



—  
Catalog Guide  
—

# HyperRegistry 사용자 가이드

2021. 12. 28



## 목차

1. 계정 로그인 .....	3
1.1 로그인 .....	3
2. HyperRegistry Project 생성.....	4
2.1 https://demoharbor.tmaxcloud.com/ 접속 (Login via OIDC Provider 클릭) .....	5
2.2 Create a project (프로젝트 생성).....	6
2.3 Registry ID / Password 확인.....	6
3 Docker Image Pull / Push .....	7
3.1 Docker Login .....	7
3.2 Image Push .....	7
3.3 Image Push 결과 확인 .....	8
4 Pod 생성 .....	9
4.1 Image Pull Secret 생성.....	9
4.1.1 하이퍼 클라우드 접속 - 콘솔(마스터 클러스터)- 워크로드 - 시크릿 .....	9
4.1.2 시크릿 생성 (우측 상단 - 시크릿 생성 - 이미지 풀 시크릿).....	9
4.2. 파드 생성 / 확인.....	10

# SuperCloud Portal Guide – HyperRegistry

## 1. 카탈로그 개요

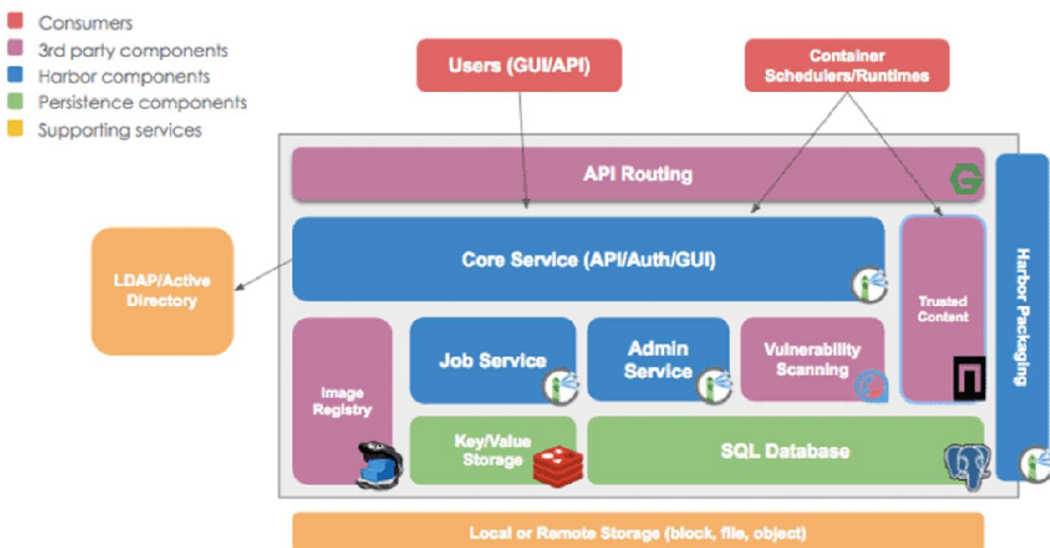
Registry는 컨테이너 작업을 수행하는데 있어서 구성 요소 중 하나로 컨테이너를 실행 시 컨테이너이미지가 필요하게 되는데 컨테이너 이미지를 보관하는 위치가 Registry 입니다. Registry에 이미지를 저장하고 이를 호스팅하여 여러 호스트서버에서 이미지를 다운로드 / 실행을 하게 됩니다. 대표적으로 Docker Registry가 있는데 이는 가장 기본 단위입니다. 특별한 기능없이 단순 저장/호스팅 기능만 있기 때문에 사용자 입장에서 사용하는데 불편하여 SuperCloud에서는 HyperRegistry 라는 컨테이너 이미지 저장소를 사용합니다.

HyperRegistry 는 유저가 사용하기 편하게 Web UI를 제공하고, 여러 모듈을 통해 이미지의 유효성, 안정성을 확인 가능합니다. 여러 유저가 자신만의 레지스트리를 생성할 수 있고, Public 레지스트리를 생성해 여러 사람들과 공유도 가능합니다.

## 2. 카탈로그 특징

HyperRegistry 는 Core Service 의 API 를 통해 유저 생성 / 프로젝트 생성을 하고 유저, 인증, 프로젝트, 이미지관리는 SQL Database 에 저장을 하고 이미지의 Layer 는 Image Registry 에 저장합니다. 유저들에게 어려운 Registry 관리를 Web UI 통해 제공합니다.

