serveRless
computing for R

useR! 2019 – Toulouse

by Christoph Bodner & Thomas Laber

Agenda

01
The Problem

Building a scalable and flexible pipeline to deploy R models

02
Serverless

What does this buzzword actually mean?

03
Architecture

A solution architecture for Azure

Agenda

01
The Problem

Building a scalable and flexible pipeline to deploy R models

02
Serverless

What does this buzzword actually mean?

03
Architecture

A solution architecture for Azure

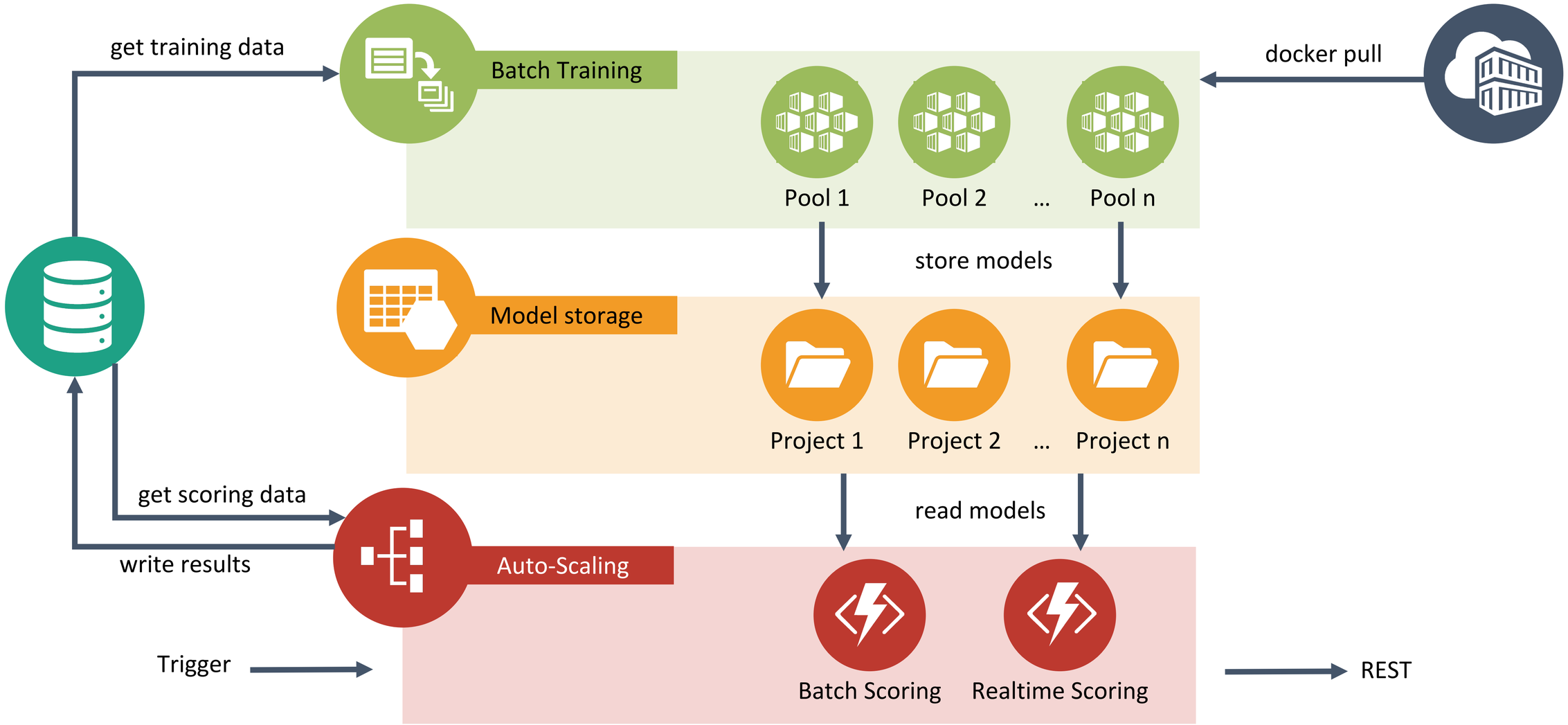
Rare picture of the fabled „eierlegende Wollmilch“

The Problem

“How can we build a cost effective data science pipeline that allows data scientists using R to easily put their models into production, that scales well and is cheap and easy to maintain?”

