Student’s Full Name: Anand Ramaswamy Jayshree

Course Title: INFO531

Term name and year: Summer 2024

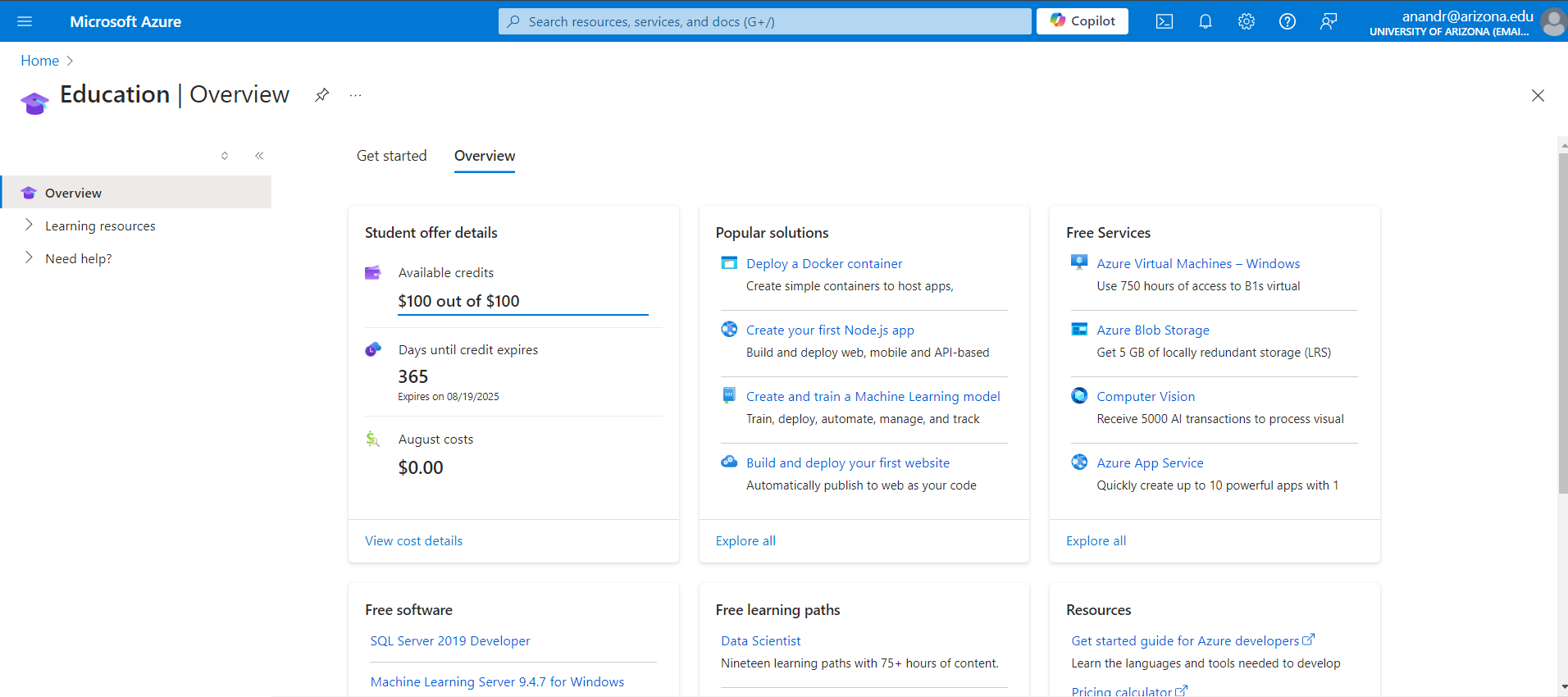
Submission Week: Week 6- Assignment 4

Instructor’s Name: Dr. Nayem Rahman

Date of Submission: August 11, 2024

**Q1. Create your Azure free account**

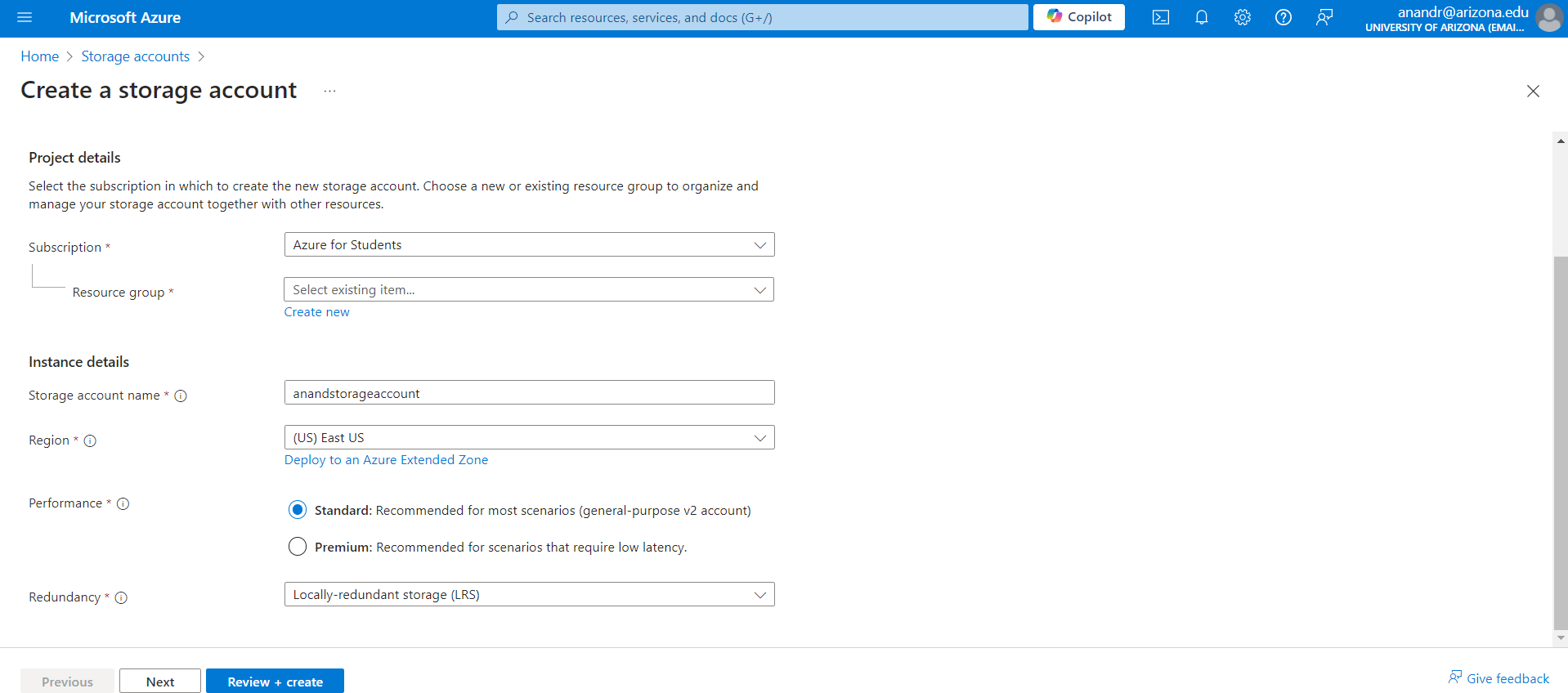
Go to the address https://azure.microsoft.com/en-us/free/students/ and create a free Azure account using your university email address. Note that you do NOT need a credit card to create the account. If you find yourself entering your credit card information know that you are doing it wrong.



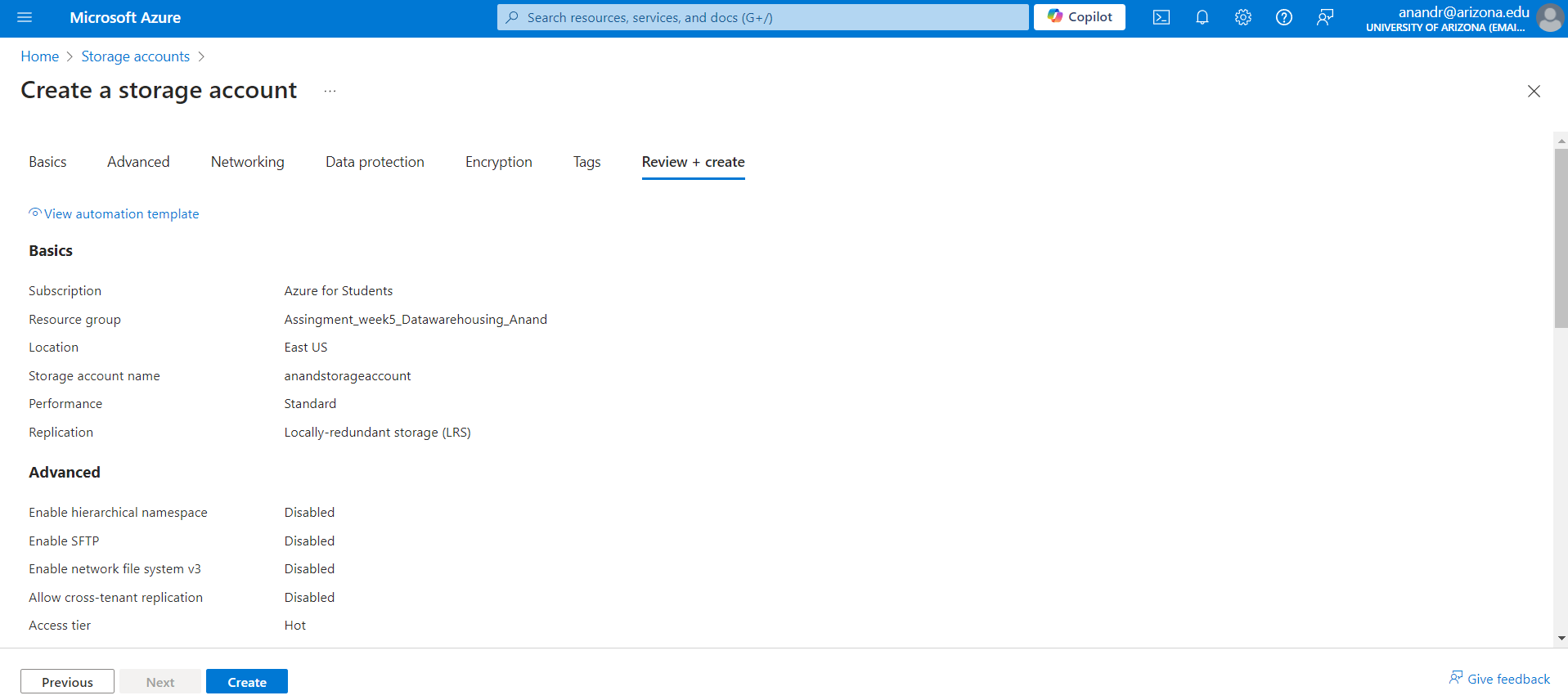
*Azure Account Created in my name with respect to university mail id*

