# Variables

$batchAccount = "lvbatchdev"

$resourceGroup = "lv-batch-dev"

$appId = "ccdc-batch-jobs"

$appName = "ccdc-batch-jobs"

$activeVersion = "10978"

az batch account login --name $batchAccount --resource-group $resourceGroup

$jobId = "$appId-$(Build.BuildId)-$(Release.ReleaseId)"

$jobJson = @"

{

"id": "$jobId",

"poolInfo": {

"autoPoolSpecification": {

"autoPoolIdPrefix": "autopool",

"poolLifetimeOption": "job",

"keepAlive": false,

"pool": {

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

"targetDedicatedNodes": 1,

"virtualMachineConfiguration": {

"imageReference": {

"publisher": "canonical",

"offer": "0001-com-ubuntu-server-focal",

"sku": "20\_04-lts",

"version": "latest"

},

"nodeAgentSKUId": "batch.node.ubuntu 20.04"

},

"applicationPackageReferences": [

{"applicationId": "$appName", "version": "$activeVersion"}

],

"startTask": {

"commandLine": "/bin/bash -c 'echo Pool starting && apt-get update && apt-get install -y unzip && echo Pool ready'",

"userIdentity": {"autoUser": {"scope": "pool", "elevationLevel": "admin"}},

"waitForSuccess": true

}

}

}

},

"onAllTasksComplete": "terminatejob"

}

"@

$jobJson | Out-File "job.json" -Encoding UTF8

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

Remove-Item "job.json" -Force

Start-Sleep -Seconds 30

$taskId = "task-$(Build.BuildId)"

$taskJson = @"

{

"id": "$taskId",

"commandLine": "/bin/bash -c 'echo Task started && ls -la \$AZ\_BATCH\_APP\_PACKAGE\_${appName}\_$activeVersion && echo Task completed'",

"applicationPackageReferences": [

{"applicationId": "$appName", "version": "$activeVersion"}

],

"userIdentity": {"autoUser": {"scope": "pool", "elevationLevel": "nonadmin"}},

"constraints": {"maxWallClockTime": "PT15M", "maxTaskRetryCount": 1}

}

"@

$taskJson | Out-File "task.json" -Encoding UTF8

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

Remove-Item "task.json" -Force

$timeout=300;$elapsed=0;$checkInterval=10

while ($elapsed -lt $timeout) {

Start-Sleep -Seconds $checkInterval

$elapsed += $checkInterval

$task = az batch task show --job-id $jobId --task-id $taskId -o json | ConvertFrom-Json

if ($task.state -eq "completed") {

$code=$task.executionInfo.exitCode

Write-Host "Task exit code: $code"

az batch task file download --job-id $jobId --task-id $taskId --file-path "stdout.txt" --destination "-" 2>$null

az batch task file download --job-id $jobId --task-id $taskId --file-path "stderr.txt" --destination "-" 2>$null

exit $code

}

}

Write-Host "Task did not complete in $timeout seconds"

exit 1