## Architecture

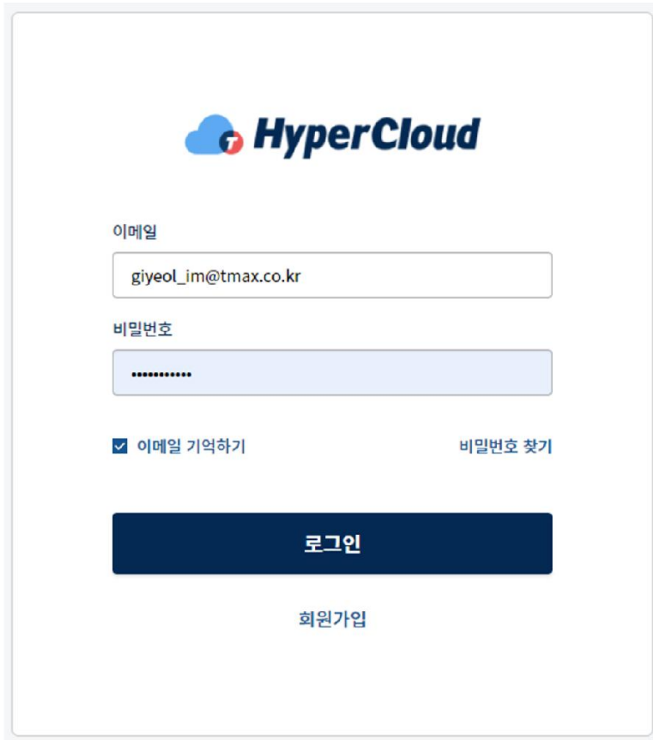


## STEP1 – 계정 로그인

---

### 1. 사용자 계정으로 로그인

demo.tmaxcloud.com에 접속하여 로그인



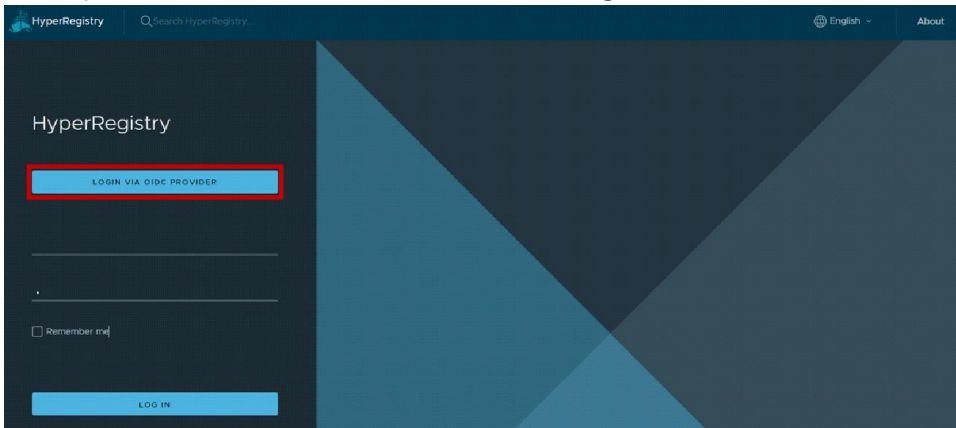
The image shows the HyperCloud login interface. At the top is the HyperCloud logo. Below it are two input fields: '이메일' (Email) with the value 'giyeol\_im@tmax.co.kr' and '비밀번호' (Password) with masked characters. There are two checkboxes: '이메일 기억하기' (Remember email) which is checked, and '비밀번호 찾기' (Find password). Below these is a large blue '로그인' (Login) button and a smaller link for '회원가입' (Sign up).