**Q2. Create a container in Azure Storage, and to upload and download block blobs in that container.**

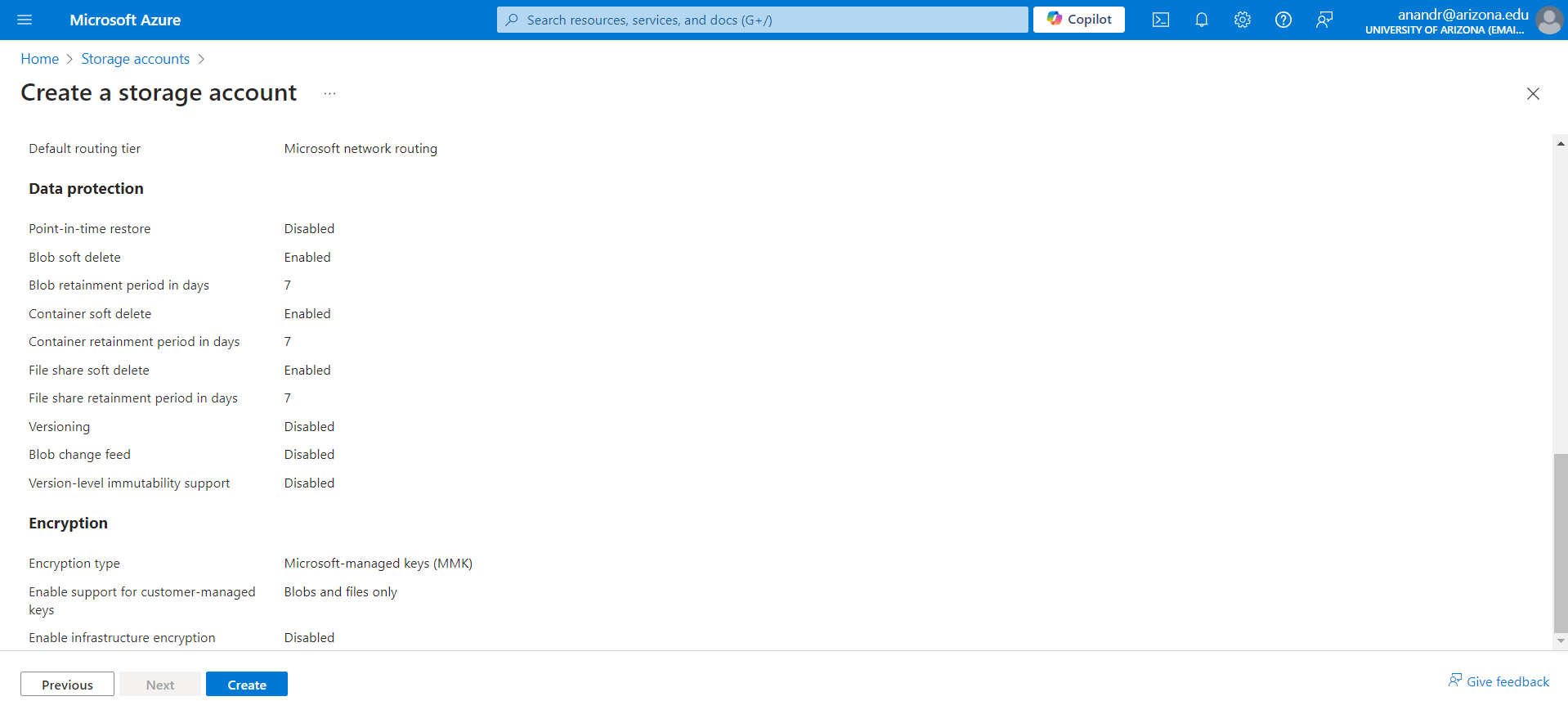
Follow the instructions here to create a container in Azure Storage, and to upload and download block blobs in that container. https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-portal Assumption: You already have an account using your university email address. Go to the portal and login.



