Harbor 설치

* 버전 정보

Docker Ver: 24.0.1

Docker compose Ver: v2.17.2

Harbor Ver: v2.4.1

OS: CentOS7.9

* 설치

1. **인증서 생성**

$ mkdir -p ~/certs

$ cd ~/certs

## CA Certificates 생성

$ openssl genrsa -out ca.key 4096

$ openssl req -x509 -new -nodes -sha512 -days 365 \

-key ca.key \

-out ca.crt

## Server Certificates 생성

$ openssl genrsa -out server.key 4096

$ openssl req -sha512 -new \

-key server.key \

-out server.csr

## SAN 등록

$ vi v3ext.cnf

|  |
| --- |
| # v3ext.cnf 파일 내용  subjectAltName = IP:10.100.0.204,IP:127.0.0.1 |

$ openssl x509 -req -sha512 -days 365 \

-extfile v3ext.cnf \

-CA ca.crt -CAkey ca.key -CAcreateserial \

-in server.csr \

-out server.crt

## Certificate Update

$ openssl x509 -inform PEM -in server.crt -out server.cert

# host의 이름을 k8s-nfs 라고 했으므로, /certs.d/k8s-nfs 로 생성합니다.

mkdir -p /etc/docker/certs.d/k8s-nfs

cp server.cert /etc/docker/certs.d/k8s-nfs

cp server.key /etc/docker/certs.d/k8s-nfs

cp ca.crt /etc/docker/certs.d/k8s-nfs

# Host

sudo cp ca.crt /etc/pki/ca-trust/source/anchors

sudo cp server.crt /etc/pki/ca-trust/source/anchors

sudo update-ca-trust extract

1. **Docker 설치**

######### docker 설치 - Ubuntu #########

Set up the repository

Update the apt package index and install packages to allow apt to use a repository over HTTPS:

1. sudo apt-get update

2. sudo apt-get install ca-certificates curl gnupg lsb-release

Add Docker’s official GPG key:

3. sudo mkdir -m 0755 -p /etc/apt/keyrings

4. curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

Use the following command to set up the repository:

5. echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] <https://download.docker.com/linux/ubuntu> $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Install Docker Engine

Update the apt package index:

6. sudo chmod a+r /etc/apt/keyrings/docker.gpg

7. sudo apt-get update

Install Docker Engine, containerd, and Docker Compose.

8. sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y

9. sudo docker run hello-world

######### docker 설치 - CentOS #########

1. yum install -y yum-utils

2. docker repo 등록

yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

3. docker install

yum install -y docker-ce docker-ce-cli containerd.io

4. docker start

systemctl start docker

5. docker version 확인

docker version

**3 Docker Compose 설치**

$ curl -SL <https://github.com/docker/compose/releases/download/v2.17.2/docker-compose-linux-x86_64> -o /usr/local/bin/docker-compose

$ chmod +x /usr/local/bin/docker-compose

$ docker-compose --version

**4. Harbor 설치**

- 설치파일 다운로드

$ wget <https://github.com/goharbor/harbor/releases/download/v2.4.1/harbor-offline-installer-v2.4.1.tgz>

- 압축 해제

$ tar zxvf harbor-offline-installer-v2.4.1.tgz

- 설치 작업 시작

$ cd harbor

$ cp harbor.yml.tmpl harbor.yml

$ vi harbor.yml

|  |
| --- |
| hostname: 10.100.0.204  http:  # port for http, default is 80. If https enabled, this port will redirect to https port  port: 80  https:  # https port for harbor, default is 443  port: 443  # The path of cert and key files for nginx  certificate: /etc/docker/certs.d/k8s-nfs/server.cert  private\_key: /etc/docker/certs.d/k8s-nfs/server.key  harbor\_admin\_password: dlshxmfl1!  database:  # The password for the root user of Harbor DB. Change this before any production use.  password: dlshxmfl1!  # The maximum number of connections in the idle connection pool. If it <=0, no idle connections are retained.  max\_idle\_conns: 50  # The maximum number of open connections to the database. If it <= 0, then there is no limit on the number of open connections.  # Note: the default number of connections is 1024 for postgres of harbor.  max\_open\_conns: 1000  data\_volume: /data |

