



Microsoft Azure Solution Summit

Gemeinsam Cloud neu denken

#SummlTup





Building great apps on Azure – using OpenSource



Dennis Zielke

Technical Lead Cloud Apps

dennis.zielke@microsoft.com



denniszielke



@denzielke

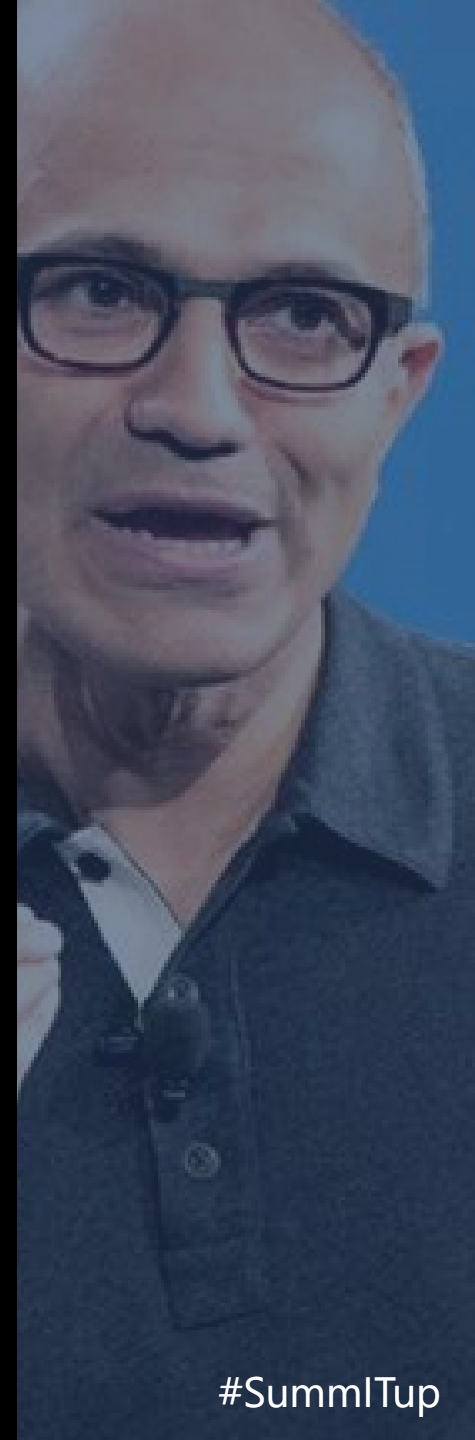


Questions we want to answer:

Why would you want to use open source?
How can open source technology help you?
How to decide which technology to use?

// Tech companies born with an open source mentality get it. It's our ability to work together that makes our dreams believable and, ultimately, achievable. We must learn to build on the ideas of others"

