



Azure Container Services

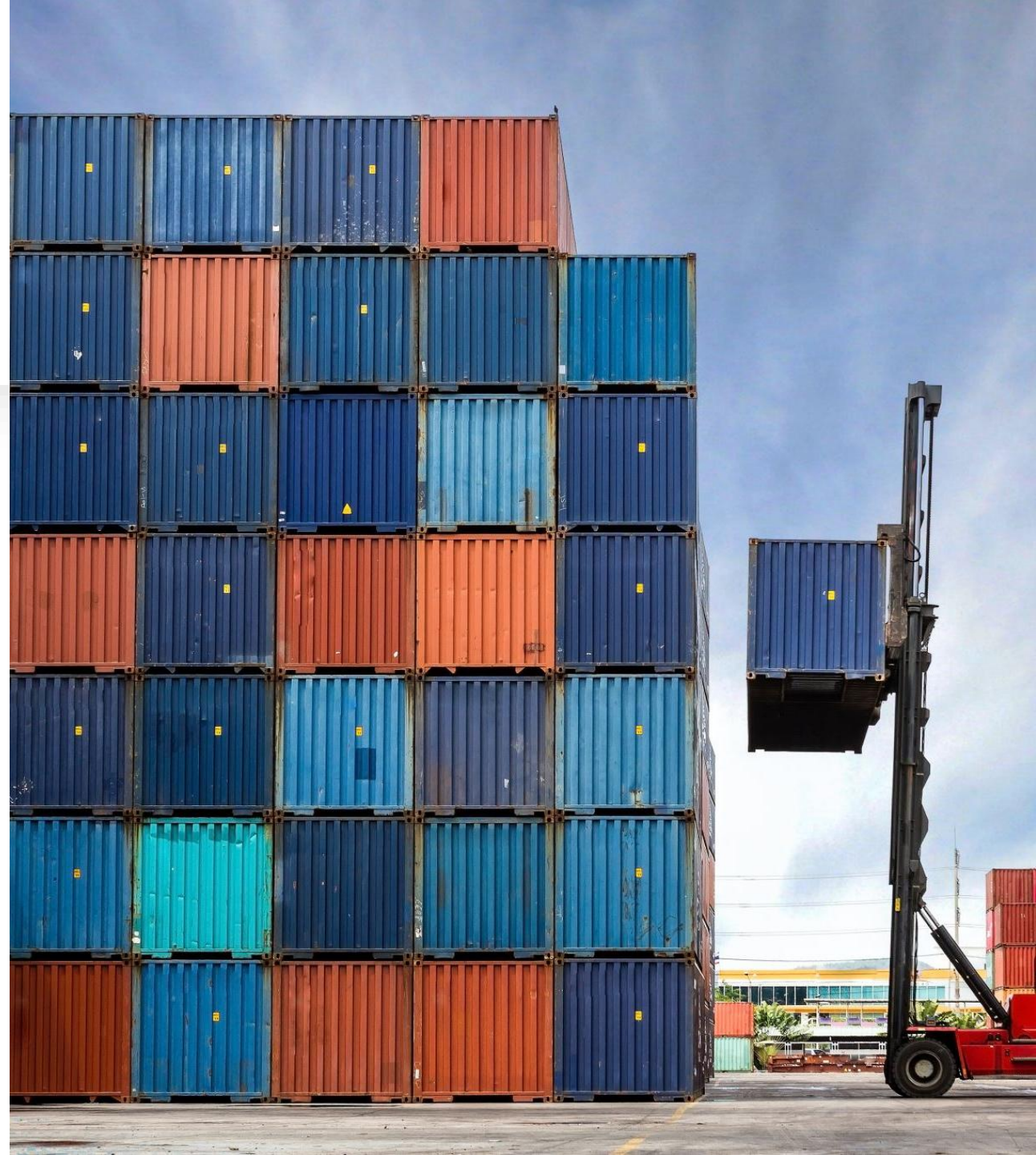
A Comprehensive Guide to Azure's Container Services

