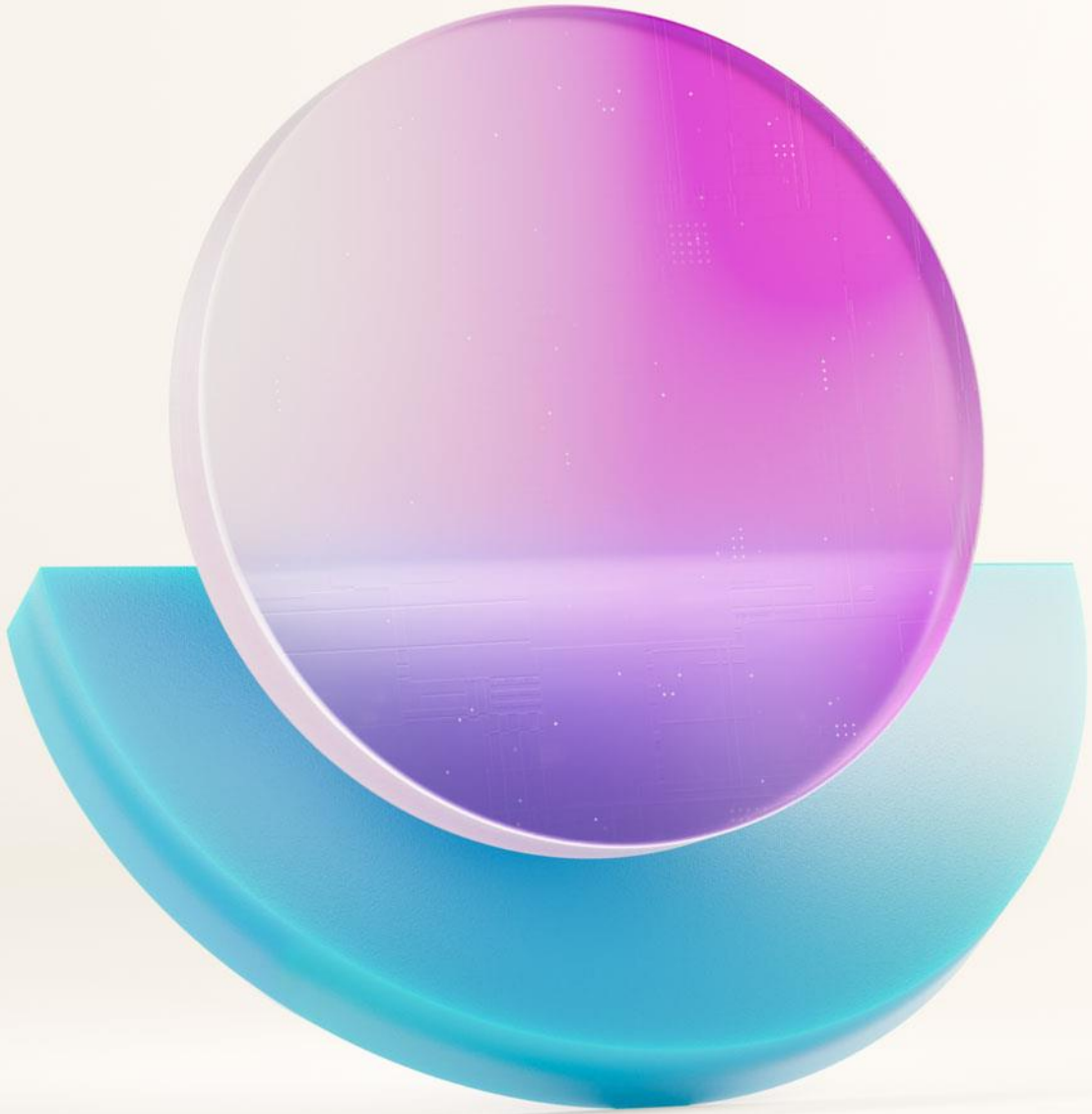
Microsoft

Microsoft Fabric Community Conference

Microsoft Fabric
Community Conference



Fast-track your career in data and AI

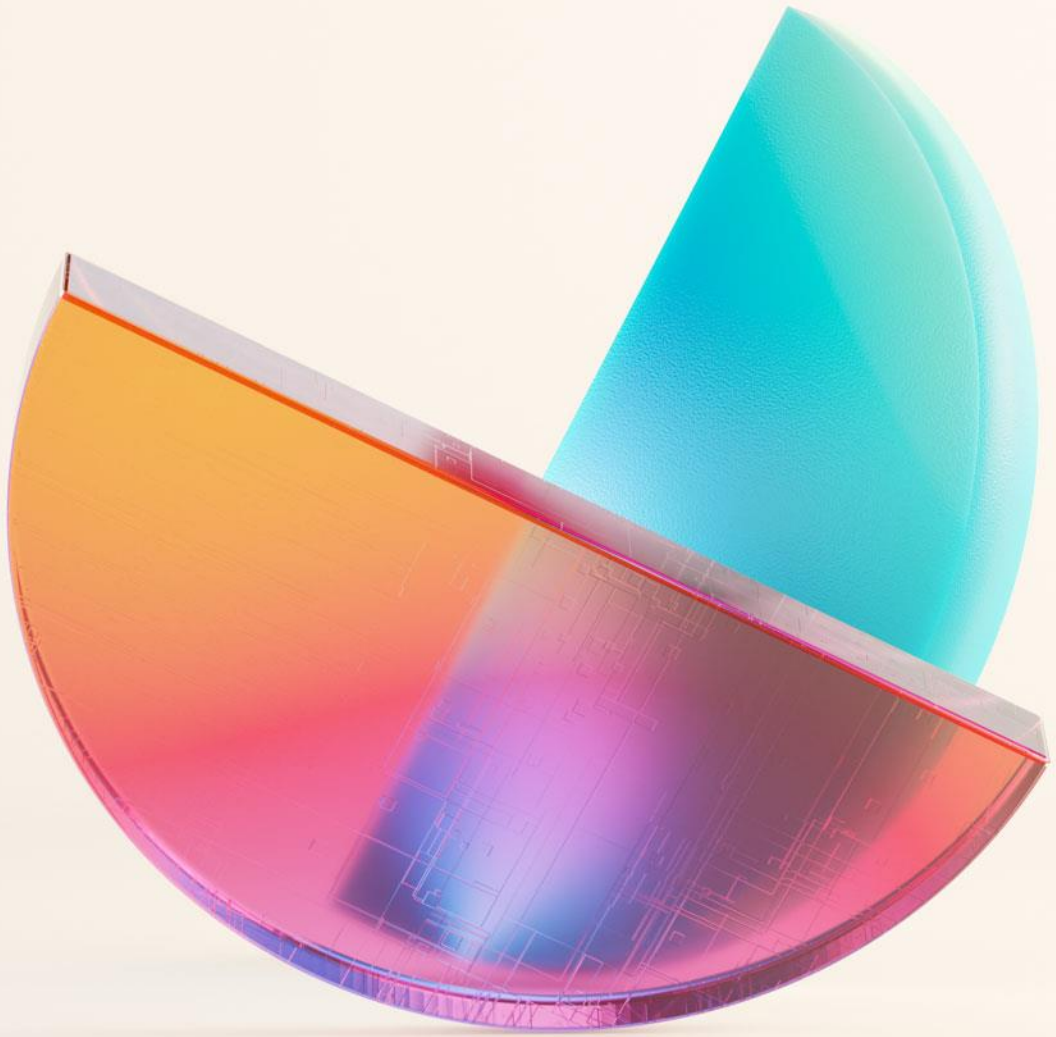
Become a Microsoft Certified Fabric Data Engineer Associate – take Exam DP-700 for free!

Visit the Fabric Community Lounge to learn more about this *limited-time offer*.

