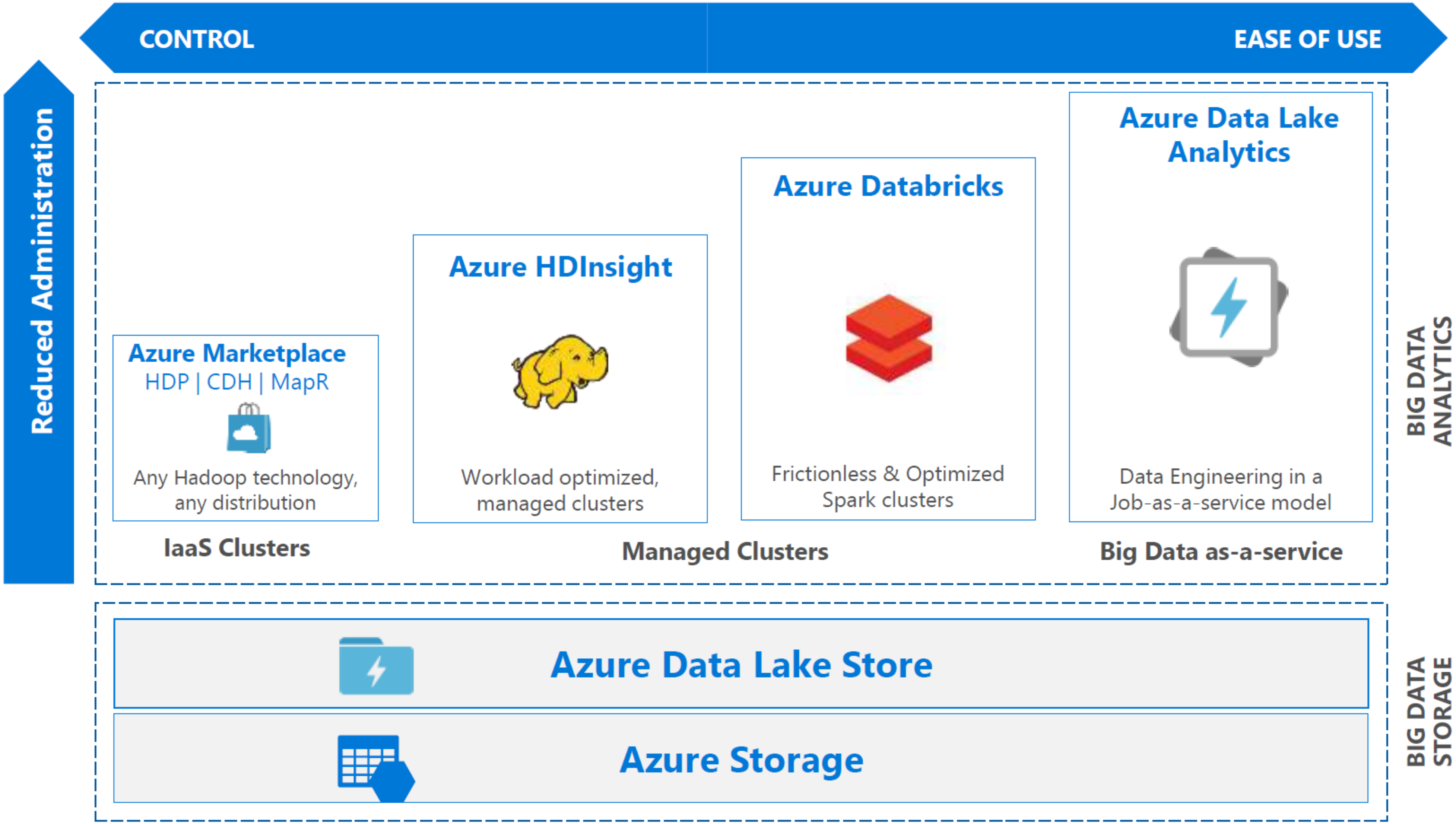
Agenda

- Perform data engineering with Databricks
  - – Create a Databricks workspace
  - – Apache Spark notebooks
  - – Read and write data by using Azure Databricks
  - – Perform basic data transformations
  - – Perform advanced data transformation
  - – Create data pipelines by using Databricks Delta
  - – Create data visualizations by using Azure Databricks and Power BI

Create a Databricks workspace

Knowing the various big data solutions



CONTROL

EASE OF USE

Reduced Administration

Azure Marketplace
HDP | CDH | MapR

Any Hadoop technology, any distribution

Azure HDInsight

Workload optimized, managed clusters

Azure Databricks

Frictionless & Optimized Spark clusters

Azure Data Lake Analytics

Data Engineering in a Job-as-a-service model

IaaS Clusters

Managed Clusters

Big Data as-a-service

BIG DATA ANALYTICS

Azure Data Lake Store

Azure Storage

BIG DATA STORAGE

What is Azure Databricks ?

A fast, easy and collaborative Apache® Spark™ based analytics platform optimized for Azure

databricks

Best of Databricks

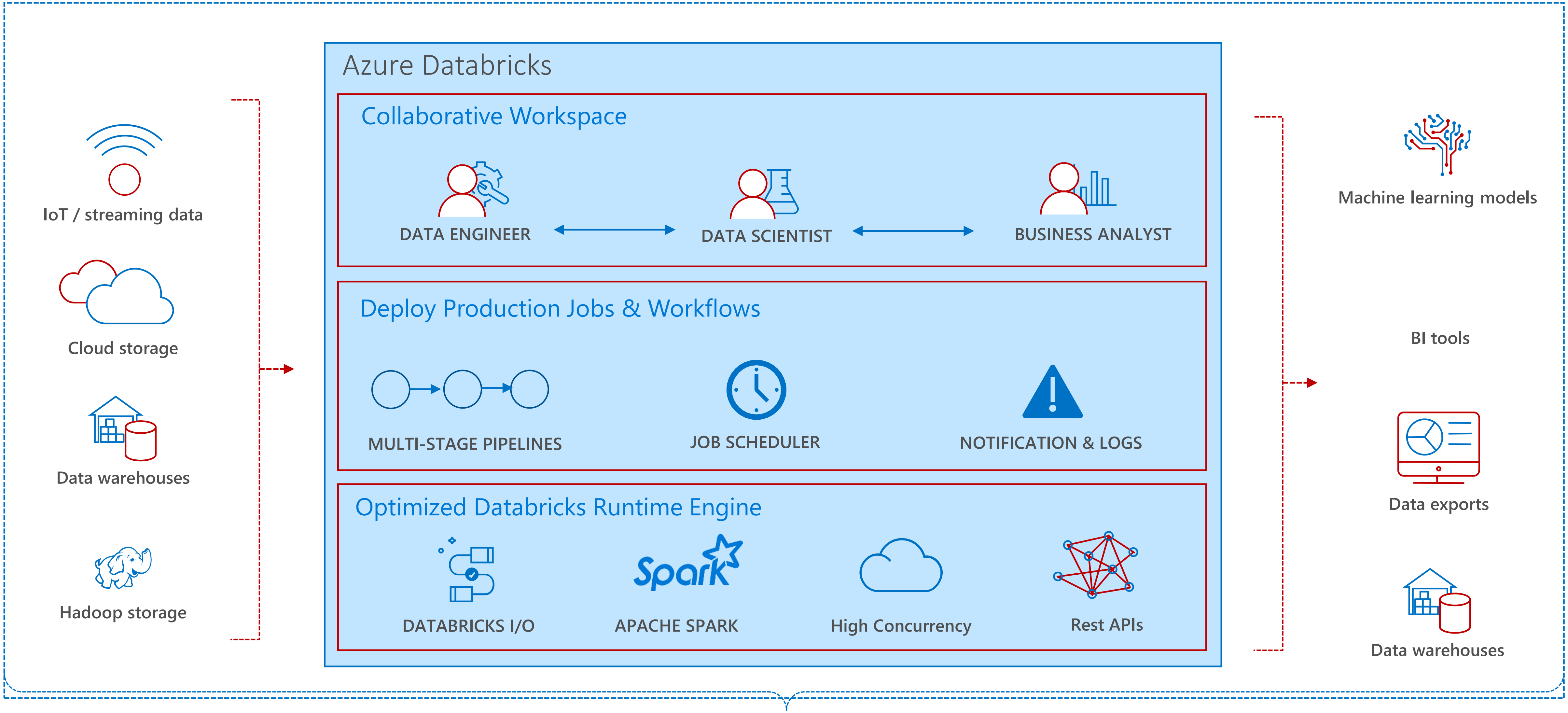
+

Microsoft

Best of Microsoft

- Designed in collaboration with the founders of Apache Spark
- One-click set up; streamlined workflows
- Interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.
- Native integration with Azure services (Power BI, SQL DW, Cosmos DB, Blob Storage, ADF, SQL DB, AAD)
- Enterprise-grade Azure security (Active Directory integration, compliance, enterprise-grade SLAs – 99.95%)

