小伙伴们，想不想自己建立自己的 docker 私库。本篇介绍了什么是 Harbor，如何安装。来，动起手来。

本篇介绍了使用官方在线/离线安装包，进行 Harbor 的安装的过程。

## ■ 什么是 Harbor?

Harbor——Manage and serve container images in a secure environment.

这是关于 Harbor 的官方介绍，用来管理与服务容器，说白了就是私仓。

Harbor 是一个开源的云容器仓库，用于存储、签名和扫描容器映像以查找漏洞。

Harbor 通过提供信任、合规性、性能和互操作性来解决容器的共性问题。它填补了无法使用公有云、基于云容器或者是想获得跨云的一致体验的企业们和应用程序之间的空白。

链接：Harbor 官网（httprbos://gohar.io/）

## ■ 安装 Harbor

### 安装方法简介

- 在线安装方式

  主要是从 Docker Hub 下载 Harbor 镜像，所以这种安装方式，对于空间的要求非常小。

- 离线安装方式

  当没有网络的时候，可以使用这种安装方法，离线包里面有预打包的容器，所以比较大。

- Kubernetes 安装方式

需要 Kubernetes v1.6.5 and Harbor v1.2.0，参考：Integration with Kubernetes（https://github.com/goharbor/harbor/blob/master/docs/kubernetes_deployment.md）

## ■ 硬件环境

| 资源 | 最小配置 | 描述 |
| --- | --- | --- |
| CPU | >2 CPU | 4 CPU 最好 |
| Mem(内存) | >4GB | 8GB 最好 |
| Disk(硬盘) | >40GB | 160GB 最好 |

## ■ 软件环境

| 软件环境 | 版本 | 描述 |
| --- | --- | --- |
| Python | >2.7 | 有一些 Linux 发行版是没有默认安装 Python 的，需要自己手动安装 |
| Docker engine | >1.10 | 安装 Docker，官方教程：Install Docker CE（https://docs.docker.com/engine/installation/） |
| Docker Compose | >1.6.0 | 请参考安装教程：官方教程：Install Docker Compose（https://docs.docker.com/compose/install/） |
| Openssl | lastest | 用于产生证书与密钥(一般 Linux 发行版会默认安装) |

## ■ 网络端口

| 端口 | 协议 | 描述 |
| --- | --- | --- |
| 443 | HTTPS | Harbor portal 和 core API 将会使用 443 接口用于 HTTPS 协议 |
| 4443 | HTTPS | 当 Notray 使用时，Harbor 使用这个端口用于 Docker 中可信任内容的传输。 |
| 80 | HTTP | Harbor portal 和 core API 将会使用 80 接口用于 HTTP 协议 |

## 安装步骤

1.基本的安装步骤

- 下载安装包，在 release 页面进行下载
- 配置 harbor.cfg
- 运行 install.sh

2.使用如下命令进行解压

- Online installer:

  复制

  $ tar xvf harbor-online-installer-<version>.tgz

- Offline installer:

  复制

  $ tar xvf harbor-offline-installer-<version>.tgz

3.配置 Harbor

- Required parameters
- Optional parameters

**其中，Required parameters 有：**

- hostname：使用 ip 或者域名，切忌使用 localhost 或者 127.0.0.1
- ui*url*protocol：默认为 http 协议（可选 https 或者 http），如果需要配置 SSL 证书，请参考 HTTPS 配置（https://github.com/goharbor/harbor/blob/master/docs/configure_https.md）
- db_password：数据库 PostgreSQL 的 root 密码
- max*job*workers：默认为 10，根据现有的任务数量需要进行更改，每个任务会占用 network/CPU/IO 资源，所以分配了之后注意观察。
- customize*crt：默认为 on，可以设置成 off。harbor 会自动生成证书与密钥对，如果需要自己生成，请参考：[Customize Harbor token service with your key and certificate](https://github.com/goharbor/harbor/blob/master/docs/customize*token_service.md)
- ssl_cert：SSL 证书路径，只有当设置成 https 生效。
- ssl*cert*key：SSL 密钥，同上。
- secretkey_path：用于加密或者解密远程仓库的密钥路径(去访问其他仓库使用的密钥)
- log*rotate*count：如果设置成 0，则老旧的日志文件就会被清除掉，而不是进行滚动增加。
- log*rotate*size：滚动日志的大小，默认单位是 kb，可以设置成 100M 或者是 100G
- http_proxy：http 代理路径