- Containers
- Container Registries
- Azure Container Instance Service (ACI)
- ACA Azure Container Apps Service (ACA)
- AKS Azure Kubernetes Service (AKS)
- Summary

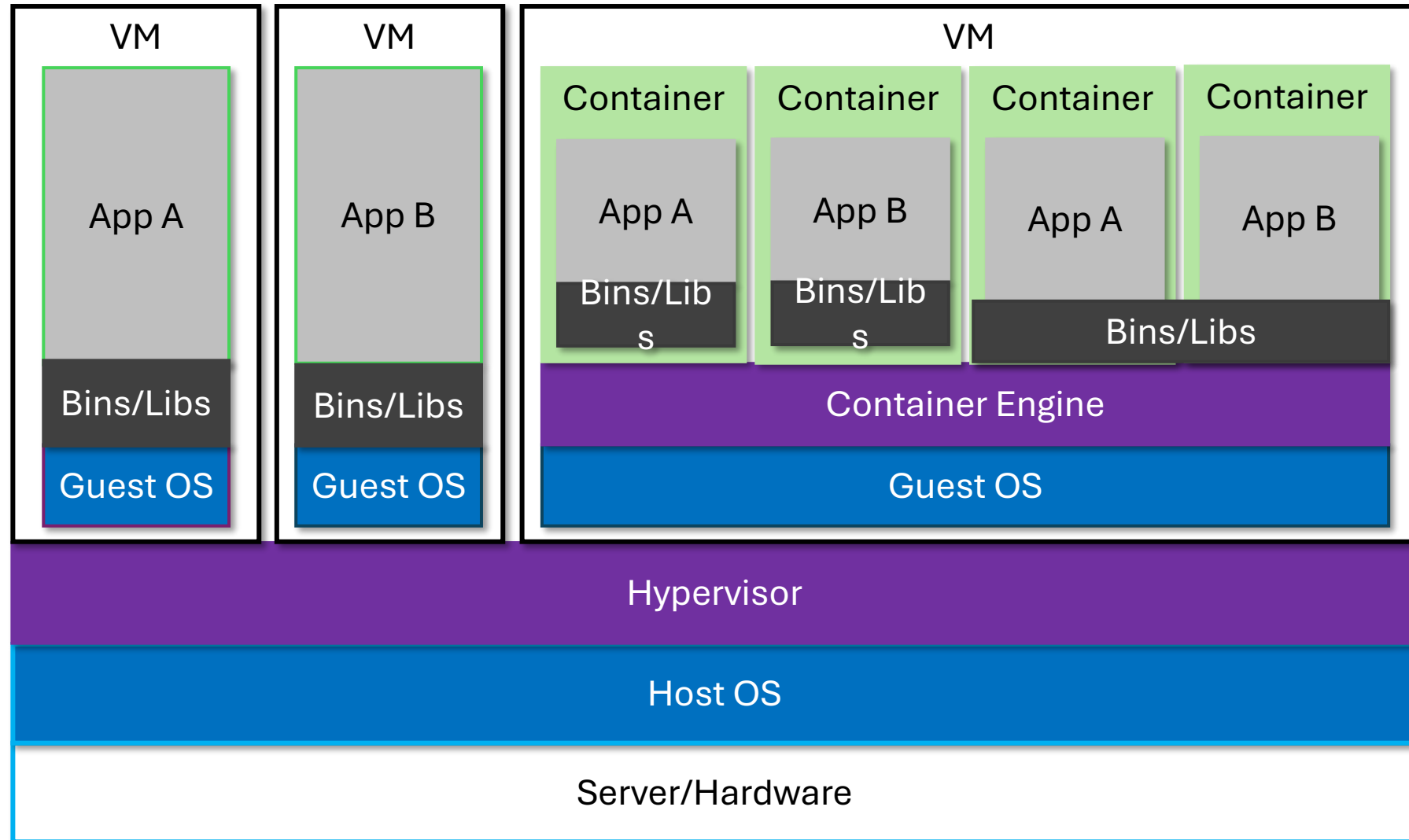
- Containers
- Container Registries
- Azure Container Instance Service (ACI)
- ACA Azure Container Apps Service (ACA)
- AKS Azure Kubernetes Service (AKS)
- Summary

