Architecture Design

ANALYZING AMAZON SALES DATA

Revision Number – 1.3

Last Date of Revision – 16-04-2022

R Shashikiran

Document Version Control

Date	Version	Description	Author
13-04-2022	1.0	Introduction	R Shashikiran
15-04-2022	1.1	Architecture	R Shashikiran
16-04-2022	1.2	Deployment	R Shashikiran

Contents

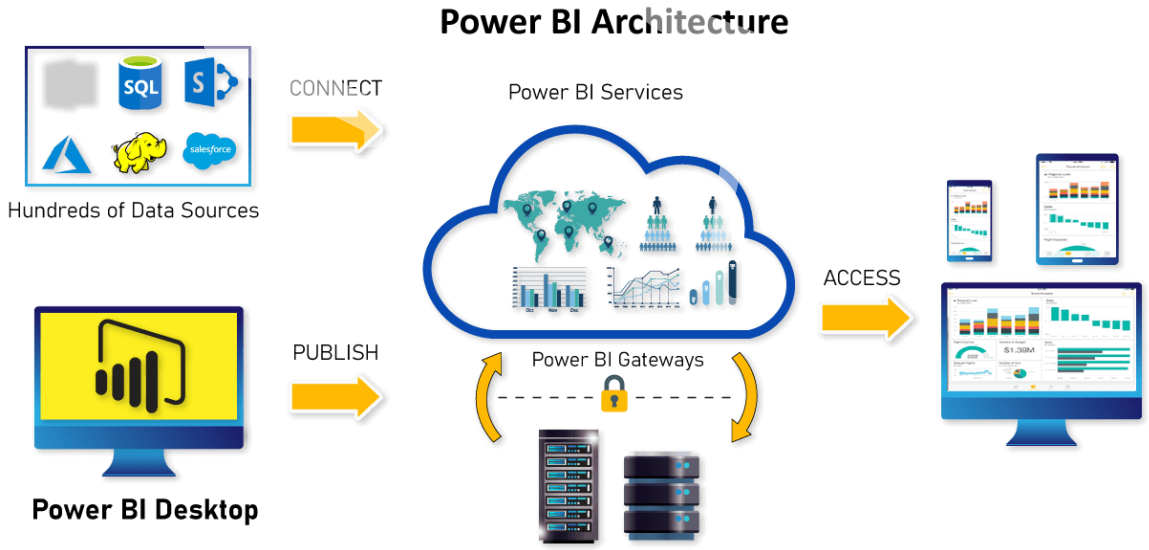
1. Introduction

1.1. Why this Architecture Design documentation?

The design documentation provides written documentation of the design factors and the choices the architect has made in the design to satisfy the business and technical requirements. The design documentation also aids in the implementation of the design.

1.2. Scope

Architecture document is a map of the software. We use it to see, at a glance, how the software is structured. It helps you understand the software's modules and components without digging into the code. It's a tool to communicate with others—developers and non-developers—about the software.