Azure Databricks

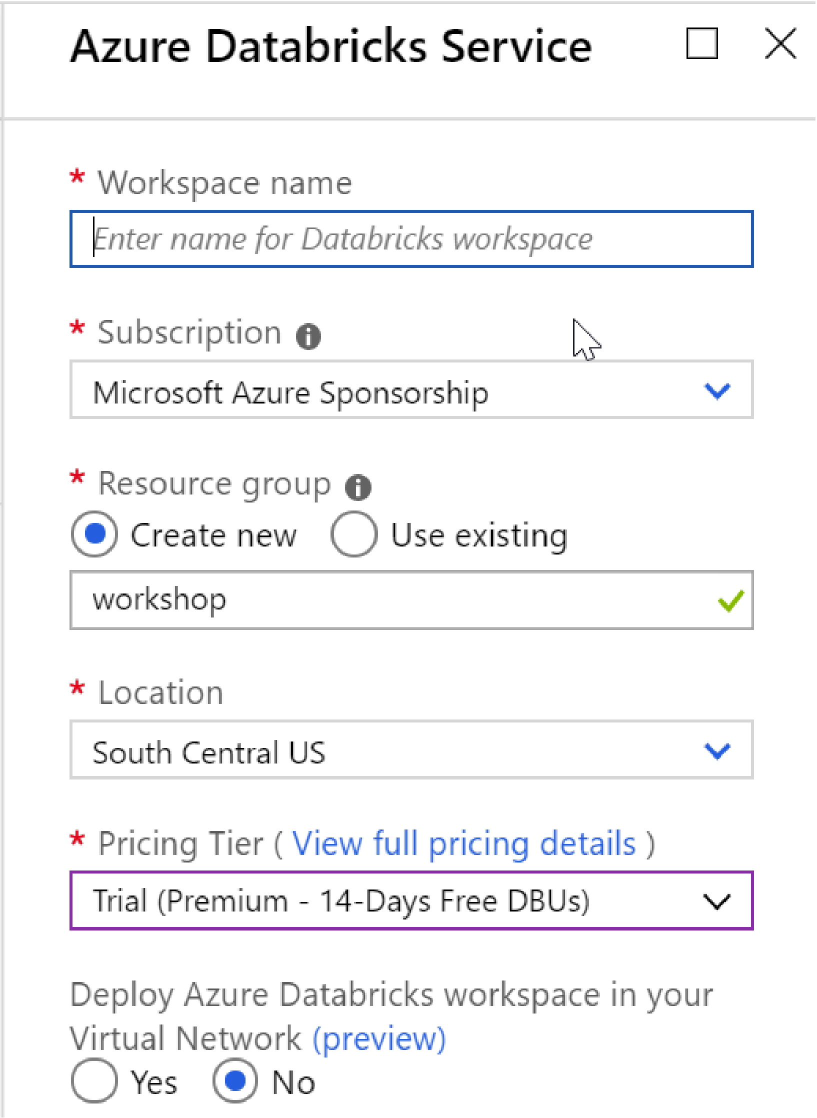


Enhance Productivity

Build on secure & trusted cloud

Scale without limits

Lab: Create a Databricks workspace



Azure Databricks Service

* Workspace name

Enter name for Databricks workspace

* Subscription

Microsoft Azure Sponsorship

* Resource group

● Create new ○ Use existing

workshop ✓

* Location

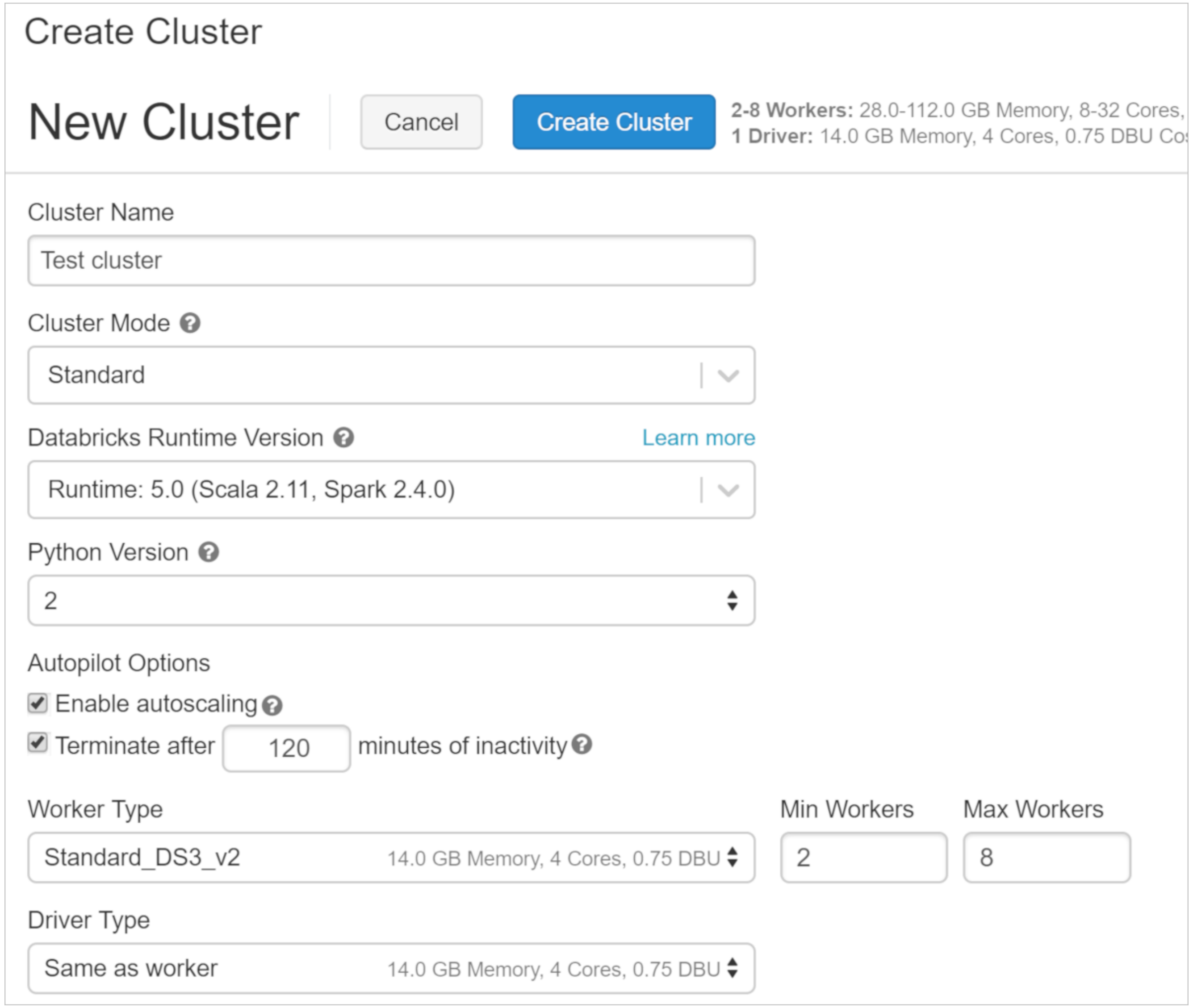
South Central US

* Pricing Tier (View full pricing details)

Trial (Premium - 14-Days Free DBUs)

Deploy Azure Databricks workspace in your Virtual Network (preview)

○ Yes ● No



Create Cluster

New Cluster

2-8 Workers: 28.0-112.0 GB Memory, 8-32 Cores,
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU Co

Cluster Name

Test cluster

Cluster Mode

Standard

Databricks Runtime Version

Runtime: 5.0 (Scala 2.11, Spark 2.4.0)

Python Version

2

Autopilot Options

☑ Enable autoscaling

☑ Terminate after 120 minutes of inactivity

Worker Type		Min Workers	Max Workers
Standard_DS3_v2	14.0 GB Memory, 4 Cores, 0.75 DBU	2	8

Driver Type

Same as worker 14.0 GB Memory, 4 Cores, 0.75 DBU

Apache Spark notebooks

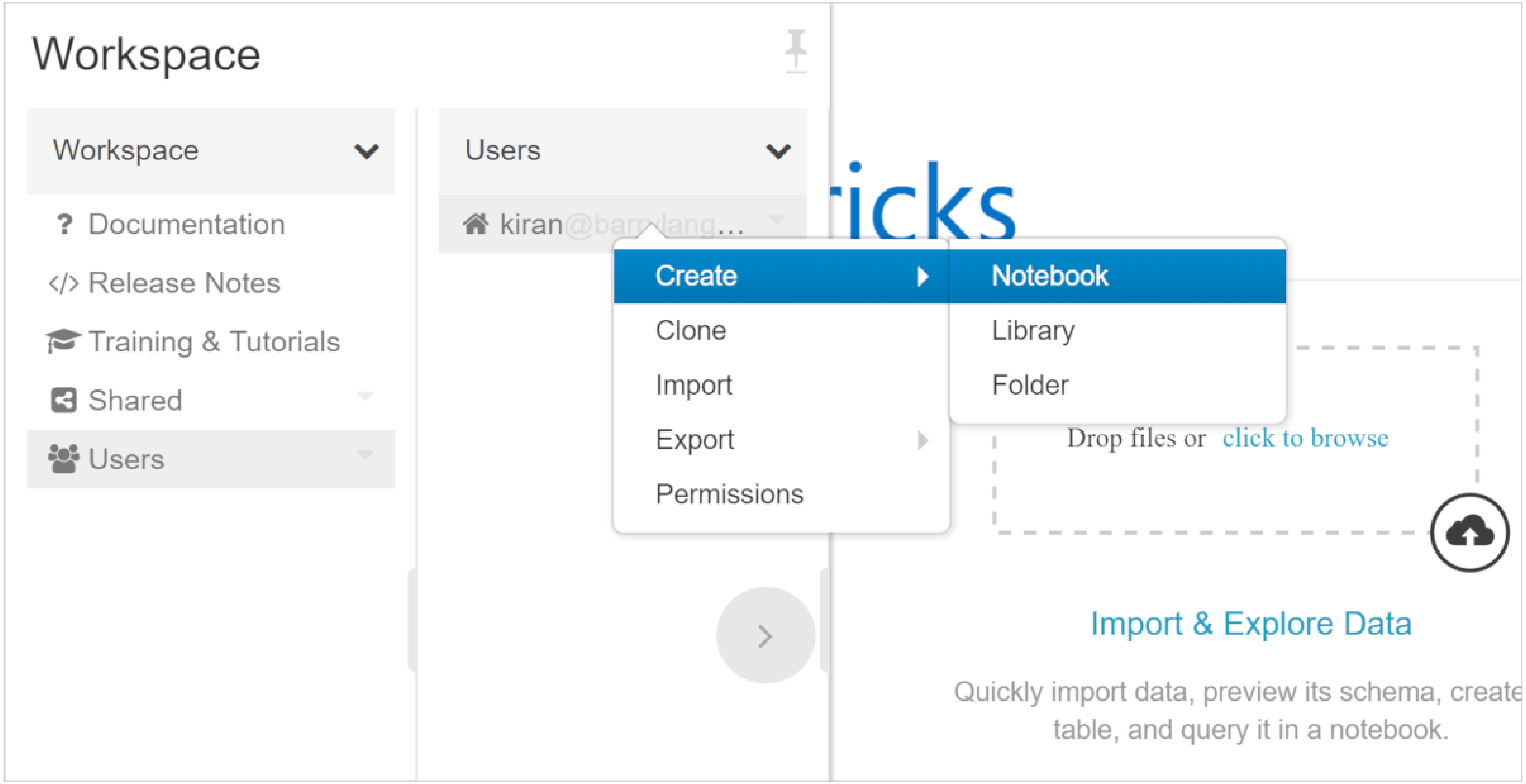
Notebooks

- A notebook is a collection of cells
- These cells are run to execute code, to render formatted text, or to display graphical visualizations
- The notebooks are backed by clusters, or networked computers, that work together to process your data

Use Apache Spark notebooks

- You can use Apache Spark notebooks to:
  - Read and process huge files and data sets
  - Query, explore, and visualize data sets
  - Join disparate data sets found in data lakes
  - Train and evaluate machine learning models
  - Process live streams of data
  - Perform analysis on large graph data sets and social networks

Lab: Apache Spark notebooks



Drop files or click to browse

Import & Explore Data

Quickly import data, preview its schema, create table, and query it in a notebook.

