

InfraEye 2.0 설치 가이드 (oracleLinux8.6)

작성자 : 진인철

작성일 : 2024-10-07

목차

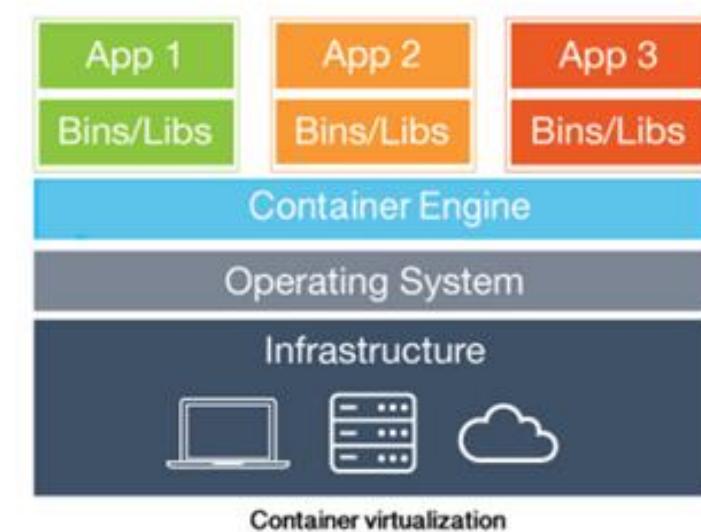
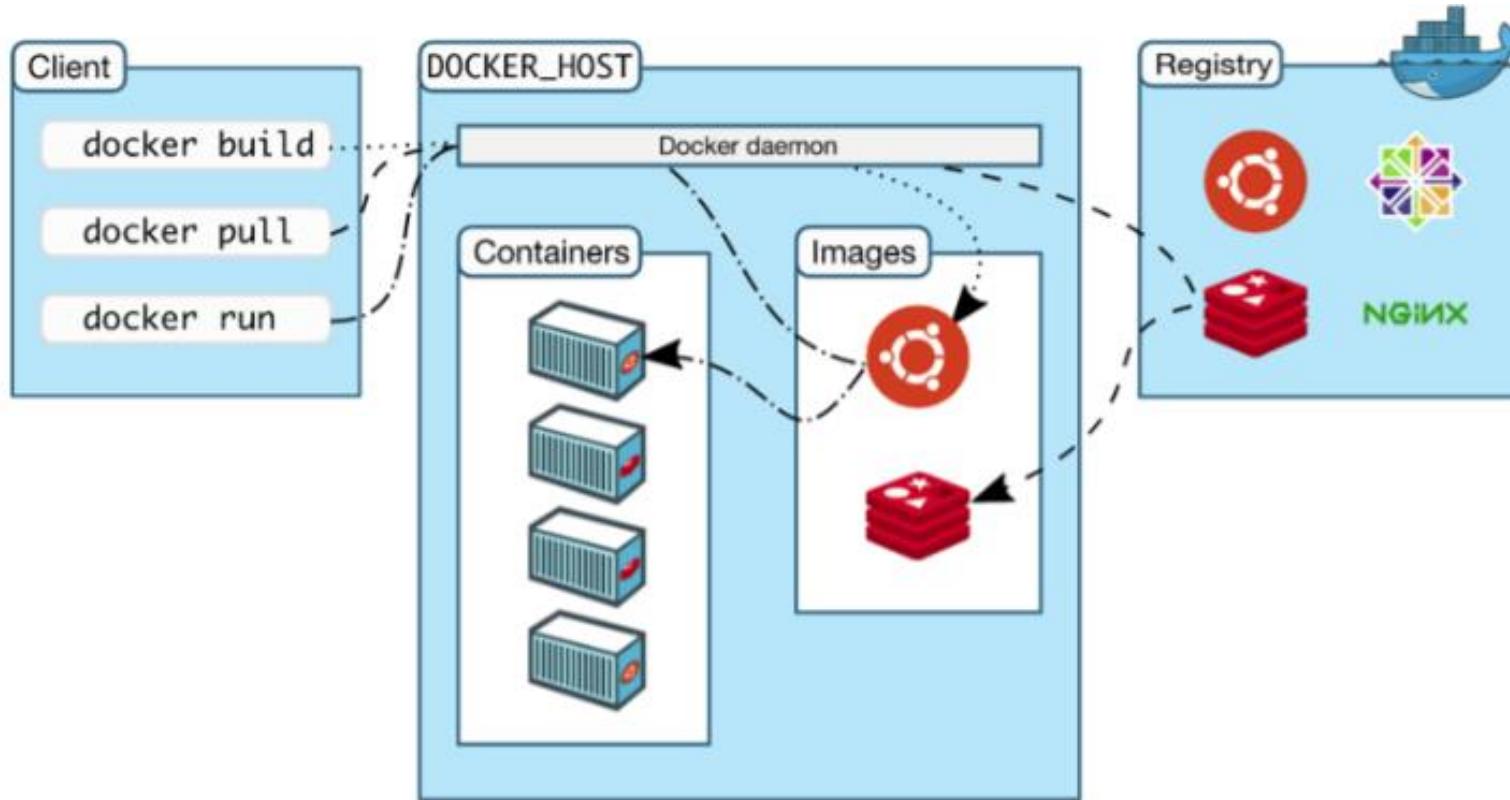
1. 개요
2. 디렉터리 구조
3. 준비 단계(docker 설치 및 계정 설정)
4. 단독형 설치
5. 분리형 설치
6. web test 컨테이너 설치
7. ai 컨테이너 설치
8. Patch
9. 참고 사항 (포트 변경, 네트워크 변경, TLS list변경, TMS 수동 설치, Container Util, system log, docker log, docker build log...)

* 모든 작업은 기본적으로 root 계정으로 진행합니다.

1. 개요 (docker 개념 및 구조)



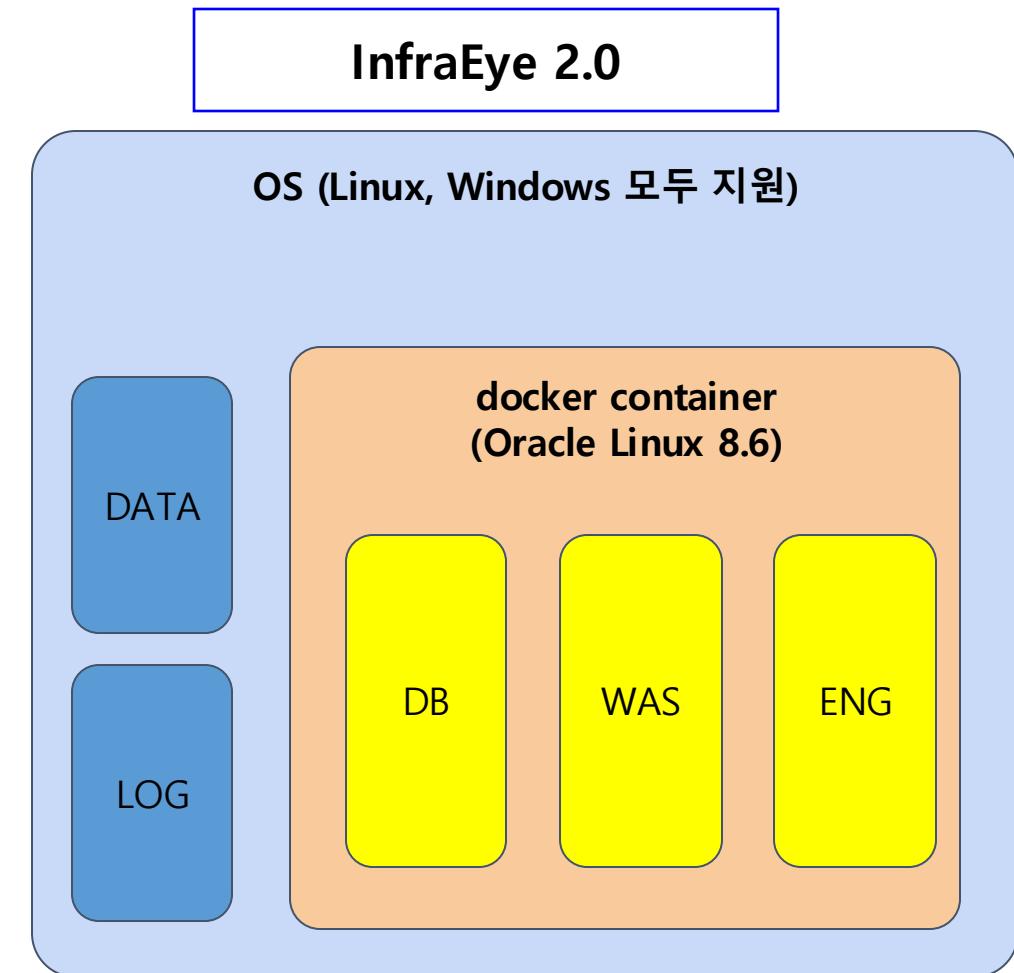
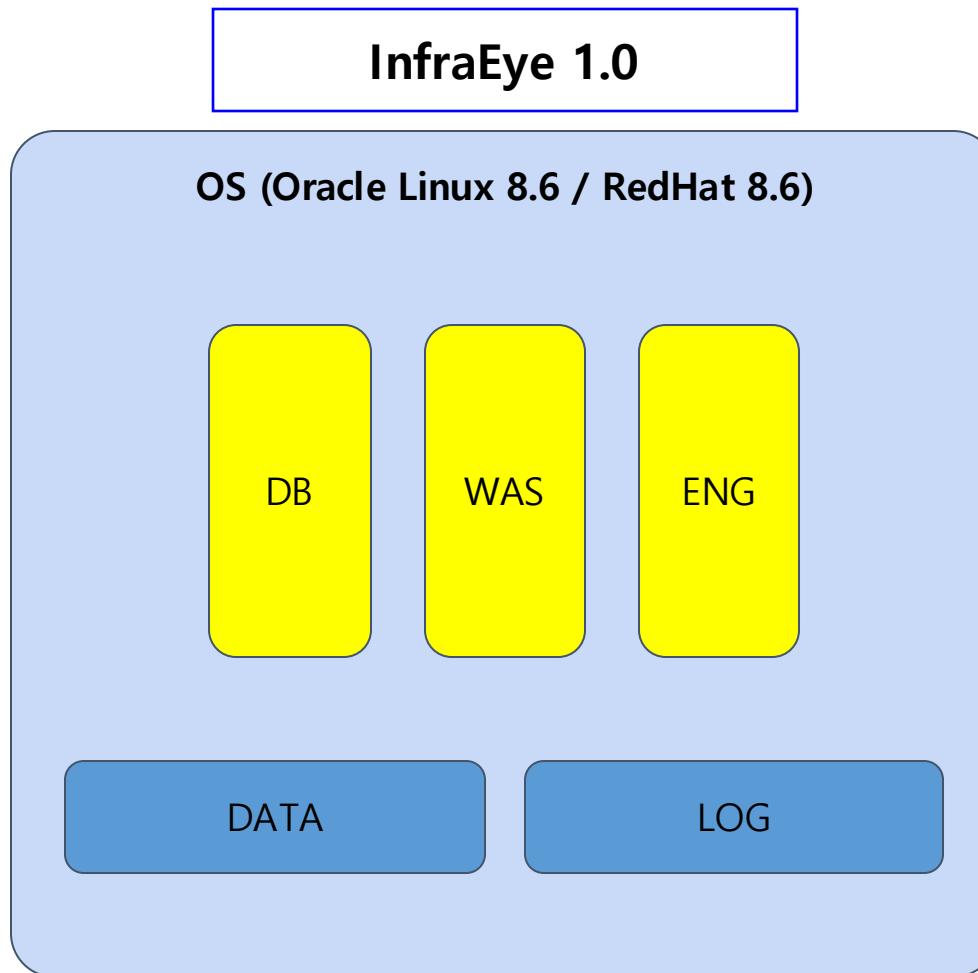
항목 (영문)	구분	설명
docker (도커)	서비스	컨테이너 기반 가상화 플랫폼으로 응용 프로그램과 그 종속성을 격리된 환경인 컨테이너로 패키징하여 실행하는 기술
docker-compose (도커 컴포즈)	관리툴	여러 개의 도커 컨테이너를 정의하고 실행하기 위한 도구로, 하나의 설정 파일로 여러 개의 컨테이너를 관리하고, 컨테이너 간의 네트워크 및 종속성을 설정하는 데 사용된다



1. 개요 (중요 변경 항목)

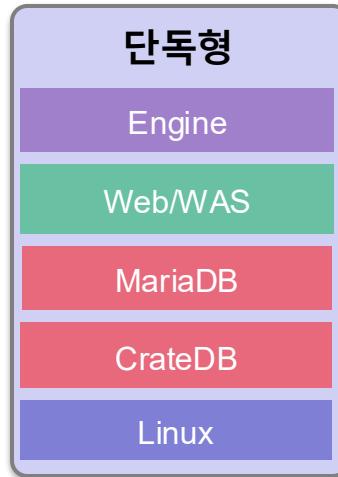
항목	Infraeye 1.0	Infraeye 2.0
OS	OracleLinux 8.6	서버 OS(OracleLinux, Ubuntu, windows...) 컨테이너 OS(OracleLinux 8.6)
docker	X	O
IP	IP 할당	컨테이너 내부 IP 사용
저장소	모든 영역(호스트 서버)	데이터영역(호스트 서버) 서비스영역(컨테이너)
보안 취약점 쉘 제공	기본 패키징 제공	호스트 서버 기본 패키징 제공 컨테이너 적용X
사용자 계정	netcruz	infraeye
패키지 경로	/opt/netcruz/nms	/opt/ infraeye /nms
systemctl	O	지원 X
REDIS	X	O
반응형 UI	X	O (개인 테마 설정)
테스트용 UI 제공	X	O
AI	X	O (9월 이후 적용 예정)

1. 개요 (설치 아키텍처)

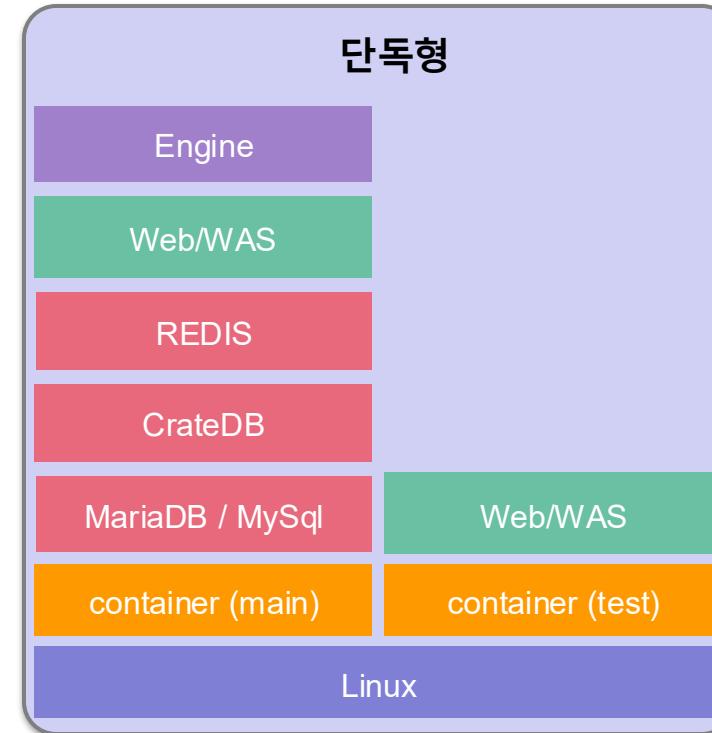


1. 개요 (서버 종류별 아키텍처)

