Azure SQL Database - how to provision, access and development guide

Azure SQL Database is a cloud-based relational database service provided by Microsoft Azure, designed to meet the demands of modern, data-driven applications. Leveraging the power of Microsoft SQL Server, Azure SQL Database offers a fully managed, scalable, and secure platform for organisations to store, manage, and analyse their data in the cloud. Whether you are a small business or an enterprise, Azure SQL Database provides a flexible and cost-effective solution to meet your database needs.

Key Features and Uses:

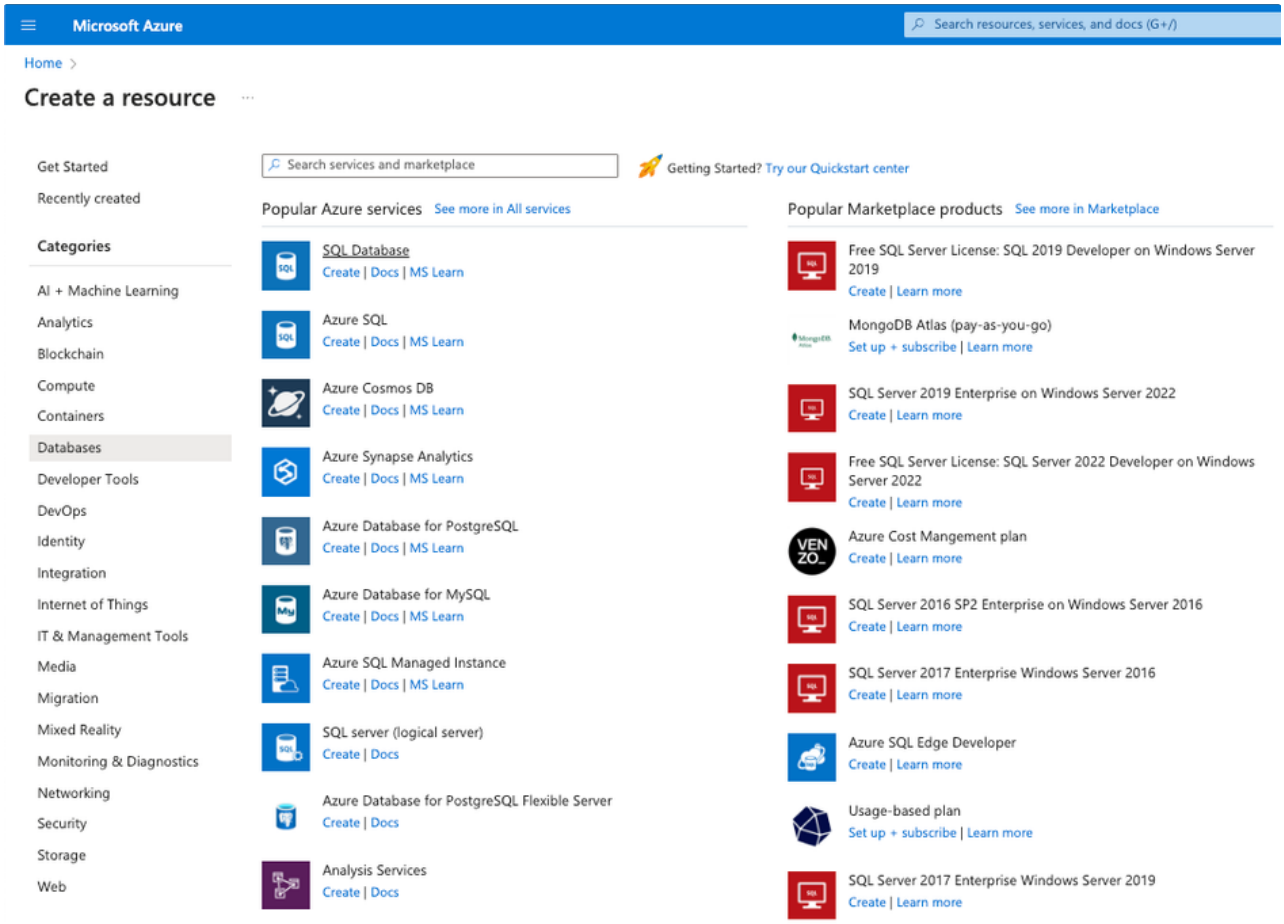
- **Managed Service:** Azure SQL Database is a fully managed service, eliminating the need for organisations to worry about database administration tasks such as patching, backups, and monitoring. This allows businesses to focus on building and optimizing their applications.
- **Scalability:** With Azure SQL Database, you can easily scale your database resources up or down based on demand. This flexibility ensures that your application performance remains optimal, even during peak usage periods.
- **High Availability:** Azure SQL Database offers built-in high availability features, including automatic backups, geo-replication, and failover capabilities. This ensures that your data is protected against unexpected failures, providing a reliable and resilient database solution.
- **Security and Compliance:** Benefit from robust security features, including advanced threat protection, data encryption, and compliance certifications. Azure SQL Database helps organisations meet regulatory requirements and safeguard sensitive information.
- **Advanced Analytics:** Integrate seamlessly with Azure services like Azure Machine Learning and Power BI to unlock advanced analytics capabilities. This enables organisations to derive valuable insights from their data, driving informed decision-making.
- **Compatibility:** Azure SQL Database is compatible with existing SQL Server applications, making it easy for organisations to migrate their on-premises databases to the cloud without significant code changes.

Benefits:

