Microsoft

Azure

Azure Developer Series

Application Migration to Azure

Peter De Tender

@pdtit

@007FFFLearning

CEO & Lead Technical Trainer at 007FFFLearning.com

April 2019

About Me…

Peter De Tender – MCT, Azure MVP

- CEO and Lead Technical Trainer of 007FFFLearning.com, +20 years IT experience, mainly datacenters and Microsoft Infrastructure background
- Full-time in Azure since 2013 (Readiness & Architect)
- Azure Advisor, Azure Certified Architect
- Technical Writer, Book author, Courseware Creator
- Living in Belgium, but traveling worldwide 90% of my time, helping larger Microsoft Partners, customers and Microsoft FTEs in learning about and using Azure, by providing workshops with passion

peter@pdtit.be

@pdtit @007FFFLearning

http://www.facebook.com/pdtit

http://www.linkedin.com/in/pdtit

Setting the scene

Overview of the workshop

About the workshop content…

About:
In this workshop, you will learn how to build a proof of concept (POC) that will transform an existing ASP.NET-based Web application to a container-based application. This POC will deliver a multi-tiered web app solution from a Virtual Machine architecture into Azure, leveraging Azure WebApps and different Azure container solutions available today. You will also migrate the underlying database from a SQL 2014 Virtual Machine architecture to SQL Azure. Easter Bonus: Every now and then, we will showcase similar steps using a Node.JS and MongoDB, migrating to Azure Web Apps, Containers and CosmosDB.

At the end of this workshop, you will have a good understanding of container concepts, Docker architecture and operations, Azure Container Services, Azure Kubernetes Services and SQL Azure PaaS solutioning.

Target Audience:
The workshop is targeted to Cloud Architects, Cloud Solution designers, developers and IT sysadmins, CIO's, CTO's and anybody else who is interested in learning about Azure, containers, application cloud migration and digital transformation.

Focus of the workshop (40%) is getting hands-on experience, complemented with presentations and whiteboard sessions (if in-person delivery).

Time Estimate:
16 hours (+/- 10 hours presentations, 6 hours of optional hands-on labs for attendees)

Workshop Agenda - Presentations

What we will talk about…

- Module 1: Digital App Transformation with Azure
- Module 2: Infrastructure as Code using ARM templates
- Module 3: Azure Database Solutions – SQL Azure
- Module 4: Azure App Services – Azure Web Apps (.NET + Node.JS)
- Module 5: Introduction to Docker
- Module 6: Deploying Azure Container Registry / Azure Container Instance
- Module 7: Migrating Apps to Azure Container Services / Kubernetes Services
- Module 8: ACS / AKS Management and Monitoring

Workshop Agenda – Hands-On-Labs

Learn by doing…

- Module 2: Infrastructure as Code using ARM templates
  - **Lab 1:** Setup your Azure subscription and deploy the source Virtual Machine environment with Visual Studio 2017
- Module 3: Azure Database Solutions – SQL Azure
  - **Lab 2:** Migrating a SQL VM database to SQL Azure using SQL Management Studio
- Module 4: Azure App Services – Azure Web Apps
  - **Lab 3:** Migrating your legacy ASP.NET application to Azure Web Apps with Visual Studio 2017
  - Easter Egg Bonus: Deploying a Node.JS app with MongoDB / CosmosDB
- Module 5: Introduction to Docker
  - **Lab 4:** Containerizing your legacy ASP.NET application with Docker CE for Windows

Technical Requirements

What you need…

<Could vary based on the actual delivery-method>, but overall:

- Client workstation running recent Windows, Linux or Mac OS and latest internet browser
- Access to ports 80 (HTTP), 443 (HTTPS) and 3389 (Remote Desktop)
- Full Azure subscription (MSDN, AzurePass, Paid subscription, AE, CSP,…)
- Lab consumption estimate: $15-35 (when shutdown all resources)

Questions and HOL support

msdevseriessupport@007FFFLearning.com

Subject: Azure Developer Series – Containers

Response Time: within 4-8 hours

Check GitHub for FAQ and Updates:
http://www.github.com/007FFFLearning/MSDevSeriesSupport

Microsoft

Azure

Application Migration

Docker Containers

Peter De Tender

@pdtit @007FFFlearning

Key Objectives

What you will learn in this section

- Introduction to Containers
- Docker Containers Overview
- Migrating applications into Containers
- Azure Container Registry
- Azure Container Instance

