

# SD WORX | Insights

SD WORX INSIGHTS CLOUD  
UNATTENDED CONNECTION SETUP

<b>UNATTENDED REPORTS REFRESH AND DATA EXTRACTION</b>	<b>2</b>
Solution	2
Setup process overview	3
Key Pair Authentication	4
1.1.1 Generate keys	4
1.1.2 Share your public key and IP address with us	6
1.1.3 Configure ODBC driver	6

# Unattended reports refresh and data extraction

If you wish to have your reports and/or your data warehouse refreshed without human involvement - aka unattended - this setup will allow you to do so.

When using the [generic Insights Cloud Security setup](#), a personal user is mandatory and an MFA-enforced authentication process prevents an unattended data refresh and extraction being performed by a background service on your side.

Allowing a service to launch unattended sessions requires additional security measures to prevent unauthorized access and/or misuse of the service identity. We require a key pair authentication combined with an IP whitelisting for it to be implemented.

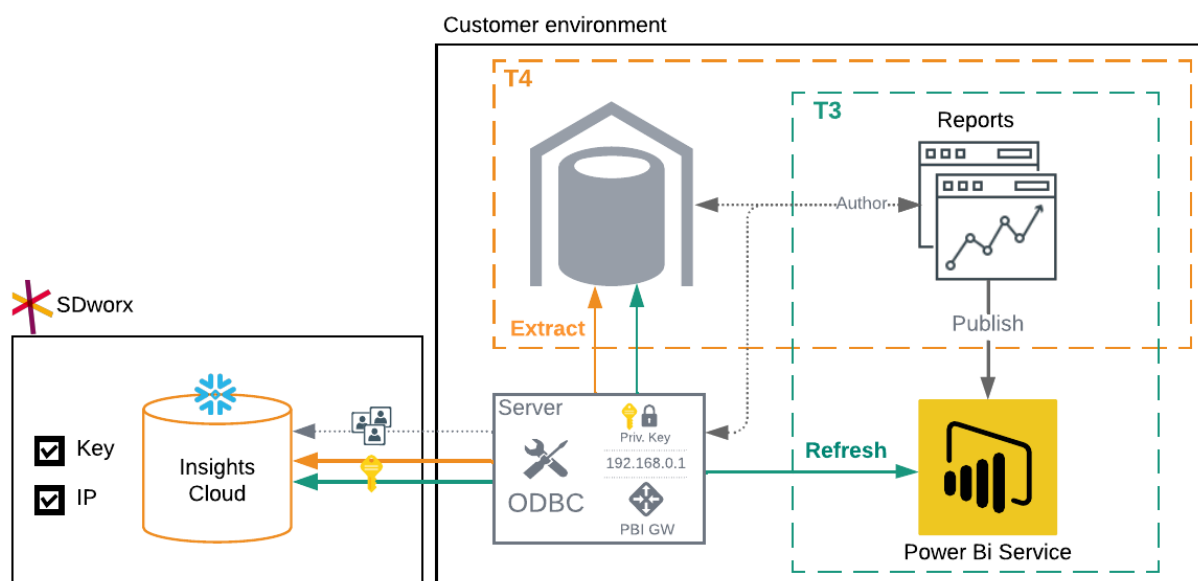
This procedure is intended for usage in a server context, not as a replacement for the authentication method on user machines.

## Solution

While the solution provided is generic for any system that works via an ODBC connection, the schema below is specific for a setup utilizing Microsoft Power BI. If you use another reporting tool we can provide on-demand support for setup guidance.

The Power BI setup requires a server in your environment with:

