



Azure for Go developers

Abhishek Gupta

Senior Developer Advocate, Microsoft



@abhi_tweeter



abhirockzz



I'm here to...

- (try and) Sell you Go
- Chat about Azure, Go
- Share useful info
- Pray to Demo Gods!
- Questions?

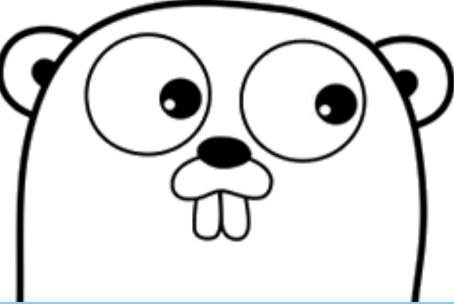


Image credit: [Ashley McNamara](#)

aka.ms/azure-for-gophers

What is Go (aka *golang*)?

Go is an open source programming language that makes it easy to build **simple, reliable, and efficient** software.



[!\[\]\(35de7ce9c97e259aff6f01ac90da87f8_img.jpg\) Download Go](#)

Binary distributions available for Linux, macOS, Windows, and more.

Try Go [Open in Playground ↗](#)

```
// You can edit this code!
// Click here and start typing.
package main

import "fmt"

func main() {
    fmt.Println("Hello, 世界")
}
```

Hello, World! ▾ [Run](#) [Share](#) [Tour](#)

<https://golang.org/>

Hello World

main.go



```
package main

import "fmt"

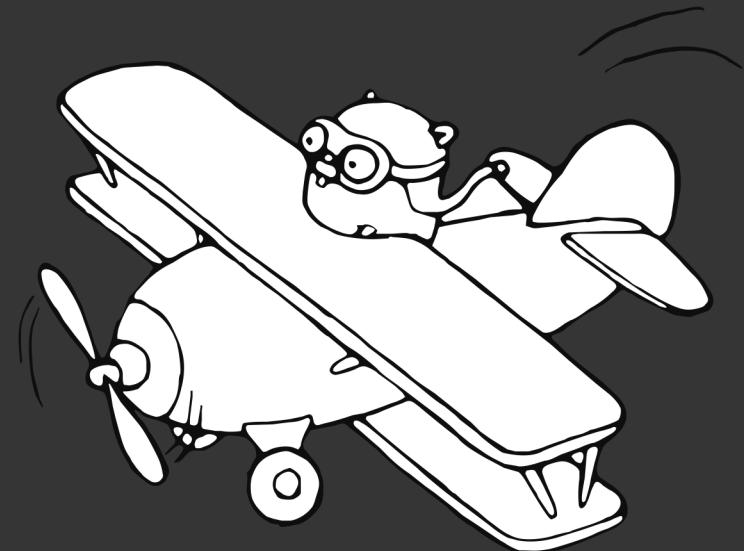
func main() {
    hello("Azure")
}

func hello(what string) {
    fmt.Println("Hello,", what)
}
```

```
$ go run main.go
$ Hello, Azure
```

Go, the good parts (my top picks)

- The good parts
 - Simplicity - [Simplicity is Complicated \(Rob Pike\)](#)
 - Solid [standard library](#) – to *use* and *learn* from
 - One way to do things (most of the times)
 - Hard to change – less fluff, sometimes boring
- Not so good
 - Dep management, app structure
- What can you use it for??
 - Cloud & Networking services - dominant
 - Web/APIs – great std lib support!
 - CLI apps – my personal favorite
 - DevOps & SRE



<https://go.dev/>

In the Cloud...

- Low resource overhead
- (again) Standard lib – Concurrency, HTTP, gRPC, DB etc.
- Tooling – profiling, benchmarking etc.
- Static binary (also applies to CLIs)

Go was created to address exactly these concurrency needs for scaled applications, microservices, and cloud development. In fact, over 75 percent of projects in the Cloud Native Computing Foundation are written in Go.

<https://go.dev/solutions/cloud/>

- Want to get started?
 - Install Go, VS Code on your laptop
 - Tour of Go: <https://tour.golang.org/>
 - Effective Go https://golang.org/doc/effective_go.html
- Dig in!
 - [Rob Pike](#) (co-creator of Go)- pretty much any talk
 - [The Why of Go](#) - by [Carmen Andoh](#)
 - [Sameer Ajmani](#) – concurrency talks
 - [Francesc Campoy](#) – you must watch [justforfunc](#) !

Did I manage to convince you???



<https://go.dev/solutions#case-studies>



nats.io

prometheus.io



The world's fastest framework for
building websites

gohugo.io

What's up with Go on Azure?

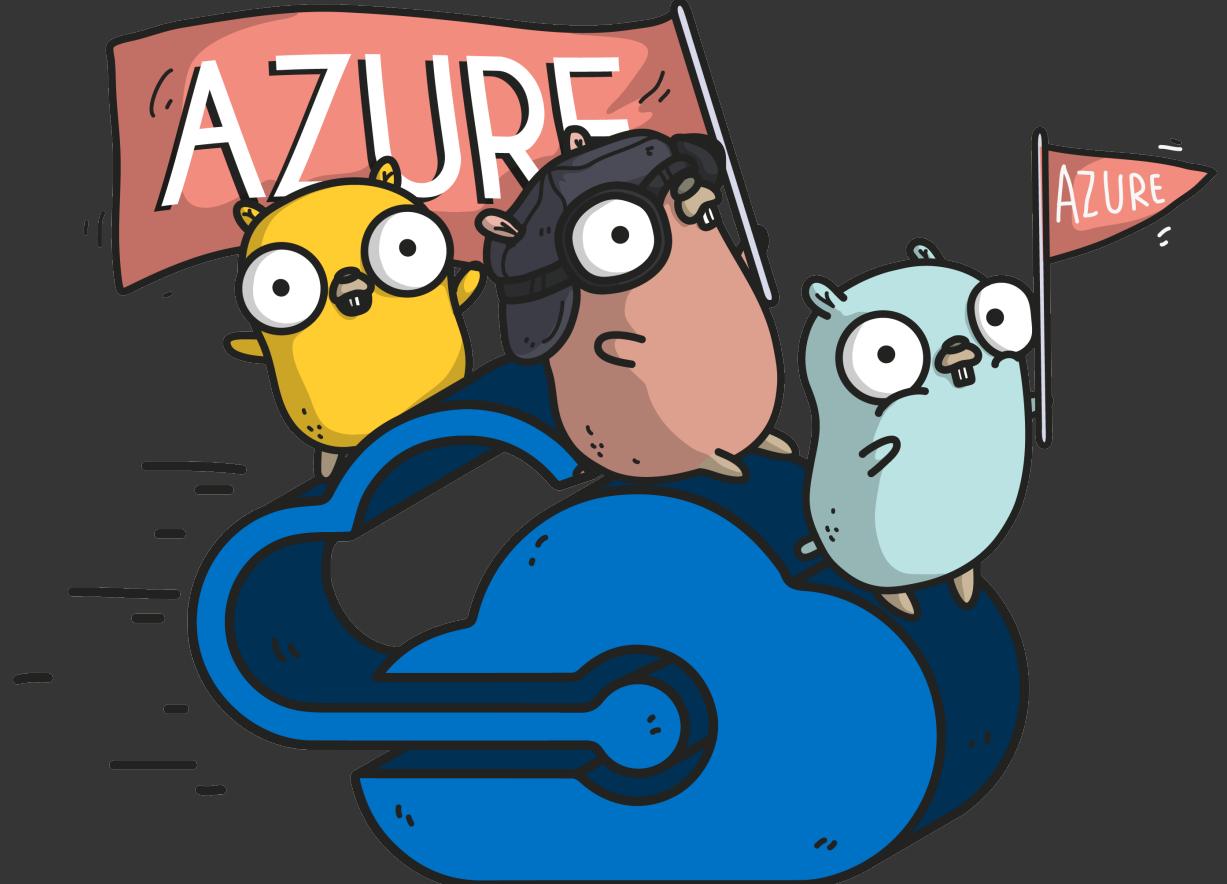


Image credit: [Ashley McNamara](#)

Azure Go SDK

- Go packages for *managing* Azure services
 - Authenticate using a variety of options (common)
 - Create VMs, attach a Disk
 - Create a Cosmos DB account and updates its throughput
 - Create/List/Delete Key Vaults
- Managing vs Using?
 - Managing – control plane op
 - Using – data plane op
- Go SDKs for specific services (data plane)
<https://github.com/Azure/azure-sdk-for-go#other-azure-go-packages>



Image credit: [Ashley McNamara](#)

Cloud Native Apps (PaaS, Containers, Kubernetes)