Read and write data by using Azure Databricks

Resilient Distributed Datasets (RDDs)

- DataFrames are data structure where data is organized into named columns, like a table in a relational database, but with richer optimizations available
- DataFrames are derived from resilient distributed datasets (RDDs)
- RDDs and DataFrames are immutable distributed collections of data
  - Resilient: RDDs are fault tolerant, so if part of your operation fails, Spark quickly rebuilds any lost data
  - Distributed: RDDs are distributed across networked machines known as a cluster

Read and write functions

- A DataFrame object comes with methods attached to it
- Spark provides a number of built-in functions that can be used directly with DataFrames
  - Accessing data
  - Joins and aggregation
  - Working with hierarchical data
  - And more

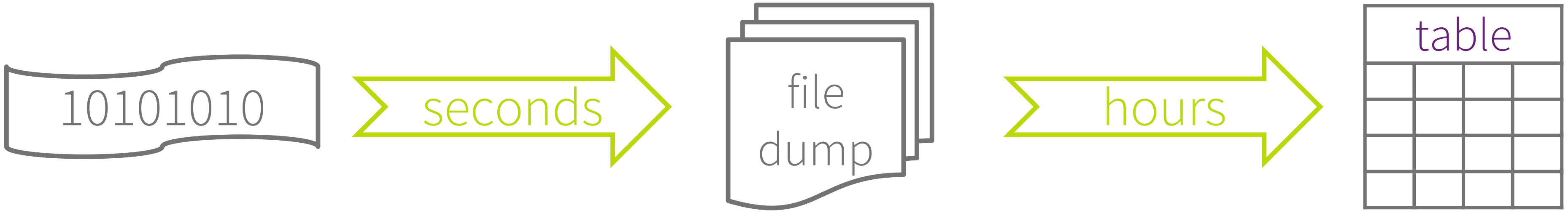
Lab: Read and write data by using Azure Databricks

- **01-Getting-Started**: This notebook gets you started with your Databricks workspace.
- **02-Querying-Files**: This notebook contains exercises to help you query large data files and visualize your results.
- **03-Joins-Aggregations**: You do basic aggregation and joins in this notebook.
- **04-Accessing-Data**: This notebook lists the steps for accessing data from various sources by using Databricks.
- **05-Querying-JSON**: In this notebook, you learn how to query JSON and hierarchical data with DataFrames.
- **06-Data-Lakes**: This notebook lists exercises that show how to create an Azure Data Lake Storage Gen2 instance and use Databricks DataFrames to query and analyze this data.
- **07-Azure-Data-Lake-Gen2**: In this notebook, you use Databricks to query and analyze data stores in Azure Data Lake Storage Gen2.
- **08-Key-Vault-backed-secret-scopes**: This notebook lists the steps for configuring a Key Vault-backed secret scope. You'll create a Key Vault-backed secret scope and securely store in it usernames and passwords for a sample SQL database and an Azure Cosmos DB instance to be used in the following notebooks.
- **09-SQL-Database-Connect-Using-Key-Vault**: In this notebook, you'll connect to a SQL database by using your Azure SQL username and password that you created and securely stored in the Key Vault-backed secret scope in the previous notebook.
- **10-Cosmos-DB-Connect-Using-Key-Vault**: In this notebook, you'll connect to an Azure Cosmos DB instance by using the Azure Cosmos DB username and password that you previously created and securely stored in the Key Vault-backed secret scope.
- **Exploratory-Data-Analysis**: This notebook is in the *Optional* subfolder. It includes a sample project for you to explore later.

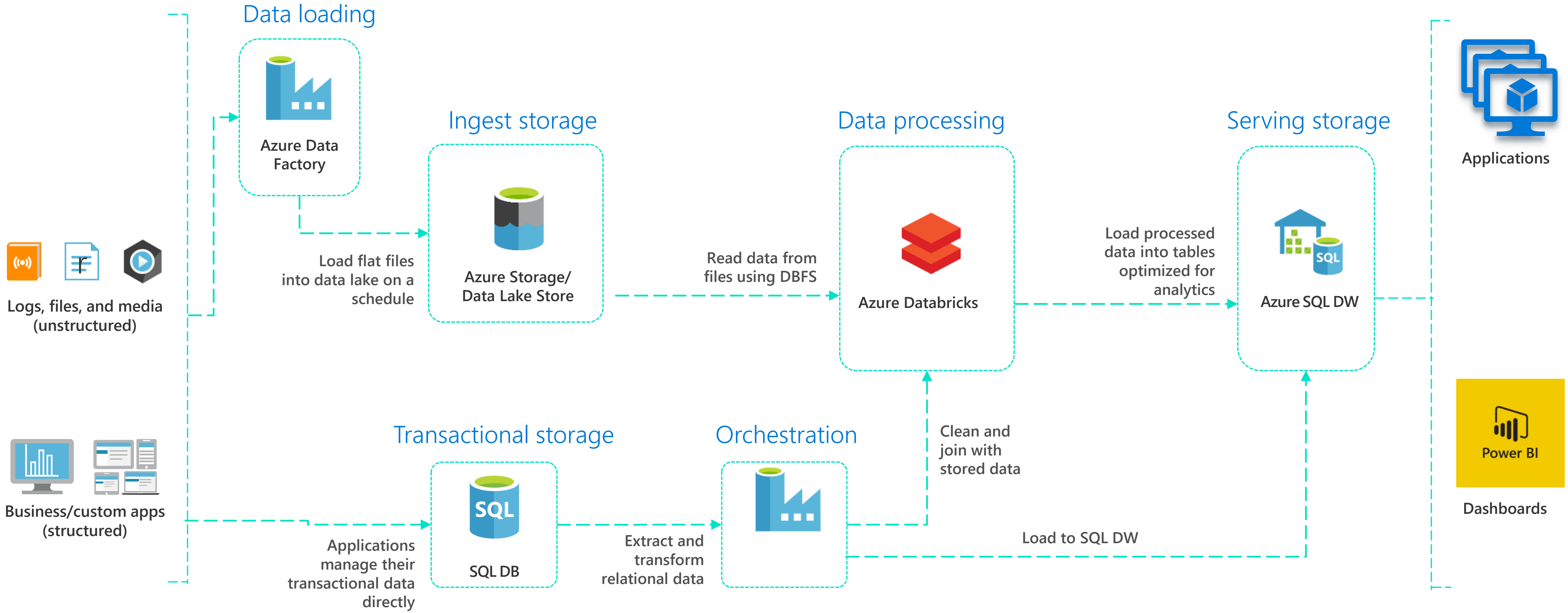
Perform basic data transformations

Traditional ETL

- Raw, dirty, un/semi-structured is data dumped as files
- Periodic jobs run every few hours to convert raw data to structured data ready for further analytics



Modern data warehousing


Data loading
Azure Data Factory
Ingest storage
Data processing
Serving storage
Applications
Logs, files, and media (unstructured)
Load flat files into data lake on a schedule
Azure Storage/ Data Lake Store
Read data from files using DBFS
Azure Databricks
Load processed data into tables optimized for analytics
SQL
Azure SQL DW
Transactional storage
Orchestration
Clean and join with stored data
Power BI
Dashboards
Business/custom apps (structured)
SQL
SQL DB
Extract and transform relational data
Load to SQL DW
Applications manage their transactional data directly

