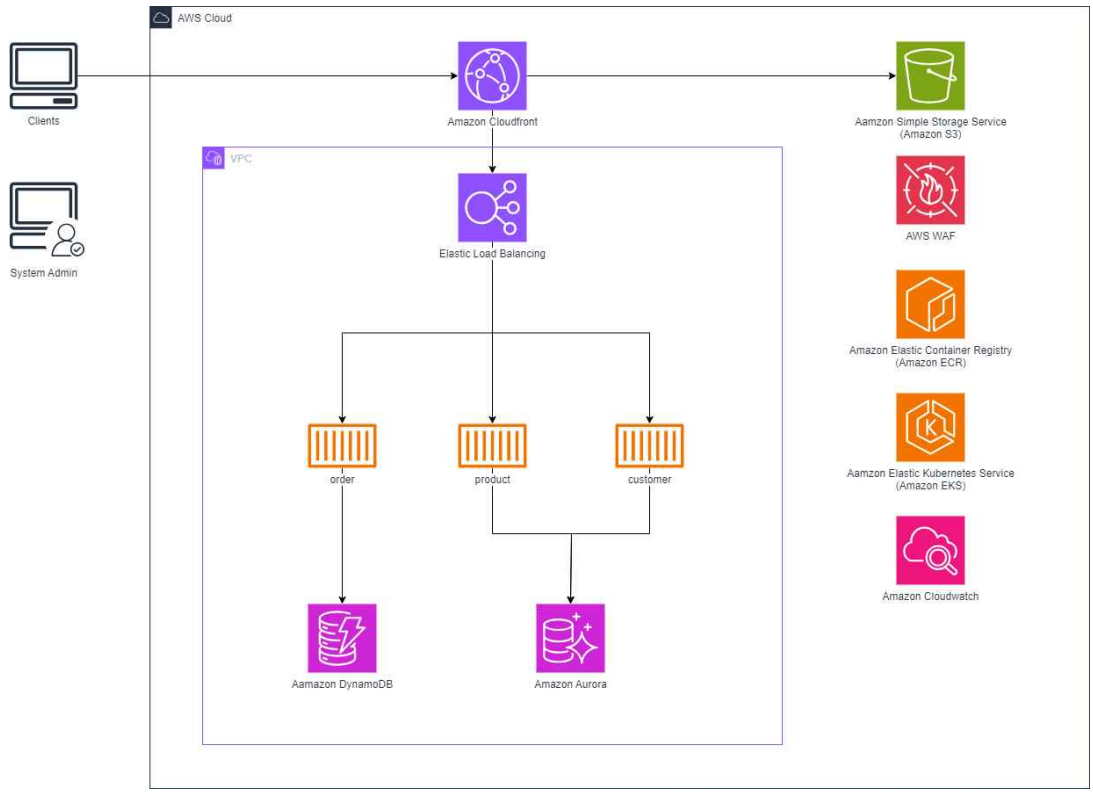
2024년도 전국기능경기대회

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 환경을 구성하고 컨테이너 오케스트레이션 툴로 AWS EKS를 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	
- WAF	
- S3	
- Cloudwatch	
- RDS	
- DynamoDB	

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) 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, 변수과 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 분리를 할 수 있도록 아래 요구사항을 참고하여 VPC를 구성합니다. 컨테이너를 호스팅하는 서브넷(App Subnets)에서는 AWS 내부 네트워크만을 거쳐 안전하게 컨테이너 이미지를 내려받을 수 있도록 구성합니다. ECR 이미지를 내부 네트워크로 다운로드 받기위한 모든 구성을 해야합니다. 서브넷 이름 뒤의 알파벳은 Availability Zone을 의미합니다.

