



PREDICTING EMPLOYEE ATTRITION USING FUNCTION APP AND POWER BI

SUMMARY:

In order to predict employee attrition, the human resources team must make decisions in an organization. There are many complex, interrelated variables that impact the likelihood of employees quitting, which makes it extremely difficult to manually predict which employees will quit when they'll quit and why they'll quit, especially at scale.



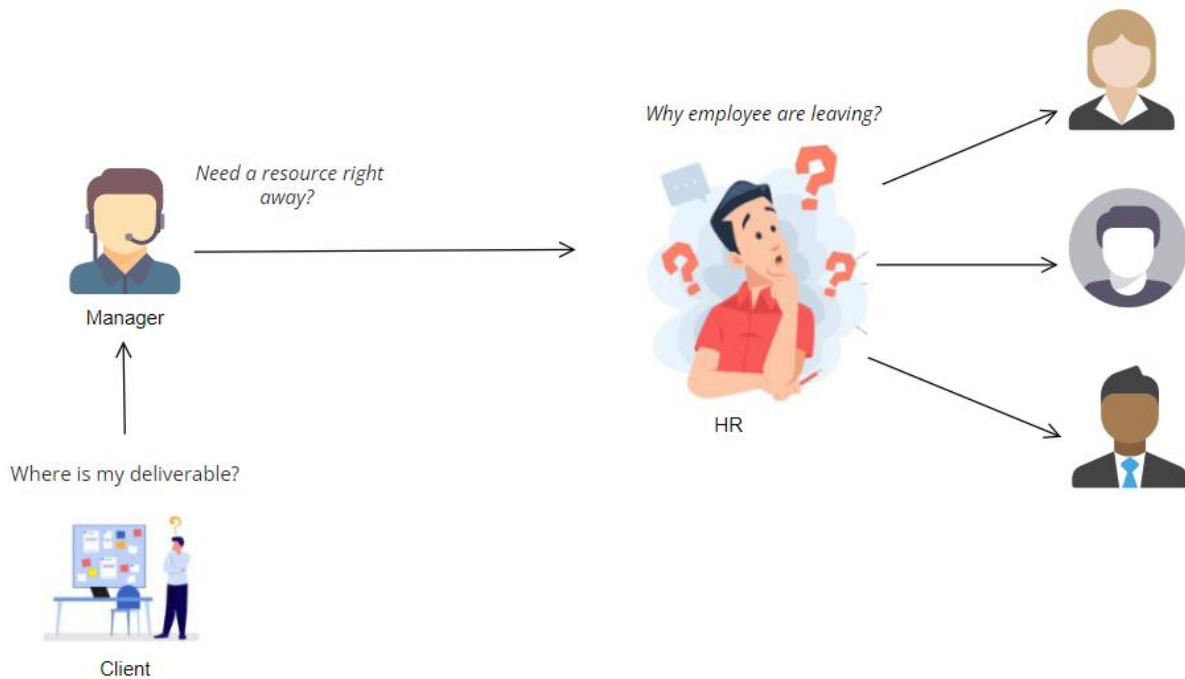
Contents

PROBLEM STATEMENT	3
SOLUTION/ ARCHITECTURE	4
ARCHITECTURE	4
POWER BI DASHBOARD	5
TECHNICAL DETAILS AND IMPLEMENTATION OF SOLUTION	6
GIT REPOSITORY	6
DATA PATTERN	6
STORAGE ACCOUNT	6
FUNCTION APP	10
SQL DATABASE	11
ANALYSIS SERVICE	13
POWER BI DESKTOP	14
CHALLENGES IN IMPLEMENTING THE SOLUTION	15
BUSINESS BENEFIT	15



Problem statement

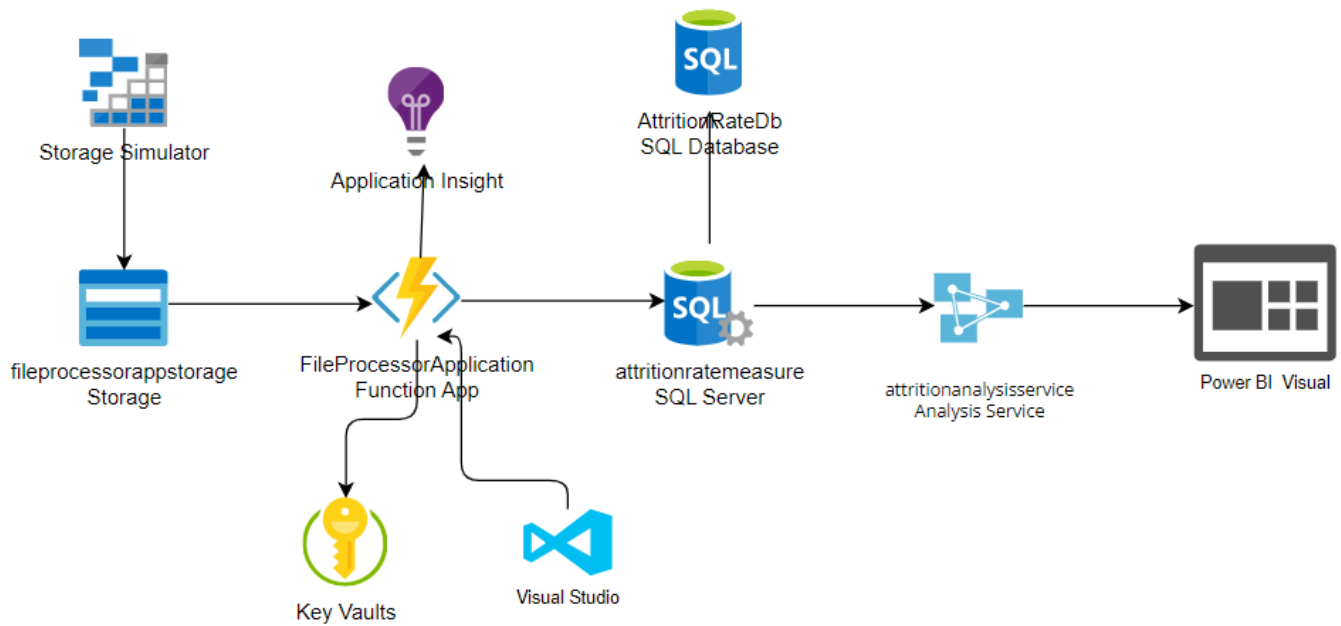
It is critical to managing staff attrition to achieve low and healthy turnover rates to preserve organizational performance and, as a result, a competitive advantage. Data-driven decisions and function app have been introduced as a result of the pressure on HR departments to give value to the firm. High employee attrition rates are now seen as a problem for businesses, putting more pressure on HR teams to keep attrition.





