

AZ-204

Developing solutions for Microsoft Azure

Lab 04- Part 1

Constructing a polyglot data solution

Table of Contents

| | |
|--|----|
| Constructing a polyglot data solution | 1 |
| 1 Pre-requisites | 3 |
| 1.1 Sign in to the lab virtual machine | 3 |
| 1.2 Review the installed applications | 3 |
| 2 Exercise 1: Creating database resources in Azure | 4 |
| 2.1 Task 1: Open the Azure portal | 4 |
| 2.2 Task 2: Create an Azure SQL Database server resource | 4 |
| 2.3 Task 3: Create an Azure Cosmos DB account resource..... | 7 |
| 2.4 Task 4: Create an Azure Storage account resource | 10 |
| 2.5 Review | 11 |
| 3 Exercise 2: Import and validate data | 12 |
| 3.1 Task 1: Upload image blobs | 12 |
| 3.2 Task 2: Upload an SQL .bacpac file | 14 |
| 3.3 Task 3: Import an SQL database..... | 15 |
| 3.4 Task 4: Use an imported SQL database..... | 18 |
| 3.5 Review | 20 |
| 4 Exercise 3: Open and configure a .NET web application | 21 |
| 4.1 Task 1: Open and build the web application..... | 21 |
| 4.2 Task 2: Update the SQL connection string..... | 21 |
| 4.3 Task 3: Update the blob base URL | 22 |
| 4.4 Task 4: Validate the web application | 22 |
| 4.5 Review | 23 |

1 Pre-requisites

1.1 Sign in to the lab virtual machine

Sign in to your Windows 10 virtual machine (VM) by using the following credentials:

- Username: **Admin**
- Password: **Pa55w.rd**

Note: Instructions to connect to the virtual lab environment will be provided by your instructor.

1.2 Review the installed applications

Find the taskbar on your Windows 10 desktop. The taskbar contains the icons for the applications that you'll use in this lab:

- Microsoft Edge
- File Explorer
- Azure CLI
- Windows PowerShell

2 Exercise 1: Creating database resources in Azure

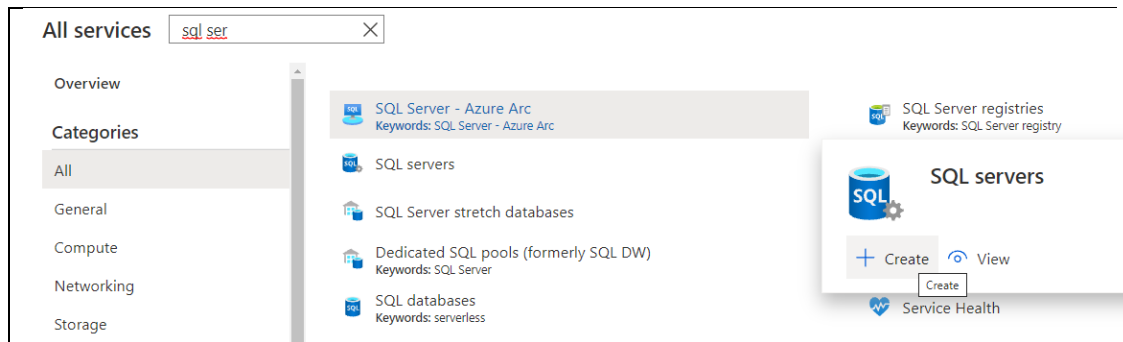
2.1 Task 1: Open the Azure portal

1. On the taskbar, select the **Microsoft Edge** icon.
2. In the open browser window, browse to the Azure portal (portal.azure.com).
3. Enter the email address for your Microsoft account, and then select **Next**.
4. Enter the password for your Microsoft account, and then select **Sign in**.

Note: If this is your first time signing in to the Azure portal, you will be offered a tour of the portal. Select **Get Started** to skip the tour and begin using the portal.

2.2 Task 2: Create an Azure SQL Database server resource

1. In the Azure portal's navigation pane, select **All services**.
2. From the **All services** blade, select **SQL servers**.
3. From the **SQL servers** blade, find your list of SQL server instances.
4. From the **SQL servers** blade, select **+ Create**.



5. From the **Create SQL Database Server** blade, observe the tabs from the blade, such as **Basics**, **Networking**, and **Additional settings**.

Note: Each tab represents a step in the workflow to create a new Azure SQL Database server. You can select **Review + Create** at any time to skip the remaining tabs.

1. From the **Basics** tab, perform the following actions:
2. Leave the **Subscription** drop-down list set to its default value.
3. In the **Resource group** section, select **Create new**, enter **PolyglotData**, and then select **OK**.
4. In the **Server name** text box, enter **polysqlsrvr[yourname]**.
5. In the **Location** drop-down list, select **(US) East US**.
6. In the **Server admin login** text box, enter **testuser**.
7. In the **Password** text box, enter **TestPa55w.rd**.
8. In the **Confirm password** text box, enter **TestPa55w.rd** again.
9. Select **Next: Networking**.

Microsoft Azure

Search resources, services, and docs (G+/)

[All services >](#)

Create SQL Database Server

Microsoft

[Create new](#)

Server details

Enter required settings for this server, including providing a name and location.

Server name *

polysqlsrvsrini ✓

.database.windows.net

Location *

(US) East US ▼

Authentication

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Azure AD authentication [Learn more](#) using an existing Azure AD user, group, or application as Azure AD admin [Learn more](#), or select both SQL and Azure AD authentication.

Authentication method

☒ Use SQL Authentication
 ☐ Use only Azure Active Directory (Azure AD) authentication
 ☐ Use both SQL and Azure AD Authentication

Server admin login *

testuser ✓

Password *

..... ✓

Confirm password *

..... ✓

Review + create

Next : Networking >

6. From the **Networking** tab, perform the following actions:
- In the **Allow Azure services and resources to access this server** section, select **Yes**.
 - Select **Review + Create**.

All services >

Create SQL Database Server

Microsoft

Basics

Networking

Additional settings

Tags

Review + create

Configure networking access for your server.

