



Containers with Microsoft Azure Overview

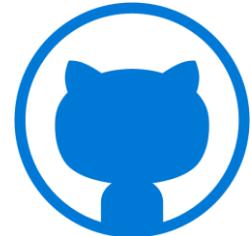
Mathieu Benoit
CSA - Cloud Application Development | OCP | Microsoft

Quebec, Canada | October 26

Speaker as Code

```
{
```

```
  "firstName": "Mathieu",
  "lastName": "Benoit",
  "city": "Quebec",
  "country": "Canada",
  "company": "Microsoft",
  "role": "Cloud Solution Architect",
  "gitHub": "mathieu-benoit",
  "blog": "aka.ms/mabenoit",
  "misc": [
    "software development",
    "cloud + web + mobile",
    "agile & devops enthusiast",
    "continuous learning"
  ]
}
```



[GitHub account](#)



[Personal blog](#)

Agenda

1. Introduction
2. Application Modernization & Innovation
3. Microservices
4. Serverless
5. Containers
6. Other initiatives around Containers
7. Conclusion & Resources

Session objectives and takeaways

At the end of this session, you should be better able to understand...

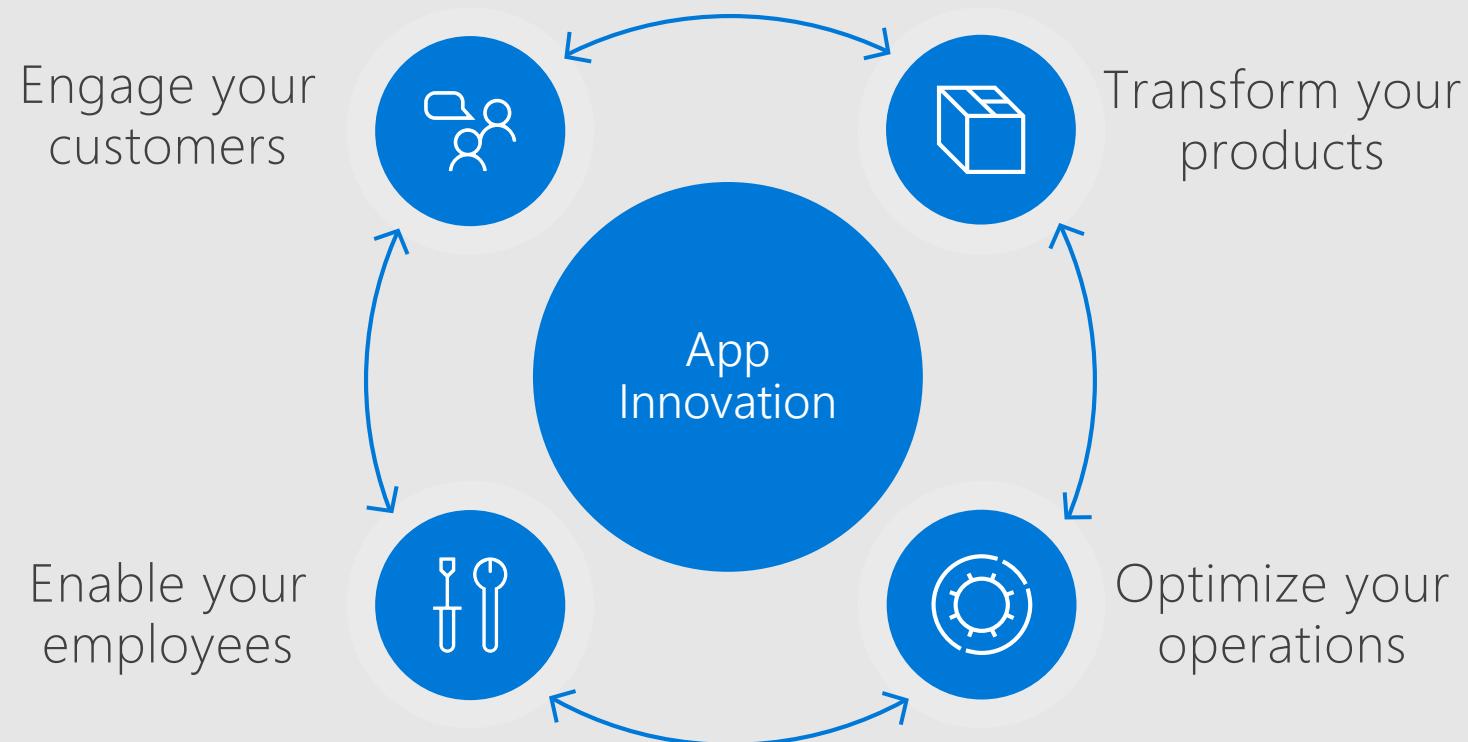
- The momentum with OSS and Azure
- The momentum with Containers and Azure
- The difference between Microservices, Serverless and Containers
- The different offers Azure provides for your Containers workloads and how/where to start your journey with

But it's not:

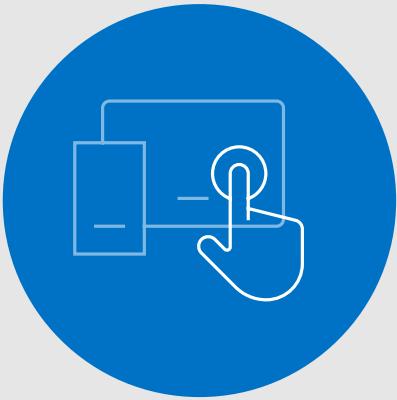
- A Docker training
- A Kubernetes training
- A deep Containers (on Azure) conversation

App Innovation & Modernization

App Innovation **is** Digital Transformation



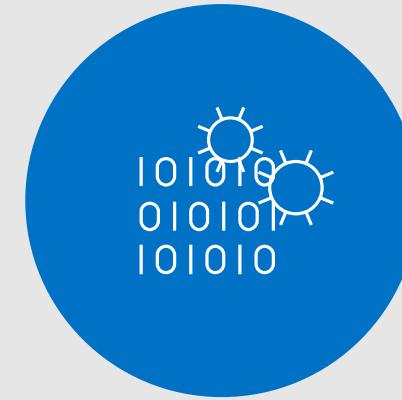
Continuous Innovation



Cross-device
Experiences



Open & Hybrid
Cloud



Data-driven
Intelligence

Azure + OSS momentum

1.5x YoY

Growth of **Node.js** web apps deployed

15% MoM

Cloud Foundry driven consumption growth

#1

Kubernetes is the most used orchestrator in ACS

2.5x QoQ

Growth of **serverless Node.js** apps in Azure

8.7% MoM

Linux growth rate in Azure

1.4x

Linux growth rate vs. Windows

+7P

Increased monthly growth rate for **Red Hat PAYG** (@15% MoM)

Azure + OSS App Plat

Languages



PaaS

Pivotal
Cloud Foundry



Data



OSS DevOps



ANSIBLE
by Red Hat®

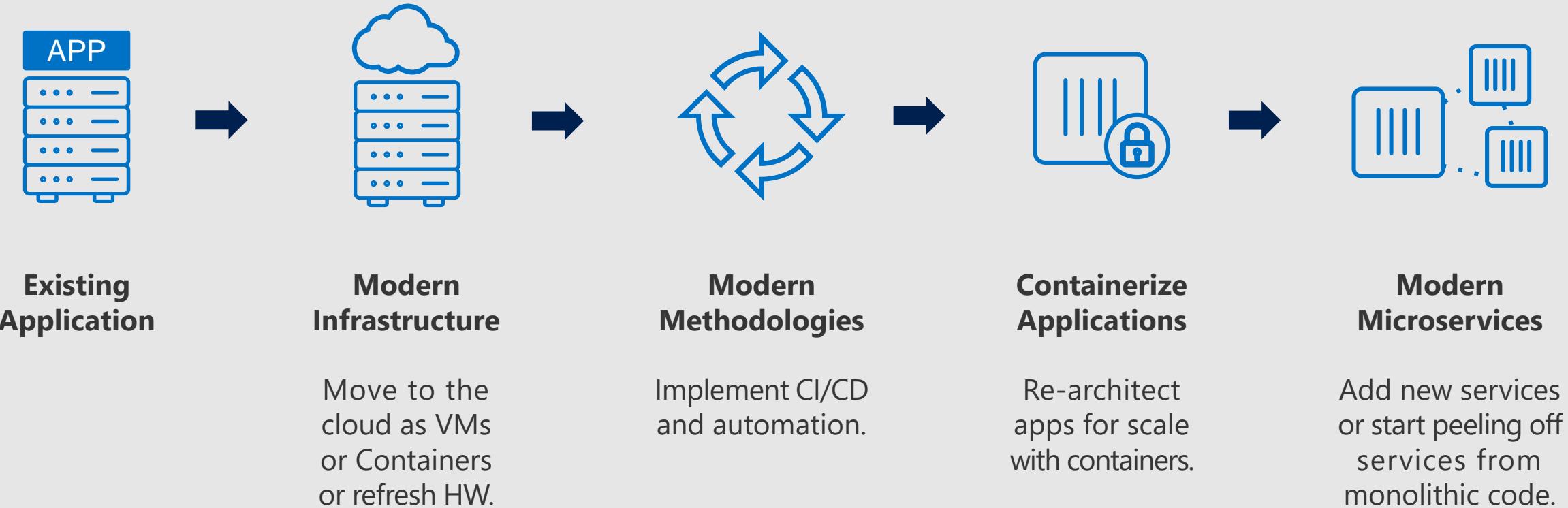


Building Blocks

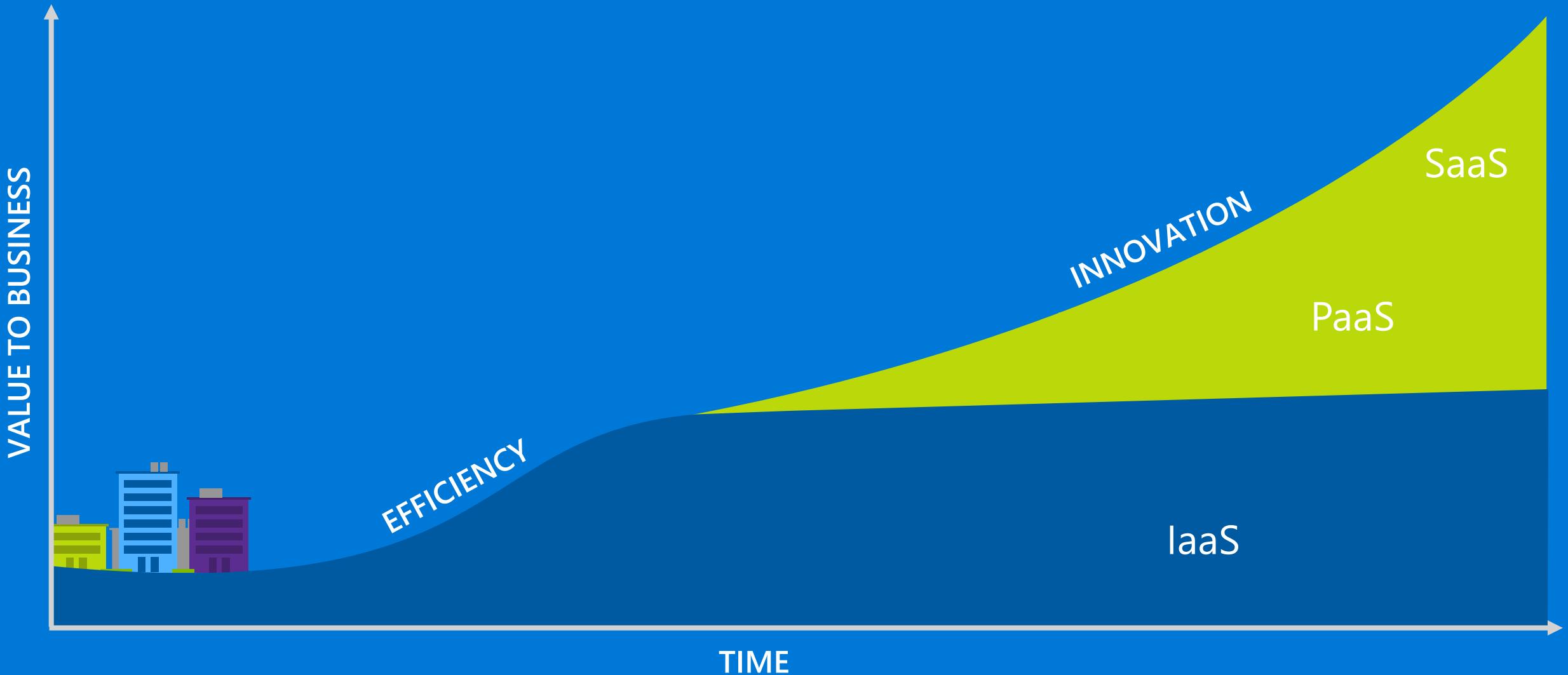
Linux



From traditional app to modern app



From infrastructure to innovation



Balance of responsibility

Balance of control and responsibility depends on the category of the service

MOVE-IN READY

Use immediately with minimal configuration

SOME ASSEMBLY REQUIRED

Existing services are a starting point, with additional configuration for a custom fit

