

2022-01-23

# 학과 서버 수요와 활용방안

내부 시스템 소개 및 구성

차주형

# 학과 서버 수요와 활용방안

## 내부 시스템 소개 및 구성

### 현재 시스템 구성과 사용 지수

동아리방에 있는 서버 컴퓨터를 있음을 인지하고 생각하고 실제로 적용하기 까지의 내용을 서술하고자 합니다. 기간은 2021년 6월부터 현재 문서 작성중인 2022년 1월 22일 까지의 이야기입니다. 우선 기존 시스템과 현재 변경이 된 시스템이 어떠한 차이점이 있는지, 어떤 장단점이 있는지 서술합니다.

### 기존 시스템, 온프레미스(On-Premise)

과거 클라우드 서비스가 제공되기 이전에 IT 프로젝트를 구성하기 위해서는 우선 운영체제를 설치하고, 특정 프로그램을 설치하는 온프레미스 형식으로 프로젝트를 진행하였습니다. 이 과정에서 한번 시스템을 구축하고 수정이 불가능 합니다. 그래서 컴퓨터를 구매할 때 CPU, GPU, 메모리와 같은 부분은 예산 내에서 항상 최고 사양으로 구매해야 하므로 많은 비용이 발생했습니다. 반면 클라우드 시스템 경우 컴퓨터를 구매하여 사용하는 것이 아닌, 대여 형식으로 제공이 됩니다. 언제나 필요한 만큼 요청만 한다면 언제든지 컴퓨터의 사양을 자유롭게 변경이 되므로 상황에 따라 컴퓨터의 성능을 변경이 가능합니다.

### 타 업체, 회사에서 온프레미스 시스템 구성

온프레미스로 시스템을 구성했을 때 크게 2가지의 방법으로 시스템을 구성합니다. 대표적으로 1개의 컴퓨터당 1개의 서비스 구동하는 방식과 컴퓨터를 가상화 하여 1개의 컴퓨터이지만 여러 개의 컴퓨터로 구성하는 방식이 있습니다. 컴퓨터당 1개의 서비스를 구동하는 방식은 프로그램을 root 나, 사용자 컴퓨터에 설치하는 것이기 때문에 가상화 종류에 대해서 이야기합니다.

### 컴퓨터의 자원, 하드웨어 장비의 가상화

하드웨어 장비의 가상화는 흔히 가상 머신(Virtual Machine, 이하 VM)이라 불립니다. VM은 자체 CPU, GPU, 메모리를 갖춘 시스템에서 가상적으로 동작하는 가상 컴퓨터입니다. VM은 VM마다 완전 별개인 저장공간, 메모리, CPU 코어와 같이 격리를 하는 기술입니다. 그래서 하나의 컴퓨터에서 컴퓨터의

성능만 받쳐준다면 여러 개의 VM 을 생성할 수 있습니다. 각 VM 은 독립된 시스템에서 동작하므로, 우분투를 설치한 뒤 GNOME 을 실행하여 GUI 프로그램을 실행하여 결과를 확인할 수도 있습니다. 그리고 네트워크 라우팅 설정만 적절하게, 올바르게 설정만 한다면 1 개의 컴퓨터에서 여러 개의 운영체제, 소프트웨어를 실행하는 것이 가능합니다.

### 컴퓨터의 프로세스의 가상화

프로세스의 가상화는 하드웨어 장비의 가상화와 비슷해 보이면서 많이 다릅니다. 프로세스의 가상화와 하드웨어의 가상화의 특징적인 차이는 “가상화”의 위치로 둡니다. VM 의 경우는 각각의 운영체제 마다

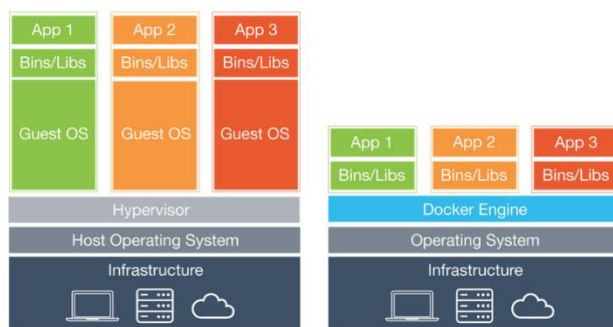


그림 1 VMware 와 Docker

메모리, CPU 를 할당하여 동작하도록 하며, 자신에게 할당된 공간의 사양만큼 사용할 수있어 할당 시 “최소 사양”을 고려해야했습니다. 하지만 프로세스의 가상화 경우는 커널, 운영체제를 “공유” 하기 때문에 전체적인 시스템을 나눠서 사용하지 않습니다.

그렇기 때문에 최소 사양을 지정할 필요 없이 전체의 가상화 프로세스의 합이 물리적인

장비의 성능을 넘지만 않으면 되므로 내부적인 프로그램의 성능이 요구사양이 바뀌더라도 능동적인 대처가 가능합니다.

### 현재 학과 서버 사용 현황

현재 학과내에서 교수님께서 다양한 용도로 서버를 사용하고 계십니다. 예를 들어, 우영운 교수님의 1 학년 프로그래밍 실습에서 사용되는 온라인 저지 시스템, 3 학년 데이터베이스 설계에서 사용되는 데이터베이스 서버, 이광의 교수님의 자료 구조 수업에서 사용되는 웹서버, 그 외에 연구과제에서 사용되고 있는 다양한 컴퓨터가 있습니다. 현재는 1 개의 서비스당 1 개의 컴퓨터가 사용이 되고 있습니다. 그래서 현재 구조에서 가장 큰 단점으로는 사용 중인 컴퓨터가 고장이 나게 된다면 해당 서비스가 중지가 되고, 복구에 많은 장애가 있습니다. 그래서 동아리방에 있는 학과 서버를 가상화를 통해 하나의 컴퓨터가 장애가 생기거나, 문제가 발생하더라도 항상 동작하는 시스템을 구성하고자 하였습니다.