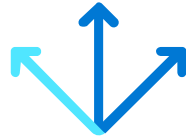
—Satya Nadella, CEO
Microsoft



Open Source Strategy on Azure



Choosing what works best for a customer scenario

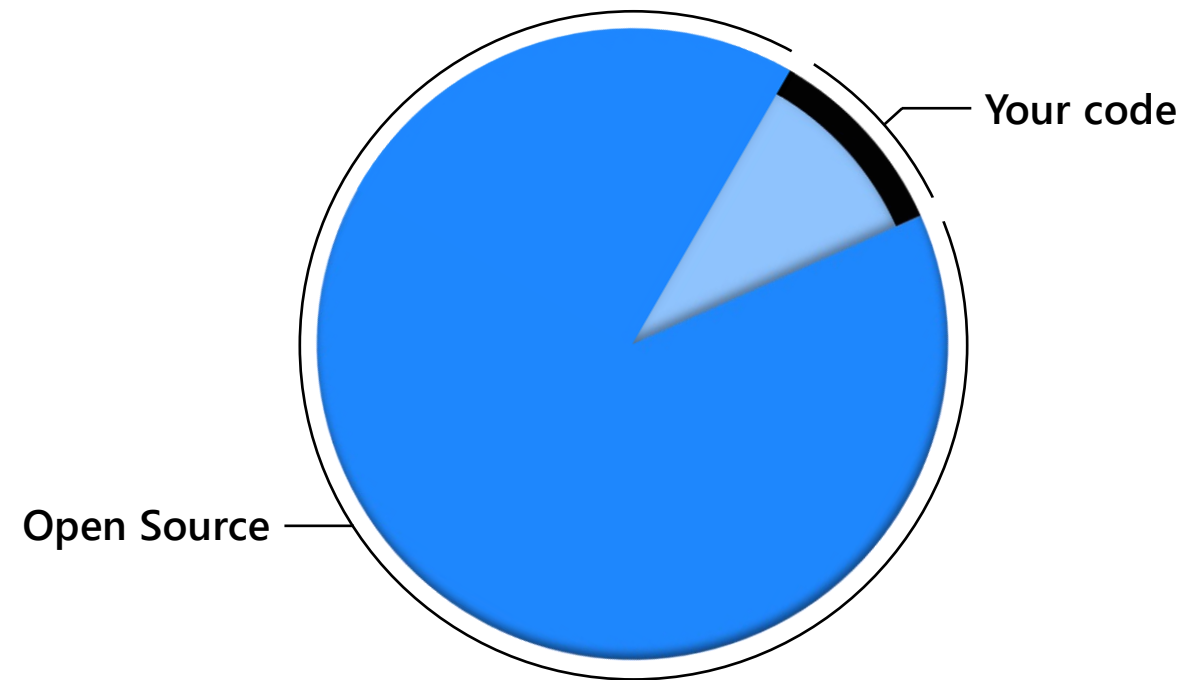


Contributions to open source initiatives and creating new ones

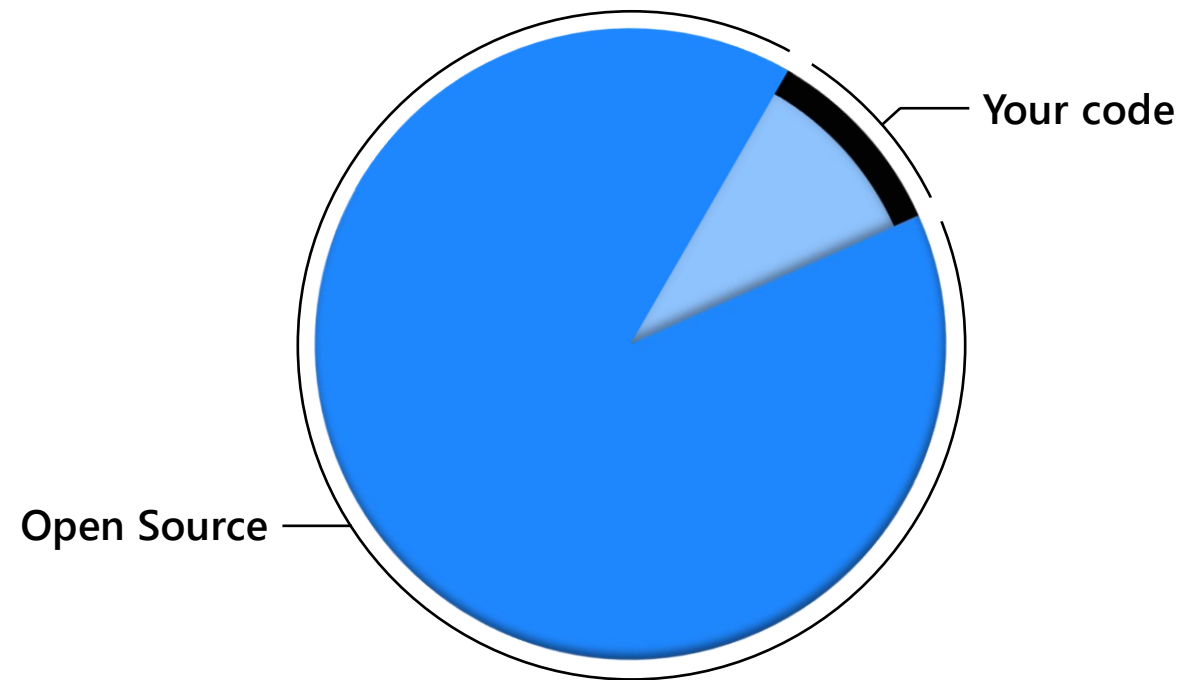


Enhancing managed services with open source solutions

In modern enterprise applications **90%** of the code comes from **open source**

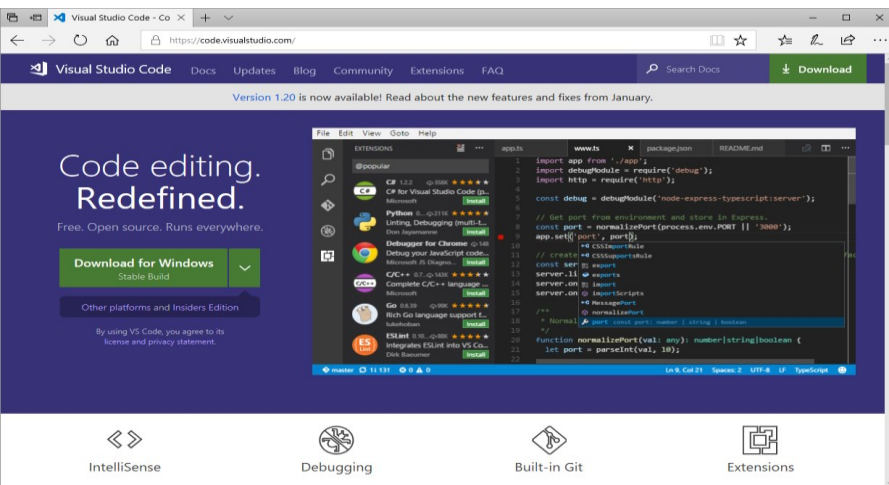


In modern enterprise applications **90%** of the code comes from **open source**

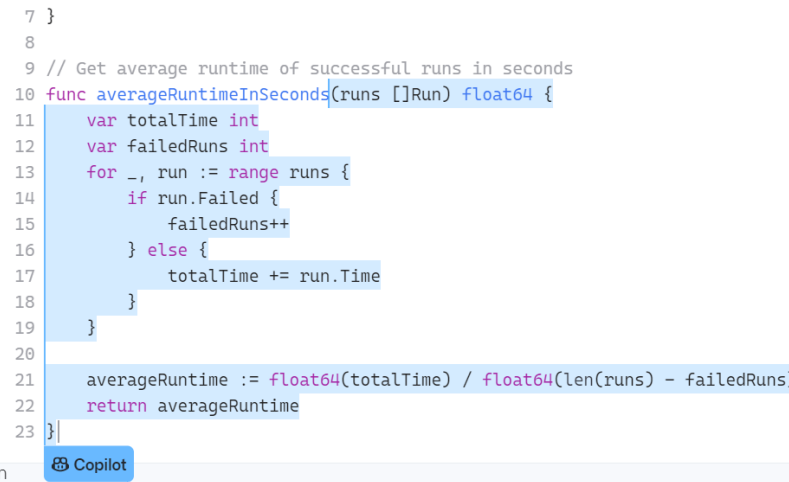


Most of that code lives on GitHub

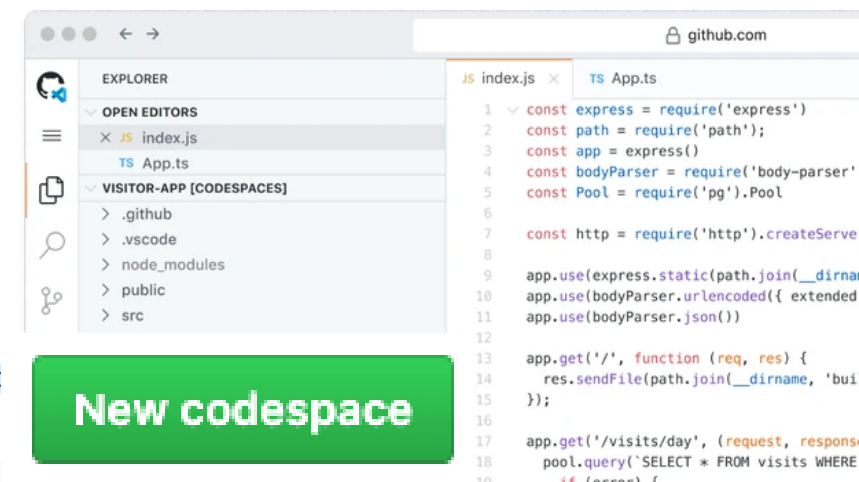
Innovations coming from GitHub to you



VSCoDe



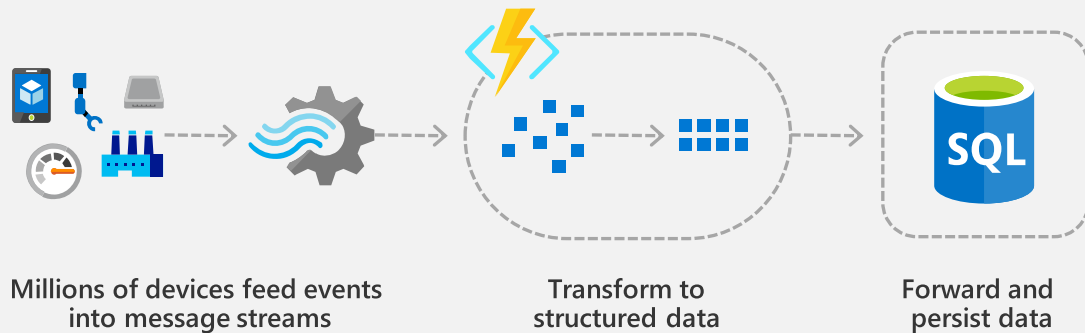
GitHub Copilot



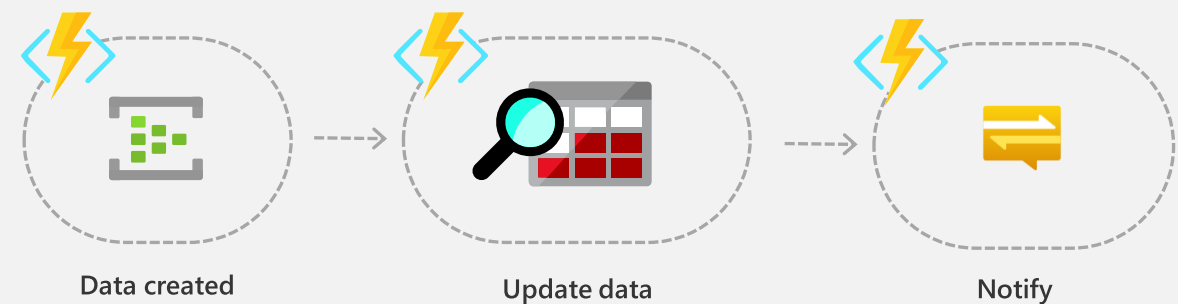
Codespaces

Typical application patterns in the cloud

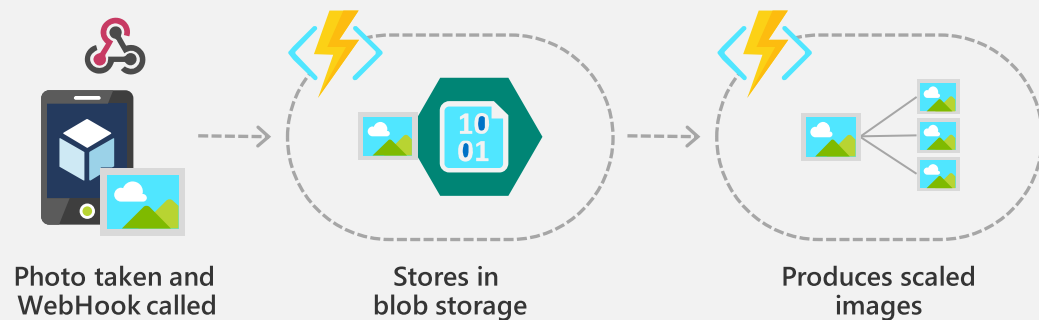
Event driven applications



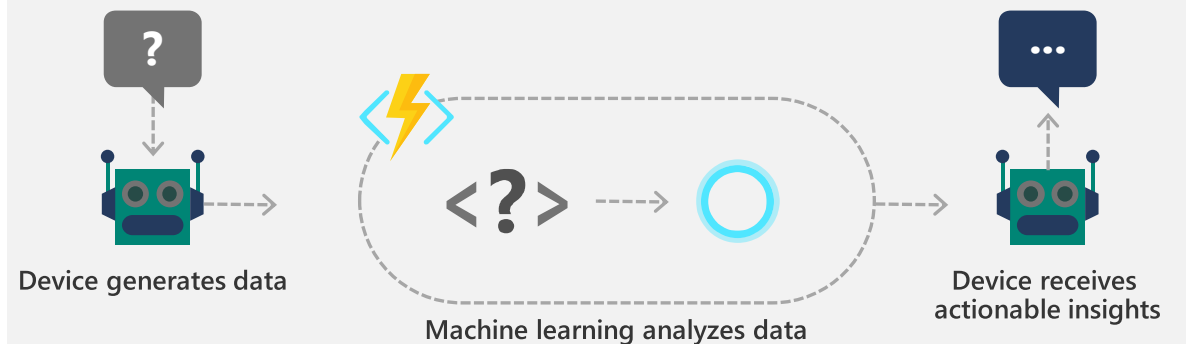
Microservice applications



Interactive applications



Real-time inferencing of insights



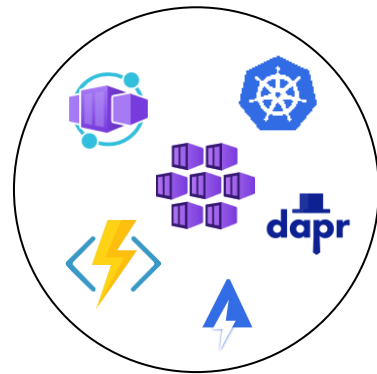
Let's talk about Developer problems

- Deciding on architecture patterns
- Selecting the right platforms
- Implementing continuous change
- Integrating managed services
- Securing applications and assets
- Automating Infrastructure deployments
- Solving Observability
- Ensuring governance

Building blocks for cloud applications



Code



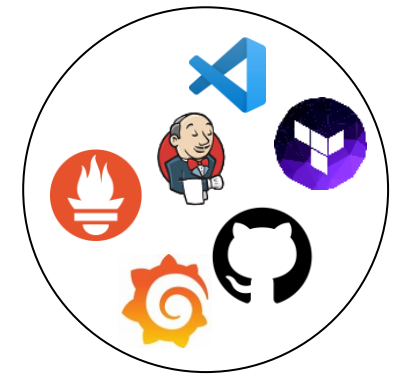
Runtime



State



Router



Tools

How should you design applications?



Dr Snooks
@drsnooks



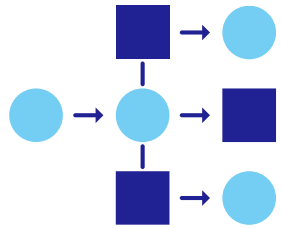
microservices (n,pl): **an efficient device for transforming** business problems into distributed transaction problems

14:27 · 06.10.15 · [TweetDeck](#)

123 Retweets **2** Quote Tweets **66** Likes



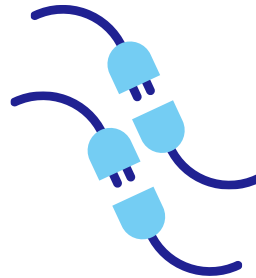
What is so hard about cloud native applications?



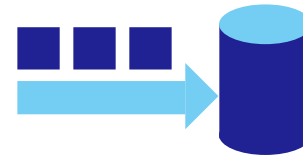
Designing boundaries, managing dependencies and changing them



Implementing and validating elastic scalability in your architecture



Empowering teams to make their own choices without increasing complexity



Keeping consistency and availability of data in check and integrating the right data store



Ensuring consistent security controls and governance

12 factor app principles (12factor.net)

1	Code Base	Single code base for each microservice, stored in its own repository, tracked with revision control, able to deploy to across environments (Dev, QA, Prod)	7	Port Binding	Each microservices is built within a container, typically exposed as HTTP on a web server, with all interfaces and functionality bound through ports, once again, providing isolation from other microservices.
2	Dependencies	Each microservice isolates and packages its own dependencies, embracing changes without impacting the entire system.	8	Concurrency	Scale out across a large number of small identical processes (copies) as opposed to scaling-up a single large instance on the most powerful machine available.
3	Configurations	Configuration information is moved out of the microservice and externalized through a configuration management tool outside of the code. The same deployment can propagated across environments with the correct configuration applied.	9	Disposability	Service instances should be disposable, favoring fast startups to increase scalability opportunities and graceful shutdowns to leave the system in a correct state. Docker containers along with an orchestrator inherently satisfy this requirement.
4	Backing Services	All required ancillary resources (data stores, caches, message brokers) should be accessed as RESTful services via an addressable URL decoupling the resource from the microservices and enabling it to be easily interchanged.	10	Dev/Prod Parity	Keep environments across the application lifecycle (Dev, QA, Staging and Prod) as similar as possible, avoiding costly shortcuts. Here, the adoption of containers can greatly contribute by promoting the same execution environment.
5	Build, Release, Run	Each release must enforce a strict separation across the build, release and run stages. Each should be tagged with a unique ID and support the ability to roll back. Modern CI/CD systems help fulfill this principle.	11	Logging	Treat logs generated by microservices as event streams, processed by event aggregator infrastructure and propagated to data-mining/log management tools like Azure Monitor or Splunk and eventually long-term archival.
6	Statelessness	Each microservice will be stateless with any necessary state externalized to a backing service (i.e., distributed cache, data store), providing seamless scalability and fault tolerance.	12	Admin Processes	Run administrative/management tasks as one-off processes. These tasks might include data cleanup or pulling analytics for a report. Tools performing such tasks should be invoked from the production environment, but separately from the application.

Dapr: writing less code and achieving more



Distributed Application Runtime

Portable, event-driven, runtime for building distributed applications across cloud and edge

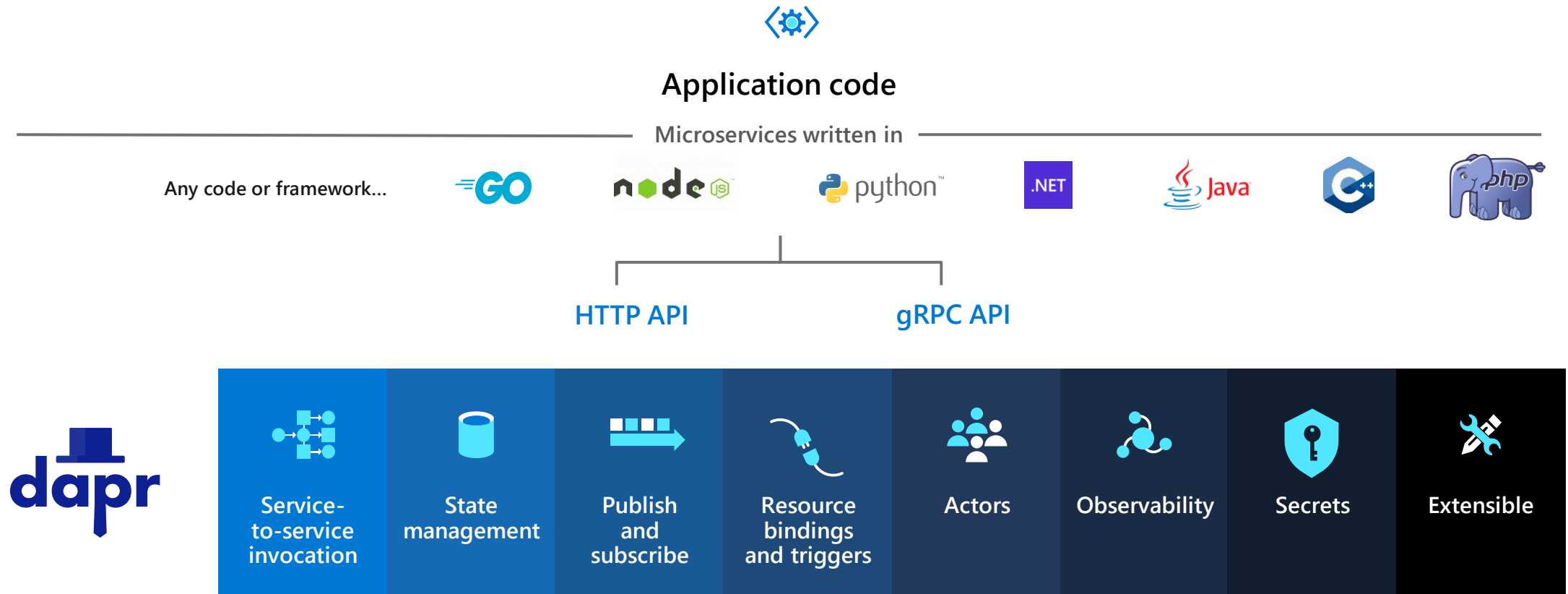
dapr.io

The screenshot shows the Dapr website homepage. At the top, there is a navigation bar with the Dapr logo, links for 'Blog', 'Docs', 'GitHub', and 'Discord', a 'Star 11,791' badge, and a 'Try Dapr' button. The main content area features the headline 'Simplify cloud-native application development' and the subtext 'Focus on your application's core logic and keep your code simple and portable'. A diagram illustrates two applications, 'App A' and 'App B', both running on the Dapr runtime (represented by a small Dapr logo icon). App A is connected to a database icon, and App B is connected to a lock icon. A double-headed arrow indicates communication between the two apps. A 'Get Started' button is positioned below the text. At the bottom of the main content area, a dark blue banner contains the text 'Announcing Dapr v1.0! Dapr is now production ready! Learn more >>'. The footer includes the text 'What is Dapr?' and a video player for 'Introducing Dapr: The Distributed Application Runtime' with 'Watch later' and 'Share' options.

What is Dapr?

