

오픈소스 MANO 에서의 자동 복원 기능 분석

이장원, 김영한*

숭실대학교, *숭실대학교

jangwon.lee@dcn.ssu.ac.kr, *younghak@ssu.ac.kr

Analysis of automatic restoration in open source MANO

Jangwon Lee, Younghan Kim*

Soongsil Univ., *Soongsil Univ.

요약

클라우드 인프라는 다양한 서비스에서 플랫폼으로 이용되고 있으며, 오픈소스에 대한 LCM(Life Cycle Management)을 제공한다. 이와 관련하여, ETSI(European Telecommunication Standards)도 클라우드 인프라의 LCM을 위한 기능 및 절차 등을 정의하고 있다. 한편, 대표적인 오픈소스 플랫폼인 오픈스택에서도 LCM을 제공하기 위한 기능들을 제공하고 있으나, ETSI NFV-REL에서 정의된 다양한 경우의 LCM을 위한 기능들을 모두 제공하고 있지는 않다. 본 논문에서는 오픈소스 클라우드 환경에서 LCM을 제공하기 위해 필요한 기능들을 분석하고, 추가 필요한 기능들에 대한 설계 및 구조를 제안하고자 한다.

I. 서론

가상화 기술이 발전함에 따라 라우팅, 파이어월과 같은 네트워크 기능들을 고성능 서버에 가상화해 동작시키는 NFV(Network Function Virtualization) 구조가 ETSI(European Telecommunication Standards Institute)에 의해 정의되었다. 이후, VNF 관리 및 오픈스택을 위해서 NFVO(NFV Orchestrator), VNFM(VNF Manager), VIM(Virtualized Infrastructure Manager)으로 구성된 MANO(Management AND Orchestration)가 정의되었다. MANO를 구현하고 있는 오픈소스 프로젝트에는 크게 LFN의 ONAP, ETSI의 OSM, 오픈스택의 Tacker가 있다. 각 프로젝트의 목표에 따라 MANO 구조와 LCM(Life Cycle Management)을 위한 스케일링, 재생성, 헬링과 같은 기능들이 구현되고 있다. 하지만 대부분의 복원 기능 및 절차가 VNF 서비스 문제 혹은 VIM 리소스 문제로 좁혀져 있다. ETSI NFV-REL 표준에 의하면, VIM 혹은 VNF 업데이트 시 복원 방법, HA(High Availability) 관련 스위칭을 위한 복원 방법 등의 다양한 케이스[1][2]가 정의되어 있다. 본 논문에서는 문제 및 유지에 대한 관리 프로젝트인 OPNFV Doctor로부터 파생된 Fenix 프로젝트에서 구현한 오픈스택 기반 호스트 유지관리를 위한 Workflow 절차와 오픈스택의 MANO 프로젝트인 Tacker를 이용하여 자동 복원을 위한 일반적 절차에서 필요한 기능을 분석하고자 한다. 먼저 Fenix와 Tacker의 구조 및 기능을 분석하고 절차와 관련된 Tacker의 기능과 절차에서 필요한 기능들을 제안하여 일반적인 자동 복원 기능을 구성할 수 있도록 한다.

II. 관련연구

오픈스택은 대표적인 프라이빗 클라우드 오픈소스 프로젝트로 2010년 NASA에서 시작하였으며 현재 다양한 기업들이 참여하여 Victoria 버전이 진행되고 있고, Tacker는 오픈스택 NFV Orchestration 프로젝트로 2015년 Kilo 버전부터 진행되었다. Tacker는 OASIS의 TOSCA를 기반으로 자체적으로 Descriptor 스펙을 정의하였고 VNF, NS(Network Service), VNFFG(VNF Forwarding Graph)를 생성 및 관리하는 기능이 구현되어 있다. 현재 ETSI NFV-SOL에 기반한 VNF 패키지, VNFLCM과 NFV-IFA 029에서 정의된 CNF를 위한 CISM(Container Infrastructure Service Management) 구조가 논의 및 개발되고 있다.[3]

