Feature Utilization Transition

Resouces

Repository

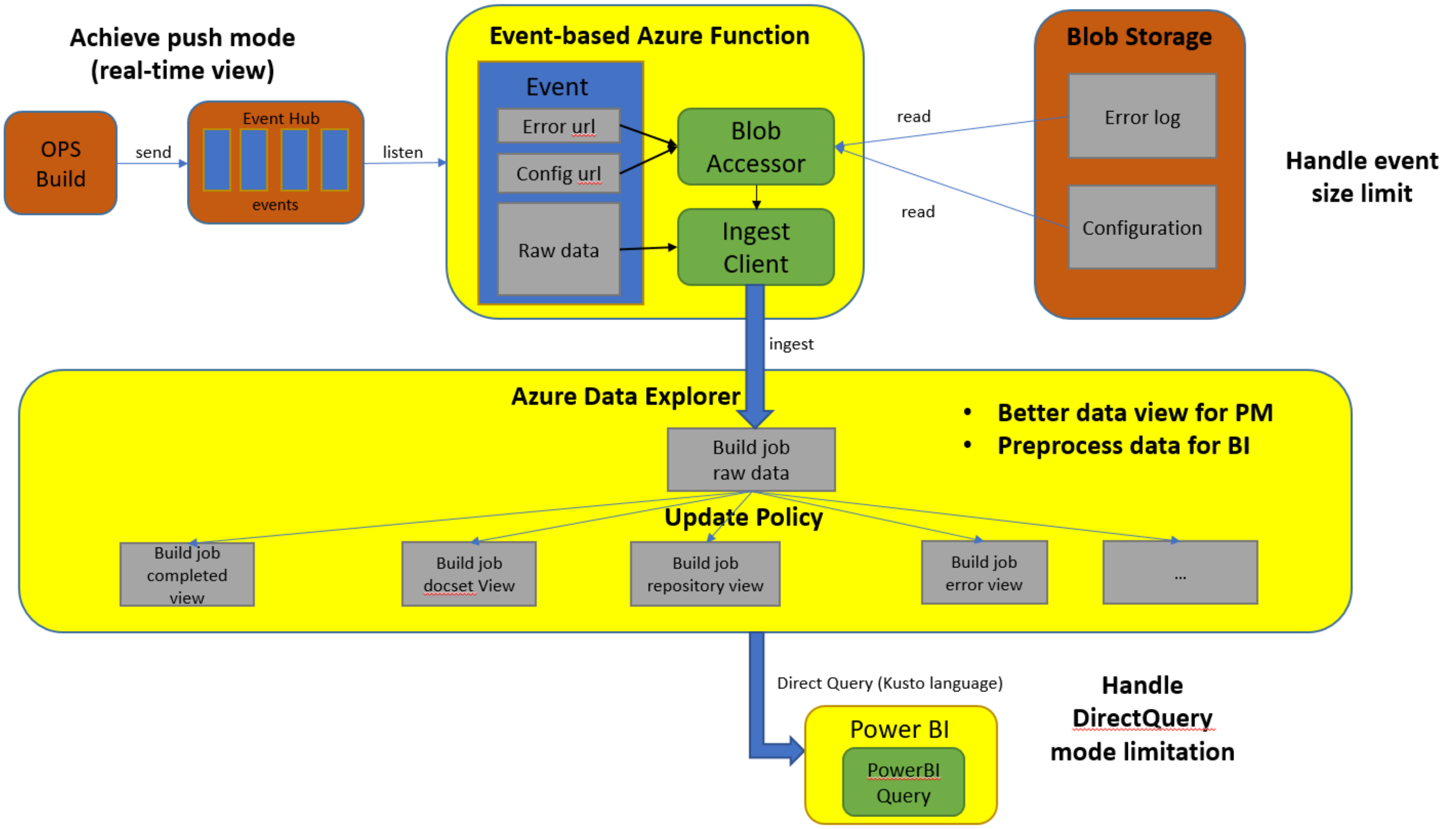
https://ceapex.visualstudio.com/Engineering/_git/Docs.FeatureUtilization (working branch, details in readme file)

Azure Resouces (PPE)

1. Subscription
   Id: 939d5667-b96d-4142-b5d5-6449f5bd0bc3
   Name: Open Publishing - Portal Sandbox
   Resource group: featureUtilizationProject
2. Resource List

Type	Resource Name	Function
Key vault	FeatureUtilization	Store credentials
Azure Data Explorer Cluster	featureutilization	Main data storage
Data factory (V2)	featureUtilizationADF	Used for docfx tracing synchronization task
App Service	SyncBuildJobProd	Used for build job events synchronization task
Application Insights	SyncBuildJobProd	Used for monitoring App Service
App Service plan	EastUSPlan	App service consumer plan
Storage account	featureutilization	Storage account for logs etc.

