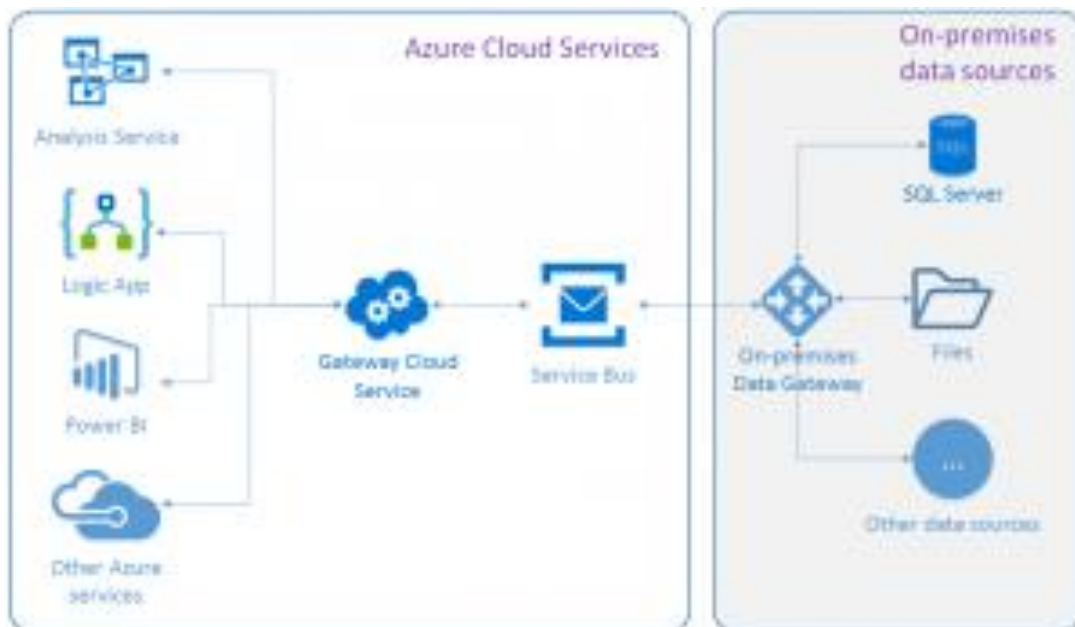


Azure Databases Syncing with LogicApps and PowerApps

Problem Statement :

To enable users in your organization to securely access on-premises data, you'll need to set up an on-premises data gateway. This gateway serves as a secure bridge between your cloud services, including Power BI, Power Automate, Power Apps, Logic Apps, and Azure Analysis Services, and your on-premises data sources. It ensures fast and encrypted communication, allowing data to flow seamlessly between the cloud and your on-premises resources.

Solution Architecture :



Steps Followed:

- **Cloud Service Initialization:**
The process starts with the cloud service generating a query and encrypting the access credentials for the on-premises data source. These are then placed in a queue for the gateway to handle.
- **Query Processing and Routing:**
The gateway cloud service examines the query and forwards the request to Azure Service Bus for further processing.
- **Request Handling by the Gateway:**
Azure Service Bus transmits the pending queries to the on-premises gateway, which receives the query, decrypts the access credentials, and establishes connections with the data sources using these credentials.
- **Data Retrieval and Return:**

Subsequently, the gateway sends the query to the data source for execution. The results are obtained from the data source, relayed back to the gateway, and finally routed to the cloud service, where they are utilized as needed.

Prerequisites

You'll need an active Azure account and subscription.

Ensure your Azure account belongs to a single Azure Active Directory (Azure AD) tenant. Use the same account for both local gateway tasks.

During gateway setup, sign in with your Azure account to link it. Use this linked account consistently when creating an Azure gateway resource.

