2024-05-12 Arch DBT Finance Team Enablement - Set Up

Set Up Instructions

1. Create your Snowflake Trial Account

1. Create a Trial Account here.
    1. Select the following options when prompted: Enterprise Edition, on AWS, location Europe (Paris)
    2. Review and agree to Terms and Conditions
    3. Press "Get Started"
2. Click the activation link in the sign-up email
3. Set the username and password as instructed. Follow the convention "`<firstname><lastname>`" for the Username.
    1. Press Get Started when complete
4. Run the following set up script
5. In the Snowflake UI, run the provided set up script ("arch_snowflake_setup.sql")
    1. Using the Left Hand Side navigation pane, access the worksheets section
    2. In the worksheets section, select the `run all` option by clicking the drop down arrow next to the blue "+" button.
    3. Copy in the provided "arch_snowflake_setup.sql" setup code
    4. Run the code by pressing the blue play button in the top right corner. This should return the following response: "Statement executed successfully."
6. Bookmark or save the Snowflake account URL, which takes the following structure: `https://<organisation identifier>-<account locator>.snowflakecomputing.com/console/login`
7. Share details of the account to the Kubrick Team by emailing your name and snowflake account URL to bavandeepmalhi@kubrickgroup.com.

```
-----------------------------------------

-- dbt credentials

-----------------------------------------

USE ROLE securityadmin;

-- dbt roles

CREATE OR REPLACE ROLE dbt_dev_role;

CREATE OR REPLACE ROLE dbt_prod_role;
```

```
-- create dbt role to be used as part of enablement sessions

CREATE OR REPLACE USER dbt_user PASSWORD = "Arch2024Enablement!";



-- create admin user to be used by Kubrick Team should access be required

CREATE OR REPLACE USER KUBRICKADMIN

PASSWORD = 'Arch2024Enablement!'

LOGIN_NAME = 'KUBRICKADMIN'

EMAIL = 'bavandeepmalhi@kubrickgroup.com'

MUST_CHANGE_PASSWORD = false

DEFAULT_WAREHOUSE = COMPUTE_WH;



GRANT ROLE ACCOUNTADMIN to USER KUBRICKADMIN;



GRANT ROLE dbt_dev_role,dbt_prod_role TO USER dbt_user;

GRANT ROLE dbt_dev_role,dbt_prod_role TO ROLE sysadmin;



-----------------------------------------

-- dbt objects

-----------------------------------------

USE ROLE sysadmin;



CREATE OR REPLACE WAREHOUSE dbt_dev_wh WITH WAREHOUSE_SIZE = 'XSMALL' AUTO_SUSPEND = 60
AUTO_RESUME = TRUE MIN_CLUSTER_COUNT = 1 MAX_CLUSTER_COUNT = 1 INITIALLY_SUSPENDED =
TRUE;

CREATE OR REPLACE WAREHOUSE dbt_dev_heavy_wh WITH WAREHOUSE_SIZE = 'LARGE' AUTO_SUSPEND
= 60 AUTO_RESUME = TRUE MIN_CLUSTER_COUNT = 1 MAX_CLUSTER_COUNT = 1 INITIALLY_SUSPENDED
= TRUE;

CREATE OR REPLACE WAREHOUSE dbt_prod_wh WITH WAREHOUSE_SIZE = 'XSMALL' AUTO_SUSPEND =
60 AUTO_RESUME = TRUE MIN_CLUSTER_COUNT = 1 MAX_CLUSTER_COUNT = 1 INITIALLY_SUSPENDED =
TRUE;
```

```
CREATE OR REPLACE WAREHOUSE dbt_prod_heavy_wh WITH WAREHOUSE_SIZE = 'LARGE'
AUTO_SUSPEND = 60 AUTO_RESUME = TRUE MIN_CLUSTER_COUNT = 1 MAX_CLUSTER_COUNT = 1
INITIALLY_SUSPENDED = TRUE;



GRANT ALL ON WAREHOUSE dbt_dev_wh TO ROLE dbt_dev_role;

GRANT ALL ON WAREHOUSE dbt_dev_heavy_wh TO ROLE dbt_dev_role;

GRANT ALL ON WAREHOUSE dbt_prod_wh TO ROLE dbt_prod_role;

GRANT ALL ON WAREHOUSE dbt_prod_heavy_wh TO ROLE dbt_prod_role;



CREATE OR REPLACE DATABASE dbt_dev;

CREATE OR REPLACE DATABASE dbt_prod;

GRANT ALL ON DATABASE dbt_dev TO ROLE dbt_dev_role;

GRANT ALL ON DATABASE dbt_prod TO ROLE dbt_prod_role;

GRANT ALL ON ALL SCHEMAS IN DATABASE dbt_dev TO ROLE dbt_dev_role;

GRANT ALL ON ALL SCHEMAS IN DATABASE dbt_prod TO ROLE dbt_prod_role;
```

