

Openstack을 이용한 Cloud Computing 환경 구축 및 활용

김영훈, 지호영, 문봉교
동국대학교 컴퓨터공학과

E-mail : terryaaaa@naver.com, hoyungji@gmail.com, bkmoon@dgu.edu

The Construction and Utilization of Cloud Computing Environment with Openstack

YoungHoon Kim, Hoyoung Ji, BongKyo Moon
Dept. of Computer Science and Engineering
Dongguk University

요 약

Cloud Computing 환경은 사용자 하여금 네트워크를 통하여 운영플랫폼, 저장매체 등이 운영 가능한 컴퓨터 자원을 신속하게 이용할 수 있는 컴퓨팅 환경이다. 이 환경을 이용하여 사용자들에게 가상의 컴퓨팅 공간을 만들어 resource를 사용하게 만들어주는 IaaS(Infrastructure as a Service)가 급증하고 있다. OpenStack은 Cloud Computing 환경을 이용하여 사용자 하여금 *aaS제공을 가능하게 해주는 오픈소스 소프트웨어 프로젝트이다. 이 연구는 Openstack을 활용하여 Cloud환경을 구축하고, 이를 활용 하는 방안에 대한 연구이다. GitLab를 이용한 교내 GitLab Service Launching을 시도하고, Hadoop을 통하여 Cloud 환경을 활용한 분산처리 시스템을 구현, Cloud의 활용 방안을 탐구한다.

1 서 론

Cloud Computing 환경은 사용자들에게 공유된 네트워크, 서버, 저장매체, 운영플랫폼 등의 운영이 가능한 여러 computing resource를 빠르게 제공해줄 수 있는 컴퓨팅 시스템 환경이다. 이를 네트워크를 통해 확장하여, C9, Amazon의 AWS 등 단순 볼륨제공을 넘어서, Cloud Computing 환경을 이용하여 다양한 Resource 제공을 인터넷을 통하여 일반 사용자에게 제공 해주는 서비스가 급증하고 있고, ICT 산업에서 큰 주목을 받고 있다. 그중에서도 Openstack은 Hypervisor기반의 가상화를 제공하는 오픈소스 프로젝트이며, 여러 세부 프로젝트로 구성되어있다. Openstack을 이용하여 구축한 Cloud Computing환경은 여러 서버의 자원을 한 대로 모아 사용하게 해주고, 이를 통한 Computing Resource 제공 서비스는 이용자로부터 가변적인 자원운용을 가능하게 한다. 본 연구에서는 Openstack을 이용하여 Cloud Computing환경을 구축하고, 어떻게 활용 될 수 있는지에 대한 방안을 탐구하고, 결과를 제시하고자 한다.

Openstack의 자원관리, 네트워킹, 인증 등을 담당하는 내부 프로젝트들이 Cloud환경에서 관리자와 사용자가 Computing 자원을 자유롭게 사용하고 관리하게 해준다. 5대의 서버 자원을 활용하여, Openstack을 이용한 Cloud Computing 환경을 구축한다. 이렇게 구축된 Cloud 에 대하여, 여러 방면의 활용 방안을 탐구한다. Openstack의 내부 프로젝트들이 Cloud를 활용하기 위해 제공하는 VM, 컴퓨팅 자원 관리 기능 등을 이용해 보고, Cloud 내부에서의 서버 활용방안, Cloud의 분산처리 기능 등, Cloud에 대한 활용 방안을 연구한다. 실 예로, 교내 학생들에게 리눅스, 윈도우 등 필요한 플랫폼을 서비스 받아 사용해볼 수 있는 기회를 제공하고, 추가로 GitLab 서버를 구축하여 사용자간에 소스코드를 공유하고 수정 할 수 있는 환경을 조성한다. 분산 처리의 경우, Hadoop을 사용하여 생성된 인스턴스들을 Cluster로 묶어서 데이터를 처리하는 방안을 시험해본다.

2. 본 론

2.1. OpenStack

Cloud Computing 환경 구현을 위해 본 연구에서 사용하는 open-source project OpenStack은 다양한 소프트웨어로 구성되어있고, 클라우드 환경 구성자의 목적에 맞게 이를 선택할 수 있다. 본 연구에서 다루는 프로젝트는 아래와 같다.

