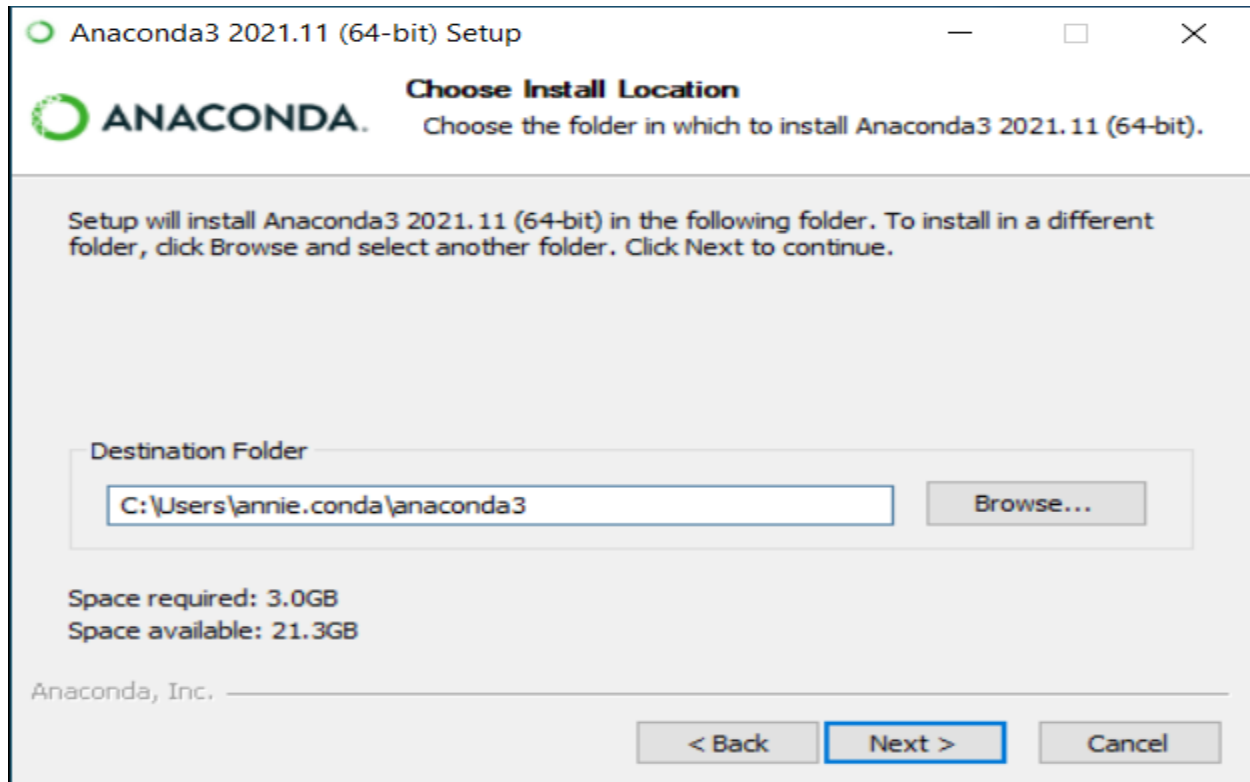


# Anaconda Installation

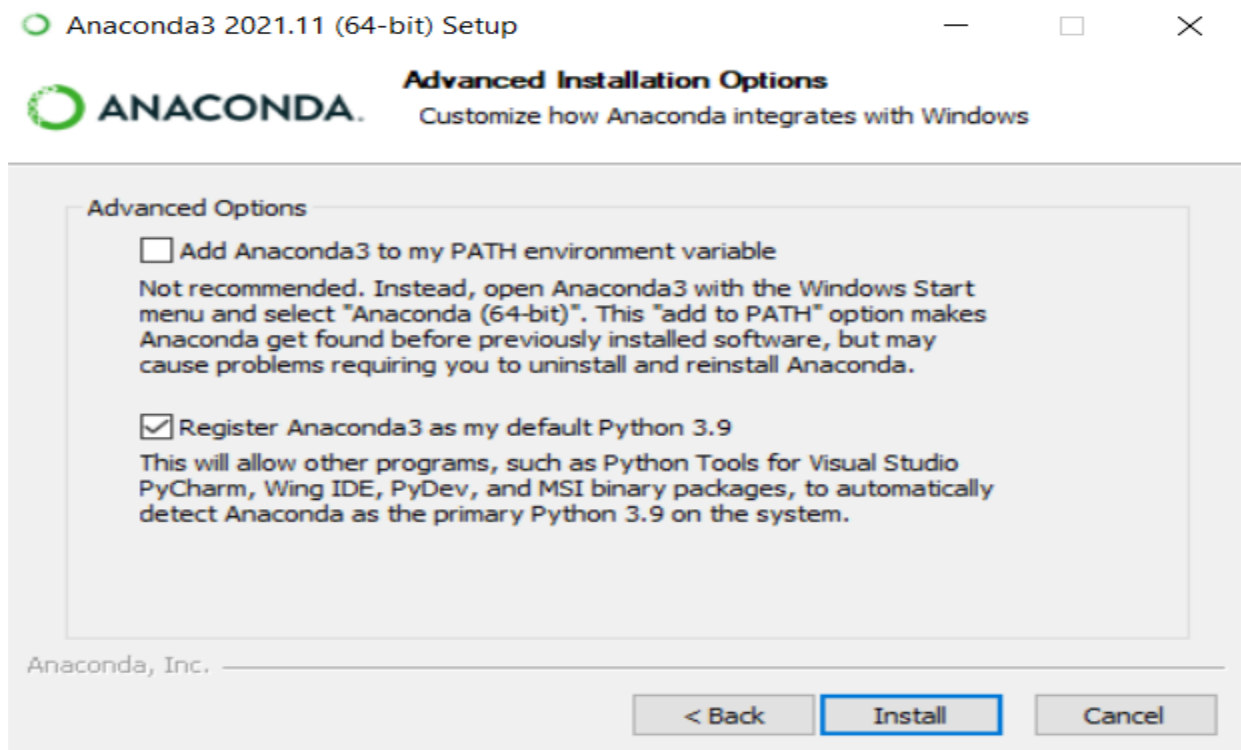
## IMPORTANT:

- If you are using a machine with an Apple M1 chip, follow [these instructions](#) to create the virtual environment and install Snowpark Python instead of what's described below.

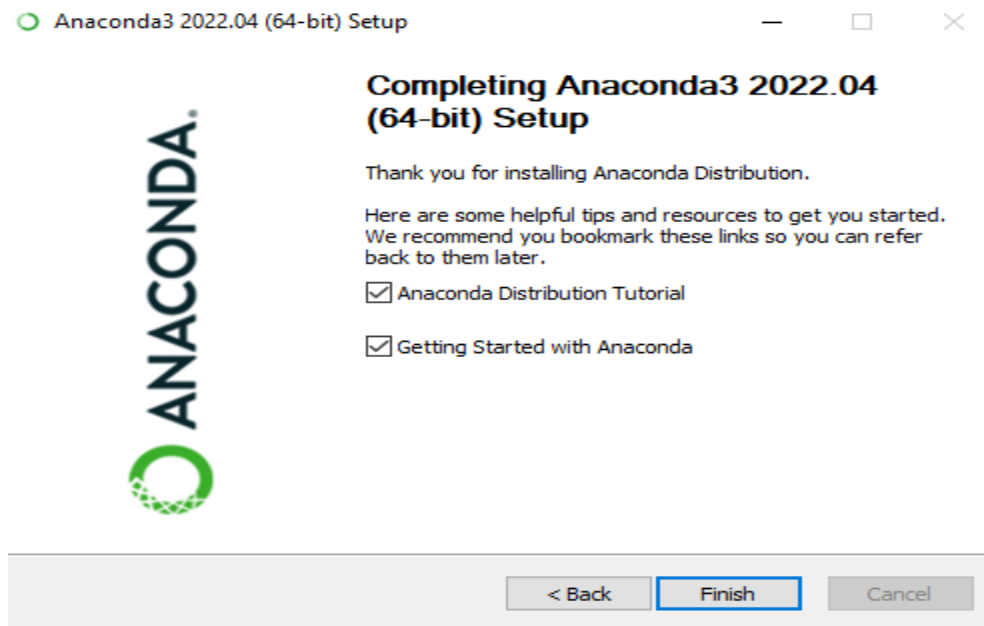
1. [Download the Anaconda installer](#).
2. Go to your Downloads folder and double-click the installer to launch. To prevent permission errors, do not launch the installer from the [Favorites folder](#).
3. Click **Next**.
4. Read the licensing terms and click **I Agree**.
5. It is recommended that you install for **Just Me**, which will install Anaconda Distribution to just the current user account. Only select an install for All Users if you need to install for all users' accounts on the computer (which requires Windows Administrator privileges).
6. Click **Next**.
7. Select a destination folder to install Anaconda and click **Next**. Install Anaconda to a directory path that does not contain spaces or unicode characters.



8. Click on **Install**.



9. Click **Install**. If you want to watch the packages Anaconda is installing, click Show Details.
10. Click **Next**.
11. After a successful installation you will see the “Thanks for installing Anaconda” dialog box:



12. If you wish to read more about Anaconda.org and how to get started with Anaconda, check the boxes “Anaconda Distribution Tutorial” and “Learn more about Anaconda”. Click the Finish button.

13. [Verify your installation](#).

## Snowpark Setup with Anaconda

Go to Anaconda prompt and create a conda env and name it as - “snowpark\_migration” for this quickstart:

```
conda env create -f conda_env.yml
conda activate snowpark_migration
!python kernel install --name “snowpark_migration”
```

Conda will automatically install `snowflake-snowpark-python` and all other dependencies for you.

