



Microsoft Developer School

30 листопада 2017 року, Київ

<https://aka.ms/msdevschoolnov>

#msdevschool

Building serverless apps

Anton Boyko

Microsoft Azure MVP

Solution Architect @ Ciklum

me@boykoant.pro

Agenda



Serverless architecture – how it's even possible to run server side code without a server?

What is Azure Logic Apps?

What is Azure Functions?

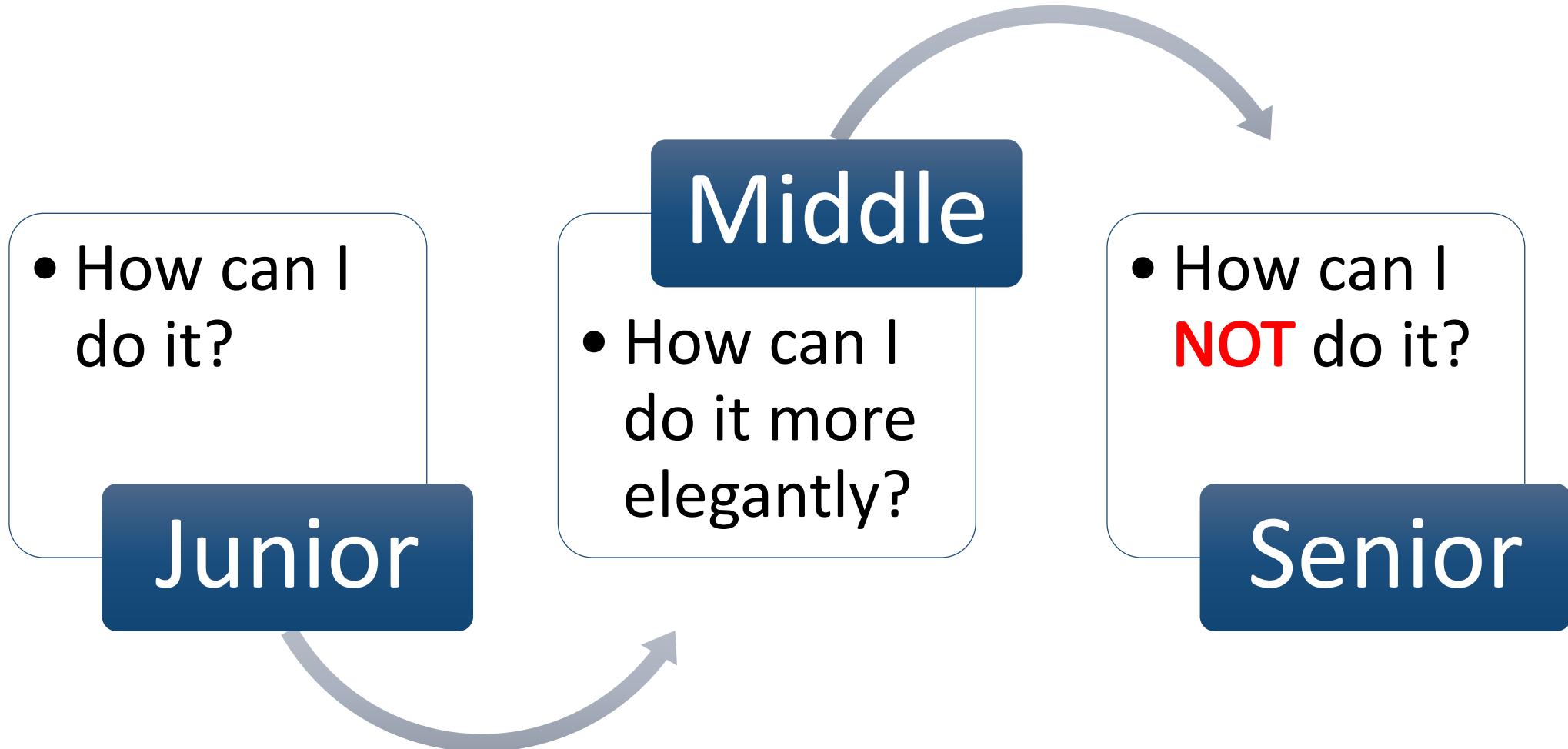
What is Azure Container Services?

Questions

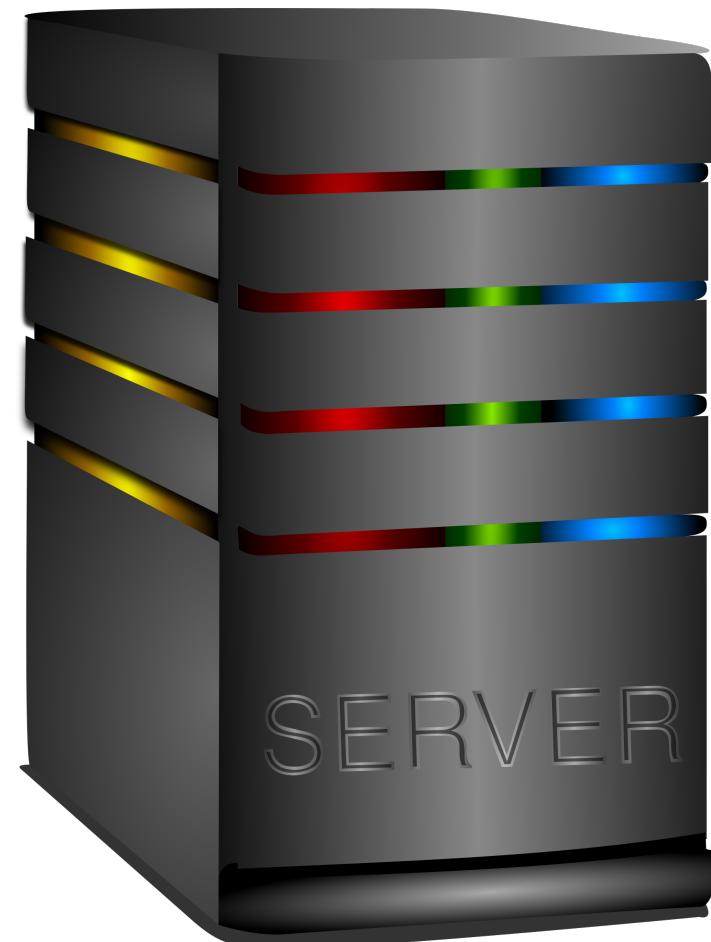
Others...

Serverless?