Use a business/professional email id for login

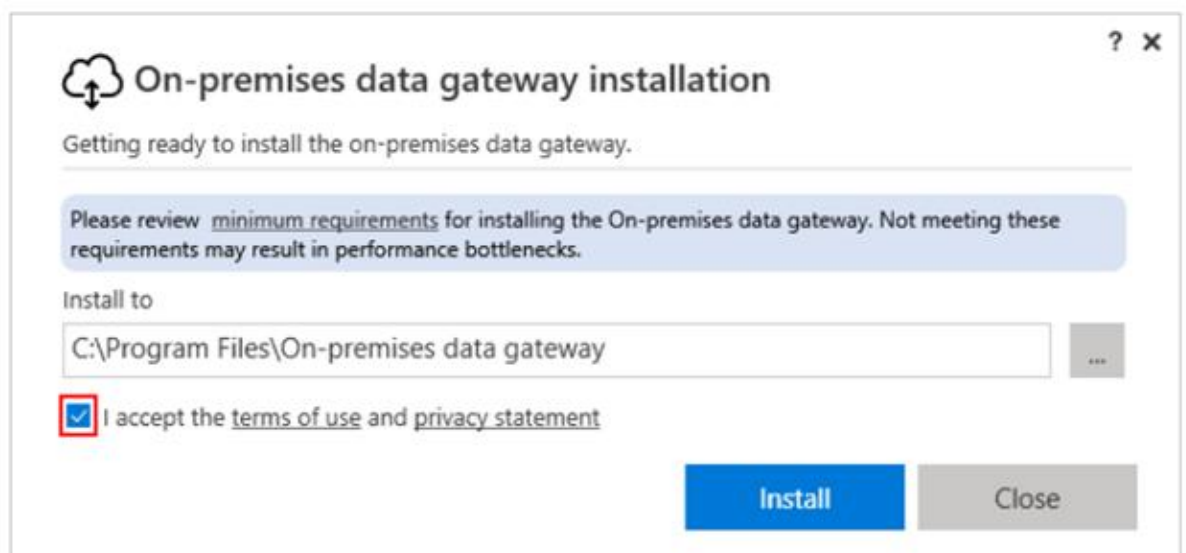
Software & Hardware Requirements:

Minimum: .NET Framework 4.6.2, 64-bit Windows 8 r newer.

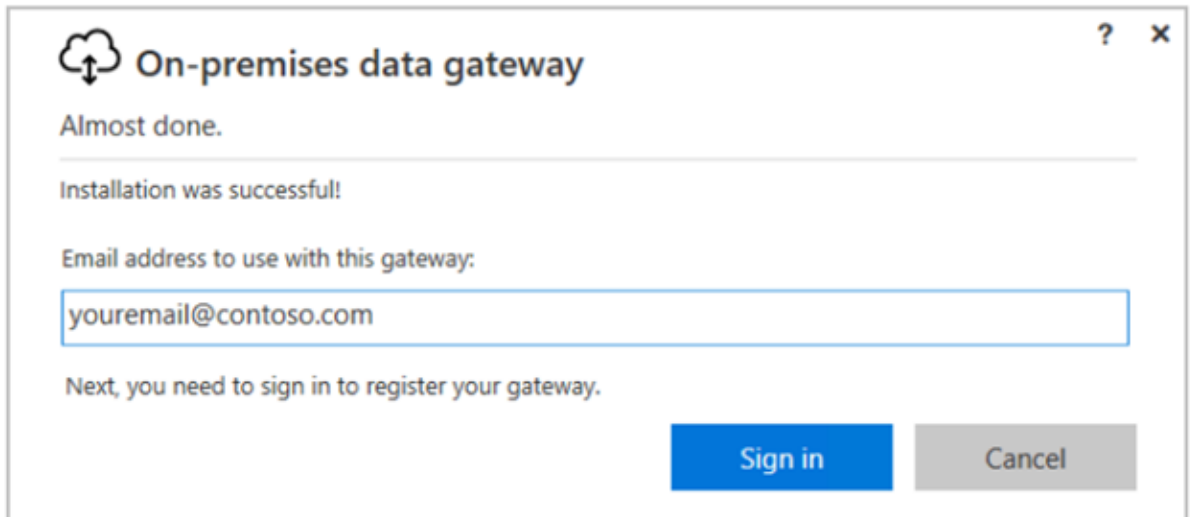
Recommended: 8-core CPU, 8 GB RAM, 64-bit Windows Server 2016 or newer, SSD for improved performance.

