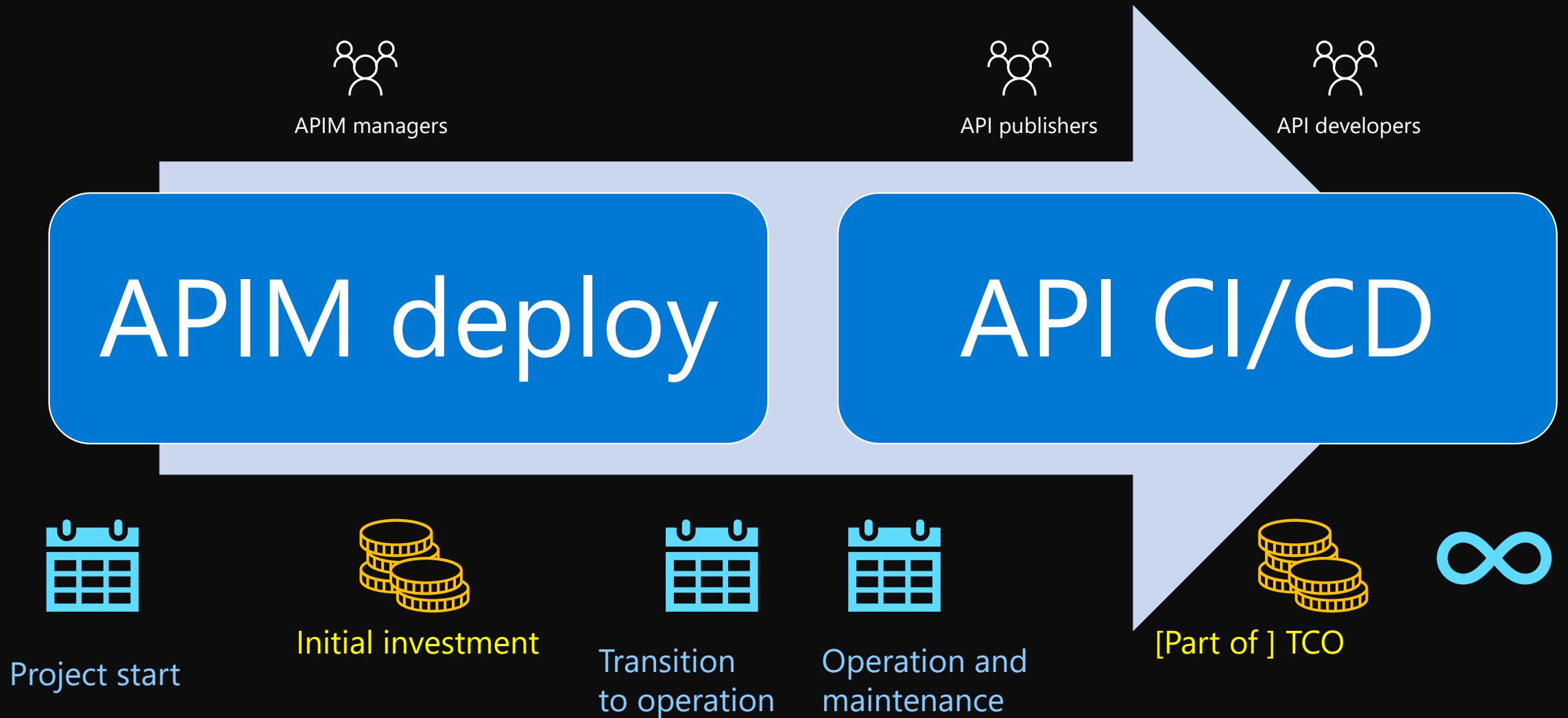


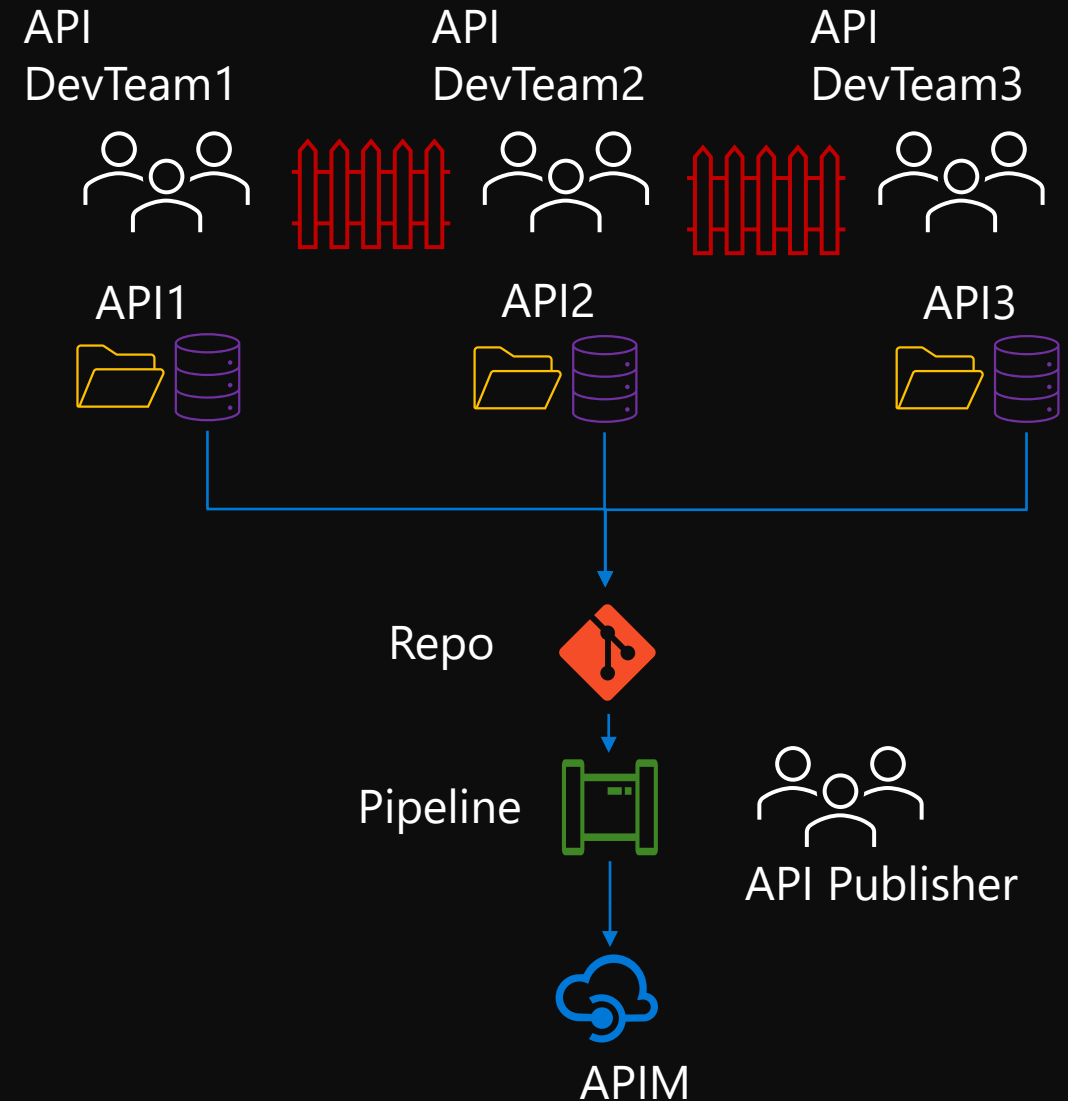
API Deployment approach
mbecker@microsoft.com
2021

APIM lifecycle



API deployment challenges

- **Isolated** API development teams
 - In some cases: internal, outsourced, external
- **Scope**: products, APIs, policies, other assets
- **IP ownership**
- **Collision prevention**
- **Async development** cycles
- **Consistency** requirement
- **Full source** for each APIM deployment and milestone
- **Staging** concept with granulated privileges



