Swag and more

- Claim your attendee Learner Badge here:

- 30 Days to learn it: aka.ms/global-azure/30D2L

- Virtual background and ANOTHER Badge: blog.globalazure.net/Swag

BY COMMUNITY · FOR COMMUNITY ·
#GLOBALAZURE
2021

SQL Player
Play with data & have fun!

Azure Data Factory Deployments

Microsoft®
MVP
Most Valuable Professional

Kamil Nowiński

Microsoft Data Platform **MVP**
Speaker, blogger, data enthusiast

Group Manager at Avanade UK&I (www.avanade.com)
Almost 20 yrs experience as DEV/BI/(DBA)

Member of the Data Community PL

Project member of „SCD Merge Wizard"

Founder of blog SQLPlayer (www.SQLplayer.net)

SQL Server Certificates:
MCITP, MCP, MCTS, MCSA, MCSE Data Platform,

MCSE Data Management & Analytics

Moreover: Bicycle, Running, Digital photography
@NowinskiK, @SQLPlayer

Blog

SQL Player
Play with data & have fun!

www.SQLPlayer.net

- Technical posts
- Various skill level
- Cheet sheets
- Recommended books
- Many useful other links
- Interviews (Podcast)
- YouTube Channel:
  www.SQLPlayer.net/YouTube

Scan me

“Ask SQL Family” #podcast

Scan me

Slides available

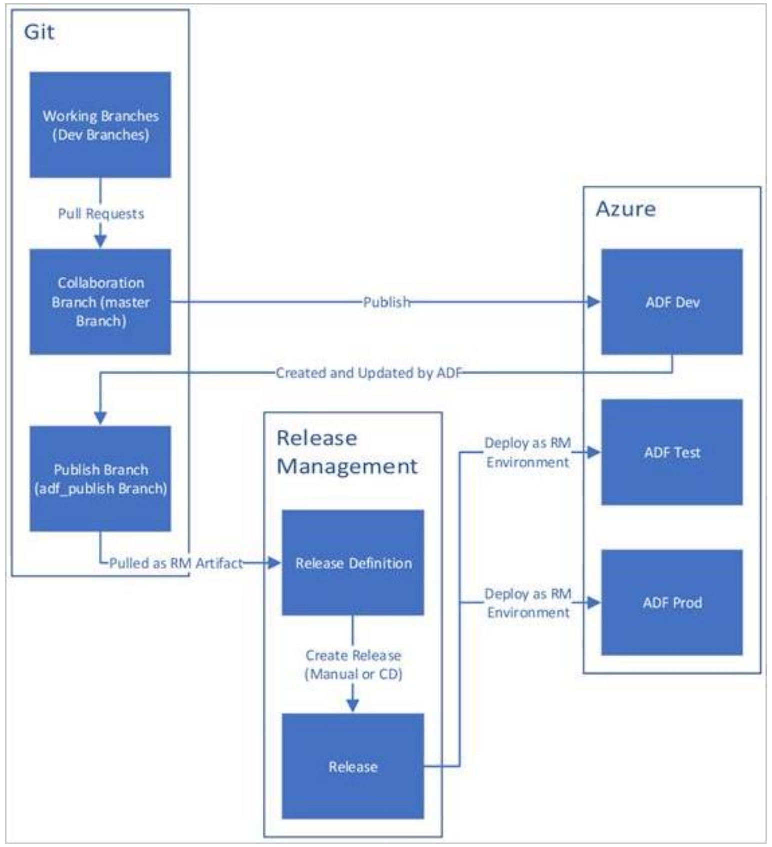
github.com
/NowinskiK/CommunityEvents

AGENDA

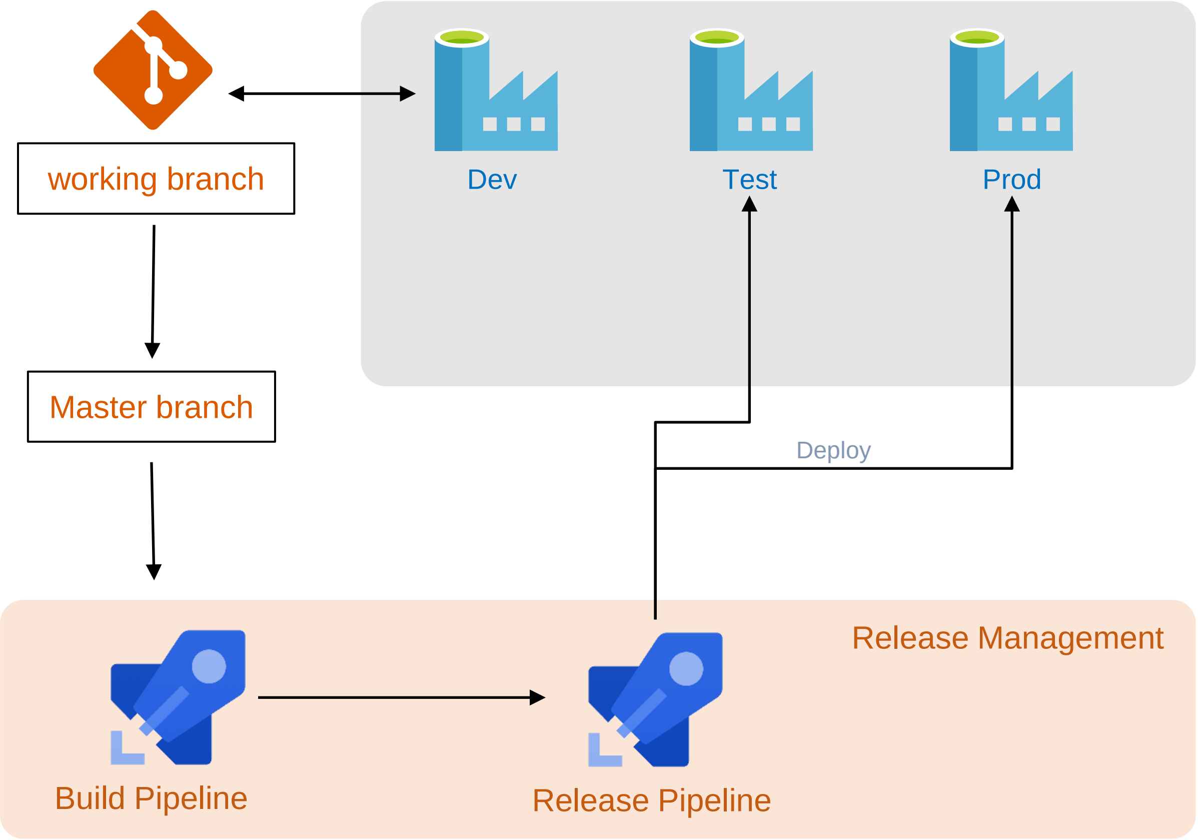
- Azure Data Factory – DEPLOYMENT only
- Two (three?) methods of ADF deployment
- How these methods work
- Differences
- npm module from Microsoft – now you can fully automate CI (build)
- #adftools – make your life easier!