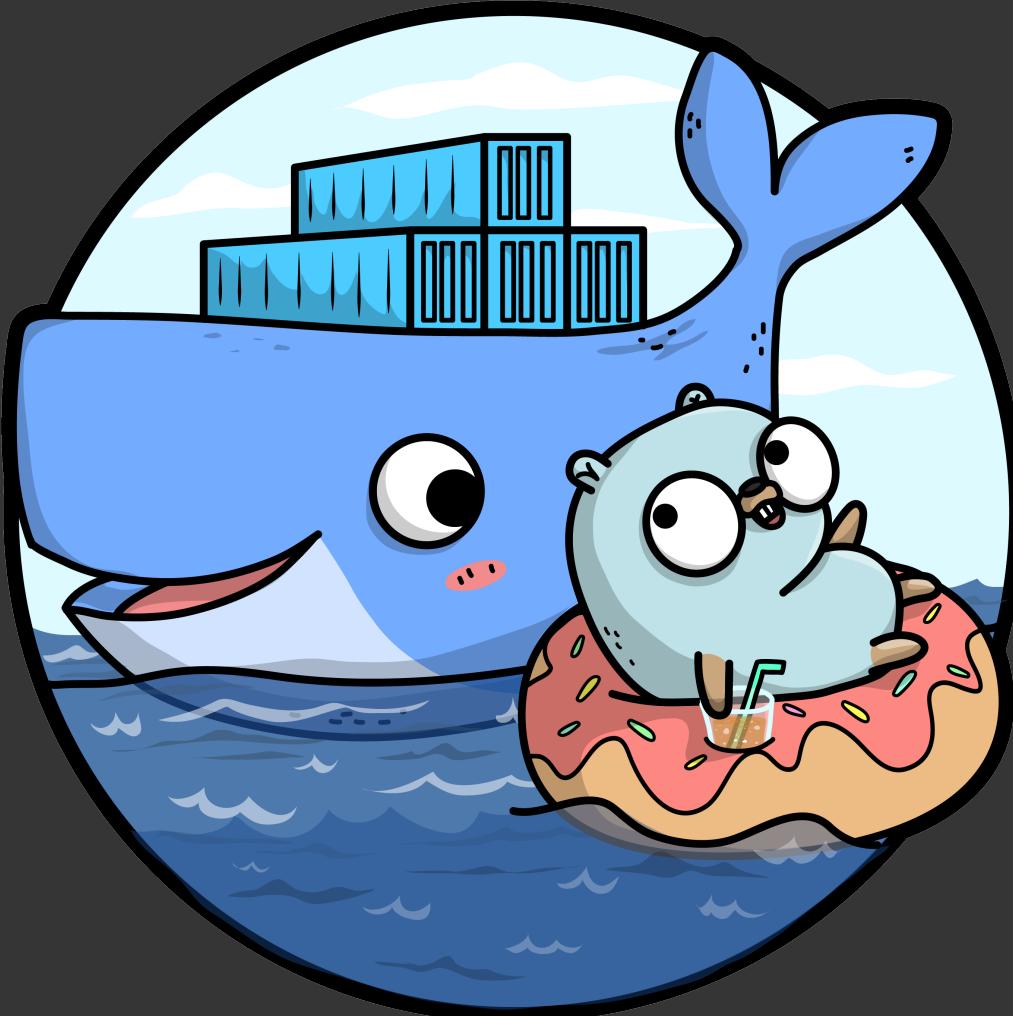


Image credit: [Ashley McNamara](#)

Databases - RDBMS

- Fully managed DBaaS
- You don't need anything special!
- [Go db/sql](#) standard library
 - Other libs – [sqlx](#), [squirrel](#) etc.
- PostgreSQL – [lib/pq](#), [pgx](#)
 - Connection pooling [Docker container](#) (PgBouncer)
- MySQL – battle tested [mysql driver](#)
 - How GitHub uses it, [finds and resolves critical bugs](#)
- SQL Server – yes [there is a Go driver](#)



Databases – NoSQL, Big Data & friends!

- [Go + Redis] - Simple, lightweight and lightning fast!
- Redis = Swiss army knife
 - cache, session store, pub/sub
 - queue, leaderboard, geospatial
 - analytics, stream processing, Redis modules (ML etc.)
- Libraries
 - go-redis - <https://github.com/go-redis/redis>
 - gorilla/sessions - <https://github.com/gorilla/sessions>
- Azure Cache for Redis + Redis Labs partnership



Databases – NoSQL, Big Data & friends!

- Azure Cosmos DB
 - globally distributed, scalable, fast!
- Multi-model
 - SQL (Core)
 - Cassandra, MongoDB
 - Gremlin (Graph)
 - Table storage and etcd



aka.ms/cosmosdb-intro

Cosmos DB for Gophers

- It's easy!
 - Mongo DB – [official Go driver](#)
 - Cassandra – [gocql](#) is quite promising!
- Examples
 - Go Quick-starts for [Cassandra](#) and [MongoDB](#) APIs
 - More (Go examples)
 - [How to handle rate-limiting in Cassandra API for Cosmos DB ?](#)
 - [How to use Mongo DB Change Streams ?](#)

Databases – NoSQL, Big Data & friends!

- Azure Data Explore (Kusto) for Big Data analytics
- Key parts: Ingestion, Query (KQL)
- Go SDKs (separate)
 - Management: Cluster, DB etc.
 - Operations: Batch & Streaming Ingestion, Query etc.
- Interested? [Try it out!](#)



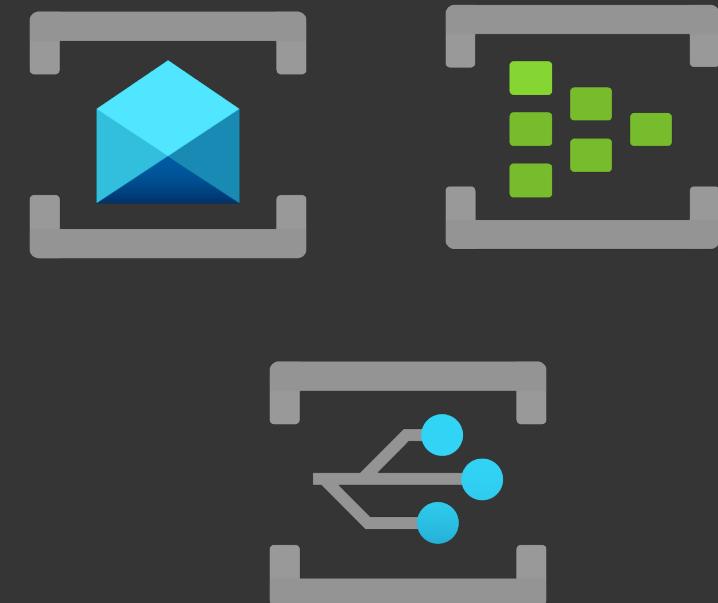
	Azure SQL Database	Azure SQL Managed Instance	SQL Server on Virtual Machines	Azure Database for PostgreSQL	Azure Database for MySQL	Azure Database for MariaDB	Azure Cosmos DB	Azure Cache for Redis
Relational Database	✓	✓	✓	✓	✓	✓		
Non-Relational Database (NoSQL)							✓	
In-Memory Database								✓
Data Models	Relational	Relational	Relational	Relational	Relational	Relational	Multi-Model: Document Wide-column Key-Value Graph	Key-Value
Hybrid	✓	✓	✓	✓ (Hyperscale)				
Serverless Compute	✓						✓	
Storage Scale Out	✓ (Hyperscale)			✓ (Hyperscale)			✓	✓
Compute Scale Out	✓ (Hyperscale - read-only)			✓ (Hyperscale)			✓	✓
Distributed Multi-Master Writes (Write data to different regions)							✓	✓ (Coming Soon)
OSS Based Service (Community edition and open extension support)				✓	✓	✓		✓
HTAP (Available with Azure Synapse Link)	✓ (Coming Soon)			✓ (Coming Soon)			✓	

Messaging (for async communication)

- Everything cannot be sync RPC/JSON over HTTP!
- Service Bus, Event Hubs, Event Grid

Comparison of services

Service	Purpose	Type	When to use
Event Grid	Reactive programming	Event distribution (discrete)	React to status changes
Event Hubs	Big data pipeline	Event streaming (series)	Telemetry and distributed data streaming
Service Bus	High-value enterprise messaging	Message	Order processing and financial transactions



aka.ms/azure-messaging

CI/CD, Monitoring, Dev tools

- GitHub Actions, Azure Dev Ops
- Azure App Insights
- VS Code



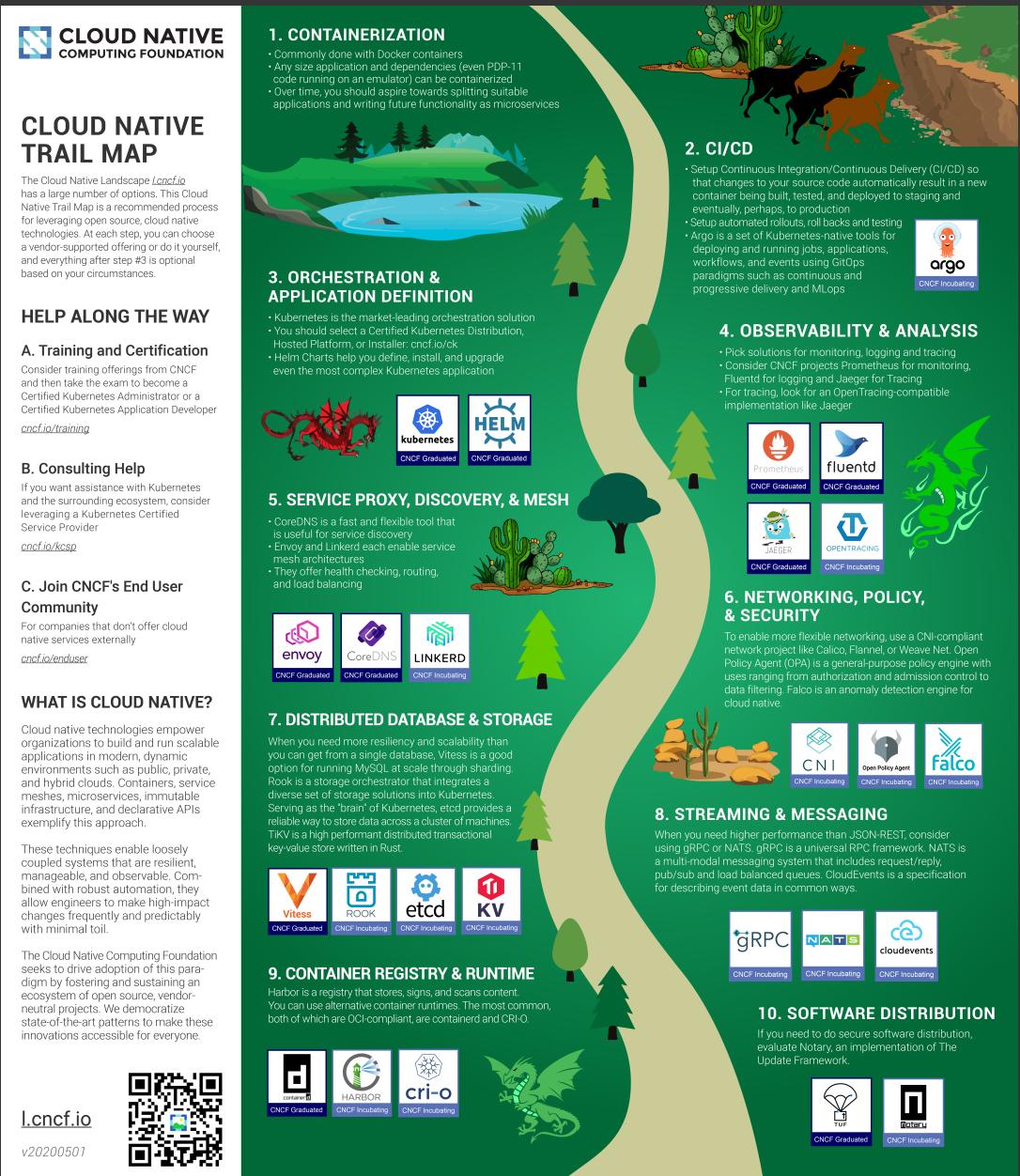
Image credit: [Ashley McNamara](#)

Cloud Native



Welcome!

