Microsoft

POWER BI BRUSSELS

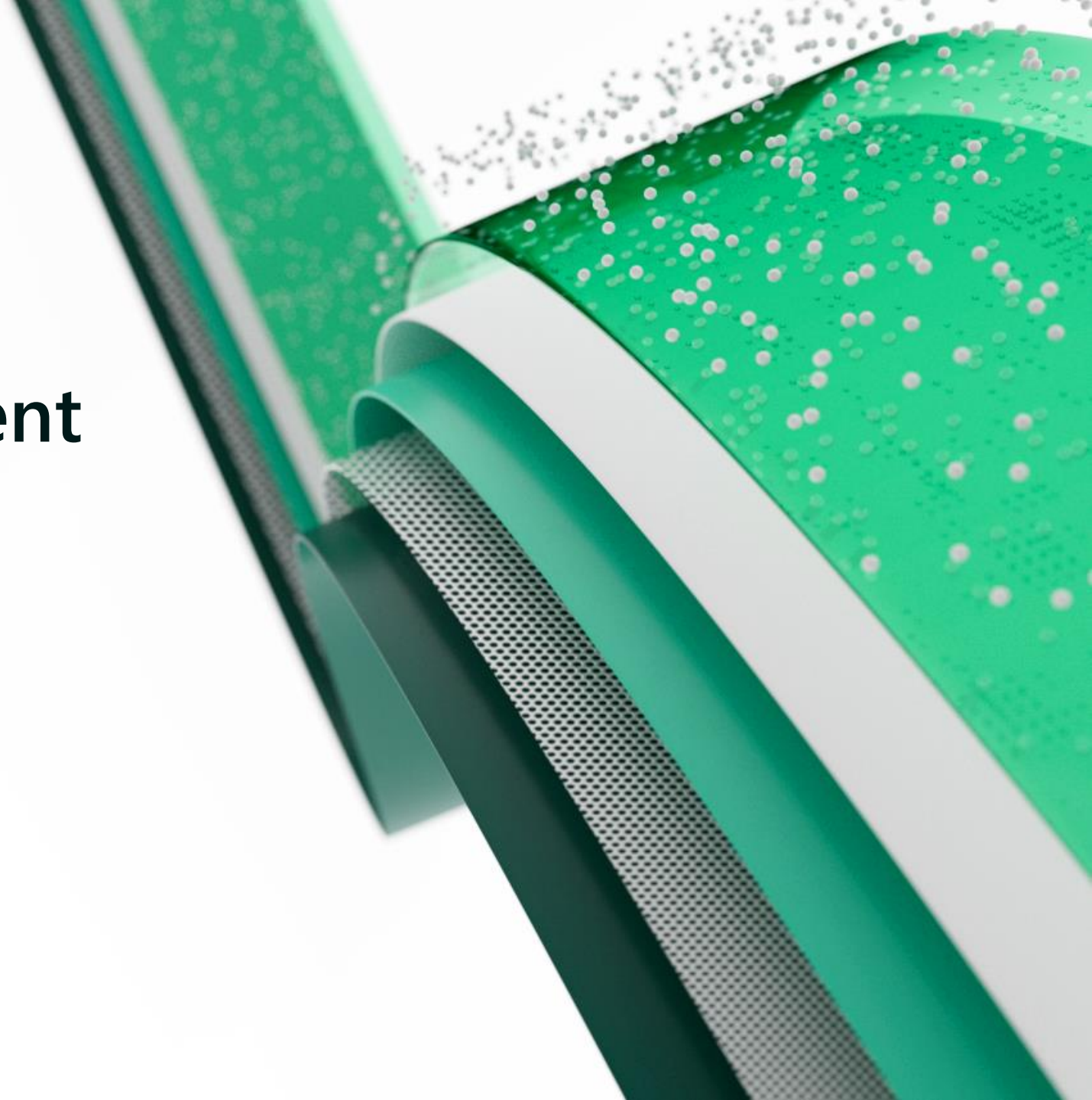
Move from deployment to production with Fabric and Power BI

Power BI Brussels User Group

Florian Boigner
Technical Specialist, Intelligence

/in/fboigner

florian.boigner@microsoft.com



What is lifecycle management?

(according to GPT)

- The process of **planning**, **developing**, **deploying**, and **maintaining** Power BI content across different environments within an organization.
- The goal is to ensure that Power BI content is developed, tested, and deployed in a **controlled and consistent manner**, from initial development through to production use.
- Key components:

Environments (Dev / Test / Prod)

Version Control

Documentation

Monitoring

Security and Permissions

Why is lifecycle management important?

Some reasons to consider


Keep track of changes across multiple versions


Backup and roll back functionality


Compare changes between versions


Collaborate with others


Separate tasks and permissions


Orchestrate your process with testing and approvals


Rules and automation

What is Git and Azure DevOps (ADO)?

(according to GPT)

- Git is a **distributed version control system** that tracks changes in code, enabling collaboration among developers and providing a history of revisions for software projects.

- Azure DevOps is a comprehensive **set of tools and services that facilitates the entire (software) development lifecycle**, from planning and coding to building, testing, and deployment, all while fostering collaboration among development teams.

Native tools available in Fabric

Power BI Desktop Developer Mode Preview

Development

Git integration Preview

Versioning & Collaboration

Deployment Pipelines

Orchestration

+APIs and many 3rd party solutions

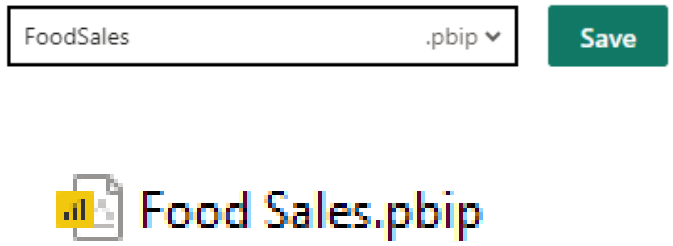
Power BI Desktop Developer Mode (preview)

Deep-dive blog

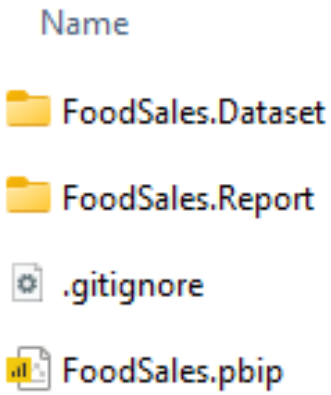
Documentation

Power BI Developer Mode (public preview)