Microsoft

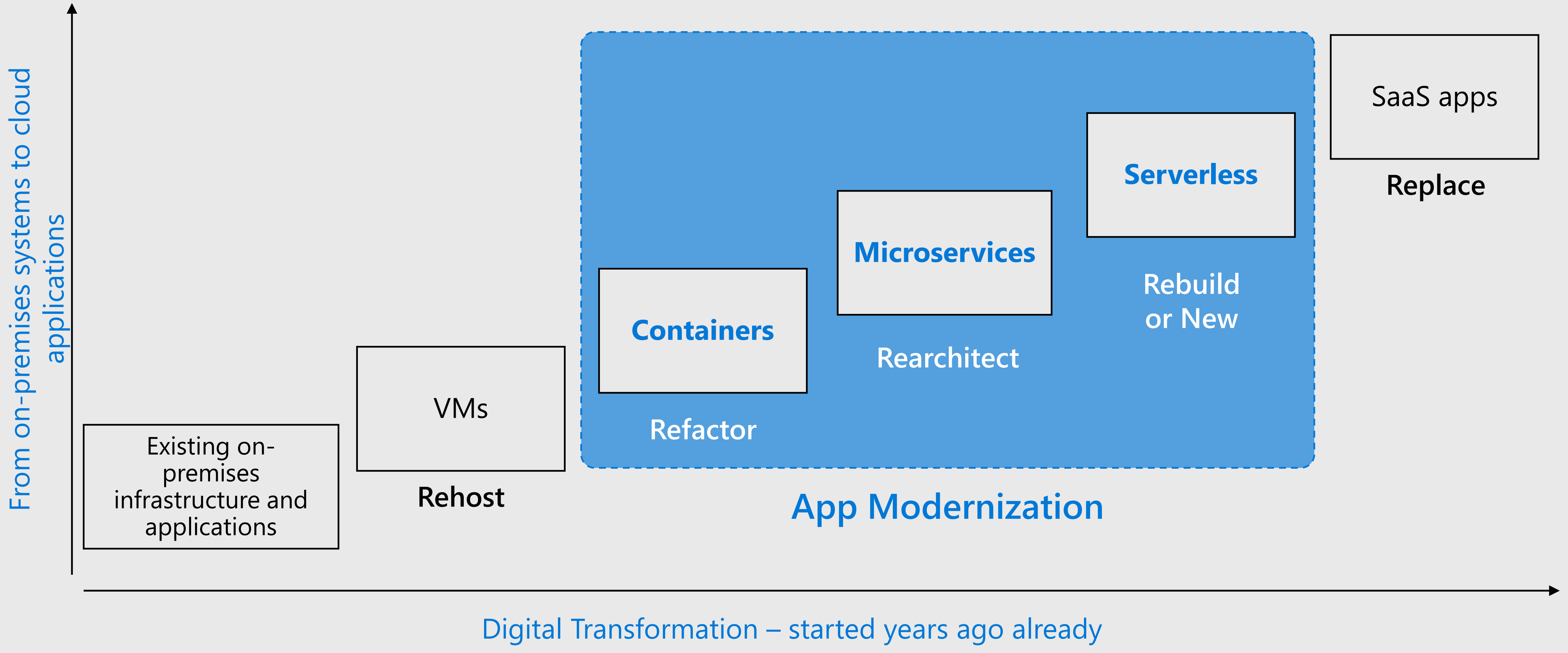
Azure

Containers

What are Containers, why and how using them

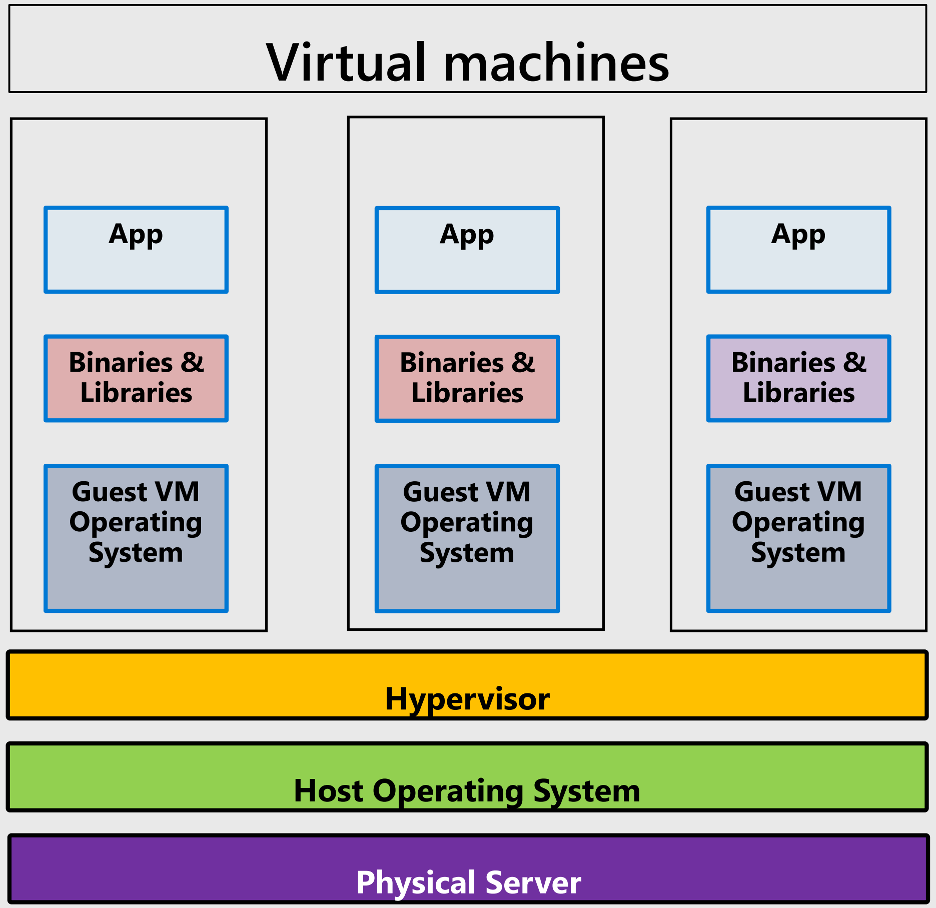
Introduction to Containers

The journey to the cloud

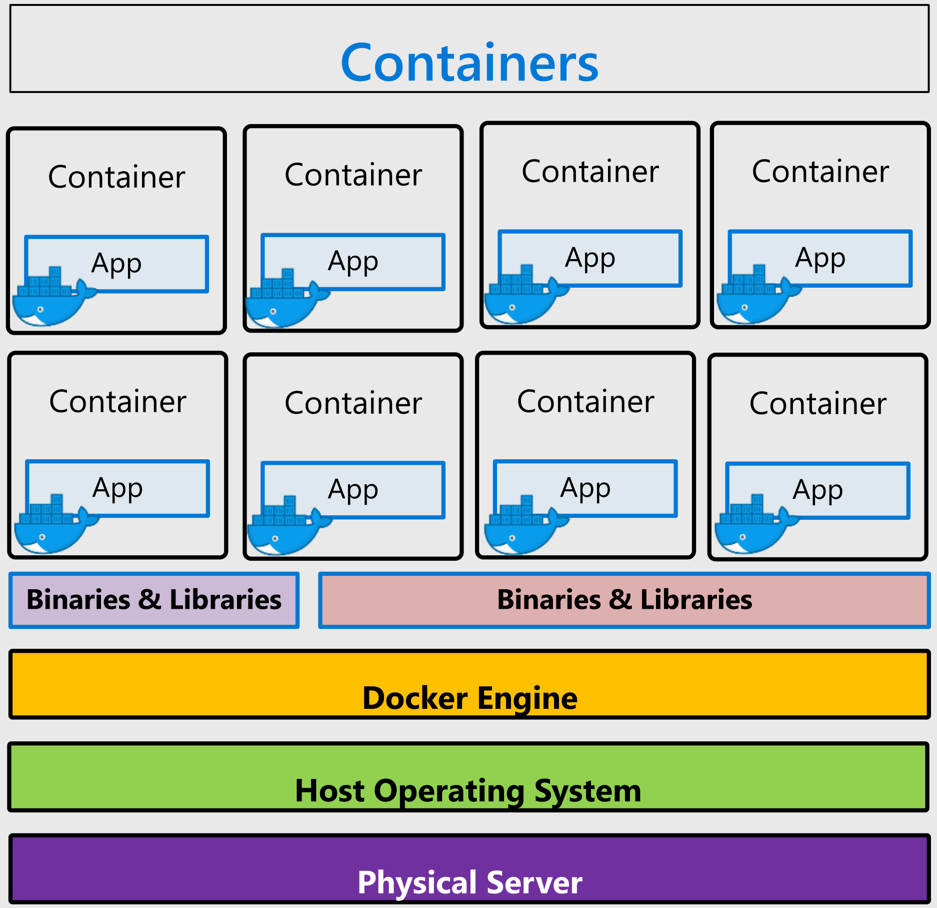


From on-premises systems to cloud applications
Existing on-premises infrastructure and applications
VMs
Rehost
Containers
Refactor
Microservices
Rearchitect
Serverless
Rebuild or New
App Modernization
SaaS apps
Replace
Digital Transformation – started years ago already

What are containers?



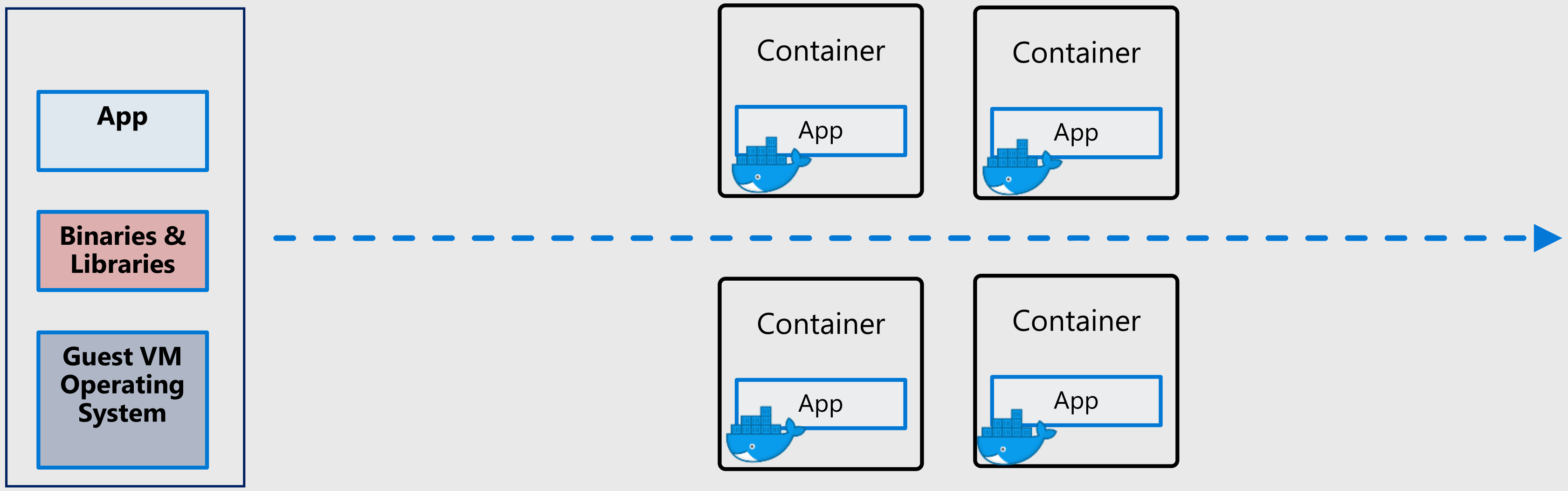
- Virtualize the **hardware**
- **VMs** as units of scaling
- Hypervisor **dependent**
- **Not** easily movable



- Virtualize the **operating system**
- **Applications** as units of scaling
- Platform **independent**
- **Easily** movable across environments (on-premises, multi-cloud)

How do containers help in app modernization?

Containers are stand-alone, smaller silos of app instances, running at scale



Existing/New Applications

(=rehome)

Lift and shift to containers

(= rehost, refactor)

Running Containers on Azure: a full set of choices

App Service	Service Fabric	Kubernetes Service	Container Instance	Partner Ecosystem
Deploy web apps or APIs using **containers** in a PaaS environment	Modernize **.NET applications** to microservices using **Windows Server containers**	Scale and orchestrate **Linux containers using Kubernetes**	**Elastically** burst from your **Azure Kubernetes Service** (AKS) cluster	Bring your **Partner solutions** that run great on Azure

