

ProjectView 설치형 설치 가이드

본 문서는 ProjectView 설치형 버전에 대한 설치 가이드입니다.

지원하는 운영 체제

- **Linux OS**
 - 권장 OS : Redhat Enterprise Linux, Ubuntu
 - 본 설치 패키지는 Redhat Enterprise Linux 8.5, Ubuntu 20.04.6에서 테스트 되었습니다.
- **Windows OS는 현재 지원하지 않습니다.**

사전 준비 사항

- **설치 대상 장비**
 - 단일 장비 설치를 원칙으로 합니다. 동일 장비 내 다른 툴 / 서비스가 설치될 경우, 포트 중복 등의 이슈로 설치가 되지 않을 수 있습니다.
 - 설치 장비와 별도로 백업 스토리지가 필요합니다.
 - 설치 시 192.168.30.0/24 대역으로 내부 네트워크를 구성해서 사용합니다. 설치된 장비의 네트워크 IP가 192.168.30.0/24 대역을 사용할 경우 충돌로 설치가 되지 않습니다.

- **Linux OS**
- **Docker CE (17.06.0 이상)**
- **Docker Compose (1.18.0 이상)**
- **Docker와 Docker Compose는 사전 설치가 필요합니다.**

Linux OS의 종류, 버전, 설치된 라이브러리의 버전, Package Manager(yum, apt-get 등) 사용 가능 여부 등에 따라 설치 방식이 다릅니다.

게다가 네트워크 이슈 등의 이유로 Package Manager를 활용할 수 없는 경우 설치에 필요한 라이브러리 버전들을 모두 준비할 수가 없어 본 설치 패키지 및 가이드에서는 사전 조건으로 가져옵니다.

최소 사양

- CPU 4코어
- 메모리 16GB
- Storage 100GB
 - 데이터베이스 데이터, 첨부파일, 로그파일
 - 프로젝트 기간, 사용자수, 사용 기능에 따라 필요한 용량은 달라질 수 있습니다.
 - 내부 의사소통을 위해 프로젝트 메시지를 사용하는 경우, 메시지 첨부 파일들이 많아 추가 스토리지가 필요합니다.

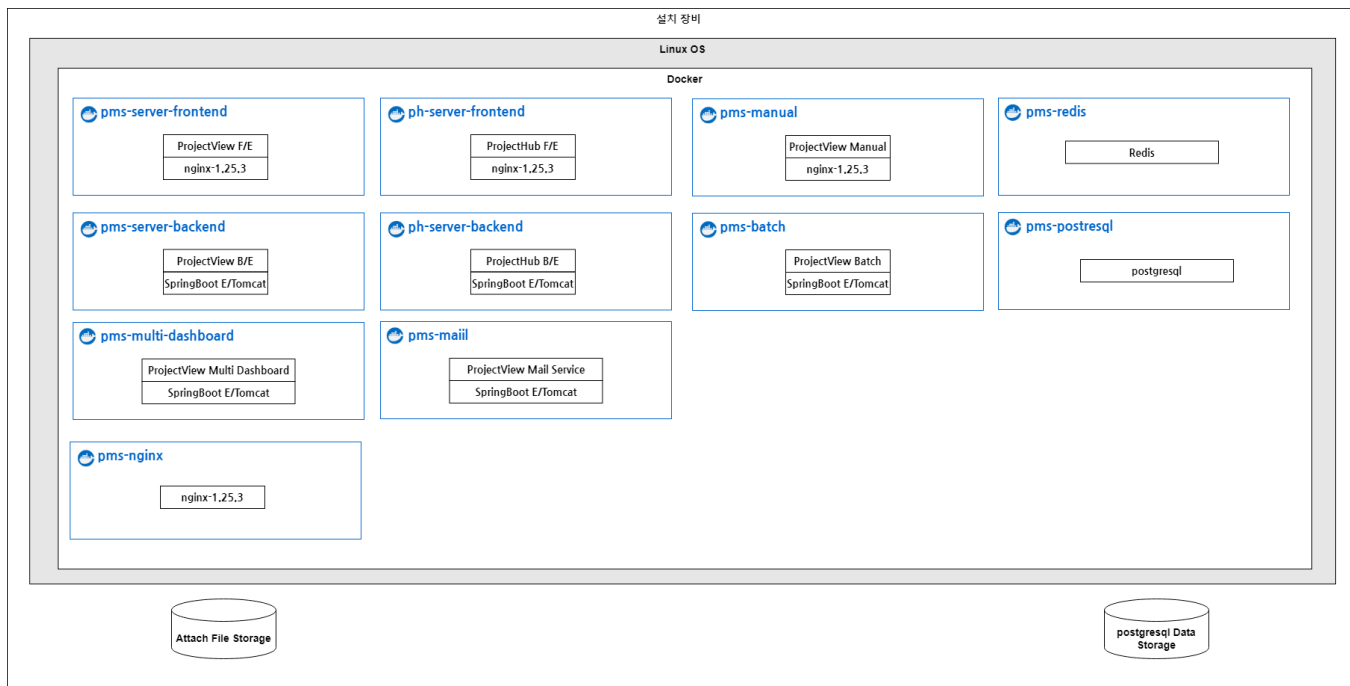
제약 조건

- 단일 장비 설치를 기준으로 작성이 되었으며, HA 지원을 위한 아키텍처 구성은 설치 패키지에 포함되어 있지 않습니다.