2. Set up Python + DBT Project on your VM

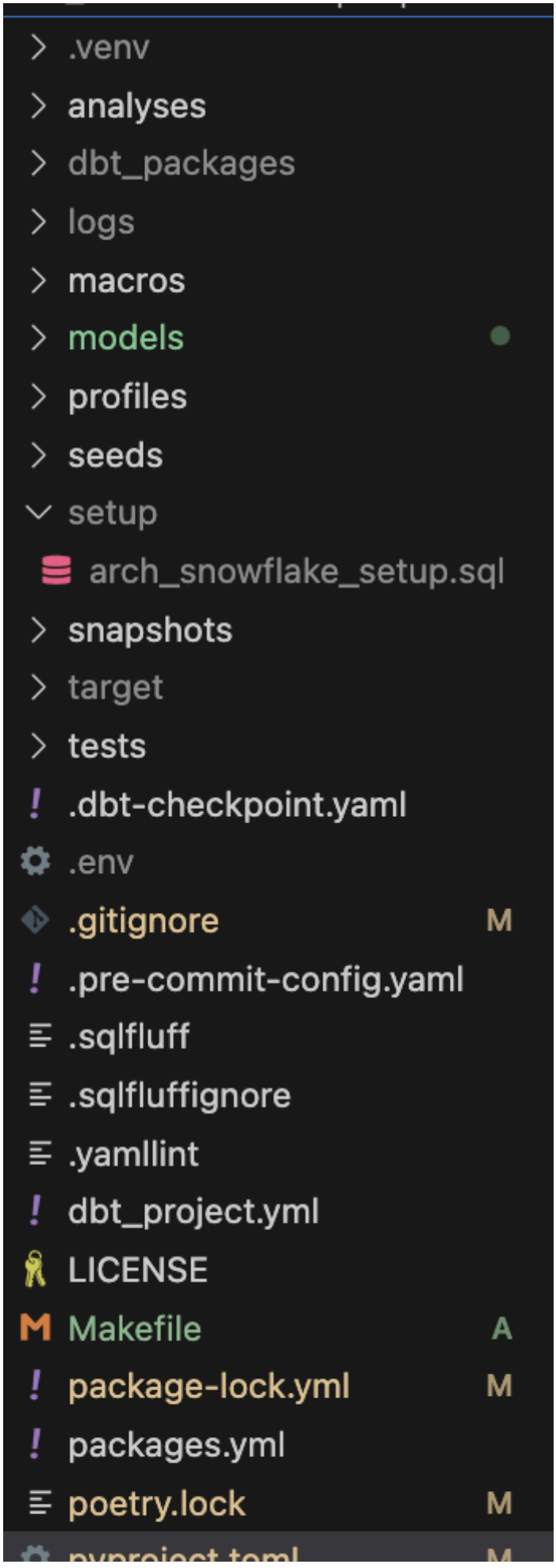
- Install `Python 3.10` on VM
- Open command prompt (use the search bar, type in `cmd` and press enter).
- Confirm Python is installed by typing in `python --version` and pressing enter. This should return the following `Python 3.10.0`
- Run `pip install poetry` ensuring this is using Python 3.10 on the VM and no other versions which can be an issue
- Run `poetry config virtualenvs.in-project true` to set the default location of poetry virtual environments to store in your local project directory in a `.venv` folder
- In the VM - create a directory/folder called `dbt_demo` for your dbt project in the following location `C:\Users\<usename>\`
- In your VM, open command prompt and move to this directory/folder you have created by running the following command: `cd C:\Users\<username>\dbt_demo`
- Set up the `poetry environment`:
  - Run `poetry init`
  - When instructed, press enter to skip the `Package name`, `Version`, `Description`, `Author`, `License` and `Compatible Python versions` settings