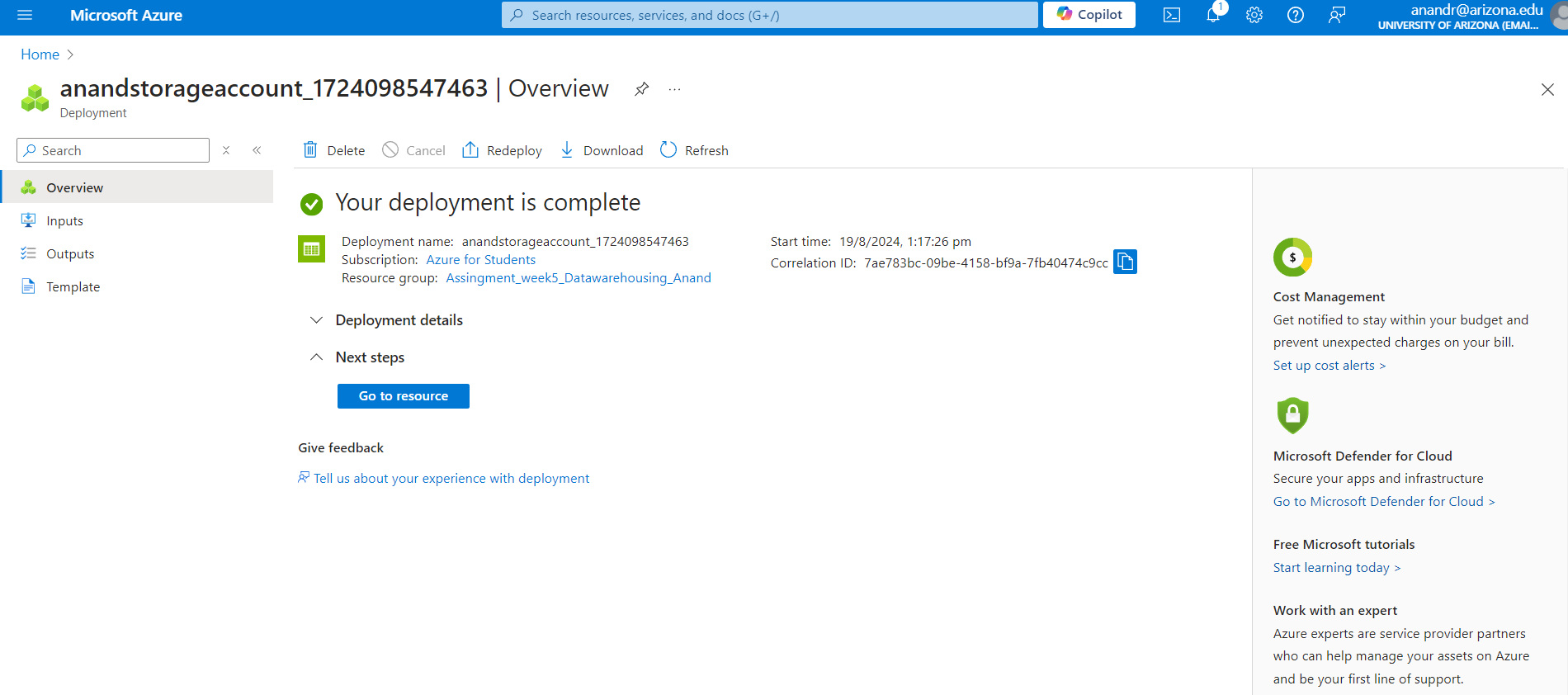
*Create Storage Account*



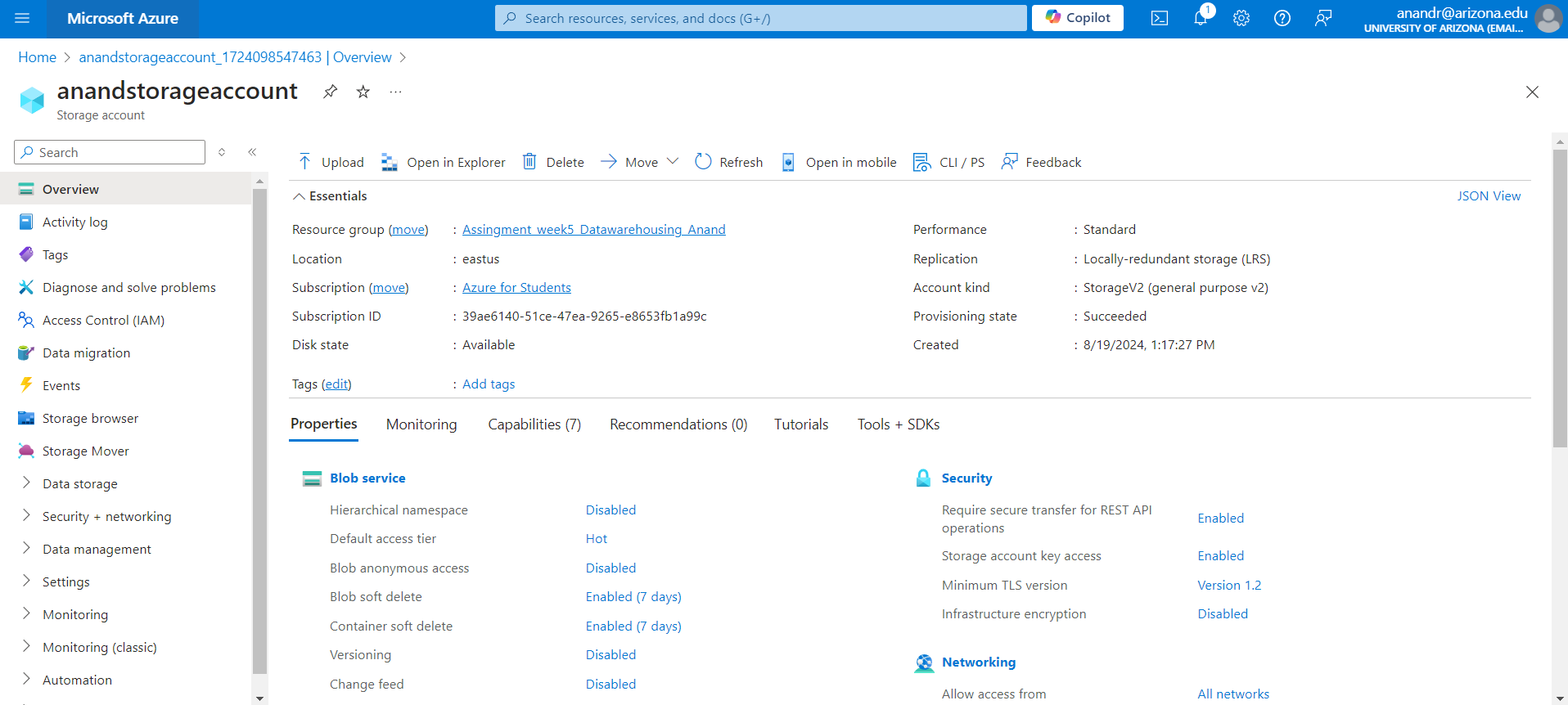
*Review the created storage account*



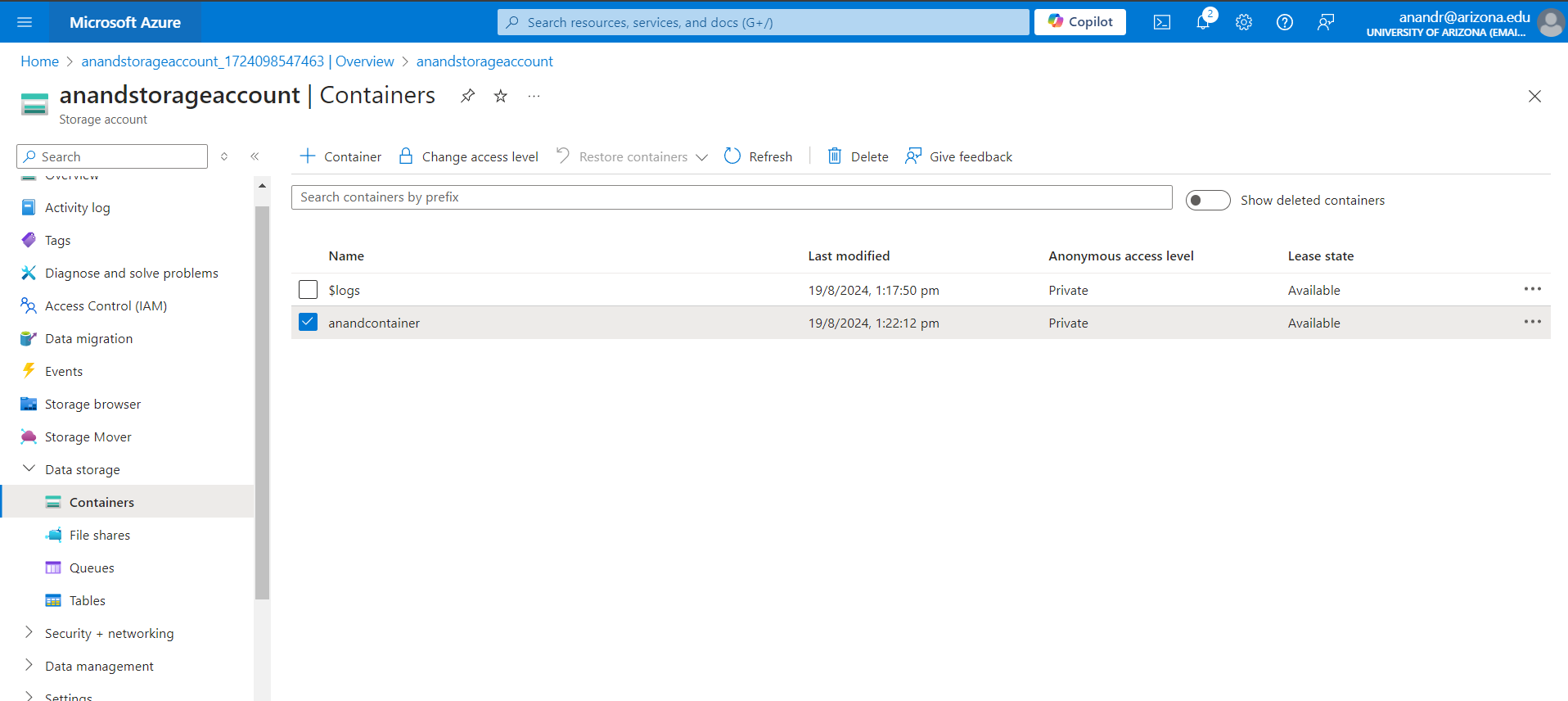
*Review the created storage account*



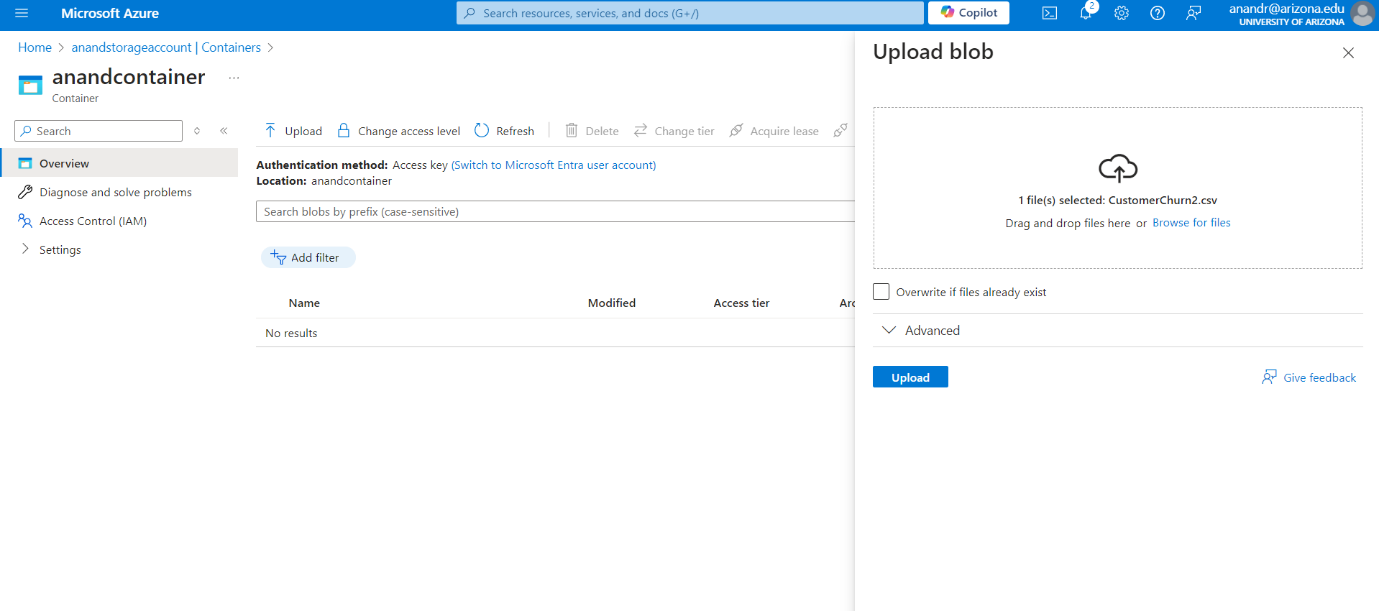
*Confirmation of Deployment of Storage account*



