

Red Hat OpenStack 17

저자 직강 + 스터디 그룹 - 발표회

Nalee Jang, Senior Consultant of Red Hat Korea

이대하, 단국대학교 빅데이터정보원 정보인프라팀 엔지니어
문성업, 삼성전자 소프트웨어 엔지니어

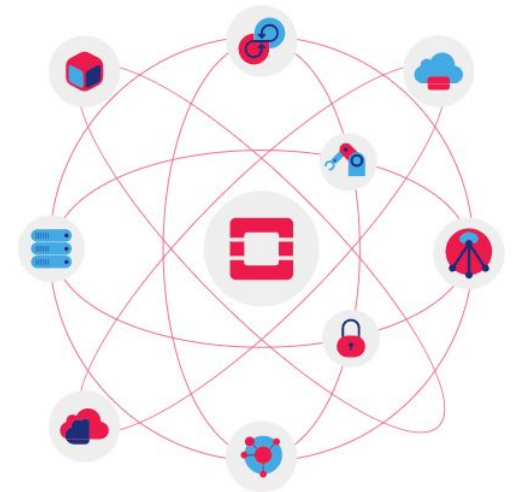
Today's Agenda

레드햇 오픈스택 17 스터디 그룹을 진행하면서
겪은 좌충우돌 삽질이야기와 스터디 그룹을 진행하게 된 동기
그리고, 스터디 그룹을 통해 우리가 배운것들에 대해 이야기하고자 합니다.

- 스터디 그룹 소개 및 동기 그리고 스토리 ----- NaleeJang
- 설치부터 운영까지 단 2일! 초심자의 트리플O 배포기 ----- 이대하
- 오픈스택 컨테이너 환경 및 스터디 참여 스토리 ----- 문성업

스터디 그룹 소개! 스터디 그룹을 하게 된 이야기

Nalee Jang



About this Study Group

- 세미나명 : “레드햇 오픈스택 17 저자직강 + 스터디 그룹”
- 신청자수: 30명
- 진행기간 : 4월 첫주 ~ 5월 중순, 매주 목요일 저녁 7시 30분 ~ 8시 30분, 단 오프라인 미팅은 변경될 수 있음.
- 진행방법: 2번의 오프라인 모임 + 4번의 온라인 모임
- 준비물 : 리눅스가 설치된 최소 사양 이상의 컴퓨터 또는 서버(CPU 8 Core, Memory 32 GB, 500 GB), 레드햇 오픈스택 17 제본 도서

The Curriculum

언제	커리큘럼	모임유형
1주차 (4월 6일)	오리엔테이션 - 진행방향 소개 - 그룹 만들기 - 세션 : 기본 구성환경 및 네트워크 설계	오프라인 미팅
2주차 (4월 13일)	Repo 서버 및 언더클라우드 구성	온라인 미팅
3주차 (4월 20일)	오버클라우드 노드 준비 및 운영체제 설치	온라인 미팅
4주차 (4월 27일)	오버클라우드 배포 및 트러블슈팅	온라인 미팅
5주차 (5월 4일)	기능 테스트 및 스터디 과제 진행	온라인 미팅
6주차 (5월 11일)	그룹별 스터디 과제 진행상황 확인 및 공유	그룹별 미팅
7주차 (5월 18일)	스터디 공유의 시간 - 스터디 개요 소개 - 스터디 그룹별 발표 - 스터디 참여 소감 공유	오프라인 미팅

Rules of Making Group

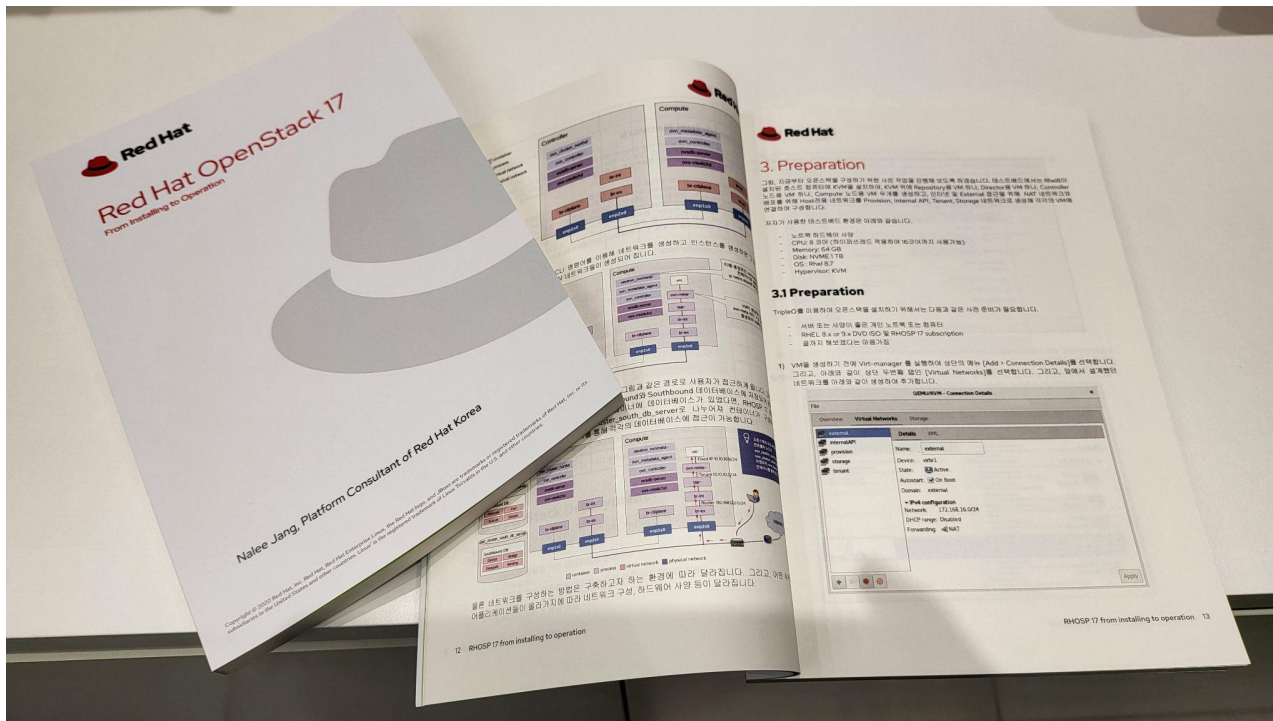
- 그룹을 5개로 만들겠습니다. 그럼, 한그룹당 5~6명정도가 되겠죠!
- 그룹을 만들고 그룹장을 선출해 주세요!!
- 아래의 5개의 그룹 미션 과제가 있습니다.
- 그룹장은 그룹원들과 함께 미션을 선택해 주세요!
- 그룹장은 선택한 그룹 미션에 대해 소개해 주세요!

- OVN(Open Virtual Network) 파헤치기
- 컴퓨터 노드에는 어떻게 네트워크가 구성될까?
- 옥타비아를 살펴보자!
- 인스턴스 생성과정과 마이그레이션 과정 알아보기
- 컨테이너 환경! 을 살펴보자.

그룹 미션 과제

* 그룹 간에 같은 미션을 선택할 수 없습니다.

스터디 그룹을 하게 동기!

