

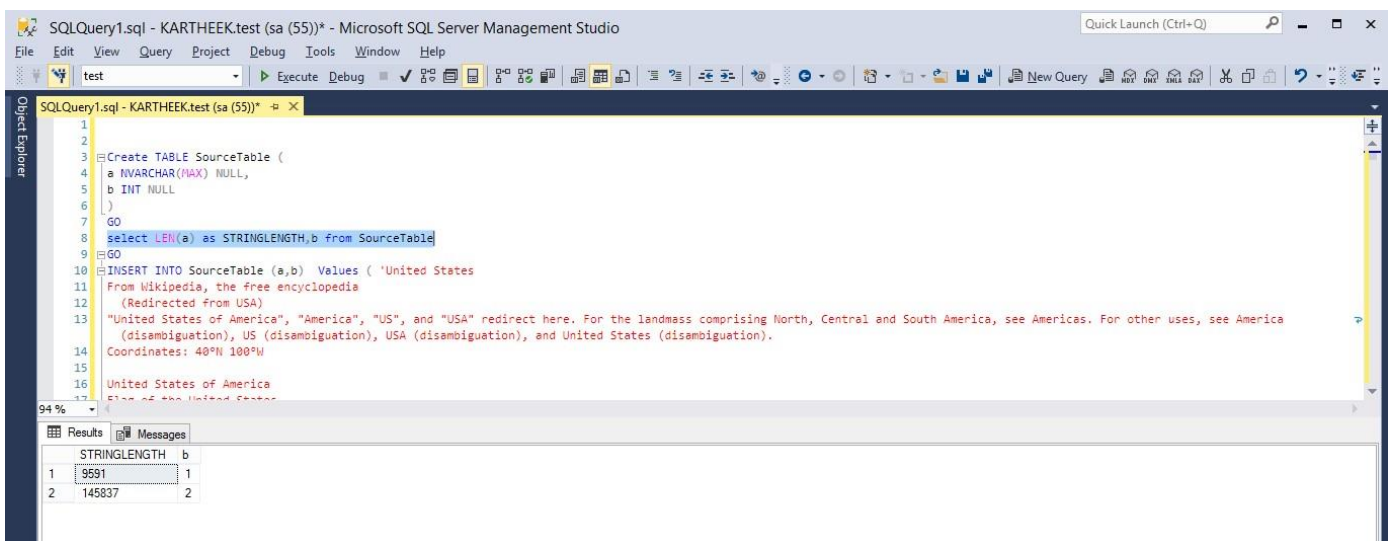
For the task of loading data from **SQL Server On-Premise to Azure SQL Database, please follow the below** steps

To fetch the data

- Access to On-Premise SQL SERVER via SQL Authentication
- Installation of Integration Runtime Gateway
 - This is required for the Azure Data Factory to take data from the On-Premise DB and load into Azure SQL Database

To do this operation, we would need Azure Data Factory V1

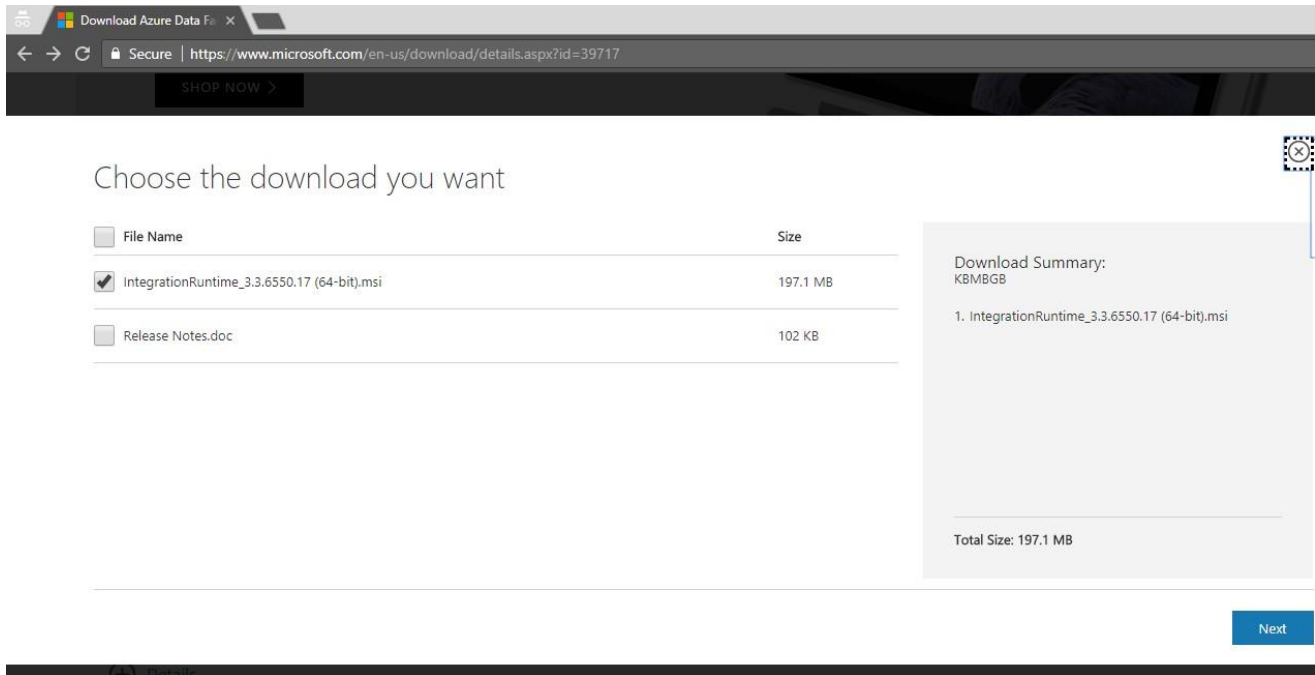
Firstly, I create a sample Table with a sample record to check I can really put the entire information in SQL Azure Database without an intermediate Blob Storage



