

포팅 매뉴얼

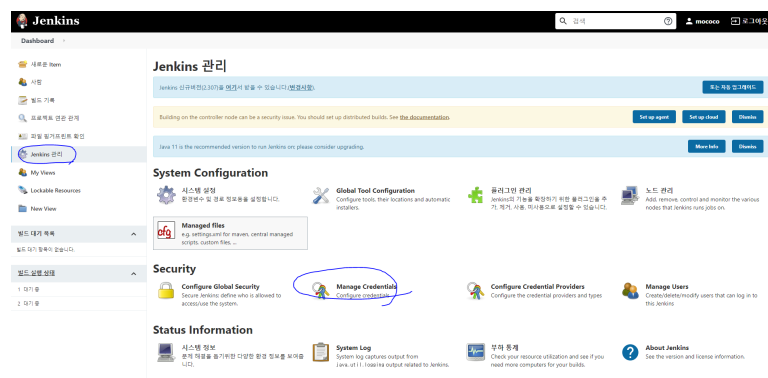
기본 설치

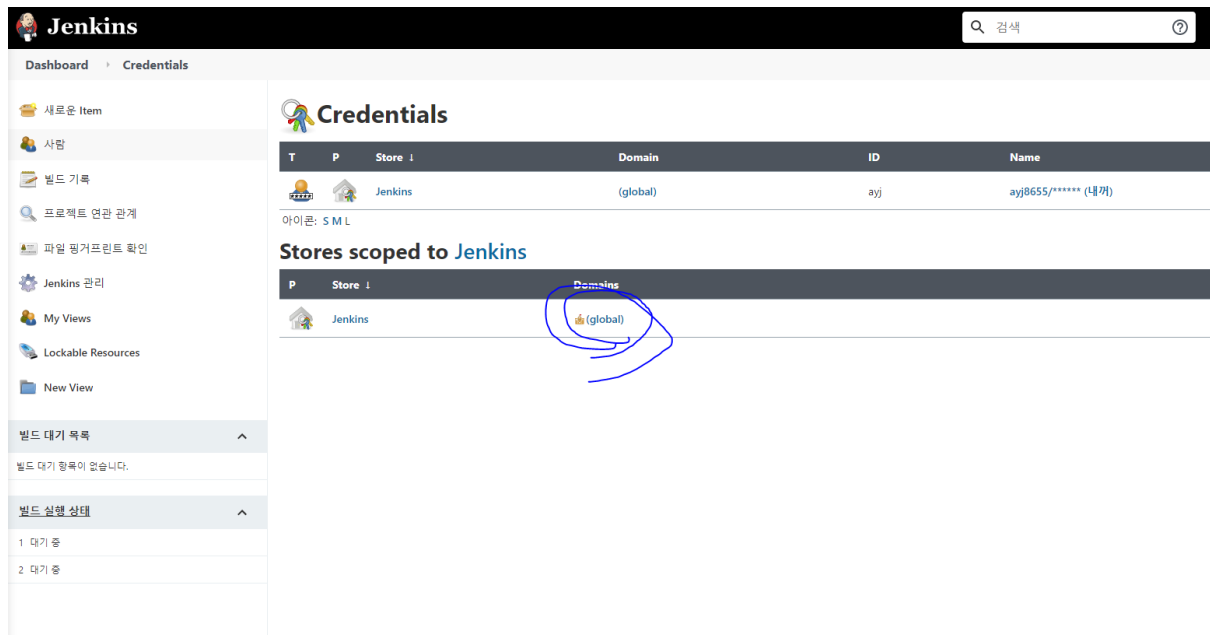
- AWS 서버에서 mysql Ver 8.0.26-0ubuntu0.20.04.2 for Linux on x86_64 ((Ubuntu)) 설치
- AWS 서버에서 node.js v10.19.0 설치
- AWS 서버에서 Python 3.7 이상 설치
- 로컬(윈도우)에서 공식문서(<https://docs.ros.org/en/eloquent/Installation.html>) 에 따라 ros2 eloquent 설치

배포 방법

- ubuntu home/jenkins_war 폴더 생성
- wget <https://get.jenkins.io/war-stable/2.303.2/jenkins.war> 실행 후 jenkins 설치
- java -jar jenkins.war 실행하여 <http://j5b202.p.ssafy.io:8080/> 로 접속
- `vi /var/lib/jenkins/secrets/initialAdminPassword` 입력하여 비밀번호 확인 후 접속한 페이지에 입력하고 기본 플러그인 설정 및 계정 설정

이후 로그인하고 메인화면에서 깃 계정 설정을 위해 Manage Credentials 클릭 이후 global 선택





Add Credentials

Dashboard > Credentials > System > Global credentials (unrestricted)

Back to credential domains

Add Credentials

Kind

Username with password

Scope

Global (Jenkins, nodes, items, all child items, etc)

Username

ayj8655

☐ Treat username as secret

Password

.....

