Monitoring Fabric

Make do, Buy or Build

Thank you, partners

element 61
experience & expertise

Cloubis

Microsoft

MICROPOLE BELUX

PLAINSIGHT

Adapt and Enable

Tabular Editor

delaware

datashift

u2u

cegeka

EpicData.

LACO

tillit data shapers

lytix

bmatix
Act informed

randstad digital

KOHERA

MONIN
Database Managed Services

inetum realdolmen
Positive digital flow

TriFinance
BEYOND ADVISORY

ÁSGEIR GUNNARSSON

- Data Platform MVP
- Partner at Data Lab ApS
- In BI since 2007
- Co-organizer of Data Platform Next Step, Power BI Next Step and Power BI Cruise
- Co-author of Pro Microsoft Power BI Administration

- E-mail: asgeir@thesmalldatalab.com
- Twitter: @bidgeir
- LinkedIn: linkedin.com/in/asgeirgun

Agenda

1. Why monitor?
2. Out of the box solutions
3. Buy ready-made solutions
4. Build your own solution

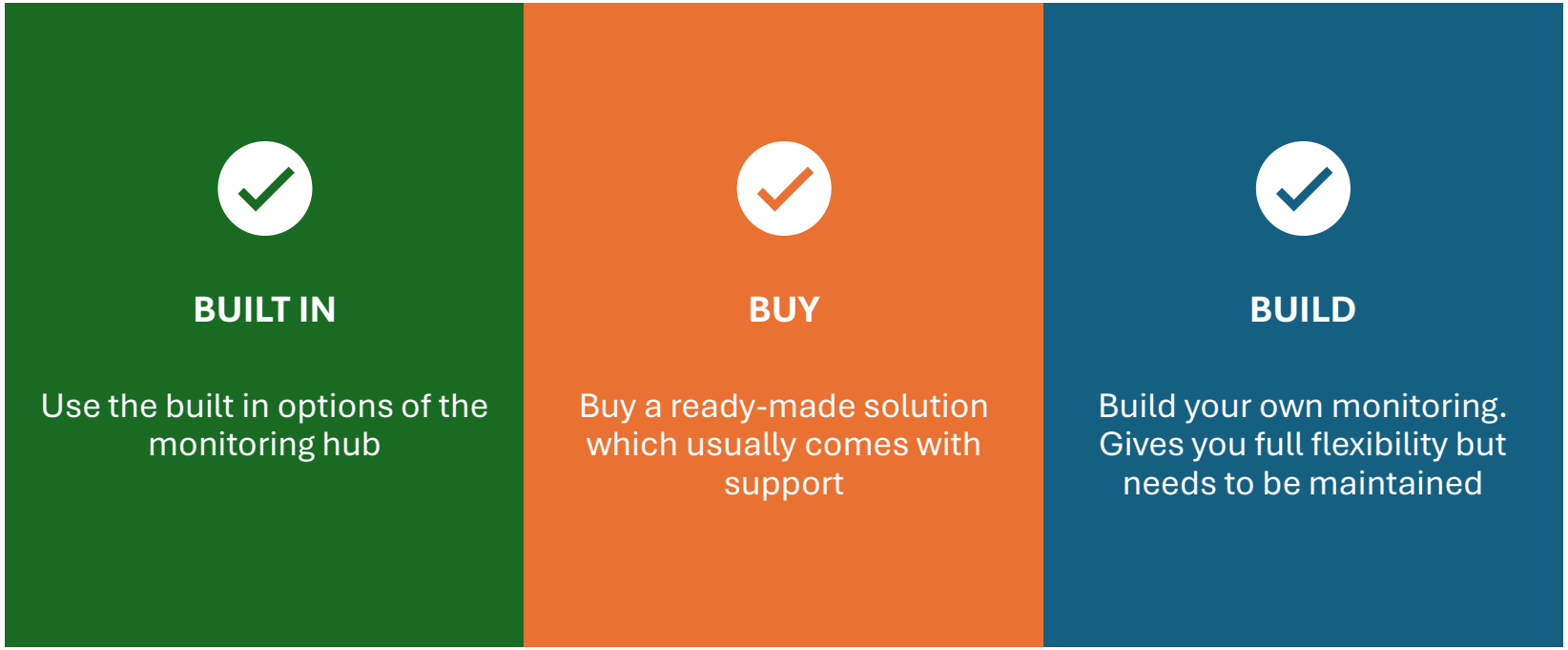
Why monitor?

Overview of Monitoring in Fabric

- **Operational efficiency:**
  - Track performance of reports, queries, compute usage, and pipelines.
  - Monitor system health and ensure uptime.
  - Adoption of Fabric
- **Security and Compliance:**
  - Audit access, roles, and permissions.
  - Monitor security events for regulatory compliance (GDPR, SOC).
- **Data Quality:**
  - Validate data throughout transformation processes.
  - Detect and address inconsistencies proactively.
- **Cost:**
  - Monitor workload usage and optimize resources.
  - Set alerts for cost overruns and resource efficiency.

How can you monitor

There is no one way to do things. It depends a lot on the needs and budget what is right for each organization



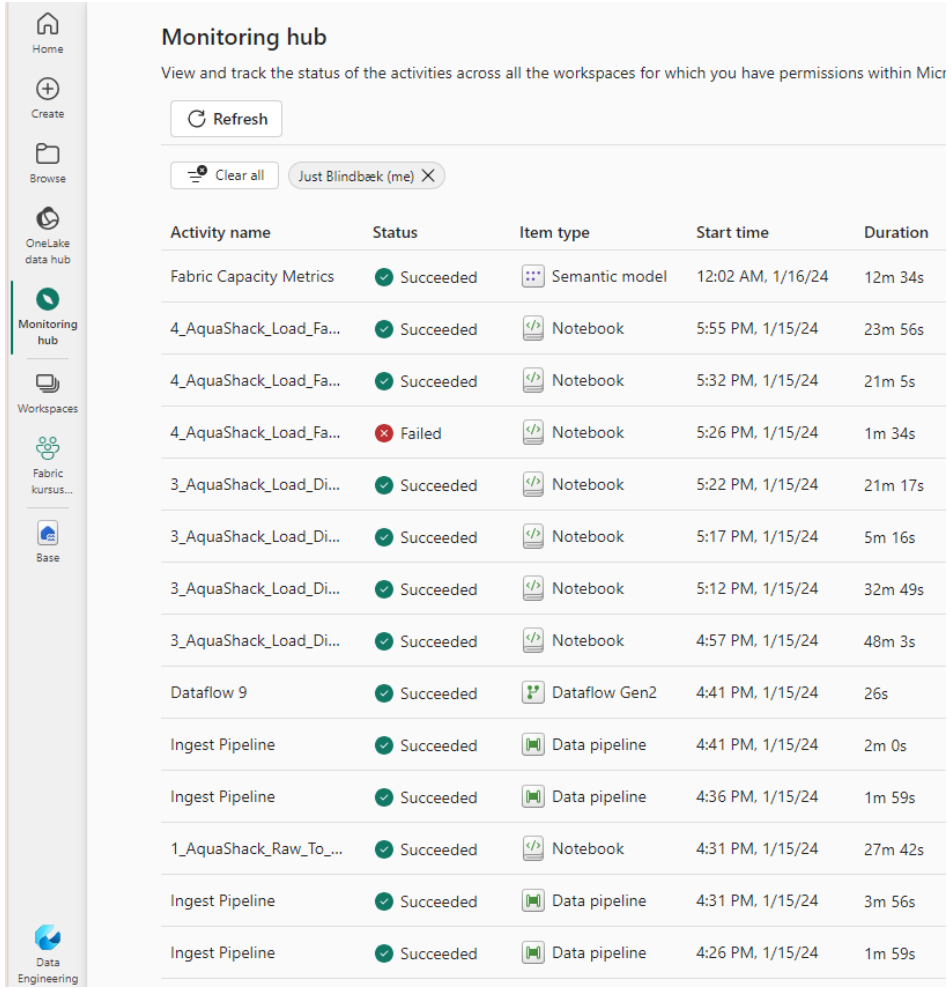
Out of the box solutions

1. Monitoring (hub)
2. Admin Monitoring workspace
3. Capacity Metrics
4. Azure Log Analytics in Power BI
5. Platform Monitoring (private preview)