Fenix 프로젝트는 OPNFV Doctor에서 파생된 프로젝트로 목표는 ETSI FEAT03인 NFVI 유지 관리를 위한 일반적인 Workflow를 정의하고 관리하여 절차에 따른 정보 및 VNF 관리를 VNFM에 요청한다. 현재 오픈스택 및 쿠버네티스를 기반으로 하는 예제 형식의 Workflow가 정의되어 있어 이 기반을 통해 일반적인 복원 Workflow를 구성할 수 있을 것이라 기대된다. 현재 단점으로는 현재 예제 Workflow가 각 NFVI의 특정 커뮤니티에 요청하는 방식으로 정의되어 있어 확장성이 떨어지는 부분이 있다.[3]

III. 본문

Tacker의 VNFM은 실패, 종료, 모니터링과 같은 기본적인 것들과 다른 오픈스택 프로젝트와 협업 된 헬링, 스케일링, 알람, 리소스 예약 등의 정책이 지원된다. 알람 종류나 리소스 예약 정책이 VNFD에 존재하면 생성단계에서 해당 모니터를 통해 알람 및 추후 관련 기능 동작을 위한 리소스를 만들고 Aodh를 통해 알람이 등록된다. 헬, 스케일, 재생성은 Heat의 동작으로

수행되며, 재생성은 전체 Heat 스택 삭제 후 재생성하는 것이고 힐은 스택 내부의 리소스 일부분을 재생성 하는 것이다. VDU 배치 정책은 한 VNF 의 다중 VDU 에 대한 Anti/Affinity 가 제한적으로 구현되어 있다. 리소스 예약 정책은 Blazar 프로젝트와 연계되는 기능으로, 추후 스케일링 및 이전 기능 등을 위해 자원을 미리 예약해 놓아 필요 자원을 보장해주는 기능이다.

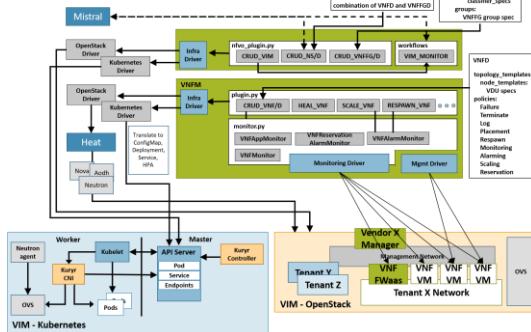


그림 1. Tacker NFVO/VNFM 구조

그림 2 는 Fenix 에서 NFVI 유지 관리를 위해 정의된 workflow 이다. 인프라의 호스트 중 제일 여유로운 호스트 N 개를 선택하여 해당 호스트들의 VNF VDU 들을 옮기고 유지관리를 순차적으로 진행하는 방법을 사용한다. SCALE_IN 과 PREPARE 절차는 선택된 호스트들의 VDU 들을 옮길 수 있는 여유 자원이 전체 호스트에서 부족할 경우, VDU 를 줄이고 선택된 호스트의 VDU 들을 다른 호스트로 이전하는 과정이다. 이 과정에서 VNFM 에 스케일링 기능과 이전 기능이 요청된다. 이 단계에서 문제가 될 수 있는 부분은 VNFM 에서 스케일링 정책의 변화하는 VDU 개수가 유동성이 가능여부이다. 현재 Fenix 에서 구현한 오픈스택 기반 Workflow 에서는 원하는 개수 N 개를 한번에 요청하는 방식이지만 Tacker 의 스케일링 정책은 변화하는 개수가 정해져 N 이 더 클 경우 정책을 여러 번 수행해야한다. 그렇기에 VNFM 이 가질 수 있는 특성들이 시작 부분인 MAINTENANCE 절차에서 작용될 필요가 있다. Fenix 에서 이전 기능은 크게 MIGRATE, LIVE_MIRATE, OWN_ACTION 과 쿠버네티스를 위해 EVACUATE 타입을 추가로 정의하여 VNF 마다 사전에 정의된 이전 기능을 수행한다. 현재 Fenix 에서 VM 인 경우 오픈스택 Nova 의 이전 기능을 통해 수행되지만, OWN_ACTION 을 사용하면 VNFM 에서 특정한 이전 기능을 수행할 수 있게 하였다. Nova 의 이전 작업은 많은 부분이 고려되지 않은 단순한 이전이기에 OWN_ACTION 을 기반한 VNFM 에서의 이전 기능 구현 혹은 Fenix 자체적으로 이전 기능을 위한 3rd 파티 프로젝트를 호출할 수 있게 정의할 필요가 있다. 또한 이전 작업이 진행되기 때문에 HA VNF 의 VDU 일 경우 스위칭을 하는 기능이 VNFM 에 있어야 한다. 또한 배치 정책이 등록 되어있는 VDU 를 배치할 수 없을 때를 위한 대체 방법이 필요하며 Tacker 에는 이러한 기능들이 없기에 구현이 필요하다. PLANNED 절차는 PREAPRE 절차와 동일하며 순차적으로 모든 호스트가 수행된다. 호스트 유지 관리를 위한 절차에서만 동일한 것이고 Fault 가 발생한 상황에 대한 절차 경우에는 이전 작업이 아닌 재생성 혹은 힐링의 작업이 가능하도록 정의 되어야한다. Tacker 힐링 정책은 하나의 VDU 그룹 리소스(VNFC)를 재생성하는 것으로 정해져 있기에 힐링의 영역을 자유롭게

구성할 수 있도록 정책을 강화해야한다. 유지관리를 위한 전체 과정이 완료된 COMPLETE 절차에서는 SCALE_IN에서 줄어든 VDU 수를 복원하는 기능이 VNFM에서 수행되어야 하며 HA가 고려된 VNF에서 특정 호스트의 VDU를 ACTIVE로 바꾸고 싶을 경우 스위칭 하는 과정이 추가적으로 필요하다.

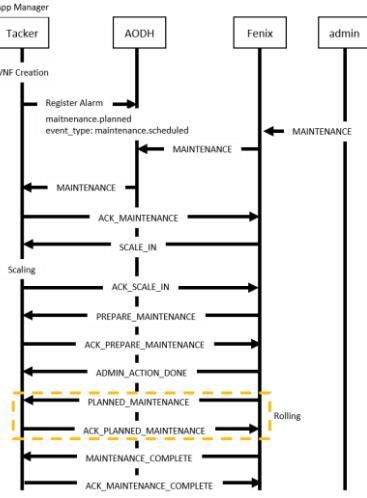


그림 2. Fenix 호스트 유지 관리 Workflow

IV. 결 론

본논문에서는 많은 오픈소스 MANO 프로젝트에서 발생하고 있는 복원의 범위가 좁아지는 문제가 있다. 이를 개선하기 위해 일반화된 전체 절차로 활용될 수 있는 Fenix 의 호스트 유지관리 절차와 오픈스택 Tacker 를 분석하면서 필요한 기능들을 제안하였다. 언급된 필요 기능들이 구현되고 Fenix 의 예제를 드라이버 형식의 일반화된 Workflow 로 발전시켜 구현한다면 관리자는 다양한 복원 절차를 구분 짓지 않고 손쉽게 가능하게 될 것이라 기대된다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원 (No.2020-0-00946, 하이브리드 클라우드 환경에서의 고속, 자동 서비스 복구 및 이전 소프트웨어 개발)과 대학 ICT 연구센터지원사업의 연구결과로 수행되었음(IITP-2020-2017-0-01633)

참 고 문 헌

- [1] ETSI, “Network Functions Virtualisation (NFV); Report on Models and Features for End-to-End Reliability”, ETSI GS NFV-REL 003 v1.1.1, April. 2016.
 - [2] ETSI, “Network Functions Virtualisation (NFV) Release 3; Reliability: Maintaining Service Availability and Continuity Upon Software Modification”, ETSI GS NFV 006 v3.1.1, February. 2018.
 - [3] 양현식, “NFV 의 고가용성 기술 동향”, OSIA Standards & Technology Review, pp.10-15, Sep. 2019