#Luedinghausen

Application Lifecycle Management in Azure Data Factory

Stefan Kirner

@KirnerKa

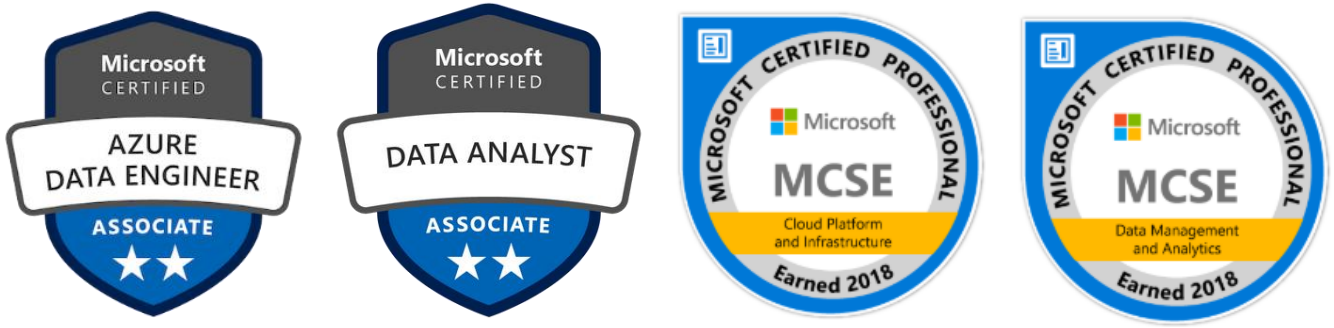
scieneers
DRIVEN BY DATA

"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

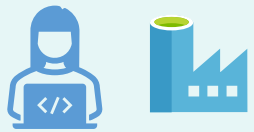
- Kick-off
- Source Code Integration
- ADF Config best practices
- Azure Pipelines
- Teamwork
- ADF vs. Synapse
- Close

Why DevOps for ADF?

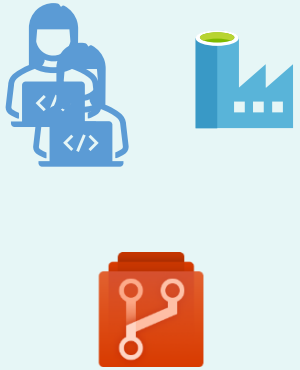
- Lessons learned in (classic) software development:
- Transparent code base (track changes)
- Safe development (revert changes)
- Stable production environment (CI/CD)
- Avoid boring work (automation)
- Enable teamwork (collaboration)
- Better ADF dev. performance (10x)