What we want

a serverless data science architecture



Agenda

01
The Problem

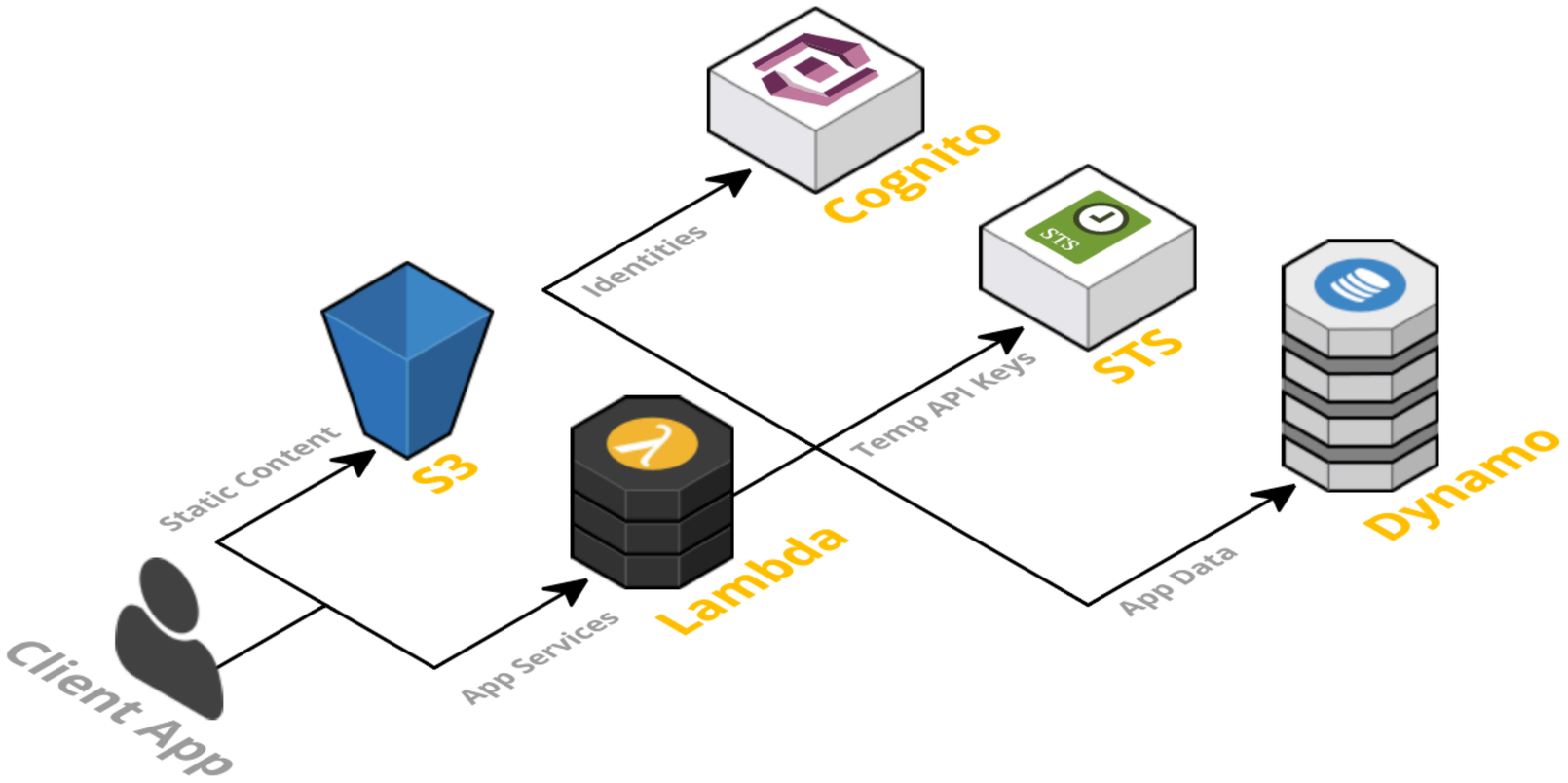
Building a scalable and flexible pipeline to deploy R models

02
Serverless

What does this buzzword actually mean?

03
Architecture

A solution architecture for Azure



Components of a serverless architecture

The Solution

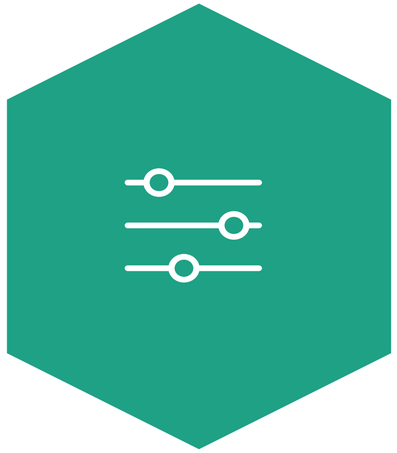
“Just like wireless internet has wires somewhere, serverless architectures still have servers somewhere.

What ‘serverless’ really means is that, as a developer you don’t have to think about those servers. You just focus on code.”

serverless.com

Why serverless?

The promise: Focus on coding, not maintenance



NO ADMINISTRATION

No server provisioning and maintenance is necessary. Hardware and OS are abstracted away.

SCALE ON DEMAND

Scaling is automatic and part of the service.

