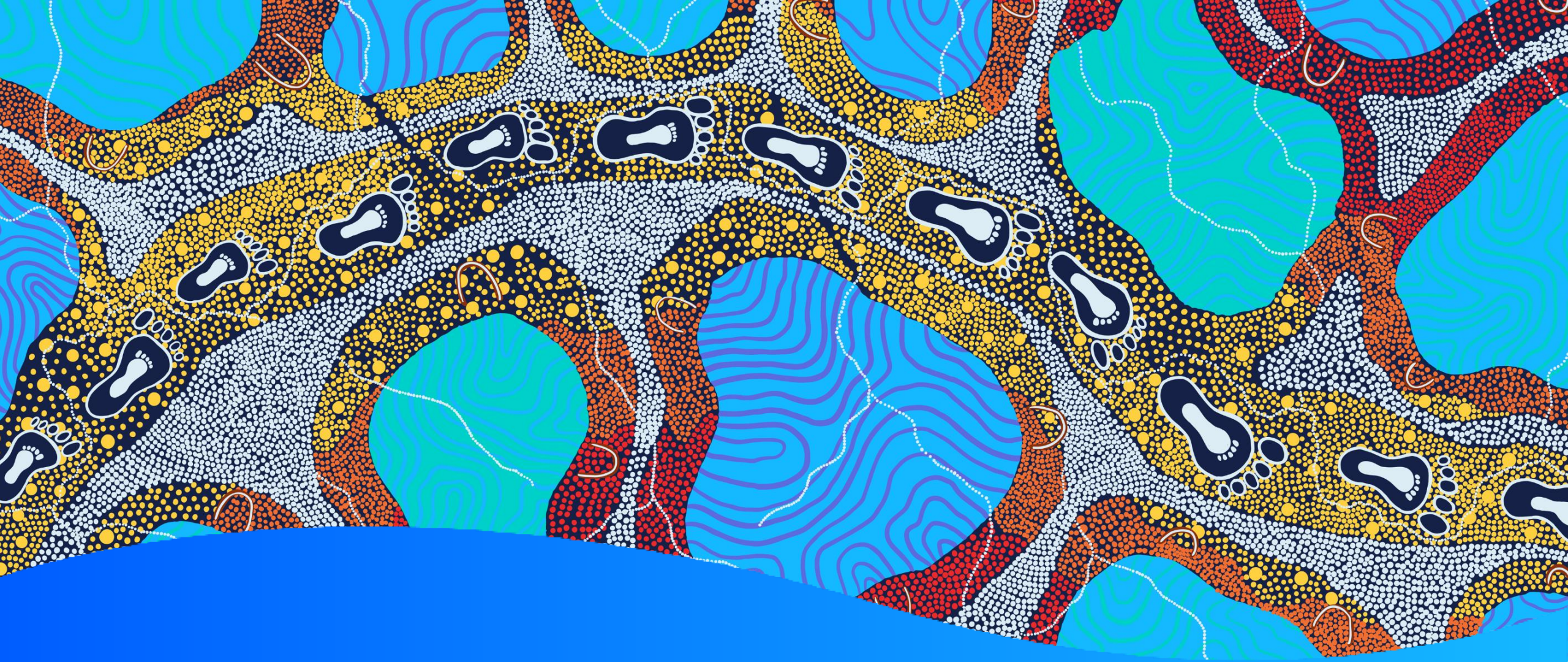


Day 5 – Azure Container Apps

Container hosting in Azure





HSS acknowledges the traditional custodians throughout Western Australia and their continuing connection to the land, waters and community. We pay our respects to all members of Aboriginal communities and their cultures, and acknowledge the wisdom of Elders both past and present.