설치 항목

- PostgreSQL Database
- Redis
- ProjectHub - ProjectView 인증서비스
- ProjectView
- ProjectView 매뉴얼
- ProjectView 배치
- ProjectView 메일 서비스
- 멀티대시보드

설치 구성 요소



설치 가이드

0. 본 문서에서는 아래의 경로에 설치 패키지를 복사하여 설치하는 것을 기준으로 설명합니다.

프로젝트 상황에 따라서 경로는 적절하게 변경하여 진행해주시기 바랍니다.

패키지 파일 복사 경로 : /app/installer
설치 경로 : /app/projectview

1. 제공된 설치 패키지를 다운로드, 설치 장비로 복사합니다.

복사 방식은 OS의 종류, GUI 지원 여부 등에 따라 다 다를 수 있습니다.

우선 필요한 디렉토리를 생성합니다.

```
mkdir -p /app/installer  
mkdir -p /app/projectview
```

위 커맨드에서 권한이 없는 경우, sudo 권한을 부여하여 생성합니다. 이후 Owner User, Owner User Group을 변경해줍니다.

```
sudo mkdir -p /app/installer  
sudo mkdir -p /app/projectview
```

sudo chown -r promise:promise /app #chown [Owner]:[Owner Group] 형태로 사용합니다. Owner로 현재 로그인 하신 사용자를 넣어주시면 됩니다.

(Ubuntu 22.04에서는 chown -R 혹은 --recursive 옵션 사용해야함)

아래는 상황 별 copy 가이드 입니다.

1. GUI를 제공하는 OS 일 경우, USB, 이메일을 통한 다운로드 등 가능한 방법을 통해서 직접 위 경로에 복사해주시면 됩니다.

2. GUI를 제공하지 않는 경우, 장비에 연결된 다른 PC를 통해서 복사합니다. WinSCP나 Filezilla 같은 프로그램을 사용해서 복사를 진행합니다.

디렉토리 생성 및 복사가 완료되면 아래의 상태가 됩니다.

```
promise@projectview-on-premise1:/app/installer$ pwd  
/app/installer  
promise@projectview-on-premise1:/app/installer$ ll  
total 4234332  
drwxrwxr-x 2 promise promise      4096 Mar 19 03:48 ./  
drwxr-xr-x 4 promise promise      4096 Mar 19 03:48 /  
-rw-rw-r-- 1 promise promise 4335943680 Mar 19 03:47 projectview-installer-v2.0.4.tar  
promise@projectview-on-premise1:/app/installer$
```

2. 설치 패키지 압축을 해제합니다.

설치파일 복사한 directory로 이동하여 tar 명령어를 이용하여 압축을 해제 합니다.

```
cd /app/installer
```

```
tar -xvf projectview-installer-v2.0.5.tar #projectview-installer- 파일의 버전 정보는 제공된 패키지의 버전에 따라 다를 수 있습니다.
```

```

promise@projectview-on-premise1:/app/installer$
promise@projectview-on-premise1:/app/installer$ tar xvf projectview-installer-v2.0.4.tar
./
./.git/
./.git/logs/
./.git/logs/HEAD
./.git/logs/refs/
./.git/logs/refs/remotes/
./.git/logs/refs/remotes/origin/
./.git/logs/refs/remotes/origin/master
./.git/config
./.git/index
./.git/objects/
./.git/objects/info/
./.git/objects/pack/
./.git/objects/pack/pack-632de91a103bb1b6eb60556e24016b7103477cb6.pack
./.git/objects/pack/pack-632de91a103bb1b6eb60556e24016b7103477cb6.idx
./.git/HEAD
./.git/description
./.git/hooks/
./.git/hooks/fsmonitor-watchman.sample
./.git/hooks/update.sample
./.git/hooks/pre-commit.sample

