

➤ **Vendor: Microsoft**

➤ **Exam Code: DP-200**

➤ **Exam Name: Implementing an Azure Data Solution**

➤ **New Updated Questions from [Braindump2go](#) (Updated in [May/2020](#))**

### [Visit Braindump2go and Download Full Version DP-200 Exam Dumps](#)

#### QUESTION 153

You are a data engineer for your company. Your company has an on-premises SQL Server instance that contains 16 databases. Four of the databases require Common Language Runtime (CLR) features. You must be able to manage each database separately because each database has its own resource needs. You plan to migrate these databases to Azure. You want to migrate the databases by using a backup and restore process by using SQL commands. You need to choose the most appropriate deployment option to migrate the databases. What should you use?

- A. Azure SQL Database with an elastic pool
- B. Azure Cosmos DB with the SQL (DocumentDB) API
- C. Azure SQL Database managed instance
- D. Azure Cosmos DB with the Table API

**Answer: C**

#### **Explanation:**

You should use an Azure SQL Database managed instance deployment. This deployment option is almost 100% compatible with an on-premises instance, including the ability to use CLR features. When you back up the databases on-premises, you can execute a restore command to migrate the databases in Azure. This is referred to as lift and shift. You should not use an Azure Cosmos DB with the SQL (DocumentDB) API deployment. Cosmos DB is a multimodel database that supports five APIs for storage and queries, including SQL, Table, Cassandra, Gremlin, and MongoDB. The SQL API allows you to access data by using SQL-like queries. You cannot restore SQL Server databases to Cosmos DB by using SQL commands. You should not use an Azure Cosmos DB with the Table API deployment. The Table API is similar to Azure Tables. This deployment is useful if you are migrating an application from Azure Tables to Cosmos DB. With Azure Tables, you can access data by using Language Integrated Query (LINQ) and OData. You cannot restore SQL Server databases to Cosmos DB by using SQL commands. You should not use an Azure SQL Database with an elastic pool deployment. An elastic pool allows you to deploy multiple databases to a single logical instance and have all databases share a pool of resources. You configure the resource usage up front by choosing a purchasing model. You cannot take advantage of CLR features with an elastic pool.

#### QUESTION 154

##### SIMULATION

Use the following login credentials as needed:

Azure Username: xxxxx

Azure Password: xxxxx

The following information is for technical support purposes only:

Lab Instance: 10543936



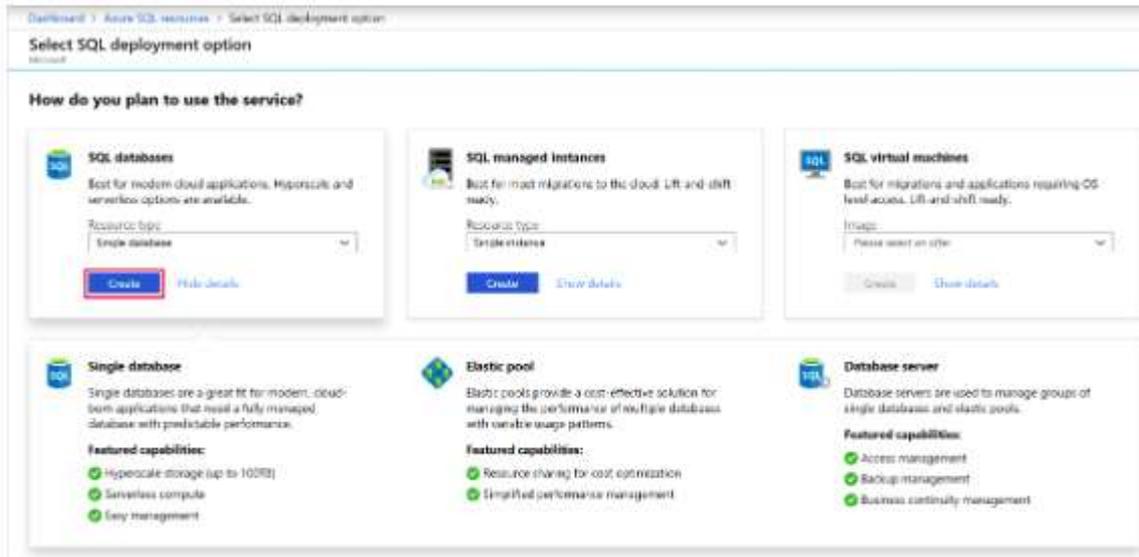
You need to create an elastic pool that contains an Azure SQL database named db2 and a new SQL database named db3.

**To complete this task, sign in to the Azure portal.**

**Answer:**

Step 1: Create a new SQL database named db3

1. Select SQL in the left-hand menu of the Azure portal. If SQL is not in the list, select All services, then type SQL in the search box.
2. Select + Add to open the Select SQL deployment option page. Select Single Database. You can view additional information about the different databases by selecting Show details on the Databases tile.
3. Select Create:



Deployment > Azure SQL resources > Select SQL deployment option

### Select SQL deployment option

Microsoft

How do you plan to use the service?

**SQL databases**

Best for modern cloud applications. Hyperscale and serverless options are available.

Resource type: Single database

[Create](#) [Show details](#)

**SQL managed instances**

Best for most migrations to the cloud. Lift-and-shift ready.

Resource type: Single instance

[Create](#) [Show details](#)

**SQL virtual machines**

Best for migrations and applications requiring OS-level access. Lift-and-shift ready.

Image: Please select an offer

[Create](#) [Show details](#)

---

**Single database**

Single databases are a great fit for modern, cloud-born applications that need a fully managed database with predictable performance.

**Featured capabilities:**

- ✓ Hyperscale storage (up to 100TB)
- ✓ Serverless compute
- ✓ Easy management

**Elastic pool**

Elastic pools provide a cost-effective solution for managing the performance of multiple databases with variable usage patterns.