Workflow



1.Use event-based Azure Function to listen buildjob raw data from Event Hub.

2.Read error log, configuration from blob.

3.Ingest buildjob raw data in Azure Data Explorer (Kusto).

4.Use Azure Data Explorer update policy to generate subtable data view for better query experience.

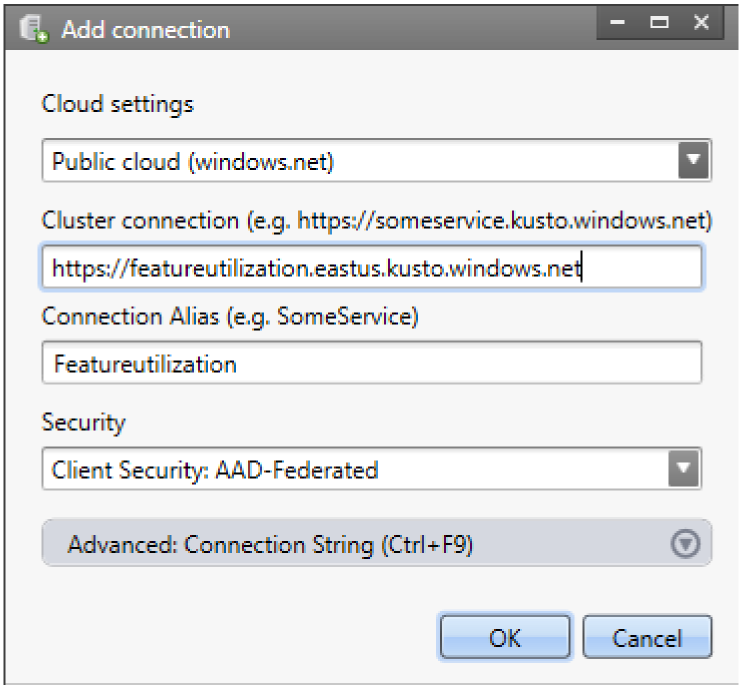
5.Power BI directly queries data from Kusto to support real-time view.

Guidelines

Azure Data Explorer (Kusto)

1. Working Place

- (Recommend) Use Kusto Explorer (download). Add connection to the database.



Cloud settings

Public cloud (windows.net)

Cluster connection (e.g. https://someservice.kusto.windows.net)

https://featureutilization.eastus.kusto.windows.net

Connection Alias (e.g. SomeService)

Featureutilization

Security

Client Security: AAD-Federated

(Cluster connection: https://featureutilization.eastus.kusto.windows.net)

- Portal query url
- Web query url

2. Databases

- featureUtilizaiton (sandbox data as source)
- FeatureUtilizationProd (newest, production data as source)

3. Tables (FeatureUtilizationProd)

- BuildJobRawData : raw data from buildjob Event Hub
- BuildJobCompletedView : extracted data view from BuildJobRawData, storing completed buildjob data with publish config
- BuildJobErrorView: extracted data view from BuildJobRawData, storing buildjob error data
- BuildJobDocsetView: extracted data view from BuildJobRawData, storing buildjob docset data with related docfx config.

4. Functions
   Four functions for updating “view” table using Kusto Update Policy (refer here)

- Sample query to show the function details: .show function UpdateBuildJobCompletedView (replaced by function name here)

5. Sample queries (query language refer here)

- Find docsets that use markdig engine: BuildJobDocsetView | where docfx_config.build.markdownEngineName =="markdig" | summarize arg_max(timestamp, *) by docset_name
- Get average build duration of all completed build jobs in the last hour: BuildJobCompletedView| where timestamp > ago(1h)| summarize avg(building_duration)

6. Notice

- Not every row of data in BuildJobRawData has configuration info (publish_config and docfx_config), due to that we are currently using personal OPS Build token (doesn’t have permissions of every repo) to get configuration info from OBS Build API.
  When the configuration is written in blob with blob url in buildjob event in future, this won’t be a problem.

Power BI demos

- BuildJobTracing report: https://msit.powerbi.com/groups/me/reports/c82d72b4-ea50-4adb-b881-3da4688e8cf5/ReportSection
- Edit
  - In web: you can click in the interested visual and modify Filters in the right board.
  - In Power BI Desktop (download in Microsoft APP Store): you can download the report file in web and open it in Power BI Desktop to edit. Then publish it to web.
  - In Power BI Desktop: you can create your own report by importing the data in Kusto as source.