

[#Luedinghausen](#)

Application Lifecycle Management in Azure Data Factory

Stefan Kirner



@KirnerKa



[#GlobalAzure](#)

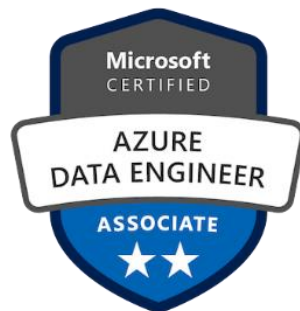
“Instead of wondering when your next vacation is, maybe you should set up a life you don't need to escape from.” — Seth Godin

Who is talking?

Stefan Kirner



- › PASS Chapter Lead Karlsruhe
- › Co-Founder scieneers GmbH
- › DRIVEN BY DATA since 2002
- › Twitter: @KirnerKa



Topics

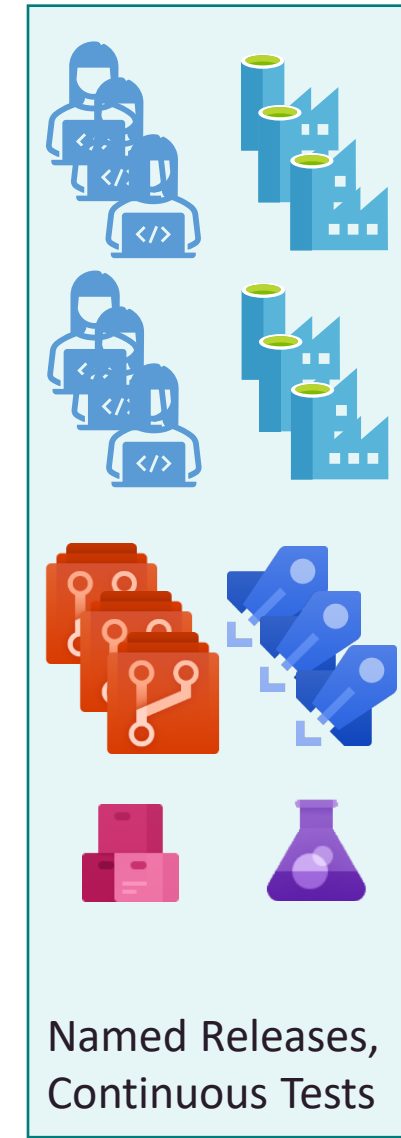
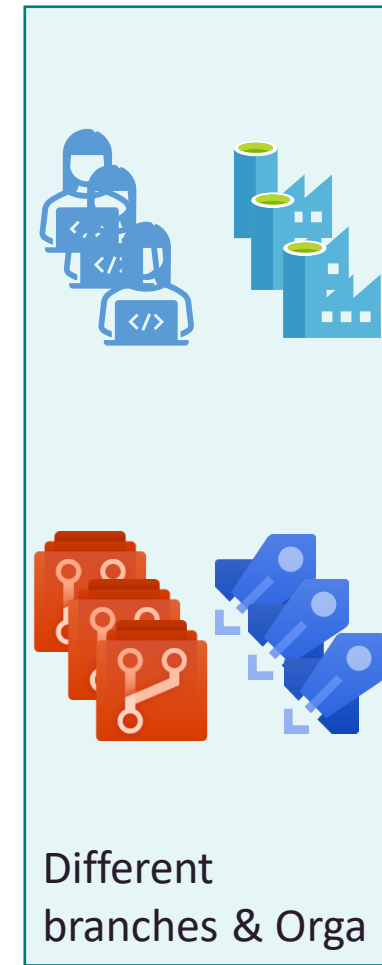
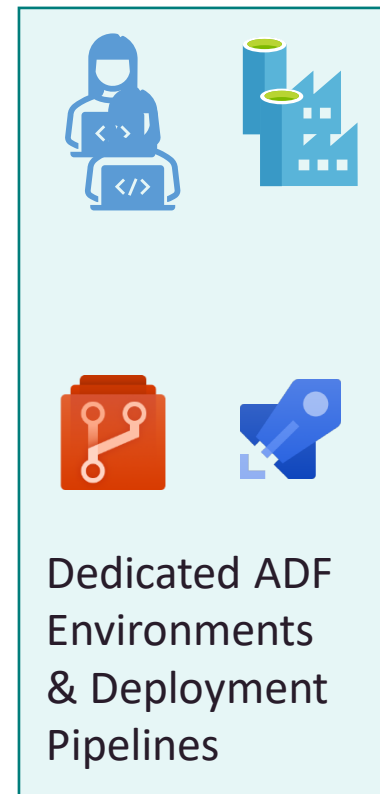
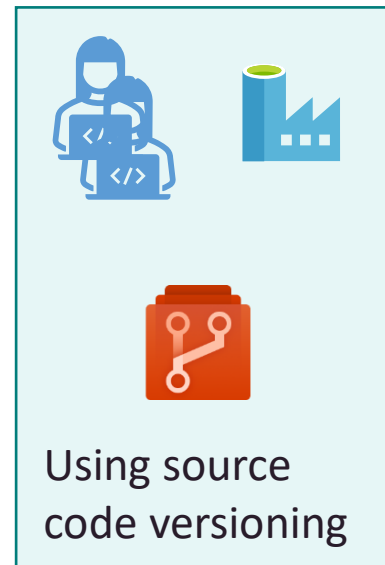
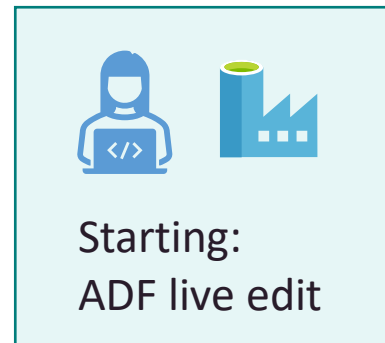
- Kick-off
- Source Code Integration
- ADF Config best practices
- Azure Pipelines
- Teamwork
- ADF vs. Synapse
- Close

Why DevOps for ADF?