**Featured capabilities:**

- ✓ Resource sharing for cost optimization
- ✓ Simplified performance management

**Database server**

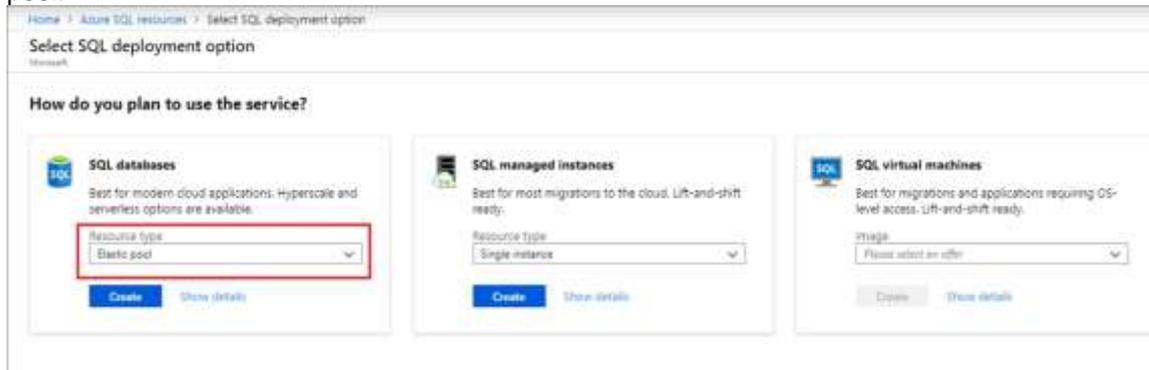
Database servers are used to manage groups of single databases and elastic pools.

**Featured capabilities:**

- ✓ Access management
- ✓ Backup management
- ✓ Business continuity management

4. Enter the required fields if necessary.
  5. Leave the rest of the values as default and select Review + Create at the bottom of the form.
  6. Review the final settings and select Create. Use Db3 as database name.
- On the SQL Database form, select Create to deploy and provision the resource group, server, and database.
- Step 2: Create your elastic pool using the Azure portal.

1. Select Azure SQL in the left-hand menu of the Azure portal. If Azure SQL is not in the list, select All services, then type Azure SQL in the search box.
2. Select + Add to open the Select SQL deployment option page.
3. Select Elastic pool from the Resource type drop-down in the SQL Databases tile. Select Create to create your elastic pool.



Home > Azure SQL resources > Select SQL deployment option

### Select SQL deployment option

Microsoft

How do you plan to use the service?

**SQL databases**

Best for modern cloud applications. Hyperscale and serverless options are available.

Resource type: Elastic pool

[Create](#) [Show details](#)

**SQL managed instances**

Best for most migrations to the cloud. Lift-and-shift ready.

Resource type: Single instance

[Create](#) [Show details](#)

**SQL virtual machines**

Best for migrations and applications requiring OS-level access. Lift-and-shift ready.

Image: Please select an offer

[Create](#) [Show details](#)

4. Configure your elastic pool with the following values:  
 Name: Provide a unique name for your elastic pool, such as myElasticPool.  
 Subscription: Select your subscription from the drop-down.  
 ResourceGroup: Select the resource group.  
 Server: Select the server

Home > Azure SQL resources > Select SQL deployment option > Create SQL Elastic pool

## Create SQL Elastic pool

Microsoft

**Basics** • Tags Review + create

Create a SQL Elastic pool with your preferred configurations. Elastic pools provide a simple and cost effective solution for managing the performance of multiple databases within a fixed budget. Complete the Basic tab, then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

\* Subscription

\* Resource group  [Create new](#)

### Elastic pool details

Enter required settings for this pool, including picking a logical server and configuring the compute and storage resources.

\* Elastic Pool Name  ✓

\* Server  [Create new](#)

\* Compute + storage **GeneralPurpose**  
 Gen5, 2 vCores, 32 GB, 0 databases  
[Configure elastic pool](#)

5. Select Configure elastic pool

6. On the Configure page, select the Databases tab, and then choose to Add database.

Home > Azure SQL resources > Select SQL deployment option > Create SQL Elastic pool > Configure

## Configure

Feedback

Looking for basic, standard, premium?

**General Purpose**  
 Scalable compute and storage options  
 Up to 7,000 IOPS  
 5-10 ms latency

**Add databases**

Select all

Search to filter databases...

DATABASE NAME

mySampleDatabase

Pool settings **Databases** Per database settings

[+ Add databases](#) [Revert selected](#)

Search to filter databases...

DATABASE NAME	PRICING TIER
Currently, there are no databases selected to be added to the pool. To add databases, click 'Add databases' above.	

7. Add the Azure SQL database named db2, and the new SQL database named db3 that you created in Step 1.

8. Select Review + create to review your elastic pool settings and then select Create to create your elastic pool.

Reference:

<https://docs.microsoft.com/bs-latn-ba/azure/sql-database/sql-database-elastic-pool-failover-group-tutorial>

### QUESTION 155

#### SIMULATION

Use the following login credentials as needed:

[DP-200 Exam Dumps](#) [DP-200 Exam Questions](#) [DP-200 PDF Dumps](#) [DP-200 VCE Dumps](#)

<https://www.braindump2go.com/dp-200.html>

Azure Username: xxxxx  
Azure Password: xxxxx  
The following information is for technical support purposes only:  
Lab Instance: 10543936



You need to create an Azure Storage account named account10543936. The solution must meet the following requirements:

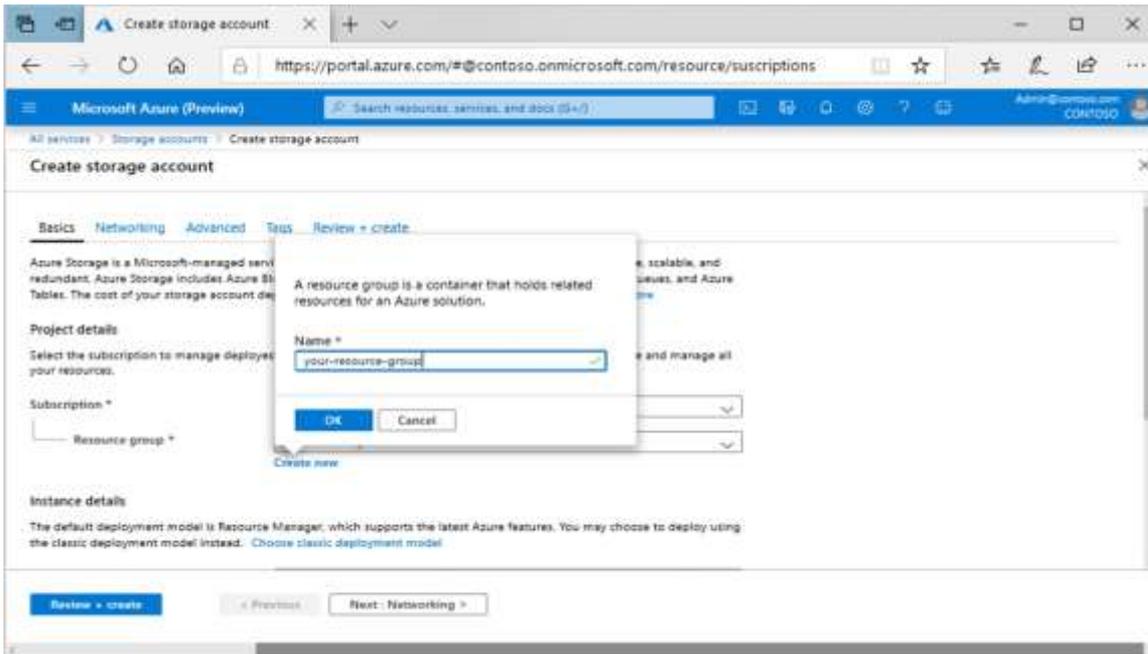
- Minimize storage costs.
- Ensure that account10543936 can store many image files.
- Ensure that account10543936 can quickly retrieve stored image files.