Solution/ Architecture

Architecture

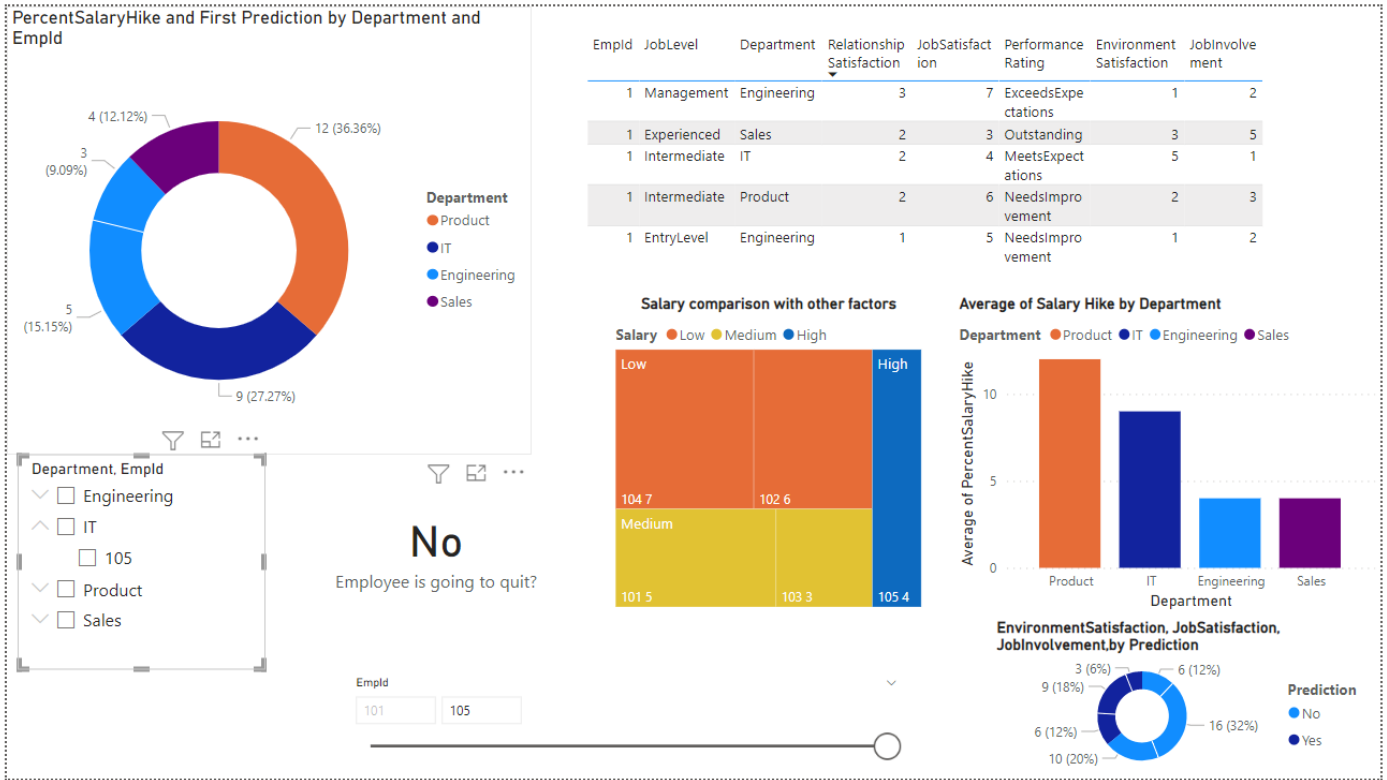


Following are the Azure service used to create the solution:

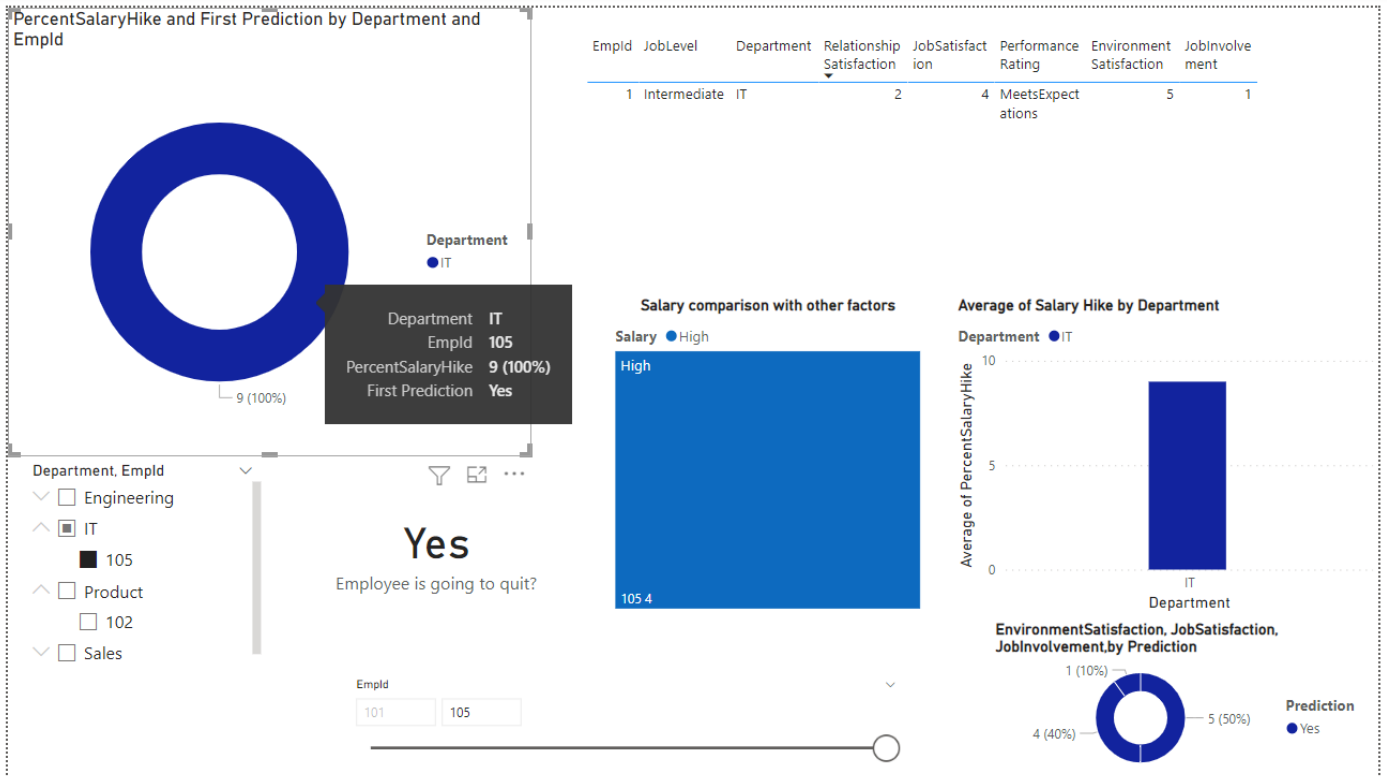
- Storage Account
- Function App
- Application insight
- Key Vault
- SQL Server
- SQL Database
- Analytic Service
- Power BI Desktop
- Storage Simulator



Power BI Dashboard



Single Employee





Technical Details and Implementation of solution

Git Repository

<https://github.com/SandyGhule/PREDICTING-EMPLOYEE-ATTRITION-USING-FUNCTION-APP-AND-POWER-BI.git>

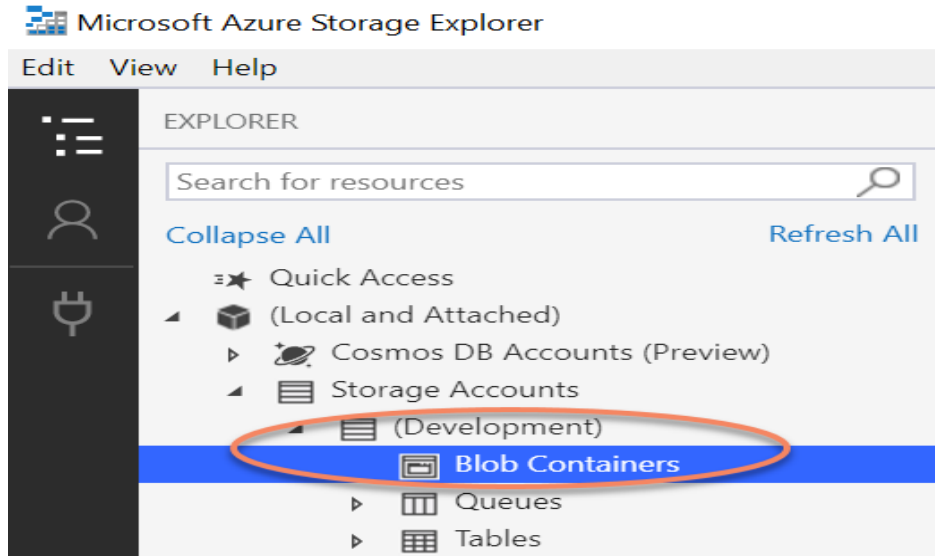
Data Pattern

Following are the several parameters that will help us make the prediction. This data is available with the HR. Some of the questions we must ask employees as part of the survey include environmental satisfaction, job satisfaction, job involvement, and so on.

