Azure Container Service

What is Azure Container service

Azure container service addresses these challenges by providing simplified configurations of proven open source container orchestration technology, optimized to run in the cloud. With just a few clicks you can deploy your container-based applications on a framework designed to help manage the complexity of containers deployed at scale, in production.

Step by step procedure to run ACS in Windows.

1. Install Azure CLI from [here](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?view=azure-cli-latest).
2. Login to Azure using command

Az login

1. Set subscription using command.

az account set -–subscription <subscription ID>

1. Create resource group using command

az group create --resource-group <group name> --location eastus

1. Create ACR (azure container registry) using command

az acr create --resource-group <group name> --name <registry name> --sku Standard --location eastus

1. Copy application docker file to c:\<application folder> (locally) Eg : Nats application (c:\nats)
2. Build and push image to ACR using.

az acr build --registry testcontainerregistrysupmek --image nats:v1 c:\nats --platform=windows

Note: default platform is Linux

1. Get ACR username and password using command

az acr update -n <acr name> --admin-enabled true

az acr credential show -n <acr name>

1. Create default container using command.

az container create -g <resource group name> --name natscontainer --image <acr name>.azurecr.io/nats:v2 --os-type Windows --registry-username <username copied from above step> --registry-password <password copied from above step> --ports 4222 8222 6222 --dns-name-label natscontainer-<acr name> --query "{FQDN:ipAddress.fqdn}" --output table

1. Custom Command run in container

az container create -g <resource group name> --name natscontainer --image <acr name>.azurecr.io/nats:v1 --os-type Windows --registry-username <username> --registry-password <password> --command-line "nats version" --dns-name-label natscontainer-<acr name> --query "{FQDN:ipAddress.fqdn}" --output table

1. Once container is created you will get fqdn id as output copy it and open it in url

For example :

http:// natscontainer-testcontainerregistrysupmek.eastus.azurecontainer.io:8222

1. Observe log files/ output for custom command using.

az container attach --resource-group <resource group name> --name natscontainer

1. Delete container using

az container delete --resource-group <resource group name> --name natscontainer.

1. List containers using command

az container list --resource-group <resource group name> --output table

1. Delete Resource Group.(Optional)

az group delete --name myResourceGroup

Running containers in container group windows (don’t support multiple containers for now)

1. Create Below example yaml file (azuredeploy.yaml)

For example: Take above nats application

apiVersion: 2018-10-01

location: eastus

name: myContainerGroup

properties:

containers:

- name: natscontainer

properties:

image: containerregistrysupmek.azurecr.io/nats:v1

resources:

requests:

cpu: 1

memoryInGb: 1.5

ports:

- port: 4222

- port: 8222

- port: 6222

osType: Windows

ipAddress:

type: Public

ports:

- protocol: tcp

port: '4222'

- protocol: tcp

port: '8222'

- protocol: tcp

port: '6222'

imageRegistryCredentials:

- server: <acr name>.azurecr.io

username: <acr user name>

password: <acr password>

tags: null

type: Microsoft.ContainerInstance/containerGroups

1. Create container using command

az container create --resource-group <resource group name> --file azuredeploy.yaml

1. Show the container using Command

az container show --resource-group <resource group name> --name myContainerGroup --output table.

1. View log files of running container.

az container logs --resource-group <resource group name> --name myContainerGroup --container-name natscontainer

Running containers in container group Linux (Supports Running multiple containers)

1. Create Below example yaml file (linux.yaml)

For example: Take nats and caddy application

apiVersion: 2018-10-01

location: eastus

name: mylinuxContainerGroup

properties:

containers:

- name: caddylinuxcontainer

properties:

image: caddy:latest

resources:

requests:

cpu: 1

memoryInGb: 1.5

ports:

- port: 80

- port: 8080

- name: natslinuxcontainer

properties:

image: nats:latest

resources:

requests:

cpu: 1

memoryInGb: 1.5

ports:

- port: 4222

- port: 6222

- port: 8222

osType: Linux

ipAddress:

type: Public

ports:

- protocol: tcp

port: '80'

- protocol: tcp

port: '8080'

- protocol: tcp

port: '4222'

- protocol: tcp

port: '8222'

- protocol: tcp

port: '6222'

tags: null

type: Microsoft.ContainerInstance/containerGroups

1. Create container using command

az container create --resource-group <resource group name> --file linux.yaml

1. Show the container using Command

az container show --resource-group <resource group name> --name mylinuxContainerGroup --output table.

1. View log files of running container.

az container logs --resource-group <resource group name> --name mylinuxContainerGroup --container-name natslinuxcontainer

az container logs --resource-group <resource group name> --name mylinuxContainerGroup --container-name caddylinuxcontainer