://docs.docker.com/engine/docker-overview/#docker-architecture

Docker Architecture



docker build

Registry

dockerfile

```
FROM microsoft/dotnet:1.0.0-rc2-core
WORKDIR /app
COPY /app /app
ENTRYPOINT dotnet HelloWorld.dll
```

DOCKER_HOST

Docker daemon

Containers

Images

Private Registry

Code/Binaries

0001Program.cs110

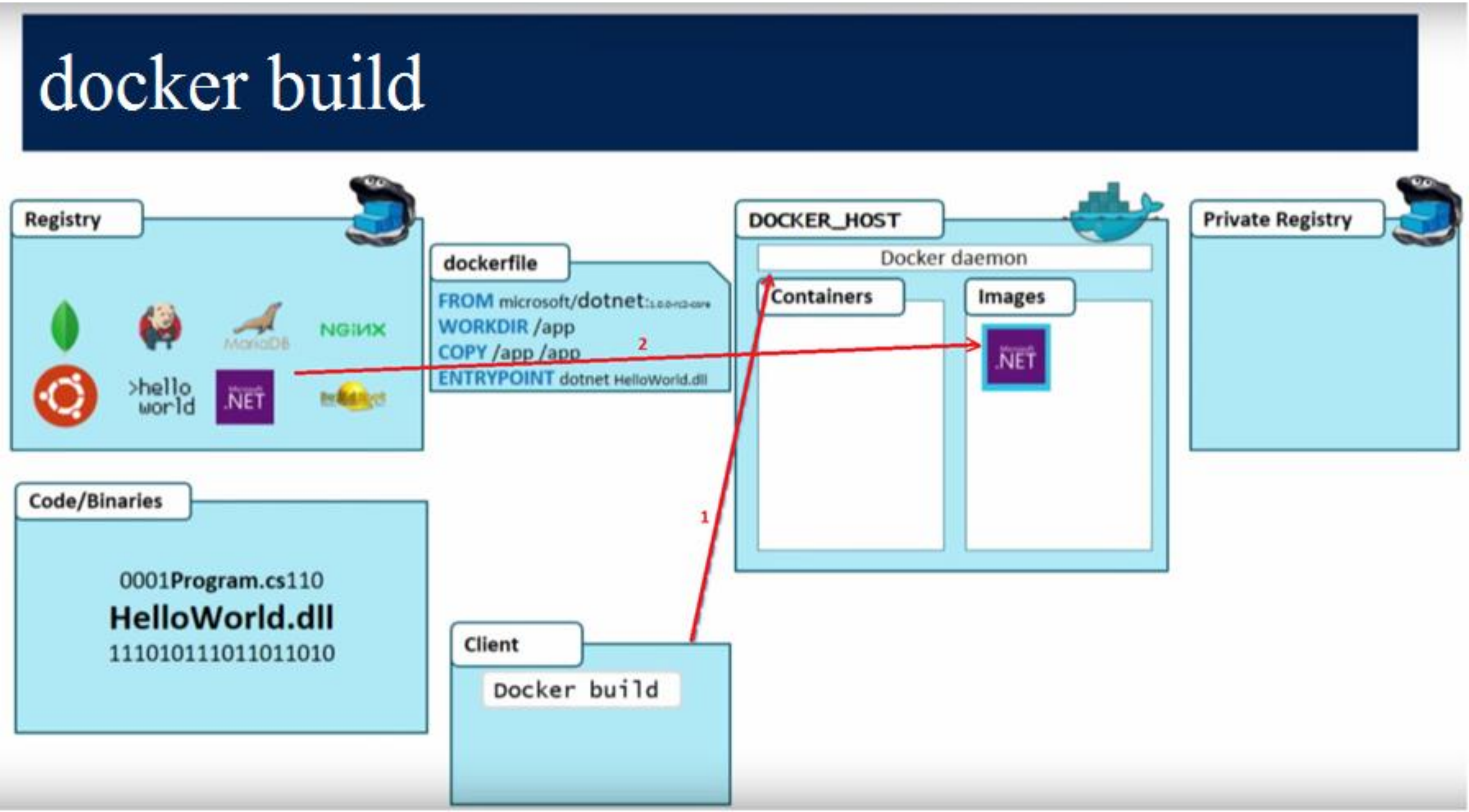