Firewall rules

Allow Azure services and resources to access this server ⓘ

Yes

No

Review + create

< Previous

Next : Additional settings >

7. From the **Review + Create** tab, review the options that you selected during the previous steps.

Create SQL Database Server

Microsoft

Basics Networking Additional settings Tags Review + create

Product details

SQL Database Server
by Microsoft
[Terms of use](#) | [Privacy policy](#)

Estimated cost per month
No additional charges

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace and agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the product.

Basics

| | |
|-----------------------|------------------------|
| Subscription | Pay-As-You-Go Dev/Test |
| Resource group | PolyglotData |
| Server name | polysqlsvrsrini |
| Authentication method | SQL Authentication |
| Server admin login | testuser |
| Location | East US |

Networking

| | |
|---------------------------------------|-----|
| Allow Azure services to access server | Yes |
|---------------------------------------|-----|

Additional settings

| | |
|-------------------------------|---------|
| Enable Azure Defender for SQL | Not now |
|-------------------------------|---------|

Create

< Previous

[Download a template for automation](#)

8. Select **Create** to create the SQL Database server by using your specified configuration.

Note: At this point in the lab, we are only creating the Azure SQL logical server. We will create the Azure SQL database instance later in the lab.

Note: Wait for the creation task to complete before you move forward with this lab.

Delete
 Cancel
 Redeploy
 Refresh

We'd love your feedback! →

... Deployment is in progress

Deployment name: Microsoft.SqlServer.createServer_aa0bb971e10... Start time: 10/5/2021, 5:27:33 AM
 Subscription: Pay-As-You-Go Dev/Test Correlation ID: 30830d04-3fee-4fb6-a955-7fdf21262478
 Resource group: PolyglotData

Deployment details [\(Download\)](#)

| Resource | Type | Status | Operation details |
|-----------------|-----------------------|----------|-----------------------------------|
| polysqlsvrsrini | Microsoft.Sql/servers | Accepted | Operation details |

2.3 Task 3: Create an Azure Cosmos DB account resource

1. In the Azure portal's navigation pane, select **All services**.
2. From the **All services** blade, select **Azure Cosmos DB**.
3. From the **Azure Cosmos DB** blade, observe your list of Azure Cosmos DB instances.
4. From the **Azure Cosmos DB** blade, select **+ Create**.

5. From the **Select API option** blade, select **Core (SQL) - Recommended**.
6. From the **Create Azure Cosmos DB Account** blade, observe the tabs from the blade, such as **Basics**, **Network**, and **Tags**.

Note: Each tab represents a step in the workflow to create a new Azure Cosmos DB account. You can select **Review + Create** at any time to skip the remaining tabs.

7. From the **Basics** tab, perform the following actions:
 1. Leave the **Subscription** list set to its default value.
 2. In the **Resource group** section, select **PolyglotData** from the list.
 3. In the **AccountName** text box, enter **polycosmos[yourname]**.
 4. In the **Location** drop-down list, select the **(US) East US** region.
 5. In the **Capacity mode** section, select **Serverless**.

The screenshot shows the 'Create Azure Cosmos DB Account - Core (SQL)' wizard in the 'Basics' tab. The interface includes a breadcrumb trail at the top: Basics, Global Distribution, Networking, Backup Policy, Encryption, Tags, and Review + create. Below this is a brief description of Azure Cosmos DB. The 'Project Details' section prompts the user to select a subscription and resource group. The 'Subscription' dropdown is set to 'Pay-As-You-Go Dev/Test', and the 'Resource Group' dropdown is set to 'PolyglotData'. The 'Instance Details' section contains the 'Account Name' field with the value 'polycosmosrini', the 'Location' dropdown set to '(US) East US', and the 'Capacity mode' section where 'Serverless' is selected with a radio button. At the bottom, there are three buttons: 'Review + create' (highlighted in blue), 'Previous', and 'Next: Global Distribution'.

Create Azure Cosmos DB Account - Core (SQL) ...

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. [Try it for f](#)

Project Details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage al

Subscription * Pay-As-You-Go Dev/Test

Resource Group * PolyglotData
[Create new](#)

Instance Details

Account Name * polycosmosrini

Location * (US) East US

Capacity mode ☐ Provisioned throughput ☒ Serverless
[Learn more about capacity mode](#)

[Review + create](#) Previous Next: Global Distribution

8. Select **Review + Create**.
9. From the **Review + Create** tab, review the options that you selected during the previous steps.
10. Select **Create** to create the Azure Cosmos DB account by using your specified configuration.

Create Azure Cosmos DB Account - Core (SQL) ...

✓ Validation Success

[Basics](#)
[Global Distribution](#)
[Networking](#)
[Backup Policy](#)
[Encryption](#)
[Tags](#)
[Review + create](#)

Creation Time

Estimated Account Creation Time (in minutes) 2

i The estimated creation time is calculated based on the location you have selected

Basics

| | |
|--------------------|------------------------|
| Subscription | Pay-As-You-Go Dev/Test |
| Resource Group | PolyglotData |
| Location | East US |
| Account Name | (new) polycosmossrini |
| API | Core (SQL) |
| Capacity mode | Serverless |
| Availability Zones | Disable |

Backup Policy

| | |
|---------------------------|------------------------------|
| Backup policy | Periodic |
| Backup storage redundancy | Geo-redundant backup storage |

Networking

| | |
|---------------------|--------------|
| Connectivity method | All networks |
|---------------------|--------------|