- Lots of things...
- Microservices, Containers, DevOps
- Storage/Databases – polyglot persistence
- Networking, Service Mesh
- Observability
-



https://raw.githubusercontent.com/cncf/trailmap/master/CNCF_TrailMap_latest.png

l.cncf.io

v20200501



Title not found (404)

<https://landscape.cncf.io/>

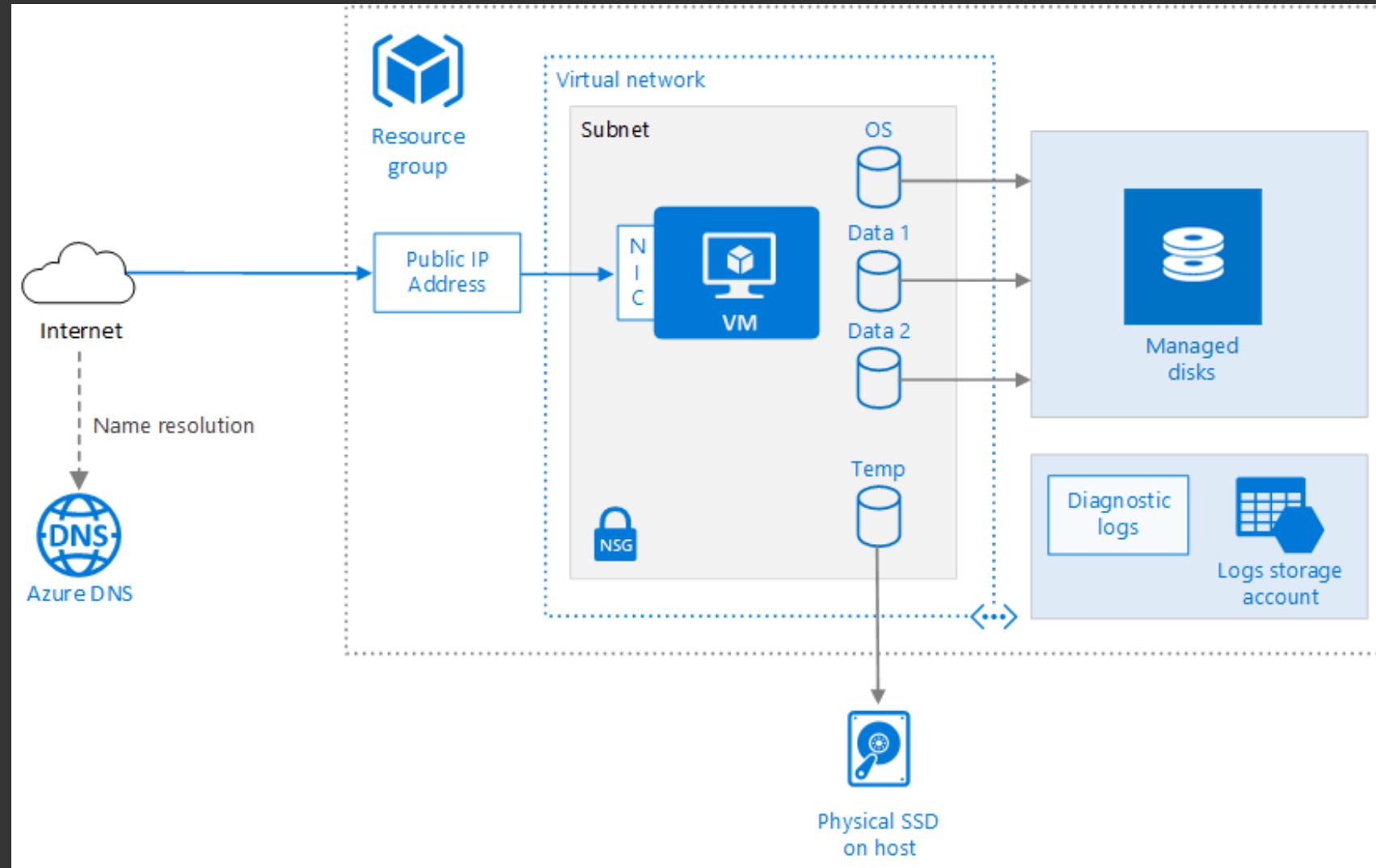


<https://twitter.com/dastbe/status/1303858170155081728/photo/1>

For today, let's just stick to...



IaaS: Lift and shift Go apps



aka.ms/az-linux-vm

PaaS: Azure App Service

- Fully managed platform
- Build Web apps, APIs, mobile app backends etc.
- Polyglot: Java, .NET, Node.js, Python etc.
- Scaling, CI/CD, Monitoring, Security
- Windows, Linux or *Containers*



Run Go apps with Docker support!

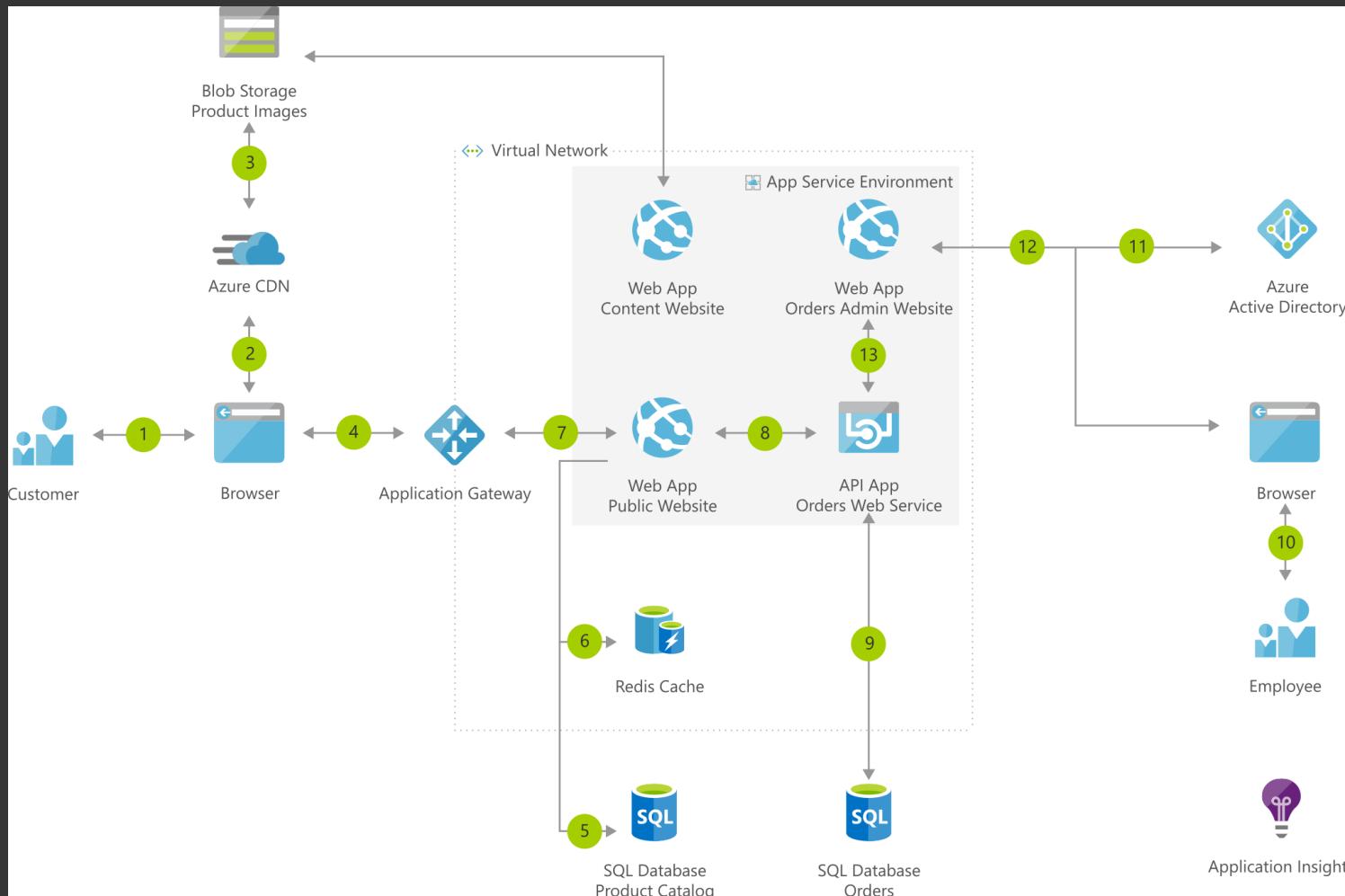
```
func translateText(text, toLang string) (translateAPIResponse, error) {  
  
    cognitiveServiceEndpoint := baseEndpoint + endpointSuffix + "&to=" + toLang  
  
    reqBody := `[{`Text`:"` + text + `"}]`  
    req, err := http.NewRequest(http.MethodPost, cognitiveServiceEndpoint,  
        strings.NewReader(reqBody))  
  
    req.Header.Add(subscriptionKeyHeader, subscriptionKey)  
    req.Header.Add("Content-Type", "application/json")  
  
    res, err := http.DefaultClient.Do(req)  
  
    var result translateAPIResponse  
    err = json.NewDecoder(res.Body).Decode(&result)  
    return result, nil  
}
```