- https_proxy：https 代理
- no_proxy：不需要代理的地址或者域名，如：127.0.0.1

使用 vi 命令对 harbor.cfg 文件进行编辑

**其中，Optional parameters 有：**

- Email settings

  参考如下配置：

  - email_server = smtp.mydomain.com
  - email*server*port = 25
  - email_identity =
  - email*username = [sample*admin@mydomain.com](mailto:sample_admin@mydomain.com)
  - email_password = abc
  - email*from = admin [sample*admin@mydomain.com](mailto:sample_admin@mydomain.com)
  - email_ssl = false
  - email_insecure = false

- harbor*admin*password：管理员密码
- auth*mode：默认是`db*auth，可选 ldap*auth 或者 db*auth`

重要提示：当从已有的 Harbor 进行升级时，要确保 auth_mode 与 harbor.cfg 中的配置是一样的。

- ldap*url：LDAP 入口地址：e.g.ldaps://ldap.mydomain.com，只有`ldap*auth`模式下才有效

- ldap_searchdn：DN 用户，e.g. uid=admin,ou=people,dc=mydomain,dc=com
- ldap*search*pwd：搜索用户的密码
- ldap_basedn：基础用户，e.g. ou=people,dc=mydomain,dc=com
- ldap_filter：搜索过滤，e.g. (objectClass=person)
- ldap_uid：用于在搜索的时候，对用于进行匹配：可以是 uid, cn, email 或者是其他属性。
- ldap*scope：搜索的范围：0-LDAP*SCOPE*BASE, 1-LDAP*SCOPE*ONELEVEL, 2-LDAP*SCOPE_SUBTREE. 默认是 2.
- ldap_timeout：超时时长，默认是 5.
- ldap*verify*cert：是否从 LDAP 服务器进行认证，默认是 true.
- ldap*group*basedn：基础的搜索组， e.g. ou=group,dc=mydomain,dc=com
- ldap*group*filter：组过滤
- ldap*group*gid：组属性，如 cn, name
- ldap*group*scope：搜索的范围：0-LDAP*SCOPE*BASE, 1-LDAP*SCOPE*ONELEVEL, 2-LDAP*SCOPE*SUBTREE.默认是 2.
- self*registration：(**on** or **off**. Default is **on**) 是否允许自注册，一般来说`auth*mode 被设置成 ldap_auth`模式的时候，自注册是被关闭的。
- token_expiration：默认的 token 过期时长，默认 30 分钟。
- project*creation*restriction：设置哪些用户可以创建项目，默认是允许所有的用户进行创建，当设置成 adminonly 时，只允许管理员进行创建。

以下是示例配置：

复制

```
## Configuration file of Harbor
## Configuration file of Harbor

#This attribute is for migrator to detect the version of the .cfg file, DO NOT
 MODIFY!
_version = 1.5.0
#The IP address or hostname to access admin UI and registry service.
#DO NOT use localhost or 127.0.0.1, because Harbor needs to be accessed
 by external clients.
hostname = test.toimc.cn

#The protocol for accessing the UI and token/notification service, by defau
lt it is http.
#It can be set to https if ssl is enabled on nginx.
ui_url_protocol = https

#Maximum number of job workers in job service
max_job_workers = 50

#Determine whether or not to generate certificate for the registry's token.
#If the value is on, the prepare script creates new root cert and private key
#for generating token to access the registry. If the value is off the default k
ey/cert will be used.
#This flag also controls the creation of the notary signer's cert.
customize_crt = off

#The path of cert and key files for nginx, they are applied only the protocol
 is set to https
```

```
ssl_cert = /home/ssh/fullchain.cer
ssl_cert_key = /home/ssh/toimc.cn.key

#The path of secretkey storage
secretkey_path = /data

#Admiral's url, comment this attribute, or set its value to NA when Harbor i
s standalone
admiral_url = NA

#Log files are rotated log_rotate_count times before being removed. If cou
nt is 0, old versions are removed rather than rotated.
log_rotate_count = 50
#Log files are rotated only if they grow bigger than log_rotate_size bytes. If
 size is followed by k, the size is assumed to be in kilobytes.
#If the M is used, the size is in megabytes, and if G is used, the size is in gi
gabytes. So size 100, size 100k, size 100M and size 100G
#are all valid.
log_rotate_size = 200M

#Config http proxy for Clair, e.g. http://my.proxy.com:3128
#Clair doesn't need to connect to harbor ui container via http proxy.
http_proxy = socks5://192.168.4.250:2080
https_proxy = socks5://192.168.4.250:2080
no_proxy = 127.0.0.1,localhost,ui

#NOTES: The properties between BEGIN INITIAL PROPERTIES and END INIT
IAL PROPERTIES
#only take effect in the first boot, the subsequent changes of these proper
ties
#should be performed on web ui
```

#************************BEGIN INITIAL PROPERTIES************************

#Email account settings for sending out password resetting emails.