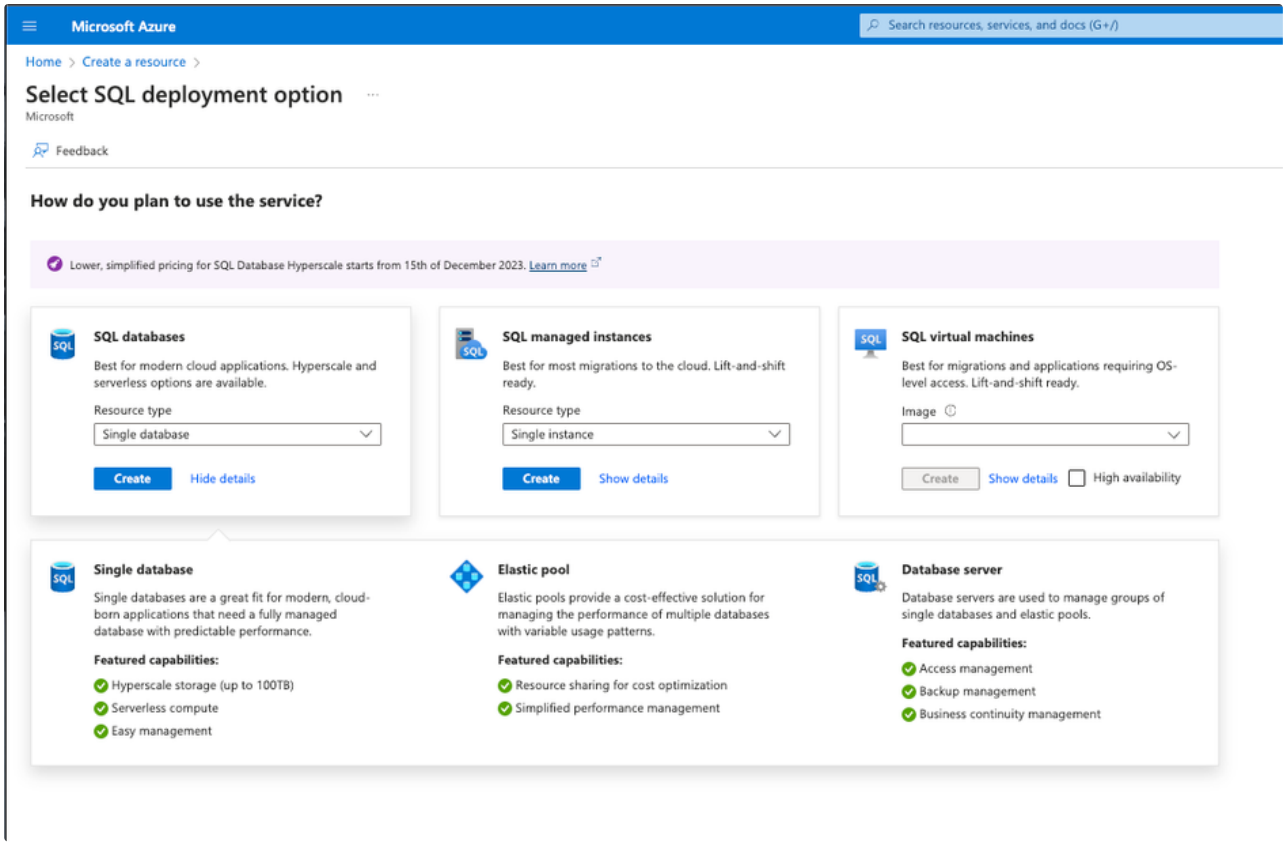
- **Cost Efficiency:** Pay only for the resources you use, with the ability to scale up or down as needed. This cost-effective model allows organisations to optimize their database expenses and allocate resources efficiently.
- **Global Reach:** Azure SQL Database provides global distribution, allowing you to deploy databases in multiple regions. This ensures low-latency access for users worldwide and enhances the overall performance of your applications.
- **Time-to-Market:** With a managed service and simplified administration, developers can focus on building features and functionality rather than managing infrastructure. This accelerates development cycles and reduces time-to-market for new applications.
- **Reliability and Disaster Recovery:** Benefit from built-in high availability and disaster recovery features, ensuring that your data is always available and protected. This enhances the reliability of your applications and minimizes downtime.
- **Continuous Innovation:** Azure SQL Database is part of the broader Azure ecosystem, providing access to a wide range of integrated services. This allows organizations to leverage the latest technologies and innovations as they evolve in the Azure cloud environment.

In order to create your own Azure SQL Database , please follow the steps as mentioned below -

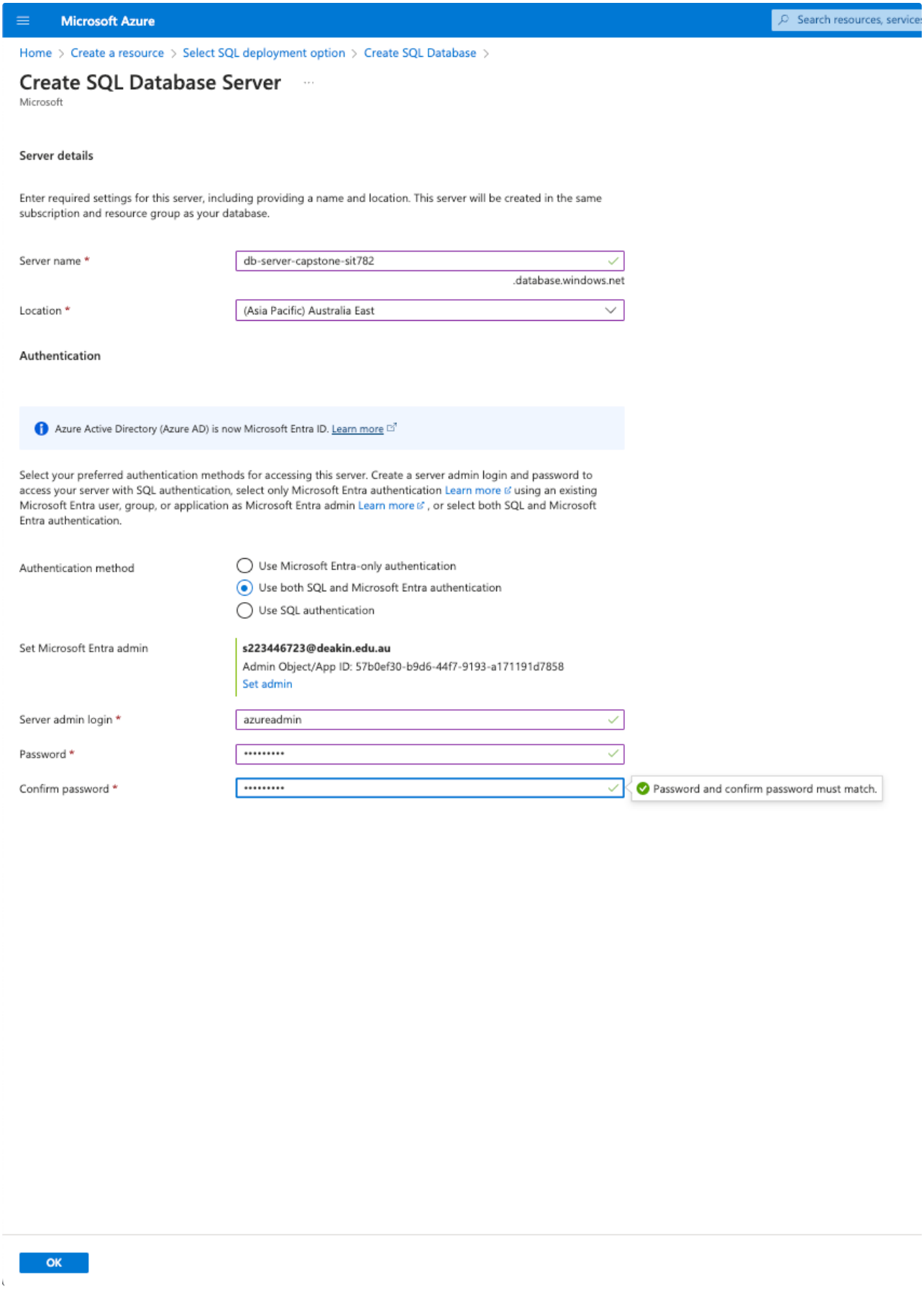
1. Click “Create a Resource” on the homepage and then select “Databases” on the left pane as below -



