

科： 硕士 ▾

harbour / docs / installation_guide.md

查找文件

复制路径

 stonezjd 删除adminserver

0cba36d的 所有内容

on 22 Feb

22个贡献者 

409行 (332 sloc) 23.2 KB

原始 责备 历史   

安装和配置指南

港口可以通过以下三种方法之一安装：

- 在线安装程序：安装程序从Docker hub下载Harbor的图像。因此，安装程序的尺寸非常小。
- 脱机安装程序：当主机没有Internet连接时使用此安装程序。安装程序包含预先构建的图像，因此其大小更大。

所有安装程序都可以从[官方发布](#)页面下载。

本指南介绍使用联机或脱机安装程序安装和配置Harbor的步骤。安装过程几乎相同。

如果运行以前版本的Harbor，则可能需要更新 `harbor.cfg` 和 迁移数据以适应新的数据库架构。有关更多详细信息，请参阅“[港口迁移指南](#)”。

此外，社区还创建了Kubernetes的部署说明。有关详细信息，请参阅[Harbour on Kubernetes](#)。

目标主机的先决条件

Harbor被部署为多个Docker容器，因此可以部署在任何支持Docker的Linux发行版上。目标主机需要安装Python，Docker和Docker Compose。

硬件

资源	容量	描述
中央处理器	最小2 CPU	4 CPU是首选
纪念品	最小4GB	8GB是首选
磁盘	最小40GB	160GB是首选

软件

软件	版	描述
蟒蛇	2.7或更高版本	请注意，您可能必须在Linux发行版（Gentoo，Arch）上安装Python，默认情况下不安装Python解释器
Docker引擎	版本1.10或更高版本	有关安装说明，请参阅： https://docs.docker.com/engine/installation/
Docker撰写	版本1.6.0或更高版本	有关安装说明，请访问： https://docs.docker.com/compose/install/

软件	版	描述
OpenSSL 的	最新的是首选	为Harbor生成证书和密钥

网络端口

港口	协议	描述
443	HTTPS	港口门户和核心API将接受此端口上的https协议请求
4443	HTTPS	只有在启用“公证”时才需要连接到Dock的Docker Content Trust服务
80	HTTP	Harbor端口和核心API将接受此端口上的http协议请求

安装步骤

安装步骤归结为以下内容

1. 下载安装程序;
2. 配置harbor.cfg ;
3. 运行install.sh安装并启动Harbor;

下载安装程序：

可以从[发布](#)页面下载安装程序的二进制文件。选择在线或离线安装程序。使用tar命令提取包。

在线安装：

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

脱机安装：

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

配置港口

配置参数位于文件harbor.cfg中。

在harbor.cfg中有两类参数，必需参数和可选参数。

- 必需参数：需要在配置文件中设置这些参数。如果用户更新它们 harbor.cfg 并运行 install.sh 脚本以重新安装 Harbor，它们将生效。
- 可选参数：这些参数对于更新是可选的，即用户可以将它们保留为默认值，并在启动Harbour后在Web Portal上更新它们。如果它们已经启用 harbor.cfg ，它们只会在首次启动Harbour时生效。对这些参数的后续更新 harbor.cfg 将被忽略。

注意：如果您选择通过Portal设置这些参数，请务必在Harbour启动后立即执行此操作。特别是，您必须在Harbour中注册或创建任何新用户之前设置所需的auth_mode。当系统中有用户时（除默认管理员用户外），无法更改auth_mode。

参数如下所述 - 请注意，至少需要更改hostname属性。

必需参数：

- hostname：目标主机的主机名，用于访问Portal和注册表服务。它应该是目标计算机的IP地址或完全限定的域名（FQDN），例如， 192.168.1.10 或 reg.yourdomain.com 。不要使用 localhost 或 127.0.0.1 作为主机名 - 外部客户端需要访问注册表服务！
- ui_url_protocol :(http或https。默认为http) 用于访问Portal和令牌/通知服务的协议。如果启用了公证，则此参数必须为https。默认情况下，这是http。要设置https协议，请参阅[使用HTTPS访问配置Harbor](#)。
- db_password：用于db_auth的PostgreSQL数据库的root密码。更改此密码以用于任何生产用途！

