

컨테이너 환경에서의 고속 장애 복구를 위한 워크플로우 설계

양현식, 김영한*

송실대학교

yangun@dcn.ssu.ac.kr, *younghak@ssu.ac.kr

A Design of Workflow for Fast Fault Recovery in a Container Infrastructure

Yang Hyun Sik, Kim Young Han*

Soongsil Univ

요약

본 논문은 컨테이너 기반 클라우드 환경에서 장애가 발생하는 경우 서비스의 빠른 복구를 위한 워크플로우 관리자를 설계하고 연동구조를 제시한다. 가상인프라 위에 배포된 컨테이너 환경에서의 서비스 제공을 위한 절차들을 정의하고 장애 종류 및 상황에 따른 workflow를 정의하였다. 이를 통해 제안 환경에서 요구되는 워크플로우의 동작 구조 및 유즈케이스를 확인하였으며, 기존 구조에 비해 다양한 환경에서의 적합한 장애 복원 기능을 제공할 수 있음을 보였다.

I. 서론

클라우드 환경은 다양한 서비스를 제공하기 위한 인프라로 사용된다. 클라우드 인프라는 VM이나 컨테이너 기반 인프라로 구성될 수 있으며, 각 인프라를 관리하기 위한 오케스트레이터가 존재한다. 각 클라우드 환경의 오케스트레이터는 생성된 인스턴스에 대한 라이프 사이클 관리를 기능을 제공한다. 각 클라우드 인프라에서는 오케스트레이터를 통해 각 인프라에 배포된 인스턴스를 관리 하며 이를 위해 인스턴스에 대한 모니터링 기능 및 복원 기능을 제공한다. 기본적으로 각 인프라는 개별 오케스트레이터를 가지고 있으며, 대표적인 오케스트레이터로는 OpenStack, Kubernetes 등이 있다[1,2]. 한편, 컨테이너는 VM에 비해 이식성이 좋고 가벼우며, 베어메탈 및 VM 기반 인프라에 배포 가능하다[3]. 최근에는 아마존이나 MS Azure와 같은 VM 인프라 위에 컨테이너가 배포되는 형태의 클라우드 구조도 제안되고 있다. 단일 클라우드 환경에서는 하나의 오케스트레이터를 이용하여 인스턴스에 대한 라이프 사이클 관리를 가능하다. 그러나 다중 인프라로 구성된 환경에서는 각 인프라의 상태 정보를 모르기 때문에 라이프 사이클 관리에 한계가 있으며, 전체 환경에 대한 라이프사이클 기능 제공이 어렵다. 따라서 다중 인프라가 공존하는 환경에서 라이프 사이클 관리를 위해서는 각 인프라를 통합 관리 할 수 있는 구조가 필요하다. 본 논문에서는 VM 위에 배포된 컨테이너 기반 클라우드 환경에서 다양한 환경을 고려한 통합 관리 기능인 Maintenance Manager를 설계하고 이에 따른 유즈케이스를 소개하고자 한다.

II. 관련 연구

클라우드 환경은 인프라 종류에 따라 VM 기반 인프라와 클라우드 기반 인프라로 구성된다. 각 인프라는 개별 인프라를 관리하기 위한 오케스트레이터가 있으며, 대표적인 오케스트레이터로는 OpenStack Tacker와 Kubernetes가 있다. OpenStack에서는 기본적으로 인스턴스에 대한 관리 기능을 제공하며, OpenStack Tacker를 통해 VNF 및 그 이상의 서비스에 대한 라이프 사이클을 제공하기 위한 모니터링 기능 및 복원기능들을 제공한다[4]. OpenStack Tacker에서는 VM의 생성을 위한 VNFD를 제공하며, 이를 통해 인스턴스 생성시, 인스턴스에 대한 모니터링 기능, 자동복원 기능 등을 정의하여 제공 할 수 있다.

이 외에도 Zabbix와 같은 3rd Party 모니터링 툴도 연동 가능하다. Kubernetes에서는 노드의 장애 관리 및 서비스의 장애 관리를 위해 Kubelet을 이용하여 상태정보를 주기적으로 수신한다. 이를 통해 현재 리소스 사용량 및 컨테이너 정보들을 확인하여 장애 발생시 노드의 장애 복원 절차를 시작한다. 이외에도 프로메테우스와 같은 3rd Party 모니터링 툴을 이용하여 장애를 모니터링 할 수 있다. 기본적인 Kubernetes 환경에서의 장애 탐지 기술은 주기적으로 상태 정보를 업데이트 하고 업데이트 정보에 따라 Kubernetes Master에서 장애 정보를 확인하여 노드의 상태 정보를 변경하는 절차로 구성된다. 다중 인프라로 구성되는 환경에서 라이프 사이클을 위해 고려해야 하는 요소들은 그림 1과 같다.