BUILD FROM THE GROUND UP

Building blocks, create your own solution or apps from scratch

| Responsibility | On-Prem | IaaS | PaaS | SaaS |
|----------------|----------|-----------|-----------|-----------|
| Applications | Customer | Customer | Customer | Microsoft |
| Data | Customer | Customer | Customer | Microsoft |
| Runtime | Customer | Customer | Microsoft | Microsoft |
| Middleware | Customer | Customer | Microsoft | Microsoft |
| O/S | Customer | Customer | Microsoft | Microsoft |
| Virtualization | Customer | Microsoft | Microsoft | Microsoft |
| Servers | Customer | Microsoft | Microsoft | Microsoft |
| Storage | Customer | Microsoft | Microsoft | Microsoft |
| Networking | Customer | Microsoft | Microsoft | Microsoft |

█ Customer █ Microsoft

App modernization options



Microservices

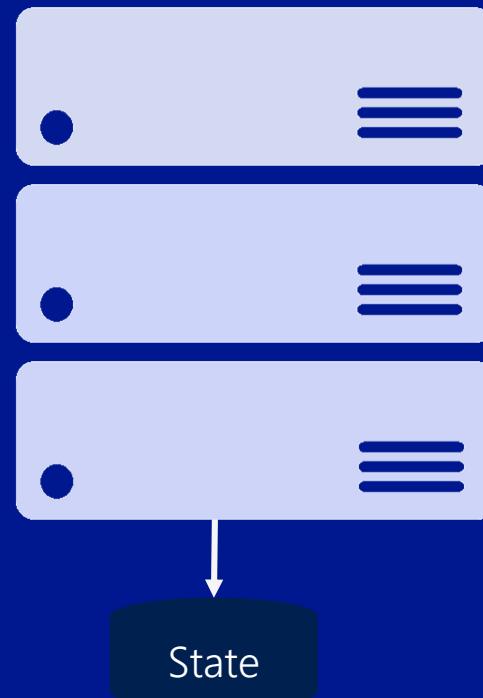
Serverless

Containers

Microservices

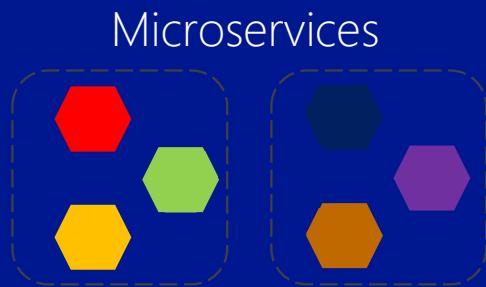
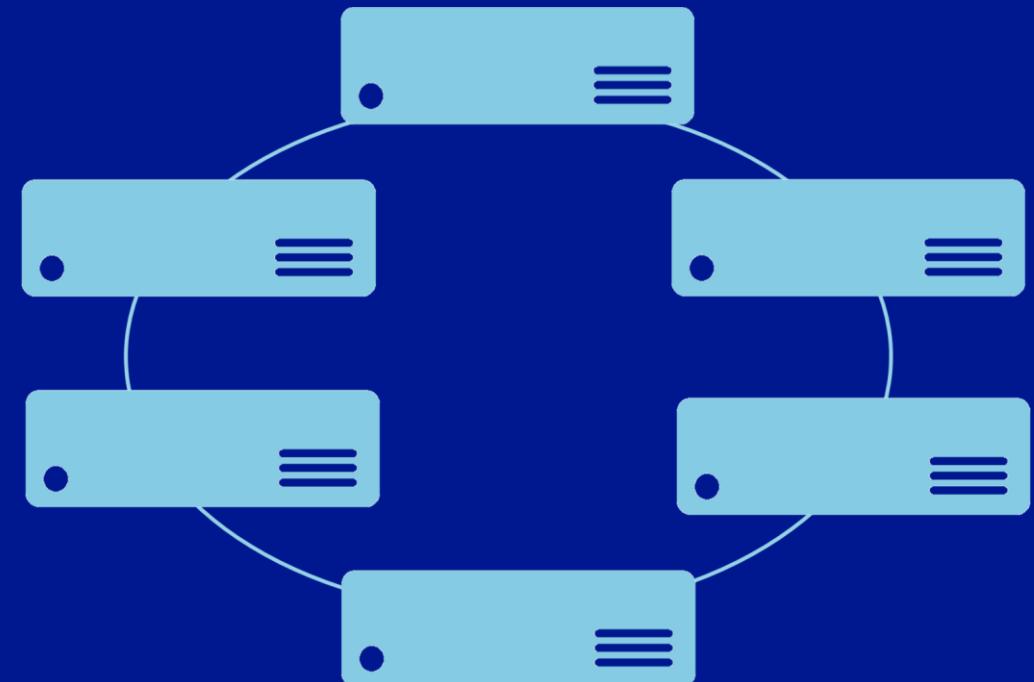
Monolithic application approach

- A monolithic application has most of its functionality within a single process that is commonly componentized with libraries.
- Scales by cloning the app on multiple servers/VMs/Containers



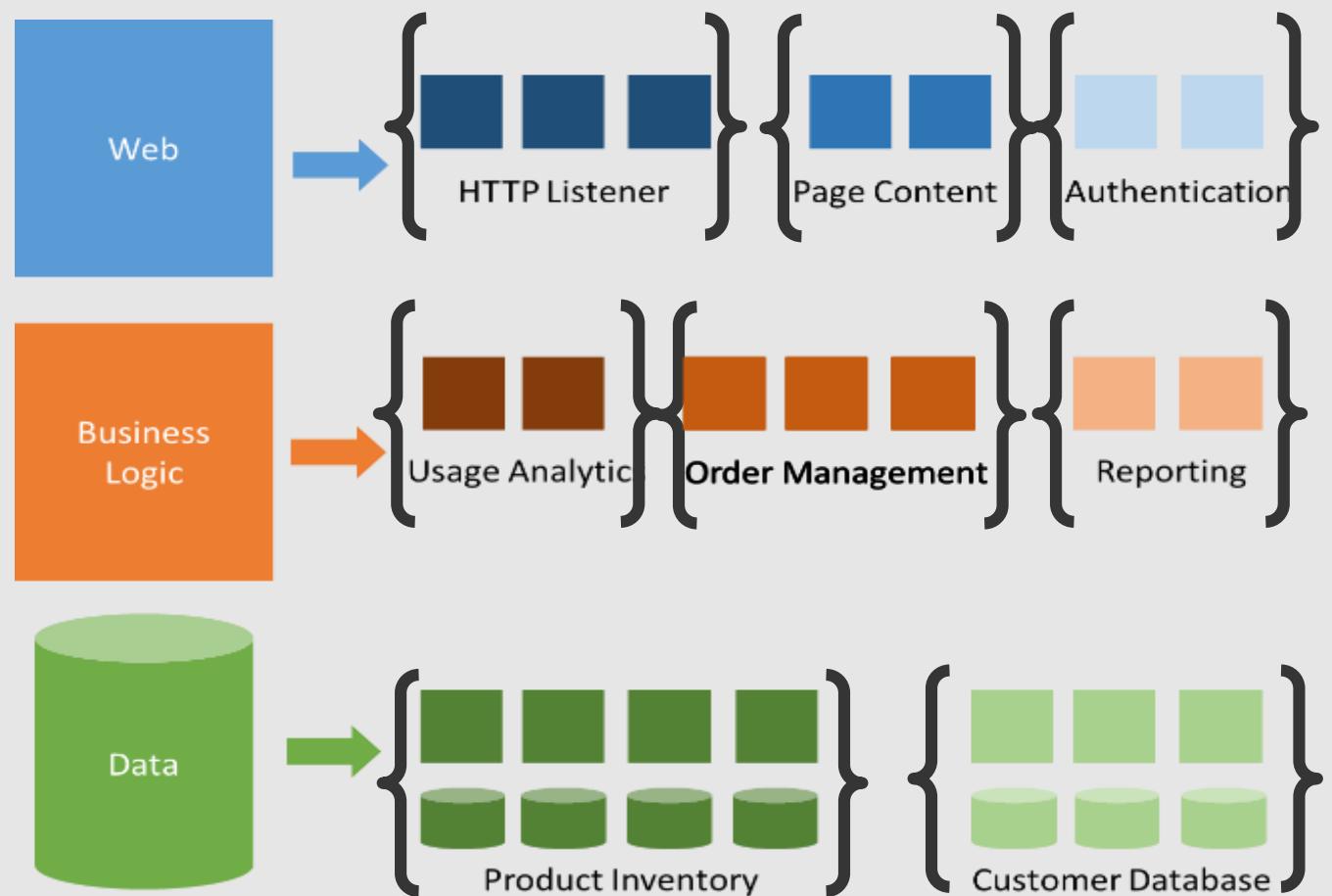
Microservices application approach

- A microservice application separates functionality into separate smaller services.
- Scales out by deploying each service independently creating instances of these services across servers/VMs/containers



Modernization with microservices

- Individually built and deployed
- Small, independent services
- Integrate using published API
- Fine-grained, loosely coupled



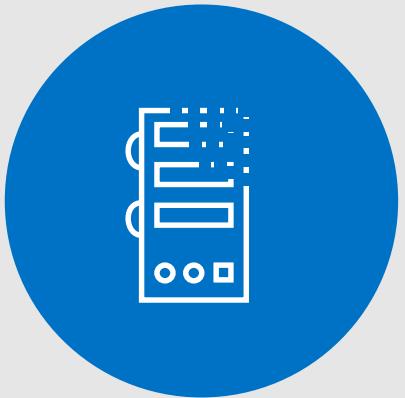
12-Factor Apps - <http://12factor.net>

1. Single root repo; don't share code with another app
2. Deploy dependent libs with app
3. No config in code; read from environment vars
4. Handle unresponsive app dependencies robustly
5. Strictly separate build, release, & run steps
 - Build: Builds a version of the code repo & gathers dependencies
 - Release: Combines build with config ReleaseId (immutable)
 - Run: Runs app in execution environment
6. App executes as 1+ stateless process & shares nothing
7. App listens on ports; avoid using (web) host
8. Use processes for isolation; multiple for concurrency
9. Processes can crash/be killed quickly & start fast
10. Keep dev, staging, & prod environments similar
11. Log to stdout (dev=console; prod=file & archived)
12. Deploy & run admin tasks (scripts) as processes

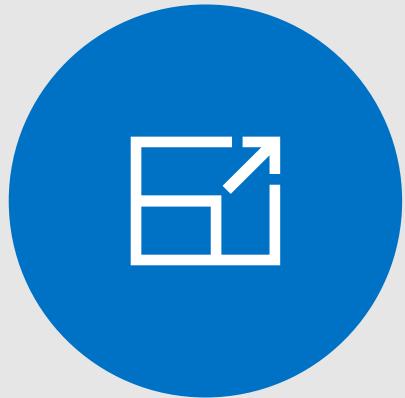


Serverless

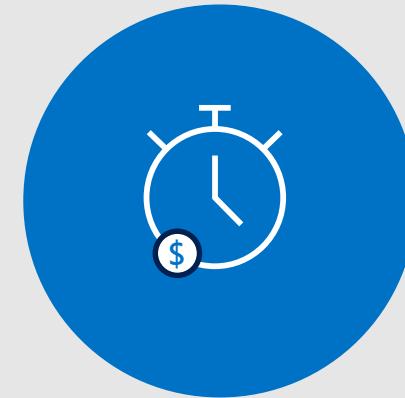
What is Serverless?