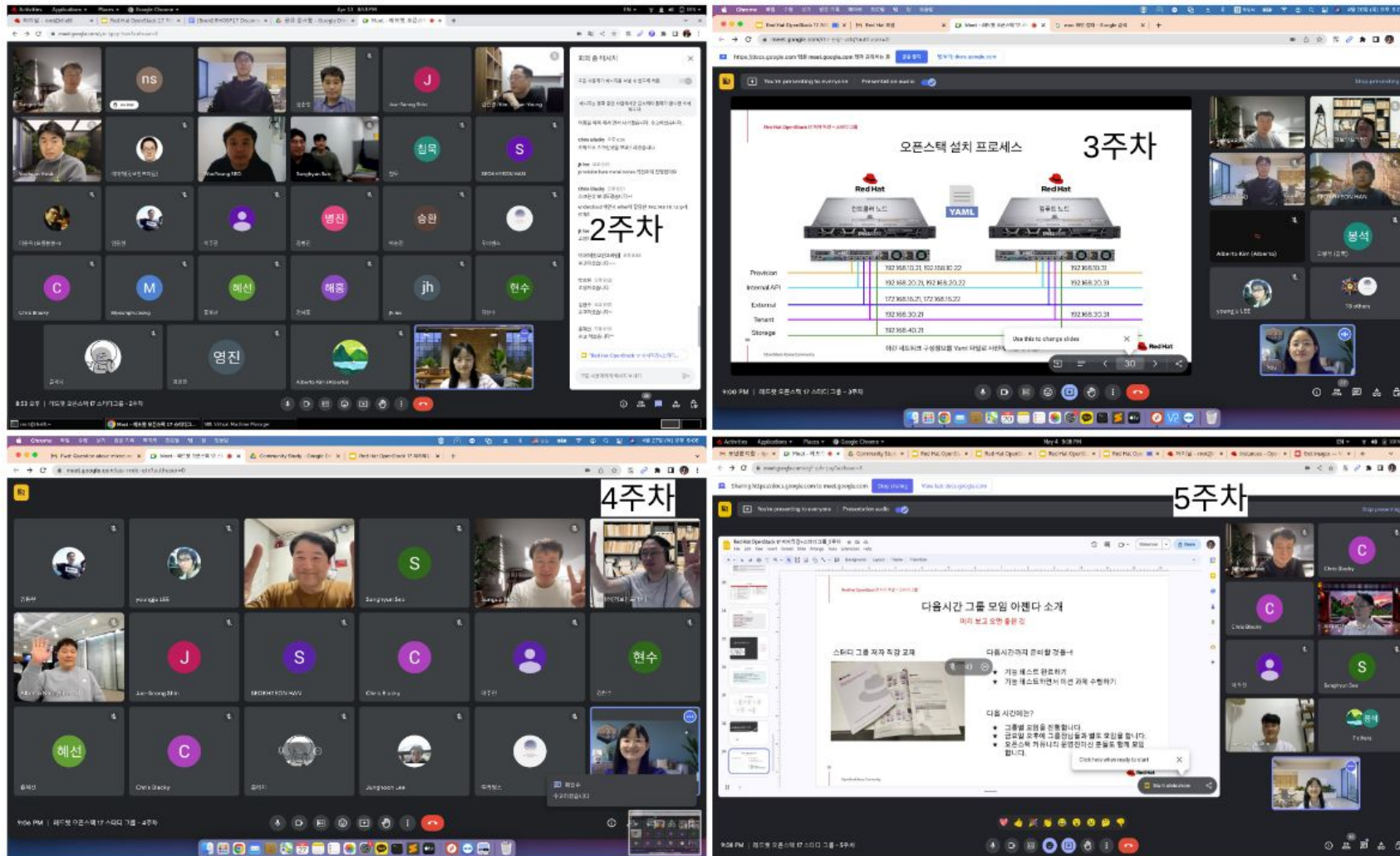


스터디 그룹을 하게 된 이유 중 하나는 바로 직접 작성하고 편집해서 만든 이 책 때문입니다.

스터디 그룹 첫 모임!



4번의 온라인 모임



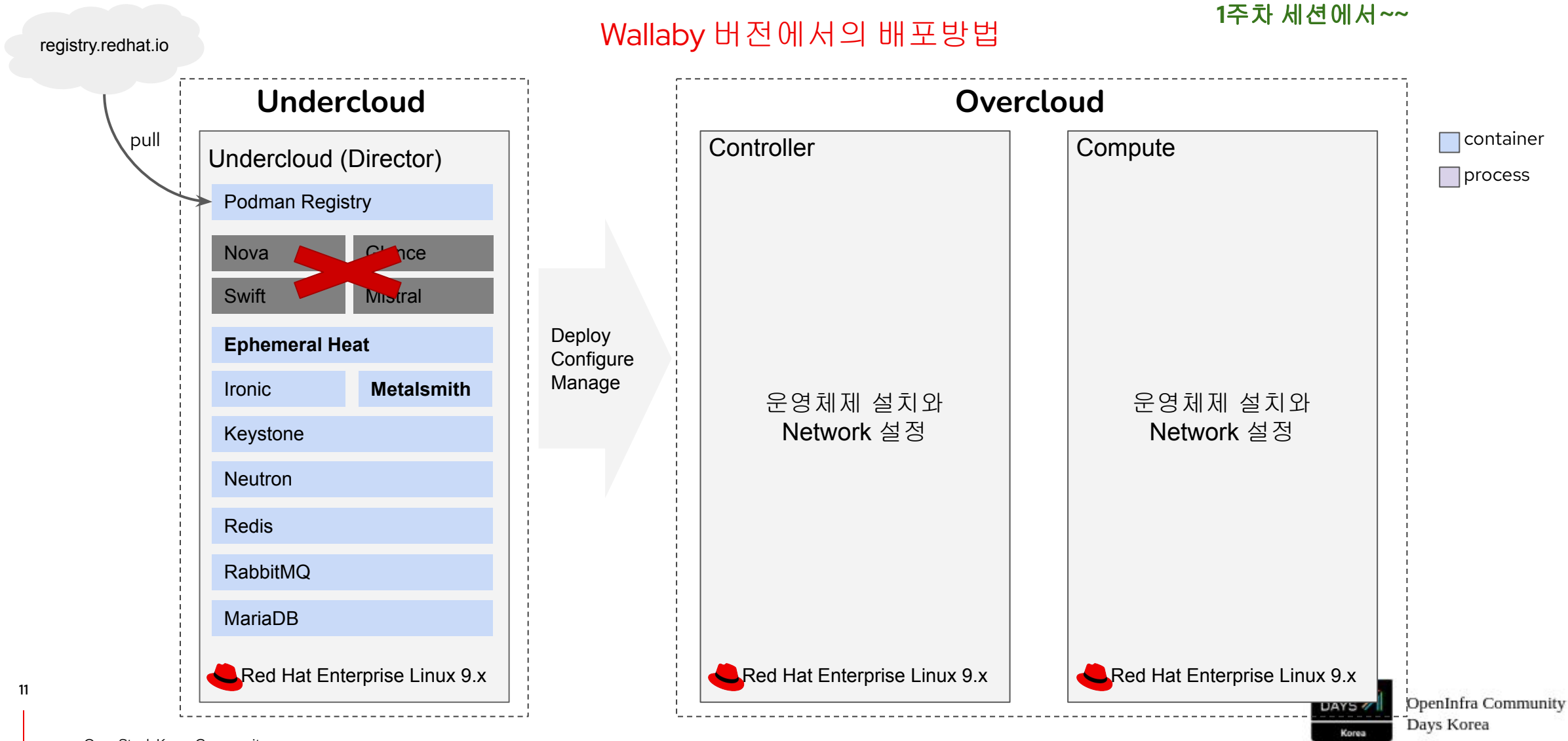
우리만의 작은 발표회~!!