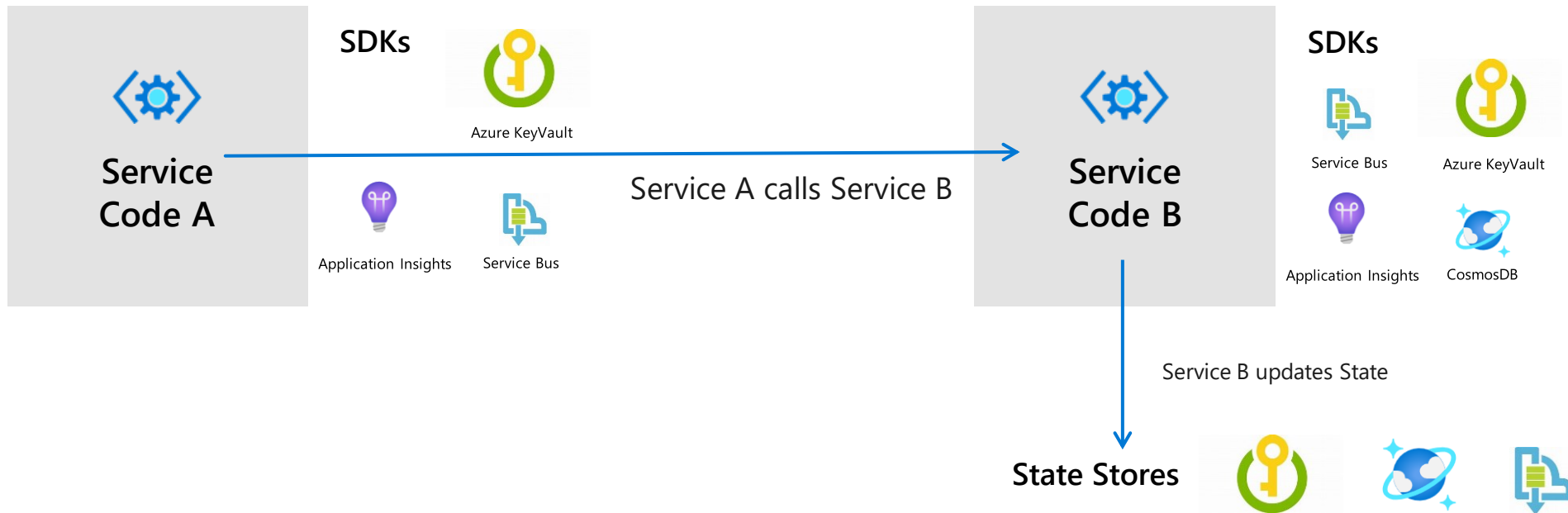
A video player interface showing the title 'Introducing Dapr: The Distributed Application Runtime' and controls for 'Watch later' and 'Share'.

Dapr is about building blocks for every developer



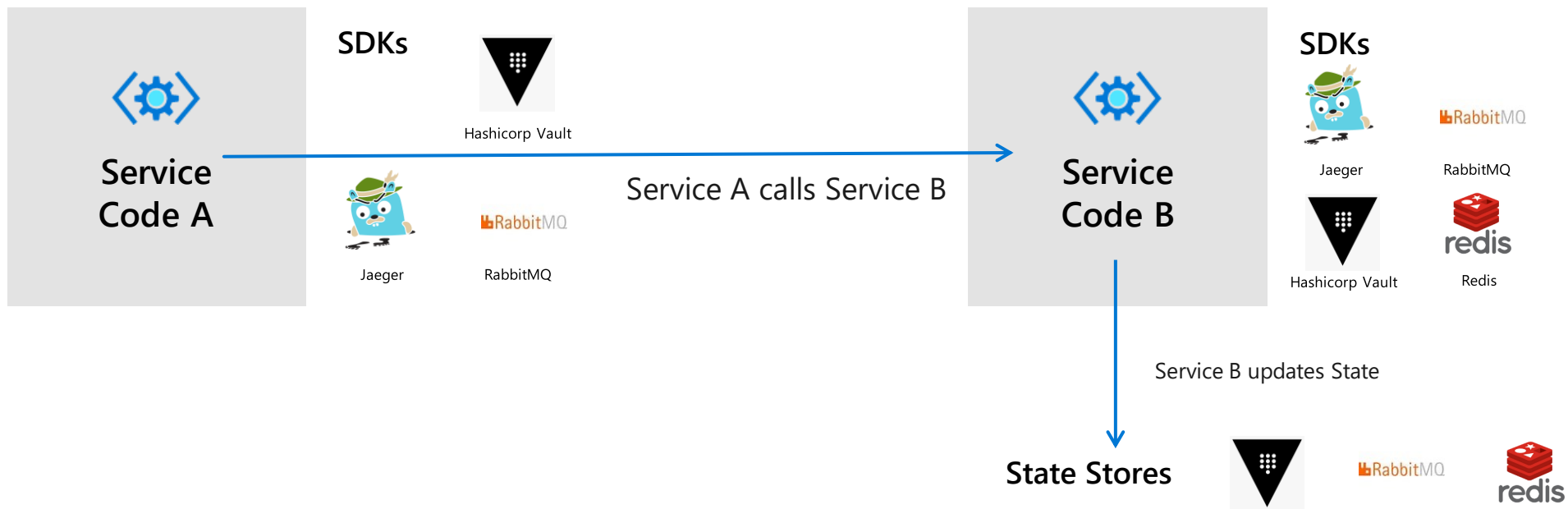
How to deal with complexity

Distributed Applications on Infrastructure



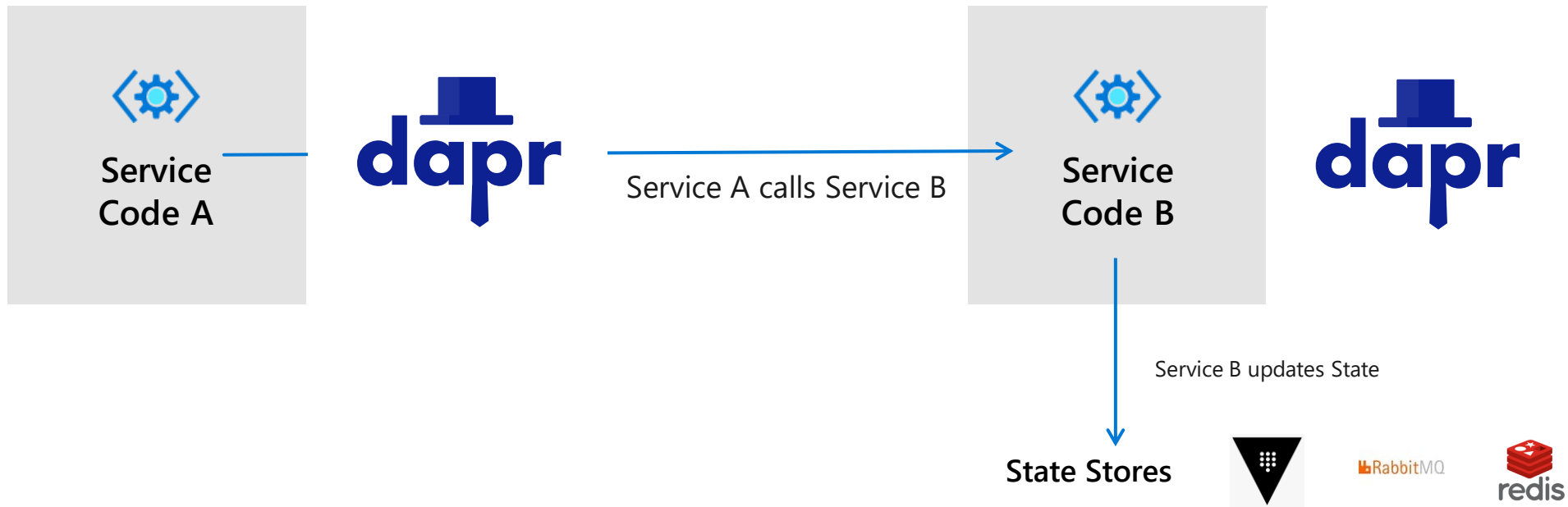
How to deal with complexity

Distributed Applications on Infrastructure

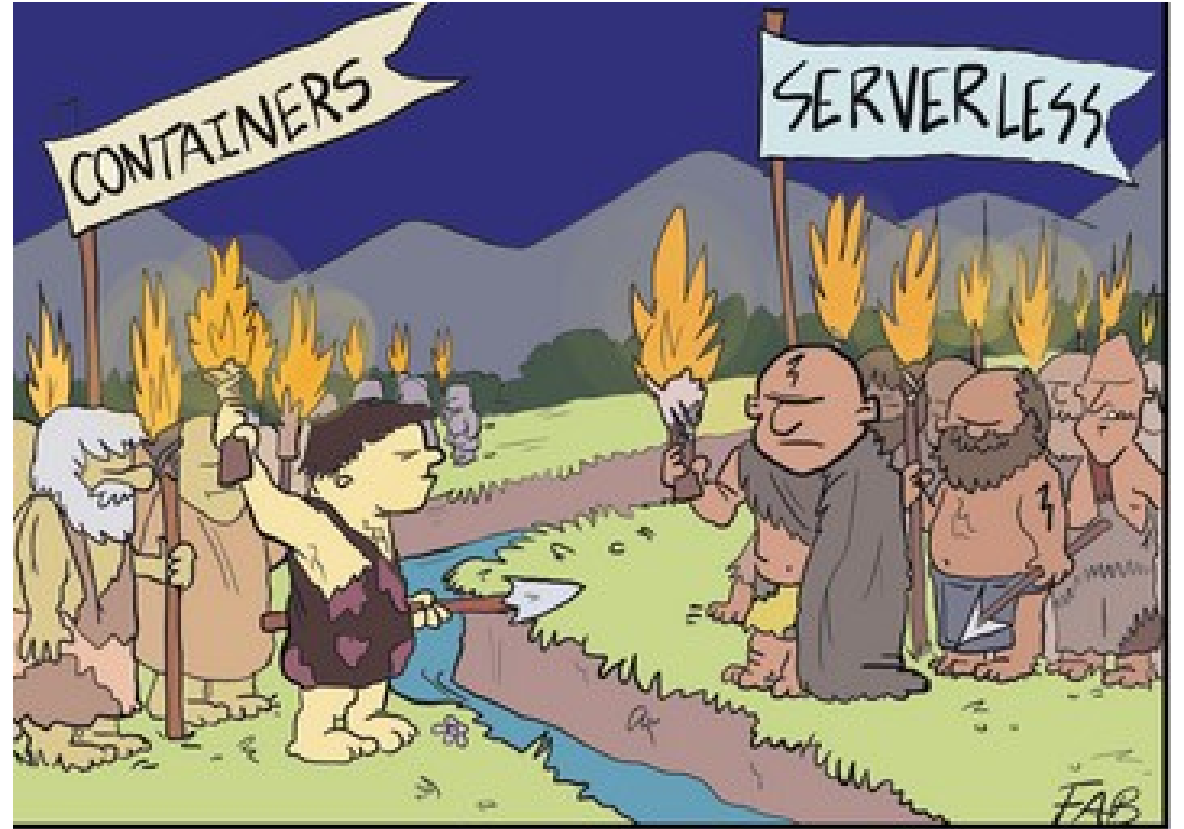


How to deal with complexity

Distributed Applications on Infrastructure



How do you choose your platform?



Platform choices and trade offs



Azure Serverless



Productivity
Locked Ecosystem
Fully supported

Platform choices and trade offs



Azure Serverless



Azure Container Apps



Productivity
Locked Ecosystem
Fully managed and supported

Managed Applications
Smaller Open Source Ecosystem
Fully managed and supported

Platform choices and trade offs



Azure Serverless



Azure Container Apps



Azure Kubernetes



Productivity
Locked Ecosystem
Fully managed and supported

Managed Applications
Smaller Open Source Ecosystem
Fully managed and supported

Control
Open CNCF Ecosystem
Fully managed and supported
Infrastructure and addons

Platform choices and trade offs



Azure Serverless



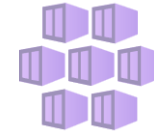
AppServices



Azure Container Apps



Spring Cloud



Azure Kubernetes



SF



ACI



OPENSIFT

ARO



CLOUD NATIVE
COMPUTING FOUNDATION



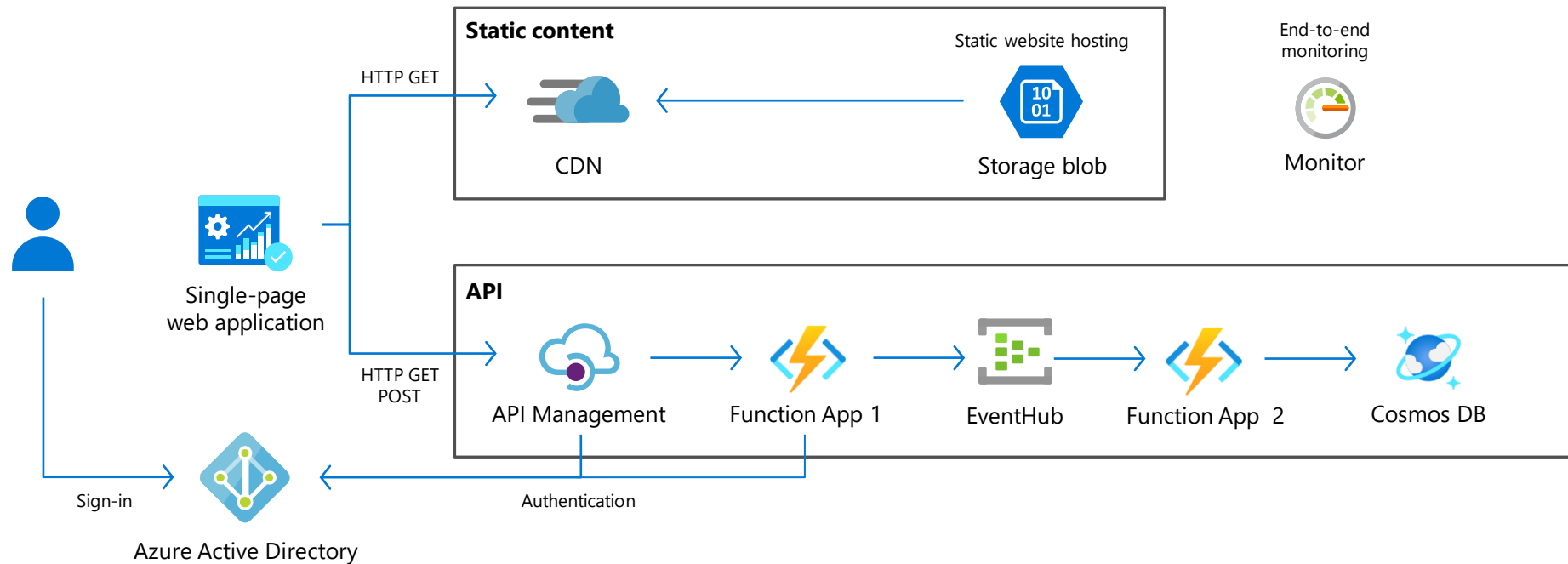
Productivity
Locked Ecosystem
Fully managed and supported