**To complete this task, sign in to the Azure portal.**

**Answer:**

Create a general-purpose v2 storage account, which provides access to all of the Azure Storage services: blobs, files, queues, tables, and disks.

1. On the Azure portal menu, select All services. In the list of resources, type Storage Accounts. As you begin typing, the list filters based on your input. Select Storage Accounts.
2. On the Storage Accounts window that appears, choose Add.
3. Select the subscription in which to create the storage account.
4. Under the Resource group field, select Create new. Enter the name for your new resource group, as shown in the following image.



5. Next, enter the name account10543936 for your storage account.
  6. Select a location for your storage account, or use the default location.
  7. Leave these fields set to their default values:  
Deployment model: Resource Manager  
Performance: Standard  
Account kind: StorageV2 (general-purpose v2)  
Replication: Read-access geo-redundant storage (RA-GRS)  
Access tier: Hot
  8. Select Review + Create to review your storage account settings and create the account.
  9. Select Create.
- Reference:  
<https://docs.microsoft.com/en-us/azure/storage/common/storage-account-create>

#### **QUESTION 156**

##### **SIMULATION**

Use the following login credentials as needed:

Azure Username: xxxxx

Azure Password: xxxxx

The following information is for technical support purposes only:

Lab Instance: 10543936



You need to ensure that users in the West US region can read data from a local copy of an Azure Cosmos DB database named cosmos10543936.

**To complete this task, sign in to the Azure portal.**

**NOTE: This task might take several minutes to complete. You can perform other tasks while the task completes or end this section of the exam.**

**Answer:**

You can enable Availability Zones by using Azure portal when creating an Azure Cosmos account.

You can enable Availability Zones by using Azure portal.

Step 1: enable the Geo-redundancy, Multi-region Writes

1. In Azure Portal search for and select Azure Cosmos DB.

2. Locate the Cosmos DB database named cosmos10543936

3. Access the properties for cosmos10543936

4. enable the Geo-redundancy, Multi-region Writes.

Location: West US region

**Instance Details**

\* Account Name:  ✓

\* API:  ▾

Apache Spark:     
 You're on the waitlist for Azure Cosmos with support for Apache Spark preview

\* Location:  ▾

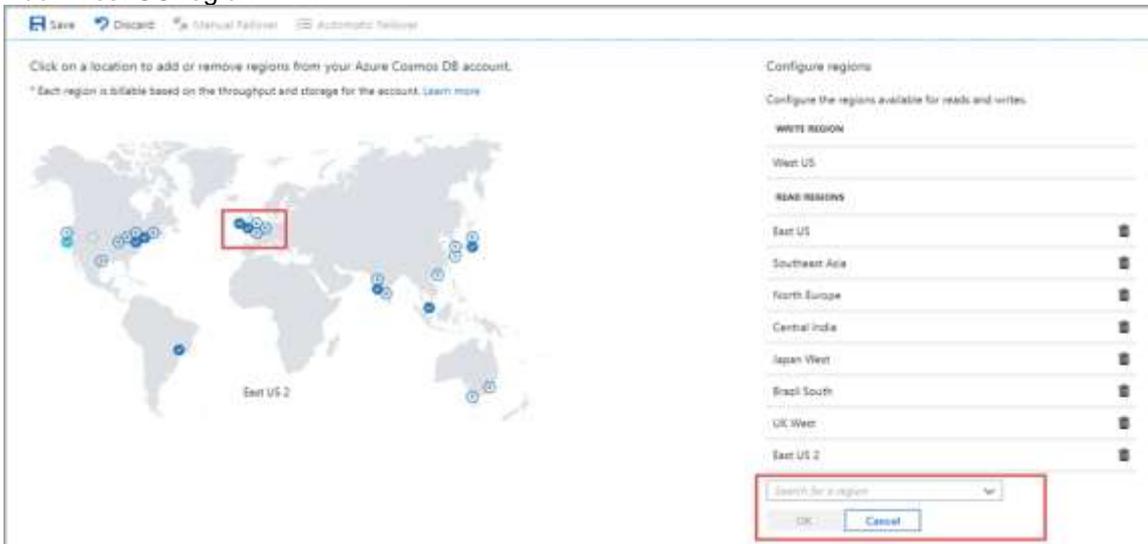
Geo-Redundancy:

Multi-region Writes:

Availability Zones:

Step 2: Add region from your database account

1. In to Azure portal, go to your Azure Cosmos account, and open the Replicate data globally menu.
2. To add regions, select the hexagons on the map with the + label that corresponds to your desired region(s). Alternatively, to add a region, select the + Add region option and choose a region from the drop-down menu. Add: West US region



3. To save your changes, select OK.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/high-availability>

<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-manage-database-account>

**QUESTION 157**

**SIMULATION**

Use the following login credentials as needed:

Azure Username: xxxxx

Azure Password: xxxxx

The following information is for technical support purposes only:

Lab Instance: 10543936



You plan to enable Azure Multi-Factor Authentication (MFA).

You need to ensure that User1-10543936@ExamUsers.com can manage any databases hosted on an Azure SQL server named SQL10543936 by signing in using his Azure Active Directory (Azure AD) user account.