PAY-PER-USE

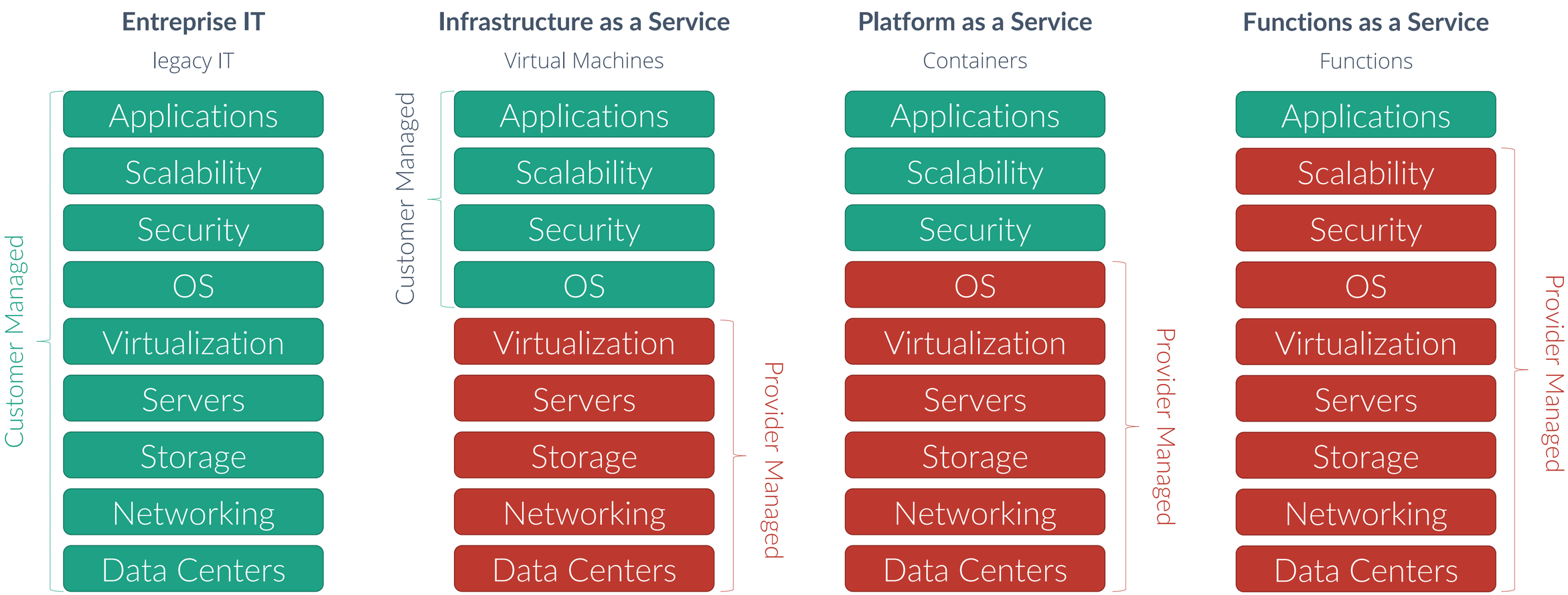
Billing is based on actual compute resources used. No compute used, no costs.

FASTER TURNAROUND

Spinning up new environments is quick and allows for faster experimentation.

