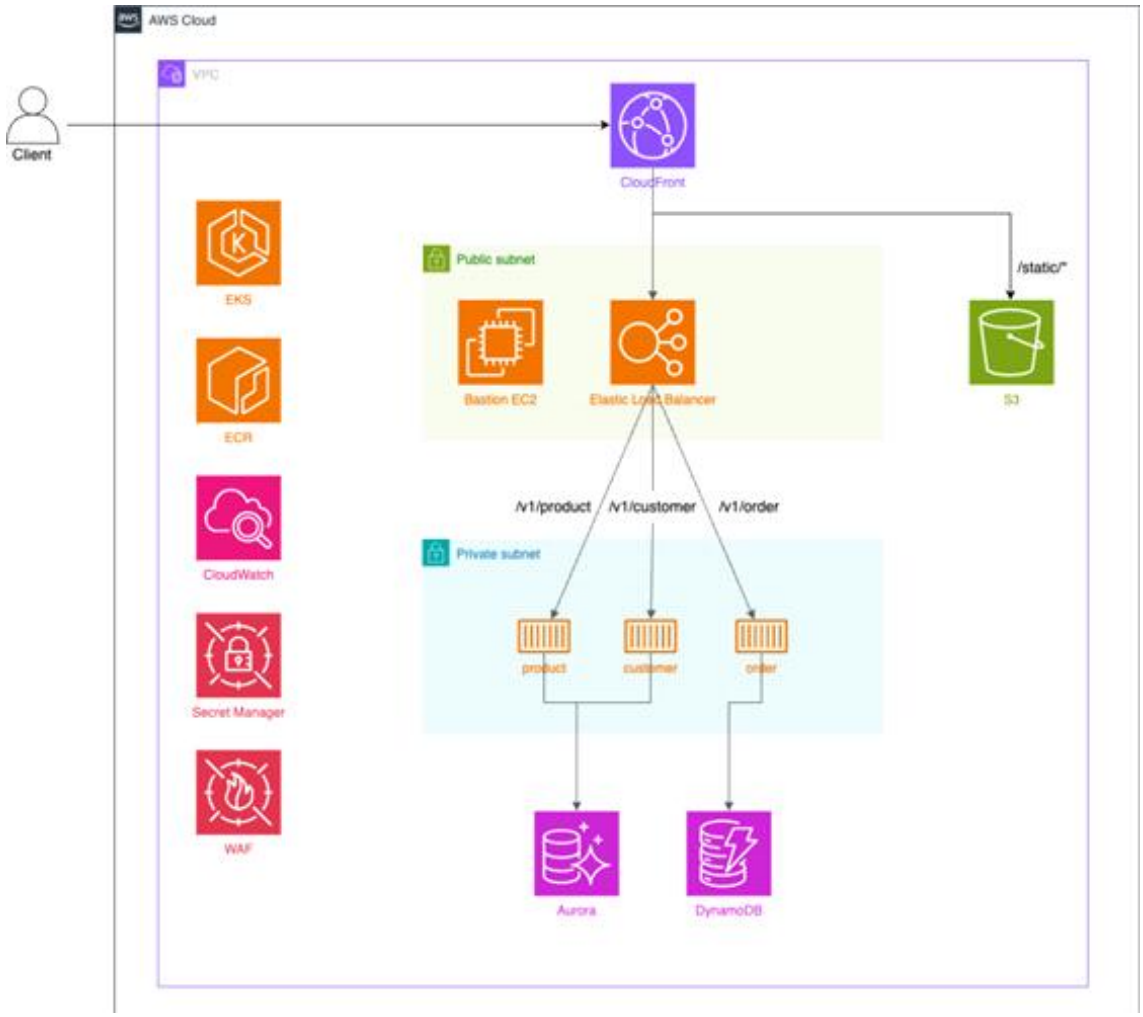
2024년도 전국기능경기대회 과제

직종명	클라우드컴퓨팅	과제명	Web Service Provisioning	과제번호	제1과제
경기시간	4시간	비번호		심사위원 확 인	(인)

