AUTO-SYS-OPS

> Level up

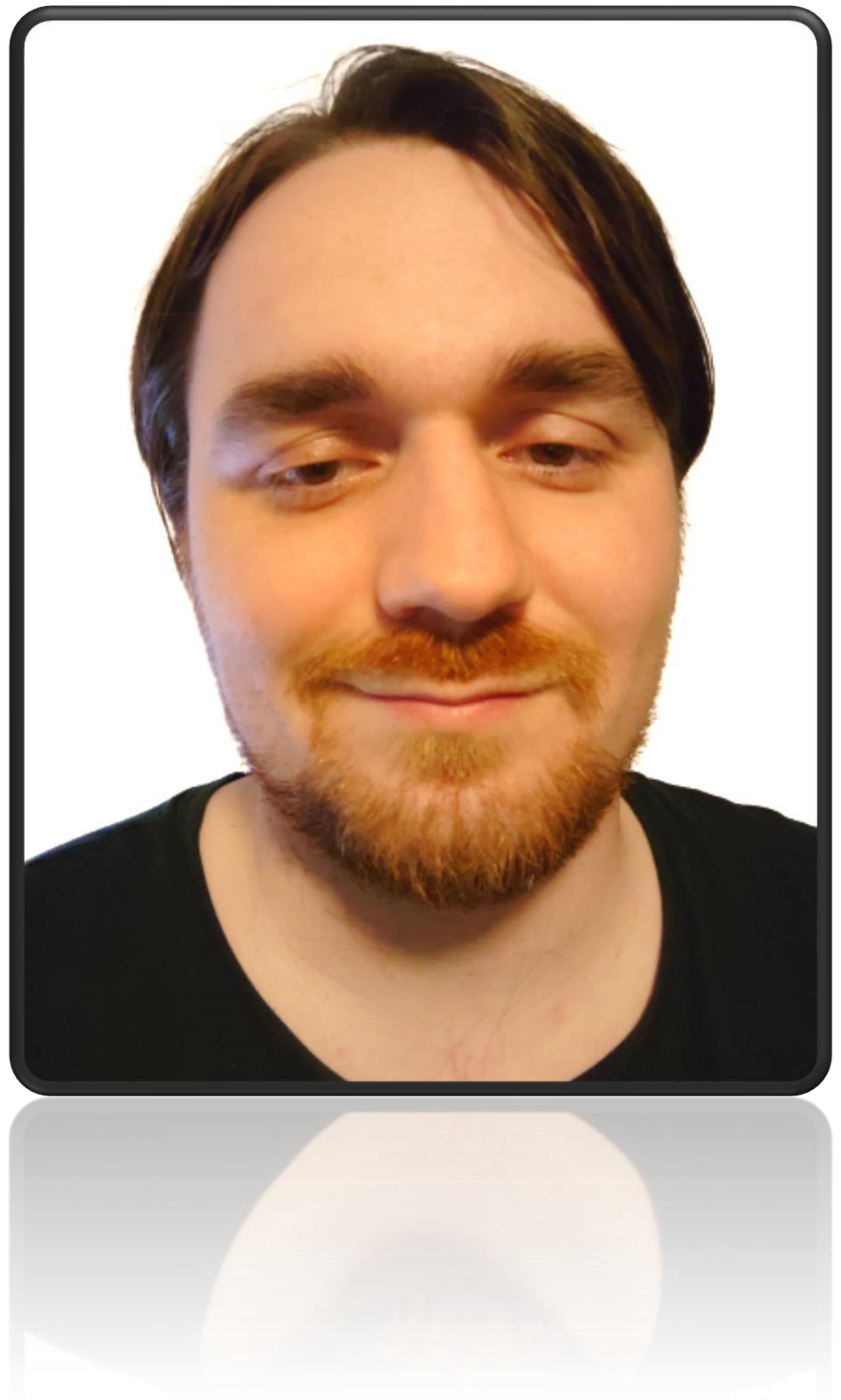
/ Your Azure DevOps pipelines

Introduction

Who is Leo Visser?

- Cloud Consultant @ OGD-ict diensten
  - IAC, CI/CD, Automation, Agile, DevOps

- Previous experiences:
  - Technical Lead SysOps Team
  - SysOps Engineer
  - Functional Application Specialist
  - PHP Backend Developer



AUTO-SYS-OPS

Twitter: @AutoSysOps
Blog: www.autosysops.com

Agenda

- What are Azure DevOps pipelines
- Templates
- Automatic retry
- Approvals
- Parallelize jobs
- Automatic rollback

Azure DevOps pipelines

/ Introduction

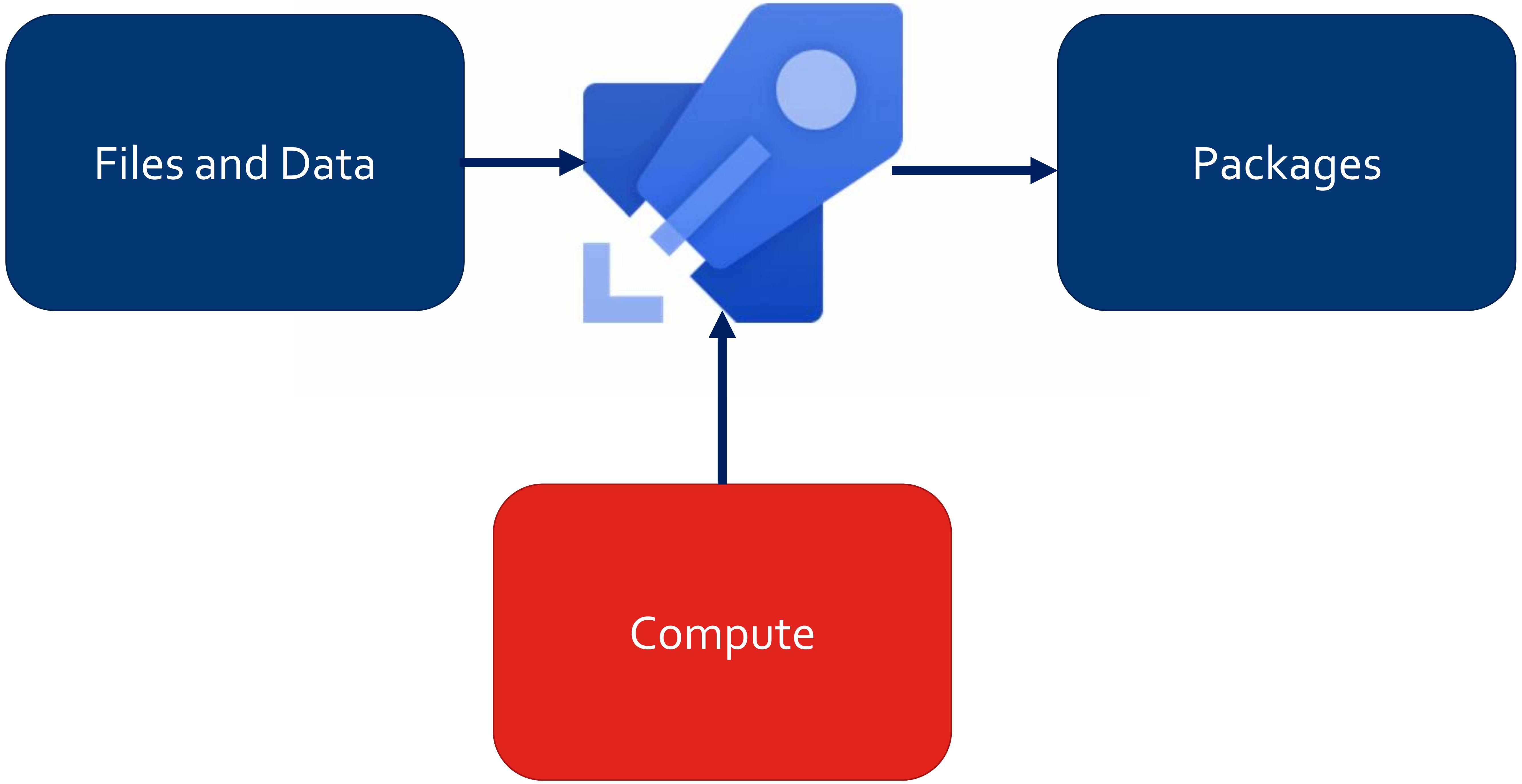
Azure DevOps pipelines

What is a pipeline?

- Utilize Compute to
    - Package applications
    - Deploy resources
    - Perform tasks

- Queued execution
    - Not time sensitive
    - Run on agents

- Build vs Release pipeline
    - Use Build

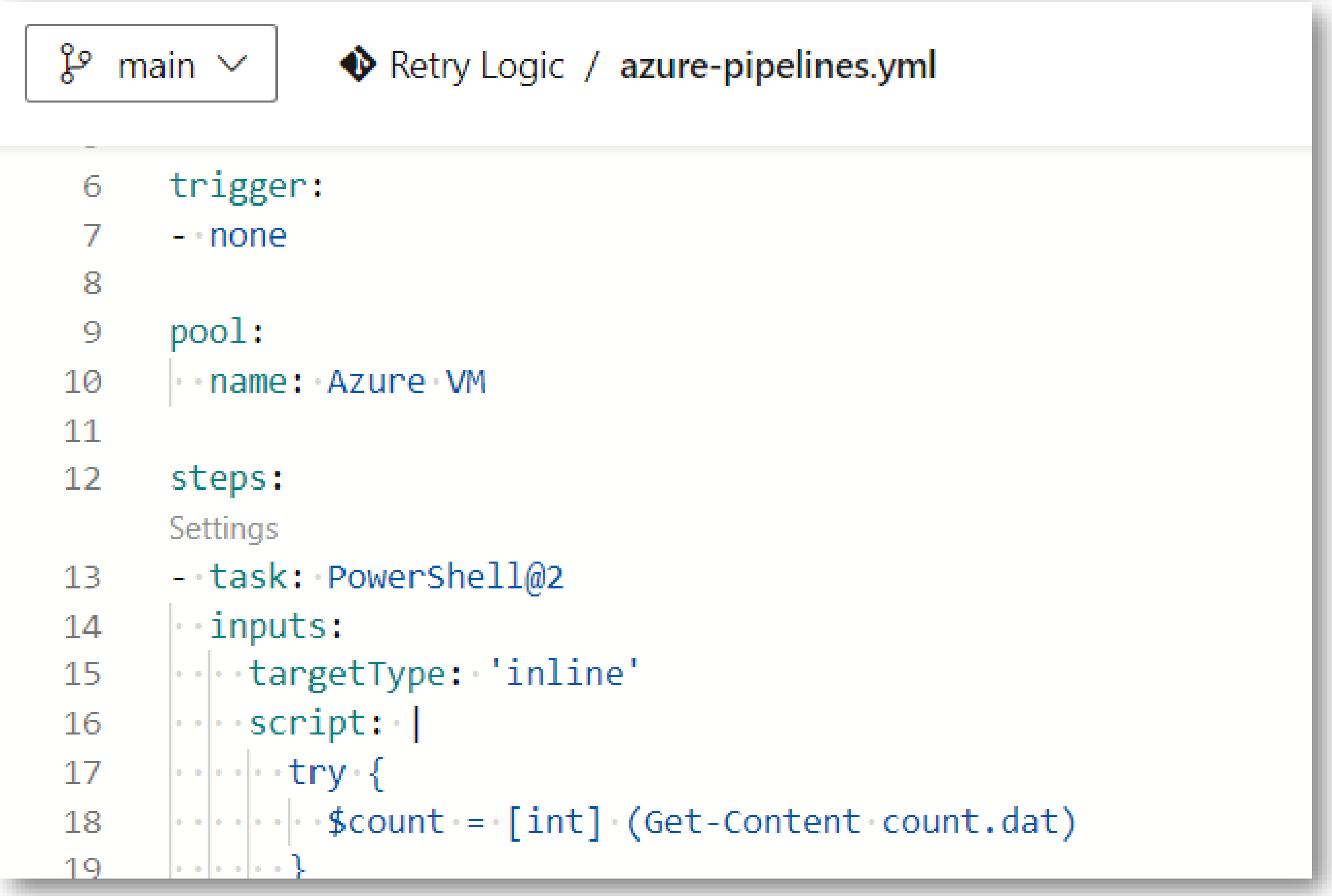


Azure DevOps pipelines

What is a pipeline?