Managed Applications
Smaller Open Source Ecosystem
Fully managed and supported

Control
Open CNCF Ecosystem
Fully managed and supported
Infrastructure and addons

Serverless Applications

High scale web application with first party services



Serverless Applications

```
public static class MyFunctions
{
    [FunctionName("SayHello")]
    public static IActionResult Run([HttpTrigger("get", "post")]
        HttpRequest req, ILogger log)
    {
        log.LogInformation("Received request to say hello");
        string name = req.Query["name"];
        name = name ?? "friend";
        return new OkObjectResults($"Hello, {name}!");
    }

    [FunctionName("AddToQueue")]
    [return: Queue("myqueue-items")]
    public static string Run([HttpTrigger] dynamic input, ILogger log)
    {
        log.LogInformation($"Adding to queue: {input.Text}");
        return input.Text;
    }
}
```

Output to Client

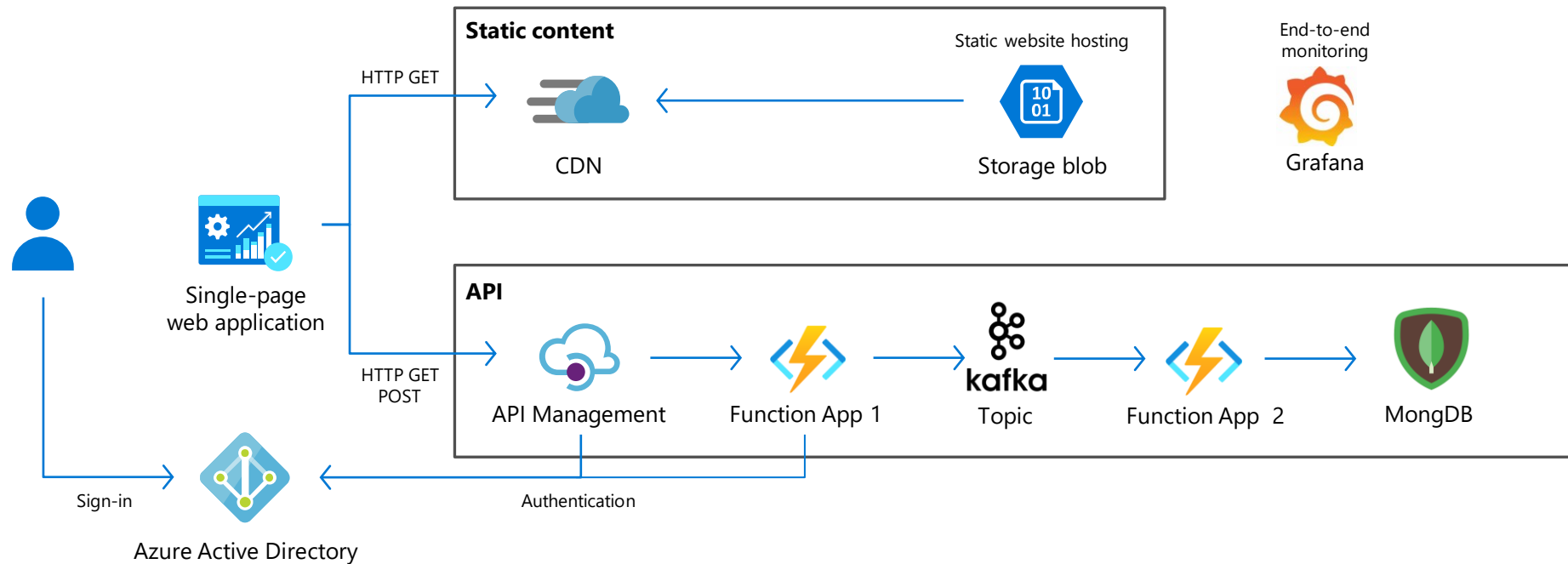
Function Trigger

Function Code

Output to queue

Serverless Applications

High scale web application open source technology

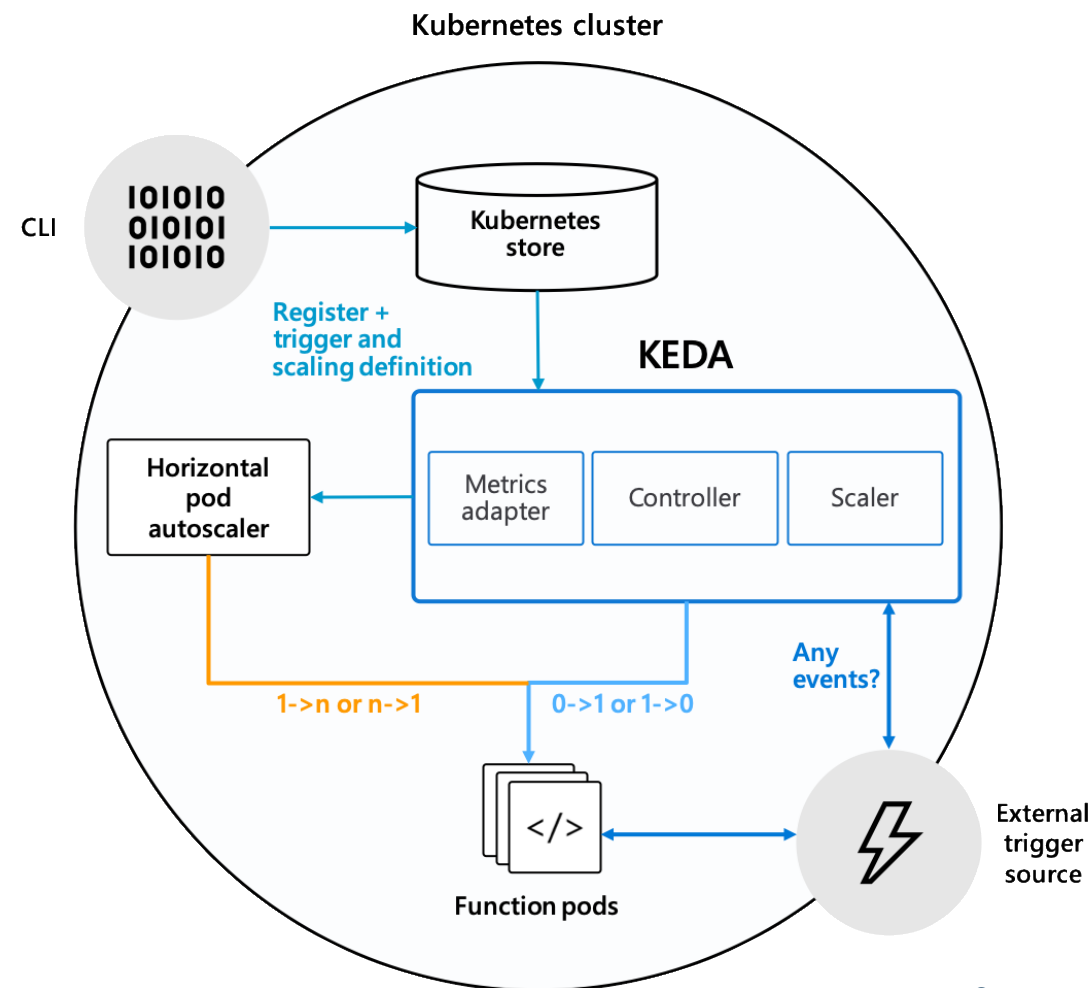


Ensure scaling applications



KEDA

- AWS CloudWatch
- AWS Simple Queue Service
- Azure Event Hub
- Azure Service Bus Queues and Topics
- Azure Storage Queues
- GCP PubSub
- Kafka
- Prometheus
- RabbitMQ
- Redis Lists



Public preview

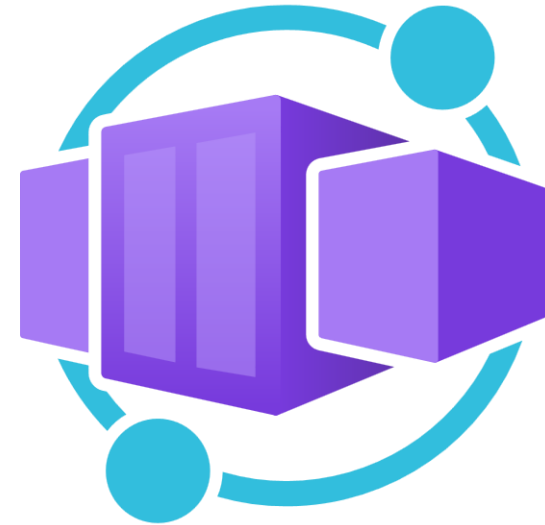
Azure Container Apps

Serverless containers for microservices

Build modern apps on open source

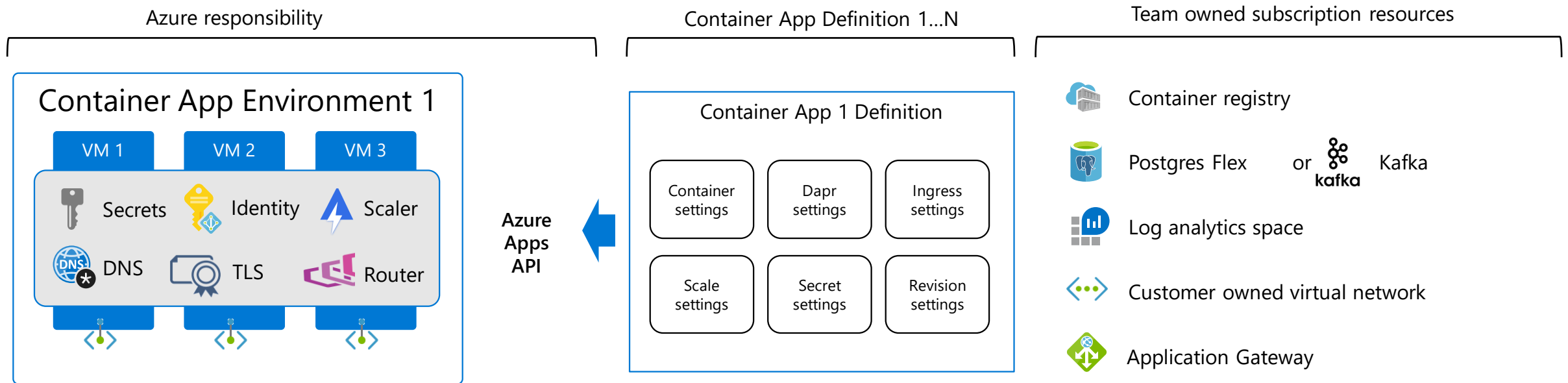
Focus on apps, not infrastructure

Seamlessly port to Kubernetes



Azure Container Apps

Azure Container Apps hosts apps and microservices that scale dynamically based on HTTP traffic or events, as well as long-running background jobs.



High developer productivity

The screenshot shows the Azure portal interface for a Container App. The top navigation bar includes the Microsoft Azure logo, a search bar, and the user profile 'Connie Wilson'. The breadcrumb trail is 'Home > Container Apps > contosoContainerApp'. The left sidebar contains navigation options: Overview, Access control (IAM), Tags, Worker App settings (Secrets, Ingress, Continuous deployment), and Revisions (Revision management). The main content area displays the 'Essentials' section with a table of configuration details:

Property	Value	Property	Value
Resource group (change)	: contosoRG	Application URL	: https://contosoContainerApp.azureworkerapp.io
Status	: Running	Worker App Environment	: contosoEnvironment
Location	: West US 2	Virtual network	: contosoVNet
Subscription (change)	: contosoSubscription	Log Analytics	: contosoLA
Subscription ID	: b9184e8a-0517-4848-8c79-db9aa4716efd	Application Insights	: contosoAI
Tags (change)	: Click here to add tags		

Below the Essentials section, there is a heading 'Create revisions to manage traffic and scaling' with a sub-heading 'With Azure Container Apps, you create different revisions of the app that address different markets for example, or handle different configurations for autoscaling, container images, or Dapr. [Learn more](#)'. Three action cards are displayed:

- Manage your app with revisions**: Use revisions to set up autoscaling, specify Dapr settings, and configure your container. Every change you make creates a new revision, giving you complete control over your deployments. [Learn more](#)
[View revisions](#)
- Set up continuous deployment**: Set up GitHub Actions for automatic deployment of the container image and the application code. [Learn more](#)
[Set up deployment](#)
- Create secrets**: Protect sensitive data by adding secrets to your app. Once you create a secret you can reference it in the next app revision. [Learn more](#)
[Create secrets](#)