```

```

promise@projectview-on-premise1:/app/installer$ ll
total 8412084
drwxr-xr-x 4 promise promise      4096 Mar 19 03:47 ./
drwxr-xr-x 4 promise promise      4096 Mar 19 03:48 ../
-rw-rw-r-- 1 promise promise     3475 Mar 19 01:21 common.sh
-rw-r--r-- 1 promise promise      429 Mar 19 00:16 config.env
-rwxr--r-x 1 promise promise      520 Mar 14 07:00 create_nginx_log_rotate.sh*
drwxr-xr-x 5 promise promise      4096 Mar 14 07:00 data/
-rw-r--r-- 1 promise promise     7191 Mar 14 07:00 docker-compose.yml
-rw-r--r-- 1 promise promise    16384 Mar 19 02:25 .docker-compose.yml.swp
drwxr-xr-x 8 promise promise      4096 Mar 14 07:00 .git/
-rwxr--r-x 1 promise promise     2929 Mar 19 01:22 install.sh*
-rw-r--r-- 1 promise promise     2731 Mar 14 07:00 LICENSE
-rw-rw-r-- 1 promise promise 4335943680 Mar 19 03:47 projectview-installer-v2.0.4.tar
-rw-r--r-- 1 promise promise 4277955232 Mar 14 07:07 projectview-v2.0.4.tgz
-rw-r--r-- 1 promise promise       57 Mar 14 07:00 README.md
promise@projectview-on-premise1:/app/installer$

```

3. 설치 환경과 설치 패키지 버전에 맞게 환경 설정 파일을 수정합니다.

vi 에디터를 이용하여 수정하셔도 되고, 다른 PC에서 수정 후 복사해주셔도 됩니다.

vi 에디터를 사용하는 경우, i 버튼을 누르면 INSERT 모드로 변경해서 수정하실 수 있습니다. 수정이 완료되면, ESC를 누르고 :wq를 입력, 엔터하면 저장 후 종료가 됩니다.

```
# 설치 환경에 맞도록 config.env 파일을 수정합니다.  
vi config.env  
  
# config.env 파일의 내용은 아래와 같습니다.  
  
# ProjectView Version - Use for docker image version to load and docker compose  
# 설치할 ProjectView의 버전입니다.  
PROJECTVIEW_VERSION=v2.0.5  
  
# Install target directory path  
# ProjectView 설치 경로입니다.  
INSTALL_TARGET_PATH=/app/projectview  
  
# Path to install postgresql database data files  
# Database의 데이터 파일 경로입니다. 수정하지 않습니다.  
DATABASE_DATA_PATH=${INSTALL_TARGET_PATH}/postgresql/data  
  
# Path of attach files, logs  
# 로그파일 및 첨부파일 경로입니다. 수정하지 않습니다.  
PROJECTVIEW_ATTACH_FILE_PATH=${INSTALL_TARGET_PATH}/uploadFiles  
PROJECTVIEW_LOG_PATH=${INSTALL_TARGET_PATH}/logs
```

4. 설치를 시작하기 전 docker, docker compose의 버전을 확인합니다.

```
# docker와 docker compose 버전을 확인합니다. docker 17.06.0 이상, docker compose 1.18.0 이상이어야 합니다.  
docker version  
docker compose version
```

```
promise@projectview-on-premise1:/app/installer$ docker version
Client: Docker Engine - Community
 Version:      25.0.4
 API version:  1.44
 Go version:   go1.21.8
 Git commit:   1a576c5
 Built:        Wed Mar  6 16:32:14 2024
 OS/Arch:     linux/amd64
 Context:      default

Server: Docker Engine - Community
 Engine:
  Version:      25.0.4
  API version:  1.44 (minimum version 1.24)
  Go version:   go1.21.8
  Git commit:   061aa95
  Built:        Wed Mar  6 16:32:14 2024
  OS/Arch:     linux/amd64
  Experimental: false
 containerd:
  Version:      1.6.28
  GitCommit:    ae07eda36dd25f8a1b98dfbf587313b99c0190bb
 runc:
  Version:      1.1.12
  GitCommit:    v1.1.12-0-g51d5e94
 docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0
promise@projectview-on-premise1:/app/installer$ docker compose version
Docker Compose version v2.24.7
promise@projectview-on-premise1:/app/installer$
```

5. 설치를 실행합니다.

설치 파일을 실행하기에 앞서, 설치 파일의 위치한 디렉토리로 이동하여 실행해주어야 합니다.

```
# 우선 installer가 있는 디렉토리로 이동합니다.
cd /app/installer

# 설치 파일에 실행 권한을 부여합니다.
chmod 747 install.sh

# 설치 파일을 실행합니다.
./install.sh

# 필요한 환경 구성, DB 구성 및 서비스 기동까지 자동으로 진행됩니다.
```



```
promise@projectview-on-premise1:/app/installer$ chmod 747 install.sh
promise@projectview-on-premise1:/app/installer$ ./install.sh
```

--- ProjectView installation is started.---

[Step 1]: checking if docker is installed ...

Note: docker version: 25.0.4

[Step 2]: checking docker-compose is installed ...

Note: Docker Compose version v2.24.7

[Step 3]: loading ProjectView docker images ...

→ Load projectview v2.0.4

Loaded image: projectview/pms-server-frontend:v2.0.4

Loaded image: projectview/ph-server-frontend:v2.0.4

Loaded image: projectview/postgres:13.13

Loaded image: projectview/redis:7.2.3

Loaded image: projectview/pms-server-backend:v2.0.4

Loaded image: projectview/pms-multi-dashboard:v2.0.4

Loaded image: projectview/pms-manual:v2.0.4

Loaded image: projectview/pms-mail:v2.0.4

Loaded image: projectview/pms-batch:v2.0.4

Loaded image: projectview/ph-server-backend:v2.0.4

Loaded image: projectview/base-frontend:rhel-8.9-1107.1705420509-nginx-1.25.3

[Step 4]: preparing environment ...

