Hack Together from 16 August

JavaScript on Azure

aka.ms/JavaScriptonAzure

© Casey Hroner – Unsplash.com

It hosts
16M
developers
From 1.9 billion websites, 98% use JavaScript
© NASA– Unsplash.com

JavaScript is everywhere

© Adi Goldstein - Unsplash.com

JavaScript Developer Hub @Microsoft

https://developer.microsoft.com/javascript/

Microsoft Azure

DEVELOPERS

Your hosts today

Natalia Venditto

Principal Lead e2e
JavaScript DX on Azure

@anfibiacreativa

Nitya Narasimhan

Senior Cloud Advocate
JavaScript on Azure

@nitya

Attribution: *This talk was first delivered by @anfibiacreativa and @nitya as the Opening Keynote for #JSOnAzure Global Hack – Aug 16, 2023*

Code to Azure: Want to get a head start?

Quickstart

Validate setup with GitHub Codespaces

1. **Visit** - https://aka.ms/contoso-real-estate/github
2. **Fork the repo** (and give us a star!)
3. **Launch Codespaces on `main`** branch

Setup takes around 10 minutes to complete –
and will help ensure your development environment is ready for the hackathon.
We will have a checkpoint later for questions.

👉 👉 **Need help or have questions?**
https://aka.ms/contoso-real-estate/discussions

Learning to Build Complex Apps is Hard!

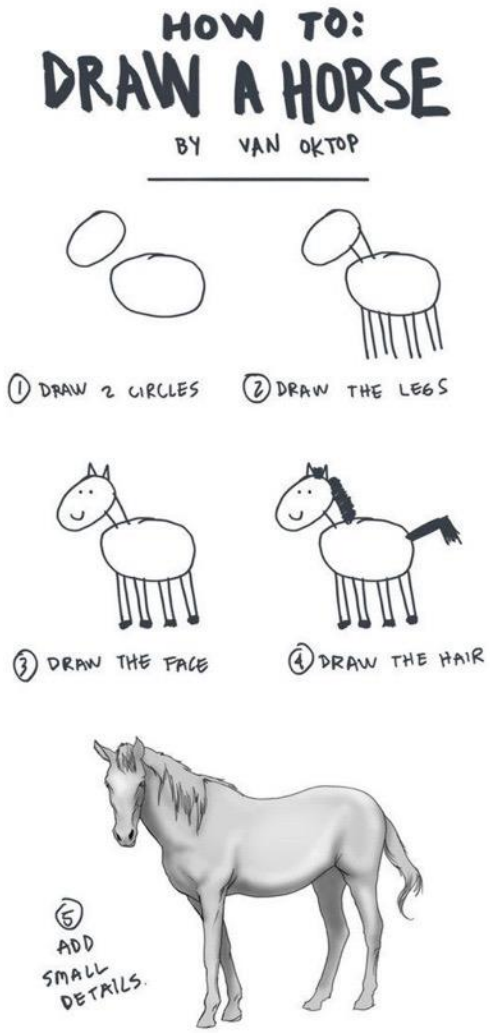


Image Credit: HuffPost Article (2017)

QuickStart – validate dev environment

How-To – learn to use core features or tools

The Missing Trail – how do I go from building blocks ↑ to building apps ↓

Real-World App – build a secure, scalable, multi-scenario, multi-team solution that is reliable, performant!

We need a **paved path** that explains "the small details".

Hello, Contoso Real Estate Sample!

Let's define a

paved path

for JavaScript developers to build an end-to-end composable architecture on Azure!

And let's use

best-in-class

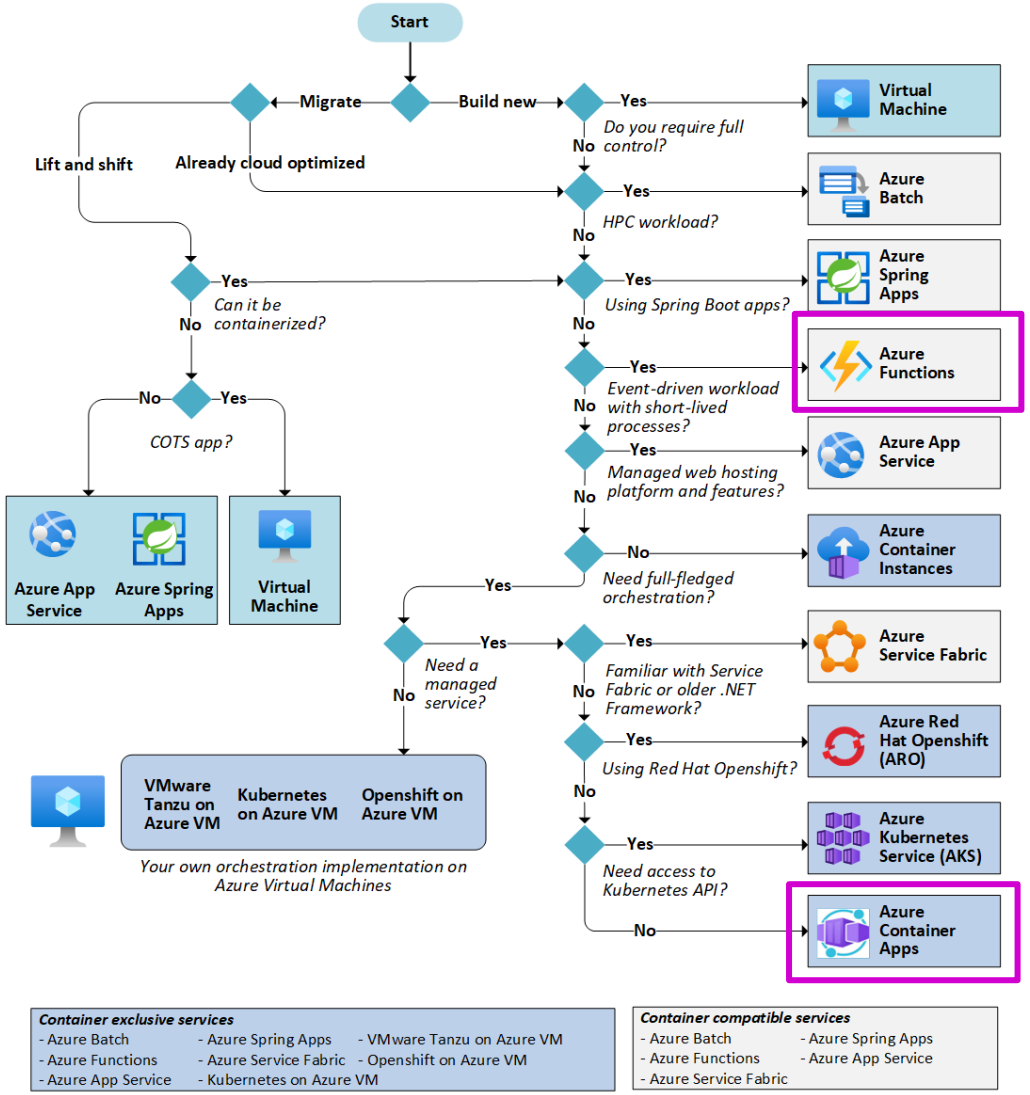
Azure services and Developer Tools to enhance end-to-end developer experience (DX)

