Lab. CI/CD 환경 구성하기 using Jenkins

1. Lab 환경 구성

1)SSH Key pair
-Name : user-00-key
-키 페어 유형 : RSA
-프라이빗 키 파일 형식 : .pem

2)Security Group 생성
-Name : user-00-sg
-설명 : Security group for Jenkins Controller
-VPC : Default VPC
-인바운드 규칙
--유형 : 모든 트래픽
--소스 : 0.0.0.0/0

3)AWS EC2 Instance
-Name : User-00-Jenkins-Controller
-OS : Ubuntu 22.04 LTS, 64-bit(x86)
-Instance Type : t2.medium
-Key pair : user-00-key
-Network > 편집
--서브넷 : ap-northeast-2a
--퍼블릭 IP 자동 할당 : 활성화
--방화벽(보안 그룹) > 기존 보안 그룹 선택 : user-00-sg
-Storage : 30GB

4)SSH EC2 Instance 연결 후 해야할 작업
-Docker 설치
--https://docs.docker.com/engine/install/ubuntu/

```
    # Add Docker's official GPG key:
        $ sudo apt-get update
        $ sudo apt-get install ca-certificates curl gnupg
        $ 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

    # Add the repository to Apt sources:
        $ 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
        $ sudo apt-get update

    # Install the Docker packages
        $ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
        $ sudo usermod -aG docker $USER
        $ logout
```

-jdk 설치
--Correctto 17 Installation
--Ref : https://docs.aws.amazon.com/corretto/latest/corretto-17-ug/generic-linux-install.html#debian-install-instruct

```
    $ wget -O - https://apt.corretto.aws/corretto.key | sudo gpg --dearmor -o /usr/share/keyrings/corretto-keyring.gpg && ₩
echo "deb [signed-by=/usr/share/keyrings/corretto-keyring.gpg] https://apt.corretto.aws stable main" | sudo tee /etc/apt/sources.list.d/corretto.list
```

```
          $  sudo apt-get update; sudo apt-get install -y java-17-amazon-corretto-jdk

          $ java --version
          openjdk 17.0.10 2024-01-16 LTS
          OpenJDK Runtime Environment Corretto-17.0.10.7.1 (build 17.0.10+7-LTS)
          OpenJDK 64-Bit Server VM Corretto-17.0.10.7.1 (build 17.0.10+7-LTS, mixed mode, sharing)

          $ javac --version
          javac 17.0.10
```

-MySQL 설치

```
          $ docker container run -d ₩
--name=mysqldb ₩
--restart=always ₩
-e MYSQL_ROOT_PASSWORD=education ₩
-e MYSQL_DATABASE=guestbook ₩
-p 3306:3306 ₩
yu3papa/mysql_hangul:2.0
```

5)MySQL Workbench 설치 후 연결하기

-Hostname : EC2 Instance Public IP
-Port : 3306
-Username : root
-Password : education
-Default Schema : guestbook

2. 수동 빌드 및 배포

1)lab 압축 파일 다운로드 및 압축해제

```
        $ wget https://github.com/swacademy/fss/raw/main/CICD/guestbook.zip
        $ unzip guestbook.zip
        Command 'unzip' not found, but can be installed with:
        sudo apt install unzip

        $ sudo apt install unzip
        $ unzip guestbook.zip
        $ chmod u+x mvnw
        $ ./mvnw clean package
        ...
        ...
        [INFO] Replacing main artifact with repackaged archive
        [INFO] ------------------------------------------------------------------------
        [INFO] BUILD SUCCESS
        [INFO] ------------------------------------------------------------------------
        [INFO] Total time:  43.621 s
        [INFO] Finished at: 2024-01-24T12:28:43Z
        [INFO] ------------------------------------------------------------------------

        $ ls -l target/*.jar
        -rw-rw-r-- 1 ubuntu ubuntu 25061688 Jan 24 12:28 target/guestbook-0.0.1-SNAPSHOT.jar
```

2)Docker Image 작성하기

```
         $ docker image build -t guestbook .
         $ docker image ls
        REPOSITORY    TAG        IMAGE ID        CREATED            SIZE
        guestbook     latest     d98cb618f1b2    11 seconds ago    497MB
        mysql         5.7        5107333e08a8    6 weeks ago       501MB
```