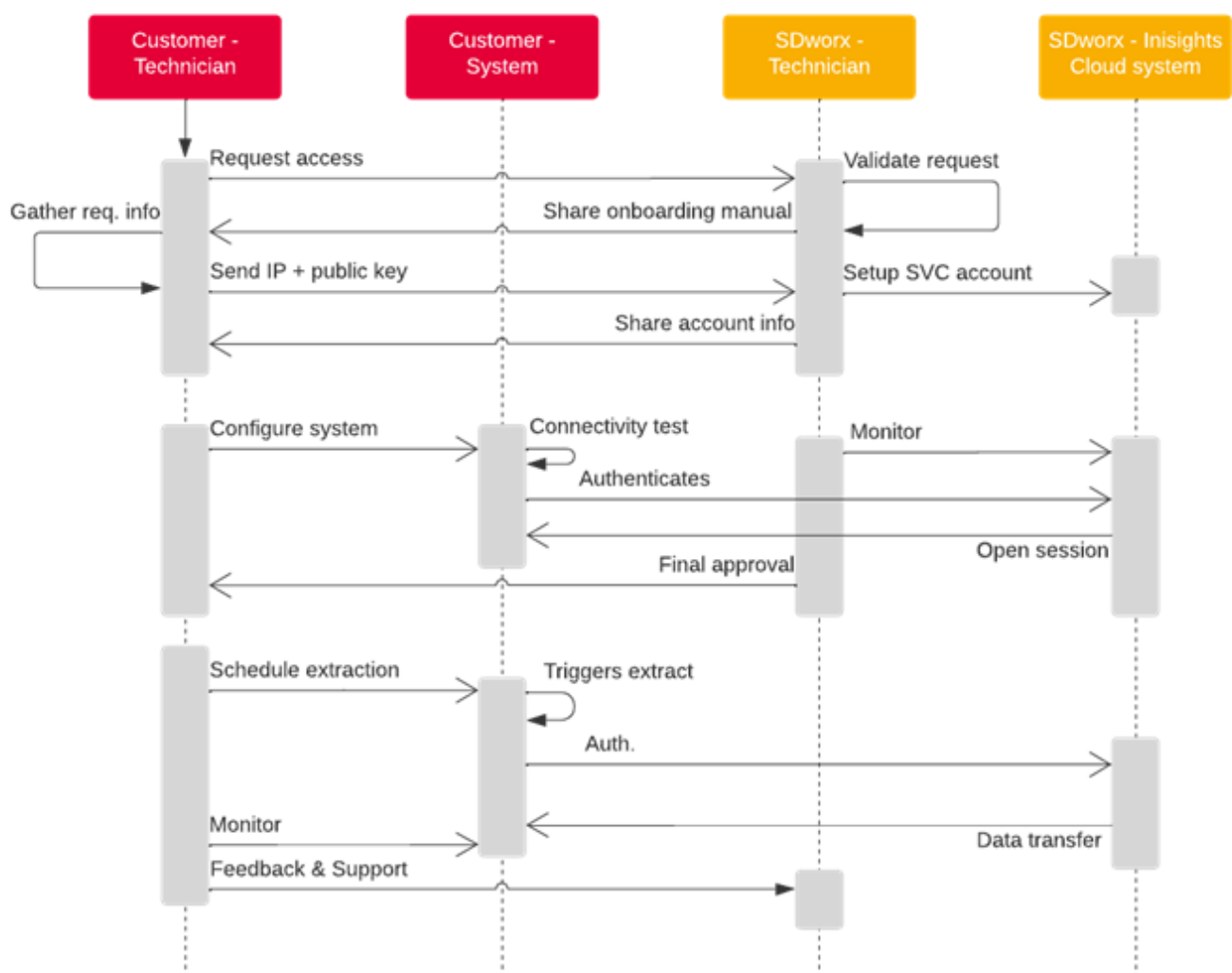
- a static IP address
- a Power BI gateway
- the Snowflake ODBC driver installed and configured with your private key



The gateway is required because Power BI service (cloud) does not support ODBC connection.

Note: Our infrastructure runs in Azure West Europe. It is therefore recommended though not mandatory to deploy your Power BI Gateway server in the same region for optimal performance.


## Setup process overview



## Key Pair Authentication

The following instructions guide you through the key pair generation and configuration.

### 1.1.1 Generate keys

 The command to generate an encrypted key prompts for a passphrase to regulate access to the key. SDworx recommends using a passphrase that complies with PCI DSS standards to protect the locally generated private key. Additionally, Snowflake recommends storing the passphrase in a secure location. If you are using an encrypted key to connect to Snowflake, you will input the passphrase during the initial connection. The passphrase is only used for protecting the private key and will never be sent to Snowflake.

To generate a long and complex passphrase based on PCI DSS standards:

1. Access the [PCI Security Standards Document Library](#).
2. For **PCI DSS**, select the most recent version and your desired language.
3. Complete the form to access the document.
4. Search for Passwords/passphrases must meet the following: and follow the recommendations for password/passphrase requirements, testing, and guidance. Depending on the document version, the phrase is likely located in a section called “Requirement 8: Identify and authenticate access to system components” (or a similar name).

#### 1.1.1.1 Step 1: Generate a private key

To start, open a terminal window and generate a private key.

You can generate either an encrypted version of the private key or an unencrypted version of the private key.

To generate an unencrypted version, use the following command:

```
$ openssl genrsa 2048 | openssl pkcs8 -passout *your password* -topk8 -inform PEM -out rsa_key.p8 -nocrypt
```

To generate an encrypted version, use the following command (which omits “-nocrypt”):

```
$ openssl genrsa 2048 | openssl pkcs8 -passout *your password* -topk8 -v2 des3 -inform PEM -out rsa_key.p8
```

The commands generate a private key in PEM format.

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIE6TAbBgkqhkiG9w0BBQMwDgQILYPyCppzOwECAGgABIIIEyLiGSpeeGSe3xHP1
wHLjfCYycUPennlX2bd8yX8xOxGSGfvB+99+PmSlex0FmY9ov1J8H1H9Y3lMWXbL
...
-----END ENCRYPTED PRIVATE KEY-----
```

#### 1.1.1.2 Step 2: Generate a public key

From the command line, generate the public key by referencing the private key. The following command assumes the private key is encrypted and contained in the file named `rsa_key.p8`.

```
$ openssl rsa -in rsa_key.p8 -pubout -out rsa_key.pub
```

The command generates the public key in PEM format.

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAY+Fw2qv4Roud3l6tjPH4
zxybHjmZ5rhtCz9jppCV8UTWvEXxa88IGRIHbJ/PwKW/mR8LXdfI7l/9vCMXX4mk
...
-----END PUBLIC KEY-----
```

#### 1.1.1.3 Step 3: Store the private and public Keys securely

Copy the public and private key files to a local directory for storage. Record the path to the files. Note that the private key is stored using the PKCS#8 (Public Key Cryptography Standards) format and is encrypted using the passphrase you specified in the previous step.