- `max_job_workers` : (默认值为10) 作业服务中的最大复制工作数。对于每个映像复制作业，工作程序将存储库的所有标记同步到远程目标。增加此数量可以在系统中执行更多并发复制作业。但是，由于每个工作者都消耗一定量的网络/ CPU / IO资源，请根据主机的硬件资源仔细选择该属性的值。
- `customize_cert`: (上或关闭，默认为上)，如果此属性上，在准备脚本创建注册表的令牌生成/验证私钥和根证书。当外部源提供密钥和根证书时，将此属性设置为off。有关详细信息，请参阅[自定义密钥和港口令牌服务证书](#)。
- `ssl_cert`: SSL证书的路径，仅在协议设置为https时应用。
- `ssl_cert_key`: SSL密钥的路径，仅在协议设置为https时应用。
- `secretkey_path`: 用于加密或解密复制策略中远程注册表密码的密钥路径。
- `log_rotate_count`: 日志文件在被删除之前会被轮换`log_rotate_count`次。如果count为0，则删除旧版本而不是旋转。
- `log_rotate_size`: 仅当日志文件大于`log_rotate_size`字节时才会轮换日志文件。如果大小后跟k，则假定大小以千字节为单位。如果使用M，则大小以兆字节为单位，如果使用G，则大小以千兆字节为单位。尺寸100，尺寸100k，尺寸100M和尺寸100G都是有效的。
- `http_proxy`: 为Clair配置http代理，例如 `http://my.proxy.com:3128` 。
- `https_proxy`: 为Clair配置https代理，例如 `http://my.proxy.com:3128` 。
- `no_proxy`: 没有为Clair配置代理，例如 `127.0.0.1,localhost,core,registry` 。

可选参数

- 电子邮件设置: Harbor需要这些参数才能向用户发送“密码重置”电子邮件，并且仅在需要该功能时才需要。另外，请注意，在默认情况下SSL连接时没有启用-如果你的SMTP服务器需要SSL，但不支持STARTTLS，那么你应该通过设置启用SSL `email_ssl = TRUE`。如果电子邮件服务器使用自签名证书或不受信任证书，则设置`email_insecure = true`。有关“email_identity”的详细说明，请参阅[rfc2595](#)

- `email_server = smtp.mydomain.com`
- `email_server_port = 25`
- `email_identity =`
- `email_username = sample_admin@mydomain.com`
- `email_password = abc`
- `email_from = admin sample_admin@mydomain.com`
- `email_ssl = false`
- `email_insecure = false`

- `harbor_admin_password`: 管理员的初始密码。此密码仅在Harbor首次启动时生效。之后，将忽略此设置，并且应在Portal中设置管理员密码。请注意，默认用户名/密码为`admin / Harbor12345`。
- `auth_mode`: 使用的身份验证类型。默认情况下，它是`db_auth`，即凭据存储在数据库中。对于LDAP身份验证，请将其设置为`ldap_auth`。

重要信息: 从现有Harbor实例升级时，必须确保在启动新版本的Harbor之前`auth_mode`相同 `harbor.cfg` 。否则，用户可能无法在升级后登录。