TripleO를 이용한 배포방법 소개

Wallaby 버전에서의 배포방법

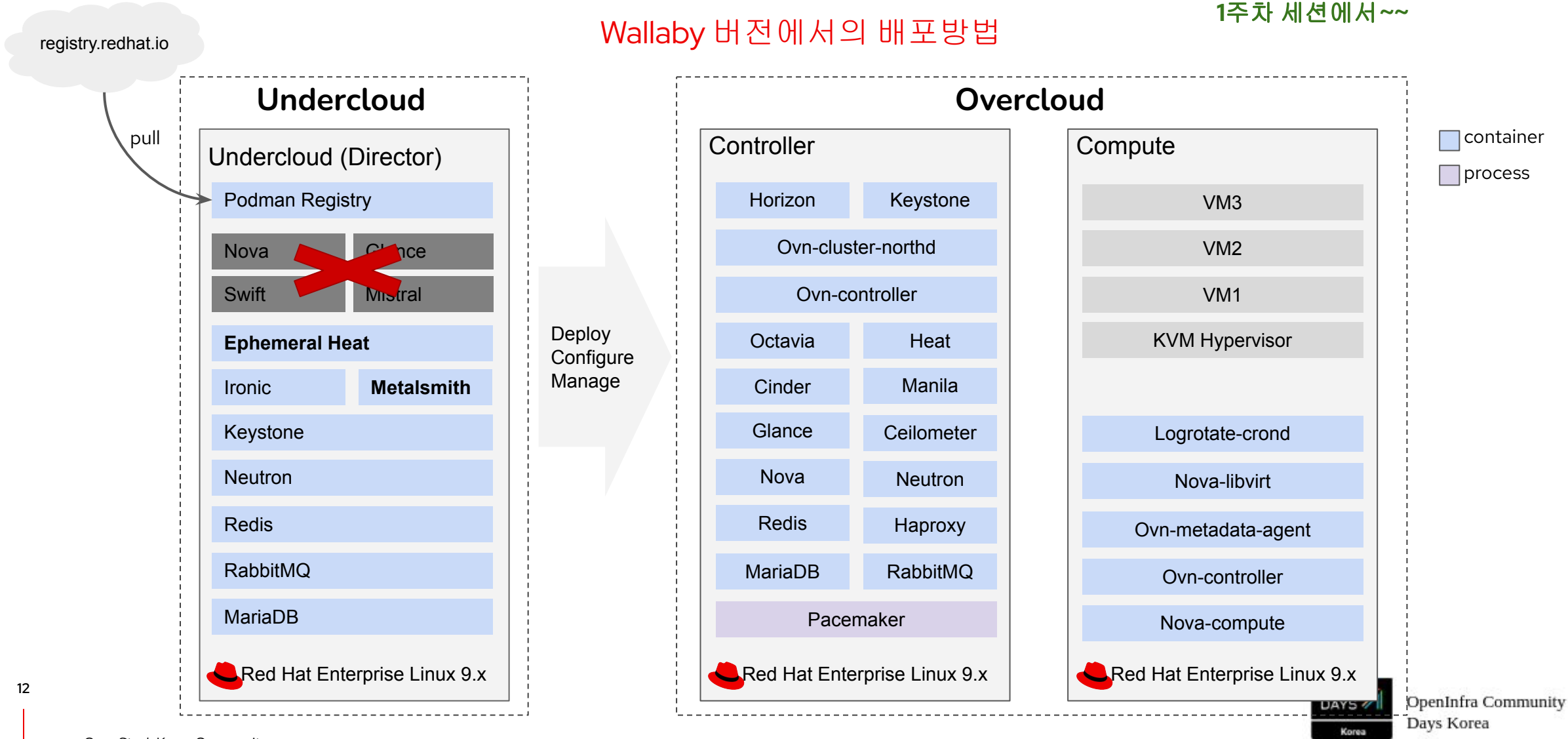
1주차 세션에서~~



TripleO를 이용한 배포방법 소개

Wallaby 버전에서의 배포방법

1주차 세션에서~~



인프라 용어 알고 가기

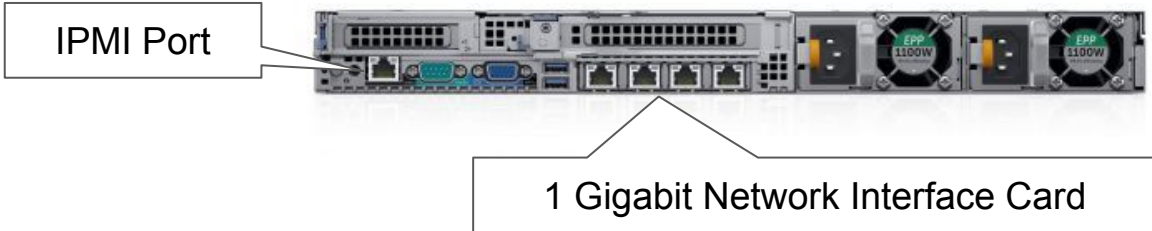
오픈스택 엔지니어는 하드웨어도 알아야 합니다.

2주차 세션에서~~

서버가 얇고 슬림해 보이죠? 이런 서버를 1U 서버라고 합니다.

IPMI

Intelligent Platform Management Interface 의 약자로 하드웨어를 원격으로 관리하는데 사용되는 인터페이스를 정의함.



1G NIC



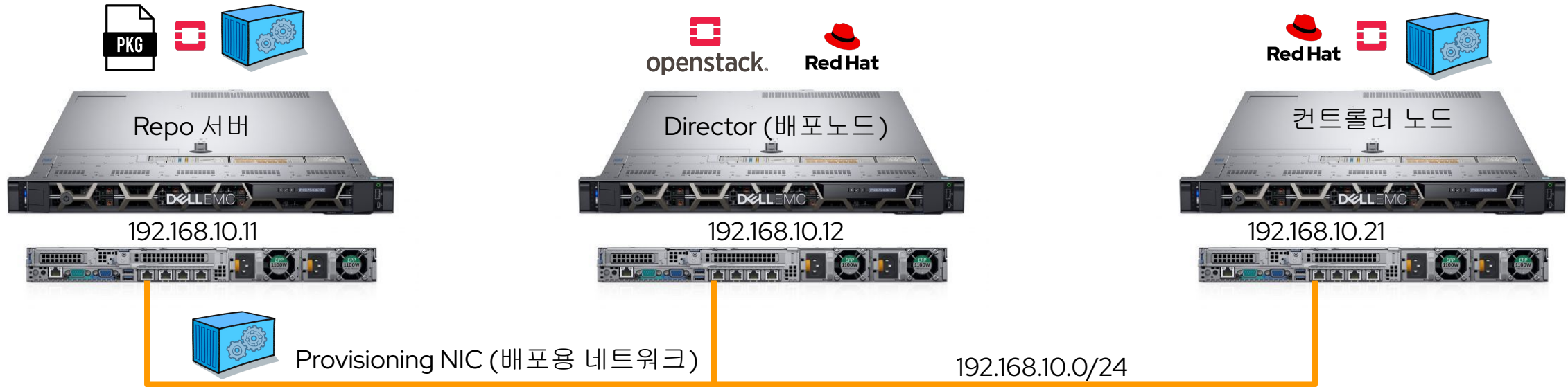
10G NIC



40G NIC

오픈스택 설치 프로세스

2,3주차 세션에서~~



오버클라우드 노드의 운영체제 설치 및 네트워크 구성이 끝나면
사전에 정의한 YAML 템플릿 파일을 이용하여 오버클라우드 노드에
Repo 서버의 컨테이너 이미지를 이용하여 오픈스택 서비스를
설치함.

스토리지 종류 알아보기

ISCSI Protocol



DELL EMC Unity



Fujitsu ETERNUS



Huawei 18000



Netapp



DELL EMC VMAX



Hitachi VSP



FC, ISCSI

HP 3Par



IBM DS8900F



Fiber Channel



스토리지 Support Matrix

4주차 세션에서~~

The screenshot shows the Red Hat Ecosystem Catalog interface. The search results are filtered for 'Storage' and 'Red Hat OpenStack Platform'. 15 results are found, sorted by relevance. The results include:

- Fujitsu:** ETERNUS OpenStack VolumeDriver 1.7 for RHOSP16.2
- Hewlett Packard Enterprise:** HPE 3PAR, Primera and Alletra 9000
- Hitachi:** Hitachi Block Storage Driver for OpenStack
- HP:** HPE StoreVirtual Storage and HPE StoreVirtual VSA
- Dell Technologies:** Dell EMC XtremIO, PowerFlex, and Dell EMC VMAX and PowerMax
- NEC:** NEC Storage Cinder

Current supported release is 2023.1.antelope. [Back to Top](#)

Summary

Feature	Status	(Ceph) iSCSI Storage Driver (iSCSI)	DataCore Storage Driver (FC, iSCSI)	Datera Storage Driver (iSCSI)	Dell PowerFlex (ScaleIO) Storage Driver (ScaleIO)	Dell PowerMax (2000, 8000) Storage Driver (iSCSI, FC)	Dell PowerStore NFS Driver (NFS)	Dell PowerStore Storage Driver (iSCSI, FC, NVMe-TCP)	Dell PowerVault ME Series (iSCSI, FC)	Dell SC Series Storage Driver (iSCSI, FC)	Dell Unity Storage Driver (FC, iSCSI)
Supported Vendor Driver	optional	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Extend an Attached Volume	optional	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓
QoS	optional	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓
Volume Replication	optional	✓	✗	✗	✓	✓	✗	✓	✗	✓	✓
Consistency Groups	optional	✗	✗	✗	✓	✓	✗	✓	✗	✓	✓
Thin Provisioning	optional	✓	✗	✗	✓	✓	✓	✓	✗	✓	✓
Volume Migration (Storage Assisted)	optional	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓
Multi-Attach Support	optional	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓
Revert to Snapshot	optional	✓	✓	✗	✓	✓	✗	✓	✗	✗	✓
Active/Active High Availability Support	optional	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗

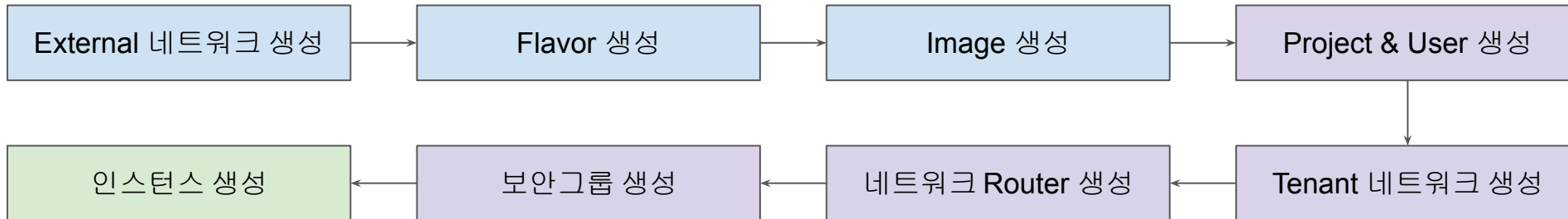
https://catalog.redhat.com/software/search?target_platforms=Red%20Hat%20OpenStack%20Platform&functionalCategories=Storage

<https://docs.openstack.org/cinder/latest/reference/support-matrix.html>

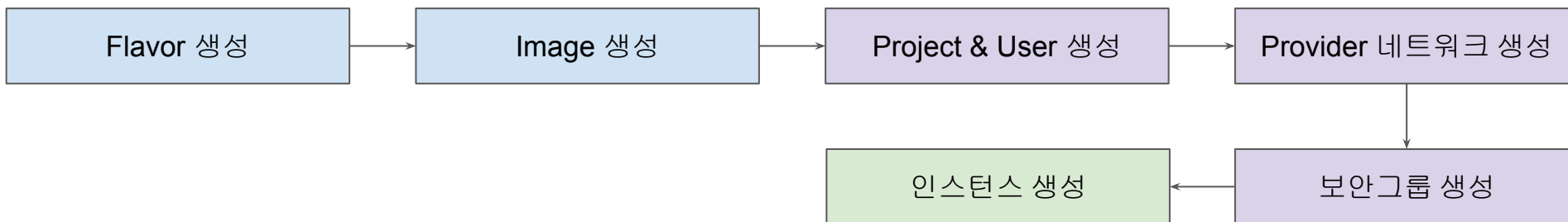
첫번째 인스턴스 생성하기

5주차 세션에서~~

테넌트 네트워크를 사용할 경우



프로바이더 네트워크를 사용할 경우



스터디 그룹 저자 직강 공유 자료

The screenshot shows a Tistory blog post with a table of weekly activities and a list of links to previous posts.

4주차 (4월 27일)	오버클라우드 배포 및 트러블슈팅	온라인 미팅
5주차 (5월 4일)	기능 테스트 및 스터디 과제 진행	온라인 미팅
6주차 (5월 11일)	그룹별 스터디 과제 진행상황 확인 및 공유	그룹별 미팅
7주차 (5월 18일)	스터디 공유의 시간 - 스터디 개요 소개 - 스터디 그룹별 발표 - 스터디 참여 소감 공유	오프라인 미팅

그리고, 5월 4일 마지막 온라인 세션을 진행을 했습니다. 차주(5월 11일)에는 그룹별 스터디 과제를 진행을 할 예정입니다. 그룹별로 모여서, 서로 연구하고 느끼고 공부한 내용들을 공유하고, 그룹별 미션을 진행하는 시간을 가질 예정입니다. 마지막 7주차에서는 스터디 공유의 시간을 가질 예정입니다. 이때는 다른 커뮤니티 회원님들도 초대하여 진행할 예정이니 많은 참여 부탁드립니다요~!!

마지막으로 그동안 진행했던 "레드햇 오픈스택 17 저자직강 + 스터디 그룹" [세션 자료](https://naleejang.tistory.com/268)를 공유합니다.

- [Red Hat OpenStack 17 저자직강+스터디그룹_1주차](#)
- [Red Hat OpenStack 17 저자직강+스터디그룹_2주차](#)
- [Red Hat OpenStack 17 저자직강+스터디그룹_3주차](#)
- [Red Hat OpenStack 17 저자직강+스터디그룹_4주차](#)
- [Red Hat OpenStack 17 저자직강+스터디그룹_5주차](#)

<https://naleejang.tistory.com/268>

Red Hat OpenStack 17 저자직강+스터디그룹...
저자 직강 + 스터디 그룹 - 5주차 Red Hat OpenStack 17 Nalee Jang, Senior Consultant of Red Hat Korea 1 The Curriculum Red Hat OpenStack 17 저자 직강 + 스터디 그룹...
www.slideshare.net

The screenshot shows a Tistory blog post with a main title and a grid of photos.

레드햇 오픈스택 17 스터디그룹을 진행하면서

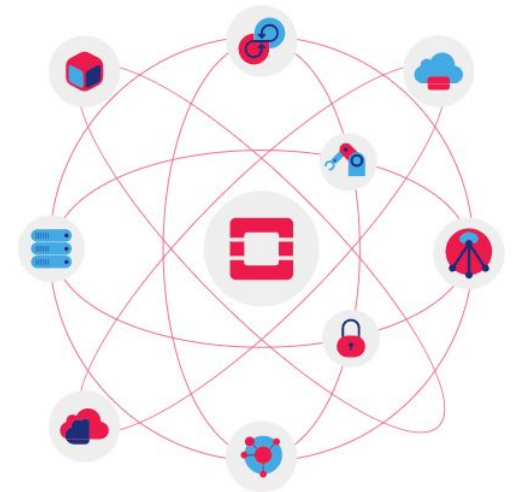
<https://naleejang.tistory.com/270>

옛그제 시작한 것 같은데, 벌써 스터디그룹이 끝이 났다. 사실 이런거 진행할 생각조차 안했었는데, 회사에서 우연히 파트너사에 오픈스택 교육을 갔다가 커뮤니티 회원들한테도 교육을 해 주면 좋을 것 같다는 생각이 문득 들었다. 그래서, 페이스북 그룹에 내 생각을 공유하니 많은 분들이 관심을 갖고, 듣고 싶어해서 신청을 받고, 어떤 내용으로 진행을 할까 고민을 하고, 커리큘럼을 만들고, 회사에 진행 여부를 컨펌 받고, 진행을 하기 시작했다.

첫 오프라인 모임은 스터디 그룹 참여를 신청해 주신 대부분의 분들이 참여를 했다. 오픈스택이 나오지 왜 오랜 시간이 지났지만, 여전히 오픈스택은 사랑 받고, 관심을 받고 있음을 난 알 수 있었다. 커리큘럼을 만들면서, 스터디 그룹이니 무언가 스스로 공부할 수 있는 기회를 가지길 바랐다. 그래서, 프로젝트를 하면서 많은 사람들로부터 많은 질문들~!! 그리고, 많은 사람들이 궁금해 하는 부분들을 스스로 살펴보고, 공부하는 시간을 가지면 좋겠다고 생각했다. 그래서, 모인 분들을 소규모로 다시 그룹핑을 해서, 그룹별 미션을 주었다.

설치부터 운영까지 단 2일! 초심자의 TripleO 배포기

이대하



학교 서버 관리자는 안하는거 빼고 다합니다.

그럼에도 불구하고 포기할 수 없는 건 학생들을 위한 클라우드 서비스 환경

<이미지 출처 : 본인촬영 "단국축제">



대안이 없다.

□ OpenStack

서버 스토리지 운영 관리 (보안, 유지보수, 구축)

- 서버 설치, 업데이트
- 인프라 SW 운영 관리
 - 파트 불량 교체
 - OS 설치, 패키지 설치
 - 매년 ISMS 심사 대비
 - 교육부 보안 권고 조치 적용
- 방화벽 정책 요청
- 업체 출입 허용 등.

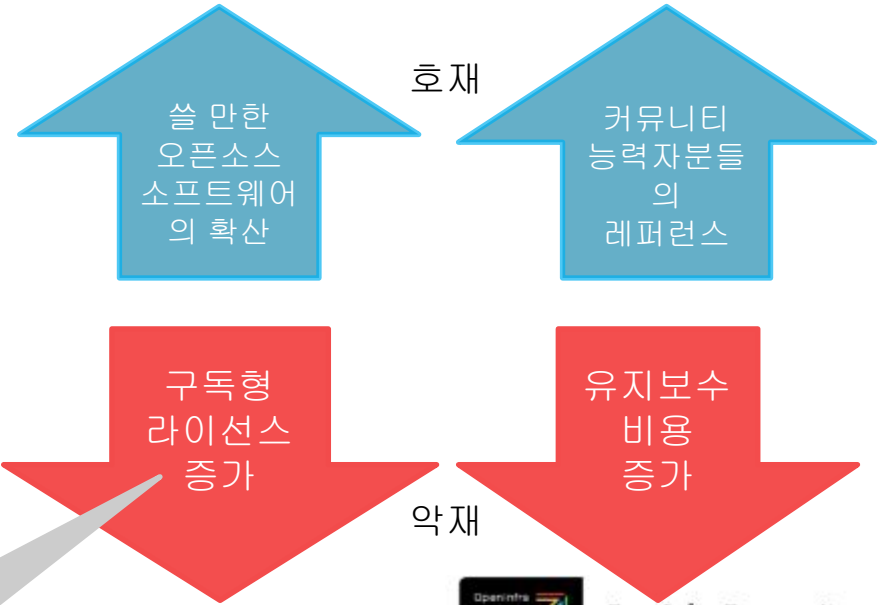
랙마운트 케이블 포설도 직접 합니다.