- ❑ **Define** – Prioritized Scenarios
- ❑ **Develop** – Composable Architecture
- ❑ **Deploy** – Cloud-native, Serverless, Azure
- ❑ **Deliver** – Architecture, Docs, Dev Experience

Contoso In A Nutshell

https://www.youtube.com/watch?v=GxeENsvwZrI

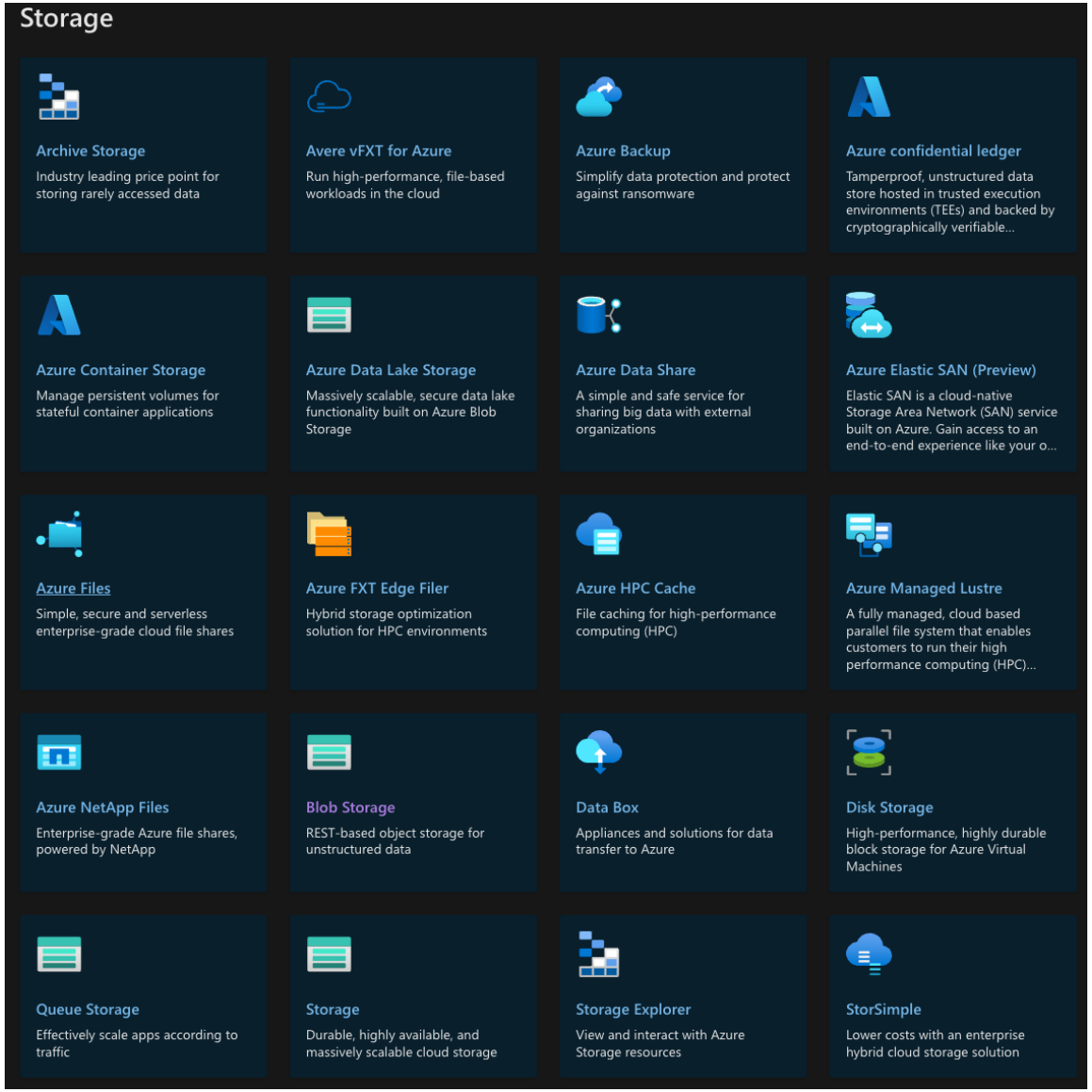
Azure Compute Decisions



What works best for this use case?

- Full-stack static JavaScript applications
- Event-driven workloads
- **Short-lived, stateless tasks**
- Unstructured data
- Intelligent search
- **Containerized 3rd party APIs**

Azure Storage Decisions



What about our data and storage needs?

- Unstructured blob storage – static files
- Structured data storage – dynamic activity
- Secrets storage – keys and credentials
- Content Management System – blog
- Application Insights – log analytics

Why Serverless (On Azure)?

Serverless means using cloud-provider *managed services* while you focus on application logic.
Get *auto-scaling* *(up or down to zero)* and *cost-effective pricing* (pay as you go, for what you use)

Use case:
We are developing and deploying an application that introduces new features and capabilities to an existing platform, the Contoso platform.

No infrastructure management

Using fully managed services enables developers to avoid administrative tasks and focus on core business logic. With a serverless platform, you simply deploy your code, and it runs with high availability.

Dynamic scalability

With serverless computing, the infrastructure dynamically scales up and down within seconds to match the demands of any workload.

Faster time to market

Serverless applications reduce the operations dependencies on each development cycle, increasing development teams' agility to deliver more functionality in less time.

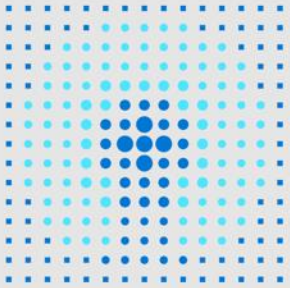
More efficient use of resources

Shifting to serverless technologies helps organizations reduce TCO and reallocate resources to accelerate the pace of innovation.

See: *https://aka.ms/cloud-native/serverless*

Serverless application patterns

Developers build serverless applications using a variety of application patterns—many of which align with approaches that are already familiar—to meet specific requirements and business needs.



Serverless functions

Serverless functions accelerate development by using an event-driven model, with triggers that automatically execute code to respond to events and bindings to seamlessly integrate additional services. A pay-per-execution model with sub-second billing charges only for the time and resources it takes to execute the code.

Serverless Kubernetes

Developers bring their own containers to fully managed, Kubernetes-orchestrated clusters that can automatically scale up and down with sudden changes in traffic on spiky workloads.

Serverless workflows

Serverless workflows take a low-code/no-code approach to simplify orchestration of combined tasks. Developers can integrate different services (either cloud or on-premises) without coding those interactions, having to maintain glue code, or learning new APIs or specifications.