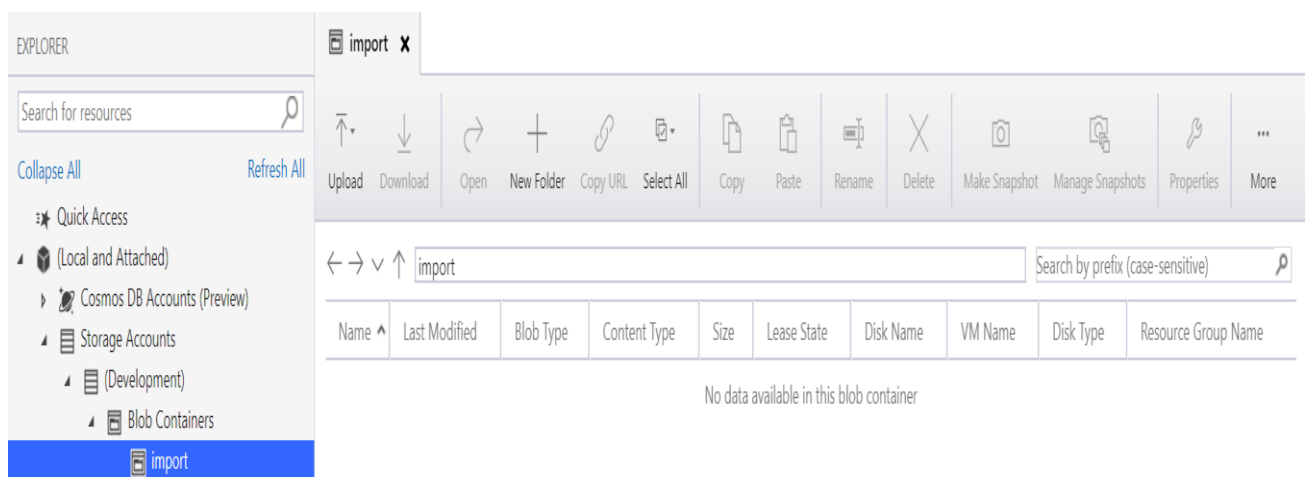
Employee No.	XXXX
Salary	Low/Medium/High
Age	22-60
DistanceFromHome	40-1
Education	Graduate/PostGraduate/PGDiploma
EnvironmentSatisfaction	0-5
JobInvolvement	0-5
JobLevel	EntryLevel/Intermediate/Experienced/Management/SeniorManagement
JobSatisfaction	0-10
Department	Sales/Engineering/Product/Support/Admin/IT/Finance/Management/Marketing
NumCompaniesWorked	0-8
PercentSalaryHike	0-50
PerformanceRating	Unacceptable/NeedsImprovement/MeetsExpectations/ExceedsExpectations/Outstanding
RelationshipSatisfaction	0-5
TotalWorkingYears	0-40
TrainingTimesLastYear	10-120
WorkLifeBalance	Poor/Fair/Good
YearsAtCompany	0-30
YearsInCurrentRole	0-15
YearsSinceLastPromotion	0-10
YearsWithCurrManager	0-20

Storage Account

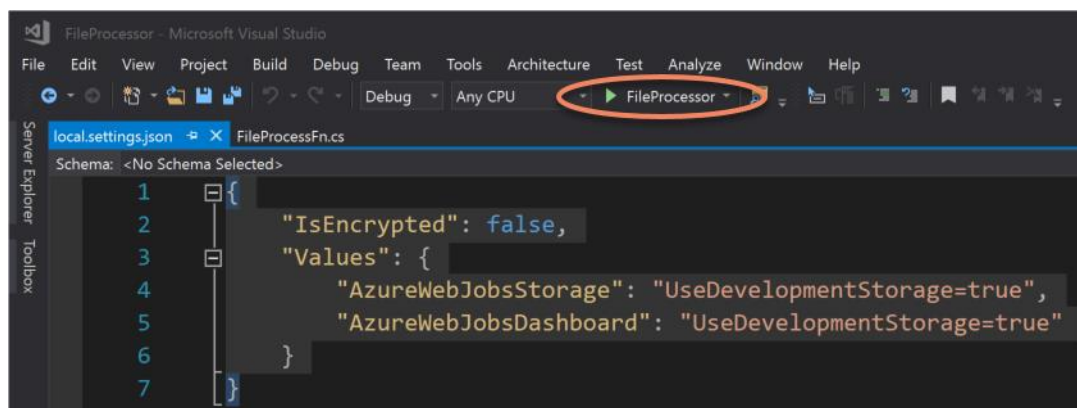
- To test on a local machine
 1. Launch the Storage Emulator.
 2. Open Storage Explorer and navigate to Blob Containers in developer storage.



3. Right-click on Blob Containers and choose Create Blob Container. This opens a node that you can type the name for the container: `import`. Hit `ENTER` and the container details load.



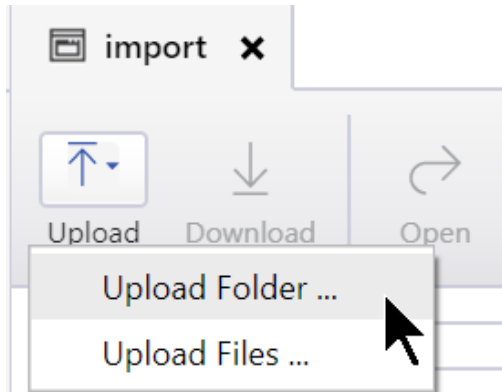
4. In Visual Studio, click the debug button or press `F5` to start debugging.



