

# serveRless

## computing for R

.....

**useR! 2019 – Toulouse**

by Christoph Bodner & Thomas Laber





# Hey, I am Christoph

Likes that  $\pi \cdot z^2 \cdot a$  is a food that approximates its own volume\*\*

Sound of Music is here

I am from a small town called Lienz in the Tyrolian Alps, but live in Vienna now

My mother tongue is Tyrolian, the most beautiful Austrian dialect according to a study:)\*

\*n=1



This is me😊

**Group Lead Analytics  
BILLA Supermarkets**

Before that I build the Analytics Team @ the Austrian Postal Service and was a Financial Consultant at KPMG

\*\* I quote that fact a lot, because come on, this is fu\*\*ing awesome:D



# Hi, my name is Thomas

Thinks pizza & coke should be part of any proper diet (got a Xmas present from a pizza chef)

Christoph won't admit it, but Carinthia is actually way nicer than Tyrol, I mean look at this picture!

I am originally from Villach, but also live in Vienna nowadays



Thomas Laber

**Senior Data Scientist**  
**Austrian Postal Service**

Before joining Austrian Post, I worked for  
Accenture and Detecon Consulting

# 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



# The Problem

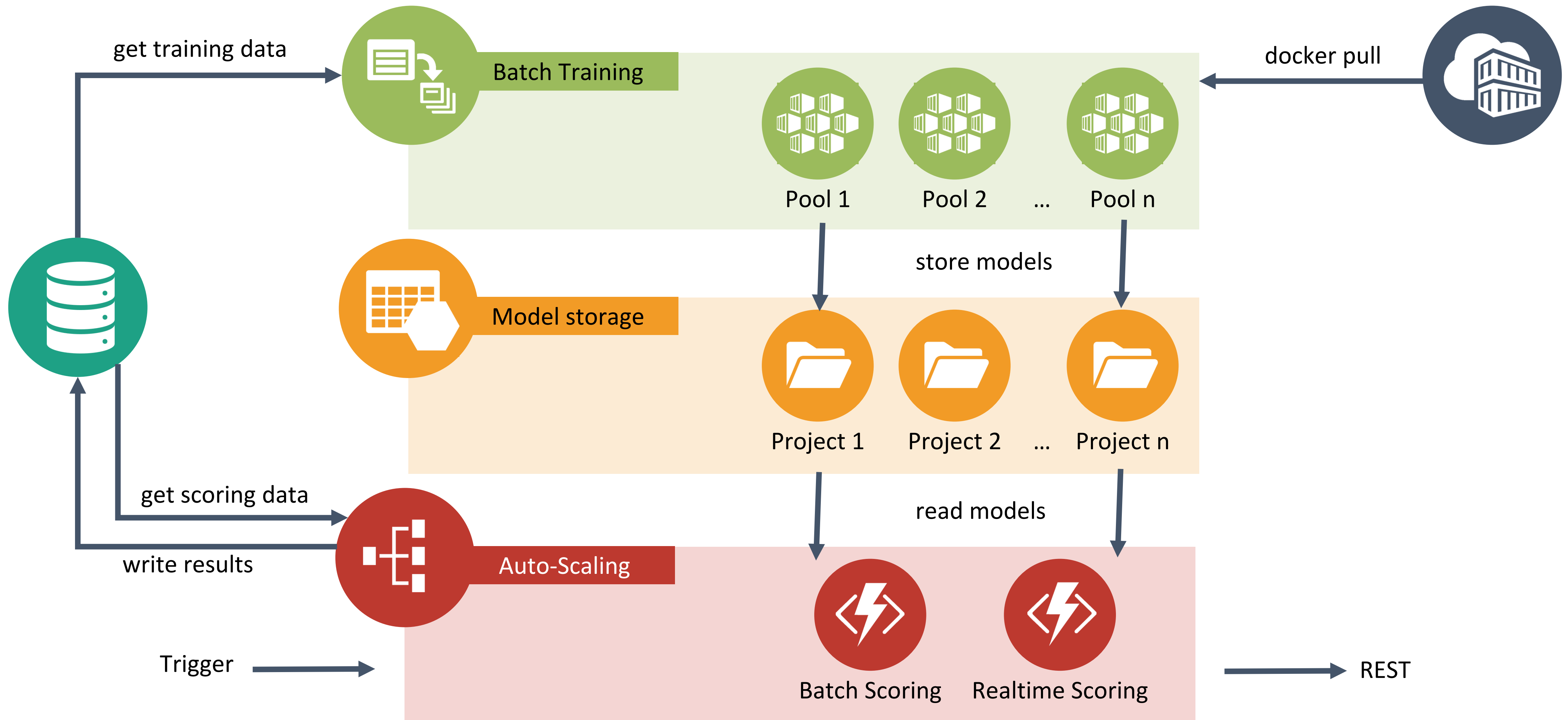
“How can we build a cost effective data science pipeline that allows data scientists using R to easily put their models into production, that scales well and is cheap and easy to maintain?”



