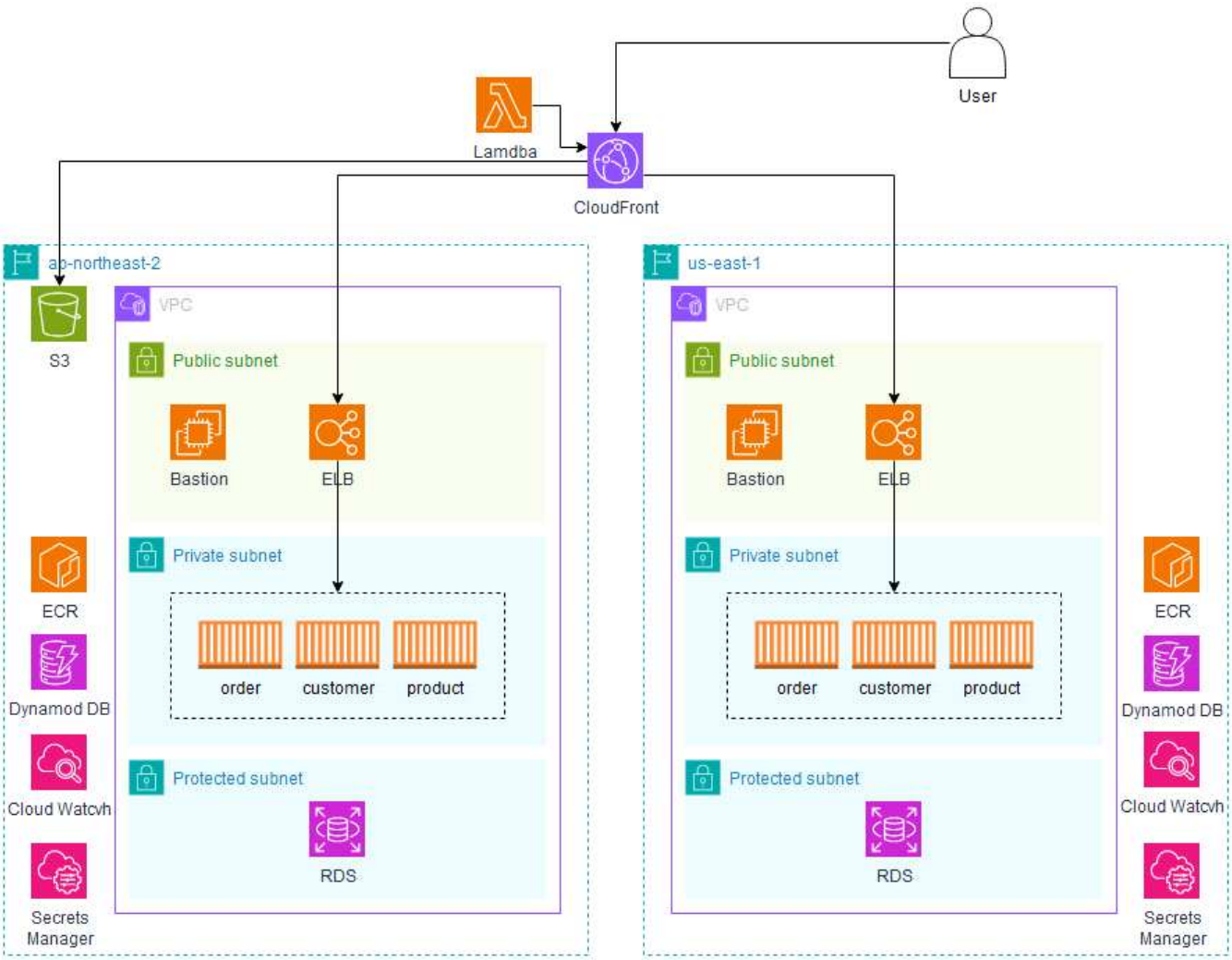
2024년도 전국기능경기대회

직 종 명	클라우드컴퓨팅	과 제 명	Solution Architecture	과제번호	제 1과제
경기시간	4시간	비 번 호		심사위원 확 인	(인)