- Written in YAML
  - Azure DevOps specific syntax requirements
  - Stored in source control
  - Use tasks available in a marketplace

- Set up in Azure DevOps
  - Use YAML file
  - Execute and check logs
  - Compiled



```yaml
trigger:
- none

pool:
  name: Azure VM

steps:
- task: PowerShell@2
  inputs:
    targetType: 'inline'
    script: |
      try {
        $count = [int] (Get-Content count.dat)
      }
```

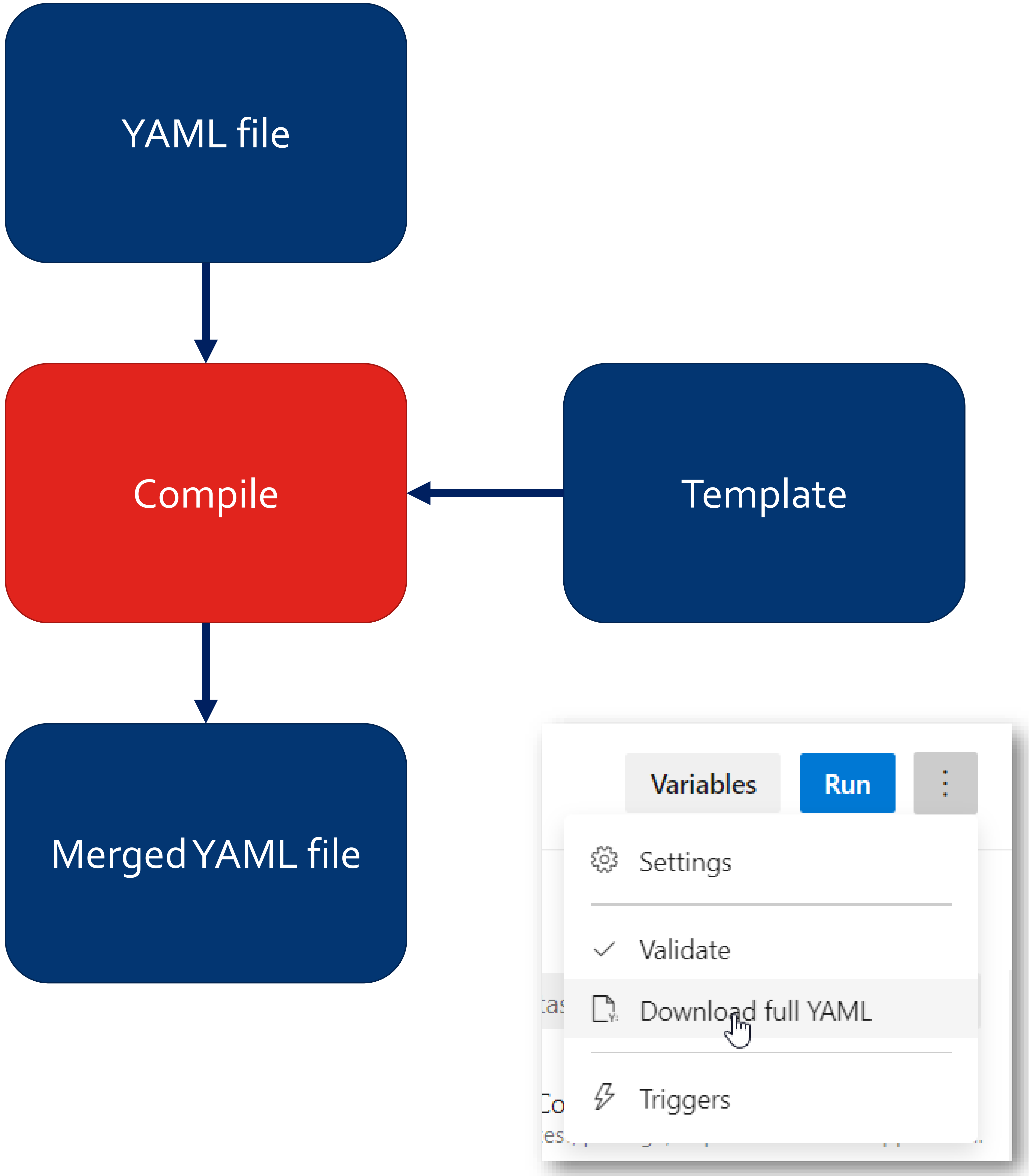
> Using templates

/ In Azure DevOps pipelines

Templates

How to use templates

- YAML files containing parts of a pipeline
- Reusable in other pipelines
- Use Download full YAML option to troubleshoot
- Contains parameters but can still use variables defined in main YAML
- Allows to use expressions like “each”



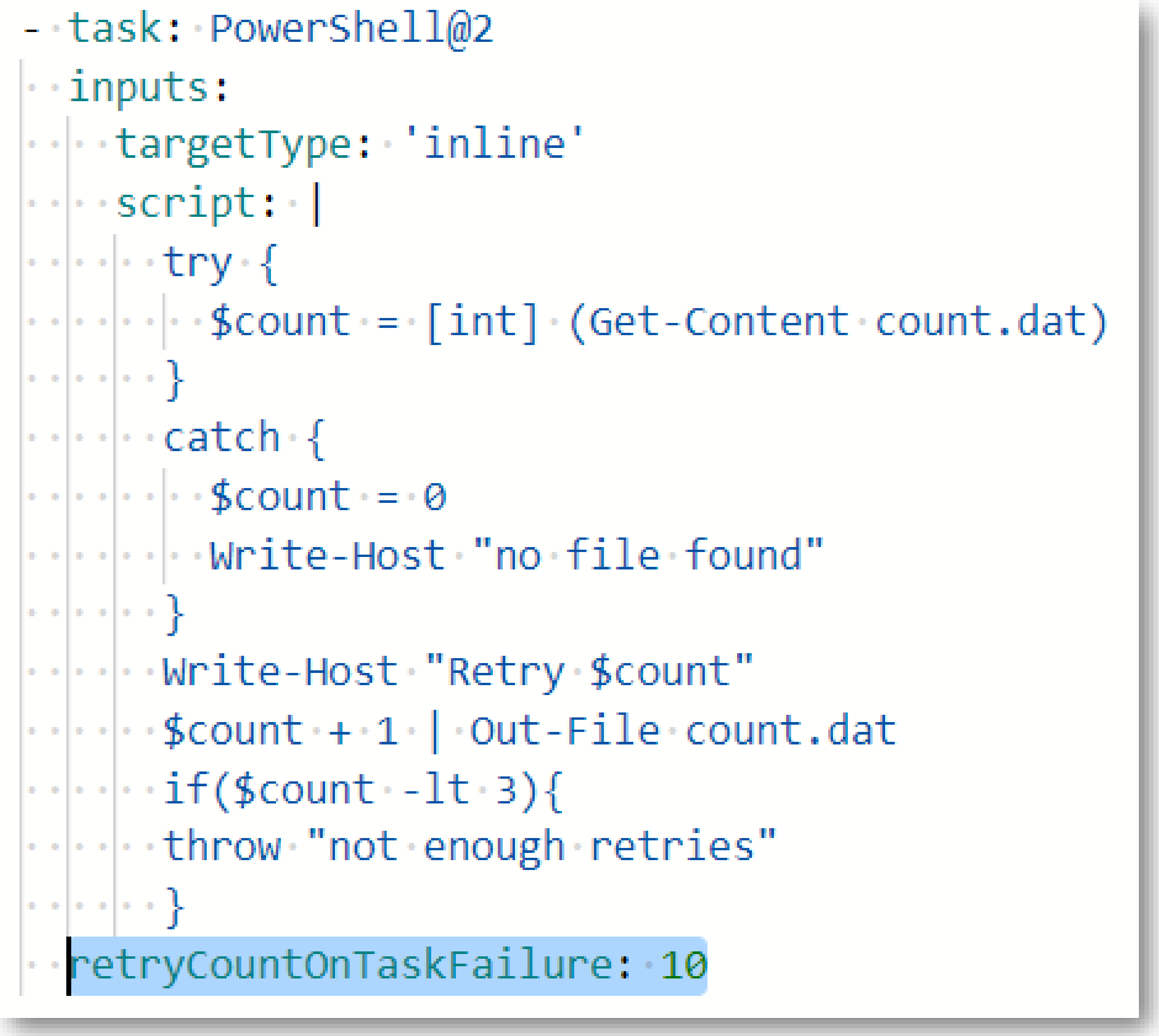
Automatic Retry

/ In Azure DevOps pipelines

Retry

How to implement automatic retries

- Build in option to retry a task
- Define amount of retries before failing
- Build in incremental backoff
- Visible in logs

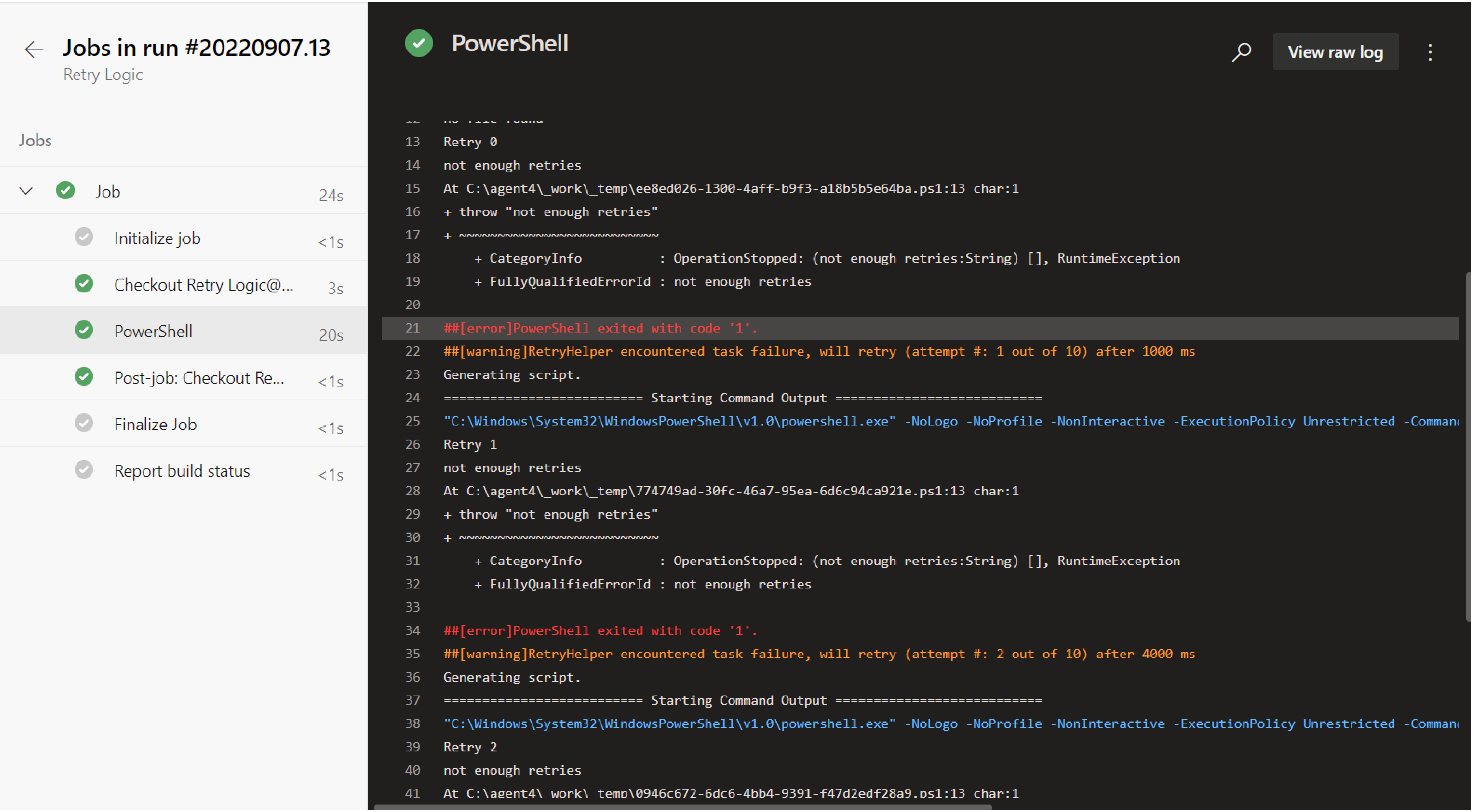


```
- task: PowerShell@2
  inputs:
    targetType: 'inline'
    script: |
      try {
        $count = [int] (Get-Content count.dat)
      }
      catch {
        $count = 0
        Write-Host "no file found"
      }
      Write-Host "Retry $count"
      $count + 1 | Out-File count.dat
      if($count -lt 3){
      throw "not enough retries"
      }
  retryCountOnTaskFailure: 10
```

