

IaC 기반 인프라 자동화 및 컨테이너 환경 구축

이름 이정필

지도교수 이환용

연구배경

현대 기술 환경에서 애플리케이션 배포는 개발자의 필수 역량 중 하나로 자리잡고 있습니다. 그럼에도 불구하고, 많은 신입 개발자와 심지어 경험이 많은 개발자들조차도 배포 과정에 어려움을 겪는 상황을 흔히 목격하곤 합니다. 애플리케이션 코드의 경우, 버전 관리 시스템을 통해 체계적으로 관리되는 반면, 인프라 구성은 주로 직관적이지 않은 UI 콘솔을 통해 이루어지며, 이는 개발자 개인의 경험치에 크게 의존하게 됩니다.

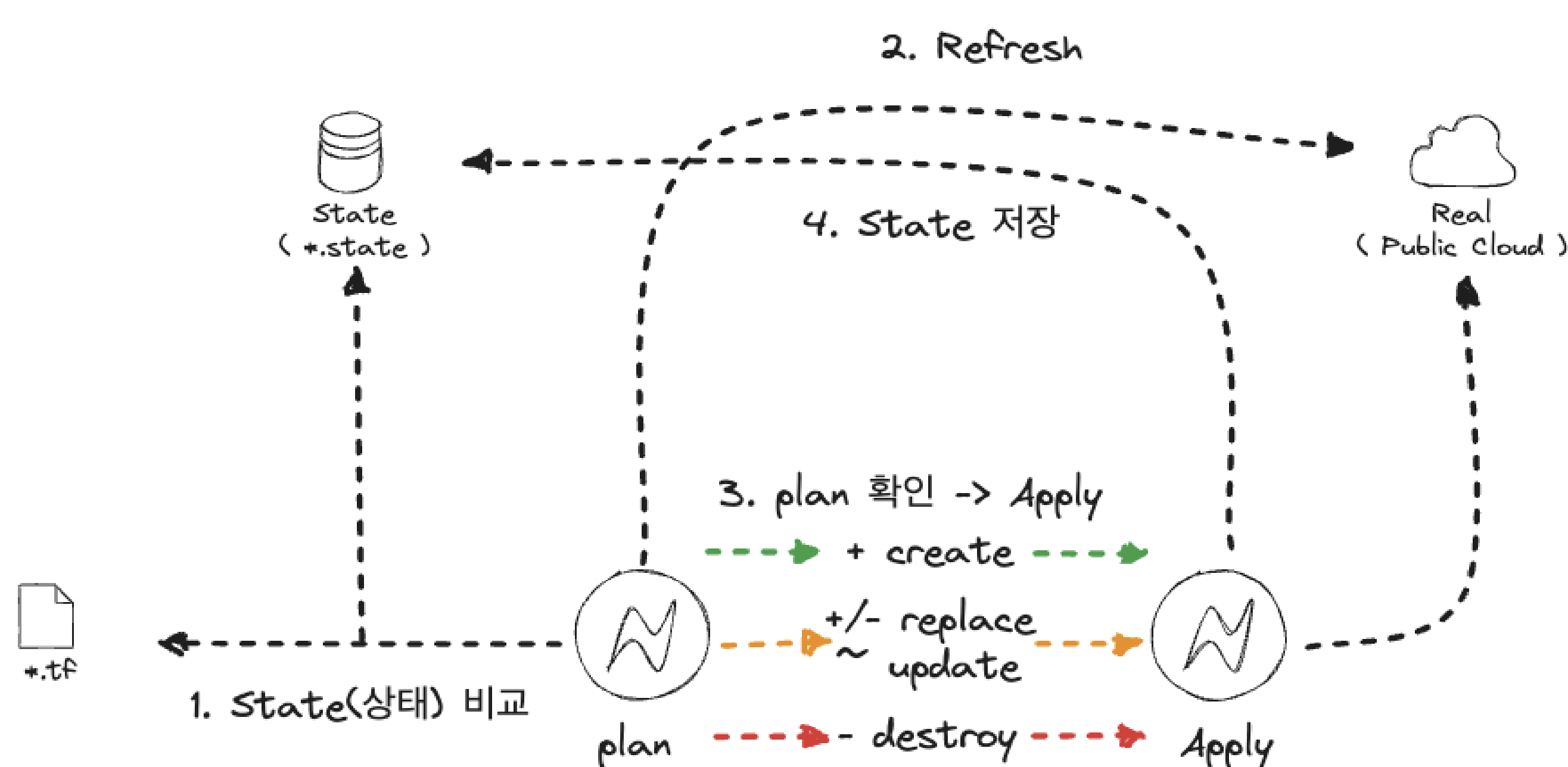
이에 본 프로젝트는 개발자들이 인프라 구성의 어려움 없이, 템플릿을 활용하여 신속하고 효율적으로 개발 환경을 구축할 수 있는 방법을 제시하고자 합니다. 이는 특히 클라우드 서비스에 대한 초기 접근성 문제를 겪는 국내외 수많은 개발자들에게 실질적인 도움이 될 것으로 예상됩니다.

2022년 1학기 소프트웨어 캡스톤 디자인 과정을 수강하며, 우수한 기능을 갖춘 프로젝트가 인프라 설정의 복잡함으로 인해 배포에 실패하는 사례를 종종 목격하였습니다. AWS, GCP, NCP와 같은 주요 클라우드 서비스 플랫폼들이 제공하는 기능적 장점에도 불구하고, 실제로 이를 효과적으로 활용하지 못하는 개발자들이 많다는 점이 문제로 대두되었습니다.

이 프로젝트는 이러한 문제에 대응하기 위해, 인프라 구축의 장애를 최소화하고 기능 배포의 용이성을 극대화하는 방안을 모색합니다. 이를 위해 현대 개발 팀에서 널리 채택하고 있는 GitHub을 통한 소스코드 형상 관리 방법을 도입할 것입니다. GitHub을 통해 코드 관리뿐만 아니라, 인프라의 코드화(Infrastructure as Code)를 통한 효율적인 배포 흐름을 구축할 계획입니다.

이 프로젝트를 통해, 개발자들은 더 나은 인프라 환경 구축과 함께 기능의 신속한 배포를 실현할 수 있을 것으로 기대됩니다.

결과 및 분석



Terraform 코드를 작성하고 git을 이용한 형상 관리를 통하여 인프라의 변경 이력을 체계적으로 관리할 수 있습니다. 조직 내에서의 운영 및 코드 리뷰 프로세스를 도입하여 업무 효율성을 향상시킬 수 있는 구조적 접근법을 제공합니다.

EC2, ECR, RDS, ALB와 같은 퍼블릭 클라우드 리소스 기반의 모듈을 제공함으로써, 인프라에 대한 기초적인 이해가 있는 개발자가 스켈레톤 코드를 편집하여 리소스를 쉽고 빠르게 프로비저닝 할 수 있는 구조를 제공합니다.

오픈소스 URL

<https://github.com/mrlee7/infra-builder>

연구진행과정

1. 목표 정의

인프라 구성에 익숙하지 않은 개발자들이 인프라를 쉽고 빠르게 프로비저닝 할 수 있는 스켈레톤 코드 제공 및 튜토리얼 제공을 목표로 합니다.

2. 유저 인터뷰 진행

- 인프라스트럭처 구성 경험에 대해 응답자의 60% 이상이 초보자 (ex. 몇 번 구성해보았으나, 어려움을 겪음) 라고 응답
- 인프라 구성 시 어려움을 겪는 부분에 대해 응답자의 60% 이상이 인프라스트럭처 구성의 복잡성 때문이라고 응답
- 현재 사용중인 Infra 서비스 (AWS 기준)을 EC2, RDS, S3, Lambda, ALB 순이라고 응답
- Terraform 등 IaC 사용 경험에 대해 응답자 70%가 전혀 없음이라고 응답

3. 필요 리소스 및 IaC 레퍼런스 검토

설문 기반으로 모듈 개발이 필요한 리소스 정의 및 infra-builder 패키지 구조를 설계하였습니다. aws-cli 설치 및 Docker 환경 구성과 같은 로컬 개발 환경 구성 이후 쉽고 빠르게 인프라 배포를 진행할 수 있는 구조를 작성하였습니다.

4. ECS Fargate 기반 인스턴스에 postfix활용하여 SMTP 서버구축하여 PoC 진행

컨테이너 환경 기반에서, SMTP 서버 구축을 통해 PoC(Proof Of Concept)를 진행하였습니다. VPC, ELB, ECS Fargate, Cloudwatch 리소스를 프로비저닝 한 뒤 로그를 통해 이메일 발송 요청에 따른 수신이 정상 동작함을 확인하였습니다.

5. Amazon ECR, Amazon EC2, Amazon ALB, VPC 구성을 위한 모듈 작성

2번 항목의 유저 인터뷰를 통해 도출된 리소스 사용 빈도를 기준으로 필요 모듈을 설정하였습니다. Infra-builder에 각 모듈의 패키지 구조 및 코드를 작성하고, 프로젝트 별 Docker 파일을 작성한 뒤, 배포가 가능한 구조를 개발하였습니다.

인프라 배포가 진행되지 않은 사이드 프로젝트 팀에 참여하여, 인프라 설계 및 구성을 완료한 뒤 프로비저닝을 진행하였습니다.

6. 문제점과 해결책 모색

인프라 구성의 복잡성, IaC 작성을 위한 코드 작성에서의 오류 발생 및 낮은 인프라 이해도로 인한 유지보수 관리 어려움이 여전한 문제점으로 남아있습니다.

문서화 및 튜토리얼 제공으로 인프라 구성에 진입장벽을 낮추고 로깅 및 모니터링 시스템 강화, 유저 피드백을 수집하여 리소스 모듈을 오픈소스 형태로 업데이트하는 것을 해결책으로 제안합니다.

7. 결론 및 추후 진행 방안

IaC를 통한 인프라 프로비저닝, 컨테이너 환경 구축을 통해 애플리케이션 개발자가 인프라 구성하는 과정에서의 진입장벽을 낮추게 되었습니다.

Terragrunt, 테라폼, 테스트 코드, Packer 와 같은 리소스를 도입하여 안정적이고 쉬운 인프라 구성이 가능하도록 개선합니다.