Abstraction
of servers



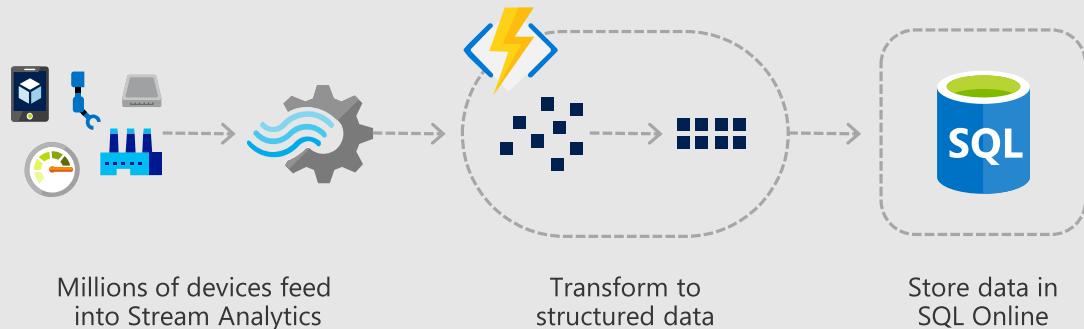
Event-driven
instant scale



Micro-billing

Functions to respond to events

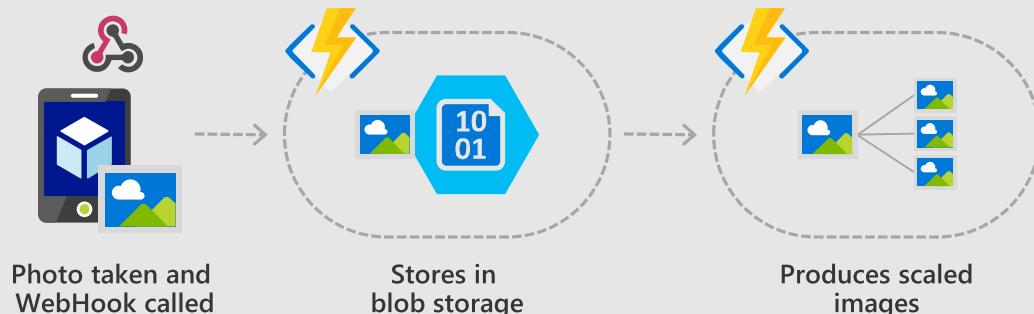
Real-time stream processing



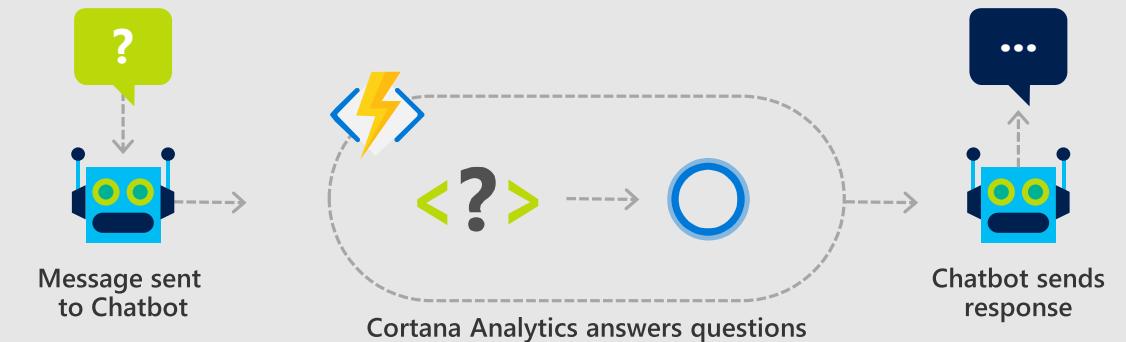
Timer-based processing



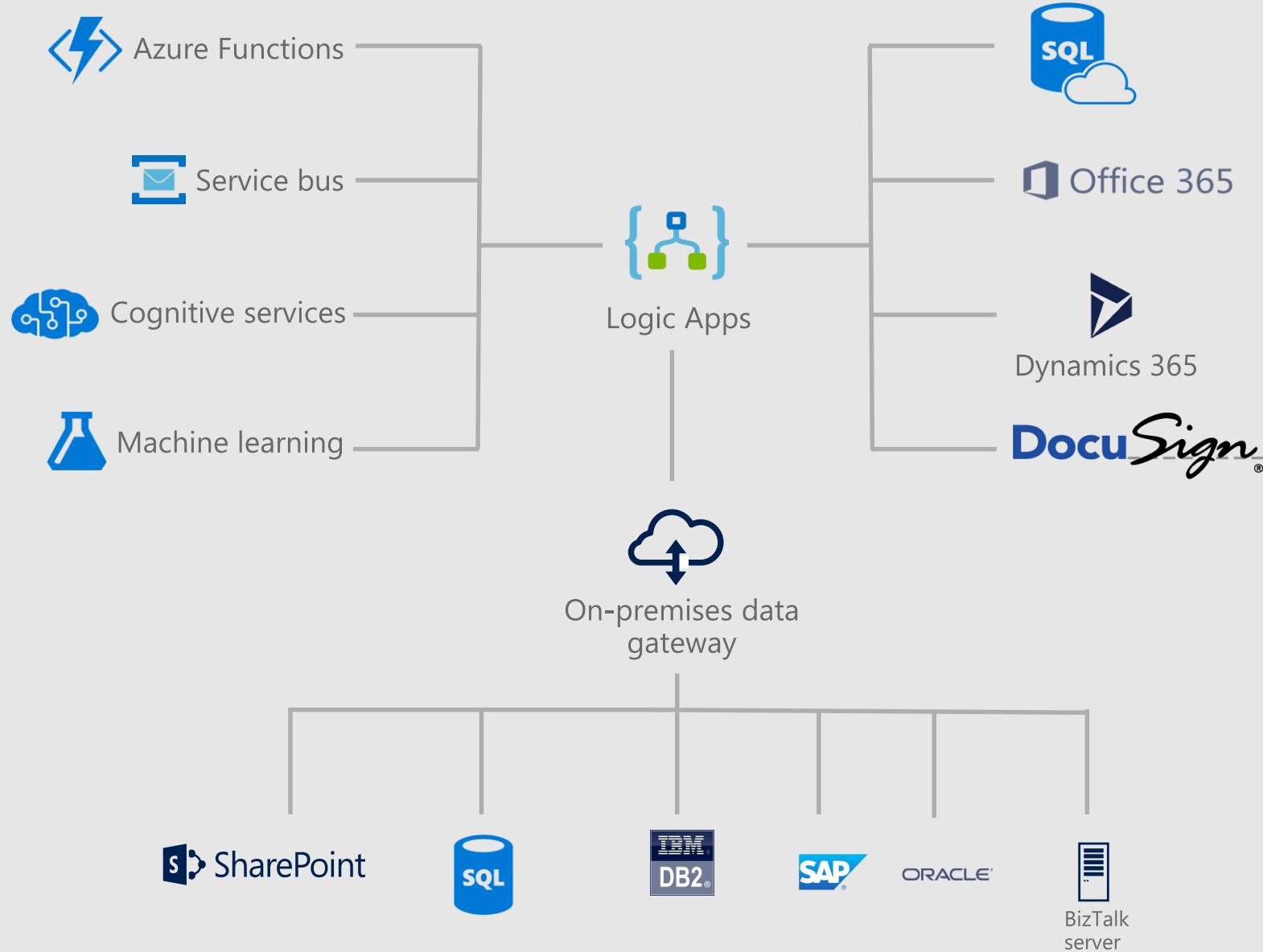
Mobile app backends



Real-time bot messaging



Logic apps connect everything

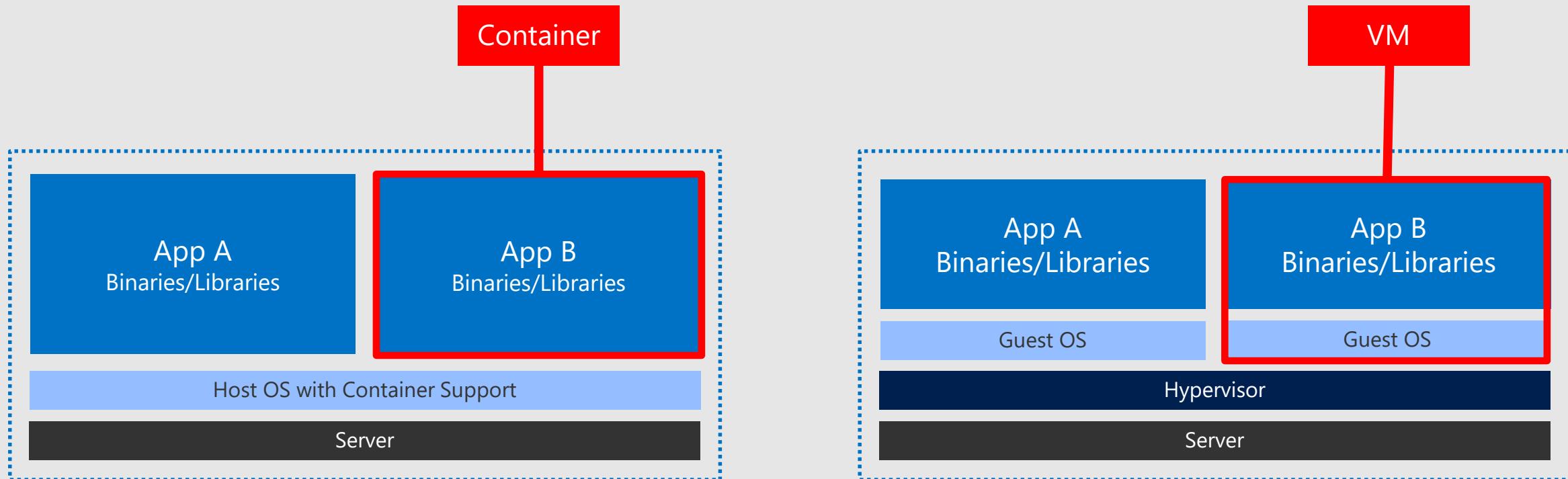


Containers

What is a container?

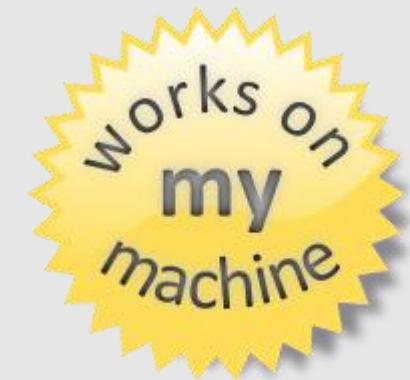
Next generation of virtualization:

An isolated environment to run your applications.

