|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **직 종 명** | Cloud Computing | **과 제 명** |  | **과제 번호** | 제1과제 |
| **경기시간** | 4시간 | **비 번 호** |  | **심사위원**  **확 인** | (인) |

2022년 클라우드컴퓨팅 친경경기대회

1. **과제 개요**  
   당신은 현재 Worldskills International에서 DevOps 엔지니어로 근무중입니다. 하지만 곧 개최될 예정인 제46회 상하이 국제기능올림픽 준비를 위해 필요한 인프라를 구성하고자 합니다. 하지만, 코로나바이러스의 여파로 인하여 모든 직원들이 재택근무를 시작했습니다. 재택근무를 하는 직원들의 원활한 재택근무를 위해 AWS 내부망에 접근할 수 있는 VPN 구축 및 각 나라의 대표 선수 등록 및 조회할 수 있는 서비스를 각 나라의 운영본부만 접근이 가능하도록 구성합니다.

A picture containing graphical user interface

Description automatically generated

1. **Software Stack**

- AWS - Application

- VPC - Go / net/http

- ECS

- EKS

- S3

- CloudWatch

1. **주의 사항**
   * 기계 및 공구 등의 사용 시 안전에 유의하시고, 필요 시 안전장비 및 복장 등을 착용하여 사고를 예방하여 주시기 바랍니다.
   * 작업 중 화상, 감전, 찰과상 등 안전사고 예방에 유의하시고, 공구나 작업 도구 사용시 안전보호구 착용 등 안전 수칙을 준수하시기 바랍니다.
   * 작업 중 공구의 사용에 주의하고, 안전수칙을 준수하여 사고를 예방하여 주시기 바랍니다.
   * 경기 시작 전 가벼운 스트레칭 등으로 긴장을 푸시고, 작업도구의 사용 시 안전에 주의하십시오.
   * 선수의 게정에는 비용제한이 존재하며 이보다 더 높게 과금 될 시 계정 사용이 불가능할 수 있습니다.
   * 문제에 제시된 괄호 박스 <>은 변수를 뜻함으로 선수가 적절히 변경하여 사용해야 합니다.
   * 문제의 효율을 위해 Security Group의 80/443 outbound는 any open하여 사용할 수 있도록 합니다.
   * Bastion EC2는 채점 시 사용되기 때문에 종료되어 불이익을 받지 않도록 주의해주시기 바랍니다.
   * 비밀번호가 따로 명시되지 않을 경우 "Skill53##"을 사용합니다.
   * 모든 Resource 및 Instance의 Instance Type은 "t3.small"을 사용합니다.
2. **배포 자료**

* Certificates/
  + server.crt (Self-Signed Certificate)
  + server.key (Self-Signed RSA Private Key)
  + ca.crt (Self-Signed Chain)
  + Client Certificates/
    - client.wsc-france.internal.crt (Client Certificate)
    - client.wsc-france.internal.key (Client RSA Private Key)

1. **배포 자료**

* Certificates/
  + server.crt (Self-Signed Certificate)
  + server.key (Self-Signed RSA Private Key)
  + ca.crt (Self-Signed Chain)
  + Client Certificates/
    - client.wsc-france.internal.crt (Client Certificate)
    - client.wsc-france.internal.key (Client RSA Private Key)

1. **Task (Worldskills)**해당 섹션에서 진행하는 모든 서비스 및 리소스는 ap-northeast-2리전에서 구성합니다.

**Networking**

각 Subnet의 이름 맨 뒤 알파벳은 각 Subnet이 위치할 가용영역을 뜻합니다. 또한 아래의 표를 참고하여 VPC를 구성합니다.

|  |  |  |
| --- | --- | --- |
| **Name** | **CIDR** | **Service Type** |
| wscf-manage.vpc | 10.105.0.0/16 | VPC |
| wscf-manage.pub-a | 위 VPC CIDR의 101번째 Network 주소 | Subnet |
| wscf-manage.pub-c | 위 VPC CIDR의 103번째 Network 주소 | Subnet |
| wscf-manage.priv-a | 위 VPC CIDR의 201번째 Network 주소 | Subnet |
| wscf-manage.priv-c | 위 VPC CIDR의 203번째 Network 주소 | Subnet |

또한 아래의 표를 참고해 DHCP 서비스를 구성합니다.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Domain** | **DNS Server** | **NTP** |
| wsc-dhcp-ops | wsc-france.internal | Route53의 DNS 서버 주소 | Amazon Time Sync Service |

**Client Virtual Private Network (VPN)**

현재 재택근무중인 직원들이 VPC 내부에 접근하기 위해 Client VPN을 사용해 접근이 되도록 구성합니다.

