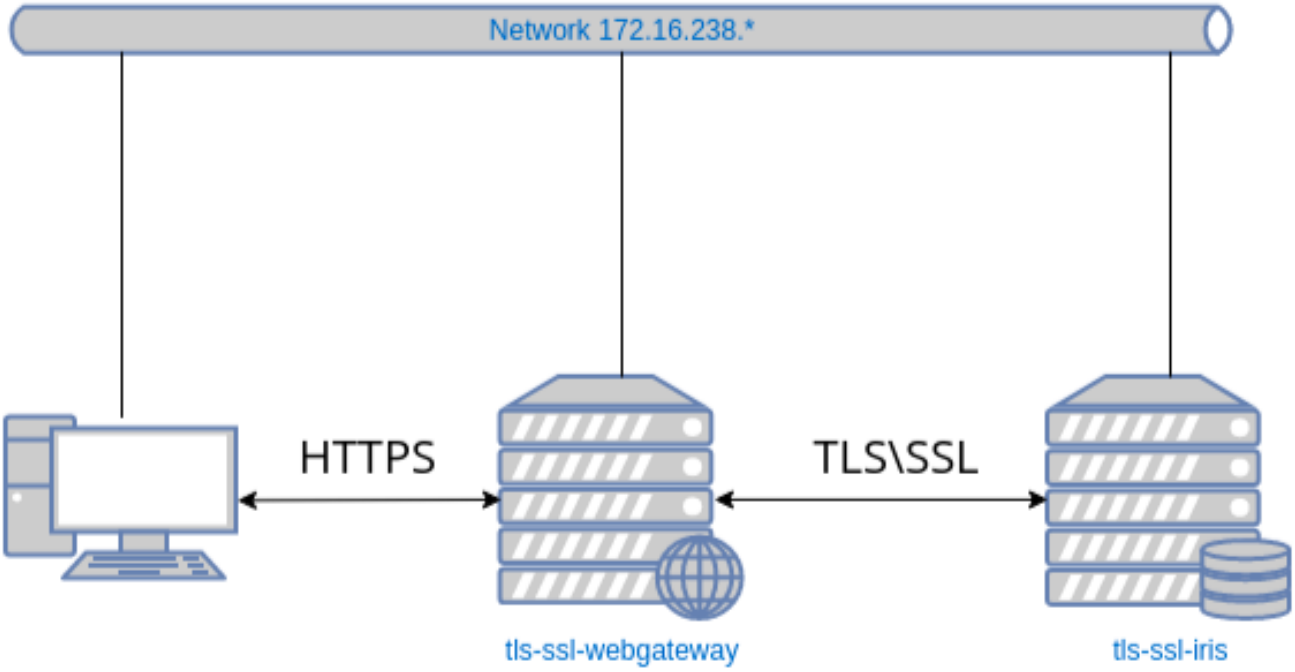
Article
Lorenzo Scalese · Juil 13, 2022 9m de lecture

La WebGateway Apache avec Docker

Salut, communauté.

Dans cet article, nous allons configurer de manière programmatique la WebGateway Apache avec Docker en incluant :

- Protocole HTTPS.
- TLS \SSL pour sécuriser la communication entre la WebGateway et l'instance IRIS.



Nous utiliserons deux images : une pour la WebGateway et la seconde pour l'instance IRIS.

Tous les fichiers nécessaires sont disponibles dans ce repository GitHub.

Commençons par un clone git :

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

Préparation de votre système

Pour éviter les problèmes de permissions, votre système a besoin d'un utilisateur et d'un groupe :

- www-data
- irisowner

Il est nécessaire de partager les fichiers de certificats avec les conteneurs. S'ils n'existent pas sur votre système, exécutez simplement :

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

Génération des certificats

Dans cet exemple, nous allons utiliser trois certificats :

1. Utilisation du serveur web HTTPS.
2. Cryptage TLS \SSL sur le client de la WebGateway.
3. Cryptage TLS \SSL sur l'instance IRIS.