#Email server uses the given username and password to authenticate on T
LS connections to host and act as identity.
#Identity left blank to act as username.
email_identity =

email_server = email-smtp.us-east-1.amazonaws.com
email_server_port = 587
email_username = AKIAJ3A3D3R5NMUUEBJA
email_password = Amz5jvLaeq13VLk5j4QtmTRJhZ/AnxEnmuHUB4DsQ+op
email_from = liwei <lw96@live.com>
email_ssl = true
email_insecure = false

##The initial password of Harbor admin, only works for the first time when
Harbor starts.
#It has no effect after the first launch of Harbor.
#Change the admin password from UI after launching Harbor.
harbor_admin_password = 2239hOb2tVgOSwoW123FDs23vs324hmuy766
7ghfbv32FGDS43

##By default the auth mode is db_auth, i.e. the credentials are stored in a l
ocal database.
#Set it to ldap_auth if you want to verify a user's credentials against an LD
AP server.
auth_mode = db_auth

```
#The url for an ldap endpoint.
ldap_url = ldaps://ldap.mydomain.com

#A user's DN who has the permission to search the LDAP/AD server.
#If your LDAP/AD server does not support anonymous search, you should
configure this DN and ldap_search_pwd.
#ldap_searchdn = uid=searchuser,ou=people,dc=mydomain,dc=com

#the password of the ldap_searchdn
#ldap_search_pwd = password

#The base DN from which to look up a user in LDAP/AD
ldap_basedn = ou=people,dc=mydomain,dc=com

#Search filter for LDAP/AD, make sure the syntax of the filter is correct.
#ldap_filter = (objectClass=person)

# The attribute used in a search to match a user, it could be uid, cn, email,
sAMAccountName or other attributes depending on your LDAP/AD
ldap_uid = uid

#the scope to search for users, 0-LDAP_SCOPE_BASE, 1-LDAP_SCOPE_ONE
LEVEL, 2-LDAP_SCOPE_SUBTREE
ldap_scope = 2

#Timeout (in seconds)  when connecting to an LDAP Server. The default val
ue (and most reasonable) is 5 seconds.
ldap_timeout = 5

#Verify certificate from LDAP server
ldap_verify_cert = true
```

```
#The base dn from which to lookup a group in LDAP/AD
ldap_group_basedn = ou=group,dc=mydomain,dc=com

#filter to search LDAP/AD group
ldap_group_filter = objectclass=group

#The attribute used to name a LDAP/AD group, it could be cn, name
ldap_group_gid = cn

#The scope to search for ldap groups. 0-LDAP_SCOPE_BASE, 1-LDAP_SCOP
E_ONELEVEL, 2-LDAP_SCOPE_SUBTREE
ldap_group_scope = 2

#Turn on or off the self-registration feature
self_registration = on

#The expiration time (in minute) of token created by token service, default
is 30 minutes
token_expiration = 30

#The flag to control what users have permission to create projects
#The default value "everyone" allows everyone to creates a project.
#Set to "adminonly" so that only admin user can create project.
project_creation_restriction = everyone

#**********************END INITIAL PROPERTIES**********************
###### #Harbor DB configuration section#######

# The address of the Harbor database. Only need to change when using ex
ternal db.
```

db_host = mysql

# The password for the root user of Harbor DB. Change this before any pro
duction use.

db_password = root123

# The port of Harbor database host

db_port = 3306

# The user name of Harbor database

db_user = root

##### End of Harbor DB configuration#######

# The redis server address. Only needed in HA installation.

# address:port[,weight,password,db_index]

redis_url = redis:6379

###### ####Clair DB configuration############

# Clair DB host address. Only change it when using an exteral DB.

clair_db_host = postgres

# The password of the Clair's postgres database. Only effective when Harb

or is deployed with Clair.