- `ldap_url`: LDAP端点URL (例如 `ldaps://ldap.mydomain.com`)。仅在`auth_mode`设置为`ldap_auth`时使用。
- `ldap_searchdn`: 具有搜索LDAP / AD服务器权限的用户的DN (例如 `uid=admin,ou=people,dc=mydomain,dc=com`)。
- `ldap_search_pwd`: `ldap_searchdn`指定的用户密码。
- `ldap_basedn`: 查找用户的基本DN，例如 `ou=people,dc=mydomain,dc=com` 。仅在`auth_mode`设置为`ldap_auth`时使用。
- `ldap_filter`: 用于查找用户的搜索过滤器，例如 `(objectClass=person)` 。
- `ldap_uid`: 用于在LDAP搜索期间匹配用户的属性，它可以是`uid`, `cn`, `email`或其他属性。
- `ldap_scope`: 搜索用户的范围，0-LDAP_SCOPE_BASE, 1-LDAP_SCOPE_ONELEVEL, 2-LDAP_SCOPE_SUBTREE。默认值为2。
- `ldap_timeout`: 连接到LDAP服务器时超时 (以秒为单位)。默认值为5。
- `ldap_verify_cert`: 验证来自LDAP服务器的证书。默认为true。
- `ldap_group_basedn`: 在LDAP / AD中查找组的基本dn，例如 `ou=group,dc=mydomain,dc=com` 。
- `ldap_group_filter`: 搜索LDAP / AD组的过滤器，例如 `objectclass=group` 。

- `ldap_group_gid`: 用于命名LDAP / AD组的属性，可以是cn, name。
- `ldap_group_scope`: 搜索ldap组的范围。0-LDAP_SCOPE_BASE, 1-LDAP_SCOPE_ONELEVEL, 2-LDAP_SCOPE_SUBTREE。默认值为2。
- `self_registration`: (打开或关闭。默认打开) 启用/禁用用户注册他/她自己的能力。禁用时，新用户只能由管理员用户创建，只有管理员用户可以在Harbor中创建新用户。注意：当`auth_mode`设置为`ldap_auth`时，始终禁用自注册功能，并忽略此标志。
- `token_expiration`: 令牌服务创建的令牌的到期时间（以分钟为单位），默认为30分钟。
- `project_creation_restriction`: 用于控制用户有权创建项目的标志。默认情况下，每个人都可以创建一个项目，设置为“adminonly”，这样只有管理员才能创建项目。

配置存储后端（可选）

默认情况下，Harbor将图像存储在本地文件系统中。在生产环境中，您可以考虑使用其他存储后端而不是本地文件系统，如S3, OpenStack Swift, Ceph等。这些参数是注册表的配置。

- `registry_storage_provider_name`: 注册表的存储提供程序名称，可以是filesystem, s3, gcs, azure等。默认为filesystem。
- `registry_storage_provider_config`: 存储提供程序配置的逗号分隔“key: value”对，例如“key1: value, key2: value2”。默认为空字符串。
- `registry_custom_ca_bundle`: 自定义根ca证书的路径，它将注入到注册表和图表存储库容器的信任库中。当用户使用自签名证书托管内部存储时，通常需要这样做。

例如，如果您使用Openstack Swift作为存储后端，则参数可能如下所示：

```
registry_storage_provider_name = swift
registry_storage_provider_config = "username: admin, password: ADMIN_PASS, authurl: http://keystone_addr:
```

注意：有关注册表存储后端的详细信息，请参阅“[注册表配置参考](#)”。

完成安装和启动港口

一旦`harbor.cfg`和存储后端（可选）配置，安装和使用的启动港 `install.sh` 脚本。请注意，在线安装程序可能需要一些时间才能从Docker hub下载Harbor映像。

默认安装（没有Notary / Clair）

Harbor已与Notary和Clair集成（用于漏洞扫描）。但是，默认安装不包括Notary或Clair服务。

```
$ sudo ./install.sh
```

如果一切正常，您应该能够打开浏览器访问<http://reg.yourdomain.com>上的管理门户（将`reg.yourdomain.com`更改为您配置的主机名 `harbor.cfg` ）。请注意，默认管理员用户名/密码为admin / Harbor12345。

登录管理门户并创建一个新项目，例如 `myproject` 。然后，您可以使用`docker`命令登录和推送映像（默认情况下，注册表服务器侦听端口80）：

```
$ docker login reg.yourdomain.com
$ docker push reg.yourdomain.com/myproject/myrepo:mytag
```