Create

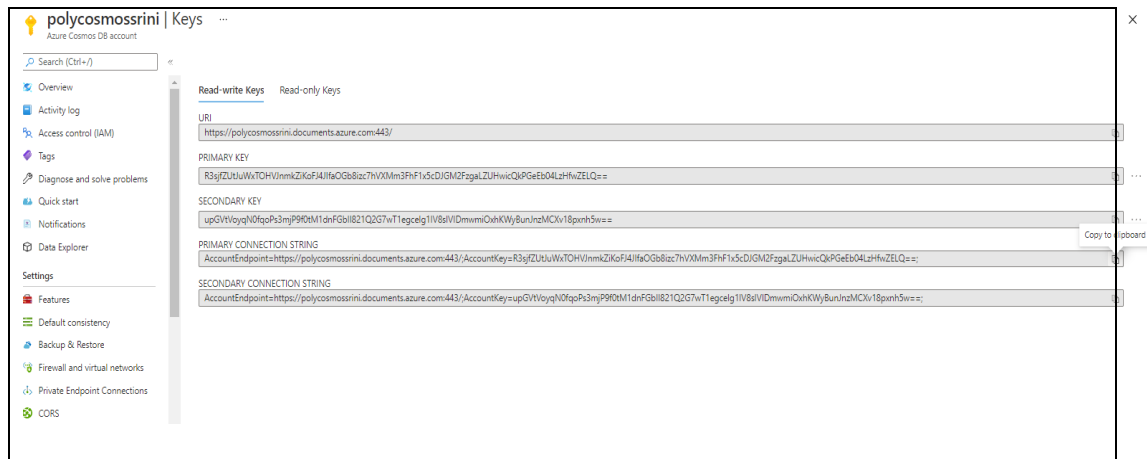
Previous

Next

[Download a template for automation](#)

Note: Wait for the creation task to complete before you move forward with this lab.

11. In the Azure portal's navigation pane, select the **Resource groups** link.
12. From the **Resource groups** blade, find and then select the **PolyglotData** resource group that you created earlier in this lab.
13. From the **PolyglotData** blade, select the **polycosmos[yourname]** Azure Cosmos DB account that you created earlier in this lab.
14. From the **Azure Cosmos DB account** blade, find the **Settings** section from the blade, and then select the **Keys** link.
15. In the Keys pane, record the value in the **PRIMARY CONNECTION STRING** text box. You'll use this value later in this lab.



2.4 Task 4: Create an Azure Storage account resource

1. In the Azure portal's navigation pane, select **All services**.
2. From the **All services** blade, select **Storage Accounts**.
3. From the **Storage accounts** blade, find your list of Storage instances.
4. From the **Storage accounts** blade, select **New**.
5. From the **Create storage account** blade, observe the tabs from the blade, such as **Basics**, **Advanced**, and **Tags**.

Note: Each tab represents a step in the workflow to create a new Azure Storage account. You can select **Review + Create** at any time to skip the remaining tabs.

6. From the **Basics** tab, perform the following actions:
 1. Leave the **Subscription** list set to its default value.
 2. In the **Resource group** section, select **PolyglotData** from the list.
 3. In the **Storage account name** text box, enter **polystor[yourname]**.
 4. In the **Region** drop-down list, select the **(US) East US** region.
 5. In the **Performance** section, select **Standard**.
 6. In the **Redundancy** drop-down list, select **Locally-redundant storage (LRS)**.
 7. Select **Review + Create**.
7. From the **Review + Create** tab, review the options that you selected during the previous steps.

[All services](#) >

Create a storage account

Basics Advanced Networking Data protection Tags Review + create

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *

Resource group *
[Create new](#)

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name ⓘ *

Region ⓘ *

Performance ⓘ *

☒ Standard: Recommended for most scenarios (general-purpose v2 account)

☐ Premium: Recommended for scenarios that require low latency.

Redundancy ⓘ *

[Review + create](#) [< Previous](#) [Next : Advanced >](#)

8. Select **Create** to create the storage account by using your specified configuration.

Note: Wait for the creation task to complete before you move forward with this lab.

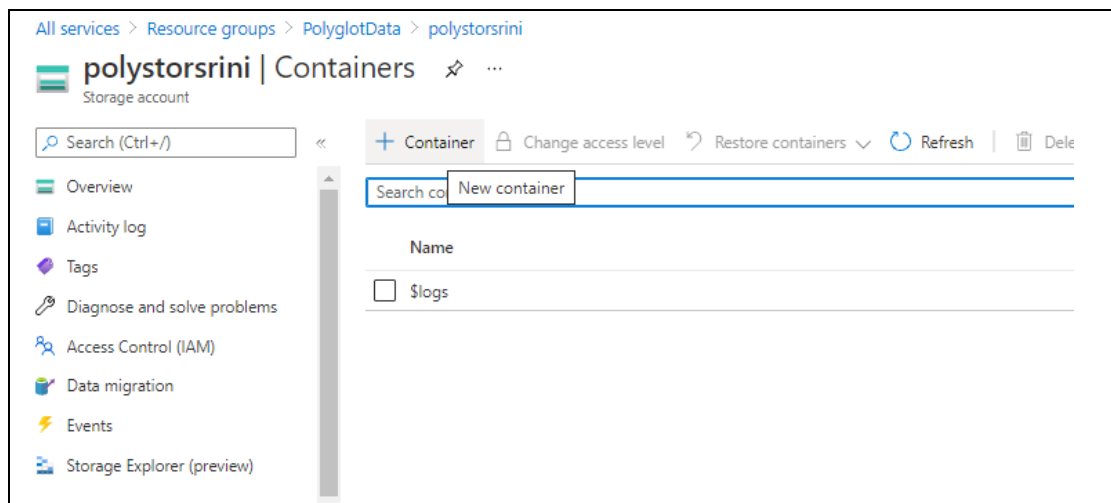
2.5 Review

In this exercise, you created all the Azure resources that you'll need for a polyglot data solution.

3 Exercise 2: Import and validate data

3.1 Task 1: Upload image blobs

1. In the Azure portal's navigation pane, select the **Resource groups** link.
2. From the **Resource groups** blade, find and select the **PolyglotData** resource group that you created earlier in this lab.
3. From the **PolyglotData** blade, select the **polystor[yourname]** storage account that you created earlier in this lab.
4. From the **Storage account** blade, select the **Containers** link in the **Data storage** section from the blade.
5. In the **Containers** section, select **+ Container**.



