

Email Notification Service 2 (ENS2)

VMware Workspace ONE UEM



vmware®

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2020 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

1	Introduction	4
	ENS2 Architecture	4
	ENS2 Deployment Options	5
	ENS2 Requirements and Prerequisites	12
2	Configuring ENS2 on Cloud and On-Premises Deployment	17
	Email Notification Service for Cloud	17
	ENS Endpoints and IP Whitelist	18
	Verify VMware Boxer Settings	18
	Email Notification Service for On-Premises	19
	Configure CNS and Download Email Notification Service Configuration Files	19
	Install and Upgrade Email Notification Service 2	20
	Upgrade ENS2	26
	Configure Workspace ONE Boxer for On-Premises	26
	ENS2 Application Configuration Keys for Boxer	26
	Migrating from ENS On-Premises Server to Cloud Server	28
	Configuring SEG as the EWS Proxy for ENS	28
	Configure ENS2 with SEG	30
	Configure SEG for Authentication	30
	Enable Certificate-Based Authentication for ENS	31
	Configure ENS2 for Certificate-Based Authentication	32
	Configure Certificate-Based Authentication on the Exchange Server	32
	Using Office 365 with ENS2 and Certificate-Based Authentication	34
3	Troubleshooting ENS	36
	ENS2 Resubscription and Badge Count Accuracy Limitations	36
	Troubleshooting Accessibility Issues to the ENS Server	38
	Troubleshooting the Console Configuration Issues	43
	Troubleshooting ENS2 Notification Issues	44
	Troubleshooting Connection Issues to the ENS Database	72
	Troubleshooting SSL Errors	73
	ENS2 Response Code and Error Code Details	76
4	Frequently Asked Questions	80

Introduction

1

Workspace ONE UEM powered by AirWatch ENS adds push notification support to Exchange.

The Workspace ONE Boxer provides notifications about your emails by running in the background. Due to platform limitations, Boxer can only run in the background for a limited time. Email Notification Service (ENS2) provides a solution to deliver notifications to the user's device when Boxer is not running.

ENS2 supports notifications that include the email subject and a badge icon (iOS only) to notify the number of unread emails in the Inbox on the server. However, for Android, ENS2 does not support notifying the number of unread emails in the Inbox on the server.

ENS2 can be configured with the Secure Email Gateway (SEG) V2 to secure your organization's email infrastructure. For more information about SEG, see the *Workspace ONE UEM Secure Email Gateway Guide (SEG) V2* guide.

This documentation provides the information required to install and configure the ENS2 as a cloud-hosted or on-premises service.

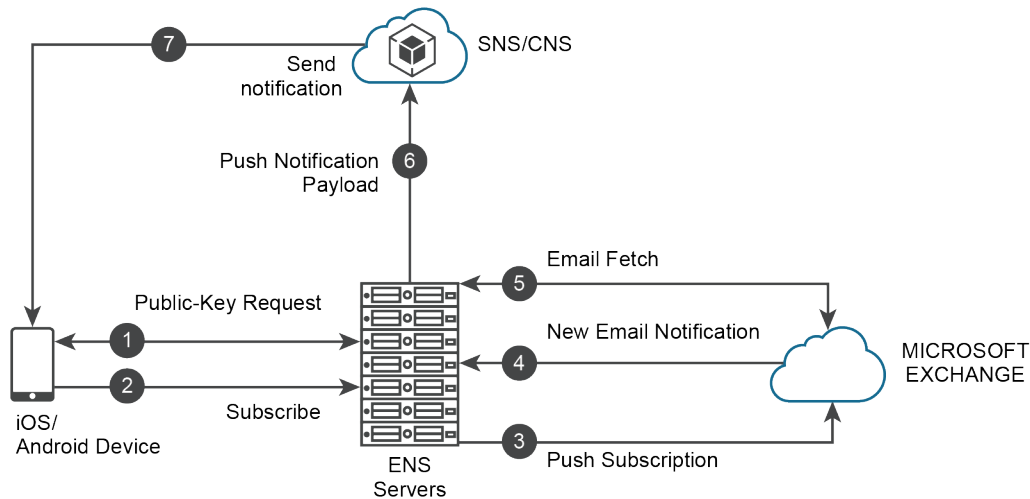
This chapter includes the following topics:

- [ENS2 Architecture](#)
- [ENS2 Deployment Options](#)
- [ENS2 Requirements and Prerequisites](#)

ENS2 Architecture

This section provides information about the architecture design and functionality of ENS2.

ENS2 Architectural Flow



ENS2 architectural flow description:

- 1 Public-Key Request - The device requests a public key to encrypt the account credentials.
- 2 Subscribe - The device sends an encrypted payload with credentials and all the necessary information to subscribe and get email notifications.
- 3 Push Subscription - ENS authenticates with EWS and subscribes for push notifications using a webhook URL. The webhook URL contains the encrypted credentials. The credentials are now kept encrypted on the Exchange server.
- 4 New Email Notification -
 - Exchange sends notification about the mailbox changes to the provided webhook URL.
 - ENS extracts and decrypts the credentials and prepares a call to fetch emails.
- 5 Email Fetch - ENS performs a fetch for the email details (subject and sender) required for providing a notification.
- 6 Push Notification Payload - ENS pushes email details for delivery to all devices belonging to the user through SNS (ENS cloud deployments) or CNS (ENS on-premises deployments).
- 7 SNS or CNS sends notifications to iOS or Android devices. For iOS devices, SNS or CNS uses Apple Push Notification Service (APNS), and for android devices, SNS or CNS uses Firebase Cloud Messaging (FCM).

