
BY COMMUNITY • FOR COMMUNITY •
#GLOBALAZURE
2024
MELBOURNE EDITION

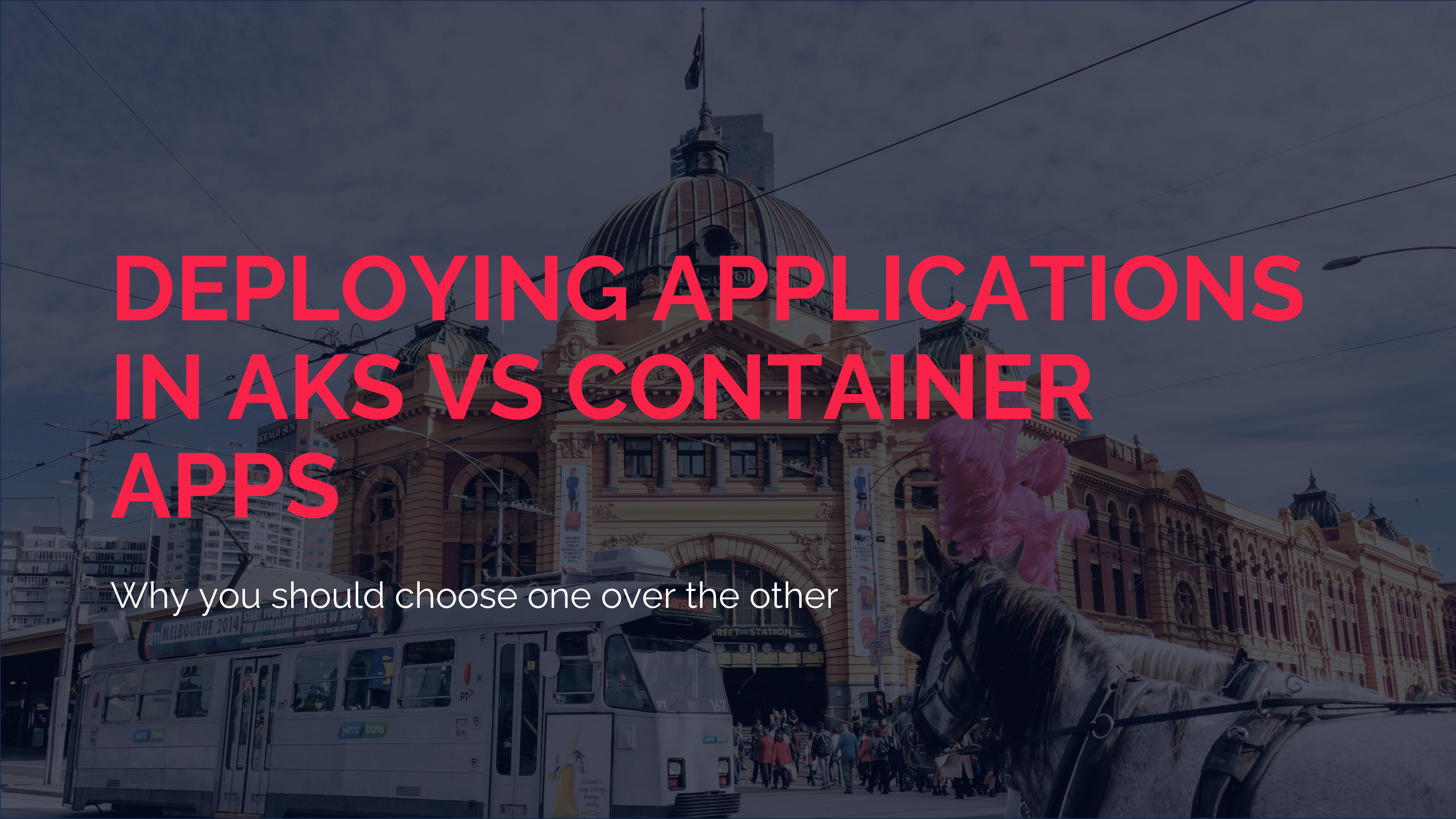
2024 SPONSORS

ARINCO

Mantel group

vNEXT

#GLOBALAZURE 2024



DEPLOYING APPLICATIONS IN AKS VS CONTAINER APPS

Why you should choose one over the other

WHO AM I?