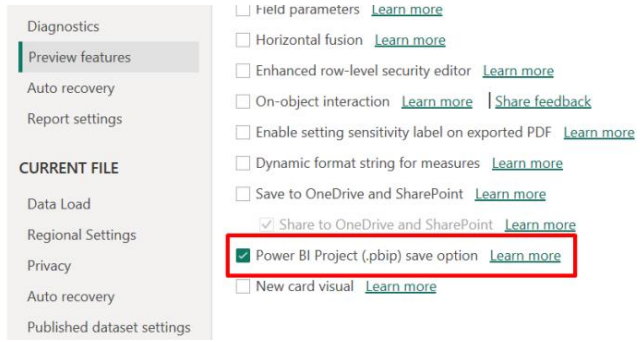
Save-as, open and publish Power BI Project files (.PBIP)



Folder with metadata files for Dataset and Report



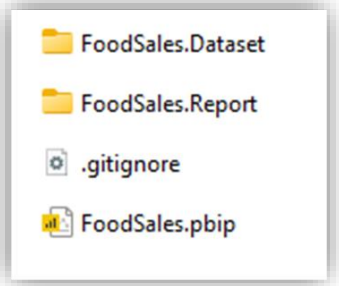
Needs to be enabled as a preview feature



Power BI Developer Mode (public preview)

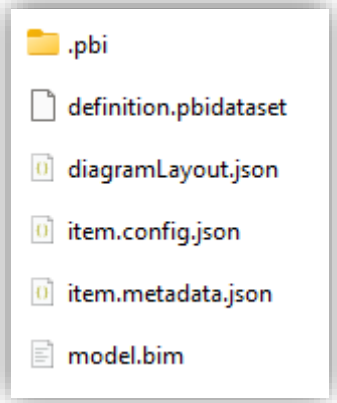
Power BI Metadata Files

Project folder



- PBIP file allows to open the project in Power BI Desktop
- PBIP file is only a shortcut to the folder

FoodSales.Dataset



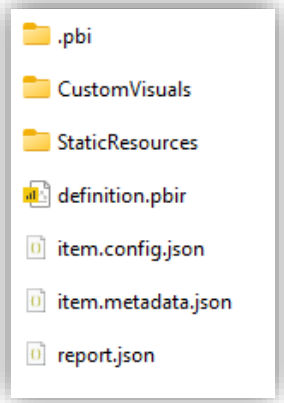
Dataset folder holds model definitions

Model.bim

```
{
  "compatibilityLevel": 1550,
  "model": {
    "annotations": [
      {
        "name": "__PBI_TimeIntelligenceEnabled",
        "value": "1"
      },
      {
        "name": "PBIDesktopVersion",
        "value": "2.121.903.0 (23.09)"
      },
      {
        "name": "PBI_QueryOrder",
        "value": "[\"Association\",\"Sales\",\"Store\",\"Calenda
      },
      {
        "name": "PBI_ProTooling",
        "value": "[\"DevMode\"]"
      }
    ],
    "culture": "en-US",
    "cultures": [
      {
        "name": "en-US",
        "linguisticMetadata": {
          "content": {
            "DynamicImprovement": "HighConfidence",
            "Entities": {
              "analysis_dax.net_sales": {
                "Definition": {
                  "Binding": {
                    "ConceptualEntity": "Analysis DAX",
                    "ConceptualProperty": "Net Sales"
                  }
```

Report folder holds report definitions

FoodSales.Report



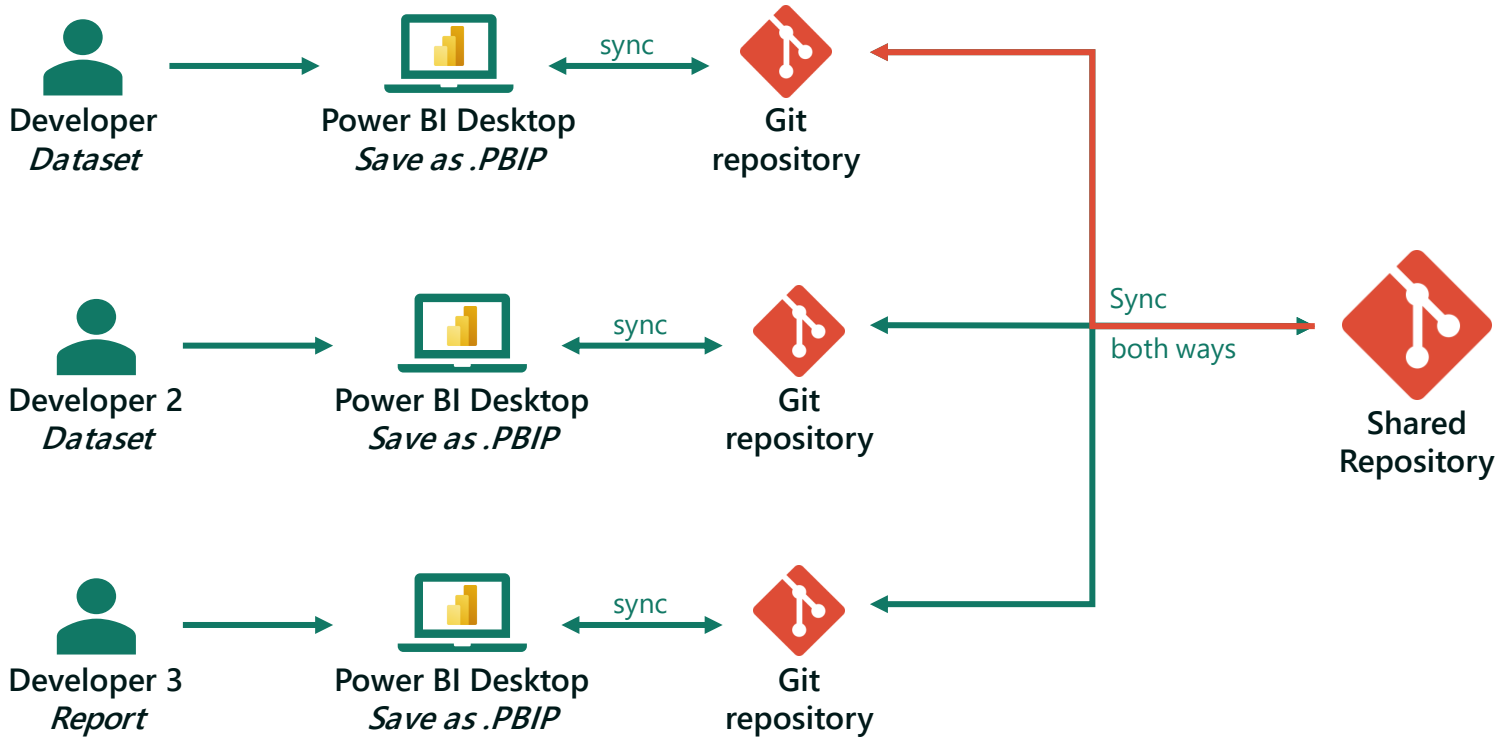
- PBIR file opens report and points to the dataset
- Report.json holds pages & visual definitions