시스템 설계

- 학교에 도움될 SW PoC
- 학교에 도움될 HW PoC
- 각종 오픈소스 테스트
- 업데이트/업그레이드 전 사전 환경 구축 후 테스트

행정, 투자, 검수, 기타 학교 업무

- 공문작성, 결재, 공문회신
- 장비 투자에 따른 많은 문서 작성
- 업체 견적요청, 네고요청
 - 납품/구축일자 조율
 - 기타 영수증처리



아무래도 이걸 넷플릭스 탕이 아닐까 (제 개인적인 생각)



OpenInfra Community Days Korea

실패 없는 오픈 스택 배포를 위한 사전 준비

- GUI 기반 NAS , 네트워크 환경 community sw 사용하자!
pfSense (네트워크) , TrueNAS (스토리지) 를 사용했습니다.
-> KVM 자체 네트워크나, 패키지를 설치할 수도 있지만, 문제가 생겼을 때 매일 같이 다루는 SW가 아니면 config를 찾는데 시간을 소요한다.
GUI 로 쉽게 확인되는 기반 환경으로 테스트 시작한다.
- 나는 Openstack 커맨드에 익숙하지 않다!
-> 실행 커맨드 .yaml 꼼꼼히 곱씹어본다. (어떤 의도인가 파악)
-> 에러가 발생하면 Openstack command 찾느라 고생을 해야한다.
(그럼에도 불구하고 문제는 발생하였습니다.)
- 어떤 부분에서 Outbound 통신이 필요한지 모른다.
-> 모든 네트워크 인터넷 접근 가능하게 구성 (테스트 니깐)
-> 나중에 테스트 하면서 하나씩 제한한 뒤에 방화벽 로그를 떠본다.
- 네트워크 대역폭은 최대한 넓힌다. (1서버 구성은 제외)
-> 최근 클라우드 컴퓨팅 솔루션이 네트워크 비용이 점점 낮아짐에 따라 SAN방식보다 네트워크를 통한 파일 교환 방식으로 전환되고 있고, 최소한 10Gbe 권고 된다.
(1Gbe 도 테스트 시 문제는 없지만 1->N 배포방식에는 문제가 될 수 있다.)
domain 네임은 example.com 등 누구나 알기 쉬운 이름 혹은 예약어는 제외



<이미지 출처 : 영화 “인터스텔라”
중>

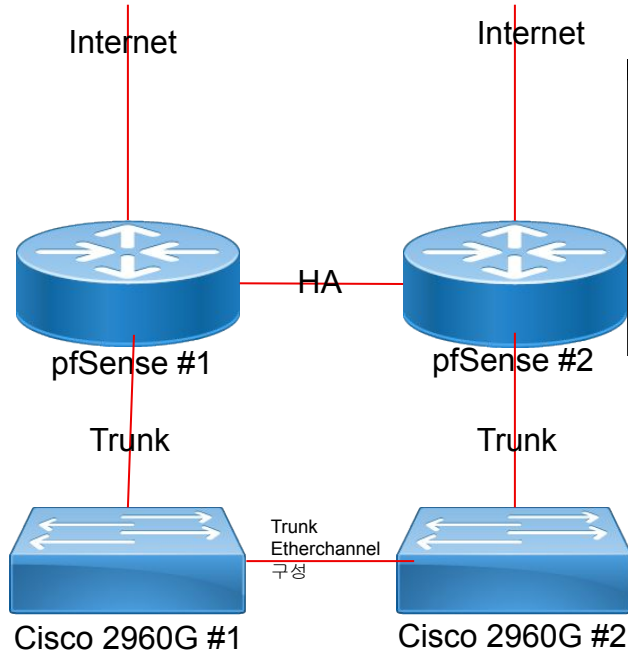
구성도

* 모든 서버는 전 포트 활용 802.3ad LACP본딩 구성

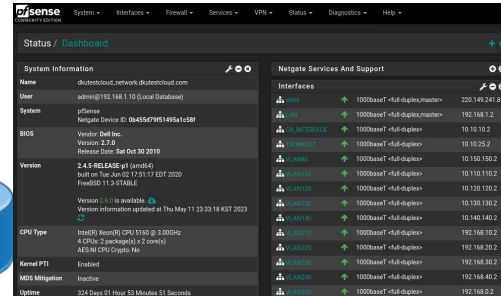
- 트리플O의 구조가 Repo 1대에서 컨트롤러, 컴퓨트 노드의 OS구성을 위해 동시에 많은 파일을 내려받는 구조라,

- 10Gbe는 아니더라도 본딩구성하면 파일서버 쓰는데 대역폭 여유가 있어 문제의 소지가 없을 것으로 생각되었습니다.

이런 내려받아서 설치하는 구조의 경우 레드햇 오픈스택 Ansible 설정을 천천히 봐야겠지만, OKD4 배포 설치해봤을때 네트워크 속도로 인해 Task Timeout 에 걸리며 끝나는 경우가 발생되었음.

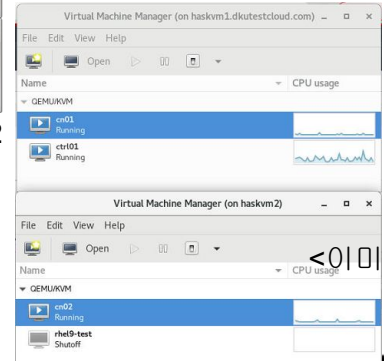
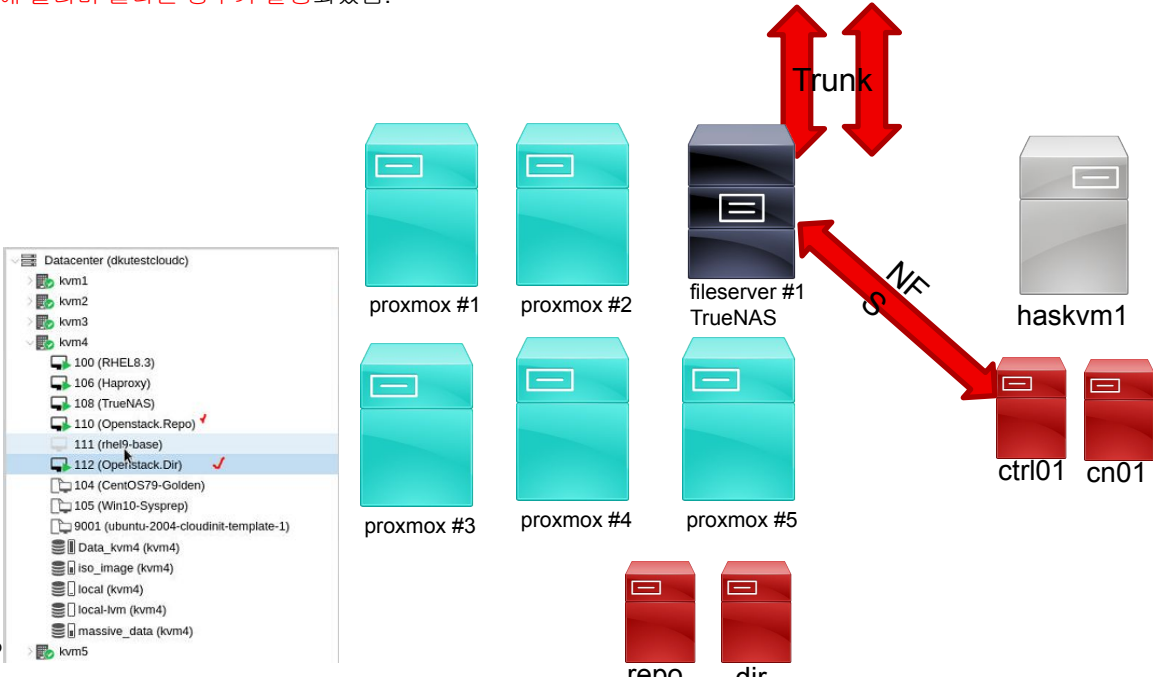


pfSense에서 필요한 만큼 VLAN 네트워크 생성해서 내림



<이미지 : pfSense Console 환경>

1. 네트워크는 Hypervisor 의 호스트에 생성하지 않고 pfSense 를 이용해서 구성 (호스트의 네트워크 라우팅이 복잡해지지 않도록) (pfSense 는 네트워크 설정에 여러 장점이 많습니다.)
2. director 노드에서 지원되는 Ipmi 환경이 아래 처럼 많지만, proxmox vm과 호환되는 driver를 찾기 어려워 데스크톱 두대를 컴퓨터, 컨트롤러 노드 전용 호스트로 구성 함. (proxmox 도 ubuntu 기반이라 VirtualBMC를 설치할 수 있지만, 그렇게까지 안하고 싶었음. , dell ipmi 를 이용해서도 해볼까 했지만, 장비 Generation에 따라 ipmi 구성이 꽤 까다로워서 패스...)
3. 각각 두대의 장비에 VirtualBMC 설치해서 포트 열어줌



