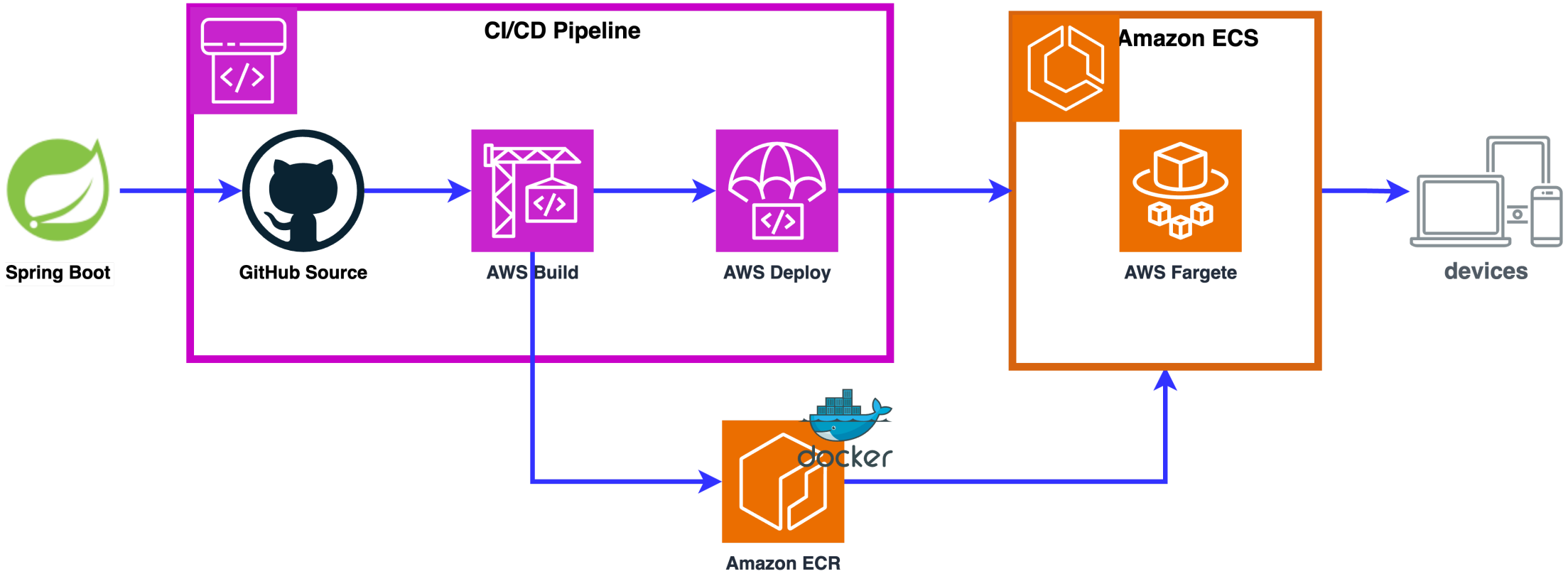
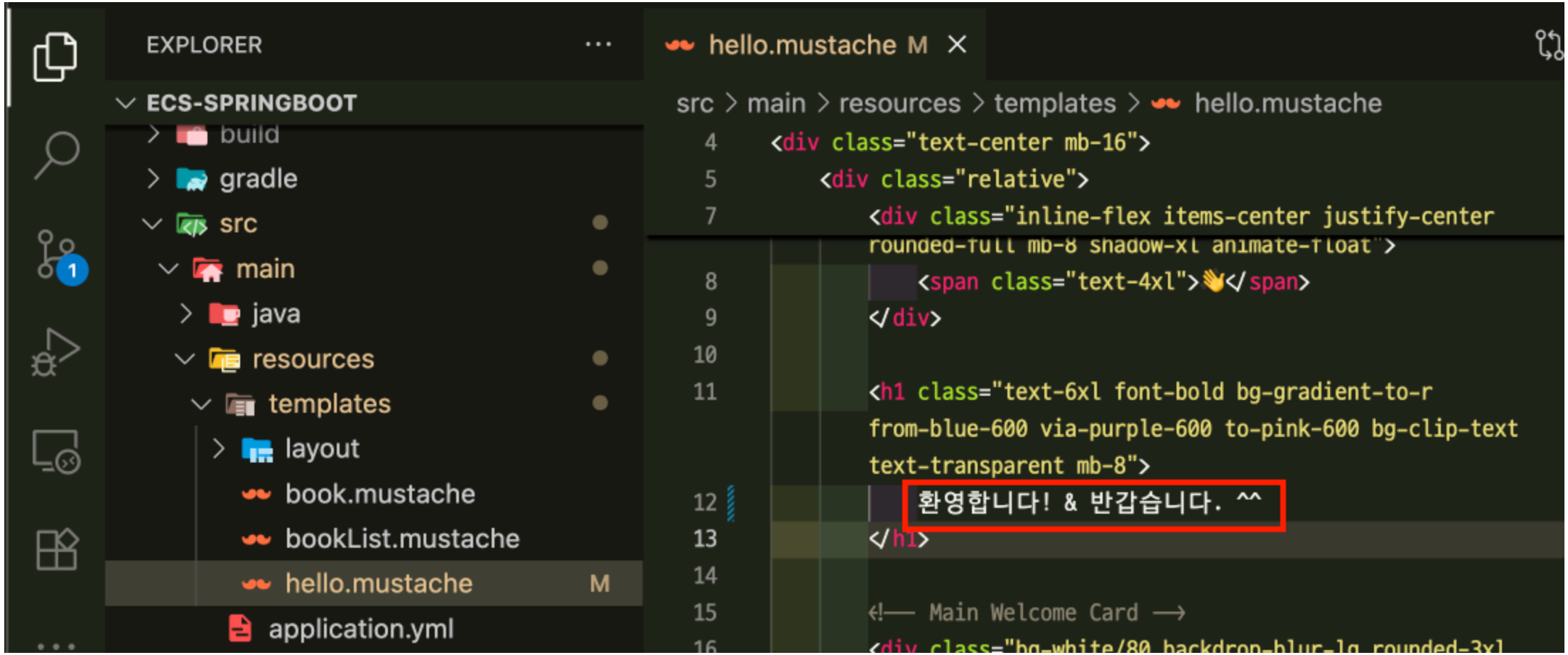
전체 아키텍처


CI/CD Pipeline
Amazon ECS
Spring Boot
GitHub Source
AWS Build
AWS Deploy
AWS Fargete
devices
docker
Amazon ECR

코드 수정 및 Github 반영

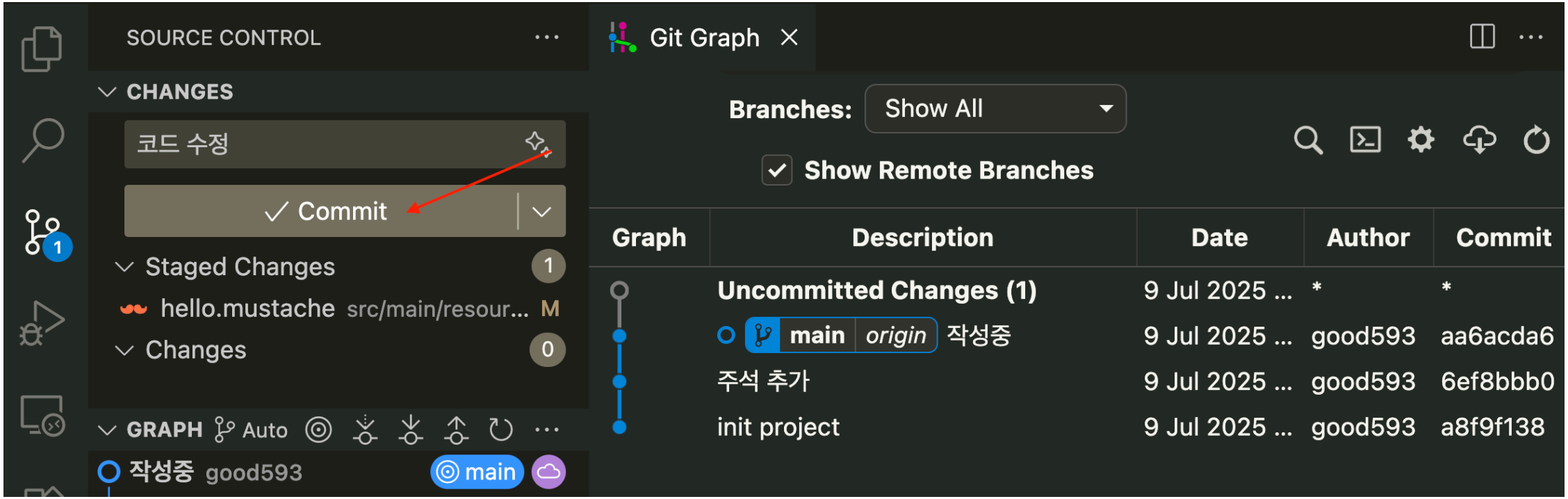
단계1: 코드 수정


EXPLORER
ECS-SPRINGBOOT
build
gradle
src
main
java
resources
templates
layout
book.mustache
bookList.mustache
hello.mustache
M
application.yml
hello.mustache M
src > main > resources > templates > hello.mustache

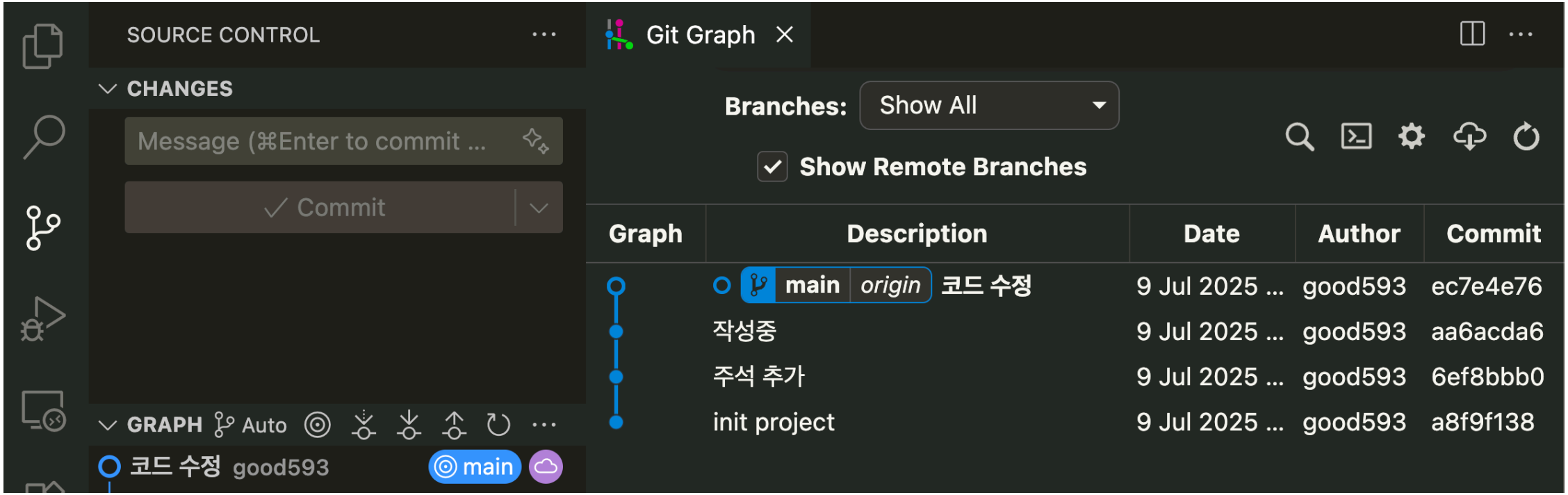
<div class="inline-flex items-center justify-center
