Workshop on serverless computing

Eamon O'Reilly
Lead PM on Azure Functions

Agenda

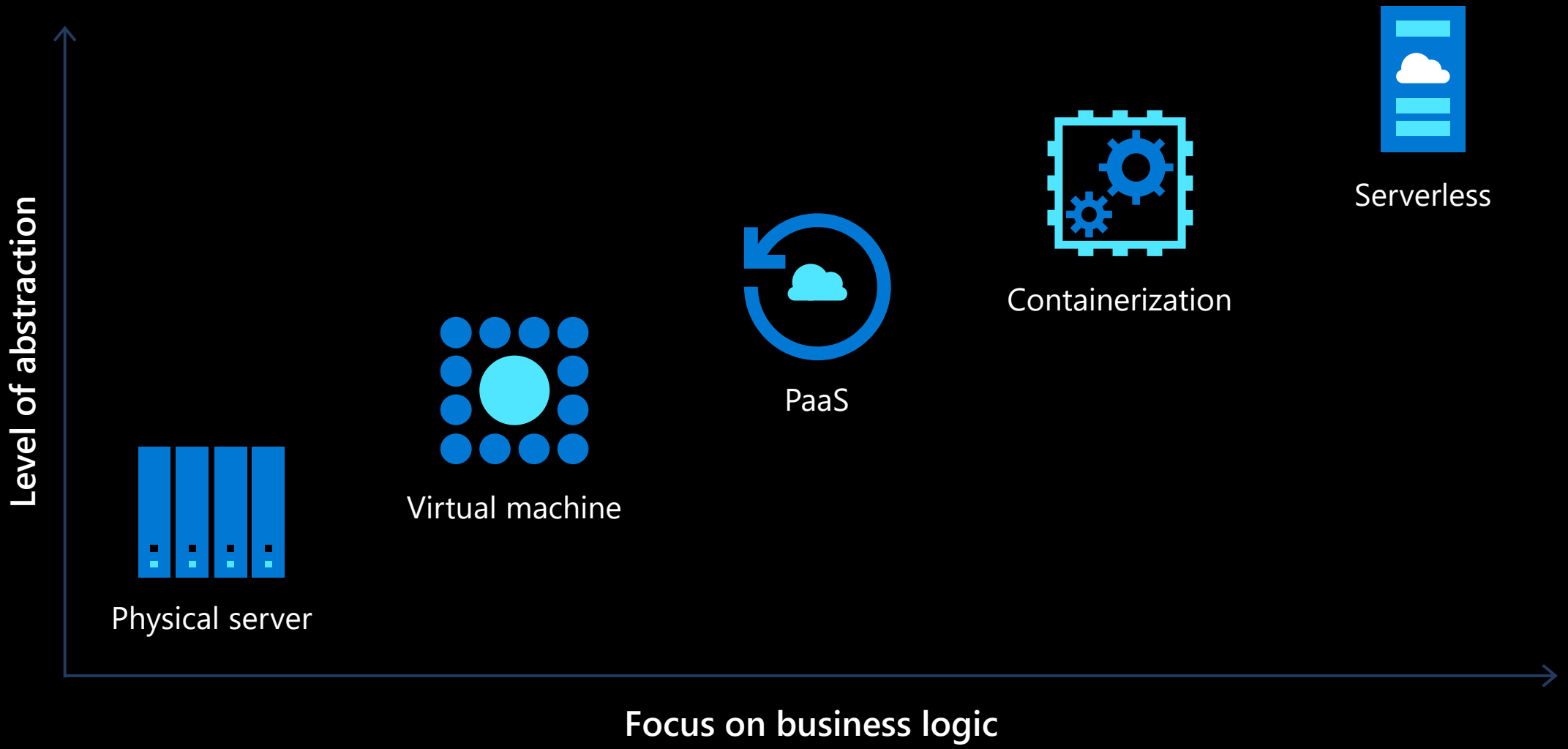
Productivity gains across each area of the application lifecycle with using serverless technologies

01 Review overall benefits of serverless computing

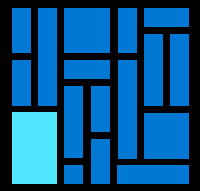
02 Identify key areas of application lifecycle management and how these are applied to serverless

03 Key takeaways when considering serverless

Evolution of computing



What is serverless?