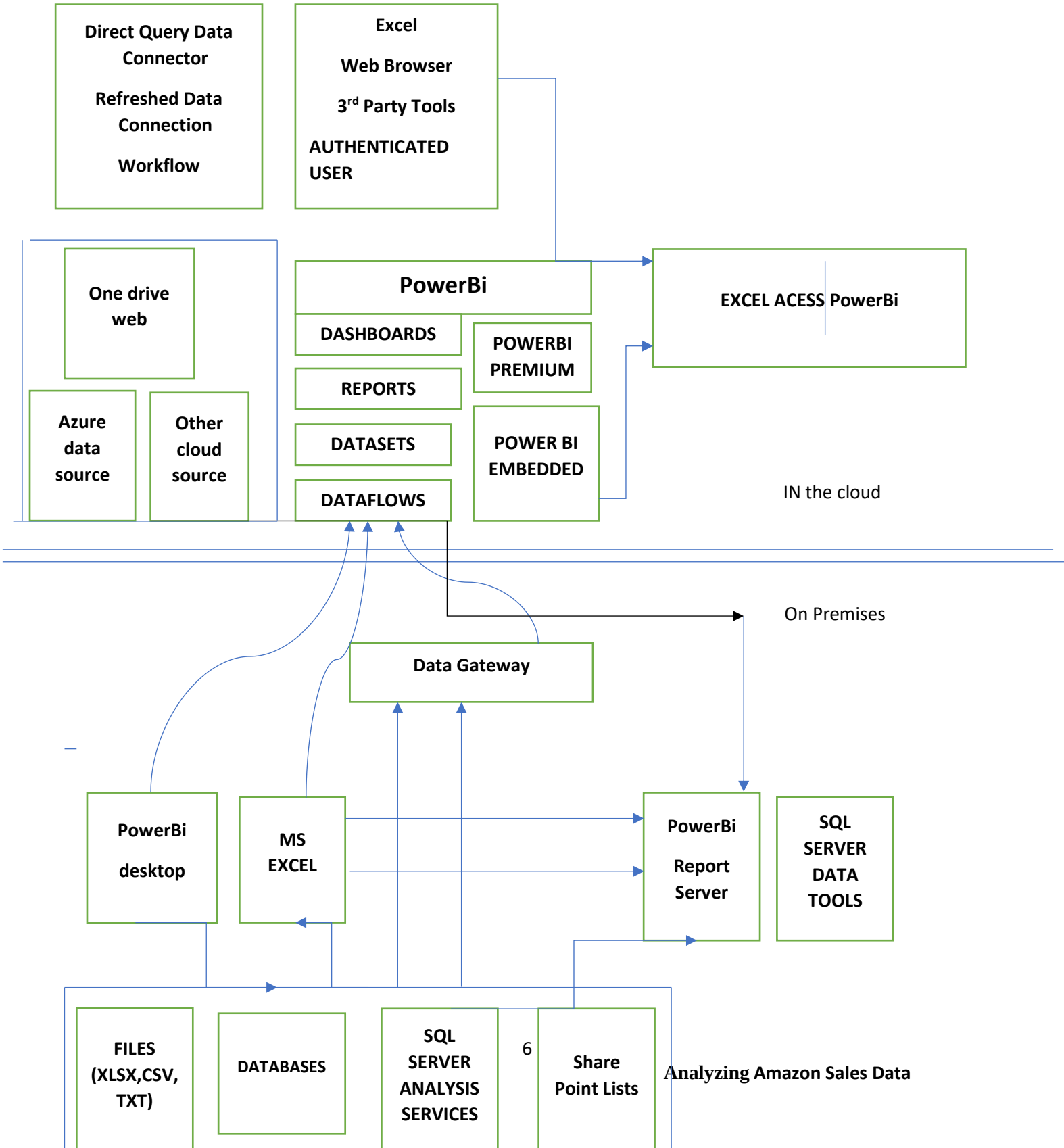


Power BI is a business platform that includes several technologies to work together. It delivers outstanding business intelligence solutions. Power BI Architecture contains four steps.

1. Data Integration
2. Data Transforming
3. Report & Publish
4. Creating and Dashboard

2. Architecture

2.1. PowerBi Architecture



In the above diagram, it is clear that the upper half part represents On-Cloud services, and the lower half part represents the On-Premise services.

If you observe in the top of the image excel, web browsers and other sources are streaming into Power BI components, and they are called data sources. These data sources are authenticated users. Power BI has different data sources like On-Premise, Cloud databases, direct connections, etc.

On-Premise:

Power BI Desktop is accomplished with the authenticating, development and publishing tools. You can transfer the data from data sources to Power BI Desktop. And also, it allows users to create and publish reports on the Power BI Report Server or Power BI Service.

Power BI Publisher allows you to publish the Excel workbooks to the Power BI Report Server. Report Publisher and SQL server Data tools help in creating the KPIs, datasets, paginated reports, mobile reports, etc. All kinds of reports are published at the Power BI Report Server, and from there, reports are distributed to the end-users.

