On-demand Block Storage for Docker

NAVER JangSeon Ryu

My history with openstack & ceph

- 2012, openstack diablo 구축 및 분석 시작
- ~ 2015, public / private cloud 구축 / 운영 (IaaS)
- ~ 2017, openstack, ceph storage, docker plugin, kolla-ansible … (PaaS)

NAVER

NAVER

- 2016, Mitaka Keystone, Horizon 구축 using Kolla (keystone driver 개발)
- 2016, Newton Upgrade using Kolla
- 2016, Ceph Storage(Jewel), Cinder 구축 using kolla
- 2017, Ocata Upgrade 준비
- 2017, Ceph Storage Liminous Upgrade 준비
- 2017, ceph-docker-plugin 개발

현재까지 kolla, kolla-ansible, zun, keystone 등 20여건 이상 commit 진행 중 …

Persistent Storage Service

(Ceph Storage 를 Docker Container 에 Persistent Storage 로 제공하는 방법)

01 Persistent Storage

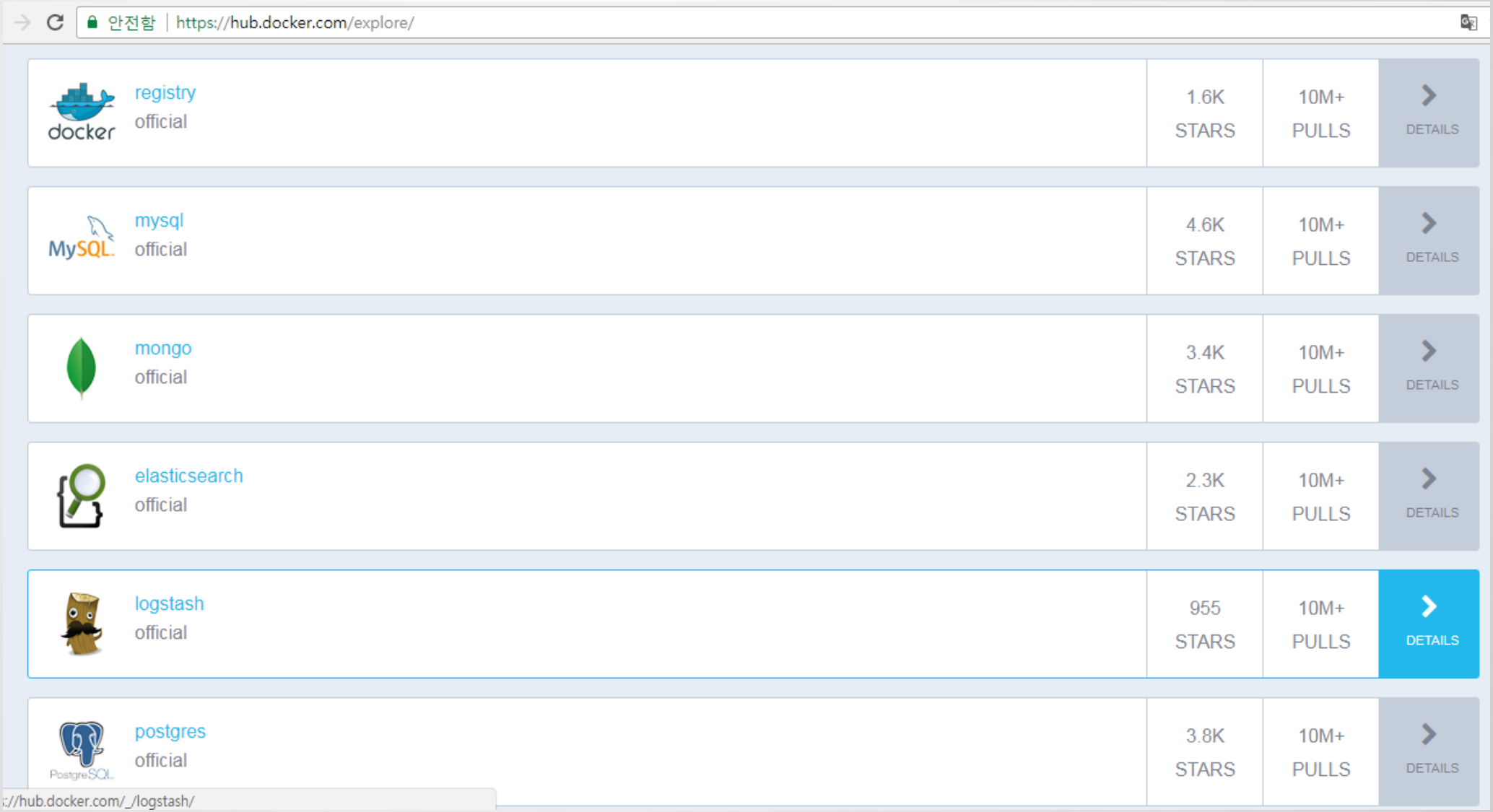
02 Ceph Storage Deployment

03 Docker Volume Plugin

NAVER

Docker 는 Stateless Container 만의 기술인가?

NAVER


안전함 | https://hub.docker.com/explore/
registry
official
1.6K
STARS
10M+
PULLS
DETAILS
mysql
official
4.6K
STARS
10M+
PULLS
DETAILS
mongo
official
3.4K
STARS
10M+
PULLS
DETAILS
elasticsearch
official
2.3K
STARS
10M+
PULLS
DETAILS
logstash
official
955
STARS
10M+
PULLS
DETAILS
postgres
official
3.8K
STARS
10M+
PULLS
DETAILS
://hub.docker.com/_/logstash/

NAVER

Stateful Container를 지원하기 위한 필수 조건은?

