😊 Azure Batch

Azure **Batch Service** is a cloud-based **job scheduling** and **compute management** service that enables users to run **large-scale parallel and high-performance computing (HPC) applications** efficiently. It automates resource provisioning, job scheduling, and workload execution on a **pool of virtual machines (VMs)** in Azure.

How Azure Batch Works

1. **Create a Batch Account** – This acts as a container for managing compute resources.
2. **Provision a VM Pool** – A set of VMs that execute tasks in parallel.
3. **Upload Applications & Data** – Store the necessary application binaries and input files in Azure Storage.
4. **Create & Manage Jobs** – A job consists of multiple tasks that execute the application on VMs.
5. **Run & Monitor Jobs** – Azure Batch distributes workloads across VMs, executing tasks efficiently.
6. **Store & Analyze Results** – The processed output is saved in **Azure Storage** or **Azure SQL Database**.

Use Cases of Azure Batch Service

1. Data Processing & ETL Pipelines

- Used for processing massive datasets in parallel.
- Example: Analyzing **log files** or running **big data analytics**.

2. Media Rendering & Transcoding

- Transcode large video files into multiple formats.
- Example: A **video streaming platform** converting videos into different resolutions.

3. Scientific Simulations & Modeling

- Run **complex simulations** in **engineering, physics, or finance**.
- Example: **Weather forecasting models** that process large datasets.

4. AI & Machine Learning Model Training

- Train large-scale ML models across multiple compute nodes.
- Example: Processing millions of **images for deep learning**.

5. Financial Risk Analysis

- Compute-intensive tasks such as **Monte Carlo simulations**.
- Example: **Banking & insurance** companies assessing financial risks.

6. Image Processing & OCR

- Process and analyze large image datasets for **object recognition or text extraction**.
- Example: **Digitizing historical documents** with **OCR (Optical Character Recognition)**.

Benefits of Azure Batch Service

1. Fully Managed Service

- No need to manage infrastructure; **Azure provisions and scales VMs automatically**.

2. Cost-Effective

- Supports **low-priority (spot) VMs**, reducing compute costs significantly.

3. High Scalability

- Automatically scales resources based on workload demands.

4. Parallel Processing

- Distributes tasks across multiple VMs for **faster execution**.

5. Integration with Azure Ecosystem

- Works with **Azure Storage, SQL Database, and AI services** for a complete cloud workflow.

6. Custom VM Images & Software

- Supports **custom VM images** with pre-installed software for specialized tasks.

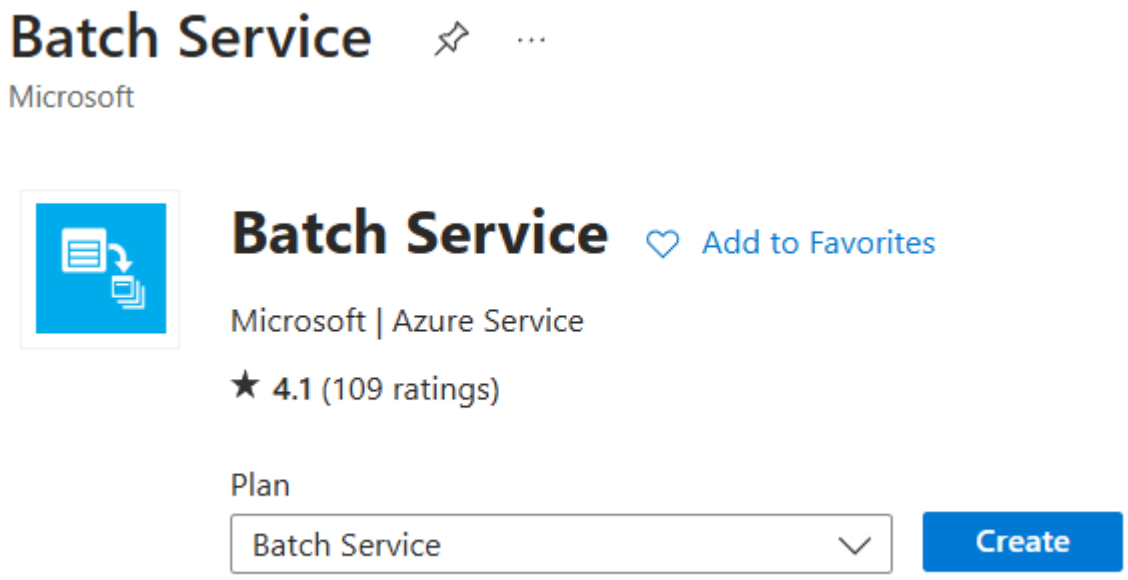
Conclusion

Azure Batch Service is ideal for **compute-intensive workloads** requiring **parallel processing** and **automation**. It simplifies **job scheduling, resource management, and scaling**, making it a powerful solution for industries like **finance, healthcare, media, AI, and scientific research**.

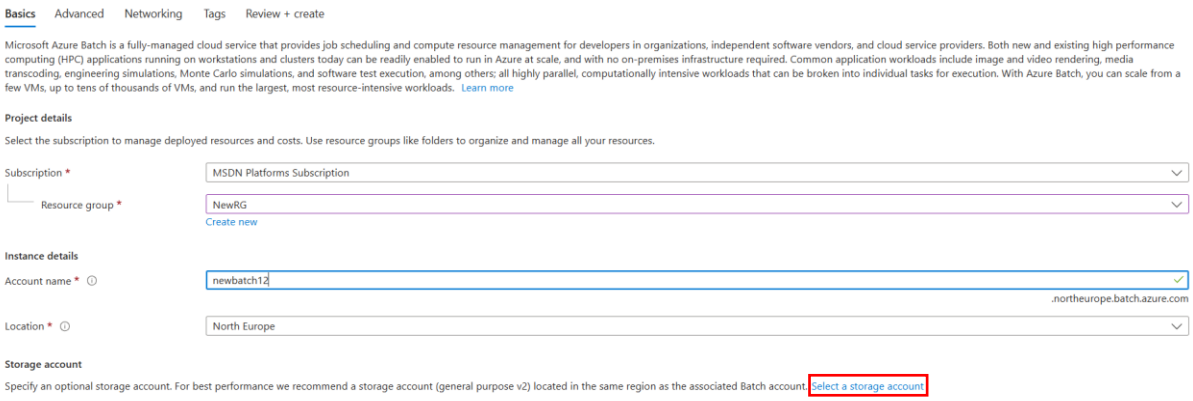
This process sets up an Azure Batch Service to automate data processing. You create a Batch account, SQL database, and storage account. After building a .NET app in Visual Studio to handle data, you upload the app as a zip file to Azure. You create a Batch Pool, add the app, and run a job with a command line to execute the app. The app reads data from Azure Blob Storage and writes it to the SQL database. The end goal is to automate data ingestion into SQL Server using Azure Batch for scalable, serverless processing. Let me know if you'd like me to refine this!