## STEP2 – HyperRegistry Project 생성

---

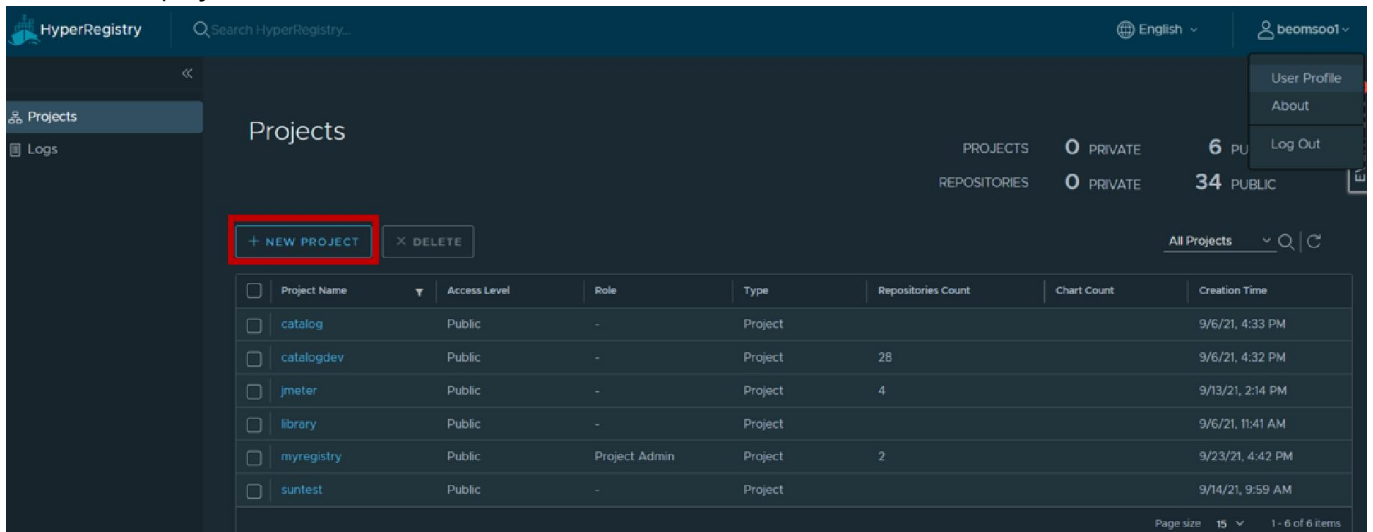
### 1. HyperRegistry 로그인

1.1 <https://demoharbor.tmaxcloud.com/> 접속 (Login via OIDC Provider 클릭)



The image shows the HyperRegistry login page. The header includes the HyperRegistry logo, a search bar, and links for 'English' and 'About'. The main content area has the title 'HyperRegistry' and a prominent blue button labeled 'LOGIN VIA OIDC PROVIDER' which is highlighted with a red rectangle. Below this are two input fields for username and password, a 'Remember me' checkbox, and a blue 'LOGIN' button at the bottom.

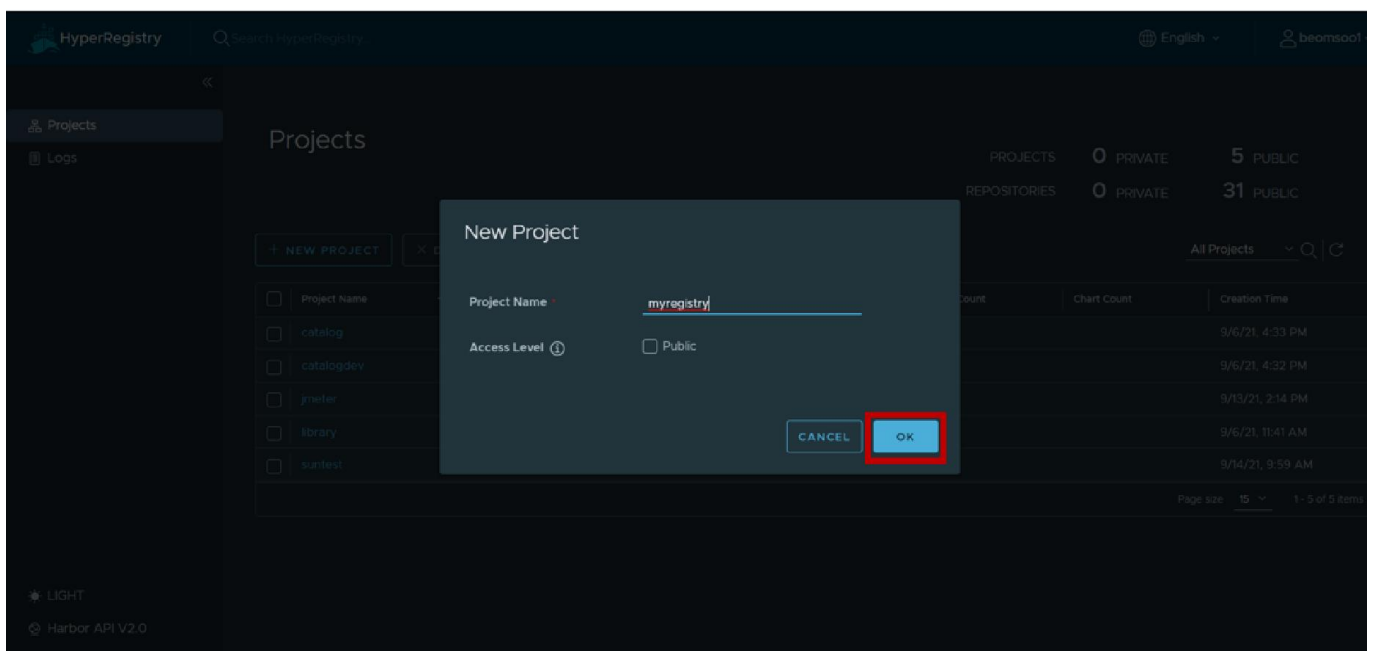
## 1.2 Create a project (프로젝트 생성)



The screenshot shows the HyperRegistry interface. The left sidebar has 'Projects' and 'Logs' options. The main area is titled 'Projects'. At the top right, there are statistics: PROJECTS 0 PRIVATE 6 PUBLIC and REPOSITORIES 0 PRIVATE 34 PUBLIC. A user profile dropdown for 'beomsol' is visible. Below the statistics, there are two buttons: '+ NEW PROJECT' (highlighted with a red box) and 'X DELETE'. Below these buttons is a table of existing projects.

