

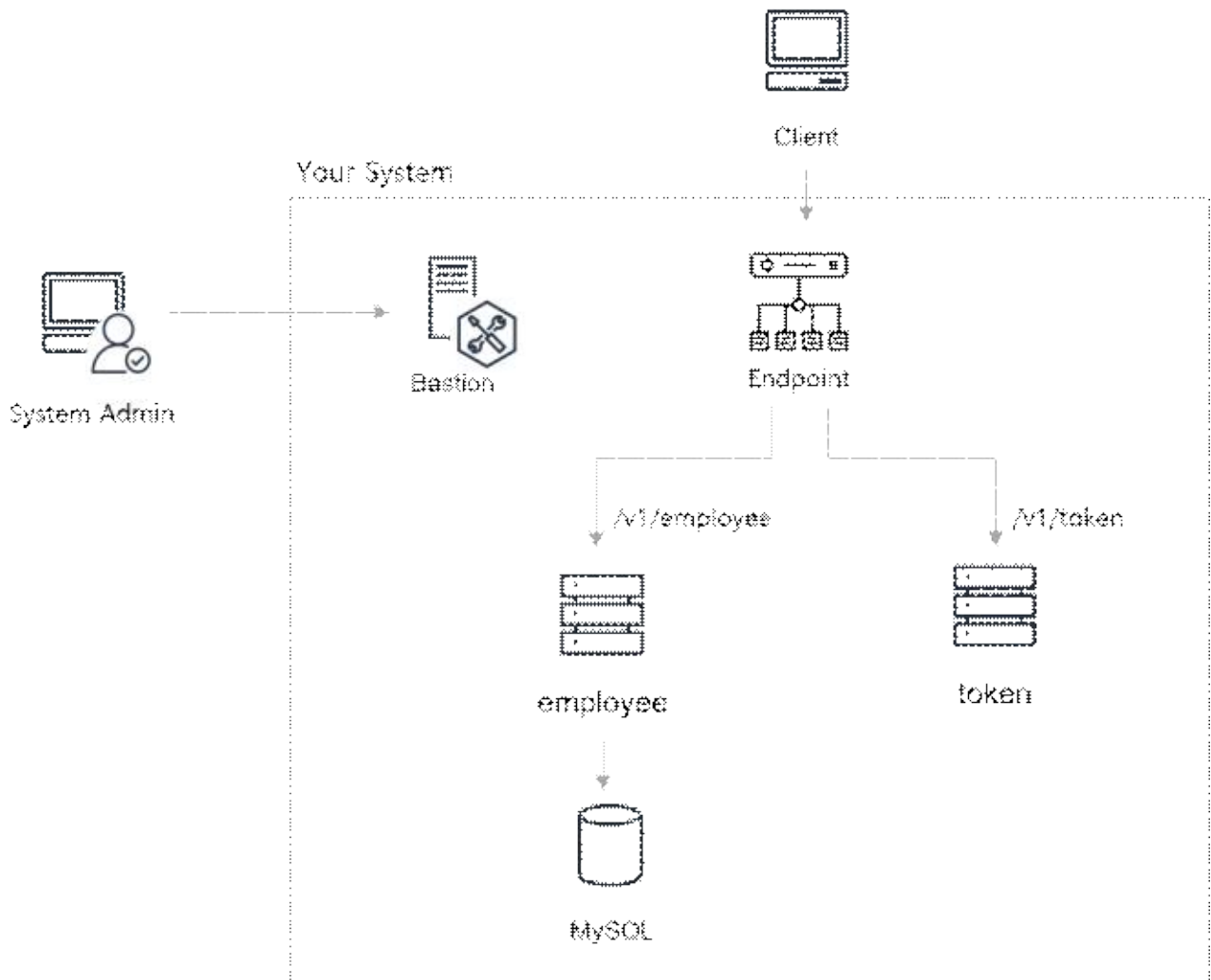
2024년도 전국기능경기대회

직 종 명	클라우드컴퓨팅	과 제 명	System operation	과제번호	제 3과제
경기시간	3시간	비 번 호		심사위원 확 인	(인)

1. 요구사항

개발된 어플리케이션을 배포하여 사용자들에게 서비스를 제공하는 시스템을 구축하고 운영을 위한 모니터링 시스템을 구축합니다. 목표로 설정한 서비스 수준을 만족하도록 트래픽을 처리해야 하며, 구축한 시스템에 발생할 수 있는 장애와 오류를 빠르게 감지하여 최적의 사용자 경험을 제공하도록 시스템을 운영해야 합니다.

다이어그램



필수 Software Stack

AWS	개발언어/프레임워크
- VPC - EC2 - RDS	- Golang / Gin

사용 가능 Software Stack

AWS	개발언어/프레임워크
- ECS, EKS, ECR - ELB, API Gateway - CloudFront - S3 - Cloudwatch - RDS - WAF	- 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 괄호박스 <>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. Bastion

EC2를 활용해 Bastion 서버를 구성합니다. bastion 서버의 접근을 위해서 SSH 프로토콜과 TCP 22를 사용합니다. Bastion 서버는 외부에서 접근할 수 있어야 합니다. 채점을 위해 생성하는 EC2 인스턴스로, 구성에 문제가 있을 시 채점에 불이익이 있을 수 있으므로 주의하도록 합니다.

- Instance type : t3.small
- 설치 패키지 : awscli, jq, curl
- Tag : Name=apdev-bastion
- EC2 IAM Role : 모든 리소스에 대해 full access를 가지는 role을 생성하여 붙입니다.
role 이름은 apdev-bastion-role로 설정합니다.