Quick Recap of Day 4

1. Serverless – a way of computing by abstracting the underlying infrastructure implementation from the developers. Don't worry about infrastructure and concentrate on your code.
2. Azure Functions
 1. It's a "compute on demand" type of "application" where it reacts to an event (trigger), process the event action) and produce some outputs from the processing
 2. Different Hosting type: Consumption plan, Premium plan and Dedicated plan (aka App Service Plan)
 3. Trigger – event that the Function reacts to
 4. Action – Perform something against the event
 5. Output – Outcome of the processing
 6. Durable Function – Extensions of Azure Function to allow for stateful function
3. Homework solution

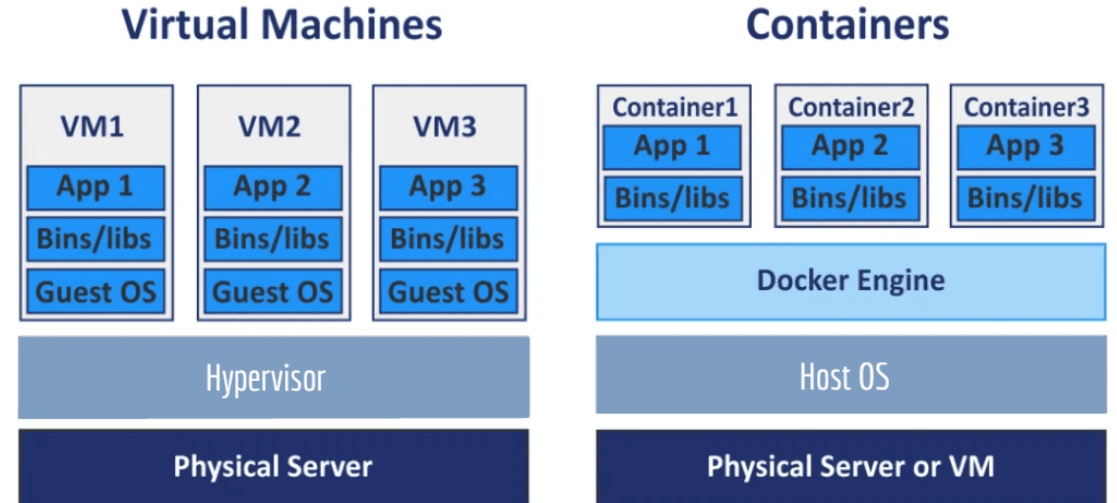
Azure Container Apps



HSS health
support
services

What is container?

From **Docker**: A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.



What makes up a container?

For your container to run, it must have:

1. Your application code or compiled code
2. Runtime (.NET, NodeJS, PHP etc.)
3. System tools (NGINX, Kestrel etc.)
4. System libraries – any libraries your application need
5. Configurations – networking settings (port number, IP address), app-related configurations (connection string, endpoint URL etc.)

How to host containers in Azure?

1. To store your container image:
 1. Azure Container Registry (ACR) – a repository to store all your available container images
2. To host your container applications
 1. Azure App Service (AS) – as mentioned before, you can host containers using App Service
 2. Azure Container Instance (ACI) – Fully managed serverless container hosting environment to host your containers without orchestration
 3. Azure Container Apps (ACA) – Fully managed serverless container **service** for building and deploying applications at scale
 4. Azure Kubernetes Services (AKS) – Fully managed Kubernetes platform to provide container orchestration service for your containers

Azure Container Apps – features & concepts

1. A fully managed container services by Azure that has the following features:
 1. Ability to run multiple revisions of your container
 2. Autoscale based on CPU or memory load, HTTP traffic, event, KEDA-supported scaler
 3. Enable HTTPS ingress without any infrastructure – support TLS certificates, port 443 routing
 4. Split HTTP traffic across different revisions – enabling Blue-Green deployment and testing
 5. Dapr support out-of-the-box which is a Microservices building framework
 6. Virtual Network (VNET) integration
 7. Securely manage secrets directly from the application

Playtime!

1. Go to Azure Portal and create a Container Registry in the resource group. This will be used for you to publish a Docker image
 1. Choose Basic SKU
 2. Once created, go to Access Keys and enabled “Admin User”. Note down the username and password
2. Go to the folder where you store your WhereIsPatient solution from last week.
3. Make a copy of it to another folder.
4. Open the newly copied WhereIsPatient solution in Visual Studio.
5. Right click on the WhereIsPatient WebAssembly “Server” project and go to “Add Docker Support”
 1. Choose “Linux” on the Target OS page
 2. This should then create a Dockerfile in the project
6. Right click on the same project and then click Publish
 1. Choose Docker Container Registry