Persistent Storage

NAVER

01 Persistent Storage

NAVER

01 Persistent Storage

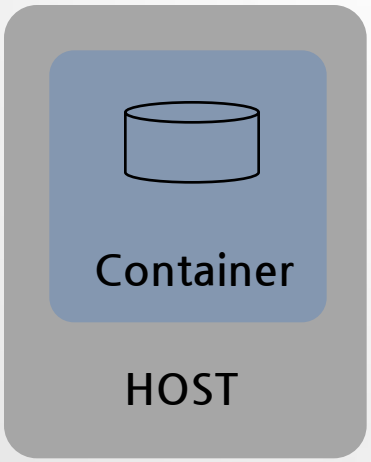
Ephemeral Storage

- Data Lost
- Local Storage
- Stateless Container

VS

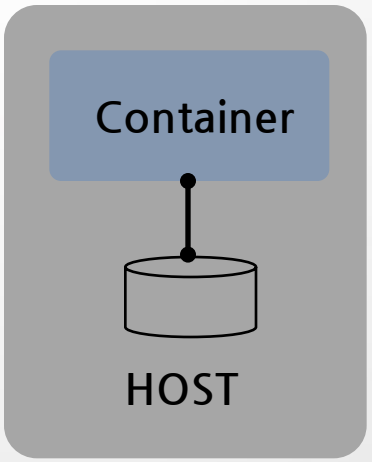
Persistent Storage

- Data Save
- Network Storage
- Stateful Container



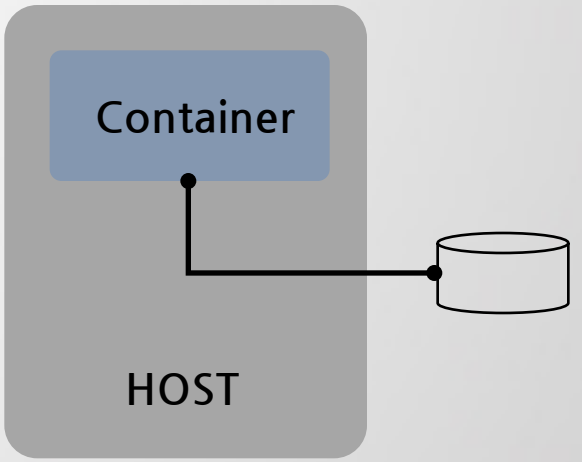
컨테이너 종료 시

데이터 삭제됨



호스트 장애 시

데이터 잃을 수 있음



컨테이너 종료 /

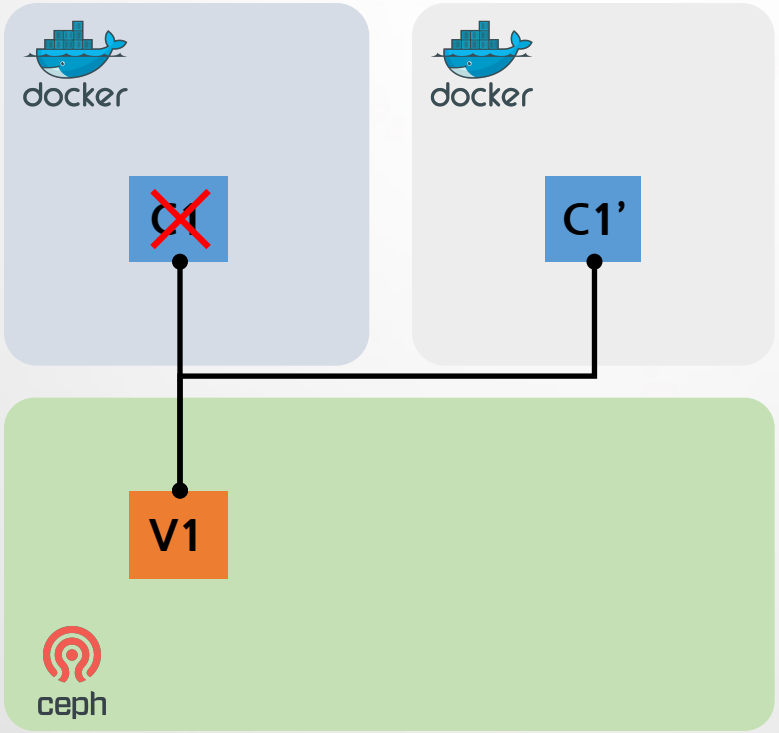
호스트 장애에 영향 없음

NAVER

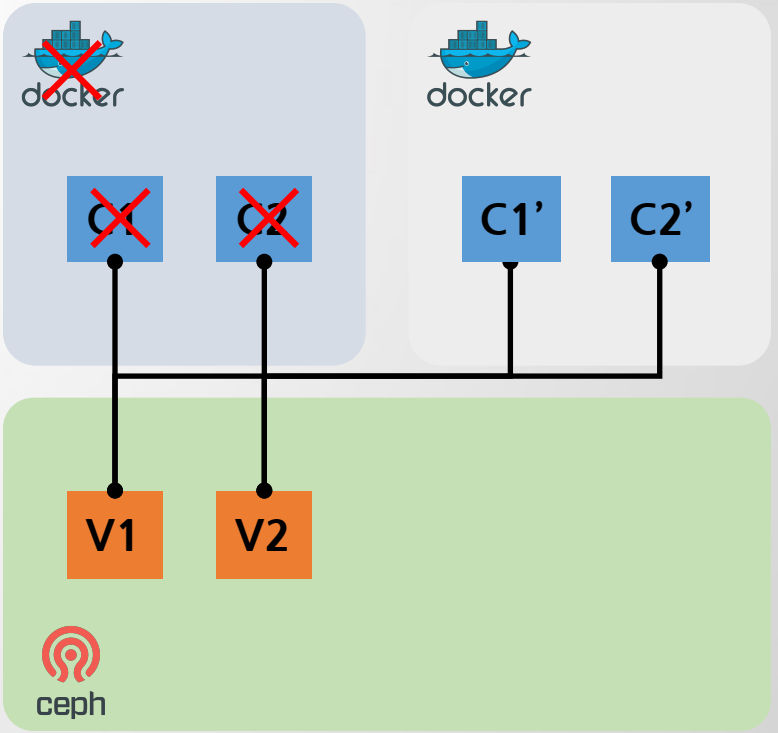
01 Persistent Storage

Docker 의 **Agility** 와 **Automation** 을 유지하는 상태로

Persistent Volume 을 제공



[case1] 컨테이너 종료

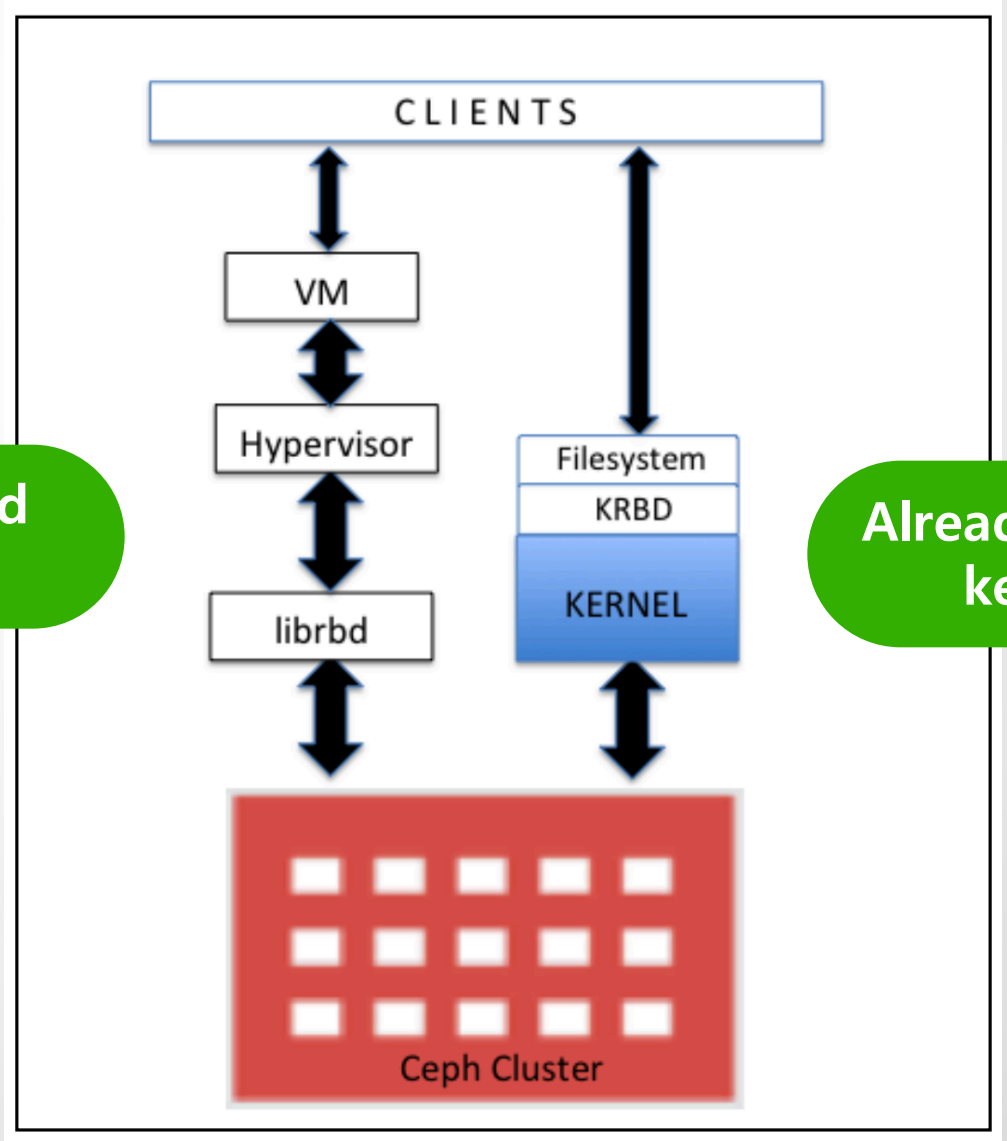


[case2] 호스트 장애

NAVER

01 Persistent Storage

KVM integrated with RBD

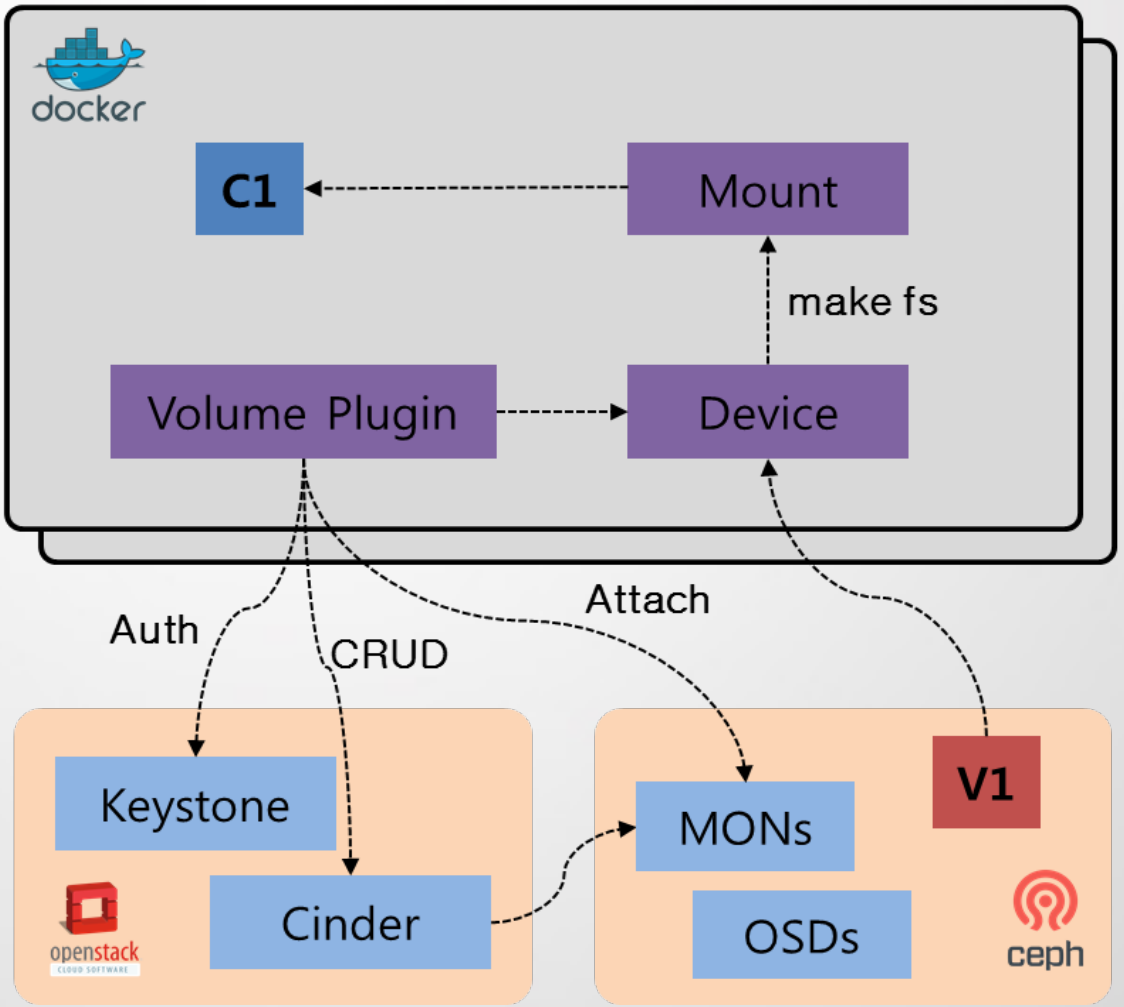


Already involved in linux kernel above 2.6

출처 : learning ceph

01 Persistent Storage

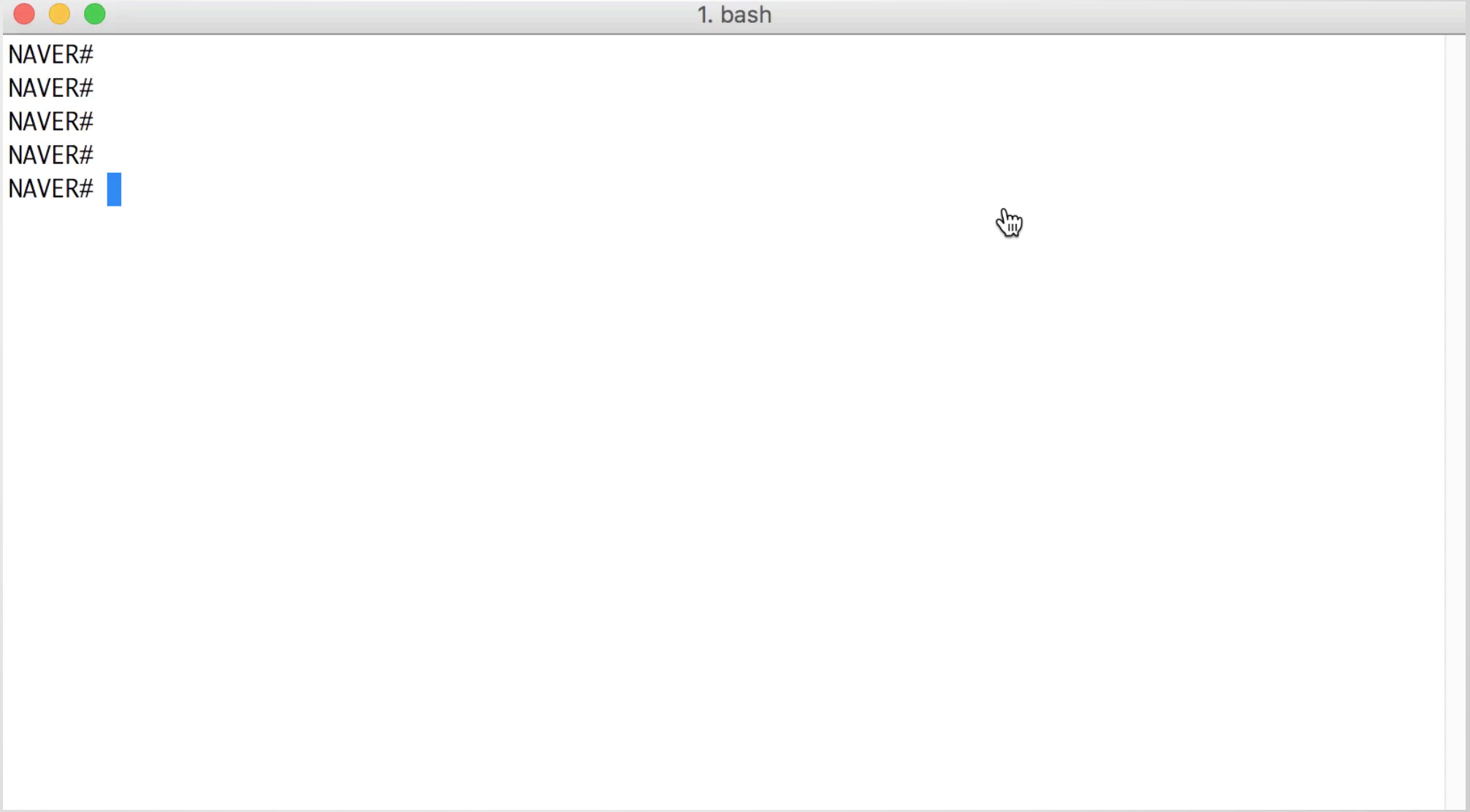
OpenStack Keystone, Cinder 와 Ceph Storage 를 연동하는 Docker Volume Plugin 을 개발함



NAVER

01 Persistent Storage

DEMO



NAVER

02 Ceph Storage Deployment

NAVER

