Lab 1: Preflight & Environment Readiness

Course: IT4065C – Data Technologies Administration
Repository: https://github.com/ntious/IT4065C-Labs
Time Required: ≤ 1 hour
Environment: Individual Ubuntu VM (VMware Cloud Services)

Goal of Lab 1

By the end of this lab, you will be able to:

1. Log into your Ubuntu virtual machine
2. Clone the course GitHub repository
3. Install PostgreSQL locally on your VM
4. Install dbt for PostgreSQL
5. Configure dbt to connect to a local database
6. Run dbt debug successfully
7. Run basic SQL queries against a local database

Note:
No transformations, no modeling, and no dbt run are required in this lab.

Step 1: Choose Your Ubuntu Environment (Required)

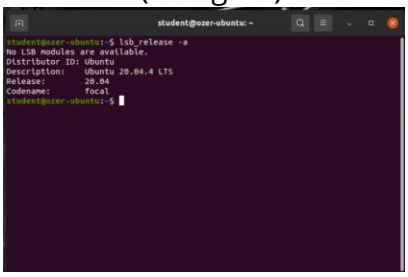
In **Step 1**, you must choose **one** of the following options:

- **Option A:** Use the UC-provided Ubuntu Virtual Machine (Sandbox)
- **Option B:** Install Ubuntu on your own Windows machine using the Microsoft Store

👉 **You only need to complete ONE option. Do not do both.**

Step 1.1 (OptionA): Log into Your Ubuntu VM (Using the UC Sandbox)

1. Log into your UC Sandbox Ubuntu VM at:
   https://range.ocri.io
2. Use the following credentials (UC Sandbox only):
   - **Ubuntu VM Password:** Pa$$w0rd
3. Once logged in, open a Terminal window.
4. Confirm the operating system by running:
   lsb_release -a
5. You should see Ubuntu 22.04 LTS (or higher)



⚠️ If the VM does not load correctly or you cannot log in, stop and contact your instructor or IT support before proceeding.

Step 1.2 (Option B): Creating Your Own Ubuntu Environment on Windows (Microsoft Store)

In this step, you will install Ubuntu Linux on Windows using the Microsoft Store. This creates a Linux environment using Windows Subsystem for Linux (WSL), no full virtual machine is required.
⚠️ This option is required only if you are not using the UC Sandbox VM.

Requirements
Before you begin, make sure you have:

- Windows 10 (version 21H2 or later) or Windows 11
- Administrator access on your computer
- Internet connection

Step 1.2.1: Open the Microsoft Store

1. Click the Start menu
2. Type Microsoft Store
3. Open the Microsoft Store app

Step 1.2.2: Download Ubuntu

1. In the Microsoft Store search bar, type Ubuntu
2. Select Ubuntu 22.04 LTS
    - ⚠️ Do not select Ubuntu Server or preview versions
3. Click Get or Install
4. Wait for the download to complete

Step 1.2.3: Launch Ubuntu

1. Once installation finishes, click Open
   (or open Ubuntu from the Start menu)

Ubuntu will initialize for the first time. This may take a few minutes.

Step 1.2.4: Create Your Linux User Account
When prompted in the Ubuntu window:

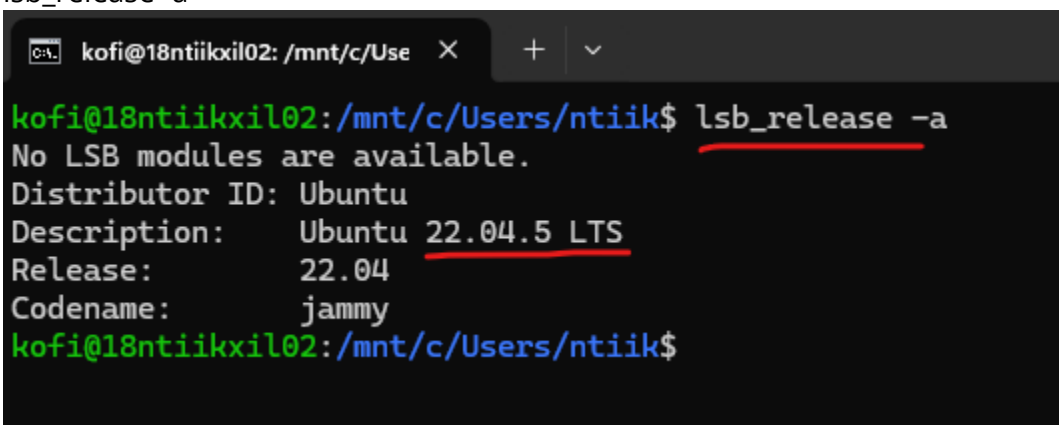
1. Enter a Linux username
    - Lowercase letters only (e.g., student, ubuntu)
2. Enter a password
    - You will not see characters as you type; this is normal
3. Confirm the password

⚠️ Remember this password. You will need it throughout the course.

