iNeuron

Architecture Design

Bank Marketing Analytics

Written By	**Aditya Gautam**
Document Version	**2.0**
Last Revised Date	**10-April-2023**

DOCUMENT CONTROL

Change Record:

VERSION	DATE	AUTHOR	COMMENTS
1.0	02-April-2023	Aditya Gautam	Introduction and architecture defined
2.0	10-April-2023	Aditya Gautam	Architecture & Architecture description appended and updated.

Reviews:

VERSION	DATE	REVIEWER	COMMENTS

Approval Status:

VERSION	REVIEW DATE	REVIEWED BY		APPROVED BY	COMMENTS

Contents

1. Introduction

1.1 What is Architecture design document?

Architecture design document is used to represent the organization of the code. Any software needs the architectural design to represents the design of software. A well-defined architectural document makes easier for the new programmers to get familiar with the code. The architecture design document should identify major system components and describe their static attributes and dynamic patterns of interaction.
Dataflow diagrams are also helpful for understanding the interaction between components and overall flow of the data through various systems. IEEE defines architectural design as "the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system." The software that is built for computer-based systems can exhibit one of these many architectures.

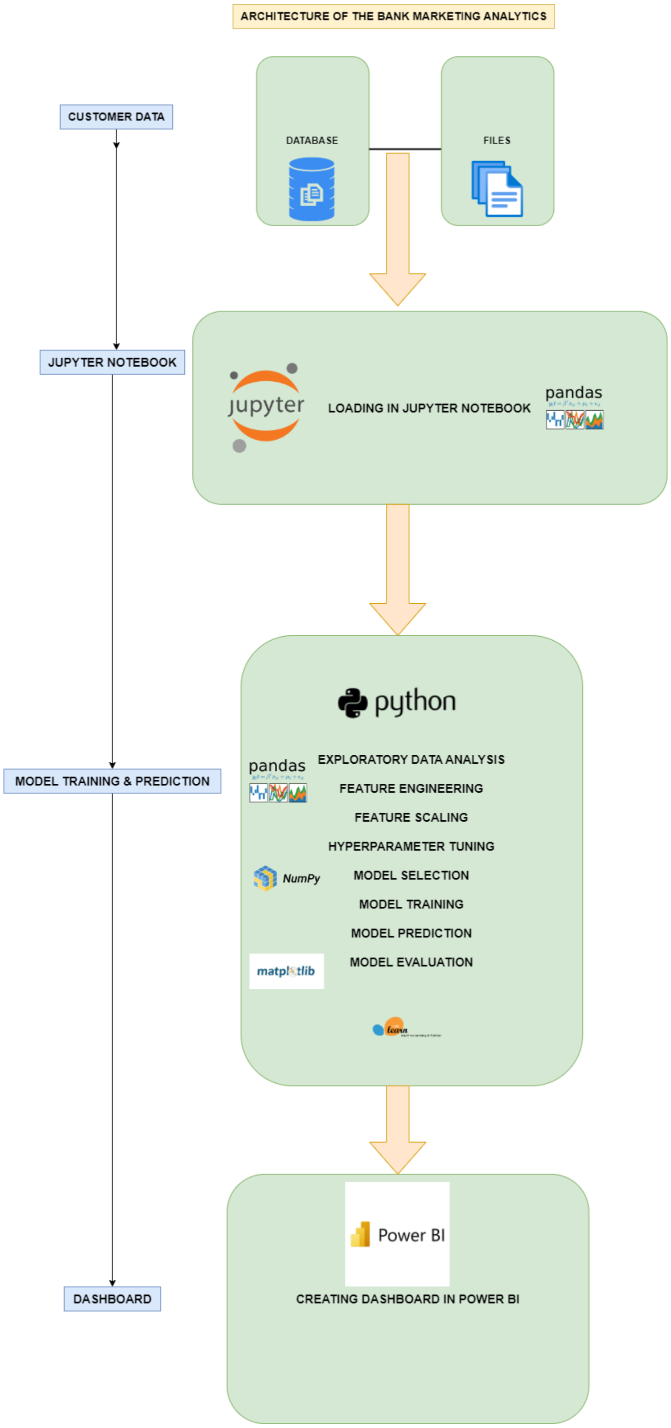
Each style will describe a system category that consists of:
(1) A set of components (Example: a database, computational modules) that will perform a function required by the system.
(2) The set of connectors will help in coordination, communication, and cooperation between the components.
(3) Conditions that how components can be integrated to form the system.
(4) Semantic models that help the designer to understand the overall properties of the system.

1.2 Scope

Architecture Design Document (ADD) is architecture document that helps to understand the data structures, software architecture, flow of the source code and ultimately, performance algorithms. Using the Architecture Design Document (ADD) We define the design principle during the requirement analysis and then refined during architectural design work.