02 Ceph Storage Deployment

Ceph Deployment Tools

- **ceph-deploy** : ceph 에서 제공하는 deployment tool
- **ceph-ansible with vagrant** : virtualbox 환경에 테스트 환경 구축
- **Openstack Kolla(kolla-ansible)** : docker 환경에 container 로 구축
- **OpenStack kolla-kubenetes** : k8s 환경에 container 로 구축
- **OpenStack Helm** : k8s 환경에 container 로 구축
- **ceph-docker** : ceph 에서 제공하는 ceph container 이미지
- …

Ceph Storage 를 Container 로 동작??

과연 괜찮을까…

NAVER

Host

안정성

성능

VS

Container

관리편의성

업그레이드

NAVER

02 Ceph Storage Deployment

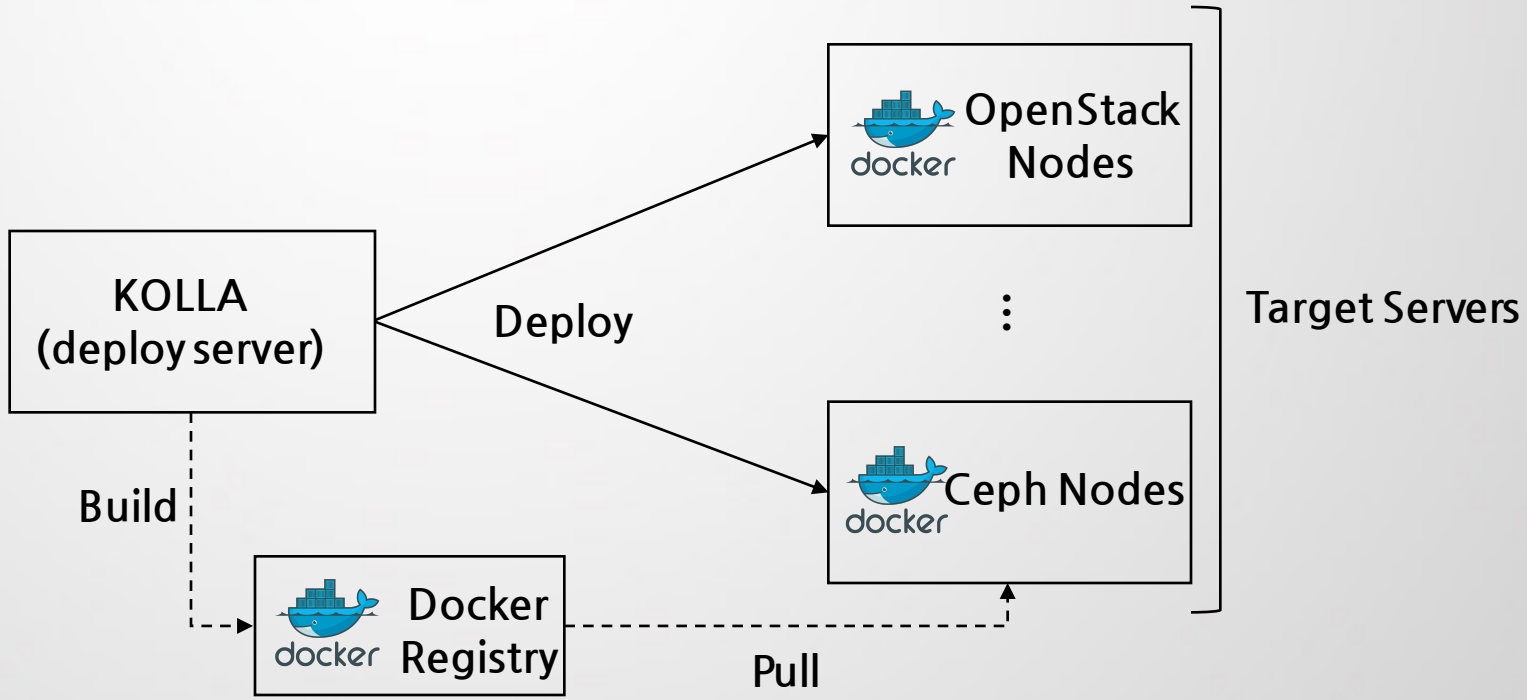
다양한 시도들…

- Kolla-ansible : ceph container deployment 제공
- Kolla-kubernetes
- OpenStack-helm
- ceph/ceph-docker
- …

02 Ceph Storage Deployment

OpenStack Kolla #1

Kolla's mission is to provide **production-ready containers** and **deployment tools for operating OpenStack clouds** that are scalable, fast, reliable, and upgradable using community best practices



02 Ceph Storage Deployment

OpenStack Kolla #2

Kolla Project (7)

- Juno Release
- Redhat
- Dockerfile
- Tools
- Common Library

Kolla-ansible Project (8)

