1. Prerequisites & Setup

Before you begin, ensure you have the following:

1. **A Redshift Cluster:** You need a running Amazon Redshift cluster (provisioned or Serverless).
2. **A Database User & Permissions:** Create a dedicated user for dbt (e.g., dbt\_user) with appropriate permissions on the schemas it will use (typically CREATE on a dbt\_ schema and SELECT on source schemas).
3. **Network Access:** Your dbt execution environment (your laptop, a CI/CD runner, etc.) must have network access to your Redshift cluster on port 5439.
4. **Python Environment:** dbt is written in Python. Install it using pip in a virtual environment.

Installation and Initial Configuration

1. **Install dbt Redshift Adapter:**

bash

pip install dbt-redshift

1. **Initialize a dbt Project:**

bash

dbt init my\_dbt\_project

cd my\_dbt\_project

This creates the standard dbt project structure.

1. **Configure Profile (**~/.dbt/profiles.yml**):** This file contains your connection details.

yaml

my\_dbt\_project:

target: dev

outputs:

dev:

type: redshift

host: your-cluster.abc123xyz789.us-east-1.redshift.amazonaws.com # Without the port

port: 5439

user: dbt\_user

password: your\_secure\_password

dbname: dev

schema: dbt\_myuser # The schema where dbt will build its models

threads: 4 # Number of parallel connections dbt can use

# Optional: Use IAM authentication instead of password

# method: iam

# cluster\_identifier: 'your-redshift-cluster-identifier'

# iam\_profile: 'your-aws-profile-name'

# region: 'us-east-1'

**Security Note:** Never commit your profiles.yml to version control. Use environment variables for secrets (e.g., password: "{{ env\_var('DBT\_REDSHIFT\_PASSWORD') }}").

2. Key Configuration in Your dbt\_project.yml

Configure your project file to work optimally with Redshift.

yaml

name: 'my\_dbt\_project'

version: '1.0.0'

config-version: 2

profile: 'my\_dbt\_project' # This must match the profile name

# Redshift-specific settings

models:

my\_dbt\_project:

# Materialization for models in the 'models' directory

+materialized: view

# Example of configuring a specific model

marts:

+materialized: table

+dist: key # Or 'all', 'even'

+sort: ['date\_day', 'user\_id'] # Define compound sort keys

# Define which macros to run on start/end of the dbt run

on-run-start:

- "{{ redshift\_begin\_transaction() }}" # Optional, uses a transaction for the whole run

on-run-end:

- "{{ redshift\_commit\_transaction() }}" # Optional

- "{{ vacuum\_model\_workflow() }}" # Example custom macro to run VACUUM