# Please update it before deployment. Subsequent update will cause Clair's API server and Harbor unable to access Clair's database.

clair_db_password = password

# Clair DB connect port

clair_db_port = 5432

# Clair DB username

clair_db_username = postgres

# Clair default database

clair_db = postgres

###### ####End of Clair DB configuration############

# The following attributes only need to be set when auth mode is uaa_auth

uaa_endpoint = uaa.mydomain.org
uaa_clientid = id
uaa_clientsecret = secret
uaa_verify_cert = true
uaa_ca_cert = /path/to/ca.pem

### Docker Registry setting

# registry_storage_provider can be: filesystem, s3, gcs, azure, etc.

registry_storage_provider_name = filesystem

# registry_storage_provider_config is a comma separated "key: value" pairs, e.g. "key1: value, key2: value2".

# Refer to https://docs.docker.com/registry/configuration/#storage for all available configuration.

registry_storage_provider_config =

4.配置完成之后，在当前的解压缩的目录下，运行 sudo ./install.sh。

是否可以在解压缩的目录下找到相应的文件 install.sh？如果找不到，可以使用 find 命令或者是 ls 命令。

安装完成之后，就可以访问之前设置的域名或者 IP 地址了，比如上例中的 test.toimc.com。

## ■ 常见问题

1. 如何推送镜像？

   复制

   docker login test.toimc.com
   docker push test.toimc.com/myproject/myrepo:mytag

2. 如何申请证书，如何设置 SSL？

   有两个先决条件：(1)需要有一个域名； (2)使用 acme 或者 caddy 这种服务进行申请证书。

3. 安装 Notary, 安装 Clair, 安装 chart repository service

安装 Notary，使用如下命令：sudo ./install.sh --with-notary

安装 Clair，使用如下命令：sudo ./install.sh --with-clair

安装 Chart repository service，使用如下命令：sudo ./install.sh --with-chartmuseum

PS： 可以这样使用：sudo ./install.sh --with-notary --with-clair --with-chartmuseum

4. Harbor 生命周期的管理：

直接使用 docker-compose 管理命令进行管理(PS:需要 cd 到之前的 Harbor 的解压目录，即 install 目录)

停止 Harbor:

复制

```
$ sudo docker-compose stop
Stopping nginx             ... done
Stopping harbor-portal     ... done
Stopping harbor-jobservice ... done
Stopping harbor-core       ... done
Stopping registry          ... done
Stopping redis             ... done
Stopping registryctl       ... done
Stopping harbor-db         ... done
Stopping harbor-log        ... done
```

重启 Harbor：

复制

```
$ sudo docker-compose start
Starting log         ... done
Starting registry    ... done
```

```
Starting registryctl ... done
Starting postgresql  ... done
Starting core        ... done
Starting portal      ... done
Starting redis       ... done
Starting jobservice  ... done
Starting proxy       ... done
```

当修改了 Harbor 的配置文件后，需要使用如下的方式更新镜像：

复制

```
$ sudo docker-compose down -v
$ vim harbor.cfg
$ sudo prepare
$ sudo docker-compose up -d
```

删除 Harbor 镜像，但是保留文件系统的做法：

复制

```
$ sudo docker-compose down -v
```

需要全部删除时(包括镜像、文件、仓库)——重装需要/卸载需要

复制

```
$ rm -r /data/database
$ rm -r /data/registry
```

5. 日志文件路径：/var/log/harbor
6. 自定义端口：

   – 修改 docker-compose.yml 修改”80”端口到一个指定的用户端口, e.g. 8888:80.

复制

```
    proxy:
image: goharbor/nginx-photon:v1.6.0
    container_name: nginx
    restart: always
volumes:
        - ./common/config/nginx:/etc/nginx:z
    ports:
        - 8888:80
        - 443:443
    depends_on:
        - postgresql
        - registry
        - core
        - portal
        - log
    logging:
        driver: "syslog"
    options:
    syslog-address: "tcp://127.0.0.1:1514"
    tag: "proxy"
```

– 修改 harbor.cfg 文件中的 hostname 属性：

复制

```
hostname = 192.168.0.2:8888
```

– 参考上面的管理 harbor 的生命周期的内容，对 Harbor 进行更新。

**同理，对于 HTTPS 协议，参考：**

– 打开 HTTPS 协议，并配置 SSL 证书：参考 guide.

– 修改 docker-compose.yml 修改”443” 端口为用户自定义端口, e.g. 8888:443.

复制

```
proxy:
image: goharbor/nginx-photon:v1.6.0
container_name: nginx
restart: always
volumes:
        - ./common/config/nginx:/etc/nginx:z
ports:
        - 80:80
        - 8888:443
depends_on:
        - postgresql
        - registry
        - core
        - portal
        - log
logging:
        driver: "syslog"
        options:
syslog-address: "tcp://127.0.0.1:1514"
tag: "proxy"
```

– 修改 harbor.cfg 文件中的 hostname 属性：

复制

```
hostname = 192.168.0.2:8888
```

– 参考上面的管理 harbor 的生命周期的内容，对 Harbor 进行更新。