- Newton Release
- 99cloud
- Docker Base

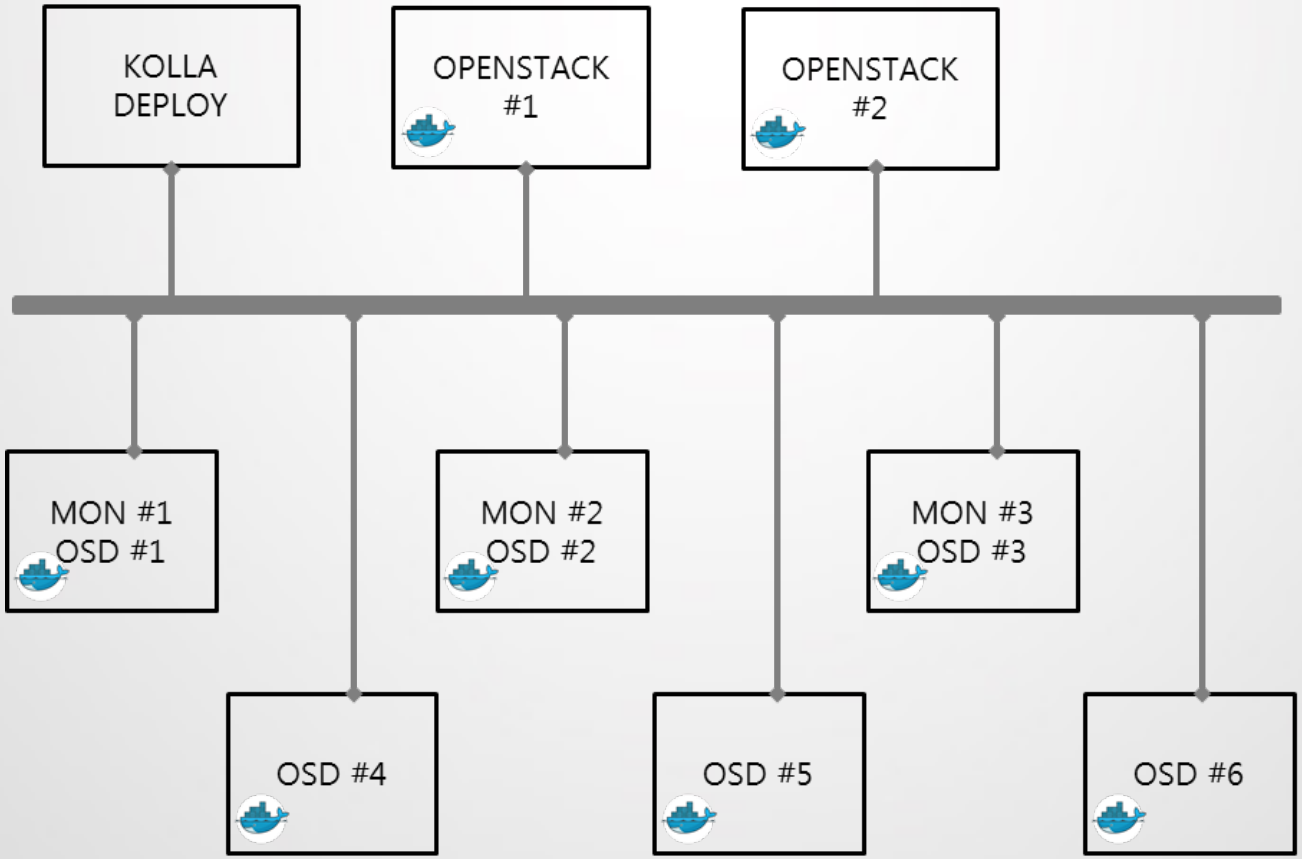
Kolla-kubernetes Project

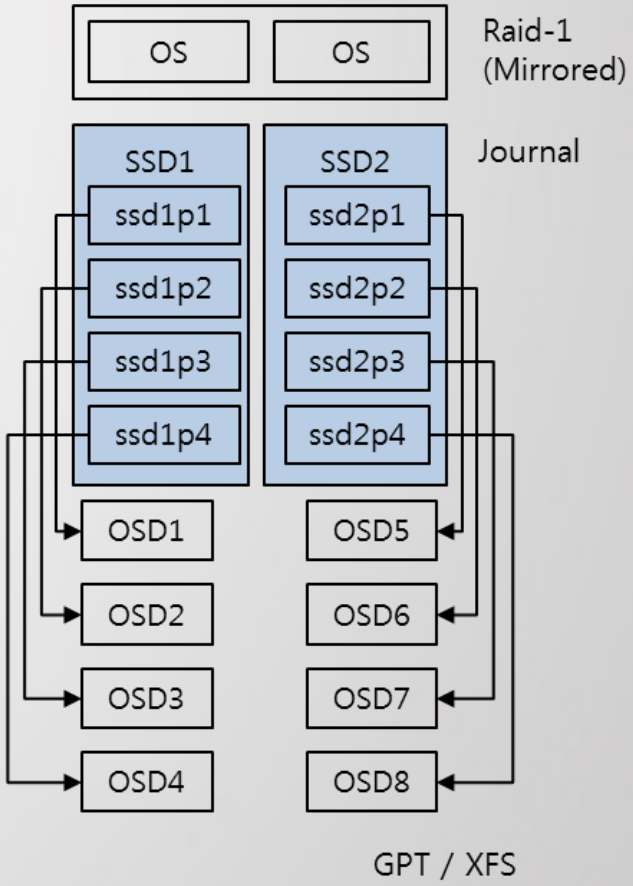
- Newton Release
- Cisco Systems
- Kubernetes Base

02 Ceph Storage Deployment

Architecture

Public
/ Cluster Network (10Gib)





NAVER

02 Ceph Storage Deployment

Deployment using kolla

1. 패키지 설치
$ yum install -y epel-release
$ yum install python-pip
$ pip install -U pip
$ yum install –y python-devel libffi-devel gcc openssl-devel libselinux-python

2. kolla-ansible 설치
$ git clone https://github.com/openstack/kolla-ansible
$ cp -r kolla-ansible/etc/kolla /etc/kolla/
$ cp kolla-ansible/ansible/inventory/* .
$ cd kolla-ansible
$ python setpu.py build
$ python setup.py install

3. 이미지를 별도 빌드할 경우에 도커 설치
$ curl –sSL https://get.docker.io | bash

02 Ceph Storage Deployment

Deployment using kolla

Kolla Deploy 서버 구축 → **Config 수정** → Target Server 설정 → Deploy

 # /etc/kolla/global.yml

```
kolla_base_distro: "centos"
kolla_install_type: "binary"
openstack_release: "4.0.0"
network_interface: "eth0"

enable_ceph: "yes"
enable_cinder: "yes"
enable_horizon: "yes"
enable_keystone: "yes"
enable_mariadb: "yes"
enable_rabbitmq: "yes"
```

 # /etc/kolla/multinode

```
[control]
control1 ansible_user=user-name

[storage-mon]
ceph1 ansible_user=user-name
[storage-osd]
ceph1 ansible_user=user-name
```

 # /etc/kolla/password.yml

$ kolla-genpwd

```
ceph_cluster_fsid: 7e29e46b-7bef-47ee-...
cinder_database_password: NaGGgW3Uw...
cinder_keystone_password: fLoArdYywXV...
cinder_rbd_secret_uuid: ff2f970a-...
```

02 Ceph Storage Deployment

Deployment using kolla

1. 패키지 설치

```
$ sudo yum install -y epel-release
$ sudo yum -y install python-pip
$ pip install -U docker-py
```

2. 도커 설치

```
$ curl -sSL https://get.docker.io | bash
```

3. Disk 설정

```
$ parted /dev/sdb -s -- mklabel gpt mkpart KOLLA_CEPH_OSD_BOOTSTRAP_FOO1_J 1 10G
$ parted /dev/sdb -s mkpart KOLLA_CEPH_OSD_BOOTSTRAP_FOO1 10G 100%

$ parted -l
Number  Start   End     Size    File system  Name                          Flags
1      1049kB  5000MB  4999MB                KOLLA_CEPH_OSD_BOOTSTRAP_FOO1_J
2      2       5000MB  107GB   102GB         KOLLA_CEPH_OSD_BOOTSTRAP_FOO1
```

NAVER

02 Ceph Storage Deployment

Deployment using kolla

 # prechecks
$ kolla-ansible prechecks -i multinode

 # deploy
$ kolla-ansible deploy -i multinode

 # pull image
$ kolla-ansible pull -i multinode

02 Ceph Storage Deployment

Under the hood

```
# docker ps -a
CONTAINER ID   IMAGE                                          NAMES
71d50f2bb687   centos-source-ceph-osd:exp-3.0.2.0010          ceph_osd_8
4731594c438f   centos-source-ceph-osd:exp-3.0.2.0010          ceph_osd_5
51f075439816   centos-source-ceph-osd:exp-3.0.2.0010          ceph_osd_1
03c56a6bd6cf   centos-source-ceph-mon:exp-3.0.2.0010          ceph_mon
eb4a25b21351   centos-source-cron:exp-3.0.2.0010              cron
3c8b5b187031   centos-source-kolla-toolbox:exp-3.0.2.0010     kolla_toolbox
f0bd9f6f3d32   centos-source-heka:exp-3.0.2.0010              heka
```

```
# parted -l
Disk /dev/sdb: 480GB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name                 Flags
 1      1049kB  470GB   470GB   xfs          KOLLA_CEPH_DATA_1
 2      470GB   480GB   9558MB               KOLLA_CEPH_DATA_1_J
```

NAVER

02 Ceph Storage Deployment

Under the hood

```
# blkid
/dev/sdb1: UUID="c4a809b7-b46b-4551-8dfc-529f1553a408" TYPE="xfs" PARTLABEL="KOLLA_CEPH_DATA_1"

# df -h
/dev/sdb1       438G  180G  259G  41% /var/lib/ceph/osd/c4a809b7-b46b-4551-8dfc-529f1553a408

# docker inspect ceph_osd_1 | grep c4a8
"/var/lib/ceph/osd/c4a809b7-b46b-4551-8dfc-529f1553a408:/var/lib/ceph/osd/ceph-1:rw",
"Source": "/var/lib/ceph/osd/c4a809b7-b46b-4551-8dfc-529f1553a408",

# docker exec -it ceph_osd_1 df -h
/dev/sdb1       438G  180G  259G  41% /var/lib/ceph/osd/ceph-1

# ps -ef | grep osd
root      23977  23971  0 Jul04 ?        00:40:55 /usr/bin/ceph-osd -f --public-addr
    --cluster-addr  -i 1 --osd-journal /dev/sdb2 -k /var/lib/ceph/osd/ceph-1/keyring
```

NAVER

02 Ceph Storage Deployment

Tuning - TCMalloc

Memory Allocator : TCMalloc TC(Thread Cache) Size
Default

- Ceph : TCMalloc 128MB
- gperftool 2.4 : 32MB
- OpenStack Kolla : 32MB
- PR : https://bugs.launchpad.net/kolla-ansible/+bug/1693692

```
Samples: 4M of event 'cycles:ppp', Event count (approx.): 651602631668
Overhead  Shared Object                      Symbol
  17.51%  libtcmalloc.so.4.2.8               [.] 0x00000000000249cd
   9.65%  libtcmalloc.so.4.2.8               [.] 0x0000000000027f0b
   1.96%  libtcmalloc.so.4.2.8               [.] 0x00000000000249b7
   1.46%  libtcmalloc.so.4.2.8               [.] 0x000000000003407d
   1.41%  libtcmalloc.so.4.2
   1.31%  libtcmalloc.so.4.2
   0.74%  [kernel]
   0.69%  libc-2.17.so
   0.67%  [kernel]
   0.66%  libtcmalloc.so.4.2
   0.62%  libtcmalloc.so.4.2
   0.58%  libtcmalloc.so.4.2
   0.56%  [kernel]
   0.55%  [kernel]
   0.51%  libtcmalloc.so.4.2
```

```
Samples: 3M of event 'cycles:ppp', Event count (approx.): 119217144867
Overhead  Shared Object                      Symbol
   2.06%  libtcmalloc.so.4.2.8               [.] 0x0000000000037575
   1.41%  libtcmalloc.so.4.2.8               [.] 0x0000000000027f0b
   1.30%  [kernel]                           [k] __schedule
   1.17%  [kernel]                           [k] _xfs_buf_find
   1.15%  libc-2.17.so                       [.] __memcpy_ssse3_back
   1.02%  [kernel]                           [k] _raw_spin_lock
   0.97%  libstdc++.so.6.0.19                [.] std::__ostream_inser
   0.95%  [kernel]                           [k] __switch_to
   0.88%  libpthread-2.17.so                 [.] pthread_mutex_lock
   0.85%  libtcmalloc.so.4.2.8               [.] 0x000000000003674a
```