HelloWorld.dll

111010111011011010

Client

https://docs.docker.com/engine/docker-overview/#docker-architecture

Docker Architecture



docker build

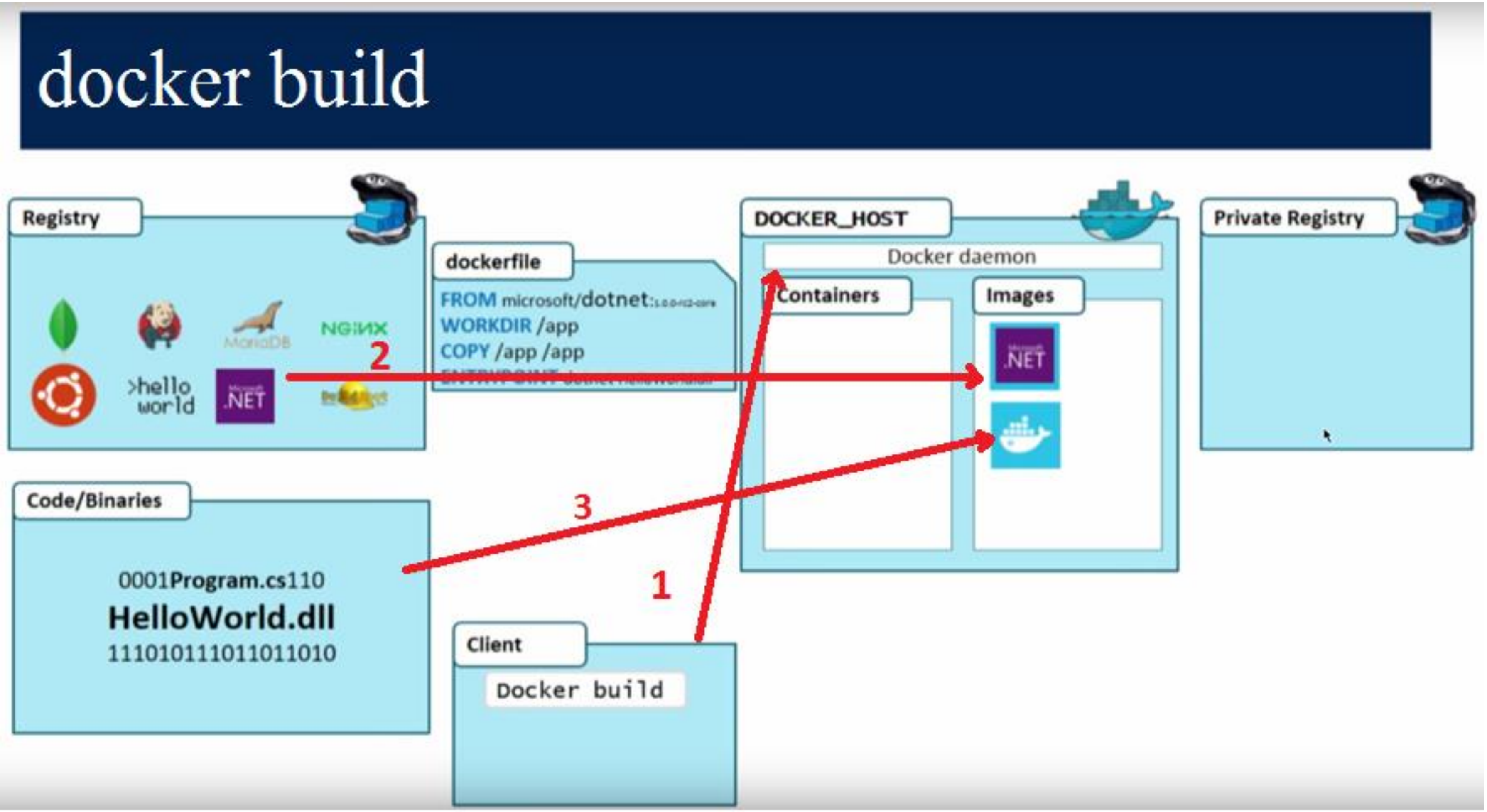
dockerfile

```
FROM microsoft/dotnet:1.0.0-rc2-core
WORKDIR /app
COPY /app /app
ENTRYPOINT dotnet HelloWorld.dll
```

```
Docker build
```