Azure Container Registry

Docker Hub

Choice of developer tools and clients

How Containers help in App Migration

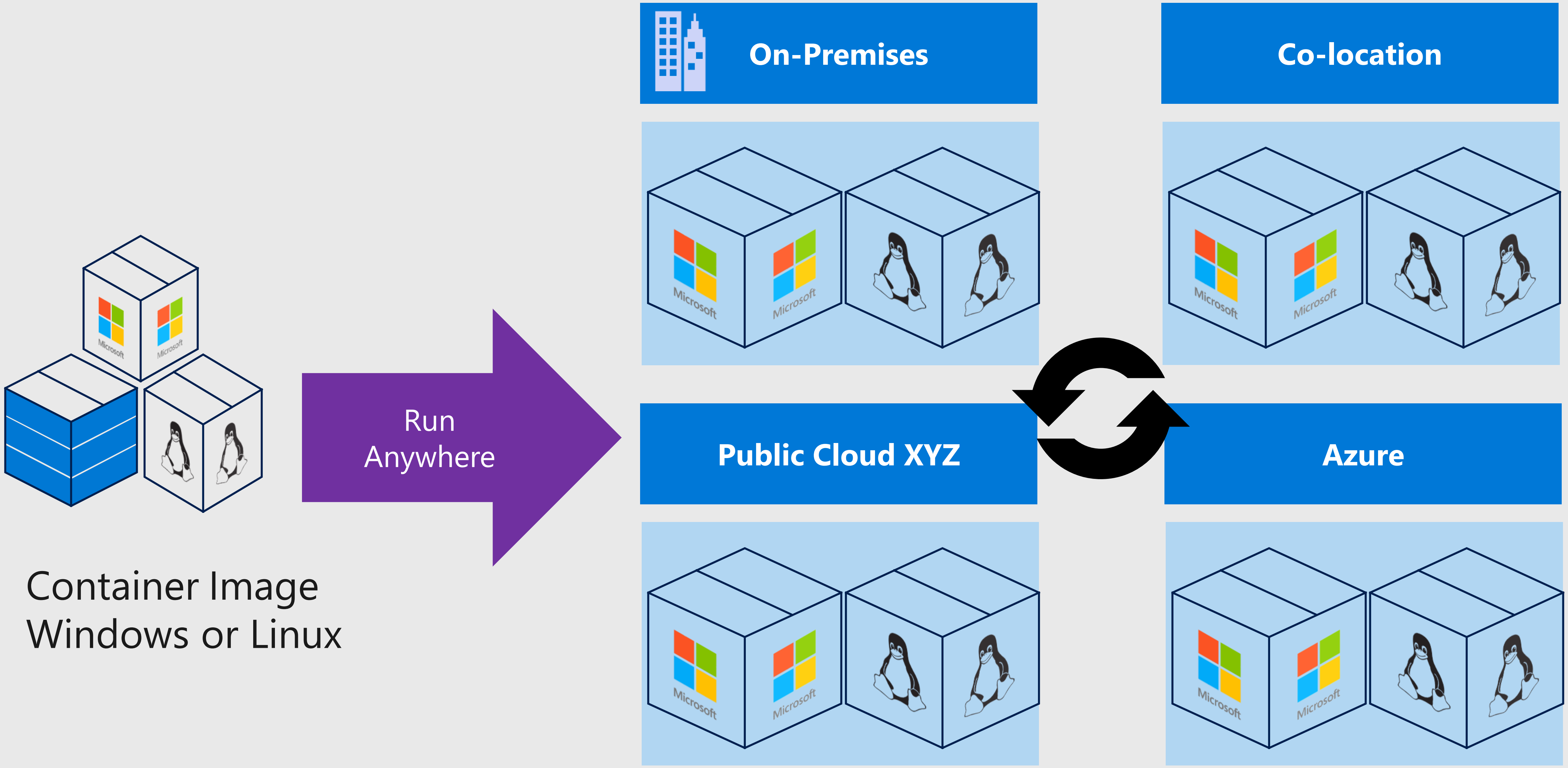
Enterprises can’t get away from their legacy apps just like that:

- Expensive
- Risk
- Developers are gone
- Complexity

Enterprises benefit from Containers:

- Legacy apps are supported
- Containers are future-proof
- Cost optimized
- Secure

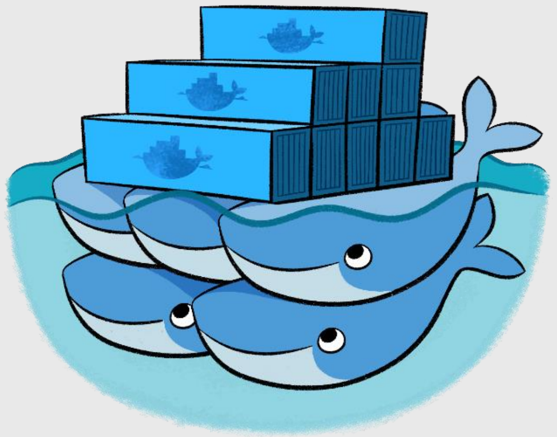
How Containers help in App Migration



Docker Containers Overview

Docker Containers - Overview

What is Docker?



- Leading Open-Source Containerization Platform

- Natively Supported in Azure

- Cross-Platform (Win, Linux,...)

Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries – anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in

Source: www.docker.com

Docker High-Level Architecture

Docker Concepts

1 Client

Docker build

Docker pull

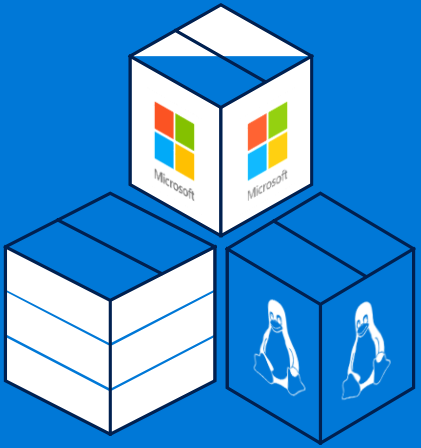
Docker Run

Docker …

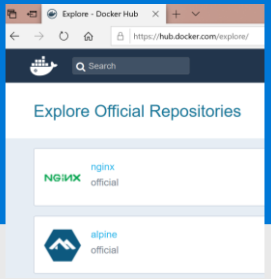
2 Host

Docker Daemon / Docker Service

Containers



Images



Explore Official Repositories

nginx official

alpine official

3 Registry

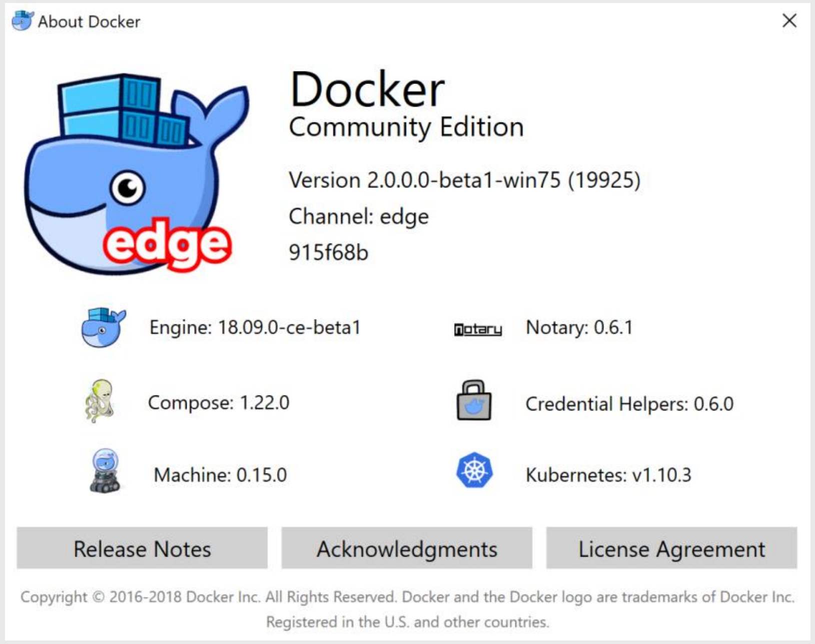
Docker Hub

Azure Container Registry

Cloud XYZ Container Registry

Docker Host

Windows-based (Win10/Win2016-2019)

