Avaya Experience Portal 8.1.2
Mobile Web Best Practices Guide

Abstract

This paper provides information about recommended strategies for deploying Avaya Orchestration Designer Mobile Web applications with Avaya Experience Portal 8.1.2, detailing configuration for security, scalability and high availability.

Issue 1.0
Oct 2022

Contents

1. Overview

Deployment of Avaya Orchestration Designer Mobile Web applications requires proper configuration and connection of multiple software components. This document discusses the recommended configuration of the Avaya Experience Portal system in relation to the Orchestration Designer Mobile Web application with strategies to maximize security, scalability and availability. Accordingly, this document enumerates the steps required to setup a single recommended configuration, but alternate configurations are also valid and may be preferred for installations with lesser requirements. Guidance on what simplifications can be made will also be provided.

2. Recommended Architecture

The recommended system architecture for employing Mobile Web applications is shown in Figure 1 which outlines the logical components and their interconnections.

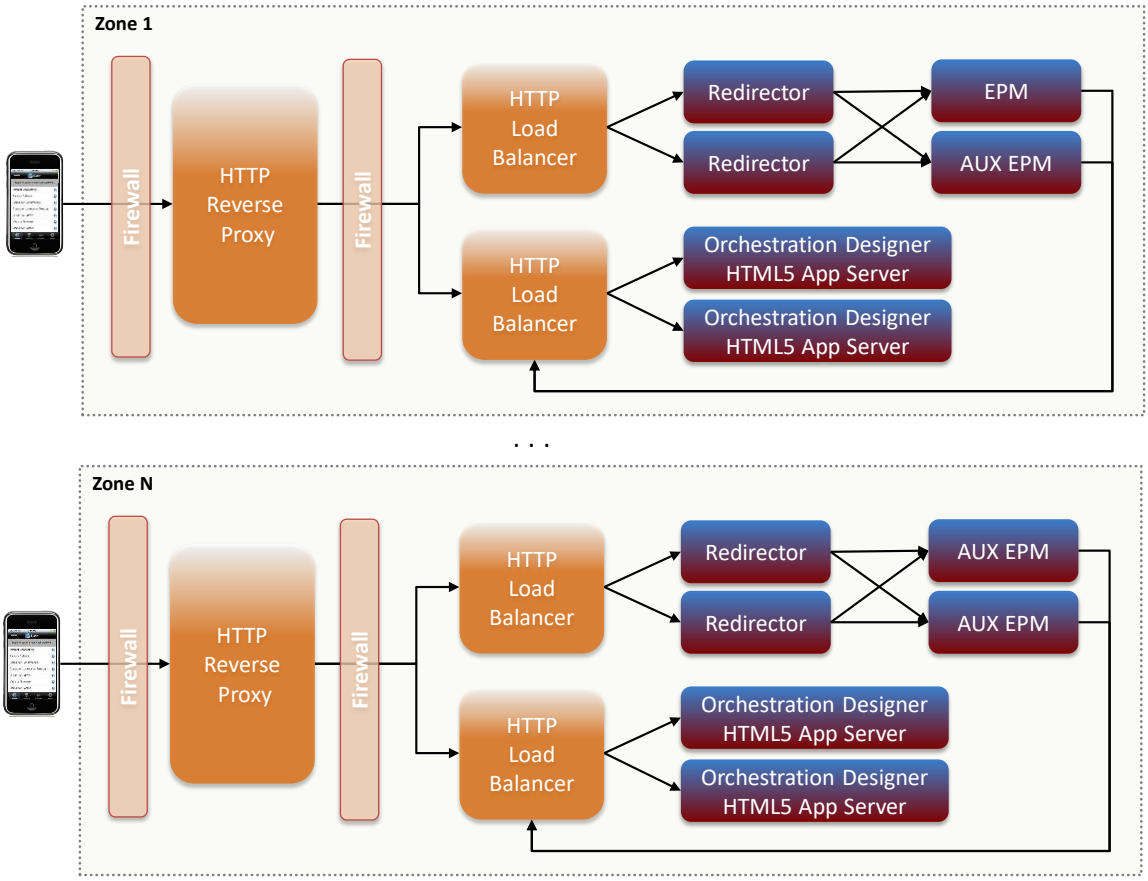

Figure 1 Logical System Architecture

Fault tolerance and scalability is achieved by ensuring that each component is duplicated for a fully redundant system. Specifically, the components of the Experience Portal system, the Redirector,

EPM/Auxiliary EPM, and OD Application Server, are deployed in pairs in the recommended configuration. For installations that don't require high availability though, the duplicate servers can be omitted. Conversely, extra capacity and redundancy can be added by deploying additional servers. For the Reverse Proxy, Load Balancer and Firewalls, few specifications are made; however, it is assumed that these components are also fault tolerant systems. In the case of the Load Balancer, a sample configuration utilizing redundant Apache HTTP servers sharing a virtual IP address is supplied to exemplify the exact functionality that is required for this component. Alternatives with the same behavior can also be used though. In particular, the Load Balancer and Reverse Proxy can actually consist of a single redundant system (with minor modifications in the firewall setup). The recommended configuration considers the reverse proxy as a separate server however.

Security is prioritized in the recommended configuration by utilizing authenticated TLS connections with the HTTPS (HTTP Secure) protocol between all of the various components. However, for some configurations it may be preferable to utilize just TCP connections with the HTTP protocol or a mix of TCP and authenticated TLS connections. This is accomplished by adjusting the appropriate URLs in the recommended configuration to use the http scheme instead of the https scheme. In Figure 1, the arrows indicate the direction that HTTP request flow. HTTP responses flow in the opposite direction.

In the steps provided for setting up the recommended configuration, only the default zone is considered. For installations with multiple zones, the configuration steps are essentially repeated for each additional zone. Redirectors and Auxiliary EPMs are assigned to particular zones and only handle requests for Mobile Web Applications configured in that zone. OD Application servers and Load Balancers can serve multiple zones, although it is also acceptable to dedicate servers to individual zones.

The physical view of the recommended configuration is somewhat different than the logical architecture and is seen in Figure 2. In particular, since the Redirector component is a simple Java servlet, it is installed on the same Tomcat servers that host the OD Mobile Web applications. For some installations though it might be desirable to install the Redirectors on standalone servers which is also a valid configuration. For VMware installations, each of the physical servers in the diagram would be separate virtual machines and some care must be taken to ensure that duplicate components are running on different hosts to maintain redundancy.