😄 To begin with the Lab

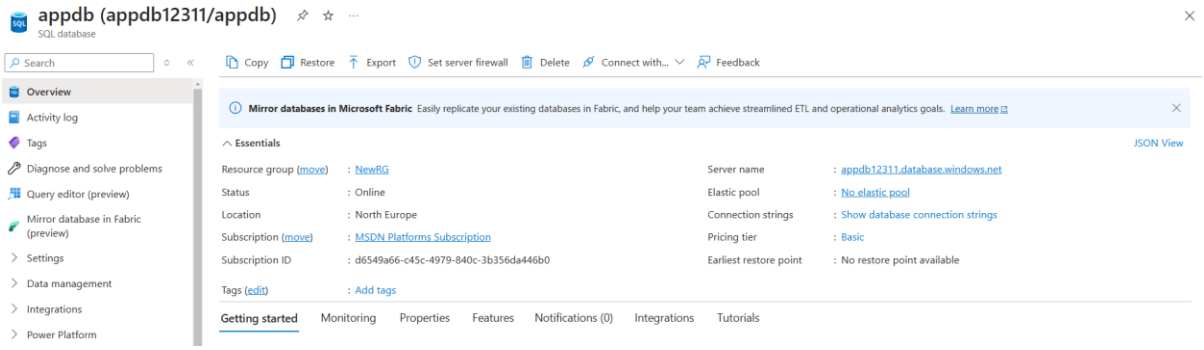
1. Go to the marketplace in the Azure Portal then search for Batch Service. Click on Create.



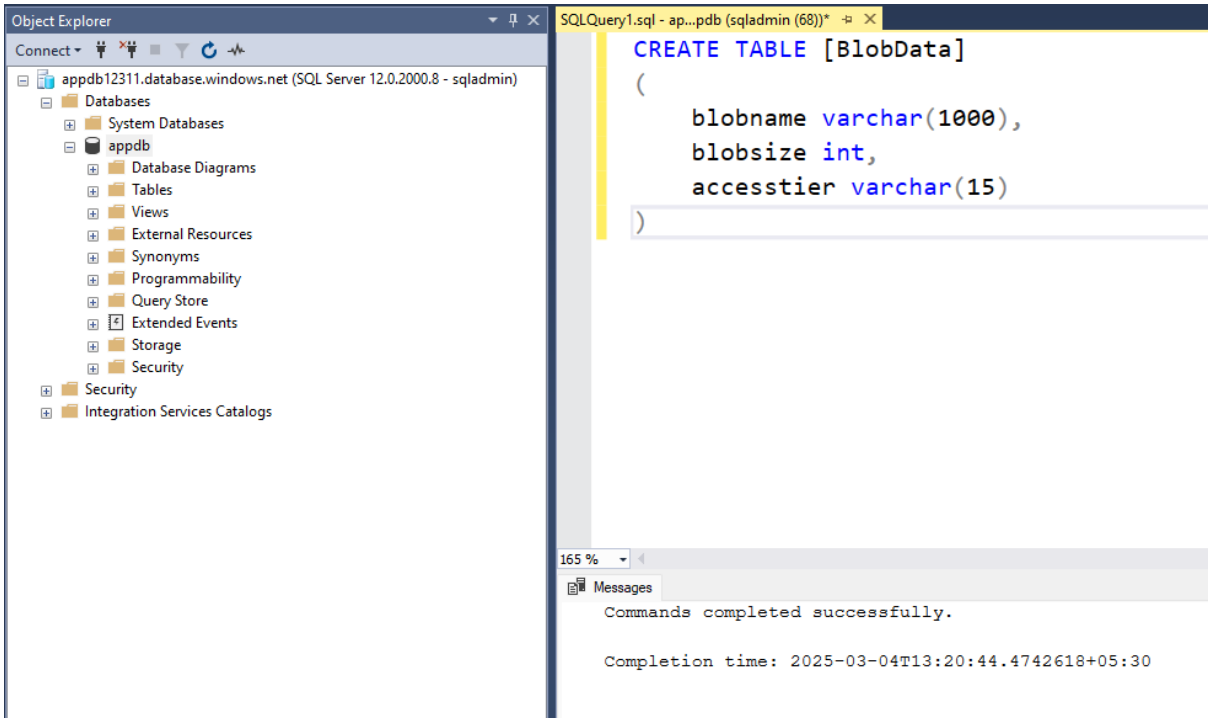
2. Choose your resource group then give an account name, and choose your location. You will also need a storage account for this service. So, if you don't have a storage account, please create a new one.



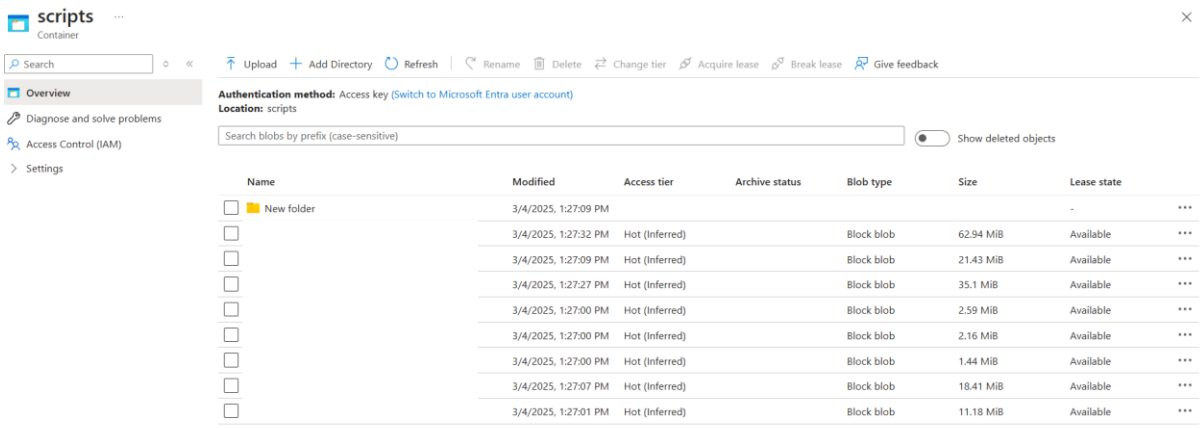
3. After that leave advanced and networking settings as they are. Move to the review page and create your Batch service.
4. Once your batch service is created then you have to create an **SQL database and connect it with the SQL Server Management Studio.**



5. Once you have connected your database in SSMS then create a table.
   CREATE TABLE [BlobData]
   (
   blobname varchar(1000),
   blobsize int,
   accesstier varchar(15)
)



