Introduction to Docker and Azure

Agenda

- Introduction to Docker
- Building and Running Containers
- Containers and Azure

Running Code

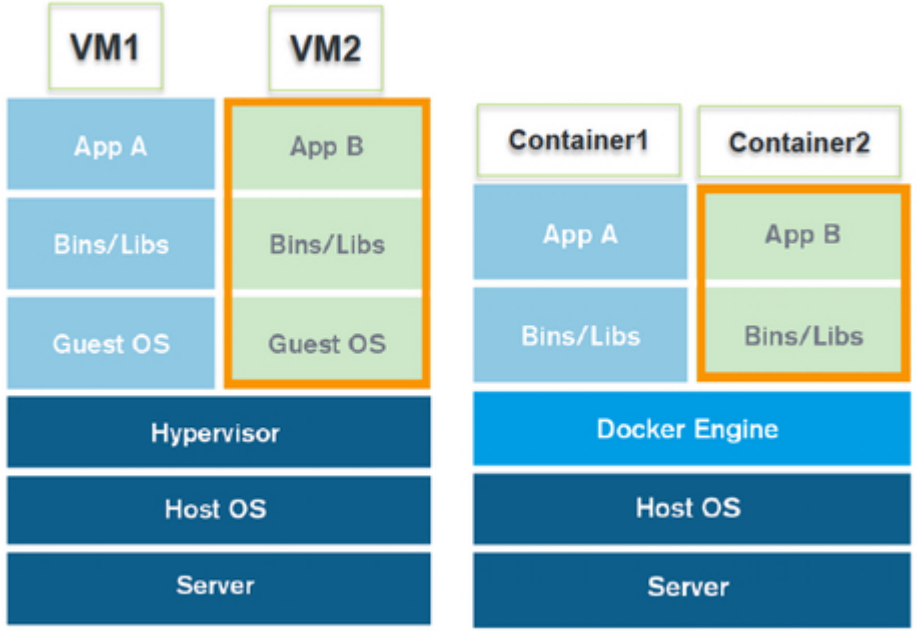
- Physical
- Virtualization
- Containers

Containers

“**Containerization** is an approach to **software** [deployment] in which an application and its versioned set of **dependencies** plus its environment configuration … are **packaged** altogether (the container **image**), tested as a unit and finally **deployed** (the **container** or image instance) to the **host OS**”

Source: Containerized Docker Application Lifecycle with Microsoft Platform and Tools

Containers vs Virtualization

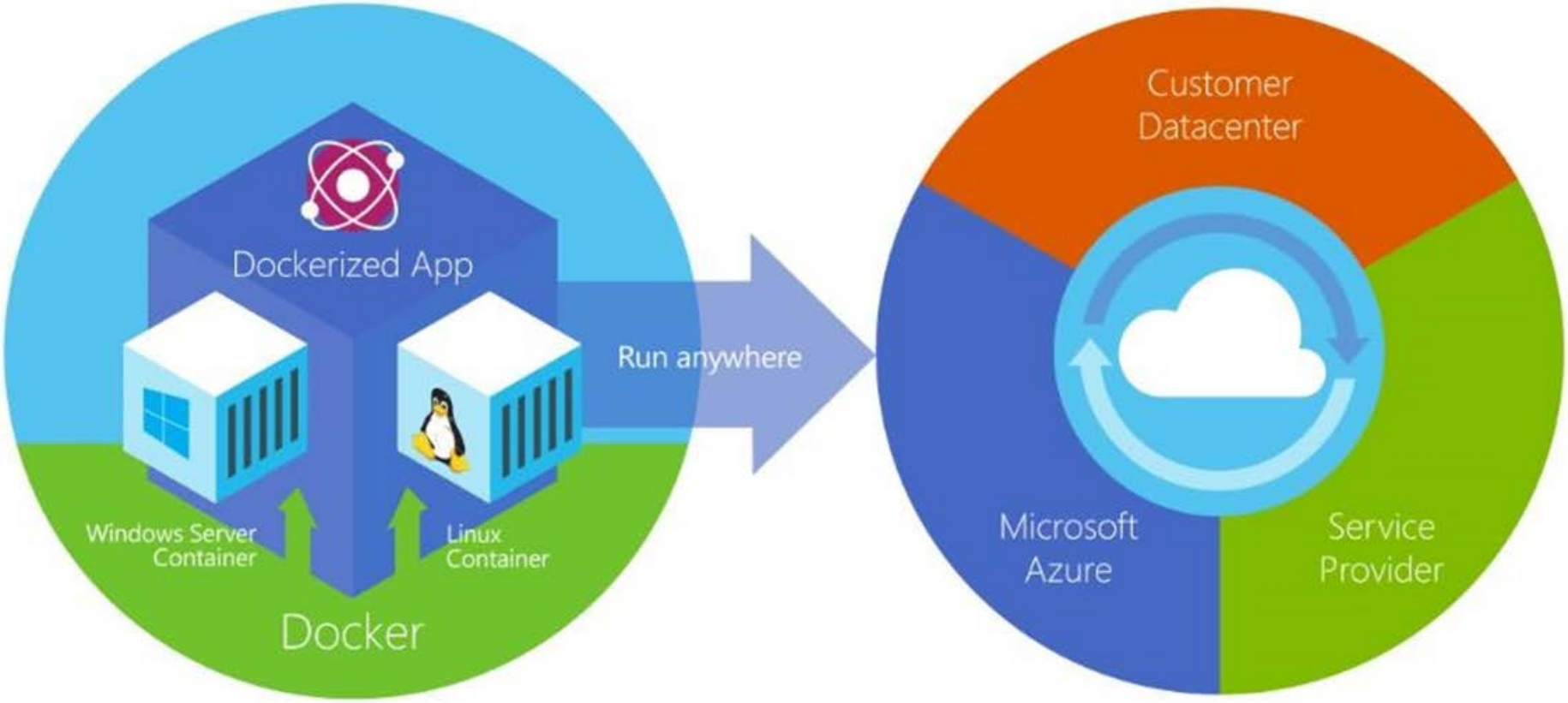


Containers	Virtualization
Lightweight	Heavyweight
Single shared O/S	Multiple O/S instances
Unified environment specification	Separation of VM and application specification
Rapid deployment	Full O/S build and configuration
High density	Low density
Portable	Limited Portability

Why Docker?