Developer maturity level



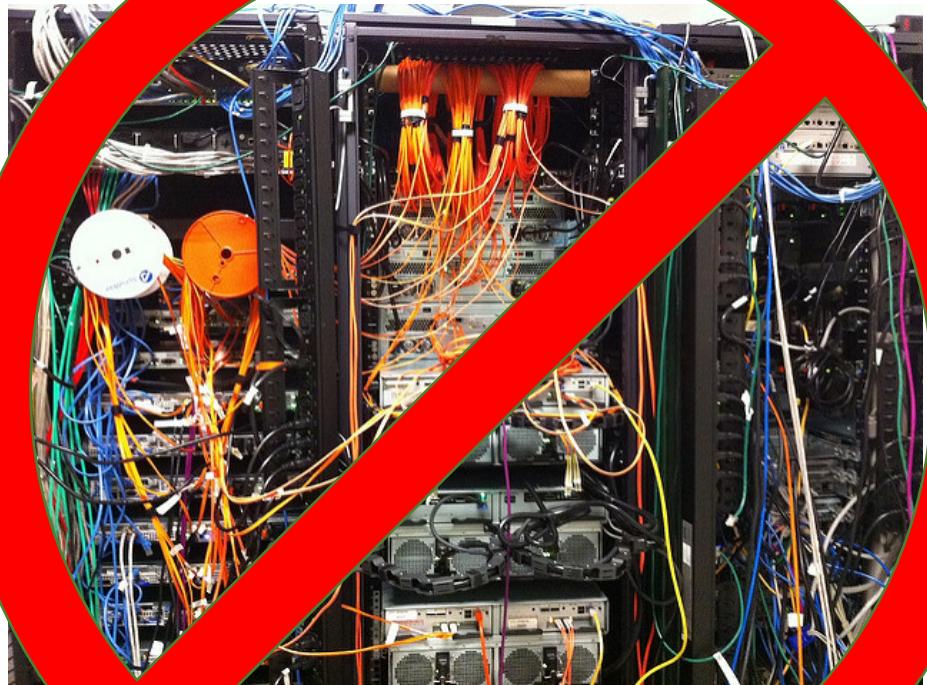
Traditional server



H16M Standard	
16	Cores
224	GB
 32	Data disks
 32x500	Max IOPS
 2000 GB	Local SSD
	Load balancing
2,231.26	
USD/MONTH (ESTIMATED)	

- Fixed reserved CPU
- Fixed reserved RAM
- Pay for reserved resourced

Serverless

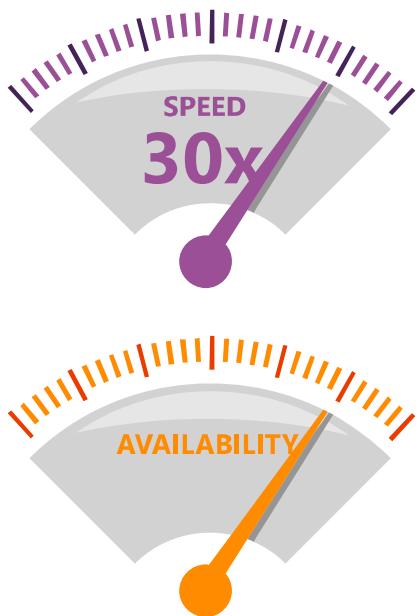


- On-demand CPU
(per second)
- On-demand RAM
(per Mb)
- Pay for number of executions
and on-demand resources
consumed