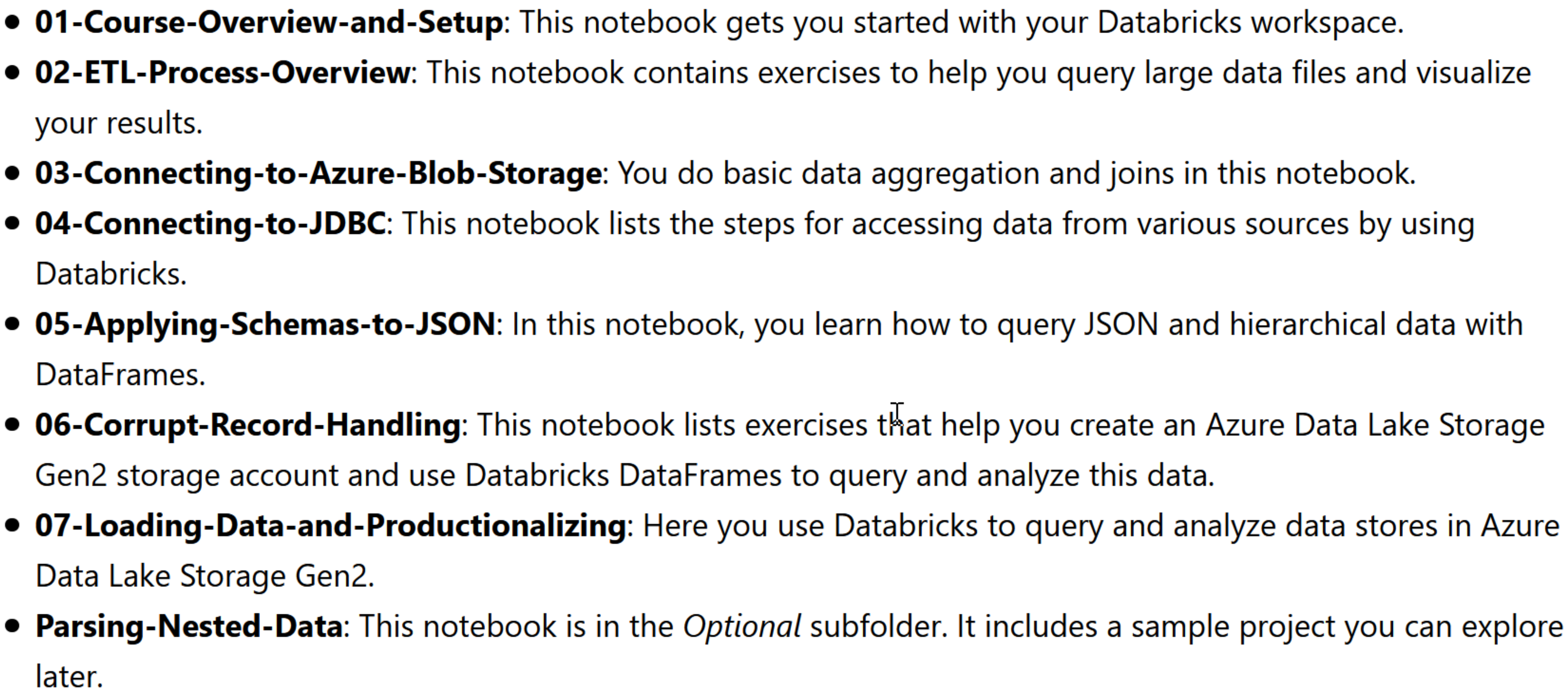
An ETL query in Spark

```
val csvTable = spark.read.csv("/source/path")

val jdbcTable = spark.read.format("jdbc")
  .option("url", "jdbc:postgresql:...")
  .option("dbtable", "TEST.PEOPLE")
  .load()

csvTable
  .join(jdbcTable, Seq("name"), "outer")
  .filter("id <= 2999")
  .write
  .mode("overwrite")
  .format(”delta")
  .saveAsTable("outputTableName")
```

Performance tips

- Land data in Blob Store/ADLS partitioned into separate directory
  - Avoid high list cost on large directories
- Parallelization of Azure Databricks streaming is driven by number of partitions in Eventhub
  - For best query performance use Delta table
  - Alternatively, use regular Spark table backed by Parquet
- Avoid small files
  - File size 100s MB – 1GB preferred
  - Delta supports compaction

Lab: Perform basic data transformations



- **01-Course-Overview-and-Setup**: This notebook gets you started with your Databricks workspace.
- **02-ETL-Process-Overview**: This notebook contains exercises to help you query large data files and visualize your results.
- **03-Connecting-to-Azure-Blob-Storage**: You do basic data aggregation and joins in this notebook.
- **04-Connecting-to-JDBC**: This notebook lists the steps for accessing data from various sources by using Databricks.
- **05-Applying-Schemas-to-JSON**: In this notebook, you learn how to query JSON and hierarchical data with DataFrames.
- **06-Corrupt-Record-Handling**: This notebook lists exercises that help you create an Azure Data Lake Storage Gen2 storage account and use Databricks DataFrames to query and analyze this data.
- **07-Loading-Data-and-Productionalizing**: Here you use Databricks to query and analyze data stores in Azure Data Lake Storage Gen2.
- **Parsing-Nested-Data**: This notebook is in the *Optional* subfolder. It includes a sample project you can explore later.

Perform advanced data transformation

Business requirements for the BI

- The business community must accept the BI system to deem it successful
- Make information easily accessible
- Present information consistently
- Adapt to change
- Present information in a timely way
- Be a secure bastion that protects the information assets
- Serve as the authoritative and trustworthy foundation for improved decision making

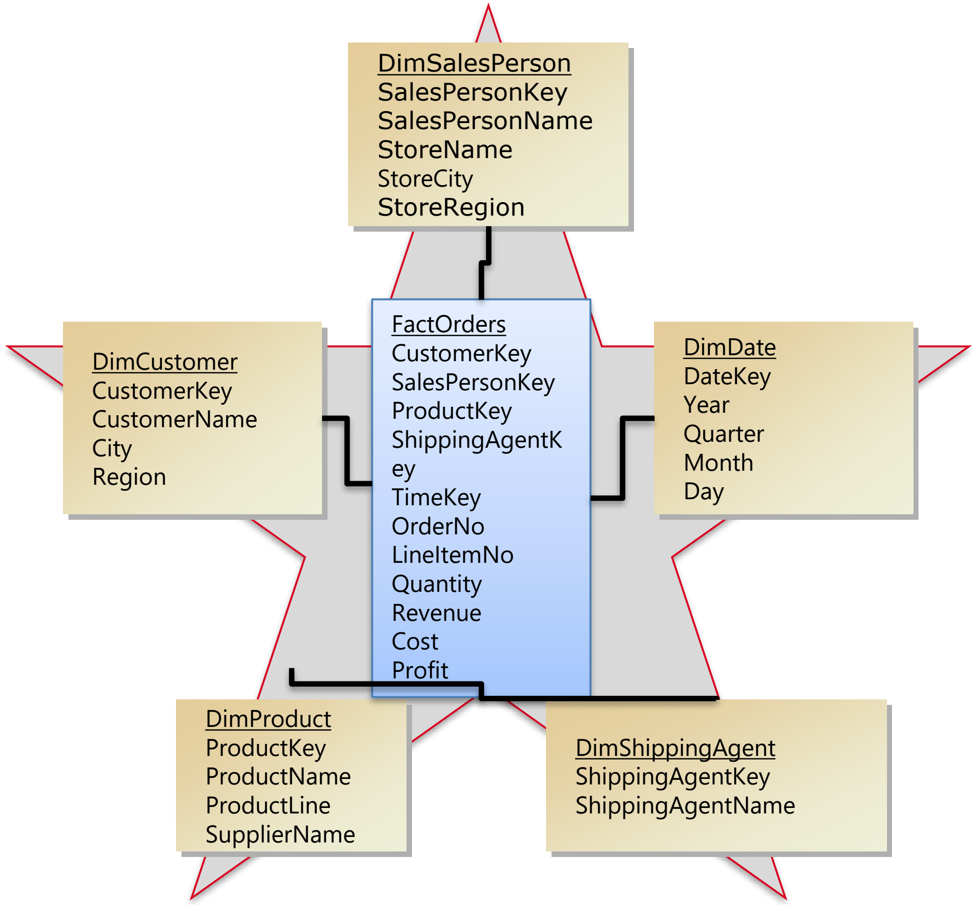
Dimensional modeling

Make everything as simple as possible, but not simpler

Albert Einstein

Star schema

- Group related dimensions into dimension tables
- Group related measures into fact tables
- Relate fact tables to dimension tables by using foreign keys



DimSalesPerson
SalesPersonKey
SalesPersonName
StoreName
StoreCity
StoreRegion

DimCustomer
CustomerKey
CustomerName
City
Region

FactOrders
CustomerKey
SalesPersonKey
ProductKey
ShippingAgentKey
TimeKey
OrderNo
LineItemNo
Quantity
Revenue
Cost
Profit

DimDate
DateKey
Year
Quarter
Month
Day

DimProduct
ProductKey
ProductName
ProductLine
SupplierName

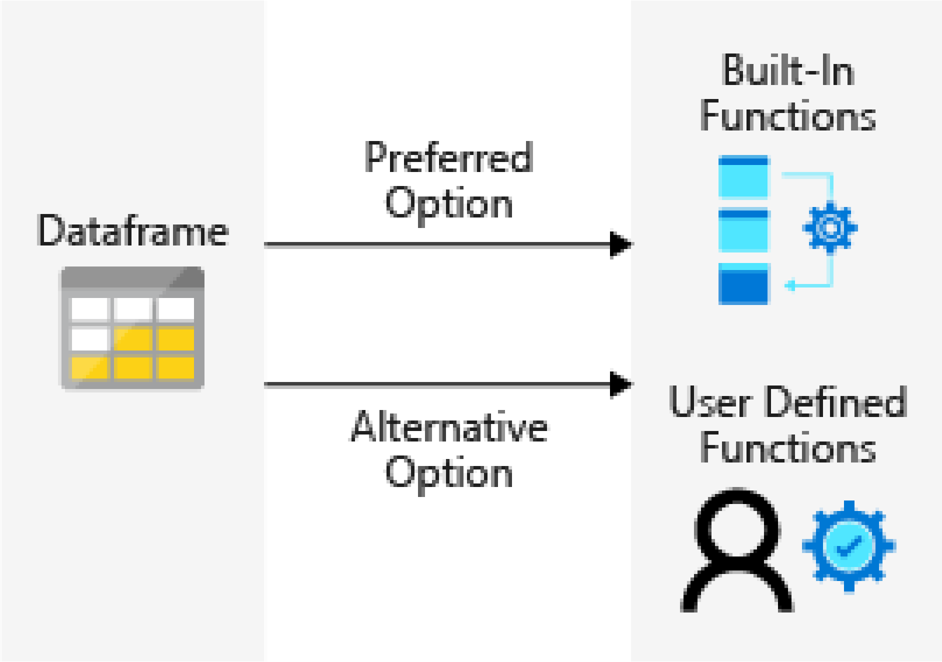
DimShippingAgent
ShippingAgentKey
ShippingAgentName

Common transformations

- Normalizing values
- Imputing null or missing data
- Deduplicating data
- Performing database rollups
- Exploding arrays
- Pivoting DataFrames

Custom and complex transformations with UDFs

- The highly optimized built-in functions in Spark provide a wide array of functionality, covering most data transformation use cases
- Use UDFs when there's no clear way to accomplish a task by using built-in functions



Joins and lookup tables

- A standard (or shuffle) join moves all the data on the cluster for each table to a specific node on the cluster
- A broadcast join fixes this situation when one dataframe is sufficiently small
  - A broadcast join duplicates the smaller dataframe on each node of the cluster, which avoids the cost of shuffling the bigger dataframe