ENS2 Deployment Options

The Email Notification Service (ENS2) can be deployed on a cloud-hosted service or hosting your own ENS instance an on-premises installation.

The various deployment methods for ENS2 are explained in the following sections.

- Deploying ENS2 on a Cloud-hosted Service with Office 365 or on an On-Premises Exchange

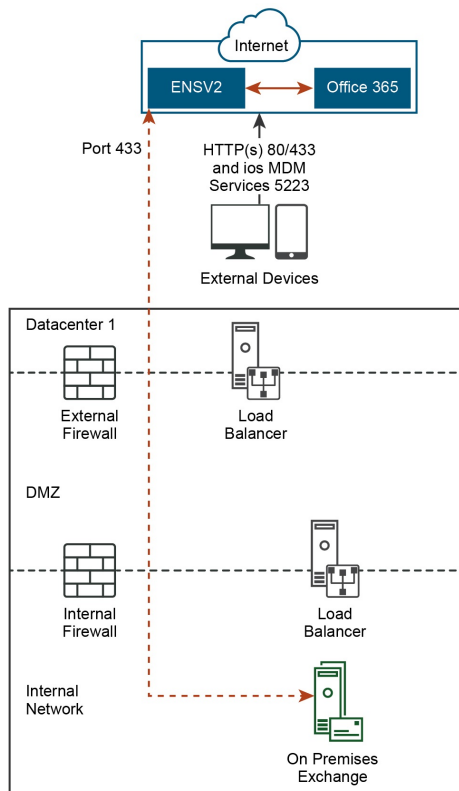
- Deploying ENS2 on a Cloud-hosted Service with Office 365 or on an On-Premises Exchange with SEGV2 Proxy
- Deploying On-Premises ENS2 with Office 365 or Exchange in a Single and Multidata Center
- Deploying On-Premises ENS2 with SEGV2 as the EWS Proxy for Office 365 or Exchange in a Single and a Multidata Center

Note Deploy ENS2 through a cloud-hosted service unless there is a specific requirement to deploy ENS2 an on-premises installation.

Deploying ENS2 on a Cloud-hosted Service with Office 365 or on an On-Premises Exchange

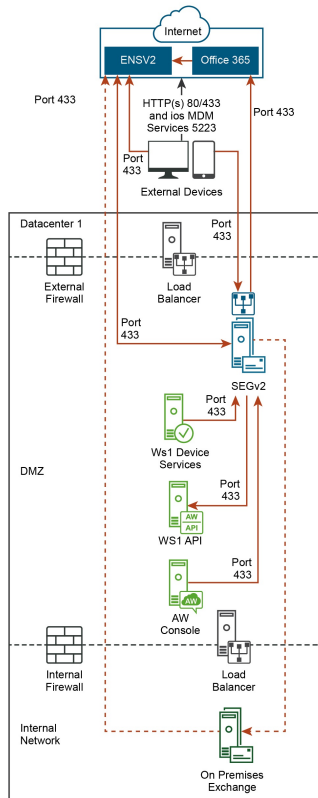
In this deployment scenario, the ENS2 and Office 365 are in a cloud-hosted environment. The external devices such as iOS, Android, and so on, can access the ENS2 and Office 365 through port 443 to subscribe to the ENS2 and get email notifications.

In an on-premises Exchange setup, as shown in the following topology, the on-premises Exchange server can access the ENS2 and Office 365 through port 443 to subscribe to the ENS2 and get email notifications. ENS2 and the Exchange server interact with each other over port 443 through the EWS protocol.



Deploying ENS2 on a Cloud-hosted Service with Office 365 or on an On-Premises Exchange with SEGV2 Proxy

In this deployment scenario, ENS2 and Office 365 are in a cloud-hosted environment and the ENS2 can be configured with the SEGV2 to secure your organization's email infrastructure. When an external device initiates a registration request to the ENS2, the ENS2 sends the request to SEG, and then request is routed to the Office 365. Any email exchanges or push notifications are routed through the SEG proxy.