Web App for Containers

aka.ms/az-cognitive-services

Example: E-Commerce backend



aka.ms/ecommerce-backend

CaaS: Azure Container Instances

- Fast and simple!
- Fully managed infrastructure
- Per-second billing

```
az container create  
-n mycontainer --image microsoft/aci-helloworld  
-g mygroup --ip-address public
```



Azure Container Instances

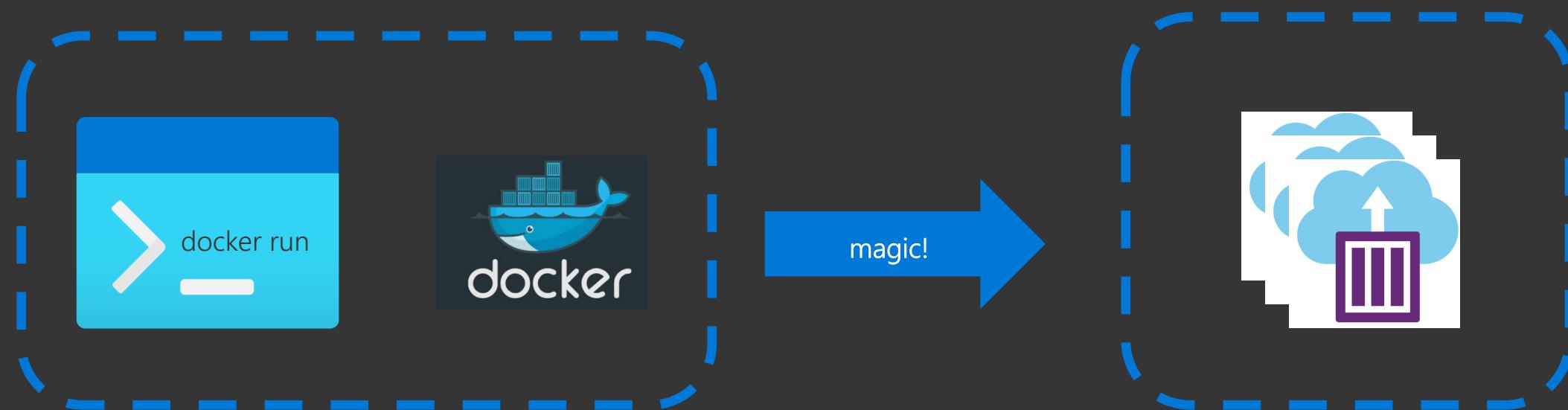
- Use cases: Event driven apps
 - Integrate with Logic Apps
 - Build jobs
 - “bursting” with AKS (advanced)
- Features:
 - Container groups – multi-container apps
 - Persistence – Azure File Share etc.
 - CI/CD – Azure Container Registry
 - Virtual Network
 - Monitoring, logging

Deploy to ACI using Docker

Integration: Docker Desktop + Azure

Use native Docker commands, deploy to ACI!

Docker Compose support = Multi-container apps on ACI



aka.ms/aci-docker

Not built for complex orchestration

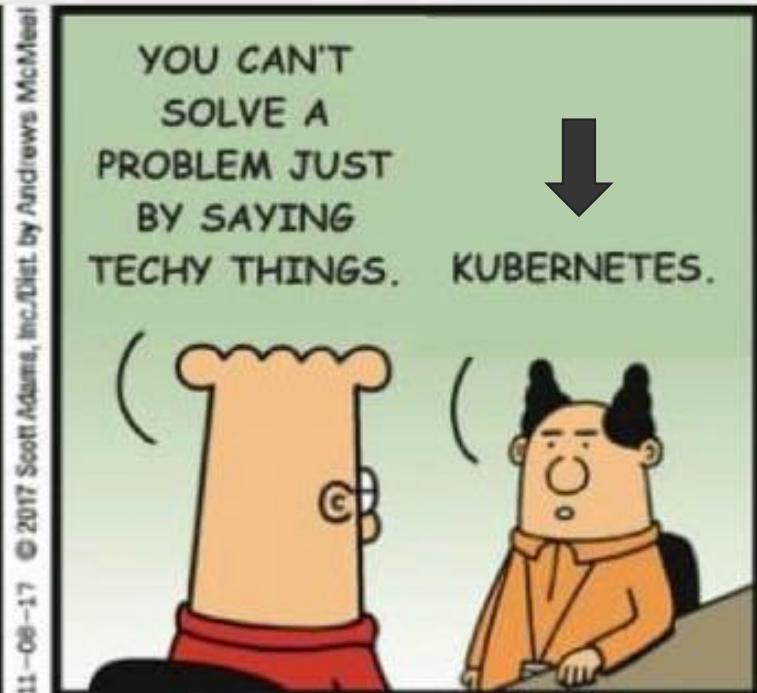
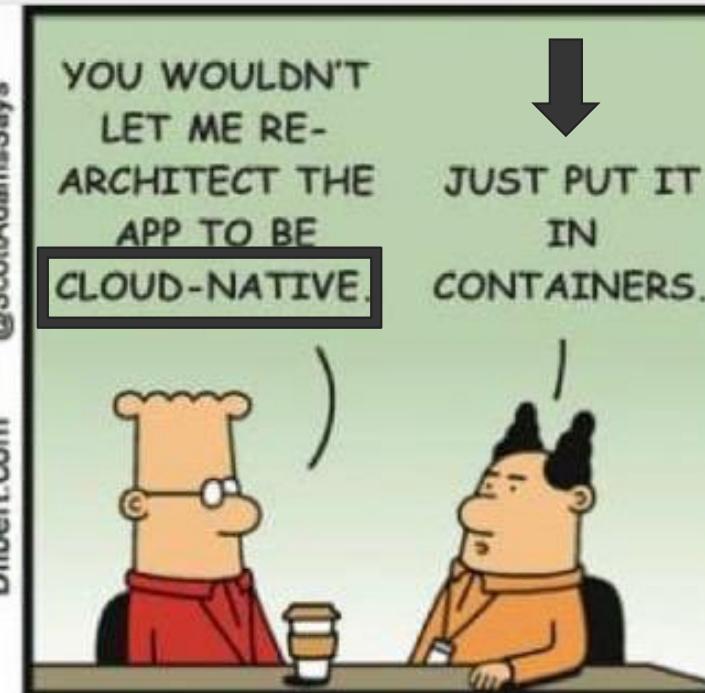
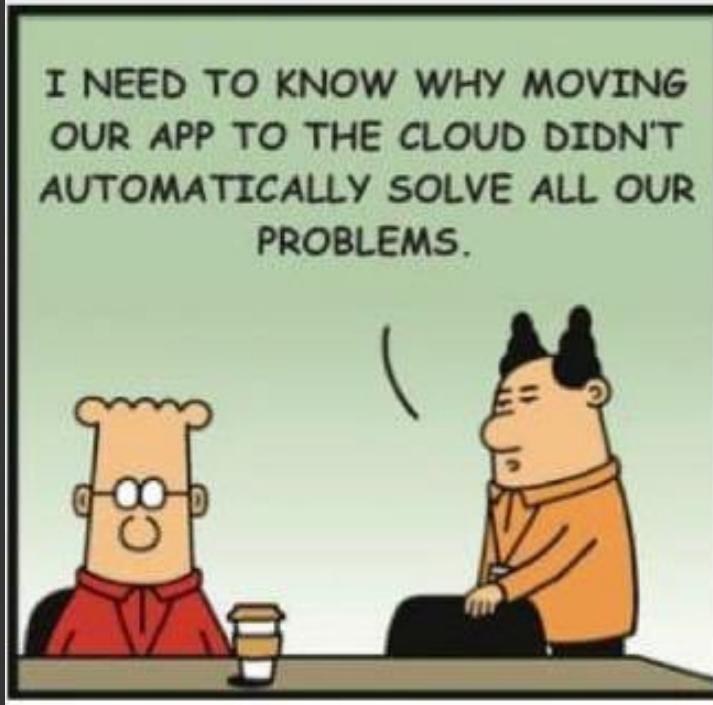
Lacks:

- Auto scale
- Service discovery
- Load balancing
- Zero downtime upgrades



Just to set the context....

Solved all your problems. You're welcome.