Monitoring (hub)

Monitoring (hub)

- Monitor various Microsoft Fabric activities, such as semantic model refresh and Spark Job runs
- Is available for workloads from Power BI, Data Factory, Data Engineering and Data Science
- Filters are applied by default to limit the number of items initially displayed
- Select different filters by using the filter drop-down in the upper right corner
- Select an item from the list and get detailed information about that item
- Hover over an item's name and get available quick actions for the item, such as stop, start, re-run, or other quick actions

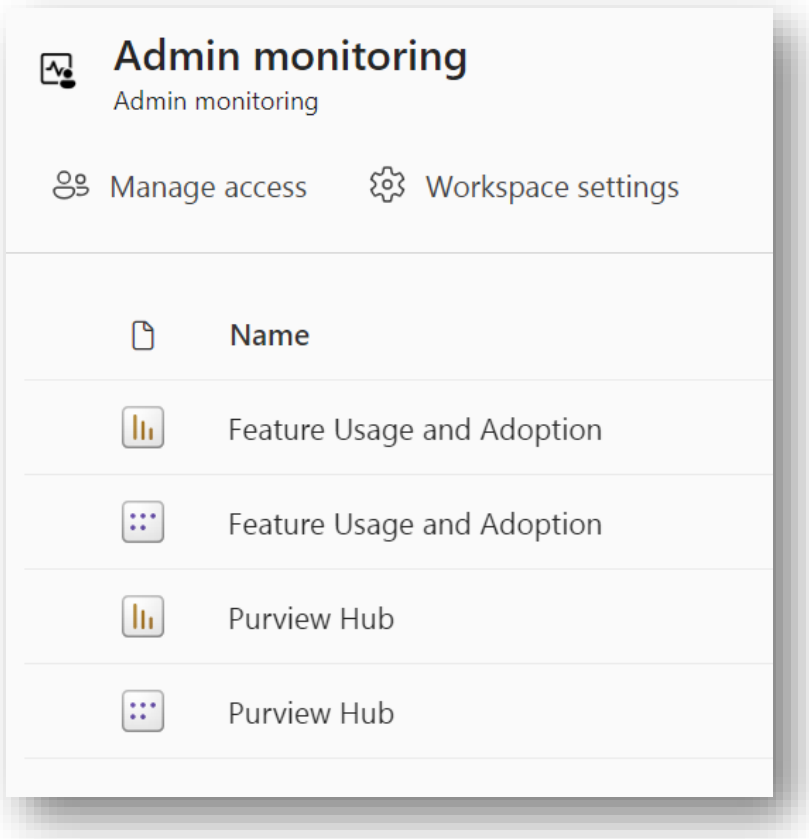


https://learn.microsoft.com/en-us/fabric/admin/monitoring-hub

Admin Monitoring workspace

Admin monitoring workspace

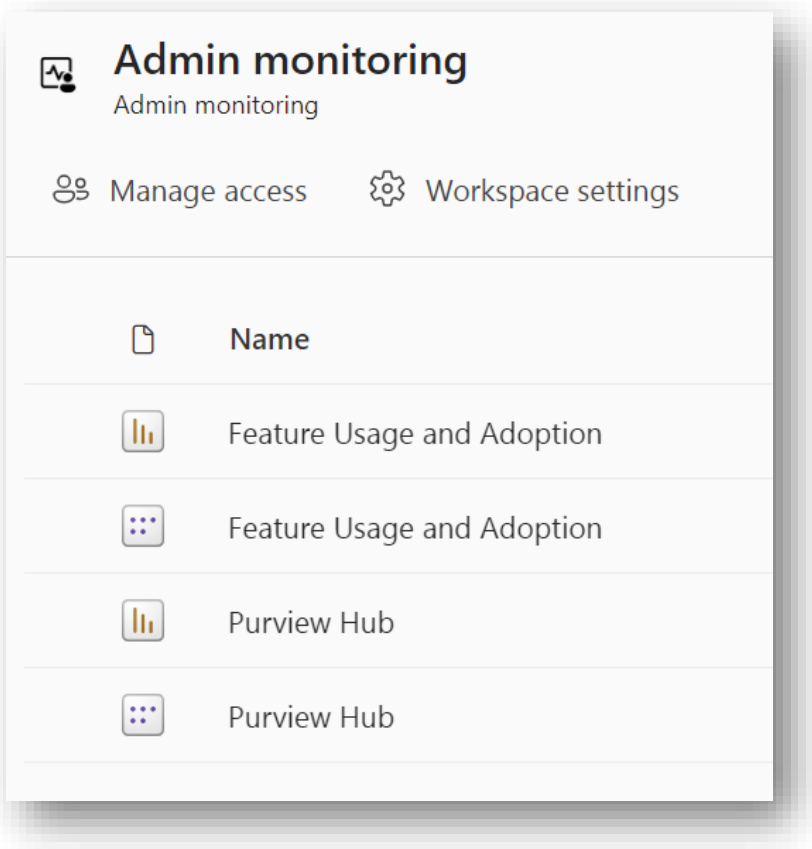
- One-stop shop for enterprise reporting and analytics
- Data queried from multiple sources, transformed, and landed in a single repository specific to each tenant
- Includes out of the box reporting focused on Fabric tenant management scenarios
- Also includes semantic models for customization
  - Out of the box reporting and curated semantic models made readily available via the workspace
  - Managed, automated data refresh of all semantic models
- Available to all tenants regardless of licensing type or total # of users



https://learn.microsoft.com/en-us/fabric/admin/monitoring-workspace

Who is the workspace intended for?

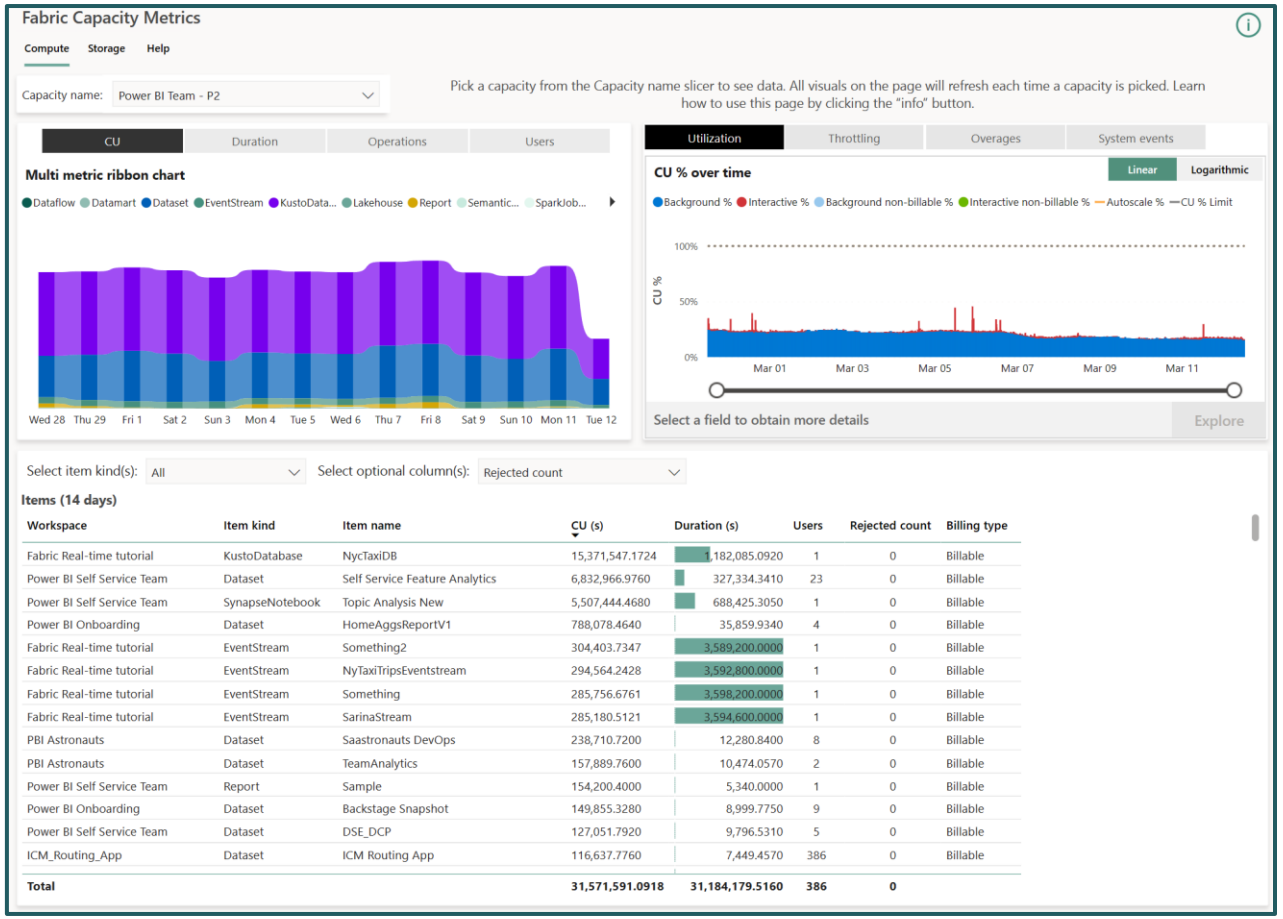
- Global admins and Fabric administrators
- Capacity admins and domain admins
- “Power users” such as multi-workspace admins, COE leads