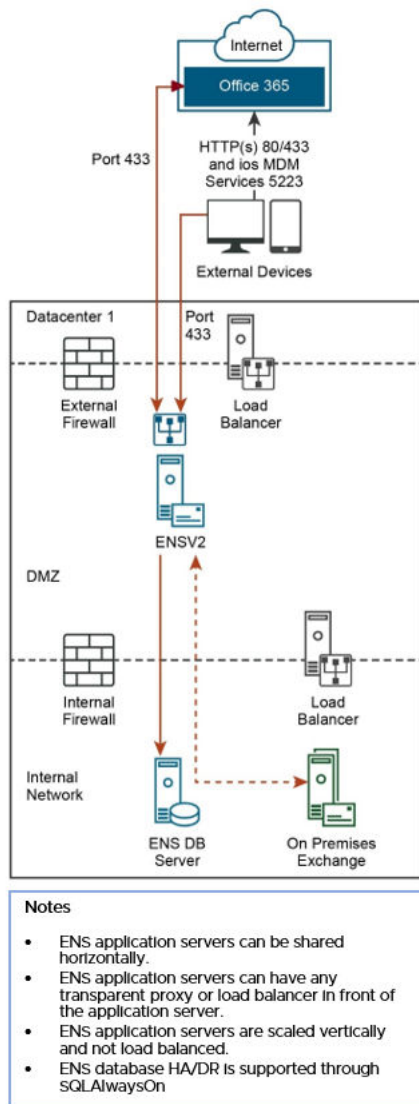


On an on-premises setup, all traffic from the ENS2 to the Exchange is routed through the SEG v2. However, the Exchange server can directly interact with the ENS2.

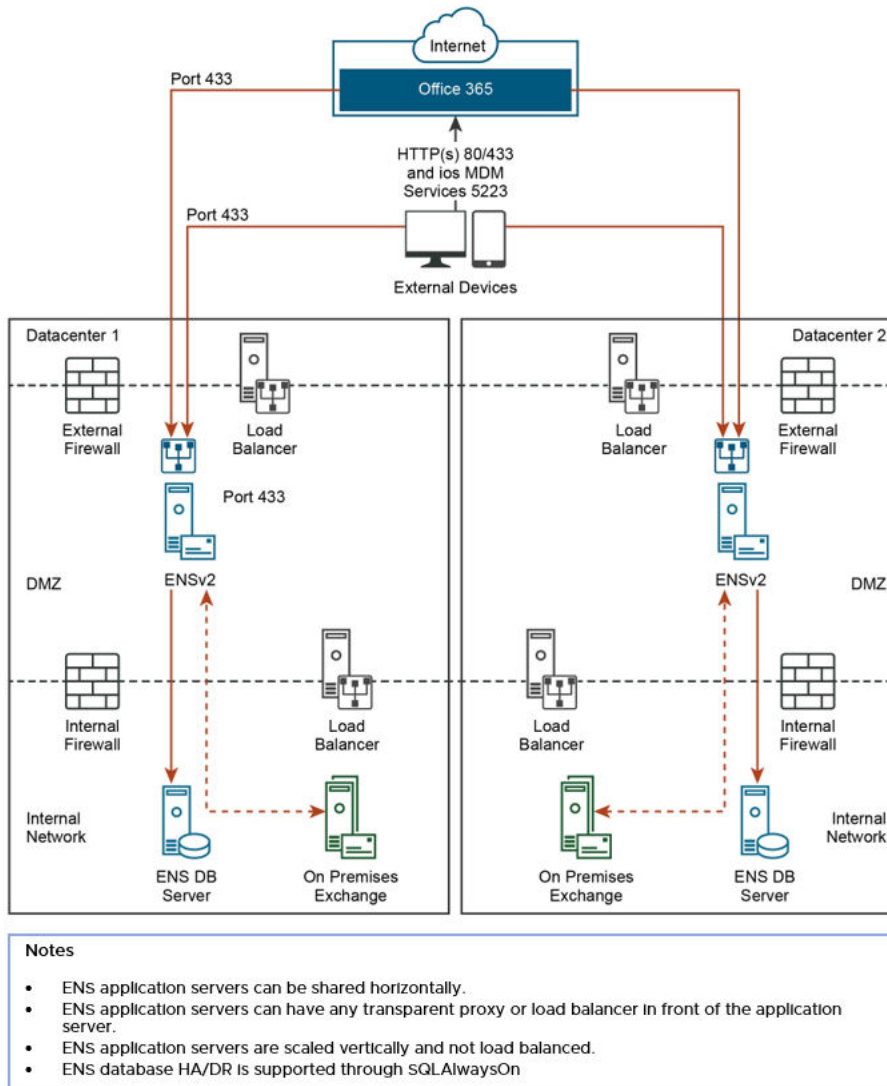
Deploying On-Premises ENS2 with Office 365 or Exchange in a Single and Multidata Center

In a single data center ENS2 deployment scenario, as shown in the following topology diagram, the ENS2 is hosted on an on-premises network within the DMZ zone so that the ENS2 is publicly accessible. The external devices such as iOS, Android, and so on, have access to ENS2 through port 443 to subscribe to the ENS2 and get email notifications.

The ENS database server can be hosted on the on-premises network behind the internal firewall and the ENS2 can communicate with the ENS database through the internal firewall. The ENS database server can be scaled vertically to upgrade the capacity of the existing ENS database server.