https://docs.docker.com/engine/docker-overview/#docker-architecture

Docker Architecture



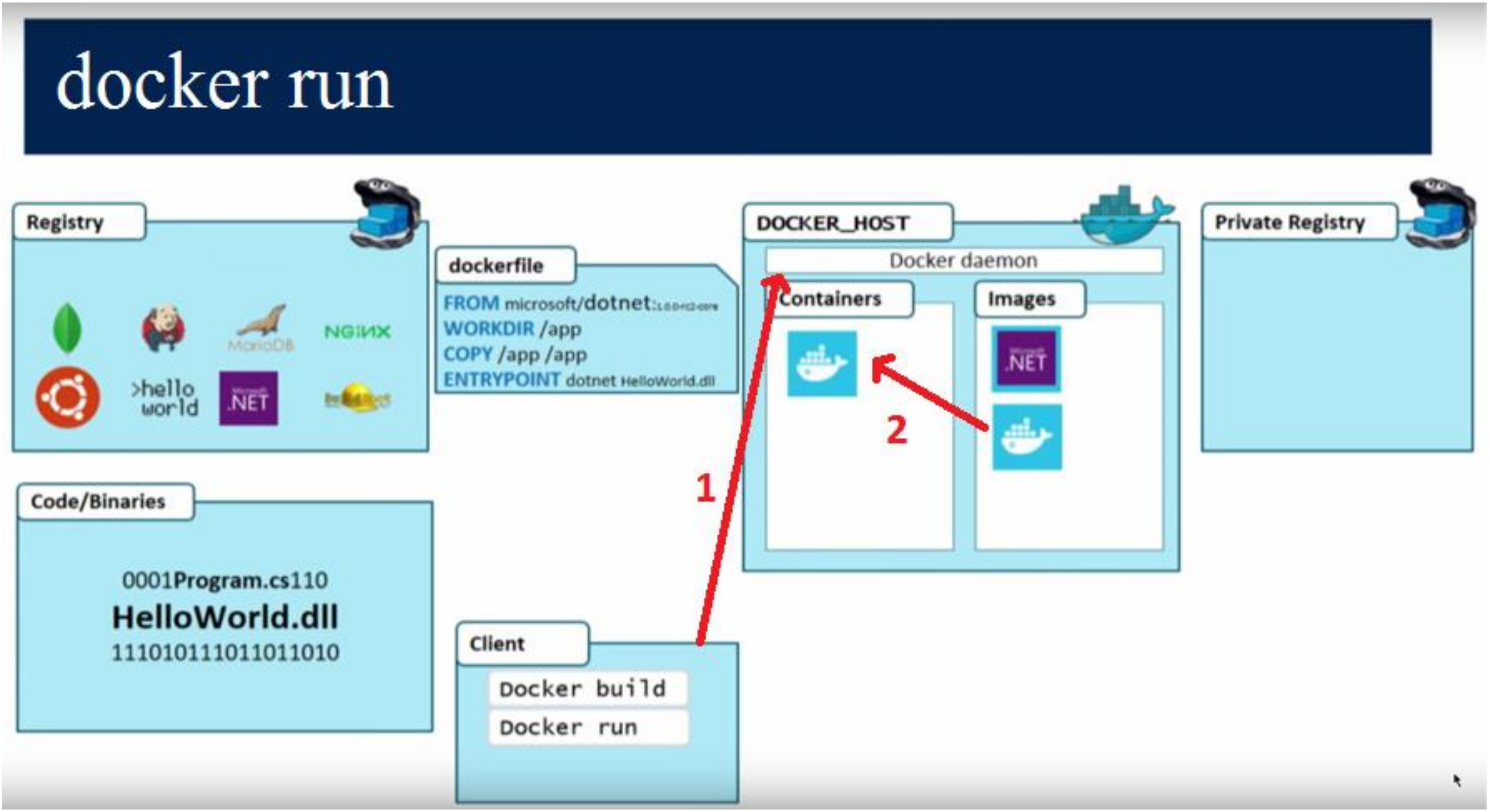
docker build

```
FROM microsoft/dotnet:1.0.0-rc2-core
WORKDIR /app
COPY /app /app
ENTRYPOINT dotnet HelloWorld.dll
```

```
Docker build
```