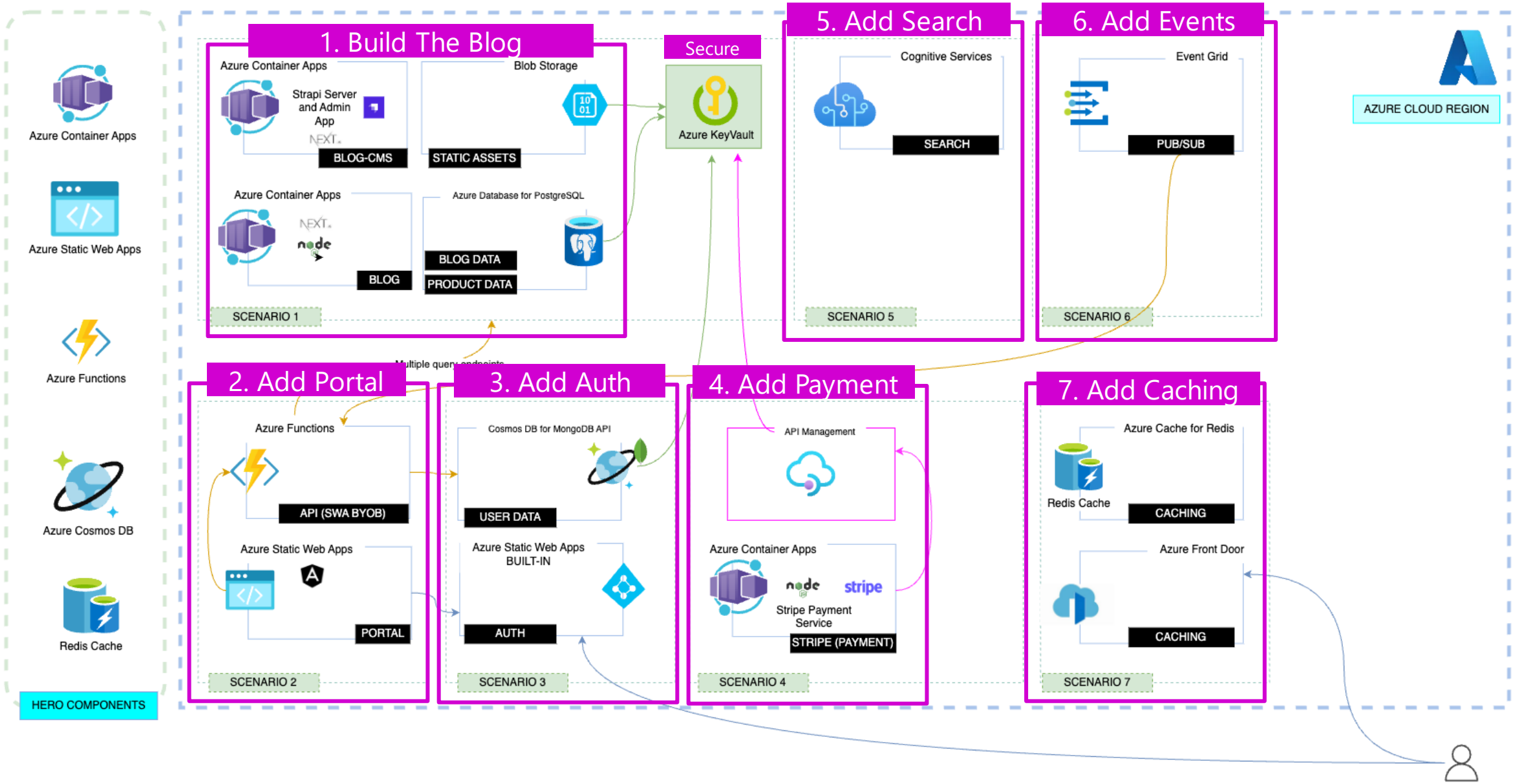
Serverless application environments

With a serverless application environment, both the back end and front end are hosted on fully managed services that handle scaling, security, and compliance requirements.

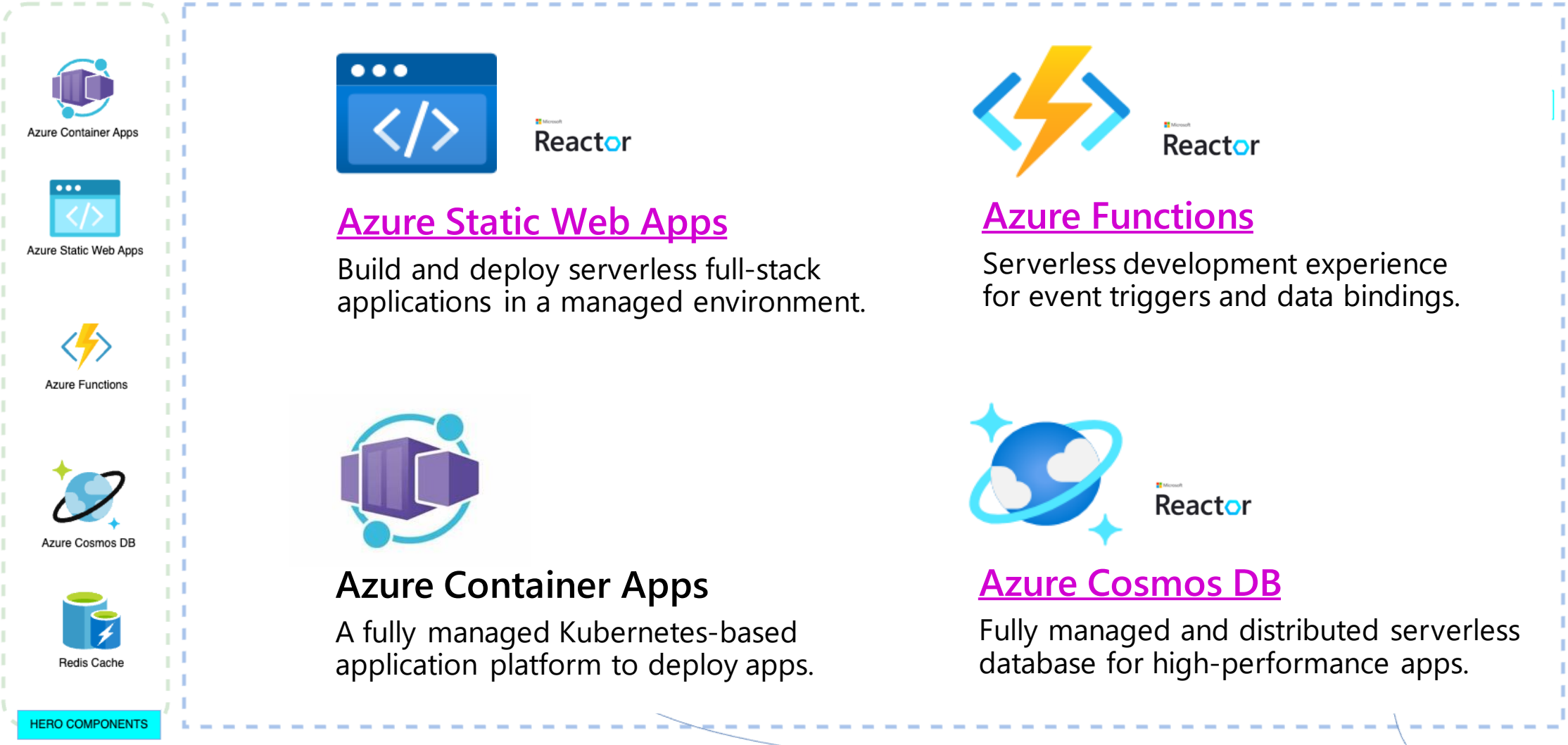
Serverless API gateway

A serverless API gateway is a centralized, fully managed entry point for serverless backend services. It enables developers to publish, manage, secure, and analyze APIs at global scale.

