

# Harbor 1.7 https仓库管理安装说明

- 新版本Harbor 1.7新增功能
  - 镜像构建历史
  - 镜像复制 (retagging)
  - 在线GC (垃圾回收)
  - 支持定制 Logger
  - 新增Helm Chart库
- 服务器的安装环境需求
  - 硬件要求:
  - 软件要求:
  - 网络端口要求:
- 安装Harbor 1.7
  - 下载Harbor 1.7
  - 解压修改并修改配置
  - 安装Harbor
- 常用命令
- 更新Harbor配置

## 新版本Harbor 1.7新增功能

相比harbor 1.5 版本, harbor 新增一下功能

- 镜像构建历史
  - -用户可查看容器镜像的构建历史和内容;
- 镜像复制 (retagging)
  - -提供了在镜像上传至Harbor后重新创建镜像tag的能力。此功能在CI流水线中提升镜像到生产状态或者通过编程方式重新tag镜像, 亦或将特定镜像重新tag或者移动到其它仓库或者项目等场景中特别有用;
- 在线GC (垃圾回收)
  - -现在 Harbor 可以清理从后端存储中已删除的镜像且在执行GC操作之前不再要求中断 Harbor 的运行。这对那些在 CI 构建流水线中使用Harbor而导致大量镜像垃圾碎片的用户来说是个福音;
- 支持定制 Logger
  - -允许用户为在 job 服务中运行的任务定制不同的 logger, 支持STDOUT / STDERR / FILE / DB多种类型;
- 新增Helm Chart库
  - - 新增 Helm Chart 管理,
  - 通用搜索结果中包含 Helm Chart 的相关结果;
  - 支持给 Helm Chart 打标签;
  - 删除指定的 Helm Chart (删除指定 Chart 下所有的版本)

## 服务器的安装环境需求

硬件要求:

Resource	Capacity	Description
CPU	minimal 2 CPU	4 CPU is preferred

Mem	minimal 4GB	8GB is preferred
Disk	minimal 40GB	160GB is preferred

## 软件要求：

Software	Version	Description
Python	version 2.7 or higher	Note that you may have to install Python on Linux distributions (Gentoo, Arch) that do not come with a Python interpreter installed by default
Docker engine	version 1.10 or higher	For installation instructions, please refer to: <a href="https://docs.docker.com/engine/installation/">https://docs.docker.com/engine/installation/</a>
Docker Compose	version 1.6.0 or higher	For installation instructions, please refer to: <a href="https://docs.docker.com/compose/install/">https://docs.docker.com/compose/install/</a>
Openssl	latest is preferred	Generate certificate and keys for Harbor

## 网络端口要求：

Port	Protocol	Description
443	HTTPS	Harbor portal and core API will accept requests on this port for https protocol
4443	HTTPS	Connections to the Docker Content Trust service for Harbor, only needed when Notary is enabled
80	HTTP	Harbor portal and core API will accept requests on this port for http protocol

## 安装Harbor 1.7

- 下载Harbor 1.7  
<https://github.com/goharbor/harbor/releases>
- 解压修改并修改配置

```

#
$ tar vf harbor-offline-installer-v1.7.1.tgz

$ cd harbor
total 590236
drwxr-xr-x. 3 root root      23 Jan 29 21:26 common
-rw-r--r--. 1 root root    946 Jan 29 21:38
docker-compose.chartmuseum.yml
-rw-r--r--. 1 root root    975 Jan 29 21:36 docker-compose.clair.yml
-rw-r--r--. 1 root root   1434 Jan  4 06:23 docker-compose.notary.yml
-rw-r--r--. 1 root root   5692 Jan 29 21:39 docker-compose.yml
-rw-r--r--. 1 root root   8040 Jan 29 21:30 harbor.cfg
-rw-r--r--. 1 root root 603562385 Jan  4 06:24 harbor.v1.7.1.tar.gz
-rwxr-xr-x. 1 root root   5739 Jan  4 06:23 install.sh
-rw-r--r--. 1 root root  11347 Jan  4 06:23 LICENSE
-rw-r--r--. 1 root root 748160 Jan  4 06:23 open_source_license
-rwxr-xr-x. 1 root root  36351 Jan 29 21:46 prepare

#
## harbor.cfg
sed -i "s#hostname = reg.mydomain.com#hostname = registry.odc.sunline.cn#g"
harbor.cfg      ## IP
sed -i "s#ui_url_protocol = http#ui_url_protocol = https#g" harbor.cfg
## httpshttps
sed -i "s#ssl_cert = /data/cert/server.crt#ssl_cert =
/data/harbor/cert/sunline.com.crt#g" harbor.cfg  ##
sed -i "s#ssl_cert_key = /data/cert/server.key#ssl_cert_key =
/data/harbor/cert/sunline.com.key#g" harbor.cfg
sed -i "s#secretkey_path = /data#secretkey_path = /data/harbor#g"
harbor.cfg

##
sed -i "s#/data/registry#/data/harbor/registry#g" docker-compose.yml
sed -i "s#/data/ca_download#/data/harbor/ca_download#g" docker-compose.yml
sed -i "s#/data/config#/data/harbor/config#g" docker-compose.yml
sed -i "s#/data/database#/data/harbor/database#g" docker-compose.yml
sed -i "s#/data/job_logs#/data/harbor/job_logs#g" docker-compose.yml
sed -i "s#/data/psc#/data/harbor/psc#g" docker-compose.yml
sed -i "s#/data/redis#/data/harbor/redis#g" docker-compose.yml
sed -i "s#/data/registry#/data/harbor/registry#g" docker-compose.yml
sed -i "s#/data/secretkey#/data/harbor/secretkey#g" docker-compose.yml
sed -i "s#/data/clair-db#/data/harbor/clair-db#g" docker-compose.clair.yml
sed -i "s#/data/notary-db#/data/harbor/notary-db#g" docker-compose.yml
sed -i "s#/data/:/data/:z#/data/harbor/:/data/:z#g" docker-compose.yml
sed -i "s#/data/chart_storage#/data/harbor/chart_storage#g"
docker-compose.chartmuseum.yml
sed -i "s#/data#/data/harbor#" prepare

```

- 安装Harbor

```
./install.sh --with-notary --with-clair --with-chartmuseum
#
HarborNotary--with-notaryharbor.cfgui_url_protocol/ssl_cert/ssl_cert_
keyhttps
# Clair in Harbour--with-clair
# HarborChartmuseum--with-chartmuseum
```

## 常用命令

```
docker-compose start ## Harbor
docker-compose stop  ## Harbor
docker-compose restart ## Harbor
docker-compose ps    ##
docker-compose create ##
docker-compose down   ## networkimagesvolumes
docker-compose log    ##
docker-compose up     ##
```

## 更新Harbor配置

```
docker-compose down -v
vim harbor.cfg    ##
./prepare
docker-compose up -d
```

参考: [https://github.com/goharbor/harbor/blob/master/docs/installation\\_guide.md](https://github.com/goharbor/harbor/blob/master/docs/installation_guide.md)