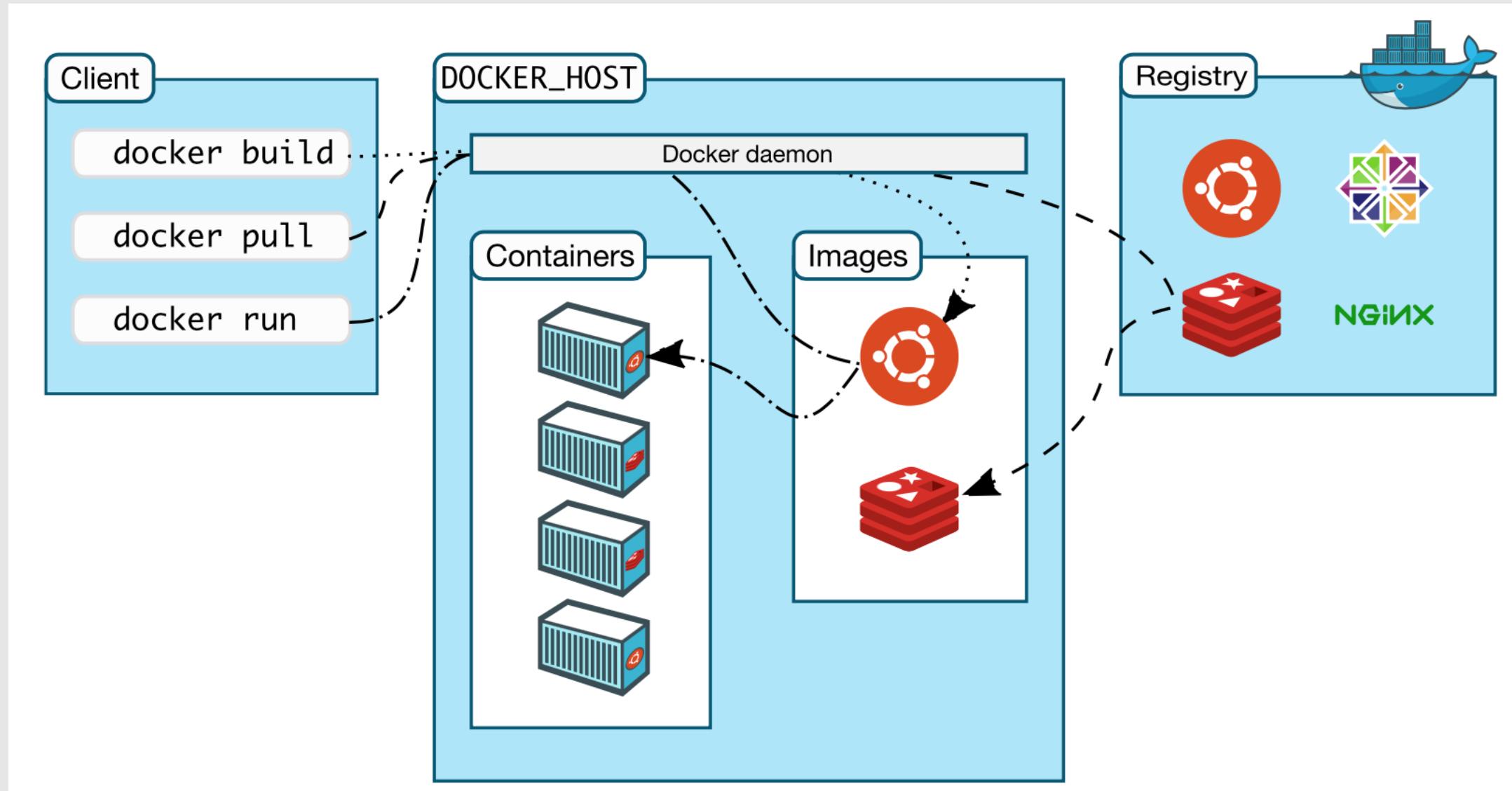


Why containers?

- Repeatable execution
 - immutable environment
 - reusable and portable code ("Build, Ship, and Run")
- Consistency across development, test, & production
- Fast & agile app deployment; instant startup
- Cloud portability
- Density, partitioning, scale
- Diverse developer framework support
- Microservices



Docker workflow



Orchestrators: Container management at scale

| Cluster Management Deploy and manage cluster resources | Scheduling When containers run | Lifecycle & Health Keep containers running despite failure | Naming & Discovery Where are my containers | Load Balancing Distribute traffic evenly |
|--|---|--|---|--|
| Scaling Make container sets elastic in number | Image Repository Centralized, secure container images | Continuous Delivery CI/CD pipeline and DevOps workflow | Logging & Monitoring Track events in containers and cluster | Storage Volumes Persistent data for containers |

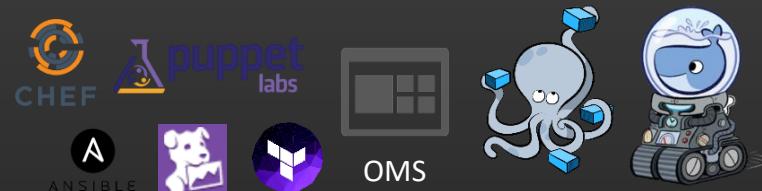
Azure Open Source Container Portfolio

Developer & Data Agility with Containers in the Cloud

Developers



Operations



Data scientists



Docker VM Extension for Azure

Easy and programmatic way to add Docker capabilities to your VMs



Open source container-based PaaS platforms in Azure

Container-ready application platforms that benefit from Azure's native partitioning, capacity management and high availability



Azure Container Service

Optimized container hosting in the cloud with familiar tooling and your choice of orchestrator