https://docs.microsoft.com/en-us/azure/data-factory/source-control

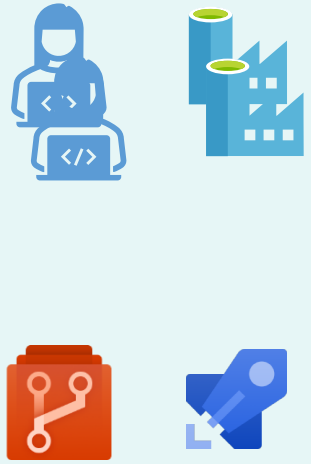
Different states of ADF development



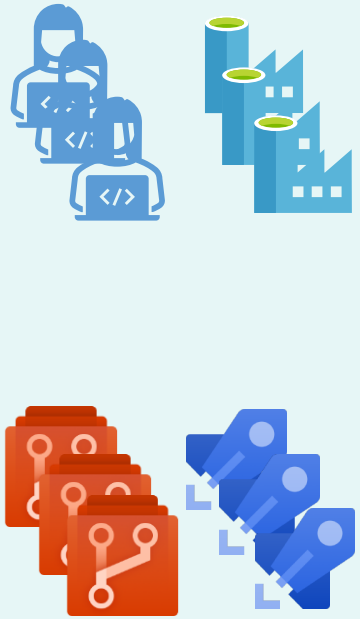
Starting:
ADF live edit



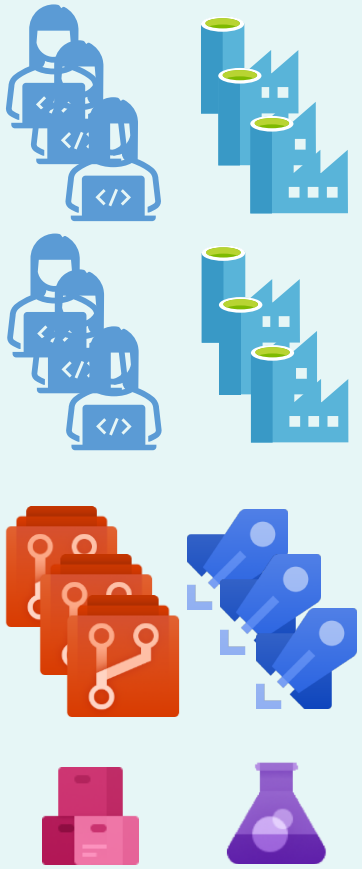
Using source
code versioning



Dedicated ADF
Environments
& Deployment
Pipelines



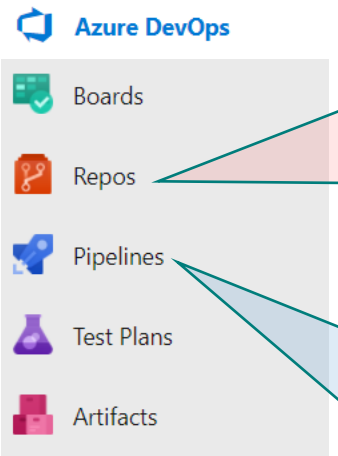
Different
branches & Orga

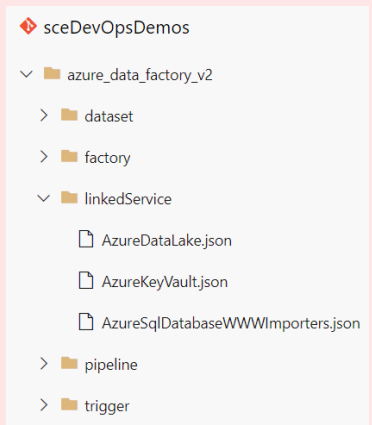