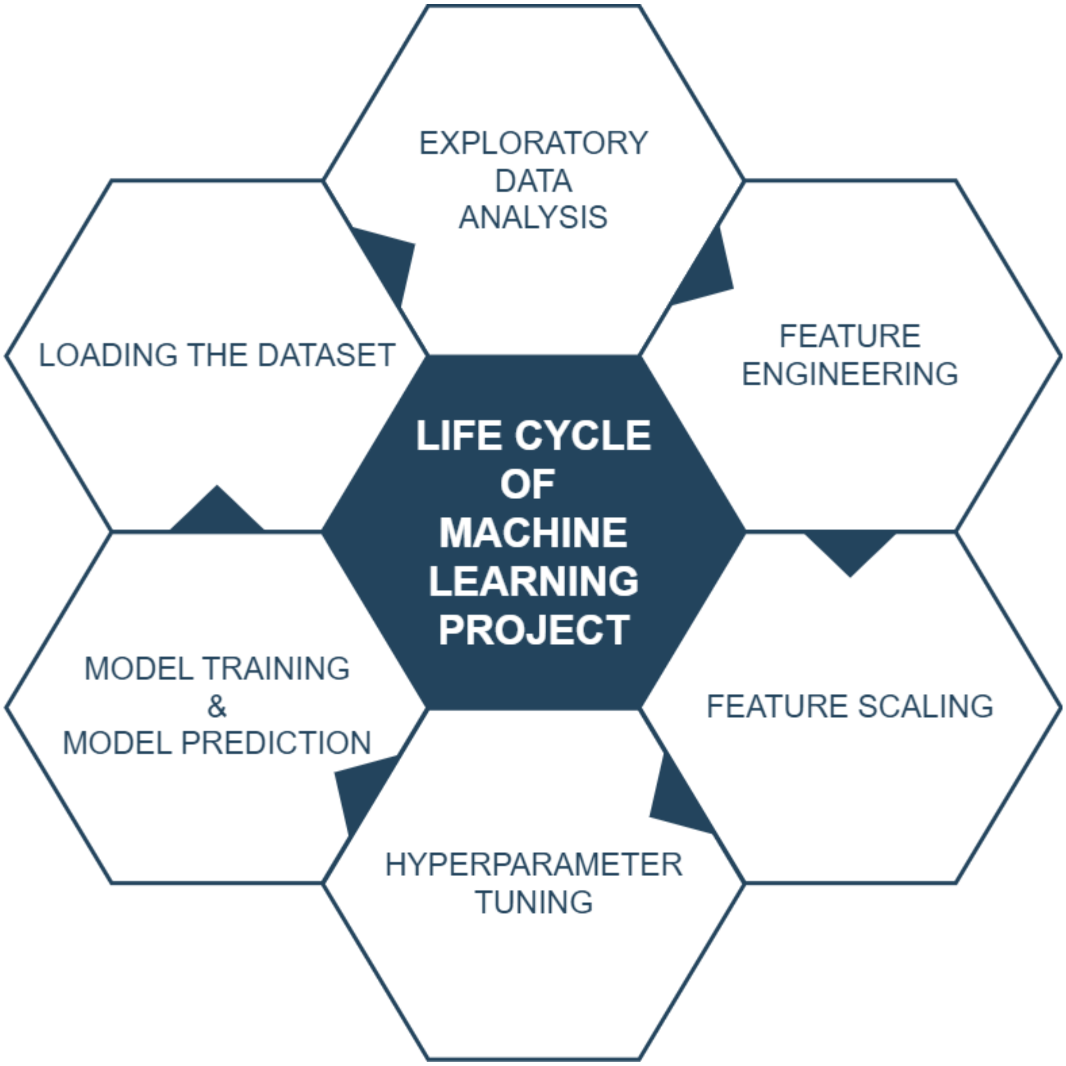
2. **Architecture**

2.1 **Architecture of the Bank Marketing Analytics**

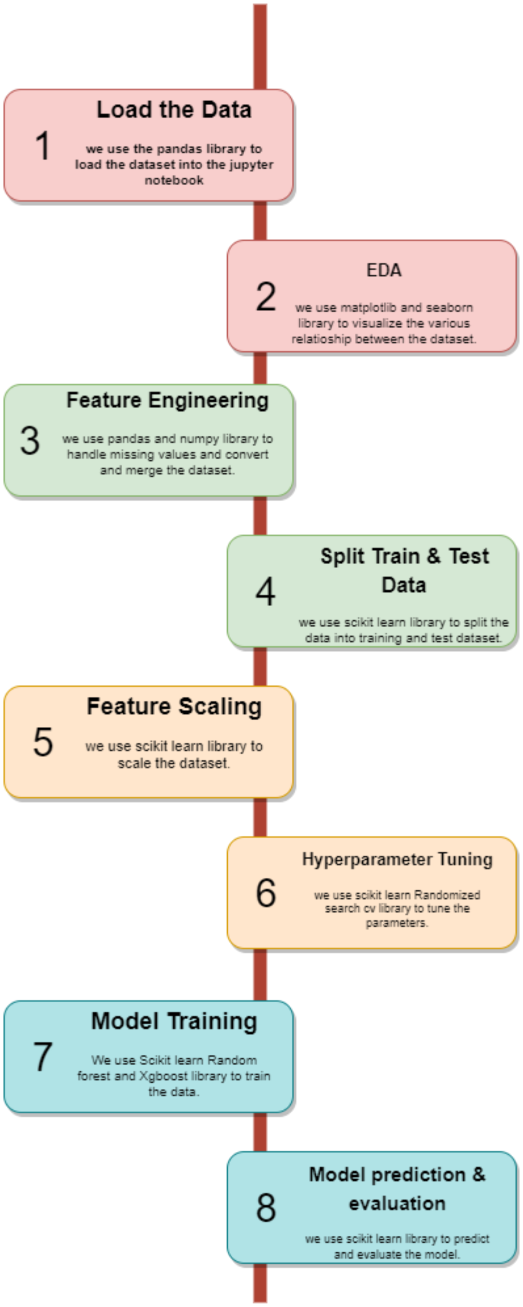


The detailed architecture of the Bank Marketing Analytics has been discussed in the above architecture diagram which gives a overview of the step by step process of the project which gives an idea about flow of the data from original sources to database, then exporting the data from database to importing the data into jupyter notebook by using pandas library for data cleaning process, then for visualize the data, visualization library such Matplotlib and seaborn is used for the purpose and pandas library is used for Feature engineering. Then scikit learn library is used for feature selection , model training, hyperparameter tuning and model evaluation of the data. And finally, deploying the trained data into Power Bi for creating an interactive dashboard.

2.2 **Life Cycle of Machine Learning Project**


EXPLORATORY DATA ANALYSIS
LOADING THE DATASET
FEATURE ENGINEERING
LIFE CYCLE OF MACHINE LEARNING PROJECT
MODEL TRAINING & MODEL PREDICTION
FEATURE SCALING
HYPERPARAMETER TUNING

2.3 **Detailed Architecture of Bank Marketing Analytics**



1 Load the Data

we use the pandas library to load the dataset into the jupyter notebook

2 EDA

we use matplotlib and seaborn library to visualize the various relatioship between the dataset.

3 Feature Engineering

we use pandas and numpy library to handle missing values and convert and merge the dataset.

4 Split Train & Test Data

we use scikit learn library to split the data into training and test dataset.

5 Feature Scaling

we use scikit learn library to scale the dataset.

6 Hyperparameter Tuning

we use scikit learn Randomized search cv library to tune the parameters.