1. 요구사항

E-Commerce 서비스를 제공하기 위한 REST API를 배포하고 운영하고자 합니다. REST API는 추후 확장성, 안정성 등을 고려하여 MSA(Micro Service Architecture) 패턴으로 설계되고 개발되었습니다. MSA 패턴의 REST API를 운영하는 것에 있어 여러 장점을 가지는 컨테이너 기반 환경을 선택하였고, 효율적인 컨테이너 관리를 위해 오케스트레이션 툴인 Kubernetes를 사용하기로 선택하였습니다. 그 외에도 성능, 보안, 고가용성, 운영효율성, 지속가능성 등 여러 요소를 고려하여 어플리케이션을 배포하고 인프라를 구축하세요.

다이어그램



Software Stack

AWS	개발언어/프레임워크
- VPC - EC2 - EKS - ELB - ECR - CloudFront - S3 - CloudWatch - RDS - DynamoDB - WAF	- Golang / Gin - Docker

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, 찰과상 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) 선수의 계정에는 비용제한이 존재하며, 이보다 더 높게 과금될 시 계정 사용이 불가능할 수 있습니다.
6) 문제에 제시된 괄호박 는 변수를 뜻함으로 선수가 적절히 변경하여 사용해야 합니다.
7) EC2 인스턴스의 TCP 80/443 outbound는 anyopen하여 사용할 수 있도록 합니다.
8) 과제의 Bastion 서버에서 대부분의 채점이 이루어짐으로 인스턴스를 생성하지 않았거나 종료된 상태면 채점이 불가능하니 각별히 주의하도록 합니다.
9) 과제 종료 시 진행 중인 테스트를 모두 종료하여 서버에 부하가 발생 하지 않도록 합니다.
10) 별도 언급이 없는 경우, ap-northeast-2 리전에 리소스를 생성하도록 합니다.
11) 1페이지의 다이어그램은 구성을 추상적으로 표현한 그림으로, 세부적인 구성은 아래의 요구사항을 만족시킬 수 있도록 합니다. (ex. 서브넷이 2개 이상 존재할 수 있습니다.)
12) 모든 리소스의 이름, 태그, 변수는 대소문자를 구분합니다.
13) 문제에서 주어지지 않는 값들은 AWS Well-Architected Framework 6 pillars를 기준으로 적절한 값을 설정해야 합니다.
14) 불필요한 리소스를 생성한 경우, 감점의 요인이 될 수 있습니다. (e.g. VPC 추가 생성)

3. 네트워크 구성

클라우드 인프라에 대해 네트워크 레벨의 격리 및 분리가 가능하도록 아래 요구사항을 참고하여 VPC를 구성합니다. 10.1.0.0/16 네트워크를 사용하며 고가용성을 고려하여 2개의 az를 가지도록 VPC를 설계합니다. VPC 이름은 wsi-vpc입니다. 구성 시 Name은 Key를 Name으로 갖는 Tag를 의미합니다. 서브넷 구성 시 zero subnet을 허용하며, 서브넷 마스크는 모두 24bit을 사용합니다. 또한 컨테이너를 호스팅하는 서브넷(wsi-app-*)에서는 AWS 내부 네트워크만을 거쳐 안전하게 컨테이너 이미지를 내려받을 수 있도록 구성합니다. 서브넷 이름 뒤의 알파벳은 Availability Zone을 의미합니다.

Name	Internet access	Route table	구분
wsi-app-a	NAT G/W	wsi-app-a-rt	사용 가능한 1번째
wsi-app-b	NAT G/W	wsi-app-b-rt	사용 가능한 2번째
wsi-public-a	Internet G/W	wsi-public-rt	사용 가능한 3번째
wsi-public-b	Internet G/W	wsi-public-rt	사용 가능한 4번째
wsi-data-a	-	wsi-data-rt	사용 가능한 5번째
wsi-data-b	-	wsi-data-rt	사용 가능한 6번째

4. Bastion 서버

EC2를 활용해 Bastion 서버를 구성합니다. bastion 서버의 접근을 위해서 SSH 프로토콜과 TCP 4272를 사용합니다. Bastion 서버는 외부에서 SSH 프로토콜만을 허용하도록 Security Group을 구성하세요. Bastion 서버의 ec2-user는 kubernetes 내의 모든 리소스를 수정, 조회할 수 있는권한이 미리 설정되어 있어야 합니다. (세션을 다시 연결했을 경우에도 적용되어야 합니다.)

Bastion 서버는 채점을 위해서 사용됩니다. 잘 못 구성하였을 경우 모든 채점 항목에서 불이익을 받을 수 있으니 주의합니다.

- Instance type : t3.small
- Subnet : Public subnet A
- 설치 패키지 : awscli, jq, curl, kubectl
- Tag : Name=wsi-bastion
- Security group name : wsi-bastion-sg (명시된 Security Group 하나만을 사용합니다.)
- EC2 IAM Role : 모든 리소스에 대해 full access를 가지는 role을 생성하여 붙입니다. role 이름은 wsi-bastion-role로 설정합니다.

5. 관계형 데이터베이스

product, customer 어플리케이션의 데이터를 저장하기 위해서 관계형 데이터베이스를 구성합니다. 아래의 데이터베이스 구성 요구사항에 따라 테이블을 설계하고 생성하세요. 포트는 기본 포트에서 다른 포트로 반드시 변경해야 합니다. 보안을 위해 인터넷과 연결되지 않는 환경에 구성합니다.

- DB cluster identifier : wsi-aurora-mysql
- Instance class : db.t3.medium
- Engine Version : Aurora MySQL 3.05.2 (compatible with MySQL 8.0.32)
- Encryption : KMS 고객관리형 키
- Log Exports : Audit log, Error log
- Description of schema

database	table	column	data type
product	product	id	VARCHAR(255)
		name	VARCHAR(255)
		category	VARCHAR(255)
customer	customer	id	VARCHAR(255)
		name	VARCHAR(255)
		gender	VARCHAR(255)

6. NoSQL 데이터베이스

어플리케이션의 데이터를 저장하기 위해서 DynamoDB를 구성합니다. 아래의 데이터베이스 정보가 작성된 표에 따라 테이블을 설계하고 생성하세요. 또한 VPC 내부에서 안전하게 통신하도록 구성하여야 하며 order 어플리케이션을 제외한 customer, product에서는 접근하지 못하도록 구성합니다.

- Table name : order
- Encryption at rest : KMS 고객관리형 키
- Description of table

Key Name	Data Type	ETC
id	String	PK
customerid	String	
productid	String	

7. 웹 어플리케이션

서비스를 구성하기 위한 product, customer, order 총 3개의 어플리케이션이 있습니다. 제공된 binary는 Golang/Gin으로 개발하였으며 x86기반 시스템에서 빌드하였습니다.
- 모든 어플리케이션은 실행 시 TCP 8080 포트로 바인딩 됩니다
- 모든 어플리케이션은 표준 출력으로 접근 로그를 출력합니다.
- 모든 어플리케이션은 /healthcheck 경로를 통해 상태 확인을 제공합니다.
- 모든 어플리케이션은 환경변수를 통해 데이터베이스 연결 정보를 어플리케이션에 제공합니다. 어플리케이션 엔드포인트와 데이터베이스 연결을 위한 환경변수 키 값은 아래 테이블들을 참고합니다.

Product

Product 정보를 생성하고 제공하는 API입니다.

- API Spec

Path	Method	Request Format
/v1/product	GET	Query String
		?id=xxxxxx
/v1/product	POST	Request Body
		'{"id":"xxxxxx", "name":"xxxxxx", "category":"xxxxxx"}'

- OS Environment

Environment Key	Description
MYSQL_USER	RDBMS연결에 사용할 사용자명
MYSQL_PASSWORD	RDBMS연결에 사용할 사용자 암호
MYSQL_HOST	RDBMS연결에 사용할 호스트 이름
MYSQL_PORT	RDBMS연결에 사용할 포트번호
MYSQL_DBNAME	RDBMS에 사용할 데이터베이스 이름

Customer

Customer 정보를 생성하고 제공하는 API입니다.

- API Spec

Path	Method	Request Format
/v1/customer	GET	Query String
		?id=xxxxxx
/v1/customer	POST	Request Body
		'{"id":"xxxxxx", "name":"xxxxxx", "gender":"xxxxxx"}'

- OS Environment

Environment Key	Description
MYSQL_USER	RDBMS연결에 사용할 사용자명
MYSQL_PASSWORD	RDBMS연결에 사용할 사용자 암호
MYSQL_HOST	RDBMS연결에 사용할 호스트 이름
MYSQL_PORT	RDBMS연결에 사용할 포트번호
MYSQL_DBNAME	RDBMS에 사용할 데이터베이스 이름

Order

Order 정보를 생성하고 제공하는 API입니다.

- API Spec

Path	Method	Request Format
/v1/order	GET	Query String
		?id=xxxxxx
/v1/order	POST	Request Body
		'{"id":"xxxxxx", "customerid":"xxxxxx", "productid":"xxxxxx"}'

- OS Environment

Environment Key	Description
AWS_REGION	DynamoDB 연결에 사용할 리전 코드 (e.g. ap-northeast-2)

8. Secret Store

어플리케이션에서 Database연결을 위한 환경변수들을 Secrets Manager에 저장해 관리합니다. 어플리케이션은 Secrets Manager에서 환경변수를 사용할 수 있습니다.

- customer 어플리케이션 Secret Name : customer
- product 어플리케이션 Secret Name : product
- order 어플리케이션 Secret Name : order

9. 컨테이너라이징

어플리케이션을 컨테이너로 만든 후 쿠버네티스로 배포합니다. 컨테이너 이미지를 관리하기 위해서 ECR에 업로드를 해서 사용합니다. 이미지는 업로드 시 자동으로 스캔하도록 하며 태그는 모두 latest로 설정합니다. 모든 이미지에 Low 레벨 이상의 취약성이 없도록 합니다 컨테이너가 쿠버네티스에서 실행되어 각 컨테이너의 프로그램은 customer 어플리케이션은 customer, product 어플리케이션은 product, order 어플리케이션은 order 라는 이름의 USER로 실행되어야 합니다. 채점을 위해 customer 와 product 컨테이너에 curl을 포함하여야 하고, 채점 시 정상적으로 작동하여야 합니다.

- customer ECR Name : customer-ecr
- product ECR Name : product-ecr
- order ECR Name : order-ecr

10. 컨테이너 오케스트레이션

어플리케이션들은 컨테이너환경에 배포하고 운영합니다. EKS를 통해서 컨테이너를 배포하고 관리합니다. customer, product 어플리케이션 Pod는 정해진 Managed Node Group에서 운용하고 order 어플리케이션은 명시된 Fargate에서 운용합니다.

- Cluster Name : wsi-eks-cluster
- Kubernetes Version : 1.29

App Nodegroup

노드의 개수는 4개 입니다. customer, product 어플리케이션이 실행됩니다.

- Nodegoup Name : wsi-app-nodegroup
- Node EC2 Instance Tag : Name=wsi-app-node
- Node EC2 Instance Type : t3.large

App Fargate Profile

order 어플리케이션이 실행됩니다.

- Fargate Profile Name : wsi-app-fargate-profile
- Fargate Resource : 0.5vCPU, 1GB Memory

Addon Nodegroup

노드의 개수는 4개 입니다. customer, product, order 어플리케이션을 제외한 나머지 Deployment들은 모두 여기에서 실행됩니다.

- Nodegroup Name : wsi-addon-nodegroup
- Node EC2 Instance Tag : Name=wsi-addon-node
- Node EC2 Instance Type : t3.medium

Kubernetes Resource

customer, product, order 어플리케이션은 wsi Namespace에 생성되어야 합니다.
customer, product, order Deployment의 Pod는 각각 2개씩 운용되어야 합니다.

- customer Deployment Name : customer
- product Deployment Name : product
- order Deployment Name : order

- customer Pod Label : “app: customer”
- product Pod Label : “app: product”
- order Pod Label : “app: order”

11. 로드밸런싱

외부에서 customer, product, order 어플리케이션에 접근하기 위해 ALB를 활용합니다.

- Load Balancer Name : wsi-app-alb
- Load Balancer Security Group Name : wsi-app-alb-sg
- Load Balancer Scheme : internet-facing
- Load Balancer Listen Port : 80

12. 정적 호스팅

S3를 통하여 정적 콘텐츠를 저장하고 제공합니다. 정적 콘텐츠는 ap-northeast-2 리전에 저장되어야 합니다. 모든 콘텐츠는 업로드 시 자동으로 KMS의 임의의 CMK로 암호화되어야 합니다. 모든 콘텐츠 파일은 /static/ 접두사를 가진 오브젝트 키(e.g. /static/index.html)로 업로드 합니다.

- S3 버킷 이름 : apne2-wsi-static-<비번호 3자리>

13. CDN

CloudFront를 통하여 정적 콘텐츠 및 어플리케이션에 접근이 가능하도록 합니다. S3에 업로드되는 정적 콘텐츠를 캐싱할 수 있도록 구성합니다. ALB로의 요청에 대해서는 캐싱하지 않고 Query String도 모두 Origin으로 전달해야 합니다. 사용자가 CloudFront에 HTTP 접근 시에도 HTTPS로 리디렉션 되어 HTTPS로만 접근할 수 있도록 구성합니다.

- Origin : S3와 ALB 2개의 origin을 가지도록 구성
- Edge : 한국뿐만 아니라 전 세계의 유저가 빠른 속도로 접근할 수 있도록 구성
- Tag : Name=wsi-cdn
- 기타 : 채점 시 오동작 예방으로 IPv6는 비활성화하고, 하나의 CloudFront만 생성

14. Logging

Fluentd가 FluentBit로부터 로그를 수신하여 CloudWatch Logs에 저장하도록 중앙 집중식 로깅 솔루션을 구성합니다. customer, product, order 각 어플리케이션에서 발생하는 접근 로그를 각 Log Group에 저장합니다. 다만 /healthcheck 경로에 대한 접근 로그는 Cloudwatch Logs 에 저장되지 않아야 합니다. 로그는 최소 2분 이내에 Cloudwatch Logs로 수집되어야 합니다. Log Group은 임의의 고객 관리형 Key를 사용 합니다.

- order 어플리케이션 로그
- 로그 그룹 이름 : /wsi/webapp/order
- product 어플리케이션 로그
- 로그 그룹 이름 : /wsi/webapp/product
- customer 어플리케이션 로그
- 로그 그룹 이름 : /wsi/webapp/customer

15. 보안 구성

- 보안성 향상을 위해 네트워크 수준의 보안을 적용합니다. Security Group을 구성하여 ALB는 CloudFront 외에는 모든 접근을 허용하지 않도록 구성합니다. product와 customer 어플리케이션이 서로 통신하지 못하도록 합니다.

- 보안성 향상을 위해 어플리케이션 수준의 보안을 적용합니다. AWS WAF를 구성하여 user-agent 에서의 접근을 제한하고자 합니다. AWS CloudFront에 접근할 때 safe-client 라는 단어가 포함된 user-agent 에서 요청한 접근은 Allow 처리하고 이외에 다른 요청들은 모두 "Access Denied" 문구와 403 응답 코드를 반환하도록 구성합니다. AWS CloudFront에서 ALB로 요청을 보낼 때 특정 헤더를 포함하여 요청을 보내고, ALB에서 해당하는 헤더의 값이 맞지 않은 경우 "Access Denied" 문구와 403 응답 코드를 반환하도록 구성합니다.
  - Header Name : X-wsi-header
  - Header Value : Skills2024