6. In the **New container** pop-up window, perform the following actions:
 1. In the **Name** text box, enter **images**.
 2. In the **Public access level** drop-down list, select **Blob (anonymous read access for blobs only)**.
 3. Select **Create**.

New container ✕

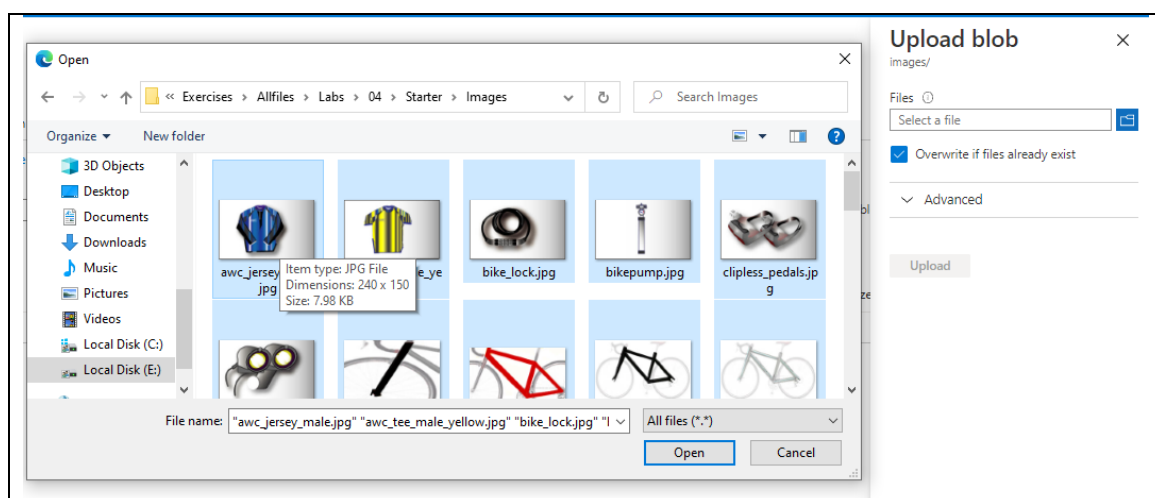
Name * ✓

Public access level ⓘ
 ▼

⚠ Blobs within the container can be read by anonymous request, but container data is not available. Anonymous clients cannot enumerate the blobs within the container.

▼ Advanced

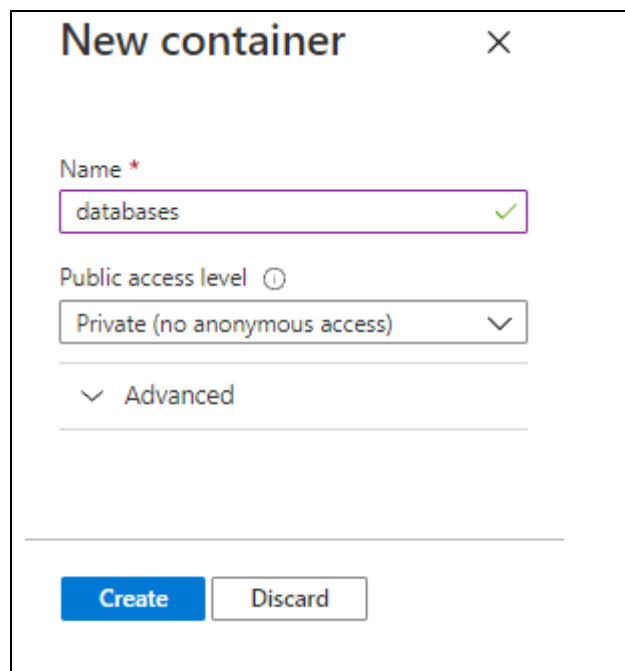
7. Back in the **Containers** section, select the newly created **images** container.
8. From the **Container** blade, find the **Settings** section from the blade, and then select the **Properties** link.
9. In the Properties pane, record the value in the **URL** text box. You'll use this value later in this lab.
10. Find and select the **Overview** link from the blade.
11. From the blade, select **Upload**.
12. In the **Upload blob** pop-up, perform the following actions:
 1. In the **Files** section, select the **Folder** icon.
 2. In the **File Explorer** window, browse to **Allfiles (F):\Allfiles\Labs\04\Starter\Images**, select all 42 individual **.jpg** image files, and then select **Open**.
 3. Ensure that **Overwrite if files already exist** is selected, and then select **Upload**.



Note: Wait for all the blobs to upload before you continue with this lab.