Why Composable Enterprise Architecture?



Serverless Hero Services



Reactor

Azure Static Web Apps

Build and deploy serverless full-stack applications in a managed environment.

Reactor

Azure Functions

Serverless development experience for event triggers and data bindings.

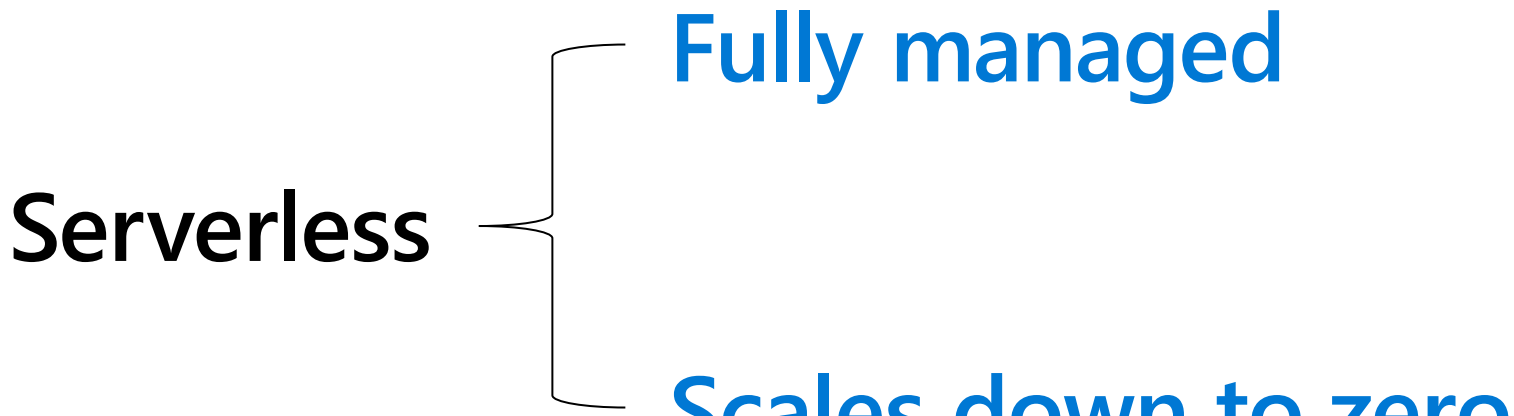
Azure Container Apps

A fully managed Kubernetes-based application platform to deploy apps.

Reactor

Azure Cosmos DB

Fully managed and distributed serverless database for high-performance apps.



Reduced TCO (Total Cost of Ownership)

Serverless + Intelligent Apps

Reactor

[GitHub Copilot]

The world's most trusted and widely used AI development assistant tool.

Using AI to improve the developer experience for authoring, testing, debugging and task automation

Reactor

[Azure OpenAI Service]

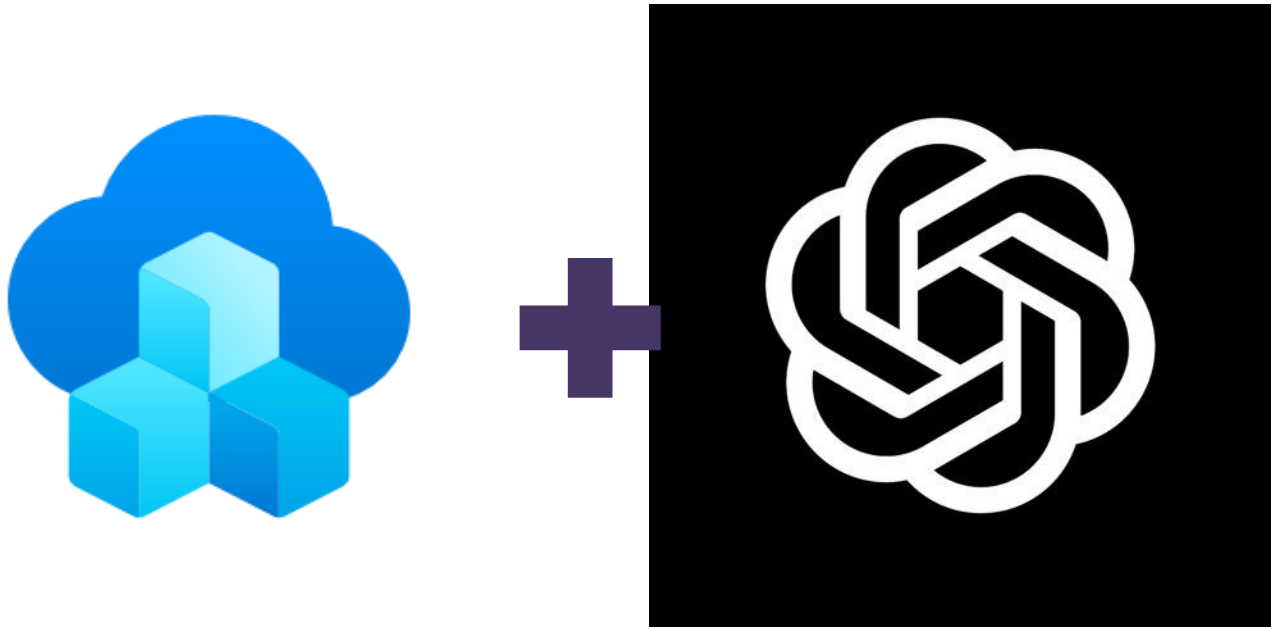
Large scale generative AI models that enable apps with new reasoning capabilities.

Using AI to improve the end-user experience by integrating AI-generated content, AI-driven conversations and AI-enabled suggestions

What's new!