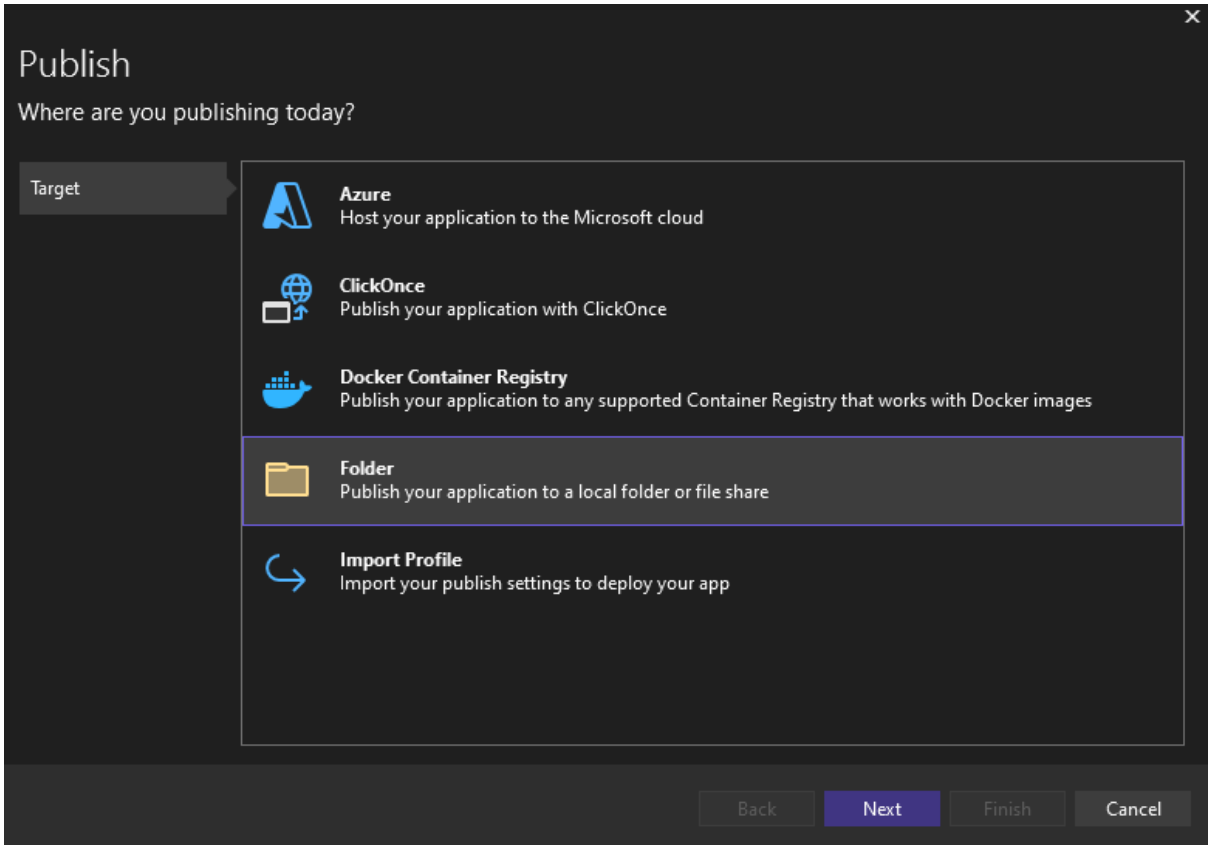
6. After that you have to download a zip file for an application from GitHub and open it in Visual Studio 2022.
7. **Now on the Azure Portal go to your Storage account create a container with the name scripts and upload some data on it.**



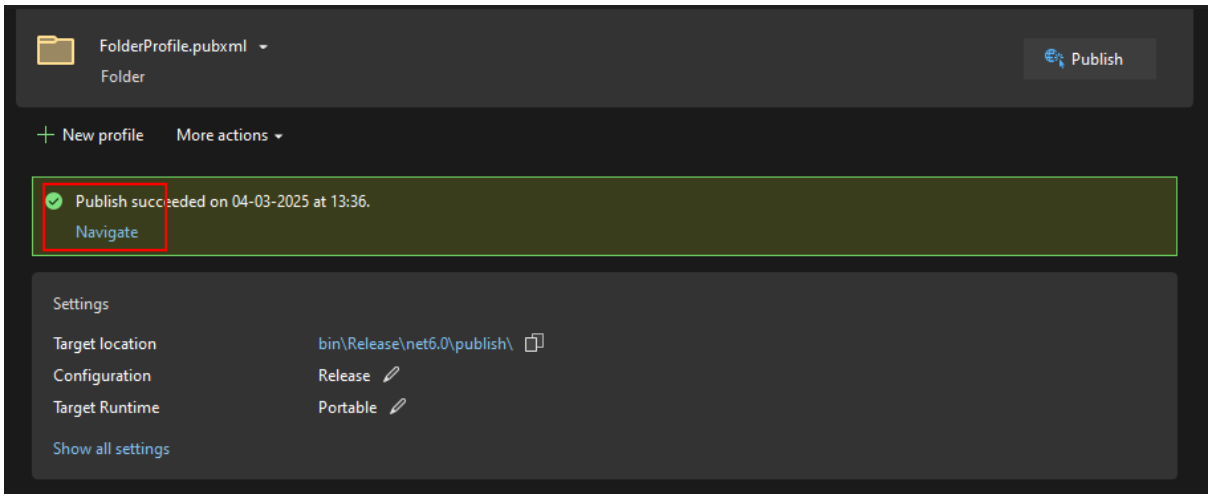
8. Open the Visual Studio and here you can see that on line number 7 we have the container name then on line number 6 you have to give the connection string of your storage account which you can find in the access keys tab.
9. Then you have to change the connection string for SQL Database as well and save the changes.

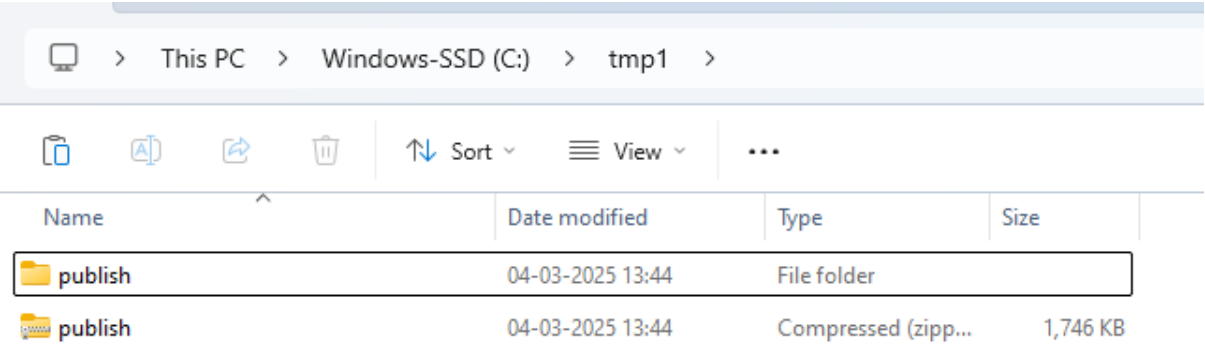
```
1   using Azure.Storage.Blobs;
2   using Azure.Storage.Blobs.Models;
3   using System.Data;
4   using System.Data.SqlClient;
5
6   string conenctionString = "DefaultEndpointsProtocol=https;AccountName=thestorageaccount1201;Accour
7   string containerName = "scripts";
8   int Id = 1;
9
10  BlobContainerClient blobContainerClient = new BlobContainerClient(conenctionString, containerName)
11  SqlConnection _connection = new SqlConnection("Server=tcp:appserver4434334.database.windows.net,14
12
13  SqlParameter paramblobname = new SqlParameter();
14  paramblobname.ParameterName = "@blobname";
15
16  SqlParameter paramblobsize = new SqlParameter();
17  paramblobsize.ParameterName = "@blobsize";
18
```