No infrastructure management

Developers can just focus on their code—without needing to worry about provisioning and managing infrastructure

Instant, event-driven scalability

Application components react to events and triggers in near real-time with virtually unlimited scalability

Pay-per-use

Only pay for what you use: billing is typically calculated on the number of function calls, code execution time, and memory used*

FaaS is at the center of serverless

Functions-as-a-Service programming model use functions to achieve true serverless compute

Single responsibility

Functions are single-purposed, reusable pieces of code that process an input and return a result

Short-lived

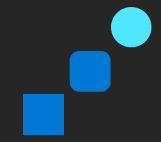
Functions don't stick around when finished executing, freeing up resources for further executions

Stateless

Functions don't hold any persistent state and don't rely on the state of any other processes

Event-driven and scalable

Functions respond to predefined events, and are instantly replicated as many times as needed



Azure Functions

An event-based, serverless compute experience that accelerates app development

Integrated programming model

Use built-in triggers and bindings to define when a function is invoked and to what data it connects

End-to-end development experience

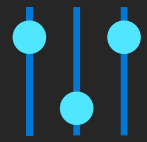
Take advantage of a complete, end-to-end development experience with Functions—from building and debugging locally on major platforms like Windows, macOS, and Linux to deploying and monitoring in the cloud

Hosting options flexibility

Choose the deployment model that better fits your business needs without compromising development experience