On-Cloud:

Power BI Gateway is the essential component in the Power BI architecture. The Power BI Gateway acts as a bridge or secure channel to transfer the data from On-premise data to On-cloud data sources or apps.

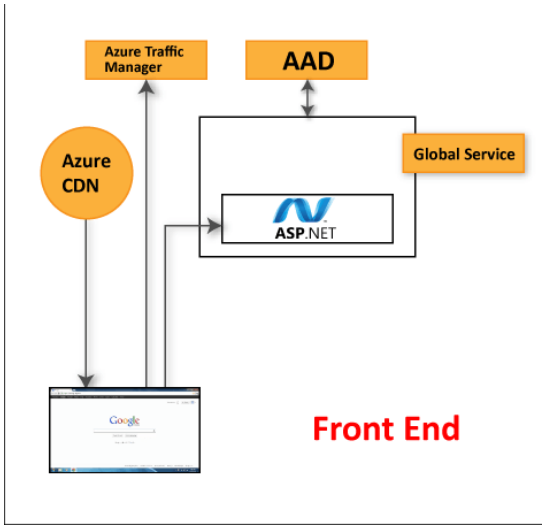
Cloud side architecture consists of a lot of components including Power suite having datasets, dashboards, reports, Power BI Premium, Power BI Embedded, etc. Users can embed the dashboards, reports into applications, SharePoint, Teams, etc. There are Cloud data sources and they are connected to the Power BI tools.

2.2. PowerBi Server Architecture

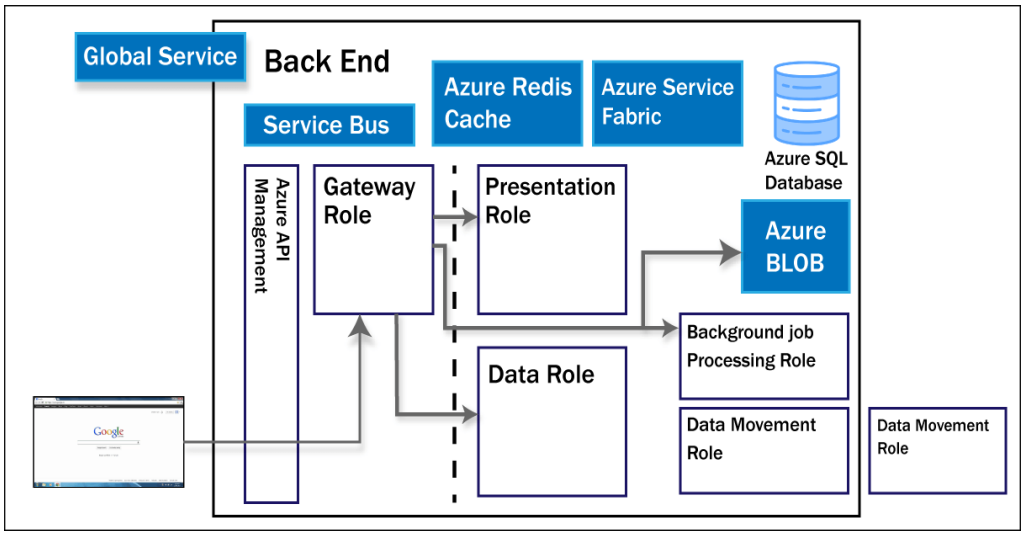
Power BI Service Architecture consists of two clusters. The following are the two clusters.

- Front End Cluster
- Back End Cluster

Front End Cluster: Front end cluster acts as an intermediate between the back-end cluster and the clients. It is also called a Web Front End Cluster. It establishes the initial connection and authenticates the users or clients using the Azure Active Directory. After user authentication, Azure Traffic Manager directs the user requests to the nearest data centers and Azure Content Delivery Network (CDN) allocates the statice files/content to the users or clients based on the geographical locations.



2. Back End Cluster: It manages the datasets, reports, storage, visualizations, data refreshing, data connections, and other services in the Power BI. At the back end cluster, the web client has only two direct points to interact with the data, i.e., Gateway Role and Azure API Management. These two components are responsible for authorizing, load balancing, routing, authentication, etc.



