COLLEGE OF CREATIVE INNOVATION

Programme

Diploma in Cloud Computing and Cyber Security
(120 Credits)

Course

CCC602: Data Engineering in the Public Cloud
(Level 6, 30 Credits, Version 1.1)

Assessment Title

# Azure Practical Scenario Evaluation

## CCC602 | Assessment-3

Weighting within the course

**60%**

## Objective:

This assessment aligns with the **Course Learning Outcomes (LO1 and LO2)** and **Graduate Profile Outcomes (GPO2 and GPO8)**. You will showcase your ability to deploy infrastructure that supports organizational goals for billing and cost optimization, evaluate and implement cloud services, and configure resilient and secure data environments.

By completing this project, you will:

- Utilize Infrastructure as Code to deploy and manage Azure resources efficiently.
- Automate data pipelines and analytical workflows using Azure Synapse, Data Lake, and Databricks.
- Analyze and visualize large-scale datasets to inform business insights.
- Apply cloud best practices to ensure system resilience, security, and cost control.

This project reflects real-world industry practices and prepares you to support data-driven decision-making in a modern cloud environment.

## Course Learning Outcomes (LOs) covered:

**LO1:** Deploy cloud services, utilizing infrastructure as code, to align with organizational requirements for billing and cost management.

**LO2:** Evaluate service options from different cloud vendors and implement the appropriate option to automate IT infrastructure.

## Qualification Graduate Profile Outcomes (GPOs) covered:

**GPO2:** Implement, configure, and deploy a range of services in the cloud, including infrastructure as code, billing, and cost management, to meet organisational requirements.

**GPO8:** Implement and monitor a range of resilient and secure systems and network services, including identity management and directory services, to meet organisational requirements.

## Assessment Tasks to Learning Outcome and GPOs Mapping:

| LO | GPO | Task | Task Component | Weighting |
|---|---|---|---|---|
| LO 2 | GPO2 | **Task 1:** Cloud Vendor Evaluation. | Task 1: (Theoretical Evaluation) | 15% |
| LO 1 | GPO2 and GPO8 | **Task 2:** Dataset Acquisition and Azure Resource Deployment using IaC | Task 2: (Practical Implementation) | 20% |
| LO1 and LO2 | GPO 2 and GPO 8 | **Task 3:** Data Ingestion Using Azure Data Factory. | Task 3: (Practical Implementation) | 25% |
| LO1 and LO2 | GPO 2 and GPO 8 | **Task 4:** Create an Azure Databricks Workspace and perform data transformation using code to automate the process. | Task 4: (Practical Implementation) | 25% |
| LO1 and LO2 | GPO 2 and GPO 8 | **Task 5:**Create Azure Synapse Analytics and Automate Data Integration | Task 5: (Data Analysis and Report Generation) | 15% |
| | | | | |
| | | | Total | 100% |

## Recommended Tasks Completion Timeline:

| Full Time Week | Part Time Week | Progress | Submission |
|---|---|---|---|
| Week 9 | Week 17,18 | Start working on the Assessment | |
| Week 10 | Week 19,20 | Complete Task 1 and Task 2 | |
| Week 11 | Week 21,22 | Complete Task 3 and Task 4 | |
| Week 12 | Week 23,24 | Complete Task 5 and submit | Assessment due by Week 12 (Full Time)<br>Assessment due by Week 24 (Part Time) |

## Grading:

The final grade will be determined by the score achieved in this assessment based on the following table. Should a second or third attempt be required, the maximum contribution toward the overall mark for the tasks that required a second or third assessment attempt is 50%. **A late submission is considered a second attempt, so the contribution will be capped at 50%.**

**To pass this assessment, you must meet the requirements of each of the learning outcomes (irrespective of the numerical grade awarded).**

| Grade | Range |
|---|---|
| **A** | Meet all course requirements, range (80—100%) |
| **B** | Meet all course requirements, range (65—79%) |
| **C** | Meet all course requirements, range (50—64%) |
| **D** | Did not meet all course requirements, range (40—49%) |
| **E** | Did not meet all course requirements, mark range (0—39%) |

## Candidate's Assessment Instructions:

- This assessment is an **open-book activity**; you can use your course and review notes, and offline or online resources, such as textbooks or online journals.
- You can always ask your online tutor if you need further explanation if the instructions are unclear.
- Your work should not be plagiarised. Plagiarism includes copying material without acknowledging it, copying from another student, getting another person to help you with your assessment, using material from commercial essays or assignment services, or using AI to create the answers.
- The purpose of this assessment is to assess your knowledge. In the event Yoobee suspects collusion, this will be addressed. For more information on plagiarism, please refer to the Student Handbook.
- Submit your completed assessment online in the correct space provided.
- Marks and feedback will be returned within 15 days of the submission date.
- By completing and submitting an assessment, you are authenticating that the work is original and does not violate plagiarism or copyright law. Authenticity is checked where any breaches of academic integrity are suspected. Please refer to the Student Handbook for further information.

## Submission Instructions:

Submit **one PDF report** document to the LMS by the specified due date.

Your report should:

- Include your name and ID number
- Include the Azure account login details, a cover page, and a report index for verification purposes in your report.
- Use a standard citation format if external sources are referenced
- Clearly label tasks and subtasks, and Diagrams must be clear and labeled properly.
- Include screenshots of each practical step in sequence, naming and numbering the screenshots. Screenshots must display the relevant settings or outputs for each step.
- Include your answers to the assessment questions for each task, describing choices, configurations, and learned insights with an appropriate practical and theoretical understanding.
- **Submission must be in PDF format only because other formats may cause issues with accessing screenshots.**

# Assessment Tasks: End-to-End Azure Data Engineering Project.

## Scenario: Tokyo Olympic Data Analyst Project.

You recently joined Yoobee as a Junior Data Engineer Intern. The organization is adopting a Microsoft Azure cloud-first strategy to build a scalable, automated, and cost-effective data analytics solution. As part of this initiative, the organization will analyse the **Tokyo Olympic dataset** to gain insights into athlete performance, medal tallies, and country-wise statistics.

To support automation and cost control, Yoobee is emphasizing the use of **Infrastructure as Code (IaC)** for deploying Azure resources. Your role as a student Data Engineer involves both **technical implementation** and **strategic evaluation of cloud services**.

**Note:**

Your Tutor has provided a GitHub repository at link below for your reference including ARM template, datasets, codes, SQL commands and Notebook pyspark code for your reference.

GitHub Repository link to access datasets and required files:

https://github.com/yoobeecloud/Assessment_3_Azure_Project_Codes.git

**Delete all resources after completing the Assessment, including the main resource group, to avoid any billing on your Azure account.**

## Task 1: Cloud Vendor Evaluation (Theoretical Evaluation).

As a Junior Data Engineer Intern at Yoobee, your first task is to evaluate and recommend the most suitable cloud platform to support a scalable, automated, and cost-effective data analytics solution.

### 1.1 Comparative Evaluation: Microsoft Azure vs. Amazon Web Services (AWS)

Conduct a detailed comparison between **Microsoft Azure** and **Amazon Web Services (AWS)** with a focus on the following key criteria:

- **Big Data and Analytics Capabilities**: Assess the range and maturity of services offered for data ingestion, processing, storage, and analysis.
- **Integration with Business Intelligence (BI) Tools**: Evaluate native compatibility with any one of the tools, such as Power BI, Tableau, Looker Studio, etc.
- **Cost Structures and Pricing Tiers**: Compare pricing models for storage (e.g., Blob vs. S3) and compute (e.g., Azure Databricks vs. AWS EMR), and identify cost-saving opportunities.
- **Scalability and Automation**: Examine how each platform, like Microsoft Azure and AWS, supports scaling workloads and automating infrastructure deployment through tools like ARM templates and AWS CloudFormation.