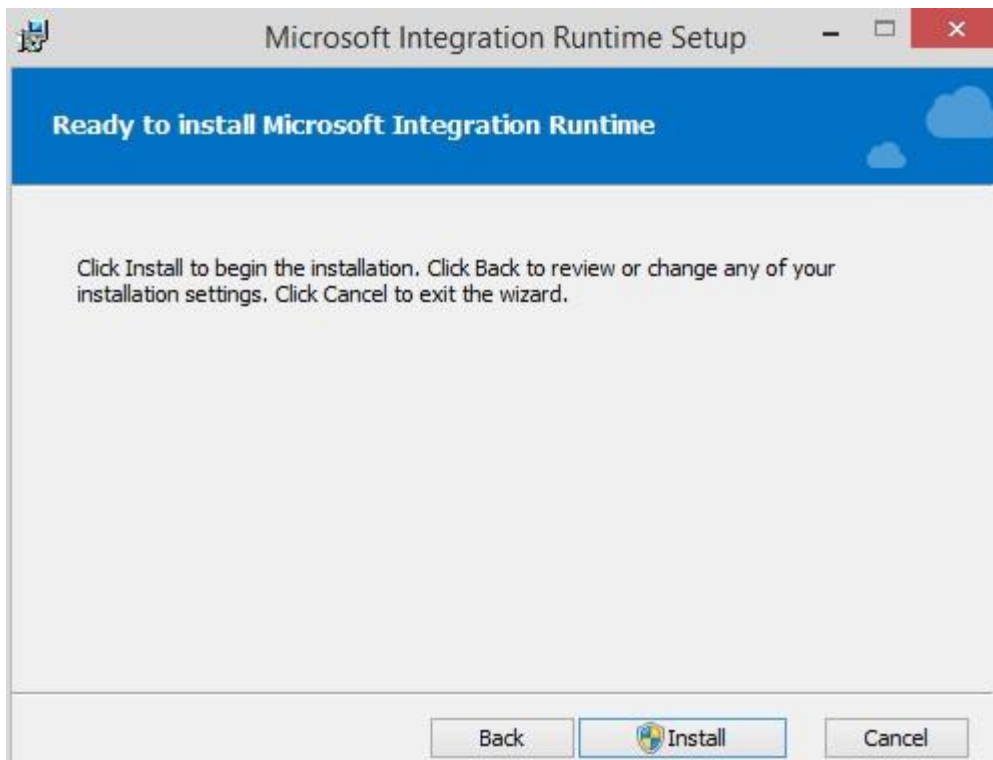
Download the "Azure Data Factory Integration Runtime" using below link