Capacity Metrics App

Fabric Capacity Metrics

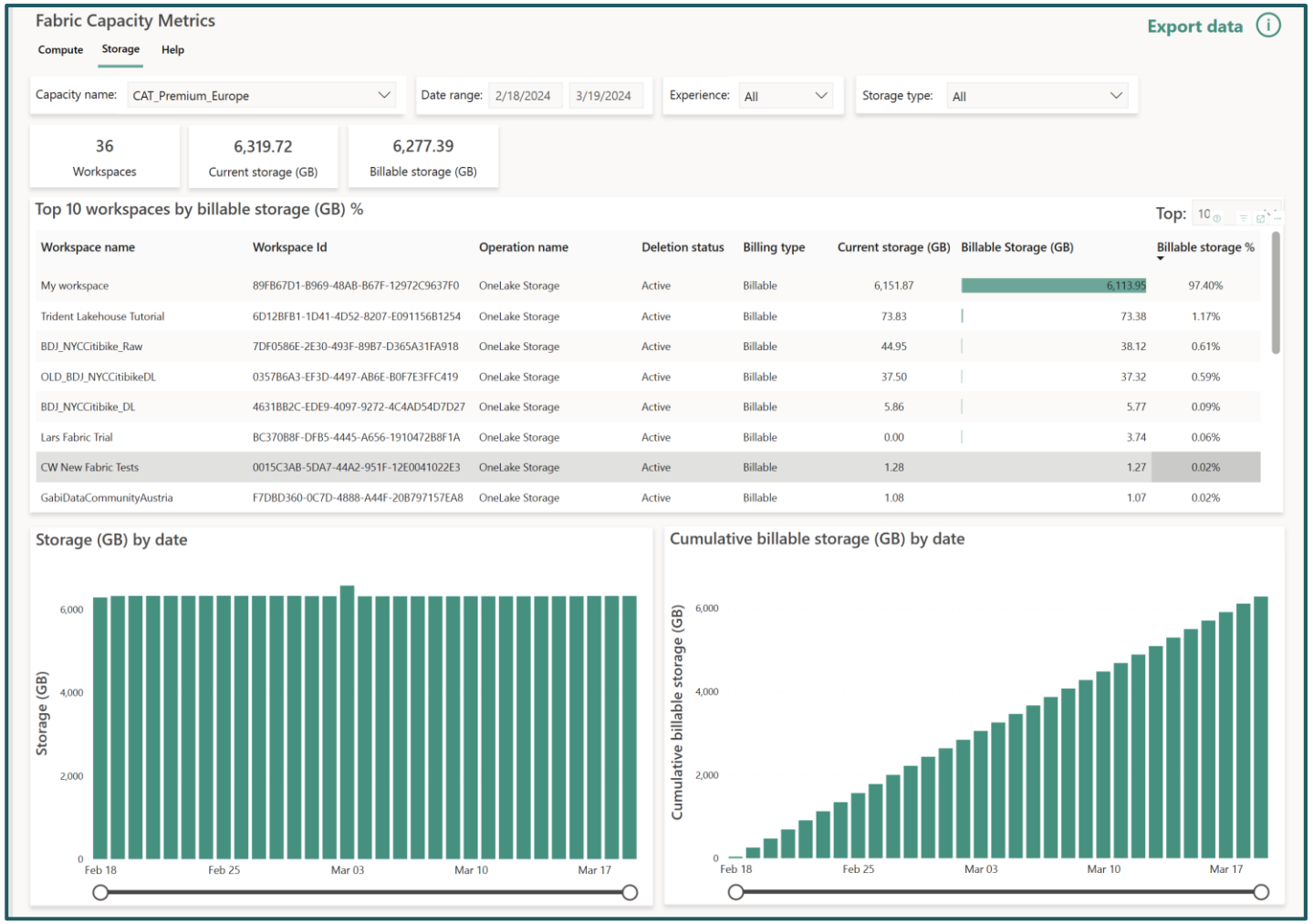
- Gain tenant-wide visibility into capacity usage across all Fabric workloads.
- Identify resource usage trends to optimize autoscale and mitigate throttling impacts.
- Compare preview and production workloads for data-driven capacity sizing decisions.
- Zoom in on workload operations and artifacts with detailed granularity down to 30 seconds.
- Monitor real-time operations and the impact of long-running jobs on capacity limits.
- Analyze user experience to plan efficient scale-ups and optimize resource allocation.



https://learn.microsoft.com/en-us/fabric/enterprise/metrics-app

Fabric Capacity Metrics

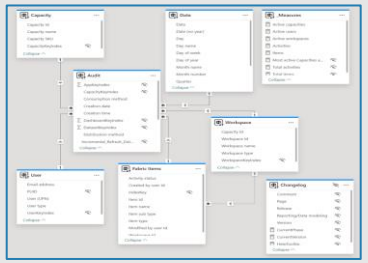
- Monitor OneLake consumption
- Measure the trends of workspace storage consumption against capacity limits, by day or hour
- Reconcile costs with internal chargeback processes



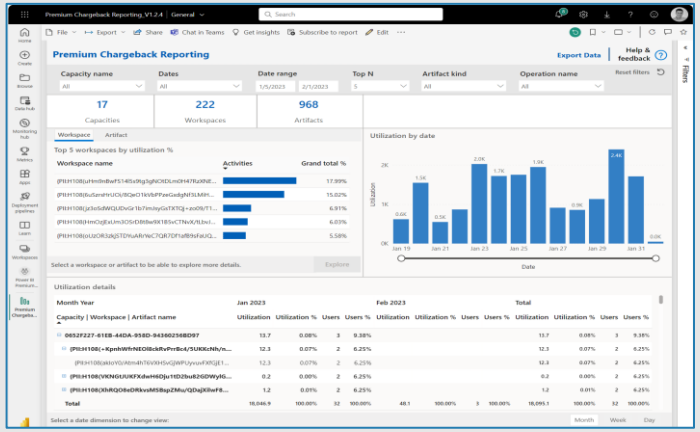
Capacity Metrics Roadmap

Customers can automate and extend their governance workflows usage data
(Codename Telemetry Store)

- Direct access to aggregated Capacity usage data for **extensibility** and **retention** showcasing **best of fabric** for exploratory and data lake consumption

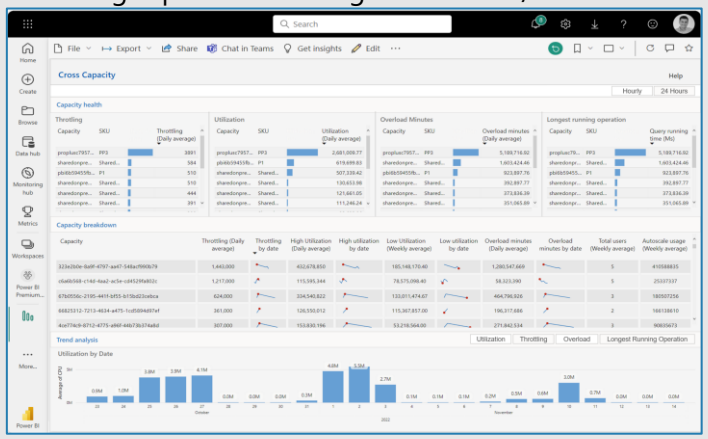


Chargeback GA
Customers can use turnkey analytics to spread Fabric bill based on resource consumption **& domain info**