    - Enter `no` and press enter when asked `Would you like to define your main dependencies interactively? (yes/no)`
    - Enter `no` for `Would you like to define your development dependencies interactively?`
    - Enter `yes` for `Do you confirm generation? (yes/no)`
- Add the follow dependencies under `[tool.poetry.dependencies]`
    - dbt-core = "^1.7.0"
    - sqlfluff = "^2.3.5"
    - sqlfluff-templater-dbt = "^2.3.5"
    - pytest-sqlfluff = "^0.1.1"
    - dbt-snowflake = "^1.7.0"
    - pre-commit = "^3.5.0"
- Add the following configuration under `[tool.poetry]`
    - package-mode = false
- In the terminal:
    - Navigate to the repo in the terminal
    - Run `poetry install`
- Run `poetry run dbt init` and follow set up instructions
    - Select `password authentication`
    - For the account add the `<organisation identifier>-<account locator>` which can be taken from the account URL (`https://<organisation identifier>-<account locator>.snowflakecomputing.com/console/login`)
- Reorganise dbt project directory.
    - dbt has a limitation in how it sets up a new project. When you run `dbt init` to create a new project - it always creates the blank project in a new directory. There is no way to create the project in the current directory. This add unnecessary complication to the project directory and can cause issues with the virtual environment.
    - To fix this, move the contents of the created `dbttest` folder into the project repo root and delete the `dbttest` folder.
        - To do this in bash, run `mv dbttest/* .` & `rm -rf dbttest`
        - To do this in windows cmd, run `xcopy dbttest\* . /E /H /Y` & then `rmdir /s /q dbttest`
            - `/E` : Copies all subdirectories, including empty ones.
            - `/H` : Copies hidden and system files.
            - `/Y` : Suppresses prompting to overwrite files.
    - For further context - see a github issue here
- To test connection to Snowflake, run `poetry run dbt debug` - this should return `Connection Ok`. Ignore any git related errors.

Connect to VSCode and Git

1. Ensure VSCode in installed on VM, install this from here if missing.
2. Confirm you are able to log into Arch Insurance's GitHub Organisation github.com/orgs/archinsurance. Work with the platform team to resolve any access issues.

3. On your VM, access the GitHub Desktop Application. If this is missing, this can be installed from here
4. Once installed, open the application and select the option to log into github.com (not the github organisation option)
    1. This should open your browser with instructions for you to follow to log in.
5. Once logged in, follow the below steps to push your local `dbt_demo` project to Github.
    1. Select `File` > `Add local repository`
    2. Select the `dbt_demo` folder (`C:\Users\<username>\dbt_demo`)
    3. After selecting the project folder, you should see a warning message similar to the following: "This directory does not appear to be a Git Repository. Would you like to *create a repository* here instead?"
    4. Click **create a repository**
    5. In the following window, enter the following configurations:
        1. Name: `dbt_demo`
        2. In the Git Ingore dropdown, select `python`
        3. Then select `Create Repository`
6. Finally, connect the local dbt_demo project to VSCode.
    1. Select `File` > `Open Folder`
    2. In explorer window, navigate to your `dbt_demo` folder (`C:\Users\<username>\dbt_demo`)
    3. Click Open
    4. You should see something like the below appear in the VSCode Left Hand Side Navigation Panel


.venv
analyses
dbt_packages
logs
macros
models
profiles
seeds
setup
arch_snowflake_setup.sql
snapshots
target
tests
.dbt-checkpoint.yaml
.env
.gitignore
M
.pre-commit-config.yaml
.sqlfluff
.sqlfluffignore
.yamllint
dbt_project.yml
LICENSE
Makefile
A
package-lock.yml
M
packages.yml
poetry.lock
M

1.