

NHN Cloud Openstack 버전업 후기

NHN 클라우드인프라개발랩
장재완

NHN Cloud

- Openstack Icehouse로 개발 시작
- 2015년, Openstack Kilo 버전으로 서비스 출시
- 2021년 Openstack Queens 버전으로 서비스 운영중

버전업하자!

- 2015년 ~ 2017년 : 트러블슈팅, 신규 기능 개발, ...
- 2018년 : 버전업이 필요!
 - Openstack Kilo 는 이미 2016년에 EOL
 - 신규 기능, 버그 픽스 - 백포팅이 점점 많아진다
 - 이제 사람도 많이 늘었고 조금 여유가 생겼네?

버전업 대상

- 버전업된 컴포넌트
 - Heat, Mistral, Ceilometer, ...
 - 초기 구축 이후에 신규로 들어갈 때 이미 높은 버전을 이용
 - 고객 자원 운영과 직접 관계가 없는 컴포넌트

버전업 대상

- 버전업된 컴포넌트
 - Heat, Mistral, Ceilometer, ...
 - 초기 구축 이후에 신규로 들어갈 때 이미 높은 버전을 이용
 - 고객 자원 운영과 직접 관계가 없는 컴포넌트
- 버전업 대상
 - Nova, Neutron, Cinder, Glance

버전업 대상

- 버전업된 컴포넌트
 - Heat, Mistral, Ceilometer, ...
 - 초기 구축 이후에 신규로 들어갈 때 이미 높은 버전을 이용
 - 고객 자원 운영과 직접 관계가 없는 컴포넌트
- 버전업 대상
 - Nova, Neutron, Cinder, Glance
- 버전업 제외
 - Keystone

어디로 버전업할 것인가?

- Openstack 버전
 - Kilo (2015) -> LMNOP -> Queens (2018)
- CentOS vs. Ubuntu
 - CentOS 의 커널과 패키지 버전이 너무 낮음
 - Ubuntu Openstack 지원이 더 좋음
 - Ubuntu 로 갈아타자!

Queens 패키지 준비

- pip → deb 우분투 패키지
- pip : 패키지 의존성 지옥

```
$ more requirements.txt
# The order of packages is significant to the
# of appearance. Changing the order
# process, which may cause weird
#

pbr!=2.1.0,>=2.0.0 # Apache-2
SQLAlchemy!=1.1.5,!1.1.6,!1.1.7
decorator>=3.4.0 # BSD

```

- 패치
 - NHN에서 진행한 패치들을 손 머지 진행
 - 백포팅이 많아서 다행

버전업 순서

- Glance → Cinder → Nova/Neutron
 - Glance 의 영향 범위가 가장 좁다.
 - Queens Nova/Neutron 에서 cinderv3 이용하므로 Cinder 부터 버전업

Glance 버전업

- 2019년 7월 착수, 2020년 2월에 완료
- Glance API v1 deprecated
 - v1 을 사용하는 모든 부서에 연락해서 전환 요청
- Visibility
 - is_public → visibility

Glance 버전업 작업순서

- Queens Glance를 Ubuntu 서버에 사전 구축
- 사용자 액션을 차단 (콘솔, public API)
- Kilo Glance 서버 (API, registry) 중지
- DB 백업
- DB 스키마 업그레이드 (K→LMNOP→Q)
 - `glance-manage db_sync`
- Queens Glance 서버 (API, registry) 시작
- VIP 업데이트

db_sync

- LMNOPQ 까지 모든 배포판에서 `glance-manage db_sync` 실행
 - 정석대로 모든 버전에 대해서 db_sync 실행
 - 어느 버전을 건너 뛴지 찾는 것도 일이다!
 - Virtualbox VM 으로 LMNOP 를 설치

Glance 버전업

- 별 다른 문제 없이 버전업 성공
- 난이도 : ★☆☆☆☆
- 자신감 : +1 상승

MQ 버전업

- RabbitMQ 버전업도 동시에 진행
 - RPC/Notification queue 분리
 - Notification queue : 감사 로그, 과금, ...
 - RPC 큐 설정 : HA 설정 삭제
 - Notification 큐 설정: lazy 이용
 - queue-master-locator=min_master 이용
- RMQ 도 너무 낮은 버전이어서 rolling update 불가
- 신규 RMQ 설치 후 Queens Glance 부터는 신규 RMQ 이용하도록 변경
- Cinder, Nova, Neutron 에서도 동일한 방식으로 신규 RMQ 이용하도록 변경

Cinder 버전업

- 20년 2월 착수, 20년 6월 완료
- NFS를 volume backend 로 이용
- Cinder API v1 deprecated

Cinder 버전업

- 20년 2월 착수, 20년 6월 완료
- NFS를 volume backend 로 이용
- Cinder API v1 deprecated
- 버전업 영향도
 - cinder volume 이 내려간 상태에서 고객 instance에서 볼륨 사용할 때 아무런 문제가 없는가? ➔ 진짜 없음
 - 버전업 중에는 볼륨 신규 생성, 제거 불가