## 목표. 프라이빗 클라우드 서비스

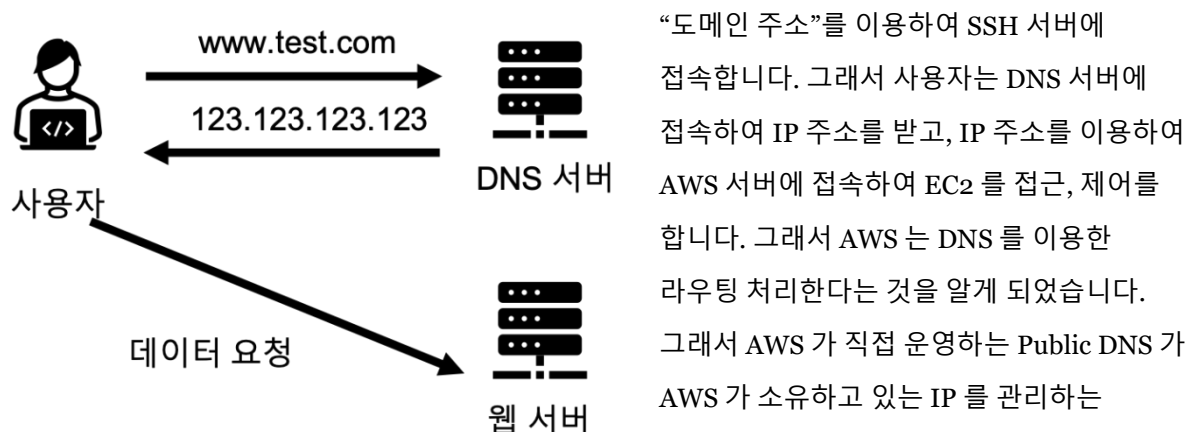
프라이빗 클라우드는 컴퓨팅 서비스와 기업 자체의 인프라넷이나, 특정 데이터 센터 내에서만 사용이 가능한 형태입니다. 즉 외부 기관에서 인프라나, 컴퓨터를 관리하는 것이 아닌, 자체적으로 관리, 유지 보수, 운영을 총괄합니다. 그렇기 때문에 AWS, Azure, GCP 와 같은 퍼블릭 클라우드 보다 많은 제어 권한, 다양한 보안을 프라이빗 클라우드 서비스에 적용하고 관리하기 위해서는 많은 기술이 필요합니다. 그렇기 때문에 2021 년 6 월부터 2021 년 8 월까지 네트워크 설계를 위한 공부를 하였습니다. 2021 년 8 월부터 2021 년 10 월까지 프라이빗 클라우드 서비스를 위한 Kubernetes 와 OpenStack 에 대해서 개념, 학교 시스템에 알맞은 시스템 분석 진행하였습니다. 시스템 분석을 마친 뒤 Kubernetes 가 더 적합하다고 판단하여 Kubernetes 를 적용하여 시스템을 클러스터링 하였습니다.

## 퍼블릭 클라우드 서비스 동작 원리

컴퓨터가 여러대이면서 여러명이 동시에 접근하여 사용할 방법을 생각을 하다가 만약 AWS 처럼 서비스를 구축한다면 현재 교수님께서 운영 중인 온라인 저지 시스템이나, 교육용 서버를 옮겨 일괄적으로 관리를 할 수 있으며, 시스템에 문제가 생기더라도 다른 컴퓨터에서 동작하므로 안정성이 더 높아지게 되면서 좋아지지 않을까? 라는 생각을 하였습니다. 그리고 만약 AWS 처럼 서비스를 구축이 가능하다면 일반 학생 또한 서버를 구매하거나, 고정 아이피를 자유롭게 사용이 가능 할 것이라 생각 하였습니다. 실제로 산학협력관 411 호에서 캡스톤 디자인에 사용할 서버로 사용하거나, 딥러닝을 돌리는 모습을 실제로 확인한 적이 있습니다. 그래서 동의대학교 응용 소프트웨어 공학과 학생이라면 언제든지 사용이 가능한 서버로 만들고자 하였습니다.

## AWS 의 Elastic Compute Cloud(EC2) 서버 동작 원리

누구나 손쉽게 사용이 가능하며, 유동적으로 컴퓨터 자원 관리가 가능하며, 가상머신처럼 사용이 가능한 EC2 가 있습니다. 그래서 EC2 를 실제로 구축하기 위해서 EC2 가 어떻게 동작하는지 이해를 하고 EC2 를 Kubernetes 에 올렸어야 했습니다. EC2 의 경우는 [“어떻게 SSH 로 EC2 에 접속합니까?”](#)



“도메인 주소”를 이용하여 SSH 서버에 접속합니다. 그래서 사용자는 DNS 서버에 접속하여 IP 주소를 받고, IP 주소를 이용하여 AWS 서버에 접속하여 EC2 를 접근, 제어를 합니다. 그래서 AWS 는 DNS 를 이용한 라우팅 처리한다는 것을 알게 되었습니다. 그래서 AWS 가 직접 운영하는 Public DNS 가 AWS 가 소유하고 있는 IP 를 관리하는

그림 2 DNS 서버 원리

방식입니다. 즉, 중간에 IP 가 바뀌더라도 사용자는 도메인으로 접속하기 때문에 문제가 발생하지 않는 이유입니다. AWS 는 초기에 많은 양의 고정 IP 대역을 얻고, 사용자가 서버에 접속할 때만 고정 IP 를 할당하고, 사용하지 않으면 IP 를 회수하는 방식을 사용합니다. “유동적인 IP 주소”라는 이름으로 기능이 제공되고 있다.

### 학교 서버에 Elastic IP 적용이 가능 한가?

구축은 가능 합니다. 하지만 학교에서 제공 가능한 IP 주소는 AWS 서비스와 다르게 제한적이며, 유한적입니다. 그래서 AWS 처럼 Elastic IP 를 구현에는 문제가 있습니다. 그래서 구현을 만약 한다면 L4 스위치 기능을 이용하여 포트포워딩으로 접근하거나, L7 스위치인 리버스 프록시 서버를 활용하여 시스템을 나눠서 구성해야 했습니다.

### 학과 프라이빗 클라우드 서비스

퍼블릭 클라우드 서비스의 경우 컴퓨팅 연산을 하기 위한 EC2 와 운영체제와 데이터를 저장하는 곳은



EBS 로 별개의 시스템에서 동작합니다. 그렇기 때문에 중간에 EC2 시스템이 강제로 중지가 되거나, 데이터 센터가 문제가 생기더라도 AWS 의 EBS 에 문제가 생기지 않는다면 영구적으로 데이터를 보존이 가능합니다. 그리고 2 개 이상의 EC2 에서 1 개의 EBS 를 연결하여 사용하는 것이 가능합니다. 그래서

학과 프라이빗 클라우드 서비스를 구축할 때, 하나의 컴퓨터가 중지가 되더라도 다른 컴퓨터에서 “똑같은 환경”으로 동작 해야 하므로, 데이터를 공유하고, 저장을 할 데이터 센터가 필요하게



그림 3 학과 서버와 나스, UPS

되었습니다. 그래서 집에서 사용 중이던 NAS(Network Attach Storage)를 가져와서 데이터 공유하도록 시스템을 구축하였습니다. 하지만 교내 학교는 일반 가정집 보다 비교적 잦은 횡수로 정전 또는 전기 점검을 하기 때문에 시스템에 갑작스럽게 전기 공급이 끊겨 HDD 나 SSD 가 문제가 발생해 컴퓨터가 고장이 난 경우도 존재합니다. 그래서 모든 데이터를 한 곳에 모으는 것은 매우 위험한 행위입니다. 그래서 시스템이 강제로 전원 공급이 끊기더라도 지속적인 전원 공급이 필요 함을 느끼게

되면서 무정전전원장치(UPS)를 구매하였습니다. 갑작스럽게 정전이 나더라도 약 15 분 정도 NAS 에 전원을 공급하여 시스템이 안정적으로 종료가 될 수 있도록 만들게 되었습니다. 컴퓨터 안에 있는 HDD,

SSD 에 문제가 생기더라도 실제로 사용하는 핵심 데이터는 안정적으로 보관이 되는 시스템을 구축하였습니다. 실제로 처음 구매한 9 월 11 일에 총 2 회의 정전이 발생하였는데 모두 시스템이

전원 공급	AK-NAS이(가) UPS에 연결되었습니다.	2021-09-11 19:58:58
전원 공급	AK-NAS에 연결된 UPS가 배터리 부족에 도달했습니다.	2021-09-11 19:56:52
전원 공급	AK-NAS에 연결된 UPS 장치가 배터리 모드로 들어갔습니다.	2021-09-11 19:55:10
전원 공급	AK-NAS에 연결된 UPS 장치가 AC 모드로 돌아갔습니다.	2021-09-11 16:08:23
전원 공급	AK-NAS에 연결된 UPS 장치가 배터리 모드로 들어갔습니다.	2021-09-11 16:07:57
전원 공급	AK-NAS이(가) UPS에 연결되었습니다.	2021-09-11 15:19:17
전원 공급	AK-NAS에 연결된 UPS 장치가 AC 모드로 돌아갔습니다.	2021-09-11 15:14:19
전원 공급	AK-NAS에 연결된 UPS 장치가 배터리 모드로 들어갔습니다.	2021-09-11 15:14:13

안정적으로 종료가 되면서 데이터에 손실이 발생하지 않음을 확인하였습니다. 그 외 똑 같은 크기의

하드 디스크를 2 개 연결하여 RAID 1 로 묶어 하드디스크가 1 개가 완전히 고장이 나더라도 똑 같은 데이터를 가지고 있는 하드 디스크가 한 개 더 있어 문제가 발생할 가능성을 줄였습니다. 이로써 Kubernetes 를 활용하여 EC2 서비스와 유사한 서비스를 제공이 가능해짐과 동시에 NAS 를 이용하여 데이터 손실이 발생하지 않은 EBS 서비스를 구축하였습니다. 그리고 EBS 서비스는 실제로는 NAS 시스템에서 동작 하므로 학교 내부 망 안이라면 실제 하드 디스크처럼 사용이 가능합니다.

### 네트워크 드라이브의 성능이 정상적으로 나오는가?

원래 학과에서 사용하던 스위칭 허브의 제품은 ipTIME 사의 SW1600plus 제품입니다. 해당 제품의



경우 스위치 허브의 성능상 인바운딩, 아웃바운딩 성능이 100mbps 속도로 나오기 때문에 네트워크 드라이브의 성능이 아무리 뛰어나도, 최대 속도는 100mbps(약 11mb/s) 입니다. 100mbps 정도의 속도는 USB 2.0 의 속도는 480mbps 이므로 약 4 배 이상 느립니다. 그래서 필수적으로 기가 인터넷이 필수적으로 필요해지면서 기가 스위치를 따로 구매하여 기가 인터넷을 연결하였습니다. 100mbps 스위치 허브와

1000mbps 공유기가 있고, 각각의 워크 스테이션 모두 랜 포트가 2 개 이상이 있었기 때문에 외부망은

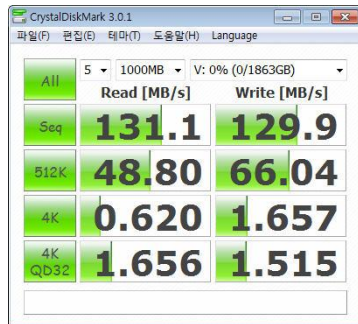
100mbps 로 외부망을 연결하고, 내부망으로 기가망으로 연결하였습니다. 기가망으로 인터넷을 설치한 뒤 내부적으로 파일을 옮기는 속도를 측정하였을 때 FTP 의 경우

4 파일, 총 크기: 4,963,597,521 바이트				1 파일 및 3 디렉터리, 총 크기: 19 바이트	
서버/로컬 파일	방향	리포트 파일	크기	우선...	상태
sftp://sftpuser@...					
C:\Users\Wdod...	-->	/data/2.mp4	3,705,...	일반	전송 중
	00:00:30 경과	00:00:05 남음	90.4%		3,353,018,368 바이트 (105.7 MiB/s)
4 파일, 총 크기: 4,963,597,521 바이트				5 파일 및 1 디렉터리, 총 크기: 3,270,44	
서버/로컬 파일	방향	리포트 파일	크기	우선...	상태
aoikazto@192.1...					
C:\Users\Wdod...	-->	/docker/2.mp4	3,705,...	일반	전송 중
	00:00:05 경과	00:00:28 남음	16.6%		617,316,352 바이트 (113.1 MiB/s)

113mb/s 가 나왔으며, sFTP 의 경우는 105.7mb/s 가 나왔습니다. 해당 속도를 mbps 로 변환을 할 경우 각각 904.8mbps, 845.6mbps 입니다. 기가 인터넷 망에서 최대 속도가 904.8mbps 나오는 것은 현재 네트워크 인터넷 속도를 활용하여 데이터 전송하는 것을 볼 수 있습니다.

### 실제 HDD 의 성능은 어느 정도 나오는가?

성능은 서버용 HDD 의 경우 약 읽고 쓰기 속도가 150mb/s 이며, 1200mbps 가 나옵니다. 실제 데이터로



그렇기 때문에 네트워크의 속도가 더 빠른 10 기가 인터넷으로 바꾸더라도 NAS 에 있는 HDD 의 성능의 최대치인 150mb/s 로 나올 것으로 예상이 됩니다. 현재 시스템 내부에 있는 NAS 시스템을 이용하여 모든 컴퓨터의 데이터를 한 군데에서 저장을 하며, 성능은 HDD 의 성능은 온전히 모두 다 내줌과 동시에 안정성까지 확보가 됩니다. 그렇기 때문에 딥러닝이라는 방대한 데이터를 수집하여 학습시킬 때 활용하거나, 연구 데이터 보관, 개발 자료와 같은 중요 데이터

보관 할 때 많은 영향이 있을 거라 생각합니다.

네트워크 드라이브를 선택하면서 생긴 과정에 대해서 겪었던 문제 사항과 왜 NFS 라는 네트워크 드라이브로 데이터 공유 시스템을 구성하게 되었는지 기록하였습니다. [\[Kubernetes 에 NFS 로 데이터 공유 시스템 구성\]](#)

## 학과 서버 현 시스템 구성

학과 서버는 학생과 교수님과 연계하여 프라이빗 클라우드 서비스를 구축하고자 목표로 하였으며, 추후 학생들의 캡스톤 디자인, 포트폴리오 서버, 연구 과제와 같이 다양한 방면으로 사용하고자 했습니다.

현재 시스템 구성은 총 5 대의 컴퓨터로 구성이 되며 부가적인 무정전전원, 스위치허브로 시스템이

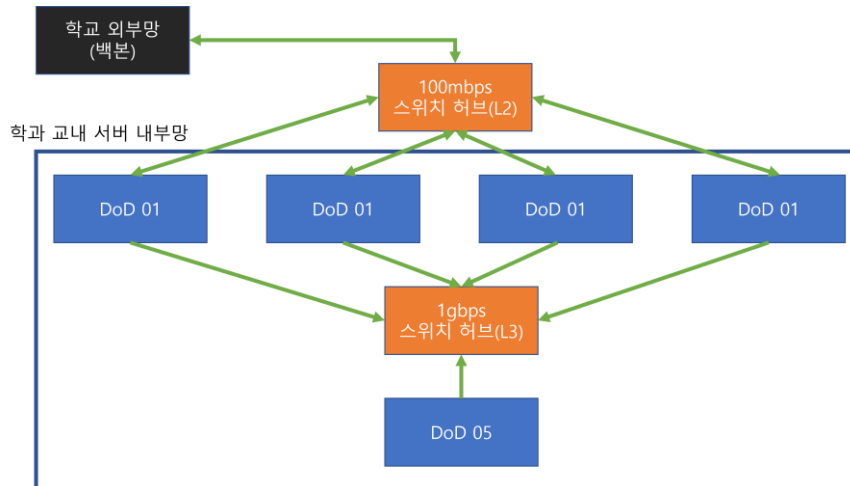
PC 명	소유자	내부망 IP 주소	외부망 IP 주소	연동	OS	주 목적
DOD 01	김남규 교수님	192.168.0.10	113.198.230.10	클러스터 A	Ubuntu 20.04	무 중단 서비스 (Kubernetes)
DOD 02	김남규 교수님	192.168.0.11	113.198.230.11		Ubuntu 20.04	
DOD 03	김현태 교수님	192.168.0.12	113.198.230.12		Ubuntu 20.04	
DOD 04	김현태 교수님	192.168.0.13	113.198.230.13	클러스터 B	Windows 10	도커 컨테이너 (Docker)
DOD 05	차주형	192.168.0.40	113.198.229.155	네트워크 드라이브	DSM 7.0	클러스터 A, B의 데이터 센터

이뤄져 있습니다. 현재 DOD 01 과 DOD 02, DOD 03 서버는 Kubernetes 를 활용하여 컴퓨터 하나가 장애가 발생하거나 문제가 생기더라도 소프트웨어나, 시스템이 죽지 않도록 하는 “무 중단 서비스”를 제공하기 위함으로 하나의 프로그램으로 묶었습니다. 대표적으로 무 중단 서비스로 활용하기 좋은 대표적인 예로는 연구 과제 서버, 학과 내 온라인 저지 시스템과 같은 시스템이 중단되면 문제가 발생하는 서비스를 적용할 때 사용합니다. DOD 04 의 경우 로컬 저장 공간을 사용해야 하거나, 가상머신이나 Windows 10 만의 기능인, RDP, 윈도우 폼 인 기능을 사용하기 위해서 Docker 와 VM



Ware 로 따로 제공하기 위해서 DOD 04 의 컴퓨터를 활용하였습니다. 그리고 모든 데이터를 저장 및 처리를 하는 DOD 05 를 통하여 DOD 01, 02, 03, 04 컴퓨터가 갑작스럽게 전원이 나가거나, 문제가 발생하더라도 데이터를 다운로드 하거나 데이터를 업로드 할 수 있는 시스템으로 구성하였습니다. 자세한 시스템 구성은 아래의 이미지와 같이 구성이 되어있습니다. 외부와의 통신은 L2 스위치 허브를 통하여 통신하므로 학교 외부에서는 100mbps 속도를 가지지만 내부망에서 파일 주고받는 속도는

1기가 인터넷을  
사용합니다.



실제 서버 내부망, 외부망을 구축하면서 내용 정리하였습니다. [\[정리, 내 외부망 통신 서버 구축\]](#)

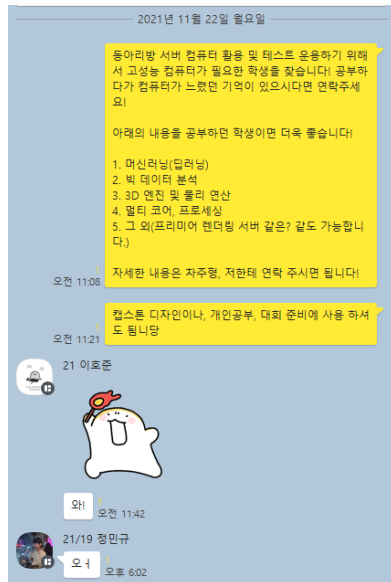
## 각 클러스터 별 시스템 구성

클러스터 A와 B는 물리적으로 연결이 된 구성이 아닌, 논리적으로 연결이 된 구성입니다. 클러스터 A의 경우 성능 보다 서버의 이중화, 고 가역성을 위하여 여러 대의 컴퓨터가 구성이 되어 있습니다. 그래서 클러스터 A는 최소 1대 이상의 컴퓨터가 있어야 시스템이 정상적으로 동작 합니다. 프로그램은 동작하고 있는 컴퓨터의 기준으로 동작 하므로 더 좋은 컴퓨터에서 동작 한다면, 성능이 달라질 가능성이 있습니다. 하지만 클러스터 B의 경우 가상 머신으로 동작하고, 이중화가 가능한 구조로 설계가 되어 있지 않습니다. 그래서 장비를 추가하는 것은 쉽지만, 장비를 제거하는 방식은 많이 어려운 구조를 가지고 있습니다. 장비를 제거해야 한다면 USB 로 가상 하드디스크 파일로 저장을 하고, 시스템을 옮겨야하는 문제가 있습니다.

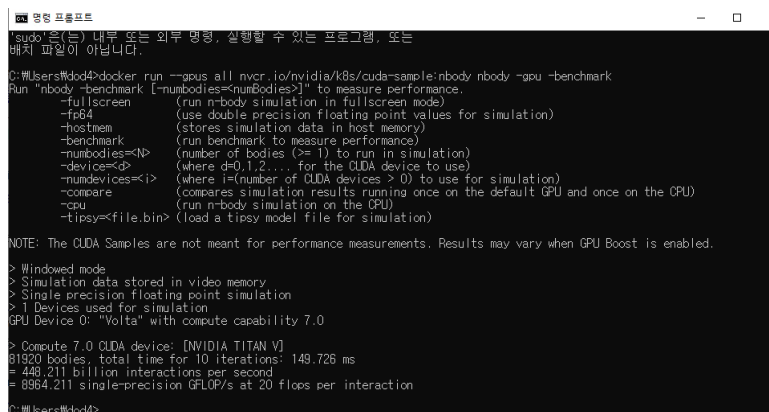


## 성능 테스트 및 시스템 운용

실제로 구성한 시스템이 정상적으로 동작하는지 확인하기 위하여 2021년 11월 22일에 딥러닝,

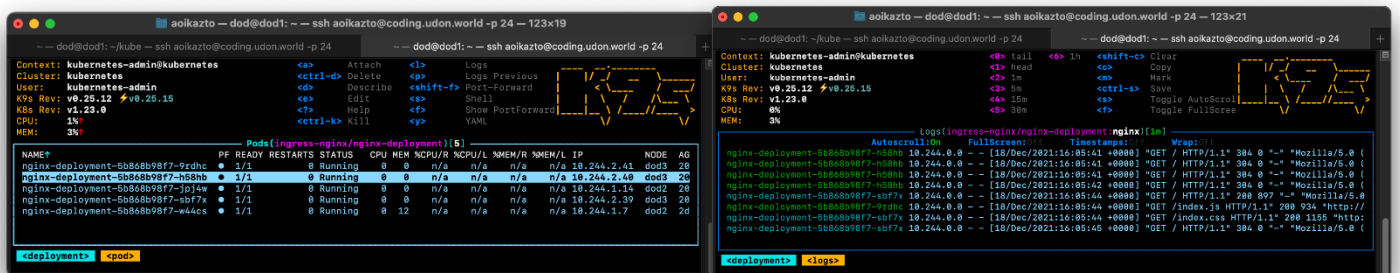


빅데이터 분석, 3D 엔진 및 물리 연산, 멀티코어 프로세싱, 렌더링 서버, 빌드 서버와 같이 다양한 방면으로 필요한 학생을 모집을 진행했습니다. 이때 성능, 부하, 스트레스 테스트하기 위해서, 고가역성(High Availability) 구성이 정상적으로 동작하여 하나의 시스템이 종료가 되거나, 순차적으로 컴퓨터를 종료를 하여도 정상적으로 동작 하는지 확인하였습니다. 이때 스트레스 테스트는 딥러닝과 빌드 서버를 테스트 진행하였고, 고 가역성 테스트는 간단한 Nginx 서버로 테스트 진행하였습니다. 그 덕분에 부족하였던 부분을 보완함과 동시에 다음 시스템을 어떻게 구성해야 하는지 방향성을 잡았습니다. 내부적으로 테스트 한 결과로는 딥러닝에서 GPU 를 사용할 때 1 개의 GPU 당 1 개의 학습만 가능하다는 것 이였습니다.



그 외에 빌드 서버는 따로 GPU 를 사용하는 것이 아닌, CPU 와 저장 장치만 활용하므로 빌드 서버는 큰 용량의 저장 공간을 할당해야 한다는 것입니다. 왼쪽에 있는 이미지는 docker 라는 가상화 프로그램에서 GPU 가 정상적으로 인식하는지 알 수 있습니다. 그래서

GPU 를 사용하는 딥러닝 프로그램을 돌리더라도 정상적으로 동작하는 하는 것을 테스트해 볼 수 있었습니다.



1 개의 서비스가 여러 개의 컴퓨터에서 동작하는 NginX 가 동작함과 동시에 매번 접근할 때 마다 다양한 컴퓨터에서 접근하고 있는 것을 확인 가능합니다. 이로써 클러스터 A 는 무 중단 서비스를 정상적으로 제공을 하고 있으며, 컴퓨터에 문제가 생기더라도 정상적으로 동작 하는 것을 확인이 가능합니다.

## 학과 서버 현 운용 현황

현재 학과 서버를 이용 중인 학생은 총 16 명이며, 다양한 목적으로 이용이 되고 있습니다. 학생들의 연구 과제나, 캡스톤 디자인 때 필요한 서버를 사용하거나, 딥러닝 모델을 학습시켜, TTS 서비스를 만드거나, JAVA 빌드 서버를 통하여 Continuous Integration(이하 CI) 쪽으로 공부하는 학생, 학우도 있습니다. 그래서 시스템을 사용하고자 하는 학생, 학우에 맞춰서 시스템이 중단이 되지 않아야 하는 서버 쪽을 이용하는 경우 클러스터 A 에 할당하였으며, 우분투나 리눅스처럼 가상머신이 필요한 경우 클러스터 B 에 할당하여 가상머신을 제공해주고 있습니다. 이때 클러스터 B 의 경우는 현재 작성하고 있는 기준 컴퓨터가 1 대이지만 과거 12 월에는 2 대를 운용 한 내부 테스트를 했습니다. 그렇기 때문에 언제든지 VM 수요가 많으면 언제든지 확장 가능하게 시스템을 구성하였습니다.

## 서버 이용 학생 목록

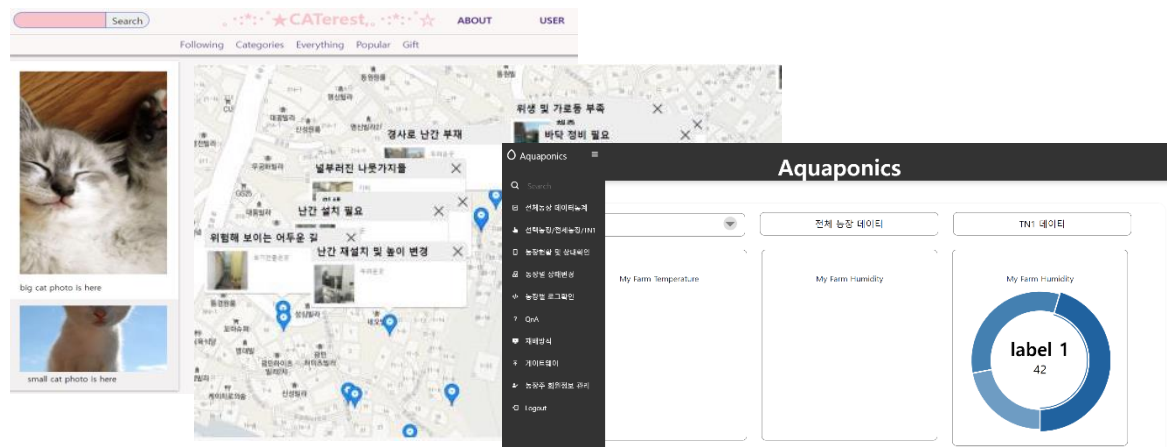
현재 서버를 이용하고 있는 학생은 아래와 같습니다.

번호	사용자	주 목적	할당 위치	할당 프로그램	비고
1	차주형	관리 및 시스템 운용, 네트워크 공부	모든 컴퓨터	Kubernetes, Docker, Podman, WSL	동아리방 412호 서버
2	장주영	딥러닝 공부를 위한	클러스터 A	Jupyter Notebook GPU	<a href="https://www.kubeflow.org/">https://www.kubeflow.org/</a>
3	정민규	개인 공부하기 위한 IDE	클러스터 B	Code-Server	<a href="http://113.198.230.13:443">http://113.198.230.13:443</a> , <a href="https://github.com/coder/code-server">https://github.com/coder/code-server</a>
4		웹 서비스 운용	클러스터 B	Apache, Php, Mysql, PhpMyAdmin	<a href="https://113.198.230.13:80">https://113.198.230.13:80</a>
5	차주형, 이현섭 교수님	안드로이드 어플 연계를 위한 서버	클러스터 A	Apache, Php, Mysql	<a href="http://113.198.230.12:30001/rest/load.php">http://113.198.230.12:30001/rest/load.php</a>
6		부산 가야의 위험 지역 안내 서비스	클러스터 A		<a href="http://113.198.230.10:30001/manager/index.php">http://113.198.230.10:30001/manager/index.php</a>
7	김지우, 이지훈, 양창윤	아쿠아포닉스 연구과제	클러스터 B	Apache, Php, Mysql, PhpMyAdmin	<a href="http://113.198.230.13:10001/aquaphonics/login.html">http://113.198.230.13:10001/aquaphonics/login.html</a>
8	임미선	웹 포트폴리오	클러스터 A	Apache, Php, Ftp	<a href="http://113.198.230.11:30003/">http://113.198.230.11:30003/</a>
9	DoD	학과, 학생용 클라우드 서비스 제공	클러스터 A	Next Cloud	<a href="https://cloud.udon.party/">https://cloud.udon.party/</a>
10	이혜규, 황진주, 차주형	연구과제, 외주 내부 테스트 서버	클러스터 B	Ubuntu 20.04, sshd	113.198.230.13:9001
11	이승훈	개인 Git 서버	클러스터 A	Gitlab	<a href="https://about.gitlab.com/">https://about.gitlab.com/</a>
12	이승훈	안드로이드 빌드 서버	클러스터 A	Jenkins	<a href="https://www.jenkins.io/">https://www.jenkins.io/</a>
13	김시찬	웹 포트폴리오	클러스터 A	Apache, Php	<a href="http://113.198.230.12:30005/">http://113.198.230.12:30005/</a>
14	황진주, 정현수, 김성안, 조은희	캡스톤 1, 호스팅 서버	클러스터 B	Ubuntu 20.04, sshd	<a href="http://113.198.230.13:8000">http://113.198.230.13:8000</a>
15	이호준	리눅스 공부	클러스터 B	Ubuntu 20.04, sshd	113.198.230.13:10101
16	이혜규	딥러닝 모델 학습	클러스터 A	Jupyter Notebook GPU	<a href="https://www.kubeflow.org/">https://www.kubeflow.org/</a>
17	최범진	웹페이지 공부 및 포폴	클러스터 A	Apache, Php, Mysql	<a href="http://113.198.230.12:30007/">http://113.198.230.12:30007/</a>

대부분 학생이 사용하는 서버는 대학교 2 학년때 배우는 내용인 Apache-Php-Mariadb 가 현재 가장 많으며, 리눅스를 공부하거나, 개인적으로 서버를 구축하기 위해서 우분투를 사용하는 친구순으로 많습니다. 비고 사안에서 현재 적용한 프로그램이나, 외부에서 바로 접근이 가능한 경로를 기입했습니다.

## 사용 중인 서비스 및 모습

제일 많은 사용률을 가지는 것은 MariaDB 와 Apache-Php 가 제일 많은 수요를 가지고 있습니다. 대표적으로 김시찬, 최범진, 임미선 학우가 포트폴리오로 쓰기 위해서 웹을 이용하여 사용하고 있으며, 김지우, 이지훈, 양창윤 학우는 이현섭 교수님의 연구과제를 진행하면서 웹 서버가 필요하여 할당하였습니다.



그 외에 클러스터 A 에는 장주영, 이해규 학생의 딥러닝 공부, 모델 학습을 위하여 Jupyter Notebook 의 GPU 를 할당했습니다. 클러스터 B 에는 Ubuntu, Centos, Windows 와 같이 다양한 운영체제를 가상적으로 제공해 주었기 때문에, 리눅스 공부, 대회 참가, 캡스톤 디자인에 활용 한 학우가 많았습니다.

## 서비스 가동 결과

12 월부터 1 월까지 약 2 개월간 시스템을 가동하여 얻은 결과로는 상당히 좋은 결과를 얻을 수 있었습니다. 학생 별로 시스템 활용하였던 분야나, 수요에 대해서 정리, 요약하였습니다.

정민규 학우는 현재 군대에 있지만 사지방에서 Visual Studio Code 를 이용하여 Java 나, 웹 개발을 통하여 개발 공부하고 있습니다. 사지방에서 서버를 통하여 야학 운영, 코딩 테스트 문제 풀고 있습니다.

이승훈 학우는 현재 Java 와 CI, 서버 쪽에 관심이 많아 Jenkins 와 Gitlab 에 대해서 공부하고 있습니다. 개인 컴퓨터에서 설치하여 사용하는 것은 리소스가 많이 필요하고, 고정 IP 가 필요하여 힘든 점이 많다고 하였습니다. 그래서 학교 서버는 고정 IP 와, 고성능 컴퓨터가 있어 편리하다고 하였습니다.

이해규 학우는 현재 키보드에 관심이 많아, TTS 나, 한글을 한자로 변환해주는 안드로이드 어플을 개발하고 있습니다. [\[안드로이드 어플\]](#)에서 기존 한글을 한자로 변환할 때 안드로이드에서 Java 로 성능이 부족하여 JNI 를 통한 C++로 트리 탐색이나 최적화했습니다. 그러나 이번 시스템을 활용하여 딥러닝 모델을 학습 통한 다양한 방식으로 적용하여 어플을 출시하고 있습니다.

## 서버 컴퓨터의 수요

외부에서 접근이 가능한 컴퓨터의 수요는 포트폴리오를 위하여 웹 서버와, 캡스톤 디자인, 대회 프로젝트, 연구 과제를 진행할 때 많은 수요가 있었습니다. 그리고 최근, 2월에 진행하는 인공지능 교육으로 인해 Tensorflow에 대한 관심이 높아지면서 공부하거나 딥러닝 모델을 학습하기 위해 일부 학우가 있었습니다.

만약 추후 시스템이 고도화가 되고 안정화 단계에 들어선다면 다양하고, 수업 때 서버를 제공해 주는 방식도 가능할 것이라 생각합니다. 예를 들어 2학년때 웹 프로그래밍 실습, 3학년 때 영상처리, 3,4학년 때 캡스톤 디자인에 적용할 수 있을 것이라 생각합니다. 현재는 홍보를 따로 하지 않아 정확한 수요 예측은 힘들지만, 향후 홍보를 하여 캡스톤 디자인, 포트폴리오, 대회 참가, 교수님께서 수업에 활용을 하신다면 1년에 100~120개 정도 서비스가 만들어질 것으로 예상합니다.

## 클라우드 서비스 구축

현재는 서버가 필요하다면 차주형 학우가 매번 명령어를 입력하여 서버를 가동해야 합니다. 일부 자동화를 하였지만 Apache, Php, Mariadb, Ubuntu 정도입니다. 그래서 서버를 사용 중인 학생이 지금 현재는 16명이며 수동으로 생성이 가능했지만, 추후 교수님이 사용하실 서버가 추가가 되거나, 많은 학생이 사용하게 될 경우 서버를 구축하는 작업이 자동화가 되어야 합니다. 그래서 동아리방 서버 컴퓨터 근처에 라즈베리 파이와 터치 디스플레이를 활용하여 라즈베리 파이에서 터치를 통해 자동화하고자 하였습니다. 최초 개발 목표는 클라우드 서비스가 목적이 아닌, 사물함 자물쇠와 연동하여 동아리방에 있는 책, 자재 관리하는 프로그램으로 기획이 되었습니다. 하지만 책이나 라즈베리 파이와 같은 장비는 학생, 학우가 자유롭게 사용하도록 하는 것이 더 좋은 환경을 만들 수 있을 거라 생각하여 계획을 변경하였습니다. 그래서 라즈베리 파이를 통하여 클라우드 서비스를 제공하고자



계획을 변경하였습니다. 현재는 또 다시 계획을 변경하여 라즈베리 파이뿐 아니라 스마트폰 모바일 기기나 웹을 통하여 로그인 하여 서버를 만들 수 있도록 할 예정입니다. 마치 AWS처럼 서비스를 제공하고자 합니다. 실제 클라우드 서비스가 된다면 왼쪽에 있는 이미지처럼

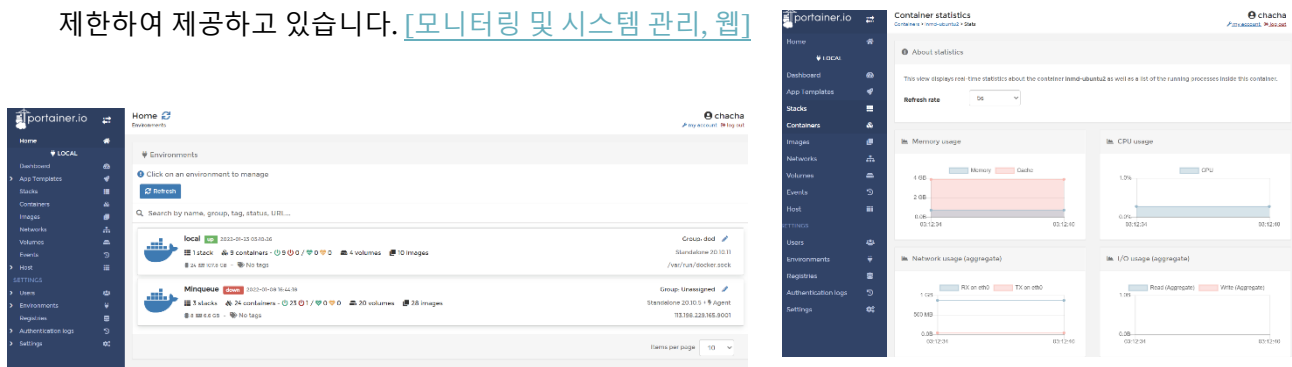
동작할 예정입니다. 동아리방 서버 컴퓨터의 리소스 할당이나, 자재 관리를 어떻게 하는게 효율적인지, 내부적인 통신은 어떻게 할 것인지 정리하였습니다. [\[클라우드 구상 전, ERP 시스템 구성 아이디어\]](#)

## 현재 클라우드 서비스 구축 상태

클러스터 A와 B는 모두 각기 다른 방식으로 시스템이 구성이 되어 있습니다. 클러스터 A, B 모두 스마트폰에서 구동하기 위해서 UI 프로토 타입과, 통신 프로토콜을 모두 정의하였습니다. 그리고 현재 어떤 언어로 개발을 할지 어떤 프레임워크를 사용할지 요구 분석 중에 있습니다. 개발 일정은 2월 28일까지 프로토 타입 완성으로 목표로 하고 있습니다. 프로토 타입은 클러스터 A만 스마트폰으로 회원가입, 로그인을 진행한 뒤 Apache, Php, MariaDB, FTP, Storage 서버를 터치 만으로 구축할 수 있도록 할 예정입니다. 현재는 아래의 사진처럼 템플릿을 만들어 놓은 뒤, 수동으로 명령어를 통해 할당하는 방식으로 사용하고 있습니다. 이렇게 클러스터 A의 목표는 클라우드 서비스를 어떻게 구축하고 구현을 할 것인지 요구 분석 및 설계하고 있는 단계입니다.

```
dod@dod1:~/kube$ ls
beomjin-apache-5g.yaml  ingress.yaml  prometheus  template
beomjin-mariadb.yaml  meason-apache-1g.yaml  prometheus-kubernetes.yaml  test-pod.yaml
example-xxxxx.yaml    nfs          sichan-apache-1g.yaml      user
dod@dod1:~/kube$
```

클러스터 B의 경우 가상머신을 대여해주는 기능은 반 자동화와, 모니터링 시스템 구축을 모두 성공적으로 적용을 한 상태입니다. 가상머신을 대여해주는 시스템은 portainer.io란 기업에서 무료로 제공해주는 시스템을 활용하였습니다. Portainer는 도커라는 가상화 프로그램을 여러 대의 컴퓨터를 효율적으로 관리해주는 프로그램과 동시에 시스템이 정상적으로 동작하는지 알려 주면서 문제가 발생하게 되면 자동적으로 사용자에게 알림 메시지를 보내거나, 로그를 남깁니다. 그리고 원격으로 클러스터 B에 접근 권한 하거나, 새로운 가상머신을 돌리는 것이 가능합니다. 현재는 클러스터 B에 가상머신을 할당하여 제공하고, 시스템 리소스가 얼마큼 사용하고 있는지 추적할 수 있습니다. 하지만 무슨 작업을 하고 있는지 알 수 가 없는 단점이 있어 현재는 CPU, 메모리, GPU, 스토리지 성능을 제한하여 제공하고 있습니다. [\[모니터링 및 시스템 관리, 웹\]](#)



N 개의 컴퓨터 관리 시스템과 각 컴퓨터 별 모니터링 시스템입니다.

이렇게 클러스터 B의 경우 가상 컴퓨터를 언제든지 생성이 가능하게 반 자동화 작업을 하였습니다.

## 정리

과거 클라우드 서비스가 나타나기 이전에는 IT 프로젝트를 진행한다면 우선 컴퓨터를 구매한 뒤 컴퓨터에 운영체제 설치, 개발용 프로그램 설치하는 순서로 진행이 되었습니다. 그래서 컴퓨터를 구매할 때 요구 사양을 한번 정하면 변경이 어려움이 많습니다. 하지만 Amazon Web Service 가 2006 년에 나오면서 클라우드 서비스가 나오기 시작했습니다. 클라우드 서비스가 나오면서 벤처기업, 핀테크 기업, 스타트업 기업, 게임 등 다양한 분야에서 사용하고 있습니다. 그 덕분에 클라우드 서비스가 시장성이 커지면서 클라우드란 개념이 중요하게 되었습니다. 그 덕분에 수원대학교에서 클라우드 학과를 신설하거나, 많은 학교에서 클라우드를 다루는 대학원이 생기기도 하였습니다.

솔직하게 말씀을 드리자면 제가 관심이 있는 분야가 AWS 나, Azure 와 같이 클라우드 서비스를 제공해주는 분야 이기 때문에 한번쯤 직접 구축해보고 싶었습니다. 하지만 실제로 구축하는 과정을 공부하면서 여러가지 살이 게 되면서 저희 학과만 할 수 있는, 시스템을 구축하고 학생에게 제공을 하는 것을 목표로 잡게 되었습니다. 그리고 2022 년 동아리 회장 직책을 맡게 되면서 응용 소프트웨어과를 더 발전하고 더 좋은 환경을 만들고 싶습니다.

현재 제가 생각하고 있는 최종적인 목표로는 학과 내에 학과 학생이 자유롭게 사용이 가능한 클라우드 서비스를 제공함과 동시에 2, 3 학년 때 웹프로그래밍과, 캡스톤 디자인에서 필요한 서버를 사용하는 것입니다. 진행하는 팀 프로젝트를 클라우드 서비스를 통해 구축하고 과제 제출할 때 소스 코드를 제출하는 것이 아닌, 클라우드의 계정만 제출하는 방향이 더 좋은 학과 경험을 주지 않을까 생각합니다. 그리고 학과 서버로 동작하므로, 학과 홍보할 때 시연용으로 사용이 가능할 것이며, 학생 포트폴리오로도 좋을 것이라 생각합니다. 그 외에 교수님께서도 학과 클라우드 서비스를 이용해 딥러닝 프로젝트를 하시거나, 연구를 진행하실 때 많은 도움이 될 것이라 예상합니다.

긴 글을 읽어 주셔서 감사합니다.

---

출처:

차주형 학생의 개인 블로그 : <https://blog.udon.party/>

황진주 학생의 개인 블로그 : <https://mfdo.tistory.com>

프라이빗 클라우드 개념:

<https://www.vmware.com/kr/topics/glossary/content/private-cloud.html>

AWS 의 Public DNS 와 SSH 접속 방법:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html>

AWS 에서 Public DNS 에서 IP 관리 및 동작 원리:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>

수원대학교 클라우드 학과:

<https://www.suwon.ac.kr/index.html?menuno=3023>

실제 클러스터 B 의 Portainer 의 환경:

<http://113.198.230.13:9000/#!/home>

---



---

실제 적용하기 위해서 공부하였 던 책

1. 쿠버네티스 완벽 가이드
  2. 이것이 우분투 리눅스다
  3. AWS Lambda 로 시작하는 서버리스
  4. 후니의 쉽게쓴 시스코 네트워킹
  6. Building Microservices with GO
  7. 쉽게 배우는 데이터 통신과 컴퓨터 네트워크
-