Now, either you can launch Jupyter Notebook from Anaconda UI or by running following command on your local machine:

```
jupyter notebook
```

7. Initialize Notebook, import libraries and create Snowflake connection. For setting up the Snowflake connection from the local environment, download the snowpark setup folder structure from [snowpark\\_setup](#)

This folder contains two files as following:

### 1. **conda\_env.yaml**

The conda\_env.yml file refers to the environment we created and all the packages required to be installed to work with Snowpark

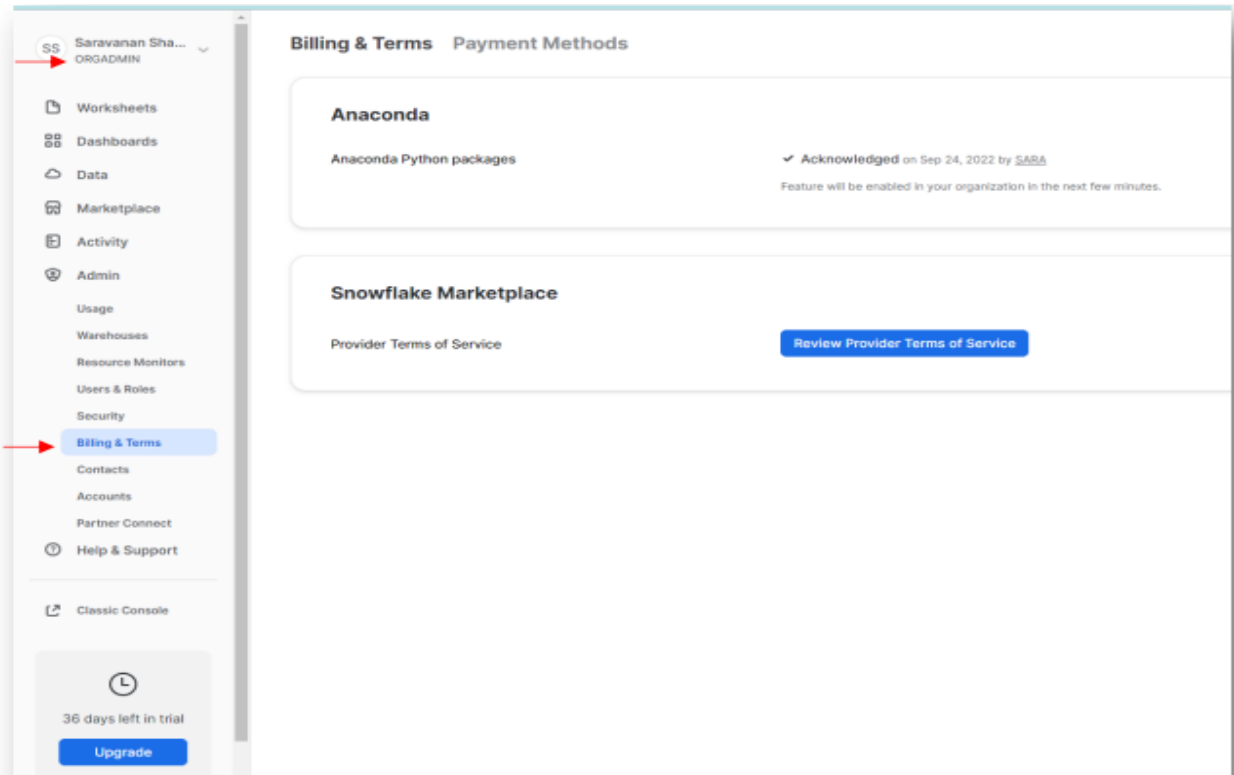
### 2. **config.py**

This file contains all the connection parameters required to connect with your snowflake account. Update the Snowflake credentials in config.py as shown below

```
SNOWFLAKE_CONN_PROFILE = {  
    "User": "<snowflake username>",  
    "password": "<snowflake password>",  
    "account": "<snowflake account identifier>",  
    "database": "<snowflake database>",  
    "schema": "<snowflake schema>",  
    "role": "<snowflake role>",  
    "warehouse": "<snowflake warehouse>",  
}
```

Note: The snowflake account identifier can be copied from the account url  
Example: [https://\\*\\*\\*\\*\\*.ap-south-1.aws.snowflakecomputing.com](https://*****.ap-south-1.aws.snowflakecomputing.com)

## **Specifying Dependencies for a UDF**



Following code block in jupyter notebook for snowpark setups Snowflake connection

[Code](#)

```
from snowflake.snowpark import Session
from snowflake.snowpark import functions as F

# Snowflake connection info is saved in config.py
from config import SNOWFLAKE_CONN_PROFILE
session = Session.builder.configs(SNOWFLAKE_CONN_PROFILE).create()
session.sql("use role Accountadmin").collect()
session.sql("create database if not exists
{}".format(SNOWFLAKE_CONN_PROFILE['database'])).collect()
session.sql("use database {}".format(SNOWFLAKE_CONN_PROFILE['database'])).collect()
session.sql("use schema {}".format(SNOWFLAKE_CONN_PROFILE['schema'])).collect()
session.sql("use warehouse {}".format(SNOWFLAKE_CONN_PROFILE['warehouse']))
print(session.sql('select current_warehouse(), current_database(),
current_schema()').collect())
```