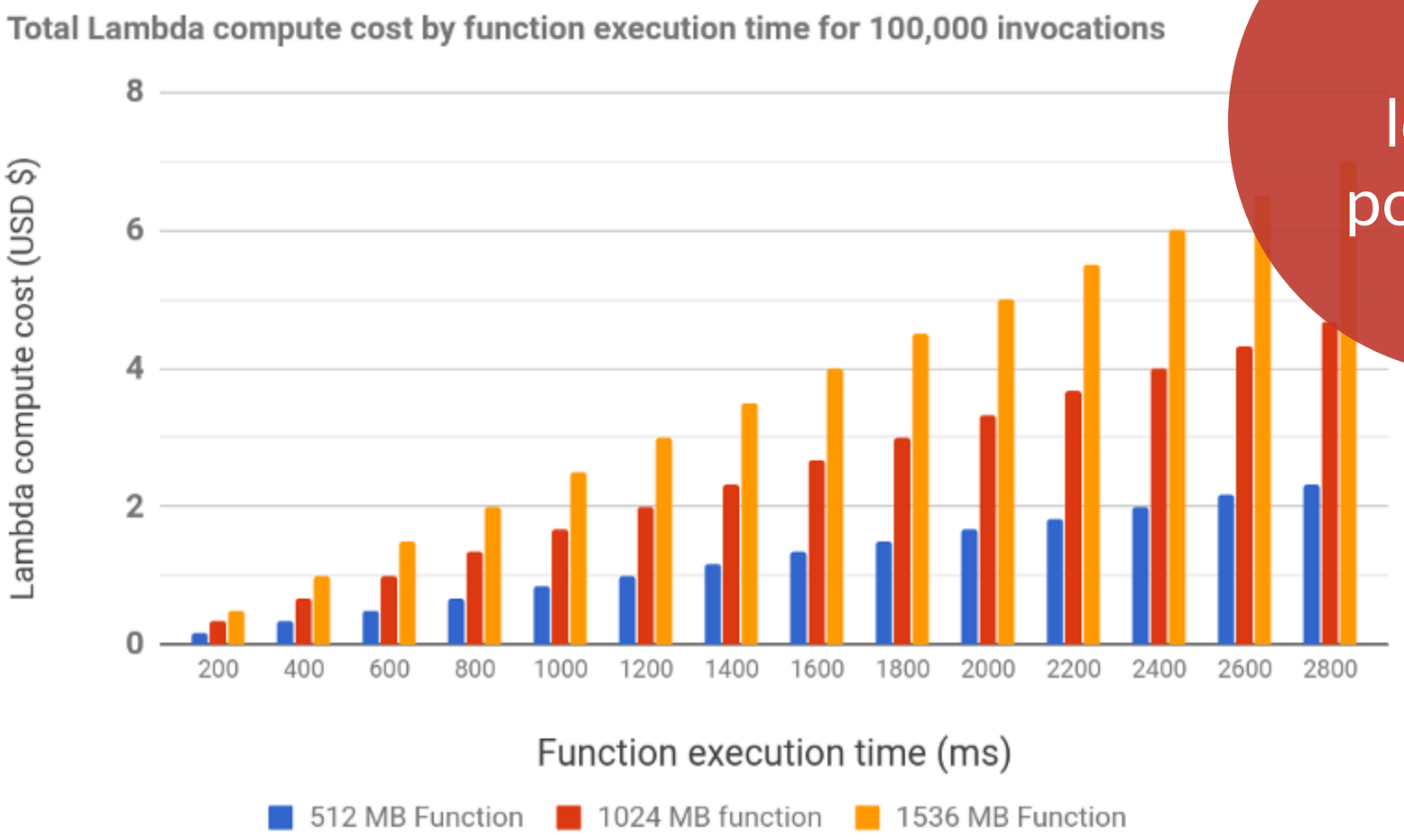
The Evolution of the Cloud

Cloud provider versus customer roles for managing cloud services



Cost Comparison

Serverless can be cheap, but depends on work load



Big lock-in potential!

Agenda

01
The Problem

Building a scalable and flexible pipeline to deploy R models

02
Serverless

What does this buzzword actually mean?

03
Architecture

A solution architecture for Azure

Two Use Cases

Model training and scoring have different architecture requirements

TRAINING

- Usually long running tasks
- Resource intensive
- Mostly in batch mode

SCORING

- Mostly short running tasks
- Resource usage low
- Either adhoc or on schedule

OUR FOCUS TODAY

Serverless Options

We primarily looked at the following options:

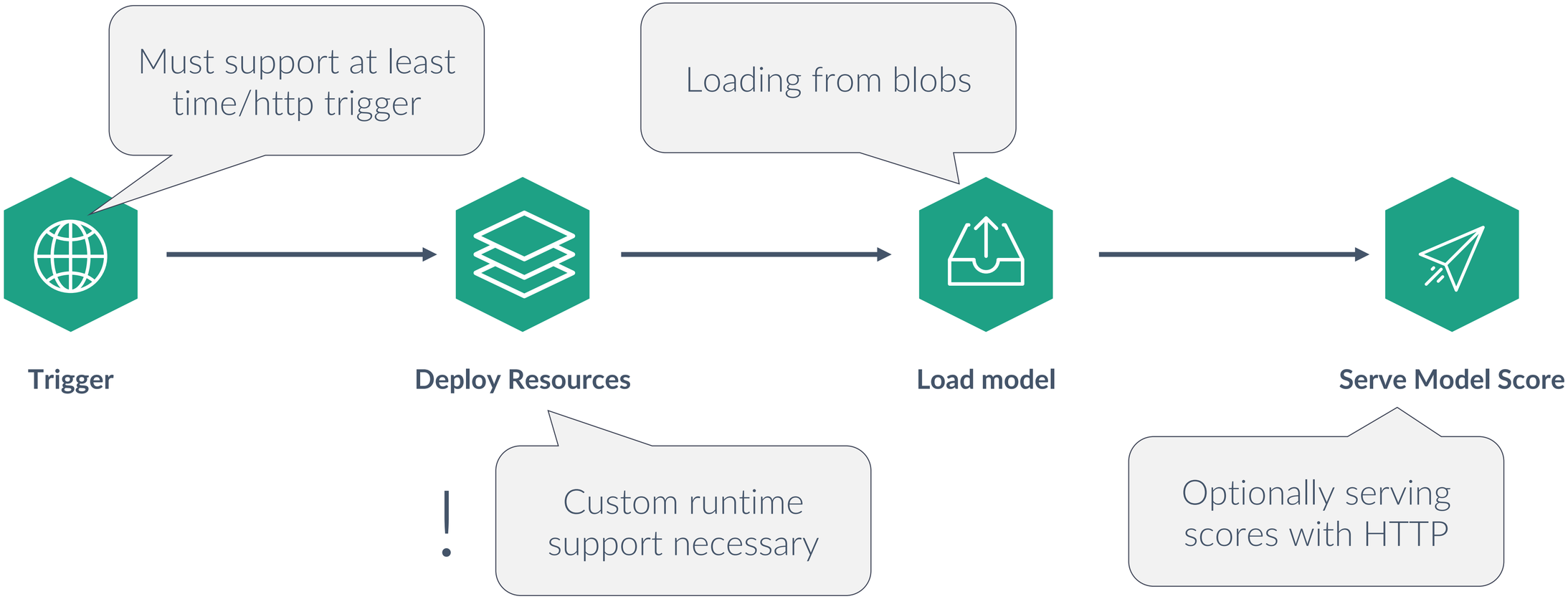
AWS Lambda

Azure Functions

Azure Container Instances

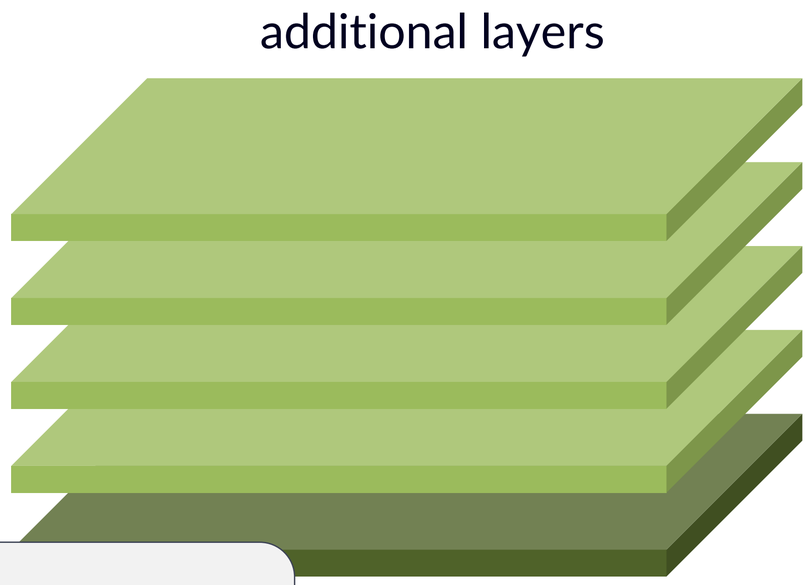
Requirements

Many ways to realize serverless scoring architecture with different pros and cons



Function as a Service

AWS Lambda




```
R/
└─library/
    ├─ package 1/
    ├─ package 2/
    ├─ package .../
    └─ package n/
```

```
R/
├─bin/
├─lib/
├─library/
└─...
bootstrap
runtime.r
```