<https://www.microsoft.com/download/details.aspx?id=39717>

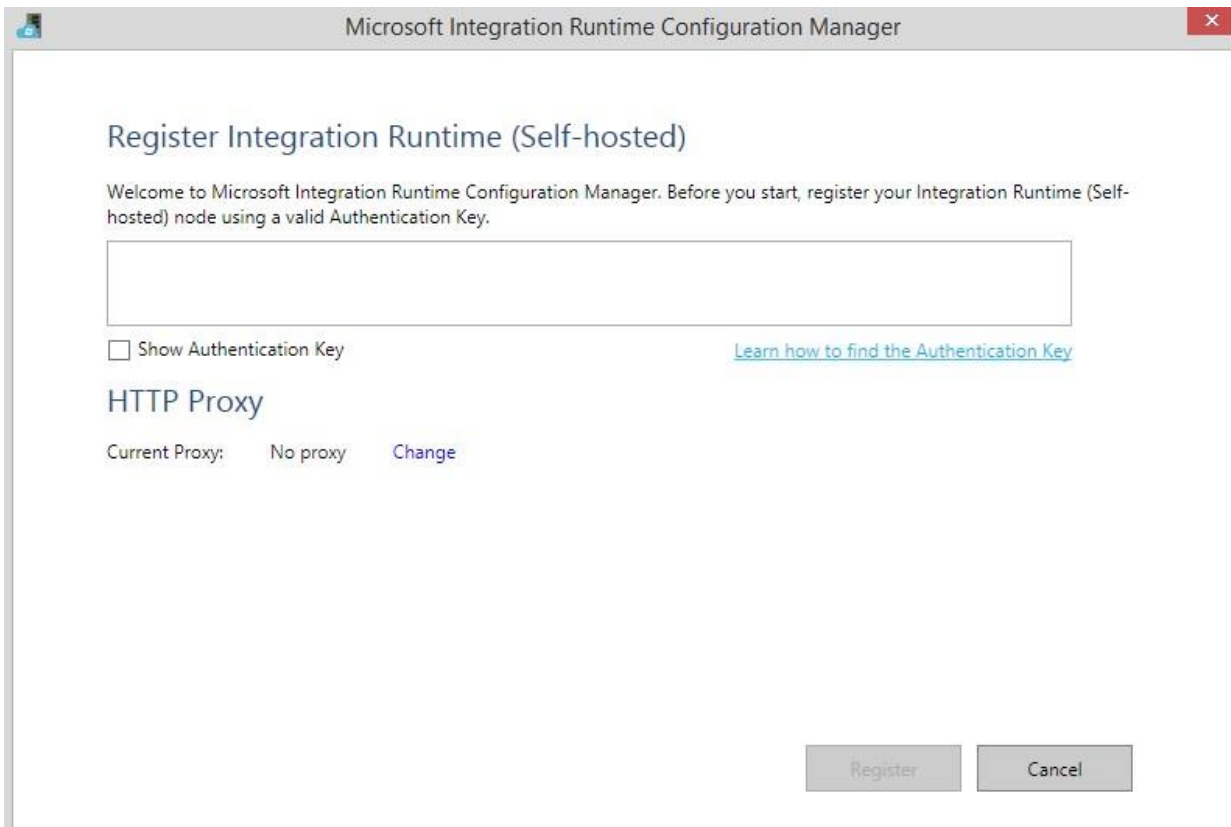
I am using a 64-bit machine.



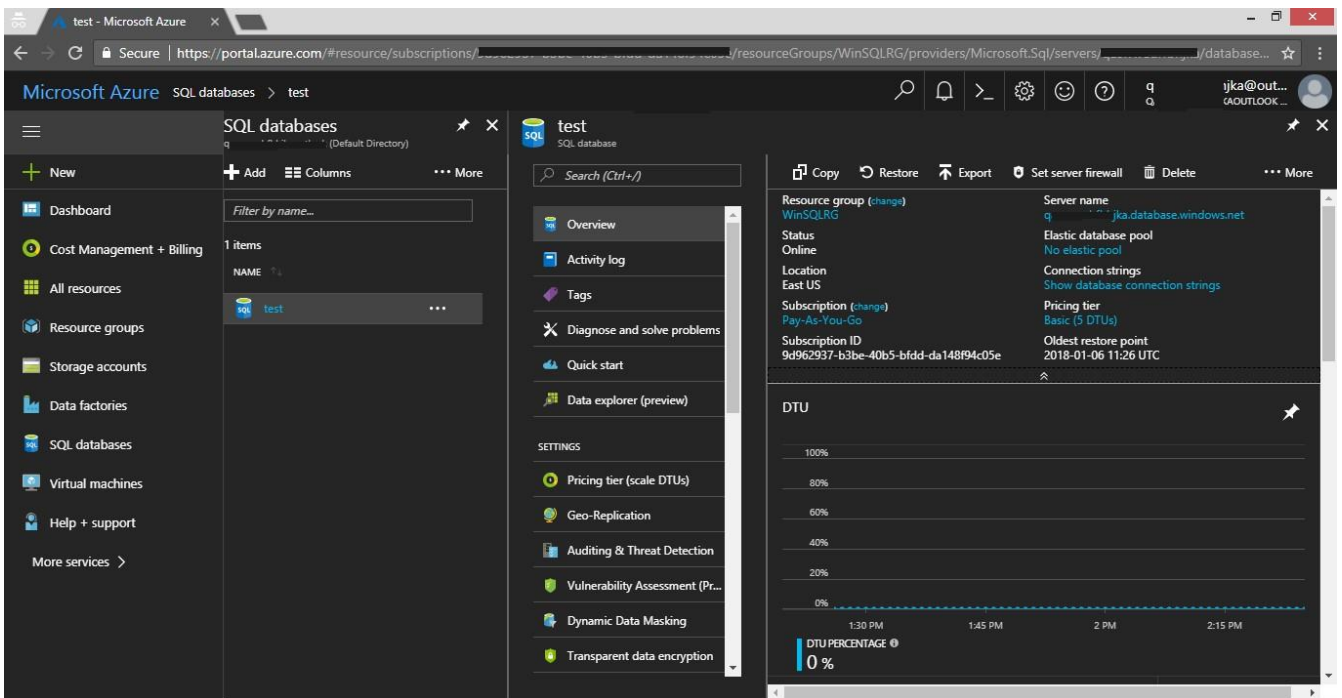
Accept the licensing Terms and Click Install, the Application may ask you to change Power Options of the Machine in case of Sleep Mode being active in the Machine.