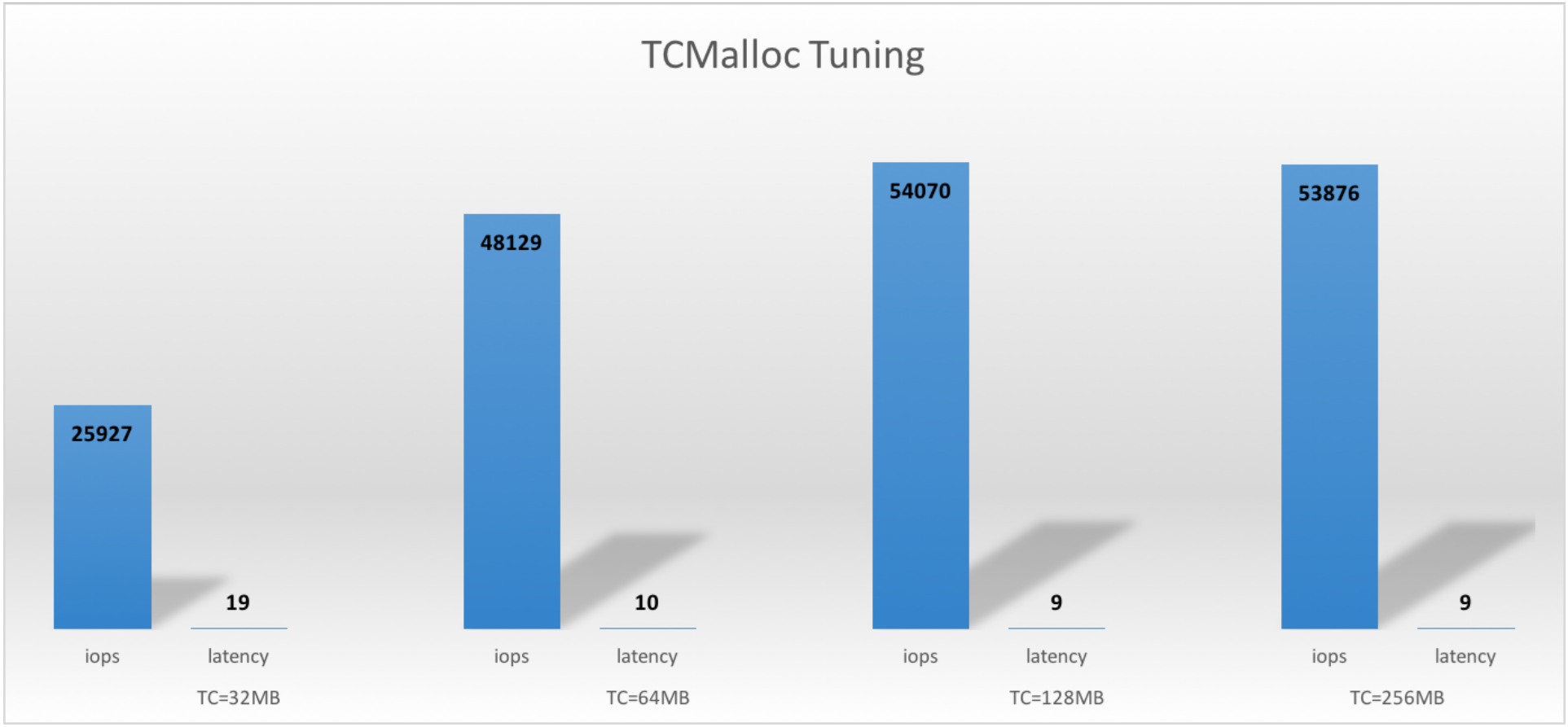
perf top

NAVER

02 Ceph Storage Deployment

Tuning - TCMalloc

TC Size 가 128MB 일때 가장 좋은 성능을 보임 (4k rand-write)



NAVER

02 Ceph Storage Deployment

Tuning - ETC

TCP

```
sysctl -w net.core.rmem_max=67108864
sysctl -w net.core.wmem_max=67108864
sysctl -w net.ipv4.tcp_rmem="4096  87380  33554432"
sysctl -w net.ipv4.tcp_wmem="4096  87380  33554432"
sysctl -w net.core.netdev_max_backlog=30000
```

DISK

Increase IO queue size

```
echo 1024 > /sys/block/sdc/queue/nr_requests
echo 2048 > /sys/block/sdc/queue/read_ahead_kb
```

Change scheduler (noop:SSD/deadline:SATA)

```
echo noop > /sys/block/sdb/queue/scheduler
```

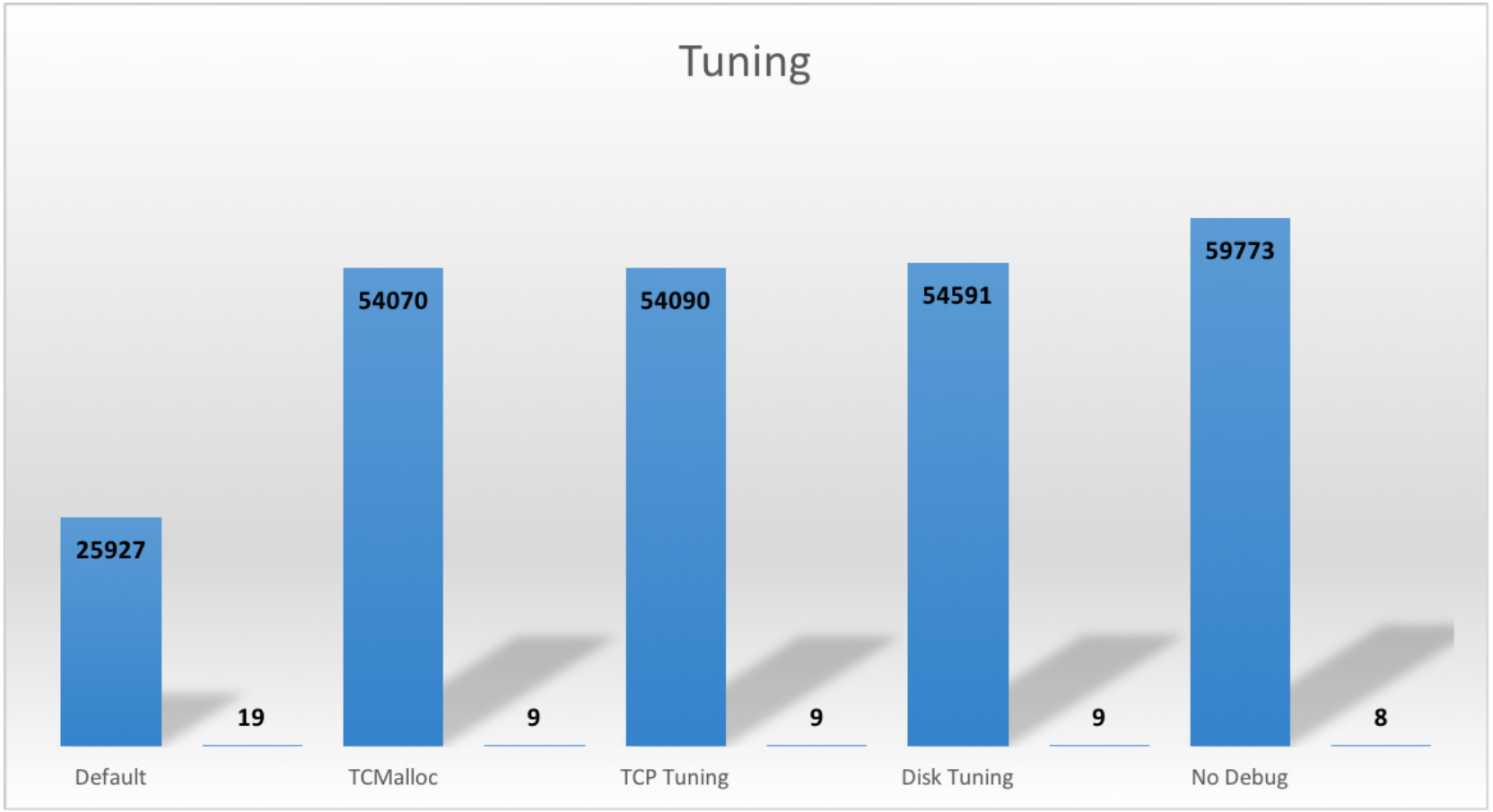
No Debug

```
debug paxos = 0/0
debug journal = 0/0
debug mds_balancer = 0/0
debug mds = 0/0
...
```