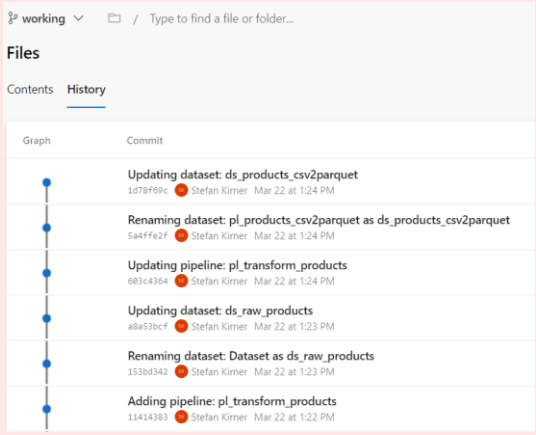
Named Releases,
Continuous Tests

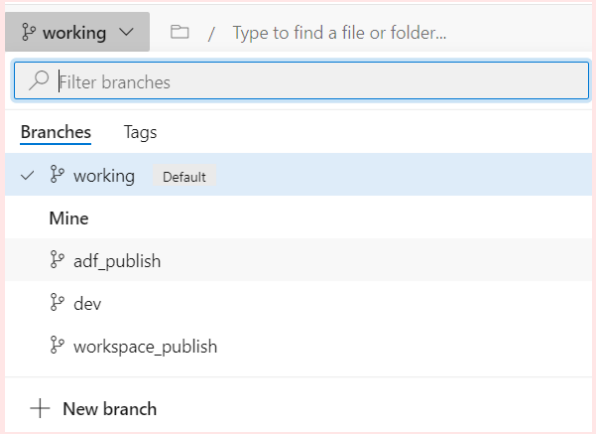
(Most) Useful things in Azure DevOps for Data Factory?

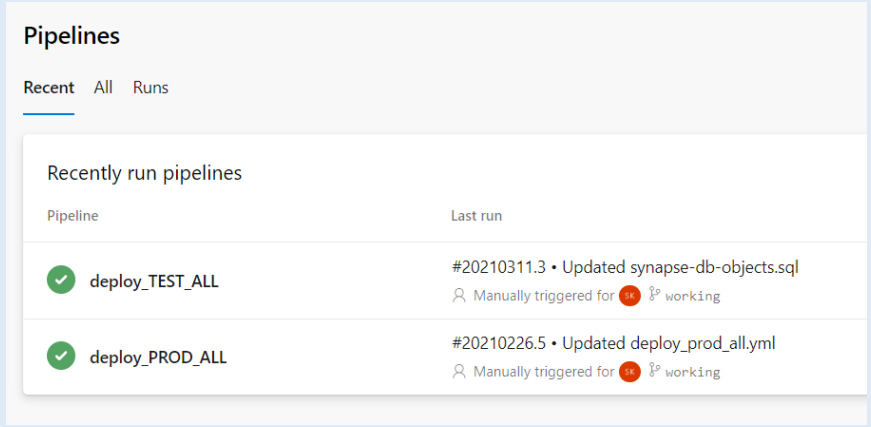
scieneers
DRIVEN BY DATA

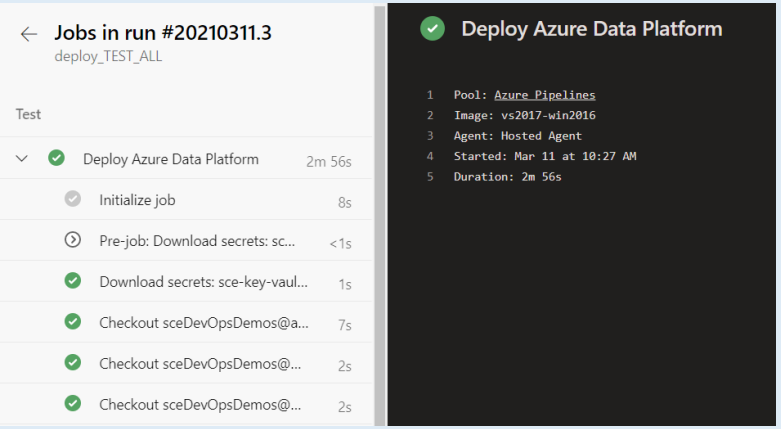








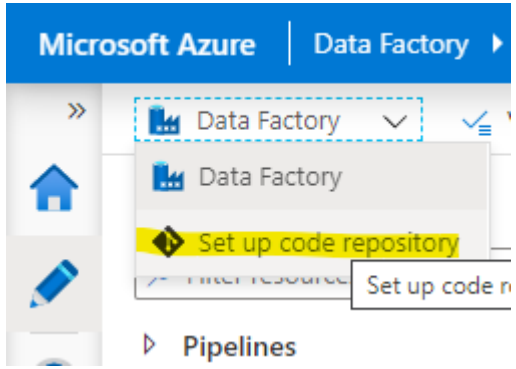


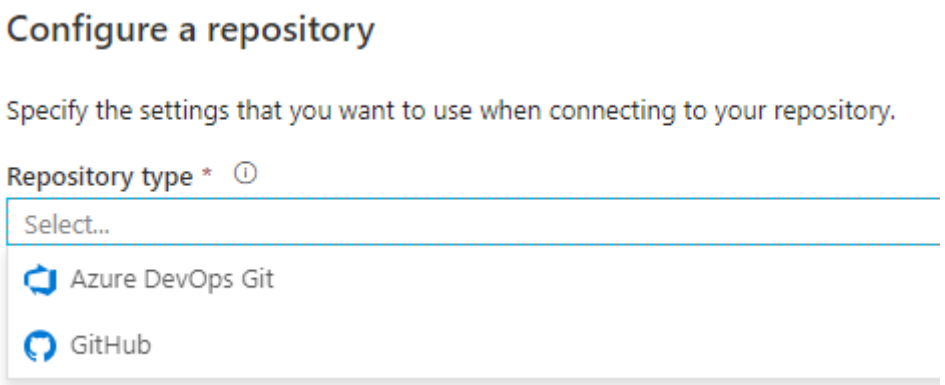


Setup Source Code Integration

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.