Rare picture of the fabled „eierlegende Wollmilch“

# 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



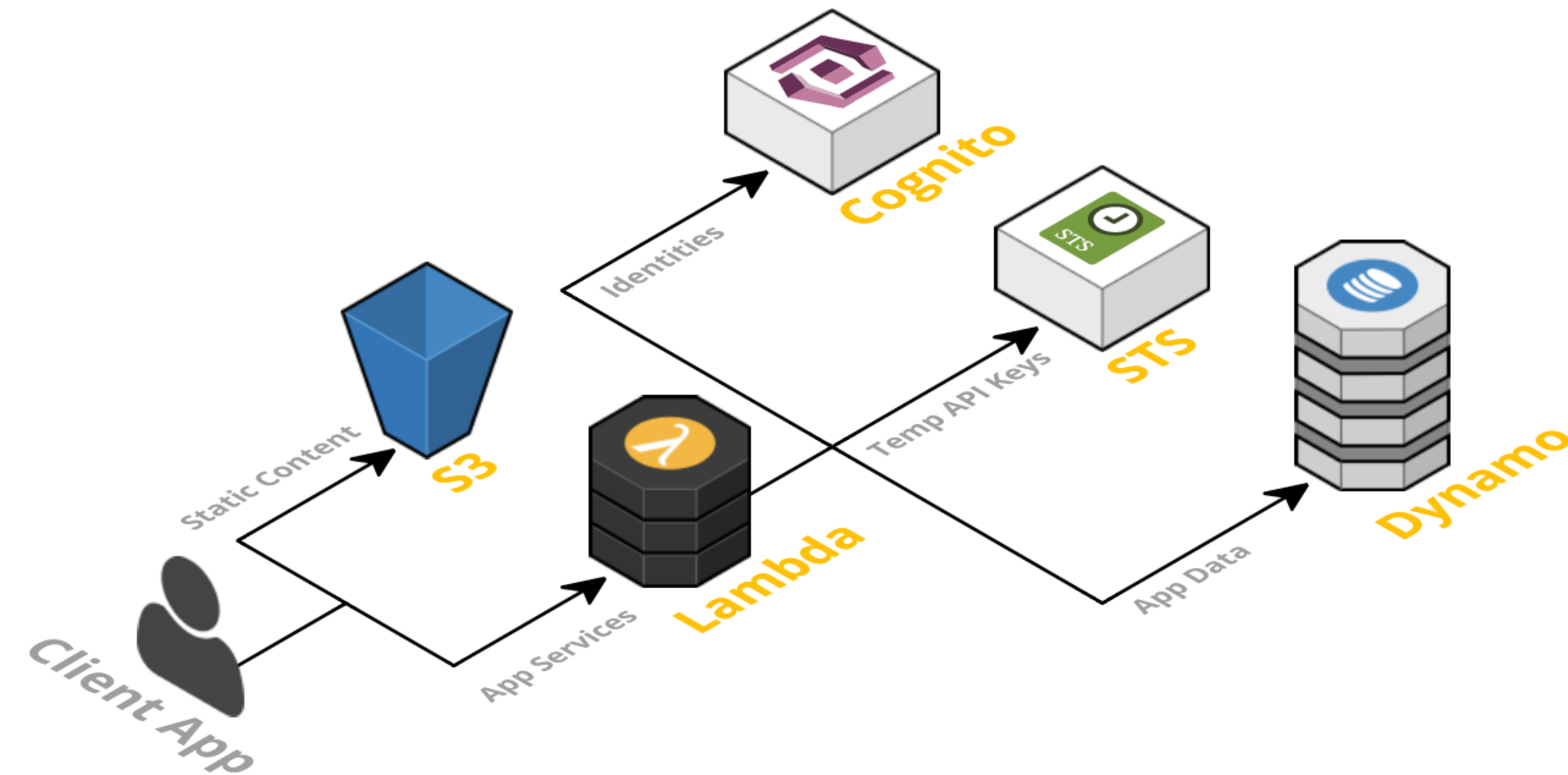
# The solution

“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](https://serverless.com)

