Test Project

Cloud Computing

Friendly match

2023. Aug

This is not the official competition and not related to the national selection competition.

Overview

- EKS 기반 REST API를 제공하는 클라우드 인프라를 만드는 것이 목표 입니다.
- 인프라 설계 시 보안, 운영효율, 신뢰성, 고가용성, 성능을 고려해야 합니다.
- 경기 시작 한시간 후 부터 request가 주입 되며, 이를 적절하게 처리해야 합니다.

Architecture

1. VPC

- Network CIDR : 10.100.0.0/16
- 아래 서브넷들을 각각 3개의 AZ를 가지도록 구성합니다.
  - 인터넷에서 직접 접근 가능한 public subnet
  - NAT를 사용하는 private subnet
  - 인터넷으로 부터 격리된 protected subnet

2. Application

- Python Fast API 기반 RDB에 데이터 생성 및 조회하는 REST API를 제공하는 user 앱입니다.
- 해당 어플리케이션은 데이터의 수정, 삭제를 제공하는 API만 존재 합니다.
- 앱 실행 시 Python3.9 환경에서 아래 명령어 사용을 권장합니다.
  Python3 ./user.pyc
- 실행 시 파라메터 정보를 위해 환경 변수 등록 필요합니다.
  - DB_WRITE_HOST : 데이터베이스의 쓰기가 가능한 엔드포인트
  - DB_READ_HOST : 데이터베이스의 읽기 전용 엔드포인트
  - DB_NAME : SQL 데이터베이스 이름
  - DB_USER : 데이터베이스 유저
  - DB_PASSWORD : 데이터베이스 유저의 비밀번호
  - DB_PORT : 데이터베이스에 접속할 때 사용하는 포트
- API Spec

Verb	Path	Parameter
GET	/user	/user?name=<name>
POST	/user	/user {“name”:”<name”, “password”:”<pw>”}
GET	/healthz	-

3. Container orchestration

- 제공된 어플리케이션을 EKS 1.27 환경에서 구동될 수 있도록 합니다.
- Cluster가 인터넷에서 접근 불가한 Private Endpoint Type을 사용하도록 합니다.
- Cluster에서 발생하는 모든 정보를 저장해 보안 감사 및 트러블슈팅에 이용할 수 있도록 합니다.
- 제공된 User 앱은 Karpenter를 이용하여 Node를 구성해 운영 하도록 합니다.
- EC2 Metadata 접근 시 인증 없는 API를 호출하지 않도록 합니다.
- 모든 Pod는 Worker Node EC2에 할당된 IAM role을 사용할 수 없어야 합니다.

4. Database

- 앱이 사용할 PostgreSQL Compatible 14.6를 구축 합니다.
- 아래 명령어를 참고하여 과제 수행에 필요한 적절한 테이블을 생성 하세요.

```
create schema dev;
CREATE TABLE dev.users (
  idx SERIAL PRIMARY KEY,
  name VARCHAR(255) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL
);
```

5. Operation

- 구축된 시스템에서 워크로드를 처리하며 실제 운영을 해야 합니다.
- Provisioning 후 User API 호출 가능한 Endpoint를 전달 해주세요.
- 과제 시작 한시간 후 부터 종료까지 계속 Traffic이 들어갑니다.
- 제공 되는 API들은 1초 내에 응답 해야 합니다.
- 앱의 코드상 버그로 인해 읽기 작업의 경우 늦게 응답 하는 케이스가 일부 존재합니다. 하지만 현재 버그 수정을 하기 어려운 상황임으로 적절하게 대처하여 운영하세요.

Automation

1. kube-host EC2

- Amazon Linux 2023 AMI 기반 EC2를 만들고 SSH 접근이 가능하도록 구성하세요.
- Kubectl이 설치되어 있어야 합니다.
- read01, admin01 이라는 Linux 계정을 생성 합니다.
- bastion의 모든 유저는 assume role 없이 kubectl 실행 시 cluster에 접근 불가해야 합니다.

2. EKS access management

- read01 계정은 평소에는 EKS 접근이 불가 하지만, "read" IAM role로 assume 후에는 kubectl 을 통해서 Pod의 리스트를 볼 수 있습니다. 하지만 Pod를 삭제할 수는 없어야 합니다.
- admin01 계정은 평소에는 EKS 접근이 불가 하지만, "admin" IAM role로 assume 후에는 kubectl을 통해서 Pod 리스트 확인 및 삭제를 할 수 있습니다.

3. IP management

- User API를 제공하는 동일한 Endpoint에 /allowlist API가 호출 가능 하도록 구성 합니다.
- /allowlist?ip=<ip> API 호출 시 <ip>:2222 tcp ingress 룰이 kube-host의 Security Group에 추가 되어야 합니다. 해당 룰은 추가된 후 15분 뒤에 자동으로 삭제 되어야 합니다.