**2조 프로젝트 계획서**

**제출일: 2024 – 03 - 20**

|  |  |  |  |
| --- | --- | --- | --- |
| 프로젝트명 | IaC 클라우드 및 DevOps 서비스 자동화  ( AWS 3-Tier Architecture ) | | |
| 구분 | 이름 | 연락처 | 역할 |
| 팀원 | \*조재형 | 010-8962-7307 | 계획서 작성, Terraform CloudWatch ,  Ansible Prometheus / Grafana 을 통한 EC2 모니터링 구축 |
| 장재영 | 010-7543-4343 | Terraform RDS, S3 구축, Ansible Docker 및 Jenkins 서비스 설치 |
| 김근보 | 010-2314-4644 | 구성도 작성, Terraform AutoScaling / ELB 설치 및 테스트, 발표ppt 작성 및 정리 |
| 최기운 | 010-6374-8033 | Terraform AWS EC2, VPC 통합 테스트, Web Application 구축, 프로젝트 발표 |
| 프로젝트 주제 | 1. IaC를 이용한 클라우드 구축 및 DevOps 서비스 구성 자동화  2. Terraform 이용 클라우드 인프라 구축  3. Terraform/Ansible 이용한 모니터링 시스템 구성 | | |
| 선정이유 | 1. 인프라를 코드로 관리하여 유지보수와 재사용성을 향상시키는 IaC의 중요성 강조  2. 클라우드 인프라 구축에 Terraform을 활용하여 자동화된 프로비저닝의 효율성 강조  3. Terraform과 Ansible을 통한 모니터링 시스템 구축으로 인프라 및 서비스 상태를 실시간으로 관찰하여 안정성 확보  4. DevOps 서비스 구성을 통해 개발과 운영 간의 협업과 소통 강화하여 개발 생산성 향상에 기여 | | |
| 프로젝트  기간 | 2024.03.20 ~ 03.27 : Project | | |
| 프로젝트 목표 | 1. IaC를 통한 클라우드 인프라 구축 및 관리 방법 습득하여 협업에 활용 가능한 수준의 이해도 획득 2. On-premises환경에서 클라우드 인프라를 구축하는 과정을 경험하여 이해도를 높임 3. AWS 리소스를 활용하여 클라우드 환경에 대한 심층적인 이해도 확보 4. Git을 활용한 형상 관리를 통해 팀 내 협업 효율성 향상 5. 로깅 및 모니터링 시스템에 대한 이해를 통해 서비스 상태를 실시간으로 파악하는 능력 강화 6. 팀원 간 협업 역량 및 커뮤니케이션 능력을 강화하여 프로젝트 진행의 효율성을 높임 7. 3-Tier 아키텍처를 기반으로 한 인프라 구축 및 서비스 배포를 통해 실전 프로젝트 경험을 쌓음 | | |
| 프로젝트 내용 및 범위 | 1. IaC를 이용한 클라우드 인프라 구축  2. DevOps 서비스 구성  3. 모니터링 관리 시스템 구성 | | |

**작업 내용**

* 1. **네트워크 생성** 
     1. 2개 이상의 AZ에 각각의 Public subnet 1개, Private subnet 3개(Web, WAS, DB) 포함
     2. pubic subnet의 bastion을 통해 Web, WAS에 접속, 유지 보수
     3. private instance 들은 public subnet의 Nat Gateway를 통해 외부로 통신
     4. 3가지의 subnet을 각각 묶어 3 Tier layer 구성
     5. external-ELB를 통해 사용자와 presentation layer 사이 연결 및 Web instance의 스트레스 관리
     6. internal-ELB를 presentation layer와 application layer 연결 및 WAS의 스트레스 분산
     7. DB subnet 들은 RDS subnet group으로 묶어 별도의 database layer 구성
  2. **instance 생성 및 관리** 
     1. Web instance에서 nginx 설치 및 구동
     2. Web instance의 부하도에 따라 auto scaling을 통해 스트레스 분산
     3. RDS MySql을 통해 database 생성
     4. Master, StandBy database를 통해 안정성 확보
  3. **보안 설정** 
     1. security group의 inbound source 제한을 통해 트래픽 관리
     2. IAM role을 통해 리소스 간의 역할과 권한을 관리
  4. **로그 수집 및 분석** 
     1. Cloud Watch를 통해 서버와 애플리케이션 발생 로그 수집
     2. Prometheus / Grafana 를 통한 로그 수집
     3. 각 로그 수집 및 분석을 통한 모니터링
  5. **app 개발 및 배포** 
     1. Amazon linux 2023 기반 Web instance
     2. Python + Flask Framework를 통한 Application 개발
     3. RDS MySql instance 로 DB layer 구축
     4. 3 Tier 간의 연동

**구성도**