<input type="checkbox"/>	Project Name	Access Level	Role	Type	Repositories Count	Chart Count	Creation Time
<input type="checkbox"/>	catalog	Public	-	Project			9/6/21, 4:33 PM
<input type="checkbox"/>	catalogdev	Public	-	Project	28		9/6/21, 4:32 PM
<input type="checkbox"/>	jmeter	Public	-	Project	4		9/13/21, 2:14 PM
<input type="checkbox"/>	library	Public	-	Project			9/6/21, 11:41 AM
<input type="checkbox"/>	myregistry	Public	Project Admin	Project	2		9/23/21, 4:42 PM
<input type="checkbox"/>	suntest	Public	-	Project			9/14/21, 9:59 AM

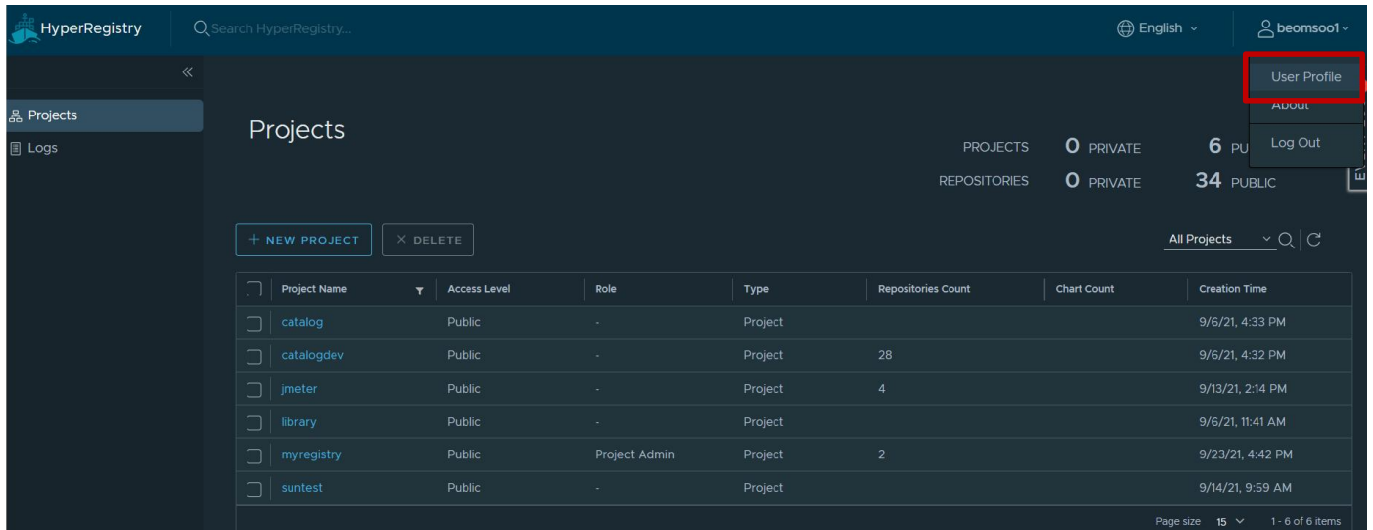
Page size: 15 | 1 - 6 of 6 items



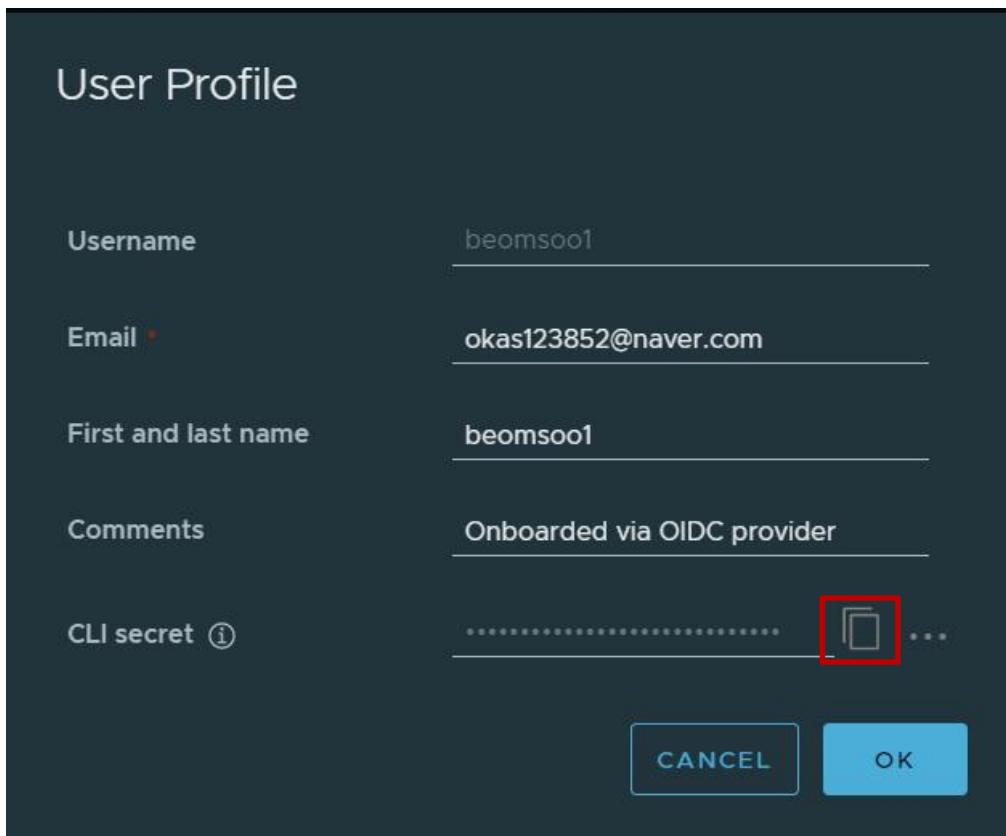
The screenshot shows the same HyperRegistry interface, but with a 'New Project' modal dialog open. The dialog has two fields: 'Project Name' with the value 'myregistry' and 'Access Level' with a dropdown menu showing 'Public'. At the bottom of the dialog are 'CANCEL' and 'OK' buttons, with the 'OK' button highlighted by a red