Take appropriate measures to protect the file from unauthorized access using the file permission mechanism provided by your operating system. It is your responsibility to secure the file when it is not being used.

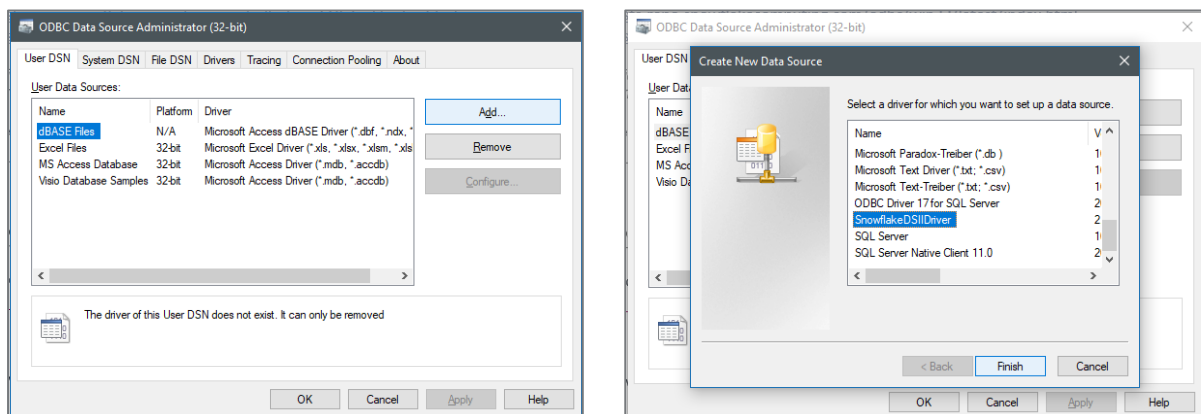
### 1.1.2 Share your public key and IP address with us

Send an email with the public key and your server IP address to [data-insights@sdworx.com](mailto:data-insights@sdworx.com) for SD Worx to register and whitelist them.

### 1.1.3 Configure the ODBC driver

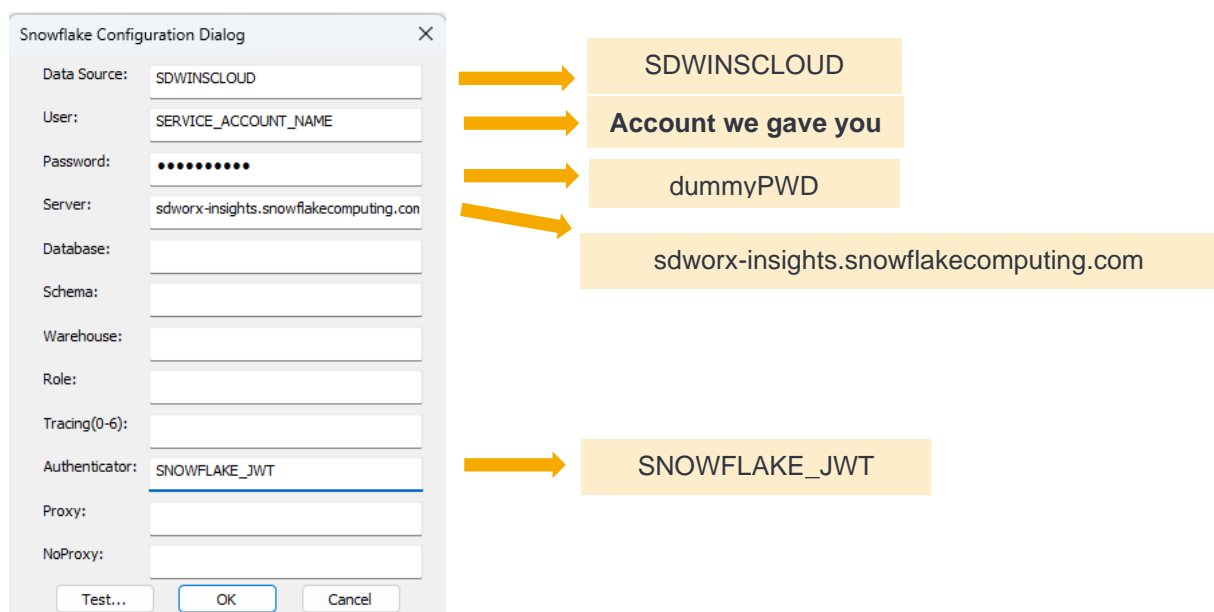
Click on the Windows start button and type 'ODBC' + Enter to start the ODBC Data Source Administrator (32-bit or 64-bit) app.

