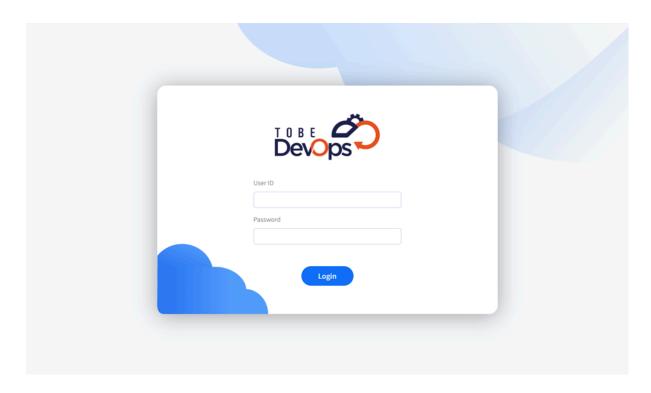
TobeDevOps Getting Started

TobeDevOps 프로젝트를 처음 접하거나 익숙하지 않은 개발자를 위한 튜토리얼입니다.

1.TobeDevOps 포털 접속하기

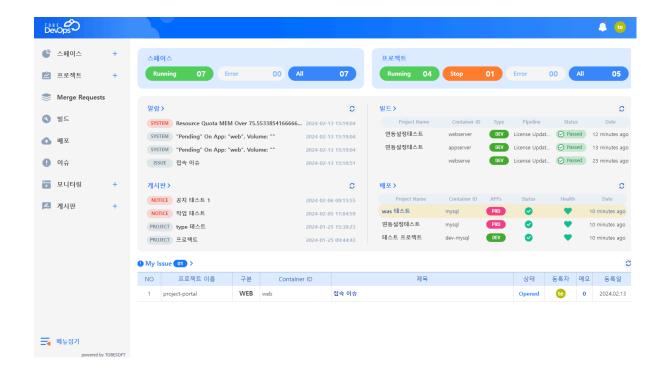
TobeDevOps는 프로젝트 구성과 CI/CD를 지원하는 포털을 제공합니다.

포털은 관리자가 생성한 계정을 부여받아 로그인할 수 있으며 권한별 사용할 수 있는 메뉴가 제한됩니다. 개발자 권한 설명은 [개발자 가이드 - 2. 사용자 권한]을 확인하세요.



포털에서 생성된 프로젝트 정보를 확인하고 소스를 다운로드 받기 위해서는멤버 권한의 계정이 필요합니다. 로그인 후 생성된 프로젝트가 보이지 않을때는 프로젝트 관리자에게 문의하세요.

포털에 로그인하면 다음과 같은 홈 화면을 확인할 수 있습니다. 홈 화면 및 메뉴의 자세한 설명은 [개발자 가이드 > 3. 메인화면] 을 확인하세요.



2. 프로젝트 정보 확인하기

프로젝트 > 프로젝트 개요 메뉴를 선택하면 프로젝트 카드 리스트를 볼 수 있습니다. 이 화면을 통해 개발자는 자신이 포함된 프로젝트를 확인합니다. 만약 프로젝트가 카드 내역에 존재하지 않으면 프로젝트 멤버로 추가가 필요합니다.

프로젝트 카드를 선택하여 상세 정보를 확인하는 프로젝트 구성 메뉴로 이동합니다.

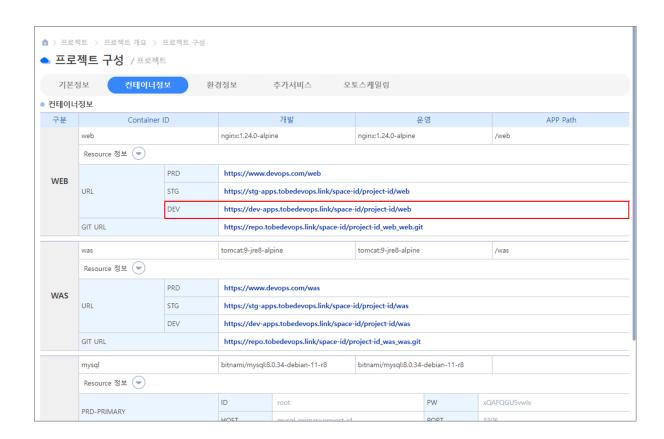


프로젝트 구성 화면의 첫번째 기본정보 탭에서는 생성된 프로젝트의 ID, 이름, 멤버 등의 기본 구성 정보를 확인할 수 있습니다.