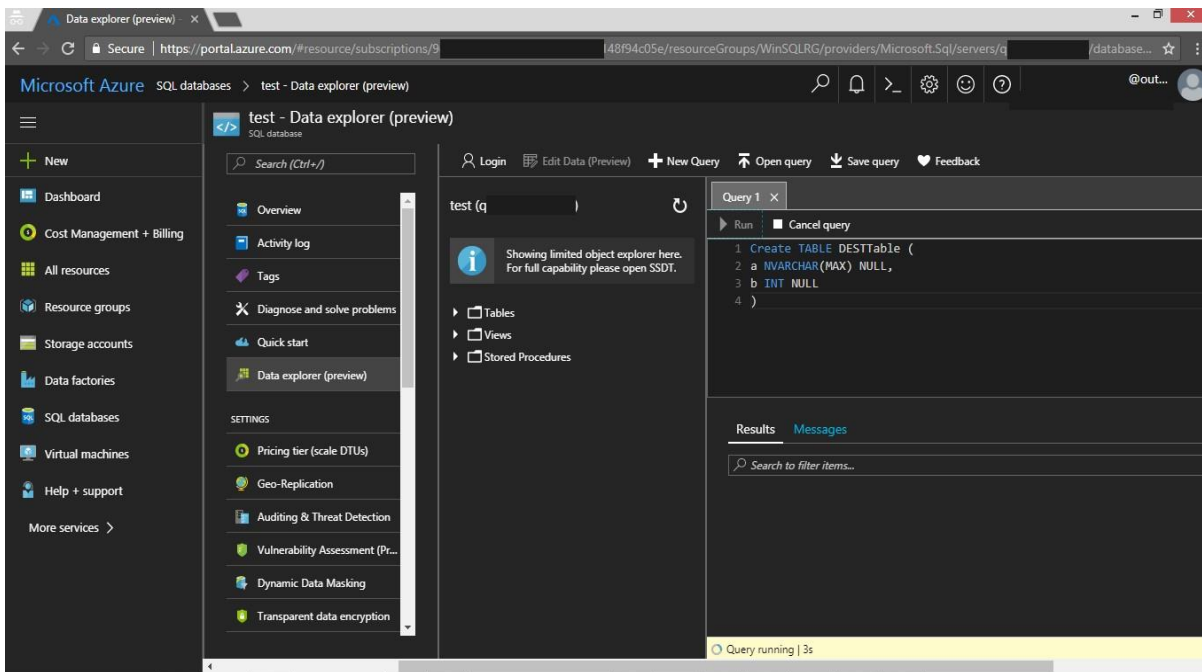
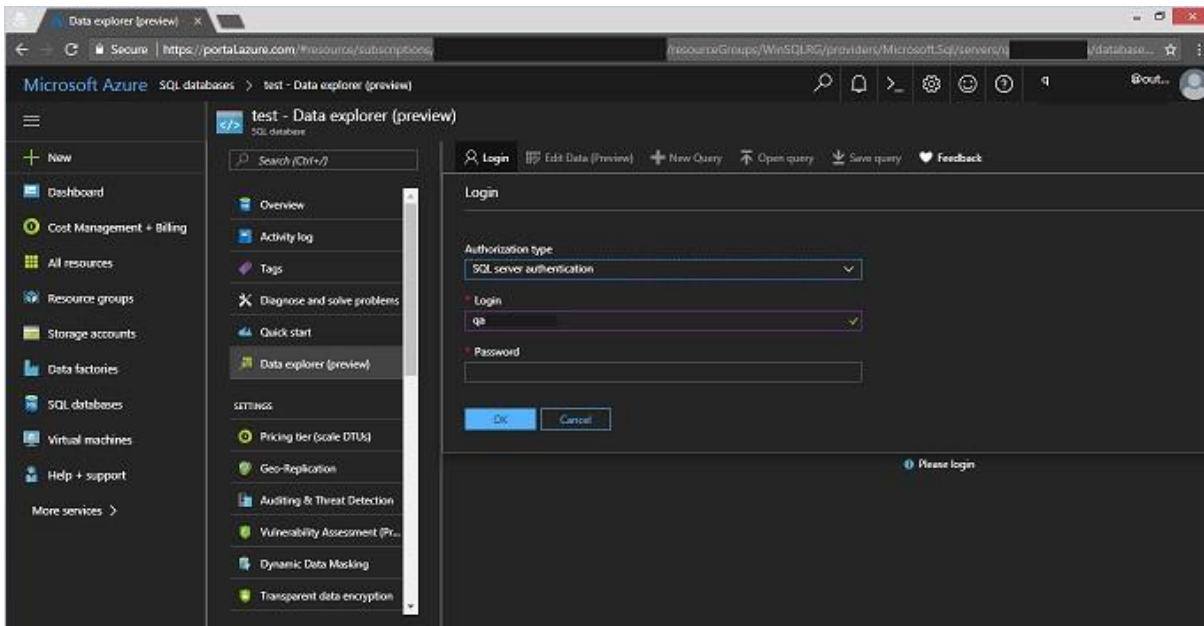
Post installation, the application looks like this.