Install data gateway

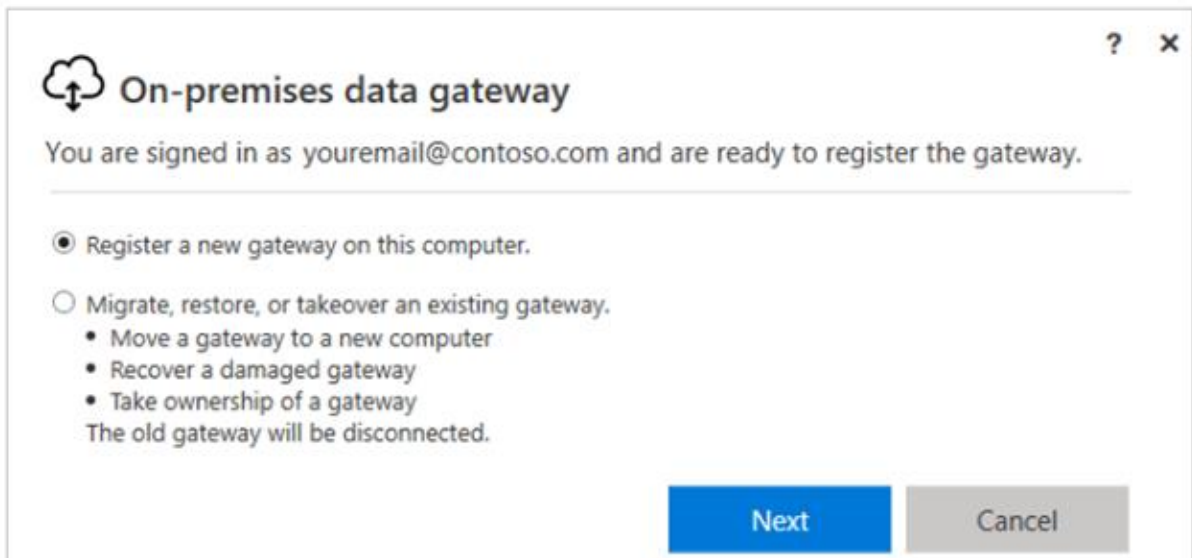
1. Download and run the gateway installer on a local computer.
2. Review the minimum requirements, keep the default installation path, accept the terms of use, and then select **Install**.



3. After the gateway successfully installs, provide the email address for your Azure account, and then select **Sign in**, for example:



4. Select **Register a new gateway on this computer>Next**. This step registers your gateway installation with the gateway cloud service.



5. Provide this information for your gateway installation:
 - A gateway name that's unique across your Azure AD tenant
 - The recovery key, which must have at least eight characters, that you want to use
 - Confirmation for your recovery key
6. Check the region for the gateway cloud service and Azure Service Bus that's used by your gateway installation. By default, this region is the same location as the Azure AD tenant for your Azure account.

Confirm recovery key

.....

[Learn more about gateway clusters](#)

We'll use this region to connect the gateway to cloud services: West US [Change Region](#)

<< Back Configure

7. To accept the default region, select **Configure**. However, if the default region isn't the one that's closest to you, you can change the region.

Why change the region for your gateway installation?

- For example, to reduce latency, you might change your gateway's region to the same region as your logic app. Or, you might select the region closest to your on-premises data source. Your gateway resource in Azure and your logic app can have different locations.

On-premises data gateway ? x

You are signed in as youremail@contoso.com and are ready to register the gateway.

Determine where your new on-premises data gateway will be deployed.