Select any container image using any language or framework



Choose vCPU cores, memory, and scale settings based on events or HTTP requests

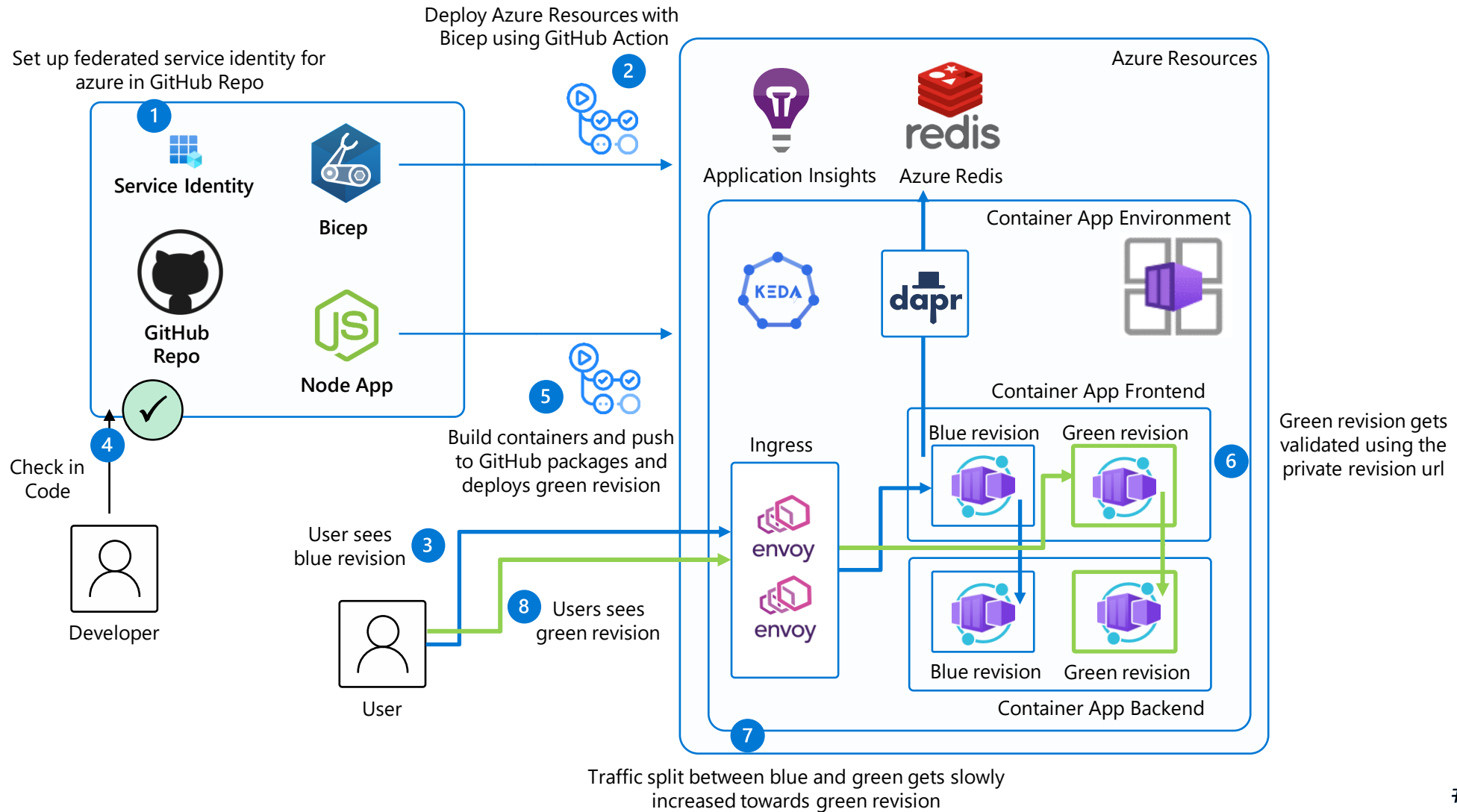


Enable service-to-service communication, configure ingress, and event sources

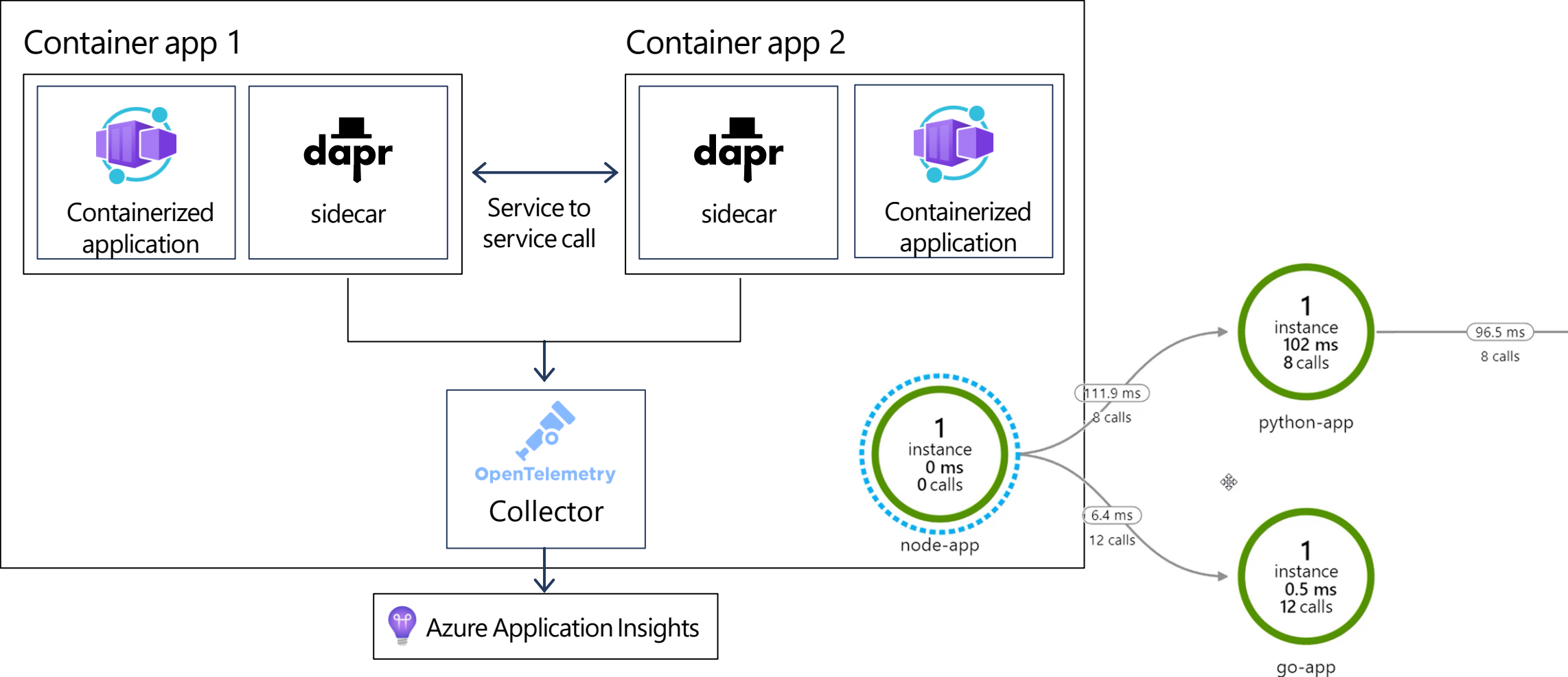


Create and deploy your application

Continuous blue/ green deployments



Builtin observability



Why are you using Kubernetes?



David Samuelsson

@dasamuelsson

Follow



Wednesday [#funny](#):

2014 - We must adopt [#microservices](#) to solve all problems with monoliths.

2016 - We must adopt [#docker](#) to solve all problems with microservices.