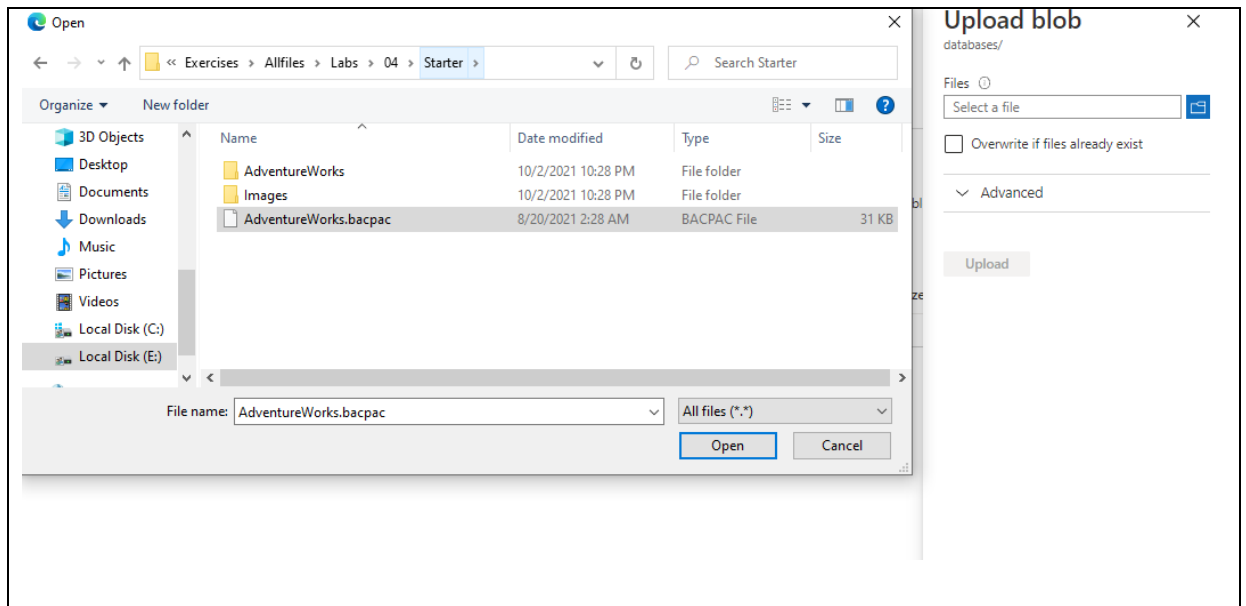
3.2 Task 2: Upload an SQL .bacpac file

1. In the Azure portal's navigation pane, select the **Resource groups** link.
2. From the **Resource groups** blade, find and select the **PolyglotData** resource group that you created earlier in this lab.
3. From the **PolyglotData** blade, select the **polystor[yourname]** storage account that you created earlier in this lab.
4. From the **Storage account** blade, select the **Containers** link in the **Data storage** section from the blade.
5. In the **Containers** section, select **+ Container**.
6. In the **New container** pop-up, perform the following actions:
 1. In the **Name** text box, enter **databases**.
 2. In the **Public access level** drop-down list, select **Private (no anonymous access)**.



The screenshot shows the 'New container' dialog box. The 'Name' field is filled with 'databases' and has a green checkmark. The 'Public access level' dropdown is set to 'Private (no anonymous access)'. There is an 'Advanced' section with a chevron icon. At the bottom are 'Create' and 'Discard' buttons.

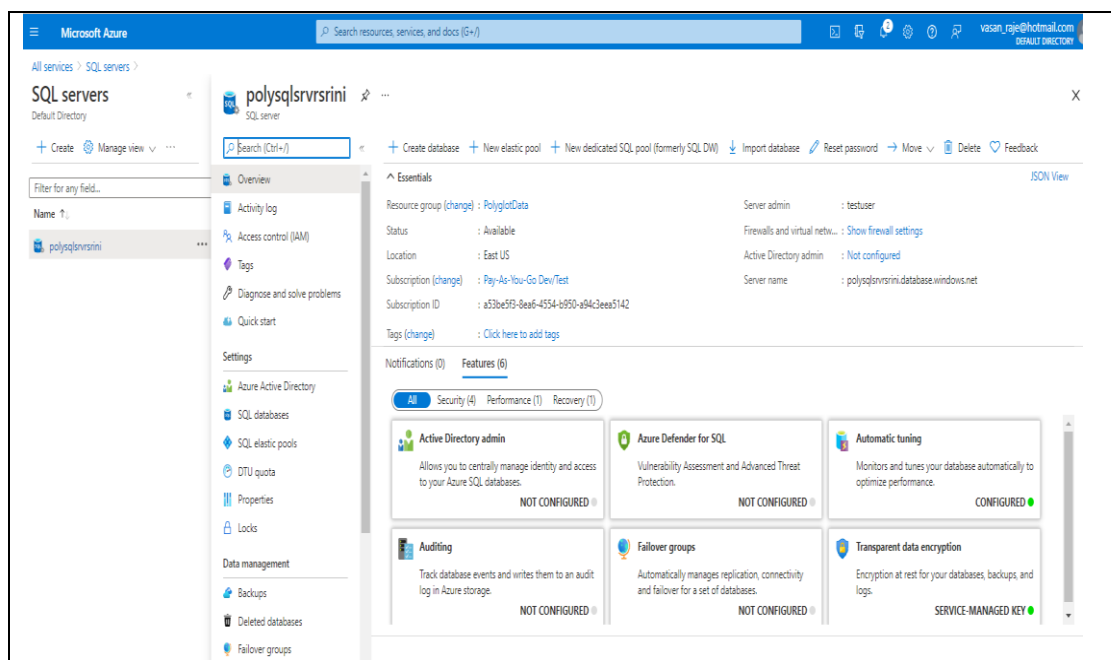
3. Select **Create**.
7. Back in the **Containers** section, select the newly created **databases** container.
8. From the **Container** blade, select **Upload**.
9. In the **Upload blob** pop-up, perform the following actions:
 1. In the **Files** section, select the **Folder** icon.
 2. In the **File Explorer** window, browse to **Allfiles (F):\Allfiles\Labs\04\Starter**, select the **AdventureWorks.bacpac** file, and then select **Open**.
 3. Ensure that **Overwrite if files already exist** is selected, and then select **Upload**.



Note: Wait for the blob to upload before you continue with this lab.

3.3 Task 3: Import an SQL database

1. In the Azure portal's navigation pane, select the **Resource groups** link.
2. From the **Resource groups** blade, find and select the **PolyglotData** resource group that you created earlier in this lab.
3. From the **PolyglotData** blade, select the **polysqlsrvr[yourname]** SQL server that you created earlier in this lab.
4. From the **SQL server** blade, select **Import database**.



5. From the **Import database** blade, perform the following actions:
 1. Leave the **Subscription** list set to its default value.
 2. In the **Storage** section, select **Select backup**.
 3. From the **Storage accounts** blade, select the **polystor[yourname]** storage account that you created earlier in this lab.
 4. From the **Containers** blade, select the **databases** container that you created earlier in this lab.
 5. From the **Container** blade, select the **AdventureWorks.bacpac** blob that you created earlier in this lab, and then select **Select** to close the blade.

Microsoft Azure

Search resources, services, and docs (G+)

All services > SQL servers > polysqlsvrsrini > Import database > Storage accounts > Containers > databases

databases ...