## Deliverables:

- **Cost Analysis Report:**
  Prepare a comparative report that includes pricing details for equivalent storage and compute services (e.g., Azure Data Lake Gen2 vs. Amazon S3, Azure Databricks vs. AWS EMR).
- **Design Justification**:
  Describe how your recommended design enables:
  - Scalable and flexible infrastructure deployment
  - Cost transparency and billing governance
  - Automated provisioning through Infrastructure as Code (IaC)
- **Vendor Selection Report**:
  Write a formal report (maximum 500 words, +/- 10%) justifying the selection of **Microsoft Azure** as the preferred cloud platform. Your justification should consider Yoobee's strategic focus on data analytics and alignment with the Microsoft ecosystem.

## Task 2: Dataset Acquisition and Azure Resource Deployment using IaC

As part of Yoobee's Azure-based data analytics solution, your next task is to acquire the required dataset and deploy the foundational Azure resources using IaC (Infrastructure as a Code) to ensure repeatability, scalability, and cost control.

### 2.1 Azure Resource Deployment Using ARM Template

Provision core Azure infrastructure using an Azure Resource Manager (ARM) template:

- Create a Resource Group **(Your Firstname_Cloud_RG)** and a Storage Account **(yourfirstnamecloudstorage)** in your preferred Azure region (e.g., Australia Central, East US).
- The Storage Account must be of type StorageV2 and have Hierarchical Namespace enabled to support Data Lake Storage Gen2 features.

Ensure your ARM template includes the following configurations:

- Resource Group Name = **(Your Firstname_Cloud_RG)**
- Storage Account Name = **(yourfirstnamecloudstorage)**
- Azure Region
- Advanced Settings such as:
    - Blob Storage configuration
    - Hierarchical Namespace enabled
    - Replication type (e.g., LRS-Locally Redundant Storage or GRS-Geo-Redundant Storage)

Apply Yoobee's cost governance and tagging policies:

- Add resource tags **(e.g., Department: Data, Project: Olympic2021)** for cost tracking.
- Select a region that offers optimal cost and performance.
- Configure replication settings that balance durability and budget.

## Deliverables:

Take step-by-step screenshots of your deployment process, as illustrated in the example shown in **Appendix 2 Task-2,** including:

- ARM template code **(Download the Template code for submission)**
- Deployment in the Azure portal **(Take a screenshot of the storage account and resource group deployment)**
- Confirmation of successful resource creation

Also, include a documented breakdown of the ARM template structure and parameters used.

### 2.2 Tokyo Olympic Dataset Acquisition

- Access the Tokyo Olympic dataset from the GitHub link provided by your tutor.
- Download or clone the repository to your local environment or Azure storage.
- Review and document:
    - The file formats (e.g., XLS)
    - Key fields and schema present in each file
    - Any relevant metadata or folder structures

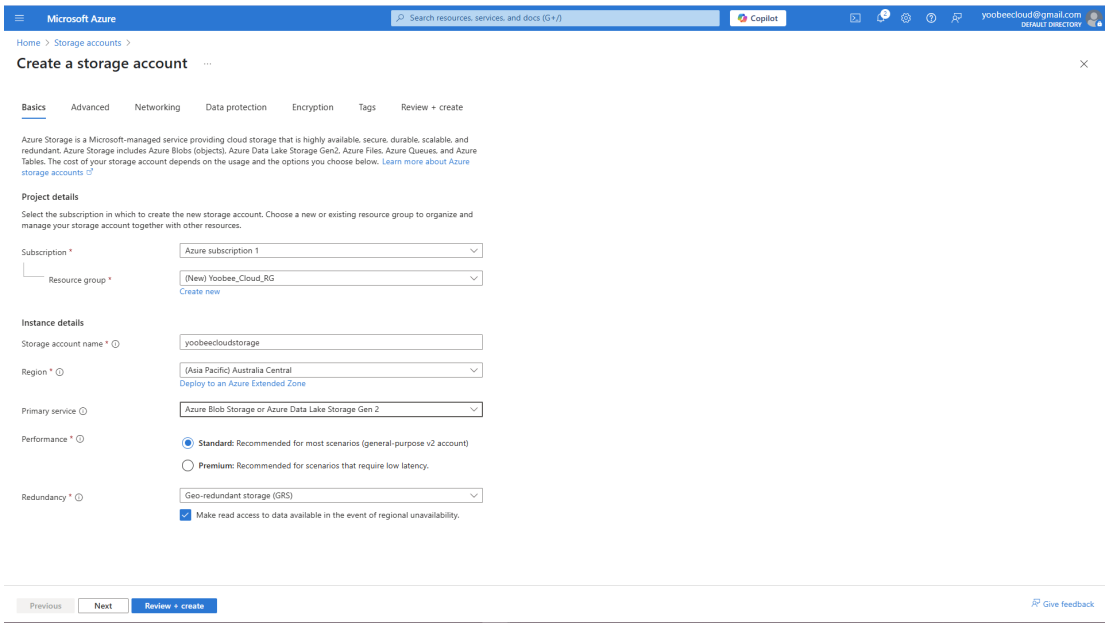
## Deliverables:

Provide a short **data summary report**, listing:

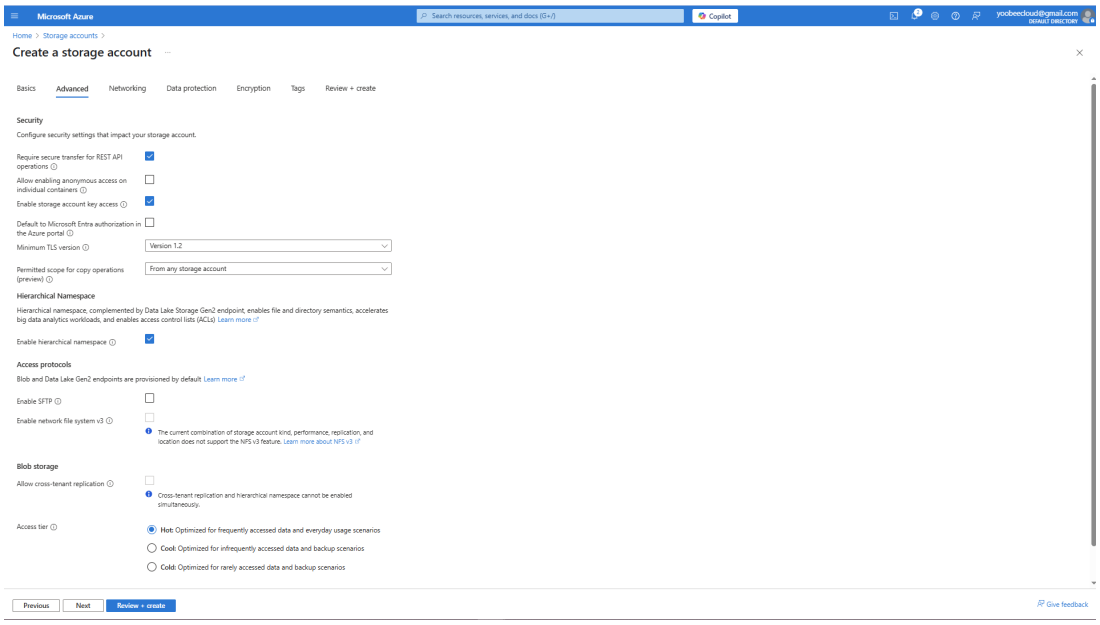
- File types and names
- Schema or key fields
- Observations about the structure or content

