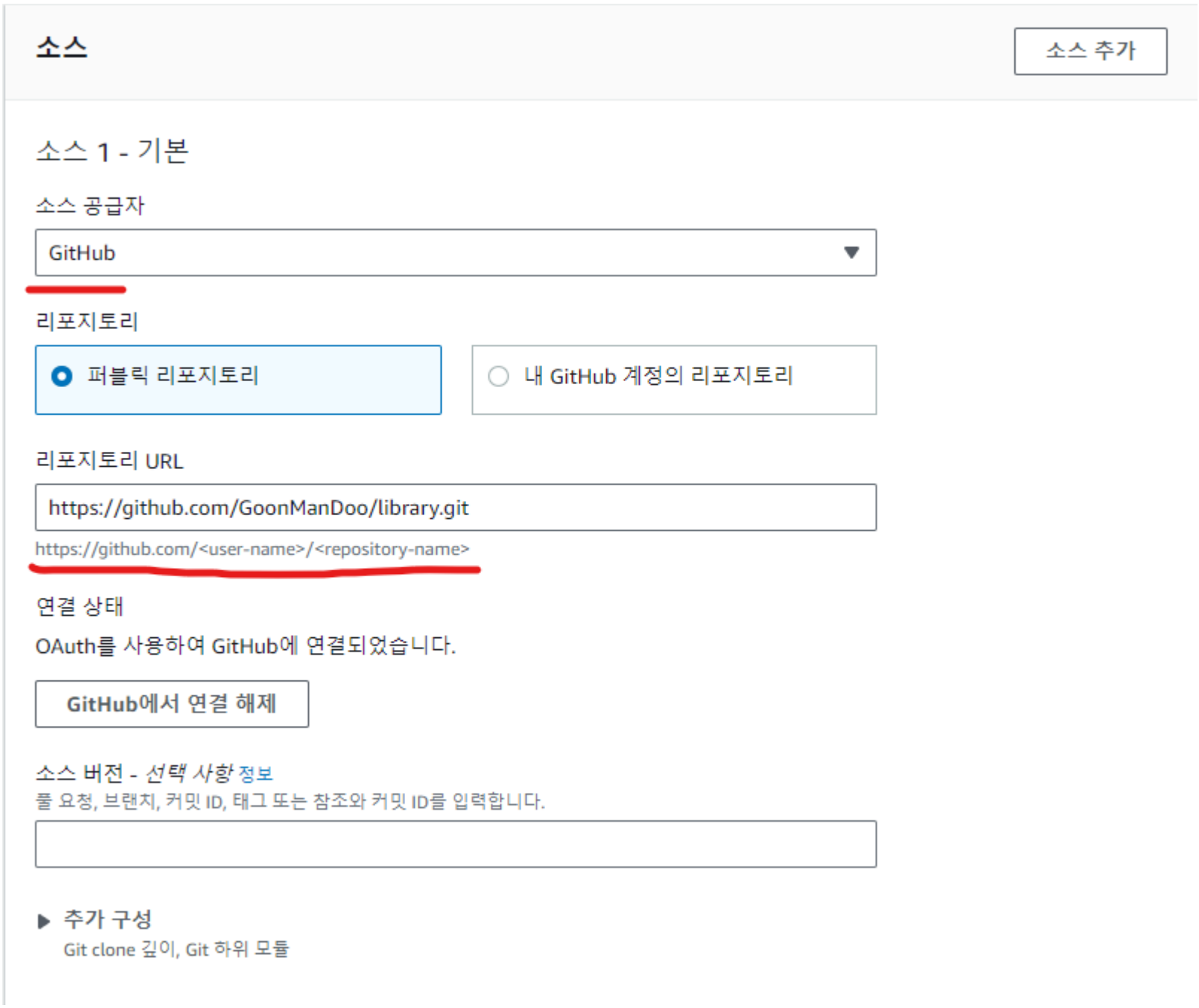
AWS 구성 내역, 내부 구축 내용

1. AWS CodeBuild

1. 소스



2. 환경

Java 17 버전이므로 호환되는 standard:5.0 설정

환경

현재 환경 이미지

aws/codebuild/amazonlinux2-aarch64-standard:2.0

새 환경 이미지

관리형 이미지
AWS CodeBuild에서 관리하는 이미지 사용

사용자 지정 이미지
도커 이미지 지정

운영 체제

Amazon Linux

런타임

Standard

이미지

aws/codebuild/amazonlinux2-x86_64-standard:5.0

이미지 버전

이 런타임 버전에 항상 최신 이미지 사용

Use GPU-enhanced compute

권한이 있음

도커 이미지를 빌드하거나 빌드의 권한을 승격하려면 이 플래그를 활성화합니다.

세션 연결

대화형 브라우저 기반 셸을 통해 빌드 머신의 컨테이너를 관리하도록 AWS 세션 관리자를 설정합니다. 서비스 역할에 세션 연결을 활성화하려면 AWS 세션 관리자 권한이 있어야 합니다. 자세히 알아보십시오

세션 연결 활성화

서비스 역할

계정에서 기존 서비스 역할 선택

arn:aws:iam::815602520517:role/service-role/codebuild-lala_project-service-r

3. BuildSpec

Buildspec

현재 buildspec

소스 코드 루트 디렉터리의 buildspec.yml 사용 중

빌드 사양

buildspec 파일 사용
YAML 형식의 buildspec 파일에 빌드 명령 저장

빌드 명령 삽입
빌드 명령을 빌드 프로젝트 구성으로 저장

Buildspec 이름 - *선택 사항*

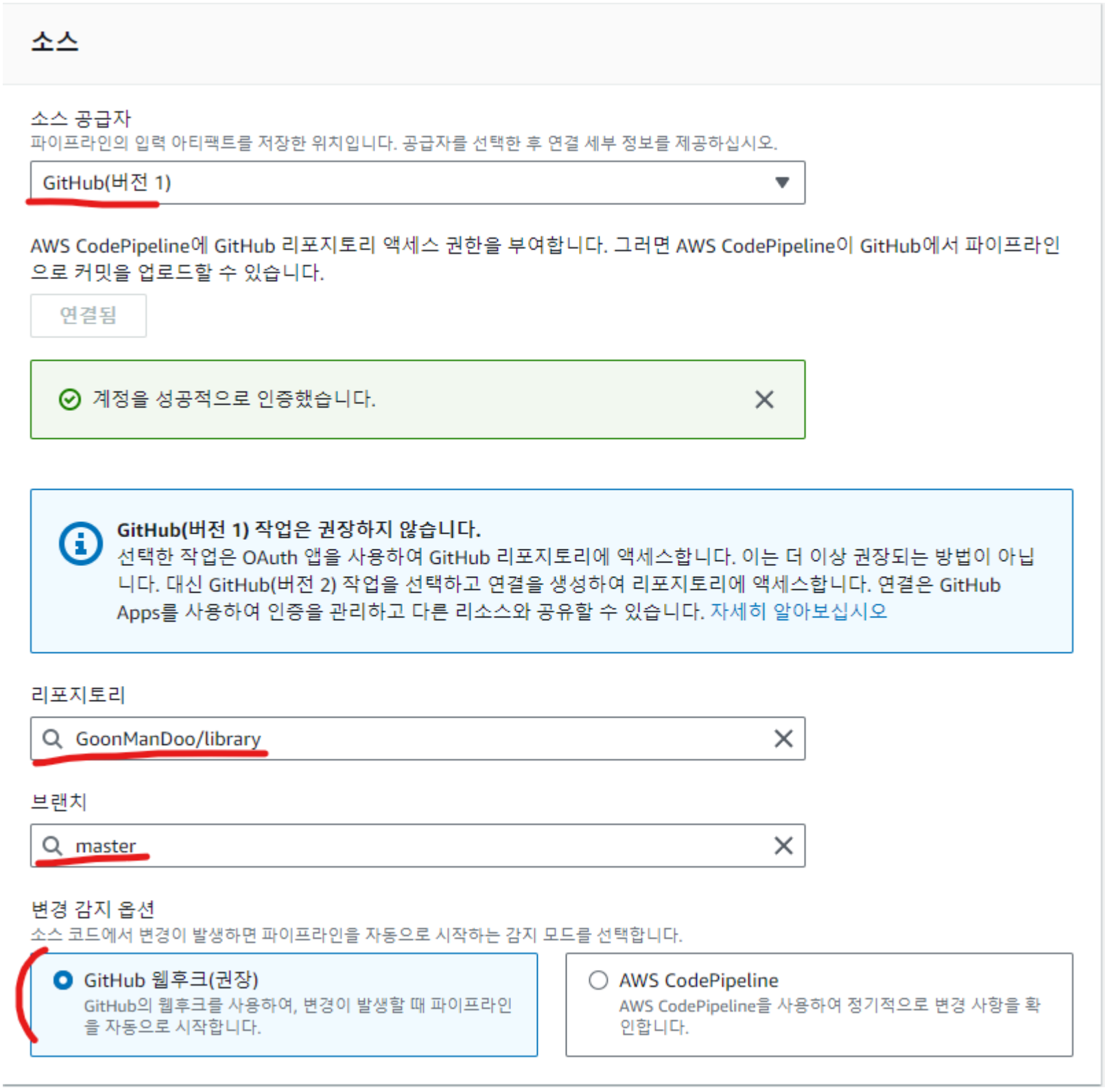
기본적으로 CodeBuild는 소스 코드 루트 디렉터리에서 buildspec.yml 파일을 찾습니다. buildspc 파일이 다른 이름 또는 위치를 사용하는 경우 여기에 소스 루트의 경로를 입력하십시오(예: buildspec-two.yml 또는 configuration/buildspec.yml).

```
# buildspec.yml
# AWS CodeBuild 빌드 버전
version: 0.2
```

```
phases:
  # 빌드 환경 설정
  install:
    # 사용할 Java 런타임 버전[17 지원이 안됨으로 커맨드로 java 17 설치]
    runtime-versions:
      java: corretto11
    commands:
      - export JAVA_17_HOME="/usr/lib/jvm/java-17-amazon-corretto.x86_64"
      - export JDK_17_HOME="/usr/lib/jvm/java-17-amazon-corretto.x86_64"
      - export JRE_17_HOME="/usr/lib/jvm/java-17-amazon-corretto.x86_64"
      - export JAVA_HOME="$JAVA_17_HOME"
      - export JRE_HOME="$JRE_17_HOME"
      - export JDK_HOME="$JDK_17_HOME"
      - export JAVA_HOME=$CODEBUILD_AGENT_JAVA_HOME
      - echo "JAVA_HOME set to $JAVA_HOME"
      - |-
        export GNUPGHOME="$(mktemp -d)" \
            && curl -fL -o corretto.key https://yum.corretto.aws/corretto.key \
            && gpg --batch --import corretto.key \
            && gpg --batch --export --armor '6DC3636DAE534049C8B94623A122542AB04F24E3' > corretto.key \
            && rpm --import corretto.key \
            && rm -r "$GNUPGHOME" corretto.key \
            && curl -fL -o /etc/yum.repos.d/corretto.repo https://yum.corretto.aws/corretto.repo \
            && grep -q '^gpgcheck=1' /etc/yum.repos.d/corretto.repo \
            && yum install -y java-17-amazon-corretto-devel \
            && (find /usr/lib/jvm/java-17-amazon-corretto.x86_64 -name src.zip -delete || true) \
            && yum install -y fontconfig
      - |-
#       java 실행 파일에 대한 심볼릭 설정
        for tool_path in "$JAVA_HOME"/bin/*;
         do tool=$(basename "$tool_path");
          if [ $tool != 'java-rmi.cgi' ];
          then
           rm -f /usr/bin/$tool /var/lib/alternatives/$tool \
            && update-alternatives --install /usr/bin/$tool $tool $tool_path 20000;
          fi;
        done
# 빌드 전 실행할 명령어
  pre_build:
    commands:
# gradlew 실행 권한 부여
      - chmod +x gradlew
  build:
    commands:
# gradlew을 사용하여 프로젝트 빌드
      - ./gradlew build
# 빌드 후 생성된 파일 처리
artifacts:
# 빌드된 파일의 경로를 지정
  files: '**/*'
# 파일의 기본 디렉토리 지정
  base-directory: 'build/libs'
  discard-paths: yes
```

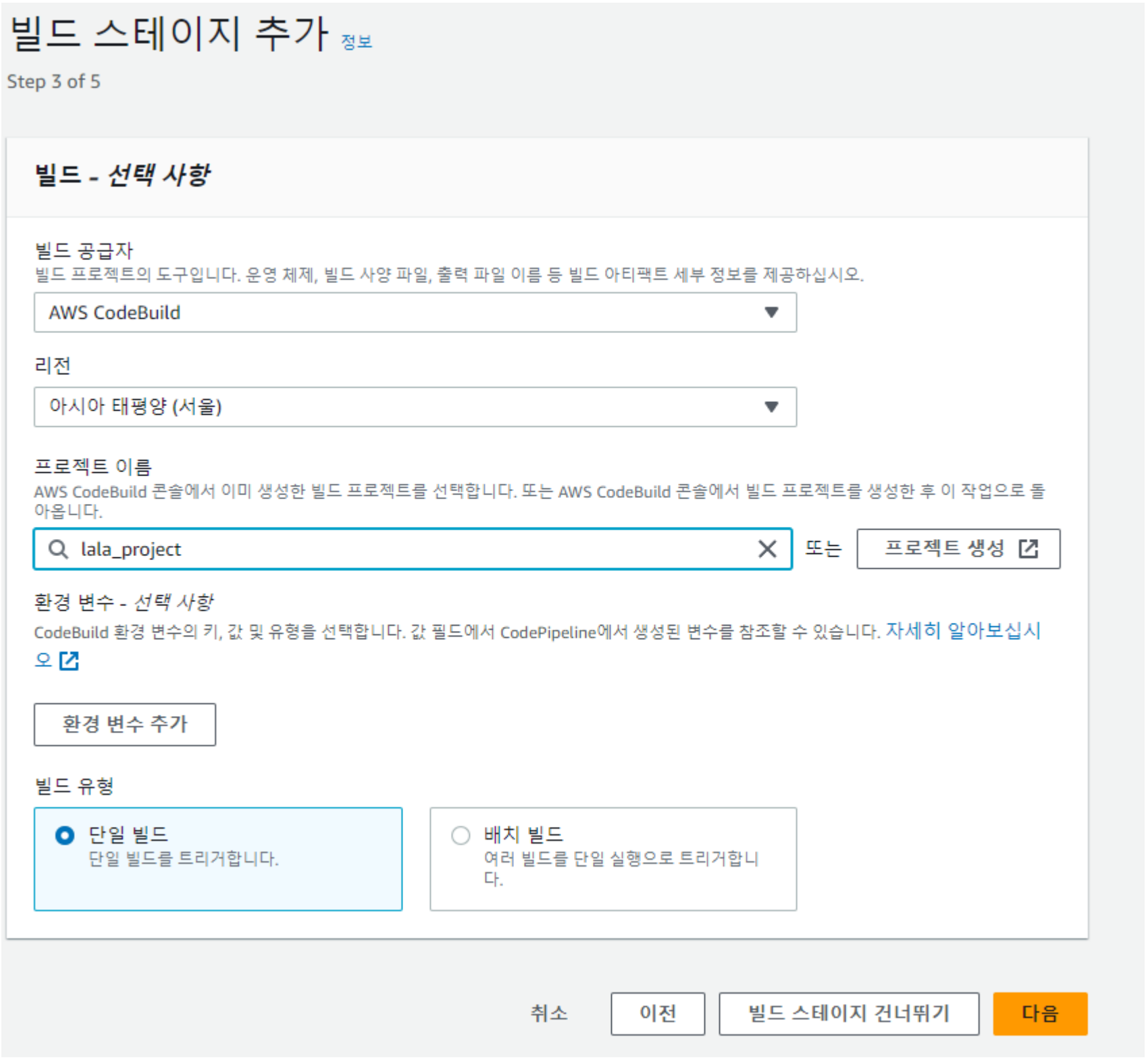
2. AWS CodePipeline

1. 소스

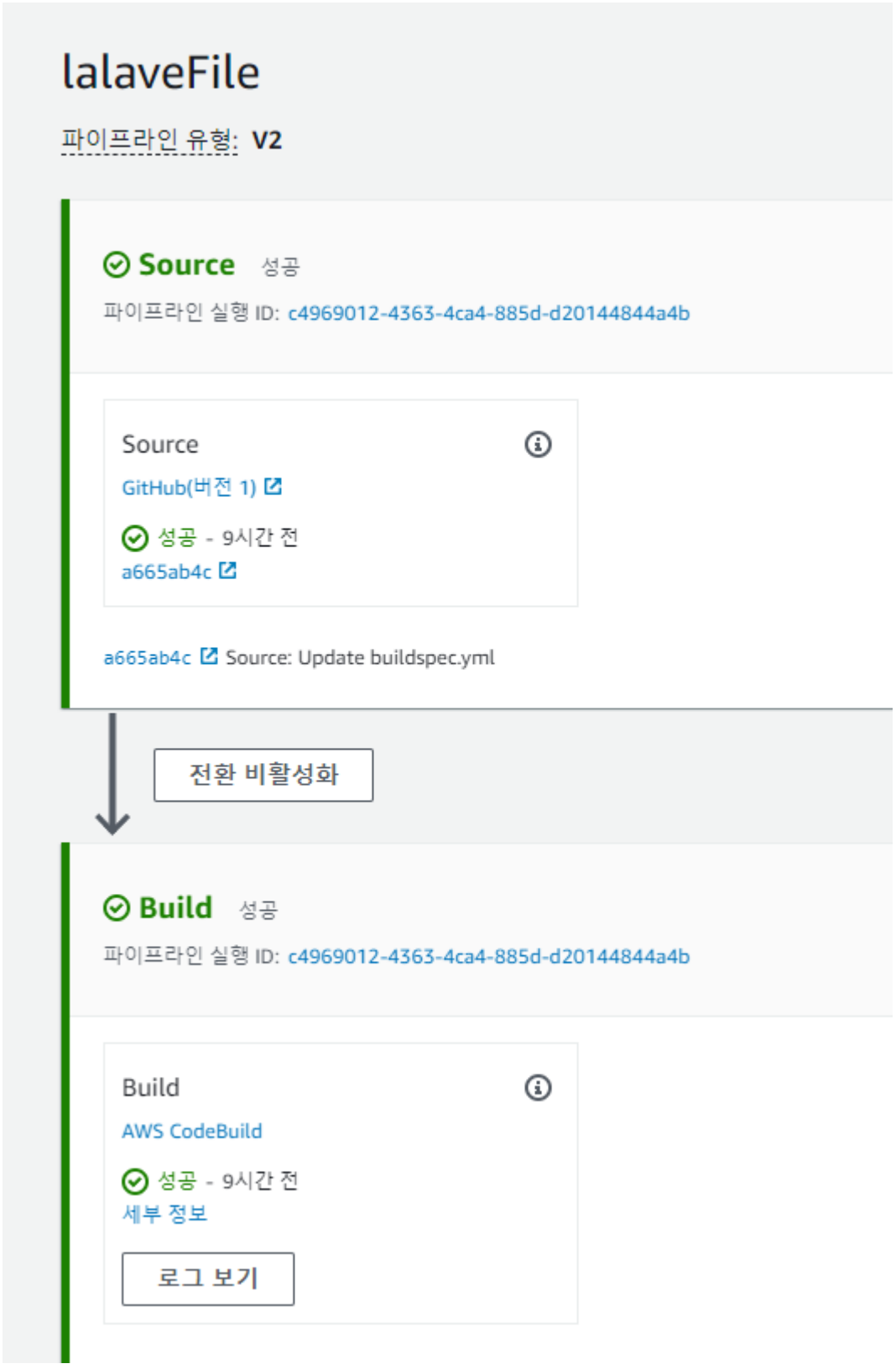


2. 빌드 스테이지 추가

위에서 생성한 AWS CodeBuild 선택



3. CodePipeline 실행


lalaveFile
파이프라인 유형: V2
Source 성공
파이프라인 실행 ID: c4969012-4363-4ca4-885d-d20144844a4b
Source
GitHub(버전 1)
성공 - 9시간 전
a665ab4c
a665ab4c Source: Update buildspec.yml
전환 비활성화
Build 성공
파이프라인 실행 ID: c4969012-4363-4ca4-885d-d20144844a4b
Build
AWS CodeBuild
성공 - 9시간 전
세부 정보
로그 보기