Serverless



Abstraction of
servers



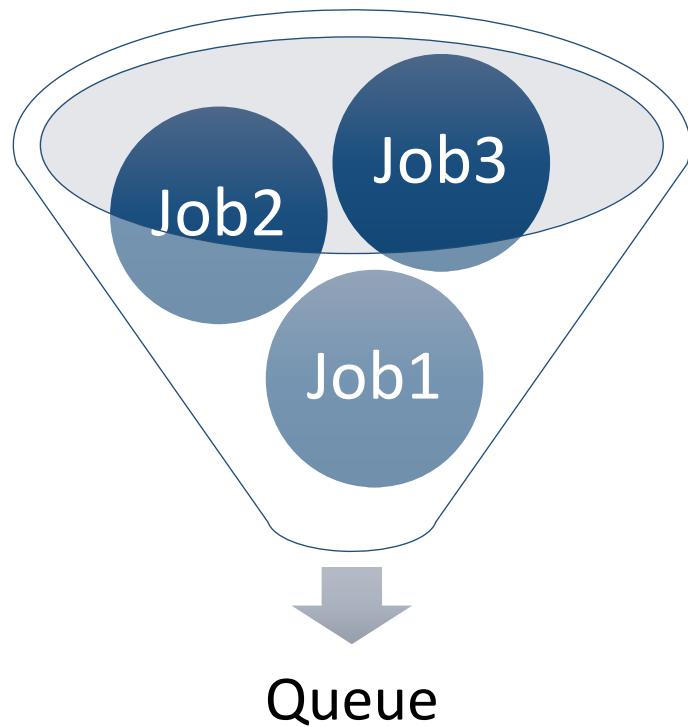
Event-driven
scale



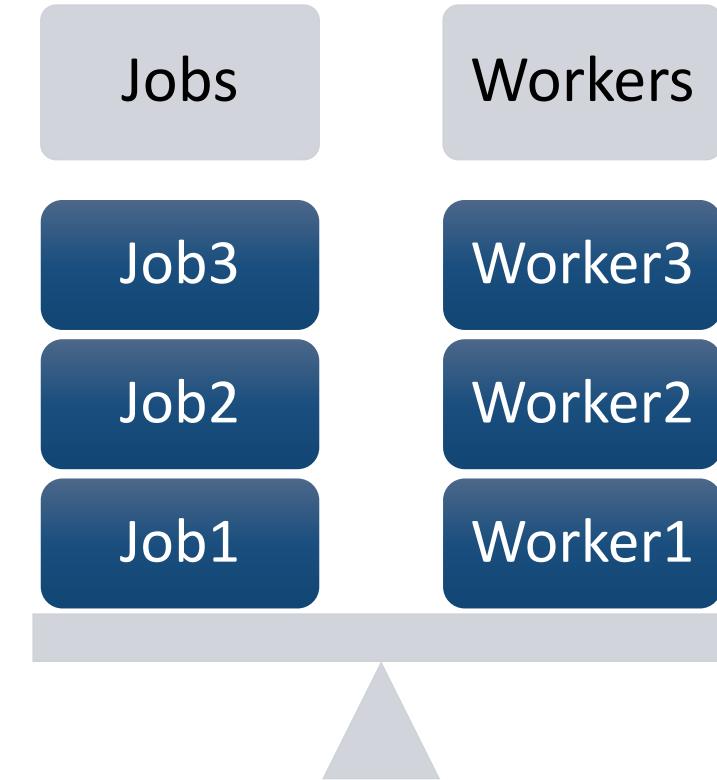
Sub-second
billing

Reserved vs On-demand

Reserved



On-demand



There is no silver bullet



Key attributes of serverless code

Stateless

- Subsequent executions can run on a different hosts

Small

- Start fast, finish fast

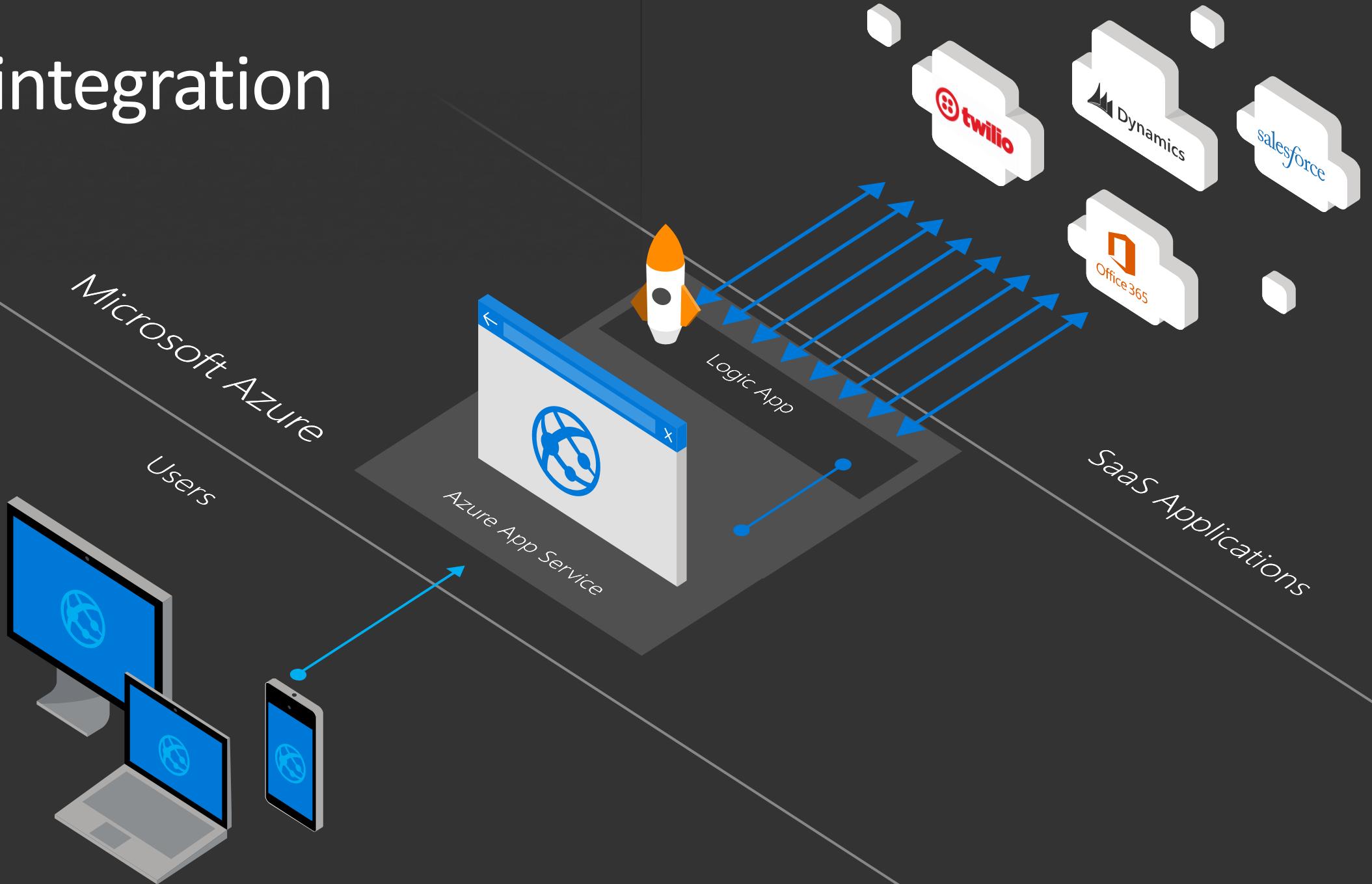
Single unit of work

- Easier to maintain



Azure Logic Apps

SaaS integration



Built-in API Connectors



Connectors

- Box
- Chatter
- Delay
- Dropbox
- Azure HD Insight
- Marketo
- Azure Media Services
- OneDrive
- SharePoint
- SQL Server
- Office 365
- Oracle
- QuickBooks
- SalesForce
- Sugar CRM
- SAP
- Azure Service Bus
- Azure Storage
- Timer / Recurrence
- Twilio
- Twitter
- IBM DB2
- Informix
- Websphere MQ
- Azure Web Jobs
- Yammer
- Dynamics CRM
- Dynamics AX
- Hybrid Connectivity



Protocols

- HTTP, HTTPS
- File
- Flat File
- FTP, SFTP
- POP3/IMAP
- SMTP
- SOAP + WCF

BizTalk

- Batching / Debatching
- Validate
- Extract (XPath)
- Transform (+Mapper)
- Convert (XML-JSON)
- Convert (XML-FF)
- X12
- EDIFACT
- AS2
- TPMOM
- Rules Engine



Target audience is *anyone* who can use Azure

... but not necessarily business users or consumers

New Logic Apps for easy automation



LOGIC APPS

Automate SaaS and
on-premises systems

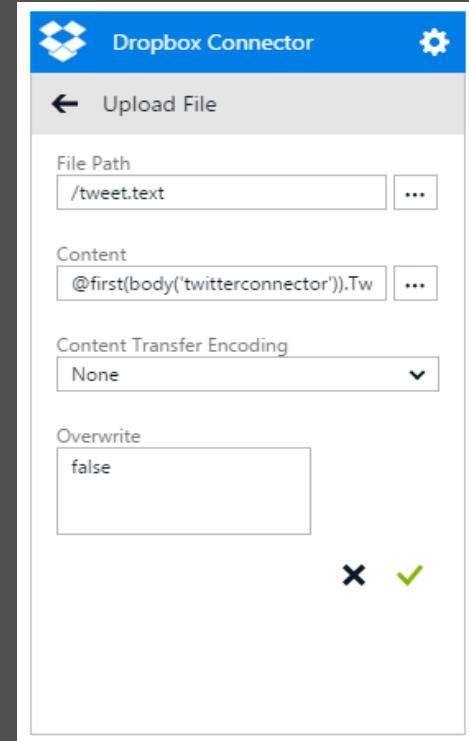
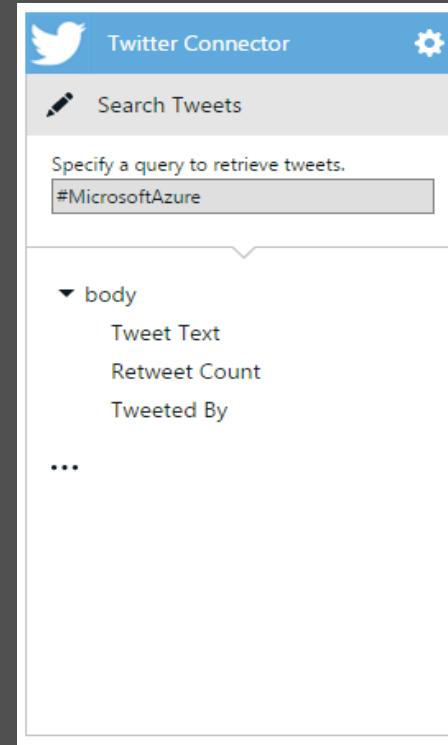
- No code designer for rapid creation
- Dozens of pre-built templates to get started
- Out of box support for popular SaaS and on-premises apps
- Use with custom API apps of your own
- Biztalk APIs for expert integration scenarios

All of the Azure-native capabilities you'd expect

- Full audit logs of all management operations
- Role-based access control
- Deployment lifecycle with Resource Manager
- Resource Management API + resource PowerShell
- On-prem support with release of Azure Stack

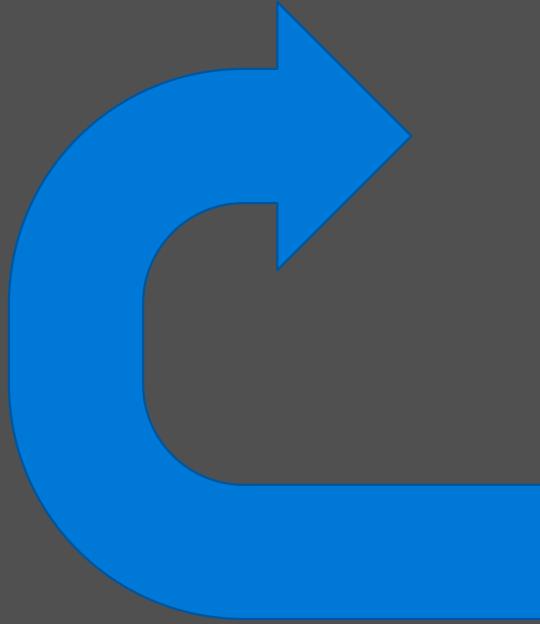
Logic apps for orchestration across API apps