- Cross platform container packaging and runtime allowing applications to share a common operating system kernel
- Containers are NOT o/s independent but container specifications and tooling are agnostic
- Modern DevOps practices
  - Application and runtime artifacts defined in code

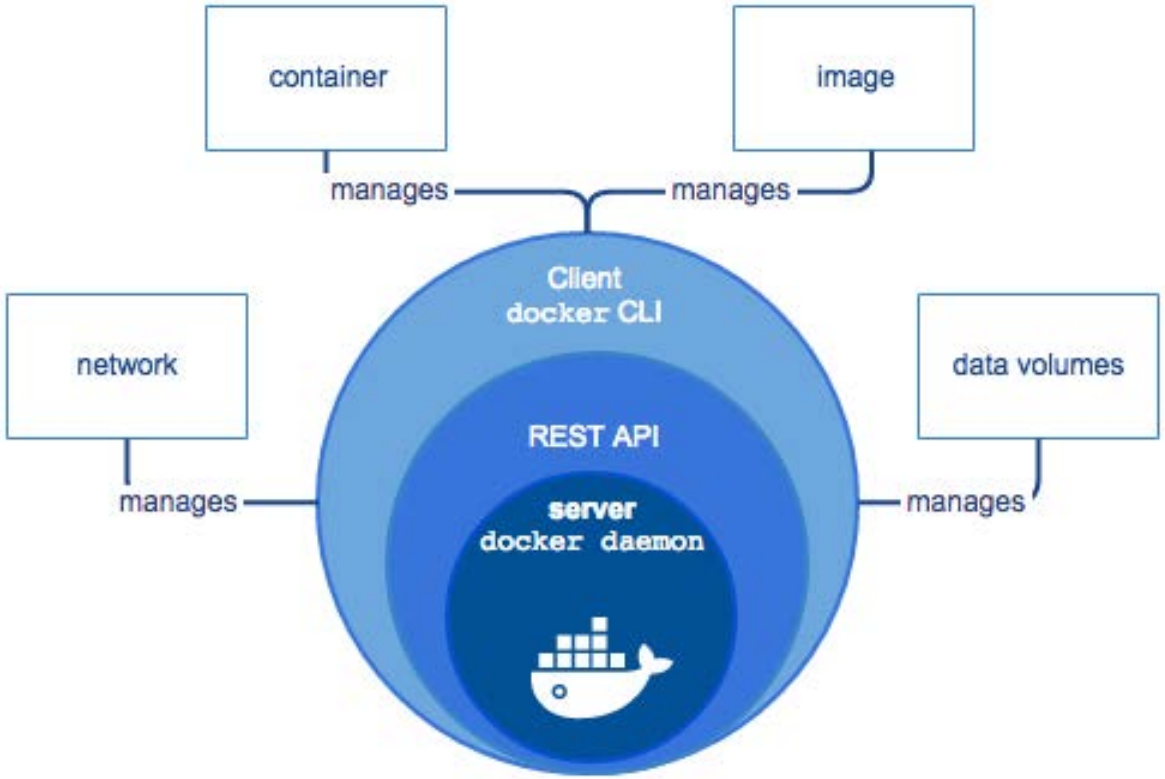
Docker and the Hybrid Cloud



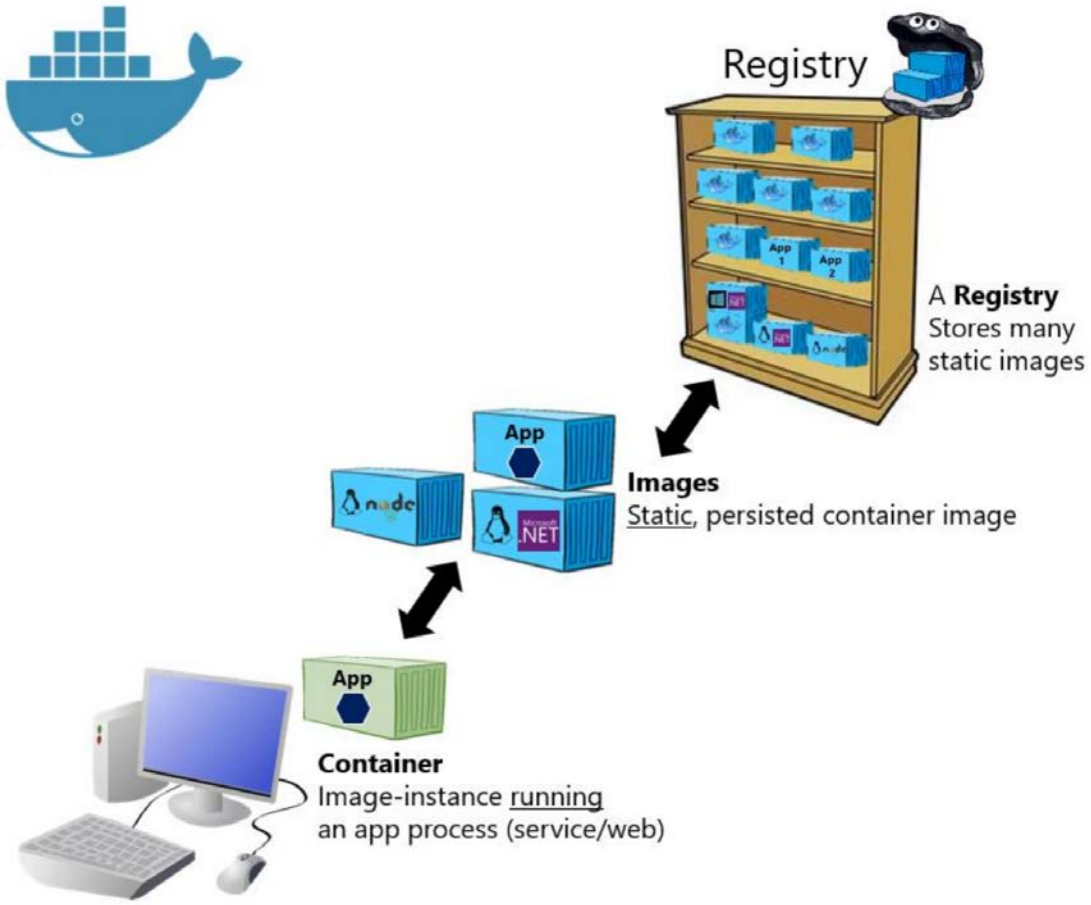
Docker Concepts

- Docker Client
- Docker Engine
- Images
- Docker File
- Docker Registry