2. Select “Azure SQL” on the right hand side, and then you will get the following page which asks what kind of service is required. We do not need managed instances nor virtual machines, hence we will be selecting SQL database, which is the first option as below -

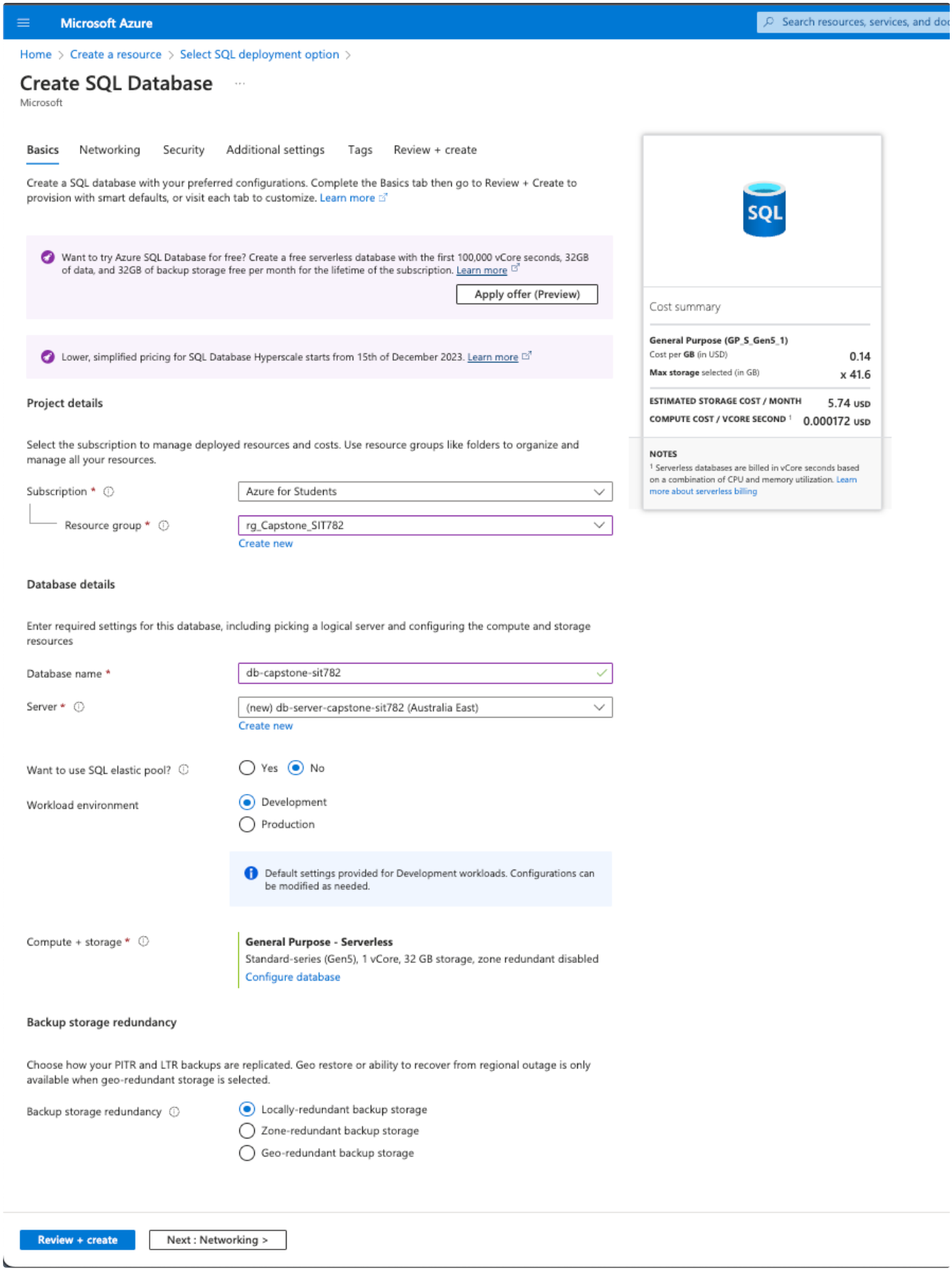


3. Once you click on “Create”, you will land in a page like this. Fill in your server name and select location as Australia East. Server name can be anything but needs to be unique in the region. If there is a name collision, it will prompt you to change it to something unique. Next, under “Authentication method”, select “Use both SQL and Microsoft Entra authentication. Set your Deakin id as the admin. Also, set a Server admin login, which is different from your Deakin id. Click “Ok” when done at the bottom of the page.