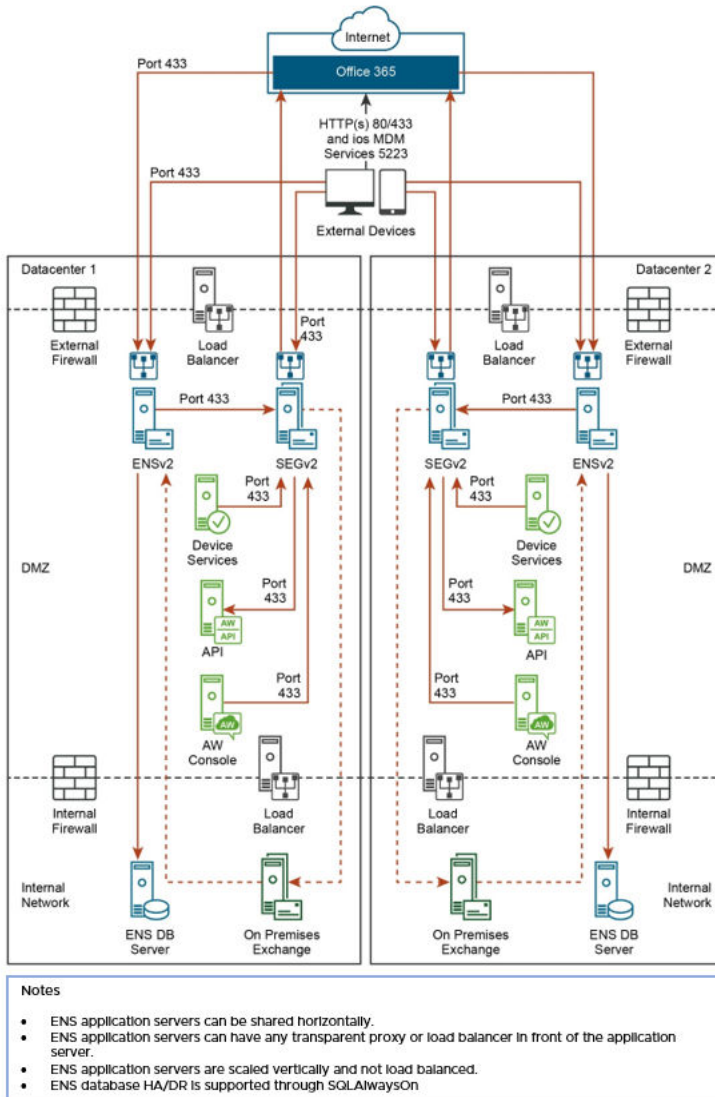
The following topology shows the ENS2 deployed in a multidata center, where there might be more than one data center to support a failover. In every data center, for each instance of the ENS, there is always a paired instance of the ENS database and each ENS database can host their own data. In case, the data center 1 fails then the data center 2 becomes active to support failover scenarios.



Deploying On-Premises ENS2 with SEGV2 as the EWS Proxy for Office 365 or Exchange in a Single and a Multidata Center

In this deployment scenario, the ENS2 is hosted on-premises and the SEG is installed in between the external devices and the on-premises Exchange. All the EWS traffic coming from the external devices must pass through the SEGV2 and then reach the on-premises Exchange. However, the on-premises Exchange can directly communicate with the ENS2.





Difference between ENS2 Cloud-hosted Deployment and ENS2 On-Premises Deployment

The following table describes the benefits and limitations of deploying ENS2 through a cloud-hosted service and an on-premises deployment.

ENS2 Cloud-hosted Deployment	ENS2 On-Premises Deployment
<p>Benefits of deploying ENS2 through a cloud-hosted service:</p> <ul style="list-style-type: none"> ■ Easiest method of deployment as no infrastructure or maintenance is required. ■ Easily scalable as you can automatically scale up to meet the increasing demands of the user. ■ ENS2 supports the Office 365 cloud strategy deployments. <p>Limitations of deploying ENS2 through a cloud-hosted service:</p> <ul style="list-style-type: none"> ■ ENS2 requires an internet-facing or proxied EWS endpoint (can be restricted to IP ranges) and the email data flows outside the organization network. 	<p>Benefits of deploying ENS2 on-premises:</p> <ul style="list-style-type: none"> ■ Controls the upgrade cadence and can be deployed to the DMZ without exposing the Exchange Web Services (EWS). <p>Limitations of deploying ENS2 on-premises:</p> <ul style="list-style-type: none"> ■ Requires additional manual installation and maintenance of the ENS2 and the CNS components. ■ Requires periodic updates to stay updated. ■ Environment scaling requires additional setup and maintenance. ■ High availability requires additional installation and manual resource allocation. ■ Requires additional licensing (Microsoft Windows Server and Microsoft SQL Server) and hardware.

ENS2 Requirements and Prerequisites

This section explains the requirements and prerequisites for using the ENS2 with Workspace ONE UEM.

Email Server Integration Supported Versions

- Email Client - For Android support, you must have ENS2 1.3.0.4 or later and Workspace ONE Boxer 5.2 or later.
- Email Server - Exchange 2010 SP3, Exchange 2013 SP1, Exchange 2016, Exchange 2019 (for on-premises ENS2 version 1.7 and later), or Office 365.
- For ENS2 on-premises with ENS2 version 1.8 and later, Office 365 is supported.

