

ProjectView 설치형 백업 및 복구 가이드

본 문서는 ProjectView 백업 및 복구에 대한 가이드입니다.

기본적으로 설치 프로젝트 별로 백업 환경이 다를 수 있기 때문에 백업은 자동으로 등록이 되지 않습니다.

본 문서를 참고하여 정기적으로 백업을 할 수 있도록 구성하시기 바랍니다.

ProjectView 백업

설치가 완료되면 설치 디렉토리에 backup-projectview.sh 파일이 생성됩니다.

ProjectView 백업은 아래 3가지의 백업으로 구성됩니다.

- 애플리케이션 백업
- 데이터베이스 백업
- 첨부파일 백업

애플리케이션 백업은 설치 가이드에서 설치 후 설치 패키지를 별도로 보관하도록 가이드하고 있습니다.

애플리케이션 복구는 설치를 다시 진행하면 되기 때문에 본 백업에서는 데이터베이스 백업과 첨부파일 백업만을 다룹니다.

아래는 백업 구성 절차입니다.

1. 백업 환경 확인하기

설치 환경에 따라 백업을 할 대상 환경이 다를 수 있습니다.

본 문서에서는 백업 스토리지를 별도로 설치 장비에 마운트하여 사용하고 있는 것을 기준으로 작성합니다.

별도의 장비에 백업을 하는 등 환경이 다른 경우에는 백업 스크립트를 프로젝트 환경에 맞게 수정하여 적용을 해주시기 바랍니다.

2. 백업 디렉토리 생성

백업 스토리가 되어 있는 디렉토리 경우에 백업할 디렉토리를 생성합니다.

필요한 경우 백업 디렉토리에 권한을 부여합니다.

```
# 백업 디렉토리를 생성합니다. 아래 경로는 샘플이며 마운트 된 백업 스토리지 경로 아래 생성해 줍니다.  
# 권한이 없는 경우 sudo를 활용합니다.  
mkdir -p /projectview-backup  
  
# sudo로 생성할 경우, 백업 디렉토리의 owner를 projectview 관리에 사용하는 계정으로 변경합니다.  
sudo chown promise:promise /projectview-backup
```

3. backup-projectview.sh 파일 수정

설치 경로 아래에 있는 backup-projectview.sh 파일 내 백업 스토리지 경로를 수정합니다.

```
# 설치 디렉토리로 이동합니다. 아래는 샘플 경로입니다.  
cd /app/projectview  
  
# backup-projectview.sh 파일을 수정합니다.  
vi backup-projectview.sh  
  
# 아래는 backup-projectview.sh 파일 내용입니다.  
#!/bin/bash  
  
BACKUP_PATH=/projectview-backup # 이 경로를 백업 디렉토리 경로로 변경해줍니다.
```

4. 백업 테스트

backup-projectview.sh 파일을 실행하여 백업이 정상적으로 잘 되는지 확인합니다.

```
# backup-projectview.sh 파일을 실행합니다.  
./backup-projectview.sh
```

정상적으로 실행이 되면 설정한 백업 디렉토리에 오늘 날짜로 디렉토리가 생성되며, 첨부 압축 파일과 DB Dump 파일이 생성되어 있습니다.

* 백업은 3일치만 보관하고 삭제되도록 backup-projectview.sh 스크립트에 구성이 되어 있습니다. 필요 시 수정해서 사용해주시기 바랍니다.

ProjectView 정기 백업 구성

OS의 crontab을 이용하여 정기적으로 백업이 실행되도록 구성합니다.

```
# crontab 명령어를 사용하여 crontab 파일을 엽니다.  
crontab -e  
  
# 열린 파일에 아래의 내용을 추가합니다.  
# i 키를 누르면 수정 모드로 수정할 수 있습니다.  
0 1 * * * /app/projectview/backup-projectview.sh  
  
# 저장합니다.  
# :wq 를 입력하면 저장할 수 있습니다.  
  
# 등록된 crontab을 확인합니다.  
crontab -l
```

ProjectView 복구

ProjectView의 복구는 애플리케이션 설치, 데이터베이스 Import, 첨부파일 복구 순으로 진행이 됩니다.

애플리케이션 설치

설치가이드를 참고하여 신규 설치를 진행합니다.

데이터베이스 복구

본 가이드에서는 public 스키마를 전체 삭제한 후 다시 복구하는 과정으로 진행합니다.

```
# pms-postgresql 컨테이너 내 DB Dump 파일을 복사합니다.
# 아래에서 보이는 경로와 파일명은 샘플입니다.
docker cp /projectview-backup/20240325/projectview-20240325.dat pms-postgresql:/projectview-20240325.dat

# pms-postgresql 컨테이너로 들어갑니다.
docker exec -it pms-postgresql /bin/bash

# 컨테이너 안에서 psql 명령을 실행합니다.
psql -p 35432 -d promise

# 아래 명령어들을 입력하여 public 스키마를 삭제하고 다시 생성해줍니다.
DROP SCHEMA public CASCADE;
CREATE SCHEMA public;

GRANT ALL ON SCHEMA public TO postgres;
GRANT ALL ON SCHEMA public TO public;
GRANT ALL ON SCHEMA public TO promise;

# \q를 입력하여 psql을 종료합니다.

# 아래 커맨드를 이용하여 DB Dump를 Import합니다.
psql -p 35432 -d promise -f projectview-20240325.dat

# 컨테이너 내 DB Dump를 삭제합니다.
rm -f projectview-20240325.dat

# exit 명령어로 컨테이너에서 빠져나옵니다.
```

첨부파일 복구

본 가이드에서는 첨부파일 디렉토리를 삭제한 후 다시 복구하는 과정으로 진행합니다.

```
# uploadFiles 디렉토리를 삭제합니다.
# 아래는 샘플 경로와 파일명을 사용합니다. 설치 환경에 맞게 수정하여 사용하시기 바랍니다.
rm -rf /app/projectview/uploadFiles

# 디렉토리를 다시 생성합니다.
mkdir -p /app/projectview/uploadFiles

# 첨부파일 백업 압축파일을 복사합니다.
cp /app/projectview/uploadFile/projectview-attach-backup-20240325.tar.gz /app/projectview/uploadFiles/

# 첨부파일 경로로 이동합니다.
cd /app/projectview/uploadFiles/

# 압축은 풀니다.
tar -zxvf projectview-attach-backup-20240325.tar.gz

# 압축 파일을 삭제합니다.
rm -f projectview-attach-backup-20240325.tar.gz
```

복구가 완료되었습니다.

ProjectView에 접속하여 복구가 잘 되었는지 테스트합니다.