

# Creating and Running Containers in Azure

---

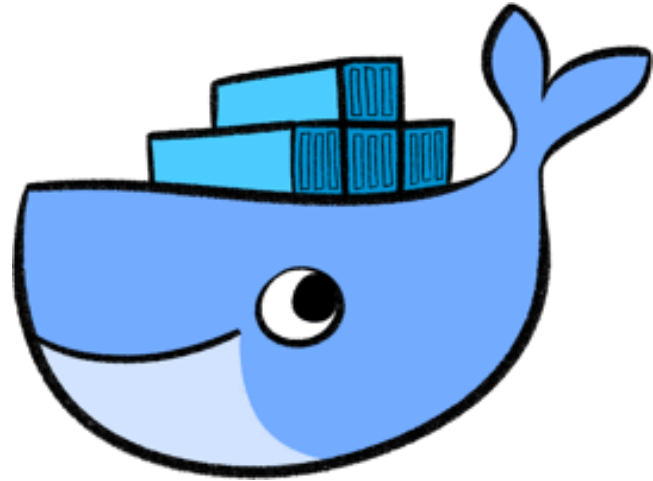


**Anthony E. Nocentino**

ENTERPRISE ARCHITECT @ CENTINO SYSTEMS

@nocentino [www.centinosystems.com](http://www.centinosystems.com)