packrat

ZIP runtime.zip

Philipp Schirmer

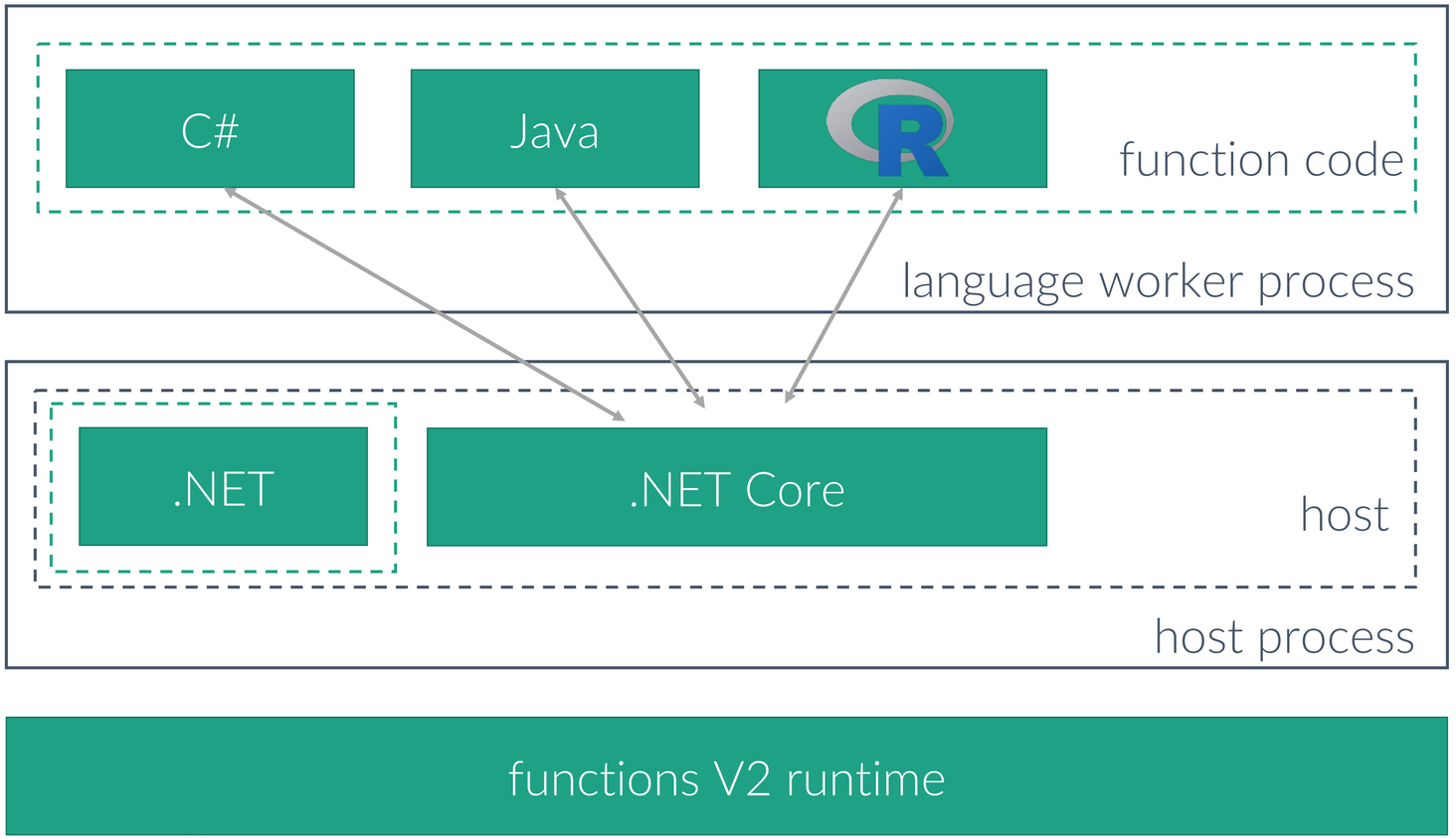
ZIP runtime.zip

Compiled packages can be a headache...

⚠ A function can use up to 5 layers at a time. The total unzipped size of the function and all layers can't exceed the unzipped deployment package size limit of 250MB.

Function as a Service

Azure Functions



gRPC

Neal Fultz

modern open source high performance RPC framework

Protocol Buffers

Dirk Eddelbuettel

Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data

Why Azure Container?

Container give us maximum flexibility regarding runtime and reduce vendor lock-in

PROS

- **Supports arbitrary runtimes**
- **No problems with compiled libraries**
- **Lots of supported triggers in combination with logic apps**
- **Low vendor lock-in**
- **Pay-as-you-go**

CONS

- **More setup involved compared to FaaS such as AWS Lambda**
- **Higher startup times compared to FaaS depending on Image**

