**Deployment pipelines best practices**

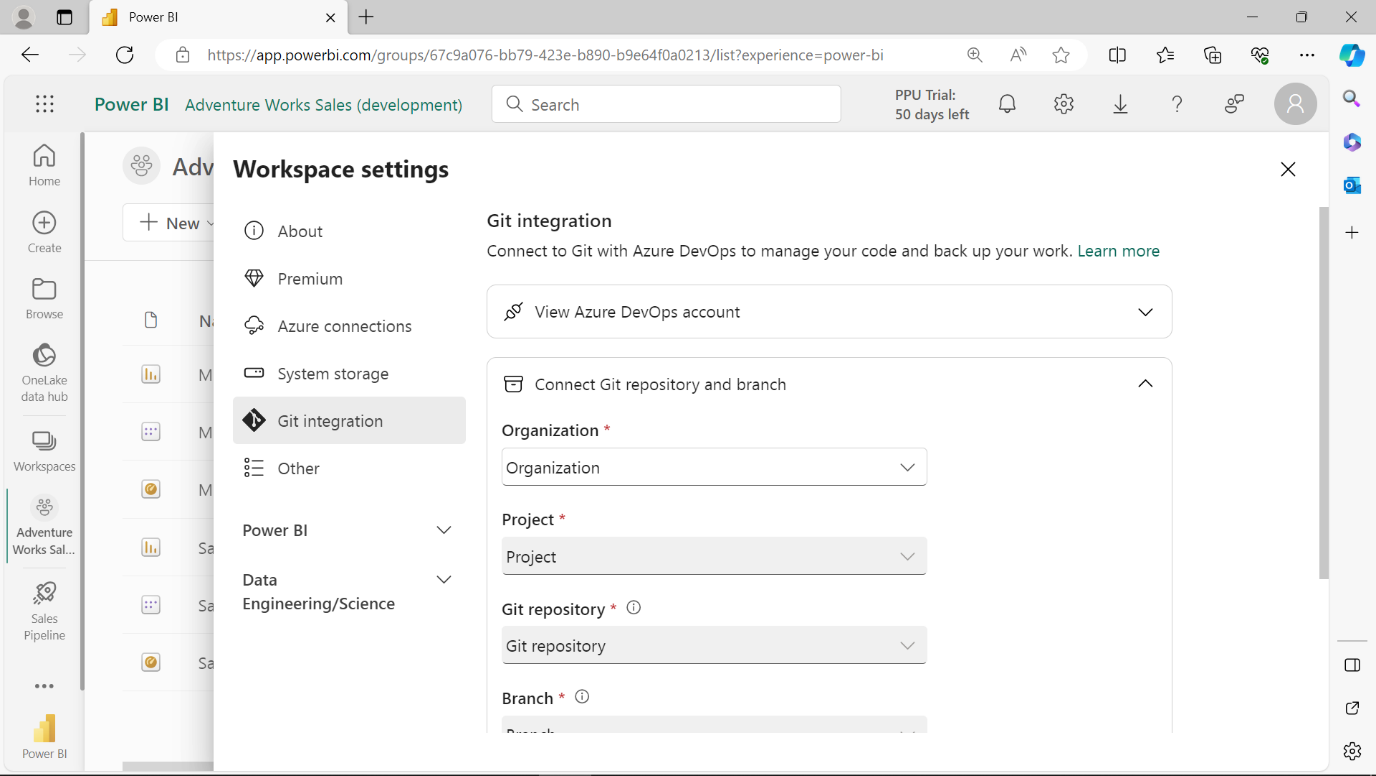
**Introduction**

In this reading, you'll review the best practices for deploying content in Power BI pipelines. You'll learn to manage content smoothly, from creation to production, with fewer errors, by following a structured approach that includes development and testing.

**Deployment pipelines**

In Power BI, optimizing development pipelines involves applying best practices at all stages of a deployment pipeline, including development, testing, and production deployment. This involves creating separate work environments for teams, designing permission models, and linking various stages to databases.

**Git environment**



**Git** is an external system for tracking file changes and managing collaborative work among multiple people. It is most used in larger projects that require detailed tracking and tighter control of the development process.