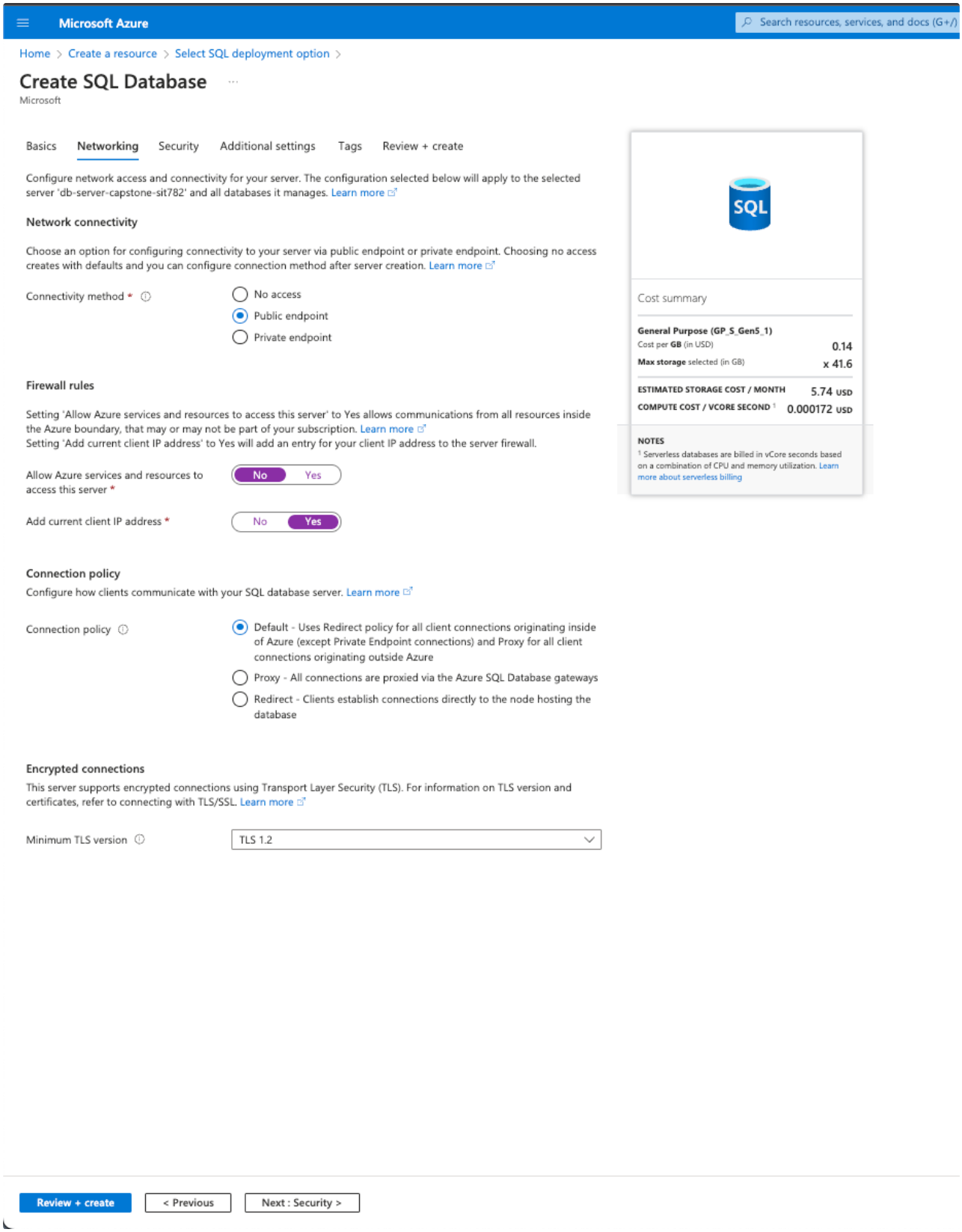


4. Under “Basics” tab, select the resource group name which you had created earlier during Azure Blob Storage creation. Next type in your database name (this is different than database server name). Select the database server which you named in the previous step next.

Under SQL Elastic pool, select “No”, since you will not need this in Capstone. Select Workload as Development and select Backup Storage Redundancy as LRS (similar to what was selected during Azure Blob Storage creation). Select “Networking” once all the above steps are completed to go to the next step.

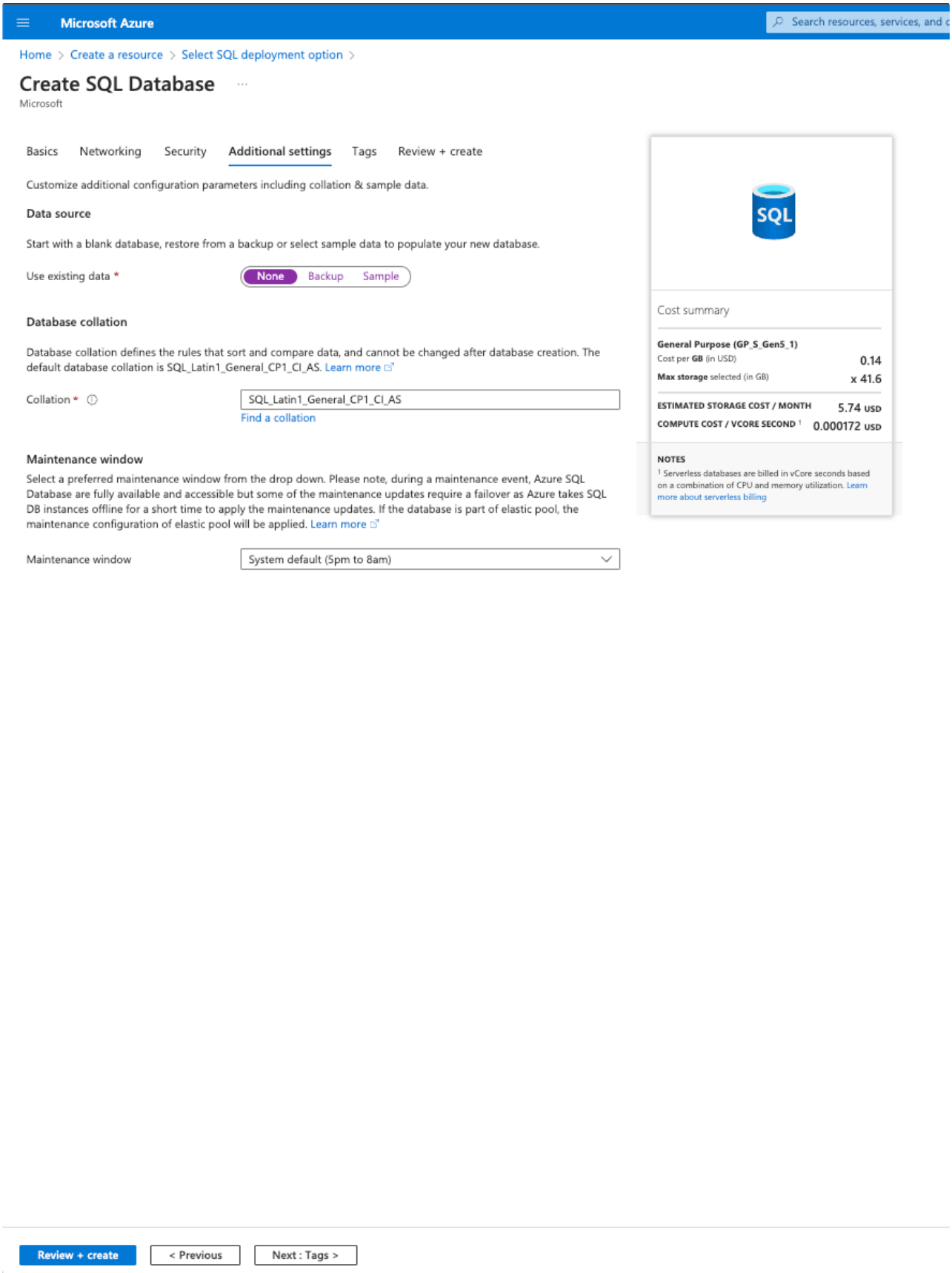


5. Under Networking, select Connectivity method as “Public endpoint” and under Firewall rules, select “No” for “Allow Azure services and resources to access this server” and “Yes” for “Add current client IP address“. The connection policy should be set as “Default” and under Encryption, select “Minimum TLS version” as “TLS 1.2”.