© Kelli McClintock – Unsplash.com

Azure SDK for JavaScript | OpenAI



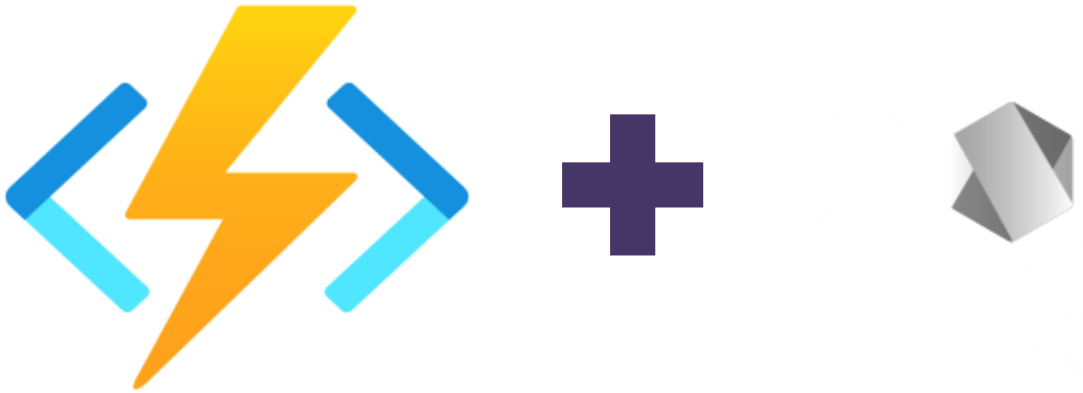
Libraries - OpenAI API

azure-sdk-for-js/sdk/openai/openai at main · Azure/azure-sdk-for-js (github.com)

Azure | Node.js LTS

Node.js version 20 will be available in public preview for Azure functions, this September.

Azure Functions | Node.js programming model v.4



[Migrate apps from Azure Functions version 3.x to 4.x | Microsoft Learn)]

[Step by step guide to migrating from v.3 to v.4 for Contoso Real Estate]

[Azure Functions: Version 4 of the Node.js programming model is in preview - Microsoft Community Hub]

Azure Static Web Apps | Next.js hybrid rendering

Next.js support on Azure Static Web Apps | Microsoft Learn

Tutorial: Deploy hybrid Next.js websites on Azure Static Web Apps | Microsoft Learn

Azure Cosmos DB | Data API Builder

| Diagnostic logs using JavaScript SDK

| Vercel seamless integration

Announcing Data API builder for Azure Cosmos DB - (microsoft.com)

azure-sdk-for-js/sdk/cosmosdb/cosmos/README.md at main · Azure/azure-sdk-for-js (github.com)

https://aka.ms/Vercel-Integration

Playwright

Reliable, E2E Testing for Modern Web Apps

playwright.dev | Read the Docs
dev.to/playwright | Catch up on News
aka.ms/playwright/discord | Community
github.com/microsoft/playwright | Repo

Cross-Browser

Cross-Language

Code to Azure: Development Walkthrough

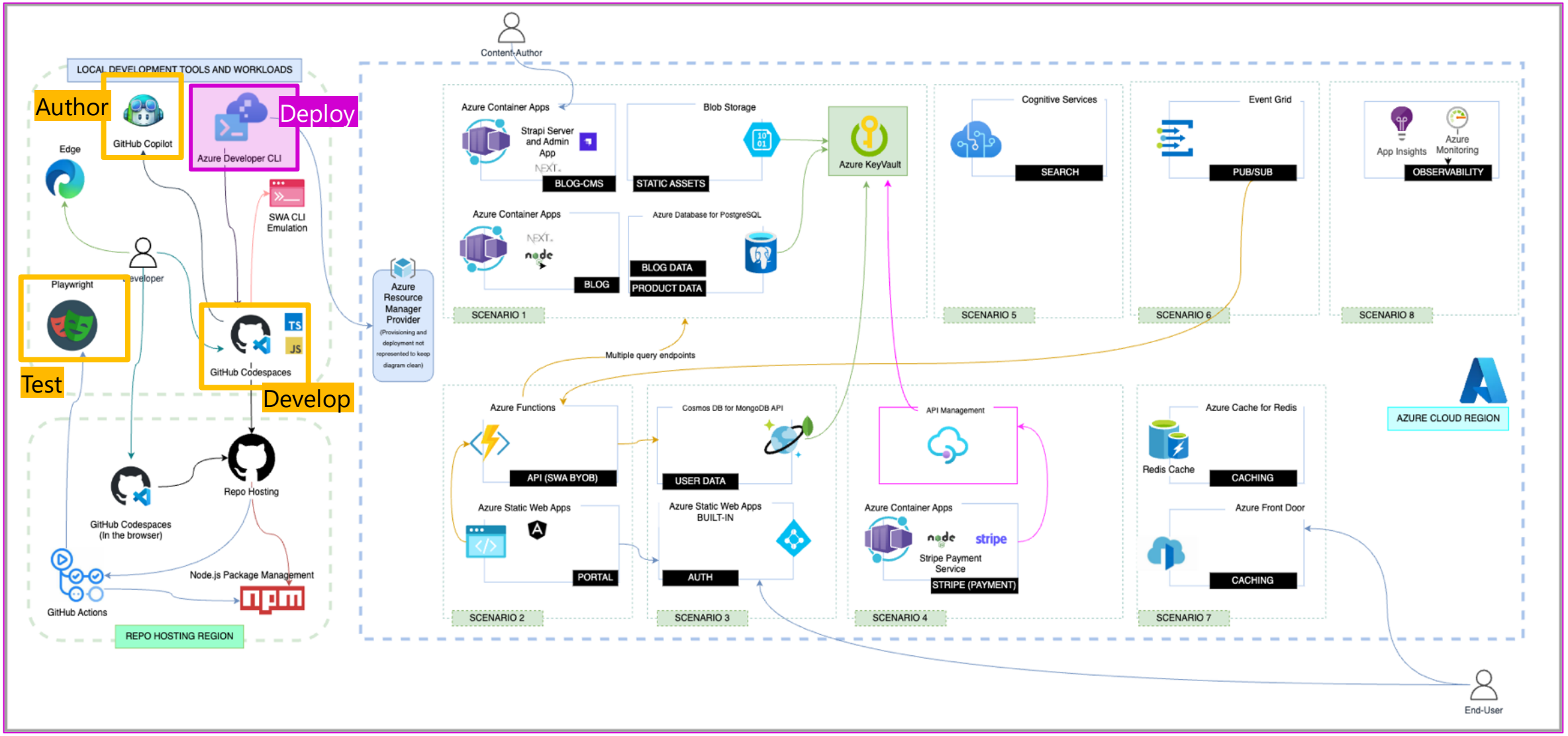
Checkpoint

Stop Your Running Codespaces

(cost optimization)

👉 👉 **Need help or have questions?**
https://aka.ms/contoso-real-estate/discussions

