

Dev, Back-End/CI & CD

[JenKins] ③ 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS)

TROLL's | 2021. 6. 2. 22:31

③
JenKins를 이용해서
node.js CI/CD
자동 배포하는 방법
(NestJS)

JenKins를 이용해서 node.js 배포하는 방법 (NestJS) 2편에 이어서

3편을 포스팅하겠습니다.

▼ Jenkins를 통해 자동배포 (NodeJS) 시리즈

- ▶ ① [젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 - 바로가기](#)
- ▶ ② [젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 - 바로가기](#)
- ▶ ③ [젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 - 바로가기](#)
- ▶ ④ [젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 - 바로가기](#)
- ▶ ⑤ [젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 - 바로가기](#)



오늘은 GitHub에서 push하면 자동배포할 수 있는 GitHub Repository를 생성하고 NestJS 웹 스크립트 작성을 해주겠습니다.

※ 최종 GitHub 주소 : <https://github.com/choseongho93/Jenest>

● GitHub에서 레포지토리 생성 & local에서 NestJS 설치 / 구동

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *



choseongho93 ▾

Repository name *

Great repository names are short and memorable. Need inspiration? How about **animated-computing-machine**?

Description (optional)



Public

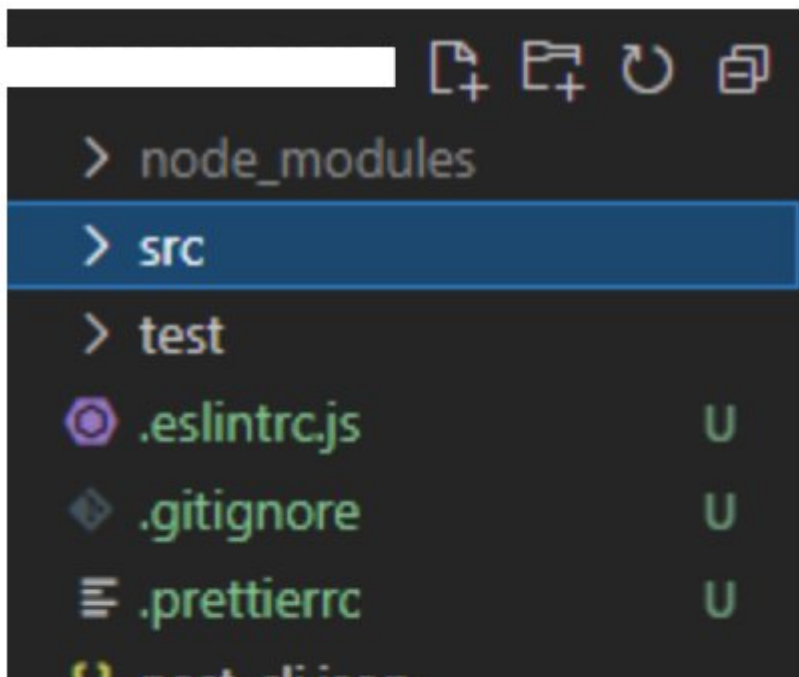
Anyone on the internet can see this repository. You choose who can commit.