Select Region: West US (Recommended) v

Individual Service Region:

- Logic Apps, Azure Analytics**
West US
- PowerApps, Microsoft Dynamics 365**
Default environment
- Power BI**
Default environment

West US (Recommended)

Australia Southeast

Brazil South

Canada Central

North Europe

West India

Japan East

South Central US

Southeast Asia

East Asia

UK South

East US

East US 2

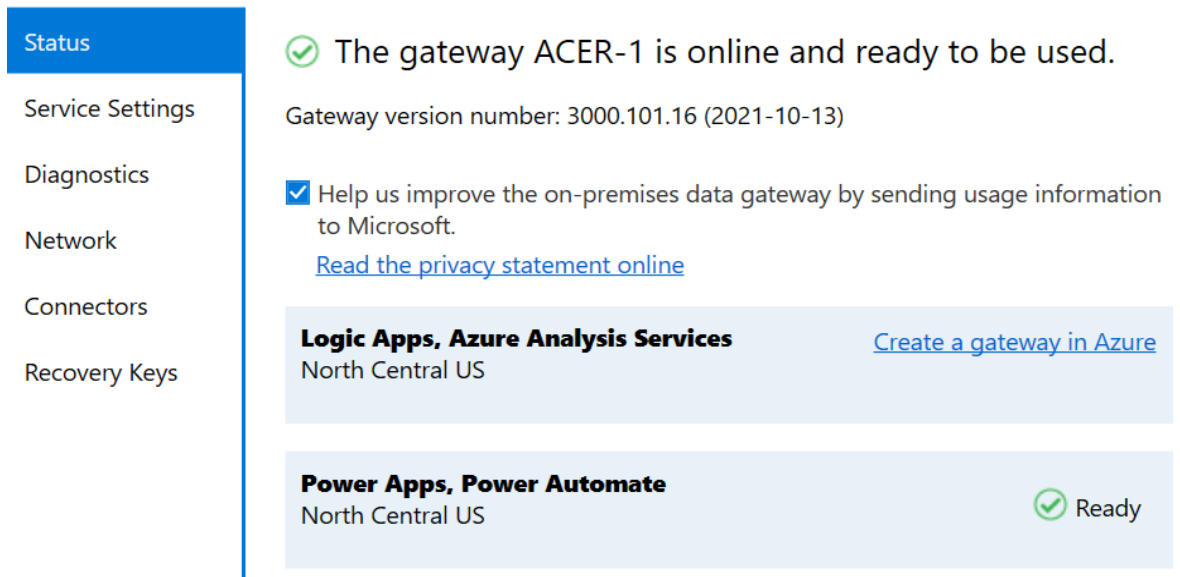
North Central US

West Europe

Done

8. Review the information in the final confirmation window. This example uses the same account for Logic Apps, Power BI, Power Apps, and Power Automate, so the gateway is available for all these services. When you're ready, select **Close**.

On-premises data gateway



Status

Service Settings

Diagnostics

Network

Connectors

Recovery Keys

✔ The gateway ACER-1 is online and ready to be used.

Gateway version number: 3000.101.16 (2021-10-13)

Help us improve the on-premises data gateway by sending usage information to Microsoft.

[Read the privacy statement online](#)

Logic Apps, Azure Analysis Services [Create a gateway in Azure](#)
North Central US

Power Apps, Power Automate ✔ Ready
North Central US

Azure On-Premises Resource Creation

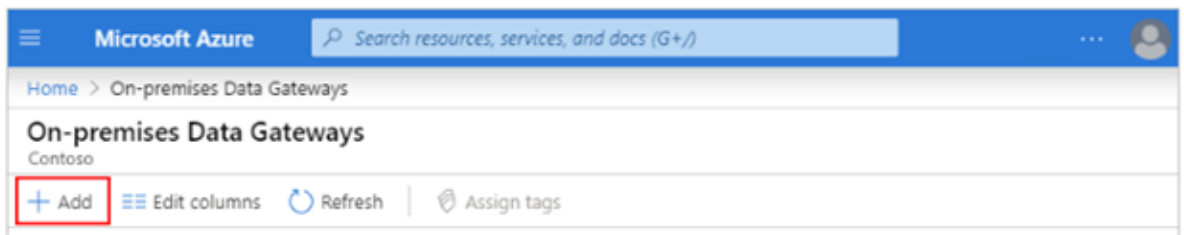
Prerequisites

- You have already installed the on-premises data gateway on a local computer.
- You're using the same Azure account and subscription that was used when installing that data gateway. This Azure account must belong to a single Azure Active Directory (Azure AD) tenant or directory.