* + 해당 Client VPN의 이름은 "wscf-client-vpn"으로 지정합니다.
  + Client VPN은 wscf-manage.vpc에 생성된 모든 Private Subnet만 접근이 가능해야합니다.
  + 해당 VPN은 UDP 1194번로 Listen되도록 구성합니다.
  + 배포파일로 제공된 client 인증서를 사용하여 사용자 인증이 가능하도록 구성합니다.
  + 해당 VPN을 통해 접근한 Client들에게 부여되는 IP는 대역은 "172.29.8.0/22" 대역을 사용합니다.
  + 해당 VPN에 접근하면 "Worldskills Lyon 2024 is comming soon"라는 Login Banner 메시지가 출력되도록 구성합니다.
  + 해당 VPN의 채점은 Linux 환경에서 진행합니다.
  + VPN으로 접근 시 내부 DNS 주소에 DNS Query가 가능해야합니다.
  + Client VPN에 접근하기 위한 발급 된 Client Configuration(확장자 .opvn) 파일은 S3 Bucket에 /vpn-config/ 라는 경로에 client.opvn 이라는 이름으로 업로드 되어 있어야합니다.

**Control Host**

쿠버네티스를 원활하게 운영 및 관리를 하기 위해 Control Host를 구축합니다.

* + 해당 Instance의 Name Tag는 "wscf-manage-control-host"로 지정합니다.
  + 해당 Instance의 인스턴스의 IP주소는 10.105.202.19/24를 사용합니다.
  + 해당 Instance의 OS는 Amazon Linux 2를 사용합니다.
  + 기본적으로 awscliv2, kubectl, eksctl, jq, curl를 미리 설치합니다.
  + 해당 Instance는 Private Subnet, AZ=ap-northeast-2c에 위치하도록 구성합니다.
  + 해당 Instance는 VPN을 통해 들어온 사용자만 접근이 가능하도록 구성합니다.
  + SSH 접근할 때 사용하는 포트 번호는 "30245"번을 사용하도록 구성합니다.
  + 해당 Instance는 Password를 사용하여 SSH를 통해 접근이 되도록 구성합니다.
  + 해당 Instance에서 kubectl, eksctl과 같은 명령어가 정상 동작이 되어야합니다.
  + 해당 Instance에서 kubectl을 사용하여 Resource들이 배포 및 삭제 등의 작업이 가능해야 합니다.

**Load Balancer**

웹 서버를 운영하면서 생기는 트래픽에 대해 처리하기 위해 로드벨런싱을 구축합니다.

* 사용자가 접근하면 아래와 같은 방식으로 라우팅이 되어야합니다.  
  End User ->Network Load Balancer -> Application Load Balancer -> Web Server

**Network Load Balancer**

* + - NLB의 이름은 "wscf-front-nlb"로 지정합니다.
    - 해당 Load Balancer의 scheme는 "internal"로 지정합니다.
    - Application Load Balancer를 Routing 타겟으로 설정합니다.
    - 해당 NLB는 80번을 리슨 하도록 구성합니다.

**Application Load Balancer**

* + - ALB의 이름은 "wscf-back-alb"로 지정합니다.
    - 해당 Load Balancer의 scheme는 "internal"로 지정합니다.
    - 해당 ALB는 80번을 리슨 하도록 구성합니다.
    - Kubernetes를 사용해 운영중인 API 서버를 Routing 타겟으로 설정합니다.

**Load Balancer Security**

* + - NLB 통하지 않고 직접적으로 ALB의 /users에 접근 할 경우 403 상태코드와 함께 "Error: Your not Admin" 이라는 message를 출력합니다. 단, wget 명령어를 통해 ALB의 /users에 접근하면 500 상태코드와 "Can't Access using VPN"라는 message를 출력합니다.