Retry

Demo

Retry



```
No file found
Retry 0
not enough retries
At C:\agent4\_work\_temp\ee8ed026-1300-4aff-b9f3-a18b5b5e64ba.ps1:13 char:1
+ throw "not enough retries"
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (not enough retries:String) [], RuntimeException
    + FullyQualifiedErrorId : not enough retries

##[error]PowerShell exited with code '1'.
##[warning]RetryHelper encountered task failure, will retry (attempt #: 1 out of 10) after 1000 ms
Generating script.
========================== Starting Command Output ===========================
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -NoLogo -NoProfile -NonInteractive -ExecutionPolicy Unrestricted -Commanc
Retry 1
not enough retries
At C:\agent4\_work\_temp\774749ad-30fc-46a7-95ea-6d6c94ca921e.ps1:13 char:1
+ throw "not enough retries"
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OperationStopped: (not enough retries:String) [], RuntimeException
    + FullyQualifiedErrorId : not enough retries

##[error]PowerShell exited with code '1'.
##[warning]RetryHelper encountered task failure, will retry (attempt #: 2 out of 10) after 4000 ms
Generating script.
========================== Starting Command Output ===========================
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -NoLogo -NoProfile -NonInteractive -ExecutionPolicy Unrestricted -Commanc
Retry 2
not enough retries
At C:\agent4\_work\_temp\0946c672-6dc6-4bb4-9391-f47d2edf28a9.ps1:13 char:1
```

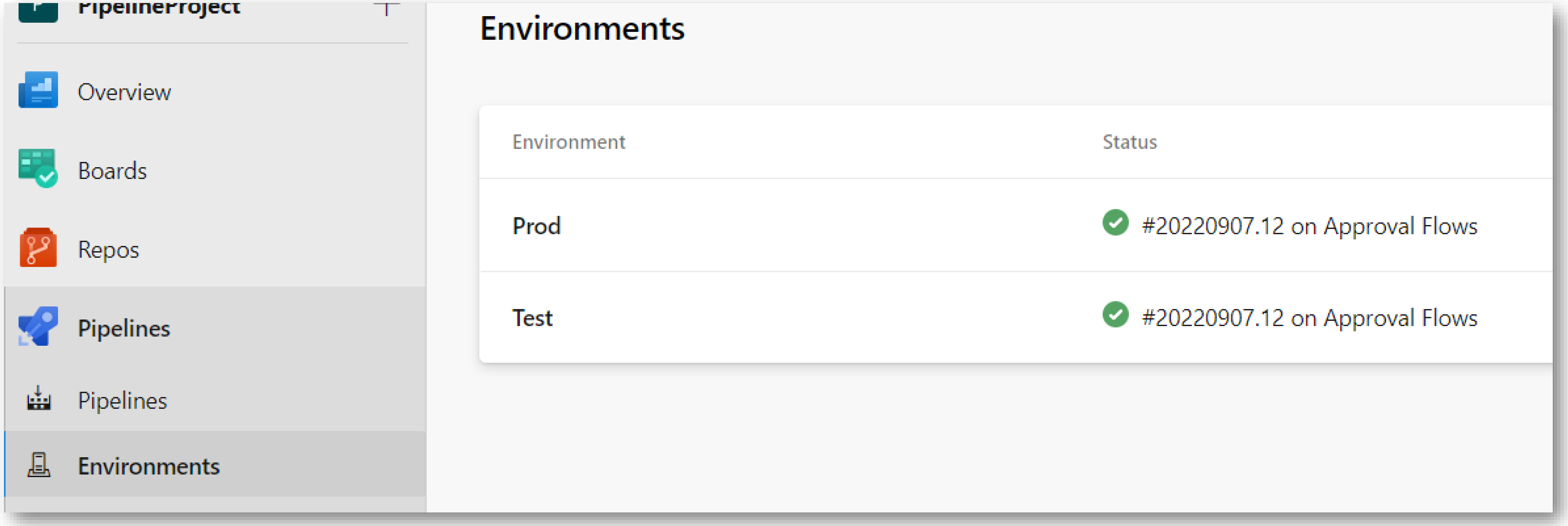
Approvals

/ In Azure DevOps pipelines

Approval

Use Environments

- Allows for deployment jobs to be linked to the environment
- Allows for approvals and checks to be implemented
- Not defined in code



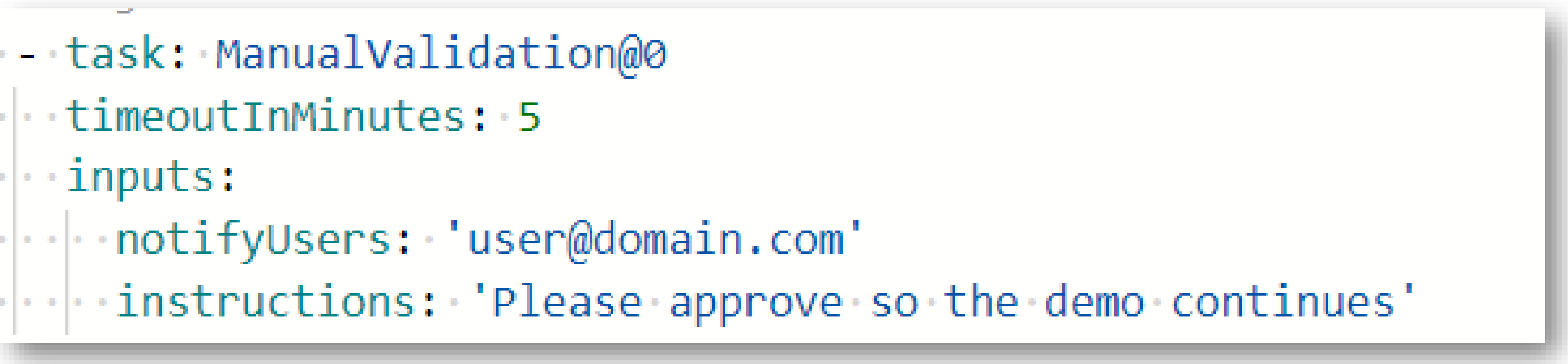
Environments

Environment	Status
Prod	✓ #20220907.12 on Approval Flows
Test	✓ #20220907.12 on Approval Flows

Approval

Use Manual Validation

- Needs to be defined for every job
- Allows for a timeout with specified action
- Defined in code

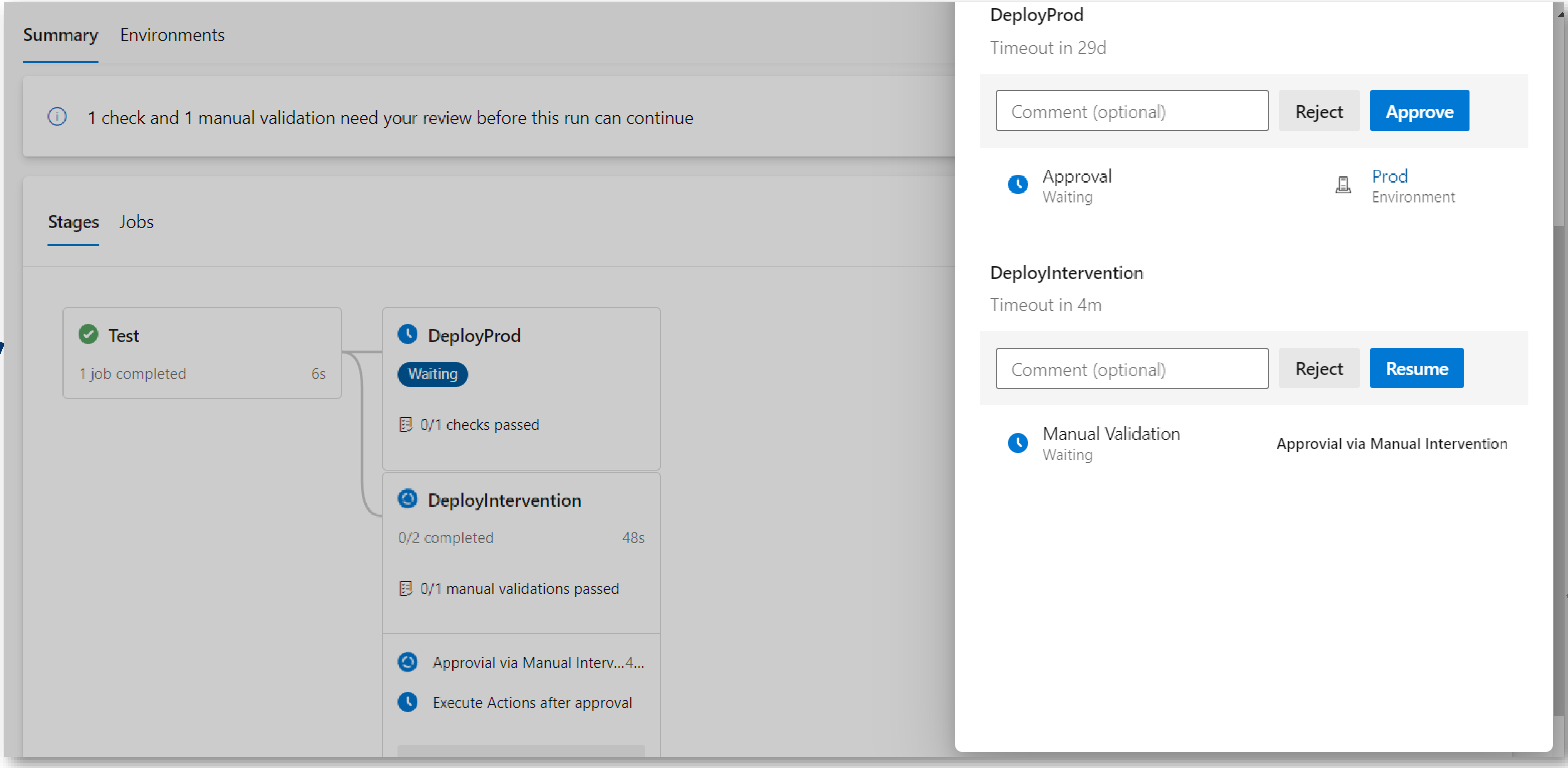


```yaml
- task: ManualValidation@0
  timeoutInMinutes: 5
  inputs:
    notifyUsers: 'user@domain.com'
    instructions: 'Please approve so the demo continues'
```

Approval

Demo

Approval



Summary Environments

ⓘ 1 check and 1 manual validation need your review before this run can continue

Stages Jobs

Test
1 job completed 6s

DeployProd
Waiting
0/1 checks passed

DeployIntervention
0/2 completed 48s
0/1 manual validations passed
Approvial via Manual Interv...4...
Execute Actions after approval

DeployProd
Timeout in 29d
Comment (optional) Reject Approve
Approval
Waiting
Prod
Environment

DeployIntervention
Timeout in 4m
Comment (optional) Reject Resume
Manual Validation
Waiting
Approvial via Manual Intervention

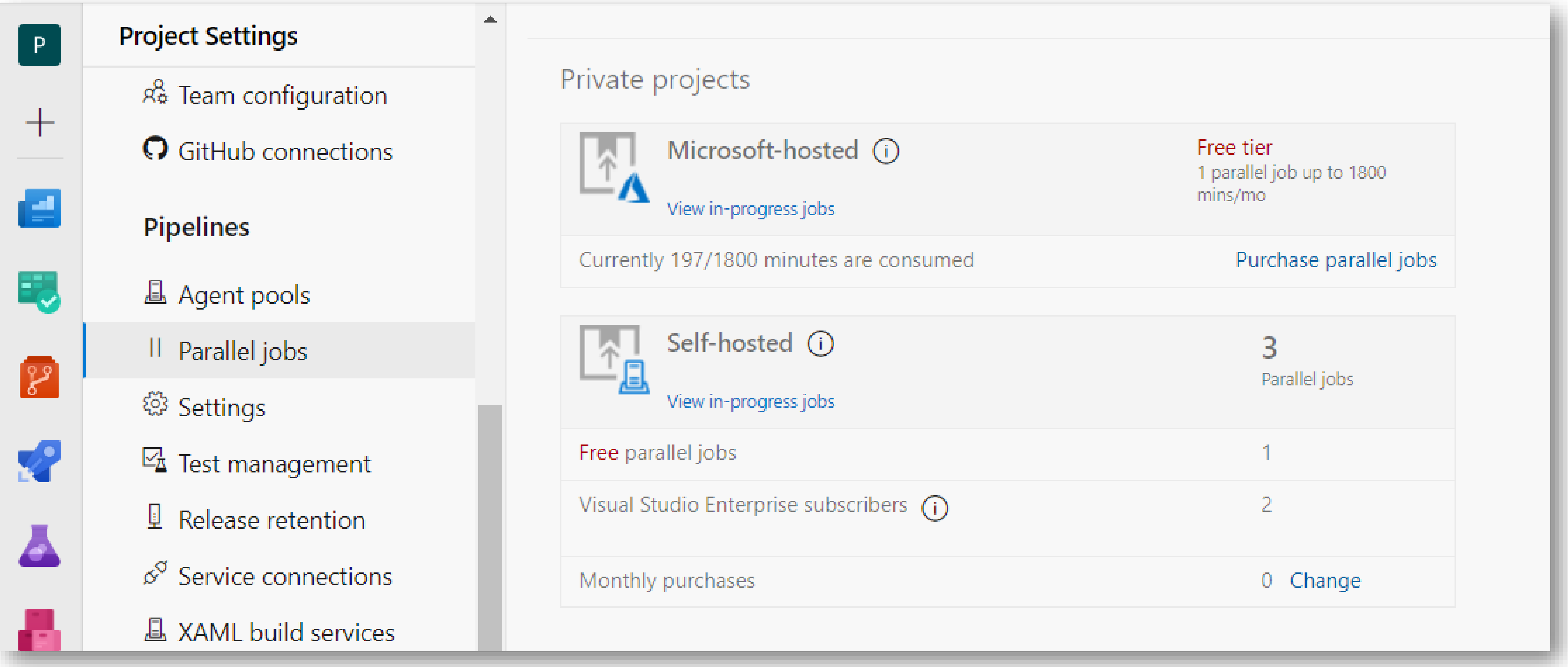
Parallelize Jobs

/ In Azure DevOps pipelines

Parallelize

Parallel jobs

- Need to buy option to run jobs in parallel
- Allow for Microsoft-hosted or Self-hosted
- $40 for extra Microsoft-hosted, $15 for extra Self-hosted (per month)
- Define multiple agents (on the same machine) to run in parallel