ID

ayj-test

Description

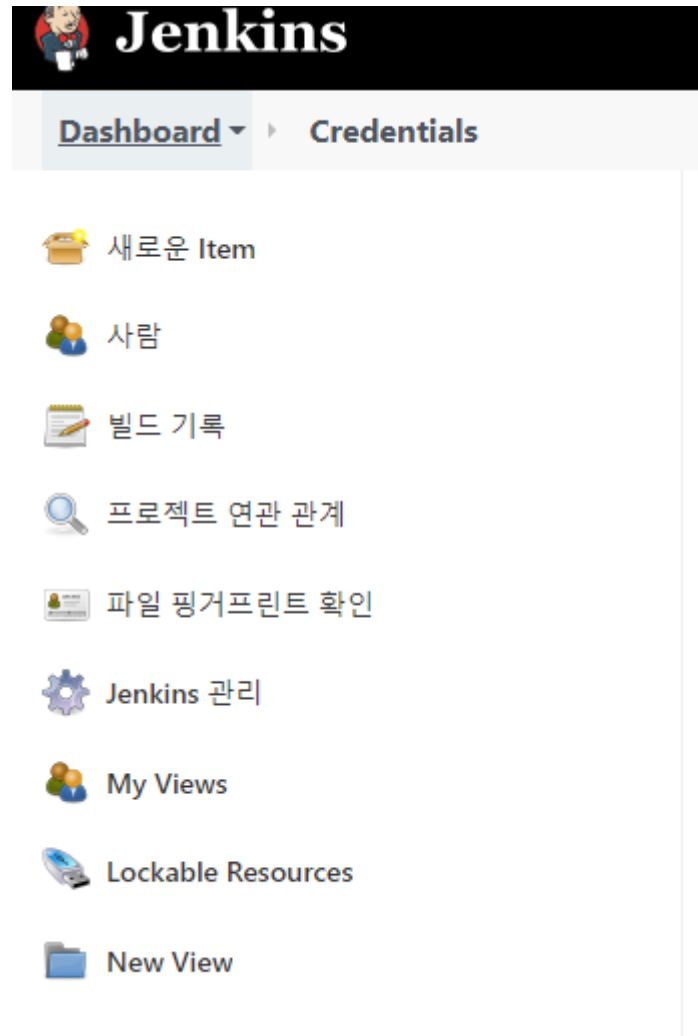
OK

Username = git 아이디

Password = git 비밀번호

ID = 젠킨스에 저장될 키 이름

새로운 Item → 이름정하고 Freestyle project 이후 OK



Enter an item name

» Required field



Freestyle project

이것은 Jenkins의 주요 기능입니다. Jenkins은 어



Pipeline

Orchestrates long-running activities that can sp



Multi-configuration project

다양한 환경에서의 테스트, 플랫폼 특성 빌드, 기



Folder

Creates a container that stores nested items in i
as they are in different folders.



GitHub Organization

Scans a GitHub organization (or user account) f



Multibranch Pipeline

Creates a set of Pipeline projects according to c

If you want to create a new item from other existin



Copy from

OK

소스 코드 관리에서 깃 URL 입력시 아래 에러 발생시 Credentials에서 아까 만든 키 설정

General **소스 코드 관리** 빌드 유발 빌드 환경 Build 빌드 후 조치

☐ 이 빌드는 매개변수가 있습니다
☐ 빌드 안함
☐ 필요한 경우 concurrent 빌드 실행

소스 코드 관리

☐ None
☒ Git


Repositories

Repository URL

`https://lab.ssafy.com/s05-webmobile2-sub3/S05P13B203`

Failed to connect to repository : Command "git ls-remote -h -- https://lab.ssafy.com/s05-webmobile2-sub3/S05P13B203 HEAD" returned status code 128:
stdout:
stderr: remote: HTTP Basic: Access denied
fatal: Authentication failed for 'https://lab.ssafy.com/s05-webmobile2-sub3/S05P13B203.git/'

Credentials

- none - 

소스 코드 관리


☐ None
☒ Git

Repositories

Repository URL

`https://lab.ssafy.com/s05-webmobile2-sub3/S05P13B203`

Credentials

`avj8655/***** (내꺼)` 

Branches to build

빌드 환경에서 npm 사용하기 위해 체크

(플러그인에서 Nodejs 설치하고 젠킨스 관리에서 Global Tool Configuration 에서 NodeJS 추가)

빌드 환경

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Provide Configuration files
- ☐ Send files or execute commands over SSH before the build starts
- ☐ Send files or execute commands over SSH after the build runs
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build scans
- ☒ Provide Node & npm bin/ folder to PATH

NodeJS Installation

nodejs

Specify needed nodejs installation where npm installed packages will be provided to the PATH

npmrc file

- use system default -

Cache location

Default (~/.npm or %APP DATA%\npm-cache)

- ☐ With Ant

NodeJS

NodeJS installations

Add NodeJS

NodeJS

Name

nodejs

- ☒ Install automatically

Install from nodejs.org

Version

NodeJS 15.6.0

- ☐ Force 32bit architecture

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'