<이미지 : KVM Virt-manager>

진행

3 days . 어렵게만 느껴졌던 오픈스택 설치가 너무 쉬워졌습니다. (유지보수 제외)

인프라 구성

Repo서버 구성
director서버 구성

Undercloud
deploy

Overcloud
deploy

계정 생성
클라우드 이미지
업로드

클라우드 사용

Floating IP
DNS 매칭

배포 인프라 구성

1day

클라우드 기반 인프라 환경 배포

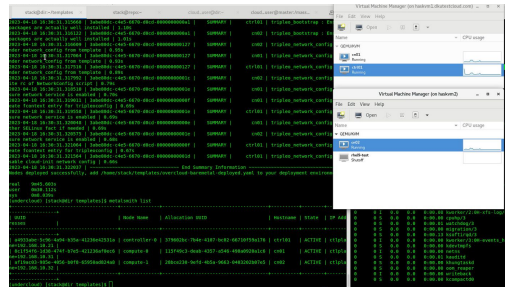
1 day

End User를 위한 환경 배포

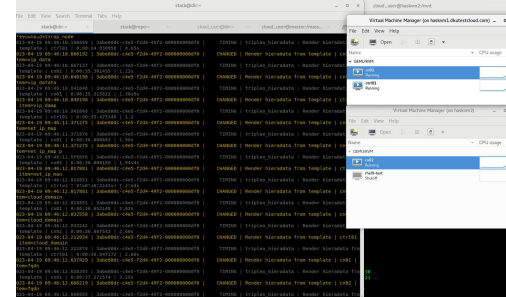
<End User 인스턴스 생성> 1 day



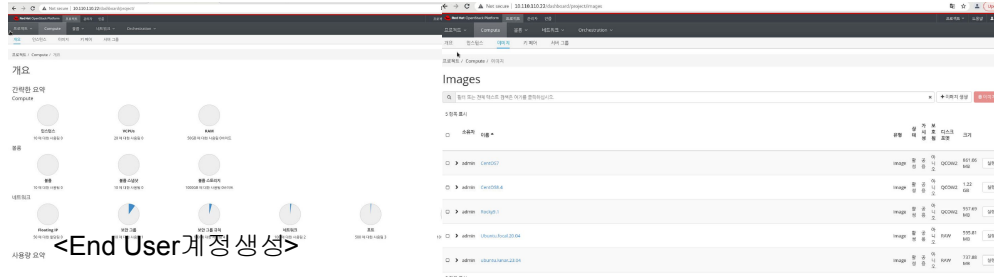
<강력한 스쿼트
랙마운트
2U는 무겁습니다.>



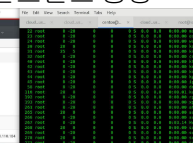
<언더클라우드 배포 캡처>



<오버클라우드 배포 캡처>



<클라우드 이미지 업로드>



4. Nalee Jang nalee999@gmail.com
나래가
와~~ 진도가 상당히 빠르시군요~
다른 분들도 도와드리면 좋겠습니다. ㅎㅎ
컨트롤러 노드 이용하시는 하지 않습니다. 대신 컨트롤러 노드 3중복을 합니다.
그리고, 컴퓨트 노드는 언제든지 추가 할 수 있어요
컨트롤러 노드 추가하려면 현재 구성된 stack을 삭제하고, 컴퓨트 수량 다시 배포하면 됩니다.
컴퓨트 노드는 컴퓨트 노드 추가 후 운영체제 설치후 배포하면 됩니다.
다시 말해 컨트롤러 노드는 사용중에 추가할 수 없다는 말이고, 컴퓨트 노드는 사용중에도 언제든지 추가할 수 있다는 의미입니다.

2023년 4월 19일 (수) 오후 12:33, 이대하(장보라비더) <nalee999@naver.com>님이 작성:

수신: 나리님께,
안녕하세요. 이대하입니다.

백분 확실 감사드립니다~

덕분에 배포도 다 해보고 인스턴스도 만들고 책 끝까지 따라해보았습니다.

저 한가지 더 질문이 있는데, 만약에 제가 컴퓨트 노드를 더 추가한다면 컨트롤러 노드를 이중화한다면지는 부분론
책에 나온내용에서 배워받은 노드 추가하고 구성 파일들 수정해서 배포 과정을 한번 더 돌리던 되는것인지?
심심하게 할 수 있는 방법이 있는것인지? 궁금합니다.



<스승님께 보고?!>

오픈스택 구축 소감

16년도 했던 오픈 스택 배포와는 비교할 수 없을 정도로 쉬워졌다.

테스트로 쓰는 것이라면 어쩌면 플랫폼에 문제가 생긴다면.
아예 새로 배포하는게 빠를지도 모르는 생각이 들었습니다.

컴포넌트들이 예전에 비해 컴팩트해진 느낌이다.

IaC (Terraform , Ansible...) 등이 강력해져 오픈스택과 조합하여 잘 쓴다면
“나 혼자라도 할 수 있겠다.” 라는 용기를 얻었다.

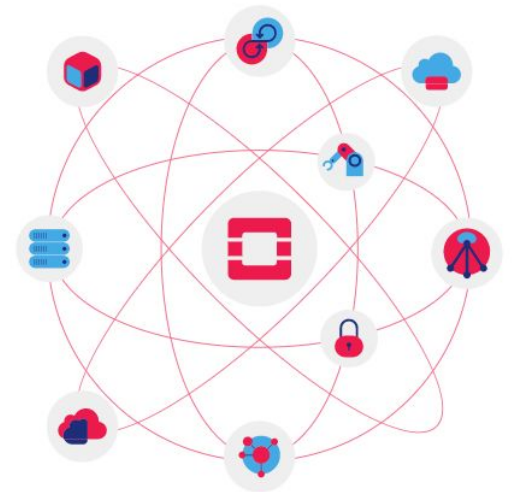
스터디 + 강의 소감

구독 경제가 범람하는 이 시기에 마른 대지위의 한줄기 빗방울 같은 강의 였다.

7주간의 여정동안 경험복인 높은 수준의 강의를 해 주셔서 감사합니다.

오픈스택 컨테이너 환경! 그리고 스터디 참여 스토리~

문성업

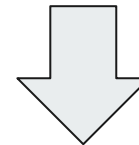


다들 어떻게 모이게 되었을까???

- 생각 보다 많은 인원
- 하지만 안그래도 어려운 오픈스택을 더 어렵게 (Triple-O)
- 그럼 조별 과제는 어떤걸로??