Un script prêt à l'emploi est disponible pour les générer.

Cependant, vous devez personnaliser l'objet du certificat ; il suffit pour cela de modifier le fichier gen-certificates.sh.

Voici la structure de l'argument subj d'OpenSSL :

1. **C** : Code pays
2. **ST** : État
3. **L** : Localisation
4. **O** : Organisation
5. **OU** : Unité d'organisation
6. **CN** : Nom commun (essentiellement le nom de domaine ou le nom d'hôte)

N'hésitez pas à modifier ces valeurs afin que ce soit cohérent avec votre localisation, organisation, etc...

```
# sudo est nécessaire pour chown, chgrp, chmod ...
sudo ./gen-certificates.sh
```

Si tout est ok, vous devriez voir deux nouveaux répertoires ./certificates/ et /webgateway-apache-certificates/ avec des certificats :

Fichier	Conteneur	Description
./certificates/CAServer.cer	webgateway,iris	Certificat du serveur d'autorité
./certificates/irisserver.cer	iris	Certificat pour l'instance IRIS (utilisé pour le cryptage de la communication entre le miroir et la WebGateway)
./certificates/irisserver.key	iris	Clé privée associée
/webgateway-apache-certificates/apachewebgateway.cer	webgateway	Certificat pour le serveur web apache
/webgateway-apache-certificates/apachewebgateway.key	webgateway	Clé privée associée
./certificates/webgatewayclient.cer	webgateway	Certificat pour crypter la communication entre webgateway et IRIS
./certificates/webgatewayclient.key	webgateway	Clé privée associée

Gardez à l'esprit que ce sont des certificats auto-signés, les navigateurs web afficheront des alertes de sécurité. Évidemment, si vous avez un certificat délivré par une autorité certifiée, vous pouvez l'utiliser à la place d'un certificat auto-signé (en particulier pour le certificat du serveur Apache).

Fichiers de configuration de la WebGateway

Jetons un coup d'œil aux fichiers de configuration.

CSP.INI

Vous pouvez voir un fichier CSP.INI dans le répertoire webgateway-config-files.
Il sera inséré dans l'image, mais son contenu peut être modifié lors de l'exécution.
Considérez ce fichier comme un modèle.

Dans cet exemple, les paramètres suivants seront remplacés au démarrage du conteneur :

- IpAddress
- TCPPort
- SystemManager

Référez-vous à startUpScript.sh pour plus de détails. De façon générale, le remplacement est effectué avec la ligne d'instruction sed.

Ce fichier contient également la configuration SSL\TLS pour sécuriser la communication avec l'instance IRIS :

```
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
```

Ces lignes sont importantes. Nous devons nous assurer que les fichiers de certificats seront disponibles pour le conteneur.
Nous ferons cela plus tard dans le fichier docker-compose avec un volume.

000-default.conf

Il s'agit d'un fichier de configuration d'Apache. Il permet l'utilisation du protocole HTTPS et redirige les appels HTTP vers HTTPS.
Les fichiers de certificat et de clé privée sont configurés dans ce fichier :

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

L'instance IRIS

Pour notre instance IRIS, nous ne configurons que le minimum requis pour permettre la communication SSL\TLS avec la WebGateway ; cela implique les éléments suivants :

1. Configuration SSL de %SuperServer.
2. Activation du paramètre de sécurité SSLSuperServer.
3. Restriction de la liste des IPs qui peuvent utiliser le service Web Gateway.

   Pour faciliter la configuration, on utilise config-api avec un simple fichier de configuration JSON.

```
{
   "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"
       }
   }
}
```

Aucune action n'est nécessaire. La configuration sera automatiquement chargée au démarrage du conteneur.

