Install Nexus Push Service

For the Nexus Personal Mobile standard app, the Nexus Push Service is hosted by Nexus. If the customer does not want to hand over push certificates or firebase token API, the Nexus Push Service can be installed on-premises.

Expand/Collapse All

Prerequisites

Prerequisites

- Knowledge about docker.
- A host with docker and docker-compose installed and configured.
- A public DNS name which devices can reach.
- Matching certificates for the public address.
- Installed instance of MongoDB or Microsoft Azure Cosmos DB (unless included in the installation procedure described in this article).

Step-by-step instruction

Create structure and copy configuration and certificate files

1. Create a file structure for the configuration and certificate files.

   Example: File structure

   ```
   mydir/
   |-- cacerts
   |   `-- cacert.cer
   |-- certificates
   |   |-- push-development.p12
   |   |-- push-production.p12
   |   `-- sslcert.p12
   |-- config
   |   |-- application.yml -> cod-nps.yml
   |   `-- cod-nps.yml
   `-- docker-compose.yml
   ```

2. Copy the configuration and certificate files from git to the structure.

Load docker image

The docker image can be loaded locally to each machine, or to the docker registry, and used remotely from several machines.

1. If you want to load the released docker image locally on the target host:

Example: load image locally

```
docker load -i nps-1.0.2.RELEASE.tar
```

2. If you have a docker registry, load the image there.

Edit configuration files

1. Edit the configuration files `docker-compose.yml` and `config/cod-nps.yml` with the correct values for your environment. Below you can find examples, but the actual values must match the specific deployment scenarios.

Example: docker-compose.yml

```
version: '2'
services:
  nps:
 restart: unless-stopped
 image:
ng-docker01.ad.nexusgroup.com:5000/nexus-cod/nps:1.0.2.
RELEASE
 ports:
  - "20200:20200"
 extra_hosts:
  - "nps-host:10.5.0.5"
 environment:
  - TZ=Europe/Stockholm
  - JAVA_OPTS=-Xms256m -Xmx512m
-XX:MaxMetaspaceSize=512m
  -XX:CompressedClassSpaceSize=64m -Xss256k -Xmn8m
-XX:InitialCodeCacheSize=4m
  -XX:ReservedCodeCacheSize=64m
-XX:MaxDirectMemorySize=64m
 mem_limit: 1024m
 volumes:
  - ./certificates:/home/docker/certificates:z
  - ./cacerts:/cacerts:z
 logging:
  options:
   max-size: 10m
 command: "--server.contextPath=/
--spring.cloud.config.failFast=false
--spring.boot.admin.client.enabled=false"
```

Example: cod-nps.yml

```
logging:
  level:
 org.springframework.context.annotation.AnnotationConfi
```

```
gApplicationContext: ERROR
 org.springframework.boot.SpringApplication: ERROR
 org.springframework.cloud.config.client: ERROR
 com.netflix: INFO
 com.nexusgroup: TRACE
 com.relayrides: TRACE
  pattern:
 console: "%d{yyyy-MM-dd}T%d{HH:mm:ss.SSS}Z
${LOG_LEVEL_PATTERN:- %5p} [%t]
%-40.40logger{39} [%mdc] :
%m%n${LOG_EXCEPTION_CONVERSION_WORD:%wEx}"
spring:
  data:
 mongodb:
  database: nps
  host: mongo-host # Mongo server host.
  # You can also specify a uri to Microsoft Dynamo db
like
  # uri:
"mongodb://docdb:SECRET-API-KEY@docdb.documents.azure.c
om:10250/?ssl=true"
application:
  nps:
 rest:
  uribase: ""
  log: false
    allowedClients:

 # Note!
 # The X-Api-Key should be created using
base64(clientId:key)
 #
 # NPS has a helper endpoint to generate configuration.
Simply use (make sure you have the correct  host/port)
 # curl
'http://localhost:20200/util/generateclient/default'
 # to get a snippet which can be pasted to the
configuration file
 #
 # X-Api-Key:
ZGVmYXVsdDoyY2QxNzE1Y2Q3YmE0NTM2OGQxNGI2MDBiMjllOWUxNGE
4ZDNjOThlNDM2MDRmMjQ4NGY3Yjg1NmY5ZGRiMjU2
 -  clientId: default
  key:
2cd1715cd7ba45368d14b600b29e9e14a8d3c98e43604f2484f7b85
6f9ddb256

  pushConfiguration:
      appIdentifiers:
  # Note!
  # If you are using configserver, the password can
  # be encrypted via 'curl localhost:20000/encrypt -d
MY_PASSWORD'
  # Add the {cipher} prefix to encrypted password
  #
  # Personal beta app
  #
        -   appIdentifier: com.nexusgroup.personal.beta
   androidConfig:
    timeToLive: 300
    priority: HIGH
```

```
    googleApiKey: "FIREBASE-API-KEY"
   appleConfig:
    timeToLive: 300
    priority: HIGH
    bundleId: com.nexusgroup.personal.beta
    developmentP12:
/home/docker/certificates/push-development.p12
    developmentPassword: "{cipher}ENCRYPTED-PASSWORD"
    productionP12:
/home/docker/certificates/push-production.p12
    productionPassword: "{cipher}ENCRYPTED-PASSWORD"

      personalMobile:
  notificationSettings:
   -  pushType: AUTH
    locTitleKey: "notification_auth_title"
    locTitleArgs:
     - "{requestor}"
             locBodyKey: "notification_auth_body"
    locBodyArgs:
     - "{requestor}"
           -  pushType: SIGN
    locTitleKey: "notification_sign_title"
    locTitleArgs:
     - "{requestor}"
```

```
            locBodyKey: "notification_sign_body"
    locBodyArgs:
              - "{requestor}"
```

Start Nexus Push Service

1. Start Nexus Push Service with the following command:
   `docker-compose up -d`