- Deploy

$ ./prepare

$ ./install.sh

$ docker-compose ps

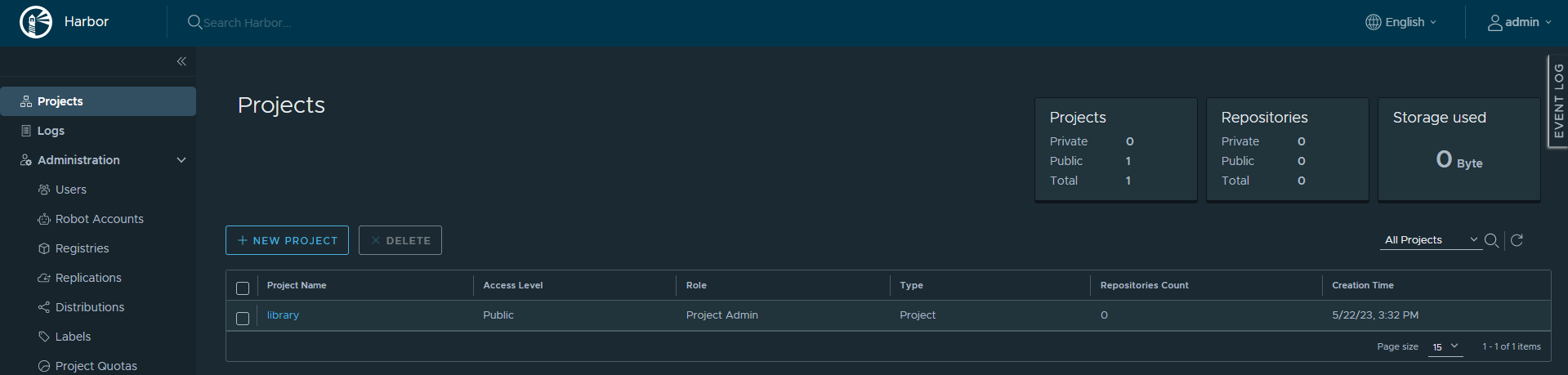
## 이후 10.100.0.204 URL로 접속

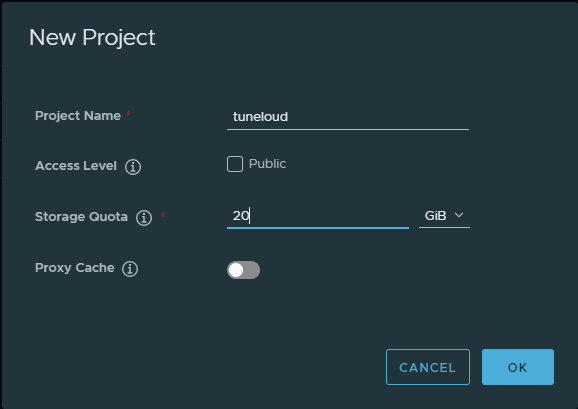
초기 계정 정보는 admin에 {harbor\_admin\_password} 항목의 pw로 접근이 가능



