Purpose

vSphere Mobile Client supports push notifications. In order to enable that functionality, a service must be deployed and configured first. The purpose of the service is to enable communication with vCenter servers and provide data to the mobile application (push notifications). There are three different ways to deploy the service, as listed below

- Deploy, configure and run the spring boot service on one of your already configured hosts/VMs
- Deploy a docker container on an already configured host/VM that executes the service inside it
- Deploy an OVA image that has PhotonOS preconfigured

Below, you will find instructions on how to deploy and configure the service using your preferred method.

Certificates and security

Keeping the application data and communications secure is of very high importance, thus, all communication between the mobile application (running on a smartphone/tablet) and the mobile service (deployed with the instructions provided in this document) is encrypted with SSL. To secure communication between the mobile service and a given vCenter server, a trustore file is provided to the former, that contains all certificates that it can trust (reffered to as csm-trust-store.pfx in this document). Typically, the store contains the root certificate fetched from a given VCSA.

To secure the connection between the mobile client and the mobile service, a keystore containing a certificate should be provided to the spring boot server. A certificate might be one provided by your organization, or a self-signed certificate. We strongly advice against using self-signed certificates in production environments.

Network requirements

In order to enable push notifcations, the service should be able to access external resources via https (remote REST API calls).

OVA Deployment

Requirements

- Host system with 4 GB of RAM
- 18 GB of storage available (thick provisioning)

Credentials

- User **root** has password vsphere-mobile
- User **mobile** has password vsphere-mobile

Installation

1. Deploy the provided OVA image. During deployment you can select one of the following options for the network configuration
    a. Provide hostname, ip address, netmask, gateway, dns and dns domain in order to use specific configuration
    b. Leave everything empty if DHCP network configuration is desired
2. Change the password for the root and mobile users
3. Open console/SSH to the machine as the user **mobile**
4. Service directory is located at */home/mobile/service*, cd to it
5. Use the cloudsmith configuration utility (described below) to generate the required certificates and the application.properties file. Edit the latter if usage of custom certificates is desired
6. Switch to the **root** user
7. Open the port specified in the application.properties file using IPTABLES (it is recommended to create a permanent rule for it)
    a. iptables -A INPUT -p tcp --dport 8888 -j ACCEPT
8. Enable the vsphere-mobile service
    a. systemctl enable vsphere-mobile
9. Start the vsphere-mobile service
    a. systemctl start vsphere-mobile
10. In case there is some error, use journalctl to see the log
    a. journalctl -u vsphere-mobile -b

Using the cloudsmith configuration service

In order to simplify the service configuration process, a utility is provided that can generate the necessary certificates for both the notification service and the trust store that is required to be able to connect to the target vCenter Servers.

The utility comes in the form of a JAR file that can be invoked with the following command

- java -jar cloudsmith-configuration-service.jar

You may specify the following arguments:

--debug	(Optional) Enables verbose logging. Enable this if something is not working as expected and you need additional information displayed while the utility is running
--port	(Optional) Specify the port on which the notification service will run on. Port 443 will be used if not specified
--vcenter	[Required] A comma separated list of vCenter servers to which the service should be able to connect to. For each server specified, the SSH login credentials will be requested while the utility is running, in order to enable it to fetch the server's root certificate

After running it and if there are no errors the following files will be generated

application.properties	Configuration script that is used by spring boot to configure the application
service-keystore.pfx	Applications trying to connect to the notification service should validate and accept the certificate in this keystore
csm-trust-store.pfx	Contains all root certificates that allow the notification service to connect to the various vCenter servers

You can replace the application.properties file that came with the .OVA, or the one you may have previously created for the docker setup, with the one generated by the utility, and place the .pfx files to the locations specified within that file.

Docker Image

Requirements

- Docker installation on the target host system
- Enough memory to run spring boot application (recommended: at least 2 GB of RAM)
- keytool

Installation

1. Obtain a copy of the docker image (vsphere-mobile-client-service.tar.bz2) from the fling site
2. Load the image into docker. This creates an image entry into docker's image repository
    - *docker load -i vsphere-mobile-client-service.tar.bz2*
3. Create a container from the docker image. The `-p` option specifies that port 8888 on the container (can be changed in the *application.properties* file, see below) maps to port 443 on the host. You may use a different port instead of 443 on the host.
    - *docker create -p 443:8888 vsphere-mobile-client-service*
4. Copy the truststore file to the container (find instructions below on how to obtain it)
    - *docker cp <truststore>.pfx <container_id>:/csm-trust-store.pfx*
5. Copy the service keystore to the container (find instructions below on how to generate a self-signed one)
    - *docker cp <keystore>.pfx <container_id>:/service-keystore.pfx*
6. Create an *application.properties* file for the spring boot application

```
spring.application.admin.enabled=false
logging.level.org.apache=INFO
logging.level.com.vmware.cloudsmith=INFO
server.port=8888
server.ssl.key-store=service-keystore.pfx
server.ssl.key-store-type=PKCS12
server.ssl.key-store-password=<key-store-password>
server.ssl.protocol=TLSv1.2
app.cloudsmith.trust-store=csm-trust-store.pfx
app.cloudsmith.trust-store-password=<trust-store-password>
```

7. Copy the *application.properties* file to the container
   - *docker cp application.properties <container_id>:/application.properties*
8. Start the container (skip to step 9 if you want the container to start running automatically)
   - *docker start <container_id>*
9. Configure the container's restart policy, so that the container stays always running
   - *docker update --restart=always <container_id>*

Generating the trust store file (*csm-trust-store.pfx)*

1. SSH to the VCSA and fetch the root certificate located at */var/lib/vmware/vmca/root.cer*
2. Create a keystore file containing the certificate
   a. *keytool -import -file root.cer -alias vcsaCA -storetype PKCS12 -keystore csm-trust-store.pfx*

Generating a keystore with a self-signed certificate (*service-keystore.pfx)*

1. *keytool -genkeypair -alias service -keyalg RSA -keysize 2048 -storetype PKCS12 -keystore service-keystore.pfx -validity 365*