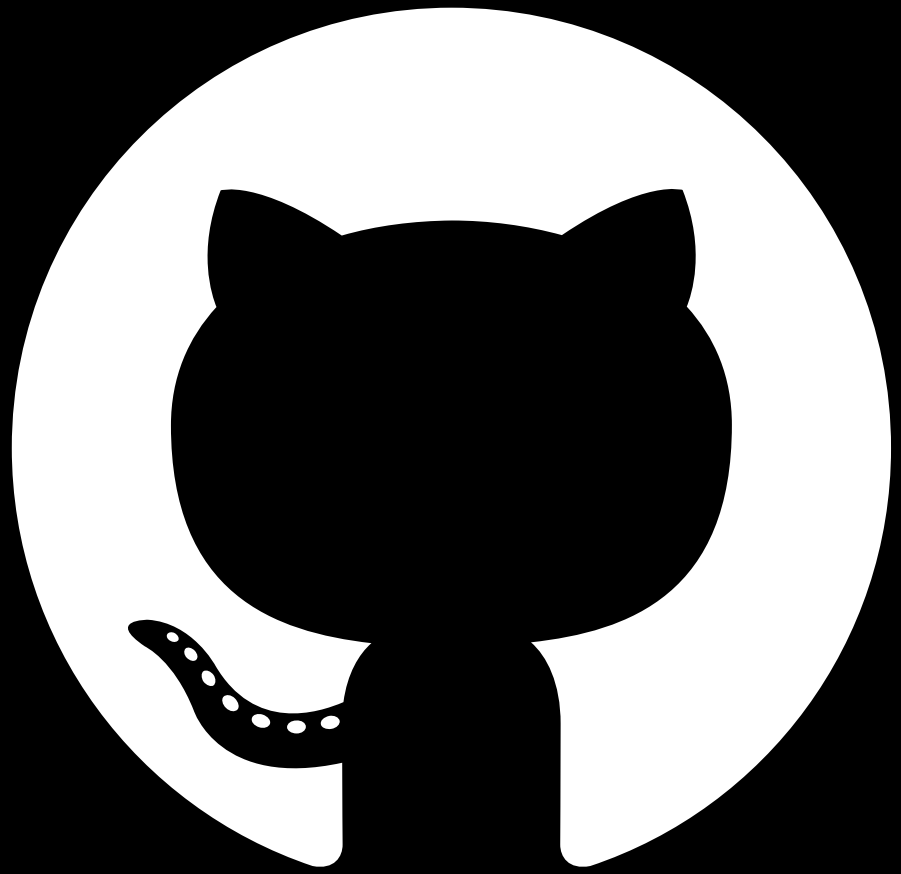
Fully managed and cost-effective

Automated and flexible scaling based on your workload volume, keeping the focus on adding value instead of managing infrastructure



Azure Functions is an **open-source** project

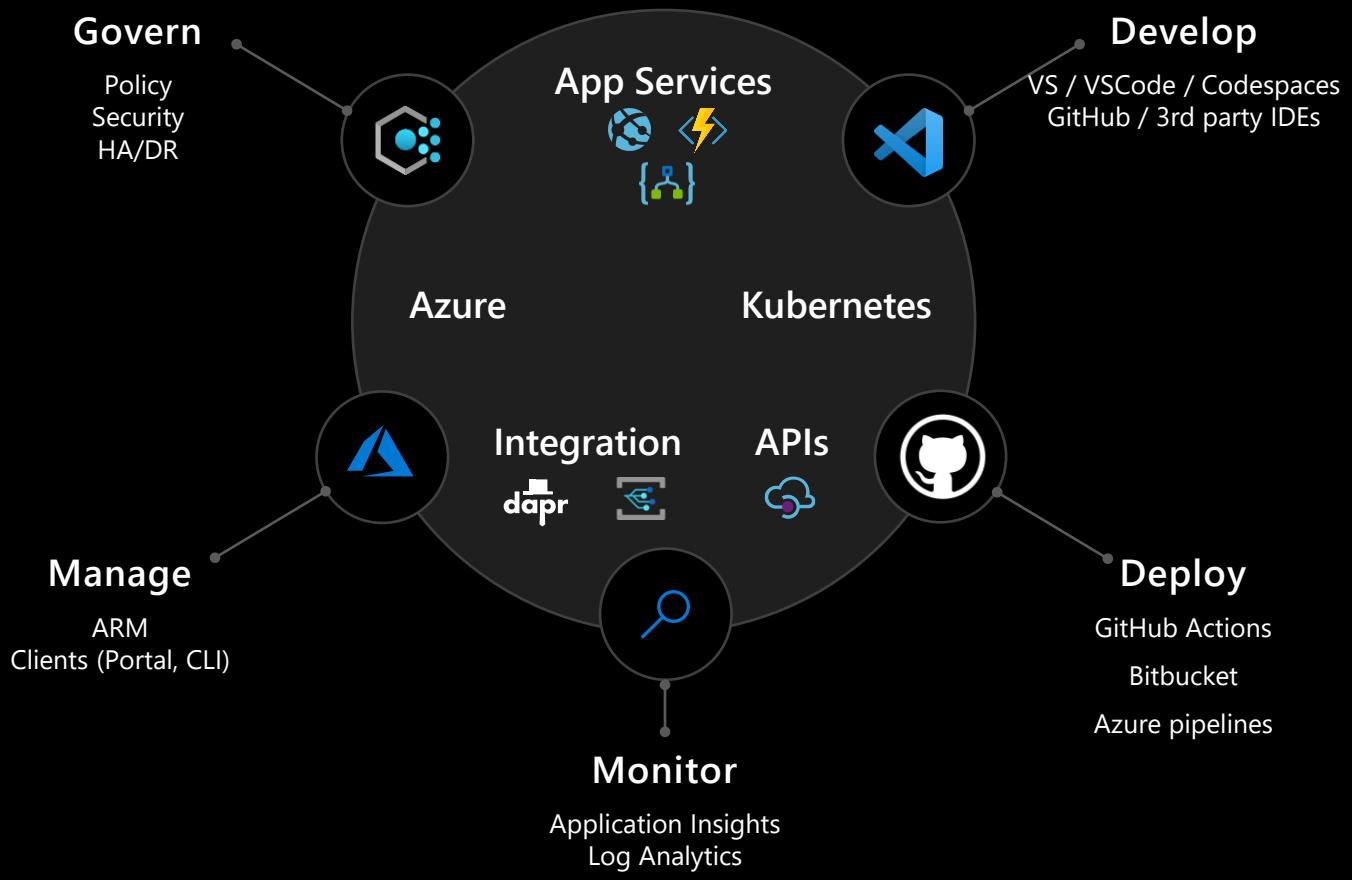
Functions runtime and all extensions are fully open source