1. Implicitly – whenever you reference the output of an action you'll depend on that action executing first
2. Explicit “dependsOn” condition – you can mark certain actions to run only after previous ones have completed
3. Explicit “expression” condition – a complex function that evaluates properties of other actions



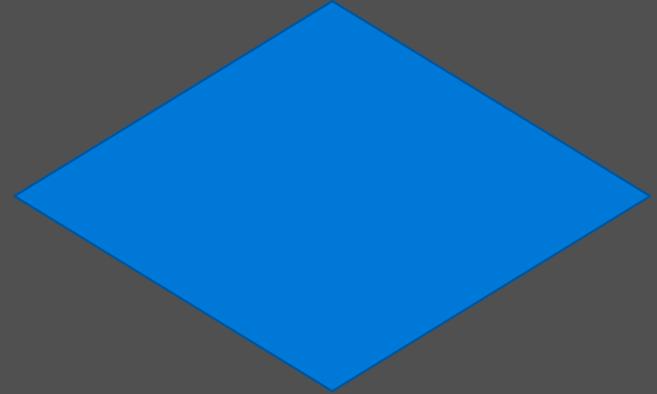
Repeating

- Loop a single action over a list of items
- Runs the action N times
- You can get at all of the statuses for each action
- Tip: when you have multiple collections use:
 - "repeat" :
"@range(0,length(body('connector1')))"
 - ...
 - "inputs" :
"@concat(body('connector1')[repeatItem()],
body('connector2')[repeatItem()])"



Conditionals

- Logic on a trigger or action
 - For triggers conditionals are post-conditions
 - For actions conditionals are pre-conditions
- You can do conditionals inside of repeats to perform as a filter



Same underlying engine...

RESOURCE MANAGER

- Can handle thousands of parallel deployments per stamp
- Resilient against failure – retries with “at least once” guarantee
- Simple, declarative JSON template
- Automatically infers dependences between resources

LOGIC APPS

- Can handle thousands of parallel runs per stamp
- Resilient against failure – retries with “at least once” guarantee
- Simple, declarative JSON definition
- Automatically infers dependences between actions

... more power

RESOURCE MANAGER

- 8 functions for basic referencing and string functions
- Can only PUT resources into resource groups
- Auth must be through AAD
- Triggered manual through UI or API

LOGIC APPS

- Over 50 functions from string manipulations to math, to sets operations, to logical operators
- All HTTP operations supported on any arbitrary endpoint
- Supports many OAuth providers, AAD, Cert auth, or Basic auth
- Different ways to be triggered including recurrence

Pricing

Actions

	Price per execution
Actions	\$ 0.000025
Standard Connector	\$ 0.000125
Enterprise Connector	\$ 0.001

Every time a Logic App definition runs the triggers, action and connector executions are metered.

Integrations

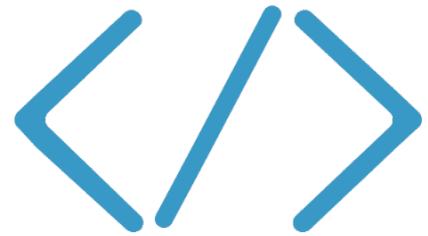
	Basic	Standard
XMLMaps	50	500
XMLSchemas	50	500
EDI Trading Partners	2	500
EDI Agreements	1	500
Price / Month	\$ 300	\$ 1000

Pricing example:

- 10 Actions = $10 * \$ 0.000025 = \$ 0.00025$
- 5 Standard Connectors = $5 * \$ 0.000125 = \$ 0.000625$
- 4320 Executions (per 10 minutes) = $4320 * (\$ 0.00025 + \$ 0.000625) = \$ 3.78$



Azure Functions

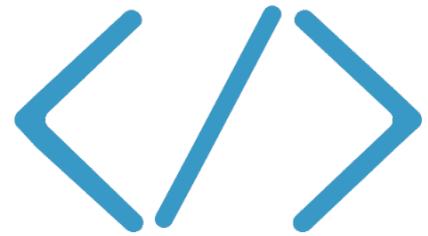


I have a “code”



I have an “event data”





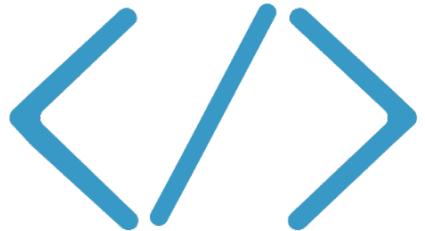
I have a “code”



I have an “event data”

BOOM!





C#
F#
Java
NodeJS
Python
PowerShell
...
...

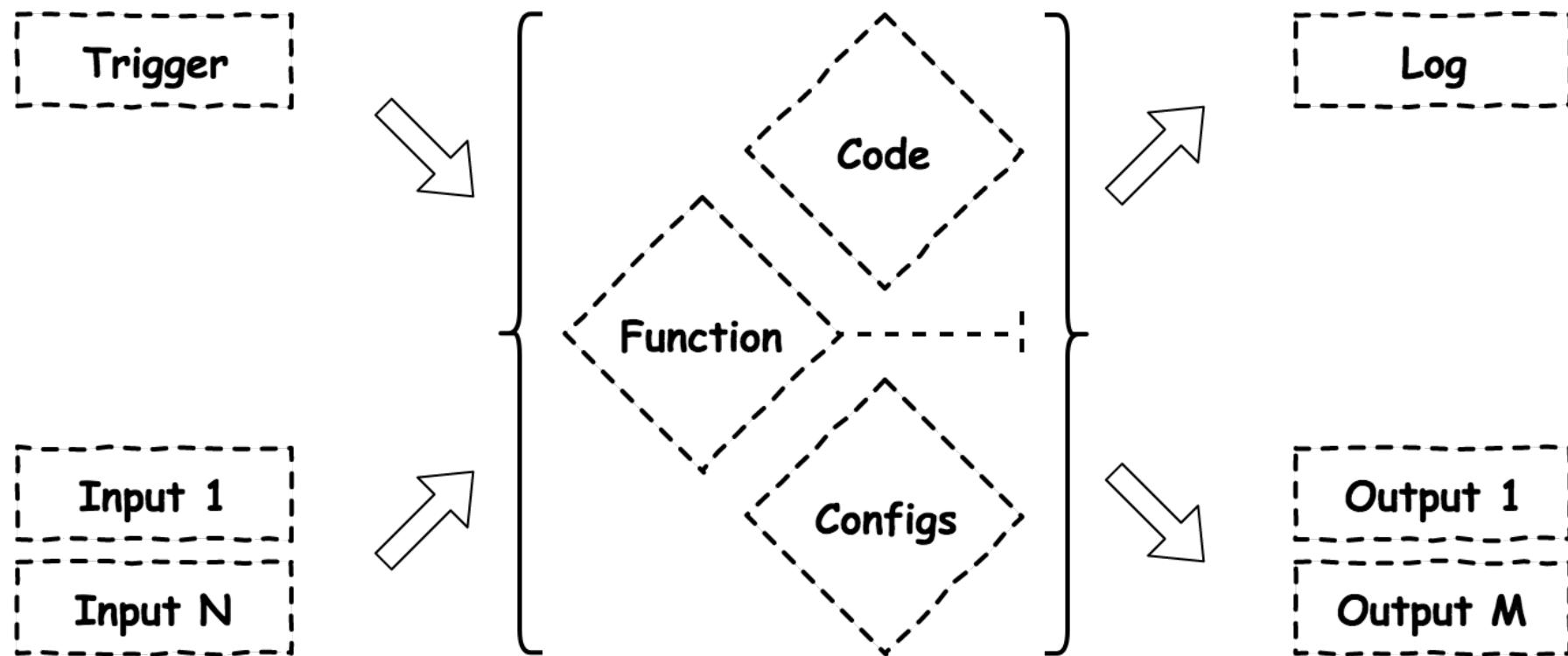


Azure
Functions



HTTP(S)
Timer
Queue
Storage
...

