IITP DABT DataCollector

배포 실행 가이드

문서 버전: 1.0.0

작성일: 2025-11-27

(주)스위트케이

문서 History

버전	일자	작성자	변경 내용
1.0.0	2025-11-27	(주)스위트케이	최초 작성

목차

1. 개요

1.1. 문서 목적

본 문서는 IITP DABT DataCollector를 Ubuntu 22.04 서버 환경에 설치하고 실행하는 방법을 상세히 안내합니다.

본 가이드를 따라 설치 및 설정을 완료하면, CSV 파일에 포함된 이미지 URL을 기반으로 대량의 이미지를 자동으로 다운로드할 수 있습니다.

1.2. 적용 범위

본 문서는 다음 환경을 기준으로 작성되었습니다:

- **운영체제**: Ubuntu 22.04 LTS
- **Python 버전**: 3.8 이상
- **실행 계정**: iitp-app
- **기본 설치 경로**: `/home/iitp-app/IITP-DABT-DataCollector`

2. 사전 요구사항

2.1. 시스템 요구사항

다음 사항을 확인하세요:

2.1.1. 운영체제 확인

```
lsb_release -a
```

예상 출력:

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
```

2.1.2. Python 버전 확인

```
python3 --version
```

요구사항: Python 3.8 이상

예상 출력:

```
Python 3.10.12
```

Python이 설치되어 있지 않은 경우:

```
sudo apt update
sudo apt install python3 python3-venv python3-pip -y
```

2.1.3. 네트워크 연결 확인

외부 이미지 URL에 접근할 수 있어야 합니다:

```
ping -c 3 8.8.8.8
curl -I https://www.google.com
```

2.2. 사용자 계정

2.2.1. iitp-app 계정 확인

```
id iitp-app
```

예상 출력:

```
uid=1001(iitp-app) gid=1001(iitp-app) groups=1001(iitp-app)
```

계정이 없는 경우 생성:

```
sudo useradd -m -s /bin/bash iitp-app
sudo passwd iitp-app
```

2.2.2. 홈 디렉토리 확인

```
echo $HOME
```

예상 출력:

```
/home/iitp-app
```

3. 설치 절차

3.1. 디렉토리 구조 생성

iitp-app 계정으로 로그인한 후 다음 디렉토리 구조를 생성합니다:

```
# iitp-app 계정으로 로그인
su - iitp-app

# 기본 디렉토리 생성
mkdir -p /home/iitp-app/IITP-DABT-DataCollector
cd /home/iitp-app/IITP-DABT-DataCollector

# 하위 디렉토리 생성
mkdir -p data
mkdir -p downloads
mkdir -p logs
```

생성되는 디렉토리 구조:

```
/home/iitp-app/IITP-DABT-DataCollector/
├── data/          # CSV 파일 저장 디렉토리
├── downloads/     # 다운로드된 이미지 저장 디렉토리
└── logs/          # 로그 파일 저장 디렉토리
```

3.2. 소스 코드 배포

소스 코드를 `/home/iitp-app/IITP-DABT-DataCollector` 디렉토리에 배포합니다.

필수 파일:

- `downloader.py`
- `analyze_errors.py`
- `requirements.txt`
- `env.sample`

배포 방법은 Git 클론, 압축 파일 해제, 직접 복사 등 적절한 방법을 사용하세요.

권한 확인:

```
ls -la /home/iitp-app/IITP-DABT-DataCollector/
```

모든 파일의 소유자가 `iitp-app` 인지 확인하세요.

3.3. Python 가상환경 설정

Python 가상환경을 생성하고 활성화합니다:

```
cd /home/iitp-app/IITP-DABT-DataCollector

# 가상환경 생성
python3 -m venv venv

# 가상환경 활성화
source venv/bin/activate
```

활성화 확인:
프롬프트 앞에 `(venv)` 가 표시되어야 합니다:

```
(venv) iitp-app@server:~/IITP-DABT-DataCollector$
```

가상환경 비활성화:

```
deactivate
```

3.4. 의존성 패키지 설치

가상환경이 활성화된 상태에서 의존성 패키지를 설치합니다:

```
# pip 업그레이드 (선택사항)
pip install --upgrade pip

# 의존성 패키지 설치
pip install -r requirements.txt
```

설치되는 패키지:

- requests==2.32.3
- python-dotenv==1.0.1

설치 확인:

```
pip list
```

예상 출력:

```
Package       Version
------------- -------
pip           23.x.x
python-dotenv 1.0.1
requests      2.32.3
setuptools    x.x.x
```

4. 환경 설정

4.1. .env 파일 생성

`env.sample` 파일을 복사하여 `.env` 파일을 생성합니다:

```
cd /home/iitp-app/IITP-DABT-DataCollector
cp env.sample .env
```

4.2. 환경 변수 설정

`.env` 파일을 편집하여 환경 변수를 설정합니다:

```
nano .env
```

또는

```
vi .env
```

4.2.1. 필수 환경 변수

URL_CSV_PATH

처리할 CSV 파일의 전체 경로를 설정합니다:

```
URL_CSV_PATH=/home/iitp-app/IITP-DABT-DataCollector/data/images.csv
```

주의사항:

- 절대 경로를 사용하세요
- 파일 확장자는 반드시 `.csv` 여야 합니다
- 파일이 실제로 존재하는지 확인하세요

파일 존재 확인:

```
ls -l /home/iitp-app/IITP-DABT-DataCollector/data/images.csv
```

4.2.2. 선택 환경 변수

LOG_LEVEL

로깅 레벨을 설정합니다:

```
LOG_LEVEL=INFO
```

가능한 값: `DEBUG` , `INFO` , `WARNING` , `ERROR` , `CRITICAL`

ROOT_DIR

다운로드한 이미지가 저장될 루트 디렉토리를 설정합니다:

```
ROOT_DIR=/home/iitp-app/IITP-DABT-DataCollector/downloads
```

기본값: 현재 작업 디렉토리의 `downloads` 폴더

주의사항:

- 절대 경로를 권장합니다
- 디렉토리가 존재하지 않으면 자동으로 생성됩니다
- 쓰기 권한이 있어야 합니다

THREADS

병렬 다운로드에 사용할 스레드 수를 설정합니다:

```
THREADS=8
```

권장값:

- 네트워크 대역폭이 충분한 경우: 8~16
- 서버 부하를 고려해야 하는 경우: 4~8
- 매우 느린 네트워크: 2~4

HEAD_CHECK

다운로드 전 HEAD 요청으로 이미지 타입을 사전 검증할지 여부:

```
HEAD_CHECK=false
```

가능한 값: `true` , `false` , `1` , `0` , `yes` , `no` , `y` , `n` , `on` , `off`

VERIFY_SSL

HTTPS 요청 시 SSL 인증서 검증 여부:

```
VERIFY_SSL=true
```

주의사항:

- 운영 환경에서는 반드시 `true` 로 설정 권장
- 자체 서명 인증서를 사용하는 내부 서버의 경우에만 `false` 고려

REQUEST_HEADERS_JSON

추가 HTTP 헤더를 JSON 형식으로 설정:

```
REQUEST_HEADERS_JSON={"Referer":"https://example.com"}
```

또는 여러 헤더:

```
REQUEST_HEADERS_JSON={"Referer":"https://example.com","User-Agent":"Mozilla/5.0"}
```

주의사항:

- JSON 형식이 올바른지 확인하세요
- 따옴표 이스케이프가 필요한 경우 주의하세요

4.2.3. .env 파일 예시

전체 `.env` 파일 예시:

```
# Logging level: DEBUG, INFO, WARNING, ERROR, CRITICAL
LOG_LEVEL=INFO

# Root directory where date-based folder (YYYY-MM-DD) will be created to store images
ROOT_DIR=/home/iitp-app/IITP-DABT-DataCollector/downloads

# Number of parallel download threads
THREADS=8

# Full path (including filename) to the CSV file with columns: No,Type,Title,Img-link
URL_CSV_PATH=/home/iitp-app/IITP-DABT-DataCollector/data/images.csv

# Optional: perform HTTP HEAD to check Content-Type is image/* before GET (true/false)
HEAD_CHECK=false

# Optional: verify SSL certificates on HTTPS requests (true/false). Set false only if necessary
VERIFY_SSL=true

# Optional: extra request headers in JSON (e.g., Referer or custom User-Agent)
# Example: {"Referer":"https://example.com","User-Agent":"Mozilla/5.0"}
REQUEST_HEADERS_JSON=
```

4.2.4. 환경 변수 검증

설정한 환경 변수가 올바른지 확인합니다:

```
# 가상환경 활성화
source venv/bin/activate

# Python으로 환경 변수 확인 (테스트)
python3 -c "
import os
from dotenv import load_dotenv
load_dotenv()
print('URL_CSV_PATH:', os.getenv('URL_CSV_PATH'))
print('ROOT_DIR:', os.getenv('ROOT_DIR'))
print('THREADS:', os.getenv('THREADS'))
"
```

5. 실행 방법

5.1. 수동 실행

5.1.1. 기본 실행

가상환경을 활성화한 후 프로그램을 실행합니다:

```
# iitp-app 계정으로 로그인
su - iitp-app

# 작업 디렉토리로 이동
cd /home/iitp-app/IITP-DABT-DataCollector

# 가상환경 활성화
source venv/bin/activate

# 프로그램 실행
python3 downloader.py
```

5.1.2. 실행 예시

정상 실행 시 출력 예시:

```
2025-10-23 14:30:15,123 [INFO] CSV: /home/iitp-app/IITP-DABT-DataCollector/data/images.csv
2025-10-23 14:30:15,123 [INFO] Output dir: /home/iitp-app/IITP-DABT-DataCollector/downloads/202
2025-10-23 14:30:15,123 [INFO] Log file: /home/iitp-app/IITP-DABT-DataCollector/logs/image_down
2025-10-23 14:30:15,123 [INFO] Threads: 8
2025-10-23 14:30:15,123 [INFO] HEAD_CHECK: false
2025-10-23 14:30:15,123 [INFO] VERIFY_SSL: true
2025-10-23 14:30:16,456 [INFO] Saved: /home/iitp-app/IITP-DABT-DataCollector/downloads/2025-10-
...
2025-10-23 14:30:20,789 [INFO] Total links: 100
2025-10-23 14:30:20,789 [INFO] Succeeded: 95
2025-10-23 14:30:20,789 [INFO] Failed: 5
2025-10-23 14:30:20,789 [INFO] Error rows file: /home/iitp-app/IITP-DABT-DataCollector/data/imag
Total links: 100
Succeeded: 95
Failed: 5
Error rows file: /home/iitp-app/IITP-DABT-DataCollector/data/images_errorRow.csv
```

5.1.3. 실행 스크립트 생성 (선택사항)

반복 실행을 위해 실행 스크립트를 생성할 수 있습니다:

```bash
cat > /home/iitp-app/IITP-DABT-DataCollector/run.sh << 'EOF'
#!/bin/bash
cd /home/iitp-app/IITP-DABT-DataCollector
source venv/bin/activate
python3 downloader.py
deactivate
EOF

chmod +x /home/iitp-app/IITP-DABT-DataCollector/run.sh
```

실행:

```
/home/iitp-app/IITP-DABT-DataCollector/run.sh
```

5.2. Cron을 이용한 자동 실행

Cron을 사용하여 주기적으로 자동 실행하도록 설정할 수 있습니다.

5.2.1. Cron 설정

iitp-app 계정의 crontab을 편집합니다:

```
# iitp-app 계정으로 로그인
su - iitp-app

# crontab 편집
crontab -e
```

5.2.2. Cron 실행 스크립트 생성

Cron에서 사용할 실행 스크립트를 생성합니다:

```
cat > /home/iitp-app/IITP-DABT-DataCollector/run_cron.sh << 'EOF'
#!/bin/bash
# 환경 변수 설정
export PATH=/usr/local/bin:/usr/bin:/bin
cd /home/iitp-app/IITP-DABT-DataCollector

# 가상환경 활성화 및 실행
source venv/bin/activate
python3 downloader.py
deactivate
EOF

chmod +x /home/iitp-app/IITP-DABT-DataCollector/run_cron.sh
```

Cron에 등록:

```
# 매일 오전 2시 실행
0 2 * * * /home/iitp-app/IITP-DABT-DataCollector/run_cron.sh >> /home/iitp-app/IITP-DABT-DataCo
```

5.2.3. Cron 설정 확인

```
# 현재 등록된 cron 작업 확인
crontab -l

# Cron 서비스 상태 확인
sudo systemctl status cron
```

5.2.4. Cron 로그 확인

```
# Cron 실행 로그 확인
tail -f /home/iitp-app/IITP-DABT-DataCollector/logs/cron.log

# 시스템 Cron 로그 확인
sudo tail -f /var/log/syslog | grep CRON
```

6. 실행 확인 및 검증

6.1. 실행 결과 확인

프로그램 실행 후 콘솔 출력에서 다음 정보를 확인합니다:

- **Total links**: 처리된 총 URL 개수
- **Succeeded**: 성공한 다운로드 개수
- **Failed**: 실패한 다운로드 개수
- **Error rows file**: 에러 리포트 파일 경로 (실패한 경우)

정상 실행 예시:

```
Total links: 100
Succeeded: 95
Failed: 5
Error rows file: /home/iitp-app/IITP-DABT-DataCollector/data/images_errorRow.csv
```

6.2. 로그 확인

6.2.1. 로그 파일 위치

로그 파일은 다음 위치에 저장됩니다:

```
/home/iitp-app/IITP-DABT-DataCollector/logs/image_downloader_YYYY-MM-DD.log
```

6.2.2. 로그 확인 방법

```
# 최신 로그 파일 확인
ls -lt /home/iitp-app/IITP-DABT-DataCollector/logs/ | head -5

# 로그 파일 내용 확인
tail -100 /home/iitp-app/IITP-DABT-DataCollector/logs/image_downloader_2025-10-23.log

# 실시간 로그 모니터링
tail -f /home/iitp-app/IITP-DABT-DataCollector/logs/image_downloader_2025-10-23.log
```

6.2.3. 로그 레벨별 확인

INFO 레벨 로그 확인:

```
grep "\[INFO\]" /home/iitp-app/IITP-DABT-DataCollector/logs/image_downloader_2025-10-23.log
```

ERROR 레벨 로그 확인:

```
grep "\[ERROR\]" /home/iitp-app/IITP-DABT-DataCollector/logs/image_downloader_2025-10-23.log
```

6.3. 다운로드 파일 확인

6.3.1. 다운로드 디렉토리 확인

```
# 날짜별 폴더 확인
ls -la /home/iitp-app/IITP-DABT-DataCollector/downloads/

# 특정 날짜의 다운로드 파일 확인
ls -lh /home/iitp-app/IITP-DABT-DataCollector/downloads/2025-10-23/ | head -20

# 다운로드된 파일 개수 확인
find /home/iitp-app/IITP-DABT-DataCollector/downloads/2025-10-23/ -type f | wc -l
```

6.3.2. 파일 크기 확인

```
# 전체 다운로드 용량 확인
du -sh /home/iitp-app/IITP-DABT-DataCollector/downloads/2025-10-23/

# 파일별 크기 확인
ls -lhS /home/iitp-app/IITP-DABT-DataCollector/downloads/2025-10-23/ | head -10
```

6.3.3. 에러 리포트 확인

다운로드 실패가 있는 경우 에러 리포트를 확인합니다:

```
# 에러 리포트 파일 확인
ls -lh /home/iitp-app/IITP-DABT-DataCollector/data/*_errorRow.csv

# 에러 리포트 내용 확인
head -20 /home/iitp-app/IITP-DABT-DataCollector/data/images_errorRow.csv
```

에러 분석 유틸리티 실행:

```
cd /home/iitp-app/IITP-DABT-DataCollector
source venv/bin/activate
python3 analyze_errors.py /home/iitp-app/IITP-DABT-DataCollector/data/images_errorRow.csv
deactivate
```

7. 문제 해결

7.1. 일반적인 오류 및 해결 방법

7.1.1. 환경 변수 오류

오류 메시지:

```
Missing required env var: URL_CSV_PATH
```

해결 방법:

1. `.env` 파일이 존재하는지 확인:

   ```
   ls -l /home/iitp-app/IITP-DABT-DataCollector/.env
   ```

2. `.env` 파일의 `URL_CSV_PATH` 값 확인:

   ```
   grep URL_CSV_PATH /home/iitp-app/IITP-DABT-DataCollector/.env
   ```

3. CSV 파일 경로가 올바른지 확인:

   ```
   ls -l /home/iitp-app/IITP-DABT-DataCollector/data/images.csv
   ```

7.1.2. CSV 파일 오류

오류 메시지:

```
ERROR: CSV headers must be exactly: No, Type, Title, Img-link; got: ...
```

해결 방법:

1. CSV 파일의 헤더 확인:

   ```
   head -1 /home/iitp-app/IITP-DABT-DataCollector/data/images.csv
   ```

2. 헤더가 정확히 `No,Type,Title,Img-link` 인지 확인 (대소문자 구분)

3. CSV 파일 인코딩 확인:

```
file -i /home/iitp-app/IITP-DABT-DataCollector/data/images.csv
```

UTF-8 또는 UTF-8 BOM이어야 합니다.

7.1.3. Python 모듈 오류

오류 메시지:

```
ModuleNotFoundError: No module named 'requests'
```

해결 방법:

1. 가상환경이 활성화되어 있는지 확인:

```
which python3
# 출력: /home/iitp-app/IITP-DABT-DataCollector/venv/bin/python3
```

2. 의존성 패키지 재설치:

```
cd /home/iitp-app/IITP-DABT-DataCollector
source venv/bin/activate
pip install -r requirements.txt
```

7.2. 권한 문제

7.2.1. 디렉토리 쓰기 권한 오류

오류 메시지:

```
PermissionError: [Errno 13] Permission denied: '/home/iitp-app/IITP-DABT-DataCollector/download
```

해결 방법:

1. 디렉토리 소유권 확인:

```
ls -ld /home/iitp-app/IITP-DABT-DataCollector/downloads
```

2. 소유권 변경:

```
sudo chown -R iitp-app:iitp-app /home/iitp-app/IITP-DABT-DataCollector
```

3. 권한 확인:

```
chmod -R 755 /home/iitp-app/IITP-DABT-DataCollector
```

7.2.2. CSV 파일 읽기 권한 오류

해결 방법:

```
chmod 644 /home/iitp-app/IITP-DABT-DataCollector/data/images.csv
```

7.3. 네트워크 문제

7.3.1. 연결 타임아웃

오류 메시지:

```
Timeout
Connection timeout
```

해결 방법:

1. 네트워크 연결 확인:

```
ping -c 3 8.8.8.8
curl -I https://www.google.com
```

2. 방화벽 설정 확인:

```
sudo ufw status
```

3. 프록시 설정이 필요한 경우 `.env` 파일에 `REQUEST_HEADERS_JSON` 추가

7.3.2. SSL 인증서 오류

오류 메시지:

```
SSL: CERTIFICATE_VERIFY_FAILED
```

해결 방법:

1. 운영 환경에서는 `VERIFY_SSL=true` 유지 권장
2. 내부 서버의 경우에만 임시로 `VERIFY_SSL=false` 설정 (보안 위험)
3. 인증서 설치:

```
sudo apt install ca-certificates
```

8. 부록

A. 디렉토리 구조

설치 완료 후 전체 디렉토리 구조:

```
/home/iitp-app/IITP-DABT-DataCollector/
├── venv/                    # Python 가상환경
│   ├── bin/
│   ├── lib/
│   └── ...
├── data/                    # CSV 파일 저장 디렉토리
│   ├── images.csv           # 입력 CSV 파일
│   ├── images_errorRow.csv  # 에러 리포트 (실패 시 생성)
│   └── images_summary.csv   # 에러 요약 (analyze_errors.py 실행 시 생성)
├── downloads/               # 다운로드된 이미지 저장 디렉토리
│   └── YYYY-MM-DD/          # 날짜별 폴더
│       └── *.jpg, *.png 등  # 다운로드된 이미지 파일
├── logs/                    # 로그 파일 저장 디렉토리
│   ├── image_downloader_YYYY-MM-DD.log
│   └── cron.log             # Cron 실행 로그 (선택사항)
├── downloader.py            # 메인 다운로더 프로그램
├── analyze_errors.py        # 에러 분석 유틸리티
├── requirements.txt         # Python 패키지 의존성
├── env.sample               # 환경 변수 샘플 파일
├── .env                     # 환경 변수 설정 파일 (생성 필요)
├── run.sh                   # 수동 실행 스크립트 (선택사항)
└── run_cron.sh              # Cron 실행 스크립트 (선택사항)
```

B. 환경 변수 참고

B.1. 환경 변수 요약

변수명	필수	기본값	설명
URL_CSV_PATH	예	없음	처리할 CSV 파일 경로

변수명	필수	기본값	설명
LOG_LEVEL	아니오	INFO	로깅 레벨
ROOT_DIR	아니오	./downloads	이미지 저장 루트 디렉토리
THREADS	아니오	8	병렬 다운로드 스레드 수
HEAD_CHECK	아니오	false	HEAD 요청 사전 검증 여부
VERIFY_SSL	아니오	true	SSL 인증서 검증 여부
REQUEST_HEADERS_JSON	아니오	빈 문자열	추가 HTTP 헤더 (JSON 형식)

B.2. .env 파일 템플릿

```
# Logging level: DEBUG, INFO, WARNING, ERROR, CRITICAL
LOG_LEVEL=INFO

# Root directory where date-based folder (YYYY-MM-DD) will be created to store images
ROOT_DIR=/home/iitp-app/IITP-DABT-DataCollector/downloads

# Number of parallel download threads
THREADS=8

# Full path (including filename) to the CSV file with columns: No,Type,Title,Img-link
URL_CSV_PATH=/home/iitp-app/IITP-DABT-DataCollector/data/images.csv

# Optional: perform HTTP HEAD to check Content-Type is image/* before GET (true/false)
HEAD_CHECK=false

# Optional: verify SSL certificates on HTTPS requests (true/false). Set false only if necessary.
VERIFY_SSL=true

# Optional: extra request headers in JSON (e.g., Referer or custom User-Agent)
# Example: {"Referer":"https://example.com","User-Agent":"Mozilla/5.0"}
REQUEST_HEADERS_JSON=
```

B.3. 유용한 명령어

환경 변수 확인:

```
cd /home/iitp-app/IITP-DABT-DataCollector
source venv/bin/activate
python3 -c "from dotenv import load_dotenv; import os; load_dotenv(); print('URL_CSV_PATH:', os
deactivate
```

디스크 사용량 확인:

```
du -sh /home/iitp-app/IITP-DABT-DataCollector/downloads/*
df -h /home/iitp-app/IITP-DABT-DataCollector
```

프로세스 확인:

```
ps aux | grep downloader.py
```

포트 및 네트워크 확인:

```
netstat -tuln | grep :80
curl -I https://example.com
```