Private projects

Microsoft-hosted

Free tier
1 parallel job up to 1800 mins/mo

Currently 197/1800 minutes are consumed

Self-hosted

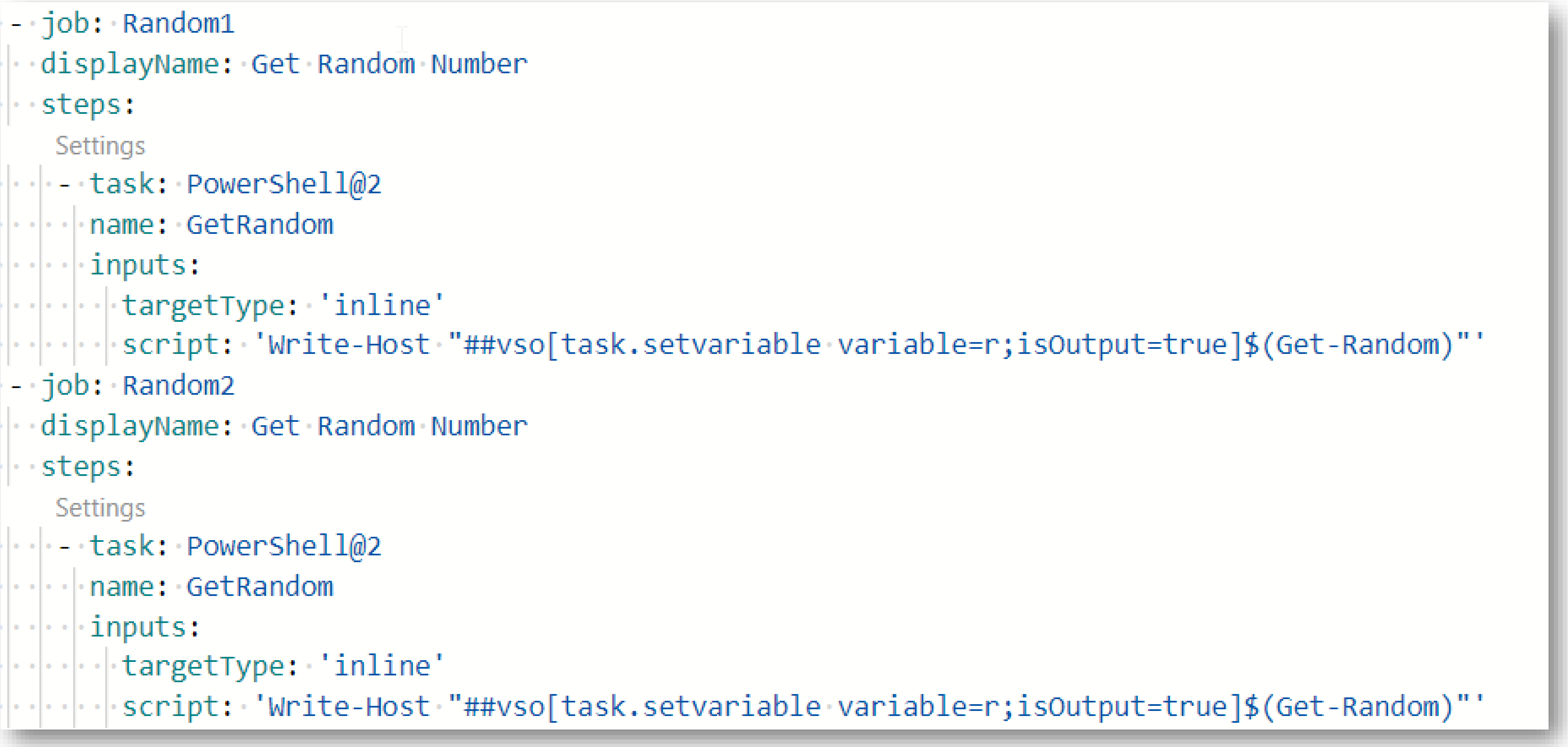
3
Parallel jobs

Free parallel jobs	1
Visual Studio Enterprise subscribers	2
Monthly purchases	0

Parallelize

Use dependsOn

- Define different jobs
- Use depensOn to determine order of execution and which ones can go in parallel

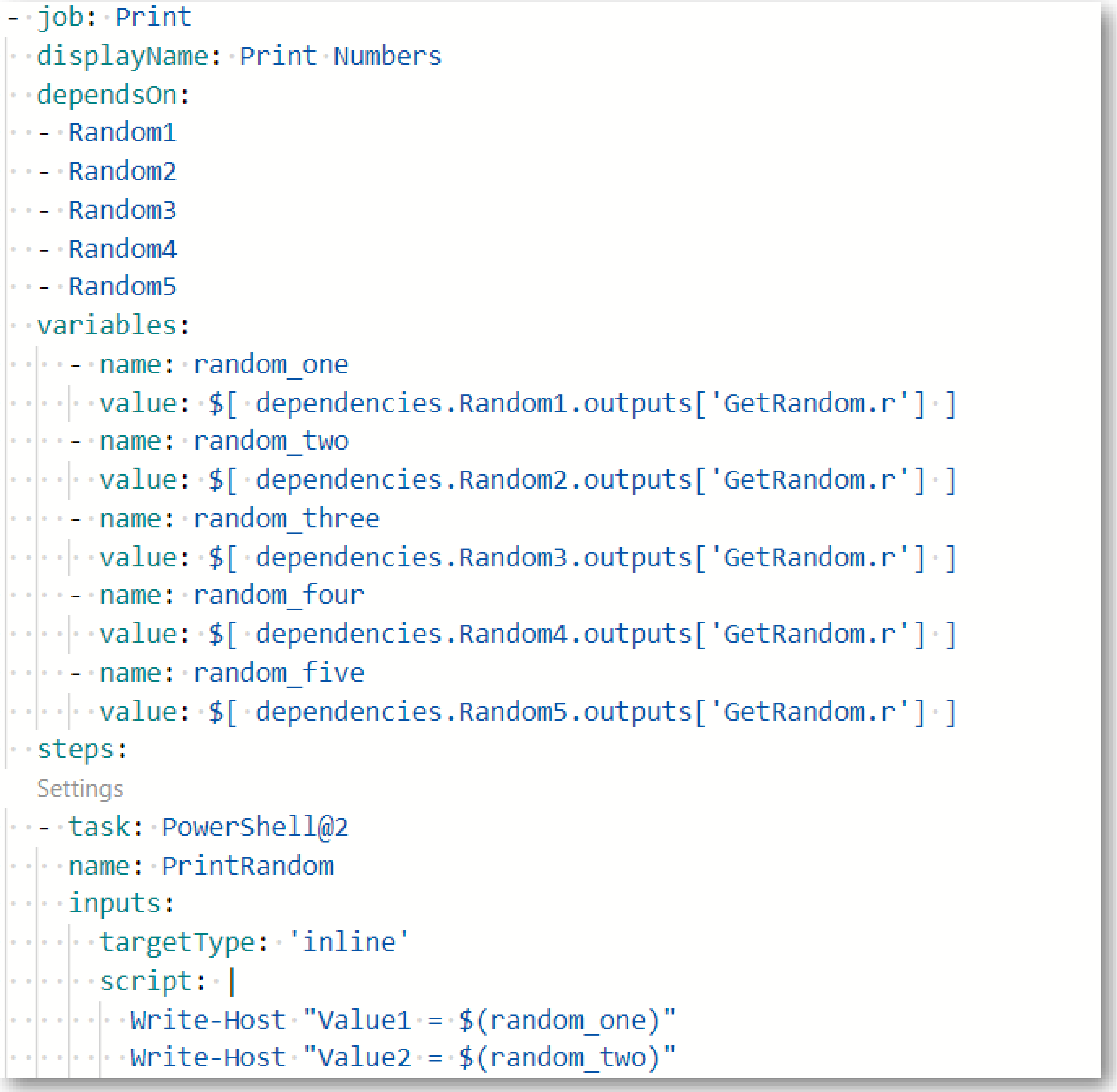


```
- job: Random1
  displayName: Get Random Number
  steps:
   Settings
  - task: PowerShell@2
    name: GetRandom
    inputs:
      targetType: 'inline'
      script: 'Write-Host "##vso[task.setvariable variable=r;isOutput=true]$(Get-Random)"'
- job: Random2
  displayName: Get Random Number
  steps:
   Settings
  - task: PowerShell@2
    name: GetRandom
    inputs:
      targetType: 'inline'
      script: 'Write-Host "##vso[task.setvariable variable=r;isOutput=true]$(Get-Random)"'
```

Parallelize

Use dependsOn

- Define different jobs
- Use depensOn to determine order of execution and which ones can go in parallel
- Use dependencies to get values from jobs

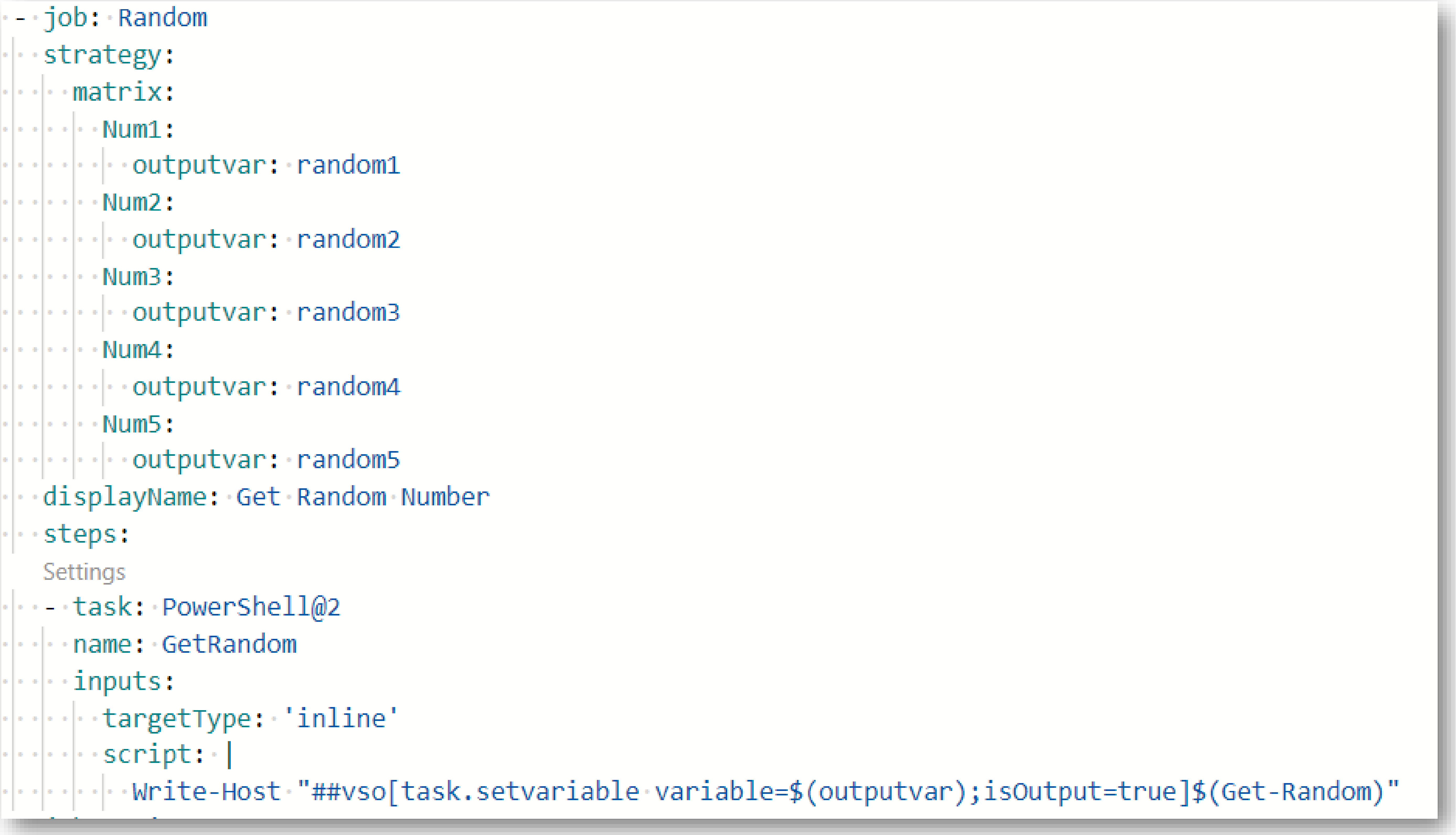


```
- job: Print
  displayName: Print Numbers
  dependsOn:
  - Random1
  - Random2
  - Random3
  - Random4
  - Random5
  variables:
    - name: random_one
      value: $[ dependencies.Random1.outputs['GetRandom.r'] ]
    - name: random_two
      value: $[ dependencies.Random2.outputs['GetRandom.r'] ]
    - name: random_three
      value: $[ dependencies.Random3.outputs['GetRandom.r'] ]
    - name: random_four
      value: $[ dependencies.Random4.outputs['GetRandom.r'] ]
    - name: random_five
      value: $[ dependencies.Random5.outputs['GetRandom.r'] ]
  steps:
  Settings
  - task: PowerShell@2
    name: PrintRandom
    inputs:
      targetType: 'inline'
      script: |
        Write-Host "Value1 = $(random_one)"
        Write-Host "Value2 = $(random_two)"
```

