

github repo 생성

The screenshot shows a GitHub repository page for 'simple-api'. The repository is public and was created by 'noel-sumini' on Aug 6, 2025, at 2:23 PM GMT+9. It has 1 branch and 0 tags. The main branch is 'main'. The repository has 19 commits, with the most recent being a modification to the 'ci/cd pipeline' by 'noel-sumini' 4 minutes ago. Other commits include adding a 'status code', the first commit, modifying a Jenkinsfile, and adding a 'numpy library'. A 'README' section is present but empty, with a button to 'Add a README'. The repository has 0 forks, 0 stars, and 0 watching. It has no releases or packages published. The languages used are Dockerfile (71.9%) and Python (28.1%). Suggested workflows are listed on the right.

Aug 6, 2025, 2:23 PM GMT+9 [GitHub.com/noel-sumini/simple-api](#)

noel-sumini / simple-api

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

simple-api Public

Pin Watch 0 Fork 0 Star 0

main 1 Branch 0 Tags Go to file + Code

noel-sumini ci/cd pipeline modified 8e43399 · 4 minutes ago 19 Commits

app status code added 3 days ago

Dockerfile first commit 3 weeks ago

Jenkinsfile ci/cd pipeline modified 4 minutes ago

requirements.txt numpy library added last week

README

Add a README

Add a README

No description, website, or topics provided.

Activity 0 stars 0 watching 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Languages

Dockerfile 71.9% Python 28.1%

Suggested workflows

Dockerfile

```
# 베이스 이미지
FROM python:3.11-slim

# 작업 디렉토리 생성
WORKDIR /app

# 의존성 복사 및 설치
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

# 애플리케이션 코드 복사
COPY app ./app

# 컨테이너 run할때 실행할 커맨드
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0",
      "--port", "18080"]
```

Jenkinsfile

```
pipeline {
    agent any
    triggers { githubPush() }

    environment {
        SHORT_COMMIT = "${GIT_COMMIT[0..6]}"
    }

    stages {
        stage('Checkout') {
            steps {
                git url: env.GIT_REPO, branch: 'main'
            }
        }

        stage('Build') {
            steps {
                timeout(time: 20, unit: 'MINUTES') {
                    sh ""
                    : "${DOCKER_IMAGE:?DOCKER_IMAGE not set}"
                    docker build -t "${DOCKER_IMAGE}:${SHORT_COMMIT}" -t "${DOCKER_IMAGE}:latest".
                }
            }
        }

        stage('Push') {
            steps {
                timeout(time: 10, unit: 'MINUTES') {
                    withCredentials([usernamePassword(
                        credentialsId: 'dockerhub-creds',
                        usernameVariable: 'DOCKER_USER',
                        passwordVariable: 'DOCKER_PASS'
                    )]) {
                        sh ""
                        echo "$DOCKER_PASS" | docker login -u "$DOCKER_USER" --password-stdin
                        docker push "${DOCKER_IMAGE}:${SHORT_COMMIT}"
                        docker push "${DOCKER_IMAGE}:latest"
                        docker logout || true
                    }
                }
            }
        }

        stage('Pull (remote)') {
            steps {
                timeout(time: 5, unit: 'MINUTES') {
                    sshagent([env.SSH_CRED_ID]) {
                        withCredentials([usernamePassword(
                            credentialsId: 'dockerhub-creds',
                            usernameVariable: 'DOCKER_USER',
                            passwordVariable: 'DOCKER_PASS'
                        )]) {
                            sh """
                            ssh -o StrictHostKeyChecking=no ${env.DEPLOY_USER}@${env.DEPLOY_HOST} \"\
                            (echo ${DOCKER_PASS} | docker login -u ${DOCKER_USER} --password-stdin || true) && \
                            docker pull ${env.DOCKER_IMAGE}:${SHORT_COMMIT} \
                            """
                            """
                            """
                        }
                    }
                }
            }
        }

        stage('Deploy (remote)') {
            steps {
                timeout(time: 5, unit: 'MINUTES') {
                    sshagent([env.SSH_CRED_ID]) {
                        sh """
                        ssh -o StrictHostKeyChecking=no ${env.DEPLOY_USER}@${env.DEPLOY_HOST} \"\
                        (docker rm -f ${env.CONTAINER_NAME} || true) && \
                        docker run -d --name ${env.CONTAINER_NAME} -p 3000:3000 \
                        ${env.DOCKER_IMAGE}:${SHORT_COMMIT} \
                        """
                        """
                        """
                    }
                }
            }
        }

        post {
            success { echo 'Deploy 성공' }
            failure { echo 'Deploy Fail' }
        }
    }
}
```

VM instance 생성

The screenshot shows the Google Cloud Compute Engine interface for managing VM instances. The left sidebar is collapsed, and the main header bar includes the Google Cloud logo, the project name "My First Project", a search bar, and various navigation icons.

The main content area is titled "VM 인스턴스" (VM Instances) and features a sub-header "인스턴스" (Instances). Below this, there's a table listing a single VM instance:

상태	이름	영역	권장사항	다음에서 사용 중:	내부 IP	외부 IP	연결
<input type="checkbox"/>	<input checked="" type="checkbox"/> instance-20250808-134311	us-central1-b			10.128.0.2 (nic0)	35.193.196.115 (nic0)	SSH

Below the table, there's a section titled "관련 작업" (Related Tasks) containing six cards:

- 보호 요약 살펴보기 (View summary): 무료로 데이터 보호의 취약점을 파악하고 VM 백업을 구성하세요.
- 결제 보고서 보기 (View payment reports): Compute Engine 결제를 확인하고 관리합니다.
- VM 모니터링 (Monitor VM): CPU 및 네트워크와 같은 측정항목 전반에서 이상점 VM을 확인합니다.
- VM 로그 살펴보기 (View logs): VM 인스턴스 로그를 보고 검색하고 분석하며 다운로드할 수 있습니다.
- 방화벽 규칙 설정 (Set firewall rules): VM 인스턴스와 주고받는 트래픽을 제어합니다.
- 패치 관리 (Manage patches): 패치 업데이트를 예약하고 VM 인스턴스의 패치 규정 준수를 확인합니다.

VM instance 생성

Google Cloud My First Project 리소스, 문서, 제품 등 검색(/) 검색 5 ? :

네트워크 보안 방화벽 정책 방화벽 정책 만들기 **방화벽 규칙 만들기** 알아보기

Cloud Armor DDoS 대시보드 Cloud Armor 정책 Adaptive Protection Cloud Armor 서비스 등급

Cloud IDS IDS 대시보드 IDS 엔드포인트 IDS 위협

방화벽 관리자 대시보드 **방화벽 정책** 위협 방화벽 앤드포인트

보안 웹 프록시 웹 프록시

SWP 정책 URL 목록

공통 구성요소 인증 구성 주소 그룹 보안 프로필 TLS 검사 정책 SSL 정책

실시간 분석 시작하기
포괄적인 모니터링 및 문제 해결에 Network Intelligence Center를 사용합니다. [자세히 알아보기](#)
 ✓ 네트워크 리소스를 시작화합니다.
✓ 연결 문제를 진단 및 방지합니다.
✓ 패킷 손실 및 지연 시간 측정항목을 확인합니다.
✓ 방화벽 규칙을 엄격하고 효율적으로 유지합니다.

사용해 보기 나중에 알림

VPC 방화벽 규칙
방화벽 규칙은 인스턴스로 수신 및 발신되는 트래픽을 제어합니다. 기본적으로 네트워크로 수신되는 모든 트래픽이 차단됩니다. [자세히 알아보기](#)
참고: App Engine 방화벽은 [App Engine 방화벽 규칙 섹션](#)에서 관리됩니다.

이 프로젝트에서 SMTP 포트 25가 허용되지 않습니다. [자세히 알아보기](#)

새로고침	로그 구성	삭제
필터 속성 이름 또는 값 입력		
<input type="checkbox"/> <u>default-allow-icmp</u>	인그레스 전체 적용 IP 범위: icmp 허용 65534 <u>default</u> 사용 안함	
<input type="checkbox"/> <u>default-allow-internal</u>	인그레스 전체 적용 IP 범위: tcp:0-65535 udp:0-65535 icmp 허용 65534 <u>default</u> 사용 안함	
<input type="checkbox"/> <u>default-allow-rdp</u>	인그레스 전체 적용 IP 범위: tcp:3389 허용 65534 <u>default</u> 사용 안함	
<input type="checkbox"/> <u>default-allow-ssh</u>	인그레스 전체 적용 IP 범위: tcp:22 허용 65534 <u>default</u> 사용 안함	

네트워크 방화벽 정책
방화벽 정책을 사용하면 여러 방화벽 규칙을 그룹화할 수 있으므로 한 번에 모든 규칙을 업데이트하고 Identity and Access Management(IAM) 역할로 효과적으로 제어할 수 있습니다.
[자세히 알아보기](#)

VM instance 생성

Google Cloud My First Project 리소스, 문서, 제품 등 검색(/) Q 검색

네트워크 보안 방화벽 규칙 만들기

Cloud Armor DDoS 대시보드 Cloud Armor 정책 Adaptive Protection Cloud Armor 서비스 등급

Cloud IDS IDS 대시보드 IDS 연드포인트 IDS 위협

방화벽 관리자 대시보드 방화벽 정책 위협 방화벽 엔드포인트

보안 웹 프록시 웹 프록시 SWP 정책 URL 목록

공통 구성요소 인증 구성 주소 그룹 보안 프로필 TLS 검사 정책 SSL 정책

방화벽 규칙은 인스턴스로 수신 및 발신되는 트래픽을 제어합니다. 기본적으로 네트워크로 수신되는 모든 트래픽이 차단됩니다.[자세히 알아보기](#)

이름 * jenkins

설명

로그

방화벽 로그를 사용 설정하면 대량의 로그가 생성되어 Logging 비용이 증가할 수 있습니다.[자세히 알아보기](#)

● 사용 ● 사용 안함

네트워크 * default

우선순위 * 1000

우선순위 범위는 ~65535입니다.

트래픽 방향

● 인그레스 ● 이그레스

일치 시 작업

● 허용 ● 거부

대상

지정된 대상 태그

다른 태그 * jenkins X

소스 필터

IPV4 범위 0.0.0.0/0 예: 10.0.0.0/16

보조 소스 필터

없음

대상 필터

없음

프로토콜 및 포트

○ 모두 허용 ○ 지정된 프로토콜 및 포트

TCP

포트 18080,50000

예: 20, 50~60

UDP

포트 예: 모두

SCTP

포트

예: 20, 50~60

VM instance 생성

Compute Engine ← instance-20250808-134311 인스턴스 수정

Intel Broadwell

디스플레이 기기

디스플레이 기기를 사용 설정하려면 VM 인스턴스를 중지해야 합니다.
화면 캡처 및 녹화 도구를 사용하려면 사용 설정하세요.

디스플레이 기기 사용 설정

네트워킹

네트워크 성능 구성

네트워크 대역폭 ②

네트워크 대역폭을 수정하려면 VM 인스턴스를 중지해야 합니다.

VM당 Tier_1 네트워크 성능 사용 설정
최대 아웃바운드 네트워크 대역폭: 8Gbps
VM과 공개 IP 간: 7Gbps

네트워크 인터페이스 ②

네트워크 인터페이스는 영구적입니다.

default default IPv4(10.128.0.0/20)

네트워크 인터페이스 추가

방화벽

Allow HTTP traffic
 Allow HTTPS traffic
 Allow Load Balancer Health checks

네트워크 태그

네트워크 태그
jenkins

저장용량

부팅 디스크

이름	instance-20250808-134311
이미지	ubuntu-minimal-2404-noble-amd64-v20250725
크기	100GB
프로비저닝된 IOPS	0
인터페이스 유형	SCSI
유형	균형 있는 영구 디스크
암호화 유형	Google 관리
모드	부팅, 읽기/쓰기
스냅샷 일정	default-schedule-1

삭제 규칙

인스턴스 삭제 시

디스크 유지
 디스크 삭제

기기 이름 ②

☰ 메타데이터

☰ 영역

docker 설치

```
sudo apt-get update  
sudo apt-get install -y ca-certificates curl gnupg  
lsb-release
```

```
sudo install -m 0755 -d /etc/apt/keyrings  
curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg |   
sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg  
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

```
echo   
"deb [arch=$(dpkg --print-architecture)  
signed-by=/etc/apt/keyrings/docker.gpg]   
https://download.docker.com/linux/ubuntu   
$(lsb_release -cs) stable" |   
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update  
sudo apt-get install -y docker-ce docker-ce-cli  
containerd.io docker-buildx-plugin  
docker-compose-plugin
```

```
Docker version 28.3.3, build 980b856  
[ysm2820@instance-20250808-134311:~]$ sudo apt install docker-compose-plugin  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
docker-compose-plugin is already the newest version (2.39.1-1~ubuntu.24.04~noble).  
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
```

Jenkins docker install

```
[ysm2820@instance-20250808-134311:~$ sudo vi docker-compose.yml  
ysm2820@instance-20250808-134311:~$ ]
```

```
version: '3.8'  
  
services:  
  jenkins:  
    image: jenkins/jenkins:lts  
    container_name: jenkins  
    user: root          # Docker 소켓에 접근 하려면 root 권장  
    restart: unless-stopped  
    environment:  
      # 초기 셋업 위치드 비활성화  
      - JAVA_OPTS=-Djenkins.install.runSetupWizard=false  
      # 시스템 타임존 맞추기  
      - TZ=Asia/Seoul  
      # 추가 Jenkins 옵션 (필요 시 prefix 등)  
      # - JENKINS_OPTS=--prefix=/jenkins  
    ports:  
      - "8080:8080"      # Jenkins UI  
      - "50000:50000"    # Jenkins agent 연결  
    volumes:  
      - jenkins_home:/var/jenkins_home  
      # Docker-in-Docker: 호스트 Docker 소켓 바인딩  
      - /var/run/docker.sock:/var/run/docker.sock  
      # Docker CLI 바이너리 바인딩 (필요 시)  
      - /usr/bin/docker:/usr/bin/docker  
  
volumes:  
  jenkins_home:  
~  
~  
~  
~  
~
```

```
Docker Compose version v2.39.1  
[ysm2820@instance-20250808-134311:~$ sudo docker-compose up -d  
Creating network "ysm2820_default" with the default driver  
Creating volume "ysm2820_jenkins_home" with default driver  
Pulling jenkins (jenkins/jenkins:lts)...  
lts: Pulling from jenkins/jenkins  
ebed137c7c18: Pull complete  
a16eed992861: Pull complete  
d0340747bfc5: Pull complete  
b3198e29cbc4: Pull complete  
9fa62d58a0db: Pull complete  
31a9a2be77c3: Pull complete  
3cd347526f2b: Pull complete  
654460caae81: Pull complete  
6ce3758f4dbb: Pull complete  
2b1923fe83ed: Pull complete  
a3391befb457: Pull complete  
48e332b529b0: Pull complete  
Digest: sha256:0e66af38c9272490ba18757d5d4d41e4ac2160278ae40b69d6da9b5adbe98794  
Status: Downloaded newer image for jenkins/jenkins:lts  
Creating jenkins ... done  
[ysm2820@instance-20250808-134311:~$ ]
```

```
[ysm2820@instance-20250808-134311:~$ sudo docker ps -a  
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                         NAMES  
03f22bbddfd        jenkins/jenkins:lts   "/usr/bin/tini -- /u..."   3 minutes ago       Up 3 minutes      0.0.0.0:50000->50000/tcp, [::]:50000->50000/tcp, 0.0.0.0:18080->8080/tcp, [::]:18080->8080/tcp   jenkins  
[ysm2820@instance-20250808-134311:~$ ]
```

jenkins plugin 설치

The image shows two screenshots of the Jenkins interface. The left screenshot is the Jenkins home page at 34.135.14.10:18080, featuring sections for 'Jenkins에 오신 것을 환영합니다.' (Welcome to Jenkins), 'Start building your software project', and 'Set up a distributed build'. The right screenshot is the Jenkins management page at 34.135.14.10:18080/manage/, showing various system configuration options like System Configuration, Security, Status Information, and About Jenkins. A red box highlights the 'Plugins' section under System Configuration, which contains a brief description and a link to the Jenkins documentation.

Jenkins 관리

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as BUILD_URL.

Please provide an accurate value in [Jenkins configuration](#).

Jenkins is currently unsecured and allows anyone on the network to launch processes on your behalf. It is recommended to set up security and to limit anonymous access even on private networks.

보안 설정 Dismiss

Plugins

Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.

System Configuration

System

환경 변수 및 경로 정보들을 설정합니다.

Tools

Configure tools, their locations and automatic installers.

Nodes

Add, remove, and control and monitor the various nodes that Jenkins runs jobs on.

Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

Configure the look and feel of Jenkins

Security

Security

Secure Jenkins; define who is allowed to access/use the system.

Status Information

시스템 정보

문제 해결을 돕기 위한 다양한 환경 정보를 보여줍니다.

System Log

System log captures output from java.util.logging output related to Jenkins.

About Jenkins

See the version and license information.

jenkins plugin 설치

Jenkins 기본 추천 플러그인 목록 (예示)		
카테고리	플러그인 이름	기능
SCM 연동	Git plugin	Git 저장소와 연동
	GitHub plugin	GitHub와 인증/빌드 트리거
	Subversion plugin	SVN 저장소 연동
빌드 파이프라인	Pipeline	Jenkins Pipeline (스크립트 기반 CI/CD)
	Declarative Pipeline	선언형 파이프라인 지원
노드/에이전트 관리	SSH Agent plugin	빌드 시 SSH 키 사용
	Matrix Authorization Strategy	권한 매트릭스 기반 접근 제어
	Node and Label parameter plugin	빌드 시 노드/레이블 선택
UI 및 관리	Blue Ocean (선택적)	파이프라인 시각화 UI
	Credentials Binding plugin	빌드 환경에서 안전하게 비밀 값 주입
	Email Extension plugin	빌드 결과 이메일 알림
유ти리티	Ant plugin	Apache Ant 빌드 지원
	Gradle plugin	Gradle 빌드 지원
	JUnit plugin	JUnit 테스트 결과 리포트
Workflow Aggregator		
파이프라인 관련 플러그인 모음		

34.135.14.10:18080/manage/pluginManager/available

Plugins

Updates Available plugins Installed plugins Advanced settings

Install Name Released Health

Install	Name	Released	Health
<input checked="" type="checkbox"/>	GitHub 1.44.0	18 days ago	99
<input checked="" type="checkbox"/>	Subversion 1287.vd2d507146906	6 mo 2 days ago	96
<input checked="" type="checkbox"/>	Pipeline 608.v67378e9d3db_1	4 mo 22 days ago	100
<input checked="" type="checkbox"/>	Pipeline: Declarative 2.2258.v4e96d2b_da_f9b_	16 days ago	96
<input checked="" type="checkbox"/>	SSH Agent 386.v36cc0c7582f0	2 mo 26 days ago	99
<input checked="" type="checkbox"/>	Blue Ocean 1.27.21	1 mo 22 days ago	99
	Common API for Blue Ocean 1.97.21		

github login credential 등록

The screenshot shows the Jenkins Management interface. At the top, there is a navigation bar with the Jenkins logo, a search bar, and a settings icon. Below the navigation bar, the title "Jenkins 관리" (Jenkins Management) is displayed. A search bar with the placeholder "Search settings" is located on the right side of the header.

The main content area is organized into several sections:

- System Configuration**: Includes links for "System", "Nodes", "Tools", "Clouds", "Plugins", and "Appearance".
- Security**: Includes links for "Security" and "Credentials". The "Credentials" link is highlighted with a red box.
- Status Information**: Includes links for "시스템 정보", "System Log", and "부하 통계".
- About Jenkins**: Provides version and license information.
- Troubleshooting**: Includes links for "Manage Old Data".
- Tools and Actions**: Includes links for "Reload Configuration from Disk", "Jenkins CLI", and "Script Console".
- GitHub Login Credential Registration**: This section is partially visible at the bottom of the page, showing steps to register GitHub credentials.

A large red box highlights the "Credentials" link under the Security section, indicating it is the target of the screenshot.

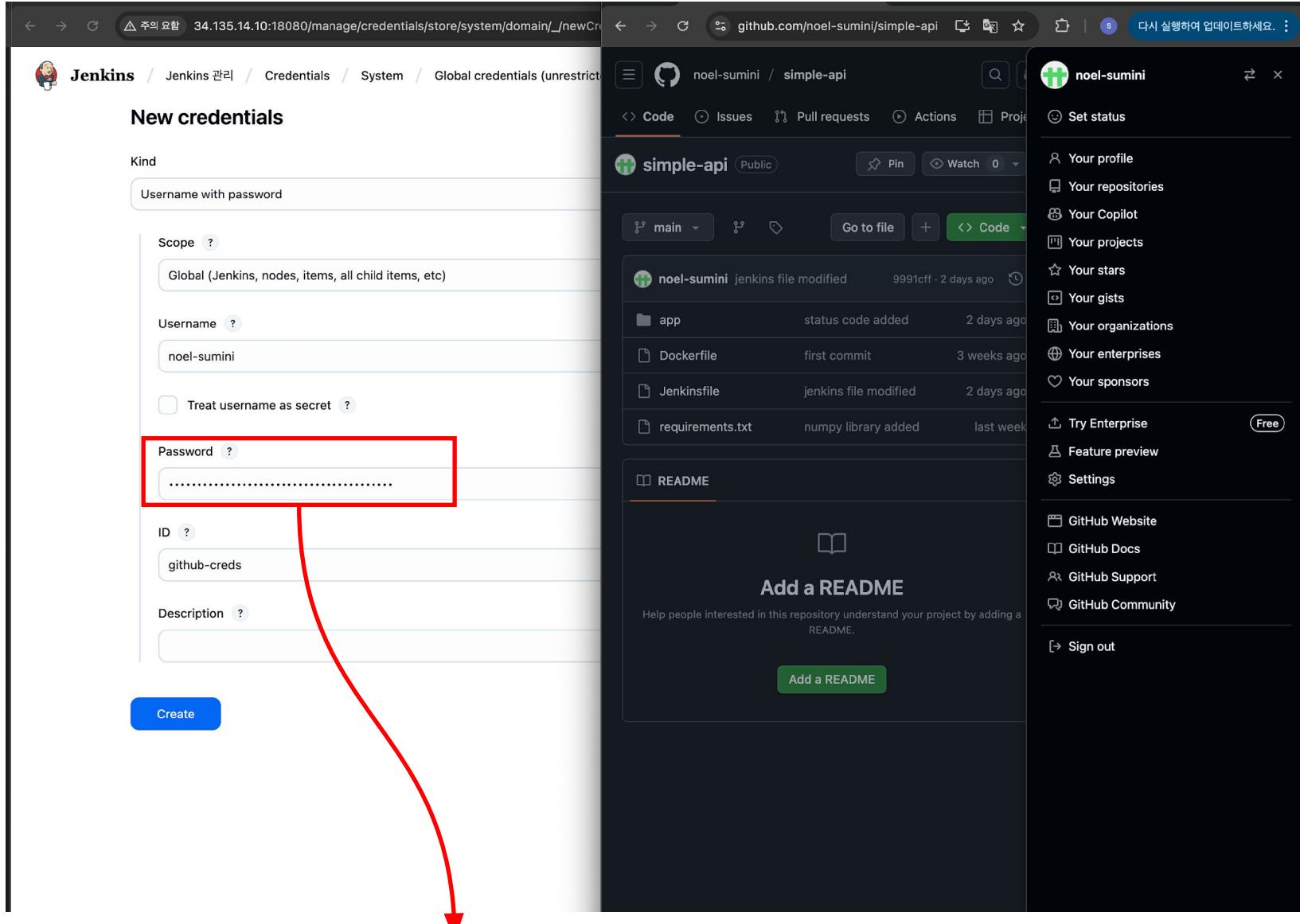
github login credential 등록

The screenshot shows the Jenkins Credentials management interface. At the top, there is a navigation bar with icons for Jenkins, Jenkins 관리 (Management), and Credentials, along with search and filter tools. Below the navigation is a header titled "Credentials" with columns for Type (T), Provider (P), Store (Store), Domain, ID, and Name.

The main section is titled "Stores scoped to Jenkins" and displays a list of credential stores. One store, "System", is selected and highlighted with a red box around its "Domains" dropdown menu. The dropdown menu shows "(global)" as the current selection, with an "Add credentials" option below it. There are also icons for "Add" and "Edit".

At the bottom left, there are size options: 아이콘: S M L.

github login credential 등록



github personal access
tokens

dockerhub login credential 등록

The screenshot shows the Jenkins 'Credentials' management interface. At the top, there's a navigation bar with icons for Jenkins, Jenkins 관리, and Credentials, along with search and filter tools. Below the header, the title 'Credentials' is displayed. A horizontal toolbar below the title includes filters for 'T' (Type), 'P' (Provider), and 'Store' (sorted by ID), followed by columns for 'Domain', 'ID', and 'Name'. The main content area is titled 'Stores scoped to Jenkins' and lists a single entry: 'System' under the 'Domains' column. The 'Provider' column shows a user icon, and the 'ID' column shows '(global)'. A red box highlights the 'Add credentials' button at the bottom right of this list. Below the list, there are icons for sorting by 'Name' (A-Z) and 'Last modified' (L).

dockerhub login credential 등록

The screenshot shows the Docker Hub user interface for managing personal access tokens. The left sidebar is dark-themed and includes sections for Home, Hub, Build Cloud, Scout, Testcontainers Cloud, Docker Desktop, Settings (with Personal access tokens selected), Connected accounts, Convert, Privacy, Deactivate, and Billing. The main content area has a blue header "Personal access tokens". It contains a message: "You haven't created any personal access tokens." Below this is a "Generate new token" button. The top right of the screen shows standard Docker Hub navigation icons.

The screenshot shows the "Create access token" form on the Docker Hub website. The left sidebar is identical to the first screenshot. The main form has a title "Create access token" and a sub-section "Personal access tokens / New access token". It includes fields for "Access token description" (set to "jenkins"), "Expiration date" (set to "None"), and "Optional Access permissions" (set to "Read, Write, Delete"). A note below states: "Read, Write, Delete tokens allow you to manage your repositories." At the bottom are "Cancel" and "Generate" buttons. The "Personal access tokens" section in the sidebar is highlighted with a gray box.

dockerhub login credential 등록

The screenshot shows the Jenkins 'New credentials' configuration page. The URL in the browser is `Jenkins / Jenkins 관리 / Credentials / System / Global credentials (unrestricted)`. The page title is 'New credentials'. The 'Kind' dropdown is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'ysm2820'. The 'Treat username as secret' checkbox is unchecked. The 'Password' field is filled with several dots ('.....'). The 'ID' field is set to 'dockerhub-creds'. The 'Description' field is empty. At the bottom left is a blue 'Create' button.

Jenkins / Jenkins 관리 / Credentials / System / Global credentials (unrestricted)

New credentials

Kind

Username with password

Scope

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

Username

ysm2820

Treat username as secret

Password

.....

ID

dockerhub-creds

Description

Create

webhook 등록

The image shows two screenshots illustrating the configuration of a webhook.

Left Screenshot (GitHub): The GitHub settings page for a repository. The left sidebar shows 'Webhooks' selected. The main area displays the 'Webhooks' section with an 'Add webhook' button highlighted by a red box. A message indicates a failed delivery to the URL `http://152.69.230.75:18080/github-webhook/` due to an invalid HTTP response (403).

Right Screenshot (Jenkins): The Jenkins 'Webhooks / Manage webhook' configuration page. The URL `http://34.31.224.157:18080/github-webhook/` is entered into the 'Payload URL' field, which is also highlighted by a red box. A red arrow points from the GitHub 'Add webhook' button to this field, indicating the connection between the two configurations.

docker group내 id 등록

docker group에 id를 등록 (sudo 없이도 docker 사용 가능하도록)

```
Last login: Fri Aug  8 15:37:36 2025 from 34.61.43.148
smyang9777@instance-20250808-150304:~$ sudo usermod -aG docker smyang9777
```

세션 재로그인 후 확인

The screenshot shows an SSH terminal window titled "브라우저에서 SSH를 통해 연결". The terminal displays the following text:

```
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.14.0-1011-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

21 updates can be applied immediately.
13 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug  8 16:12:47 2025 from 35.235.245.129
smyang9777@instance-20250808-150304:~$ id
uid=1001(smyang9777) gid=1002(smyang9777) groups=1002(smyang9777),4(adm),20(dialout),24(cdrom),25(floppy),29(au
dio),30(dip),44(video),46(plugdev),101(1xd),105(netdev),991(docker),1000(ubuntu),1001(google-sudoers)
smyang9777@instance-20250808-150304:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
smyang9777@instance-20250808-150304:~$ [ ]
```

jenkins 서버의 ssh key를 원격지에 등록

jenkins서버에서 ssh-key 발급

```
sumin - ysm2820@instance-20250808-134311:~ -- ssh -i ~/.ssh/gcp-250808 ysm2820@34.61.43.148 - 172x34
[sshd] password for ysm2820:
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
 NAMES
03f22bbddfd jenkins/jenkins:lts "/usr/bin/tini -- /u..." 2 hours ago Up 55 minutes 0.0.0.0:50000->50000/tcp, [::]:50000->50000/tcp, 0.0.0.0:18080->8080/tcp, [::]:18080->8080/tcp
jenkins
ysm2820@instance-20250808-134311:~$ sudo docker exec -it jenkins /bin/bash
root@03f22bbddfd:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@03f22bbddfd:/# ssh-keygen -e ed25519 -f ~/.ssh/jenkins
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/jenkins
Your public key has been saved in /root/.ssh/jenkins.pub
The key fingerprint is:
SHA256:en5mzrYSpkRBWx9+uec:dJ5nd1R13RFw7IYXwpJc root@03f22bbddfd
The key's randomart image is:
+--[ED25519 256]--+
| ..o+oB|
| o .oo+o|
| . + .+oEo|
| o .. o o|
| S .. |
| . + ..+|
| = o oo+|
| * ..,+o=+|
| o= ..o=+|
+---[SHA256]--+
root@03f22bbddfd:/#
root@03f22bbddfd:/#
root@03f22bbddfd:/# cat ~/.ssh/jenkins.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIr6dmUQi0FNiPaBjwYtiuE+DlxUgnHMykAeNaIUMGSN root@03f22bbddfd
root@03f22bbddfd:/#
```

Jenkins 서버의 public key를 원격지서버(배포대상 서버) 의 authorized_keys에 등록



```
smyang9777@instance-20250808-150304:~$ mkdir -p ~/.ssh && chmod 700 ~/.ssh
smyang9777@instance-20250808-150304:~$ echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIKz9H/vZy8sUp99L1ogQbElpLzNullNScMBdH1AOec8V ysm2820@instance-20250808-134311" > ~/.ssh/authorized_keys
smyang9777@instance-20250808-150304:~$ chmod 600 ~/.ssh/authorized_keys
smyang9777@instance-20250808-150304:~$
```

jenkins docker 서버 -> 원격지 서버로 ssh 접속하여 확인

```
root@03f22bbddfd:/# ssh -i ~/.ssh/jenkins smyang9777@34.172.222.28
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.14.0-1011-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
```

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

21 updates can be applied immediately.

13 of these updates are standard security updates.

To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

Last login: Fri Aug 8 16:14:35 2025 from 35.235.244.34

[smyang9777@instance-20250808-150304:~\$ exit

logout

Connection to 34.172.222.28 closed.

jenkins 서버의 ssh key(private)을 jenkins에 credential로 등록

The screenshot shows the Jenkins web interface for managing credentials. The top navigation bar includes the Jenkins logo, 'Jenkins' text, and 'Jenkins 관리 / Credentials'. On the right side of the header are search, filter, and settings icons. Below the header, the page title is 'Credentials'. A horizontal toolbar below the title contains buttons for 'T' (Text), 'P' (Password), 'Store' (dropdown), 'Domain' (dropdown), 'ID' (dropdown), and 'Name' (dropdown). The main content area is titled 'Stores scoped to Jenkins'. It lists a single item: 'System' under the 'Domains' column, with '(global)' selected. A red box highlights the '(global)' dropdown. Below the table, there is a note in Korean: '아이콘: S M L' (Icons: S M L). At the bottom right of the table area is a blue 'Add credentials' button.

jenkins 서버의 ssh key(private)을 jenkins에 credential로 등록

Jenkins / Jenkins 관리 / Credentials / System / Global credentials (unrestricted) / smyang9777

Update Delete Move

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

ID ? deploy-server-ssh

Description ?

Username smyang9777

Treat username as secret ?

Private Key Enter directly

Key

-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaclZxtjdEAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAAMwAAAAAtzc2gt2W
0yNTUx0QAAACDK-nzIEItBTYj2g8GLYrhPgMVIIxzMphWfDBkjOAAAJqg1Xk4KpV5
OAAAAtzc2gt2W0yNTUx0QAAACDK-nzIEItBTYj2g8GLYrhPgMVIIxzMphWfDBkjQ
AAAEDPR83Fw9fhw+0Tyb740KpZg2kly032lygeBD05WP5crfdmUQj0FN1pBjwYtiuE+
DixUgnhMyAeNaIUMGSNAAAEXJvb3RANDNMMjJ1YmRkZmRkAQIDBA==
-----END OPENSSH PRIVATE KEY-----

Passphrase

```
sumin -- smyang9777@instance-20250808-150304: ~ -- ssh -i ./ssh/gcp-250808 ysm2820@34.61.43.148 - 172x34
root@03f22bbddffd:/# ssh -i ./ssh/jenkins smyang9777@34.61.43.148
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.14.0-1011-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

21 updates can be applied immediately.
13 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug  8 16:27:10 2022 from 34.61.43.148
[smyang9777@instance-20250808-150304:~$ exit
logout
Connection to 34.172.222.28 closed.
root@03f22bbddffd:/# cat ./ssh/jenkins
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaclZxtjdEAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAAMwAAAAAtzc2gt2W
0yNTUx0QAAACDK-nzIEItBTYj2g8GLYrhPgMVIIxzMphWfDBkjOAAAJqg1Xk4KpV5
OAAAAtzc2gt2W0yNTUx0QAAACDK-nzIEItBTYj2g8GLYrhPgMVIIxzMphWfDBkjQ
AAAEDPR83Fw9fhw+0Tyb740KpZg2kly032lygeBD05WP5crfdmUQj0FN1pBjwYtiuE+
DixUgnhMyAeNaIUMGSNAAAEXJvb3RANDNMMjJ1YmRkZmRkAQIDBA==
-----END OPENSSH PRIVATE KEY-----
root@03f22bbddffd:/#
```

jenkins pipeline 생성

The screenshot shows the Jenkins dashboard at the URL `34.61.43.148:18080`. A red box highlights the 'Create a job' button in the top-left corner of the sidebar. The main content area displays a message in Korean: 'Jenkins에 오신 것을 환영합니다.' (Welcome to Jenkins). Below this, there is a brief introduction: 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' A large 'Start building your software project' button is centered. On the left, there are sections for 'Build Queue' and 'Build Status'. On the right, there are links for setting up distributed builds: 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

△ 주의 요함 34.61.43.148:18080

Jenkins

+ 새로운 Item

빌드 기록

빌드 대기 목록

빌드 대기 항목이 없습니다.

빌드 실행 상태 0/2

상세 내용 입력

Jenkins에 오신 것을 환영합니다.

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job +

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds ?

jenkins pipeline 생성

← → ⌛ 주의 요함 34.61.43.148:18080/view/all/newJob ⚙ ☆ ⏺ | 다시 실행하여 업데이트하세요

Jenkins / All / New Item

New Item

Enter an item name
test

Select an item type

 Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 Multi-configuration project
다양한 환경에서의 테스트, 플랫폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.

 Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

 Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

 Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK

jenkins pipeline 생성

- Trigger 조건 등록

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Build after other projects are built [?](#)
 - Build periodically [?](#)
 - GitHub hook trigger for GITScm polling [?](#)
 - Poll SCM [?](#)
-

jenkins pipeline 생성 – git repo 등록

The screenshot shows the Jenkins Pipeline configuration page for a job named 'test'. The left sidebar lists 'General', 'Triggers', 'Pipeline' (which is selected), and 'Advanced'. The main area is titled 'Pipeline' with the sub-instruction 'Define your Pipeline using Groovy directly or pull it from source control.' Below this, a dropdown menu is set to 'Pipeline script from SCM'. A 'SCM' section is expanded, showing 'Git' selected. Under 'Repositories', there is one entry with 'Repository URL' set to 'https://github.com/noel-sumini/simple-api' and 'Credentials' set to 'noel-sumini/*****'. A '+ Add' button is available for adding more repositories. Below the repository section is 'Branches to build' with 'Branch Specifier (blank for \'any\')' set to '*/*main'. An 'Add Branch' button is present. At the bottom, there is a 'Repository browser' dropdown set to '(자동)' and an 'Additional Behaviours' section with an 'Add' button.

jenkins pipeline 생성

- parameter(env)

등록

Jenkins / test / Configuration

Configure

General

Triggers

Pipeline

Advanced

이 빌드는 매개변수가 있습니다 ?

String Parameter ?

매개변수명 ? DOCKER_IMAGE

Default Value ? ysm2820/simple-api

설명 ?

Plain text 미리보기

Trim the string ?

String Parameter ?

매개변수명 ? DEPLOY_HOST

Default Value ? 34.172.222.28

설명 ?

Plain text 미리보기

Trim the string ?

String Parameter ?

매개변수명 ? DEPLOY_USER

Default Value ? smyang9777

설명 ?

Plain text 미리보기

Trim the string ?

Q ⓘ =

매개변수명 ? GIT_REPO

Default Value ? https://github.com/noel-sumini/simple-api.git

설명 ?

Plain text 미리보기

Trim the string ?

String Parameter ?

매개변수명 ? APP_DIR

Default Value ? /home/smyang9777/app

설명 ?

Plain text 미리보기

Trim the string ?

String Parameter ?

매개변수명 ? CONTAINER_NAME

Default Value ? simple-api

설명 ?

Plain text 미리보기

Trim the string ?

String Parameter ?

매개변수명 ? SSH_CRED_ID

Default Value ? deploy-server-ssh

CI/CD

Test

- Git Push

```
● sumin@MacBook-Pro-2 sknetworks % git add .
● sumin@MacBook-Pro-2 sknetworks % git commit -m "deployment validation check msg changed"
[main 6cecb64] deployment validation check msg changed
  1 file changed, 2 insertions(+), 2 deletions(-)
● sumin@MacBook-Pro-2 sknetworks % git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 10 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/noel-sumini/simple-api.git
  6b5072e..6cecb64  main -> main
○ sumin@MacBook-Pro-2 sknetworks % █
```

CI/CD Test

- Github webhook / Jenkins trigger 확인

Webhooks / Manage webhook

General Access Collaborators Moderation options Code and automation Branches Tags Rules Actions Models Webhooks Copilot Environments Codespaces Pages Security Advanced Security Deploy keys Secrets and variables Integrations GitHub Apps Email notifications

Request Response 200 Headers Request URL: http://34.31.224.157:18080/github-webhook/ Request method: POST Accept: */* Content-Type: application/json User-Agent: GitHub-Hookshot/832ccca X-GitHub-Delivery: 51f8d8b2-7478-11f0-9080-4898c4309cfe X-GitHub-Event: push X-GitHub-Hook-ID: 562761271 X-GitHub-Hook-Installation-Target-ID: 1022374174 X-GitHub-Hook-Installation-Target-Type: repository Payload

```
{ "ref": "refs/heads/main", "before": "6b5072ed7c8b002fcfb473f14d529946a670b1c3", "after": "6cecb640482fc286ea117bc52814850e8a2d1ee1", "repository": { "id": 1022374174, "node_id": "R_kgDOPPAxHg", "name": "simple-api", "full_name": "noel-sumini/simple-api", "private": false, "owner": { "name": "noel-sumini", "email": "63632804+noel-sumini@users.noreply.github.com", "login": "noel-sumini", "id": 63632804, "node_id": "MDQ6VXNlcjYzNjMyODA0", "avatar_url": "https://avatars.githubusercontent.com/u/63632804?v=4", "gravatar_id": "", "url": "https://api.github.com/users/noel-sumini", "html_url": "https://github.com/noel-sumini" } } }
```

Jenkins / test / #10

Status Changes Console Output Edit Build Information Delete build #10 Polling Log Parameters Git Build Data 블루 오션 열기 Restart from Stage Replay Pipeline Steps Workspaces Previous Build

#10 (2025. 8. 9. 오전 1:54:25) Started by GitHub push by noel-sumini Revision: 6cecb640482fc286ea117bc52814850e8a2d1ee1 Repository: https://github.com/noel-sumini/simple-api refs/remotes/origin/main Changes 1. deployment validation check msg changed (details / githubweb) 2. deployment validation check msg changed (details / githubweb) 3. deployment validation check msg changed (details / githubweb)

상세 내용 입력 이번 빌드 영구 보관하기 18 sec 전에 업데이드 됨. 소요 5.2 sec

CI/CD Test

- console stdout 확인

34.31.224.157:18080/job/test/13/console

Jenkins / test / #13

Status Changes Console Output (Selected) Edit Build Information Polling Log Parameters Git Build Data 블루 오션 열기 Thread Dump Pause/resume Replay Pipeline Steps Workspaces Previous Build

콘솔 출력

Started by GitHub push by noel-sumini
Obtained Jenkinsfile from git <https://github.com/noel-sumini/simple-api>
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/test
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-creds
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url <https://github.com/noel-sumini/simple-api> # timeout=10
Fetching upstream changes from <https://github.com/noel-sumini/simple-api>
> git --version # timeout=10
> git --version # 'git version 2.39.5'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- <https://github.com/noel-sumini/simple-api> +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 8e4339919e234558ea605cac08478520398be1e8 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 8e4339919e234558ea605cac08478520398be1e8 # timeout=10
Commit message: "ci/cd pipeline modified"
> git rev-list --no-walk 9bd030597a6ff63bccb994b77c6566028e3a6473 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url <https://github.com/noel-sumini/simple-api.git> # timeout=10
Fetching upstream changes from <https://github.com/noel-sumini/simple-api.git>
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- <https://github.com/noel-sumini/simple-api.git> +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 8e4339919e234558ea605cac08478520398be1e8 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 8e4339919e234558ea605cac08478520398be1e8 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D main # timeout=10
> git checkout -b main 8e4339919e234558ea605cac08478520398be1e8 # timeout=10
Commit message: "ci/cd pipeline modified"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] timeout
Timeout set to expire in 20 min
[Pipeline] {
[Pipeline] sh

Download Copy View as plain text

CI/CD Test

- console stdout

확인

```
+ : ysm2820/simple-api
+ docker build -t ysm2820/simple-api:8e43399 -t ysm2820/simple-api:latest .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
  Install the buildx component to build images with BuildKit:
  https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 162.8kB

Step 1/6 : FROM python:3.11-slim
3.11-slim: Pulling from library/python
59e22667830b: Pulling fs layer
abd846fa1cdb: Pulling fs layer
b7b61708209a: Pulling fs layer
4085babbc570: Pulling fs layer
4085babbc570: Waiting
abd846fa1cdb: Verifying Checksum
abd846fa1cdb: Download complete
4085babbc570: Verifying Checksum
4085babbc570: Download complete
59e22667830b: Verifying Checksum
59e22667830b: Download complete
b7b61708209a: Verifying Checksum
b7b61708209a: Download complete
59e22667830b: Pull complete
abd846fa1cdb: Pull complete
b7b61708209a: Pull complete
4085babbc570: Pull complete
Digest: sha256:0ce77749ac83174a31d5e107ce0cfab28a2fd6b0615e029d9d84b39c48976ee
Status: Downloaded newer image for python:3.11-slim
--> f3bfd8e9386c
Step 2/6 : WORKDIR /app
--> Running in 77873af6f919
--> Removed intermediate container 77873af6f919
--> 9ccae6135b0b
Step 3/6 : COPY requirements.txt .
--> dc10e7621207
Step 4/6 : RUN pip install --no-cache-dir -r requirements.txt
--> Running in 4055a384d8bd
Collecting fastapi (from -r requirements.txt (line 1))
  Downloading fastapi-0.116.1-py3-none-any.whl.metadata (28 kB)
Collecting uvicorn (from -r requirements.txt (line 2))
  Downloading uvicorn-0.35.0-py3-none-any.whl.metadata (6.5 kB)
Collecting pandas (from -r requirements.txt (line 3))
  Downloading pandas-2.3.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (91 kB)
--> 91.2/91.2 kB 6.1 MB/s eta 0:00:00
Collecting numpy (from -r requirements.txt (line 4))
  Downloading numpy-2.3.2-cp311-cp311-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (62 kB)
--> 62.1/62.1 kB 243.1 MB/s eta 0:00:00
Collecting starlette<=0.48.0,>=0.40.0 (from fastapi->-r requirements.txt (line 1))
  Downloading starlette-0.47.2-py3-none-any.whl.metadata (6.2 kB)
Collecting pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 (from fastapi->-r requirements.txt (line 1))
  Downloading pydantic-2.11.7-py3-none-any.whl.metadata (67 kB)
--> 68.0/68.0 kB 199.0 MB/s eta 0:00:00
Collecting typing-extensions<=4.8.0 (from fastapi->-r requirements.txt (line 1))
  Downloading typing_extensions-4.14.1-py3-none-any.whl.metadata (3.0 kB)
Collecting click=>7.0 (from uvicorn->-r requirements.txt (line 2))
  Downloading click-8.2.1-py3-none-any.whl.metadata (2.5 kB)
Collecting h11=>0.8 (from uvicorn->-r requirements.txt (line 2))
  Downloading h11-0.16.0-py3-none-any.whl.metadata (8.3 kB)
Collecting python-dateutil=>2.8.2 (from pandas->-r requirements.txt (line 3))
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>>2020.1 (from pandas->-r requirements.txt (line 3))
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>>2022.7 (from pandas->-r requirements.txt (line 3))
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting annotated-types>=0.6.0 (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1))
  Downloading annotated_types-0.7.0-py3-none-any.whl.metadata (15 kB)
Collecting pydantic-core==2.33.2 (from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 1))
  Downloading pydantic_core-2.33.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.8 kB)
```

CI/CD Test

– console stdout 확인

```
Downloading snifferio-1.3.1-py3-none-any.whl (10 KB)
Installing collected packages: pytz, tzdata, typing-extensions, sniffio, six, numpy, idna, h11, click, annotated-types, uvicorn, typing-inspection, python-dateutil, pydantic-core, anyio, starlette, pydantic, pandas, fastapi
Successfully installed annotated-types-0.7.0 aiohttp-4.10.0 click-8.2.1 fastapi-0.116.1 h11-0.16.0 idna-3.10 numpy-2.3.2 pandas-2.3.1 pydantic-2.11.7 pydantic-core-2.33.2 python-dateutil-2.9.0.post0 pytz-2025.2 six-1.17.0 sniffio-1.3.1 starlette-0.47.2 typing-extensions-4.14.1 typing-inspection-0.4.1 tzdata-2025.2 uvicorn-0.35.0
[91mWARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[0m [91m
[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: pip install --upgrade pip
[0m ----> Removed intermediate container 4055a384d8bd
----> 89b6db2f6343
Step 5/6 : COPY app ./app
----> 893d170f67f5
Step 6/6 : CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "18080"]
----> Running in d9786c945cb8
----> Removed intermediate container d9786c945cb8
----> 5fcc6d02ef5d
Successfully built 5fcc6d02ef5d
Successfully tagged ysm2820/simple-api:8e43399
Successfully tagged ysm2820/simple-api:latest
[Pipeline]
[Pipeline] // timeout
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push)
[Pipeline] timeout
Timeout set to expire in 10 min
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKER_PASS
[Pipeline] {
[Pipeline] sh
+ echo ****
+ docker login -u ysm2820 --password-stdin

WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
+ docker push ysm2820/simple-api:8e43399
The push refers to repository [docker.io/ysm2820/simple-api]
1b023e321a47: Preparing
b960a4e3e07f: Preparing
ed19f53fb005: Preparing
c3f071bbaa4b: Preparing
d22cc68b10d7: Preparing
943faa7467a0: Preparing
0a00f6ce5fb7: Preparing
7cc7fe68eff6: Preparing
943faa7467a0: Waiting
0a00f6ce5fb7: Waiting
7cc7fe68eff6: Waiting
d22cc68b10d7: Mounted from library/python
943faa7467a0: Mounted from library/python
c3f071bbaa4b: Pushed
1b023e321a47: Pushed
ed19f53fb005: Pushed
0a00f6ce5fb7: Mounted from library/python
7cc7fe68eff6: Mounted from library/python
b960a4e3e07f: Pushed
8e43399: digest: sha256:767a8ff6d54fd975c1492bc9e229d9ea4c6d69ea4b364552ba3bab37632d417b size: 1991
+ docker push ysm2820/simple-api:latest
The push refers to repository [docker.io/ysm2820/simple-api]
```

CI/CD Test

- console stdout

확인

```
c3f071bbba4b: Layer already exists
943faa7467a0: Layer already exists
0a00f6ce5fb7: Layer already exists
7cc7fe8eff6: Layer already exists
latest: digest: sha256:767a8ff6d54fd975c1492bc9e229d9ea4c6d69ea4b364552ba3bab37632d417b size: 1991
+ docker logout
Removing login credentials for https://index.docker.io/v1/
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // timeout
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Pull (remote))
[Pipeline] timeout
Timeout set to expire in 5 min 0 sec
[Pipeline] {
[Pipeline] sshagent
[ssh-agent] Using credentials smyang9777
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-abDjB075FBiR/agent.1273
SSH_AGENT_PID=1276
Running ssh-add (command line suppressed)
Identity added: /var/jenkins_home/workspace/test@tmp/private_key_8232160050000440382.key (root@03f22bbddfd)
[ssh-agent] Started.
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKER_PASS
[Pipeline] {
[Pipeline] sh
Warning: A secret was passed to "sh" using Groovy String interpolation, which is insecure.
Affected argument(s) used the following variable(s): [DOCKER_PASS]
See https://jenkins.io/redirect/groovy-string-interpolation for details.
+ ssh -o StrictHostKeyChecking=no smyang9777@34.172.222.28          (echo *** | docker login -u 'ysm2820' --password-stdin || true) &&
docker pull ysm2820/simple-api:8e43399
Login Succeeded

WARNING! Your credentials are stored unencrypted in '/home/smyang9777/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

8e43399: Pulling from ysm2820/simple-api
59e22667830b: Already exists
abd846fa1cddb: Already exists
b7b61708209a: Already exists
4085babcc570: Already exists
da500b9dbec1: Pulling fs layer
096d7a8e120e: Pulling fs layer
b6e68d70ffbd: Pulling fs layer
95f9fd475817: Pulling fs layer
95f9fd475817: Waiting
096d7a8e120e: Verifying Checksum
096d7a8e120e: Download complete
da500b9dbec1: Download complete
da500b9dbec1: Pull complete
096d7a8e120e: Pull complete
95f9fd475817: Verifying Checksum
95f9fd475817: Download complete
b6e68d70ffbd: Verifying Checksum
b6e68d70ffbd: Download complete
b6e68d70ffbd: Pull complete
95f9fd475817: Pull complete
Digest: sha256:767a8ff6d54fd975c1492bc9e229d9ea4c6d69ea4b364552ba3bab37632d417b
Status: Downloaded newer image for ysm2820/simple-api:8e43399
docker.io/ysm2820/simple-api:8e43399
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
```

CI/CD Test

- console stdout

확인

```
Status: Downloaded newer image for ysm2820/simple-api:8e43399
docker.io/ysm2820/simple-api:8e43399
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
$ ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 1276 killed;
[ssh-agent] Stopped.
[Pipeline] // sshagent
[Pipeline] }
[Pipeline] // timeout
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy (remote))
[Pipeline] timeout
Timeout set to expire in 5 min 0 sec
[Pipeline] {
[Pipeline] sshagent
[ssh-agent] Using credentials smyang9777
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-D0PawEVVTjn1/agent.1299
SSH_AGENT_PID=1302
Running ssh-add (command line suppressed)
Identity added: /var/jenkins_home/workspace/test@tmp/private_key_13379223122626633910.key (root@03f22bbddfd)
[ssh-agent] Started.
[Pipeline] {
[Pipeline] sh
+ ssh -o StrictHostKeyChecking=no smyang9777@34.172.222.28          (docker rm -f simple-api || true) &&           docker run -d --name
simple-api -p 3000:3000                                     ysm2820/simple-api:8e43399
simple-api
9b79f1a6872629ddeb36fcf5ecfceca232a62ce4a6f98b04ebf4dd67337ab10b
[Pipeline] }
$ ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 1302 killed;
[ssh-agent] Stopped.
[Pipeline] // sshagent
[Pipeline] }
[Pipeline] // timeout
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deploy 성공
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

CI/CD Test

- Dockerhub image push / 원격지 서버 Deploy 확인

The screenshot shows the Docker Hub interface. On the left, there's a sidebar for the user 'ysm2820' with options like 'Repositories', 'Collaborations', 'Settings', 'Default privacy', 'Notifications', and 'Billing'. The main area is titled 'Repositories' and shows a single entry: 'ysm2820/simple-api'. Below the entry, it says 'less than a minute ago' under 'Last Pushed', 'IMAGE' under 'Contains', 'Public' under 'Visibility', and 'Inactive' under 'Scout'.

```
smyang9777@instance-20250808-150304:~$ docker ps -a
CONTAINER ID        IMAGE           COMMAND                  CREATED             STATUS              PORTS
f07d4c9fb0f6        null            "uvicorn app.main:ap..."   3 minutes ago      Up 3 minutes       0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp
smyang9777@instance-20250808-150304:~$
```