→ Install target path : /app/projectview

→ Database install path : /app/projectview/postgresql/data

→ Directory for database data is created. - /app/projectview/postgresql/data

→ Copy initial database files.

→ Directory for attach files is created. - /app/projectview/uploadFiles

→ Directory for log files is created. - /app/projectview/logs

→ Copy nginx conf files.

→ Copy cert files.

→ Copy service scripts.

Note: stopping existing ProjectView instance ...

Warning: No resource found to remove for project "projectview".

[Step 5]: starting ProjectView ...

[+] Running 11/12

Network projectview_promise-network	Created
✓ Container pms-manual	Healthy
✓ Container pms-redis	Healthy
✓ Container pms-postgresql	Healthy
✓ Container pms-server-backend	Healthy
✓ Container ph-server-backend	Healthy
✓ Container pms-multi-dashboard	Healthy
✓ Container pms-batch	Healthy
✓ Container pms-mail	Healthy
✓ Container pms-server-frontend	Healthy
✓ Container ph-server-frontend	Healthy
✓ Container pms-nginx	Healthy

```
✓ --- ProjectView has been installed and started successfully.---
```

6. 설치 결과를 확인합니다.

설치가 완료되면 ProjectView 서비스가 모두 기동되어 있는 상태입니다. ProjectView 서비스가 잘 기동 되어 있는지 점검하고, ProjectView 서비스 시작, 종료에 필요한 파일들이 잘 구성되어 있는지 확인합니다.

```
# ProjectView 설치 디렉토리로 이동합니다.
cd /app/projectview
```

```
# 생성된 파일들을 확인합니다.
ls -l
```

```
# 아래와 같이 조회되면 설치가 잘 된 것입니다.
```

```
promise@projectview-on-premise1:/app/projectview$ cd /app/projectview
promise@projectview-on-premise1:/app/projectview$ ll
total 48
drwxrwxr-x 7 promise promise 4096 Mar 19 03:59 ./
drwxr-xr-x 4 promise promise 4096 Mar 19 03:48 ../
drwxrwxr-x 2 promise promise 4096 Mar 19 03:59 cert/
drwxrwxr-x 6 promise promise 4096 Mar 19 03:59 conf/
-rw-r--r-- 1 promise promise 429 Mar 19 03:59 config.env
-rw-r--r-- 1 promise promise 7473 Mar 19 03:59 docker-compose.yml
drwxrwxr-x 8 promise promise 4096 Mar 19 03:59 logs/
drwxrwxr-x 3 promise promise 4096 Mar 19 03:59 postgresql/
-rwxr--rwx 1 promise promise 62 Mar 19 03:59 shutdown-projectview.sh*
-rwxr--rwx 1 promise promise 128 Mar 19 03:59 startup-projectview.sh*
drwxrwxr-x 2 promise promise 4096 Mar 19 04:00 uploadFiles/
```

```
# ProjectView 서비스들이 정상적으로 기동 되어 있는지 확인합니다.
docker ps
```

```
# 아래와 같이 조회 되면 정상적인 상태입니다.
```

```
promise@projectview-on-premise1:/app/projectview$ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
1e05ade07d5e	projectview/base-frontent:rhel-8.9-1107.1705420509-nginx-1.25.3	pms-nginx	"/usr/local/nginx/sbin..."	11 minutes ago	Up 8 minutes (healthy)	0.0.0.0:443->443/tcp, :::443->443/tcp
1ab18229c9a4	projectview/pms-server-frontent:v2.0.4	pms-server-frontent	"/usr/local/nginx/sbin..."	11 minutes ago	Up 8 minutes (healthy)	0.0.0.0:8090->8090/tcp, :::8090->8090/tcp
1d6aa1b97b72	projectview/ph-server-frontent:v2.0.4	ph-server-frontent	"/usr/local/nginx/sbin..."	11 minutes ago	Up 8 minutes (healthy)	0.0.0.0:8020->8020/tcp, :::8020->8020/tcp
c976c9fe7668	projectview/pms-multi-dashboar:v2.0.4	pms-multi-dashboar	"/bin/sh -c 'exec ja..."	11 minutes ago	Up 9 minutes (healthy)	0.0.0.0:8060->8060/tcp, :::8060->8060/tcp
9fe538b03a1f	projectview/pms-mail:v2.0.4	pms-mail	"/bin/sh -c 'exec ja..."	11 minutes ago	Up 9 minutes (healthy)	0.0.0.0:8050->8050/tcp, :::8050->8050/tcp
1e3f4949c34e	projectview/pms-batch:v2.0.4	pms-batch	"/bin/sh -c 'exec ja..."	11 minutes ago	Up 9 minutes (healthy)	0.0.0.0:8081->8081/tcp, :::8081->8081/tcp
32c4de5f1a13	projectview/ph-server-backend:v2.0.4	ph-server-backend	"java -jar /promise-..."	11 minutes ago	Up 9 minutes (healthy)	0.0.0.0:8010->8010/tcp, :::8010->8010/tcp
1a53015ac4f3	projectview/pms-server-backend:v2.0.4	pms-server-backend	"/bin/sh -c 'exec ja..."	11 minutes ago	Up 10 minutes (healthy)	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
ffe32e9353e2	projectview/pms-manual:v2.0.4	pms-manual	"/usr/local/nginx/sbin..."	11 minutes ago	Up 11 minutes (healthy)	80/tcp, 5432/tcp, 8080-8081/tcp, 8090/tcp, 0.0.0.0:8088->8088/tcp, :::8088->8088/tcp
09e2a2d4122	projectview/redis:7.2.3	pms-redis	"docker-entrypoint.s..."	11 minutes ago	Up 11 minutes (healthy)	0.0.0.0:36379->6379/tcp, :::36379->6379/tcp
b051610ec8a6	projectview/postgres:13.13	pms-postgresql	"docker-entrypoint.s..."	11 minutes ago	Up 11 minutes (healthy)	5432/tcp, 0.0.0.0:35432->35432/tcp, :::35432->35432/tcp