# Appendix: Task-2 Screen Shots

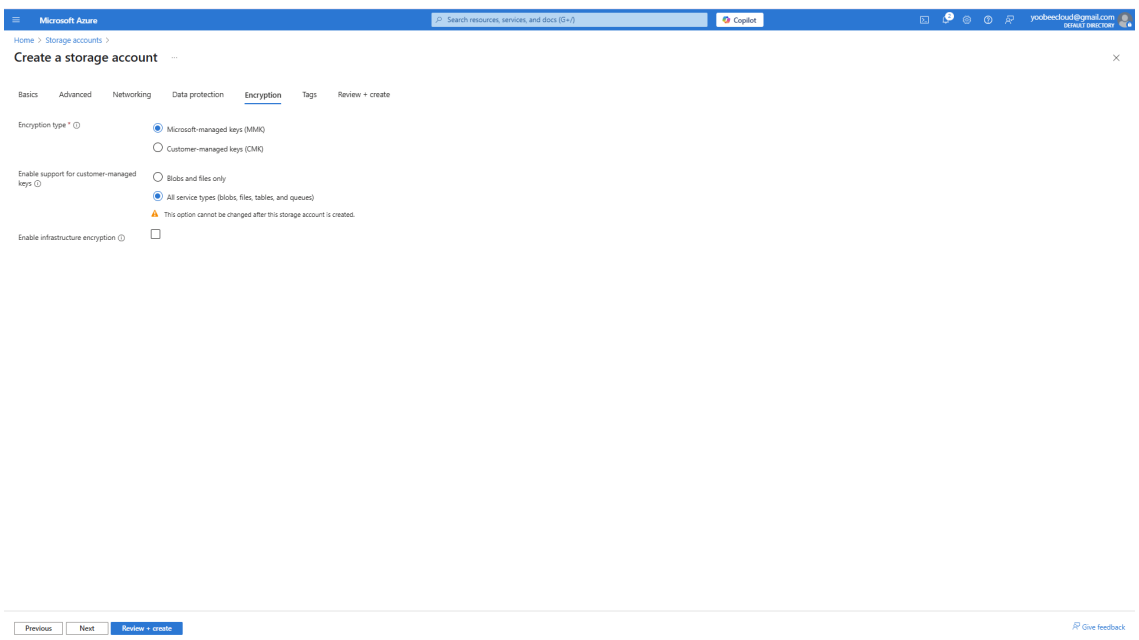
## Screenshot: 2.1: Create Azure Storage Account




## Screenshot: 2.2




## Screenshot: 2.3



## Task 3: Data Ingestion Using Azure Data Factory.

**3.1 Design and implement a data ingestion pipeline using Azure Data Factory to automate the movement of data into your storage environment.**

- **Source Configuration**: Set up the source to extract data from **GitHub**, or **local files,** or **Azure Blob Storage** if the dataset has been staged there.
- **Sink Configuration**: Configure the sink to write data into **Azure Data Lake Storage Gen2**, specifically into the **/raw/** folder to maintain the raw state of the ingested data.
- **Pipeline Validation and Testing**: Use the **Validate** and **Debug** options in Azure Data Factory to test the pipeline execution.
- **Verification**: After successful execution, verify the data ingestion by checking the target location in the Storage Account to confirm that the files have been uploaded correctly.

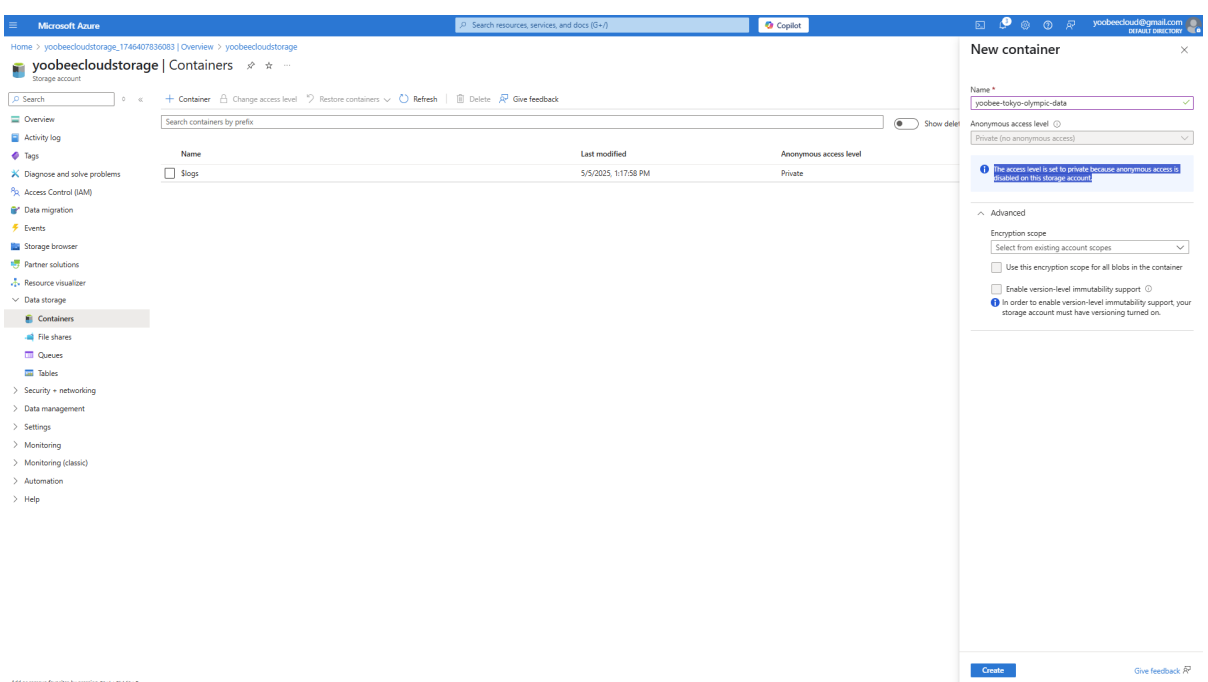
## Deliverables: Screenshot

Take step-by-step screenshots of your deployment process, as illustrated in the example shown in **Appendix Task-3,** including:

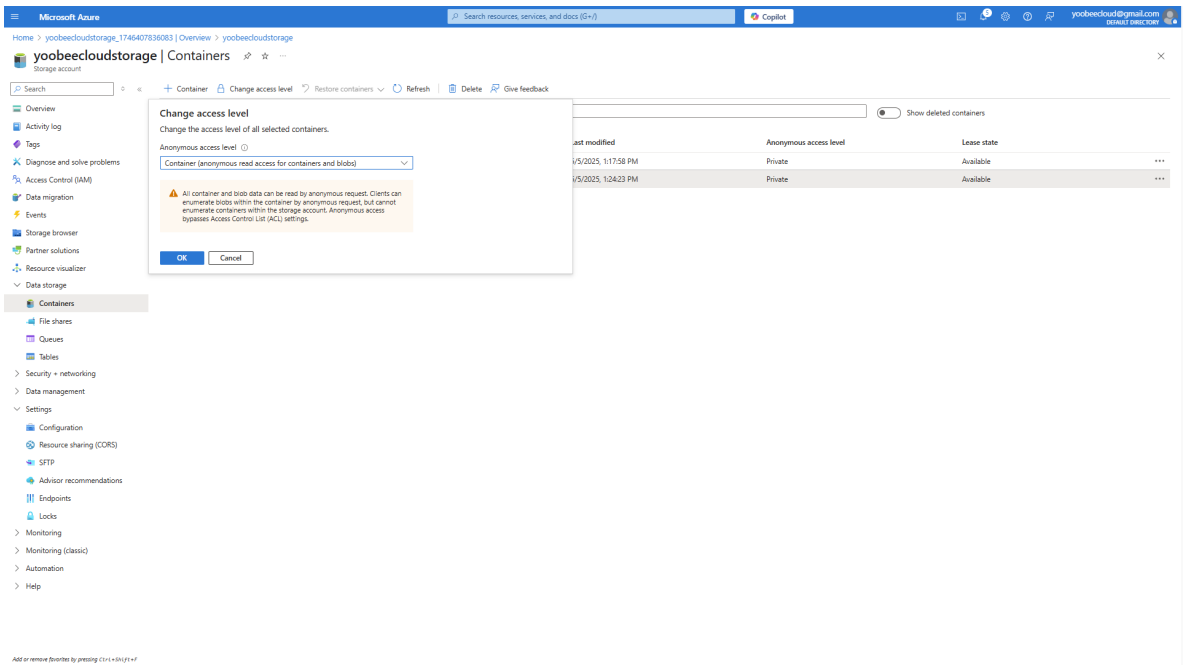
- **Screenshot of Azure Data Factory Pipeline Design**: Visual layout of the pipeline with source and sink datasets configured.
- **Source Configuration Details**: Document the source settings (GitHub/local/Blob Storage), including:
  - Dataset path or URL
  - File format (e.g., CSV, JSON)
  - Linked service configuration
- **Sink Configuration Documentation**
  - Show the setup for writing to Azure Data Lake Storage Gen2 in the /raw/ directory.
  - Include the linked service and dataset configuration used for the sink.
- **Pipeline Execution Proof**
  - Screenshots of successful pipeline validation, debug run, and execution status from the Monitor tab in Azure Data Factory.
- **Target Storage Verification**
  - Screenshot of the /raw/ folder in Azure Storage Explorer or the Azure Portal, confirming that the dataset was ingested and stored successfully.
- **Brief Pipeline Summary Report:** A short explanation **(approx. 500 words)** describing:
  - The purpose of the pipeline
  - The data flow (source to sink)
  - Any challenges or observations during implementation