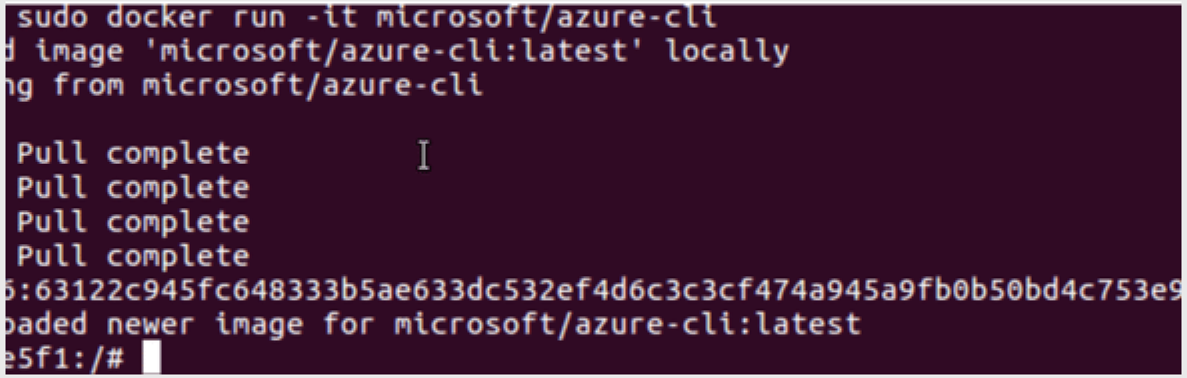


Docker
Community Edition
Version 2.0.0.0-beta1-win75 (19925)
Channel: edge
915f68b

Engine: 18.09.0-ce-beta1
Notary: 0.6.1
Compose: 1.22.0
Credential Helpers: 0.6.0
Machine: 0.15.0
Kubernetes: v1.10.3

Copyright © 2016-2018 Docker Inc. All Rights Reserved. Docker and the Docker logo are trademarks of Docker Inc. Registered in the U.S. and other countries.

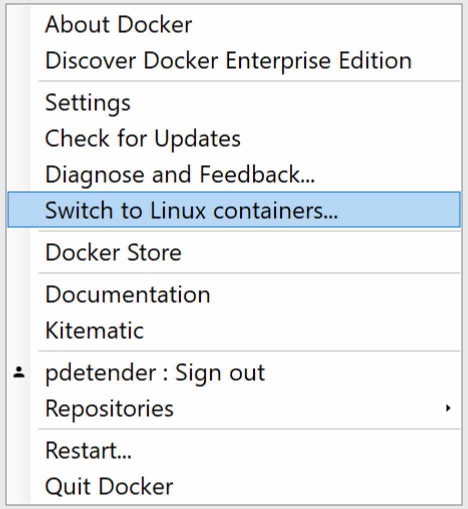
Linux-based



```
 sudo docker run -it microsoft/azure-cli
d image 'microsoft/azure-cli:latest' locally
ng from microsoft/azure-cli

 Pull complete
 Pull complete
 Pull complete
 Pull complete
5:63122c945fc648333b5ae633dc532ef4d6c3c3cf474a945a9fb0b50bd4c753e9
oaded newer image for microsoft/azure-cli:latest
e5f1:/#
```

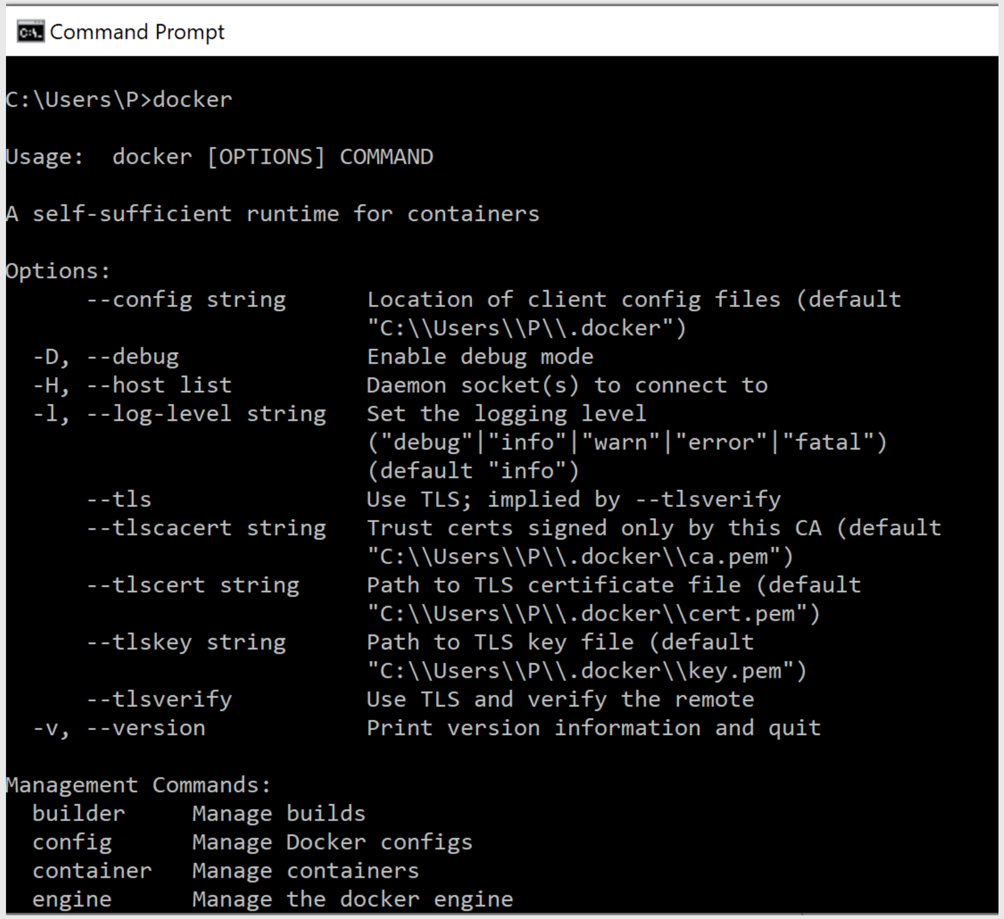
Docker Containers on Windows

- Runs on Windows 10 client or Windows Server 2016/2019
- Supports both Windows and Linux Containers, and you can easily switch
- Requires Hyper-V or « Containers » Feature
- CLI integration with PowerShell or command prompt
- Docker CE = Free !!