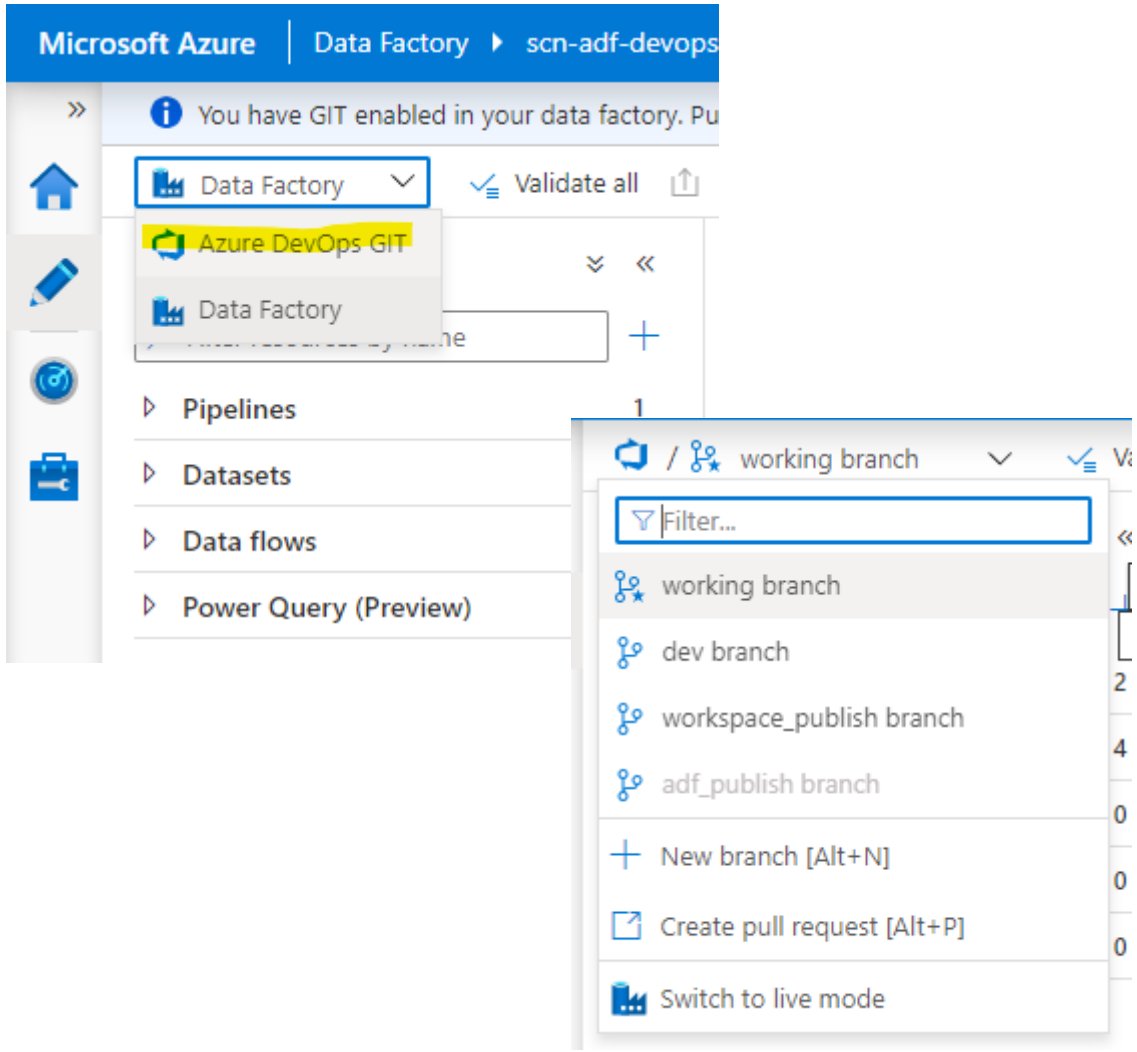




https://docs.microsoft.com/en-us/azure/data-factory/source-control

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



https://docs.microsoft.com/en-us/azure/data-factory/source-control

Setup Source Code Integration

Branch which is used for publishing
Contains development artefacts in json

Repository name *
Create new
Use existing
test

Collaboration branch *
working

Publish branch *
adf_publish

Root folder *
/azure_data_factory_v2

Import existing resource
Import existing resources to repository

Import resource into this branch *
Use Collaboration
Create new
Use existing
working

Branch which is target of publishing
Contains ARM template for Deployment

- azure_data_factory_v2
  - dataset
    - ds_azure_sql.json
    - ds_datalake_generic_parquet.json
    - ds_datalake_product_csv.json
    - ds_datalake_specfile_parquet.json
  - factory
    - scn-adf-devops-demo-dev.json
  - linkedService
  - pipeline
  - trigger

- scn-adf-devops-demo-dev
  - globalParameters
    - scn-adf-devops-demo-dev_GlobalParameters.json
  - linkedTemplates
  - PartialArmTemplates
  - ARMTemplateForFactory.json
  - ARMTemplateParametersForFactory.json
  - readme.md