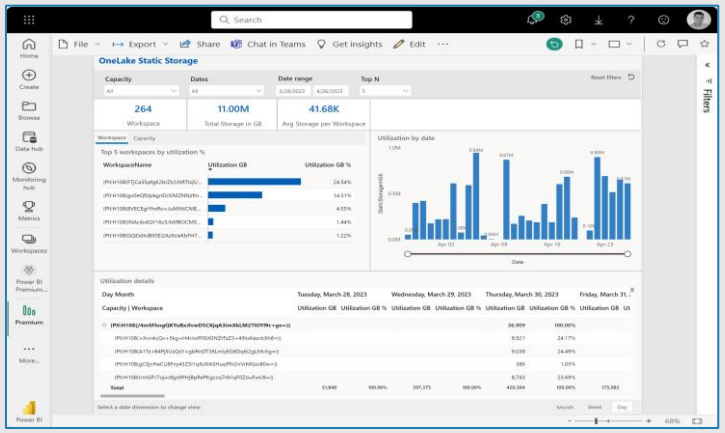


Lineage / Historical artifact analysis
Customers can identify regressions, success rates and scheduling abuse

Tenant Capacity Health (Dt)
Customers can quickly identify health issues from throttling. Optimized for large customers / ISV's



OneLake / Storage
Customers can monitor static storage, distribute costs by capacity / workspace **& domain**



Platform updates

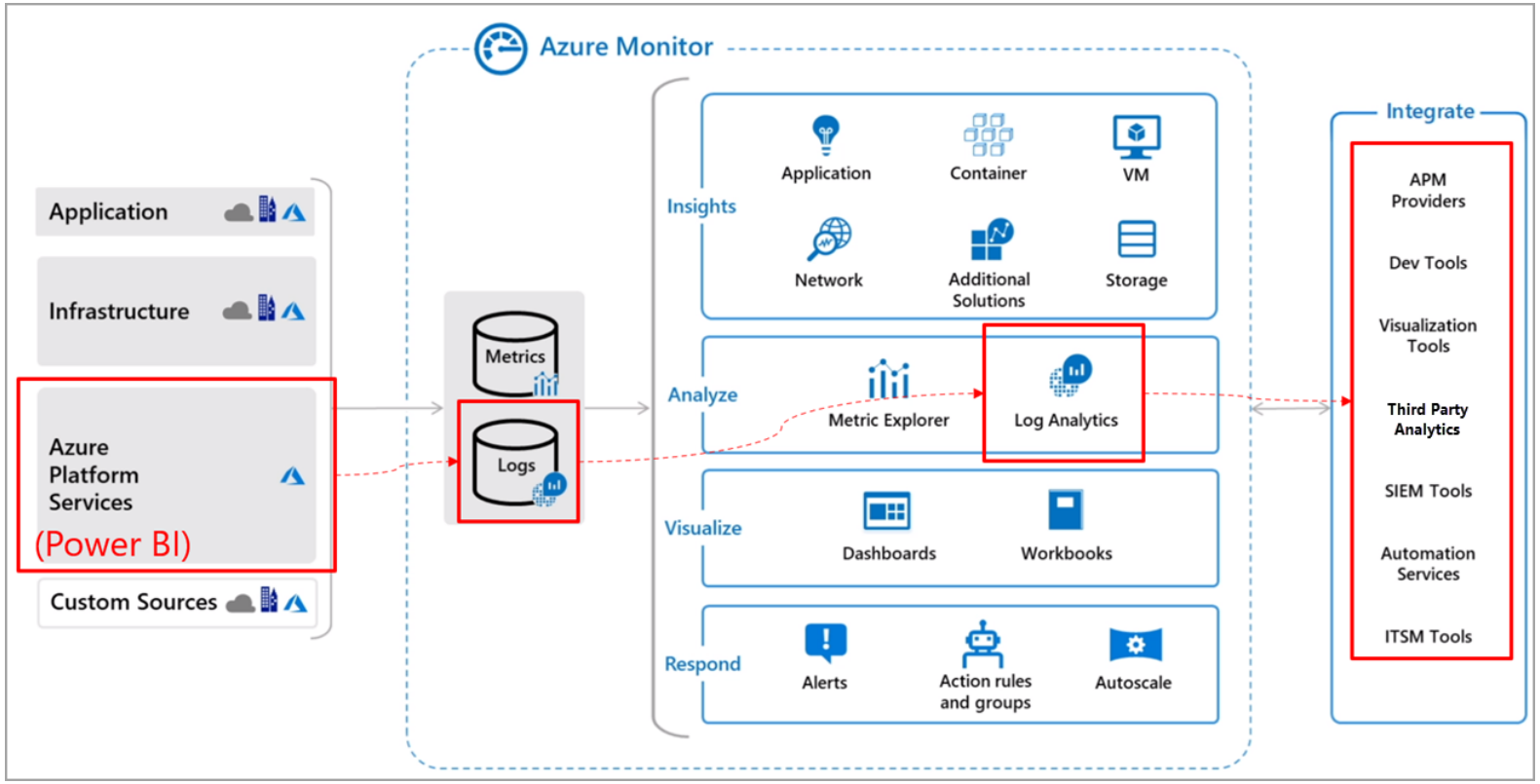
- ✓ *Surge protection to provide richer controls for priority scenarios and policies to limit unexpected consumption*
- ✓ *Autoscale policy refinement & support for feature level autoscale*
- ✓ *More new Fabric experiences onboarded!* 😉

Azure Log Analytics in Power BI

Save activity logs

Azure Log Analytics in Power BI

- Semantic model activity logs for premium workspaces
- Detailed analysis Services engine trace events
  - Queries down on user and visual level
  - Processing / refresh events
- Typical use cases:
  - Performance investigations
  - Scale/load testing
  - Pre-release validation
- Detailed and can be high volume and large
- Stored in Log Analytics (Kusto database)



https://learn.microsoft.com/en-us/power-bi/transform-model/log-analytics/desktop-log-analytics-overview