3)Docker Hub에 Upload 하기

-Docker Hub에 Repository 'guestbook' 생성하기

```
    $ docker login
    Username:
    Password:

    Login Succeeded

    $ docker tag guestbook {{docker hub account}}/guestbook:v1
    $ docker image ls
    REPOSITORY              TAG        IMAGE ID        CREATED          SIZE
    devtrainer/guestbook    v1         d98cb618f1b2    7 minutes ago    497MB
    guestbook               latest     d98cb618f1b2    7 minutes ago    497MB

    $ docker push {{docker hub account}}/guestbook:v1
```

4)Docker Container로 방명록 웹어플리케이션 실행

```
    $ docker container run ₩
--name=guestbookapp ₩
--rm ₩
--network=host ₩
-e MYSQL_IP=172.17.0.2 ₩
-e MYSQL_PORT=3306 ₩
-e MYSQL_DATABASE=guestbook ₩
-e MYSQL_USER=root ₩
-e MYSQL_PASSWORD=education ₩
{{docker hub account}}/guestbook:v1


  .   ____          _            __ _ _
 /₩₩ / ___'_ __ _ _(_)_ __  __ _ ₩ ₩ ₩ ₩
( ( )₩___ | '_ | '_| | '_ ₩/ _` | ₩ ₩ ₩ ₩
 ₩₩/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_₩__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.6)

2024-01-24 13:07:12.893  INFO 1 --- [           main] c.j.guestbook.GuestbookApplication       : Starting GuestbookApplication v0.0.1-SNAPSHOT using Java 17.0.2 on ip-172-31-13-45 with PID 1 (/app/guestbook.jar started by root in /app)
2024-01-24 13:07:12.897  INFO 1 --- [           main] c.j.guestbook.GuestbookApplication       : No active profile set, falling back to 1 default profile: "default"
2024-01-24 13:07:14.578  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 80 (http)
2024-01-24 13:07:14.595  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-01-24 13:07:14.596  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.60]
2024-01-24 13:07:14.689  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-01-24 13:07:14.690  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1654 ms
2024-01-24 13:07:15.359  INFO 1 --- [           main] o.s.b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page template: index
2024-01-24 13:07:15.494  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-01-24 13:07:15.670  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-01-24 13:07:15.852  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 80 (http) with context path ''
2024-01-24 13:07:15.863  INFO 1 --- [           main] c.j.guestbook.GuestbookApplication       : Started GuestbookApplication in 3.661 seconds (JVM running for 4.246)
```

5)EC2 Instance Public 주소로 웹 접속 후 방명록 샘플 웹어플리케이션 동작 확인 후 컨테이너 종료

http://public ip

```
$ Ctrl + C
```

6)MySQL Workbench에서 확인

-guestbook > post table 확인

3. Jenkins 설치

-다양한 설치 방법 가능

1)Jenkins WAR 방식으로 실행

-java -jar jenkins.jar

2)Docker로 설치하기

-https://www.jenkins.io/download/ <---최신 버전 확인하기

※java와 javac 현재 버전 확인

```
$ sudo update-alternatives --config java
$ sudo update-alternatives --config javac
$ java --version
openjdk 17.0.10 2024-01-16 LTS
OpenJDK Runtime Environment Corretto-17.0.10.7.1 (build 17.0.10+7-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.10.7.1 (build 17.0.10+7-LTS, mixed mode, sharing)
$ javac --version
javac 17.0.10
```

ⓐ $ docker run --name jk -p 8080:8080 -p 50000:50000 -v /var/run/docker.sock:/var/run/docker.sock -u root --restart=on-failure jenkins/jenkins:lts-jdk17

```
...
...

*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

edbd384ca13a400b9a9d1f71d35efd93

This may also be found at: /home/ubuntu/.jenkins/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************
```

-Tabby의 별도의 세션을 오픈하여

```
$ docker logs jk
```

-initialAdminPassword의 값 복사

-메모장에 붙여넣기 할 것

-http://EC2 Instance Public IP:8080 접속 후

Administrator password : <----- Admin 초기 패스워드 붙여넣기

ⓑ Plugin Install 페이지에서, [Select plugins to install] 선택 > [Getting Started] 페이지에서, 상단의 [None] 선택 > [Install] 버튼 클릭

ⓒ [Create First Admin User]에서

-계정명 : admin

-암호 : education

-이름 :
-값 입력 후 [Save and Continue] 클릭

ⓓ [Instance Configuration] 페이지에서, [Save and Finish] 클릭 > [Start using Jenkins] 클릭

```
ⓔ $ docker exec -it jk bash
ⓕ # cd /var/jenkins_home/plugins/
ⓖ # curl https://henry-jenkins.s3.ap-northeast-2.amazonaws.com/jenkins-plugin-jpi.tar.gz --output jenkins-plugin-jpi.tar.gz
ⓗ # tar xvfz jenkins-plugin-jpi.tar.gz
ⓘ # exit
ⓙ $ docker stop jk
ⓚ $ docker start jk
```

ⓛ Jenkins logout --> Jenkins Login
ⓜ Jenkins의 Dashboard > Jenkins 관리 > Plugins > Installed plugins에서 확인

3)Jenkins Debian Packages

-https://www.jenkins.io/download/

```
$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] ₩
https://pkg.jenkins.io/debian-stable binary/ | sudo tee ₩
/etc/apt/sources.list.d/jenkins.list > /dev/null
$ sudo apt-get update
$ sudo apt-get install fontconfig openjdk-17-jre
$ sudo apt-get install jenkins
```

-http://PUBLIC_IP:8080

```
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
    54aeaa121edf48caa40b674c2010c3d2    <---복사해서 브라우저에 붙여넣기
```

--[Customize Jenkins] 페이지에서, [Install suggested plugins] 선택
--[Create First Admin User]에서
-계정명 : admin
-암호 : education
-이름 :
-이메일 주소 :
-값 입력 후 [Save and Continue] 클릭
--[Save and Finish]
--[Start using Jenkins]

4. Jenkins 체험하기

1)첫번째 Pipeline Project

-Jenkins 페이지에서 [새로운 Item] > [Enter an item name] : guestbook > [Pipeline] 선택 > [OK] 버튼 클릭
-[General] > [Pipeline] > [try sample Pipleline..]에서 [Hello World] 선택
-[저장] 버튼 클릭
-Project 페이지창에서, 왼쪽 메뉴의 [지금 빌드(Build Now)] 클릭
-[#1] 링크 클릭 > [Console Output] 클릭하여 Finishged: SUCCESS 확인

```
Started by user Henry
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/guestbook
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
```

```
            [Pipeline] // node
            [Pipeline] End of Pipeline
            Finished: SUCCESS
```

-Build 후 Jenkins Directory 관찰

--/var/lib/jenkins/jobs/guestbook/builds/1/에서 확인 가능

```
build.xml  log  log-index  workflow-completed
```

5. SCM(Software Change Management; 형상관리) - GIT & GitHub

1)git version 확인

2)현재 디렉토리를 git repository 등록하기

```
$ git init
```

3)사용자 등록

```
$ git config --global user.name 'Henry'
$ git config --global user.email 'javaexpert@nate.com'
$ git config --global credential.helper cache
$ cat ~/.gitconfig
[user]
        name = Henry
        email = javaexpert@nate.com
[credntial]
        helper = cache
```

4)방명록 샘플 웹애플리케이션 Maven Build

```
$ ./mvnw package
...
...
[INFO] Replacing main artifact with repackaged archive
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  43.621 s
[INFO] Finished at: 2024-01-24T12:28:43Z
[INFO] ------------------------------------------------------------------------
```

5)방명록 샘플 웹애플리케이션 실행

```
    $ export MYSQL_IP=172.31.7.28     <---현재 Private IP address
    $ export MYSQL_PORT=3306
    $ export MYSQL_DATABASE=guestbook
    $ export MYSQL_USER=root
    $ export MYSQL_PASSWORD=education

    $ docker rm -f mysqldb
    $ docker container run -d ₩
--name=mysqldb ₩
--restart=always ₩
-e MYSQL_ROOT_PASSWORD=education ₩
-e MYSQL_DATABASE=guestbook ₩
-p 3306:3306 ₩
yu3papa/mysql_hangul:2.0

        $ docker ps -a

    $ java -Dserver.port=9090 -jar target/guestbook-0.0.1-SNAPSHOT.jar
      .   ____          _            __ _ _
     /₩₩ / ___'_ __ _ _(_)_ __  __ _ ₩ ₩ ₩ ₩
    ( ( )₩___ | '_ | '_| | '_ ₩/ _` | ₩ ₩ ₩ ₩
     ₩₩/  ___)| |_)| | | | | || (_| |  ) ) ) )
      '  |____| .__|_| |_|_| |_₩__, | / / / /
```

```
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.6)

2024-01-26 03:17:19.932  INFO 39693 --- [           main] c.j.guestbook.GuestbookApplication       : Starting GuestbookApplication v0.0.1-SNAPSHOT using Java 17.0.10 on ip-172-31-13-45 with PID 39693 (/home/ubuntu/target/guestbook-0.0.1-SNAPSHOT.jar started by ubuntu in /home/ubuntu)
2024-01-26 03:17:19.937  INFO 39693 --- [           main] c.j.guestbook.GuestbookApplication       : No active profile set, falling back to 1 default profile: "default"
2024-01-26 03:17:22.182  INFO 39693 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 9090 (http)
2024-01-26 03:17:22.200  INFO 39693 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-01-26 03:17:22.200  INFO 39693 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.60]
2024-01-26 03:17:22.401  INFO 39693 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-01-26 03:17:22.402  INFO 39693 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2317 ms
2024-01-26 03:17:23.311  INFO 39693 --- [           main] o.s.b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page template: index
2024-01-26 03:17:23.482  INFO 39693 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-01-26 03:17:23.751  INFO 39693 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-01-26 03:17:24.012  INFO 39693 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 9090 (http) with context path ''
2024-01-26 03:17:24.027  INFO 39693 --- [           main] c.j.guestbook.GuestbookApplication       : Started GuestbookApplication in 4.959 seconds (JVM running for 5.85)
```

6)Web Browser에서 확인

http://Public IP:9090

7)또 다른 세션을 오픈하여 git 상태확인

```
$ git status
```

8)형상관리 대상 등록하기

```
$ git add .

$ git status
```

9)형상관리 대상 등록 후 파일 수정

-src/main/resources/templates/index.html 파일의 <title> 내용을 <title>방명록 (K8S)</title> —▶ <title>GUESTBOOK</title>

10)GIT 저장소 워킹트리의 상태를 확인

```
$ git status
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/main/resources/templates/index.html
```

11)재 등록

```
$ cd ~
$ git add src/main/resources/templates/index.html
$ git status
```

12)형상관리 대상 등록 취소

```
$  git status src/main/resources/templates/index.html
On branch master

No commits yet
```

```

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   src/main/resources/templates/index.html

$ git rm --cached src/main/resources/templates/index.html
rm 'src/main/resources/templates/index.html'

$  git status src/main/resources/templates/index.html
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        src/main/resources/templates/index.html

nothing added to commit but untracked files present (use "git add" to track)
```

13)다시 index.html을 tracked 상태로 변경하기

```
$ git add src/main/resources/templates/index.html
$ git status src/main/resources/templates/index.html
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
new file:   src/main/resources/templates/index.html
```

14)commit 하기

```
$ git commit -m "[INITIAL] 프로젝트 최초 등록"
```

15)commit 후에 GIT 저장소 상태 확인

```
$ git status
On branch master
nothing to commit, working tree clean
```

16)commit 로그 확인

-첫번째 commit의 로그를 확인

```
$ git log
commit 60b5e9e14057dac375d3e10a703cd509895a1fd9 (HEAD -> master)
Author: Henry <javaexpert@nate.com>
Date:   Fri Jan 26 03:34:21 2024 +0000

    [INITIAL] 프로젝트 최초 등록
```

-특정 파일의 commit 로그를 확인하려면 해당 파일을 지정하면 됨.

-src/main/resources/templates/index.html 파일의 commit 로그를 확인

```
$ git log src/main/resources/templates/index.html
Author: Henry <javaexpert@nate.com>
Date:   Fri Jan 26 03:34:21 2024 +0000

    [INITIAL] 프로젝트 최초 등록
```

17)마지막 commit message 수정

-마지막 commit message에 오늘 날짜를 추가하여 수정하고, commit 로그를 확인

```
$ git commit --amend
[INITIAL] 프로젝트 최초 등록(2024.01.26)    <---오늘 날짜 등록
```

-Nano 편집기 저장 후 닫기

-commit message 수정 확인

```
$ git log
commit 082dc13eeb00631ffcd6a06d511c199fdd0beee8 (HEAD -> master)
Author: Henry <javaexpert@nate.com>
Date:   Fri Jan 26 03:34:21 2024 +0000

    [INITIAL] 프로젝트 최초 등록(2024.01.26)       <--------수정 내용 확인
```

18)특정 commit의 상세 정보 확인

```
$ git show COMMIT-ID

commit 082dc13eeb00631ffcd6a06d511c199fdd0beee8 (HEAD -> master)
Author: Henry <javaexpert@nate.com>
Date:   Fri Jan 26 03:34:21 2024 +0000

    [INITIAL] 프로젝트 최초 등록(2024.01.26)

diff --git a/.bash_history b/.bash_history
new file mode 100644
index 0000000..db27ee2
--- /dev/null
+++ b/.bash_history
...
...
```

19)리눅스 쉘 프롬프트에서 git branch 표시하기

```
$ nano ~/.bashrc
    parse_git_branch() {
        git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* ₩(.*₩)/ (₩1)/'
    }
    export PS1="[₩u@₩h ₩[₩033[32m₩]₩W₩[₩033[33m₩]₩$(parse_git_branch)₩[₩033[00m₩]]$ "
```

-로그아웃하고 쉘에 다시 로그인한 변경내역 확인

```
    $ logout

Last login: Fri Jan 26 03:20:48 2024 from 182.208.131.42
[ubuntu@ip-172-31-13-45 ~ (master)]$     <----변경사항 확인
```

20)원격저장소 Github에 guestbook Repository 생성

-[Create a new repository]
- --[Repository name] : guestbook
- --Public
- --[Add a README file] no check
- --[Create repository] 버튼 클릭

```
git remote add origin https://github.com/infinian/guestbook.git
git branch -M main
git push -u origin main
```

21)GITHUB “Personal Access Token” 발급

-2021년 8월 13일 이후로 ID/Password를 인증을 통한 Git Operation 만료
https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/

-비밀번호를 대체함과 동시에 각 Token의 권한을 조절하고, Token의 만료 날짜를 지정할 수 있어 보안 강화

-Token 발급

① https://github.com 로그인 후 https://github.com/settings/tokens URL로 이동후
[Personal access tokens] > Tokens(classic) > [Generate new token] 버튼 클릭 > [Generate news token(classic)] 선택
[Note] : guestbook_auth

② “만료기한” 선택, Select Scope는 repo, user 체크 < [Generate Token] 클릭

③ 반드시 꼭 꼭 패스워드 복사하여 별도 파일로 저장할 것 ==> guestbook_auth.txt

22)GITHUB(원격 저장소) 앨리어스 등록/변경/삭제

-등록 및 확인

```
$ git remote add origin https://github.com/infinian/guestbook.git
$ git remote
origin

$ git remote -v
origin  https://github.com/infinian/guestbook.git (fetch)
origin  https://github.com/infinian/guestbook.git (push)
```

-origin 원격 저장소의 상세 정보 확인

```
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/infinian/guestbook.git
  Push  URL: https://github.com/infinian/guestbook.git
  HEAD branch: (unknown)
```

-Git을 원격 저장소에 연결하면 ./git/config 파일 안에 Remote 연결에 대한 설정 정보가 자동으로 추가됨.

```
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "origin"]
    url = https://github.com/infinian/guestbook.git
    fetch = +refs/heads/*:refs/remotes/origin/*
```

-GITHUB 원격저장소로 커밋된 파일들을 PUSH 수행

```
$ git branch -M main
(main)$ git push -u origin main
Username for 'https://github.com': infinian             <---Github ID 또는 Email
Password for 'https://infinian@github.com':             <---guestbook_auth 키 값
Enumerating objects: 5752, done.
Counting objects: 100% (5752/5752), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4567/4567), done.
Writing objects: 100% (5752/5752), 540.12 MiB | 14.35 MiB/s, done.
Total 5752 (delta 1055), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1055/1055), done.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File jenkins-plugin-jpi.tar.gz is 86.81 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: File jenkins/jenkins.war is 90.55 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/infinian/guestbook.git
 * [new branch]       main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

-Github에 접속하여 정상적으로 PUSH가 되었는지 확인

-https://github.com/infinian/guestbook

6. SCM 체크아웃

1)guestbook Project 왼쪽 메뉴에서 [Pipeline Syntax] 선택

2)[Snippet Generator] > Overview

-[Steps]

--[Sample Step] : git: Git

--[Repository URL] : https://github.com/infinian/guestbook.git

--[Branch] : main
--[Generate Pipeline Script] 버튼 클릭
--git branch: 'main', url: 'https://github.com/infinian/guestbook.git'   <---이 라인을 복사

3)guestbook Project 왼쪽 메뉴에서 [구성] 선택
4)Pipeline 섹션에서
-Hello World 선택 후 자동으로 생성되는 Script에서

```
pipeline {
    agent any
    stages {
        stage('Checkout') {
            steps {
                // Get some code from a GitHub repository
                git (branch: 'main' , url:'https://github.com/infinian/guestbook.git')
            }
        }
    }
}
```

5)[저장] 버튼 클릭
6)[지금 빌드] 클릭
-Console Output

```
Started by user Henry
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/guestbook
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
The recommended git tool is: git
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/infinian/guestbook.git
 > git init /var/jenkins_home/workspace/guestbook # timeout=10
Fetching upstream changes from https://github.com/infinian/guestbook.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > git fetch --tags --force --progress -- https://github.com/infinian/guestbook.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/infinian/guestbook.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 082dc13eeb00631ffcd6a06d511c199fdd0beee8 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 082dc13eeb00631ffcd6a06d511c199fdd0beee8 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git checkout -b main 082dc13eeb00631ffcd6a06d511c199fdd0beee8 # timeout=10
Commit message: "[INITIAL] 프로젝트 최초 등록(2024.01.26)"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

7)Jenkins Containier 로 들어가서 확인

```
$ cd /var/lib/jenkins/workspace/guestbook
$ ls
Dockerfile  guestbook.zip  mvnw  mvnw.cmd  pom.xml  src
```

-프로젝트 workspace 디렉토리에 git clone 완료됨.

7. SW Build 자동화

1)guestbook Project 왼쪽 메뉴에서 [구성] 선택

2)Pipeline 섹션에서

-위에서 생성한 스크립트 수정

```
pipeline {
    agent any
    stages {
        stage('Checkout') {
            steps {
                // Get some code from a GitHub repository
                git (branch: 'main' , url:'https://github.com/infinian/guestbook.git')
            }
        }
        stage('Build') {
            steps {
                sh "./mvnw -Dmaven.test.failure.ignore=true clean package"
            }

            post {
                success {
                    archiveArtifacts 'target/*.jar'
                }
            }
        }
    }
}
```

3)[저장] 버튼 클릭

4)[지금 빌드] 클릭

-Console Output

```
...
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/28.2-android/guava-28.2-android.jar (2.6 MB at 3.5 MB/s)
[INFO] Replacing main artifact with repackaged archive
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  42.002 s
[INFO] Finished at: 2024-01-26T07:26:07Z
[INFO] ------------------------------------------------------------------------
Post stage
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

5)결과 확인

```
$ cd /var/lib/jenkins/workspace/guestbook/target
$ ls
classes  generated-sources  guestbook-0.0.1-SNAPSHOT.jar  guestbook-0.0.1-SNAPSHOT.jar.original
```

```
maven-archiver   maven-status
```

-Build 결과물인 guestbook-0.0.1-SNAPSHOT.jar 존재