10. Then we are going to publish our application in the folder.

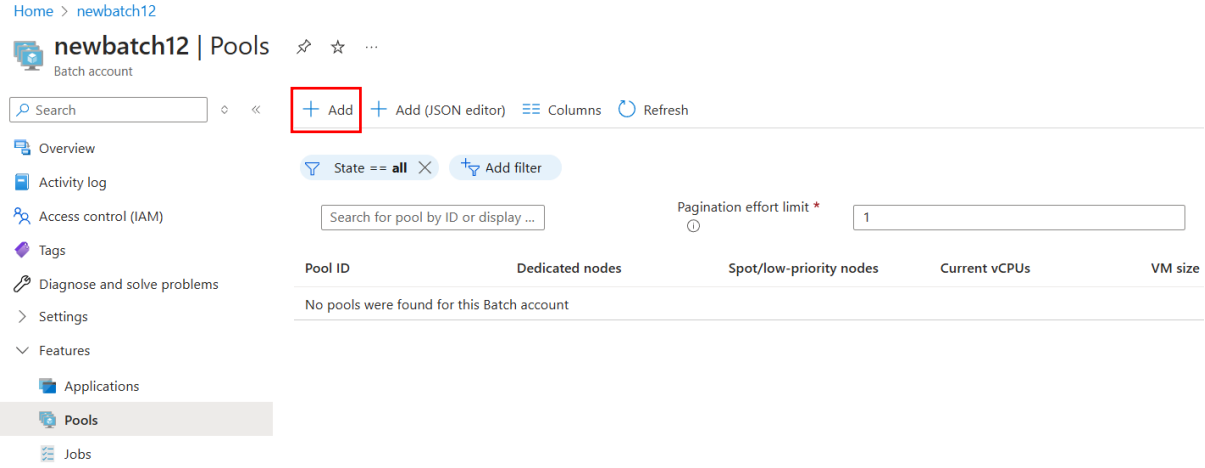


11. Once you have published the application open the folder copy the publish folder and paste it into another folder. Then convert the publish folder into a zip file.

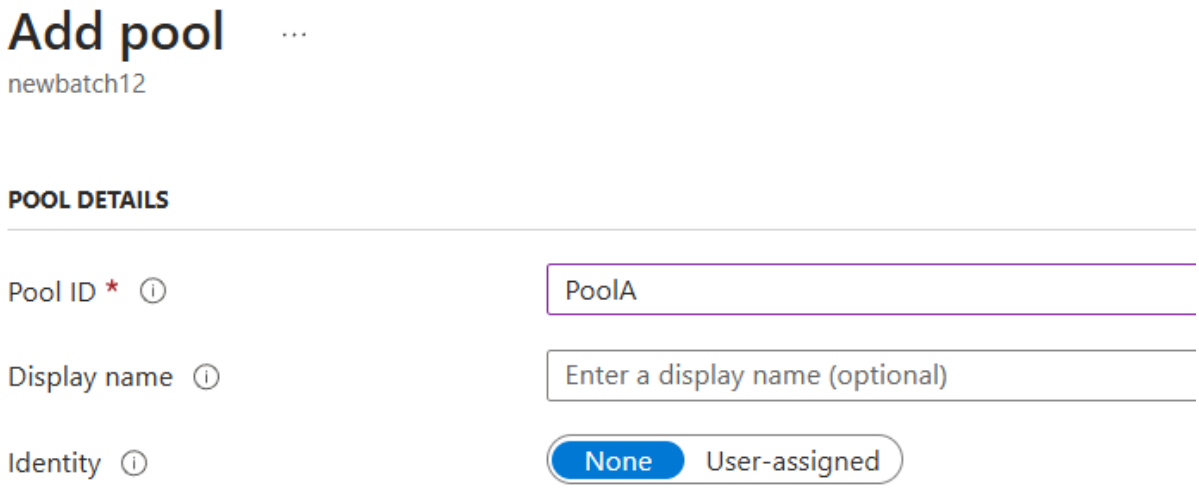




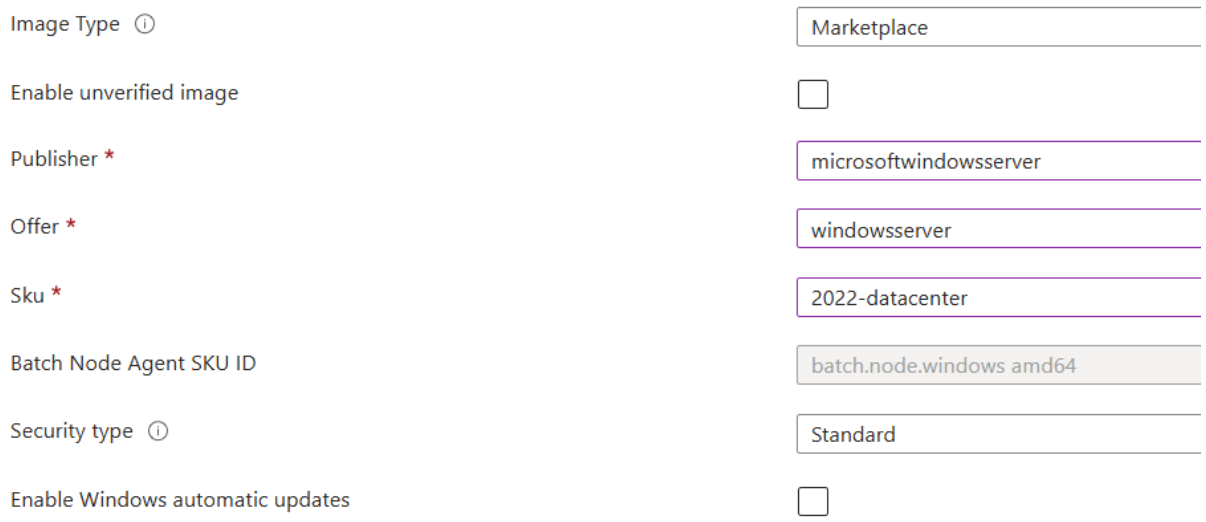
12. Now open your Portal and navigate to the Batch service. Here you have to click on Pools from the left pane and then click on Add.