重要信息： Harbor的默认安装使用`HTTP` - 因此，您需要将该选项添加 `--insecure-registry` 到客户端的Docker守护程序并重新启动Docker服务。

使用公证人安装

要使用公证服务安装Harbour，请在运行时添加参数 `install.sh` ：

```
$ sudo ./install.sh --with-notary
```

注意：对于带有公证的安装，参数`ui_url_protocol`必须设置为“https”。有关配置HTTPS的信息，请参阅以下部分。

有关Notary和Docker Content Trust的更多信息，请参阅Docker的文档：https://docs.docker.com/engine/security/trust/content_trust/

用Clair安装

要使用Clair服务安装Harbour，请在运行时添加参数 `install.sh`：

```
$ sudo ./install.sh --with-clair
```

有关Clair的更多信息，请参阅Clair的文档：<https://coreos.com/clair/docs/2.0.1/>

使用图表存储库服务安装

要使用图表存储库服务安装Harbor，请在运行时添加参数 `install.sh`：

```
$ sudo ./install.sh --with-chartmuseum
```

注意：如果要安装Notary，Clair和图表存储库服务，则必须在同一命令中指定所有参数：

```
$ sudo ./install.sh --with-notary --with-clair --with-chartmuseum
```



当前

英语

网页，是否翻译为

中文

翻译

取消划词



使用HTTPS访问配置Harbor

Harbor不附带任何证书，默认情况下使用HTTP来处理请求。虽然这使得设置和运行相对简单 - 特别是对于开发或测试环境 - 但不建议用于生产环境。要启用HTTPS，请参阅[使用HTTPS访问配置Harbour](#)。

管理港湾的生命周期

您可以使用docker-compose来管理Harbor的生命周期。一些有用的命令如下所示（必须与`docker-compose.yml`在同一目录中运行）。

停止港口：

```
$ 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
```

停止后重启港口：

```
$ 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
```

要更改Harbour的配置，请先停止现有的Harbor实例并进行更新 `harbor.cfg`。然后运行 `prepare` 脚本以填充配置。最后重新创建并启动Harbor的实例：

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

```
$ sudo prepare
$ sudo docker-compose up -d
```

删除Harbor的容器，同时将图像数据和Harbor的数据库文件保存在文件系统中：

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

删除Harbor的数据库和图像数据（用于干净的重新安装）：

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

在与Notary一起安装Harbour时管理Harbour的生命周期

当Harbour与Notary一起安装时， docker-compose.notary.yml docker-compose命令需要一个额外的模板文件。用于管理Harbor生命周期的docker-compose命令是：

```
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.notary.yml [ up|down|ps|stop|start ]
```

例如，如果要在 harbor.cfg 使用Notary安装Harbor时更改配置并重新部署Harbour，则应使用以下命令：



网页翻译

当前

英语

网页，是否翻译为

中文

翻译

取消划词



```
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.notary.yml down -v
$ vim harbor.cfg
$ sudo prepare --with-notary
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.notary.yml up -d
```

在与Clair一起安装时管理Harbour的生命周期

使用Clair安装Harbour时 docker-compose.clair.yml， docker-compose命令需要额外的模板文件。用于管理Harbor生命周期的docker-compose命令是：

```
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.clair.yml [ up|down|ps|stop|start ]
```

例如，如果要在 harbor.cfg 使用Clair安装Harb时更改配置并重新部署Harbour，则应使用以下命令：

```
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.clair.yml down -v
$ vim harbor.cfg
$ sudo prepare --with-clair
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.clair.yml up -d
```

在安装图表存储库服务时管理Harbor的生命周期

当Harbor安装了图表存储库服务时， docker-compose.chartmuseum.yml docker-compose命令需要一个额外的模板文件。用于管理Harbor生命周期的docker-compose命令是：

```
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.chartmuseum.yml [ up|down|ps|stop|start ]
```