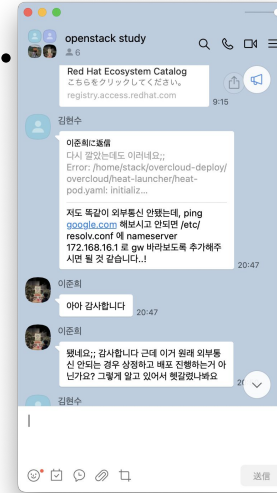
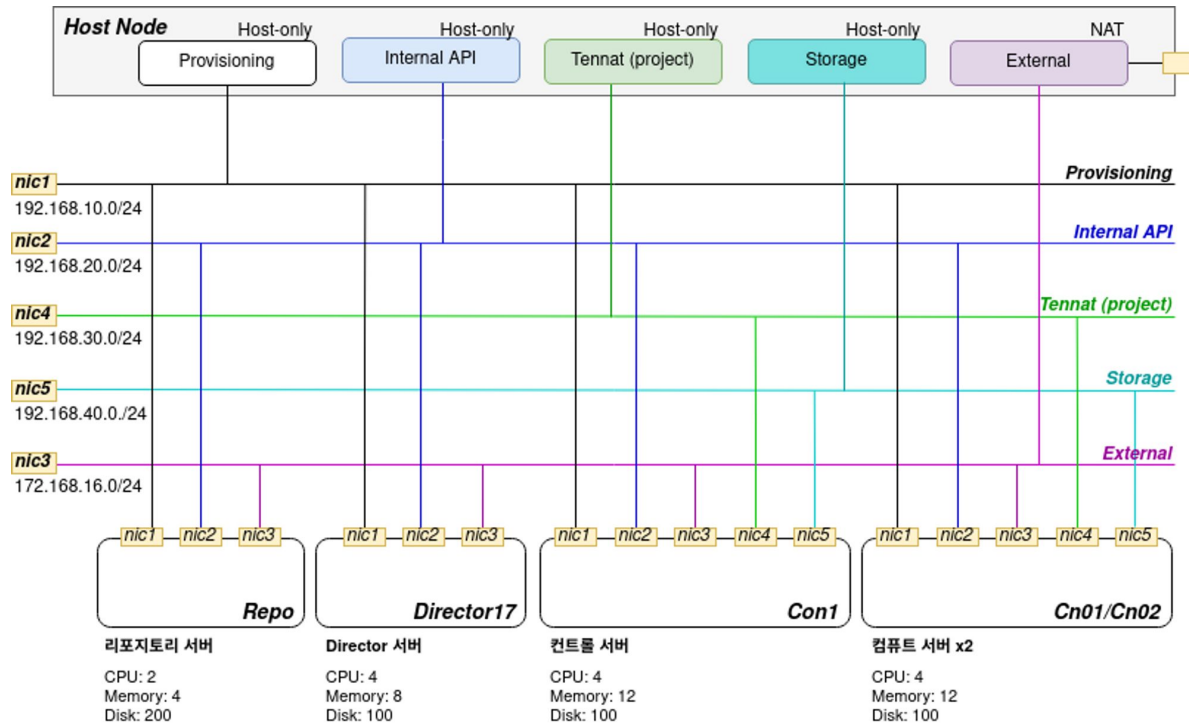


Triple-O 하는 김에 좀
신선한걸로



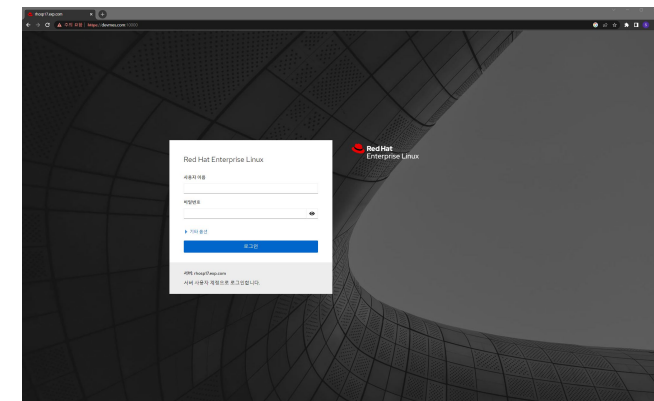
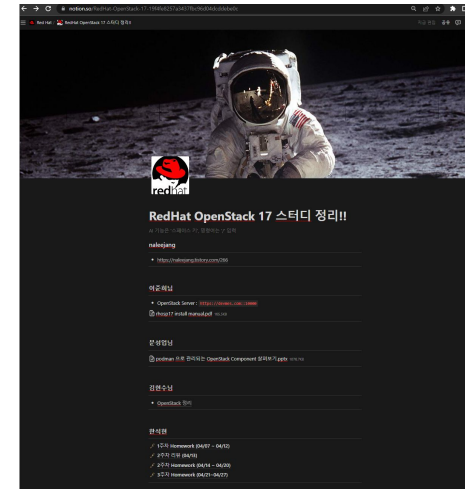
Triple-O를 구성하는 서비스
Pod들의 설정 및 동작
확인해보기
(그나마 K8s와 비슷할 줄 알고...)

이렇게 생긴거 하나 만들어 볼려고..



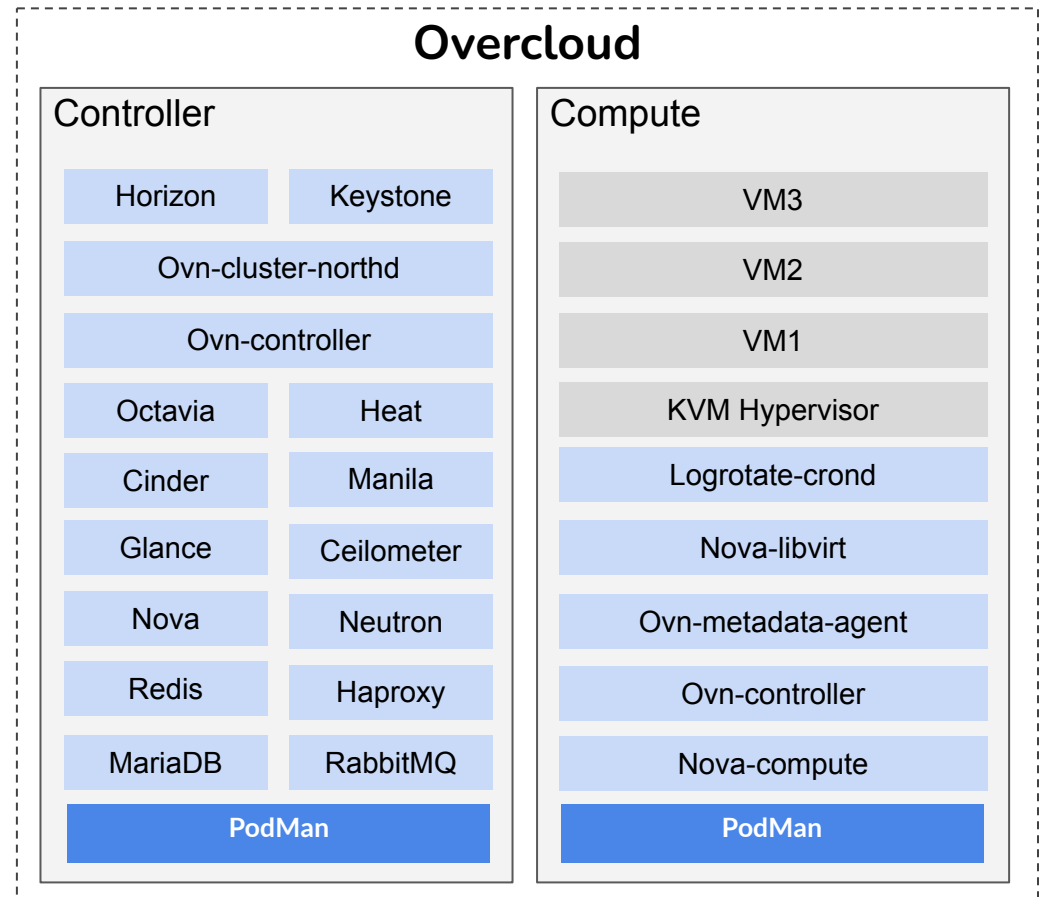
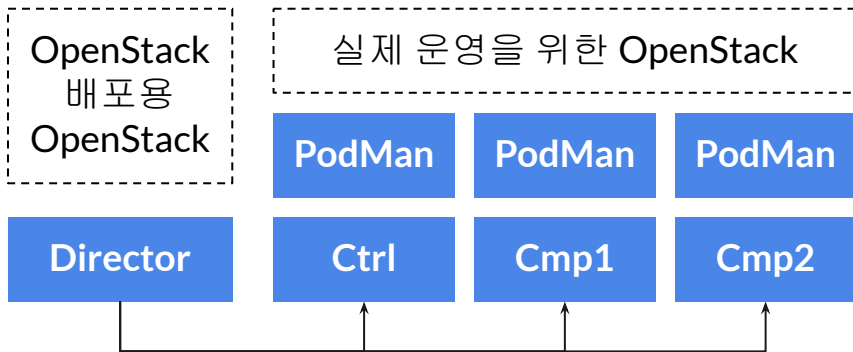
서로 논의하고...

스터디 한 내용 공유하면서



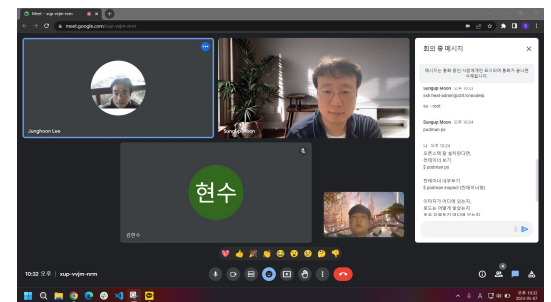
그래서 지금 부터 조별 과제...