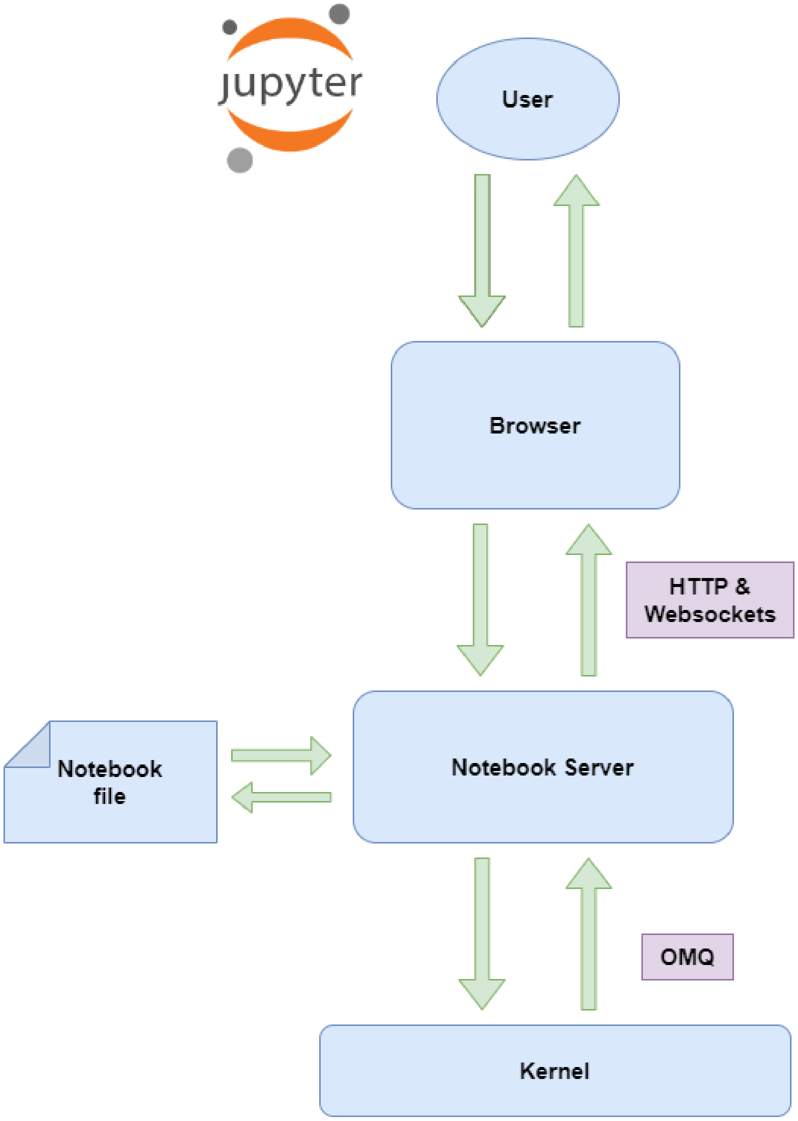
7 Model Training

We use Scikit learn Random forest and Xgboost library to train the data.

8 Model prediction & evaluation

we use scikit learn library to predict and evaluate the model.

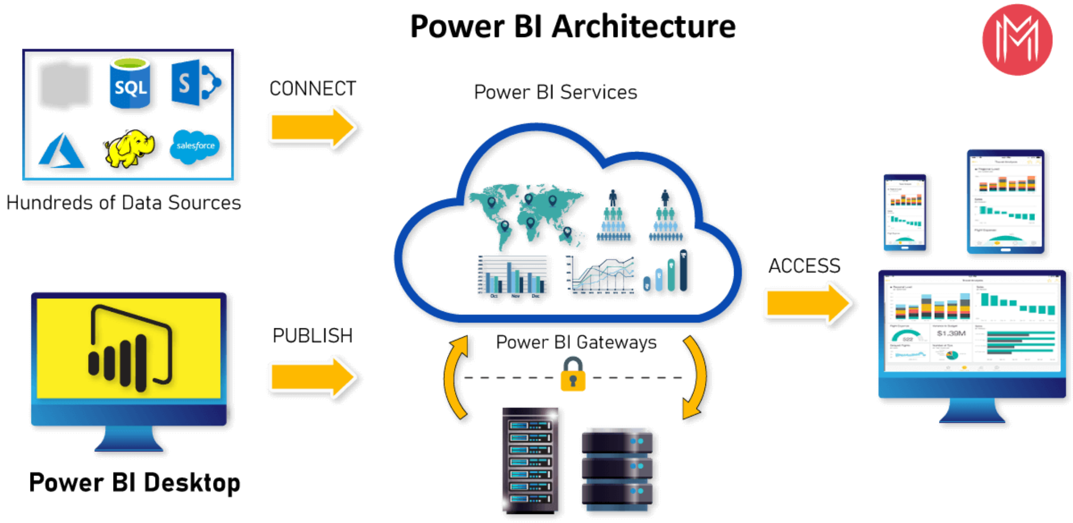
2.4 Jupyter Notebook Architecture



Jupyter Notebook and its flexible interface extends the notebook beyond code to visualization, Multimedia, collaboration, and more. In addition to running your code and output, together with markdown notes, in an editable document called. When you save it, this is sent your browser to the notebook server, which saves it on disk as JSON file with a .ipynb extension.

The notebook server is a communication hub. The browser, notebook file on disk, and kernel cannot talk to each other directly. They communicate through the notebook server. The notebook server, not the kernel, is responsible for saving and loading notebooks, so you can edit notebooks even if you don't have the kernel for that language-- you just won't be able to run code. The kernel doesn't know anything about the notebook document, it just gets sent cells of code to execute when the user runs them.

2.5 **Power BI Architecture**



Power BI

Power BI is a business analytics solution that lets you visualize your data and share insights across your organization, or embed them in your app or website. Connect to hundreds of data sources and bring your data to life with live dashboards and reports.

2.5.1. **Data sources**

The most important component of Power BI is its vast range of data sources.
You can import data from files in your system, cloud-based online data sources or connect directly to live connections. The some of the important data sources are given below:

(1) Excel
(2) Text/CSV
(3) JSON
(4) XML
(5) MySQL Database
(6) PostgreSQL Database
(7) Snowflake

2.5.2. **Power BI Desktop**

Power BI Desktop is a client-side tool known as a companion development and authoring tool.

Power BI Desktop has tools and functionalities to connect the data sources, transform the data, data modeling and creating interactive dashboard.

Using Power BI Desktop features, one can do *data cleansing, create business metrics and data models, define the relationship between data, define hierarchies, create visuals and publish reports.*

(1) Connecting to your data, Access data from hundreds of supported on-premises and cloud-based sources, such as Dynamics 365, Salesforce, Azure SQL DB, Excel, and SharePoint. Ensure it’s always up to date with automated, incremental refreshes. Power BI Desktop enables you to develop deep, actionable insights for a broad range of scenarios.

