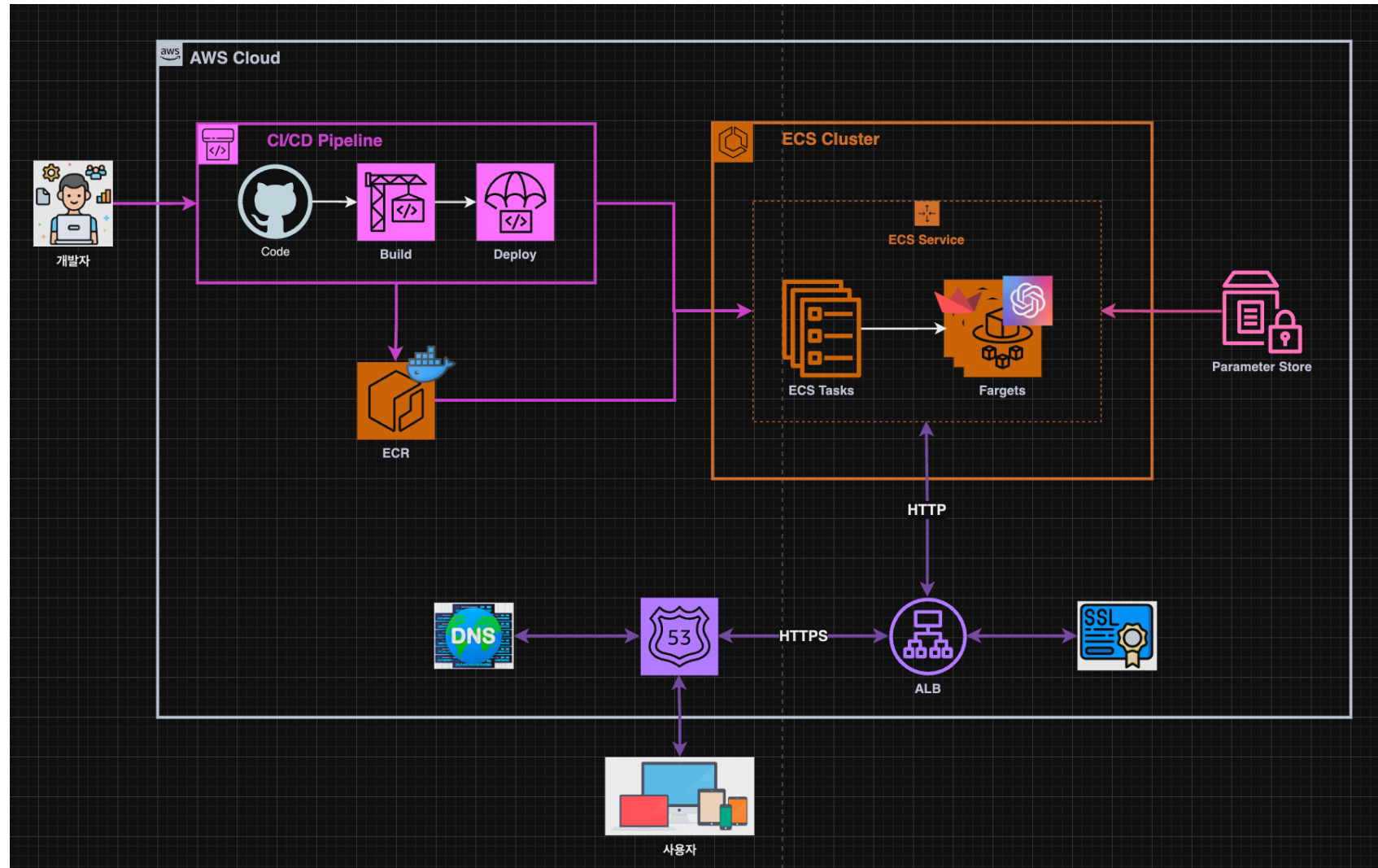
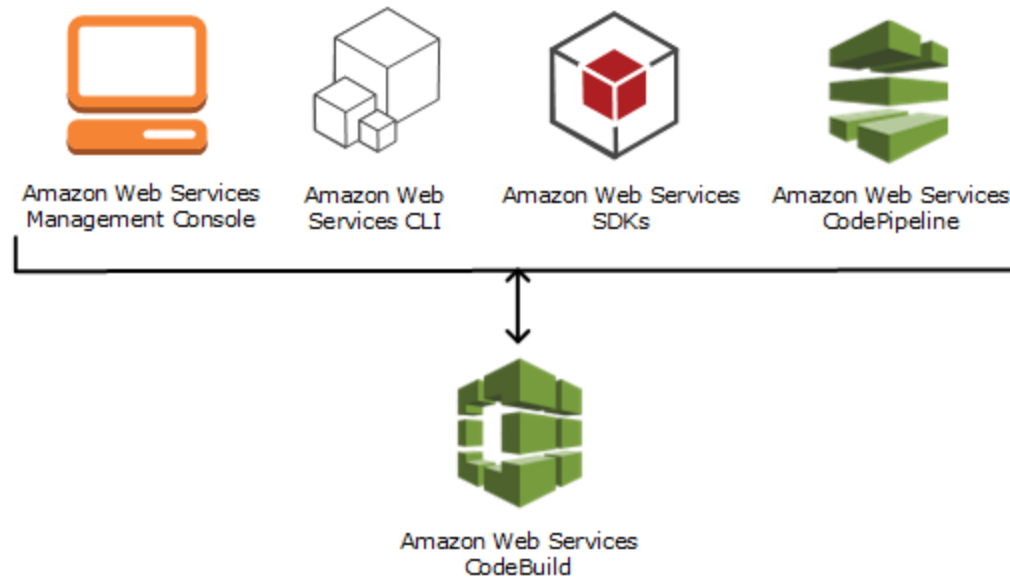


Architecture

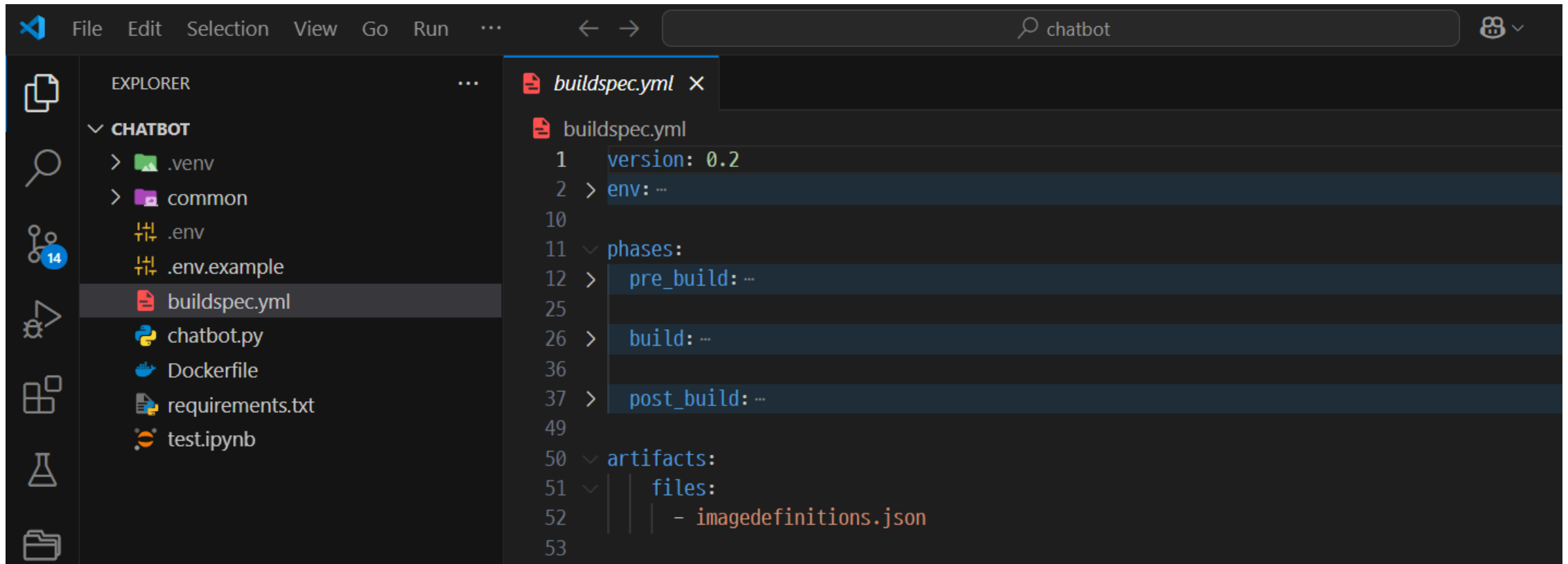


AWS CodeBuild

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



Create buildspec.yml



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Editor on the right. The Explorer sidebar shows a project named 'CHATBOT' with several files and folders. The 'buildspec.yml' file is selected and highlighted. The Editor shows the content of 'buildspec.yml' with line numbers 1 through 53. The file content is as follows:

```
1 version: 0.2
2 > env: ...
10
11 < phases:
12 >   pre_build: ...
25
26 >   build: ...
36
37 >   post_build: ...
49
50 < artifacts:
51 < |   files:
52 |   | - imagedefinitions.json
53
```

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

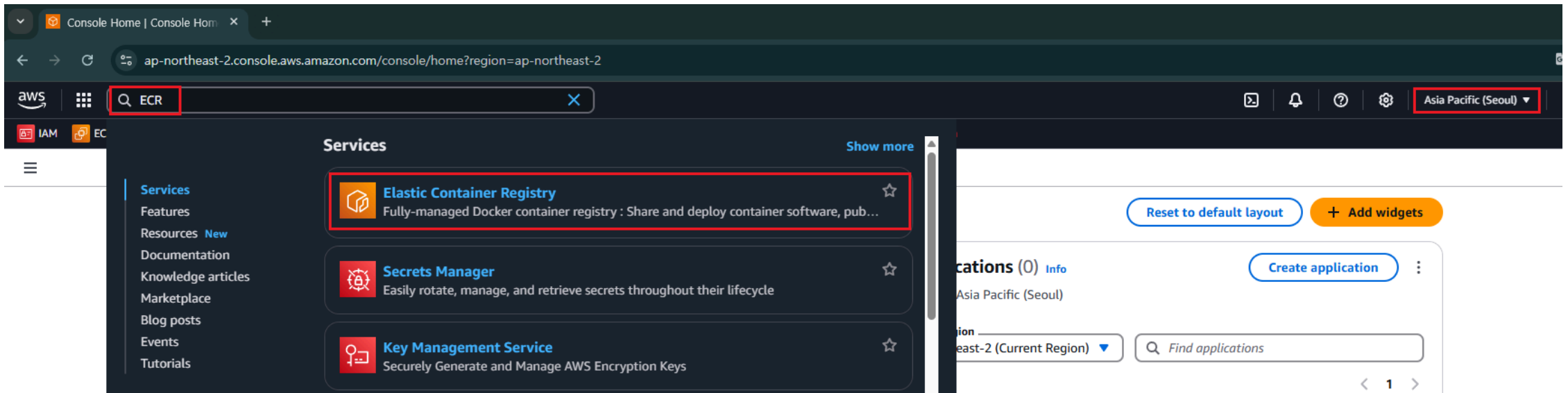
```
artifacts:
```

```
  files:
```

- imagedefinitions.json

Create AWS ECR

단계1: AWS ECR 접속



단계2: Create repository

The screenshot shows the Amazon Elastic Container Registry (ECR) console interface. The browser address bar displays the URL: `ap-northeast-2.console.aws.amazon.com/ecr/private-registry/repositories?region=ap-northeast-2`. The AWS navigation bar at the top includes the AWS logo, a search bar, and various service icons. The left sidebar shows the 'Amazon ECR' menu with 'Private registry' expanded, and 'Repositories' selected. The main content area is titled 'Private repositories' and features a search bar with the placeholder text 'Search by repository substring'. Below the search bar is a table with columns: 'Repository name', 'URI', 'Created at', 'Tag immutability', and 'Encryption type'. The table is currently empty, displaying the message 'No repositories' and 'No repositories were found'. In the top right corner of the main content area, there are buttons for 'View push commands', 'Delete', 'Actions', and a prominent orange 'Create repository' button.

단계3: Repository name

- buildspec.yml에 정의한 이름으로 설정!

Create private repository

General settings

Repository name

Enter a concise name. Repositories support namespaces, which you can use to group similar repositories.

426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/

17 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, and special characters . _ / .

Image tag mutability [Info](#)


Choose the tag mutability setting.

☒ **Mutable**
Image tags can be overwritten.

☐ **Immutable**
Image tags can't be overwritten.

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*



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)

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

View push commands Delete Actions Create repository

- 만들어진 ECR에는 등록된 Image가 없음

Amazon ECR > Private registry > Repositories > ecs-streamlit-ecr

Amazon Elastic Container Registry

▼ Private registry

Repositories

Summary

Images

Permissions

Lifecycle Policy

Repository tags

Features & Settings

▼ Public registry

Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

Images (0)

Search artifacts

Image tag

Artifact type

Pushed at

Size (MB)

Image URI

Digest

Last recorded pull time

No images

No images to display

14

Create CodeBuild

단계1: CodePipeline 접속

The screenshot displays the AWS Management Console interface. At the top, the search bar contains 'CodePipeline'. The left-hand navigation menu is open, showing the 'Amazon ECR' section. The main content area is titled 'Services' and features a card for 'CodePipeline' with the description 'Release Software using Continuous Delivery', which is highlighted by a red rectangular box. Below this, a 'Resources' section contains a blue notification banner titled 'Introducing resource search' with a 'Go to Resource Explorer' button. The right-hand pane shows a table with headers: 'Image (MB)', 'Image URI', 'Digest', and 'Last recorded pull time'. The table currently displays 'No images' and 'No images to display'.

단계2: Create project

Developer Tools

CodeBuild

▶ Source • CodeCommit

▼ Build • CodeBuild

Getting started

Build projects

Build history


Report groups


Report history




Account metrics

Developer Tools > CodeBuild > Build projects

Build projects Info

 Actions ▼ Create trigger View details Debug build Start build ▼ **Create project**

Your projects ▼ < 1 > 

	Name ▼	Source provider	Repository	Latest build status	Description	Last Modified
	Investment_crypto-build	GitHub	good593/Investment_crypto 	 Succeeded	-	4 months ago

단계3: Project configuration


Project configuration

Project name

ecs-streamlit-build

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

Project type

Select what type of project you would like to create. [Info](#) 

☒ **Default project**

Create a custom CodeBuild project.

☐ **Runner project**

Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.

▶ **Additional configuration**

Description, public build access, build badge, concurrent build limit, tags

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

▼ Primary source webhook events [Info](#)

Webhook - *optional* [Info](#) [↗](#)

☒ Rebuild every time a code change is pushed to this repository

Build type

☒ **Single build**
Triggers single build

☐ **Batch build**
Triggers multiple builds as single execution

▶ Webhook event filter groups

[Add filter group](#)

A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.

▶ Additional configuration

단계6: Environment

Environment image

☒ **Managed image**
Use an image managed by AWS CodeBuild

☐ **Custom image**
Specify a Docker image

Running mode

☒ **Container**
Running on Docker container

☐ **Instance**
Running on EC2 instance directly

Operating system

Amazon Linux ▼

Runtime(s)

Standard ▼

Image

aws/codebuild/amazonlinux-x86_64-standard:5.0 ▼

Image version

Always use the latest image for this runtime version ▼

- 생성된 CodeBuild Role Name 확인

☐ Use GPU-enhanced compute

Service role



New service role

Create a service role in your account



Existing service role

Choose an existing service role from your account

Role name

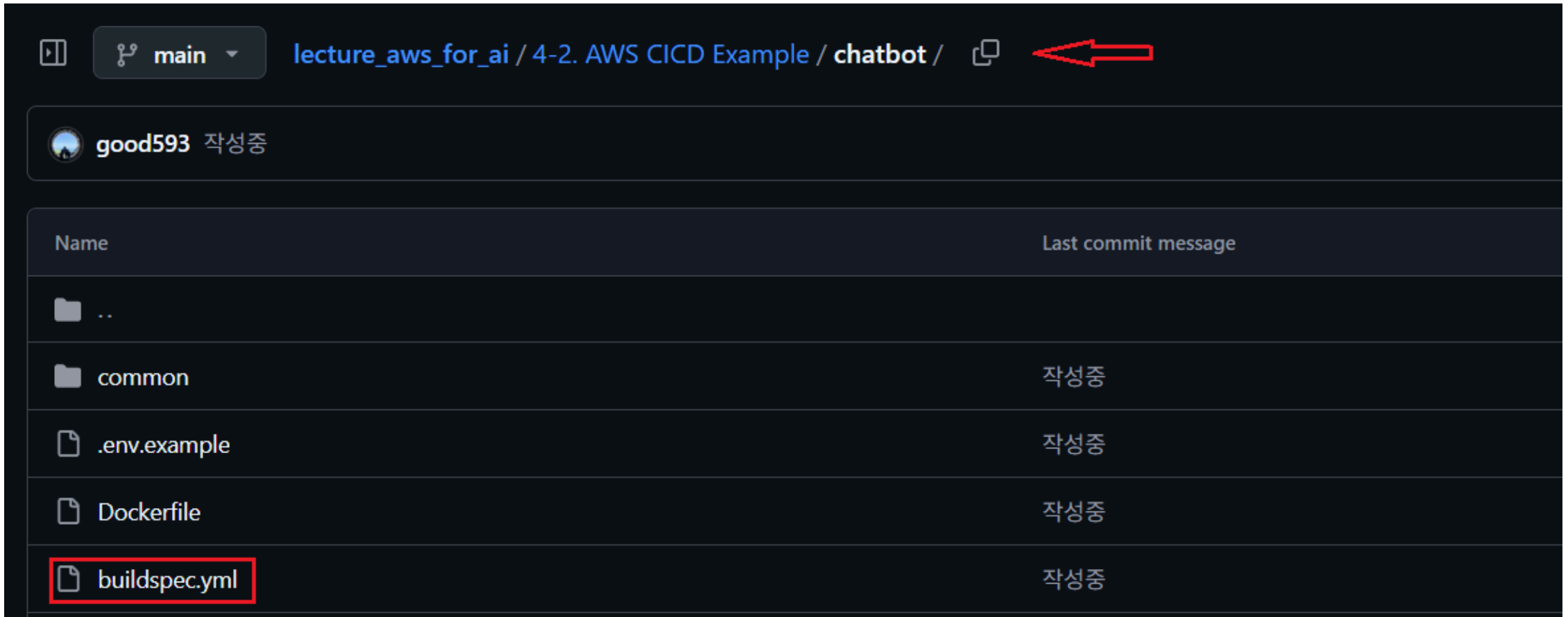
codebuild-ecs-streamlit-build-service-role

Type your service role name

► Additional configuration

Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, registry credential

단계7: Github에서 buildspec.yml 위치 확인



The screenshot shows the GitHub interface for a repository named 'lecture_aws_for_ai / 4-2. AWS CICD Example / chatbot'. The 'main' branch is selected. A red arrow points to the repository path. Below the repository name, the status 'good593 작성중' is shown. A table lists the files and folders in the repository:

Name	Last commit message
..	
common	작성중
.env.example	작성중
Dockerfile	작성중
buildspec.yml	작성중

The file 'buildspec.yml' is highlighted with a red box.

- Buildspec name에 해당 위치 정보 작성

▼ Buildspec

Build specifications



Insert build commands

Store build commands as build project configuration



Use a buildspec file

Store build commands in a YAML-formatted buildspec file

Buildspec name - *optional*

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

4-2. AWS CICD Example/chatbot/buildspec.yml



단계7: Create build project

▶ **Batch configuration**
You can run a group of builds as a single execution. Batch configuration is also available in advanced option when starting build.


▶ **Artifacts**

Add artifact

▶ **Logs**

Cancel

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

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

Analzers

codebuild-ecs-streamlit-build-service-role [Info](#)

Delete

Summary

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

Edit

Permissions

Trust relationships

Tags

Last Accessed

Revoke sessions

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Search

Filter by Type
All types

Simulate

Remove

Add permissions

Attach policies

Create inline policy

1

<input type="checkbox"/>	Policy name ?	Type	Attached entities
<input type="checkbox"/>	CodeBuildBasePolicy-ecs-streamlit-build-ap-northeast-2	Customer managed	1

단계3: Add AmazonEC2ContainerRegistryPowerUser

AmazonEC2ContainerRegistryPowerUser

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


Attach policy to codebuild-ecs-streamlit-build-service-role

▶ Current permissions policies (1)

Other permissions policies (1/1043)

Filter by Type 1 match

< 1 >

<input checked="" type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	 AmazonEC2ContainerRegistryPowerUser	AWS managed	Provides full access to Amazon EC2 Co...

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

↺

Simulate

Remove

Add permissions

Search

Filter by Type

All types

< 1 > ⚙

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AmazonEC2ContainerRegistryPowerUser	AWS managed	1
<input type="checkbox"/>	CodeBuildBasePolicy-ecs-streamlit-build-ap-northeast-2	Customer managed	1

Start CodeBuild

단계1: Build projects > Start now

Developer Tools

CodeBuild

► Source • CodeCommit

▼ Build • CodeBuild

Getting started

Build projects

Build history

Report groups

Report history

Account metrics

▼ Related integrations

Developer Tools > CodeBuild > Build projects

Build projects Info

↻

Actions ▼

Create trigger

View details

Debug build

Start build ▲

Create

	Name	Source provider	Repository	Latest build status	Description	Last Modified
<input checked="" type="radio"/>	ecs-streamlit-build	GitHub	good593/lecture_aws_for_ai	-	-	7 minutes ago
<input type="radio"/>	Investment_crypto-build	GitHub	good593/Investment_crypto	✔ Succeeded	-	4 months ago

단계3: Build status > Succeeded

Developer Tools

CodeBuild

▶ Source • CodeCommit

▼ Build • CodeBuild

Getting started

Build projects

Build history

Report groups

Report history

Account metrics

Developer Tools > CodeBuild > Build projects

Build projects Info

⌂ Actions ▼ Create trigger View details Debug build

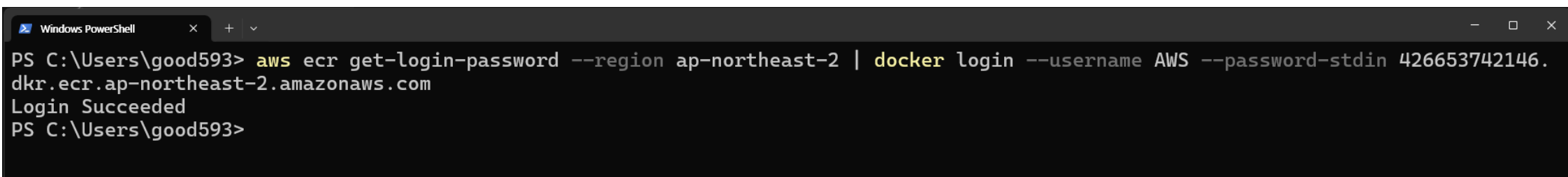
	Name ▼	Source provider	Repository	Latest build status	Description
<input type="radio"/>	ecs-streamlit-build	GitHub	good593/lecture_aws_for_ai	✔ Succeeded	-
<input type="radio"/>	Investment_crypto-build	GitHub	good593/Investment_crypto	✔ Succeeded	-

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

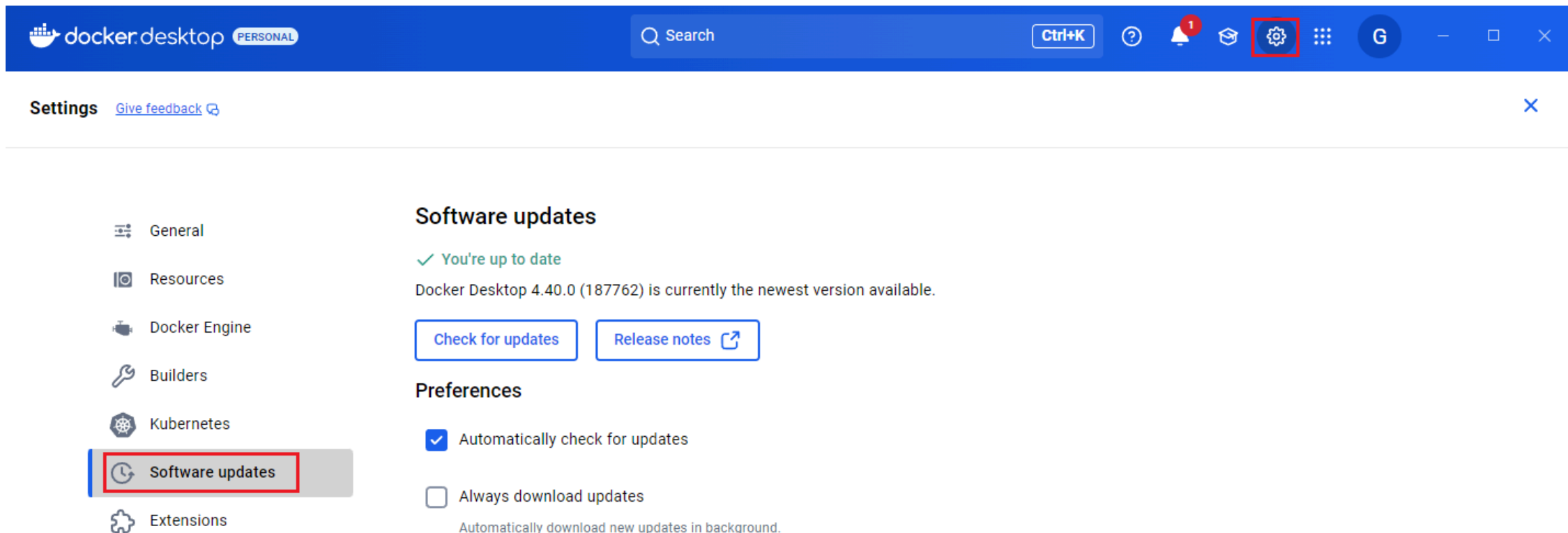
# 만약 오류가 발생하면.. 아이디와 리전을 직접 입력..
aws ecr get-login-password --region ap-northeast-2 | docker login --username AWS --password-stdin 426653742146.dkr.ecr.ap-northeast-2.amazonaws.com
```

```
# 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
```



```
Windows PowerShell
PS C:\Users\good593> aws ecr get-login-password --region ap-northeast-2 | docker login --username AWS --password-stdin 426653742146.dkr.ecr.ap-northeast-2.amazonaws.com
Login Succeeded
PS C:\Users\good593>
```

- 만약 오류발생시 참고



The screenshot shows the Docker Desktop application window. The title bar is blue with the Docker Desktop logo, a 'PERSONAL' badge, a search bar, a 'Ctrl+K' shortcut, and several icons including a help icon, a notification bell with a red '1', a mail icon, a settings gear icon (highlighted with a red box), a grid icon, and a user profile icon 'G'. Below the title bar, the 'Settings' window is open, showing a sidebar on the left with options: General, Resources, Docker Engine, Builders, Kubernetes, Software updates (highlighted with a red box), and Extensions. The main content area is titled 'Software updates' and shows a green checkmark with the text 'You're up to date'. Below this, it states 'Docker Desktop 4.40.0 (187762) is currently the newest version available.' and provides two buttons: 'Check for updates' and 'Release notes'. Under the 'Preferences' section, there are two checkboxes: 'Automatically check for updates' (checked) and 'Always download updates' (unchecked). A note below the second checkbox says 'Automatically download new updates in background.'

docker:desktop PERSONAL

Search Ctrl+K ? 1

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 [REDACTED]
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

The screenshot displays the AWS Management Console interface. At the top, the 'Services' search bar contains the text 'ECR'. Below the search bar, the 'Search results for 'ECR'' are shown. The 'Services' section lists three results: 'Elastic Container Registry', 'Secrets Manager', and 'Key Management Service'. A red arrow points to the 'Elastic Container Registry' entry. The right sidebar shows the 'Latest build status' as 'Succeeded'.

Search results for 'ECR'

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

Latest build status | **Description**

✓ Succeeded	-
✓ Succeeded	-

단계6: ecs-streamlit-ecr

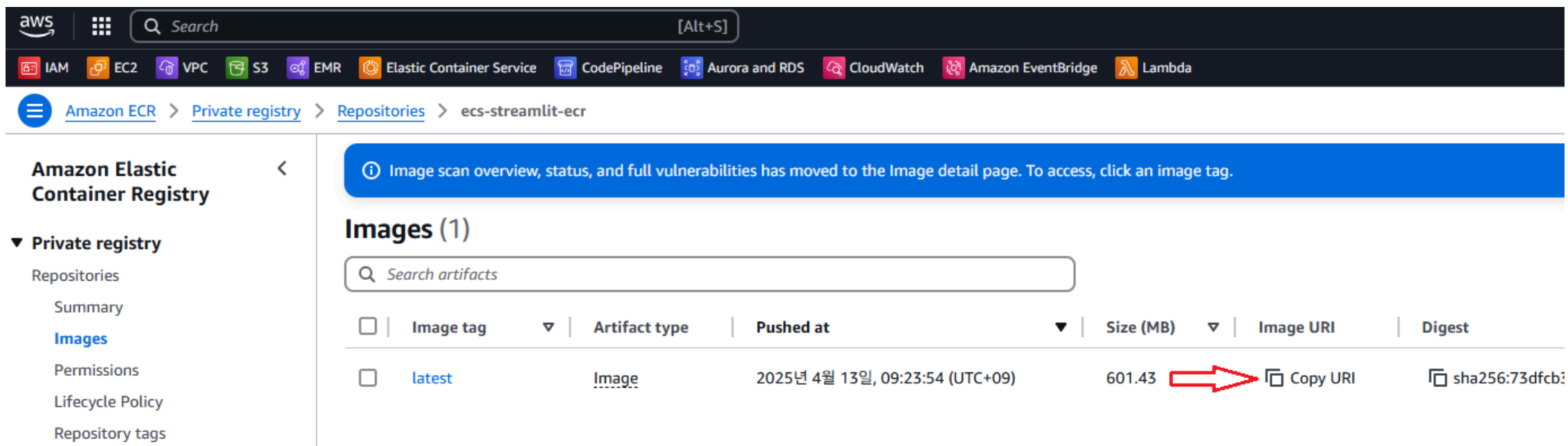
The screenshot shows the AWS Management Console interface for the Amazon Elastic Container Registry (ECR). The top navigation bar includes the AWS logo, a search bar, and a list of services: IAM, EC2, VPC, S3, EMR, Elastic Container Service, CodePipeline, Aurora and RDS, CloudWatch, Amazon EventBridge, and Lambda. The breadcrumb trail indicates the current location: Amazon ECR > Private registry > Repositories.

On the left sidebar, under 'Amazon Elastic Container Registry', the 'Private registry' section is expanded, and the 'Repositories' link is highlighted with a red box. Below it, 'Features & Settings' is listed. The 'Public registry' section is also visible with a 'Repositories' link.

The main content area is titled 'Private repositories (1)' and contains a search bar labeled 'Search by repository substring'. Below the search bar is a table with the following columns: 'Repository name', 'URI', and 'Created a'. A red arrow points to the first row of the table, which contains the repository 'ecs-streamlit-ecr'.

Repository name	URI	Created a
ecs-streamlit-ecr	426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr	2025년 4월

단계7: Copy URI



The screenshot shows the AWS Management Console interface for the Amazon Elastic Container Registry (ECR). The top navigation bar includes the AWS logo, a search bar, and a list of services. The left sidebar shows the 'Amazon Elastic Container Registry' menu with options like 'Private registry', 'Repositories', 'Summary', 'Images', 'Permissions', 'Lifecycle Policy', and 'Repository tags'. The main content area displays the 'Images (1)' page for the 'ecs-streamlit-ecr' repository. A blue banner at the top of the main area states: 'Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.' Below this, there is a search bar for artifacts. A table lists the images, with columns for 'Image tag', 'Artifact type', 'Pushed at', 'Size (MB)', 'Image URI', and 'Digest'. The table contains one entry with the tag 'latest', artifact type 'Image', pushed at '2025년 4월 13일, 09:23:54 (UTC+09)', size '601.43', and a 'Copy URI' button. A red arrow points to this button.

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

<input type="checkbox"/>	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
<input type="checkbox"/>	latest	Image	2025년 4월 13일, 09:23:54 (UTC+09)	601.43	 Copy URI	 sha256:73dfcb3

단계8: docker pull Image

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

```
Windows PowerShell
PS C:\Users\good593> docker pull 426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr:latest
latest: Pulling from ecs-streamlit-ecr
59c80d13c4d6: Pull complete
973b3d8077af: Pull complete
4ed69909f3a9: Pull complete
52af68e71e12: Pull complete
b3530be17fa0: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:73dfcb32f3afe550710bbe133c0e4efed232e0d82d68ad3e52d78033cda94c90
Status: Downloaded newer image for 426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr:latest
426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr:latest
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>
```

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` # 실행 중인 컨테이너 확인

리눅스

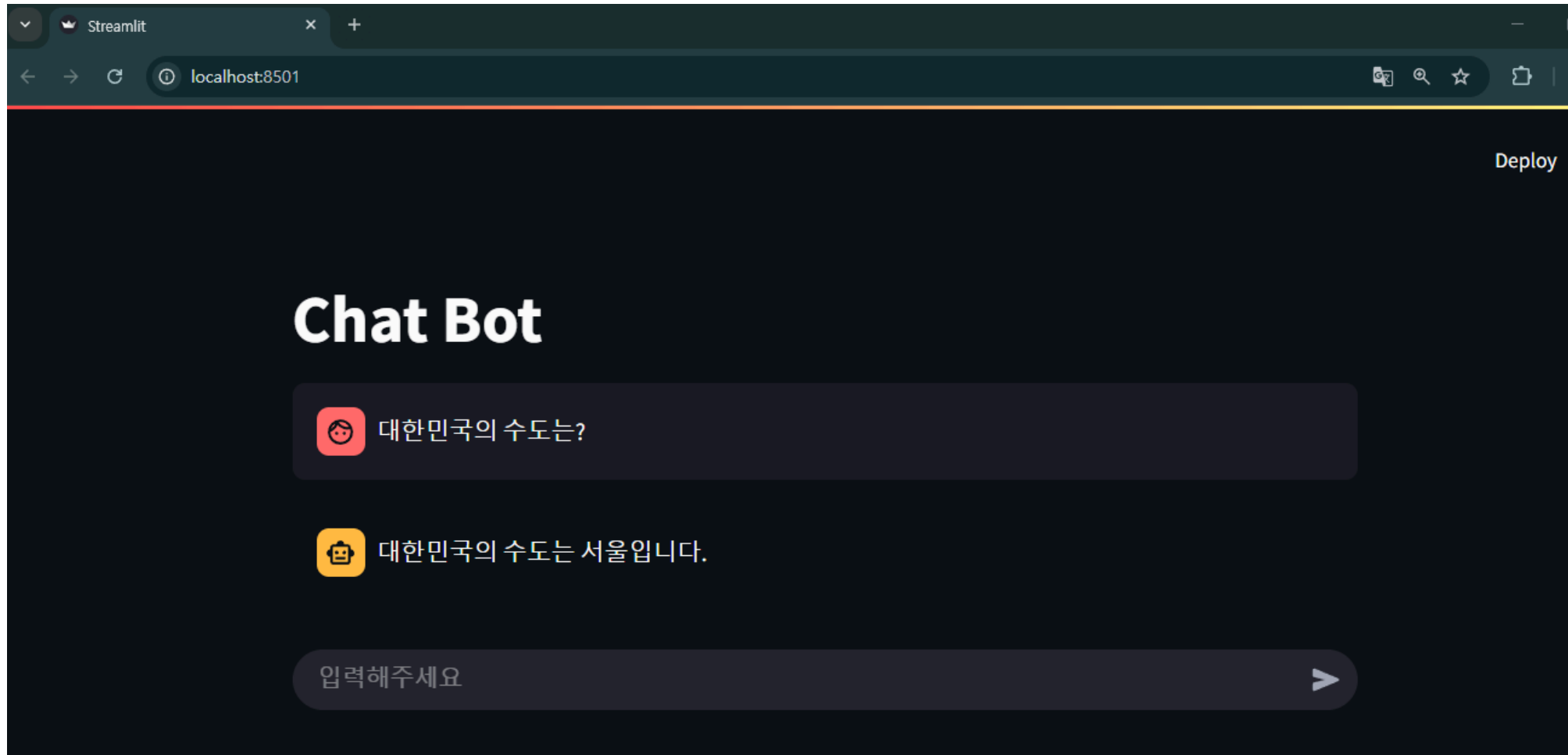
```
docker run --name streamlit-container -d \
-e GROQ_API_KEY=$(aws ssm get-parameters --region ap-northeast-2 \
--name "/TEST/CICD/STREAMLIT/GROQ_API_KEY" --query "Parameters[0].Value" | tr -d '"') \
-p 8501:8501 [이미지명]
```

`docker ps` # 실행 중인 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>
```

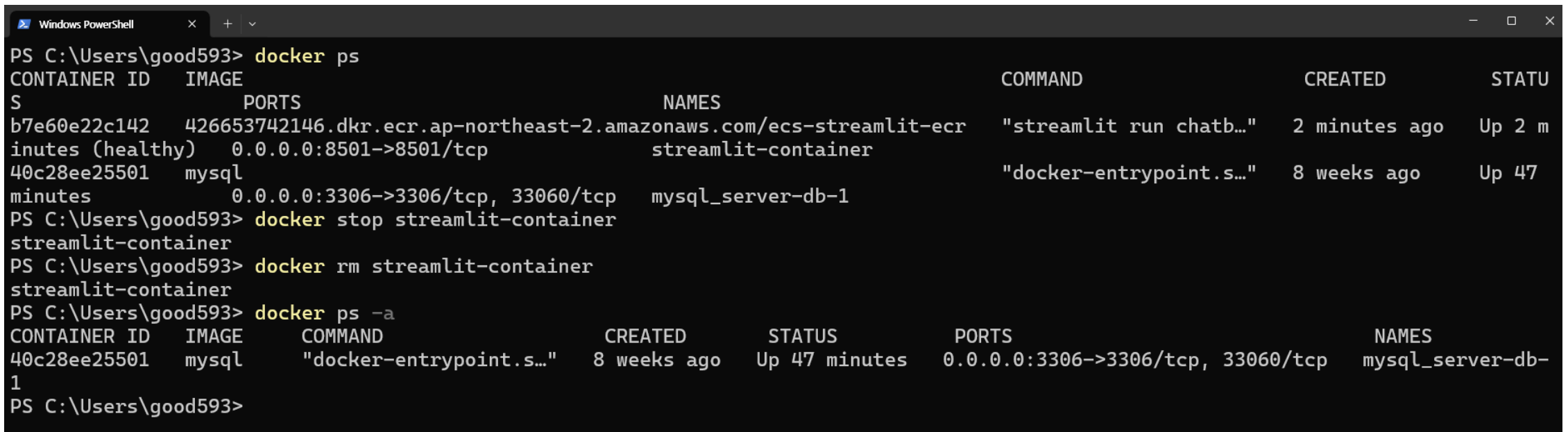
단계10: 접속 및 실행

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



참고

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



```
Windows PowerShell
PS C:\Users\good593> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
b7e60e22c142   426653742146.dkr.ecr.ap-northeast-2.amazonaws.com/ecs-streamlit-ecr "streamlit run chatb..." 2 minutes ago Up 2 m
inutes (healthy) 0.0.0.0:8501->8501/tcp
streamlit-container
40c28ee25501   mysql                                "docker-entrypoint.s..." 8 weeks ago   Up 47
minutes         0.0.0.0:3306->3306/tcp, 33060/tcp
mysql_server-db-1
PS C:\Users\good593> docker stop streamlit-container
streamlit-container
PS C:\Users\good593> docker rm streamlit-container
streamlit-container
PS C:\Users\good593> docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
40c28ee25501   mysql                                "docker-entrypoint.s..." 8 weeks ago   Up 47 minutes  0.0.0.0:3306->3306/tcp, 33060/tcp  mysql_server-db-1
PS C:\Users\good593>
```