2018 - We must adopt [#kubernetes](#) to solve all problems with docker

12:41 AM - 19 Sep 2018

27 Retweets 52 Likes



4



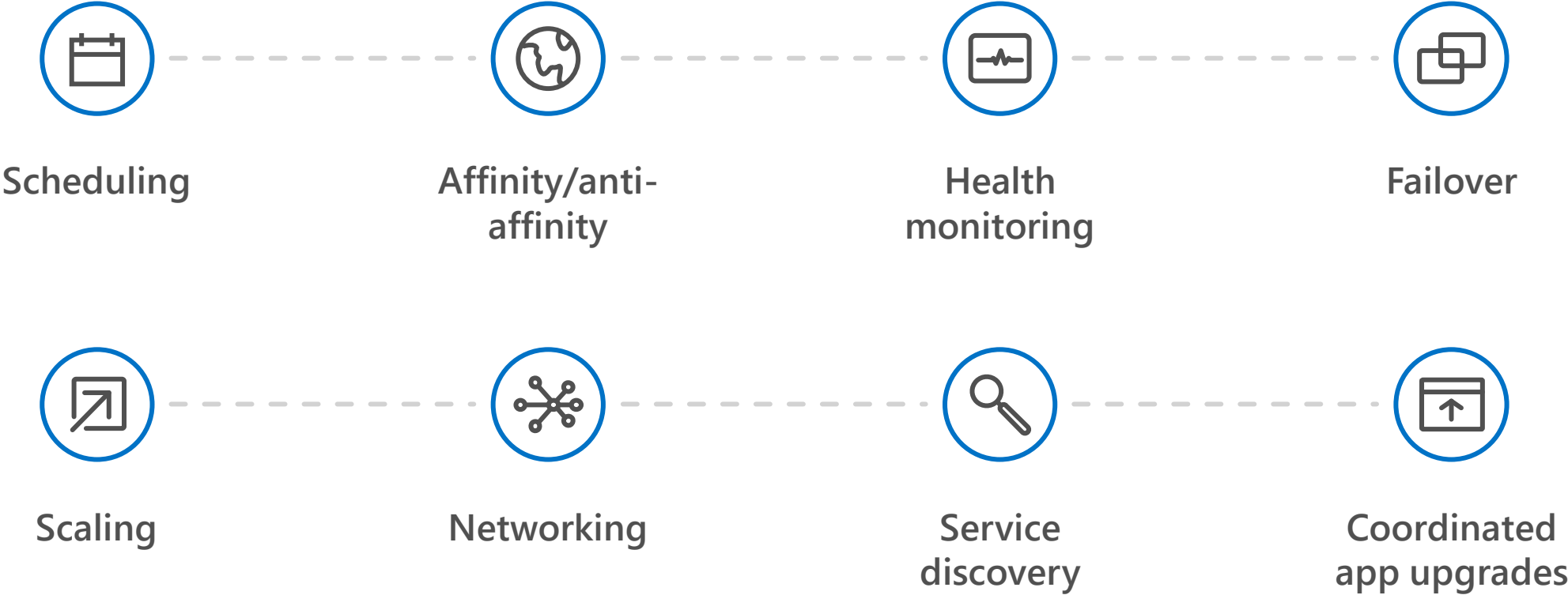
27



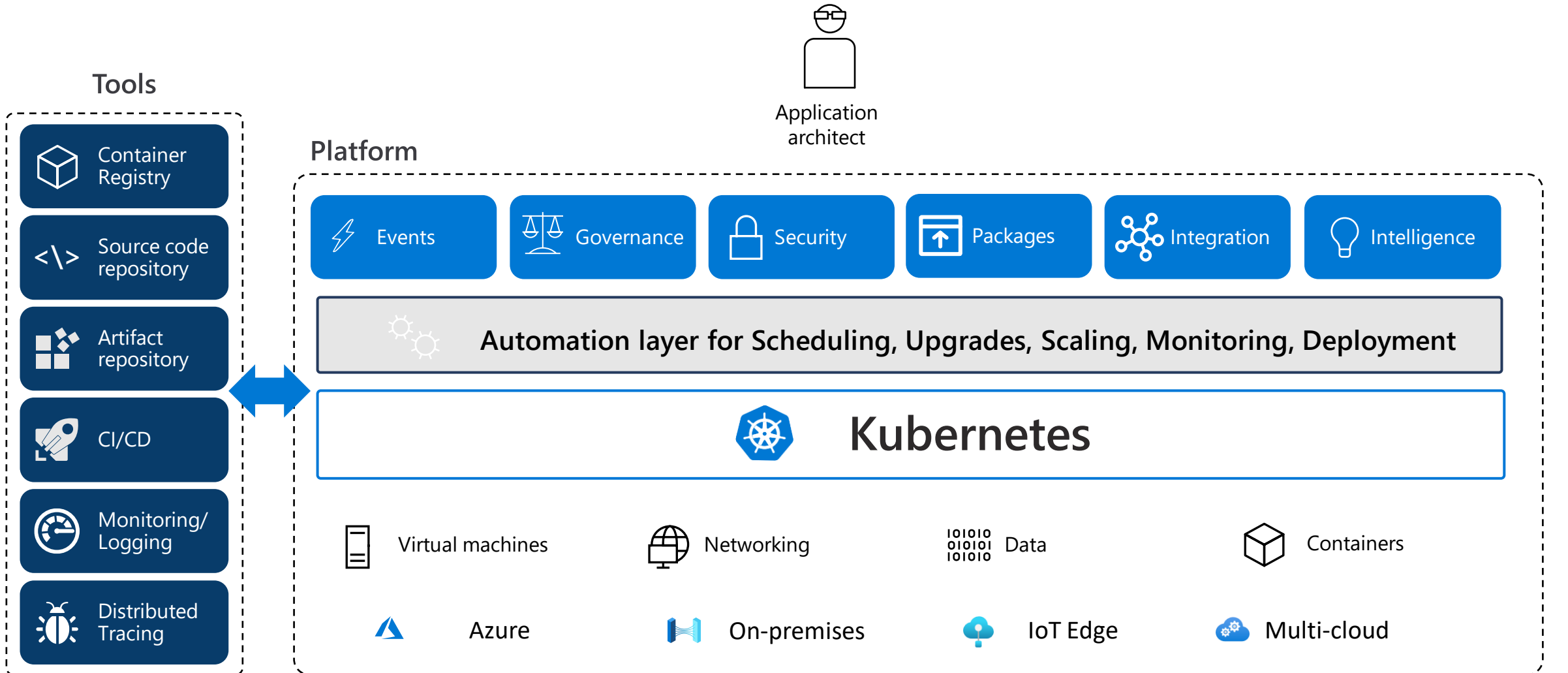
52



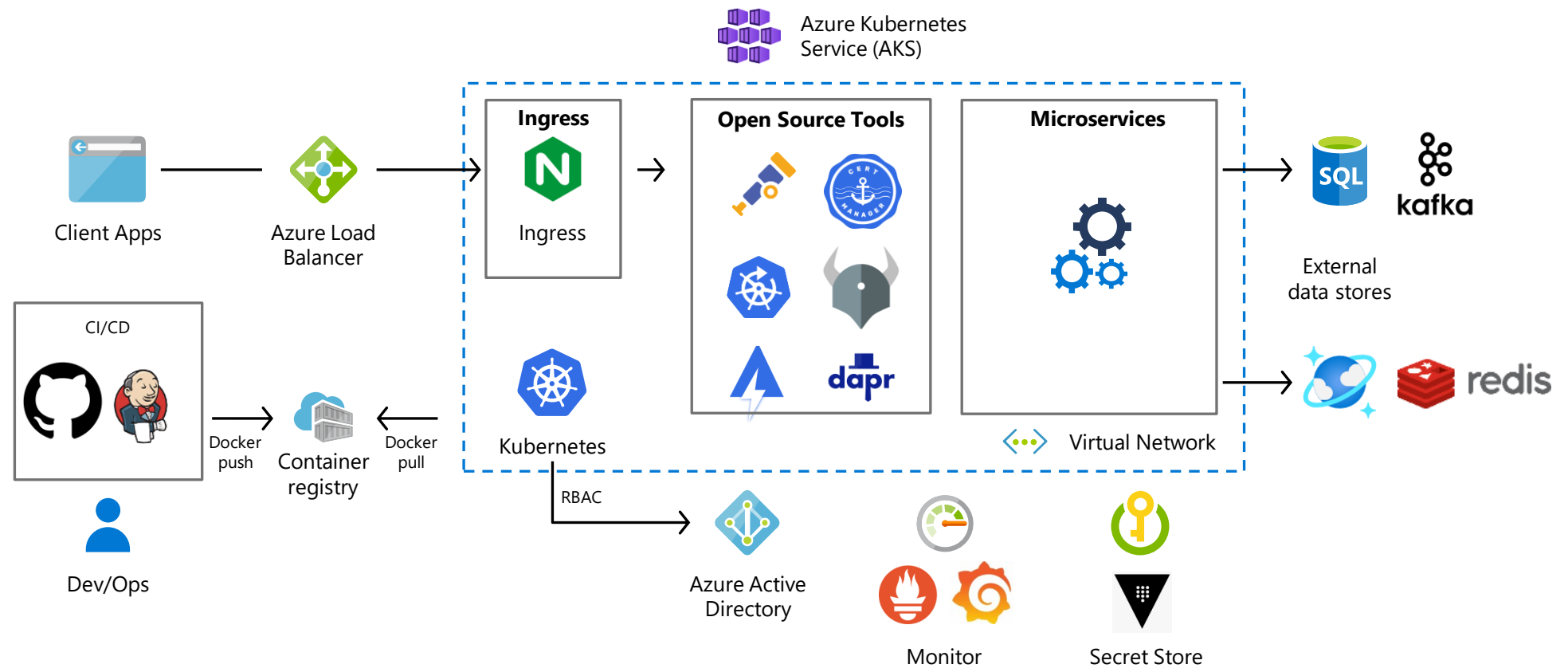
Benefits of Kubernetes



Kubernetes creates need for more concepts



Azure Kubernetes Service



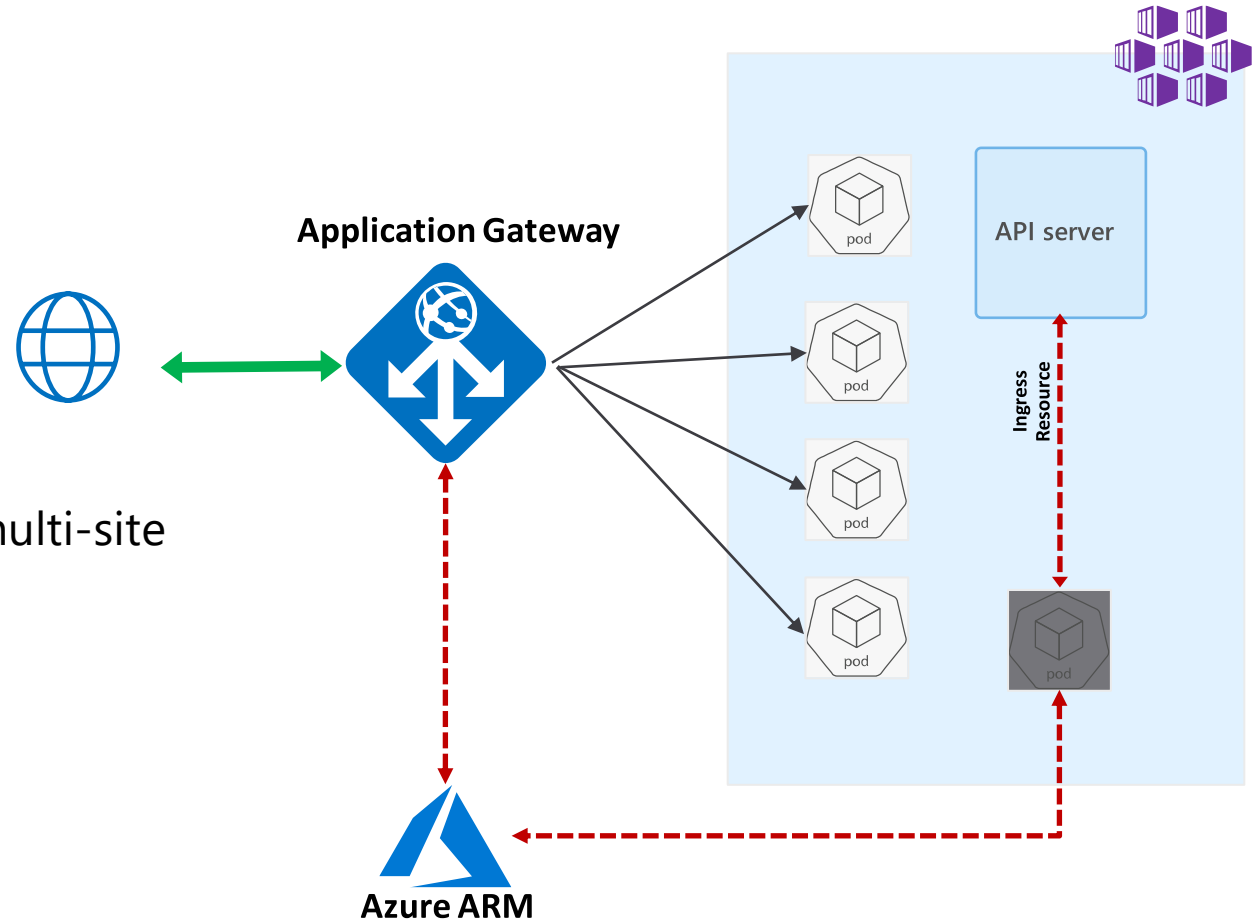
Managed Components: Ingress

Attach Application Gateways to AKS Clusters

Load Balance from the Internet to pods

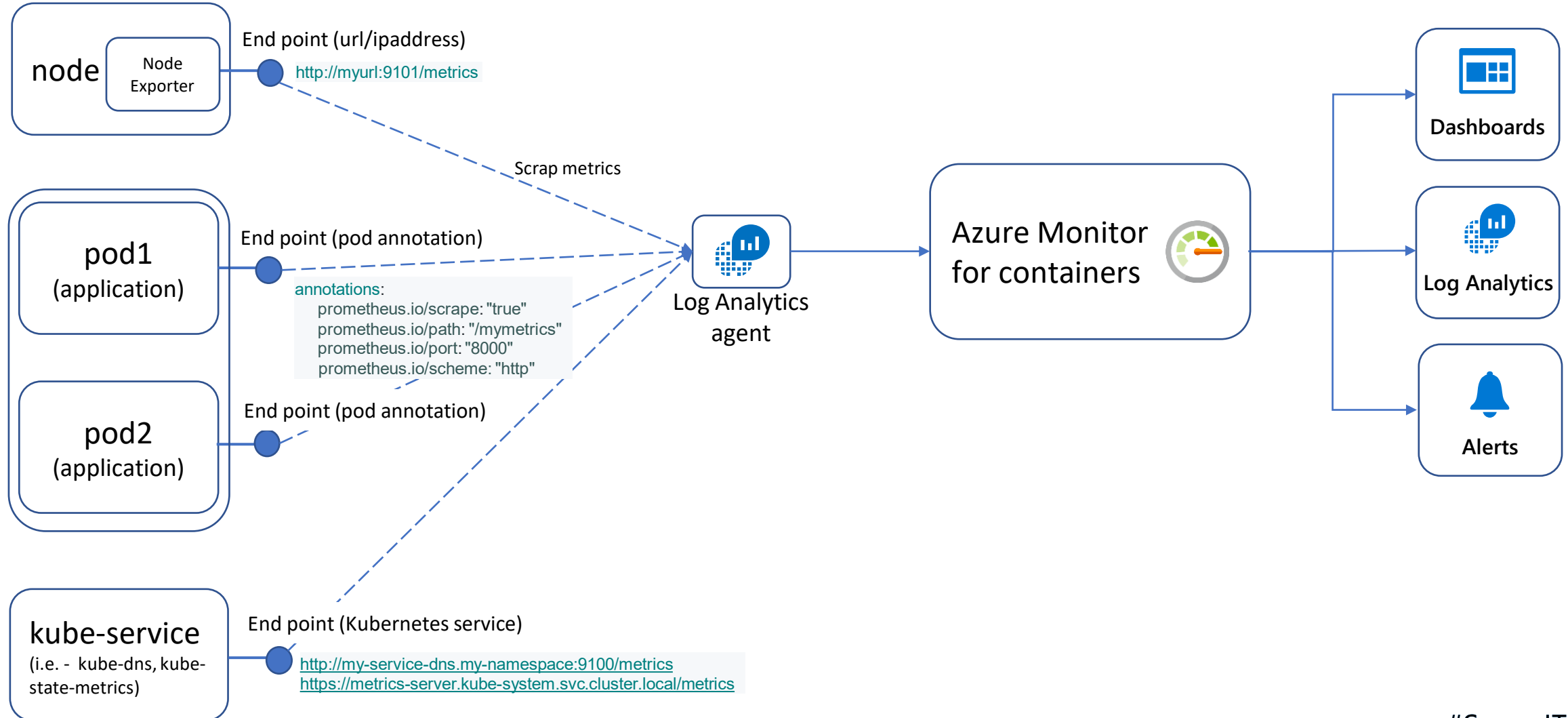
Supports features of k8s ingress resource – TLS, multi-site and path-based routing

