JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 8, pp. 1-16, August 2026
https://doi.org/10.9708/jksci.2026.31.08.001

# A Study on Security-Enhanced Meta-Cloud Architecture

Sungkap Park*, Miyoung Bang*, Jiyun Kim*, Changhyun Kim*, Dongsu Lee**

*Ph.D, Dept. of Digital Management, Graduate School, Korea University, Sejong, Korea
**Professor, Corporate Network Center Director, Gachon University, Incheon, Korea

## [Abstract]

Multi-cloud environments improve resource utilization but also increase management complexity and security risks caused by configuration errors. This study proposes a security-enhanced meta-cloud architecture that integrates visual infrastructure design, Terraform-based Infrastructure as Code (IaC) generation, pre- and post-deployment security validation, and multi-cloud operations into a unified framework. The proposed architecture applies consistent security policies across AWS, Microsoft Azure, and Naver Cloud Platform (NCP), demonstrating consistent operation in heterogeneous cloud environments. Experiments involving 144 non-expert participants achieved an average architecture implementation completeness of 66.8%, while the automated security inspection process significantly improved operational efficiency compared with conventional manual approaches. The results demonstrate that the proposed architecture can effectively enhance both the security posture and operational efficiency of multi-cloud environments.

▸ **Key words**: Meta-cloud, Multi-cloud, Infrastructure as Code (IaC), Terraform, Security Automation

## [요 약]

멀티클라우드 환경은 자원 활용성을 높이는 반면 관리 복잡성과 설정 오류에 따른 보안 위험을 증가시킨다. 본 연구는 시각적 인프라 설계, Terraform 기반 IaC 자동 생성, 배포 전·후 보안 검증, 멀티클라우드 운영을 통합한 보안 강화 메타클라우드 아키텍처를 제안하였다. AWS, Azure 및 Ncloud 환경에서 동일한 보안 정책을 적용하여 일관된 운영을 확인하였으며, 비전문가 144명을 대상으로 한 실험에서 평균 66.8%의 아키텍처 구현 수준과 수동 방식 대비 향상된 운영 효율성을 확인하였다. 연구 결과는 제안한 아키텍처가 멀티클라우드 환경에서 보안성과 운영 효율성을 향상시킬 수 있음을 보여준다.

▸ **주제어**: 메타 클라우드, 멀티 클라우드, 코드형 인프라, 인프라 자동화 도구, 보안 자동화

• First Author: Sungkap Park, Corresponding Author: Dongsu Lee
*Sungkap Park (xman8590@kismi.kr), Dept. of Digital Management, Graduate School, Korea University
*Miyoung Bang (evalikesme@gmail.com), Dept. of Digital Management, Graduate School, Korea University
*Jiyun Kim (rapidprog@naver.com), Dept. of Digital Management, Graduate School, Korea University
*Changhyun Kim (super_midas@naver.com), Dept. of Digital Management, Graduate School, Korea University
**Dongsu Lee (leedongsu.kr@gmail.com), Corporate Network Center Director, Gachon University
• Received: 2026. 02. 09, Revised: 2026. 07. 14, Accepted: 2026. 07. 16.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

# I. Introduction

디지털 전환의 가속화로 다중 클라우드(Multi-Cloud) 환경이 보편화되고 있으나[2], 이기종 자원 및 도구의 혼재는 인프라 관리의 복잡성을 증대시키고 보안 설정 표준의 파편화를 야기하는 구조적 문제를 낳고 있다[1]. 특히 Gartner는 클라우드 보안 사고의 대부분이 공급자의 기술적 결함보다 사용자의 설정 오류(Misconfiguration)와 인적 오류(Human Error)에서 발생할 것으로 전망하였다. 이에 따라 설정 오류를 사전에 탐지하고 지속적으로 관리할 수 있는 보안 자동화 기술의 필요성이 더욱 강조되고 있다[10]. 인프라 구성이 고도화될수록 관리자의 수동 설정에 의존하는 기존 방식은 치명적인 취약점에 노출될 가능성이 매우 높다. 그러나 기존의 연구나 관리 프레임워크는 단순한 자원 관리 기능에 치우쳐 있거나 사후 대응 중심의 보안 구조에 머물러 있어, 설계 단계에서부터 보안을 내재화하거나 전체 수명주기를 유기적으로 통합하는 데 한계를 지닌다[6]. 또한, 숙련된 전문가에 의존하는 수동 보안 점검 방식으로는 대규모 다중 클라우드 환경의 복잡성을 제한된 시간 내에 실시간으로 감시하고 통제하는 것이 불가능한 실정이다[5].

최근 Meta-Cloud, Multi-cloud 및 Infrastructure as Code(IaC) 기반 자동화 기술에 관한 연구가 활발히 수행되고 있으나, 대부분의 연구는 클라우드 자원의 통합 관리 또는 자동 배포에 초점을 두고 있다[9]. 또한 보안 취약점 점검은 배포 이후 수행되는 사후 점검(Post-deployment)에 집중되어 있으며, IaC 작성 단계에서 보안 정책을 자동으로 검증하는 연구는 상대적으로 부족한 실정이다[4]. 특히 멀티클라우드 환경에서 보안 정책 검증, 인프라 자동 생성 및 배포를 하나의 프레임워크로 통합하여 검증한 연구는 제한적으로 보고되고 있다. 따라서 보안성과 운영 효율성을 동시에 고려한 통합 관리 프레임워크에 대한 연구가 필요하다[19].

이러한 한계를 극복하기 위해 본 논문에서는 시각적 인프라 설계와 자동화된 코드형 인프라(Infrastructure-as-Code(IaC)) 생성부터 배포 전후의 보안 검증 및 멀티클라우드 운영에 이르는 전체 수명 주기를 단일 프레임워크로 통합한 '보안 강화 메타클라우드 아키텍처'를 제안한다.

본 연구의 목적은 멀티클라우드 환경에서 발생하는 설정 오류와 관리 복잡성을 해결하기 위해 보안 검증을 설계 단계부터 운영 단계까지 통합한 Meta-Cloud 기반 CloudS 프레임워크를 제안하고 그 효과를 실증적으로 검증하는 것이다. 증가하고 있는 이기종 클라우드 자원의 관리 복잡성과 관리자의 설정 오류(Misconfiguration)에 기인하는 보안 문제를 해결하는 것을 주요 연구목적으로 한다. 기존의 멀티클라우드 관리 연구들은 주로 자원 통합 관리, 서비스 오케스트레이션 또는 배포 자동화에 초점을 두고 있으며, 보안 기능 역시 배포 이후의 사후 점검(Post-deployment Security) 중심으로 적용되는 경우가 대부분이다[6]. 따라서 인프라 설계 단계부터 보안 검증을 내재화하고, 설계·배포·운영을 하나의 프레임워크로 통합하는 연구는 상대적으로 부족한 실정이다.

기존 연구의 한계를 보완하기 위해 다음과 같은 연구문제를 설정하였다.

RQ1. Meta-Cloud 환경에서 보안 정책을 자동으로 검증할 수 있는 통합 프레임워크를 설계할 수 있는가?

RQ2. 제안한 CloudS는 기존 수동 보안 점검 방식보다 보안 점검 효율성을 향상시키는가?

RQ3. 제안한 CloudS는 AWS, Azure 및 Naver Cloud 환경에서 일관된 보안 검증 결과를 제공하는가?

RQ4. 제안한 CloudS는 멀티클라우드 환경에서 실무 적용 가능성을 확보할 수 있는가?

기존 연구들은 Meta-Cloud 관리, Infrastructure as Code(IaC) 기반 자동화, 보안 취약점 점검 및 멀티클라우드 운영을 각각 독립적인 연구 영역으로 다루어 왔다. 반면 본 논문은 이러한 요소들을 단일 프레임워크인 CloudS로 통합하여 설계 단계에서부터 보안 검증을 수행하고, 인프라 생성, 자동 배포 및 운영 관리까지 전체 수명주기를 일관되게 지원하는 통합 아키텍처를 제안한다. 또한 다양한 클라우드 서비스 제공자(CSP)에 동일한 보안 정책을 적용하고 이를 정량적으로 검증하였다는 점에서 기존 연구와 차별성을 가진다.

본 논문의 주요 기여는 다음과 같다.

첫째, Meta-Cloud 관리와 IaC 기반 보안 검증을 통합한 프레임워크를 제안하였다.

둘째, tfsec 기반 자동 보안 검증 절차를 멀티클라우드 환경에 적용하였다.

셋째, Security Compliance Score 기반의 정량 평가 방법을 제시하였다.

넷째, 실제 클라우드 환경을 대상으로 적용 가능성을 검증하였다.

본 논문에서는 제안한 CloudS 프레임워크의 유효성을 검증하기 위하여 AWS, Microsoft Azure 및 Naver Cloud Platform 환경에서 동일한 실험 시나리오를 적용하였다. 또한 tfsec 기반 보안 검증, Terraform 기반 자동 배포, 성능 평가 및 전문가 인터뷰를 수행하여 보안성,

운영 효율성 및 실무 적용 가능성을 종합적으로 검증하였다. 따라서 본 연구는 Meta-Cloud 기반 멀티클라우드 관리 분야에서 보안 자동화, IaC 기반 보안 검증 및 통합 운영 프레임워크를 하나의 구조로 통합적으로 제안하고 실증적으로 검증하였다는 점에서 기존 연구와 차별화되는 학술적·기술적 의의를 가진다[17].

## II. Theoretical Background

### 2.1 Concept and Characteristics of Meta-Cloud

메타클라우드는 다양한 클라우드 자원을 하나의 추상화 계층으로 통합하여 관리하는 구조로 정의 된다[9, 16]. 이는 '스카이 컴퓨팅(Sky Computing)'의 개념과 궤를 같이 하며, 사용자에게 이기종 클라우드 간의 상호운용성(Interoperability)과 자원 이동성을 보장한다[9]. 특히 복잡한 클라우드 API를 추상화함으로써 관리 효율성을 높이고, 특정 공급자에 대한 기술적 종속성을 완화할 수 있는 구조적 장점을 가진다.