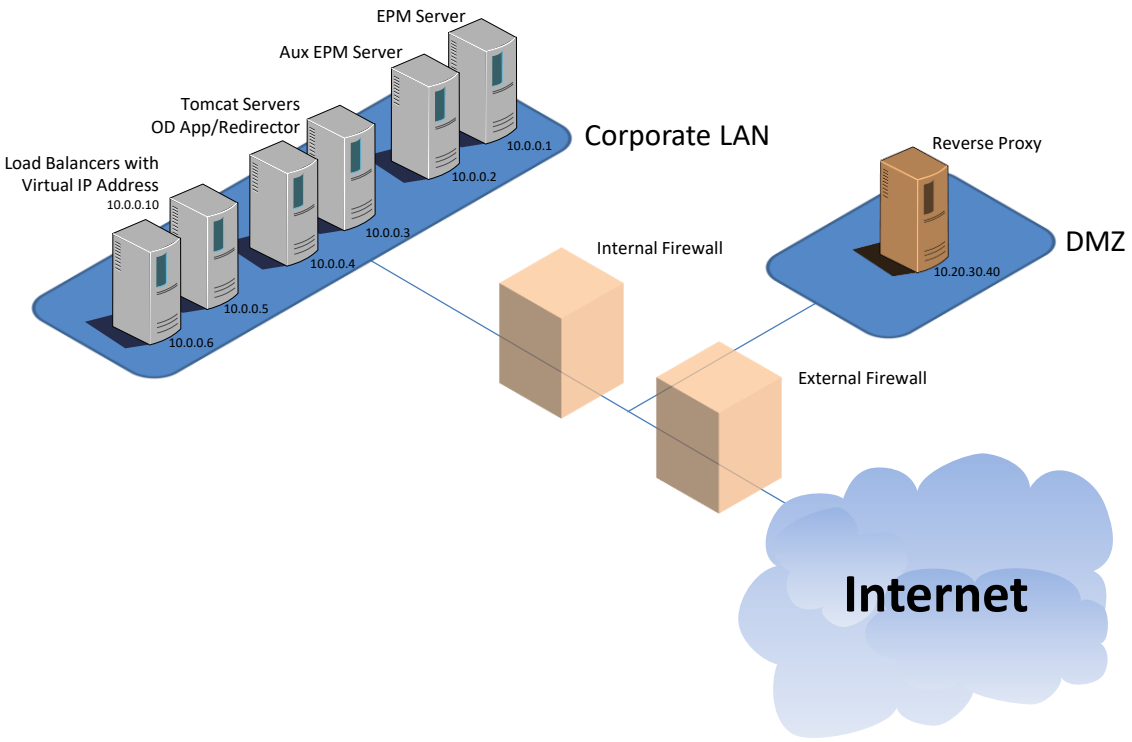

Figure 2 Physical View

3. Configuration Assumptions

Minimum software requirements met

Experience Portal installed and configured

OD Application servers installed and configured with OD Mobile Web applications already deployed

4. Administering Experience Portal

The following describes the configuration changes needed on the EPM to setup a Mobile Web application.

4.1. Adding a Mobile Web Application

This section describes the recommended steps to configure a Mobile Web application on the EPM.

Procedure

1. On the Primary EPM web administration, navigate to **System Configuration > Applications** in the left navigation pane to open the **Applications** page (Figure 3).

2. Click on the **Add** button to add a Mobile Web application.
3. Enter the appropriate information on the **Add Application** page.
    - Use the drop down box in the **Zone** field to select the appropriate zone.
    - Enter a descriptive name in the **Name** field.
    - Change the drop down box in the **Type** field to **HTML**.
4. Click **Continue**. The URI configuration section for an HTML type application will be displayed.
5. Continue entering information into the URI section on the **Change Application** page (Figure 4).
    - Ensure that the **Single** radio button is selected.
    - Enter the URL that an EPM should use to directly connect with the OD Application servers when initializing a new session of the Mobile Web Application into the **Internal URL** field. This URL should utilize the https (Secure HTTP) scheme and includes the virtual IP address maintained by the load balancers (or a hostname that resolves as such).
    - Enter the URL that an EPM should return as the location for an external browser to access the same application in the **External URL** field. This URL should also use the https (Secure HTTP) scheme and includes an IP address or hostname that resolves to the reverse proxy. The path for this URL must match a configuration in the reverse proxy that maps to the load balancer configuration for the application (see Reverse Proxy configuration).
    - Ensure that the **Yes** radio button is selected in the **Mutual Authentication** field.
6. Click **Save**.