Step 1.2.5: Verify Ubuntu Installation

Once you see the Linux prompt, run:

lsb_release -a



Step 2: Update System Packages (Required for All Students)

Regardless of which option you chose, run the following commands **once**:

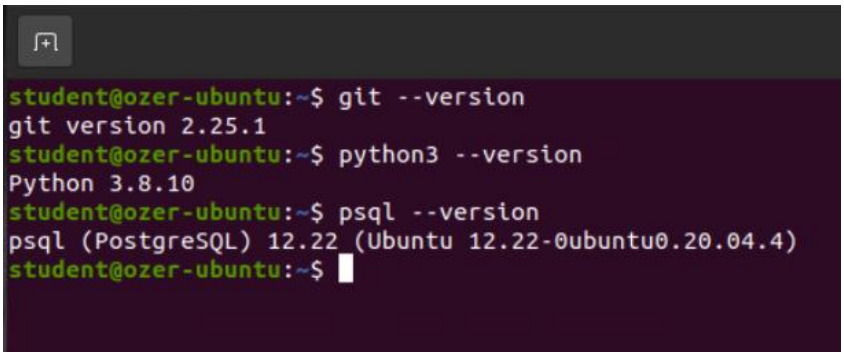
1. Run the following commands once:
   sudo apt update
   sudo apt upgrade -y

This helps prevent package conflicts later.

Step 3: Install Required System Tools

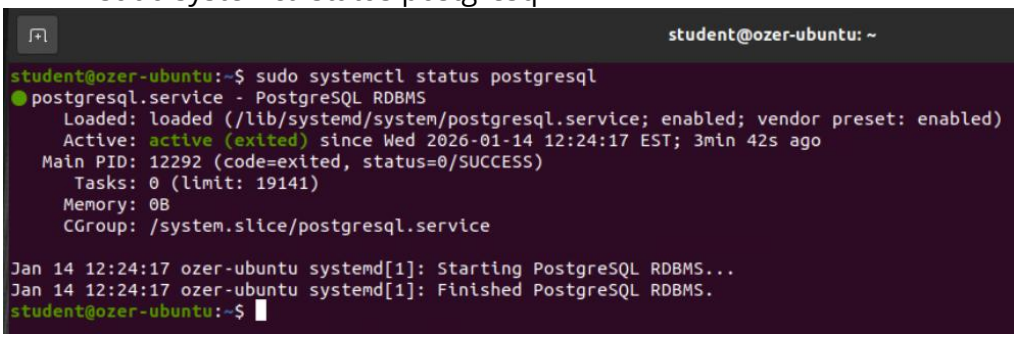
1. Install Git, Python, and PostgreSQL:
   sudo apt install -y git python3 python3-pip python3-venv postgresql postgresql-contrib
2. Verify each installation:
   git --version
   python3 --version
   psql --version

Each command should return a version number.

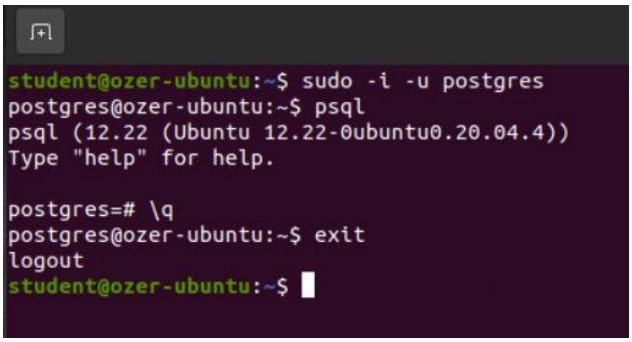


Step 4: Create Database and Schemas

1. Check PostgreSQL service status:
   sudo systemctl status postgresql



2. Switch to the postgres user:
   sudo -i -u postgres
3. Open the PostgreSQL shell:
   Psql
4. Exit PostgreSQL:
   \q
5. Exit the postgres user:
   exit



Step 5: Create Your Local Course Database and Schema

1. From your normal Ubuntu user account, run:
sudo -u postgres createdb it4065c || echo "Database already exists, continuing..."
sudo -u postgres psql it4065c

If you see "database already exists," this is expected. Continue to the next step.