Workspace ONE UEM Requirements

- On-premises and Cloud deployment: Workspace ONE UEM console 1902 and later

Hardware Requirements (On-Premises Only)

Table 1-1. Web Server

CPU Core	RAM	Hard Disk Storage	Notes
2 (Intel processor)	16 GB	30 GB	Per 100,000 users.

Table 1-2. Database Server

CPU Core	RAM	Hard Disk Storage	Notes
2 (Intel processor)	16 GB (minimum)	Approx. 0.0477 MB per user to estimate the DB storage size.	Per 100,000 users.

Software Requirements

For ENS2 version 1.7 and later, you must upgrade your CNS from CNS v1.0 to CNS v2.0 to support notifications.

Requirement (On-Premises)	Notes
Windows Server 2008 R2 or Windows Server 2012 R2 or Windows Server 2016	The servers should be externally accessible via https (SSL Cert) and with a Fully Qualified Domain Name (FQDN)
SQL Server 2016, 2017, and 2019 (Database Server)	<p>The db_owner role and public role must be assigned to the SQL server user that is used for running the application. The database option must be selected for external database and you must set the collation to SQL_Latin1_General_CP1_CI_AS. VMware recommends a dedicated SQL instance for ENS. The steps to create an ENS database and the Workspace ONE UEM database are the same. For more information on creating the Workspace ONE UEM database, see <i>Create the Workspace ONE UEM Database</i> topic in the <i>Installing Workspace ONE UEM</i> guide.</p> <p>Note A shared SQL instance can only be used for demonstration purpose, where a small set of users can use the ENS.</p>
Basic Authentication for the Exchange environment	OAuth and Certificate Based Authentication (CBA) is supported for Exchange Web Services
CNS Certificate	
Secure Channel Certificate	
IIS 7 or later	Installed on Web Server
Requirement (Cloud)	Notes
Basic Authentication for the Exchange environment	OAuth and Certificate Based Authentication (CBA) is supported for Exchange Web Services
Autodiscovery enabled in Exchange environment and Internet-facing EWS environment. If autodiscovery is disabled, you can use the EWSUrl key value pair to configure ENS.	

Networking Requirements

Table 1-3. Network Ports

Source	Destination	Protocol (Port)
ENS	Exchange (EWS)	HTTPS (443)
Exchange (EWS)	ENS	HTTPS (443)
Mailbox/CAS	ENS	HTTPS (443)
ENS	AirWatch Cloud Notification Service (CNS)	HTTPS (443)
ENS	SQL Server Instance	SQL (1433)
Internet (Devices)	ENS	HTTPS (443)

Table 1-4. IIS Services

Component Name	Required Services
Web Management Tools	IIS 6 Management Compatibility
IIS Management Console	
IIS Management Scripts and Tools	
IIS Management Service	

Table 1-5. World Wide Web Services

Component Name	Required Services
Application Development Features	.NET Extensibility 3.5
.NET Extensibility 4.6	
Application Initialization	
ASP	
ASP.NET 3.5	
ASP.NET 4.6	
ISAPI Extensions	
ISAPI Filters	
Server-Side Includes	
WebSocket Protocol	
Common HTTP Features	Default Document
Directory Browsing	
HTTP Errors	
Static Content	
Health and Diagnostics	HTTP Logging
Performance Features	Static Content Compression
Security	Request Filtering

SQL Server and High Availability Support

High availability configuration - ENS2 supports **SQL Server AlwaysOn** high availability configuration. To set up the **SQL Server AlwaysOn** for active/active or active/passive setup, see [Overview of Always On Availability Groups \(SQL Server\)](#). If you are using AlwaysOn, point to the availability group when choosing the database server during ENS2 installation.

TLS Support for ENS

ENS supports TLS version 1.0 to TLS version 1.3. ENS does not choose any protocol, but allows the OS to choose the strongest available TLS version and the cipher suites. The following table lists the recommended cipher suites.

Cipher Suites	SSL Cipher Strength	TLS Protocol Version	Elliptic Curve Variants	Cryptographic Algorithm	Authenticated Encryption	Cryptographic Hash Algorithm
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	ECDHE-ECDSA-AES128-GCM-SHA256	TLS 1.2	ECDH-ephemeral	ECDSA	AESGCM (128)	SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	ECDHE-ECDSA-AES256-GCM-SHA384	TLS 1.2	ECDH-ephemeral	ECDSA	AESGCM (256)	SHA256 and SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	ECDHE-ECDSA-AES128-SHA	TLS 1.2	ECDH-ephemeral	ECDSA	AES (128)	SHA1
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE-ECDSA-AES256-SHA	TLS 1.2	ECDH-ephemeral	ECDSA	AES (256)	SHA1
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	ECDHE-ECDSA-AES128-SHA256	TLS 1.2	ECDH-ephemeral	ECDSA	AES (128)	SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	ECDHE-ECDSA-AES256-SHA384	TLS 1.2	ECDH-ephemeral	ECDSA	AES (256)	SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDHE-RSA-AES128-GCM-SHA256	TLS 1.2	ECDH-ephemeral	RSA	AESGCM (128)	SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDHE-RSA-AES256-GCM-SHA384	TLS 1.2	ECDH-ephemeral	RSA	AESGCM (256)	SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDHE-RSA-AES128-SHA	TLS 1.2	ECDH-ephemeral	RSA	AES (128)	SHA1
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	ECDHE-RSA-AES256-SHA	TLS 1.2	ECDH-ephemeral	RSA	AES (256)	SHA1