본 논문의 메타클라우드 아키텍처는 단순한 멀티클라우드 관리 프레임워크를 넘어, 인프라 설계 단계부터 보안 검증, 배포 및 운영까지 전 과정을 통합하는 계층적 구조를 갖는다. 특히, 추상화 계층을 이용하여 이기종 클라우드 자원을 표준화하고, 이를 기반으로 IaC를 자동 생성함으로써 배포의 재현성과 일관성을 확보한다. 또한 보안 검증 기능을 설계 단계에 포함하여 배포 이전 단계에 내재화함으로써 기존 사후 대응 중심의 보안 구조를 개선하고, 설정 오류로 인한 취약점을 사전에 제거하는 것을 특징으로 한다.

### 2.2 Infrastructure as Code(IaC)

IaC는 인프라 구성 요소를 코드로 정의하여 버전 관리, 자동 배포, 재현성을 확보하는 핵심 기술이며, 최근에는 설계 단계에서 보안 요구사항을 이해하고 적용하기 위한 Security-by-Design 접근이 중요하게 강조되고 있다[24]. DevSecOps 환경에서 보안 자동화의 기반 기술로 활용된다. 수동 설정에 의존하던 기존 방식의 한계를 극복하기 위해 등장하였다. 테라폼(Terraform)과 같은 도구를 활용하여 인프라를 코드화함으로써 배포의 일관성과 재현성을 확보할 수 있다. 이는 특히 대규모 다중 클라우드 환경에서 설정 오류로 인한 보안 사고를 방지하고, 버전 관리와 자동화된 검증을 가능케 하는 핵심 기술 요소이다.

Table 1. Technical Classification and Summary of Characteristics of IaC

| Tool | Key Characteristics |
|---|---|
| Terraform | Declarative DSL (Domain-Specific Language) based on HashiCorp Language; excellent Multi-cloud support and modularity [22] |
| Ansible | Python-based automation tool; focuses on procedural scripting and configuration management [12] |
| AWS Cloud Formation | AWS-specific declarative tool; strong capabilities in security and template-based deployment [7] |
| Pulumi | Code-based IaC platform supporting general-purpose languages (e.g., JavaScript, Python) [7] |

### 2.3 Cloud Security and Compliance

클라우드 보안은 공급자와 이용자가 책임을 분담하는 '공동 책임 모델'을 기본으로 한다[14]. 최근에는 설정 오류에 의한 데이터 유출 사고가 급증함에 따라, 설계 단계부터 보안을 내재화하는 'Secure by Design'과 지속적인 규제 준수(Compliance) 확인이 강조되고 있다[5].

본 논문에서는 ISO/IEC 27017, CSA CCM 및 ISMS-P의 주요 통제 항목을 참조하여 보안 점검 기준을 구성하였으며, 각 규제에서 공통적으로 요구하는 접근통제, 암호화, 로깅 및 네트워크 보안 항목을 CloudS의 기본 보안 정책으로 적용하였다.

### 2.4 Intercloud Broker for Multi-Cloud Interoperability

클라우드 환경에서 최적의 자원을 배분하기 위한 인터클라우드 브로커는 서비스 품질(QoS), 비용, 보안 정책 등을 고려하여 워크로드를 배치하는 역할을 수행한다. 브로커는 각 CSP(Cloud Service Provider)의 서비스 가용성과 가격 모델을 실시간으로 분석하여 사용자에게 최적의 클라우드 조합을 권고하거나 자동 배치함으로써 자원 활용의 극대화를 지원한다.

Table 2. Major Cloud Regulations and Standards

| Regulation / Standard | Description |
|---|---|
| GDPR | Regulates the processing of personal data and controls cross-border data transfers |
| CCPA | California Consumer Privacy Act of the United States; protects consumer personal information and restricts data deletion and sale |
| ISO/IEC[1)]27017 | Security control guidelines for cloud service providers and cloud service customers[12] |
| ISO/IEC27018 | Code of practice for the protection of personally identifiable information (PII) in public cloud environments[13] |
| CSA CCM[2)] | Cloud Controls Matrix developed by the Cloud Security Alliance; a framework for cloud security controls covering security, compliance, and risk management |

### 2.5 Comparison with Existing Multi-Cloud Management Technologies

기존의 멀티클라우드 관리 기술은 각각 특정 기능에 초점을 맞추고 있다. Terraform은 Infrastructure as Code(IaC)를 통한 인프라 자동 배포 기능을 제공하지만 시각적 모델링 기능과 통합 보안 검증 기능은 제공하지 않는다[9]. tfsec은 Terraform 코드에 대한 정적 보안 분석 기능을 제공하지만 런타임 보안 감시와 멀티클라우드 운영 기능은 지원하지 않는다[11]. 또한 CSPM(Cloud Security Posture Management)은 운영 중인 클라우드 환경의 보안 상태를 지속적으로 모니터링할 수 있으나 인프라 설계 단계에서의 Security by Design 기능은 제한적이다[8].

반면 본 논문에서 제안하는 CloudS 플랫폼은 시각적 인프라 설계, Terraform 기반 IaC 자동 생성, 배포 이전의 정적 보안 분석, 배포 이후의 지속적인 보안 점검 및 멀티클라우드 통합 운영을 하나의 플랫폼에서 지원함으로써 기존 연구와 차별성을 갖는다[21].

Table 3. Functional Comparison between Existing Approaches and CloudS

| Item | Terraform | tfsec | CSPM | CloudS (Proposed) |
|---|---|---|---|---|
| Visual Modeling | ✗ | ✗ | ✗ | ✓ |
| IaC Generation | ✓ | ✗ | ✗ | ✓ |
| Static Security Scan | ✗ | ✓ | ✗ | ✓ |
| Runtime Monitoring | ✗ | ✗ | ✓ | ✓ |
| Multi-cloud Management | △ | ✗ | ✓ | ✓ |
| Drift Detection | ✗ | ✗ | ✓ | ✓ |
| Cost Analysis | ✗ | ✗ | △ | ✓ |
| Security by Design | ✗ | ✗ | △ | ✓ |
| Integrated Workflow | ✗ | ✗ | ✗ | ✓ |

## III. Design of Metacloud Infrastructure Automation

### 3.1 Design Principles and Architecture Overview

본 논문에서 제안하는 시스템은 이기종 클라우드 자원을 효율적으로 통합하고 자동화하기 위해 설계되었으며, 선언적 구성 및 멱등성 보장 등을 핵심 설계 원칙으로 채택하고 있다[15]. 제안 시스템은 사용자가 원하는 목표 상태만 정의하면 플랫폼이 이를 자동으로 처리하는 구조를 가지며, 동일한 코드를 반복 실행해도 일관된 결과를 생성함으로써 인적 오류 발생 가능성을 감소시킨다[17].

시스템은 느슨한 결합(Loose Coupling) 원칙을 기반으로 3개의 논리 계층으로 설계하였다.

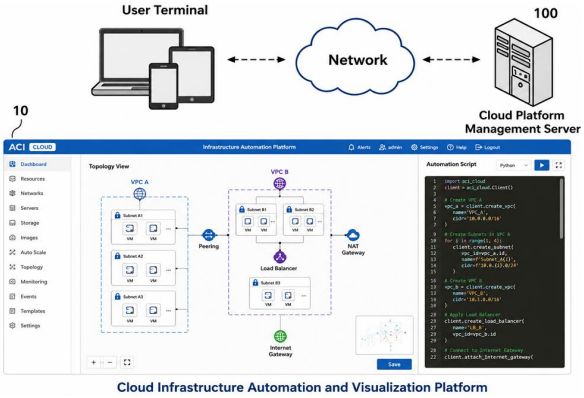


Fig. 1. Cloud Infrastructure Automation and Visualization

### 3.2 Detailed Layered Architecture of Meta-Cloud

첫째, 프레젠테이션 계층은 Drag & Drop 방식의 웹 콘솔을 통해 시각적 인프라 설계를 지원하며, IaC 코드와 GUI와 IaC 간 양방향 동기화 및 역공학 기능을 지원한다. 둘째, 제어 계층은 워크플로우 엔진과 정책 엔진(OPA)을 통해 사용준비 작업의 순서 및 의존성을 제어하고, 비용 기반 최적화 알고리즘을 실행하여 보안 정책 위반 여부를 사전에 검증한다[10]. 셋째, 통합 및 실행 계층은 각 클라우드 사업자(CSP)별 어댑터를 통해 이기종 API 호출을 추상화하며, 테라폼(Terraform) 기반의 실행 엔진을 통해 실제 자원의 생성, 수정, 삭제를 수행한다.

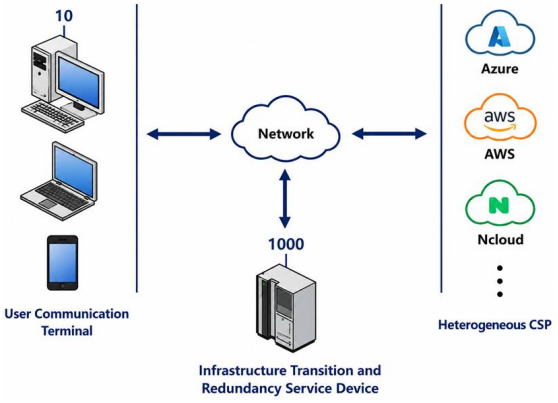


Fig. 2. Cloud Infrastructure Migration and Redundancy

또한, 데이터의 무결성과 보안을 위해 데이터 및 보안

1) ISO/IEC: International Organization for Standardization/International Electrotechnical Commision

2) CSA CCM: Cloud Security Alliance의 Cloud Controls Matrix

계층이 핵심적인 역할을 수행한다. 시스템은 상태 파일(State File)의 원격 저장 및 잠금 메커니즘을 적용하여 실제 인프라와 코드 간의 불일치(Drift)를 방지하며, 제로 트러스트(Zero Trust) 모델을 기반으로 신원 인증 및 접근 제어를 통합 관리한다[22]. 특히 비밀 관리 시스템(Vault)을 통해 API 키와 암호화 키를 안전하게 보호하여 클라우드 환경 전반에 걸친 보안 거버넌스를 강화한다.

결과적으로 제안된 작업흐름은 초기화, 계획, 정책 검증, 적용 및 감찰의 단계로 구성된다. 이를 CI/CD 파이프라인과 통합함으로써 인프라의 전 수명주기에 걸쳐 지능형 자율 관리 체계를 구현한다.