Pod-AAD for ARM authentication

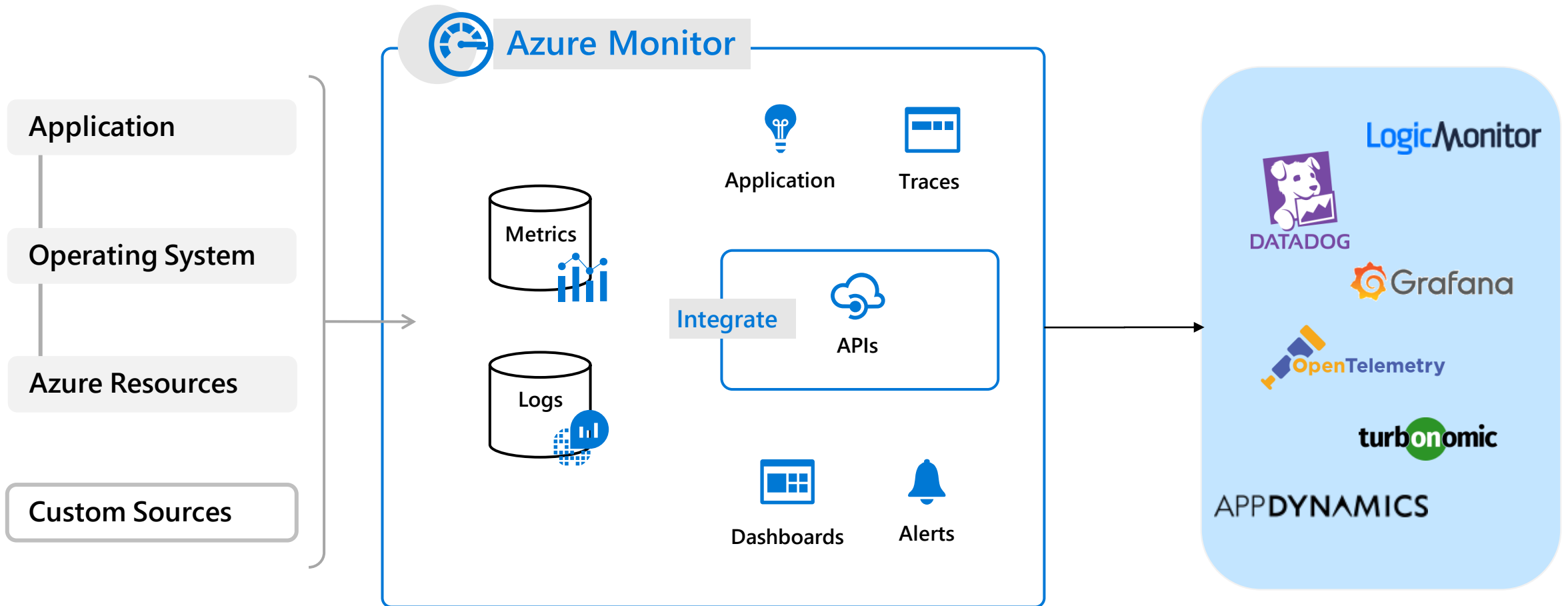


<https://github.com/Azure/application-gateway-kubernetes-ingress>

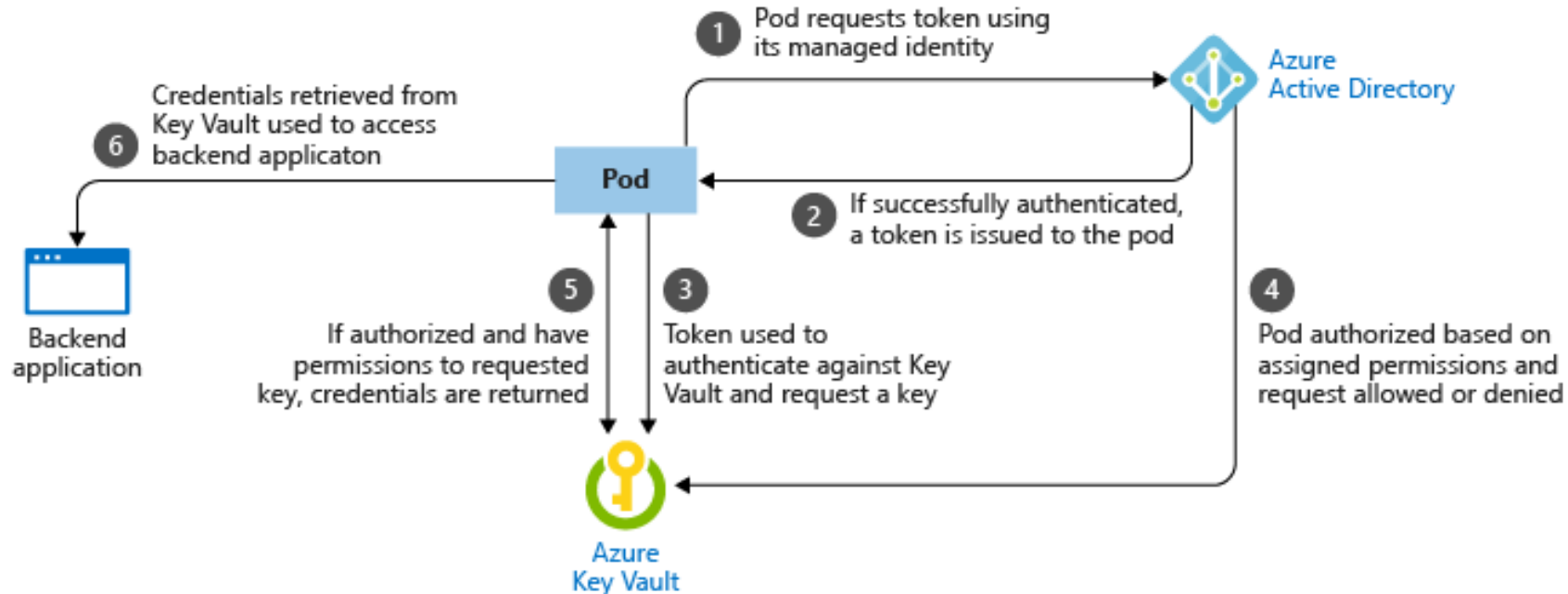
Managed Components: Monitoring



Azure Monitor



Managed Components: Secret Store Driver



<https://github.com/kubernetes-sigs/secrets-store-csi-driver>

Do you need a service mesh?

Traffic Management

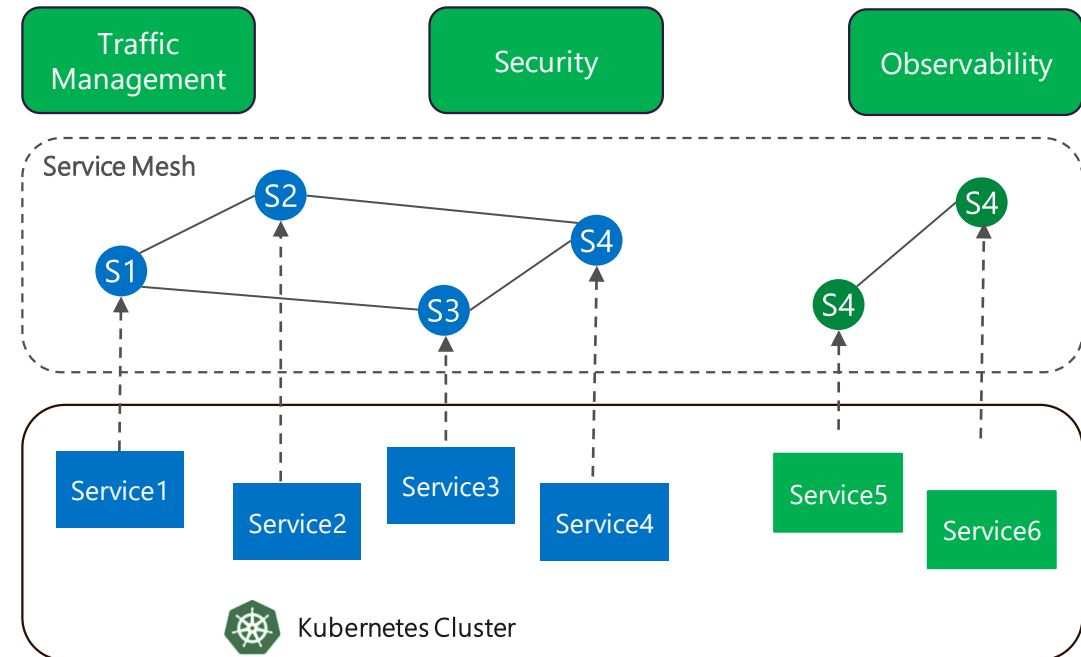
Request routing, weighted load-balancing, fault injection, circuit breaker patterns etc.

Security

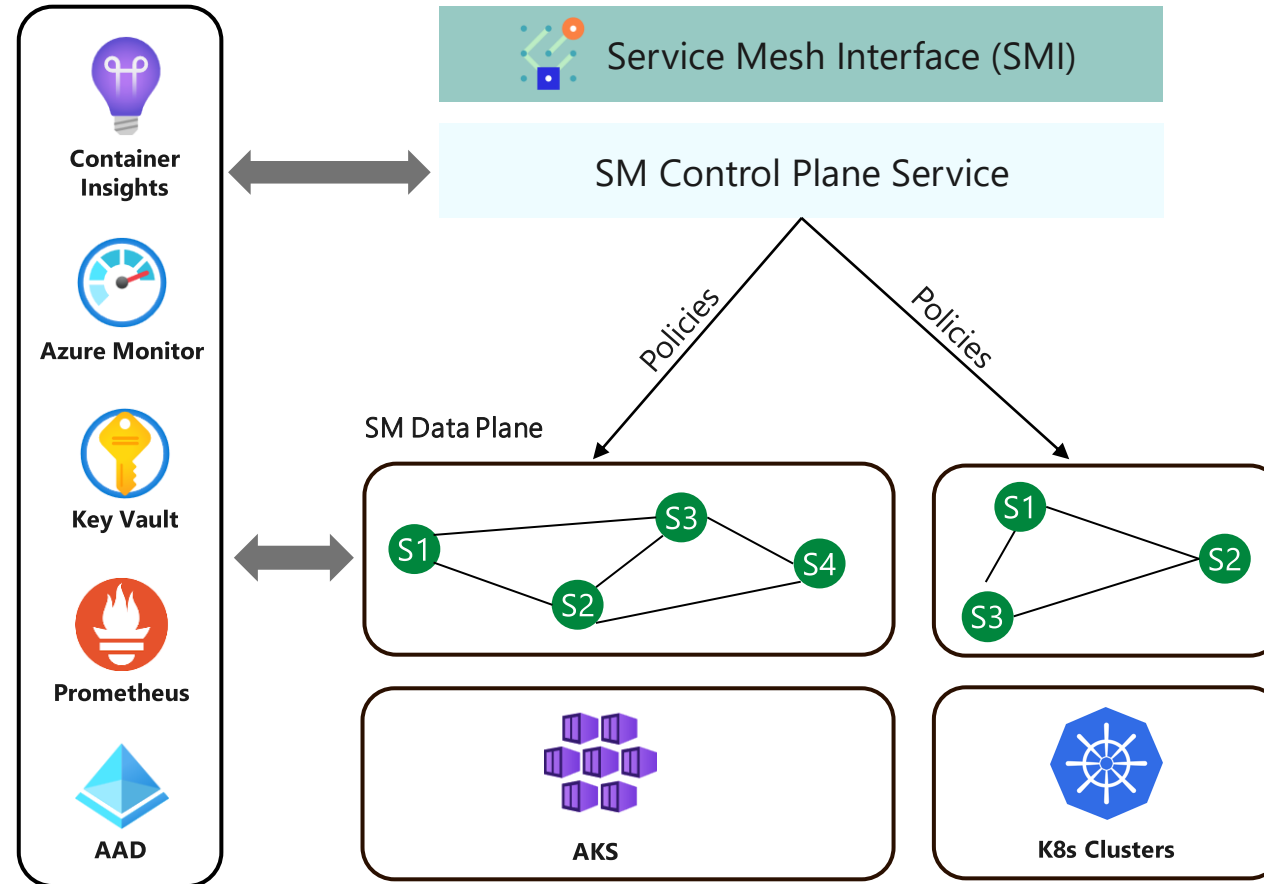
mTLS encryption+ authentication/authorization for communication between services

Observability

Traffic tracing; visibility into connections accepted/denied



Managed Components: Open Service Mesh



<https://github.com/openservicemesh/osm>

Ensuring consistency

FaaS and Furious by Forrest Brazeal



"Come on, make up your mind -
or it's back to the Sinkhole of Nested XML."

Decide on Infrastructure as Code Toolchain



Azure Bicep



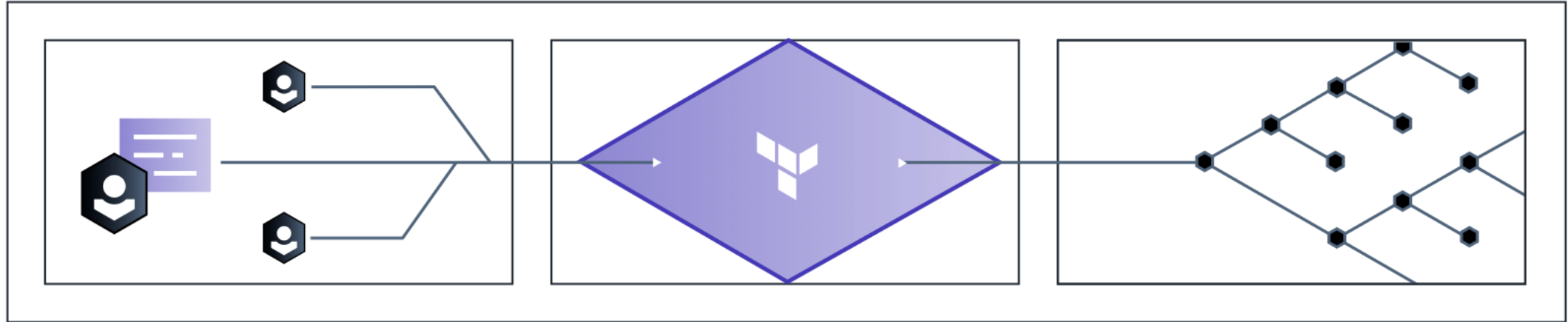
HashiCorp Terraform



Pulumi

...