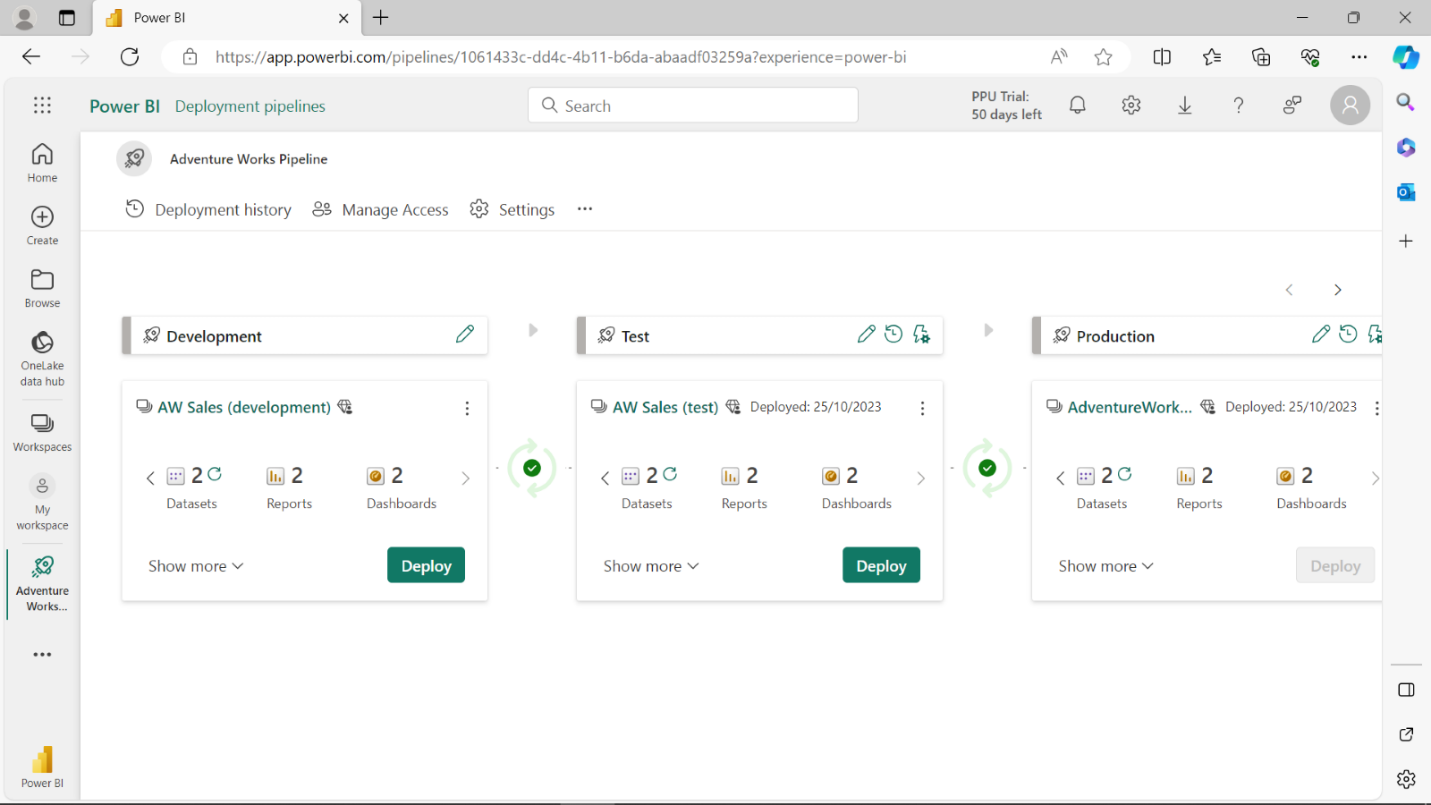
Integrating **Git** into Power BI projects improves the deployment processes by allowing you to track changes and support collaboration. You can use **Git** in any workspace setting.

For example, if Lucas modifies a dataset, such as changing a table, **Git** will record this change. Then, when Adioadjusts the report, **Git** will track whether these changes align correctly with the dataset modifications made by Lucas.

This way, there's a clear record of who made changes. If any issues arise, it's easy to identify the cause of an error or return to a previous project version.

These techniques ensure a smooth, error-free progression of content from start to production, which promotes robust lifecycle management.

Now, let's explore best practices for deployment pipelines in various environments.



**Development environment best practices**

The following is a list of best practices to observe when working in a development environment:

**Separate development between teams**

It's good practice to allocate separate workspaces for different teams. This helps to maintain a clear distinction of everyone's work responsibilities, ensuring no overlap or interference with others. This is particularly useful when working on sensitive or initial drafts. You can work in a **Private** workspace to ensure your work remains isolated and secure.

**Configure permissions**

Understanding and planning who has access to what within the deployment pipelines is important. By doing so, you can ensure a secure and streamlined workflow.

**Back up work in a Git repository**

You can preserve your work by committing it to **Git**. This practice enables version control and ensures that your progress is saved at different stages of development.

**Roll back changes**

If needed, you can revert to previous versions of your work. This can be done using **Git** functionalities, which allow you to manage and control different versions of your content effectively.

**Testing environment best practices**

Testing in Power BI is a critical phase to validate the functionality and performance, with the following practices aimed at achieving an accurate testing environment:

**Simulate your production environment**

By simulating a real production environment during the testing stage, you can anticipate and address various issues that could happen if the two environments differed in data volume or license capacity.

**Check related items**

Verify that your changes don't negatively impact or break the performance of existing items, which might depend on those you update.

**Update data items**

Be cautious when updating data items. Understand the changes and their impact in advance through impact analysis. Test your changes in a development or test environment to understand their impact.

**Test your app**

If you are distributing content through an app, review it by testing it from an end user's point of view to ensure it meets the desired expectations.

**Production environment best practices**

The production stage in Power BI is where the finalized content is made available to end-users. By adhering to the following practices, you can ensure data integrity and availability:

**Manage deployment permissions**

It's good practice to restrict production deployment operations to specifically authorized individuals to maintain a well-controlled and secure environment.

**Update the production app**

Upon deployment to production, it's essential to manually update the app to ensure end users have access to the latest version of the content, so always remember to update your apps manually after a deployment.

**Deploy content using Git branches**

Utilizing **Git** branches facilitates a structured deployment process, allowing updates to be deployed in different stages directly from **Git**, ensuring a streamlined and organized deployment workflow.

**Quick fixes to content**

When encountering production issues, it's always advisable to keep the deployment protocol. This means that instead of resolving the issue directly in production to save time, it’s always advised to deploy the fixes initially to the development stage, test them, and then transition them through the deployment pipeline to production, ensuring they are effective and safe.

**Conclusion**

In Power BI, managing content from development to production is crucial for data accuracy, security, and availability. Best practices in each stage create an efficient deployment pipeline for reliable content delivery.