본 논문에서는 클라우드 환경에서 빈번하게 발생하는 설정 오류(Misconfiguration)를 주요 보안 위협으로 정의하였다. 주요 점검 대상은 공개 접근(Public Access), 과도한 권한(Over-Privileged IAM), 저장소 암호화 미적용(Unencrypted Storage), 네트워크 접근제어(Security Group), 로그 설정(Log Configuration) 등이며, 이러한 항목을 기반으로 보안 정책 검증을 수행하였다.

### 3.3 Security Verification Framework and Methodology

제안한 아키텍처는 설계, 배포, 운영의 전 수명주기(Lifecycle)에 걸쳐 보안 검증을 지속적으로 수행하도록 설계하였다[5]. 보안 강화 메타클라우드 아키텍처의 핵심은 인프라 전 수명주기 전반에 걸쳐 설정 오류로 인한 취약점을 사전에 차단하고 지속적인 거버넌스를 유지하는 것이다. 본 논문에서 제안하는 보안 검증 체계는 설정 오류를 조기에 탐지하고 보안 위험을 감소시키기 위한 자동화 지원 기술이며, 실제 보안 사고의 발생률 자체를 직접 측정하거나 예방 효과를 정량적으로 검증하는 것을 연구 범위로 하지는 않는다. 시스템은 Terraform 코드 생성 이후 자동 보안 검증 절차를 수행한다. 정교한 위협 모델을 기반으로 사전 점검과 사후 점검을 유기적으로 연계한 이원화된 보안 검증 방법론을 채택한다.

본 논문의 보안 검증 절차는 동일한 Terraform 코드, 동일한 tfsec(v1.28) Rule Set 및 동일한 보안 정책을 적용하여 수행되었다. 또한 모든 실험은 AWS, Microsoft Azure 및 Naver Cloud Platform 환경에서 동일한 절차와 평가 기준을 적용하여 수행하였으며, 동일한 입력 조건에서 동일한 보안 검증 결과를 재현할 수 있도록 실험 프로토콜을 구성하였다.

#### 3.3.1 Threat Model and Vulnerability Definition

가트너(Gartner)의 분석에 따르면 클라우드 보안 사고의 대부분이 공급자가 아닌 사용자의 설정 오류(Misconfiguration) 및 인적 오류(Human Error)에서 기인한다. 본 아키텍처에서는 이를 방지하기 위해 다중 클라우드 환경의 고유한 위협 요소를 도출하고 시스템적 제어 메커니즘을 설계하였다. 주요 위협 모델은 다음과 같이 세 가지 범주로 정의된다.

본 논문에서는 STRIDE 기반 분석과 CSP 보안 가이드라인을 참고하여 위협 모델을 정의하였다. 특히 클라우드 환경에서 실제 발생 빈도가 높은 설정 오류(Misconfiguration)를 중심으로 위협을 분류하였으며, 이를 다음의 세 가지 범주로 구분하였다.

첫째, 인터넷 노출(Exposure)은 Security Group, Network ACL, Load Balancer 등의 잘못된 설정으로 인해 관리 포트(SSH, RDP) 또는 데이터베이스 서비스가 외부에 노출되는 위협이다.

둘째, 권한 오남용(Privilege Misuse)은 IAM 정책의 최소 권한 원칙 미준수, 과도한 관리자 권한 부여, 공개 스토리지(Bucket) 설정 등으로 발생하는 위협이다.

셋째, 형상 불일치(Configuration Drift)는 IaC 코드와 실제 운영 환경이 서로 달라짐으로써 보안 정책이 지속적으로 유지되지 못하는 상황을 의미한다.

본 논문에서는 이 세 가지 위협을 핵심 통제 대상으로 정의하고 모든 보안 점검 규칙을 이에 대응하도록 설계하였다[18].

각 위협은 tfsec에서 제공하는 Terraform 보안 Rule과 CSP 보안 가이드라인을 기준으로 식별하였으며, 동일한 위협 모델을 AWS, Azure 및 Naver Cloud Platform 환경에 일관되게 적용하였다.

#### 3.3.2 Dual-Layer Security Assessment Framework

본 시스템은 인프라의 전 수명주기에 걸쳐 보안 검증이 이루어질 수 있도록 배포 전·후 2단계 보안 통제 체계를 적용하였다. 사전 점검 (Shift-Left Security): 사용자가 시각적 캔버스를 통해 설계를 완료하면, 플랫폼은 이를 테라폼(Terraform) 코드로 자동 변환한 후 배포 전 단계에서 정적 분석 스캔을 수행한다.

Table 4. Comparison of Pre-deployment and Post-deployment Security Validation

| Category | Pre-deployment | Post-deployment |
|---|---|---|
| Inspection Target | Terraform IaC Code | Running Infrastructure |
| Inspection Stage | Before Deployment | After Deployment |
| Inspection Method | tfsec Static Analysis | CSP APIs and State File Analysis |
| Primary Objective | Early Detection and Elimination of Security Vulnerabilities | Drift Detection and Continuous Security Monitoring |
| Result | Deployment Blocked upon Security Policy Violations | Alerts and Continuous Monitoring |

오픈소스 보안 스캔 도구인 tfsec (v1.28) 엔진을 백엔드에 내재화하여, 실제 자원이 CSP에 프로비저닝되기 전에 코드 레벨에서 보안 베스트 프랙티스 위반 여부를 검증하고 취약점이 발견될 경우 배포 프로세스를 원천 차단한다[12].

사후 점검: 인프라가 배포된 이후에는 주기적으로 각 CSP의 API 및 상태 파일(State File) 원격 저장 및 잠금 메커니즘을 활용하여 실제 구동 중인 인프라의 런타임 보안 상태를 점검한다. 수동 콘솔 조작으로 인해 발생하는 형상 불일치(Drift)와 새로운 실시간 위협을 상시 감찰함으로써 배포 전후를 아우르는 자동화된 거버넌스(CSPM)를 완성한다.

본 논문에서는 tfsec(v1.28)의 기본 Rule Set을 적용하였으며, 공개 접근(Public Access), 최소 권한(Least Privilege), 저장소 암호화(Storage Encryption), 감사 로깅(Audit Logging) 등 총 42개의 보안 정책 규칙을 자동으로 검증하였다[23].

#### 3.3.3 Security Domains and Assessment Items

보안 점검의 체계성을 확보하기 위해 국내외 표준 규제인 ISMS-P, ISO/IEC 27017, CSA CCM을 참조하여 플랫폼 내 보안 점검 표준을 6대 영역으로 분류하고 점검 항목을 도출하였다.

- 망 보안 (Network Security): 인터넷 개방형 인바운드 보안 그룹 규칙 존재 여부(예: Port 22, 3389의 외부 전면 개방 제한), 로드밸런서(ALB)의 안전하지 않은 SSL/TLS 버전 사용 여부 등을 점검한다.
  - SSH(22), RDP(3389)의 0.0.0.0/0 허용 여부
  - Public Subnet 내 Database 배치 여부
  - Internet Gateway 직접 연결 여부
- 접근 제어 및 신원 관리 (IAM): 자원 관리용 IAM 정책 내 과도한 권한 할당 여부, 자원별 접근 제어 정책(Bucket Policy 등)의 퍼블릭 허용 여부를 검증한다.
  - AdministratorAccess 사용 여부
  - Wildcard(*) 권한 포함 여부
  - MFA 적용 여부
- 데이터 암호화 및 보호 (Data Protection): S3 버킷, EBS 볼륨, RDS 인스턴스 등의 서버 측 암호화(SSE) 활성화여부 및 기본 암호화키(KMS) 설정 적정성을 점검한다.
  - EBS Volume 암호화 여부
- 로깅 및 모니터링 (Logging & Audit): AWS CloudTrail, Azure Activity Log 등 감사 추적 기능의 상시 활성화 여부, 로그 저장소의 무결성 보호 설정을 검증한다[3].
  - AWS CloudTrail 활성화 여부
  - Azure Activity Log 활성화 여부
  - Log 무결성 보호 여부
- 취약점 및 설정 관리 (Configuration): 각 클라우드 서비스별 기본 설정 유지 여부 및 퍼블릭 액세스 차단(Public Access Block) 설정 누락 여부를 확인한다.
  - S3 Bucket Public Access Block 활성화 여부
  - Object Storage Versioning 활성화 여부
- 백업 및 재해 복구 (Resilience): 핵심 데이터 자산에 대한 자동 백업 정책 수립 여부 및 백업본의 다중 지역(Multi-Region) 복제 설정을 점검한다.
  - Multi-AZ 구성 여부
  - 자동 백업 정책 여부
  - Cross-Region 복제 여부

본 논문에서는 총 42개의 보안 정책 규칙을 정의하여 모든 클라우드 환경에서 동일한 통제 목표를 적용하였다.

예를 들어 네트워크 보안에서는 SSH(22) 및 RDP(3389)의 외부 공개 여부를 점검하였으며, 접근제어 영역에서는 IAM 최소 권한 원칙 준수 여부를 확인하였다. 또한 저장소 영역에서는 암호화 적용 여부를 동일한 기준으로 검증하였다.

#### 3.3.4 Regulatory Compliance Scoring Method

취약점 정보를 정량적으로 평가하기 위해 위험도 기반 가중치(Severity Weight)를 적용한 Security Compliance Score를 산출하여 클라우드 보안 수준을 정량적으로 평가하였다. 조직의 거버넌스 지표로 활용할 수 있도록 위험도 가중치 기반의 정량적 '보안 컴플라이언스 점수(Compliance Score)' 산출 방식 식(1)을 도입하였다[25]. 최종 보안 점수는 100점 만점을 기준으로 진단된 취약점의 위험 등급별 개수와 가중치를 차감하여 계산된다.

$$Security\ Score = 100 - \min\left(100, \sum_{i \in \{Critical,High,Medium,Low\}} (W_i \times N_i)\right) \quad (1)$$

본 논문의 실험은 "① 시각적 인프라 설계 → ② Terraform 코드 생성 → ③ tfsec(v1.28) 기반 사전 보안 점검 → ④ 정책 위반 수정 → ⑤ Terraform 배포 → ⑥ 배포 후 형상 및 보안 상태 점검 → ⑦ Security Compliance Score 산출"의 동일한 절차에 따라 수행하였다. 모든 CSP 환경에 동일한 실험 시나리오와 평가 기준을 적용하여 실험의 재현성과 객관성을 확보하였다.

