**Microsoft Purview Integration with Microsoft Fabric: Current Capabilities**

Microsoft Purview and Microsoft Fabric are part of the Microsoft Intelligent data platform that allows you to store, analyze, and govern your data. With Microsoft Fabric and Microsoft Purview together you're able to govern your entire estate and lineage of data.

The integration currently provides:

* **Microsoft Purview Unified Catalog** - automatically view metadata about your Microsoft Fabric items in the Microsoft Purview Unified Catalog with live view in Microsoft Purview
* **Microsoft Purview Information Protection** - allows you to discover, classify, and protect Fabric data using sensitivity labels from Microsoft Purview Information Protection
* **Microsoft Purview Data Loss Prevention (DLP)** - DLP policies are currently supported in Fabric for Power BI semantic models only
* **Microsoft Purview Audit** - all Microsoft Fabric user activities are logged and available in the Microsoft Purview audit log

**End-to-End Implementation Guide**

**Phase 1: Foundation Setup**

**1.1 Enable Microsoft Purview Hub in Fabric**

Microsoft Purview hub is a centralized place in Fabric that helps Fabric administrators manage and govern their organization's Fabric data estate

**Steps:**

1. Navigate to Fabric settings → Microsoft Purview hub
2. Access requires Fabric administrator role or higher
3. If this is the first time insights are being generated, it might take some time for them to appear

**1.2 Configure Workspace Architecture for Dev/Test/Prod**

**Best Practice Recommendations:**

* Build separate databases for development and testing in order to protect production data and not overload the development database with the entire volume of production data
* For development purposes, a best practice is to have isolated workspaces per developer, so that they can work on their own without interfering with the shared workspace

**Recommended Structure:**

Development Environment:

├── Dev-DataEngineering-WS

├── Dev-Analytics-WS

└── Individual Developer Workspaces

Test Environment:

├── Test-DataEngineering-WS

└── Test-Analytics-WS

Production Environment:

├── Prod-DataEngineering-WS

└── Prod-Analytics-WS

**Phase 2: Security Configuration**

**2.1 Implement Sensitivity Labels**

**Create Sensitivity Label Hierarchy:**

Public (Priority 0)

Internal (Priority 1)

Confidential (Priority 2)

Highly Confidential (Priority 3)

**Configuration Steps:**

1. Access Microsoft Purview compliance portal → Information Protection → Sensitivity labels
2. Create labels with appropriate encryption and protection settings
3. Sensitivity labels from Purview Information Protection provide a simple way for your users to classify critical content in Power BI without compromising productivity or the ability to collaborate

**2.2 Configure Data Loss Prevention (DLP)**

**For Power BI Semantic Models:**

* A DLP policy for Power BI is set up in the Microsoft Purview compliance portal. It can detect sensitive data in a semantic model that's been published to a Premium workspace in the Power BI service
* Configure policies based on sensitivity labels and sensitive information types
* When a DLP policy for Power BI detects a sensitive dataset, a policy tip can be attached to the dataset in the Power BI service

**2.3 Workspace-Level Security**

**Development Workspace Security:**

* Restrict to data engineers and developers only
* Enable workspace identity for secure data access
* Fabric workspace identities can only be created in workspaces associated with a Fabric F SKU capacity

**Test Workspace Security:**

* Access for QA team, data engineers, and business analysts
* Implement row-level security (RLS) testing
* Use managed private endpoints for secure data source connections

**Production Workspace Security:**

* You can limit viewer access to data using row-level security (RLS), column-level security (CLS) and object-level security (OLS)
* Strict access controls with business users having viewer permissions only
* Enable audit logging for all activities

**Phase 3: Persona-Based Access Controls**

**3.1 Role-Based Access Matrix**

| **Persona** | **Development** | **Test** | **Production** | **Permissions** |
| --- | --- | --- | --- | --- |
| Data Engineers | Admin | Admin | Contributor | Full data transformation access |
| Business Analysts | Contributor | Contributor | Viewer | Report creation and analysis |
| Data Scientists | Contributor | Contributor | Viewer | Model development and testing |
| Business Users | None | Viewer | Viewer | Report consumption only |
| Compliance Officers | Viewer | Viewer | Viewer | Audit and governance oversight |

**3.2 Implement Item-Level Security**

Item security provides the flexibility to grant access to an individual Fabric item without granting access to the entire workspace

**Phase 4: Data Protection Implementation**

**4.1 Sensitivity Label Inheritance**

* Data owners can apply a sensitivity label to a lakehouse or any other Fabric item, and the label will flow with the data to all downstream items in Fabric
* Configure automatic label inheritance from data sources
* Power BI datasets that connect to sensitivity-labeled data in Azure Synapse Analytics Azure SQL Database and Excel files stored in OneDrive or SharePoint Online can automatically inherit those labels

**4.2 Export Protection**

* When a user doesn't have sufficient permissions according to the encryption settings of the sensitivity label on the .pbix file, they won't be able to open the file
* These labels and their protection settings are also automatically applied to Microsoft 365 files that are exported from Fabric

**Phase 5: Deployment Pipeline Configuration**

**5.1 Implement Git Integration**

Microsoft Fabric Git integration allows you to introduce version control to various items within a Microsoft Fabric workspace

**Configuration Steps:**

1. Connect workspaces to Azure DevOps repositories
2. Make sure you have an isolated environment to work in, so others don't override your work before it gets committed
3. Implement branch protection policies
4. Configure automated deployment pipelines

**5.2 Set Up Deployment Pipelines**

The deployment process lets you clone content from one stage in the deployment pipeline to another, typically from development to test, and from test to production

**Pipeline Configuration:**

Development Stage → Test Stage → Production Stage

**Phase 6: Monitoring and Compliance**

**6.1 Configure Audit Logging**

* You can use the Power BI activity log to track activities related to sensitivity labels. When you've implemented DLP for Power BI, the activity log tracks when there's a DLP rule match
* Set up alerts for sensitive data access
* Monitor label changes and downgrades

**6.2 Implement Continuous Monitoring**

Use the Purview hub insights to monitor:

* Sensitive data distribution across workspaces
* Item endorsement status
* Domain governance
* DLP policy violations

**Additional Security Measures**

**6.1 Network Security**

* Managed private endpoints can be created in your Fabric workspaces to connect to data sources in Azure that have blocked public internet access
* Implement trusted workspace access for ADLS Gen2
* Configure firewall rules for data sources

**6.2 Encryption**

* In Fabric, all data that is stored in OneLake is encrypted at rest
* Customer-managed keys (CMK) allows you to encrypt data at-rest using your own keys
* Consider workspace customer-managed keys for additional encryption layer

**6.3 Multi-Geo Considerations**

* Customers can also create Multi-Geo capacities located in geographies (geos) other than their home region
* Ensure data residency compliance
* Configure appropriate capacity assignments

**Implementation Timeline**

**Week 1-2:** Foundation setup and workspace architecture **Week 3-4:** Security configuration and sensitivity labels **Week 5-6:** DLP policies and access controls **Week 7-8:** Deployment pipelines and Git integration **Week 9-10:** Monitoring setup and testing **Week 11-12:** User training and go-live

**Key Success Metrics**

* 100% of sensitive data classified with appropriate labels
* Zero unauthorized data exports detected
* Full audit trail for all data access
* Successful automated deployments across environments
* User adoption of security controls

This implementation approach ensures comprehensive protection of your Microsoft Fabric platform while maintaining productivity and enabling secure collaboration across your development, test, and production environments.