https://github.com/Azure/Azure-Functions

Serverless platform

Services for specific use case | Consistency across developer and operator lifecycle | Integrates with common services



Develop

Inner loop development experience to enhance productivity

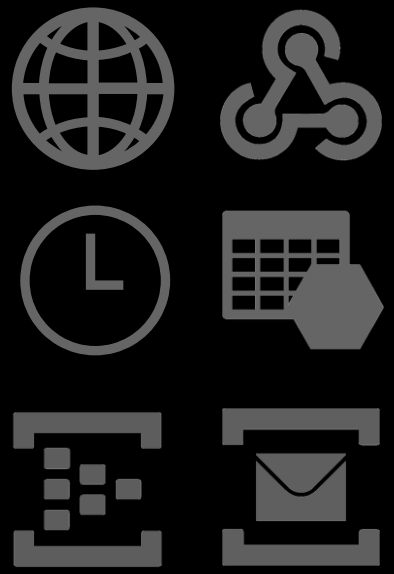
01 Develop / Test / Debug locally

02 Integrate across a broad set of IDEs and command line

03 Push to cloud or CI / CD consistently for end-to-end validation

Azure Functions Programming Model

Events

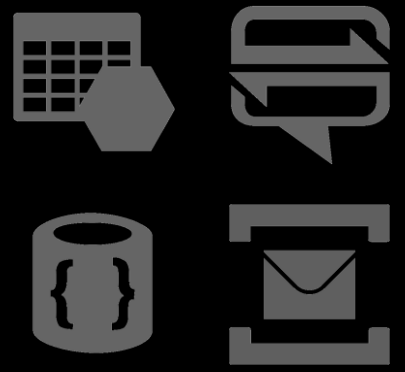


React to timers, HTTP, or events from your favorite Azure services, with more on the way