Cinder 버전업 작업순서 (1)

- Queens cinder를 Ubuntu 서버에 사전 구축 (cinder volume도 포함)
- 사용자 액션을 차단 (콘솔, public API)
- 사용자 action을 진행 중인 볼륨이 있는지 확인
- Kilo cinder API, scheduler, cinder-volume 중지
- DB 백업
- DB 스키마 업그레이드 (K→LMNOP→Q)
 - glance때와 동일한 방식으로 진행
 - `cinder-manage db_sync`
 - 생각보다 오래 걸림

Cinder 버전업 작업순서 (2)

- Kilo cinder volume, cinder scheduler 서비스 제거
- Volume backend host 변경
 - `cinder-manage volume update_host`
- Queens Cinder 시작
- volumev3 엔드포인트 등록
- volumev2 엔드포인트 VIP 업데이트

Nova/Neutron 버전업

- 20년 2월 착수, 20년 7월 테스트 시작, 21년 2월 완료
- Nova와 neutron 버전업은 동시에 진행
 - Compute node 에 같이 구동
 - Neutron 쪽 이슈
- Neutron 버전업 내용은 간략히...

Compute node 버전업 방안

- 신규 Queens compute node 구축 + live-migration
 - 운영과 개발 모두에게 부담

Compute node 버전업 방안

- 신규 Queens compute node 구축 + live-migration
 - 운영과 개발 모두에게 부담
- In-place upgrade 로 결정
 - CentOS를 그대로 사용
 - qemu, libvirt 버전업

Compute node in-place 버전업 방안

- virtualenv
 - virtualenv 로 Queens nova 파이썬 패키지를 패키징
 - virtualenv 환경 전체를 compute node에 복사
 - qemu, libvirt 는 host의 RPM 패키지를 업데이트

Compute node in-place 버전업 방안

- virtualenv
 - virtualenv 로 Queens nova 파이썬 패키지를 패키징
 - virtualenv 환경 전체를 compute node에 복사
 - qemu, libvirt 는 host의 RPM 패키지를 업데이트
 - VM 생성은 잘 되기는 하는데...
 - 한 호스트의 VM 들이 다양한 qemu과 라이브러리 버전을 섞어서 구동
 - ➔ 과연 기존 VM 들에 영향을 주지 않는가?
 - 롤백 시에 host RPM 패키지도 같이 롤백해야 함

Compute node in-place 버전업 방안

- B안 – Docker 사용
 - Ubuntu dockerfile 에 nova, qemu, libvirt 를 설치
 - (+) Ubuntu 를 쓸 수 있다
 - (+) 기존 Host 의 패키지 변경이 없어서 롤백이 손쉽다

Dockerizing Nova

- Host pid namespace 사용
 - libvirtd 가 기존 qemu 프로세스를 관리하여야 함
- Host network namespace 사용
 - Host의 ovs 명령어를 사용해야 함
- 그 외의 여러가지 경로들 접근이 가능해야 함

Dockerizing Nova

- Host pid namespace 사용
 - libvirtd 가 기존 qemu 프로세스를 관리하여야 함
- Host network namespace 사용
 - Host의 ovs 명령어를 사용해야 함
- 그 외의 여러가지 경로들 접근이 가능해야 함
- libvirtd & systemd
 - libvirtd 는 systemd 서비스로 구동
 - libvirtd 데몬을 수동으로 구동???

Compute node in-place 버전업 방안

- B안 – Docker 사용
 - Ubuntu dockerfile 에 nova, qemu, libvirt 를 설치
 - (+) Ubuntu 를 쓸 수 있다
 - (+) 기존 Host 의 패키지 변경이 없어서 롤백이 손쉽다
 - (-) Docker 이미지 배포 시스템을 또 구성해야 하나?
 - (-) Docker 안의 libvirtd가 과연 기존 인스턴스에 대한 관리를 잘 할 수 있을까?

Compute node in-place 버전업 방안

- C안 – virtualenv + qemu/libvirt 버전업 안 함
 - virtualenv 로 Queens nova 파이썬 패키지를 패키징
 - virtualenv 환경 전체를 compute node에 복사
 - qemu, libvirt 는 host의 RPM 패키지를 업데이트 그냥 씬

C안 검토

- libvirt/qemu 관련 LMNOPQ 릴리즈노트 확인
 - 낮은 버전 그냥 써도 되겠네?
- Queens Nova 에 고정되어 있는 qemu, libvirt 최소 사용 버전을 수정
- **virtualenv 로 nova 만 패키징해서 써 보자!**

Nova/Neutron 버전업 작업순서

- Queens Nova/Neutron 컨트롤러를 Ubuntu 서버에 사전 구축
- virtualenv로 묶은 nova-compute 패키지 사전 배포
- Kilo Nova controller 중단
- Kilo nova compute, Kilo neutron agent 중단
- Kilo neutron server 중단
- DB 백업

