

Introduction This document provides an overview of the steps undertaken by CA Support when testing the ability to install the Software edition of APIM Gateway using the headless scripts provided by CA for the auto provisioning an Appliance form factor APIM Gateway. This document covers the following:

- Environment and Preparation
- Auto-Provisioning script installation
 - a. Create the Properties Template
 - b. Edit the Template File
 - c. Create the Gateway Node
- Start the Gateway
- Post install cleanup

Background The standard APIM Gateway documentation specifically states that the auto provisioning is for Appliance based systems only. Initial testing has shown that a successful "headless" installation of a Software edition can be achieved with some small updates to the install process and scripts (as documented below). The basis of the testing was to utilize the headless option which we have on the Virtual Appliance and run this for a Software form factor installation. The end result is a vanilla install of a Gateway instance. The example in this doc is based on CA API Gateway 9.3

DISCLAIMER: At this time, any use of this process should be for testing purposes

Environment

Instructions Environment and Preparation

- RedHat/CentOS Server as the product manual suggest we support.
- Latest JDK as per the product online manual.
- JCE Policy as per the product manual.
- MYSQL Server and Client
- CA APIM Gateway RPM
- CA APIM Gateway license file.

<https://docops.ca.com/ca-api-gateway/9-3/en/release-notes-9-3/requirements-and-compatibility/software-gateway-specifications>

1. Set JAVA_HOME in the environment. For example:

```
[root@localhost tmp]# export JAVA_HOME=/usr/java/jdk1.8.0_181-amd64/
[root@localhost tmp]# export PATH=$PATH:$JAVA_HOME
```

2. Auto-Provisioning script installation

Create the Properties Template

Refer to the current auto provision manual:

<https://docops.ca.com/ca-api-gateway/9-3/en/install-configure-upgrade/auto-provision-a-gateway/auto-provision-a-gateway-node>

The following is the command line, slightly modified to enable it to run for a Software edition install.

```
[root@localhost tmp]# /opt/SecureSpan/Gateway/config/ssgconfig.sh -headless create -template > create-node.properties
```

3. Edit the Template File

Edit the created template to include changes to incorporate the specific requirements for the intended install.

Below an example:

```
##### Headless config create template properties file #####
```

```
### Cluster Configuration ###
```

```
## Cluster Hostname (not required if joining an existing cluster)
```

```
cluster.host=<HOSTNAME>
```

```
## Cluster Passphrase
```

```
cluster.pass=<PASSWORD>
```

```
### Node Configuration ###
```

```
## Node Enabled State
```

```

node.enable=true
## Configure the node.properties
configure.node=true

### CA API Gateway Policy Manager Administrative Account ###
## CA API Gateway Policy Manager Username (not required if joining an existing cluster)
admin.user=admin
## CA API Gateway Policy Manager Password (not required if joining an existing cluster)
admin.pass=<PASSWORD>

### Database Connection ###
## Creates the database
configure.db=true
## The database type, either 'mysql' or 'embedded'
database.type=mysql
## Database Hostname
database.host=localhost
## Database Port
database.port=3306
## Database Name
database.name=ssg
## Database Username
database.user=gateway
## Database Password
database.pass=<PASSWORD>
## Administrative Database Username (not required if joining an existing cluster)
database.admin.user=root
## Administrative Database Password (not required if joining an existing cluster)
database.admin.pass=<PASSWORD>

```

NOTE: Password must meet complexity policy

4. Create the Gateway Node

Use the modified template to create the node. These steps also cover the Auto-provisioning of the following:

- provision a license (filename needs to be license.xml),
- provision the RESTman service:

The following steps/commands perform the equivalent functions

```

[root@localhost tmp]# su - gateway
Last login: Fri Jul 20 00:23:43 EDT 2018 on pts/0
[gateway@localhost ~]$ /opt/SecureSpan/Controller/bin/pc.sh start
Starting Gateway Services: done.
[gateway@localhost ~]$
[gateway@localhost ~]$ exit

[root@localhost tmp]# cat create-node.properties | /opt/SecureSpan/Gateway/config/ssgconfig.sh -headless create
Configuration Successful

[root@localhost tmp]# echo "node.java.path=/usr/java/jdk1.8.0_181-amd64
> node.java.heap=8096" >> /opt/SecureSpan/Gateway/node/default/etc/conf/node.properties
[root@localhost tmp]#

[root@localhost tmp]# chown layer7:gateway /opt/SecureSpan/Gateway/node/default/etc/conf/node.properties
[root@localhost tmp]# chmod 540 /opt/SecureSpan/Gateway/node/default/etc/conf/node.properties
[root@localhost tmp]# chmod 750 /opt/SecureSpan/Gateway/node/default/etc/conf
[root@localhost tmp]#
[root@localhost tmp]# mkdir -p /opt/SecureSpan/Gateway/node/default/etc/bootstrap/services
[root@localhost tmp]# mkdir -p /opt/SecureSpan/Gateway/node/default/etc/bootstrap/license
[root@localhost tmp]# chmod -R 775 /opt/SecureSpan/Gateway/node/default/etc/bootstrap/
[root@localhost tmp]# touch /opt/SecureSpan/Gateway/node/default/etc/bootstrap/services/restman
[root@localhost tmp]#
[root@localhost tmp]# cp license.xml /opt/SecureSpan/Gateway/node/default/etc/bootstrap/license/
[root@localhost tmp]# chmod -R 775 /opt/SecureSpan/Gateway/node/default/etc/bootstrap/license/license.xml
[root@localhost tmp]# chown -R layer7:gateway /opt/SecureSpan/Gateway/node/default/etc/bootstrap/

```

5. Start the Gateway

```
[root@localhost tmp]# su - gateway  
Last login: Fri Jul 20 00:35:07 EDT 2018 on pts/0  
[gateway@localhost ~]$ /opt/SecureSpan/Gateway/runtime/bin/gateway.sh start  
Starting Process Controller...  
Starting Gateway Services: done.
```

6. Post install cleanup

After the gateway is completely started up with the license and the service in place remove the files:

```
[gateway@localhost ~]$ rm -rf /opt/SecureSpan/Gateway/node/default/etc/bootstrap
```

This will get you with an up and running Vanilla Software Gateway.