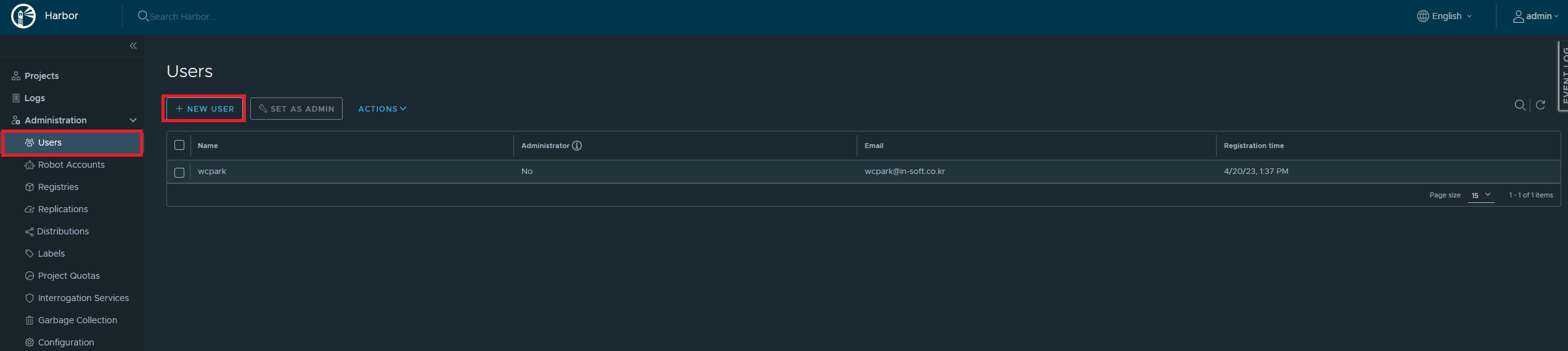
* 테스트

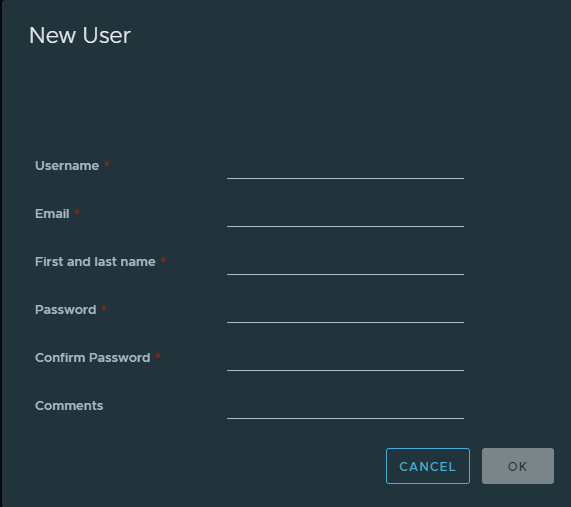
1. 프로젝트 생성





1. 사용자 생성 및 배정





1. 이미지 PUSH & PULL

$ docker pull nginx ## docker hub 에서 image pull 테스트를 위한 nginx latest

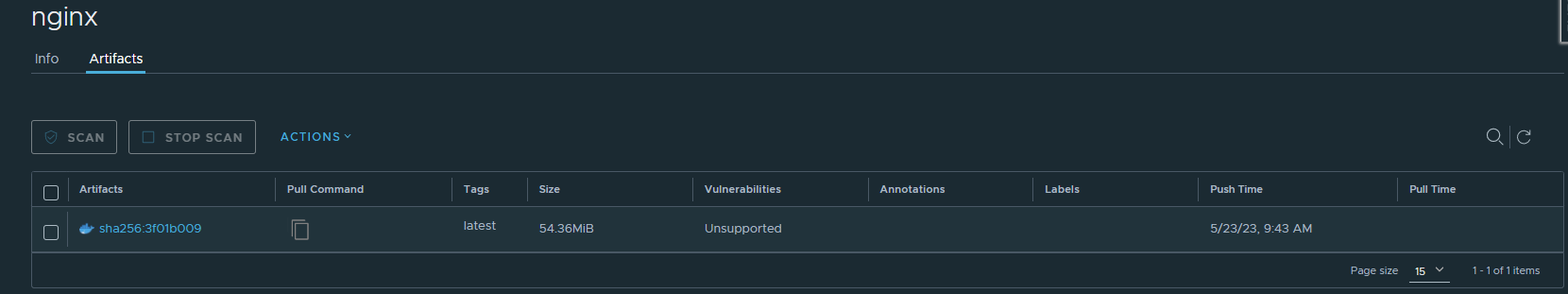
$ docker images

$ docker tag nginx:latest 10.100.0.204/tunecloud/nginx:latest ## tag 생성

$ docker login <https://10.100.0.204> ## harbor log in

|  |
| --- |
| ## image를 push 하려는 서버에서 인증을 위한 ca 파일이 있어야함  경로 : /etc/docker/certs.d/{harbor 서버 ip}    테스트를 위한 harbor가 설치된 10.100.0.204 에서의 인증서 파일 위치:  /etc/docker/certs.d/10.100.0.204 해당 위치의 파일들을 push를 하려는 서버의 위치에 위치 시켜야함  인증서 이동 후 docker 재시작 필요  $ systemctl restart docker |

$ docker push 10.100.0.204:80/tunecloud/nginx:latest ## image push



1. **k8s 연동 (CRI = containerd)**

\*\* k8s 1.24 버전 부터 docker 컨테이너 런타임 지원 중단으로 이상버전에서 사용 하려면 conainerd or CRI-O 같은 호환 가능한 컨테이너 런타임을 사용 해야함

default path : /etc/contaienrd/config.toml

1. containderd 설정 파일 이동

$ cd /etc/containerd

\*\* 해당 경로에 config.toml 파일이 없거나 내용이 없는경우 설정파일 생성 필요함

$ containerd config default > /etc/containerd/config.toml

1. config.toml 파일 수정

|  |
| --- |
| * private registry 주소 지정   [plugins.”io.containerd.grpc.v1.cri”.registry] 하위 mirrors 설정이 private registry 주소 등록하는 부분     * harbor 인증 정보 설정   [plugins."io.containerd.grpc.v1.cri".registry.configs] 하위에 auth 설정 추가 username 과 passwd 는 harbor 설치시 설정한 계정 정보 입력     * tls 인증서 설정   [plugins."io.containerd.grpc.v1.cri".registry.configs] 하위에 tls 설정 추가    [plugins."io.containerd.grpc.v1.cri".registry.configs." 10.100.0.204".tls]  ca\_file = "/etc/docker/certs.d/10.100.0.204/ca.crt"  cert\_file = "/etc/docker/certs.d/10.100.0.204/server.cert"  key\_file = "/etc/docker/certs.d/10.100.0.204/server.key"   * 설정 파일 최종 형태     [plugins."io.containerd.grpc.v1.cri".registry]  [plugins."io.containerd.grpc.v1.cri".registry.mirrors]  [plugins."io.containerd.grpc.v1.cri".registry.mirrors."10.100.0.204"]  endpoint = ["https://10.100.0.204"]  [plugins."io.containerd.grpc.v1.cri".registry.configs]  [plugins."io.containerd.grpc.v1.cri".registry.configs."10.100.0.204".auth]  username = "admin"  password = "dlshxmfl1!"  [plugins."io.containerd.grpc.v1.cri".registry.configs."10.100.0.204".tls]  ca\_file = "/etc/docker/certs.d/10.100.0.204/ca.crt"  cert\_file = "/etc/docker/certs.d/10.100.0.204/server.cert"  key\_file = "/etc/docker/certs.d/10.100.0.204/server.key" |

$ systemctl restart containerd

설치 삭제

$ docker-compose down

참고 사이트

<https://goharbor.io/docs/2.6.0/install-config/download-installer/>

<https://ikcoo.tistory.com/229>