Azure Container + Logic App

Our setup currently looks like this

01 Logic App

Logic App implements trigger (time/event) and spawns Container Instances

02 Container Instances

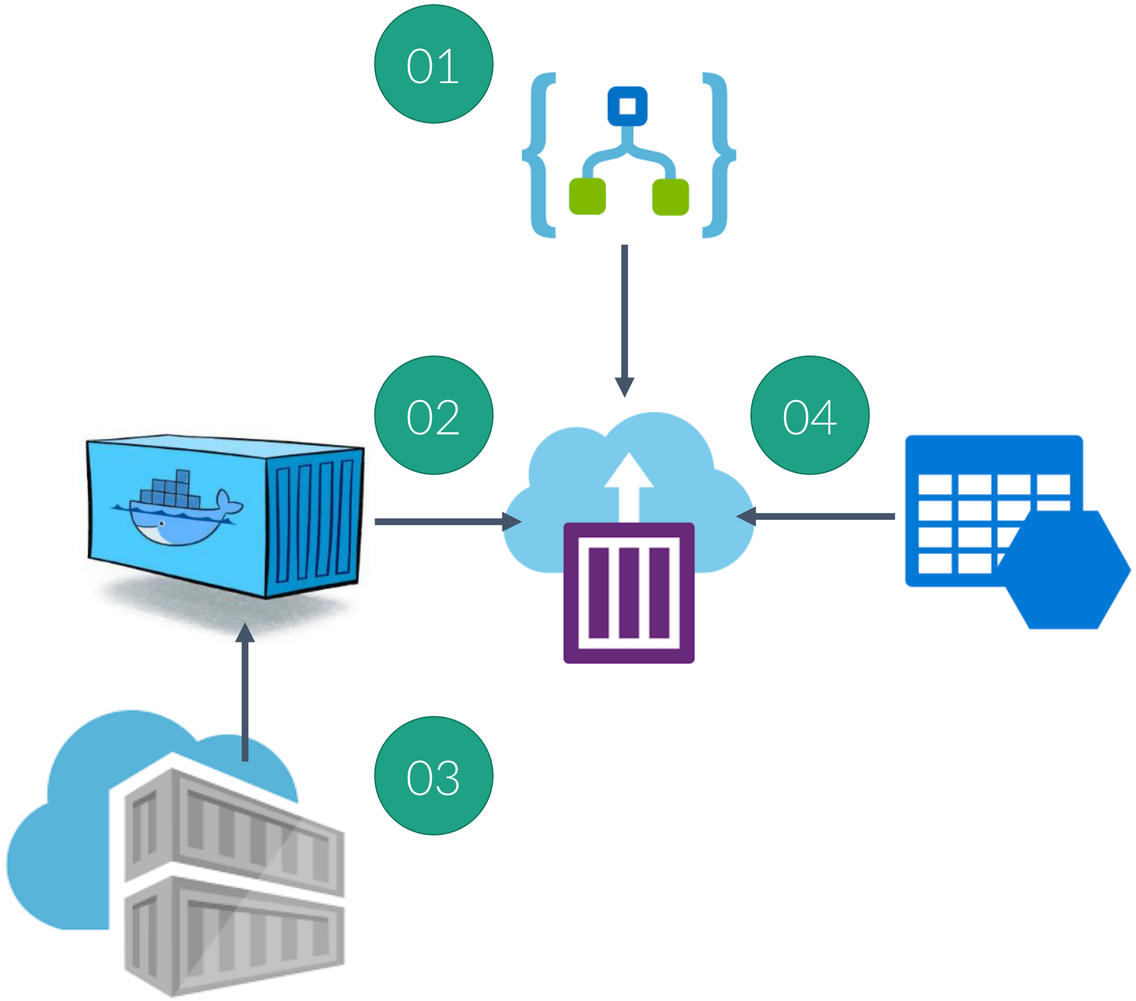
Container Instances pulls Docker image from Azure Container Registry or other Container Registry

03 Container Registry

Model scoring code in Docker image gets pulled to ACI

04 Blob Store

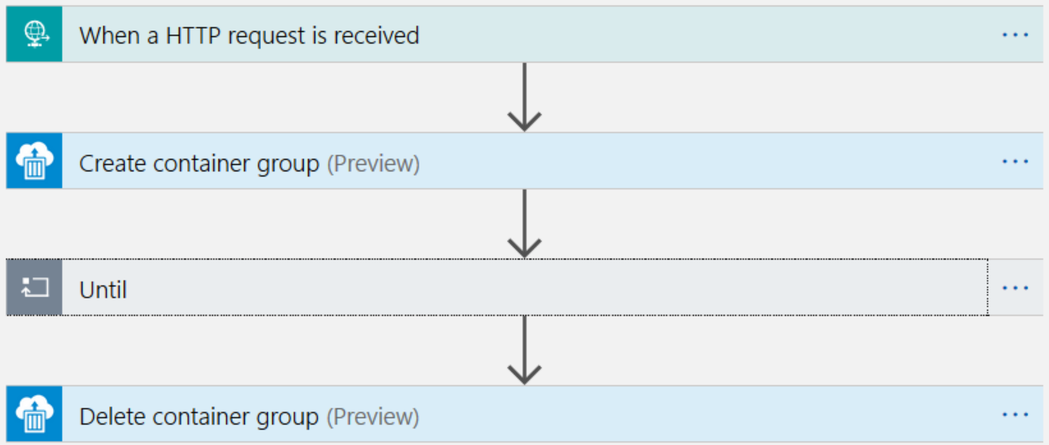
Load serialized models for scoring from blob storage