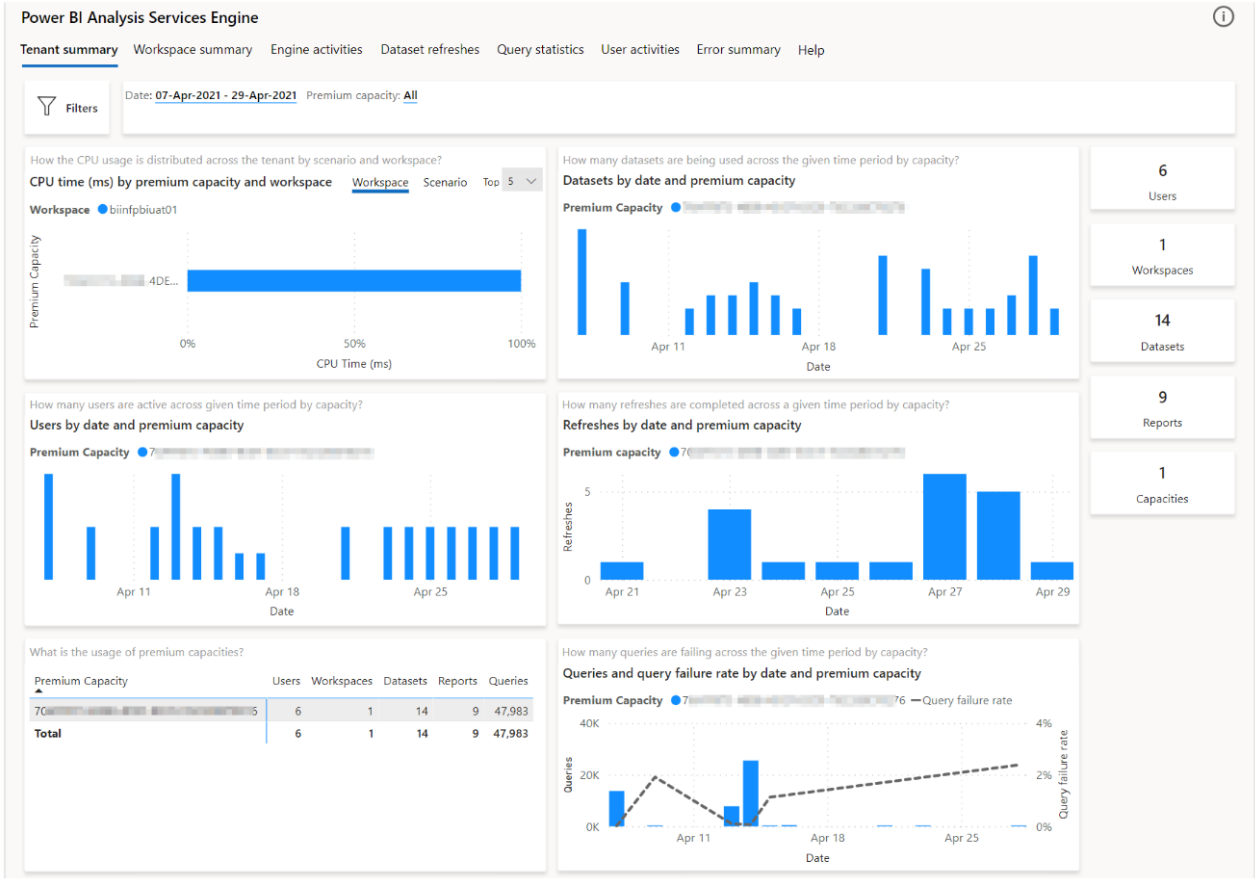
Analyze logs with Log Analytics

- Log Analytics is a dedicated portal used to work with log queries and their results.
- Log queries help you to fully leverage the value of the data collected in Azure Monitor Logs
- Azure Monitor uses a version of the Kusto query language (KQL)
- Create alert rules to get notified

```kusto
Perf
| where CounterName == "% Processor Time" and ObjectName == "Processor" and InstanceName == "_Total"
| summarize AggregatedValue = avg(CounterValue) by bin(TimeGenerated, 1hr), Computer
| render timechart
```

Prebuilt Power BI Templates

- Reports that you can point to your Azure Log Analytics workspaces to load data and get insights
- **Analysis Services Engine**: Ocean of trace events that enables to go from data to insight to action quickly
- **Usage and performance logs**: Identify load patterns, investigate user actions, look at query performance trends, visualize refreshes, and much more



https://github.com/microsoft/PowerBI-LogAnalytics-Template-Reports

Fabric “Platform” Monitoring

Private Preview sneak view

Fabric “Platform” Monitoring (private preview)

- **Built-in solution** for root-cause analysis, historical log analysis, and anomaly detection.
- Simplifies access to diagnostic logs and metrics.
- Leverages Eventhouse (KQL database) for efficient, self-serve troubleshooting and diagnostics.
- Provides seamless, end-to-end visibility across Fabric native data apps.
- Correlates events from origination to subsequent operations and queries across services.

- **Workspace-level enablement:**
  - Send diagnostics to an Eventhouse for analysis.
  - Access pre-built dashboards and Querysets for quick insights.
  - Raw data available for custom dashboards, insights, and alerts.

https://learn.microsoft.com/en-us/fabric/release-plan/admin-governance#fabric-monitoring

Scope of Platform Monitoring (private preview)

- Power BI
  - Semantic Models Queries
  - Semantic Models Refreshes
- Real-time analytics
  - Eventhouse Queries
  - Eventhouse Metrics
- Planned:
  - Spark
  - Pipelines
  - SQL
  - Capacity data

```
// log count per day for last 30d
SemanticModelLogs
| where Timestamp > ago(30d)
| summarize count() by format_datetime(Timestamp, 'yyyy-MM-dd')

// average query duration by day for last 30d
SemanticModelLogs
| where Timestamp > ago(30d)
| where OperationName == 'QueryEnd'
| summarize avg(DurationMs) by format_datetime(Timestamp, 'yyyy-MM-dd')
```

```
//Succeeded vs Failed Queries over time in past 30 days Linechart
let Duration =timespan(30);
EventhouseQueryLogs
| where Timestamp  > ago (Duration)
| where Status in("Failed" , "Throttled")
|summarize Count=count() by Status,  bin(Timestamp, 1d)
|render linechart
```

Buy ready-made solutions

Buy a monitoring tool

- There are several vendors out there selling solutions
- Some solutions are actual software, some are solutions companies build for their customers
- Many of these solution will deliver more than just monitoring
- Most come with support
  - Check that well
- You don´t need to maintain any code
- You don´t have full flexibility

KEY CONSIDERATIONS

- Figure out if it´s a software solution or a custom-built solution
- Make sure there is support included or if not, how much it costs
- Make sure it covers all your monitoring needs
- Where is the data stored
- What access do you get to the data

MONITORING SOLUTIONS

- Power BI Sentinel (https://powerbisentinel.com)
  - Governance, Auditing and Disaster Recovery for Power BI and Fabric
- Power BI Control room from BI Samurai (https://bisamurai.com/power-bi-control-room/)
  - Power BI Monitoring, Audit and Governance Tool
- Argus BPI tenant monitoring (https://www.arguspbi.com/)
  - User activity & permissions, Data lineage and Refresh monitoring

Build your own solution

1. Tenant monitoring
2. Capacity monitoring
3. Job monitoring
4. Semantic Model monitoring

Tenant monitoring

Build your own solution

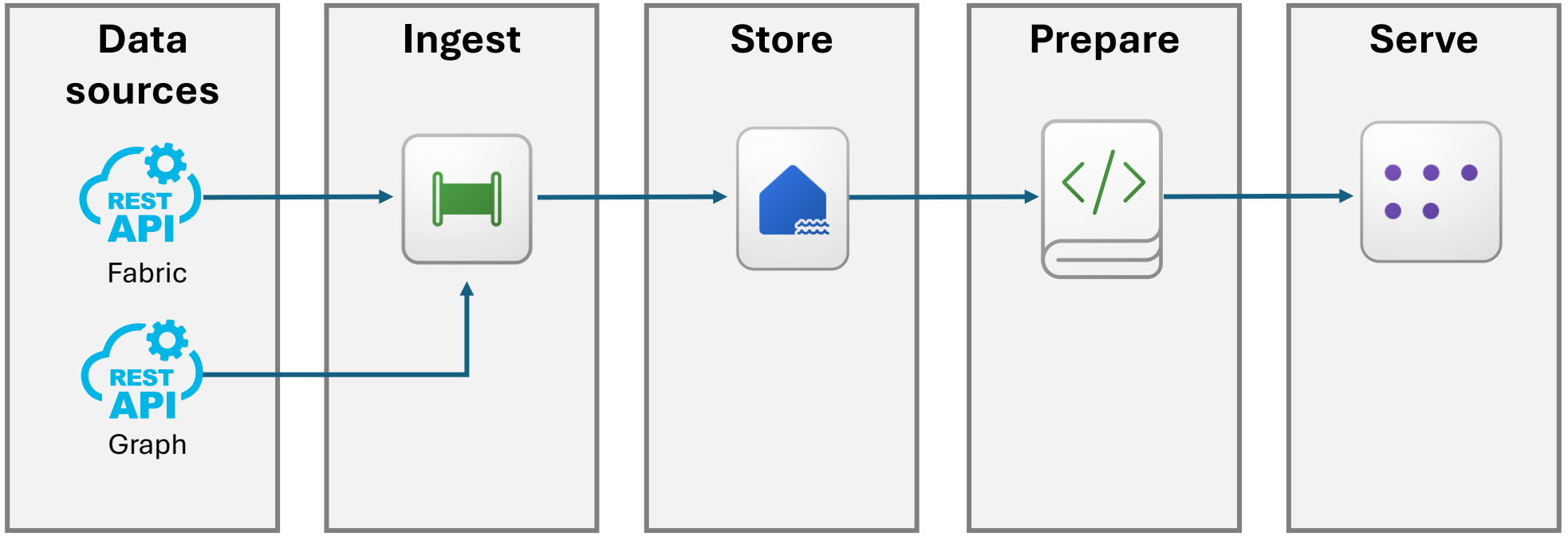
A complete solution to ingest and prepare