Image tls-ssl-webgateway

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"]
```

Par défaut, le point d'entrée est /startWebGateway, mais il faut effectuer quelques opérations avant de démarrer le serveur web. Rappelez-vous que notre fichier CSP.ini est un template, et que nous devons changer certains paramètres (IP, port, gestionnaire de système) au démarrage. startUpScript.sh va effectuer ces changements et ensuite exécuter le script de point d'entrée initial /startWebGateway.

Conteneurs de départ

docker-compose file

Avant de démarrer les conteneurs, le fichier docker-compose.yml doit être modifié ::

- **SYSTEMMANAGER** doit être défini avec l'IP autorisée à avoir un accès à **Web Gateway Management** https://localhost/csp/bin/Systems/Module.cxw.
  En gros, c'est votre adresse IP (Cela peut être une liste séparée par des virgules).
-

IRISWEBAPPS doit être défini avec la liste de vos applications CSP. La liste est séparée par des espaces, par exemple : IRISWEBAPPS=/csp/sys /swagger-ui. Par défaut, seul /csp/sys est exposé.

- Les ports 80 et 443 sont mappés. Adaptez-les à d'autres ports s'ils sont déjà utilisés sur votre système.

```
version: '3.6'
services:

 webgateway:
   image: tls-ssl-webgateway
   container_name: tls-ssl-webgateway
   networks:
     app_net:
       ipv4_address: 172.16.238.50
   ports:
     # modification du port local déjà utilisé sur votre système.
     - "80:80"
     - "443:443"
   environnement:
     - IRIS_HOST=172.16.238.20
     - IRIS_PORT=1972
     # Remplacement par la liste des adresses IP autorisées à ouvrir le gestionnaire
du système CSP
     # https://localhost/csp/bin/Systems/Module.cxw
     # consultez le fichier .env pour définir les variables d'environnement.
     - "SYSTEM_MANAGER=${LOCAL_IP}"
     # la liste des applications web
     # /csp permet à la webgateway de rediriger toutes les requêtes commençant par /c
sp vers l'instance iris
     # Vous pouvez spécifier une liste séparée par un espace : "IRIS_WEBAPPS=/csp /ap
i /isc /swagger-ui"
     - "IRIS_WEBAPPS=/csp/sys"
   volumes:
     # Montage des fichiers de certificats.
     - ./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
```

```
   # Chargement du fichier de configuration IRIS ./iris-config-files/iris-config.json
   commande: ["-a","sh /opt/config-files/configureIris.sh"]

networks:
 app_net:
   ipam:
     driver: default
     config:
       - subnet: "172.16.238.0/24"
```

Construction et démarrage :

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

Les conteneurs tls-ssl-iris et tls-ssl-webgateway doivent être démarrés.

Test de l'accès au Web

Page par défaut d'Apache

Ouvrez la page http://localhost.
Vous serez automatiquement redirigé vers https://localhost.
Les navigateurs affichent des alertes de sécurité. C'est le comportement standard avec un certificat auto-signé, dans notre cas vous pouvez accepter le risque et continuer.

Page de gestion de la WebGateway

Ouvrez https://localhost/csp/bin/Systems/Module.cxw et testez la connexion du serveur.

Portail de gestion

Ouvrez https://localhost/csp/sys/utilhome.csp

Parfait! Notre sample WebGateway fonctionne!

Miroir IRIS avec la WebGateway

Dans l'article précédent, nous avons construit un environnement miroir, mais la WebGateway était une pièce manquante. Maintenant, nous pouvons y remédier.
Un nouveau dépôt iris-miroring-with-webgateway est disponible incluant la WebGateway et quelques autres améliorations :

1. Les certificats ne sont plus générés à la volée mais dans un processus séparé.
2. Les adresses IP sont remplacées par des variables d'environnement dans les fichiers de configuration docker-compose et JSON. Les variables sont définies dans le fichier '.env'.
3. Le repository peut être utilisé comme modèle.

Consultez le fichier du dépôt README.md pour exécuter un environnement comme celui-ci :

#DevOps #Web Gateway #InterSystems IRIS

URL de la source:https://fr.community.intersystems.com/post/la-webgateway-apache-avec-docker