(2) Preparing and modeling your data with ease, Data prep can take most of your time, but it doesn’t have to with data modeling tools. Reclaim hours in your day using the self-service Power Query experience familiar to millions of Excel users. Ingest, transform, integrate and enrich data in Power BI.

(3) Provide advanced analytics with the familiarity of Excel, Dig deeper into data and find patterns you may have otherwise missed that lead to actionable insights. Use features like quick measures, grouping, forecasting, and clustering. Give advanced users full control over their model using powerful DAX formula language. If you’re familiar with Excel, you’ll feel at home in Power BI.

(4) Create stunning reports with interactive data visualizations. Tell your data story using a drag-and-drop canvas and hundreds of modern data visuals from Microsoft

and partners—or create your own, using the Power BI open source custom visuals framework. Design your report with theming, formatting, and layout tools.

2.5.3. **Power BI Service**

Power BI Service is a web-based platform from where you can share reports made on Power BI Desktop, collaborate with other users, and create dashboards.
It is available in three versions:

- Free version
- Pro version
- Premium version

2.5.4. **Power BI Report Server**

The Power BI Report Server is similar to the Power BI Service. The only difference between these two is that Power BI Report Server is an on-premise platform. It is used by organizations who do not want to publish their reports on the cloud and are concerned about the security of their data.

Power BI Report Server enables you to create dashboards and share your reports with other users following proper security protocols. To use this service, you need to have a Power BI Premium license.

2.5.5. **Power BI Gateway**

This component is used to connect and access on-premise data in secured networks. Power BI Gateways are generally used in organizations where data is kept in security and watch. Gateways help to extract out such data through secure channels to Power BI platforms for analysis and reporting.

2.5.6. **Power BI Mobile**

Power BI Mobile is a native Power BI application that runs on iOS, Android, and Windows mobile devices. For viewing reports and dashboards, these applications are used.

2.5.7. **Power BI Embedded**

Power BI Embedded offers APIs which are used to embed visuals into custom applications.

3. Deployment in Power BI

3.1 Understand the deployment process

The deployment process lets you clone content from one stage in the pipeline to another, typically from development to test, and from test to production.

During deployment, Power BI copies the content from the current stage, into the target one. The connections between the copied items are kept during the copy process. Power BI also applies the configured deployment rules to the updated content in the target stage. Deploying content may take a while, depending on the number of items being deployed. During this time, you can navigate to other pages in the Power BI portal, but you can't use the content in the target stage.

You can also deploy content programmatically, using the deployment pipelines REST APIs. You can learn more about this process in the Automate your deployment pipeline using APIs and DevOps article.

3.2 Deploying content to an empty stage

When you deploy content to an empty stage, the metadata of the reports, dashboards, and datasets in the workspace you're deploying from, is copied to the stage you're deploying to. A new workspace for the stage you deployed to, is created on a Premium capacity.

There are two ways to deploy content from one stage to the next one. You can deploy all the content, or you can select which content items to deploy.

You can also deploy content backwards, from a later stage in the deployment pipeline, to an earlier one.

After the deployment is complete, refresh the datasets so that you can use the newly copied content. The dataset refresh is required because data isn't copied from one stage to another. To understand which item properties are copied during the deployment process, and which item properties are not copied, review the item properties copied during deployment section.

3.3 Creating a Premium workspace

During first-time deployment, deployment pipelines checks if you have Premium permissions.

If you have Premium permissions, the content of the workspace is copied to the stage you're deploying to, and a new workspace for that stage is created on the Premium capacity.

If you don't have Premium permissions, the workspace is created but the content isn’t copied. You can ask a capacity admin to add your workspace to a capacity, or ask for assignment permissions for the capacity. Later, when the workspace is assigned to a capacity, you can deploy content to this workspace.

If you're using Premium Per User (PPU), your workspace is automatically associated with your PPU. In such cases Premium permissions aren't required. However, workspaces created by a PPU user, can only be accessed by other PPU users. In addition, content created in such workspaces can only be consumed by PPU users.

3.4 Workspace and content ownership

The deploying user automatically becomes the dataset owner of the cloned datasets, and the only admin of the new workspace.

3.5 Deploy content to an existing workspace

Deploying content in a working production pipeline, to a stage that has an existing workspace, includes the following:

(1) Deploying new content as an addition, to a stage that already contains content.
(2) New content deployed to replace old content, in a current working stage.

3.6 Deployment process

Content from the current stage is copied over to the target stage. Power BI identifies existing content in the target stage and overwrites it. To identify which content item needs to be overwritten, deployment pipelines uses the connection between the parent item and its clones. This connection is kept when new content is created. The overwrite operation only overwrites the content of the item. The item's ID, URL, and permissions remain unchanged.