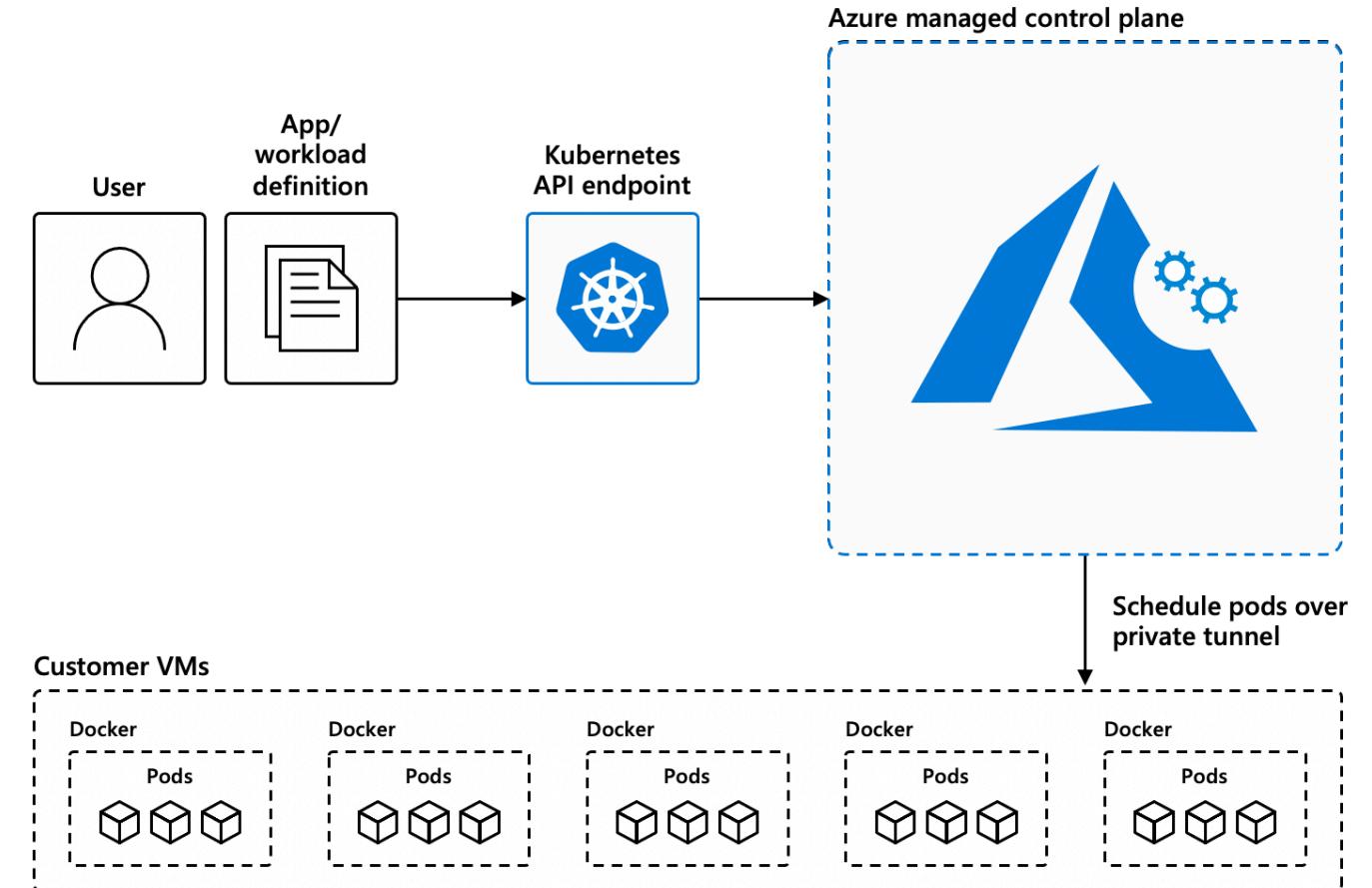
Azure Kubernetes Service



Managed Kubernetes

Responsibilities	DIY with Kubernetes	Managed Kubernetes on Azure
Containerization		
Application iteration, debugging		
CI/CD		
Provisioning, upgrades, patches		
Reliability availability		
Scaling		
Monitoring and logging		

Customer Microsoft



But.....



Kelsey Hightower 
@kelseyhightower

Even managed Kubernetes must be managed.

The truth is you're ultimately responsible for testing upgrades against your exact configuration and collection of third party extensions.

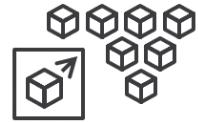
2:36 PM · Jun 12, 2019 · Twitter Web App

126 Retweets 19 Quote Tweets 504 Likes

Comment  Retweet  Like  Share 

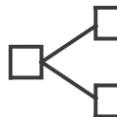
Key Use cases

Lift and shift to containers



Cost saving
without refactoring
your app

Microservices



Agility
Faster application
development

Secure DevOps



Automation
Deliver code faster and
securely at scale



Machine learning



Performance
Low latency
processing

IoT



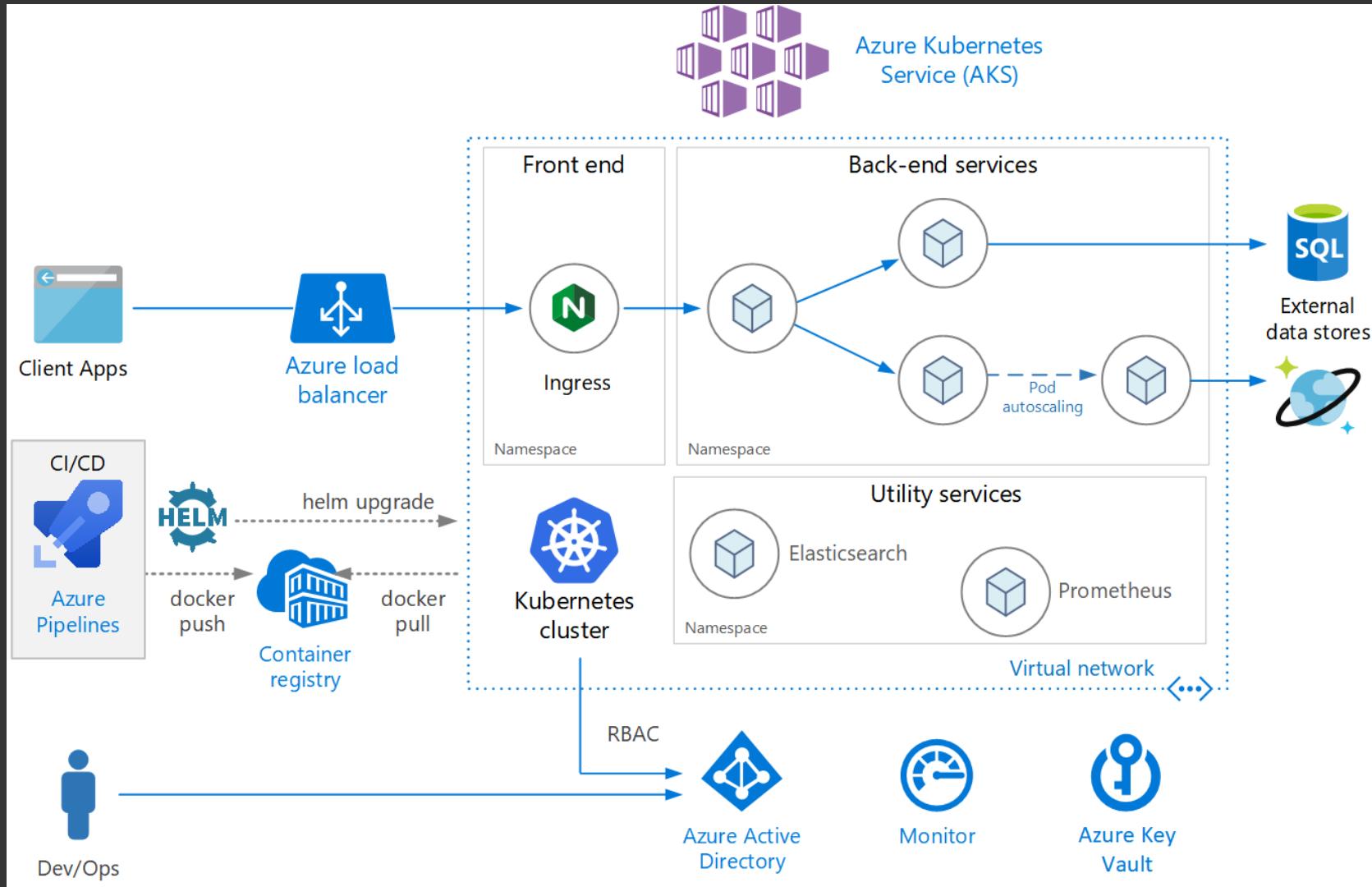
Portability
Build once,
run anywhere

Data streaming

— 101010
— 010101
— 101010

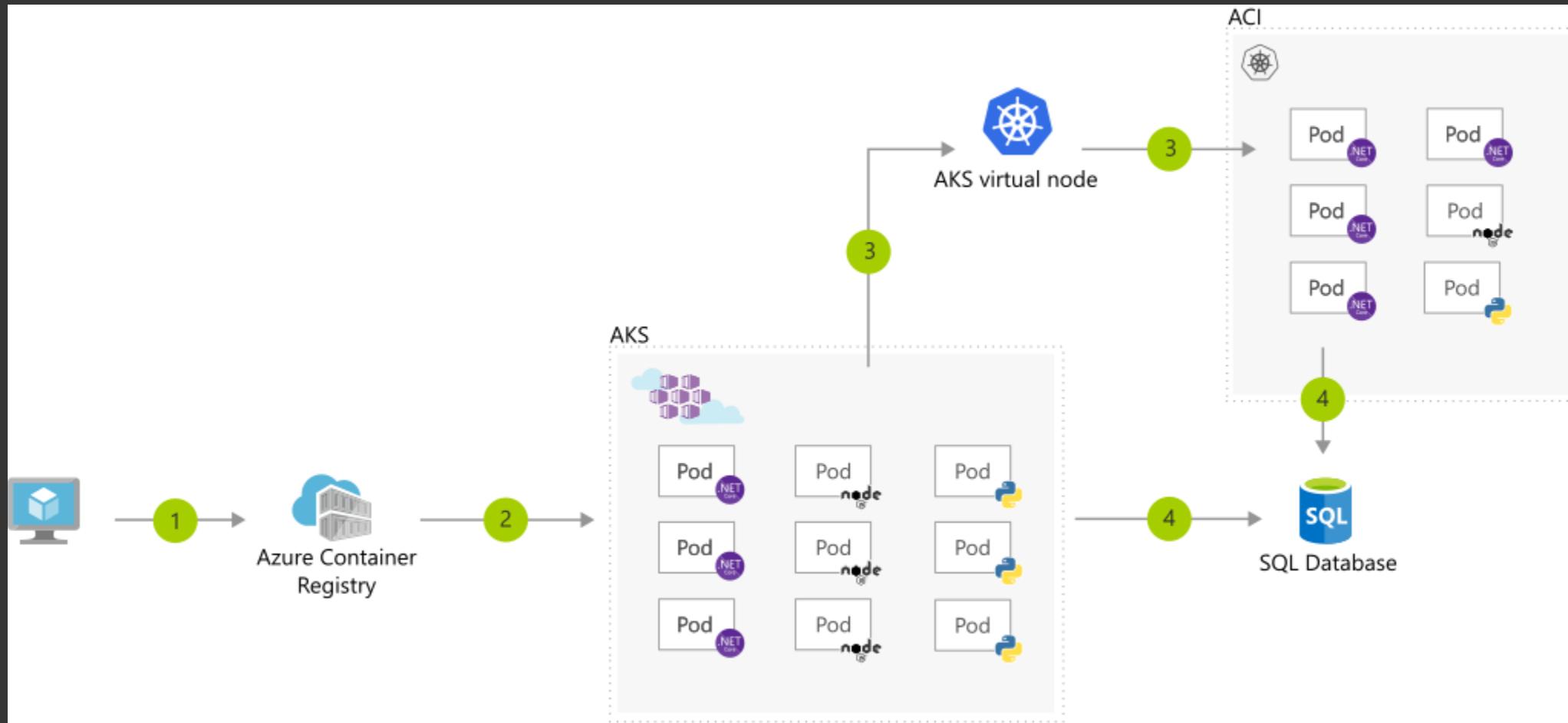
Analytics
Real-time data
collection and streaming

AKS for Microservices



aka.ms/aks-microservices

Serverless scale: From AKS to ACI



aka.ms/aks-serverless-scale

Lots of resources!