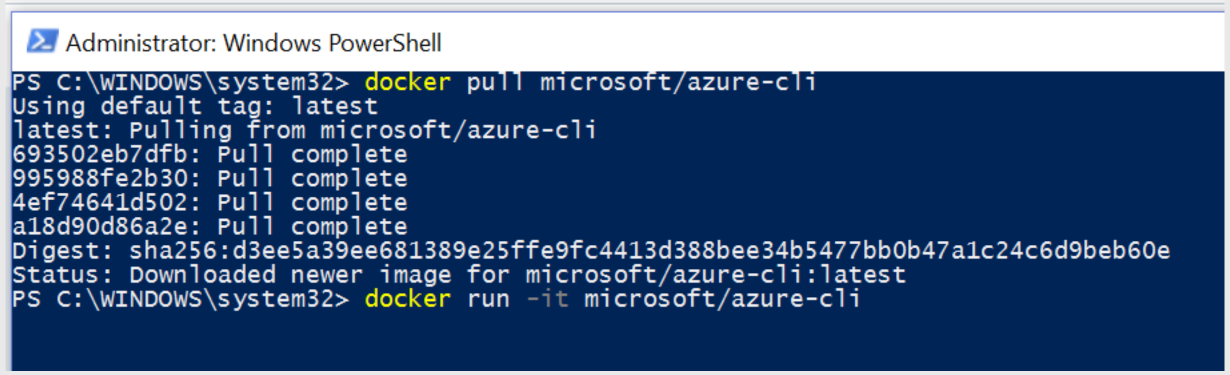
Docker CLI

Command Line Interface



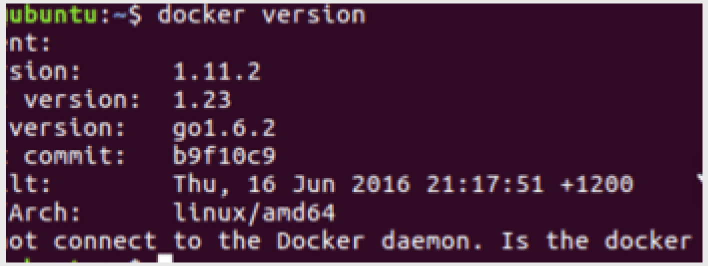
Command Prompt

```
C:\Users\P>docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "C:\\Users\\P\\.docker")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level
                           ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default
                           "C:\\Users\\P\\.docker\\ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "C:\\Users\\P\\.docker\\cert.pem")
      --tlskey string      Path to TLS key file (default
                           "C:\\Users\\P\\.docker\\key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Management Commands:
  builder     Manage builds
  config      Manage Docker configs
  container   Manage containers
  engine      Manage the docker engine
```



Administrator: Windows PowerShell

```
PS C:\WINDOWS\system32> docker pull microsoft/azure-cli
Using default tag: latest
latest: Pulling from microsoft/azure-cli
693502eb7dfb: Pull complete
995988fe2b30: Pull complete
4ef74641d502: Pull complete
a18d90d86a2e: Pull complete
Digest: sha256:d3ee5a39ee681389e25ffe9fc4413d388bee34b5477bb0b47a1c24c6d9beb60e
Status: Downloaded newer image for microsoft/azure-cli:latest
PS C:\WINDOWS\system32> docker run -it microsoft/azure-cli
```



```
ubuntu:~$ docker version
nt:
sion:       1.11.2
 version:   1.23
version:    go1.6.2
 commit:    b9f10c9
lt:         Thu, 16 Jun 2016 21:17:51 +1200
Arch:       linux/amd64
ot connect to the Docker daemon. Is the docker
```

Demo

Running Docker for Windows

Docker Host

Where do you run your Docker Container Images?

On top of a « host » Operating System, like:

- Windows 10
- Windows Server 2016/2019
- Linux
- Mac OS

On top of a « Docker » cloud platform:

- Azure Container Instance
- Azure Container Services
- Azure Virtual Machines

Where do Docker Images come from?

Docker Hub

- Hub.docker.com
- Free resource of PUBLIC images
- Option to create PRIVATE images

Docker Hub

Cloud Container Registry

- Library of Docker Images
- Azure Container Registry
- Mainly used for storing PRIVATE images

Azure Container Registry

Running a Docker Container

```
docker run -i -t ubuntu /bin/bash
```

- `docker`: Docker CLI command
- `run -i -t`: Run container with interactive terminal
- `ubuntu`: Pull "ubuntu" image from Docker Hub or local registry
- `/bin/bash`: Command to execute inside the container

