

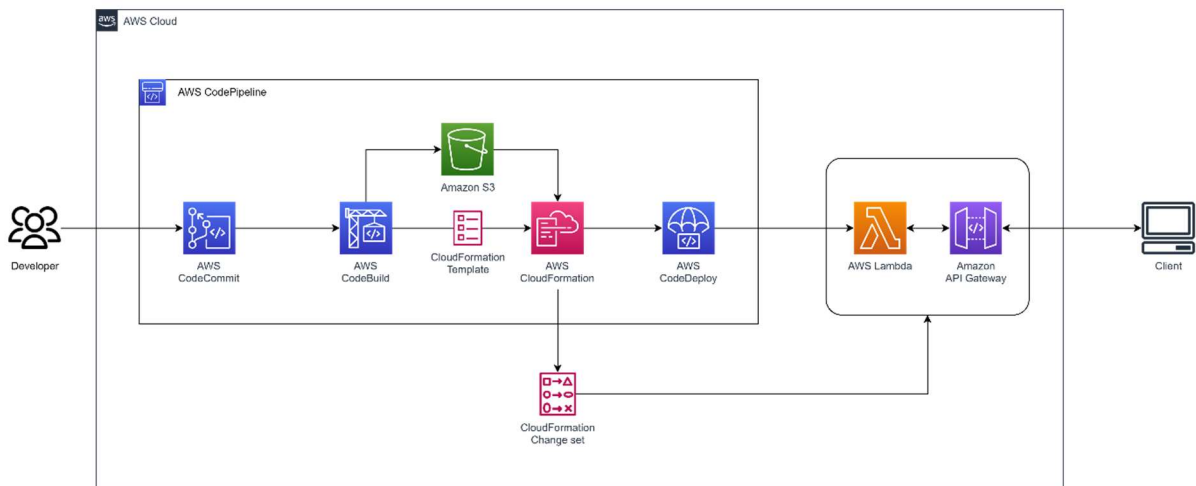
## 2022 클라우드컴퓨팅 기능경기대회 친선평가전

직 종 명	클라우드컴퓨팅	과 제 명	Automation	과제번호	제2과제
경기시간	4시간	비 번 호		심사위원 확 인	(인)

### 1. 요구사항

개발 효율을 위하여 CI/CD 파이프라인을 구성하고자 합니다. AWS 환경에서 개발하고 있기 때문에 모두 AWS solution을 활용하여 파이프라인을 구성할 예정입니다. AWS의 대표적인 IaC 서비스인 CloudFormation과 AWS SAM 오픈 소스 프레임워크를 활용해 서버리스 인프라 및 어플리케이션 배포 자동화 환경을 구성해야 합니다.

#### Architecture



## Software Stack

- AWS
  - AWS Lambda
  - Amazon API Gateway
  - Amazon S3
  - AWS KMS
  - AWS CloudFormation(AWS SAM)
  - AWS CodeCommit
  - AWS CodeBuild
  - AWS CodeDeploy
  - AWS CodePipeline
- 개발언어
  - Python

## 2. 선수 유의사항

- 1) 기계 및 공구 등의 사용 시 안전에 유의하시고, 필요 시 안전장비 및 복장 등을 착용하여 사고를 예방하여 주시기 바랍니다.
- 2) 작업 중 화상, 감전, 철과상 등 안전사고 예방에 유의하시고, 공구나 작업도구 사용 시 안전보호구 착용 등 안전수칙을 준수하시기 바랍니다.
- 3) 작업 중 공구의 사용에 주의하고, 안전수칙을 준수하여 사고를 예방하여 주시기 바랍니다.
- 4) 경기 시작 전 가벼운 스트레칭 등으로 긴장을 풀어주시고, 작업도구의 사용 시 안전에 주의하십시오.
- 5) 경기 중 완전하게 공개된 자료만 인터넷에서 열람하여 참고할 수 있습니다.  
(공개된 자료가 아니면, 실격으로 처리될 수 있습니다.)
- 6) 문제에 제시된 괄호박스 <>는 변수를 뜻함으로 선수가 적절히 변경하여 사용해야 합니다.
- 7) Region은 서울(ap-northeast-2)을 사용합니다.
- 8) 제공된 지급자료에 파일이 알맞게 있는지 확인합니다.
- 9) 문제에서 지정하지 않은 리소스라도 임의로 적절한 Name Tag를 부여해야 합니다. (단, Name Tag를 설정할 수 없는 Resource는 무관합니다.)
- 10) 문제에서 지정하지 않는 설정은 AWS Best Practice로 구성합니다.
- 11) 경기종료전까지 아키텍처 및 문제에서 제시한 리소스들이 모두 생성되어있어야 합니다.
- 12) 채점을 위해 AdministratorAccess 권한을 가지면서 Console Access, Programmatic Access가 가능한 IAM User를 생성하고, 해당 IAM User의 Account ID, User name, Password가 담긴 텍스트 파일을 제공해야 합니다.

### 3. Application

제공된 app.py 는 Python을 통해 개발된 어플리케이션입니다. app.py 는 람다 함수 생성 시에 사용되며, 채점 시 해당 파일을 수정할 수도 있으며, 람다 함수를 호출하여 응답 값을 체크해 정답유무를 측정합니다. 해당 파일을 선수 본인이 수정 하는 경우에 채점에 불이익을 받을 수 있습니다.

### 4. AWS CodeCommit

어플리케이션의 소스코드를 관리하기 위해 AWS에서 제공하는 CodeCommit 서비스를 사용합니다. CodeCommit Repository를 생성하고 제공된 app.py와 app\_test.py 파일을 업로드 합니다. 제공받은 app.py와 app\_test.py 는 Repository에 src 디렉토리를 생성하고 아래에 위치시킵니다.

- Repository Name : wsi-codecommit
- Default branch : main
- app.py & app\_test.py 위치 : src/app.py & src/app\_test.py

## 5. AWS CodeBuild

어플리케이션 소스코드를 테스트하고 빌드 하기 위해 AWS CodeBuild를 사용합니다. 지급된 app\_test.py 를 테스트하고 Application을 빌드한 결과물을 S3로 업로드합니다. 테스트 결과는 CodeBuild Report Group에 표시되어야 하며, 빌드된 결과물은 CloudFormation 템플릿 배포 시 사용되어야 합니다. Cloudwatch Log를 활성화하여 빌드 시작 후 진행상황 로그를 볼 수 있도록 합니다.

- CodeBuild Project Name : wsi-codebuild
- 추천 테스트 명령어 : `pytest --junit-xml ./junit.xml`
- 추천 빌드 명령어 : `sam build && sam package`

아래 정보를 가지고 Report Group을 생성합니다.

- Report Group Name : wsi-report-group
- Report Type : Test
- Export : disable

## 6. AWS CloudFormation

IaC를 활용하여 인프라를 구축하기 위해 CloudFormation을 사용합니다.

제시된 아키텍처 및 문제에 제시된 내용에 따라 적절한 템플릿을 구성해야 합니다. 템플릿은 CodeCommit Repository의 root경로에 업로드 되어야합니다.

Change Set명은 선수 본인이 지정합니다. 채점 시 템플릿을 수정할 수도 있으며, 수정된 내용에 따라 인프라가 자동으로 변경되어야 합니다.

- Stack Name : wsi-stack
- Template File Name : template.yaml

## 7. AWS CodePipeline

앞에서 생성한 CodeCommit을 Source로 CodeBuild를 Build로 CloudFormation을 Deploy로 가지는 Pipeline을 생성합니다. 각 Stage의 이름은 Source, Build, Deploy로 설정합니다. wsi-codecommit의 main branch에서 소스 코드가 변경되면 자동으로 빌드되고 배포되어야 합니다.

- Pipeline Name : wsi-pipeline

### 7.1 Source Stage 구성

- Stage Name : Source
- Source Stage : wsi-codecommit의 main branch

### 7.2 Build Stage 구성

- Stage Name : Build
- Build Stage : wsi-codebuild

### 7.3 Deploy Stage 구성

Deploy Stage는 세 가지 Action으로 구성되어야 합니다. 첫 번째는 CloudFormation 템플릿을 가지고 Change Set을 생성해야 하고, 두 번째는 어떤 부분이 변경되었는지 확인하기 위해 리뷰 및 승인 작업을 구성해야 합니다. 세 번째는 승인되었을 경우 생성된 Change Set을 실행시켜 변경된 내용을 배포해야 합니다.

- Stage Name : Deploy

### 7.3.1 Change Set 생성

- Action Name : CreateChangeSet
- Action Provider : AWS CloudFormation
- Stack Name : wsi-stack

### 7.3.2 리뷰 및 승인 작업 구성

- Action Name : ApproveChangeSet
- Action Provider : Manual approval
- Comments : Confirm this commit is ready for deployment: <가장 최근의 Commit ID>
- URL for review : CodeCommit의 커밋 중 가장 최근 커밋의 URL

### 7.3.3 Change Set 실행

- Action Name : ExecuteChangeSet
- Action Provider : AWS CloudFormation
- Stack Name : wsi-stack

## 8. Amazon S3

아래 정보를 이용하여 S3 버킷을 생성합니다. Build Stage에서 어플리케이션을 빌드한 결과물을 S3에 업로드 합니다. 어플리케이션을 빌드한 결과물은 KMS의 Amazon S3-managed key를 통해 암호화되어 S3에 업로드되어야 합니다.

- S3 Bucket Name : wsi-<임의의 영문 4자리>-<비번호>-cfm-build

## 9. AWS Lambda

서버리스 환경의 어플리케이션을 운영을 위하여 람다를 사용합니다. 람다 함수는 CloudFormation에 의해 관리 되어야 하며, 람다 함수 생성 시 지급 된 app.py가 배포 되어야 합니다. 람다 함수는 별칭(Alias)과 버전을 사용해 버전을 관리하고, 소스 코드 변경 시 CodeDeploy에 의해 Canary배포가 진행되어야 하며, 배포가 완료되면 람다 함수의 새 버전이 생성되고 별칭이 새 버전으로 지정되어야 합니다.

- Lambda Function Name : wsi-api-function
- Runtime : python3.8
- Alias : live
- Deployment : Canary10Percent15Minutes



## 10. Amazon API Gateway

외부에서 람다를 호출하기 위해 API Gateway를 생성합니다. 생성된 API의 Endpoint의 어떠한 경로로든 요청을 보냈을 때 람다 함수가 호출되어야 하며, 해당 API는 CloudFormation에 의해 관리되어야 합니다. 채점 시 GET 메소드로 접근하여 채점합니다.