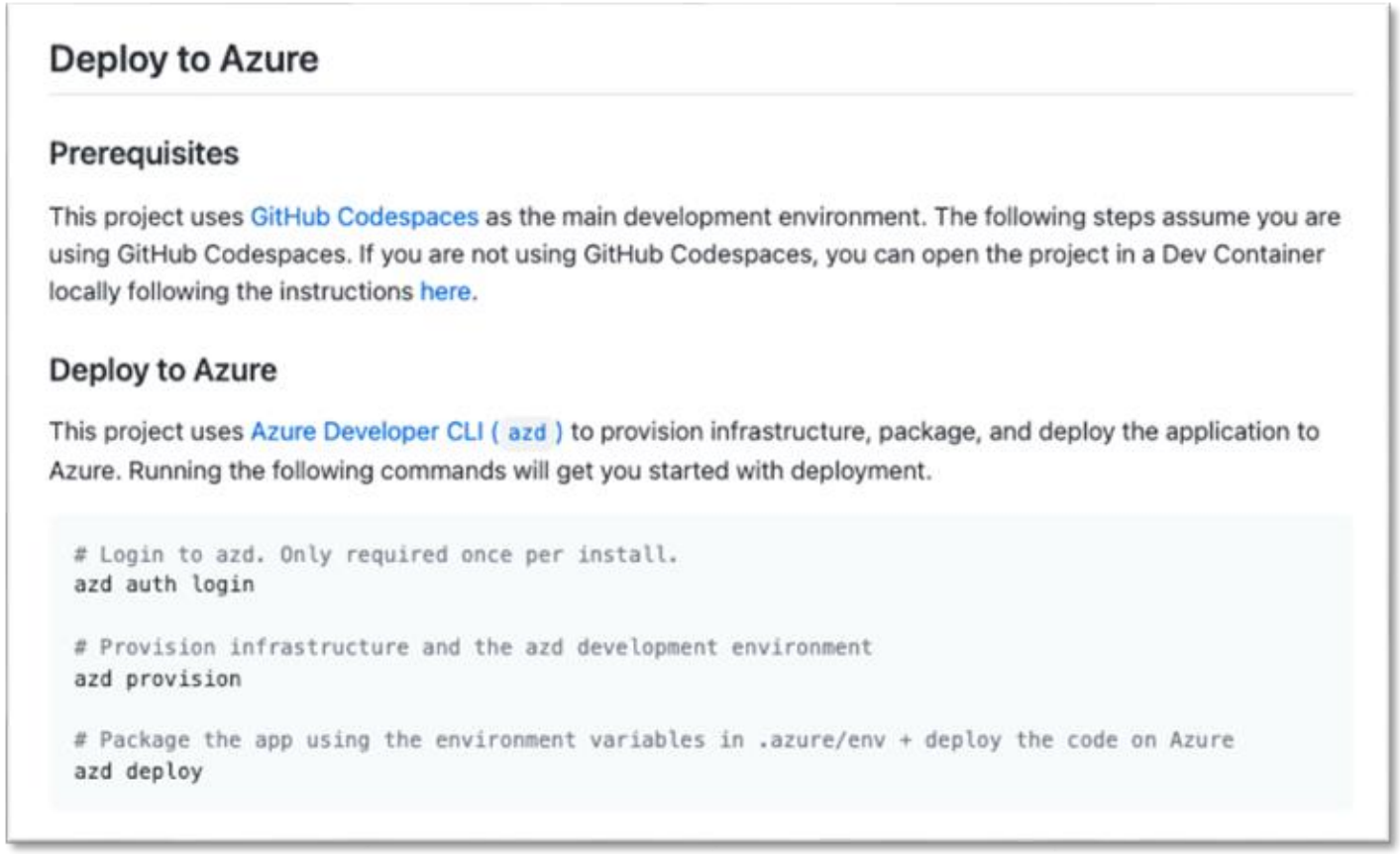
Code to Azure: Developer Tooling



Code to Azure: Deployment Guidelines

Deploy App with Azure Developer CLI

You will need an active Azure subscription to try this on your own later. Please refer to the documentation for a discussion on costs and suitable regions for successful deployment.



Deploy to Azure

Prerequisites

This project uses GitHub Codespaces as the main development environment. The following steps assume you are using GitHub Codespaces. If you are not using GitHub Codespaces, you can open the project in a Dev Container locally following the instructions here.

Deploy to Azure

This project uses Azure Developer CLI (`azd`) to provision infrastructure, package, and deploy the application to Azure. Running the following commands will get you started with deployment.

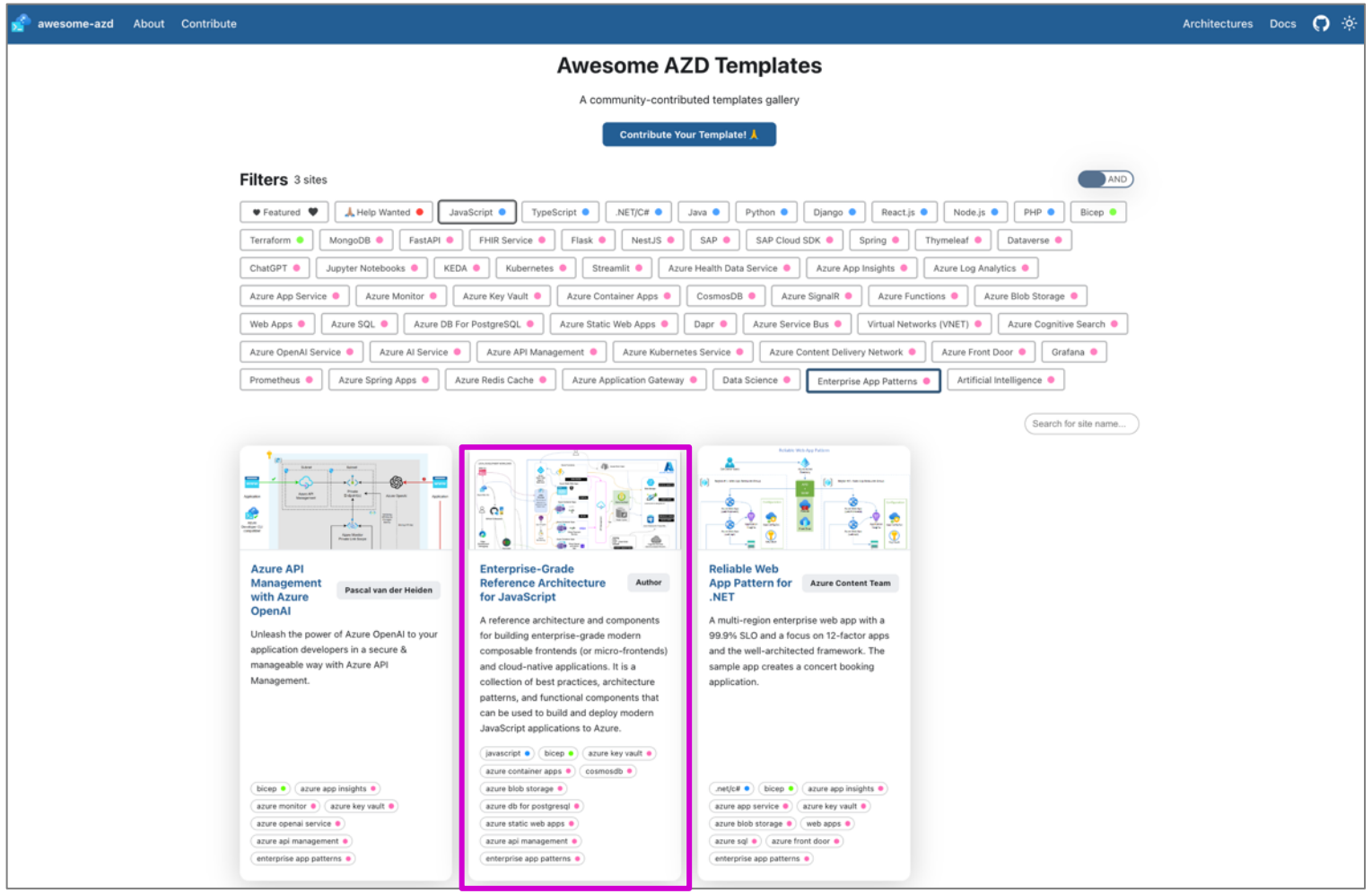
```
# Login to azd. Only required once per install.
azd auth login

# Provision infrastructure and the azd development environment
azd provision

# Package the app using the environment variables in .azure/env + deploy the code on Azure
azd deploy
```