ADF – Currently available methods of deployment

ARM Template from “adf_publish” branch



Rest-Api/PowerShell script from code (JSON objects)



working branch

Dev

Test

Prod

Master branch

Deploy

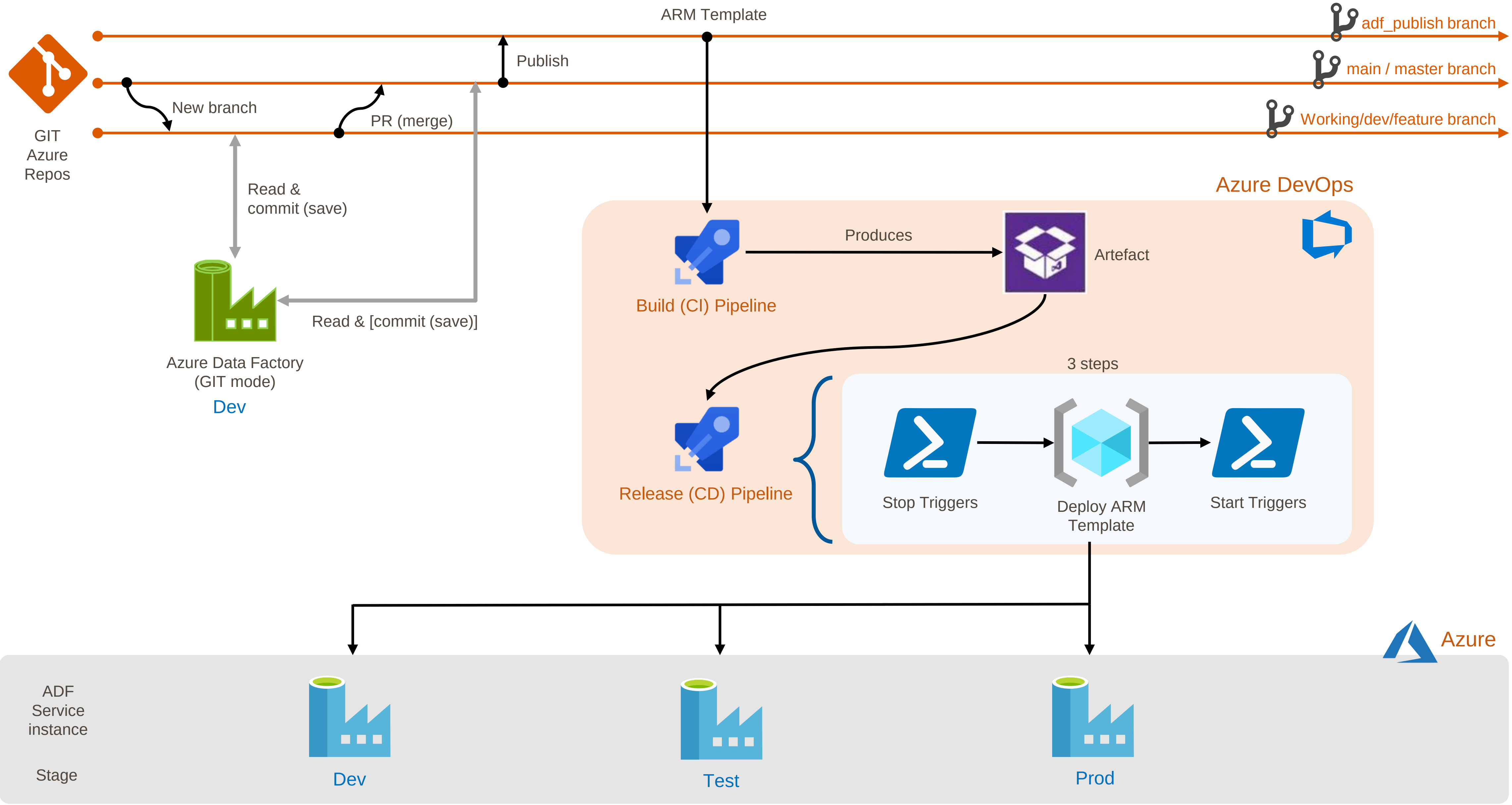
Release Management

Build Pipeline

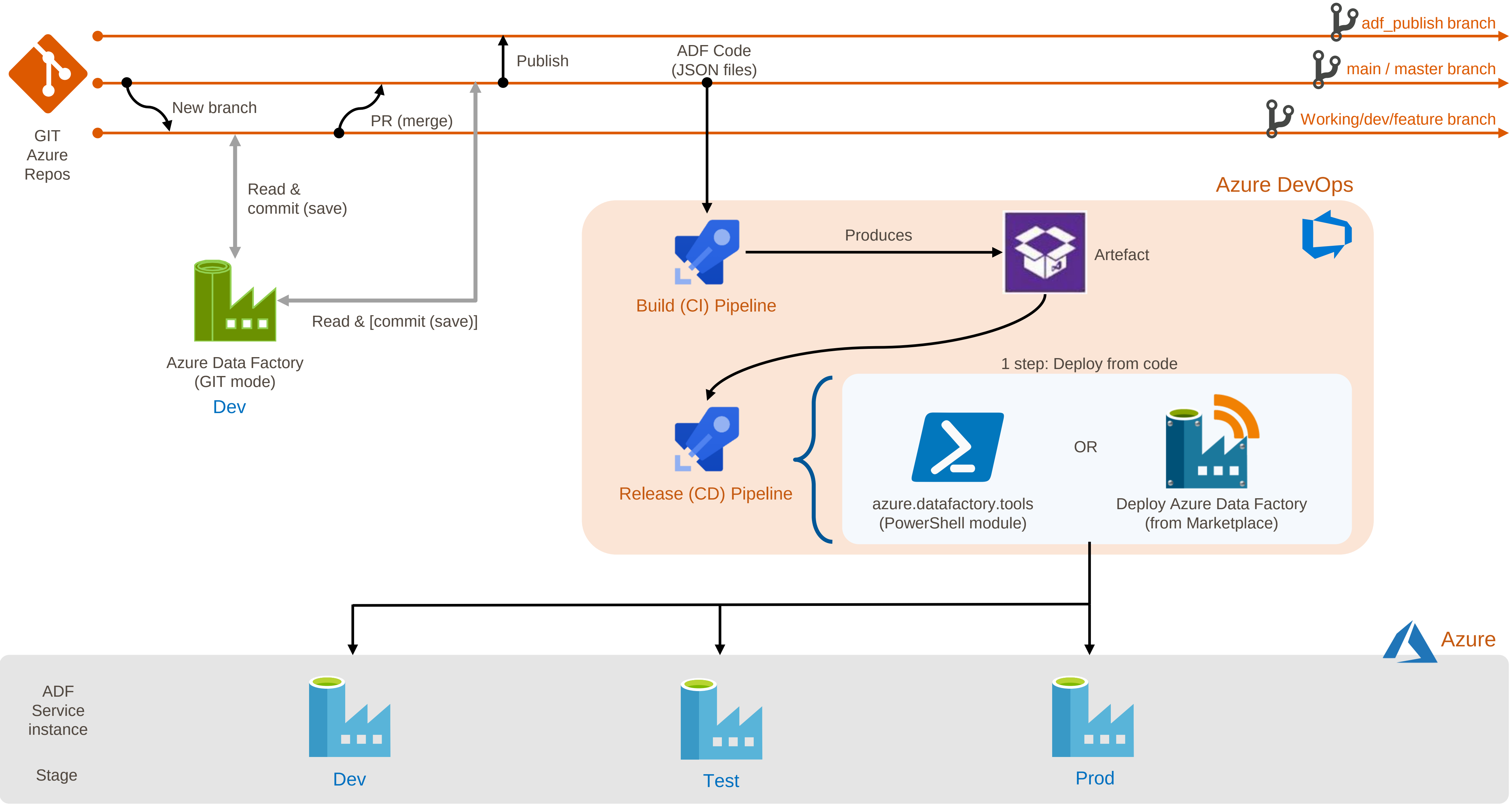
Release Pipeline

https://docs.microsoft.com/en-us/azure/data-factory/continuous-integration-deployment

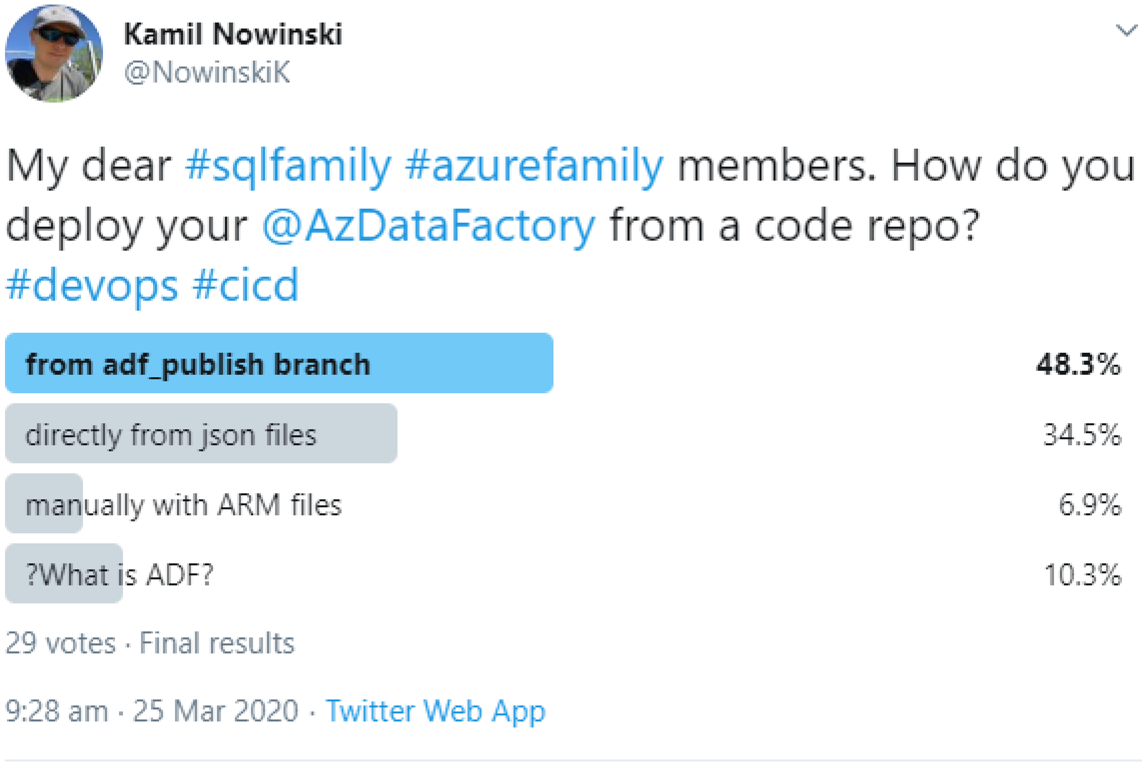
Deployment (1): Microsoft's method (ARM Template)



Deployment (2): Directly from code method



Who use what?



Kamil Nowinski
@NowinskiK

My dear #sqlfamily #azurefamily members. How do you deploy your @AzDataFactory from a code repo? #devops #cicd

from adf_publish branch	**48.3%**
directly from json files	34.5%
manually with ARM files	6.9%
?What is ADF?	10.3%

29 votes · Final results

9:28 am · 25 Mar 2020 · Twitter Web App

https://twitter.com/NowinskiK/status/1242745169394442240

ADF – Currently available methods of deployment – pros & cons

ARM Template from "adf_publish" branch

- Faster
- Appears in „Deployments" for Resource Group

- Parametrize elements exposed within the ARM Template Parameter *
- Full ADF (all artefacts) can be deploy only
- Restriction of 256 parameters
- Limitation to one publish branch only (adf_publish)
- Manual "Publish" step

Rest-Api/PowerShell script from code (JSON objects)

- Slower
- Doesn't appear in „Deployments" for Resource Group

- Parameterize any artefact of the Data Factory
- Selectively deploy a subset of artefacts
- Eliminates an enforcement to use only one (adf_publish) branch if company's branches policy is much complex