위험 등급별 가중치(Wi) 설정 기준: Critical / High (상): 시스템 침해로 직결되는 치명적 오설정 (가중치 W= 5.0), Medium (중): 간접적인 취약점을 유발하거나 보안 모범사례 미준수 (가중치 W = 3.0), Low (하): 경미한 설정 누락 또는 권장 사항 미이행 (가중치 W = 1.0) 본 아키텍처 기반의 'CloudS'플랫폼은 이 점수화 알고리즘을 실시간으로 연산하여 대시보드에 종합 점수 및 영역별 달성률을 시각화함으로써, 보안 관리자가 인프라의 전반적인 거버넌스 상태를 직관적으로 식별할 수 있도록 지원한다.

각 점검 항목은 ISO/IEC 27017, CSA CCM 및 ISMS-P의 관련 통제 항목과 매핑하여 구성하였으며, 동일한 규제 준수 기준을 모든 CSP 환경에 적용하였다.

## IV. Implementation and Functional Verification

본 논문에서 제안한 아키텍처의 실효성을 입증하기 위해 주요 클라우드 서비스 제공업체(AWS, Azure, NCP 등)를 통합 관리할 수 있는 플랫폼인 'CloudS'를 구현하였다. CloudS는 중소 규모 조직과 공공기관에서도 GUI 기반의 시각적 인터페이스를 이용하여 클라우드 인프라를 설계하고, 이를 IaC 코드로 변환하여 배포할 수 있도록 구현하였다. 시스템은 React 기반의 프론트엔드와 Python FastAPI 기반의 백엔드로 구성하였으며, Terraform을 IaC 엔진으로 적용하여 이기종 클라우드 환경에서도 일관된 인프라 관리가 가능하도록 설계하였다.

CloudS는 시각적 설계, IaC 생성, 보안 검증 및 운영 관리 기능으로 구성된다. 첫째, 시각적 인프라 설계 기능은 사용자가 캔버스 상에서 자원을 Drag & Drop 방식으

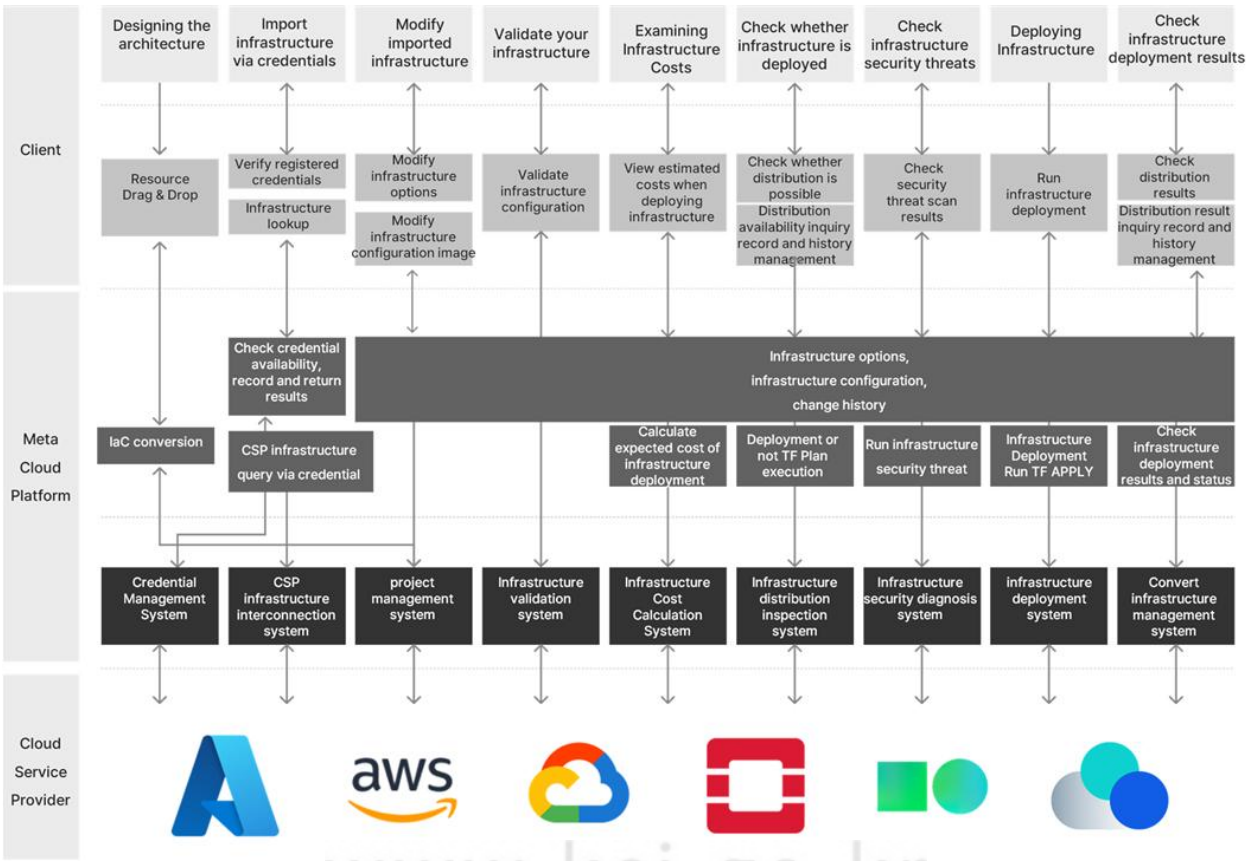


Fig. 3. Meta-Cloud CloudS Architecture and Detailed Functional Components

로 배치하고 연결 관계를 정의하면 이를 실시간으로 분석하여 자원 간의 의존성을 파악한다. 둘째, IaC 코드 자동 변환 기능을 통해 시각화된 설계 결과물은 업계 표준인 테라폼 코드로 즉시 전환되며, 이를 통해 수동 코딩 과정에서 발생 가능한 오류를 줄일 수 있다[27].

셋째, 클라우드 인프라의 설정 오류(Misconfiguration)로 인한 보안 위협을 사전에 차단하기 위해, IaC 보안 스캔 도구인 Tfsec을 활용한 정적 분석 기반의 보안 검증 방법론을 기반으로 tfsec은 Terraform 코드를 분석하여 퍼블릭 클라우드(AWS, Azure, GCP 등)의 보안 베스트 프랙티스 준수 여부를 검증하며, 배포 전 단계(Pre-deployment)에서 취약점을 식별하는 사전 보안 검증 단계에서 망 보안, 접근 제어, 암호화 등 6대 보안 영역을 중심으로 취약점을 자동 분석하고 위험 수준을 가시화한다[20]. 동시에 각 CSP의 비용 API를 활용하여 월간 및 연간 예상 비용을 다국적 통화로 산출함으로써 재무적 가시성을 제공한다[26].

넷째, 지속적 감찰 및 인프라 Import 기능은 운영 중인 환경의 설정 변경을 실시간으로 감지할 뿐만 아니라, 기존에 수동으로 구축된 인프라를 캔버스로 불러와 코드 기반의 관리 체계로 전환할 수 있는 연계 기능을 수행한다.

다음 장에서는 전문가 심층 면담을 통한 질적 검증 결과와 CloudS 플랫폼 기반의 정량적 실증 분석 결과를 제시한다.

## V. Analysis

본 장에서는 제안한 메타클라우드 아키텍처와 'CloudS' 플랫폼의 실무 적용 가능성 및 효과성을 검증하기 위해 심층 분석과 실증 분석을 병행하여 수행하였다.

### 5.1 Qualitative Validation through Expert Interviews

#### 5.1.1 Interview Participant Selection Strategy

클라우드 산업 생태계 내 다양한 이해관계자의 관점을 고르게 수렴하고 다양한 이해관계자의 의견을 수집하기 위해 목적표집 기반 인터뷰를 수행하였다. 인터뷰 대상자는 클라우드 공급기업, 자원 구축 실무자, 보안 및 컴플라이언스 전문가, 실제 클라우드를 도입하는 수요기업의 최고책임자 등 각 영역을 대표할 수 있는 20년 이상의 경력을 보유한 전문가 4인으로 구성하였다. 전문가 그룹의 구체적인 명세는 다음 Table 5과 같다.

본 논문에서는 질적 연구의 대표적인 목적표집(Purposive Sampling) 방법을 적용하여, 각 이해관계자 집단을 대표할 수 있는 전문가를 선정하였다. 선정 기준은 (1) 클라우드 관련 실무경력 20년 이상, (2) 실제 구축 또는 운영 프로젝트 수행 경험, (3) 클라우드 보안 또는 거버넌스 분야의 전문성, (4) 연구 주제와 직접적인 관련성을 모두 충족하는 경우로 제한하였다.

Table 5. Interview Subjects and Sampling Strategy

| Participant | Expert Group | Expert Category | Detailed Role | Experience |
|---|---|---|---|---|
| Expert 1 | Cloud Service Provider Expert | CSP Company | AWS Certified Solutions Architect | 20+ years |
| Expert 2 | | MSP[3] Company | Cloud Implementation Project Manager (PM-level) | 20+ years |
| Expert 3 | Cloud Security Expert | Security Company | CISO[4] level | 20+ years |
| Expert 4 | Cloud Service Consumer Expert | Insurance Company | Chief Information Officer (CIO-level) | 20+ years |

#### 5.1.2 Question Design and Briefing

질적 인터뷰의 신뢰도를 높이고 플랫폼에 대한 인터뷰 결과는 동일한 분석 기준에 따라 분류하고 연구자 간 교차 검토를 수행하였다. 사전 학습 단계: 인터뷰 대상자들에게 객관적 지표인 '2024년 클라우드 산업 실태조사 현황'을 먼저 제공하여 거시적 시장 흐름을 인지하게 한 후, 제안 시스템인 '메타클라우드 클라우드 스튜디오'의 핵심 기능 설명과 실제 사용 시연을 선행하여 플랫폼에 대한 충분한 이해를 도모하였다. 본 질의 단계: 기업이 다중 클라우드를 도입하고 운영할 때 직면하는 실무적 애로사항을 1) 비용, 2) 전문인력, 3) 보안기술, 4) 사업 수행의 4가지 차원으로 분류하여 구조화된 질의를 진행하였다.