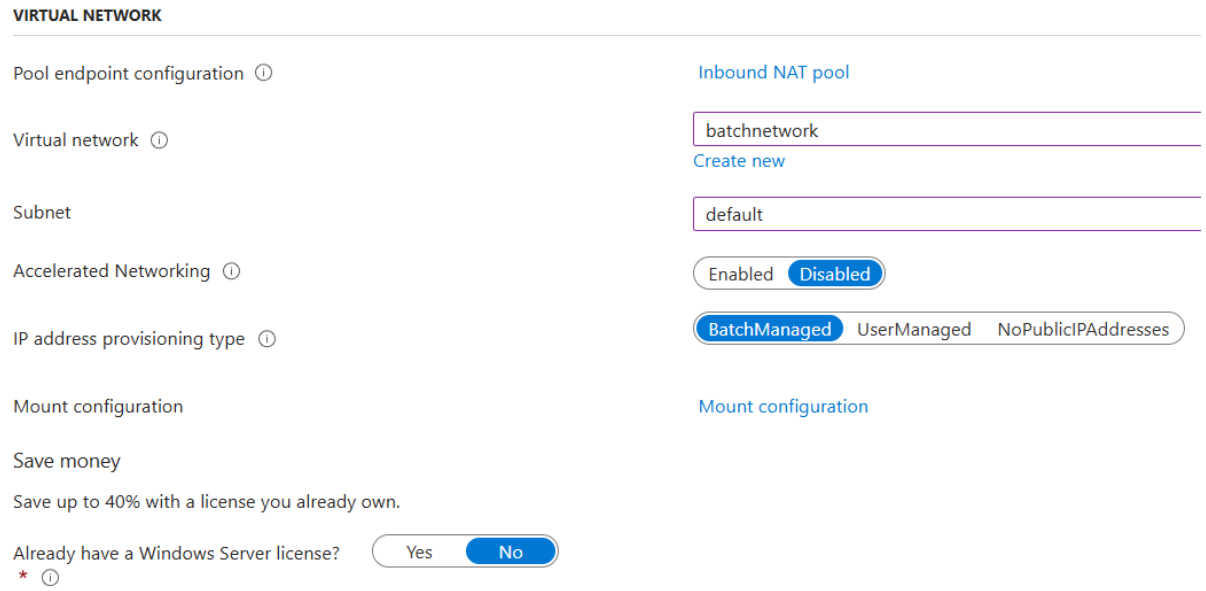
13. First, we have to give a name to our Pool.



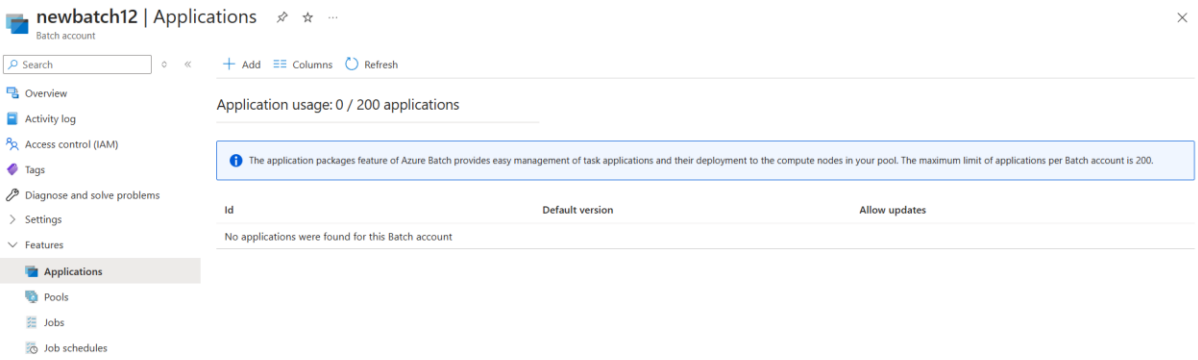
14. Then we have to choose an image from the market place and we have to choose Microsoft Windows Server 2022 data centre.



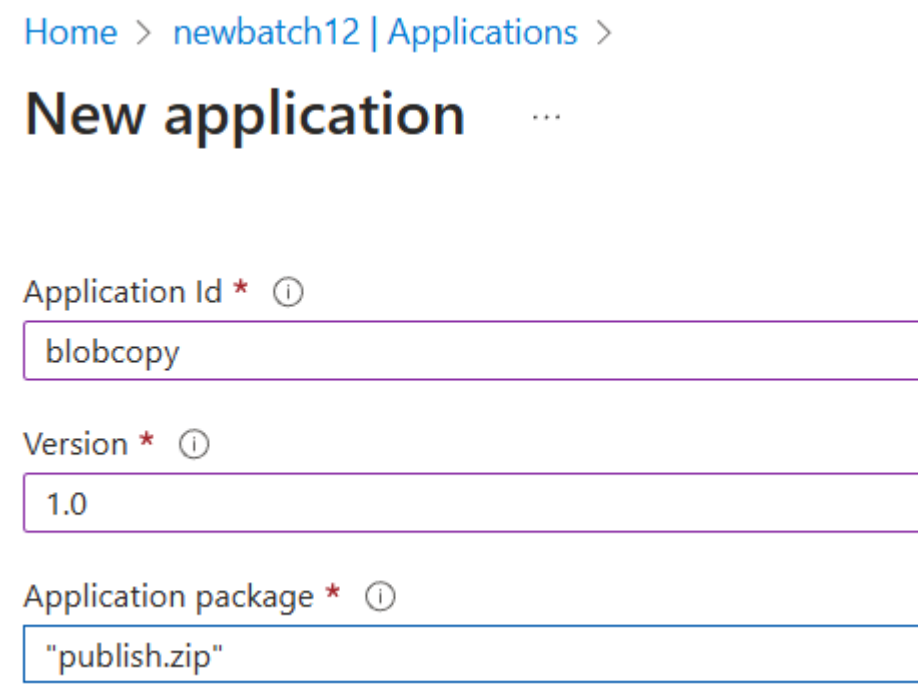
15. After that in the network you have to create a new network and then click on ok to add your Pool.