Containers

- Solution to the problem of how to get software to run reliably when moved from one environment to another.
- Immutable
- Lightweight
- Faster Startup



Containers vs VMs

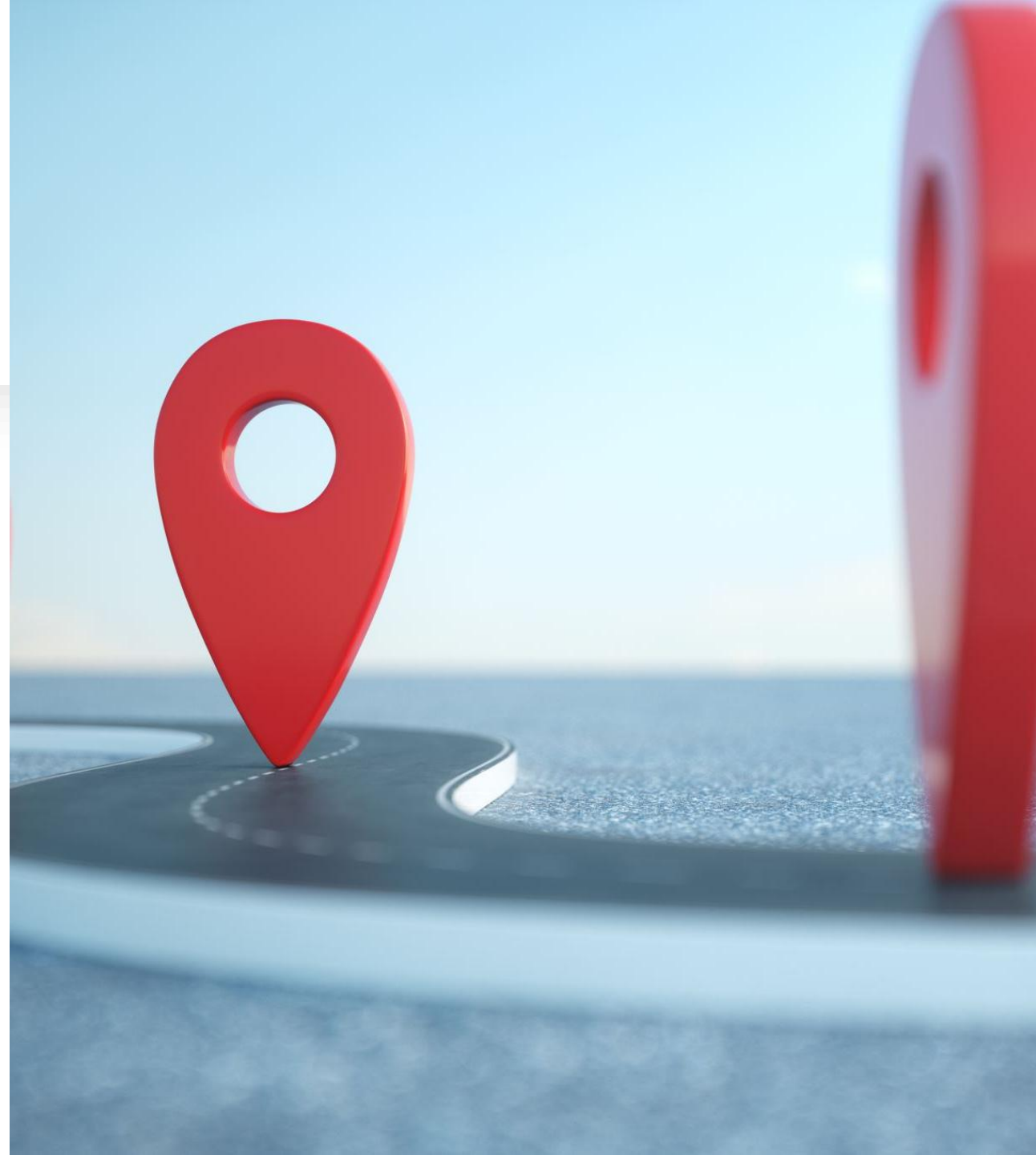


Containers Share
OS
And where Appropriate
Bins/Libs

Setting Up Your Environment

Required tools: Azure CLI, Docker, VS Code

- Azure CLI
<https://learn.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest>
- Docker
<https://www.docker.com/>
- VS Code
<https://code.visualstudio.com/>



Create a Docker Image



Index.html

```
<center><h1><b>Global Azure 2025</b></h1></center>
```

Dockerfile

```
FROM nginx:alpine  
COPY . /usr/share/nginx/html
```

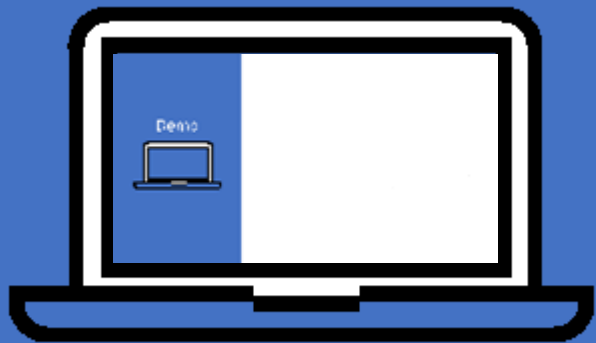
Build Docker Image

```
docker build -t web-image:v1 .
```

Run Container

```
docker run -d -p 80:80 web-image:v1