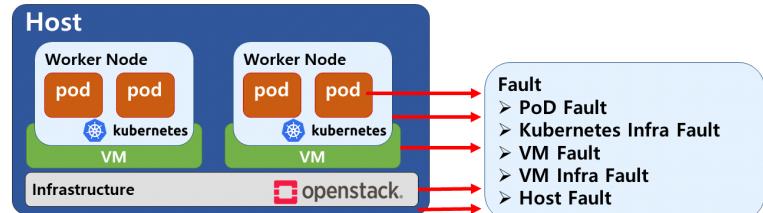


그림 1. VM위에서의 컨테이너 클러스터 구조

그림 1과 같이 컨테이너 클러스터 구성을 위해 여러개의 인프라 및 관리 계층이 존재한다. 각 계층은 독립적으로 존재하지만 장애가 발생하는 경우에는 계층 구조에 따라 영향을 받을 수 있다. 그러나 현재 구조에서는 각 인프라 별 관리자가 독립적으로 존재하기 때문에 장애로 인해 인프라에 영향을 미치기 전까지는 장애를 확인할 수 없다.

III. 제안 기법

본 논문에서는 VM위에 배포된 컨테이너 클라우드 환경에서 장애 관리 및 복원을 위한 워크 플로우 생성 및 관리 기능을 설계하고 유즈케이스를 소개하고자 한다. 그림 1과 같이 VM위에 하나의 클러스터가 구성되며, Kubernetes Master 및 Worker Node는 하나의 VM을 호스트 노드로서 인지한다. 예를 들어 한 개의 Master Node와 20개의 Worker Node가 구성 되기 위해서는 21개의 VM이 필요하다. 위와 같은 클라우드 구조에서 장애 복구를 빠르고 효율적으로 제공하기 위해서는 장애 원인 및 위치에 따

라 정해진 워크플로우를 기반으로 빠르게 장애를 탐지하고 복원할 수 있는 구조가 필요하다. 또한, 개별 오케스트레이터를 이용하는 것이 아니라 각 인프라의 오케스트레이터를 통합하여 관리하고 제공할 수 있는 구조가 필요하다.

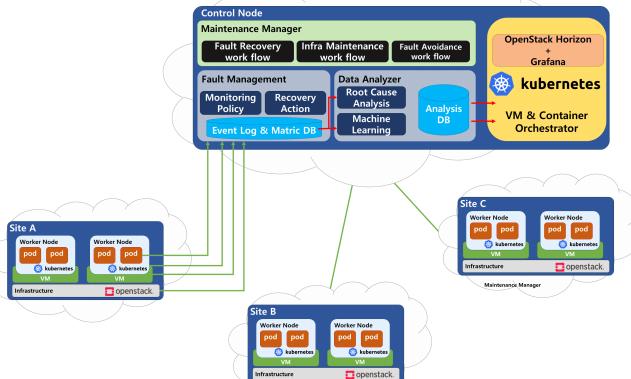


그림2 . 고속 장애 복구를 위한 워크플로우 매니저

그림2는 고속 장애 복구를 위한 워크플로우 매니저이다. 워크 플로우 매니저는 각 장애 종류 및 복원 절차에 따라 다양한 워크플로우를 제공하며, 장애 탐지 결과 및 장애 복원 절차 연동을 위해 VM 및 Container 관리자와 연동된다. 장애 탐지 정보 획득을 위해 장애 관리기능과 연동되며, 이후 워크 플로우 관리 및 복원을 위해 개별 Orchestrator와 연동되는 구조를 가진다. 워크플로우는 크게 세 가지로 구분된다. 첫 번째는 Infra Maintenance workflow이다. Infra Maintenance workflow는 인프라의 업그레이드 및 유지 보수를 위한 작업이 필요한 절차들을 정의한 것으로 현재 노드의 상황을 고려하여 서비스를 이전하면서 각 인프라에 대한 유지보수를 진행한다. Fault Avoidance Workflow는 장애가 발생할 확률이 미리 정해놓은 지점을 넘어가거나 특정 리소스의 사용량이 증가하는 경우 장애를 피하기 위해 서비스를 관리하는 절차를 의미한다. 마지막으로 Fault Recovery Workflow이다. Fault Recovery Workflow는 장애가 발생한 경우, 장애를 빠르게 복구하기 위해 필요한 절차를 의미한다. 그림3은 장애 발생 시 필요한 Fault Recovery Workflow 절차이다.