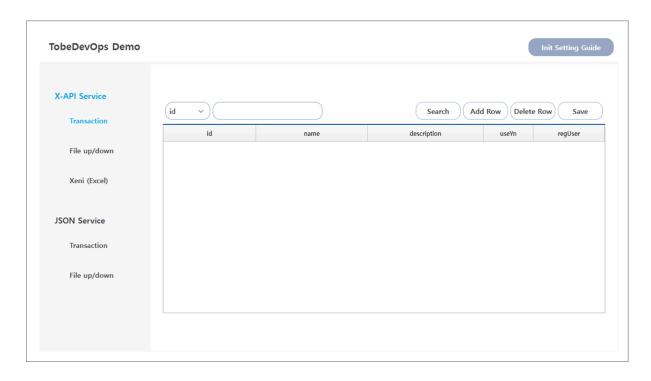


프로젝트에 구성된 컨테이너의 상세 정보 확인을 위해 컨테이너정보 탭을 클릭합니다. 넥사크로 화면은 WEB 컨테이너에서 확인합니다.

컨테이너 정보 > WEB 구분 Container ID > DEV URL 클릭 (URL 복사 및 붙여넣기 사용 가능)



WEB 컨테이너의 URL 클릭 시 새로운 브라우저 창에 넥사크로 화면이 출력됩니다. 만약 화면이 정상적으로 출력되지 않는다면 [개발자 매뉴얼 > APP 배포 현황 확인] 을 확인하세요.

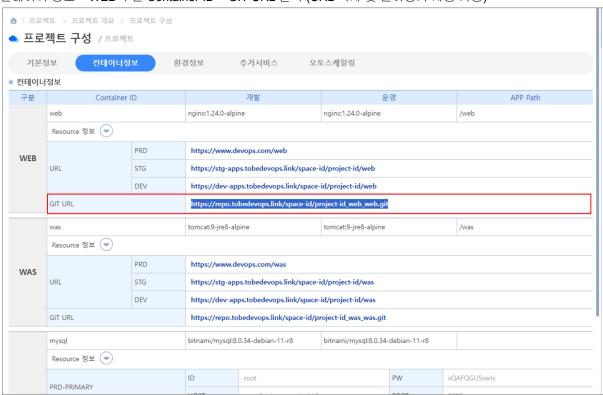


3. GitLab Access Tokens 발급하기

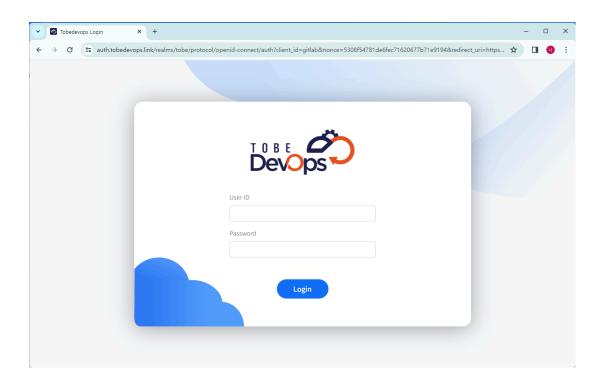
생성된 프로젝트는 컨테이너 별 GitLab(Git 저장소)에서 소스를 확인할 수 있으며 WEB 컨테이너에 넥사크로 화면 소스가 포함되어 있습니다.

또한 로컬환경에서 프로젝트 git push, pull 하기 위해 토큰을 발급받아야 합니다.

