## **Loading data into Azure SQL Data Warehouse by using Azure Data Factory**

**Introduction:**

Fabrikam has got impresses by the performance and the services that they got while using the Azure SQL Database Service. This convinced the Engineering team of Fabrikam to deploy their big data workloads to Azure. Fabrikam is now running a trail run in Azure where they are deploying an Azure SQL Data Warehouse in which they can perform the analytics over their relational data sets that are large in number by making use of the Massively Parallel Processing concept of Azure SQL Data Warehouse. To do so, they must copy their relational tables available in the Azure SQL Database using Azure Data Factory. In the following demo, we will be deploying Azure SQL Datawarehouse and Azure Data Factory. Followed by that we will deploy a SQL Server database with sample data in it and perform Azure Data copy pipelines from SQL Server Database to Azure SQL Datawarehouse. By this, the Fabrikam team will have an idea about using Azure Data Factory with SQL Data Warehouse and SQL Database.

## **Prerequisites**

Before beginning the demo, we need to create the following services.

* **Azure SQL Data Warehouse**: The data warehouse holds the data that's copied over from the SQL database.
* **Azure SQL Database:** This tutorial copies data from an Azure SQL database with Adventure Works LT sample data.
* **Azure storage account:** Azure Storage is used as the *staging* blob in the bulk copy operation.

**Creating Azure SQL Data Warehouse:**

**Step 1:** Sign into your Azure Portal using your credentials and click on **Create a resource** in the upper left-hand corner of the Azure portal and select **SQL Data Warehouse** from the **Databases** category. Fill the necessary details as below to create your SQL Data Warehouse

**Database name**: Any name as an identifier

**Subscription:** Select your subscription

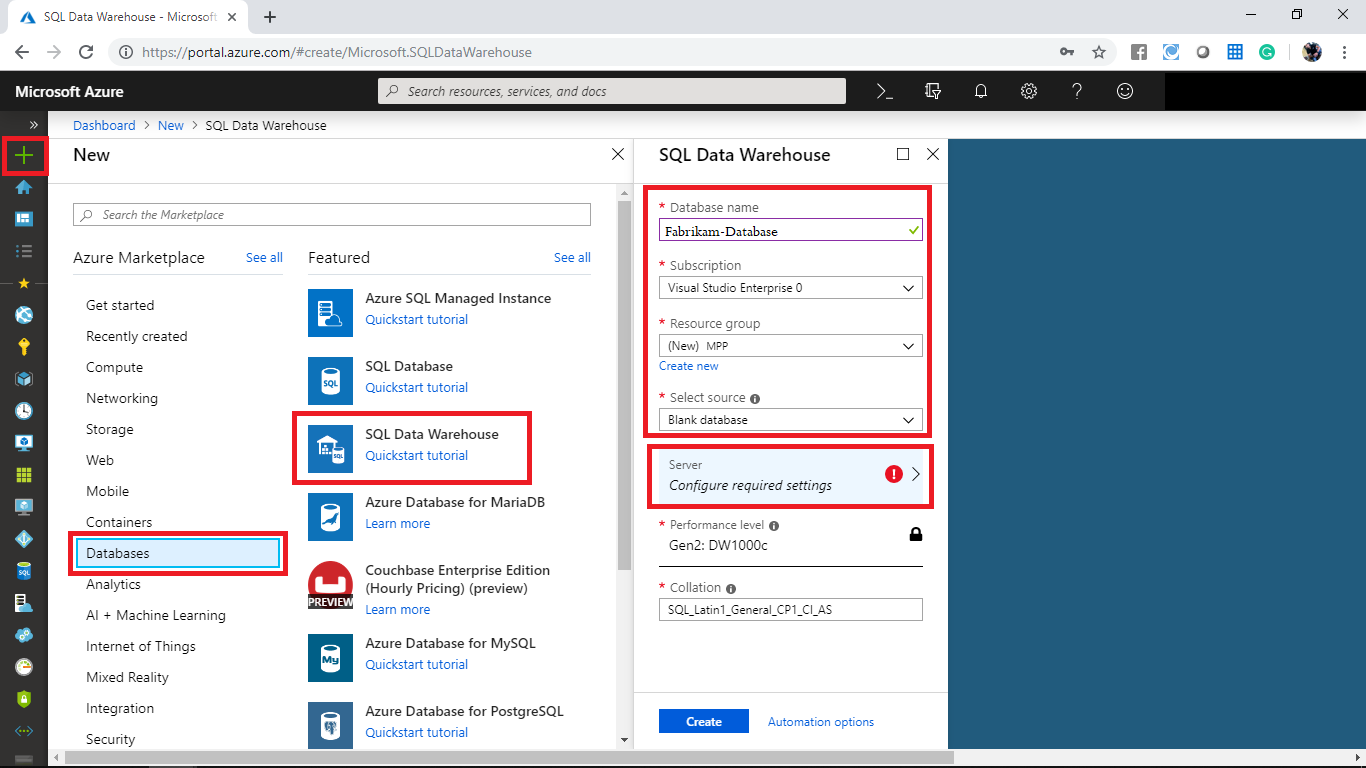
**Resource group**: Create a new resource group with a name

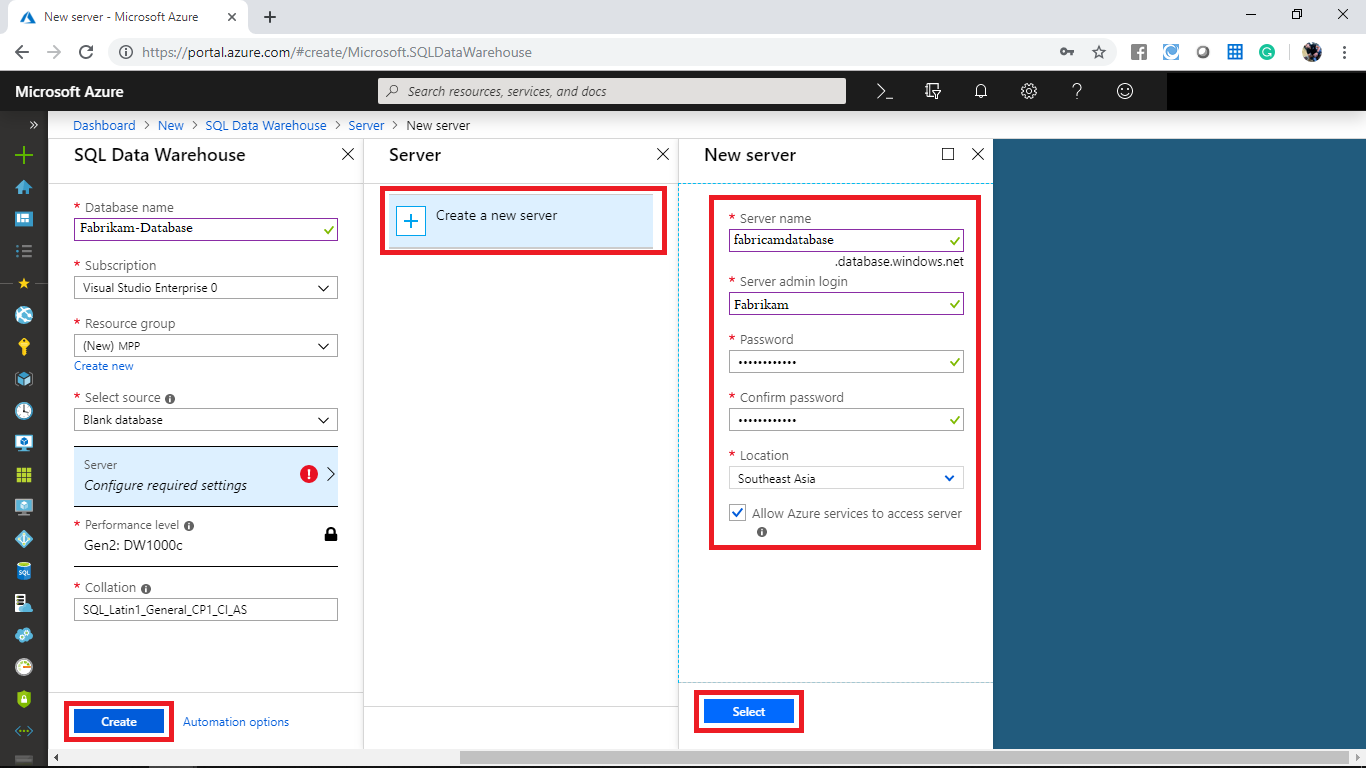
**Select source:** Blank database

**Server:** Create a new server with unique name and login creadentails. Select a location nearest to you.

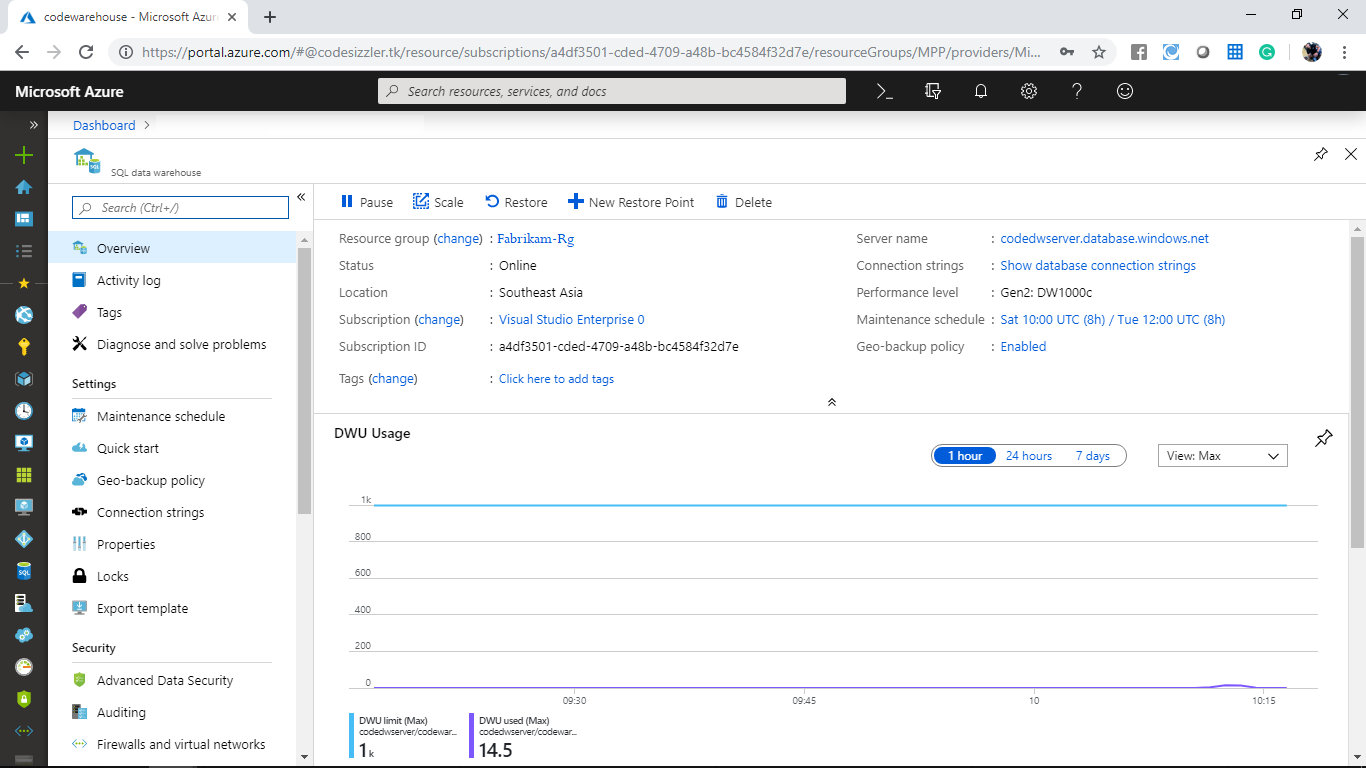
**Perfomance level:** Select a suitable tier

Click on **Create** to provision the database and wait for few mintues for the provision to be succeded.