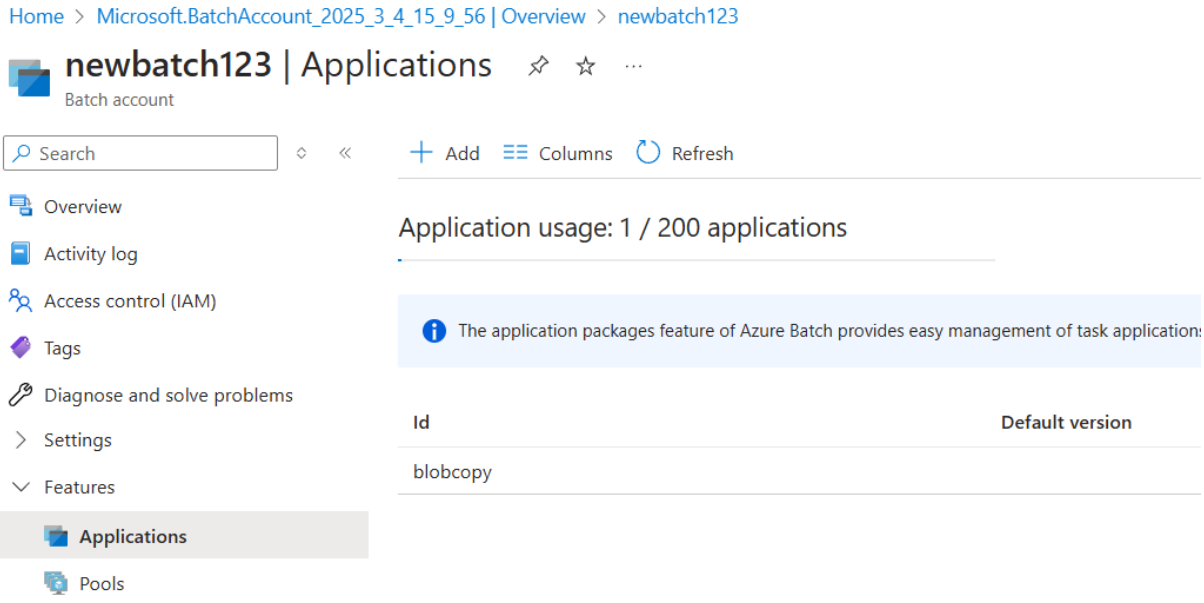
16. Now we need to open the Application tab to add the application which we published from the visual studio. Click on Add.



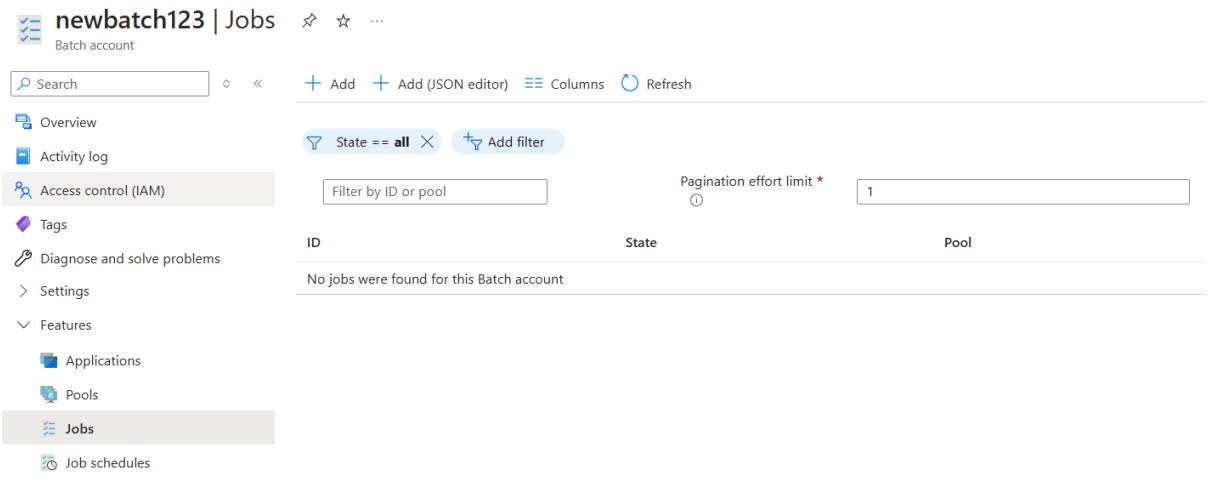
17. Give an application ID and Version number then the browser for the zip file.



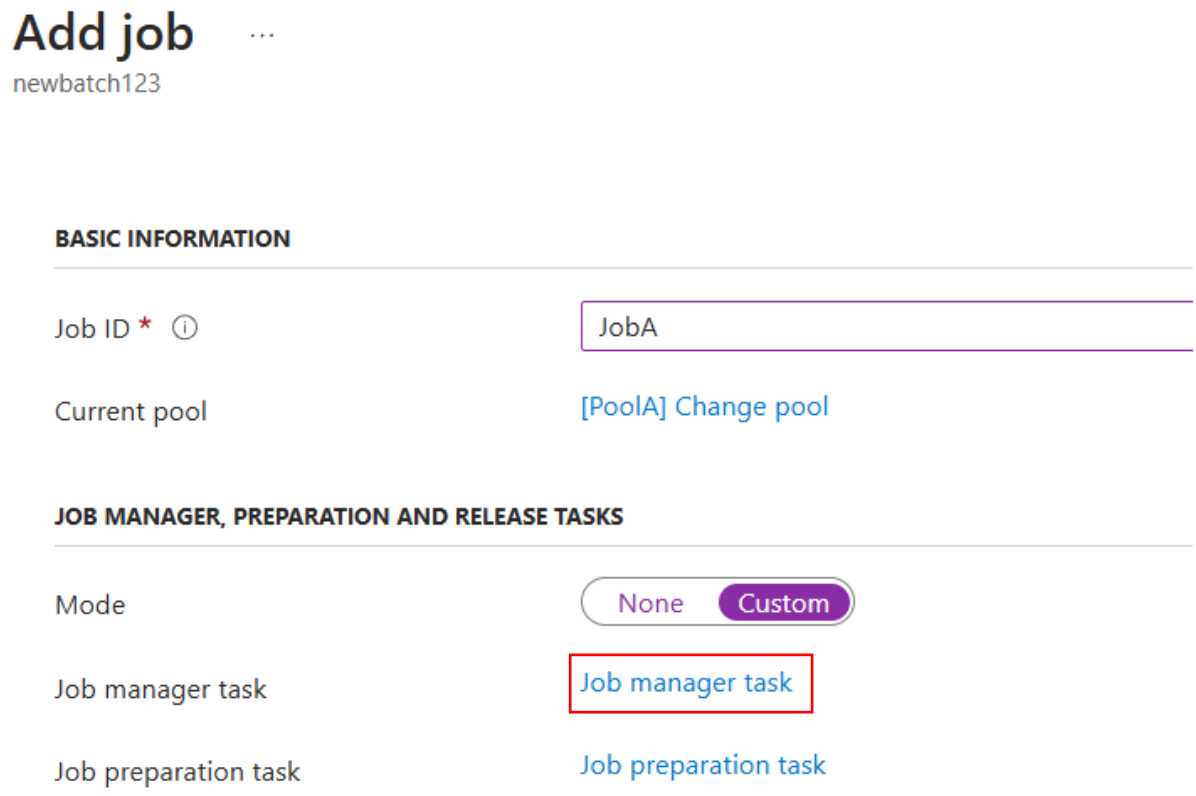
18. Now you can see that our application has been added successfully. Now we need to create a job.