Docker Client and Engine

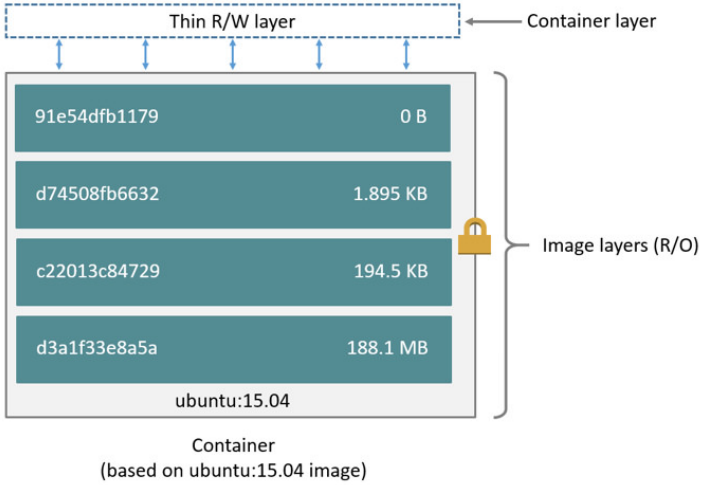


Docker Registry

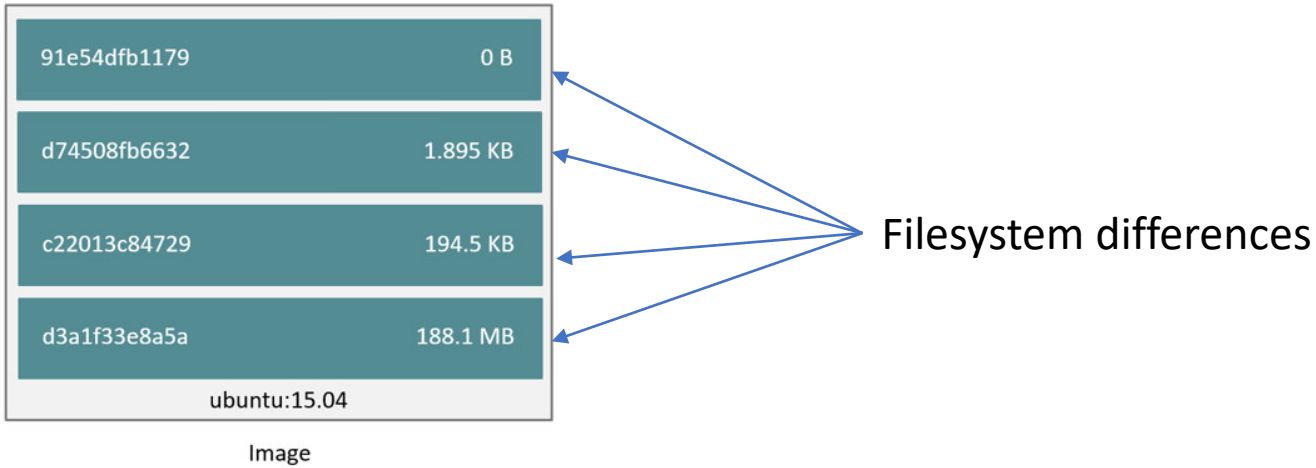


Containers and UnionFS

Containers

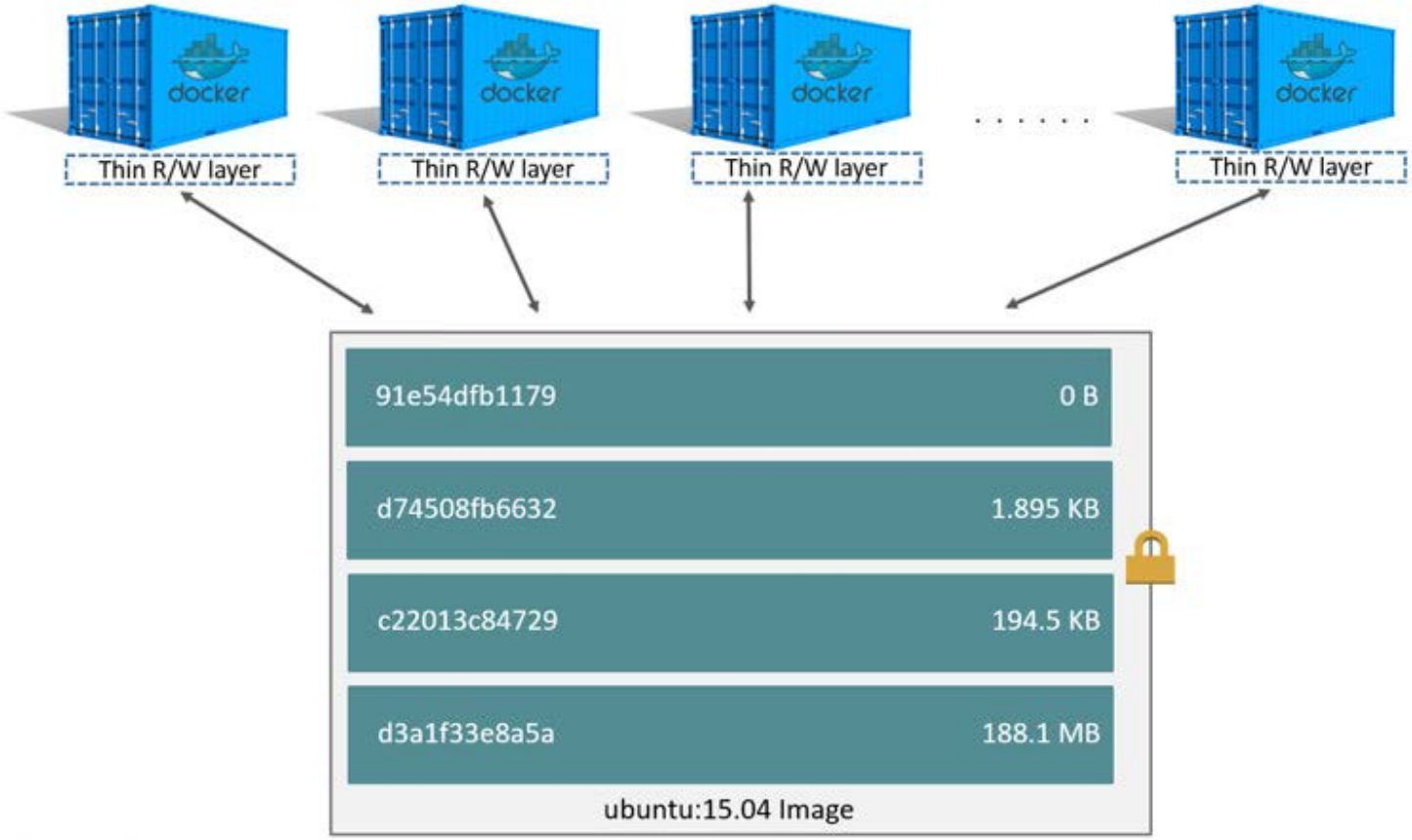


Images

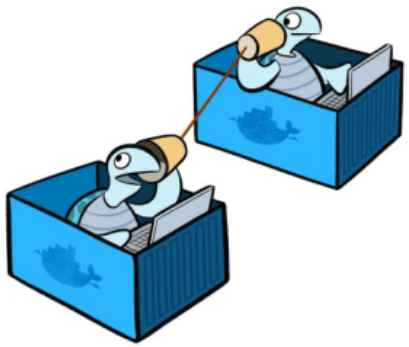


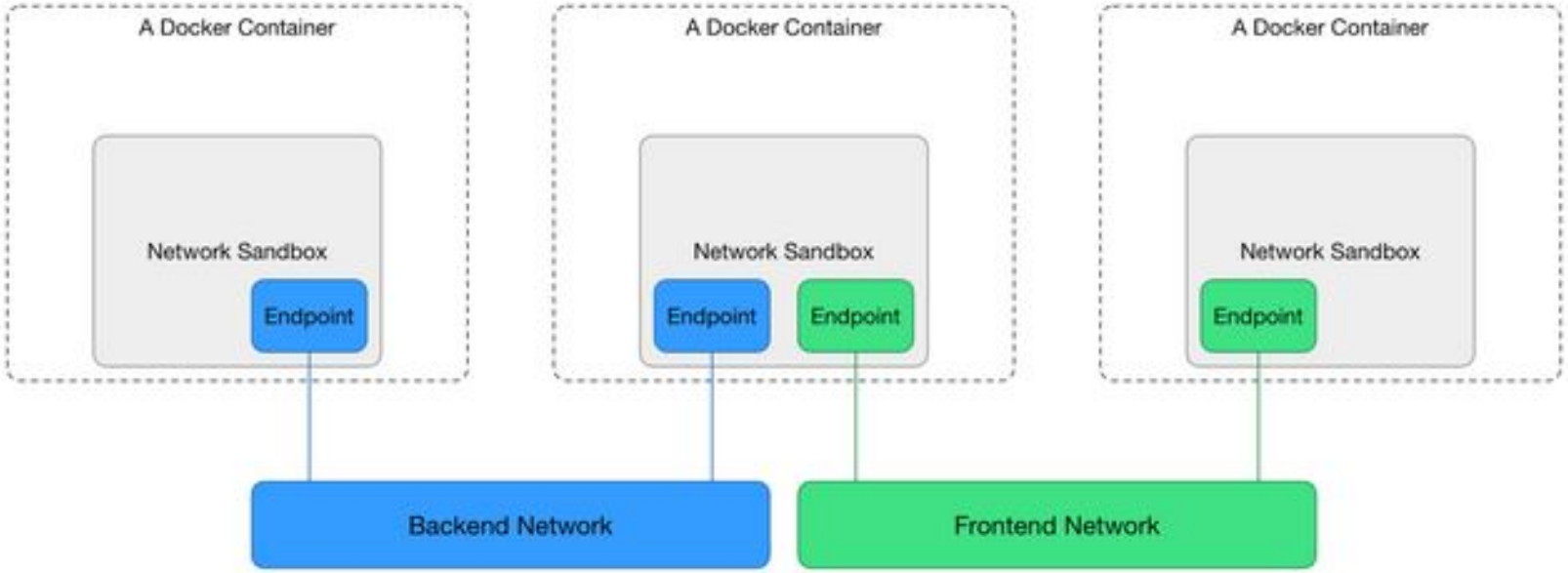
Layers

Running Containers

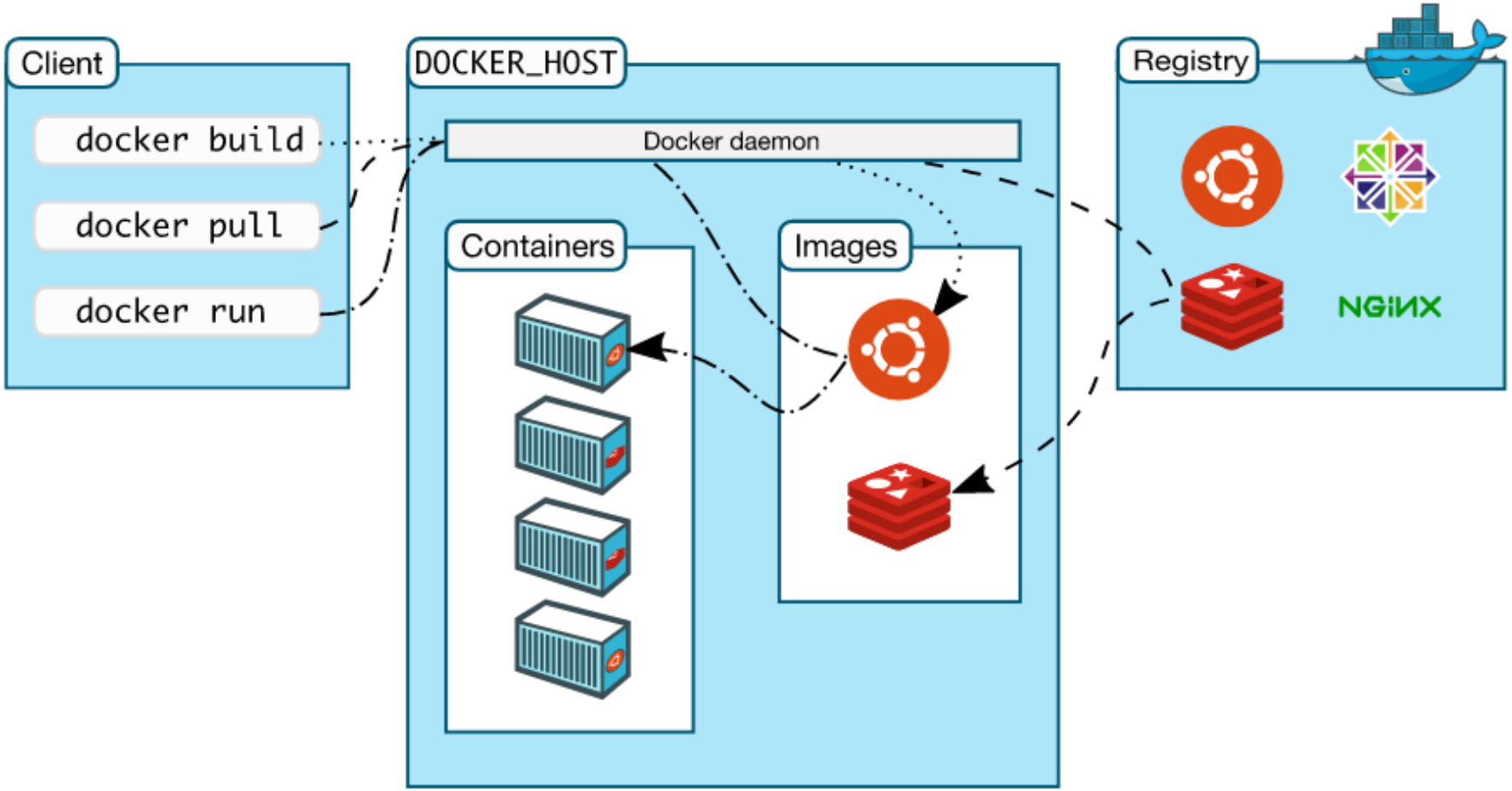


Networking: Container Network Model





Docker Workflow



DockerFile

danawoodman / docker-node-hello-world