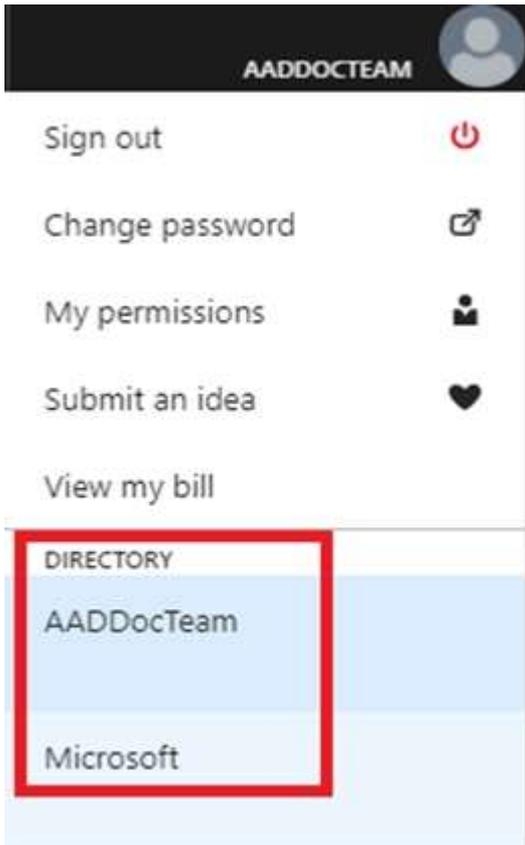
**To complete this task, sign in to the Azure portal.**

**Answer:**

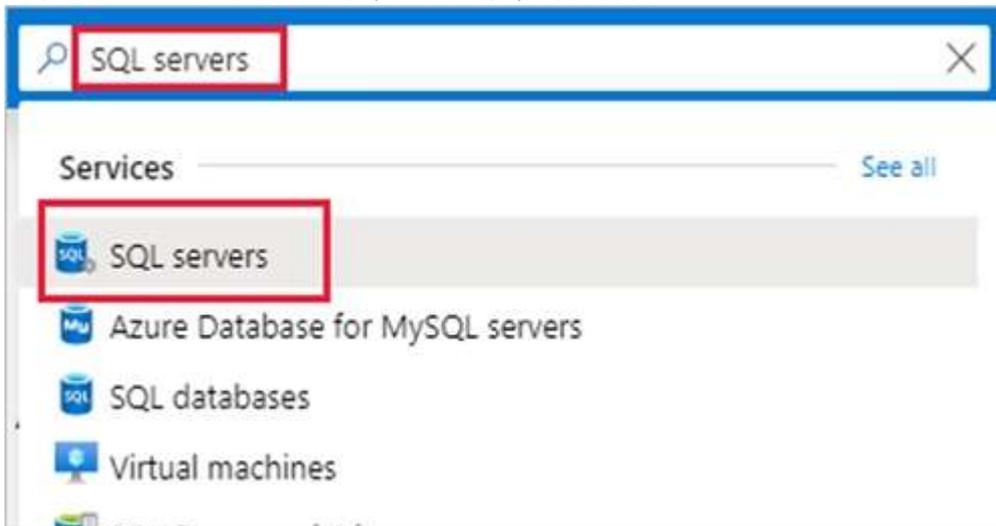
Provision an Azure Active Directory administrator for your managed instance

Each Azure SQL server (which hosts a SQL Database or SQL Data Warehouse) starts with a single server administrator account that is the administrator of the entire Azure SQL server. A second SQL Server administrator must be created, that is an Azure AD account. This principal is created as a contained database user in the master database.

1. In the Azure portal, in the upper-right corner, select your connection to drop down a list of possible Active Directories. Choose the correct Active Directory as the default Azure AD. This step links the subscription-associated Active Directory with Azure SQL server making sure that the same subscription is used for both Azure AD and SQL Server. (The Azure SQL server can be hosting either Azure SQL Database or Azure SQL Data Warehouse.)

