CodePipeline

!!!선행!!!

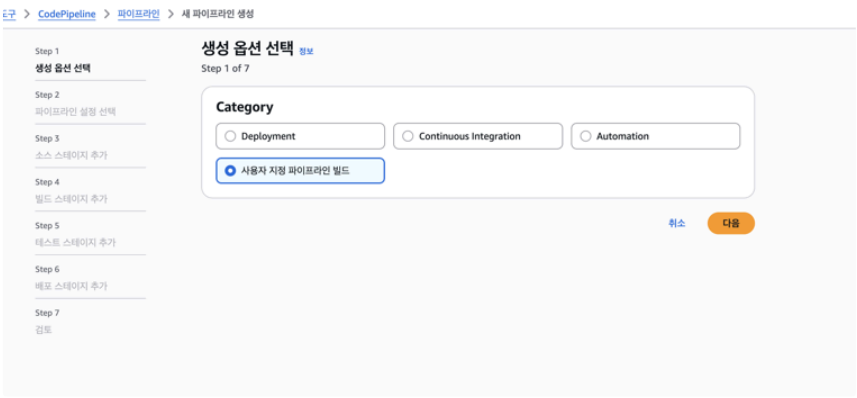
- CodeBuild가 먼저 설정되어야한다.

code commit에 코드를 push하고 자동 배포가 되려면 code pipeline이 설정되어야한다.

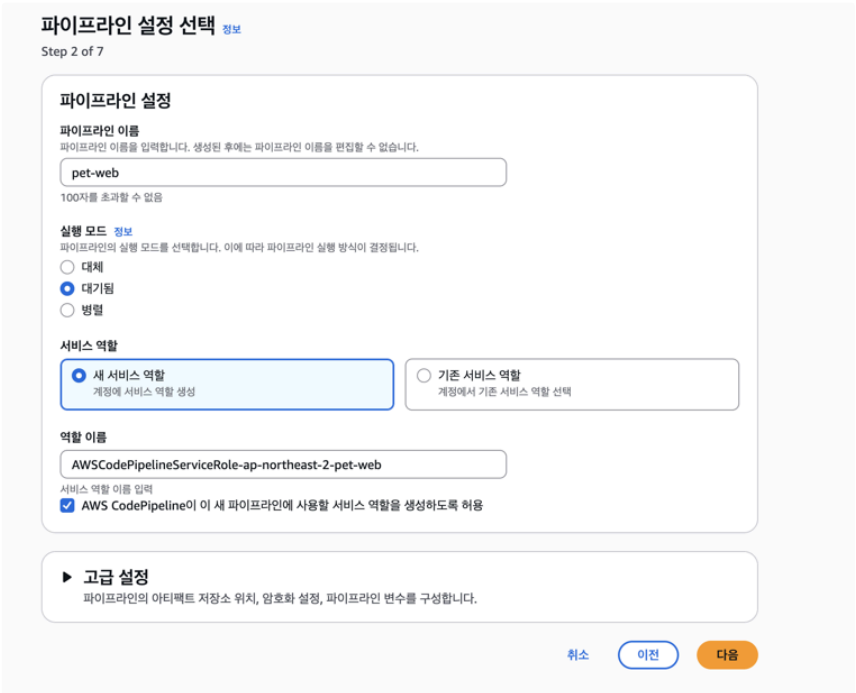
https://ap-northeast-2.console.aws.amazon.com/codesuite/codepipeline/pipelines?region=ap-northeast-2&pipelines-meta=eyJmIjp7InRleHQiOiIifSwicyI6eyJwcm9wZXJ0eSI6InVwZGF0ZWQiLCJkaXJlY3Rpb24iOi0xfSwibiI6MzAsImkiOjB9

생성 옵션 선택

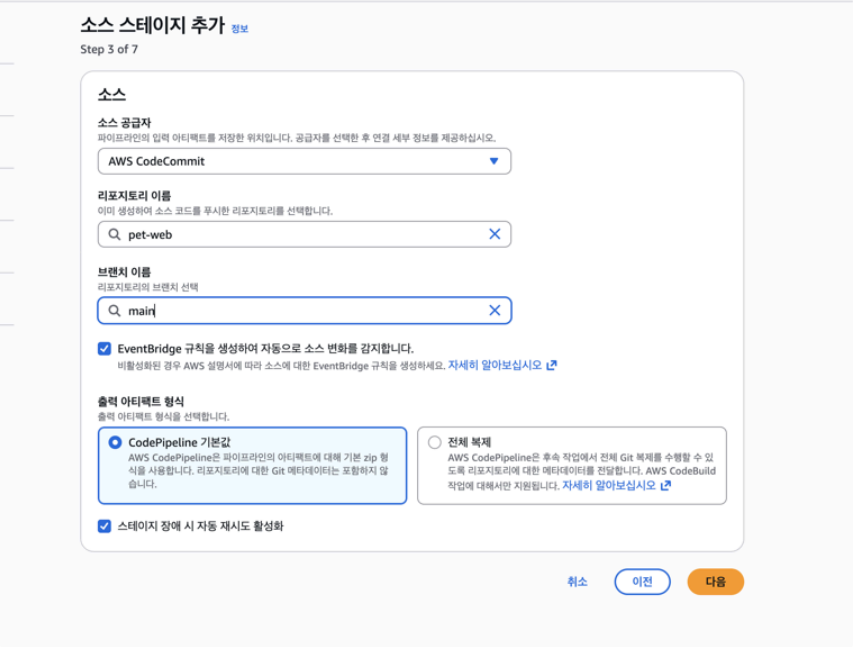
- deployment는 github gitlab등 다른 사이트의 코드를 가져온다 aws codeCommit은 사용자 지정파이프라인 빌드로 간다.



파이프라인 설정

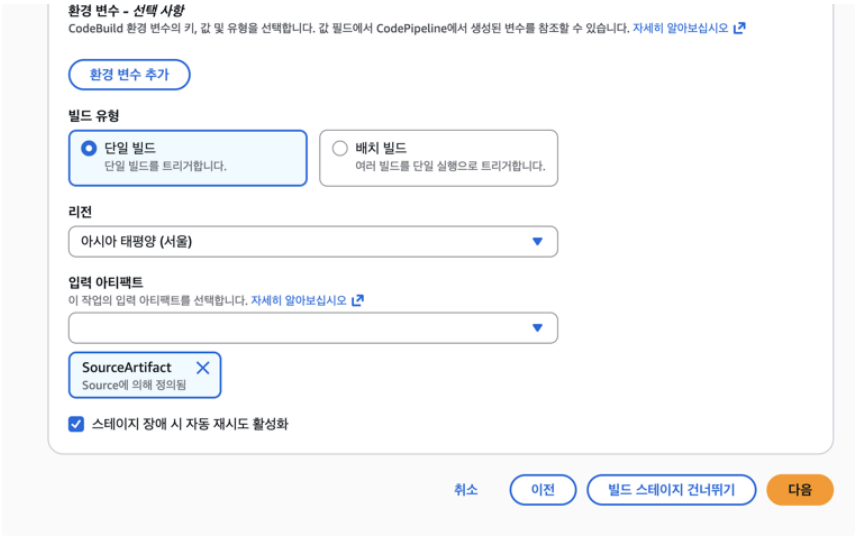


소스 스테이지 추가

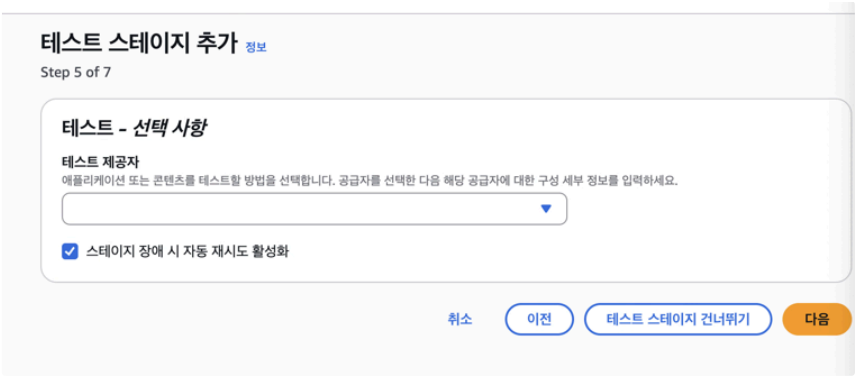


빌드 스테이지 추가

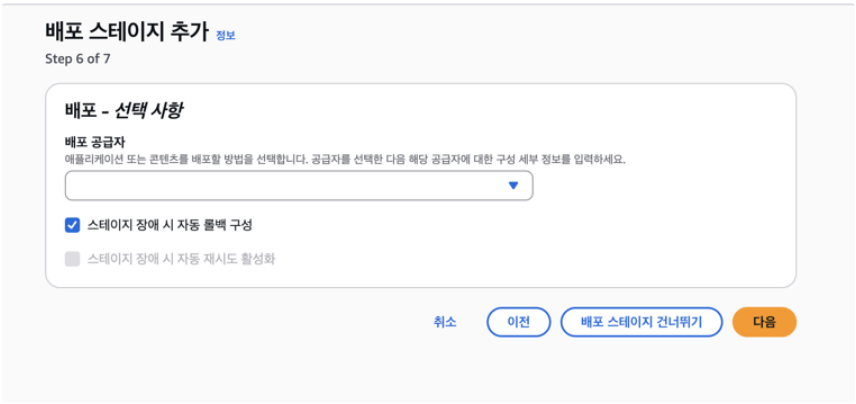
- CodeBuild를 먼저 설정하였으면 프로젝트 이름에 프로젝트가 뜬다. 그리고 프로젝트에 코드에 있는 buildspec.yml을 사용한다.



테스트 스테이지 건너뛰기 클릭



배포 스테이지 건너뛰기 클릭



파이프라인 생성 클릭

ecr로 ecs 배포 파이프라인

같이 설정하시면 됩니다.

1. CodePipeline + ECR 소스 액션 사용

파이프라인 구성:

- **소스 스테이지**: ECR 리포지토리를 소스로 설정
- **배포 스테이지**: ECS 서비스에 배포

파이프라인 생성 단계:

1단계: CodePipeline 생성

- AWS 콘솔에서 CodePipeline 서비스로 이동
- "파이프라인 생성" 선택

2단계: 소스 스테이지 설정

- 소스 공급자: **Amazon ECR** 선택
- 리포지토리 이름: 개발팀이 푸시하는 ECR 리포지토리 이름
- 이미지 태그: `latest` 또는 특정 태그 패턴

3단계: 빌드 스테이지 (선택사항)

- 이미지가 이미 준비되어 있으므로 빌드 스테이지는 생략 가능

4단계: 배포 스테이지 설정

- 배포 공급자: **Amazon ECS** 또는 **Amazon ECS (Blue/Green)**
- 클러스터 이름: 대상 ECS 클러스터
- 서비스 이름: 대상 ECS 서비스
- 이미지 정의 파일: `imageDetail.json` **(ECR 소스에서 자동 생성)**

2. EventBridge 규칙 자동 생성

CodePipeline을 콘솔에서 생성하면 **EventBridge 규칙이 자동으로 생성**됩니다:

```
{
    "detail-type": ["ECR Image Action"],
    "source": ["aws.ecr"],
    "detail": {
        "action-type": ["PUSH"],
        "image-tag": ["latest"],
        "repository-name": ["your-repo-name"],
        "result": ["SUCCESS"]
    }
}
```

3. 동작 방식

1. **개발팀이 ECR에 이미지 푸시**

   `docker push your-account.dkr.ecr.region.amazonaws.com/your-repo:latest`
2. **EventBridge가 ECR 푸시 이벤트 감지**
3. **CodePipeline 자동 트리거**
4. **ECS 서비스 자동 업데이트**
   - 새로운 태스크 정의 리비전 생성
   - 롤링 업데이트 또는 Blue/Green 배포 실행

4. 필요한 IAM 권한

CodePipeline 서비스 역할:

- ECR 이미지 읽기 권한
- ECS 서비스 업데이트 권한
- EventBridge 규칙 실행 권한

EventBridge 역할:

- CodePipeline 실행 권한

5. 추가 고려사항

이미지 태그 전략:

- `latest` 태그 사용 시: 매번 같은 태그로 덮어쓰기
- 버전 태그 사용 시: `v1.0.1`, `build-123` 등 고유 태그

배포 전략:

- **롤링 업데이트**: 기본 방식, 빠른 배포
- **Blue/Green**: 무중단 배포, 빠른 롤백 가능

이렇게 설정하면 개발팀이 ECR에 이미지를 푸시하는 즉시 자동으로 ECS 서비스가 업데이트됩니다. 추가로 궁금한 점이나 구체적인 설정 방법이 필요하시면 말씀해 주세요!

▼ Sources

```
version: 0.2
phases:
  pre_build:
    commands:
      - echo "Starting imagedefinitions.json generation..."
      - aws --version
  build:
    commands:
      - echo "Creating imagedefinitions.json file..."
      - echo '[{"name":"pet-web-container","imageUri":"132334511678.dkr
  post_build:
    commands:
      - echo "Verifying imagedefinitions.json:"
      - cat imagedefinitions.json
      - echo "File size:"
      - wc -c imagedefinitions.json
artifacts:
  files:
    - imagedefinitions.json
```