https://docs.docker.com/engine/docker-overview/#docker-architecture

Docker Architecture



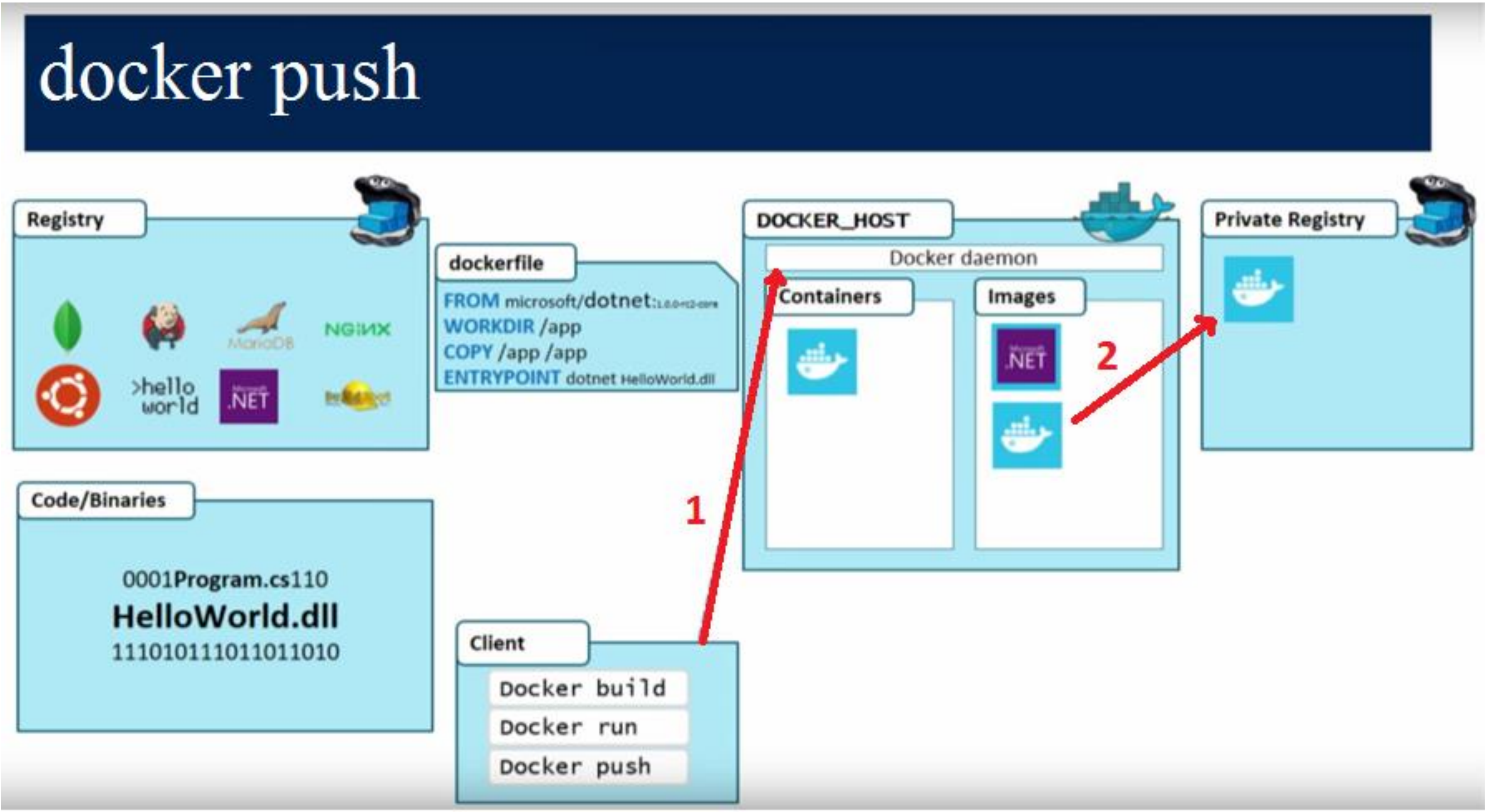
docker run

```
FROM microsoft/dotnet:1.0.0-rc2-core
WORKDIR /app
COPY /app /app
ENTRYPOINT dotnet HelloWorld.dll
```

```
Docker build
Docker run
```

https://docs.docker.com/engine/docker-overview/#docker-architecture

Docker Architecture



docker push

dockerfile

```
FROM microsoft/dotnet:1.0.0-rc2-core
WORKDIR /app
COPY /app /app
ENTRYPOINT dotnet HelloWorld.dll
```

Code/Binaries

0001Program.cs110

HelloWorld.dll

1110101110110111010

Client

```
Docker build
Docker run
Docker push
```

https://docs.docker.com/engine/docker-overview/#docker-architecture

Agenda

- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo

Docker compose

Real life application

- One application consists of multiple containers Web Server, Database, Cash Server etc.
- Mutual dependency/ startup order
- One container depends on another container e.g. Database up first

Docker compose

Multi-container apps are a hassle.