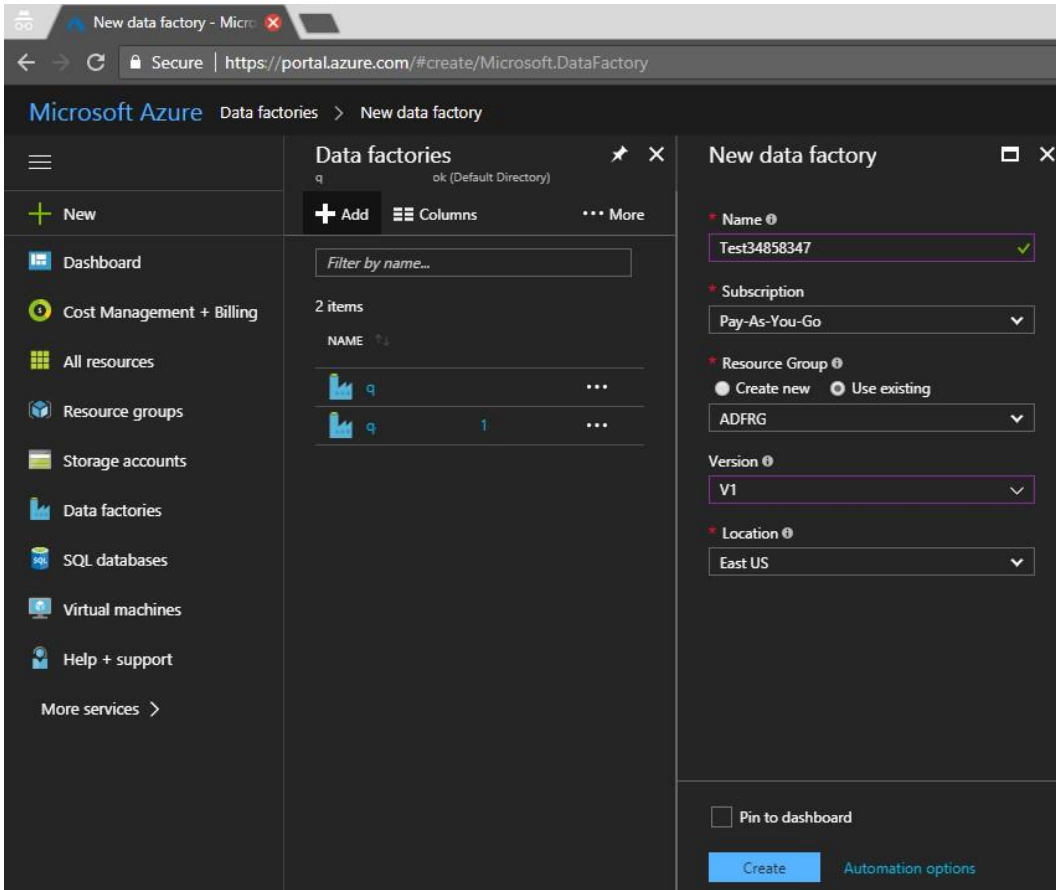
Now Login to the Azure Portal to verify the SQL Database and Azure Data Factory options.