Table 6. Structure of Expert Interview Questions

| Category | Description |
|---|---|
| Pre-interview Preparation | Platform Demonstration |
| Number of Questions | 12 |
| Interview Duration | Approximately 60-90 minutes |
| Interview Method | Semi-structured Interview |
| Data Collection | Audio Recording and Field Notes |
| Analysis Method | Thematic Analysis |

3) MSP: Managed Service Provider

4) CISO: Chief Information Security Officer

#### 5.1.3 Analysis and Stakeholder Bias Control

심층 면담을 통해 수집된 전문가들의 진술 데이터는 사전에 정의된 4대 핵심 주제(비용, 인력, 보안, 사업 수행) 별로 분류하여 분석 구조화를 거쳤다. 질적 연구에서 발생할 수 있는 주관적 해석오류 및 편향(Bias)을 통제하기 위해, 연구 설계 단계에서부터 이해관계가 상충할 수 있는 공급자(CSP/MSP), 수요자(CIO), 독자적 감사인(CISO)을 균형 있게 배치하였다. 인터뷰 내용은 연구자가 사전에 정의한 분석 프레임워크(비용, 인력, 보안, 사업 수행)에 따라 개방 코딩(Open Coding)을 수행한 후, 동일 주제별로 범주화하였다. 이후 연구자 간 교차 검토를 통해 해석의 일관성을 확인하였으며, 이해관계가 상충되는 전문가 집단의 의견을 비교하여 특정 집단의 편향이 연구 결과에 반영되지 않도록 다양한 이해관계자를 포함하여 전문가를 선정하였으며, 동일한 질문지를 모든 전문가에게 적용하여 인터뷰를 수행하였다. 또한 인터뷰 결과는 연구자 간 상호 검토를 거쳐 최종적으로 분석하여 연구자의 주관적 해석에 따른 편향을 최소화하였다. 모든 인터뷰는 반구조화(Semi-structured Interview) 방식으로 진행하였다. 이를 통해 특정 집단의 이해관계나 입장에 치우치지 않는 객관적인 분석 결과를 도출할 수 있는 구조적 통제 메커니즘을 확립하였다

#### 5.1.4 Interview Results and Implications

전문가 인터뷰 결과를 비용, 보안, 인력 및 사업 수행 측면으로 구분하여 분석하였다. CloudS가 기존 도구와 비교하여 설계 단계부터 보안 검증을 통합한다는 점에서 실무 적용성과 운영 효율성이 높다고 평가하였다.

- 비용 차원: 전문가들은 클라우드 비용 문제를 단순한 요금제의 고저가 아닌 가시성(Visibility)과 거버넌스(Governance)의 부재에서 비롯되는 구조적 문제로 진단하였다.

- 보안 차원: 기술 자체의 결함보다는 시스템 배포 단계에서의 내부 설정 실수(Misconfiguration)가 보안 사고의 핵심 원인임을 재확인하였다[6]. 이를 방지하기 위해 배포 전후 단계를 유기적으로 잇는 연속적인 통제 체계와 자동화된 클라우드 보안 태세 관리(CSPM)의 도입이 시급하다고 강조하며 본 논문의 아키텍처적 지향점에 높은 타당성을 부여하였다.

- 인력 및 사업 수행 차원: 클라우드 도입 가속화의 실질적 병목 현상은 기술적 요인보다 기업 내부의 사회적·정치적 프로세스에서 기인하므로, 최고 경영진의 강력한 지원과 함께 업무 프로세스 재설계(BPR)가 병행되어야만 실질적인 전환 효율성을 달성할 수 있음을 확인하였다[14].

Table 7. Expert Interview Protocol

| Item | Description |
|---|---|
| Number of Experts | 4 |
| Sampling Method | Purposive Sampling |
| Average Experience | More than 20 Years |
| Interview Method | Semi-structured Interview |
| Interview Topics | Cost, Human Resources, Security, and Project Execution |
| Interview Duration | 60-90 Minutes |
| Analysis Method | Thematic Analysis |
| Bias Control | Balanced Representation of Cloud Providers, Cloud Consumers, and Auditors |

종합적으로 전문가들은 제안한 메타클라우드 아키텍처가 기존의 단순 자원 관리 중심 접근 방식과 달리 설계 단계부터 보안을 내재화하고 운영 자동화를 통합적으로 지원한다는 점에서 높은 실무 적용 가능성을 갖는다고 평가하였다.

### 5.2 Quantitative Evaluation with CloudS

본 논문에서는 제안한 CloudS 플랫폼의 성능을 평가하기 위하여 기존의 대표적인 접근 방식인 수동 점검(Manual Inspection), Terraform 기반 IaC 구축 방식 및 기존 CSPM 기반 운영 관리 방식과 비교하였다. 비교 기준은 시각적 설계 지원 여부, 보안 검증 기능, 운영 자동화 수준 및 멀티클라우드 지원 범위로 설정하였다. 제안한 플랫폼의 성능을 평가하기 위해 정량 실험을 수행하였다. 구현된 'CloudS'플랫폼을 기반으로 시간 효율성, 보안 내재화 효과, 비전문가의 아키텍처 재현성 관점에서 정량적 실증 분석을 수행하였다.

본 논문에서는 제안한 CloudS의 성능을 객관적으로 평가하기 위하여 기존 방식인 Manual Inspection뿐만 아니라 Terraform 기반 IaC 배포, tfsec 기반 정적 보안 점검, CSPM(Cloud Security Posture Management), CMP(Cloud Management Platform)를 비교 기준(Baseline)으로 선정하였다. 각 비교 대상은 대표 기능을 기준으로 동일한 평가 항목에서 비교하였다.

#### 5.2.1 Evaluation Metrics and Scoring

본 평가에서는 정량적 측정 평가의 일관성을 확보하기 위해 동일한 기준을 적용하였다. 아키텍처 완성도 점수는 전체 인프라 구성을 3대 대분류 영역으로 나누고, 총 10개의 세부 항목에 대해 가중치를 차등 부여하여 100점 만점으로 환산 산출한다. 개별 피실험자 j의 최종 완성도 점수

Sj를 구하는 공식은 (2)와 같다.

$$S_j = \sum_{i=1}^{n} (W_i \times C_i) \times 100 \quad (2)$$

(단, n은 세부평가항목의 총개수(10개)이며, Wi는 항목별 가중치로 Wi = 1.0을 만족한다, Ci는 자원검증 스크립트를 통한 채점 결과값이다)

최종 AIC 점수는 10개 세부 평가항목에 대한 가중합으로 계산하였다. 각 항목은 완전 구현(1.0), 부분 구현(0.5), 미구현(0.0)의 기준으로 자동 채점되며, 항목별 가중치를 적용한 후 100점 만점으로 환산하였다. 본 논문에서 제시한 97.3점 및 89.3점은 동일한 평가항목과 가중치를 적용하여 산출된 평균 점수이다.

여기서 $W_i$는 각 평가 항목의 중요도를 반영한 가중치이며, 모든 가중치의 합은 1.0이 되도록 설정하였다.

$C_i$는 자동 채점 결과를 의미하며, 각 평가 항목별 구현 수준에 따라 1.0(완전 구현), 0.5(부분 구현), 0.0(미구현)의 값을 부여하였다. 최종 AIC 점수는 각 평가 항목의 가중치와 구현 점수를 곱한 후 합산하여 100점 만점으로 환산하였다.

예를 들어 네트워크 계층(30점), 컴퓨팅 계층(30점), 보안 계층(40점)에서 각각 27점, 26점, 36점을 획득한 경우 최종 AIC 점수는 89점으로 계산된다.

#### 5.2.2 Weights and Reference Architecture

평가의 정답 기준이 되는 준거 모델은 평균 경력 15년 이상의 전문 클라우드 아키텍트 4인이 약 3개월간의 협업을 통해 설계한 상용 수준의 '고가용성 표준 아키텍처'로 정의하였다. 본 모델은 단순 리소스 배치를 넘어 격리된 VPC 환경 및 Multi-AZ 기반 서브넷(네트워크 계층), NAT Gateway 및 ALB를 통한 트래픽 분산(통신 계층), ASG를 통한 탄력적 자원 운영(컴퓨팅 계층) 등 복잡한 상용 환경 구성을 모두 포함하며, 이 상태를 완성도 100점의 정답 기준으로 설정하였다. 중요도에 따른 세부 평가 항목 및 가중치(Wi) 배치 기준은 다음과 같다.

- 네트워크 계층 (가중치 0.30): VPC 격리성, Multi-AZ 서브넷 설계의 적정성
- 컴퓨팅 및 통신 계층 (가중치 0.30): ASG 및 로드밸런서(ALB)의 유기적 연동 및 이중화 구성.
- 보안 및 접근 제어 계층 (가중치 0.40): 보안 그룹 최소 권한 적용, 스토리지 암호화 및 IAM 역할 매핑의 적정성.

채점 기준(Ci)은 피실험자가 생성한 코드 내 자원의 존재 여부, 논리적 연결성, 필수 속성값의 일치 여부를 자동화 스크립트로 대조하여 완전 일치는 1.0점, 세부 옵션 누락 시 0.5점, 불일치/미구현은 0.0점으로 처리하였다.

정량 평가는 동일한 평가 절차에 따라 수행하였다. 모든 참여자는 동일한 실험 시나리오와 평가 기준을 적용받았으며, 각 피실험자는 동일한 과제를 3회 반복 수행하였다. 평가 결과는 반복 측정값의 평균을 사용하였으며, 외부 전문가 3인이 결과의 적정성을 교차 검증하였다.

평가항목은 모든 집단(비전문가, 숙련자, 전문가)에 동일하게 적용하였으며, 평가 절차와 채점 기준 역시 동일하게 유지하였다.