Triggers and Bindings



run.csx

```
1  using ImageResizer;
2
3  public static void Run(
4      Stream image,                                // input blob, large size
5      Stream imageSmall, Stream imageMedium) // output blobs
6  {
7      var imageBuilder = ImageResizer.ImageBuilder.Current;
8      var size = imageDimensionsTable[ImageSize.Small];
9
10     imageBuilder.Build(
11         image, imageSmall,
12         new ResizeSettings(size.Item1, size.Item2, FitMode.Max, null), false);
13
14     image.Position = 0;
15     size = imageDimensionsTable[ImageSize.Medium];
16
17     imageBuilder.Build(
18         image, imageMedium,
19         new ResizeSettings(size.Item1, size.Item2, FitMode.Max, null), false);
20 }
```

function.json

```
1  {
2      "disabled": false,
3      "bindings": [
4          {
5              "path": "sample-images/{name}",
6              "type": "blobTrigger",
7              "name": "image",
8              "direction": "in",
9              "connection": "ImagesInput"
10         },
11         {
12             "path": "sample-images-sm/{name}",
13             "type": "blob",
14             "name": "imageSmall",
15             "direction": "out",
16             "connection": "ImagesOutput"
17         },
18         {
19             "path": "sample-images-md/{name}",
20             "type": "blob",
21             "name": "imageMedium",
22             "direction": "out",
23             "connection": "ImagesOutput"
24         }
25     ]
26 }
```

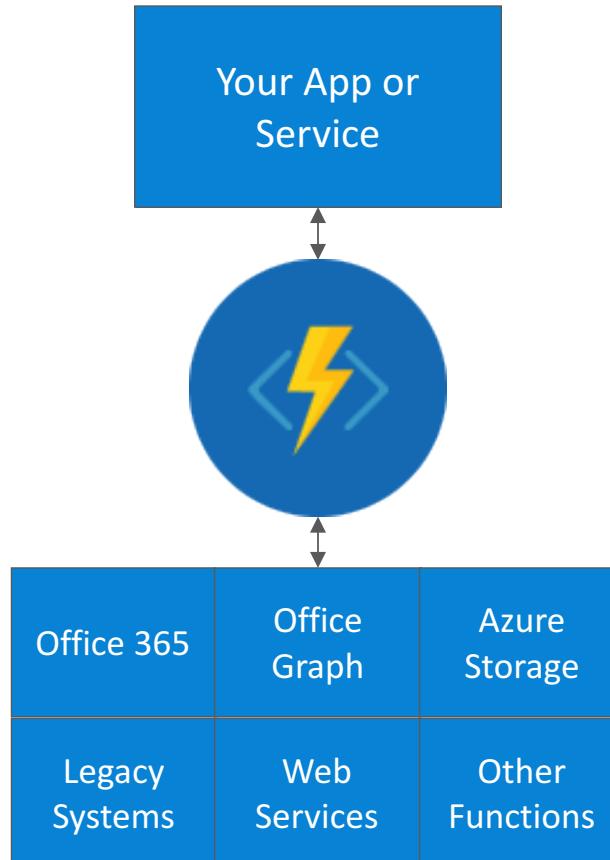
Triggers and Bindings

Bindings serve as the basis for all connections to and from a function.
Many bindings can be “bi-directional” as well.

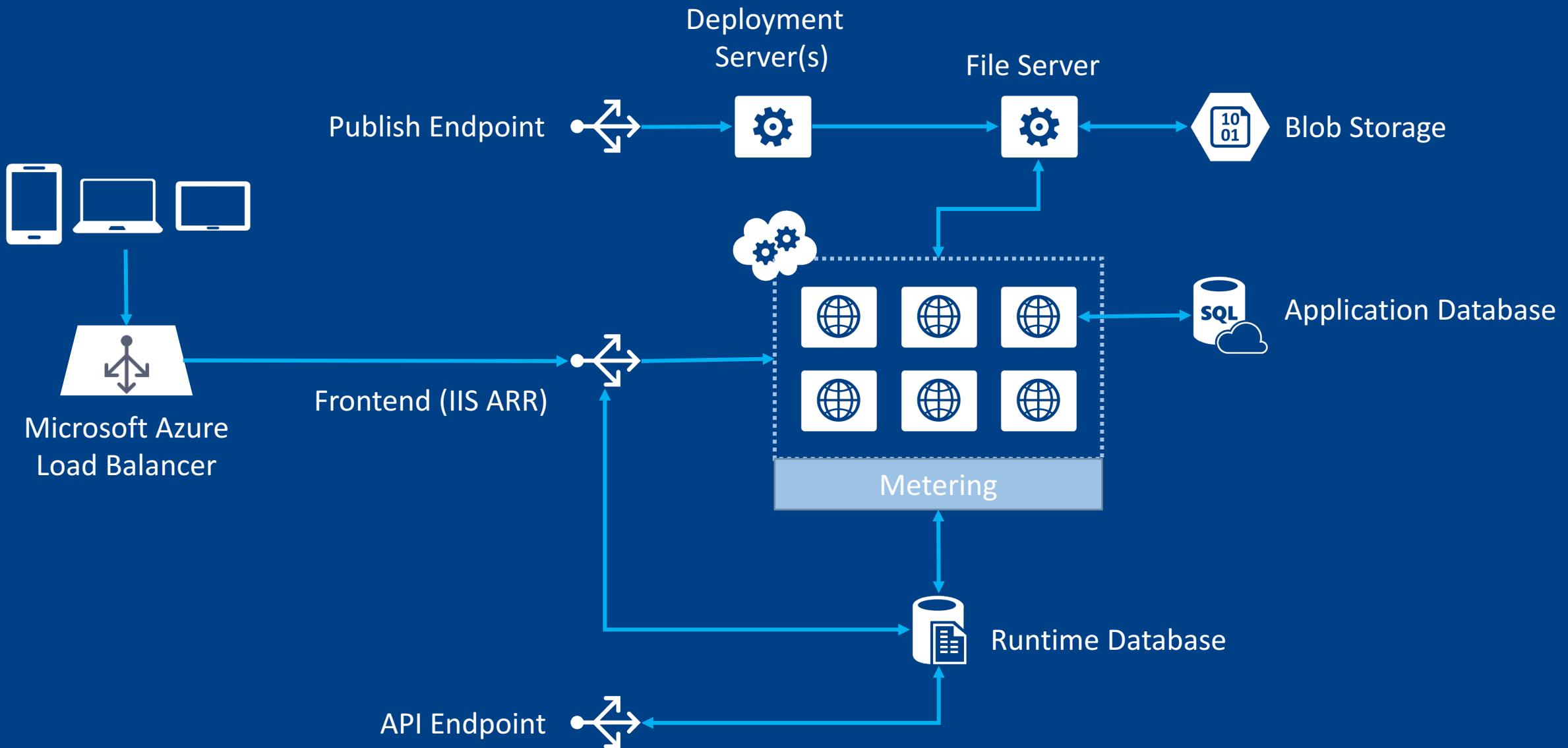
Type	Service	Trigger	Input	Output
Schedule	Azure Functions	✓		
HTTP(S)	Azure Functions	✓		✓
Blob Storage	Azure Storage	✓	✓	✓
Events	Azure Event Hubs	✓		✓
Queues	Azure Storage	✓		✓
Tables	Azure Storage		✓	✓
Tables	Azure Mobile Apps		✓	✓
No-SQL DB	Azure DocumentDB		✓	✓
Push Notifications	Azure Notification Hubs			✓