Ideal deployment and upgrade methodology

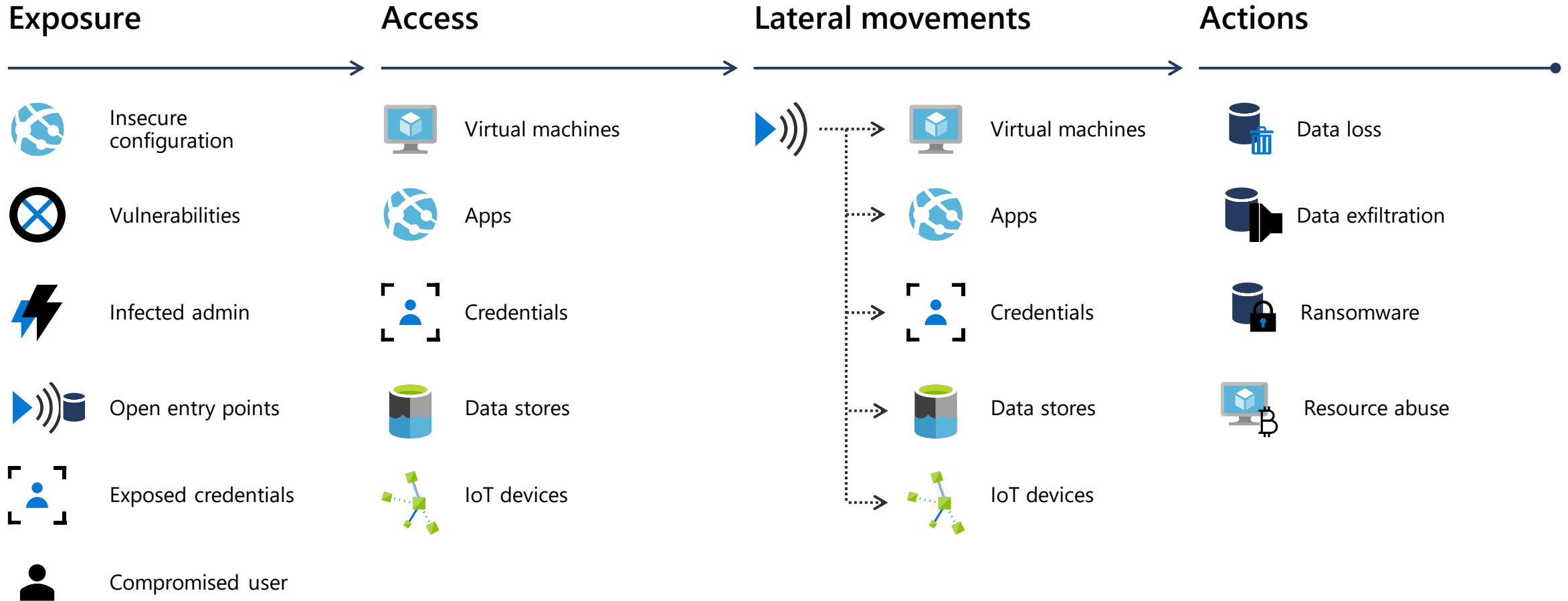


WRITE & COLLABORATE
on infrastructure as code
using a version control system

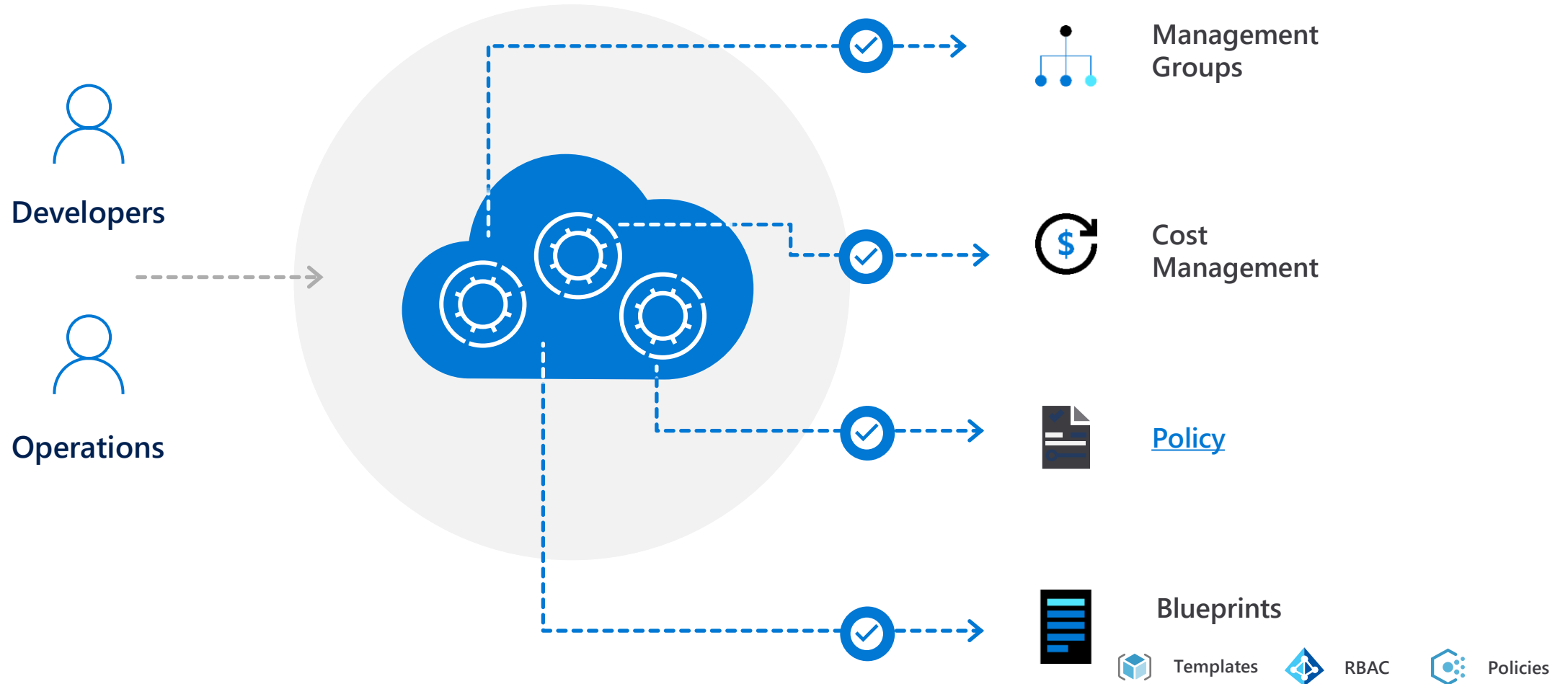
PLAN & VALIDATE
Preview changes before
applying with a common
workflow

CREATE
Reproducible
infrastructure, safely.

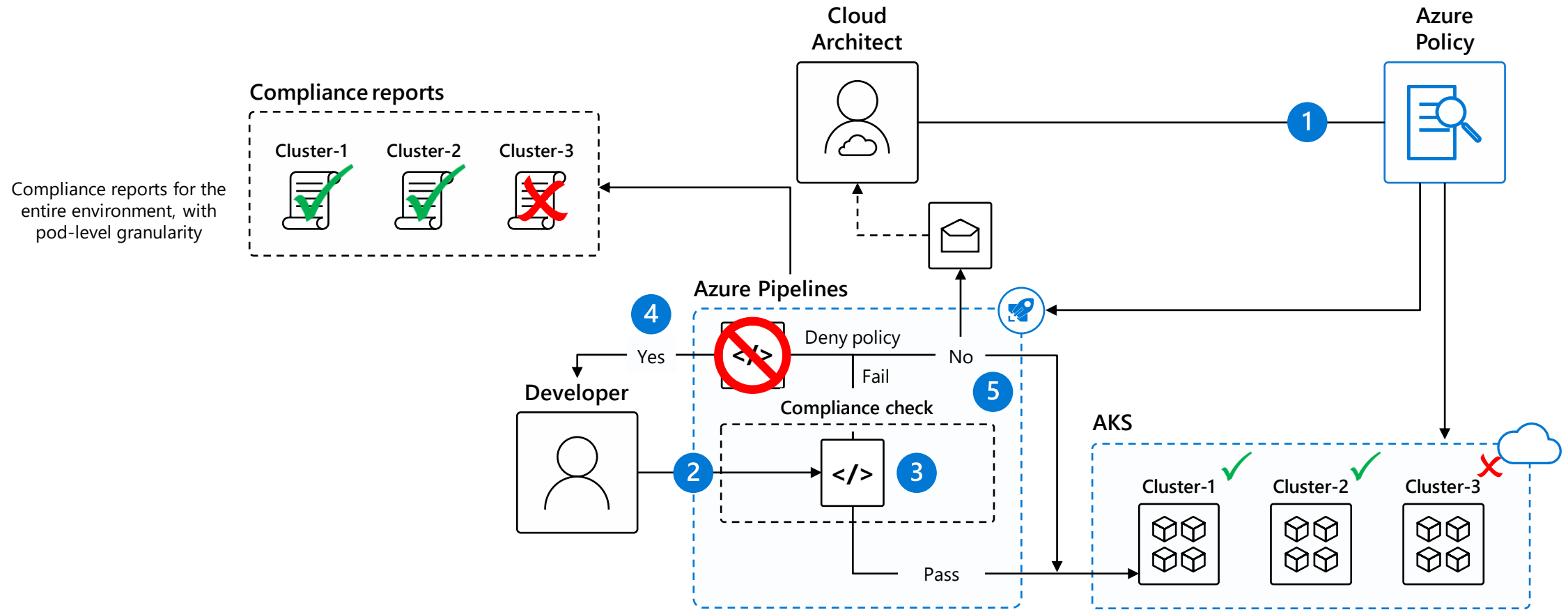
Watch for exposures in your environments



How to balance speed and control?

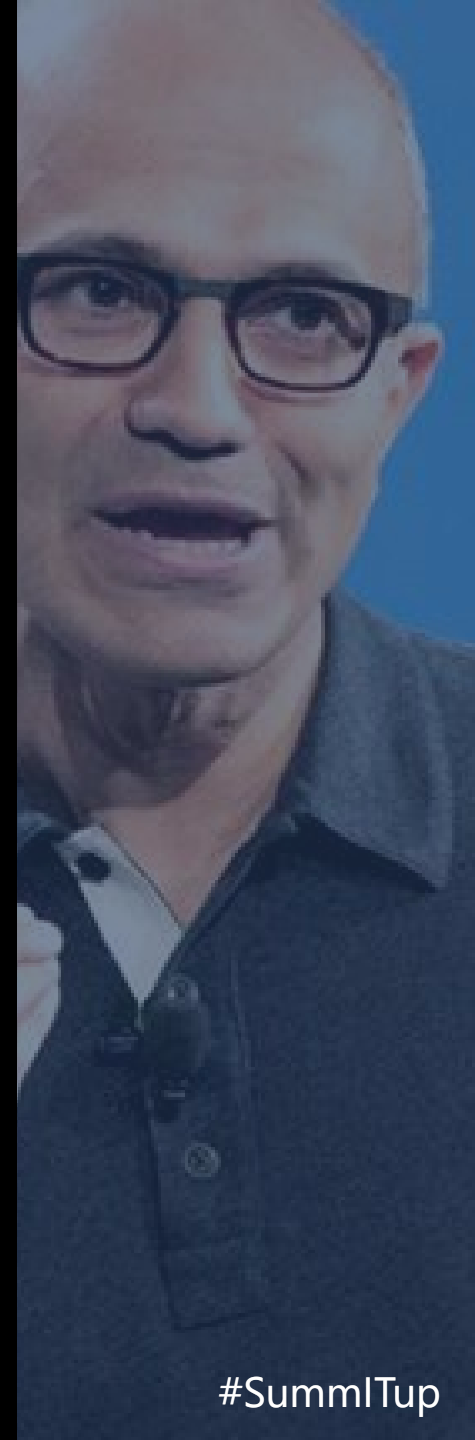


Policy at scale



// Tech companies born with an open source mentality get it. It's our ability to work together that makes our dreams believable and, ultimately, achievable. We must learn to build on the ideas of others"

—Satya Nadella, CEO
Microsoft





Thank you

#SummitUp

#SummitUp



Thank you

#SummitUp





















Appendix

Icons








Icons should be primarily used as a visual aid, as a way to break up large amounts of text and content, or as a visual cue of the content that follows.

[Get the full set of icons for presentations](#)













INDUSTRY SECTORS

											
Aeronautics	Agriculture	Architecture	Automotive	Construction	Education and research	Energy	Financial services	Healthcare	Engineering and manufacturing	Professional services, media	Sports
								-	-	-	-
Government	Retail	Technology	Media and broadcasting	Travel and transportation	Education	Law	Science	-	-	-	-




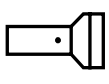





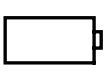



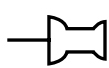

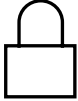




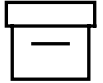

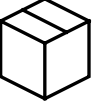





































MAPS / LOCATION

							-	-	-	-	-
Globe, global, earth	Globe, global, earth	Navigation, GPS, POI, location, map	Map, location, GPS, directions, navigation	Navigation, GPS, POI, location, map	Map, location, route, quest, destination, journey	Download map, receive location	-	-	-	-	-

MONEY / FINANCE / RETAIL / CONSUMER

											
Shopping cart, store	Store, bag, shop	Money, dollar sign, cash, revenue, currency	Revenue, financial report, funds	Money, currency, coins	Bank, piggybank	Compare, balance, finance, justice	Credit card, payment, charge, debit card	Mobile payment, digital payment, e-commerce	Barcode scanner	Tag, sale, shopping	Bank

OBJECTS / TOOLS

 Alarm clock	 Hourglass, timer	 Clock	 Flashlight	 Funnel	 Calculator	 Binoculars	 Scissors	 Camera	 Battery, power, charge	 Magnifying glass	 Tools, screwdriver, wrench
 Paperclip	 Pushpin, thumbtack	 Key	 Lock	 Safe	 Piggybank	 Bag	 Ballot box, drop box	 Box, archive, storage	 Present, giftbox	 Box, package, shipping	 Briefcase, business, work, job
 Books, reading	 Book, dictionary, novel	 Notebook, journal	 Pencil	 Paintbrush	 Paint palette, color	 3D, hologram	 Trophy, award	 Ribbon, award, reward, accomplishment	 Ribbon, award, reward, accomplishment	 Flag, finish line	 Flags, finish line, win, race
 Open door, entryway, opportunity	 Clipboard	 Megaphone, speaker, speech, announcement	 Microphone	 CD, music album	 Headphones	 Headset	 Eye glasses	 Bike, bicycle	 Umbrella, bumbershoot	 Balloon, party, celebration	 Rocking chair
 Remote control	 Robotic, manufacturing, mechanical, automation	 Forklift	 Gears	 Traffic cone, construction	 Flask, chemistry, science, beaker	 Maze, puzzle	 Target, bullseye, aim, arrow, dartboard, darts, archery	 Games, strategy, chess, poker	 Musical instrument, drums	 Pizza, food	 Lightbulb

Presentation resources

More templates

[Overview of templates on Brand Central](#)

[Product-specific PowerPoint templates](#)

[PowerPoint presentation event templates](#)

[Past event slides archive](#)

[Microsoft Story deck](#)

[Toolkit for building your own template](#)

Training

[Brand Central presentations overview page](#)

[Accessibility tips and best practices](#)