# Running Containers in Azure



**docker**

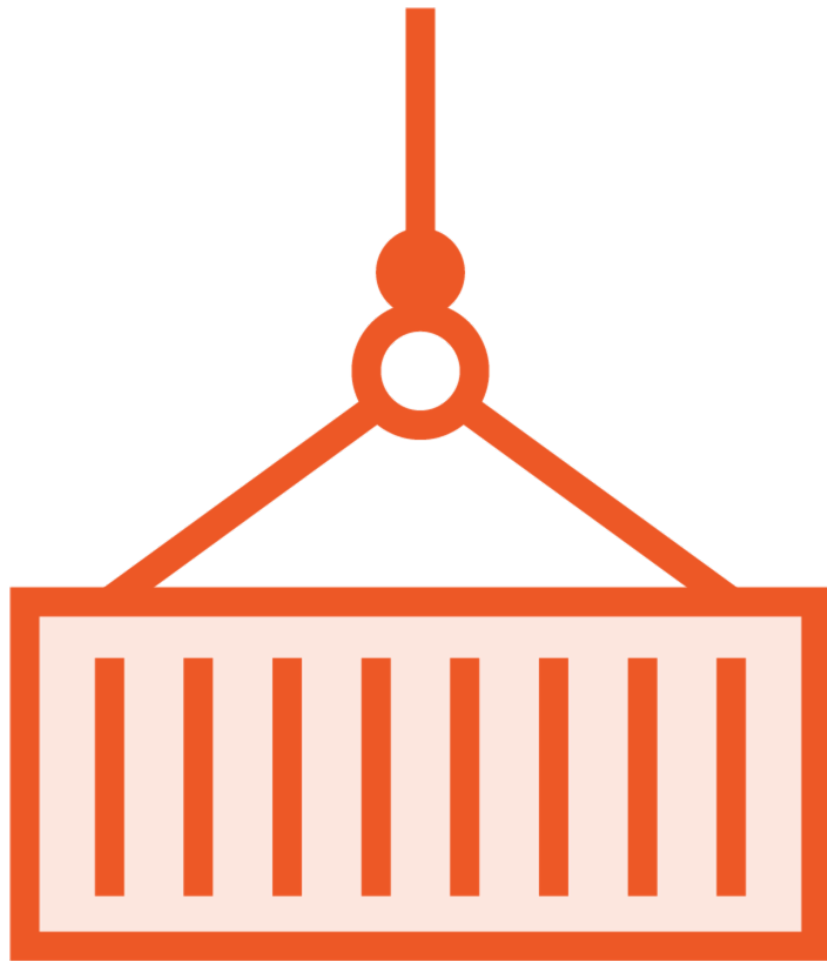


**Azure Container  
Registry (ACR)**



**Azure Container  
Instances (ACI)**

# Container Fundamentals



**Binaries, libraries and other components**

**Container image - binary application package**

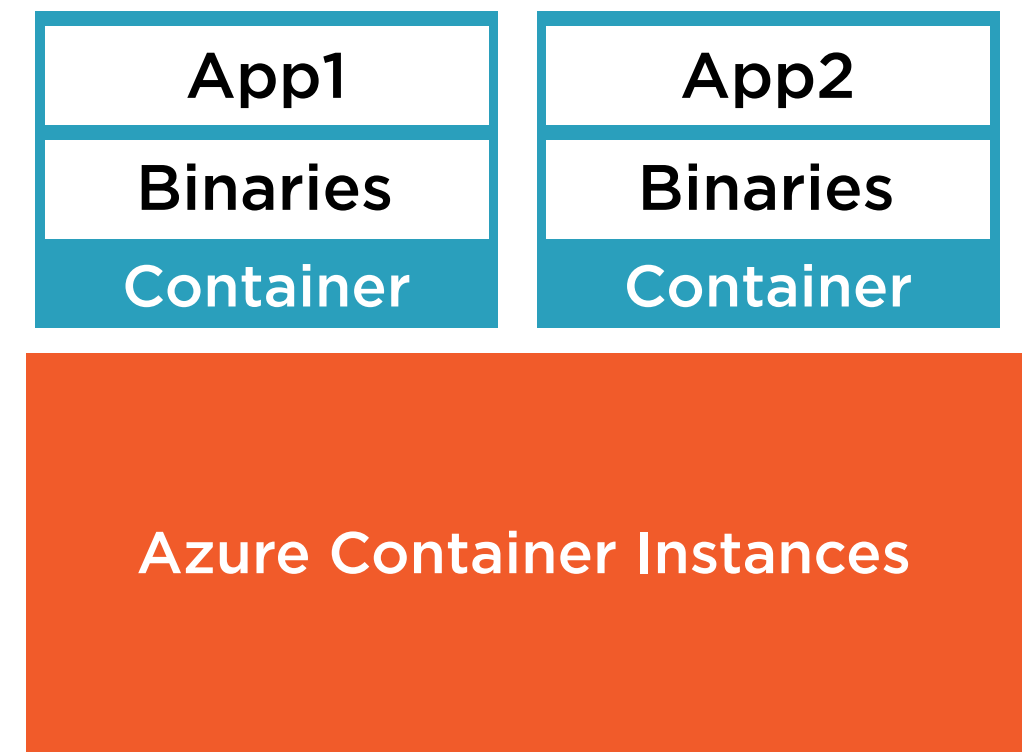
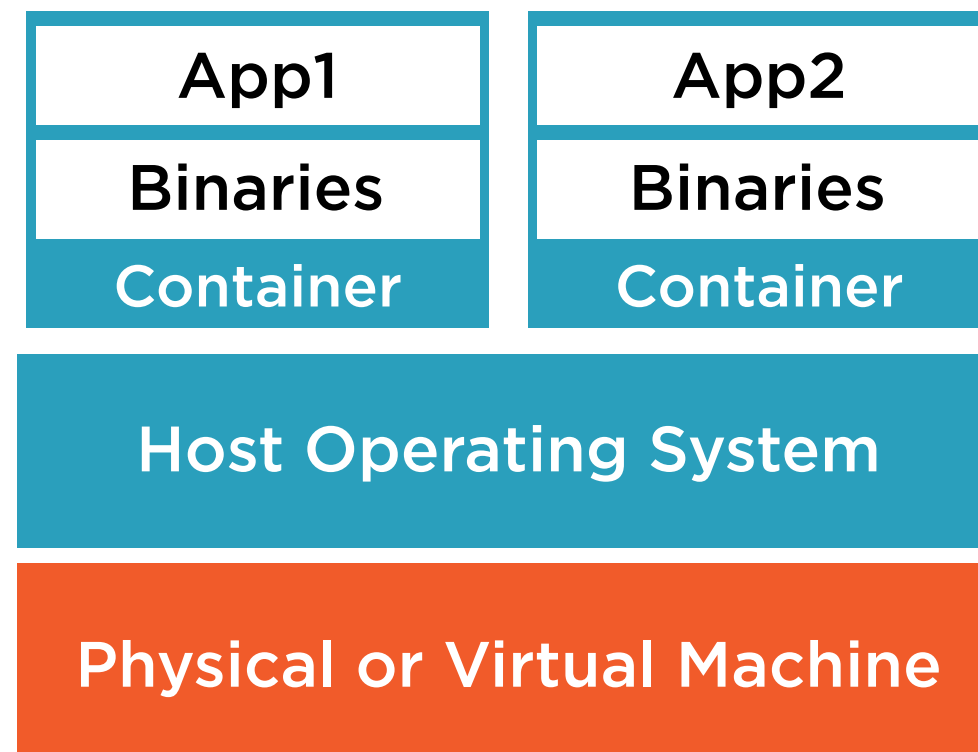
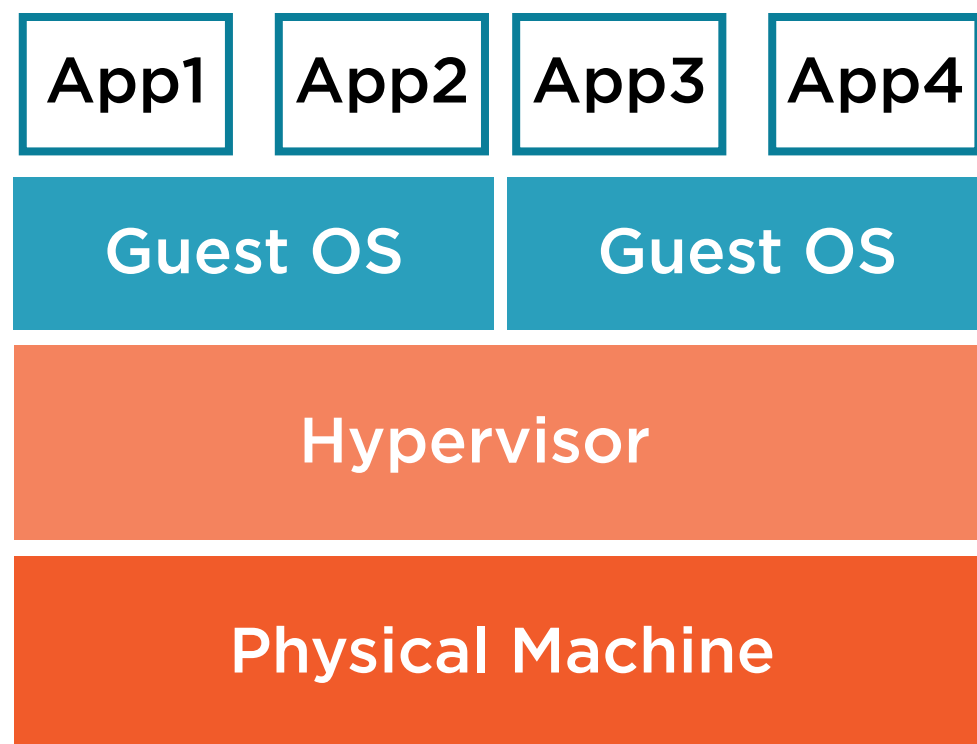
**Container - running container image**

**One app inside the container**

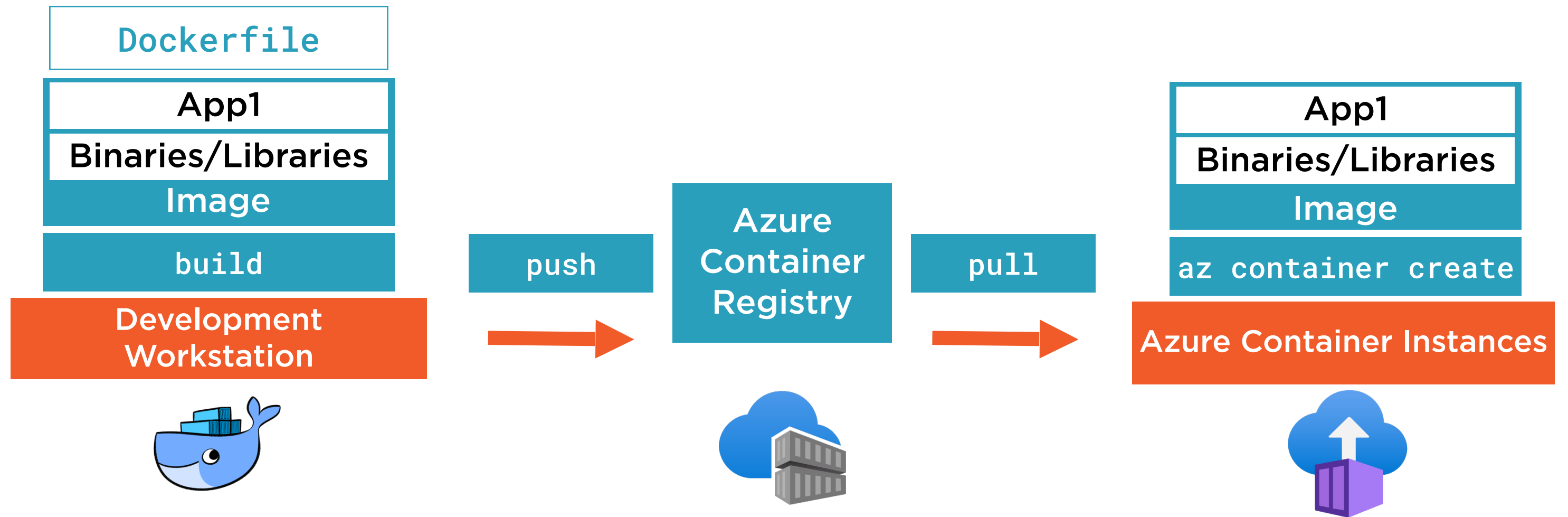
**Generally very small and very portable**

**Container Registries - enables exchanging of container images**

# Container Fundamentals



# Working with Containers in Azure



# Example Dockerfile

```
FROM mcr.microsoft.com/dotnet/core/aspnet:3.1
```

```
RUN mkdir /app
```

```
WORKDIR /app
```

```
COPY ./webapp/bin/Release/netcoreapp3.1/publish ./
```

```
COPY ./config.sh ./
```

```
RUN bash config.sh
```

```
EXPOSE 80
```

```
ENTRYPOINT ["dotnet", "webapp.dll"]
```

```
docker build -t webappimage:v1 .
```

# Demo

**Creating a container image using docker**

# Azure Container Registry (ACR)



**Build, store, and manage container images**  
**Key component of building a CI/CD pipeline**  
**ACR Tasks for container image automation**  
**Service tiers**

[\*\*https://docs.microsoft.com/en-us/azure/container-registry/container-registry-skus\*\*](https://docs.microsoft.com/en-us/azure/container-registry/container-registry-skus)



# ACR Authentication and Security Options



**Requires authentication for operations**

**Azure Active Directory Identities**

**Users**

**Service Principals**

**ACR Admin**

**Orchestrators, tools and applications should use 'headless' authentication**

**az acr login OR docker login**

**Role-based access controls**

# ACR Role-Based Authentication

Role/Permission	Access Resource Manager	Create/delete registry	Push image	Pull image	Delete image data	Change policies	Sign images
Owner	X	X	X	X	X	X	
Contributor	X	X	X	X	X	X	
Reader	X			X			
AcrPush			X	X			
AcrPull				X			
AcrDelete					X		
AcrImageSigner							X

From: <https://docs.microsoft.com/en-us/azure/container-registry/container-registry-roles>

# Creating and Authenticating to Azure Container Registry

```
ACR_NAME='psdemoacr' #<---- THIS NEEDS TO BE GLOBALLY unique in Azure
```

```
az acr create \  
  --resource-group psdemo-rg \  
  --name $ACR_NAME \  
  --sku Standard
```

```
az acr login --name $ACR_NAME
```

# Pushing an Image into ACR

```
ACR_NAME='psdemoacr'
```

```
ACR_LOGINSERVER=$(az acr show --name $ACR_NAME --query loginServer --output tsv)
```

```
#psdemoacr.azurecr.io
```

```
docker tag webappimage:v1 $ACR_LOGINSERVER/webappimage:v1
```

```
docker push $ACR_LOGINSERVER/webappimage:v1
```

```
#Build using ACR Tasks
```

```
az acr build --image "webappimage:v1-acr-task" --registry $ACR_NAME .
```

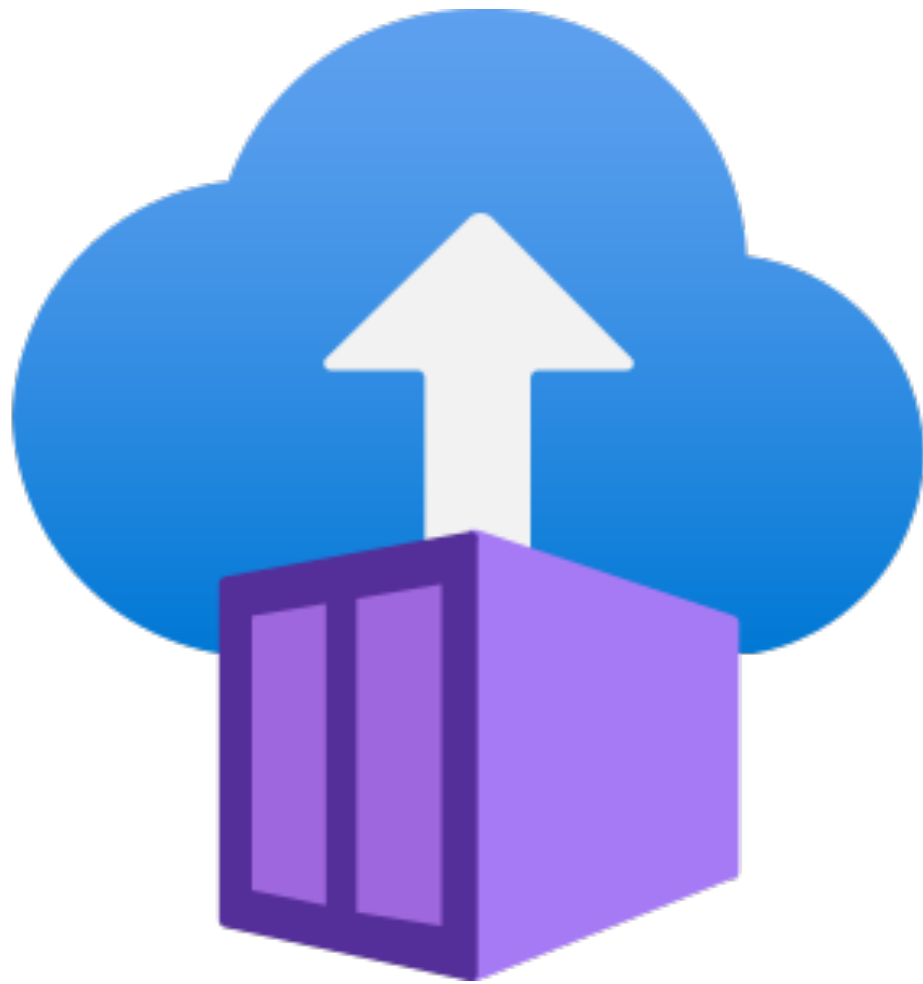
# Demo

**Creating an Azure Container Registry (ACR)**

**Pushing an image into ACR**

**Building an image in ACR using Tasks**

# Deploying Containers in Azure Container Instances



**Serverless container platform**

**Access application via Internet or on an Azure Virtual Network**

**Windows and Linux containers**

**Resource requests for CPU and memory**

**Use Azure Files for persistent storage**

**Deployed in Groups**

**Restart policy - always, on failure and never**

# Deploying Containers in ACI from Container Registries



**Azure Container Registry**



**Docker Hub or other container registries**



**Public or private**



**Login server**



**Username and password**

# Creating a Service Principal for ACI to Pull From ACR

```
ACR_NAME='psdemoacr'
```

```
ACR_REGISTRY_ID=$(az acr show --name $ACR_NAME --query id --output tsv)
```

```
SP_NAME=acr-service-principal
```

```
SP_PASSWD=$(az ad sp create-for-rbac \  
    --name http://$ACR_NAME-pull \  
    --scopes $ACR_REGISTRY_ID \  
    --role acrpull \  
    --query password \  
    --output tsv)
```

```
SP_APPID=$(az ad sp show \  
    --id http://$ACR_NAME-pull \  
    --query appId \  
    --output tsv)
```



# Running a Container from ACR in ACI

```
ACR_LOGINSERVER=$(az acr show --name $ACR_NAME --query loginServer --output tsv)
```

```
az container create \  
  --resource-group psdemo-rg \  
  --name psdemo-webapp-cli \  
  --dns-name-label psdemo-webapp-cli \  
  --ports 80 \  
  --image $ACR_LOGINSERVER/webappimage:v1 \  
  --registry-login-server $ACR_LOGINSERVER \  
  --registry-username $SP_APPID \  
  --registry-password $SP_PASSWD
```

psdemo-webapp-cli.centralus.azurecontainer.io

# Demo

## Deploying containers in Azure Container Instances (ACI)

- Azure Portal
- Azure CLI

# Summary

**Provision VMs**

**Configure VMs for remote access**

**Create ARM templates**

**Create container images for solutions by using Docker**

**Publish an image to the Azure Container Registry**

**Run containers by using Azure Container Instance**

Thank You!  
@nocentino