APIM CI/CD approach

1

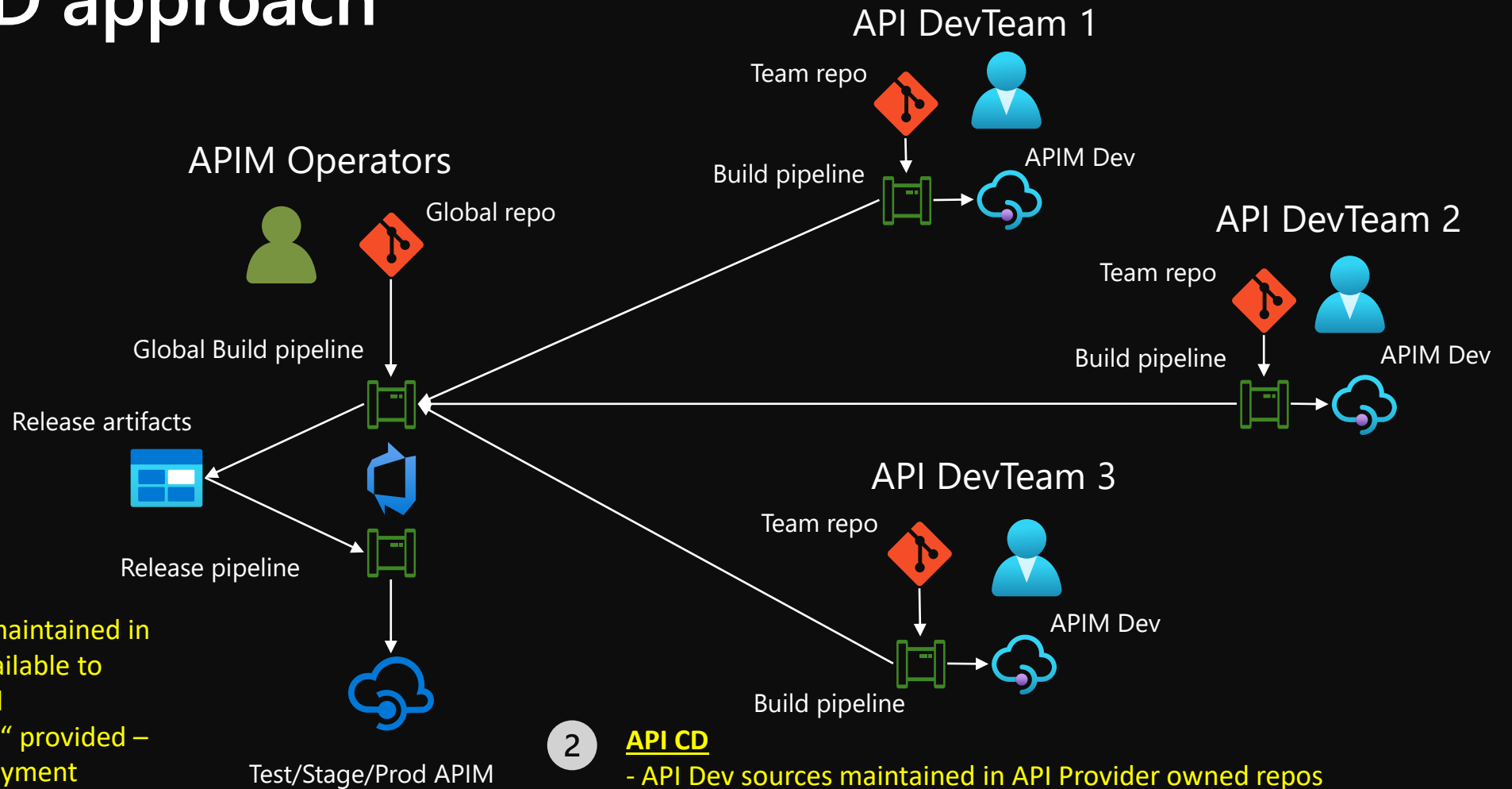
APIM CD

- Main APIM repository for APIM service
- Maintained by APIM Mgmt staff
- Separate build/release pipeline