19. Click on add to add a job.

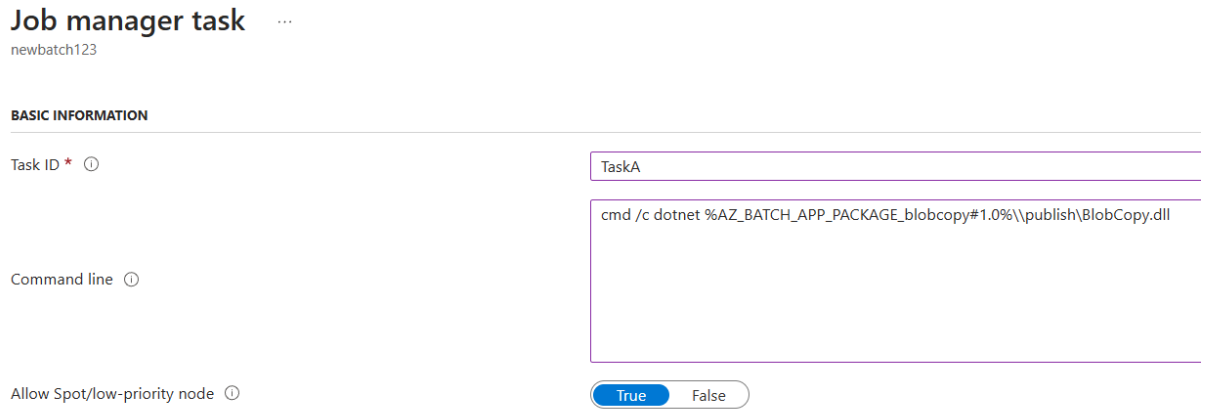


20. Here you need to give a job name then choose your Pool, after that in the Mode choose Custom and click on Job Manager task.

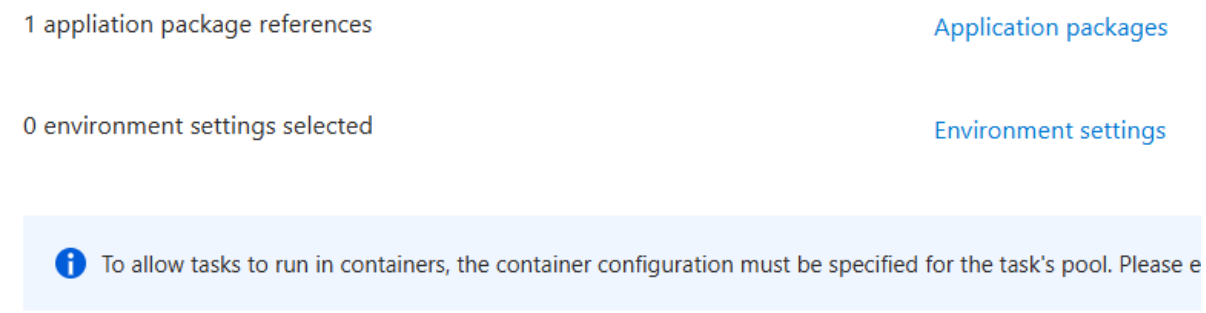


21. In the job manager task give a task ID then put a command line and scroll down.

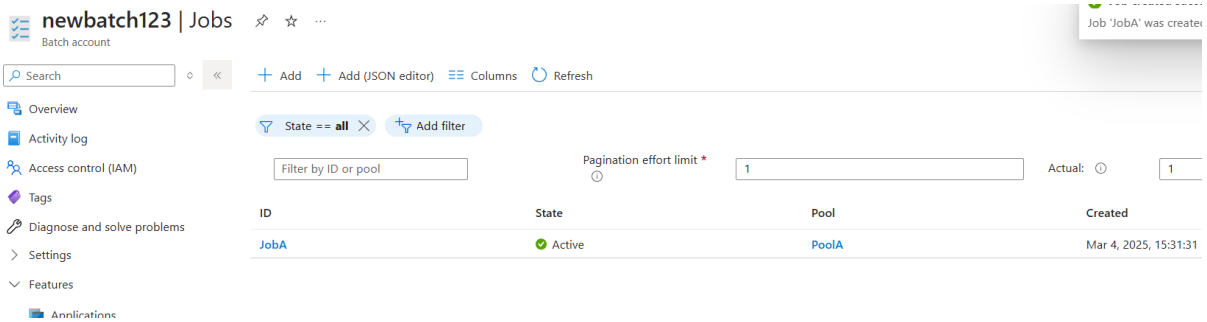
cmd /c dotnet %AZ_BATCH_APP_PACKAGE_blobcopy#1.0%\\publish\BlobCopy.dll



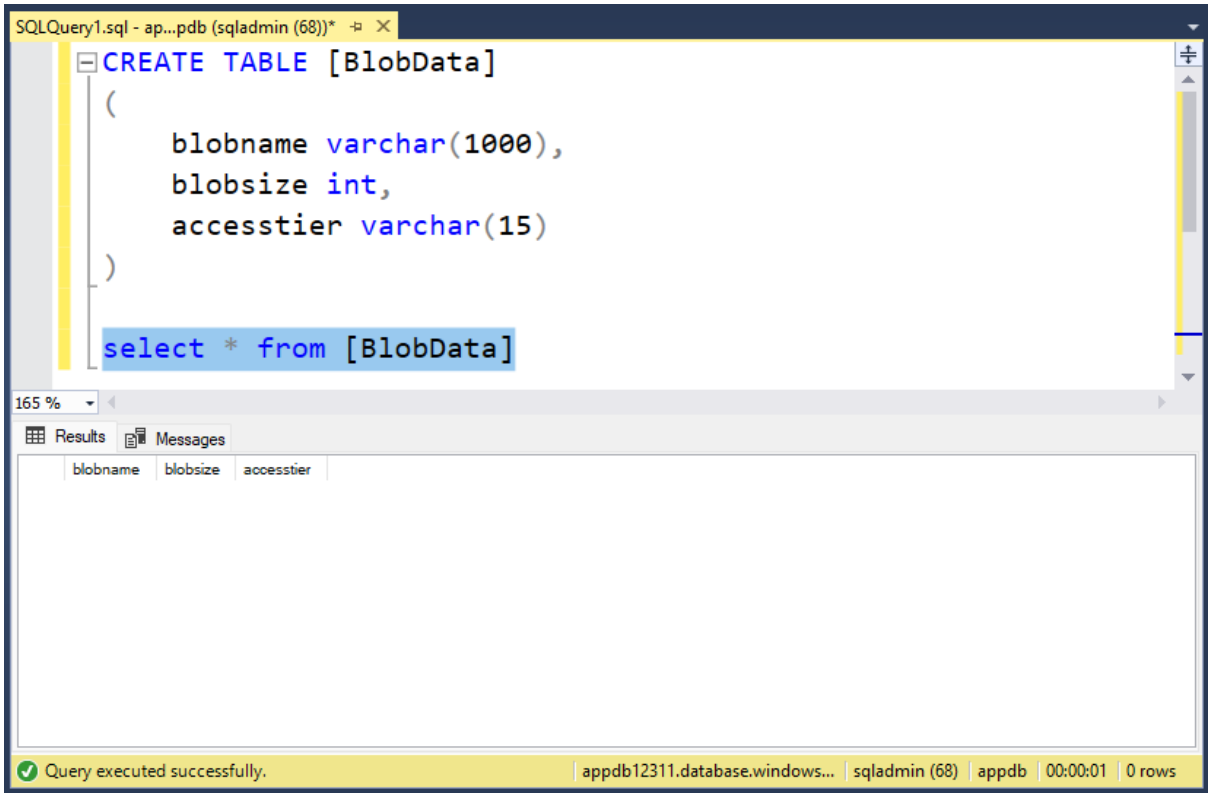
22. Then in the application packages choose your application name and version number. Just create your Job.