3. Deployment Description

3.1. Deployment Pipeline in Power BI

The Deployment Pipelines is a Power BI feature that helps developers to keep the different release environments (namely Development, Test, and Production) in sync with each other.

It is just a simple UI, where you can set up a Power BI release pipeline for the three different environments.

Deployment Pipeline Considerations

- In order to use Deployment Pipelines, you or your organization must have a Power BI Premium account.
- You don't have to mandatorily have all the 3 environments. Even 2 will do. If you have just 2, you can use the Test and Production pipelines.
- The feature also supports backward deployment. So, if you already have a steady production environment, you can deploy it backward (to Test or Dev, based on your need) and set up the pipeline. The feature is not limited to any sequential orders.
- Recently, Power BI released REST APIs for the deployment pipelines. The APIs give you the wings to automate the syncing as well. The API documentation covers a few critical functions as of now, and Power BI will probably add more in the future.
- The Deployment Pipeline feature doesn't support all data sources at present.

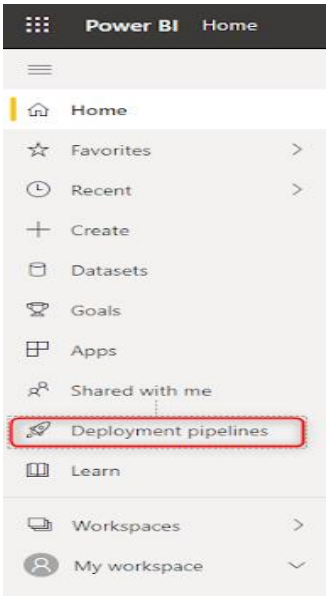
Prerequisites for Power BI Deployment Pipelines Set Up
Power BI Premium Account – As mentioned in the considerations section, you must have a Power BI premium account. Any of the following scenarios will get you access to the Power BI deployment pipelines feature:

- An organizational Power BI Premium capacity account, with a Pro account for the users. Here you need the Pro account for publishing reports from the Power BI Desktop.
- Power BI Premium account at per-user plan

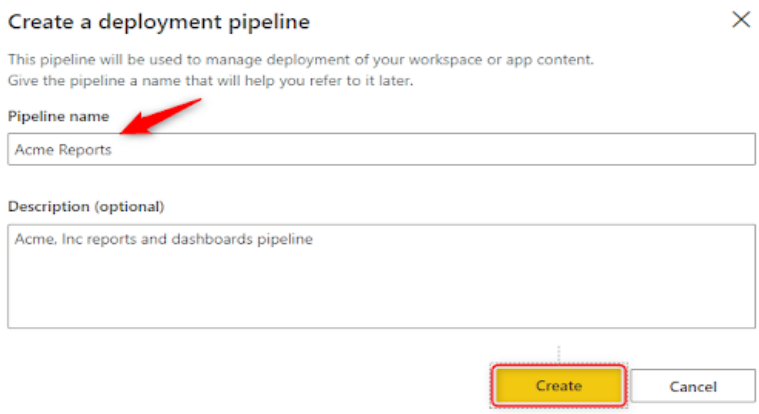
Workspace Permissions – The Deployment Pipelines are created at the workspace level. The workspace must be a Premium workspace (created with a Premium account), and you must be the admin of that workspace.