**Container Orchestration**

컨테이너를 배포, 관리, 확장, 네트워킹을 자동화하기 위해 Container Orchestration을 구성합니다. 또한 선수는 이미 배포된 Image를 가지고 Kubernetes를 활용해 API Service를 운영할 줄 알아야하므로, 아래의 요구사항을 참고하여 구성하시길 바랍니다.

* + EKS Cluster의 이름은 "wscf-eks-cluster"로 지정합니다.
  + Kubernetes의 버전은 "1.21"을 사용합니다.
  + Node Group의 이름은 "wscf-worker-ng"로 지정합니다.
  + OS는 Amazon Linux 2를 사용합니다.
  + 해당 Node들의 Name Tag는 "wscf-worker-node"로 지정합니다.
  + 해당 Cluster는 외부에서 접근이 불가능하도록 구성합니다.

**Kubernetes**

미리 컨테이너 이미지 형태로 배포된 API 서비스를 쿠버네티스 환경에 배포합니다.

* + API를 운영하는 Pod는 최소 2개 이상 운영이 되어야합니다.
  + 모든 리소스들은 wscf-api라는 namespace에 생성되어야합니다.
  + 모든 리소스들의 이름은 wscf-<resource>로 지정합니다.  
    ex) deployment 생성시 "wscf-deployment"로 지정

**Application Description**

* + Image URL: jeonilshin/wsi-skills (Docker Hub)
  + Listen Port: 8080
  + Log: 해당 Application은 Log가 따로 저장되지 않습니다. 하지만 Log들은 STDOUT과 STDERR 형식으로 출력됩니다.
  + Sample API Requests

|  |
| --- |
| **User를 넣는 명령어**  $ curl -XPOST -d '{"nickname": "player101", "email": "admin@samsung.local"}' localhost:8080/users  **user를 조회하는 명령어**  $ curl localhost:8080/users |

**Domain Name Service**

내부에서 운영 중인 서비스에 내부 도메인을 사용해 접근을 하기위해 DNS 서비스를 구축합니다.

* + Doamin 주소는 "wsc-france.internal"로 지정합니다.
* 해당 Domain은 해당 섹션에서 생성한 VPC에 생성된 모든 서비스 및 리소스들만 질의가 가능하도록 구성합니다.
* 아래의 표를 참고하여 Record를 구성합니다.

|  |  |
| --- | --- |
| Record name | Value |
| control | 10.105.202.19 |
| api | <NLB의 주소> |

**Storage Service**

여러가지 파일들을 효율적으로 관리하기 위해 S3 Bucket을 생성해 관리하기이해서 S3 Bucket을 구성합니다.

* + S3 Bucket의 이름은 "wsc-s3-bucket-<자신이 AWS Account ID>"로 지정합니다.

**Application Logging**  
애플리케이션에서 발생하는 log들을 효과적으로 관리하기 위해서 CloudWatch를 사용해 중앙 집중 식 관리이 가능하도록 구성합니다. Kubernetes의 모든 Pod에서 사용자가 요청한 API에서 발생하는 모든 Log들을 CloudWatch Logs에 아래의 정보를 참고하여 구성하세요.

* + - Log Group: /aws/wsc/france/
    - Stream: wscf-deployment

1. **Task (Developer)**
   * 기본 VPC를 생성 후 Public Subnet에 EC2 Instance를 생성합니다.
   * 해당 Instance의 Name Tag는 "developer-online-client"로 지정합니다.
   * 해당 Instance의 Type은 t3.micro로 하며 OS는 Amazon Linux 2를 사용합니다.
   * 해당 Instance는 재부팅 하더라도 공인 IP가 바뀌어서는 안 됩니다.
   * 해당 Instance에게는 PowerUser 권한을 부여합니다.
   * awsv2, jq를 미리 설치합니다.
   * 직렬 콘솔를 통해서 접근이 가능하도록 구성합니다.
   * ec2-user 유저에게 "Skill53##"이라는 비밀번호를 부여합니다.
   * VPN이 연결되어 있지 않은 상태에서는 Worldskills 항목에서 구축한 서비스에 접근이 되어서는 안 됩니다.