**최종 프로젝트 주제 선호도 조사지**

**1반 김건호**

**■ 최종 프로젝트 1지망(필수)**

|  |  |
| --- | --- |
| **주제명** | **마이크로서비스 기반 웹 애플리케이션 환경 구축** |
| **주제**  **선정**  **사유** | 슈퍼앱에 마이크로서비스가 가장 적합하다고 생각하여서 선정하였습니다.  현재 디지털 환경에서 슈퍼앱의 마이크로서비스 아키텍처 적용은 전 세계적으로 확산되고 있는 주요 트렌드 중 하나입니다. 이는 여러 서비스를 유기적으로 통합하면서도 각각의 서비스를 독립적으로 운영할 수 있게 하는 기술적 유연성 때문입니다. 특히 금융, 소매, 여행 등 다양한 산업에서 슈퍼앱은 소비자에게 다양한 서비스를 손쉽게 제공하면서도, 각 서비스의 개별적인 혁신을 지속적으로 추진할 수 있는 구조를 요구합니다.  마이크로서비스로 구성하여 소비자들은 여러 서비스를 하나의 통합된 플랫폼에서 접근할 수 있도록 하는 슈퍼앱의 이점을 누릴 수 있습니다. 각각의 서비스가 독립적으로 운영되면서도 전체 시스템의 부담 없이 빠르게 확장하고, 신속하게 개선 사항을 적용할 수 있습니다.  aws에서 데이터베이스, 보안, 모니터링 시스템 등과 같은 다양한 AWS 리소스와 원활하게 연동할 수 있게하면서 슈퍼앱의 기능을 확장하고 강화하며, 안정적인 서비스 운영을 가능하게 할 수 있을것입니다.  비용 측면에서도 사용한 만큼만 요금을 지불하는 구조를, 만들어 불필요한 리소스 낭비를 줄이며 운영 비용을 최적화시키고 시장 변화에 빠르게 대응하여 사용자에게 지속적으로 개선된 서비스를 제공할 수 있게합니다.  해당 장점들이 슈퍼앱이 디지털 환경에서 경쟁력을 유지하고 성장하는 데 결정적인 역할을 할 것이라고 생각하여 선정하였습니다. |

**■ 최종 프로젝트 2지망(필수)**

|  |  |
| --- | --- |
| **주제명** | **Data Analytics 및 CI/CD 환경의 컨테이너 기반 아키텍처 설계 및 구현** |
| **주제**  **선정**  **사유** | 데이터 분석을 통해 얻은 통계를 기반으로 CI/CD 환경을 구축하고 관리함으로써 개발 및 배포 프로세스를 지속적으로 향상시키는 것이 목표입니다. 사용자의 행동 및 선호도를 분석하여 개인화된 서비스를 제공하고, 신속하게 변경 및 업데이트될 수 있을 것입니다.  데이터 기반의 의사 결정을 통해 애플리케이션의 기능 및 성능을 지속적으로 최적화하고, 사용자 경험을 개선하여 시장에서의 경쟁력을 강화하는 것이 최종 목표입니다.  이를 위해 AWS의 EKS를 활용하여 컨테이너 기반의 아키텍처를 구축하고, 데이터 분석 결과를 바탕으로 자동화된 배포 프로세스를 구현할 것입니다.  데이터 분석을 통해 얻은 인사이트를 CI/CD 파이프라인에 통합하여, 개발자들이 데이터 기반의 의사 결정을 내릴 수 있도록 지원할 것입니다. 이를 통해 애플리케이션의 기능이나 성능에 대한 변경사항을 신속하게 반영하고, 사용자들에게 개인화된 경험을 제공하여 시장에서의 경쟁력을 강화할 것입니다. |

**■ 구성도 및 목적(필수)**

|  |  |
| --- | --- |
| **구성도** |  |
| **구성**  **목적** | React + Springboot로 구성하기 위해서 다음과 같은 구성도를 만들었습니다.  3tier 구성과 CI/CD, 모니터링을 기본사항으로 구성하였습니다.  기본 3tier 구성   * Route 53: DNS 서비스를 제공하여 사용자가 애플리케이션에 액세스할 수 있는 도메인 이름을 관리하고 트래픽을 애플리케이션으로 라우팅 * WAF : 웹 애플리케이션을 보호하기 위해 특정 HTTP 트래픽을 필터링하고, 보안 위협으로부터 애플리케이션을 방어 * Shield: DDoS 공격으로부터 리소스를 보호하고, 가용성과 성능을 유지 * NAT 게이트웨이: 프라이빗 서브넷의 인스턴스가 인터넷과 통신할 수 있게 하며, 인바운드 직접 액세스는 방지 * ALB : 들어오는 웹 트래픽을 web,was 서버별로 각각 분산시켜 부하를 관리 * RDS : 관계형 데이터베이스를 쉽게 설정, 운영, 스케일링, 데이터의 안정성과 스케일링을 자동화 * Session Manager: 시스템 관리자가 AWS 리소스에 안전하게 접근하고 관리 * IAM : 사용자, 서비스, 리소스에 대한 액세스 권한을 관리하고, 안전한 인증 및 권한 부여를 수행 * S3 : 데이터를 안전하게 저장하고 접근할 수 있는 스토리지를 제공   CI/CD 구성   * CodePipeline : 소스 코드에서부터 배포까지 전체 배포 프로세스를 자동화하고 관리 * CodeCommit : 안전한, 확장 가능한 소스 컨트롤 서비스로 코드를 저장하고 버전 관리 * CodeBuild : 소스 코드 컴파일, 테스트 실행, 패키징 등 빌드 및 테스트 프로세스를 자동화 * CodeDeploy : 자동화된 배포 도구로 애플리케이션을 컴퓨팅 서비스에 배포   모니터링 구성   * CloudWatch : 애플리케이션 및 AWS 리소스의 성능을 모니터링하고 로깅 * Alarm : CloudWatch 알람을 설정하여 정의된 임계값을 초과하는 경우 알림을 받음 * SNS (Simple Notification Service) : 알람이나 이벤트에 대해 사용자에게 알림을 보낼 수 있도록 함 * Lambda : 이벤트 기반의 코드 실행을 관리하여 자동화된 작업을 수행 * Slack : 모니터링 시스템의 알림을 실시간 채팅으로 전달하여 공유 * Parameter Store: Lambda 함수가 특정한 조치를 취하기 전에 필요한 구성 값을 Parameter Store에서 불러옴 |