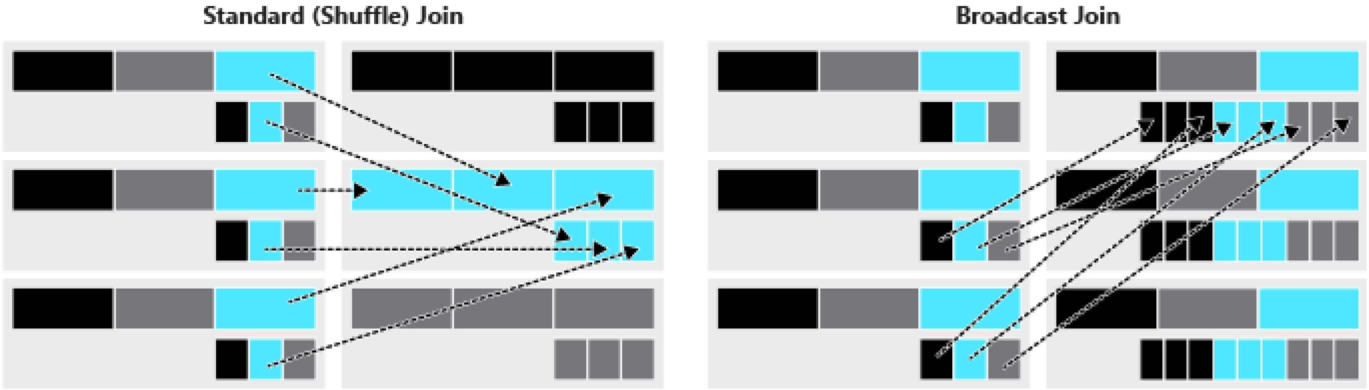
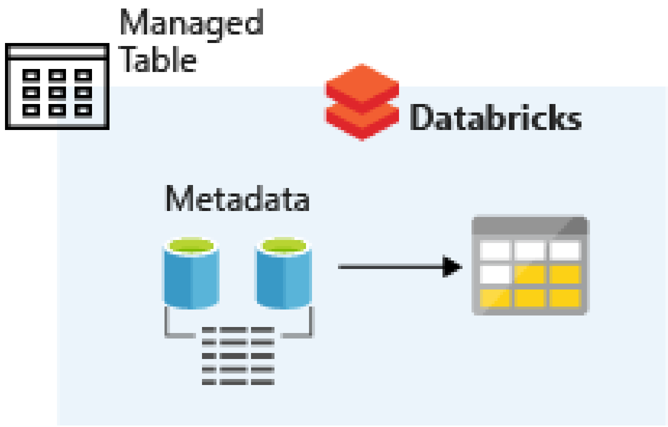
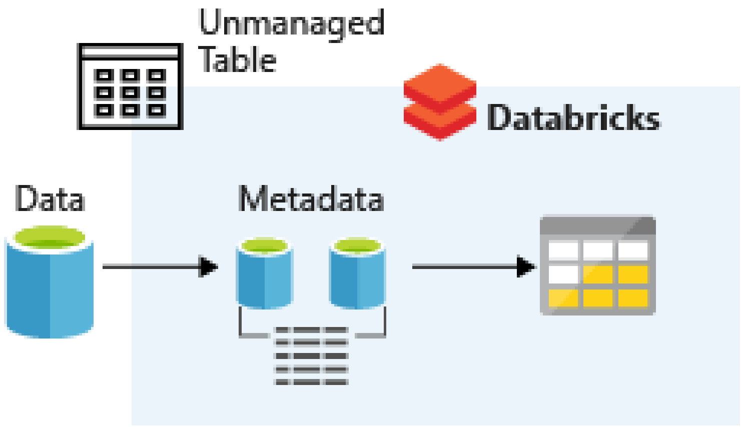


Table management

- A managed table is a table that manages both the actual data and the metadata
  - In this case, a DROP TABLE command removes both the metadata for the table and the data itself
- Unmanaged tables manage the metadata from a table, while the actual data is managed separately and is often backed by a blob store such as Azure Blob storage
  - Dropping an unmanaged table drops only the metadata that is associated with the table while the data remains in place





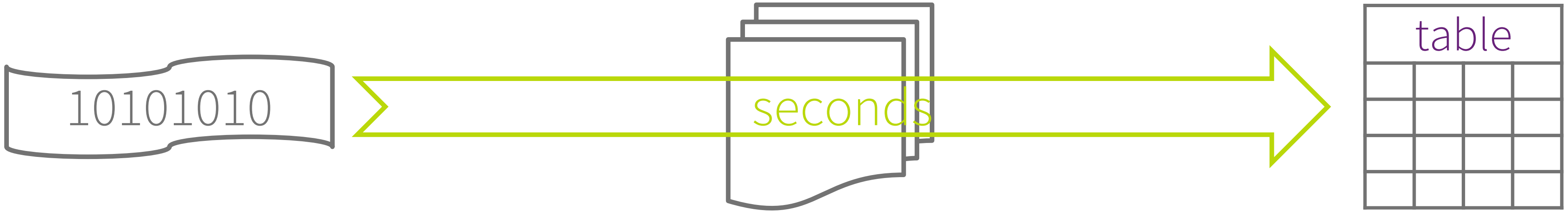
Lab: Perform advanced data transformation

- **01-Course-Overview-and-Setup** - This notebook gets you started with your Databricks workspace.
- **02-Common-Transformations** - In this notebook, you perform some common data transformation by using Spark built-in functions.
- **03-User-Defined-Functions** - In this notebook, you perform custom transformation by using UDFs.
- **04-Advanced-UDFs** - In this notebook, you use advanced UDFs to perform complex data transformations.
- **05-Joins-and-Lookup-Tables** - In this notebook, you learn how to use standard and broadcast joins for tables.
- **06-Database-Writes** - This notebook contains exercises to write data to target databases in parallel, storing the transformed data from your ETL job.
- **07-Table-Management** - In this notebook, you handle managed and unmanaged tables to optimize your data storage.
- **Custom-Transformations** - This notebook is located in the Optional subfolder and includes a sample project for you to explore later.

Create data pipelines by using Databricks Delta

Streaming ETL w/ Structured Streaming

- Structured streaming enables raw data to be available as structured data as soon as possible



Databricks Delta

- Databricks Delta is a transactional storage layer designed specifically to work with Apache Spark and Databricks File System (DBFS)
- Optimize Spark queries for faster analytics, better data reliability guarantees and simplified data pipelines

Azure Databricks Delta

Handle terabytes & petabytes of data

Low latency streaming ingestion

Avoid corrupt & messy data while reading & writing

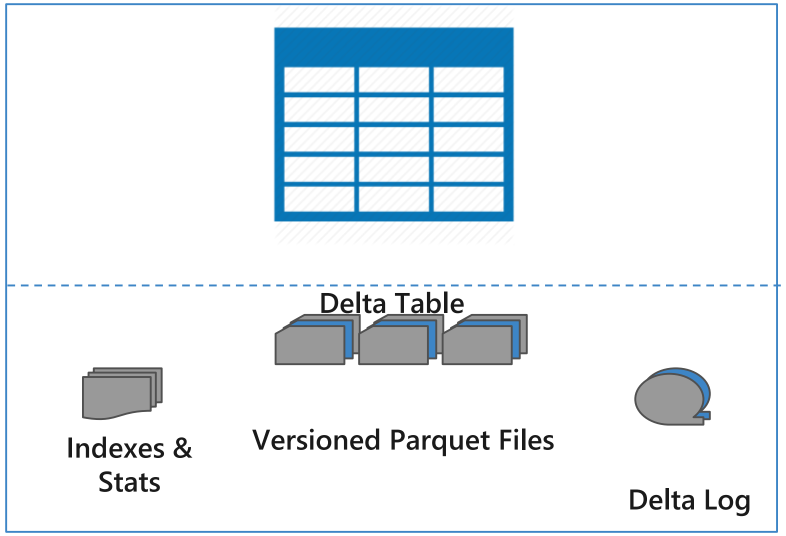
Control on how to adapt to changing schema

Enable scientists & analysts to read data quickly for interactive analysis - Indexing

Azure Databricks Delta tables

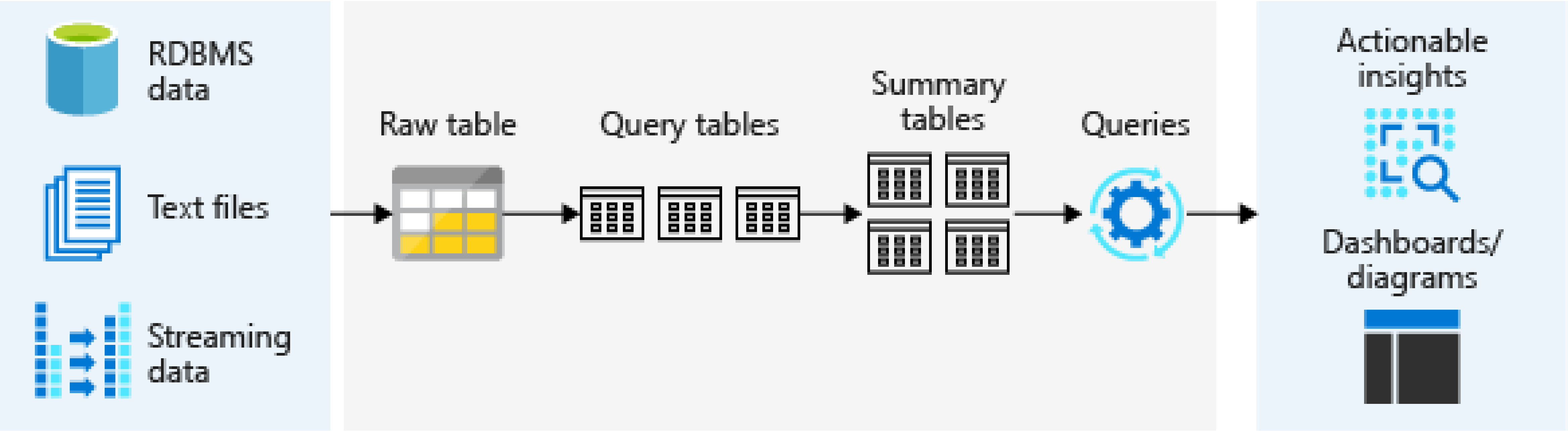
- Delta table = Parquet + Transaction Log
  - Linear history of atomic changes
  - Optimistic Concurrency Control
  - Log checkpoint is stored as Parquet
  - Lazy GC = Free Snapshot Isolation