Common Scenarios

- Timer-based processing
- Azure service event processing
- SaaS event processing
- Serverless web application architectures
- Serverless mobile backends
- Real-time stream processing
- Real-time bot messaging



App Services Architecture



Pricing

	Free	Consumption
Execution Time	400 000 GB-s	\$ 0.000016 / GB-s
Total Executions	1 million executions	\$ 0.20 per million executions

Observed resource consumption is calculated by multiplying average memory size in gigabytes by the time in milliseconds it takes to execute the function. Memory used by a function is measured by rounding up to the nearest 128 MB, up to the maximum memory size of 1,536 MB, with execution time calculated by rounding up to the nearest 1 ms. The minimum execution time and memory for a single function execution is 100 ms and 128 mb respectively.

Pricing example:

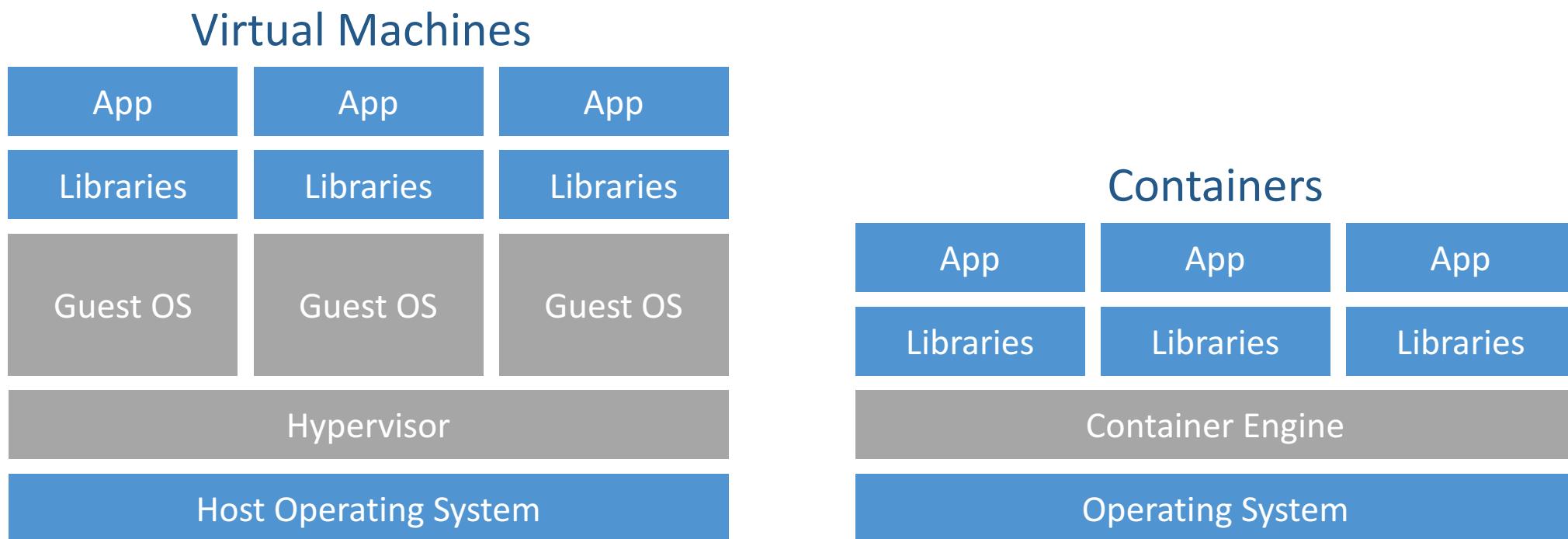
- 3M Executions * 1sec = 3M sec
- 512M Memory * 3M sec = 1.5M GB-s
- Billable Executions = $(3M - 1M) * \$0.20 = \0.40
- Billable Memory = $(1.5M \text{ GB-s} - 0.4M \text{ GB-s}) * \$0.000016 = \$17.6$



Azure Container Instances

Containers

- Lightweight alternative to virtual machines
- Smaller, less expensive, faster to start up, and self-contained