Nova/Neutron 버전업 작업순서

- Nova DB 스키마 업그레이드 (K→LMNOP→Q)
 - Glance/Cinder때와 동일한 방식으로 진행
 - `nova-manage db_sync`, `nova-manage online-migration`
 - 상당히 오래 걸림. 인스턴스 수에 비례
 - Newton에서 nova_api DB 분리되면서 작업이 많음
 - DB purge 추천
- Neutron DB 스키마 업그레이드

Nova/Neutron 버전업 작업순서

- Queens Nova controller, Neutron server 시작
- Nova endpoint v2.1 로 신규 생성
- Nova, Neutron VIP 업데이트
- Neutron agent 재시작
 - 시간 매우 오래 걸림
- Nova compute 시작
 - 시간 매우 오래걸림

Nova/Neutron 버전업 작업순서

- Queens Nova controller, Neutron server 시작
- Nova endpoint v2.1 로 신규 생성
- Nova, Neutron VIP 업데이트
- Neutron agent 재시작
 - 시간 매우 오래 걸림
- Nova compute 시작
 - 시간 매우 오래걸림

➔ 대략 15시간 정도 소요

Nova/Neutron 버전업 롤백

- 21년 1월 작업시 Neutron agent 재시작 작업 이슈
 - Compute node의 수가 너무 많아서 재시작할 때 neutron DB 부하가 너무 높음
- **롤백 결정**
- 여러 리전에서 알파/베타 환경에서 버전업, 롤백 테스트를 10+회 진행
 - 버전업/롤백의 달인
 - 버전업 시간과 롤백 시간을 계속 측정해가면서 보완 작업

대규모 테스트 환경 마련

- 알파/베타 환경에서 Compute node 수가 너무 적다
- VMWare 를 이용해서 2GB RAM VM 을 다수 생성, Compute node로 사용
- 인스턴스는 cirros 로 대량 생성

Nova/Neutron 버전업

- 21년 2월
- 며칠 밤새고 많은 문제를 해결하면서 버전업 성공
- 난이도 : ★★★★★
- 자신감 : +100000 상승

꼭 하고 싶은 이야기

- 오픈스택이 시키는 대로 하면 고객 자원에 영향을 주지 않고 버전업이 됨!
- 리얼 환경과 동일하게 구성하여 테스트가 꼭 필요
 - Control plane 에 충분히 부하를 줄 수 있는 테스트 방법 마련
- 롤백 플랜은 확실히 세우고 롤백도 테스트
- 자신감이 붙을 때까지 테스트 반복

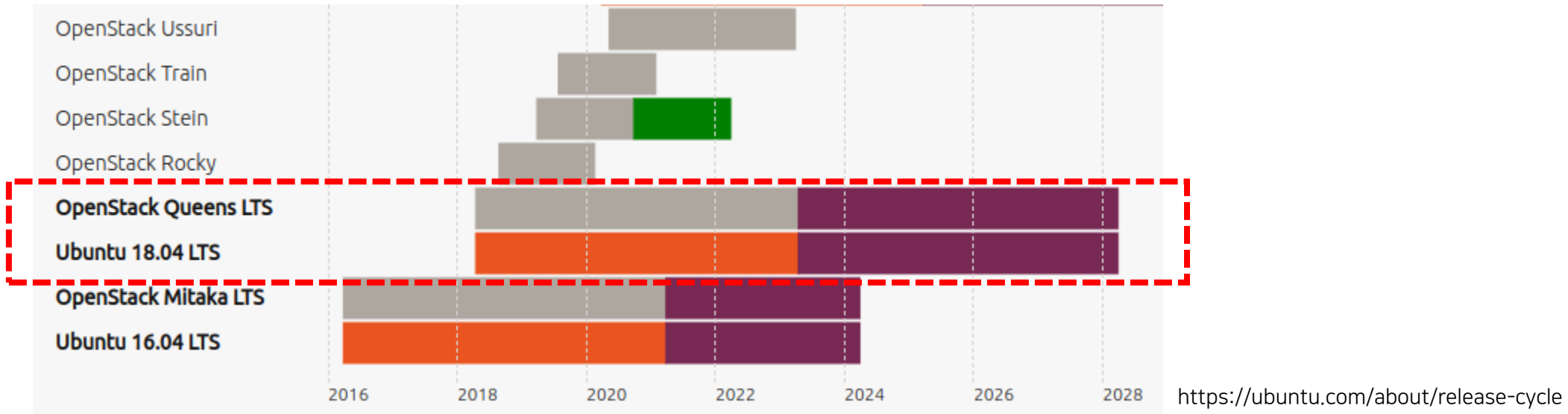
그래도 버전업은 힘들다...

- Kilo Keystone
- 2018년 Queens 버전도 다음 버전으로 업그레이드

Q&A

Extra slide

Ubuntu + Queens



Queens	Extended Maintenance (see note below)	2018-02-28	Unmaintained <i>TBD</i>
Pike	Extended Maintenance (see note below)	2017-08-30	Unmaintained <i>TBD</i>
Ocata	Extended Maintenance (see note below)	2017-02-22	Unmaintained <i>estimated 2020-06-04</i>
Newton	End Of Life	2016-10-06	2017-10-25

<https://releases.openstack.org/>