Code to Azure: AZD Templates Gallery

Discover, Use & Deploy Templates

The awesome-azd gallery provides a quick look-up directory for discovering AZD-compliant templates for various languages and architectures. Find featured templates – or contribute your own

https://aka.ms/awesome-azd

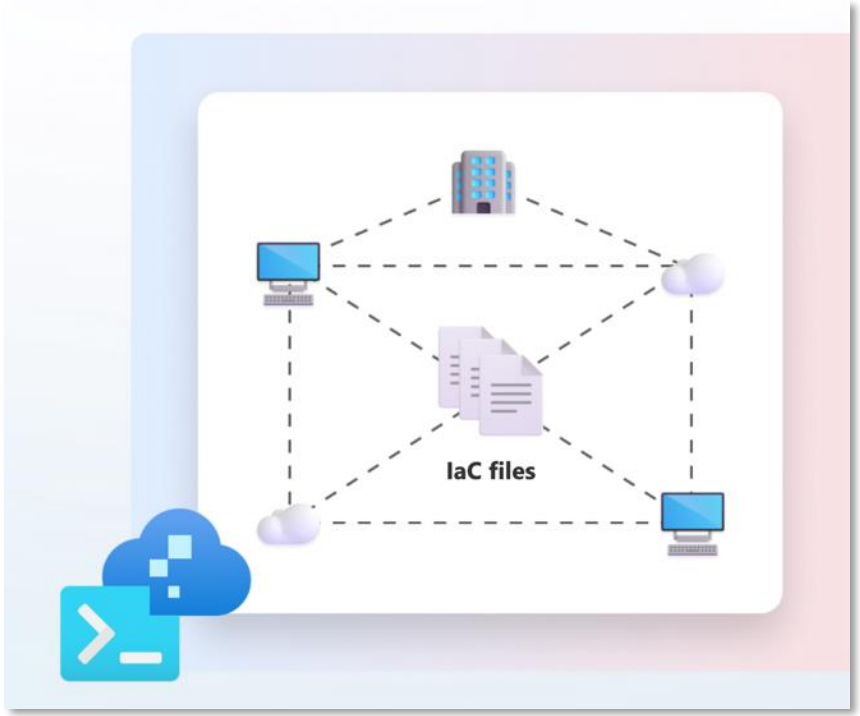


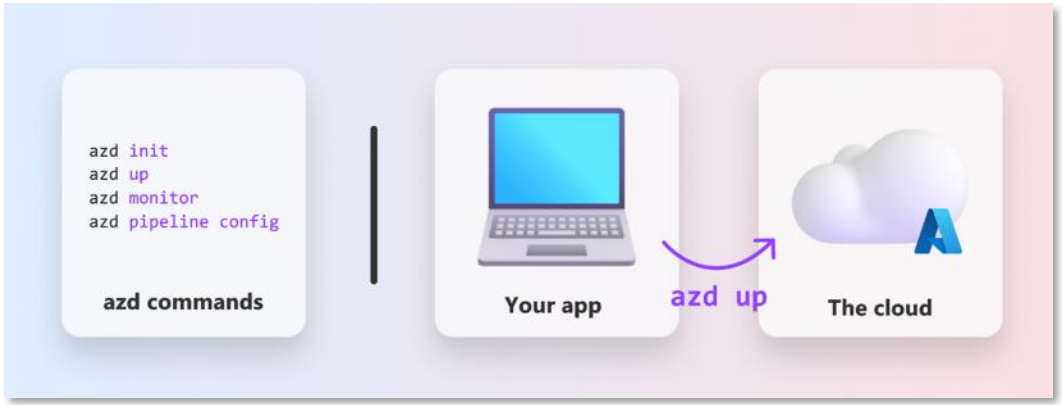
Code to Azure: Deployment Guidelines

Why Azure Developer CLI?

Infrastructure As Code

Infrastructure as code (files) enable version control. Declarative IaC enables **repeatability and reusability**