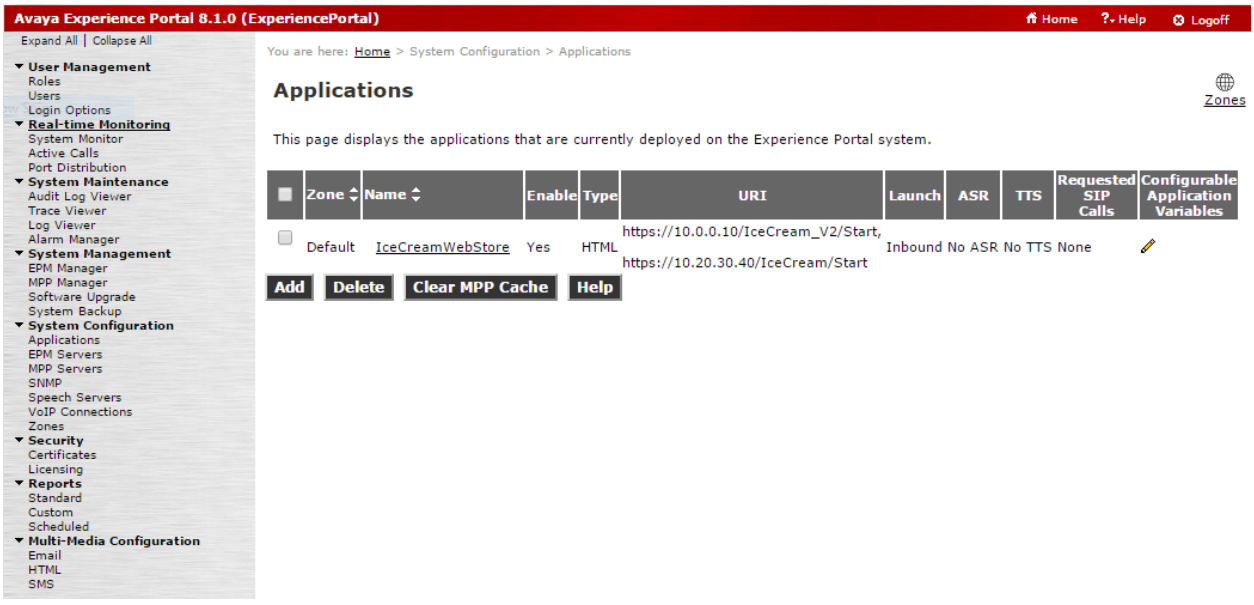

Figure 3 Applications web page

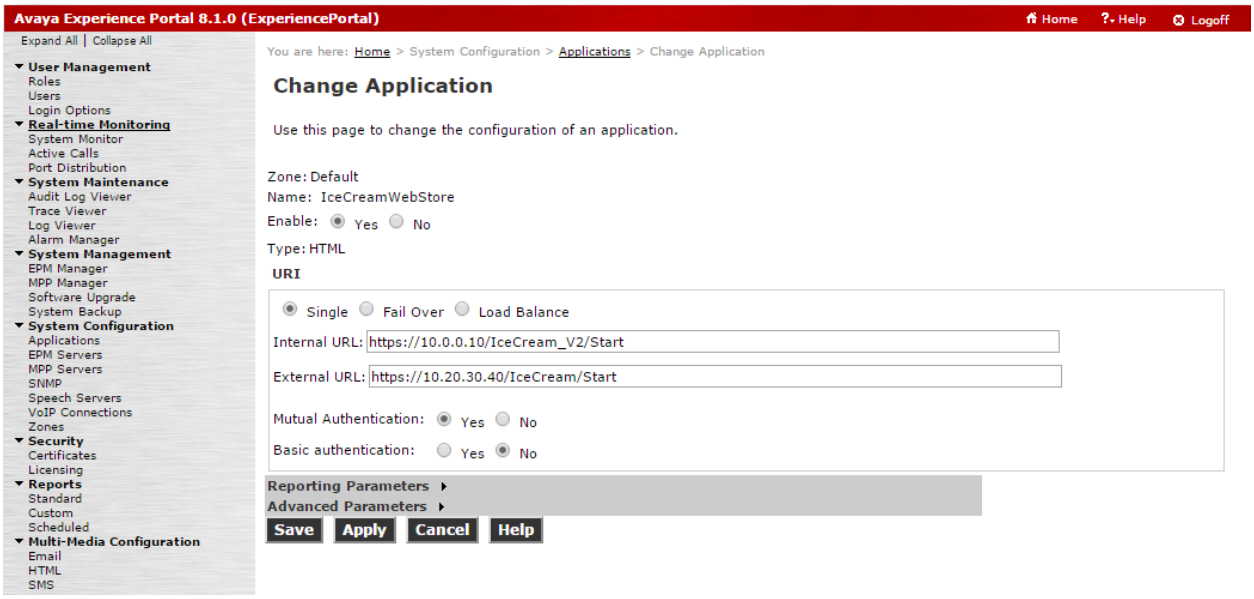

Figure 4 Change Application web page

4.2. Adding Auxiliary EPMs

The AppIntf web service hosted on Primary and Auxiliary EPMs provides a method to launch Mobile Web applications. The web service takes a provided application name and uses the matching configuration to locate an application web server hosting that Mobile Web application. The EPM then starts a new session of the application by sending an HTTP request to the application web server that supplies configured initialization parameters. If the initialization request is successful the web service returns a URL pointing to the application on the found web server which also includes a unique session key for the newly launched session.

The default zone needs at least one auxiliary EPM configured for redundancy, where the Primary EPM is the alternate. Additional zones must have at least two Auxiliary EPMs configured for each zone to meet redundancy requirements for high availability.

Procedure

1. On the Primary EPM web administration, navigate to **System Configuration > EPM Servers** in the left navigation pane to open the **EPM Servers** page (Figure 5).
2. Click on the **Add** button to add a new auxiliary EPM.
3. Enter the appropriate information on the **Add EPM Server** page (Figure 6).
    - Ensure that the Default zone is selected in the **Zone** field.
    - Enter a descriptive name in the **Name** field.
    - Enter an IP address or resolvable hostname in the **Host Address** field.
4. Click **Continue**. The EPM Certificate will be fetched from the entered address and displayed. Verify that the certificate is correct.
5. Click the **Trust this certificate** check box to accept the certificate.