Watch 1 | Star 58 | Fork 27

Code | Issues 0 | Pull requests 0 | Projects 0 | Pulse | Graphs

Branch: master | docker-node-hello-world / Dockerfile | Find file | Copy path

danawoodman Initial commit — 647075b on Oct 27 2015

1 contributor

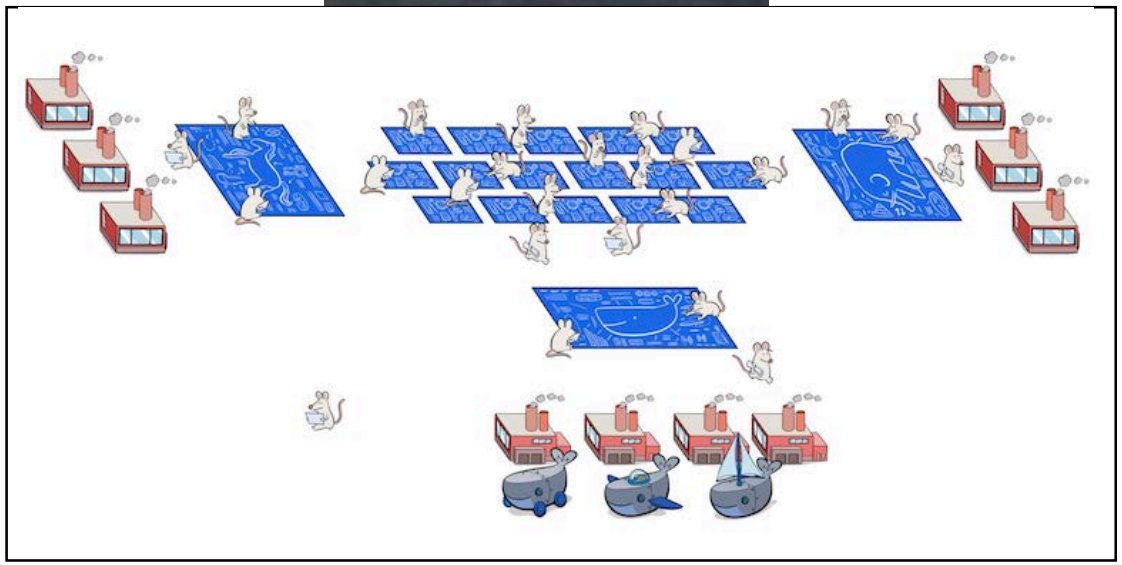
6 lines (5 sloc) | 96 Bytes — Raw | Blame | History

```
FROM node:4.2
COPY . /src
RUN cd /src && npm install
EXPOSE 4000
CMD ["node", "/src/server.js"]
```

docker build –t rivaaj/nodehello .