例如，如果要在 harbor.cfg 安装了图表存储库服务时更改配置并重新部署Harbor，则应使用以下命令：

```
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.chartmuseum.yml down -v
$ vim harbor.cfg
$ sudo prepare --with-chartmuseum
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.chartmuseum.yml up -d
```

使用Notary，Clair和图表存储库服务安装Harbour的生命周期

如果要同时安装Notary，Clair和图表存储库服务，则应在docker-compose和prepare命令中包含所有组件：

```
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.notary.yml -f ./docker-compose.clair.yml --
$ vim harbor.cfg
$ sudo prepare --with-notary --with-clair --with-chartmuseum
$ sudo docker-compose -f ./docker-compose.yml -f ./docker-compose.notary.yml -f ./docker-compose.clair.yml --
```

有关docker-compose的更多信息，请查看[Docker Compose命令行参考](#)。

持久数据和日志文件

默认情况下，注册表数据保留在主机上的 `/data/` 目录中。即使移除和/或重新创建Harbor的容器，此数据仍保持不变。

此外，Harbor使用rsyslog来收集每个容器的日志。默认情况下，这些日志文件存储在 `/var/log/harbor/` 目标主机上的目录中以进行故障排除。

配置Harbor侦听自定义端口

默认情况下，Harbor侦听管理门户和docker命令的端口80（HTTP）和443（HTTPS，如果已配置），您可以使用自定义命令对其进行配置。

对于HTTP协议



当前

英语

网页，是否翻译为

中文

翻译

取消划词



将第一个“80”替换为自定义端口，例如8888：80。

```
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"
```

2.Modify harbor.cfg，将端口添加到参数“hostname”

```
hostname = 192.168.0.2:8888
```

3.重新部署港口，参考上一节“管理港口的生命周期”。

对于HTTPS协议

1.按照本[指南](#)在Harbor中启用HTTPS。

2.Modify docker-compose.yml

将第一个“443”替换为自定义端口，例如8888：443。

```
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"
```

3.Modify harbor.cfg，将端口添加到参数“hostname”

```
hostname = 192.168.0.2:8888
```

4.重新部署港口，参考上一节“管理港口的生命周期”。

性能调整

默认情况下，Harbor将Clair容器的CPU使用率限制为150000，并避免耗尽所有CPU资源。这在docker-compose.clair.yml文件中定义。您可以根据硬件配置对其进行修改。

 网页翻译

当前

英语

 网页，是否翻译为

中文

翻译 取消划词



故障排除

1. 当Harbor无法正常工作时，请运行以下命令以查明Harbor的所有容器是否处于UP状态：

\$ sudo docker-compose ps

Name	Command	State	Ports
harbor-core	/harbor/start.sh	Up	
harbor-db	/entrypoint.sh postgres	Up	5432/tcp
harbor-jobservice	/harbor/start.sh	Up	
harbor-log	/bin/sh -c /usr/local/bin/ ...	Up	127.0.0.1:1514->10514/tcp
harbor-portal	nginx -g daemon off;	Up	80/tcp
nginx	nginx -g daemon off;	Up	0.0.0.0:443->443/tcp, 0.0.0.0:4443->4443/tcp, 0.0.0.0:80->80/tcp
redis	docker-entrypoint.sh redis ...	Up	6379/tcp
registry	/entrypoint.sh /etc/regist ...	Up	5000/tcp
registryctl	/harbor/start.sh	Up	

如果容器不是UP状态，检查目录容器的日志文件 /var/log/harbor 。例如，如果容器 harbor-core 未运行，则应查看日志文件 core.log 。

2，当设置港湾后面的nginx的代理或弹性负载均衡，寻找线下，在 common/templates/nginx/nginx.http.conf 从部分删除它，如果代理已经有类似的设置： location / ， location /v2/ 和 location /service/ 。

```
proxy_set_header X-Forwarded-Proto $scheme;
```

并重新部署Harbour参考上一节“管理港口的生命周期”。