Post: Two methods of deployment Azure Data Factory

ADF – ARM Template enhanced deployment method

ARM Template from “adf_publish” branch

- Faster
- Appears in „Deployments” for Resource Group

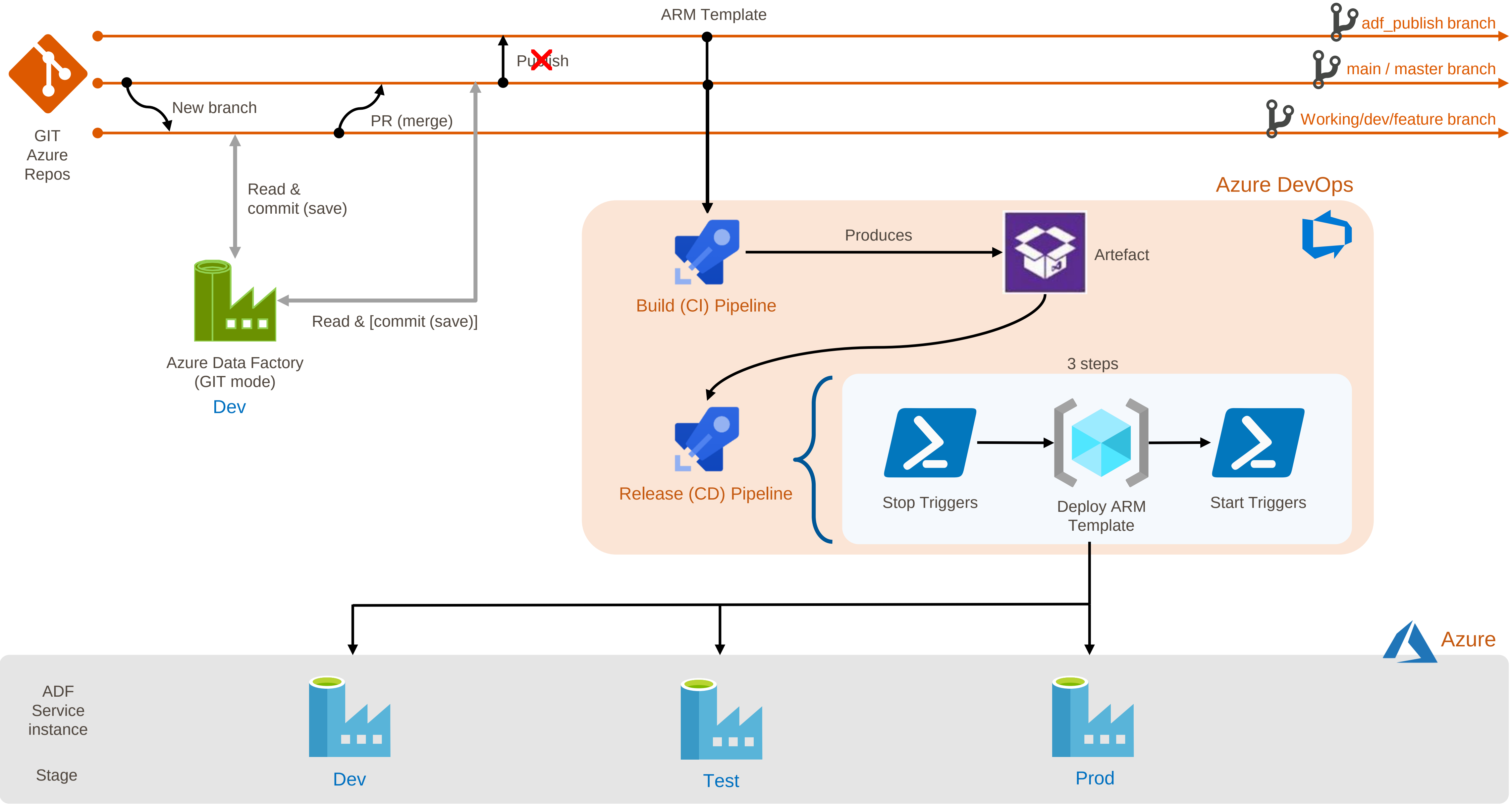
- Parametrize elements exposed within the ARM Template Parameter *
- Full ADF (all artefacts) can be deploy only
- Restriction of 256 parameters
- ~~Limitation to one publish branch only (adf_publish)~~
- ~~Manual “Publish” step~~

Rest-Api/PowerShell script from code (JSON objects)

- Slower
- Doesn’t appear in „Deployments” for Resource Group

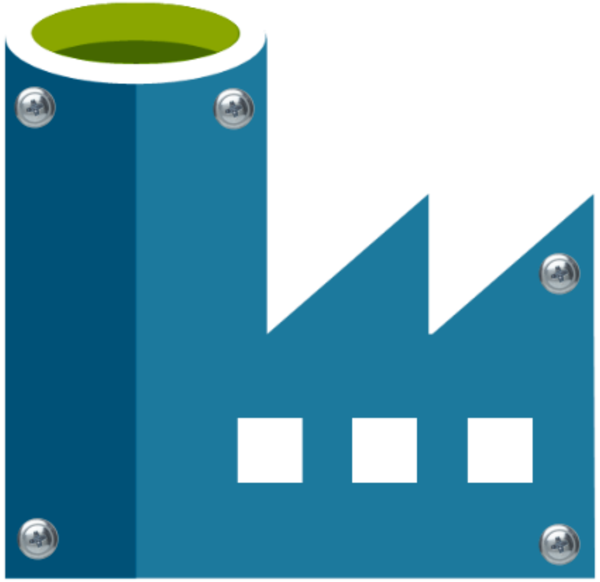
- Parameterize any artefact of the Data Factory
- Selectively deploy a subset of artefacts
- Eliminates an enforcement to use only one (adf_publish) branch if company’s branches policy is much complex

Deployment (1b): ARM Template exported by CI



DEMO #1
ARM Template deployment

#adftools



#adftools

SQL Player
Play with data & have fun!

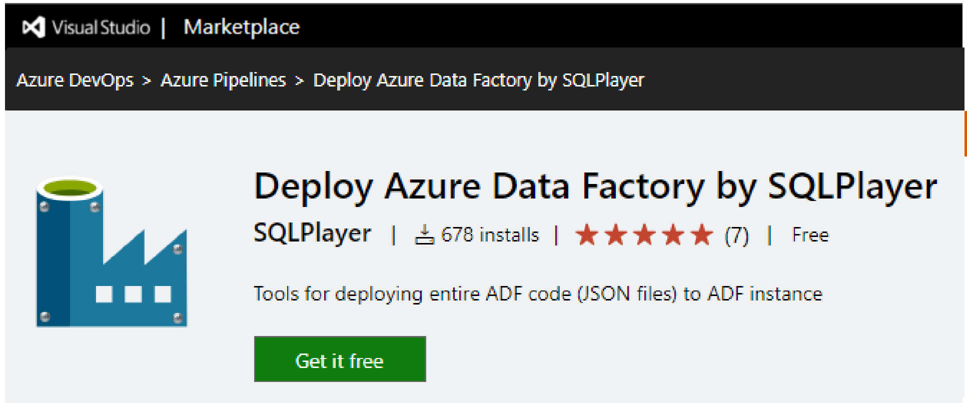
Two tools:

- PowerShell module (azure.datafactory.tools)
- Azure DevOps extension (3 tasks)

https://sqlplayer.net/adftools/

DEMO #2
Deployment with PowerShell

Azure DevOps: Custom Tasks


Visual Studio | Marketplace
Azure DevOps > Azure Pipelines > Deploy Azure Data Factory by SQLPlayer

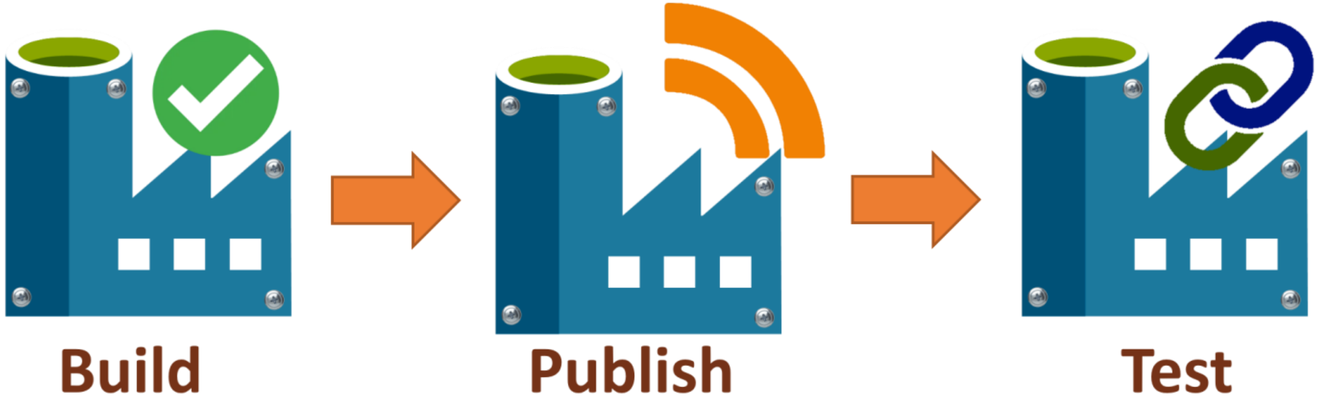
Deploy Azure Data Factory by SQLPlayer

SQLPlayer | 678 installs | ★★★★★ (7) | Free

Tools for deploying entire ADF code (JSON files) to ADF instance

Key concept:

- Free & Open-Source
- One task for everything when it comes to the publishing of ADF
- Basically, it is another (UI) layer on top of „azure.datafactory.tools" module
- Public Release (GA) since 23/12/2020
- Extension contains 3 tasks that cover full deployment life-cycle for ADF



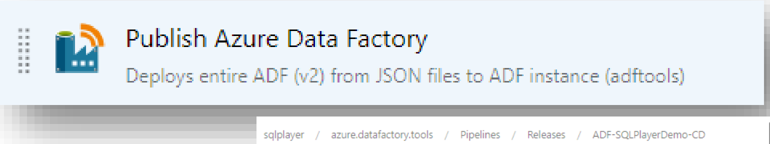
Any ideas or questions? Leave it here:
https://github.com/SQLPlayer/azure.datafactory.tools/issues

PowerShell module: azure.datafactory.tools

- Fully written in PowerShell, compatible with 5.1
- Uses Microsoft's PS module (Az.DataFactory) for management of ADF objects
- Available in PowerShell gallery
- `Publish-AdfV2FromJson` function - capabilities:
  - Creation of Azure Data Factory, if not exist (option)
  - Deployment of all type of objects: pipelines, datasets, linked services, data flows & power query, triggers, integration runtimes
  - Copes with dependencies (multiple levels) between objects when deploying (no more worrying about object names)
  - Build-in mechanism to replace the properties with the indicated values (CSV file)
  - Stop/start triggers (option)
  - Dropping objects when not exist in the source (code) (option)
  - Selective deployment: Filtering (include or exclude) objects to be deployed by name and/or type
  - Publish options allow you to control:
    - Whether stop and restarting triggers
    - Whether delete or not objects not in the source
    - Whether create or not a new instance of ADF if it not exist
- Free & Open-Source

https://github.com/SQLPlayer/azure.datafactory.tools

Task in Azure DevOps: Publish ADF (CD)

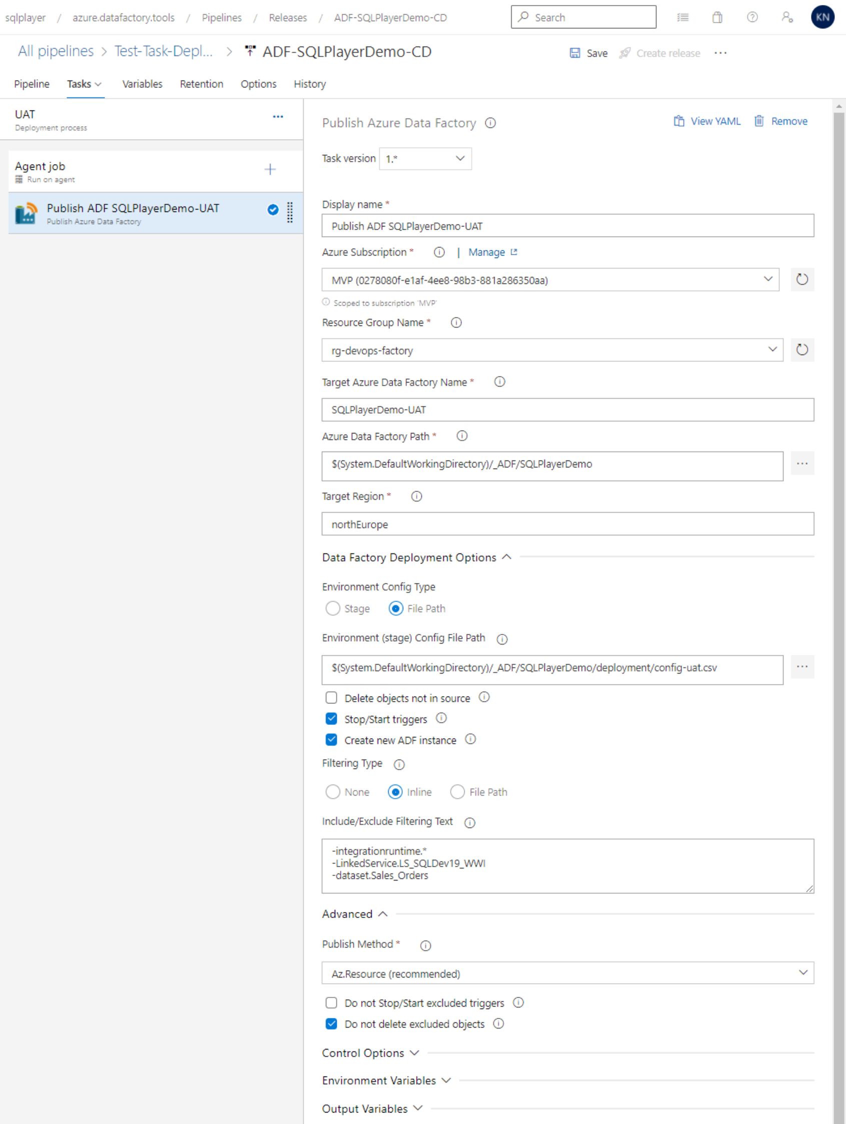


Publish Azure Data Factory

Deploys entire ADF (v2) from JSON files to ADF instance (adftools)