Azure Marketplace container partners

Partner solutions that address management challenges of containers



Microsoft Azure



Workload portability

Windows Server



Cross-cloud orchestration

Azure Stack



Tools integration

Your own platform



Azure Container support

Pivotal
Cloud Foundry



Mesos DC/OS



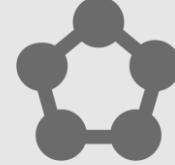
kubernetes



Azure Container Instance



Container Service



Service Fabric

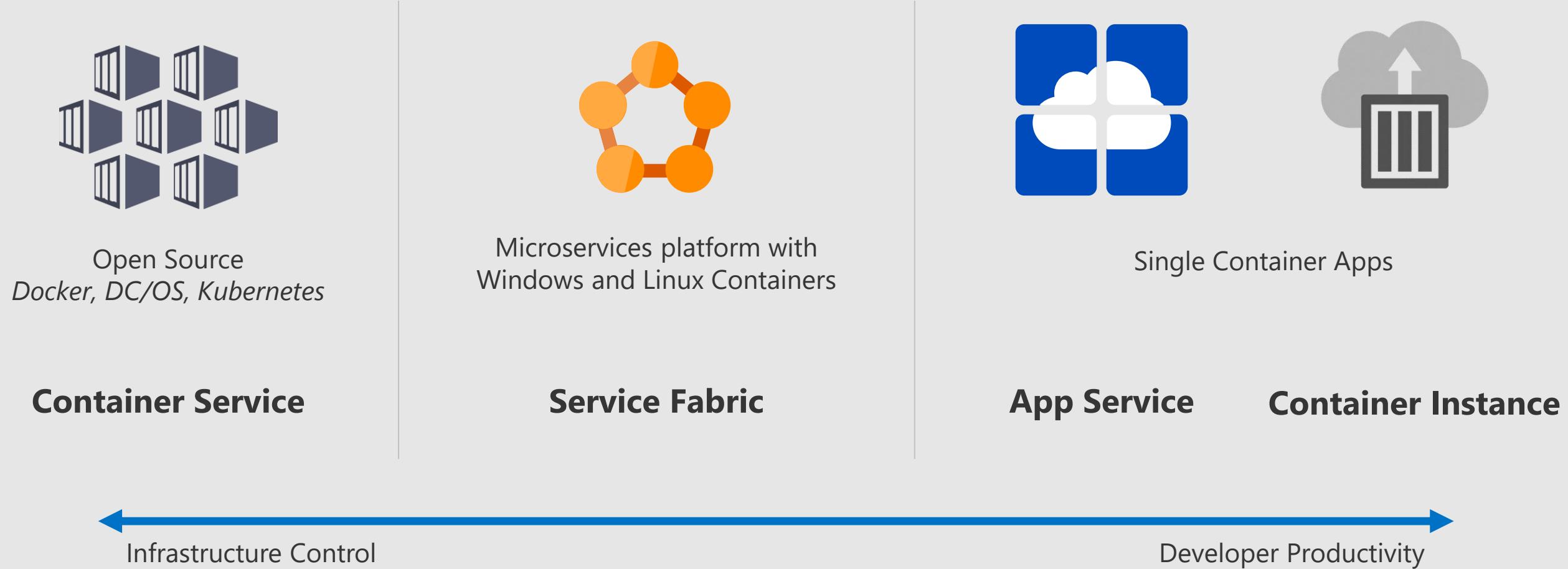


Web Apps



Batch

Azure Container technologies

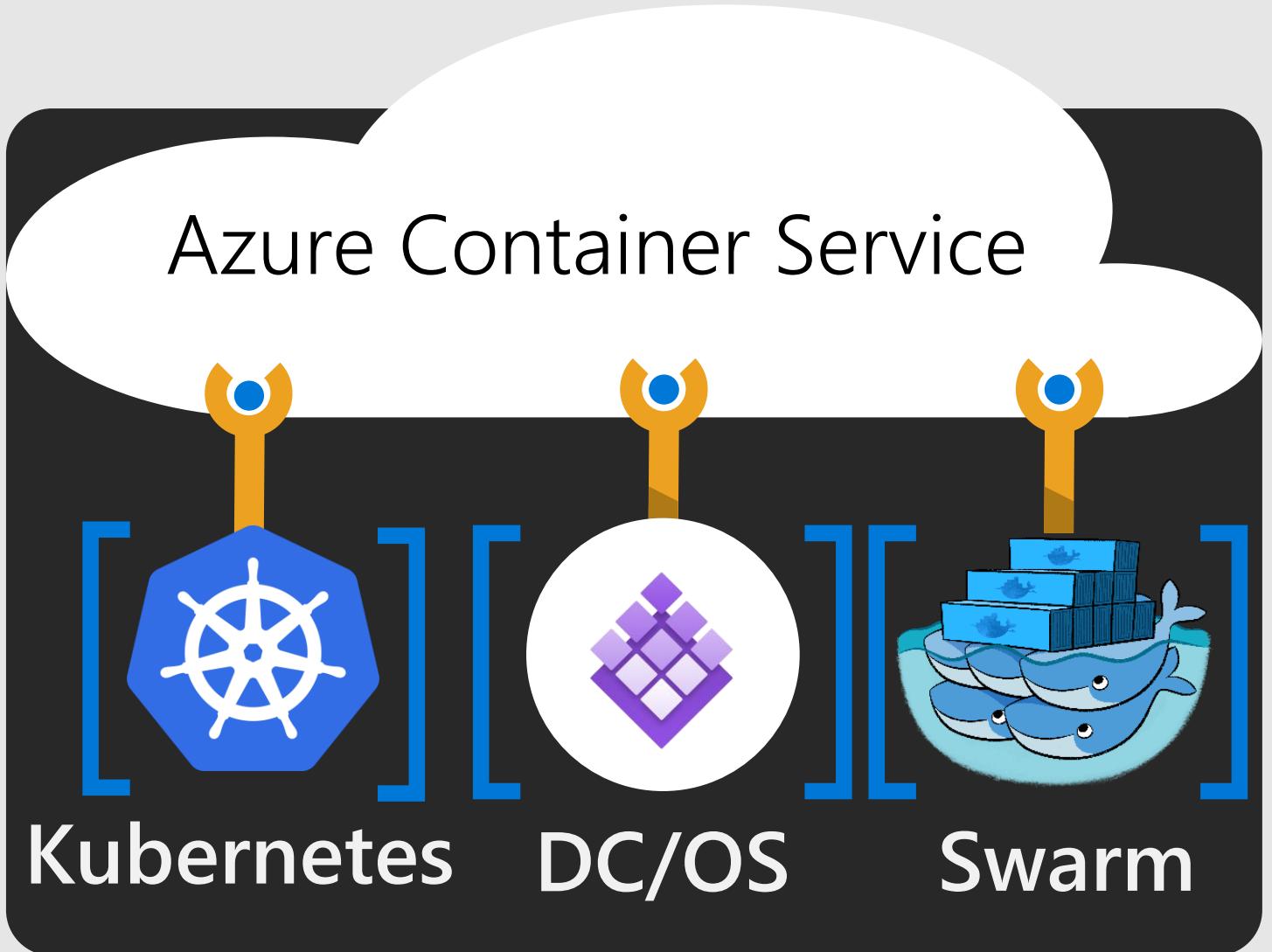


Azure Container Service

Deploy and manage containers using the tools you choose

Standard tooling and API support

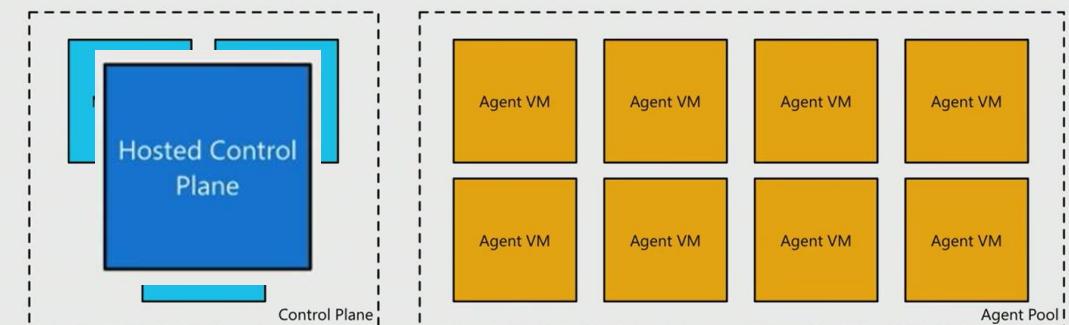
Streamlined provisioning of Mesosphere DC/OS, Docker Swarm, and Kubernetes



Managed Kubernetes for ACS

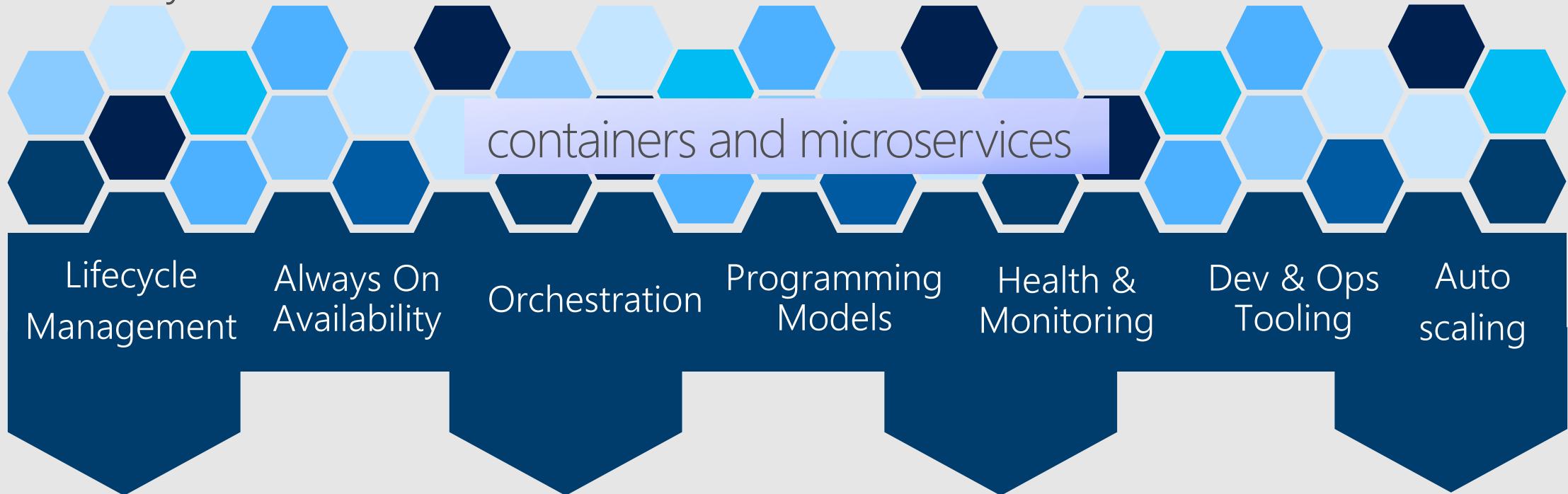
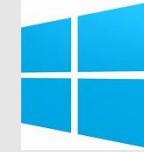
Why AKS?

- Easy to use
 - Fastest path to Kubernetes on Azure
 - Up and running with 3 simple commands
- Easy to manage
 - Automated upgrades and patching
 - Easily scale the cluster up and down
 - Self-healing control plane
- Uses Open APIs
 - 100% upstream Kubernetes



Azure Service Fabric

Any OS, Any Cloud



Dev Machine



Azure



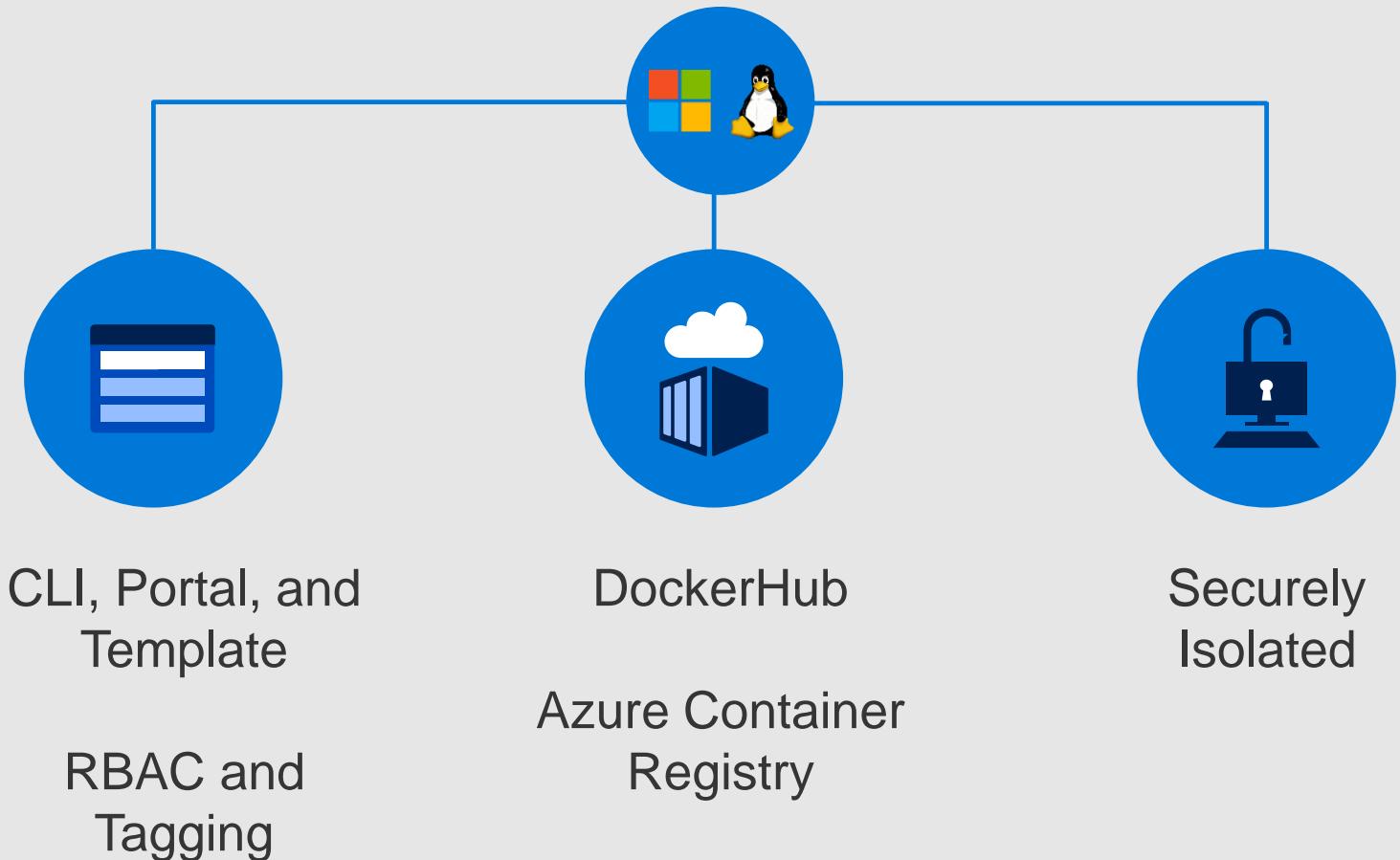
On Premise
Infrastructure



Other Clouds

Azure Container Instances (ACI)

- Fastest and easiest way to run a container in the cloud
- No VM management
- Per-second billing based on resource requirements (CPU + Memory)
- Deploy images from DockerHub or Azure Container Registry (ACR)



Azure App Services (Linux or for Containers)

- Can deploy a single custom image that can be scaled in the same way as all App Services
- Great for gaining production realism in your dev flow or gaining control of your environment

The screenshot shows the Azure portal interface for creating a new web application. The top navigation bar includes 'Home', 'Resource groups', 'cgidemo', 'Everything', 'Web App', 'Web App', and 'Docker Container'. The main window is titled 'Web App Create' and contains the following fields:

- * App name: Enter a name for your App (example: cgidemo.azurewebsites.net)
- * Subscription: Microsoft Azure Internal Consumption
- * Resource Group: Create new (selected) or Use existing (cgidemo)
- * OS: Windows or Linux (selected: Linux)
- * App Service plan/Location: ServicePlan005bc63b-bd4e(West...)
- * Configure container: node 6.11

To the right of this window is a 'Docker Container' configuration panel with the following sections:

- Docker Container** heading with a Docker logo.
- Web App for Containers** lets you bring your own Docker formatted container images and easily deploy and run them at scale with Azure.
- Info icon**: You can develop your app using Java, Python, and Go run-time stacks by deploying a custom container.
- Runtime Stack** dropdown menu:
 - Node.js 6.11 (selected)
 - Node.js 4.4
 - Node.js 4.5
 - Node.js 6.2
 - Node.js 6.6
 - Node.js 6.9
 - Node.js 6.10
 - Node.js 6.11
 - Node.js 8.0
 - Node.js 8.1
 - PHP
 - PHP 5.6
 - PHP 7.0
 - .Net Core
 - .Net Core 1.0
 - .Net Core 1.1
 - Ruby
 - Ruby 2.3

The bottom section of the portal window is titled 'Docker Container' and contains the following fields:

- Image source** dropdown: Azure Container Registry, Docker Hub, Private registry (selected: Azure Container Registry)
- Repository Access** dropdown: Public or Private (selected: Public)
- * Image and optional tag (eg 'image:tag'): A text input field.

Other initiatives around Containers

Azure Container Registry (ACR)

Manage images for all types of containers

Azure Container Registry allows you to store images for all types of container deployments including DC/OS, Docker Swarm, Kubernetes, and Azure services such as App Service, Batch, Service Fabric, and others. Your DevOps team can manage the configuration of apps isolated from the configuration of the hosting environment.



Manage your storage account caching

Better meet your throughput demand and API calls by having Container Registry manage the caching of your storage accounts. Choose between Basic, Standard, or Premium SKU options, and easily move between them to meet your business needs—from Basic for limited deployments to Premium for complex deployments and higher throughput for replicating images across multiple storage accounts.

Keep container images close

Reduce network latency and eliminate ingress/egress charges by keeping your Docker registry in the same data center as your deployments. Azure Container Registry gives you local, network-close storage of your container images within your subscriptions, and full control over access and image names.