Learning path
aka.ms/LearnKubernetes



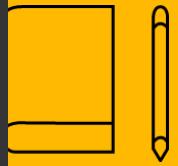
What is Kubernetes
aka.ms/k8sLearning



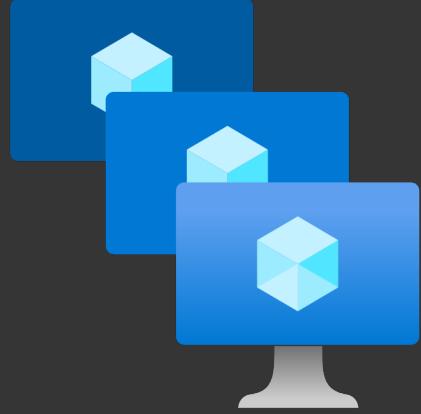
Hear from experts
aka.ms/AKS/videos



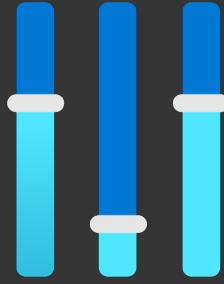
Case studies
aka.ms/aks/casestudy



Azure Functions - Serverless Compute



Lots of “servers”



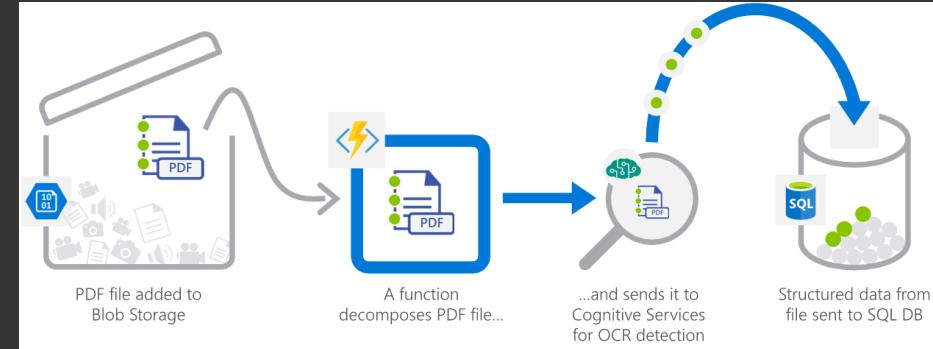
Scale on demand



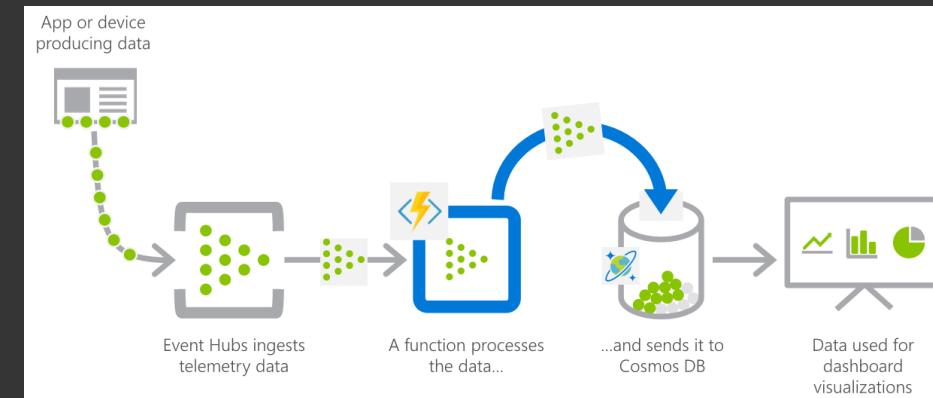
Pay for use

Common Use cases