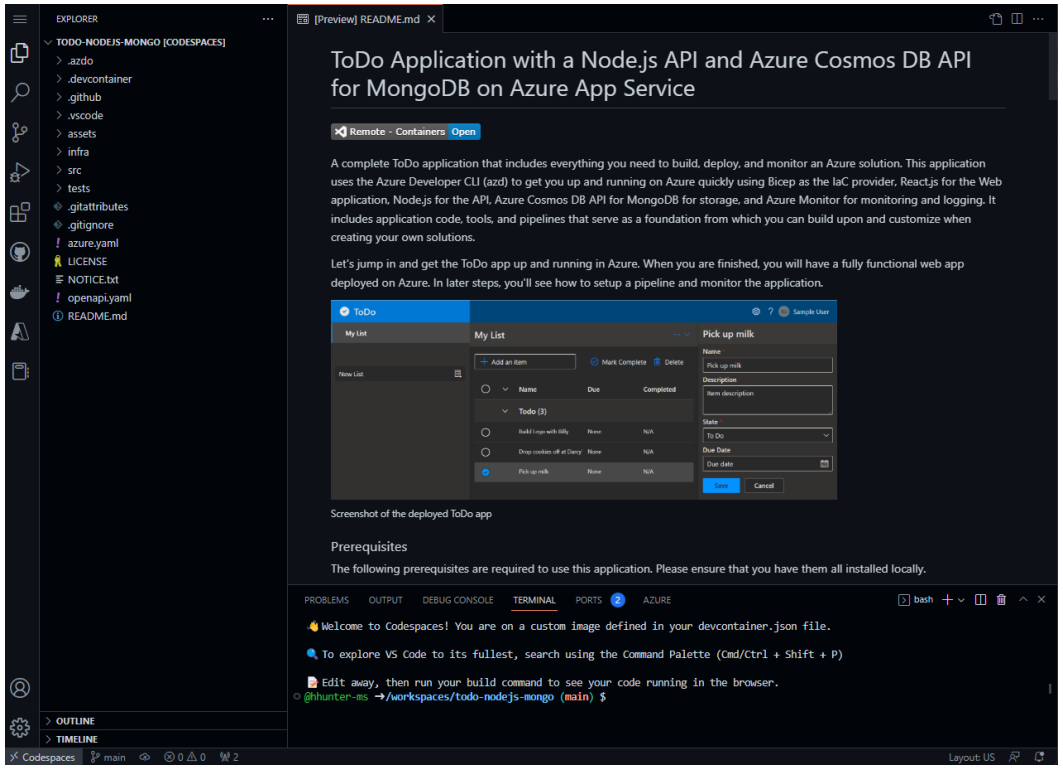




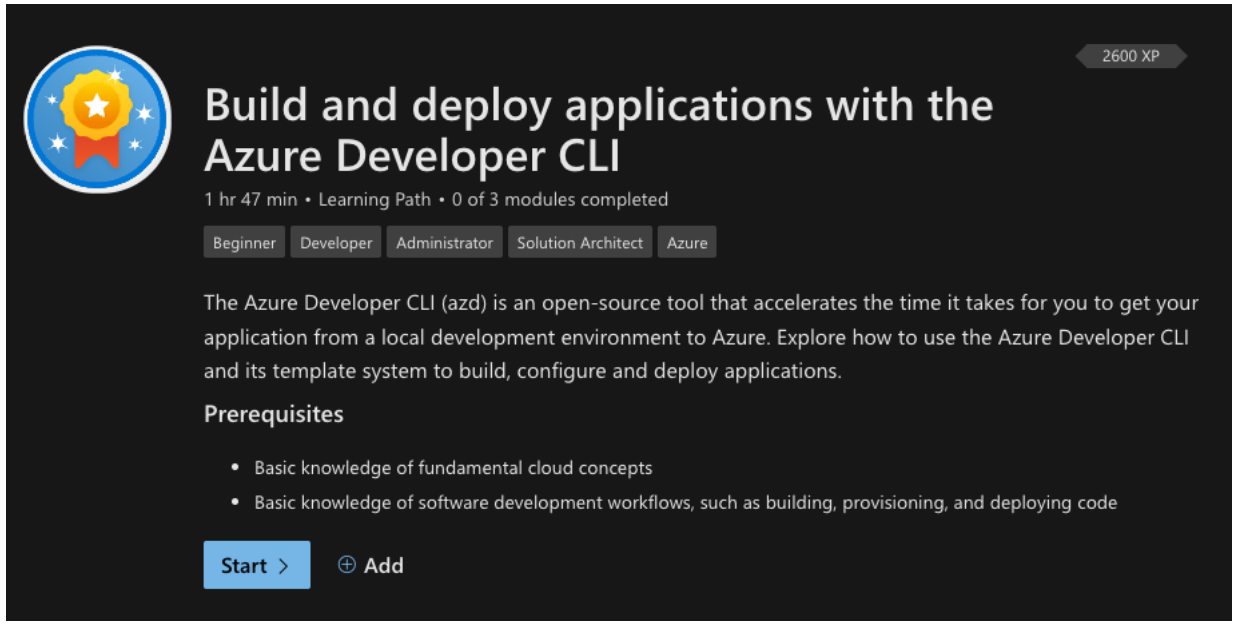
Seamless Developer Experience

From local development (code) to Azure deployment (cloud) **in one step**

Code to Azure: Keep on learning IaC



https://learn.microsoft.com/training/**paths/azure-developer-cli/**



Docs | Tutorial

Learn to deploy existing AZD Template

Training | Learn Path

Learn to build your first AZD Template!

Open-Source Contribution Guidelines

Visit - https://aka.ms/contoso-real-estate-github

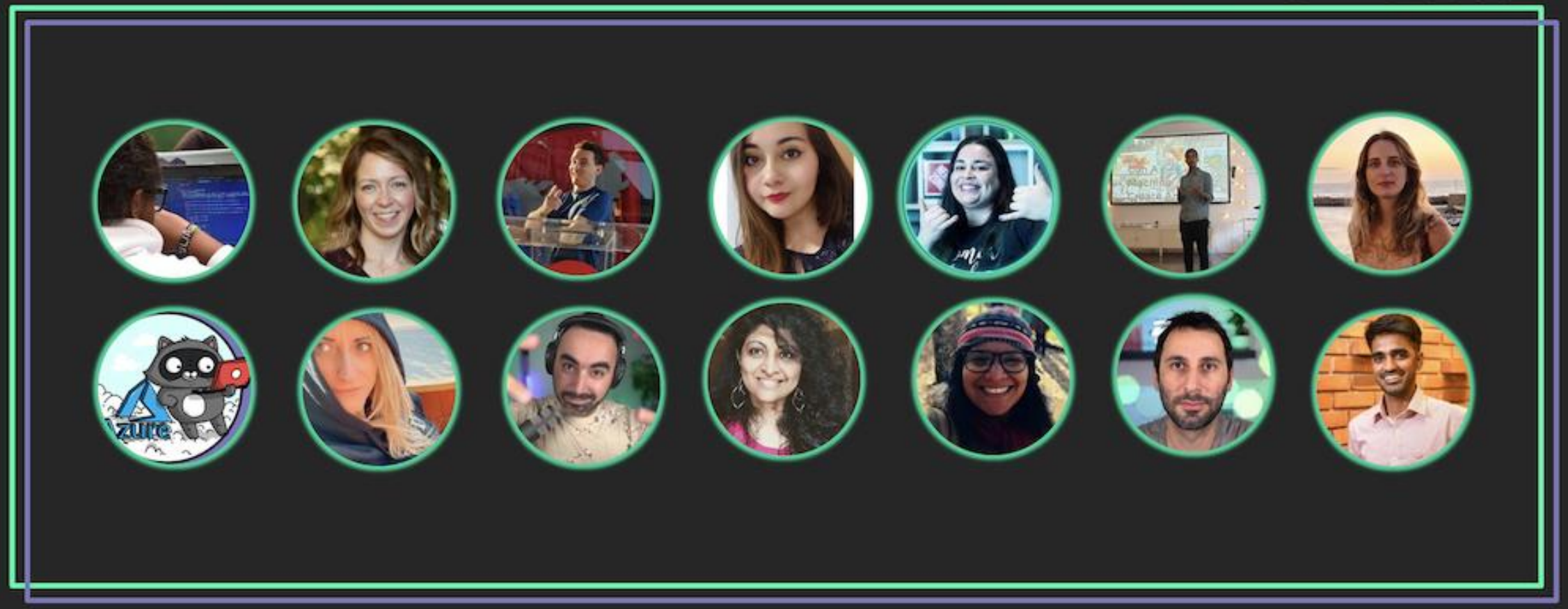
Login and fork the repo

Launch Codespaces on `main` branch

And start contributing, today!

Microsoft

Thank You For Learning & Hacking With Us!



#HackTogether | Learn to build JavaScript Apps On Azure