

文章

[Michael Lei](#) · 四月 24, 2022 阅读大约需 7 分钟

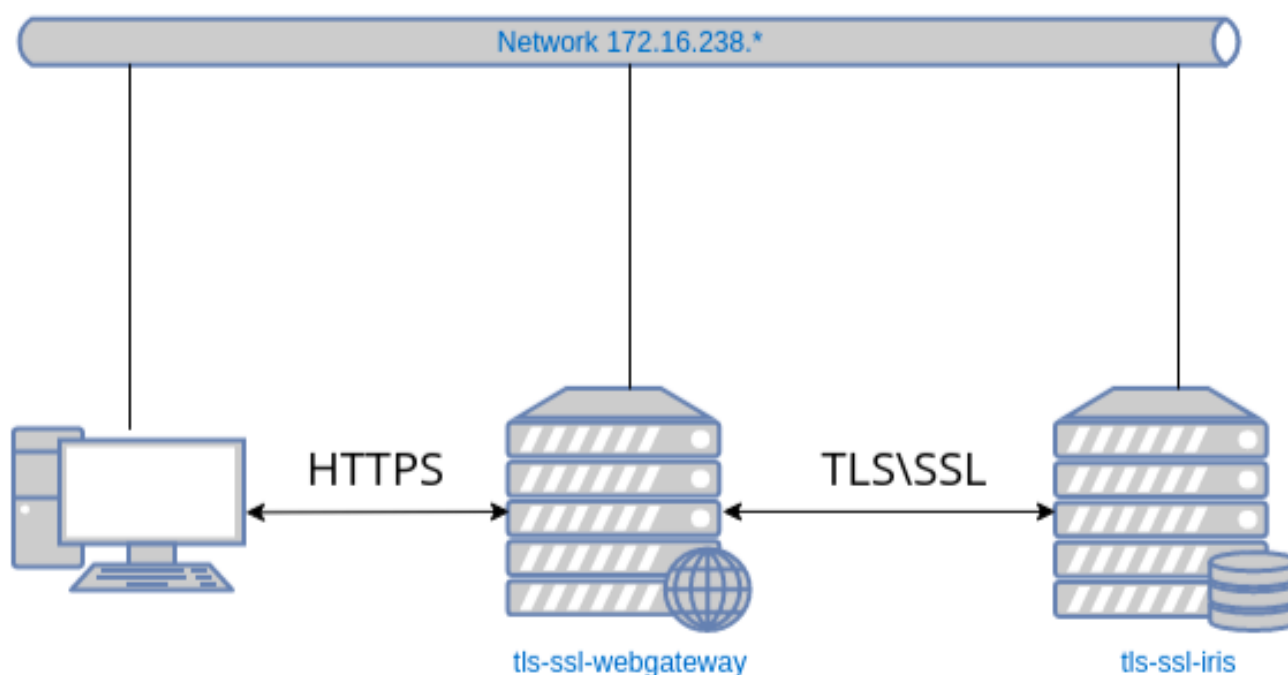
基于Docker的Apache Web Gateway

基于Docker的Apache Web Gateway

Hi 社区

在本文中，我们将基于Docker程序化地配置一个Apache Web Gateway，使用。：

- HTTPS protocol.
- TLS\SSL to secure the communication between the Web Gateway and the IRIS instance.



我们将使用两个镜像：一个用于Web网关，第二个用于IRIS实例。

所有必需的文件都在这 [GitHub repository](#).

我们从git clone开始:

```
git clone https://github.com/lscalese/docker-webgateway-sample.git
cd docker-webgateway-sample
```

准备系统

为了避免权限方面的问题，你的系统需要一个用户和一个组:

- www-data

- irisowner

需要与容器共享证书文件。如果你的系统中不存在这些文件，只需执行：

```
sudo useradd --uid 51773 --user-group irisowner
sudo groupmod --gid 51773 irisowner
sudo useradd -user-group www-data
```

生成证书

在这个示例中，我们使用以下三个证书：

1. HTTPS web server usage.
2. TLS\SSL encryption on Web Gateway client.
3. TLS\SSL encryption on IRIS Instance.

有一个随时可用的脚本来生成它们。 .

然而，你应该自定义证书的主题；只需编辑这个文件 [gen-certificates.sh](#) .