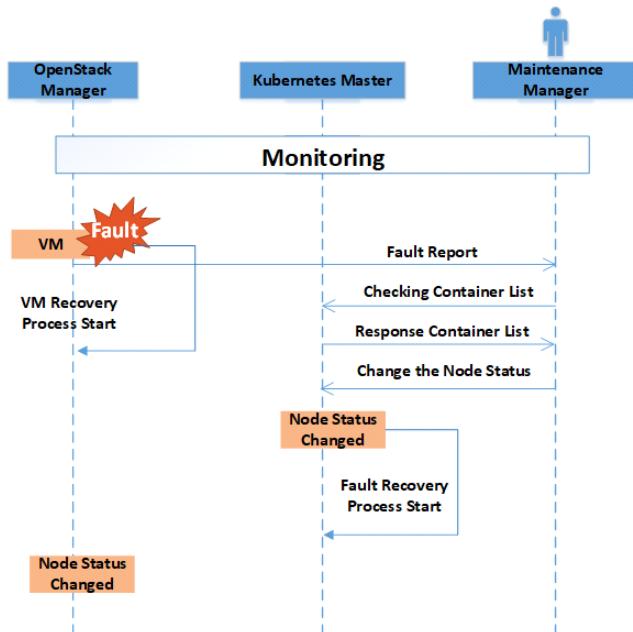


그림3 . 다중인프라 환경에서의 장애 복구 워크 플로우

그림3은 VM위에 배포된 컨테이너 클라우드 환경에서 워크 플로우 기반 장애 복원 절차를 소개하고 있다. 일반적인 환경에서는 단일 클라우드 오케스트레이터를 통해 장애를 복원하지만 그림3과 같이 다중 클라우드 환경에서는 장애 지점에 따라 통합 워크플로우 매니저를 이용하여 빠르게 복원할 수 있다. 그림3에서는 VM의 장애가 발생한 경우로, VM의 장애가 결국 위에서 동작하는 컨테이너 노드 및 Pod에 영향을 미치기 때문에 장애 정보를 컨테이너 환경에서 인지 하기 전에 전달하는 유즈케이스이다. 또한, 통합 워크플로우를 이용하면, 제안구조와 같은 환경에서는 컨테이너의 클러스터 확장을 통한 장애 복원 기능도 제공 가능 하다.

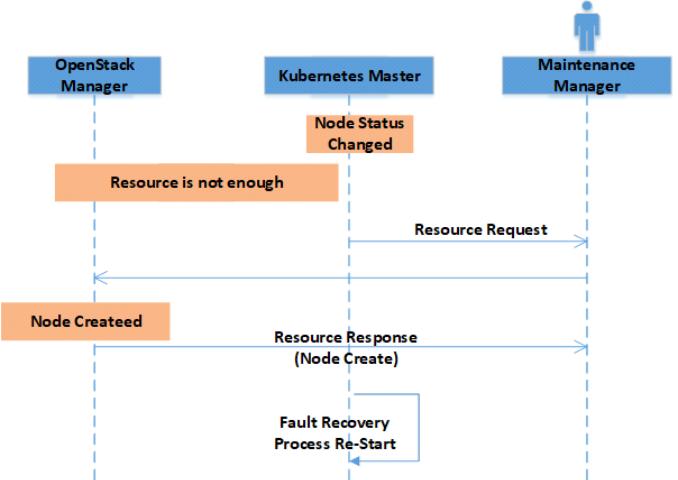


그림4 . 다중인프라 환경에서의 노드 확장 시나리오

그림4는 장애 복원 시 현재 컨테이너 클라우드의 리소스가 부족한 경우로 이때는 장애 복원을 위해 새로운 노드의 생성을 요청하게 된다. 리소스가 부족한 경우 기본적으로는 현재 동작하는 리소스를 줄여서 공간을 확보하지만 제안 구조를 통해 새로운 노드를 생성하고 생성된 노드로의 배포가 가능하다.

III. 결론

본 논문에서는 다중 인프라 구성된 클라우드 환경에서 장애 및 유지보수를 위한 구조를 설계하고 그에 따른 유즈케이스를 소개하였다. 단일 클라우드 환경에서는 하나의 클라우드 관리자로부터 관리가 가능하지만 다중 클라우드 환경에서는 통합된 장애 관리 및 유지보수 기능을 제공할 수 있는 구조가 필요함을 확인하였다. 특히 이를 통해 기존 환경보다 빠르게 장애를 탐지하고 유지보수가 가능 함을 확인 할 수 있었다. 추후 연구에서는 실제 워크플로우별 기능들을 설계하고 구현하고자 한다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획 평가원의 지원을 받아 수행된 연구임 (No.2020-0-00946, 하이브리드 클라우드 환경에서의 고속, 자동 서비스 복구 및 이전 소프트웨어 개발)

참 고 문 헌

- [1] Openstack <https://wiki.openstack.org>
- [2] Kubernetes <https://kubernetes.io/>
- [3] P. R. Desai. A survey of performance comparison between virtual machines and containers. ijcseonline.org, 2016.
- [4] Openstack Tacker: <https://wiki.openstack.org/wiki/Tacker>