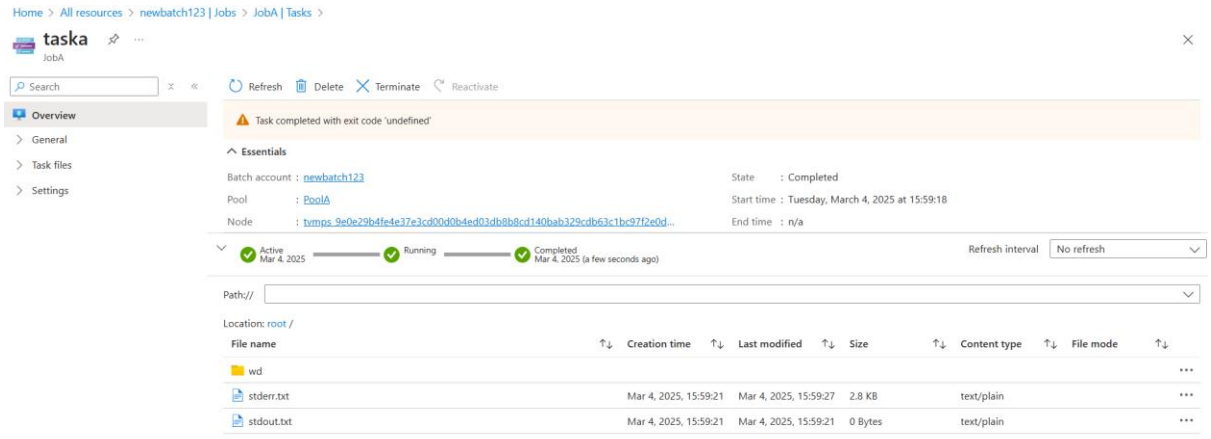
23. Now you can see that our job has been created.



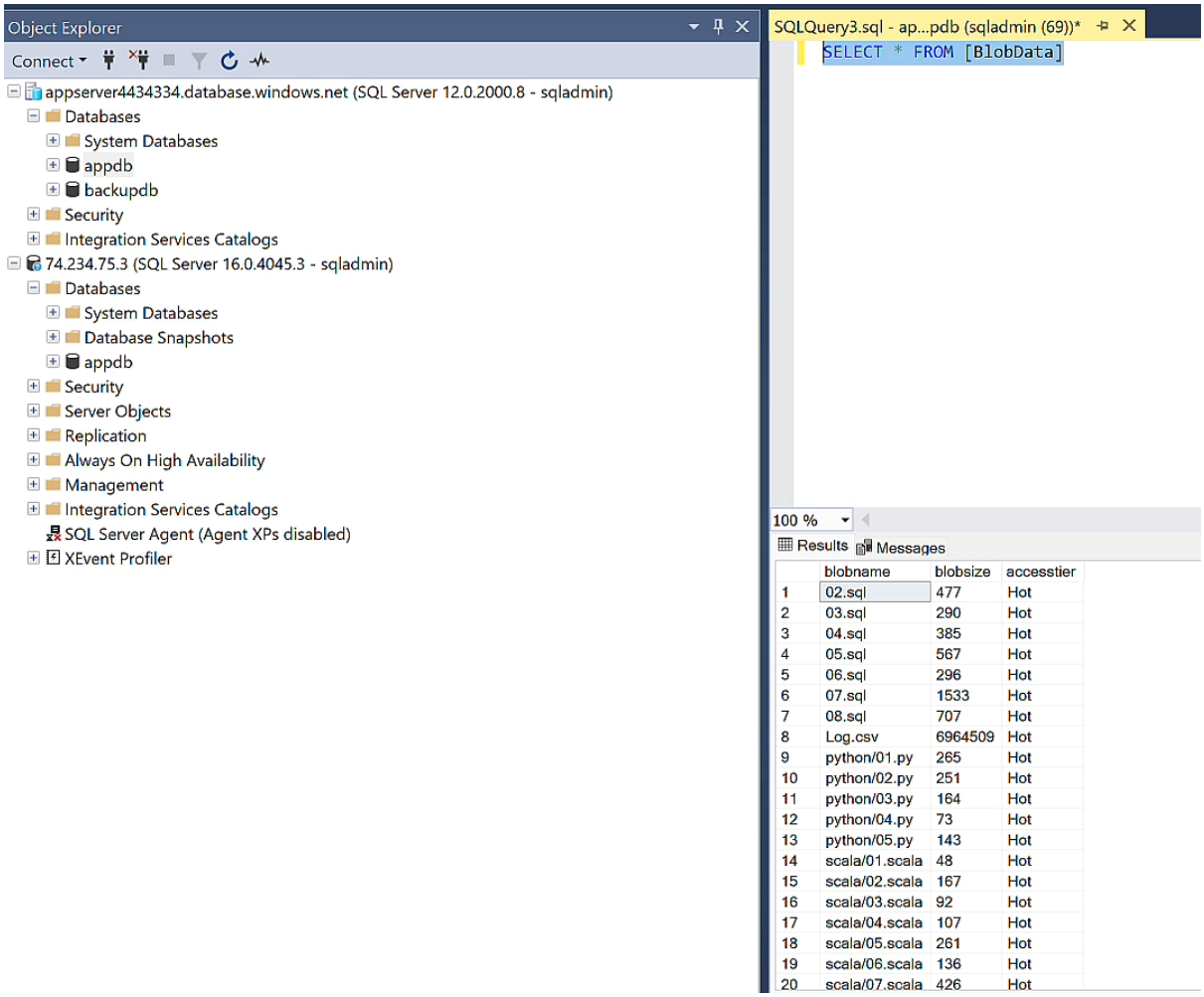
24. Also, you can see that our table is currently empty.



25. You can see that our task has been completed. Go to SQL Server Management Studio and execute the select query again.



26. You can see that the data has been loaded to the our SQL Table.



	blobname	blobsize	accesstier
1	02.sql	477	Hot
2	03.sql	290	Hot
3	04.sql	385	Hot
4	05.sql	567	Hot
5	06.sql	296	Hot
6	07.sql	1533	Hot
7	08.sql	707	Hot
8	Log.csv	6964509	Hot
9	python/01.py	265	Hot
10	python/02.py	251	Hot
11	python/03.py	164	Hot
12	python/04.py	73	Hot
13	python/05.py	143	Hot
14	scala/01.scala	48	Hot
15	scala/02.scala	167	Hot
16	scala/03.scala	92	Hot
17	scala/04.scala	107	Hot
18	scala/05.scala	261	Hot
19	scala/06.scala	136	Hot
20	scala/07.scala	426	Hot