Write-Host ">>> Step 0 : ensure pool exists"

$endpoint = "https://$(BatchAccount).$(Region).batch.azure.com"

$poolId = "$(PoolId)"

# Set ErrorActionPreference to Continue so the script doesn't stop on az command failures

$ErrorActionPreference = "Continue"

# Try to show the pool — exit code 0 if found, 1 if not

az batch pool show `

--pool-id $poolId `

--account-name $(BatchAccount) `

--account-endpoint $endpoint 2>&1 | Out-Null

if ($LASTEXITCODE -ne 0) {

Write-Host "Pool $poolId not found — creating"

$image = "$(ImagePublisher):$(ImageOffer):$(ImageSku)"

az batch pool create `

--id $poolId `

--vm-size $(VmSize) `

--target-dedicated-nodes 1 `

--image $image `

--account-name $(BatchAccount) `

--account-endpoint $endpoint `

--start-task-command-line "/bin/echo pool ready"

if ($LASTEXITCODE -ne 0) {

throw "Pool creation failed."

}

Write-Host "Pool $poolId created successfully"

}

else {

Write-Host "Pool $poolId already exists"

}

2nd method

Write‑Host ">>> Step 0 : ensure pool exists"

$endpoint = "https://$(BatchAccount).$(Region).batch.azure.com"

$poolId = "$(PoolId)"

$batchAcc = "$(BatchAccount)"

$vmSize = "$(VmSize)"

$image = "$(ImagePublisher):$(ImageOffer):$(ImageSku)"

# Try to show pool

az batch pool show --pool-id $poolId --account-name $batchAcc --account-endpoint $endpoint | Out-Null

if ($LASTEXITCODE -eq 0) {

Write‑Host "Pool $poolId already exists"

return

}

Write‑Host "Pool not found — creating"

az batch pool create `

--id $poolId `

--vm-size $vmSize `

--target-dedicated-nodes 1 `

--image $image `

--account-name $batchAcc `

--account-endpoint $endpoint `

--start-task-command-line "/bin/echo pool ready"

if ($LASTEXITCODE -ne 0) {

throw "Pool creation failed."

}