Table 8. Architecture Implementation Completeness (AIC) Evaluation Items

| Evaluation Category | Evaluation Item | Weight |
|---|---|---|
| Network | VPC Configuration | 0.1 |
| Network | Subnet Configuration | 0.1 |
| Network | Route Table | 0.1 |
| Compute | EC2 Instance Deployment | 0.1 |
| Compute | Auto Scaling | 0.1 |
| Compute | Load Balancer | 0.1 |
| Security | Security Group | 0.1 |
| Security | IAM Role | 0.1 |
| Security | Storage Encryption | 0.1 |
| Security | Logging and Audit | 0.1 |
| Total | | 1 |

#### 5.2.3 Validation Protocol and Experimental Design

평가의 주관성을 배제하고 실험은 동일한 조건에서 반복 수행하여 결과의 신뢰성을 확인하였다.

- 실험군 구성 및 규모: 클라우드 관련 업무 경험이 없거나 입문 단계에 있는 비전문가 총 144명을 실험군으로 선정하였다.
- 반복 측정 및 편차 통제: 데이터의 신뢰성을 제고하기 위해 실험은 총 3차에 걸쳐 독립적으로 나누어 진행하였으며(1차 44명, 2차 48명, 3차 52명), 피실험자당 동일한 미션을 3회 반복 수행하도록 하여 요행 요소를 차단하였다.
- 교차 검증: 자동화 채점의 오탐을 통제하기 위해 독립적인 외부 클라우드 감리 전문가 3인이 크로스체크를 수행하였다. 집단별 최종 결과는 산술 평균과 집단 내 편차를 나타내는 표준편차를 함께 산출하여 데이터의 통계적 타당성을 검증하였다.

각 실험 결과는 자동 채점 시스템을 통해 1차 평가를 수행하였으며, 이후 클라우드 감리 전문가 3인이 독립적으로 결과를 검토하여 자동 채점 결과와 비교하였다. 평가 결과가 상이한 경우에는 합의를 통해 최종 점수를 확정하였다.

최종 실험 결과는 각 차수별 평균(Mean), 표준편차(Standard Deviation), 최솟값(Minimum), 최댓값

(Maximum)을 산출하여 집단 간 편차를 분석하였다. 또한 동일한 평가 기준을 적용하여 실험의 재현성과 객관성을 확보하였다.

집단 간 비교는 동일한 평가항목과 가중치를 적용하여 수행하였으며, 최종 점수는 각 집단의 평균(Mean)과 표준편차(Standard Deviation)를 기준으로 비교하였다.

#### 5.2.4 Heterogeneous Multi-Cloud (AWS, Azure, and Ncloud) Experimental Setup and Controls

메타클라우드 아키텍처의 핵심 기여인 '이기종 환경에 대한 일반성(Generality)'을 검증하기 위해, 국내외 대표 공급사인 AWS, Azure, Ncloud(NCP) 3대 공급사 환경을 대상으로 실험 환경을 완전히 동일하게 동기화하고 통제 조건을 수립하였다.

- 실험 환경의 동일성: 연산 및 네트워크 편향을 차단하기 위해, 플랫폼 엔진(CloudS)이 구동되는 백엔드 서버는 호스트 인프라(동일 CPU 사양, 메모리 배정, 아웃바운드 대역폭 보장) 환경 내에서 독립된 컨테이너로 격리되어 수행되었다. 인프라 배포 대상 리전은 물리적 거리에 따른 지연율 편차를 통제하기 위해 모두 국내 리전(AWS 서울 리전, Azure Korea Central, Ncloud 한국 리전)으로 통일하였다.

- 인프라 자원 및 토폴로지의 대등성: 준거 모델이 되는 고가용성 아키텍처의 구현을 위해 각 CSP별 API 컴포넌트를 일대일로 대등 매핑하였다. 실험에 투입된 총 검증 자원 수는 3개 CSP 모두 동일하게 24개의 핵심 컴포넌트(1개 인프라 경계, 4개 서브넷, 2개 라우팅 테이블, 2개 게이트웨이, 4개 보안 그룹, 4개 가상 머신, 2개 로드밸런서, 2개 자율 확장 그룹, 3개 객체 스토리지 버킷)로 설계되어 완전히 동일한 인프라 복잡도를 유지하였다.

- 보안 통제 룰셋의 등가성: 각 CSP 환경에서 작동하는 검증 규칙의 공평성을 확보하기 위해 'SK쉴더스 클라우드 보안 가이드라인' 기반의 핵심 취약점 점검 항목을 각 CSP용 tfsec 규칙으로 치환하되, 동일한 통제 목표(예: 인바운드 22/3389 전면 개방 차단, 스토리지 암호화 강제, 감사 로깅 상시 활성화 등)를 가진 총 42개의 정책 룰셋을 동일 개수로 적용하여 검증의 일관성을 확보하였다.

적용된 42개의 보안 정책 규칙은 tfsec 기본 Rule과 SK 쉴더스 클라우드 보안 가이드라인의 공통 통제 항목을 기반으로 구성하였으며, 모든 CSP에서 동일한 Rule Set을 적용하여 실험의 일관성과 재현성을 확보하였다.

Table 9. Experimental Configuration across Multi-cloud Service Providers

| Item | AWS | Azure | Ncloud |
|---|---|---|---|
| Region | Seoul | Korea Central | Korea Region |
| VPC/VNet | Same | Same | Same |
| Subnets | 4 | 4 | 4 |
| Virtual Machines | 4 | 4 | 4 |
| Load Balancer | 2 | 2 | 2 |
| Auto Scaling | 2 | 2 | 2 |
| Security Groups | 4 | 4 | 4 |
| Object Storage | 3 | 3 | 3 |
| Security Rules | 42 | 42 | 42 |
| Evaluation Criteria | Same | Same | Same |

#### 5.2.5 Multi-Cloud Time Efficiency Analysis

기존의 수동 점검 방식이 지닌 시공간적 한계를 정량적으로 실증하기 위해, 국내 표준인 'SK쉴더스 클라우드 보안 가이드라인'을 기준으로 이기종 CSP(AWS, Azure, Ncloud)환경에서의 취약점 점검 성능 실험을 수행하였다. 실험 결과, AWS, Azure 및 Ncloud 환경 모두에서 동일한 보안정책을 적용한 결과 유사한 수행 결과가 나타났다.

Table 10. Experimental Configuration across Multi-cloud Service Providers

| Baseline | Terraform | tfsec | CSPM | CMP | CloudS |
|---|---|---|---|---|---|
| Visual Modeling | ✗ | ✗ | ✗ | △ | ✓ |
| IaC Generation | ✓ | ✗ | ✗ | ✗ | ✓ |
| Static Security Validation | ✗ | ✓ | ✗ | ✗ | ✓ |
| Runtime Monitoring | ✗ | ✗ | ✓ | ✓ | ✓ |
| Multi-cloud Support | △ | ✗ | ✓ | ✓ | ✓ |
| Deployment Automation | ✓ | ✗ | ✗ | △ | ✓ |
| Integrated Workflow | ✗ | ✗ | ✗ | ✗ | ✓ |

비교 결과 Terraform은 IaC 생성에는 효과적이지만 보안 검증과 운영 관리 기능은 제공하지 않았다. tfsec은 정적 보안 분석에는 효과적이었으나 배포 및 운영 기능은 제공하지 않았다. CSPM과 CMP는 운영 단계의 관리 기능은 지원하지만 설계 단계의 Security by Design 기능은 제한적이었다[8]. 반면 CloudS는 설계, 보안 검증, 자동 배포 및 운영을 단일 플랫폼으로 통합하여 기존 접근 방식 대비 높은 통합성과 활용성을 제공하였다.

각 CSP는 내부 API 구조와 자원 명칭은 상이하지만, 메타클라우드 추상화 계층을 통해 동일한 보안 통제 정책과 Terraform 기반 IaC를 적용할 수 있었으며, 보안 점검 결과 또한 동일한 기준으로 산출되었다. 이는 제안한 플랫폼이 특정 CSP에 종속되지 않고 다양한 멀티클라우드 환경에서 일관된 보안 거버넌스를 제공할 수 있음을 보여준다. 숙련된

보안 컨설턴트가 자원 분석부터 보고서 작성까지 수행하던 기존 방식과의 비교 결과는 다음 Table 11과 같다.

Table 11. Comparison of Security Inspection Workflow Time

| Comparison Item | Manual Inspection[5)] | CloudS |
|---|---|---|
| Inspection Target Identification | Manual | Automated |
| Policy Validation | Manual | Automated |
| Vulnerability Analysis | Manual | Automated |
| Result Compilation | Manual | Automated |
| Report Generation | Manual | Automated |
| Total Execution Time | 16 hours | ≤ 0.38 s |

본 논문에서 CloudS의 수행시간은 단순한 tfsec 스캔 시간만을 의미하지 않는다. Terraform 코드 분석, 보안 정책 검증, 취약점 식별, 결과 정리 및 보안 점검 리포트 자동 생성까지 포함한 전체 자동화 워크플로우의 수행시간을 측정하였다.

수동 방식의 수행시간은 숙련된 클라우드 보안 컨설턴트가 ① 점검 대상 식별, ② 취약점 분석, ③ 결과 정리, ④ 점검 보고서 작성까지 수행하는 전체 워크플로우에 소요된 시간을 의미한다.

반면 CloudS 플랫폼은 동일한 보안 정책을 기준으로 ① Terraform 코드 분석, ② 정책 규칙 검증, ③ 취약점 식별, ④ 결과 리포트 자동 생성까지 수행하는 전체 자동화 워크플로우의 수행시간을 측정하였다.

따라서 본 논문의 시간 비교는 수동 점검과 CloudS 모두 점검 대상 식별, 보안 정책 검증, 취약점 분석, 결과 정리 및 점검 보고서 생성까지 포함한 동일한 산출물 기준의 전체 워크플로우 수행시간을 비교한 결과이다.

실험 결과, 기존에 16시간이 소요되던 3대 공급사 계정의 취약점 진단 미션이 CloudS의 백엔드 자동화 엔진을 통해 각각 1초 미만(0.38초/0.21초/0.13초)에 완료되어 전체 보안 점검 워크플로우의 수행시간이 크게 감소하였다. 제안한 CloudS 플랫폼은 동일한 보안 점검 절차를 기준으로 수동 방식 대비 전체 보안 점검 워크플로우의 수행시간을 수동 점검 방식과 비교하여 전체 수행시간이 감소하였다. 특히 자동화된 정책 검증과 결과 리포트 생성 기능을 통해 반복적인 보안 점검 업무의 효율성을 크게 향상시킬 수 있음이 나타났다.