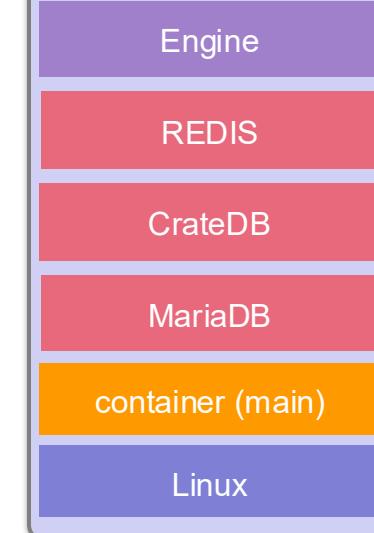
InfraEye 1.0



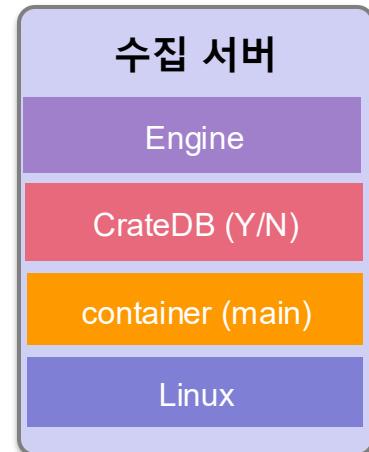
InfraEye 2.0



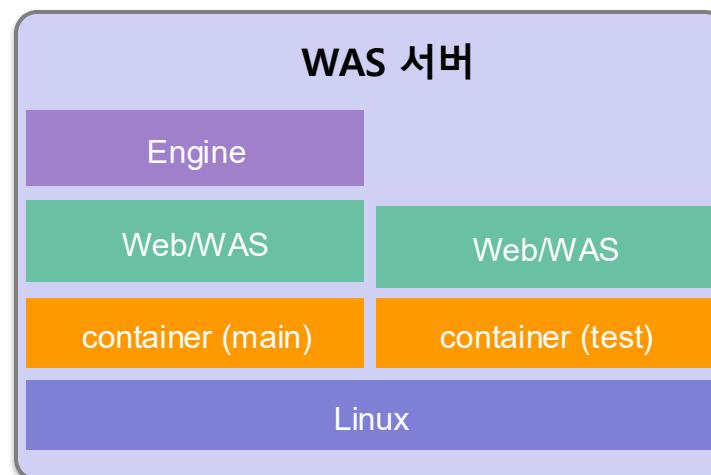
DB 서버



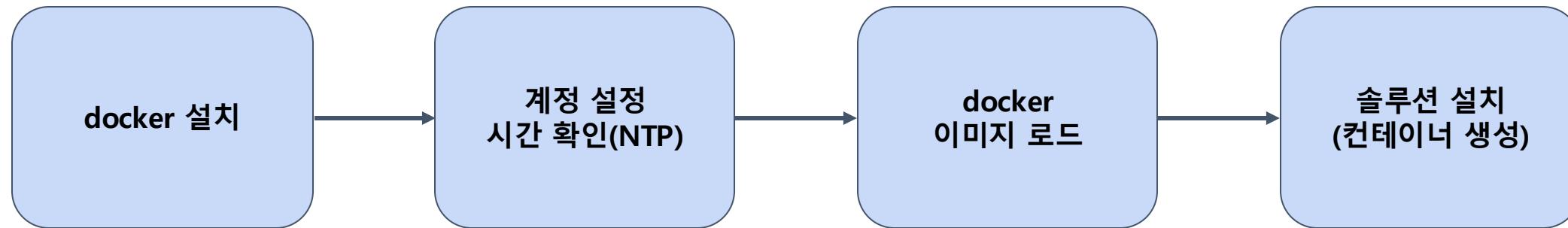
수집 서버



WAS 서버



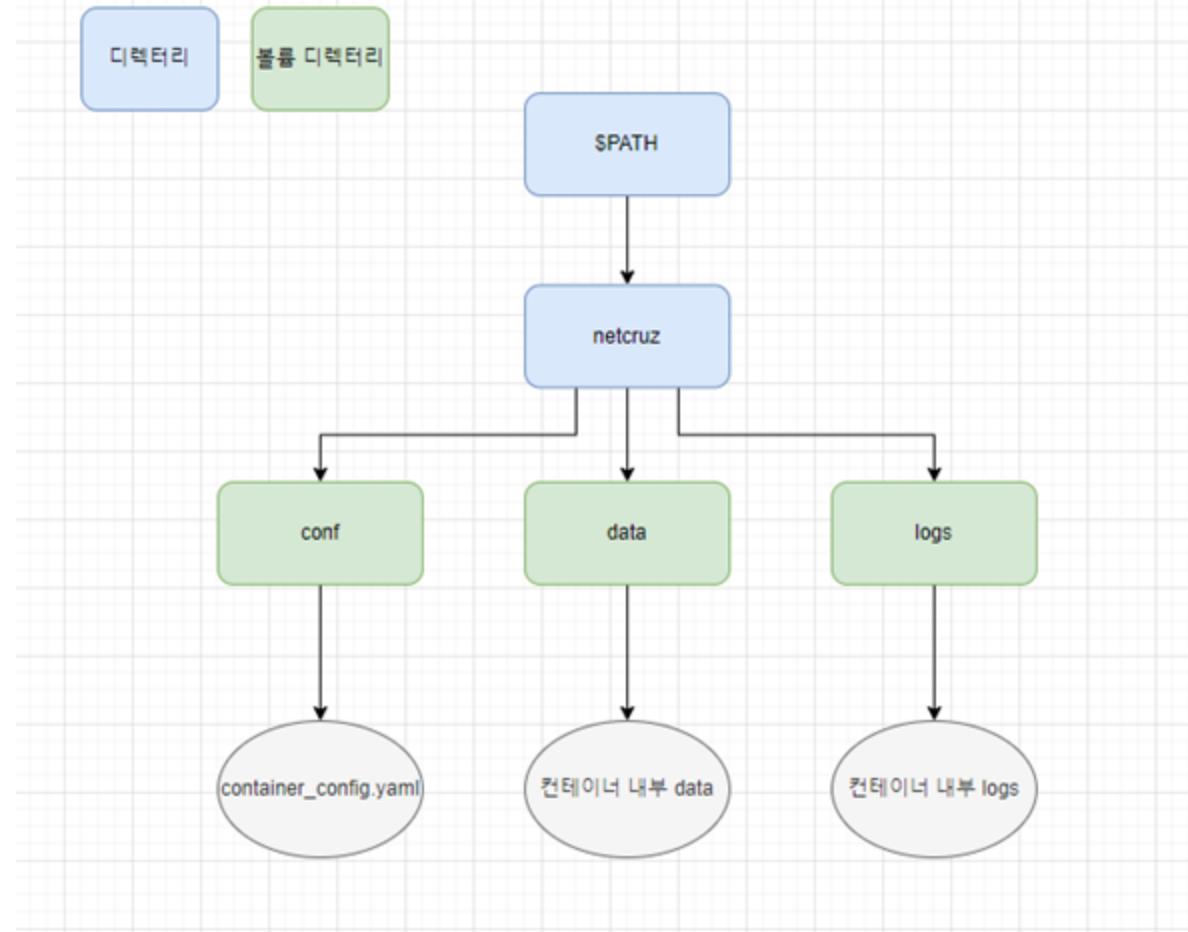
1. 개요 (패키징 설치 단계)



2. 디렉터리 구조 (Host Server)

- 패키징 설치시 호스트 서버에 아래의 디렉터리가 볼륨(공유)으로 생성됩니다.

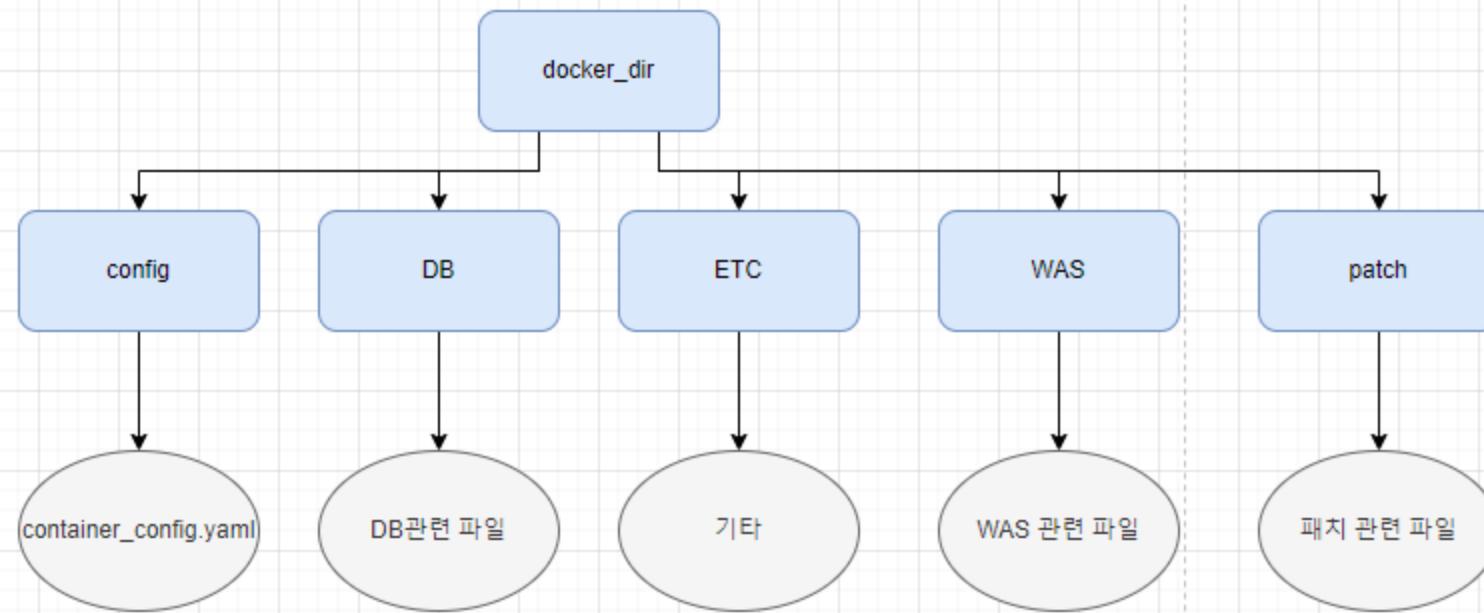
infra2 Docker 로컬서버 디렉터리 구조



2. 디렉터리 구조 (Container)

- 컨테이너 내부에 /docker_dir 디렉터리 하위에 솔루션 설치에 필요한 정보가 포함됩니다.

infra2 Docker 컨테이너 디렉터리 구조



2. 디렉터리 구조 (Package)

- 컨테이너 내부의 패키징 디렉터리 구조는 기존의 Infraeye1 과 동일합니다.
- 해당 디렉터리 구조를 확인하시려면 팀즈에 "패키징 디렉터리 구조.drawio"를 이용해주세요.

3. 준비 단계(docker 설치)

- WWW10.110.1.99WWWpackageWWWinfraEye2WWWdockerWWWpackage/InfraEye-2.0.0.241010-STD.tar.gz 을 다운로드후 압축을 해제합니다. (최신버전을 이용해주세요.)

```
[root@localhost infraeye]# ll  
합계 144900  
-rwxr-xr-x. 1 infraeye infraeye 148375754 10월 10 17:03 InfraEye-2.0.0.241010-STD.tar.  
[root@localhost infraeye]# tar -xvf InfraEye-2.0.0.241010-STD.tar.gz
```

InfraEye2 설치 패키지

- infraeye2_package/toolbox/docker_packages/ 경로로 이동 후 install_docker.sh 를 실행합니다

```
[root@localhost infraeye]# cd infraeye2_package/toolbox/docker_packages/  
[root@localhost docker_packages]# ll  
합계 165908  
-rwxr-xr-x. 1 root root 71157 9월 30 14:36 container-selinux-2.229.0-2.module+el8.10.0+90298+77a9814d.noarch.rpm  
-rwxr-xr-x. 1 root root 37048204 9월 30 14:36 containerd.io-1.6.32-3.1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 70624 9월 30 14:36 dnf-plugins-core-4.0.18-4.el8.noarch.rpm  
-rwxr-xr-x. 1 root root 14279736 9월 30 14:36 docker-buildx-plugin-0.14.0-1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 28542392 9월 30 14:36 docker-ce-26.1.3-1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 8181560 9월 30 14:36 docker-ce-cli-26.1.3-1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 5210852 9월 30 14:36 docker-ce-rootless-extras-26.1.3-1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 61431093 9월 30 14:36 docker-compose  
-rwxr-xr-x. 1 root root 14046812 9월 30 14:36 docker-compose-plugin-2.27.0-1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 70421 9월 30 14:36 fuse-overlayfs-1.13-1.module+el8.10.0+90298+77a9814d.x86_64.rpm  
-rwxr-xr-x. 1 root root 55268 9월 30 14:36 fuse3-3.3.0-19.0.1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 97276 9월 30 14:36 fuse3-libs-3.3.0-19.0.1.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 1603 9월 30 14:36 install_docker.sh  
-rwxr-xr-x. 1 root root 71192 9월 30 14:36 libcgrouper-0.41-19.el8.x86_64.rpm  
-rwxr-xr-x. 1 root root 71001 9월 30 14:36 libslirp-4.4.0-2.module+el8.10.0+90337+0d7b6e74.x86_64.rpm  
-rwxr-xr-x. 1 root root 268824 9월 30 14:36 python3-dnf-plugins-core-4.0.21-23.0.1.el8.noarch.rpm  
-rwxr-xr-x. 1 root root 56301 9월 30 14:36 slirp4netns-1.2.3-1.module+el8.10.0+90298+77a9814d.x86_64.rpm  
-rwxr-xr-x. 1 root root 199864 9월 30 14:36 unzip-6.0-45.el8_4.x86_64.rpm  
-rwxr-xr-x. 1 root root 76144 9월 30 14:36 yum-utils-4.0.21-14.1.0.1.el8.noarch.rpm  
[root@localhost docker_packages]#
```

Docker 설치 패키지로 이동

3. 준비 단계(docker 설치)

- 기존에 oraclelinux8.6 에서는 runc는 docker 설치시 충돌이 나기때문에 runc를 삭제후 docker을 설치합니다.

```
[root@localhost docker_packages]# ./install_docker.sh
runc가 설치 되어 있 습 니 다 . docker 설치를 위해 저는 삭제 가 필요 합 니 다 . 삭제 하시 겠 습 니 까 ?(Y/N)
y
종속 성 이 해 결 되 었 습 니 다 .

구조
-----
```

파일	크기	버전
runc	x86_64	2:4.0.2-6.module+el8.6.0+20656+53f7e955.x86_64
buildah	x86_64	1:1.24.2-4.module+el8.6.0+20656+53f7e955.noarch
cockpit-podman	noarch	43-1.module+el8.6.0+20656+53f7e955.x86_64
podman	x86_64	2:4.0.2-6.module+el8.6.0+20656+53f7e955.x86_64
common	x86_64	3.3.0-15.el8.x86_64
container-selinux	noarch	3.3.0-15.el8.noarch
containers-common	x86_64	3.3.0-15.el8.x86_64
criu	x86_64	3.3.0-15.el8.x86_64
fuse-overlayfs	x86_64	2:2.329.0-2.module+el8.10.0+90298+77a9814d
fuse3	x86_64	1.6.32-3.1.el8
fuse3-libs	x86_64	0.14.0-1.el8
libnet	x86_64	3:26.1.3-1.el8
libslirp	x86_64	1:26.1.3-1.el8
podman-cata tonit	x86_64	2:27.0-1.el8
shadow-utils-subid	x86_64	1.13-1.module+el8.10.0+90298+77a9814d
slirp4netns	x86_64	3.3.0-19.0.1.el8
연결 모듈		3.3.0-19.0.1.el8
삭제 16 부록		0.41-19.el8
사용 가능한 공간 : 96 M		4.4.0-2.module+el8.10.0+90337+0d7b6e74
연결 확인 실행 중		1.2.3-1.module+el8.10.0+90298+77a9814d
예상 외 예상 가능 해 습 니 다		6.0-45.el8_4

```
-----
```

Docker 설치

runc : 컨테이너 런타임
OracleLinux 기본설치항목

```
-----
```

파일	크기	버전
buildah	x86_64	1:1.24.2-4.module+el8.6.0+20656+53f7e955.noarch
containers-common	2:1.27.0.1.module+el8.6.0+20656+53f7e955.x86_64	3.3.0-15.el8.noarch
fuse-overlayfs	x86_2.1.module+el8.6.0+20656+53f7e955.x86_64	2:2.329.0-2.module+el8.10.0+90298+77a9814d
libnet	x86_64	1.13-1.module+el8.10.0+90298+77a9814d
podman-cata tonit	2:4.0.2-6.module+el8.6.0+20656+53f7e955.x86_64	4.4.0-2.module+el8.10.0+90337+0d7b6e74
shadow-utils-subid	x86_64	1.13-1.module+el8.10.0+90298+77a9814d
slirp4netns	x86_64	1.2.3-1.module+el8.10.0+90298+77a9814d
연결 모듈		6.0-45.el8_4
파일	13 부록	-----
파일	1 부록	-----
전체 크기 : 103 M		-----
부록 미 대비 할 기준 :		-----
전체 파일 목록 :		-----
전체 파일 확장 명 :		-----
전체 파일에 설정 했 습 니 다 .		-----
전체 파일 및 경로 :		-----
전체 파일에 설정 했 습 니 다 .		-----
전체 파일 및 경로 :		-----
전체 파일 :		-----
파일	: fuse3-libs-3.3.0-19.0.1.el8.x86_64	-----

```
-----
```

3. 준비 단계(docker 설치)

- Docker이 정상적으로 설치되었다면 docker을 시작하고 정상설치 여부를 확인합니다.

```
=====  
complete installed docker_packages  
  
정상적으로 설치가 완료되었습니다.  
systemctl enable docker  
systemctl daemon-reload  
systemctl restart docker  
다시 종료아래 참고해 docker를 시작해주세요.  
[root@localhost docker_packages]# docker version  
Client: Docker Engine - Community  
  Version: 26.1.3  
  API version: 1.45  
  Go version: go1.21.10  
  Git commit: b72abbb  
  Built: Thu May 16 08:34:39 2024  
  OS/Arch: linux/amd64  
  Context: default  
  
Server: Docker Engine - Community  
  Engine:  
    Version: 26.1.3  
    API version: 1.45 (minimum version 1.24)  
    Go version: go1.21.10  
    Git commit: 8e96dbl  
    Built: Thu May 16 08:33:34 2024  
    OS/Arch: linux/amd64  
    Experimental: false  
  containerd:  
    Version: 1.6.32  
    GitCommit: 8b3b7ca2e5ce38e8f31a34f35b2b68ceb8470d89  
  runc:  
    Version: 1.1.12  
    GitCommit: v1.1.12-0-g51d5e94  
  docker-init:  
    Version: 0.19.0  
    GitCommit: de40ad0  
[root@localhost docker_packages]# docker-compose --version  
Docker Compose version v2.24.6  
[root@localhost docker_packages]#
```

- systemctl enable docker -> 서버가 부팅될때 자동 시작 설정
- systemctl daemon-reload -> system 서비스 반영
- systemctl restart docker -> docker 서비스 재시작
- docker version -> docker 버전확인
- docker-compose --version -> docker compose 버전확인

Docker 실행 및 버전확인

3. 준비 단계(계정 group 설정)

- 사용할 infraeye 계정을 docker group에 등록해 docker명령어를 사용 가능하도록 변경합니다.
 - Infraeye2 부터는 netcruz 계정이 아닌 infraeye 계정이 필수로 필요합니다.
 - 일반 계정을 docker group에 등록하지 않을 경우 docker 명령어를 사용할 수 없습니다.

```
[root@localhost docker_packages]# cd ../../
[root@localhost infraeye2_package]# ll
합계 60
-rwxr-xr-x. 1 root root 15524 10월 10 17:07 Install_function.sh
-rwxr-xr-x. 1 root root 36407 10월 10 13:35 NMS2_install.sh
drwxr-xr-x. 3 root root 4096 10월 10 14:19 host_util
drwxr-xr-x. 5 root root 4096 9월 30 14:36 toolbox
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 주기
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_was(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
1
추가 할 계정을 입력 해 주세요.(default : infraeye)
infraeye
사용자 infraeye 는 그룹 docker에 추가 합니다.
[root@localhost infraeye2_package]#
```

infraeye 계정 docker 권한설정

3. 준비 단계(docker 설치)

- 10.110.1.99/package/InfraEye2/docker/package 경로에서 필요한 이미지를 다운로드 합니다. (infraeye2, ai, web_test ...)

```
sftp>
sftp> pwd
Remote working directory: /package/InfraEye2/docker/package
sftp> ls -l
drwxrwxrwx 1 admin users 120 Oct 10 16:55 infraeye2_ai_image
drwxrwxrwx 1 admin users 112 Oct 10 16:03 infraeye2_image
drwxrwxrwx 1 admin users 124 Oct 10 16:38 infraeye2_was_image
drwxrwxrwx 1 admin users 128 Oct 10 15:40 install_file
sftp>
```

sftp -oPort=20022 admin@10.110.1.99 접속 후 경로이동

- 다운로드한 이미지를 infraeye2_package 으로 옮긴 후 로드합니다.

```
[root@localhost infraeye2_package]# ll
합계 6179008
-rwxr-xr-x. 1 root root 15445 10월  7 10:57 Install_function.sh
-rwxr-xr-x. 1 root root 36234  9월 30 14:36 NMS2_install.sh
drwxr-xr-x. 3 root root 4096 10월  7 13:07 host_util
-rwxr-xr-x. 1 root root 6327238479 10월  7 13:50 infra2_img_2.0_241007.tar.gz
drwxr-xr-x. 5 root root 4096  9월 30 14:36 toolbox
[root@localhost infraeye2_package]
```

mv \${img} infraeye2_package 명령어로 패키지 경로에 이미지 push

3. 준비 단계(docker 이미지 로드)

- 컨테이너를 생성할 이미지를 로드합니다. docker 이미지 로드 선택시 실행파일과 동일한 경로에 있는 .tar 파일의 리스트가 출력됩니다. (대략 5분 소요)

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 워크한 주 가
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_was(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
[2]
docker 에 현재 로드된 이미지를 꼭 확인해 주세요.(image:ver 이 중복될 경우 충돌합니다.)
=====
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
=====
계속 진행하시겠습니까? (Y/N): [y]

docker에서 load 할 파일을 선택해주세요.
1: /root/infraeye/infraeye2_package/infra2_img_2.0_241010.tar.gz
[1]
infra2_img_2.0.tar
ff500048e221: Loading layer [=====] 236.2MB/236.2MB
b52334b6d9ef: Loading layer [=====] 19.97kB/19.97kB
9c8b53676034: Loading layer [=====] 6.144kB/6.144kB
8ee632dlf81f: Loading layer [=====] 3.072kB/3.072kB
4300ed8fe489: Loading layer [=====] 343.4MB/343.4MB
.....
dc338c4278eb: Loading layer [=====] 7.168kB/7.168kB
04f6cb4839b5: Loading layer [=====] 15.36kB/15.36kB
07573eb2cc7f: Loading layer [=====] 7.168kB/7.168kB
Loaded image: infraeye:2.0
[root@localhost infraeye2_package]#
```

docker 이미지 로드
(cmd : docker load -i \${img.tar})

3. 준비 단계(docker 이미지 로드)

- docker 이미지가 정상적으로 로드되었는지 확인합니다.