```
promise@projectview-on-premise1:/app/projectview$
```

서비스를 종료해 봅니다.
./shutdown-projectview.sh

아래는 서비스 종료 결과입니다.

```
promise@projectview-on-premise1:/app/projectview$ pms-postgresql
promise@projectview-on-premise1:/app/projectview$ ./shutdown-projectview.sh
[+] Running 12/12
✓ Container pms-nginx Removed
✓ Container pms-batch Removed
✓ Container ph-server-frontend Removed
✓ Container pms-server-frontend Removed
✓ Container pms-manual Removed
✓ Container pms-multi-dashboard Removed
✓ Container ph-server-backend Removed
✓ Container pms-mail Removed
✓ Container pms-server-backend Removed
✓ Container pms-postgresql Removed
✓ Container pms-redis Removed
✓ Network projectview_promise-network Removed
```

서비스를 시작해 봅니다.
./startup-projectview.sh

아래는 서비스 시작 결과입니다.

```
promise@projectview-on-premise1:/app/projectview$ ./startup-projectview.sh
Warning: No resource found to remove for project "projectview".
[+] Running 11/12
: Network projectview_promise-network Created
✓ Container pms-postgresql Healthy
✓ Container pms-redis Healthy
✓ Container pms-manual Healthy
✓ Container pms-batch Healthy
✓ Container pms-server-backend Healthy
✓ Container ph-server-backend Healthy
✓ Container pms-multi-dashboard Healthy
✓ Container pms-mail Healthy
✓ Container pms-server-frontend Healthy
✓ Container ph-server-frontend Healthy
✓ Container pms-nginx Healthy
promise@projectview-on-premise1:/app/projectview$
```

아래는 설치 디렉토리 내 파일들 설명입니다.

■ cert : site.projectview.io 도메인에 대한 인증서가 있는 디렉토리입니다.

- **conf** : ProjectView 서비스의 웹서버인 nginx 기동에 필요한 설정파일들이 있는 디렉토리입니다.
- **logs** : 각 서비스 별 로그파일이 있는 디렉토리입니다.
- **postgresql** : DBMS인 PostgreSQL의 데이터 파일이 있는 디렉토리입니다.
- **uploadFiles** : ProjectView에서 올린 첨부파일이 저장되는 디렉토리입니다.
- **config.env** : ProjectView 서비스 기동에 사용하는 설정 파일입니다.
- **docker-compose.yml** : ProjectView 서비스를 일괄 기동, 종료하는데 사용하는 docker compose 설정 파일입니다.
- **shutdown-projectview.sh** : ProjectView 서비스 종료 실행 파일입니다.
- **startup-projectview.sh** : ProjectView 서비스 시작 실행 파일입니다.

7. ProjectView 서비스 접속이 잘 되는지 테스트 합니다.

접속 테스트에 사용할 PC의 hosts 파일에 ProjectView 도메인과 IP가 추가되어야 합니다.

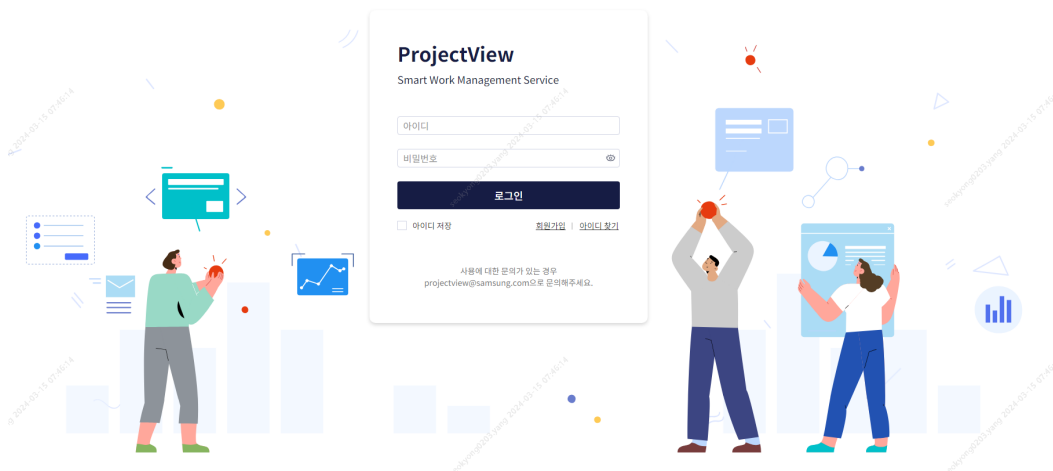
그리고 ProjectView와 https 프로토콜로 통신하기 위해 사설 인증서를 신뢰할 수 있는 인증서로 설치해주어야 합니다.

문서 하단의 [[ProjectView 설치형 사용자 초기 구성 가이드](#)]를 참고하여 개인 PC에 구성합니다.

구성이 완료되었으면 아래 url을 이용하여 브라우저에서 접속합니다.

<https://site.projectview.io/projectview/>

아래의 화면이 보인다면 설치 및 접속에 성공한 것입니다.



8. ProjectView 웹서버 로그 백업을 설정합니다.

ProjectView에서는 웹서버로 nginx를 사용하고 있습니다. nginx의 로그를 일 단위로 나눠서 보관하기 위해서는 logrotate 구성이 필요합니다.

linux의 logrotate를 이용하여 구성을 할 것이며, linux 로그인 사용자에게 sudo 권한이 필요합니다.