Use familiar, open-source Docker CLI tools

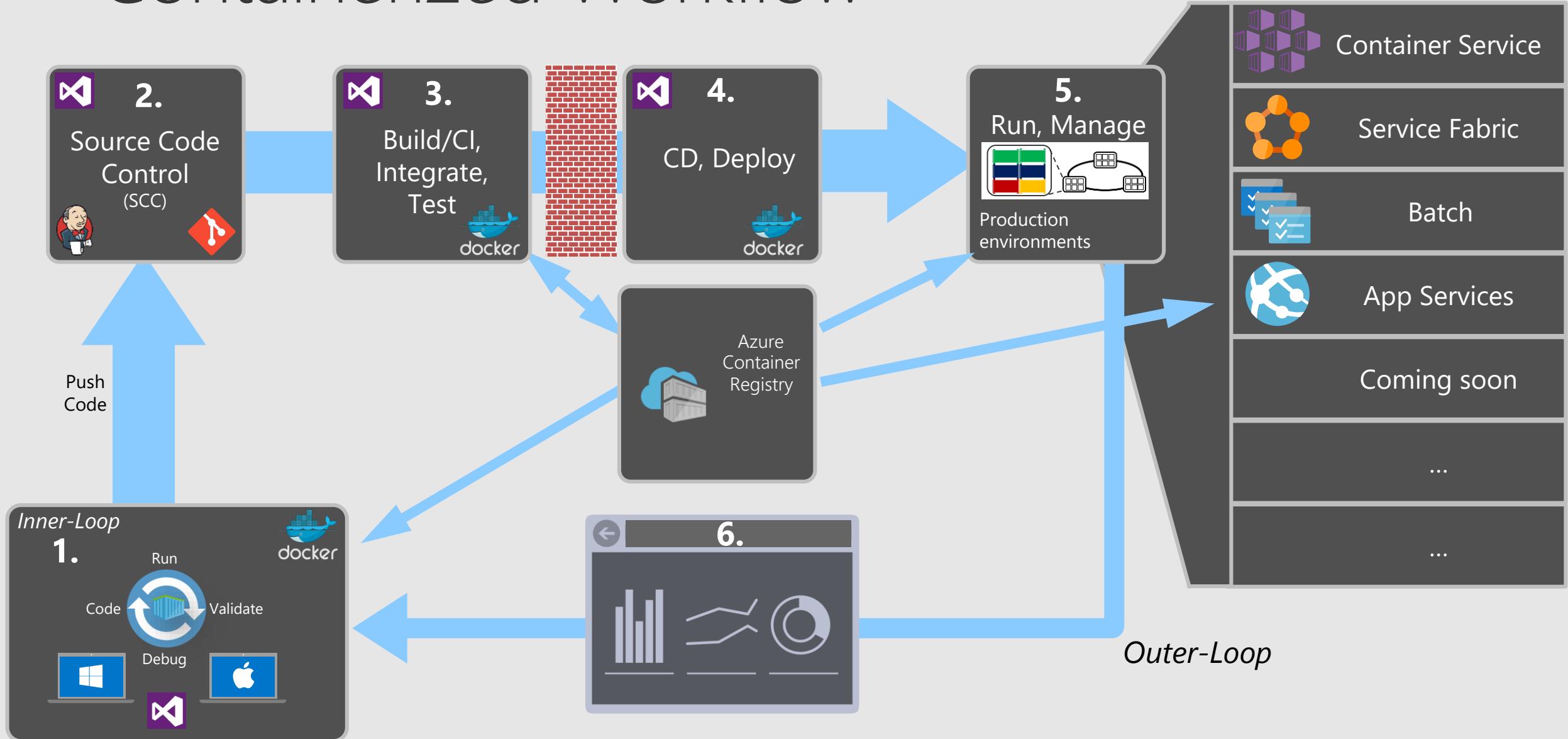
You don't have to learn new APIs or commands. Because Azure Container Registry is compatible with the open-source Docker Registry v2, you can use the same open-source Docker CLI tools you already know and the skills you have to efficiently interact with the registry.

Expand registry functionality

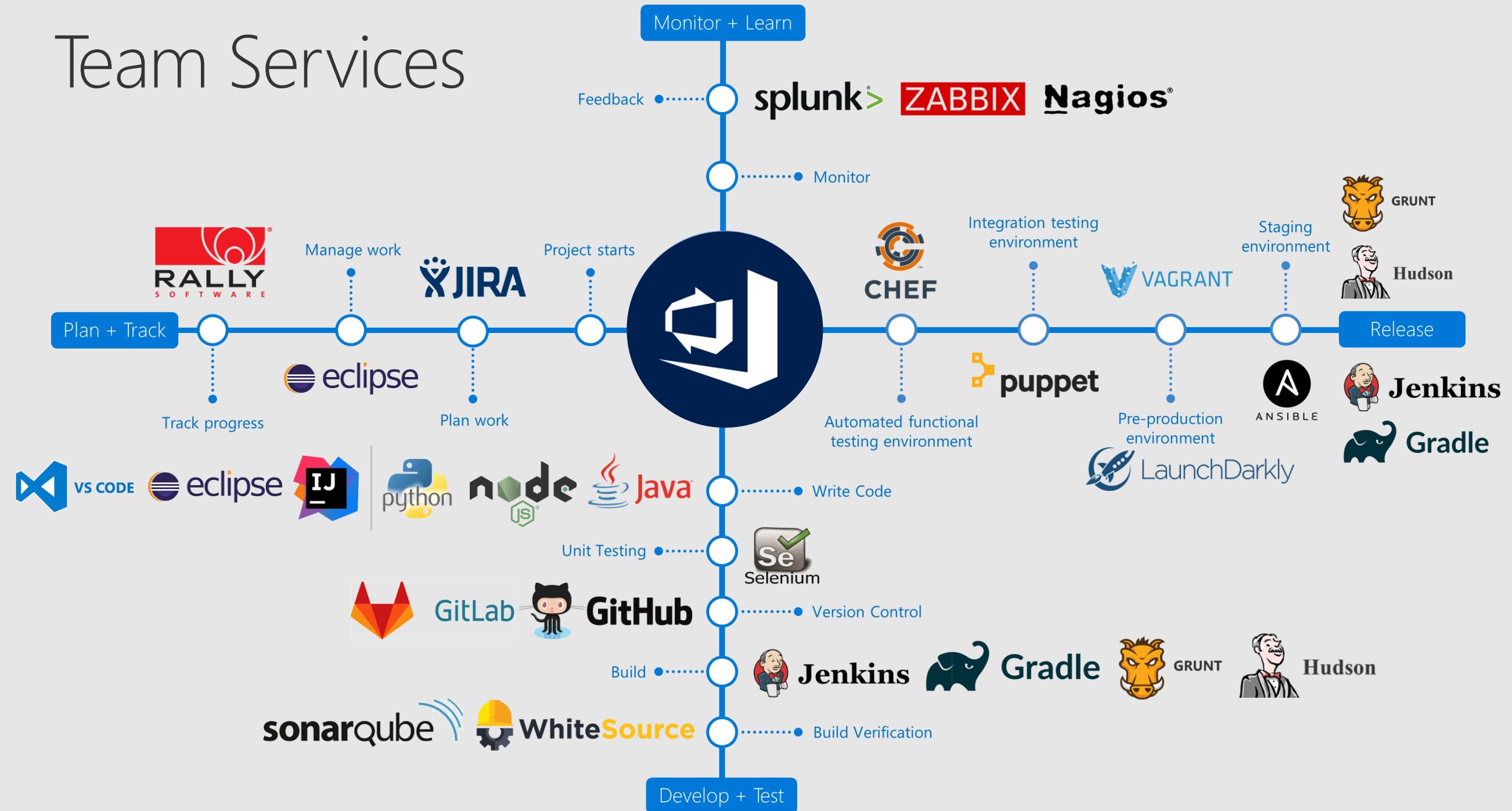
Keep your images safe by authenticating and managing access with Azure Active Directory. Trigger events based on container actions with Webhooks. Both technologies are compatible with standard registries.



Containerized Workflow



Team Services



VSTS - Agent on Docker

hub.docker.com/r/microsoft/vsts-agent

hub.docker.com

Search

Explore Help Sign up Sign In

PUBLIC REPOSITORY

microsoft/vsts-agent ☆

Last pushed: 10 days ago

Repo Info Tags

Short Description

Official images for the Visual Studio Team Services (VSTS) agent.

Full Description



Visual Studio Team Services agent

This repository contains images for the Visual Studio Team Services (VSTS) agent that runs tasks as part of a build or release.

Docker Pull Command

```
docker pull microsoft/vsts-agent
```

Owner



microsoft

VSTS – Docker Extension

marketplace.visualstudio.com/items?itemName=ms-vscs-rm.docker

Visual Studio | Marketplace Sign in

Visual Studio Team Services > Build and release > Docker Integration

 Docker Integration Microsoft | 6,274 installs | ★★★★☆ (41)

Build, push, run or deploy Docker images and multi-container Docker applications.

[Install](#) [Download](#)

Click **Install** for Visual Studio Team Services and **Download** for Team Foundation Server.

[Overview](#) [Q & A](#) [Rating & Review](#)

The world's largest enterprises rely on Docker to develop the world's best applications. With the Docker extension, you can integrate Docker images and containers into your existing agile and DevOps workflows.

NOTE: the Docker and Docker Compose tasks in this extension have been deprecated in favor of equivalent built-in VSTS tasks. If you are looking to use only one or both of these tasks and not the Docker Deploy task, please do not install this extension and instead use the built-in VSTS tasks.

The Docker extension adds a task that enables you to build Docker images, push Docker images to an authenticated Docker registry, run Docker images or execute other operations offered by the Docker CLI. It also adds a Docker Compose task that enables you to build, push and run multi-container Docker applications or execute other operations offered by the Docker Compose CLI. Lastly, it offers a Docker Deploy task that enables you to deploy single Docker images or multi-container Docker applications into Azure.

The Docker extension introduces two new service endpoints for Docker hosts and registries. The tasks default to using a local Docker host if available (this currently requires a custom VSTS agent), otherwise they require a Docker host connection to be provided. Actions that depend on being authenticated with a Docker registry, such as pushing an image, require a Docker registry connection to be provided.

Categories

Build and release

Tags

Build task Deploy task docker docker-compose

Works with

Visual Studio Team Services
Team Foundation Server

More Info

Version 0.4.0
Last updated 6/22/2017 1:32:11 PM
Publisher Microsoft
Report [Report Abuse](#)

Image Registries

Docker Hub and Docker Store

Public, Official and Private image repositories

Granular access controls with organization support

Automated image build support

Docker Trusted Registry

Enterprise Grade Private Registries

Runs on your infrastructure (on-prem or cloud)

Active Directory and Role Based Access Controls

Azure Container Registry

Store and manage container images across Azure deployments

Maintain Windows and Linux container images

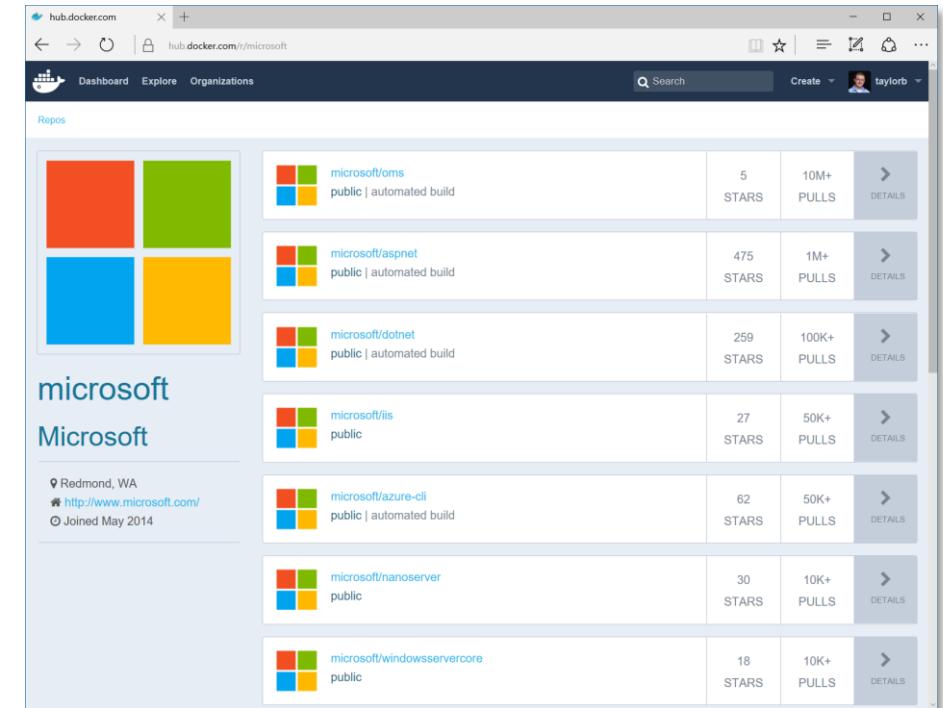
Same API and Tools as Docker Hub/Store/Registry