Parallelize

Use matrix

- Define multiple of the same job
- Possibility to add variables for specific jobs
- Output variables have name appended so are not overwritten

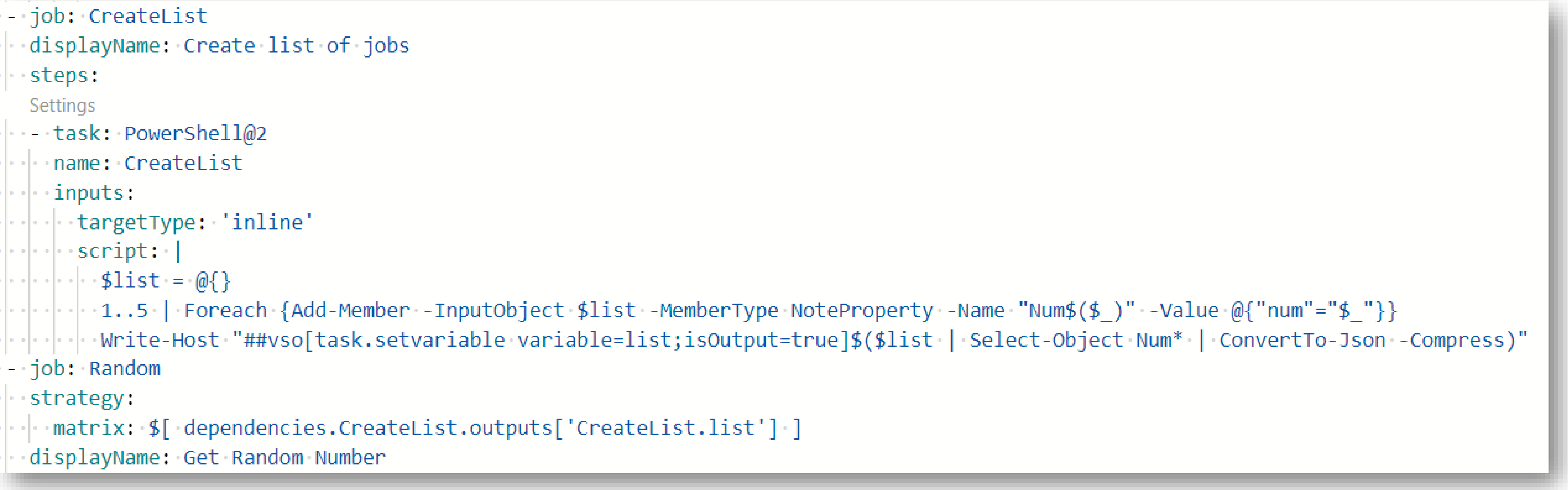


```yaml
- job: Random
  strategy:
    matrix:
      Num1:
        outputvar: random1
      Num2:
        outputvar: random2
      Num3:
        outputvar: random3
      Num4:
        outputvar: random4
      Num5:
        outputvar: random5
  displayName: Get Random Number
  steps:
  Settings
  - task: PowerShell@2
    name: GetRandom
    inputs:
      targetType: 'inline'
      script: |
        Write-Host "##vso[task.setvariable variable=$(outputvar);isOutput=true]$(Get-Random)"
```

Parallelize

Build matrix in runtime

- Matrix is generated in runtime
- Allow for creation of stringyfied JSON to describe Matrix
- Using for example azure storage queue instead of output variables to make output size variable too

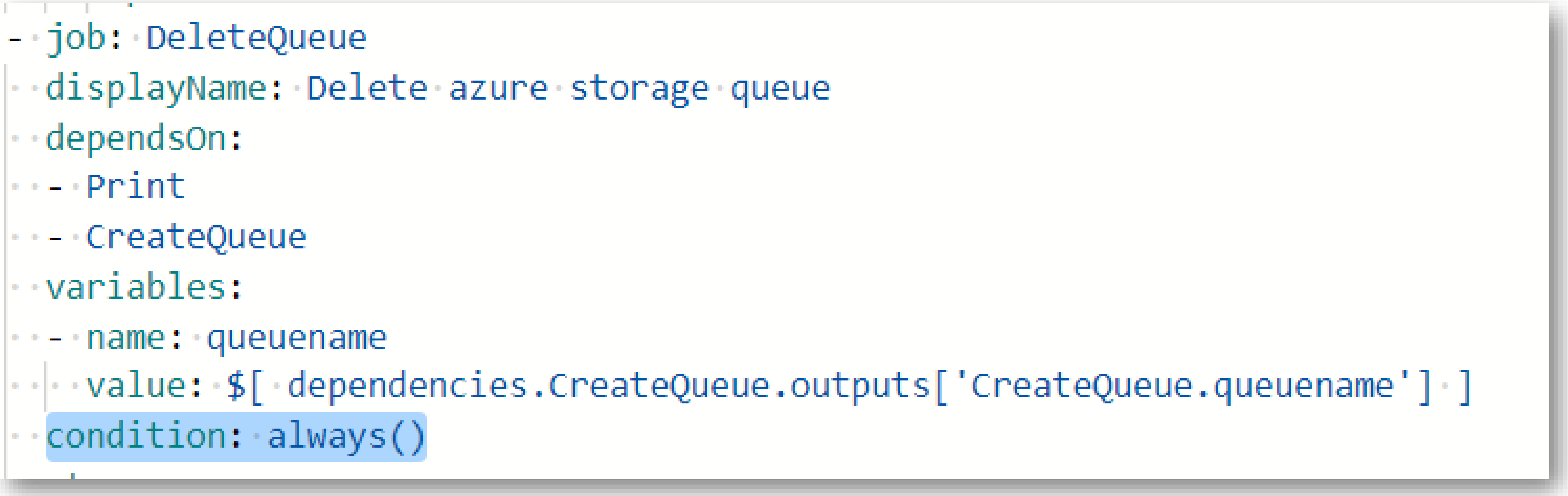


```
- job: CreateList
  displayName: Create list of jobs
  steps:
  Settings
  - task: PowerShell@2
    name: CreateList
    inputs:
      targetType: 'inline'
      script: |
        $list = @{}
        1..5 | Foreach {Add-Member -InputObject $list -MemberType NoteProperty -Name "Num$($_)" -Value @{"num"="$_"}}
        Write-Host "##vso[task.setvariable variable=list;isOutput=true]$($list | Select-Object Num* | ConvertTo-Json -Compress)"
- job: Random
  strategy:
    matrix: $[ dependencies.CreateList.outputs['CreateList.list'] ]
  displayName: Get Random Number
```

Parallelize

Use condition always()

- Useful for cleanup actions
- Will run even if jobs before it fail

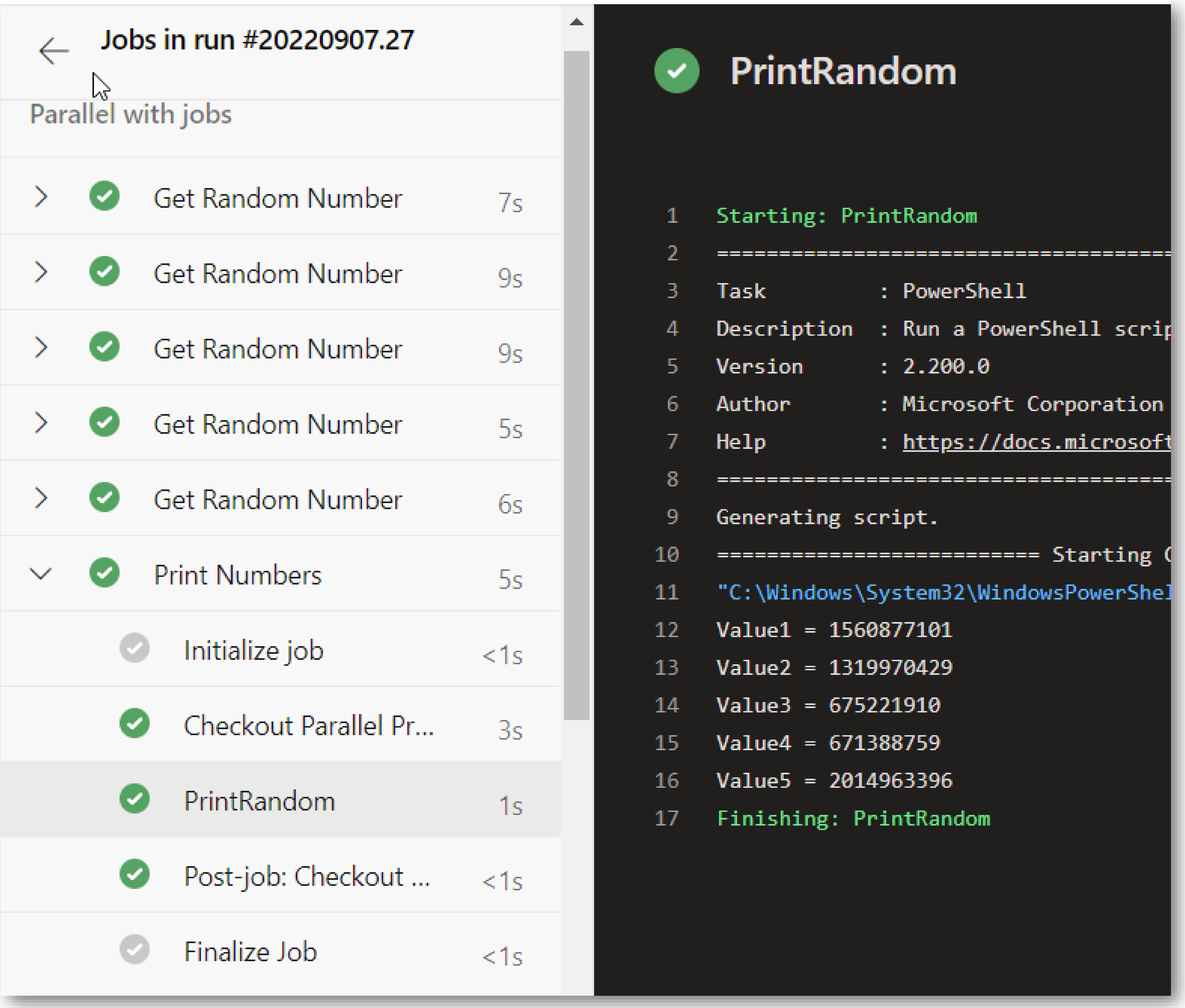


```yaml
- job: DeleteQueue
  displayName: Delete azure storage queue
  dependsOn:
  - Print
  - CreateQueue
  variables:
  - name: queuename
    value: $[ dependencies.CreateQueue.outputs['CreateQueue.queuename'] ]
  condition: always()
```