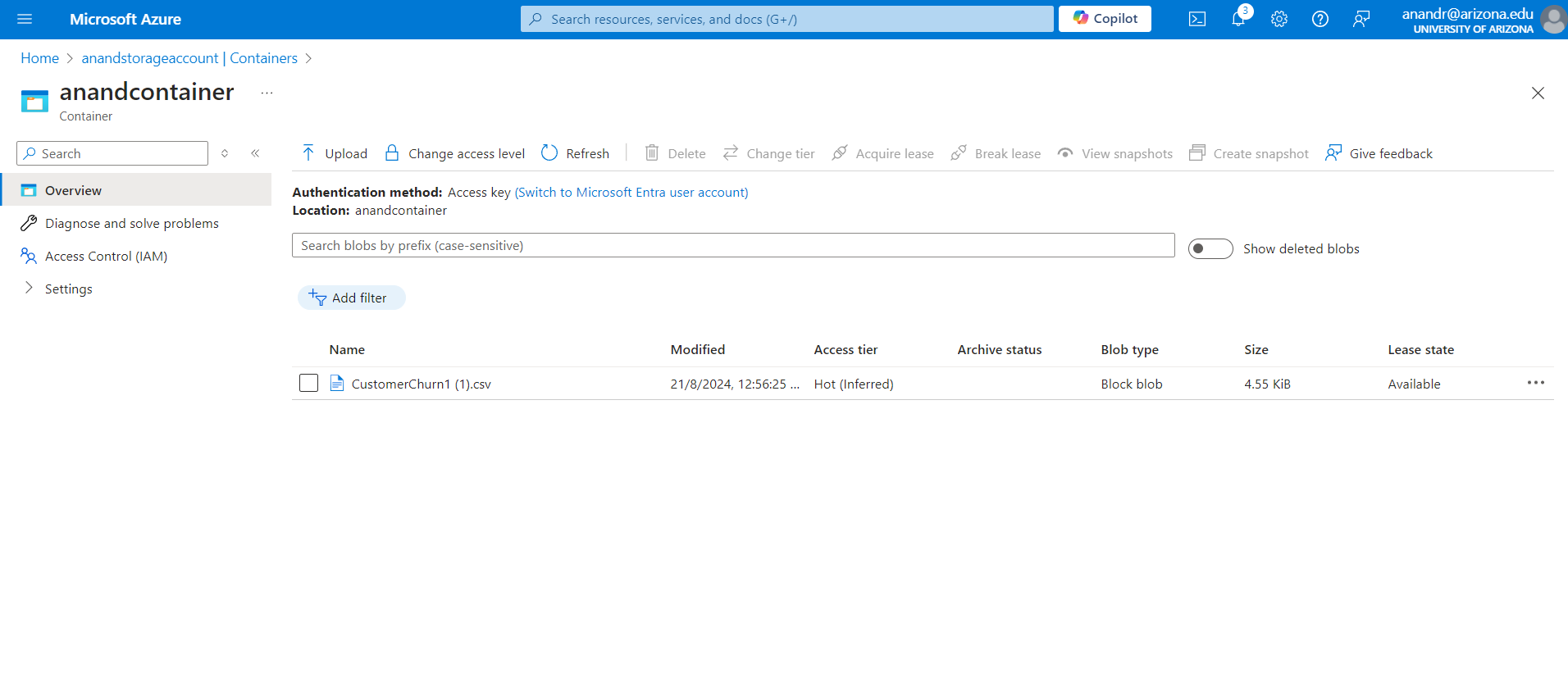
*Storage Account Details*



*Container Created*

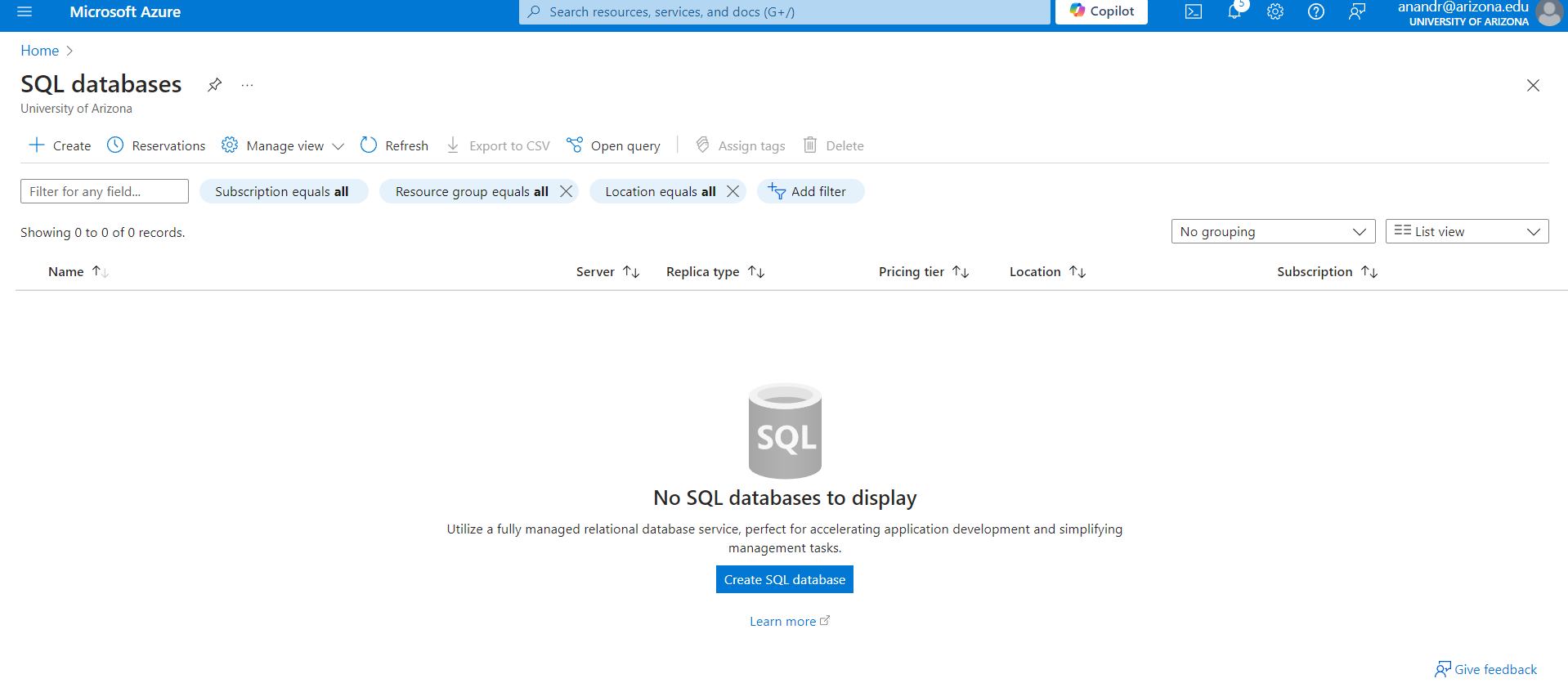


*Upload a blob in the container*

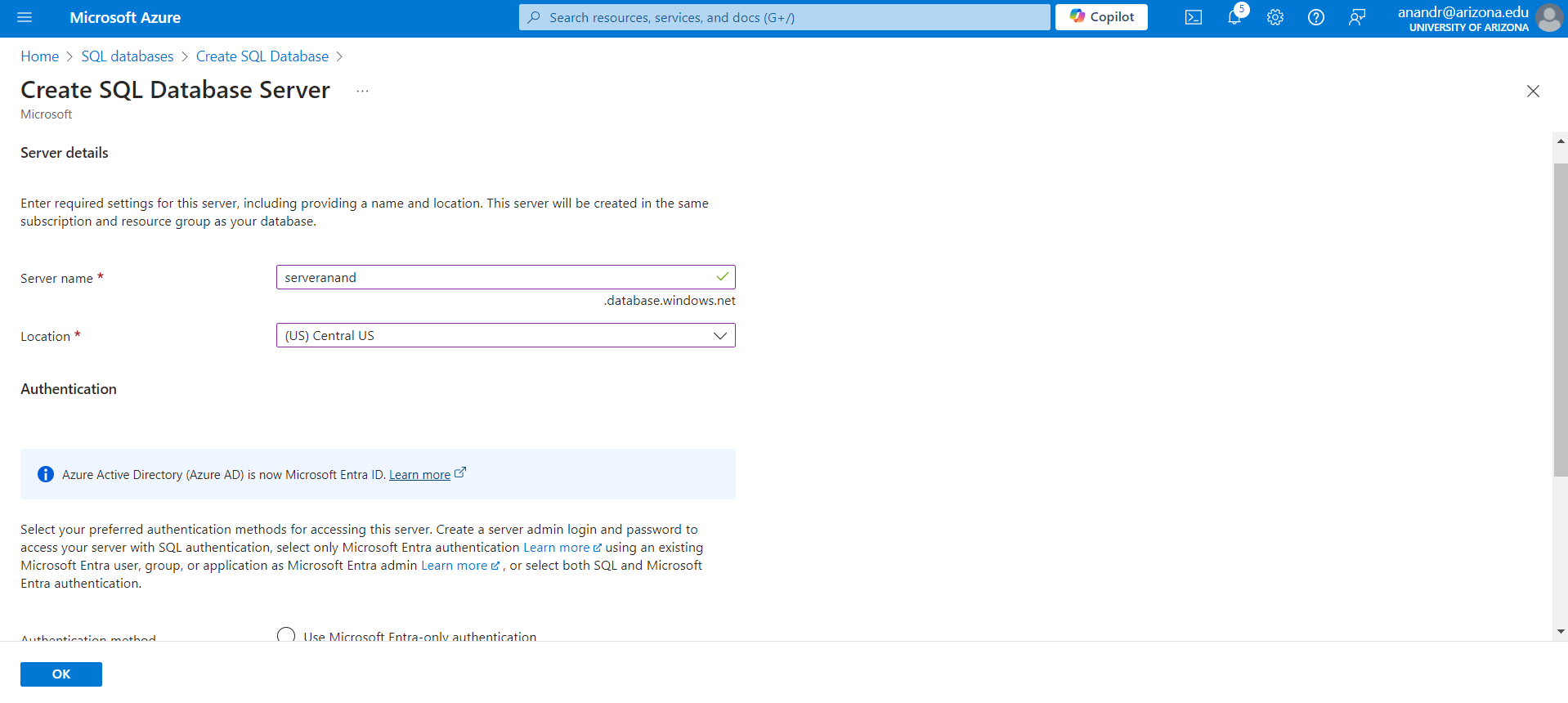


*Check the container for the blob*

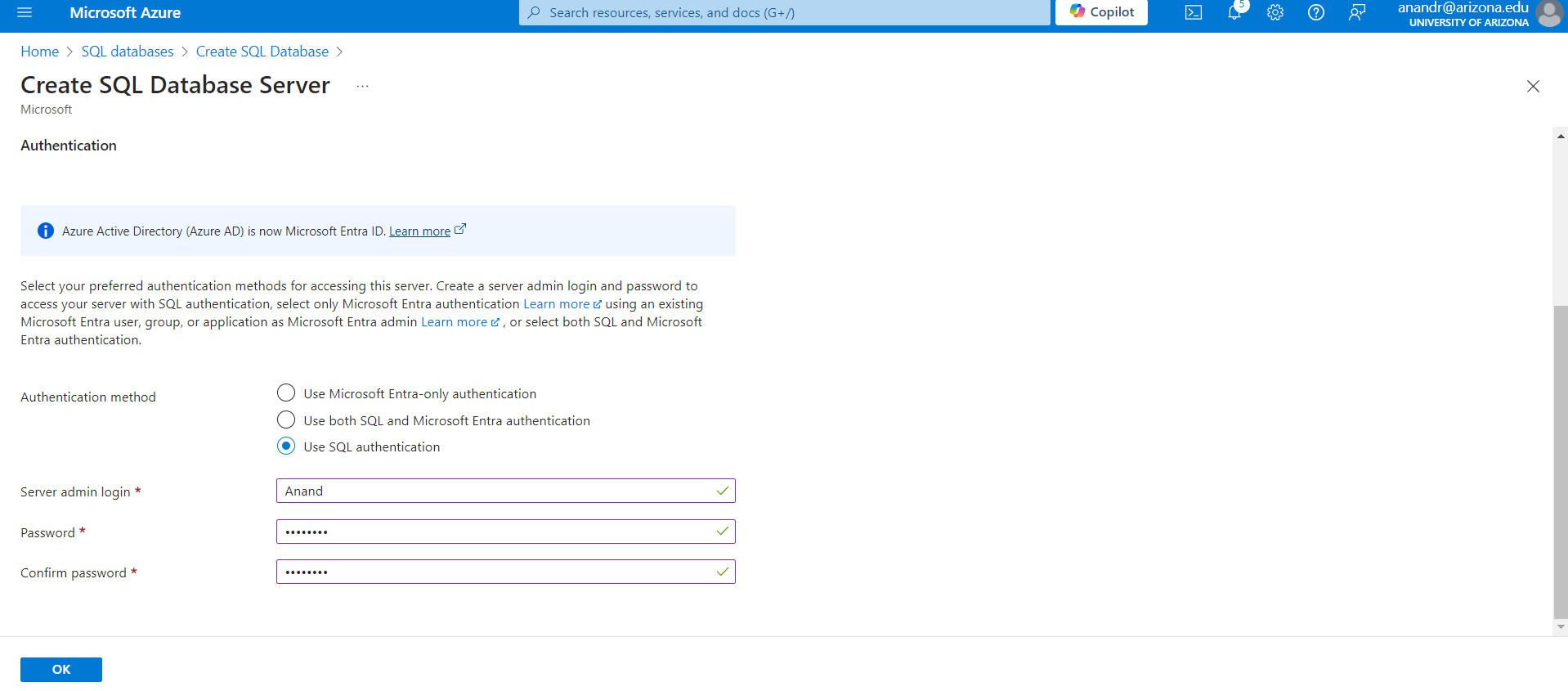
**Q3. Create a SQL database on Azure**



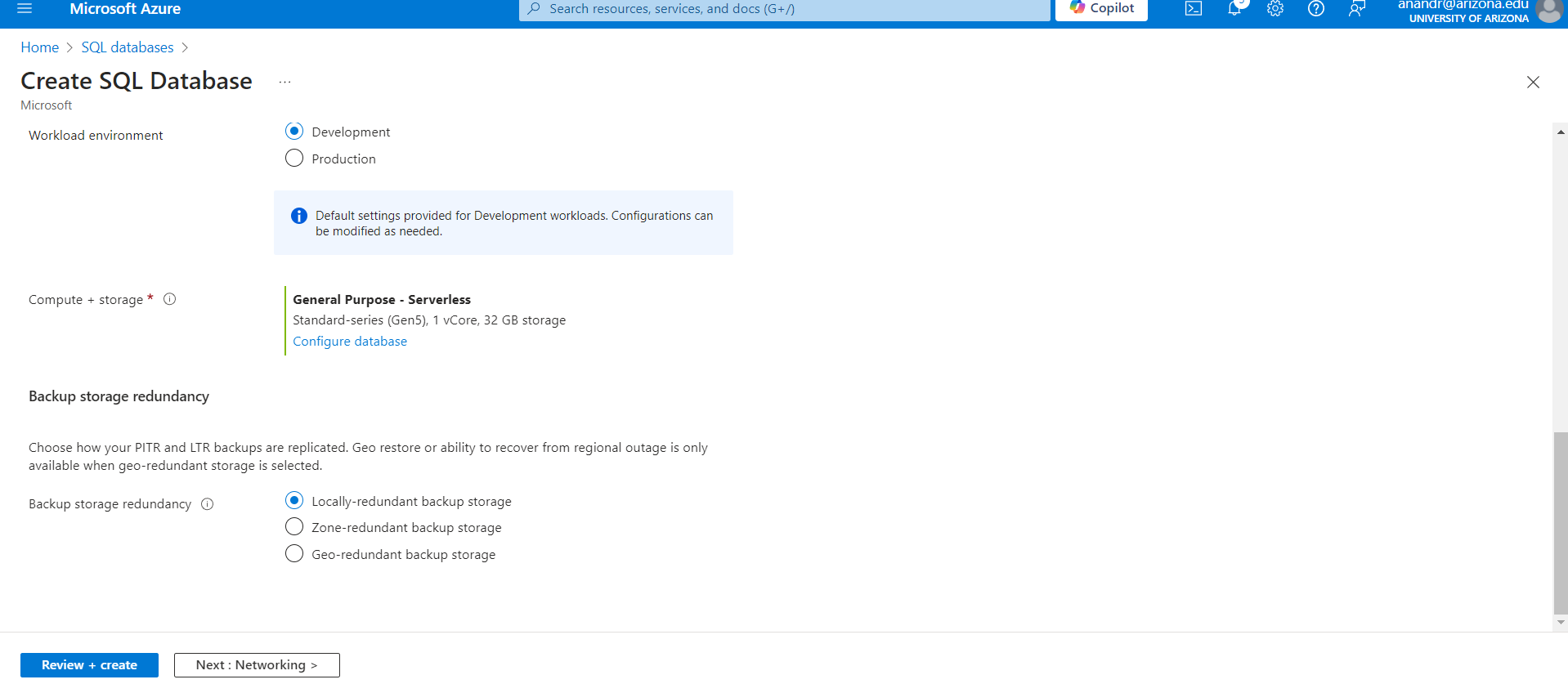
*Create SQL Database*



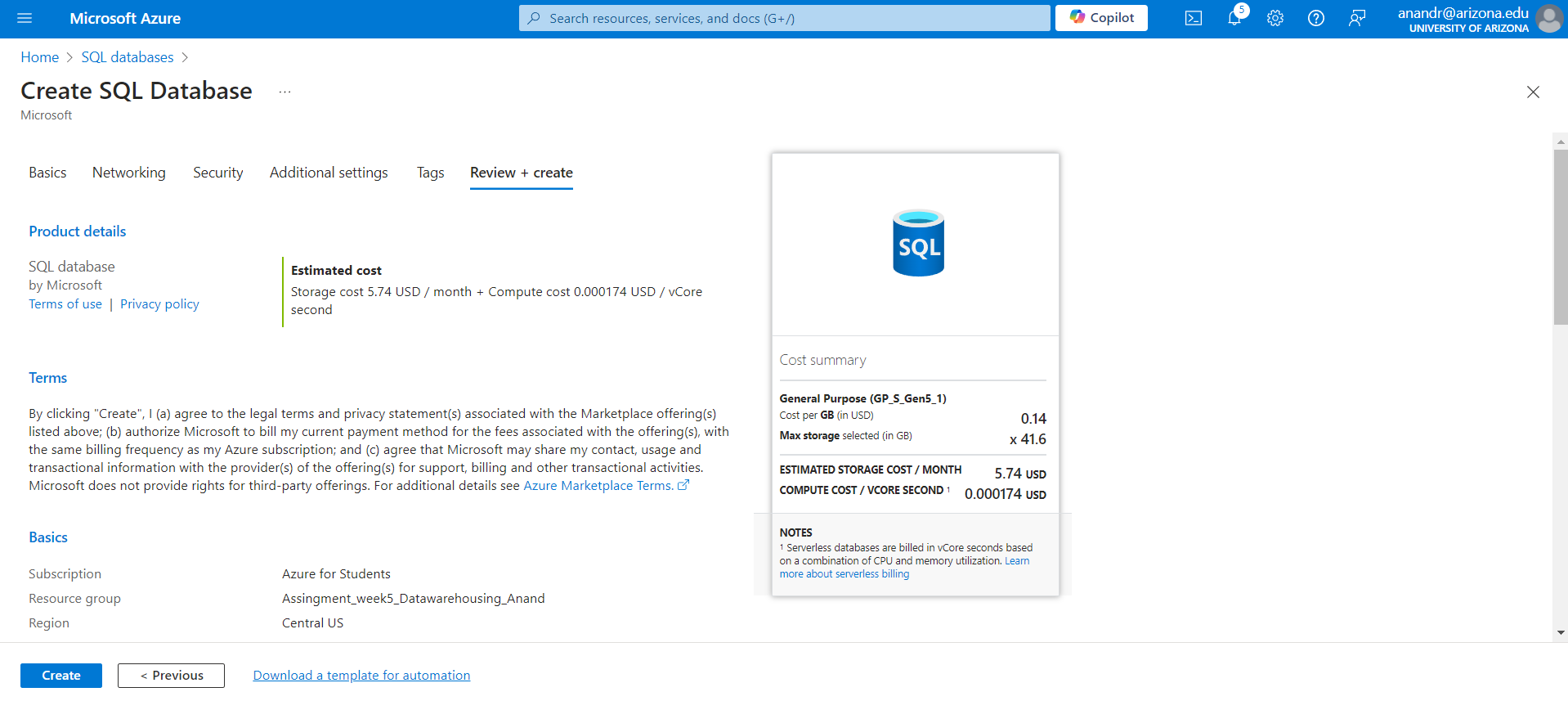
*Create SQL Database Server - 1*



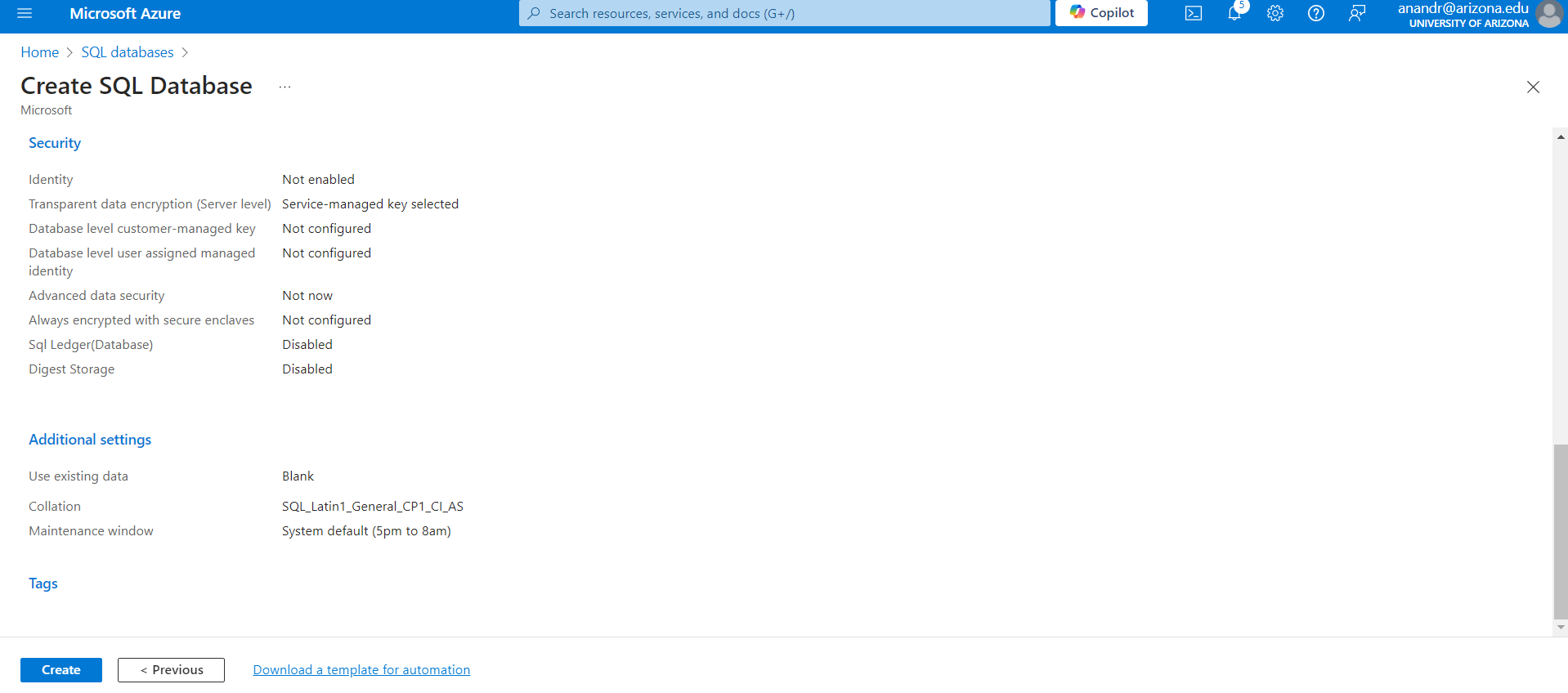
*Create SQL Database Server - 2*



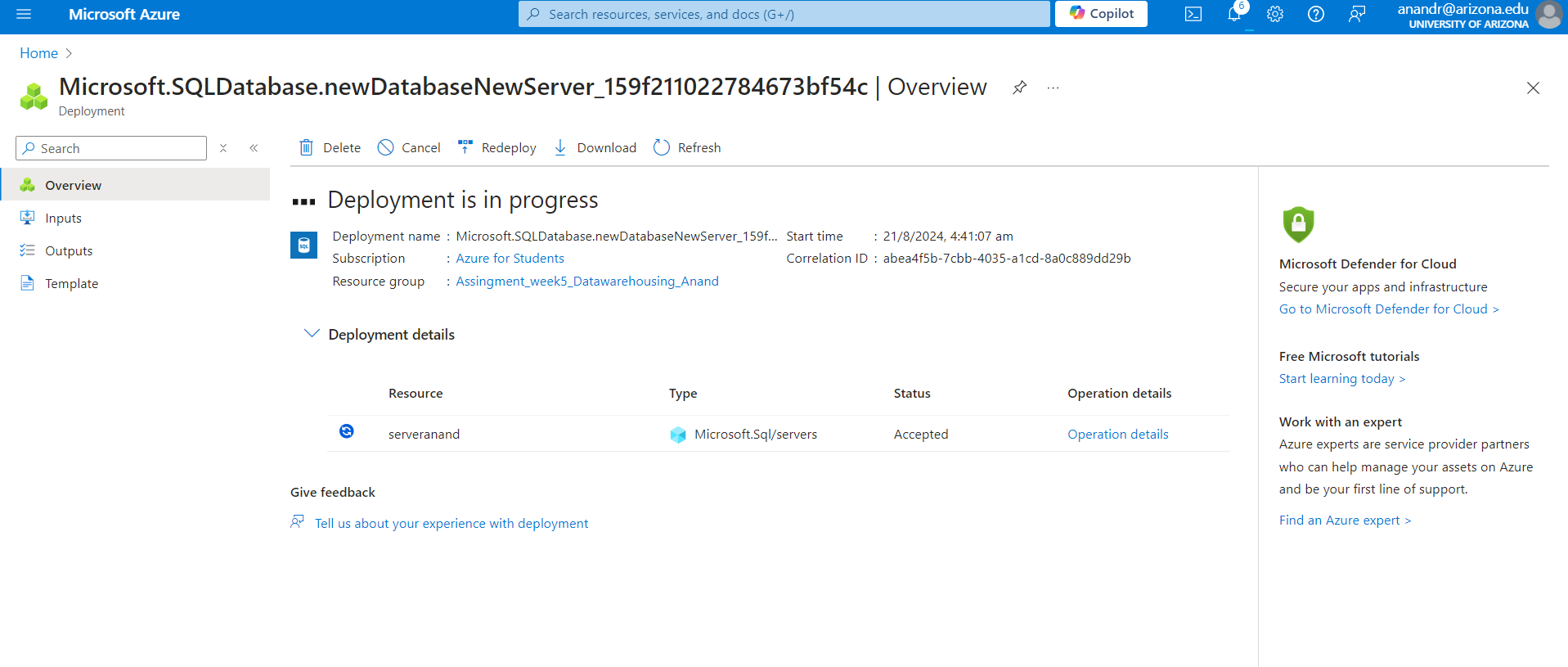
*Create SQL Database Server - 3*



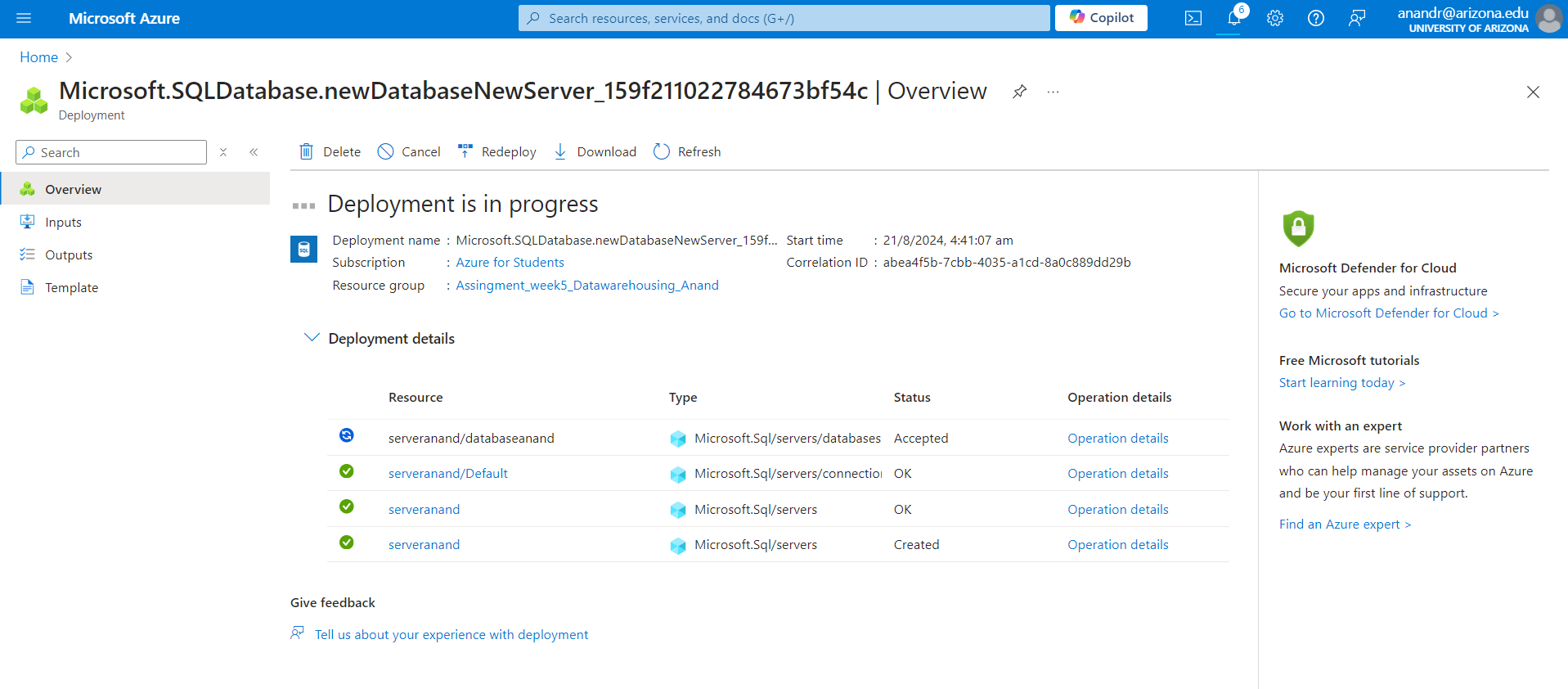