Global npm packages refresh hours

72

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

Delete Installer

Add Installer

Delete NodeJS

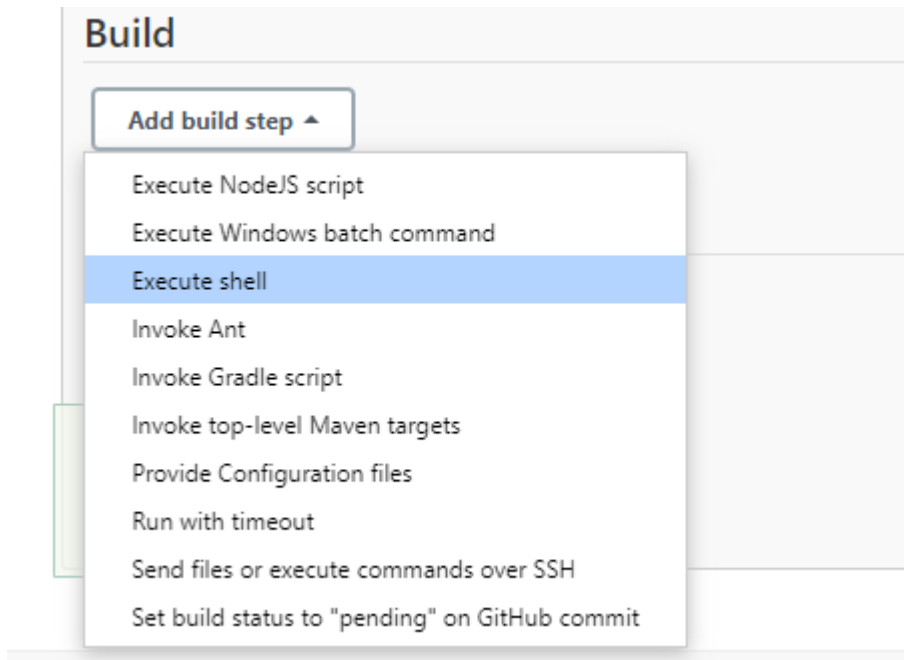
Add NodeJS

List of NodeJS installations on this system

Save

Apply

빌드 명령어 입력 위해 Execute Shell 추가



- excute shell

```
cd ~/jenkins_home/workspace/kjw/Backend
npm install
nohup npm start &
```

- 이후 설정한 깃랩 레포지터리에 커밋하면 서버에 자동 배포가 된다.

배포 특이사항

- 서버 연결중 socketio 버전 문제 발생시 로컬-서버 버전 가이드

- python-socketio 5.4.0
- websocket-client 1.2.1
- node.js 14.17.3
- socketio 4.2.0

시뮬레이터 실행 메뉴얼

- SSAFY Simulator 설치
- 로그인 후 네트워크 세팅 확인(기본 세팅 연결이 잘 되었는지)
-

시뮬레이터는 서버와 소켓 통신하며, 앱에서 요청한 사항을 서버에서 데이터와 함께 전달해 주면 그에 맞는 동작을 수행한다. 이를 위해 해당 동작에 필요한 .py 파일을 사전에 실행해 놓아야 한다. 다음은 사용하고자 하는 동작에 따른 ros 이용 메뉴얼이다.

클론받은 폴더의 경로가 다음과 같을 때, 우선 파이썬 파일을 빌드해야 한다.

```
C:\Users\multicampus\Desktop\S05P21B202\ros2_smart_home
```

```
call C:\dev\ros2_elloquent\setup.bat // ros2_elloquent 설치 경로
cd C:\Users\multicampus\Desktop\S05P21B202\ros2_smart_home
colcon build
```

각 ros2 run이나 launch, .py 파일 실행은 모두 명령 프롬프트를 켜 후 다음 과정을 사전에 거쳐야 한다.

```
call C:\dev\ros2_elloquent\setup.bat
call C:\Users\multicampus\Desktop\S05P21B202\ros2_smart_home\install\local_setup.bat
```

1. 청소 실행

모든 명령어는 새로운 명령 프롬프트에서 실행

```
ros2 run final load_map
ros2 run final odom
ros2 run final a_star
```



```
ros2 run final a_star_local_path
ros2 run final path_tracking
ros2 run final cleaning2
```

2. 가전 제어

```
ros2 run final load_map
ros2 run final odom
ros2 run final a_star
ros2 run final a_star_local_path
ros2 run final path_tracking
ros2 run final appliance_control2
```

3. 사물 인식(침입, 분실물 확인)

```
ros2 run final load_map
ros2 run final odom
ros2 run final a_star
ros2 run final a_star_local_path
ros2 run final path_tracking
ros2 run sub3 tf_detector
```

4. 순찰 모드

```
ros2 run final load_map
ros2 run final odom
ros2 run final a_star
ros2 run final a_star_local_path
ros2 run final path_tracking
ros2 run final patrol
```

각 모드의 마지막 명령어는 동시 실행이 가능하다. 구체적인 명령을 하기 위해서는 앱과 서버를 연결한 후 앱에서 명령을 보내야 한다.

App 실행 메뉴얼

```
git clone https://lab.ssafy.com/s05-iot-ctrl/S05P21B202.git
cd S05P21B202/ssavis2
```

```
npm install
```

```
expo start
```

데이터베이스 접속 정보

- database: ssafy_app_db
- ID: root
- pwd: 1q2w3e4r1!