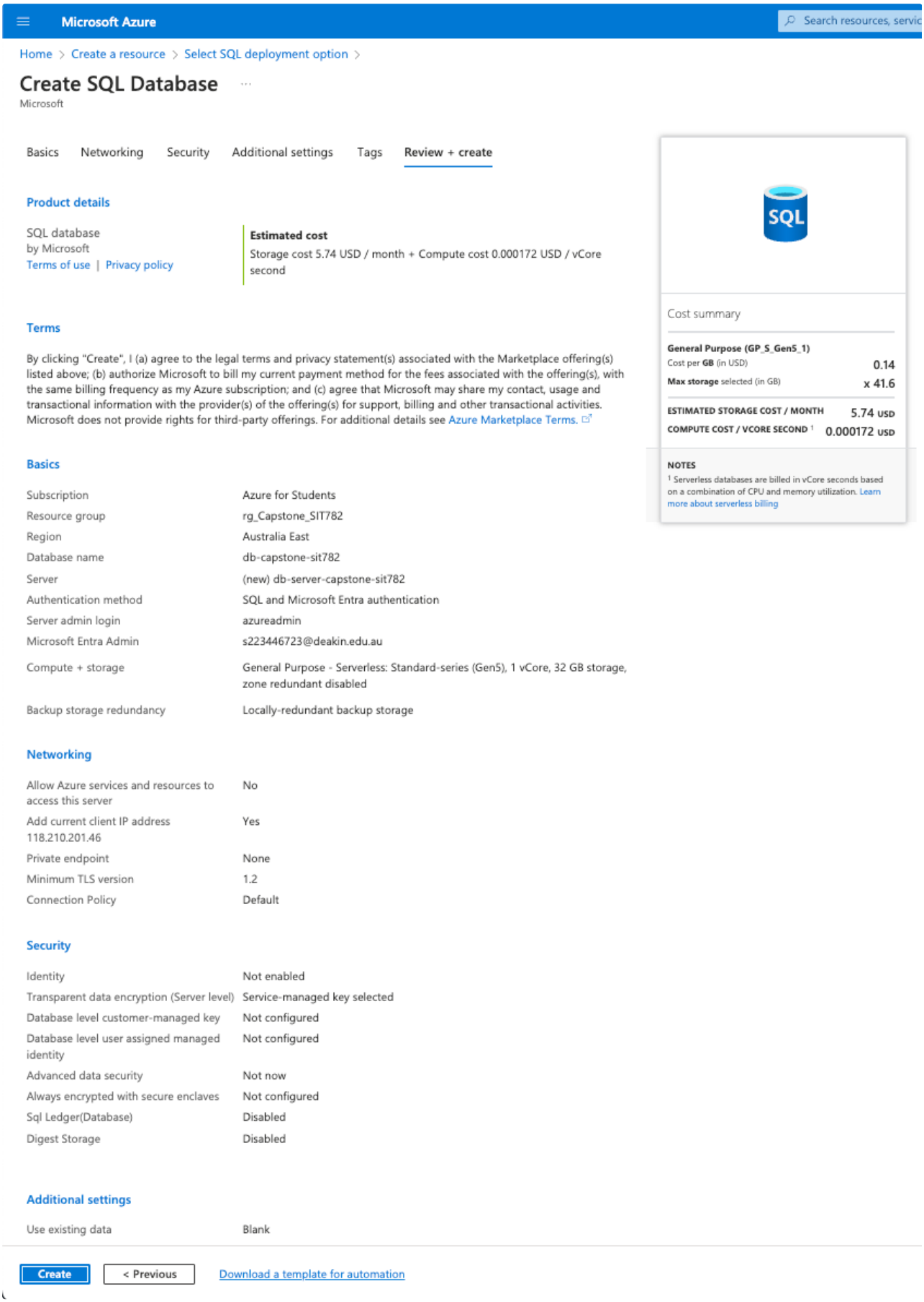
6. Select “Additional settings” next. Under Data source, select “None” as “Data source”. If you select “Sample”, it will create the AdventureWorks database, but this will also consume space in your new database. Use this if you are new to database development, but

is not required.

