# Fabric options with Azure DevOps

In Fabric, continuous integration and development (CI/CD) features with Git Integration & deployment pipelines allow users to import/export workspace resources with individual updates, deviating from the Azure Data Factory model where whole factory updates using ARM template export methodology is preferred. This change in methodology allows customers to selectively choose which pipelines to update without pausing the whole factory. Both Git integration (bring-your-own Git) and deployment pipelines (built-in CI/CD) use the concept of associated a single workspace with a single environment. You need to map out different workspaces to your different environments such as dev, test, and production.

## Git Integration to Fabric

Git integration in Microsoft Fabric enables developers to integrate their development processes, tools, and best practices straight into the Fabric platform. It allows developers who are developing in Fabric to:

* Backup and version their work
* Revert to previous stages as needed
* Collaborate with others or work alone using Git branches
* Apply the capabilities of familiar source control tools to manage Fabric items

A screenshot of a computer

Description automatically generated

Git supports following items in a workspace.

* Lakehouse
* Notebooks
* Paginated reports
* Reports (except reports connected to semantic models hosted in Azure Analysis Services, SQL Server Analysis Services or reports exported by Power BI Desktop that depend on semantic models hosted in MyWorkspace)
* Semantic models (except push datasets, live connections, model v1, and semantic models created from the Data warehouse/lakehouse.)
* Data pipelines – available in git integration and deployment pipelines
* Warehouse – available in git integration and deployment pipelines
* Spark Environment – available in git integration and deployment pipelines.
* Spark Job Definition – available in git integration.

## Deployment pipeline in Fabric

Deployment pipelines in Fabric allows developers to move their workspace onto higher environments. Fabric's deployment pipelines tool provides content creators with a production environment where they can collaborate to manage the lifecycle of organizational content. Deployment pipelines enable creators to develop and test content in the service before it reaches the users.

A diagram of a diagram

Description automatically generated

## Current limitations

* Currently, only Git in Azure Repos with the same tenant as the Fabric tenant is supported.
* If the workspace and Git repo are in two different geographical regions, the tenant admin must enable cross-geo exports.
* Azure DevOps on-prem isn't supported.
* Sovereign clouds aren't supported.
* Deployment rules in Deployment Pipelines (built-in Fabric CI/CD) aren't yet supported.

## References:

<https://learn.microsoft.com/en-us/fabric/data-factory/git-integration-deployment-pipelines>

<https://learn.microsoft.com/en-us/fabric/cicd/deployment-pipelines/intro-to-deployment-pipelines>

<https://blog.fabric.microsoft.com/en-us/blog/microsoft-fabric-lifecycle-management-getting-started-with-git-integration-and-deployment-pipelines?ft=All>

<https://learn.microsoft.com/en-us/fabric/cicd/deployment-pipelines/get-started-with-deployment-pipelines?tabs=from-fabric>