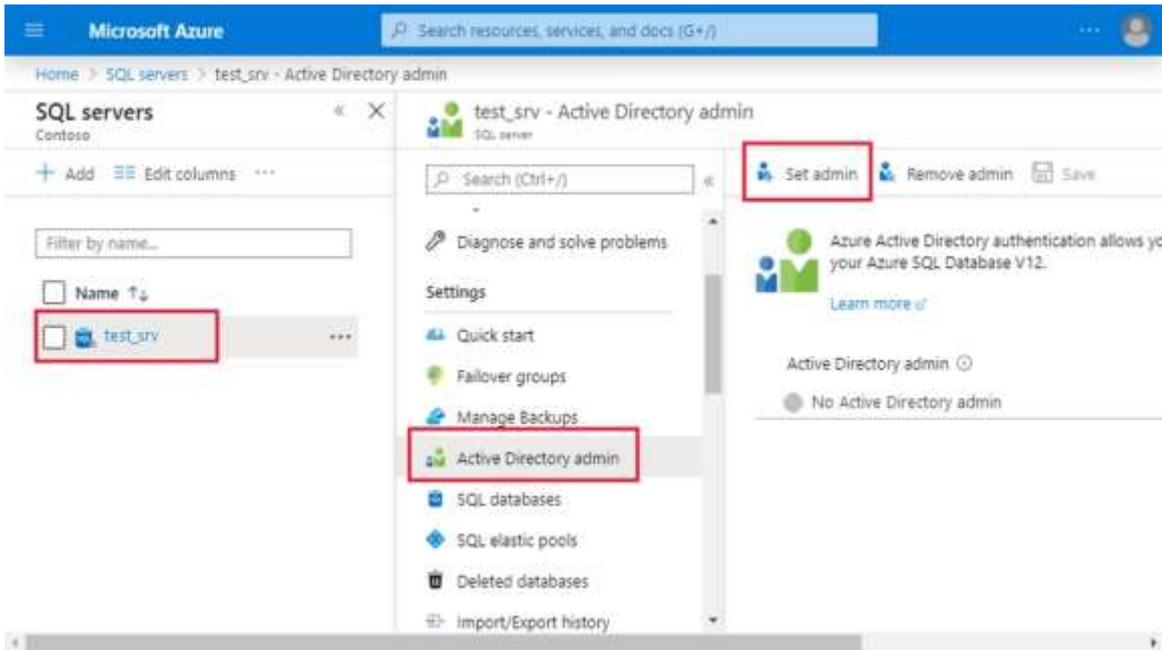


2. Search for and select the SQL server SQL10543936

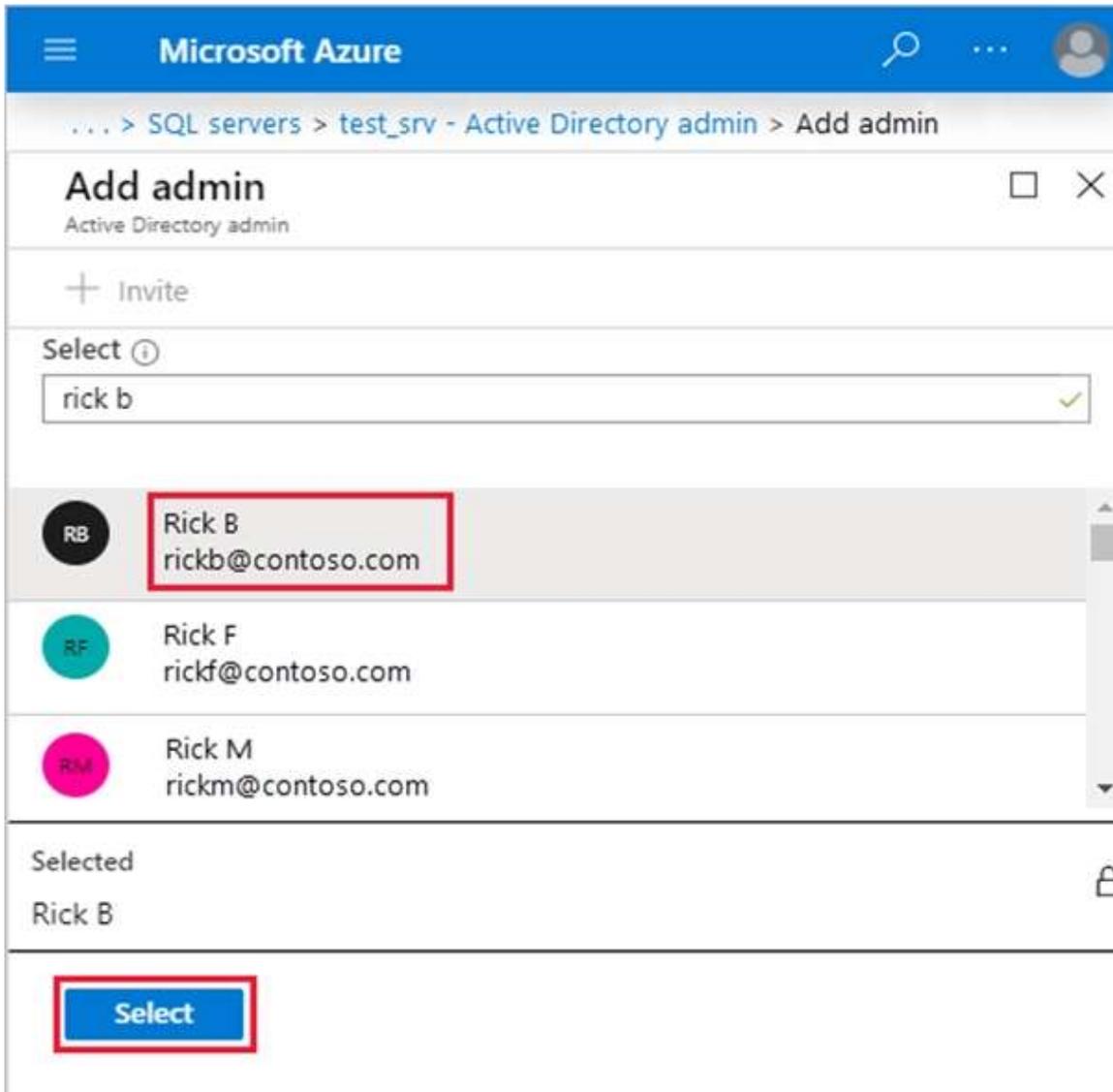


3. In SQL Server page, select Active Directory admin.

4. In the Active Directory admin page, select Set admin.



5. In the Add admin page, search for user User1-10543936@ExamUsers.com, select it, and then select Select. (The Active Directory admin page shows all members and groups of your Active Directory. Users or groups that are grayed out cannot be selected because they are not supported as Azure AD administrators.)



Microsoft Azure

... > SQL servers > test\_srv - Active Directory admin > Add admin

### Add admin

Active Directory admin

+ Invite

Select ⓘ

rick b ✓

	<b>Rick B</b> rickb@contoso.com
	Rick F rickf@contoso.com
	Rick M rickm@contoso.com

Selected

Rick B

Select

6. At the top of the Active Directory admin page, select SAVE.

---

 **Set admin**  Remove admin  Save

---

 Azure Active Directory authentication allows you to connect to Azure SQL Database V12.

[Learn more](#) 

Active Directory admin 

 rickb@contoso.com

---

Reference:

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-aad-authentication-configure?>

