

# ADF v2.0 Pull data from on-prem database and load it into Azure SQL database

Monday, January 11, 2021 7:00 PM

Download and install [Download Microsoft Integration Runtime from Official Microsoft Download Center](#) on your local machine

Go to your Azure portal and create an instance of Azure Data Factory v2.0

In the Overview details screen, click on Author & Monitor



Click on Manage menu item on the left side of the screen



Click on Integration Runtimes

Connections

Linked services

Integration runtimes

Click on + New link

## Integration runtimes

The integration runtime (IR) is the compute ir

+ New Refresh

Click on Azure, Self-Hosted option



**Azure, Self-Hosted**

Perform data flows, data movement and dispatch activities to external compute.

Click on Continue

## Integration runtime setup

### Network environment:

Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:



**Azure**

Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.



**Self-Hosted**

Use this for running activities in an on-premise / private network

[View more](#) ▾

Click on Self-Hosted and click on Continue

## Integration runtime setup

Private network support is realized by installing integration runtime to machines in the same on-premises network/VNET as the resource the integration runtime is connecting to. Follow below steps to register and install integration runtime on your self-hosted machines.

Name \* ⓘ

Description

Type

Enter a name in the Name field and click on Create

## Integration runtime setup

Settings   Nodes   Auto update   Sharing

Install integration runtime on Windows machine or add further nodes using the Authentication Key.

Name ⓘ

### Option 1: Express setup

[Click here to launch the express setup for this computer](#)

### Option 2: Manual setup

Step 1: [Download and install integration runtime](#)

Step 2: Use this key to register your integration runtime

Name	Authentication key	
Key1	IR@291a3b57-4ca8-4583-841b-3f45727426c8@dc1-adfv2@ServiceEndpo	 
Key2	IR@291a3b57-4ca8-4583-841b-3f45727426c8@dc1-adfv2@ServiceEndpo	 

Coy the Key1 value and save it. You will need it later on.

Download and Install integration runtime (Option 2: Manual Setup)

When Integration Runtime (Self-hosted) installation completes, you will come to the Register Integration Runtime (Self-hosted) dialog box.

## Register Integration Runtime (Self-hosted)

Welcome to Microsoft Integration Runtime Configuration Manager. Before you start, register your Integration Runtime (Self-hosted) node using a valid Authentication Key.



☐ Show Authentication Key

[Learn how to find the Authentication Key](#)

### HTTP Proxy

Current Proxy: No proxy

[Change](#)

Register

Cancel

Paste the Key value you saved earlier into the registration box and click on Register button at the bottom  
Upon successful registration, you will see the screen shown below

Microsoft Integration Runtime Configuration Manager

## New Integration Runtime (Self-hosted) Node

Integration Runtime (Self-hosted) node name: ⓘ

DC-LAPTOP



Below is the list of Integration Runtime (Self-hosted) Nodes:

▲ dc1-integration-runtime1  
DC-LAPTOP : Current New Node

☐ Enable remote access from intranet ⓘ

Finish

Cancel

Click on Finish

Upon completion, click on Launch Configuration Manager

Microsoft Integration Runtime Configuration Manager



## Register Integration Runtime (Self-hosted)

Welcome to Microsoft Integration Runtime Configuration Manager. Before you start, register your Integration Runtime (Self-hosted) node using a valid Authentication Key.



☐ Show Authentication Key

[Learn how to find the Authentication Key](#)

### HTTP Proxy

Current Proxy: No proxy

[Change](#)



Integration Runtime (Self-hosted) node has been registered successfully.

Note: You can associate up to 4 physical nodes with a Self-hosted Integration Runtime. This enables high availability and scalability for the Self-hosted Integration Runtime.

We recommend you setup at least 2 nodes for higher availability. [See Integration Runtime \(Self-hosted\) article for details.](#)

Note: You can associate up to 4 physical nodes with a self-hosted integration runtime. This enables high availability and scalability for the Self-hosted Integration Runtime. We recommend you setup at least 2 nodes for higher availability. [See Integration Runtime \(Self-hosted\) article for details.](#)

Launch Configuration Manager

Close

You will see the screen shown below:

Microsoft Integration Runtime Configuration Manager

Home Settings Diagnostics Update Help

## Self-hosted node is connected to the cloud service

Data Factory: dc1-adv2  
Integration Runtime: dc1-integration-runtime1  
Node: DC-LAPTOP

Stop Service

### Data Source Credential ⓘ

Credential store: On-premises  
Credential status: In sync  
Last backup time: N/A

Generate Backup

Import Backup

Connected to the cloud service (Data Factory V2)

Click on Diagnostics menu item at the top

Microsoft Integration Runtime Configuration Manager

Home Settings Diagnostics Update Help

### Logging ⓘ

View logs

Send logs

### Test Connection ⓘ

For supported driver versions, please refer to 'Prerequisites' section under each [data store](#)

Data source type

☐ Encrypt connection

Server

Database

Authentication mode

User name

Password

Test


## Logging ⓘ

View logs

Send logs

## Test Connection ⓘ






For supported driver versions, please refer to 'Prerequisites' section under each [data store](#)

SqlServer	<input type="checkbox"/> Encrypt connection
DC-LAPTOP	dc1-on-prem-sql-db1
Basic	
dc-admin	●●●●●●●●
<div>Test </div>	

Fill the screen with appropriate values and click on Test. If you see Green checkmark next to Test button, the Integration Runtime setup is complete.

Go back to your Azure portal and go to the Author & Monitor of your ADFv2 instance you created earlier.

Click on the Manage icon on the left --> Linked services --> + New



Connections

Linked services

Integration runtimes

Source control

Git configuration

ARM template

### Linked services

Linked service defines the connection information to a data store or compute. [Learn more](#)

+ New

Annotations : Any

Showing 1 - 2 of 2 items

## New linked service

Data store **Compute**All **Azure** Database File Generic protocol NoSQL Services and apps

Search for SQL Server, select it and click on Continue button at the bottom

## New linked service (SQL Server)

Name \*  
dc1\_src\_sqldb1

Description

Connect via integration runtime \* ⓘ  
dc1-integration-runtime1

Connection string Azure Key Vault

Server name \*  
DC-LAPTOP

Database name \*  
dc1-on-prem-sql-db1

Authentication type  
SQL authentication

User name \*  
dc-admin

Password Azure Key Vault

Password \*

✓ Connection successful

Create Back Test connection Cancel

Fill out all the fields. Please pay special attention to **Connect via integration runtime**. **Make sure you select the Integration Runtime you created earlier.**  
Click in Test Connection. If all goes well, you will see Green checkmark Connection successful.  
Click on Create.

## Linked services

Linked service defines the connection information to a data store or compute. [Learn more](#)

+ New

Click on the + New link on the Linked services screen to create the target linked service

## New linked service

Data store Compute

azure sql Database

All Azure Database Fi

SQL

Azure SQL Database

Search for Azure SQL Database and select it.  
Click on Continue

## New linked service (Azure SQL Database)

**Name \***  
dc1\_target\_db\_ls1

**Description**

**Connect via integration runtime \*** ⓘ  
AutoResolveIntegrationRuntime

**Connection string** **Azure Key Vault**

**Account selection method** ⓘ  
☒ From Azure subscription ☐ Enter manually

**Azure subscription**  
Pay-As-You-Go (dee29fa2-d695-439c-80a5-158ce469882b)

**Server name \***  
dc1-sql-server1

**Database name \***  
dc1-sql-db1

**Authentication type \***  
SQL authentication

**User name \***  
dc-admin

**Password** **Azure Key Vault**

**Password \***  
\*\*\*\*\*

**Create** **Back** **Test connection** **Cancel**

Click on Test Connection. If successful, you will see a Green checkmark next to it. Click on Create

You can create Global variables, should you choose to do so

» Data Factory | Validate all | Publish all | Refresh | Discard all | Data flow debug | ARM template

**Connections**

- Linked services
- Integration runtimes
- Source control
- Git configuration
- ARM template
- Parameterization template
- Author
- Triggers
- Global parameters**

**Global parameters**

Global parameters are constants across a Data Factory that can be consumed by a pipeline in any expression. [Learn more](#)

+ New Edit all

Name	Type	Value
sale_date_month	int	1


Click on Author (pencil) icon  
Right click on Datasets, click on New dataset

## New dataset

In pipeline activities and data flows, refer data within a data store. [Learn more](#)

Select a data store

[All](#) [Azure](#) [Database](#) [File](#)