VPC 정보

- VPC CIDR : 10.1.0.0/16
- VPC Tag : Name=skills-vpc
- Internet G/W Tag : Name=skills-igw

App subnet A 정보

- CIDR : 10.1.0.0/24
- Tag : Name=skills-app-a
- 외부 통신 : NAT G/W를 구성하여 인터넷 접근이 가능하도록 구성
- Route table Tag : Name=skills-app-a-rt
- NAT G/W Tag : Name=skills-natgw-a

App subnet B 정보

- CIDR : 10.1.1.0/24
- Tag : Name=skills-app-b
- 외부 통신 : NAT G/W를 구성하여 인터넷 접근이 가능하도록 구성
- Route table Tag : Name=skills-app-b-rt
- NAT G/W Tag : Name=skills-natgw-b

Public subnet A 정보

- CIDR : 10.1.2.0/24
- Tag : Name=skills-public-a
- 외부 통신 : Internet G/W 를 구성하여 인터넷을 접근
- Route table Tag : Name=skills-public-rt

Public subnet B 정보

- CIDR : 10.1.3.0/24
- Tag : Name=skills-public-b
- 외부 통신 : Internet G/W를 구성하여 인터넷을 접근
- Route table Tag : Name=skills-public-rt

Data subnet A 정보

- CIDR : 10.1.4.0/24
- Tag: Name=skills-data-a
- 외부 통신 : 인터넷 접근이 필요하지 않음
- Route table Tag : Name=skills-data-rt

Data subnet B 정보

- CIDR : 10.1.5.0/24
- Tag: Name=skills-data-b
- 외부 통신 : 인터넷 접근이 필요하지 않음
- Route table Tag : Name=skills-data-rt

4. Bastion 서버

EC2를 활용해 Bastion 서버를 구성합니다. 해당 서버에 접근이 불가할 시 채점이 불가함으로 반드시 SSH를 통한 접속과 권한 문제가 없도록 합니다. kubectl 사용 시에도 클러스터 접근에 문제가 없고 모든 권한을 가지고 있어야 합니다.

- Instance type : t3.small
- AMI Type : Amazon Linux 2023
- Subnet : Public subnet A
- 설치 패키지 : awscli, jq, curl, kubectl
- Tag : Name=skills-bastion
- EC2 IAM Role : AdministratorAccess 정책을 지닌 role을 생성하여 붙입니다.
  role 이름은 skills-bastion-role로 설정합니다.

5. 관계형 데이터베이스

Customer와 Product 애플리케이션의 데이터를 저장하기 위해서 관계형 데이터베이스를 구성합니다. 테이블에 임의의 데이터를 삽입하면 성능 저하가 생길 수 있으므로 추가적인 데이터 삽입을 하지 않도록 합니다. 아래 설명을 참고하여 데이터베이스와 테이블을 설계, 생성해야 합니다. 데이터베이스는 인터넷 통신이 되지 않는 data 서브넷에 위치해야 합니다.
또한 고가용성을 위해 읽기 복제본을 생성해야 합니다. 데이터베이스에 저장되는 데이터는 암호화 되어야 합니다.

- RDS db identifier : skills-aurora-mysql
- instance type : Serverless v2 (Minimum capacity : 2ACUs, Maximum capacity : 8ACUs)
- Engine : Aurora MySQL 메이저 버전 8.0의 기본값
- database name : skills
- table name : customer, product
- customer table columns : id(VARCHAR 255), name(VARCHAR 255), gender(VARCHAR 255)
- product table columns : id(VARCHAR 255), name(VARCHAR 255), category(VARCHAR 255)

6. 비관계형 데이터베이스

Order 애플리케이션의 데이터를 저장하기 위해서 비관계형 데이터베이스를 구성합니다. 비관계형 데이터베이스를 구성하기위해 Amazon DynamoDB를 사용합니다. DynamoDB에 저장되는 데이터는 CMK로 암호화 되어야 합니다.

- table name : order
- key name : id(String, PK), customer(String), productid(String)
- Capacity mode : On-demand

7. 웹 애플리케이션

Customer, Product, Order 애플리케이션이 있습니다. 제공된 binary는 golang/gin을 사용하여 개발되었으며, x86 시스템에서 빌드하였습니다. 애플리케이션 실행 시 바인딩되는 포트는 TCP/8080입니다. 모든 애플리케이션은 표준 출력으로 접근 로그를 출력합니다.