NAME Craig Moyle

WORK Mantel group

SOCIALS https://www.linkedin.com/in/craig-moyle/

PASSIONS

Netball

YouTube/TV/Movies

Music

IT Stuff I guess

#GLOBALAZURE 2024

ABOUT ME

- NAB for 11 years
- Computershare for 7 years
  - Started working in Azure in 2014
- Kloud for 1 year
- Nebulr/Contino for 1 year
- Olikka/Accenture for 2.5 years
- Arinco for 2 years
- Mantel Group for 6 months

Me when my submission got accepted

Me when I realised the Bootcamp
was on Saturday

#GLOBALAZURE
BY COMMUNITY • FOR COMMUNITY
2024

WHAT ARE AKS AND CONTAINER APPS ANYWAY?

AKS

Azure Kubernetes Service (AKS) is a managed Kubernetes service. It simplifies deploying and managing Kubernetes clusters in Azure.

- Simplified Deployment: AKS streamlines the process of creating and maintaining Kubernetes clusters. It offloads operational overhead to Azure, allowing you to focus on your applications.
- Automated Control Plane: When you create an AKS cluster, Azure automatically sets up and configures the Kubernetes control plane. This control plane is provided at no cost as a managed Azure resource, abstracted from the user. You only pay for and manage the nodes attached to the AKS cluster.
- Integration with Azure Services: AKS seamlessly integrates with other Azure services, such as Azure Monitor and Entra ID.

Pros:

- Scalability: Easily scale your applications
- Security: Integrate with Entra ID for identity management and access control.
- Monitoring: Azure Monitor's Container Insights provides health and performance monitoring for AKS clusters.

Cons:

- Some advanced Kubernetes configurations may not be directly accessible.
- Steep engineering effort required to support AKS

WHAT ARE AKS AND CONTAINER APPS ANYWAY?

Container Apps

Azure Container Apps is a serverless platform that simplifies deploying and managing containerized applications.

- Azure Container Apps allows you to run containerized apps without worrying about server configuration, orchestration, or deployment details.
- It's based on Kubernetes and leverages open-source technologies like Dapr, KEDA, and envoy.
- Key features include service discovery, traffic splitting, and event-driven scaling.

Pros:

- Serverless: Requires less infrastructure management, reducing costs.
- Scalability: Dynamically scales based on HTTP traffic, events, or CPU/memory load.
- Integrated Services: Works seamlessly with other Azure services.
- Managed Experience: Abstracts Kubernetes complexities.
- Flexible Deployment: Supports running on demand, scheduled, or event-driven jobs.

Cons:

- Limited Kubernetes Control: No direct access to Kubernetes APIs.
- Vendor Lock-In: Tied to Azure ecosystem.

AKS - Amazing but large and unwieldy and overkill for most

Container Apps – small and nimble

AREN'T CONTAINER APPS A BIT SHIT?

Improvements to Container Apps

Feature	Description
Generally Available: Landing zone accelerators	Landing zone accelerators provide architectural guidance, reference architecture, reference implementations and automation packaged to deploy workload platforms on Azure at scale.
Public Preview: Dedicated GPU workload profiles	Azure Container Apps support GPU compute in their dedicated workload profiles to unlock machine learning computing for event driven workloads.
Public preview: Vector database add-ons	Azure Container Apps now provides add-ons for three open source vector database variants: Qdrant, Milvus and Weaviate.
Public preview: Policy-driven resiliency	The new resiliency feature enables you to seamlessly recover from service-to-service request and outbound dependency failures just by adding simple policies.
Public preview: Code to cloud	Azure Container Apps now automatically builds and packages application code for deployment.