```
[root@localhost infraeye2_package]# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
infraeye        2.0      728bdbafa61a  About an hour ago  11.5GB
infraeye_ai     2.0      6c7d0722dde9  2 hours ago   13.1GB
infraeye_was    2.0      933692e57f4a  6 hours ago   4.51GB
[root@localhost infraeye2_package]#
```

docker image list 출력

4. Infraeye2 Install 단독형

- NMS2_install.sh 쉘을 이용해 패키지 설치를 진행해주세요.

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 주기
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_wasm(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
3
```

Infraeye2 설치 (단독형)

- 설치 전 주의사항을 확인 후 y를 입력해주세요.

```
!! 주의사항 !!
1. 기존에 존재하는 compose(컨테이너 생성 파일), config(컨테이너 설정 파일)은 해당 경로의 backup디렉터리로 이동됩니다.
2. 동일 포트를 포워딩하는 컨테이너가 start 상태인 경우 컨테이너가 정상적으로 실행될 수 없습니다.
3. Docker Image Name에는 대문자가 포함될 수 없습니다.
4. Database 설치 유무를 꼭 확인해주세요.
5. CrateDB의 HEAP_SIZE는 서버의 메모리를 측정하여 default를 정합니다.
6. TMS 사용 여부를 N으로 할 경우 logstash를 설치하지 않습니다.
7. 설치 시 반드시 root 계정을 이용해주세요.
8. 컨테이너 생성 시 포트포워딩을 하게 되면 docker에서 방화벽을 열게 됩니다.
9. Redis 사용 시 클러스터링 구성으로서 프트가 12개가 사용된다 cluster-bus 포트는 지정포트 + 10000이 사용된다.(master:3, slave:3, cluster-bus:6)
10. 분리형 설치의 경우 Redis 클러스터를 별도로 다시 해주어야 합니다. 가이드 문서를 참고해주세요.

주의사항을 확인 했다면 Y를 입력해주세요 :
```

4. Infraeye2 Install 단독형 (계속)

- Infraeye2 설치를 위해 필요한 정보를 입력해주세요.

```
=====
InfraEye2 설치 정보를 입력해 주세요.

=====
===== Docker Image List =====
1. infraeye:2.0
2. infraeye_ai:2.0
3. infraeye_was:2.0
=====
컨테이너 생성에 사용할 이미지를 선택해주세요 :1
Docker Image Name : infraeye:2.0
Docker Container Name (default:infraeye_2.0) : infraeye_2.0
Select Install Solution Type. (Single Server - 1, DB Server - 2, WAS Server - 3, Collector Server - 4, TMS Server - 5) : 1
TMS Use YN (default:n) : n
DB IP : 10.60.2.2
Local Server IP (default: 10.60.2.2) : 10.60.2.2
Set Container root Account Password (default: netcruz!1) : netcruz!1
Set Container infraeye Account Password (default: netcruz!1) : netcruz!1
Solution Path (default:/opt) : /opt
Select NMS Database Type (Mariadb - 1 , Mysql - 2) : 1
NMS Database Password : netcruz!#$134
NMS Databases Port(default:13306) : 13306
Create Heap Size(default:1g) : 1g
Create Databases Port(default:15432) : 15432
https Use YN (default:n) : n
Was Port(default:60000) : 60000
```

infraeye 설치 정보 입력

4. Infraeye2 Install 단독형 (계속)

- 입력한 정보를 확인후 y를 입력해 컨테이너를 생성합니다.

```
=====
Docker Image Name : infraeye:2.0
Docker Container Name : infraeye_2.0
DB IP : 10.60.2.2
LOCAL서버 IP : 10.60.2.2
Server Type : 1
root Account Password : netcruz!1
infraeye Account Password : netcruz!1
Solution Path : /opt
DB 사용 유무 (Y/N) : y
Crate 사용 유무 (Y/N) : y
Redis 사용 유무 (Y/N) : y
WAS 사용 유무 (Y/N) : y
수집기 사용 유무 (Y/N) : y
TMS 사용 유무 (Y/N) : n
NMS Database Password : netcruz!#$134
NMS Database Port : 13306
Crate Database Port : 15432
Exist Data Use(Y/N) : N
NMS Database Type : Mariadb
Crate Heap Size : lg
https Use(Y/N) : n
Was Port : 60000
=====
Start create container (Y/N) : y
[+] Running 1/1
  ✓ Container infraeye_2.0 Started
===== Container Init Setting Start =====
== 호스트 서버와 uid, gid를 통일화합니다. ==
컨테이너 내부 uid는 1002로 변경합니다.
컨테이너 내부 gid는 1003로 변경합니다.
== root, infraeye 계정에 환경 변수를 설정합니다. ==
== 기존 사용 디렉터리 Check ==
[cp_check_dir] check_dir : /docker_dir/DB/crate-5.4.7
[cp_check_dir] cp_dir : /opt/infraeye/data/crate/crate_bin
[cp_check_dir] check_dir : /docker_dir/WAS/mib_browser
```

↑ 입력 정보 확인 후 설치 진행

4. Infraeye2 Install 단독형 (계속)

- 설치 완료 후 에러로그를 확인합니다.

```
SUCCESS! REDIS:55502
SUCCESS! Redis-stat stop
Shutting down MariaDB.. SUCCESS!
== logrotate 설정 합니다.(ALL) ==
setenforce: SELinux is disabled
create logrotate ALL
tomcat not exist
access_log not exist
mod_jk not exist
ssl_request_log not exist
setenforce: SELinux is disabled
logrotate config file not exist
make logrotate config file success
== NMS IP Setting ==
== EngineConfig 설정 파일 Setting ==
== crontab setting ==
[Add_Crontab] Add to crontab = [InfraEye_check_watchdog]
[Add_Crontab] Add to crontab = [InfraEye_check_was]
[Add_Crontab] Add to crontab = [InfraEye_check_crate.sh]
[Add_Crontab] Add to crontab = [InfraEye_check_redis.sh]
[Add_Crontab] Add to crontab = [Del_Corefile.sh]

[ OK] crontab 설정이 추가되었습니다.

===== Container Init Setting Eng =====

===== Error List =====
=====
[root@localhost infraeye2_package]#
```

설치 완료

4. Infraeye2 Install 단독형 (계속)

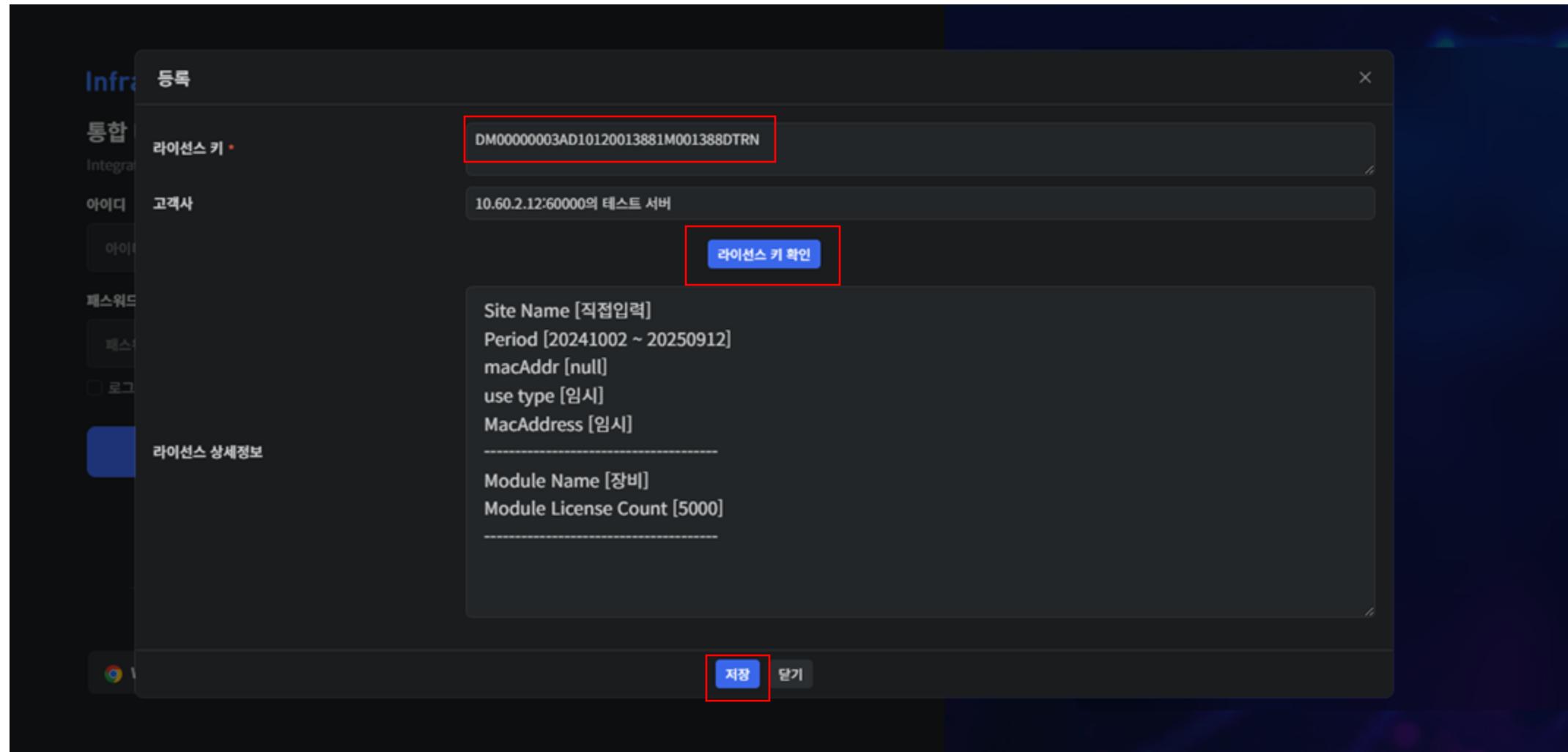
- 정상적으로 infraeye2가 올라왔는지 확인합니다.

```
[root@localhost infraeye2_package]# InfraEye db status
SUCCESS! CRATE_DB running (1933)
SUCCESS! REDIS:55501 running (581)
SUCCESS! REDIS:55502 running (599)
SUCCESS! Redis-stat running (2043)
SUCCESS! MariaDB running (2428)
[root@localhost infraeye2_package]# InfraEye was status
httpd is running
tomcat is running
[root@localhost infraeye2_package]# InfraEye eng status
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin:/usr/#### ENGINE STATUS CHECK #####
[ OK][ NC_CONF] COUNT [1]      PID [ 3310]    VER [1.4.6.240926-3-STD.R8]
[ OK][ NC_CONFIGBACK] COUNT [1]  PID [ 3553]    VER [1.4.0.240827-1-STD.R8]
[ OK][ NC_FAULT_CP] COUNT [1]  PID [ 3484]    VER [1.4.0.240930-1-STD.R8]
[ OK][ NC_L4] COUNT [1]        PID [ 3834]    VER [1.3.2.240905-1-STD.R8]
[ OK][ NC_PERF] COUNT [1]      PID [ 3404]    VER [1.4.1.240925-1-STD.R8]
[ OK][ NC_SMS] COUNT [1]      PID [ 3893]    VER [1.4.6.240925-1-STD.R8]
[ OK][ NC_SVC_CHK] COUNT [1]  PID [ 3650]    VER [1.4.7.231106-1-STD.R8]
[ OK][ NC_UI_CP] COUNT [1]    PID [ 3600]    VER [1.3.0.240621-1-STD.R8]
[ OK][ NC_UPS] COUNT [1]      PID [ 4070]    VER [1.4.6.240823-1-STD.R8]
[FAIL][ NC_X25] COUNT [0]      PID [ ]        VER [ ]
[ OK][ NC_FAULT_MS] COUNT [1]  PID [ 3158]    VER [1.4.2.240527-1-STD.R8]
[ OK][ NC_NET_SCAN] COUNT [1]  PID [ 3701]    VER [1.3.0.210914-1-STD.R8]
[FAIL][ NC_NOTI] COUNT [0]      PID [ ]        VER [ ]
[ OK][ NC_FMS] COUNT [1]      PID [ 4118]    VER [1.4.6.240627-1-STD.R8]
[ OK][ NC_PERF_LEARN] COUNT [1] PID [ 3218]    VER [1.0.1.230726-2-STD.R8]
[ OK][ NC_SYSTRAP] COUNT [1]   PID [ 3246]    VER [1.4.1.240805-1-STD.R8]
[ OK][ NC_AP] COUNT [1]       PID [ 3975]    VER [1.4.6.240927-2-STD.R8]
[FAIL][NC_EVENT_SENDER] COUNT [0] PID [ ]        VER [ ]
[FAIL][ NC_VPN] COUNT [0]      PID [ ]        VER [ ]
[FAIL][ NC_IPSLA] COUNT [0]    PID [ ]        VER [ ]
[ OK][ NC_RTT_CLI] COUNT [1]   PID [ 3791]    VER [1.4.4.220329-1-STD.R8]
[FAIL][ NC_TMS] COUNT [0]      PID [ ]        VER [ ]
[ OK][ NC_WATCHDOG] COUNT [1]  PID [ 2709]    VER [1.4.2.240923-2-STD.R8]
[FAIL][ NC_VMM] COUNT [0]      PID [ ]        VER [ ]
#####
[root@localhost infraeye2_package]#
```

DB, WAS, ENGINE 실행 여부 확인

4. Infraeye2 Install 단독형 (계속)

- 설치시 입력한 IP로 접속해 패키징이 정상적으로 설치되었는지 확인합니다.



UI 접속 여부 확인

4. Infraeye2 Install 분리형 - DB서버

- NMS2_install.sh 쉘을 이용해 패키지 설치를 진행해주세요.

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 주기
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_wasm(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
3
```

Infraeye2 설치 (DB서버)

- 설치 전 주의사항을 확인 후 y를 입력해주세요.

```
!! 주의사항 !!
1. 기존에 존재하는 compose(컨테이너 생성 파일), config(컨테이너 설정 파일)은 해당 경로의 backup디렉터리로 이동됩니다.
2. 동일 포트를 포워딩하는 컨테이너가 start 상태인 경우 컨테이너가 정상적으로 실행될 수 없습니다.
3. Docker Image Name에는 대문자가 포함될 수 없습니다.
4. Database 설치 유무를 꼭 확인해주세요.
5. CrateDB의 HEAP_SIZE는 서버의 메모리를 측정하여 default를 정합니다.
6. TMS 사용 여부를 N으로 할 경우 logstash를 설치하지 않습니다.
7. 설치 시 반드시 root 계정을 이용해주세요.
8. 컨테이너 생성 시 포트포워딩을 하게 되면 docker에서 방화벽을 열게 됩니다.
9. Redis 사용 시 클러스터링 구성으로서 프트가 12개가 사용된다 cluster-bus 포트는 지정포트 + 10000이 사용된다.(master:3, slave:3, cluster-bus:6)
10. 분리형 설치의 경우 Redis 클러스터를 별도로 다시 해주어야 합니다. 가이드 문서를 참고해주세요.

주의사항을 확인 했다면 Y를 입력해주세요 :
```

4. Infraeye2 Install 분리형 - DB서버 (계속)

- Infraeye2 DB 서버 설치를 위해 필요한 정보를 입력해주세요.

```
=====
InfraEye2 설치 정보를 입력해 주세요 .
=====
===== Docker Image List =====
1. infraeye:2.0
2. infraeye_ai:2.0
3. infraeye_was:2.0
=====
컨테이너 생성에 사용할 이미지를 선택해 주세요 :1
Docker Image Name : infraeye:2.0
Docker Container Name (default:infraeye_2.0) : infraeye_2.0
Select Install Solution Type. (Single Server - 1, DB Server - 2, WAS Server - 3, Collector Server - 4, TMS Server - 5) : 2
DB IP : 10.60.2.2
Local Server IP (default: 10.60.2.2) : 10.60.2.2
Set Container root Account Password (default: netcruz!1) : netcruz!1
Set Container infraeye Account Password (default: netcruz!1) : netcruz!1
Solution Path (default:/opt) : /opt
Select NMS Database Type (Mariadb - 1 , Mysql - 2) : 1
NMS Database Password : netcruz!#$134
NMS Databases Port(default:13306) : 13306
Create Heap Size(default:1g) : 1g
Create Databases Port(default:15432) : 15432
```

infraeye(DB) 설치 정보 입력

4. Infraeye2 Install 분리형 - DB서버 (계속)

- 입력한 정보를 확인후 y를 입력해 컨테이너를 생성합니다.