这是 OpenSSL subj argument的结构：

1. **C**: Country code
2. **ST**: State
3. **L**: Location
4. **O**: Organization
5. **OU**: Organization Unit
6. **CN**: Common name (basically the domain name or the hostname)

可以随意改动这些值.

```
# sudo is needed due chown, chgrp, chmod ...
sudo ./gen-certificates.sh
```

如果一切都OK，应该能看到两个带证书的新目录 `./certificates/` and `/webgateway-apache-certificates/` with certificates:

File	Container	Description
<code>./certificates/CAServer.cer</code>	webgateway,iris	Authority server certificate
<code>./certificates/irisserver.cer</code>	iris	Certificate for IRIS instance (used for mirror and wegateway communication encryption)
<code>./certificates/irisserver.key</code>	iris	Related private key
<code>/webgateway-apache-certificates/apachewebgateway.cer</code>	webgateway	Certificate for apache webserver
<code>/webgateway-apache-certificates/apachewebgateway.key</code>	webgateway	Related private key
<code>./certificates/webgatewayclient.cer</code>	webgateway	Certificate to encrypt communication between webgateway and IRIS
<code>./certificates/webgatewayclient.key</code>	webgateway	Related private key

请记住，如果有自签名的证书，浏览器会显示安全警报。

显然，如果你有一个由认证机构交付的证书，你可以用它来代替自签的证书（尤其是Apache服务器证书）

Web Gateway 配置文件

让我们来看看配置文件。

CSP.INI

你能看到在 webgateway-config-files 目录下 CSP.INI 文件。
将被推到镜像里, 但内容可以在runtime被修改。
可以把这个文件作为模版。

在这个示例中, 以下参数将在容器启动时被覆盖:

- IpAddress
- TCPPort
- SystemManager

更多细节请参考 [startUpScript.sh](#) . 大致上, 替换是通过sed命令行进行的。

同时, 这个文件包含 SSL \TLS 配置来确保与 IRIS 实例的通信:

```
SSLCC_Certificate_File=/opt/webgateway/bin/webgateway_client.cer
SSLCC_Certificate_Key_File=/opt/webgateway/bin/webgateway_client.key
SSLCC_CA_Certificate_File=/opt/webgateway/bin/CA_Server.cer
```

这些语句都比较重要. 我们必需确保证书文件可用。
我们稍后将在docker-compose文件中用一个卷来做这件事。

000-default.conf

这是一个Apache 配置文件. 允许使用HTTPS协议并将HTTP请求重定向到HTTPS。
证书和私钥文件在这个文件里设置:

```
SSLCertificateFile /etc/apache2/certificate/apache_webgateway.cer
SSLCertificateKeyFile /etc/apache2/certificate/apache_webgateway.key
```

IRIS 实例

对我们 IRIS实例, 我们仅仅配置最低要求来允许SSL \TLS 和Web Gateway 之间的通信; 这涉及到:

1. %SuperServer SSL Config.
2. Enable SSLSuperServer security setting.
3. Restrict the list of IPs that can use the Web Gateway service.

为简化配置, config-api 用一个简单的JSON 配置文件。

```
{
  "Security.SSLConfigs": {
    "%SuperServer": {
      "CAFile": "/usr/irissys/mgr/CA_Server.cer",
```

```
        "CertificateFile": "/usr/irissys/mgr/iris_server.cer",
        "Name": "%SuperServer",
        "PrivateKeyFile": "/usr/irissys/mgr/iris_server.key",
        "Type": "1",
        "VerifyPeer": 3
    },
    "Security.System": {
        "SSLSuperServer": 1
    },
    "Security.Services": {
        "%Service_WebGateway": {
            "ClientSystems": "172.16.238.50;127.0.0.1;172.16.238.20"
        }
    }
}
```

不需要做任何动作. 在容器启动时这个配置会自动加载.

tls-ssl-webgateway 镜像

dockerfile

```
ARG IMAGEWEBGTW=containers.intersystems.com/intersystems/webgateway:2021.1.0.215.0
FROM ${IMAGEWEBGTW}
ADD webgateway-config-files /webgateway-config-files
ADD buildWebGateway.sh /
ADD startUpScript.sh /
RUN chmod +x buildWebGateway.sh startUpScript.sh && /buildWebGateway.sh
ENTRYPOINT ["/startUpScript.sh"]
```

默认的 entry point 是 /startWebGateway, 但是在启动 webserver 前需要执行一些操作. 记住我们的 CSP.ini 文件只是个模版, 并且我们需要在启动时改变一些参数 (IP, port, system manager). startUpScript.sh 将执行这些变化并启动初始 entry point 脚本 /startWebGateway.