Container

Upload Refresh

Authentication method: Access key (Switch to Azure AD User Account)

Location: databases

Search blobs by prefix (case-sensitive)

Add filter

| Name | Modified |
|-----------------------|----------------|
| AdventureWorks.bacpac | 10/5/2021, 6:1 |

Select

6. Back from the **Import database** blade, leave the **Pricing tier** option set to its default value.
7. In the **Database name** text box, enter **AdventureWorks**.
8. Leave the **Collation** text box set to its default value.
9. In the **Server admin login** text box, enter **testuser**.
10. In the **Password** text box, enter **TestPa55w.rd**.

[All services](#) > [SQL servers](#) > [polysqlsrvsrini](#) >

↓ Import database ...

polysqlsrvsrini

Subscription *

Pay-As-You-Go Dev/Test

Storage (Premium not supported) *

AdventureWorks.bacpac
polystorsrini/databases
[Select backup](#)

Pricing tier * ⓘ

General Purpose
Gen5, 2 vCores, 32 GB storage, zone
redundant disabled
[Configure database](#)

Database name

AdventureWorks ✓

Collation * ⓘ

SQL_Latin1_General_CP1_CI_AS

Authentication type

SQL Server

Server admin login *

testuser

*Password

..... ✓

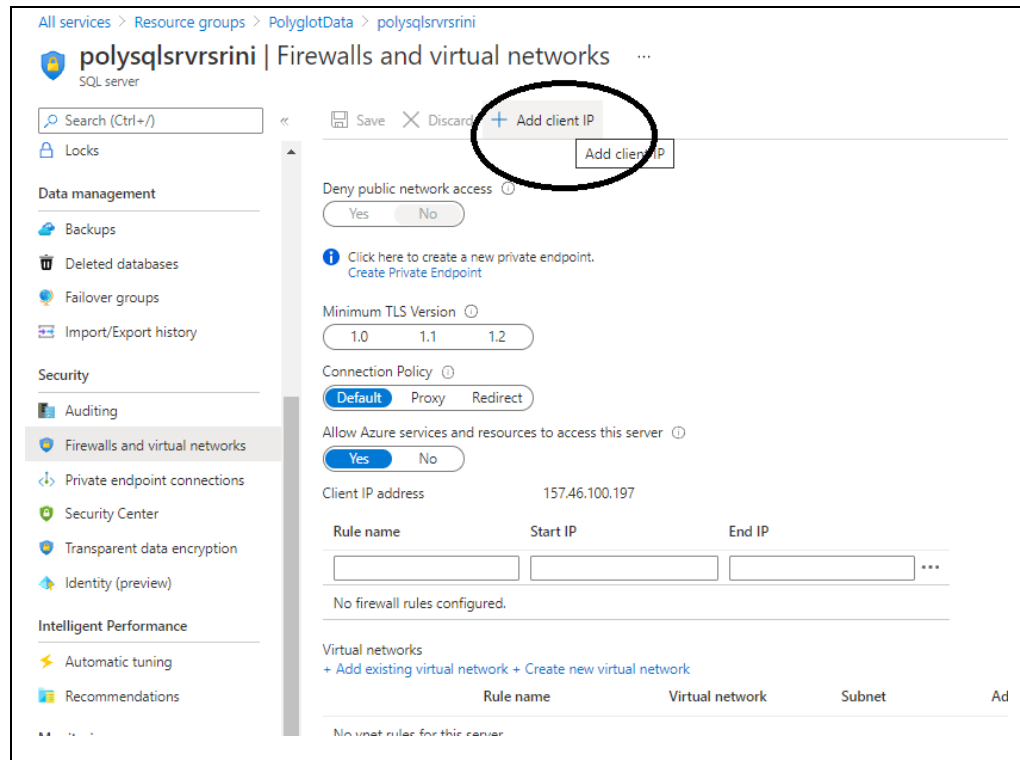
OK

11. Select **OK**.

Note: Wait for the database to be created before you continue with this lab. If you receive a firewall-related error on the import step, it means you did not correctly configure the **Allow Azure services to access server** setting on your SQL Server earlier in the lab. Review your settings, delete the empty **AdventureWorks** database, and then attempt your import again.

3.4 Task 4: Use an imported SQL database

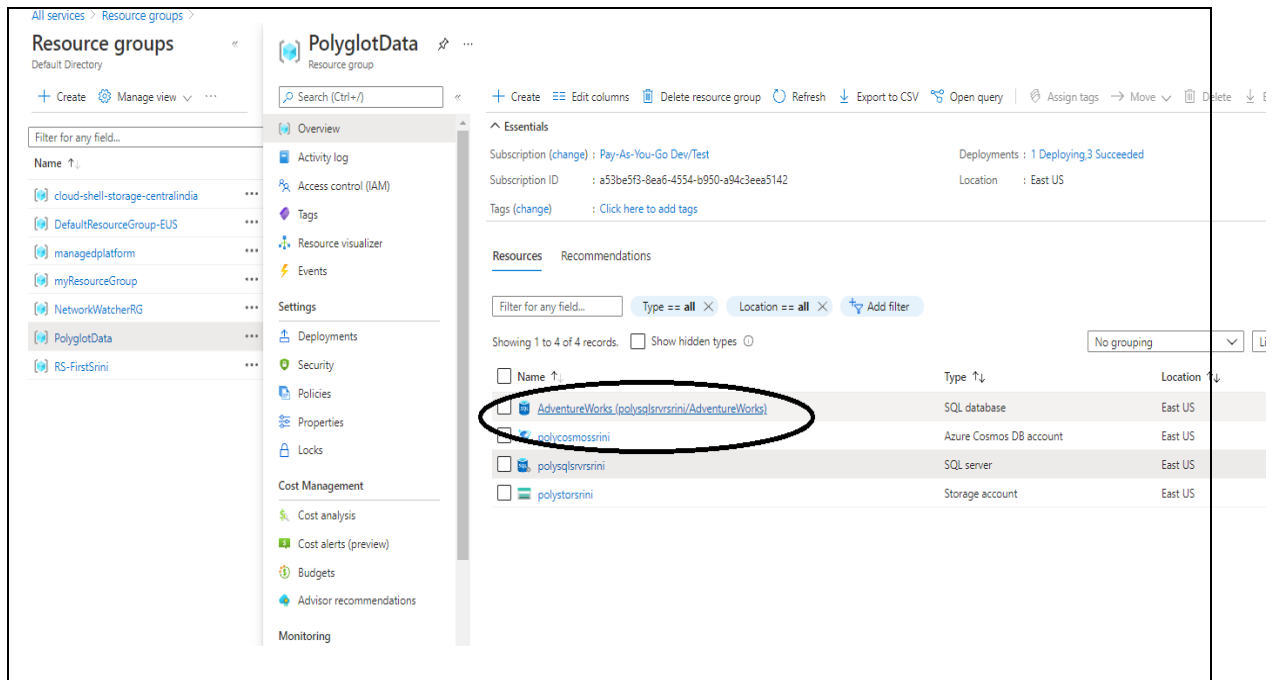
1. In the Azure portal's navigation pane, select the **Resource groups** link.
2. From the **Resource groups** blade, find and select the **PolyglotData** resource group that you created earlier in this lab.
3. From the **PolyglotData** blade, select the **polysqlsrvr[yourname]** SQL server that you created earlier in this lab.
4. From the **SQL server** blade, find the **Security** section from the blade, and then select the **Firewalls and virtual networks** link.
5. In the Firewalls and virtual networks pane, perform the following actions:
 1. Select **Add client IP**