https://docs.microsoft.com/en-us/azure/data-factory/source-control

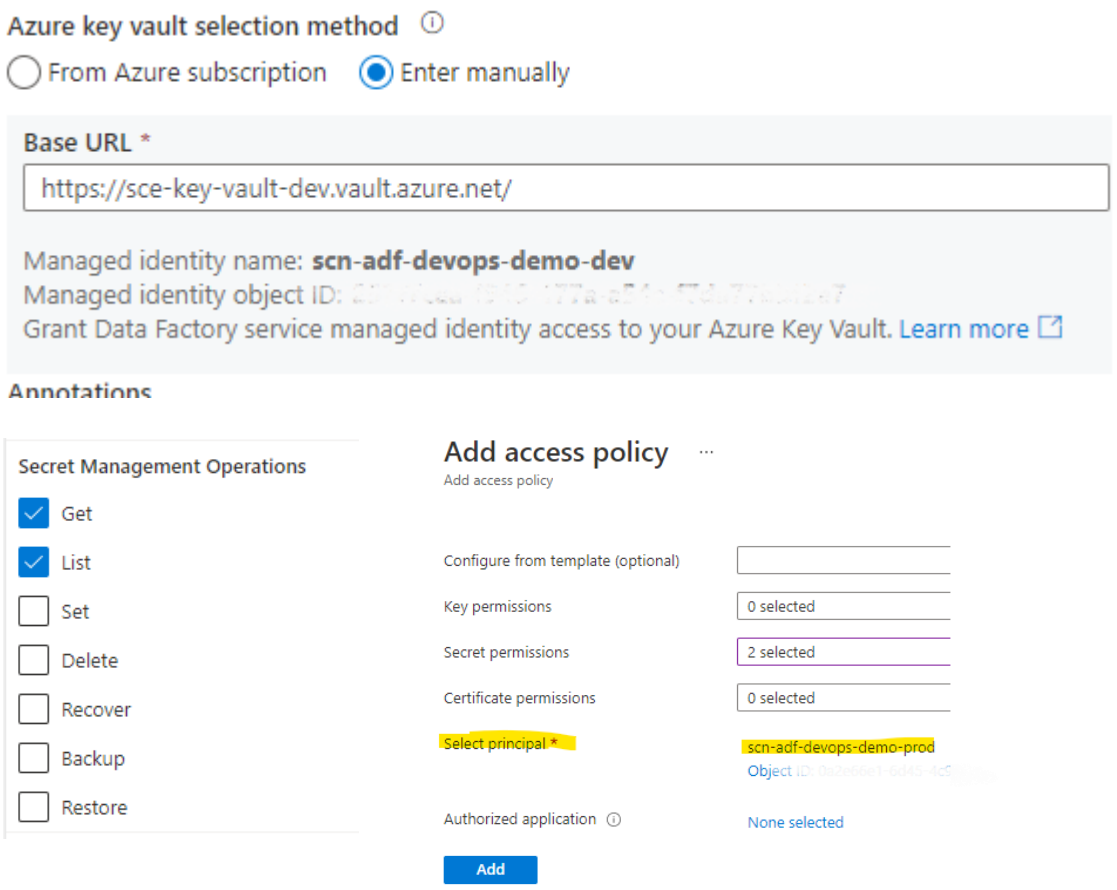
scieneers
DRIVEN BY DATA

Demo

Source Code Integration for ADF

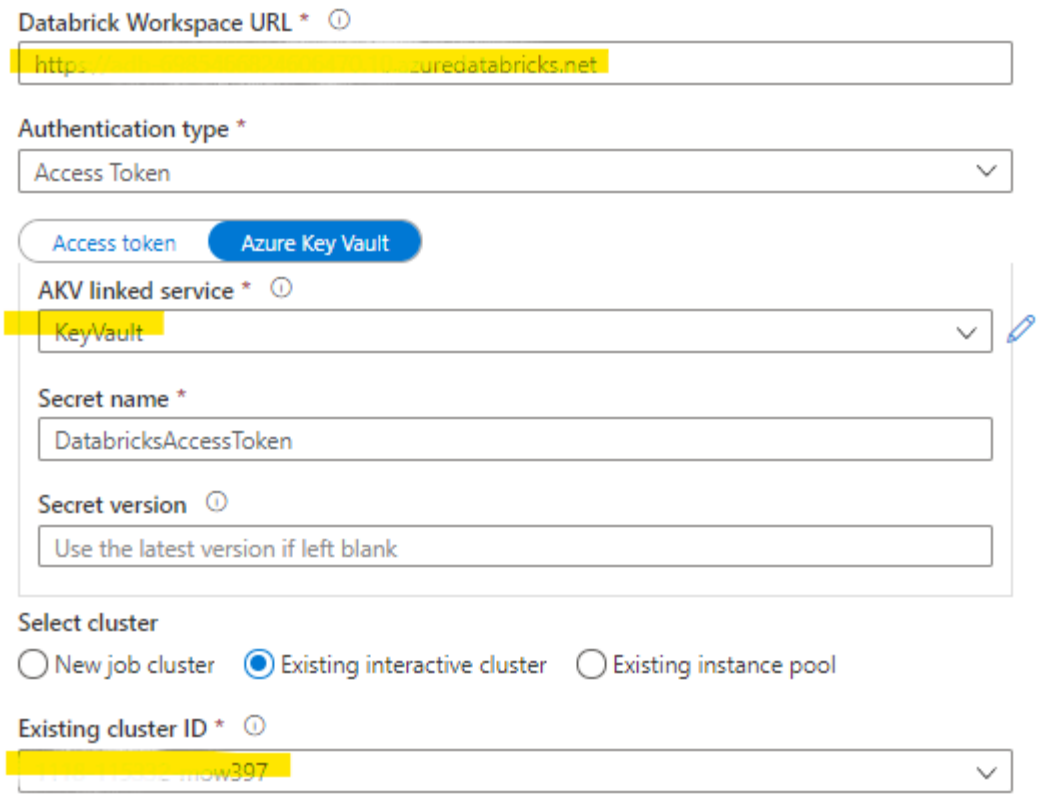
Config ADF best practices for CI/CD

- 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



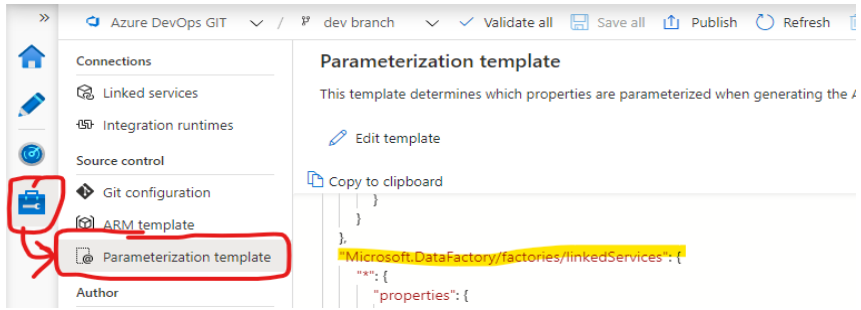
How make statics values configurable

- 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?



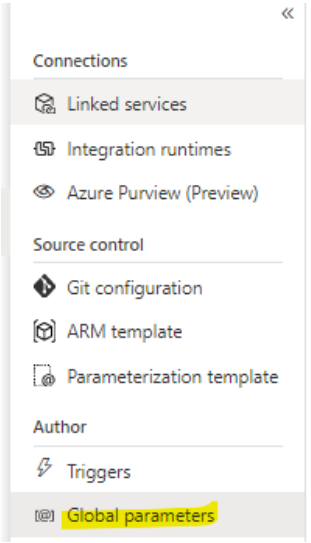
Linked services - change parametrization template

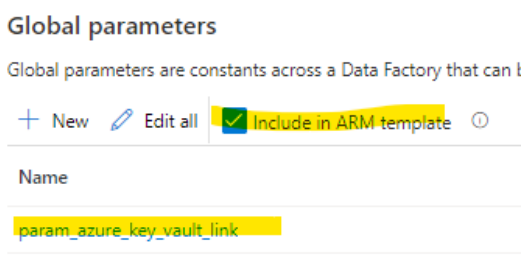
- Insert the additional properties, check ADF documentation for linked services special items
- Pass the values for configuration in Azure DevOps

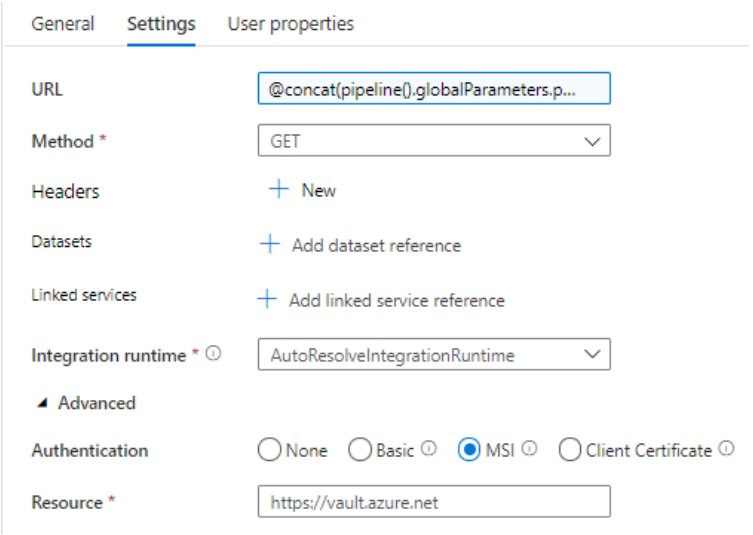


Alternative: Using global parameters

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

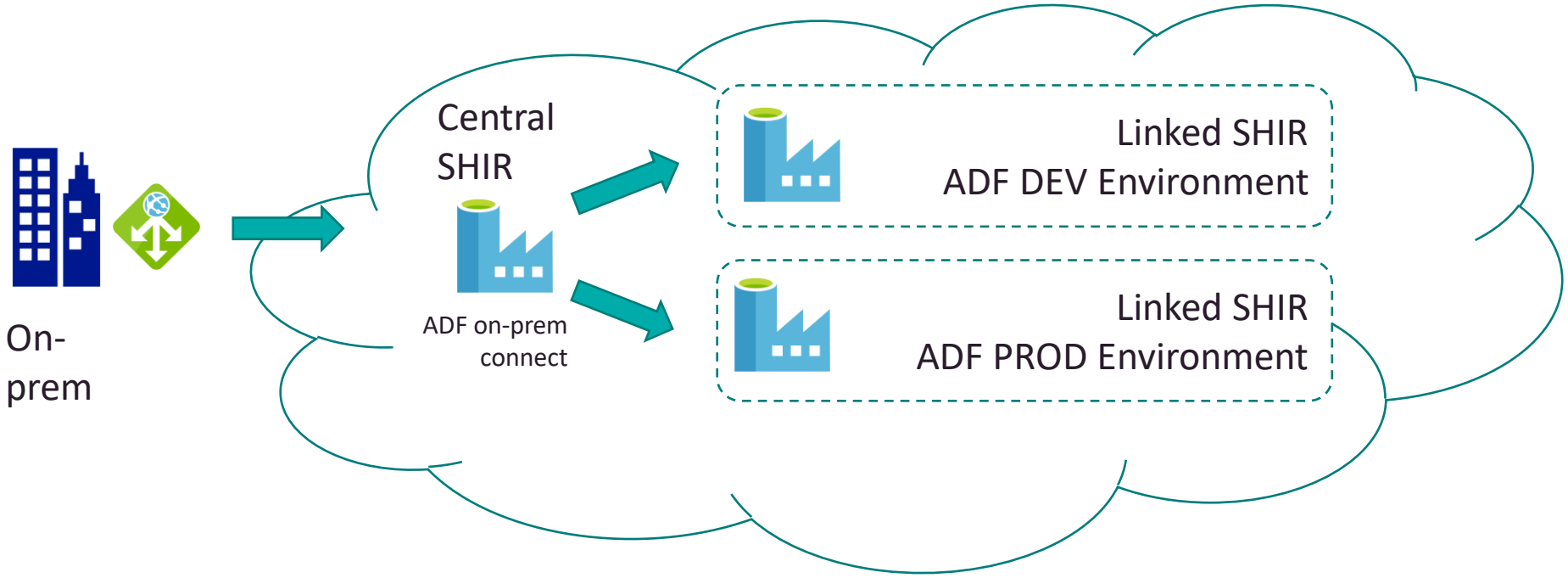






Linked Self-hosted Integration Runtimes

Same setup for SHIR in different environments



scieneers
DRIVEN BY DATA

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

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

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…

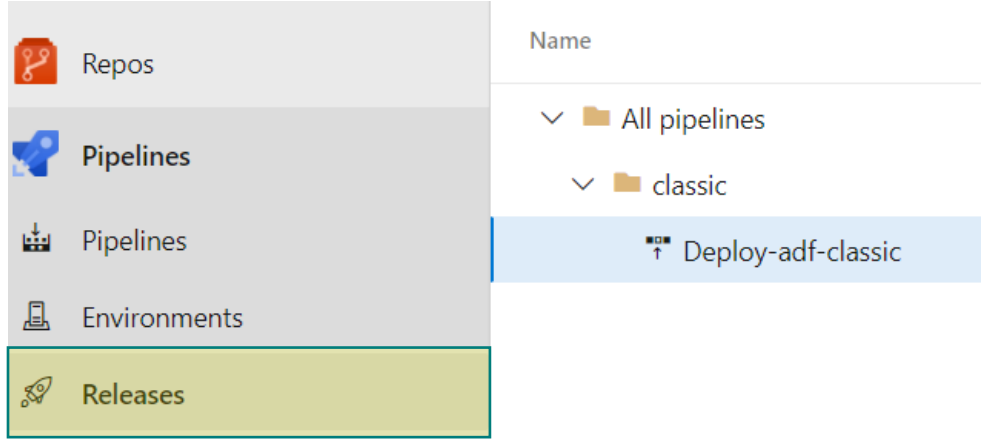
ADF Deployment – what has to be adapted

- 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?

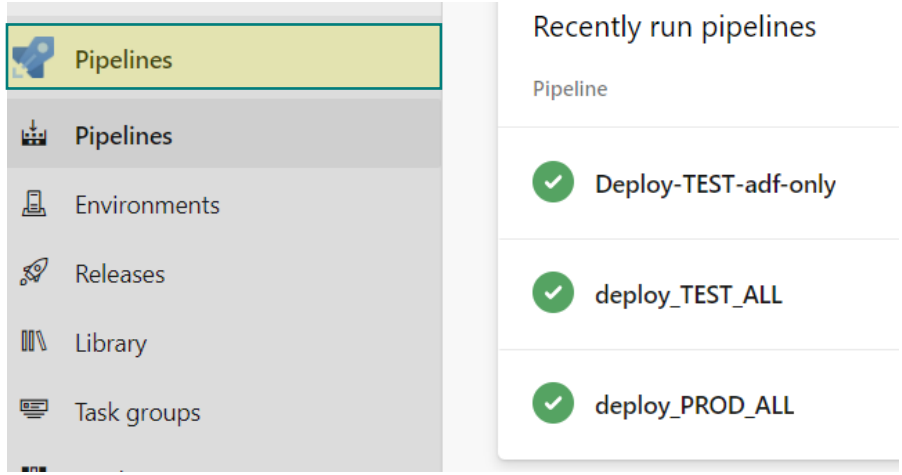
Pro Classic:

- Easiest way
- many 3rd Party components available



Pro YAML:

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



Azure Pipelines - Classic

Add tasks | Refresh

azure data factory

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

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**

```yaml
trigger:
- none

# Get repository checked out on adf_publish branch for DataFactory deployment.
resources:
  repositories:
    - repository: ADFPublish
      type: git
      name: sceDevOpsDemos/sceDevOpsDemos
      ref: adf_publish
    - repository: SynPublish
      type: git
      name: sceDevOpsDemos/sceDevOpsDemos
      ref: workspace_publish

variables:
- group: 'TestEnvironmentKeyVault'
- name: project_prefix
  value: ''

  #devops_service_connection_name_dev:  'Azure Subscription Dev'
- name: devops_service_connection_name_test
  value: 'Azure Subscription Test'
- name: build_pipeline_id
  value: -1


stages:

# TEST environment
- template: Stages/stage-template-adf-only.yml
  parameters:
    stage_name: 'Test'
    env_short: 'test'
    env_devops: 'sceDevOpsDemos - Test'
    azureSubscription: $(devops_service_connection_name_test)
    rg_adf: 'pj_dataplatform_test'
```