Docker Inc

moby project



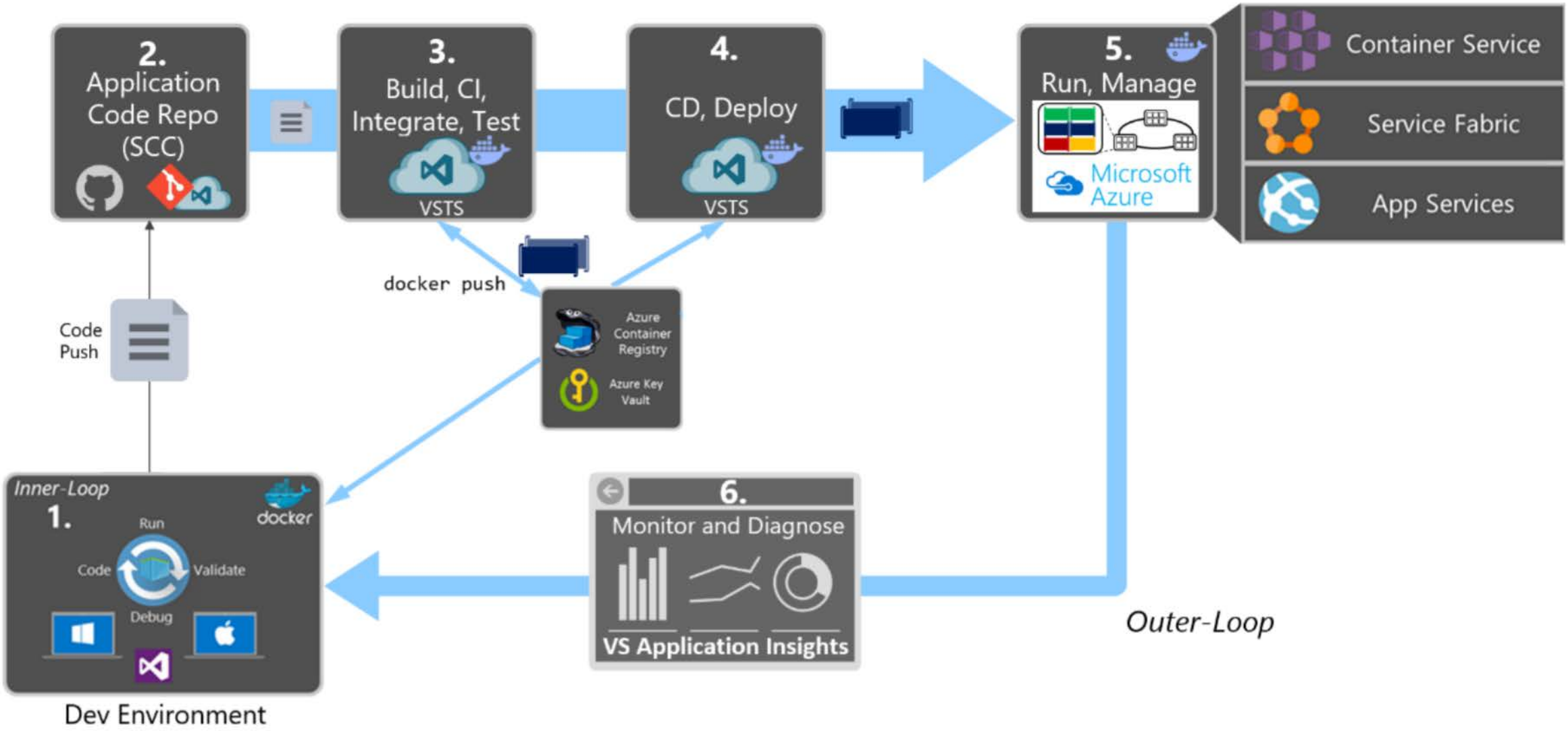
Docker CE

Docker EE

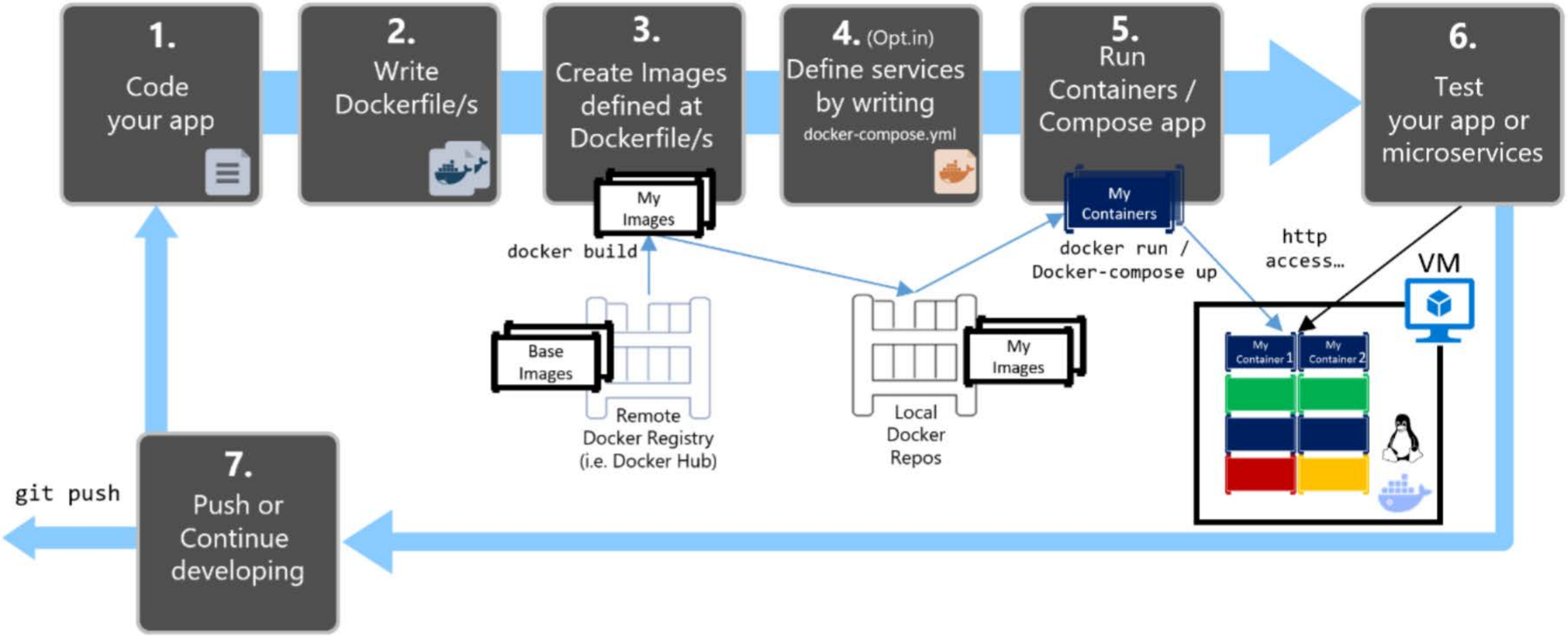
Docker in Azure

- Workflow
- Deployment

Docker End to End Workflow