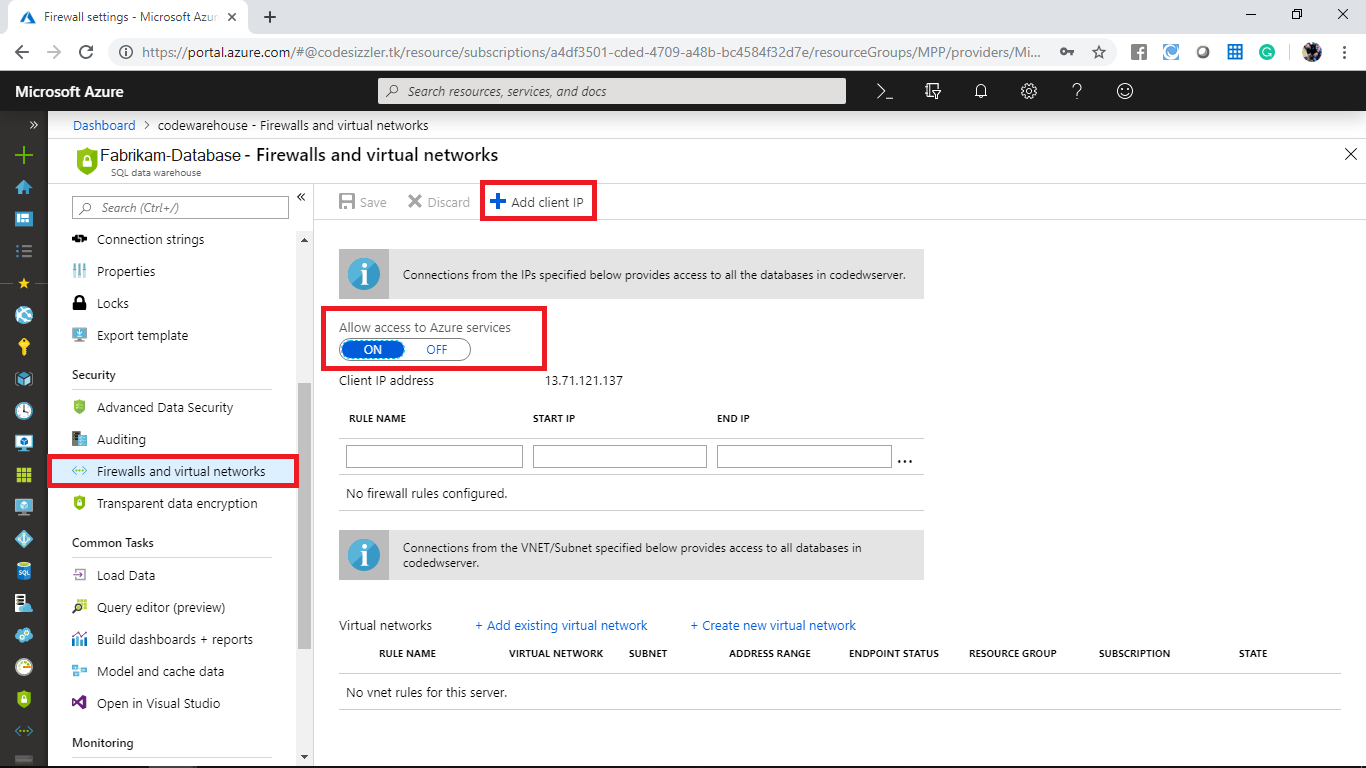




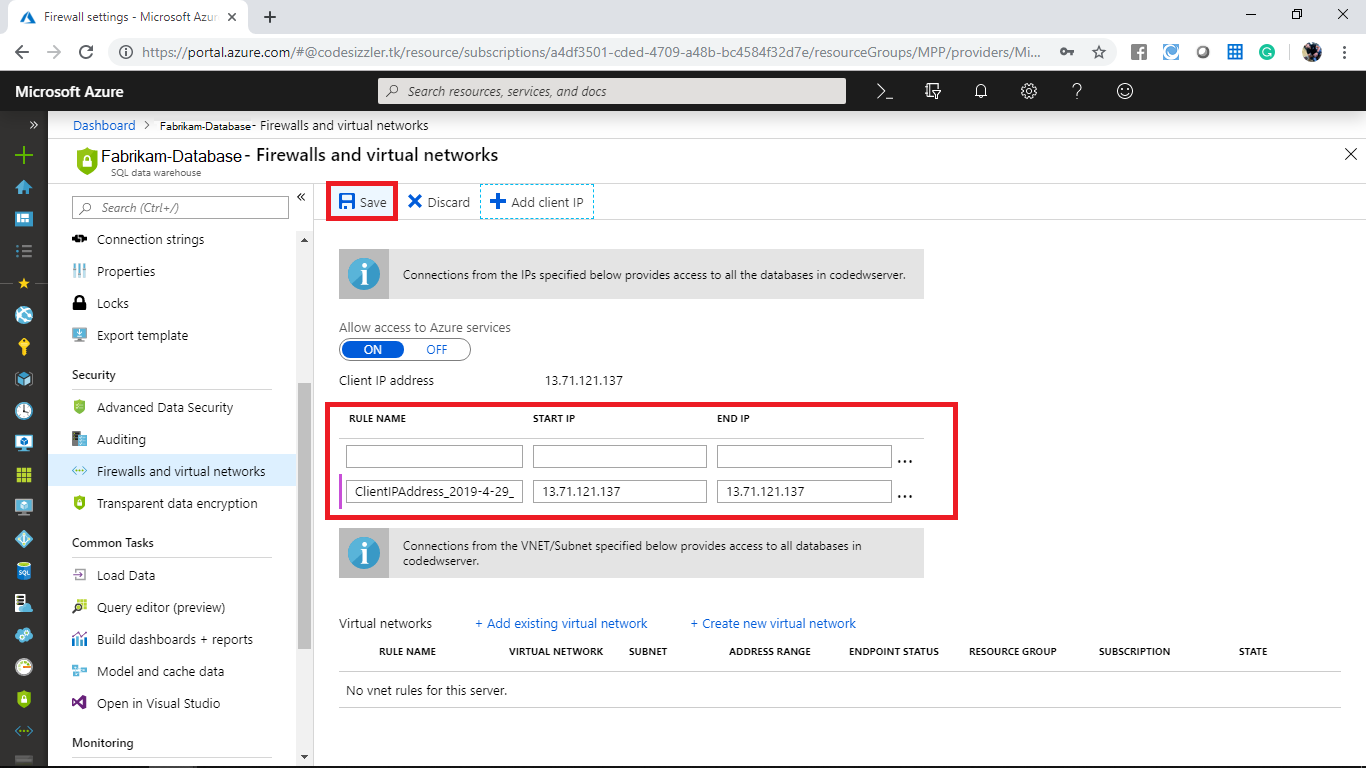
**Step 2:** Once the provision is completed open the resource and you can see the settings blade for the newly created Azure Data Warehouse. Click on the **Server name** to open the settings for the server.

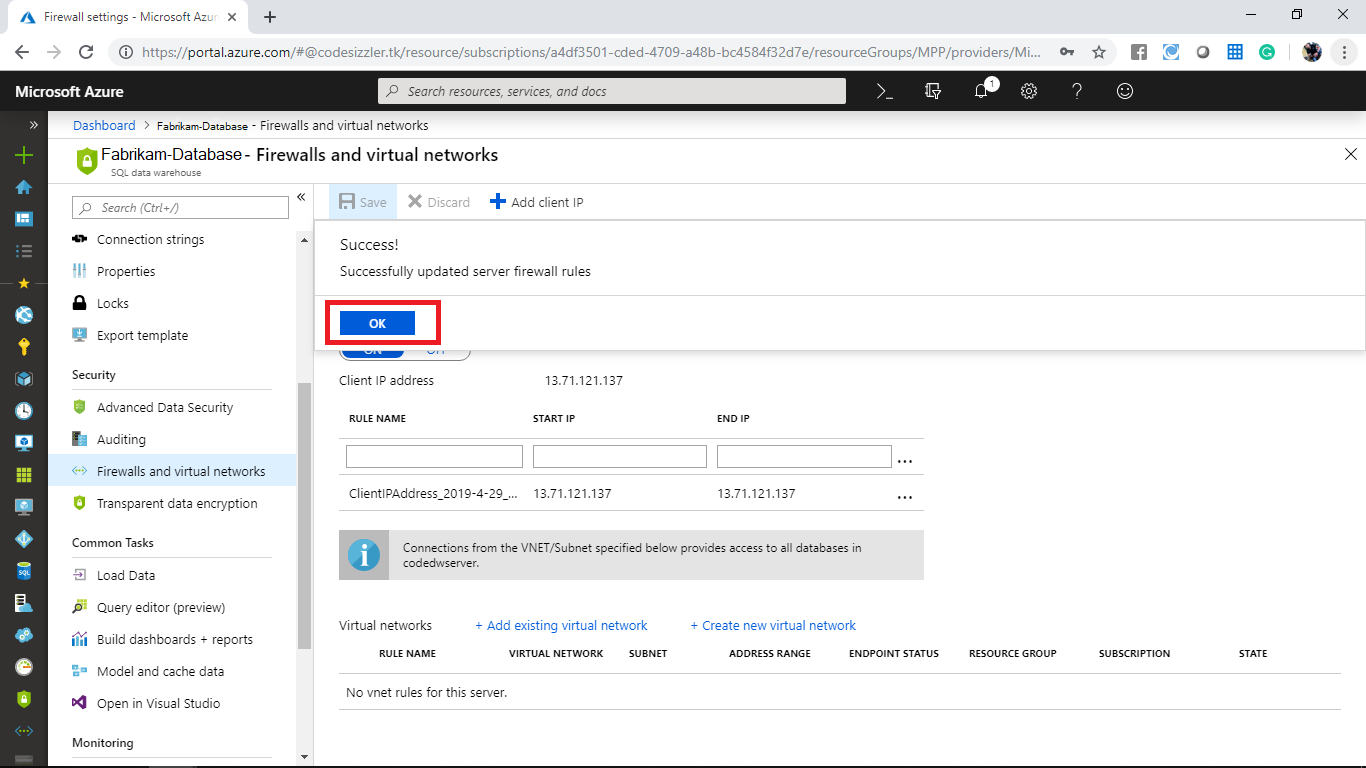


**Step 3:** Now you have got the settings for the server. The SQL Data Warehouse allows you to secure it with Firewalls. Navigate to **Firewalls and Virtual networks** to view the firewall options. By default, all the Azure services are allowed to access Azure SQL Datawarehouse. In the same blade, to add client machines IP, click on **+ Add client IP**.



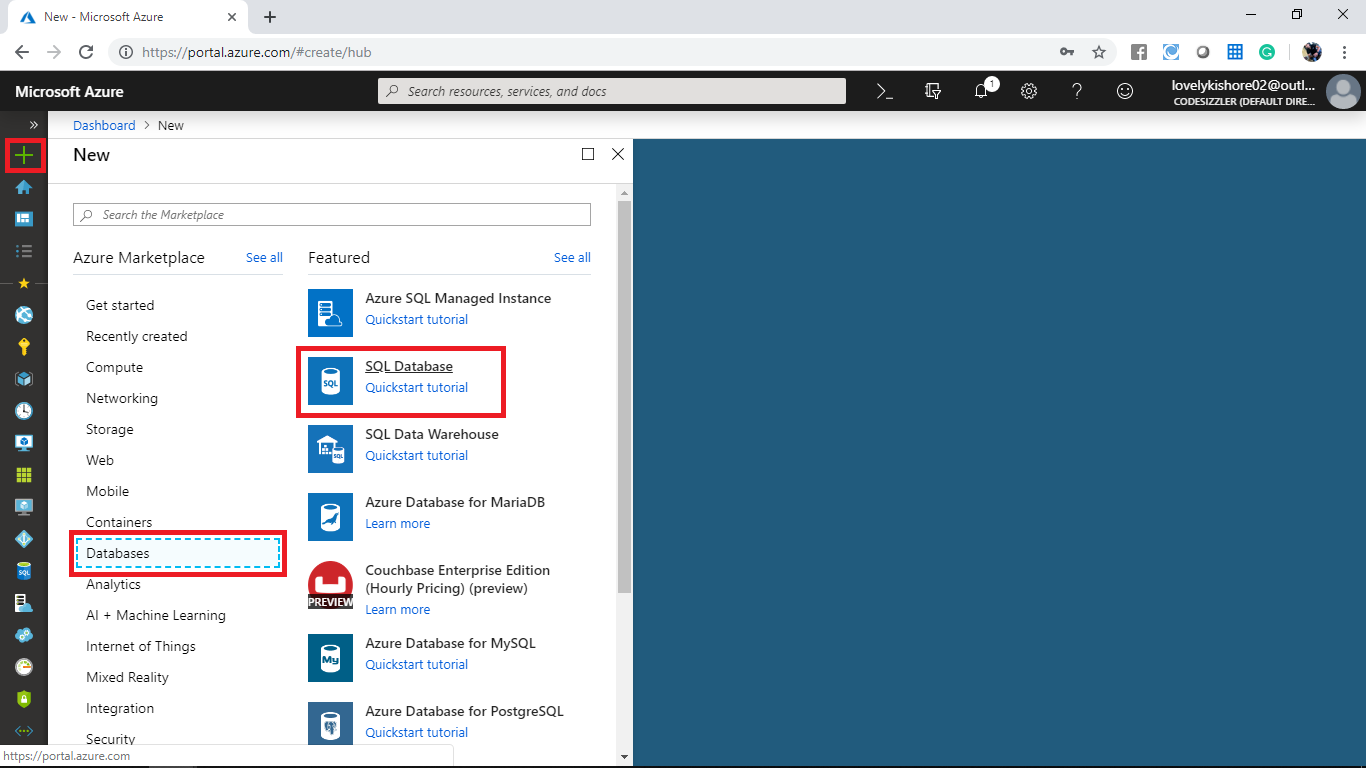
**Step 4:** It will open the List of Client IPs allowed to access the server. Click on **Add client IP** to add your current IP address, and click on **Save** to save the changes. A firewall rule can open port 1433 for a single IP address or a range of IP addresses.



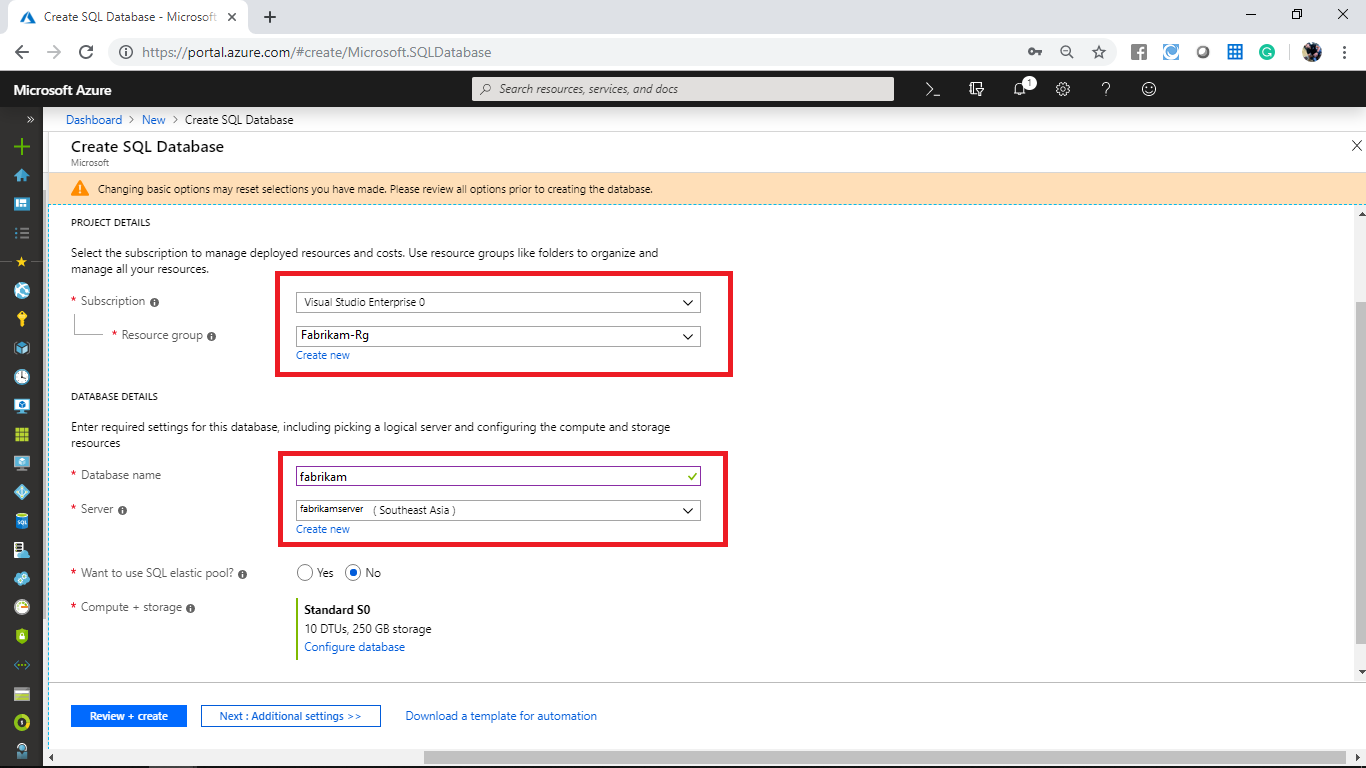


**Creating SQL DB:**

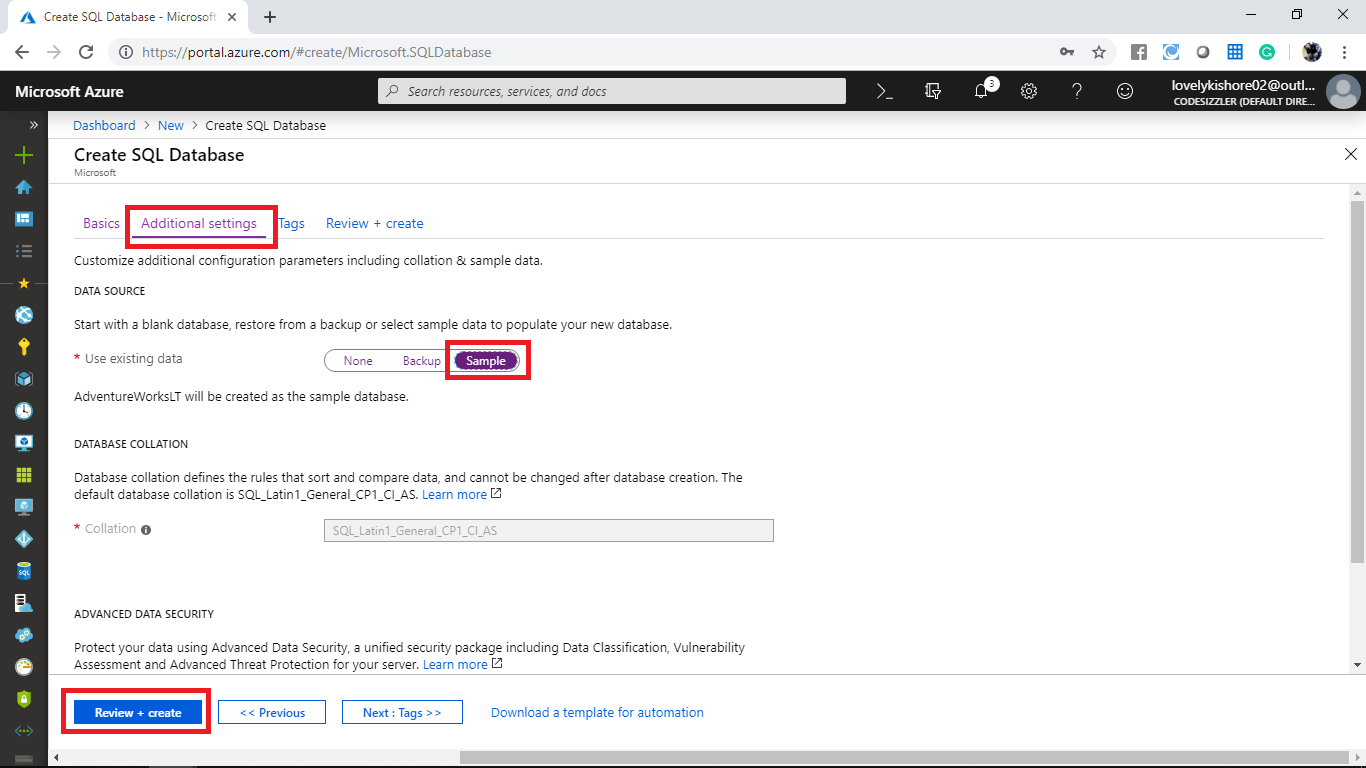
In the azure portal, go to **+ Create a resource -> Database -> SQL Database.**



