# 쿠버네티스 기반 클라우드 환경에서 CNF 오케스트레이션 보안구조 설계

이장원, 김권준, 김영한*
숭실대학교

{jangwon.lee, lotring}@dcn.ssu.ac.kr, *younghak@ssu.ac.kr

# A Design of Security Architecture for CNF Orchestration in Kubernetes-based Cloud

JangWon Lee, KwonJun Kim, YoungHan Kim*
Soongsil Univ.

## 요 약

쿠버네티스 기반의 CNF 배포 및 관리 시스템의 오픈소스 프로젝트로서 Nephio 가 시작되었다. Nephio 에서는 쿠버네티스 리소스 모델을 통해 CNF 를 정의하고 GitOps 방식으로 배포되도록 설계되었다. 그러나, 설계된 Nephio 환경에서는 인프라 및 리소스의 접근 제어, CNF 의 검증 및 암호화된 통신, Git 저장소 내 민감한 정보 암호화 등의 보안적 요소가 부족하다. 본 논문에서는 오픈소스를 활용하여 Nephio 내 통합된 보안 구조를 설계하였다.

## Ⅰ. 서 론

클라우드 기반의 이동통신 시스템에서 쿠버네티스 설계 개념을 반영한 CNF 오케스트레이션 시스템을 위해 LFN(Linux Foundation Networking) 산하에 KRM(Kubernetes Resource Model) 및 GitOps 방식의 CNF 배포, 관리 수행을 기반으로 하는 Nephio[1] 오픈 소스 프로젝트가 진행되고 있다.

최근에는 Nephio 2 차 소프트웨어 공개를 통해 클러스터 및 워크로드 배포의 주요 기능이 동작함을 보였으나 보안 요소 관점에서의 구조 및 요구사항은 아직 논의 단계이며, 유사한 목표를 위해 오랜 시간 진행되어왔던 ONAP 프로젝트와 기술 공유를 진행하고 있다[2].

본 논문에서는 Nephio 구조에서 인프라 및 CNF 간 통신, 접근제어, 그리고 CNF 내 민감한 정보 관리 등에 대한 보안 요소를 정의하고 오픈소스 프로젝트를 통해 제안 구조를 구성할 수 있음을 보인다. 이를 통해, 기존 CNF 배포 구조의 변경없이 보안 적용 구조로의 확장이 가능함을 보인다.

## Ⅱ. 본 론

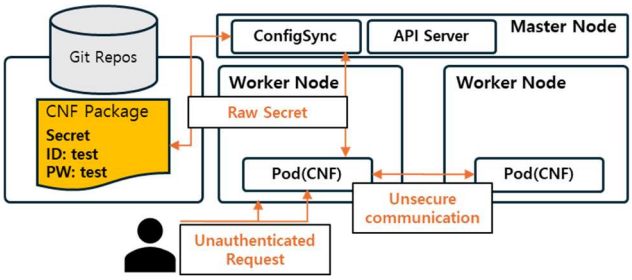


**그림 1. Nephio 기반 클러스터의 보안체계 요구사항**

그림 1 은 설계된 Nephio 환경에서 부족한 보안적 요소를 보인다. Nephio 기반의 클러스터는 내부의 ConfigSync 를 통해 GitOps 방식으로 CNF 를 배포 및 관리한다. Git 저장소에 존재하는 CNF 패키지는 KRM 형태의 리소스로 구성되어 있는 장점이 있지만, 아직 민감한 정보를 담는 시크릿 리소스에 대한 Nephio 의 대응 방법이 없다. 배포된 CNF 는 MSA(Micro Service Architecture) 기반의 여러 Pod 혹은 단일로 구성될 수 있으며, 임의의 대상과 신뢰성 있는 통신이 요구된다. 이를 위해, 일반적으로 사용되는 서비스 메시 기반의

mTLS 통신 구조와 인프라 및 CNF 접근 시 인증 방법 등의 추가적인 보안 강화가 Nephio 에 필요하다.