Parallelize

Demo

Parallelize



Jobs in run #20220907.27

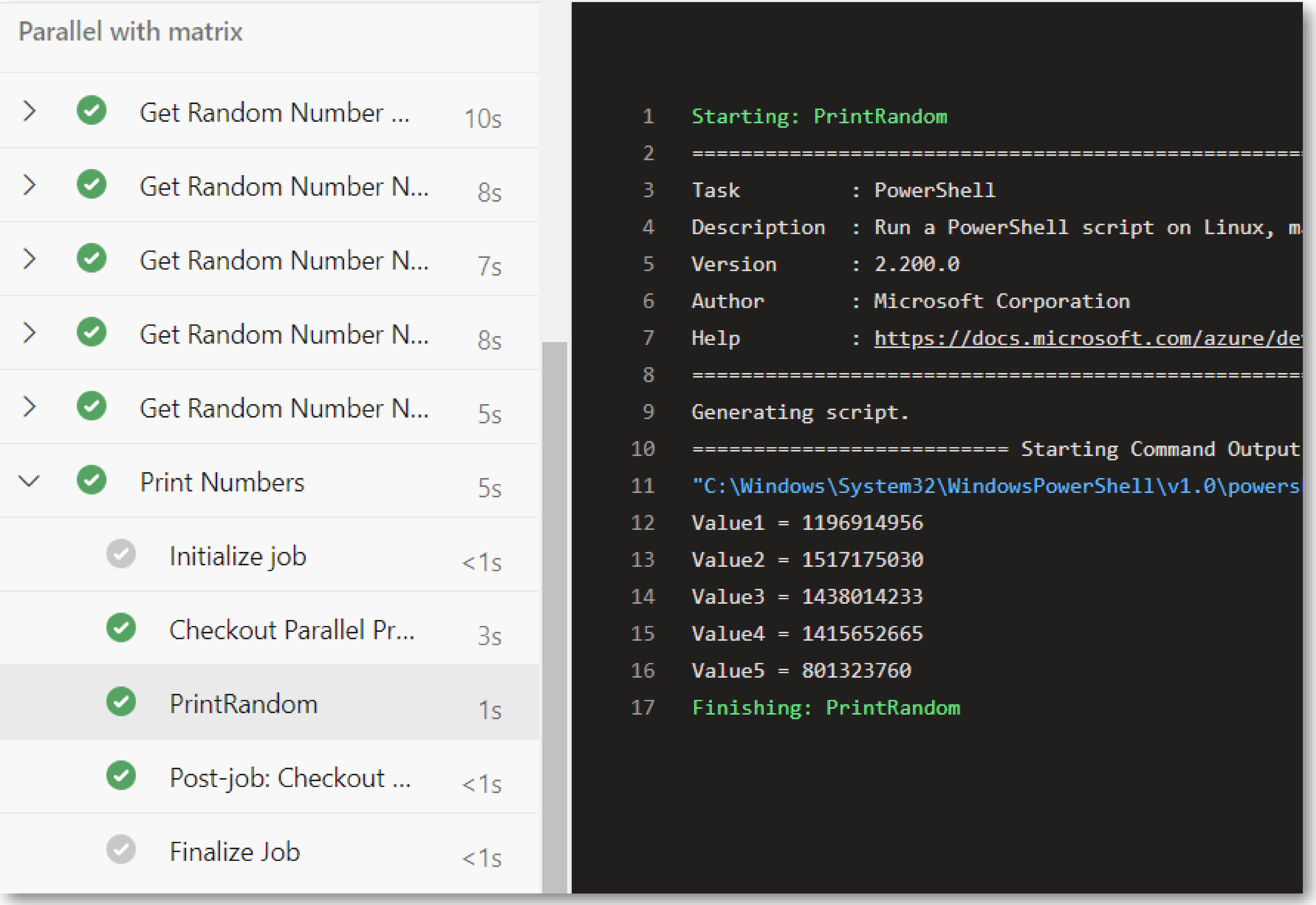
Parallel with jobs

- ✓ Get Random Number 7s
- ✓ Get Random Number 9s
- ✓ Get Random Number 9s
- ✓ Get Random Number 5s
- ✓ Get Random Number 6s
- ✓ Print Numbers 5s
  - ✓ Initialize job <1s
  - ✓ Checkout Parallel Pr... 3s
  - ✓ PrintRandom 1s
  - ✓ Post-job: Checkout ... <1s
  - ✓ Finalize Job <1s

✓ PrintRandom

```
Starting: PrintRandom
====================================
Task         : PowerShell
Description  : Run a PowerShell scrip
Version      : 2.200.0
Author       : Microsoft Corporation
Help         : https://docs.microsoft
====================================
Generating script.
========================== Starting C
"C:\Windows\System32\WindowsPowerShel
Value1 = 1560877101
Value2 = 1319970429
Value3 = 675221910
Value4 = 671388759
Value5 = 2014963396
Finishing: PrintRandom
```

Parallelize

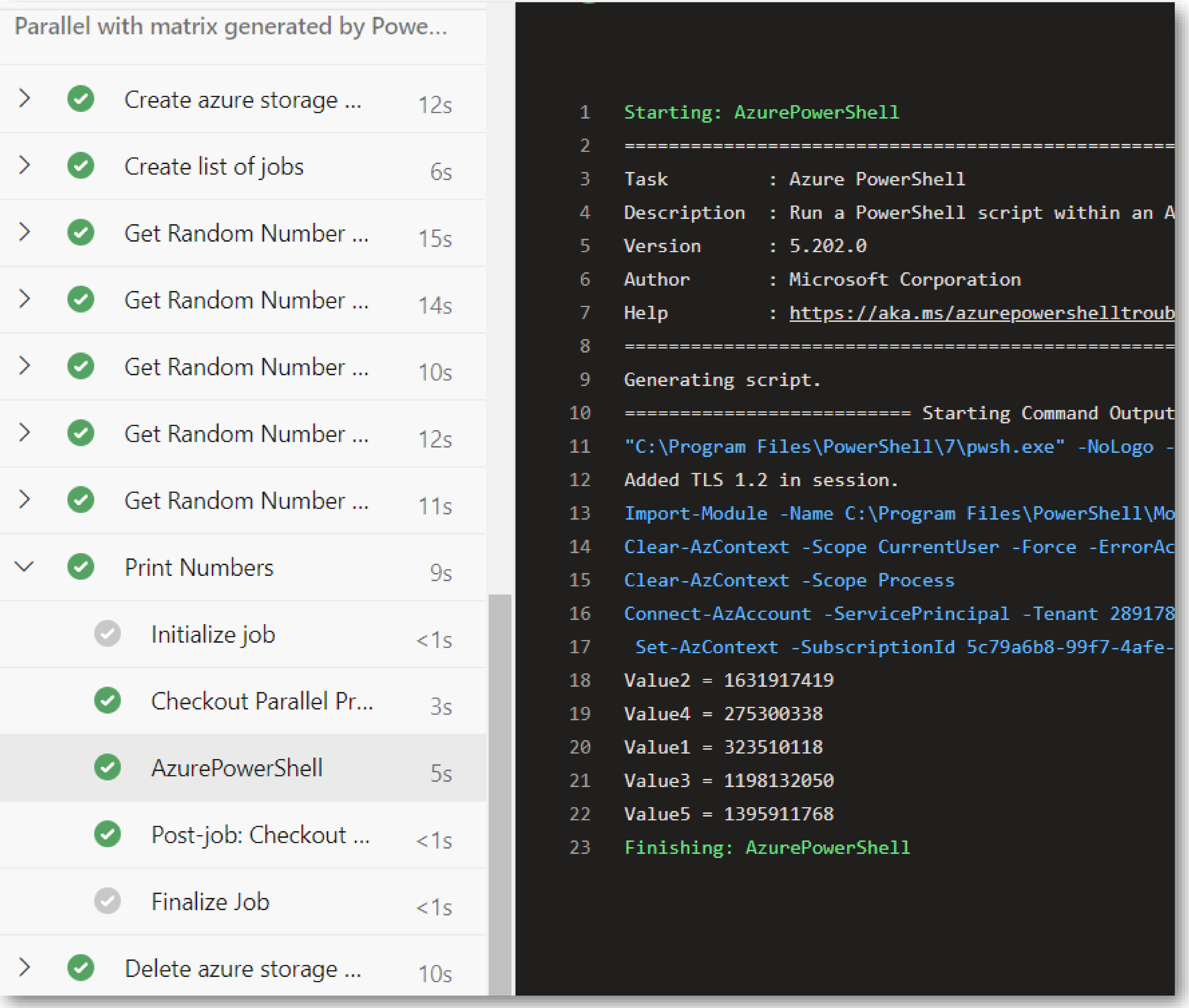


Parallel with matrix

- › ✓ Get Random Number ... 10s
- › ✓ Get Random Number N... 8s
- › ✓ Get Random Number N... 7s
- › ✓ Get Random Number N... 8s
- › ✓ Get Random Number N... 5s
- ⌄ ✓ Print Numbers 5s
  - ✓ Initialize job <1s
  - ✓ Checkout Parallel Pr... 3s
  - ✓ PrintRandom 1s
  - ✓ Post-job: Checkout ... <1s
  - ✓ Finalize Job <1s

```
Starting: PrintRandom
==============================================================
Task         : PowerShell
Description  : Run a PowerShell script on Linux, ma
Version      : 2.200.0
Author       : Microsoft Corporation
Help         : https://docs.microsoft.com/azure/de
==============================================================
Generating script.
========================== Starting Command Output
"C:\Windows\System32\WindowsPowerShell\v1.0\powers
Value1 = 1196914956
Value2 = 1517175030
Value3 = 1438014233
Value4 = 1415652665
Value5 = 801323760
Finishing: PrintRandom
```

Parallelize



Parallel with matrix generated by Powe...

- ✓ Create azure storage ... 12s
- ✓ Create list of jobs 6s
- ✓ Get Random Number ... 15s
- ✓ Get Random Number ... 14s
- ✓ Get Random Number ... 10s
- ✓ Get Random Number ... 12s
- ✓ Get Random Number ... 11s
- ✓ Print Numbers 9s
  - ✓ Initialize job <1s
  - ✓ Checkout Parallel Pr... 3s
  - ✓ AzurePowerShell 5s
  - ✓ Post-job: Checkout ... <1s
  - ✓ Finalize Job <1s
- ✓ Delete azure storage ... 10s

```
Starting: AzurePowerShell
==============================================================
Task         : Azure PowerShell
Description  : Run a PowerShell script within an A
Version      : 5.202.0
Author       : Microsoft Corporation
Help         : https://aka.ms/azurepowershelltroub
==============================================================
Generating script.
========================== Starting Command Output
"C:\Program Files\PowerShell\7\pwsh.exe" -NoLogo -
Added TLS 1.2 in session.
Import-Module -Name C:\Program Files\PowerShell\Mo
Clear-AzContext -Scope CurrentUser -Force -ErrorAc
Clear-AzContext -Scope Process
Connect-AzAccount -ServicePrincipal -Tenant 289178
 Set-AzContext -SubscriptionId 5c79a6b8-99f7-4afe-
Value2 = 1631917419
Value4 = 275300338
Value1 = 323510118
Value3 = 1198132050
Value5 = 1395911768
Finishing: AzurePowerShell
```

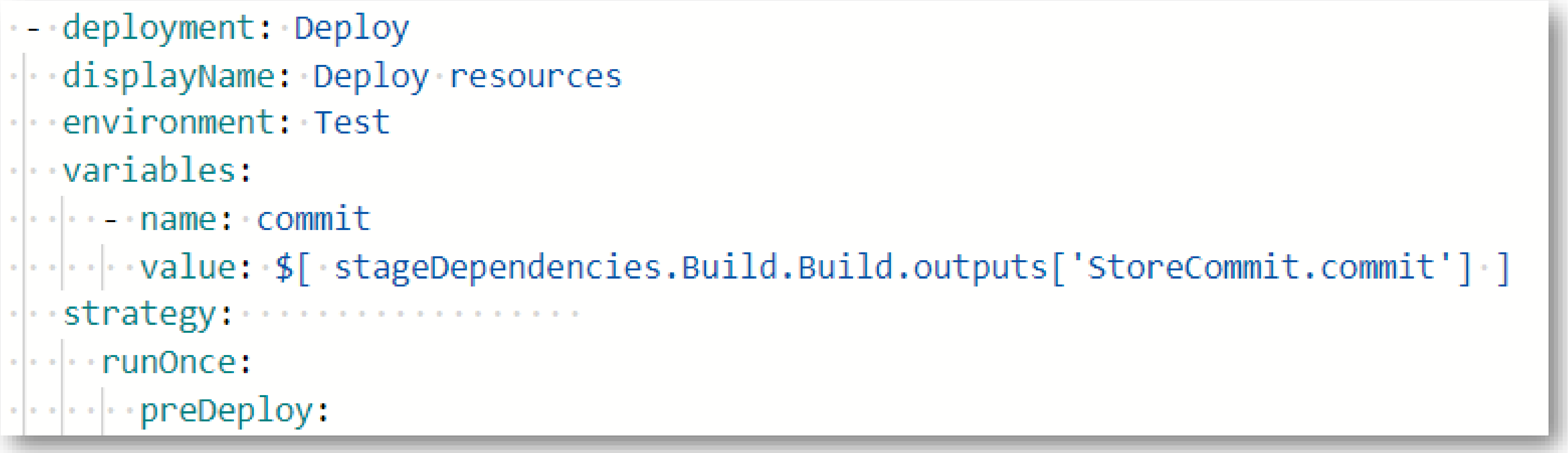
Automatic Rollback

/ In Azure DevOps pipelines

Rollback

Use deployment jobs

- Define different deployment levels
  - preDeploy, deploy, routeTraffic, postRouteTraffic, on
- Define what happens [on] Succes and [on] Failure