Cipher Suites	SSL Cipher Strength	TLS Protocol Version	Elliptic Curve Variants	Cryptographic Algorithm	Authenticated Encryption	Cryptographic Hash Algorithm
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	ECDHE-RSA-AES128-SHA256	TLS 1.2	ECDH-ephemeral	RSA	AES (128)	SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDHE-RSA-AES256-SHA384	TLS 1.2	ECDH-ephemeral	RSA	AES (256)	

ENS2 Prerequisites

To enable and secure communication between the Exchange server and the ENS server, note the following points:

- Communication between ENS and Exchange servers should not have any SSL errors.
- telnet and ping commands should work seamlessly between ENS and Exchange CAS/Mailbox servers.
- SSL certificates used for ENS and Exchange servers must not have any errors when they run through SSL checkers.

Note If you want to enable certificate-based authentication or configure ENS2 with SEG, see [Enable Certificate-Based Authentication for ENS](#) and [Configuring SEG as the EWS Proxy for ENS](#).

Upload the Root CA Certificate

To upload the root CA certificate to the Exchange server, perform the following steps:

- 1 Download the SSL certificate from the on-premises ENS server. Access the ENS Alive endpoint in a browser and download the certificate from the address bar.

Note You must only download the root certificate issued by a trusted authority and signed by an internal CA. For cloud deployment, you can download the root certificate from <https://ens.getboxer.com/api/ens/alive>, <https://ens-eu.getboxer.com/api/ens/alive>, <https://ens-apj.getboxer.com/api/ens/alive>, or <https://ens-uk.getboxer.com/api/ens/alive> based on your region, issued by VMware for your account.

For On-Premise deployment, download the root certificate and replace acme.com with the resolved name or IP address of your ENS server.

- 2 Import this certificate on the Exchange Server into the **Trusted Root Certification Authorities** through MMC.

Configuring ENS2 on Cloud and On-Premises Deployment

2

ENS2 can be deployed on a cloud-hosted service or on an on-premises set up. This topic describes configuring ENS2 on cloud and on-premises environment.

This chapter includes the following topics:

- [Email Notification Service for Cloud](#)
- [Email Notification Service for On-Premises](#)
- [ENS2 Application Configuration Keys for Boxer](#)
- [Migrating from ENS On-Premises Server to Cloud Server](#)
- [Configuring SEG as the EWS Proxy for ENS](#)
- [Enable Certificate-Based Authentication for ENS](#)

Email Notification Service for Cloud

Use Workspace ONE UEM console to configure Workspace ONE Boxer for your cloud deployment.

Configure the Email Notification Service 2 (ENS2) related settings for Workspace ONE Boxer on the Workspace ONE UEM console.

Prerequisites

- An API token and ENS2 server URL received from VMware is required to activate the ENS service using the Workspace ONE UEM console. To provision the ENS cloud API token, contact VMware support.
- Ensure the ENS server certificate is available on the user's Exchange server. See [#unique_9](#).

Procedure

- 1 Select the required organization group.
- 2 Select **APPS & BOOKS** and then select the **Public** tab.
- 3 Select **VMware Boxer**.
- 4 Select **Edit** on the upper right corner of the page and then select the **Assignment** tab.
- 5 In the **Application Configuration (Optional)** section, add the required keys. The details of the required keys to be added are listed in the [ENS2 Application Configuration Keys for Boxer](#) topic.

6 Select **Save & Publish** and then select **Publish** on the next page.

ENS Endpoints and IP Whitelist

The API endpoints supported by ENS2 are listed in this topic.

When using cloud ENS servers, you must ensure that the ENS is accessible from the Exchange or Office 365 environment. The inbound IP addresses must be whitelisted to permit the ENS traffic into Exchange or Office 365. The IP address is selected based on the region the ENS is hosted in. The following table describes the Exchange server IP whitelisting requirements.

Table 2-1. Exchange Server IP Whitelisting Requirements

Location	API Endpoint	ENS Outbound to Exchange Inbound
North America	https://ens.getboxer.com/api/ens	35.170.156.92
		52.203.205.147
Asia Pacific	https://ens-apj.getboxer.com/api/ens	54.248.56.175
		54.249.212.171
European Union (EU)	https://ens-eu.getboxer.com/api/ens	18.195.84.245
		18.196.197.192
United Kingdom (UK)	https://ens-uk.getboxer.com/api/ens	3.10.97.61
		18.132.5.114

For information on the architecture design and functionality of ENS2, see [ENS2 Architecture](#).