2. Click Add..., search for SnowflakeDSIIDriver and click Finish.

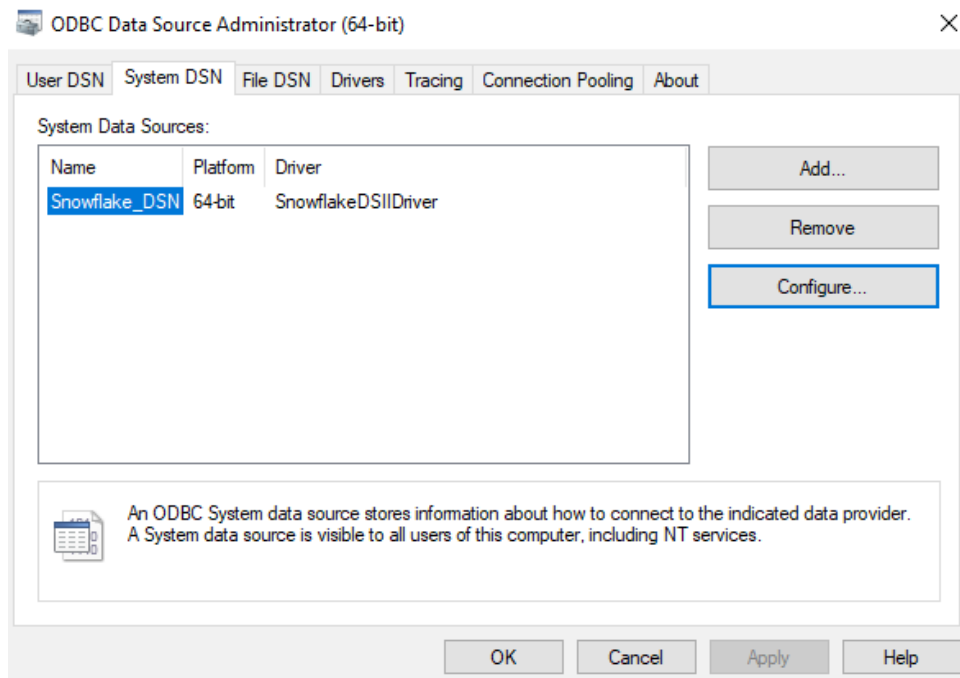


3. Type in a Data Source file name and Copy/Paste the following 5 parameters.

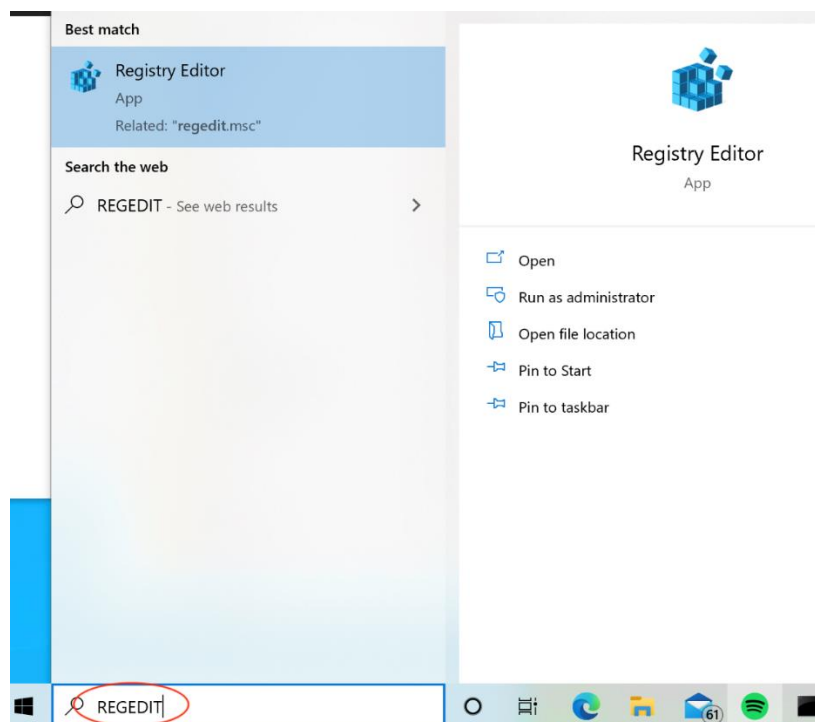
Make sure to provide a dummy password to be able to use the "OK" button to complete the configuration. Click **OK**.