2. Select **Save**.
3. In the **Success!** confirmation dialog, select **OK**.

Note: This step will ensure that your local machine will have access to the databases that are associated with this server.

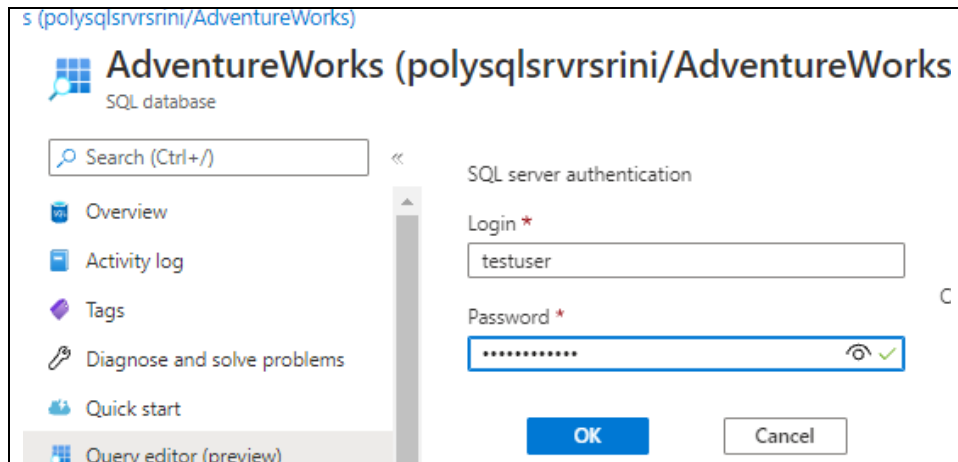
6. In the Azure portal's navigation pane, select the **Resource groups** link.
7. From the **Resource groups** blade, find and select the **PolyglotData** resource group that you created earlier in this lab.
8. From the **PolyglotData** blade, select the **AdventureWorks** SQL database that you created earlier in this lab.



9. From the **SQL database** blade, find the **Settings** section from the blade, and then select the **Connection strings** link.
10. In the Connection strings pane, record the value in the **ADO.NET (SQL Authentication)** text box. You'll use this value later in this lab.
11. Update the connection string that you recorded by performing the following actions:
 1. Within the connection string, find the *your_username* placeholder and replace it with **testuser**.
 2. Within the connection string, find the *your_password* placeholder and replace it with **TestPa55w.rd**.

Note: For example, if your connection string was originally
 Server=tcp:polysqlsvrinstructor.database.windows.net,1433;Initial
 Catalog=AdventureWorks;User ID={your_username};Password={your_password};,
 your updated connection string will be
 Server=tcp:polysqlsvrinstructor.database.windows.net,1433;Initial
 Catalog=AdventureWorks;User ID=testuser;Password=TestPa55w.rd;

12. Find and select the **Query editor (preview)** link from the blade.
13. In the Query editor pane, perform the following actions:
 1. In the **Login** text box, enter **testuser**.
 2. In the **Password** text box, enter **TestPa55w.rd**.



3. Select **OK**.
14. In the open query editor, enter the following query:

```
SELECT * FROM AdventureWorks.dbo.Models
```

15. Select **Run** to run the query, and then observe the results.

Note: This query will return a list of models from the home page of the web application.

16. In the query editor, replace the existing query with the following query:

```
SELECT * FROM AdventureWorks.dbo.Products
```

16. Select **Run** to run the query, and then observe the results.

Note: This query will return a list of products that are associated with each model.

3.5 Review

In this exercise, you imported all the resources that you'll use with your web application.

4 Exercise 3: Open and configure a .NET web application

4.1 Task 1: Open and build the web application

1. On the **Start** screen, select the **Visual Studio Code** tile.
2. From the **File** menu, select **Open Folder**.
3. In the **File Explorer** window that opens, browse to **Allfiles (F):\Allfiles\Labs\04\Starter\AdventureWorks**, and then select **Select Folder**.
4. In the **Visual Studio Code** window, right-click or activate the shortcut menu for the Explorer pane, and then select **Open in Terminal**.
5. At the open command prompt, enter the following command, and then select Enter to build the .NET web application:

`dotnet build`

5. **Note:** The **dotnet build** command will automatically restore any missing NuGet packages prior to building all projects in the folder.
6. Observe the results of the build printed in the terminal. The build should complete successfully with no errors or warning messages.
7. Select **Kill Terminal** or the **Recycle Bin** icon to close the currently open terminal and any associated processes.

