installation_and_setup.md 2025-07-29



Installing dbt (Data Build Tool) with Snowflake

This guide walks you through installing **dbt Core** with the **Snowflake adapter** on **Windows**, **Linux**, or **macOS**, including Python setup best practices.

Prerequisites

- Python (3.8–3.11 recommended)
- pip

☑ Recommended:

Use **pyenv** to manage multiple Python versions.

Use a virtual environment (venv or virtualenv) to isolate your project environment.

Windows

1. Install Python

Download from python.org

During installation, check: Add Python to PATH

2. Install dbt with Snowflake adapter

```
pip install dbt-snowflake
```

3. Verify Installation

```
dbt --version
```

Linux (Ubuntu/Debian)

1. Install Python and pip

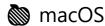
```
sudo apt update
sudo apt install -y python3 python3-pip
```

2. Install dbt with Snowflake adapter

```
pip install dbt-snowflake
```

3. Verify Installation

```
dbt --version
```



1. Install Python

Install via Homebrew or python.org

2. Install dbt with Snowflake adapter

```
pip install dbt-snowflake
```

3. Verify Installation

dbt --version

Snowflake Authentication

By default, dbt connects to Snowflake using **username/password login**, which may prompt for **browser-based multi-factor authentication (MFA)** if the user has 2FA enabled.

Development Use Case

To avoid 2FA prompts during development or in CI environments, it's recommended to create a **dedicated service user** with **key-pair authentication**.

- ☑ Key-Pair Authentication Steps
 - 1. Generate an RSA private key and store it securely (e.g., ~/.ssh/snowflake_key.pem)
 - 2. Upload the public key to the Snowflake user
 - 3. In your profiles.yml, set:

```
private_key_path: /path/to/snowflake_key.pem
```

This allows fully automated access without needing a browser prompt or 2FA.

This is especially important when running dbt inside CI/CD pipelines or scheduled jobs.

Helpful Links

- dbt Core Installation Docs
- dbt Snowflake Adapter Docs
- pyenv GitHub
- Python Virtual Environments
- development_process