```

Demo



Creating a Container

Using Container Registries

Container Registries

- DockerHub
- ACR (Azure Container Registry)

Pull/Push images to Registry

- `docker login` **REGISTRY**
- `docker pull/push` **NAME[:TAG]**

Azure Container Registry (ACR)



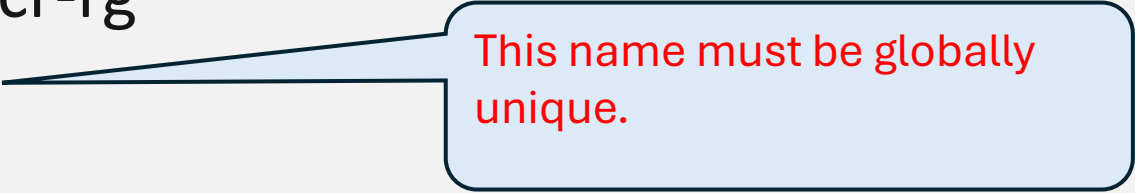
- Private registry for hosting container images
- Store Docker-formatted images

Create a Resource Group

```
az group create --name acr-rg --location eastus
```

Create a container registry

```
az acr create --resource-group acr-rg  
--name mycontainerregistry  
--sku Standard
```



This name must be globally unique.

Log into registry

```
az acr login --name mycontainerregistry
```

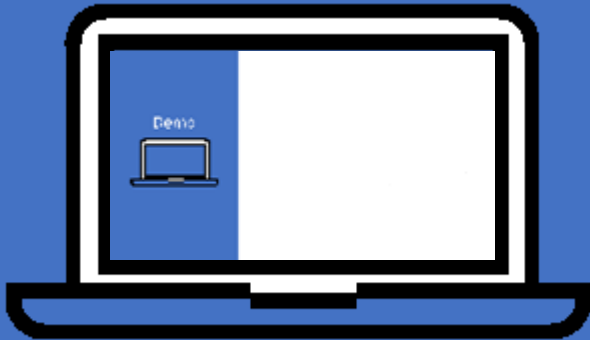
Tag an existing image

```
docker tag  
mycontainerregistry.azurecr.io/web-image  
web-image:v1
```