- Fabric activity/audit events
- Fabric artifacts metadata – the tenant catalogue
- Fabric tenant settings

- Extra: Microsoft Graph data (users, groups, licenses)

https://github.com/justBlindbaek/PowerBIMonitor

Dataflow architecture



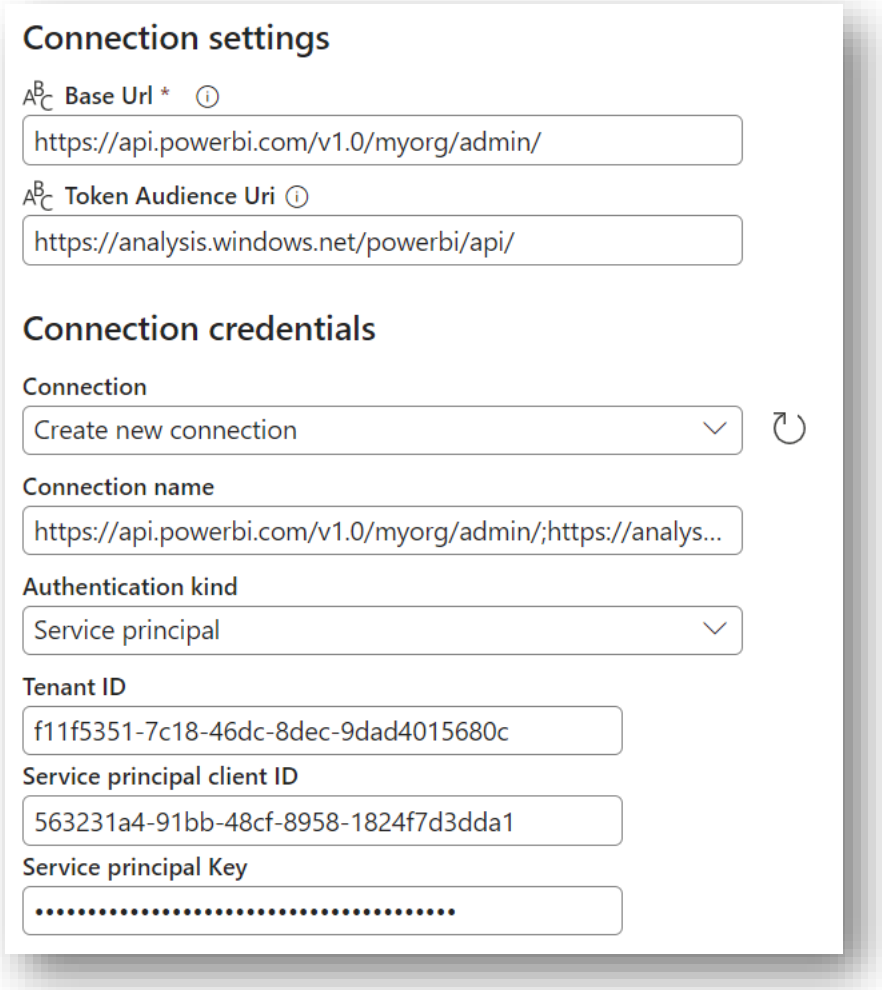
Access to the data: pre-req setup

- Create a Service Principal (App Registration)
- Create a new Security Group in Azure Active Directory
- Add the Service Principal as a member of the security group
- Enable service principal authentication for read-only admin APIs in Admin Tenant Settings and add the security group
- Optional: Enable the Enhance admin APIs responses with “detailed metadata” and ”DAX and mashup expressions”

https://justb.dk/blog/2021/02/extracting-the-power-bi-activity-log-with-data-factory/

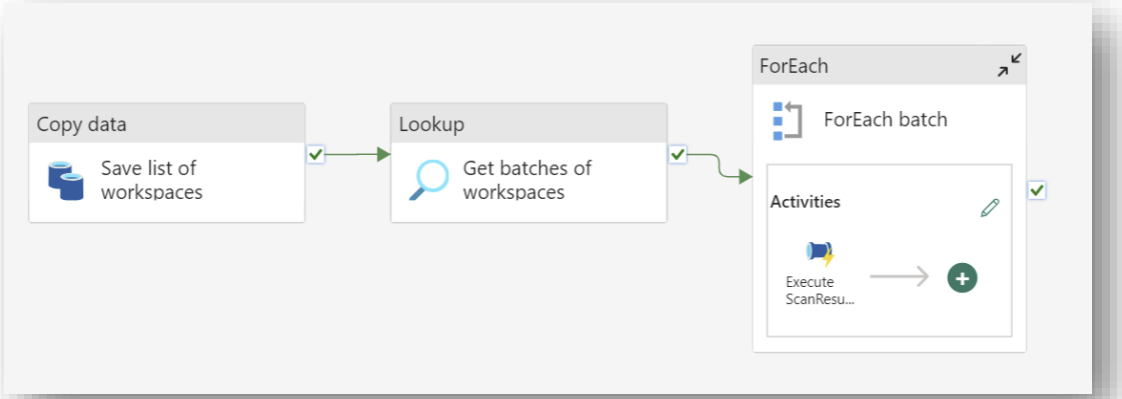
Connection settings

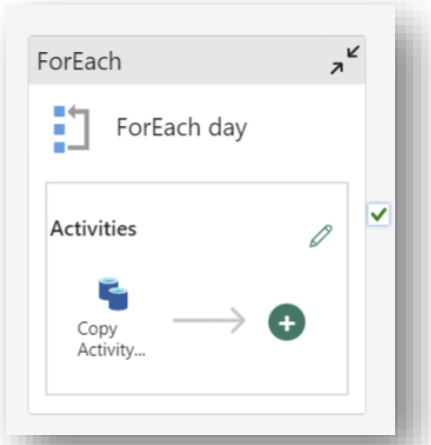
- Type: REST (Web V2)
- Base URL: https://api.powerbi.com/v1.0/myorg/admin/
- Token Audience Uri: https://analysis.windows.net/powerbi/api
- Authentication kind: Service Principal

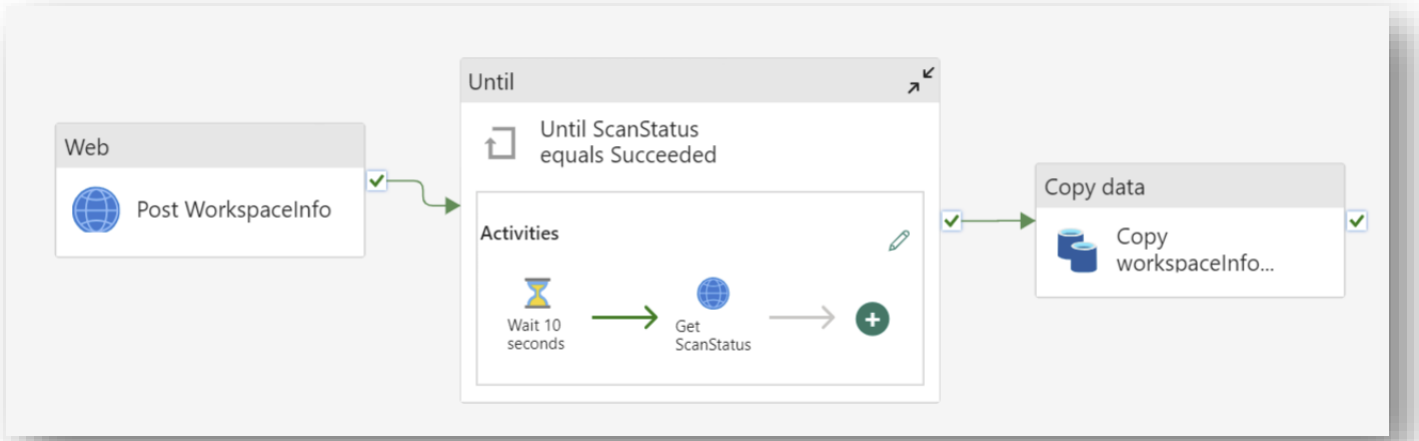


Ingest with Data Factory Pipelines

Name	Type
PL_FabricGetTenantSettings	Data pipeline
PL_PowerBIGetActivityEvents	Data pipeline
PL_PowerBIWorkspaceInfo	Data pipeline
PL_PowerBIWorkspaceInfoScanResult	Data pipeline