02 Ceph Storage Deployment

Tuning - ETC

No Debug 를 제외하고, 다른 튜닝은 효과가 미비함



NAVER

02 Ceph Storage Deployment

Performance

OSD Node : 6

Replicas : 2 copy

Tool : fio

Network : 10Gib

4K Rand Read

평균 : 130,000 IOPS

Latency : ~ 1ms

4K Rand Write

평균 : 64,000 IOPS

Latency : ~ 1ms

64K Rand Read

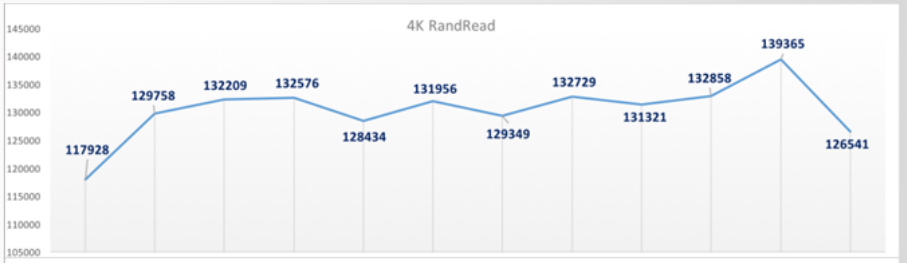
평균 : 552 MB/s

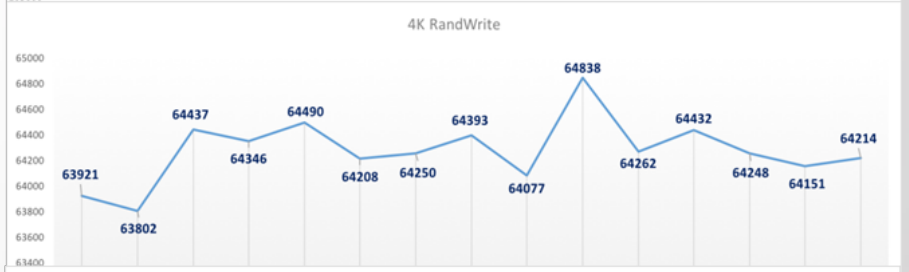
Latency : 20 ~ 26 ms

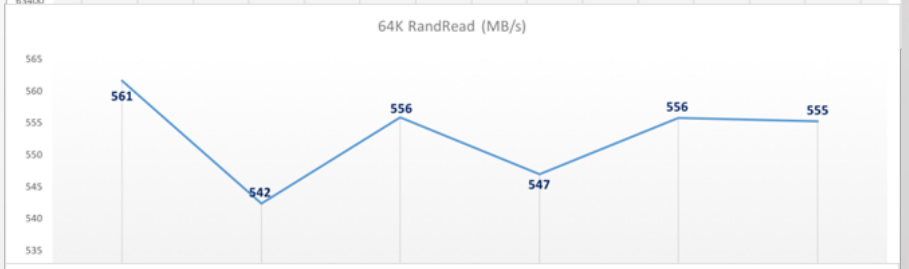
64K Rand Write

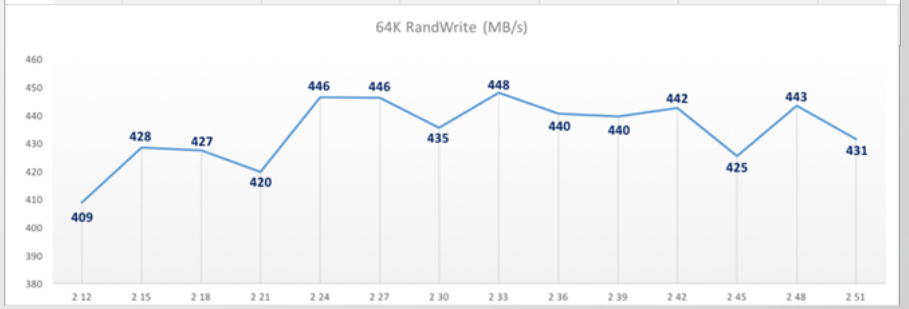
평균 : 429 MB/s

Latency : 20 ~ 26 ms









NAVER

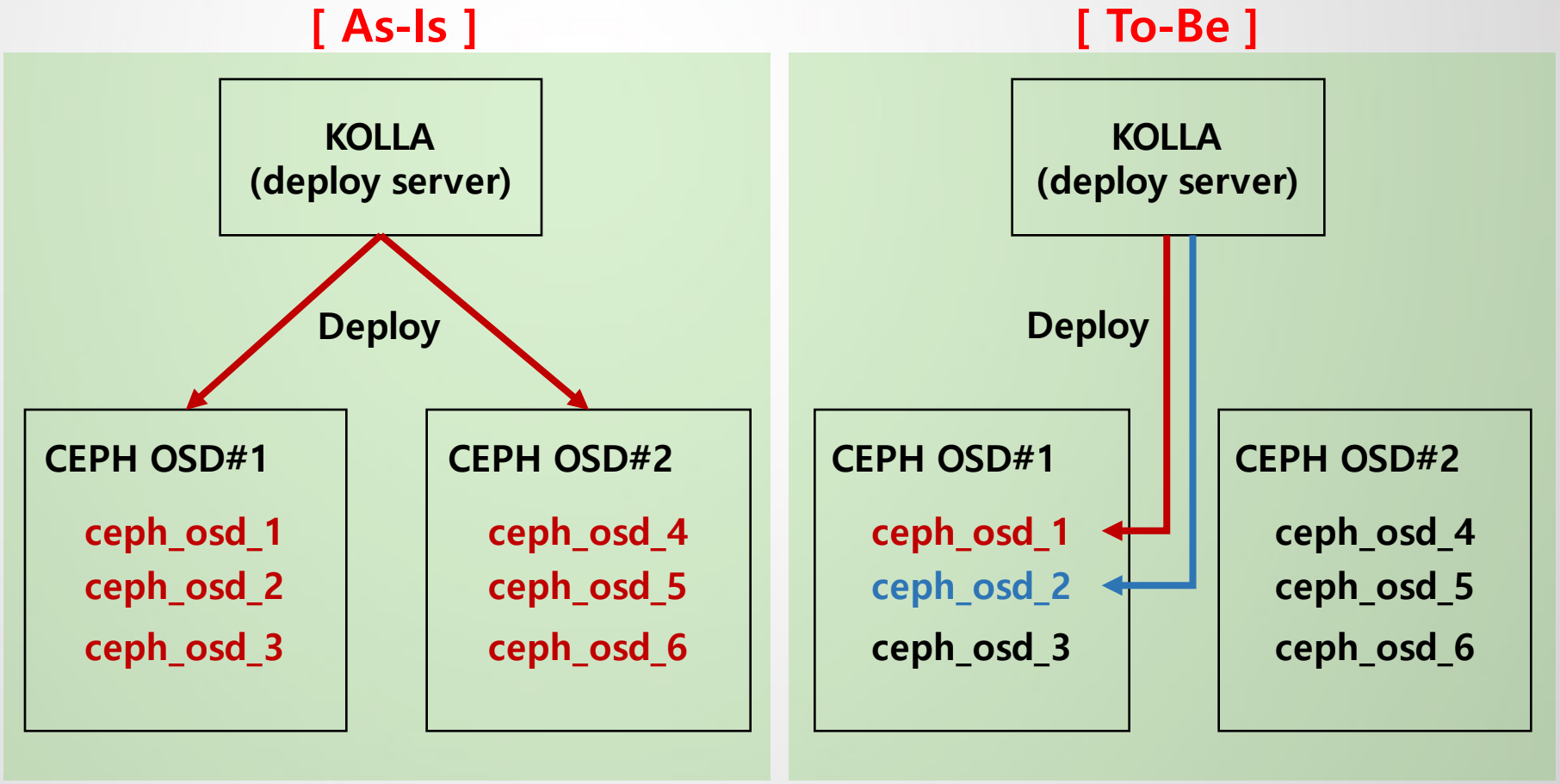
Kolla 를 이용한 Ceph Storage 업그레이드.

가능한가?

NAVER

02 Ceph Storage Deployment

Ceph Storage Upgrade using kolla



Kraken 으로 업그레이드 테스트 완료하였으며, 실환경 업그레이드 완료 후 컨트리뷰션 예정

NAVER

03 Docker Volume Plugin

NAVER

이미 많은 Volume Plugin 이 지원되는데..
새로 만들필요가..?

NAVER

03 Docker Volume Plugin

Docker Volume Plugin

DigitalOcean

25

gce-docker

Flocker™ by ClusterHQ™

OpenStack Fuxi

NetApp™

nimble storage

Local Persistence

Contiv

REX-Ray

DRBD

Quobyte

BeeGFS

Netshare

GlusterFS

HPE

https://docs.docker.com/engine/extend/legacy_plugins/#volume-plugins

NAVER

03 Docker Volume Plugin

Docker Plugin 개발

Docker Go Plugin SDK : docker/go-plugins-helpers

Example using TCP sockets:

```
d := MyVolumeDriver{}
h := volume.NewHandler(d)
h.ServeTCP("test_volume", ":8080")
```

Example using Unix sockets:

```
d := MyVolumeDriver{}
h := volume.NewHandler(d)
h.ServeUnix("root", "test_volume")
```

Volume plugin protocol

/VolumeDriver.Create

/VolumeDriver.Remove

/VolumeDriver.Mount

/VolumeDriver.Path

/VolumeDriver.Unmount

/VolumeDriver.Get

/VolumeDriver.List

/VolumeDriver.Capabilities

https://github.com/docker/go-plugins-helpers/tree/master/volume

NAVER

03 Docker Volume Plugin

Docker Plugin 문제점

Plugin 이 내려간 경우,

docker daemon 에서 volume 관련 요청에 15초간 wait 상태가 발생한다.

아래와 같은 volume driver 에 요청이 필요한 명령 수행 시

$ docker volume ls

$ docker run

$ systemctl restart docker

/var/log/messages

dial unix /run/docker/plugins/ceph.sock: connect: connection refused, retrying in 1s”

dial unix /run/docker/plugins/ceph.sock: connect: connection refused, retrying in 2s”

dial unix /run/docker/plugins/ceph.sock: connect: connection refused, retrying in 4s”

dial unix /run/docker/plugins/ceph.sock: connect: connection refused, retrying in 8s”

03 Docker Volume Plugin

Docker Plugin V2

Plugin 을 설치, 시작, 중지, 삭제, 업그레이드 등의 작업을 지원

- Docker Engine 1.13 이후

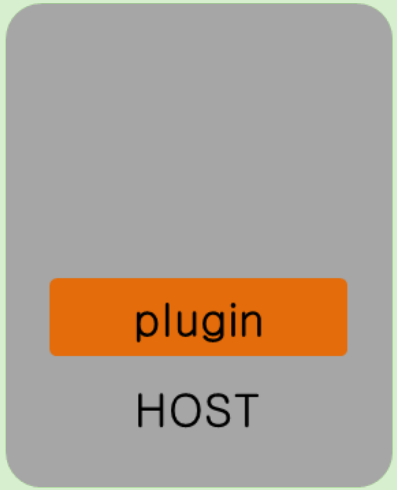
지원 기능

- create : plugin 생성
- disable : plugin 중지
- enable : plugin 시작
- install : plugin 설치
- push : plugin 을 registry 로 업로드
- upgrade : plugin 업그레이드
- rm : plugin 삭제
- ls : plugin 조회
- inspect : plugin 상세조회
- set : plugin metadata 설정

03 Docker Volume Plugin

Plugin 진화

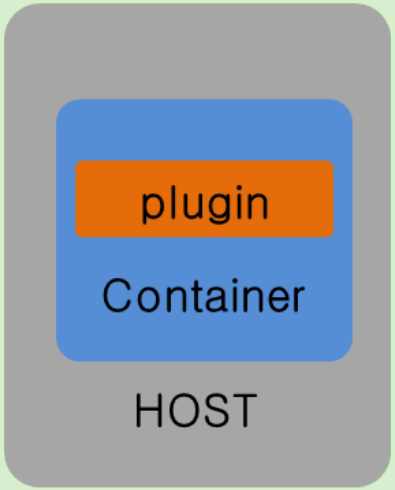
[Host]



Plugin 관리(업그레이드)

어려움

[Container]

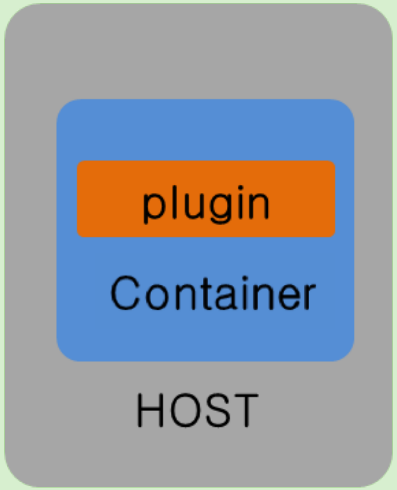


Mount 실행 시 Host 에는 보이지 않음

/proc/1/ns/mnt → container read

nsenter --mount=/…/mnt -- mount

[Plugin v2]