4. 관계형 데이터베이스

Employee 어플리케이션의 데이터를 저장하기 위해서 관계형 데이터베이스를 구성합니다. 어플리케이션은 MySQL Community 엔진을 지원합니다.

- DB identifier : apdev-rds-instance
- Deployment options: Multi-AZ DB instance
- DB instance class: db.t3.micro
- Storage type: General Purpose SSD (gp3)
- Engine : MySQL Community 8.0

아래의 명령어를 참고하여 테이블을 생성할 수 있습니다.

```
CREATE TABLE employees (  
  emp_no      INT          NOT NULL,  
  birth_date  DATE          NOT NULL,  
  first_name  VARCHAR(14)   NOT NULL,  
  last_name   VARCHAR(16)   NOT NULL,  
  gender      ENUM ('M','F') NOT NULL,  
  hire_date   DATE          NOT NULL,  
  PRIMARY KEY (emp_no)  
);
```

정상적인 동작을 위해서 load_employees.dump 파일을 사용하여 데이터를 삽입해야 하며, 삽입한 데이터는 임의로 수정하거나 삭제하지 않도록 합니다. 발생하는 트래픽 외 임의의 데이터를 삽입하면 성능 저하가 생길 수 있으므로 주의하도록 합니다. 어플리케이션 동작과 트래픽 패턴에 맞게 테이블 구조 재설계가 필요할 수 있습니다.

5. 웹 어플리케이션

Token, Employee 두 개의 어플리케이션이 있습니다. 제공된 binary는 x86기반 EC2의 Amazon Linux 2023 에서 빌드하고 동작을 확인하였습니다. go version은 go1.22.2 linux/amd64 입니다. 어플리케이션 실행 시 바인딩되는 포트는 TCP/8080 입니다. 모든 어플리케이션은 access log를 stdout, stderr로 출력합니다.

- Token

임의의 토큰을 발급하는 API를 제공합니다. 어플리케이션의 동작은 아래를 참고합니다. Token API는 최소한의 가용성을 위해 5초 이하의 응답시간을 보장하고자 하며, 1초 이하의 응답시간을 서비스 수준 목표로 설정하였습니다. 해당 기준으로 서비스 수준을 측정하고 어플리케이션을 운영하도록 합니다.

Path	Method	Request (example)	Response code
/v1/token	POST	{"length": 256}	201
/healthcheck	GET	-	200

- Employee

직원 정보를 관리하는 API를 제공합니다. 어플리케이션의 동작은 아래를 참고합니다. Employee API는 최소한의 가용성을 위해 5초 이하의 응답시간을 보장하고자 하며, 0.2초 이하의 응답시간을 서비스 수준 목표로 설정하였습니다. 해당 기준으로 서비스 수준을 측정하고 어플리케이션을 운영하도록 합니다.

Path	Method	Request (example)	Response code
/v1/employee	POST	{ "emp_no":500001, "birth_date":"1957-05-02", "first_name":"dbdump", "last_name":"dbdump", "gender":"M", "hire_date":"1997-11-30" }	201
/v1/employee	GET	?first_name=dump1&last_name=dump1	200
/healthcheck	GET	-	200

Employee 어플리케이션은 실행 시 아래의 환경변수를 읽어 관계형 데이터베이스에 연결합니다.

Environment Key	Data Type	Value
MYSQL_USER	string	db 접근 권한을 가지는 유저의 이름
MYSQL_PASSWORD	string	db 접근 권한을 가지는 유저의 암호
MYSQL_HOST	string	db 읽기/쓰기가 가능한 주소
MYSQL_PORT	integer	db 읽기/쓰기가 가능한 주소의 포트번호
MYSQL_DBNAME	string	논리적인 데이터베이스 이름 (dev)

6. 트래픽 처리

어플리케이션을 배포하고 외부 사용자들에게 서비스를 제공할 수 있도록 시스템을 구축합니다. 어플리케이션 배포를 위해 EC2 컴퓨팅 리소스를 사용할 수 있으며, 온디맨드 t3.micro 인스턴스 타입 사용해야 합니다. 오케스트레이션을 위한 다른 AWS 서비스를 사용해도 무관하지만 Fargate, Lambda 등 EC2 외의 컴퓨팅 리소스를 사용할 수 없습니다.

트래픽 처리를 위해 최소한의 리소스만을 사용하도록 합니다. 과도한 리소스를 사용할 경우 감점의 원인이 될 수 있습니다.

트래픽 처리를 위해 사용자에게 제공되는 엔드포인트를 하나로 단일화하도록 합니다. 경기 시작 1시간 뒤부터 트래픽이 발생하여 당신의 시스템(엔드포인트)에 주입됩니다. 경기 중 발생하는 트래픽을 목표 서비스 수준에 만족하도록 처리하세요. 모니터링 환경과 로깅 솔루션을 구축하여 로그 분석, 트래픽 패턴 분석, 시스템 개선, 시스템 장애/오류 감지 및 대처를 수행해야 합니다.

서비스 중단 없이 안정적으로 서비스를 제공할 수 있도록 가용성 확보합니다. 이를 고려하여 시스템을 설계하고 구축한 뒤 운영할 수 있도록 합니다.

사용자에게 제공하는 엔드포인트로의 비정상적인 요청은 Block 하도록 하며, 403 응답코드를 내려주도록 합니다. 단, 제공하는 API 외의 요청은 404 응답코드를 내려주도록 합니다. 예제)

- /v1/token 으로의 비정상 요청: 403 응답코드
- /v1/none 으로의 비정상 요청: 404 응답코드