box. The background shows the same project list as the first screenshot, but with updated statistics: PROJECTS 0 PRIVATE 5 PUBLIC and REPOSITORIES 0 PRIVATE 31 PUBLIC.

## 2. Registry ID / Password 확인

좌측 상단 Name – User Profile



ID : Username / Password : 복사 버튼을 눌러 확인

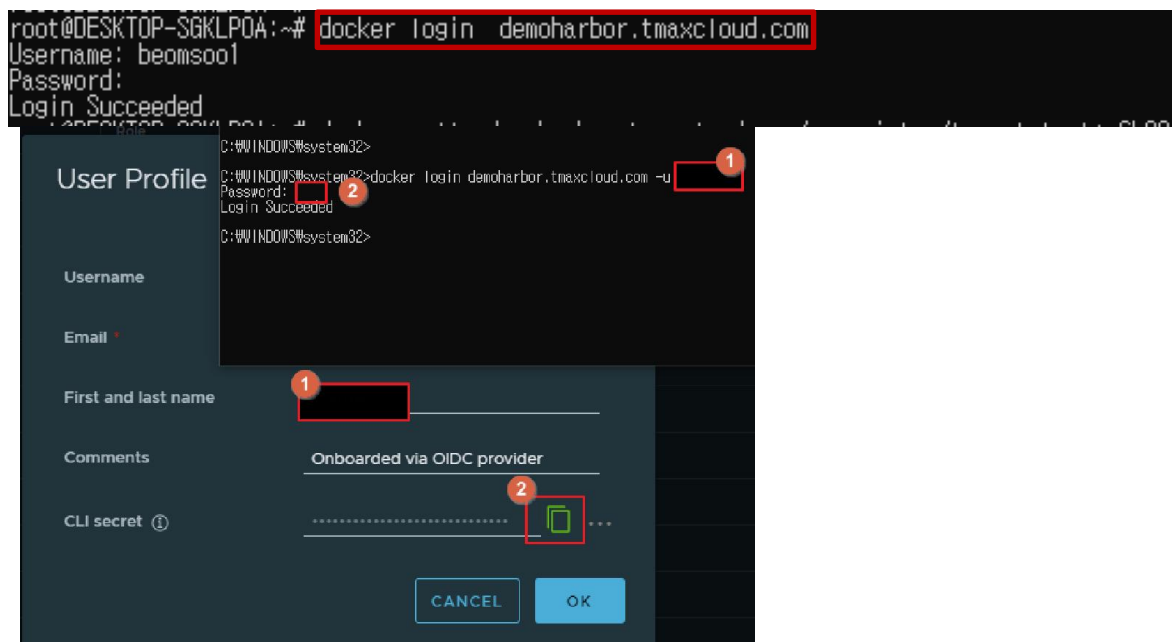


## STEP3 – Docker Image Pull / Push

### 1. Linux

#### 1.1 Docker Login

```
# docker login demoharbor.tmaxcloud.com
Username: [ 유저 이름 ]
Password: [ 유저 패스워드 ]
```



#### 1.2 Image Pull(Public Image Pull -> Docker)

```
# 예제 이미지 nginx:latest
```

```
docker pull nginx
```

```
root@DESKTOP-SGKLPOA:~# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
a330b6cecb98: Pull complete
e0ad2c0621bc: Pull complete
9e56c3e0e6b7: Pull complete
09f31c94adc6: Pull complete
32b26e9cdb83: Pull complete
20ab512bbb07: Pull complete
Digest: sha256:853b221d3341add7aaadf5f81dd088ea943ab9c918766e295321294b035f3f3e
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

#### 1.3 Image Tag