2. Inside PostgreSQL, run:
CREATE SCHEMA raw;
CREATE SCHEMA student_kofi;
3. \q

This simulates schema-level governance isolation used later in the course.

```
student@ozer-ubuntu:~$ sudo -u postgres createdb it4065c || echo "Database already exixts, continuing..."
student@ozer-ubuntu:~$ sudo -u postgres psql it4065c
psql (12.22 (Ubuntu 12.22-0ubuntu0.20.04.4))
Type "help" for help.

it4065c=# CREATE SCHEMA raw;
CREATE SCHEMA
it4065c=# CREATE SCHEMA student_kofi;
CREATE SCHEMA
it4065c=# \q
student@ozer-ubuntu:~$
```

4. Verify Schema Creation (Optional but Recommended)
   sudo -u postgres psql it4065c -c "\dn+"

```
student@ozer-ubuntu:~$ sudo -u postgres psql it4065c -c"\dn+"
                          List of schemas
    Name     |  Owner   |  Access privileges   |      Description
-------------+----------+----------------------+------------------------
 public      | postgres | postgres=UC/postgres+| standard public schema
             |          | =UC/postgres         |
 raw         | postgres |                      |
 student_kofi | postgres |                      |
(3 rows)

student@ozer-ubuntu:~$
```

Step 6: Clone the Course GitHub Repository

1. From your home directory run:
   git clone https://github.com/ntious/IT4065C-Labs.git
   cd IT4065C-Labs
2. Confirm the repository structure:
   ls

You should see the following folders:

- docs
- labs
- dbt
- scripts
- submissions

📸 **Screenshot 1:** Take a screenshot showing the repository structure (docs, labs, dbt).

```
kofi@18ntiikxil02:~$ git clone https://github.com/ntious/IT4065C-Labs.git
Cloning into 'IT4065C-Labs'...
remote: Enumerating objects: 77, done.
remote: Counting objects: 100% (77/77), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 77 (delta 23), reused 9 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (77/77), 23.24 KiB | 1.11 MiB/s, done.
Resolving deltas: 100% (23/23), done.
kofi@18ntiikxil02:~$ cd IT4065C-Labs
kofi@18ntiikxil02:~/IT4065C-Labs$ ls
LICENSE  README.md  dbt  docs  labs  scripts  submissions
kofi@18ntiikxil02:~/IT4065C-Labs$
```

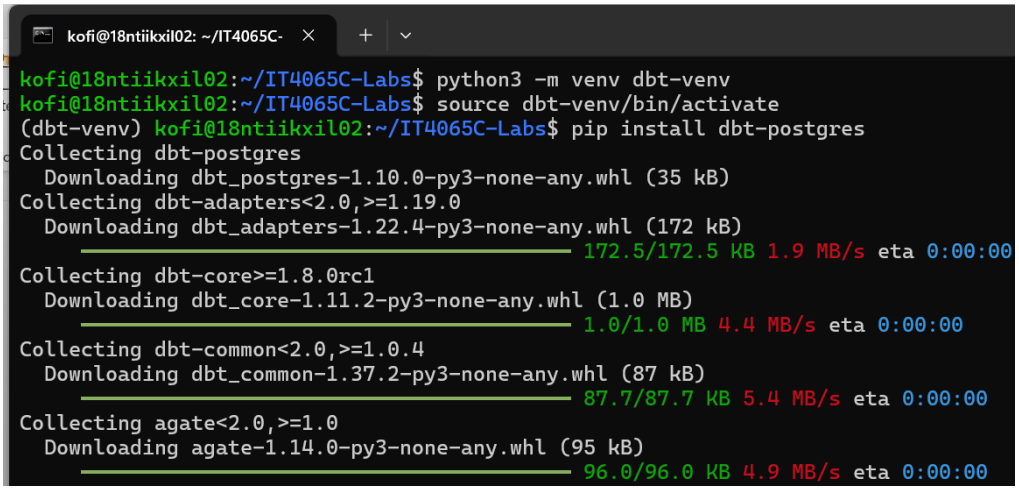
You are now in the root directory of the course repository. All lab instructions will assume you are working from this location unless stated otherwise.

