

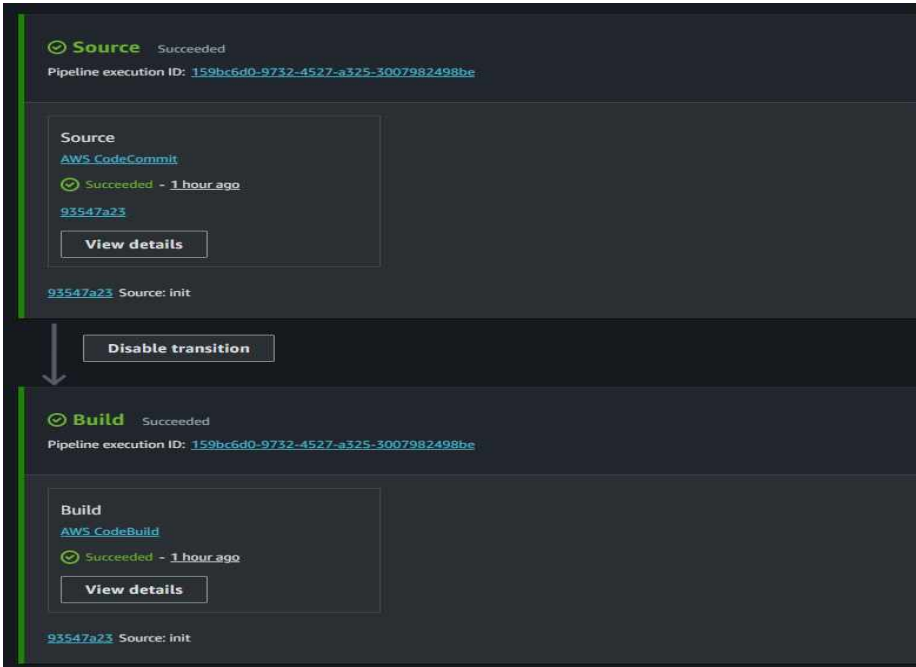
2024 2과제 전국기능경기대회 채점기준

1. 채점상의 유의사항	직 종 명	클라우드컴퓨팅
※ 다음 사항을 유의하여 채점하시오.		
1) AWS의 지역은 us-west-1을 사용합니다.		
2) 웹페이지 접근은 크롬이나 파이어폭스를 이용합니다.		
3) 웹페이지에서 언어에 따라 문구가 다르게 보일 수 있습니다.		
4) shell에서의 명령어의 출력은 버전에 따라 조금 다를 수 있습니다.		
5) 문제지와 채점지에 있는 <> 는 변수입니다. 해당 부분을 변경해 입력합니다.		
6) 채점은 문항 순서대로 진행해야 합니다.		
7) 삭제된 채점자료는 되돌릴 수 없음으로 유의하여 진행하며, 이의신청까지 완료 이후 선수가 생성한 클라우드 리소스를 삭제합니다.		
8) 부분 점수가 있는 문항은 채점 항목에 부분 점수가 적혀져 있습니다.		
9) 부분 점수가 따로 없는 문항은 모두 맞아야 점수로 인정됩니다.		
10) 리소스의 정보를 읽어오는 채점항목은 기본적으로 스크립트 결과를 통해 채점을 진행하며, 만약 선수가 이의가 있다면 명령어를 직접 입력하여 확인해볼 수 있습니다.		
11) [] 기호는 채점에 영향을 주지 않습니다.		
12) 명령어 입력 Box 안의 명령줄은 한 줄 명령어입니다. 별도의 지시가 없으면 수정 없이 박스 안의 전체 내용을 복사하고 쉘에 붙여넣어 명령을 실행합니다.		
13) (예상 출력)은 바로 이전 (명령어 입력)의 예상 출력을 의미합니다.		
14) Bastion 서버에 SSH를 통해 접근합니다. (별도 명시가 없는 경우 모든 채점은 Bastion 서버에서 진행합니다.)		
15) Bastion 명령어 및 권한을 확인합니다. (awscli permission, jq, curl, awscli region)		
16) mark.sh 스크립트들을 /root/mark.sh에 다운로드 합니다.		
17) /root/mark 경로에서 스크립트를 실행합니다. 실행 결과를 기반으로 채점을 진행하되 선수가 이의를 제기할 경우 수동으로 채점을 진행할 수 있도록 합니다.		
18) 채점을 진행하는 Bastion 서버의 쉘을 초기 실행할 때 다음 명령어를 실행하여 환경 변수를 초기화합니다. (채점 스크립트로 진행 시 생략)		

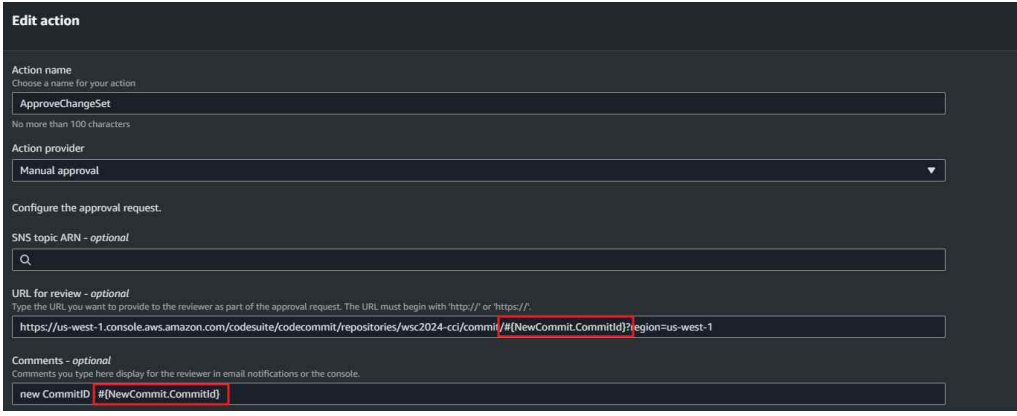
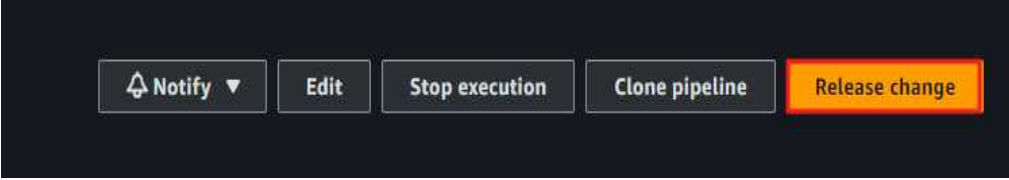

2) 채점방법 및 기준

과제 번호	일련 번호	주요항목	일련 번호	세부항목(채점방법)	배점
1과제	1	Version Control Continuous	1	CodeCommit 이름	0.10
			2	CodeCommit 구성 & CodePipeline 테스트	1.5
	2	Integration	1	CodeBuild 이름	0.10
	3	Continuous Delivery	1	CodeDeploy 이름	0.10
	4	CI/CD Pipeline	1	CodePipeline 이름	0.10
			2	Manual approval	1.5
	5	Application	1	배포 확인	0.35
	총점				3.75

3) 채점내용

순번	채점 항목	
1-1	1-1-A (명령어 입력)	aws codecommit get-repository --repository-name wsc2024-cci ₩ --query "repositoryMetadata.repositoryName"
	1-1-A <u>정확히 일치</u>	["wsc2024-cci"]
1-2	1-2-A (명령어 입력)	aws codecommit get-folder --repository-name wsc2024-cci ₩ --commit-specifier master --folder-path / --region us-west-1 grep Dockerfile
	1-2-A <u>Dockerfile이 없는지 확인</u>	출력된 결과에 Dockerfile이라는 문구가 없을 경우 이어서 진행하며, 만약 있을 경우 해당 항목은 0점 처리합니다.
	1-2-B (명령어 입력)	aws codepipeline start-pipeline-execution --name wsc2024-pipeline --region us-west-1
	1-2-B <u>Succeeded가 뗏는지 확인</u>	<p>위와 같은 명령어를 입력했을 경우에 pipeline이 실행이 됩니다. CodeCommit -> CodeBuild가 모두 성공이 되어야 합니다</p> 

순번	채점 항목	
2-1	2-1-A (명령어 입력)	aws codebuild batch-get-projects --names wsc2024-cbd --query "projects[*].name"
	2-1-A <u>정확히 일치</u>	["wsc2024-cbd"]
3-1	3-1-A (명령어 입력)	aws deploy get-application --application-name wsc2024-cdy \ --query "application.applicationName"
	3-1-A <u>정확히 일치</u>	["wsc2024-cdy"]
4-1	4-1-A (명령어 입력)	aws codepipeline get-pipeline --name wsc2024-pipeline --query "pipeline.name"
	4-1-A <u>정확히 일치</u>	["wsc2024-pipeline"]

순번	채점 항목	
4-2	4-2-A	https://us-west-1.console.aws.amazon.com/codesuite/codepipeline/pipelines/wsc2024-pipeline/view?region=us-west-1 으로 이동 합니다. 이동해서 Edit 부분을 클릭합니다. CodeBuild – CodeDeploy 사이 스테이지의 설정을 들어갑니다.
	4-2-A <u>정확히 일치</u>	<p>변수가 #{SourceVariables.CommitId} 으로 지정되어 있을 경우에 이 항목은 0점 처리가 되고, #{SourceVariables.CommitId} 가 아닌 다른 변수로 지정되어 있을 경우에 이어서 진행 합니다.</p> 
	4-2-B	code pipeline을 재시작을 했을 때 , 수동 승인 부분에서 View details을 누른 후에 URL for review에 있는 URL을 클릭 했을 때 커밋 페이지로 이동 되어야 합니다.
	4-2-B <u>커밋</u> <u>페이지로</u> <u>이동되는지</u> <u>확인</u>	 

순번	채점 항목	
5-1	5-1-A (명령어 입력)	elb_dns=\$(aws elbv2 describe-load-balancers --query "LoadBalancers[*].DNSName" --output text) curl \$elb_dns/healthcheck
	5-2-A <u>정확히 일치</u>	{"status": "200 OK"}