Feature	Description
Generally Available: Dedicated plan	Azure Container Apps dedicated plan is now generally available in the new workload profiles environment type. When using dedicated workload profiles you're billed per compute instance, compared to consumption where you're billed per app.
Generally Available: UDR, NAT Gateway, and smaller subnets	Improved networking features now allow you to have greater control of egress and support smaller subnets in workload profiles environments.
Generally Available: Azure Container Apps jobs	In addition to continuously running services that can scale to zero, Azure Container Apps now supports jobs. Jobs enable you to run serverless containers that perform tasks that run to completion.
Generally Available: Cross Origin Resource Sharing (CORS)	The CORS feature allows specific origins to make calls on their app through the browser. Azure Container Apps customers can now easily set up Cross Origin Resource Sharing from the portal or through the CLI.
Generally Available: Init containers	Init containers are specialized containers that run to completion before application containers are started in a replica. They can contain utilities or setup scripts not present in your container app image.
Generally Available: Secrets volume mounts	In addition to referencing secrets as environment variables, you can now mount secrets as volumes in your container apps. Your apps can access all or selected secrets as files in a mounted volume.
Generally Available: Session affinity	Session affinity enables you to route all requests from a single client to the same Container Apps replica. This is useful for stateful workloads that require session affinity.
Generally Available: Azure Key Vault references for secrets	Azure Key Vault references enable you to source a container app's secrets from secrets stored in Azure Key Vault. Using the container app's managed identity, the platform automatically retrieves the secret values from Azure Key Vault and injects it into your application's secrets.
Public preview: additional TCP ports	Azure Container Apps now support additional TCP ports, enabling applications to accept TCP connections on multiple ports. This feature is in preview.
Public preview: environment level mTLS encryption	When end-to-end encryption is required, mTLS will encrypt data transmitted between applications within an environment.
Retirement: ACA preview API versions 2022-06-01-preview and 2022-11-01-preview	Starting on November 16, 2023, Azure Container Apps control plane API versions 2022-06-01-preview and 2022-11-01-preview will be retired. Before that date, migrate to the latest stable API version (2023-05-01) or latest preview API version (2023-04-01-preview).
Dapr: Stable Configuration API	Dapr's Configuration API is now stable and supported in Azure Container Apps. Learn how to do Dapr integration with Azure Container Apps

WHAT METHODS CAN BE USED TO DEPLOY APPS?

AKS

- Bicep via the Bicep extensibility Kubernetes provider (preview)
- Manifest file via kubectl
- CICD - Manifest file via GitHub Actions (Azure/k8s-deploy) or Azure Pipelines (KubernetesManifest) – can also use Helm charts

Container Apps

- Azure CLI via *az containerapp up*
- Bicep via a Bicep file using the *Microsoft.App/containerApps* resource
- Terraform using the *azurerm_container_app* resource
- CICD via GitHub Actions (*azure/container-apps-deploy-action@v1*) or Azure Pipelines (*AzureContainerApps@1*) which support:
  - Build from a Dockerfile and deploy to Container Apps
  - Build from source code without a Dockerfile and deploy to Container Apps. Supported languages include .NET, Java, Node.js, PHP, and Python
  - Deploy an existing container image to Container Apps
- Integration into GitHub Actions using *az containerapp github-action*
- No access to Container Apps via kubectl

DEMO 1

Setup AKS Cluster

Azure Verified Modules https://azure.github.io/Azure-Verified-Modules/

Mantel group

DEMO 2

Deploy a basic app to both AKS and Container Apps using various methods

Container Apps

- Az CLI
- VS Code
- Bicep
- Terraform

AKS

- Kubectl
- Bicep provider

Mantel group

DEMO 2

How to deploy a basic app to both AKS and Container Apps using Azure Pipelines

Mantel group

SUMMARY

So, what's better?

Well, I'm a consultant so… it depends.

AKS – best suited for companies with a lot of container-based applications, a requirement for scaling and direct access to the Kubernetes APIs and with an engineering team that can maintain the environment.

Container Apps – best suited for companies with a smaller estate of container-based apps that want a more managed approach to deploying containers or who do not have the internal capacity or desire to deal with the complexities of running Kubernetes clusters.

Mantel group

QUESTIONS

#GLOBALAZURE 2024

Mantel group