*SQL Database Cost estimation*

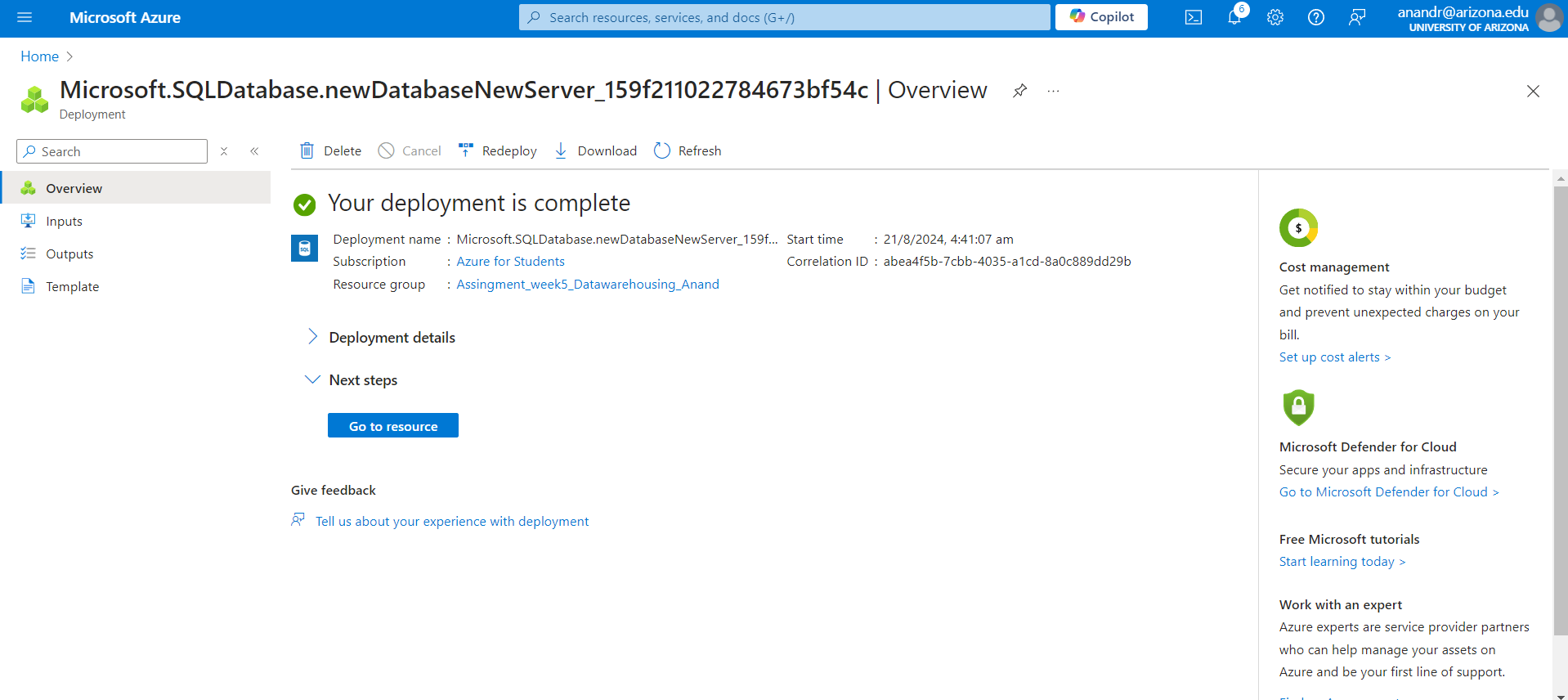


*Review the SQL Database*

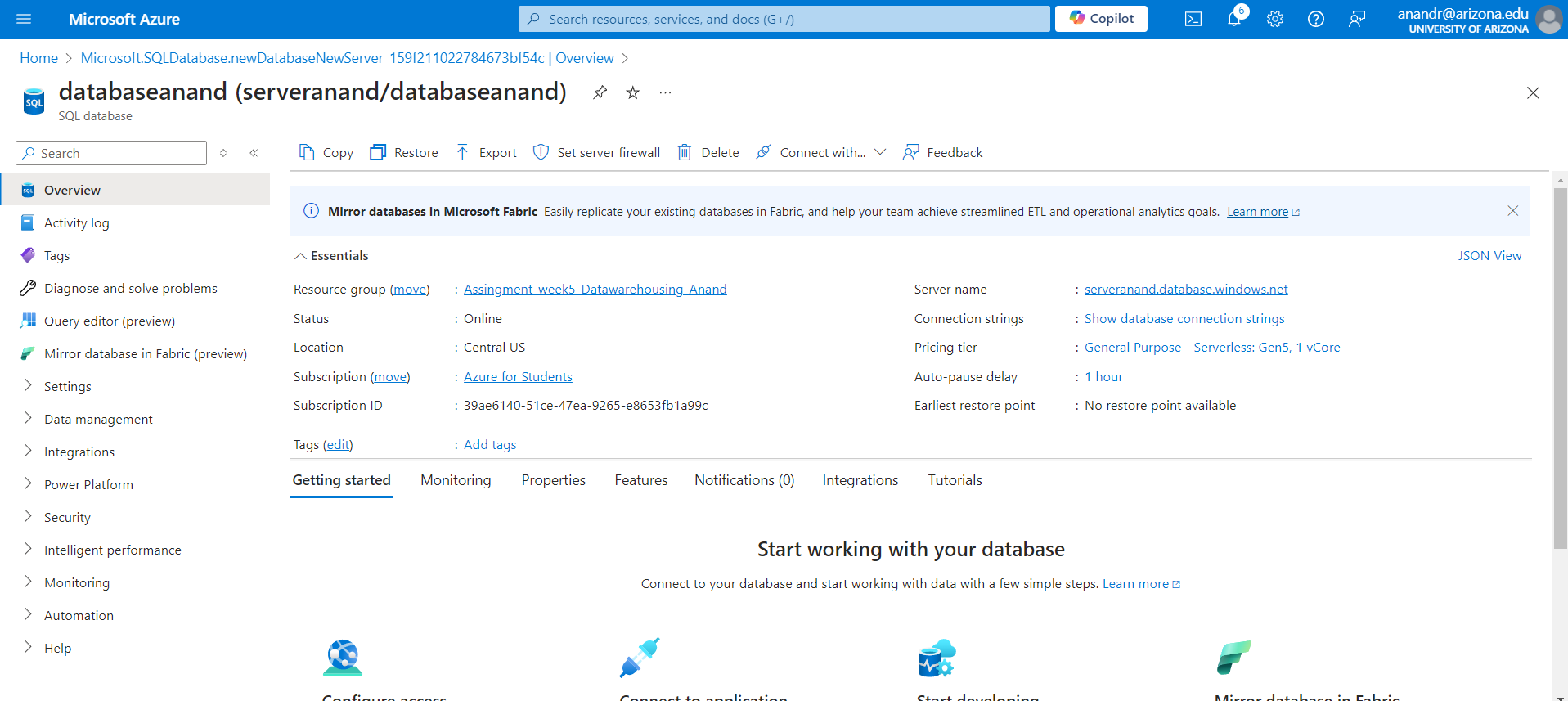


*Deploy SQL Database*

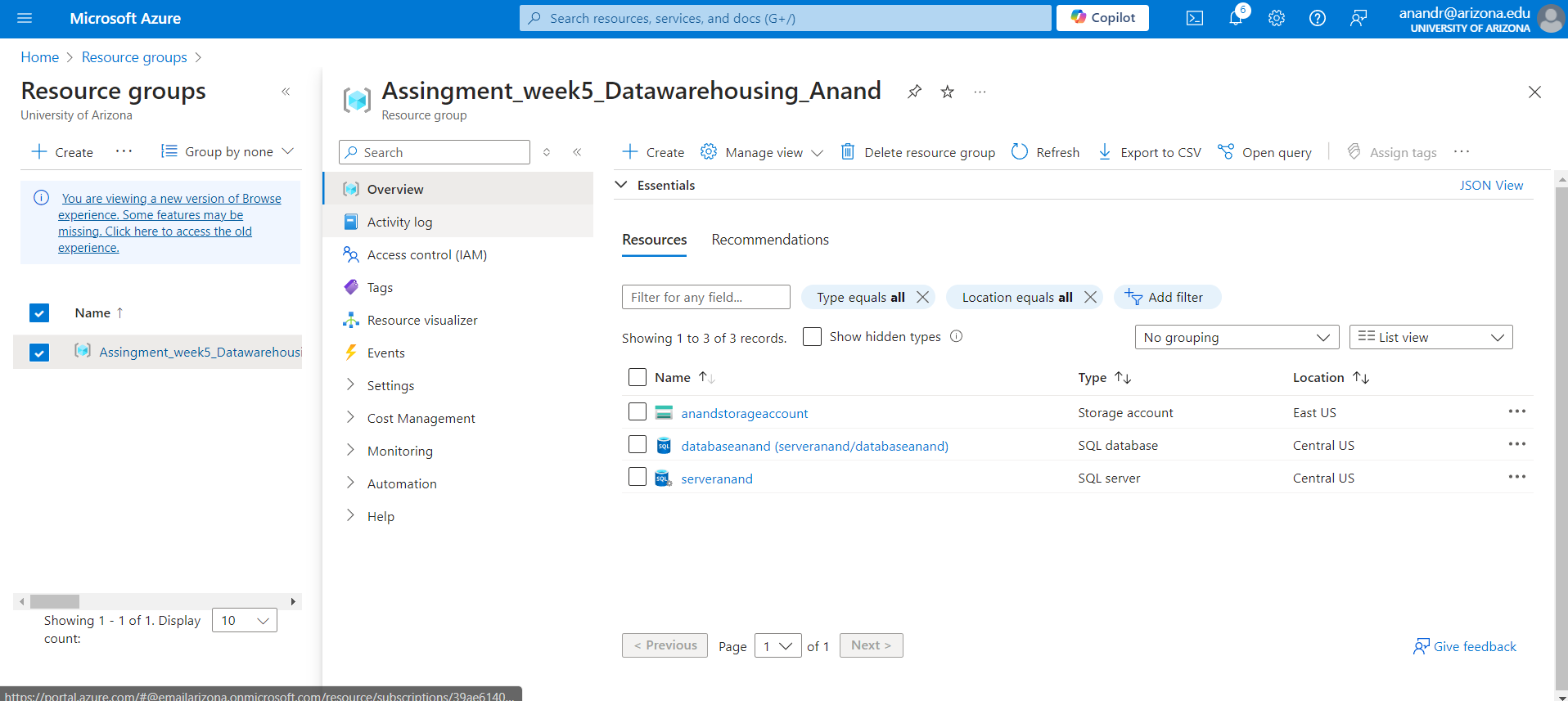




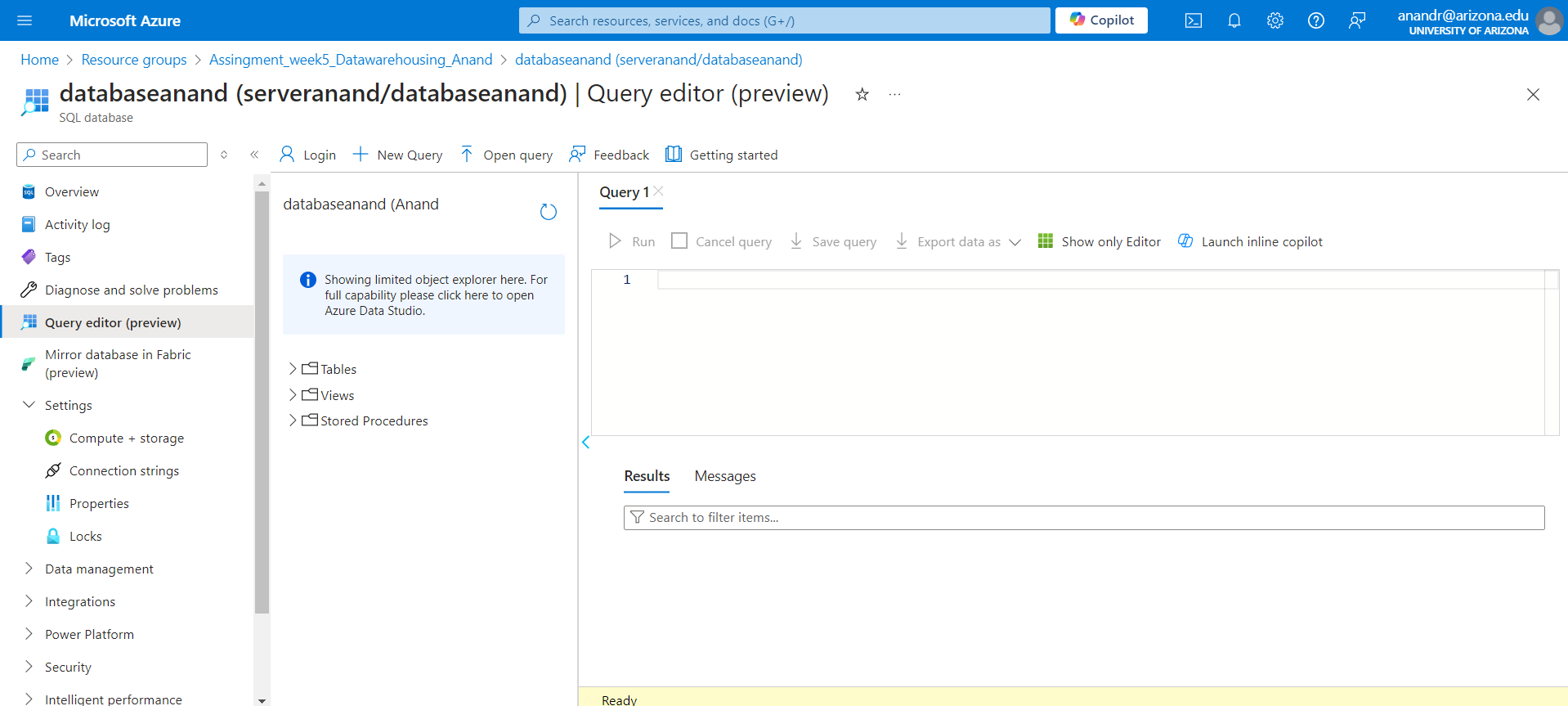
*Deployment of SQL Database Completed*



*Reviewing the Database*



*Resource group*



*Query Editor Platform*

**Q4. Summarize SQL database creation in Azure**

Write in one or two paragraphs what you have done and learned by creating a single database - Azure SQL Database – in Azure Portal.

1.**Create a Resource Group:** Organize Azure resources for better management.

2.**Choose SQL Database:** Select this option from the available services.

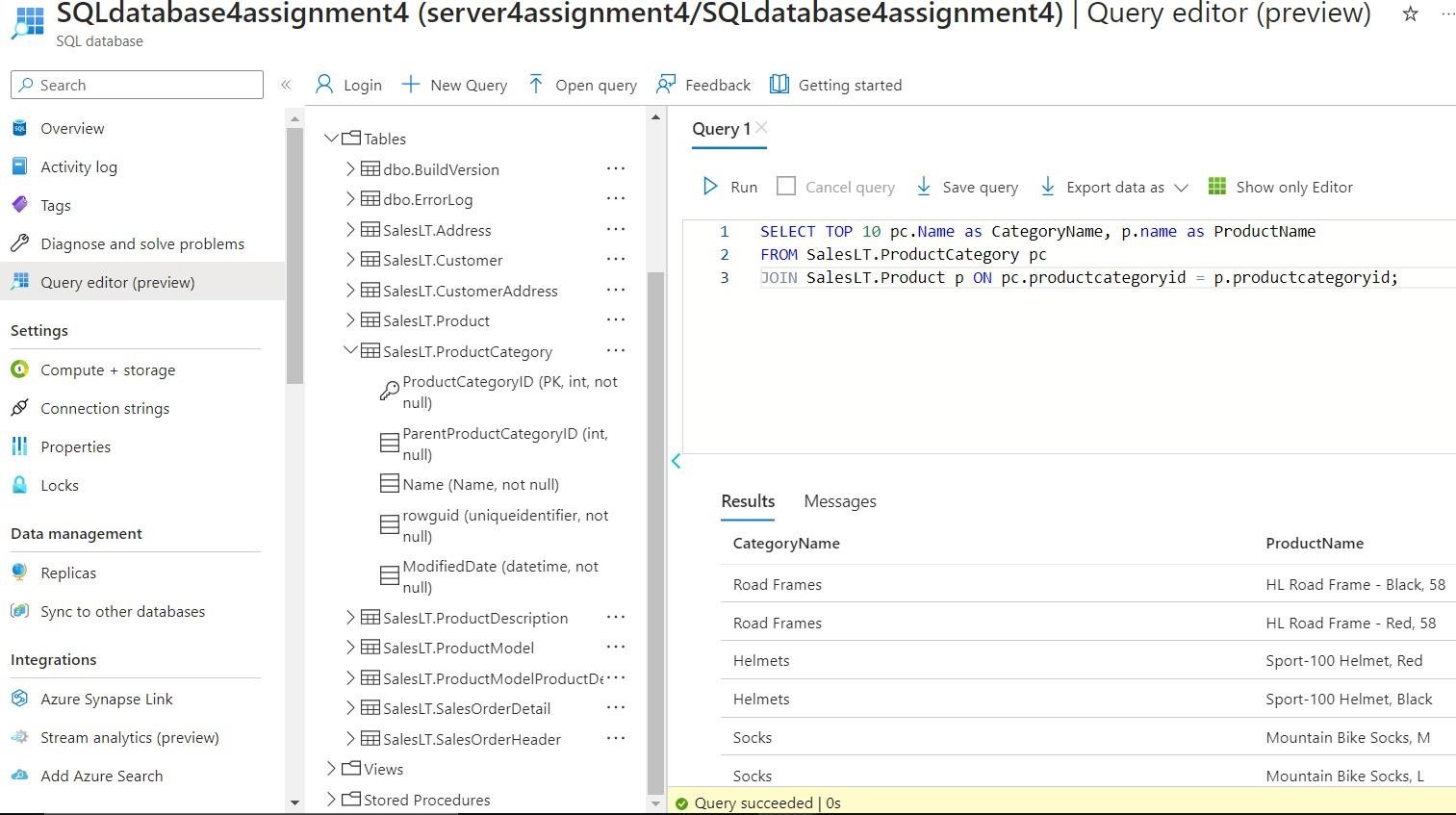
3. **Configure Settings:** Provide details like resource group, storage account, replication, network connectivity, performance tier, and storage size.

4. **Optional: Sample Data:** Use sample data to practice queries in the editor.

5. **Learn and Configure:** Understand performance tiers, backup options, and security features to optimize your database.

**Q5. Query the SQL database in Azure**

Query the database you created in Azure. You can use the Query editor (preview) in the Azure portal to connect to the database and query data. Run the following query in the query editor of the database on the Azure portal and take a screenshot of the result of the query. SELECT TOP 10 pc.Name as CategoryName, p.name as ProductName FROM SalesLT.ProductCategory pc JOIN SalesLT.Product p ON pc.productcategoryid = p.productcategoryid;



*Using SQL Query given in the question we get the following table output*

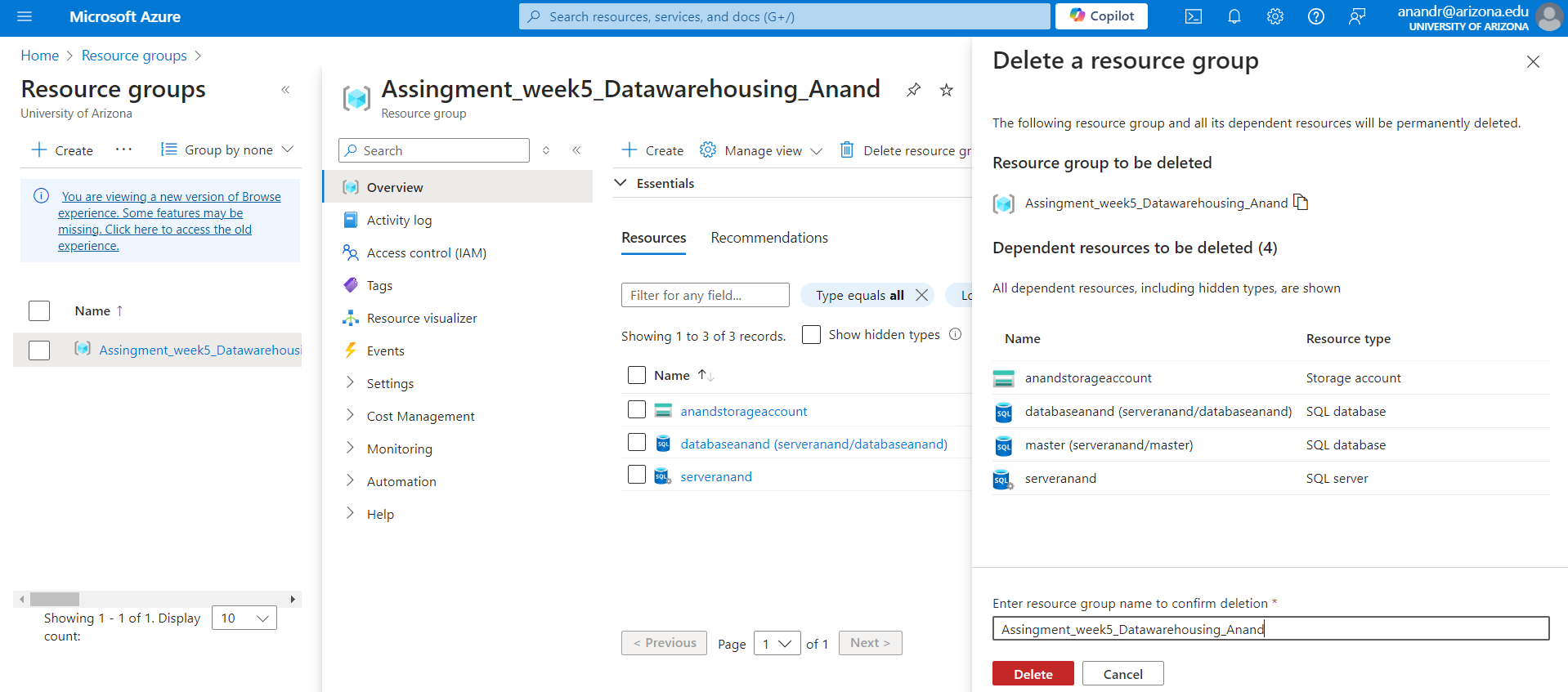
**Q6. Describe the SQL database "SalesLT" on Azure.**

**Analyze data in the tables in the SalesLT database schema using the Azure Query editor. Identify the primary key (PK) columns, and different data types, and observe the relationships among these tables. Write at least one paragraph.**