4. The newly created system data source file will be displayed.



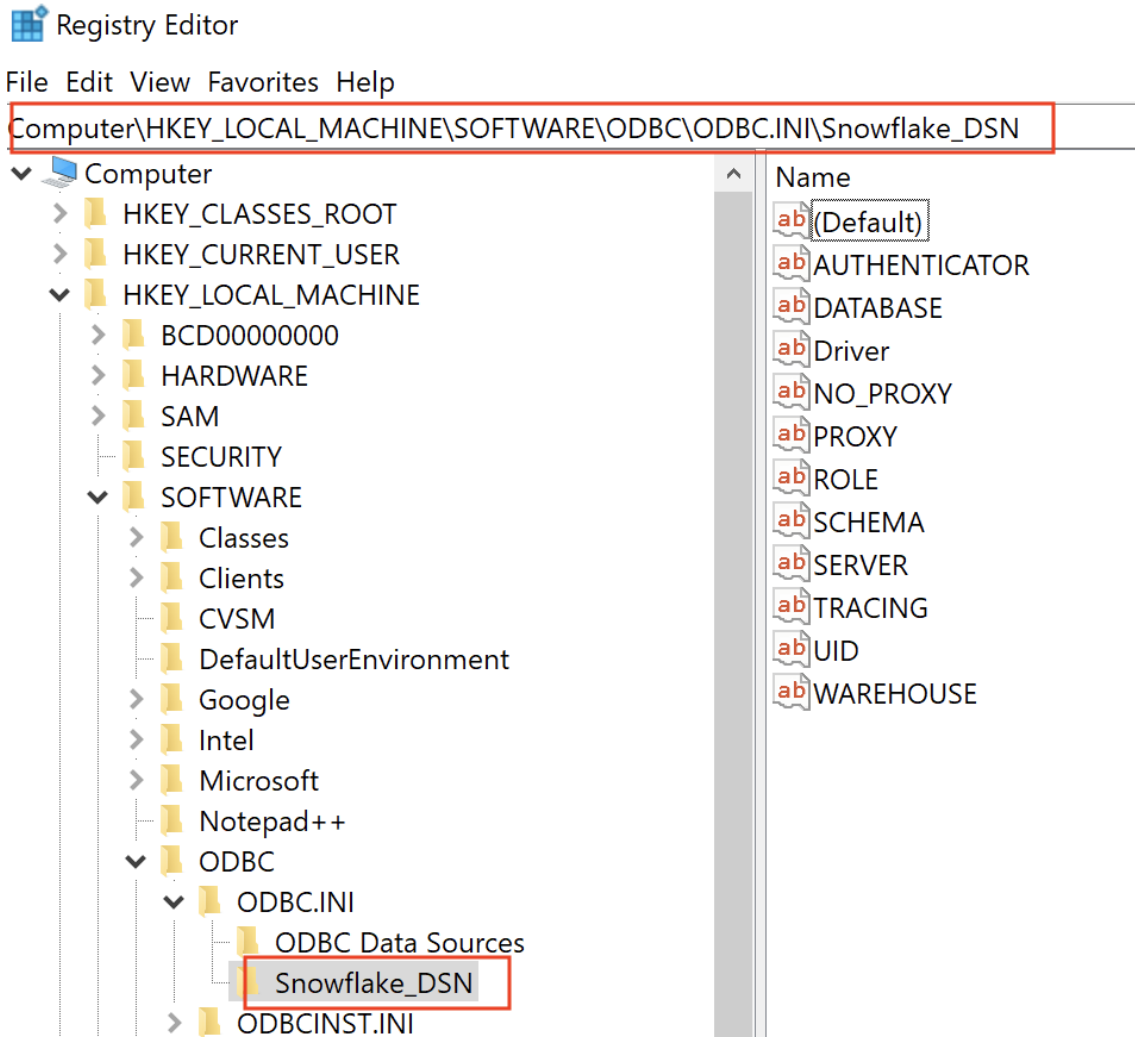
5. Type "**Regedit**" from the Windows search bar. Select the app to bring up the Registry Editor App.



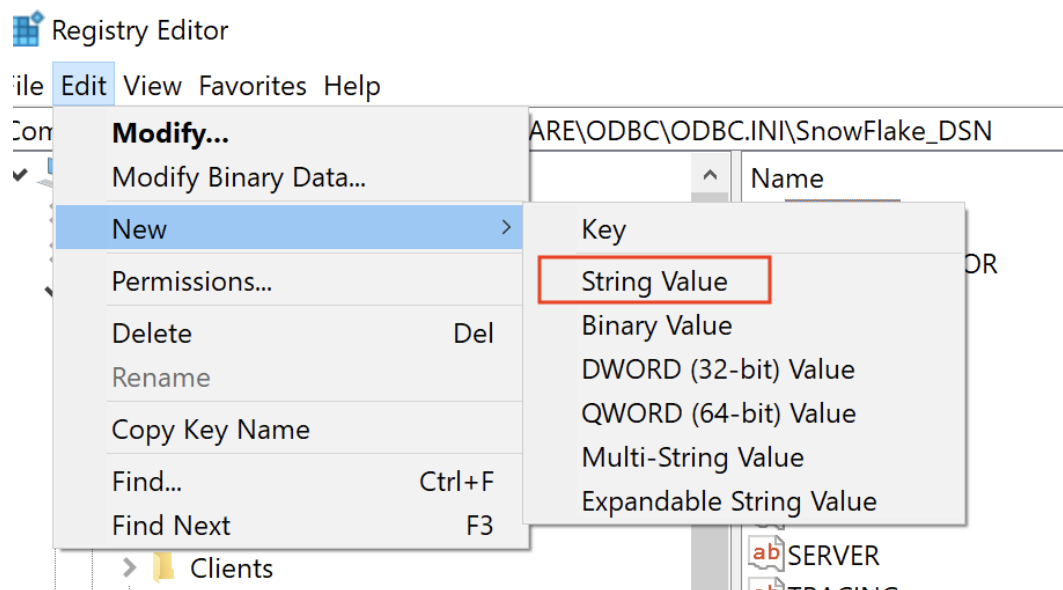


6. Navigate to "**Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\ODBC\ODBC.INI\**", if the installation path is default. Otherwise, the path might vary.