1. 요구사항

MSA(Micro Service Pattern)의 REST API를 포함하는 웹 서비스 환경을 구축하고 운영하고자 합니다. MSA 패턴의 REST API를 운영하는 데 있어 여러 장점이 있는 컨테이너 기반의 환경과 AWS 관리형 서비스인 EKS를 컨테이너 오케스트레이션 툴을 사용해야 합니다. 그 외에도 여러 가지 AWS 서비스를 사용하여 웹 서비스를 운영할 수 있는 클라우드 플랫폼을 구성해야 합니다. 주어진 아키텍처를 바탕으로 고가용성, 성능, 보안, 운영효율성 등 여러 가지 요소를 고려하여 웹 애플리케이션이 구동할 수 있는 클라우드 플랫폼을 구축해야 합니다.

다이어그램



Software Stack

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

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 비용의 제한이 존재하며, 이보다 더 높게 요금이 부과될 시 계정 사용이 불가능할 수 있습니다.
6) 문제에 제시된 괄호 박스 <>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 발생하지 않도록 합니다.
10) 1페이지의 다이어그램은 구성을 추상적으로 표현한 그림으로, 세부적인 구성은 아래의 요구사항을 만족시킬 수 있도록 합니다. (ex. 서브넷이 2개 이상 존재할 수 있습니다.)
11) 모든 리소스의 이름, 태그, 변수과 변수는 대소문자를 구분합니다.
12) Password 지정이 필요할 경우 "Skill53##"으로 접근이 구성합니다.
13) 불필요한 리소스를 생성한 경우, 감점의 요인이 될 수 있습니다. (e.g. VPC 추가 생성)

3. Disaster Recovery

과제지에 별다른 지시사항이 없을 경우 요구되는 모든 리소스들은 ap-northeast-2(서울), us-east-1(버지니아) 리전에 똑같이 구성합니다. 만약 ap-northeast-2에 구성된 인프라가 어떤 이유로 인해 웹 사이트를 운영할 수 없게 되면, 유입되는 사용자들에게 us-east-1에 생성된 인프라에서 웹 사이트를 제공할 수 있도록 구성합니다. 이때 ap-northeast-2에서 데이터에 대한 CRUD 작업을 진행한 경우, us-east-1에서도 해당 데이터를 읽거나 수정할 수 있어야 하며, 반대의 경우도 마찬가지로 적용되어야 합니다.

4. Networking

클라우드 상에 네트워크 환경을 구축합니다. 아래 주어진 요구사항을 참고하여 구성합니다.

Name	Internet Access	Route Table	CIDR
hrdkorea-vpc	-	-	10.129.0.0/16
hrdkorea-public-sn-a	Direct Access	hrdkorea-public-rt	10.129.0.0/24
hrdkorea-public-sn-b	Direct Access	hrdkorea-public-rt	10.129.1.0/24
hrdkorea-private-sn-a	NAT G/W	hrdkorea-private-a-rt	10.129.11.0/24
hrdkorea-private-sn-b	NAT G/W	hrdkorea-private-b-rt	10.129.12.0/24
hrdkorea-protect-sn-a	Block	hrdkorea-protect-rt	10.129.21.0/24
hrdkorea-protect-sn-b	Block	hrdkorea-protect-rt	10.129.22.0/24

5. Bastion

EC2를 활용하여 Bastion 서버를 구성합니다. 해당 서버에서 접근 불가할 시 채점이 불가함으로 **반드시 SSH를 통한 접속과 권한 문제가 없도록 합니다.** Bastion은 awscli 입력시 Admin Poilcy에 상응하는 권한을 가지고 있어야 합니다. kubectl 사용 시에도 클러스터 접근에 문제가 없고 모든 권한을 가지고 있어야 합니다.

- Instance Type : t3.small
- Name : hrdkorea-bastion
- package: awscliv2, curl, jq, kubectl

6. Relational Database

Customer,Product 애플리케이션의 데이터를 저장하기 위해 관계형 데이터베이스를 구성합니다.
엔진은 Aurora MYSQL Engine을 사용합니다. 데이터베이스의 요금은 요청한 만큼 요금을 지불하는 방식으로 구성하도록 합니다. 포트는 3409로 변경하여 사용하도록 합니다.

