

Deploy Registry Server

사전 구성

Add host on /etc/hosts file

내부에서 사용하는 DNS가 없다면 호스트명으로 리졸빙(resolving) 하기 위해 /etc/hosts 파일을 다음과 같이 수정합니다.

```
vi /etc/hosts
.....
172.16.0.200      myregistry.com
```

Create ssl to Use for container

안전한 통신을 위해 TLS Termination 기능을 사용하기 위한 내부 인증서와 htpasswd로 구성된 사용자 및 암호를 설정한 파일을 생성합니다. 사용자는 **demouser** 그리고 암호는 **password**로 설정합니다. 그리고 로컬에서 테스트하기 위해 client 인증서를 구성합니다.

```
mkdir /auth ; cd /auth

docker run --rm -v /auth:/auth alpine/openssl \
req -newkey rsa:4096 -nodes -sha256 -x509 \
-days 365 -keyout /auth/myregistry.com.key -out /auth/myregistry.com.crt \
-subj '/CN=myregistry.com' \
-addext "subjectAltName = DNS:myregistry.com"

# create htpasswd file
docker run --rm --entrypoint htpasswd httpd:2.4 -Bbn demouser password > /auth/htpasswd
```

```
# client certification
mkdir -p /etc/docker/certs.d/myregistry.com
cp /auth/myregistry.com.crt /etc/docker/certs.d/myregistry.com/ca.crt
```

Deploy Registry and test

아래의 1,2,3번중 하나의 방식을 선택해서 registry 서버를 배포합니다. 포트 충돌 때문에 동시에 배포 하지 못합니다.

1. Run Basic Docker Registry

제한이 없는 기본 레지스트리입니다.

```
docker run -d -p 5000:5000 --name registry-server registry:latest
```

```
# test 이미지 생성 및 publishing
docker pull docker.io/hello-world
docker tag docker.io/hello-world localhost:5000/hello-world
docker push localhost:5000/hello-world

# 기존 로컬에서 이미지 삭제 후 registry에서 다운로드 및 실행
docker rmi docker.io/hello-world localhost:5000/hello-world
docker run --name demo1 localhost:5000/hello-world

# clean up
docker rm -f demo1
docker rm -f registry-server
docker rmi localhost:5000/hello-world
```

2. Run Docker Registry with SSL

SSL 인증서로 암호화된 통신을 사용하는 레지스트리입니다.

```
docker run -d -p 443:443 \
--restart=always --name registry \
-v /auth:/certs \
-v ~/image-data:/var/lib/registry \
-e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/myregistry.com.crt \
-e REGISTRY_HTTP_TLS_KEY=/certs/myregistry.com.key \
-e REGISTRY_HTTP_ADDR=0.0.0.0:443 \
registry
```

```
# test 이미지 생성 및 publishing
docker pull docker.io/hello-world
docker tag docker.io/hello-world myregistry.com/hello-world
docker push myregistry.com/hello-world

# 기존 로컬에서 이미지 삭제 후 registry에서 다운로드 및 실행
docker rmi docker.io/hello-world myregistry.com/hello-world
docker run --name demo2 myregistry.com/hello-world

# clean up
docker rm -f demo2 registry
docker rmi myregistry.com/hello-world
```

3. Run Docker Registry with Restriction

SSL Termination 과 Authentication 기능을 모두 함께 사용하는 레지스트리입니다.

```
docker run -d -p 443:443 --restart=always --name registry \
-v /auth:/certs \
-v /image-vol:/var/lib/registry \
-e REGISTRY_HTTP_ADDR=0.0.0.0:443 \
-e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/myregistry.com.crt \
-e REGISTRY_HTTP_TLS_KEY=/certs/myregistry.com.key \
-e "REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e REGISTRY_AUTH_HTPASSWD_PATH=/certs/htpasswd \
registry
```

```
# login to registry
docker login myregistry.com
username: demouser
password: password
```

```
# test 이미지 생성 및 publishing
docker pull docker.io/hello-world
docker tag docker.io/hello-world myregistry.com/hello-world
docker push myregistry.com/hello-world

# 기존 로컬에서 이미지 삭제 후 registry에서 다운로드 및 실행
docker rmi docker.io/hello-world myregistry.com/hello-world
docker run --name demo3 myregistry.com/hello-world

# clean up
docker rm -f demo3 registry
docker rmi myregistry.com/hello-world
```

(추가 선택사항) Other Method

Run Docker Registry with docker-compose

docker compose 파일을 활용하여 Registry 컨테이너를 실행시킬 수 있습니다. 다음은 docker-compose.yaml 파일 내용입니다.

```
# docker-compose.yml
services:
  registry:
    image: registry
    ports:
      - 443:443
    environment:
      REGISTRY_HTTP_ADDR: 0.0.0.0:443
      REGISTRY_HTTP_TLS_CERTIFICATE: /certs/myregistry.com.crt
      REGISTRY_HTTP_TLS_KEY: /certs/myregistry.com.key
      REGISTRY_AUTH: htpasswd
      REGISTRY_AUTH_HTPASSWD_PATH: /certs/htpasswd
      REGISTRY_AUTH_HTPASSWD_REALM: Registry Realm

  volumes:
    - /image-vol:/var/lib/registry
    - /auth:/certs
```

파일을 생성한 위치에서 docker compose 명령어를 사용하여 컨테이너를 실행할 수 있습니다.

```
docker compose up -d
# docker compose down 명령어로 컨테이너를 삭제 할 수 있습니다.
```

Install Distribution

registry는 컨테이너로 배포할 수 있지만, 또 다른방법으로는 데몬으로 실행할 수 있습니다. docker-distribution이라는 패키지를 사용하는 데몬으로 호스트에서 직접 관리 할 수 있습니다.

```
yum install docker-distribution
systemctl start docker-distribution
systemctl enable docker-distribution

cat /etc/docker-distribution/registry/config.yml # distribution 구성파일 확인
```