rounded-full mb-8 shadow-xl animate-float">
👋

<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">
환영합니다! & 반갑습니다. ^^
</h1>
<!-- Main Welcome Card -->
<div class="bg-white/80 backdrop-blur-lg rounded-3xl

단계2: Commit


SOURCE CONTROL
CHANGES
코드 수정
Commit
Staged Changes 1
hello.mustache src/main/resour... M
Changes 0
GRAPH Auto
작성중 good593 main
Git Graph
Branches: Show All
Show Remote Branches
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



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

Go to file t

good593 코드 수정

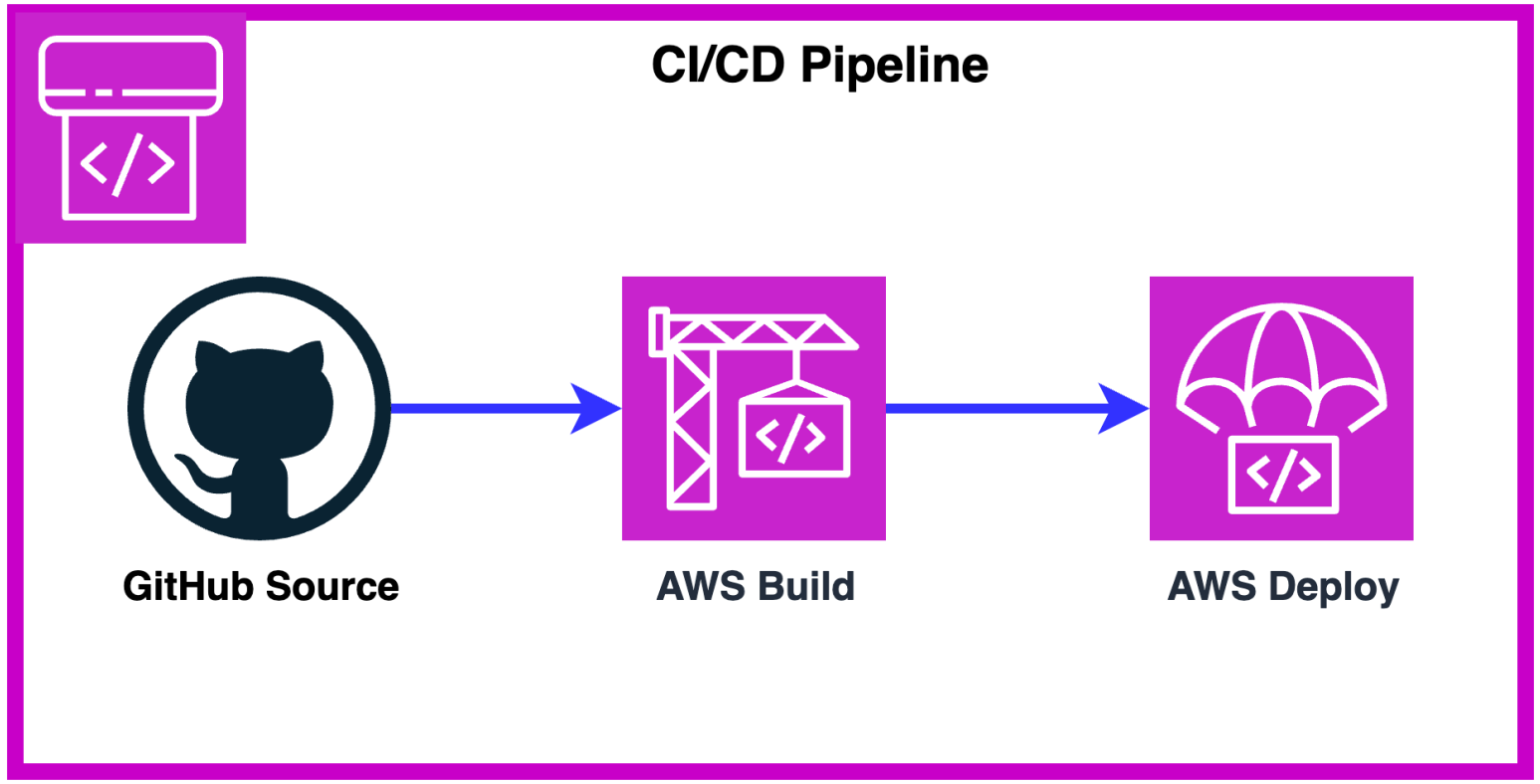
