

베어메탈 환경에 쿠버네티스 클러스터 구축을 위한 오퍼레이터 구현

정승준, 이장원, 김영한

숭실대학교

{jsj0612, jangwon.lee}@dcn.ssu.ac.kr, younghak@ssu.ac.kr

Implementation of an Operator for Deploying Kubernetes Clusters on Bare Metal Environments

Seung Jun Jeong, Jang Won Lee, Young Han Kim

Soongsil Univ.

요약

쿠버네티스 플랫폼이 클라우드 인프라의 사실상 표준이 되었고, 베어메탈 환경에서도 이를 활용한 인프라 구축 및 관리 기술이 요구되고 있다. 쿠버네티스 진영에서는 Cluster API(CAPI) 프로젝트가 활용되나, 최소한의 기능으로만 구성되어 관리자의 개입이 요구된다. 본 논문에서는 구축 시나리오 및 문제를 정의하고 해결한 Operator를 설계하여, 베어메탈 기반 쿠버네티스 클러스터 구축 시간 단축 및 관리 효율성 향상을 보인다.

I. 서론

쿠버네티스 플랫폼이 사실상 클라우드 인프라의 표준이 되며, 인프라의 기본 계층인 베어메탈 환경에서도 쿠버네티스 클러스터 구축 및 관리가 시도되고 있다[1-2]. 이는 기존 가상화 환경 대비 성능 이점과 관리 간소화 및 특수 하드웨어와의 가상화 기술 문제를 해소할 수 있다[3-4]. 하지만 베어메탈 환경에서의 구축은 물리적 자원의 접근이 요구되므로, 이를 위한 프로토콜과 복잡한 절차가 수반된다. 베어메탈은 실제 동작 여부와 관계없이, IPMI 등의 프로토콜을 통해 모니터링 및 구동이 조절될 수 있어야 한다. 또한, 베어메탈 내 구성 설치 절차는 시작 단계에 따라 달라져야 한다.

오픈소스 진영에서는 쿠버네티스 산하의 Cluster API(CAPI)[5]를 중심으로 다양한 인프라 프로바이더와의 연계를 통해 클러스터 구성을 자동화하는 기술을 제공하고 있다. 베어메탈 환경의 경우, 요구 기능들을 수행하는 베어메탈 관리 프로젝트와의 연계를 통해 쿠버네티스 클러스터가 구성될 수 있다. 그러나, 실제 운용에서 CAPI 및 관련 프로젝트를 통한 베어메탈 기반 인프라 구축은 관리자의 개입이 필수적이다. 첫 번째로, 베어메탈의 다양한 하드웨어 구성을 갖는 대규모 인프라에서 쿠버네티스 환경의 유연한 구성을 명세하기 어렵다. 두 번째로, 인프라 내 환경 구축을 위해 다양한 툴이 조합되어 사용될 수 있으나, 이에 따른 구축 명세서를 관리자가 직접 작성해야만 한다. 예를 들어, 베어메탈의 OS는 다양한 방법으로 사전 구성될 수 있고 파생된 접근 계정 정보는 구축 명세서에 포함되어야 한다.

이를 위해, 본 논문에서는 베어메탈 환경에서의 구축 시나리오를 설명하고 요구사항을 정의한다. 그리고, 요구사항을 만족하기 위한 CAPI를 기반한 베어메탈 관리 및 쿠버네티스 클러스터 구축 자동화 Operator 설계 및 제안 구조 기반 쿠버네티스 클러스터 구축 절차를 보인다. 성능 평가에서는 본 논문에서 정의한 시나리오를 통해 특정 베어메탈로 환경을 구성할 수 있음과 환경 구축시간이 기존 대비 단축될 수 있음을 그래프로 보이고 설명한다. 결론에서는 본 논문의 활용성 및 이후 연구 방향에 대해 설명하며, 본 논문을 마무리한다.

II. 본론

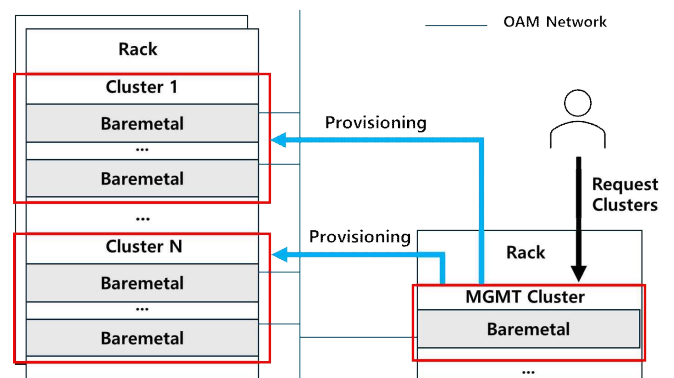


그림 1. 베어메탈 기반 쿠버네티스 클러스터 구축 시나리오

그림 1은 본 논문에서 요구하는 베어메탈 기반 쿠버네티스 클러스터 구성 시나리오를 보인다. 기존 베어메탈은 관리 시스템에 서버의 정보를 등록하여, IPMI 등의 프로토콜 방식으로 PXE 부팅 진행 후 OS를 설치 및 관리한다. 이처럼 관리되는 많은 랙들이 존재하는 환경에서, 원하는 서버만을 그룹화하여 쿠버네티스 클러스터를 구축하고자 한다. 일반적으로 베어메탈 관리를 위해 대중적으로 활용되는 Metal3와 IroniC가 CAPI와 연계하여 사용될 수 있다[6]. 하지만 크게 두 가지의 문제가 발생한다. 첫 번째로, CAPI 기반 클러스터의 노드에 대응되는 Metal3Machine 리소스에 필요한 베어메탈 정보를 관리자가 개입하여 기술해야 한다. 두 번째로, 실제 노드에 OS가 설치되어 있더라도, 클러스터 구축 과정의 일관성을 위해 OS 설치 단계부터 진행된다. OS 설치의 클러스터 구성 전, 3자에 의해 이뤄질 수 있고 사전 여유 시간에 진행하여 시간을 절약할 수 있다. 이 두 문제를 해결하기 위해, 기존 CAPI 기반 자동화를 확장한 Operator 기반 시스템을 제안한다.

그림 2는 제안한 Operator 기반 쿠버네티스 클러스터 구축 절차를 보인다. Operator는 BMHPool 컨트롤러와 BMCluster 컨트롤러로 구성된다.

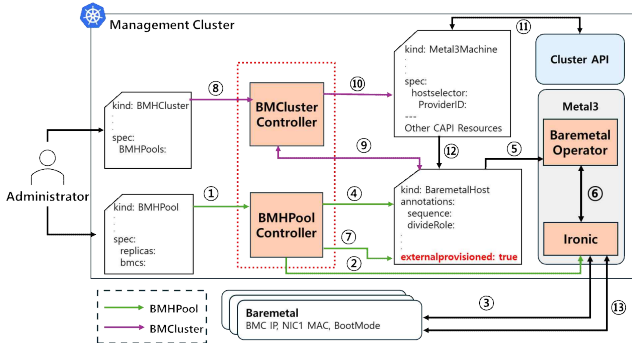


그림 2. 제안 Operator 기반 쿠버네티스 클러스터 구축 절차

BMHPool 컨트롤러는 베어메탈 리소스를 그룹으로 관리하는 목적을 갖는다. BMCluster 컨트롤러는 BMHPool 리소스에 기반하여 쿠버네티스 클러스터가 구축될 수 있게 CAPI 리소스를 생성한다. 관리자는 BMHPool 리소스를 사용하여 베어메탈 그룹 및 인증정보와 그룹 내 각 베어메탈의 순서와 역할을 정의한다. BMHPool 컨트롤러는 리소스를 토대로 세 가지의 단계를 수행한다. 첫 번째로, 리소스의 BMC IP 주소와 인증정보를 토대로 IroniC에 Inspection을 요청한다. 이는 베어메탈의 인터페이스 MAC 정보를 취합하여 Metal3의 BMH(BareMetalHost) 리소스를 1:N으로 등록하기 위함이다. 두 번째로, 각 베어메탈의 BMH 리소스를 생성한다. 이 과정에서 OS 정보 및 BMHPool에 의해 생성됨을 확인할 수 있는 Annotation을 추가한다. Metal3 및 IroniC은 등록된 BMH 리소스를 확인하여 OS를 설치하고 BMH 리소스에 Provisioned 상태로 업데이트한다. 마지막으로 BMHPool 컨트롤러는 Metal3가 업데이트한 정보를 확인하고 외부에 의해 OS가 설치됨을 표시한다. 이후 BMHPool 컨트롤러는 전력 소모를 줄이기 위해 베어메탈의 전원을 종료한다.

관리자는 특정 베어메탈 구성으로 쿠버네티스 클러스터를 구성하기 위해 BMHPool 리소스 및 클러스터 구성 정보(Master 및 Worker 수, CAPI 리소스의 옵션 등)를 포함한 BMCluster 리소스를 정의한다. BMCluster 컨트롤러는 선택된 BMHPool 리소스를 확인하여 각 BMH 리소스의 사용 순서와 역할을 조회하고 CAPI 리소스들을 생성한다. 이 과정에서 특정 베어메탈과의 매핑 정보 및 사전 구성된 OS 계층 정보가 관리자의 개입 없이 자동으로 포함될 수 있게 한다. 이후 과정은 기존 CAPI의 클러스터 생성 절차가 진행되어, 베어메탈 기반 쿠버네티스 환경이 구축된다.

III. 성능 평가

그림 1의 시나리오에 기반하여, 관리 인프라를 통한 기존 및 제안 구조에서의 쿠버네티스 클러스터 구축을 진행하였다. 클러스터는 한 개의 마스터 노드와 두 개의 워커 노드로 구성되며, 마스터와 워커 노드가 생성되어 서비스 되는 시간을 비교 분석한다. 그림 3과 같이 각 노드의 구축 과정은 총 세 단계로 구분된다. OS 설치 시간이 전체 구축 시간의 절반 이상을 차지함을 확인할 수 있으며, 재부팅과 쿠버네티스 서비스 구성 시간 순서대로 비중을 차지한다. 실제 쿠버네티스 클러스터 구축 핵심은 마스터 노드 내 컨트롤 플레인 기능 구성 및 워커노드와의 연결이다. 따라서, 실험을 통해 확인된 선행 단계들에 사용되는 시간을 최소화할 수 있어야 한다. 제안 방식은 제시한 단계 중 가장 큰 비중을 차지한 OS 구성을 환경 구축 이전에 진행한다. 마스터 노드와 워커 노드 구축 단계가 두 단계로 단순화되며, 전체적으로 인프라 구축 시간이 기존 대비 절반 이하로 단축됨을 확인할 수 있다.

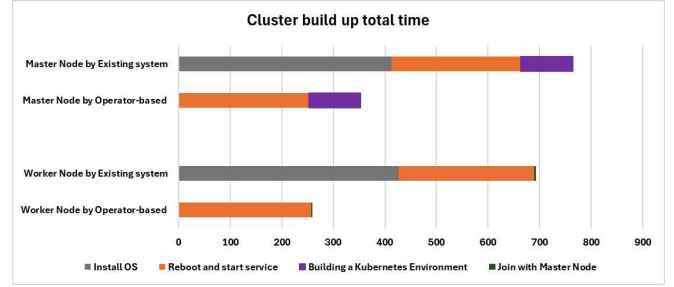


그림 3. 기존 방식과 Operator 기반 클러스터 구축 시간

IV. 결론

본 논문에서는 베어메탈 기반 쿠버네티스 클러스터 자동화 구축을 위한 Operator 시스템을 제안하였다. 제안 시스템은 Operator를 통해 OS 설치를 분리하고 수동 작업을 자동화함으로써, 기존 방식의 유연성 부족과 관리자 개입 문제를 해소하였다. 실험 결과, 베어메탈 기반 쿠버네티스 클러스터 구축에 소요되는 시간이 절반 이하로 감소할 수 있음을 보였다. 추후 성능 평가에서 확인된 다양한 요소들에 대하여 최적화 방안을 연구하여 베어메탈 기반 인프라 구축 방법을 확장할 계획이다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학ICT연구센터(ITRC)의 지원(IITP-2025-RS-2023-00258649, 50)과 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.RS-2024-00398379, 50)

참 고 문 헌

- [1] I. Syrigos, N. Makris, and T. Korakis, "Multi-Cluster Orchestration of 5G Experimental Deployments in Kubernetes over High-Speed Fabric," in Proc. 2023 IEEE Globecom Workshops (GC Wkshps), Kuala Lumpur, Malaysia, Dec. 2023.
- [2] Canonical, "Kubernetes and Cloud Native Operations Report 2022," Canonical Ltd., pp. 9-15, May 2022. (<https://juju.is/cloud-native-kubernetes-usage-report-2022>)
- [3] R. M. Redoli and A. Ullah, "Kubernetes in the Cloud vs. Bare Metal: A Comparative Study of Network Costs," Proc. 2025 Int. Conf. Adv. Inf. Netw. Appl. (AINA), Taipei, Taiwan, Apr. 2025.
- [4] K. Klues and S. Chatterjee, "Keynote: Accelerating AI Workloads with GPUs in Kubernetes," presented at KubeCon + CloudNativeCon Europe 2024, Mar. 2024. [Online]. Available: (<https://kccnceu2024.sched.com/event/1YhI5>)
- [5] Cluster API Book, "Kubernetes Cluster API", 2024, (<https://cluster-api.sigs.k8s.io/>).
- [6] Metal3 user-guide, "Metal3", 2024, (<https://book.metal3.io/>).