3.2. Steps to Setup Deployment Pipelines

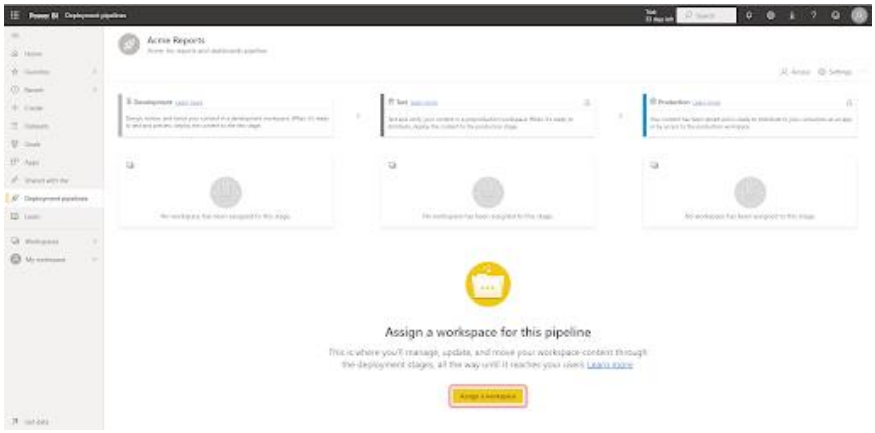
1. Sign in to app.powerbi.com.
2. In the left pane, click the **Deployment Pipelines** menu.



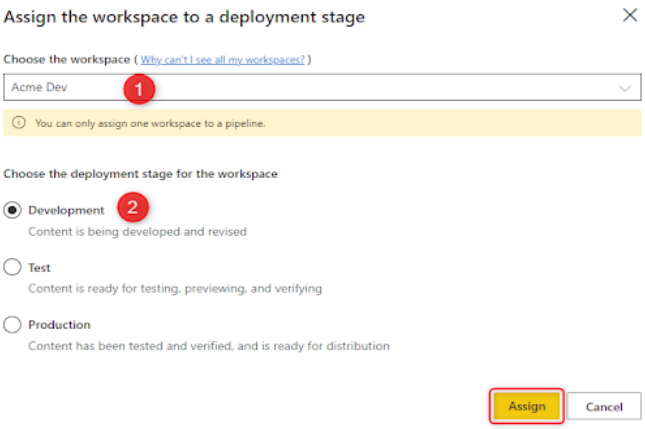
3. Click the Create a Pipeline button. In the popup that opens, enter your Pipeline name (mandatory) and Description (optional).



4. Next, click on the **Assign a Workspace** button.

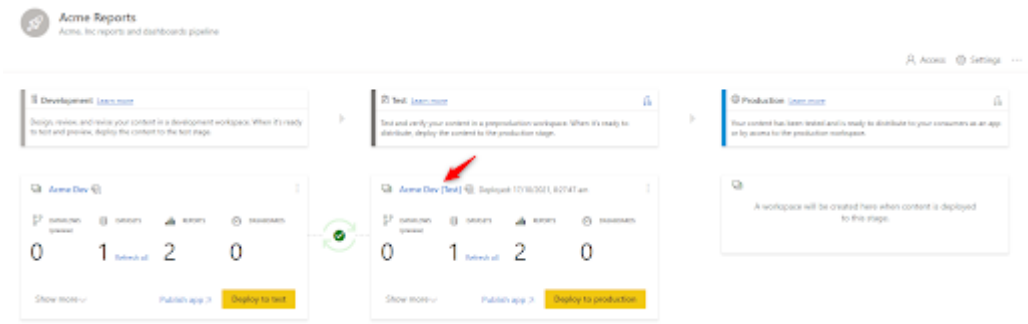


5. Select the Workspace. (Note, only premium workspaces show up in the drop-down). Then select the deployment stage, and click on the **Assign** button.



6. Now your development environment is set up in Power BI. The card shows the number of Dataflows, Datasets, Reports, and Dashboards involved in the environment.
Next, just click on the **Deploy to Test** button.

7. The dataset and content in your development workspace are now copied to the Test Workspace. Power BI created a new premium workspace automatically with the name **YourDevelopmentWorkspaceName [Test]**, as indicated in the screenshot below.


Acme Reports
Development
Test
Production
Acme Dev
Acme Dev [Test]
0
1
2
0
Deploy to test
Deploy to production
A workspace will be created here when content is deployed to this stage.

8. As the datasets in the Test workspace are directly deployed from the Development environment, they are connected to the Dev data sources still. You can point them to the data sources in your Test environment by following the steps below in your Test environment. Click on the lightning icon in the top of your test card: