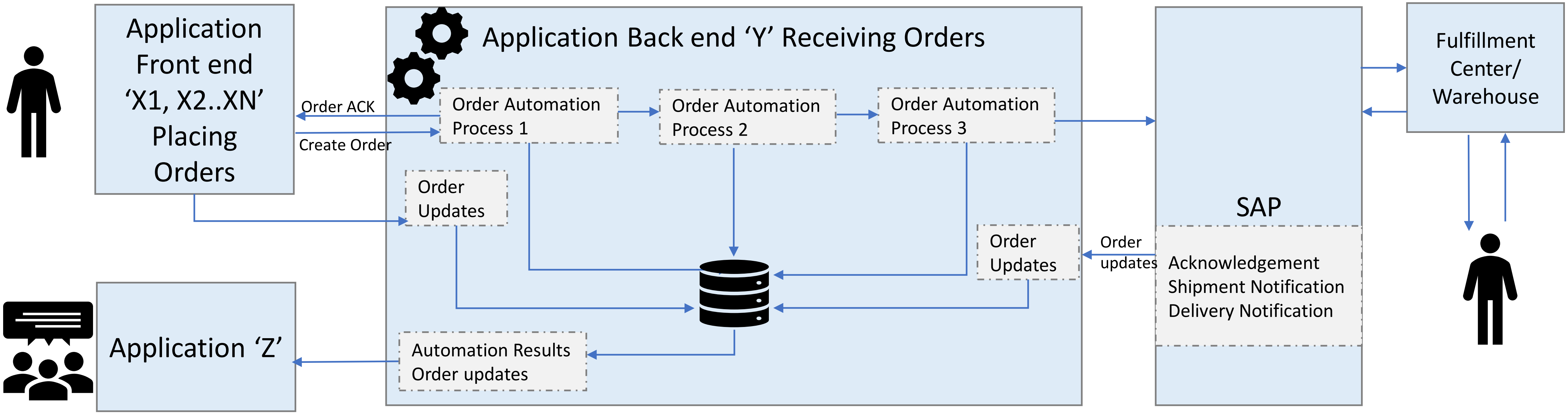
Reporting

1. Business use case Example
2. Design : Scheduled refresh reports in MS Power BI using MS Azure Active Directory
3. Design : Live Streaming reports in MS Power BI using MS Azure Active Directory & MS Azure Stream Analytics
4. Detailing all Components
5. Workflow, One-time setup - MS Azure Active Directory & Azure Stream Analytics
6. ReactJS, MS Power BI, MS Azure Active Directory, micro service – Detailed Design for Reporting
7. Scheduled reports, Live Streaming reports demo & Azure App Functions demo
8. Pre-requisites for Reporting

1. Business use case

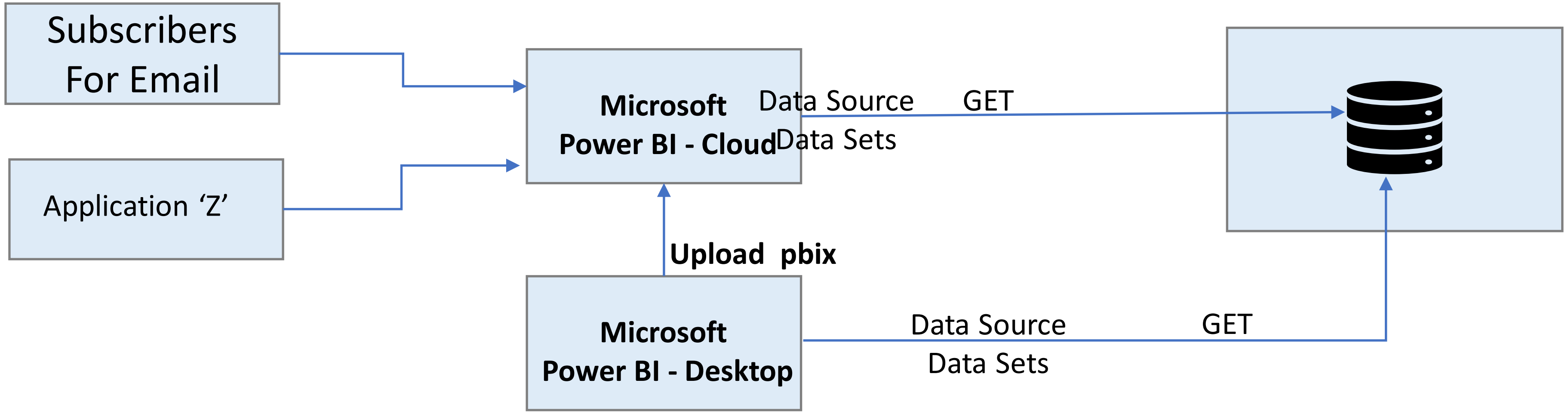

Application Front end 'X1, X2..XN' Placing Orders
Order ACK
Create Order
Application Back end 'Y' Receiving Orders
Order Automation Process 1
Order Automation Process 2
Order Automation Process 3
Order Updates
Order Updates
Automation Results Order updates
Application 'Z'
Order updates
SAP
Acknowledgement
Shipment Notification
Delivery Notification
Fulfillment Center/ Warehouse

Application Z or Business needs following

1. Number of orders placed from Application X, sent to automation for Application Y
2. How many processed in Application Y – Success, Failed, In Process…any other status
3. Reports in near real time and batched updates
4. Batched updates report with slicing and dicing available to see the order volume for any duration

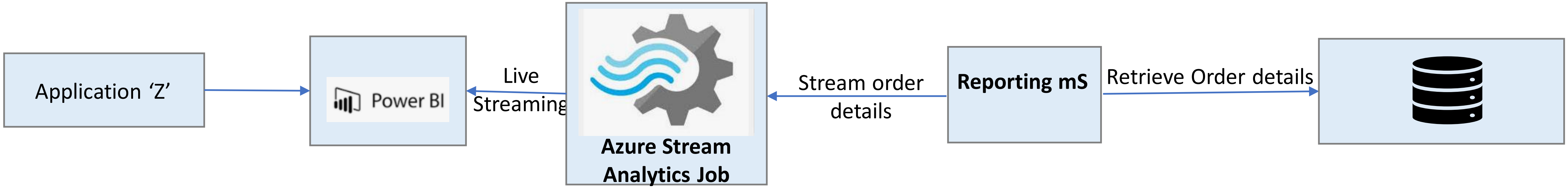
Real-time	When you need information processed immediately (such as at a bank ATM)
Near real-time	When speed is important, but you don't need it immediately
Batch	When you can wait > 5 mins

2. Scheduled refresh reports using MS Power BI, MS Azure Active Directory



- MS Power BI can be scheduled for 48 refreshes (@ 30 mins interval per hour)
- Application 'Z' to fetch refresh reports from MS Power BI @ 5th min and @ 35th min in an hr

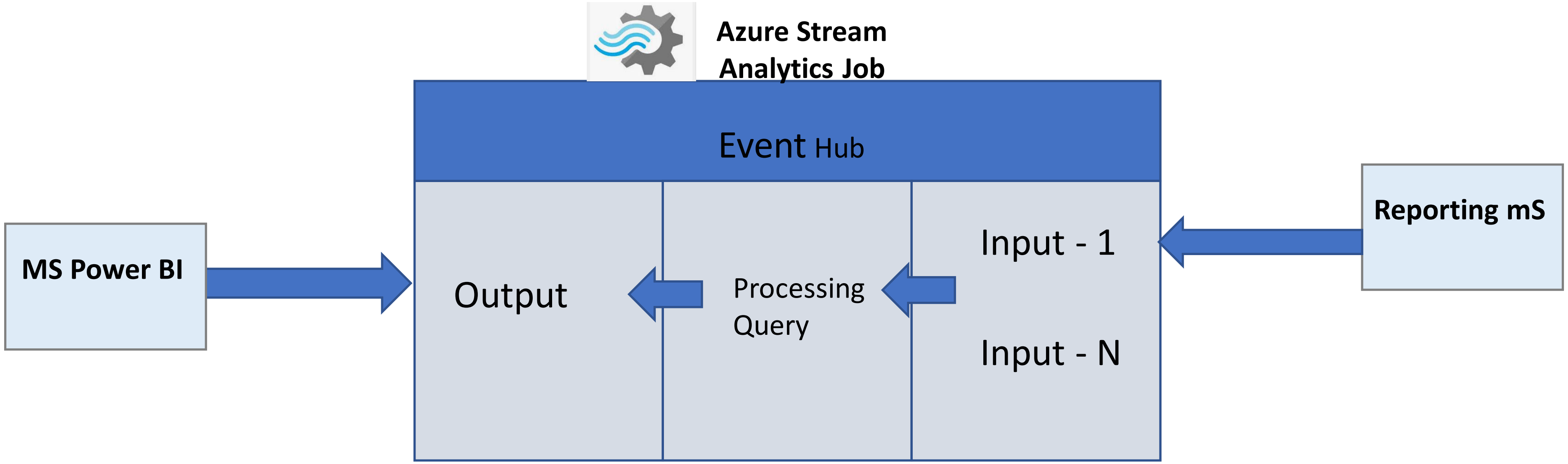
3.1 Live Streaming reports using MS Power BI, MS Azure Active Directory, MS Azure Stream Analytics



Live Streaming

- LIVE Streaming jobs configured in "Azure Stream Analytics" and shows status of orders in near real time

3.2 Reporting – Azure Stream Analysis



4.1 Component details

Components involved for Batch reporting

1. DB for data
2. MS Power BI Desktop version
    1. To create PBIX (report)
    2. Connection parameters to DB
    3. Charts as needed for display of reports
3. Application ‘Z’ for retrieving reports
4. MS Azure Active Directory
    1. Create tenant and Subscription(optional)
    2. Create users and have MS Power BI Cloud account (as Step 3)
    3. App registration
    4. Create Security group
5. MS Power BI Cloud account
    1. To create workspace
    2. To publish PBIX report to Cloud account
    3. Schedule refresh intervals
    4. Email Subscription for publishing reports via email
    5. Admin consent to ‘Allow service principals to use Power BI APIs’
    6. Provide workspace access to Security group(Service principal)

4.2 Work-flow in MS Azure Active Directory for Reporting

Components involved for Live Stream reporting

1. MS Azure Stream Analytics Job
    1. To create Input event for accepting data
    2. To create Output for MS Power BI data set
    3. Query editor to format output
2. Reporting mS
    1. To pull data from DB
    2. To publish data to MS Azure Stream Analytic Job
3. MS Power BI Cloud account
    1. To fetch data from Live Streaming data set
4. Application ‘Z’ for retrieving reports

5.1 Workflow, One-time setup - MS Azure Active Directory & MS Power BI

https://portal.azure.com

Login - emailID@org.com

Navigate to
Azure Active Directory

Create 1 user
(pbembed)

Login using pbembed

App Registrations
Register your application
“myapplication”

Above steps generates –
Client ID for the
applicaiton

Generate
Client secret code

Grant Power BI Service API
Permissions - Scope

Grant Admin Consent

Navigate to
Groups

Security group creation
“myapplication-security-group”
Security object ID

Assign client ID as member to
security group created

Final outcome
1. Client ID
2. Client Secret
3. Tenant ID
(org.onmicrosoft.com)
1. Security object ID

http://app.powerbi.com

Login - pbembed@org.com

Navigate to
Admin Portal
Possible on PRO license only

Navigate to
Tenant Settings

Enable check mark - “Allow
service principals to use
Power BI APIs”

Provide security group
“Security object ID”

Navigate to
MyWorkspace

Navigate to
Workspace Access

Navigate to
Workspace Access

Provide client ID
“Client ID”

Navigate to
MyWorkspace

Go to any report and from
URL, we can get
“report ID”

5.2 Workflow, One-time setup - MS Azure Active Directory & MS Power BI

https://portal.azure.com

Login - emailID@org.com

Navigate to
Azure Active Directory

Create new Tenant

Create 2 users
(admin, pbembed)

Login using pbembed

App Registrations
Register your application
“myapplication”

Above steps generates –
Client ID for the applicaiton

Generate
Client secret code

Grant Power BI Service API Permissions - Scope

Grant Admin Consent

Navigate to
Groups

Security group creation
“myapplication-security-group”
Security object ID

Assign client ID as member to security group created

Final outcome

1. Client ID
2. Client Secret
3. Tenant ID
   (tommjerry.onmicrosoft.com)

1. Security object ID

http://app.powerbi.com

Login - pbembed@org.com

Navigate to
Admin Portal
Possible on PRO license only

Navigate to
Tenant Settings

Enable check mark - “Allow service principals to use Power BI APIs”

Provide security group
“Security object ID”

Navigate to
MyWorkspace

Navigate to
Workspace Access

Navigate to
Workspace Access

Provide client ID
“Client ID”

Navigate to
MyWorkspace

Go to any report and from URL, we can get
“report ID”

5.3 Workflow, One-time setup - MS Azure Stream Analytics

https://portal.azure.com

Login

Navigate to Event Hubs, create new policy

Navigate to Stream Analytics Job

Navigate to Managed Identity, enable and save it

Add Inputs by giving details of Event Hub created

Add Output by selecting PBI.

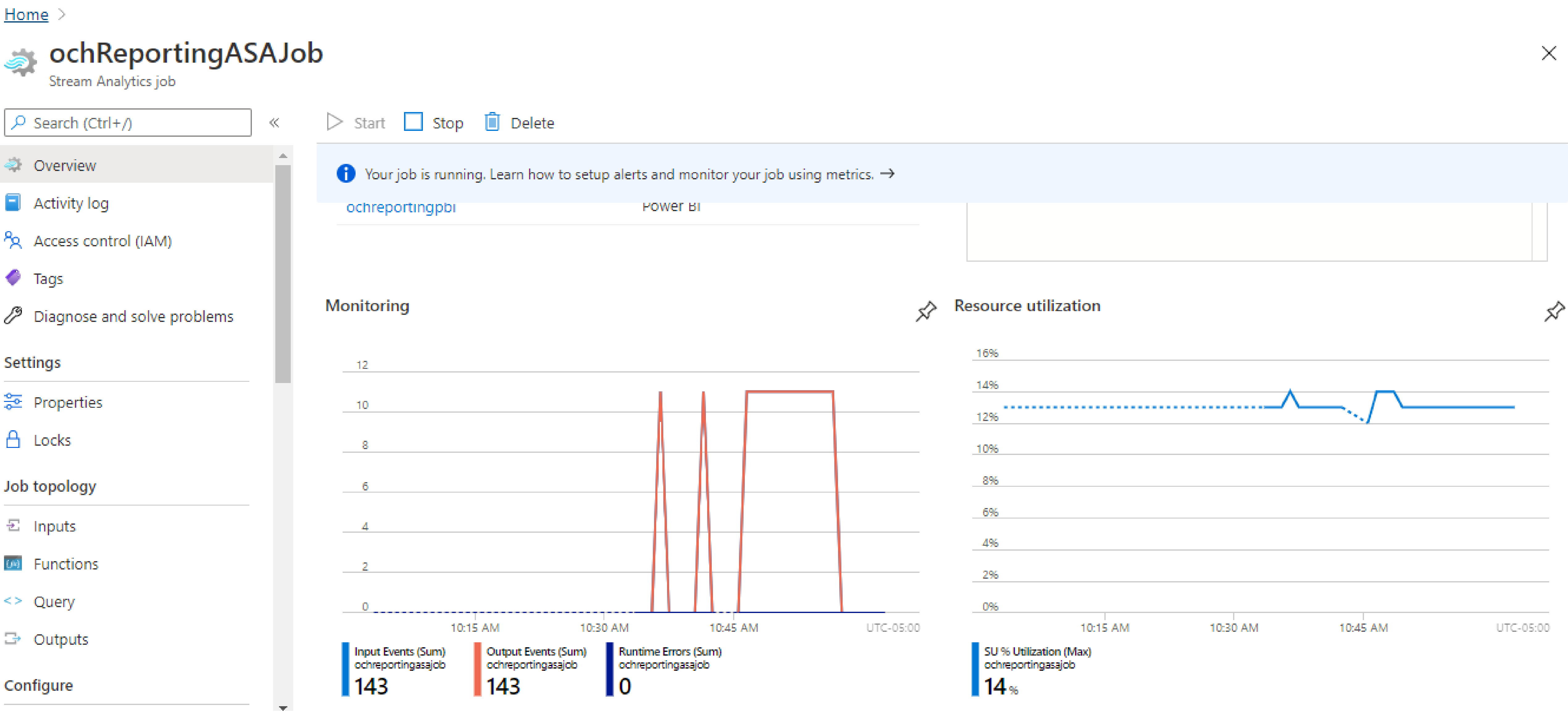
Login to PBI and to your workspace

Under workspace access, provide Stream Analytics Job name and admin to add it as Service Principal

Edit Query based on Input and Output

Start the job and copy the end points from policy to application which send live streaming

5.4 Workflow, One-time setup - MS Azure Stream Analytics



ochReportingASAJob

Stream Analytics job

Your job is running. Learn how to setup alerts and monitor your job using metrics. →

ochreportingpbi Power BI

Monitoring

Input Events (Sum) ochreportingasajob 143

Output Events (Sum) ochreportingasajob 143

Runtime Errors (Sum) ochreportingasajob 0

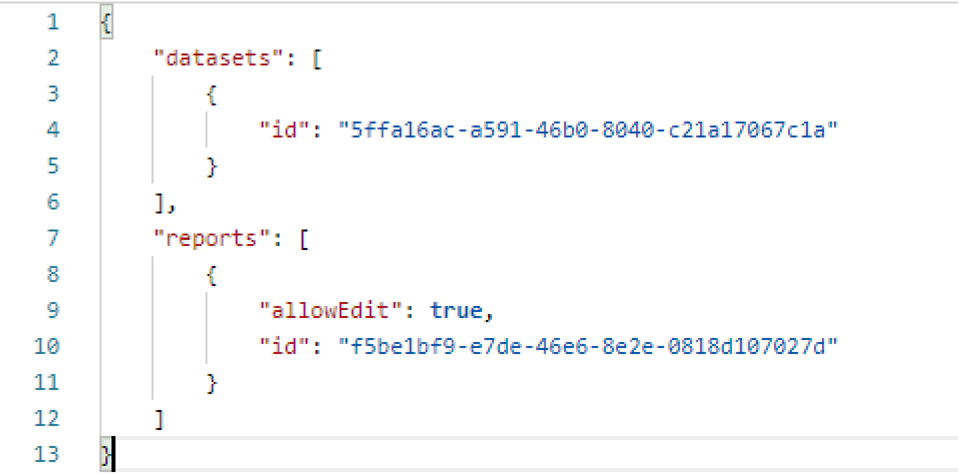
Resource utilization

SU % Utilization (Max) ochreportingasajob 14 %

6.1 POC : Report tag in ReactJS

Input from user reportNeeded = “**Prod1Rpt**”

```
<Report
        tokenType="Embed“
        accessToken= {accessToken}
        embedUrl= {embedUrl}
        embedId= {embedId}
</Report>
```



```
{
    "datasets": [
        {
            "id": "5ffa16ac-a591-46b0-8040-c21a17067c1a"
        }
    ],
    "reports": [
        {
            "allowEdit": true,
            "id": "f5be1bf9-e7de-46e6-8e2e-0818d107027d"
        }
    ]
}
```

{token}

Post API https://login.microsoftonline.com/{tenantId}/oauth2/v2.0/token

With client id, client secret, scope, grant type(client credentials)

Response – auth token from MS Identity Platform

{embedUrl}

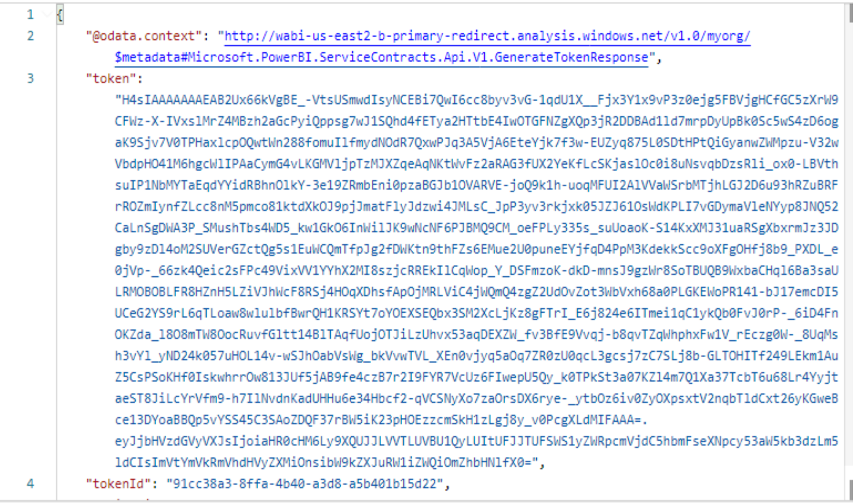
Using “reportNeeded”,

search in couchbase document to get embed URL and data set ID

{embedId}

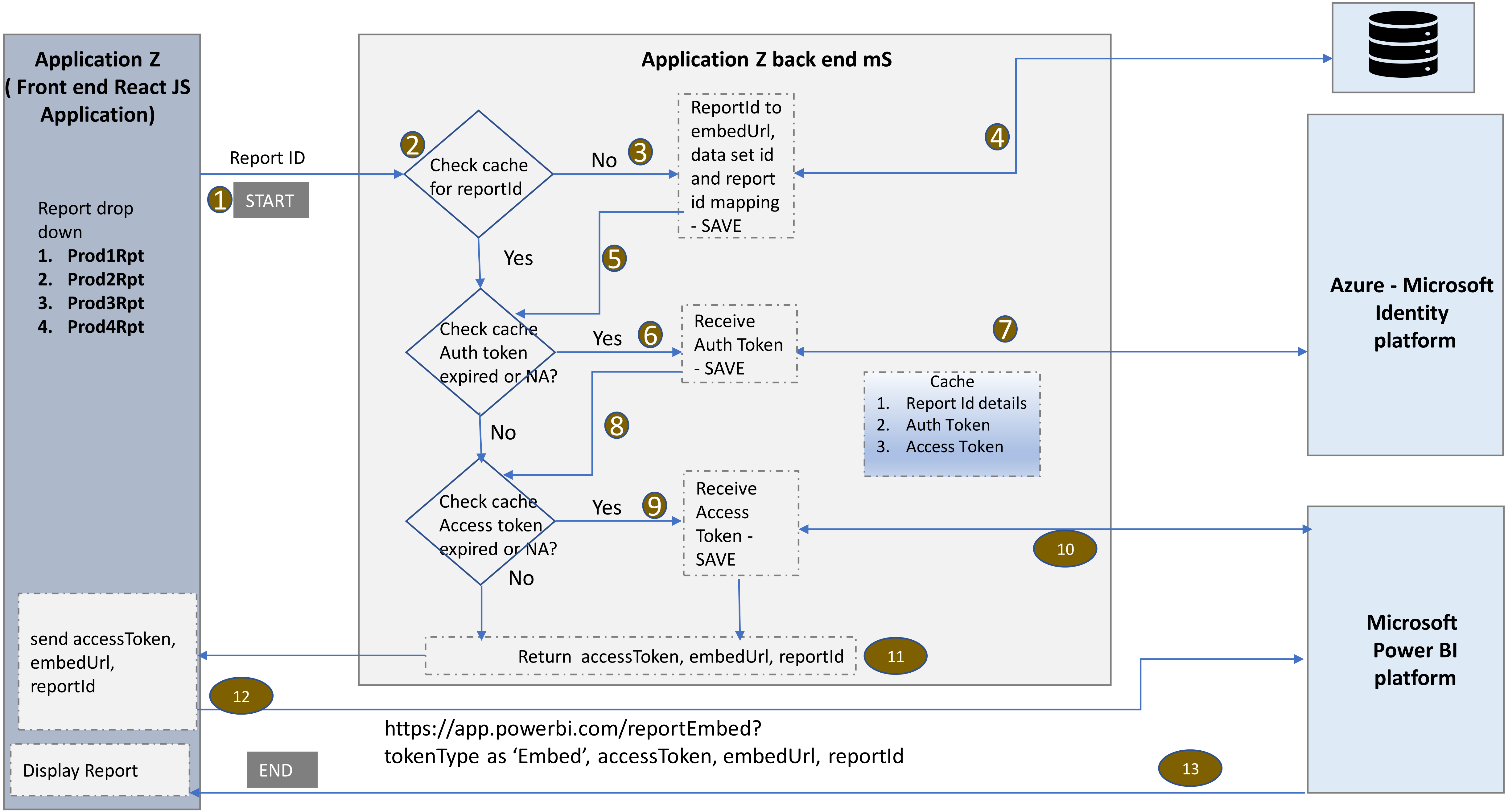
Post API https://api.powerbi.com/v1.0/myorg/GenerateToken, with datasets, report ids and auth token

Response – accessToken from MS Power BI



```
{
    "@odata.context": "http://wabi-us-east2-b-primary-redirect.analysis.windows.net/v1.0/myorg/
      $metadata#Microsoft.PowerBI.ServiceContracts.Api.V1.GenerateTokenResponse",
    "token":
      "H4sIAAAAAAAEAB2Ux66kVgBE_-VtsUSmwdIsyNCEBi7QwI6cc8byv3vG-1qdU1X__Fjx3Y1x9vP3z0ejg5FBVjgHCfGC5zXrW9
      CFWz-X-IVxslMrZ4MBzh2aGcPyiQppsg7wJ1SQhd4fETya2HTtbE4IwOTGFNZgXQp3jR2DDBAd1ld7mrpDyUpBk0Sc5wS4zD6og
      aK9Sjv7V0TPHaxlcpOQwtWn288fomuIlfmydNOdR7QxwPJq3A5VjA6EteYjk7f3w-EUZyq875L0SDtHPtQiGyanwZWMpzu-V32w
      VbdpHO41M6hgcWlIPAaCymG4vLKGMV1jpTzMJXZqeAqNKtWvFz2aRAG3fUX2YeKfLcSKjaslOc0i8uNsvqbDzsRli_ox0-LBVth
      suIP1NbMYTaEqdYYidRBhnOlkY-3e19ZRmbEni0pzaBGJb1OVARVE-joQ9k1h-uoqMFUI2AlVVaWSrbMTjhLGJ2D6u93hRZuBRF
      rROZmIynfZLcc8nM5pmco81ktdXkOJ9pjJmatFlyJdzwi4JMLsC_JpP3yv3rkjxk05JZJ61OsWdKPLI7vGDymaVleNYyp8JNQ52
      CaLnSgDWA3P_SMushTbs4WD5_kw1GkO6InWilJK9wNcNF6PJBMQ9CM_oeFPLy335s_suUoaoK-S14KxXMJ31uaRSgXbxrmJz3JD
      gby9zDl4oM2SUVerGZctQg5s1EuWCQmTfpJg2fDWKtn9thFZs6EMue2U0puneEYjfqD4PpM3KdekkScc9oXFgOHfj8b9_PXDL_e
      0jVp-_66zk4Qeic2sFPc49VixVV1YYhX2MI8szjcRREkI1CqWop_Y_DSFmzoK-dkD-mnsJ9gzWr8SoTBUQB9WxbaCHql6Ba3saU
      LRMOBOBLFR8HZnH5LZiVJhWcF8RSj4HOqXDhsfApOjMRLViC4jWQmQ4zgZ2UdOvZot3WbVxh68a0PLGKEWoPR141-bJ17emcDI5
      UCeG2YS9rL6qTLoaw8wlulbfBwrQH1KRSYt7oYOEXSEQbx3SM2XcLjKz8gFTrI_E6j824e6ITmei1qC1ykQb0FvJ0rP-_6iD4Fn
      OKZda_18O8mTW8OocRuvfGltt14BlTAqfUojOTJiLzUhvx53aqDEXZW_fv3BfE9Vvqj-b8qvTZqWhphxFw1V_rEczg0W-_8UqMs
      h3vYl_yND24k057uHOL14v-wSJhOabVsWg_bkVvwTVL_XEn0vjyq5aOq7ZR0zU0qcL3gcsj7zC7SLj8b-GLTOHITf249LEkm1Au
      Z5CsPSoKHf0IskwhrrOw813JUf5jAB9fe4czB7r2I9FYR7VcUz6FIwepU5Qy_k0TPkSt3a07KZl4m7Q1Xa37TcbT6u68Lr4Yyjt
      aeST8JiLcYrVfm9-h7IlNvdnKadUHHu6e34Hbcf2-qVCSNyXo7zaOrsDX6rye-_ytbOz6iv0ZyOXpsxtV2nqbTldCxt26yKGweB
      ce13DYoaBBQp5vYSS45C3SAoZDQF37rBW5iK23pHOEzzcmSkH1zLgj8y_v0PcgXLdMIFAAA=.
      eyJjbHVzdGVyVXJsIjoiaHR0cHM6Ly9XQUJJLVVTLUVBU1QyLUItUFJJTUFSWS1yZWRpcmVjdC5hbmFseXNpcy53aW5kb3dzLm5
      ldCIsImVtYmVkRmVhdHVyZXMiOnsibW9kZXJuRW1iZWQiOmZhbHNlfX0=",
    "tokenId": "91cc38a3-8ffa-4b40-a3d8-a5b401b15d22",
```

6.2 ReactJS and MS Power BI Integration for Reporting



Application Z (Front end React JS Application)

Report drop down

1. **Prod1Rpt**
2. **Prod2Rpt**
3. **Prod3Rpt**
4. **Prod4Rpt**

Report ID

1 START

Application Z back end mS

2 Check cache for reportId

No 3

ReportId to embedUrl, data set id and report id mapping - SAVE

4

Yes

5

Check cache Auth token expired or NA?

Yes 6

Receive Auth Token - SAVE

7

Cache

1. Report Id details
2. Auth Token
3. Access Token

No

8

Check cache Access token expired or NA?

Yes 9

Receive Access Token - SAVE

10

No

Return accessToken, embedUrl, reportId

11

send accessToken, embedUrl, reportId

12

https://app.powerbi.com/reportEmbed?
tokenType as ‘Embed’, accessToken, embedUrl, reportId

Azure - Microsoft Identity platform

Microsoft Power BI platform

13

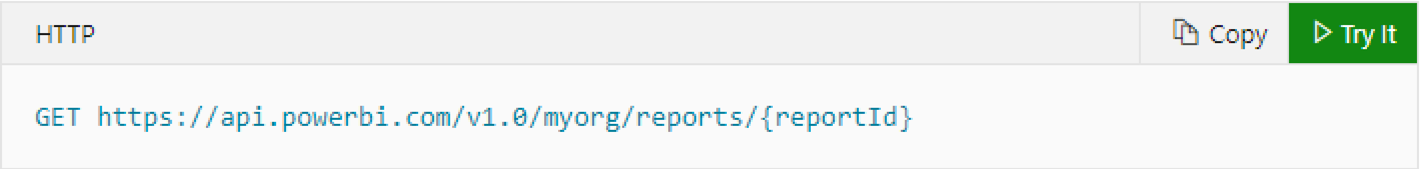
Display Report

END

6.3 POC : One time-Setup for each report – embed url and data set id

1. Navigate to below URL and Sign in using the new user created
https://docs.microsoft.com/en-us/rest/api/power-bi/reports/getreport

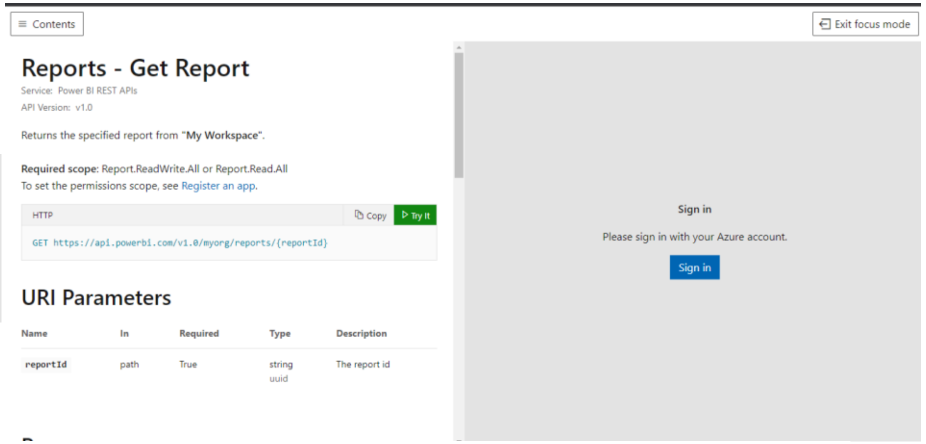
2. Click “Try it” and



```
GET https://api.powerbi.com/v1.0/myorg/reports/{reportId}
```

3. Sign in and in Parameters tab, provide “reportid” from Power BI url of the report. Click run to execute the api

4. The response will have embed url and data set id.



Reports - Get Report

Service: Power BI REST APIs
API Version: v1.0

Returns the specified report from "**My Workspace**".

Required scope: Report.ReadWrite.All or Report.Read.All
To set the permissions scope, see Register an app.

HTTP

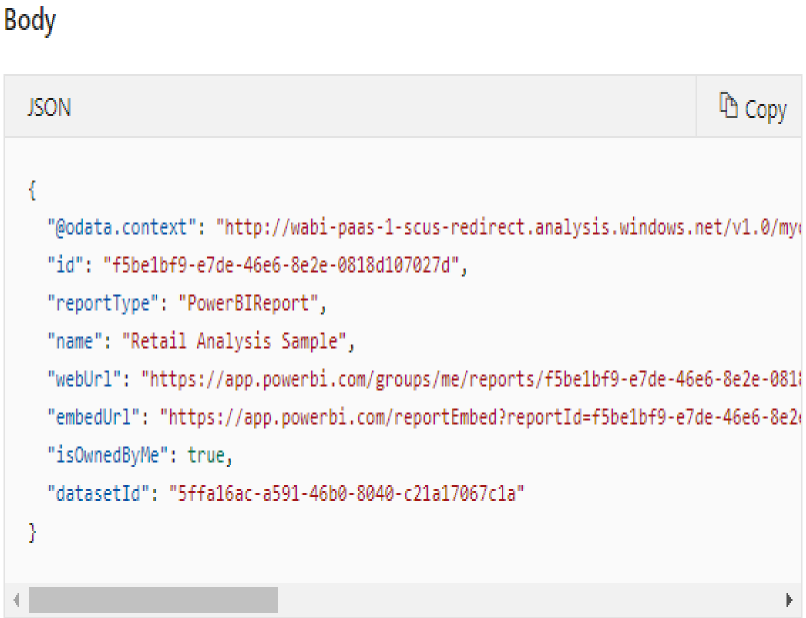
```
GET https://api.powerbi.com/v1.0/myorg/reports/{reportId}
```

URI Parameters

Name	In	Required	Type	Description
reportId	path	True	string uuid	The report id

Sign in

Please sign in with your Azure account.



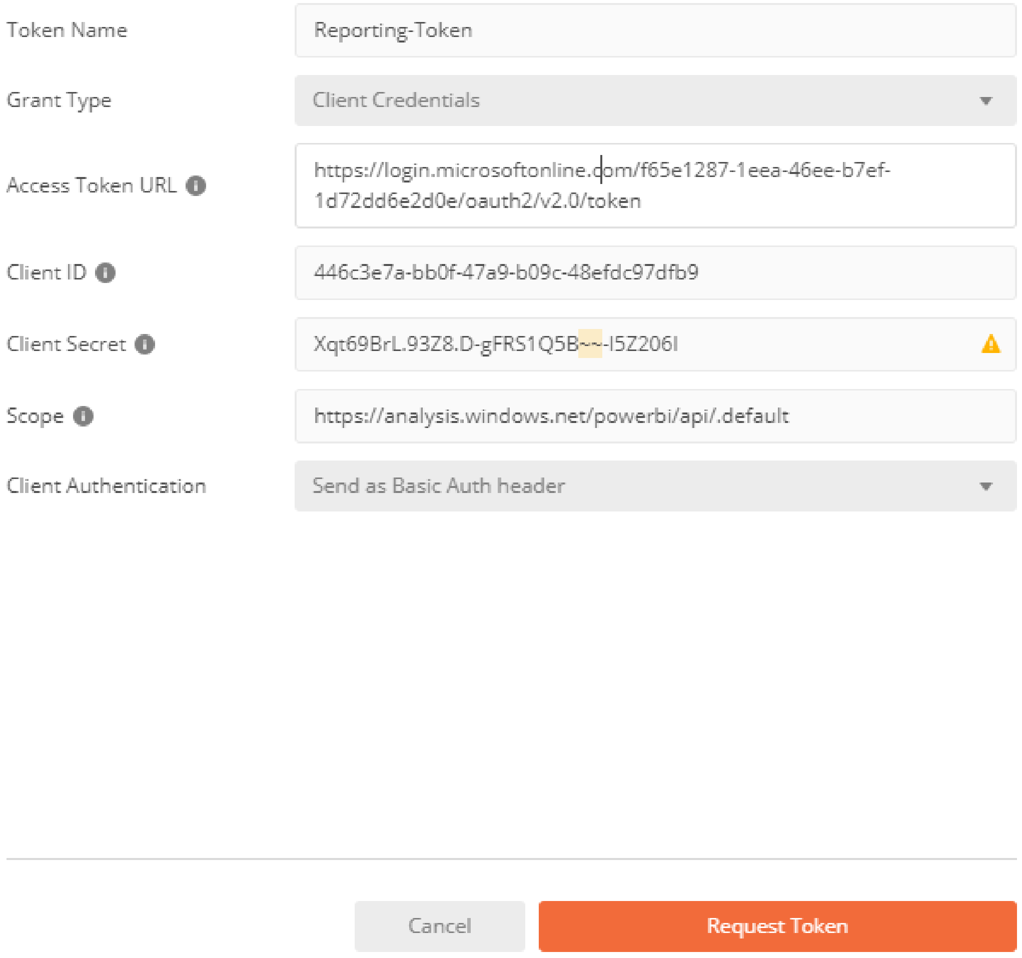
Body

```
{
  "@odata.context": "http://wabi-paas-1-scus-redirect.analysis.windows.net/v1.0/my
  "id": "f5be1bf9-e7de-46e6-8e2e-0818d107027d",
  "reportType": "PowerBIReport",
  "name": "Retail Analysis Sample",
  "webUrl": "https://app.powerbi.com/groups/me/reports/f5be1bf9-e7de-46e6-8e2e-081
  "embedUrl": "https://app.powerbi.com/reportEmbed?reportId=f5be1bf9-e7de-46e6-8e2
  "isOwnedByMe": true,
  "datasetId": "5ffa16ac-a591-46b0-8040-c21a17067c1a"
}
```

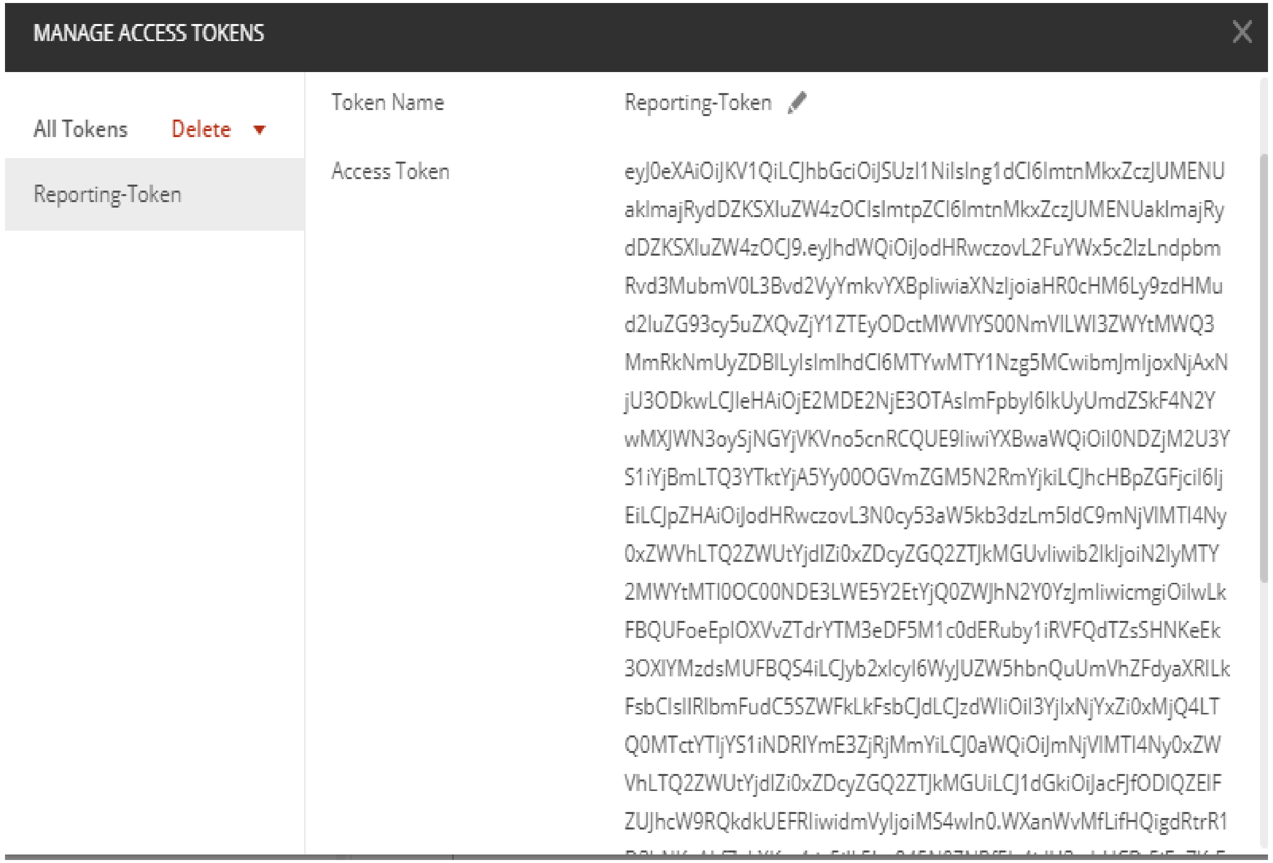
6.5 POC : One time-Setup for each report – embed url and data set id

Postman :

Generate auth token from MS Identity Platform



Token Name	Reporting-Token
Grant Type	Client Credentials ▾
Access Token URL ⓘ	https://login.microsoftonline.com/f65e1287-1eea-46ee-b7ef-1d72dd6e2d0e/oauth2/v2.0/token
Client ID ⓘ	446c3e7a-bb0f-47a9-b09c-48efdc97dfb9
Client Secret ⓘ	Xqt69BrL.93Z8.D-gFRS1Q5B~~-I5Z206I ⚠
Scope ⓘ	https://analysis.windows.net/powerbi/api/.default
Client Authentication	Send as Basic Auth header ▾



Token Name	Reporting-Token
Access Token	

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6ImtnMkxZczJUMENU
akImajRydDZKSXluZW4zOCIsImtpZCI6ImtnMkxZczJUMENUakImajRy
dDZKSXluZW4zOCJ9.eyJhdWQiOiJodHRwczovL2FuYWx5c2lzLndpbm
Rvd3MubmV0L3Bvd2VyYmkvYXBpIiwiaXNzIjoiaHR0cHM6Ly9zdHMu
d2luZG93cy5uZXQvZjY1ZTEyODctMWVlYS00NmVlLWI3ZWYtMWQ3
MmRkNmUyZDBlLyIsImlhdCI6MTYwMTY1Nzg5MCwibmJmIjoxNjAxN
jU3ODkwLCJleHAiOjE2MDE2NjE3OTAsImFpbyI6IkUyUmdZSkF4N2Y
wMXJWN3oySjNGYjVKVno5cnRCQUE9IiwiYXBwaWQiOiI0NDZjM2U3Y
S1iYjBmLTQ3YTktYjA5Yy00OGVmZGM5N2RmYjkiLCJhcHBpZGFjciI6Ij
EiLCJpZHAiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC9mNjVlMTI4Ny
0xZWVhLTQ2ZWUtYjdlZi0xZDcyZGQ2ZTJkMGUvIiwib2lkIjoiN2IyMTY
2MWYtMTI0OC00NDE3LWE5Y2EtYjQ0ZWJhN2Y0YzJmIiwicmgiOiIwLk
FBQUFoeEplOXVvZTdrYTM3eDF5M1c0dERuby1iRVFQdTZsSHNKeEk
3OXlYMzdsMUFBQS4iLCJyb2xlcyI6WyJUZW5hbnQuUmVhZFdyaXRlLk
FsbCIsIlRlbmFudC5SZWFkLkFsbCJdLCJzdWIiOiI3YjIxNjYxZi0xMjQ4LT
Q0MTctYTljYS1iNDRlYmE3ZjRjMmYiLCJ0aWQiOiJmNjVlMTI4Ny0xZW
VhLTQ2ZWUtYjdlZi0xZDcyZGQ2ZTJkMGUiLCJ1dGkiOiJacFJfODlQZEIF
ZUJhcW9RQkdkUEFRIiwidmVyIjoiMS4wIn0.WXanWvMfLifHQigdRtrR1
```

6.6 POC : One time-Setup for each report – embed url and data set id

Postman :

Generate datasetid for report in MS Power BI(can be done for any report)

GET API https://api.powerbi.com/v1.0/myorg/reports/{reportid} -> reportid shall be from Power BI url

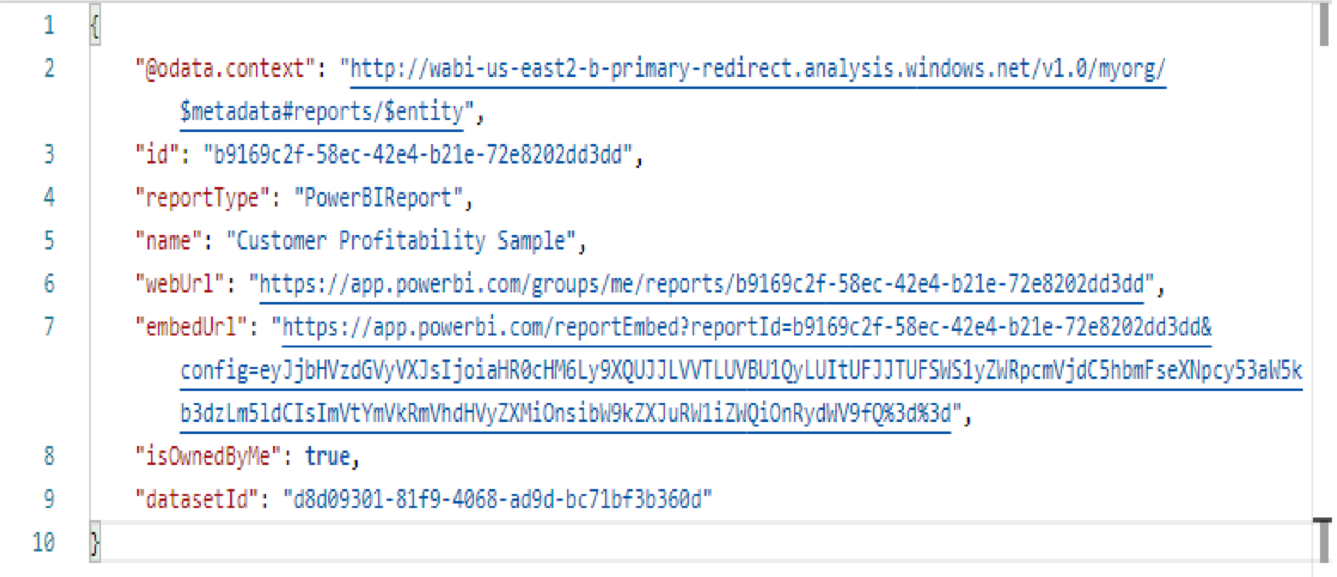
https://api.powerbi.com/v1.0/myorg/reports/71e06a2b-1310-4339-9abf-2db0d8f99346

Auth Token – Received from MS Identity Platform thru login api

Response

Embed URL,

Data set ID



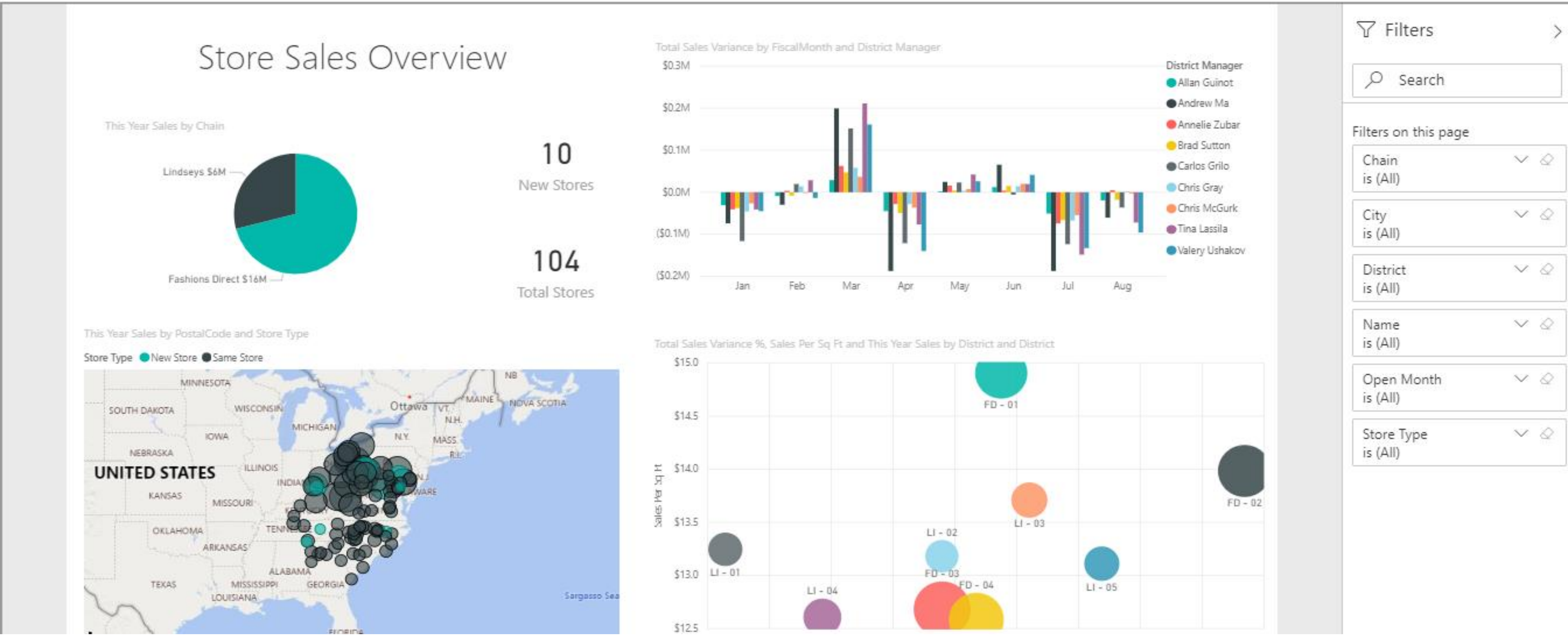
```
{
    "@odata.context": "http://wabi-us-east2-b-primary-redirect.analysis.windows.net/v1.0/myorg/
        $metadata#reports/$entity",
    "id": "b9169c2f-58ec-42e4-b21e-72e8202dd3dd",
    "reportType": "PowerBIReport",
    "name": "Customer Profitability Sample",
    "webUrl": "https://app.powerbi.com/groups/me/reports/b9169c2f-58ec-42e4-b21e-72e8202dd3dd",
    "embedUrl": "https://app.powerbi.com/reportEmbed?reportId=b9169c2f-58ec-42e4-b21e-72e8202dd3dd&
        config=eyJjbHVzdGVyVXJsIjoiaHR0cHM6Ly9XQUJJLVVTLUVBU1QyLUItUFJJTUFSWS1yZWRpcmVjdC5hbmFseXNpcy53aW5k
        b3dzLm5ldCIsImVtYmVkRmVhdHVyZXMiOnsibW9kZXJuRW1iZWQiOnRydWV9fQ%3d%3d",
    "isOwnedByMe": true,
    "datasetId": "d8d09301-81f9-4068-ad9d-bc71bf3b360d"
}
```

Configuration in Couchbase

```
[
        {
                "report-1 :
                {
                        "embedUrl": "<embedUrl>",
                        "datasetId": "<datasetId>“,
                         "reportId": "<reportId>"
                }
        }
]
```

7.1 ReactJS and MS Power BI Integration for Scheduled Reporting

Report



Filters

Search

Filters on this page

Chain
is (All)

City
is (All)

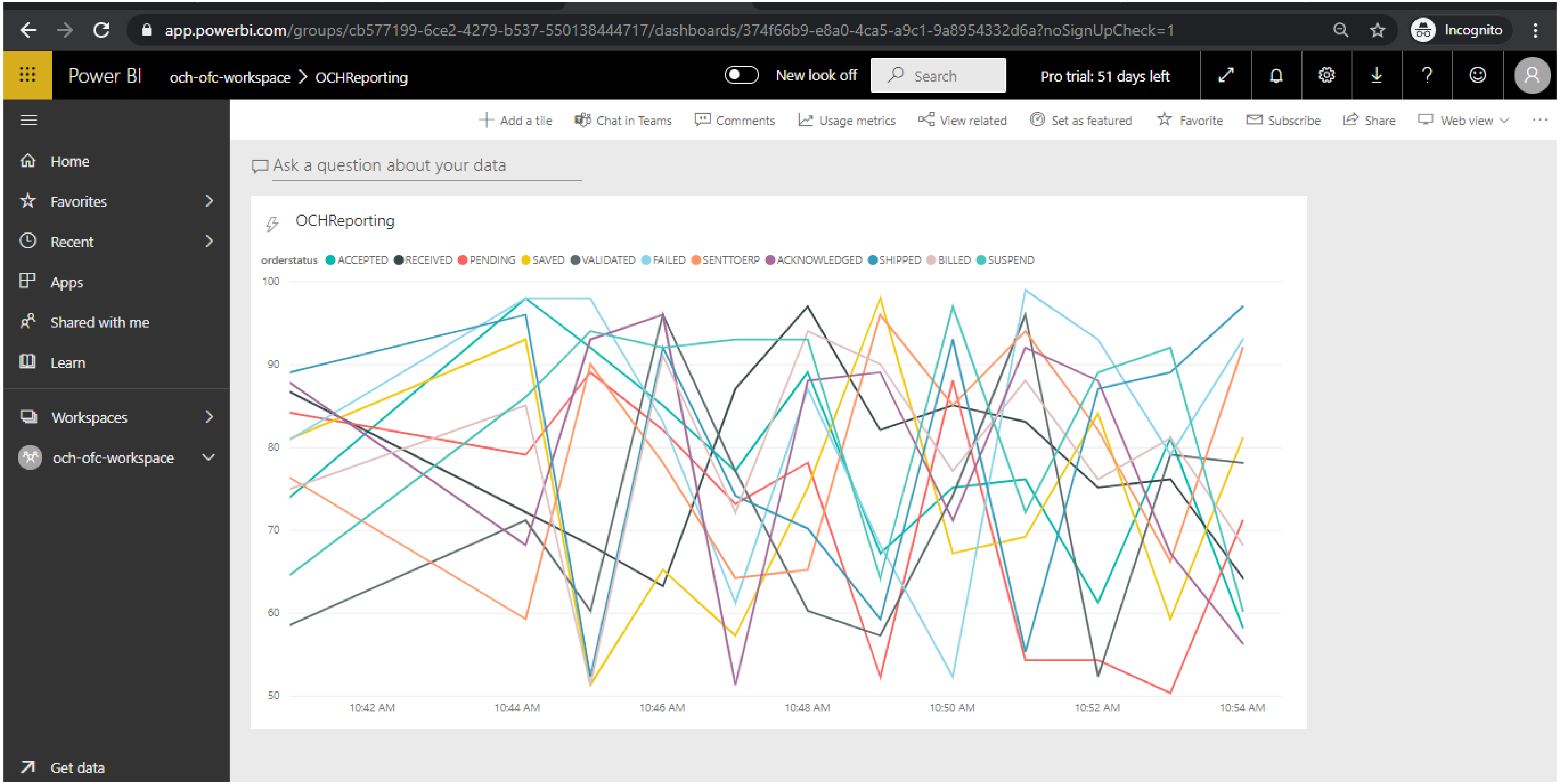
District
is (All)

Name
is (All)

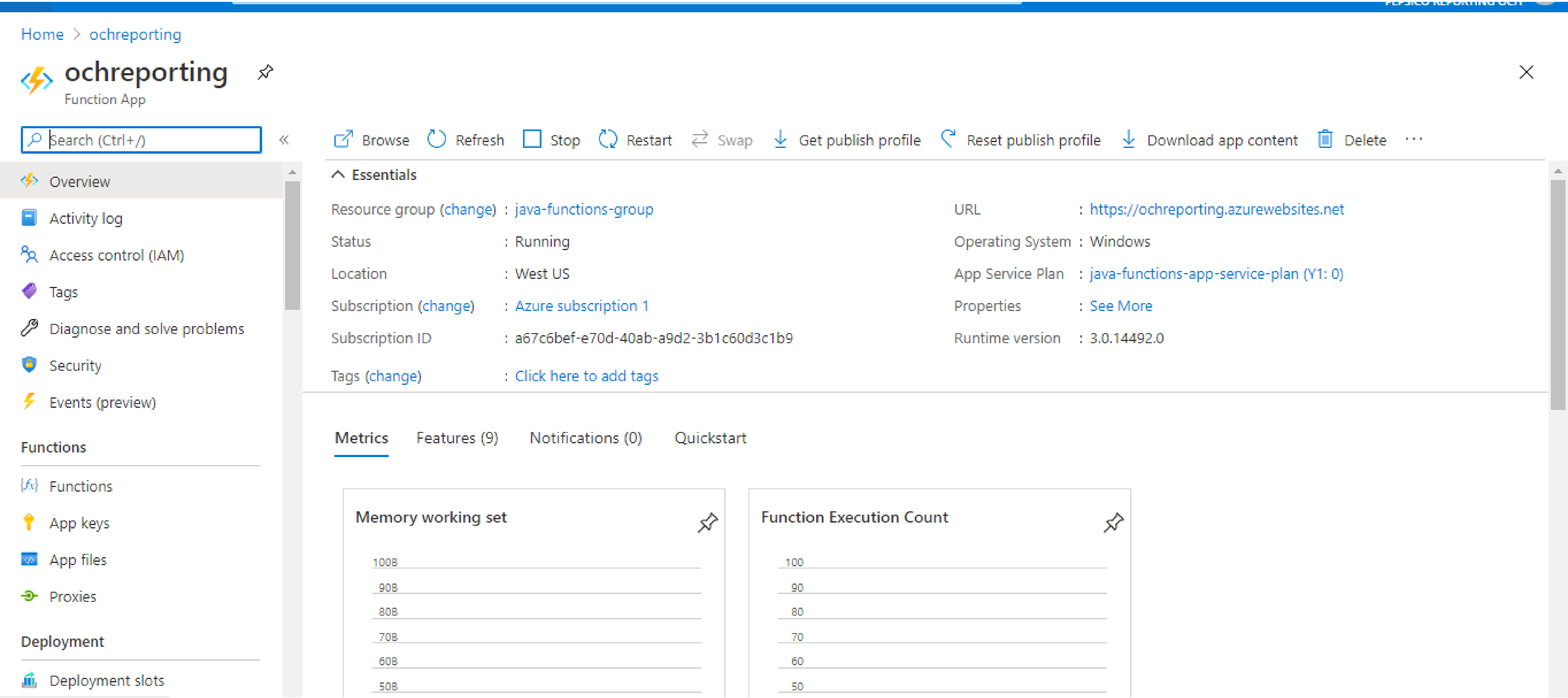
Open Month
is (All)

Store Type
is (All)

7.2 ReactJS and MS Power BI Integration for Near Real Time stream Reporting



7.3 Azure App Functions – Demo(can replace reporting mS)



Resource group (change) : java-functions-group

Status : Running

Location : West US

Subscription (change) : Azure subscription 1

Subscription ID : a67c6bef-e70d-40ab-a9d2-3b1c60d3c1b9

Tags (change) : Click here to add tags

URL : https://ochreporting.azurewebsites.net

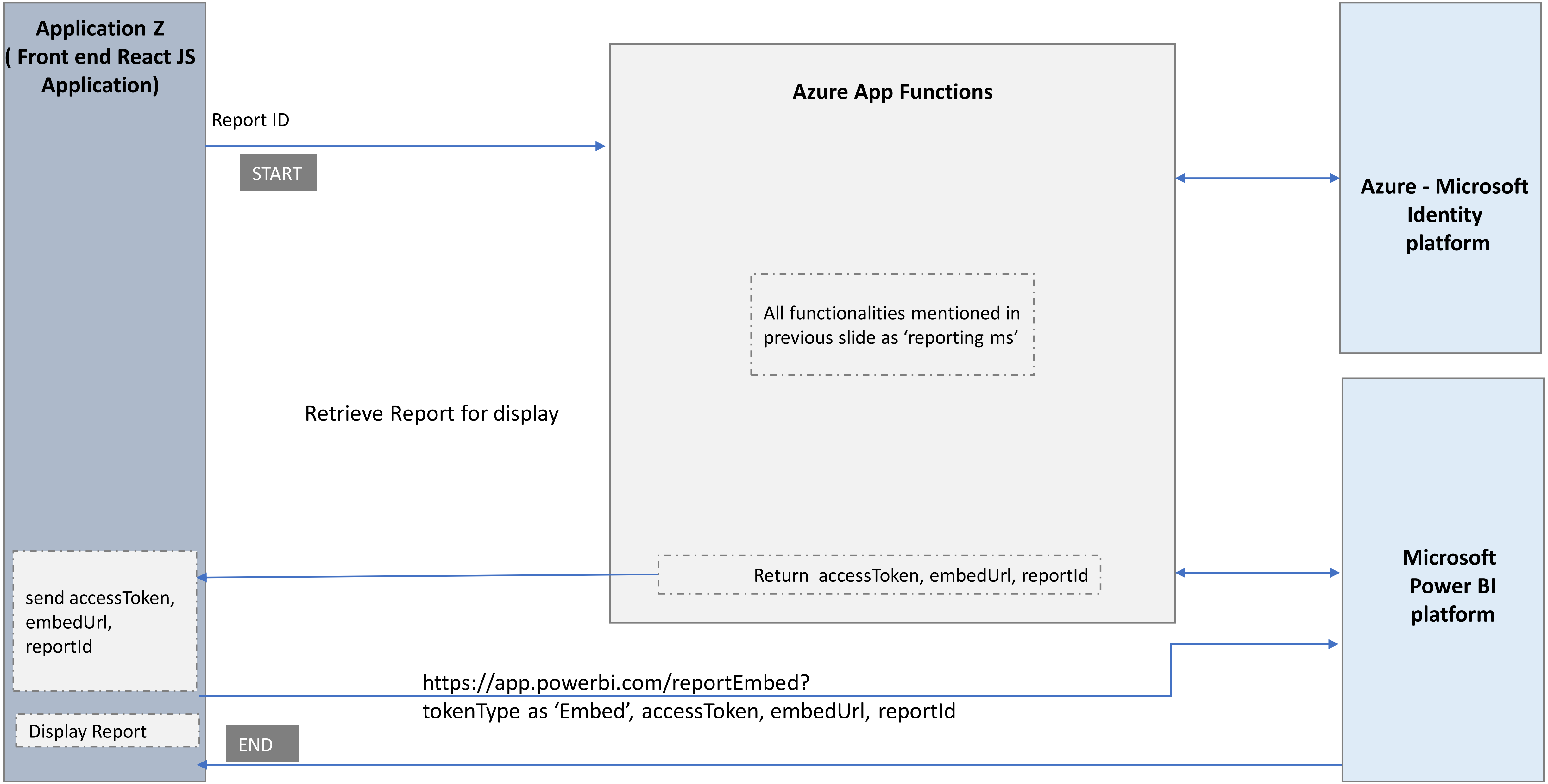
Operating System : Windows

App Service Plan : java-functions-app-service-plan (Y1: 0)

Properties : See More

Runtime version : 3.0.14492.0

7.4 Azure App Functions – Demo(can replace reporting mS)



Application Z (Front end React JS Application)

Report ID

START

Azure App Functions

Azure - Microsoft Identity platform

All functionalities mentioned in previous slide as ‘reporting ms’

Retrieve Report for display

Return accessToken, embedUrl, reportId

send accessToken, embedUrl, reportId

Microsoft Power BI platform

https://app.powerbi.com/reportEmbed?
tokenType as ‘Embed’, accessToken, embedUrl, reportId

Display Report

END

8. Pre-requisites for Reporting

1. Power BI Pro license for subscription of reports
2. ‘reportinguser’ required in existing tenant for getting token from MS Identity Platform
3. Existing tenant admin to grant Power BI api permissions and grant admin consent to ‘reportinguser’
4. MS Azure subscription to create Azure Stream Analytics job & Azure App Functions