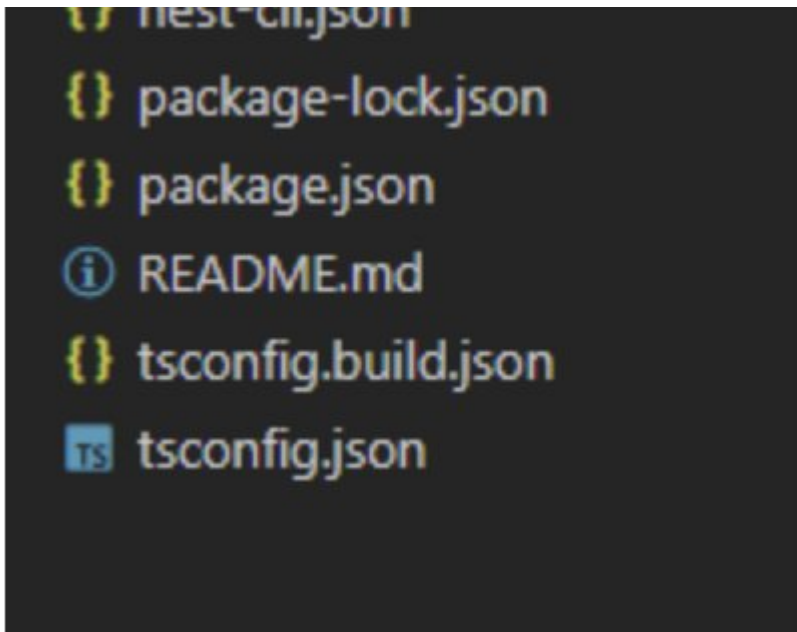


Private

You choose who can see and commit to this repository.

1. 깃헙에서 레포지토리를 생성해주고 저는 ovni라는 이름으로 생성해주겠습니다.
(현재는 Jenest라는 프로젝트로 변경됨을 알려드립니다.)

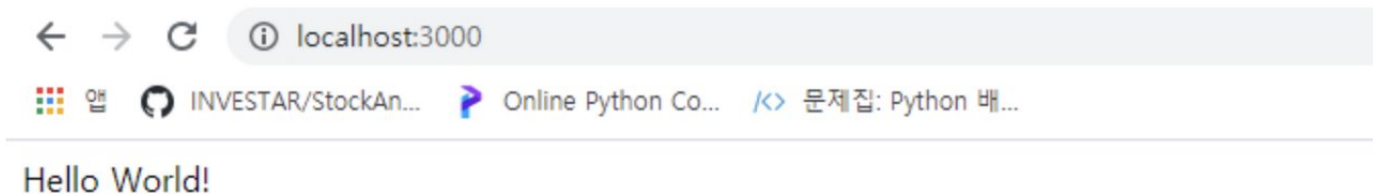






2. 제 로컬에 위 깃헙을 Clone해와서 NestJS를 설치해줍니다. (이때 디렉토리 경로는 여러분들이 원하는 곳에 하시면 됩니다!)

★ NestJS 설치에 대해서는 아래를 참고해주세요. ★

▶ [\[NestJS\] ① nestjs-API 간단하게 설치하는 방법- 바로가기](#)



3. NestJS 기본 포트인 3000로 브라우저를 통해 로컬에 접속해서 위 사진과 동일하게 접속된다면 정상입니다.
(바로 위 ★ NestJS 설치에 대해서는 아래를 참고해주세요. ★ 를 반드시 하고 오셔야 정상적으로 작동되고, 이 후에 이어지는 실습이 가능합니다.)

 choseongho93 Update README.md		93dda6d 22 minutes ago	 31 commits
script	배포		2 days ago
src	redploy		2 days ago
test	nestjs default		2 months ago
.gitignore	.gitignore update		4 days ago
Dockerfile	docker add		4 days ago
README.md	Update README.md		22 minutes ago
ecosystem.config.js	jenkins setting		2 days ago
nest-cli.json	nestjs default		2 months ago
package-lock.json	nestjs default		2 months ago
package.json	nestjs default		2 months ago
tsconfig.build.json	nestjs default		2 months ago
tsconfig.json	nestjs default		2 months ago

4. 로컬에서 정상 작동되는 것을 확인하고 깃헙에서 만든 레파지토리에 푸시해줍니다.