- Lessons learned in (classic) software development:
- Transparent code base (track changes)
- Safe development (revert changes)
- Stable production environment (CI/CD)
- Avoid boring work (automation)
- Enable teamwork (collaboration)
- Better ADF dev. performance (10x)



Different states of ADF development



(Most) Useful things in Azure DevOps for Data Factory?

Azure DevOps

Boards

Repos

Pipelines

Test Plans

Artifacts

sceDevOpsDemos

azure_data_factory_v2

dataset

factory

linkedService

AzureDataLake.json

AzureKeyVault.json

AzureSqlDatabaseWWWImporters.json

pipeline

trigger

working / Type to find a file or folder...

Files

Contents History

Graph

Commit

Updating dataset: ds_products_csv2parquet
1d78f69c Stefan Kirner Mar 22 at 1:24 PM

Renaming dataset: pl_products_csv2parquet as ds_products_csv2parquet
5a4ffe2f Stefan Kirner Mar 22 at 1:24 PM

Updating pipeline: pl_transform_products
603c4364 Stefan Kirner Mar 22 at 1:24 PM

Updating dataset: ds_raw_products
a8a53bct Stefan Kirner Mar 22 at 1:23 PM

Renaming dataset: Dataset as ds_raw_products
153b0342 Stefan Kirner Mar 22 at 1:23 PM

Adding pipeline: pl_transform_products
11414381 Stefan Kirner Mar 22 at 1:22 PM

working / Type to find a file or folder...

Filter branches

Branches Tags

working Default

Mine

adf_publish

dev

workspace_publish

+ New branch

Pipelines

Recent All Runs

Recently run pipelines

Pipeline

Last run

deploy_TEST_ALL #20210311.3 • Updated synapse-db-objects.sql
Manually triggered for working

deploy_PROD_ALL #20210226.5 • Updated deploy_prod_all.yml
Manually triggered for working

Jobs in run #20210311.3

deploy_TEST_ALL

Test

Deploy Azure Data Platform 2m 56s

Initialize job 8s

Pre-job: Download secrets: sc... <1s

Download secrets: sce-key-vaul... 1s

Checkout sceDevOpsDemos@a... 7s

Checkout sceDevOpsDemos@... 2s

Checkout sceDevOpsDemos@... 2s

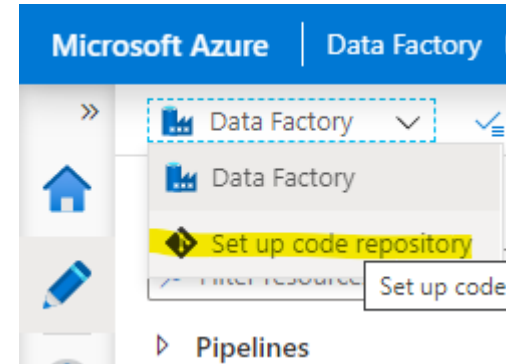
Deploy Azure Data Platform

1 Pool: Azure Pipelines
2 Image: vs2017-win2016
3 Agent: Hosted Agent
4 Started: Mar 11 at 10:27 AM
5 Duration: 2m 56s

Setup Source Code Integration

Choose Git server:

- Azure DevOps Git.
Requirements: Azure AD, Azure DevOps Account & Project
- Public GitHub or GitHub Enterprise: public/private repos supported.
Administrator permissions for Azure subscription.



Configure a repository

Specify the settings that you want to use when connecting to your repository.

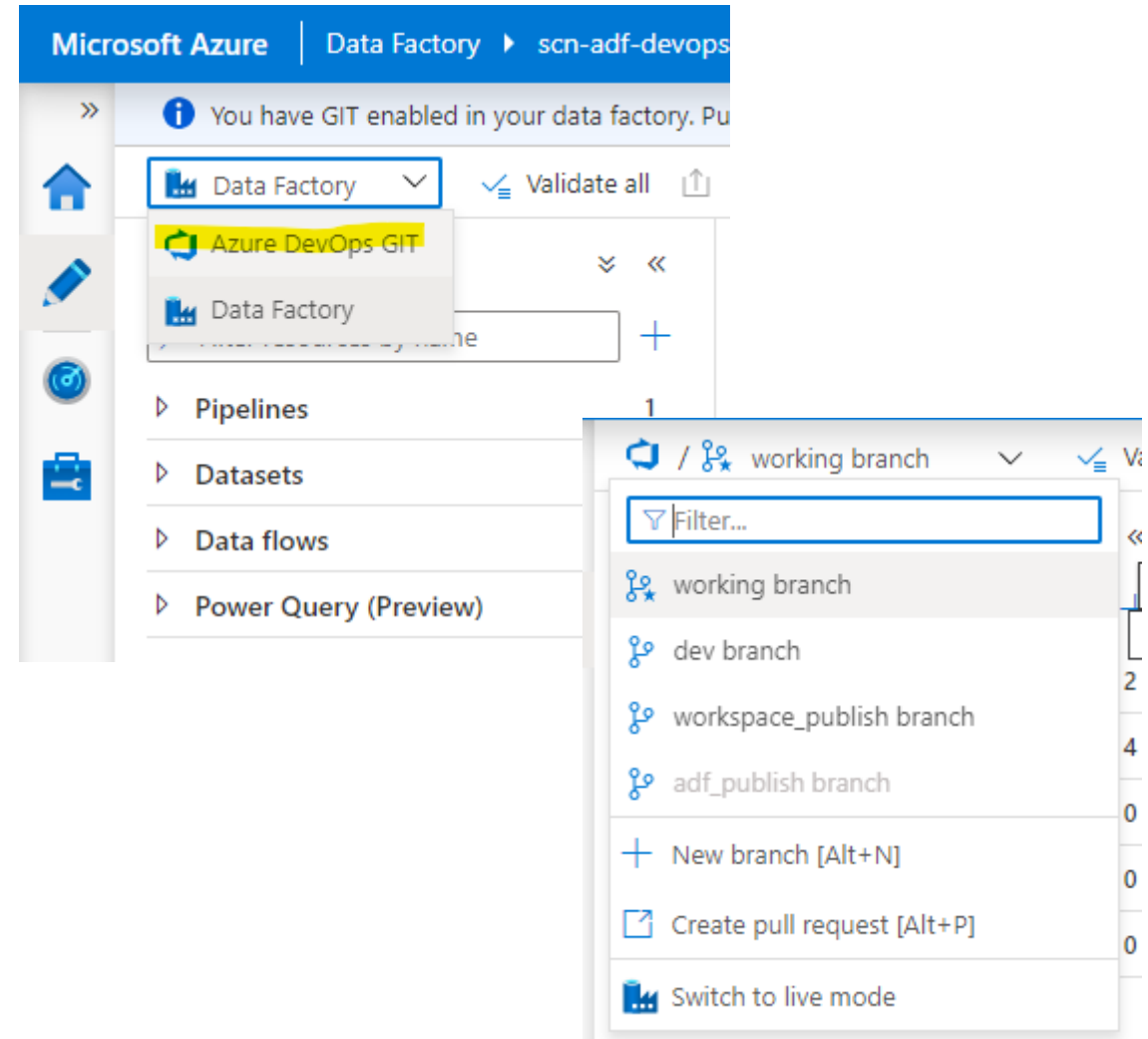
Repository type * ⓘ

Select...

-  Azure DevOps Git
-  GitHub

Source Code Integration

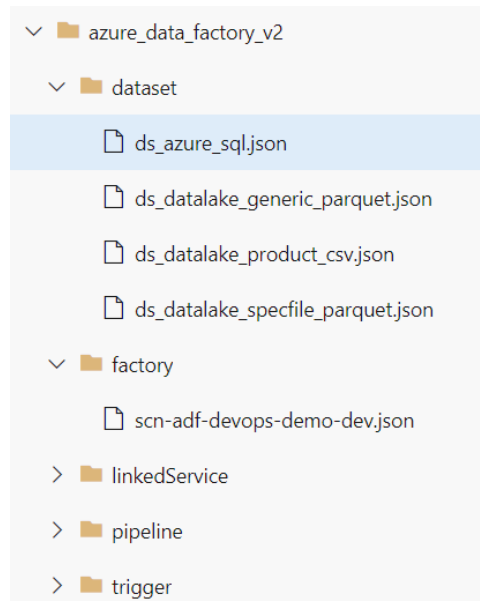
- IDE for Azure Data Factory browser based in Azure Portal
- Direct Source Code integration optional
- Switch between live mode and source code
- Different development branches ... and adf_publish



Setup Source Code Integration

Branch which is used for publishing
Contains development artefacts in json

Branch which is target of publishing
Contains ARM template for Deployment



Repository name * ⓘ
☒ Create new ☐ Use existing

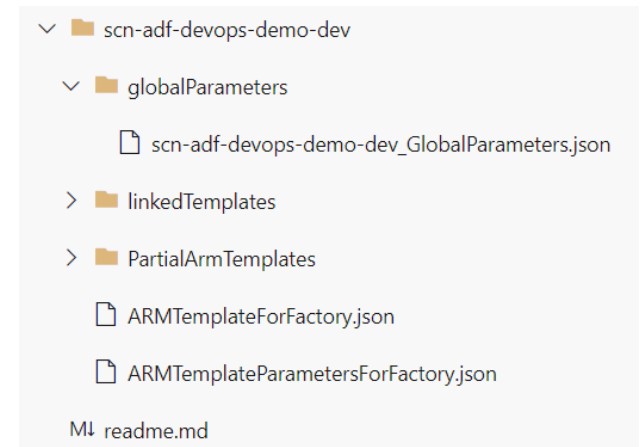
Collaboration branch * ⓘ

Publish branch * ⓘ

Root folder * ⓘ

Import existing resource
☒ Import existing resources to repository

Import resource into this branch * ⓘ
☒ Use Collaboration ☐ Create new ☐ Use existing



Demo

Source Code Integration for ADF

Config ADF best practices for CI/CD

- Set Name of ADF instances to x+_[environment] e.g. adf_dev
- Use Azure Key Vault for password storage, create one instance per environment
- Use Azure Key Vault integration in ADF to lookup secrets whenever possible
- Create linked service for Azure Key Vault in ADF

Azure key vault selection method ⓘ

☐ From Azure subscription ☒ Enter manually

Base URL *

Managed identity name: **scn-adf-devops-demo-dev**

Managed identity object ID: 00000000-0000-0000-0000-000000000000

Grant Data Factory service managed identity access to your Azure Key Vault. [Learn more](#) ⓘ

Annotations

Secret Management Operations

- ☒ Get
- ☒ List
- ☐ Set
- ☐ Delete
- ☐ Recover
- ☐ Backup
- ☐ Restore

Add access policy ...

Add access policy

Configure from template (optional)

Key permissions

0 selected

Secret permissions

2 selected

Certificate permissions

0 selected

Select principal *

scn-adf-devops-demo-prod

Object ID: 00000000-0000-0000-0000-000000000000

Authorized application ⓘ

None selected

Add

How make statics values configurable

- Databricks Access Token is configurable from Azure Key Vault
- Other properties as Workspace URL or a cluster ID have no direct support BUT have be changed during deployment
- HOW to handle this?

Databricks Workspace URL * ⓘ
http://databricks.net

Authentication type *
Access Token

Access token Azure Key Vault

AKV linked service * ⓘ
KeyVault

Secret name *
DatabricksAccessToken

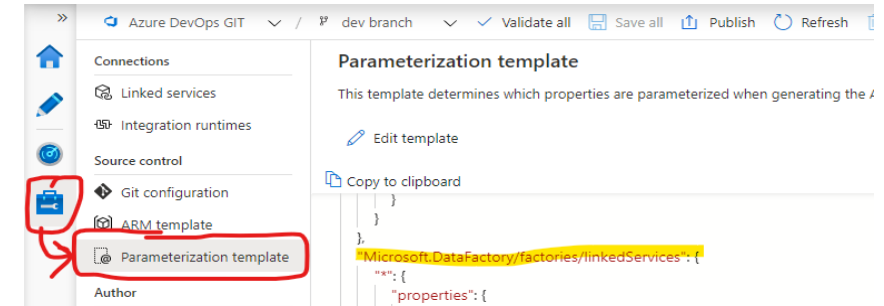
Secret version ⓘ
Use the latest version if left blank

Select cluster
☐ New job cluster ☒ Existing interactive cluster ☐ Existing instance pool

Existing cluster ID * ⓘ
msw397

Linked services - change parametrization template

- Insert the additional properties, check ADF documentation for linked services special items
- Pass the values for configuration in Azure DevOps



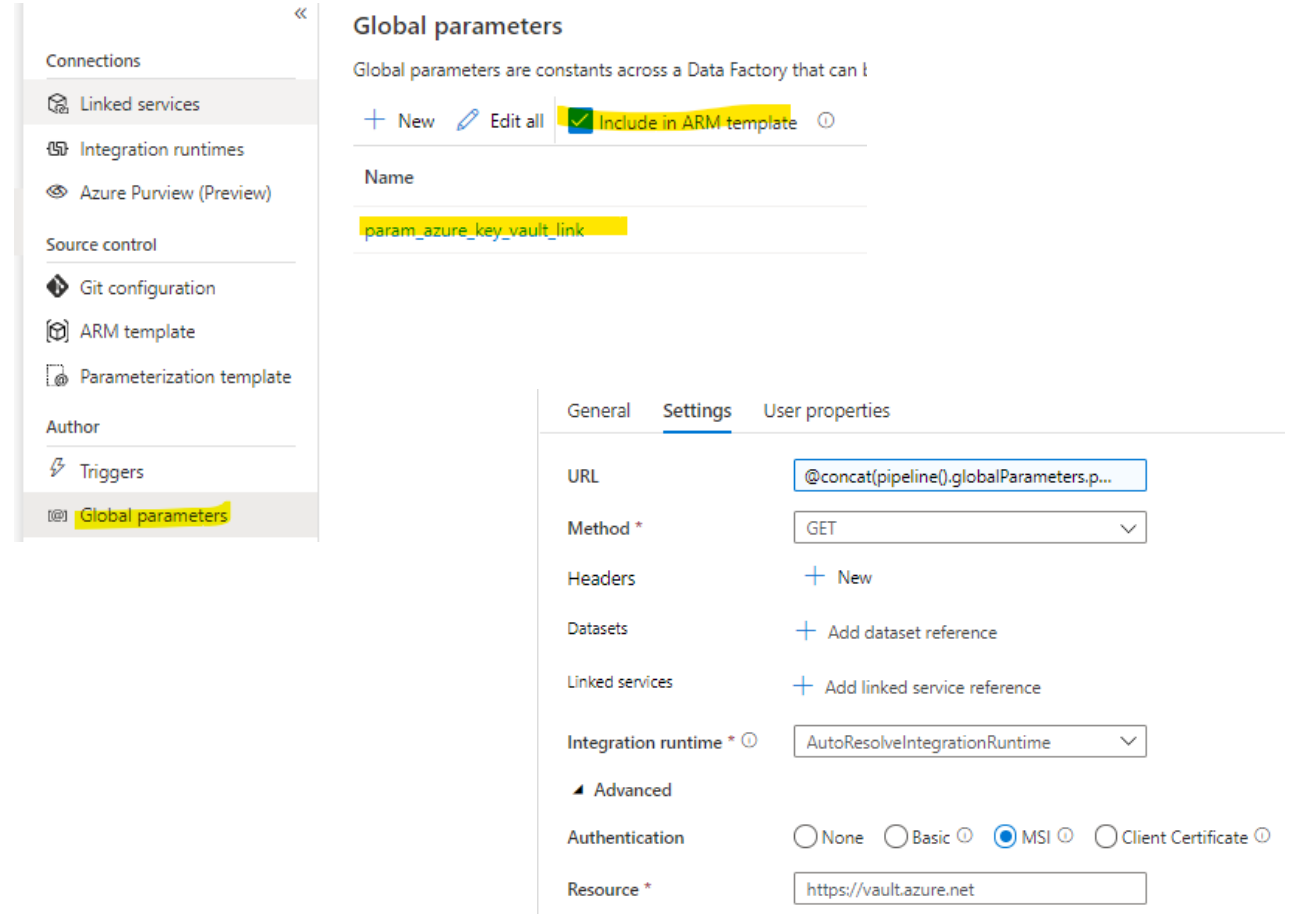
```

systemnumber : = ,
"server": "=",
"url": "=",
"functionAppUrl": "=",
"environmentUrl": "=",
"aadResourceId": "=",
"sasUri": "[-sasUri:secureString",
"sasToken": "]",
"connectionString": "[-connectionString:secureString",
"hostKeyFingerprint": "=",
"existingClusterId": "=",
"domain": "-."
}

```


Alternative: Using global parameters

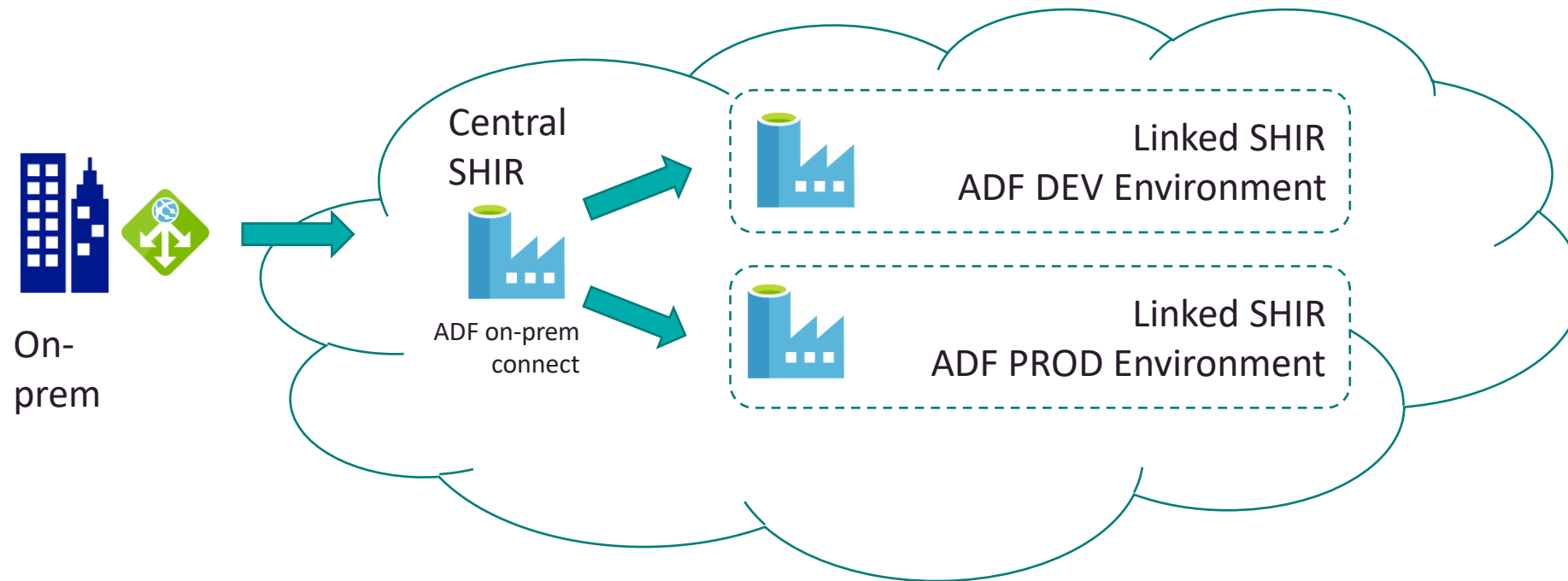
- Global parameters could be included and configured in ARM template
- Usable in any dynamic property – could be a path to a REST resource



The screenshot displays the Azure Data Factory configuration interface. On the left, a sidebar menu lists various components: Connections, Linked services, Integration runtimes, Azure Purview (Preview), Source control, Git configuration, ARM template, Parameterization template, Author, Triggers, and Global parameters (highlighted). The main panel is titled 'Global parameters' and includes a description: 'Global parameters are constants across a Data Factory that can be used in ARM templates'. It features buttons for '+ New', 'Edit all', and 'Include in ARM template' (checked). Below, a table lists global parameters, with 'param_azure_key_vault_link' highlighted. To the right, the 'Settings' tab for a linked service is shown, with fields for URL (@concat(pipeline().globalParameters.p...)), Method (GET), Headers (+ New), Datasets (+ Add dataset reference), Linked services (+ Add linked service reference), Integration runtime (AutoResolveIntegrationRuntime), Authentication (MSI selected), and Resource (https://vault.azure.net).

Linked Self-hosted Integration Runtimes

Same setup for SHIR in different environments



Demo

Config ADF

Azure Pipelines

- Azure Pipelines != Azure Data Factory Pipelines ;-)
- Pre- and Post Deployment steps
- ADF ARM Template – what is deployed what not?
- Author in Classic Editor (Visual) vs. YAML (Code)
- Components of ADF ARM template deployments
- Settings

Azure Pipelines – pre and post deployment steps

- ARM Deployment incremental vs. full
- Garbage collection
- De/Activate triggers on target system during deployment

ADF ARM Template – what is deployed what not?

- All parts of Azure Data Factory like linked services, datasets, pipelines, integrated runtimes...
- Not included is external processing infrastructure and code parts:
 - E.g. EXEC Databricks Notebook
 - Included: Databricks Configuration, Dataset definition, Pipeline with exec task
 - Not included: Databricks Workspace, Cluster, Code in Notebook
 - Optionally included: Cluster Configuration
 - applies to SSIS, SQL Server Stored Procs...

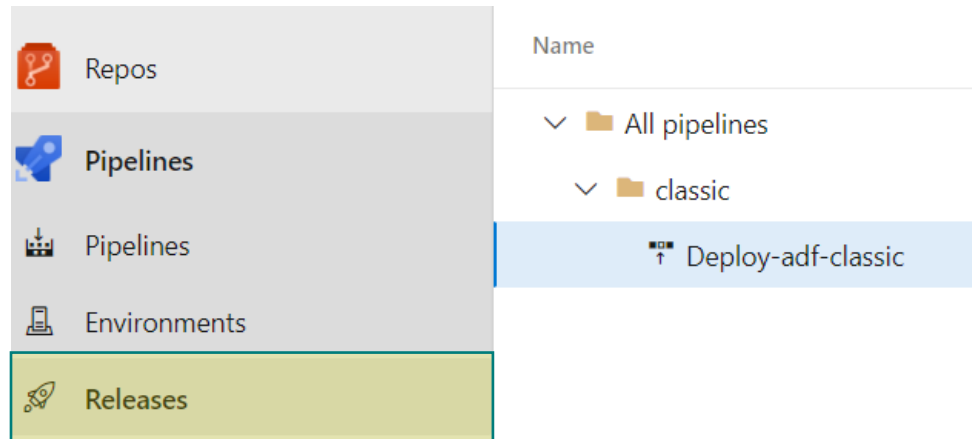
ADF Deployment – what has to be adapted

- Change connection definitions to linked services in other environment
- Change credentials & secrets (or use different Azure Key Vaults)
- Change central properties like name of Data Factory and paths

Azure Pipelines: Classic oder YAML?

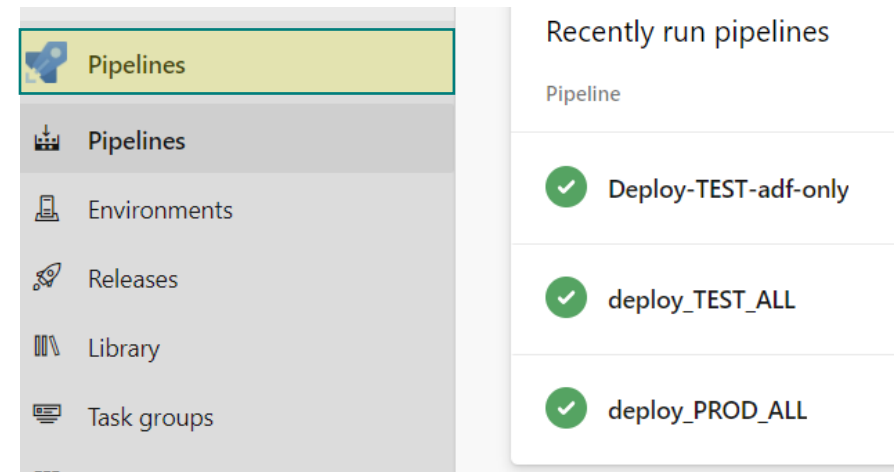
Pro Classic:

- Easiest way
- many 3rd Party components available



Pro YAML:

- Transparent code
- Versionable in Repo
- Good Overview
- Widely used



Azure Pipelines - Classic

Add tasks | [Refresh](#)

Marketplace ^



Azure Data Factory

This extension adds release tasks related to Azure Data Factory: publishing JSON, suspend/resume pipelines

by Jan Pieter Posthuma | [2,890 installs](#)



Deploy Azure Data Factory by SQLPlayer

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

[azure data factory](#) [×](#)

[New release pipeline](#) > [Release-2](#) > [Deploy-TEST-adf-only](#) [✓](#) Succeeded

[← Pipeline](#) [Tasks](#) [Variables](#) [Logs](#) [Tests](#) | [Deploy](#) [Cancel](#) [Refresh](#) [Download a](#)

Deployment process

Succeeded

[Agent-Job-Deploy-TEST_adf-only](#)

Succeeded

Agent-Job-Deploy-TEST_adf-only

Pool: [Azure Pipelines](#) · Agent: Hosted Agent

- [✓ Initialize job · succeeded](#)
- [✓ Download Artifacts · succeeded](#)
- [✓ Disable Triggers in Data Factory on TEST · succeeded](#)
- [✓ Deploy Data Factory · succeeded](#)
- [✓ Enable Triggers in Data Factory on TEST · succeeded](#)
- [✓ Finalize Job · succeeded](#)

Deploy-TEST-adf-only

Deployment process

Agent-Job-Deploy-TEST_adf-only

[Run on agent](#)



Disable Triggers in Data Factory on TEST

Azure Data Factory Trigger



Deploy Data Factory

ARM template deployment



Enable Triggers in Data Factory on TEST

Azure Data Factory Trigger

Azure Pipelines - YAML

← Deploy-TEST-adf-only

working

sceDevOpsDemos / azure_pipelines/deploy_test_adf_only.yml

```

1 trigger:
2   - none
3
4 # Get repository checked out on adf_publish branch for DataFactory deployment.
5 resources:
6   repositories:
7     - repository: ADFPublish
8       type: git
9       name: sceDevOpsDemos/sceDevOpsDemos
10      ref: adf_publish
11     - repository: SynPublish
12       type: git
13       name: sceDevOpsDemos/sceDevOpsDemos
14       ref: workspace_publish
15
16 variables:
17   - group: 'TestEnvironmentKeyVault'
18   - name: project_prefix
19     value: ''
20
21   #devops_service_connection_name_dev: 'Azure-Subscription-Dev'
22   - name: devops_service_connection_name_test
23     value: 'Azure-Subscription-Test'
24   - name: build_pipeline_id
25     value: -1
26
27
28 stages:
29
30 # TEST environment
31 - template: Stages/stage-template-adf-only.yml
32   parameters:
33     stage_name: 'Test'
34     env_short: 'test'
35     env_devops: 'sceDevOpsDemos--Test'
36     azureSubscription: $(devops_service_connection_name_test)
37     rg_adf: 'pj_dataplatform_test'

```

Use snippets

Tasks



data factory



Azure Data Factory Delete Items

Delete Azure Data Factory V2 items, like Datasets,...



Azure Data Factory Deployment

Deploy Azure Data Factory Datasets, Pipelines an...



Azure Data Factory Trigger

Start/stop an Azure Data Factory Trigger



Azure Data Factory Trigger Pipeline

Trigger Azure Data Factory V2 Pipelines



Build Azure Data Factory code

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



Publish Azure Data Factory

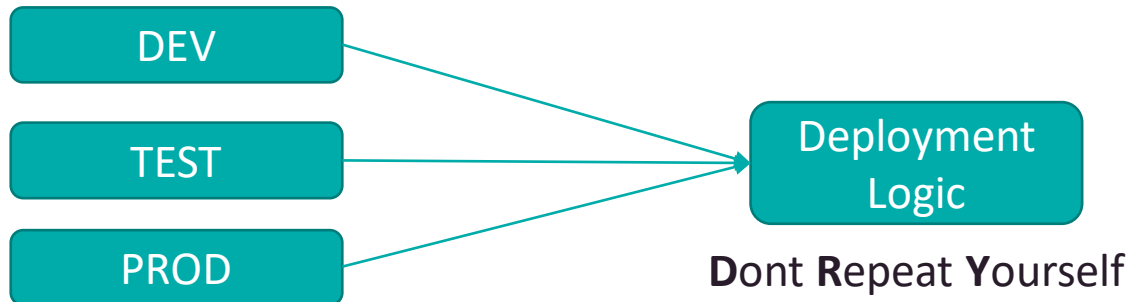
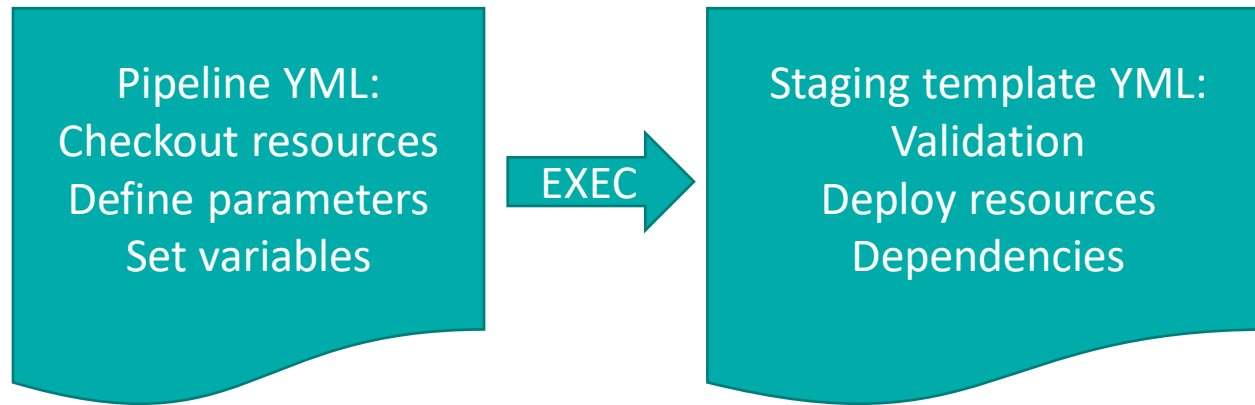
Deploys entire ADF (v2) from JSON files to ADF in...



Query Azure Data Factory runs

Observe the configured Azure Data Factory runs f...

Azure Pipelines – YAML: DRY principle



Azure Pipelines - Settings

- Define Service Connections under Project Settings
- Define your Environments once and reuse them



Service connections

 Filter by keywords

 Azure Subscription Prod

 Azure Subscription Test

Environments


Environment	Status
sceDevOpsDemos - Production	 #20210226.5 on deploy_PROD_ALL
sceDevOpsDemos - Test	 #20210414.1 on sceDevOpsDemos


Azure Pipelines - Settings

- Permissions needed for new pipelines – watch requests in DevOps

⚠ This pipeline needs permission to access a resource before this run can continue to Test [View](#)


Checks for Test

 Permission
Permission needed

 **sceDevOpsDemos**
Repository

[Permit](#)

- Link your Key Vault instances under Pipelines Library to access your secrets

Library >  TestEnvironmentKeyVault

[Variable group](#) | [Save](#) | [Clone](#) | [Security](#) | [Help](#)

Properties

Variable group name

TestEnvironmentKeyVault

Description

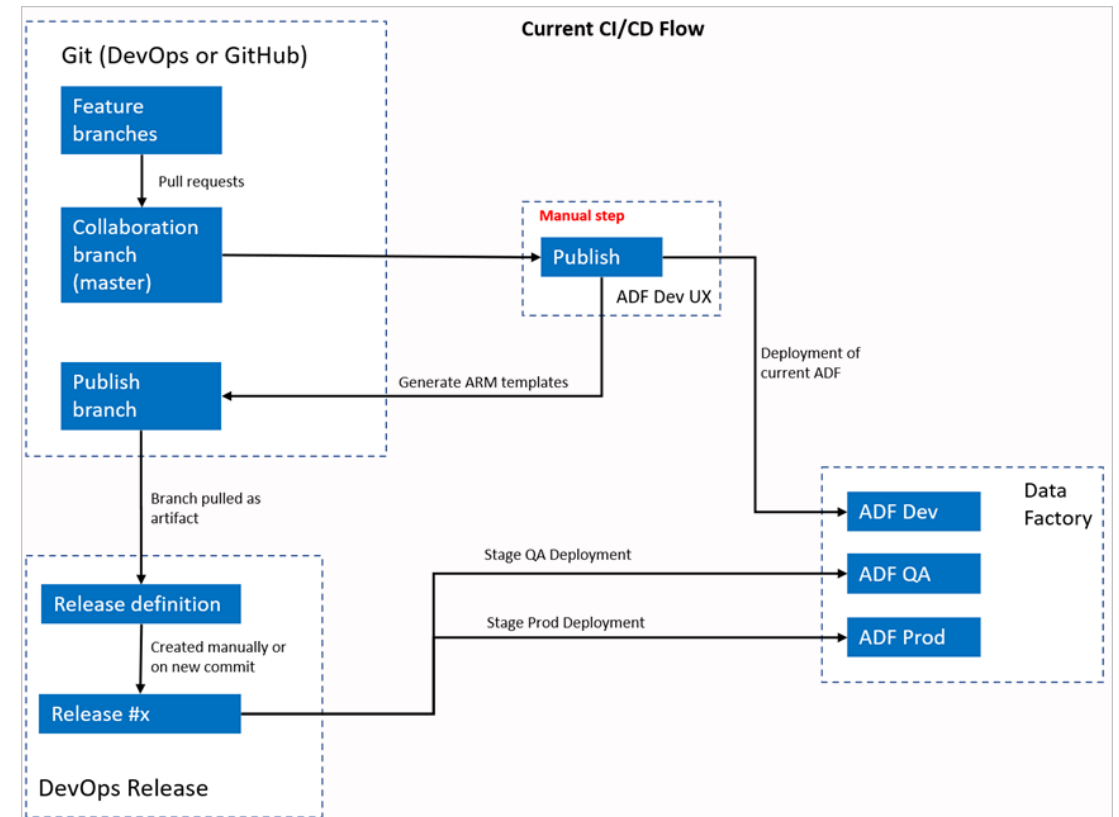
☒ Allow access to all pipelines
☒ Link secrets from an Azure key vault as variables ⓘ

