DVC 실습

---

1. DVC 설치

- **python** 설치
    - python 3.8 이상의 환경을 준비합니다.
        - (python 설치 방법은 별도로 첨부된 강의자료를 확인하세요.)

```
python -V
# Python 3.9.6
```

- **git** 설치
    - git 을 다운받습니다.

```
#sudo apt install git

git --version
# git version 2.25.1
```

```
git --help
# 정상 설치되었는지 확인
```

- **dvc** 설치
  - dvc 2.6.4 버전을 다운 받습니다.
  - `dvc[all]` 에서 `[all]` 은 dvc 의 remote storage 로 s3, gs, azure, oss, ssh 모두를 사용할 수 있도록 관련 패키지를 함께 설치하는 옵션입니다.

```
sudo pip install dvc[all]==2.6.4

dvc --version
# 2.6.4

dvc --help
# 정상 설치되었는지 확인
```

---

2. DVC 저장소 세팅

1) 새 Directory 를 생성합니다.

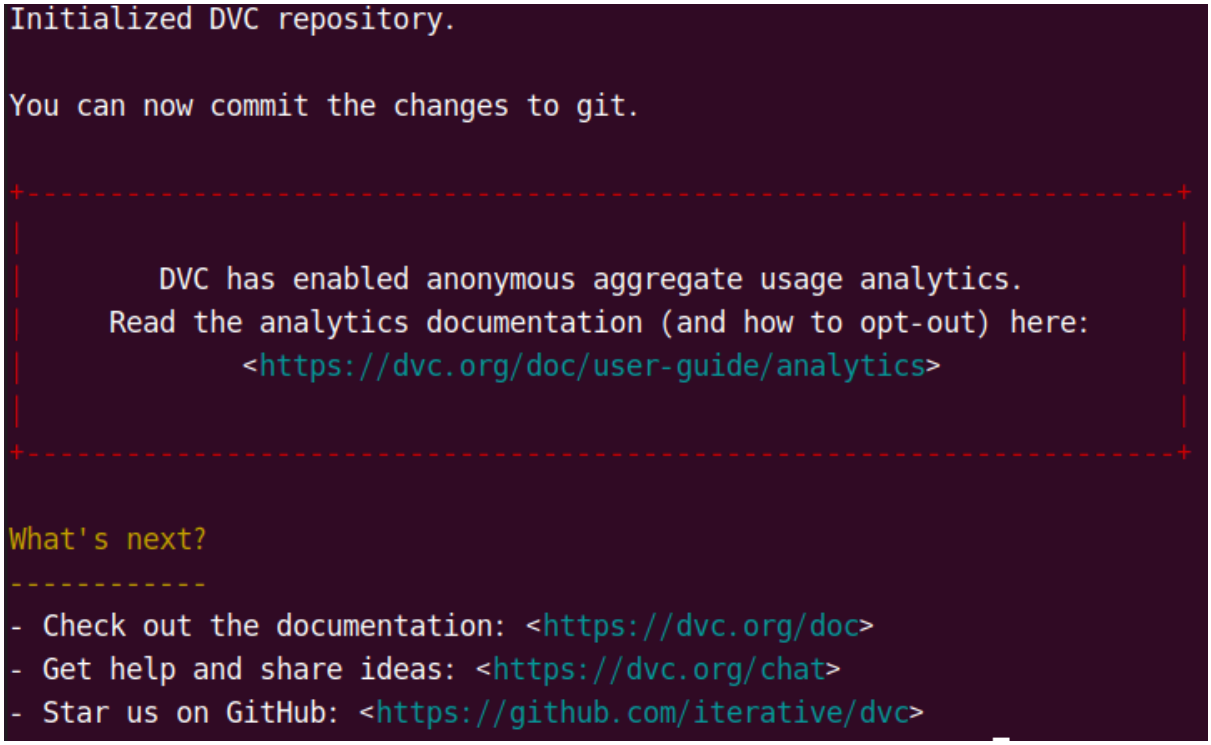
```
# STEP 1) 새로운 directory 를 만들고 이동합니다.
mkdir dvc-tutorial

cd dvc-tutorial
```

2) 해당 Directory 를 git 저장소로 초기화합니다.

```
# STEP 2) git 저장소로 초기화합니다.
git init
```

3) 해당 Directory 를 dvc 저장소로 초기화합니다.

```
# STEP 3) dvc 저장소로 초기화합니다.
dvc init
```



3. DVC 기본 명령 1

1) dvc 로 버전 tracking 할 data 를 생성합니다.

```
# data 를 저장할 용도로 data 라는 이름의 디렉토리를 생성하고 이동합니다.
mkdir data

cd data

# 가볍게 변경할 수 있는 데이터를 카피해오거나, 새로 만듭니다.
vi demo.txt

cat demo.txt
# Hello!
```

2) 방금 생성한 데이터를 dvc 로 tracking 합니다.

```
cd ..

dvc add data/demo.txt

# To track the changes with git, run:
git add data/demo.txt.dvc data/.gitignore
```

3) dvc add 에 의해 자동 생성된 파일들을 확인합니다.

```
cd data
ls
# demo.txt.dvc 파일이 자동 생성된 것을 확인

cat demo.txt.dvc
# demo.txt 파일의 메타정보를 가진 파일입니다.
# git 에서는 demo.txt 파일이 아닌, demo.txt.dvc 파일만 관리하게 됩니다.
```

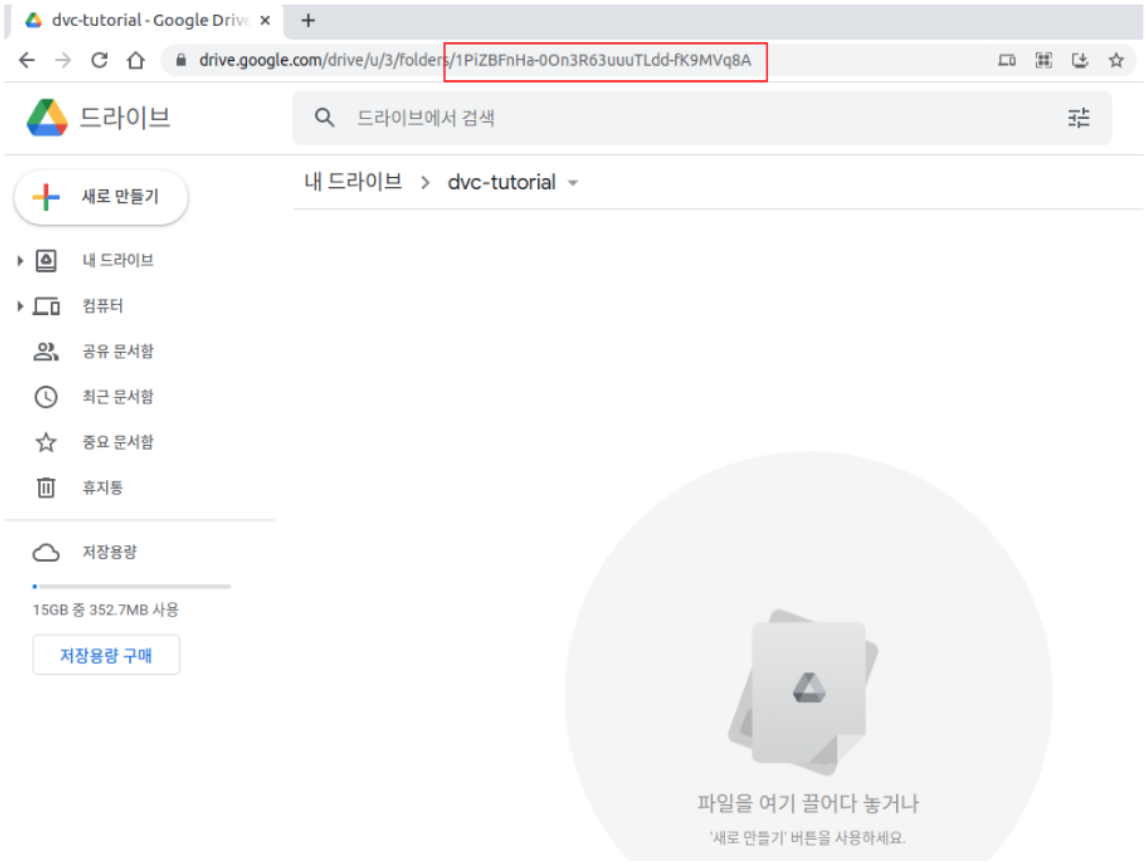
4) git commit 을 수행합니다.

```
git commit -m "Add demo.txt.dvc"

#git remote add origin <git주소>
#git push origin master
```

- (`.dvc` 파일은 `git push` 를 수행하여, git repository 에 저장합니다.)

5) data 가 실제로 저장될 remote storage 를 세팅합니다.

- 본인의 google drive 에 새로운 폴더를 하나 생성해준 뒤, url 로부터 ID 를 복사합니다.
    - 아래 스크린샷의 빨간 네모박스에 해당하는 부분을 복사합니다.




```
dvc remote add -d storage gdrive://<GOOGLE_DRIVE_FOLDER_ID>
# dvc 의 default remote storage 로 gdrive://<GOOGLE_DRIVE_FOLDER_ID> 를 세팅합니다.
```

6) dvc config 를 git commit 합니다.

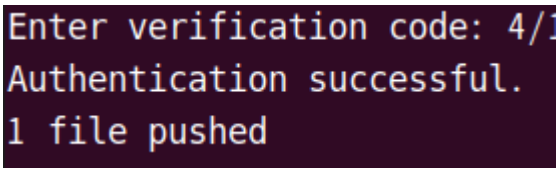
```
git add .dvc/config
git commit -m "add remote storage"
#git push origin master
```

7) dvc push

- 데이터를 remote storage 에 업로드합니다.

```
dvc push

#Go to the following link in your browser:
#
#    https://accounts.google.com/o/oauth2/.........
#
# Enter verification code:
```

- dvc push 를 수행하기 위해서는 인증 과정이 필요합니다.
  - 위의 주소로 이동하여, google login 을 통해 인증을 수행해주시기 바랍니다.
  - 인증이 완료되면 다음과 같은 화면이 나타납니다.



- 구글 드라이브로 이동하여 파일이 정상적으로 업로드되었는지 확인합니다.
  - 새로운 폴더가 하나 생성되었고, 폴더 내부에 긴 이름의 파일이 하나 업로드된 것을 확인하실 수 있습니다.
  - 이 파일을 다운로드받은 뒤 열어보면 업로드한 파일과 동일한 파일임을 확인할 수 있습니다.

---

4. DVC 기본 명령 2

1) dvc pull

- 데이터를 remote storage 로부터 다운로드합니다.

```
cd dvc-tutorial

# dvc 캐시를 삭제합니다.
rm -rf .dvc/cache/
# dvc push 했던 데이터를 삭제합니다.
rm -rf data/demo.txt

# dvc pull 로 google drive 에 업로드했던 데이터를 다운받습니다.
dvc pull

# 방금 다시 다운받은 데이터가 이전 데이터와 동일한지 확인합니다.
cat data/demo.txt
```

2) dvc checkout

- data 의 버전 변경하는 명령어입니다.
- 버전 변경 테스트를 위해, 새로운 버전의 data 를 dvc push 합니다.

```
# 데이터를 변경합니다. (새로운 데이터를 같은 이름으로 copy 해와도 좋습니다.)
vi data/demo.txt

# 변경되었는지 확인합니다.
cat data/demo.txt

# dvc add (data/demo.txt.dvc 를 변경시켜주는 역할)
dvc add data/demo.txt

# git add and commit
git add data/demo.txt.dvc
git commit -m "update demo.txt"

# dvc push (and git push)
dvc push # 새로운 버전의 data 파일을 remote storage 에 업로드

(git push) # .dvc 파일을 git repository 에 업로드
```

- 구글 드라이브로 이동하여 new 파일이 정상적으로 업로드되었는지 확인합니다.
  - 새로운 폴더가 추가로 생성되었고, 폴더 내부에 긴 이름의 파일이 하나 업로드된 것을 확인하실 수 있습니다.
  - 이 파일을 다운로드받은 뒤 열어보면 방금 변경한 파일이 업로드 된 것을 확인할 수 있습니다.
- 이전 버전의 data 로 되돌아가보겠습니다.

```
# git log 를 확인합니다.
git log --oneline

# demo.txt.dvc 파일을 이전 commit 버전으로 되돌립니다.
git checkout <COMMIT_HASH> data/demo.txt.dvc

# dvc checkout 합니다. (demo.txt.dvc 의 내용을 보고 demo.txt 파일을 이전 버전으로 변경합니다.)
dvc checkout

# 데이터가 변경되었는지 확인합니다.
cat data/demo.txt
```

---

5. DVC 의 추가 기능

- 이번 강의에서 다루지 않은 DVC 의 추가 기능
  - Python API 를 사용한 제어
    - https://dvc.org/doc/api-reference

- S3, HDFS, SSH 등의 remote storage 연동
- DAG 를 통한 Data pipeline 관리
  - https://dvc.org/doc/start/data-pipelines
- `dvc metrics`, `dvc plots` 를 사용한 각 실험의 metrics 기록 및 시각화