ec7e4e7 · 11 minutes ago History

Code Blame 111 lines (99 loc) · 6.35 KB Raw

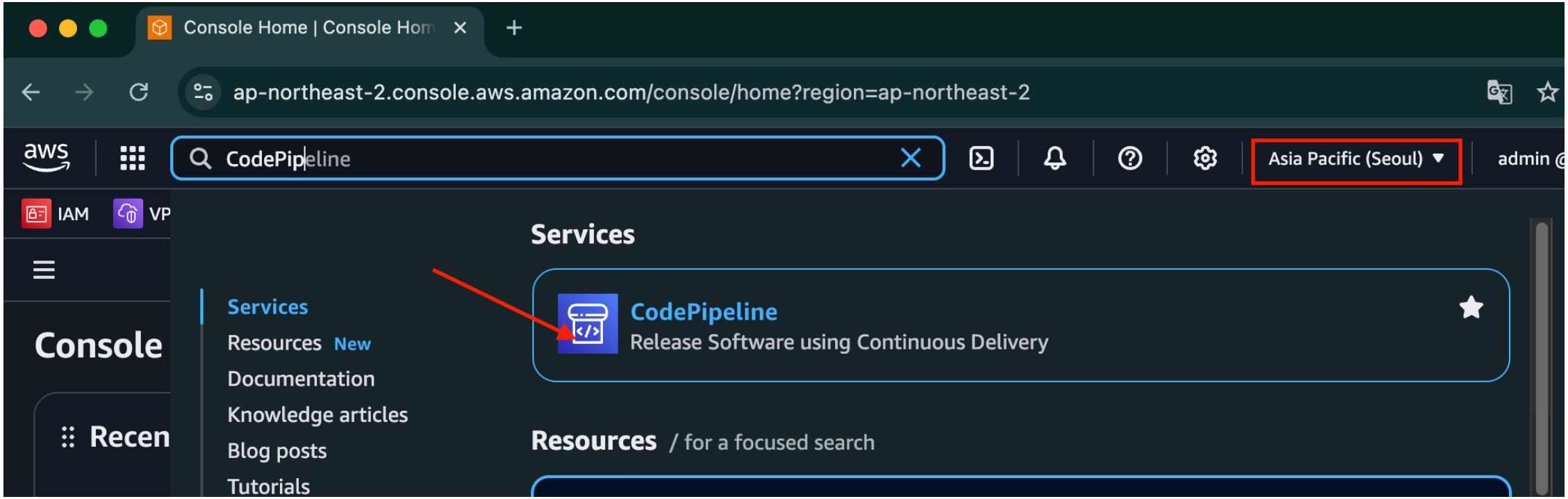
```
1   {{> layout/header}}
2
3   <!-- Welcome Hero Section -->
4   <div class="text-center mb-16">
5       <div class="relative">
6           <!-- Animated Welcome -->
7           <div class="inline-flex items-center justify-center w-24 h-24 bg-gradient-to-r from-blue-500 to-purple-600 rounded-full 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

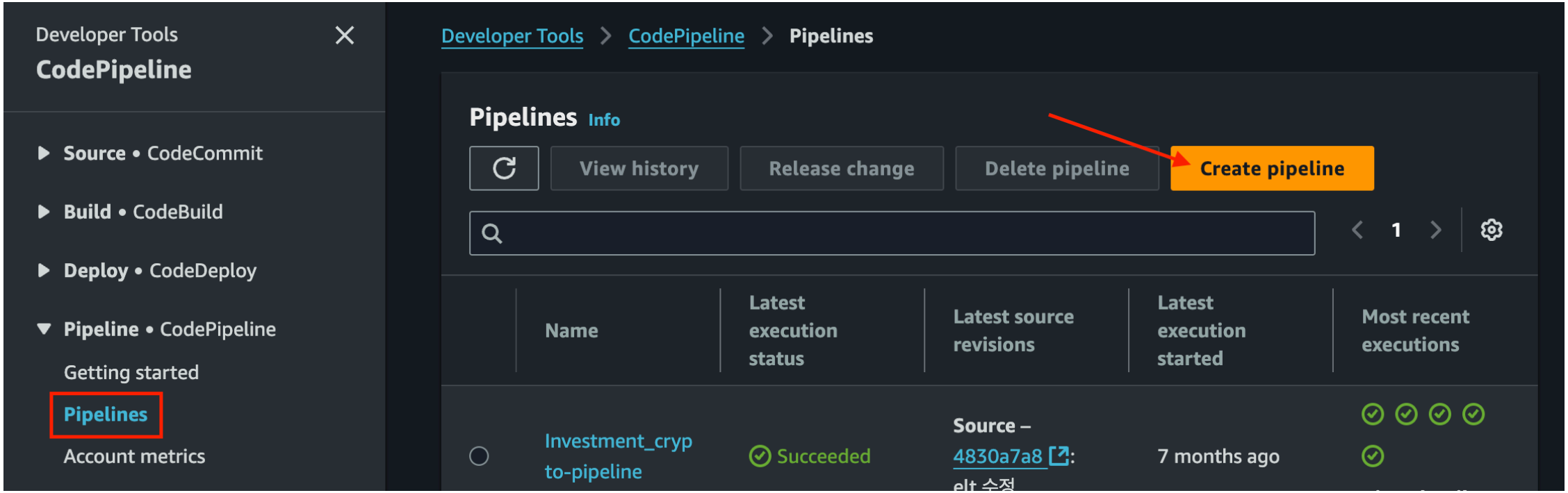
- AWS CodePipeline은 AWS에서 제공하는 지속적 통합(CI) 및 지속적 배포(CD) 서비스를 위한 자동화 파이프라인 도구입니다.

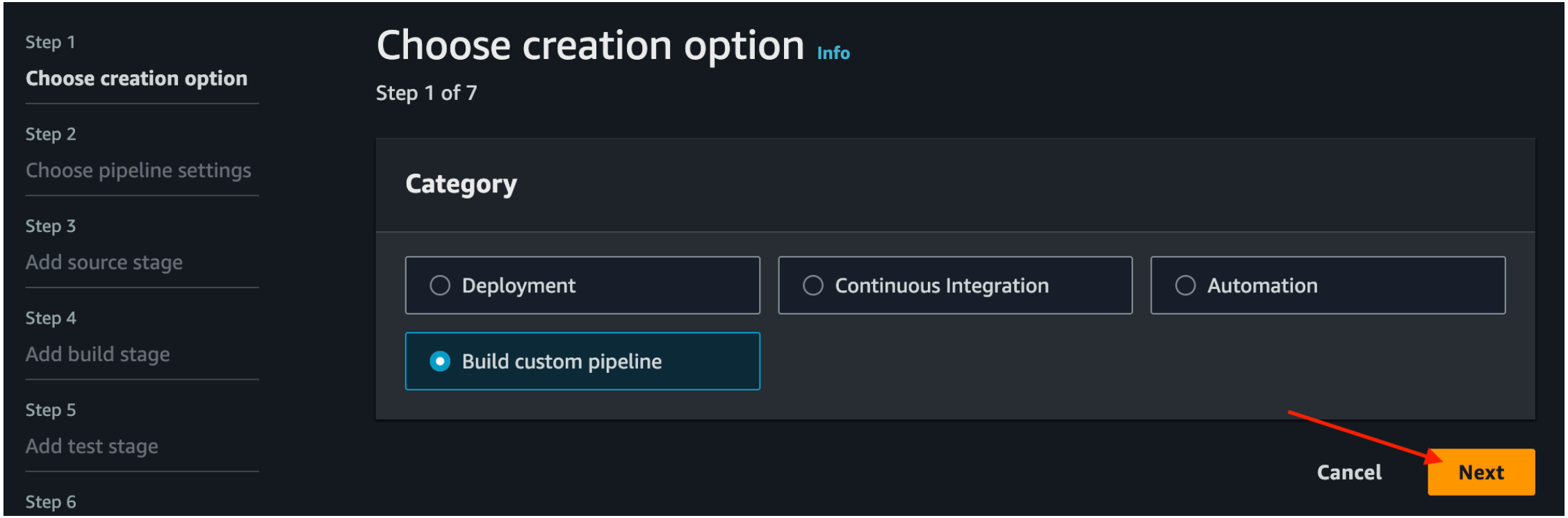


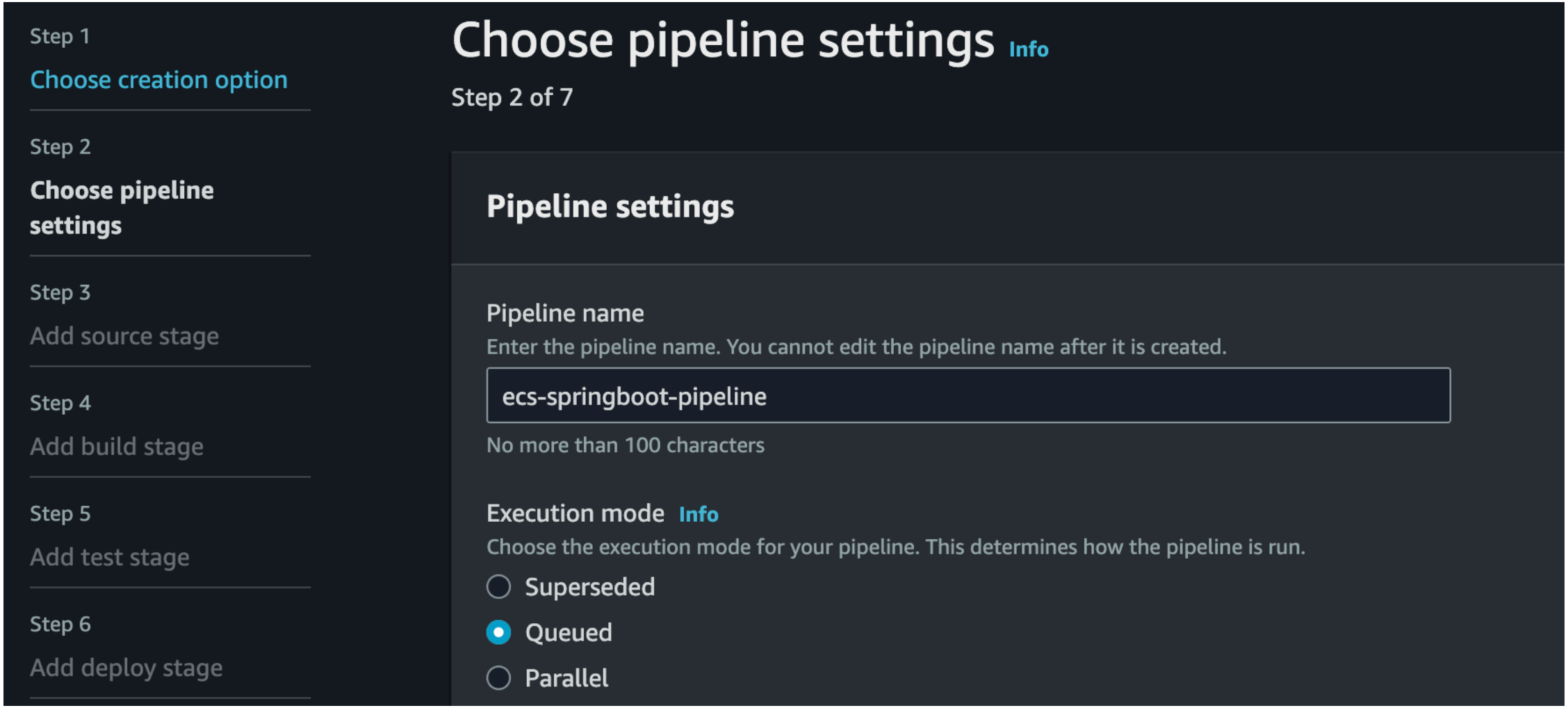
단계1: CodePipeline 접속

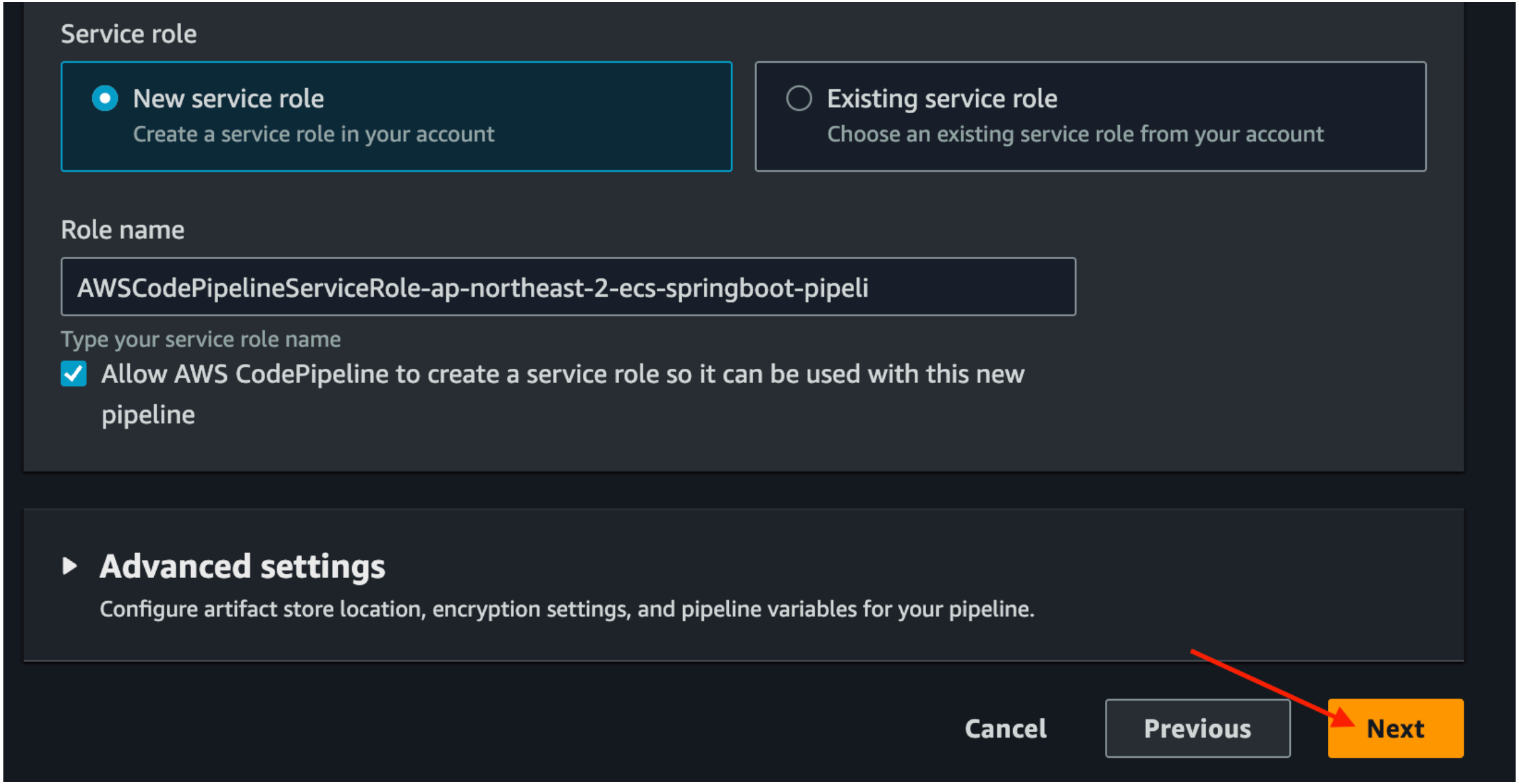


단계2: Create pipeline

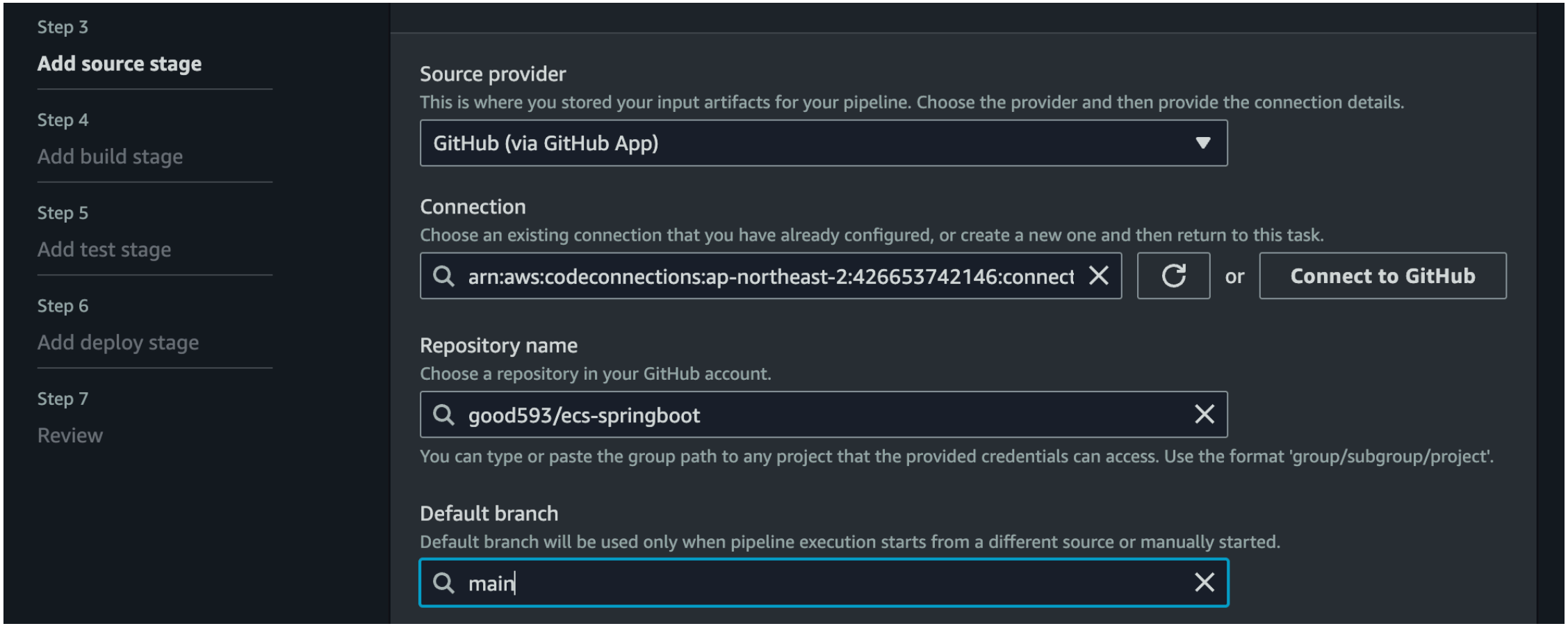


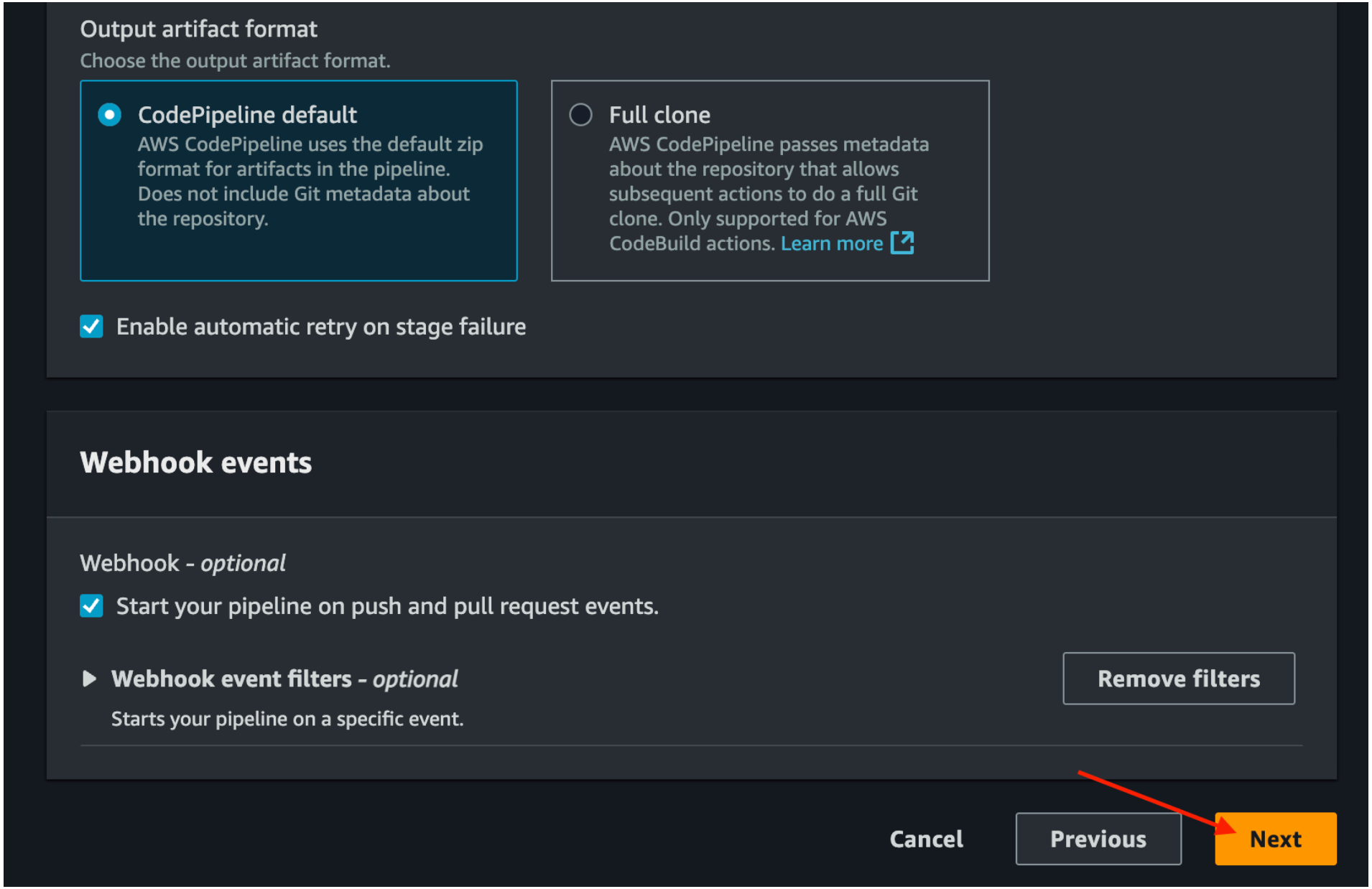

Step 1
Choose creation option
Step 2
Choose pipeline settings
Step 3
Add source stage
Step 4
Add build stage
Step 5
Add test stage
Step 6
Choose creation option Info
Step 1 of 7
Category
Deployment
Continuous Integration
Automation
Build custom pipeline
Cancel
Next


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.
ecs-springboot-pipeline
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

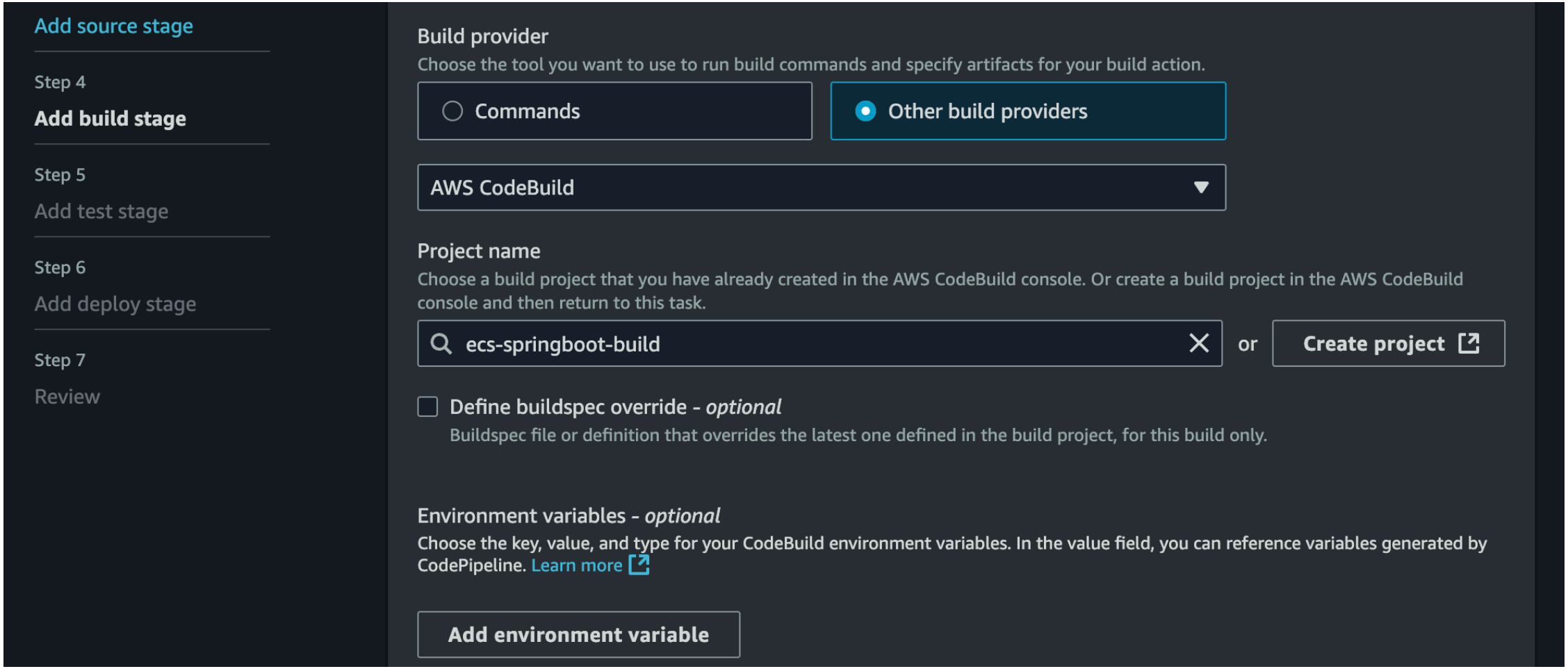

Service role
New service role
Create a service role in your account
Existing service role
Choose an existing service role from your account
Role name
AWSCodePipelineServiceRole-ap-northeast-2-ecs-springboot-pipeli
Type your service role name
Allow AWS CodePipeline to create a service role so it can be used with this new pipeline
Advanced settings
Configure artifact store location, encryption settings, and pipeline variables for your pipeline.
Cancel
Previous
Next

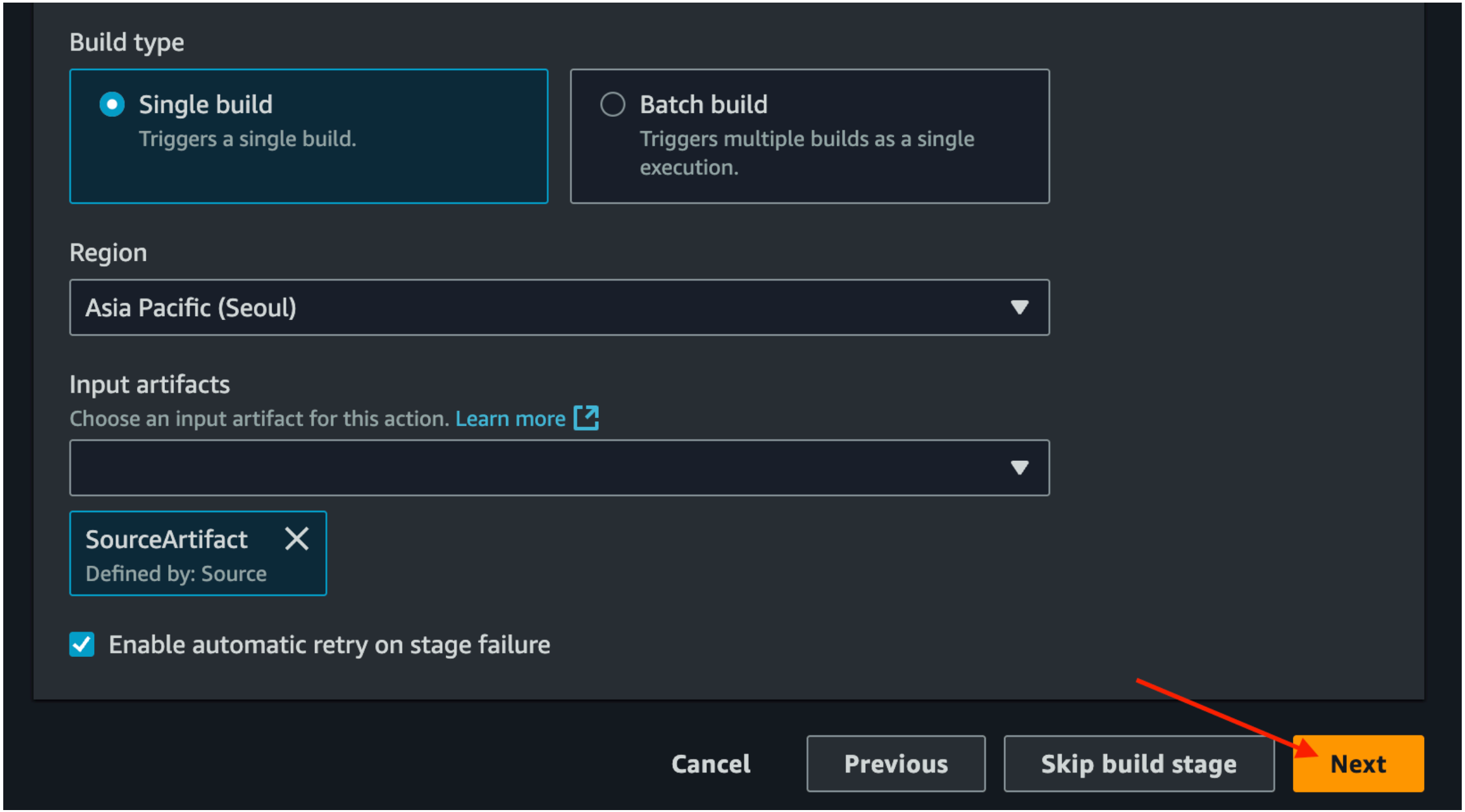
단계3: Source(Github)



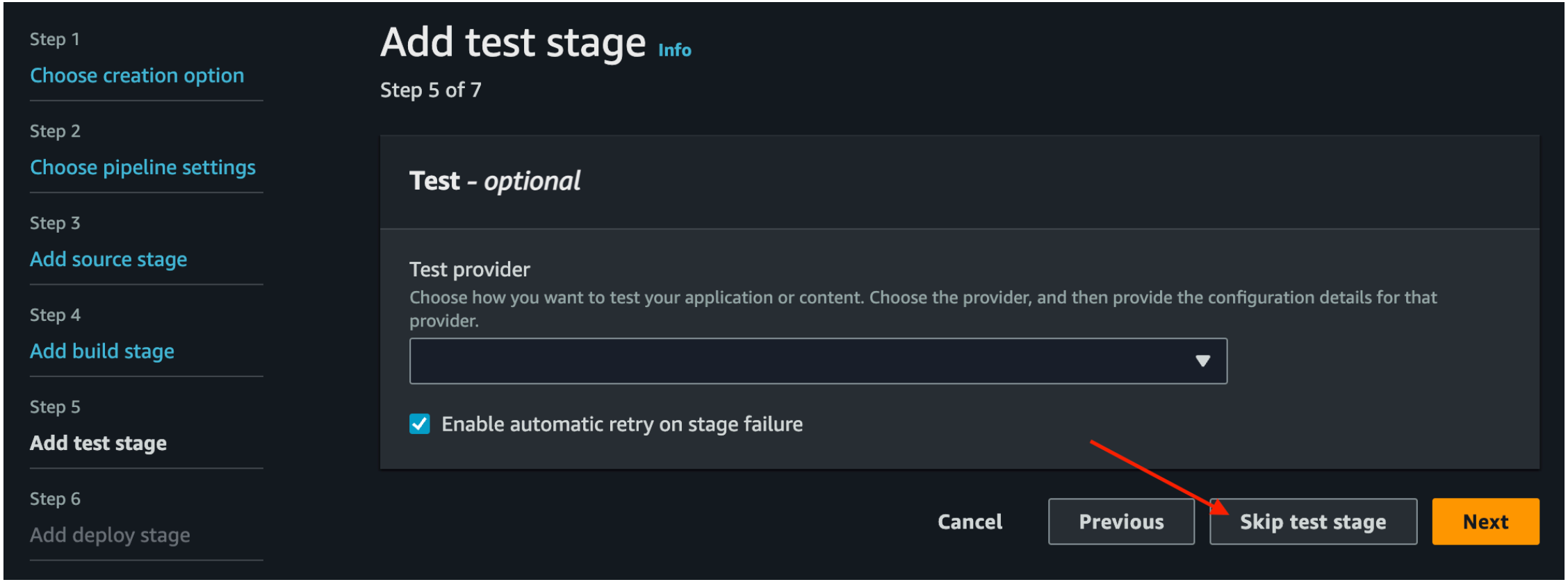

Output artifact format
Choose the output artifact format.
CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.
Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions 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

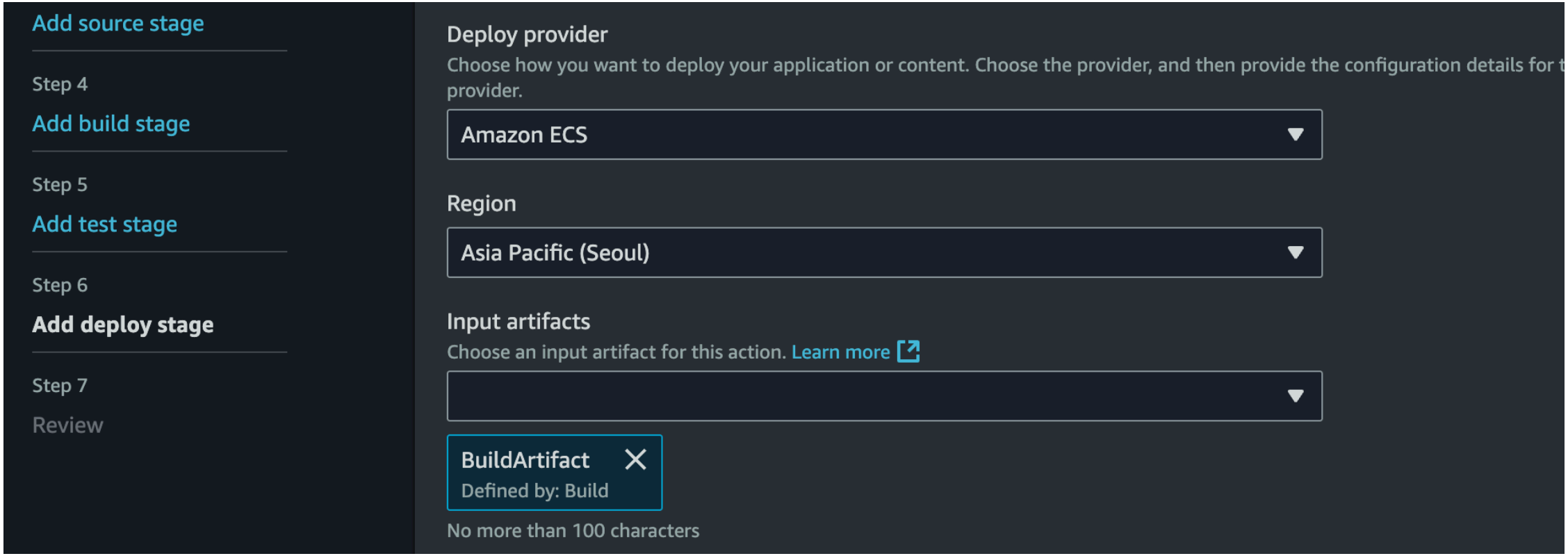

Add source stage
Step 4
Add build stage
Step 5
Add test stage
Step 6
Add deploy stage
Step 7
Review
Build provider
Choose the tool you want to use to run build commands and specify artifacts for your build action.
Commands
Other build providers
AWS CodeBuild
Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
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

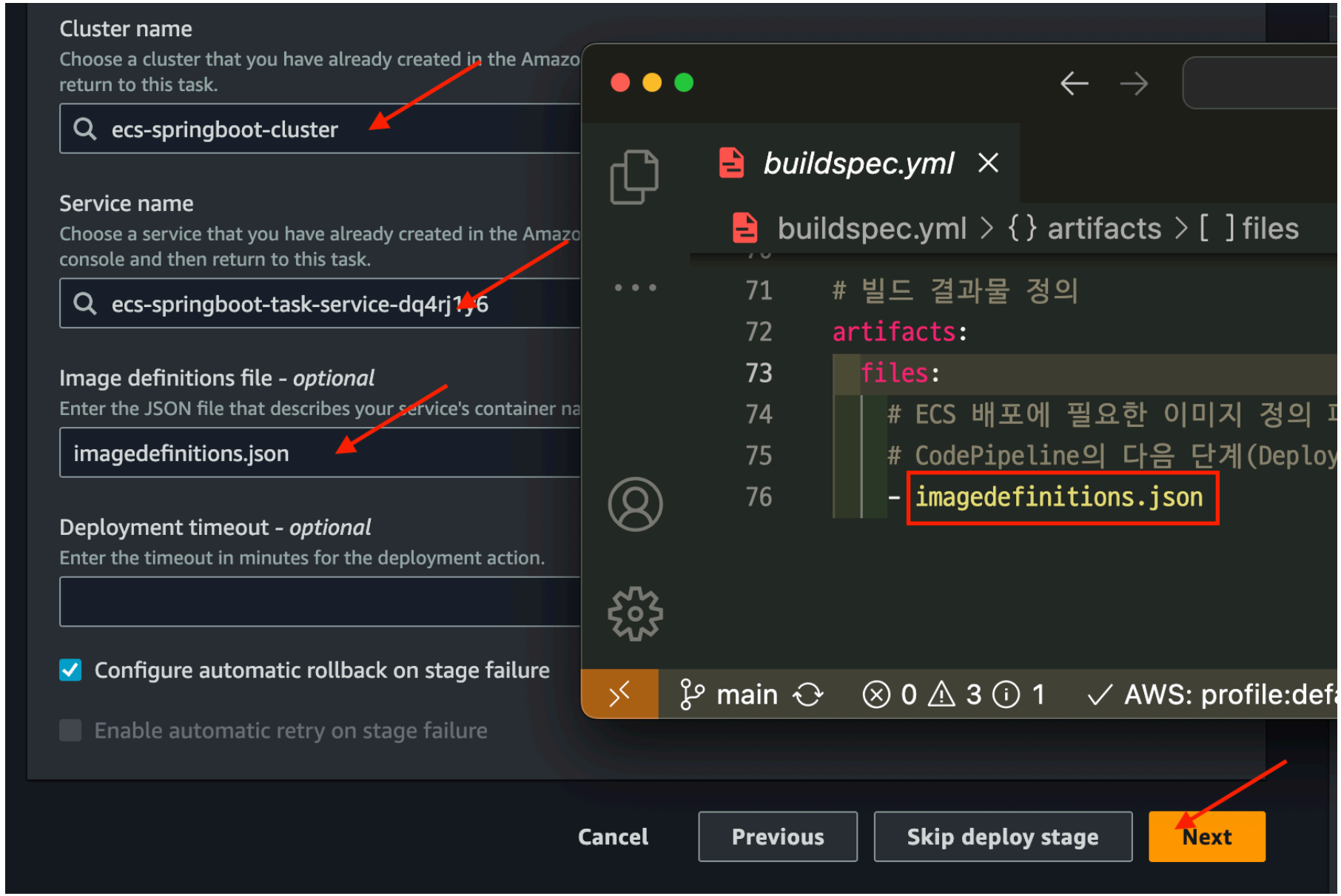

Build type
Single build
Triggers a single build.
Batch build
Triggers multiple builds as a single execution.
Region
Asia Pacific (Seoul)
Input artifacts
Choose an input artifact for this action. Learn more
SourceArtifact
Defined by: Source
Enable automatic retry on stage failure
Cancel
Previous
Skip build stage
Next

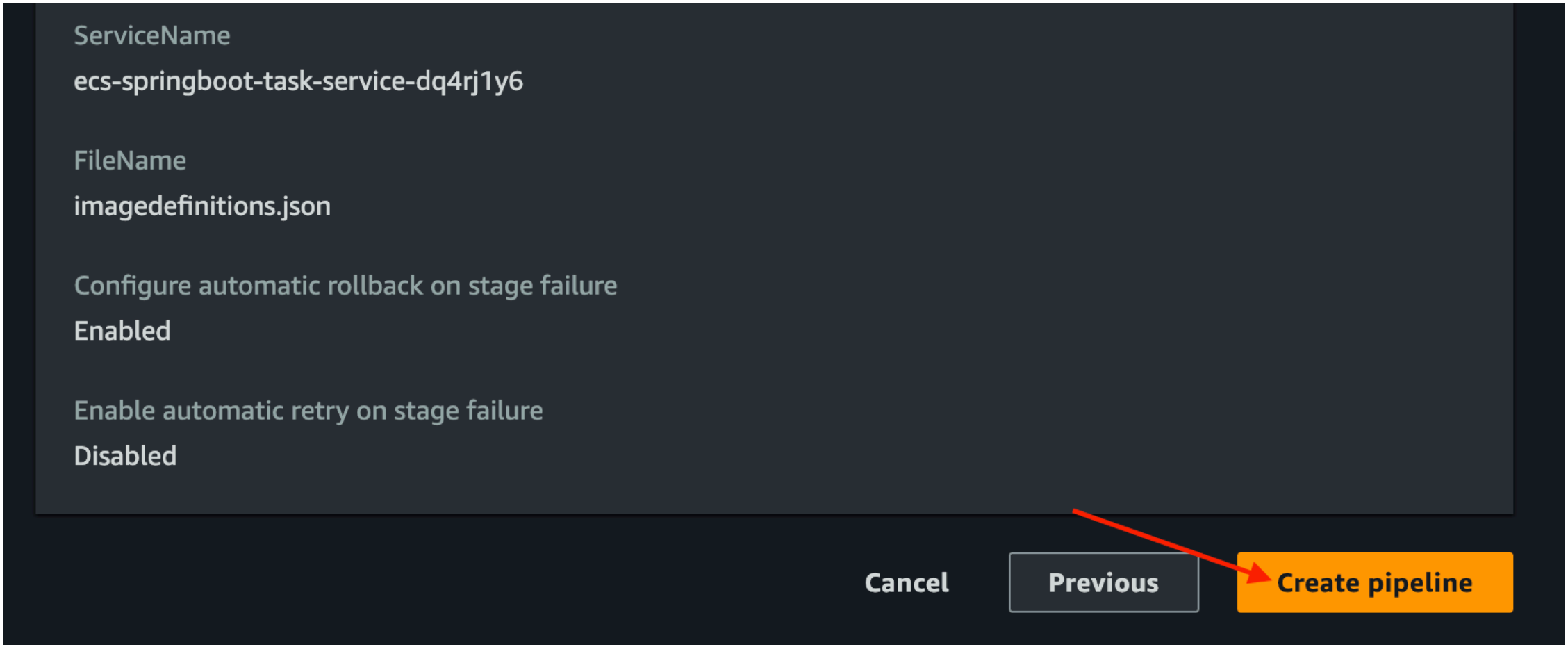
단계5: Test


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

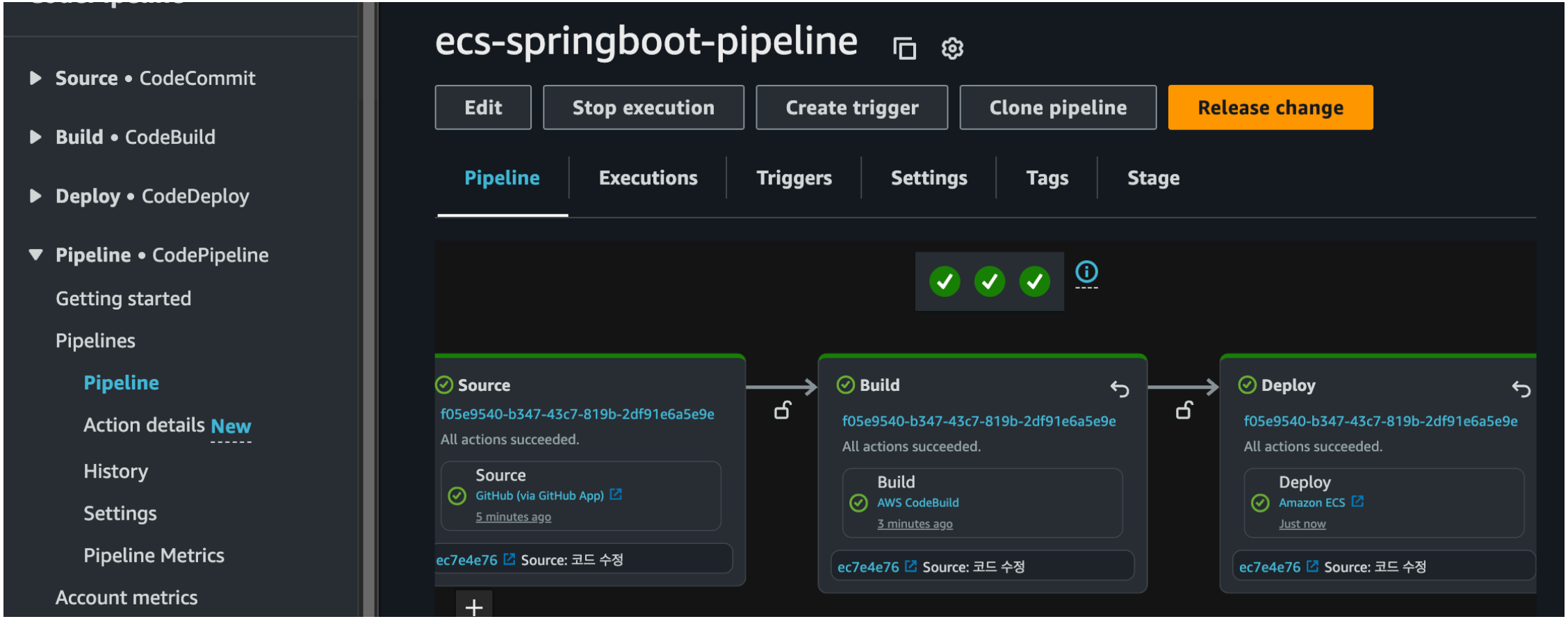
단계6: Deploy


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 t
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 Amazo
return to this task.
ecs-springboot-cluster
Service name
Choose a service that you have already created in the Amazo
console and then return to this task.
ecs-springboot-task-service-dq4rj1y6
Image definitions file - optional
Enter the JSON file that describes your service's container na
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
75 # CodePipeline의 다음 단계(Deploy
76 - imagedefinitions.json
main
0 3 1
AWS: profile:def
Cancel
Previous
Skip deploy stage
Next

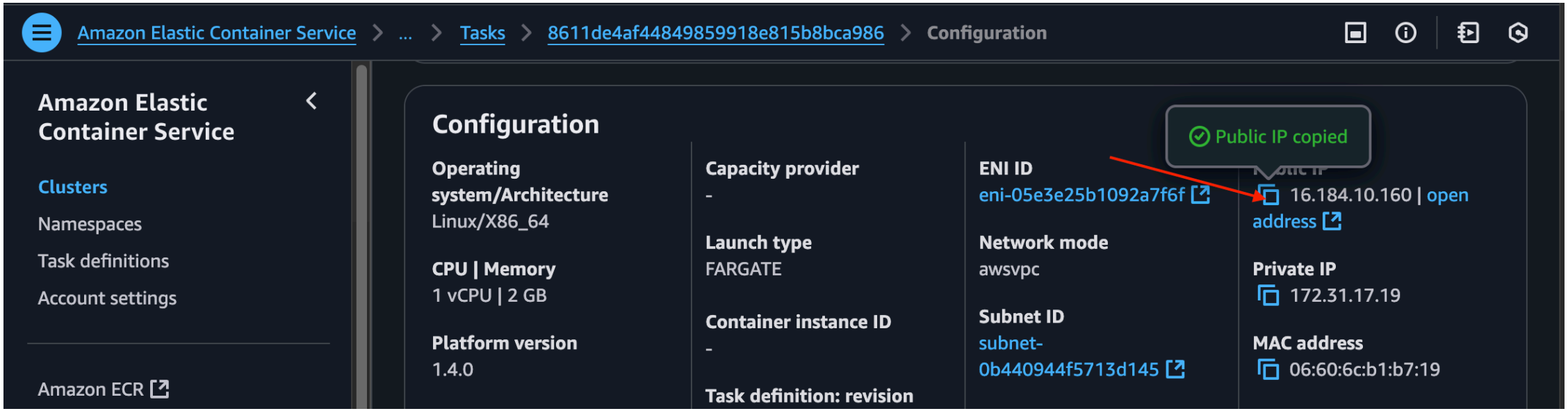

ServiceName
ecs-springboot-task-service-dq4rj1y6
FileName
imagedefinitions.json
Configure automatic rollback on stage failure
Enabled
Enable automatic retry on stage failure
Disabled
Cancel
Previous
Create pipeline

단계7: 성공


Source • CodeCommit
Build • CodeBuild
Deploy • CodeDeploy
Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
Action details New
History
Settings
Pipeline Metrics
Account metrics
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

변경된 내용 확인

단계1: ECS Public IP 복사



단계2: 확인

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