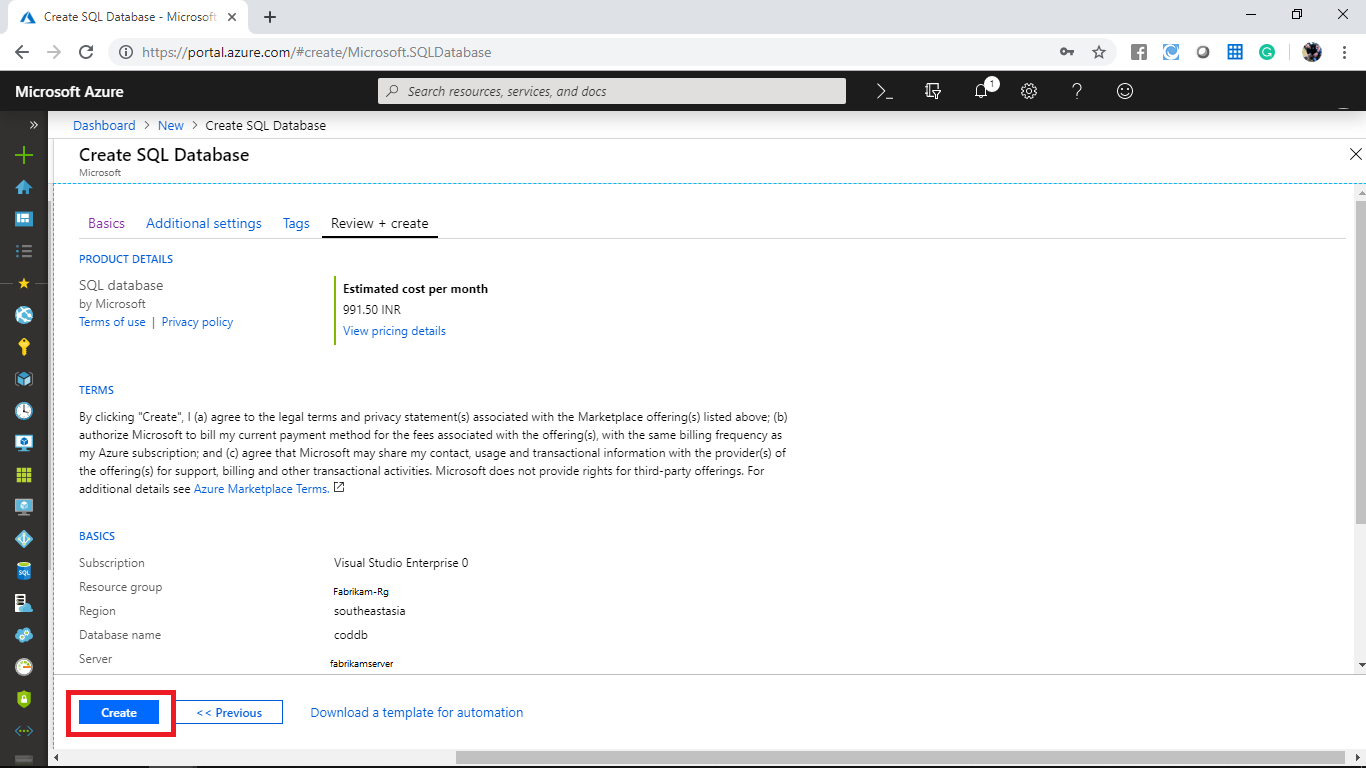
Give an unique name and choose a subscription and a resource group. Create a server with a name and username and password or choose an existing one if you already have any as denoted in the below given image and deploy the database.



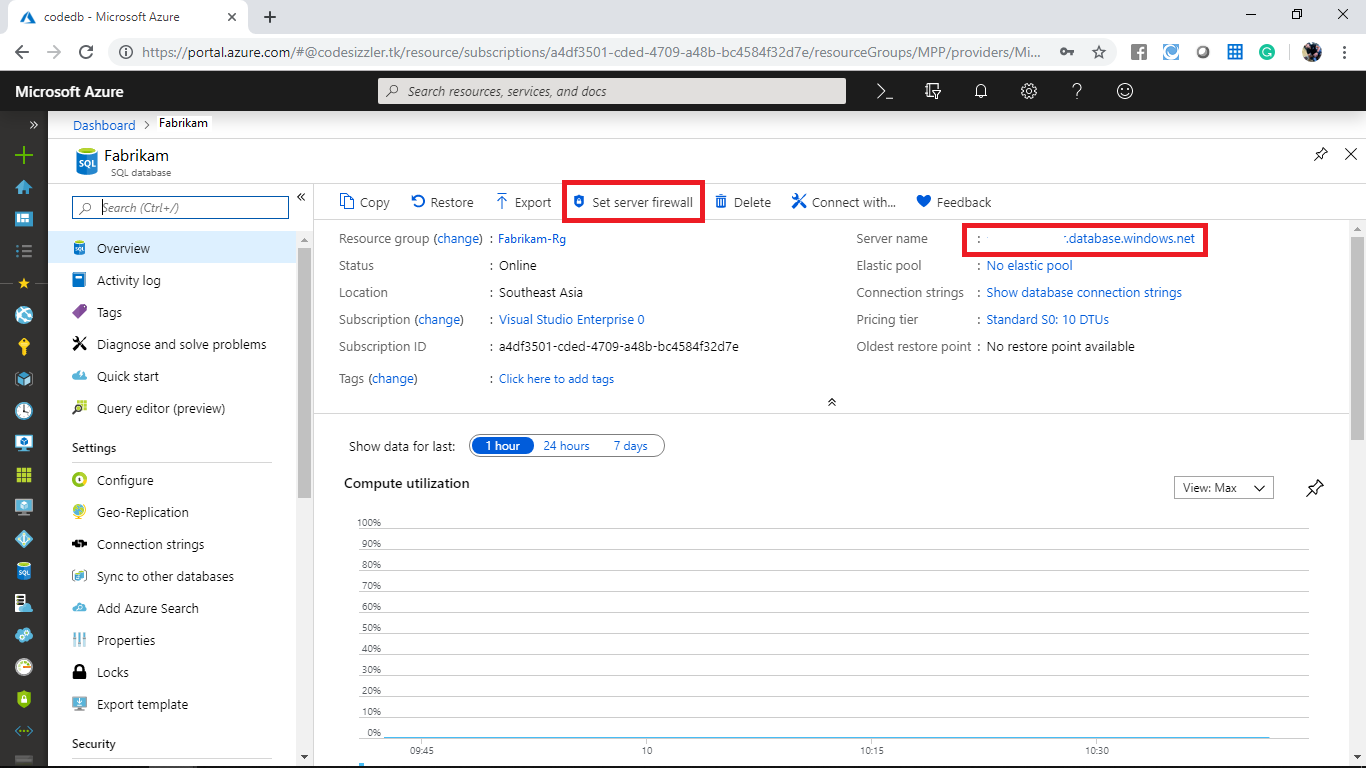
In the **Additional settings** choose **Sample** for **Use existing data** and click on **Review + create**.



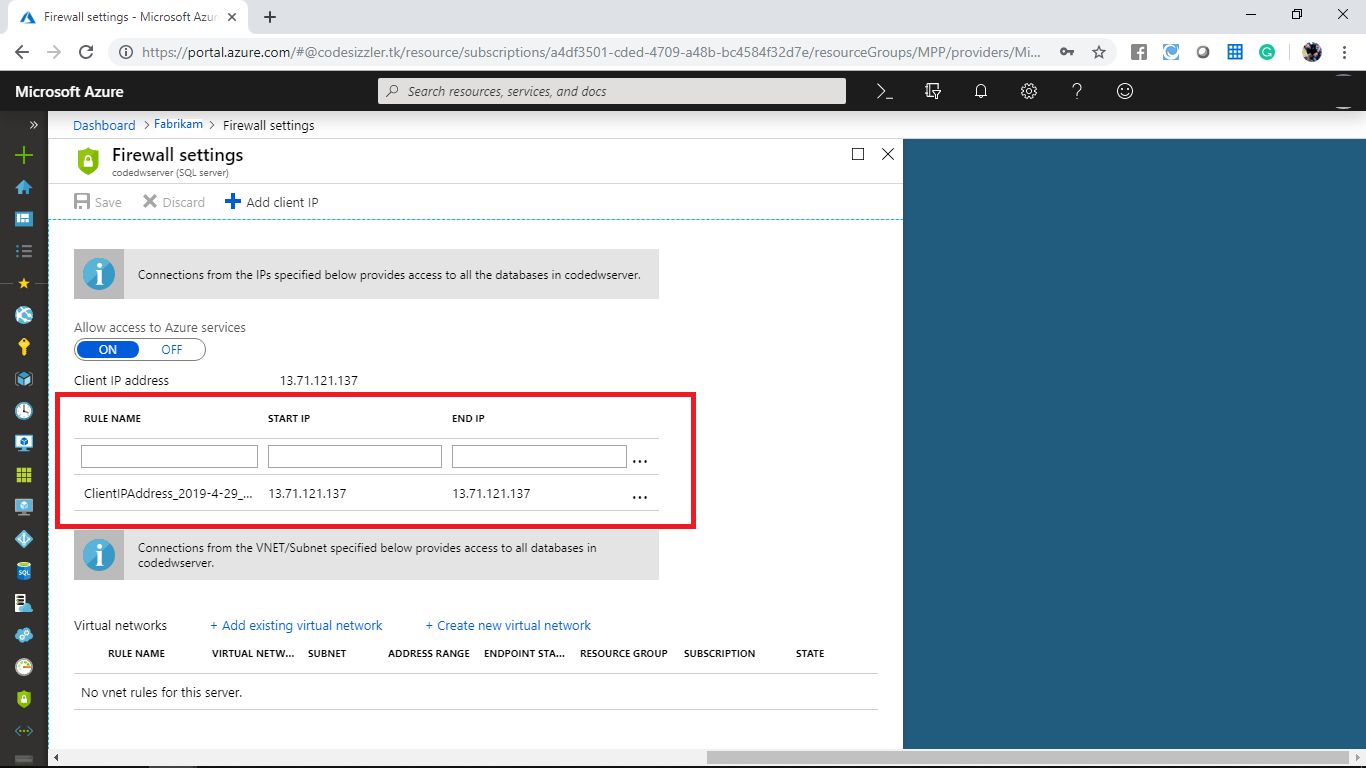
In the **Review + create** page click on **Create** button to deploy the SQL Database.



After the deployment gets over, you can view the DNS of your SQL Database in its overview page. Click on the **Set server firewall** option to configure access to the server.

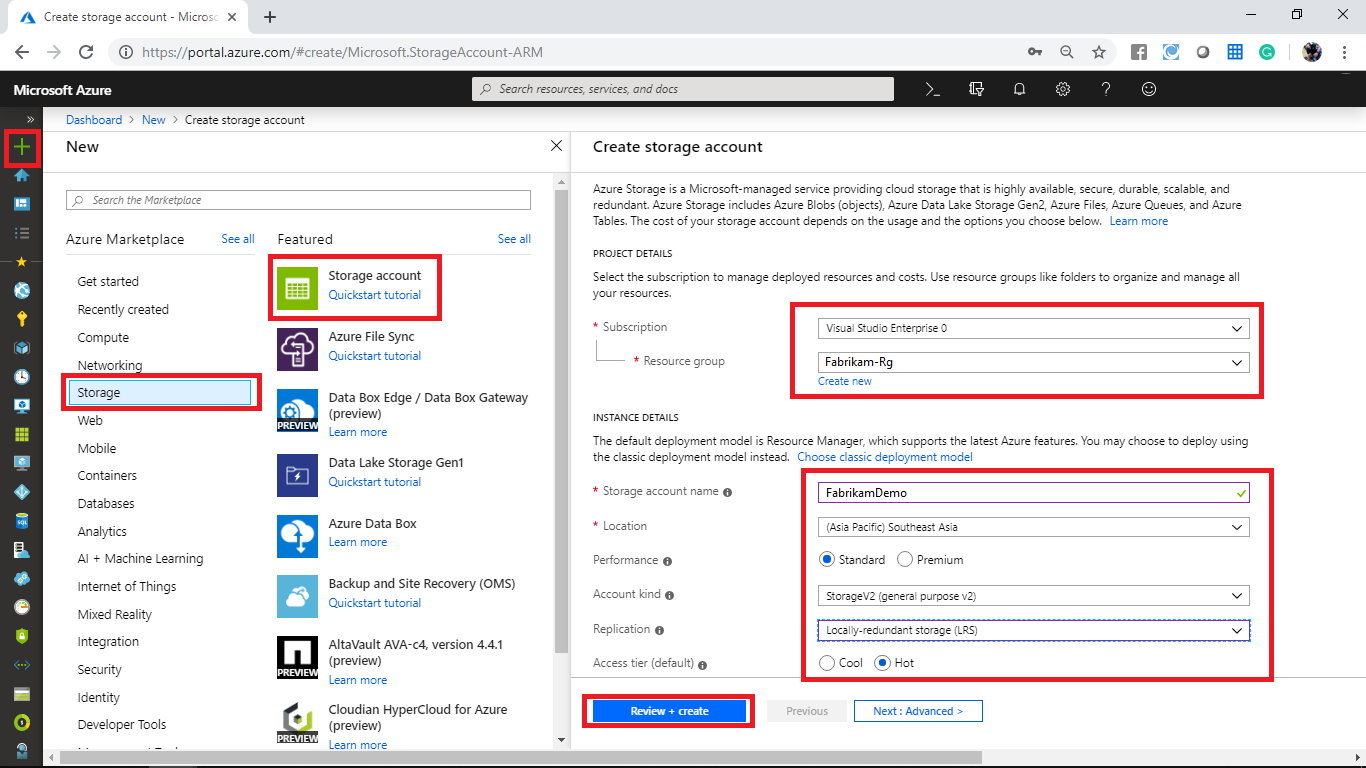


In the firewall blade, you will be able to see the client IP assigned if you are using the same SQL Server that you used while deploying SQL Server. If you have created new SQL Server for the SQL Database, then click on **+ Add client IP** to add the client IP.

