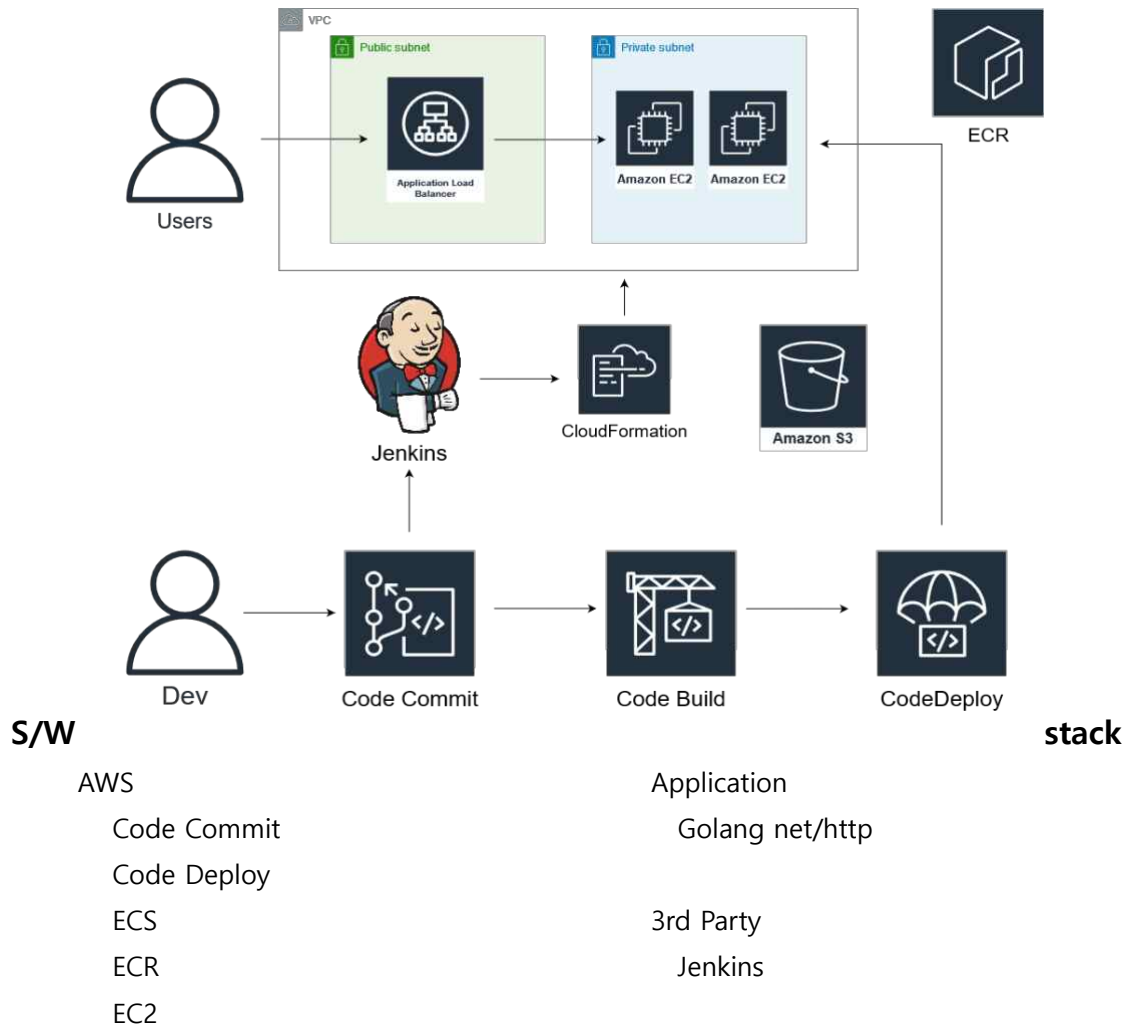


2023년 클라우드컴퓨팅 친선경기대회

직 종 명	클라우드컴퓨팅	과 제 명	Automation	과제 번호	제 2과제
경기시간	4시간	비 번 호		심사위원 확 인	(인)

1. 요구사항

당신은 현재 Undrop에서 클라우드 엔지니어로 근무 중입니다. Undrop은 클라우드를 이용해서 파일을 업로드하고 저장하는 회사입니다. 하지만 해당 애플리케이션을 개발하기 까지에는 많은 배포와 빌드가 있을 것으로 예상되어 CI/CD Pipeline을 구축하고자 합니다. CI/CD Pipeline을 작동하고 앱에서 업로드하면 AWS S3 Bucket으로 저장할 수 있게 구축하도록 합니다.



2. 선수 유의사항

※ 다음 유의사항을 고려하여 요구사항을 완성하시오.

- 1) 지정되지 않은 옵션 또는 설정은 기본 값 또는 적절한 값을 이용합니다. 단, 불필요한 설정은 감점의 요인이 될 수 있습니다.
- 2) 정해진 USB장치 이외의 저장장치의 반입을 금지합니다.
- 3) 휴대폰 등과 같은 스마트기기는 경기시작 전 심사위원(또는 관리 위원)에게 보관하도록 합니다.
- 4) 리전은 서울(ap-northeast-2) 리전을 사용합니다.
- 5) 과제지에 명시되지 않는 모든 Resource/Instance의 Instance Type은 t2.micro를 사용합니다.
- 6) Instance의 AMI는 "Amazon Linux 2"를 사용합니다.
- 7) 모든 Resource/Instance의 Volume Type은 General Purpose SSD(gp2)를 사용합니다. 만약 Provision IOPS(io1)은 절대로 사용하지 않으며, 감점 또는 실격의 요인이 될 수도 있습니다.
- 8) 과제지에 따로 Name Tag가 지정되어 있지 않은 경우 선수 마음대로 설정이 가능합니다. (Name Tag가 설정되어 있지 않은 Resource는 채점하지 않습니다. 단, Name Tag를 설정할 수 없는 Resource는 설정을 하지 않아도 무관합니다.)
- 9) 채점은 경기 시간 이후에만 진행되며, 선수는 중간에 퇴실할 수 없습니다.
- 10) Administrator 권한과 Programming, Management Console에 접근이 가능한 IAM User를 생성 후 해당 User로 과제 풀이를 진행합니다.
- 11) 해당 애플리케이션은 CloudFront으로 배포를 하지 않습니다.
- 12) 완전하게 공개된 자료만 인터넷에서 열람하여 참고할 수 있도록 합니다. (부정행위로 간주될 수 있습니다.)
- 13) 나머지 사항은 문제지에 지시한대로 수행하며, 지정되지 않은 내용은 AWS Best Practice대로 구성합니다.
- 14) 과제지에 별다른 지시가 없는 경우 모든 Resource/Instance의 Password는 "Skills23@#"로 설정합니다.
- 15) AWS CLI와 Git에 관련된 모든 설정은 Management Instance에서 모두 구성합니다.

3. Code Commit

제공된 코드를 코드커밋에 업로드 합니다. repository 이름은 code commit repository를 하나 생성하고 해당 repository는 cf와 deploy 두 개의 branch를 가지고 있습니다. 제공받은 cf.tar는 repository에 src 디렉토리를 생성하고 아래에 위치시킵니다. deploy에 이용하는 appspec파일도 repository에 추가합니다.

- repository 이름: worldskills-repository
- cf.tar 위치: src/cf.tar

```
worldskills-repository
├── cf
│   └── cf.tar
└── deploy
    ├── config.ini
    ├── Dockerfile
    ├── template/
    │   ├── index.html
    │   └── success.html
    └── upload
```

4. Code Build

어플리케이션 소스코드를 빌드 하기 위해 code build를 사용합니다. cloudwatch log를 활성화 하여 빌드 시작 후 진행상황 로그를 볼 수 있도록 합니다. deploy에 코드가 커밋되면 자동으로 Docker Image를 생성하여 worldskills-ecr에 업로드 합니다.

- Code build name: worldskills-build
- ECR name: worldskills-ecr(프라이빗)

5. Jenkins

3rd Party Tool Jenkins을 이용해서 cf.tar를 압축 풀고 cloudformation 인프라를 생성합니다.

- Jenkins Admin 계정의 이름은 administrator로 지정합니다.
- Jenkins에서 Build를 수행하는 Job의 이름은 "worldskills-jenkins-build"를 사용합니다.
- Jenkins에는 "AWS CodePipeline Plugin"라는 Plugin을 설치해서 Pipeline을 구성합니다.
- Jenkins 서비스는 8080번 포트를 사용해서 관리가 가능하며 아래의 사이트를 참고하여 설치가 가능합니다. (<https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/>)

6. App Description

- 배포된 Application은 웹 사이트에서 파일을 업로드하면 S3 Bucket에 업로드 된 파일을 저장하는 간단한 Web Application 입니다.
- 배포된 Application은 5000번 포트를 사용하여 운영됩니다.
- 배포된 Application을 정상 적으로 사용하기 위해서는 config.ini 파일을 수정 해주어야 합니다.

```
AWS_REGION=ap-northeast-2
AWS_ACCESS_KEY_ID=<KEY_ID 추가합니다.>
AWS_SECRET_ACCESS_KEY=<SECRET_KEY 추가합니다.>
BUCKET_NAME=<S3 버킷>
```

- BUCKET_NAME: 파일을 업로드 하면 저장될 S3 버킷 이름을 지정합니다.
- 해당 Application은 golang으로 만든 앱입니다.

7. Code Deploy

빌드된 산출물이 서버에 복사되어 배포하도록 합니다. appspec.yml 파일을 이용해 배포에 필요한 내용들을 스크립팅 합니다. in-place 배포를 사용해 새로운 바이너리로 배포되도록 합니다. EC2의 Name 태그에 "Deploy"="EC2" 를 가지고 있는 EC2 모드를 배포 대상으로 합니다. 아무것도 설치되지 않은 새로운 EC2가 생성되더라도 deploy가 실행되면 ECR에 있는 이미지를 docker run 명령어를 이용해서 앱을 배포할 수 있도록 하고, 배포 과정에서 ALB의 타겟그룹에도 자동으로 attach 되어야 합니다.

- stop.sh 추천 명령어: `docker kill $(docker ps -q) && docker rmi $(docker ps -q)`
- start.sh 추천 명령어: `docker run`

8. Cloudformation

cloudformation은 수동으로 생성하지 않고 Jenkins를 이용해서 생성하도록 합니다. cf는 ALB 1개, TG 1개, EC2 Instance 2개, VPC 까지 생성할 수 있습니다.

실행 추천 명령어:

```
TAG=$(uuidgen | cut -c 1-8)
```

```
STACK_NAME=worldskills-$TAG
```

```
aws cloudformation create-stack --stack-name $STACK_NAME --template-body file://cf.yml
```