```
=====
Docker Image Name : infraeye:2.0
Docker Container Name : infraeye_2.0
DB IP : 10.60.2.2
LOCAL서버 IP : 10.60.2.2
Server Type : 2
root Account Password : netcruz!1
infraeye Account Password : netcruz!1
Solution Path : /opt
DB 사용 유무 (Y/N) : y
Crate 사용 유무 (Y/N) : y
Redis 사용 유무 (Y/N) : y
WAS 사용 유무 (Y/N) : n
수집기 사용 유무 (Y/N) : n
TMS 사용 유무 (Y/N) : n
NMS Database Password : netcruz!#$134
NMS Database Port : 13306
Crate Database Port : 15432
Exist Data Use(Y/N) : N
NMS Database Type : Mariadb
Crate Heap Size : 1g
=====
Start create container (Y/N) : y
[+] Running 1/1
✓ Container infraeye_2.0 Started
===== Container Init Setting Start =====
== 호스트 서버와 uid, gid를 동기화 합니다. ==
컨테이너 내부 uid 를 1002로 변경 합니다.
```

입력 정보 확인 후 설치 진행

4. Infraeye2 Install 분리형 - DB서버 (계속)

- 설치 완료 후 에러로그를 확인합니다.

```
#####
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
Shutting down REDISERR Errors trying to SHUTDOWN. Check logs.

SUCCESS! REDIS:55501
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
Shutting down REDISERR Errors trying to SHUTDOWN. Check logs.

SUCCESS! REDIS:55502
SUCCESS! Redis-stat stop
Shutting down MariaDB.. SUCCESS!
== logrotate == 설정 합니다.(DB) ==
setenforce: SELinux is disabled
create logrotate DB
== NMS IP Setting ==
== EngineConfig 설정 파일 Setting ==
== crontab setting ==
[Add_Crontab] Add to crontab = [InfraEye_check_watchdog]
[Add_Crontab] Add to crontab = [InfraEye_check_was]
[Add_Crontab] Add to crontab = [InfraEye_check_crate.sh]
[Add_Crontab] Add to crontab = [InfraEye_check_redis.sh]
[Add_Crontab] Add to crontab = [Del_Corefile.sh]

[ OK] crontab 설정이 추가 되었습니다.

===== Container Init Setting Eng =====

===== Error List =====
[root@localhost infraeye2_package]#
```

설치 완료

4. Infraeye2 Install 분리형 - DB서버 (계속)

- DB서버가 정상적으로 실행되었는지 확인해주세요.

```
[root@localhost infraeye2_package]# InfraEye db status
SUCCESS! CRATE_DB running (90662)
SUCCESS! REDIS:55501 running (89319)
SUCCESS! REDIS:55502 running (89335)
SUCCESS! Redis-stat running (90772)
SUCCESS! MariaDB running (91159)
[root@localhost infraeye2_package]#
[root@localhost infraeye2_package]# InfraEye was status
httpd is not running
tomcat is not running
[root@localhost infraeye2_package]# InfraEye eng status
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/l

#####
# ENGINE STATUS CHECK #####
[ OK][ NC_FAULT_MS] COUNT [1]      PID [ 91613]    VER [1.4.2.240527-1-STD.R8]
[ OK][ NC_PERF_LEARN] COUNT [1]    PID [ 91674]    VER [1.0.1.230726-2-STD.R8]
[ OK][ NC_SYSTRAP] COUNT [1]      PID [ 91731]    VER [1.4.1.240805-1-STD.R8]
[ OK][ NC_WATCHDOG] COUNT [1]     PID [ 91441]    VER [1.4.2.240923-2-STD.R8]
#####
[root@localhost infraeye2_package]#
```

DB, ENGINE 실행 여부 확인

4. Infraeye2 Install 분리형 - WAS서버

- host_util/ntp.sh 실행후 DB 서버 IP를 입력해 ntp 설정을 해주세요.

```
[root@localhost infraeye2_package]# ll
합계 6179008
-rwxr-xr-x. 1 root root    15445 10월  7 10:51 Install_function.sh
-rwxr-xr-x. 1 root root    36234 9월  30 14:36 NMS2_install.sh
drwxr-xr-x. 3 root root    4096 9월  30 14:53 host_util
-rw-r--r--. 1 root root 6327238479 10월  7 12:35 infra2_img_2.0_241007.tar.gz
drwxr-xr-x. 5 root root    4096 9월  30 14:36 toolbox
[root@localhost infraeye2_package]# host_util/ntp.sh
NTP Server IP : 10.60.2.2
success
success
[SUCC] added(10.60.2.2) in /etc/chrony.conf

stop chronyd.service ....
start chronyd.service ....
200 OK

[root@localhost infraeye2_package]#
```

DB서버와 ntp 설정

4. Infraeye2 Install 분리형 - WAS서버 (계속)

- NMS2_install.sh 쉘을 이용해 패키지 설치를 진행해주세요.

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 주기
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_was(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
3
```

Infraeye2 설치 (WAS서버)

- 설치 전 주의사항을 확인 후 y를 입력해주세요.

```
!! 주의사항 !!
1. 기존에 존재하는 compose(컨테이너 생성 파일), config(컨테이너 설정 파일)은 해당 경로의 backup디렉터리로 이동됩니다.
2. 동일 포트를 포워딩하는 컨테이너가 start 상태인 경우 컨테이너가 정상적으로 실행될 수 없습니다.
3. Docker Image Name에는 대문자가 포함될 수 없습니다.
4. Database 설치 유무를 꼭 확인해주세요.
5. CrateDB의 HEAP_SIZE는 서버의 메모리를 측정하여 default를 정합니다.
6. TMS 사용 여부를 N으로 할 경우 logstash를 설치하지 않습니다.
7. 설치 시 반드시 root 계정을 이용해주세요.
8. 컨테이너 생성 시 포트포워딩을 하게 되면 docker에서 방화벽을 열게 됩니다.
9. Redis 사용 시 클러스터링 구성으로서 프트가 12개가 사용된다 cluster-bus 포트는 지정포트 + 10000이 사용된다.(master:3, slave:3, cluster-bus:6)
10. 분리형 설치의 경우 Redis 클러스터를 별도로 다시 해주어야 합니다. 가이드문서를 참고해주세요.

주의사항을 확인 했다면 Y를 입력해주세요 :
```

4. Infraeye2 Install 분리형 - WAS서버 (계속)

- infraeye2 WAS 서버 설치를 위해 필요한 정보를 입력해주세요.

```
=====
InfraEye2 설치 정보를 입력해 주세요.
=====

===== Docker Image List =====
1. infraeye:2.0
2. infraeye_ai:2.0
3. infraeye_was:2.0
=====

컨테이너 생성에 사용할 이미지를 선택해주세요 :1
Docker Image Name : infraeye:2.0
Docker Container Name (default:infraeye_2.0) : infraeye_2.0
Select Install Solution Type. (Single Server - 1, DB Server - 2, WAS Server - 3, Collector Server - 4, TMS Server - 5) : 3
DB IP : 10.60.2.2
Local Server IP (default: 10.60.2.9) : 10.60.2.9
Set Container root Account Password (default: netcruz!1) : netcruz!1
Set Container infraeye Account Password (default: netcruz!1) : netcruz!1
Solution Path (default:/opt) : /opt
Select NMS Database Type (Mariadb - 1 , Mysql - 2) : 1
NMS Database Password : netcruz!#$134
NMS Databases Port(default:13306) : 13306
Create Databases Port(default:15432) : 15432
https Use YN (default:n) : n
Was Port(default:60000) : 60000
```

infraeye(WAS) 설치 정보 입력

4. Infraeye2 Install 분리형 - WAS서버 (계속)

- 입력한 정보를 확인 후 y를 입력해 컨테이너를 생성합니다.

```
=====
Docker Image Name : infraeye:2.0
Docker Container Name : infraeye_2.0
DB IP : 10.60.2.2
LOCAL서버 IP : 10.60.2.9
Server Type : 3
root Account Password : netcruz!1
infraeye Account Password : netcruz!1
Solution Path : /opt
DB 사용 유무 (Y/N) : n
Crate 사용 유무 (Y/N) : n
Redis 사용 유무 (Y/N) : n
WAS 사용 유무 (Y/N) : y
수집기 사용 유무 (Y/N) : n
TMS 사용 유무 (Y/N) : n
NMS Database Password : netcruz!#$134
NMS Database Port : 13306
Crate Database Port : 15432
Exist Data Use(Y/N) : N
NMS Database Type : Mariadb
https Use(Y/N) : n
Was Port : 60000
=====
Start create container (Y/N) : y
[+] Running 1/1
  ✓ Container infraeye_2.0 Started
===== Container Init Setting Start =====
== 호스트 서버와 uid, gid를 동기화합니다. ==
컨테이너 내부 uid는 1002로 변경합니다.
```

입력 정보 확인 후 설치 진행

4. Infraeye2 Install 분리형 - WAS서버 (계속)

- 설치 완료 후 에러로그를 확인합니다.

```
make logrotate config file success
== NMS IP Setting ==
== EngineConfig 설정 파일 Setting ==
== crontab setting ==
no crontab for infraeye
[Add_Crontab] Add to crontab = [InfraEye_check_watchdog]
[Add_Crontab] Add to crontab = [InfraEye_check_was]
[Add_Crontab] Add to crontab = [InfraEye_check_crate.sh]
[Add_Crontab] Add to crontab = [InfraEye_check_redis.sh]
[Add_Crontab] Add to crontab = [Del_Corefile.sh]

[ OK] crontab 설정이 추가되었습니다.

===== Container Init Setting Eng =====

===== Error List =====
[root@localhost infraeye2_package]#
```

설치 완료

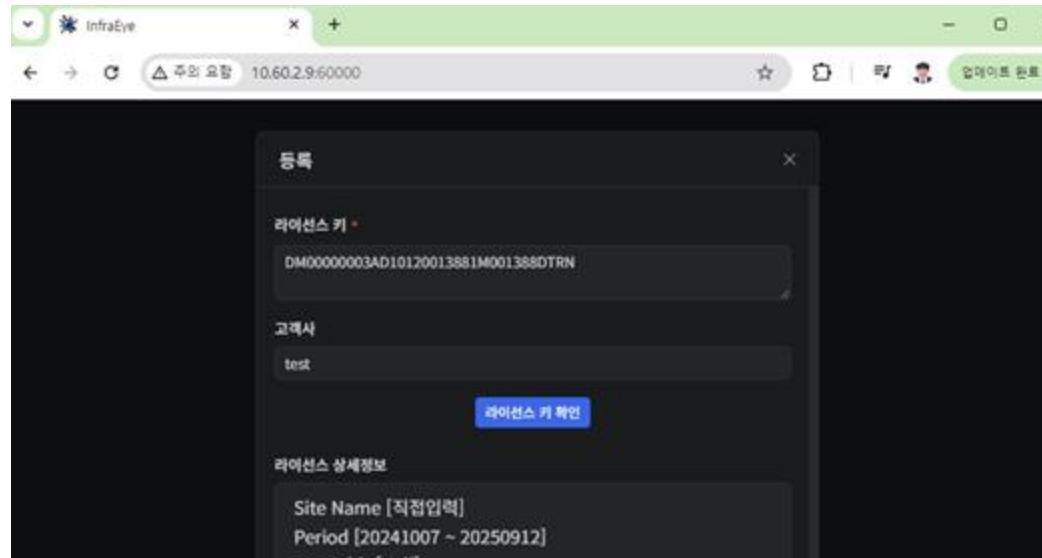
4. Infraeye2 Install 분리형 - WAS서버 (계속)

- 설치 완료 후 에러로그를 확인합니다.

```
[root@localhost infraeye2_package]# InfraEye db status
[root@localhost infraeye2_package]# InfraEye was status
httpd is running
tomcat is running
[root@localhost infraeye2_package]# InfraEye eng status
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin:/us
#####
##### ENGINE STATUS CHECK #####
[ OK][ NC_WATCHDOG] COUNT [1] PID [ 922] VER [1.4.2.240923-2-STD.R8]
#####
[root@localhost infraeye2_package]#
```

WAS, ENGINE 실행 여부 확인

- web 서버에 정상적으로 접속 되는지 확인합니다.



web 접속 가능 여부 확인

4. Infraeye2 Install 분리형 - 수집서버

- host_util/ntp.sh 실행후 DB 서버 IP를 입력해 ntp 설정을 해주세요.

```
[root@localhost infraeye2_package]# ll
합계 6179008
-rwxr-xr-x. 1 root root    15445 10월  7 10:51 Install_function.sh
-rwxr-xr-x. 1 root root    36234 9월  30 14:36 NMS2_install.sh
drwxr-xr-x. 3 root root    4096 9월  30 14:53 host_util
-rw-r--r--. 1 root root 6327238479 10월  7 12:35 infra2_img_2.0_241007.tar.gz
drwxr-xr-x. 5 root root    4096 9월  30 14:36 toolbox
[root@localhost infraeye2_package]# host_util/ntp.sh
NTP Server IP : 10.60.2.2
success
success
[SUCC] added(10.60.2.2) in /etc/chrony.conf

stop chronyd.service ....
start chronyd.service ....
200 OK

[root@localhost infraeye2_package]#
```

DB서버와 ntp 설정

4. Infraeye2 Install 분리형 - 수집서버 (계속)

- NMS2_install.sh 쉘을 이용해 패키지 설치를 진행해주세요.

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 주기
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_wasm(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
3
```

Infraeye2 설치 (ENG서버)

- 설치 전 주의사항을 확인 후 y를 입력해주세요.

```
!! 주의사항 !!
1. 기존에 존재하는 compose(컨테이너 생성 파일), config(컨테이너 설정 파일)은 해당 경로의 backup디렉터리로 이동됩니다.
2. 동일 포트를 포워딩하는 컨테이너가 start 상태인 경우 컨테이너가 정상적으로 실행될 수 없습니다.
3. Docker Image Name에는 대문자가 포함될 수 없습니다.
4. Database 설치 유무를 꼭 확인해주세요.
5. CrateDB의 HEAP_SIZE는 서버의 메모리를 측정하여 default를 정합니다.
6. TMS 사용 여부를 N으로 할 경우 logstash를 설치하지 않습니다.
7. 설치 시 반드시 root 계정을 이용해주세요.
8. 컨테이너 생성 시 포트포워딩을 하게 되면 docker에서 방화벽을 열게 됩니다.
9. Redis 사용 시 클러스터링 구성으로 프트가 12개가 사용된다 cluster-bus 포트는 지정포트 + 10000이 사용된다.(master:3, slave:3, cluster-bus:6)
10. 분리형 설치의 경우 Redis 클러스터를 별도로 다시 해주어야 합니다. 가이드문서를 참고해주세요.

주의사항을 확인 했다면 Y를 입력해주세요 :
```

4. Infraeye2 Install 분리형 - 수집서버 (계속)

- infraeye2 수집 서버 설치를 위해 필요한 정보를 입력해주세요.

```
=====
InfraEye2 설치 정보를 입력해 주세요 .
=====

===== Docker Image List =====
1. infraeye:2.0
2. infraeye_ai:2.0
3. infraeye_was:2.0
=====

컨테이너 생성에 사용할 이미지를 선택해주세요 :1
Docker Image Name : infraeye:2.0
Docker Container Name (default:infraeye_2.0) : infraeye_2.0
Select Install Solution Type. (Single Server - 1, DB Server - 2, WAS Server - 3, Collector Server - 4, TMS Server - 5) : 4
Create Use YN (default:n) : n
DB IP : 10.60.2.2
Local Server IP (default: 10.60.2.8) : 10.60.2.8
Set Container root Account Password (default: netcruz!1) : netcruz!1
Set Container infraeye Account Password (default: netcruz!1) : netcruz!1
Solution Path (default:/opt) : /opt
Select NMS Database Type (Mariadb - 1 , Mysql - 2) : 1
NMS Database Password : netcruz!#$134
NMS Databases Port(default:13306) : 13306
Create Databases Port(default:15432) : 15432
```

infraeye(ENG) 설치 정보 입력

4. Infraeye2 Install 분리형 - 수집서버 (계속)

- 입력한 정보를 확인 후 y를 입력해 컨테이너를 생성합니다.

```
=====
Docker Image Name : infraeye:2.0
Docker Container Name : infraeye_2.0
DB IP : 10.60.2.2
LOCAL서버 IP : 10.60.2.8
Server Type : 4
root Account Password : netcruz!1
infraeye Account Password : netcruz!1
Solution Path : /opt
DB 사용 유무 (Y/N) : n
Crate 사용 유무 (Y/N) : n
Redis 사용 유무 (Y/N) : n
WAS 사용 유무 (Y/N) : n
수집기 사용 유무 (Y/N) : y
TMS 사용 유무 (Y/N) : n
NMS Database Password : netcruz!#$134
NMS Database Port : 13306
Crate Database Port : 15432
Exist Data Use(Y/N) : N
NMS Database Type : Mariadb
=====

Start create container (Y/N) : y
[+] Running 1/1
  ✓ Container infraeye_2.0 Started
===== Container Init Setting Start =====
== 호스트 서버와 uid, gid 를 동기화 합니다. ==
컨테이너 내부 uid 를 1002로 변경합니다.
```

입력 정보 확인 후 설치 진행

4. Infraeye2 Install 분리형 - 수집서버 (계속)

- 설치 완료 후 에러로그를 확인합니다.

```
-- Server 정보를 세팅합니다. --
start remote add svr_info
mysql -uinfraeye -h "10.60.2.2" -P "13306" -e "CALL NMS_DB.SP_NMS_SVR_INFO_ADD('10.60.2.8',
mysql -uinfraeye -h "10.60.2.2" -P "13306" -e "CALL NMS_DB.SP_NMS_ENGINE_INFO_ADD(1, '/opt'
mysql -uinfraeye -h "10.60.2.2" -P "13306" -e "CALL NMS_DB.SP_NMS_WATCHDOG_SCH_INSTALL(1, 4
== Engine 설정을 세팅합니다. ==
10.60.2.2
13306
mysql -uinfraeye -h 10.60.2.2 -P 13306 < /docker_dir/DB/query/query_nms/conf_engine/conf_e
== NMS IP Setting ==
== EngineConfig 설정 파일 Setting ==
== crontab setting ==
no crontab for infraeye
[Add_Crontab] Add to crontab = [InfraEye_check_watchdog]
[Add_Crontab] Add to crontab = [InfraEye_check_was]
[Add_Crontab] Add to crontab = [InfraEye_check_crate.sh]
[Add_Crontab] Add to crontab = [InfraEye_check_redis.sh]
[Add_Crontab] Add to crontab = [Del_Corefile.sh]

[ OK] crontab 설정이 추가되었습니다.