aka.ms/fabcon/dp700

Microsoft

Microsoft Fabric
Community Conference

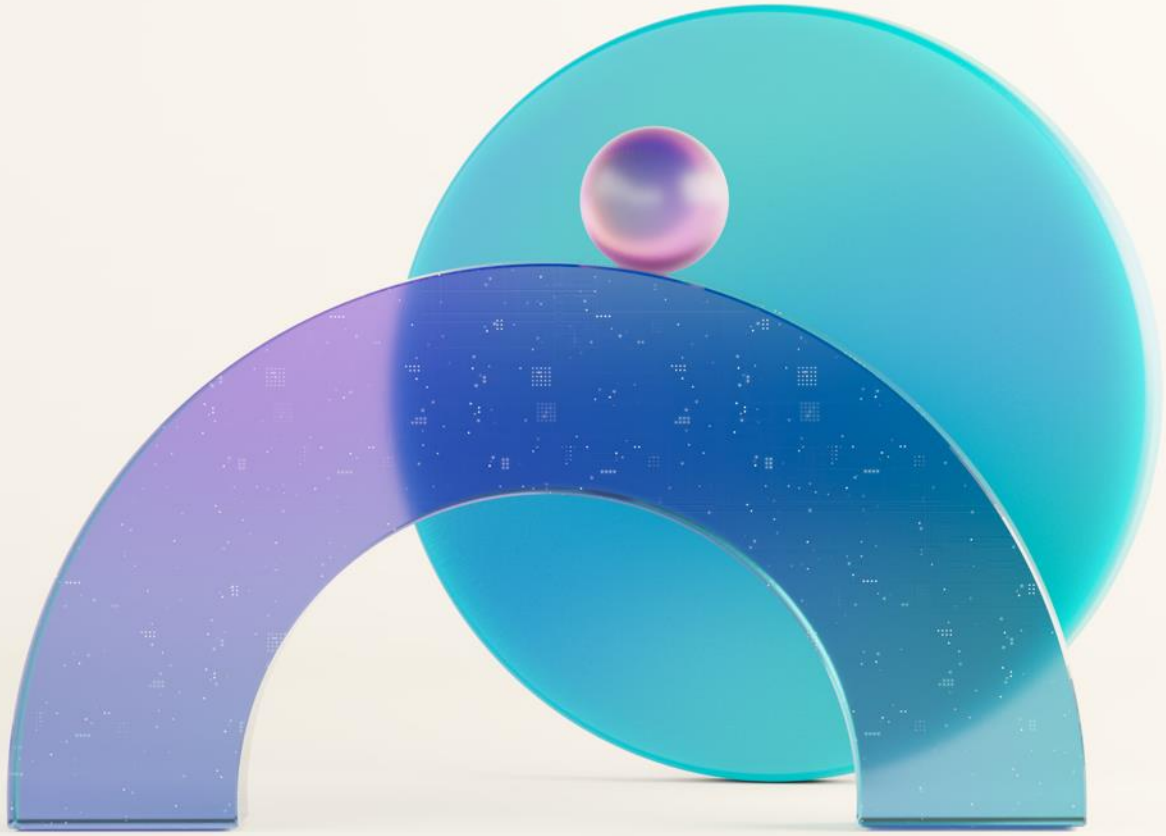


Learn more about Microsoft Fabric

Power your AI transformation with a complete data platform

Microsoft

Microsoft Fabric
Community Conference



From Setup to CI/CD

Automating Microsoft Fabric for Scalable Data Solutions

Who am I?

Peer Grønnerup

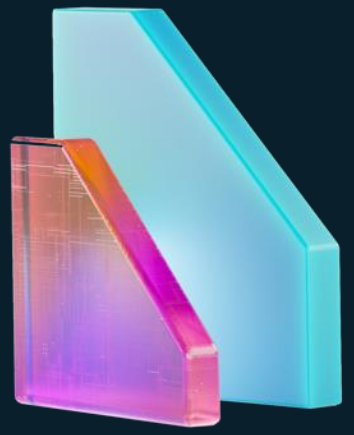
Principal architect & Consulting manager

twoday, Data & AI, Denmark

https://www.linkedin.com/in/peergroennerup/

https://peerinsights.hashnode.dev/

twoday



Download showcase code and presentation

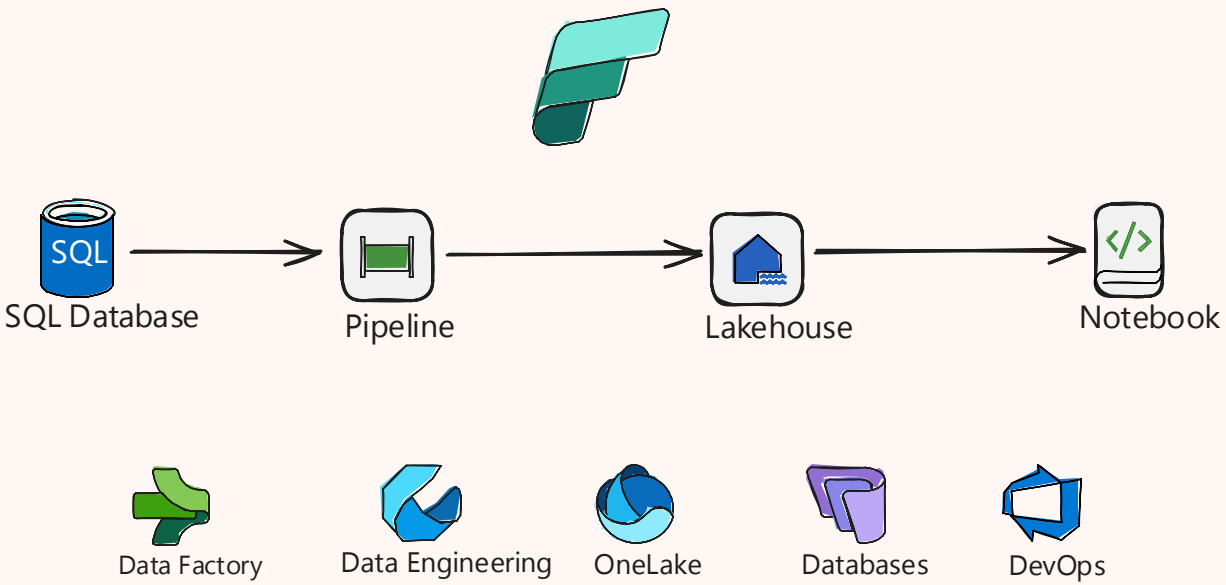
https://github.com/gronnerup/Fabric

Disclaimer:
The solution demonstrated in this session is **experimental** and provided for **showcasing purposes only**. It is **unsupported**, and there are no guarantees regarding functionality or future updates. You are free to use it **as-is** or modify it to fit your needs. Use at your own risk.

What we cover – and what we do not cover…

We will dive into…

- The Fabric REST APIs
- A Fabric Lakehouse architecture starting point
- Solution setup (IaC)
- How to utilize a metadata driven framework and ways-of-working
- Enterprise CI/CD with Azure DevOps



But we will not have time for…

- Semantic models and Power BI reports
- The Fabric Terraform Provider in details
- Fabric Deployment pipelines
- Branching strategies, security…

Fabric lifecycle management – The 4 phases covered in this session

Design

Architecture

Functional layers

Fabric components/items

Storage tiers

Setup

Infrastructure-as-code

Operate

Continuous integration and continuous delivery

Build

Workload automation

Ways-of-working best practices

Metadata-driven ingestion and transformation

Automating Fabric with Fabric REST APIs

For efficency, consistency and scale

Setup

- Setup Fabric infrastructure
- Workspace management
- Item management
- Governance and security
- Automate documentation
- And much more…

Build

- Support ways-of-working
- Automate workloads
- Automate maintaince
- And much more…

Operate

- Deploy items
- Monitoring and alerting
- Logging
- And much more…

Automating Fabric with Fabric REST APIs

Fabric Core

Managing the core operations of common service entities

- Capacities
- Workspaces
- Items
- Deployment
- Jobs
- Shortcuts
- Connections
- Git
- Private Endpoints
- Gateways
- LRO
- Onelake

Fabric Workloads

Data processing and management of workload/item specific capabilities

Fabric Admin

- Tenants
- Users
- Workspaces
- Items
- Domains
- Labels

https://learn.microsoft.com/en-us/rest/api/fabric/articles/

Fabric REST APIs and identity support

Most Fabric REST APIs support SPN and Managed Identities

- Core: Workspaces, Capacities, Connections etc.
- Power BI: Reports & semantic models
- Data Engineering: Lakehouses, Notebook etc.
- RTI: Eventhouse, Eventstream, KQL Querysets etc.
- Data Pipelines and others (added a few weeks ago)!

Identity	Support
User	Yes
Service principal and Managed identities	Yes

But - We still miss a few like...

- Core: Git integration, Job Scheduler,
- Items: Reflex, MLModel and MLExperiment

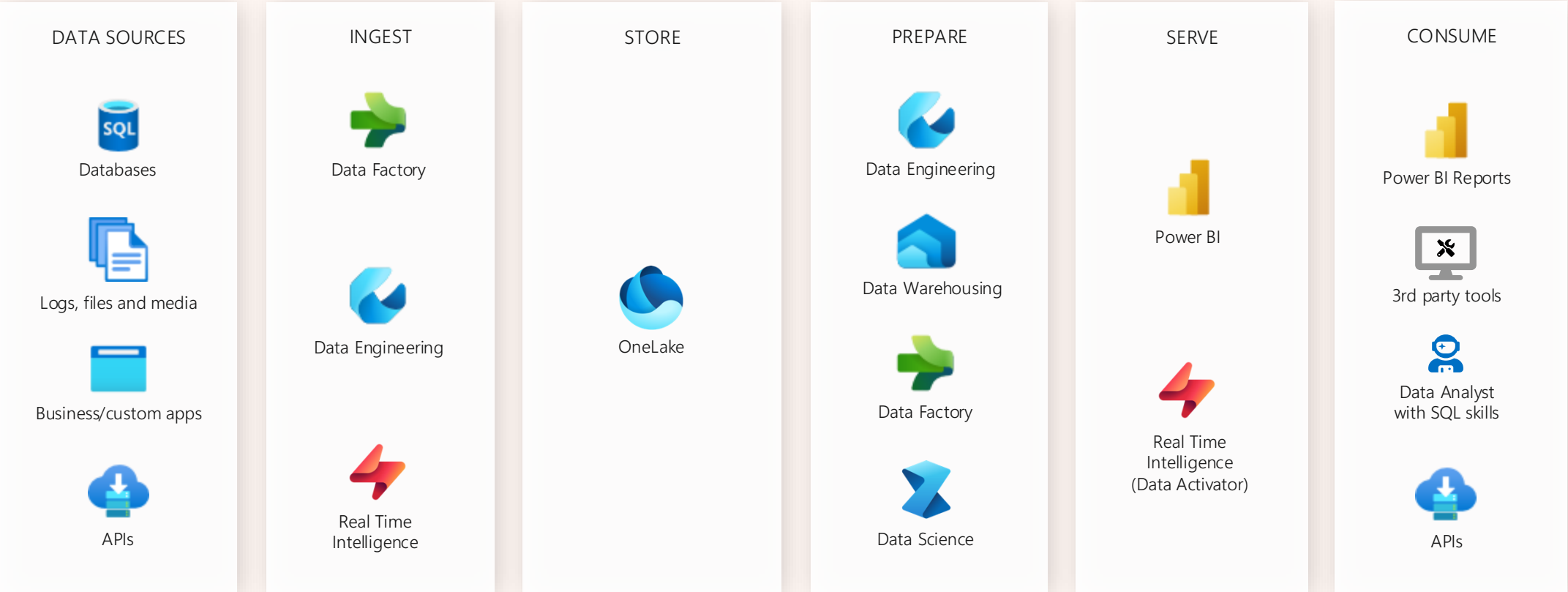
Identity	Support
User	Yes
Service principal and Managed identities	No