Create Azure gateway resource

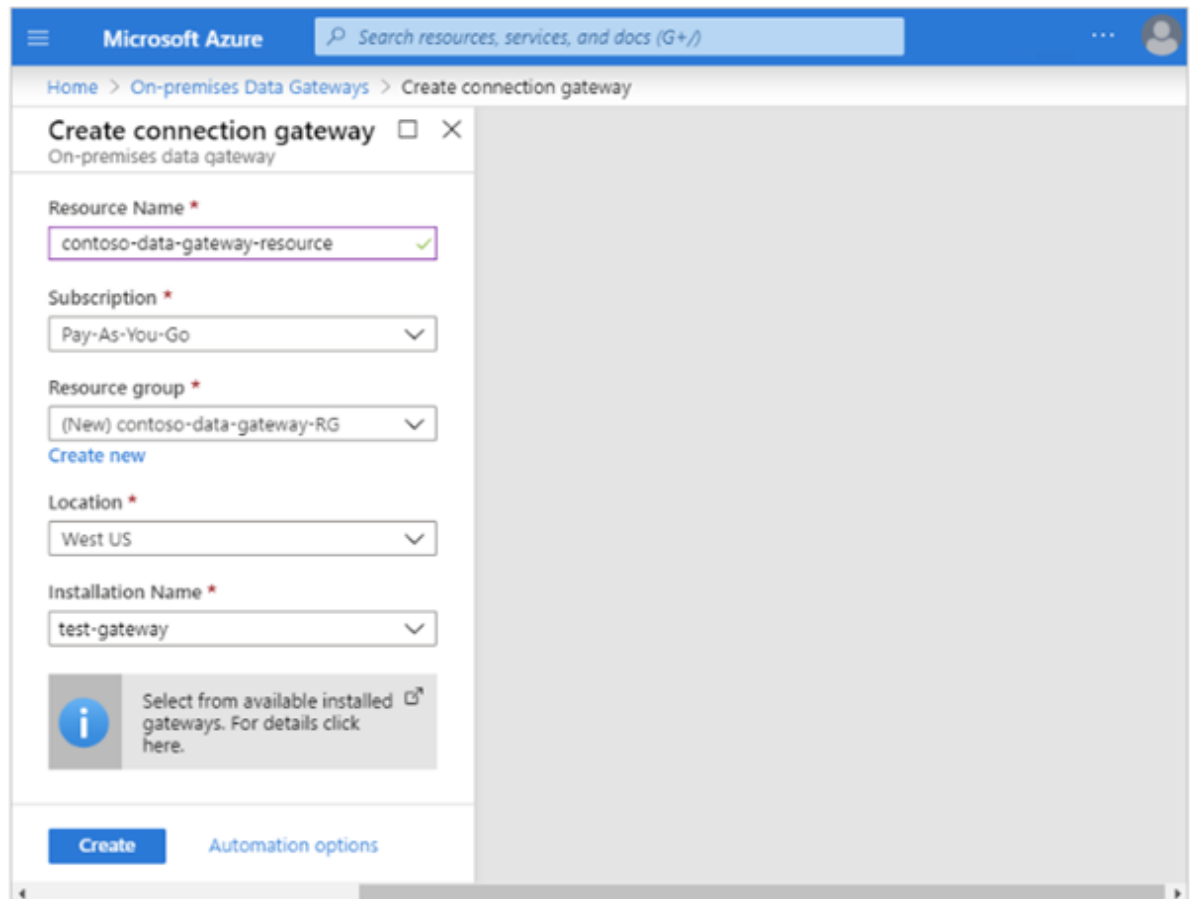
1. Sign in to the Azure portal with the same Azure account that was used to install the gateway.
2. In the Azure portal search box, enter "on-premises data gateway", and select **On-premises Data Gateways**.

