# VS code integration with GCP on Tef VDI

**Summary:**

This document will help you to install and set up the required tools needed to perform SQL operations on GCP from your TEF VDI. Below are the list of tools to be installed on your VDI.

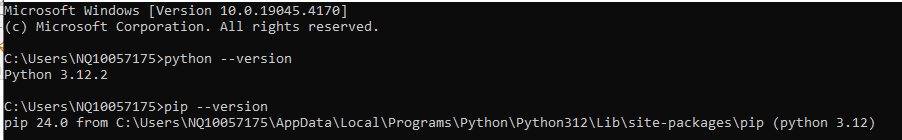
* Visual Studio Code
* Python
* Git
* Google cloud SDK Shell
* dbt-core and dbt-bigquery

**Installation and setup guidelines:**

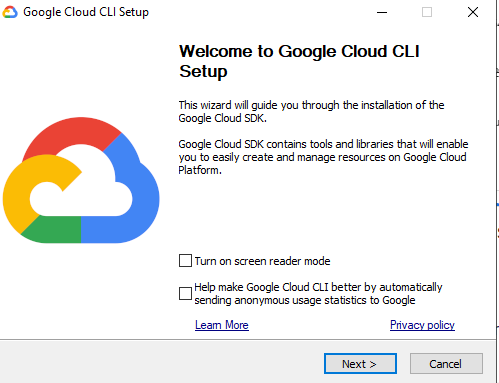
* **Visual studio code:**
  + Download Visual Studio code from official website of visual studio ie from the link below. Once exe file is download go with all default settings to install it on your tef vdi.
  + <https://code.visualstudio.com/sha/download?build=stable&os=win32-x64-user>
* **Python:**
  + Download exe file from <https://www.python.org/ftp/python/3.12.2/python-3.12.2-amd64.exe> . Double click on the exe file to install.
  + Select **Add python.exe to Path**  and click on Install Now.

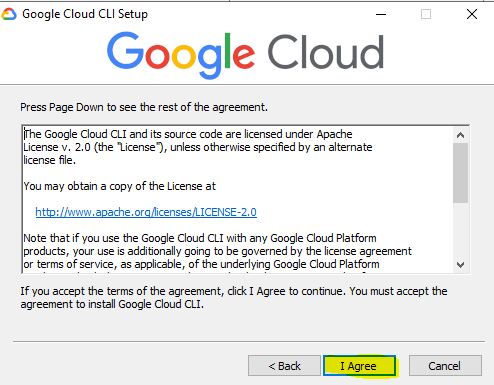


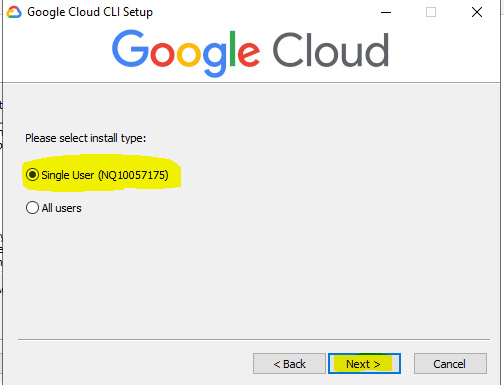
* + Click on Close after setup is successful.
  + Open Terminal and check for python and pip version.

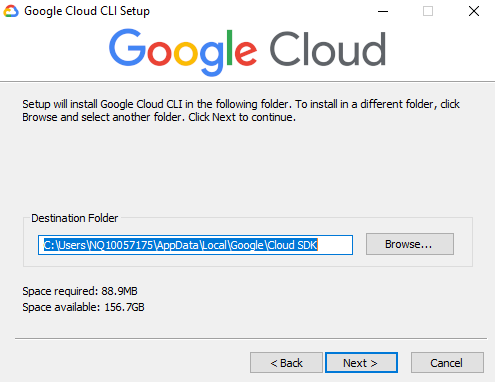


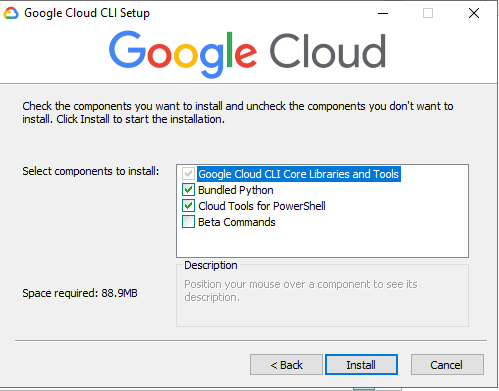
* **Git**
  + Download Git exe file from <https://github.com/git-for-windows/git/releases/download/v2.44.0.windows.1/Git-2.44.0-64-bit.exe> . Once Downloaded double click on the exe file and go with all default settings to install it.
* **Google cloud SDK**
  + Download Git exe file from <https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe> . Once Downloaded double click on the exe file and go with all default settings to install it.
  + Follow below steps to install







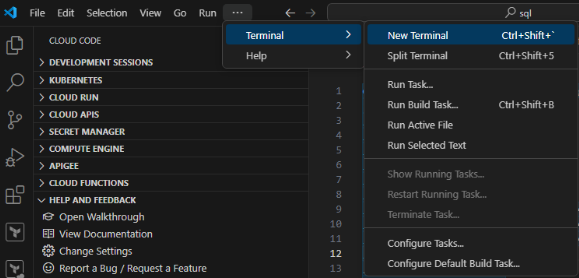




* **dbt-core and dbt-bigquery**
  + Open Gitbash terminal and run below commands,
    - export http\_proxy=http://10.169.127.8:8080 && export https\_proxy=http://10.169.127.8:8080
    - pip install dbt-core
    - pip install dbt-bigquery.

## Configure DBT on VDI

Open VS code on your VDI and open terminal. Follow below steps to configure DBT.



* Check python version .

**Ex. python –version**

* Create user virtual environment.

Python –m venv **<env name>**

Ex: python -m venv dbt\_venv\_anudeep

* + **Step2.1**: Go inside newly created directory.

Ex: cd dbt\_venv\_anudeep

* Check DBT version.

Ex: dbt --version

* initialise DBT.

Ex: dbt init

* Give <project name>

Eg -dbt\_fastoss

* Type 1 (as we have to select option 1 Bigquery)
* Type 1 (as we have to select option 1 Oauth)
* Enter GCP project ID

Ex: tefde-gcp-fastoss-dev-gke

* Enter you Dataset name created in GCP bigquery

Ex: dbt\_sql

* Enter threads as 4.
* Go with default for Job\_execution\_timeout-seconds[300].
* In location details, Type 2 (as we have to select option 2 EU). You are now set to work on DBT.
* Authenticate to GCP.
  + In VS code terminal, Run command “gcloud auth login”, which opens up GCP authentication page in your browser.
  + Click on your Tef GCP account id, then **Continue** and then **Allow.** You are now authenticated to GCP account, return back to VS code, you will see authentication successful message in terminal.
* Run “gcloud config set project <gcp project ID>”

Ex: gcloud config set project tefde-gcp-fastoss-dev-gke

* Run “gcloud auth application-default login” command and authenticate to GCP account.
* Run “gcloud auth application-default print-access-token”, if this prints a token then your authentication is successful.
* Go to your Project directory, and run your DBT operations.

Ex: cd dbt\_fastoss

## Troubleshooting steps:

* If it failed to run dbt operations or failed to locate profiles.yml file then follow next steps.
  + Open profiles.yml file under .dbt folder,

## Git Integration:

This section guides you on how to move the changes from local VDI machine to gitlab remote repo through VS code.

* Open Terminal or Git bash on your VDI and run **Git clone <gitlab\_repo\_link>** to clone a remote repo to VDI**.** If prompted login to gitlab with your credentials.

Ex: git clone <https://dot-portal.de.pri.o2.com/gitlab/fastoss_b/dbt.git>

* Open VS code and click on “open folder” under file at top left corner and select the repo folder that clone in above step..
* Open git bash terminal in VS code,
  + Click on Terminal → New Terminal from top Menu bar
  + Inside the Terminal opened, click on powershell dropdown & choose Git Bash
* Checkout to develop branch - run git checkout develop
* To create feature branch - run git checkout -b [type of work]/[domain]-[Summary in present tense]
  + <https://gist.github.com/joshbuchea/6f47e86d2510bce28f8e7f42ae84c716> has detailed information for branch name format
  + Example : git checkout -b feat/bigquery-dbt, which will create feat/bigquery-dbt branch.
* Add/update changes in local branch created above
* To add newly added/updated changes to staging area of your local repo - run command in Git Bash terminal “git add .”
* Commit changes to branch
  + syntax : git commit -m " [type of work](domain): Summary in present tense"
  + example : git commit -m "feat(bigquery-dbt): add ddl sql statements for creating schema in bigquery"
* Push changes
  + syntax : git push --set-upstream origin branch\_name
  + example : git push --set-upstream origin feat/bigquery-dbt.

Once changes are pushed to the remote repository, Merge request can be raised to move the changes to upper branches.