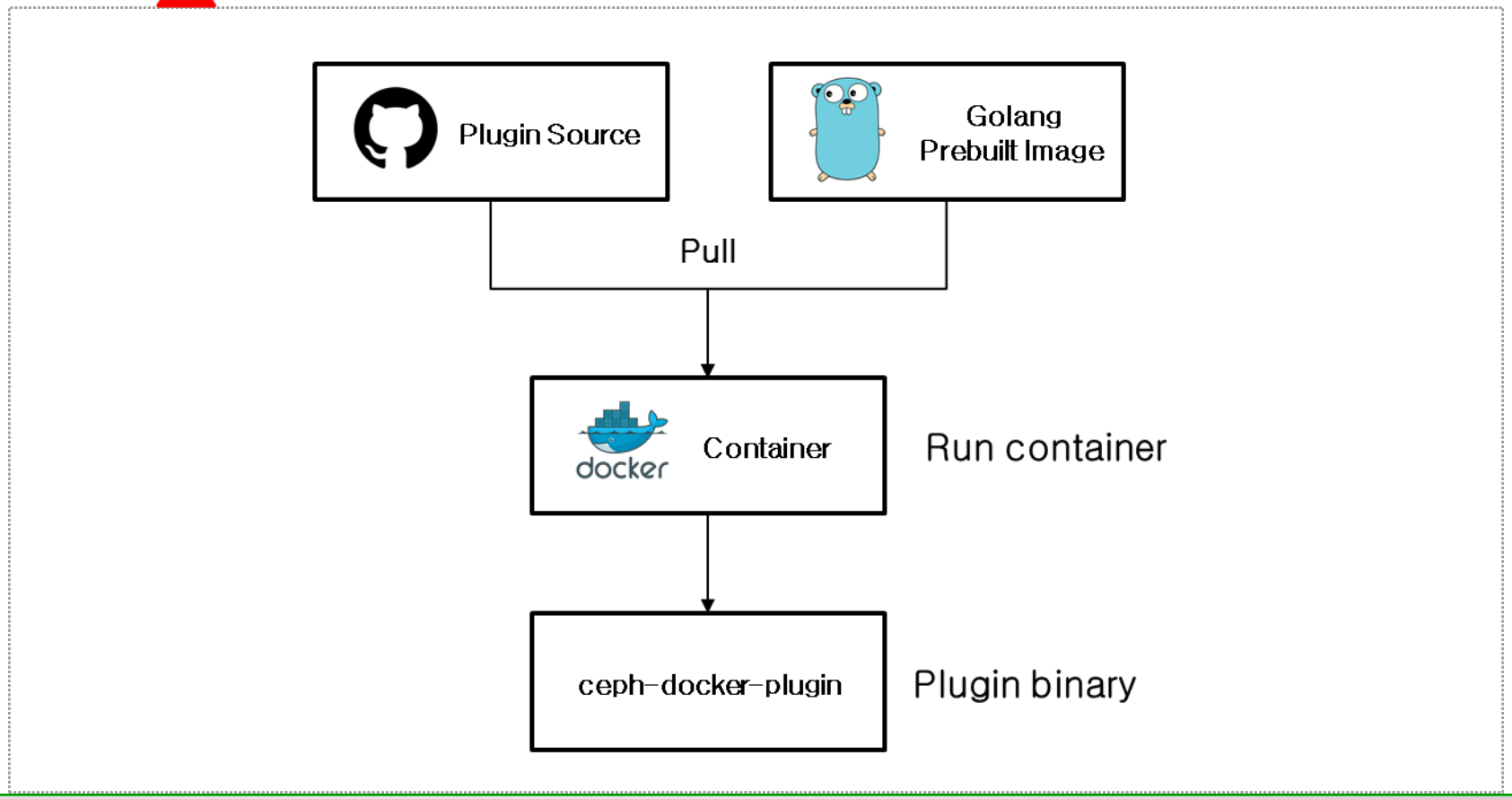
Upgrade 지원

Propagated Mount 지원

03 Docker Volume Plugin

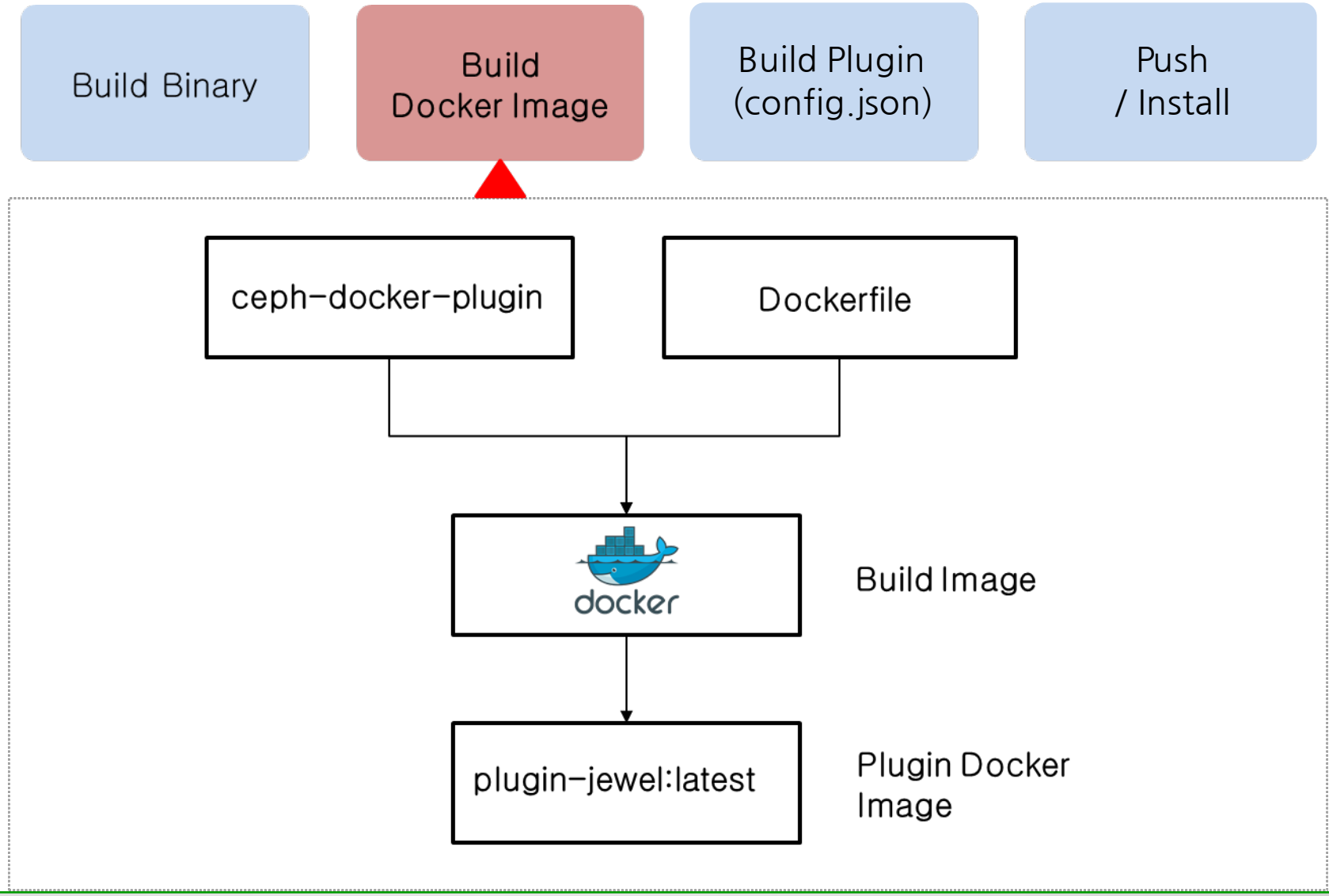
Docker Volume Plugin 생성

Build Binary | Build Docker Image | Build Plugin (config.json) | Push / Install



03 Docker Volume Plugin

Docker Volume Plugin 생성



03 Docker Volume Plugin

Docker Volume Plugin 생성

Build Binary → Build Plugin (config.json) → **Build Plugin (config.json)** → Push / Install

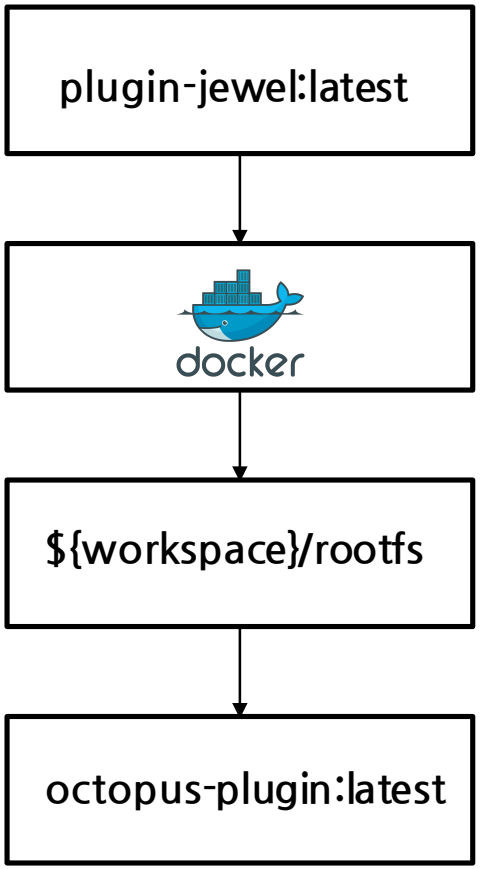
```
"entrypoint": [
    "/entrypoint.sh"
],
"Env": [
  {
    "name": "CLUSTER",
    "description": "cluster name",
    "value": "unset",
    "Settable": [
      "value"
    ]
  }
],
"interface": {
    "socket": "ceph.sock",
    "types": [
        "docker.volumedriver/1.0"
    ]
},
```

```
"Linux": {
    "Capabilities": ["CAP_SYS_ADMIN"],
    "AllowAllDevices": true,
    "Devices": [
    {
     "path":"/dev"
    }
    ]
  },
  "Mounts": [
     {
        "destination": "/dev",
        "options": [
            "rbind"
        ],
        "source": "/dev",
        "type": "bind"
     },
  "network": {
      "type": "host"
  },
  "PropagatedMount": "/var/lib/ceph/mount"
```

03 Docker Volume Plugin

Docker Volume Plugin 생성

Build Binary | Build Docker Image | **Build Plugin (config.json)** | Push / Install