启动容器

docker-compose 文件

启动容器之前, 必须修改好 docker-compose.yml 文件:

- ****SYSTEMMANAGER**** 必须配好授权的IP来访问 **Web Gateway Management**
<https://localhost/csp/bin/Systems/Module.cmx>
基本就是你自己的IP地址 (可以是一个用逗号分开的列表).
- ****IRISWEBAPPS**** 必须配好 CSP 应用列表. 这个表用空格隔开, 例如: IRISWEBAPPS=/csp/sys /swagger-ui. 默认, 只有 /csp/sys 被暴露.
- 80和 443 端口映射好. 如果你的系统中已经使用了这些端口, 请将调整为其他端口.

```
version: '3.6'
services:
```

```
webgateway:
  image: tls-ssl-webgateway
  container_name: tls-ssl-webgateway
  networks:
    app_net:
      ipv4_address: 172.16.238.50
  ports:
    # change the local port already used on your system.
    - "80:80"
    - "443:443"
  environment:
    - IRIS_HOST=172.16.238.20
    - IRIS_PORT=1972
    # Replace by the list of ip address allowed to open the CSP system manager
    # https://localhost/csp/bin/Systems/Module.cxx
    # see .env file to set environment variable.
    - "SYSTEM_MANAGER=${LOCAL_IP}"
    # the list of web apps
    # /csp allow to the webgateway to redirect all request starting by /csp to the i
ris instance
    # You can specify a list separate by a space : "IRIS_WEBAPPS=/csp /api /isc /swa
gger-ui"
    - "IRIS_WEBAPPS=/csp/sys"
  volumes:
    # Mount certificates files.
    - ./volume-
apache/webgateway_client.cer:/opt/webgateway/bin/webgateway_client.cer
  - ./volume-
apache/webgateway_client.key:/opt/webgateway/bin/webgateway_client.key
  - ./volume-
apache/CA_Server.cer:/opt/webgateway/bin/CA_Server.cer
  - ./volume-
apache/apache_webgateway.cer:/etc/apache2/certificate/apache_webgateway.cer
  - ./volume-
apache/apache_webgateway.key:/etc/apache2/certificate/apache_webgateway.key
  hostname: webgateway
  command: ["--ssl"]

iris:
  image: intersystemsdc/iris-community:latest
  container_name: tls-ssl-iris
  networks:
    app_net:
      ipv4_address: 172.16.238.20
  volumes:
    - ./iris-config-files:/opt/config-files
    # Mount certificates files.
    - ./volume-iris/CA_Server.cer:/usr/irissys/mgr/CA_Server.cer
    - ./volume-iris/iris_server.cer:/usr/irissys/mgr/iris_server.cer
    - ./volume-iris/iris_server.key:/usr/irissys/mgr/iris_server.key
  hostname: iris
  # Load the IRIS configuration file ./iris-config-files/iris-config.json
  command: ["-a","sh /opt/config-files/configureIris.sh"]

networks:
  app_net:
    ipam:
      driver: default
      config:
```

- subnet: "172.16.238.0/24"

Build and start:

```
docker-compose up -d --build
```

tls-ssl-iris 和 tls-ssl-webgateway 容器应该启动好了.

测试 Web Access

Apache 默认页

打开网页 <http://localhost>.

你将自动被重定向到<https://localhost>.

浏览器显示安全警告. 如果是自签署的证书, 这是正常的, 接受并继续.

Web Gateway 管理页面

打开 <https://localhost/csp/bin/Systems/Module.cmx> 并测试服务器连接.

管理门户

打开 <https://localhost/csp/sys/utilhome.csp>

赞! Web Gateway 例子跑起来了!

IRIS Mirror 与vWeb Gateway

在上一篇文章中, 我们建立了一个镜像环境, 但网络网关是一个缺失的部分. 现在, 我们可以改进这一点. 一个包括Web Gateway和一些更多改进的资源库就可以用了 [iris-mirroring-with-webgateway](#) :

1. 证书不再是即时生成的, 而是在一个单独的过程中生成的.
2. IP地址被docker-compose和JSON配置文件中的环境变量所取代, 变量被定义在'.env'文件中.
3. 这个repository 可以作为一个模板来使用.

查看 repository文件 [README.md](#) 来运行以下环境:

[#Web网关](#) [#开发运维](#) [#InterSystems IRIS](#)

源

URL:<https://cn.community.intersystems.com/post/%E5%9F%BA%E4%BA%8Edocker%E7%9A%84apache-web-gateway>