```
# 이미지 태그 변경
```

```
docker tag nginx demoharbor.tmaxcloud.com/[생성한 프로젝트네임]/nginx:v1
```

```
root@DESKTOP-SGKLPOA:~# docker tag nginx:latest demoharbor.tmaxcloud.com/myregistry/nginx:v1
```

## 1.4 Image Push(Docker -> HyperRegistry)

# 이미지 push

```
docker push demoharbor.tmaxcloud.com/[생성한 프로젝트네임]/nginx:v1
```

```
root@DESKTOP-SGKLPOA:~# docker push demoharbor.tmaxcloud.com/myregistry/nginx:v1
The push refers to repository [demoharbor.tmaxcloud.com/myregistry/nginx]
fac15b2caa0c: Pushed
f8bf5746ac5a: Pushed
d11eedadb34: Pushed
797e583d8c50: Pushed
bf9ce92e8516: Pushed
d000633a5681: Pushed
v1: digest: sha256:6fe11397c34b973f3c957f0da22b09b7f11a4802e1db47aef54c29e2813cc125 size: 1570
```

## 1.5 Image Push 결과 확인

Projects

myregistry *Project Admin*

Summary Repositories Helm Charts Members Labels Scanner P2P Preheat Policy Robot Accounts Webhooks Logs Configuration

X DELETE PUSH COMMAND 🔍 ☰ ↺

<input type="checkbox"/>	Name	Artifacts	Pulls	Last Modified Time
<input type="checkbox"/>	myregistry/httpd-2.4.48	1	2	9/24/21, 4:21 PM
<input type="checkbox"/>	myregistry/mariadb-10.5.12	1	4	9/24/21, 4:04 PM
<input type="checkbox"/>	myregistry/nginx	1		9/24/21, 1:22 PM
<input type="checkbox"/>	myregistry/tomcat-9.0.52	1	12	9/24/21, 4:20 PM
<input type="checkbox"/>	myregistry/tomcat-test	1	4	9/24/21, 1:35 PM

Page size 15 1 - 5 of 5 items

## 2. Window로 Image 업로드 방법

### 2.1 DockerHub Login

이미지를 다운 받을 Docker Hub에 로그인

```
C:\WINDOWS\system32>docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username:
Password:
time="2021-11-08T18:32:13+09:00" level=error msg="(661a26b0) 6ceah893-CredentialHelperPKG C<-S No response POST /registry/credstore-updated (1.0070138s): Post
```



## 1.2 이미지 받아오기

docker pull httpd

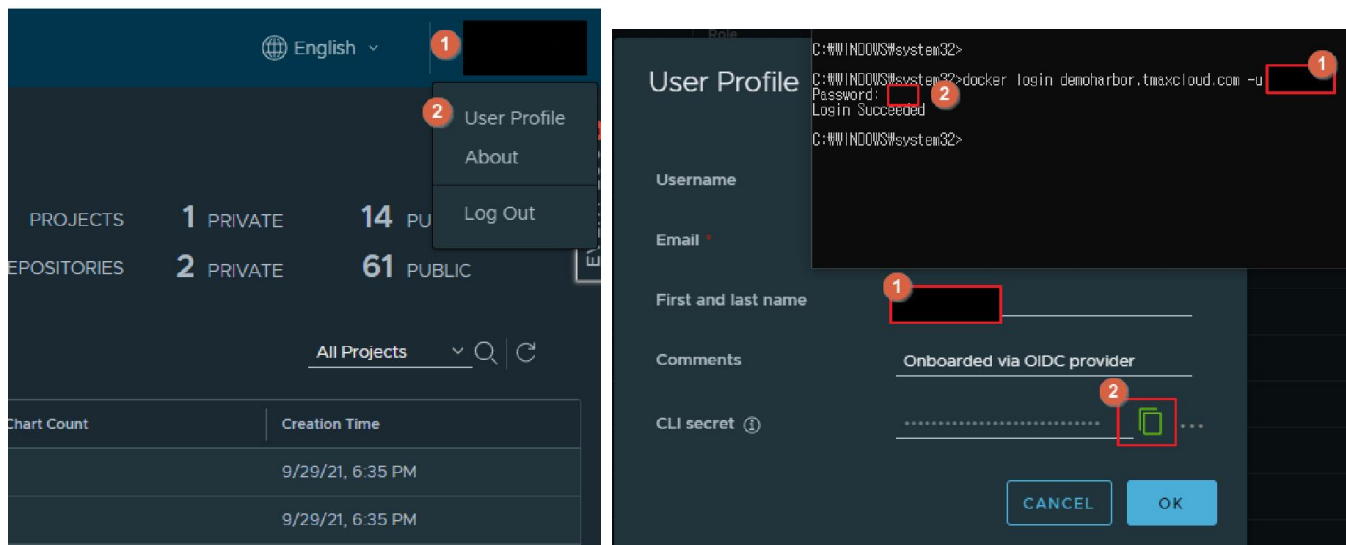
```
C:\WINDOWS\system32>docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
7d63c13d9b9b: Pull complete
ca52f3eeea66: Pull complete
448256567156: Pull complete
21d69ac90caf: Pull complete
462e88bc3074: Pull complete
Digest: sha256:f70876d78442771406d7245b8d3425e8b0a86891c79811af94fb2e12af0fadedb
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

