Azure DevOps Create CI-Build ACR and CD-Release ACI Pipeline

# Use CI (Continuous Integration) Building Pipeline to create the docker image and push to Azure Docker Registry (ACR)

* Create CI pipeline 🡪 Pipelines 🡪 create Pipeline 🡪 Azure Repos Git (Where is your code?) 🡪 AspNetCoreSample (Select a repository) 🡪 ASP.NET Core (.NET Framework) (Configure your pipeline) 🡪 Review your pipeline YAML (Review your pipeline YAML)
* If you just Run the pipeline there will be an error, need to create an agent first.
* Go to Project Settings 🡪 Pipelines 🡪 Agent pools 🡪 Default 🡪 New agent 🡪 Windows (x64) 🡪 follow the instructions on the popup

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* Get Personal Access Tokens

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* Config the agent

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

* Run the agent

A screenshot of a computer

Description automatically generated with medium confidence

## Create Build (CI) Pipeline

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

This will create the CI YAML file

Graphical user interface, text, application, email

Description automatically generated

## Create a Build Agent

* Create a new Pipeline 🡪 default pipeline for build

Graphical user interface, text, application, email

Description automatically generated

* Then delete the VS Test part task and add the Docker task

Graphical user interface, application

Description automatically generated

Graphical user interface, application, table

Description automatically generated with medium confidence

## Run the CI (Build) Pipeline

* Run Docker Windows Desktop first
* Then run CI (Build) pipeline

Graphical user interface, text

Description automatically generated

* Check Build Agent activity

Text

Description automatically generated

* Check Azure Container Registry

Graphical user interface, application, Word

Description automatically generated

Working!!

# Use CD (Continuous Delivery) Release Pipeline to deploy docker image from ACR to ACI (Azure Container Instance)

* Create ARM (Azure Resource Manager) for ACR (Azure Container Registry)

Graphical user interface, application, Word

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

* Create CD (Release) Pipeline
* Create Artifact

Graphical user interface, text

Description automatically generated

A picture containing diagram

Description automatically generated

The Source Type of “Service connection” drop down is Resource Manager and that is the reason why we need to create it!!

* Create Task

Graphical user interface, text

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* Here is the settings of Azure CLI Task:

Graphical user interface, text, application, email

Description automatically generated

Here is the Inline Script:

|  |
| --- |
| az container create  --name $(aciname)  --resource-group $(resourcegroup)  --image $(image):$(acrloginserver)/$(Build.BuildId):$(Build.BuildId)  --registry-login-server $(acrloginserver)  --registry-username $(acrusername)  --registry-password $(acrpassword)  --dns-name-label: $(dnslabel)$(Build.BuildId) |

* Variables

Graphical user interface, text, application, email

Description automatically generated

Here are the variables:

|  |
| --- |
| $aciname = “acidemo”  $acrloginserver = “johnarc.azurecr.io”  $acrpassword = “nq+yHVWotRbsYwUpi0GgyZJ9ra3tbmHc”  $acrusername = “johnarc”  $dnslabel = “wwwacidemo”  $resourcegroup = “AzureContainerRegistries” |

* Deploy

Graphical user interface, application

Description automatically generated

* Check Azure Container Instance

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* Browser to FQDN:

[Home page - WebAppDockerTest (wwwacidemo.eastus.azurecontainer.io)](http://wwwacidemo.eastus.azurecontainer.io/)

Graphical user interface, text, application, email

Description automatically generated

Working!!

* And this CICD build for Azure Container can be deploy multiple times, it just upgraded the previous Container Registry and Container Instance.