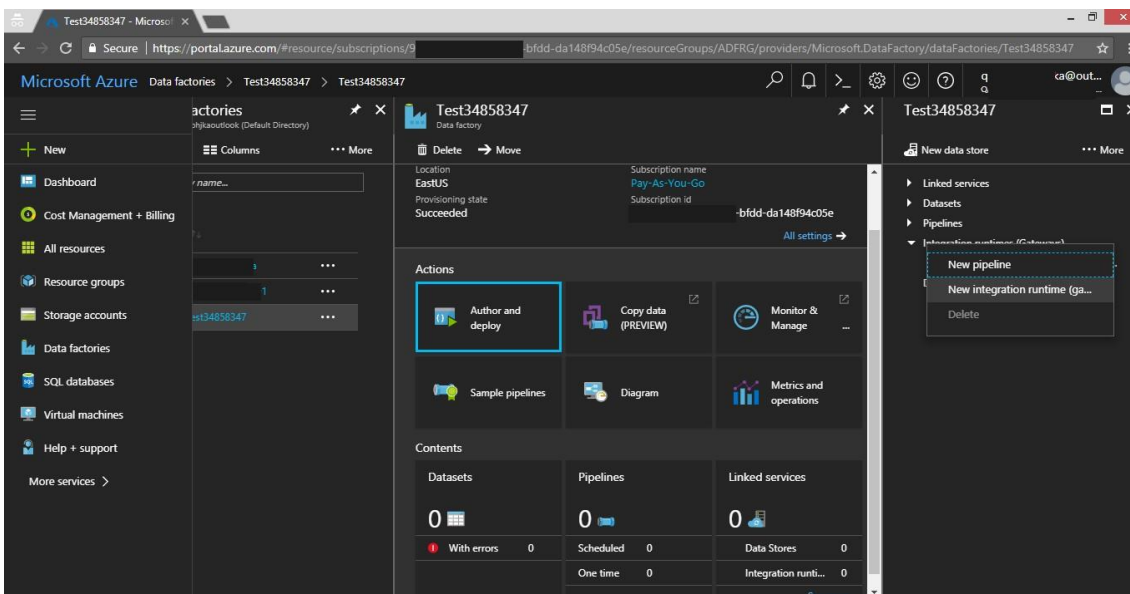
Click Data Explorer Login and create a similar Table in SQL Azure Database

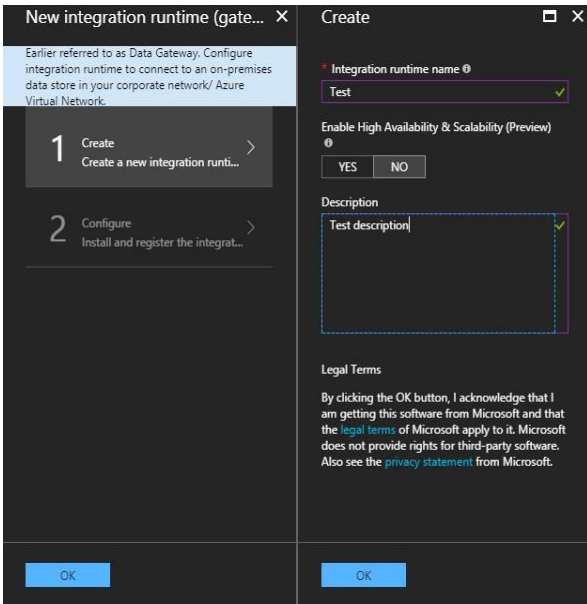


Create an Azure Data Factory. I have considered V1 for this session/tutorial.