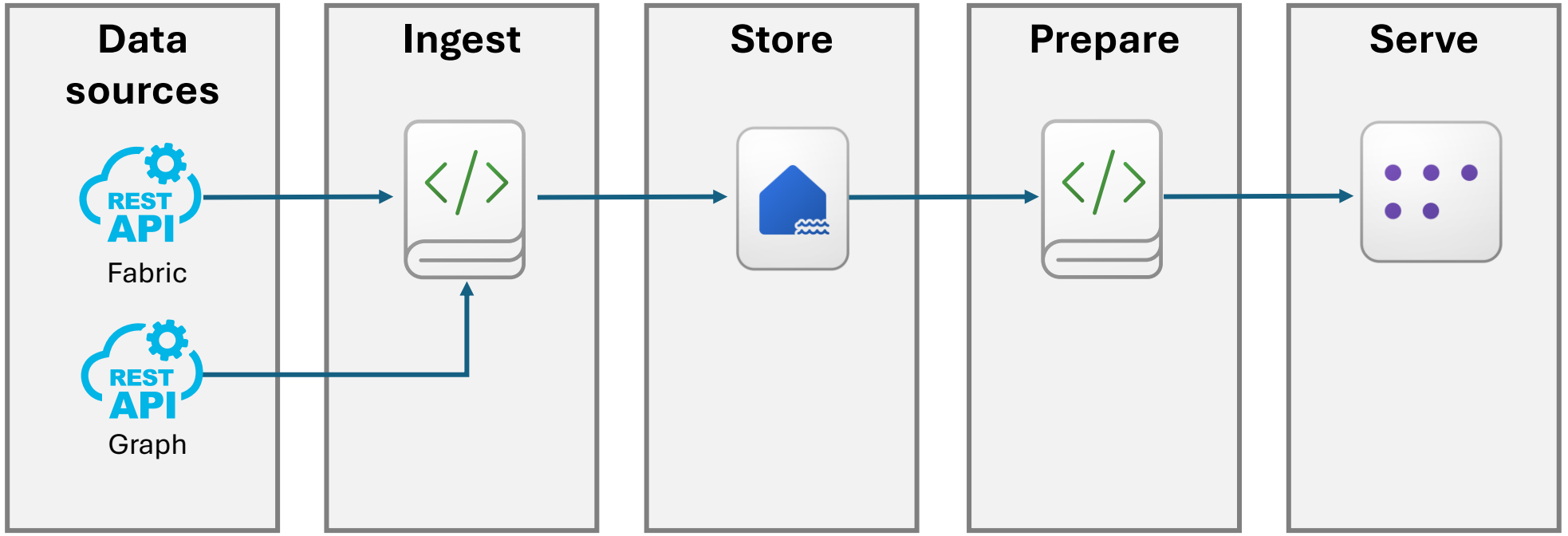
Build using spark

- Instead of using pipelines you can use Spark
- Spark has built in ways to talk to REST APIs
- Spark handles JSON responses much more elegantly than Data Factory
- It´s code first solution
  - More complicated to setup
  - More to maintain
  - Most likely more difficult to debug

KEY CONSIDERATIONS

- Code first
- Better at handling nested JSON structures
- Built in packages to talk to REST
- Need to maintain the code

Dataflow architecture



MODELING THE DATA

- Can be tricky to model the data well
- A lot of many to many relationships
  - Semantic model has one or more reports who have one or more semantic models who belong to one or more workspaces where a user has access to none, some or all of the objects.
  - Some things are easier -> Workspace has one app and each object has one workspace
- Might be best to model the activity and inventory separately
- Might be best to have many smaller models to answer specific questions

Capacity monitoring

Build your own solution

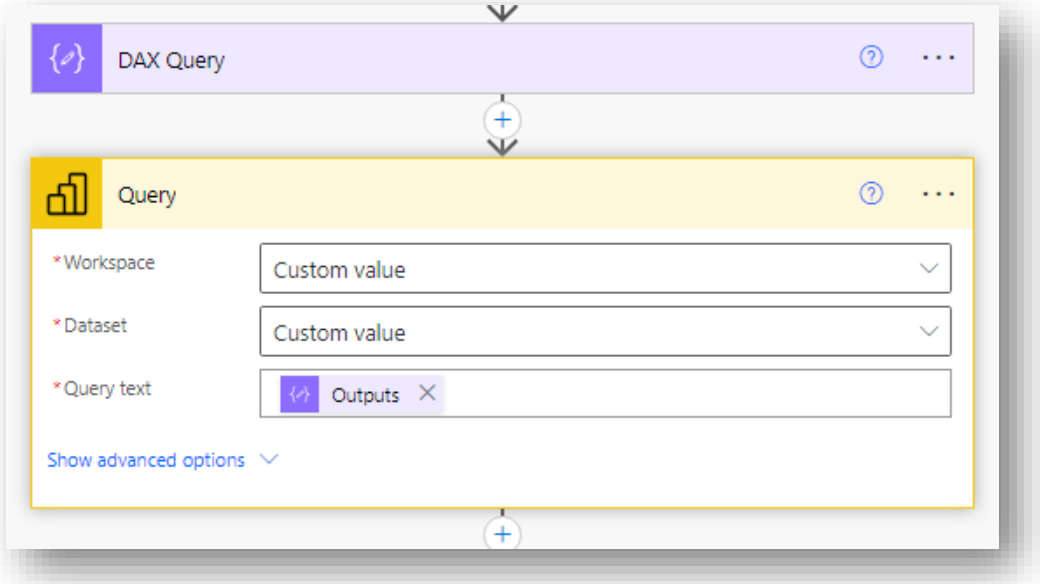
Modify the Metrics App to meet your needs

- Build a custom report off the semantic model
- Send DAX queries to the metrics app semantic model in your own solution
  - Power Automate, Notebook (SemPy), PowerShell, etc.
  - Get throttling % values (Interactive Delay, Interactive Rejection, and/or Background Rejection)
  - Latest values and/or trends over time
  - Best for summarized data only (e.g., hour, day)
- Collect data from multiple capacities and store it long term.

```
# Get max date from current delta table (to avoid loading duplicate days)
try:
    df_max = spark.sql(f'''
    SELECT MAX(Date) as MaxDate
    FROM throttling;
    '''
    )
    maxdate = df_max.first()['MaxDate']
except:
    maxdate = datetime.today() + timedelta(days=-6)
maxdateforDAX = maxdate.strftime('%Y,%m,%d')

if maxdate.date() < (datetime.today() + timedelta(days=-1)).date():

    # Get data for each capacity, write daily csv and append delta
    for capacity in lst_capacities:
        querytext = '''\
                    DEFINE
                    MPARAMETER 'CapacityID' = "{capID}"
                    VAR yesterday =
                        FILTER(ALL('Dates'[Date] ), 'Dates'[Date] < TODAY() && 'Dates'[Date] > DATE({MD}) )

                    EVALUATE
                    SUMMARIZECOLUMNS(
                        'Dates'[Date],
                        'TimePoints'[Start of Hour],
                        yesterday,
                        "IntDelay", ROUND( 'All Measures'[Dynamic InteractiveDelay %] * 100, 2 ),
                        "IntReject", ROUND( 'All Measures'[Dynamic InteractiveRejection %] * 100, 2 ),
                        "BackReject", ROUND( 'All Measures'[Dynamic BackgroundRejection %] * 100, 2 )
                    )
                    '''.format(capID=capacity, MD=maxdateforDAX)
        df_throttling = fabric.evaluate_dax(workspace=MetricsWS, dataset=MetricsModel, dax_string=querytext)
        if len(df_throttling) >= 1:
            df_throttling.columns = df_throttling.columns.str.replace(r'(.*\[)|(\].*)', '', regex=True)
            df_throttling.columns = df_throttling.columns.str.replace(' ', '_')
            df_throttling['capacityId'] = capacity
            filename = capacity + '_throttling_' + (datetime.today()).strftime('%Y%m%d') + '.csv'
            df_throttling.to_csv("/lakehouse/default/Files/ThrottlingData/" + filename)
            spk_throttle = spark.createDataFrame(df_throttling)
            spk_throttle.write.mode("append").format("delta").option("overwriteSchema", "true").saveAsTable('Throttling')
```

Modify the Metrics App to meet your needs

- Build a custom report off the semantic model
- Send DAX queries to the metrics app semantic model in your own solution
  - Power Automate, Notebook (SemPy), PowerShell, etc.
  - Get throttling % values (Interactive Delay, Interactive Rejection, and/or Background Rejection)
  - Latest values and/or trends over time
  - Best for summarized data only (e.g., hour, day)
- Incorporate Metrics App queries into custom solutions



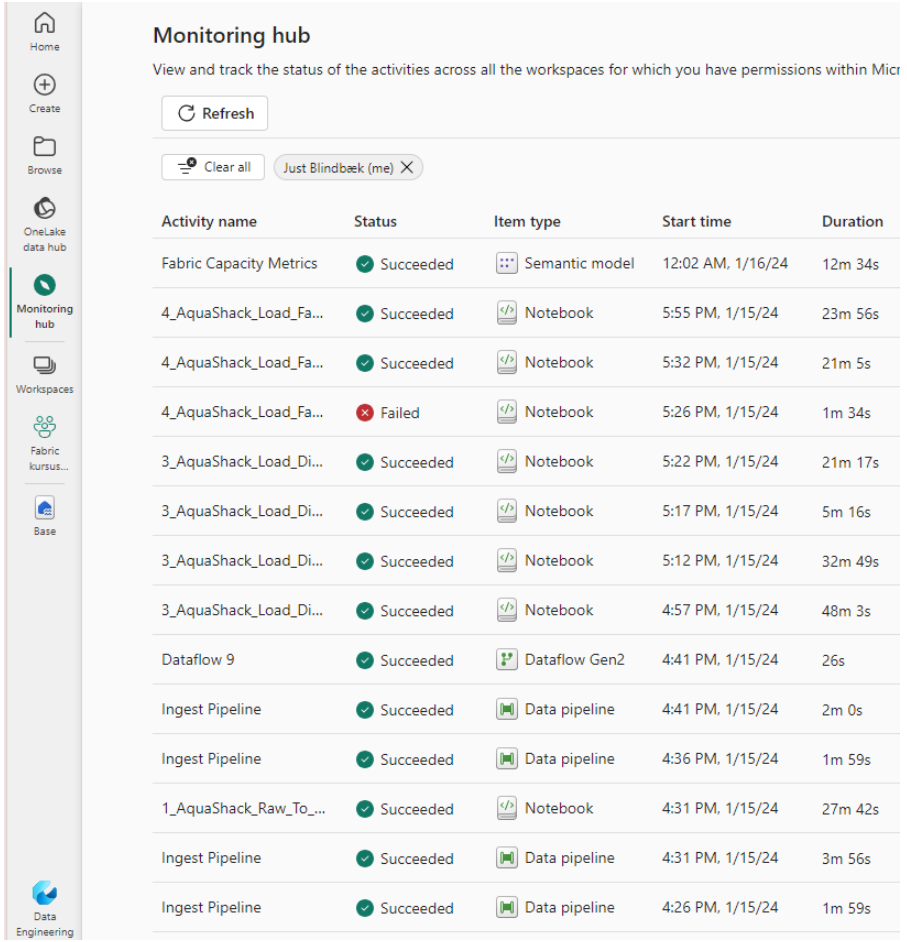
Job monitoring

Build your own solution

Job monitoring

- Built-in monitoring hub is very limited
- No reports and dashboards
- No pro-active alerts

- But then what?



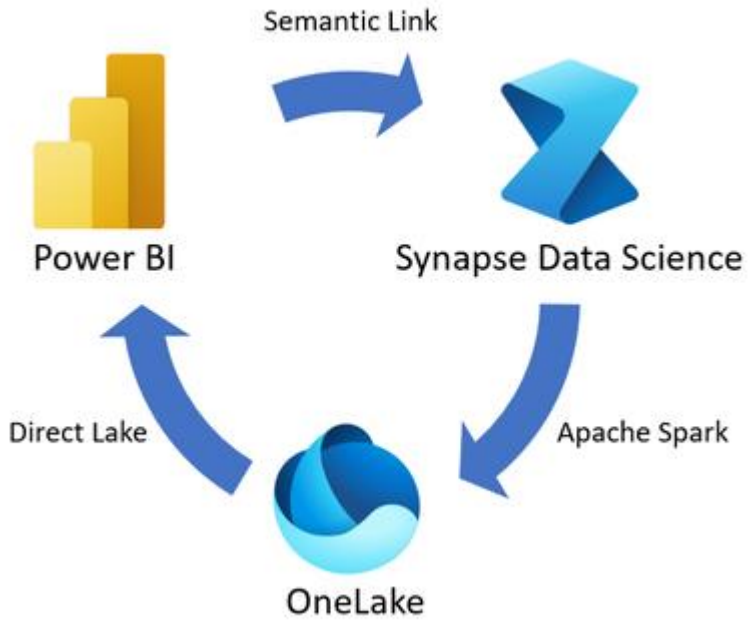
Semantic Models monitoring

Build your own solution

Semantic Models monitoring

- Use Semantic Link and Semantic Links Labs to query all or subset of semantic models in the tenant.

- **Extract:**
- Best Practice Analyzer (BPA) stats
- Vertipaq Analyzer
- Direct Lake Temperature
- And possible much more…



https://learn.microsoft.com/en-us/fabric/data-science/semantic-link-overview

https://github.com/microsoft/semantic-link-labs/

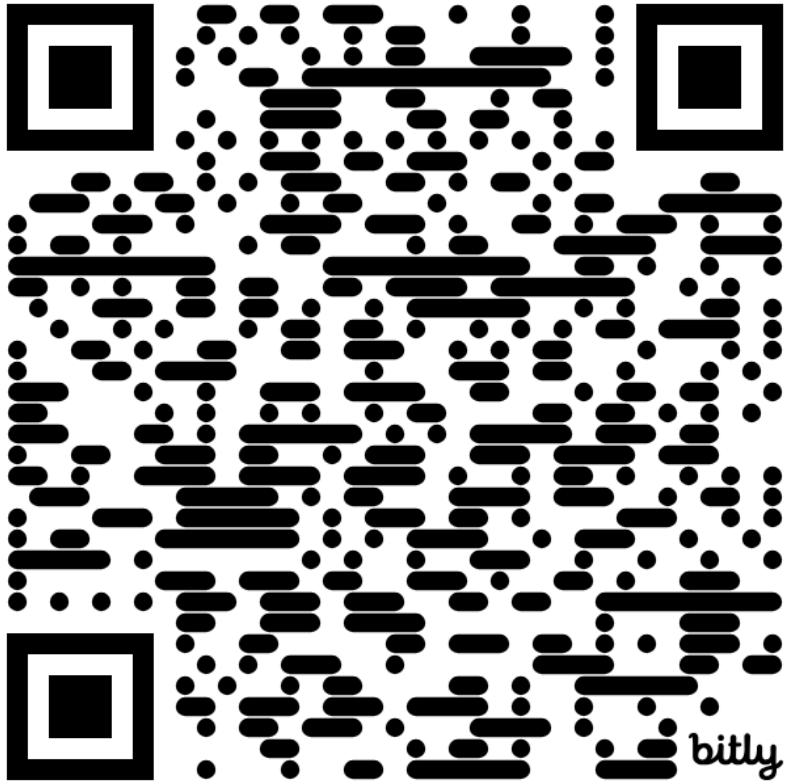
Workspace Item Events

Build your own solution

Workspace Item events

- Use Data Activator to monitor:
  - Creation
  - Modification
  - Deletion
- Of all items in Fabric workspace(s)

Session Feedback 💖



https://bit.ly/dMC2024_SessionFeedback