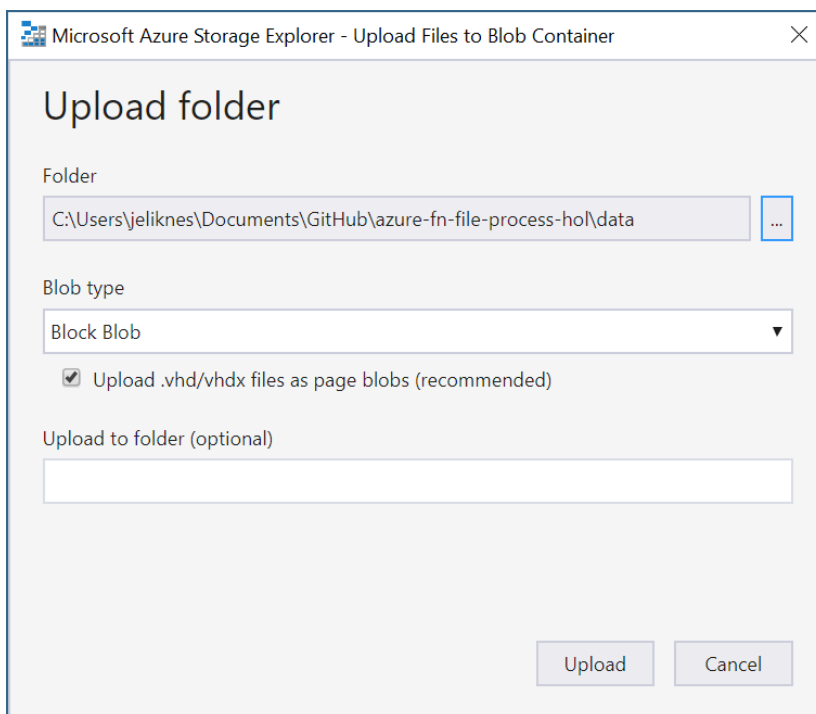
5. Wait for the functions host to start running. The console eventually shows the text `Debugger listening on [::]:5858` (your port may be different.)



6. In the Storage Explorer window for the `import` container, click the `Upload` button and choose the `Upload folder...` option.



7. In the `Upload Folder` dialog, select the `data` folder that is provided with this tutorial. Make sure `Blob type` is set to `Block blob` and `Upload to folder (optional)` is empty. Click `Upload`.



8. Confirm the files in the folder were processed by checking the logs in the function host console window.
 9. Stop the debugging session
 10. Delete the `data` folder and files from the storage emulator.
- On Azure Cloud
 1. Create storage account on portal
 2. Create container with name as `import`



import ...

Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot

Overview

Authentication method: Access key (Switch to Azure AD User Account)

Location: import

Search blobs by prefix (case-sensitive) Show deleted blobs

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state	
<input type="checkbox"/> Hr_Data.csv	1/13/2023, 8:39:49 PM			Block blob	803 B	Available	...