Logic App

A serverless workflow orchestration tool with GUI for prototyping

LOGIC APP DESIGNER

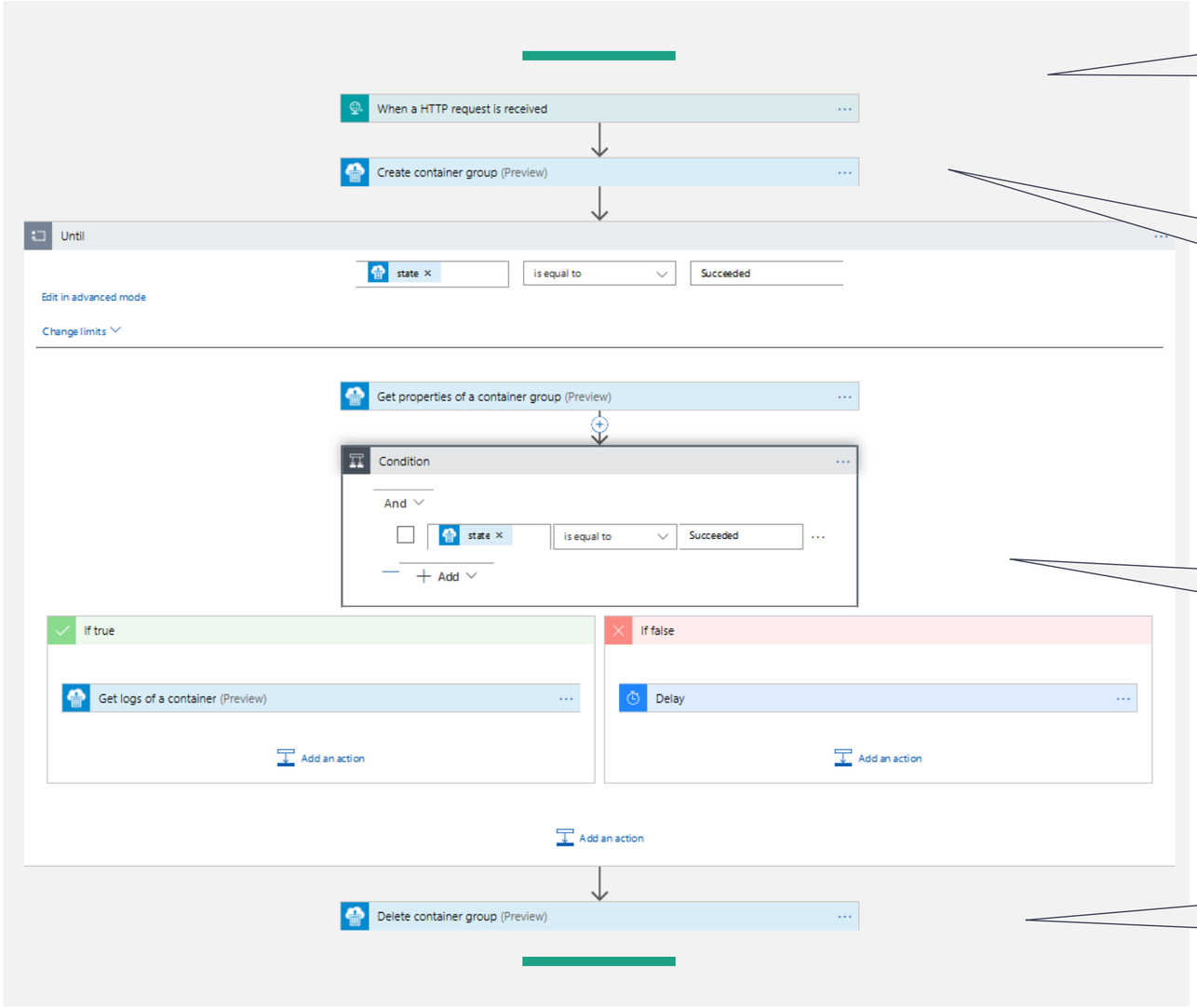


LOGIC APP TEMPLATING

```
{
    "$connections": {
        "value": {
            "aci": {
                "connectionId": "/subscriptions/<subscription>/resourceGroups/serverless/p
                "connectionName": "aci",
                "id": "/subscriptions/<subscription>/providers/Microsoft.Web/locations/wes
            }
        }
    },
    "definition": {
        "$schema": "https://schema.management.azure.com/providers/Microsoft.Logic/schemas/
        "actions": {
            "Create_container_group": {
                "inputs": {
                    "body": {
                        "location": "westeurope",
                        "properties": {
```

Scoring Workflow

Template workflow for a wide range of scenarios



Specify trigger type (time, HTTP, email, etc)

Create container resources based on spec (cpu + RAM + nodes)

Check if container group is spawned successfully

Delete container after work is done

serveRless Package

We want to build a package to help automate this setup

IDEA

Build an R package that allows R users to deploy their code in a serverless setup

STATUS

Build prototype to test setup

Build Rstudio Addin

R Package serveRless

Many thanks to Hong Ooi for his awesome work supporting R in Azure!

Questions?

Thank you
for your attention!

Feel free to reach out to us:

linkedin.com/in/christoph-bodner

linkedin.com/in/thomas-laber

What now?

VMs vs containers vs functions

	Virtual Machines	Containers	Functions
Unit of Scale	machine	application	function
Abstraction	hardware	operation system	language runtime
Packaging	image	container file	code
Configure	machine, storage, network, OS	servers, applications, scaling	run code when needed
Execution	multi-threaded, multi-task	multi-threaded, single-task	single-threaded, single-task
Runtime	hours to months	minutes to days	microseconds to seconds
Unit of Cost	per VM per hour	per container per hour	per memory/second per request
Amazon	EC2	Fargate	Lambda
Azure	Azure VM	Container Instances	Azure Functions
Google	Google Compute Engine	Google Kubernetes	Cloud Functions