Under **On-premises Data Gateways**, select **Add**.



4. Under **Create connection gateway**, provide this information for your gateway resource. When you're done, select **Create**.

Here is an example that shows a gateway installation that's in the same region as your gateway resource and is linked to the same Azure account:



Connecting using LogicApps

For Local Machine Connection

Trigger creation

1. In the Azure portal or Visual Studio, create a blank logic app, which opens Logic App Designer. This example uses the Azure portal.
2. In the search box, enter "Http Request" as your filter. From the triggers list, select the trigger you want.

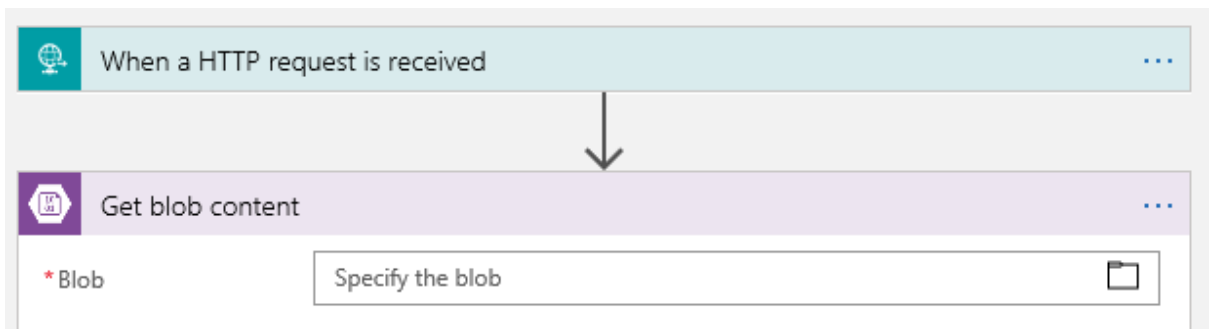
Logic Apps Designer

Save Discard Run Designer Code view Parameters Templates Connectors Help

When a HTTP request is received

Add Action for Blob Storage

When you use a trigger or action that accesses a service for the first time, the Logic Apps Designer prompts you to create a connection to that service. You can then provide the necessary connection information directly from your logic app inside the designer.



Add Action for File System

- To create a new connection with the file system - provide a unique connection name, the root folder to access your file system, authentication for your local machine (username and password)

The screenshot shows the 'File System' connection configuration dialog. It has a title bar with a folder icon and the text 'File System'. Below the title bar, there are several fields:

- * Connection name: MyFileSystemConnection-Managed
- * Root folder: C:\fabrikam-fileshare
- Authentication Type: Windows (dropdown menu)
- * Username: (empty text box)
- * Password: (password masked with dots)
- gateway: Subscription (dropdown menu)
- Subscription: Pay-As-You-Go (dropdown menu)
- Connection Gateway: (empty dropdown menu)

At the bottom of the dialog, there is a blue 'Create' button.

- Employ the 'Create File' feature in the File System Connector to produce a file locally, specifying the folder path, file name, and content.
- Afterward, save your logic app and initiate a test by uploading a file to Blob Storage.

BUSINESS BENEFITS:

Seamless Data Integration: Azure Logic Apps and the File System connector streamline data integration between cloud and on-premises resources.

Enhanced Productivity: Task automation, including file management, improves productivity and allows focus on strategic activities.

Data Security: On-premises data gateway ensures secure, encrypted data transfer, addressing security and compliance concerns.

Scalability and Flexibility: Logic Apps offer scalability for adapting workflows to evolving business needs.

CHALLENGES FACED :

Complex Gateway Configuration: Setting up and configuring the gateway can be technically challenging.

Integration Testing: Thorough testing is required to ensure seamless cloud-on-premises integration, potentially causing project delays.

Resource and Maintenance Overhead: Gateway maintenance and updates add IT resource and maintenance workload.

Latency Considerations: Data transfer latency may impact real-time processing, depending on network infrastructure.