- Build images from Dockerfiles
- Pull images from the Hub or a private registry
- Configure and create containers
- Start and stop containers
- Stream their logs

Docker compose

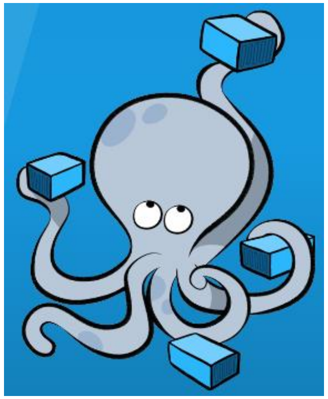
- One of the more exciting pieces of docker
- great way to automatically manage the lifecycle of your app
- get it up and running, stop it very quickly
- its logo gives away a lot about what it does [multiple images and manage]
- doing it by hand with lots of containers could be problematic and not productive

-start, stop, rebuild services

-view status of running services

-stream the log output of running services

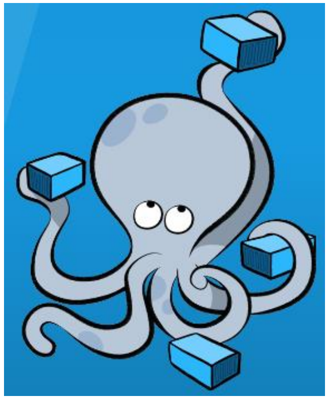
-run a one-off command on a service



Docker compose

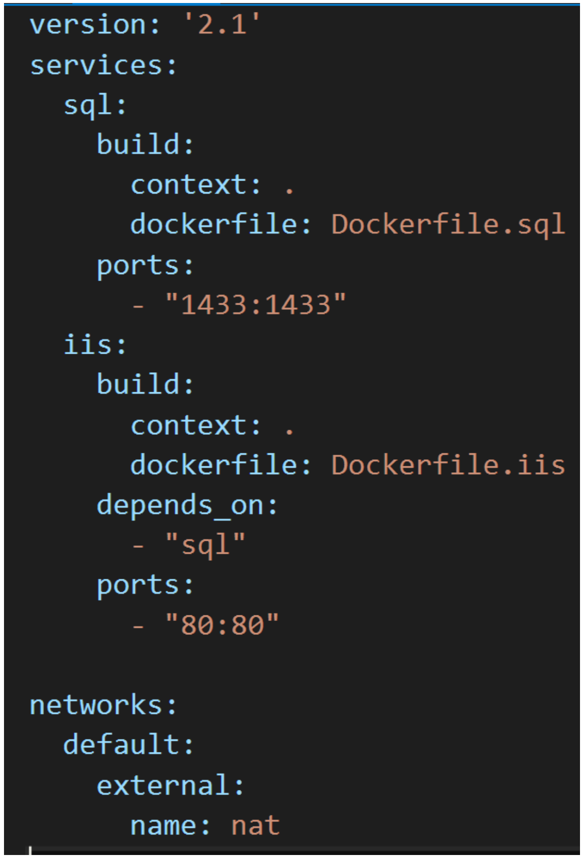
Docker Compose:

- tool for defining and running multi-container Docker applications.
- Compose file to configure your application’s services. Then, using a single command, you create and start all the services from your configuration.
- API above docker run
- Integrated with Docker Swarm