- ① Compute (Nova) : 오픈스택 클라우드 환경의 컨트롤 타워 역할을 한다. 가상 컴퓨터 인스턴스들을 생성, 제어 및 운영하기 위한 서비스이다.
- ② Dashboard (Horizon) : End-user, Administrator가 서비스를 이용&제어할 수 있는 웹 인터페이스 Dashboard를 제공한다.
- ③ Network (Neutron) : 디바이스 인터페이스와 오픈스택 서비스간의 네트워킹 기능을 제공하며, 사용자로 하여금 개인의 네트워크 환경 구성을 가능하게 해준다.
- ④ Block Storage(Cinder):클라우드에 호스트된 VM들의 스토리지 제공.
- ⑤ Object Storage (Swift) : 블록 스토리지와는 별도로 구축되는 클라우드 오브젝트 스토리지. 파일을 저장하고 로드 할수 있게 해주는 컨테이너 역할을 한다.
- ⑥ Image (Glance) : 가상 디스크 이미지들을 저장, 등록, 관리, 전달하기 위한 오픈소스 소프트웨어
- ⑦ Identity (Keystone) : 모든 오픈스택 서비스에 대한 인증관련 서비스를 관리하는 시스템.
- ⑧ Orchestration (Heat) : 문서 형태의 Template으로 Stack을 Launch하여 인스턴스를 한 번에 생성하는 조율 역할을 한다.
- ⑨ Metering Service(Ceilometer) : Cloud에서 사용되는 자원을 모니터링 하여, 사용자 및 관리자가 사용한 자원에 대한 정보를 볼 수 있게 한다.

이 서비스들은 각각의 역할에 충실하여, 필요한 부분에서 서로와 연계된다. 상호관계에 대한 지도는 Figure [1]에 나와 있다.

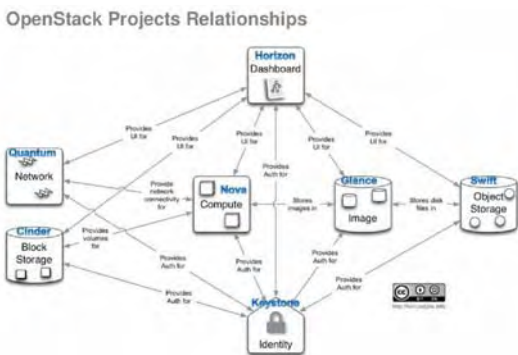


Figure 1 . OpenStack Structure Diagram

OpenStack에는 Nova-network와 Neutron, 이 두 가지 방식으로 네트워크를 서비스 한다. 첫 번째 nova-network는 기존의 nova 프로젝트에서 네트워크를 관리 해주는 것으로 관리하기가 neutron보다 쉬우나 외부와 통신을 할 때 사용하는 floating IP 범위가 서버와 같은 IP 대역을 사용하기 때문에 사용 가능한 IP 대역이 상당히 제한적이다. 두 번째는 neutron으로 이는 SDN(software Defined Network)개념을 사용한 것으로 가상의 스위치를 만들어 네트워크를 분리하고 가상의 라우터를 만들어 해당 네트워크를 외부 등으로 연결하는 방식의 서비스이다. 클라우드를 제공받을 대상 집단이 소규모라면 nova-network로 네트워크 환경을 서비스해도 문제가 없으나 규모가 커지고 집단 내에서도 소규모의 네트워크를 따로 할당해주고 좀 더 정밀한 관리가 필요하면 neutron의 사용은 필수적이다.

OpenStack은 이 서비스들의 완전한 운용을 위해 Multi-Node Structure를 지원&권유하고 있다. 본 연구에서는 가장 많이 사용되는 Three-Node Architecture을 사용한 클라우드 환경 구축을 최종 목표로 한다. 세 가지 종류의 노드는 각자 역할을 가지며, 그중 가장 중요한 첫 번째 노드는 Controller Node로 플랫폼 전체를 제어하는 역할을 한다. 사용자, 관리자의 명령을 받아들여 명령을 어떤 자원을 이용하여 어떻게 처리할 것인지 결정하는 클라우드 관리 노드이다. 가상 머신 생성, 관리를 위한 Nova, 사용자 인증을 위한 Keystone 등이 이 서버에 포함된다. 두 번째는 Network Node로, 클라우드 서비스 플랫폼의 내부 가상 네트워크 구성 및 방화벽, 유동 IP 관리 등의 작업을 수행하는 Neutron의 에이전트들이 실행되는 서버이다. 세 번째 서버는 Compute Node로, 클라우드 시스템의 저장고 역할을 한다. 실제 인스턴스들을 생성하고 가상 머신들의 저장 이 이루어지는 물리적 서버이다. 목적에 따라 Compute Node는 1개 이상으로 구성 될 수 있으며, 그를 운용하기 위한 추가 자원이 요구된다. 본 연구에서 구축한 Cloud환경의 논리적 구성은 Figure [2]와 같다.

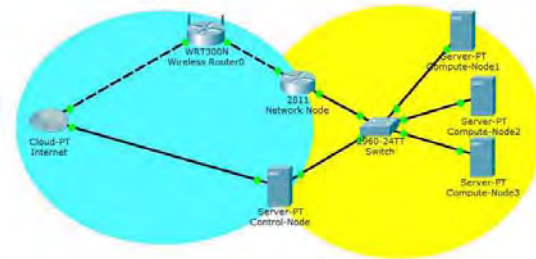


Figure 2. Network Topology

2.2. GitLab

GitLab은 Git의 원격 저장소 기능과 이슈 트래커 기능들을 제공하는 소프트웨어다. Git은 사용자의 편의성이 떨어지고, GitHub Enterprise는 유료 서비스인 것 과 동시에 요

구되어지는 자원이 상당히 크다. 이에, 오픈소스 이면서 사용자에게 어느 정도 편의성을 제공하는 GitLab을 선택했고, 가상화 서버로 구현하여 교내 학생들이 소스코드를 공유하고 수정할 수 있는 환경을 조성한다. 또한 이를 통해, 기존의 물리적 서버를 가상화 자원으로 관리하고 사용하는 방안을 시험해본다.

2.3. Hadoop

Hadoop은 분산 처리 환경에서 막대하게 큰 양의 데이터를 다루고 저장하는 것을 도와주는 오픈소스 프레임워크이다. Hadoop의 분산 파일 시스템은 서버간의 빠른 전송속도를 제공하며, 이로 인해 낮은 자료손실률을 보이며, 시스템을 Fault Tolerant 하게 운영한다. 빅데이터의 주목과 함께 많은 사람의 관심을 받는 프레임 워크중 하나이다. 가장 잘 알려져 있어 활용범과 예시가 많이 제시되어있고, 사용하기 쉬운 장점이 있어 선택하였다. 가상머신으로 생성된 여러 인스턴스들을 한 대로 묶어 데이터가 주어졌을 때, 데이터를 분산처리 하는 환경을 시험해본다.

2.4. 기대 효과

클라우드 환경 구축방안과, 다양한 활용방법에 대해 알 수 있게 된다. 또한, 실 예를 통해 동국대 내에 자체 클라우드 컴퓨팅을 경험해 볼 수 있는 환경을 조성함으로써, 학생들이 프로젝트나 실습 등을 진행함에 편리함과 유익함을 제공해준다. 어디서든 사용할 수 있는 독자적인 가상 개발 환경을 구성할 수 있게 하여, 일의 효율성을 높이고, 팀 프로젝트시 팀원 간의 소통을 용이하게 한다.

2.5. 실험 및 결과분석

본 연구에서는 5대의 서버자원과 1대의 고속스위치를 사용하여 테스트베드를 구성한다. 아키텍처는 1대의 Control-node, 1대의 Network-Node, 그리고 3대의 Compute-node로 이루어져 있다. Keystone 프로젝트를 이용해서 각 사용자들과 서비스들의 인증을 하며, Nova가 각 Compute-node의 자원을 모아 총괄적으로 관리하여 인스턴스를 생성하게 해준다.

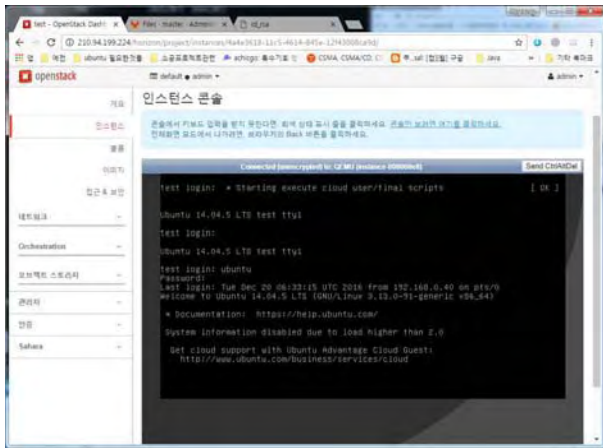


Figure 3. Instance Console

Figure [3]은 Nova를 통해 생성한 인스턴스의 모습으로, Dashboard에서 VNC를 통해 콘솔에 접속한 화면이다.

Instance간에 네트워크 환경은 Neutron Project를 통해 생성 및 관리하게 된다. Plugin 으로 OpenVSwitch를 사용하여하며, 인스턴스 들을 연결하는 가상의 네트워크를 생성하며, 외부와의 통신을 위해 Floating IP를 할당 하는 등 네트워크 관련 프로세스를 총괄한다. 이렇게 Neutron의해 생성된 가상화 네트워크는 Figure [4]와 같이 Network Topology 형태로 확인할 수 있다.

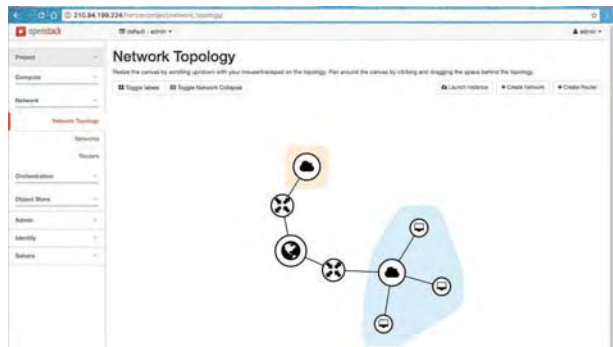


Figure 4. Neutron Network

사용자에 따라 할당되는 저장 공간을 이용 할 수 있게 해주는 Object Storage서비스를 Swift Project통해 구축하였다. 저장 방식은 Swift Proxy를 통해 Load Balancing 되어 저장되며, 각각의 사용자들은 컨테이너 단위로 분류된 저장 공간을 이용한다. Figure [5]는 하나의 컨테이너를 보여주는 예로, 컨테이너 안에 폴더나 파일들을 저장할 수 있고, 이를 다른 사용자와 공유할 수 있다.

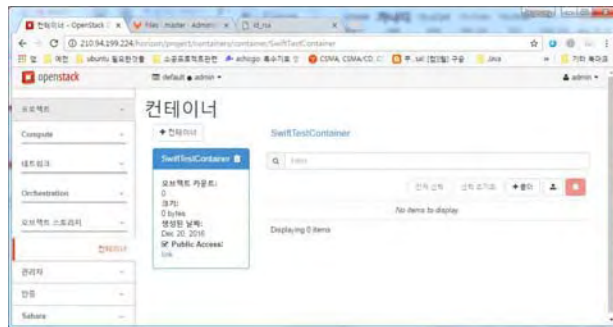


Figure 5. Swift

다음으로는, 가상화된 서버 구축 활용의 예로 사용한 GitLab이다. 해당 서버는 다른 인스턴스들과 마찬가지로 Nova를 통해서 생성되며, 필요한 만큼의 자원을 flavor(자원 사용 템플릿)로 지정하여 가상화된 서버를 생성하게 된다. Figure [6]은 이렇게 생성된 GitLab서버로, 외부 사용자들이 가상화된 서버의 원격 저장소에 웹을 통하여 접속한 화면이다.

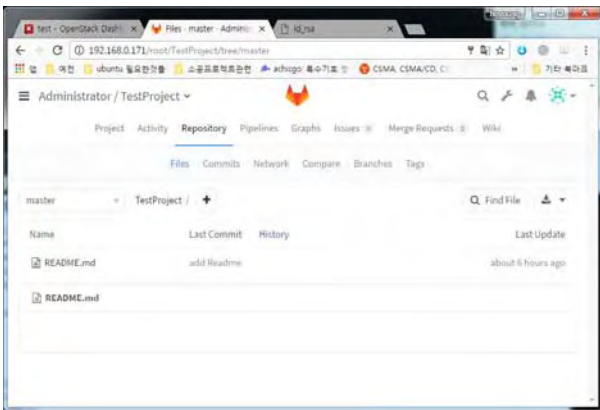


Figure 6. GitLab

GitLab을 통해 사용자들은 GitHub와 같은 환경을 사용할 수 있으며, Private한 소스코드 공유공간을 활용할 수 있게 된다. 물리적 서버에 비해서, 이러한 가상화된 자원의 사용은 개발자의 입장에서 서버의 관리와 운용소요를 줄여 주고, 서버의 확장성이 유연해진다는 장점이 있다.

Hadoop은 통상 1대의 Master Server, 여러 대의 Slave Server로 구성되는 Cluster단위로 구분지어지며, Master Server가 Openstack의 Control-Node처럼 나머지 Slave Server들의 처리 동작을 관리한다. 본 연구에선 Hadoop Cluster를 구현하기 위해 2개의 Ubuntu Instance를 생성하여, 한 개는 Master, 다른 하나는 Slave로 설정하여 시험하였다.

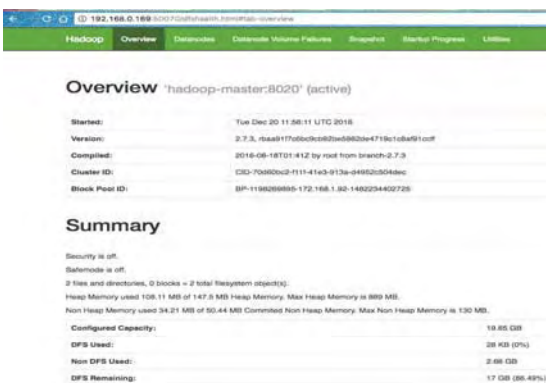


Figure 7. Hadoop

Figure [7]은 Hadoop Cluster의 Master Server에서 제공하는 웹 페이지로, Dashboard역할을 한다. 이 웹 페이지에선 Slave(Datanode)의 개수와 각각의 Slave들에게 오가는 traffic량, Slave들의 작업량, Cluster Summary, Cluster Snapshot등 다양한 기능 들을 제공한다. 많은 자료들이 생기고, 그 변화가 지속적으로 가변 하는 데이터 분산처리 시스템에서, 상황을 확인하기에 아주 효율적인 웹 페이지이다. 가상화 서버와 마찬가지로, Cloud에서 Hadoop Cluster의 사용은 Cluster에 확장성을 부여하여 데이터 처리에 더 많은 자원이 소요될 시에 비교적 쉬운 방법으로 컴퓨팅 자원을 추가 할 수 있고, 소요가 적어질시 손쉽게 자원을

Cluster와 분리하여 다른 작업에 사용할 수 있다.

3. 결론

본 연구에서는 다양한 computing resource를 네트워크를 통하여 공유된 사용자들이 신속하게 활용할 수 있는 Cloud Computing 환경을 구성하고 테스트 해보았다. Cloud 운영체제의 일종인 OpenStack를 이용하여 클라우드 서비스 플랫폼 TestBed를 구축한 뒤, Cloud에 GitLab과 Hadoop을 활용하였다. Openstack의 내부 프로젝트가 제공하는 기능만으로도 Cloud를 활용할 방법이 많다는 것을 알 수 있었다. GitLab을 통해 물리적 서버에 비해 Cloud에서 가상화된 자원에 서버를 운용하는 것이 어떤 이점인지 확인하고, Hadoop 사용 시에도 가변성을 주어 운용에 편리함을 줄 수 있음을 확인하였다.

이러한 활용 방안들을 토대로, 컴퓨터와 관련된 모든 분야에 대해서 Cloud Computing 환경과 관련하여 뭉 수 없는 관계가 되 가고 있음을 보았다. 또한 교내에 Cloud Computing 환경을 구축함으로써, 다양한 실습 환경과, GitLab을 통한 프로젝트의 손쉬운 버전관리를 제공할 수 있었다.

Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원 사업의 연구결과로 수행되었음 (R7116-16-1014)

참 고 문 헌

- [1]www.openstack.org, OpenStack Open Source CloudComputing Software.
- [2]JangHyun Kyung, “ 오픈스택을 다루는 기술,The Art of OpenStack”,길벗 출판사,2014.
- [3]ByungSik Kim,BumChul-Lee,“The Construction and Use of Cloud Service Platform by Using an OpenStack”,2014.
- [4]Grish L S,Dr. H S Guruprasad, "Building Private Cloud using OpenStack",International Journal of Emerging Trends & Technology in Computer Science,Volume3,Issue3,June.2014