Report.json

```
{
  "config": "{\"version\":\"5.42\",\"themeCollection\":{\"
  "filters": "[{\"name\":\"Filter\",\"expression\":{\"Hier
  "layoutOptimization": 1,
  "pods": [
    {
      "boundSection": "ReportSectiona37d01e834c17d07bbeb",
      "config": "{\"acceptsFilterContext\":1}",
      "name": "Pod",
      "parameters": "[{\"name\":\"Param_Filter\",\"boundFi
      "type": 1
    },
    {
      "boundSection": "ReportSection4b3fbaa7dd7908d906d9",
      "config": "{}",
      "name": "Pod1"
    },
    {
      "boundSection": "ReportSection79828a20a2b300a5d99b",
      "config": "{}",
      "name": "Pod10"
    },
    {
      "boundSection": "ReportSection23904f9ac2e638d5ed18",
      "config": "{}",
      "name": "Pod11"
    },
    {
      "boundSection": "ReportSection406bc05d6280a2c44873",
      "config": "{}",
      "name": "Pod12"
```

Development process

Collaborate with colleagues



Individuals sync (commit) their work into their own repository (branch)

Once ready, branches can be merged (push) in shared repository (main branch)

Users can sync updates (pull) back into their local repository

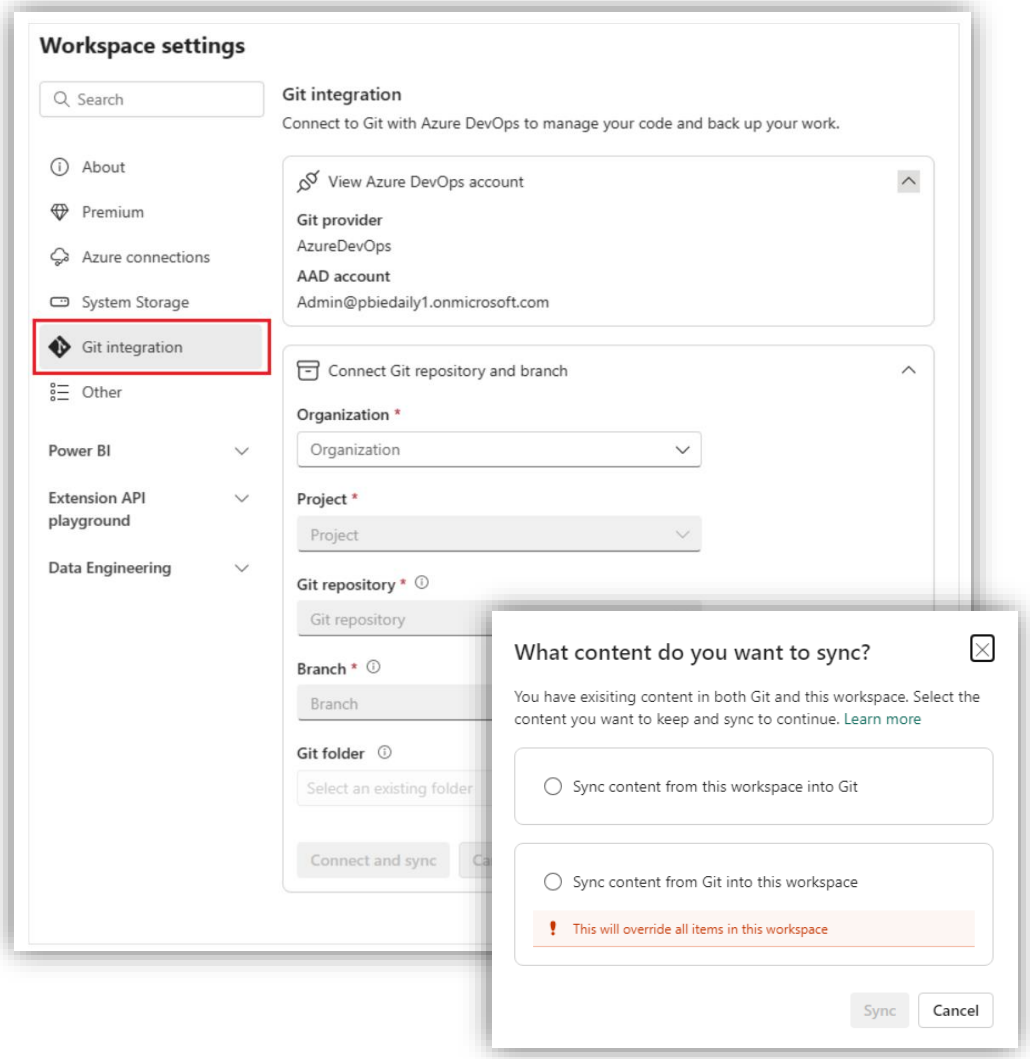
Demo

Developer Mode

Git integration (preview)

Fabric Git integration

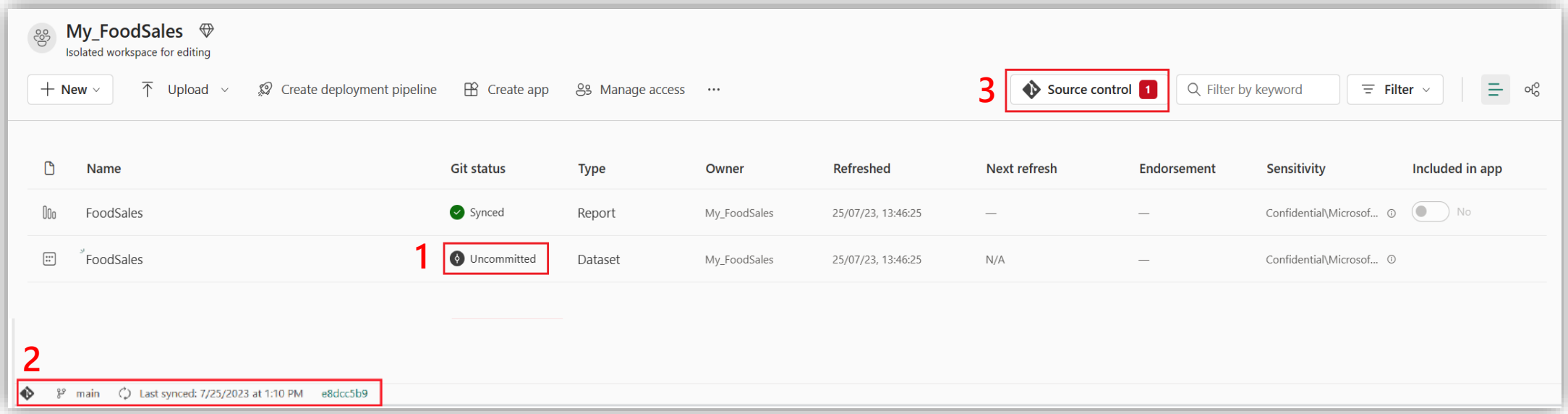
- Configured on workspace level
- Requires Premium workspace and Fabric enabled
- Currently only Azure DevOps
- Actions depend on workspace and ADO permissions
- Only syncs metadata; no “custom” subfolders
- Supports branch switching
- Supports bi-directional syncing
- Currently supports Reports, Paginated Reports and Datasets