- Customer

Path	Method	Query String	Reqeust body
/v1/customer	GET	?id=xxxxxxx	-
/v1/customer	POST	-	'{"id":"xxxxxx","name":"xxxxxxx","gender":"xxxxxx"}'
/healthcheck	GET	-	-

- Product

Path	Method	Query String	Reqeust body
/v1/product	GET	?id=xxxxxxx	-
/v1/product	POST	-	'{"id":"xxxxxx","name":"xxxxxxx","category":"xxxxxx"}'
/healthcheck	GET	-	-

Customer, Product 애플리케이션은 실행 시 환경 변수를 읽어 관계형 데이터베이스에 연결합니다. 아래 테이블의 정보를 담고 있어야 합니다. 환경변수 값은 Pod에 직접적으로 명시해선 안 되며 Secrets Manager에 저장된 값을 External Secrets을 사용해 읽어와야 합니다. 환경변수 값을 저장하기 위한 Secrets Manager의 Secret 이름은 skills-rds-secret으로 설정합니다. Secret Manager에 저장된 값은 3일에 1번씩 변경되도록 설정해야합니다

Enviroment 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	RDBMS연결에 사용할 데이터베이스 이름 (application)

- Order

Path	Method	Query String	Reqeust body
/v1/order	GET	?id=xxxxxxx	-
/v1/order	POST	-	'{"id":"xxxxxx","customerid":"xxxxxxx","productid":"xxxxxx"}'
/healthcheck	GET	-	-

Order 애플리케이션은 실행 시 환경 변수를 읽어 DynamoDB에 연결합니다. 아래 테이블의 정보를 담고 있어야 합니다.

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

제공된 애플리케이션 바이너리를 사용해서 컨테이너 이미지를 생성하고 ECR 리포지토리에 업로드 해야합니다. 각 애플리케이션 마다 다른 ECR 리포지토리를 사용해야 합니다.

8. 컨테이너 오케스트레이션

EKS 기반의 Kubernetes로 컨테이너 오케스트레이션을 구성합니다. Kubernetes Cluster의 Control Plane에서 발생하는 모든 로그는 CloudWatch Logs에서 확인할 수 있어야 하며, Kubernetes의 Secret 리소스들은 반드시 CMK로 암호화 되어야 합니다. Private 형태의 EKS 클러스터를 생성하고, Kubernetes API는 외부에서 접근 불가능해야 하며, Bastion Server에서만 접근할 수 있어야 합니다. customer, product, order 애플리케이션은 App Nodegroup에서 운용해야 하고, 각 애플리케이션마다 최소 2개의 Pod를 운용하여야 합니다. App Nodegroup은 최소 2개의 노드를 운용해야 합니다. Addon들은 반드시 Addon Nodegroup에서 운영하여야 합니다. Addon Nodegroup은 최소 2개의 노드를 운용해야 합니다. CoreDNS는 Fargate로 실행되어야 합니다. ExternalSecret 오브젝트 이름은 application-db-secret로 설정하고 app namespace에 배포 되어야 합니다.

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

Addon Node Group

- Nodegroup Name: skills-eks-addon-nodegroup
- Node EC2 Instance Tag : Name=skills-eks-addon-node
- Node EC2 Instance Type : t3.large

App Node Group

- Nodegroup Name: skills-eks-app-nodegroup
- Node EC2 Instance Tag : Name=skills-eks-app-node
- Node EC2 Instance Type : t3.large

CoreDNS Fargate Profile

- Fargate Profile Name : coredns-profile

Kubernetes Resource

- 모든 애플리케이션은 app Namespace에 생성해야 합니다.
- Deployment의 이름은 각각 애플리케이션의 이름이어야 합니다.
  (customer, product, order)
- 각 애플리케이션은 app=<애플리케이션 이름>에 해당하는 label을 가져야합니다.
  (app=customer, app=product, app=order)

9. 로드밸런서

Nginx Ingress Controller를 이용하여 NLB를 구성하여 외부에서 customer, product, order 각각의 애플리케이션 API에 접근할 수 있도록 구성합니다. 약간의 성능 저하를 감수하고 보안성을 향상시키기 위해 외부에서 CloudFront를 통해 NLB에 접근할 수 있도록 구성합니다. Path 기반으로 규칙을 작성해야 합니다. Path 라우팅 시 *을 이용한 라우팅은 금지합니다. ingress는 app namespace에 ingress-nginx라는 이름으로 생성해야 합니다. Nginx Ignress Controller는 ingress-nginx namespace에 구성해야 합니다. Nginx Ingress Controller 이미지는 반드시 아래 이미지를 사용해야 합니다.

