# Hardcoded values from your pipeline variables

$batchAccount = "lvbatchdev"

$resourceGroup = "lv-batch-dev"

$storageAccount = "lvbatchdev"

$appName = "ccdc-batch-jobs"

$appVersion = "10978"

$container = "ccdbatchjobs"

$region = "eastus2"

# Set the batch account for all future commands

az batch account set `

--name $batchAccount `

--resource-group $resourceGroup

if ($LASTEXITCODE -ne 0) {

Write-Error "Failed to set batch account"

exit 1

}

# Create Job JSON

$jobId = "$appName-$(Get-Random)"

$jobJson = @"

{

"id": "$jobId",

"poolInfo": {

"autoPoolSpecification": {

"autoPoolIdPrefix": "autopool",

"poolLifetimeOption": "job",

"keepAlive": false,

"pool": {

"vmSize": "Standard\_A1\_v2",

"targetDedicatedNodes": 1,

"virtualMachineConfiguration": {

"imageReference": {

"publisher": "MicrosoftWindowsServer",

"offer": "WindowsServer",

"sku": "2022-datacenter-smalldisk",

"version": "latest"

},

"nodeAgentSKUId": "batch.node.windows amd64"

},

"applicationPackageReferences": [

{

"applicationId": "$appName",

"version": "$appVersion"

}

]

}

}

}

}

"@

$jobJson | Out-File -FilePath "job.json" -Encoding utf8

# Create the job

az batch job create --json-file "job.json"

if ($LASTEXITCODE -ne 0) {

Write-Error "Failed to create job"

Remove-Item "job.json" -Force -ErrorAction SilentlyContinue

exit 1

}

# Wait for the auto pool to spin up

Start-Sleep -Seconds 30

# Create Task JSON

$taskId = "task-$(Get-Random)"

$taskJson = @"

{

"id": "$taskId",

"commandLine": "cmd /c echo Hello from Batch > output.txt",

"userIdentity": {

"autoUser": {

"scope": "pool",

"elevationLevel": "admin"

}

},

"applicationPackageReferences": [

{

"applicationId": "$appName",

"version": "$appVersion"

}

],

"constraints": {

"maxWallClockTime": "PT15M",

"maxTaskRetryCount": 1

}

}

"@

$taskJson | Out-File -FilePath "task.json" -Encoding utf8

# Add the task to the job

az batch task create --job-id $jobId --json-file "task.json"

if ($LASTEXITCODE -ne 0) {

Write-Error "Failed to create task"

Remove-Item "job.json","task.json" -Force -ErrorAction SilentlyContinue

exit 1

}

Write-Host "Job and Task created successfully!"

Remove-Item "job.json","task.json" -Force -ErrorAction SilentlyContinue