## Appendix: Task-3 Screen Shots

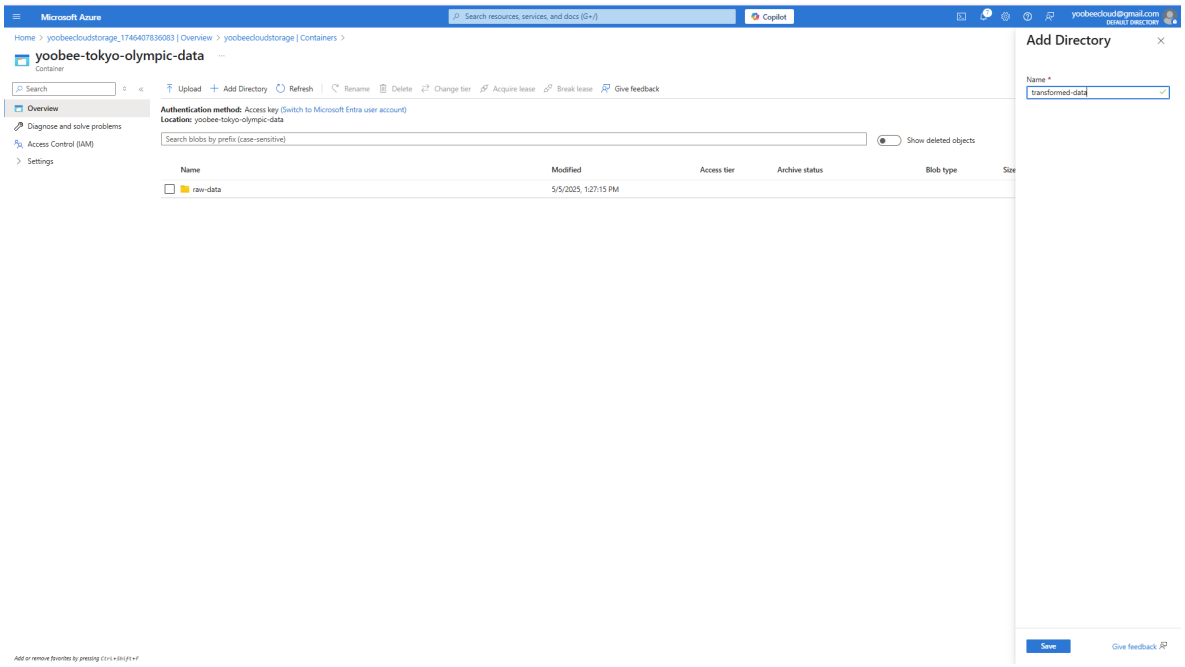
**Screenshot: 3.1**



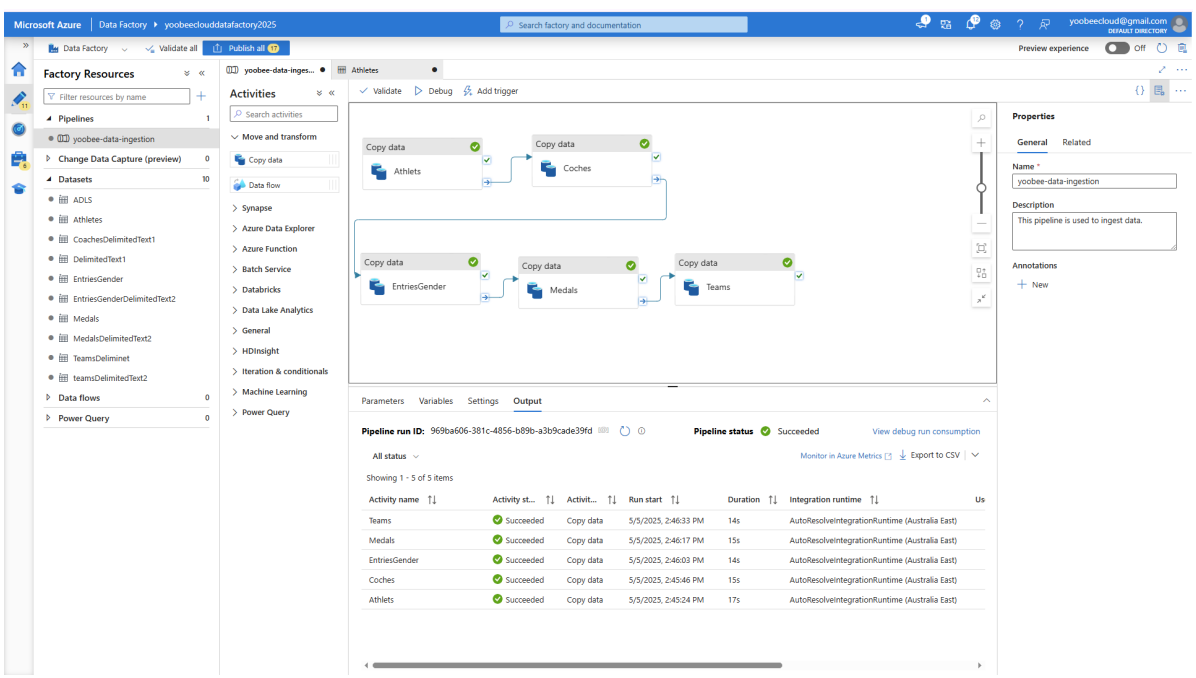
## Screenshot: 3.2




## Screenshot: 3.3




## Screenshot: 3.4



## Task 4: Create an Azure Databricks Workspace and perform data transformation using code to automate the process.

**4.1 Create the Azure Databricks Workspace:**

- Set up an Azure Databricks workspace named **Firstname-Lastname-db**.
- Create this workspace in the **same resource group and region** used in **Task 2**.
- Select the **Premium pricing tier** under your Azure subscription.

**4.2 Create a Single-Node Cluster:**

- Launch a **single-node compute cluster** in your preferred Databricks runtime environment.
- This cluster will be used to **transform data** using a Databricks notebook.