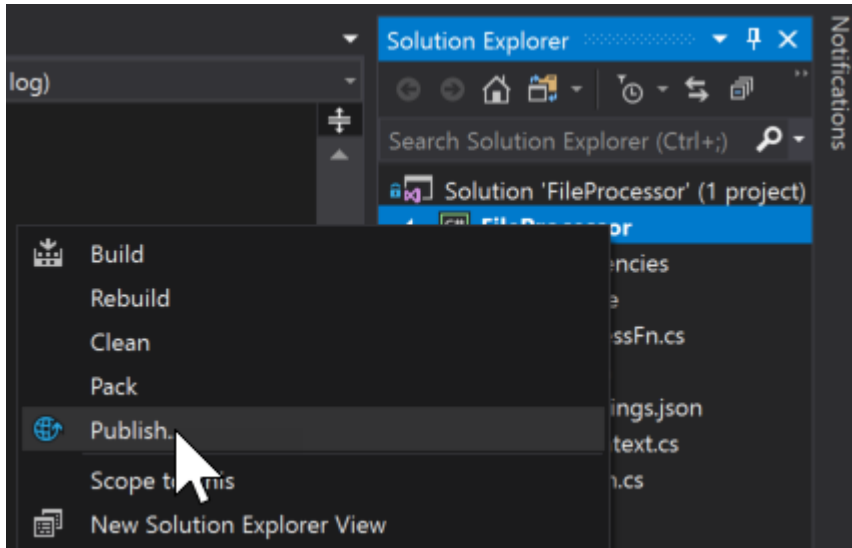
Settings

- Shared access tokens
- Access policy
- Properties
- Metadata

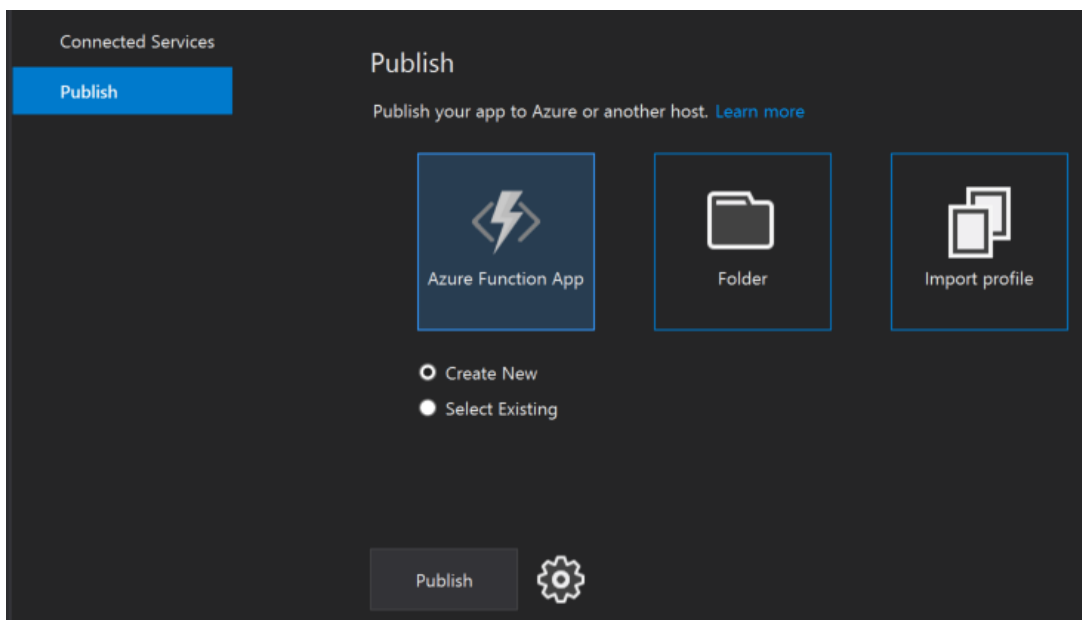


Function App

1. Inside Visual Studio, from the Solution Explorer right-click on the project name and choose Publish....



2. Choose Azure Function App, check Create New, and click Publish.



3. Give the app a unique name, choose your Subscription, and select the same Resource Group. For App Service Plan click New....
4. Give the plan a unique name, choose the Location, and pick Consumption for the Size. Click OK.
5. Back in the Create App Service dialog, click Create.



Home > FileProcessorApplication

FileProcessorApplication | Functions ...

Function App

Search

+ Create Refresh Delete

Use your local development environment to edit this Function App. This Function App was created in a local environment and cannot be edited in the Azure portal.

Filter by name...

Name ↑↓	Trigger ↑↓	Status ↑↓	Monitor ↑↓
FileProcessFn	Blob	Enabled	Invocations and more

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Events (preview)

Functions

Functions

App keys

App files

Proxies

Deployment

Deployment slots

Deployment Center

- The publish shows build output and eventually the text `Publish completed.` when it's done.
- Open your Azure SQL Database in the Azure portal and navigate to `Connection Strings`. Copy the connection string for `ADO.NET`.
- Navigate to the function app in the portal. Click `Application settings`.
- Scroll to the `Connection strings` section. Click `+ Add new connection string`. Type `TodoContext` for the name, paste the value from step 7 (be sure to update `{your_username}` and `{your_password}` to the correct values), and set the type to `SQLAzure`.

Home > FileProcessorApplication

FileProcessorApplication | Configuration ...

Function App

Search

Refresh Save Discard Leave Feedback

Deployment slots

Deployment Center

Settings

Configuration

Authentication

Application Insights

Identity

Backups

Custom domains

Custom domains (classic)

Certificates

TLS/SSL settings (classic)

Networking

Scale up (App Service plan)

Scale out

Push

Properties

Name	Value	Source	Deployment slot setting	Delete	Edit
APPINSIGHTS_INSTRUMENTATIONKEY	Hidden value. Click to show value	App Service	✓	🗑️	✎️
APPLICATIONINSIGHTS_CONNECTION_STRING	Hidden value. Click to show value	App Service		🗑️	✎️
AzureWebJobsStorage	Hidden value. Click to show value	App Service		🗑️	✎️
FUNCTIONS_EXTENSION_VERSION	Hidden value. Click to show value	App Service		🗑️	✎️
FUNCTIONS_WORKER_RUNTIME	Hidden value. Click to show value	App Service		🗑️	✎️
WEBSITE_CONTENTAZUREFILECONNECTIONST	Hidden value. Click to show value	App Service		🗑️	✎️
WEBSITE_CONTENTSHARE	Hidden value. Click to show value	App Service		🗑️	✎️

Connection strings

Connection strings are encrypted at rest and transmitted over an encrypted channel. Connection strings should only be used with a function app if you are using entity framework. For other scenarios use App Settings. [Learn more](#)

+ New connection string Show values Advanced edit

Filter connection strings

Name	Value	Source	Type	Deploym...	Delete	Edit
TodoContext	Hidden value. Click to	App Service	SQLAzure	✓	🗑️	✎️

- Above the `Connection strings` section is `Application settings`. Note the `AccountName` from the `AzureWebJobsStorage` entry to get the storage account name.
- Scroll to the top and click `Save`.



1. Create azure SQL database.

The screenshot shows the Azure portal interface for a newly created SQL database. The main content area displays the 'Essentials' section with the following details:

- Resource group: PowerBI_Demo
- Status: Online
- Location: Central India
- Subscription: AA-AS-EIT2-DEV-Sandbox
- Subscription ID: 4300dd39-cd19-4fb6-be0c-a67e3623a2be
- Tags: Click here to add tags
- Server name: [Redacted]
- Elastic pool: No elastic pool
- Connection strings: Show database connection strings
- Pricing tier: Basic
- Earliest restore point: 2023-01-10 11:21 UTC

Below the essentials, there is a 'Compute utilization' chart showing a 100% utilization rate. To the right, a 'Database data storage' gauge indicates that 1.12% of the space is used.

2. Go to Query Editor and login with credentials

The screenshot shows the Azure portal interface for the 'AttritionRateDb' SQL database. The left-hand navigation menu is visible, with the 'Query editor (preview)' option highlighted. The main content area shows the database name and a search bar.