Demo

Using Azure Pipelines for Deployments

Teamwork #1 build releases

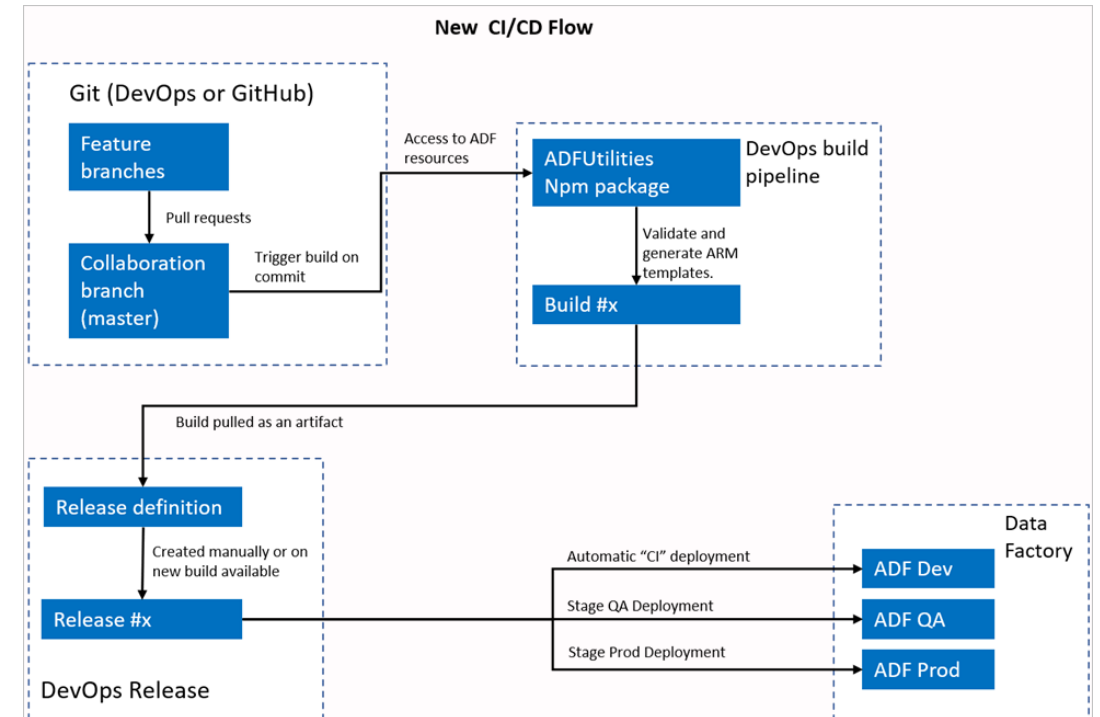
- Working in feature branches
- Pull to master for merging
- Publish in ADF GUI to DEV
- Generate Release manual from version in git (adf_publish branch)
- Deploy that versions to QA/PROD environments



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

Teamwork #2 trigger releases

- Like before but using Azure pipeline to simulate „publish“ step from ADF GUI
- Could be triggered by user or by commit into a branch (master)
- ARM templates generated from repo sources (json Objects) to repo target folders as artifact
- Release to any ADF environment possible (DEV not necessary)



<https://docs.microsoft.com/en-us/azure/data-factory/continuous-integration-deployment-improvements#overview>

Teamwork #2 Notes about releases

- Attention: existing git repo config of target is removed when deploying the ARM template
- The doc to implement this from MS is not completely correct:
 - current version of azure-data-factory-utilities is 0.1.5
 - package.json has to be on top level of repo
 - Make sure that all folders fit to your setup
 - Npm run *start* vs. *build* confusion

<https://docs.microsoft.com/en-us/azure/data-factory/continuous-integration-deployment-improvements#overview>

Demo

Named releases

ADF vs. Synapse

- Azure Synapse Workspace & Studio as clamp for data services: Synapse DBs, ADF, Notebooks, Power BI...
- Source code integration since late 2020
- Most ADF stuff supported
- Project artefacts of different components stored as whole solution in Synapse, separate branch for „adf_publish“
- Not everything working smooth now, will be solved in time

When fully working this will add value to simplicity for DevOps in Analytics projects!

Finally – what about the DoD?

There is no Definition of Done which is valid for every project

But getting better in Application Lifecycle Management makes it a lot easier to define a very good one!