**4.3 Data Integration and Authentication:**

- Transfer data **from Azure Data Lake Gen2 to Databricks** and then write it back to the **transform-data** folder in Data Lake Gen2.
- Register an **Azure App** named **first-name-app** to authenticate the connection between Databricks and Data Lake Gen2.
- Use the **Client ID**, **Client Secret**, and **Tenant ID** from the app registration for secure access.

**4.4 Notebook and Code Execution:**

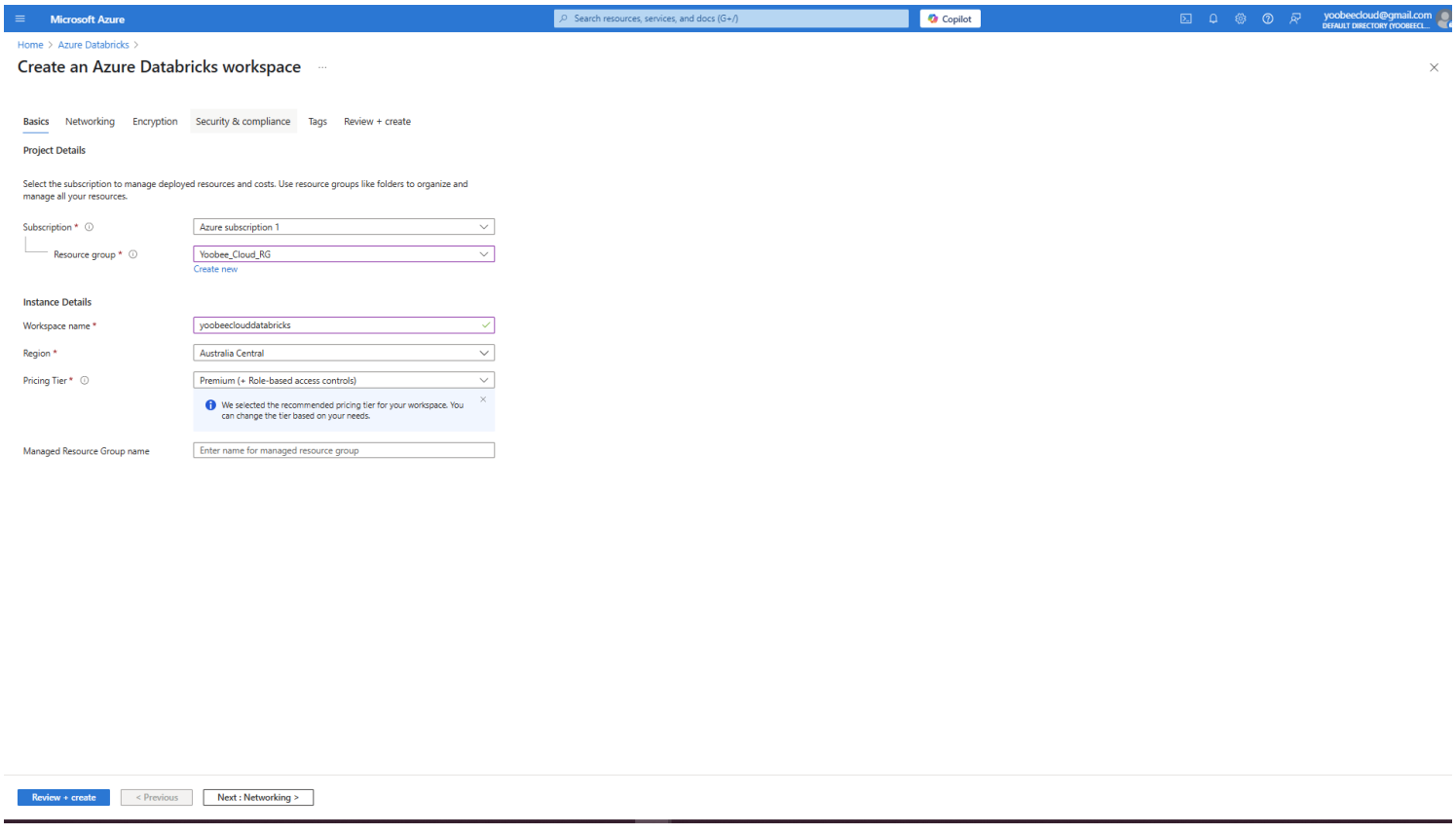
- Write and modify appropriate **PySpark (Python Spark)** code as provided by your tutor.
- Generate a **mount point** to your storage container for access within Databricks.
- Assign the appropriate **Azure Role-Based Access Control (RBAC)** to enable secure access to the **transform** folder and its datasets in Data Lake Gen2.

## Deliverables: Screenshot

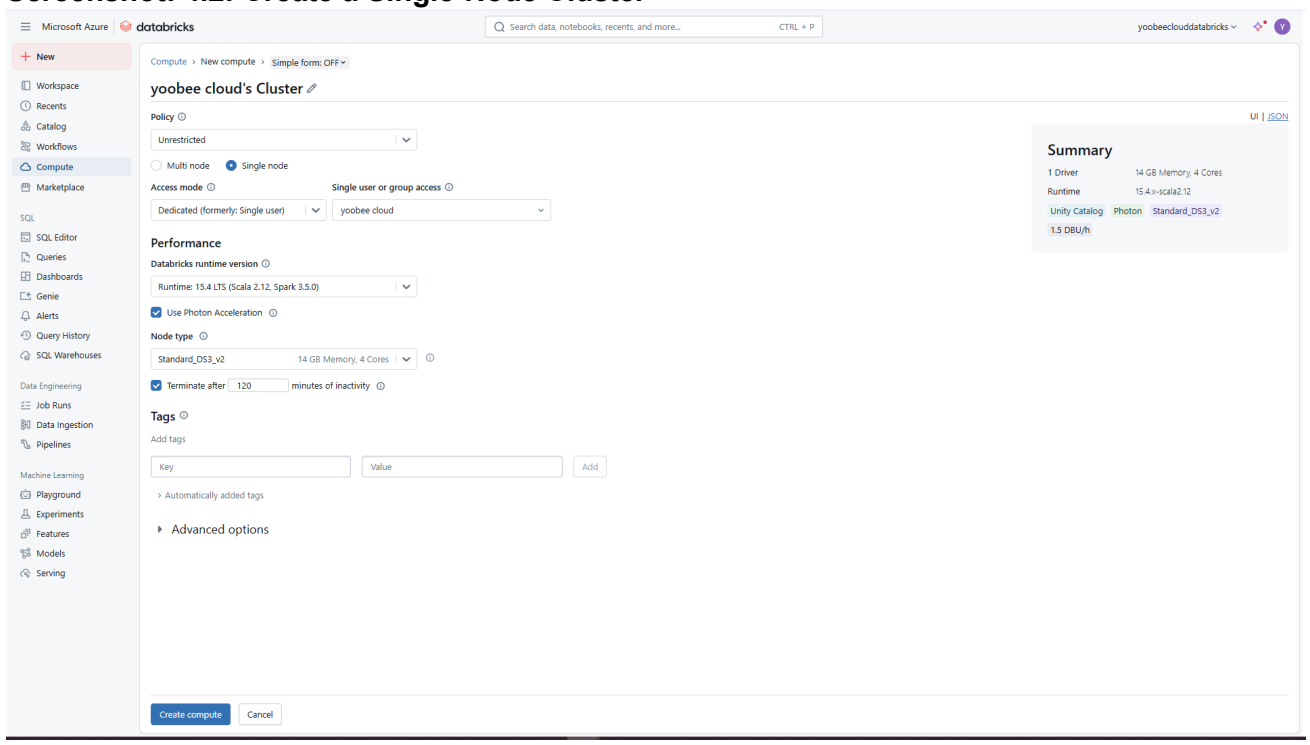
- Provide **step-by-step screenshots** as illustrated in the example shown in **Appendix Task 4**.
- Include the **complete data transformation code** used in the Databricks notebook.

# Appendix: Task-4 Screen Shots

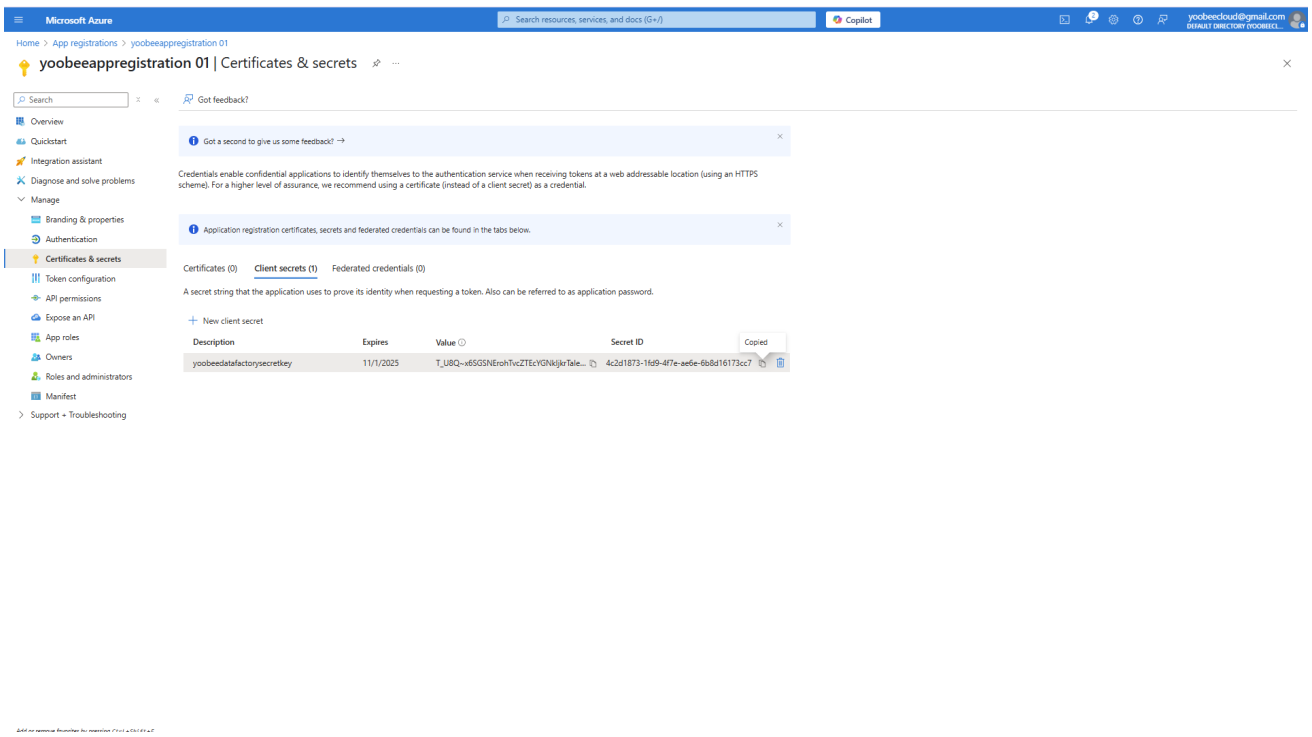
### Screenshot: 4.1: Create Azure Databricks



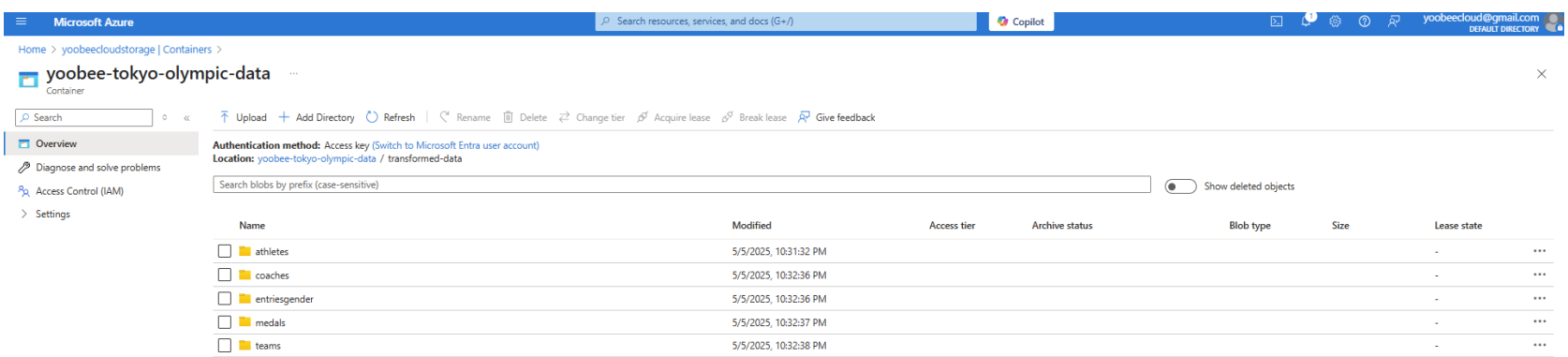
## Screenshot: 4.2: Create a Single-Node Cluster




## 4.3 Data Integration and Authentication:




## Screenshot: 4.4 Notebook and Code Execution:



## Task 5: Create Azure Synapse Analytics and Automate Data Integration.

### 5.1 Create Azure Synapse Analytics Workspace

- Set up an **Azure Synapse Analytics** workspace named **Firstname_Lastname_Yoobee**.
- Ensure the workspace is created in the **same resource group and region** used in **Task 2**.
- Link your **Azure Data Lake Storage Gen2** account to enable access to dataset files.

### 5.2 Build Azure Data Pipeline

- Design and develop an **Azure Data Factory pipeline** (or use a **Synapse pipeline**) to automate the **data transfer** process from **Azure Data Lake Gen2** to **Azure Synapse Analytics**.
- Ensure the pipeline executes successfully and transfers the **Tokyo Olympics dataset** into the Synapse workspace.

### 5.3. Create External Tables using SQL commands

- Within the Synapse workspace, use **SQL scripts** to create **external tables** that reference your **Tokyo Olympics dataset**.
- Use these tables to import and organize the data from the **transport folder** in Data Lake Gen2.

### 5.4. Perform Data Analysis

Write and execute **basic SQL queries** in Synapse Studio to analyze the dataset and capture the results:

1. Count the number of athletes from each country.
2. Calculate the total medals won by each country.
3. Calculate the average number of entries by gender for each discipline.

### 5.5. Visualize the Results

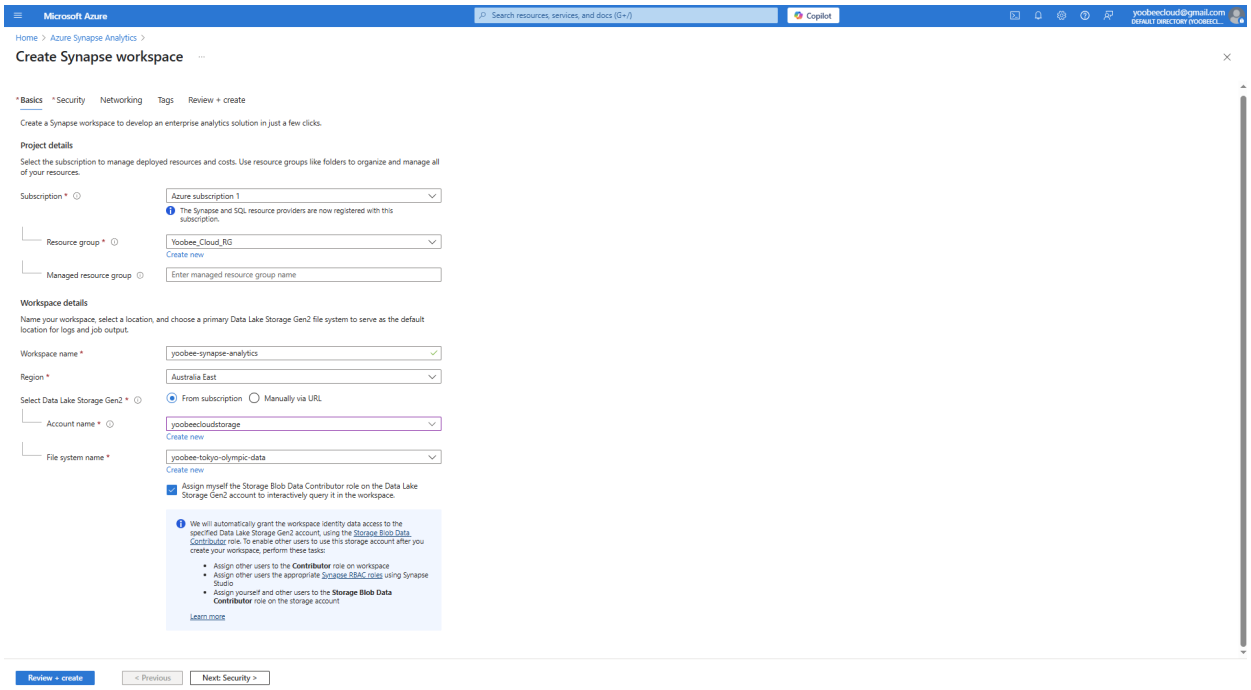
- Capture and document the **visual output** as part of your analysis within **Azure Synapse Analytics**.

## Deliverables: Screenshot

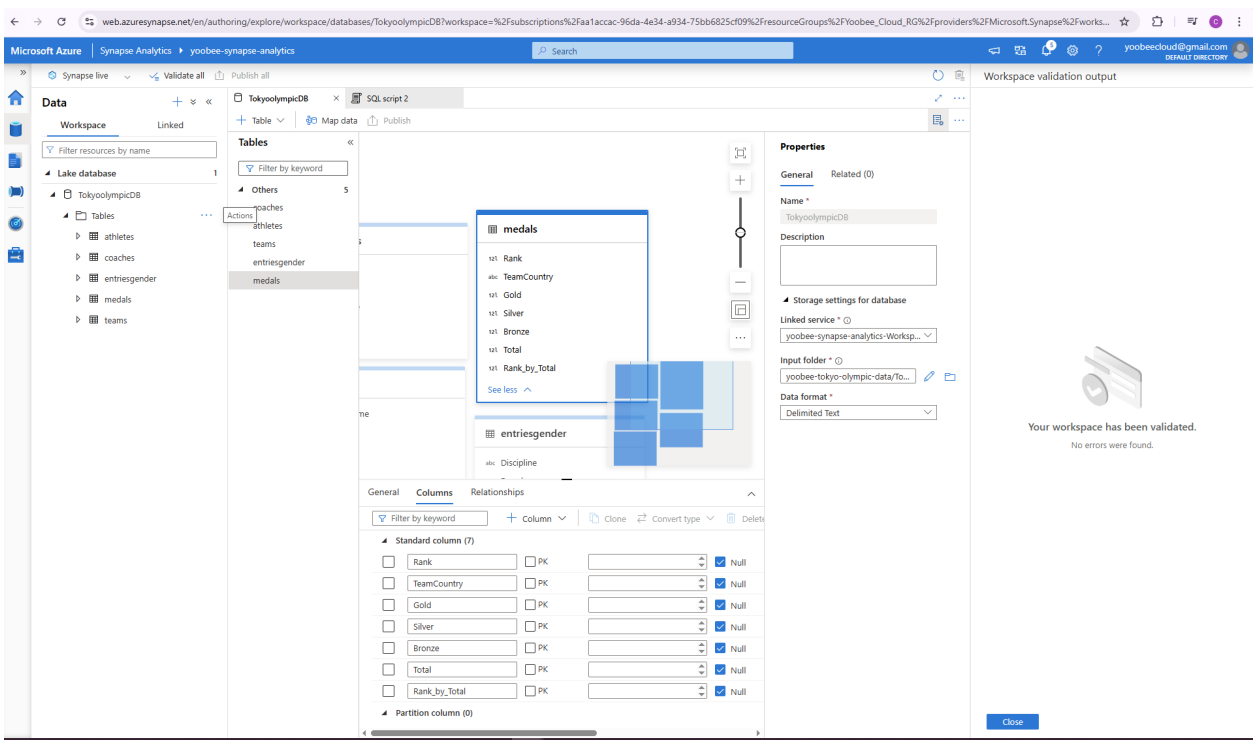
- Provide **step-by-step screenshots** as illustrated in the example shown in **Appendix Task 5**.
- Screenshot of table creation and data preview within Synapse Studio.
- Screenshots of the query outputs/results in Synapse Studio.
- Screenshots of charts/visualizations based on query results (e.g., bar charts, pie charts).

## Appendix: Task-5 Screen Shots

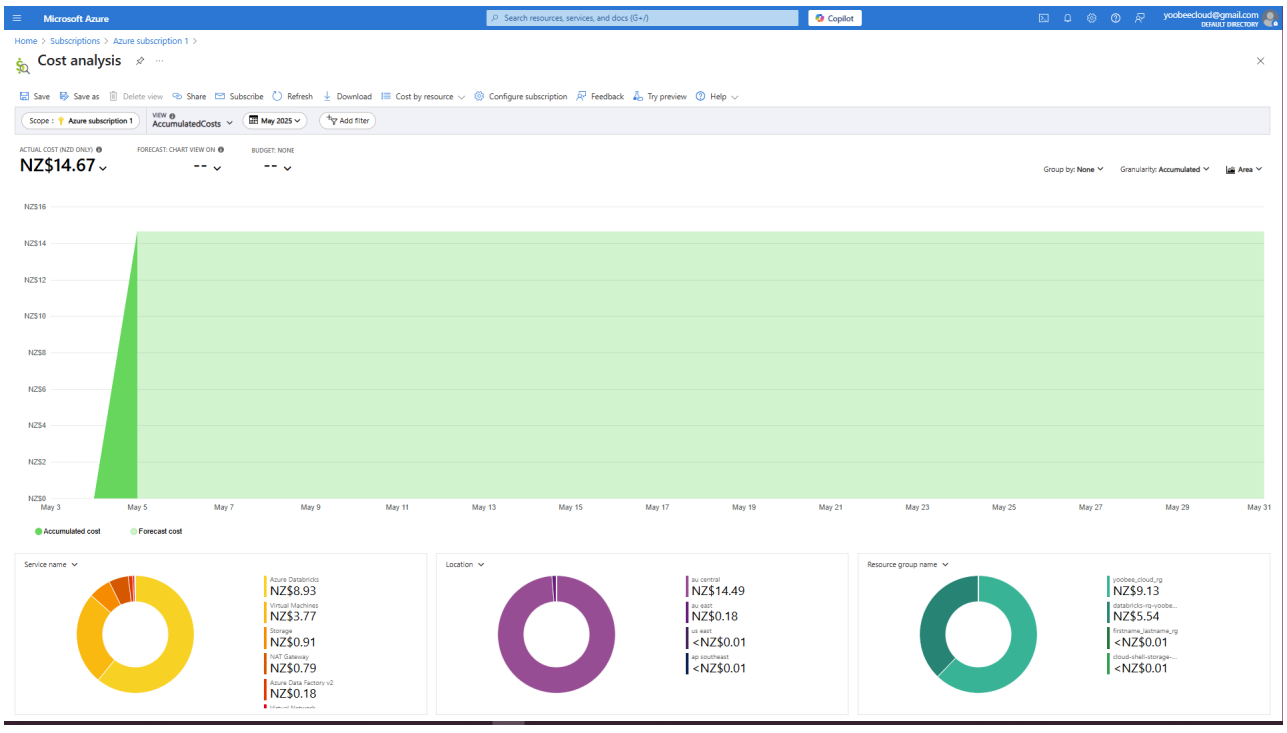
### 5.1 Create Azure Synapse Analytics Workspace