6. Click **Save**.

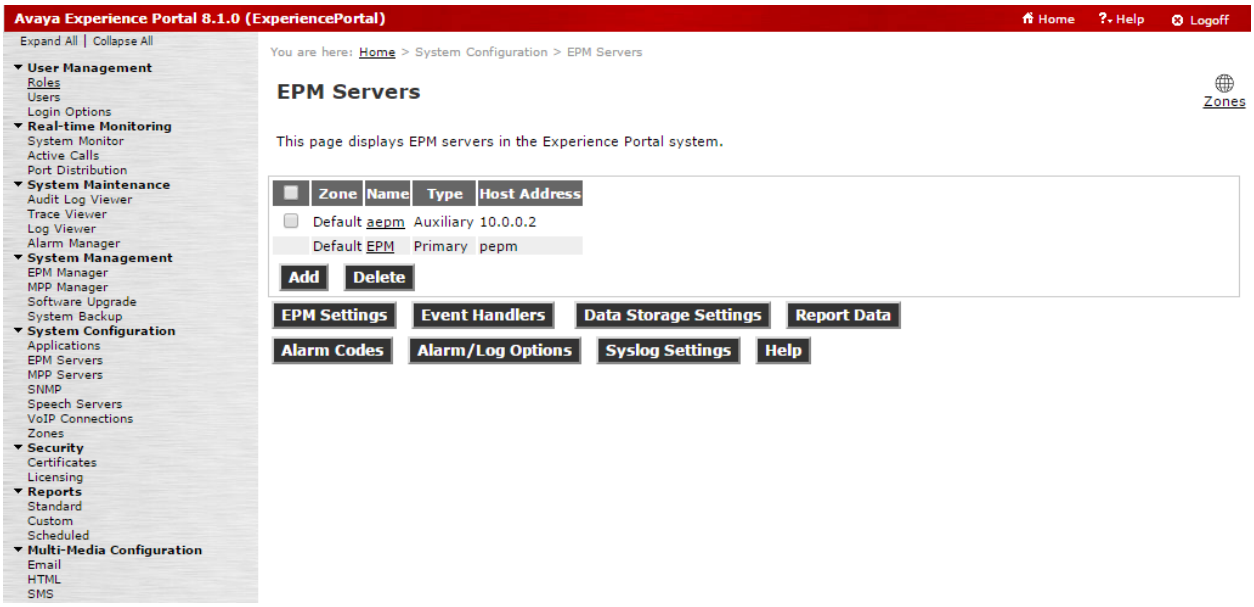

Figure 5 EPM Servers web page

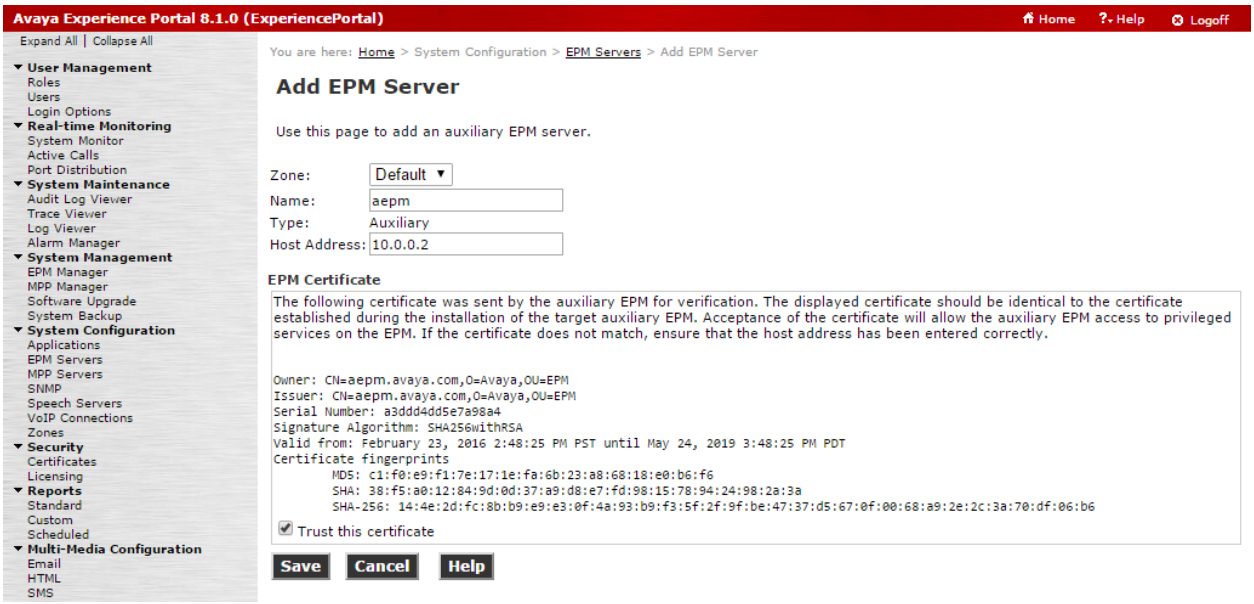

Figure 6 Add EPM Server web page

4.3. Importing Certificates

To enable TLS mutual authentication between the EPM servers and the OD App servers, certificates must be exchanged. For the EPM, the certificate associated with the OD App server load balancer's

virtual IP address will be imported. For the load balancers, the EPM's certificate must be included in the configured truststore (see load balancer configuration).

Procedure

1. On the Primary EPM web administration, navigate to **Security > Certificates** in the left navigation pane to open the **Certificates** page (Figure 7).
2. Click on the **Trusted Certificates** tab.
3. Click on the **Import** button to import a new certificate.
4. Fill in the fields on the **Import Trusted Certificate** page (Figure 8).
    - Enter a descriptive name for the certificate in the **Name** field.
    - In the **Location** field enter a URL that uses the https (Secure HTTP) scheme and includes the virtual IP address maintained by the load balancers (or a hostname that resolves as such).
5. Click **Continue**. The load balancer's certificate will be fetched from the entered address and displayed. Verify that the certificate is correct.
6. Click **Save** to accept the certificate.

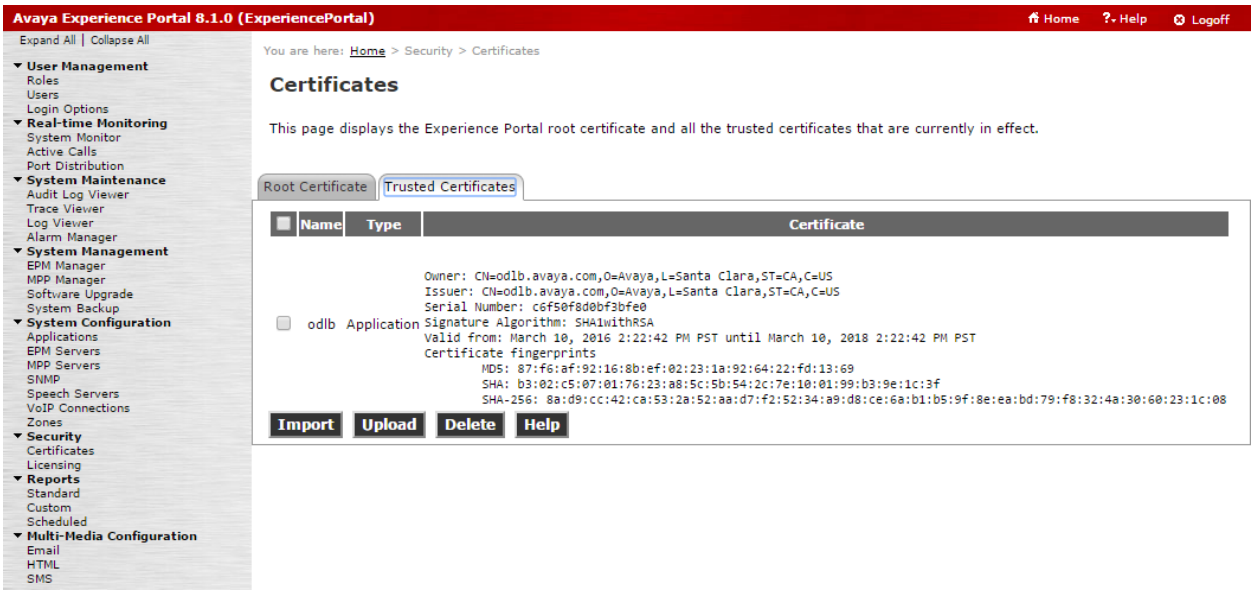

Figure 7 Certificates web page with the Trusted Certificates tab selected

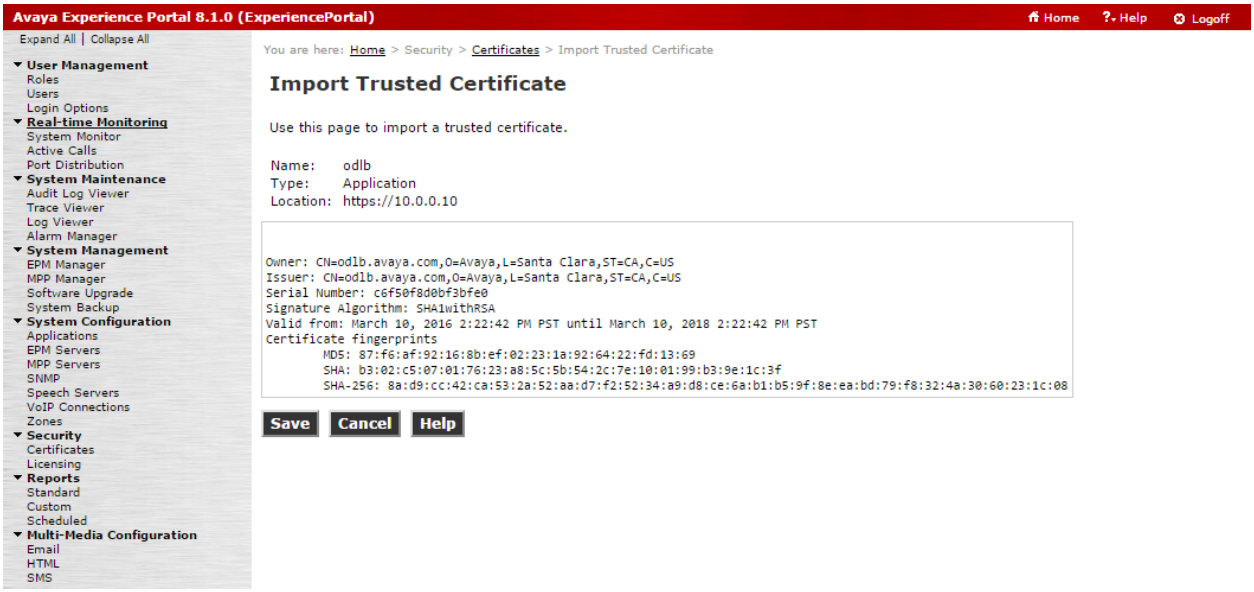

Figure 8 Import Trusted Certificate web page

4.4. Exporting the Root Certificate

The EPM root certificate is exported for later use in configuring the OD Application servers.

Procedure

1. On the Primary EPM web administration, navigate to **Security > Certificates** in the left navigation pane to open the **Certificates** page (Figure 7).
2. Click on the **Root Certificate** tab.
3. Click on the **Export** link to download the root certificate.

4.5. HTML Redirector Configuration

The OD runtime gets the redirector configuration information from global CAVs administered on the EPM. In particular, enabling the OD Generate URL to HTML Application connector requires configuring the HTML Redirectors page.

Procedure

1. On the Primary EPM web administration, navigate to **Multi-Media Configuration > HTML** in the left navigation pane to open the **HTML Redirectors** page (Figure 9).
2. Confirm that **TLS** is selected in the drop down box in the **Transport Protocol** column.
3. Fill in the appropriate fields.
    - Enter the IP address or resolvable hostname for the reverse proxy in the field in the **Host Address** column.

- In the field in the **Port** column, enter the port number used to connect to the reverse proxy or leave blank to use the default port.
- In the field in the **Path** column enter the path configured on the reverse proxy that maps to the load balancer configuration for the Redirectors (see Reverse Proxy configuration).

4. Click **Apply** to save the configuration.

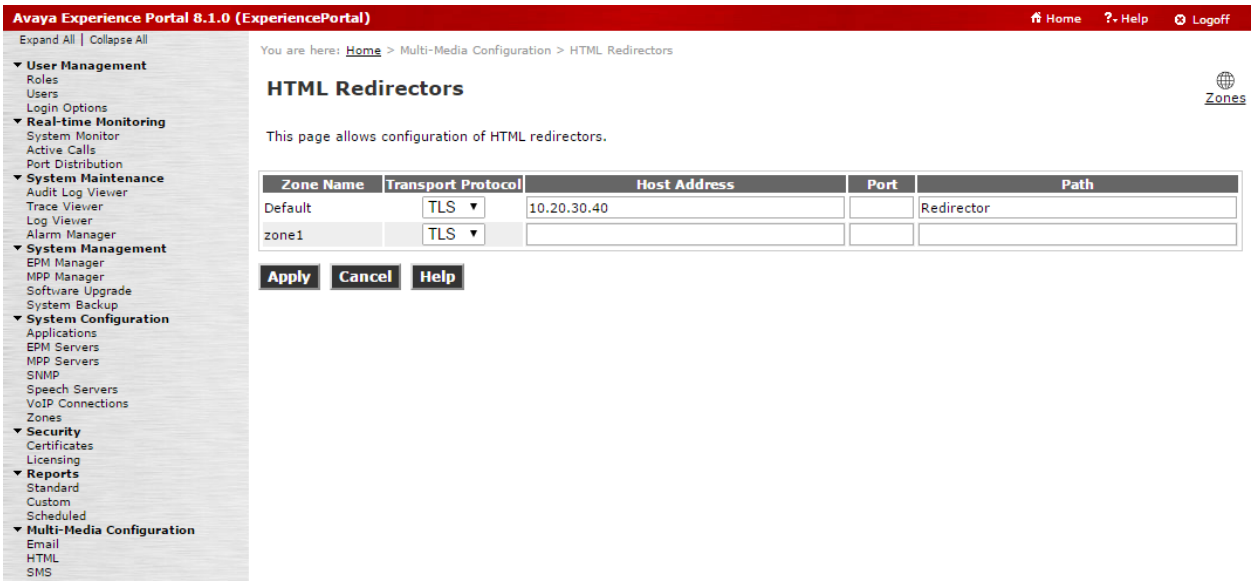

Figure 9 HTML Redirectors web page

5. Configuring Redirectors

The Redirector servlet hosts a generic URL that is the initial point of contact in launching OD Mobile Web applications. However, since configuration for launching a Mobile Web application is maintained by the Experience Portal system, starting a new session of an application is actually achieved through a web service request to an associated EPM. The Redirector servlet is a thin wrapper around this web service with the fundamental purpose of providing a layer of security by isolating the EPM servers and hiding any sensitive launch configuration (internal IP addresses, confidential query parameters, CAVs, etc.) from direct exposure to the external browser.

In the recommended configuration the Redirector WAR file will be deployed in the same Tomcat server used to host the OD Mobile Web application. However, the Redirector WAR file can be deployed in any Tomcat installation with consideration for server loading, although the Redirector servlet itself has very low resource requirements.

The Redirector includes a setup utility for configuring the list of EPM server addresses, usernames, passwords, priorities and managing certificates necessary for accessing the EPM's AppIntf web service to

launch a Mobile Web application instance. The setup utility stores the Redirector configuration in an encrypted binary file named servers.conf and certificates are written into the truststore.jks file.

At least two instances of the Redirector servlet must be deployed for high availability. For the recommended configuration one instance will be deployed on each of the two OD app servers.

5.1. Deployment

The Redirector.war file is exported from an OD installation using the "Export…" option on the File menu from Eclipse, choosing the Avaya OD Development -> Export HTML5 Redirector option. The WAR file can then be dropped into a Tomcat installation's webapp directory and, once deployed, the setup utility, configuration and log files are located in the Redirector's WEB-INF directory. The standard URL to access the Redirector and start an application is of the form:

<ADDRESS_TO_TOMCAT>/Redirector/<APP_NAME>

For example:

http://localhost:8080/Redirector/MyHTML5App

Mobile Web applications can choose to include additional query parameters on the URL, which will be sent to the application for use there.

NOTE: The Redirector will deploy successfully on Tomcat 6 and above with Java 7 or above. The latest versions of the Java runtime should be used on the application servers to enable use of the TLS v1.2 protocol which is required for connections to the EPM and Auxiliary EPMs.

5.2. Setup

Adding EPM servers and credentials is done by accessing the setup utility located in the WEB-INF directory. Launching either the setup.bat (Windows) or setup.sh (Unix) scripts will start the utility and display the current configuration:

```
[root@odapp1 WEB-INF]# bash setup.sh

Version 1.0.11

0-Timeout (ms): 5000
1-Log Enabled: false
2-Max Log Size (MB): 10

Configured Servers:

None.

(A)dd a server, (D)elete a server, (S)et a parameter, or (Q)uit
setup>
```

Figure 10 Initial configuration

Initially the configuration will have no servers configured as displayed in figure 1. Use the (A)dd command by entering "A" at the setup prompt to add a new server. Sequential prompts for the server address, priority, username and password will appear. The address can be either a hostname or IP address, optionally followed by a port number separated by a colon. The priority establishes the order in which the configured servers will be processed. Lower values for priority will be processed first and servers with equal priorities will be randomly ordered on each request (i.e. servers with equal priorities will be assigned an approximately equal number of requests). The displayed list of servers is sorted based on ascending order of priority. The username and password should match the values configured on the EPM for a user with the Web Services feature.

Once filled out, a test connection will be made to the EPM server with the supplied address and credentials. Additionally, checks are made to see if the necessary certificates are already in the trust store. If any new certificates are found when connecting to the server, the certificates are shown along with a prompt to confirm acceptance of the new certificates. If the test connection fails, a prompt to force acceptance of the configuration will be displayed. If the connection is successful and there are no new certificates, the configuration is automatically accepted and the setup prompt is again displayed.

A server configuration can be deleted using the (D)elete command from the setup prompt and entering the corresponding Id number when prompted.

Similarly, the value for one of the global Redirector parameters (Timeout, Logging, and Log Size) can be adjusted by using the (S)et command from the setup prompt, entering the parameter number, and entering the new value when prompted.

The Redirector logs activity to the log.txt file located in the WEB-INF directory, rolling the log files when the configured Max Log Size is hit. Only errors, non-302 responses and exceptions, will be stored in the log unless the "Log Enabled" flag is changed to true (the default value is false). If logging is enabled, all HTTP response codes for each request are logged, including 302 responses with the associated redirection URL.

Procedure

1. Deploy the Redirector.war file in the OD Application server's Tomcat installation.
2. Once the Redirector.war file is unpacked, navigate to the $CATALINA_BASE/webapps/Redirector/WEB-INF directory.
3. Start the setup utility using either the **setup.bat** (Windows) or **setup.sh** (Linux) script.
4. Enter "a" at the **setup>** prompt to add a new EPM server.
5. Fill in the appropriate fields as prompted (Figure 11).
    - Enter the IP address or resolvable hostname for the primary EPM at the **Enter address** prompt.
    - At the **Enter priority** prompt, type "0". For the recommended configuration the priorities of the EPM servers will be set the same so that the Redirector will load balance between the primary and auxiliary EPM.

- At the **Enter username** prompt, type the name of the user on the EPM configured with the Web Services feature.
- Enter the password for this user at the **Enter password** prompt.

6. The certificate for the EPM will be fetched and displayed.
   (Note: if the system is in FIPS mode, fetching certificate would fail in which case skip step 7.
   Once Setup is completed, refer to **Managing Certificates for FIPS Mode** section below.)
7. Type "y" to accept the certificate and add the server to the configuration.
8. Repeat steps 4-7 for adding the auxiliary EPM.
9. Type "q" at the **setup>** prompt to quit the setup utility and save the configuration.
10. Enter "y" at the **Write new configuration to disk** prompt.
11. Repeat steps 1-10 for the second OD Application server.

```
setup> a
Enter address (hostname or IP): 10.0.0.2
Enter priority (0-65535): 0
Enter username: redirector
Enter password:

New certificates:

Subject: CN=scaaep134,O=Avaya,OU=EPM
Issuer: CN=scaaep134,O=Avaya,OU=EPM
Serial Number: 8d492e702d6aad12
Signature Algorithm: SHA256withRSA
Valid from Mon Sep 07 22:33:50 PDT 2015 until Thu Sep 04 22:33:50 PDT 2025
Certificate fingerprints
        MD5: 31a1 8b3a b54b 0123 11b1 c33d 2f5b 63f3
        SHA: 0e4c 923a 41df 0bc0 6a82 8181 3e2d 36ce 0796 7152

Accept the new certificate(s) and add this server configuration? [Y/n]:


0-Timeout (ms): 5000
1-Log Enabled: false
2-Max Log Size (MB): 10

Configured Servers:

Id  Address                                         Pri   Username

0   10.0.0.1                                        0     redirector
1   10.0.0.2                                        0     redirector

(A)dd a server, (D)elete a server, (S)et a parameter, or (Q)uit
setup>
```