• • •

● AWS EC2 노드 앱 서버에 올리기

이전에 1편에서 구축한 Node 서버에 위 코드들을 모두 올려주는 작업을 하겠습니다.

```
[ubuntu@ip-172-31-30-100 ~]$ cd /home
ubuntu@ip-172-31-30-100 ~$
```

\$ cd /home

5. 서버 SSH로 접속한 후에 home 디렉토리로 이동합니다.

```
[ubuntu@ip-172-31-30-100 ~]$ sudo chmod 755 /home
[ubuntu@ip-172-31-30-100 ~]$ ls -ld /home
total 72
drwxr-xr-x 19 root root 4096 May 18 12:50 .
drwxr-xr-x 19 root root 4096 May 18 12:50 ..
lrwxrwxrwx 1 root root    7 Apr 30 23:15 bin -> usr/bin
drwxr-xr-x 3 root root 4096 Apr 30 23:37 boot
drwxr-xr-x 15 root root 3160 May 18 12:50 dev
drwxr-xr-x 92 root root 4096 May 18 12:50 etc
drwxr-xr-x 3 root root 4096 May 18 12:50 home
lrwxrwxrwx 1 root root    7 Apr 30 23:15 lib -> usr/lib
lrwxrwxrwx 1 root root    9 Apr 30 23:15 lib32 -> usr/lib32
lrwxrwxrwx 1 root root    9 Apr 30 23:15 lib64 -> usr/lib64
lrwxrwxrwx 1 root root   10 Apr 30 23:15 libx32 -> usr/libx32
drwx----- 2 root root 16384 Apr 30 23:30 lost+found
drwxr-xr-x 2 root root 4096 Apr 30 23:15 media
drwxr-xr-x 2 root root 4096 Apr 30 23:15 mnt
drwxr-xr-x 2 root root 4096 Apr 30 23:15 opt
dr-xr-xr-x 150 root root    0 May 18 12:49 proc
drwx----- 4 root root 4096 May 18 12:50 root
drwxr-xr-x 23 root root 800 May 18 12:51 run
lrwxrwxrwx 1 root root    8 Apr 30 23:15 sbin -> usr/sbin
drwxr-xr-x 7 root root 4096 Apr 30 23:36 snap
drwxr-xr-x 2 root root 4096 Apr 30 23:15 srv
dr-xr-xr-x 13 root root    0 May 18 12:49 sys
drwxrwxrwt 11 root root 4096 May 18 12:51 tmp
drwxr-xr-x 15 root root 4096 Apr 30 23:22 usr
drwxr-xr-x 13 root root 4096 Apr 30 23:26 var
[ubuntu@ip-172-31-30-100 ~]$ sudo chown ubuntu:ubuntu /home/ubuntu -R
[ubuntu@ip-172-31-30-100 ~]$ sudo chmod 700 /home/ubuntu /home/ubuntu/.ssh
[ubuntu@ip-172-31-30-100 ~]$ sudo chmod 600 /home/ubuntu/.ssh/authorized_keys
```

\$ sudo chmod 755 /home

\$ sudo chown ubuntu:ubuntu /home/ubuntu -R

\$ sudo chmod 700 /home/ubuntu /home/ubuntu/.ssh

\$ sudo chmod 600 /home/ubuntu/.ssh/authorized keys

6. 이 작업은 home 디렉토리 아래에 각종 소유권 권한을 부여해주기 위해 작업하였습니다.

(이 작업을 하는 이유는 home 디렉토리 아래에 깃 클론을 하기 위해 디렉토리를 생성하려다보니 권한 문제가 발생하여 찾은 방법입니다.)


```
[ubuntu@ip- /]$ sudo chown -R -v ubuntu /home
ownership of '/home/ubuntu/.cache/motd.legal-displayed' retained as ubuntu
ownership of '/home/ubuntu/.cache' retained as ubuntu
ownership of '/home/ubuntu/.bashrc' retained as ubuntu
ownership of '/home/ubuntu/.profile' retained as ubuntu
ownership of '/home/ubuntu/.ssh/authorized_keys' retained as ubuntu
ownership of '/home/ubuntu/.ssh' retained as ubuntu
ownership of '/home/ubuntu/.sudo_as_admin_successful' retained as ubuntu
ownership of '/home/ubuntu/.bash_logout' retained as ubuntu
ownership of '/home/ubuntu' retained as ubuntu
changed ownership of '/home' from root to ubuntu
```

\$ sudo chown -R -v ubuntu /home

7. 이 작업도 5번과 동일합니다.

```
$ cd /home/ubuntu
```

8. /home/ubuntu 디렉토리로 이동해줍니다.

(스크린샷이 존재하지 않아 텍스트로 대체합니다.)

```
[ubuntu@ip- /home]$ git clone https://github.com/choseongho93/ovni.git
Cloning into 'ovni'...
remote: Enumerating objects: 19831, done.
remote: Counting objects: 100% (19831/19831), done.
remote: Compressing objects: 100% (14065/14065), done.
remote: Total 19831 (delta 4822), reused 19823 (delta 4814), pack-reused 0
Receiving objects: 100% (19831/19831), 32.00 MiB | 16.21 MiB/s, done.
Resolving deltas: 100% (4822/4822), done.
[ubuntu@ip- /home]$ ls -al
total 16
```

```
drwxr-xr-x  4 ubuntu root    4096 May 18 13:03 .
drwxr-xr-x 19 root   root    4096 May 18 12:50 ..
drwxrwxr-x  5 ubuntu ubuntu 4096 May 18 13:04 ovni
```

\$ git clone http GitHub 주소

\$ ls -al

8. GitHub에 있는 코드들을 clone해서 받아주고 정상적으로 받아졌는지 확인해줍니다.

(위 사진에서는 /home 디렉토리 경로로 되어있지만, 추후 문제가 발생해서 /home/ubuntu로 해당 코드들이 이전하였기에 현재 경로는 무시하셔도 됩니다.)

```
ubuntu@ip-: $ curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
## Installing the NodeSource Node.js 14.x repo...

## Populating apt-get cache...
+ apt-get update
Hit:1 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 https://deb.nodesource.com/node_12.x focal InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
## Confirming "focal" is supported...
+ curl -sLf -o /dev/null 'https://deb.nodesource.com/node_14.x/dists/focal/Release'
## Adding the NodeSource signing key to your keyring...
+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | gpg --dearmor | tee /usr/share/keyrings/nodesource.gpg >/dev/null
## Creating apt sources list file for the NodeSource Node.js 14.x repo...
+ echo 'deb [signed-by=/usr/share/keyrings/nodesource.gpg] https://deb.nodesource.com/node_14.x focal main' > /etc/apt/sources.list.d/nodesource.list
+ echo 'deb-src [signed-by=/usr/share/keyrings/nodesource.gpg] https://deb.nodesource.com/node_14.x focal main' >> /etc/apt/sources.list.d/nodesource.list
```

\$ curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -

9. nodejs를 설치하기 위해 PPA를 추가합니다.

```
ubuntu@ip-: $ sudo apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 22 not upgraded.
Need to get 24.8 MB of archives.
After this operation, 120 MB of additional disk space will be used
```



```

After this operation, 128 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_14.x focal/main amd64 nodejs amd64 14.17.0-deb-1nodesource1 [24.8 MB]
Fetched 24.8 MB in 1s (40.8 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 60149 files and directories currently installed.)
Preparing to unpack .../nodejs_14.17.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (14.17.0-deb-1nodesource1) ...
Setting up nodejs (14.17.0-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-
v14.17.0
$ node -v
ubuntu@ip-
6.14.13
$ npm -v
ubuntu@ip-
$

```

\$ sudo apt install nodejs

\$ node -v

\$ npm -v

10. nodejs를 설치하고 확인해줍니다.

```

ubuntu@ip-
$ ssh-keygen -t rsa -b 4096 -C "
Generating public/private rsa key pair:
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:

```

```

ubuntu@ip-
/home/ovni$ cat ~/.ssh/id_rsa.pub

```

\$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

\$ cat ~/.ssh/id_rsa.pub

11. 이메일 부분은 깃헙 로그인했을때 계정인 이메일을 입력해줍니다.

입력하고 나서 계속 그냥 디폴트로 되도록 Enter를 눌러줍니다.