===== Container Init Setting Eng =====

===== Error List =====
[root@localhost infraeye2_package]#
```

설치 완료

4. Infraeye2 Install 분리형 - 수집서버 (계속)

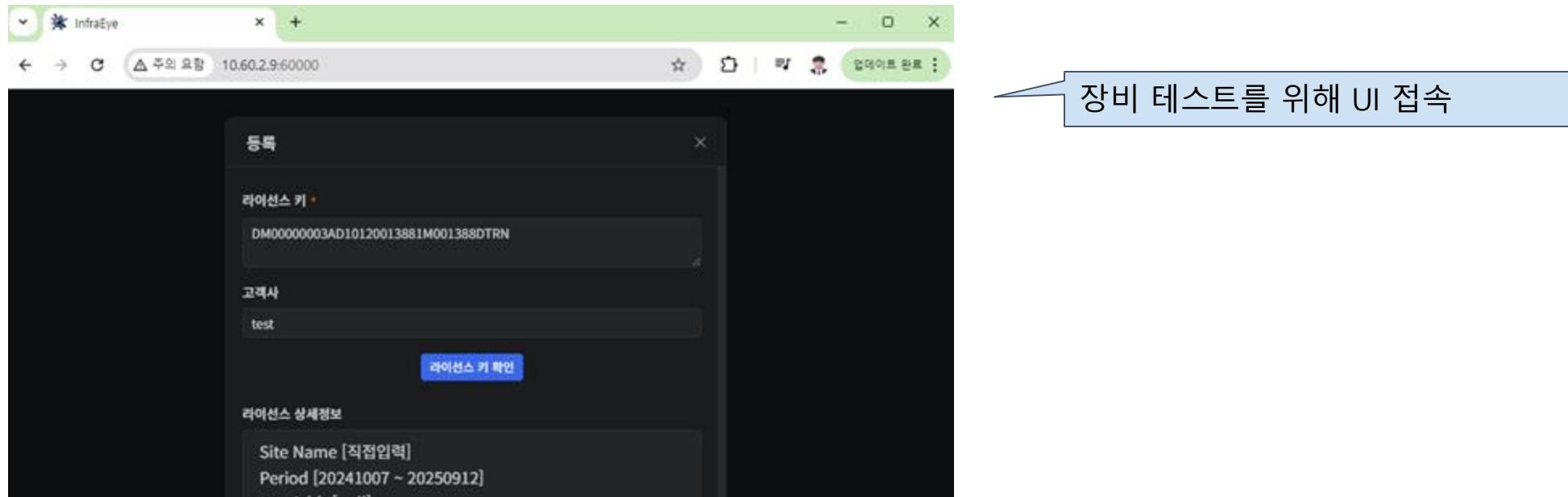
- 엔진이 정상적으로 실행 되었는지 확인합니다.

```
[root@localhost infraeye2_package]# InfraEye db status
[root@localhost infraeye2_package]# InfraEye was status
httpd is not running
tomcat is not running
[root@localhost infraeye2_package]# InfraEye eng status
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/usr/local/bin:/usr/l
#####
[ OK][      NC_CONF] COUNT [1]      PID [ 1045]      VER [1.4.6.240926-3-STD.R8]
[ OK][      NC_PERF] COUNT [1]      PID [ 1116]      VER [1.4.1.240925-1-STD.R8]
[ OK][      NC_FAULT_CP] COUNT [1]      PID [ 1190]      VER [1.4.0.240930-1-STD.R8]
[ OK][      NC_CONFIBACK] COUNT [1]      PID [ 1258]      VER [1.4.0.240827-1-STD.R8]
[ OK][      NC_UI_CP] COUNT [1]      PID [ 1305]      VER [1.3.0.240621-1-STD.R8]
[ OK][      NC_SVC_CHK] COUNT [1]      PID [ 1355]      VER [1.4.7.231106-1-STD.R8]
[ OK][      NC_NET_SCAN] COUNT [1]      PID [ 1406]      VER [1.3.0.210914-1-STD.R8]
[ OK][      NC_RTT_CLI] COUNT [1]      PID [ 1495]      VER [1.4.4.220329-1-STD.R8]
[ OK][      NC_L4] COUNT [1]      PID [ 1537]      VER [1.3.2.240905-1-STD.R8]
[ OK][      NC_SMS] COUNT [1]      PID [ 1595]      VER [1.4.6.240925-1-STD.R8]
[ OK][      NC_AP] COUNT [1]      PID [ 1677]      VER [1.4.6.240927-2-STD.R8]
[ OK][      NC_UPS] COUNT [1]      PID [ 1772]      VER [1.4.6.240823-1-STD.R8]
[ OK][      NC_FMS] COUNT [1]      PID [ 1818]      VER [1.4.6.240627-1-STD.R8]
[ OK][      NC_WATCHDOG] COUNT [1]      PID [ 935]      VER [1.4.2.240923-2-STD.R8]
#####
[root@localhost infraeye2_package]# █
```

ENGINE 실행 여부 확인

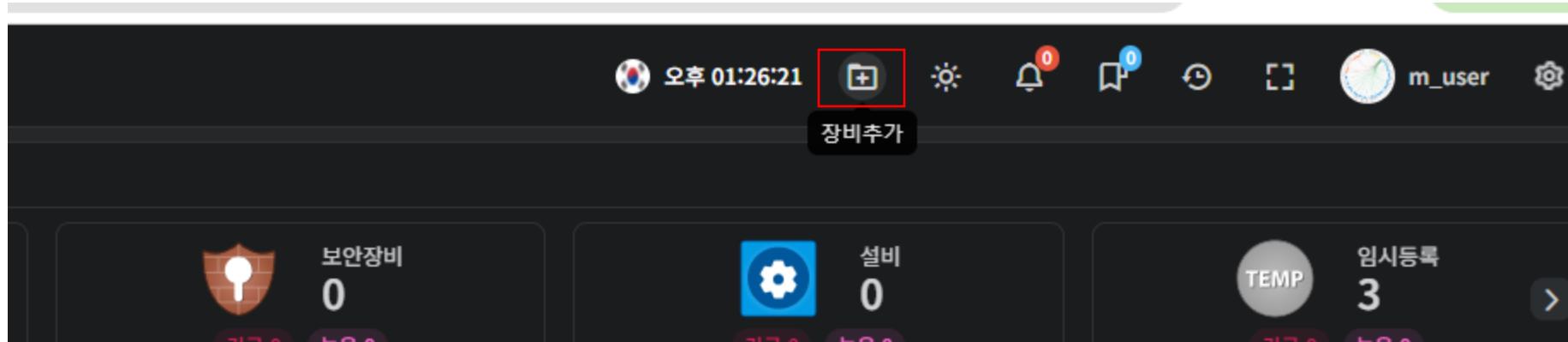
4. Infraeye2 Install 분리형 - 수집서버 (계속)

- WEB 서버에서 라이선스 등록 (id : m_user, pwd : nms12345 으로 접속)



4. Infraeye2 Install 분리형 - 수집서버 (계속)

- 장비 추가



- 정상적으로 장비가 등록 된다면 수집서버 연동 완료

The screenshot shows the '작업 결과' (Job Result) section of the Infraeye2 interface. It includes a table with columns '결과 코드' (Result Code) and '설명' (Description). The table shows one row: '성공' (Success) with the description 'PING과 SNMP 모두 성공' (Both PING and SNMP successful). Below the table is a '플터' (Filter) section with a dropdown and a search icon. At the bottom, there is a table titled '정상 등록 성공' (Successful Registration) with columns: No, 결과 (Result), 장비 IP (Device IP), PING상태 (PING Status), SNMP상태 (SNMP Status), and CPU. The table contains one row with the values: 1, 정상 등록 성공, 10.1.1.1, ○, ○, and an empty cell for CPU. A light blue callout box points to the '정상 등록 성공' row with the text '정상 등록 여부 확인'.

No	결과	장비 IP	PING상태	SNMP상태	CPU
1	정상 등록 성공	10.1.1.1	○	○	

4. Infraeye2 Install 분리형 - TMS서버

- host_util/ntp.sh 실행후 DB 서버 IP를 입력해 ntp 설정을 해주세요.

```
[root@localhost infraeye2_package]# ll
합계 6179008
-rwxr-xr-x. 1 root root    15445 10월  7 10:51 Install_function.sh
-rwxr-xr-x. 1 root root    36234 9월  30 14:36 NMS2_install.sh
drwxr-xr-x. 3 root root    4096  9월  30 14:53 host_util
-rw-r--r--. 1 root root 6327238479 10월  7 12:35 infra2_img_2.0_241007.tar.gz
drwxr-xr-x. 5 root root    4096  9월  30 14:36 toolbox
[root@localhost infraeye2_package]# host_util/ntp.sh
NTP Server IP : 10.60.2.2
success
success
[SUCC] added(10.60.2.2) in /etc/chrony.conf

stop chronyd.service ....
start chronyd.service ....
200 OK

[root@localhost infraeye2_package]#
```

DB서버와 ntp 설정

4. Infraeye2 Install 분리형 - TMS서버 (계속)

- NMS2_install.sh 쉘을 이용해 패키지 설치를 진행해주세요.

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 주기
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_wasm(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
3
```

Infraeye2 설치 (TMS서버)

- 설치 전 주의사항을 확인 후 y를 입력해주세요.

```
!! 주의사항 !!
1. 기존에 존재하는 compose(컨테이너 생성 파일), config(컨테이너 설정 파일)은 해당 경로의 backup디렉터리로 이동됩니다.
2. 동일 포트를 포워딩하는 컨테이너가 start 상태인 경우 컨테이너가 정상적으로 실행될 수 없습니다.
3. Docker Image Name에는 대문자가 포함될 수 없습니다.
4. Database 설치 유무를 꼭 확인해주세요.
5. CrateDB의 HEAP_SIZE는 서버의 메모리를 측정하여 default를 정합니다.
6. TMS 사용 여부를 N으로 할 경우 logstash를 설치하지 않습니다.
7. 설치 시 반드시 root 계정을 이용해주세요.
8. 컨테이너 생성 시 포트포워딩을 하게 되면 docker에서 방화벽을 열게 됩니다.
9. Redis 사용 시 클러스터링 구성으로 프트가 12개가 사용된다 cluster-bus 포트는 지정포트 + 10000이 사용된다.(master:3, slave:3, cluster-bus:6)
10. 분리형 설치의 경우 Redis 클러스터를 별도로 다시 해주어야 합니다. 가이드문서를 참고해주세요.

주의사항을 확인 했다면 Y를 입력해주세요 :
```

4. Infraeye2 Install 분리형 - TMS서버 (계속)

- infraeye2 TMS 서버 설치를 위해 필요한 정보를 입력해주세요.

```
=====
InfraEye2 설치 정보를 입력해 주세요 .
=====

===== Docker Image List =====
1. infraeye:2.0
2. infraeye_ai:2.0
3. infraeye_was:2.0
=====

컨테이너 생성에 사용할 이미지를 선택해주세요 :1
Docker Image Name : infraeye:2.0
Docker Container Name (default:infraeye_2.0) : infraeye_2.0
Select Install Solution Type. (Single Server - 1, DB Server - 2, WAS Server - 3, Collector Server - 4, TMS Server - 5) : 5
DB IP : 10.60.2.2
Local Server IP (default: 10.60.2.12) : 10.60.2.12
Set Container root Account Password (default: netcruz!1) : netcruz!1
Set Container infraeye Account Password (default: netcruz!1) : netcruz!1
Solution Path (default:/opt) : /opt
Select NMS Database Type (Mariadb - 1 , Mysql - 2) : 1
NMS Database Password : netcruz!#$134
NMS Databases Port(default:13306) : 13306
Create Heap Size(default:1g) : 1g
Create Databases Port(default:15432) : 15432
vflow version_5 port (default: 2055 ) : 2055
vflow version_9 port (default: 9996 ) : 9996
sflow port (default: 6343 ) : 6343
```

infraeye(TMS) 설치 정보 입력

4. Infraeye2 Install 분리형 - TMS서버 (계속)

- 입력한 정보를 확인 후 y를 입력해 컨테이너를 생성합니다.

```
=====
Docker Image Name : infraeye:2.0
Docker Container Name : infraeye_2.0
DB IP : 10.60.2.2
LOCAL서버 IP : 10.60.2.12
Server Type : 5
root Account Password : netcruz!
infraeye Account Password : netcruz!
Solution Path : /opt
DB 사용 유무 (Y/N) : n
Create 사용 유무 (Y/N) : y
Redis 사용 유무 (Y/N) : n
WAS 사용 유무 (Y/N) : n
수집기 사용 유무 (Y/N) : n
TMS 사용 유무 (Y/N) : y
NMS Database Password : netcruz!#$134
NMS Database Port : 13306
Create Database Port : 15432
Exist Data Use(Y/N) : N
NMS Database Type : Mariadb
Create Heap Size : lg
vflow Port Version 5 : 2055
vflow Port Version 9 : 9996
sflow Port : 6343
=====
Start create container (Y/N) : y
[+] Running 1/1
  ✓ Container infraeye_2.0 Started
=====
Container Init Setting Start =====
== 호스트 서버와 uid, gid 를 기화합니다. ==
컨테이너 내부 uid 를 1002로 변경합니다.
```

입력 정보 확인 후 설치 진행

4. Infraeye2 Install 분리형 - TMS서버 (계속)

- 설치 완료 후 에러로그를 확인합니다.

```
was : V3ix\6XtVnuw+xMKuDJatg==  
== Server 정보를 세팅합니다. ==  
start remote add svr_info  
mysql -uinfraeye -h "10.60.2.2" -P "13306" -e "CALL NMS_DB.SP_NMS_SVR_INFO_ADD('10.60.2.12', 5);"  
mysql -uinfraeye -h "10.60.2.2" -P "13306" -e "CALL NMS_DB.SP_NMS_ENGINE_INFO_ADD(22, '/opt', 0, 0);"  
mysql -uinfraeye -h "10.60.2.2" -P "13306" -e "CALL NMS_DB.SP_NMS_WATCHDOG_SCH_INSTALL(22, 5, '/opt');"  
== Engine 설정을 세팅합니다. ==  
10.60.2.2  
13306  
mysql -uinfraeye -h 10.60.2.2 -P 13306 < /docker_dir/DB/query/query_nms/conf_engine/conf_engine_t  
== NMS IP Setting ==  
== EngineConfig 설정 파일 Setting ==  
== Install vflow ==  
== crontab setting ==  
[Add_Crontab] Add to crontab = [InfraEye_check_watchdog]  
[Add_Crontab] Add to crontab = [InfraEye_check_was]  
[Add_Crontab] Add to crontab = [InfraEye_check_crate.sh]  
[Add_Crontab] Add to crontab = [InfraEye_check_redis.sh]  
[Add_Crontab] Add to crontab = [Del_Corefile.sh]  
  
[ OK] crontab 설정이 추가되었습니다.  
  
===== Container Init Setting Eng =====  
  
===== Error List =====  
===== [root@localhost infraeye2_package]#
```

설치 완료

4. Infraeye2 Install 분리형 - TMS서버 (계속)

- 엔진이 정상적으로 실행 되었는지 확인합니다.

```
[root@localhost infraeye2_package]# InfraEye db status
SUCCESS! CRATE_DB running (753)
[root@localhost infraeye2_package]# InfraEye was status
httpd is not running
tomcat is not running
[root@localhost infraeye2_package]# InfraEye eng status
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin:/u
#####
##### ENGINE STATUS CHECK #####
[ OK][      NC_TMS] COUNT [1]      PID [ 1178]    VER [1.4.1.240709-1-STD.R8]
[ OK][      NC_WATCHDOG] COUNT [1]      PID [ 1062]    VER [1.4.2.240923-2-STD.R8]
[ OK][      vflow] COUNT [1]      PID [ 2302]
#####
[root@localhost infraeye2_package]#
```

DB, Engine 실행 여부 확인

4. Infraeye2 Install 분리형 - TMS서버 (계속)

- web 서버 -> 관리 -> 솔루션관리 -> 서버관리에서 Flow 수집서버 등록 여부 확인

The screenshot shows the Infraeye2 web interface with a dark theme. On the left, there is a sidebar with various icons and menu items. The main area is titled '서버관리' (Server Management) and contains a table with four rows of server information. The table columns are: No, 서버명 (Server Name), 장비 IP (Device IP), 서버 구분 (Server Category), and 이중화 (Redundancy). The second row, which is highlighted with a red border, represents the 'Flow 수집 서버' (Flow Collection Server) with the following details: No 2, Server Name DB 서버 (DB Server), Device IP 10.60.2.2, Server Category DB서버 (DB Server), and Redundancy Active. A callout box with an arrow points to this row, containing the text 'TMS 서버 등록 여부 확인' (Check if TMS server is registered).

No	서버명	장비 IP	서버 구분	이중화
1	DB 서버	10.60.2.2	DB서버	Active
2	Flow 수집 서버	10.60.2.12	FLOW수집서버	-
3	NMS 수집 서버	10.60.2.8	NMS수집서버	-
4	WAS 서버	10.60.2.9	WAS서버	-

4. web test container install

- NMS2_install.sh 쉘을 이용해 Test web 설치를 진행해주세요.

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 주기
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_was(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
4
```

Test Was 설치

4. web test container install (계속)

- Test Web 설치를 위해 필요한 정보를 입력해주세요.

```
=====
InfraEye2 설치 정보를 입력해 주세요 .
=====
===== Docker Image List =====
1. infraeye:2.0
2. infraeye_ai:2.0
3. infraeye_was:2.0
=====
컨테이너 생성에 사용할 이미지를 선택해주세요 :3
Docker Image Name : infraeye_was:2.0
Docker Container Name (default:infraeye_was_2.0) : infraeye_was_2.0
NMS Container Name :infraeye_2.0 (실행 중인 컨테이너가 하나이므로 자동선택되었습니다.)
DB IP : 10.60.2.2
NMS_DB_PWD : netcruz!#$134
Test Was Port(default:60020) : 60020
```

Test Was 설정 정보 입력

```
=====
Docker Image Name : infraeye_was:2.0
Docker Container Name : infraeye_was_2.0
NMS Container Name : infraeye_2.0
DB IP : 10.60.2.2
NMS DB PWD : netcruz!#$134
TEST WAS PORT : 60020
=====
해당 정보로 컨테이너를 생성하시겠습니까? (Y/N) : y
197ac0572d7b66388a75179db6611e90e4e3afc05573alaael96a80bf932b47d
컨테이너 초기 설정 중입니다. 기다려주세요.