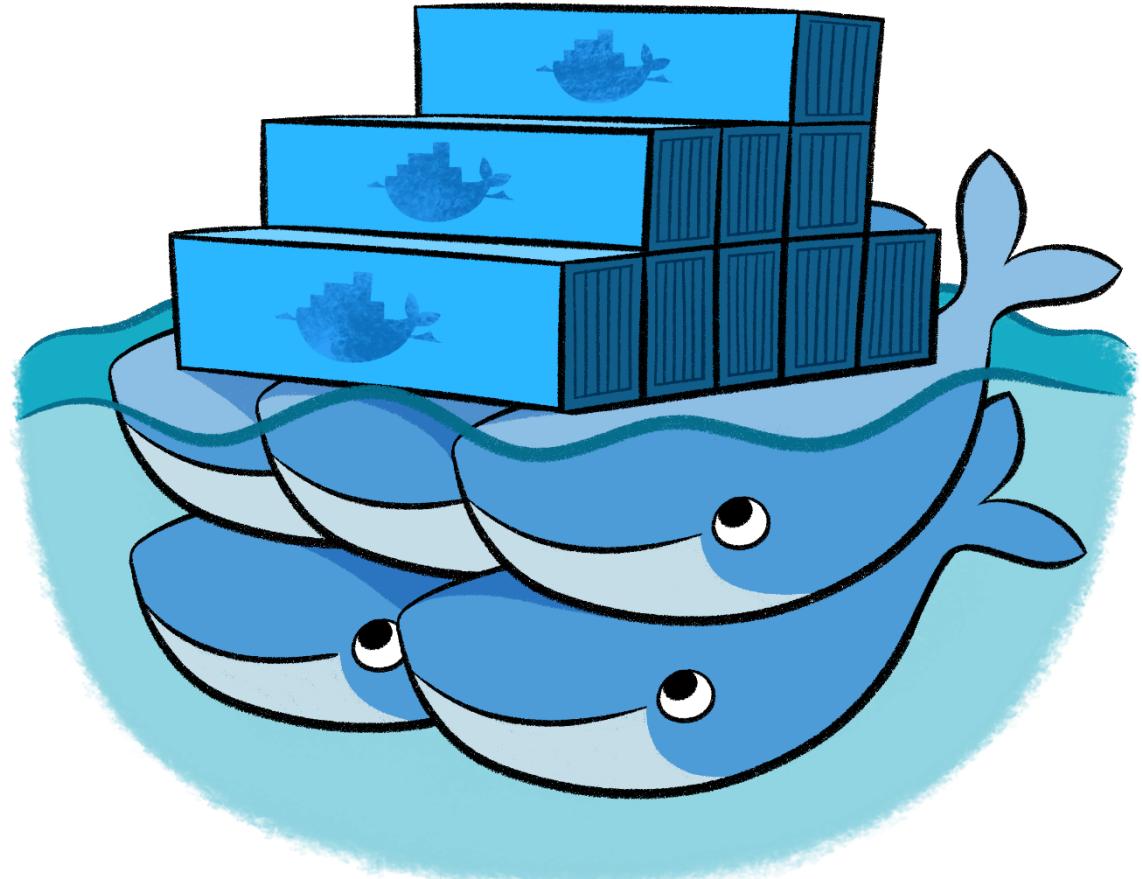


Docker

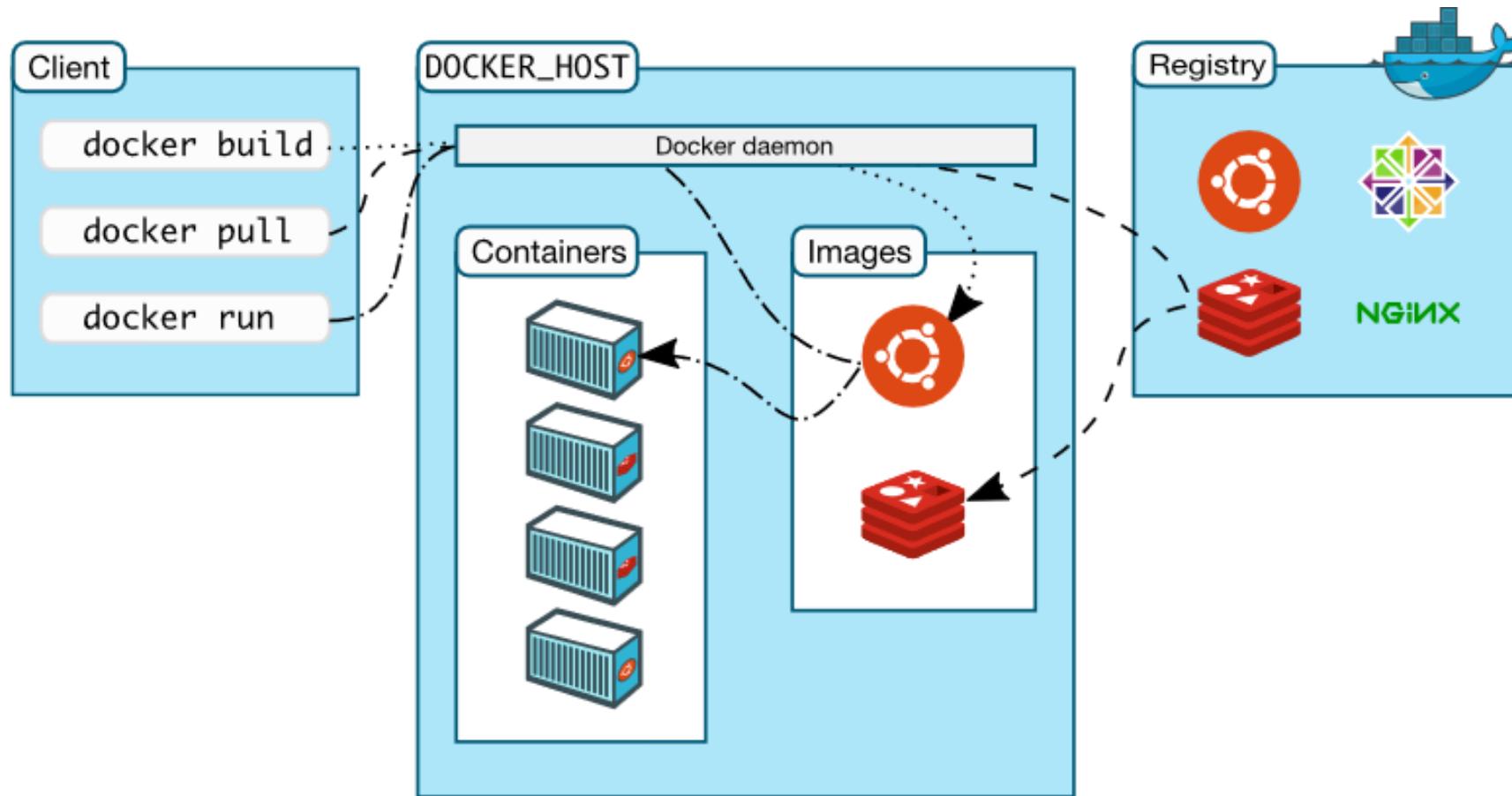
- Leading open-source containerization platform

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

- Supported natively in Azure

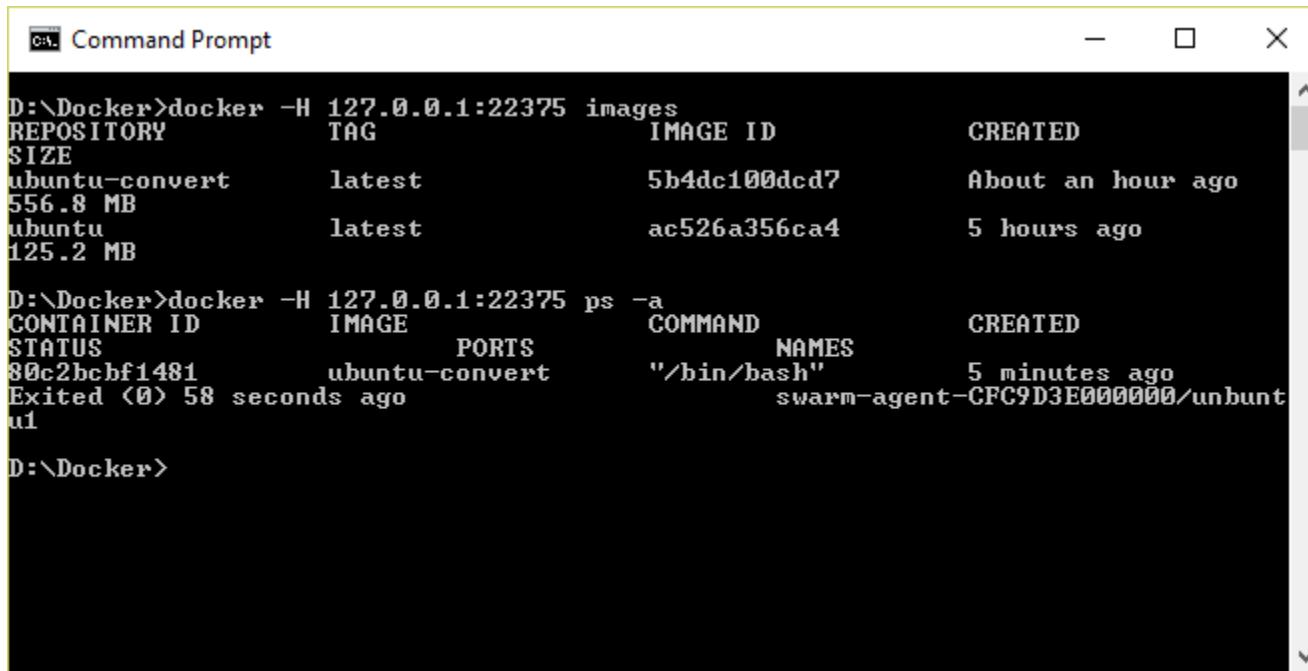


Docker Architecture



Docker CLI

- Command-line interface for Docker, available for Linux, OS X, and Windows (available separately or as part of Docker Toolbox)