Push image to registry

```
docker push mycontainerregistry.azurecr.io/web-image
```


Demo



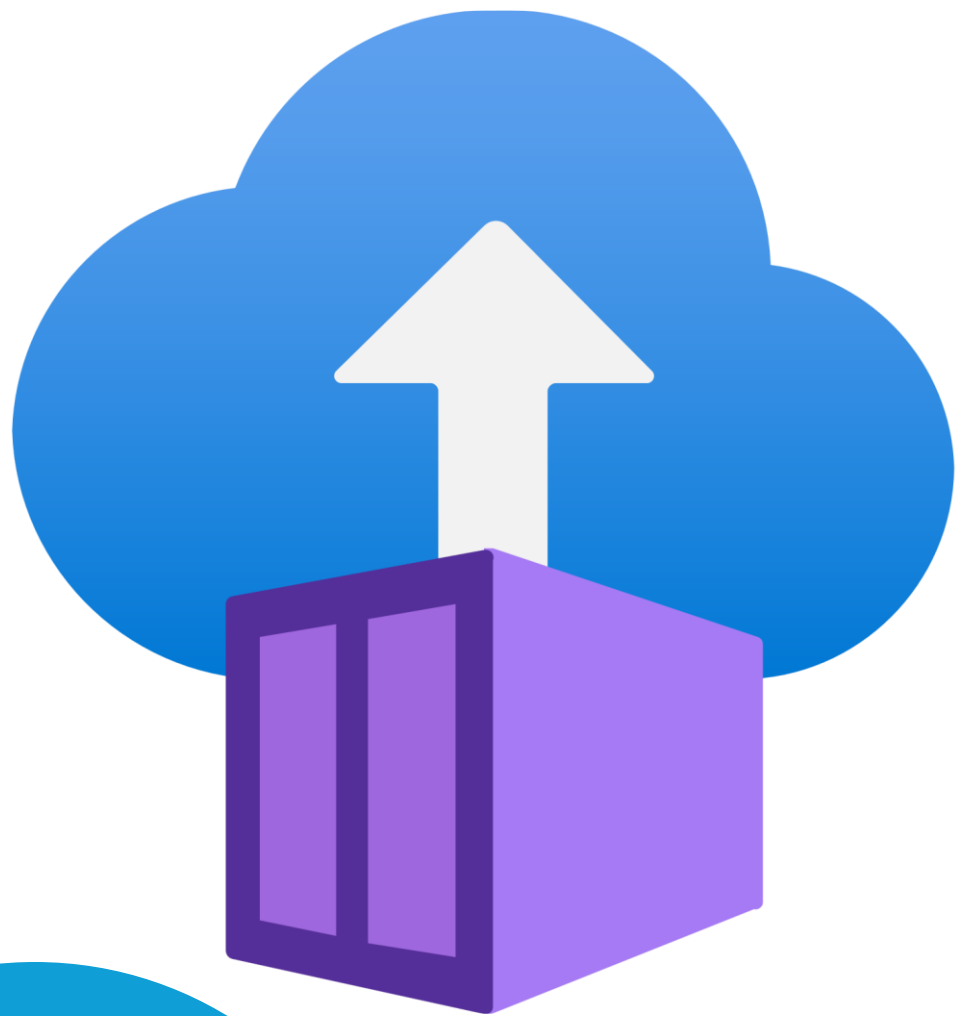
Azure Container Registry

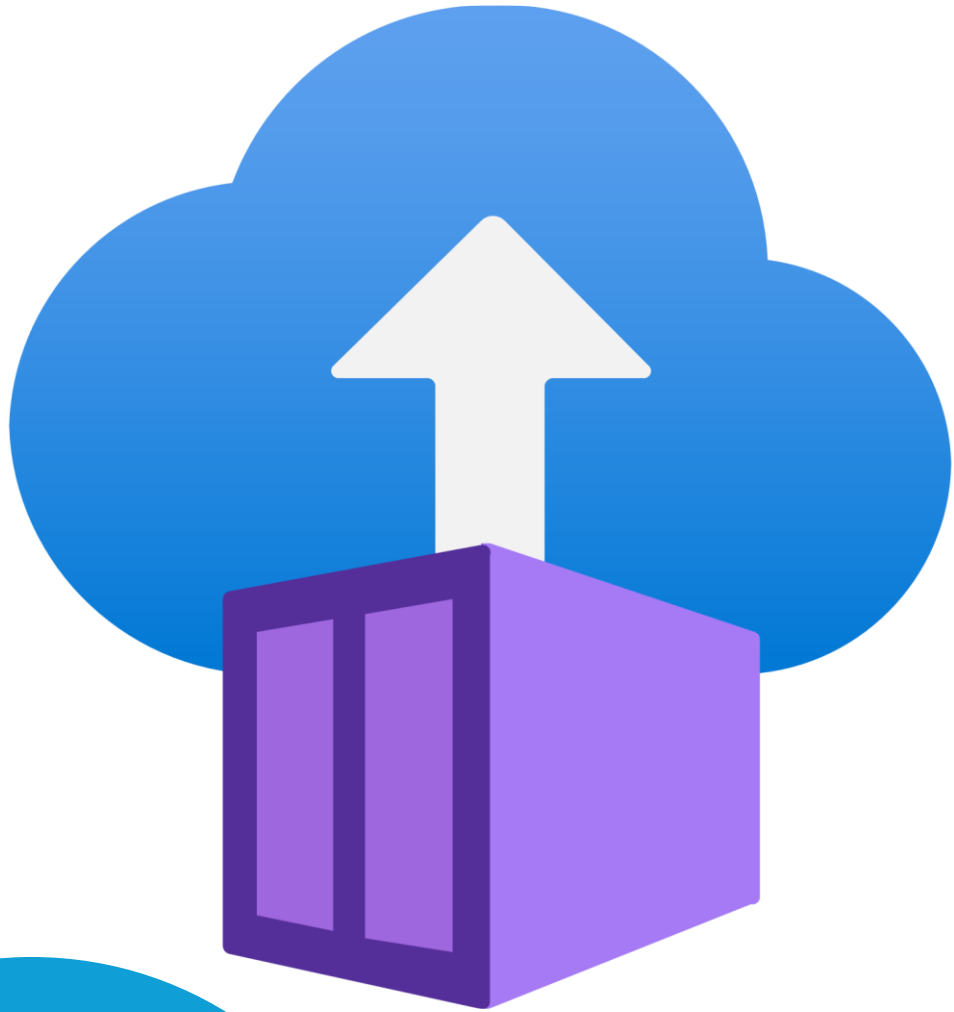
<https://learn.microsoft.com/en-us/azure/container-registry/container-registry-get-started-azure-cli>

Azure Container Instances (ACI)

Serverless container service that allows you to run containers on Azure without managing virtual machines or orchestrators

- No VM Management Required
- Fast Startup Times
- Per-Second Billing
- Supports Linux and Windows Containers





Azure Container Instances (ACI) Use Cases

- Batch and Parallel Jobs
Image rendering, video processing, or data transformation.
- Development and Testing
Disposable environments for testing application builds and scripts.
- Microservices Execution
Run lightweight, stateless services that don't require full orchestration.
- Event-Driven Jobs
Trigger containers in response to Azure Functions, Logic Apps, or webhooks.
- Temporary Compute Needs
Use serverless compute platform when long-term infrastructure isn't justified.

Azure Container Apps (ACA)

Run microservices and containerized applications without managing infrastructure.