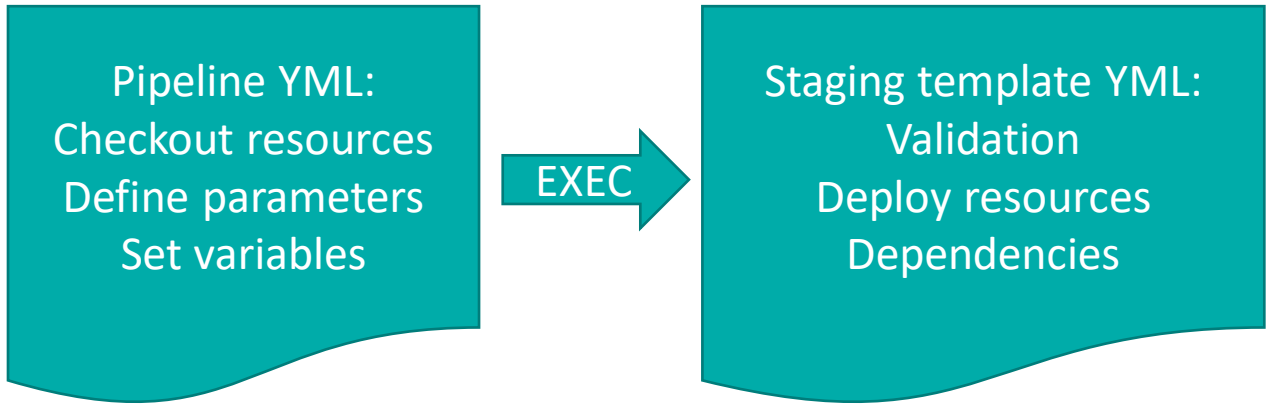
Use snippets

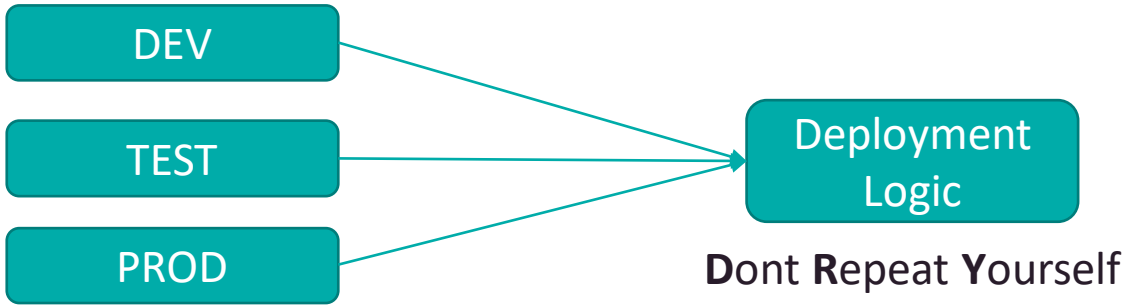
Tasks

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

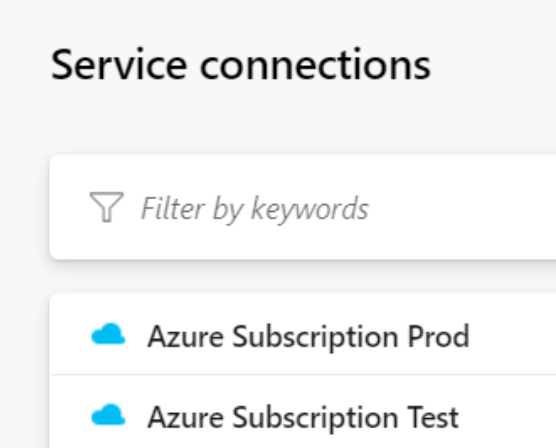




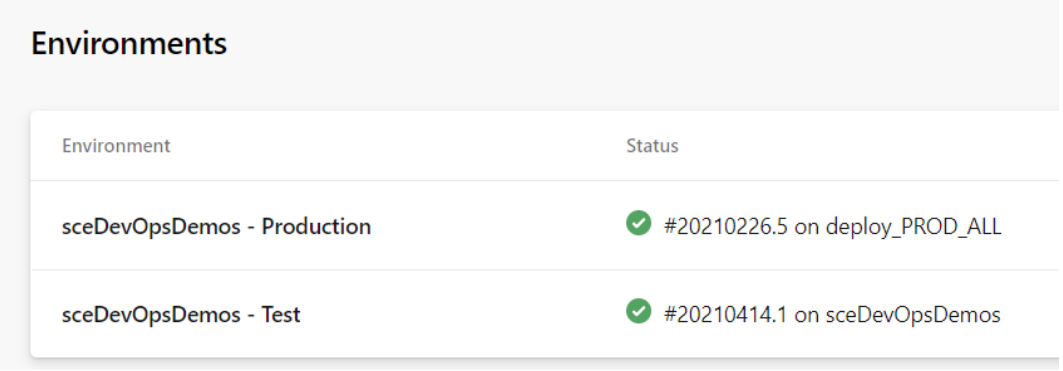
Dont **R**epeat **Y**ourself

Azure Pipelines - Settings

- Define Service Connections under Project Settings

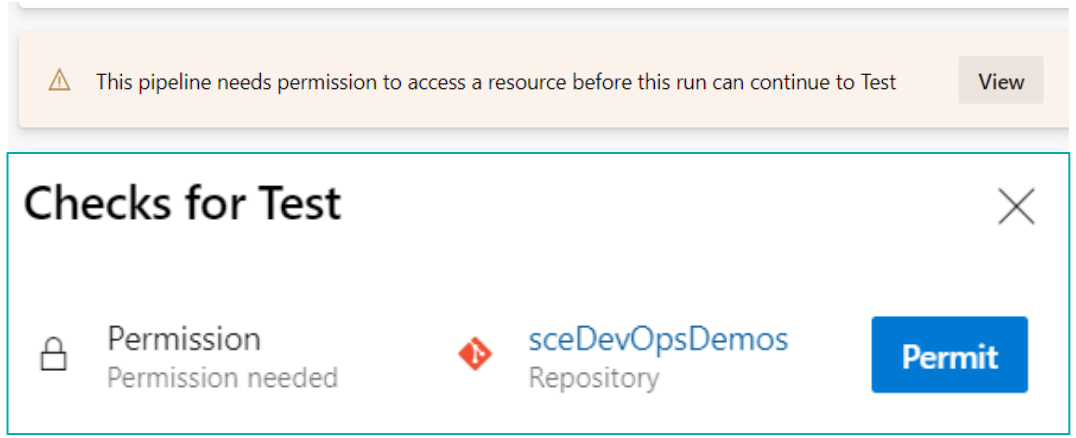


- Define your Environments once and reuse them

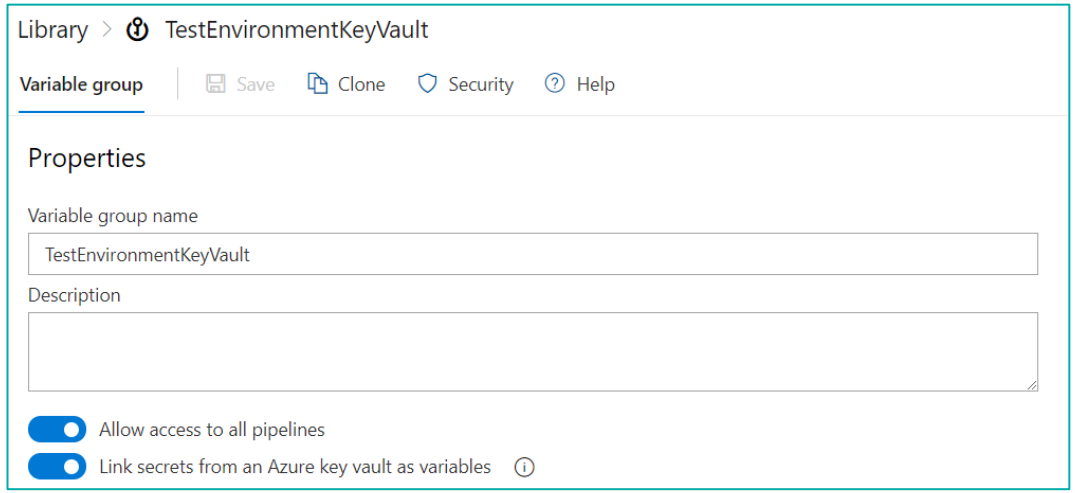


Azure Pipelines - Settings

- Permissions needed for new pipelines – watch requests in DevOps