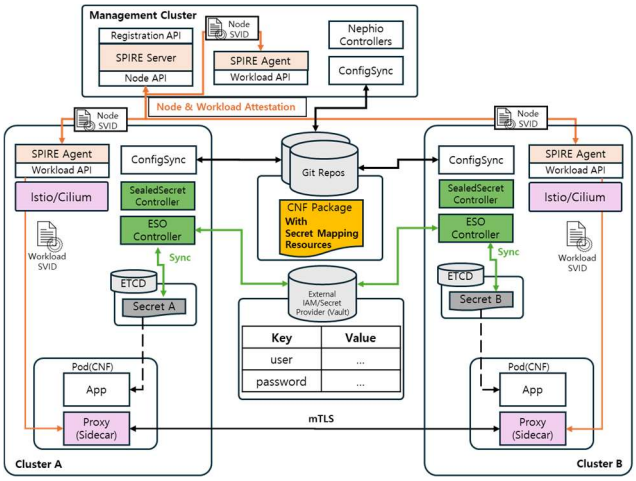


**그림 2. Nephio 기반체계의 보안 적용 구조도**

그림 2 는 본 논문에서 제안하는 보안 적용 구조도를 나타낸다. Git 저장소 내 시크릿 리소스 암호화를 위한 SealedSecret[3]과 Vault[4]를 통해 독립적으로 관리하는 민감한 정보를 쿠버네티스 인프라 내 리소스화 및 자동화하기 위해 ESO(External Secret Operator)를 연계 활용한다. 쿠버네티스 내 시크릿 리소스 관리를 위해 Vault 와 같은 대중적인 시크릿 관리 시스템을 사용할 수 있으나, KRM 방식을 수용하지 않는다. ESO 가 제공하는 주요 기능은 다양한 외부 시크릿 프로바이더와의 연결 및 쿠버네티스 시크릿 리소스와의 동기화이다. 이를 통해, CNF 패키지에 구성한 ESO 의 구성 리소스를 활용하여 쿠버네티스 내 동작하는 CNF 에 필요한 민감한 정보를 신뢰성 있게 제공할 수 있다. 다만, ESO 를 통한 외부 시크릿 프로바이더 연결 및 인증 정보는 기존 시크릿 리소스를 필요로 하므로, 그림 3 과 같이 SealedSecret 을 사용하여 Git 저장소 내 암호화하여 저장한다.

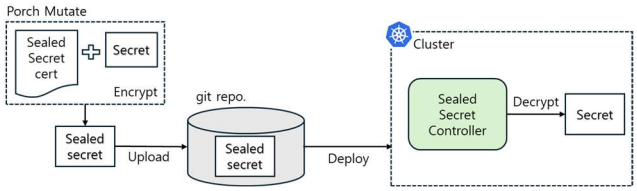


**그림 3. Git 저장소 내 Secret 암호화 절차도**

전체 인프라 및 CNF 의 인증 및 mTLS 통신 체계는 서비스 메시 프로젝트와 SPIRE 의 연계로 구성한다. 일부 서비스메시 프로젝트에서 mTLS 를 자체적으로 제공하나, 본 구조에서는 SPIRE 를 통해 전체 인프라 측면의 체계로 확장 구성한다. SPIRE 은 SPIFFE Runtime Environment 이며, 이기종 인프라에 걸쳐 통합된 Identity 컨트롤 플레인을 제공한다[5]. 그림 1 과 같이, 각 클러스터는 여러 개의 노드로 구성될 수 있고 CNF 를 배포하고 동작 시킨다. Spire 는 인프라를 구성한 노드 및 Pod 들의 고유 ID 를 부여하고 개체의 신뢰성을 검증하기 위한 SVID(SPIFFE Verifiable Identity Document)를 발급 및 관리하며, mTLS 통신을 위한 URI 체계를 제공한다. 따라서, 그림 2 의 Management Cluster 에 구성한 SPIRE Server 를 통해 발급한 SVID 를 갖는 개체들 간의 신원을 확인하고 신뢰성 있는 통신을 제공할 수 있다.

## Ⅲ. 결 론

본 논문에서는 쿠버네티스 기반의 CNF 배포 및 관리 시스템의 오픈소스 프로젝트인 Nephio 내 보안 요소 강화를 위한 구조를 제안하였다. 특히, 리소스의 접근 제어, CNF 의 검증 및 암호화된 통신, 그리고 Git 저장소 내 민감한 정보 암호화 등 부족한 보안 기능을 보완하기 위해 여러 오픈소스를 활용하였다. 본 제안을 통해 오케스트레이션 및 보안 구조를 통합함으로써, 신뢰성 있는 CNF 오케스트레이션 인프라를 손쉽게 제공할 수 있을 것이라 기대된다.


## ACKNOWLEDGMENT

이 논문은 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2024-00398379, (총괄 4-세부 2) 텔코용 고성능/고가용성 6G 크로스-클라우드 인프라 기술개발)


## 참 고 문 헌


[1] Kibeom Park. “Technology trends and challenges in SDN and service assurance for end-to-end network slicing”, Computer Networks, Vol. 234, Oct. 2023.

[2] ONAP, “ONAP: TSC Initiatives for ONAP Roadmaps and Strategies”, 2024, (https://wiki.lfnetworking.org/display/LN/2024-05+-+ONAP%3A+TSC+Initiatives+for+ONAP+Roadmaps+and+Strategies)

[3] Bitnami, “Sealed Secrets”, 2024, (https://sealed-secrets.netlify.app/)

[4] Alexander Krause. “Pushed by Accident: A Mixed-Methods Study on Strategies of Handling Secrets in Source Code Repositories”, 32nd USENIX Security Symposium, pp. 2527-2544, Aug. 2023.

[5] Eduardo Falcao. “Supporting Confidential Workloads in SPIRE”, 2022 IEEE CloudCom, pp.186-193, Dec. 2022.