Docker compose

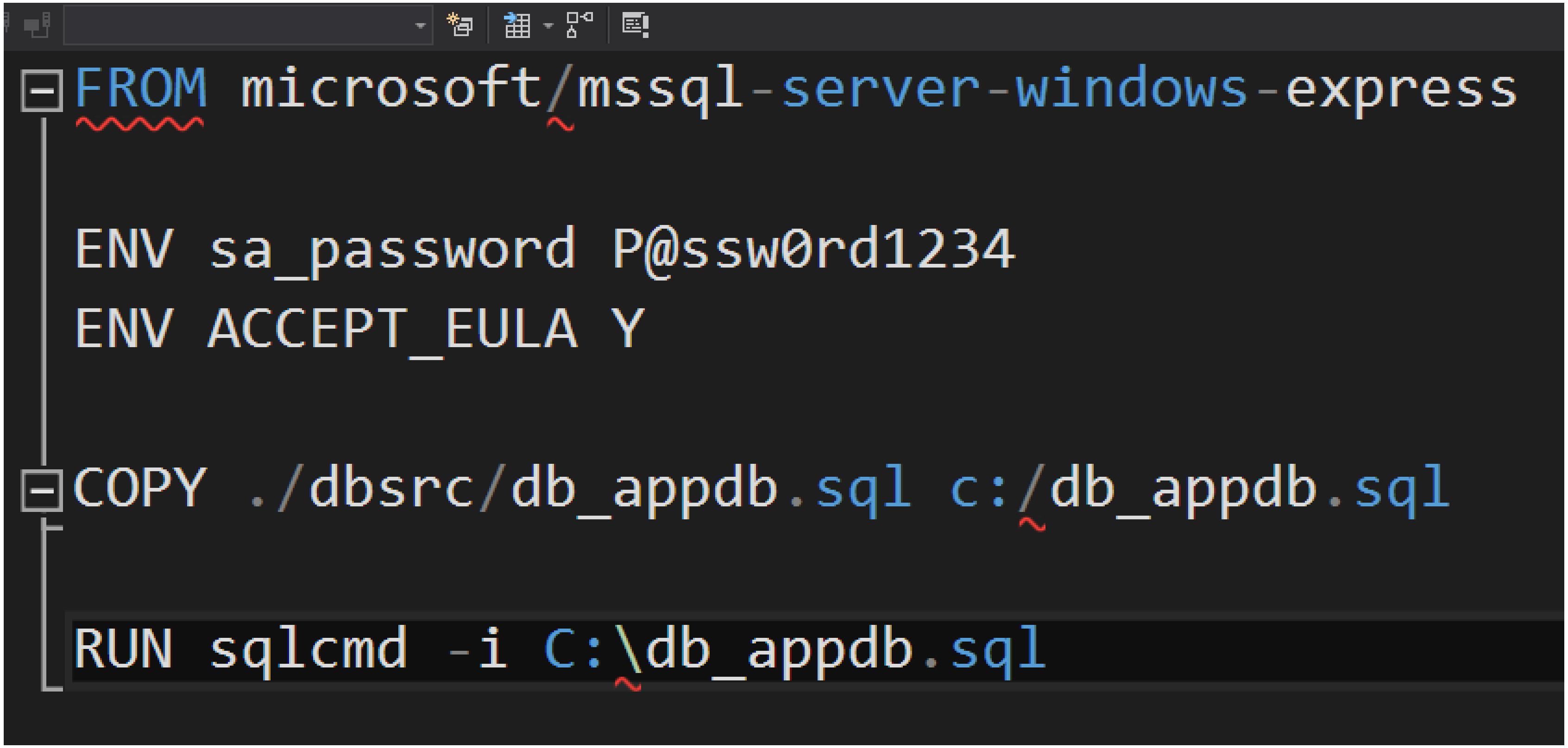
- Docker-compose.yml



```
version: '2.1'
services:
  sql:
    build:
      context: .
      dockerfile: Dockerfile.sql
    ports:
      - "1433:1433"
  iis:
    build:
      context: .
      dockerfile: Dockerfile.iis
    depends_on:
      - "sql"
    ports:
      - "80:80"

networks:
  default:
    external:
      name: nat
```

Docker compose



```
FROM microsoft/mssql-server-windows-express

ENV sa_password P@ssw0rd1234
ENV ACCEPT_EULA Y

COPY ./dbsrc/db_appdb.sql c:/db_appdb.sql

RUN sqlcmd -i C:\db_appdb.sql
```

Docker compose

Sample Application

Agenda

- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo

Azure Container Registry

- What is Container Registry? A place to push to/pull from docker images

- Docker Hub is the official registry http://hub.docker.com/

- Within each registry, we have our repositories

- Who has access to my images? It depends ☺

Docker Hub

Public Repos => everyone can Pull, but only me Push

Private Repos =>Only me Pull and Push

- Absolutely essential for Automated Workflow and CI

Azure Container Registry

- Cloud Solutions:

Docker Hub, 3rd Parties Registries e.g. Azure Container Registry, AWS EC2 Container Registry, Quay.io

- On Premise Solutions:
  - Docker Trusted Registry DTR
  - QUAY Enterprise

Azure Container Registry

- Why Azure Container Registry
  - **To have a Private Image Repository:** Docker hub in the default registry for Docker and allows to push your images to a public registry. But in your company you may want to keep your images within your datacenter as they are private to your organization. You can also use the private registry in Docker hub as well, but we will see what are other benefits using Azure container Registry.

  - **Integration with Azure Active Directory:** You might want to use your active directory groups to manage who has access, so that your development team has push and pull rights, but other teams that use your images have just pull rights.

  - **Network-Close**: You may want to have them network close so that when you are doing pushes and pulls, the physical network latency is just going to be shorter because it is right there in your datacenter. You are not going to have charges e.g. ingress and egress because they are within the data center, and their latency is shorter.

  - **Command Line Interface:** You have a familiar experience with the CLIs. Azure team stayed true to the Docker CLIs for different docker commands e.g. docker login, docker push, docker pull, so that you're going to have a common experience that you're used to.

-

Azure Container Registry

- Create Azure Container Registry On Azure

Agenda

- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo

Azure Container Service

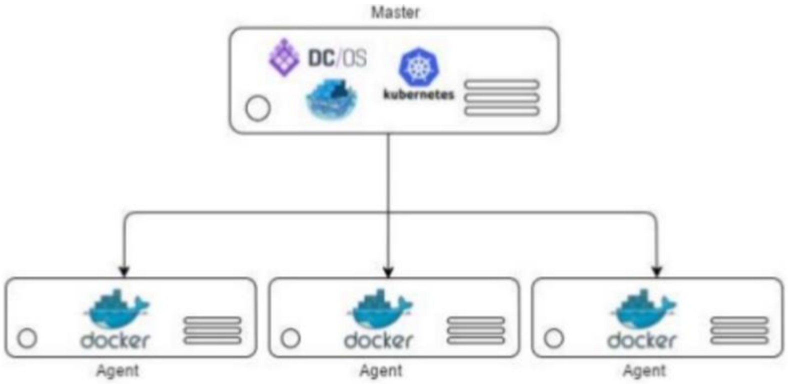
- Containers is good on development machines, how about production?
- Azure Container Service is all about to provide a container hosting environment
- Provides robust, ready-to-use Docker Hosting environment
- Uses open-source orchestration tools(Swarm, DC/OS, Kubernetes)

Container Orchestration

- Facilitates deployment and management of containers
- Containers by design are intended to be deployed in large volumes with some applications using dozens of containers or more
- With this type of scale, automating container deployment and management with orchestration software becomes necessary
- Azure container service supports Swarm, DC/OS, and Kubernetes

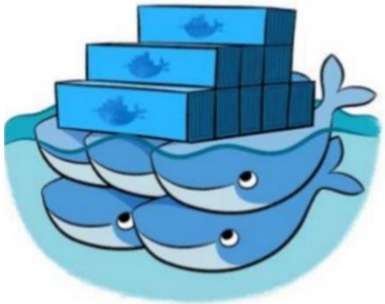
Clusters

- Facilitate load balancing, scalability and HA
- A cluster composed of master nodes which controls the orchestration, and agent nodes that hots the containers



Docker Swarm

- Docker’s orchestration engine
- Current releases of the Docker engine have “Swarm Mode” built in
- Lacks a GUI, but makes up for it with tight integration with Docker
- Natively supported by Azure Container Service



DC/OS

- Data Center Operating System built on Apache Mesos
- Creates a logical data centers and abstracts underlying hardware
- Provides resources traditionally provided by infrastructure, including networking, DNS, and load balancing
- Natively supported by Azure Container Service
- Use DC/OS CLI

Kubernetes

- Open-source orchestration engine from Google
- Provides a robust framework for container orchestration, yet remain lightweight and scalable
- Supported by Azure Container Service

Agenda

- Docker Containers
- Docker Benefits
- Docker Architecture
- Docker compose
- Azure Container Registry
- Azure Container Service
- Demo