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



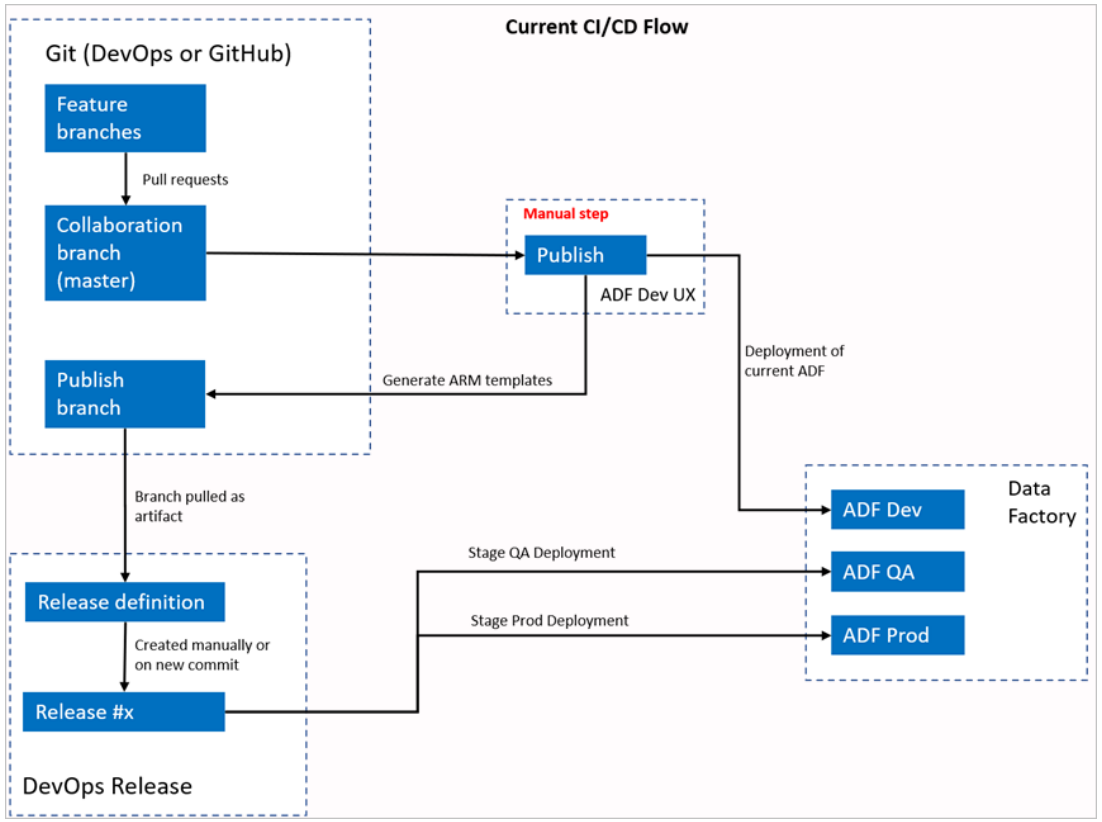
scieneers
DRIVEN BY DATA

Demo

Using Azure Pipelines for Deployments

Teamwork #1 build releases

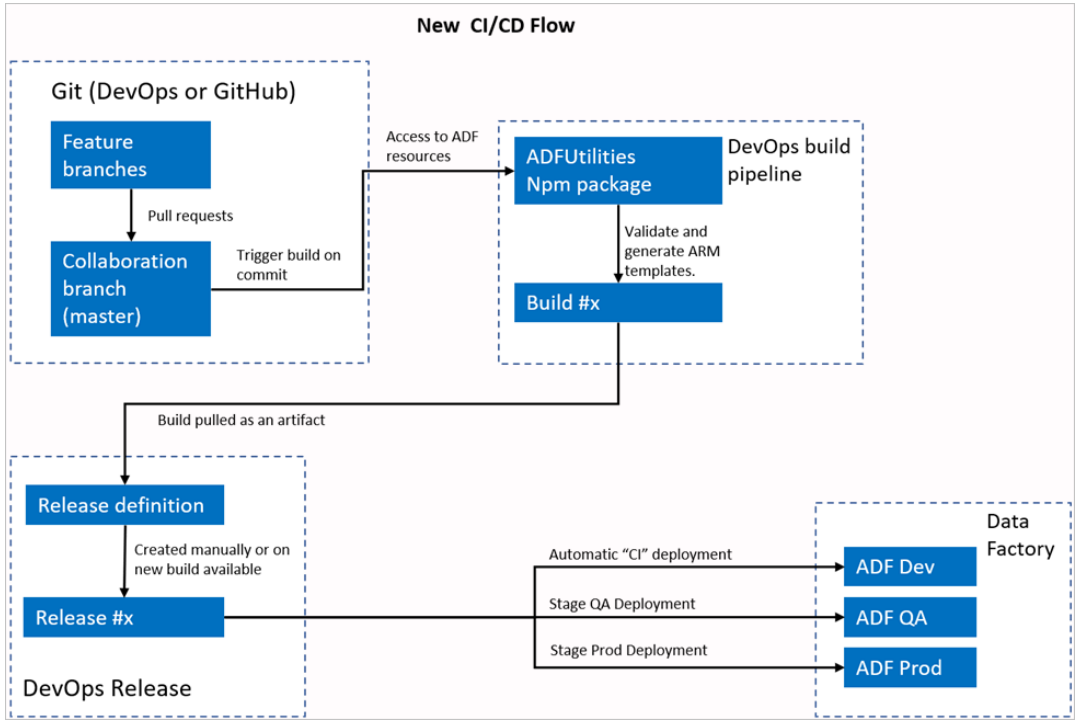
- Working in feature branches
- Pull to master for merging
- Publish in ADF GUI to DEV
- Generate Release manual from version in git (adf_publish branch)
- Deploy that versions to QA/PROD environments



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 differnt 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!


BY COMMUNITY • FOR COMMUNITY
#GLOBALAZURE
2021