Visit my blog post **"Automating Microsoft Fabric: Extracting Identity Support data"** at https://peerinsights.hashnode.dev

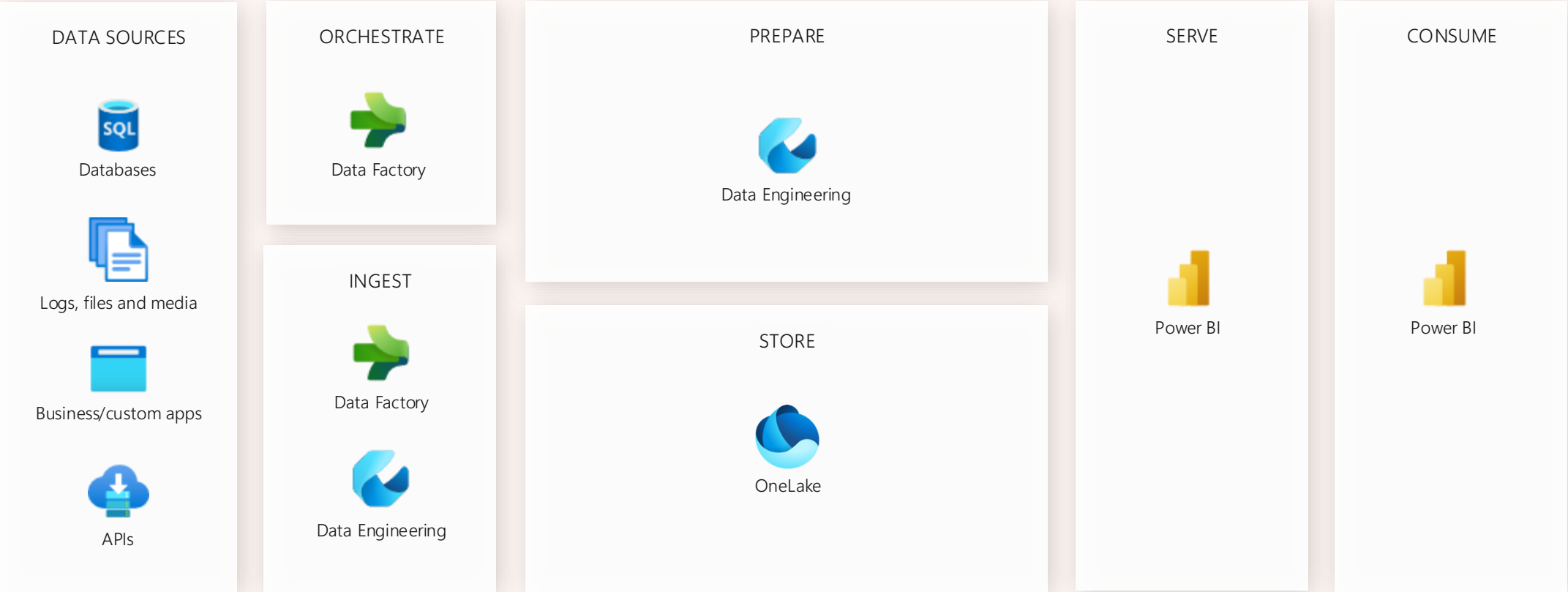
Microsoft Fabric
Community Conference

Designing our Fabric Lakehouse Platform

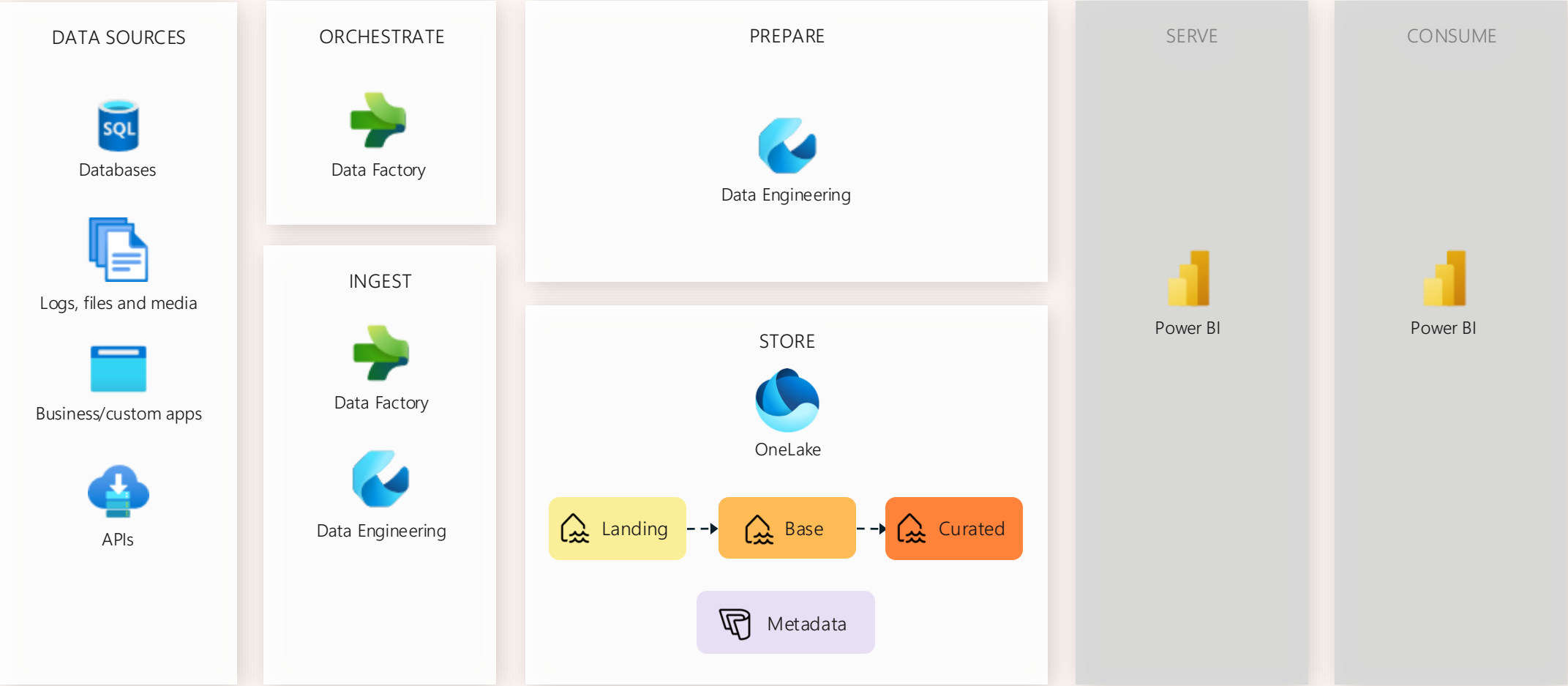
Multiple components = multiple possibilities



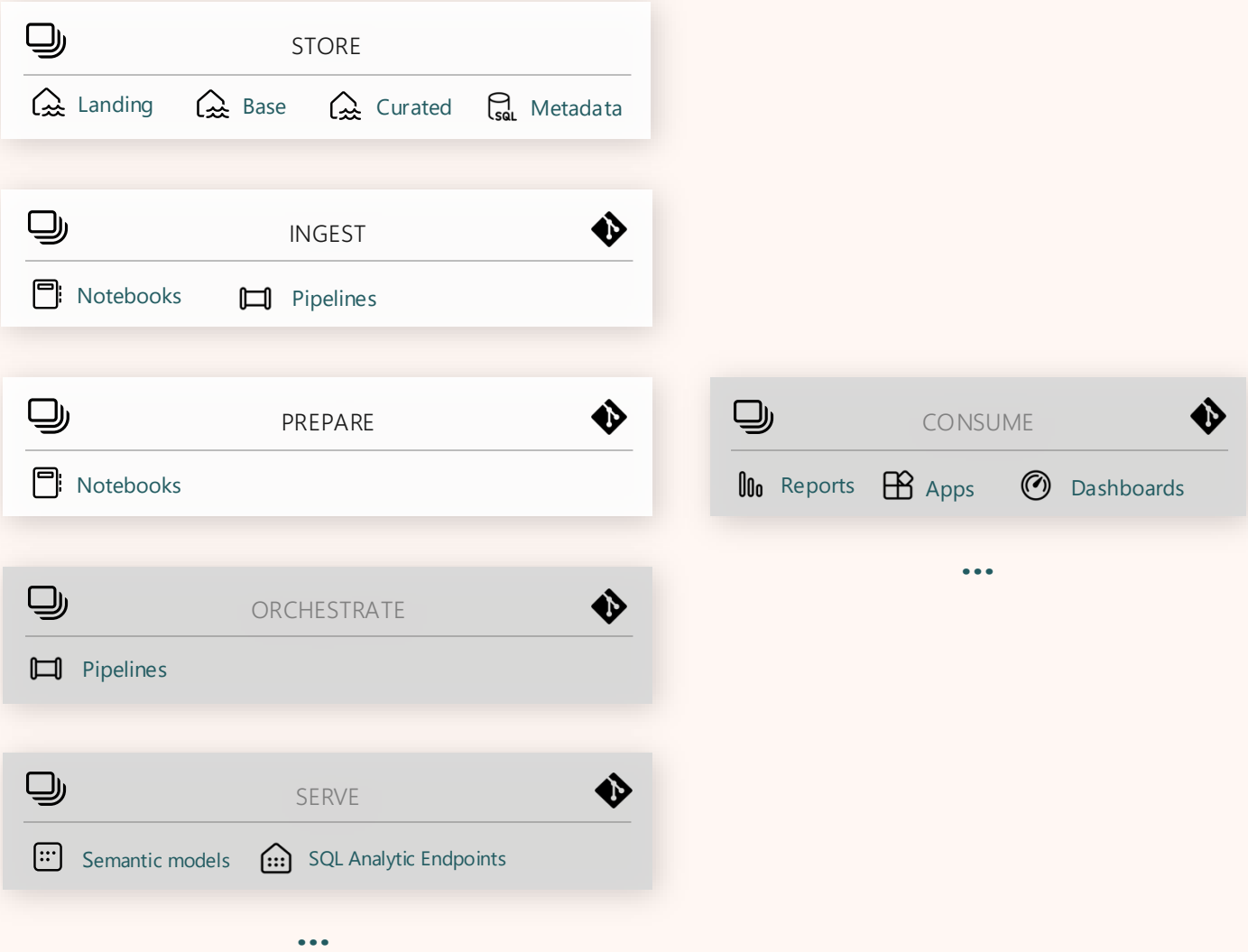
A Lakehouse starting point architecture



A Lakehouse starting point architecture



Fabric Workspace structure



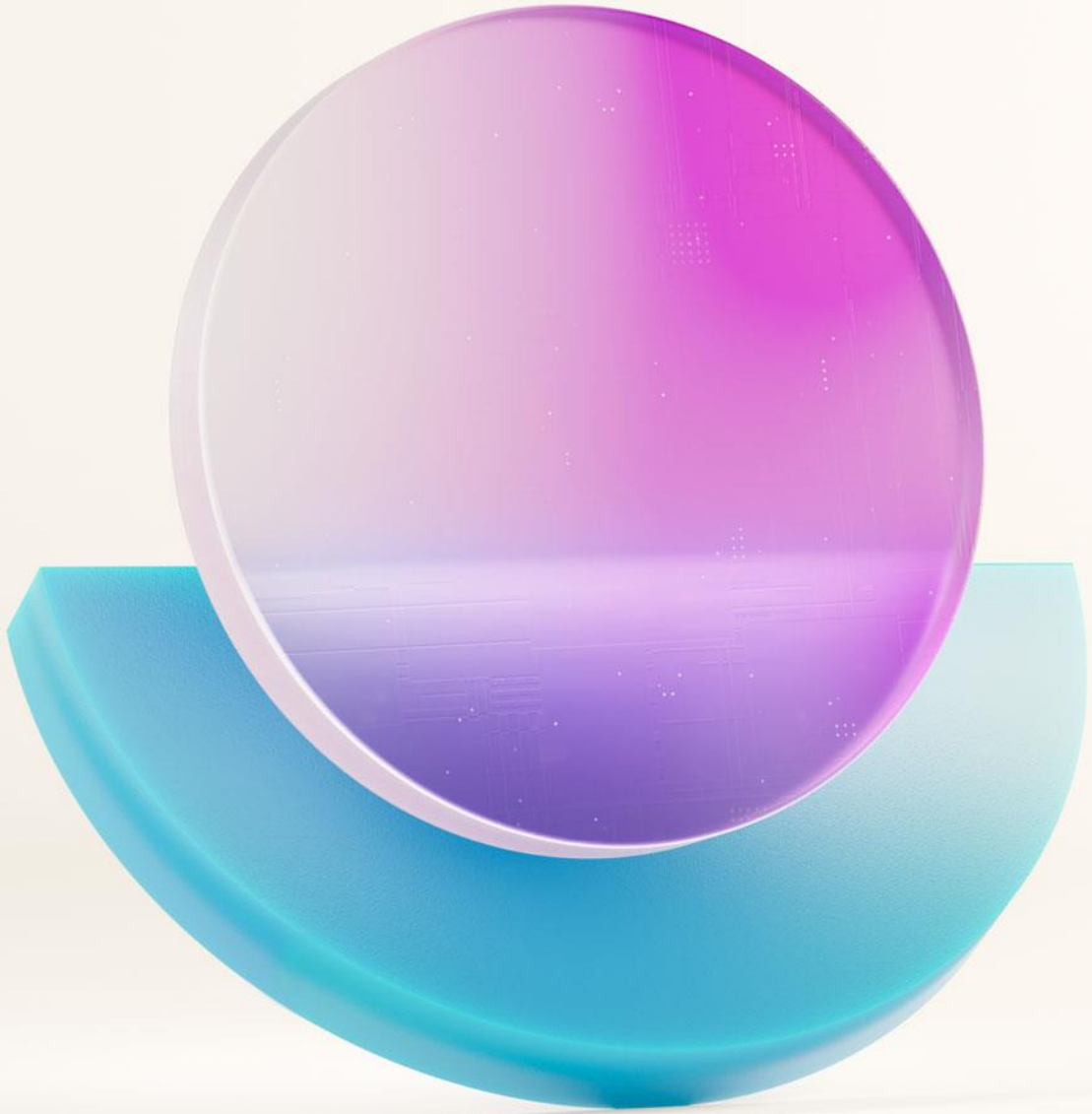
Layer seperated workspace pattern

- Item organization
- Enhanced access control
- Support capacity seperation

Git integration

- Single repository
- Layer specific folders
- Additional folders for automation, documentation etc.

Microsoft Fabric
Community Conference



Accellerate and unify Fabric solution setup

Automating Fabric solution setup - Showcase

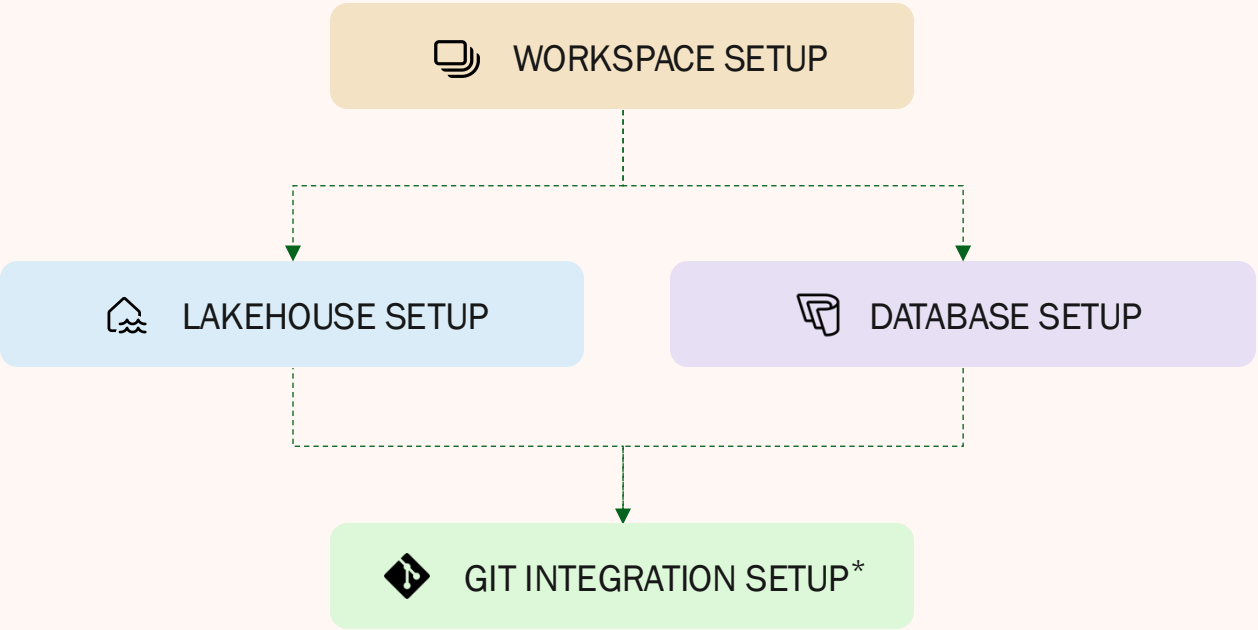
Prerequisites:

- A Fabric Capacity
- A Service Principal

Run options:

- Azure DevOps Pipeline
- Local Python script

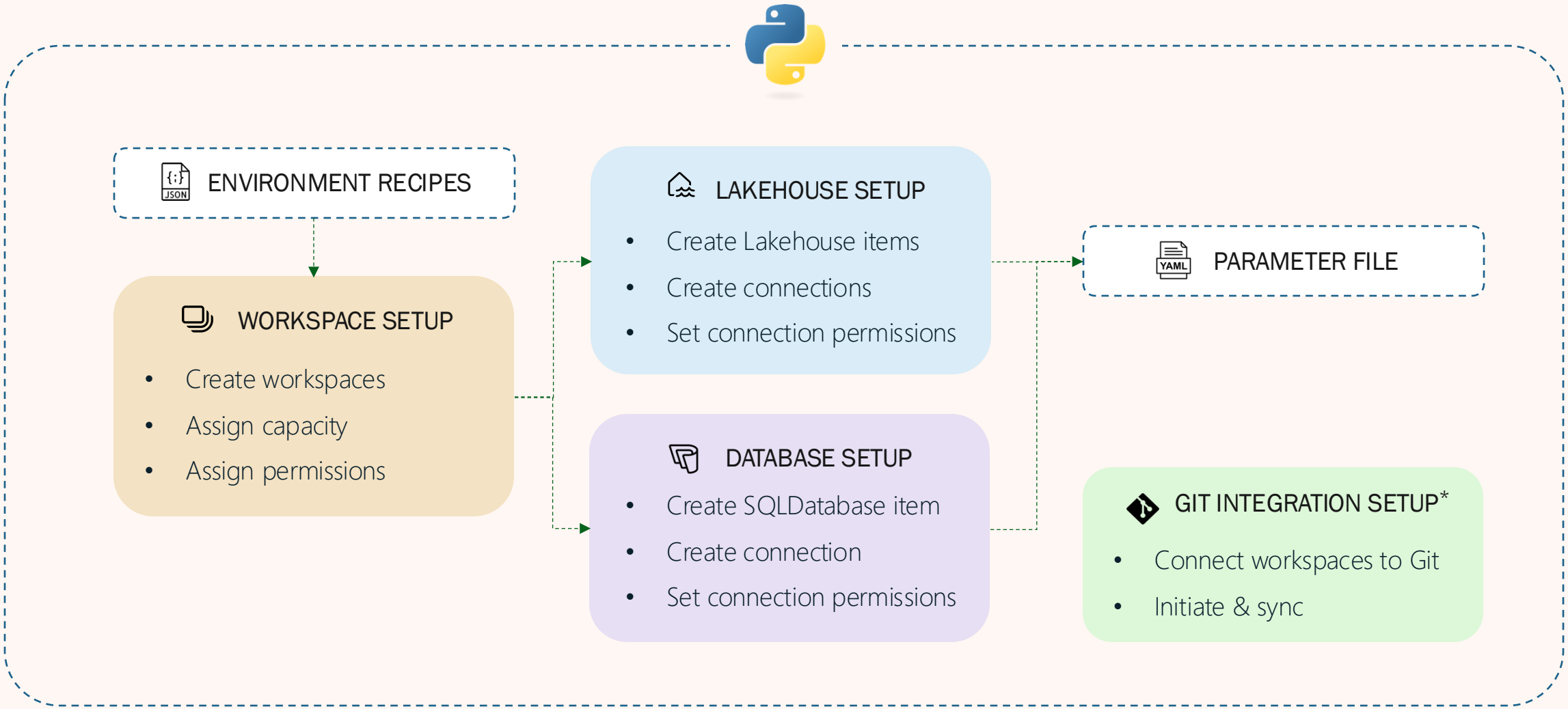
Infrastructure is declared using global recipe file and environment specific definition files.



Download source code: https://github.com/gronnerup/Fabric

Blog post "Automating Fabric: Kickstart your Fabric Setup with Python and Fabric REST APIs": https://peerinsights.hashnode.dev

Automating Fabric solution setup - Showcase



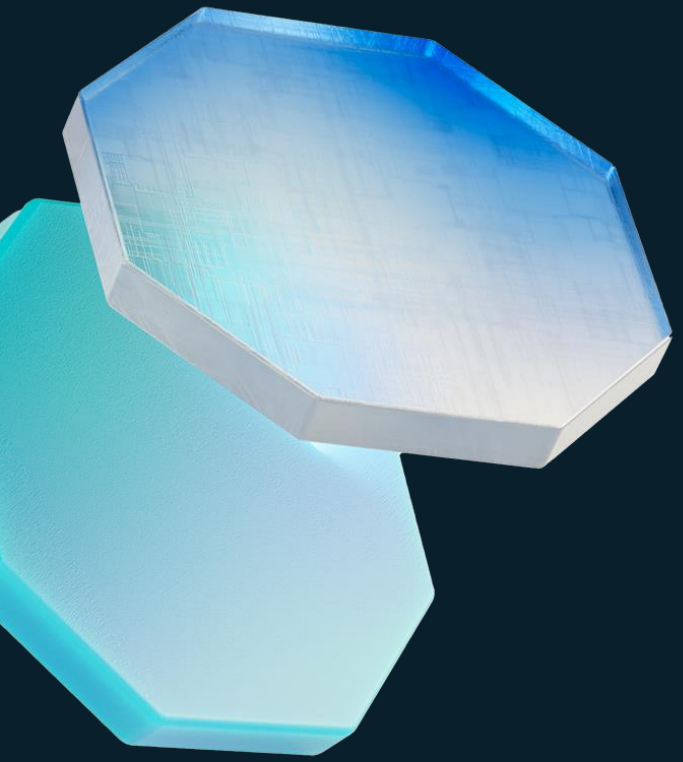
Using a recipe-based approach to infrastructure-as-code

BASE RECIPE

```
{
    "name": "*FabCon - {layer} [{environment}]",
    "generic": {
        "capacity_id": ”00000000-0000-0000-0000-000000000000",
        "permissions": { ... },
        "git_integration": { ... }
    },
    "layers" :
    {
        "Ingest": { },
        "Store": {
           "items": {
                "Lakehouse": [
                    {"item_name": "Landing", "connection_name": "FabCon-Landing [{environment}]"},
                    {"item_name": "Base", "connection_name": "FabCon-Base [{environment}]"},
                    {"item_name": "Curated", "connection_name": ”FabCon-Curated [{environment}]"}
                ],
                "SQLDatabase": [
                    {
                        "item_name": "Metadata",
                        "connection_name": ”FabCon-MetadataDB [{environment}]",
                        "sql_script": "../resources/Metadata.sql"....
```

RECIPE FILES

- infrastructure.dev.json
- infrastructure.json
- infrastructure.prd.json
- infrastructure.tst.json



Demo

Quick overview of the main setup options

REST APIs

Utilizing Fabric REST APIs

Pros
Flexible and scalable
Fully customizable
Integrates with DevOps
Improves governance
Better resource control

Cons
Higher inital setup effort
Possible API coverage gaps

ClickOps

Manually through the Fabric UI.

Pros
Quick and easy for tiny setups
Suitable for testing

Cons
Not scalable
Error-prone
No version control
Not CI/CD friendly

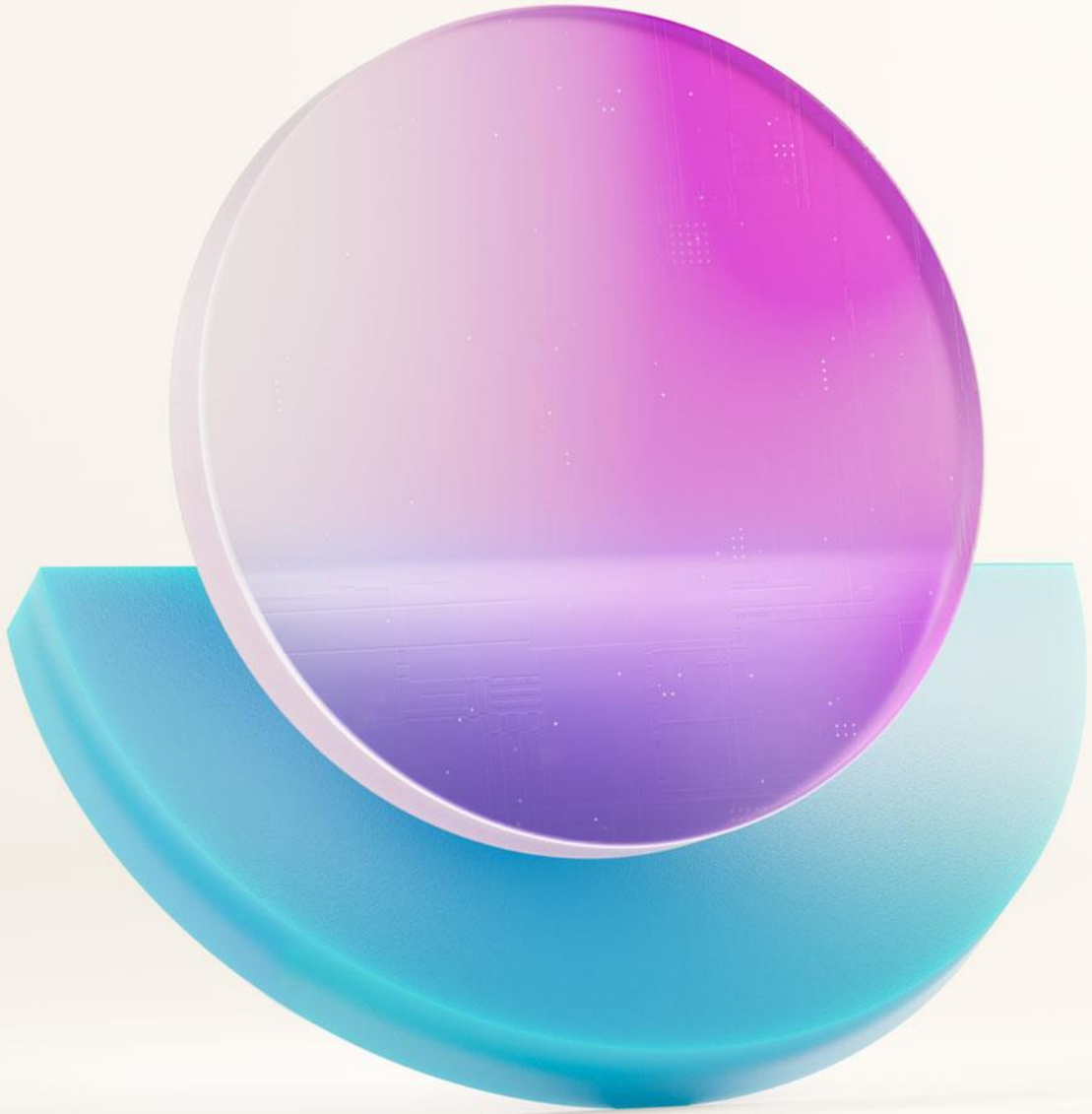
Terraform

Leading IaC tool!

Pros
Industry standard
Readable and scalable
Ensures consistent deployment
Declarative & State Management
Multi-cloud / cross-service

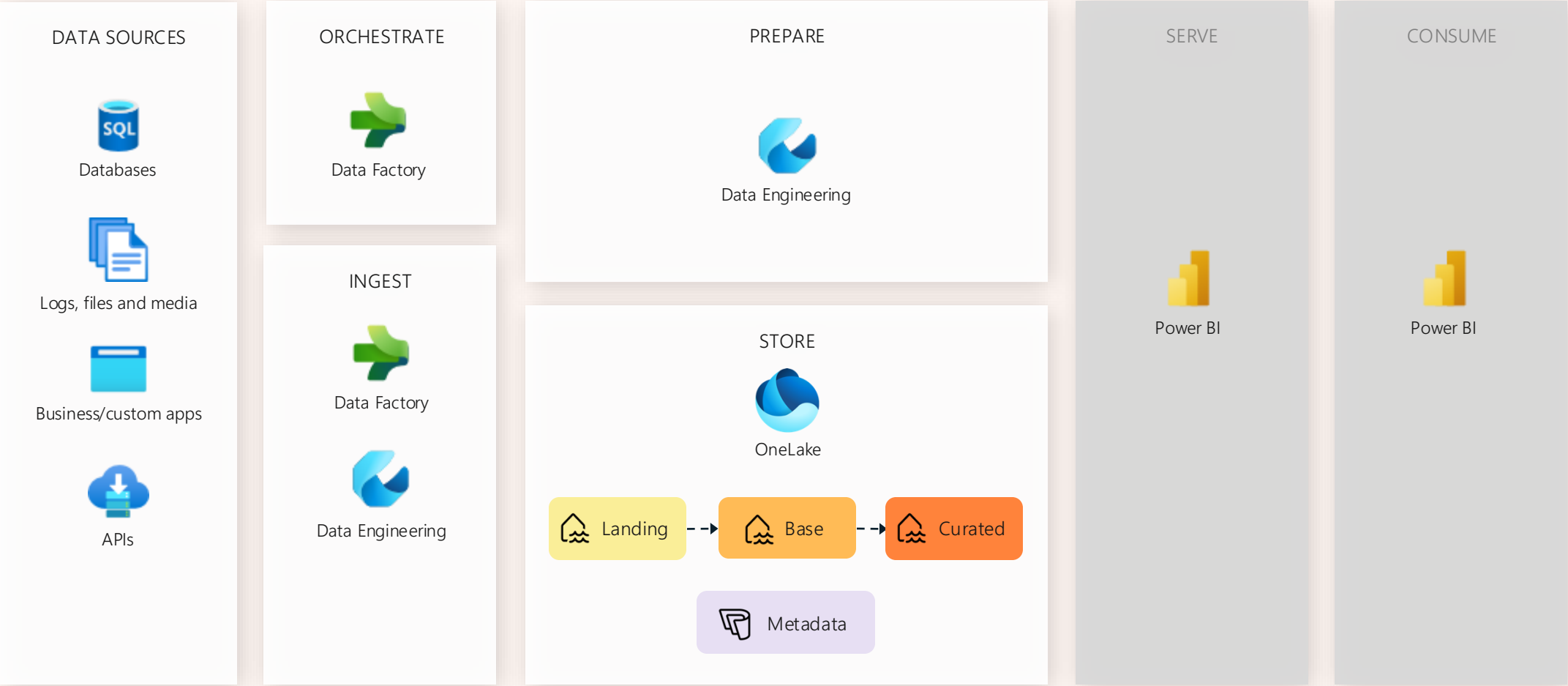
Cons
Can be complex
Would still require a hybrid-setup

Microsoft Fabric
Community Conference

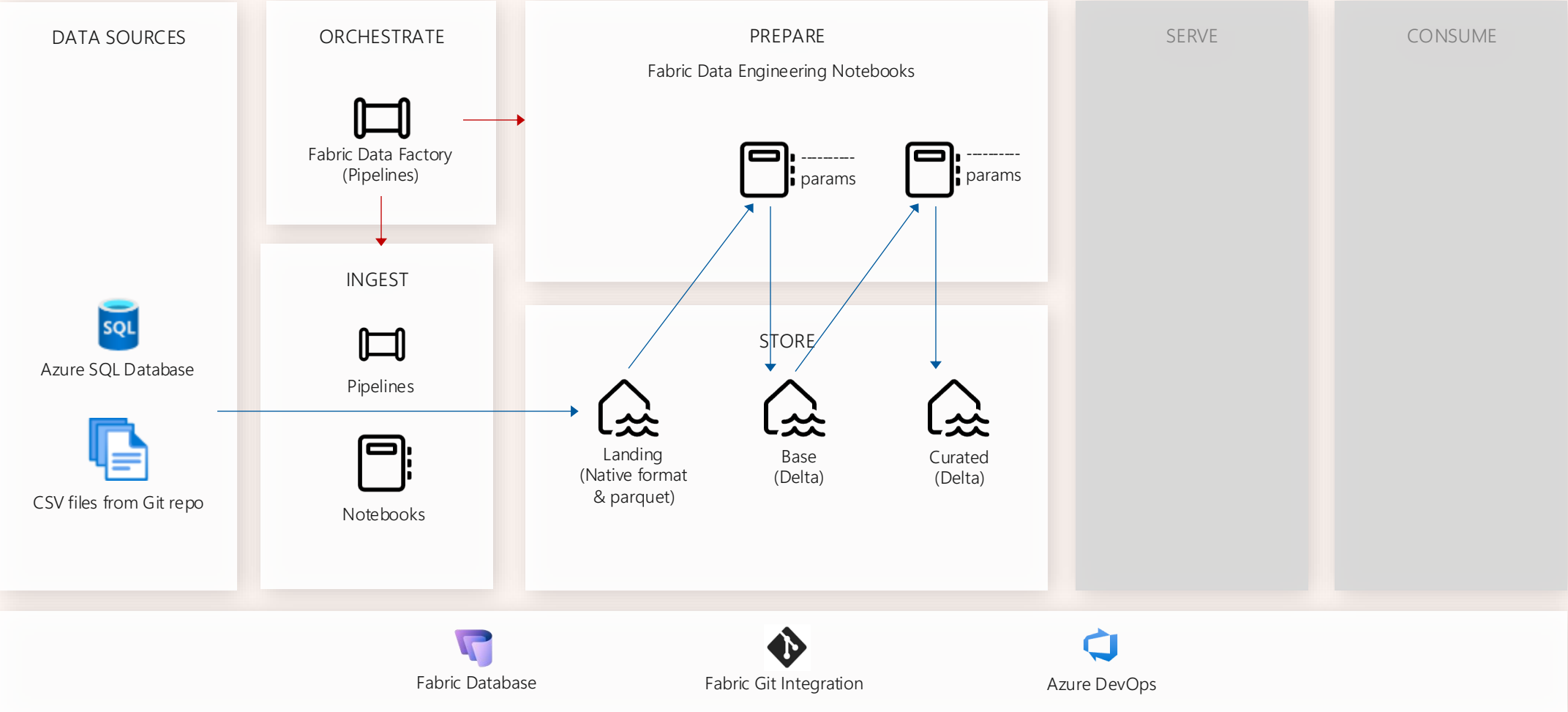


Building and maintaing a Fabric Lakehouse solution

Our starting point architecture



The data flow to be demoed in this session

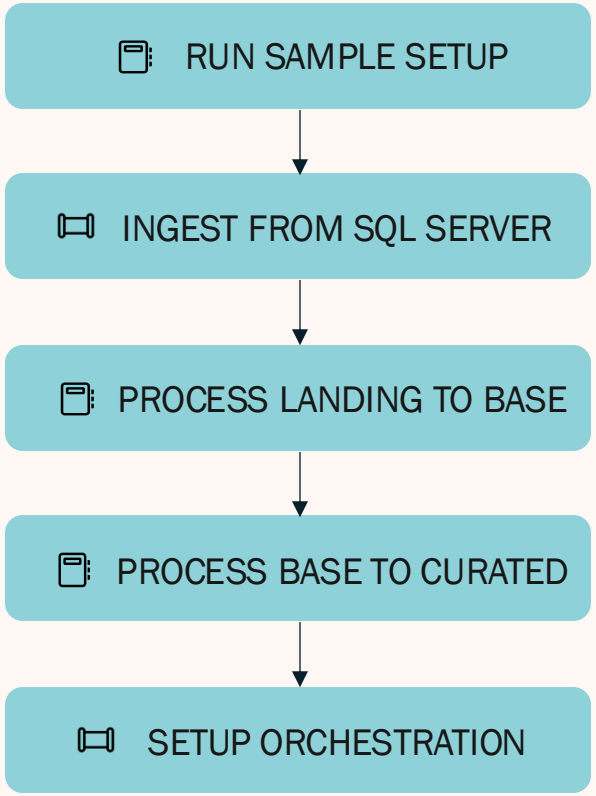


AquaShack - Data Lakehouse accelerator

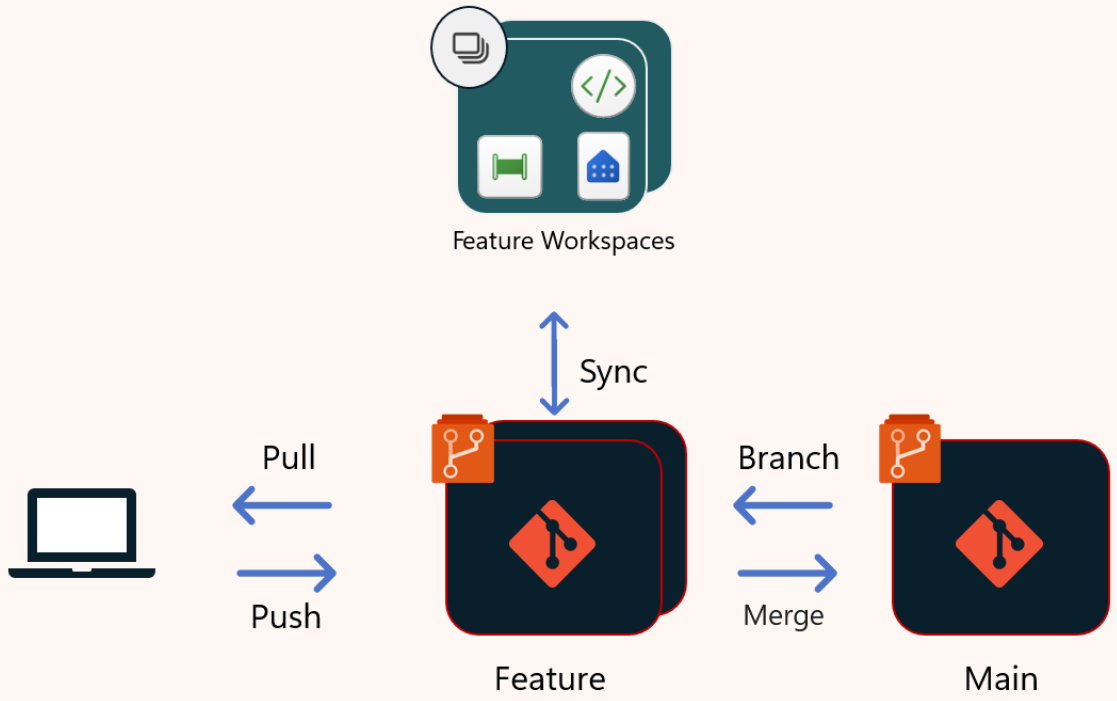
- Pico-example of a meta-data driven framework for Microsoft Fabric
- Lightweight version of twodays AquaVilla Fabric Best Practice Framework
- Supports a 3 layered medallion architecture
  - Landing: Data in its original format when possible. Relation sources in parquet
  - Base: Aligned clean data, all stored in Delta tables
  - Curated: Data for serving, business logic is applied, star schemas are defined etc.
- Common utility functions are defined in AquaShack_Functions notebook
- Based on a modified version of AquaShack by Henrik Reich: https://github.com/ChristianHenrikReich/AquaShack
- Stores metadata in a Fabric Database
- Template Data Pipelines for ingest and orchestration added for demo purposes.
- Also find additional info on AquaShack in the slidedeck from the session "Transforming data into Gold.- a real-world …" facilited by Just Blindbæk.

Metadata driven data refinement

- Setup notebooks for showcase preparation
- Dynamic parent/child pipelines are used for data ingestion
- Data movement between the Landing and Base layers is managed by the LandingToBase notebook.
- Data movement between Base and Curated is managed by individual dimension, fact and bridge notebooks.
- Orchestration of Base to Curated is based on a controller notebook (manual or metadata-driven)
- Pipeline for end-to-end orchestration is stored in the Ingest-layer for simplicity of the demo
- Metadata is maintained in the Fabric Database defining rules for moving data between layers.



Ways-of-working



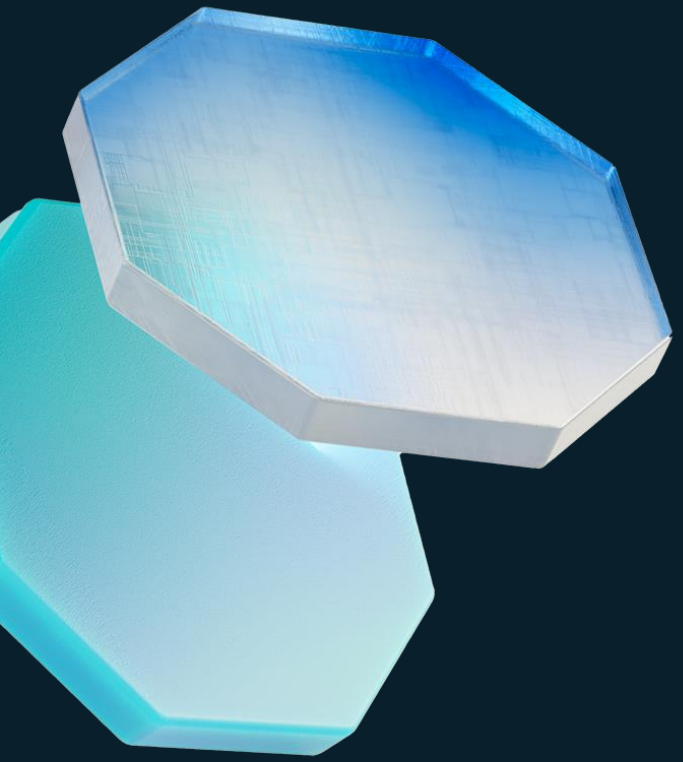
Fabric UI

Scripting

Azure Devops

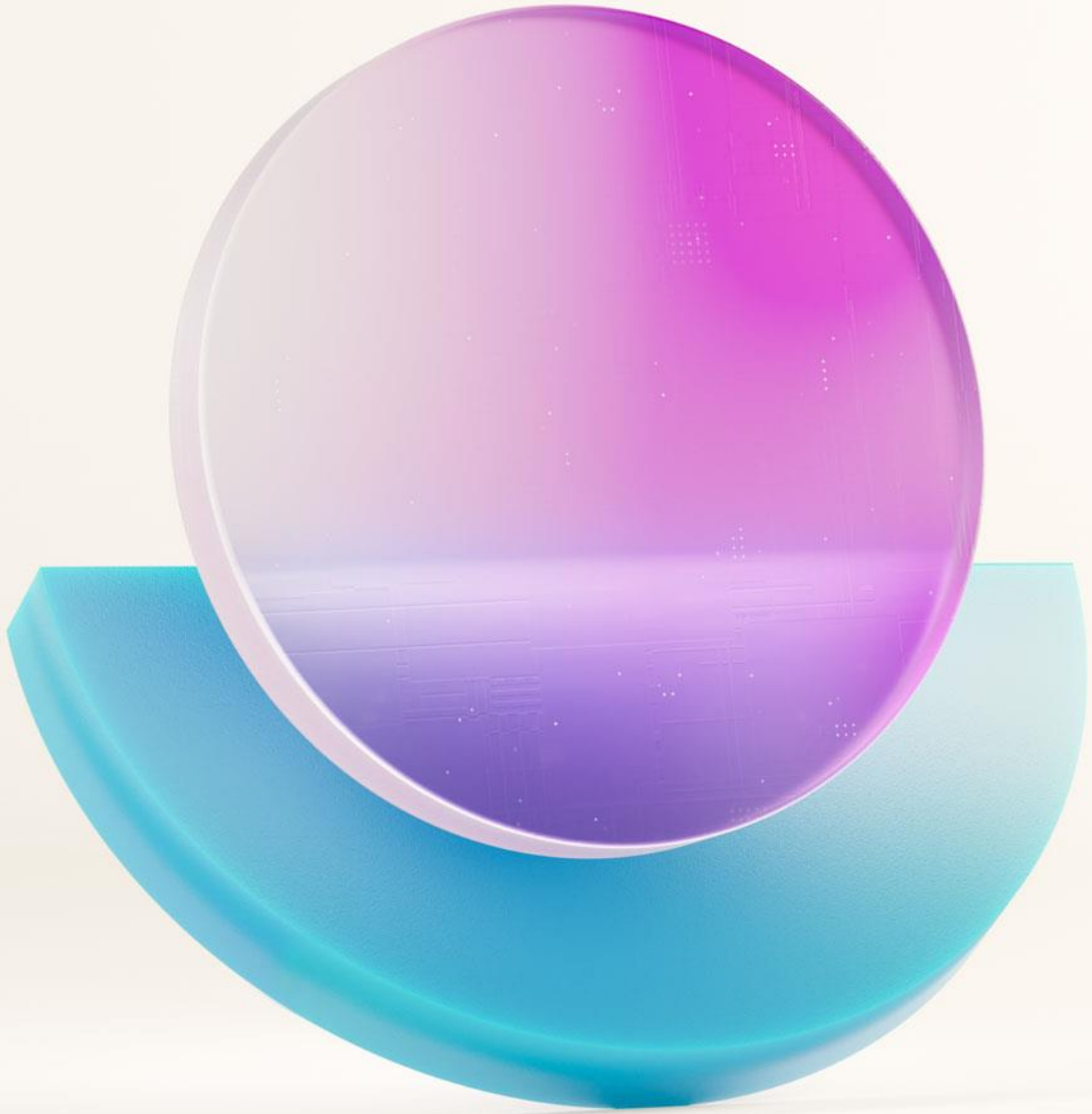
https://learn.microsoft.com/en-us/fabric/cicd/best-practices-cicd

https://learn.microsoft.com/en-us/power-bi/guidance/powerbi-implementation-planning-usage-scenario-enterprise-content-publishing