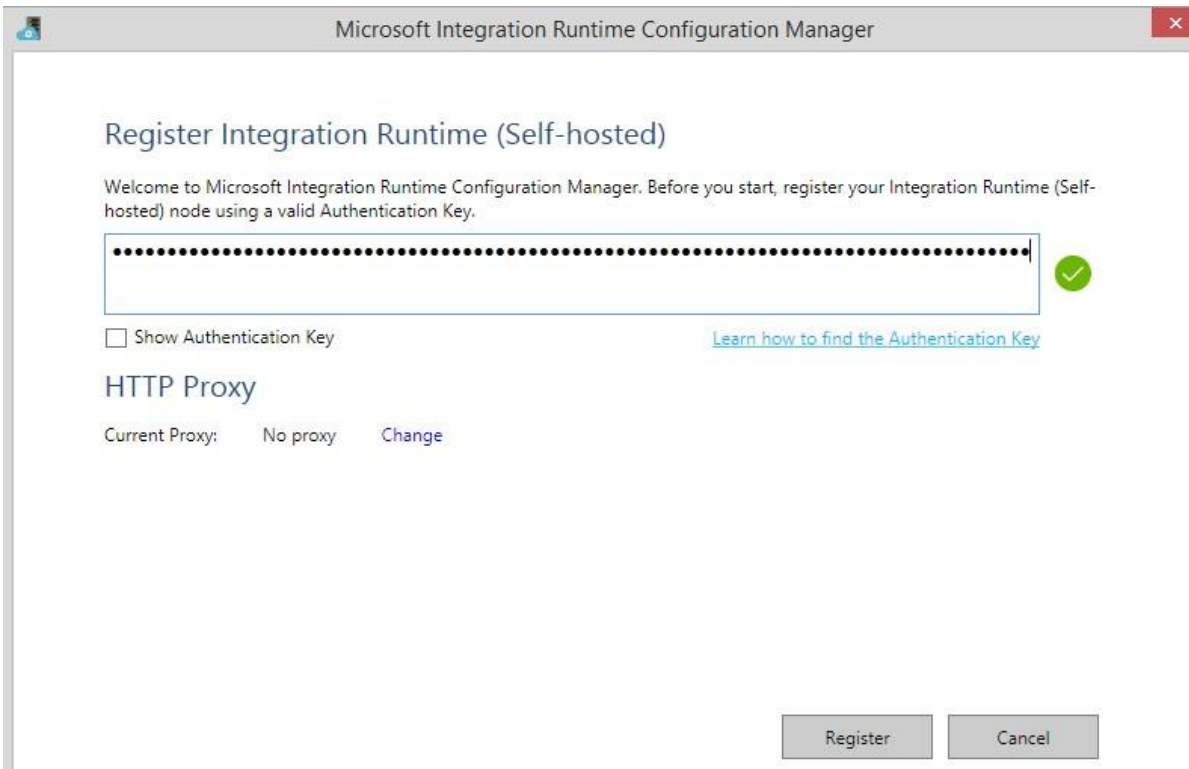
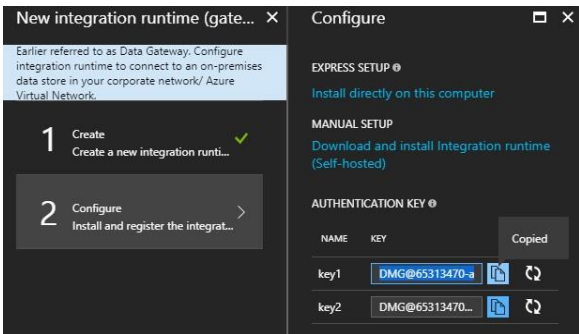


Click Author and Deploy and Create a new Integration Runtime Gateway






Now copy the Authentication Key from Configure Section and paste in the installed **Integration Runtime Application**



Now Click Register

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

Verifying Authentication Key...

Now Click finish

New Integration Runtime (Self-hosted) Node

Enabling Remote Access from intranet let's Credential Manager Application or PowerShell EncryptCredential cmdlet access this self-hosted integration runtime (within same network) remotely for setting linked service credentials.

Enable remote access from intranet

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: Credentials for on-premises data sources are stored locally on this machine. Use the Settings page to regularly back up credentials to a file. You can use this file to restore or recover the Integration Runtime (Self-hosted) in case of a failure. [See Integration Runtime \(Self-hosted\) article for details.](#)

Launch Configuration Manager

Close

Now Launch Configuration Manager

Microsoft Integration Runtime Configuration Manager

Home Settings Diagnostics Update Help

Self-hosted node is connected to the cloud service

Data Factory:	test34858347
Integration Runtime:	Test
Node:	K K

[Stop Service](#)

Data Source Credential ⓘ

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

[Generate Backup](#) [Import Backup](#)

Connected to the cloud service

You can use the Copy Data Preview as the easy wizard process to load data

Otherwise Click Author and Deploy

Create 2 Linked Services

Linked services/Source-OnPremSQL-Test

Code ()

```
{
  "name": "Source-OnPremSQL-test",
  "properties": {
    "hubName": "test34858347_hub",
    "type": "OnPremisesSqlServer",
    "typeProperties": {
      "connectionString": "Data Source=K    k;Initial Catalog = 'test' ;
encryptedCredential=*****;",
      "gatewayName": "Test",
      "userName": "",
      "password": "*****"
    }
  }
}
```

Linked services/Destination-SQLAzure-Test

```
{
  "name": "Destination-SQLAzure-Test",
  "properties": {
    "hubName": "test34858347_hub",
    "type": "AzureSqlDatabase",
    "typeProperties": {
      "connectionString": "Data Source=q*****a.database.windows.net;Initial
```