Fabric Git integration

- Synced (the item is the same in the workspace and Git branch)
- Conflict (the item was changed in both the workspace and Git branch)
- Unsupported item
- Uncommitted changes in the workspace
- Update required from Git
- Item is identical in both places but needs to be updated to the last commit

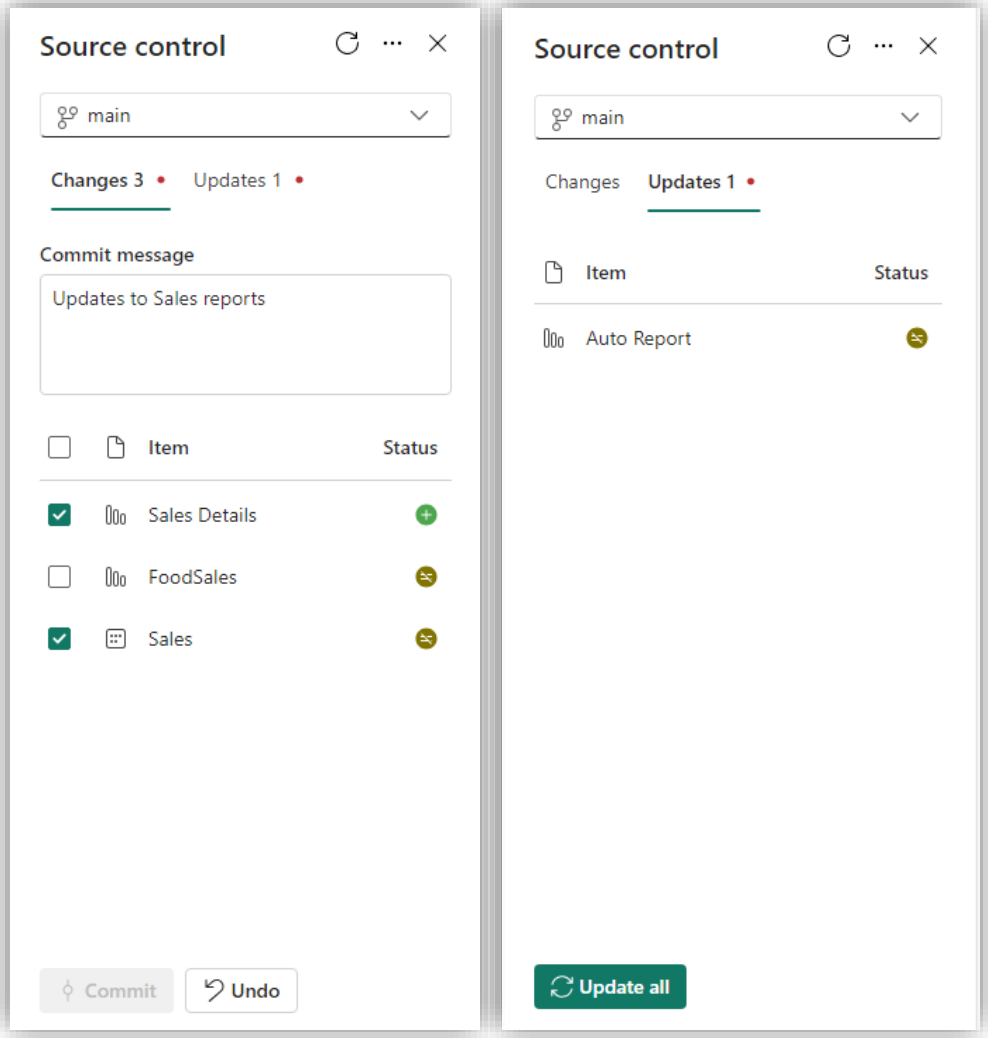
1. Status indicates status
2. Branch and sync timestamp are indicated on the bottom of the workspace
3. Source control shows number of pending changes allows to open action pane



Synchronize: Changes and Updates

Changes

- Shows changes in the workspace (web authoring)
- Option to input commit message
- Selective commit / undo possible
- Undo selected changes (revert to last commit)
- Requires to first apply updates



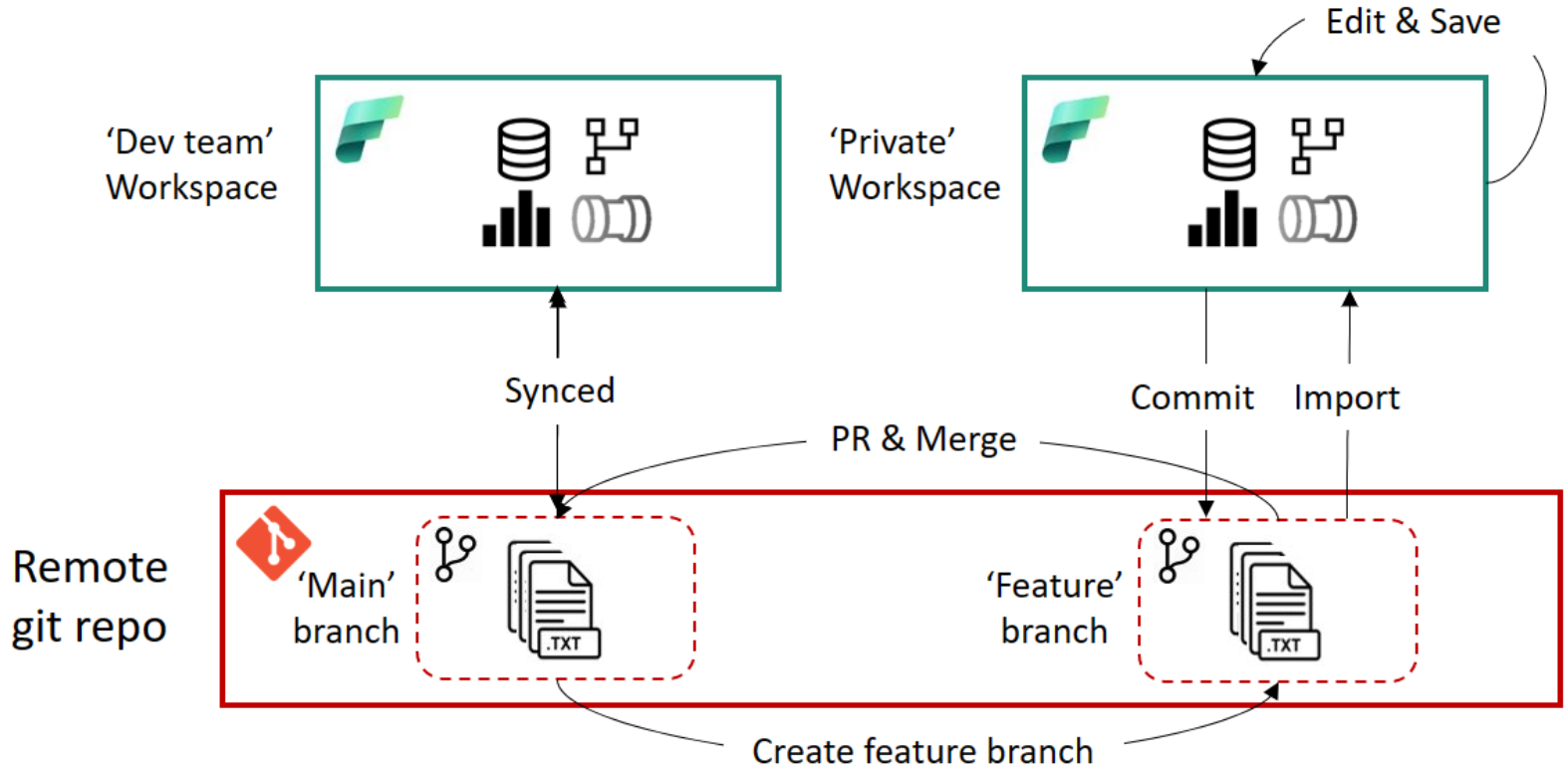
Updates

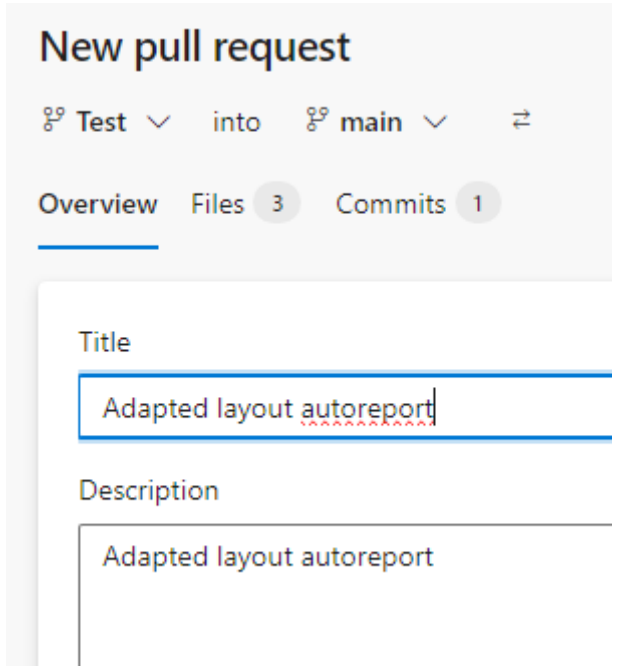
- Shows changes in the git repository
- Allows to update workspace based on connected repository (metadata only)

new
modified
deleted
conflict

Manage branches

- For local development: Each developer has local ‘feature’ branch through IDE
- For web development: Each developer has their own workspace and ‘feature’ branch
- After commit, a pull request in ADO merges the changes with the ‘main’ branch



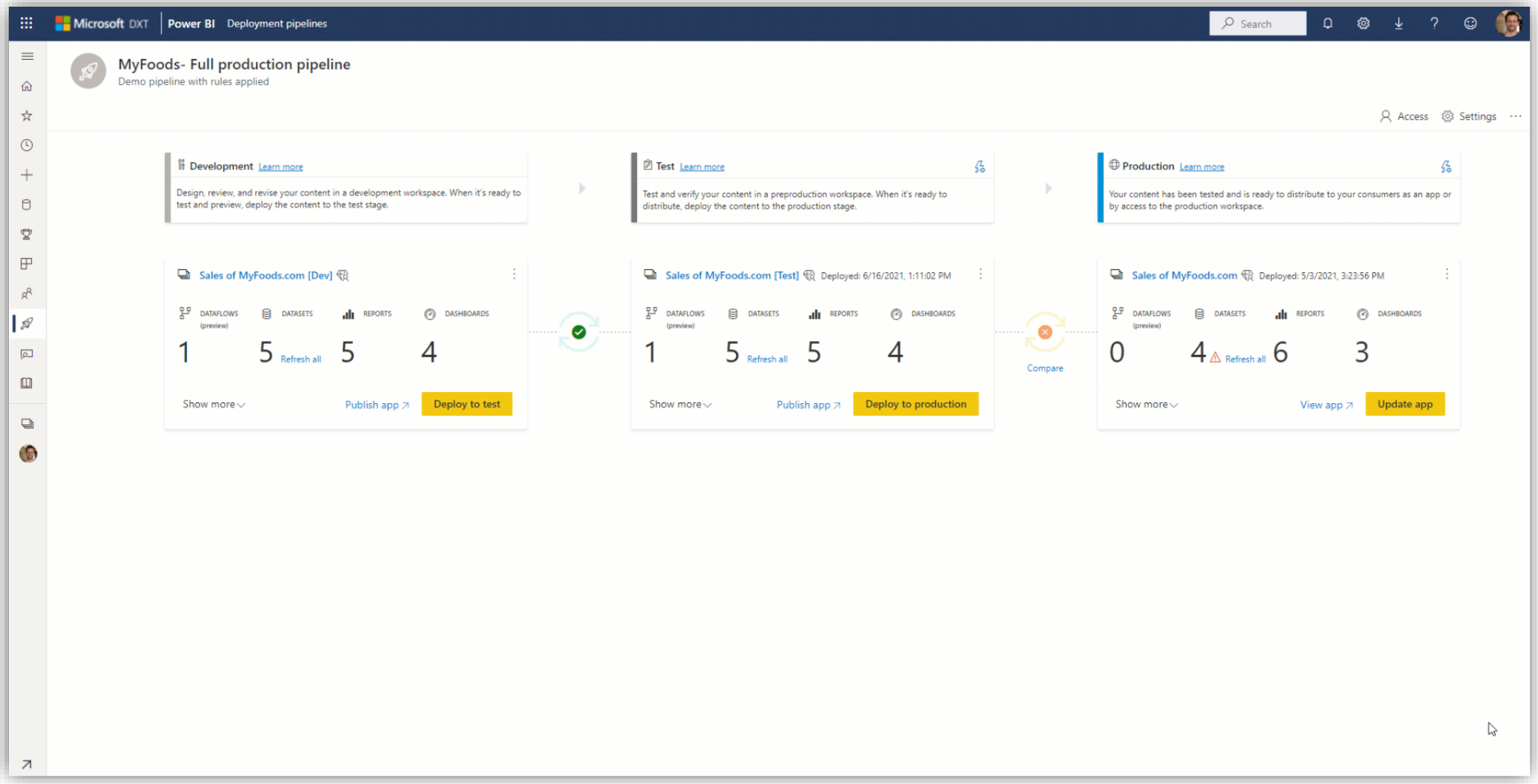


Demo

Git integration

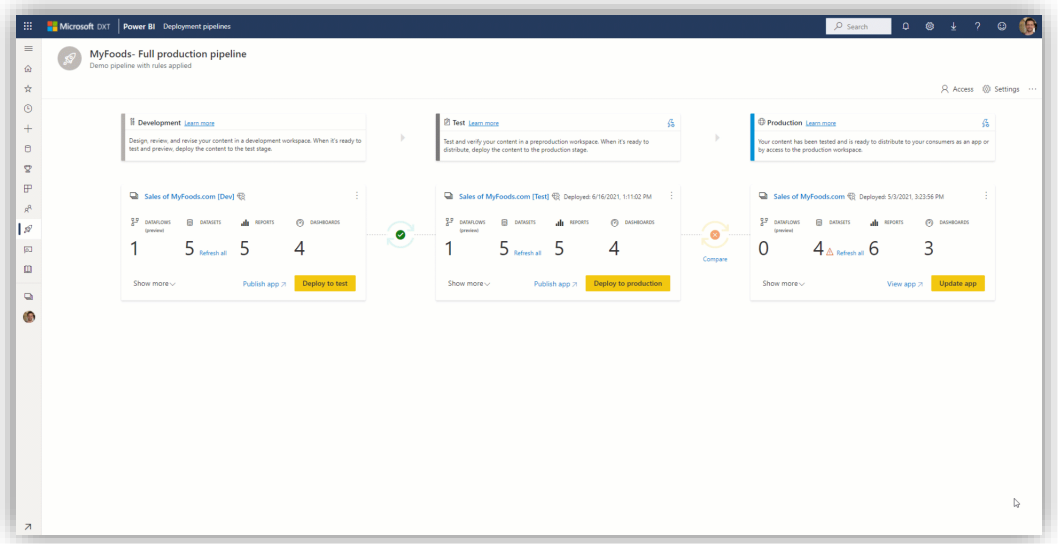
Deployment Pipelines

What are deployment pipelines?

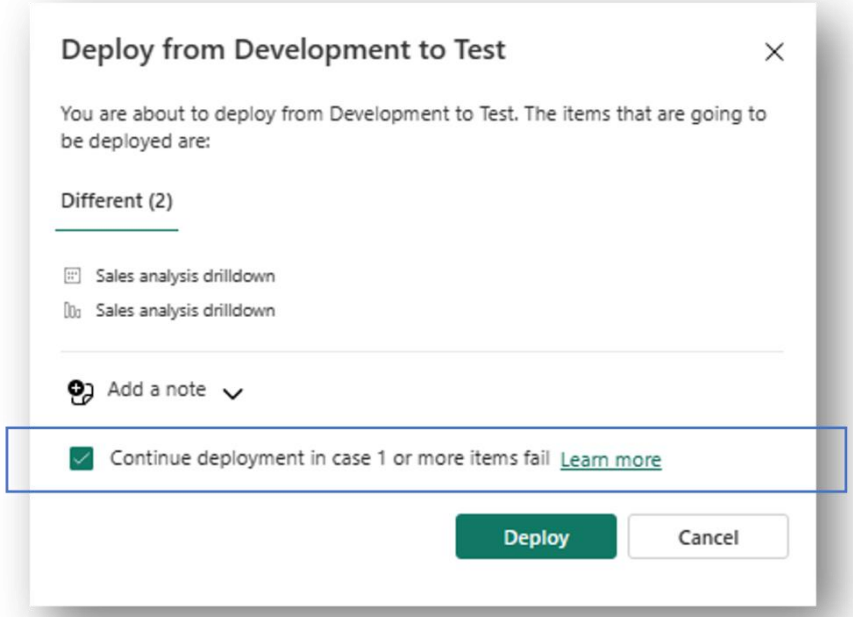


What are deployment pipelines?

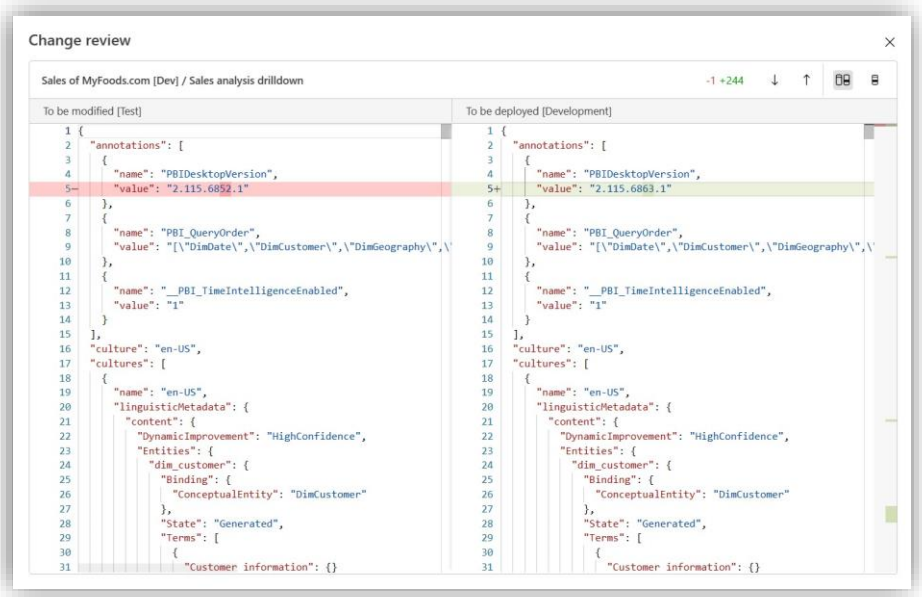
- Three different stages (Development, Test, Production)
- Easily copy items between stages (selective deploy)
- Auto-binding between stages
- Only copies metadata
- Different parameter / data sources can be configured in each stage
- Different access rights can be configured in each stage



Deployment Pipeline recent improvements



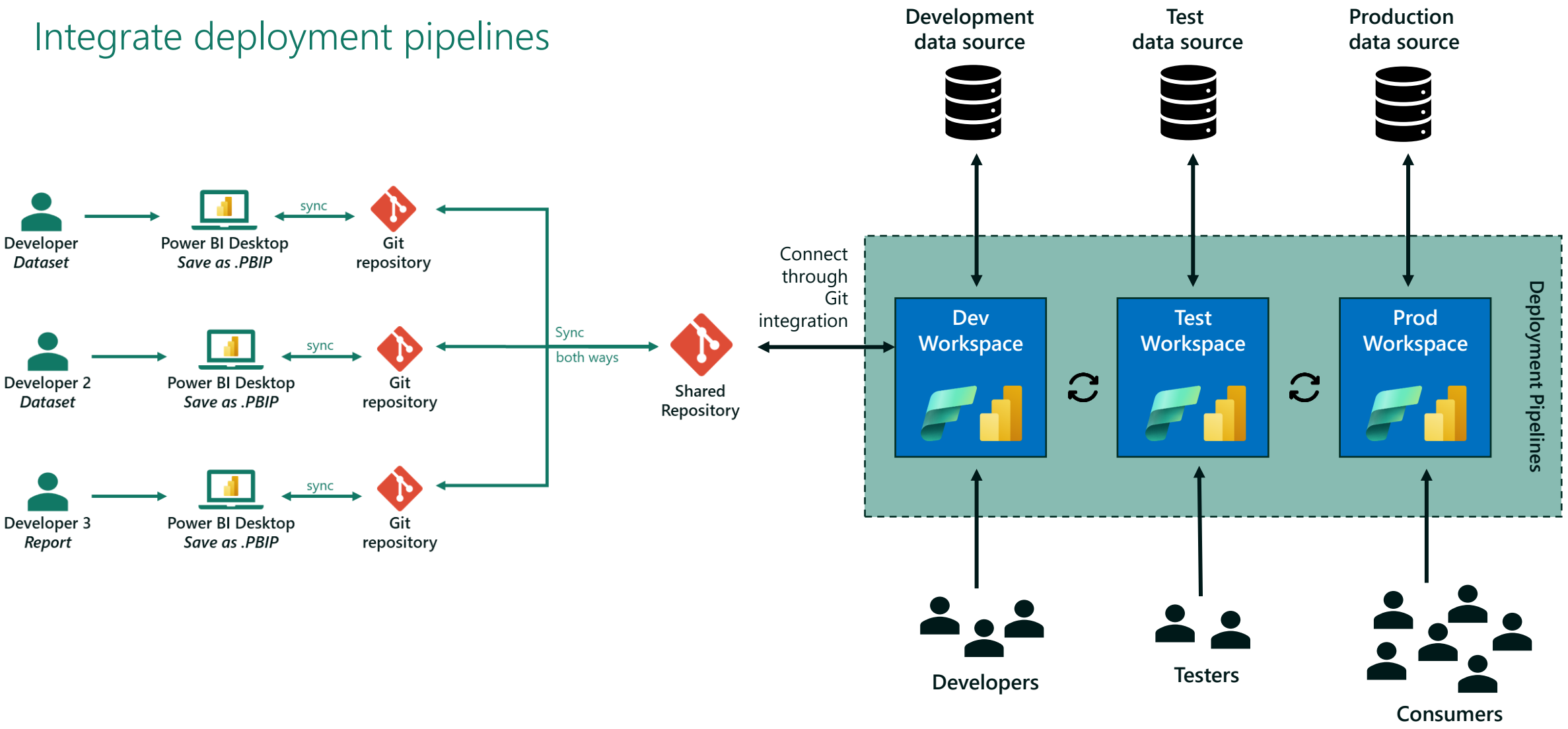
- Deployment note
- Continue deployment in case of errors



- Code change review for Datasets and Dataflows

Development process

Integrate deployment pipelines

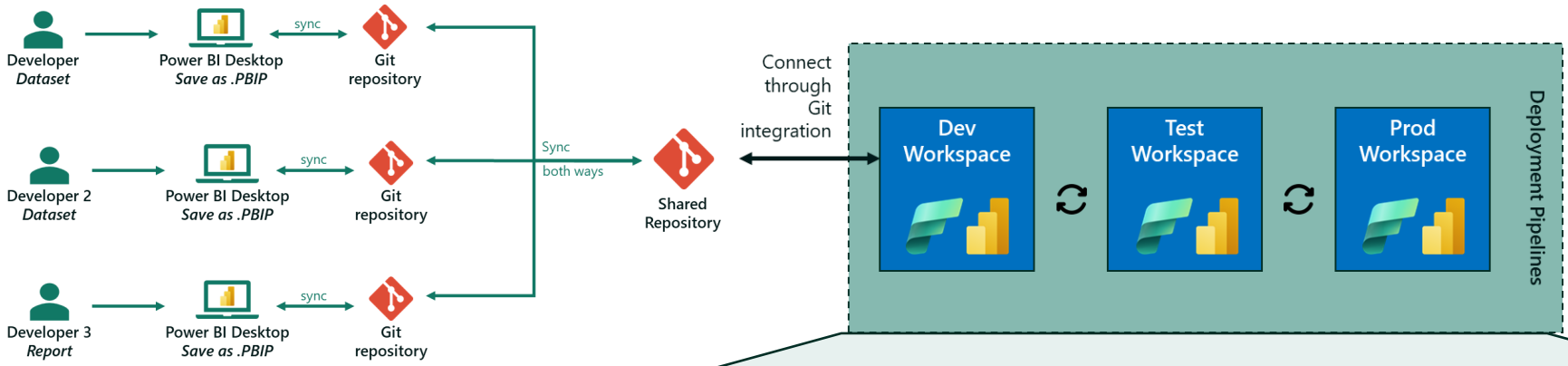


Demo

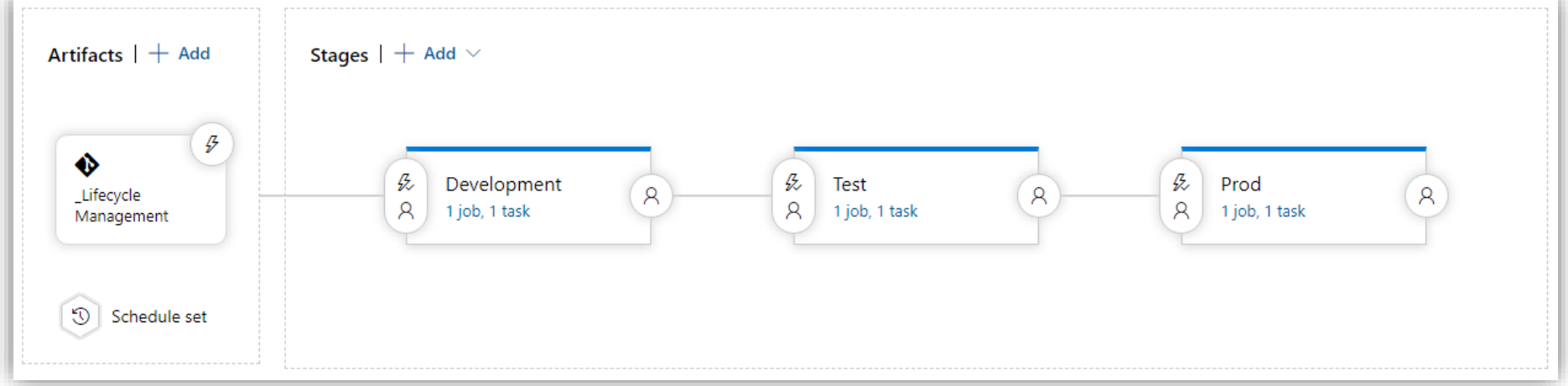
Deployment Pipelines

Development process

Leverage the power of Azure DevOps



Power BI automation tools (ADO Extension), Fabric APIs and external tools



Azure DevOps

- Release Pipelines
- Approvals
- Automate stages
- Automate tests
- Automate refresh
- Leverage external tools
- …

