# Documentation: PBI Deployment Tool

## Overview

The PBI Deployment Tool is a .NET-based application designed to streamline the deployment of Power BI semantic models in environments hosted on Azure Kubernetes Service (AKS). The tool is executed via UBS Deploy, ensuring seamless integration into your CI/CD pipeline. The application processes TMDL format files to dynamically build and deploy logical semantic models to the appropriate Power BI workspace.

## Key Features

* Ownership Validation:  
  - The application checks the ownership of the semantic model prior to deployment.  
  - If the model exists and is owned by another user, the application automatically takes ownership to proceed.
* Dynamic Logical Model Deployment:  
  - Reads TMDL files to dynamically build and deploy the logical model.  
  - Deploys the model to the environment-specific Power BI workspace as specified in the configuration.
* Parameter Population for Databricks:  
  - Post-deployment, the application populates model parameters with Databricks connection information, including:  
   - Instance: Databricks instance URL.  
   - Path: SQL path for querying.  
   - Database Name: Target database for the model.  
  - This information is managed through the environment-specific configuration file.
* Gateway and Data Source Configuration:  
  - Identifies and sets the gateway ID and data source ID for the semantic model.
* Large Semantic Model Configuration:  
  - Configures the deployed model as a 'Large Semantic Model,' enabling support for datasets larger than the 10GB size limitation in Power BI.
* Environment-Specific Configurations:  
  - Each environment (Dev, QA, PPRD, Prod) uses a separate configuration file.  
   - The configuration file contains parameters such as:  
   - Power BI workspace connection details.  
   - Databricks instance and database information.  
   - Authentication credentials (Client ID, Tenant ID, Secret, Authority).  
   - Deployment options (e.g., DeployPartitions, DeployRoles).

## Application Workflow

1. Initialization:  
   - The application reads the environment-specific configuration file.  
   - Validates the TMDL file input.
2. Ownership Validation:  
   - Checks if the semantic model exists in the target workspace.  
   - Takes ownership if the model is owned by another user.
3. Logical Model Deployment:  
   - Parses the TMDL file to build the logical model.  
   - Deploys the model to the specified Power BI workspace.
4. Post-Deployment Tasks:  
   - Populates Databricks connection parameters in the deployed model.  
   - Configures the gateway and data source IDs.  
   - Sets the model as a 'Large Semantic Model.'

## Configuration File Details

### Structure

{  
 "AppSettings": {  
 "PBI": {  
 "SemanticModelName": "[Model Name]",  
 "ConnectionString": "[Power BI API Connection String]",  
 "Workspace": "[Power BI Workspace Name]",  
 "TmdlModelPath": "[Path to TMDL File]",  
 "AggregationTableName": "[Aggregation Table Name]"  
 },  
 "Databricks": {  
 "Instance": "[Databricks Instance URL]",  
 "SqlPath": "[SQL Path]",  
 "DatabaseName": "[Database Name]"  
 },  
 "Authentication": {  
 "ClientId": "[Azure AD Client ID]",  
 "TenantId": "[Azure AD Tenant ID]",  
 "SecretStoreFile": "[Path to Secret Store File]",  
 "Authority": "[Azure AD Authority URL]",  
 "Scope": "[Power BI API Scope]",  
 "Secret": "[Secret Key]"  
 },  
 "DeploymentOptions": {  
 "DeployPartitions": false,  
 "DeployRoles": false  
 }  
 }  
}

## Prerequisites

* Azure AD Credentials:  
  - Ensure proper permissions for accessing and deploying Power BI models.
* TMDL Files:  
  - TMDL files must be prepared with the logical model structure.
* Databricks Setup:  
  - Ensure Databricks instance and database are correctly configured.
* UBS Deploy Integration:  
  - The UBS Deploy tool must be set up to execute the PBI Deployment Tool in the AKS cluster.

## Troubleshooting

* Ownership Issues:  
  - If the application cannot take ownership, ensure proper permissions in the Power BI workspace.
* Configuration Errors:  
  - Validate the configuration file for missing or incorrect values.
* Databricks Connection:  
  - Verify Databricks instance, SQL path, and database name in the configuration file.
* Large Semantic Model:  
  - Ensure sufficient capacity in Power BI Premium for large datasets.

## Future Enhancements

* Automate validation of configuration files during pipeline execution.
* Enhance logging for detailed tracking of deployment steps.
* Integrate with monitoring tools to track deployment success and performance.

## CI/CD Process

1. Code and Model Updates:  
   - Developers make changes to the Power BI model using Tabular Editor and save the changes in TMDL format.  
   - Changes are committed and pushed to a GitLab repository.
2. Pipeline Execution:  
   - The GitLab CI/CD pipeline is triggered upon changes being pushed.  
   - The pipeline performs the following stages:  
    - Validation Stage: Validates the TMDL file for syntax and format correctness.  
    - Build Stage: Builds a Docker image containing the PBI Deployment Tool.  
    - Push Stage: Pushes the Docker image to a container registry.
3. Deployment Request:  
   - Once the CI/CD pipeline completes, a request is sent to the Big Data Team to deploy the model.
4. Deployment Execution:  
   - The Big Data Team uses UBSDeploy to retrieve the Docker image and deploy the application in AKS.  
   - The PBI Deployment Tool is executed in the target environment (e.g., Dev, QA, PPRD, Prod).
5. Post-Deployment Tasks:  
   - The application performs additional configuration tasks, including:  
    - Setting the gateway and data source IDs.  
    - Updating Databricks parameters (e.g., instance, path, and database name).  
    - Enabling the model as a Large Semantic Model to handle datasets exceeding the 10GB size limitation.