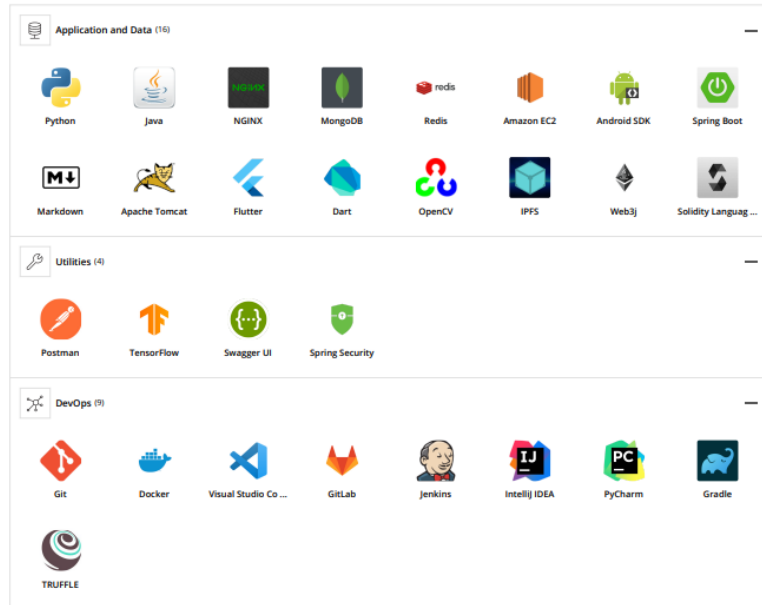


포팅메뉴얼

1. 프로젝트 구성도

기술 스택(stackshare)



시스템 아키텍처

2. Spring 서버 ec2 세팅

Server spec : t2.micro

os : Amazon Linux 2

docker 설치

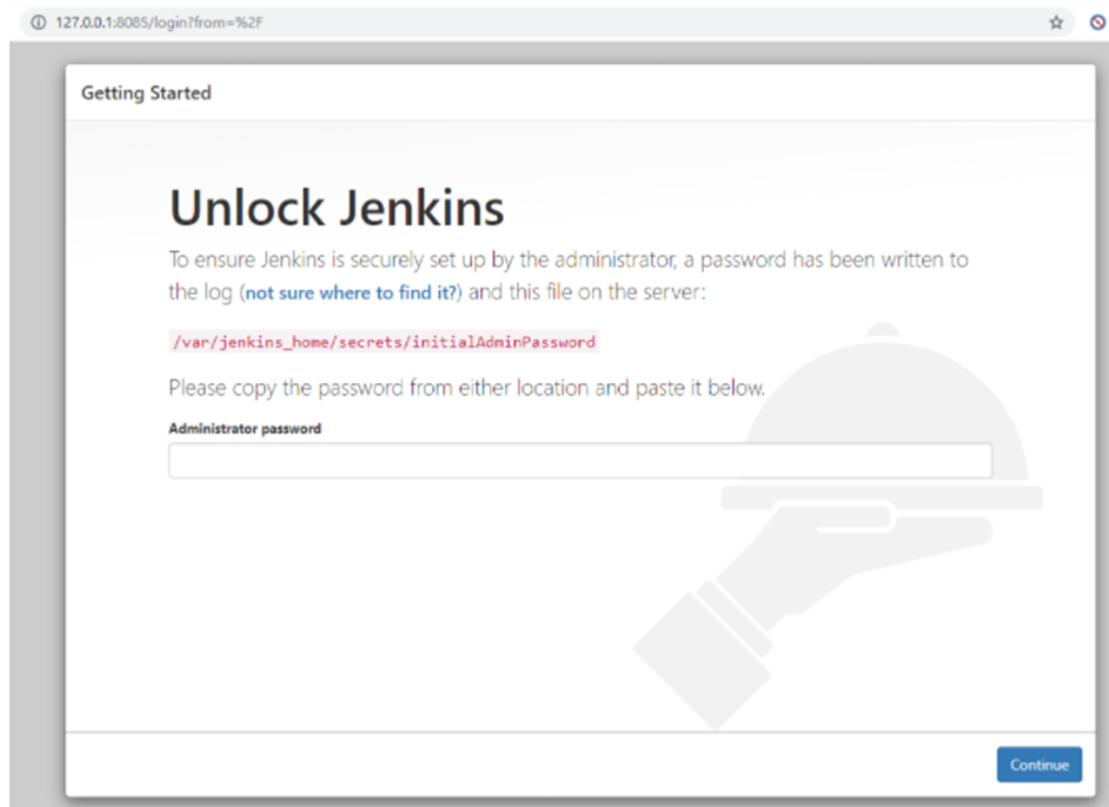
```
// docker 설치

// 패키지 업데이트
sudo apt update
// https관련 패키지 설치
sudo apt install apt-transport-https ca-certificates curl software-properties-common
// docker repository 접근을 위한 gpg 키 설정
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
// docker repository 등록
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
// 다시 업데이트
sudo apt update
// 도커 설치
sudo apt install docker-ce
// 설치 확인
docker --version
```

3. jenkins 서버 ec2 세팅

```
// Jenkins 설치

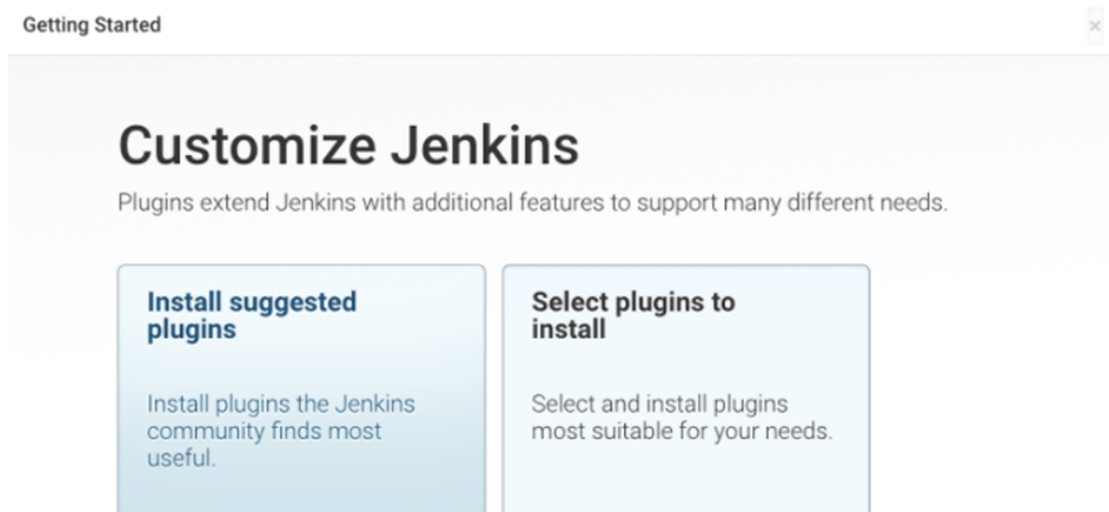
// 젠킨스 이미지 다운로드
docker pull jenkins/jenkins:ls
// 젠킨스 컨테이너 설치 및 실행
docker run -itd -p 8080:8080 -v /jenkins:/var/jenkins_home -name jenkins -u root jenkins/jenkins:ls
```



```
// Jenkins 컨테이너에 접근해서 어드민 패스워드 찾기
docker exec -it jenkins /bin/bash

cat /var/jenkins_home/secrets/initialAdminPassword

// 혹은 jenkins 로그를 출력해서 초기 비밀번호 확인
docker logs jenkins -f
```



Install suggested plugins 설치

Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	⚙ Credentials Binding	** SSH server Folders OWASP Markup Formatter ** Structs ** Pipeline: Step API ** Token Macro Build Timeout
⚙ Timestampper	⚙ Workspace Cleanup	⚙ Ant	⚙ Gradle	
⚙ Pipeline	⚙ GitHub Branch Source	⚙ Pipeline: GitHub Groovy Libraries	⚙ Pipeline: Stage View	
⚙ Git	⚙ SSH Build Agents	⚙ Matrix Authorization Strategy	⚙ PAM Authentication	
⚙ LDAP	⚙ Email Extension	⚙ Mailer		

** - required dependency

Jenkins 2.319.1

Getting Started

Create First Admin User

계정명:

암호:

암호 확인:

이름:

이메일 주소:

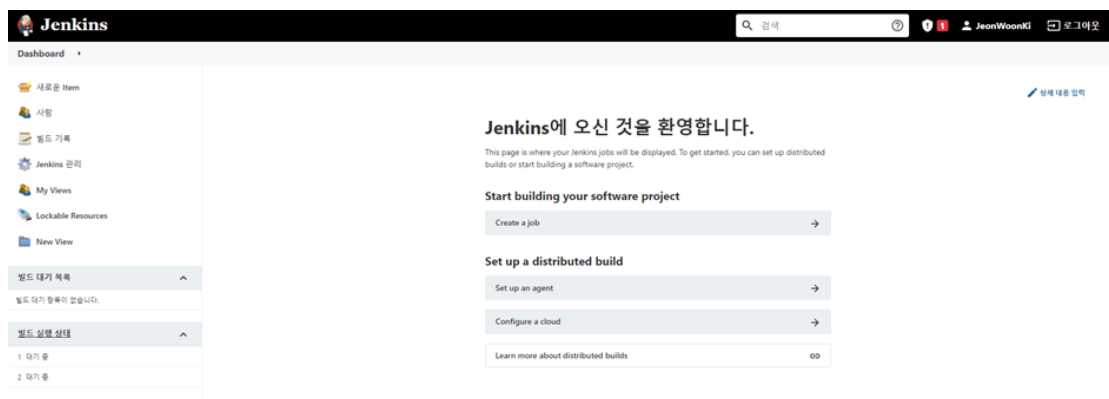
Jenkins 2.319.1

Skip and continue as admin

Save and Continue

설치가 완료되면 계정의 정보 입력

4. Jenkins 파이프라인 생성



필요 플러그인 설치

- Generic Webhook Trigger Plugin
- GitLab
- Gitlab API Plugin
- GitLab Authentication plugin
- Mattermost Notification Plugin
- Publish Over SSH

Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	GitLabID	ts7681@naver.com/***** (깃랩 설정)
		System	(global)	Blockus	GitLab API token (Blockus pj)
		System	(global)	lly	dldbud112@naver.com/***** (lly입니다)

Jenkins 관리 → Manage Credentials → Credentials 등록

Item 생성(Pipeline)

Build Triggers

☐ Build after other projects are built ?
☐ Build periodically ?
☒ Build when a change is pushed to GitLab. GitLab webhook URL: http://j8a209.p.ssafy.io:8080/project/donjoBackend ?

Enabled GitLab triggers

☒ Push Events
☐ Push Events in case of branch delete
☒ Opened Merge Request Events
☐ Build only if new commits were pushed to Merge Request ?
☐ Accepted Merge Request Events
☐ Closed Merge Request Events

Rebuild open Merge Requests

Never

☒ Approved Merge Requests (EE-only)
☒ Comments

Comment (regex) for triggering a build ?

Jenkins please retry a build

고급 ▼

☐ Generic Webhook Trigger ?
☐ GitHub hook trigger for GITScm polling ?

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://lab.ssfy.com/s08-blockchain-contract-sub2/608P22A209

Credentials ?

dldbdud112@naver.com/***** (jy입니다)

Add +

고급

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

deploy

Add Branch

Repository browser ?

(자동)

Additional Behaviours

Add +

Script Path ?

jenkins/backjenkinsfile

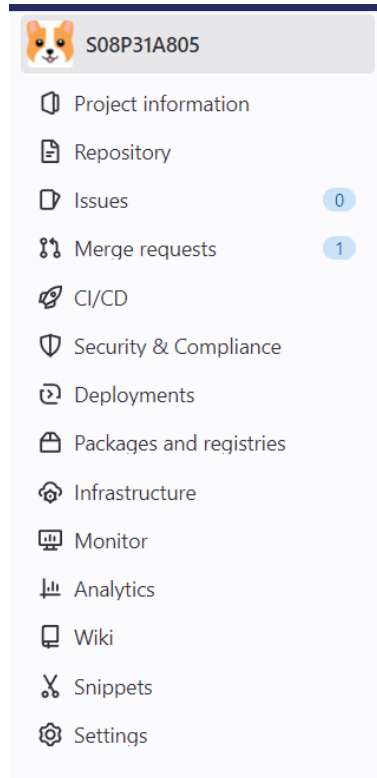
☒ Lightweight checkout ?

Pipeline Syntax

저장 Apply

설정 셋팅하기

5. Webhook 연결(Build Triggers URL과 Secret token 복사)



Webhooks

[Webhooks](#) enable you to send notifications to web applications in response to events in a group or project. We recommend using an [integration](#) in preference to a webhook.

URL

`http://k8a805.p.ssafy.io:8800/project/backend-springboot`

URL must be percent-encoded if it contains one or more special characters.

Secret token

.....

Used to validate received payloads. Sent with the request in the `X-Gitlab-Token` HTTP header.

Trigger

☒ Push events

`deploy`

Push to the repository.

☐ Tag push events

A new tag is pushed to the repository.

GitLab → Settings → WebURL Secret token 복사 붙여넣기

6. 프로젝트 이미지 생성 후 컨테이너 띄우기

Backend Pipeline

```
pipeline{
  agent any
  stages{
    stage('백엔드 자동 배포') {
      stages {
        stage('gradlew 권한'){
          steps{
            dir('backend'){
              sh "chmod +x gradlew"
            }
          }
        }
        stage('gradle 빌드'){
          steps{
            dir('backend'){
              sh './gradlew clean build --refresh-dependencies'
            }
          }
        }
      }
    }
  }
}
```

Backend DockerFile

Pipeline 생성 결과

7. Redis생성

8. Nginx 설정

포팅메뉴얼


```

listen [::]:80 ;
server_name k8a805.p.ssafy.io;

# 리다이렉트를 위한 설정
return 301 https://$server_name$request_uri;
}

server {
    listen [::]:443 ssl;
    listen 443 ssl;
    server_name k8a805.p.ssafy.io;

    client_max_body_size 25M;

    # SSL 인증서와 관련된 설정
    ssl_certificate /etc/letsencrypt/live/k8a805.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/k8a805.p.ssafy.io/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    # 실제 서비스를 제공하는 설정
    root /home/ubuntu/landing;
    index index.html index.htm index.nginx-debian.html;
    location / {
        try_files $uri $uri/ =404;
    }

    location /webui {
        proxy_pass http://localhost:5001;
        proxy_set_header Host $host;
    }

    location ~ ^/api2 {
        proxy_pass http://localhost:8700;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location ~ ^/(api|swagger|webjars|configuration|swagger-resources|v2|v3|csrf) {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

```

9. Blockchain 엔드포인트 설정

Infura를 통해 Sepolia 체인의 엔드포인트를 발급 받습니다.

Endpoints

Https

WebSockets

▽ REFINE

Our Web3 API Key works across several networks, use it on one or use it on all.



Ethereum

Mainnet ▾

https://mainnet.infura.io/v3/

🔗