.
컨테이너 생성이 완료되었습니다.
소요시간 : 1sec
[root@localhost infraeye2_package]#
```

Test Was 설정 정보 확인

4. web test container install (계속)

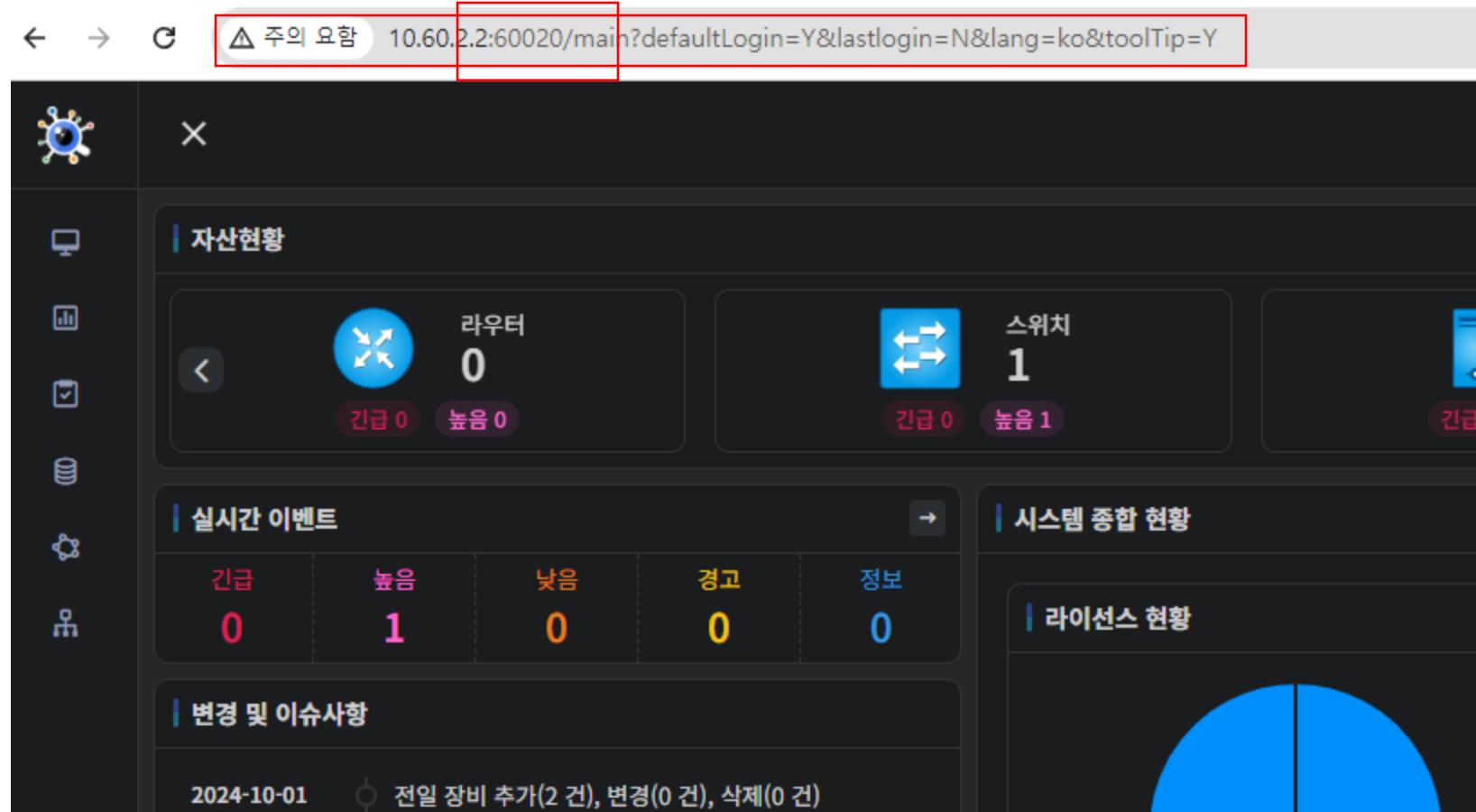
- test was컨테이너에 접속해 was실행 여부를 확인

```
[root@localhost infraeye2_package]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
197ac0572d7b        infraeye_was:2.0   "/docker_dir/ETC/was..."   27 seconds ago    Up 27 seconds      0.0.0.0:60020->60000/tcp, :::60020
                                                               NAMES
                                                               infraeye_was_2.0
a72cafd42bfc        infraeye:2.0       "/docker_dir/ETC/con..."   10 minutes ago   Up 10 minutes     0.0.0.0:162->162/udp, :::162->162/ud
:2055->2055/tcp, :::2055->2055/tcp, 0.0.0.0:4200->4200/tcp, :::4200->4200/tcp, 0.0.0.0:2055->2055/udp, :::2055->2055/udp, 0.0.0.0:6343->6343/tcp, 0.0.0.0:6343->6343/udp, :::6343->6343/tcp, :::6343->6343/udp, 0.0.0.0:9996->9996/tcp, :::9996->9996/tcp, 0.0.0.0:55500->55502->55500-55502/tcp, :::55500-55502->55500-55502/tcp, 0.0.0.0:60000-60002->60000-60002/tcp, :::60000-60002->60002/udp, 0.0.0.0:9996->9996/udp, :::9996->9996/udp, 0.0.0.0:60022->60022/tcp, :::60022->60022/tcp   infraeye_2.0
[root@localhost infraeye2_package]# docker exec -it infraeye_was_2.0 bash
[root@197ac0572d7b /]# InfraEye_was status
httpd is running
tomcat is running
[root@197ac0572d7b /]#
```

컨테이너 접속 및 was 실행 여부 확인

4. web test container install (계속)

- test was 정상 동작 여부 확인



test 컨테이너 port 접속 여부 확인

4. ai container install

- NMS2_install.sh 쉘을 이용해 AI Container 설치를 진행해주세요.

```
[root@localhost infraeye2_package]# ./NMS2_install.sh
1. 일반 사용자 docker 권한 추가
2. docker 이미지 로드
3. InfraEye 설치
4. InfraEye_was(test) 설치
5. InfraEye_AI 설치
6. 패키지 삭제
5|
```

InfraEye_AI 설치

4. ai container install (계속)

- AI 컨테이너 설치를 위해 필요한 정보를 입력해주세요.

```
=====
AI 설치 정보를 입력해 주세요.
=====
===== Docker Image List =====
1. infraeye:2.0
2. infraeye_ai:2.0
3. infraeye_was:2.0
=====
컨테이너 생성에 사용할 이미지를 선택해주세요 :2
Docker Image Name : infraeye_ai:2.0
Docker Container Name (default:infraeye_ai_2.0) : infraeye_ai_2.0
DB IP : 10.60.2.2
NMS_DB_PWD : netcruz!#$134
NMS Databases Port(default:13306) : 13306
Crate Databases Port(default:15432) : 15432

```

Infraeye_AI 설정 정보 입력

....

```
=====
Docker Image Name : infraeye_ai:2.0
Docker Container Name : infraeye_ai_2.0
DB IP : 10.60.2.2
NMS DB PWD : netcruz!#$134
NMS DB PORT : 13306
NMS CRATE DB PORT : 15432
=====
해당 정보로 컨테이너를 생성하시겠습니까? (Y/N) : y
06da88ba4554a0fd2cc54af8eee0e44122b24d9b88c105f8ccadd507034c9532
infraeye_ai_2.0
docker exec infraeye_ai_2.0 /bin/bash -c "echo 'DB_PWD: netcruz!#$134' >> /docker_conf/first_init"
[root@localhost infraeye2_package]#
```

Infraeye_AI 설정 정보 확인

4. ai container install (계속)

- AI 컨테이너 접속해 엔진 실행여부를 확인.

```
[root@localhost infraeye2_package]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
06da88ba4554        infraeye_ai:2.0   "/docker_conf/entryp..."   59 seconds ago    Up 58 seconds      NAMES
197ac0572d7b        infraeye_was:2.0  "/docker_dir/ETC/was..."  5 minutes ago     Up 5 minutes       0.0.0.0:60020->
                                                               infraeye_ai_2.0
a72cafd42bfc        infraeye:2.0      "/docker_dir/ETC/con..."  16 minutes ago    Up 16 minutes      0.0.0.0:162->1
                                                               :2055->2055/tcp, :::2055->2055/tcp, 0.0.0.0:4200->4200/tcp, :::4200->4200/tcp, 0.0.0.0:2055->2055/udp, :::
                                                               6343->6343/tcp, 0.0.0.0:6343->6343/udp, :::6343->6343/tcp, :::6343->6343/udp, 0.0.0.0:9996->9996/tcp, :::
                                                               .0.0.0:55500-55502->55500-55502/tcp, :::55500-55502->55500-55502/tcp, 0.0.0.0:60000-60002->60000-60002/tcp
                                                               0.0:9996->9996/udp, :::9996->9996/udp, 0.0.0.0:60022->60022/tcp, :::60022->60022/tcp      infraeye_2.0
[root@localhost infraeye2_package]# docker exec -it infraeye_ai_2.0 bash
[root@06da88ba4554 /]# InfraEye_AI status
Monit 5.33.0 uptime: 0m
```

Service Name	Status	Type
06da88ba4554	OK	System
python_ai	OK	Process
python_ai2	OK	Process
NC_AI_MGR	OK	Process

```
[root@06da88ba4554 /]# █
```

AI 엔진 실행 여부 확인

5. Patch (엔진 패치 - 전체 패치)

- Engine패치 /opt/infraeye/patch/eng경로에 bin 디렉터리 또는 엔진파일을 추가합니다.

```
[root@localhost eng]# pwd  
/opt/infraeye/patch/eng  
[root@localhost eng]# ls -l  
합계 4  
drwxr-xr-x. 2 root root 4096 7월  2 10:47 bin  
[root@localhost eng]#
```

- Infraeye eng patch를 입력 후 패치를 진행합니다.

```
[root@localhost patch]# InfraEye eng patch  
1. 전체 패치 ( bin 디렉터리 교체 )  
2. 부분 패치  
1  
컨테이너 내부의 bin 디렉터리를 백업 하시겠습니까? (Y/N)  
backup file path : /opt/infraeye/data/backup/ENG  
y  
bin/  
bin/NC_AP  
bin/NC_ARP  
bin/NC_CONF  
bin/NC_CONFIBACK  
bin/NC_EVENT_SENDER  
bin/NC_FAULT_CP  
bin/NC_FAULT_EX  
bin/NC_FAULT_MS  
bin/NC_FMS  
.bin/NC_TSRA
```

```
백업 완료  
1. 전체 패치 시 컨테이너의 /opt/infraeye/nms/bin 디렉터리가 로컬 서버의 /opt/infraeye/patch/eng/bin 디렉터리로 교체됩니다.  
2. 전체 엔진이 재시작 됩니다.  
패지를 전송 하시겠습니까?(Y/N)  
y  
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/usr/local/bin:/usr/local/sbin:/usr/local/  
Crontab Schedule(watchdog) 중지 하시겠습니까? (y/n)(default:y)  
Crontab Schedule(watchdog) Stop.  
[ OK] Crontab 스케줄 [watchdog] 상태가 변경되었습니다.(START->STOP)  
##### crontab list #####  
[STOP] * * * * * /opt/infraeye/nms/util/InfraEye_check_watchdog.sh >& /dev/null  
[ RUN] * * * * * /opt/infraeye/nms/util/InfraEye_check_lsync.sh >& /dev/null
```

5. Patch (엔진 패치 - 부분 패치)

- Engine패치 /opt/infraeye/patch/eng 경로에 bin 디렉터리 또는 엔진파일을 추가합니다.

```
[root@localhost eng]# ll  
합계 43944  
-rwxr-xr-x. 1 netcruz netcruz 22577712 7월  2 10:47 NC_SYSTRAP  
-rwxr-xr-x. 1 netcruz netcruz 22416248 7월  2 10:47 NC_WATCHDOG
```

- Infraeye eng patch를 입력 후 패치를 진행합니다.

```
[root@localhost patch]# InfraEye eng patch  
1. 전체 패치 ( bin 디렉터리 교체 )  
2. 부분 패치  
2  
컨테이너 내부의 bin 디렉터리를 백업 하시겠습니까? (Y/N)  
backup file path : /opt/infraeye/data/backup/ENG  
y  
bin/  
bin/NC_SVC_CHK  
bin/backup_nms.sh  
bin/NMS_COMMON.conf  
bin/capabilities.sh  
bin/NC_SYSTRAP  
bin/NC_FAULT_MS  
  
....  
백업 완료  
1. 부분 패치 시 로컬 서버의 /opt/infraeye/patch/eng 디렉터리 안에 모든 파일이 컨테이너의 /opt/infraeye/nms/bin/ 으로 복사됩니다. (이름이 동일한 파일은 전부 덮어씁니다.)  
2. 엔진 교체 시 해당 엔진을 반드시 재시작해주세요.  
패치를 진행 하시겠습니까?(Y/N)  
y  
[ NC_CONF ] 이 cap_net_raw+ep 로 적용되었습니다.  
[ NC_FAULT ] 이 cap_net_raw+ep 로 적용되었습니다.
```

5. Patch (엔진 패치 - 부분 패치)

- 엔진 덮어씌우기를 할 경우 엔진을 반드시 재시작 해야합니다.

```
[root@localhost patch]# InfraEye eng stop NC_SYSTRAP  
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/b  
  
NC_SYSTRAP is Dead!![PID : 76833]  
NC_SYSTRAP is Kill Succ  
[root@localhost patch]#
```

- 변경된 엔진이 잘 패치 되었는지 버전을 확인해주세요.

```
[root@localhost patch]# InfraEye eng status  
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin:/  
  
##### ENGINE STATUS CHECK #####  
[ OK][ NC_CONF] COUNT [2] PID [ 76878] VER [1.4.6.231107-1-STD.R8]  
[ OK][ NC_CONFBACK] COUNT [1] PID [ 77111] VER [1.4.0.240103-1-STD.R8]  
[ OK][ NC_FAULT_CP] COUNT [1] PID [ 77044] VER [1.4.0.230406-1-STD.R8]  
[ OK][ NC_L4] COUNT [1] PID [ 77388] VER [1.3.2.240109-1-STD.R8]  
[ OK][ NC_PERF] COUNT [1] PID [ 76970] VER [1.4.1.240111-1-STD.R8]  
[ OK][ NC_SNGL COUNT [1] PID [ 77446] VER [1.4.0.230705-1-STD.R8]
```

5. Patch (WEB 패치)

- Web패치 /opt/infraeye/patch/was경로에 webobjectes디렉터리, war 파일을 추가해야 합니다.

```
[root@localhost was]# pwd  
/opt/infraeye/patch/was  
[root@localhost was]# ls -l  
합계 129768  
-rw-rw-r-- 1 netcruz netcruz 132874715 6월 26 16:09 nms_solution-2.0.0.240102-1-STD.war  
drwxr-xr-x. 5 netcruz netcruz 4096 4월  5 13:56 webobjects  
[root@localhost was]#
```

- Infraeye was patch를 입력 후 패치를 진행합니다.

```
[root@localhost patch]# InfraEye was patch  
1. 전체 패치 ( war, webobjects )  
2. war 패치  
3. webobjects 패치  
1  
컨테이너 내부의 war를 백업 하시겠습니까? (Y/N)  
backup file path : /opt/infraeye/data/backup/WAS  
y  
컨테이너 내부의 webobjects 를 백업 하시겠습니까? (Y/N)  
backup file path : /opt/infraeye/data/backup/WAS  
y  
nms_solution-2.0.0.240102-1-STD.war  
war 백업 완료  
webobjects/  
webobjects/jsp/  
webobjects/jsp/js/  
webobjects/jsp/js/app_login_exec.jsp  
webobjects/jsp/js/message.jsp  
webobjects/jsp/is/setting.jsp  
....
```

5. Patch (WEB 패치)

- 백업 종료 후 이어서 패치를 진행합니다.

```
1. 패치 진행 시 패치 관련된 디렉터리가 덮어 씌워집니다.(백업 필수)
2. war파일 및 webobjects 디렉토리는 patch 경로에 꼭 하나만 있어야 합니다.
3. was가 재시작 됩니다.

[ 패치를 진행 하시겠습니까?(Y/N) ] y
Crontab Schedule(watchdog) 중지 하시겠습니까? (y/n)(default:y)
Crontab (was)스케줄을 중지합니다.
PATH : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin

[ OK] Crontab 스케줄 [was] 상태가 변경되었습니다.(START->STOP)

##### crontab list #####
[ RUN] * * * * * /opt/infraeye/nms/util/InfraEye_check_watchdog.sh >& /dev/null
[ RUN] * * * * * /opt/infraeye/nms/util/InfraEye_check_lsync.sh >& /dev/null
[ RUN] * * * * * /opt/infraeye/nms/util/InfraEye_check_chrony.sh >& /dev/null
[ STOP] #* * * * * /opt/infraeye/nms/util/InfraEye_check_was.sh >& /dev/null
[ RUN] * * * * * /opt/infraeye/nms/util/InfraEye_check_crate.sh >& /dev/null
[ RUN] * * * * * /opt/infraeye/nms/util/InfraEye_check_redis.sh >& /dev/null
[ RUN] */10 * * * * /opt/infraeye/nms/util/Del_Corefile.sh >& /dev/null
#####
```

... (war의 이름이 동일하지 않을경우 server.xml에서 변경합니다.)

```
=====
old war 파일과 new war 파일의 이름이 같지 않습니다. /opt/infraeye/was/nms-tomcat/conf/server.xml 설정 파일을 변경 하시겠습니까?(Y/N) [ nms_solution-2.0.0.230101-1-STD -> nms_solution-2.0.0.240102-1-STD ]
y
=====
>>>>>>> Apache Starting started
Tomcat started.
```

5. Patch (DB 패치)

- db패치 /opt/infraeye/patch/db 경로에 query_nms 파일을 추가해야 합니다.

```
[root@localhost db]# pwd  
/opt/infraeye/patch/db  
[root@localhost db]# ls -l  
합계 0  
drwxr-xr-x. 8 root root 96 10월  7 14:36 query_nms
```

- Infraeye was patch를 입력 후 패치를 진행합니다. (패치 후 was, engine을 재시작해주세요.)

```
[root@localhost db]# InfraEye db patch  
1. DB 패치  
2. CrateDB 패치  
1  
./  
./query_nms/  
./query_nms/INST_SQL/  
./query_nms/INST_SQL/11_DEFAULT_QRY(TRUNCATE)/  
./query_nms/INST_SQL/11_DEFAULT_QRY(TRUNCATE)/01_업무시간_입력쿼리.sql  
./query_nms/INST_SQL/11_DEFAULT_QRY(TRUNCATE)/02_라이선스_입력쿼리.sql  
./query_nms/INST_SQL/11_DEFAULT_QRY(TRUNCATE)/02_보고서_입력쿼리.sql  
./query_nms/INST_SQL/11_DEFAULT_QRY(TRUNCATE)/03_장비종류_등록정책_입력쿼리.sql  
  
....
```

```
./query_nms/version/nms_db_version_insert.sql  
./query_nms/version/version_insert.sql  
패치 파일 백업 완료  
/opt/infraeye/patch/db/query_nms 디렉터리에 SQL, PATCH_SQL, version 패치가 순차적으로 진행됩니다.  
DB패치를 진행 하시겠습니까?(Y/N)  
y
```

6. 참고 사항(포트 변경)

- docker ps 명령어로 포워딩된 포트를 확인합니다.
 - 60000:60000 ->50020:60000 으로 변경(호스트:컨테이너)

```
645df248fae0  infraeye2  "/docker_dir/ETC/sin_"  36 minutes ago  Up 36 minutes  0.0.0.0:162->162/udp, :::162->162/udp, 0.0.0.0:514->514/udp, :::514->514/udp, 0.0.0.0:443->443/tcp, :::443->443/tcp, 0.0.0.0:3306->3306/tcp, 0.0.0.0:4200->4200/tcp, :::4200->4200/tcp, 0.0.0.0:2055->2055/udp, :::2055->2055/udp, 0.0.0.0:4300->4300/tcp, :::4300->4300/tcp, 0.0.0.0:4739->4739/udp, :::4739->4739/udp, 0.0.0.0:6343->6343/udp, :::6343->6343/udp, 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp, 0.0.0.0:60000-60002->60000-60002/tcp, :::60000-60002->60000-60002/tcp, 0.0.0.0:60004-60019->60004-60019/tcp, :::60004-60019->60004-60019/tcp, 0.0.0.0:9996->9996/udp, :::9996->9996/udp, 0.0.0.0:60022->60022/tcp, :::60022->60022/tcp  infraeye2_container
```

- docker ps -a --no-trunc | grep 컨테이너ID 명령어로 전체 hash를 명을 찾습니다.