## 1.3 HyperRegistry 로그인

```
# docker login demoharbor.tmaxcloud.com
```

Username: [ 유저 이름 ] -> HyperRegistry의 프로파일 이름

Password: [ 유저 패스워드 ] -> HyperRegistry의 CLI secret값을 복사하여 기입



## 1.4 HyperRegistry 로 이미지 push 하기

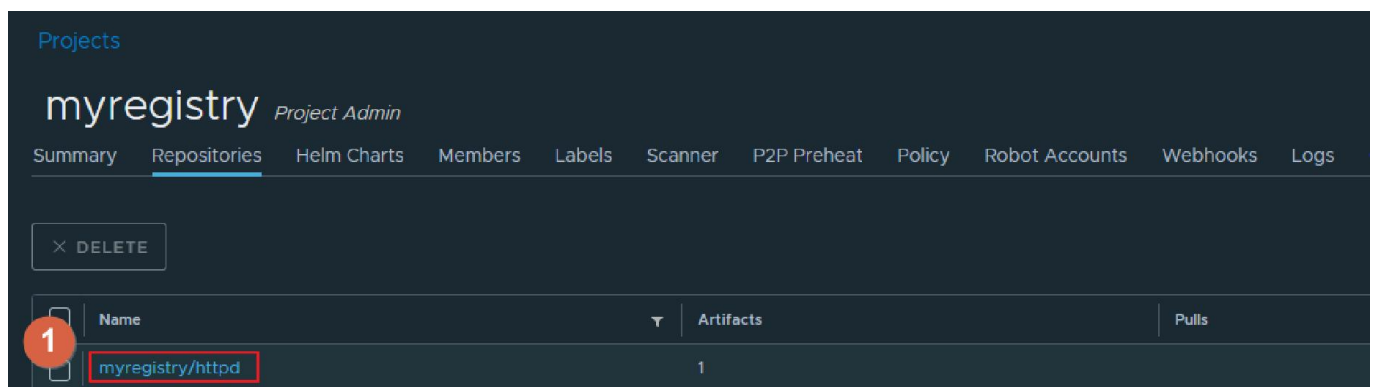
docker tag [docker hub image] [HyperRegistry\_URL/[본인의 프로젝트]/(변경할 이미지명)]

docker push [HyperRegistry\_URL/[본인의 프로젝트]/(변경할 이미지명)]

```
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe 1 2>docker tag httpd demoharbor.tmaxcloud.com/myregistry/httpd
C:\WINDOWS\system32\cmd.exe 2 2>docker push demoharbor.tmaxcloud.com/myregistry/httpd
Using default tag: latest
The push refers to repository [demoharbor.tmaxcloud.com/myregistry/httpd]
4dcdec0b7a0e: Pushed
c86537ee54f9: Pushed
ecd2b49ef243: Pushed
7511c367f47a: Pushed
e8b689711f21: Pushed
latest: digest: sha256:73c9b78280a693050838e9e3519e7f5723d742ada3e42c45f10744b4d88f486e size: 1365
```