```
# 우선 installer가 있는 디렉토리로 이동합니다.
cd /app/installer

# 실행 파일에 실행 권한을 부여합니다.
chmod 747 create_nginx_log_rotate.sh

# nginx logrotate 설정 파일을 실행합니다.sudo 권한으로 실행합니다.
sudo ./create_nginx_log_rotate.sh

# nginx logrotate 파일이 잘 위치해 있는지 확인합니다. 아래 명령어 실행 결과에서 nginx 파일이 존재하면 됩니다.
ls -l /etc/logrotate.d/
```

9. ProjectView 백업을 구성합니다.

백업 구성은 함께 제공된 별도의 백업 가이드를 통해서 설정을 진행합니다.

10. ProjectView 설치가 완료되었습니다.

ProjectView 설치가 완료되었습니다.

이제 관리자 계정 생성 및 사용하실 프로젝트를 생성합니다.

11. 회원가입을 통해 관리자 계정을 생성합니다.

ProjectView

Smart Work Management Service



로그인

☐ 아이디 저장

[회원가입](#)

[아이디 찾기](#)

사용에 대한 문의가 있는 경우
projectview@samsung.com으로 문의해주세요.

회원가입

계정 정보

이메일 *	<input type="text" value="이메일"/>
아이디 *	<input type="text" value="아이디는 영문 또는 숫자로 입력해 주세요."/>
비밀번호 *	<input type="password" value="비밀번호를 입력해 주세요."/> 
비밀번호 재확인 *	<input type="password" value="비밀번호를 확인해 주세요."/> 

사용자 정보

이름 *	<input type="text" value="실명을 입력해 주세요."/>
영문 이름	<input type="text" value="영문 이름을 입력해 주세요."/>
회사 *	<div>선택 </div>
직급	<input type="text" value="직급명"/>
프로필 이미지	<div>여기에 파일을 추가해 주세요. (500KB 이내)</div>

이용약관 동의

- ☐ 전체동의
-
- ☐ 개인정보수집 및 이용에 대한 동의 (필수)
- ☐ 서비스 이용 약관 (필수)
- ☐ 14세 이상 여부 확인 (필수)

취소

가입

12. 시스템 관리자 계정으로 프로젝트를 생성합니다.

로그아웃 후 시스템 관리자 계정으로 로그인합니다.

시스템 관리자 계정 ID와 비밀번호는 ProjectView 지원 담당에게 문의해주시기 바랍니다.

시스템 관리자로 최초 로그인 하면 아래와 같이 비밀번호를 변경해주어야 합니다.

비밀번호 재설정

비밀번호 사용 기간이 만료되었습니다. 비밀번호를 재설정해주세요.

현재 비밀번호 *	<input type="password"/>	
신규 비밀번호 *	<input type="password"/>	✓
비밀번호 재확인 *	<input type="password"/>	✓

취소

저장

시스템 관리자로 로그인 후 아래 그림을 참고하여 사용할 프로젝트를 생성합니다.

1. 하단의 추가 버튼을 클릭합니다. 프로젝트 추가 팝업이 나타납니다.

- 프로젝트 Key는 우측의 버튼을 클릭하여 중복체크를 해주어야 합니다.
- 프로젝트 관리자는 11번 단계에서 생성했던 관리자 계정을 지정합니다.
- 프로젝트 템플릿은 아래를 참고하여 필요한 템플릿 유형을 선택합니다.

· 차세대형 : 구축형 + 현행시스템분석 관리 등 추가 메뉴 구성

프로젝트가 생성이 되고, 입력한 프로젝트 관리자 계정에 해당 프로젝트의 관리자 권한이 부여됩니다.

13. 관리자 계정으로 로그인하여 생성된 프로젝트를 확인합니다.

로그아웃 한 후 11번 단계에서 생성했던 관리자 계정으로 로그인 합니다.

프로젝트가 생성된 것을 확인할 수 있습니다.

14. ProjectView를 사용하기 위한 모든 준비가 완료되었습니다!!

ProjectView를 사용하기 위한 모든 준비가 완료되었습니다.

이제는 아래 매뉴얼을 참고하여 프로젝트 관리에 필요한 기능들을 구성하고, 사용하시면 됩니다.

- 관리자 매뉴얼 : <https://site.projectview.io/projectview-manual/admin-manual/>
- 사용자 매뉴얼 : <https://site.projectview.io/projectview-manual/user-manual/>

15. ProjectView 설치 패키지를 백업 스토리지에 복사해 둡니다.

추후 복구 시 사용될 수 있습니다.

이어서 해 볼 일

- 프로젝트 업무 구성하기
- 프로젝트 공정(WBS) 관리하기
- 프로젝트 회의실 등록 및 일정관리 하기
- 프로젝트에서 하는 일들 일감으로 관리하기

ProjectView 설치형 사용자 초기 구성 가이드

ProjectView 설치형은 site.projectview.io 도메인을 사용하도록 되어 있으며,

사용자들이 site.projectview.io 도메인으로 접속하기 위해서는 DNS 서버에 도메인이 추가되거나, 개인 PC의 hosts 파일에 도메인 정보가 추가되어야 합니다.

hosts 파일에 ProjectView 도메인 추가

DNS 서버에 ProjectView의 도메인을 추가할 수 없다면, 개인 PC의 hosts 파일에 도메인 정보를 추가해야 합니다.