You should see the system data source file just created. Select the data source file. Its parameters will be displayed on the right-hand side.

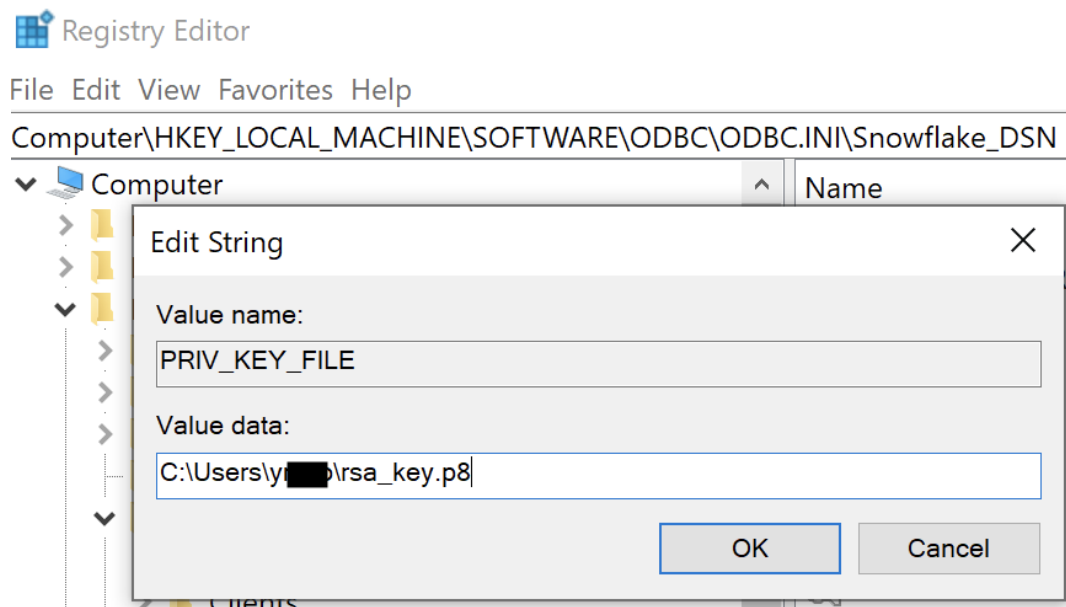


7. From the main menu, select "**Edit**" => "**New**" => "**String Value**".

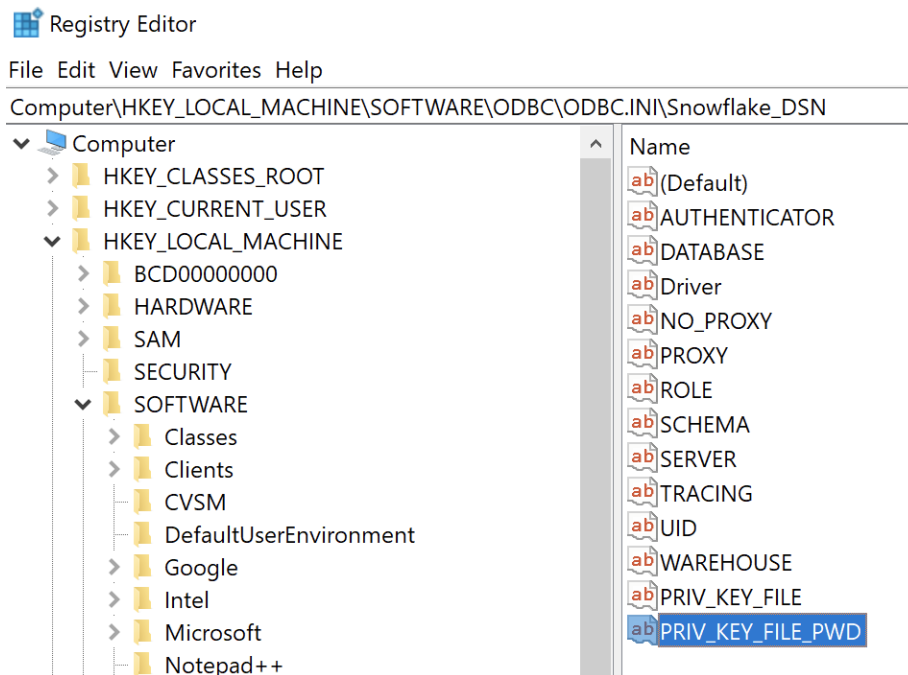


8. Type in "**PRIV\_KEY\_FILE**". Then double click on it to set the path of the private key file. Click **OK**.

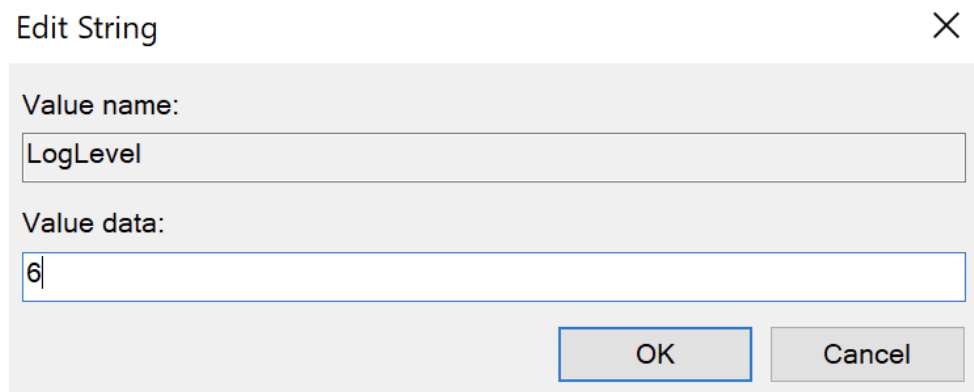
For example:



9. From the main menu, select **"Edit" => "New" => "String Value"** to add another entry for **"PRIV\_KEY\_FILE\_PWD"**. Set the value to your passphrase.

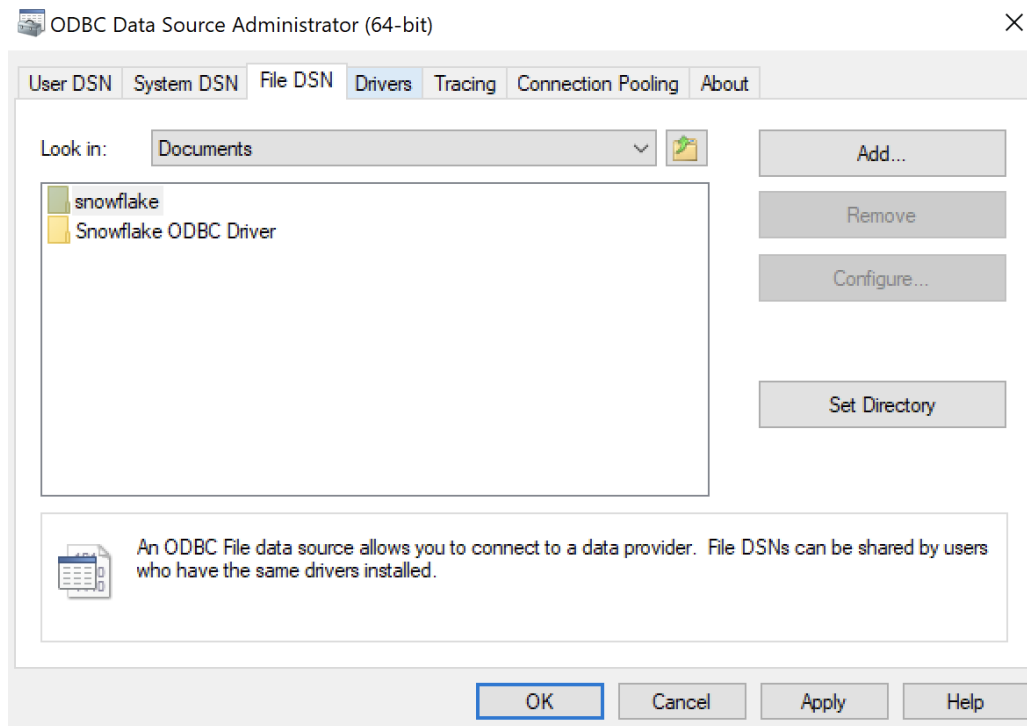


10. (Optional) In case troubleshooting is necessary, from the main menu, select **"Edit" => "New" => "String Value"** to add another parameter **"LogLevel"** and set it to 6.