3. Create table with following Query

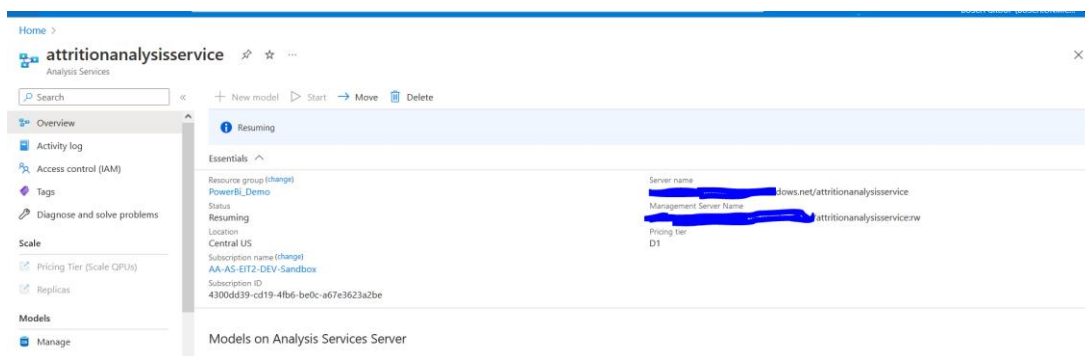
```
CREATE TABLE [dbo].[EmpDatas](
  [Id] [int] IDENTITY(1,1) NOT NULL,
  [EmpId] [int] NOT NULL,
  [Salary] [nvarchar](max) NULL,
  [Age] [int] NOT NULL,
  [DistanceFromHome] [int] NOT NULL,
  [Education] [nvarchar](max) NULL,
  [EnvironmentSatisfaction] [int] NOT NULL,
```



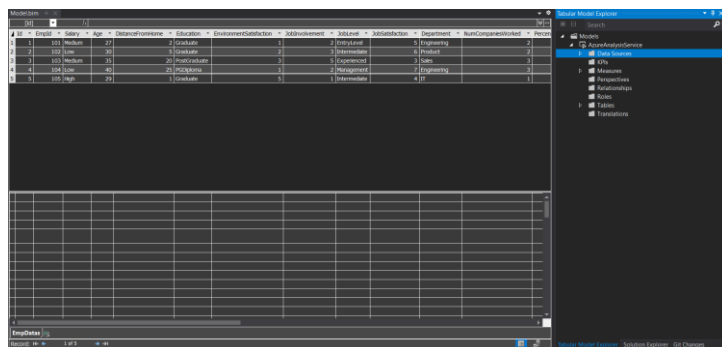
```
[JobInvolvement] [int] NOT NULL,
[JobLevel] [nvarchar](max) NULL,
[JobSatisfaction] [int] NOT NULL,
[Department] [nvarchar](max) NULL,
[NumCompaniesWorked] [int] NOT NULL,
[PercentSalaryHike] [int] NOT NULL,
[PerformanceRating] [nvarchar](max) NULL,
[RelationshipSatisfaction] [int] NOT NULL,
[TotalWorkingYears] [int] NOT NULL,
[TrainingTimesLastYear] [int] NOT NULL,
[WorkLifeBalance] [nvarchar](max) NULL,
[YearsAtCompany] [int] NOT NULL,
[YearsInCurrentRole] [int] NOT NULL,
[YearsSinceLastPromotion] [int] NOT NULL,
[YearsWithCurrManager] [int] NOT NULL,
[Prediction] [nvarchar](max) NULL)
```

Analysis Service

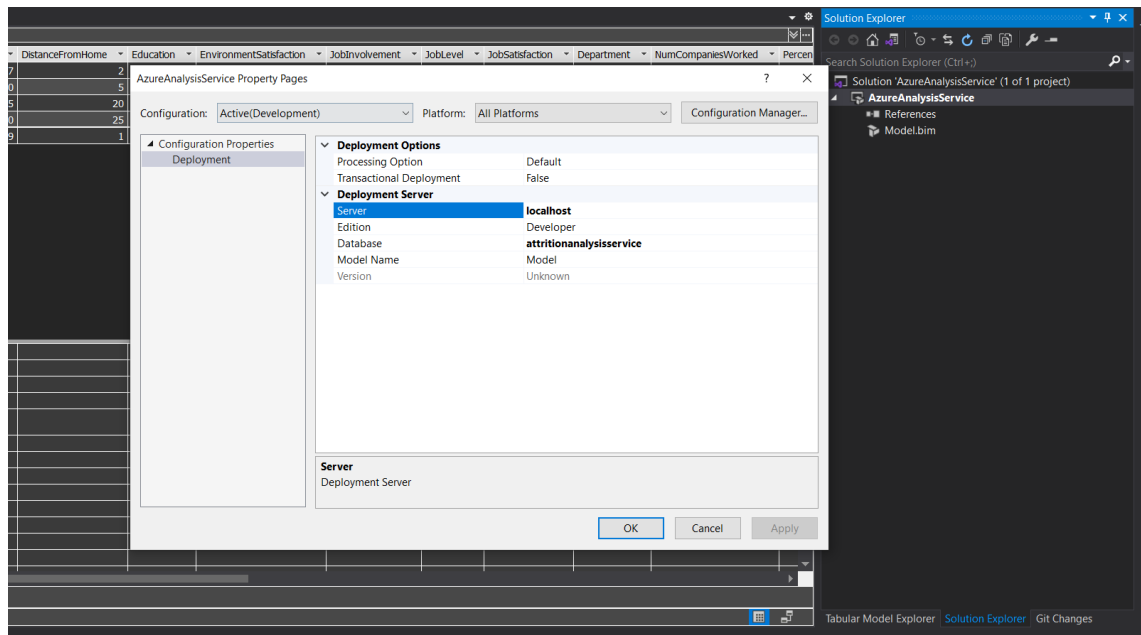
1. Create Analytic service in azure portal



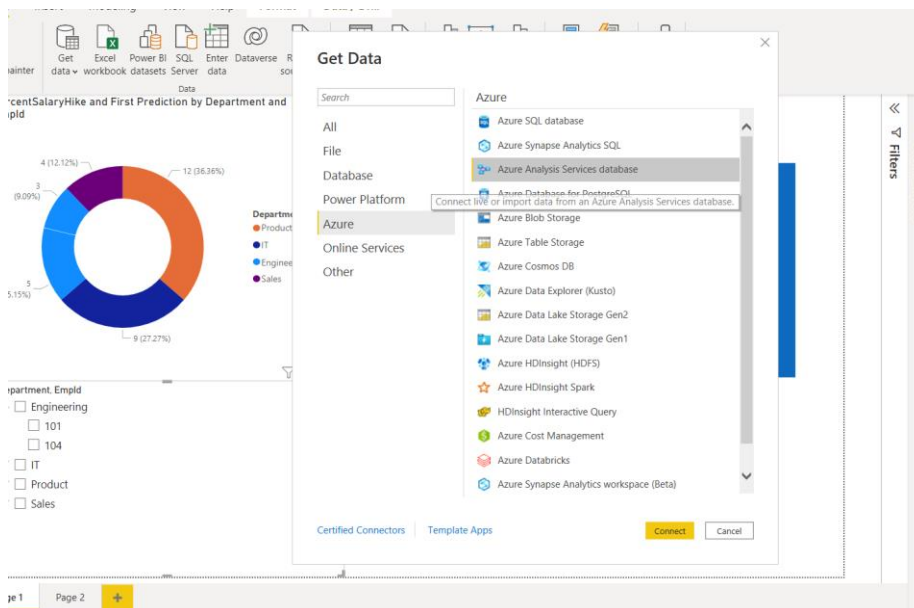
2. Connect service through visual studio



3. Change server name localhost to server name and deploy the changes.

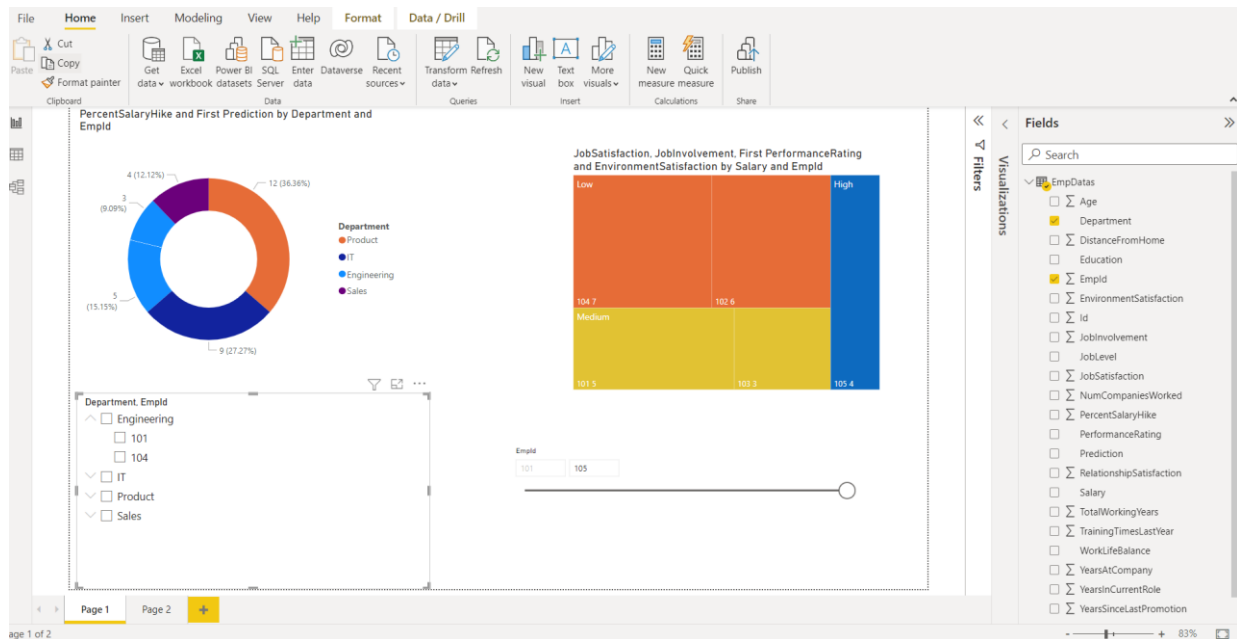


4. Connect this analysis service with Power BI desktop tool



Power BI Desktop

1. Open Power BI Desktop
2. Connect to Analysis service or Azure database
3. Create visuals according to need.



Challenges in implementing the solution

- Connect Power BI Desktop with the Analytic Service.
- I was facing issues with the SQL database connection.
- Collecting the parameters and manually creating the CSV file
- Creating different visuals in Power BI

Business Benefit

- If the company was aware of the risk of losing that resource ahead of time, they would plan ahead of time to intervene or have a road map of that resource being less critical in the upcoming project.
- Getting good visuals based on the respected data. Like we can compare the Job Satisfaction level with Salary and Promotions.
- The more years a person spends in a role, the more knowledge and expertise they acquire, which translates to higher productivity levels and an increase in revenue.
- Long-serving employees contribute to a great employee experience as they can mentor junior members of staff.
- Long-serving employees are usually well-aligned with the organisation they work for and share similar values and attitudes.
- when people join a company, different roles require different levels of training and it's highly expensive.
- Employee engagement is a concept that describes the emotional connection a person has with their place of work.
- Often, the longer your employees stay in your organisation the better they get at their job. Additionally, they may have also worked in several roles and consequently have a thorough understanding of how different departments work.
- Depending on a person's role it can take up to 8 months for them to reach an optimal level of productivity