Key capabilities:

- Creation of Azure Data Factory, if not exist (option)
- Deployment of all type of objects: pipelines, datasets, linked services, data flows, triggers, integration runtimes
- Copes with dependencies (multiple levels) between objects when deploying (no more worrying about object names)
- Build-in mechanism to replace the properties with the indicated values (CSV file)
- Update, add or remove any property of ADF artefact
- Selective deployment declared in-line or by pointed file
- Stop/start triggers (option)
- Dropping objects when not exist in the source (code) (option)
- Filtering (include or exclude) objects to be deployed by name and/or type and/or type
- Filtering supports wildcards
- Publish options allow you to control:
  - Whether stop and restarting triggers
  - Whether delete or not objects not in the source
  - Whether create or not a new instance of ADF if it not exist
- Tokenisation in config file allows replace any value by Environment Variable or Variable from DevOps Pipeline
- Global Parameters



sqlplayer / azure.datafactory.tools / Pipelines / Releases / ADF-SQLPlayerDemo-CD
Search
KN
All pipelines > Test-Task-Depl... > ADF-SQLPlayerDemo-CD
Save
Create release
Pipeline
Tasks
Variables
Retention
Options
History
UAT
Deployment process
Agent job
Run on agent
Publish ADF SQLPlayerDemo-UAT
Publish Azure Data Factory
View YAML
Remove

Publish Azure Data Factory

Task version: 1.*

Display name *: Publish ADF SQLPlayerDemo-UAT

Azure Subscription * | Manage: MVP (0278080f-e1af-4ee8-98b3-881a286350aa)

Scoped to subscription 'MVP'

Resource Group Name *: rg-devops-factory

Target Azure Data Factory Name *: SQLPlayerDemo-UAT

Azure Data Factory Path *: $(System.DefaultWorkingDirectory)/_ADF/SQLPlayerDemo

Target Region *: northEurope

Data Factory Deployment Options

Environment Config Type: ○ Stage ● File Path

Environment (stage) Config File Path: $(System.DefaultWorkingDirectory)/_ADF/SQLPlayerDemo/deployment/config-uat.csv

☐ Delete objects not in source

☑ Stop/Start triggers

☑ Create new ADF instance

Filtering Type: ○ None ● Inline ○ File Path

Include/Exclude Filtering Text:

```
-integrationruntime.*
-LinkedService.LS_SQLDev19_WWI
-dataset.Sales_Orders
```

Advanced

Publish Method *: Az.Resource (recommended)

☐ Do not Stop/Start excluded triggers

☑ Do not delete excluded objects

Control Options

Environment Variables

Output Variables

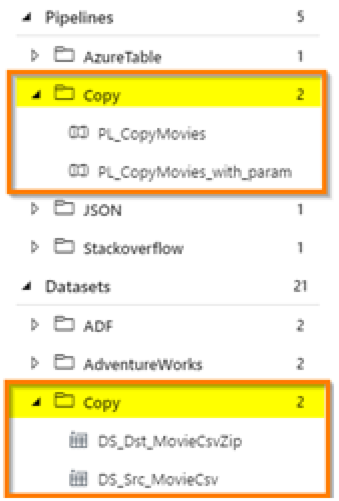
Selective deployment with PowerShell module

You can select objects by objects types & name using include or exclude option.

Allows to select the objects by belonging to a folder (picture)

Name can be **wildcarded**, so all such variants are possible:

```
trigger.*
dataset.DS_*
*.PL_*
linkedService.???KeyVault*
pipeline.ScdType[123]
*.*@testFolder
```



```
◢ Pipelines                          5
  ▷ 🗀 AzureTable                    1
  ◢ 🗀 Copy                          2
        PL_CopyMovies
        PL_CopyMovies_with_param
  ▷ 🗀 JSON                          1
  ▷ 🗀 Stackoverflow                 1
◢ Datasets                          21
  ▷ 🗀 ADF                           2
  ▷ 🗀 AdventureWorks                2
  ◢ 🗀 Copy                          2
        DS_Dst_MovieCsvZip
        DS_Src_MovieCsv
```

Selective deployment in Azure DevOps

☺ In Azure DevOps Task – the list can be provided either as (inline) **text** or from **file** in repo.

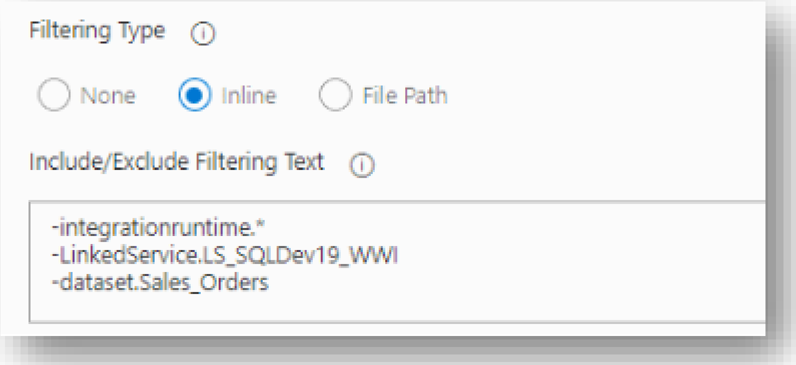
To simplify user experience – only one field is exposed in order to define include/exclude rules.

Therefore, an extra character should be provided before the name/pattern:

+ (plus) – for objects you want to include to a deployment

- (minus) - for objects you want to exclude from a deployment

If char is not provided – an inclusion rule would be applied.



Filtering Type ⓘ

○ None ● Inline ○ File Path

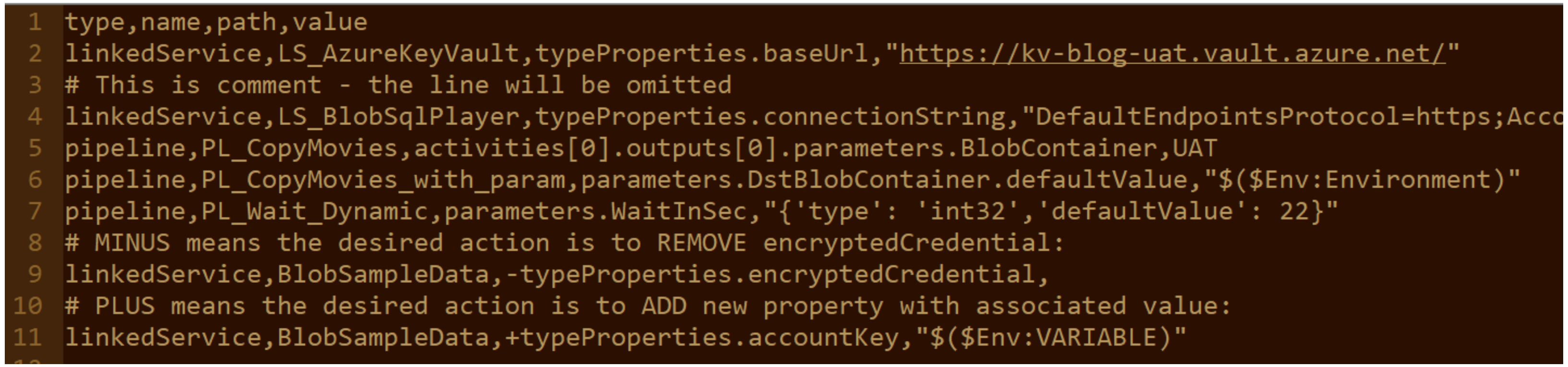
Include/Exclude Filtering Text ⓘ

```
-integrationruntime.*
-LinkedService.LS_SQLDev19_WWI
-dataset.Sales_Orders
```

DEMO #3
Deployment with Azure DevOps

PowerShell module: Parameters for Stages

How are parameters passed into the deployment? Config CSV File:



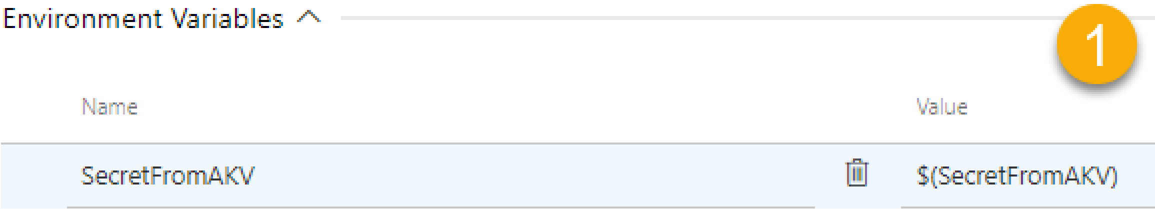
```
type,name,path,value
linkedService,LS_AzureKeyVault,typeProperties.baseUrl,"https://kv-blog-uat.vault.azure.net/"
# This is comment - the line will be omitted
linkedService,LS_BlobSqlPlayer,typeProperties.connectionString,"DefaultEndpointsProtocol=https;Acc
pipeline,PL_CopyMovies,activities[0].outputs[0].parameters.BlobContainer,UAT
pipeline,PL_CopyMovies_with_param,parameters.DstBlobContainer.defaultValue,"$($Env:Environment)"
pipeline,PL_Wait_Dynamic,parameters.WaitInSec,"{'type': 'int32','defaultValue': 22}"
# MINUS means the desired action is to REMOVE encryptedCredential:
linkedService,BlobSampleData,-typeProperties.encryptedCredential,
# PLUS means the desired action is to ADD new property with associated value:
linkedService,BlobSampleData,+typeProperties.accountKey,"$($Env:VARIABLE)"
```

Option 1 (NEW! and supported now):

Any variables can come from DevOps Pipeline, either normal as well as sensitive values.

To apply replacement for secret values:

Environment Variables must be mapped (Microsoft recommendation).

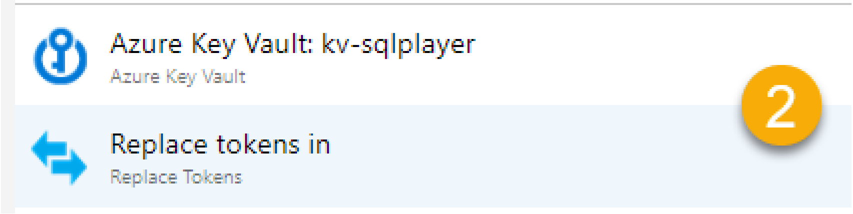


Name	Value
SecretFromAKV	$(SecretFromAKV)

Option 2:

Another option would be reading secrets directly from provided Azure Key Vault.

Therefore, “Replacement” task is still recommended here as an alternative as for now.

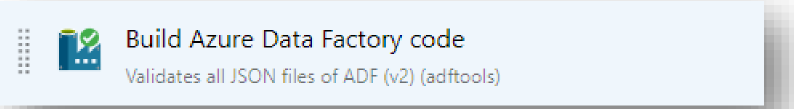


New task in Azure DevOps: Build ADF (CI)

The task has 2 modes:

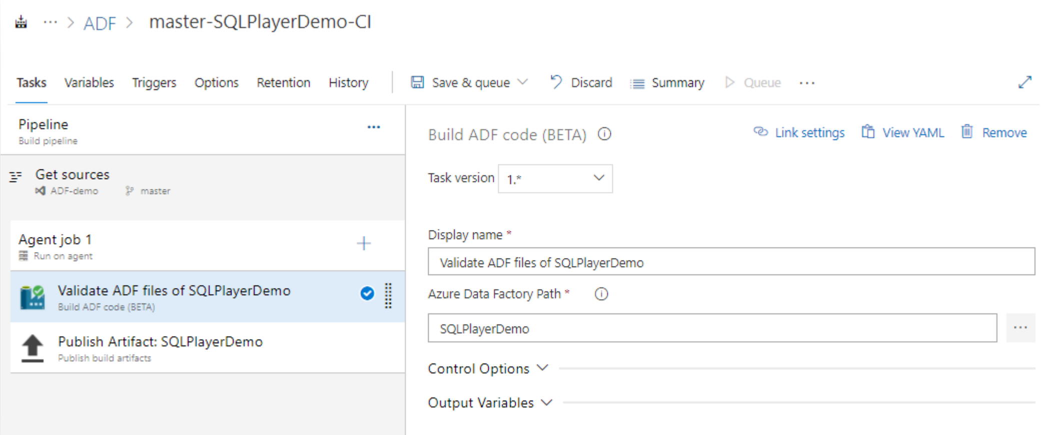
- **Build only**
  - Reads all files and validates its json format
  - Checks whether all dependant objects exist
  - Checks whether file name equals object name

- **Validate & Export ARM Template**
  - uses ADFUtilities NPM package provided by Microsoft
  - Counterpart of Validate all and Export ARM Template in ADF UI



Build Azure Data Factory code

Validates all JSON files of ADF (v2) (adftools)



··· > ADF > master-SQLPlayerDemo-CI

Pipeline
Build pipeline

Get sources
ADF-demo master

Agent job 1
Run on agent

Validate ADF files of SQLPlayerDemo
Build ADF code (BETA)

Publish Artifact: SQLPlayerDemo
Publish build artifacts

Build ADF code (BETA)

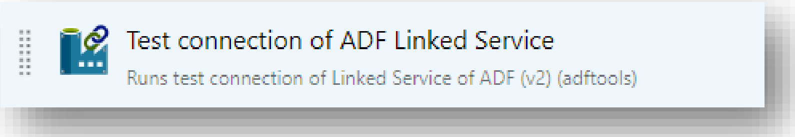
Task version: 1.*

Display name *: Validate ADF files of SQLPlayerDemo

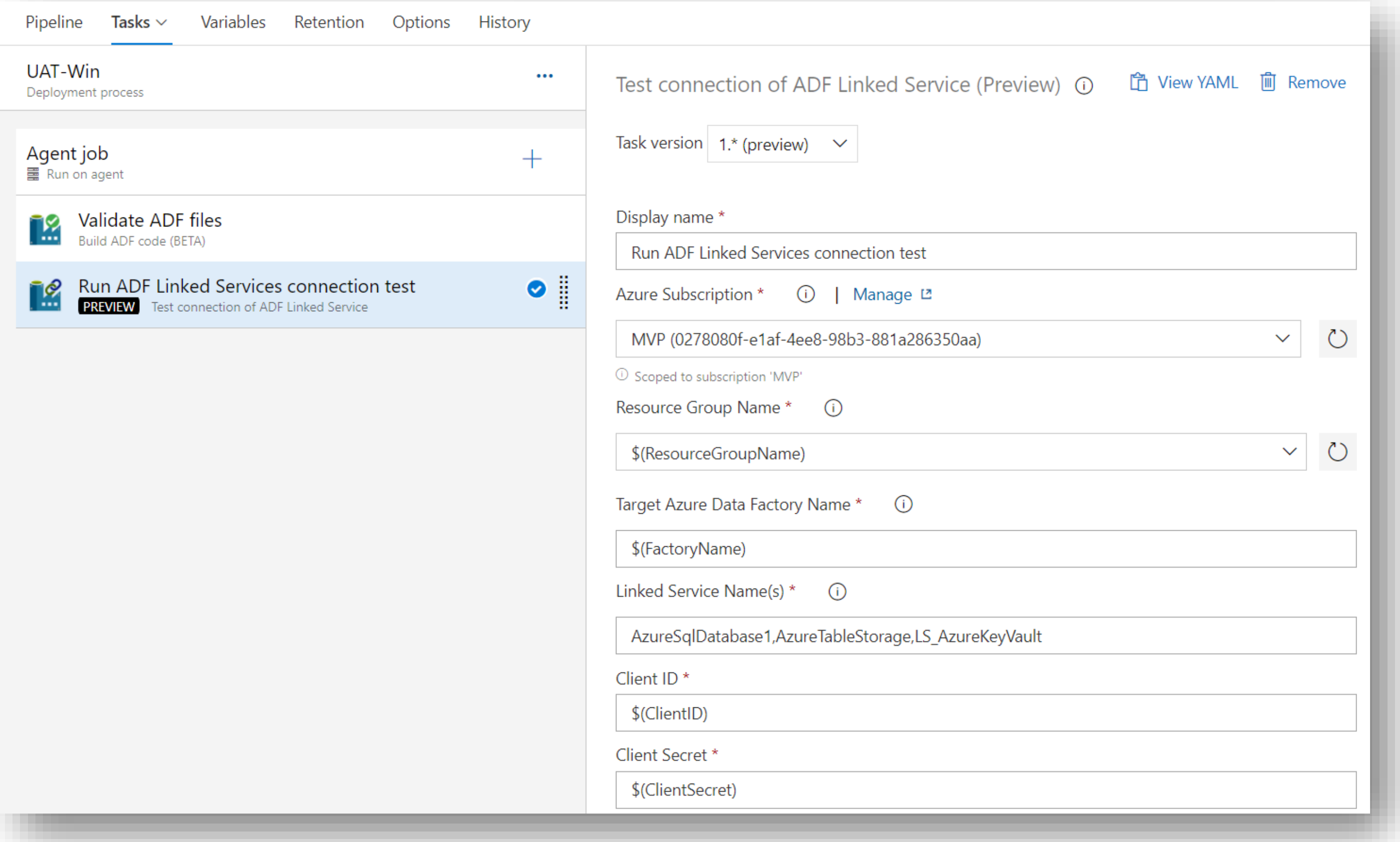
Azure Data Factory Path *: SQLPlayerDemo

New task in Azure DevOps: Test Connection

- Run „Test connection" for a Linked Service
- Smoke tests of (some) Linked Services
- In preview



Test connection of ADF Linked Service
Runs test connection of Linked Service of ADF (v2) (adftools)



UAT-Win
Deployment process

Agent job
Run on agent

Validate ADF files
Build ADF code (BETA)

Run ADF Linked Services connection test
PREVIEW Test connection of ADF Linked Service

Test connection of ADF Linked Service (Preview)

Task version: 1.* (preview)

Display name *: Run ADF Linked Services connection test

Azure Subscription * | Manage: MVP (0278080f-e1af-4ee8-98b3-881a286350aa)

Scoped to subscription 'MVP'

Resource Group Name *: $(ResourceGroupName)

Target Azure Data Factory Name *: $(FactoryName)

Linked Service Name(s) *: AzureSqlDatabase1,AzureTableStorage,LS_AzureKeyVault

Client ID *: $(ClientID)

Client Secret *: $(ClientSecret)

Questions?

Thank you!

kamil@nowinski.net

@NowinskiK @SQLPlayer

SQLPlayer.net

https://github.com/NowinskiK/CommunityEvents

Kamil Nowinski
Microsoft Data Platform MVP
MCSE Data Platform & MCSE Data Management and Analytics