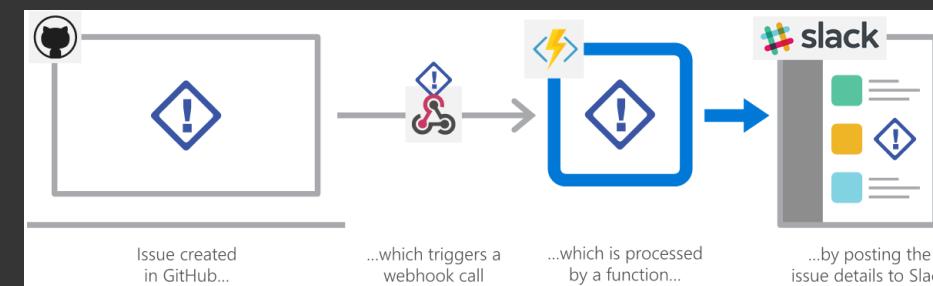
Real-time file processing



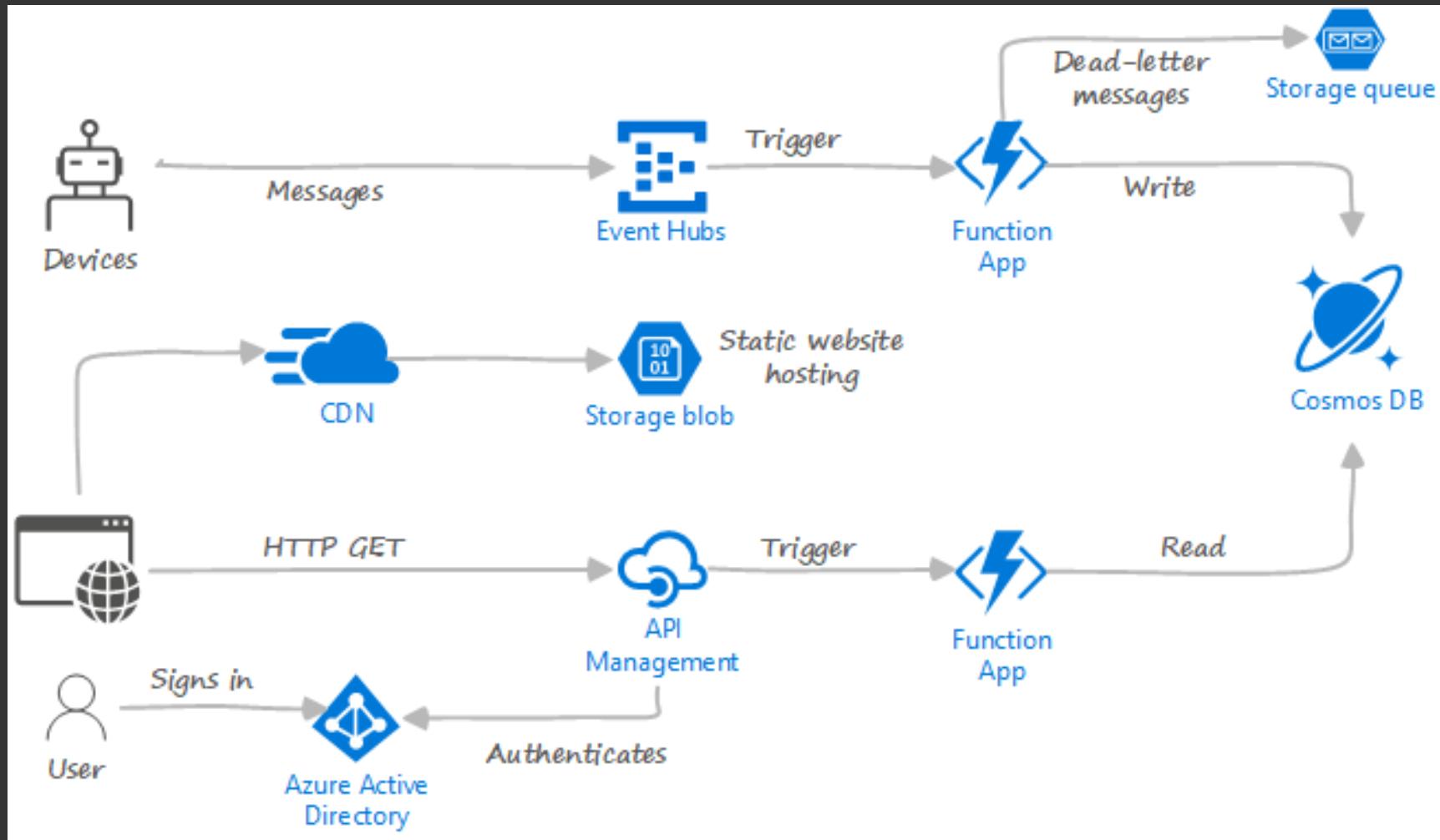
Real-time Stream processing



Extending SaaS applications

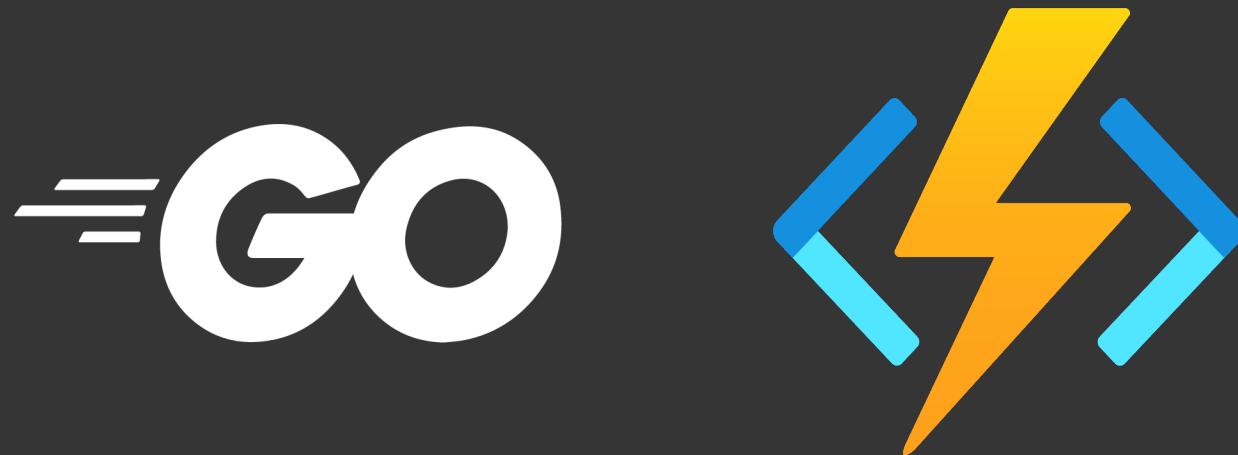


Example: Drone Telemetry System

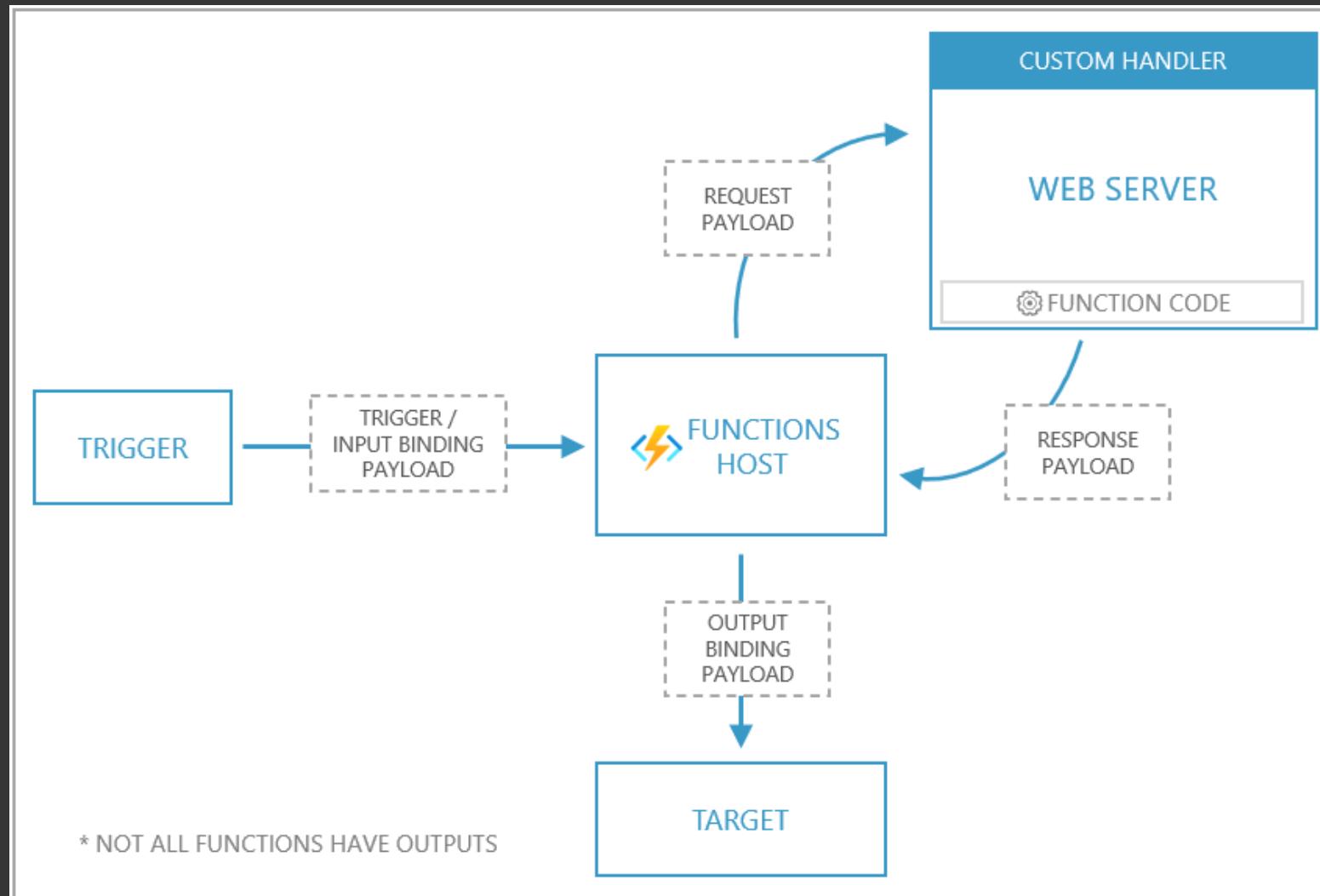


aka.ms/serverless-drone-telemetry

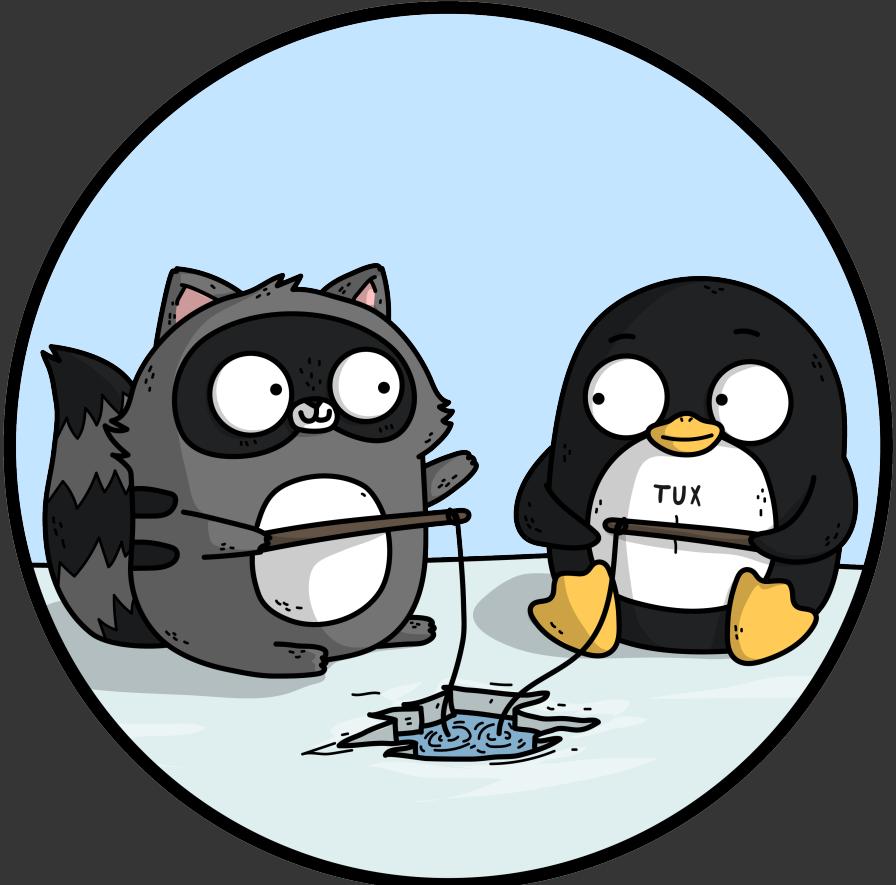
Run Go functions with Custom Handlers!



aka.ms/functions-custom-handlers



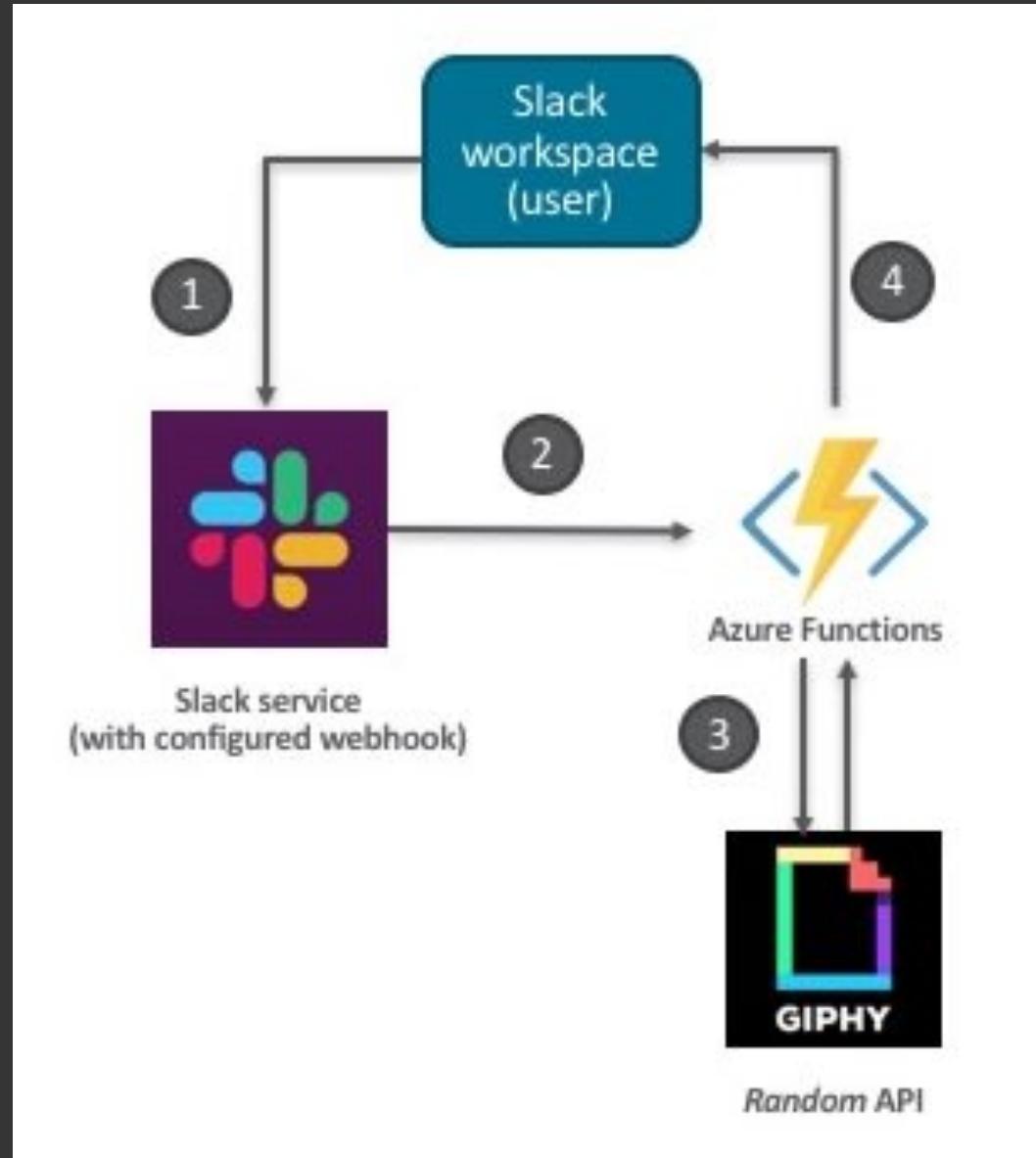
Demos!