Examples of external tools (non-exhaustive)

TABULAR EDITOR

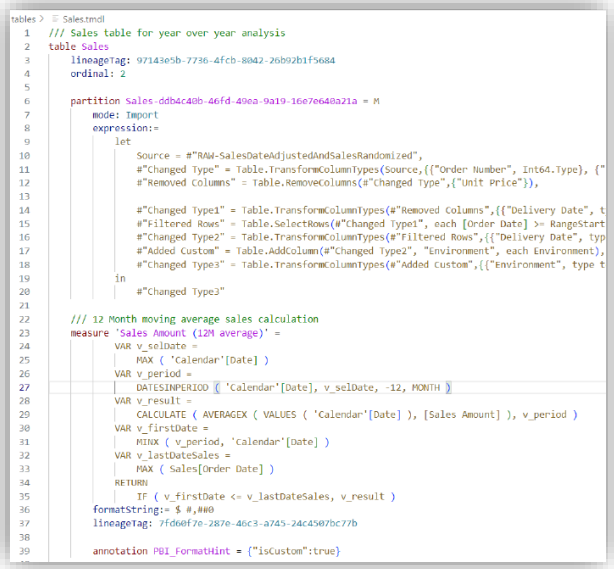
Best Practice Analyzer
by Michael Kovalsky

pbi-tools

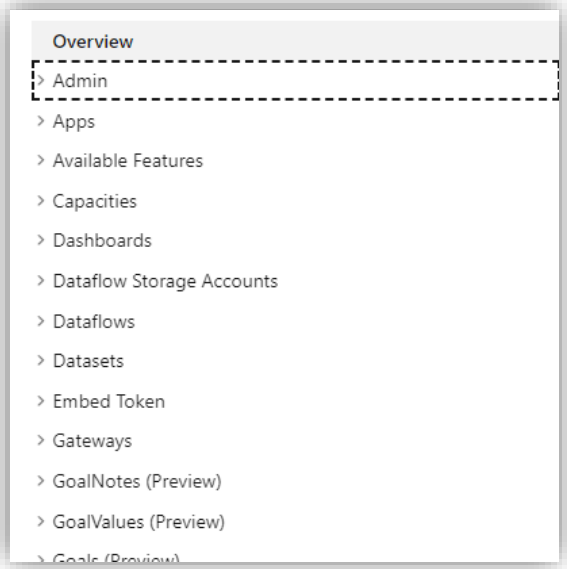
Azure DevOps Extensions

Lookout – What is coming?

TMDL: Tabular Model Definition Language



Fabric APIs



New Report definition format

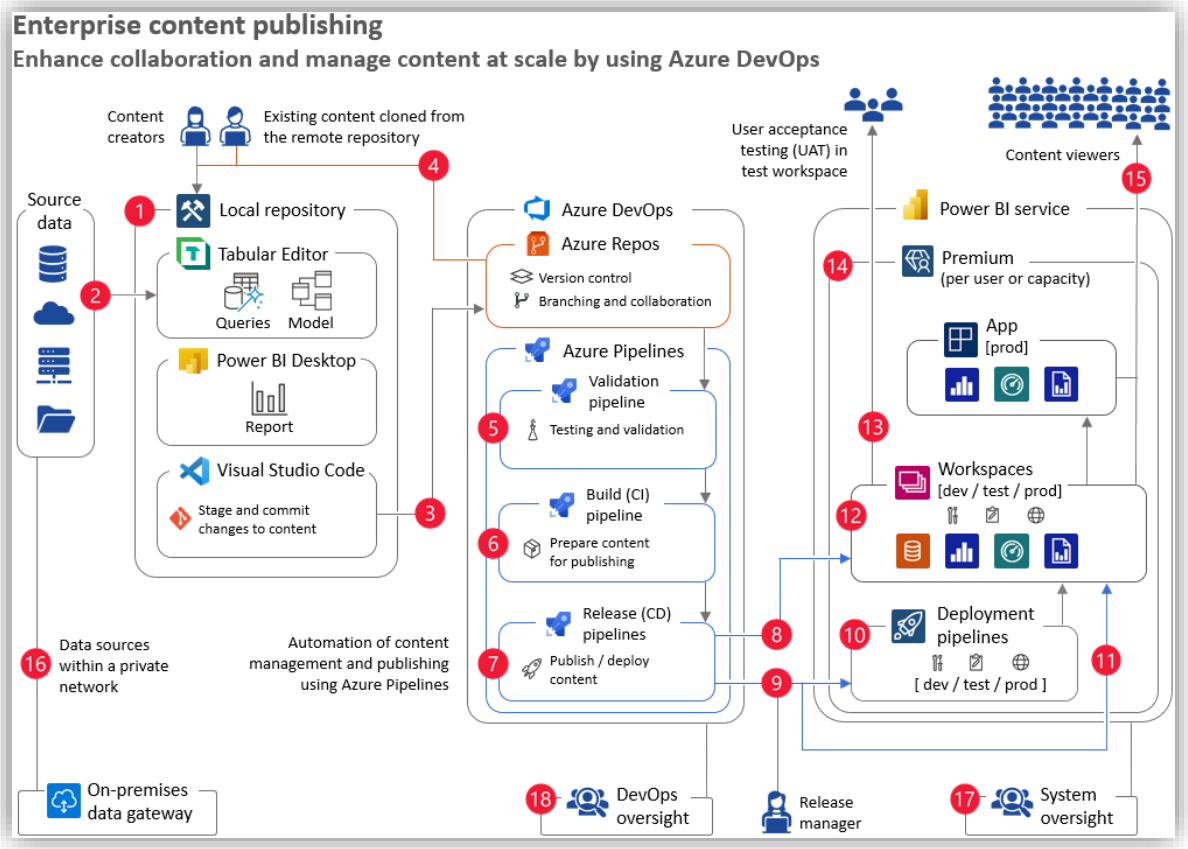
Additional resources

Documentations

- Microsoft Fabric's lifecycle management tools enable efficient product development, continuous updates, fast releases, and ongoing feature enhancements
- Power BI Desktop developer mode documentation

Implementation planning guidance

- Power BI usage scenarios: Self-service content publishing
- Power BI usage scenarios: Enterprise content publishing



Microsoft

Icon links to blogs and documentation

Thank you