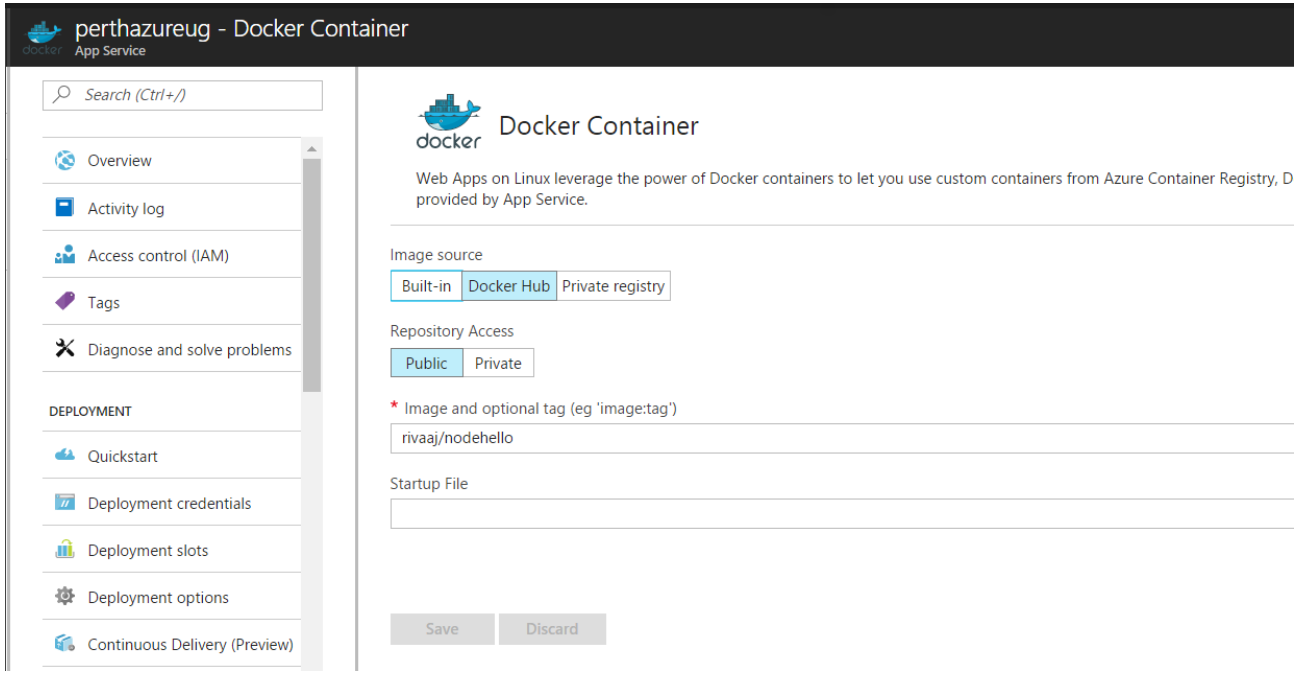
Inner-Loop Development Workflow

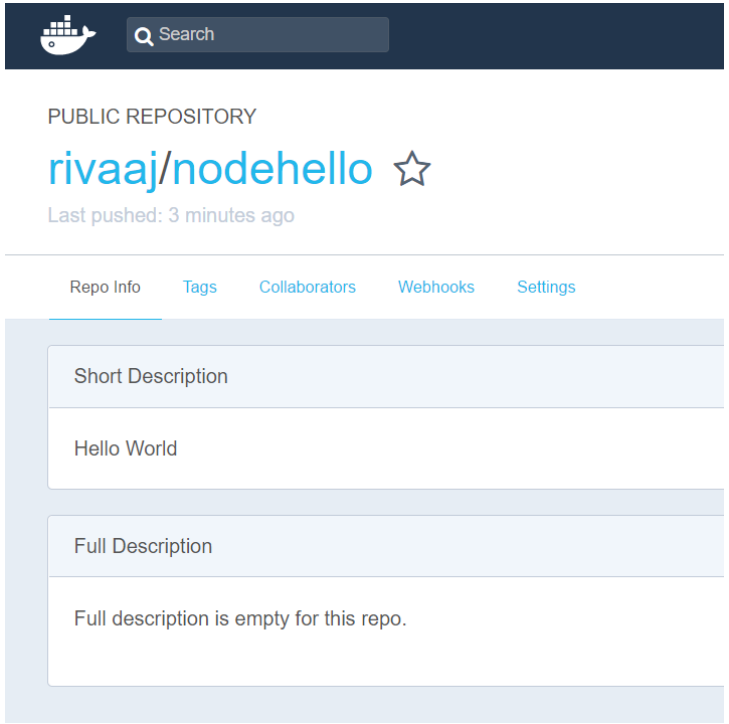


Demo – Docker Build Locally

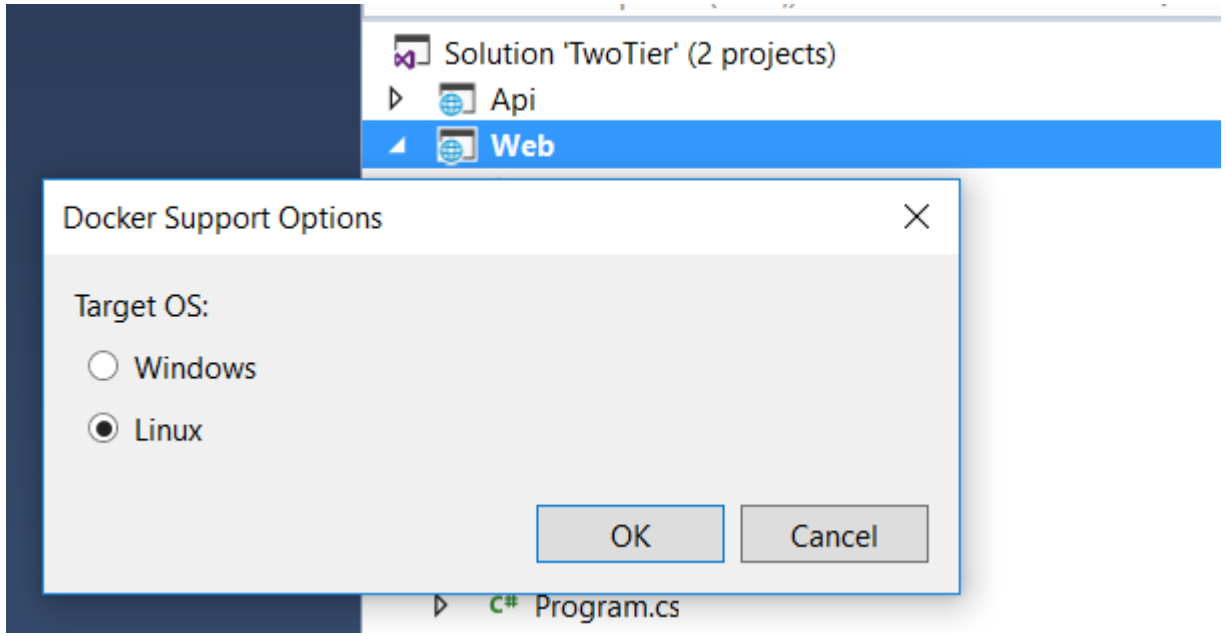
- docker build
- docker push
- Docker Registry

