

2 Docker 빌드를 위한 Jenkins CI 활용

03 Jenkins 소개와 설치

Jenkins 소개

03

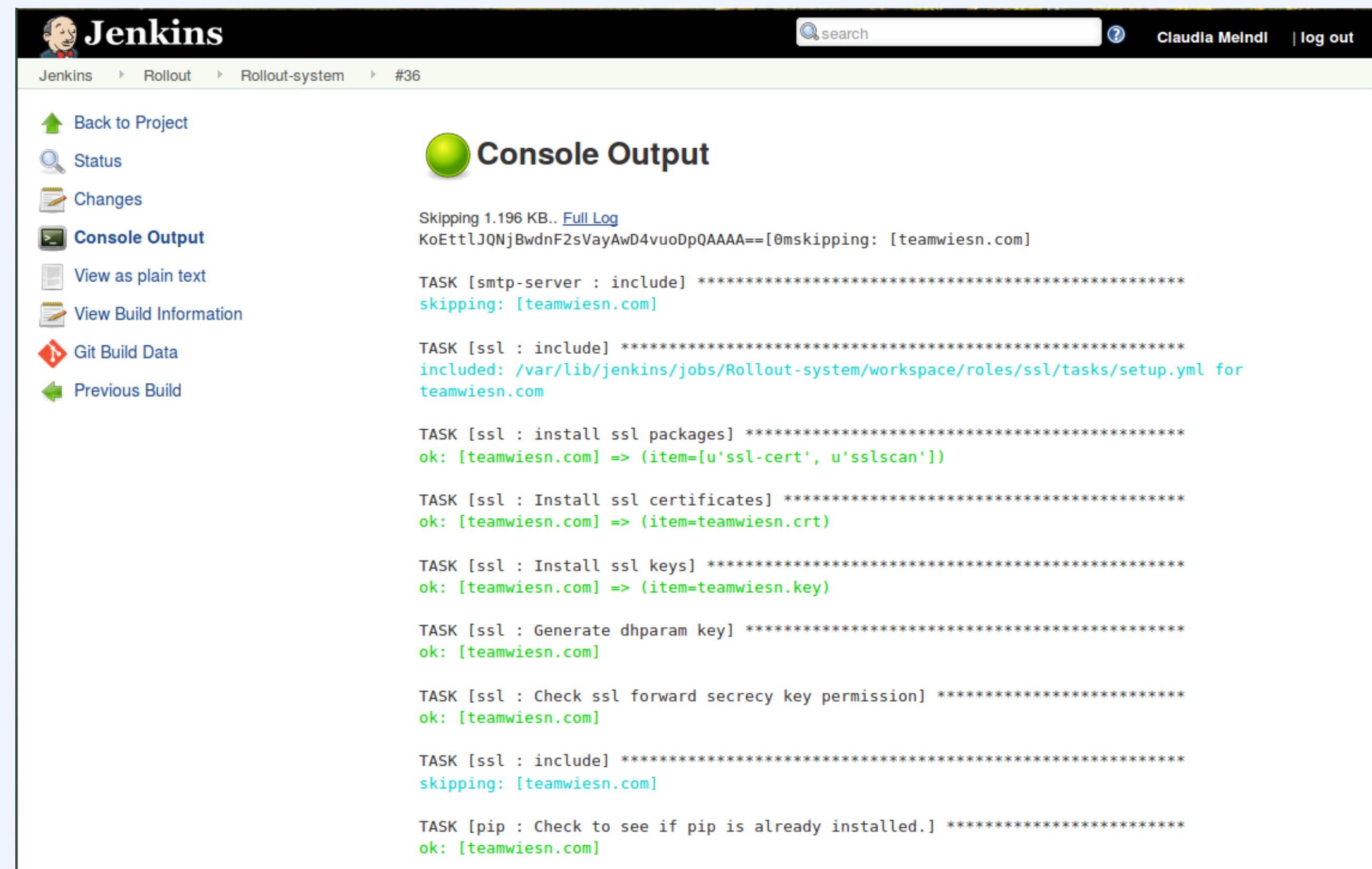
Jenkins 소개와 설치

젠킨스(Jenkins)는 소프트웨어 개발 시 **지속적 통합**(Continuous Integration) 서비스를 제공하는 툴이다. 다수의 개발자들이 하나의 프로그램을 개발할 때 버전 충돌을 방지하기 위해 각자 작업한 내용을 공유 영역에 있는 Git등의 저장소에 빈번히 업로드함으로써 **지속적 통합**이 가능하도록 해 준다.