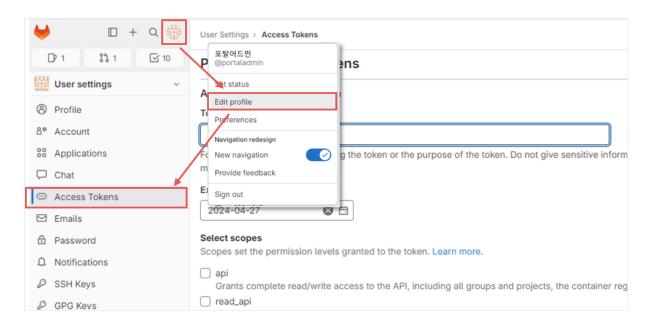
컨테이너 정보 > WEB 구분 Container ID > GIT URL 클릭 (URL 복사 및 붙여넣기 사용 가능)



복사한 GIT URL(Repo URL)을 새로운 브라우저 창에 붙여넣기 하여 GitLab 화면으로 이동합니다. 최초 접속 시 TobeDevOps Portal에서 사용한 계정 로그인이 필요하며 로그인 된 정보는 유지됩니다. (접속 정보가 변경될 경우 재 로그인이 필요합니다.)

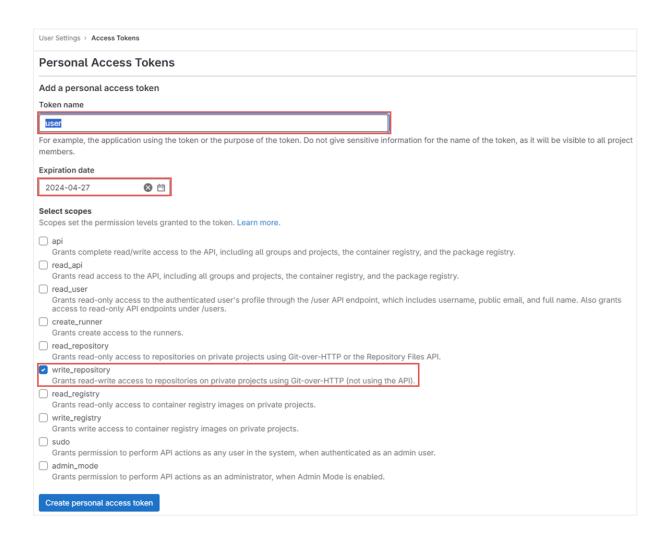


GitLab 화면 접속 후 토큰 발급을 위해 User Settings > Access Tokens 메뉴로 이동합니다.

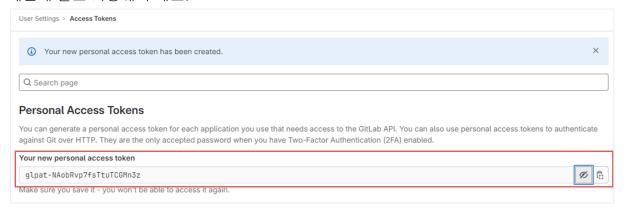


토큰 발급 화면에서 토큰 이름, 유효기간, 권한 입력 후 [Create personal access token] 버튼을 클릭합니다.

Select scope는 프로젝트 git push/pull 사용을 위한 "write_repository"를 선택해 주세요.



토큰 생성이 완료되면 발급된 토큰 값을 확인할 수 있습니다. 토큰 값은 재 발급할 수 없기 때문에 별도 저장해 주세요.