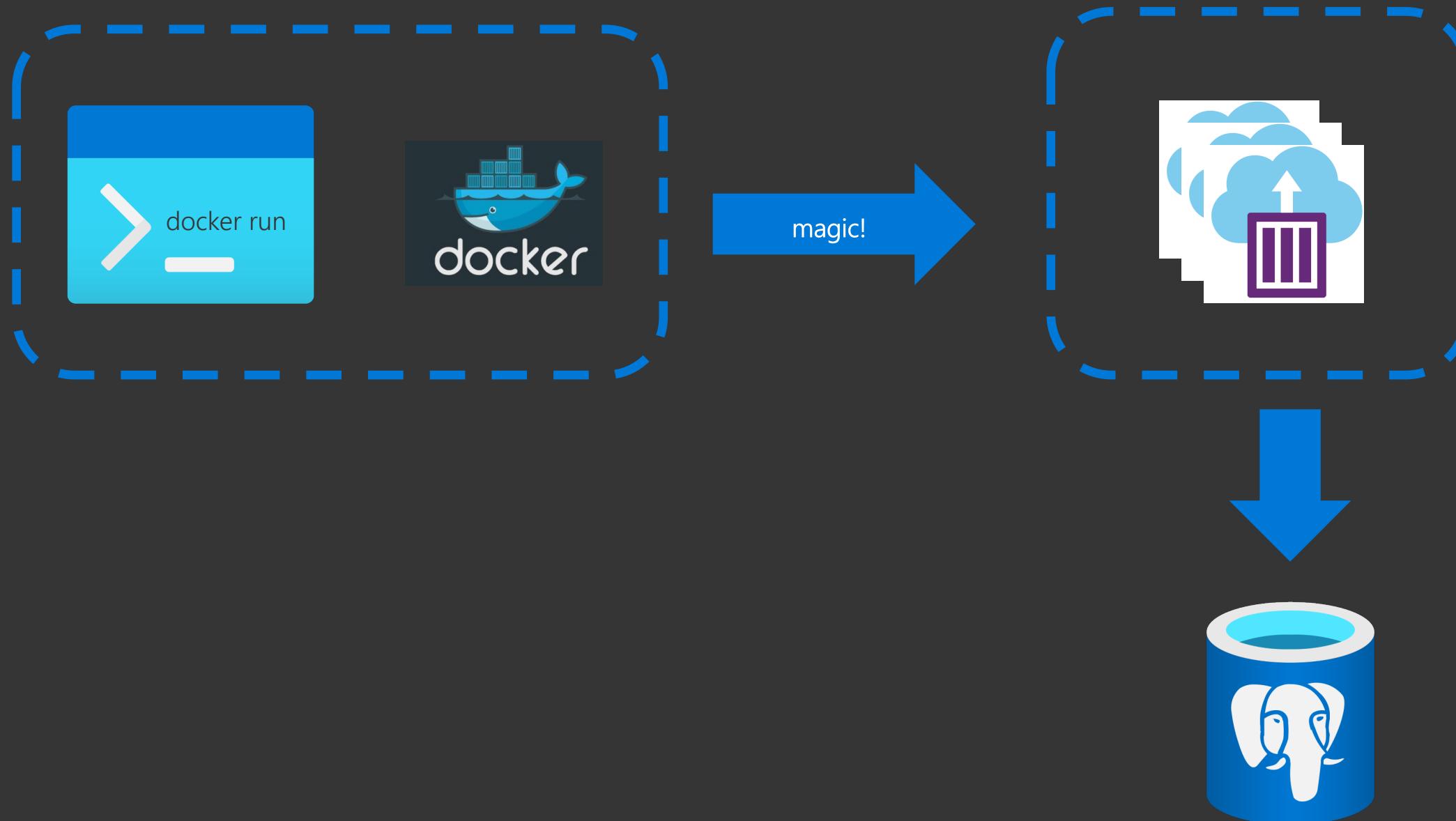
The plan...

1. [Go + Azure Functions]
 1. Slack app backend
 2. Write and deploy a Go function
2. [Go + Postgres] on ACI
 1. Write a simple Go + Postgres app, deploy to ACI using Docker
 2. Trigger ACI with Azure Go SDK
3. [Go + Redis] on AKS: auto-scale using KEDA

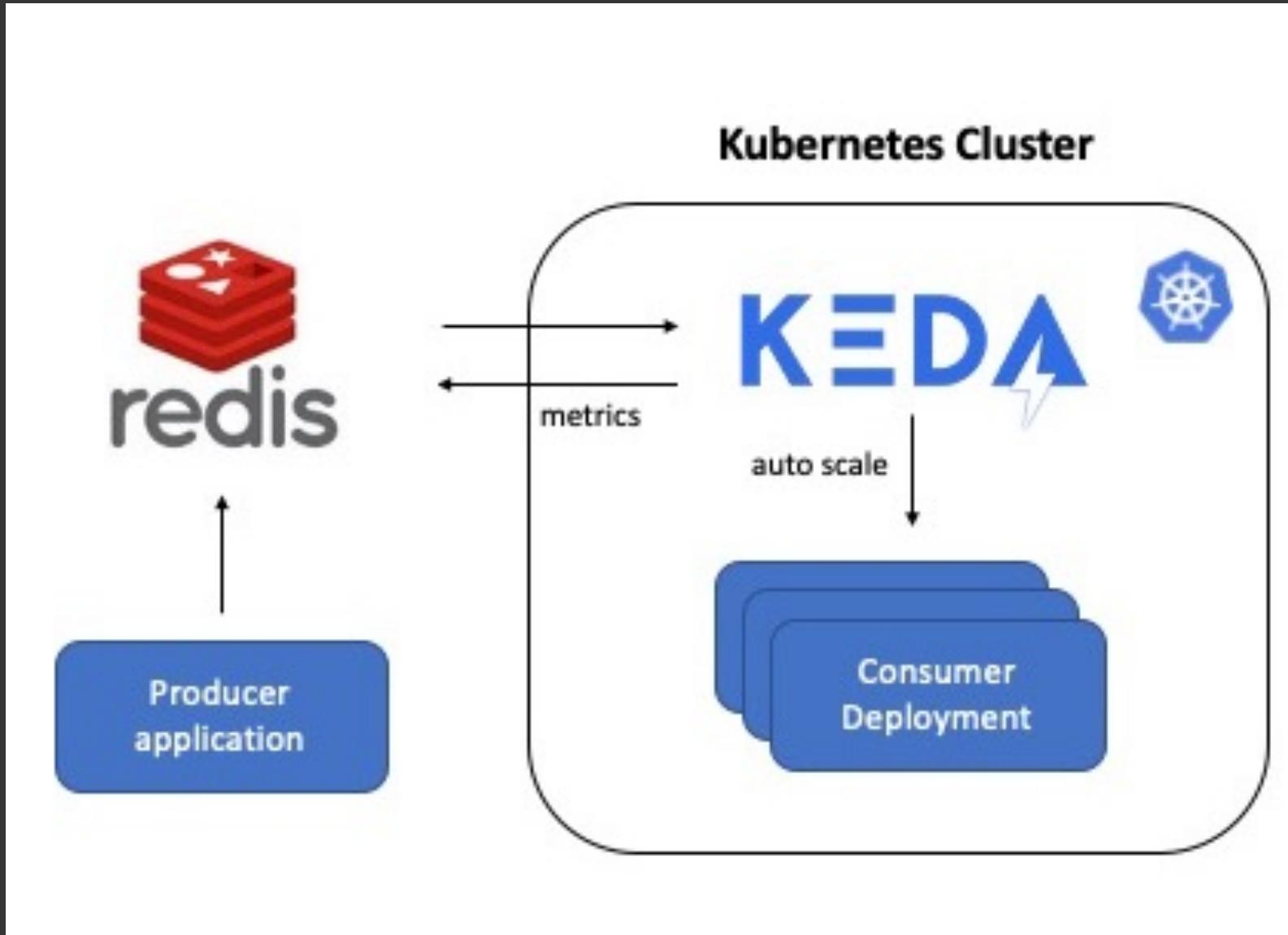
Slack app backend with Azure Functions and Go



Docker + ACI + Azure Go SDK



Scaling Go apps on Kubernetes with KEDA



Don't be shy, reach out!

- twitter.com/abhi_tweeter
- github.com/abhirockzz
- linkedin.com/in/abhirockzz/
- medium.com/@abhishek1987
- dev.to/abhirockzz



Image credit: [Ashley McNamara](#)



Thank You!