Step 7: Create and Activate dbt Virtual Environment

python3 -m venv dbt-venv

source dbt-venv/bin/activate

pip install dbt-postgres

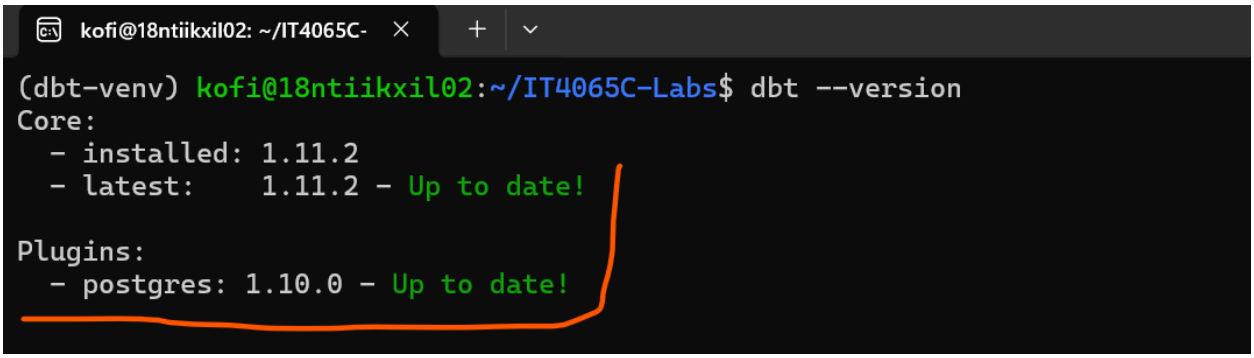


1. Verify installation:
   Run:
   dbt --version

You must see the **Postgres adapter** listed.

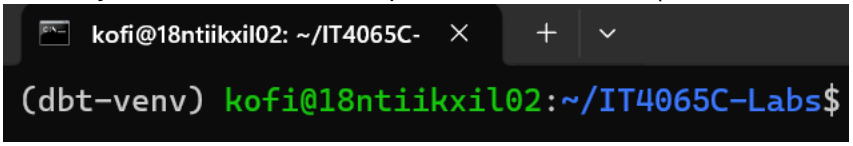
Any time you work on dbt labs, you **must activate the virtual environment first**:

source ~/IT4065C-Labs/dbt-venv/bin/activate

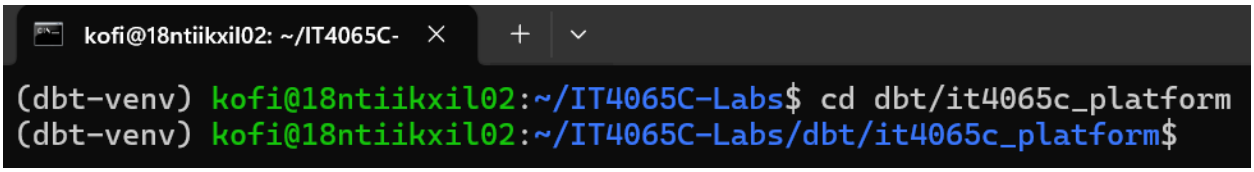


Step 8: Configure dbt profiles.yml (Project Profile)

1. Ensure you are in the course repo and venv is active (as shown below)

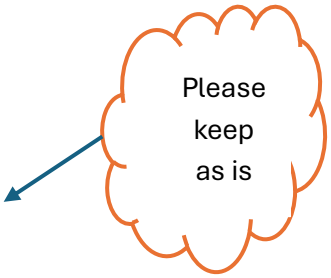


2. Navigate to the dbt project folder
   cd dbt/it4065c_platform



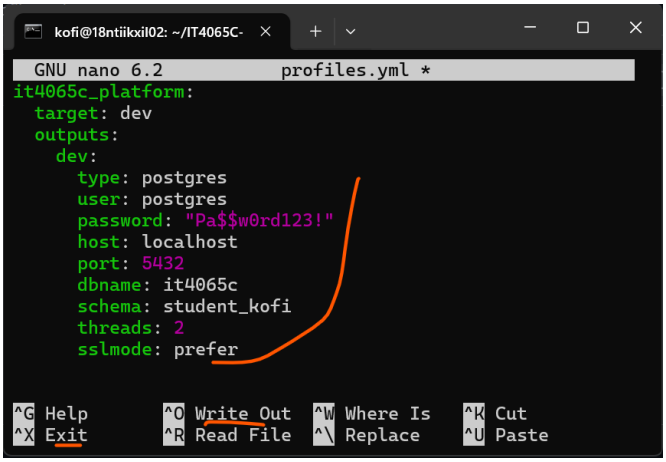
3. Edit the project profile
   nano profiles.yml
4. Set the profile to this exact configuration (copy and paste)

```
it4065c_platform:
 target: dev
 outputs:
  dev:
   type: postgres
   user: postgres
   password: "Pa$$w0rd123!"
   host: localhost
   port: 5432
   dbname: it4065c
   schema: student_kofi
   threads: 2
   sslmode: prefer
```



Note: Save changes and exit:

- **CTRL + O → Enter**
- **CTRL + X**



Note: This password was set earlier during database setup to allow dbt and psql to authenticate over localhost. Use it exactly as shown.