출처 : [https://ko.wikipedia.org/wiki/%EC%A0%A0%ED%82%A8%EC%8A%A4_\(%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4\)](https://ko.wikipedia.org/wiki/%EC%A0%A0%ED%82%A8%EC%8A%A4_(%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4))

Jenkins 특징


03 Jenkins 소개와 설치





출처 : <https://commons.wikimedia.org/wiki/File:Ansible-playbook-output-jenkins.png>


플러그인


03 Jenkins 소개와 설치


 **Jenkins**

 admin  로그아웃

[Dashboard](#) > [Plugin Manager](#)

 대시보드로 돌아가기

 Jenkins 관리

 Update Center

Plugin Manager

업데이트된 플러그인 목록

설치 가능

설치된 플러그인 목록

고급

이름 ↓	사용가능
<div>Ant Plugin 1.13</div> <div>Adds Apache Ant support to Jenkins</div> <div>Report an issue with this plugin</div>	<div><div></div><div></div></div>
<div>Apache HttpComponents Client 4.x API Plugin 4.5.13-1.0</div> <div>Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.</div> <div>Report an issue with this plugin</div> <div>This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.</div>	<div><div></div><div></div></div>
<div>Bootstrap 4 API Plugin 4.6.0-3</div> <div>Provides Bootstrap 4 for Jenkins plugins.</div> <div>Report an issue with this plugin</div>	<div><div></div><div></div></div>
<div>Bootstrap 5 API Plugin 5.1.3-4</div> <div>Provides Bootstrap 5 for Jenkins plugins.</div> <div>Report an issue with this plugin</div>	<div><div></div><div></div></div>
<div>bouncycastle API Plugin 2.25</div> <div>This plugin provides a stable API to Bouncy Castle related tasks.</div> <div>Report an issue with this plugin</div>	<div><div></div><div></div></div>
<div>Branch API Plugin 2.7.0</div> <div>This plugin provides an API for multiple branch based projects.</div> <div>Report an issue with this plugin</div>	<div><div></div><div></div></div>

실습 진행사항 목록

03

Jenkins 소개와 설치

Jenkins 설치 실습

- Jenkins 설치
- 플러그인 설치
- 인증 설정

실습 사전 준비사항 #1

03

Jenkins 소개와 설치

AWS Resource 생성

(1) AWS EC2 VM

- Jenkins 서버용 AWS EC2 VM 생성 및 접근 허용 (이름: **test-jenkins**, **Keypair 생성**)
 - **Instance Type** : **t3.medium** (2Core/4GB Mem/8GB EBS)
 - **Security Group Inbound Rule** : **22**(SSH), **8080**(Jenkins)

실습 사전 준비사항 #2

- 실습환경 OS :
 - . 로컬 PC : MacOS 혹은 Windows
 - . AWS EC2 VM : Ubuntu Linux 18.04
- 통합개발환경설치 : Visual Studio로 설치
- 개발언어설치 : JAVA 11버전으로 로컬PC 및 실습 AWS EC2 VM내 설치
- Docker 설치 버전 :
 - . 로컬 PC : 20.10.12 이상 (Desktop)
 - . AWS EC2 VM : 18.09 이상 (Daemon)
- github Repo 생성
 - . 실습 예제코드의 Repository를 Fork(복제)후 개인 Repository에서 사용
<https://github.com/DevOpsRunbook/FastCampus>

실습 진행사항1

03

Jenkins 소개와 설치

Jenkins 설치 실습

- Jenkins 컨테이너 실행
- Jenkins 접속 및 설치
 - <http://<VM 퍼블릭 주소>:8080>
- 관리 계정 생성
 - [admin](#) 으로 로그인 뒤 [test](#) 계정 생성

Jenkins 설치 명령어

Jenkins 설치 명령어 목록

- Jenkins 컨테이너 실행 명령어 :
\$ `docker run --name jenkins -d -p 8080:8080 -v ~/jenkins:/var/jenkins_home -u root jenkins/jenkins:latest`
- Admin Password 확인 명령어 :
\$ `docker exec -it jenkins bash -c "cat /var/jenkins_home/secrets/initialAdminPassword"`

실습 진행사항2

03

Jenkins 소개와 설치

플러그인 설치 실습

- Jenkins DSL 플러그인 설치
- Jenkins Pipeline 플러그인 설치
- Github 플러그인 설치
- Docker 플러그인 설치
- AWS 플러그인 설치
- SSH 플러그인 설치