- Nginx Ingress Controller Image : registry.k8s.io/ingress-nginx/controller:v1.10.0
- Nginx Ingress Controller Deployment Name : ingress-nginx-controller
- Network facing : Internet-facing
- Listen : HTTP 80

10. S3

S3를 통하여 정적 콘텐츠를 저장하고 제공합니다. 모든 콘텐츠 파일은 /static/ 접두사를 가진 오브젝트 키(e.g. /static/index.html)로 업로드 합니다. 모든 콘텐츠는 업로드 시 자동으로 KMS의 CMK로 암호화되어야 합니다. /static/ 접두사를 가진 모든 오브젝트는 CloudFront를 통해 외부 사용자들에게 제공될 수 있어야 합니다. CloudFront 및 접근 권한을 가지는 리소스 외에는 외부에서 접근하지 못하도록 구성합니다.

- S3 Bucket Name : skills-static-<임의의 4자리 영문>

11. Cloud Front

CloudFront를 통하여 정적 콘텐츠 및 애플리케이션에 접근이 가능하도록 합니다. S3에 업로드되는 정적 콘텐츠를 캐싱할 수 있도록 구성합니다. NLB로의 요청에 대해서는 캐싱하지 않고 Query String도 모두 Origin으로 전달해야 합니다. 사용자가 CloudFront에 HTTP 접근 시에

도 HTTPS로 리디렉션 되어 HTTPS로만 접근할 수 있도록 구성합니다.
  - Origin : S3와 NLB 2개의 origin을 가지도록 구성
  - Edge : 한국뿐만 아니라 전 세계의 유저가 빠른 속도로 접근할 수 있도록 구성
  - Tag : Name=skills-cdn
  - 기타 : 채점 시 오동작 예방으로 IPv6는 비활성화하고, 하나의 CloudFront만 생성

12. 로그

Opensearch를 통해 로깅 솔루션을 구성합니다. customer, product, order 각 애플리케이션에서 발생하는 접근 로그를 Fluent-bit DaemonSet을 이용하여 Opensearch에 전송해야합니다. Fluent-bit DaemonsSet은 default namespace에 배포되어야 합니다. Fluent-bit은 App nodegroup에서만 운영되어야 합니다. 로그는 30초안에 Opensearch Dashboards에서 조회가 가능해야 합니다. 또한 Log Discover을 위해 index pattern을 생성해두어야 합니다.

- Opensearch Domain Name: skills-opensearch-domain
- Opensearch Engine version : Opensearch 2.13
- Opensearch Instance Type : t3.medium.search
- Fluent-bit Daemonset Name : fluent-bit
- customer application Index Name : customer-<YYYY>.<MM>.<DD>
    - customer application index pattern : customer-*
- product application Index Name : product-<YYYY>.<MM>.<DD>
    - product application index pattern : product-*
- order application Index Name : order-<YYYY>.<MM>.<DD>
    - order application index pattern : order-*

13. 모니터링

Cloudwatc Container Insight를 통해 EKS Cluster를 모니터링 할 수 있어야 합니다.
app Namespace에 있는 Pod들의 CPU utilization과 Memory utilization을 확인할 수 있어야 합니다.

14. 보안

보안성 향상을 위해 애플리케이션 수준의 보안을 적용합니다. AWS WAF를 이용하여 CF에 대한 요청 중 HTTP Method GET과 POST를 제외한 다른 Method의 요청들은 405(Not Allowed) 응답코드를 반환하도록 구성하세요. 또한 요청의 쿼리스트링의id 필드에 baduser라는 단어가 포함 되었다면 403(Forbidden)을 반환하도록 구성하세요.