Common Docker CLI operations

`docker run` - Use an image to run a container

`docker pull` - Pull an image from a registry

`docker build` - Build a Docker image

`docker images` - List available Docker images

`docker ps` - List running Docker containers

`docker exec` - Execute a command in a container

`docker stop` - Stop a running container

Demo

Common Docker CLI Operations

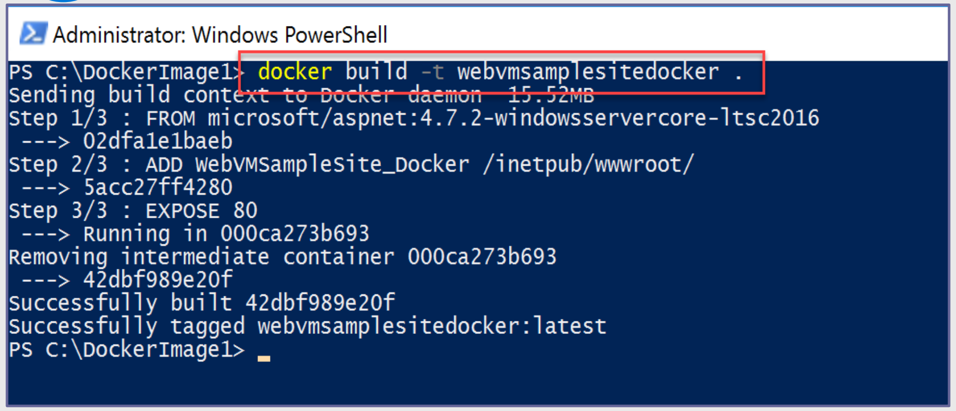
Migrating apps to Docker Images

Building a Docker Image - CLI

1 **`docker build`** - Build a Docker image

2 **`docker images`** - List available Docker images

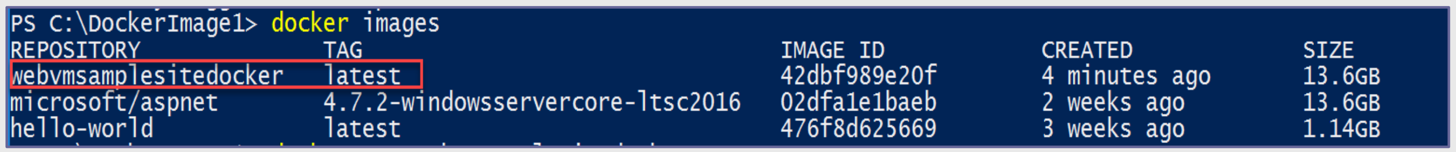
1



Administrator: Windows PowerShell

```
PS C:\DockerImage1> docker build -t webvmsamplesitedocker .
Sending build context to Docker daemon  15.52MB
Step 1/3 : FROM microsoft/aspnet:4.7.2-windowsservercore-ltsc2016
 ---> 02dfa1e1baeb
Step 2/3 : ADD WebVMSampleSite_Docker /inetpub/wwwroot/
 ---> 5acc27ff4280
Step 3/3 : EXPOSE 80
 ---> Running in 000ca273b693
Removing intermediate container 000ca273b693
 ---> 42dbf989e20f
Successfully built 42dbf989e20f
Successfully tagged webvmsamplesitedocker:latest
PS C:\DockerImage1> _
```

2



```
PS C:\DockerImage1> docker images
REPOSITORY              TAG                                  IMAGE ID            CREATED             SIZE
webvmsamplesitedocker   latest                               42dbf989e20f        4 minutes ago       13.6GB
microsoft/aspnet        4.7.2-windowsservercore-ltsc2016     02dfa1e1baeb        2 weeks ago         13.6GB
hello-world             latest                               476f8d625669        3 weeks ago         1.14GB
```

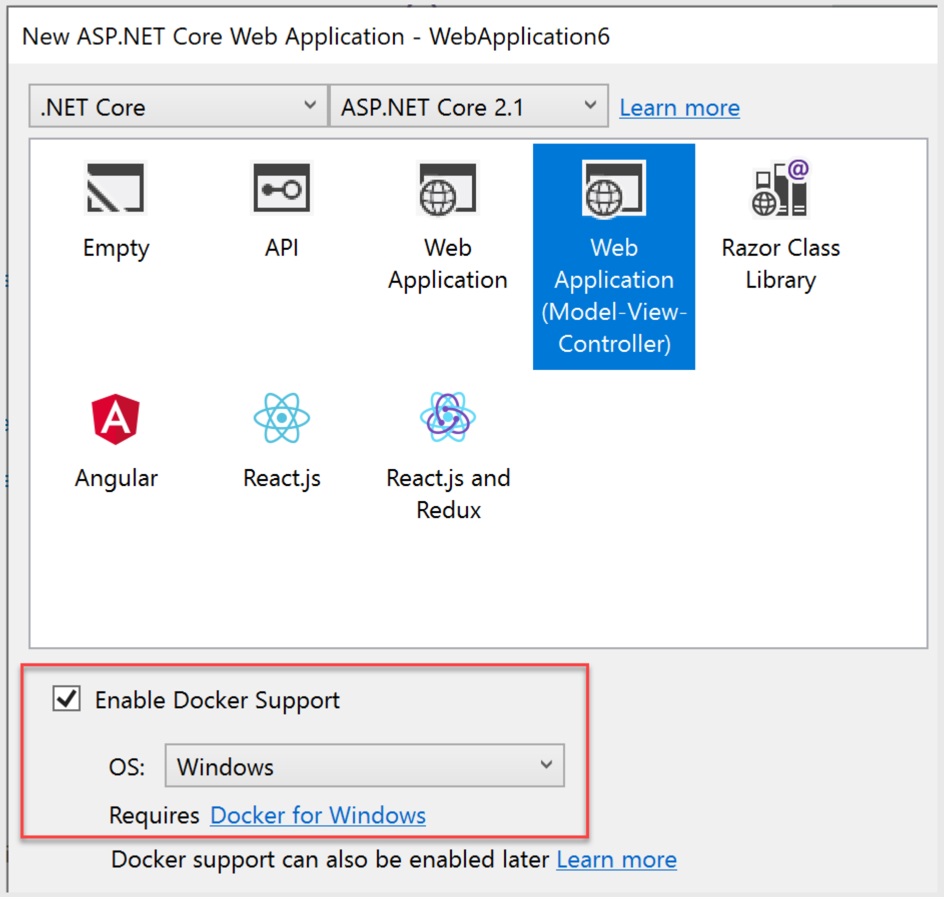
Building a Docker Image - DockerFile

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using <docker build>, users can create an automated build that executes several command-line instructions in succession.

```
FROM microsoft/aspnet:4.7.2-
windowsservercore-ltsc2016

ADD WebVMSampleSite_Docker
/inetpub/wwwroot/

EXPOSE 80
```

Building a Docker Image – VS2017

When using Visual Studio 2017 together with Docker for Windows on the same client, you get Docker integration features in VS2017…


New ASP.NET Core Web Application - WebApplication6

.NET Core | ASP.NET Core 2.1 | Learn more

☑ Enable Docker Support

OS: Windows

Requires Docker for Windows

Docker support can also be enabled later Learn more

https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/docker/visual-studio-tools-for-docker?view=aspnetcore-2.1

Demo

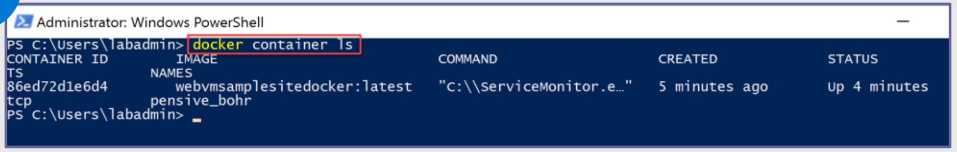
Building a Docker Image (CLI and VS2017)