- Autoscaling
- Integrated Ingress with HTTPS Support
- Dapr Integration
- Revision Management and Traffic Splitting
- Environment Isolation



Azure Container Apps (ACA) Use Cases

- Microservices Architecture
- API-Driven Applications
- Background Jobs and Event Processing
- Green/Blue Deployments and A/B Testing





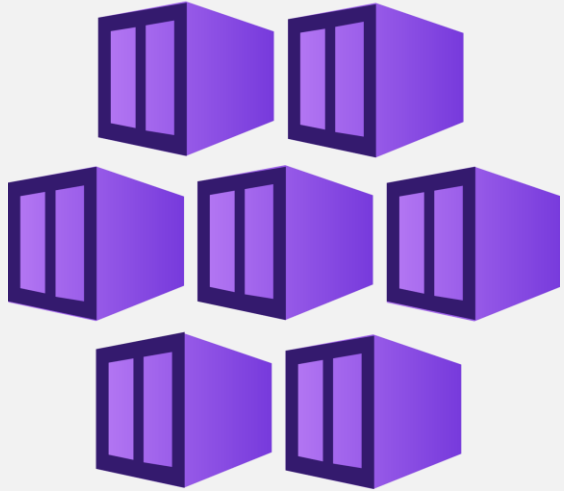
Kubernetes (K8S)

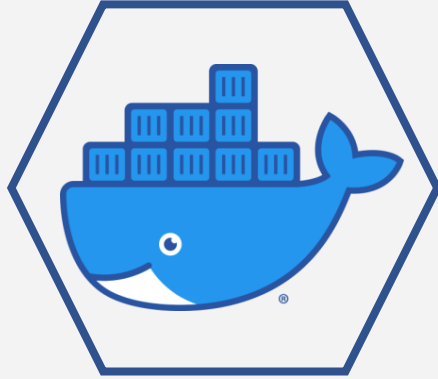
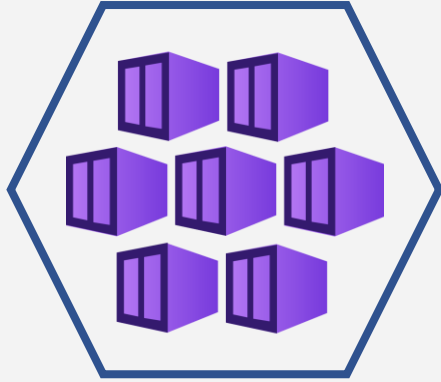
Container Based Application Management



- Containerized infrastructure
- Auto-scalable infrastructure
- Loosely coupled infrastructure
- Higher density of resource utilization
- Declarative configuration

Azure Container Solution Comparison

Azure Container Instance	Azure Container App	Azure Kubernetes Service
		
Short-live or burstable workloads	Simple, stateless workloads	Complex and scalable workloads
Low Complexity	Moderate Complexity	High Complexity
No Auto-Scaling	Auto-Scaling	Auto-Scaling Can scale higher



Summary

- Why use Containers
 - Isolate app and its dependencies
 - More efficient use of system resources
 - Enables faster software delivery cycles
- Why use ACR, ECR, or another Image Repository
 - Automate Development
 - Collaborate with Team
 - Secure Docker Images
- Know which container solution is best for a specific use case
 - ACI - POC or simple app
 - ACA – Scaling without infrastructure management
 - AKS – Scaling and need full control

Resources

- Container services documentation
<https://learn.microsoft.com/en-us/azure/containers/>
- Azure Container Registry documentation
<https://learn.microsoft.com/en-us/azure/container-registry/>
- Azure Container Instances documentation
<https://learn.microsoft.com/en-us/azure/container-instances/>
- Azure Container Apps documentation
<https://learn.microsoft.com/en-us/azure/container-apps/>
- Azure Kubernetes Service (AKS)
<https://learn.microsoft.com/en-us/azure/aks/>