```
[root@localhost /]# sudo docker ps -a --no-trunc | grep 645df248fae0
645df248fae0535a128a9fbccffd90dd83386b832d5a2d2e766d96d71e423ef9  infraeye2  "/docker_dir/ETC/single_start.sh"  2 hours ago  Up 4 seconds  0.0.0.0:162->162/udp, :::162->162/udp, 0.0.0.0:514->514/udp, :::514->514/udp, 0.0.0.0:443->443/tcp, :::443->443/tcp, 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 0.0.0.0:4200->4200/tcp, :::4200->4200/tcp, 0.0.0.0:2055->2055/udp, :::2055->2055/udp, 0.0.0.0:4300->4300/tcp, :::4300->4300/tcp, 0.0.0.0:4739->4739/udp, :::4739->4739/udp, 0.0.0.0:6343->6343/udp, :::6343->6343/udp, 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp, 0.0.0.0:60000-60002->60000-60002/tcp, :::60000-60002->60000-60002/tcp, 0.0.0.0:60004-60019->60004-60019/tcp, :::60004-60019->60004-60019/tcp, :::60004-60019->60004-60019/tcp, 0.0.0.0:9996->9996/udp, :::9996->9996/udp, 0.0.0.0:60022->60022/tcp, :::60022->60022/tcp  infraeye2_container
```

- docker 실행중인 컨테이너 종료 및 docker의 서비스를 종료합니다.

```
[root@localhost install]# docker stop 645df248fae0
645df248fae0
[root@localhost install]# sudo systemctl stop docker.socket
[root@localhost install]# sudo systemctl stop docker.service
```

- 해당 컨테이너의 정보가 있는 디렉터리로 이동합니다.

```
[root@localhost /]# cd /var/lib/docker/containers/645df248fae0535a128a9fbccffd90dd83386b832d5a2d2e766d96d71e423ef9/
[root@localhost 645df248fae0535a128a9fbccffd90dd83386b832d5a2d2e766d96d71e423ef9]#
```

6. 참고 사항(포트 변경)

- docker ps 명령어로 포워딩된 포트를 확인합니다.

```
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# ls -l
합계 128
-rw-r-----. 1 root root 91294 6월 28 13:35 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9.json.log
drwxr-----. 2 root root 4096 6월 28 11:40 checkpoints
-rw-r-----. 1 root root 8081 6월 28 13:35 config.v2.json
-rw-r-----. 1 root root 3109 6월 28 13:35 hostconfig.json
-rw-r--r--. 1 root root 13 6월 28 13:35 hostname
-rw-r--r--. 1 root root 174 6월 28 13:35 hosts
drwxr-x---. 2 root root 4096 6월 28 11:40 mounts
-rw-r--r--. 1 root root 306 6월 28 13:35 resolv.conf
-rw-r--r--. 1 root root 71 6월 28 13:35 resolv.conf.hash
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# vim hostconfig.json
```

- hostconfig.json 파일에 접속해 포트를 변경합니다.

```
["HostIp": "", "HostPort": "4300"}, "443/tcp": [{"HostIp": "", "HostPort": "60000"}], "60000/tcp": [{"HostIp": "", "HostPort": "60000"}]}, "60001, {"HostIp": "", "HostPort": "60005"}], "60006/tcp": [{"HostIp": "", "HostPort": "60006"}], "60006/tcp": [{"HostIp": "", "HostPort": "60010"}], "60010/tcp": [{"HostIp": "", "HostPort": "60010"}]
```

->

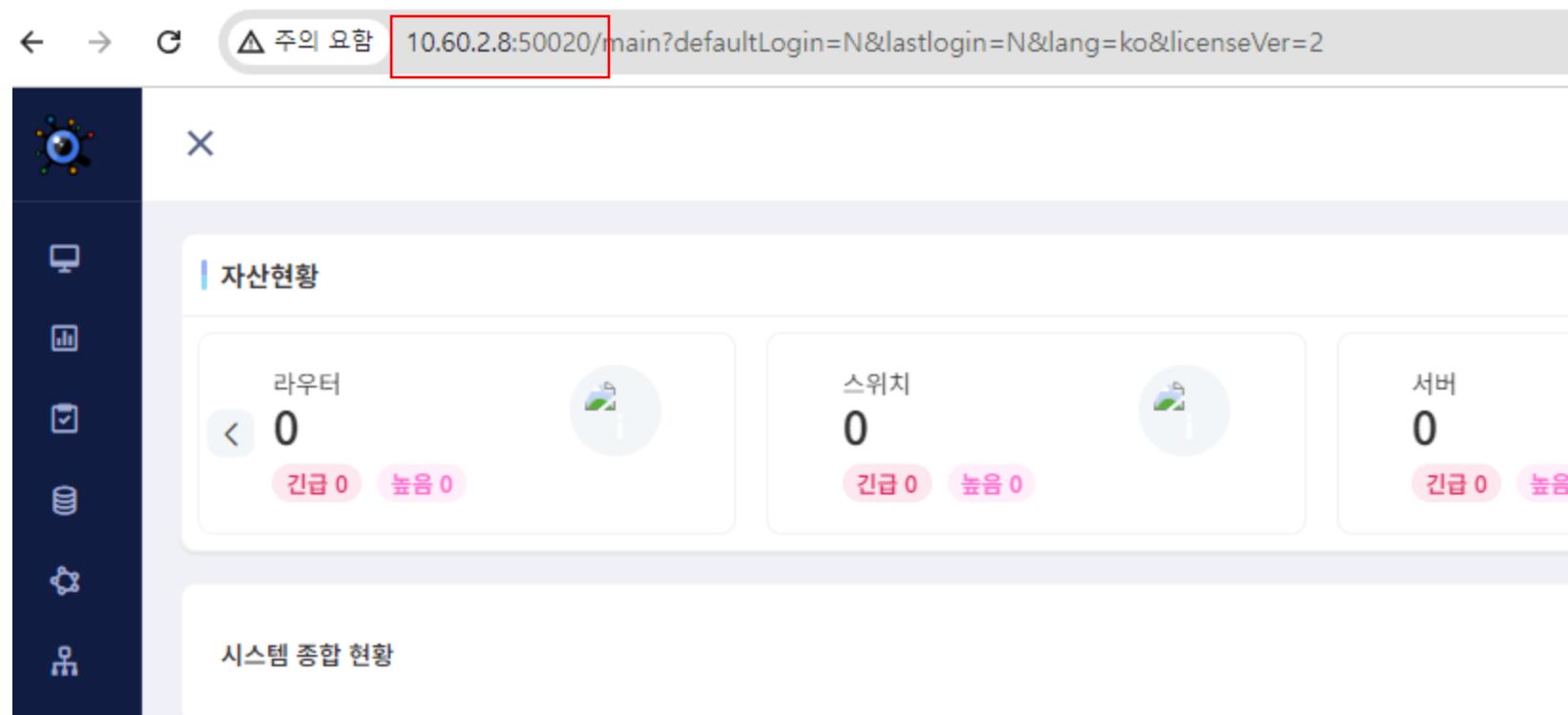
```
["HostIp": "", "HostPort": "4300"}, "443/tcp": [{"HostIp": "", "HostPort": "50020"}], "60000/tcp": [{"HostIp": "", "HostPort": "50020"}], "60001, {"HostIp": "", "HostPort": "60005"}], "60006/tcp": [{"HostIp": "", "HostPort": "60006"}], "60006/tcp": [{"HostIp": "", "HostPort": "60010"}], "60010/tcp": [{"HostIp": "", "HostPort": "60010"}]
```

- docker 서비스를 실행하고 컨테이너를 실행해 포트가 변경된걸 확인합니다. (방화벽 해제 X)

```
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# sudo systemctl start docker.service
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# sudo systemctl start docker.socket
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# docker start infraeye2_container
infraeye2_container
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# docker ps |grep 50020
645df248fae0 infraeye2 "/docker_dir/ETC/sin_" 6 hours ago Up 7 seconds 0.0.0.0:162->162/udp, ::162->162/udp, 0.0.0.0:514->514/udp, ::514->514/udp, 0.0.0.0:443->3306/tcp, ::3306->3306/tcp, 0.0.0.0:4200->4200/tcp, ::4200->4200/tcp, 0.0.0.0:2055->2055/udp, ::2055->2055/udp, 0.0.0.0:4300->4300/tcp, ::4300->4300/tcp, 0.0.0.0:473->6343/udp, ::6343->6343/udp, 0.0.0.0:5432->5432/tcp, ::5432->5432/tcp, 0.0.0.0:60001-60002->60001-60002/tcp, ::60001-60002->60001-60002/tcp, 0.0.0.0:60004-60019->619/tcp, 0.0.0.0:9996->9996/udp, ::9996->9996/udp, 0.0.0.0:60022->60022/tcp, ::60022->60022/tcp, 0.0.0.0:50020->60000/tcp infraeye2_container
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]#
```

6. 참고 사항(포트 변경)

- 변경된 포트로 정상접속 확인



6. 참고 사항(포트 추가)

- 포트 변경과 동일하게 컨테이너와 docker 서비스를 종료후 변경할 컨테이너 config.v2.json, hostconfig.json 파일을 찾습니다.

```
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# pwd  
/var/lib/docker/containers/645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9  
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# ll  
합계 184  
-rw-r-----. 1 root root 148767 6월 28 17:51 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9.json.log  
drwxr-----. 2 root root 4096 6월 28 11:40 checkpoints  
-rw-----. 1 root root 8081 6월 28 17:51 config.v2.json  
-rw-----. 1 root root 3109 6월 28 17:51 hostconfig.json  
-rw-r--r--. 1 root root 13 6월 28 17:51 hostname  
-rw-r--r--. 1 root root 174 6월 28 17:51 hosts  
drwxr-x---. 2 root root 4096 6월 28 11:40 mounts  
-rw-r--r--. 1 root root 306 6월 28 17:51 resolv.conf  
-rw-r--r--. 1 root root 71 6월 28 17:51 resolv.conf.hash  
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]#
```

- config.v2.json 파일에 해당 문구를 추가합니다.

- ,"\$hostPort}/tcp":{}

```
"AttachStdin":false  
{},"50030/tcp":{},  
}),"60015/tcp":{},
```

- hostconfig.json 파일에 해당 문구를 추가합니다.

- ,"\$ContainerPort}/tcp": [{"HostIp":"","HostPort":"HostPort"}]

```
,"HostIp":"","HostPort":443}],443,443,443,443  
],"50030/tcp": [{"HostIp":"","HostPort":50030}],  
}],"60005/tcp": [{"HostIp":
```

6. 참고 사항(포트 추가)

- docker 서비스와 컨테이너를 실행 후 추가된 포트를 확인합니다.

```
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# systemctl start docker.service
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# systemctl start docker.socket
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# docker start infraeye2_container
infraeye2_container
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]# docker ps |grep 50030
645df248fae0    infraeye2    "/docker_dir/ETC/sin..."    3 days ago   Up 4 seconds  0.0.0.0:162->162/udp, :::162->162/udp, 0.0.0.0:513306/tcp, :::3306->3306/tcp, 0.0.0.0:4200->4200/tcp, :::4200->4200/tcp, 0.0.0.0:2055->2055/udp, :::2055->2055/udp, 0.0.0.0:4300->43->6343/udp, :::6343->6343/udp, 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp, 0.0.0.0:50030->50030/tcp, :::50030->50030/tcp, 0.0.0.0:19->60004-60019/tcp, :::60004-60019->60004-60019/tcp, 0.0.0.0:9996->9996/udp, :::9996->9996/udp, 0.0.0.0:60022->60022/tcp, :::60022->60022/tcp
[root@localhost 645df248fae0535a128a9fbccfdd90dd83386b832d5a2d2e766d96d71e423ef9]#
```

6. 참고 사항(네트워크 변경)

- 변경할 컨테이너의 이미지명과 사용중인 NetworkID를 확인합니다.
 - docker inspect \${컨테이너ID} |grep Network

```
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
645df248fae0        infraeye2          "/docker_dir/ETC/sin..."   3 days ago        Up 25 minutes      0.0.0.0:162->162/udp
>3306/tcp, :::3306->3306/tcp, 0.0.0.0:4200->4200/tcp, :::4200->4200/tcp, 0.0.0.0:2055->2055/udp, :::2055/udp, :::343->6343/udp, :::6343->6343/udp, 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp, 0.0.0.0:50030->50030/tcp
019->60004-60019/tcp, :::60004-60019->60004-60019/tcp, 0.0.0.0:9996->9996/udp, :::9996->9996/udp, 0.0.0.0:9996->9996/udp
[r
[root@localhost ~]# docker inspect 645df248fae0 |grep Network
    "NetworkMode": "install_default",
    "NetworkSettings": {
        "Networks": {
            "NetworkID": "47a162915dc48fca3c4a743bb0b44c982a9d35614380618ba787322e93ae3f72",
            [root@localhost ~]#
```

- 변경할 네트워크를 생성합니다.
 - docker network create \${네트워크명}

```
[root@localhost ~]# docker network create test_net
442942c1504399b3056e08f1d6097b65b59126706f2fad486d3676414b363505
[root@localhost ~]#
```

- 기존의 네트워크를 disconnect 합니다.
 - docker network disconnect \${네트워크ID} \${컨테이너ID}

```
[root@localhost ~]# docker network disconnect 47a162915dc48fca3c4a743bb0b44c982a9d35614380618ba787322e93ae3f72 645df248fae0
[root@localhost ~]#
```

6. 참고 사항(네트워크 변경)

- 생성한 네트워크를 컨테이너에 연결 후 NetworkID를 확인합니다.

```
[root@localhost /]# docker network connect 442942c1504399b3056e08f1d6097b65b59126706f2fad486d3676414b363505 645df248fae0
[root@localhost /]# docker inspect 645df248fae0 |grep Network
    "NetworkMode": "install_default",
    "NetworkSettings": {
        "Networks": {
            "NetworkID": "442942c1504399b3056e08f1d6097b65b59126706f2fad486d3676414b363505",
[root@localhost /]#
```

- 정상적으로 포트가 포워딩 되어있는지 확인해 네트워크 연결여부를 확인합니다.

```
[root@localhost /]#
[root@localhost /]# docker ps
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS      PORTS
645df248fae0   infraeye2   "/docker_dir/ETC/sin..."   3 days ago   Up About an hour   0.0.0.0:162->162/udp,
06->3306/tcp, ::3306->3306/tcp, 0.0.0.0:4200->4200/tcp, ::4200->4200/tcp, 0.0.0.0:2055->2055/udp, ::20
0:6343->6343/udp, ::6343->6343/udp, 0.0.0.0:5432->5432/tcp, ::5432->5432/tcp, 0.0.0.0:50030->50030/tcp,
-60019->60004-60019/tcp, ::60004-60019->60004-60019/tcp, 0.0.0.0:9996->9996/udp, ::9996->9996/udp, 0.0.
iner
[root@localhost /]#
```

6. 참고 사항(TLS List 변경)

- tls 설정은 https 를 사용할 경우에만 해당됩니다. infraeye -> infraeye2로 변경되면서 tls v3만 지원하도록 바뀌었기 때문에 필요할경우 변경해주어야 합니다.
- /opt/infraeye/webserver/conf/extra/httpd-vhosts.conf 파일을 열어서 수정합니다.

```
[infraeye@57691d5f91c7 ~]$ vim /opt/infraeye/webserver/conf/extra/httpd-vhosts.conf
```

- SSLProtocol, SSLCipherSuite 두 옵션을 주석 처리하고 아래 주석을 풀어 변경합니다.
○ 변경 후 was를 재시작 해주어야 합니다.