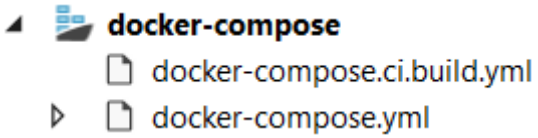
Demo – App Service





VS 2017





docker-compose.yml*

```yaml
version: '3'

services:
  api:
    image: rivaaj/apidemo
    build:
      context: ./Api
      dockerfile: Dockerfile

  web:
    image: rivaaj/webdemo
    build:
      context: ./Web
      dockerfile: Dockerfile
```

Azure Container Service

- Kubernetes
- Mesos
- Docker Swarm

Azure Container Registry

Azure Container Registry

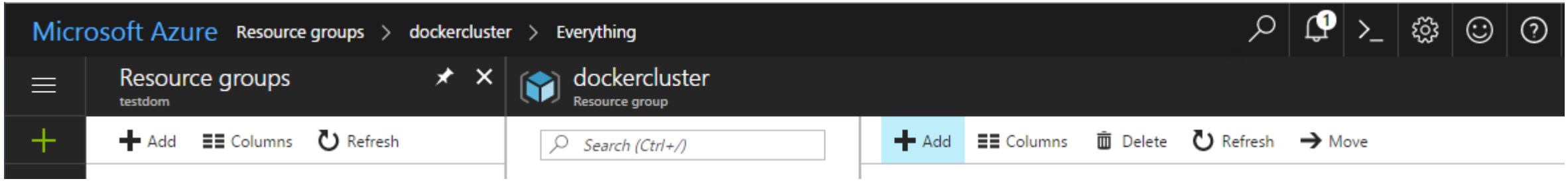
Manage a Docker private registry as a first-class Azure resource

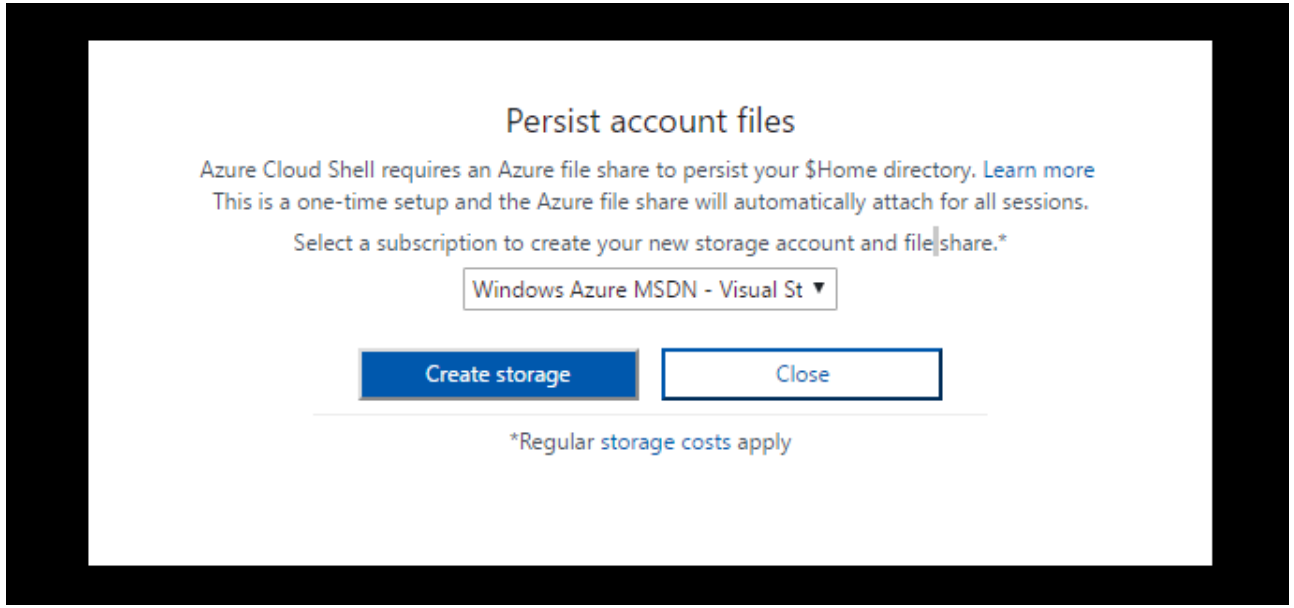
- ✔ Store and manage container images across all types of Azure deployments
- ✔ Keep container images near deployments to reduce latency and costs
- ✔ Maintain Windows and Linux container images in a single Docker registry
- ✔ Use familiar, open-source Docker command line interface (CLI) tools
- ✔ Simplify registry access management with Azure Active Directory

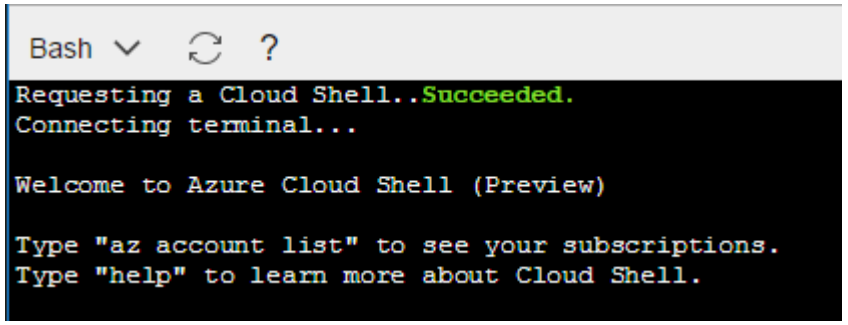
ACS Engine

The Azure Container Service Engine (`acs-engine`) generates ARM (Azure Resource Manager) templates for Docker enabled clusters on Microsoft Azure with your choice of DC/OS, Kubernetes, Swarm Mode, or Swarm orchestrators. The input to the tool is a cluster definition. The cluster definition is very similar to (in many cases the same as) the ARM template syntax used to deploy a Microsoft Azure Container Service cluster.