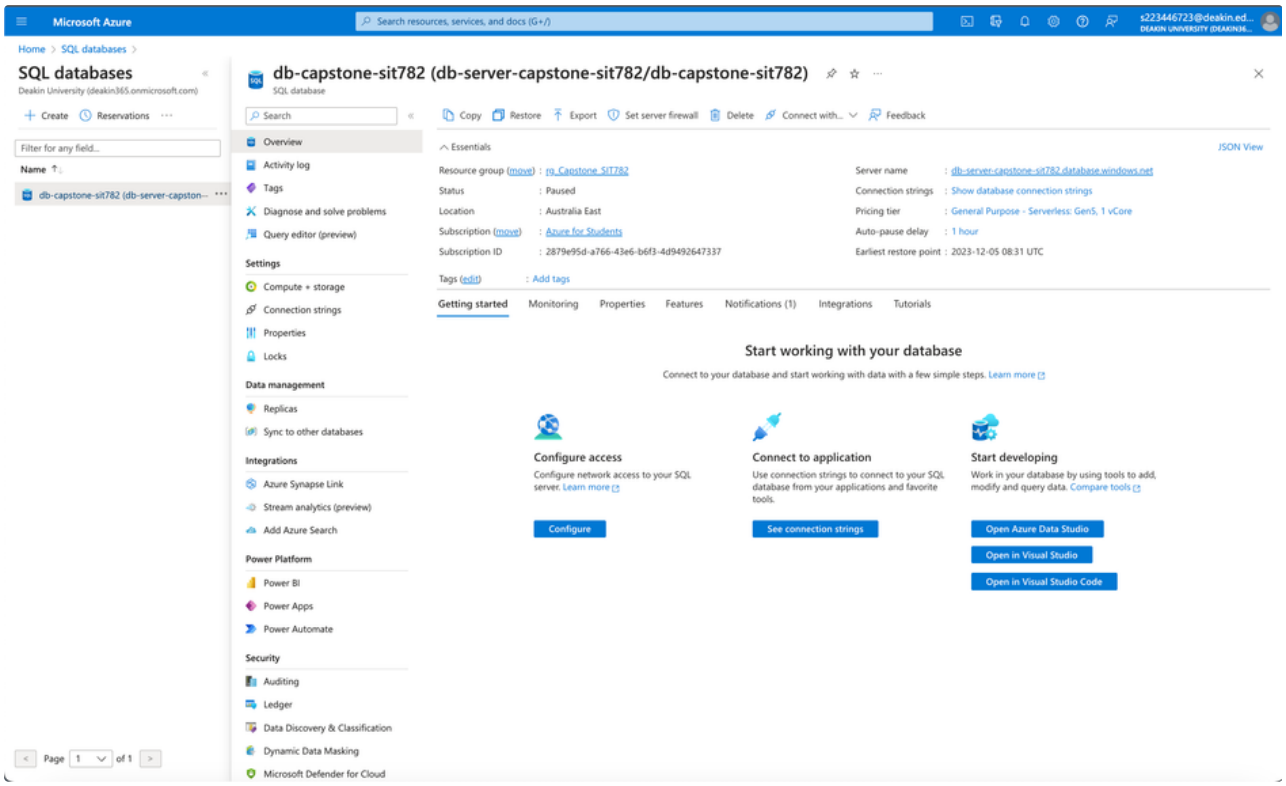


7. Click “Review and Create” next, which will validate the settings provided. After it has been completed, you will get a page similar to the one below. You will also be able to see the pricing for this new resource. Click “Create” and once it has been completed, click on “Go to

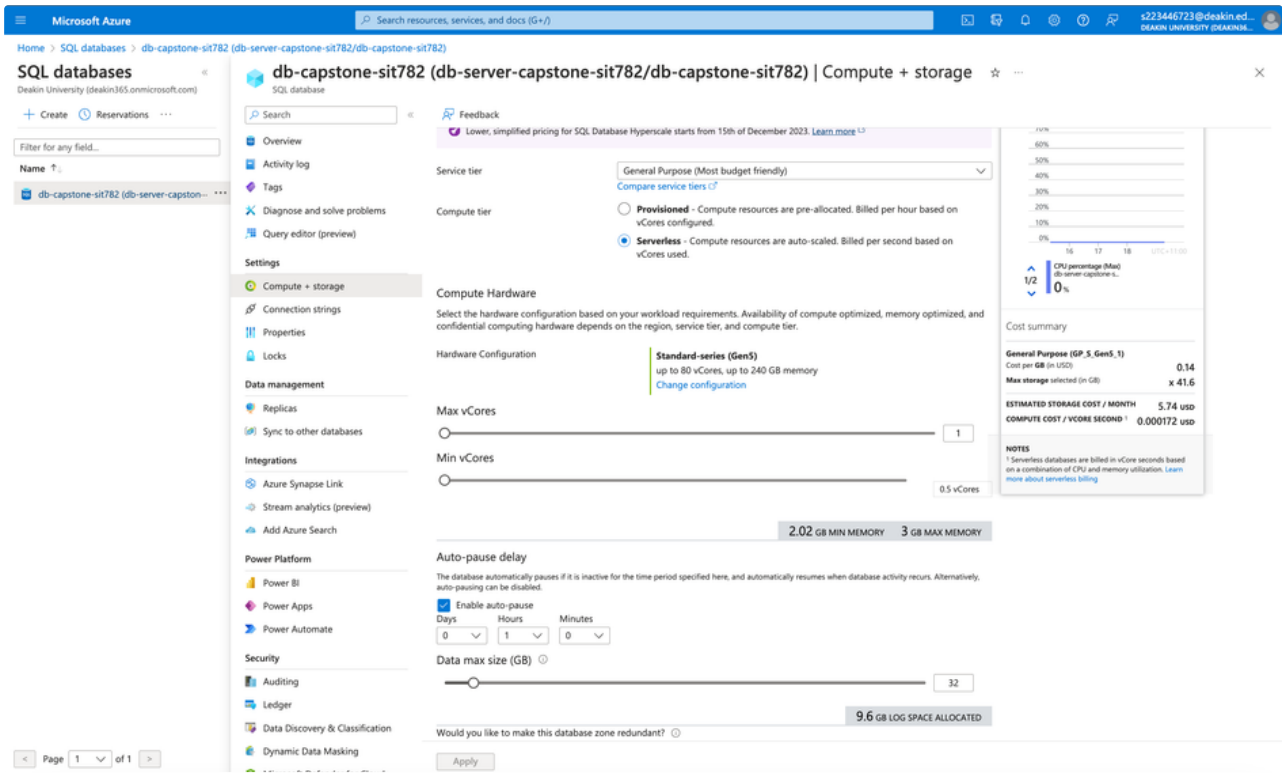
Resource” from the notification tab.



8. Once you click “Go to Resource”, you will land in the following page -

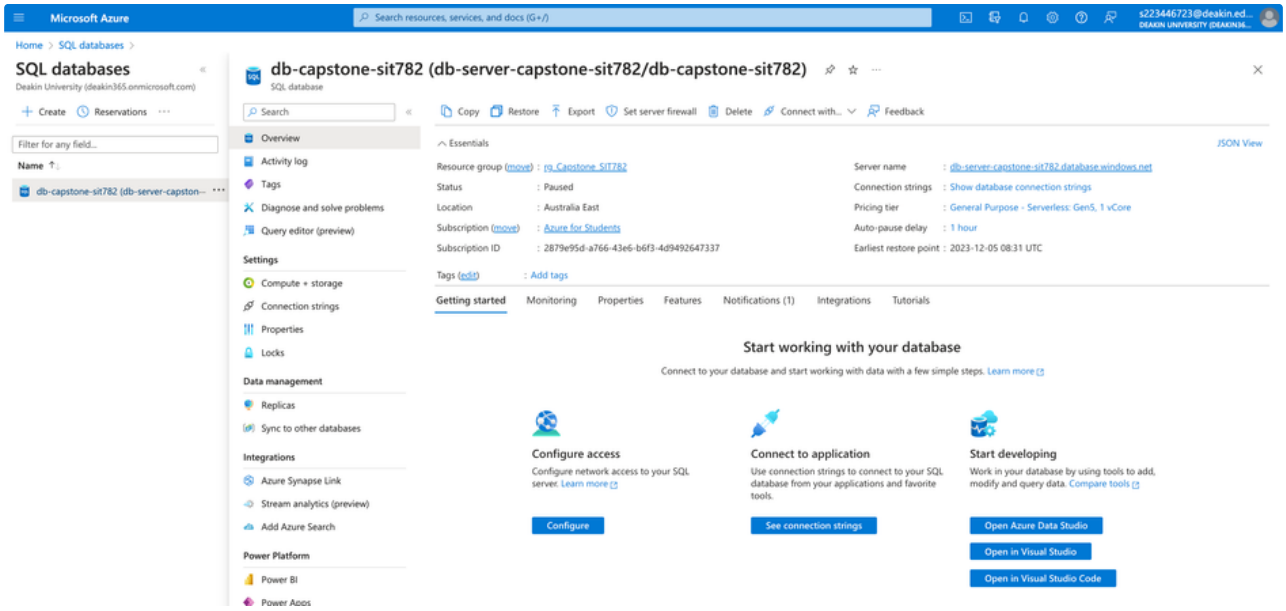


9. This page is the homepage where all changes to the newly created database server and database can be made. One of the critical options to select now is the “Auto-pause delay”. This will put the database to sleep if not used for 1 hour. This will help save you running cost for your database. To enable this, select “Compute + storage” on the left pane. Once you click this, you will land in a page like this -

