

Lesson 20 — Publishing & Sharing

1) How does Power BI handle large datasets in the Online Service, and what is the role of Premium Capacity in this?

Power BI handles large data via VertiPaq compression (Import), Incremental Refresh with partitioning, Aggregations, Composite models (Import + DirectQuery), query caching, and Gen2 capacity autoscale. Premium/PPU unlocks dedicated capacity, larger model-size limits (large models), higher refresh concurrency/frequency, XMLA read/write, query scale-out, and enhanced refresh—so very large models and heavy usage remain fast and reliable.

2) Differences between Import, DirectQuery, and Live Connection in the Service

- Import: Data is copied into the model (in-memory). Fast interactions; requires scheduled/On-demand refresh; limited by model-size quotas.
- DirectQuery: No data stored in the model; each interaction sends SQL (or native) queries to the source. Near-real-time; performance depends on the source; some DAX/time-intelligence and modeling features are limited.
- Live Connection: The report connects live to an external semantic model (e.g., Analysis Services or a published Power BI dataset). The model lives outside the report; you cannot add tables/relationships in the report; security/roles come from the source model.

3) Deployment pipelines in Power BI Online—stages

Three managed stages: Development → Test → Production. You assign workspaces to stages, compare changes, and deploy forward with rules (e.g., parameter/connection mapping), enabling gated promotion, consistency, and rollback across environments.

4) Integrating Power BI with Microsoft Teams or SharePoint

- Teams: Add the Power BI app to Teams; pin a report/dashboard as a tab; paste “share” links or visual links into chats; comment and notify colleagues directly in Teams with the same permissions as in the Service.
- SharePoint Online: Use the Power BI web part to embed a report page; viewers must have access to the underlying report/dataset; useful for portals and intranets.

5) What is the XMLA endpoint in Premium and how it helps

XMLA endpoint (read/write in Premium or PPU) exposes the tabular model so BI teams can use SSMS, Tabular Editor/ALM Toolkit, TOM/TMSL to script, automate, partition, deploy, and govern models (roles, calculations, lineage, version-control-friendly workflows). It brings enterprise-grade lifecycle management to Power BI models.

6) Usage metrics and audit logs in the Service

- Usage metrics: per-artifact reports (views, viewers, trends) for reports/dashboards and workspace-level metrics; Premium adds capacity metrics app.
- Audit/Activity logs: tenant-wide logs via Microsoft 365/Unified Audit Log and the Power BI Activity Log/REST API; capture events like view, share, export, publish, dataset actions—useful for compliance and monitoring.

7) Managing workspace access and permissions

Use workspace roles (Admin, Member, Contributor, Viewer) and Azure AD groups. Control dataset Build/Read/Reshare permissions; manage “App” audiences for read-only consumption; prefer group-based access; follow least-privilege; manage permissions per artifact when needed.

8) Enforcing data governance in the Service

Levers include: Admin portal tenant settings; sensitivity labels (MIP) and data protection; endorsements (Promoted/Certified) and discovery; DLP policies (Purview) for Power BI; lineage and impact analysis; RLS/OLS; service principals and deployment pipelines; naming standards and workspace/capacity governance.

9) Limitations of RLS with DirectQuery or Live Connection

- DirectQuery: PBI RLS filters are applied but push down to the source—performance can degrade; some modeling features are restricted; with Single Sign-On to some sources, source-side security is the authoritative layer.

- Live connection to AS or to a published dataset: RLS must be defined in the source model (you cannot define different PBI-side RLS in the thin report); mixed-source scenarios aren't supported in a single live model.

General note: RLS doesn't replace source security and must be tested for each connector.

10) Refreshing a dataset via Power Automate or the REST API

- Power Automate: Use the "Refresh a dataset" action in the Power BI connector; schedule or trigger from other events; optionally poll refresh status and send notifications.

- REST API: POST /groups/{groupId}/datasets/{datasetId}/refreshes to trigger, GET refreshes to check status; enhanced refresh (Premium/PPU) supports advanced parameters/partial refresh.

Authenticate with Azure AD (service principal or user) and grant Dataset permissions.