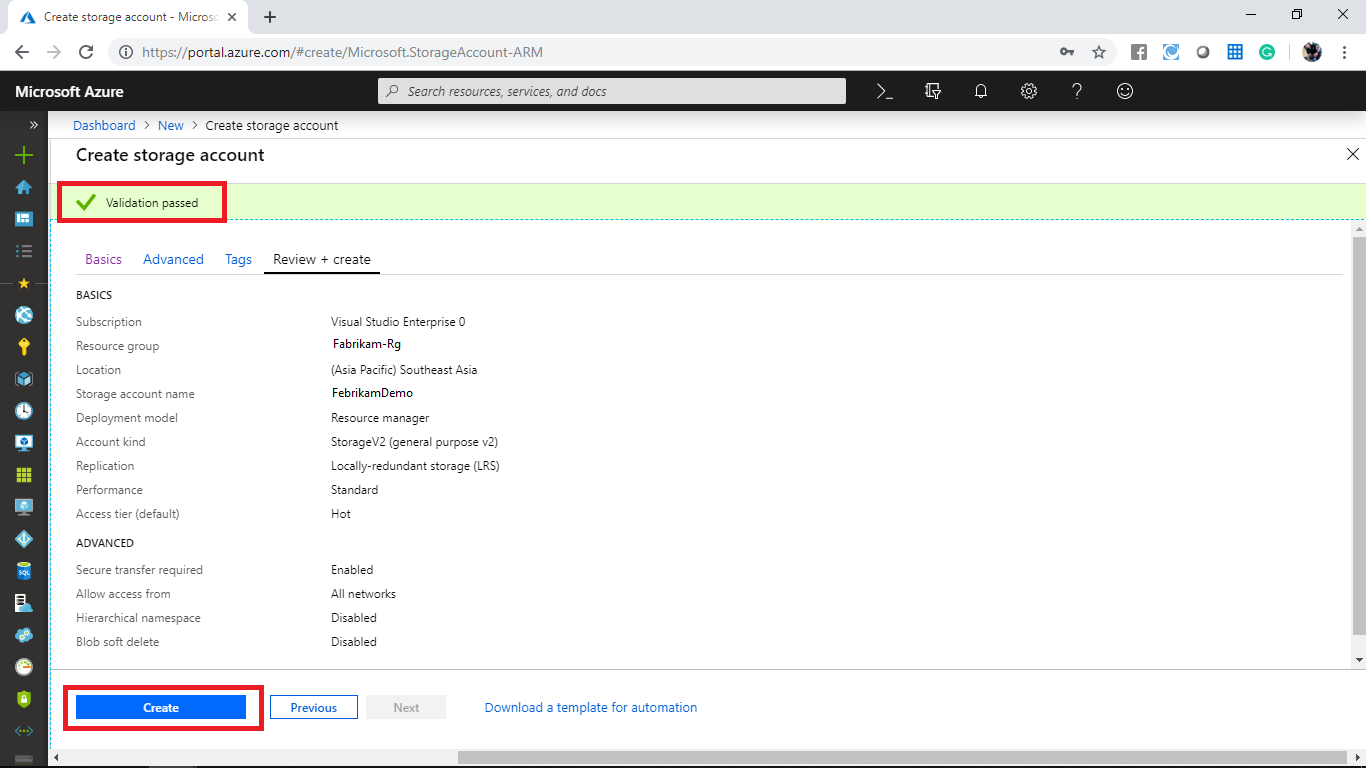


**Creating Storage Account:**

**Step 1:** Login to your Microsoft Azure portal, click on **Create new resource** and select **Storage account - blob, file, table, queue** from **Storage** category to create a **Storage account** and give the details. Choose a valid **Subscription**. For **Resource group** choose already available resource group and provide a unique name for the storage account. For **Location** choose the nearest to you and make PerformanceasStandard. Select **Account kind** as **StorageV2 (general purpose v2)**, **Replication** as **Locally-redundant storage** and **Access tier (default)** to **Hot**.

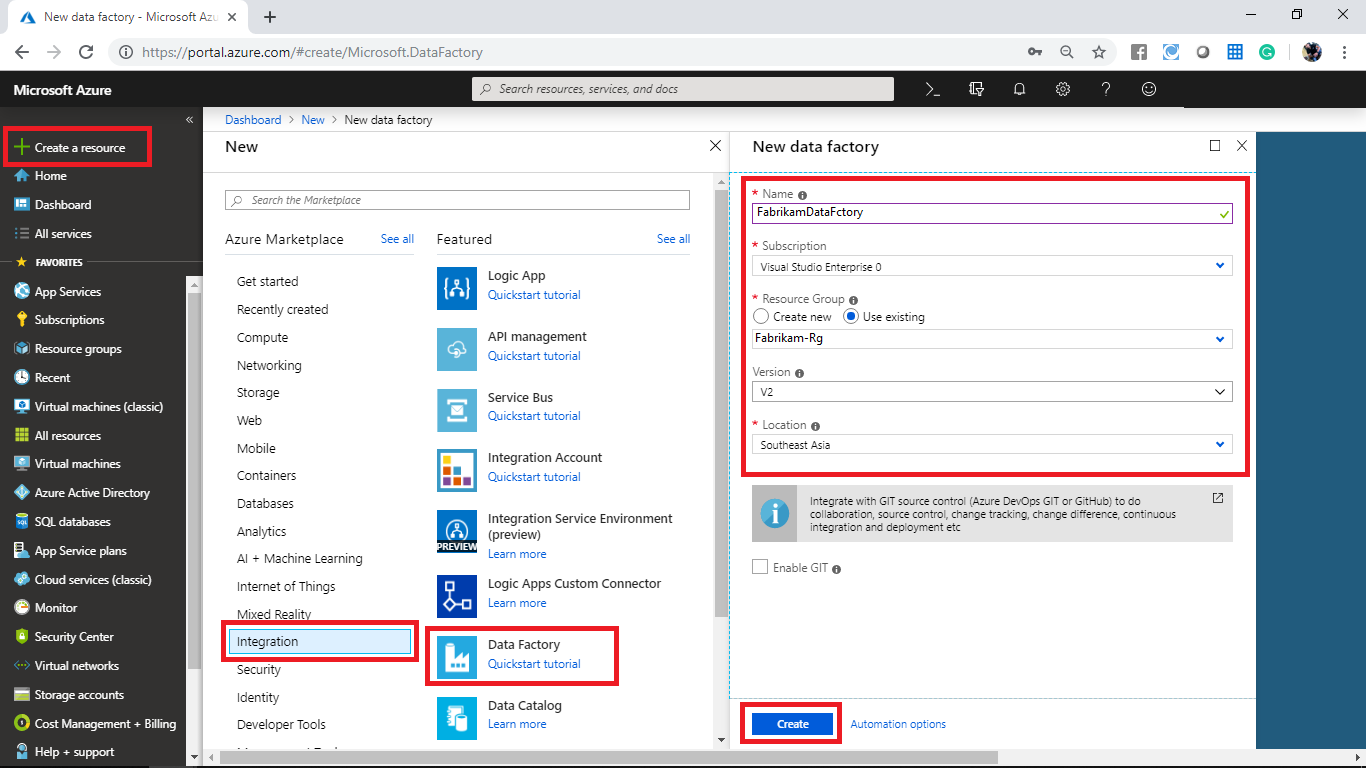


After the validation passes, click on **Create**.

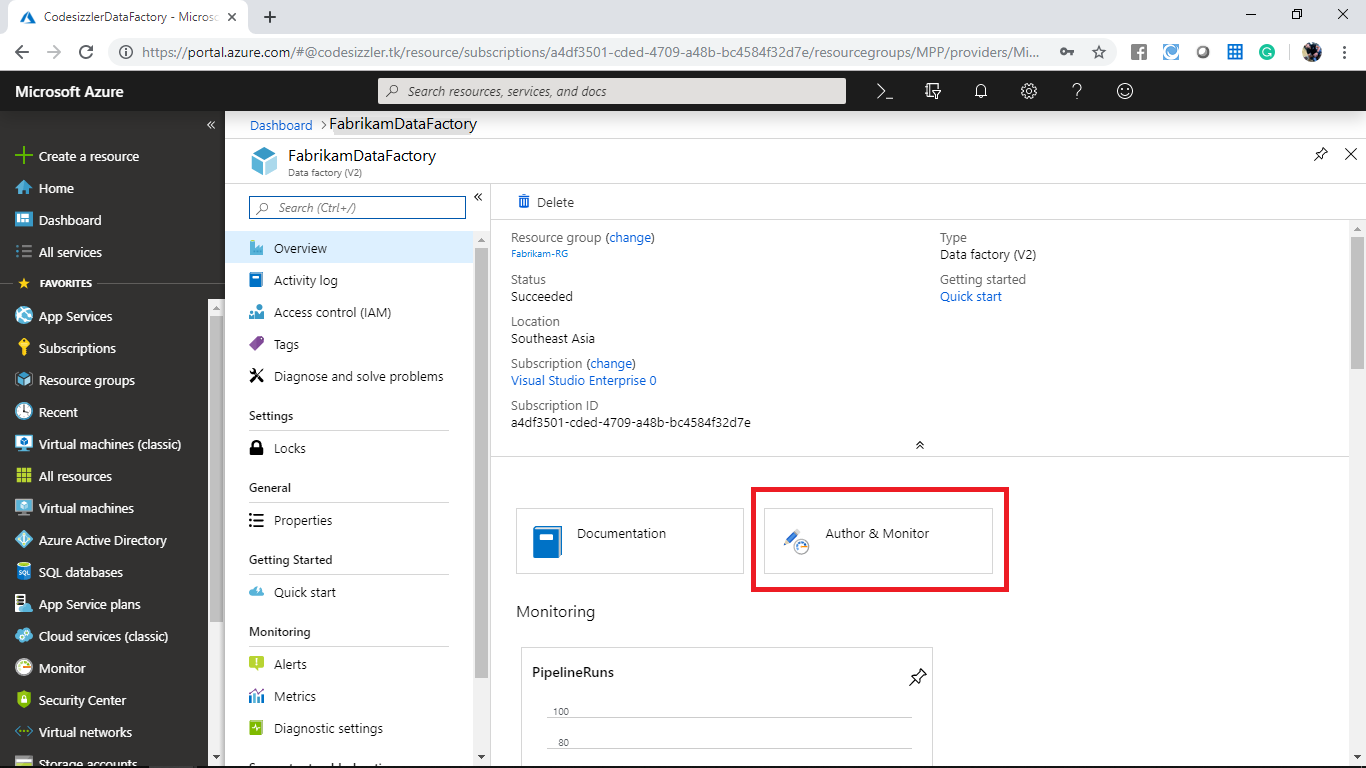


**Creating Data Factory:**

In the Azure portal, go to **+ Create Resource -> Analytics -> Data Factory.** Give a **name** for the resource, select a **subscription**, a **resource group**, **version**, **location** and click on **Create.**

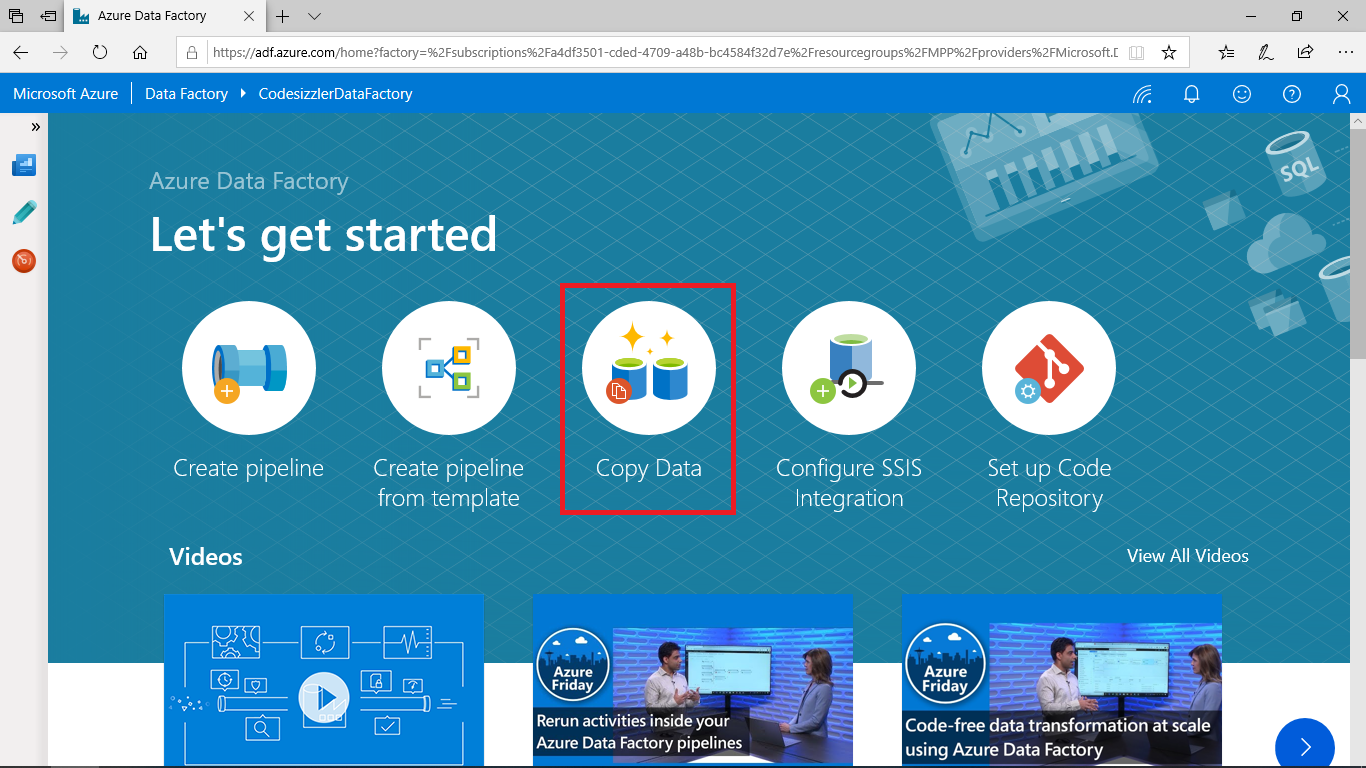


In the overview page of the data factory, click on **Author & Monitor** tile to navigate into management portal of azure data factory.