Note The Exchange outbound connections are required to Cloud ENS and CNS. VMware leverages the public cloud providers for the greatest availability of services and cannot provide a static list of IPs. If there is a requirement to limit the outbound connectivity, the following hostnames can be used. For ENS use ens.getboxer.com, ens-eu.getboxer.com, ens-uk.getboxer.com, and ens-apj.getboxer.com (based on region in which the ENS is used) and for CNS use cns.awmdm.com. The outbound IP addresses must be whitelisted from the Microsoft Exchange client access rules (including Office 365) and any other firewall. This permits the outbound communication from the Exchange server into the ENS server. You need not whitelist SEG IP addresses as all outbound connections from the Exchange server is going to the ENS server and not to the SEG EWS proxy.

Verify VMware Boxer Settings

Use Workspace ONE Boxer to verify your email connectivity.

After you have added the ENS configuration keys to VMware Boxer in Workspace ONE UEM, check the Boxer settings on your device to confirm it has received these keys and that the ENS is activated.

Procedure

- 1 Open Boxer, tap the **Settings** icon and then select the appropriate email account.
- 2 In the email settings, verify the **Use Push Service** is enabled.
- 3 In the email settings, verify the **Notifications** display **Push** as the default selection.

Results

If the **Use Push Service** is enabled and Notifications display **Push**, then the ENS is activated.

Email Notification Service for On-Premises

Configuring ENS for your On-Premises deployment in a 3-step process.

You must first configure CNS and download the ENS configuration files, then install ENS2, and finally configure Boxer for On-Premises.

You must also ensure that the ENS server certificate is available on the user's Exchange server. See [#unique_9](#).

Configure CNS and Download Email Notification Service Configuration Files

Before you install ENS in an on-premise deployment, you must configure the Cloud Notification Service (CNS) and download the configuration .xml file using the Workspace ONE UEM console.

Prerequisites

- Download the CNS public certificate from <https://resources.workspaceone.com/view/2hjxzvgkxyf8n738hy7x/en>.
- Navigate to the **System > Advanced > Secure Channel Certificate** and select **Download CNS Secure Channel Certificate Installer** if the UEM console is on-premise. Open a support ticket with the VMware Support and provide the secure channel certificate file through the support ticket.
- Assign the **db_owner** role and public role to the SQL server user that is used for running the application. ENS supports any version of the SQL server. The database option must be selected for the external database and you must set the collation to **SQL_Latin1_General_CP1_CI_AS**. For more information on creating the Workspace ONE UEM database, see the *Create the Workspace ONE UEM Database* topic in the *Installing Workspace ONE UEM* guide.
- Set up the **SQL Server AlwaysOn** for active/active or active/passive setup for the high availability configuration. If you are using AlwaysOn, point to the availability group when selecting the database server during the ENS2 installation. See the *Overview of Always On Availability Groups (SQL Server)* topic for more information.

Note To proceed with the ENS2, your console version must be 9.3 or later. If the **Download Installer** is displayed when you are configuring and downloading the configuration files, then your console version is less than 9.3 and this installer is for the earlier version of ENS. See the *VMware Email Notification Service Installation* guide for instructions and detailed information.

Procedure

- 1 Select the required Organization Group and navigate to **Groups & Settings > All Settings**.
- 2 From the System column, select **Advanced**, and then select **Site URLs**.

- 3 (On-premise UEM console only) From the site URLs values page, select **Cloud Notification Service URL** and add `https://cns.awmdm.com/nws/notify/apns`.
- 4 (On-premise UEM console only) - If the Workspace ONE UEM console is deployed on-premise, then you must upload the CNS certificate.
 - a From the left navigation, select **System > Security > SSL Pinning**.
 - b Select **ADD HOST**. In the **Add Pinned Host**, enter the host as `cns.awmdm.com`.
 - c Select **Upload** and upload the CNS certificate you downloaded earlier.
- 5 From the Settings page, select **Email** and then select **Email Notification**.
- 6 To enable Email Notification, select **Yes** and then click **Save**.
After the settings are saved, the Download Configuration option is displayed.
- 7 Select **Download Configuration**.
- 8 Enter a password in **Certificate Password**. to download the configuration.
The password is required to download the configuration and must be provided again during the ENS installation.
- 9 Select **Confirm Password**, reenter the password, and click **Download**.
- 10 Save the archived .xml file to be accessible for the upload during the ENS installation.

Install and Upgrade Email Notification Service 2

To use the Email Notification Service 2 (ENS2), you must install the ENS on an IIS server.