Code

Author functions in C#, F#, Node.JS, Java, PowerShell, Python, and more

Outputs



Send results to an ever-growing collection of services

Deploy

Setting up a CI / CD pipeline for serverless

01 Cloud resources & application code

02 Support for different pipeline options

03 Extensible for different operations teams needs

Demo

Inner loop development & CI / CD configuration

Monitor

Monitoring in an event driven application

01 Support for distributed tracing

02 Configurable so all logs across applications and infrastructure can be collected in a single place

03 Ability to export logs into other systems

Manage

Consistent management across all resources involved in the app

01 Unified management experience and API for all resources involved in the application

02 Control access to application using role based access control

03 Full SDK against the control and data plane

Govern

Apply organizational policies for serverless applications

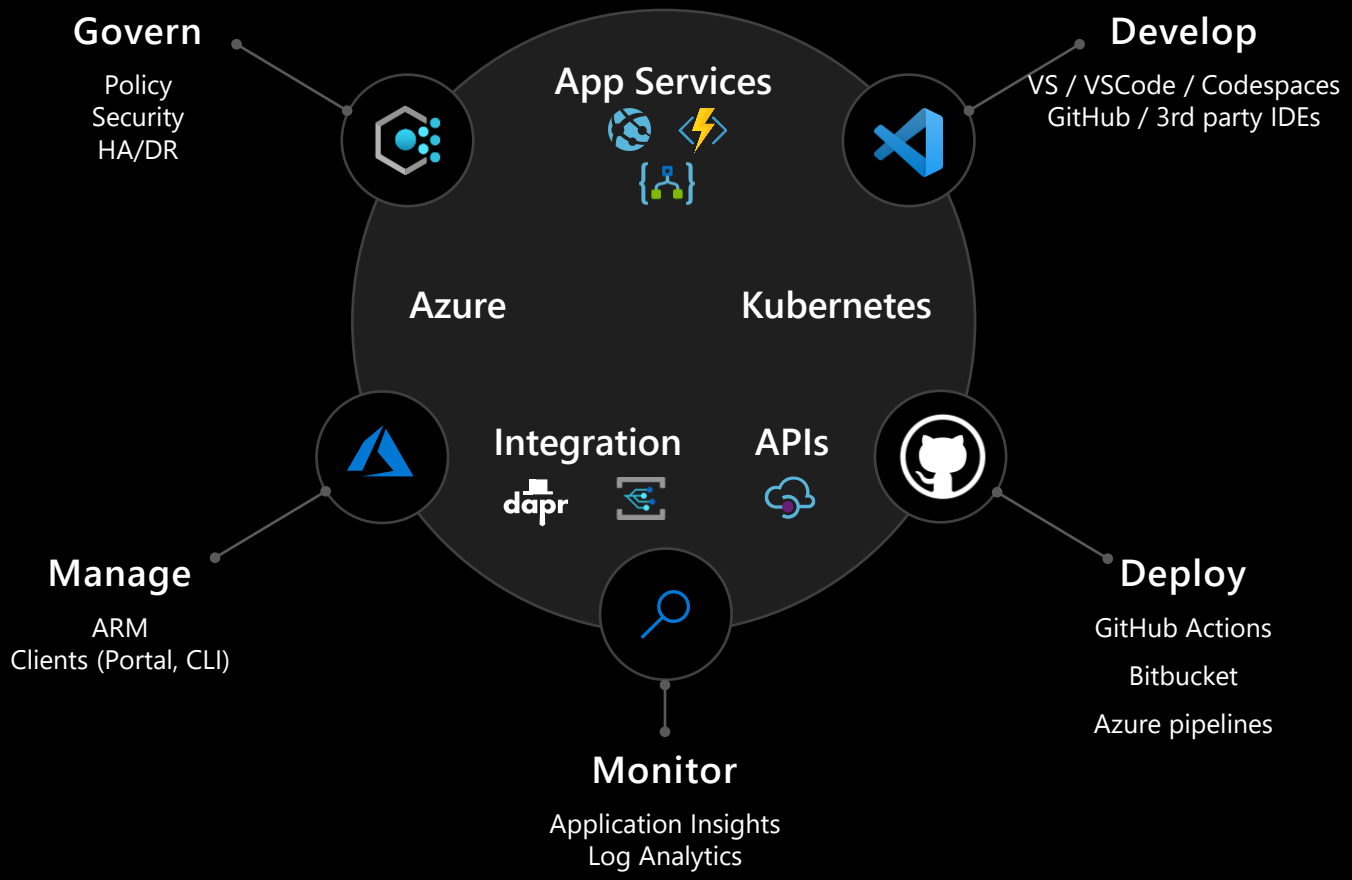
01 Enforce security best practices

02 Enable disaster recovery

03 Apply policies to enforce consistency

Serverless platform

Services for specific use case | Consistency across developer and operator lifecycle | Integrates with common services



Key takeaways

Serverless computing

01 Event driven architecture benefits

02 Focus on business logic and not infrastructure

03 Full application lifecycle support is critical when building large scale distributed applications.

Q & A