Troubleshooting Docker Containers

1 **`docker container ls`** – Lists all containers on a host

2 **`docker inspect`** – Shows all information of a container

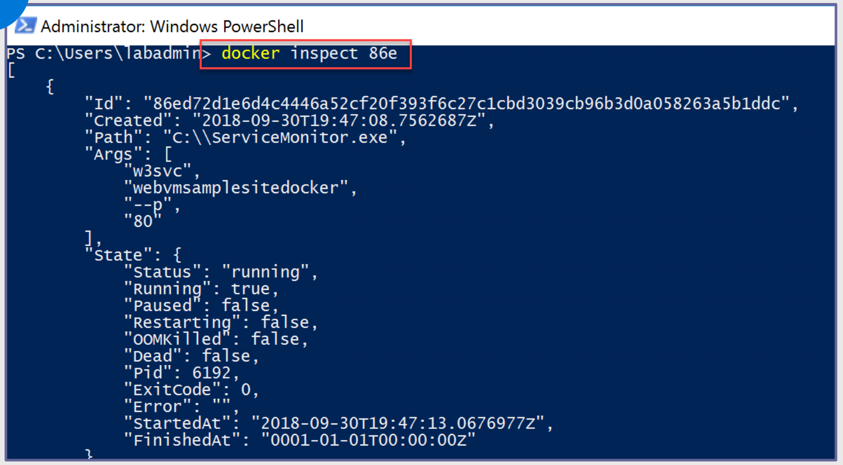
1



Administrator: Windows PowerShell

```
PS C:\Users\labadmin> docker container ls
CONTAINER ID        IMAGE                           COMMAND                  CREATED             STATUS
TS                NAMES
86ed72d1e6d4        webvmsamplesitedocker:latest    "C:\\ServiceMonitor.e…"   5 minutes ago       Up 4 minutes
tcp               pensive_bohr
PS C:\Users\labadmin> _
```

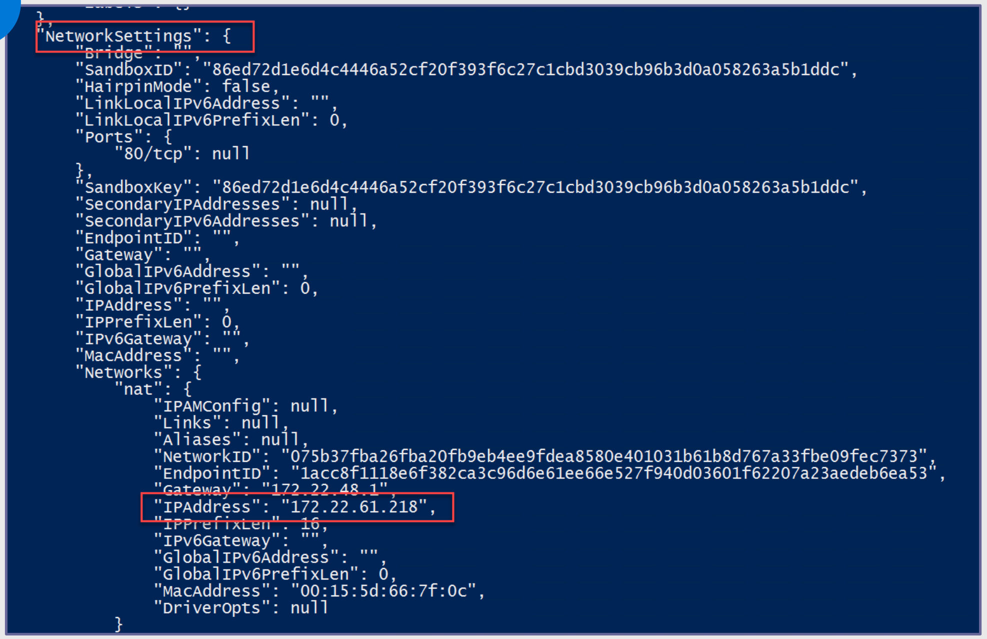
2



Administrator: Windows PowerShell

```
PS C:\Users\labadmin> docker inspect 86e
[
    {
        "Id": "86ed72d1e6d4c4446a52cf20f393f6c27c1cbd3039cb96b3d0a058263a5b1ddc",
        "Created": "2018-09-30T19:47:08.7562687Z",
        "Path": "C:\\ServiceMonitor.exe",
        "Args": [
            "w3svc",
            "webvmsamplesitedocker",
            "--p",
            "80"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 6192,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2018-09-30T19:47:13.0676977Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
```

2



```
            "Labels": {}
        },
        "NetworkSettings": {
            "Bridge": "",
            "SandboxID": "86ed72d1e6d4c4446a52cf20f393f6c27c1cbd3039cb96b3d0a058263a5b1ddc",
            "HairpinMode": false,
            "LinkLocalIPv6Address": "",
            "LinkLocalIPv6PrefixLen": 0,
            "Ports": {
                "80/tcp": null
            },
            "SandboxKey": "86ed72d1e6d4c4446a52cf20f393f6c27c1cbd3039cb96b3d0a058263a5b1ddc",
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "EndpointID": "",
            "Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "",
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
            "MacAddress": "",
            "Networks": {
                "nat": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "NetworkID": "075b37fba26fba20fb9eb4ee9fdea8580e401031b61b8d767a33fbe09fec7373",
                    "EndpointID": "1acc8f1118e6f382ca3c96d6e61ee66e527f940d03601f62207a23aedeb6ea53",
                    "Gateway": "172.22.48.1",
                    "IPAddress": "172.22.61.218",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "00:15:5d:66:7f:0c",
                    "DriverOpts": null
                }
```

Lab

Containerizing an ASP.NET application with Docker

https://github.com/007FFFLearning/MSDevSeriesSupport

Lab 3 – Quick Instructions

1. (Assumption is you finished Lab 1 – Lab 3)
2. Download the “Lab 4” Guide from GitHub (PDF)
3. Task 1: Install Docker
4. Task 2: Build CloudShop app container
5. Task 3: Run Cloudshop container
6. When having questions: msdevseriessupport@007FFFLearning.com

Lab 4 Instructions

See your trainer or workshop host for details

Section Take-Aways

1. Containers are now what VMs were in the early 2000's
2. Containers are an enabler for legacy app migration to cloud
3. Docker is the reference in containers, supporting both Linux and Windows-based flavors

Microsoft

Azure

Questions?

Peter De Tender

@pdtit @007FFFlearning

Microsoft

Azure

Next Module...

Azure Container Registry & Instance

Peter De Tender

@pdtit @007FFFlearning