## 1.5 이미지 확인

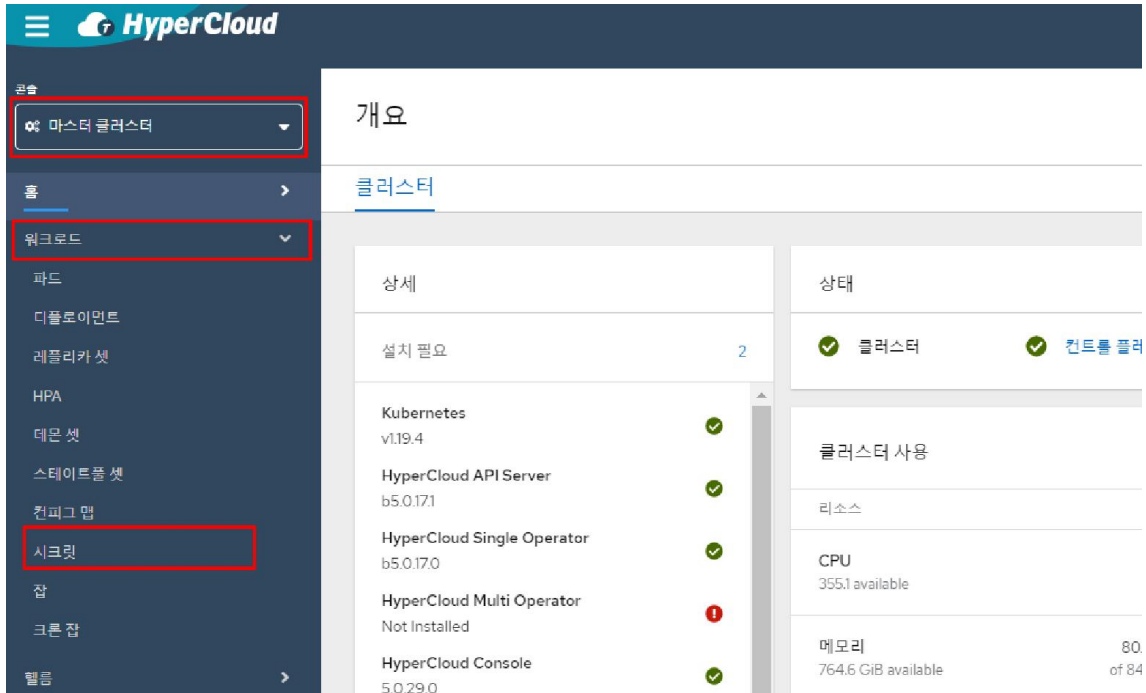
본인의 프로젝트에 이미지 리스트를 확인 – 위에서 PUSH한 httpd 이미지가 존재



## STEP4 – Pod 생성

### 1. Image Pull Secret 생성

1.1 SuperCloud 접속 > [마스터 클러스터] > [워크로드] > [시크릿] 선택



1.2 시크릿 생성 (우측 상단 – 시크릿 생성 – 이미지 풀 시크릿)

이미지 풀 시크릿은 일반사용자가 pull을 할 때 docker login이 필요한 것과 같이 SuperCloud에서 사용자가 push한 이미지를 사용하기 위해 이미지 레지스트리의 접근 정보를 저장해주는 오브젝트이다.

#### 이미지 풀 시크릿 생성

이미지 풀 시크릿을 사용하여 프라이빗 이미지 레지스트리에 대해 인증할 수 있습니다.

시크릿 이름 \*

새로운 시크릿의 고유한 이름

인증 타입

레지스트리 서버 주소 \*

사용자 이름 \*

비밀번호 \*

이메일

시크릿 이름 : imagesecret

인증 타입 : 이미지레지스트리자격증명

레지스트리 서버 주소 : demoharbor.tmaxcloud.com

사용자 이름 : HyperRegistry 계정

비밀번호 : HyperRegistry 패스워드

자격 증명 추가

생성 취소

## 2. HyperRegistry Image를 이용한 파드 생성

SuperCloud 접속 > [마스터 클러스터] > [워크로드] > [파드] > [파드 생성] 클릭

The screenshot shows the 'Pods' management interface in the SuperCloud portal. On the left, a sidebar menu is visible with '마스터 클러스터' selected. The main content area displays a table of pods. The table has the following columns: 이름 (Name), 네임스페이스 (Namespace), 상태 (Status), 준비 (Ready), 재시작 (Restart), 메모리 (Memory), CPU, and 생성 시간 (Creation Time). There are four pods listed, all in 'Running' status. A '파드 생성' (Create Pod) button is located in the top right corner of the table area.

이름	네임스페이스	상태	준비	재시작	메모리	CPU	생성 시간
centos7-sample-app-645f4d964f-lr09	beomsoo	Running	1/1	0	7.4 MiB	-	2021/09/10 09:44:23
sample-apache-5944c9654-fbb7h	beomsoo	Running	1/1	0	36.3 MiB	0.000 cores	2021/09/24 16:20:53
sample-mariadb-789461d946-ujwxf	beomsoo	Running	1/1	0	84.1 MiB	0.000 cores	2021/09/24 16:08:10
sample-tomcat-59789b7b69-jk97k	beomsoo	Running	1/1	0	91.0 MiB	0.001 cores	2021/09/24 16:20:27

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  namespace: beomsoo
spec:
  containers:
    - name: nginx
      image: 'demoharbor.tmaxcloud.com/myregistry/nginx:v1'
  imagePullSecrets:
    - name: imagesecret
```

## 파드

필터




이름

nginx

이름

nginx

모든 필터 지우기

이름 ↑	네임스페이스 ↓	상태 ↓	준비 ↓	재시작 ↓	메모리 ↓
 nginx	 beomsoo	 Running	1/1	0	-