6 Reasons why Azure Container Apps is a game changer.

On Nov 9 2021, Microsoft released a preview of its Azure Container Apps Service. Here are some of the reasons why we think this service is going to be a game-changer for building microservice-style applications in Azure.

1. Simplified deployments and application management

Standing up new applications in Azure Container Apps is simple. Deploy as many services as your application requires into each environment and manage their lifecycle with your favourite DevOps platform.

Under the hood, your workloads are running on Kubernetes, but you won't need to be an expert in Kubernetes manifests, or Helm charts – Use the Azure CLI, Arm/Bicep or other IaC tools that you are already familiar with to deploy and upgrade your services.

2. Built-in load-based scaling

Scaling your application on-demand is a breeze with built-in support for the Kubernetes Event Driven Autoscaler (KEDA). Define how many container instances you require based on queue length, concurrent requests, CPU, memory, and many other predefined or custom scalers. Event handlers and other background processing services can scale back to zero when there's nothing to do, dropping the cost to zero for these components - that's a huge win.

3. Security of data in transit

Azure Container Apps can be deployed into VNETs in your Azure subscription thereby allowing you to access your resources securely via private endpoints, giving you the confidence that your data will not travel across the public internet when interacting with your services.

Container apps can be accessed by external ingress endpoints with TLS enabled by default. Internal communication between services will not leave Azure, even when using service-to-service communication via the external ingress endpoints.

4. Revisions, blue-green deployments, and traffic-splitting

Versioning and rollouts are handled with a declarative and desired-state approach. Simply modify your deployment JSON or run CLI commands and the Container Apps service will manage the lifecycle of your services, creating new revisions and rolling out the changes.

Blue-green deployments and traffic-splitting is also a first-class feature. This is an advanced technique and is quite often difficult to achieve in practice. This capability is now within reach without requiring expert DevOps knowledge.

5. Dapr support out of the box

Dapr is fast becoming a standard way to run distributed applications. Simply enable Dapr for your container app to leverage MTLS, observability, resiliency, service location, state persistence and publish/subscribe. These building blocks would usually be handled specifically via a service-mesh or directly in your application code.

6. Built-in observability and logging

Leverage the Azure Monitor to view your application logs, traces, and metrics. OpenTelemetry instrumentation is applied transparently as part of the service when using Dapr, so you'll be able to explore visual representations of distributed requests across multiple services and resources, giving insights into latency issues or other faults in your application.

www.playtimesolutions.com.au

PlaytimeSolutions