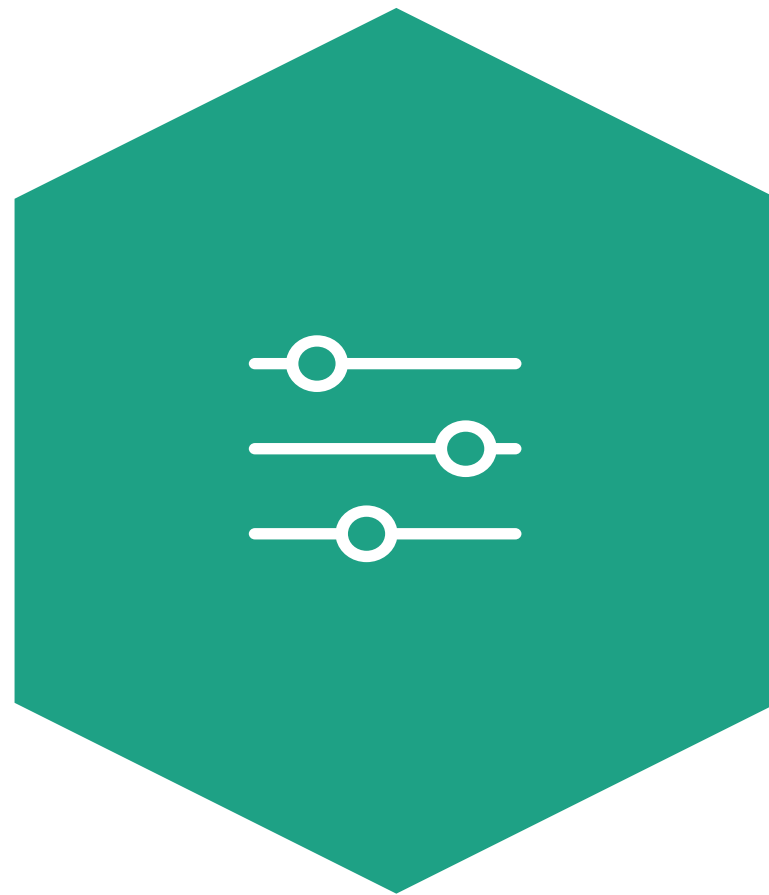


Components of a serverless architecture

# Why serverless?

The promise: Focus on coding, not maintenance

---



## NO ADMINISTRATION

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



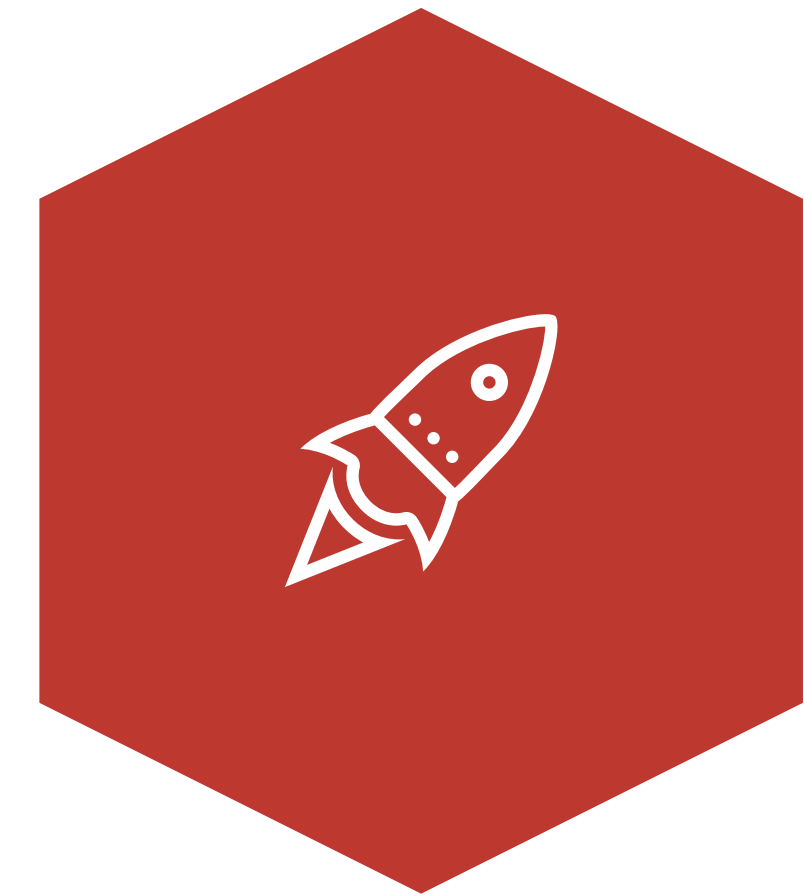
## SCALE ON DEMAND

Scaling is automatic and part of the service.