In the target stage, item properties that aren't copied, remain as they were before deployment. New content and new items are copied from the current stage to the target stage.

3.7 Auto-binding

In Power BI, when items are connected, one of the items depends on the other. For example, a report will always depend on the dataset it's connected to. A dataset can depend on another dataset, and can also be connected to several reports that depend on it. If there's a connection between two Power BI items, deployment pipelines will always try to maintain this connection.

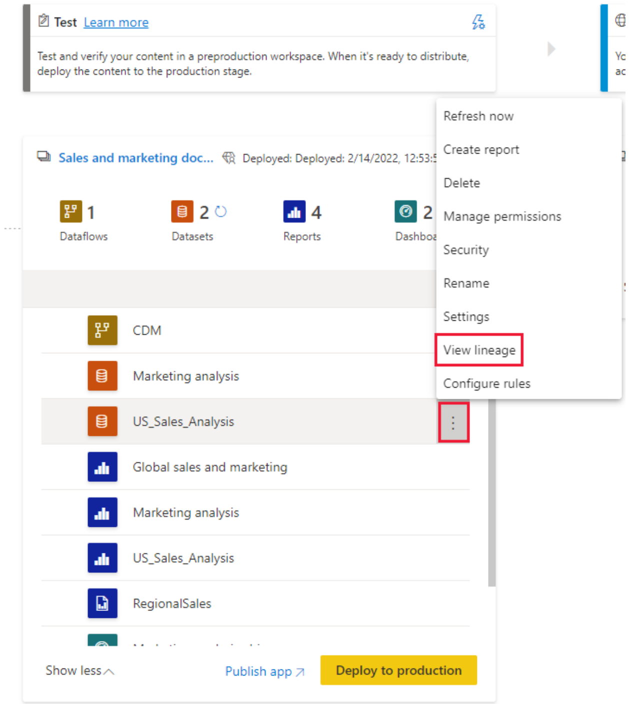
During deployment, deployment pipelines checks for dependencies. The deployment will either succeed or fail, depending on the location of the item that provides the data that the deployed item depends on.

(1) Linked item exists in the target stage - Deployment pipelines will automatically connect (auto-bind) the deployed item, to the item it depends on in the deployed stage. For example, if you deploy a paginated report from development to test, and it's connected to a Power BI dataset that was previously deployed to the test stage, it will be automatically connected to that dataset.

(2) Linked item doesn't exist in the target stage - Deployment pipelines will fail a deployment if an item has a dependency on another item, and the item providing the data isn't deployed and doesn't reside in the target stage. For example, if you deploy a report from development to test, and the test stage doesn't contain its Power BI dataset, the deployment will fail. To avoid failed deployments due to dependent items not being

deployed, use the Select related button. Select related automatically selects all the related items that provide dependencies to the items you're about to deploy.

Auto-binding works only with Power BI items that are supported by deployment pipelines and reside within Power BI. To view the dependencies of a Power BI item, from the item's More options menu, select View lineage.



Test and verify your content in a preproduction workspace. When it's ready to distribute, deploy the content to the production stage.

3.8 Auto-binding across pipelines

Deployment pipelines automatically binds Power BI items that are connected across pipelines, if they're in the same pipeline stage. When you deploy such items, deployment pipelines will attempt to establish a new connection between the deployed item and the item it's connected to in the other pipeline. For example, if you have a report in the test stage of pipeline A that's connected to a dataset in the test stage of pipeline B, deployment pipelines will recognize this connection.

Here's an example with illustrations that will help demonstrate how auto-binding across pipelines works:

(1) You have a dataset in the development stage of pipeline A.

(2) You also have a report in the development stage of pipeline B.

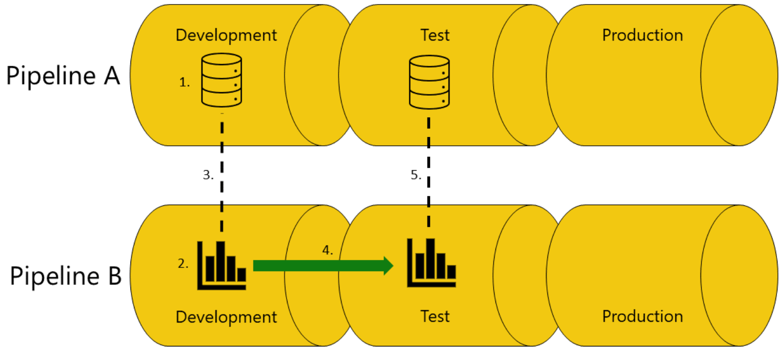
(3) Your report in pipeline B is connected to your dataset in pipeline A. Your report depends on this dataset.

(4) You deploy the report in pipeline B from the development stage to the test stage.