Delta Table

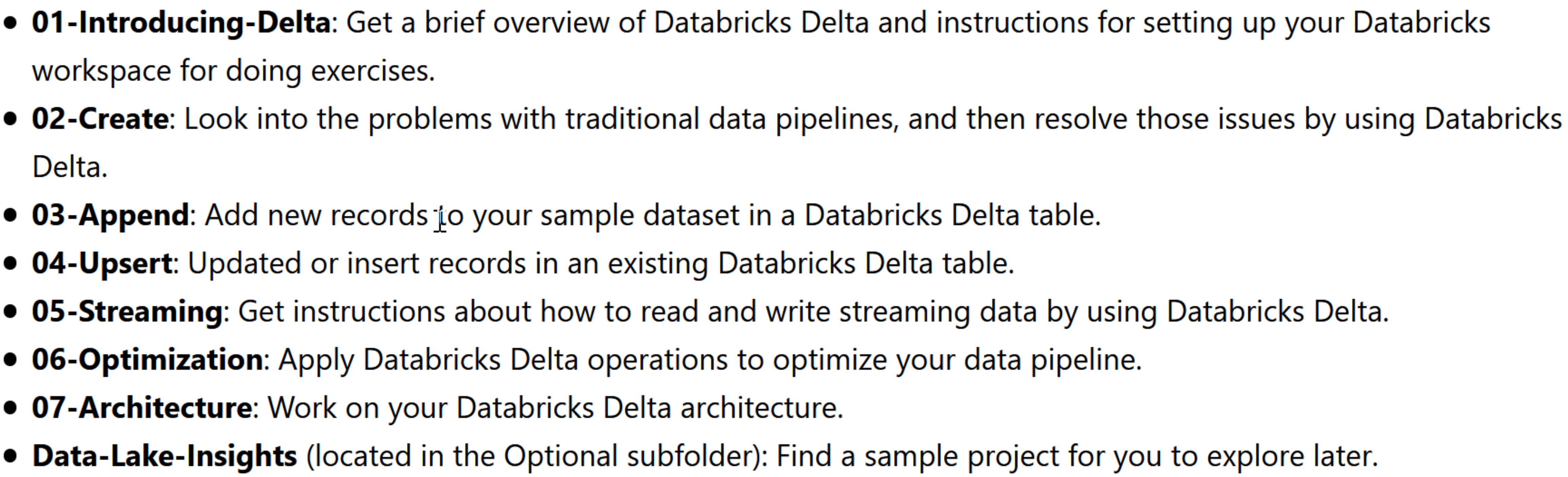


Databricks Delta architecture

- Vast improvement upon the traditional Lambda Architecture



Lab: Create data pipelines by using Databricks Delta



- **01-Introducing-Delta**: Get a brief overview of Databricks Delta and instructions for setting up your Databricks workspace for doing exercises.
- **02-Create**: Look into the problems with traditional data pipelines, and then resolve those issues by using Databricks Delta.
- **03-Append**: Add new records to your sample dataset in a Databricks Delta table.
- **04-Upsert**: Updated or insert records in an existing Databricks Delta table.
- **05-Streaming**: Get instructions about how to read and write streaming data by using Databricks Delta.
- **06-Optimization**: Apply Databricks Delta operations to optimize your data pipeline.
- **07-Architecture**: Work on your Databricks Delta architecture.
- **Data-Lake-Insights** (located in the Optional subfolder): Find a sample project for you to explore later.

Create data visualizations by using Azure Databricks and Power BI

Databricks visualization options

- Built-in functions
  - Databricks provides the built-in display function for visualizations of your data
- Visualize data with Power BI
  - You can connect your Databricks cluster to Power BI to create visualizations for your data
- Visualize data with Matplotlib
  - You can display Matplotlib objects within a Python notebook in Databricks
  - Databricks saves plots as images in the FileStore

Power BI Overview

Data sources

- SaaS solutions
  e.g. Marketo, Salesforce, GitHub, Google analytics
- On-premises data
  e.g. Analysis Services
- Organizational content packs
  Corporate data sources or external data services
- Azure services
  Azure SQL, Stream Analytics…
- Excel files
  Workbook data / data models
- Power BI Desktop files
  Data from files, databases, Azure, and other sources

Power BI service

- Content packs
- Natural language query
- Sharing & collaboration

- Live dashboards
- Visualizations
- Reports
- 01001 10101 Datasets
- Data refresh

Power BI Desktop

Prepare ▶ Explore ▶ Report ▶ Share

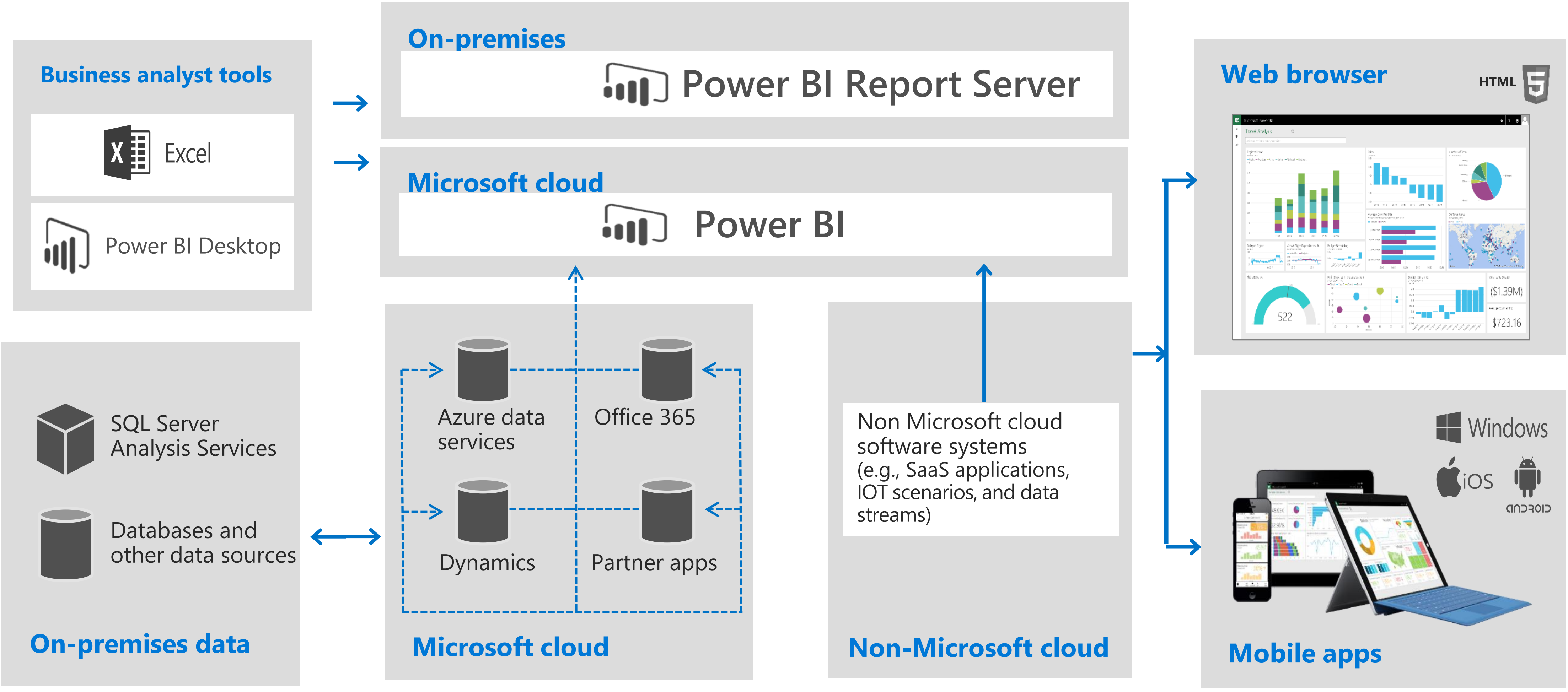
Power BI REST APIs

01100001
00010110

{
"name":{String},
"tables":[
...
]
}

{
"name":{String},
"columns":[
{
"name":{String},
"dataType":{String}
},
...
]
}

Power BI architecture



Business analyst tools

Excel

Power BI Desktop

On-premises

Power BI Report Server

Microsoft cloud

Power BI

Web browser

HTML 5

Mobile apps

Windows

iOS

android

On-premises data

SQL Server Analysis Services

Databases and other data sources

Microsoft cloud

Azure data services

Office 365

Dynamics

Partner apps

Non-Microsoft cloud

Non Microsoft cloud software systems (e.g., SaaS applications, IOT scenarios, and data streams)

