# Step 3. Create Auto pool with correct authentication

- task: AzureCLI@2

displayName: 'Create Azure Batch Job with Auto Pool'

inputs:

azureSubscription: 'CCDC-DEV-UAT-SPN'

scriptType: 'ps'

scriptLocation: 'inlineScript'

powerShellErrorActionPreference: 'continue' # Important: don't stop on non-terminating errors

addSpnToEnvironment: true

inlineScript: |

# IMPORTANT: The issue is that az batch account login doesn't properly set the auth context

# We need to use the service principal credentials directly

Write-Host "Setting up Batch authentication context..."

# Get the service principal details from the environment

$tenantId = $env:tenantId

$servicePrincipalId = $env:servicePrincipalId

$servicePrincipalKey = $env:servicePrincipalKey

# Login to Azure with the service principal

az login --service-principal -u $servicePrincipalId -p $servicePrincipalKey --tenant $tenantId

az account set --subscription "$(az account show --query id -o tsv)"

# Now login to Batch account

az batch account login --name $(BatchAccount) --resource-group $(batchResourceGroup)

# Create job ID

$jobId = "$(appId)-$(Build.BuildId)-" + (Get-Random -Maximum 9999)

Write-Host "Creating job with ID: $jobId"

# Create the JSON content directly as a string

$jobJson = @"

{

"id": "$jobId",

"poolInfo": {

"autoPoolSpecification": {

"autoPoolIdPrefix": "autopool",

"poolLifetimeOption": "job",

"keepAlive": false,

"pool": {

"vmSize": "$(vmSize)",

"virtualMachineConfiguration": {

"imageReference": {

"publisher": "canonical",

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

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

"version": "latest"

},

"nodeAgentSkuId": "batch.node.ubuntu 20.04"

},

"targetDedicatedNodes": 1

}

}

}

}

"@

# Write to file

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

# Create the job

Write-Host "Creating batch job..."

$output = az batch job create --json-file job.json 2>&1

$exitCode = $LASTEXITCODE

# Clean up

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

if ($exitCode -eq 0) {

Write-Host "Job $jobId created successfully!"

} else {

Write-Host "Failed to create job. Output: $output"

exit 1

}

2nd method

# Step 3. Create Auto pool using inline PowerShell (like Classic)

- task: PowerShell@2

displayName: 'Create Azure Batch Job with Auto Pool'

inputs:

targetType: 'inline'

script: |

# Install Azure CLI if not present

if (-not (Get-Command az -ErrorAction SilentlyContinue)) {

Write-Host "Installing Azure CLI..."

Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi

Start-Process msiexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'

$env:Path = [System.Environment]::GetEnvironmentVariable("Path","Machine") + ";" + [System.Environment]::GetEnvironmentVariable("Path","User")

}

# Get the service connection details

$serviceConnection = Get-VstsInput -Name ConnectedServiceName

$endpoint = Get-VstsEndpoint -Name $serviceConnection -Require

# Login using service principal

$env:AZURE\_CLIENT\_ID = $endpoint.Auth.Parameters.ServicePrincipalId

$env:AZURE\_CLIENT\_SECRET = $endpoint.Auth.Parameters.ServicePrincipalKey

$env:AZURE\_TENANT\_ID = $endpoint.Auth.Parameters.TenantId

az login --service-principal -u $env:AZURE\_CLIENT\_ID -p $env:AZURE\_CLIENT\_SECRET --tenant $env:AZURE\_TENANT\_ID

# Login to batch account

az batch account login --name $(BatchAccount) --resource-group $(batchResourceGroup)

# Create job

$jobId = "$(appId)-$(Build.BuildId)-" + (Get-Random)

$job = @{

id = $jobId

poolInfo = @{

autoPoolSpecification = @{

autoPoolIdPrefix = "autopool"

poolLifetimeOption = "job"

keepAlive = $false

pool = @{

vmSize = "$(vmSize)"

virtualMachineConfiguration = @{

imageReference = @{

publisher = "canonical"

offer = "0001-com-ubuntu-server-focal"

sku = "20\_04-lts"

version = "latest"

}

nodeAgentSkuId = "batch.node.ubuntu 20.04"

}

targetDedicatedNodes = 1

}

}

}

}

$job | ConvertTo-Json -Depth 10 | Out-File job.json

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

Remove-Item job.json

pwsh: true