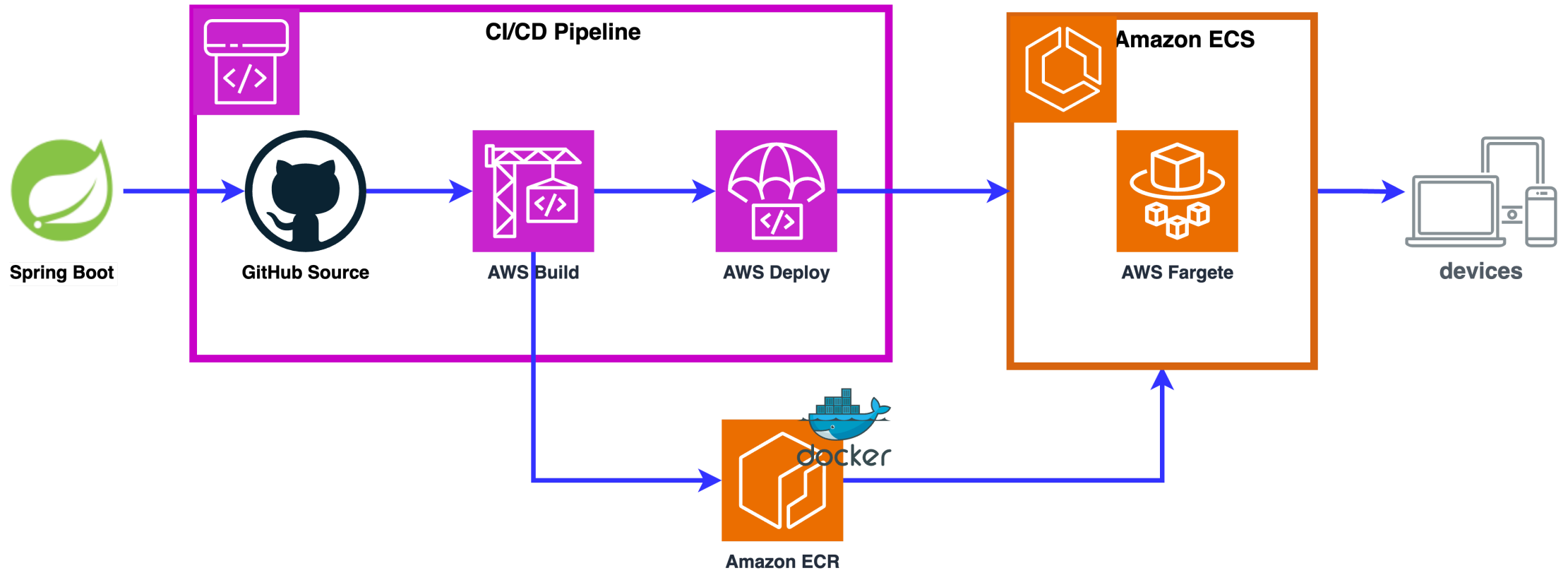
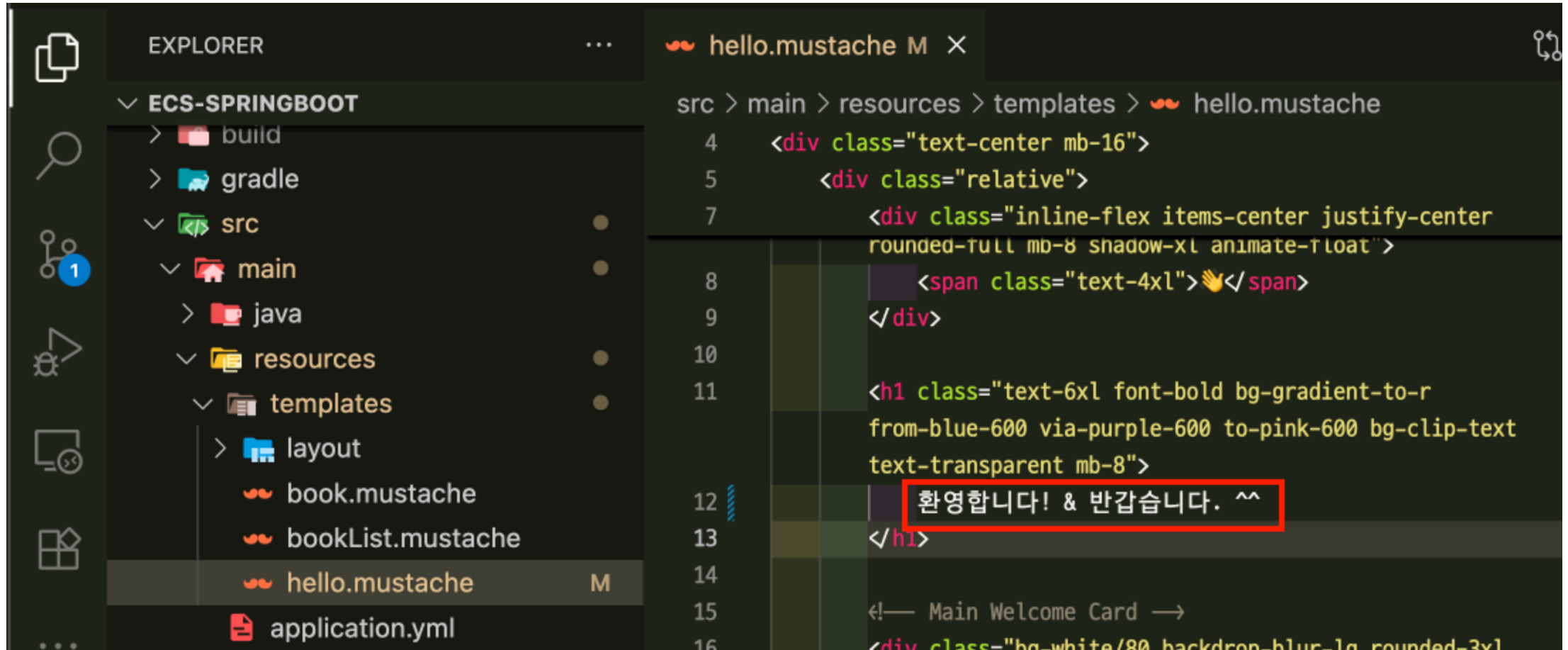


전체 아키텍처



코드 수정 및 Github 반영

단계1: 코드 수정

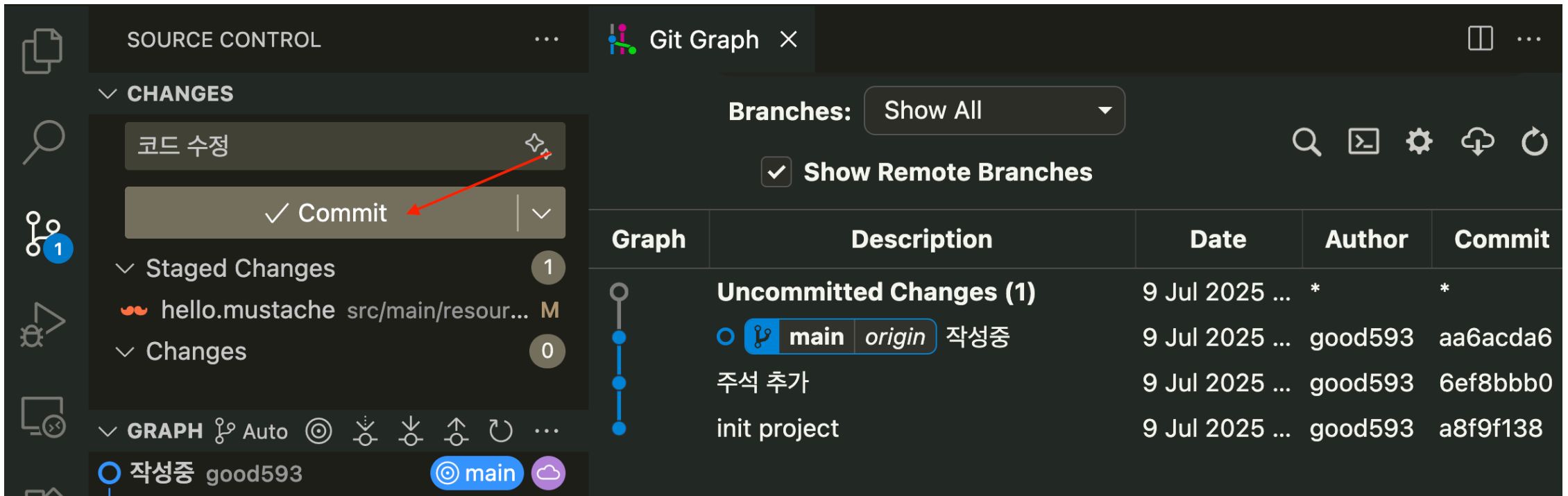


The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar showing the project structure for 'ECS-SPRINGBOOT'. The 'src' directory is expanded, showing 'main' > 'resources' > 'templates'. The file 'hello.mustache' is selected. The main editor displays the content of 'hello.mustache' with the following code:

```
src > main > resources > templates > 🍷 hello.mustache
4  <div class="text-center mb-16">
5    <div class="relative">
7      <div class="inline-flex items-center justify-center
        rounded-full mb-8 shadow-xl animate-float">
8        <span class="text-4xl">👋</span>
9      </div>
10
11     <h1 class="text-6xl font-bold bg-gradient-to-r
        from-blue-600 via-purple-600 to-pink-600 bg-clip-text
        text-transparent mb-8">
12       환영합니다! & 반갑습니다. ^^
13     </h1>
14
15     <!-- Main Welcome Card -->
16     <div class="bg-white/80 backdrop-blur-lg rounded-3xl
```

The text '환영합니다! & 반갑습니다. ^^' on line 12 is highlighted with a red rectangular box.

단계2: Commit



The screenshot shows the Git Graph interface in a code editor. The left sidebar contains icons for Source Control, Search, and other tools. The main area is divided into two panels. The top panel shows the 'CHANGES' section with a list of changes: '코드 수정' (Code Change) and 'hello.mustache' (src/main/resour...). The bottom panel shows the 'GRAPH' section with a list of changes: '작성중' (In Progress) by 'good593' on the 'main' branch.

The 'Commit' button is highlighted with a red arrow. The 'Commit' button is located in the 'CHANGES' section, below the '코드 수정' entry.

The 'GRAPH' section shows a list of changes with the following columns: Graph, Description, Date, Author, and Commit.

Graph	Description	Date	Author	Commit
	Uncommitted Changes (1)	9 Jul 2025 ...	*	*
main origin	작성중	9 Jul 2025 ...	good593	aa6acda6
	주석 추가	9 Jul 2025 ...	good593	6ef8bbb0
	init project	9 Jul 2025 ...	good593	a8f9f138

단계3: Github 반영

Git Graph interface showing commit history and options.

Commit History Table:

Graph	Description	Date	Author	Commit
○	○ main <i>origin</i> 코드 수정	9 Jul 2025 ...	good593	ec7e4e76
●	작성중	9 Jul 2025 ...	good593	aa6acda6
●	주석 추가	9 Jul 2025 ...	good593	6ef8bbb0
●	init project	9 Jul 2025 ...	good593	a8f9f138

Commit Options:

- Message (⌘Enter to commit ...)
- ✓ Commit

Branches: Show All

☒ Show Remote Branches

Bottom Bar:

- GRAPH Auto
- 코드 수정 good593
- main

 main **ecs-springboot** / src / main / resources / templates / **hello.mustache** 

 Go to file t ...

 **good593** 코드 수정  ec7e4e7 · 11 minutes ago  History

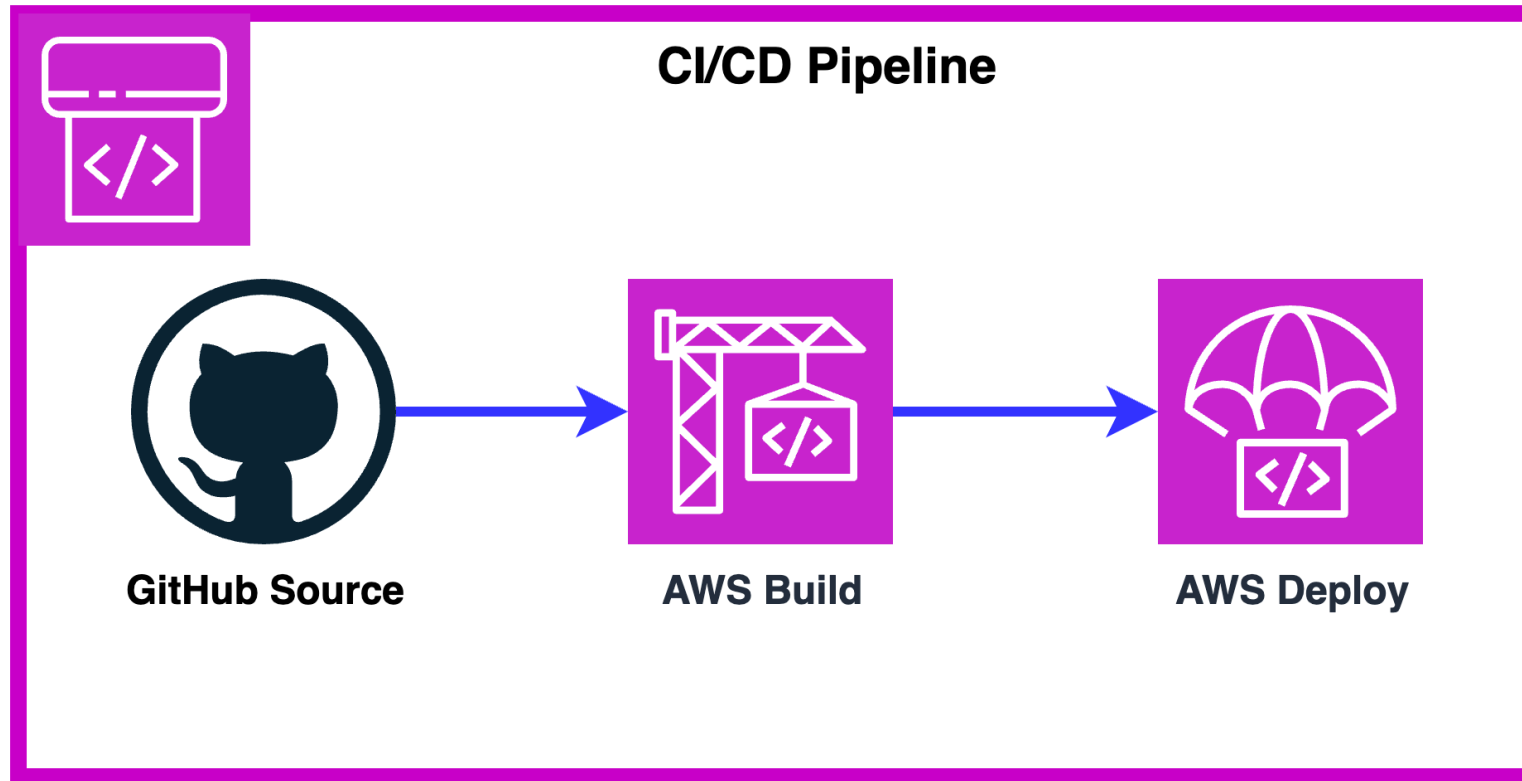
Code Blame 111 lines (99 loc) · 6.35 KB

Raw     

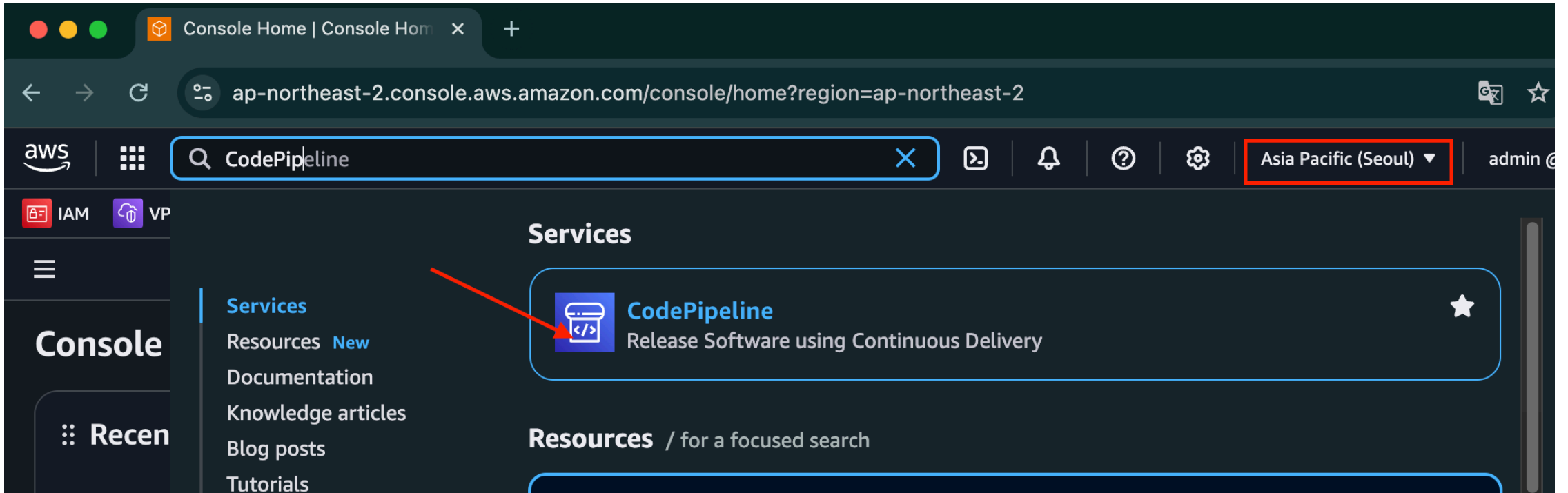
```
1  {{> layout/header}}
2
3  <!-- Welcome Hero Section -->
4  <div class="text-center mb-16">
5      <div class="relative">
6          <!-- Animated Welcome -->
7          <div class="inline-flex items-center justify-center w-24 h-24 bg-gradient-to-r from-blue-500 to-purple-600 rounded-full mb-8 shadow-xl ani
8              <span class="text-4xl">👋</span>
9          </div>
10
11         <h1 class="text-6xl font-bold bg-gradient-to-r from-blue-600 via-purple-600 to-pink-600 bg-clip-text text-transparent mb-8">
12             환영합니다! & 반갑습니다. ^^
13         </h1>
14
```

CodePipeline

- AWS CodePipeline은 AWS에서 제공하는 지속적 통합(CI) 및 지속적 배포(CD) 서비스를 위한 자동화 파이프라인 도구입니다.



단계1: CodePipeline 접속



단계2: Create pipeline

Developer Tools

CodePipeline

▶ Source • CodeCommit

▶ Build • CodeBuild

▶ Deploy • CodeDeploy

▼ Pipeline • CodePipeline

Getting started

Pipelines

Account metrics

Developer Tools > CodePipeline > Pipelines

Pipelines Info

↺

View history

Release change

Delete pipeline

Create pipeline

🔍

< 1 >

⚙️

	Name	Latest execution status	Latest source revisions	Latest execution started	Most recent executions
○	Investment_crypt to-pipeline	✔ Succeeded	Source – 4830a7a8 [↗] : elt 수정	7 months ago	✔✔✔✔ ✔

Step 1

Choose creation option

Step 2

Choose pipeline settings

Step 3

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

Choose creation option [Info](#)

Step 1 of 7

Category

☐ Deployment

☐ Continuous Integration

☐ Automation

☒ Build custom pipeline

Cancel

Next

Step 1

Choose creation option

Step 2

Choose pipeline settings

Step 3

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

Add deploy stage

Choose pipeline settings [Info](#)

Step 2 of 7

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Execution mode [Info](#)

Choose the execution mode for your pipeline. This determines how the pipeline is run.

- ☐ Superseded
- ☒ Queued
- ☐ Parallel

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

AWSCodePipelineServiceRole-ap-northeast-2-ecs-springboot-pipeli

Type your service role name



Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

► **Advanced settings**

Configure artifact store location, encryption settings, and pipeline variables for your pipeline.

Cancel

Previous

Next

단계3: Source(Github)

Step 3

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

Add deploy stage

Step 7

Review

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (via GitHub App) ▼

Connection

Choose an existing connection that you have already configured, or create a new one and then return to this task.

Q arn:aws:codeconnections:ap-northeast-2:426653742146:connect X



or

Connect to GitHub

Repository name

Choose a repository in your GitHub account.

Q good593/ecs-springboot X

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch

Default branch will be used only when pipeline execution starts from a different source or manually started.

Q main X


Output artifact format

Choose the output artifact format.

☒ CodePipeline default

AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

☐ Full clone

AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions. [Learn more](#) 

☒ Enable automatic retry on stage failure

Webhook events

Webhook - *optional*

☒ Start your pipeline on push and pull request events.

▶ Webhook event filters - *optional*

Starts your pipeline on a specific event.

Remove filters

Cancel

Previous

Next

단계4: Build

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

Add deploy stage

Step 7

Review

Build provider

Choose the tool you want to use to run build commands and specify artifacts for your build action.

☐ Commands

☒ Other build providers

AWS CodeBuild

Project name

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

Q ecs-springboot-build



or

Create project

☐ Define buildspec override - *optional*

Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

Environment variables - *optional*

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Add environment variable

Build type


☒ **Single build**
Triggers a single build.

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

Region

Asia Pacific (Seoul) ▼

Input artifacts

Choose an input artifact for this action. [Learn more](#) 

SourceArtifact ✕
Defined by: Source

☒ Enable automatic retry on stage failure

Cancel

Previous

Skip build stage

Next

단계5: Test

Step 1

Choose creation option

Step 2

Choose pipeline settings

Step 3

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

Add deploy stage

Add test stage Info

Step 5 of 7

Test - optional

Test provider

Choose how you want to test your application or content. Choose the provider, and then provide the configuration details for that provider.

☒ Enable automatic retry on stage failure

Cancel

Previous

Skip test stage

Next

단계6: Deploy

Add source stage

Step 4

Add build stage

Step 5

Add test stage

Step 6

Add deploy stage

Step 7

Review

Deploy provider

Choose how you want to deploy your application or content. Choose the provider, and then provide the configuration details for the provider.


Amazon ECS

Region

Asia Pacific (Seoul)

Input artifacts

Choose an input artifact for this action. [Learn more](#)

BuildArtifact 
Defined by: Build

No more than 100 characters

Cluster name

Choose a cluster that you have already created in the Amazon ECS console and then return to this task.

Q ecs-springboot-cluster

Service name

Choose a service that you have already created in the Amazon ECS console and then return to this task.

Q ecs-springboot-task-service-dq4rj1y6

Image definitions file - optional

Enter the JSON file that describes your service's container image definitions.

imagedefinitions.json

Deployment timeout - optional

Enter the timeout in minutes for the deployment action.

☒ Configure automatic rollback on stage failure

☐ Enable automatic retry on stage failure

buildspec.yml

buildspec.yml > {} artifacts > [] files

```
71 # 빌드 결과물 정의
72 artifacts:
73   files:
74     # ECS 배포에 필요한 이미지 정의 파일
75     # CodePipeline의 다음 단계(Deploy)에 사용
76     - imagedefinitions.json
```

<< main ↺ 0 3 1 ✓ AWS: profile:default

Cancel

Previous

Skip deploy stage

Next

ServiceName

ecs-springboot-task-service-dq4rj1y6

FileName

imagedefinitions.json

Configure automatic rollback on stage failure

Enabled

Enable automatic retry on stage failure



Disabled

Cancel

Previous


Create pipeline

단계7: 성공


ecs-springboot-pipeline  


[Edit](#) [Stop execution](#) [Create trigger](#) [Clone pipeline](#) [Release change](#)

[Pipeline](#) [Executions](#) [Triggers](#) [Settings](#) [Tags](#) [Stage](#)

✓ ✓ ✓ 


Source
f05e9540-b347-43c7-819b-2df91e6a5e9e
All actions succeeded.

Source
✓ [GitHub \(via GitHub App\)](#) 
5 minutes ago


ec7e4e76  Source: 코드 수정


Build
f05e9540-b347-43c7-819b-2df91e6a5e9e
All actions succeeded.

Build
✓ [AWS CodeBuild](#)
3 minutes ago

ec7e4e76  Source: 코드 수정

Deploy
f05e9540-b347-43c7-819b-2df91e6a5e9e
All actions succeeded.

Deploy
✓ [Amazon ECS](#) 
Just now

ec7e4e76  Source: 코드 수정

[+](#)

Source • CodeCommit

Build • CodeBuild

Deploy • CodeDeploy

▼ Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

Action details **New**

History

Settings

Pipeline Metrics

Account metrics

변경된 내용 확인

단계1: ECS Public IP 복사

The screenshot shows the Amazon Elastic Container Service (ECS) console. The breadcrumb navigation at the top reads: Amazon Elastic Container Service > ... > Tasks > 8611de4af44849859918e815b8bca986 > Configuration. The left sidebar contains links for Clusters, Namespaces, Task definitions, Account settings, and Amazon ECR. The main content area is titled 'Configuration' and displays various task configuration details in a table-like format.

Configuration	Capacity provider	ENI ID	Network mode	Subnet ID	Private IP	MAC address
Operating system/Architecture Linux/X86_64	-	eni-05e3e25b1092a7f6f	awsvpc	subnet-0b440944f5713d145	16.184.10.160 open address	06:60:6c:b1:b7:19
CPU Memory 1 vCPU 2 GB	Launch type FARGATE					
Platform version 1.4.0	Container instance ID -					
	Task definition: revision					

A notification bubble in the top right corner of the configuration table states 'Public IP copied' with a green checkmark. A red arrow points from this notification to the public IP address '16.184.10.160' in the 'Private IP' column.

단계2: 확인

http://[Public IP]:8080/hello