## PAY-PER-USE

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

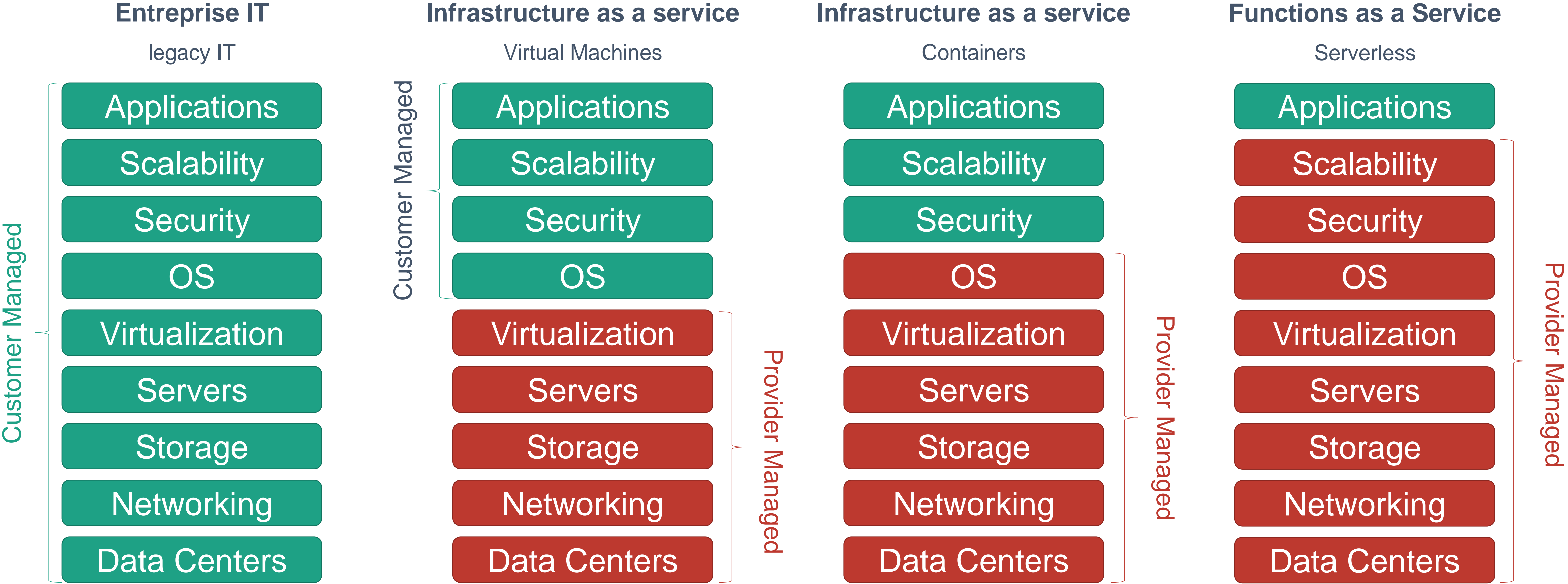


## FASTER TURNAROUND

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





# What now?

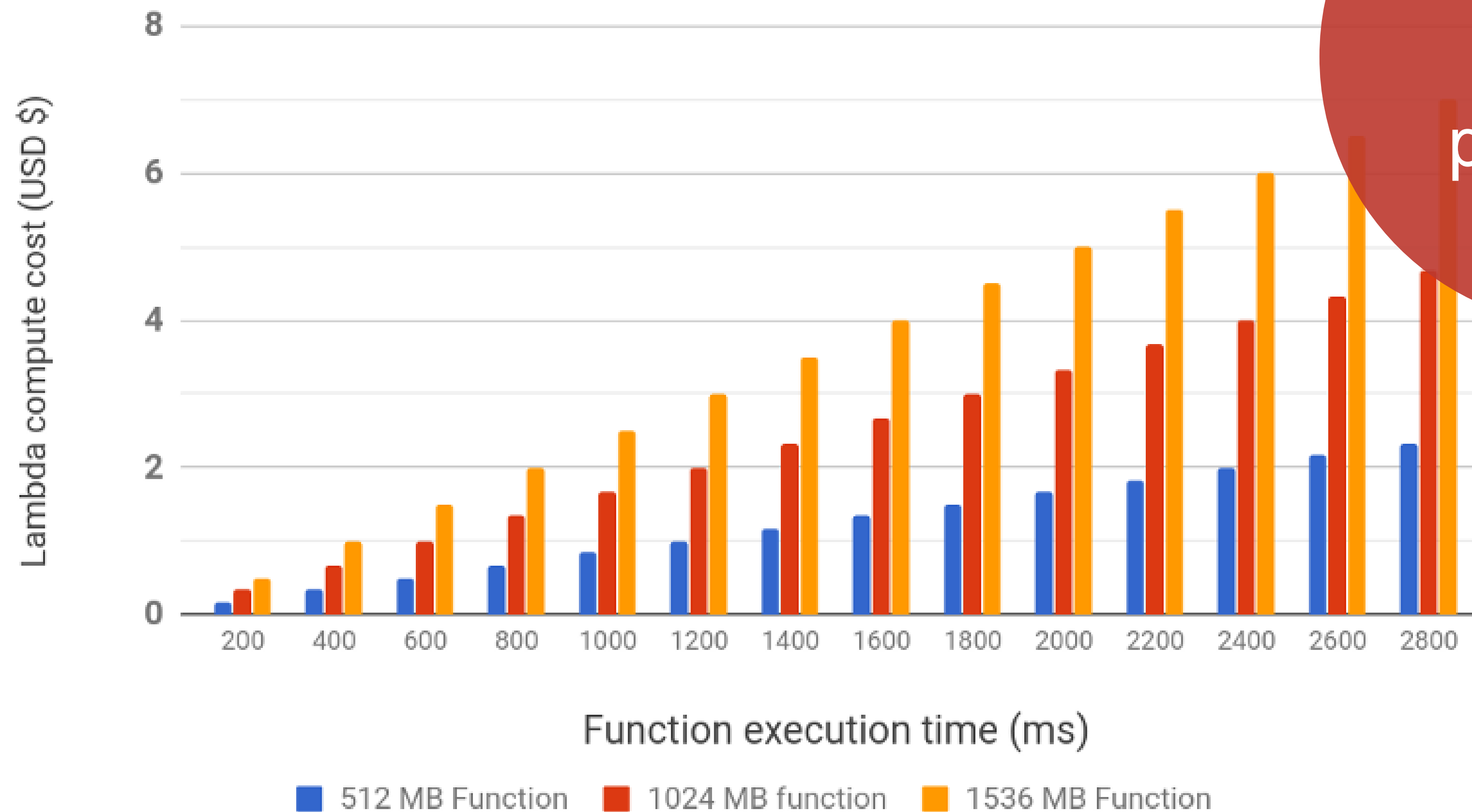
VMs vs containers vs functions

	Virtual Machines	Containers	Serverless
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 VM per hour	per memory/second per request
Amazon	EC2	Fargate	AWS Lambda
Azure	Azure VM	Container Instances	Azure Functions
Google	Google Compute Engine	Google Kubernetes	Cloud Functions

# Cost Comparison

Serverless can be cheap, but depends on work load

Total Lambda compute cost by function execution time for 100,000 invocations



Huge  
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:

---



Azure Container  
Instances



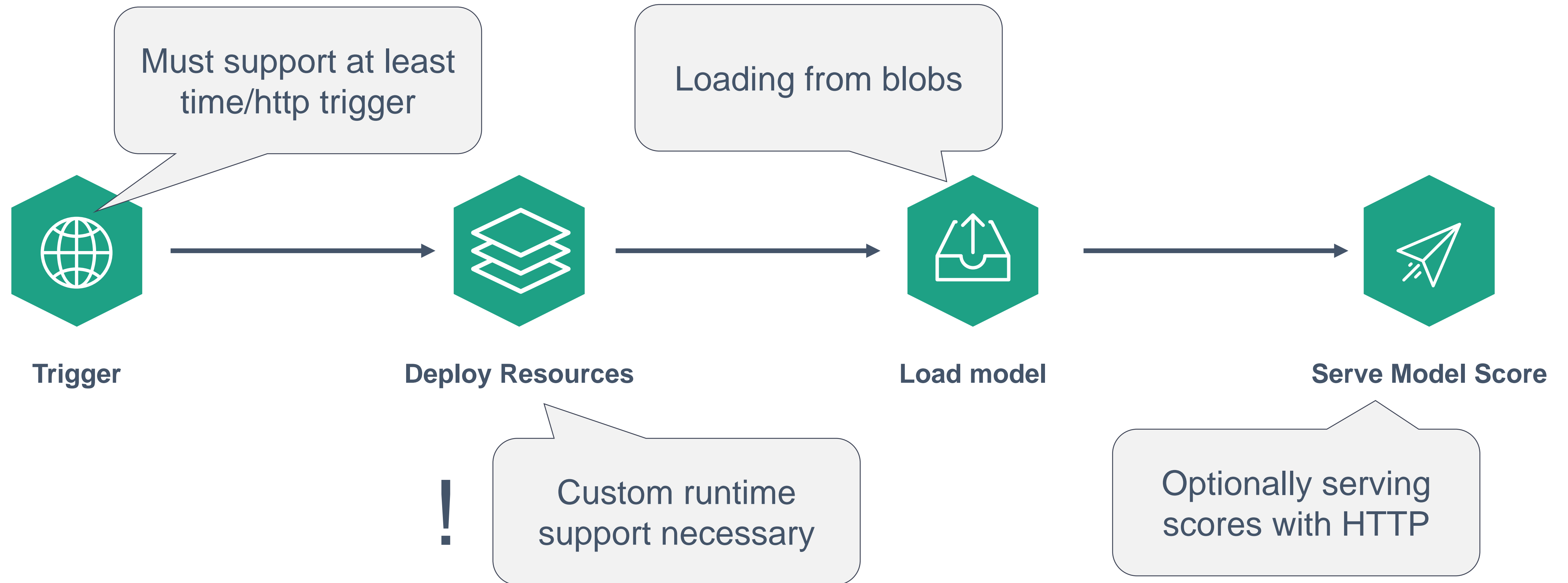
Azure Functions



AWS Lambda

# Requirements

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



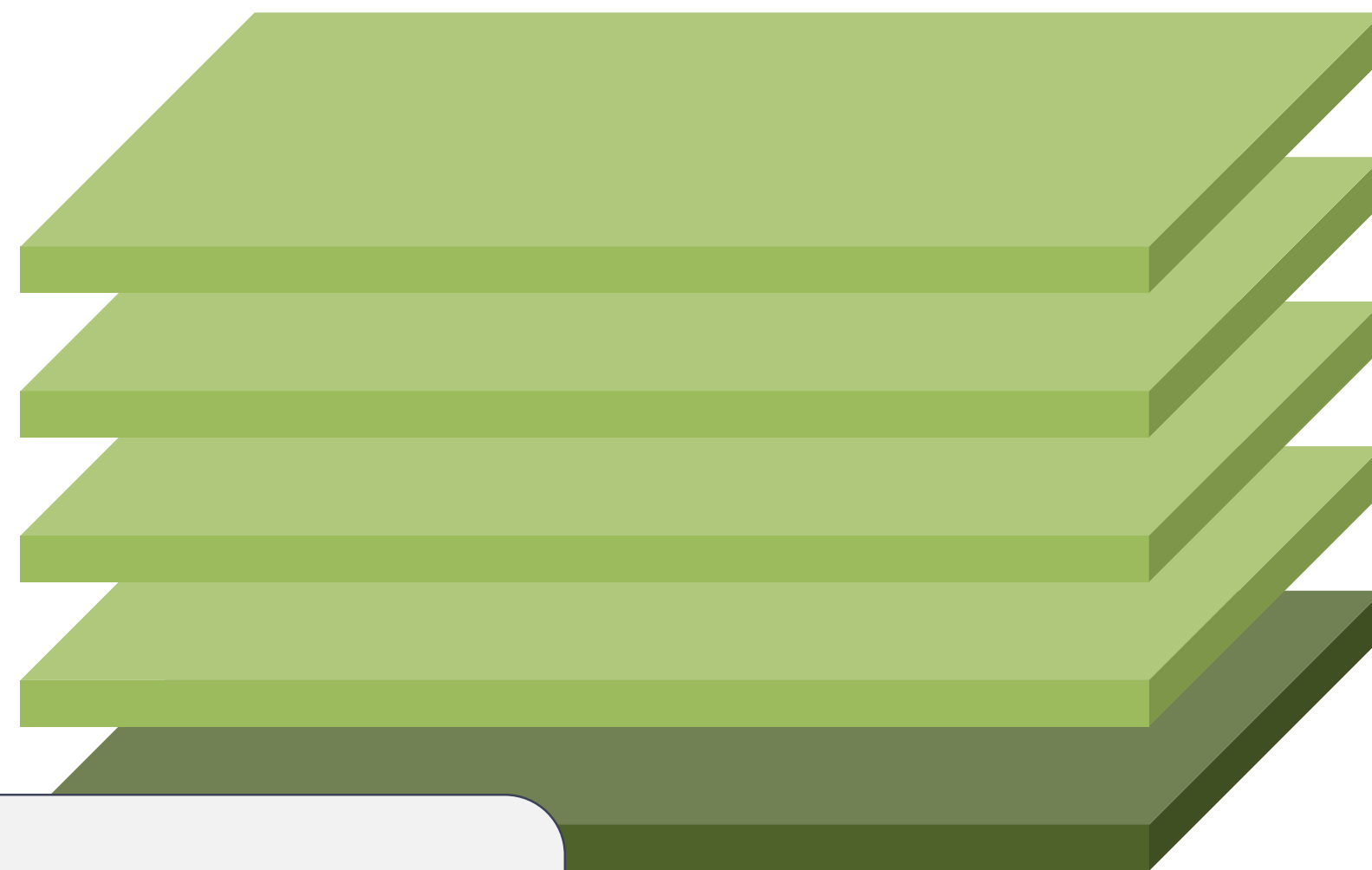


# Function as a Service

AWS Lambda



additional layers



base layer

Compiled packages  
can be a headache...

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



runtime.zip

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



runtime.zip

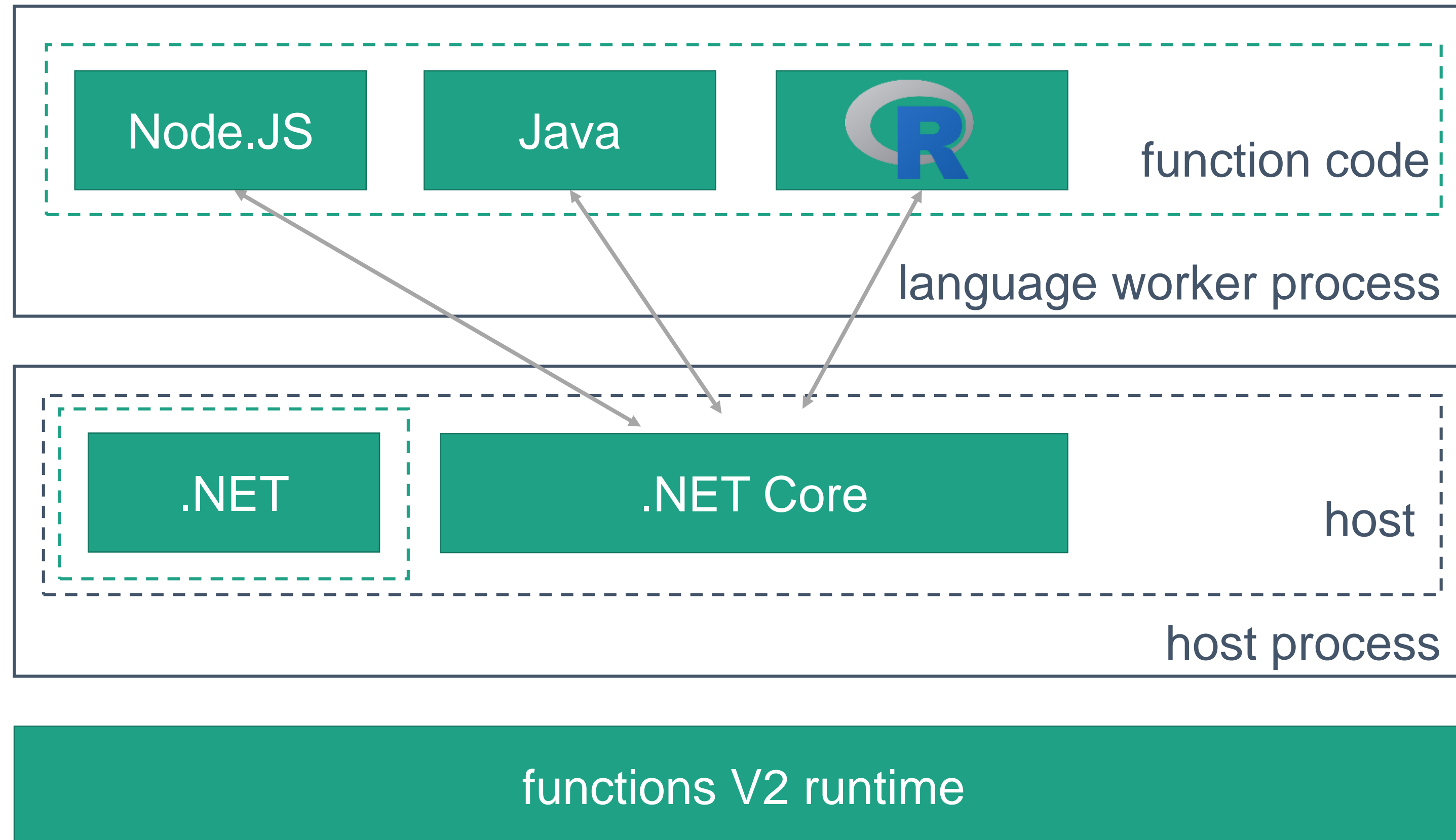
50 MB Limit



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



Neal Fultz

modern open source high performance RPC framework



Dirk Eddebuettel

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

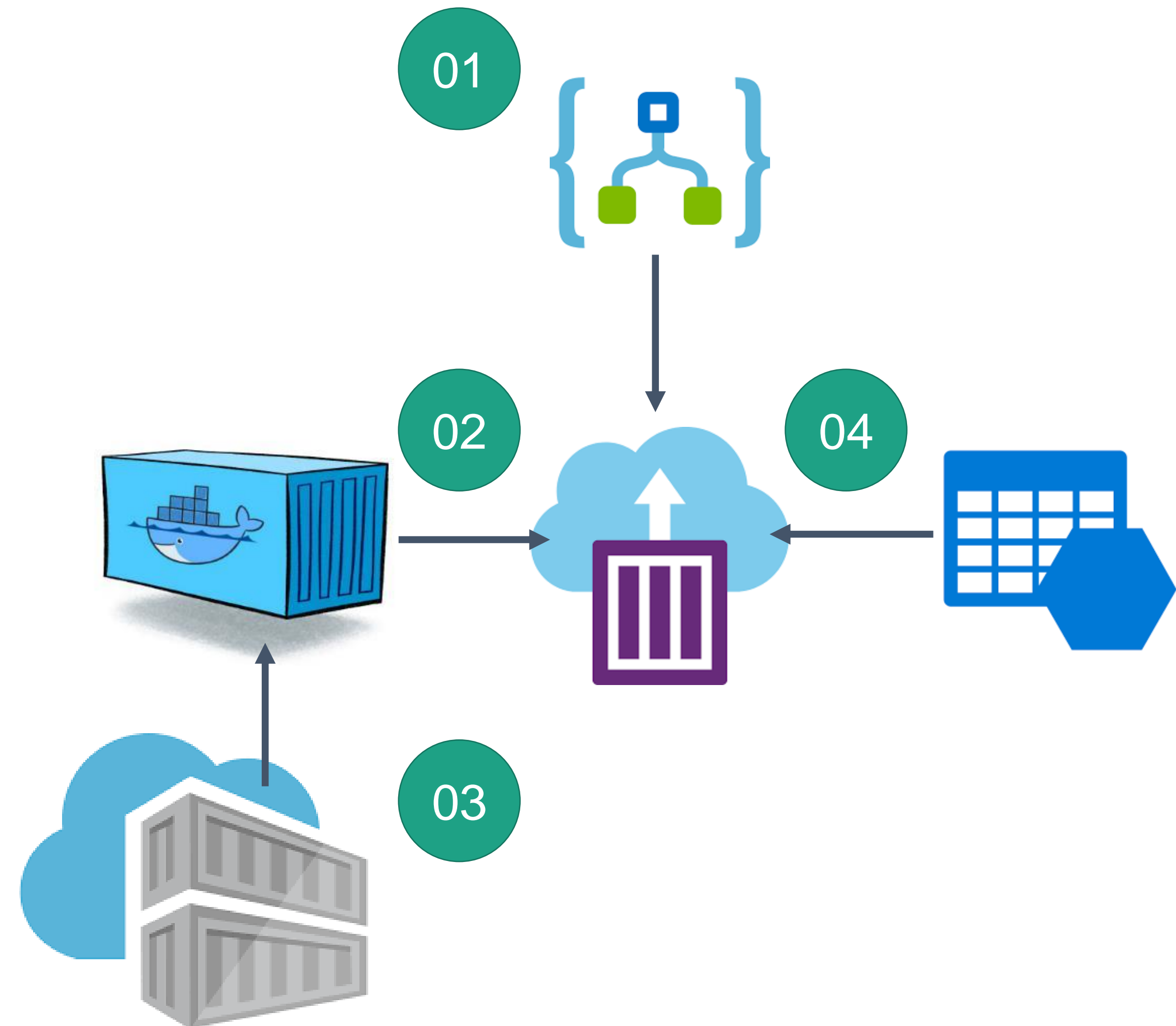
Container Instances pulls Docker image from Azure Container Registry or other Container Registry

## 03 Container Registry

Model scoring code in Docker image gets pulled to ACI

## 04 Blob Store

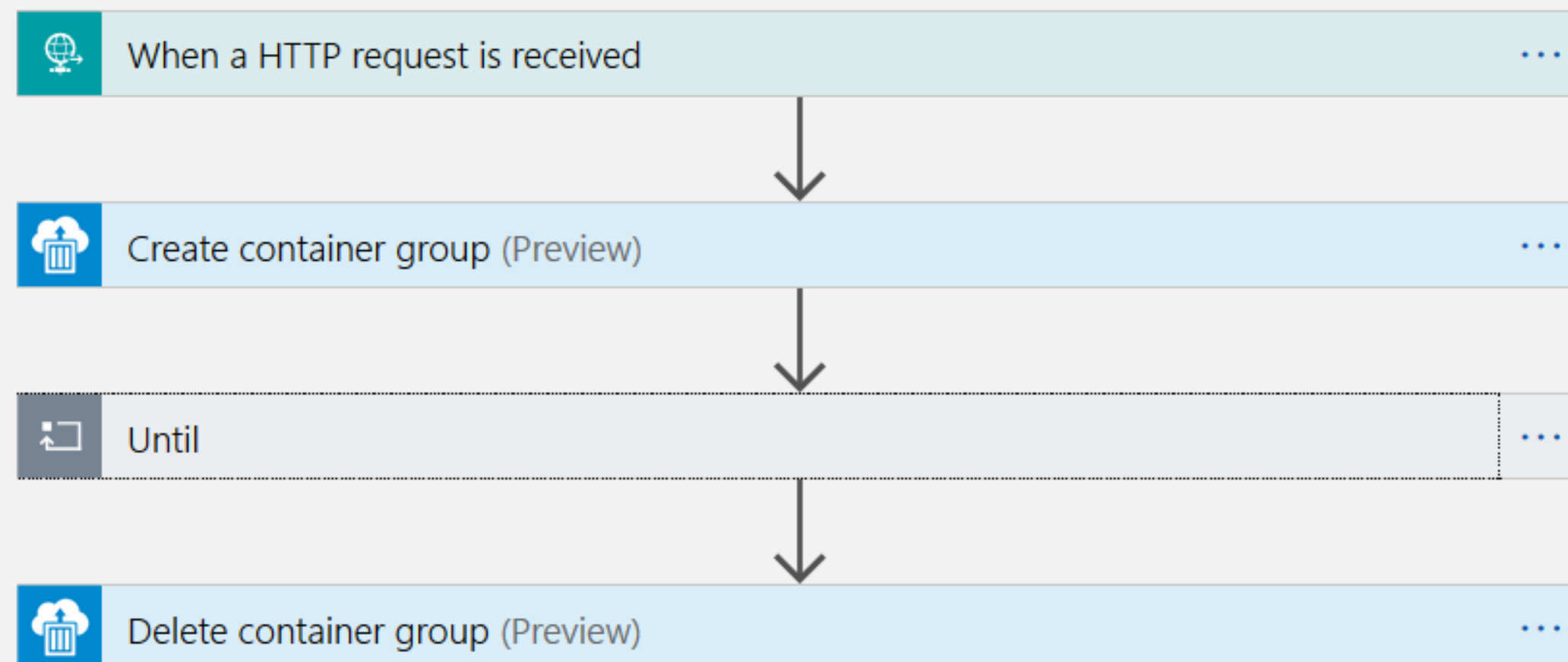
Load serialized models for scoring from blob storage