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window contains two lines of Docker CLI commands and their outputs:

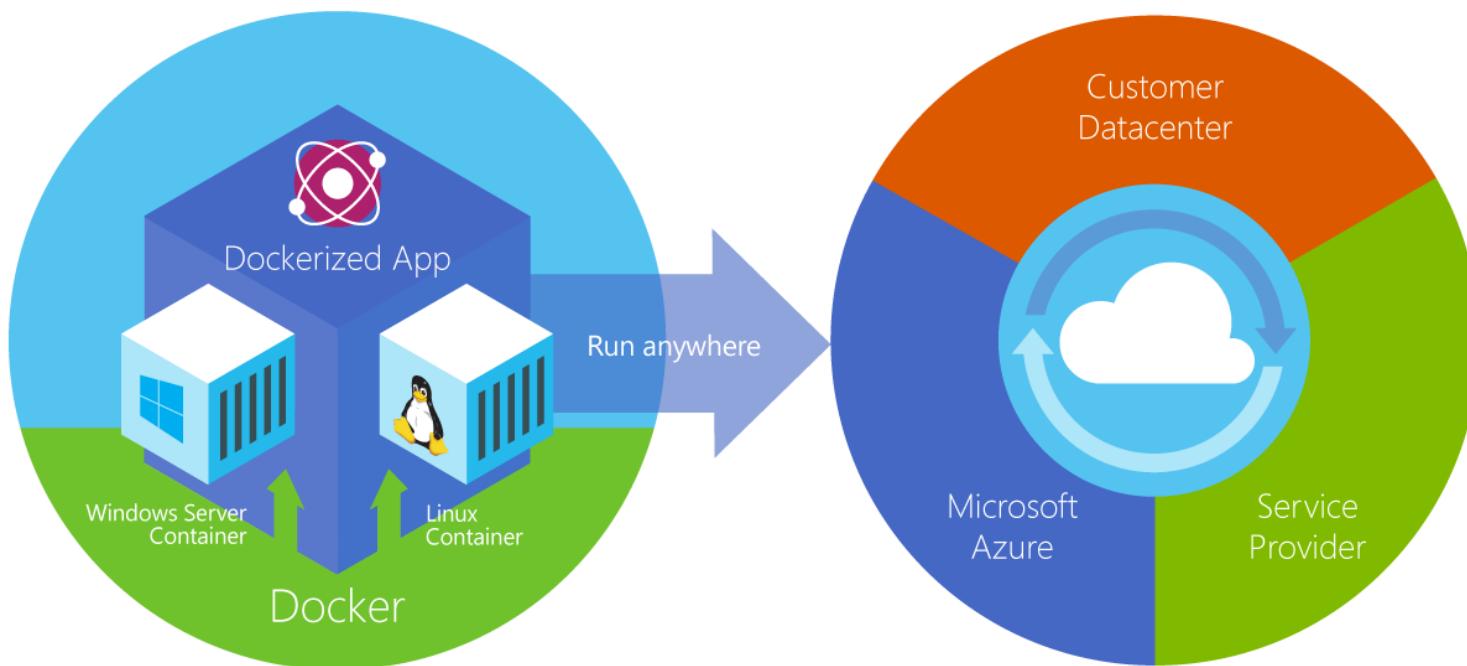
```
D:\>Docker>docker -H 127.0.0.1:22375 images
REPOSITORY          TAG      IMAGE ID      CREATED
SIZE
ubuntu-convert      latest   5b4dc100dcd7  About an hour ago
556.8 MB
ubuntu              latest   ac526a356ca4  5 hours ago
125.2 MB

D:\>Docker>docker -H 127.0.0.1:22375 ps -a
CONTAINER ID        IMAGE      COMMAND      CREATED
STATUS             PORTS      NAMES
80c2bcbf1481       ubuntu-convert "/bin/bash"  5 minutes ago
Exited (0) 58 seconds ago
swarm-agent-CFC9D3E0000000/unbunt
u1

D:\>Docker>
```

Azure Container Instances

- Provides robust, ready-to-use Docker hosting environment



Create Container Instance via Portal

Microsoft Azure New

+ New

Dashboard

All resources

Resource groups

App Services

SQL databases

SQL data warehouses

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Azure Active Directory

Monitor

Advisor

New

Search the Marketplace

Azure Marketplace See all

Featured See all

- Get started
- Compute
- Networking
- Storage
- Web + Mobile
- Databases
- Data + Analytics
- AI + Cognitive Services
- Internet of Things
- Enterprise Integration
- Security + Identity
- Developer tools
- Monitoring + Management
- Add-ons
- Containers
- Blockchain

Azure Container Service Create

Azure Container Registry Create

Azure Container Instances (preview) Create

Docker EE for Azure (Std/Adv) - [deprecated] Create

DC/OS on Azure Create

RancherOS Create

Docker on Ubuntu Server Create

Create Azure Container Inst... Basics

1 Basics Configure basic settings >

2 Configuration Specify container requirements >

3 Summary Azure Container Instances (pre... >

* Container name mycontainer

Container image type Public Private

* Container image microsoft/aci-helloworld

Subscription Microsoft Azure

* Resource group Create new Use existing myResourceGroup

* Location East US

OK

Create Container Instance via Portal

The image shows the Azure Container Instance creation interface across three main stages: Basics, Configuration, and Summary.

Basics (Step 1): Done. Shows the container name as "mycontainer".

Configuration (Step 2): Specify container requirements. This step is highlighted in blue. It includes fields for OS Type (Windows selected), Number of cores (1), Memory (1.5 GB), Public IP address (Yes), and Port (80).

Summary (Step 3): Done. This step is also highlighted in blue. It provides a summary of the configuration:

Setting	Value
Subscription	Microsoft Azure
Resource group	myResourceGroup
Location	East US
Container name	mycontainer
Container image type	Public
Container image	microsoft/aci-helloworld
Configuration	
Number of cores	1
Memory (GB)	1.5
Public IP address	Yes
Port	80

An "OK" button is visible at the bottom of the Summary screen, and a red box highlights the "OK" button on the right side of the Summary screen.

Create Container Instance via Azure CLI

```
az container create \
    --resource-group myResourceGroup \
    --name mycontainer3 \
    --image microsoft/aci-wordcount:latest \
    --restart-policy OnFailure \
    --environment-variables NumWords=3 MinLength=5 \
    --command-line "python wordcount.py http://shakespeare.mit.edu/romeo_juliet/full.html"
```

Pricing

	Free
Create container instance	\$ 0.0025 per Instance created
Container instance duration	Memory: \$ 0.0000125 per GB-s Cores: \$ 0.0000125 per Core-s

Container instance duration is calculated from the time your container begins executing until it terminates. The price depends on the number of cores and GBs of memory allocated to the instance. You are charged \$0.0000125 for every GB and core used. For each container instance, you can allocate a minimum of 1 GB, up to 3.5 GBs of memory to each core. You can allocate up to 4 cores to each instance you deploy.

Pricing example:

- 50 create requests per day * 30 days * \$ 0.0025 = \$ 3.75
- 1500 create requests * 150 seconds * 2 GB * \$ 0.0000125 per GB-s = \$ 5.625
- 1500 create requests * 150 seconds * 1 core * \$ 0.0000125 per core-s = \$ 2.8125
- Create requests + memory duration + core duration = \$ 12.19