Catalog=test;Integrated Security=False;User ID=q*****a;Password=*****;Connect
Timeout=30;Encrypt=True"

```
    }  
  }  
}
```

Pipeline Test

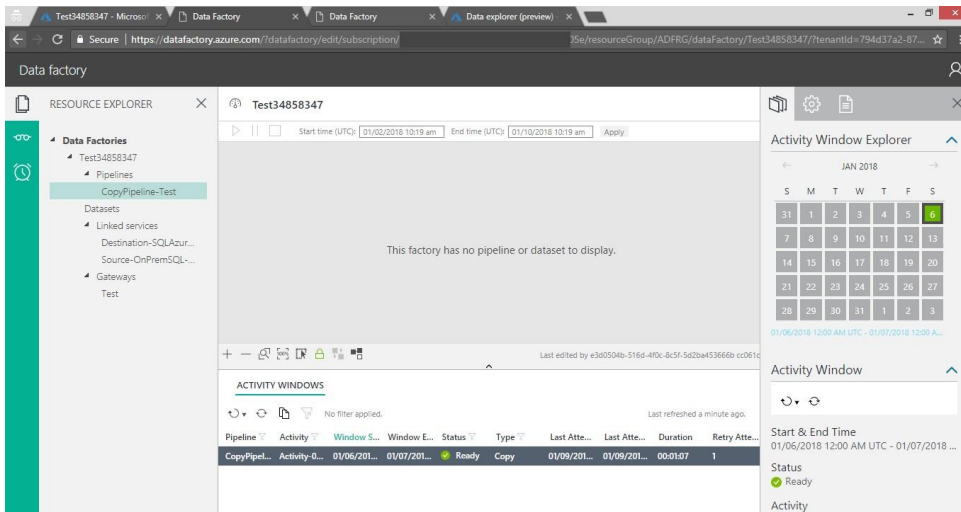
```
{  
  "name": "CopyPipeline-Test",  
  "properties": {  
    "activities": [  
      {  
        "type": "Copy",  
        "typeProperties": {  
          "source": {  
            "type": "SqlSource",  
            "sqlReaderQuery": "select * from [dbo].[SourceTable]"  
          },  
          "sink": {  
            "type": "SqlSink",  
            "writeBatchSize": 0,  
            "writeBatchTimeout": "00:00:00"  
          },  
          "translator": {  
            "type": "TabularTranslator",  
            "columnMappings": "a:a,b:b"  
          }  
        },  
        "inputs": [  
          {  
            "name": "InputDataset-test"  
          }  
        ]  
      }  
    ]  
  }  
}
```

```
    ],
    "outputs": [
      {
        "name": "OutputDataset-test"
      }
    ],
    "policy": {
      "timeout": "1.00:00:00",
      "concurrency": 1,
      "executionPriorityOrder": "NewestFirst",
      "style": "StartOfInterval",
      "retry": 3,
      "longRetry": 0,
      "longRetryInterval": "00:00:00"
    },
    "scheduler": {
      "frequency": "Day",
      "interval": 1
    },
    "name": "Activity-0-[dbo]_[SourceTable]->[dbo]_[DESTTable]"
  }
],
"start": "2018-01-06T10:09:33.14Z",
"end": "2018-01-06T10:09:33.14Z",
"isPaused": false,
"hubName": "test34858347_hub",
"pipelineMode": "OneTime",
"expirationTime": "3.00:00:00",
"datasets": [
  {
    "name": "InputDataset-test",
```

```
"properties": {
  "structure": [
    {
      "name": "a",
      "type": "String"
    },
    {
      "name": "b",
      "type": "Int32"
    }
  ],
  "published": false,
  "type": "SqlServerTable",
  "linkedServiceName": "Source-OnPremSQL-test",
  "typeProperties": {
    "tableName": "[dbo].[SourceTable]"
  },
  "availability": {
    "frequency": "Day",
    "interval": 1
  },
  "external": true,
  "policy": {}
}
{
  "name": "OutputDataset-test",
  "properties": {
    "structure": [
      {
        "name": "a",
```

```
        "type": "String"
      },
      {
        "name": "b",
        "type": "Int32"
      }
    ],
    "published": false,
    "type": "AzureSqlTable",
    "linkedServiceName": "Destination-SQLAzure-Test",
    "typeProperties": {
      "tableName": "[dbo].[DESTTable]"
    },
    "availability": {
      "frequency": "Day",
      "interval": 1
    },
    "external": false,
    "policy": {}
  }
}
]
```

Now post Creation of ADF Pipeline verify the execution in the Monitor and Manage Tab



Now verify the data in the SQL Azure Database