실험 결과 자동화 기반 플랫폼은 반복적인 보안 점검 업무의 수행시간을 크게 단축하였다. 특히 동일한 보안 정책을 기준으로 한 전체 점검 워크플로우에서 수동 방식 대비 수행시간이 크게 감소하였으며, 운영자의 반복 업무를 줄이고 지속적인 보안 거버넌스를 지원할 수 있음이 도출되었다. 또한 CSP별 배포 성공률, 자원 생성 결과 및 보안 정책 검증 결과에서도 모두 동일한 결과를 확인하였으며, 제안한 메타클라우드 아키텍처가 특정 CSP에 의존하지 않고 일관된 운영 특성을 제공함이 도출되었다. 실험 과정에서는 배포 실패 사례와 오탐 및 미탐 사례가 관찰되지 않아 동일한 실험 조건에서 안정적인 적용 가능성을 확인하였다.

Table 12. Comparison of Experimental Results across Multi-cloud Platforms

| Evaluation Item | AWS | Azure | Ncloud |
|---|---|---|---|
| Deployment Success | 100% | 100% | 100% |
| Resource Creation Success | 24/24 | 24/24 | 24/24 |
| Security Rules Applied | 42 | 42 | 42 |
| Security Policy Violations Detected | Same | Same | Same |
| False Positive | Not observed | Not observed | Not observed |
| False Negative | Not observed | Not observed | Not observed |

모든 CSP에서 동일한 Terraform 코드와 동일한 42개의 보안 정책을 적용한 결과, 배포 성공률은 모두 100%였으며 24개의 인프라 자원이 정상적으로 생성되었다. 또한 tfsec 기반 보안 검증 결과 모든 CSP에서 동일한 정책 위반 항목이 탐지되었고, 보안 정책 적용 결과 역시 일관되게 나타났다. 실험 과정에서는 오탐(False Positive) 및 미탐(False Negative)이 관찰되지 않았으며, 동일한 실험 조건에서 배포 실패 사례는 발생하지 않았다. 이러한 결과는 제안한 CloudS 플랫폼이 특정 CSP에 종속되지 않고 이기종 멀티클라우드 환경에서도 동일한 보안 검증 결과를 제공할 수 있음을 보여준다.

#### 5.2.6 CloudS-Based Architecture Reproducibility Evaluation

플랫폼이 사용자의 인프라 설계 능력을 보완하는지 검증하기 위해, 비전문가 집단(실험군)의 차수별 수행 결과와 클라우드 숙련자 집단(비교군, 경력 3~5년), 전문가 집단(대조군, 경력 15년 이상)의 AIC 점수를 비교 분석하였다. 구체적인 통계 데이터는 다음 <표 13>과 같다.

인터뷰 결과는 주제 분석(Theme Analysis) 방법을 적용하여 분석하였다. 인터뷰 응답은 연구 목적과 관련된 핵

5) 숙련된 보안 컨설턴트가 점검 수행 및 보고서 작성까지 소요한 총 (Man-hours)

심 내용을 중심으로 분류하였으며, 유사한 의견은 동일한 주제로 범주화하였다. 분석 과정에서는 연구자 2인이 독립적으로 응답 내용을 검토한 후 결과를 상호 비교하여 합의된 내용을 최종 분석 결과로 반영하였다.인터뷰 결과는 기능성, 보안성, 사용성 및 실무 적용성의 네 가지 범주로 정리하였으며, 반복적으로 제시된 의견을 중심으로 대표 의견을 도출하였다.

총 3차례의 독립 실험 결과, 클라우드 아키텍처 설계 경험이 전무한 비전문가 그룹임에도 불구하고 CloudS의 시각적 컴포넌트 제어 및 IaC 변환 인터페이스를 통해 종합 평균 66.80점(표준편차 8.40)의 아키텍처 완성도를 달성하는 데 성공하였다.

최종 평균 완성도 66.80점은 총 144명의 실험 참가자에 대한 AIC 점수를 평균한 결과이며, 집단 내 표준편차는 8.40으로 나타났다. 이는 대부분의 참가자가 평균값을 중심으로 유사한 수준의 결과를 보였음을 의미하며, 특정 참가자의 우수한 성과에 의해 평균이 왜곡되지 않았음을 보여준다. 또한 세 차례의 독립 실험에서 평균 점수의 변화 폭이 크지 않아 제안한 플랫폼의 재현성과 안정성을 확인할 수 있었다.

Table 13. Analysis of Architecture Implementation Completeness by Non-experts Using CloudS

| Experiment Round | Number of Participants | Expert Benchmark Score (Target) | Implementation Completeness Score (Result) | Achievement Rate |
|---|---|---|---|---|
| 1st | 44 | 100 points | 69.88 points | 69.90% |
| 2nd | 48 | 100 points | 61.52 points | 61.50% |
| 3rd | 52 | 100 points | 69.00 points | 69.00% |
| Averag | 48 | 100 points | 66.80 points | 66.80% |

동일한 평가 절차를 숙련자 집단과 전문가 집단에도 적용한 결과, 제안한 AIC 평가체계는 모든 집단에 동일하게 적용 가능한 평가 프로토콜임을 확인하였다. 또한 동일한 실험 시나리오를 AWS, Azure 및 Ncloud 환경에 각각 적용한 결과, 모든 CSP에서 동일한 평가 기준에 따라 AIC 점수를 산출할 수 있었으며, 플랫폼의 기능적 차이는 발견되지 않았다. 이는 제안한 메타클라우드 아키텍처가 특정 클라우드 서비스 제공자에 종속되지 않고 이기종 멀티클라우드 환경에서도 동일한 적용 가능성을 유지함을 의미한다.

Table 14. Evaluation Criteria for Architecture Implementation Completeness (AIC)

| Category | Description |
|---|---|
| Evaluators | Automated Evaluation + Three Domain Experts |
| Participants | 144 Non-expert Participants |
| Experimental Repetitions | 3 |
| Evaluation Items | 10 |
| Total Weight | 1.0 |
| Mean | 66.8 |
| Standard Deviation (SD) | 8.4 |

실험 결과는 사전 기술 지식이 없는 사용자도 CloudS 플랫폼을 활용하여 전문가가 설계한 아키텍처의 약 66.8% 수준을 안정적으로 구현할 수 있음을 보여준다. 결론적으로 본 실증 평가는 제안 시스템이 사용자의 숙련도에 따른 인프라 품질 격차와 휴먼 에러를 획기적으로 해소하며, 만성적인 클라우드 전문 인력 부족 문제를 보완할 수 있는 실무적인 표준화 도구로서 강력하게 기능함을 입증하는 실무 적용 가능성을 뒷받침하는 결과로 판단된다.

비교 결과, Terraform은 IaC 기반 자동 배포에는 우수한 성능을 보였으나 시각적 모델링과 통합 보안 검증 기능은 제공하지 않았다. tfsec은 정적 보안 분석에는 효과적이었으나 운영 환경의 지속적인 보안 관리 기능은 제공하지 못하였다. 또한 기존 CSPM 솔루션은 런타임 보안 모니터링에는 적합하지만 인프라 설계 단계의 자동화 기능은 제한적이었다. 반면 CloudS는 설계, 배포, 보안 검증 및 운영을 단일 플랫폼으로 통합함으로써 기존 접근 방식보다 높은 실무 적용성과 관리 효율성을 제공하였다.

## VI. Conclusion and Future Research Directions

본 논문은 다중 클라우드 환경의 확산으로 발생하는 관리 복잡성과 보안 취약점 문제를 해결하기 위해, 보안 기술이 강화된 메타클라우드 아키텍처를 제안하고 실증적으로 검증하였다. 또한 기존 Terraform, tfsec, CSPM 및 CMP 기반 접근 방식과의 비교를 통해 CloudS가 설계 단계부터 운영 단계까지 통합된 보안 자동화 프레임워크를 제공함을 확인하였다.

특히 AWS, Azure 및 Ncloud를 대상으로 동일한 실험 환경과 평가 기준을 적용한 결과, 제안한 메타클라우드 아키텍처는 특정 CSP에 의존하지 않고 다양한 이기종 클라우드 환경에서 일관된 보안 정책 적용과 인프라 자동화를

지원할 수 있음을 확인하였다. 특히 세 CSP 모두에서 동일한 배포 성공률과 자원 생성 결과, 보안 정책 검증 결과를 확인하였으며, 오탐 및 미탐 없이 일관된 보안 검증이 수행됨을 통해 제안한 메타클라우드 아키텍처의 일반성과 실무 적용 가능성을 확인하였다. 본 논문에서 도출된 주요 결과는 다음과 같다.

Table 15. Comparison of Experimental Results across CSPs

| Item | AWS | Azure | Ncloud |
|---|---|---|---|
| IaC Generation | Supported | Supported | Supported |
| Security Scanning | Supported | Supported | Supported |
| Drift Detection | Supported | Supported | Supported |
| Cost Analysis | Supported | Supported | Supported |
| AIC Evaluation | Supported | Supported | Supported |
| Compliance Assessment | Supported | Supported | Supported |
| Deployment Success | Success | Success | Success |
| Security Validation | Success | Success | Success |

첫째, 이기종 클라우드 자원을 추상화하고 통합 관리할 수 있는 메타클라우드 아키텍처는 특정 공급자에 대한 종속성을 완화하고 상호운용성을 확보하는데 멀티클라우드 환경에서 효과적으로 활용될 수 있음을 확인하였다. 둘째, 시각적 설계 도구와 코드형 인프라(IaC)를 결합한 자동화 기술은 인프라 구성의 일관성을 보장하며, 셋째, 실증 분석을 통해 제안 시스템이 운영 비용과 시간을 획기적으로 절감함은 물론, 비전문가도 전문가 수준의 인프라를 설계할 수 있도록 지원함으로써 클라우드 전문 인력 부족 문제를 해결할 수 있는 실무 적용 가능성을 확인하였다[28].