```
# 메모장 등 에디터 프로그램을 이용하여 hosts 파일을 관리자 권한으로 열어줍니다.  
파일 경로 : C:\Windows\System32\drivers\etc\hosts  
  
# 아래 항목을 추가합니다.  
[ProjectView 장비의 IP] site.projectview.io  
ex) 192.168.98.55 site.projectview.io
```

ProjectView 사설 인증서 설치 - 신뢰할 수 있는 루트 인증 기관으로 추가

ProjectView는 https 프로토콜을 사용하기 때문에 site.projectview.io 도메인으로 사설인증서가 기본으로 설정되어 있습니다.

ProjectView의 사설인증서를 신뢰할 수 있는 인증서로 등록하지 않으면 아래의 화면을 만나게 됩니다.



연결이 비공개로 설정되어 있지 않습니다.

공격자가 **site.projectview.io**에서 정보(예: 비밀번호, 메시지, 신용카드 등)를 도용하려고 시도 중일 수 있습니다. [자세히 알아보기](#)

NET::ERR_CERT_AUTHORITY_INVALID

세부정보 숨기기

안전한 페이지로 돌아가기

이 서버가 **site.projectview.io**임을 입증할 수 없으며 컴퓨터의 운영체제에서 신뢰하는 보안 인증서가 아닙니다. 서버를 잘못 설정했거나 불법 사용자가 연결을 가로채고 있기 때문일 수 있습니다.

[site.projectview.io\(안전하지 않음\)\(으\)로 이동](#)

올바르게 ProjectView를 사용하기 위해서는 사용자 개인 PC에 ProjectView의 사설인증서를 신뢰할 수 있는 인증서로 등록해주어야 합니다.

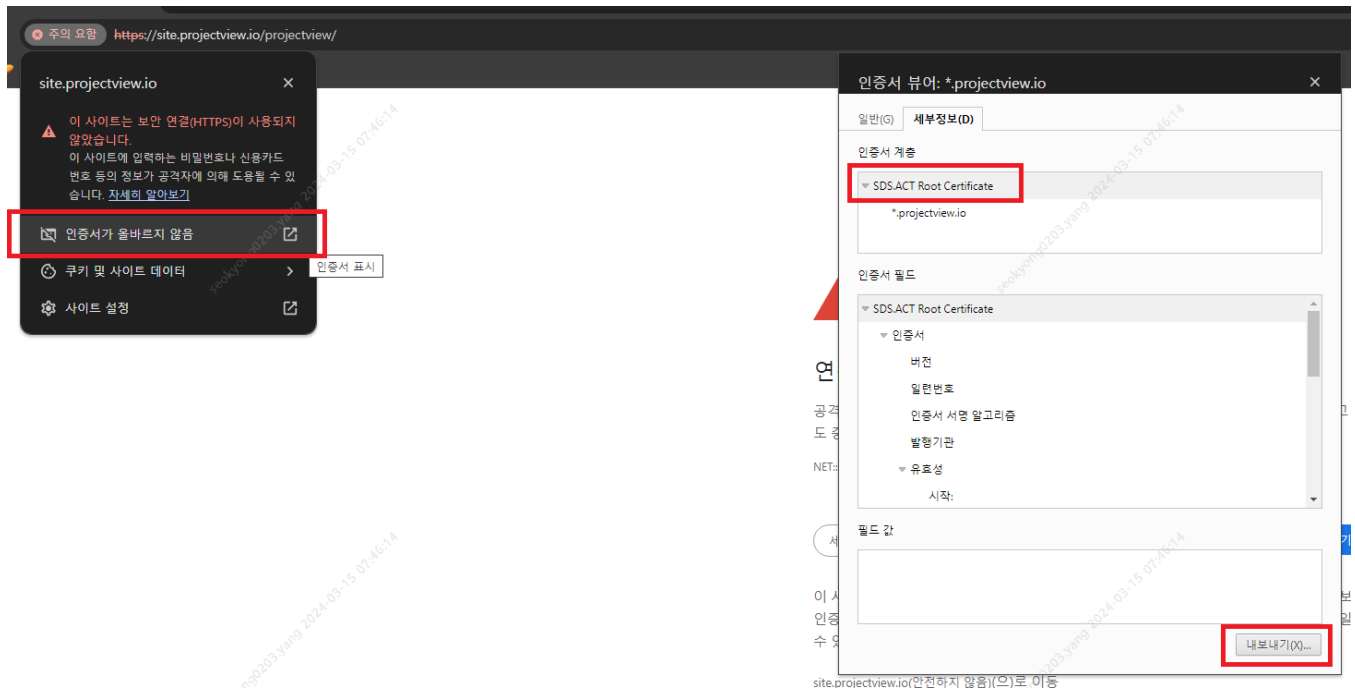
아래 절차를 따라 등록합니다.

1. 인증서 다운로드

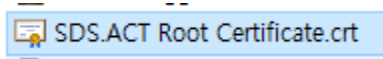
[SDS.ACT Root Certificate.crt](#)

설치패키지와 함께 제공되는 인증서를 가져옵니다.

혹 준비되지 않았다면 아래의 방식으로 다운로드도 가능합니다.



2. 인증서 실행