Docker Registry

Open source foundation of Hub and DTR

Runs on your infrastructure (on-prem or cloud) as a container

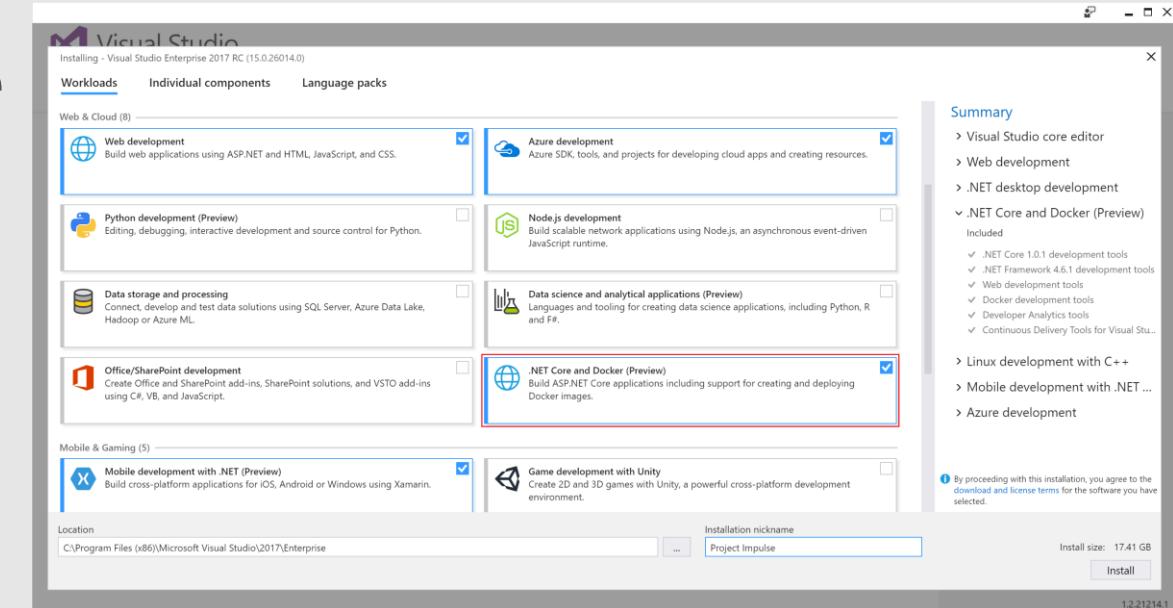
<https://docs.docker.com/registry> and or <https://github.com/docker/distribution>





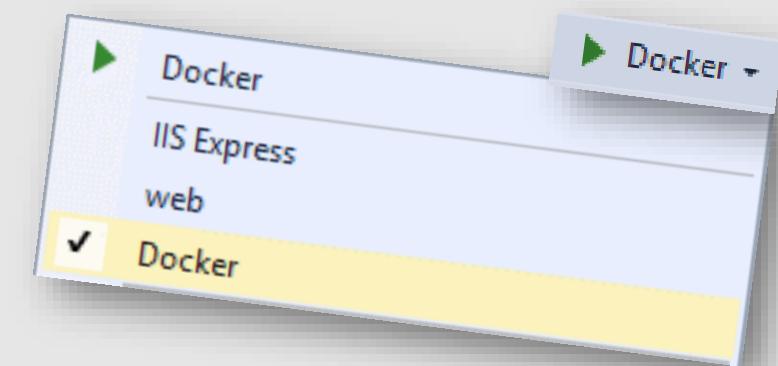
Visual Studio Docker Tools

- Run, Debug, Test Web & Console apps in docker containers
 - *Linux today, Windows Server & Nano Server coming soon*
- F5 Debugging
- Edit & Refresh of code
- Scaffolds docker assets
 - Dockerfile, docker-compose.yml



```
Dockerfile
FROM microsoft/aspnetcore:1.0.1
ARG source=.
WORKDIR /app
EXPOSE 80
COPY $source .
ENTRYPOINT ["dotnet", "Web.dll"]
```

```
Dockerfile
FROM microsoft/dotnet:1.0.1-core
ARG source=.
WORKDIR /app
COPY $source .
ENTRYPOINT ["dotnet", "BatchJob.dll"]
```



Azure Cloud Shell is hosted on Container

The screenshot shows the Microsoft Azure Cloud Shell interface. At the top, there's a browser-like header with tabs for 'My Dashboard - Micros' and a search bar for 'ms.portal.azure.com/#'. Below the header is the Microsoft Azure navigation bar with links for 'Report a bug', 'Search resources, services and docs', and notifications. On the left, there's a sidebar with options like 'New', 'Dashboard', 'All resources', and 'More services'. The main area displays a 'My Dashboard' card with sections for 'Build (minutes)', 'Runs Failed and Succeeded', and 'Billable Executions'. The 'Build (minutes)' section shows 'AccountBuildUsageHistoryPart' with values 2 and 1.5. The 'Runs Failed and Succeeded' section shows 'MYTWITTERDASHBOARD' with values 2 and 1.5. The 'Billable Executions' section shows 'MYTWITTERDASHBOARD' with values 14 and 12. At the bottom, the Azure Cloud Shell terminal window is open, showing a 'Bash' prompt. It displays a success message: 'Cloud Shell.Succeeded.' followed by 'terminal...'. Below the terminal, there's a welcome message: 'Welcome to Azure Cloud Shell (Preview)'. It also provides instructions: 'Type "az" to use Azure CLI 2.0' and 'Type "help" to learn about Cloud Shell'. A command line prompt 'mathieu@Azure:~\$' is shown at the bottom.

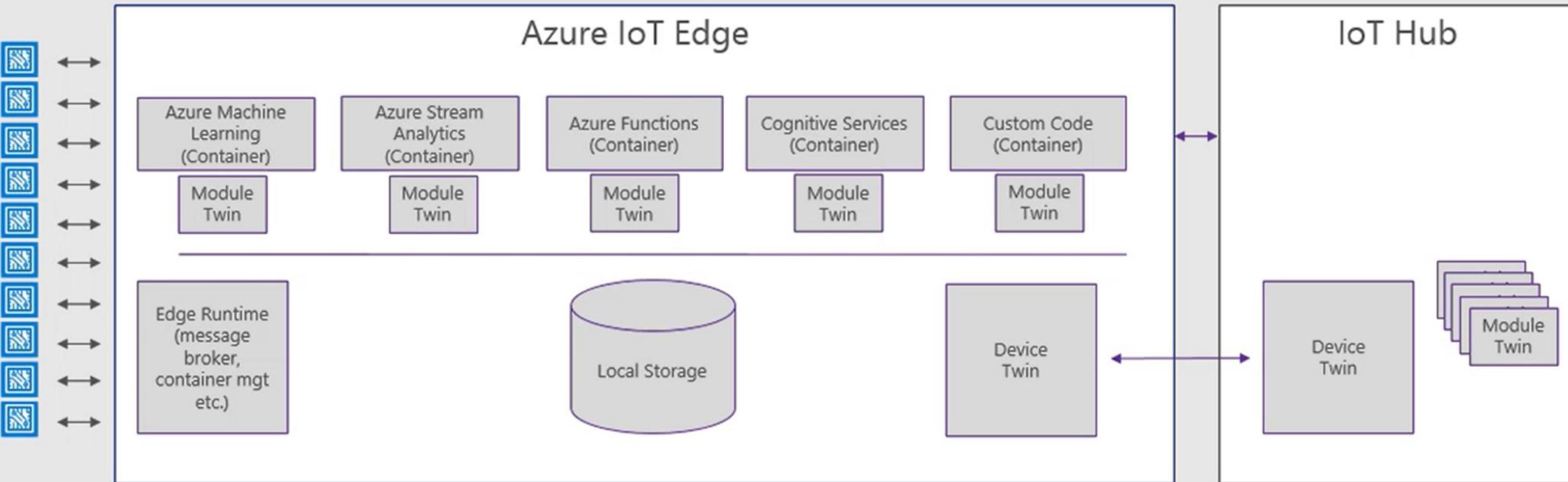
```
Bash
Cloud Shell.Succeeded.
terminal...

Welcome to Azure Cloud Shell (Preview)

Type "az" to use Azure CLI 2.0
Type "help" to learn about Cloud Shell

mathieu@Azure:~$
```

IoT Edge



Azure Machine Learning

AI-powered Data Wrangling + E2E ML Dev Productivity + Deploy Anywhere = **E2E Tooling for AI Development**

Program Synthesis

SPARK, GPU, Open Source
Lifecycle Management

Docker, Spark, IOT Edge,
On prem, AWS/GCP...

Service Fabric – Case Studies

Azure Core Infrastructure

thousands of machines

Azure Document DB

billions transactions /week

Power BI

Azure SQL Database

1.4 million databases

Intune

800k devices

Skype for Business

Hybrid Ops

Event Hubs

20bn events/day

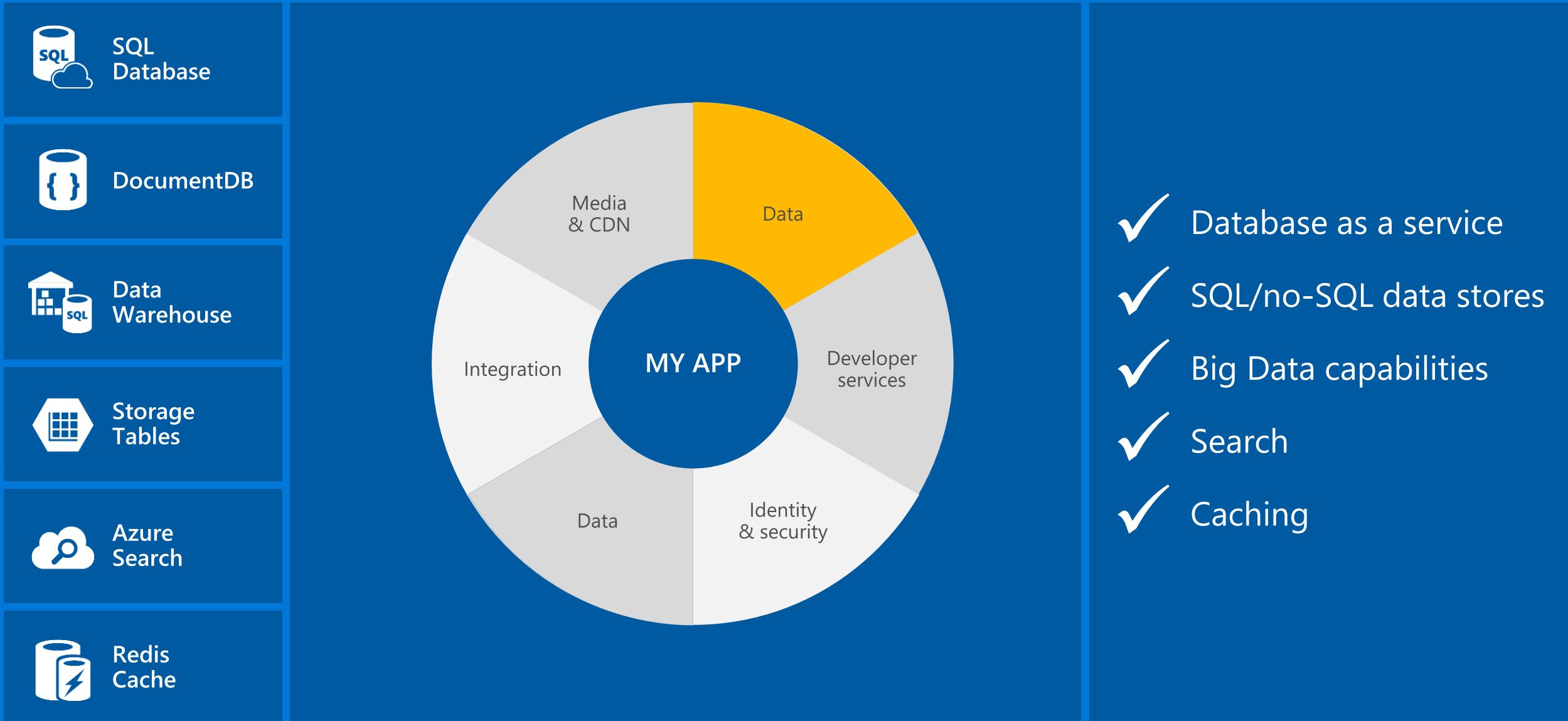
Bing Cortana

500m evals/sec

IoT Suite

...

