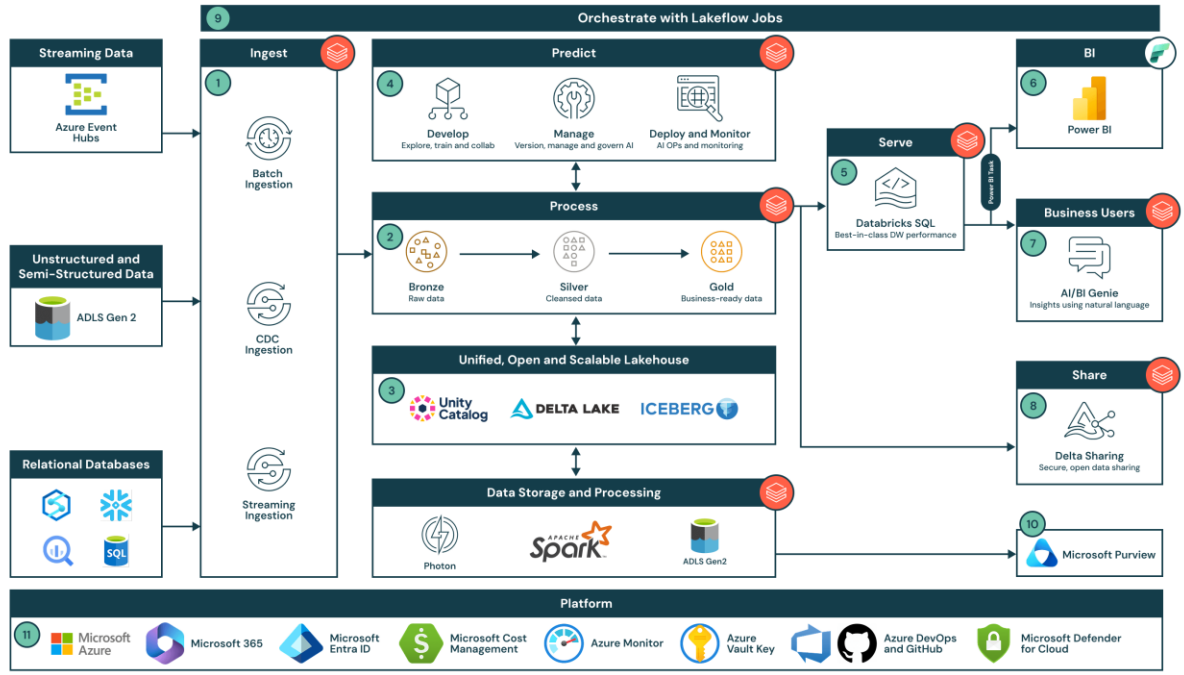
Azure Data Factory (ADF) Integration with Databricks

1. Pipeline Fundamentals

Key Integration Patterns

- **Notebook Execution**: Run Databricks notebooks as pipeline activities
- **Job Cluster Control**: Manage Databricks compute via ADF
- **Parameter Passing**: Dynamic values between ADF and notebooks
- **Error Handling**: Robust failure recovery



ADF-Databricks Integration Architecture

2. Creating Notebook Pipelines

Basic Notebook Activity

json

```
{
  "name": "Execute_Databricks_Notebook",
  "type": "DatabricksNotebook",
  "linkedServiceName": {
    "referenceName": "AzureDatabricks_LS",
    "type": "LinkedServiceReference"
  },
  "typeProperties": {
    "notebookPath": "/Shared/ETL/ProcessRawData",
    "baseParameters": {
      "input_path": "adf_pipeline().parameters.source_path",
      "processing_date": "@{pipeline().TriggerTime}"
    }
  }
}
```

Linked Service Configuration

json

```
{
  "name": "AzureDatabricks_LS",
  "type": "Microsoft.DataFactory/factories/linkedServices",
  "properties": {
    "annotations": [],
    "type": "AzureDatabricks",
    "typeProperties": {
      "domain": "https://adb-1234567890123456.12.azuredatabricks.net",
      "accessToken": {
        "type": "AzureKeyVaultSecret",
        "store": {
          "referenceName": "AzureKeyVault_LS",
          "type": "LinkedServiceReference"
        },
        "secretName": "databricks-token"
      },
      "existingClusterId": "1234-567890-reef123"
    }
  }
}
```

3. Handling Real-World Scenarios

Missing File Handling

json

```
{
  "name": "File_Validation",
  "type": "IfCondition",
  "dependsOn": [
    {
      "activity": "Check_File_Exists",
      "dependencyConditions": ["Succeeded"]
    }
  ],
  "typeProperties": {
    "expression": {
      "value": "@equals(activity('Check_File_Exists').output.exists, true)",
      "type": "Expression"
    },
    "ifFalseActivities": [
      {
        "name": "Send_Alert",
        "type": "Web",
        "typeProperties": {
          "url": "https://hooks.slack.com/services/...",
          "method": "POST",
          "body": {
            "text": "File @{pipeline().parameters.file_path} not found"
          }
        }
      }
    ]
  }
}
```

Retry Logic

json

```
{
  "name": "Databricks_Notebook",
  "type": "DatabricksNotebook",
  "policy": {
    "retry": 3,
    "retryIntervalInSeconds": 30,
    "timeout": "1.00:00:00"
  }
}
```

4. Advanced Pipeline Design

Pipeline Chaining

json

```
{
  "name": "Master_Pipeline",
  "activities": [
    {
      "name": "Execute_Preprocessing",
      "type": "ExecutePipeline",
      "typeProperties": {
        "pipeline": {
          "referenceName": "Preprocess_Data",
          "type": "PipelineReference"
        },
        "waitOnCompletion": true
      }
    },
    {
      "name": "Execute_Main_ETL",
      "type": "ExecutePipeline",
      "dependsOn": [
        {
          "activity": "Execute_Preprocessing",
          "dependencyConditions": ["Succeeded"]
        }
      ]
    }
  ]
}
```

Parameter Passing

python

```
# In Databricks notebook:
input_path = dbutils.widgets.get("input_path")
processing_date = dbutils.widgets.get("processing_date")

# In ADF:
"baseParameters": {
  "input_path": "@pipeline().parameters.source_container",
  "processing_date": "@{formatDateTime(pipeline().TriggerTime, 'yyyy-MM-dd')}"
}
```

5. Scheduling & Monitoring

Trigger Configuration

json

```
{
  "name": "Daily_6AM_Trigger",
  "type": "ScheduleTrigger",
  "typeProperties": {
    "recurrence": {
      "frequency": "Day",
      "interval": 1,
      "startTime": "2023-01-01T06:00:00Z",
      "timeZone": "UTC"
    }
  },
  "pipelines": [
    {
      "pipelineReference": {
        "referenceName": "Daily_ETL_Pipeline",
        "type": "PipelineReference"
      },
      "parameters": {
        "processing_date": "@{formatDateTime(trigger().startTime, 'yyyy-MM-dd')}"
      }
    }
  ]
}
```

Monitoring Setup

1. **ADF Monitoring Hub**:
    - Pipeline runs
    - Activity durations
    - Failure analysis
2. **Databricks Job Alerts**:

python

```
# In notebook:
if error_condition:
  dbutils.notebook.exit("FAILED: Data validation error")

# In ADF:
"activities": [
  {
    "name": "Validate_Output",
    "type": "DatabricksNotebook",
    "dependsOn": [
      {
        "activity": "Execute_ETL",
        "dependencyConditions": ["Succeeded"]
      }
    ]
  }
]
```

6. Performance Optimization

Cluster Configuration

```
{
  "newCluster": {
    "sparkVersion": "10.4.x-scala2.12",
    "nodeTypeId": "Standard_DS3_v2",
    "numWorkers": 4,
    "sparkConf": {
      "spark.sql.shuffle.partitions": "200"
    }
  }
}
```

Parallel Execution

```
{
  "name": "Parallel_Branches",
  "type": "Parallel",
  "activities": [
    {
      "name": "Process_Region_A",
      "type": "DatabricksNotebook",
      "parameters": {
        "region": "A"
      }
    },
    {
      "name": "Process_Region_B",
      "type": "DatabricksNotebook"
    }
  ]
}
```

7. Security Best Practices

Secret Management

json

```
{
  "type": "AzureDatabricks",
  "typeProperties": {
    "accessToken": {
      "type": "AzureKeyVaultSecret",
      "store": {
        "referenceName": "AzureKeyVault_LS",
        "type": "LinkedServiceReference"
      },
      "secretName": "databricks-token"
    }
  }
}
```

Network Security

```
{
  "newCluster": {
    "azureAttributes": {
      "firstOnDemand": 1,
      "availability": "ON_DEMAND_AZURE",
      "subnetId":
"/subscriptions/.../resourceGroups/.../providers/Microsoft.Network/virtualNetworks/
.../subnets/...",
      "vnetResourceGroup": "network-rg"
    }
  }
}
```

8. CI/CD Integration

ARM Template Deployment

json

```
{
  "resources": [
    {
      "type": "Microsoft.DataFactory/factories/pipelines",
      "apiVersion": "2018-06-01",
      "name": "[concat(parameters('factoryName'), '/DailyETL')]",
      "properties": {
        "activities": [
          {
            "name": "Databricks_Notebook",
            "type": "DatabricksNotebook",
            "linkedServiceName": {
              "referenceName": "AzureDatabricks_LS",
              "type": "LinkedServiceReference"
            }
          }
        ]
      }
    }
  ]
}
```

9. Troubleshooting Guide

Issue	**Solution**
Authentication failures	Verify token permissions and expiry
Cluster startup delays	Use instance pools or warm clusters
Parameter passing issues	Check widget names in notebook
Timeout errors	Increase timeout or optimize notebook

10. Complete Example

End-to-End ETL Pipeline

json

```
{
  "name": "Daily_Sales_ETL",
  "properties": {
    "activities": [
      {
        "name": "Check_Source_Files",
        "type": "GetMetadata",
        "dependsOn": [],
        "policy": {
          "retry": 2
        },
        "typeProperties": {
```

```
          "dataset": {
            "referenceName": "SourceDataStore",
            "type": "DatasetReference"
          },
          "fieldList": ["exists"]
        }
      },
      {
        "name": "Execute_Databricks_ETL",
        "type": "DatabricksNotebook",
        "dependsOn": [
          {
            "activity": "Check_Source_Files",
            "dependencyConditions": ["Succeeded"]
          }
        ],
        "typeProperties": {
          "notebookPath": "/Shared/ETL/SalesPipeline",
          "baseParameters": {
            "processing_date": "@{pipeline().parameters.execution_date}"
          }
        }
      },
      {
        "name": "Send_Success_Alert",
        "type": "Web",
        "dependsOn": [
          {
            "activity": "Execute_Databricks_ETL",
            "dependencyConditions": ["Succeeded"]
          }
        ],
        "typeProperties": {
          "url": "https://hooks.slack.com/services/...",
          "method": "POST"
        }
      }
    ],
    "parameters": {
      "execution_date": {
        "type": "String",
        "defaultValue": "@{pipeline().TriggerTime}"
      }
    }
  }
}
```