3

Requirements satisfied

- Chronology of artifacts maintained in global pipeline and is available to redeploy without rebuild
- „Least privilege principle“ provided – centrally managed deployment
- IP protection provided – isolated work of API Providers
- Build artifacts are deployment scripts and satellite components, thus verifiable deployment source available



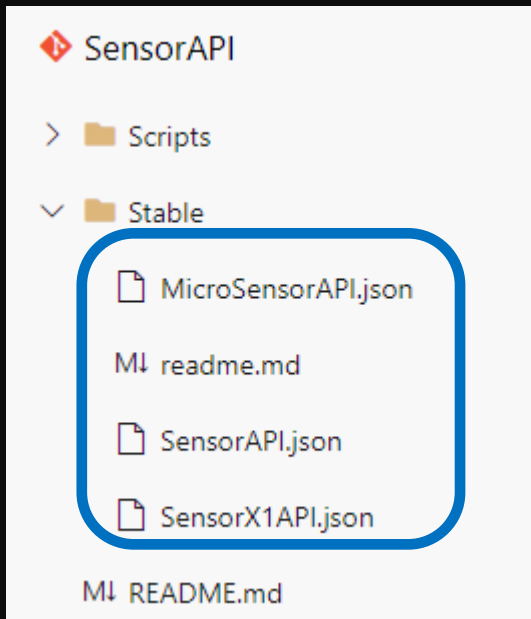
2

API CD

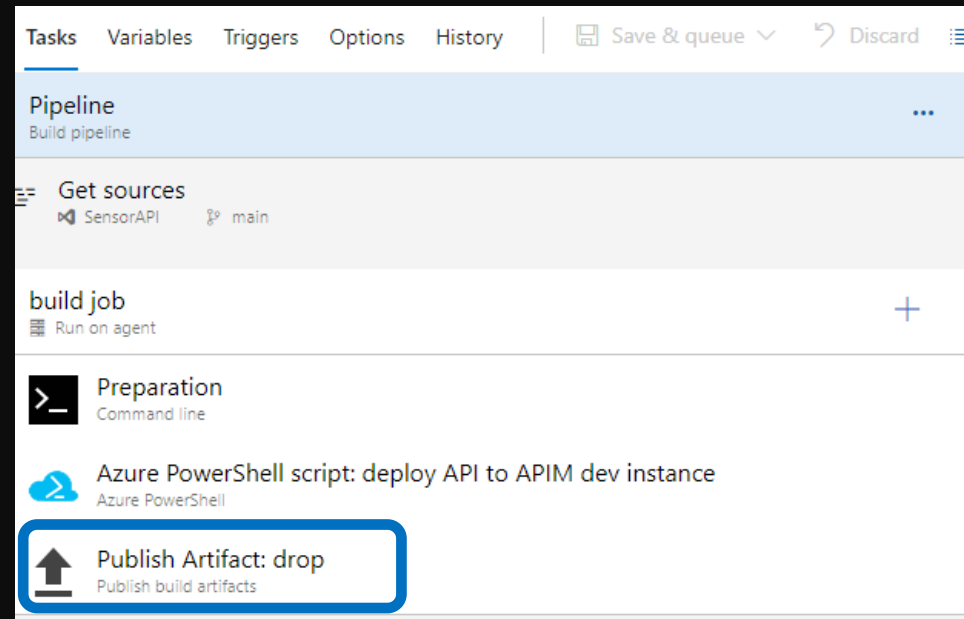
- API Dev sources maintained in API Provider owned repos
- Central build pipeline calls partial build pipelines in each registered repo (for „last stable“)
- Central build pipeline collects and labels build artefacts in centrals storage (no central repo for API Dev)
- Central build/release pipeline deploys to APIM

[Sensor/Invoice/any] API team

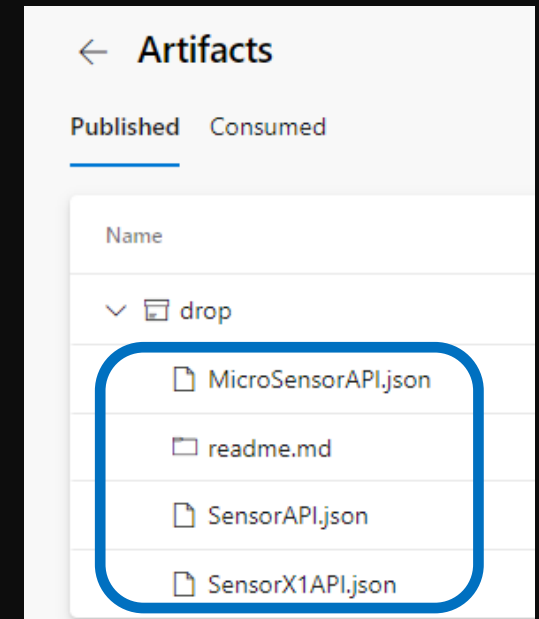
Develop



Build

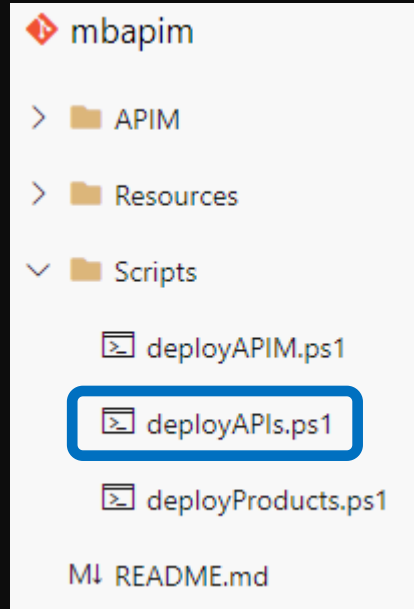


Publish



APIM operators team

Develop

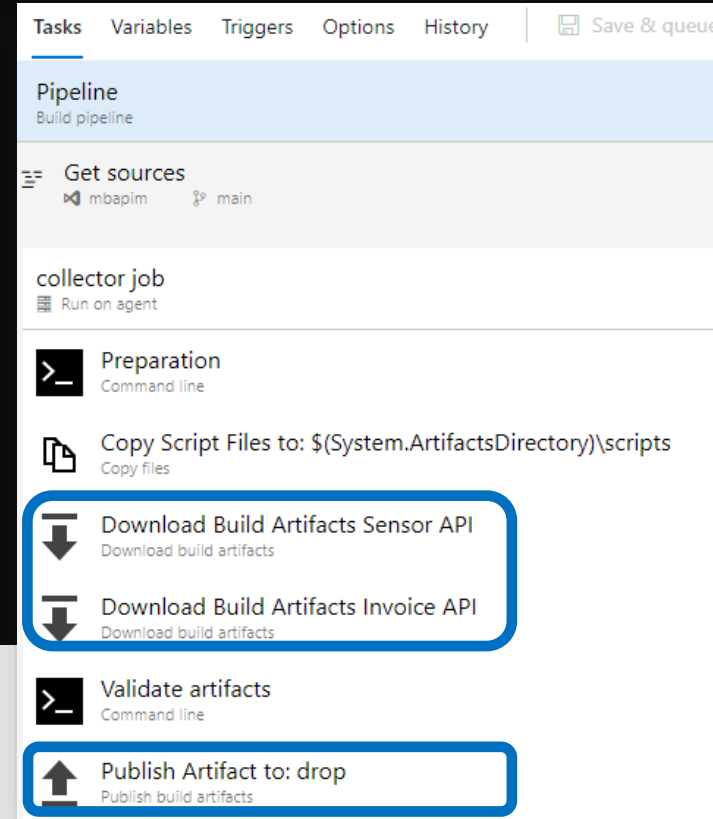


```
Write-Host "****"
Write-Host "Deploy APIs to APIM"
Write-Host "****"
$workdir = pwd
Write-Host "Executing in '$workdir'"
Import-Module -Name Az -Force
Write-Host "Get APIM context"
$ApiMgmtContext = New-AzApiManagementContext -ResourceGroupName "mbplayground" -ServiceName "mbapim"
Write-Host "Import InvoiceAPI"
Import-AzApiManagementApi -Context $ApiMgmtContext -SpecificationFormat "OpenApi" -
SpecificationPath "drop\InvoiceAPI\drop\InvoiceAPI.json" -Path "InvoiceAPI" -ApiId "invoice"
Write-Host "Import SensorAPI"
Import-AzApiManagementApi -Context $ApiMgmtContext -SpecificationFormat "OpenApi" -
SpecificationPath "drop\SensorAPI\drop\SensorAPI.json" -Path "SensorAPI" -ApiId "sensor"

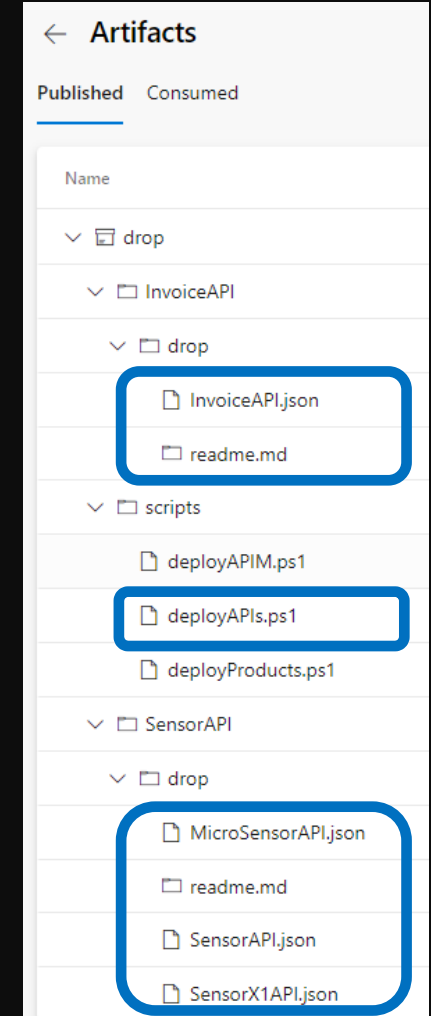
Import-AzApiManagementApi -Context $ApiMgmtContext -SpecificationFormat "OpenApi" -
SpecificationPath "drop\SensorAPI\drop\MicroSensorAPI.json" -Path "MicroSensorAPI" -ApiId "microsensor"

Import-AzApiManagementApi -Context $ApiMgmtContext -SpecificationFormat "OpenApi" -
SpecificationPath "drop\SensorAPI\drop\SensorX1API.json" -Path "SensorX1API" -ApiId "sensorx1"
```

Collect and verify

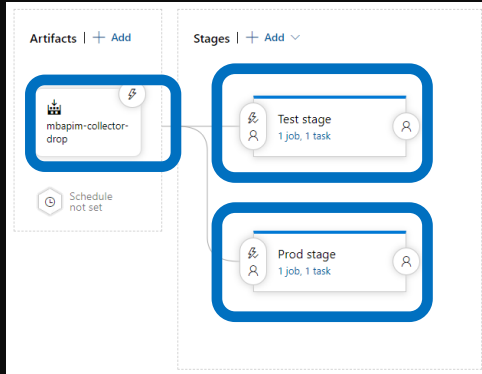


Publish



APIM operators team

Plan



The configuration view for the 'Test stage' (Deployment process) shows the 'deployment job' task. The task is an 'Azure PowerShell script: de...' using the 'Azure PowerShell' provider. The configuration details on the right are:

- Display name: Azure PowerShell script: deploy APIs
- Azure Subscription: mbplayground
- Script Type: Script File Path (selected)
- Script Path: `$(System.DefaultWorkingDirectory)/mbapim-collector-drop/drop/scripts/deployAPIs.ps1`

Deploy

deployment job

Pool: Azure Pipelines · Agent: Hosted Agent

- ✓ Initialize job · succeeded
- ✓ Download artifact - mbapim-collector-drop - drop · succeeded
- ✓ Azure PowerShell script: deploy APIs · succeeded
- ✓ Finalize Job · succeeded