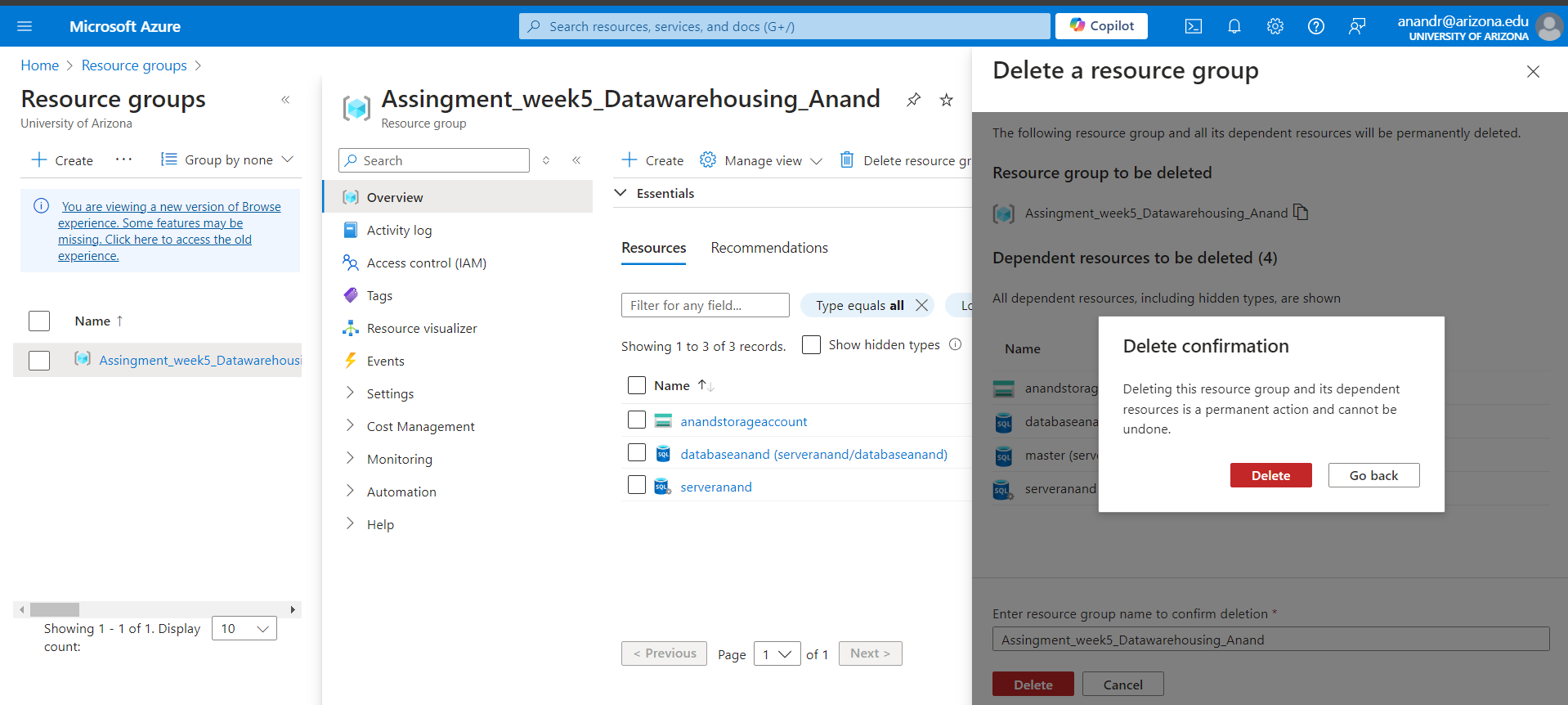
The SalesLT database is a structured relational database designed for e-commerce. Its tables, linked by primary and foreign keys, model products, sales, and customer interactions. For instance, each customer can have multiple sales orders, and each order can contain multiple product details. This structure ensures data consistency and enables complex queries to analyze sales patterns and gain valuable insights.

**Q7. Clean up resources in Azure**

When you're finished using these resources, delete the resource group you created, which will also delete the blob storage, server and single database within it. Show the screenshot that you deleted the resource group.



*Deleting the resource Group*



*Deleting Confirmation*

**Q8. Summarize the article on "Understanding Data Store Models" . Link -->** [**https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview**](https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)

**This article explores data storage models and their applications.** It distinguishes between relational and non-relational databases, highlighting their unique structures and querying methods. Relational databases, like SQL Server and MySQL, excel in handling structured data and complex queries. Non-relational databases, such as Azure Cosmos DB, MongoDB, and Cassandra, are designed for unstructured or semi-structured data and offer flexibility and scalability. The choice of data store depends on factors like data structure, scalability needs, transaction requirements, and query complexity. Understanding these differences is essential for optimizing performance and meeting application needs.

**Q9. Summarize the article on "Understanding the differences between NoSQL and relational databases". Link -->** [**https://learn.microsoft.com/en-us/azure/cosmos-db/relational-nosql**](https://learn.microsoft.com/en-us/azure/cosmos-db/relational-nosql)

This article on Microsoft Learn compares NoSQL and relational databases, exploring their design, performance, and ideal applications.

**Relational Databases**

**Structure:** Employ relational databases with predefined table structures and relationships defined by foreign keys.

**Data Model:** Data is organized into rows and columns within the tables with respect to the Schema

**Consistency:** they prove to support complex queries and transaction using SQL and strong Consistency.

**Scalability:** They are scaled vertically by adding more resources to a single server though modern relational databases are increasingly supporting horizontal scaling.

**NoSQL Databases**

**Structure:** flexible schemas available and store data in multiple format such as documents, pairs, columns or graphs.

**Data Model:** They typically deal with the unstructured or semi structured data and often don’t require predefined schema.

**Consistency:** They provide enhanced performance and scalability and optimizing for fast access.

**Scalability:** They are designed for the Horizontal scaling and deal with large volumes of data and high velocity workloads.

NoSQL databases excel in applications requiring high scalability and flexibility, while relational databases are better suited for applications needing complex transactions and strong data consistency. Businesses should carefully consider their specific needs when selecting a database model.

**Q10. Summarize the article on "Understanding Azure Cosmos DB". Link -->** [**https://learn.microsoft.com/en-us/azure/cosmos-db/introduction**](https://learn.microsoft.com/en-us/azure/cosmos-db/introduction)

**Azure Cosmos DB: A Versatile Database Service**

* **Global Distribution:** Seamlessly deploy databases in multiple regions for enhanced availability and performance.
* **Multi-Model Support:** Accommodate various data models, including document, key-value, column-family, and graph.
* **Consistency and Performance:** Choose from five consistency levels and benefit from low latency and high throughput.
* **Scalability:** Easily scale throughput and storage across global regions to handle varying workloads.
* **Integrated Security:** Protect your data with built-in security features, including encryption and role-based access control

**Q11. Summarize the article on "Azure Cosmos DB API for MongoDB". Link -->** [**https://learn.microsoft.com/en-us/azure/cosmos-db/mongodb/mongodb-introduction**](https://learn.microsoft.com/en-us/azure/cosmos-db/mongodb/mongodb-introduction)

**Azure Cosmos DB: Leveraging MongoDB API**

* **Seamless Integration:** Azure Cosmos DB provides compatibility with the MongoDB API, allowing users to migrate existing MongoDB applications with minimal changes.
* **Global Reach:** Enjoy global distribution and scalability, ensuring data replication across multiple regions and efficient handling of high-throughput workloads.
* **Performance and Consistency:** Choose from various consistency levels to balance data consistency and performance, while benefiting from automatic indexing and query optimization.
* **Security and Compliance:** Protect your data with robust security features and ensure compliance with industry standards.
* **Integrated Management:** Utilize Azure's management tools for efficient monitoring, diagnostics, and backup of your MongoDB workloads.

**Q12. Summarize the article on "Nodes and tables in Azure Database for PostgreSQL – Hyperscale". Link -->** [**https://learn.microsoft.com/en-us/azure/postgresql/hyperscale/concepts-nodes**](https://learn.microsoft.com/en-us/azure/postgresql/hyperscale/concepts-nodes)

**Azure Database for PostgreSQL - Hyperscale**

* **Horizontal Scaling:** Hyperscale enables data distribution across multiple nodes for improved scalability.
* **Nodes:** The architecture includes coordinator and worker nodes, each with specific roles in query processing and data management.
* **Distributed Tables:** Data is divided into shards and distributed across worker nodes for enhanced performance and scalability.
* **Data Distribution:** The distribution method (hash or range) determines how data is partitioned and queried.
* **Performance and Scalability:** Hyperscale offers significant improvements in query performance and scalability, allowing users to handle large datasets and high transaction volumes effectively.

**Q13. Summarize the article on "Overview - Azure Database for PostgreSQL - Flexible Server". Link -->** [**https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/overview**](https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/overview)

**Azure Database for PostgreSQL – Flexible Server**

* **Enhanced Control:** Flexible Server offers a high level of control over database configuration and management.
* **Granular Configuration:** Customize server parameters for performance tuning, maintenance windows, and backup schedules.
* **High Availability:** Ensure continuous operation with built-in replication and automatic failover.
* **Flexible Scaling:** Adjust compute and storage resources independently to match workload demands.
* **Cost Optimization:** Benefit from burstable compute and demand-based scaling for cost savings.
* **Robust Security:** Protect data with network isolation, encryption, and compliance with industry standards.
* **Automated Management:** Simplify database administration with automated backups, updates, and patching.

**Q14. Summarize the article on "Azure Database for PostgreSQL - Single Server". Link-->** [**https://learn.microsoft.com/en-us/azure/postgresql/single-server/concepts-servers**](https://learn.microsoft.com/en-us/azure/postgresql/single-server/concepts-servers)

Azure Database for PostgreSQL – Single Server

* **Managed PostgreSQL:** Enjoy a simplified setup and management with a single-instance PostgreSQL database.
* **High Availability:** Benefit from built-in high availability with automatic failover for minimal downtime.
* **Scalability:** Adjust compute and storage resources independently to meet workload demands.
* **Automated Management:** Simplify database administration with automated backups, patching, and updates.
* **Robust Security:** Protect data with encryption, virtual network integration, and advanced threat protection.
* **Monitoring and Alerts:** Track performance metrics and receive alerts to maintain optimal database health.
* **Flexible Pricing:** Choose from various pricing tiers to align with your budget and performance needs.

**Q15. Summarize the article on "What is Azure Database for PostgreSQL?". Link -->** [**https://learn.microsoft.com/en-us/azure/postgresql/single-server/overview**](https://learn.microsoft.com/en-us/azure/postgresql/single-server/overview)

**Azure Database for PostgreSQL: A Managed PostgreSQL Service**

* **Managed Service:** Simplify database deployment, management, and scaling with automated features.
* **Deployment Options:** Choose from Single Server, Flexible Server, or Hyperscale (Citus) based on your application needs.
* **Performance and Scaling:** Adjust compute and storage resources to meet workload demands.
* **Security and Compliance:** Protect data with encryption, network isolation, and compliance with industry standards.
* **High Availability:** Ensure minimal downtime with built-in high availability and automatic failover.
* **Monitoring and Management:** Utilize tools for performance monitoring, alerts, and backup management.

**Q16. Summarize the article on "Azure Database for MySQL - Flexible Server". Link -->** [**https://learn.microsoft.com/en-us/azure/mysql/flexible-server/overview**](https://learn.microsoft.com/en-us/azure/mysql/flexible-server/overview)

**Azure Database for MySQL – Flexible Server**

* **Enhanced Configurability:** Enjoy greater control over database settings and management.
* **High Availability:** Ensure continuous operation with built-in replication and automatic failover.
* **Scalability:** Adjust compute and storage resources to meet varying workload demands.
* **Cost-Effective Pricing:** Benefit from burstable compute and demand-based scaling for optimized costs.
* **Robust Security:** Protect data with encryption, network isolation, and compliance with industry standards.
* **Automated Management:** Simplify database administration with automated tasks like backups, patching, and updates.

**Q17. Summarize the article on "Azure Database for MySQL Single Server". Link -->** [**https://learn.microsoft.com/en-us/azure/mysql/single-server/single-server-overview**](https://learn.microsoft.com/en-us/azure/mysql/single-server/single-server-overview)

**Azure Database for MySQL – Single Server**

* **Managed MySQL:** Enjoy simplified deployment, management, and maintenance of MySQL databases.
* **High Availability:** Benefit from built-in high availability with automatic failover for minimal downtime.
* **Scalability:** Adjust compute and storage resources to meet workload demands.
* **Automated Management:** Simplify database administration with automated backups, monitoring, and security updates.
* **Robust Security:** Protect data with encryption, network isolation, and compliance with industry standards.
* **Flexible Pricing:** Choose from various pricing tiers to align with your budget and performance needs.