(5) The deployment will succeed or fail, depending on whether or not you have a copy of the dataset it depends on in the test stage of pipeline A:

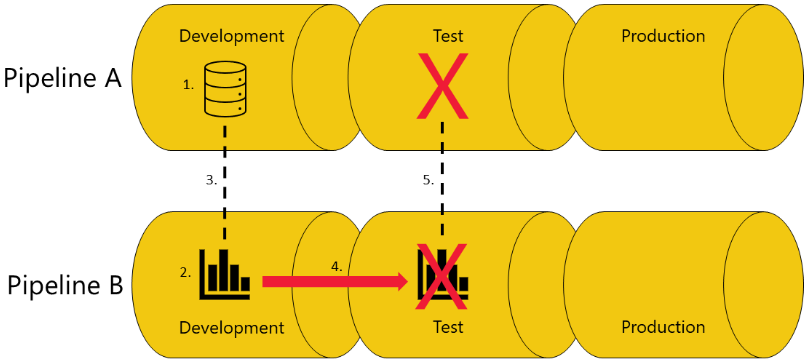
You have a copy of the dataset the report depends on in the test stage of pipeline A

The deployment will succeed, and deployment pipelines will connect (auto-bind) the report in the test stage of pipeline B, to the dataset in the test stage of pipeline A.



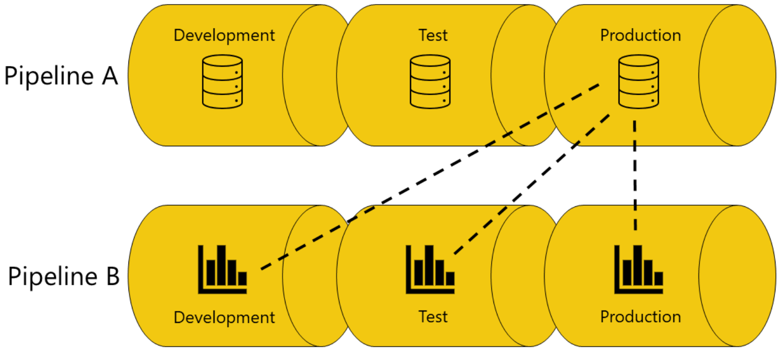
You don't have a copy of the dataset the report depends on in the test stage of pipeline A

The deployment will fail because deployment pipelines can't connect (auto-bind) the report in the test stage in pipeline B, to the dataset it depends on in the test stage of pipeline A.



3.9 Avoid using auto-binding

In some cases, you might not want to use auto-binding. For example, if you have one pipeline for developing organizational datasets, and another for creating reports. In this case, you might want all the reports to always be connected to datasets in the production stage of the pipeline they belong to. To accomplish this, you'll need to avoid using the auto-binding feature.



There are three methods you can use to avoid using auto-binding:

Don't connect the Power BI item to corresponding stages. When the items aren't connected in the same stage, deployment pipelines keeps the original connection. For example, if you have a report in the development stage of pipeline B that's connected to a dataset in the production stage of pipeline A. When you deploy the report to the test stage of pipeline B, it will remain connected to the dataset in the production stage of pipeline A.

Define a parameter rule. This option isn't available for reports, you can only use it with datasets and dataflows.

Connect your reports dashboards and tiles to a proxy dataset or dataflow, that isn't connected to a pipeline.

3.10 Auto-binding and parameters

Parameters can be used to control the connections between datasets or dataflows and the Power BI items that they depend on. When a parameter controls the connection, auto-binding after deployment won't take place, even when the connection includes a parameter that applies to the dataset's or dataflow's ID, or the workspace ID. In such cases, you'll need to rebind the items after the deployment by changing the parameter value, or by using parameter rules.

3.11 Refreshing data

Data in the target Power BI item, such as a dataset or dataflow, is kept when possible. If there are no changes to a Power BI item that holds the data, the data is kept as it was before the deployment.

In many cases, when you have a small change such as adding or removing a table, Power BI keeps the original data. For breaking schema changes, or changes in the data source connection, a full refresh is required.

3.12 Requirements for deploying to a stage with an existing workspace

A user with a Pro license or a PPU user who's a member of both the target and source deployment workspaces, can deploy content that resides on a premium capacity to a stage with an existing workspace. For more information, review the permissions section.

3.13 Deployed items

When you deploy content from one pipeline stage to another, the copied content contains the following Power BI items:

(1) Datasets
(2) Reports
(3) Dataflows
(4) Datamarts
(5) Dashboards
(6) Paginated reports

3.14 Unsupported items

Deployment pipelines doesn't support the following items:

(1) Datasets that don't originate from a PBIX
(2) PUSH datasets
(3) Streaming dataflows
(4) Reports based on unsupported datasets
(5) Template app workspaces
(6) Workbooks
(7) Metrics

3.15 Item properties copied during deployment

During deployment, the following item properties are copied and overwrite the item properties at the target stage:

(1) Data sources (deployment rules are supported)
(2) Parameters (deployment rules are supported)
(3) Report visuals
(4) Report pages
(5) Dashboard tiles
(6) Model metadata
(7) Item relationships

3.16 Incremental refresh

Deployment pipelines supports incremental refresh, a feature that allows large datasets faster and more reliable refreshes, with lower consumption.