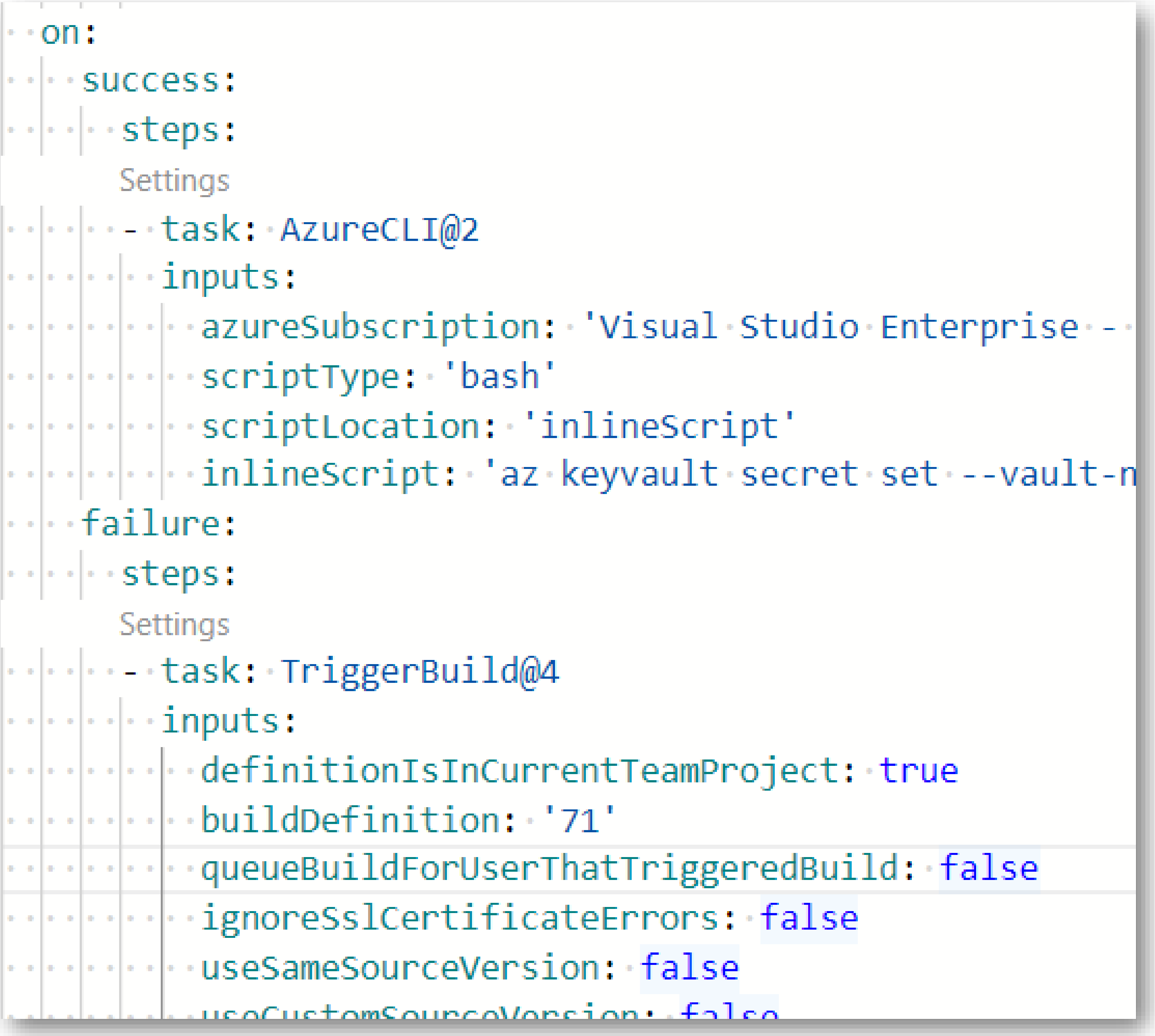


```
- deployment: Deploy
  displayName: Deploy resources
  environment: Test
  variables:
    - name: commit
      value: $[ stageDependencies.Build.Build.outputs['StoreCommit.commit'] ]
  strategy:
    runOnce:
      preDeploy:
```

Rollback

How to handle success and failure

- On success
  - Store the last known good configuration commit ID
- On failure
  - Trigger a build which will use the last known good configuration
- Trigger build can be done by API or marketplace task

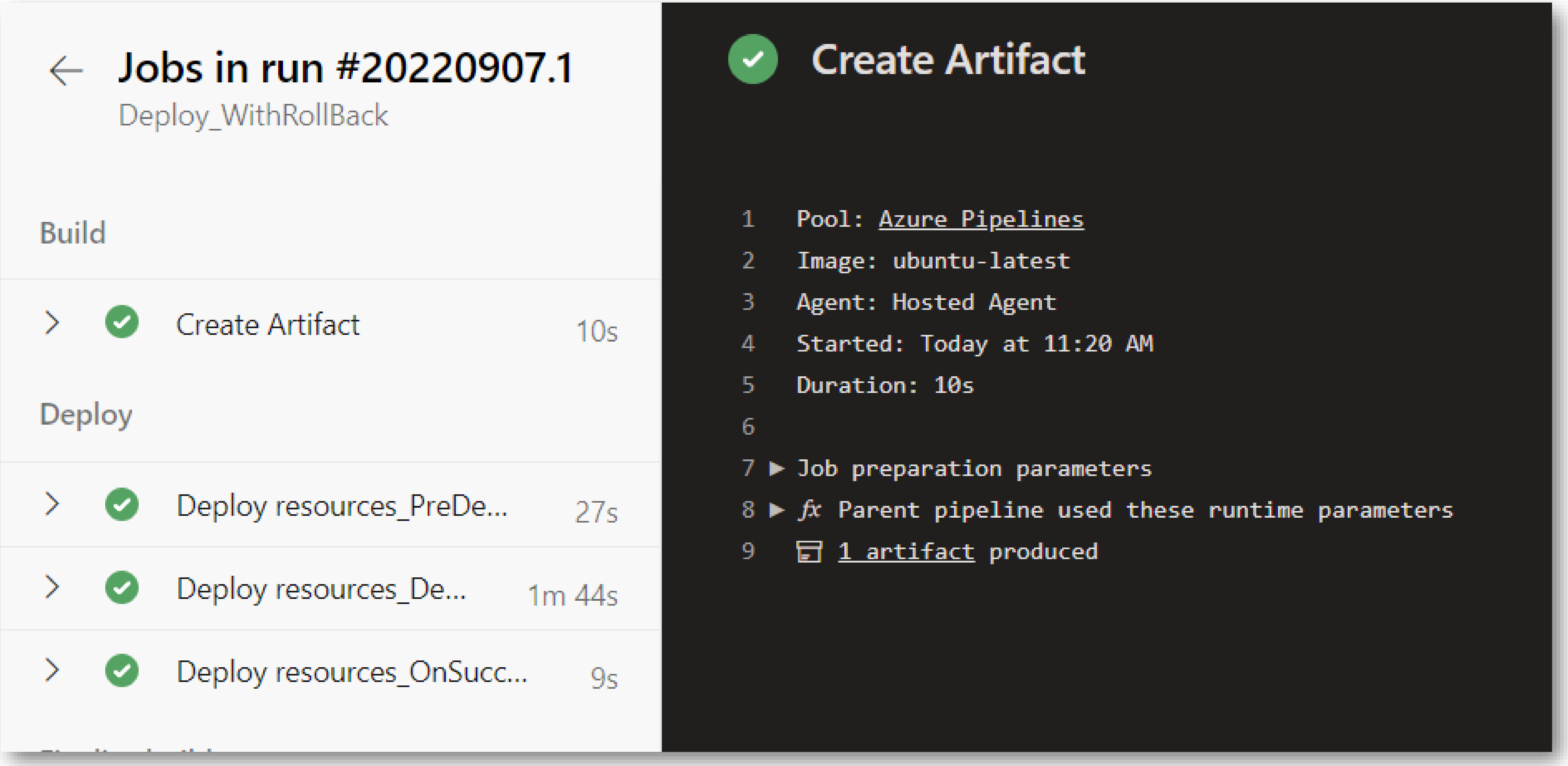


```yaml
  on:
    success:
      steps:
      Settings
      - task: AzureCLI@2
        inputs:
          azureSubscription: 'Visual Studio Enterprise - 
          scriptType: 'bash'
          scriptLocation: 'inlineScript'
          inlineScript: 'az keyvault secret set --vault-n
    failure:
      steps:
      Settings
      - task: TriggerBuild@4
        inputs:
          definitionIsInCurrentTeamProject: true
          buildDefinition: '71'
          queueBuildForUserThatTriggeredBuild: false
          ignoreSslCertificateErrors: false
          useSameSourceVersion: false
          useCustomSourceVersion: false
```

Rollback

Demo

Rollback



Jobs in run #20220907.1

Deploy_WithRollBack

Build

Create Artifact 10s

Deploy

Deploy resources_PreDe... 27s

Deploy resources_De... 1m 44s

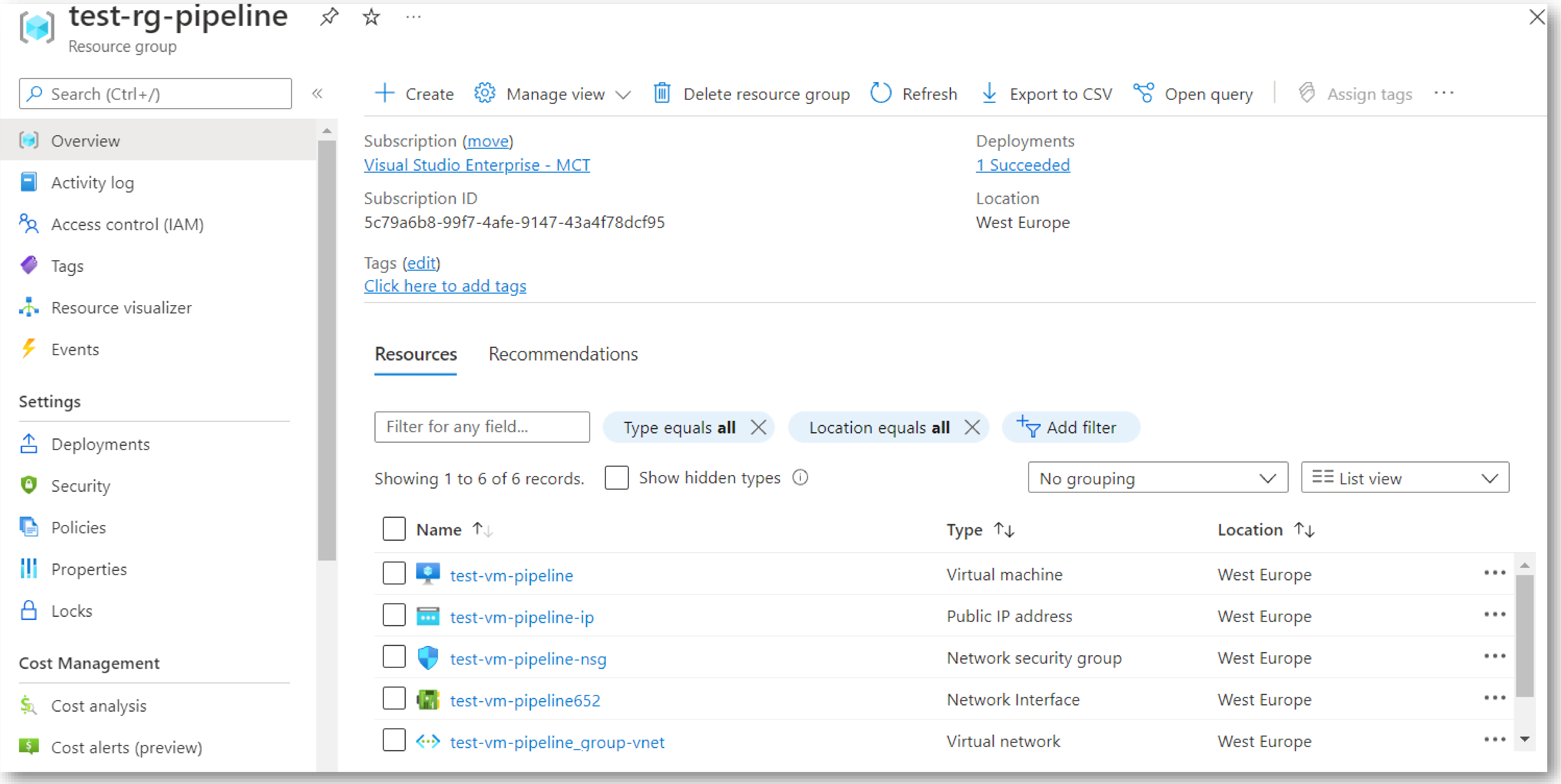
Deploy resources_OnSucc... 9s

Create Artifact

```
Pool: Azure Pipelines
Image: ubuntu-latest
Agent: Hosted Agent
Started: Today at 11:20 AM
Duration: 10s

▶ Job preparation parameters
▶ fx Parent pipeline used these runtime parameters
  1 artifact produced
```

Rollback



test-rg-pipeline

Resource group

Subscription (move)
Visual Studio Enterprise - MCT

Subscription ID
5c79a6b8-99f7-4afe-9147-43a4f78dcf95

Tags (edit)
Click here to add tags

Deployments
1 Succeeded

Location
West Europe

Type equals **all**

Location equals **all**

Showing 1 to 6 of 6 records.