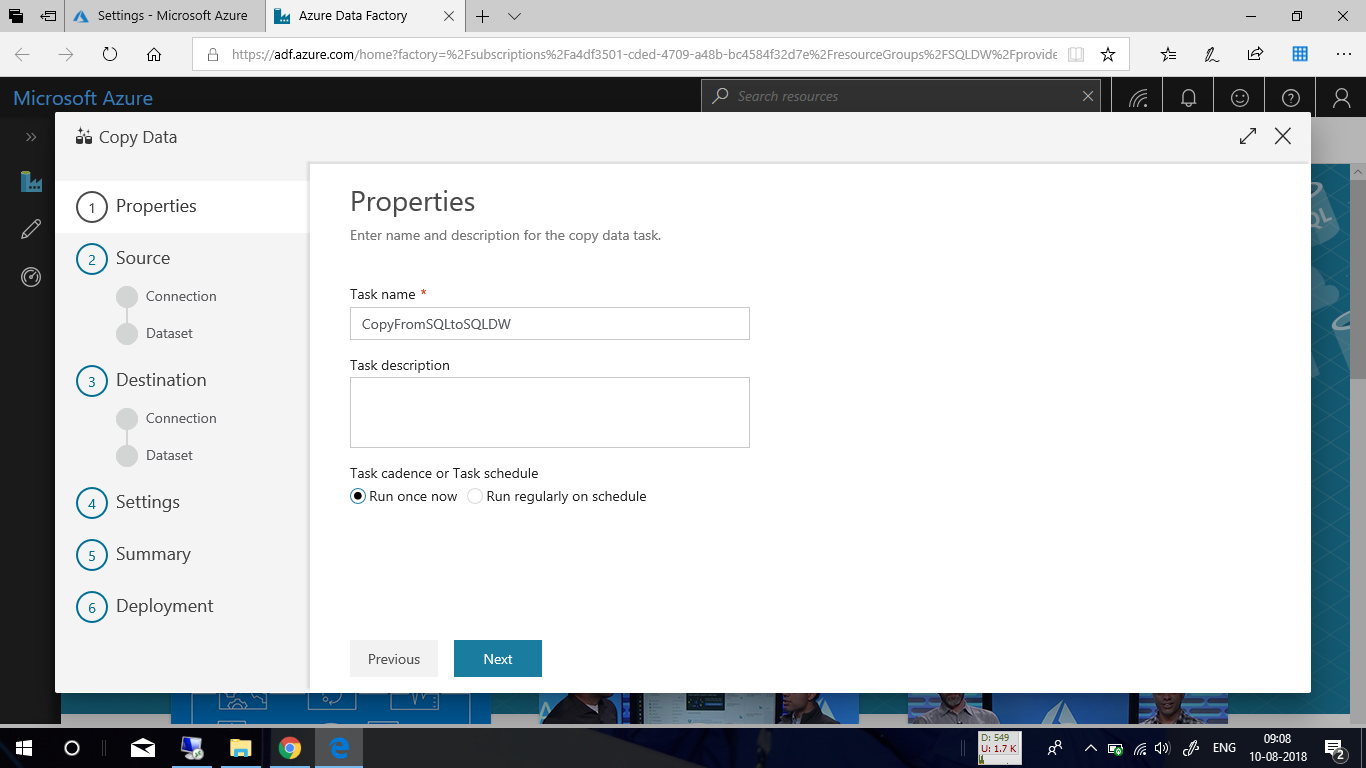


## **Load data into Azure SQL Data Warehouse**

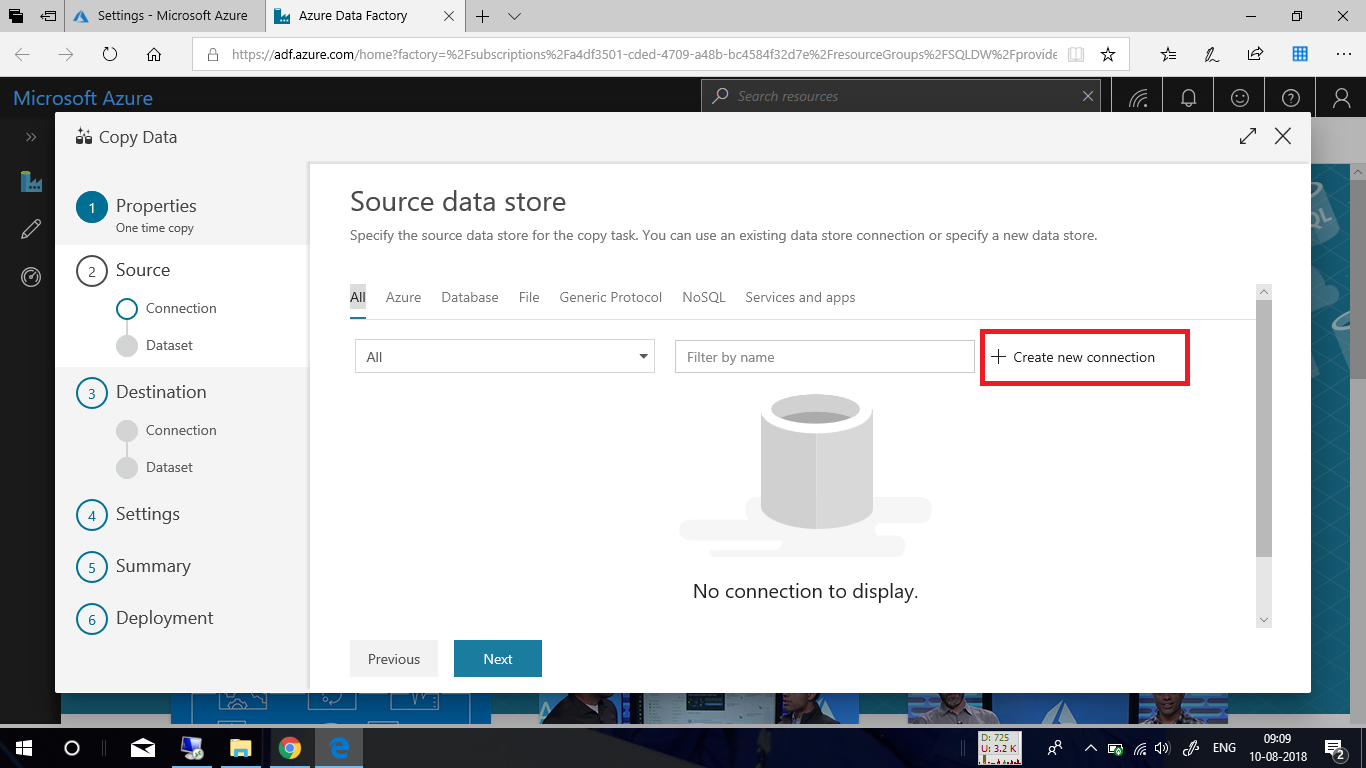
In the **Get started** page, select the **Copy Data** tile to launch the Copy Data tool.



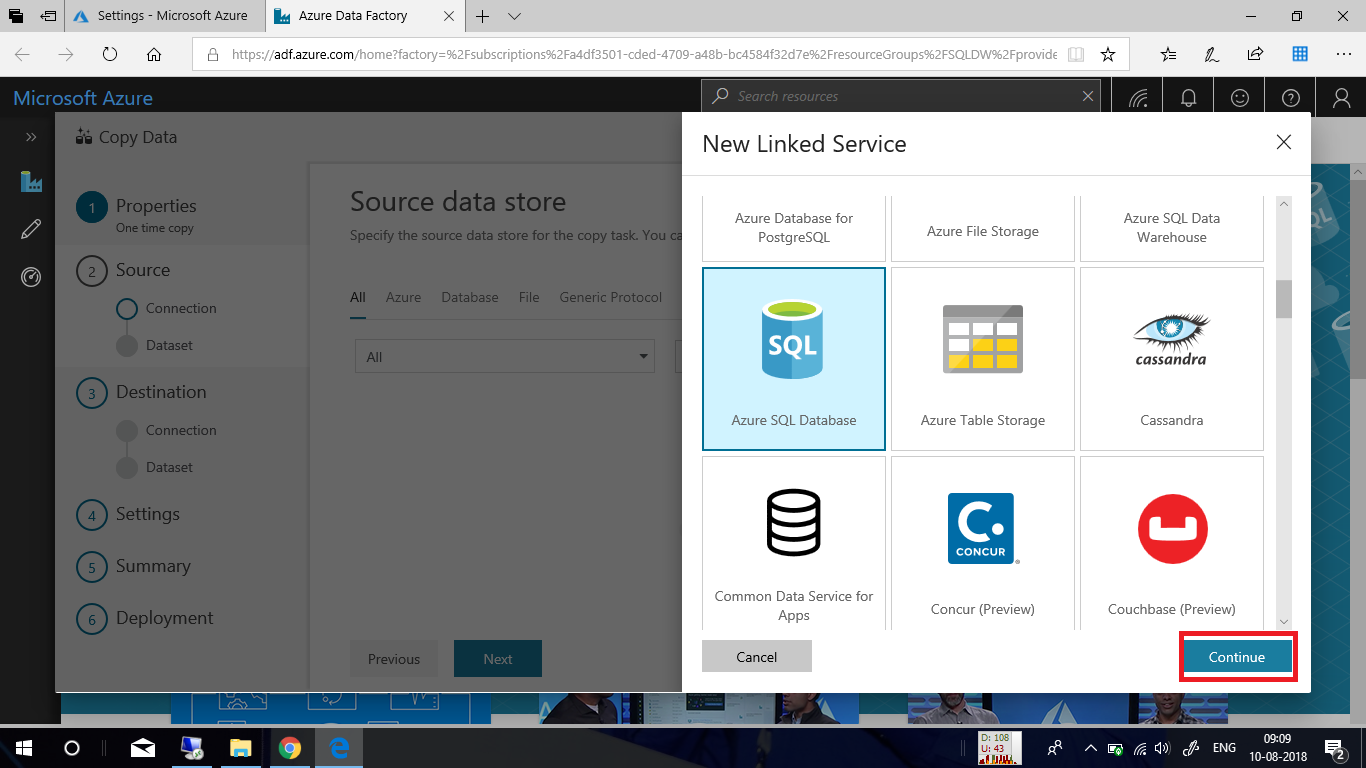
In the **Properties** page, specify **CopyFromSQLToSQLDW** for the **Task name** field, and select **Next**.



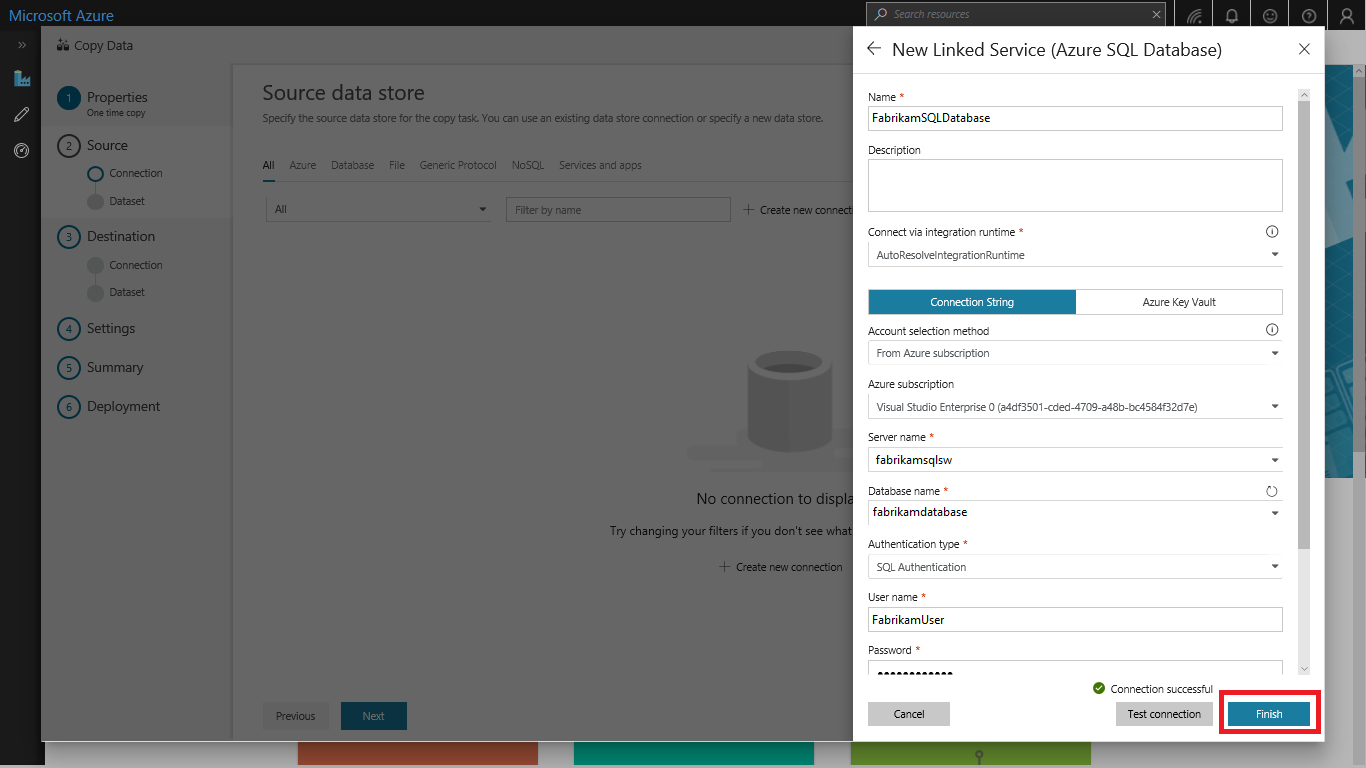
In the **Source data store** page, click **+ Create new connection**.



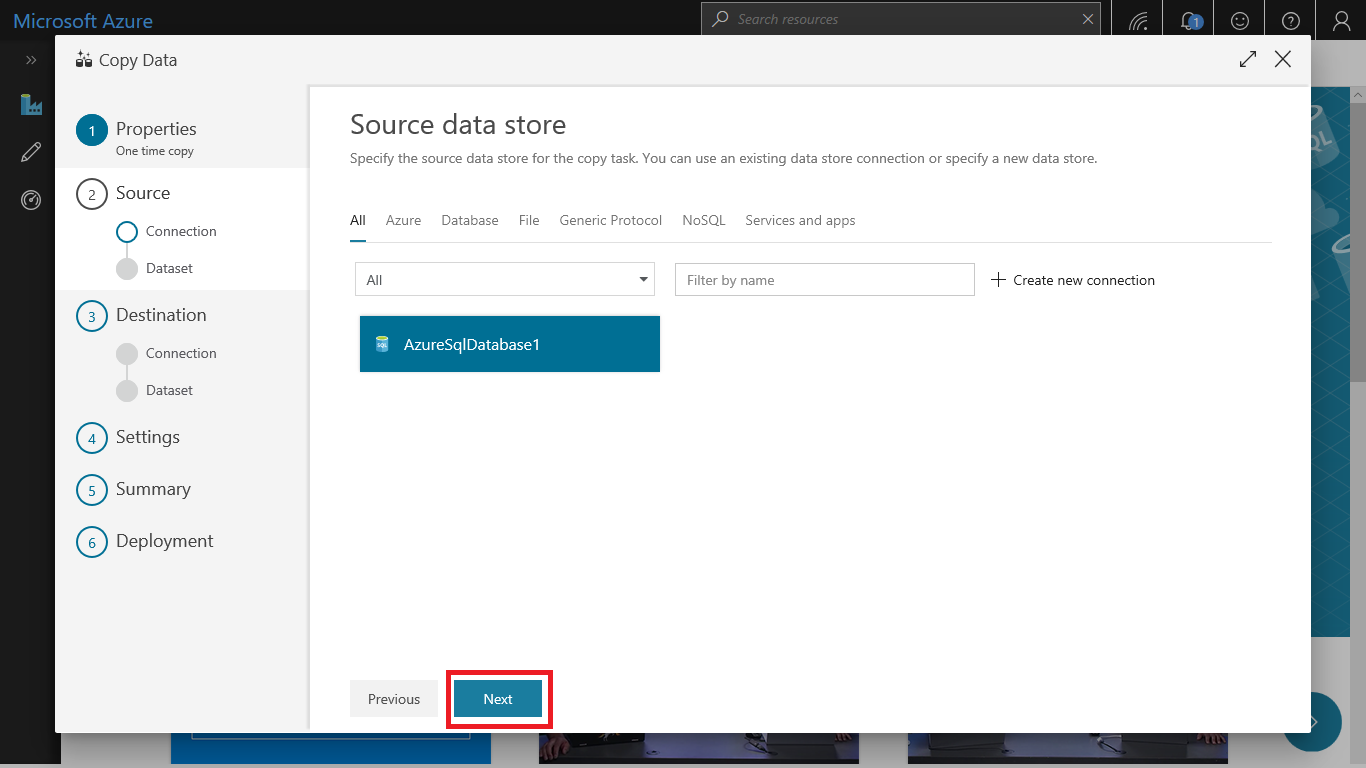
Select **Azure SQL Database** from the gallery and select **Continue**. You can type "SQL" in the search box to filter the connectors.