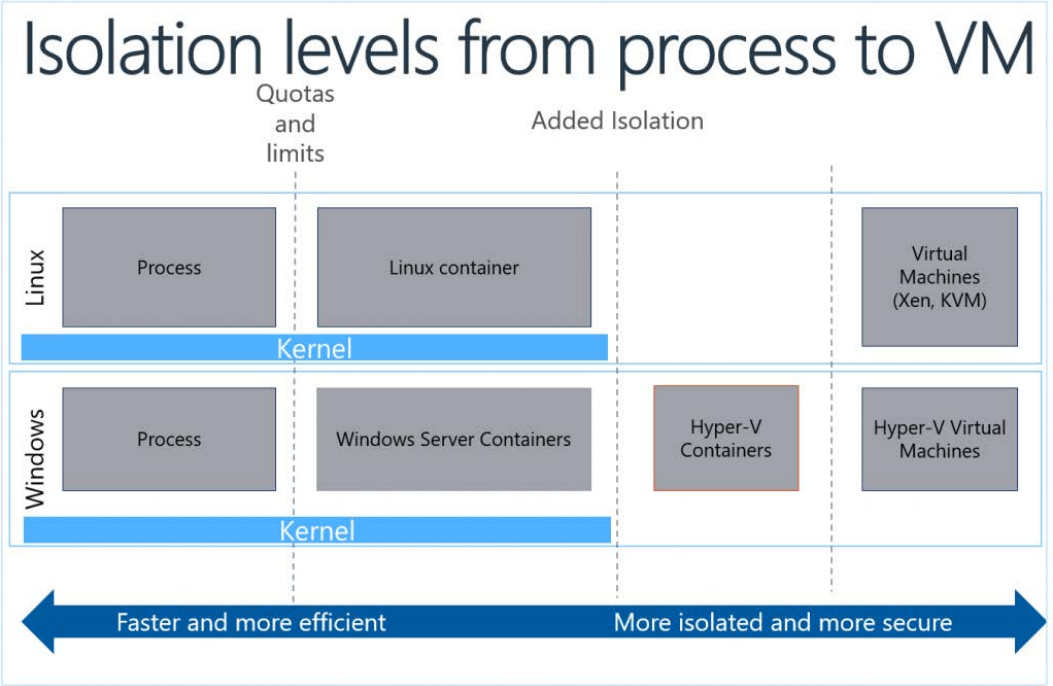
Azure Cloud Shell







Azure Service Fabric



- Deploy Windows Container (preview)
- Deploy Linux Container
- Docker compose (preview)

VM to Docker

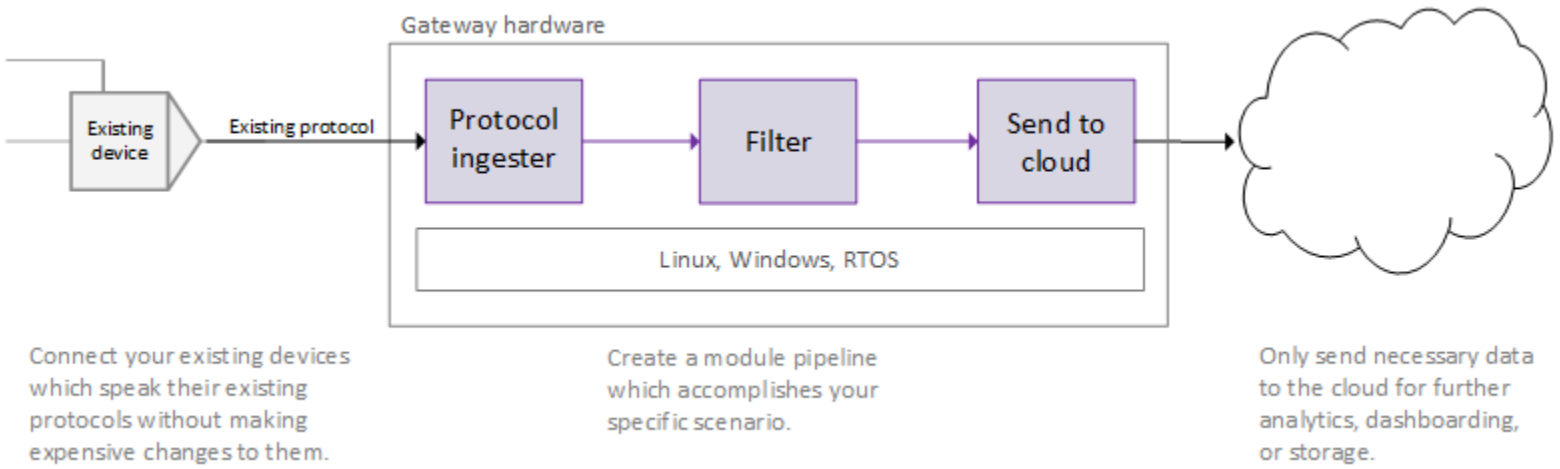
`Image2Docker` is a PowerShell module which ports existing Windows application workloads from virtual machines to Docker images. It supports multiple application types, but the initial focus is on IIS. You can use `Image2Docker` to extract ASP.NET websites from a VM, so you can run them in a Docker container with no application changes.

Azure Batch

Batch Shipyard is a tool to help provision and execute batch processing and HPC Docker workloads on Azure Batch compute pools. No experience with the Azure Batch SDK is needed; run your Dockerized tasks with easy-to-understand configuration files!

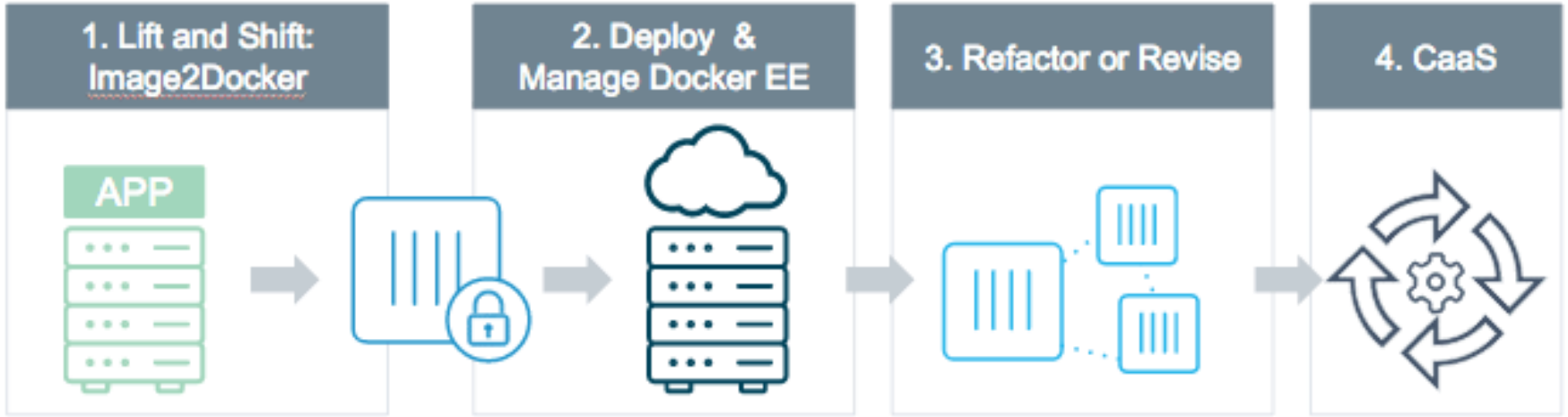
IoT Edge

This similarity means that existing solutions can evolve with the product! There will be some infrastructural changes. For example: modules will run in Docker containers and the broker used to pass messages between module code will move to a lite version of IoT Hub running locally in a module. The vast majority of this is shielded from both a module developer and gateway developer.



The Future

How Docker Modernizes Legacy Apps



App resides on **BM or VM server**

Convert to **container**

Modern Infrastructure
Move to cloud or refresh HW

Modern Microservices
Add new services or start peeling off services from monolith code base

Modern Methodologies
Integrate to CI/CD and automation systems

docker