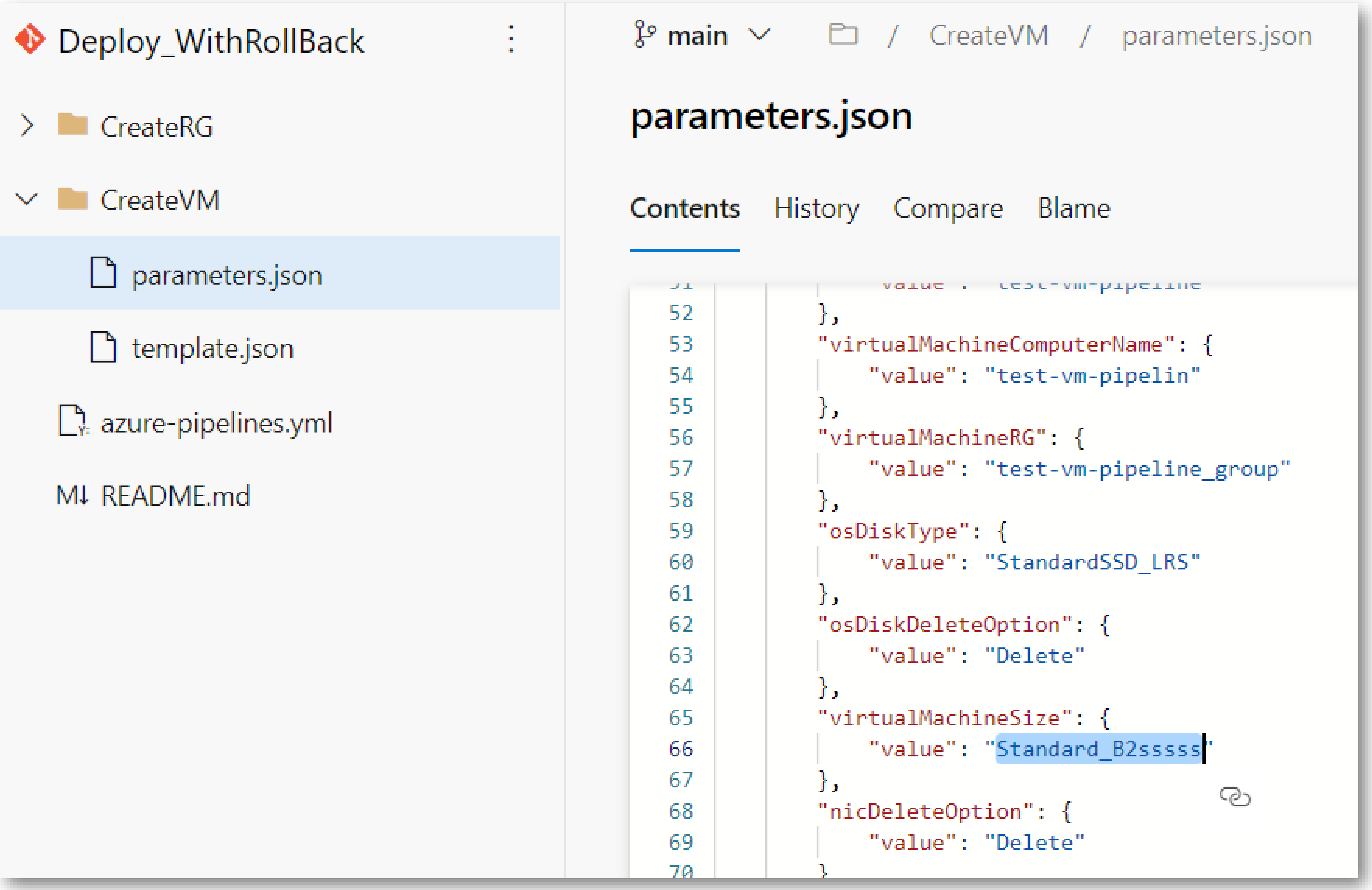
□ Show hidden types

No grouping

List view

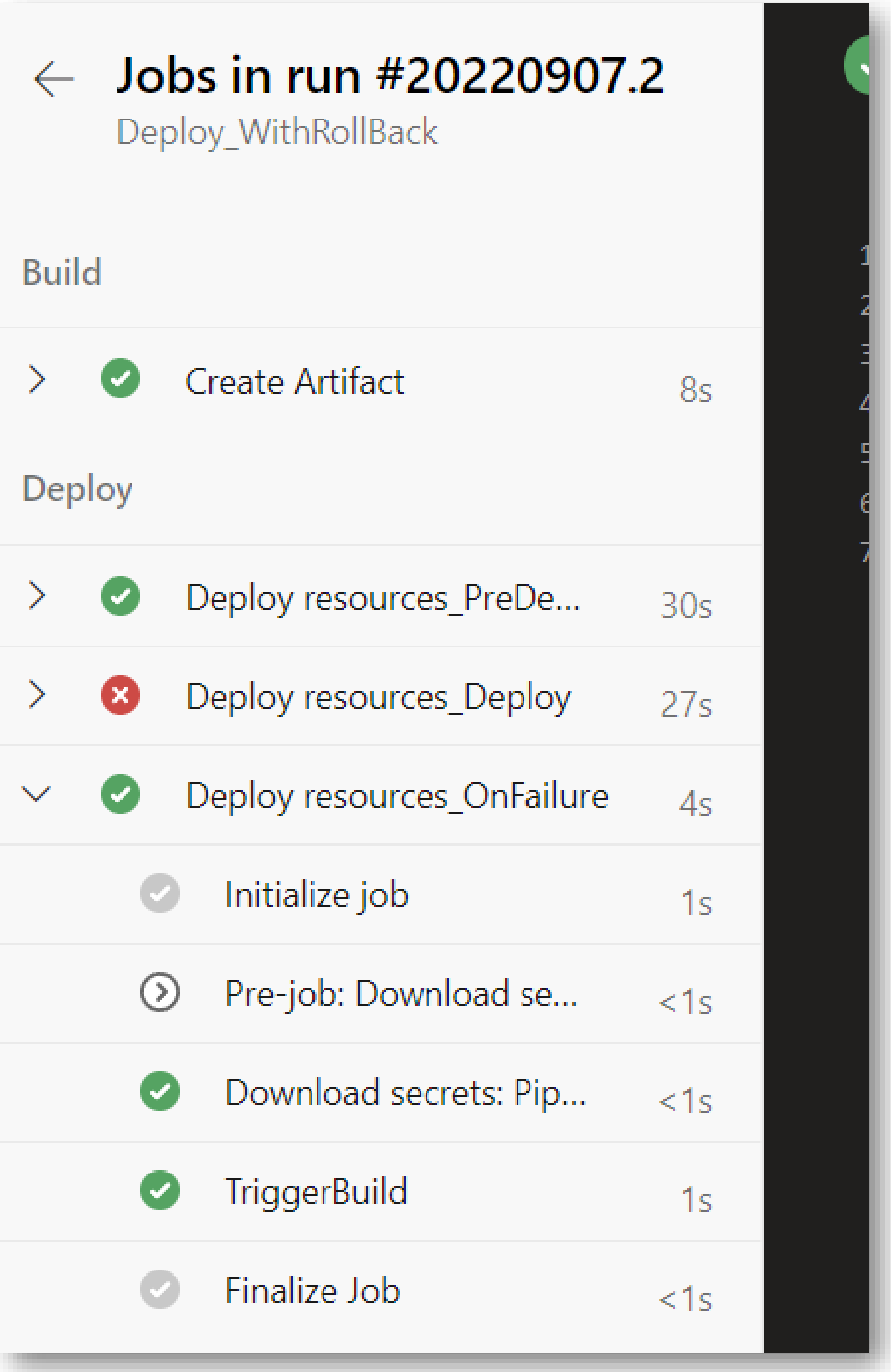
□ Name	Type	Location
□ test-vm-pipeline	Virtual machine	West Europe
□ test-vm-pipeline-ip	Public IP address	West Europe
□ test-vm-pipeline-nsg	Network security group	West Europe
□ test-vm-pipeline652	Network Interface	West Europe
□ test-vm-pipeline_group-vnet	Virtual network	West Europe

Rollback



```
            "value": "test-vm-pipeline"
        },
        "virtualMachineComputerName": {
            "value": "test-vm-pipelin"
        },
        "virtualMachineRG": {
            "value": "test-vm-pipeline_group"
        },
        "osDiskType": {
            "value": "StandardSSD_LRS"
        },
        "osDiskDeleteOption": {
            "value": "Delete"
        },
        "virtualMachineSize": {
            "value": "Standard_B2sssss"
        },
        "nicDeleteOption": {
            "value": "Delete"
        }
```

Rollback



Jobs in run #20220907.2

Deploy_WithRollBack

Build

› ✓ Create Artifact 8s

Deploy

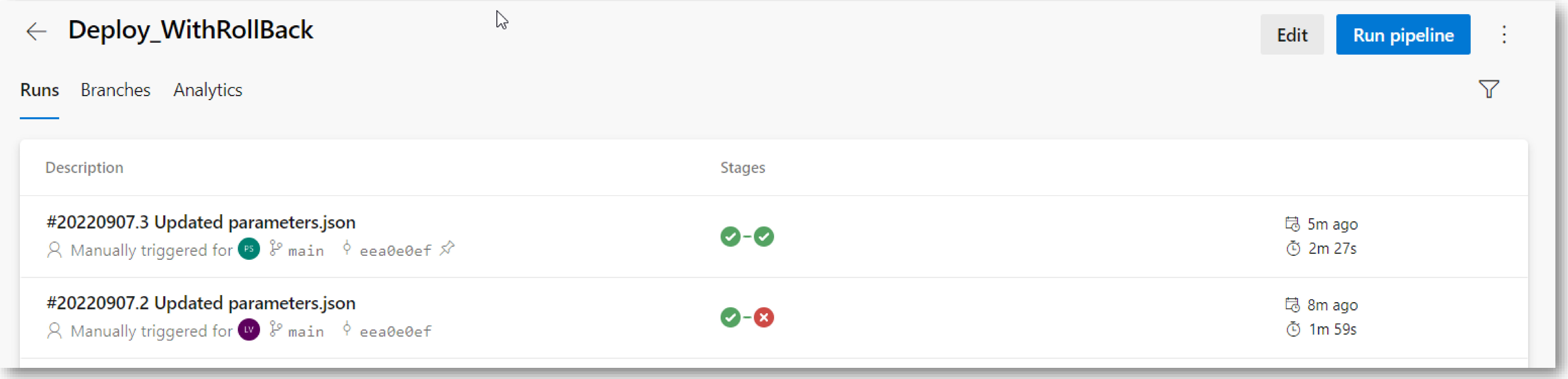
› ✓ Deploy resources_PreDe... 30s

› ✗ Deploy resources_Deploy 27s

⌄ ✓ Deploy resources_OnFailure 4s

- ✓ Initialize job 1s
- › Pre-job: Download se... <1s
- ✓ Download secrets: Pip... <1s
- ✓ TriggerBuild 1s
- ✓ Finalize Job <1s

Rollback



Description	Stages	
#20220907.3 Updated parameters.json Manually triggered for PS main eea0e0ef	✓–✓	5m ago 2m 27s
#20220907.2 Updated parameters.json Manually triggered for LV main eea0e0ef	✓–✗	8m ago 1m 59s

Questions?

Leo Visser

- Twitter: @autosysops
- GitHub: www.github.com/AutoSysOps
- Blog: www.autosysops.com

AUTO-SYS-OPS

Thank you for listening!