```
SSLEngine on
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite HIGH:!aNULL:!MD5:!3DES:!CAMELLIA:!IDEA:!SEED:!RC4:!DES:!RC3:!AES:!3DES:!CAMELLIA256:!SHA256:!AES256:!SHA:!3DES:!CAMELLIA128:!AES128:!SHA128:!SHA256:!AES128:!SHA256:!AES256:!SHA256:!DHE:!ECDH:!ECDSA:!RSA:!PSK
#SSLProtocol all -SSLv2 -SSLv3
#SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
SSLCertificateFile /opt/infraeye/webserver/conf/cert/server.crt
SSLCertificateKeyFile /opt/infraeye/webserver/conf/cert/server.key
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
CustomLog /opt/infraeye/logs/webserver/ssl_request_log "%t %h %{SSL_PROTOCOL}i %l %u %r %s %b"
</VirtualHost>
"/opt/infraeye/webserver/conf/extra/httpd-vhosts.conf" 66L, 3731C
```

```
SSLEngine on
# SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
# SSLCipherSuite HIGH:!aNULL:!MD5:!3DES:!CAMELLIA:!IDEA:!SEED:!RC4:!DES:!RC3:!AES:!3DES:!CAMELLIA256:!SHA256:!AES256:!SHA:!3DES:!CAMELLIA128:!AES128:!SHA128:!SHA256:!AES128:!SHA256:!AES256:!SHA256:!DHE:!ECDH:!ECDSA:!RSA:!PSK
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
SSLCertificateFile /opt/infraeye/webserver/conf/cert/server.crt
SSLCertificateKeyFile /opt/infraeye/webserver/conf/cert/server.key
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
CustomLog /opt/infraeye/logs/webserver/ssl_request_log "%t %h %{SSL_PROTOCOL}i %l %u %r %s %b"
</VirtualHost>
-- 끄위 넣기 --
```

6. 참고 사항(Container Util)

- /docker_dir/ETC/container_util.sh 쉘을 실행합니다.
- #설정파일 DB PWD, Port 변경

```
[root@57691d5f91c7 ETC]# ./container_util.sh
1. DB, PORT 패스 워드 변경 (NMS_COMMON.conf, conf/context.xml, EngineConfig.sh)
2. IP, PORT 변경 (odbc.ini, context.xml, EngineConfig.sh)
3. CrateDB 계정 생성
4. was 파일명 변경 (server.xml)
5. vflow 설치
6. sflow 설치
7. infraeye 계정 uid, gid변경
exec type : 1
DB 패스워드 :netcruz!#$134
NMS_COMMON.conf file path (default:/opt/infraeye/nms/bin) :/opt/infraeye/nms/bin
context file path (default:/opt/infraeye/was/nms-tomcat/conf) :/opt/infraeye/was/nms-tomcat/
Properties file path (default:/opt/infraeye/nms/properties) :/opt/infraeye/nms/properties
DB Port (default:13306) :13306
CrateDB Install YN (default:Y) :y
new : l\q7r223aToyQozL9BF4k\DWfMLn88+jgYPZZi3eHw3Jk2Jdz3\KDIBU1l9Yx201wKE0u07bTi3jmelTFK5
redis_new : 4LkzIR2dZssiCQxkK109z2FKBWs\6eFFA\auFGMaGiW7UnMPDoyOpY4Yd\yl1UIDME\VH3FwoGss
redis_ai_new : 4LkzIR2dZssiCQxkK109z2FKBWs\6eFFA\auFGMaGiW7UnMPDoyOpY4Yd\yl1UIDf1WvIj1KIK
tms_new : J4c89AyjlY7qNu\eqwLW3KiztzjwnD7SbFkPqc+y2WZKpR2y1hhkkBzBB\gidrjtTjsY6NbVQYJ4P2y
y
Y
was : V3ix\6XtVnuw+xMKuDjatg==
완료 되었습니다.
[root@57691d5f91c7 ETC]#
```

6. 참고 사항(Container Util)

- /docker_dir/ETC/container_util.sh 쉘을 실행합니다.
- #설정파일 IP,Port변경

```
[root@57691d5f91c7 ETC]# ./container_util.sh
1. DB, PORT 패스워드 변경 (NMS_COMMON.conf, conf/context.xml, EngineConfig.sh)
2. IP, PORT 변경 (odbc.ini, context.xml, EngineConfig.sh)
3. CrateDB 계정 생성
4. was 파일명 변경 (server.xml)
5. vflow 설치
6. sflow 설치
7. infraeye 계정 uid, gid변경
exec type : 2
New Ip :10.60.2.2
odbc file path (default:/etc):/etc
context file path (default:/opt/infraeye/was/nms-tomcat/conf) :/opt/infraeye/was
engineconfig file path (default:/opt/infraeye/nms/util) :/opt/infraeye/nms/util
DB Port (default:13306) :13306
CrateDB Port (default:15432) :15432
CrateDB Install YN (default:Y) :y
완료 되었습니다.
[root@57691d5f91c7 ETC]#
```

6. 참고 사항(Container Util)

- /docker_dir/ETC/container_util.sh 쉘을 실행합니다.
- #CreateDB 계정생성

```
[root@57691d5f91c7 ETC]# ./container_util.sh
1. DB, PORT 패스워드 변경 (NMS_COMMON.conf, conf/context.xml, EngineConfig.sh)
2. IP, PORT 변경 (odbc.ini, context.xml, EngineConfig.sh)
3. CrateDB 계정 생성
4. was 파일명 변경 (server.xml)
5. vflow 설치
6. sflow 설치
7. infraeye 계정 uid, gid변경
exec type : 3
ID : test_user
PWD : netcruz!#$134
CONNECT OK
CREATE OK, 1 row affected (0.158 sec)
GRANT OK, 4 rows affected (0.029 sec)
완료 되었습니다.
[root@57691d5f91c7 ETC]# ]
```

6. 참고 사항(Container Util)

- /docker_dir/ETC/container_util.sh 쉘을 실행합니다.
- #war 파일명 변경 (webapps 디렉터리에 was 파일명으로 톰캣 설정파일을 변경합니다.)

```
[root@57691d5f91c7 ETC]# ./container_util.sh
1. DB, PORT 패스워드 변경 (NMS_COMMON.conf, conf/context.xml, EngineConfig.sh)
2. IP, PORT 변경 (odbc.ini, context.xml, EngineConfig.sh)
3. CrateDB 계정 생성
4. was 파일명 변경 (server.xml)
5. vflow 설치
6. sflow 설치
7. infraeye 계정 uid, gid변경
exec type : 4
war file path (default:/opt/infraeye/nms/webapps) :/opt/infraeye/nms/webapps
server file path (default:/opt/infraeye/was/nms-tomcat/conf) :/opt/infraeye/was/nms-tomcat/conf
완료되었습니다.
[root@57691d5f91c7 ETC]#
```

- #vflow 를 설치합니다. (기본 port를 사용하는게 아닐 경우 포트 포워딩을 확인해주세요.)

```
[root@57691d5f91c7 ETC]# ./container_util.sh
1. DB, PORT 패스워드 변경 (NMS_COMMON.conf, conf/context.xml, EngineConfig.sh)
2. IP, PORT 변경 (odbc.ini, context.xml, EngineConfig.sh)
3. CrateDB 계정 생성
4. was 파일명 변경 (server.xml)
5. vflow 설치
6. sflow 설치
7. infraeye 계정 uid, gid변경
exec type : 5
vflow file path (default:/docker_dir/COLLECT/vflow/vflow) :/docker_dir/COLLECT/v
vflow port v5 (default:2055) :2055
vflow port v9 (default:9996) :9996
완료되었습니다.
[root@57691d5f91c7 ETC]#
```

6. 참고 사항(Container Util)

- /docker_dir/ETC/container_util.sh 쉘을 실행합니다.
- #sflow 를 설치합니다. (기본 port를 사용하는게 아닐 경우 포트 포워딩을 확인해주세요.)

```
[root@57691d5f91c7 ETC]# ./container_util.sh
1. DB, PORT 패스워드 변경 (NMS_COMMON.conf, conf/context.xml, EngineConfig.sh)
2. IP, PORT 변경 (odbc.ini, context.xml, EngineConfig.sh)
3. CreateDB 계정 생성
4. was 파일명 변경 (server.xml)
5. vflow 설치
6. sflow 설치
7. infraeye 계정 uid, gid 변경
exec type : 6
logstash cpu core : 1
logstash memory :1
Logstash 설치를 시작합니다.

logstash rpm 을 설치합니다 .
경고 : /docker_dir/COLLECT/logstash/rpm/logstash-7.17.22-x86_64.rpm: Header V4 RSA/SHA512 Signature, key ID d88e42b4: NOKEY
Using bundled JDK: /usr/share/logstash/jdk
Using provided startup.options file: /etc/logstash/startup.options
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
/usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/pleaserun-0.0.32/lib/pleaserun/platform/base.rb:112: warning: constant ::Fixnum is deprecated
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: 프로세스가 죽었습니다
Successfully created system startup script for Logstash
logstash rpm 설치 성공
[CPU Worker][1]
[ DataPath ][/opt/infraeye/data/logstash/data/logstash]
[ Memory ][1]
[ Sol Path ][/docker_dir/COLLECT]
sFlow 플러그인을 설치합니다 .
Using bundled JDK: /usr/share/logstash/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
Installing file: /docker_dir/COLLECT/logstash/plugin/logstash-offline-plugins-7.17.22.zip
Resolving dependencies.....
Install successful

logstash 서비스를 infraeye로 변경합니다 .
```

6. 참고 사항(Container Util)

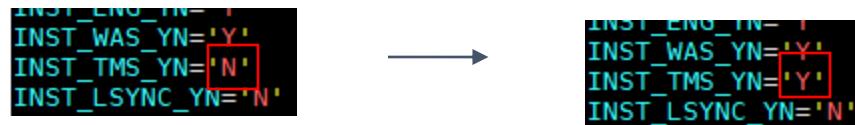
- /docker_dir/ETC/container_util.sh 쉘을 실행합니다.
- #컨테이너 내부 uid, gid 변경(호스트와 동일한 uid,gid를 사용하지 않을경우 공유 디렉터리에서 접근이 불가능합니다.)

```
[root@57691d5f91c7 ETC]# ./container_util.sh
1. DB, PORT 패스워드 변경 (NMS_COMMON.conf, conf
2. IP, PORT 변경 (odbc.ini, context.xml, EngineC
3. CrateDB 계정 생성
4. was 파일명 변경 (server.xml)
5. vflow 설치
6. sflow 설치
7. infraeye 계정 uid, gid변경
exec type : 7
user name :infraeye
uid :1000
gid :1000
컨테이너 내부 uid 를 1000로 변경합니다.
컨테이너 내부 gid 를 1000로 변경합니다.
완료되었습니다.
[root@57691d5f91c7 ETC]# ]
```

6. 참고 사항(TMS 수동 설치)

- vflow 또는 sflow 툴을 사전에 먼저 설치 후 다음 단계를 진행해 주세요.(이전 페이지 참고)
- default 포트를 사용하지 않을 경우 이전 페이지의 포트 추가 및 변경을 사용해주세요.

/opt/infraeye/nms/util/EngineConfig.sh 파일에서 INST_TMS_YN 값 Y로 변경



/docker_dir/config/container_config.yaml 파일에서 TMS_YN 값 y로 변경



#TMS 엔진 설정 추가 (sflow 사용시 FLOW_COLLECT_TYPE 값을 LOGSTASH로 변경하고 엔진 등록 시 NC_SFLOW_C 를 추가로 등록해주세요.)

```
[infraeye@45a157f44dc9 conf_engine]$ mysql -uroot -p'netcruz!#$134' < /docker_dir/DB/query/query_nms/conf_engine/conf_engine_tms.sql  
[infraeye@45a157f44dc9 conf_engine]$
```

#해당 서버의 SVR_ID를 확인하고 SP_NMS_WATCHDOG_SCH_INSTALL 을 호출해주세요

```
MariaDB [NMS_DB]>  
MariaDB [NMS_DB]> CALL SP_NMS_WATCHDOG_SCH_INSTALL(1, 5, '/opt', '/opt/netcruz');  
Query OK 15 rows affected (0.041 sec)
```

맨 앞의 인자만 확인한 SVR_ID로 변경하시면 됩니다.

6. 참고 사항(TMS 수동 설치)

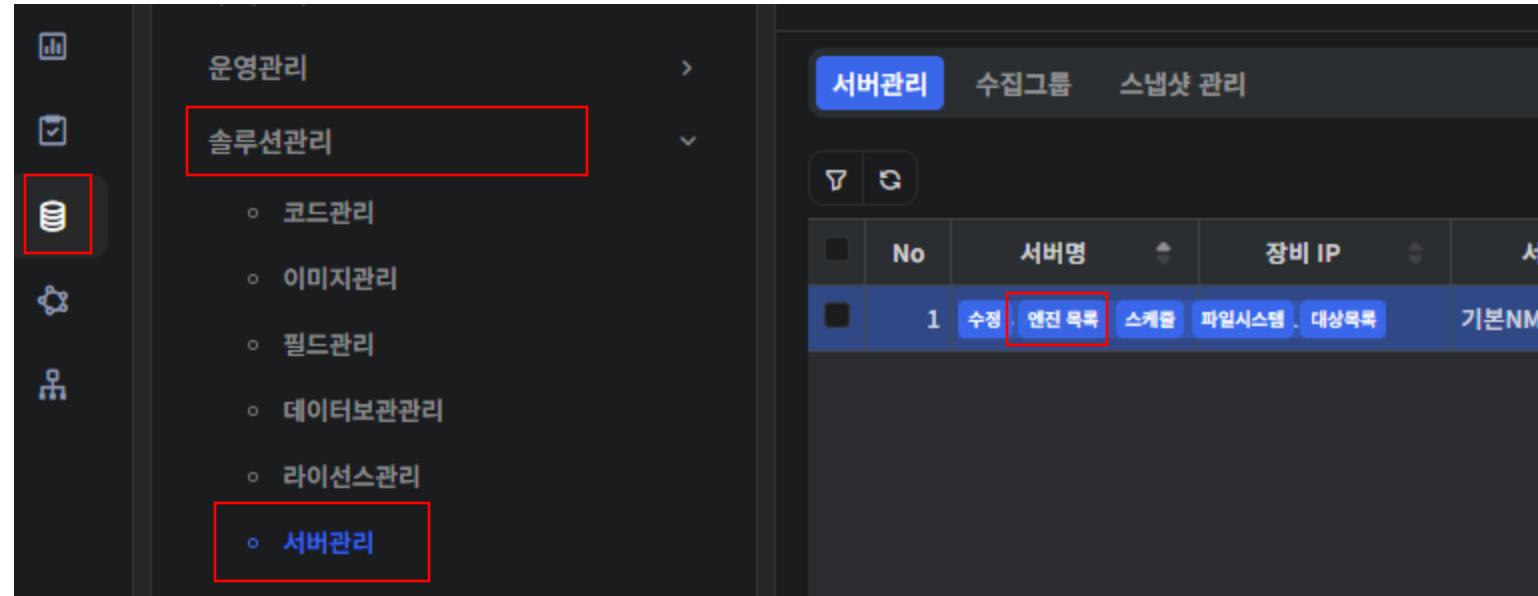
/etc/odbc.ini 파일에서 TMS_DB 주석 제거 (IP ,Port를 NMS 서버에 맞게 설정해주세요)

```
# [TMS_DB]
#Description = ODBC for MySQL Driver
#Driver = MySQL
#SERVER = 127.0.0.1
#PORT = 13306
#DATABASE = TMS_DB
```

→

```
[TMS_DB]
Description = ODBC for MySQL Driver
Driver = MySQL
SERVER = 127.0.0.1
PORT = 13306
DATABASE = TMS_DB
```

ui 화면에서 - 관리 -> 솔루션관리 -> 서버관리 -> TMS설치 서버 엔진목록 -> TMS 추가



6. 참고 사항(TMS 수동 설치)

엔진 수정

엔진명 *

NC_TMS

엔진 실행파일명 *

NC_TMS

엔진 경로 *

/opt/infraeye/nms/bin/NC_TMS&

사용 여부

ON

재시작 여부

ON

저장 닫기

오전 11:09:11

X

저장

엔진 상태	상태 변경시간	엔진 경로
실행중	2024-10-08 10:53:08	/opt/infraeye/nms/bin/NC_FMS&
실행안함	2024-10-08 10:52:52	/opt/infraeye/nms/bin/NC_IPSLA
실행중	2024-10-08 10:53:08	/opt/infraeye/nms/bin/NC_L4&
실행중	2024-10-08 10:53:08	/opt/infraeye/nms/bin/NC_NET_&
실행안함	2024-10-08 10:52:52	/opt/infraeye/nms/bin/NC_NOTIF
실행중	2024-10-08 10:53:08	/opt/infraeye/nms/bin/NC_PERF
실행중	2024-10-08 10:53:08	/opt/infraeye/nms/bin/NC_PERF
실행중	2024-10-08 10:53:08	/opt/infraeye/nms/bin/NC_RTT_&
실행안함	2024-10-08 10:52:52	/opt/infraeye/nms/bin/NC_RT_TG

6. 참고 사항(Log 관련사항)

- 컨테이너 생성 관련 로그 확인 방법

```
[root@localhost infraeye2_package]# docker logs infraeye_2.0
===== Container Init Setting Start =====
== 호스팅 서버와 uid, gid 를 기화합니다. ==
컨테이너 내부 uid 를 1002로 변경합니다.
컨테이너 내부 gid 를 1003로 변경합니다.
== root, infraeye 계정에 환경변수를 설정합니다. ==
== 기존 사용 디렉터리 Check ==
[cp_check_dir] check_dir : /docker_dir/DB/crate-5.4.7
[cp_check_dir] cp_dir : /opt/infraeye/data/crate/crate_bin
[cp_check_dir] check_dir : /docker_dir/WAS/mib_browser
[cp_check_dir] cp_dir : /opt/infraeye/data/filedata/mib_browser
[cp_check_dir] check_dir : /docker_dir/COLLECT/conf
[cp_check_dir] cp_dir : /opt/infraeye/nms/conf
[cp_check_dir] check_dir : /docker_dir/DB/redis/redis_conf
[cp_check_dir] cp_dir : /opt/infraeye/data/redis/redis_conf
[cp_check_dir] check_dir : /docker_dir/WAS/webobjects/jsp
[cp_check_dir] cp_dir : /opt/infraeye/nms/webobjects/jsp
[cp_check_dir] check_dir : /docker_dir/WAS/webobjects/patch
[cp_check_dir] cp_dir : /opt/infraeye/nms/webobjects/patch
[cp_check_dir] check_dir : /docker_dir/WAS/webobjects/resource
[cp_check_dir] cp_dir : /opt/infraeye/nms/webobjects/resource
NMS TP Setting
```

6. 참고 사항(Log 관련사항)

- 컨테이너 생성 관련 로그 확인 방법

```
[root@localhost infraeye2_package]# docker logs infraeye_2.0
===== Container Init Setting Start =====
== 호스트 서버와 uid, gid를 동기화합니다. ==
컨테이너 내부 uid 를 1002로 변경합니다.
컨테이너 내부 gid 를 1003로 변경합니다.
== root, infraeye 계정에 환경변수를 설정합니다. ==
== 기존 사용 디렉터리 Check ==
[cp_check_dir] check_dir : /docker_dir/DB/crate-5.4.7
[cp_check_dir] cp_dir : /opt/infraeye/data/crate/crate_bin
[cp_check_dir] check_dir : /docker_dir/WAS/mib_browser
[cp_check_dir] cp_dir : /opt/infraeye/data/filedata/mib_browser
[cp_check_dir] check_dir : /docker_dir/COLLECT/conf
[cp_check_dir] cp_dir : /opt/infraeye/nms/conf
[cp_check_dir] check_dir : /docker_dir/DB/redis/redis_conf
[cp_check_dir] cp_dir : /opt/infraeye/data/redis/redis_conf
[cp_check_dir] check_dir : /docker_dir/WAS/webobjects/jsp
[cp_check_dir] cp_dir : /opt/infraeye/nms/webobjects/jsp
[cp_check_dir] check_dir : /docker_dir/WAS/webobjects/patch
[cp_check_dir] cp_dir : /opt/infraeye/nms/webobjects/patch
[cp_check_dir] check_dir : /docker_dir/WAS/webobjects/resource
[cp_check_dir] cp_dir : /opt/infraeye/nms/webobjects/resource
[cp_check_dir] cp_dir : /opt/infraeye/nms/webobjects/resource
```

- Dockerfile 빌드 관련 로그 -> /docker_dir/ETC/docker_build_log/install.log
- 솔루션 DB 실행 로그 -> /opt/infraeye/logs/database/system.log