Run Container

```
$ docker create plugin-jewel true
```

Export Container

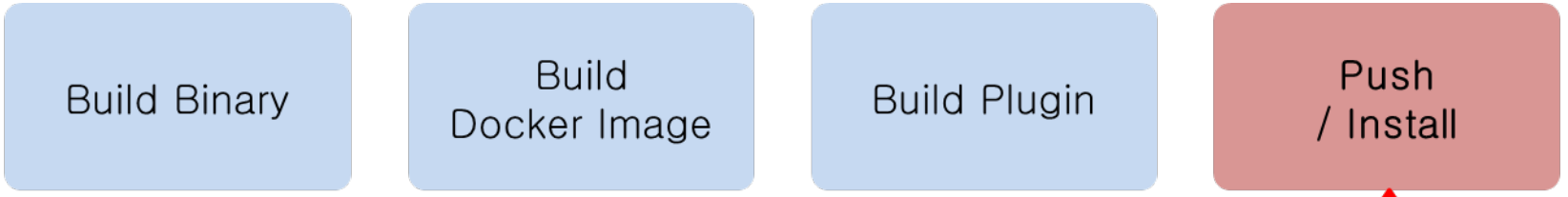
```
$ docker export "$ID" |
  tar -x -C ${workspace}/rootfs
```

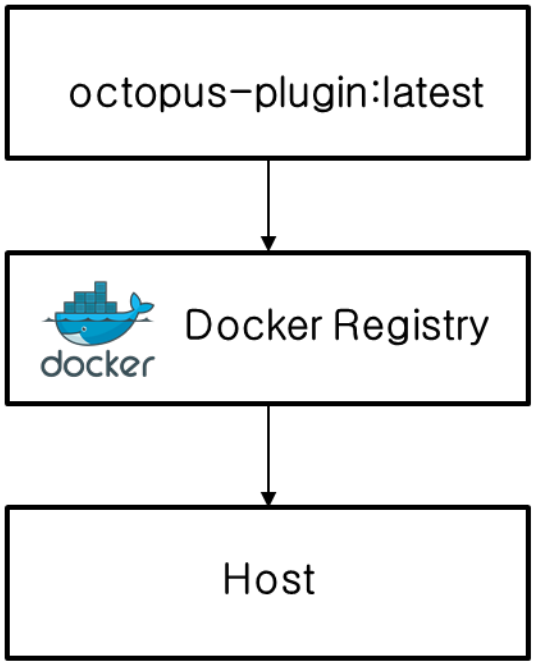
Create Plugin

```
$ docker plugin create          ₩
        octopus-plugin:latest   ₩
        ${workspace}/rootfs
```

03 Docker Volume Plugin

Docker Volume Plugin 생성





Push Plugin

$ docker plugin push octopus-plugin

Plugin Installation

$ docker plugin install octopus-plugin --alias ceph

03 Docker Volume Plugin

Under the hood

```
# lsmod | grep rbd
rbd                    83938  4
libceph               282630  1 rbd
```

← 커널모듈 활성화

```
# docker plugin ls
ID                  NAME                DESCRIPTION           ENABLED
e63e2c0c68be        ceph:latest         ceph docker plugin    true
```

```
# ls -al /var/lib/docker/plugins/
drwx------  4 root root  60 Jul 12 09:42 e63e2c0c68beda51becaeee66a7a02273092f0136d918d9a1e49f4714550750b
```

```
# ls -al e63e2c0c68beda51becaeee66a7a02273092f0136d918d9a1e49f4714550750b/
-rw-------  1 root root 2822 Jul 12 09:42 config.json
drwxr-xr-x  2 root root    6 Jul 12 09:42 propagated-mount
drwxr-xr-x 20 root root 4096 Jul 12 09:42 rootfs
```

```
# ls -al propagated-mount/
drwxr-xr-x 2 root root 6 Jul 13 01:27 test-vol-04
drwxr-xr-x 2 root root 6 Jul 13 01:29 test-vol-05
```

NAVER

03 Docker Volume Plugin

Under the hood

```
# docker-runc list
```

Plugin 은 docker ps –a 에서는 나타나지 않음

```
ID                                                                 PID         STATUS
                                CREATED
e63e2c0c68beda51becaeee66a7a02273092f0136d918d9a1e49f4714550750b   13451       running
```

```
# docker-runc exec -t e63e2c0c68beda51becaeee66a7a02273092f0136d918d9a1e49f4714550750b bash
```

```
# rbd showmapped
id pool    image        snap device
0  volumes test-vol-05 -    /dev/rbd0
1  volumes test-vol-04 -    /dev/rbd1
```

Plugin 내부에서는 ceph 관련 명령 수행이 가능함

Plugin 내부에서는 정의된 위치에 마운트됨

```
# df -h
/dev/rbd0       20G   33M   20G   1% /var/lib/ceph/mount/test-vol-05
/dev/rbd1       20G   33M   20G   1% /var/lib/ceph/mount/test-vol-04
```

```
# df -h
/dev/rbd0       20G   33M   20G   1% /var/lib/docker/plugins/
/dev/rbd1       20G   33M   20G   1% /var/lib/docker/plugins/
                                      e63e2c0c68beda51becaeee66a7a02273092f0136d/propagated-mount/test-vol-05
                                      e63e2c0c68beda51becaeee66a7a02273092f0136d/propagated-mount/test-vol-04
```

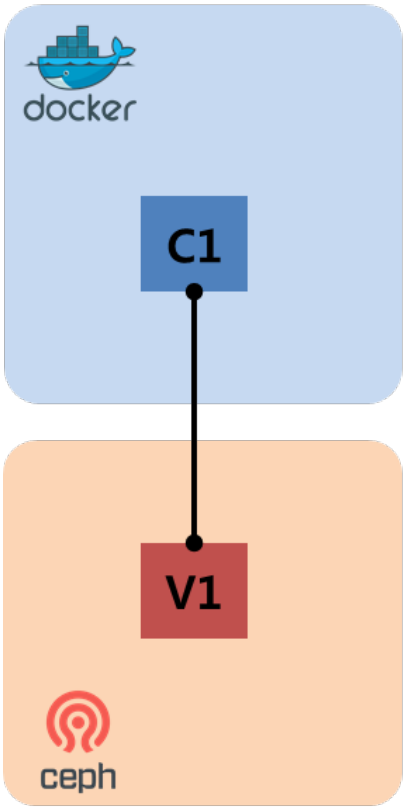
Host 에서는 plugin uuid 내 propagated-mount 에 마운트됨

NAVER

03 Docker Volume Plugin

Use Case. docker run

Volume 생성 후 Container 에 할당이 가능함




```
$ docker run --volume-driver=ceph ₩
--volume hello-openstack:/data -it centos

[root@c220963b6984 # df -h | grep data
/dev/rbd0       10G   33M   10G   1% /data
```

```
$ docker volume create --driver=ceph ₩
--name hello-openstack ₩
-o size=10 ₩
-o project-id=4d47fb34b45a4b68aac9051ffba23217
```

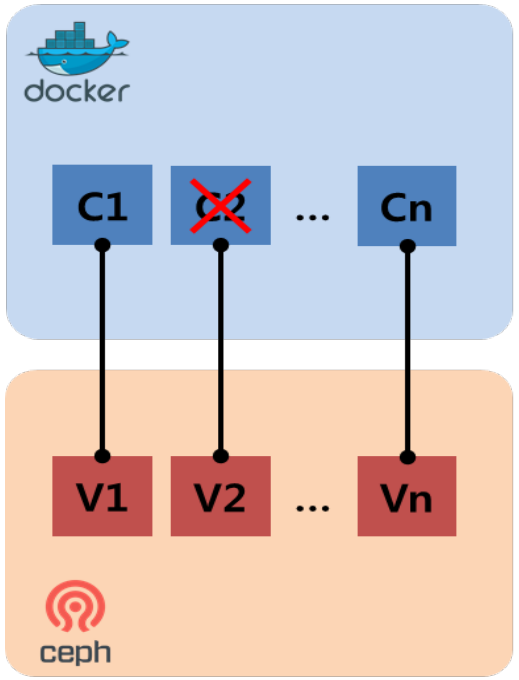
특정 Project 내 볼륨을 생성할 수 있음

NAVER

03 Docker Volume Plugin

Use Case. docker service (replicas=N)

Replication 이 가변적이므로, 볼륨을 미리 만들기가 어려움




```
$ docker service create ₩
--name os-day  ₩
--mount ₩
   type=volume, ₩
   src= os-day.{{.Task.Slot}}, ₩
   dst=/data, ₩
   volume-opt=project-id={project id}, ₩
   volume-opt=size=50
   volume-driver=ceph ₩
--replicas 5
nginx
```

```
# docker volume ls | grep os-day
ceph:latest          os-day.1
ceph:latest          os-day.2
ceph:latest          os-day.3
ceph:latest          os-day.4
ceph:latest          os-day.5
```

```
# docker service ps os-day
x8evyo901ujt          os-day.1
zpg9p601b0jr          os-day.2
am4xcvkkcad2          os-day.3
whr9moe80stl          os-day.4
6pxgzrj395wd          os-day.5
```

03 Docker Volume Plugin

Update Attach-info

기존 Attach 정보는 Nova 의 VM uuid 이기 때문에, 이를 Container ID 로 변경하여 업데이트 함

```
attachOpts := volumeactions.AttachOpts{
        MountPoint:   path,
        InstanceUUID: shortenContainerID,
        HostName:     d.Conf.Hostname,
        Mode:         "rw"}

if err := volumeactions.Attach(client, vol.ID, &attachOpts); err.Err != nil {
```

← Container ID 를 변환하여 전달

```
$ openstack volume list --all
```

```
| b3ecc2aa-dccc-4e1a-939f-584ad2b8cca8 | test-vol-04                                                  | in-use    |   20 |
                    Attached to 8d5b4390-fcf9-5928-ab4d-79ddcc77d72d on /var/lib/ceph/mount/test-vol-04
```

```
+------------------------------+-------------------------------------------------------------------------------------------+
| Field                        | Value                                                                                     |
+------------------------------+-------------------------------------------------------------------------------------------+
| attachments                  | [{u'server_id': u'8d5b4390-fcf9-5928-ab4d-79ddcc77d72d', u'attachment_id': u'f4ec769b-    |
|                              | c45f-4279-ae2d-4f2d4288a1c8', u'attached_at': u'2017-07-12T16:29:25.000000',              |
|                              | u'host_name': u'     hostname     ', u'volume_id': u'b3ecc2aa-dccc-4e1a-939f-          |
|                              | 584ad2b8cca8', u'device': u'/var/lib/ceph/mount/test-vol-04', u'id': u'b3ecc2aa-dccc-     |
|                              | 4e1a-939f-584ad2b8cca8'}]                                                                 |
```

NAVER

Summary : Compasable Infrastructure

Monolithic Cloud Platform
→ Composable and Programmable Infrastructure Building Blocks

CEPH-DOCKER-PLUGIN 오픈소스화 작업
- Docker Hub
- Github

유장선 / jangseon.ryu@navercorp.com

NAVER


NAVER
Q & A

Thank you

NAVER JangSeon Ryu