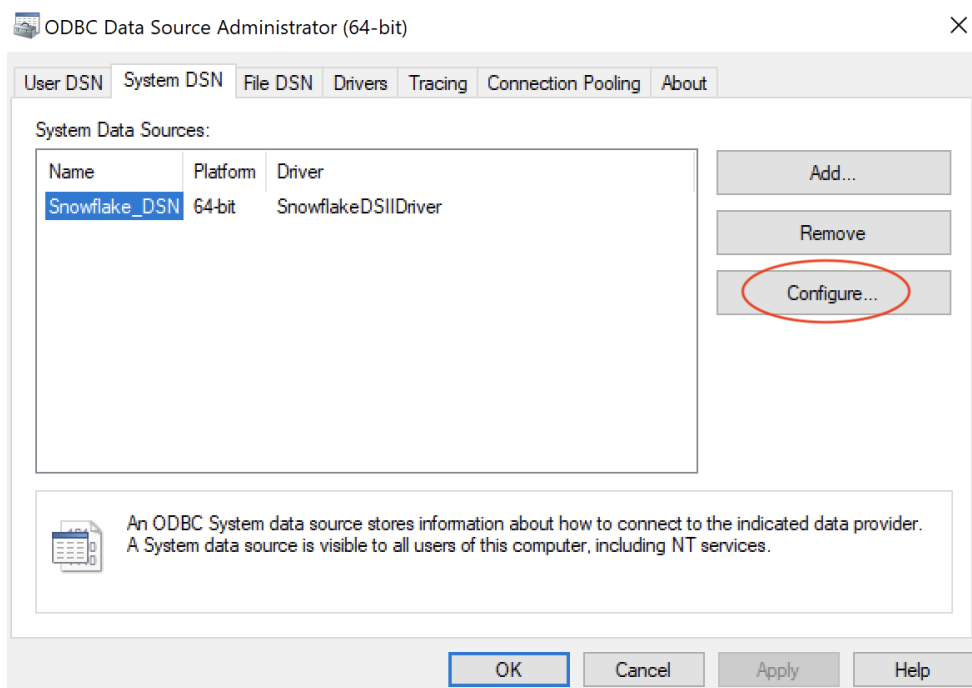


11. The file path is defined under the "**File DSN**" tab.

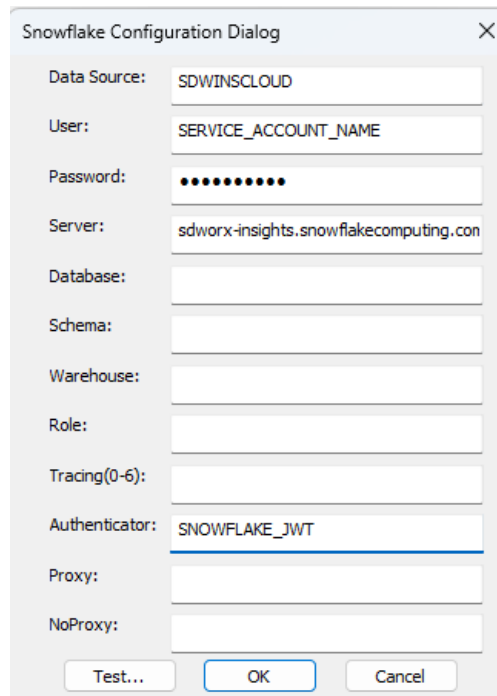
For example:



12. Go back to the ODBC app. Select the system data source file and click on "**Configure**".



13. Click on "**Test**" to test the connection.

A screenshot of the 'Snowflake Configuration Dialog' window. It contains several input fields: 'Data Source' with 'SDWINS CLOUD', 'User' with 'SERVICE\_ACCOUNT\_NAME', 'Password' with masked dots, 'Server' with 'sdworx-insights.snowflakecomputing.com', 'Database', 'Schema', 'Warehouse', 'Role', 'Tracing(0-6)', 'Authenticator' with 'SNOWFLAKE\_JWT', 'Proxy', and 'NoProxy'. At the bottom are 'Test...', 'OK', and 'Cancel' buttons.

Snowflake Configuration Dialog

Data Source: SDWINS CLOUD

User: SERVICE\_ACCOUNT\_NAME

Password: .....

Server: sdworx-insights.snowflakecomputing.com

Database:

Schema:

Warehouse:

Role:

Tracing(0-6):

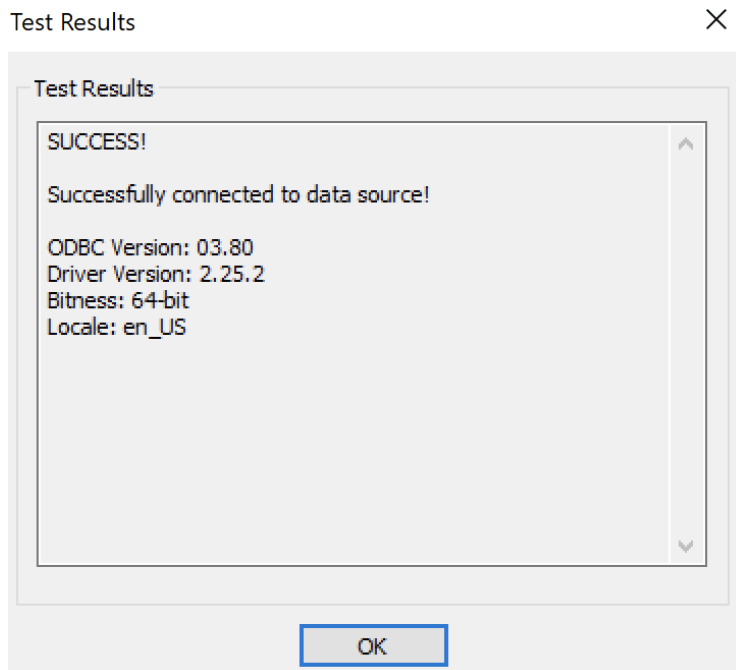
Authenticator: SNOWFLAKE\_JWT

Proxy:

NoProxy:

Test... OK Cancel

14. If the connection is successful, you should see the window below.

A screenshot of the 'Test Results' dialog box. It shows a message 'SUCCESS!' followed by 'Successfully connected to data source!'. Below this, it lists system information: 'ODBC Version: 03.80', 'Driver Version: 2.25.2', 'Bitness: 64-bit', and 'Locale: en\_US'. An 'OK' button is at the bottom.

Test Results

Test Results

SUCCESS!

Successfully connected to data source!

ODBC Version: 03.80  
Driver Version: 2.25.2  
Bitness: 64-bit  
Locale: en\_US

OK

15. In case troubleshooting is needed, check the ODBC logs generated in the directory specified above (step 9 and 10).