## 5.2 Build Azure Data Pipeline




## 5.3. Visualize the Results



# Marking Rubric

**To pass this assessment, you must meet the requirements of each of the learning outcomes (irrespective of the numerical grade awarded).**

| Criterion | | Evidence | | | | |
|---|---|---|---|---|---|---|
| **Task and Weightage** | | **A (80-100%)** | **B (65-79%)** | **C (50-64%)** | **D (40-49%)** | **E (0-39%** |
| **Task 1:**<br>**Cloud Vendor Evaluation (Theoretical Evaluation).**<br>**(LO2)** | **1.1 Comparative Evaluation: Azure vs. AWS (15%)** | Demonstrates outstanding understanding and application of cloud concepts. Provides a thorough, well-structured comparison of Azure vs. AWS, including accurate cost analysis, clear and strategic design justification, and a persuasive, well-written vendor selection report. All components are complete, insightful, and aligned with the scenario. | Shows solid grasp of cloud services and delivers a coherent comparison with mostly accurate details. Cost analysis and justification are well-presented, though they may lack depth in a few areas. The vendor selection report is relevant and clear, with minor lapses in structure or clarity. All deliverables are mostly complete. | Meets basic requirements of the task. Comparison covers essential points but lacks detailed insight. Cost analysis may be limited or partially inaccurate.<br><br>Justification and report are present but may be vague or underdeveloped. Some components may be missing minor details or clarity. | Demonstrates minimal understanding of cloud evaluation. Comparison and cost analysis are incomplete or inaccurate. Design justification lacks clarity or relevance.<br><br>The vendor report is poorly structured or unconvincing. Several required elements are missing or insufficiently addressed. | Fails to address key elements of the task. The response is largely incomplete, lacks coherence, and demonstrates little to no understanding of cloud services or the scenario.<br><br>Major components are missing, incorrect, or irrelevant. |
| **Task 2:**<br>**Dataset Acquisition and Azure Resource Deployment using IaC.**<br>**(LO1)** | **2.1** Azure Resource Deployment Using ARM Template.<br><br>**2.2** Tokyo Olympic Dataset Acquisition<br>**(20%)** | The ARM template is fully complete with correct settings (StorageV2, HNS, replication, region), cost-effective decisions, accurate tagging, and comprehensive documentation(screenshots, code, explanations).<br>Dataset is correctly acquired and stored; data summary report is thorough with clear file types, schemas, key fields, and insightful structure/metadata observations | ARM template is mostly correct with minor omissions; tagging and documentation are generally good; deployment is successful with slight improvements needed.<br><br>Dataset acquisition and storage are sound; the summary report is mostly complete with minor gaps in schema or structural insights. | Deployment is functional but lacks key elements or contains configuration issues; documentation is basic; tagging/cost governance is partially met.<br><br>Dataset is accessed and minimally stored; summary report includes basic schema/field info but lacks depth or clarity. | Template or deployment is poorly configured or incomplete; minimal tagging/governance; screenshots or explanations are unclear or missing key steps.<br><br>Dataset partially acquired; summary report is incomplete, with incorrect or missing details. | The ARM template is missing, not functional, or lacks meaningful evidence of deployment or documentation.<br><br>Dataset not acquired; report is missing or irrelevant. |
| **Task 3:**<br>**Data Ingestion Using Azure Data Factory.**<br>**(LO1 and LO2)** | **3.1** Design and implement a data ingestion pipeline using Azure Data Factory to automate the movement of data into your storage environment.<br>**(25%)** | Fully functional pipeline with correct source and sink setup (including linked services), clear validation/debug/testing, successful execution, verified data ingestion into /raw/, complete step-by-step screenshots, and a well-written summary explaining pipeline purpose, flow, and observations. | Pipeline is mostly correct with proper source-sink setup and successful execution; screenshots and documentation are mostly complete; summary is clear with minor gaps. | Pipeline executes with basic functionality; some configuration or documentation elements are missing or inaccurate; summary provides limited explanation or depth. | Pipeline is partially implemented or fails execution; unclear or incomplete documentation; screenshots and summary report are limited or poorly structured. | Pipeline not implemented or does not function; minimal or no evidence of execution, verification, or understanding; summary missing or irrelevant. |

| Criterion | | Evidence | | | | |
|---|---|---|---|---|---|---|
| **Task and Weightage** | | **A (80-100%)** | **B (65-79%)** | **C (50-64%)** | **D (40-49%)** | **E (0-39%** |
| **Task 4:**<br><br>**Create an Azure Databricks Workspace and perform data transformation using code to automate the process.**<br><br>**(LO1 and LO2)** | **4.1** Create the Azure Databricks Workspace:<br><br>**4.2** Create a Single-Node Cluster:<br><br>**4.3** Data Integration and Authentication:<br><br>**4.4** Notebook and Code Execution:<br><br>**(25%)** | Databricks workspace and single-node cluster fully set up with correct configurations (name, region, pricing tier). Secure integration with Azure Data Lake Gen2 via app registration and RBAC is properly implemented. Data is successfully transformed using well-written and functional PySpark code in a mounted notebook. Complete and clear screenshots, code submission, and access control configurations provided. | Workspace and cluster set up correctly with minor issues. Integration and authentication were mostly accurate. Code executes with expected results. Screenshots and documentation are mostly complete. | Basic workspace and cluster setup done. Some issues with integration or authentication. The code runs but lacks clarity or depth. Screenshots and deliverables are partially complete. | Incomplete or incorrect setup of workspace/cluster. Integration steps are poorly configured or missing. The code is minimal, incorrect, or untested. Documentation or screenshots are limited or unclear. | Workspace not created or not functional. No working code, authentication not implemented. Missing screenshots, code, or documentation. |
| **Task 5:**<br><br>**Create Azure Synapse Analytics and Automate Data Integration.**<br><br>**(LO1 and LO2)** | **5.1** Create Azure Synapse Analytics Workspace.<br><br>**5.2** Build Azure Data Pipeline.<br><br>**5.3.** Create External Tables using SQL commands.<br><br>**5.4**. Perform Data Analysis.<br><br>**5.5**. Visualize the Results.<br><br>**(15%)** | Synapse workspace is correctly set up and linked with Data Lake Gen2. The data pipeline works flawlessly, transferring data to Synapse. External tables created and queried using SQL with accurate syntax and logical structure. Query results are accurate and relevant. Visualizations are well-designed, correctly represent the analysis, and all deliverables (screenshots, code, outputs) are complete and clear. | All major components are implemented with only minor errors or omissions. Data loads successfully, tables are created and queried properly. Visualizations are meaningful, though they may lack polish. Documentation and screenshots are mostly complete. | Workspace and pipeline are functioning with basic success. Tables and SQL queries are attempted, but some errors or logic issues exist. Visualizations are present but lack clarity or depth. Deliverables are partially complete. | Some elements of the task are incomplete or incorrectly configured. SQL queries are mostly inaccurate or missing. Visualizations are poor or absent. Screenshots and documentation are limited. | The task is largely incomplete or non-functional. Synapse workspace or pipeline is not set up, queries are missing or fail, and deliverables are absent or irrelevant. |