기존의 Terraform, tfsec 및 CSPM 기반 접근 방식과 비교한 결과, 제안한 CloudS 플랫폼은 시각적 설계, Infrastructure as Code 자동 생성, 배포 전·후 보안 검증 및 멀티클라우드 운영을 통합적으로 지원함으로써 기존 연구의 한계를 보완할 수 있음을 확인하였다.

향후 연구에서는 메타클라우드 환경에서 Cloud, Edge, IoT를 통합하는 메타 컴퓨팅 구조를 기반으로, 인공지능을 활용한 워크로드 배치 최적화와 보안 위협의 자율 대응 기능을 통합한 지능형 운영 모델로의 확장이 필요하다. 또한, 인프라 계층을 넘어 애플리케이션 및 데이터 계층까지 보안 자동화 범위를 확장하고, 금융·공공 등 산업별 특화된 규제 준수 정책 모델을 세분화하여 향후 연구에서는 AI 기반 자율 운영 기능과 산업별 규제 준수 모델을 추가적으로 검토할 필요가 있다. 또한 본 연구는 AWS, Azure 및 NCP를 대상으로 수행되었으므로, 향후 GCP, Oracle Cloud 등 다양한 CSP를 포함한 검증과 대규모 산업 현장에서의 장기 운영 평가가 필요하다.



## REFERENCES


[1] J. Alonso, L. Orue-Echevarria, V. Casola, A. I. Torre, M. Huarte, E. Osaba, and J. L. Lobo, “Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review,” Journal of Cloud Computing, vol. 12, art. no. 6, 2023, doi:10.1186/s13677-022-00367-6.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010, doi:10.1145/1721654.1721672.

[3] Amazon Web Services, AWS Cloud Adoption Framework: Security Perspective, AWS Whitepaper, Dec. 2023. [Online]. Available: https://docs.aws.amazon.com/whitepapers/latest/aws-caf-security-perspective/aws-caf-security-perspective.html. Accessed: Jul. 2026.

[4] M. Chiari, M. De Pascalis, and M. Pradella, “Static analysis of Infrastructure as Code: A survey,” in Proc. 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA, Mar. 2022, pp. 218–225, doi:10.1109/ICSA-C54293.2022.00049.

[5] R. Chandramouli, F. Kautz, and S. Torres-Arias, *Strategies for the Integration of Software Supply Chain Security in DevSecOps CI/CD Pipelines*, NIST Special Publication 800-204D, National Institute of Standards and Technology, Gaithersburg, MD, USA, Feb. 2024, doi:10.6028/NIST.SP.800-204D.

[6] Cloud Security Alliance, *Top Threats to Cloud Computing: Pandemic Eleven*, Cloud Security Alliance, Jun. 2022. [Online]. Available: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-pandemic-eleven. Accessed: Jul. 2026.

[7] Cloud Security Alliance, Cloud Security Alliance, Cloud Controls Matrix (CCM) Implementation Guidelines, Version 2.0, Jun. 2024.

[8] Cloud Security Alliance, *Cloud Controls Matrix (CCM) v4.1*, Cloud Security Alliance, Jan. 2025. [Online]. Available: https://cloudsecurityalliance.org/research/cloud-controls-matrix. Accessed: Jul. 2026.

[9] N. Grozev and R. Buyya, “Inter-Cloud architectures and application brokering: Taxonomy and survey,” Software: Practice and Experience, vol. 44, no. 3, pp. 369–390, Mar. 2014, doi:10.1002/spe.2168.

[10] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, “An analysis of security issues for cloud computing,” Journal of Internet Services and Applications, vol. 4, art. no. 5, 2013, doi:10.1186/1869-0238-4-5.

[11] T. Ijlal, “Securing and embracing Infrastructure as Code,” ISACA Journal, vol. 4, Jul. 2020. [Online]. Available: https://www.isaca.org/resources/isaca-journal/issues/2020/volume-4/securing-and-embracing-infrastructure-as-code. Accessed: Jul. 2026.

[12] ISO/IEC 27017:2015, *Information Technology—Security Techniques—Code of Practice for Information Security Controls Based on ISO/IEC 27002 for Cloud Services*, International Organization for Standardization, Geneva, Switzerland, 2015.

[13] ISO/IEC 27018:2019, *Information Technology—Security Techniques—Code of Practice for Protection of Personally Identifiable Information (PII) in Public Clouds Acting as PII Processors*, International Organization for Standardization, Geneva, Switzerland, 2019.

[14] W. Jansen and T. Grance, *Guidelines on Security and Privacy in Public Cloud Computing*, NIST Special Publication 800-144, National Institute of Standards and Technology, Gaithersburg, MD, USA, Dec. 2011, doi:10.6028/NIST.SP.800-144.

[15] K. Morris, *Infrastructure as Code: Managing Servers in the Cloud*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2016.

[16] H. Bettaher, H. Ghanmi, and H. Touati, "Hybrid architecture for secure data sharing in multi-clouds system," The Computer Journal, vol. 68, no. 1, pp. 58–73, Jan. 2025, doi:10.1093/comjnl/bxae092.

[17] A. War, S. L. B. Nikiema, J. Samhi, J. Klein, and T. F. Bissyandé, "Security smells in infrastructure as code: A taxonomy update beyond the seven sins," arXiv preprint arXiv:2509.18761, Sep. 2025.

[18] M. Reece, T. Lander Jr., S. Mittal, N. Rastogi, J. Dykstra, and A. Sampson, "Emergent (In)Security of Multi-Cloud Environments," arXiv preprint arXiv:2311.01247, Nov. 2023.

[19] P. K. Senyo, E. Addae, and R. Boateng, "Cloud computing research: A review of research themes, frameworks, methods and future research directions," International Journal of Information Management, vol. 38, no. 1, pp. 128–139, Feb. 2018, doi:10.1016/j.ijinfomgt.2017.07.007.

[20] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1–11, Jan. 2011, doi:10.1016/j.jnca.2010.07.006.

[21] P. Trakadas, X. Masip-Bruin, F. M. Facca, S. T. Spantideas, A. E. Giannopoulos, N. C. Kapsalis, R. Martins, E. Bosani, J. Ramon, R. G. Prats, G. Ntroulias, and D. V. Lyridis, "A reference architecture for cloud-edge meta-operating systems enabling cross-domain, data-intensive, ML-assisted applications: Architectural overview and key concepts," Sensors, vol. 22, no. 22, art. no. 9003, Nov. 2022, doi:10.3390/s22229003.

[22] V. Varadarajan, Y. Zhang, T. Ristenpart, and M. Swift, "A placement vulnerability study in multi-tenant public clouds," in Proc. 24th USENIX Security Symposium (USENIX Security 15), Washington, DC, USA, Aug. 2015, pp. 913–928.

[23] E. Özdoğan, O. Ceran, and M. T. Üstündağ, "Systematic analysis of Infrastructure as Code technologies," Gazi University Journal of Science Part A: Engineering and Innovation, vol. 10, no. 4, pp. 452–471, Dec. 2023, doi:10.54287/gujsa.1373305.

[24] N. E. Díaz Ferreyra et al., "On the Understandability of Design-Level Security Practices in Infrastructure-as-Code Scripts and Deployment Architectures," ACM Transactions on Software Engineering and Methodology, vol. 34, no. 1, Art. no. 6, pp. 1–37, Jan. 2025, doi:10.1145/3691630.

Related Patents (Formatted References)

[25] S. G. Park, "Device and Method for Providing Heterogeneous Multi-Cloud Infrastructure Migration and Redundancy Services Based on C2C Infrastructure Migration," Korean Patent Application No. 10-2024-0089770, Korean Intellectual Property Office (KIPO), 2024.

[26] S. G. Park, "Device and Method for Automating Compliance-Based Cloud Infrastructure and Console Security Inspection," Korean Patent Application No. 10-2024-0089773, Korean Intellectual Property Office (KIPO), 2024.

[27] S. G. Park, "System and Method for Managing Cloud Costs and Assets," Korean Patent Application No. 10-2024-0096291, Korean Intellectual Property Office (KIPO), 2024.

[28] S. G. Park, "Method for IaC-Based Cloud Infrastructure Visualization Transformation and System Supporting the Same," Korean Patent Application No. 10-2024-0096293, Korean Intellectual Property Office (KIPO),2024.

[29] S. G. Park, "Device and Method for Calculating Cloud Infrastructure Costs," Korean Patent Application No. 10-2025-027863, Korean Intellectual Property Office (KIPO), 2025.

## Authors

Sungkap Park received the Ph.D. degree in Digital Management from Korea University, 2026. He is currently the Director of the Korea Institute of Information Security Management and a professional committee member of the Korea Internet & Security Agency (KISA). He holds a master's degree in Computer and Information Science from Yonsei University and has professional experience at SK Infosec and the National Credit Union Federation of Korea. His research interests include information security, Cloud Computing, Blockchain, MyData, and artificial intelligence (AI).

Miyoung Bang received the Ph.D. degree in Digital Management from Korea University, 2026. She has 22 years of experience as a backend developer developing escrow and accounting systems. She holds a master's degree in Computer Engineering from Kangwon National University and has professional experience as a Development Project Manager(PM) at Auction, eBay Korea, and CareDoc. Her research interests include AI, cloud computing, and anomaly detection.

Jiyun Kim is currently a Ph.D. candidate in Digital Management at Korea University. He is currently working in the Information Security Team at Kakao Games.

Jiyun Kim is currently a Ph.D. candidate in Digital Management at Korea University. He is currently working in the Information Security Team at Kakao Games.

Changhyun Kim received the Ph.D. degree in Digital Management from Korea University, 2026. He currently serves as the Chief Information Security Officer (CISO) and Chief Privacy Officer (CPO) at Wadiz, a global crowdfunding platform. Previously, His primary research interests include participation intention in reward-based crowdfunding, AI applications, AI solutions for small and medium-sized enterprises (SMEs), the Technology Acceptance Model (TAM), and the Information Systems Success Model (ISSM).

Dongsu Lee received the Ph.D. degree in Digital Management from Korea University, 2025. He is currently a Professor and Director of the Network Center at the School of Medicine and Science, Gachon University. He has previously served as an executive at major IT companies, including LG CNS, TmaxSoft, and Daewoo Information and his research interests include manufacturing AI and home healthcare AI.