Where could reside your data?

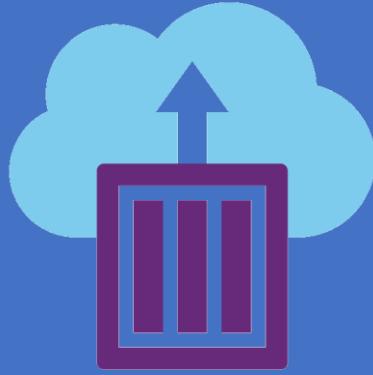


Conclusion & Resources

Serverless or Containers?

- Serverless works best at removing barriers in developing cloud-scale solutions
- Containers work best at giving consistent environments of control across any platform
- Rule of thumb: start as high on the stack as possible. If it doesn't meet your needs, move down to the next level
- SaaS vs. FaaS vs. PaaS vs. IaaS

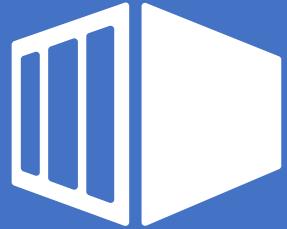
Deploy Containers Everywhere in Azure



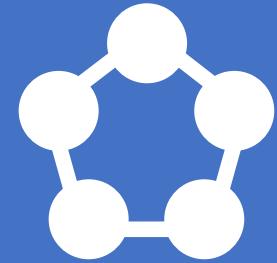
Azure Container
Instance



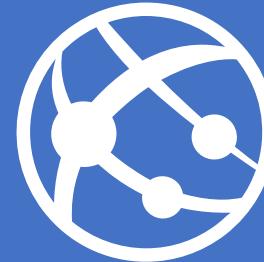
Kubernetes



Container Service



Service Fabric



Web Apps



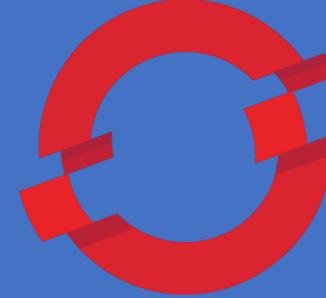
Batch



Mesos DC/OS



Pivotal



OpenShift



Docker Enterprise

Azure Container 101

Here's what service you should recommend ...

If your customer is looking for this ...

... use this

Scale and orchestrate containers using Kubernetes, DC/OS or Docker Swarm

[Container Service](#)

Easily run containers on Azure with a single command

[Container Instances](#)

Store and manage container images across all types of Azure deployments

[Container Registry](#)

Develop microservices and orchestrate containers on Windows or Linux

[Service Fabric](#)

Deploy web applications on Linux using containers

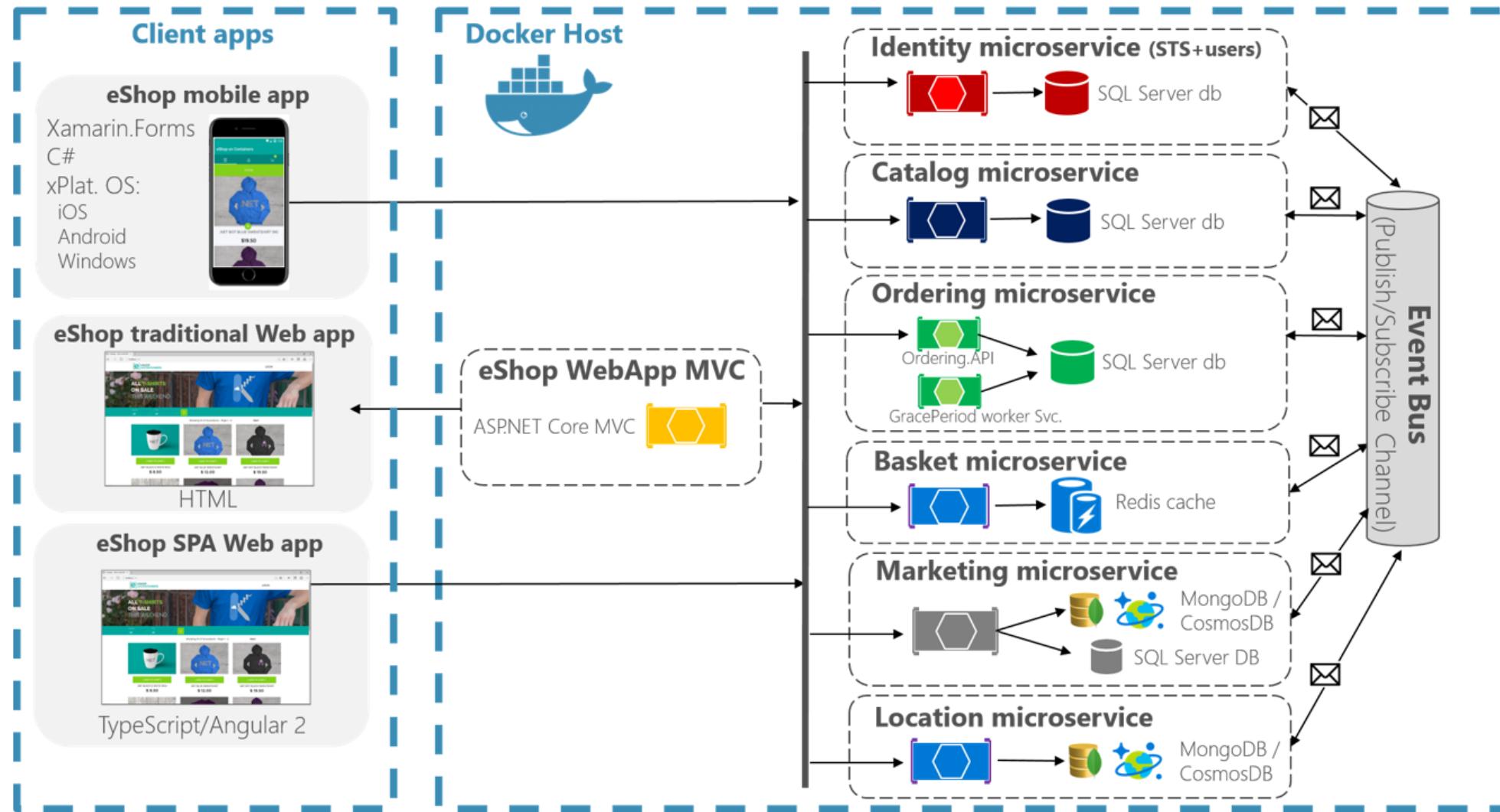
[App Service](#)

Run repetitive compute jobs using containers

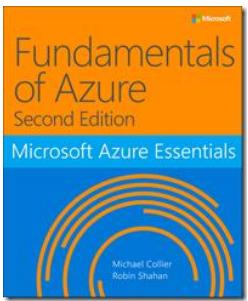
[Batch](#)

.NET Core Microservices Reference App

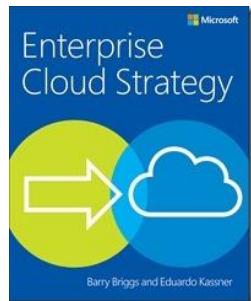
eShopOnContainers Reference Application - Architecture



Free e-books



Fundamentals



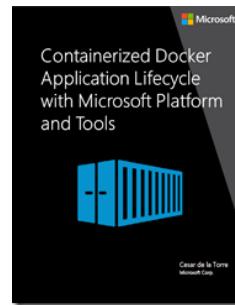
Enterprise Cloud
Strategy



Cloud Design
Patterns



.NET Microservices:
Architecture for
Containerized .NET Apps



Containerized Docker
Application Lifecycle



Azure Developer
Guide

<https://github.com/OSSCanada/microhackfest>

OSSCanada / microhackfest

Watch 3 Star 3 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

Hands-on Labs (HOLs) and presentations for Microservices, Serverless and Containers readiness.

59 commits 1 branch 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

Kevin Harris Merge branch 'master' of https://github.com/OSSCanada/microhackfest Latest commit 997e252 43 minutes ago

HOL Merge branch 'master' of https://github.com/OSSCanada/microhackfest 43 minutes ago

Presentations Added presentation. 6 days ago

AGENDA.md Updated agenda and added link to Web App for Containers for git deplo... 5 hours ago

README.md updated image for great succes 3 hours ago

README.md

Microservices, Serverless and Containers Hackfest

This hands on hackfest/workshop is designed to give the the audience an overview of what a microservices architecture is and how to accomplish your goals through various PaaS services on [Microsoft Azure](#).

This repository contains the presentation and hands-on labs (HOLs) for Microservices, Serverless and Containers readiness.

Get involved with your local communities

- Docker Meetup Quebec
- September 2017 – Docker and Azure + Lessons learned with Docker on Azure IaaS
 - CGI + PetalMD
- March 2017 – Docker and Microsoft Azure
 - Savoir Faire Linux + Microsoft
- Kubernetes Meetup
- Azure Meetup Quebec
- September 2017 – Azure WebJobs + Azure and OSS
 - Cofomo + CGI
- June 2017 - Global DevOps Bootcamp
 - Cofomo (+ CGI)
- May 2017 - DevOps with Docker on Azure
 - CGI



Azure OpenDev

| | | |
|-------------|--|--|
| 9:00 AM PT | Contributing to open source (it's not just about code) + More |  Ashley McNamara Principal Developer Advocate, Microsoft Twitter GitHub |
| 9:20 AM PT | Getting started with InnerSource—open source workflows in the enterprise + More |  Ryan Parks Solutions Engineer, GitHub Twitter GitHub |
| 9:50 AM PT | Continuous delivery of infra to Azure + More |  Tyler Croy Director of Evangelism and Jenkins project board member, CloudBees Twitter GitHub |
| 10:20 AM PT | Modernize your Java development workflow with Habitat + More |  Matt Wrock Software Developer, Chef Twitter GitHub |
| 10:50 AM PT | Reproducible infrastructure with Terraform and Microsoft Azure + More |  Nic Jackson Developer Advocate, HashiCorp Twitter GitHub |
| 11:35 AM PT | Logging, security, and analytics on Azure with the Elastic Stack + More |  Christoph Wurm Principal Solutions Architect, Elastic Twitter GitHub |

Microsoft Azure - Architecture guidance

- Know the limits of each service
- Know the SLA of each service
- Know the price of each service
- Know the regions where you could host each service
- Know the compliances of the Azure platform
- Automate early and always with ARM Templates
- Get inspired from the Microsoft Technical Case Studies and make amazing architecture diagrams with these icons

Questions

Answers