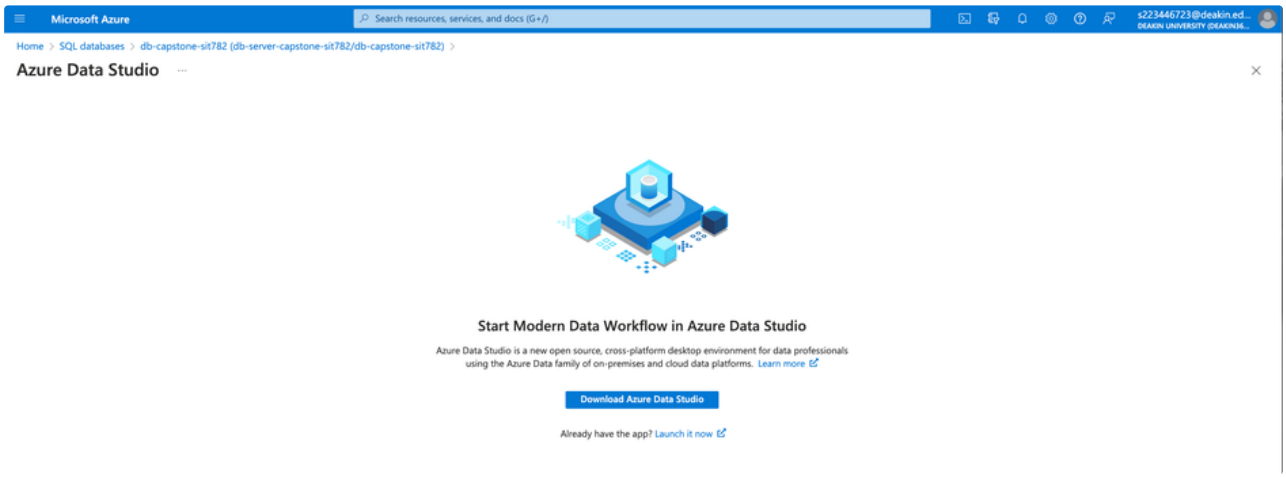


Here you can set min and max virtual cores for your database, the size of the database and most importantly, set the auto-pause delay.

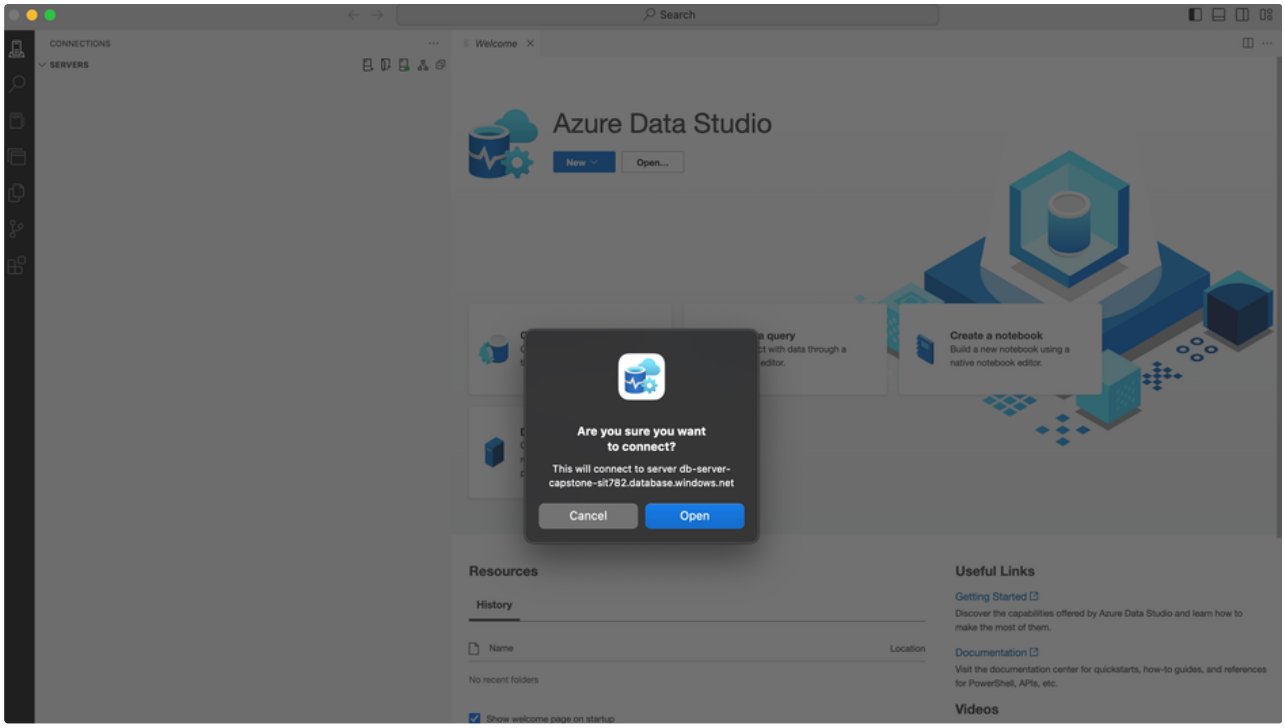
10. Once you have configured these settings, click on “Overview” to go back to the home page for the resource. Next, we will see how you can connect to this database which you just created. In the home page, under “Start developing”, click on “Open Azure Data Studio”. This will take you to a page where you can download this software to connect to the database created.