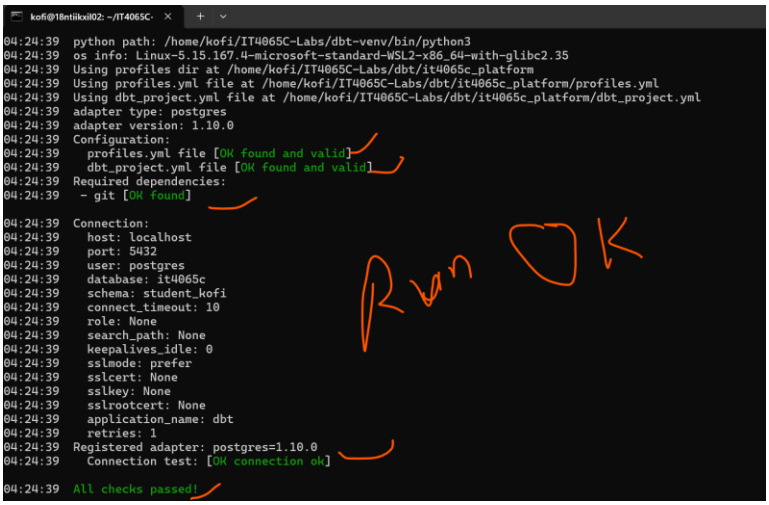
Step 9: dbt Debug

1. Run:
   dbt debug

Success Criteria

You must see:

- Profile loaded successfully
- Connection test: OK

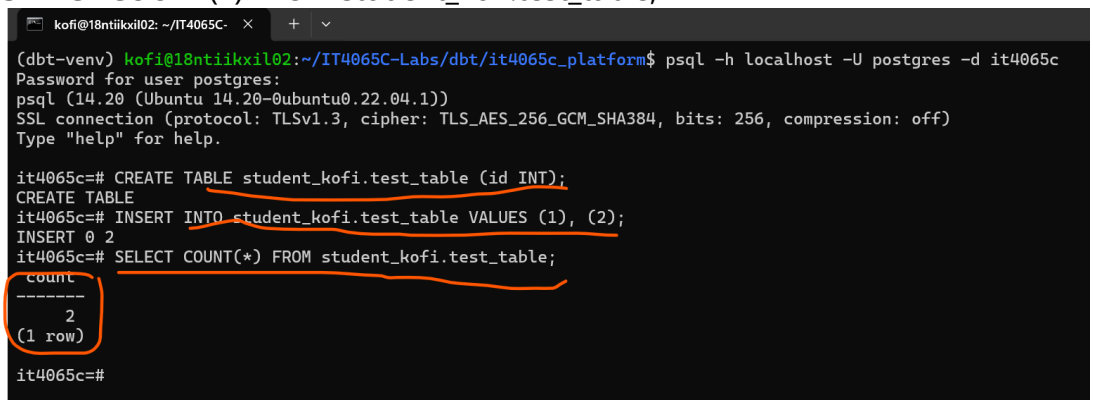


If this fails, consult docs/troubleshooting.md.

📸 **Screenshot 2:** Take a screenshot showing successful dbt debug.

Step 10: Basic SQL Validation

1. Open PostgreSQL using the postgres user:
   Run:
   psql -h localhost -U postgres -d it4065c
   Use password: Pa$$w0rd123!
2. Run:
   CREATE TABLE student_kofi.test_table (id INT);
   INSERT INTO student_kofi.test_table VALUES (1), (2);
   SELECT COUNT(*) FROM student_kofi.test_table;



📸 **Screenshot 3:**Take a screenshot showing the SELECT COUNT(*) result.

1. Exit PostgreSQL:
   \q

Step 11: Lab 1 Deliverables

Submit **one MS Word or PDF file** on Canvas containing:

1. Screenshot 1: Repository structure (docs, labs, dbt)
2. Screenshot 2: Successful dbt debug output
3. Screenshot 3: SELECT COUNT(*) query output
4. Short reflection (3–5 sentences):
   What schema am I responsible for, and why does schema isolation matter in governed data systems?

If you exit or reboot your VM, resume by:
1. Opening a terminal
   cd ~/IT4065C-Labs
   source dbt-venv/bin/activate