- RDS DB Identitier - ap-northeast-2 : hrdkorea-rds-instance
- RDS DB Identitier - us-east-1 : hrdkorea-rds-instance-us
- instance type : db.r5.large
- Engine : Aurora MySQL 3.04.0
- username : hrdkorea_user

7. NoSQL DataBase

NoSQL기반 데이터베이스인 DynamoDB를 구성합니다. 해당 DynamoDB는 요청한 만큼 요금을 지불하는 방식으로 구성하도록 합니다.

- Table Name : order

8. Application

주어진 3개의 애플리케이션을 배포하고 사용자에게 제공해야 합니다. 각 애플리케이션의 동작은 다음을 참고합니다.

- 모든 애플리케이션은 Golang/Gin을 사용하여 개발되었으며, x86 시스템에서 빌드하였습니다.
- 모든 애플리케이션은 TCP 8080 포트를 사용합니다.
- 모든 애플리케이션은 표춘 출력으로 접근 로그를 출력합니다.
- 모든 애플리케이션은 /healthcheck 경로로 상태 확인을 제공합니다.

customer

- API

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

- RDBMS Table

Column	Data Type	ETC
id	VARCHAR(255)	-
name	VARCHAR(255)	-
gender	VARCHAR(255)	-

■ Table의 이름은 customer 로 설정해야 합니다.(하드코딩되어 있음.)

- OS Environment

Environment Key	Desciption
MYSQL_USER	RDBMS 연결에 사용할 사용자명
MYSQL_PASSWORD	RDBMS 연결에 사용할 사용자 암호
MYSQL_HOST	RDBMS 연결에 사용할 호스트 이름
MYSQL_PORT	RDBMS 연결에 사용할 포트 번호
MYSQL_DBNAME	RDBMS 연결에 사용할 데이터베이스 이름

product

- API

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

- RDBMS Table

Column	Data Type	ETC
id	VARCHAR(255)	-
name	VARCHAR(255)	-
category	VARCHAR(255)	-

■ Table의 이름은 product로 설정해야 합니다.(하드코딩되어 있음.)

- OS Environment

Environment Key	Desciption
MYSQL_USER	RDBMS 연결에 사용할 사용자명
MYSQL_PASSWORD	RDBMS 연결에 사용할 사용자 암호
MYSQL_HOST	RDBMS 연결에 사용할 호스트 이름
MYSQL_PORT	RDBMS 연결에 사용할 포트 번호
MYSQL_DBNAME	RDBMS 연결에 사용할 데이터베이스 이름

order

- API

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

- DynamoDB Table

Column	Data Type	ETC
id	String	-
customerid	String	-
productid	String	-

■ Table의 이름은 order로 설정해야 합니다.(하드코딩되어 있음.)

- OS Environment

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

9. ECR

Docker Image를 저장하기 위하여 ECR을 생성합니다. 해당 ECR은 private으로 존재해야하며, 모든 이미지에는 CVE 취약성이 없도록 합니다. 또한 ap-northeast-2 리전에서 image를 업로드시 us-east-1 리전에도 같은 Image가 업로드 되도록 합니다.

- ECR Name : hrdkorea-ecr-repo
- Docker Image CustomerTag : customer
- Docker Image Product Tag : product
- Docker Image Order Tag : order

10. S3

S3를 통하여 정적 콘텐츠를 제공합니다. 정적 콘텐츠는 ap-northeast-2 리전에만 생성하도록 합니다. 제공한 html 파일은 /static 경로에 업로드하도록 합니다

- Bucket Name : hrdkorea-static-<임의의 4자리 숫자>

11. CloudFront

CloudFront를 통하여 정적 콘텐츠 및 애플리케이션에 접근이 가능하도록 합니다. S3에 업로드되는 정적 콘텐츠를 캐싱할 수 있도록 구성합니다. 사용자가 Cloudfront에 대한 HTTP 접근 시에도 HTTPS로만 접근할 수 있도록 합니다. 또한 채점 시 오동작 예방으로 IPv6는 비활성화하고, 하나의 CloudFront만 생성하도록 합니다.

- Origin : S3 1개와 ALB 2개의 Origin을 가지도록 구성
- Edge : 한국뿐만 아니라 전 세계의 유저가 빠른 접근할 수 있도록 구성
- CloudFront Name : hrdkorea-cdn

12. Load Balancer

애플리케이션을 부하 분산하기 위해 L7 Load Balancer를 생성합니다. 과제에서 제시한 API를 제외한 나머지 요청은 404 Error Code를 반환하게 구성합니다. 또한 Health Check 요청 시 아래와 같이 Query String으로 Health Check가 되게 구성합니다.

- Load Baalcner Name : hrdkorea-app-alb
- Load Balancer Schema : internet-facing
- Load Balancer Port : 80
- URI List

customer	order	product

- Query String

http://<ALB DNS>/healthcheck?path=<uri>

- ex)

http://<ALB DNS>/healthcheck?path=customer

path=customer 일 경우 customer application으로 health check 되도록 구성

13. Logging

CloudWatch를 통하여 중앙 집중식 로깅 솔루션을 구성합니다. 다음과 같은 요구사항에 맞추어 widget을 구성합니다. 메트릭의 주기는 1분이며 값의 통계는 합계 기준으로 합니다.

- ap-northeast-2 dashboard name : seoul-eks-cluster-ds
- us-east-1 dashboard name : us-eks-cluster-ds

- Widget Name : Node Active summary

현재 EKS의 Node의 개수를 모니터링 가능하도록 구성합니다. 또한 Widget의 유형은 number 유형으로 구성하도록 합니다.

14. 컨테이너라이징

EKS를 통하여 컨테이너를 배포 및 관리합니다. Kubernetes Cluster의 Control Plane에서 발생하는 모든 로그를 CloudWatch Logs에서 확인할 수 있어야 합니다. 모든 application 및 addon은 hrdkorea라는 Namespace가지고 있어야합니다. addon은 Fargate에 동작해야하며, 모든 애플리케이션은 EC2에서 동작해야합니다. 보안 관리를 위하여 secrets manager관련된 addon을 제외한 모든 addon은 fargate profile에서 동작하도록 합니다.

- Cluster Name : hrdkorea-cluster
- kubernetes version : 1.29
- Node group subnet = private subnets

Deployment

모든 Deployment는 아래의 요구사항을 참고하여 구성합니다.

- Customer Deployment Name : customer-deployment
- Product Deployment Name : product-deployment
- Order Deployment Name : order-deployment

Addon profile

- Fargate Profile Name : hrdkorea-addon-profile

Customer Node

customer 애플리케이션은 반드시 customer Nodegroup에서 운용해야 하며, customer 애플리케이션과 secrets manager관련된 addon을 제외한 다른 리소스들은 customer Nodegroup에 존재해서는 안 됩니다.

- NodeGroup Name : hrdkorea-customer-ng
- Node EC2 Instance Name : hrdkorea-customer-ng
- Node EC2 Instance Type : t3.large
- Node label : skills/dedicated: customer

Product Node

Product 애플리케이션은 반드시 Product Nodegroup에서 운용해야 하며, Product 애플리케이션과 secrets manager관련된 addon을 제외한 다른 리소스들은 Product Nodegroup에 존재해서는 안 됩니다.

- NodeGroup Name : hrdkorea-product-ng
- Node EC2 Instance Name : hrdkorea-product-ng
- Node EC2 Instance Type : t3.large
- Node label : skills/dedicated: product

Order Node

order 애플리케이션은 반드시 order Nodegroup에서 동작해야 하며, order 애플리케이션과 secrets manager관련된 addon을 제외한 다른 리소스들은 order Nodegroup에 존재해서는 안됩니다.

- NodeGroup Name : hrdkorea-order-ng
- Node EC2 Instance Name : hrdkorea-order-ng
- Node EC2 Instance Type : t3.large
- Node label : skills/dedicated: order