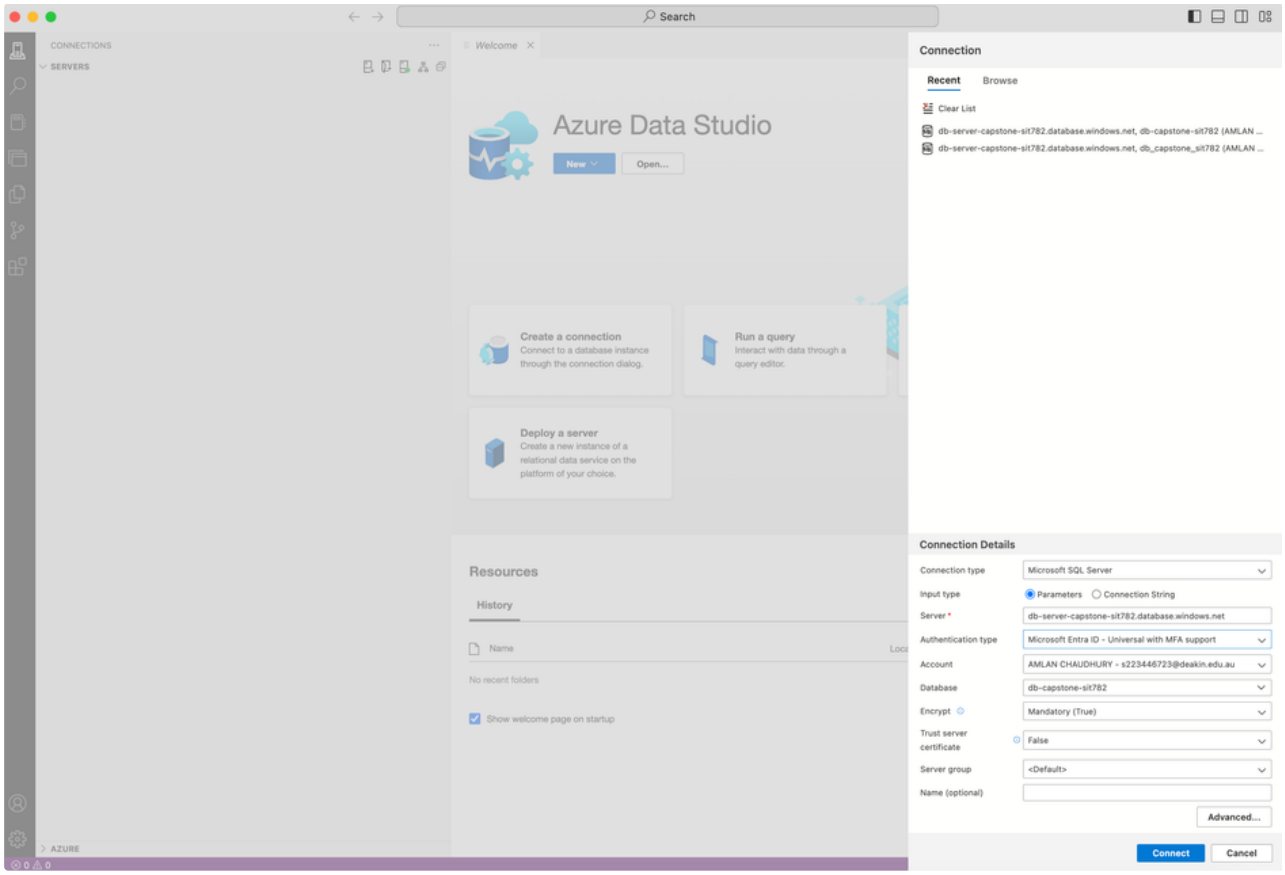
11. Click on “Launch it now”, after you have installed Azure Data Studio using the Download option in this page.



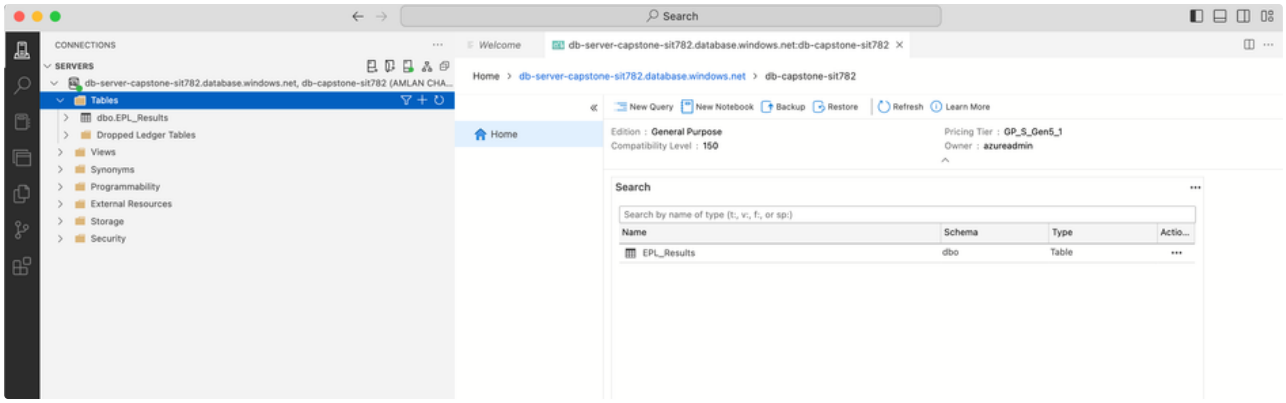
12. Clicking on “Launch it now” will open Azure Data Studio and you will get a page like this -



13. Click on “Open”. Next under “Authentication Type”, select “Microsoft Entra ID” and then authenticate to login. Click “Connect” to login.



14. After you have logged in successfully, you will see a page like this -



15. We will create a table next, using the script below. The structure of this table is aligned to the structure of the EPL Results which was created earlier (Azure Blob Storage - how to provision, access and development guide).

```
CREATE TABLE [dbo].[EPL_Results] (
    [Date_Of_Match]           VARCHAR (50) NULL,
    [Time_Of_Match]           VARCHAR (50) NULL,
    [Home_Team]               VARCHAR (50) NULL,
    [Away_Team]               VARCHAR (50) NULL,
    [Full_Time_Home_Goals]    INT          NULL,
    [Full_Time_Away_Goals]    INT          NULL,
    [Full_Time_Result]        CHAR (10)    NULL,
    [Halftime_Home_Goals]     INT          NULL,
    [Halftime_Away_Goals]     INT          NULL,
    [Halftime_Result]         CHAR (10)    NULL,
    [Referee]                 VARCHAR (50) NULL,
    [Home_Team_Shots]         INT          NULL,
    [Away_Team_Shots]         INT          NULL,
    [Home_Team_Shots_On_Goal] INT          NULL,
    [Away_Team_Shots_On_Goal] INT          NULL,
    [Home_Team_Fouls]         INT          NULL,
    [Away_Team_Fouls]         INT          NULL,
    [Home_Team_Corners]       INT          NULL,
    [Away_Team_Corners]       INT          NULL,
    [Home_Team_Yellow_Cards]  INT          NULL,
    [Away_Team_Yellow_Cards]  INT          NULL,
    [Home_Team_Red_Cards]     INT          NULL,
    [Away_Team_Red_cards]     INT          NULL
);

```

If you have followed all of the steps one after the other, then you will be able to create your database server, database and also will be able to connect to it from your local machine.