1. Power BI handles large datasets in the Online Service by allowing semantic models to grow beyond 10 GB using the large semantic model storage format, which is available in Premium capacities. Premium Capacity plays a key role by providing enhanced resources, such as increased storage limits up to 100 TB per tenant, automatic aggregations, and autoscaling to manage overloads, enabling efficient processing and querying of large-scale data without importing everything at once.

2. Import mode imports data into Power BI, allowing for fast querying but requiring scheduled refreshes. DirectQuery mode does not import data; instead, it queries the underlying data source in real-time for each visual refresh, which can be slower for large datasets but keeps data current. Live Connection is similar to DirectQuery as it doesn't import data and queries the source (like Analysis Services) live, but it's optimized for published semantic models and doesn't support composite models in the same way.

3. Deployment pipelines in Power BI Online are a CI/CD tool for managing content lifecycle across stages, allowing teams to develop, test, and deploy reports, datasets, and dashboards efficiently. They include three default stages: Development (for building and initial testing), Test (for validation in a separate environment), and Production (for live deployment to end-users). Stages can be customized, made public, or adjusted in number, with features like content comparison and automated deployments via APIs.

4. Power BI Service integrates with Microsoft Teams by allowing users to add the Power BI app, embed reports directly into channels or chats via tabs, and collaborate on content in real-time. For SharePoint, Power BI reports can be embedded into pages, and files from Teams/SharePoint can be connected to Power BI by opening them in SharePoint and using Excel app integration. This enables seamless sharing, viewing, and discussion of Power BI content within these collaboration tools.

5. The XMLA endpoint in Power BI Premium is a connectivity feature that enables read/write access to semantic models using standard XMLA protocols, supporting tools like SQL Server Management Studio or Visual Studio. It benefits developers and enterprise BI teams by allowing advanced operations such as incremental refresh, partition management, metadata modifications, and integration with provider data sources, enhancing automation, real-time data handling, and compatibility with Azure Analysis Services.

6. Usage metrics in Power BI Service track client-side interactions like views and performance for reports, dashboards, and semantic models, providing insights into adoption and popularity (e.g., via modern usage metrics reports). Audit logs capture service-side activities, such as user actions and operations, for compliance and monitoring, accessible by admins. While usage metrics include details not in audit logs (like page views), both help in understanding and optimizing usage, with settings to enable/disable them organization-wide.

7. Workspace access and permissions in Power BI are managed by assigning roles: Admin (full control, including adding users), Member (create and edit content, manage permissions), Contributor (edit content but not manage users), or Viewer (view-only access). Admins can add users or groups via the workspace settings, set contact lists, and use granular controls for flexible permissions. Sharing reports grants access without workspace roles, but direct dashboard sharing requires workspace access.

8. Data governance in Power BI Service is enforced through capabilities like data loss prevention (DLP) policies integrated with Microsoft 365 to detect and protect sensitive data, adoption roadmaps for structured governance, compliance dashboards for insights, and features for monitoring, protecting, and improving data discoverability. It includes planning for DLP implementation, using audit logs, and leveraging Fabric's governance tools to balance innovation with security and compliance.

9. Row-Level Security (RLS) in Power BI restricts data access at the row level using filters, but with DirectQuery or Live Connection, limitations include potential performance impacts from real-time queries to the source, lack of support for certain features like object-level security in live connections to on-premises sources, and dependency on the underlying data source's security model. RLS is

applied at query time, which can increase load on the source, and composite models may have additional constraints.

10. To refresh a dataset via Power Automate, create a flow that triggers on a schedule or event and uses the Power BI connector to execute a refresh action. For REST API, use the Power BI REST APIs like "Refresh Dataset" endpoint to trigger asynchronous refreshes, specifying options like notifyOption or enhanced parameters for commit modes and retries. Both methods support monitoring via get refresh history/execution details APIs, handling failures with automatic retries, and are suitable for automated, large-scale refreshes.