3. 신뢰할 수 있는 루트 인증 기관으로 등록

일반

자세히

인증 경로



인증서 정보

인증서의 용도:

- 모든 발급 정책
- 모든 응용 프로그램 정책

발급 대상: SDS.ACT Root Certificate

발급자: SDS.ACT Root Certificate

유효 기간(시작) 2024-01-23 부터 2034-01-20

인증서 설치(I)...

발급자 설명(S)

확인



← 인증서 가져오기 마법사

인증서 가져오기 마법사 시작

이 마법사를 사용하면 인증서, 인증서 신뢰 목록, 인증서 해지 목록을 디스크에서 인증서 저장소로 복사할 수 있습니다.


인증서는 인증 기관이 발급하는 것으로 사용자 신분을 확인합니다. 인증서에는 데이터를 보호하거나 보안된 네트워크 연결을 하는 데 필요한 정보가 들어 있습니다. 인증서 저장소는 인증서를 저장하는 시스템 영역입니다.

저장소 위치

☐ 현재 사용자(C)

☒ 로컬 컴퓨터(L)

계속하려면 [다음]을 클릭하십시오.

 다음(N)

취소

인증서 저장소

인증서 저장소는 인증서를 저장하는 시스템 영역입니다.

Windows가 자동으로 인증서 저장소를 선택하거나 사용자가 인증서 위치를 지정할 수 있습니다.

☐ 인증서 종류를 기준으로 인증서 저장소를 자동으로 선택(U)

☒ 모든 인증서를 다음 저장소에 저장(P)

인증서 저장소:

찾아보기(R)...

인증서 저장소 선택

사용하려는 인증서 저장소를 선택하십시오(C).

- 개요
- 신뢰할 수 있는 루트 인증 기관**
- 인터넷 익스플로러 신뢰
- 중간 인증 기관
- 신뢰할 수 있는 게시자
- 신뢰할 수 없는
- 타사 루트 인증 기관

☐ 실제 저장소 표시(S)


확인

취소

다음(N)

취소



←  인증서 가져오기 마법사

인증서 가져오기 마법사 완료

[마침]을 클릭하면 인증서를 가져옵니다.

다음 설정을 지정했습니다.

사용자가 선택한 인증서 저장소 콘텐츠	신뢰할 수 있는 루트 인증 기관 인증서
-------------------------	--------------------------

마침(F)

취소

4. 브라우저 재시작 후 ProjectView 접속