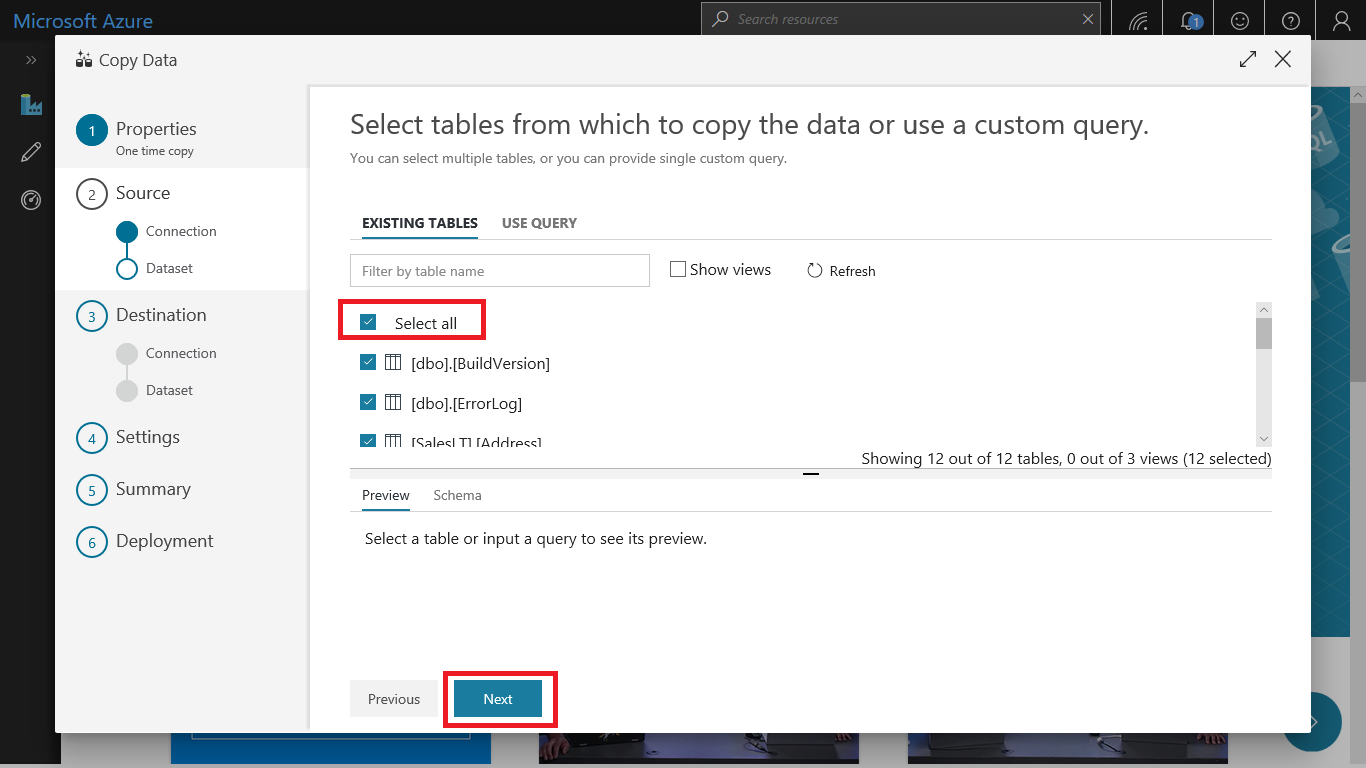
In the **New Linked Service** page, select your server name and DB name from the dropdown list, and specify the username and password. Click **Test connection** to validate the settings, then select **Finish**.



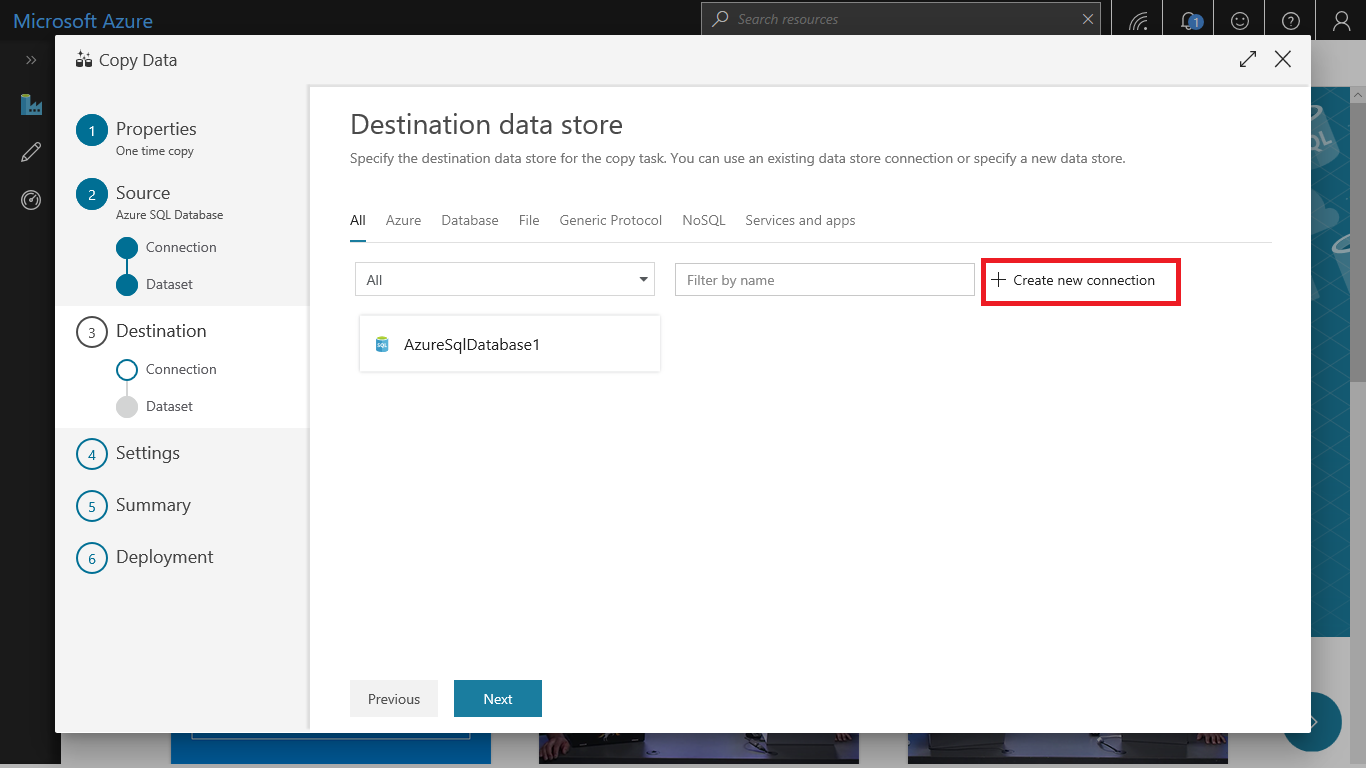
Select the newly created linked service as source, then click **Next**.



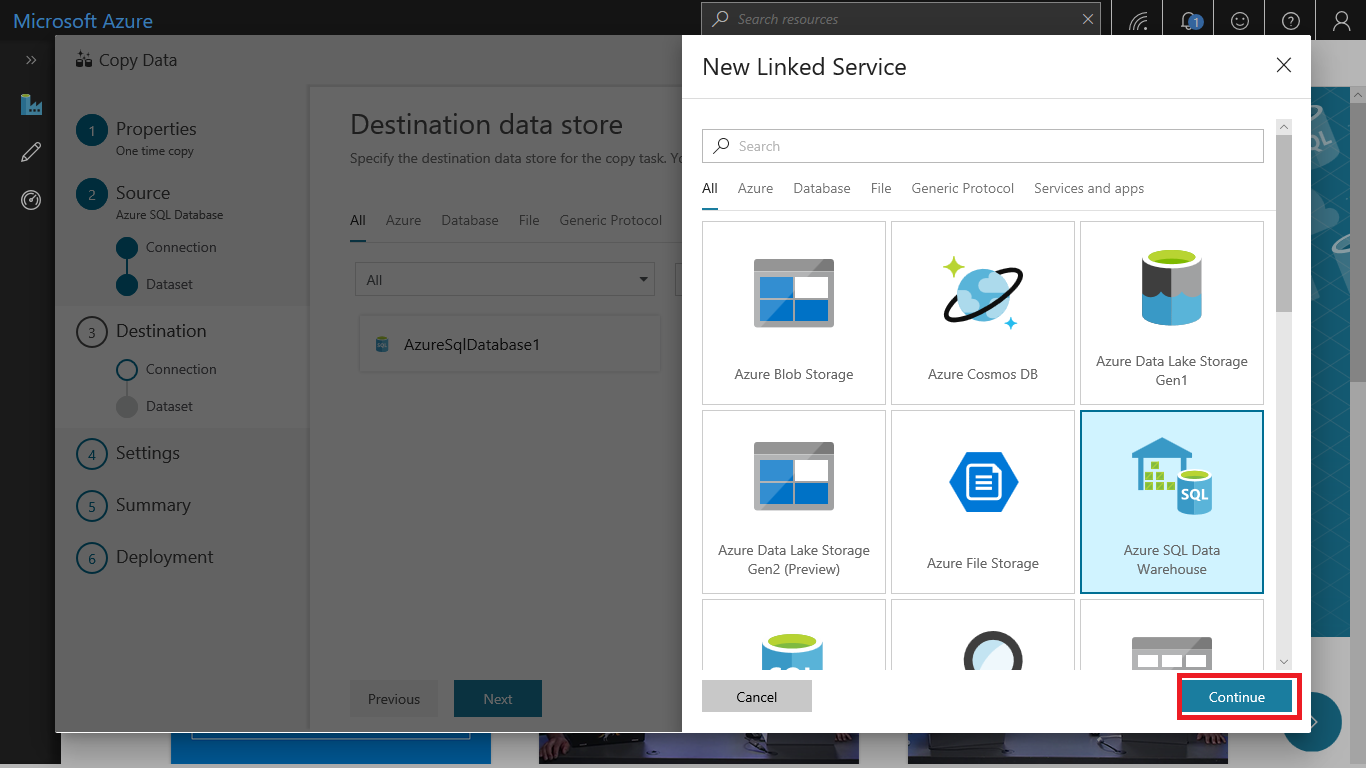
In the **Select tables from which to copy the data or use a custom query** page, enter **SalesLT** to filter the tables. Choose the **(Select all)** box to use all of the tables for the copy, and then select **Next**.



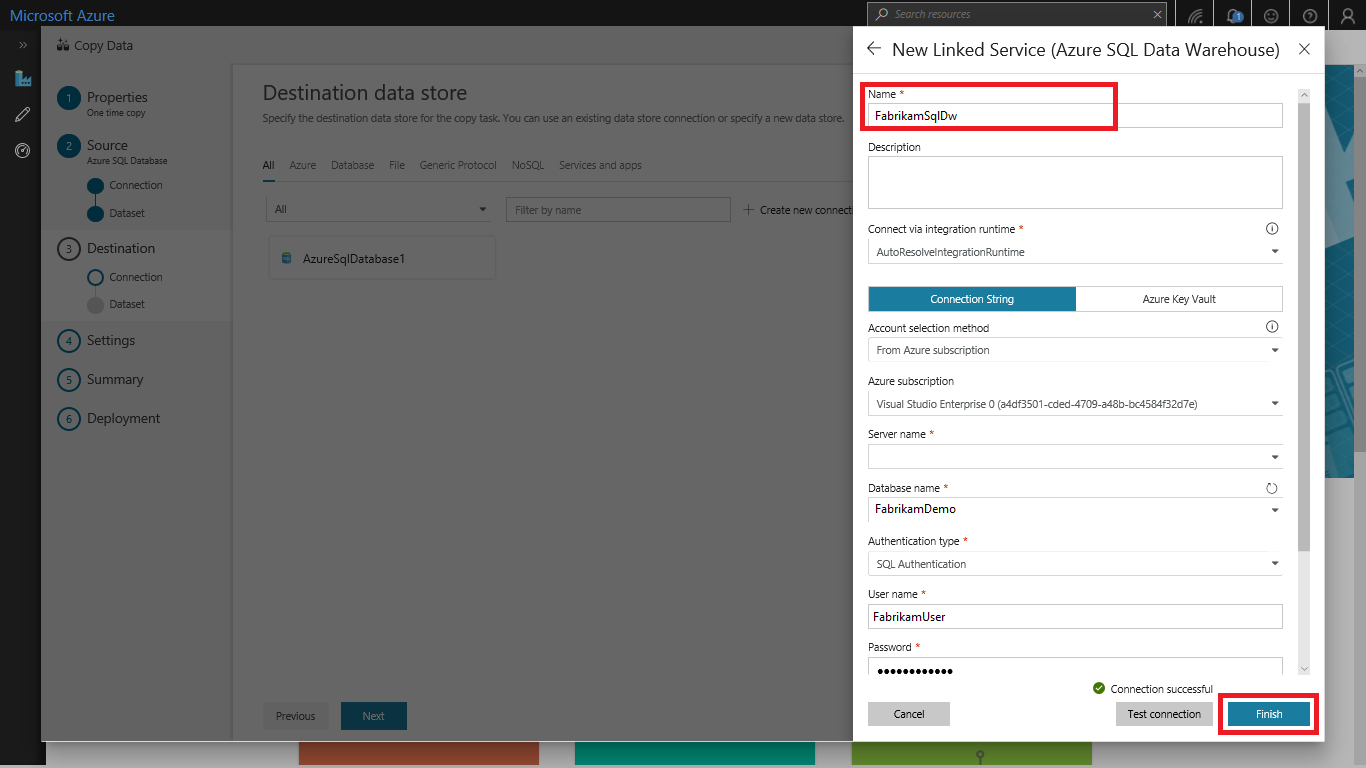
In the **Destination data store** page, click **+ Create new connection** to add a connection.



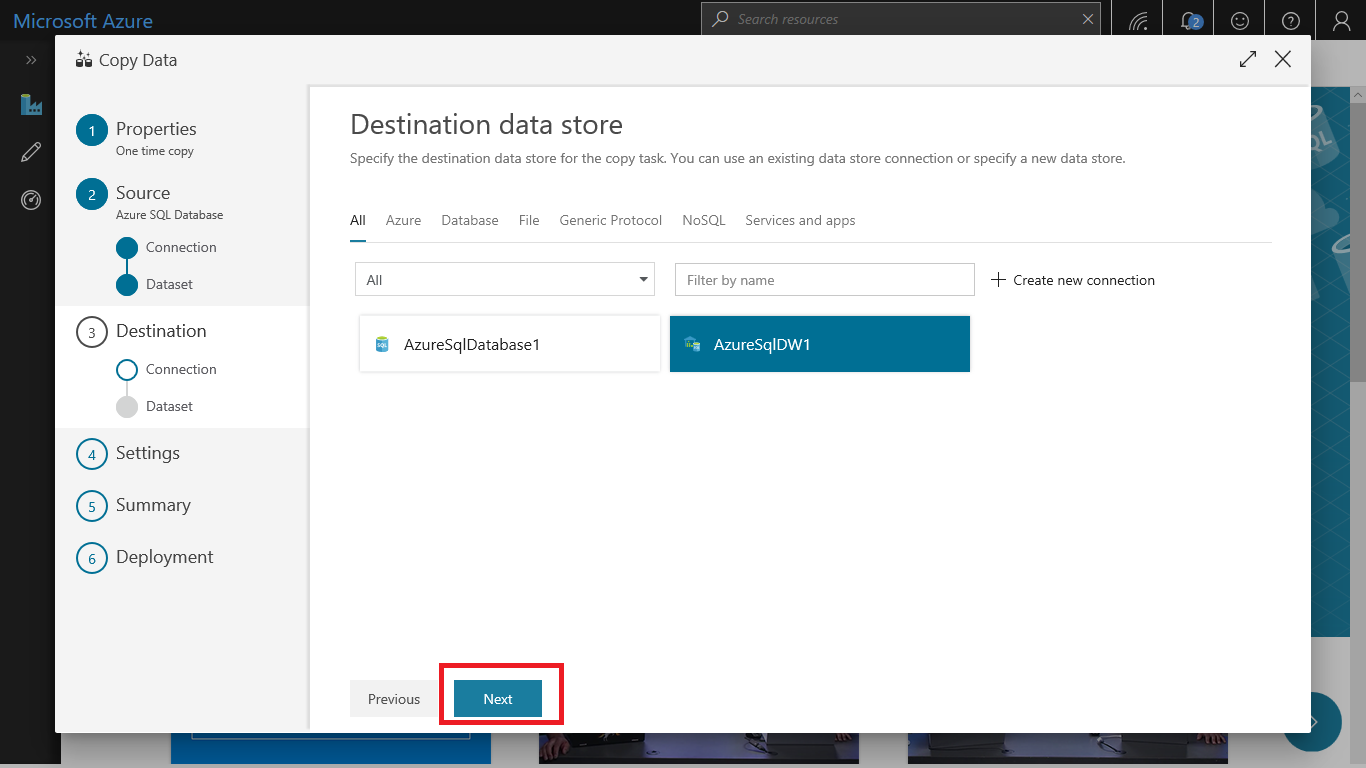
Select **Azure SQL Data Warehouse** from the gallery and select **Continue**.



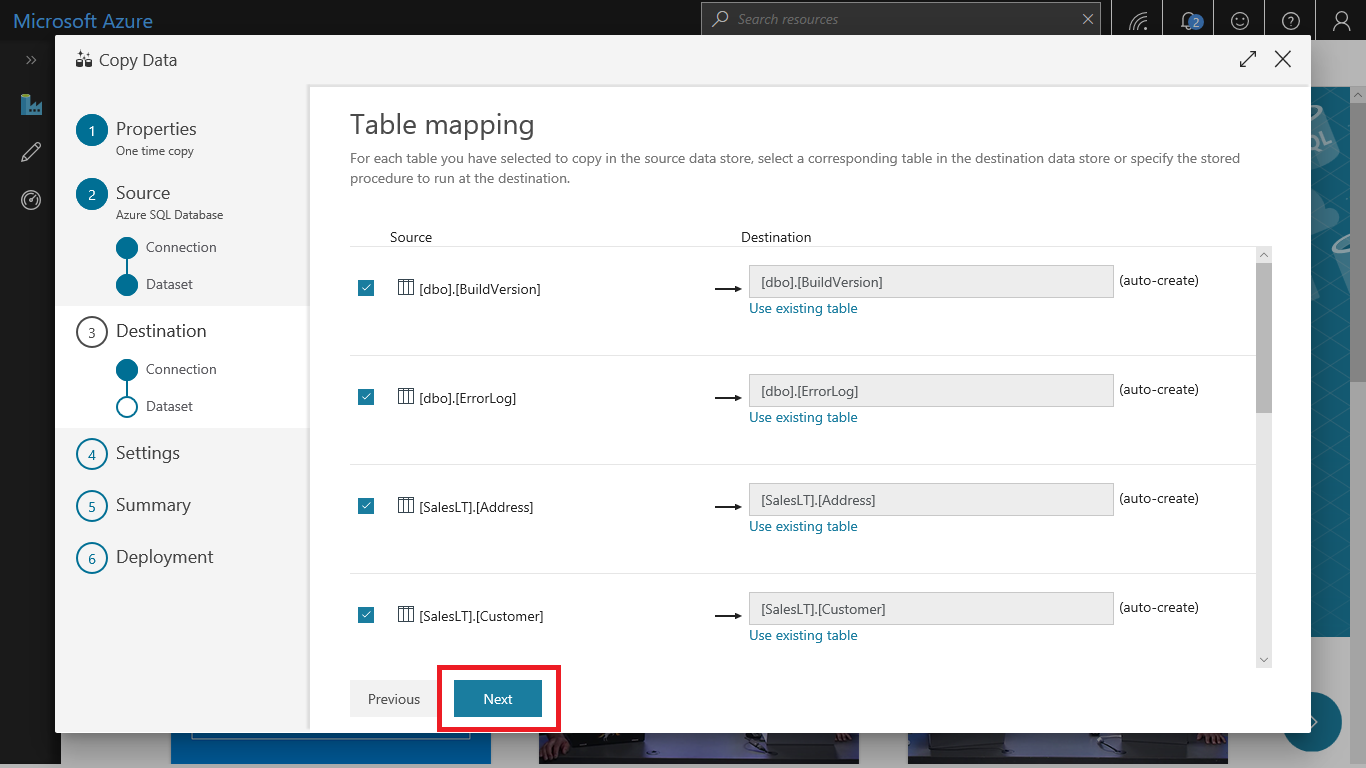
In the **New Linked Service** page, select your server name and DB name from the dropdown list, and specify the username and password. Click **Test connection** to validate the settings, then select **Finish**.



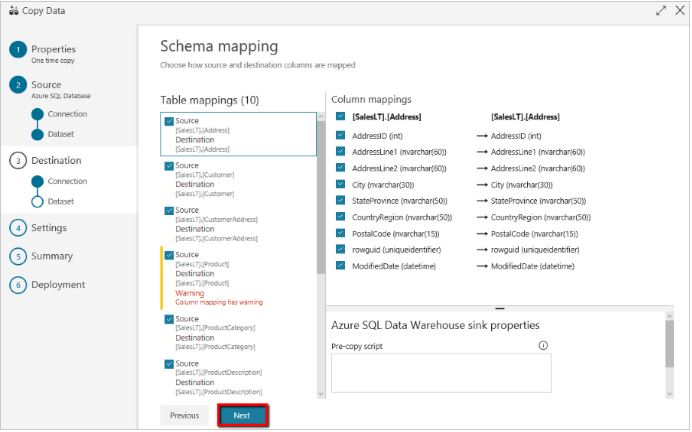
Select the newly created linked service as sink, then click **Next**.



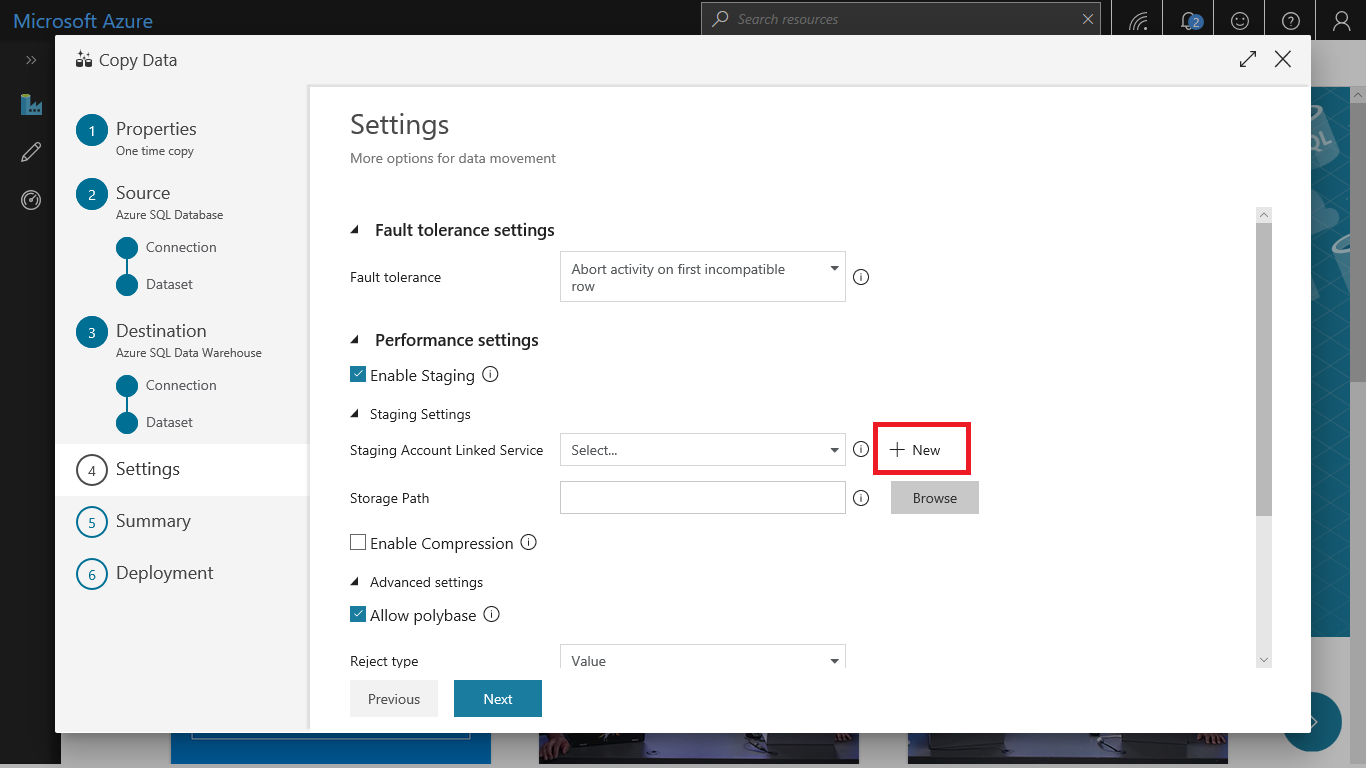
In the **Table mapping** page, review the content, and select **Next**. An intelligent table mapping display. The source tables are mapped to the destination tables based on the table names. If a source table doesn't exist in the destination, Azure Data Factory creates a destination table with the same name by default. You can also map a source table to an existing destination table.



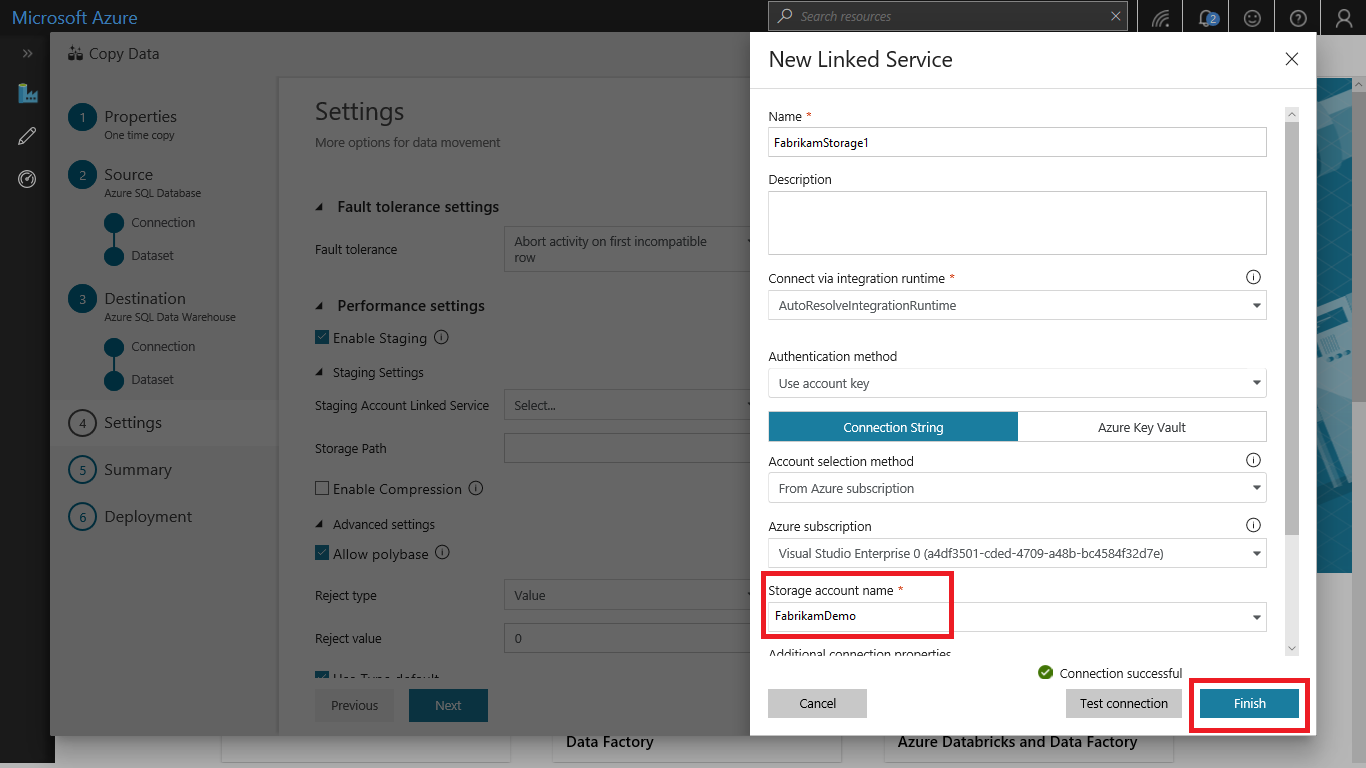
In the **Schema mapping** page, review the content, and select **Next**. The intelligent table mapping is based on the column name. If you let Data Factory automatically create the tables, data type conversion can occur when there are incompatibilities between the source and destination stores. If there's an unsupported data type conversion between the source and destination column, you see an error message next to the corresponding table.



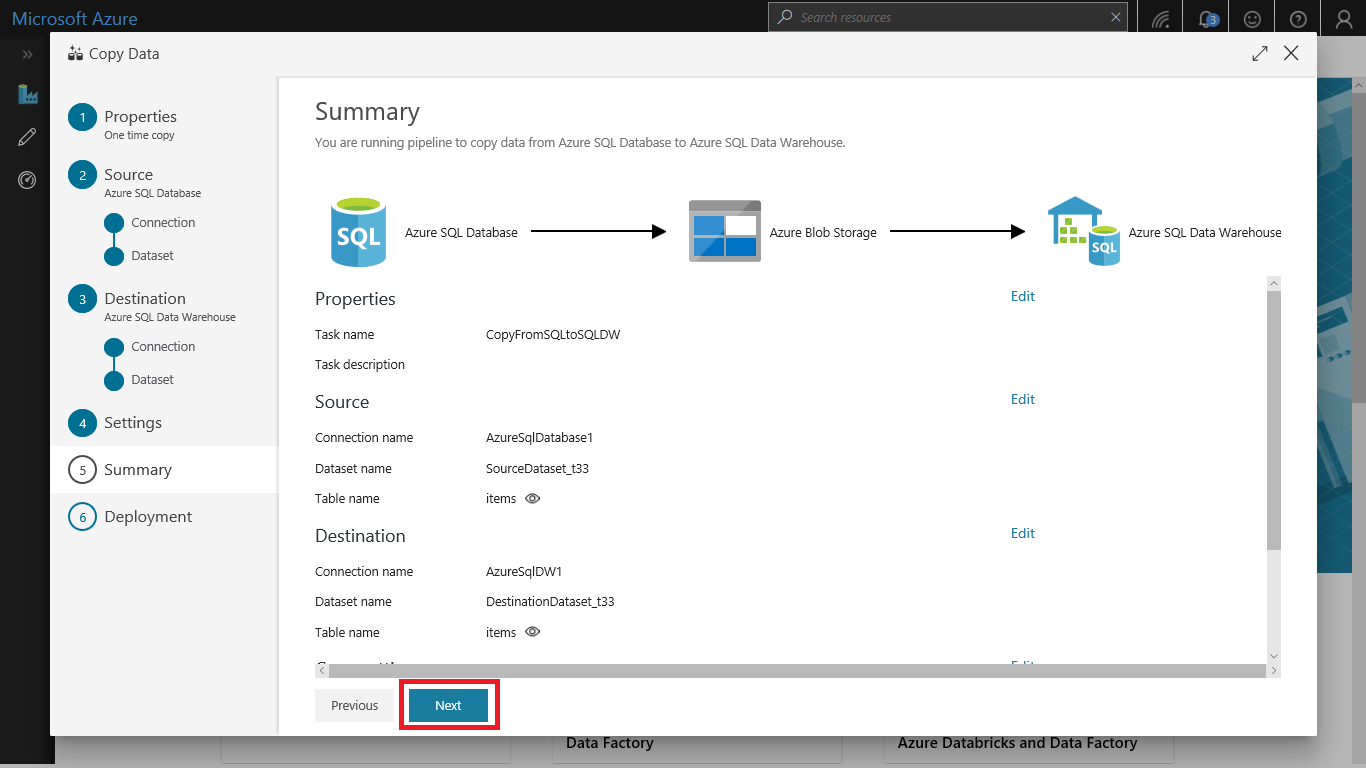
In **Staging settings** section, click **+ New** to new a staging storage. The storage is used for staging the data before it loads into SQL Data Warehouse by using PolyBase. After the copy is complete, the interim data in Azure Storage is automatically cleaned up.



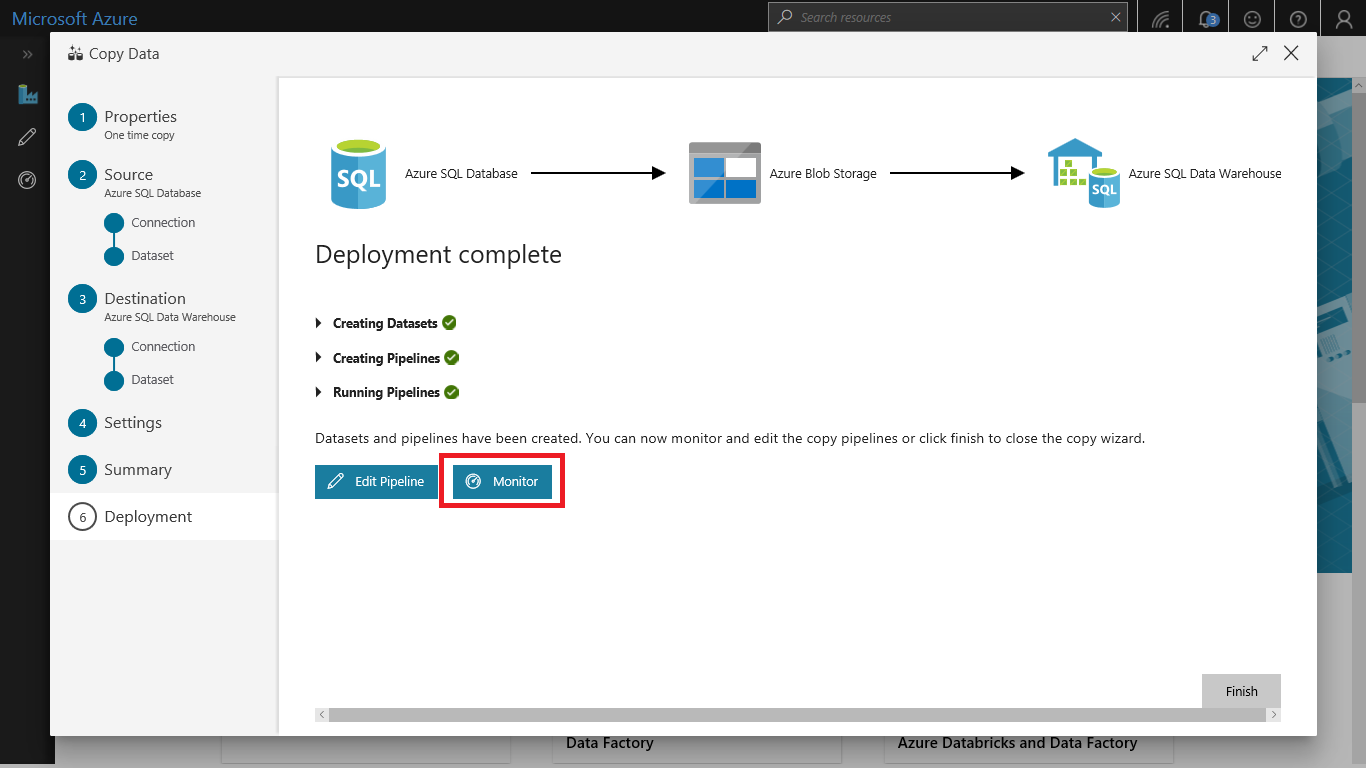
In the **New Linked Service** page, select your storage account, and select **Finish. Then click Next.**



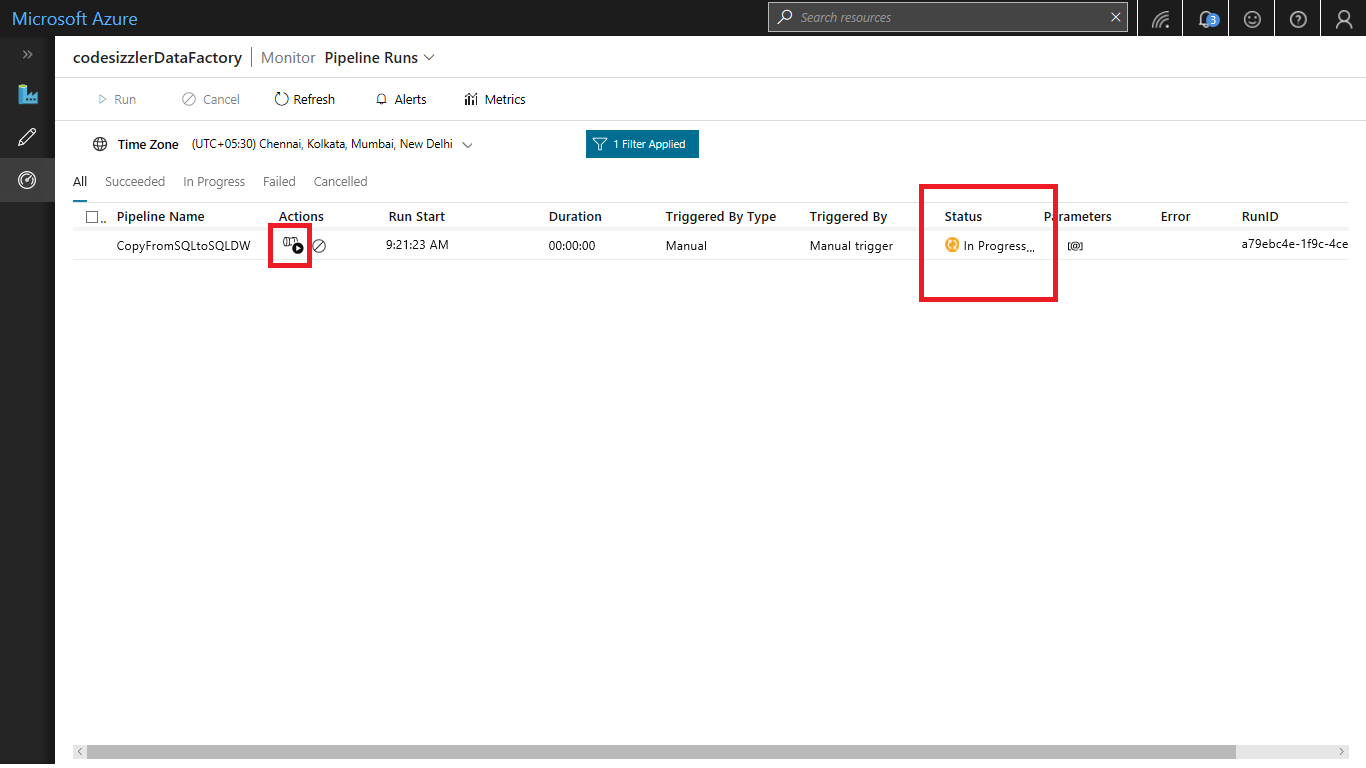
In the **Summary** page, review the settings, and select **Next**.



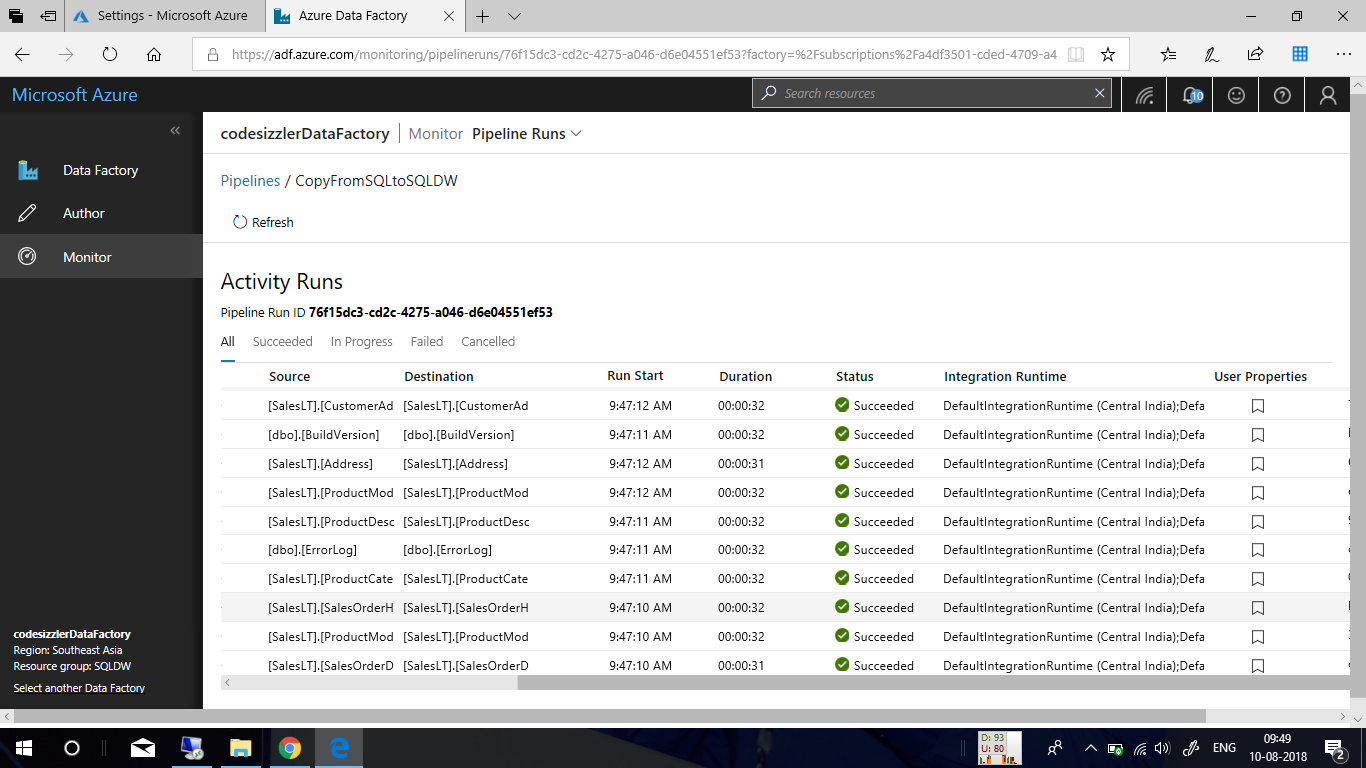
In the **Deployment page**, select **Monitor** to monitor the pipeline.



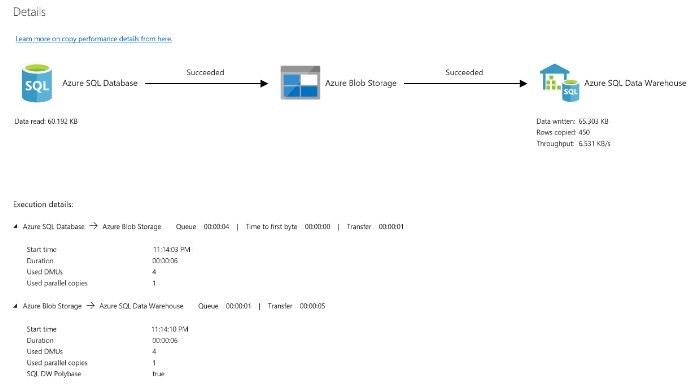
Notice that the **Monitor** tab on the left is automatically selected. The **Actions** column includes links to view activity run details and to rerun the pipeline.



To view activity runs that are associated with the pipeline run, select the **View Activity Runs** link in the **Actions** column. To switch back to the pipeline runs view, select the **Pipelines** link at the top. Select **Refresh** to refresh the list.



To monitor the execution details for each copy activity, select the **Details** link under **Actions** in the activity monitoring view. You can monitor details like the volume of data copied from the source to the sink, data throughput, execution steps with corresponding duration, and used configurations.



**Summary:**

With this demo, the team of Fabrikam has got a clear picture about uploading their workloads from SQL Server Database to Azure SQL Data Warehouse. This made the team to clear all their concerns for data movement in Azure. Now the team is looking forward for a concept where they can load terabytes of data to Azure SQL Datawarehouse with very minimal time span.