**QUESTION 158**

**SIMULATION**

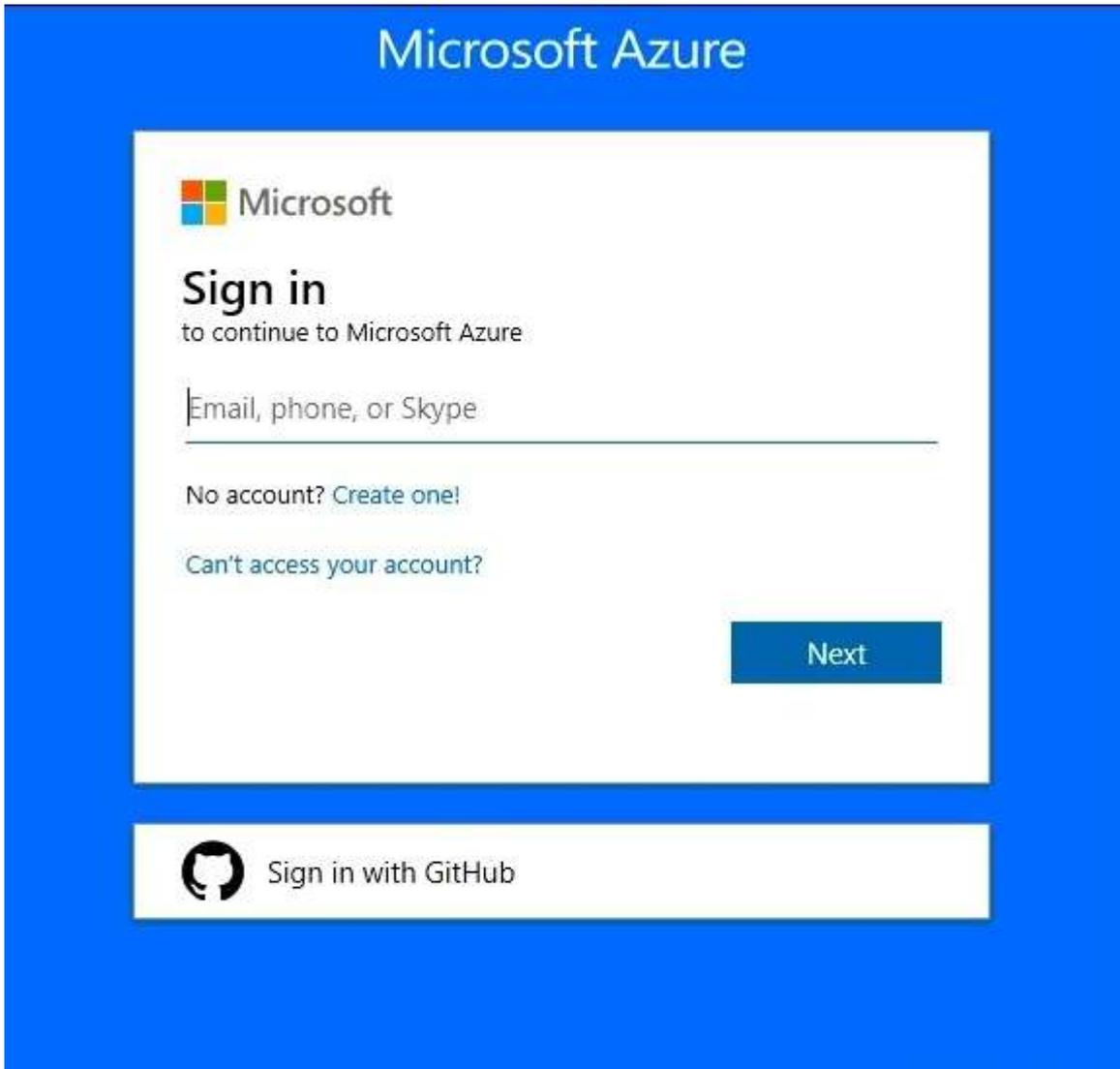
Use the following login credentials as needed:

Azure Username: xxxxx

Azure Password: xxxxx

The following information is for technical support purposes only:

Lab Instance: 10543936



You need to ensure that only the resources on a virtual network named VNET1 can access an Azure Storage account named storage10543936.

**To complete this task, sign in to the Azure portal.**

**Answer:**

You can use Private Endpoints for your Azure Storage accounts to allow clients on a virtual network (VNet) to securely access data over a Private Link.

Create your Private Endpoint

1. On the upper-left side of the screen in the Azure portal, Storage > Storage account, and select your storage account storage10543936

2. Select Networking.

3. Select Add Private Endpoint.

4. In Create Private Endpoint, enter or select this information:

Virtual network: Select VNET1 from the resource group.

5. Select OK.

6. Select Review + create. You're taken to the Review + create page where Azure validates your configuration.

Reference:

<https://docs.microsoft.com/en-us/azure/private-link/create-private-endpoint-storage-portal>

#### **QUESTION 159**

SIMULATION

Use the following login credentials as needed:

Azure Username: xxxxx

Azure Password: xxxxx

The following information is for technical support purposes only:  
Lab Instance: 10543936



You need to replicate db1 to a new Azure SQL server named db1-copy10543936 in the US West region.

**To complete this task, sign in to the Azure portal.**

**Answer:**

1. In the Azure portal, browse to the database db1-copy10543936 that you want to set up for geo-replication.
2. On the SQL database page, select geo-replication, and then select the region to create the secondary database: US West region

WideWorldImporters - Geo-Replication  
SQL database  
Select a region on the map or from the Target Regions list to create a secondary database.

Search (Ctrl+/)

- Overview
- Activity log
- Tags
- Diagnose and solve problems

SETTINGS

- Quick start
- Pricing tier (scale DTUs)
- Geo-Replication**
- Auditing & Threat detection
- Dynamic data masking
- Transparent data encryption
- Properties
- Locks
- Automation script

SERVER/DATABASE STATUS

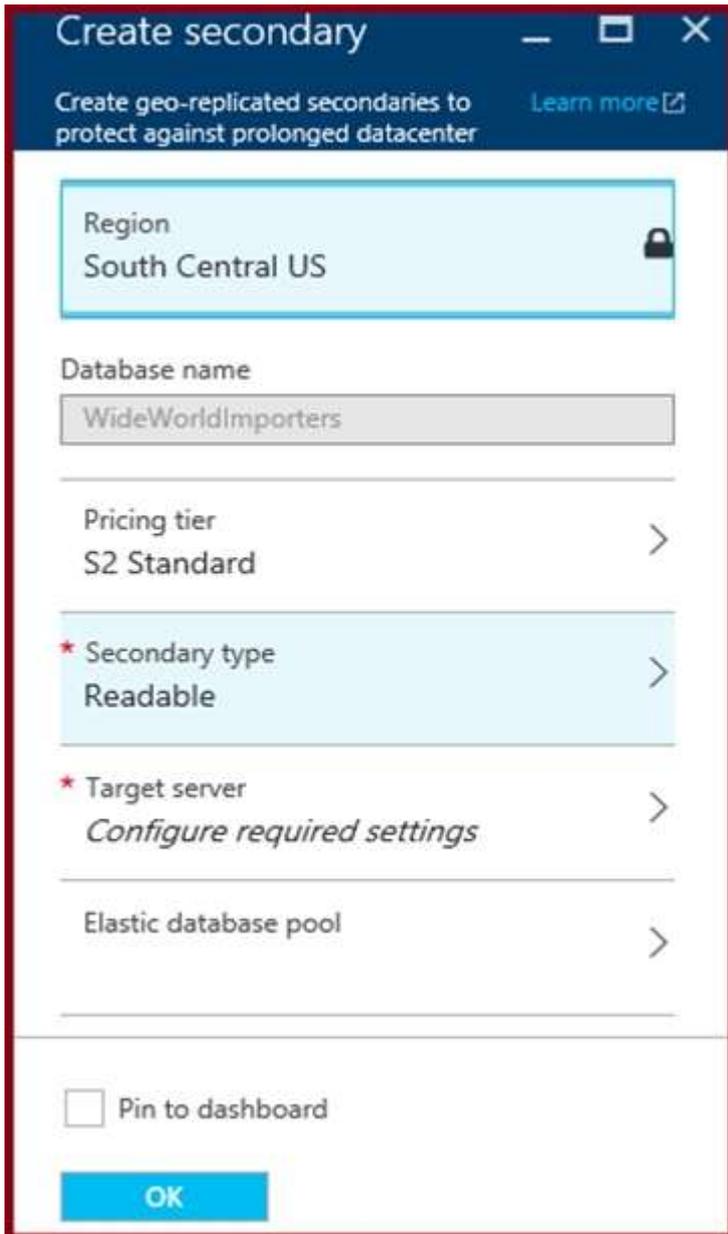
PRIMARY

North Central US	sqldbteam/WideWorldImporte...	Online
------------------	-------------------------------	--------