SQL Server

Select SQL Server and click on Continue

## Set properties

**Name**

**Linked service \***  

dc1\_src\_sqldb1

**Connect via integration runtime \***  

dc1-integration-runtime1

**Table name**  

dbo.sales\_data

☐ Edit

**Import schema**  

☒ From connection/store ☐ None


Click on OK at the bottom of the screen

## New dataset

In pipeline activities and data flows, refer data within a data store. [Learn](#)

Select a data store

[All](#) [Azure](#) [Database](#)



Azure SQL Database

Search for Azure SQL Database, select it and click on Continue at the bottom



## Set properties

Name  
dc1\_target\_database\_ds1

Linked service \*  
target\_sqldb\_ls1

Table name  
dbo.sales\_data

☐ Edit

Import schema  
☒ From connection/store ☐ None

OK Back Cancel

Enter the name, target database linked service and the name of the target table.  
Click on OK

You can create Parameter(s), if required as shown below:

» Data Factory Validate all Publish all 2 Refresh Discard all Data flow debug ARM template

Factory Resources

- Pipelines 1
- Datasets 4
  - dc1\_OnPrem\_Dataset1
  - dc1\_src\_ds1
  - dc1\_target\_database\_ds1
  - dc1\_target\_ds1
- Data flows 0

dc1\_src\_ds1

SQL Server  
dc1\_src\_ds1

Connection Schema Parameters

+ New | Delete

NAME	TYPE	DEFAULT VALUE
sales_date_month	Int	Value

Now it is tie to create a pipeline.

Click on Author (pencil) icon, right mouse click on Pipelines, and click on New Pipeline

» Data Factory Validate all Publish all 2 Refresh Discard all Data flow debug ARM template

Factory Resources

- Pipelines 1
  - Copy On-prem data to Prod
- Datasets

dc1\_src\_ds1

New pipeline  
Pipeline from template

You will see:

Factory Resources

Filter resources by name

Pipelines 2

- Copy On-prem data to Prod
- pipeline1

Datasets 4

dc1 OnPrem Dataset1

Activities

Search activities

Move & transform

- Copy data
- Data flow

Save as template Validate Validate copy runtime Debug Add trigger

Copy data

Copy data1

Give it a more meaningful name

Copy data

Pull On\_prem Data to Cloud

General Source Sink Mapping Settings User properties

Name \* Pull On\_prem Data to Cloud [Learn](#)

Description

Timeout 7:00:00:00

Retry 0

Click in the area below pipeline (Copy data) icon and you will see:

Copy data

Pull On\_prem Data to Cloud

Parameters Variables Output

+ New | Delete

NAME	TYPE	DEFAULT VALUE
sale_date_month	Int	Value

Click on Parameters menu item and click on + New to create a parameter as shown above

General Source Sink Mapping Settings User properties

Source dataset \* dc1\_OnPrem\_Dataset1

Open + New Preview data

Dataset properties

NAME	VALUE	TYPE
sale_date_month	@pipeline().globalParameters.sale_date_month	int

Use query ☒ Table ☐ Query ☐ Stored procedure

Click on Dynamic content

General **Source** Sink Mapping Settings User properties

Source dataset \* dc1\_OnPrem\_Dataset1

Open + New Preview data

Dataset properties ⓘ

NAME	VALUE	TYPE
sale_date_month	@(pipeline().globalParameters.sale_date_month)	int

Use query ☐ Table ☒ Query ☐ Stored procedure

Query

```
Select * from sales_data where
Month(sale_date) =
@{(pipeline().globalParameters.sale_date
_month)}
```

Click on Source Menu item at the bottom half of the screen

Select Source dataset, dynamic value of the parameter field, e.g., sale\_date\_month

Create query in the query box. **Please sure to note the parameter format.**

Configure Sink as shown below:

General **Sink** Mapping Settings User properties

Sink dataset \* dc1\_target\_database\_ds1 Open + New

Stored procedure name Select... Refresh

☐ Edit ⓘ

Table option ☒ None ☐ Auto create table ⓘ

Pre-copy script

Write batch timeout

Write batch size

Max concurrent connections

After the Source and the Sink are setup, Click on **Publish all**.

After all components are published, click on Add trigger, click on Trigger now

dc1\_src\_ds1 dc1\_target\_database\_... dc1\_OnPrem\_Dataset1 dc1\_on\_prem\_to\_cl...

Activities

Search activities

- Move & transform
- Azure Data Explorer
- Azure Function
- Batch Service

Copy data

Pull On\_prem Data to Cloud

Trigger now

New/Edit

That is it. You can verify the results on the Azure SQL Database.