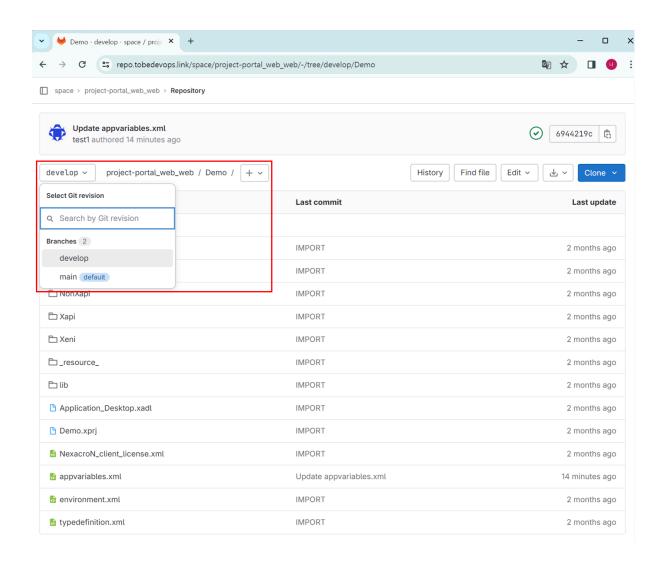
토큰 값은 로컬 개발환경 구성 시 git의 password 로 사용합니다.



4. 넥사크로 개발환경 구성하기

연결된 GitLab에는 프로젝트 별로 검증(main) 과 개발(develop) 브랜치가 존재하며 프로젝트 Member 권한의 개발자는 반드시 개발 브랜치를 사용합니다.

브랜치, 태그	사용 환경	96 H
태그 (release-*)	PRD	프로젝트 운영
브랜치 (main)	STG	프로젝트 검증
브랜치 (develop)	DEV	프로젝트 개발



git clone 하여 프로젝트 개발을 세팅할 때는 반드시 -b [브랜치 명] (특정 브랜치를 클론하는 옵션) 으로 개발 브랜치(develop Branch)를 선택합니다.

\$ git clone -b develop [REPO_URL]

```
MINGW64:/c/workspace/project-portal_web_web

tobesoft@DESKTOP-D00F609 MINGW64 /c/workspace
$ git clone -b develop https://repo.tobedevops.link/space/project-portal_web_web
.git
cloning into 'project-portal_web_web'...
remote: Enumerating objects: 1627, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 1627 (delta 2), reused 0 (delta 0), pack-reused 1618
Receiving objects: 100% (1627/1627), 33.87 MiB | 10.98 MiB/s, done.
Resolving deltas: 100% (1358/1358), done.

tobesoft@DESKTOP-D00F609 MINGW64 /c/workspace
$ cd project-portal_web_web

tobesoft@DESKTOP-D00F609 MINGW64 /c/workspace/project-portal_web_web (develop)
$
```

개발 브랜치(develop) 를 지정하지 않고 git clone 하면 자동으로 운영 브랜치(main)가 선택됩니다. 현재 브랜치가 운영 브랜치(main)일 때 git checkout 하여 반드시 개발 브랜치(develop) 로 변경 후 사용하세요.

\$git branch -a \$git checkout develop

```
** MINGW64:/c/workspace/project-portal_web_web

**Cobesoft@DESKTOP-DOOF609 MINGW64 /c/workspace

$ git clone https://repo.tobedevops.link/space/project-portal_web_web.git

Cloning into 'project-portal_web_web'...
remote: Enumerating objects: 1627, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 1627 (delta 2), reused 0 (delta 0), pack-reused 1618
Receiving objects: 100% (1627/1627), 33.87 MiB | 10.08 MiB/s, done.
Receiving objects: 100% (1537/1537), done.

**Updating files: 100% (1537/1537), done.

**tobesoft@DESKTOP-DOOF609 MINGW64 /c/workspace

$ cd project-portal_web_web

**tobesoft@DESKTOP-DOOF609 MINGW64 /c/workspace/project-portal_web_web (main)
$ git branch -a
** main
    remotes/origin/develop
    remotes/origin/develop

Switched to a new branch 'develop'
branch 'develop' set up to track 'origin/develop'.

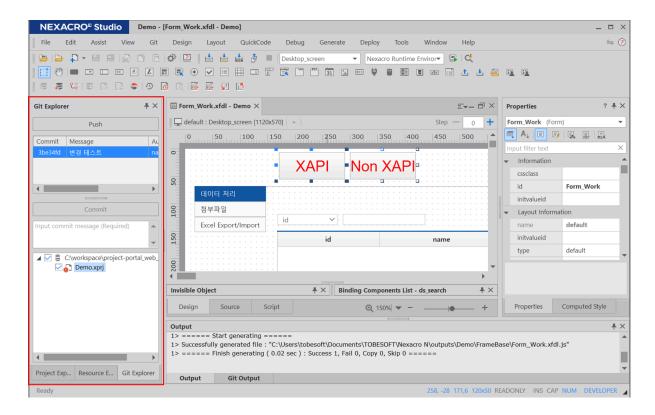
**tobesoft@DESKTOP-DOOF609 MINGW64 /c/workspace/project-portal_web_web (develop)
$ Switched to a new branch 'develop'
branch 'develop' set up to track 'origin/develop'.

**tobesoft@DESKTOP-DOOF609 MINGW64 /c/workspace/project-portal_web_web (develop)
$ Switched to a new branch 'develop'
branch 'develop' set up to track 'origin/develop'.
```