Demo

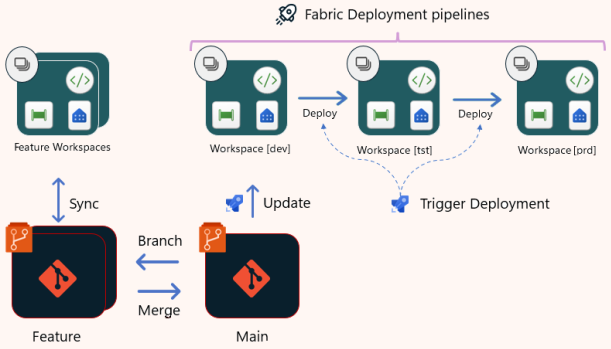
Microsoft Fabric
Community Conference



Operating the Fabric Lakehouse Data Platform

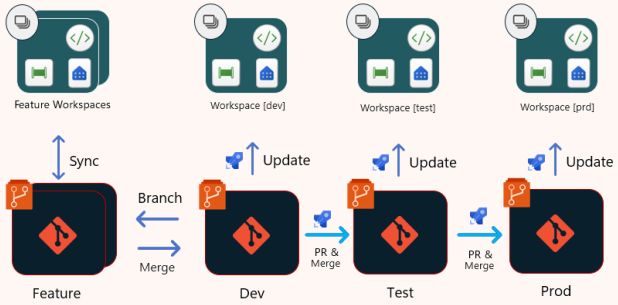
Fabric CI/CD – What are my options?

Deployment Pipelines



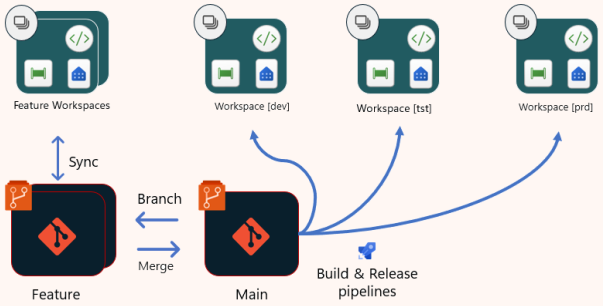
- No code experience
- For simpler solutions
- No support for pre/post deployment tasks

Git-based deployment



- Suitable when using Gitflow
- No need for building environments

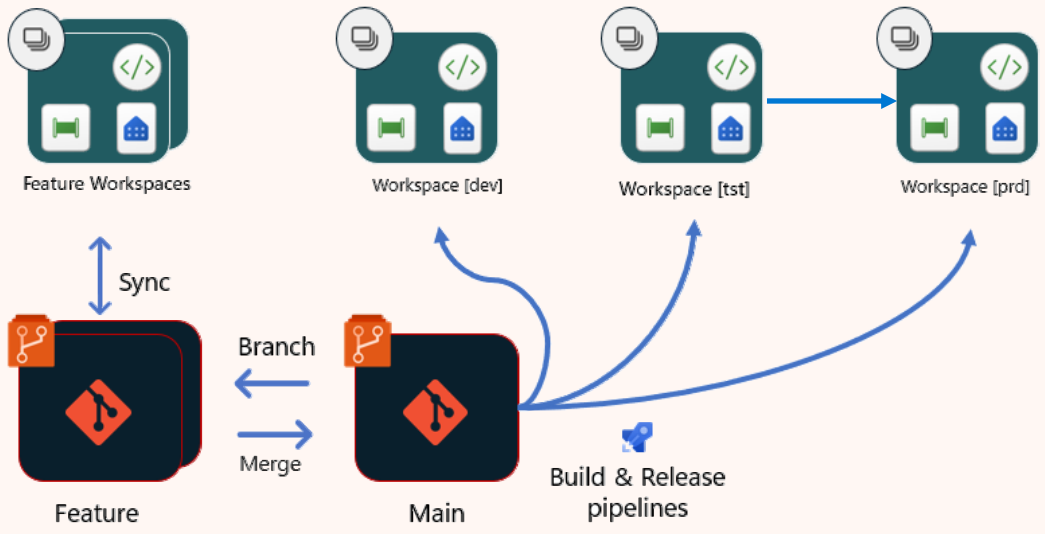
Git-based deployment using build environments



- Supports building enviroments
- Deployment through ADO Pipelines and Github actions
- Supports manipulation of item definitions

Demo of a Git-based deployment with promotion

- Azure DevOps Pipeline with 3 stages:
  - Build
  - Test
  - Production
- Release is done through a Python script utilizing the fabric-cicd Python library.
- Publishes Notebooks and Data Pipelines in Ingest and Prepare layers
- Replace environment specific values through parameter.yml file created during setup

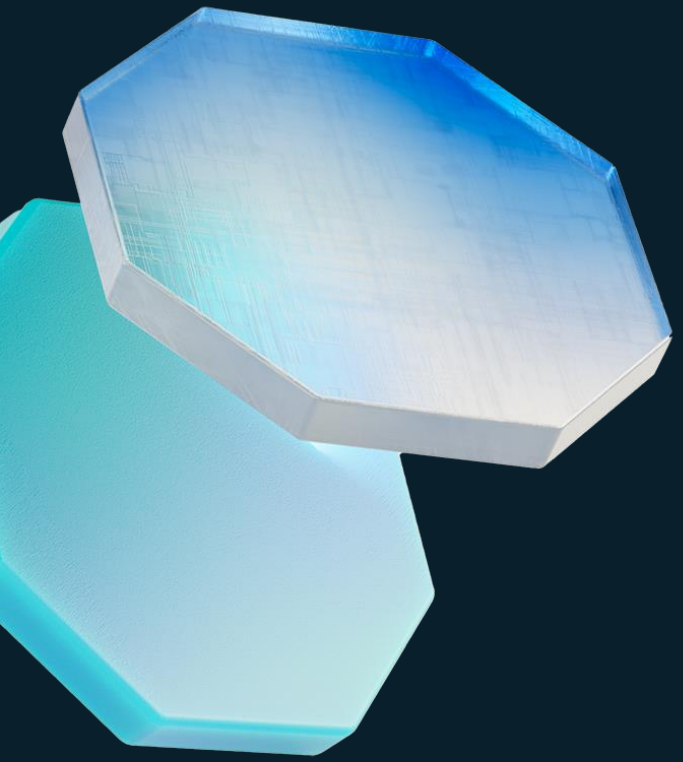


Overview of the fabric-cicd Python library

- Python library designed for use with Microsoft Fabric workspaces.
- Supports code-first CI/CD
- Aimed for developers who prefer not to interact directly with the Microsoft Fabric APIs.
- Currently in Public Preview (0.1.12)
- Supports parameterization for handling environment specific values and spark pool settings *(structure changed from version 0.1.11)*
- Supports running deployment locally, via DevOps pipeline and more…
- Only support items which are source control enabled and API support
- Full deployment by default…
- But also supports excluding items using regex (as of latest version)

	Supported Items
	DataPipeline
	Environment
	Notebook
	Report
	SemanticModel
	Lakehouse
	MirroredDatabase

https://microsoft.github.io/fabric-cicd/latest/



Demo

Summing up

- Split layers into different workspace for scalability, security, governance and performance
- Streamline and automate the full setup of Fabric workspaces and items using Fabric REST APIs
- Automate the creation of feature workspaces as well as cleanup
- Utilize a metadata-driven framework for robustness, scalability and better time-to-market
- Use parameterized and dynamic Data Pipelines and Notebooks to support more robust solutions and to seamless deployment and minimize maintenance
- Use notebookutils (mssparkutils), Semantic Link & Semantic Link Labs when it suits
- Utilize the fabric-cicd Python library for deploying your Fabric items
- Keep an eye on the Terraform Provider for Fabric as future IaC tool

Download source code: https://github.com/gronnerup/Fabric

Blog post "Automating Fabric: Kickstart your Fabric Setup with Python and Fabric REST APIs": https://peerinsights.hashnode.dev

Download of demo code, content and presentation

https://github.com/gronnerup/Fabric

Disclaimer:
This solution demonstrated in this session is **experimental** and provided for **showcasing purposes only**. It is **unsupported**, and there are no guarantees regarding functionality or future updates. You are free to use it **as-is** or modify it to fit your needs. Use at your own risk.

Recommended sessions on automating Fabric and CI/CD

Wednesday

📅 **11:15 AM - 12:15 PM | 📍 Grand BR 122**

Automating CI/CD Processes in Microsoft Fabric:

Best Practices and Latest Developments

Jacob Knightley & Nimrod Shalit

📅 **2:00 - 3:00 PM | 📍 Grand BR 120**

Building a Metadata-driven Framework with

Fabric SQL Database and Data Factory

Erwin De Kreuk

Microsoft

Get Involved in the Fabric Community

aka.ms/FabricCommunity
Connect with community members, ask questions, and learn more about Fabric

aka.ms/FabricUserGroups
Find a user group that matches your interests in your area or online

aka.ms/SuperUsers
Spread your Fabric knowledge, insights, and best practices with others

aka.ms/MVP
Technology experts that share their knowledge and passion with the community

Microsoft

Microsoft Fabric
Community Conference

Thank you