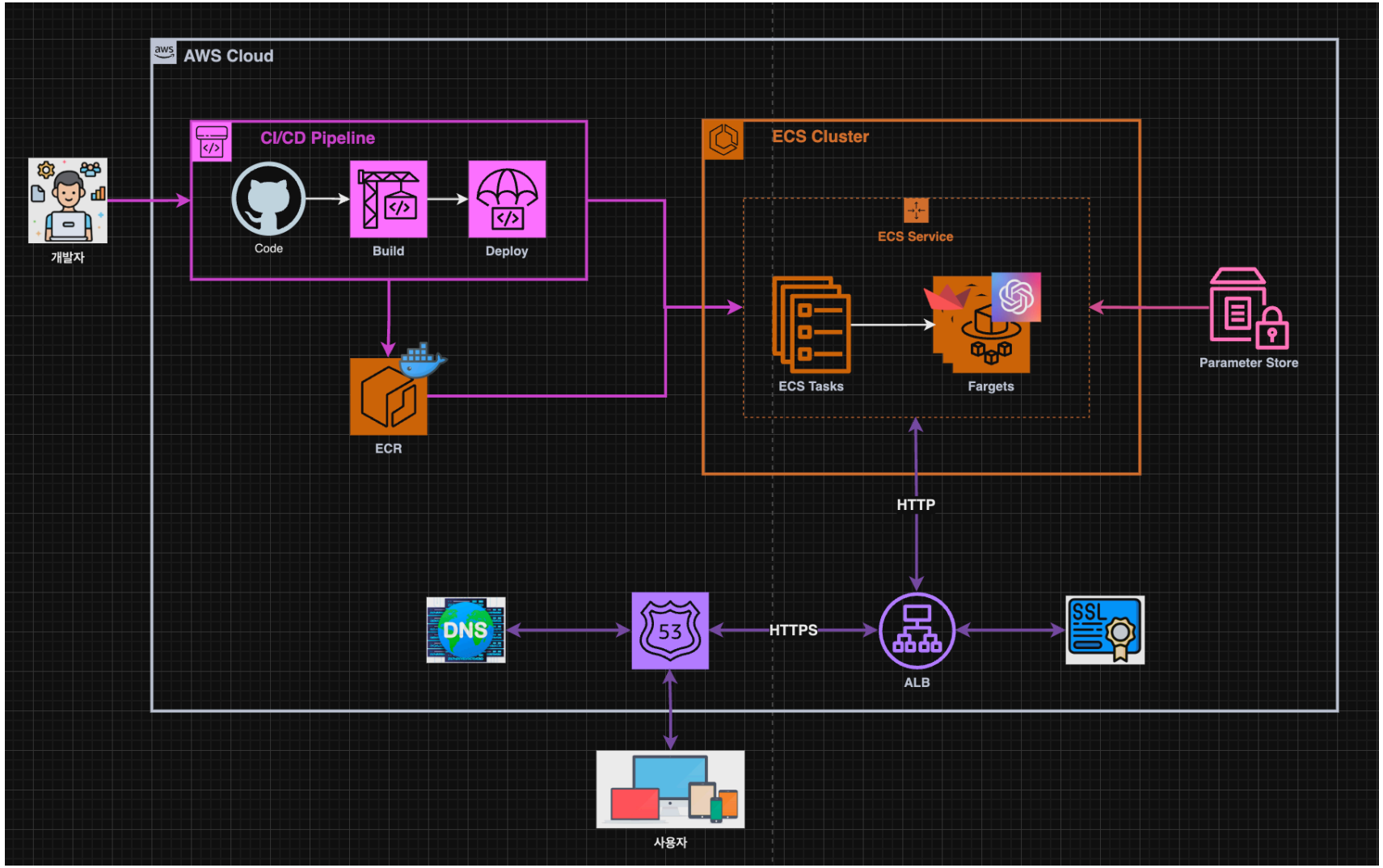
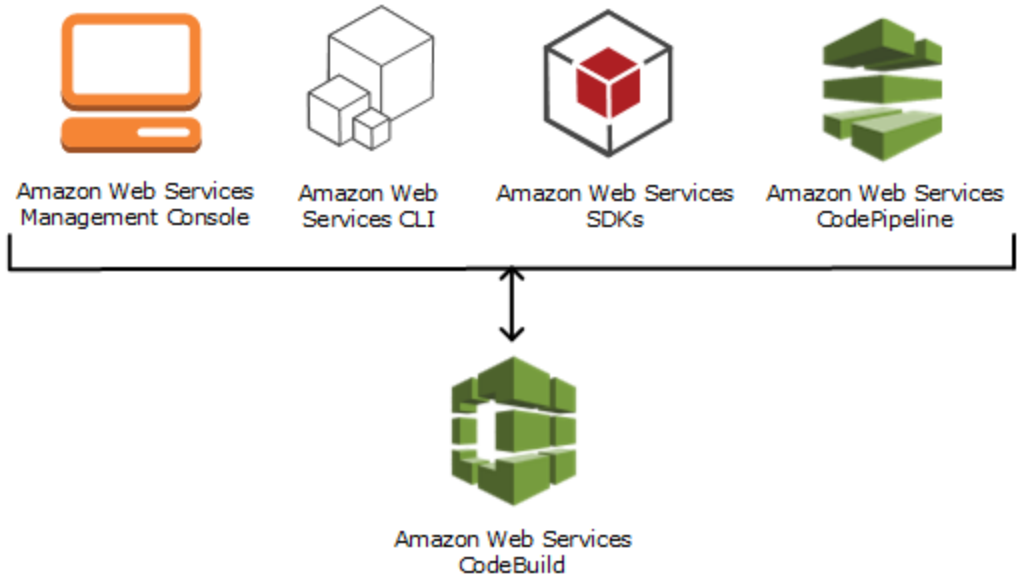
Architecture

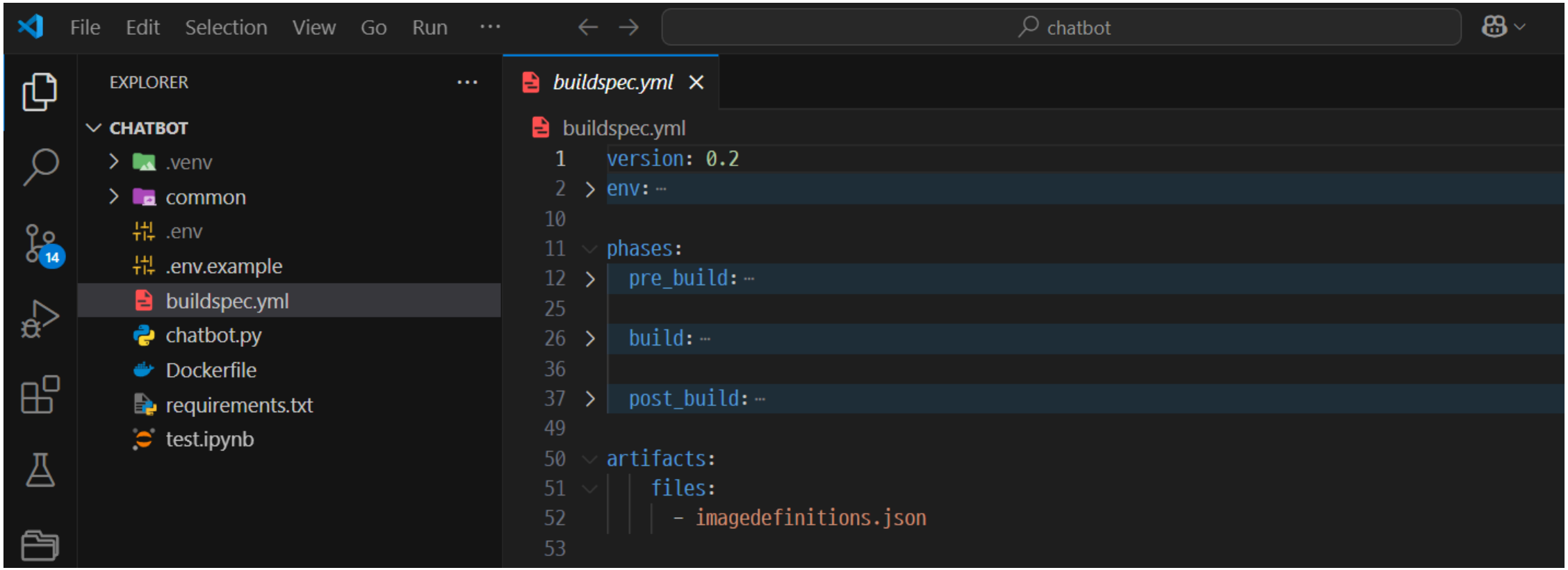


AWS CodeBuild

- AWS CodeBuild 는 클라우드의 완전 관리형 빌드 서비스입니다. 는 소스 코드를 CodeBuild 컴파일하고, 단위 테스트를 실행하고, 배포할 준비가 된 아티팩트를 생성합니다.



Create buildspec.yml



단계1: env

- build 중 사용할 변수 선언

```
version: 0.2
env:
  git-credential-helper: 'yes'
  variables:
    # docker 파라미터 정의
    ECS_CONTAINER_NAME: ecs-streamlit-container
    IMAGE_REPO_NAME: ecs-streamlit-ecr # aws ecr에 등록이 되어 있는 name
    IMAGE_TAG: latest
    AWS_DEFAULT_REGION: ap-northeast-2
```

단계2: pre_build

- build 전 사전 작업

```
phases:
  pre_build:
    commands:
      # Docker push를 할 ECR에 로그인 필요
      # https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecr/get-login-password.html
      # https://stackoverflow.com/questions/77488134/codebuild-error-exit-status-127-file-name-too-long
      - echo Logging in to Amazon ECR...
      - aws --version
      - ECR_PASSWORD=$(aws ecr get-login-password --region $AWS_DEFAULT_REGION)
      - AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output
        text)
      - echo -n ${ECR_PASSWORD} | docker login --username AWS --password-stdin
        ${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_DEFAULT_REGION}.amazonaws.com
      - REPOSITORY_URI=$AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG
```

단계3: build

- build 작업

```
build:
  commands:
    - echo Build started on `date`
    # 프로젝트 폴더로 이동
    - cd ./4-2_AWS_CICD_Example/chatbot
    - pwd
    - echo Building the Docker image...
    # Github Repository에 있는 Dockerfile의 위치에 맞춰 수정
    - docker build -f ./Dockerfile -t $IMAGE_REPO_NAME:$IMAGE_TAG .
    - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $REPOSITORY_URI
```

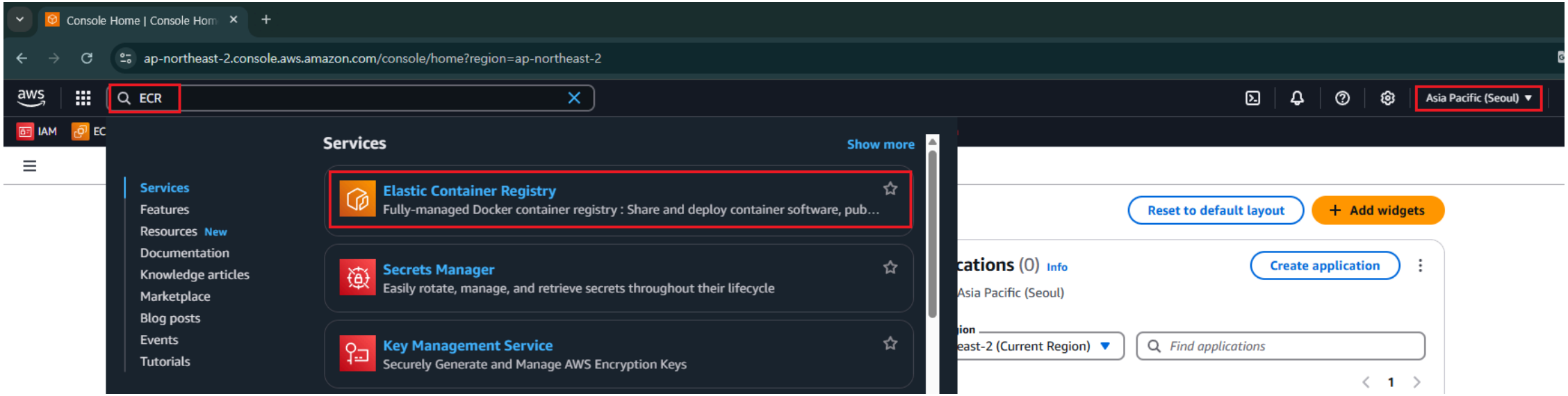
단계2: post_build & artifacts

- build 후 작업

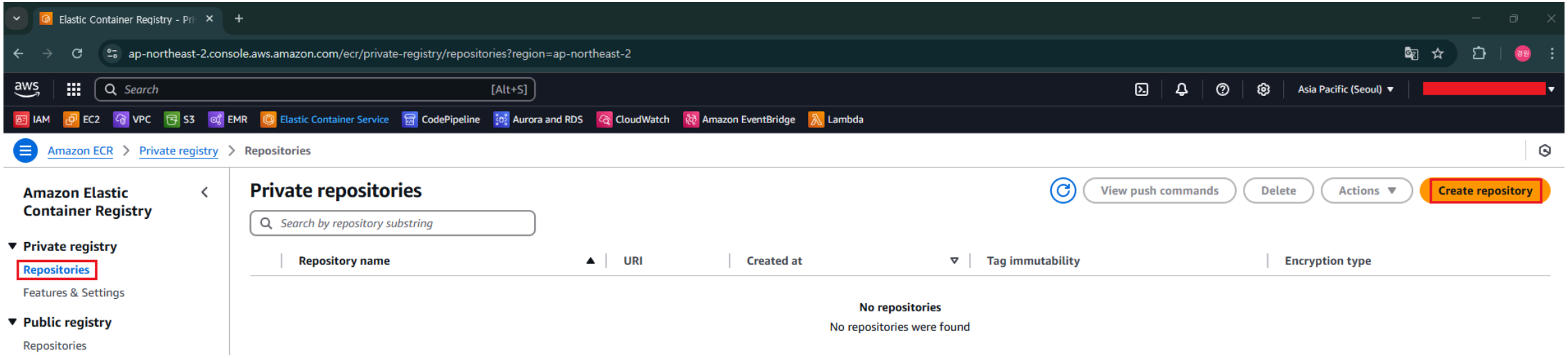
```
  post_build:
    commands:
      - echo package Source...
      - echo push the Docker image...
      - docker push $REPOSITORY_URI

      # Give your container name
      - printf '[{"name":"%s","imageUri":"%s"}]' $ECS_CONTAINER_NAME
        $REPOSITORY_URI > ../../imagedefinitions.json
      - echo $ECS_CONTAINER_NAME
      - echo printing ../../imagedefinitions.json
      - cat ../../imagedefinitions.json

artifacts:
    files:
      - imagedefinitions.json
```

Create AWS ECR

단계1: AWS ECR 접속

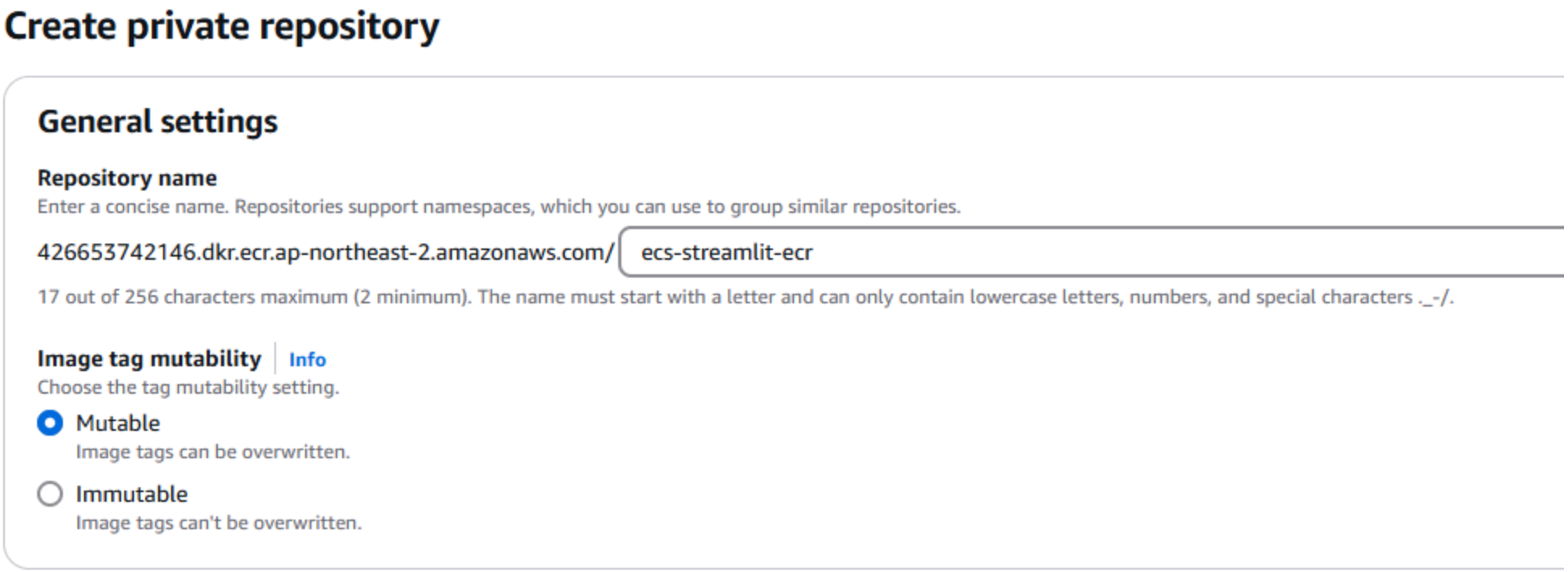


단계2: Create repository

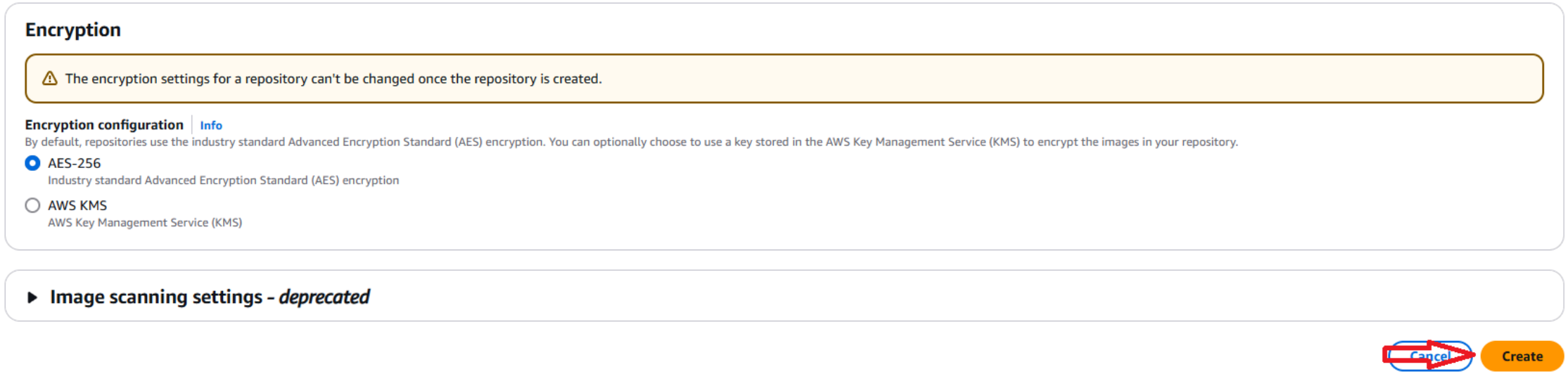


단계3: Repository name

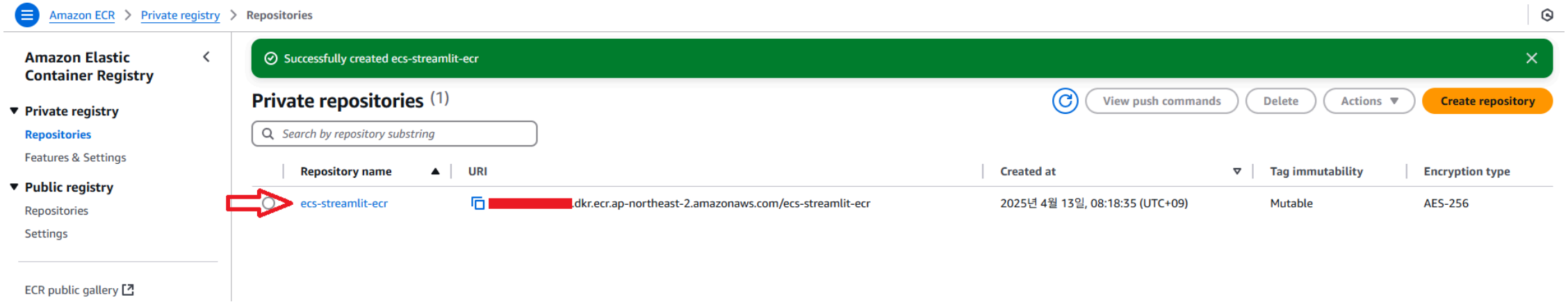
- buildspec.yml에 정의한 이름으로 설정!



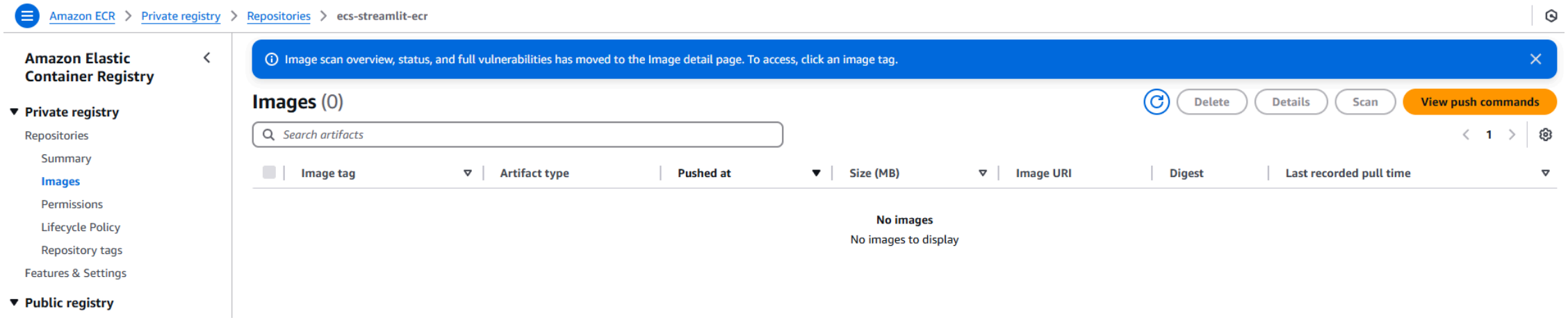
단계4: Create


Encryption
The encryption settings for a repository can't be changed once the repository is created.
Encryption configuration
Info
By default, repositories use the industry standard Advanced Encryption Standard (AES) encryption. You can optionally choose to use a key stored in the AWS Key Management Service (KMS) to encrypt the images in your repository.
AES-256
Industry standard Advanced Encryption Standard (AES) encryption
AWS KMS
AWS Key Management Service (KMS)
Image scanning settings - deprecated
Cancel
Create

단계5: 확인

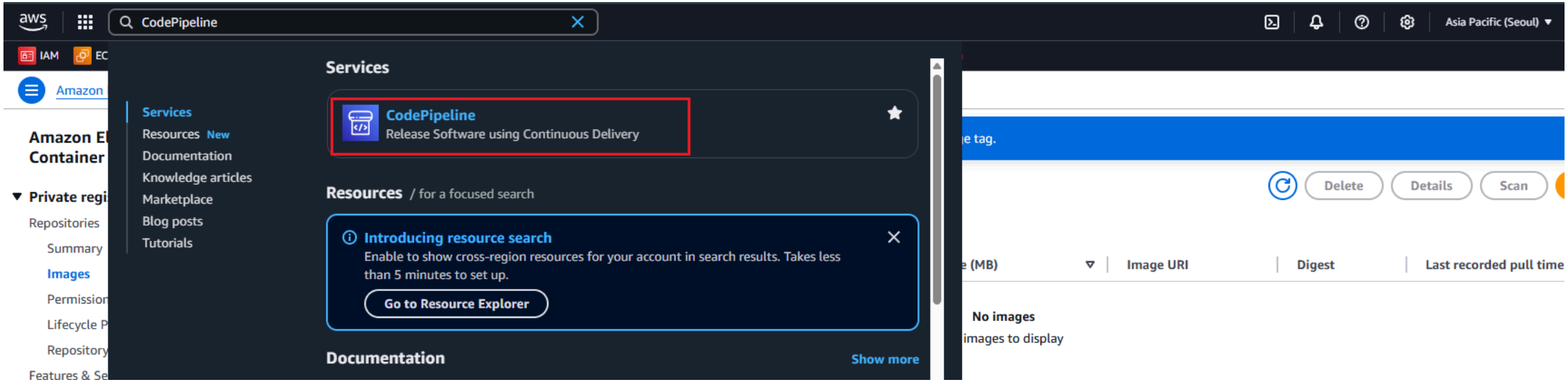

Amazon ECR > Private registry > Repositories
Amazon Elastic Container Registry
Private registry
Repositories
Features & Settings
Public registry
Repositories
Settings
ECR public gallery
Successfully created ecs-streamlit-ecr
Private repositories (1)
View push commands
Delete
Actions
Create repository
Search by repository substring
Repository name
URI
Created at
Tag immutability
Encryption type
ecs-streamlit-ecr
[redacted].dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr
2025년 4월 13일, 08:18:35 (UTC+09)
Mutable
AES-256

- 만들어진 ECR에는 등록된 Image가 없음

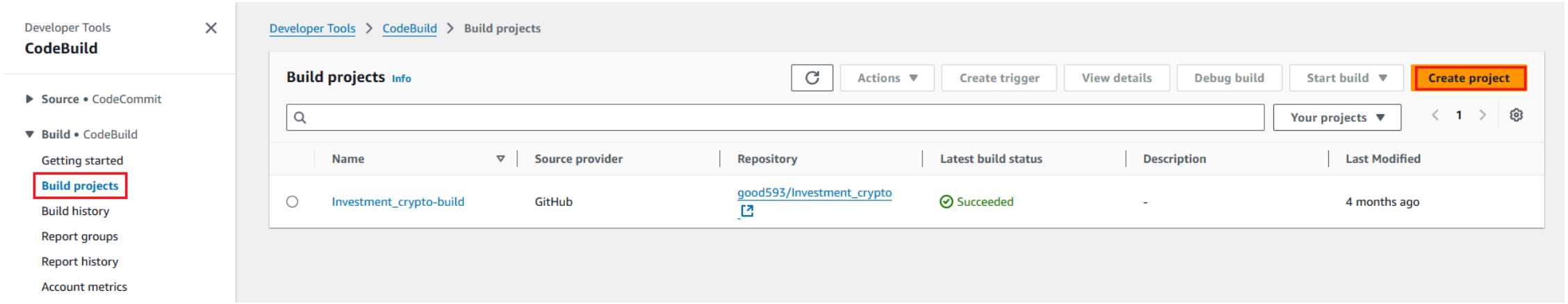


Create CodeBuild

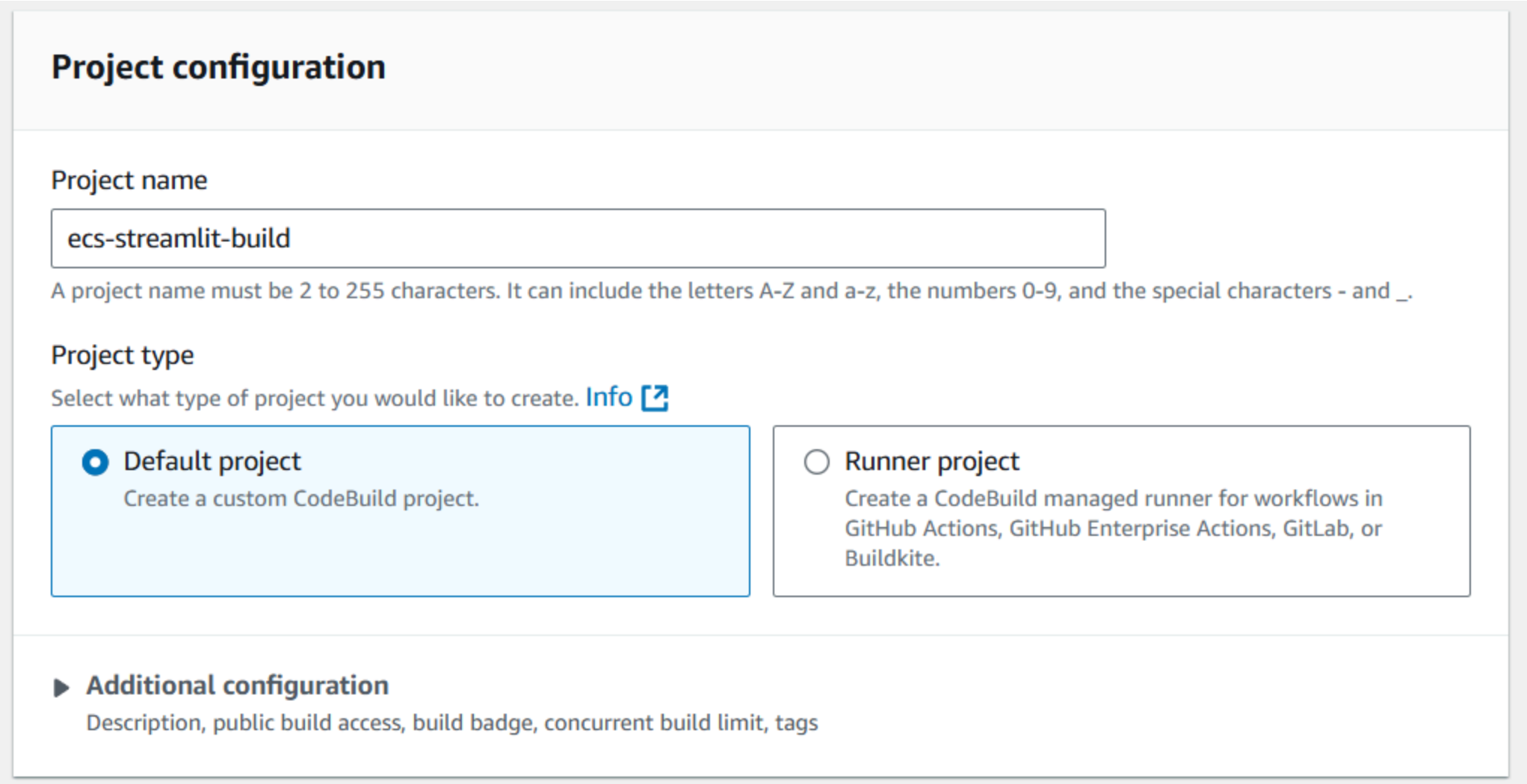
단계1: CodePipeline 접속



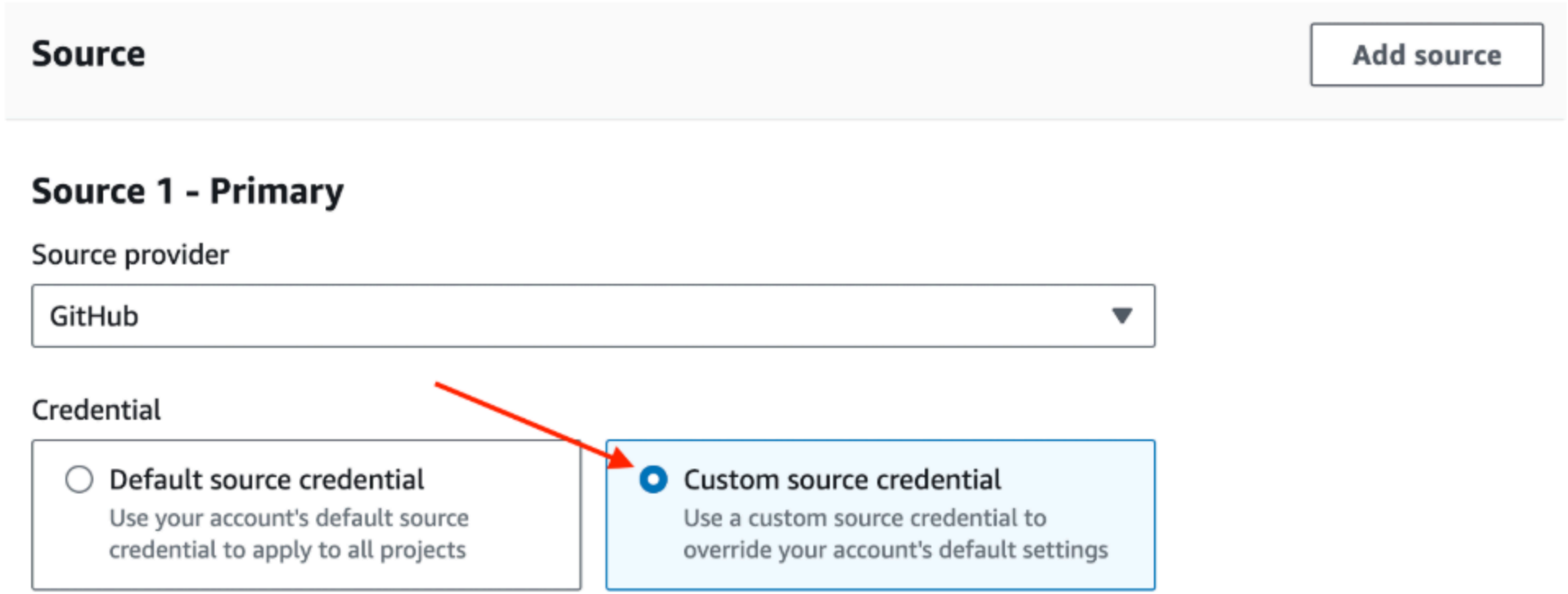
단계2: Create project

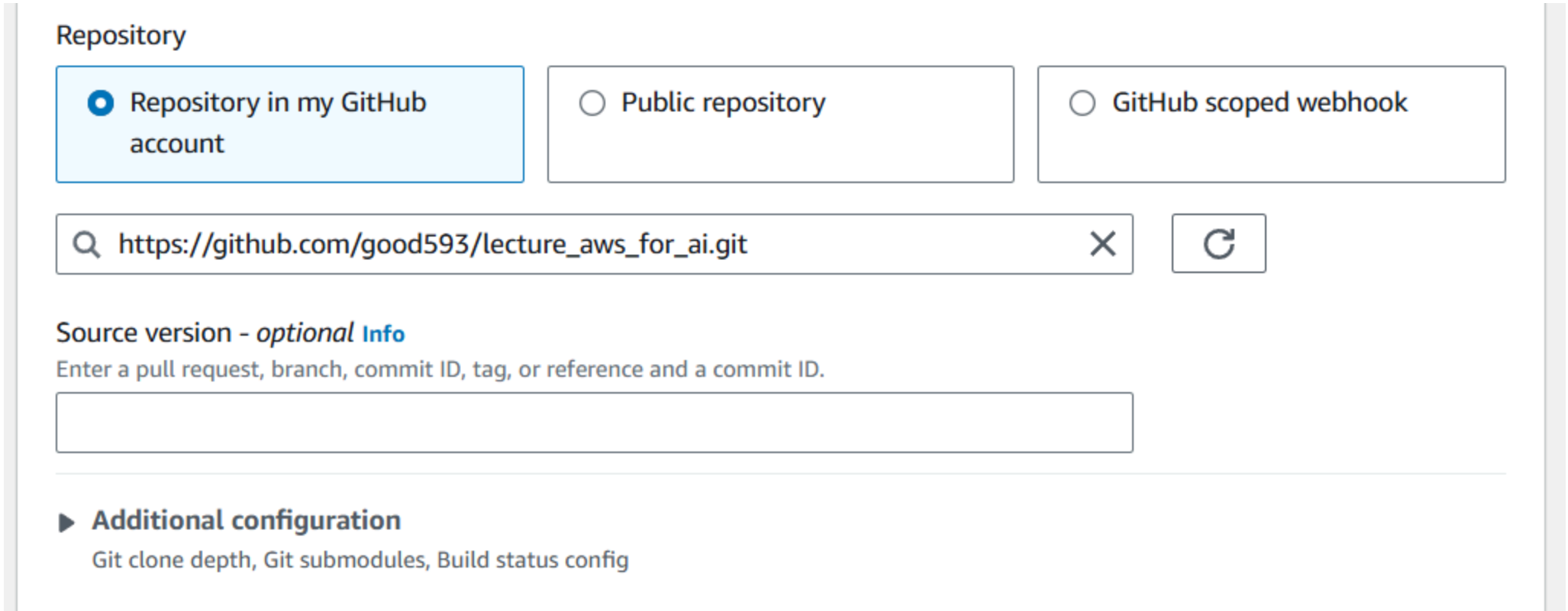


단계3: Project configuration

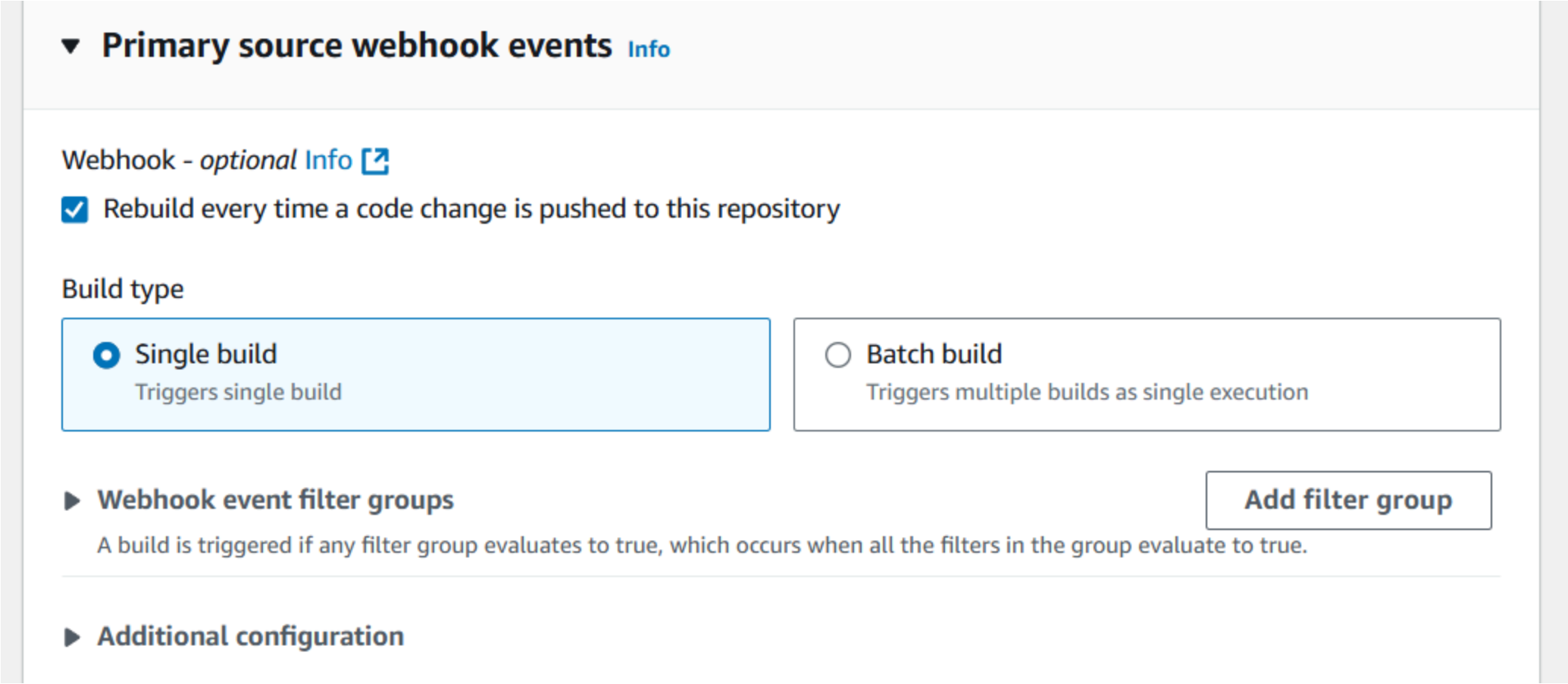


단계4: Source

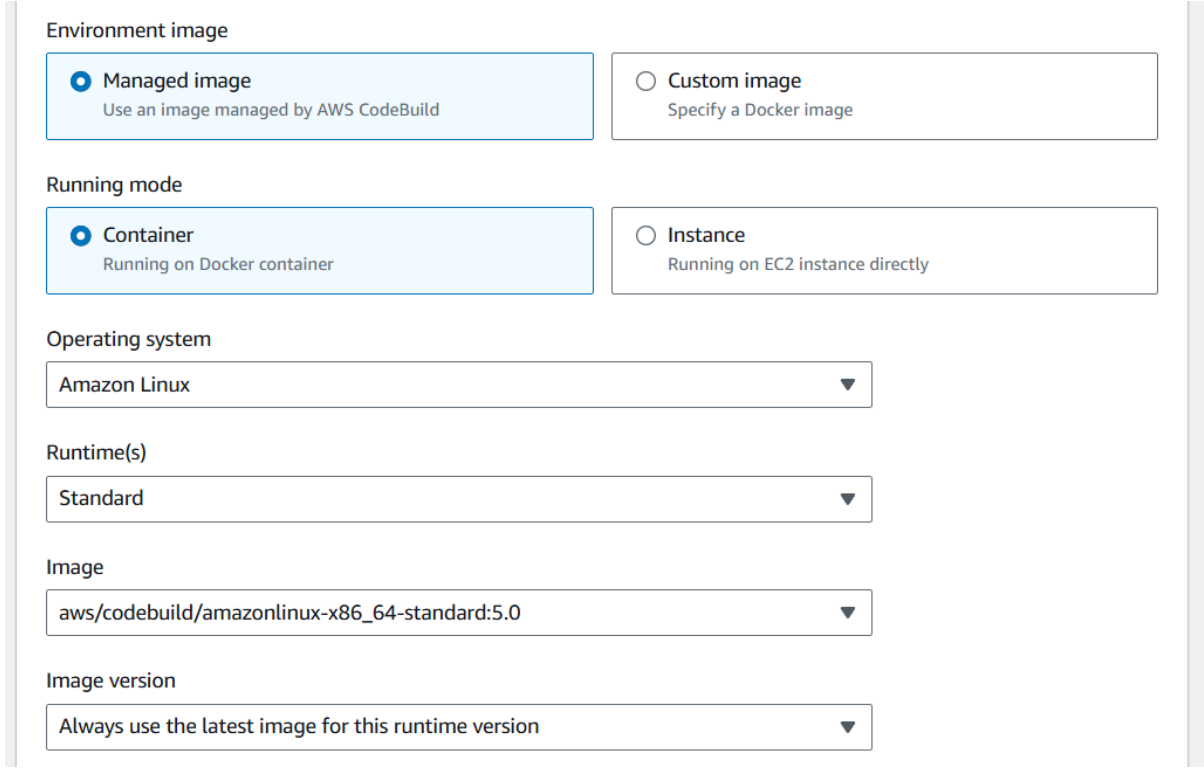

Source
Add source
Source 1 - Primary
Source provider
GitHub
Credential
Default source credential
Use your account's default source credential to apply to all projects
Custom source credential
Use a custom source credential to override your account's default settings


Repository
Repository in my GitHub account
Public repository
GitHub scoped webhook
https://github.com/good593/lecture_aws_for_ai.git
Source version - optional Info
Enter a pull request, branch, commit ID, tag, or reference and a commit ID.
Additional configuration
Git clone depth, Git submodules, Build status config

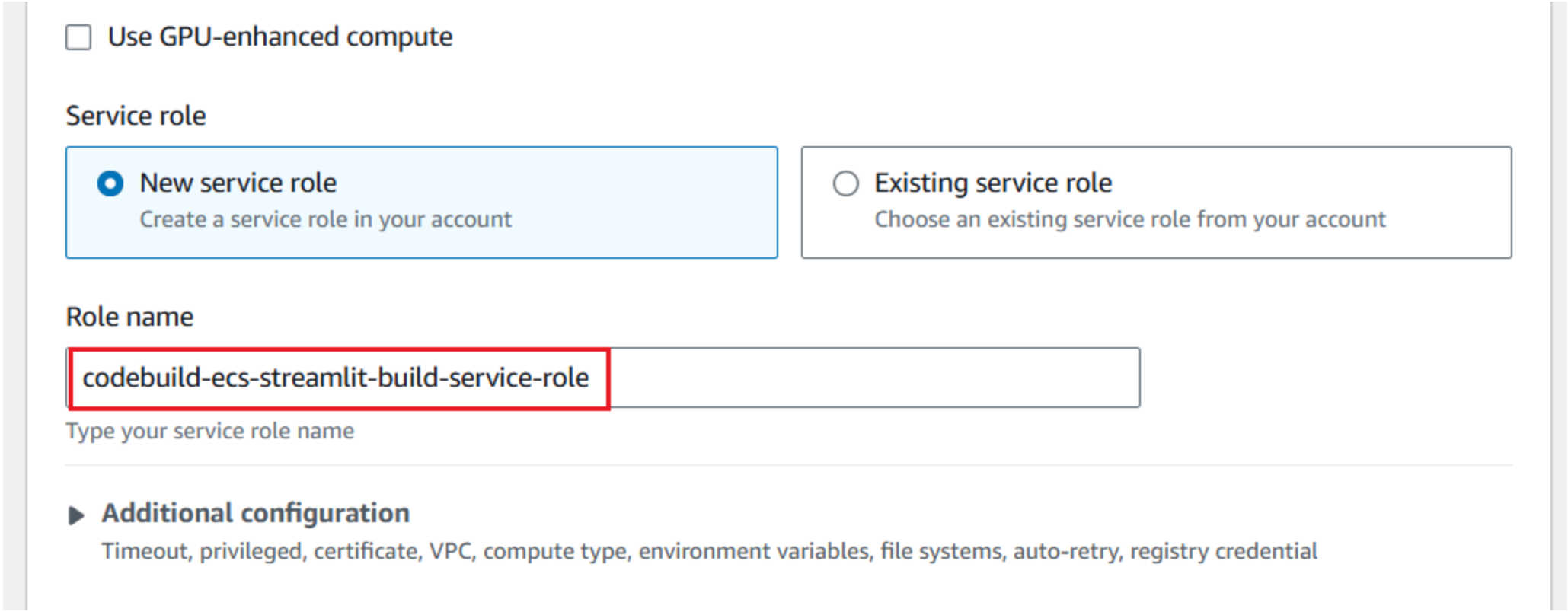
단계5: Primary source webhook events



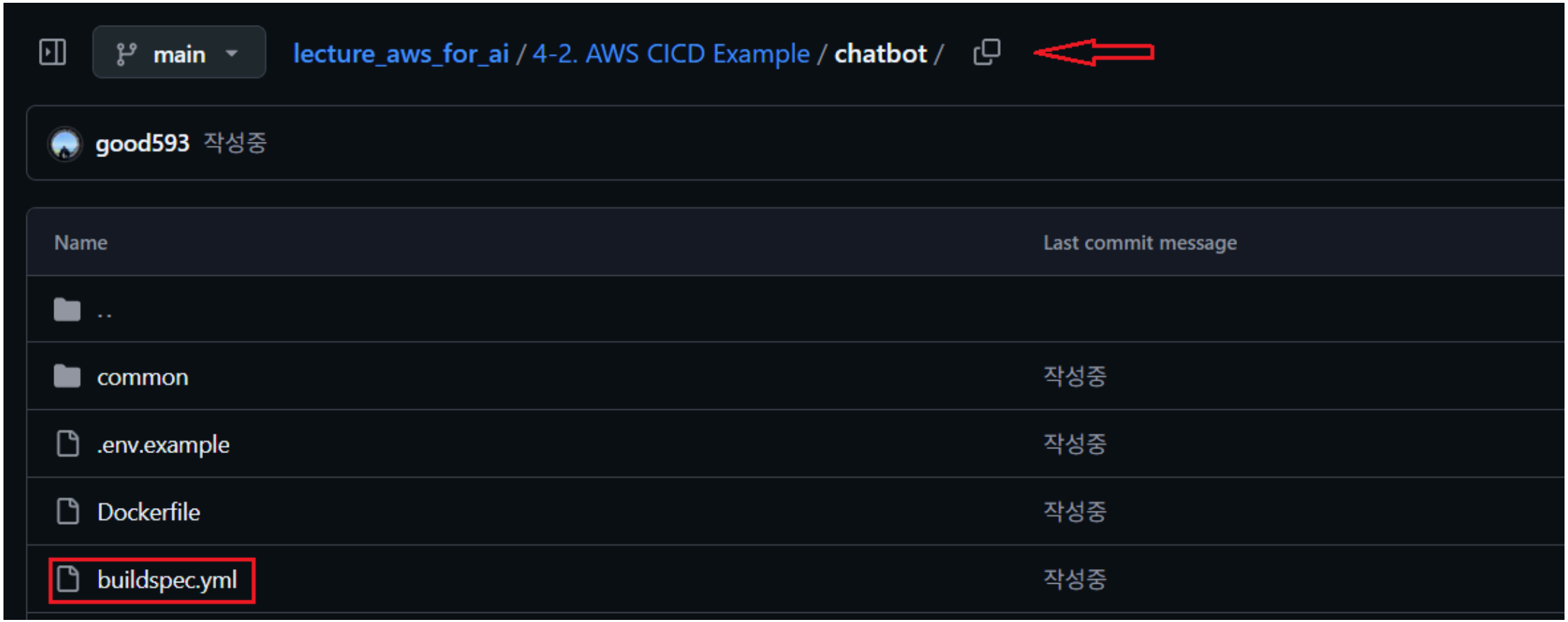
단계6: Environment



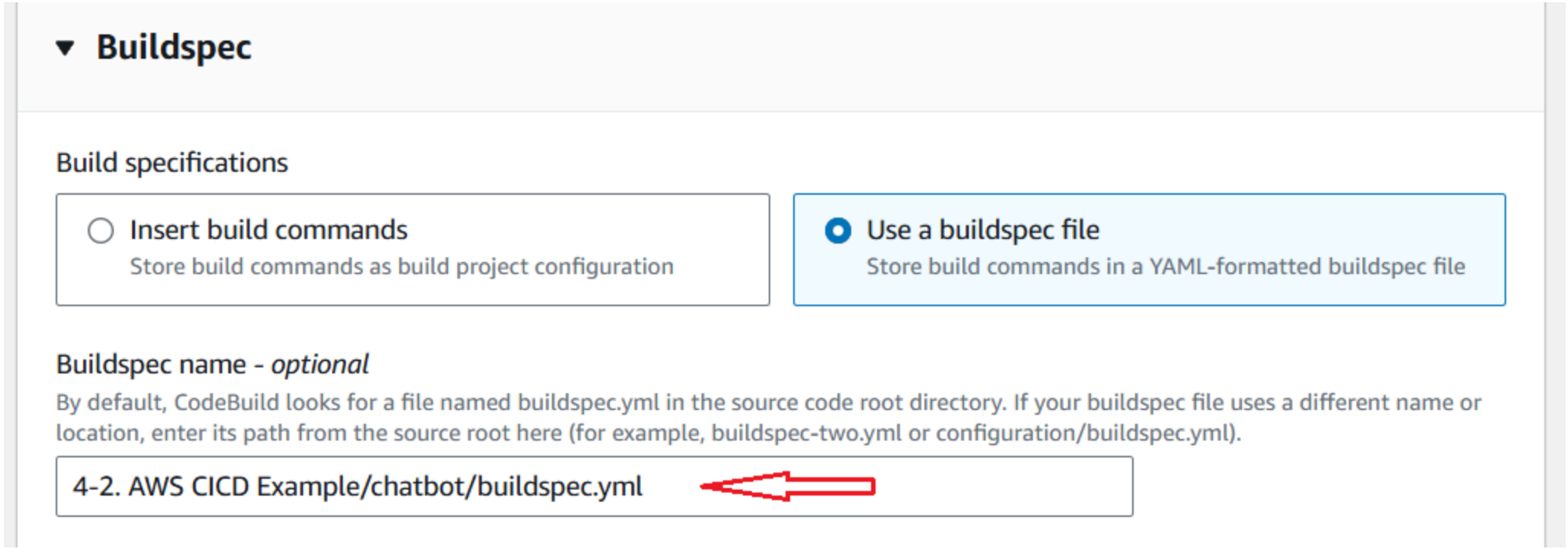
- 생성된 CodeBuild Role Name 확인



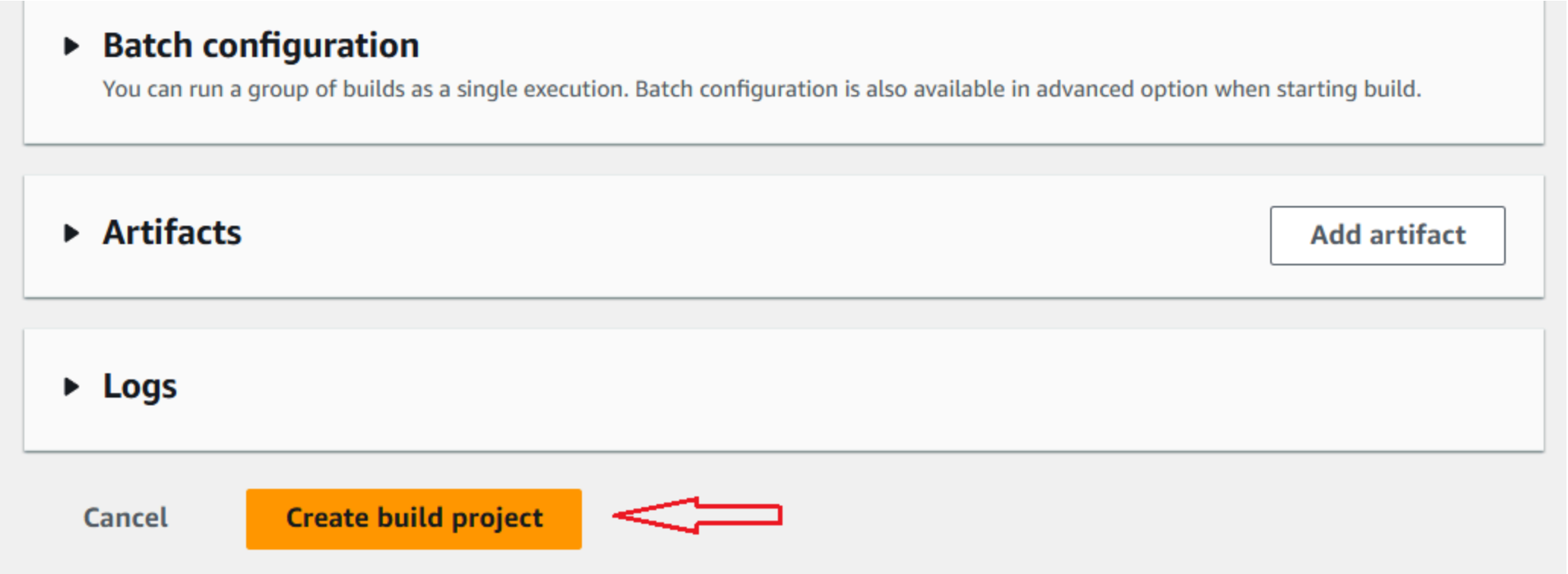
단계7: Github에서 buildspec.yml 위치 확인



- Buildspec name에 해당 위치 정보 작성

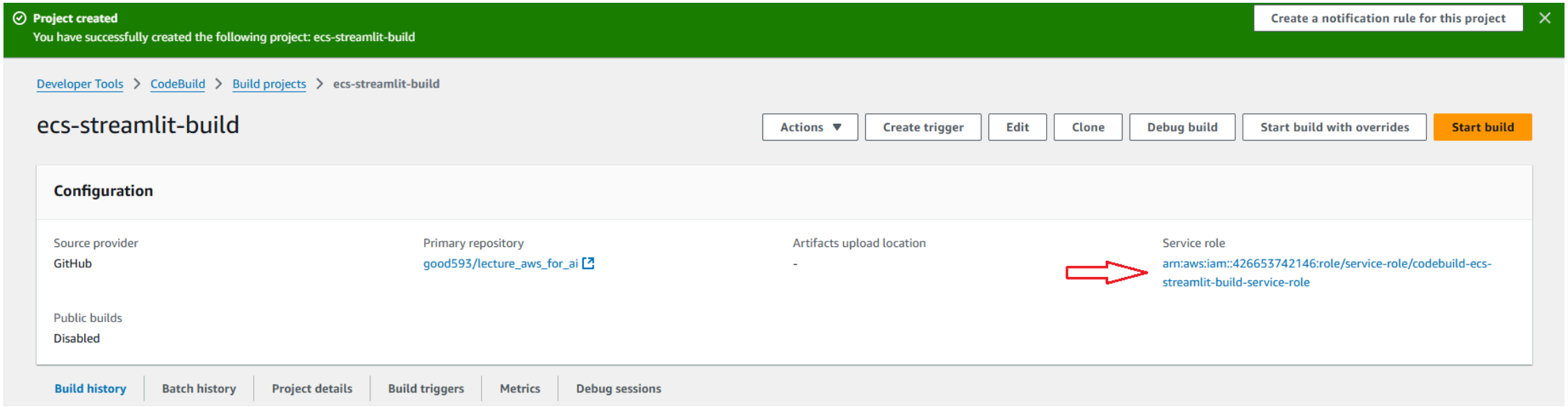


단계7: Create build project

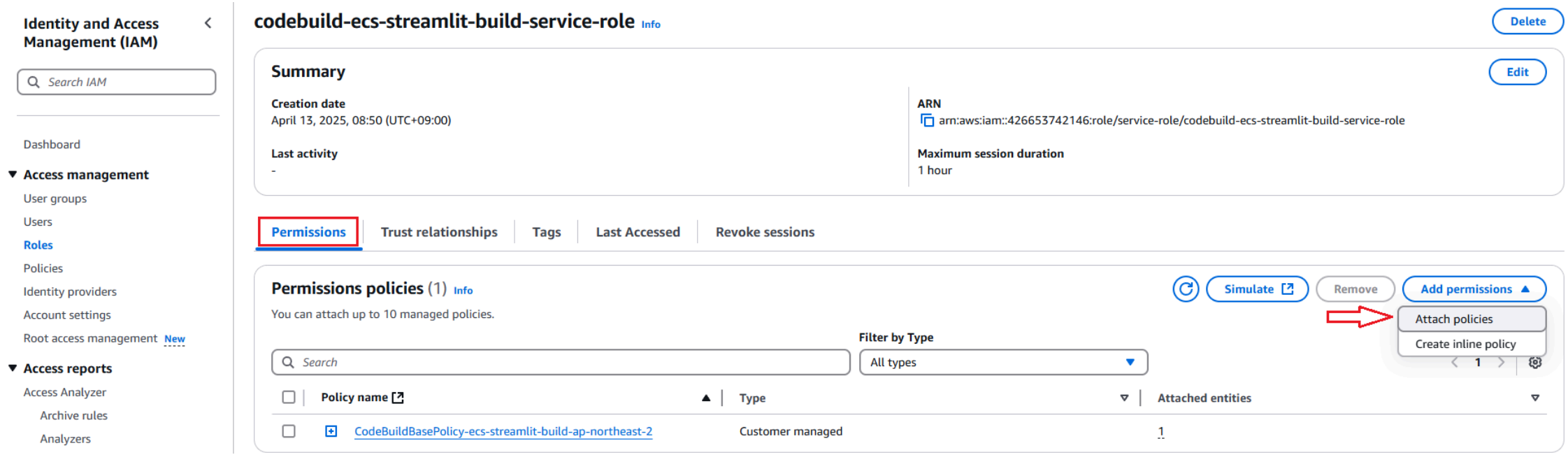


Add Permission

단계1: IAM 접속

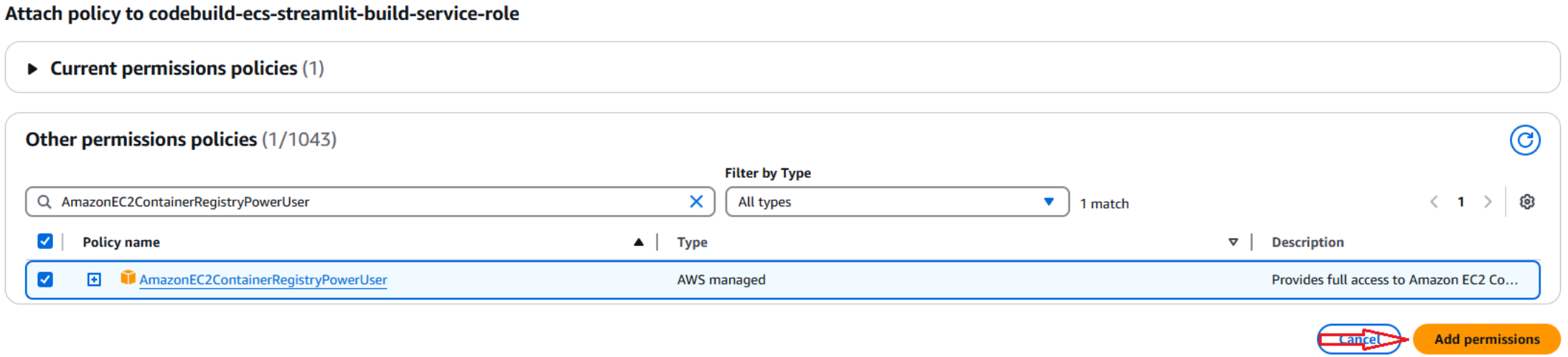

Project created
You have successfully created the following project: ecs-streamlit-build
Create a notification rule for this project
Developer Tools > CodeBuild > Build projects > ecs-streamlit-build
ecs-streamlit-build
Actions
Create trigger
Edit
Clone
Debug build
Start build with overrides
Start build
Configuration
Source provider
GitHub
Primary repository
good593/lecture_aws_for_ai
Artifacts upload location
-
Service role
arn:aws:iam::426653742146:role/service-role/codebuild-ecs-streamlit-build-service-role
Public builds
Disabled
Build history
Batch history
Project details
Build triggers
Metrics
Debug sessions

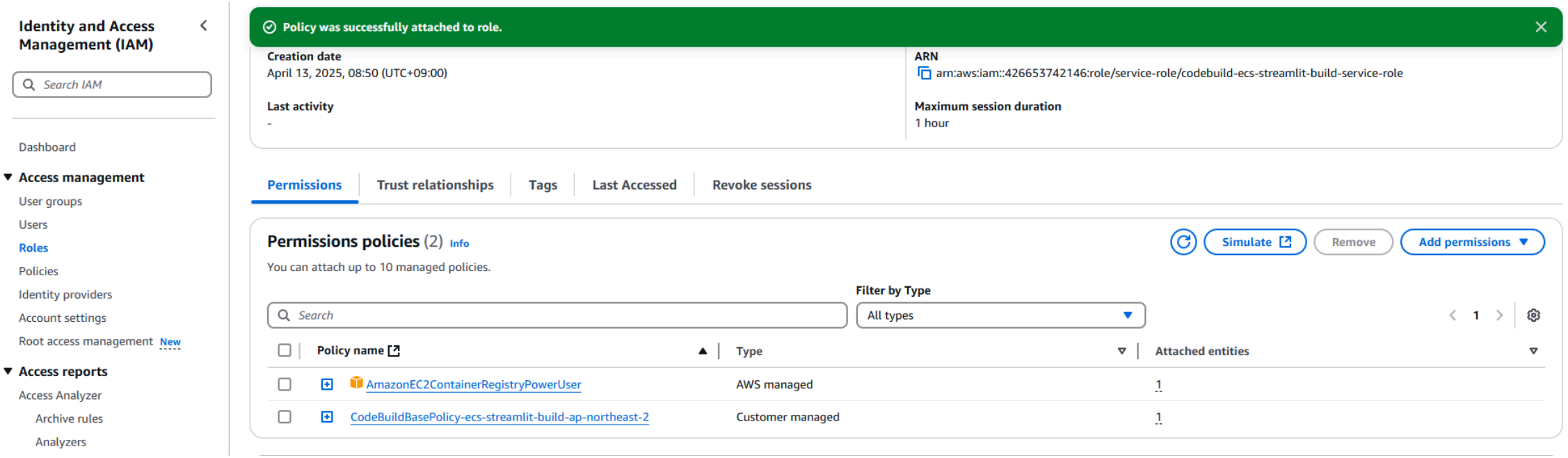
단계2: Attach policies



단계3: Add AmazonEC2ContainerRegistryPowerUser

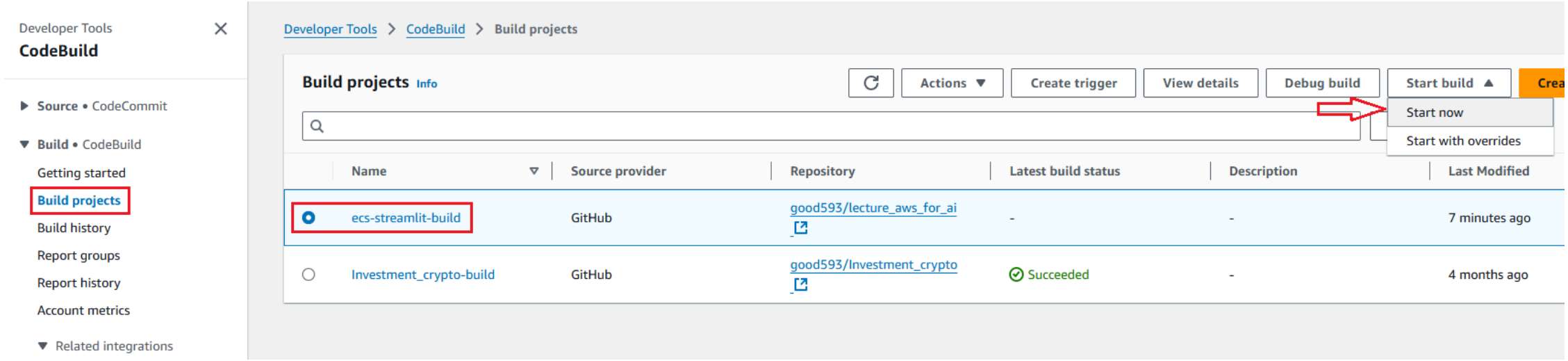
> `AmazonEC2ContainerRegistryPowerUser`
>
> - Amazon EC2 컨테이너 레지스트리 리포지토리에 대한 전체 액세스를 제공하지만 리포지토리 삭제 또는 정책 변경은 허용하지 않습니다.



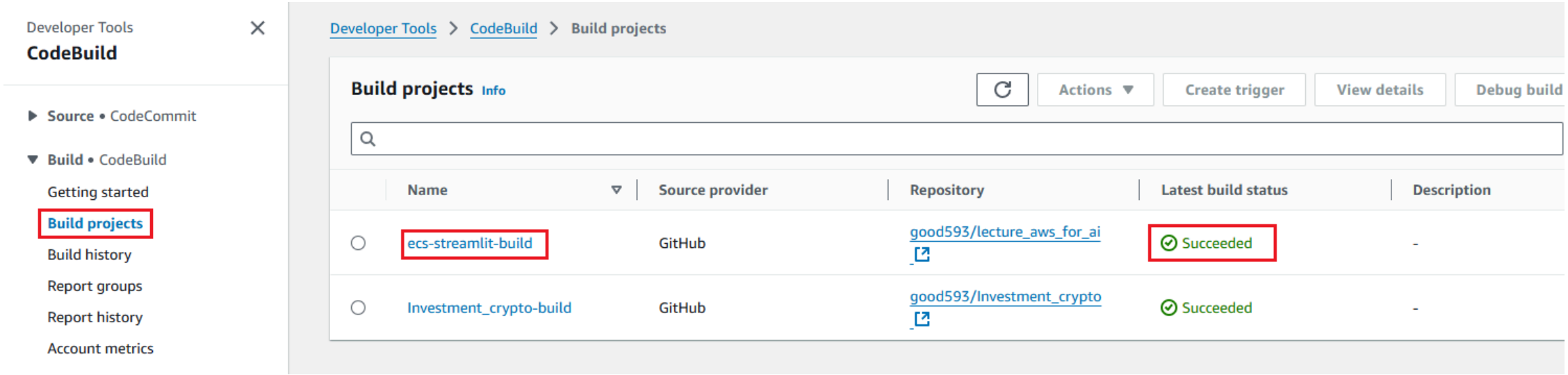

Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Archive rules
Analyzers
Policy was successfully attached to role.
Creation date
April 13, 2025, 08:50 (UTC+09:00)
Last activity
-
ARN
arn:aws:iam::426653742146:role/service-role/codebuild-ecs-streamlit-build-service-role
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Permissions policies (2) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter by Type
Search
All types
1
Policy name
Type
Attached entities
AmazonEC2ContainerRegistryPowerUser
AWS managed
1
CodeBuildBasePolicy-ecs-streamlit-build-ap-northeast-2
Customer managed
1

Start CodeBuild

단계1: Build projects > Start now



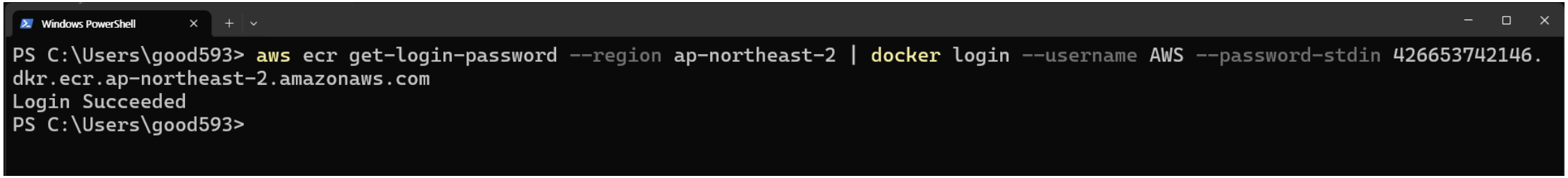
단계3: Build status > Succeeded



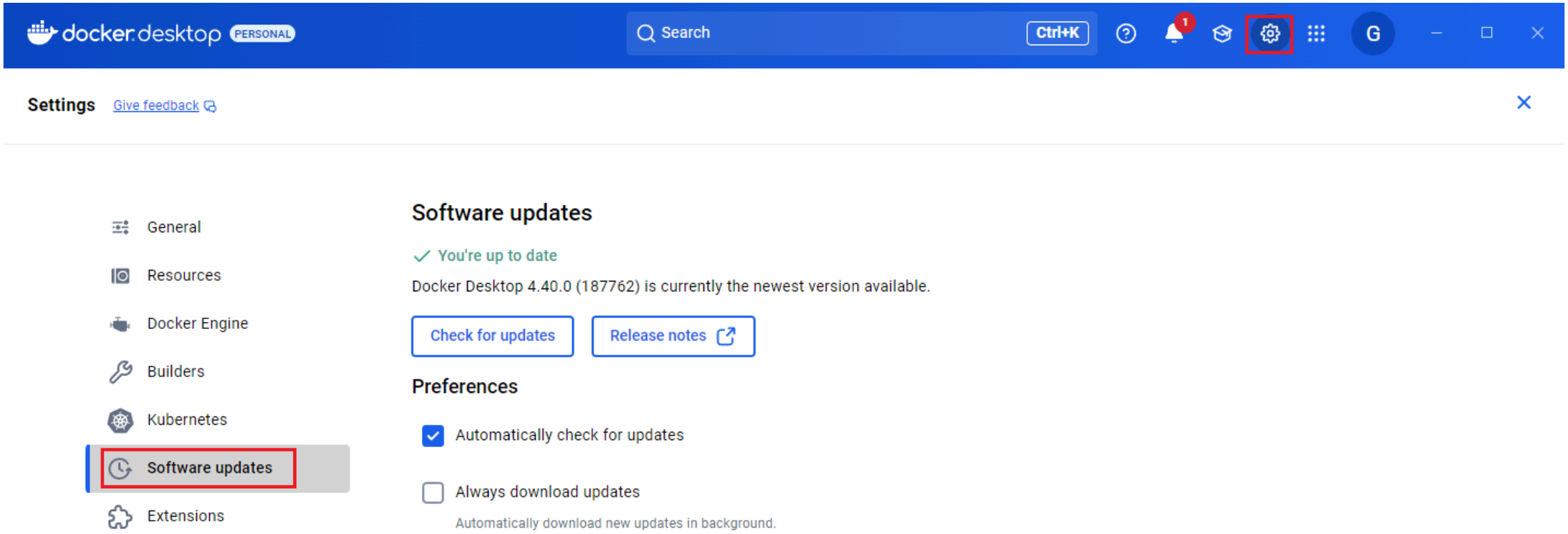
단계4: AWS ECR 로그인

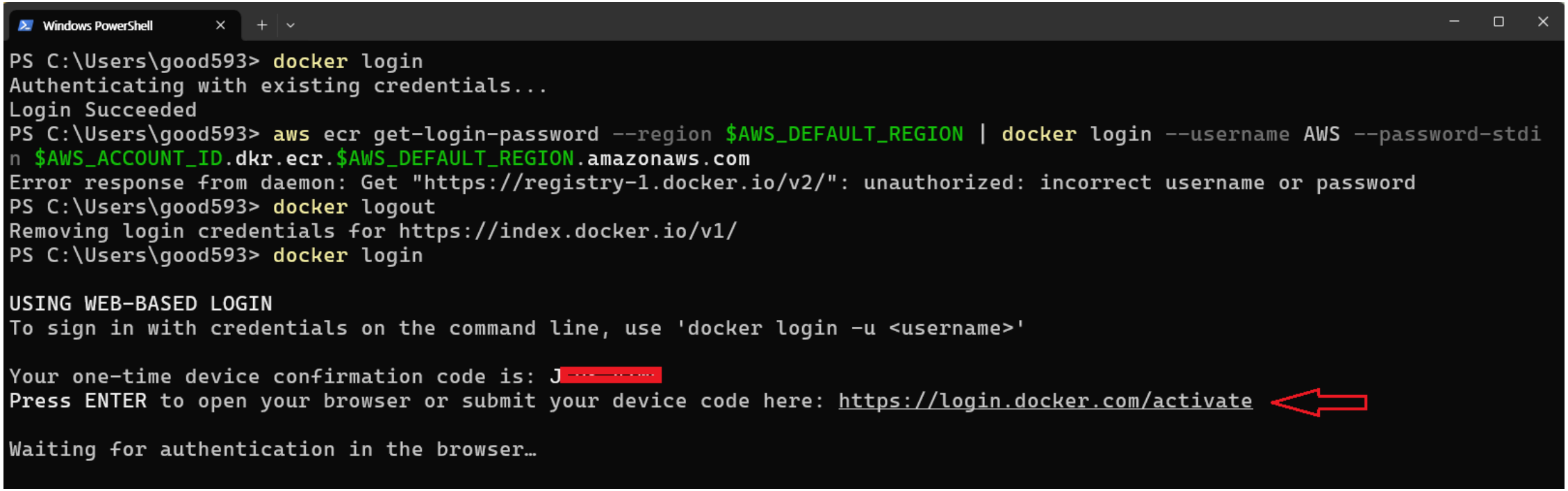
```
# 윈도우
$AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)
$AWS_DEFAULT_REGION="ap-northeast-2"
# 로그인
aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com

# 만약 오류가 발생하면.. 아이디와 리전을 직접 입력..
aws ecr get-login-password --region ap-northeast-2 | docker login --username AWS --password-stdin 426653742146.dkr.ecr.ap-northeast-2.amazonaws.com
```

```
# 리눅스
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)
AWS_DEFAULT_REGION=ap-northeast-2
# 로그인
aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
```

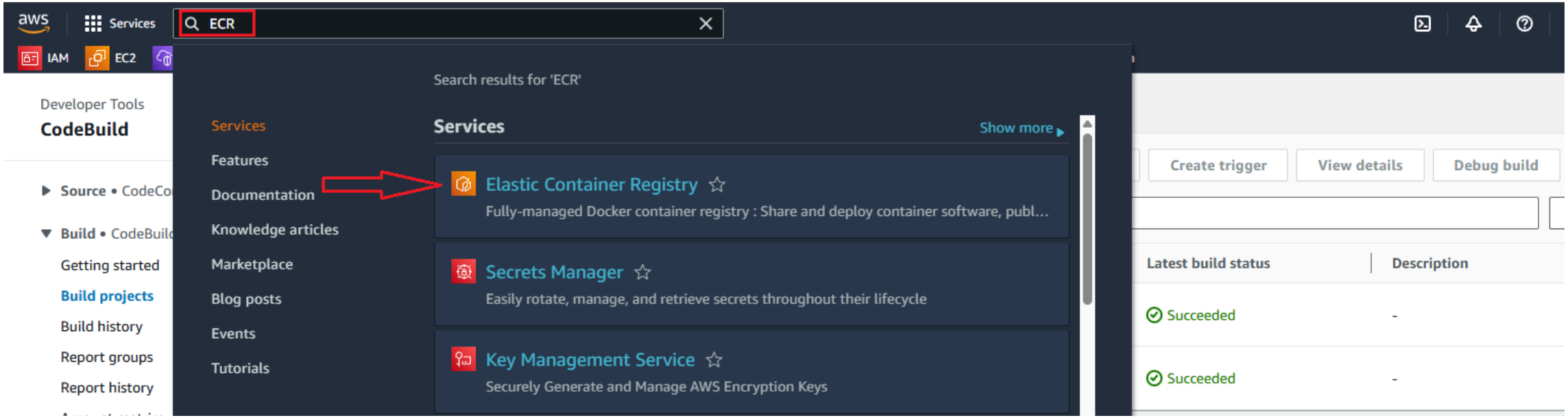


- 만약 오류발생시 참고

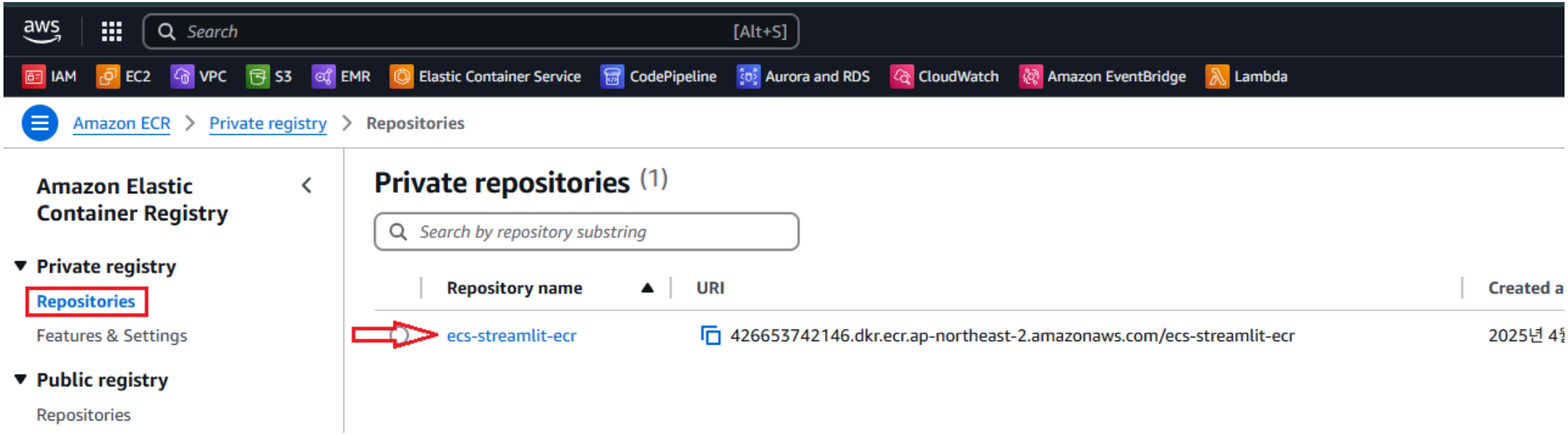

docker desktop PERSONAL
Search
Ctrl+K
G
Settings
Give feedback
General
Resources
Docker Engine
Builders
Kubernetes
Software updates
Extensions
Software updates
You're up to date
Docker Desktop 4.40.0 (187762) is currently the newest version available.
Check for updates
Release notes
Preferences
Automatically check for updates
Always download updates
Automatically download new updates in background.


Windows PowerShell
PS C:\Users\good593> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\Users\good593> aws ecr get-login-password --region $AWS_DEFAULT_REGION | docker login --username AWS --password-stdin $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
PS C:\Users\good593> docker logout
Removing login credentials for https://index.docker.io/v1/
PS C:\Users\good593> docker login
USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'
Your one-time device confirmation code is: J
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate
Waiting for authentication in the browser…

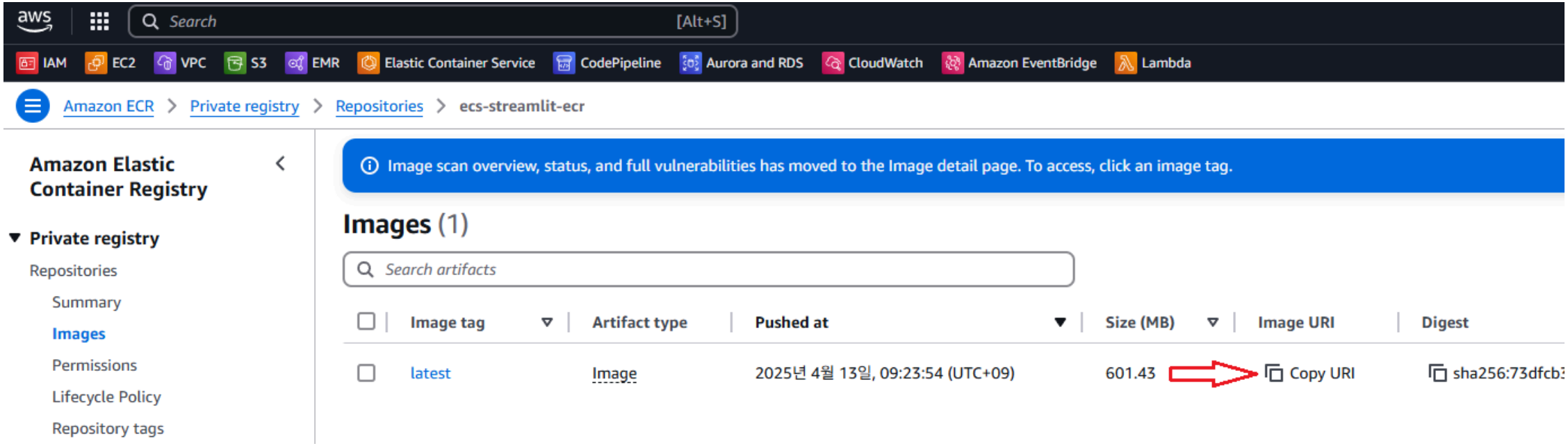
단계5: ECR 접속


aws
Services
ECR
IAM
EC2
Search results for 'ECR'
Developer Tools
CodeBuild
Services
Features
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials
Services
Show more
Elastic Container Registry
Fully-managed Docker container registry : Share and deploy container software, publ...
Secrets Manager
Easily rotate, manage, and retrieve secrets throughout their lifecycle
Key Management Service
Securely Generate and Manage AWS Encryption Keys
Source • CodeCo
Build • CodeBuil
Getting started
Build projects
Build history
Report groups
Report history
Create trigger
View details
Debug build
Latest build status
Description
Succeeded
-
Succeeded
-

단계6: ecs-streamlit-ecr

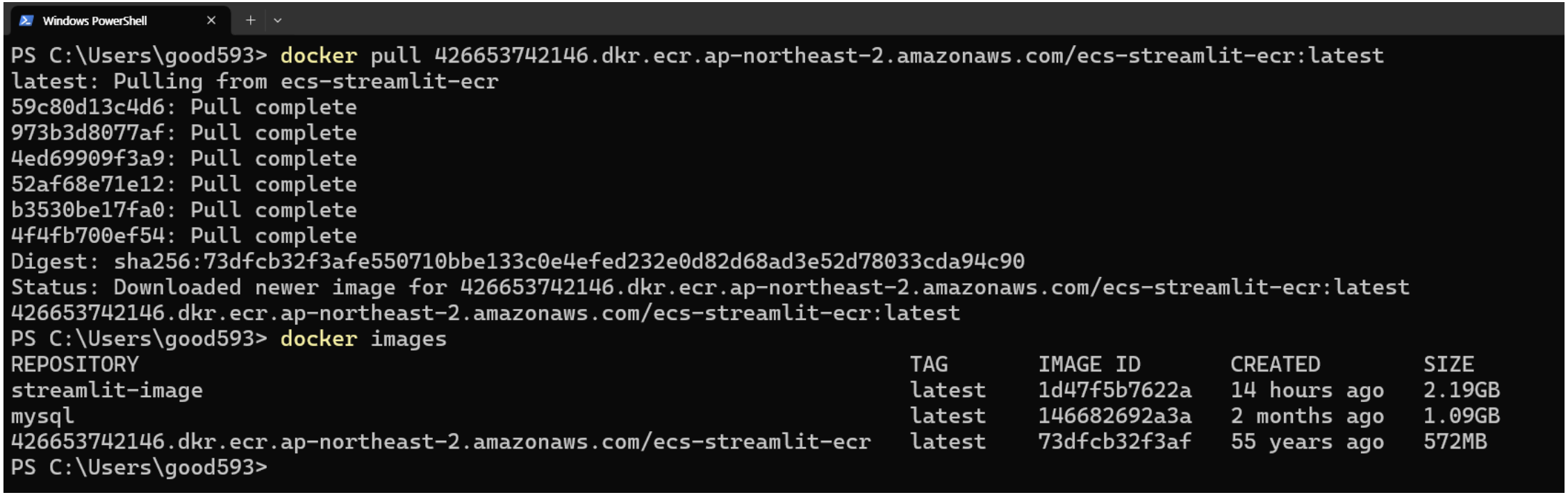


단계7: Copy URI


aws
Search
[Alt+S]
IAM
EC2
VPC
S3
EMR
Elastic Container Service
CodePipeline
Aurora and RDS
CloudWatch
Amazon EventBridge
Lambda
Amazon ECR
Private registry
Repositories
ecs-streamlit-ecr
Amazon Elastic Container Registry
Private registry
Repositories
Summary
Images
Permissions
Lifecycle Policy
Repository tags
Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.
Images (1)
Search artifacts
Image tag
Artifact type
Pushed at
Size (MB)
Image URI
Digest
latest
Image
2025년 4월 13일, 09:23:54 (UTC+09)
601.43
Copy URI
sha256:73dfcb

단계8: docker pull Image

```
docker pull [copied uri]
```

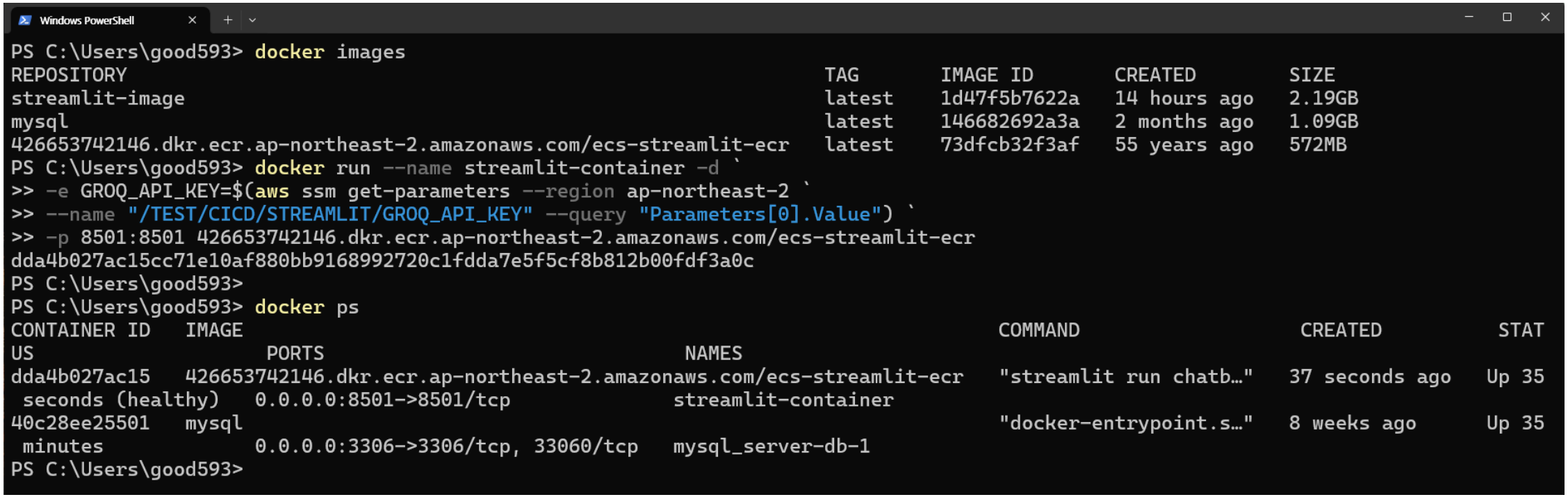


단계9: Run container

- 명령어: `docker run --name [컨테이너명] -d -e [환경변수] -p 8501:8501 [이미지명]`

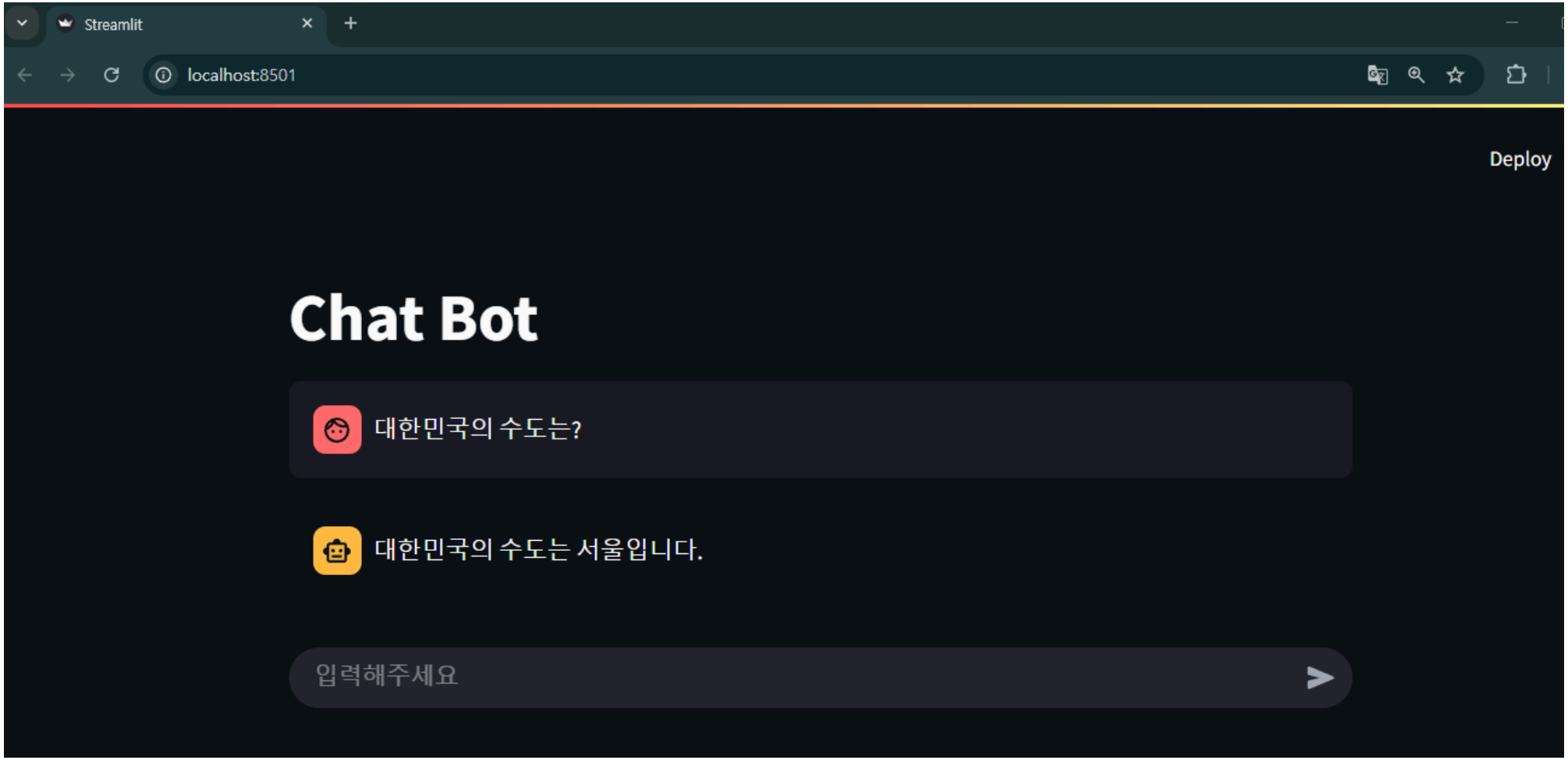
```
# 윈도우
docker run --name streamlit-container -d `
-e GROQ_API_KEY=$(aws ssm get-parameters --region ap-northeast-2 `
--name "/TEST/CICD/STREAMLIT/GROQ_API_KEY" --query "Parameters[0].Value") `
-p 8501:8501 [이미지명]

docker ps # 실행중인 컨테이너 확인
```

```
# 리눅스
docker run --name streamlit-container -d \
-e GROQ_API_KEY=$(aws ssm get-parameters --region ap-northeast-2 \
--name "/TEST/CICD/STREAMLIT/GROQ_API_KEY" --query "Parameters[0].Value" | tr -d '"') \
-p 8501:8501 [이미지명]

docker ps # 실행중인 컨테이너 확인
```


Windows PowerShell
PS C:\Users\good593> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
streamlit-image latest 1d47f5b7622a 14 hours ago 2.19GB
mysql latest 146682692a3a 2 months ago 1.09GB
426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr latest 73dfcb32f3af 55 years ago 572MB
PS C:\Users\good593> docker run --name streamlit-container -d `
>> -e GROQ_API_KEY=$(aws ssm get-parameters --region ap-northeast-2 `
>> --name "/TEST/CICD/STREAMLIT/GROQ_API_KEY" --query "Parameters[0].Value") `
>> -p 8501:8501 426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr
dda4b027ac15cc71e10af880bb9168992720c1fdda7e5f5cf8b812b00fdf3a0c
PS C:\Users\good593>
PS C:\Users\good593> docker ps
CONTAINER ID IMAGE COMMAND CREATED STAT
US PORTS NAMES
dda4b027ac15 426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr "streamlit run chatb…" 37 seconds ago Up 35
seconds (healthy) 0.0.0.0:8501->8501/tcp streamlit-container
40c28ee25501 mysql "docker-entrypoint.s…" 8 weeks ago Up 35
minutes 0.0.0.0:3306->3306/tcp, 33060/tcp mysql_server-db-1
PS C:\Users\good593>

단계10: 접속 및 실행

```
Local URL: http://localhost:8501
```



참고

```
docker stop streamlit-container # 컨테이너 멈춤
docker ps -a # 컨테이너 멈춤 확인
docker rm streamlit-container # 컨테이너 삭제
```