# Logic App

A serverless workflow orchestration tool with GUI for prototyping

## LOGIC APP DESIGNER

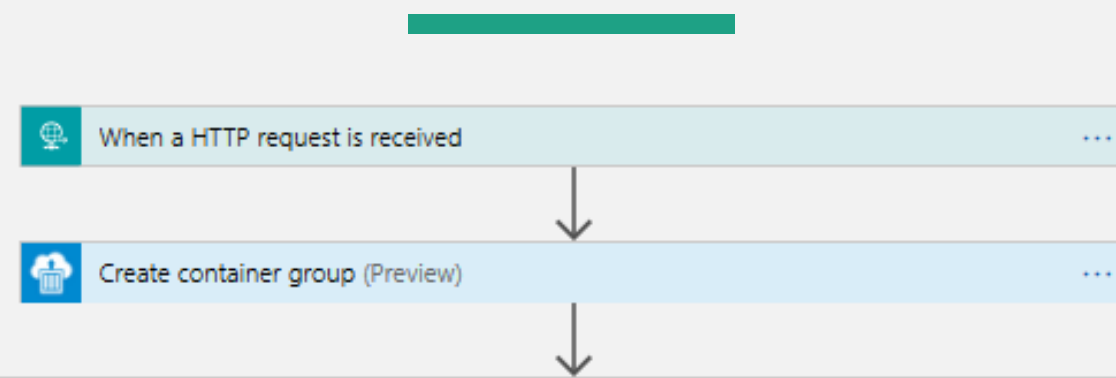


## LOGIC APP TEMPLATING

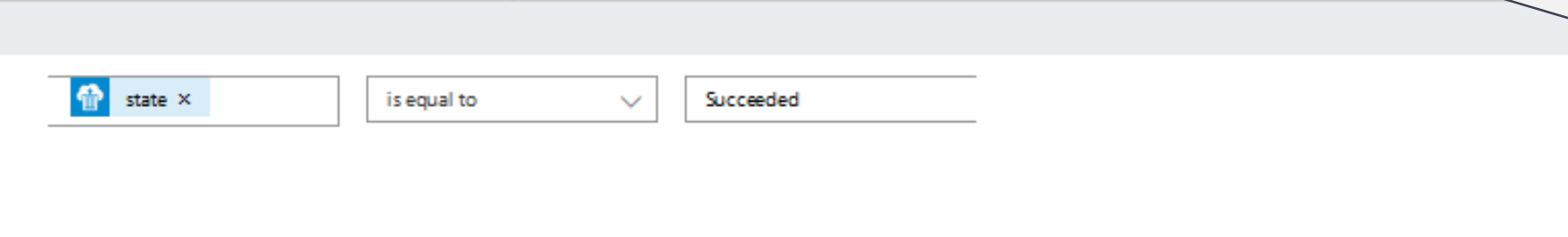
```
1 {
2   "$connections": {
3     "value": {
4       "aci": {
5         "connectionId": "/subscriptions/<subscription>/resourceGroups/serverless/p
6         "connectionName": "aci",
7         "id": "/subscriptions/<subscription>/providers/Microsoft.Web/locations/wes
8       }
9     }
10  },
11  "definition": {
12    "$schema": "https://schema.management.azure.com/providers/Microsoft.Logic/schemas/
13    "actions": {
14      "Create_container_group": {
15        "inputs": {
16          "body": {
17            "location": "westeurope",
18            "properties": {
```

# Scoring Workflow

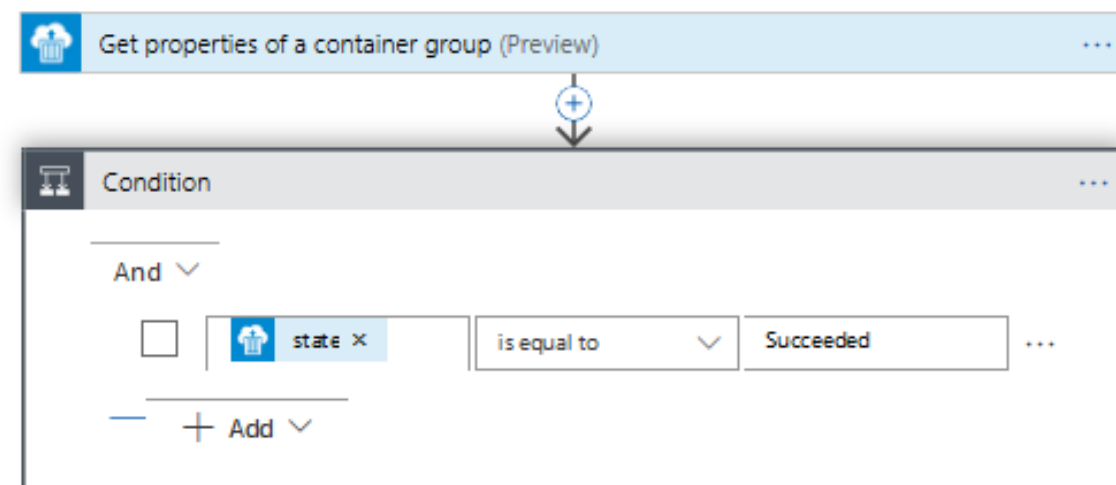
Template workflow for a wide range of scenarios



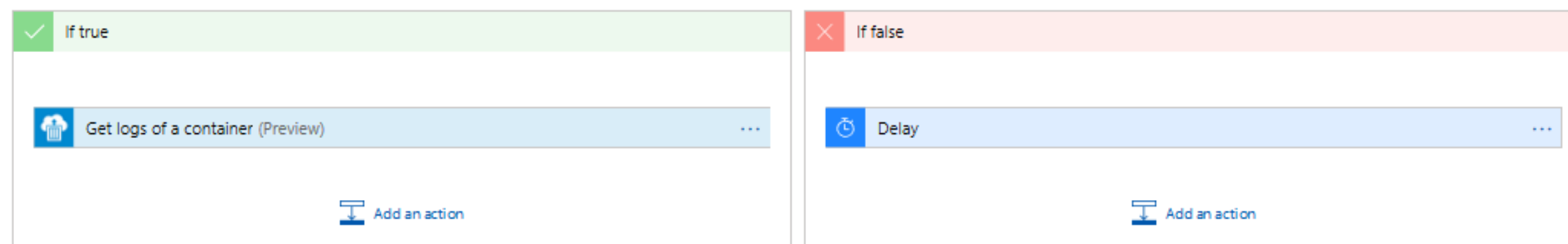
Specify trigger type (time, HTTP, email, etc)



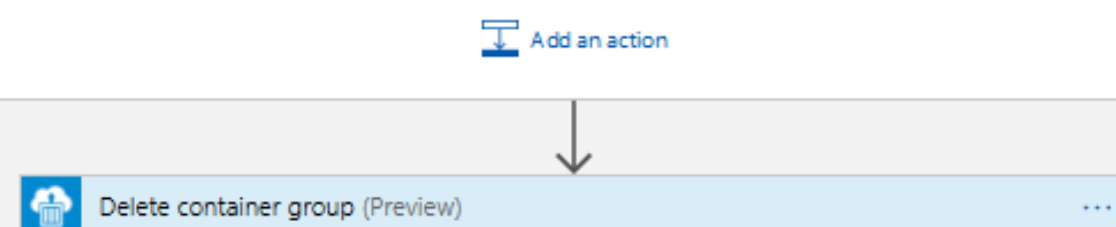
Create container resources based on spec (cpu + RAM + nodes)



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

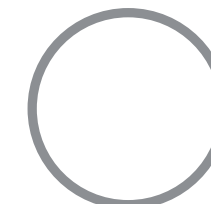
IDEA

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

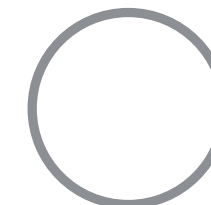
STATUS



**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](https://www.linkedin.com/in/christoph-bodner)

[linkedin.com/in/thomas-laber](https://www.linkedin.com/in/thomas-laber)