Figure 11 Adding an EPM server using the Redirector setup utility

Managing Certificates for FIPS Mode

If the system is configured in FIPS mode, certificates can not be fetched from remote systems. Step 6 in the setup procedure above would fail. The certificates have to be manually added the Redirector's trust store file. To obtain the necessary certificate from EPM, it is convenient to use the internet browser functionality by navigating to the EPM site and exporting the certificate exposed by the browser. The certificate can then be added to the Redirector's trust store file which is located in the Redirector root directory. Once the certificates are imported, the following command can be run to convert the trust store file into format supported by Bouncy Castle.

```
keytool -importkeystore -srckeystore truststore.jks -destkeystore truststore.bks -srcstoretype JKS -deststoretyp
providerpath bc-fips-1.0.2.jar -provider org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

This command can be used to import another certificate:

```
Keytool -keystore truststore.bks -import -alias epm -file epm.cer -noprompt -storepass changeit -storetype BCFKS -pro
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider
```

Finally, rename the new file truststore.bks to truststore.jks since the Redirector app only recognizes one name.

6. Configuring OD Application Servers

Two configuration changes must be made on each of the OD Application servers. First, the Tomcat jvmRoute parameter must be enabled for the load balancer to operate correctly. Finally, Tomcat should be configured for mutually authenticated TLS, including setup of the necessary certificates for authentication.

6.1. Adding jvmRoute

The jvmRoute feature of the Tomcat server is used to ensure that requests associated with a particular session of the Mobile Web application are always routed to the same OD Application server.

Procedure

1. Edit the **server.xml** file for each of the OD Application server Tomcat installations, adding a jvmRoute attribute to each installation's Engine element as shown in Figure 12.

```
<Engine name="Catalina" defaultHost=localhost" jvmRoute="node1">    for the first server
<Engine name="Catalina" defaultHost=localhost" jvmRoute="node2">    for the second server
```

Figure 12 jvmRoute configuration

6.2. Configuring mutually authenticated TLS

To enable mutually authenticate TLS on the OD Application server, the Tomcat configuration must be slightly altered and the proper certificate files must be created.

Procedure

1. Edit the **server.xml** file, adding a TLS **Connector** to the **Service** section (Figure 13).
2. Initialize the keystore and truststore files in the $CATALINA_BASE/conf directory (Figure 14).
    - Use the Java keytool program to create the keystore file by importing the PEM encoded certificate and key created for the OD Application server. The supplied certificate must be in the PKCS#12 format.
    - Use the Java keytool program to create the truststore file by importing the PEM encoded certificate for the Load Balancer's virtual IP address.
    - Use the Java keytool program to add the EPM root certificate to the truststore file by importing the PEM encoded certificate exported from the Certificates page (see Exporting Certificates).
3. Repeat steps 1 and 2 on the second OD Application server.

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
       maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
       clientAuth="true" sslProtocol="TLS"
       keystoreFile="conf/keystore" keyPass="changeit"
       truststoreFile="conf/truststore" truststorePass="changeit"/>
```

Figure 13 TLS Connector configuration

```
[root@odapp1 conf]# keytool -importkeystore -srckeystore odapp1.p12 -srcstoretype
pkcs12 -destkeystore keystore –deststorepass changeit
Enter source keystore password:
Entry for alias 1 successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or
cancelled
[root@odapp1 conf]# keytool -import -noprompt -alias odlb -file odlb.avaya.com.crt
-keystore truststore -storepass changeit
Certificate was added to keystore
[root@odapp1 conf]# keytool -import -noprompt -alias epm -file sipCA.pem -keystore
truststore -storepass changeit
Certificate was added to keystore
[root@odapp1 conf]#
```

Figure 14 Certificate file creation

7. Configuring the Load Balancers

Apache HTTP is used to route requests to the appropriate OD Application server, based on a token embedded in each request.

7.1. Administering Apache HTTP

A Virtual Host configuration for the virtual IP address is added to the Apache HTTP configuration to enable load balancing between the OD Application servers.

Procedure

1. Add a virtual host configuration to the Apache HTTP configuration (Figure 15) on each of the load balancer servers with appropriate information.
    - The address in the **<VirtualHost>** tag should match the virtual IP address.
    - The list of **BalancerMembers** in the **<Proxy>** section should include the IP addresses for each of the OD Application servers and the **route** parameter for each should match the jvmRoute parameter configured on the corresponding OD Application server.
    - The paths in the **<Location>** tags for the Redirector and application should match the URLs forwarded from the reverse proxy. Similarly, the path in the ProxyPass and ProxyPassReverse lines for each should match the locations of the Redirector servlet and application on the OD Application servers.
    - The SSL configuration in the virtual host should be setup to require mutual authentication with the reverse proxy. **SSLVerifyClient** should be set to **require** and the **SSLCACertificateFile** line should point to a file that includes the PEM encoded certificate for the reverse proxy. Additionally, the **SSLCertificateFile** and **SSLCertificateKeyFile** lines should point to files for the certificate and key of the certificate for the Load Balancer's virtual IP address.
    - The SSL configuration should also be setup for mutual authentication with the OD Application servers. The **SSLProxyEngine** line should be set to **on**, the **SSLProxyVerify** line set to **require**, and the **SSLProxyCACertificateFile** line should point to a file that includes the PEM encoded certificates for the OD Application servers. Finally, The SSLProxyMachineCertificateFile should point to a file that is the concatenation of the PEM encoded certificate and key of the certificate for the Load Balancer's virtual IP address.

```
<VirtualHost 10.0.0.10:443>
        ErrorLog logs/ssl_error_log
        TransferLog logs/ssl_access_log
        LogLevel warn
        SSLEngine on
        SSLVerifyClient require
        SSLCACertificateFile /etc/pki/tls/certs/revp.crt
        SSLProtocol all -SSLv2 -SSLv3
        SSLCipherSuite DEFAULT:!EXP:!SSLv2:!DES:!SEED
        SSLCertificateFile /etc/pki/tls/certs/odlb.avaya.com.crt
        SSLCertificateKeyFile /etc/pki/tls/private/odlb.avaya.com.key
        CustomLog logs/ssl_request_log \
                "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

        ServerName odlb.avaya.com
        SSLProxyEngine on
        SSLProxyCACertificateFile /etc/pki/tls/certs/odapp.crt
        SSLProxyVerify require
        SSLProxyMachineCertificateFile /etc/pki/tls/private/odlb.avaya.com.pem

        <Proxy balancer://appcluster>
                BalancerMember https://10.0.0.3:8443 route=node1
                BalancerMember https://10.0.0.4:8443 route=node2
        </Proxy>

        <Location /Redirector>
                ProxyPass balancer://appcluster/Redirector
                ProxyPassReverse balancer://appcluster/Redirector
        </Location>

        <Location /IceCream_V2>
                ProxyPass balancer://appcluster/IceCream_V2 \
                        stickysession=JSESSIONID|jsessionid \
                        scolonpathdelim=On lbmethod=bytraffic
                ProxyPassReverse balancer://appcluster/IceCream_V2
        </Location>
</VirtualHost>
```

Figure 15 Apache HTTP Virtual Host configuration

7.2. Configuring a Virtual IP address with keepalived

Using keepalived to maintain a virtual IP address between the two Load Balancer servers ensures that requests to Apache HTTP are always serviced by one of the servers.

Procedure

1. Modify the keepalived configuration on each load balancer to add a Virtual IP address and monitor Apache HTTP (Figure 16).
    - The **priority** for the first server should be higher (101) than the value for the second server (100).

```
vrrp_scrip chk_apache {
    script "killall -0 httpd"
    interval 2
}

vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    virtual_router_id 51
    priority 101
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 5237
    }
    virtual_ipaddress {
        10.0.0.10
    }
    track_script {
        chk_apache
    }
}
```

Figure 16 keepalived configuration

8. Configuring the Reverse Proxy

The reverse proxy interfaces with the external mobile web browser to forward HTTP requests to the Load Balancer. Two URL mappings are required for a Mobile Web application to function. The first is a mapping to the Redirector location on the Load Balancer. This URL will be used by the mobile web browser to launch a new instance of the Mobile Web application and should be consistent with the configuration for the OD Generate URL to HTML Application Pluggable Data Connector (see HTML Redirector configuration). The second URL mapping points to the application location on the Load Balancer. This is the location that will be redirected to after the Mobile Web application instance is started. Consequently, this must be consistent with the External URL configured for the Mobile Web application on the EPM (see Adding a Mobile Web Application).

The reverse proxy mappings should be configured to use mutually authenticated TLS connections to the Load Balancer.

Although no specific recommendation for a reverse proxy is made in this guide, Figure 17 shows an example of an Apache HTTP server configuration that map the two necessary URLs to the Load Balancer's virtual IP address using a mutually authenticated TLS connection. It is expected that any reverse proxy with equivalent functionality can be used though.

```
SSLProxyEngine On
SSLProxyCACertificateFile /etc/pki/tls/certs/odlb.avaya.com.crt
SSLProxyVerify require
SSLProxyMachineCertificateFile /etc/pki/tls/private/revp.avaya.com.pem

<Location /Redirector>
        ProxyPass https://10.0.0.10/Redirector
        ProxyPassReverse https://10.0.0.10/Redirector
</Location>

<Location /IceCream>
        ProxyPass https://10.0.0.10/IceCream_V2
        ProxyPassReverse https://10.0.0.10/IceCream_V2
</Location>
```

Figure 17 Example Reverse Proxy configuration

8.1. URL Minimization

Applications using the OD Generate URL to HTML Application Pluggable Data Connector (PDC) may wish to control the length of the generated URL. Especially for SMS applications using the PDC, the generated URL can be very long unless some configuration is carefully chosen. In particular, the reverse proxy facilitates this utilizing URL rewriting.

The Generate URL PDC concatenates several pieces of the configuration to construct the URL used to start a Mobile Web application instance:

<scheme>://<host address>[:<port>][/<path>]/<zone>%3A<app name>[/c<conversation id>][/u<ucid>]

- <scheme> is either “http” or “https” depending on the Transport Protocol setting on the HTML Redirectors page
- <host address> is the value configured in the Host Address field on the HTML Redirectors page
- <port> in the value configured in the Port field on the HTML Redirectors page and is omitted if it is empty
- <path> is the value configured in the Path field on the HTML Redirectors page and is omitted if it is empty
- <zone> is a numeric value representing the applications configured zone
- <app name> is the name of the Mobile Web application, as configured in the Name field on the Change Application page
- <conversation id> is a unique token used to reference the application’s contextual data in the platform’s Conversation Storage. This is omitted if the application doesn’t start a Conversation.
- <ucid> is the Universal Call ID associated with the instance of the application. This is omitted if the applications sets session:ucid to the empty string (“”).

Minimizing the URL then first requires leaving the Port and Path fields empty on the HTML Redirectors page and the Application name configured on the Change Application page should be as short as possible, but must be a unique path in the reverse proxy mapping. Additionally, the application shouldn’t utilize Conversations or UCID. With this the PDC would generate a URL like the following:

https://10.20.30.40/0%3AIceCreamWebStore

With URL rewriting in the reverse proxy this URL could still be mapped to the Redirector on the Load Balancer. For example, an Apache HTTP location mapping like the following would forward this request to https://10.0.0.10/Redirector/0%3AIceCreamWebStore instead:

```
<Location /0%3AIceCreamWebStore>
    ProxyPass https://10.0.0.10/Redirector/0%3AIceCreamWebStore
    ProxyPassReverse https://10.0.0.10/Redirector/0%3AIceCreamWebStore
</Location>
```

URL rewriting can also be used to normalize the URL that the application is redirected to. An example of this is seen in the Figure 17 example configuration where a "/IceCream" URL would be mapped to a "/IceCream_V2" URL on the Load Balancer.

9. Firewall Configuration

For the recommended configuration, firewall configuration is deliberately simple. The external firewall must allow HTTP (i.e. port 80) and/or HTTPS (i.e. port 443) traffic from the Internet to flow to the reverse proxy IP address. The internal firewall must allow HTTPS (i.e. port 443) traffic to flow only from the reverse proxy IP address to the load balancer virtual IP address.