Power BI Desktop

Prepare, explore, report and collaborate with Power BI Desktop

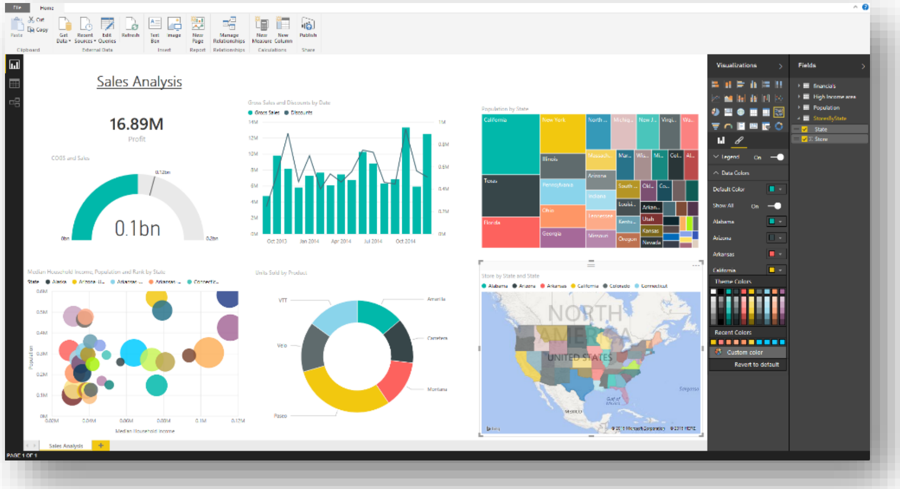
Prepare

Explore

Report

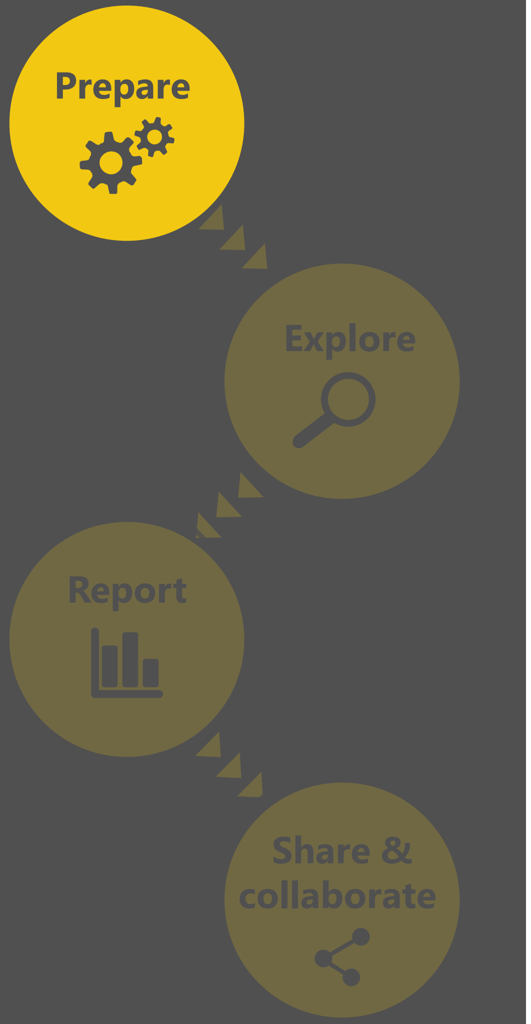
Share & collaborate

- Acquire and prepare data with extensive query capabilities
- Establish data structure and transform and analyze data
- Explore data in new ways through a freeform, drag-and-drop canvas
- Author reports with a broad range of modern data visualizations
- Publish interactive reports to Power BI



Available as a free, downloadable desktop companion to the Power BI service, ***Power BI Desktop is a visual data exploration and reporting tool***

Power BI Desktop


Prepare

Explore

Report

Share & collaborate

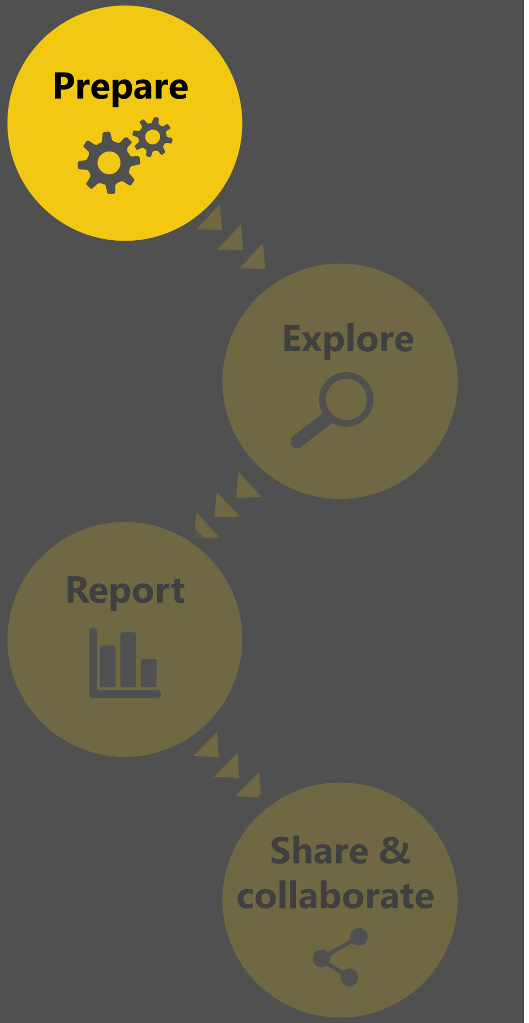
Shape data into the format and structure you need

- Transform data to fit your needs using intuitive UI
    - Select data for inclusion
    - Cleanse data and remove errors
    - Precisely tune the query step sequence: re-order, add, edit or delete steps as needed
    - Modify data types to support specific calculation requirements
- Very powerful for advanced scenarios (M, Mashups)

Common data-shaping tasks

- Remove rows or columns
- Change a data type
- Pivot columns and group rows
- Modify a table name
- Identify and fix errors
- Merge or append queries to combine data from multiple queries into a single query

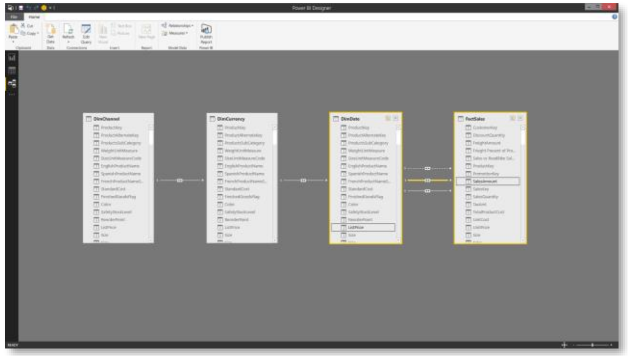
Power BI Desktop



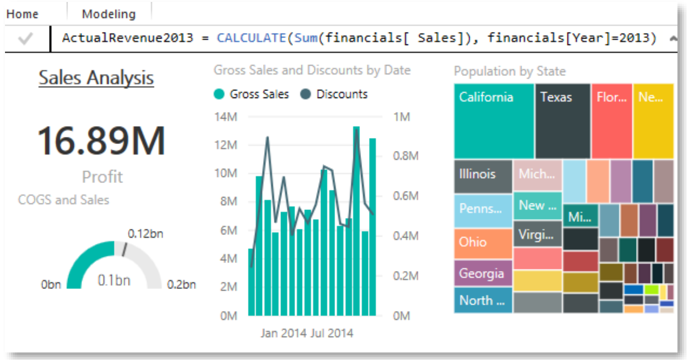
Produce rich data models using formulas and relationships

- Automatically create a model by importing data
  - Desktop detects relationships automatically, categorizes data and applies default summarization
- Refine models to enable complex calculations
  - Create relationships between tables manually or using the AutoDetect feature
  - Adjust relationship type (1:1, many: many, m:1) and cross-filter data for new insights
- Define calculations – known as measures – to generate new fields for use in reports
  - Use automatically generated measures, or create custom measures with Data Analysis Expressions (DAX) formulas
- Develop advanced analytics using a combination of measures and relationships
  - Uncover correlations, highlight exceptions and understand business outcomes

Apply complex schema and business logic to create rich, reusable data models



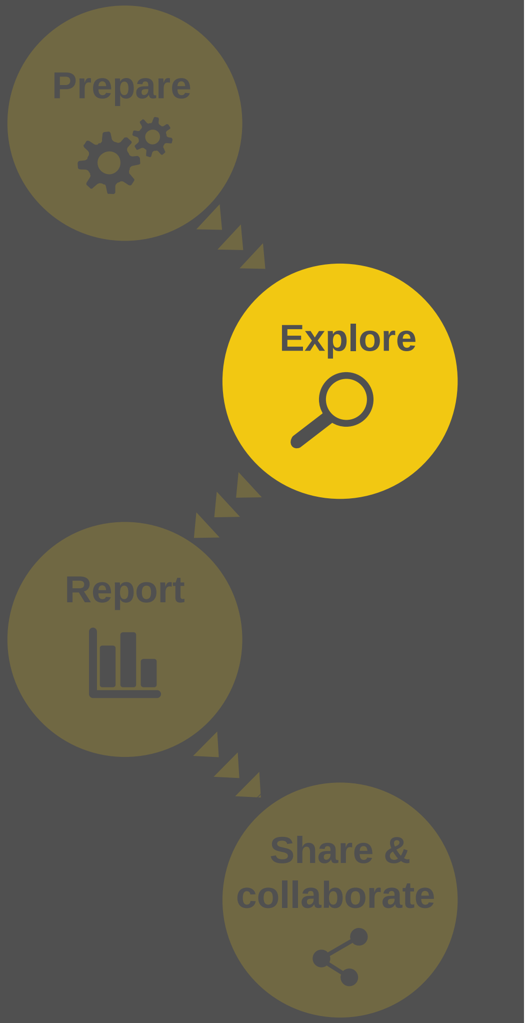
Create and modify relationships



```
ActualRevenue2013 = CALCULATE(Sum(financials[ Sales]), financials[Year]=2013)
```