- OpenStack-on-OpenStack은 어떻게 올라가 있을까?
 - OpenStack을 구성하는 서비스는 많은데 이걸 어떻게 컨테이너 화 했을까?
 - 생각보다 일반적인 K8s 사용법 하고는 달랐다...



어떻게 생겼는지 하나씩 뜯어 볼려고 하니...

```
[root@ctrl01 ~]# podman ps | awk '{printf("%-70s %s\n", $2, $NF)}' | grep -v "NAMES" | sort
192.168.10.11:8787/rhosp-rhel9/openstack-cinder-api:17.0      cinder_api
192.168.10.11:8787/rhosp-rhel9/openstack-cinder-api:17.0      cinder_api_cron
192.168.10.11:8787/rhosp-rhel9/openstack-cinder-scheduler:17.0  cinder_scheduler
192.168.10.11:8787/rhosp-rhel9/openstack-cinder-volume:pcmklatest openstack-cinder-volume-podman-0
192.168.10.11:8787/rhosp-rhel9/openstack-cron:17.0          logrotate_crond
192.168.10.11:8787/rhosp-rhel9/openstack-glance-api:17.0      glance_api
192.168.10.11:8787/rhosp-rhel9/openstack-glance-api:17.0      glance_api_cron
192.168.10.11:8787/rhosp-rhel9/openstack-haproxy:pcmklatest  haproxy-bundle-podman-0
192.168.10.11:8787/rhosp-rhel9/openstack-heat-api:17.0        heat_api
192.168.10.11:8787/rhosp-rhel9/openstack-heat-api:17.0        heat_api_cron
192.168.10.11:8787/rhosp-rhel9/openstack-heat-api-cfn:17.0    heat_api_cfn
192.168.10.11:8787/rhosp-rhel9/openstack-heat-engine:17.0     heat_engine
192.168.10.11:8787/rhosp-rhel9/openstack-horizon:17.0        horizon
192.168.10.11:8787/rhosp-rhel9/openstack-iscsid:17.0          iscsid
192.168.10.11:8787/rhosp-rhel9/openstack-keystone:17.0        keystone
192.168.10.11:8787/rhosp-rhel9/openstack-keystone:17.0        keystone_cron
192.168.10.11:8787/rhosp-rhel9/openstack-mariadb:17.0         clustercheck
192.168.10.11:8787/rhosp-rhel9/openstack-mariadb:pcmklatest  galera-bundle-podman-0
192.168.10.11:8787/rhosp-rhel9/openstack-memcached:17.0       memcached
192.168.10.11:8787/rhosp-rhel9/openstack-neutron-server:17.0   neutron_api
192.168.10.11:8787/rhosp-rhel9/openstack-nova-api:17.0        nova_api
192.168.10.11:8787/rhosp-rhel9/openstack-nova-api:17.0        nova_api_cron
192.168.10.11:8787/rhosp-rhel9/openstack-nova-api:17.0        nova_metadata
192.168.10.11:8787/rhosp-rhel9/openstack-nova-conductor:17.0   nova_conductor
192.168.10.11:8787/rhosp-rhel9/openstack-nova-novncproxy:17.0  nova_vnc_proxy
192.168.10.11:8787/rhosp-rhel9/openstack-nova-scheduler:17.0   nova_scheduler
192.168.10.11:8787/rhosp-rhel9/openstack-octavia-api:17.0      octavia_api
192.168.10.11:8787/rhosp-rhel9/openstack-octavia-api:17.0      octavia_driver_agent
192.168.10.11:8787/rhosp-rhel9/openstack-octavia-health-manager:17.0 octavia_health_manager
192.168.10.11:8787/rhosp-rhel9/openstack-octavia-housekeeping:17.0 octavia_housekeeping
192.168.10.11:8787/rhosp-rhel9/openstack-octavia-worker:17.0   octavia_worker
192.168.10.11:8787/rhosp-rhel9/openstack-ovn-controller:17.0   ovn_controller
192.168.10.11:8787/rhosp-rhel9/openstack-ovn-nb-db-server:17.0  ovn_cluster_north_db_server
192.168.10.11:8787/rhosp-rhel9/openstack-ovn-northd:17.0     ovn_cluster_northd
192.168.10.11:8787/rhosp-rhel9/openstack-ovn-sb-db-server:17.0  ovn_cluster_south_db_server
192.168.10.11:8787/rhosp-rhel9/openstack-placement-api:17.0    placement_api
192.168.10.11:8787/rhosp-rhel9/openstack-rabbitmq:pcmklatest  rabbitmq-bundle-podman-0
```



이럴때 사용하는 역할 분담

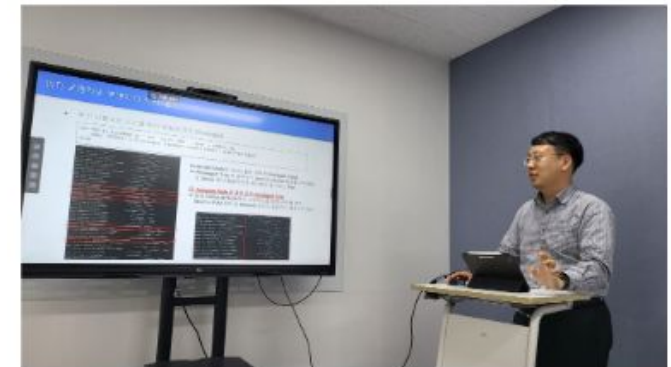
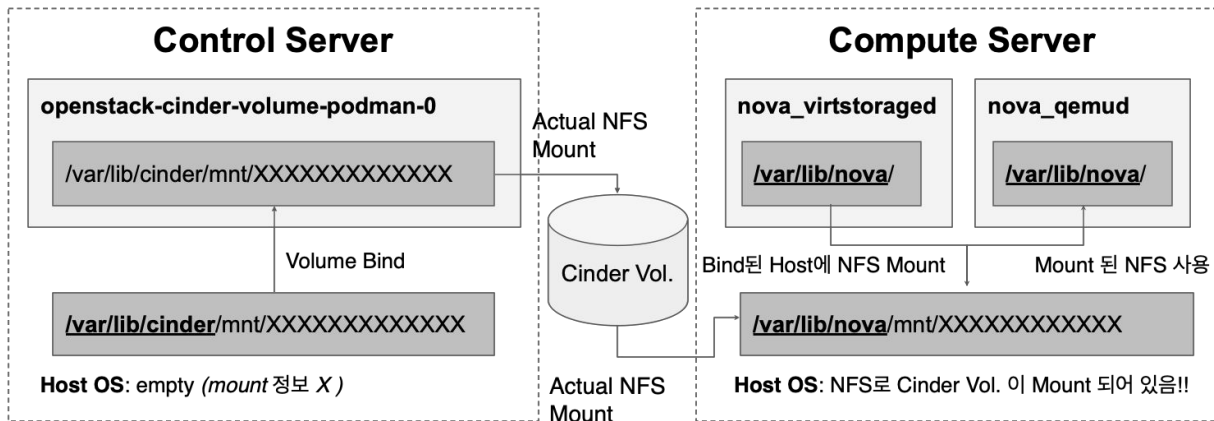
간단히 정복했던 내용들....

- 거의 모든 Pod 들이 Kolla 로 되어 있었음
- Network 들은 Host Network 로 구성
- nova_virt (libvirt 사용주체들) 들만 host namespace
 - 나머지는 private namespace
- Cinder를 통해서 내부의 Mount 정보 및 관리 차이 확인
 - Cinder는 Private Namespace로 Internal NFS Mount
 - Nova는 Host Namespace로 External NFS Mount



TRIPLEO
an OpenStack Community Project

뭐, 뭐지?
난 분명 TripleO를
보고 있었는데....



그래서 모두 고생했던 더 많은 내용을
발표했습니다

소감

- 맨 처음 PackStack으로 입문해서 DevStack에 개별 설치까지 다 해봤는데



스터디 참여할 때 마다 느끼는 신선함, 다음에는 뭐 있을려나....

이거 갖고 업무에 적용하자고 하면 회사 팀원들 죽어나가겠지???

- 대학 졸업 후 15년 만에 해보는 조별과제는 역시 달랐다!
- 마지막 까지 고생하신 멤버 분들에게 감사합니다~

Thank you



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat