**Azure Monitor, Alerts, Logs & Playbooks**

**Lab 1: Azure Monitor Alerts – Create an Alert for High Pipeline Failure Rate**

**Objective**

Set up an alert in Azure Monitor to notify when Azure Data Factory (ADF) pipelines fail above a threshold.

**Steps**

1. **Navigate to Azure Portal**
   * Go to your **Azure Data Factory resource**.
   * Under **Monitoring**, select **Alerts**.
2. **Create New Alert Rule**
   * Click **+ Create alert rule**.
   * Select the **ADF resource** as the scope.
3. **Define Condition**
   * Choose **Condition → Add Condition**.
   * Metric: Pipeline failed runs.
   * Operator: Greater than.
   * Threshold: 5 (or desired failure count).
   * Aggregation: Count.
4. **Add Action Group**
   * Click **Add action group**.
   * Provide **name**, **short name**.
   * Notification type: **Email/SMS/Teams/Logic App**.
5. **Review and Create**
   * Review settings → Create.

**Result:** You now have an active alert for ADF pipeline failures.

**Lab 2: Log Analytics – Query Logs for Failed Jobs in Databricks**

**Objective**

Use **Log Analytics** to query failed Databricks jobs.

**Steps**

1. **Enable Diagnostic Settings in Databricks**
   * Go to **Databricks workspace → Diagnostic settings**.
   * Send logs to **Log Analytics workspace**.
2. **Open Log Analytics Workspace**
   * In Azure Portal, navigate to **Log Analytics Workspace**.
3. **Run Kusto Query**
   * In Logs, run the query:
   * AzureDiagnostics
   * | where ResourceType == "DATAPROCS"
   * | where Status\_s == "Failed"
   * | project TimeGenerated, Resource, OperationName, Status\_s, FailureReason
   * | order by TimeGenerated desc
   * Modify filters for job runs, cluster logs, or notebook errors.
4. **Visualize Results**
   * Save the query as a workbook for dashboards.

**Result:** You can now track failed jobs directly via queries.

**Lab 3: Alert Testing – Trigger and Respond to an Alert Scenario**

**Objective**

Simulate and validate alert functionality.

**Steps**

1. **Trigger Failure in ADF**
   * Intentionally misconfigure a dataset or linked service.
   * Run the pipeline → it should fail.
2. **Monitor Alert Trigger**
   * Go to **Azure Monitor → Alerts**.
   * Check if your **High Pipeline Failure alert** is triggered.
3. **Respond to Alert**
   * Check **email/Teams notification**.
   * Acknowledge and document the incident.
4. **Remediate**
   * Fix the misconfiguration in ADF.
   * Re-run pipeline → confirm it succeeds.

**Result:** You validated alert detection and response workflow.

**Lab 4: Playbook Creation – Troubleshooting Workflow**

**Objective**

Create a **standardized troubleshooting playbook** for operational incidents.

**Steps**

1. **Identify Common Issues**
   * Pipeline failures.
   * Databricks job crashes.
   * Cluster scaling issues.
2. **Define Troubleshooting Workflow**  
   Example: ADF Pipeline Failure.
   * Step 1: Check pipeline logs in ADF Monitor.
   * Step 2: Query errors in Log Analytics.
   * Step 3: Validate linked service and credentials.
   * Step 4: Retry pipeline.
   * Step 5: Escalate if recurring (assign to DevOps team).
3. **Document in Azure**
   * Use **Azure DevOps Wiki, Confluence, or SharePoint**.
   * Include screenshots, queries, and escalation matrix.
4. **Automate Remediation (Optional)**
   * Link Azure Monitor Alert to **Logic App/Automation Runbook**.
   * Example: Auto-restart a failed Databricks cluster.

**Result:** You now have a reusable troubleshooting playbook for production issues.