실습 진행사항3

03

Jenkins 소개와 설치

인증 설정 실습

- SSH Key 생성 (로컬)
- Github 인증 설정 (ssh-key)
- AWS 인증 설정 (aws-key)
- 배포 서버 인증 설정 (deploy-key)

로컬에서 Git Repository 연동을 위한 SSH Key 생성 #1

- 로컬에서 pem 키 전용 디렉토리 생성 및 이동

```
$ mkdir -p ~/pems/jenkins
```

```
$ cd ~/pems/jenkins
```

- pems 디렉토리에서 Git Repository 연동을 위한 SSH Key 생성 명령어

```
$ ssh-keygen -t ed25519 -a 100 -f <SSH Key명>
```

- 생성 될때까지 Enter를 눌러 기본값 설정

- 2개의 Key파일 생성 확인 : <SSH Key명>, <SSH Key명>.pub

로컬에서 Git Repository 연동을 위한 SSH Key 생성 #2

- Git Repository 연동을 위한 SSH Key 생성 결과

```
test@najeonghoui-MacBookPro ~/pems/jenkins ssh-keygen -t ed25519 -a 100 -f ssh-key
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh-key
Your public key has been saved in ssh-key.pub
The key fingerprint is:
SHA256:LlX9bqna/JYlDdMR0hKBn01DbDmrKAz9YXQZgukDQHk test@najeonghoui-MacBookPro.local
The key's randomart image is:
+--[ED25519 256]--+
| .00  o...B+=..|
| . E o ..B 0.o |
| . + . 0 = +..|
| . + +   =0 . |
| o S o . 0+ |
| = 0 . ...0 |
| . 0      ++ |
| .   0 00 |
| ..+0. |
+-----[SHA256]-----+
test@najeonghoui-MacBookPro ~/pems/jenkins ls
ssh-key      ssh-key.pub
test@najeonghoui-MacBookPro ~/pems/jenkins
```

Github Repository 연동 #1

- Github Repository(예제 실습코드 저장용)에 Deploy Key를 설정
 - 본인의 예제 실습코드가 저장된 **Github Repository**에 접속
 - 상단 Tab중 "**Settings**"클릭

The screenshot shows the GitHub interface for a repository named 'FastCampus' by user 'skwjdg710'. The repository is public and forked from 'DevOpsRunbook/FastCampus'. The main tab is selected, showing a file tree with folders like 'Appendix', 'Part2_Docker', 'Part4_Kubernetes', and 'Part6_MSA', along with files like '.gitignore' and 'README.md'. The README content is visible, starting with 'FastCampus' and 'Kubernetes와 Docker로 한번에 끝내는 컨테이너 기반 MSA 실습 예제코드 입니다.' The right sidebar shows repository statistics: 0 stars, 0 watching, 78 forks, and a language usage chart.

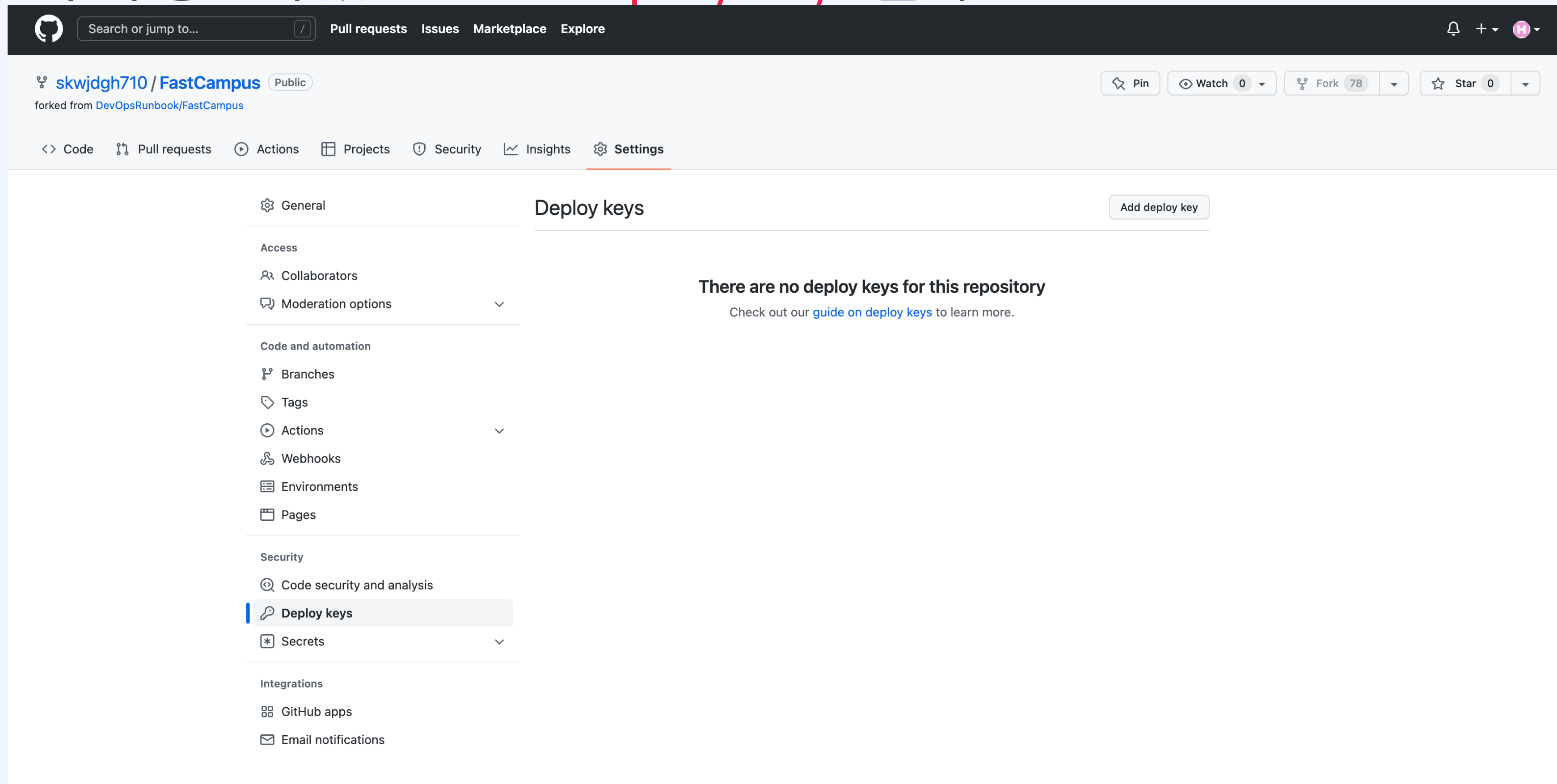
Category	Count
Stars	0
Watching	0
Forks	78

Language	Percentage
Java	42.8%
Mustache	35.2%
HCL	10.9%
Smarty	3.2%
HTML	2.3%
Go	2.1%
Other	3.5%

Github내 Repository 연동 #2

3. 좌측 메뉴중 Security > **Deploy Keys** 클릭

4. 우측 중간에 있는 "**Add deploy key**" 클릭



Github내 Repository 연동 #3

5. 다음의 정보를 입력후 "Add key" 버튼 클릭

- **Title**(제목) 입력 (ssh-key)
- Key(Public SSH Key인 **ssh-key.pub** 내용 복사) 붙여넣기
- **Allow write access** 체크 선택

```
test@najeonghoui-MacBookPro ~/pems/jenkins$ cat ssh-key.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIB0wPPd6/nkaPnE2nf9ug35DmDmwInCgStUNFtP
test@najeonghoui-MacBookPro ~/pems/jenkins$
```

Search or jump to... Pull requests Issues Marketplace Explore

skwjdg710 / FastCampus Public
forked from DevOpsRunbook/FastCampus

<> Code Pull requests Actions Projects Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Pages

Security

Code security and analysis

Deploy keys

Secrets

Deploy keys / Add new

Title

ssh-key

Key

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIB0wPPd6/nkaPnE2nf9ug35DmDmwInCgStUNFtP
test@najeonghoui-MacBookPro.local

☒ Allow write access
Can this key be used to push to this repository? Deploy keys always have pull access.

Add key

Github내 Repository 연동 #4

6. Github Repository(예제 실습코드 저장용)에 Deploy Key가 다음과 같이 등록됨을 확인

The screenshot shows the GitHub repository settings for 'skwjdggh710 / FastCampus'. The 'Deploy keys' tab is active, displaying a list of deployment keys. One key is listed: 'ssh-key' with the SHA256 hash 'L1X9bqna/JYLDdMR0hKBn01DbDmrKAz9YXQZgukDQHK'. It was added on 31 Oct 2022 by @skwjdggh710 and has 'Read/write' permissions. A 'Delete' button is present next to the key. The left sidebar shows the repository structure with 'Deploy keys' highlighted under the 'Settings' tab.