그리고 cat ~/.ssh/id_rsa.pub 명령어를 통해 SSH key를 이용하기 위해서 명령어를 입력하고 나온 보라색 박스부분을

모두 드래그해서 복사합니다.

Account settings
Profile
Account
Appearance New
Account security
Billing & plans
Security log
Security & analysis
Emails
Notifications
Scheduled reminders
SSH and GPG keys
Repositories
Packages
Organizations
Saved replies
Applications

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



Delete



Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

[New GPG key](#)

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode Beta

☐ Flag unsigned commits as unverified

12. 깃헙에 들어가서 상단 우측에 있는 계정을 클릭하고 Setting을 클릭해서 페이지 이동을 합니다.

이동된 페이지는 위 사진과 같으며 좌측 메뉴에 "SSH and GPG keys"를 클릭하여 "New SSH key"버튼을 클릭 합니다.

Account settings
Profile
Account
Appearance New
Account security
Billing & plans
Security log

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Security & analysis

Emails

Notifications

Scheduled reminders

SSH and GPG keys

Add SSH key

13. 타이틀은 본인이 알수있도록 하는 용도이기에 적당하게 입력하고, key부분에 11번에서 복사한 값을 붙여주고 "Add SSH key" 버튼을 클릭하여 등록합니다.
(만일 위에서 node서버에서 깃 클론이 안되었다면, SSH key를 등록한 이 시점부터 다시 클론을 시도해봅니다.)

젠킨스로 CI/CD하는 방법에 대해 다음에 이어서 진행하겠습니다.

Ref: <https://soojae.tistory.com/25>
<https://12340zszs.tistory.com/33>
<https://cheese10yun.github.io/PM2/>

블로그 내용이 도움이 되셨다면 아래 광고를 클릭해주시면 감사하겠습니다. ^^

♡ 1

🔗

...

구독하기

🔍

📺 카카오토리

🐦 트위터

📘 페이스북

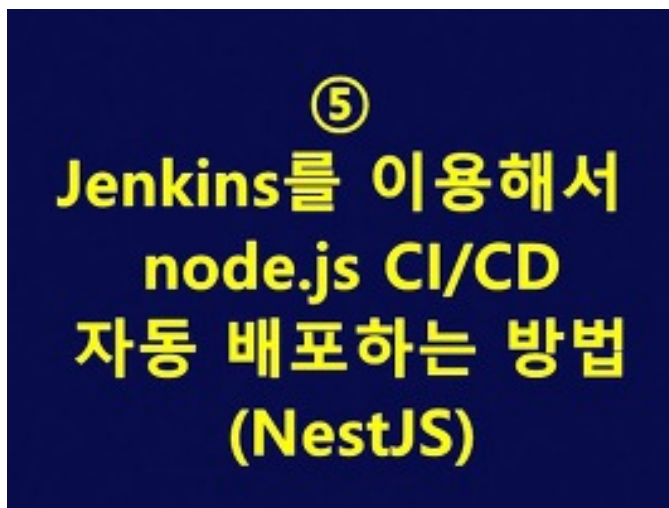
'Dev. Back-End' > CI & CD' 카테고리의 다른 글

[Jenkins] job scheduling 설정하는 방법 (0)	2021.07.24
[Jenkins] Global 환경변수 등록하는 방법 (0)	2021.07.23
Jenkins를 Docker Container로 구축시에 TimeZone 설정하는 방법 (0)	2021.07.21