SECONDARIES

*Geo-Replication is not configured*

3. Select or configure the server and pricing tier for the secondary database.



**Create secondary** — □ ×

Create geo-replicated secondaries to protect against prolonged datacenter [Learn more](#)

Region  
South Central US 

Database name  
WideWorldImporters

Pricing tier  
S2 Standard >

\* Secondary type  
Readable >

\* Target server  
*Configure required settings* >

Elastic database pool >

Pin to dashboard

**OK**

4. Click Create to add the secondary.
5. The secondary database is created and the seeding process begins.



	SERVER/DATABASE	STATUS
<b>PRIMARY</b>		
	North Central US    sqldbteam/WideWorldImporte...	Online
<b>SECONDARIES</b>		
	South Central US    server101801/WideWorldImp...	Initializing...

6. When the seeding process is complete, the secondary database displays its status.

	SERVER/DATABASE	STATUS
<b>PRIMARY</b>		
	North Central US    sqldbteam/WideWorldImporte...	Online
<b>SECONDARIES</b>		
	South Central US    server101801/WideWorldImp...	<b>Readable</b> ...

Reference:  
<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-active-geo-replication-portal>

**QUESTION 160**

**SIMULATION**

Use the following login credentials as needed:

Azure Username: xxxxx

Azure Password: xxxxx

The following information is for technical support purposes only:  
Lab Instance: 10543936



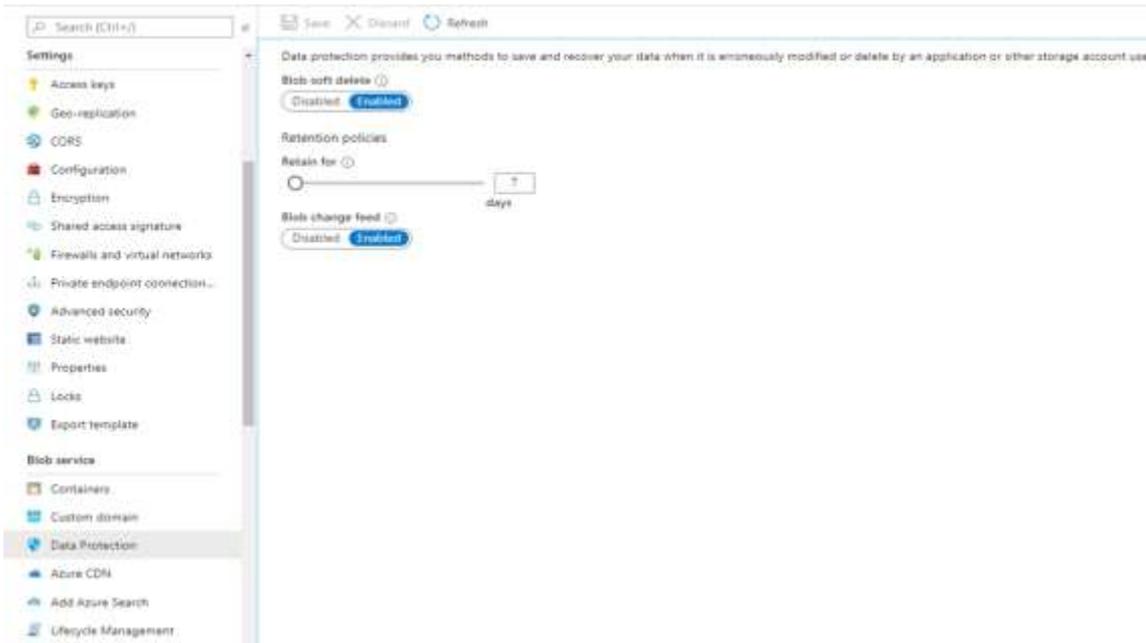
You need to ensure that you can recover any blob data from an Azure Storage account named storage10543936 up to 10 days after the data is deleted.

**To complete this task, sign in to the Azure portal.**

**Answer:**

Enable soft delete for blobs on your storage account by using Azure portal:

1. In the Azure portal, select your storage account.
2. Navigate to the Data Protection option under Blob Service.
3. Click Enabled under Blob soft delete



4. Enter the number of days you want to retain for under Retention policies. Here enter 10.

5. Choose the Save button to confirm your Data Protection settings

Note: Azure Storage now offers soft delete for blob objects so that you can more easily recover your data when it is erroneously modified or deleted by an application or other storage account user. Currently you can retain soft deleted data for between 1 and 365 days.

Reference:

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-soft-delete>

#### QUESTION 161

**Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**

**After you answer a question in this scenario, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.**

You have a container named Sales in an Azure Cosmos DB database. Sales has 120 GB of data. Each entry in Sales has the following structure.

```
{
  OrderId: number,
  OrderDetailId: number,
  ProductName: string,
  other information that might vary...
}
```

The partition key is set to the OrderId attribute.

Users report that when they perform queries that retrieve data by ProductName, the queries take longer than expected to complete.

You need to reduce the amount of time it takes to execute the problematic queries.

Solution: You increase the Request Units (RUs) for the database.

Does this meet the goal?

- A. Yes
- B. No

**Answer: A**

**Explanation:**

[DP-200 Exam Dumps](#) [DP-200 Exam Questions](#) [DP-200 PDF Dumps](#) [DP-200 VCE Dumps](#)

<https://www.braindump2go.com/dp-200.html>

To scale the provisioned throughput for your application, you can increase or decrease the number of RUs at any time. Note: The cost of all database operations is normalized by Azure Cosmos DB and is expressed by Request Units (or RUs, for short). You can think of RUs per second as the currency for throughput. RUs per second is a rate-based currency. It abstracts the system resources such as CPU, IOPS, and memory that are required to perform the database operations supported by Azure Cosmos DB.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/request-units>

#### **QUESTION 162**

You are monitoring an Azure Stream Analytics job.