With deployment pipelines, you can make updates to a dataset with incremental refresh while retaining both data and partitions. When you deploy the dataset, the policy is copied along.

3.17 Activating incremental refresh in a pipeline

To enable incremental refresh, configure it in Power BI Desktop, and then publish your dataset. After you publish, the incremental refresh policy is similar across the pipeline, and can be authored only in Power BI Desktop.

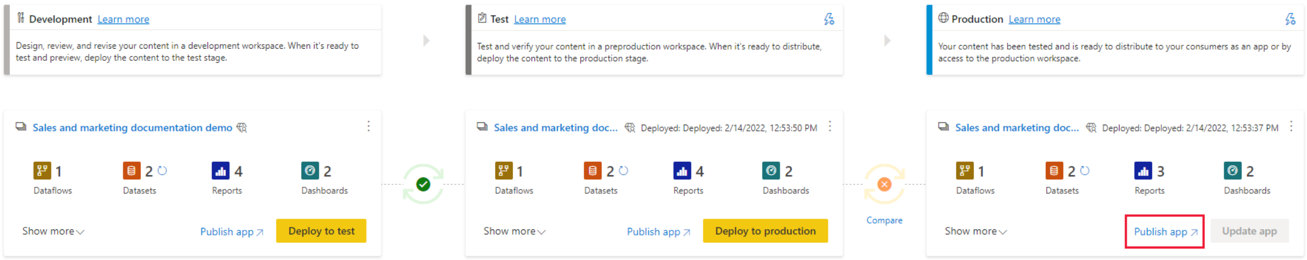
Once your pipeline is configured with incremental refresh, we recommend that you use the following flow:

(1) Make changes to your PBIX file in Power BI Desktop. To avoid long waiting times, you can make changes using a sample of your data.
(2) Upload your PBIX file to the development stage.
(3) Deploy your content to the test stage. After deployment, the changes you made will apply to the entire dataset you're using.
(4) Review the changes you made in the test stage, and after you verify them, deploy to the production stage.

3.18 Deploying Power BI apps

Power BI apps are the recommended way of distributing content to free Power BI consumers. Using deployment pipelines you can manage Power BI apps in a deployment pipeline, so that you have more control and flexibility when it comes to your app's lifecycle.

Create an app for each deployment pipeline stage, so that you can test each app update from an end user's point of view. A deployment pipeline allows you to manage this process easily. Use the publish or view button in the workspace card, to publish or view the app in a specific pipeline stage.

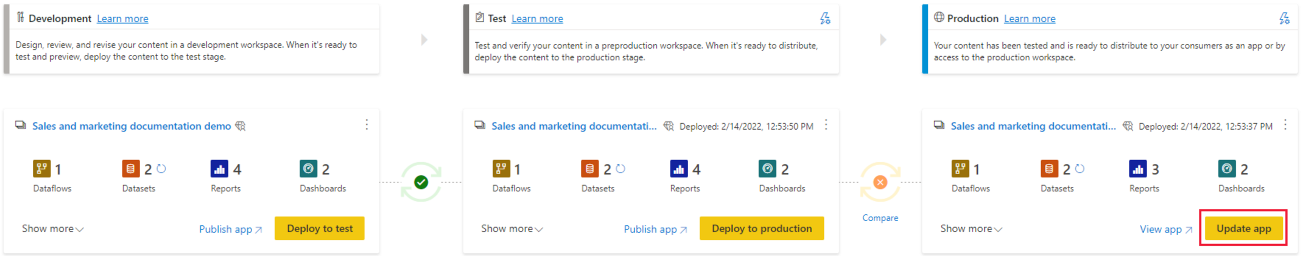


Design, review, and revise your content in a development workspace. When it's ready to test and preview, deploy the content to the test stage.

Test and verify your content in a preproduction workspace. When it's ready to distribute, deploy the content to the production stage.

Your content has been tested and is ready to distribute to your consumers as an app or by access to the production workspace.

In the production stage, the main action button on the bottom-right corner opens the update app page in Power BI, so that any content updates become available to app users.



Design, review, and revise your content in a development workspace. When it's ready to test and preview, deploy the content to the test stage.

Test and verify your content in a preproduction workspace. When it's ready to distribute, deploy the content to the production stage.

Your content has been tested and is ready to distribute to your consumers as an app or by access to the production workspace.