Looking at your Azure DevOps pipeline and the error output, I can see the issue. The error occurs when the pipeline tries to log into Azure using a service principal. Specifically:

\*\*The Error:\*\*

- Line 55-56 show two "command not found" errors when trying to execute release scripts

- Line 58 shows the actual error: "az[error]Script failed with exit code: 127"

- Exit code 127 typically means "command not found"

\*\*Root Cause:\*\*

The Azure CLI commands are failing because either:

1. The Azure CLI is not installed in the build agent

2. The inline script has syntax issues with the authentication commands

\*\*How to Fix:\*\*

1. \*\*Ensure Azure CLI is available\*\* - Add this task before your AzureCLI@2 task:

```yaml

- task: UsePythonVersion@0

inputs:

versionSpec: '3.x'

addToPath: true

- script: |

curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

displayName: 'Install Azure CLI'

```

2. \*\*Fix the inline script\*\* - Your current script seems to have issues. Here's a corrected version:

```yaml

- task: AzureCLI@2

displayName: 'Create Azure Batch Job with Auto Pool'

inputs:

azureSubscription: 'CCDC-DEV-UAT-SPN'

scriptType: 'bash'

scriptLocation: 'inlineScript'

inlineScript: |

set -e

echo "Logging into Azure Batch account..."

# Set batch account context

az batch account login \

--name $(BatchAccount) \

--resource-group $(BatchResourceGroup)

# Create job with auto pool

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

cat > job.json <<EOF

{

"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

}

}

}

}

EOF

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

rm -f job.json

echo "Job $jobId created successfully with auto-pool"

```

3. \*\*Alternative approach\*\* - If you're still having issues, you can use the Azure Batch PowerShell commands instead:

```yaml

- task: AzurePowerShell@5

displayName: 'Create Azure Batch Job with Auto Pool'

inputs:

azureSubscription: 'CCDC-DEV-UAT-SPN'

ScriptType: 'InlineScript'

Inline: |

# Login to Batch account

$context = Get-AzBatchAccountKey -AccountName $(BatchAccount) -ResourceGroupName $(BatchResourceGroup)

# Create job with auto pool

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

# Define pool specification

$poolSpec = New-Object Microsoft.Azure.Commands.Batch.Models.PSAutoPoolSpecification

$poolSpec.AutoPoolIdPrefix = "autopool"

$poolSpec.PoolLifetimeOption = "Job"

$poolSpec.KeepAlive = $false

# Create and submit job

New-AzBatchJob -Id $jobId -PoolInformation $poolInfo -BatchContext $context

azurePowerShellVersion: 'LatestVersion'

```

The main issue appears to be that the Azure CLI commands aren't being executed properly in your build environment. Make sure the Azure CLI is installed and the service connection has the proper permissions to access your Batch account.​​​​​​​​​​​​​​​​