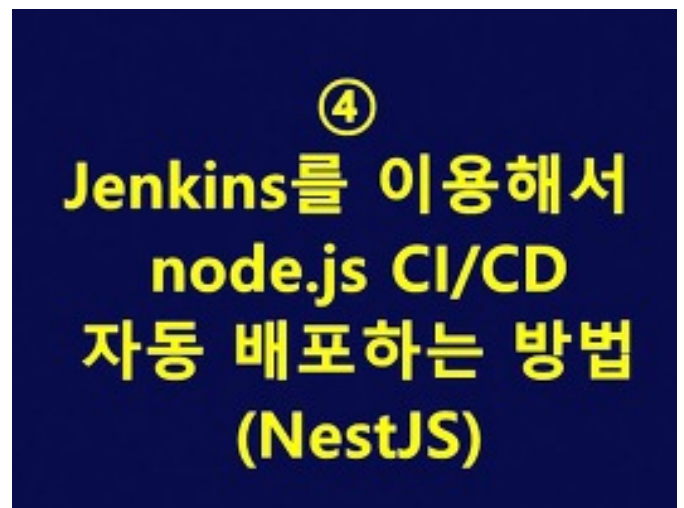
[Jenkins] docker를 이용해서 jenkins 설치하는 방법 (ubuntu) (0)	2021.07.11
[JenKins] ⑤ 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS) (0)	2021.06.04
[JenKins] ④ 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS) (0)	2021.06.03
<u>[JenKins] ③ 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS) (0)</u>	2021.06.02
[JenKins] ② 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS) (0)	2021.06.01
[JenKins] ① 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS) (0)	2021.05.29

태그 #action, #ci/cd, #clone, #github, #jenkins, #NestJS, #node, #npm, #Typescript,
#자동배포

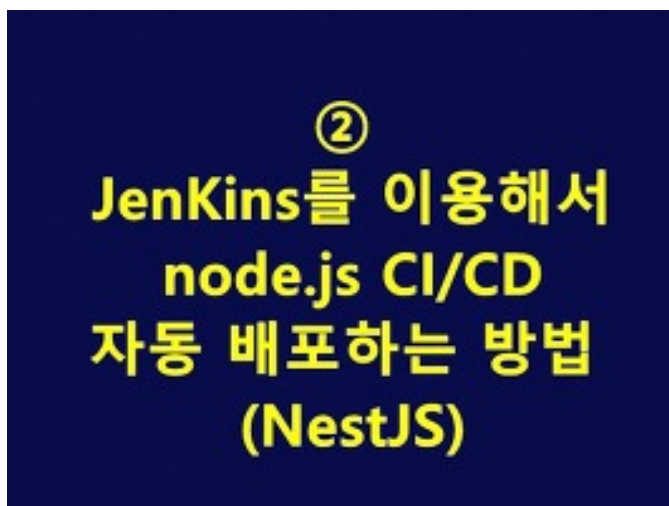
'Dev. Back-End/CI & CD' Related Articles



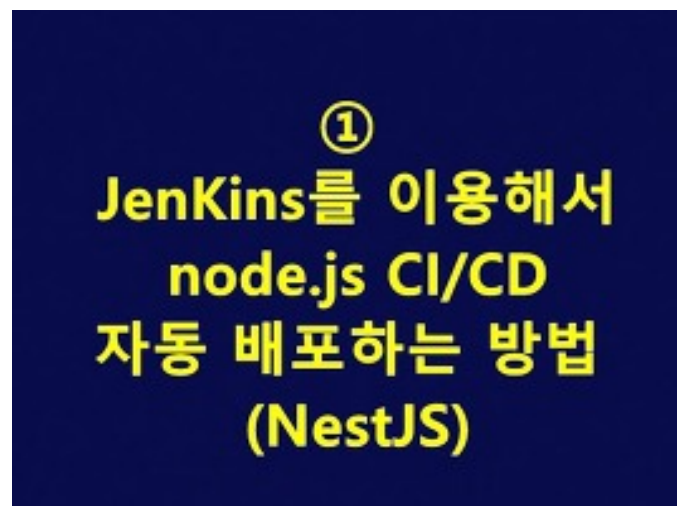
[JenKins] ⑤ 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS)



[JenKins] ④ 젠킨스를 이용해서 node.js CI/CD 자동 배포하는 방법 (NestJS)



[JenKins] ② 젠킨스를 이용해서 node.js CI/CD



[JenKins] ① 젠킨스를 이용해서 node.js CI/CD

이름

암호

☐ Secret

여러분의 소중한 댓글을 입력해주세요.

댓글달기