4.2 Task 2: Update the SQL connection string

1. In the Explorer pane of the **Visual Studio Code** window, expand the **AdventureWorks.Web** project.
2. Open the **appsettings.json** file.
3. In the JavaScript Object Notation (JSON) object on line 3, find the **ConnectionStrings.AdventureWorksSqlContext** path. Observe that the current value is empty:

```
{
  "ConnectionStrings": {
    "AdventureWorksSqlContext": "",
    "AdventureWorksCosmosContext": ""
  },
}
```

- 3.
4. Update the value of the **AdventureWorksSqlContext** property by setting its value to the **ADO.NET (SQL Authentication) connection string** of the SQL database that you recorded earlier in this lab.

Note: It's important that you use your updated connection string here. The original connection string copied from the portal won't have the username and password necessary to connect to the SQL database.

5. Save the **appsettings.json** file.

4.3 Task 3: Update the blob base URL

1. In the JSON object on line 8, find the **Settings.BlobContainerUrl** path. Observe that the current value is empty:

```
"Settings": {  
  "BlobContainerUrl": "https://polystorsrini.blob.core.windows.net/images"  
}
```

- 1.
2. Update the value of the **BlobContainerUrl** property by setting its value to the **URL** property of the Azure Storage blob container named **images** that you recorded earlier in this lab.
3. Save the **appsettings.json** file.

4.4 Task 4: Validate the web application

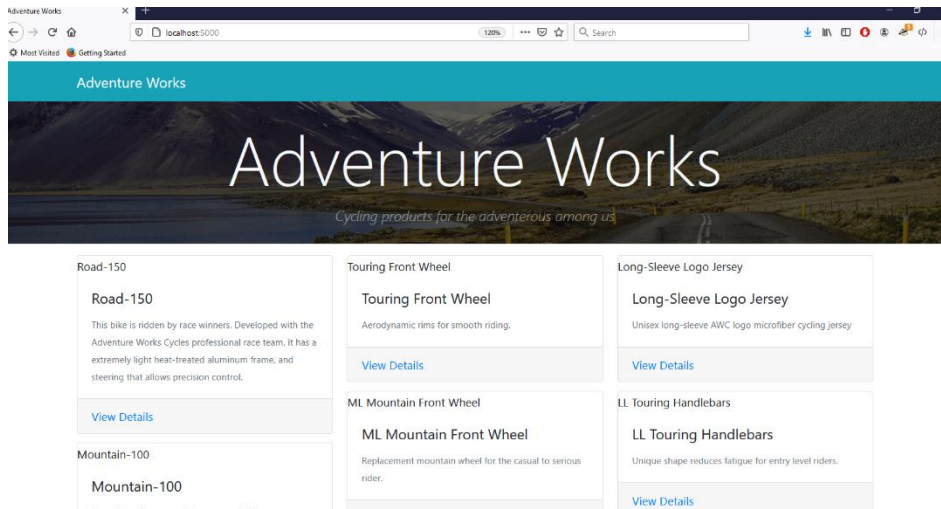
1. In the **Visual Studio Code** window, access the shortcut menu or right-click or activate the shortcut menu for the Explorer pane, and then select **Open in Terminal**.
2. At the open command prompt, enter the following command, and then select Enter to switch your terminal context to the **AdventureWorks.Web** folder:

```
cd .\AdventureWorks.Web\
```

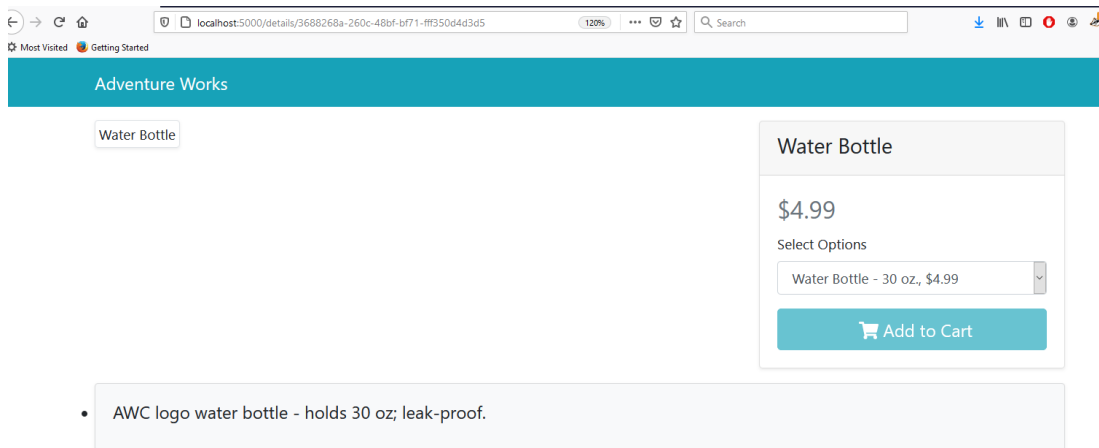
At the command prompt, enter the following command, and then select Enter to run the .NET web application:

```
dotnet run
```

3. **Note:** The **dotnet run** command will automatically build any changes to the project and then start the web application without a debugger attached. The command will output the URL of the running application and any assigned ports.
4. On the taskbar, select the **Microsoft Edge** icon.
5. In the open browser window, browse to the currently running web application (<http://localhost:5000>).



6. In the web application, observe the list of models displayed from the front page.
7. Find the **Water Bottle** model, and then select **View Details**.



8. From the **Water Bottle** product detail page, find **Add to Cart**, and then observe that the checkout functionality is currently disabled.
9. Close the browser window displaying your web application.
10. Return to the **Visual Studio Code** window, and then select **Kill Terminal** or the **Recycle Bin** icon to close the currently open terminal and any associated processes.

4.5 Review

In this exercise, you configured your ASP.NET web application to connect to your resources in Azure.