You discover that the Backlogged Input Events metric is increasing slowly and is consistently non-zero.

You need to ensure that the job can handle all the events.

What should you do?

- A. Change the compatibility level of the Stream Analytics job.
- B. Increase the number of streaming units (SUs).
- C. Create an additional output stream for the existing input stream.
- D. Remove any named consumer groups from the connection and use \$default.

**Answer: B**

**Explanation:**

Backlogged Input Events: Number of input events that are backlogged. A non-zero value for this metric implies that your job isn't able to keep up with the number of incoming events. If this value is slowly increasing or consistently non-zero, you should scale out your job. You should increase the Streaming Units.

Note: Streaming Units (SUs) represents the computing resources that are allocated to execute a Stream Analytics job. The higher the number of SUs, the more CPU and memory resources are allocated for your job.

Reference:

<https://docs.microsoft.com/bs-cyrl-ba/azure/stream-analytics/stream-analytics-monitoring>

#### **QUESTION 163**

**SIMULATION**

Use the following login credentials as needed:

Azure Username: xxxxx

Azure Password: xxxxx

The following information is for technical support purposes only:

Lab Instance: 10543936



Your company's compliance policy states that administrators must be able to review a list of the database object changes that occurred in an Azure SQL database named db2 during the last 100 days. You need to modify your Azure environment to meet the compliance policy requirements.

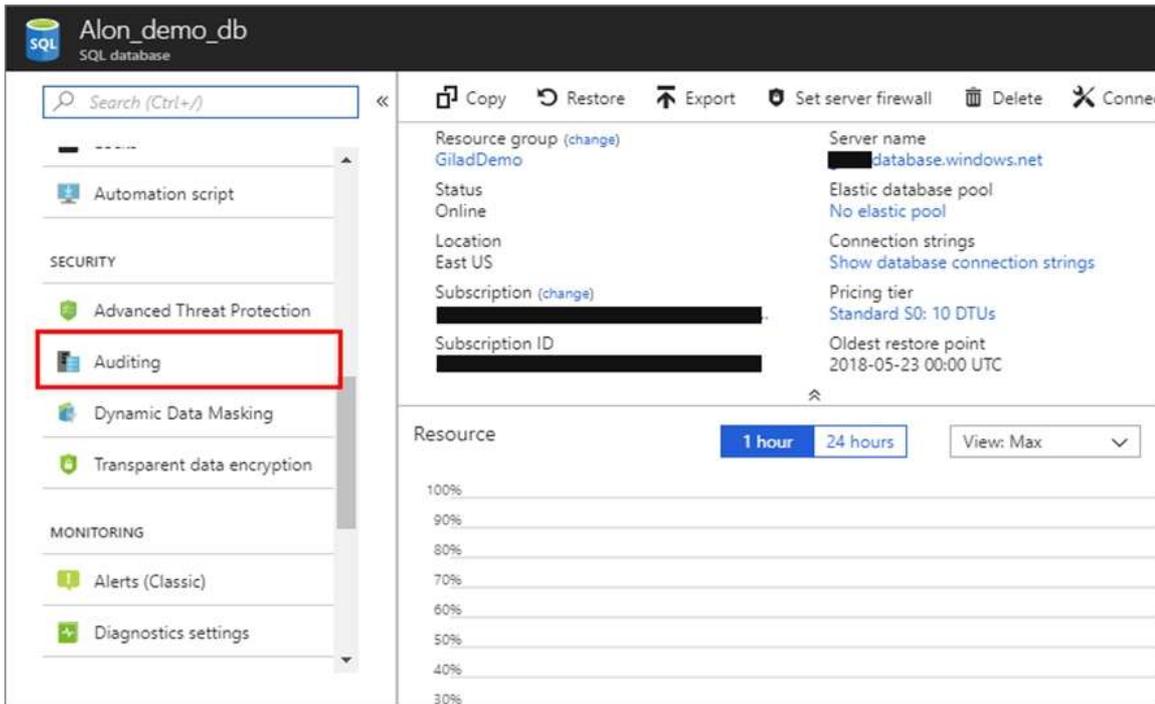
**To complete this task, sign in to the Azure portal.**

**Answer:**

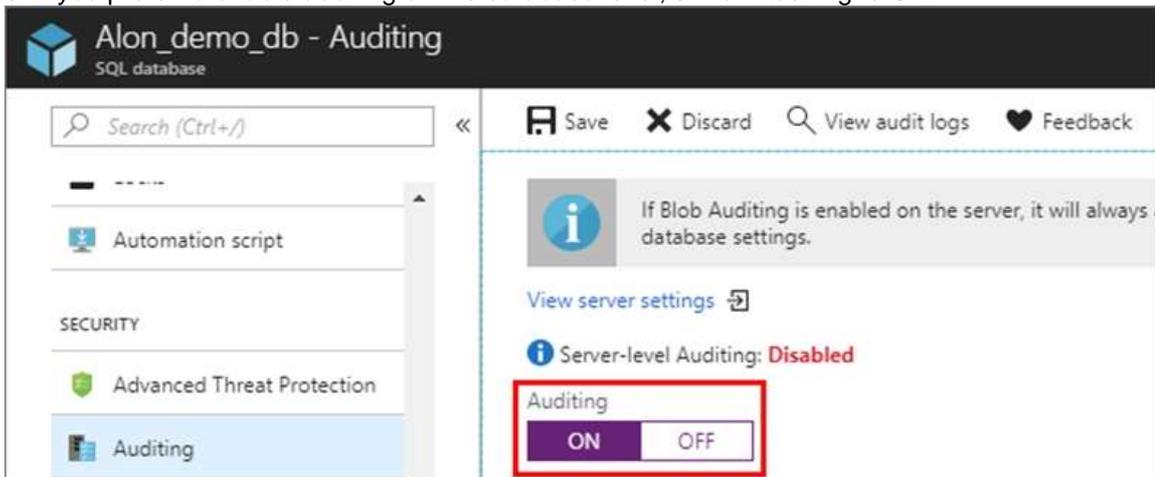
Set up auditing for your database

The following section describes the configuration of auditing using the Azure portal.

1. Go to the Azure portal.
2. Navigate to Auditing under the Security heading in your SQL database db2/server pane



3. If you prefer to enable auditing on the database level, switch Auditing to ON.



Note: By default the audit database data retention period is set to 100 days.

Reference:

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-auditing>