Define and use measures with DAX formulas

Power BI Desktop



Prepare

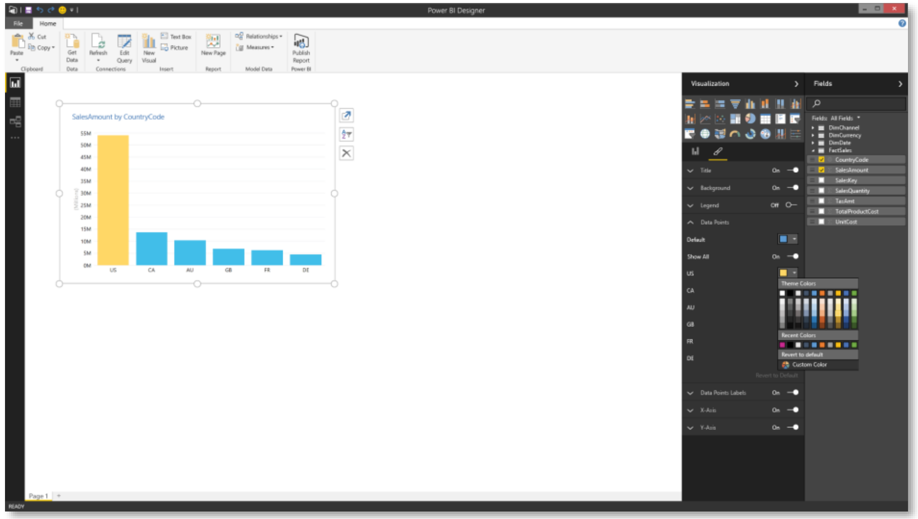
Explore

Report

Share & collaborate

Explore your data with freeform, drag and drop canvas

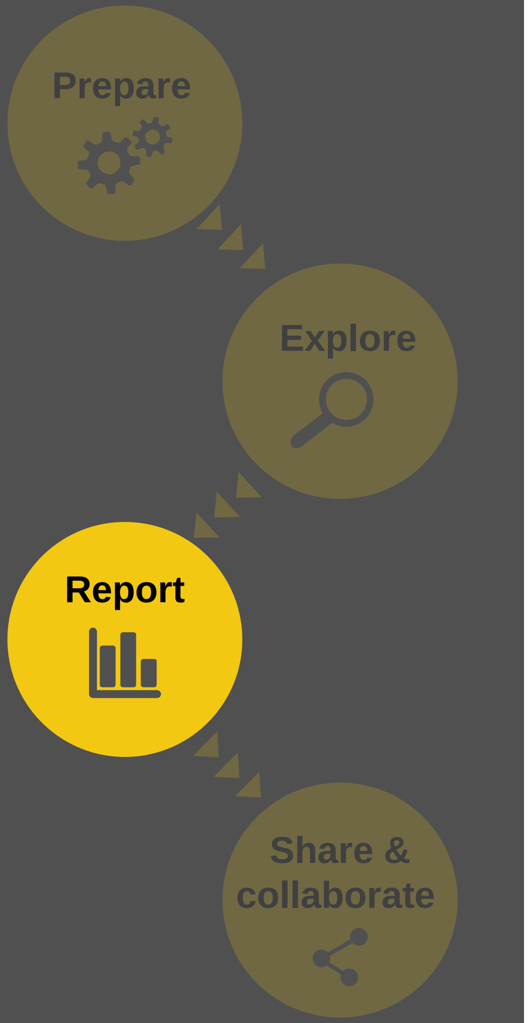
- Explore data in a variety of ways and across multiple visualizations
  - Select data elements and sort data
  - Filter data and use cross-filter capabilities
  - Drill into and across datasets
  - Pivot and slice data
  - Change visualization types
- Select, transform and mashup data via a freeform, drag-and-drop canvas



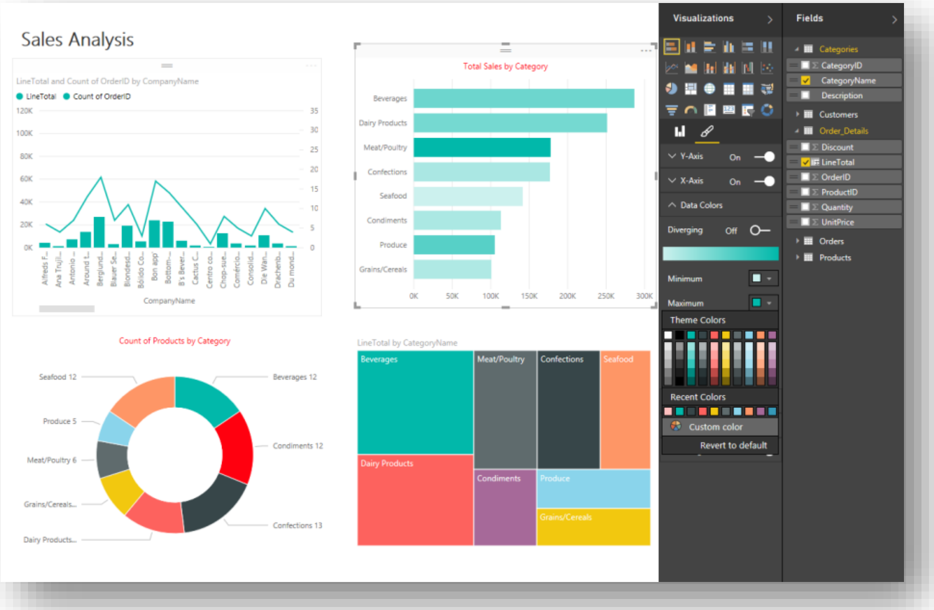
Power BI Desktop allows you to explore your data and create insightful visualizations on a freeform canvas

Power BI Desktop

Deliver valuable insights with customizable visual reports



- Visualize data and easily author reports
  - Depict data in compelling reports that tell stories using a range of interactive visualizations
  - Use data from different sources in a single, consolidated report
- Change colors, format and customize
  - Title, Background Color, Legend, Data Labels
  - New visual color formatting with fixed and data driven settings



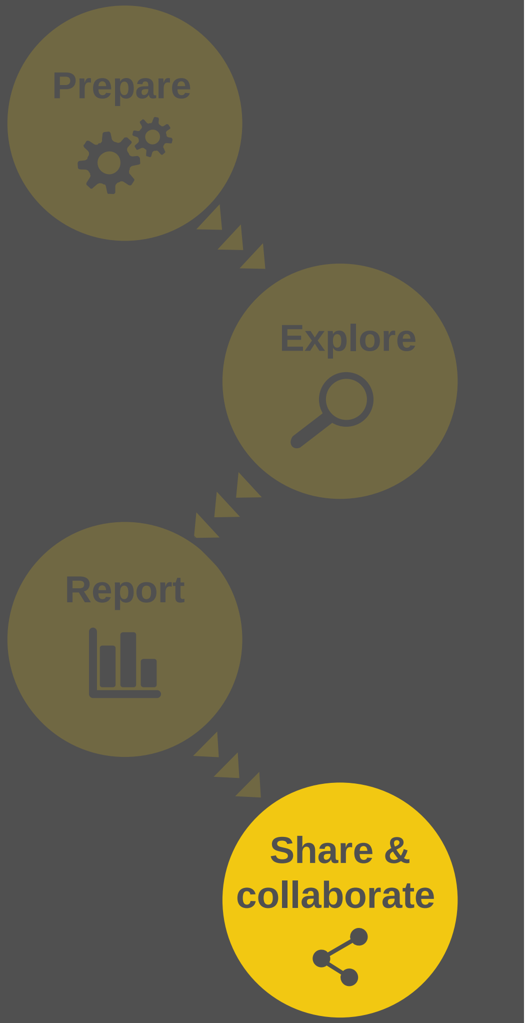
Y-Axis On
X-Axis On
Data Colors
Diverging Off
Minimum
Maximum

☐ CategoryID
☑ CategoryName
☐ Description
☐ Discount
☑ LineTotal
☐ OrderID
☐ ProductID
☐ Quantity
☐ UnitPrice

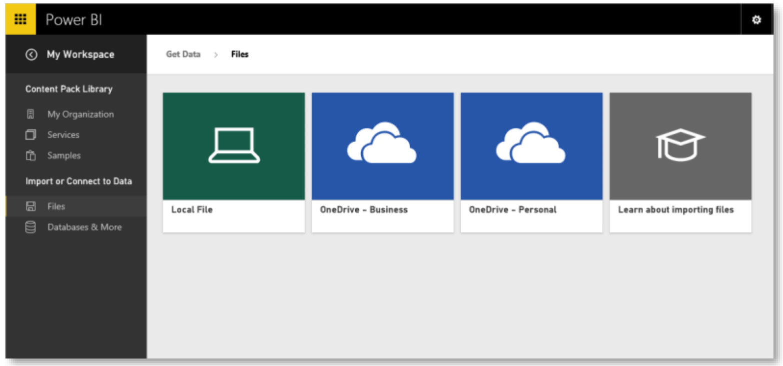
Power BI Desktop allows you to create and customize reports that tell visually compelling data stories

Power BI Desktop

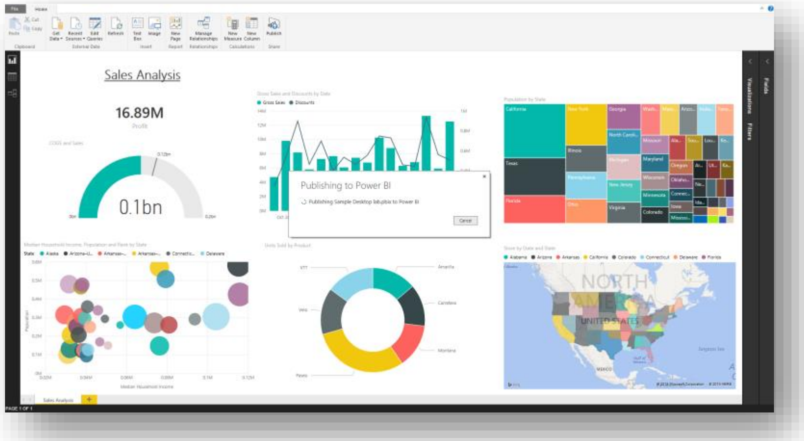
Share your reports and visualizations with a broad audience



- Save Power BI Desktop report files and easily publish them to Power BI
- Share as appropriate with other Power BI users in your organization
- Changes to dashboards automatically sync across all users



Import Power BI Desktop file in Power BI service



Publish from Power BI Desktop to Power BI service

Lab: Create data visualizations by using Azure Databricks and Power BI

- **01-Querying-Files**: This notebook contains some basic visualizations that use built-in Databricks functions and DataFrames.
- **02-Exploratory-Data-Analysis**: In this notebook, you do some basic exploratory analysis on a sample dataset to prepare it for advanced visualization in the next notebooks.
- **03-Power-BI**: In this notebook, you connect your Databricks cluster to Power BI and create visualizations by using Power BI tools.
- **04-Matplotlib**: In this notebook, you use Matplotlib to create custom visualizations for your data.

Questions

- Machine Learning practitioner
- Over 25 years of professional experience
- Artificial Intelligence MVP & MCT
- Microsoft Certified Solutions Expert
  - Data Management and Analytics
  - Cloud Platform and Infrastructure
  - Business Intelligence
- Microsoft Certified Solutions Developer
  - Azure Solution Architect