## 가로선



설치 및 구성 메뉴얼

2023년 8월 19일

**─**

김정훈

# Jenkins 설치

## Jenkins 서버 설치

Jenkins 설치 전 준비물 : Ubuntu 20.04 환경에서 Docker 설치 완료

* <https://cloudhedgehog.tistory.com/3> (작성자 본인 기술 블로그)

**Docker 기본 패키지 설치**

sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

**Docker에서 제공하는 패키지 저장소 등록**

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.헿

**패키지 저장소 경로 GPG 키와 함께 등록**

echo \

"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

"$(. /etc/os-release && echo "$VERSION\_CODENAME")" stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

**등록한 패키지 저장소 적용**

sudo apt-get update

**Docker 설치**

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

**설치 후 확인**

docker -v

**vagrant가 docker 명령어를 사용할 수 있게 권한 수정**

sudo usermod -a -G docker vagrant

newgrp docker

groups

**Docker Hub 사이트에 로그인을 한 뒤 Docker Token 발급받기**

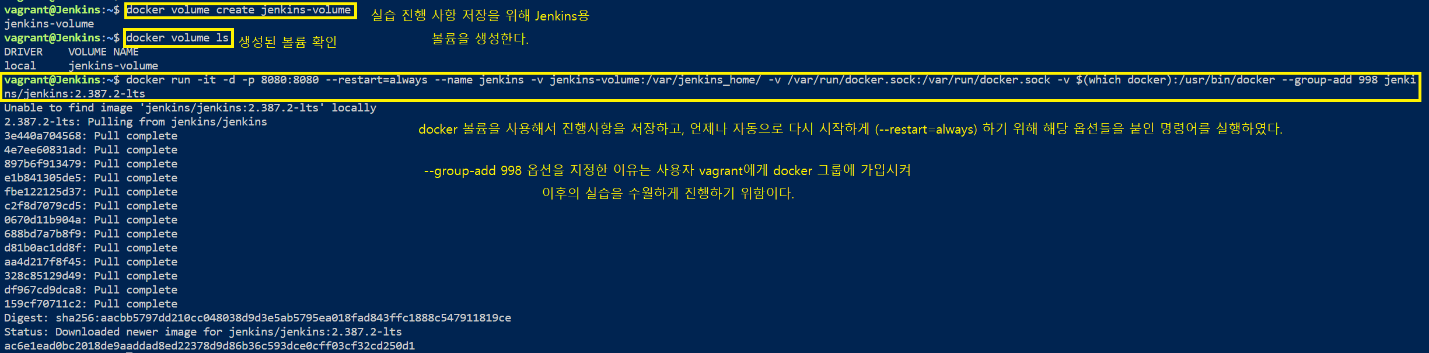
**Docker Token을 가져와서 로컬 파일로 저장하기**

생성된 Docker Token을 /vagrant/env/docker\_token 파일에 저장한다.

**Docker Token으로 로그인 성공하기**

docker login -u hedgehoon --password-stdin < /vagrant/env/docker\_token

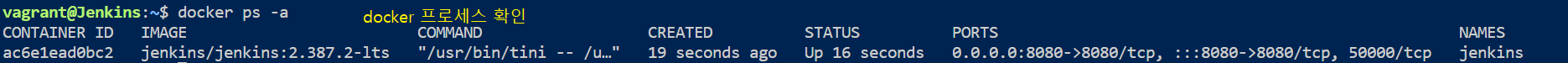
**Jenkins 실행하기 >>**



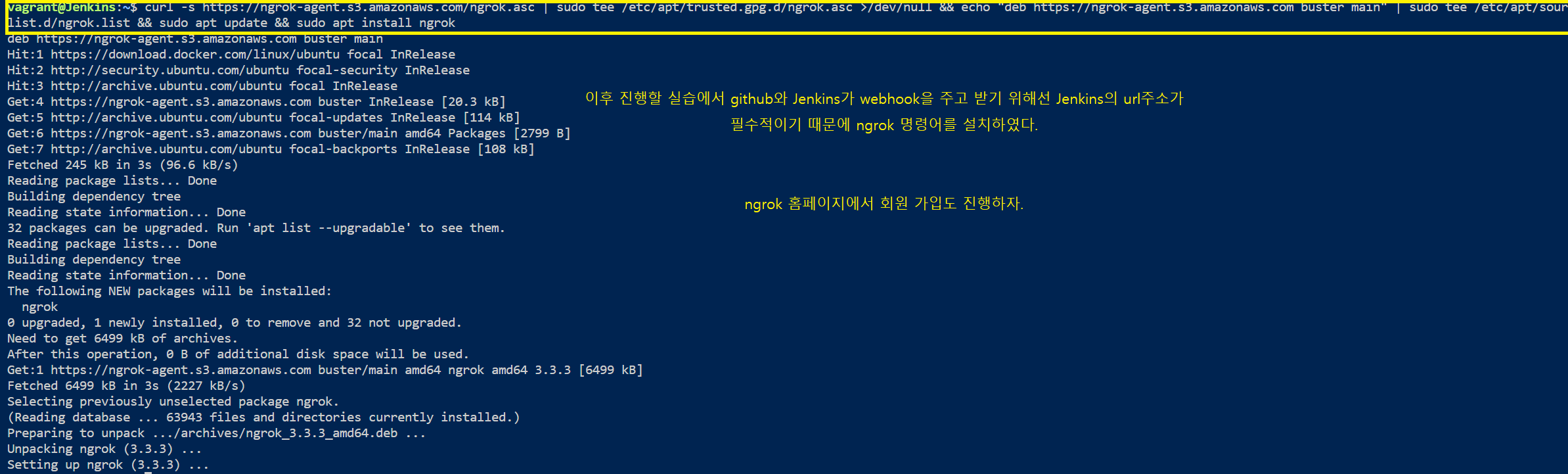
docker volume create jenkins-volume

docker volume ls

docker run -it -d -p 8080:8080 --restart=always --name jenkins -v jenkins-volume:/var/jenkins\_home/ -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/usr/bin/docker --group-add 998 jenkins/jenkins:2.387.2-lts



docker ps -a



curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list && sudo apt update && sudo apt install ngrok



바로 이전 스크린샷에서 ngrok 홈페이지에 가입해야 하는 이유가 여기에 있다.

토큰을 발급받아야 하기 때문이다.

<https://ngrok.com/> 🡪 여기서 로그인 진행. Your AuthToken에 개별 토큰이 생성된다.

아무튼 위 스크린샷의 명령어로 사용자의 Auth Token이 Jenkins 가상 서버에 저장이 된다.

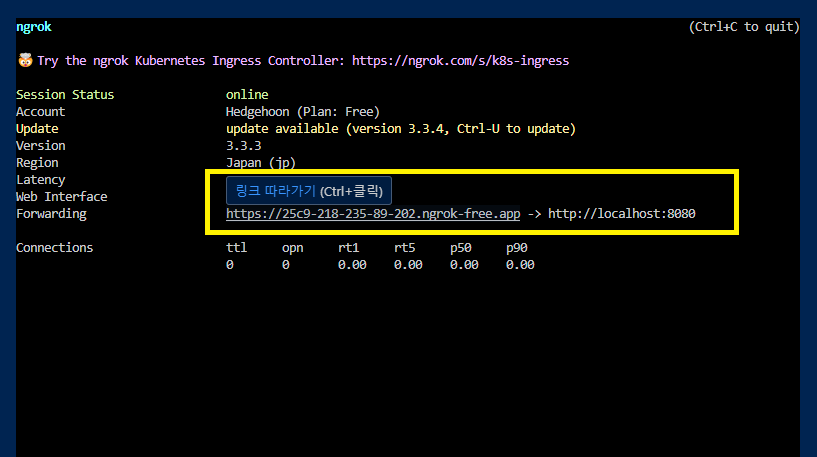


저장된 ngrok 토큰 확인 명령어.

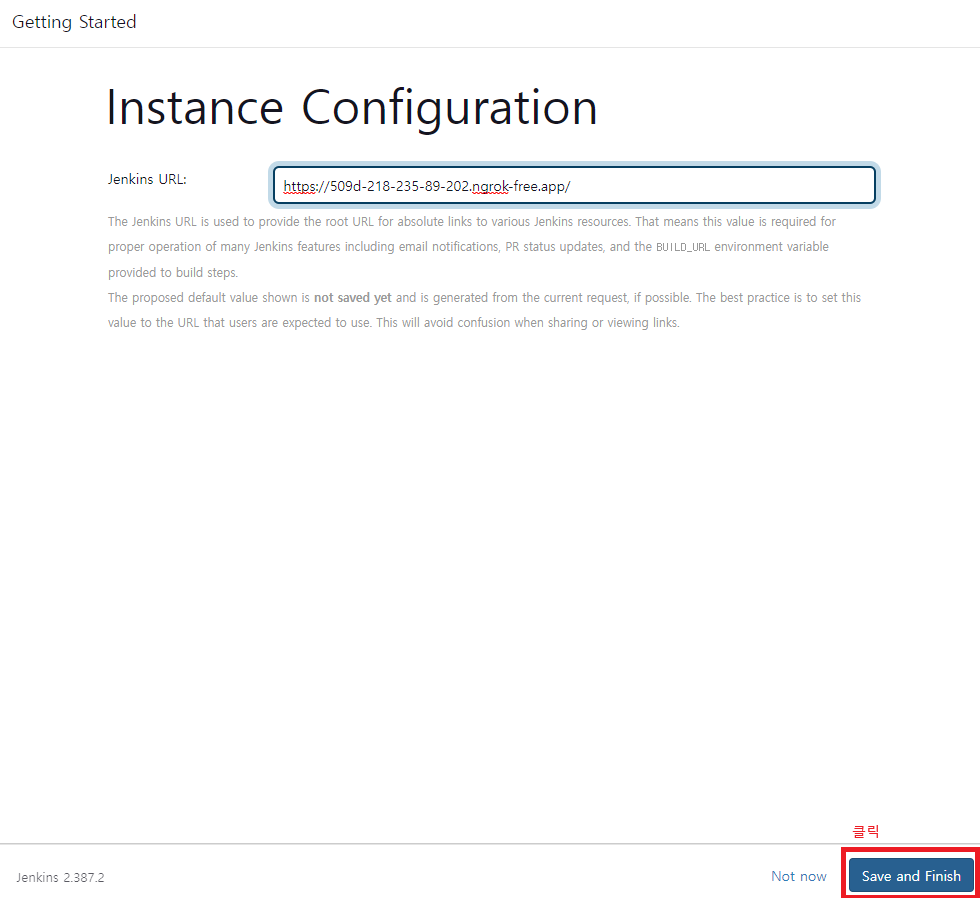
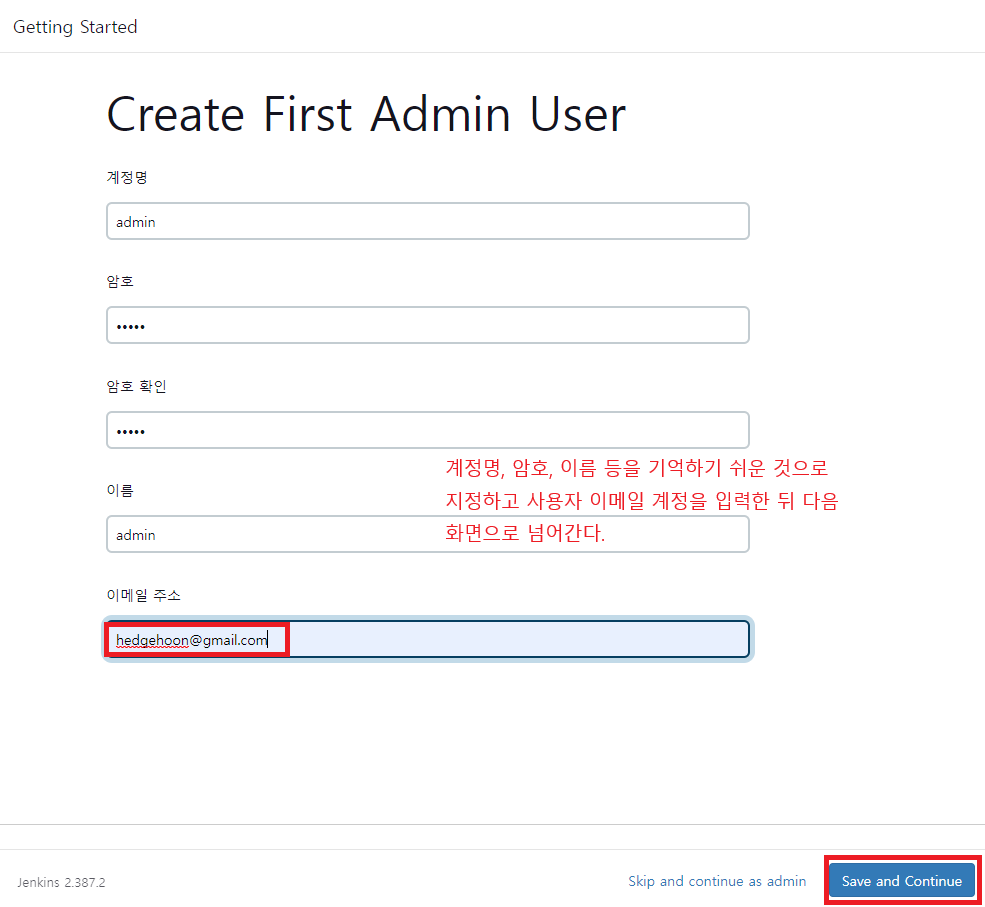
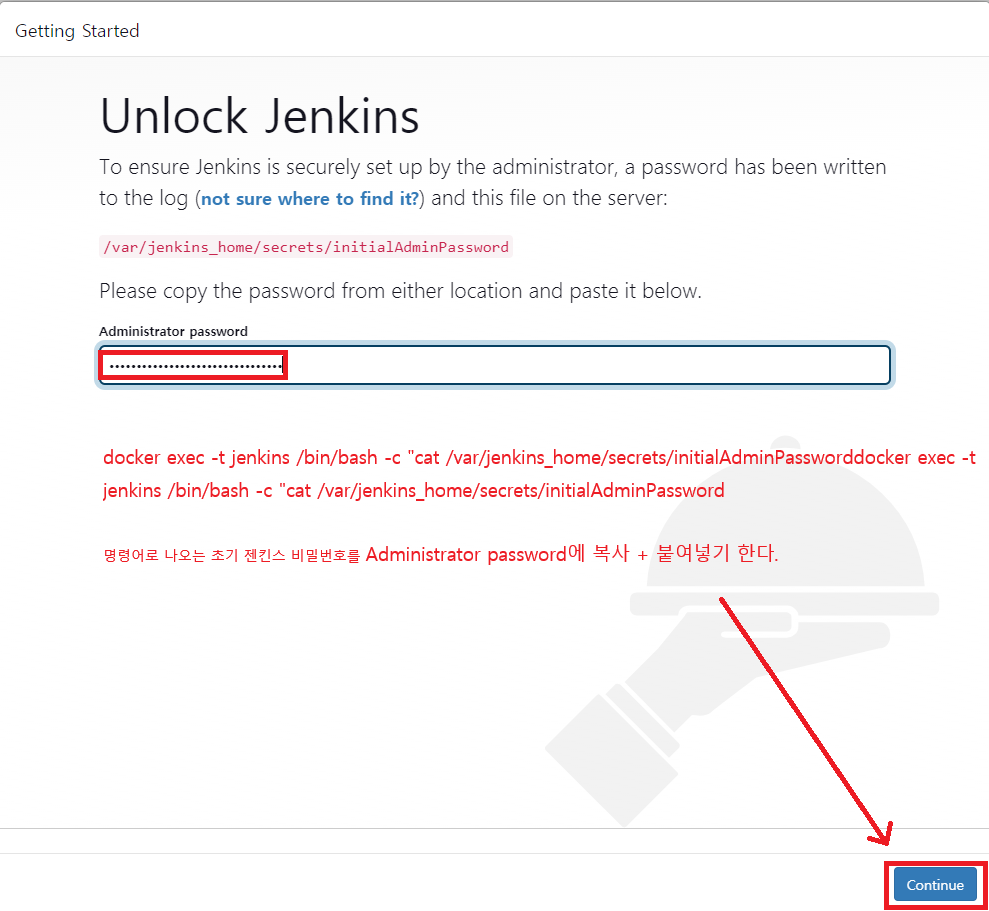


ngrok로 Jenkins를 url로 접근을 위한 명령어

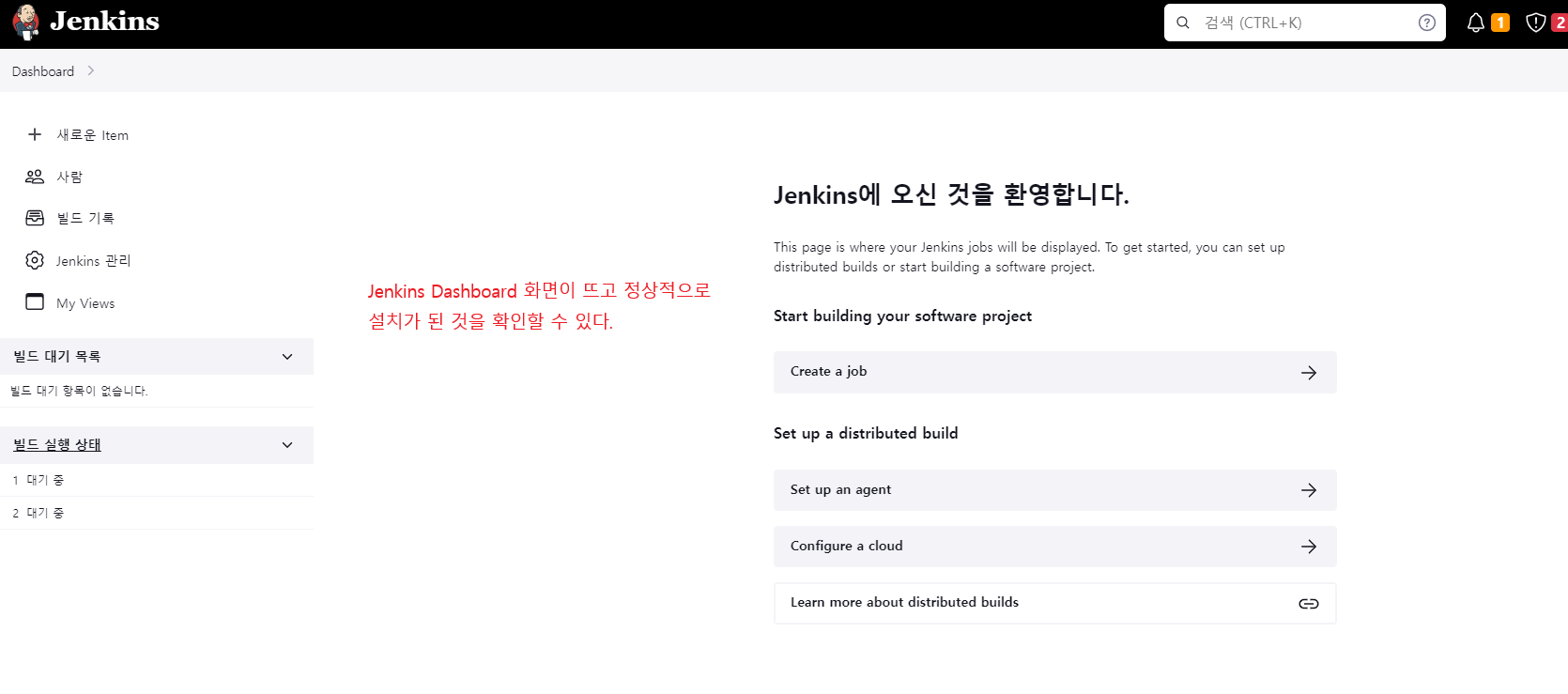




Ctrl 키를 누른 상태에서 마우스로 클릭을 해서 Jenkins를 웹 브라우저에서 사용해보자.



## 

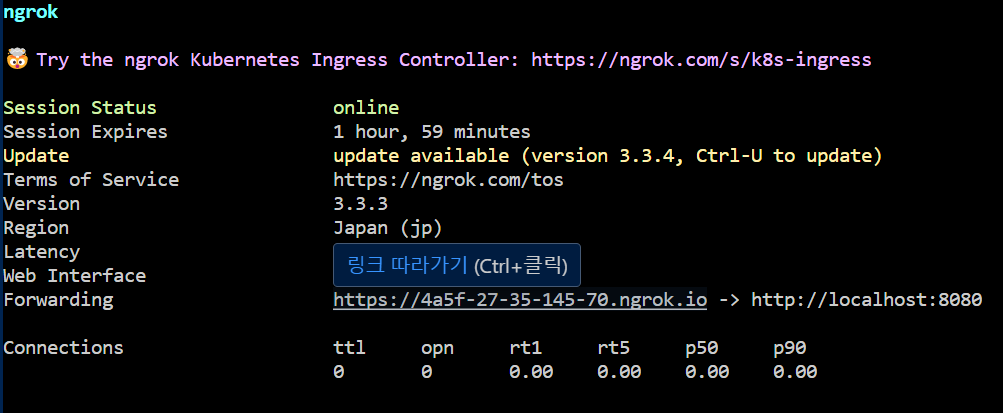


## 2. Jenkins 서비스 재시작

작성자 본인은 초반에 볼륨을 생성하고 도커 컨테이너를 실행할 때 –restart=always 옵션을 넣었으므로 매번 Jenkins 서버의 전원을 켤 때마다 컨테이너를 docker run할 필요가 없다. 따라서,



Ngrok 명령어를 통해 나타나는 Url을 통해 Jenkins 서비스를 재이용하자.



# Ubuntu 20.04 환경에서 SonarQube 설치하기

<https://cloudhedgehog.tistory.com/22> (본인 개인 기술 블로그)

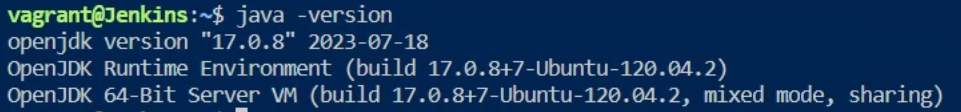
## 1. Java 17 설치

**sudo apt update**

**sudo apt install openjdk-17-jdk**

**java -version**

# 설치 완료 확인



2. SonarQube 다운로드 및 설치

**sudo curl -O https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.0.65466.zip**

**sudo unzip sonarqube-9.9.0.65466.zip -d /opt/**

3. SonarQube 서비스 설정

**sudo tee /etc/systemd/system/sonarqube.service << EOF**

[Unit]

Description=SonarQube service

After=syslog.target network.target

[Service]

Type=forking

ExecStart=/opt/sonarqube-9.9.0.65466/bin/linux-x86-64/sonar.sh start

ExecStop=/opt/sonarqube-9.9.0.65466/bin/linux-x86-64/sonar.sh stop

User=sonarqube

Group=sonarqube

Restart=always

[Install]

WantedBy=multi-user.target

EOF

4. SonarQube 사용자 및 그룹 생성

**sudo adduser --system --no-create-home --group sonarqube**

5. 권한 설정

**sudo chown -R sonarqube:sonarqube /opt/sonarqube-9.9.0.65466**

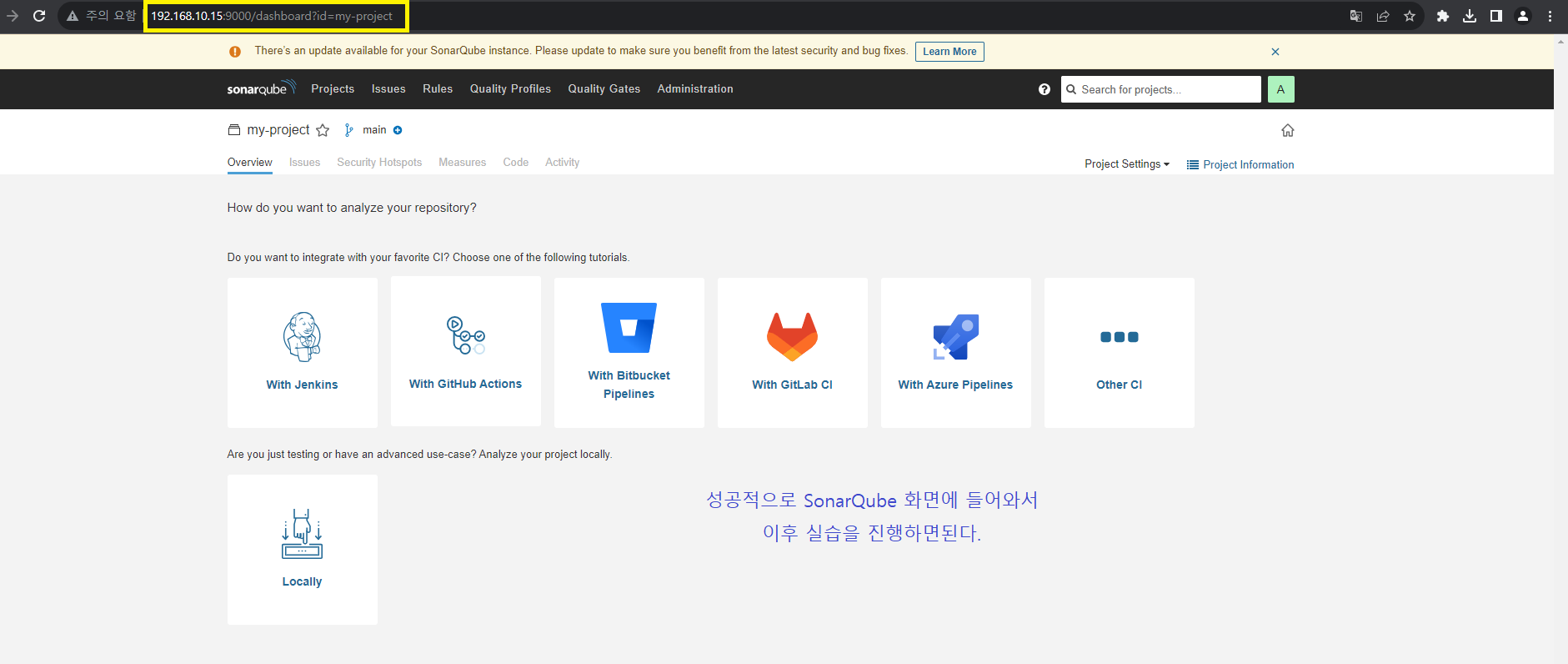
6. 서비스 시작 및 활성화

**sudo systemctl start sonarqube**

**sudo systemctl enable sonarqube**

7. SonarQube 웹 인터페이스 확인

**http://<your-jenkins-server-ip>:9000**

****

# NFS 설치

## NFS 서비스 구성

<https://cloudhedgehog.tistory.com/20> (출처: 본인 기술 블록그)

**NFS 서버 >>**

**1. 시스템 패키지 데이터베이스 업데이트 및 NFS 서버 패키지 설치**

sudo apt-get update

sudo apt-get install nfs-kernel-server

**2. NFS 서버 내의 Jenkins 서버 마운트 포인트 구축**

sudo mkdir /nfs\_jenkins

sudo chown nobody:nogroup /nfs\_jenkins

**3. NFS 서버에서 공유할 디렉토리와 클라이언트에 대한 액세스 권한 설정**

sudo vi /etc/exports

/nfs\_jenkins <Jenkins\_서버\_IP>(rw,sync,no\_subtree\_check,no\_root\_squash)

내용 추가

sudo exportfs -a

**4. NFS 서비스 재시작**

sudo systemctl restart nfs-kernel-server

**Jenkins 서버 >>**

**1. NFS 클라이언트 서비스 확인 및 설치**

dpkg -l | grep nfs-common

sudo apt-get install nfs-common

**2. NFS 서버와 연결할 마운트 포인트 생성**

sudo mkdir -p /mnt/nfs\_jenkins

sudo mount -t nfs <NFS\_서버\_IP>:/nfs\_jenkins /mnt/nfs\_jenkins

**3. 도커 볼륨 생성 및 마운트 포인트와 연결**

docker volume create --driver local --opt type=nfs --opt o=addr=<NFS\_서버\_IP>,rw --opt device=:/nfs\_jenkins jenkins-volume

**4. Jenkins 실행**

docker run -it -d -p 8080:8080 --restart=always --name jenkins -v jenkins-volume:/var/jenkins\_home/ -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/usr/bin/docker --group-add 998 jenkins/jenkins:2.387.2-lts

docker exec -it jenkins /bin/bash

# 실행해서 파일이 공유됐는지 확인

NFS 서버의 /nfs\_jenkins 에 들어가서 Jenkins 관련 파일이 공유되었는지 확인하자.