5. 넥사크로 화면 수정하기

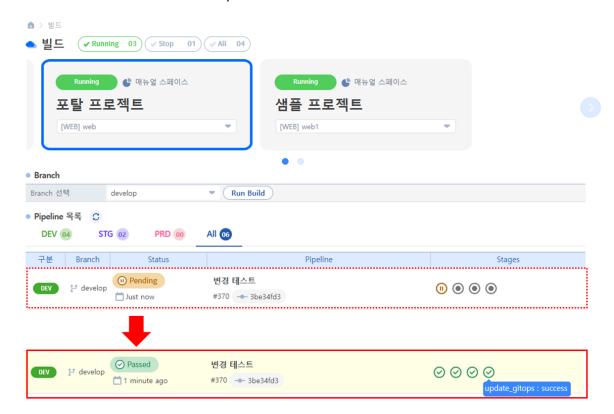
Nexacro Studio는 Git 연동을 제공하며 프로젝트 소스를 개발툴에서 형상관리할 수 있습니다.

Nexacro 화면 수정시 변경된 파일을 Git Explorer창에서 확인할 수 있으며 Commit과 Push 버튼을 사용하여 수정된 화면을 빌드/배포 할 수 있습니다.

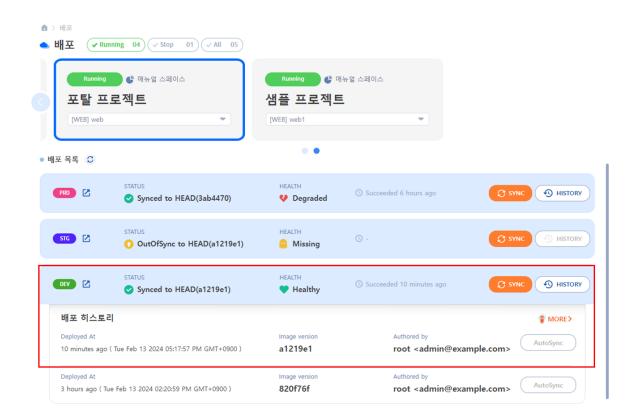


Nexacro Studio의 Git 사용 방법은 <u>온라인 매뉴얼 > 개발도구 > 넥사크로 스튜디오 > 21. 소스 코드 관리 시스템 지원 > 21.2 깃(Git)</u> 을 확인하세요.

Git Push한 프로젝트는 포털의 빌드 메뉴에서 반영중인 것을 확인할 수 있습니다. 실시간 진행 과정을 확인하기 위해서는 Pipeline 목록의 새로고침이 필요합니다.



빌드가 완료 후 자동으로 배포가 진행되며 배포목록을 새로고침하여 과정을 확인합니다. 완료된 내역은 배포 히스토리를 통해 확인할 수 있습니다.



수정된 넥사크로 파일의 빌드/배포가 완료된 후 WEB 컨테이너의 URL에 접속하여 변경된 화면을 확인합니다.