Prerequisites

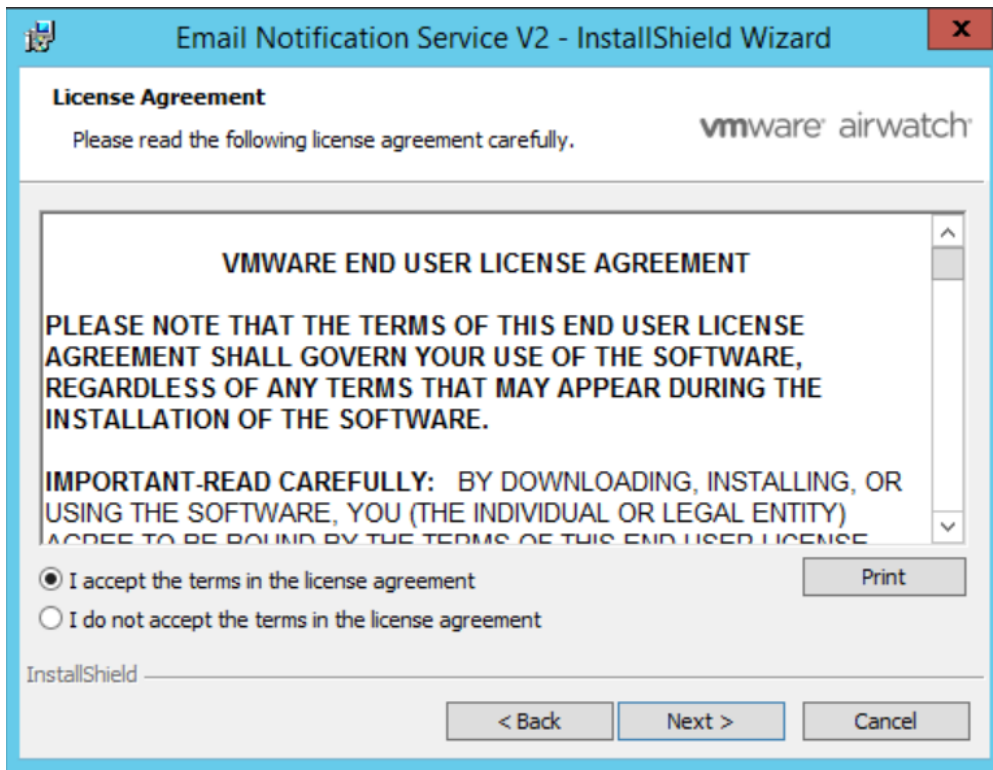
Complete the following tasks before you install ENS2:

- Install IIS 7 or later on the Web Server
- Update ASP.Net to v 4.6.2
- Download the `config.xml` file from the Workspace ONE UEM console. See [Configure CNS and Download Email Notification Service Configuration Files](#).
- Ensure that an SSL certificate with a valid hostname is set up on the IIS server. This server should be externally accessible via https (SSL cert) and with a Fully Qualified Domain Name (FQDN).
- Create a new database and name it appropriately. If you are using SQL Server AlwaysOn, you can create availability group and listeners.
- The database account user must have privileges to access and modify the database.

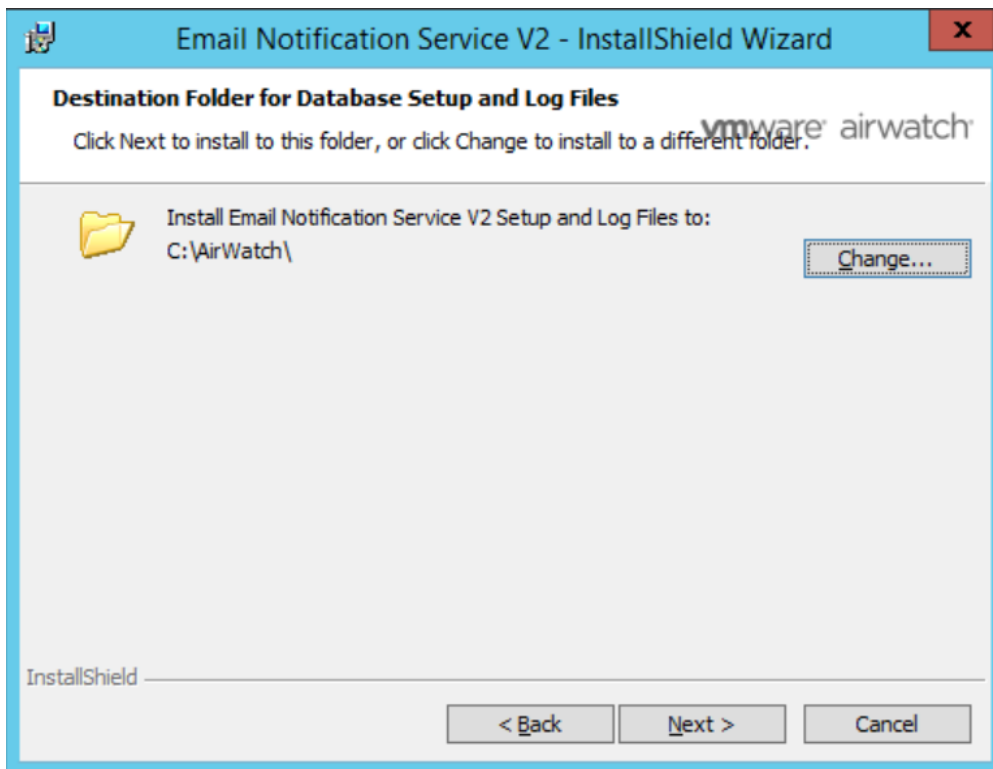
Procedure

- 1 Download the latest version of ENS2 installer from the Software section of the [My Workspace ONE portal](#).
- 2 Run the installer. The InstallShield Wizard opens and displays the License Agreement.

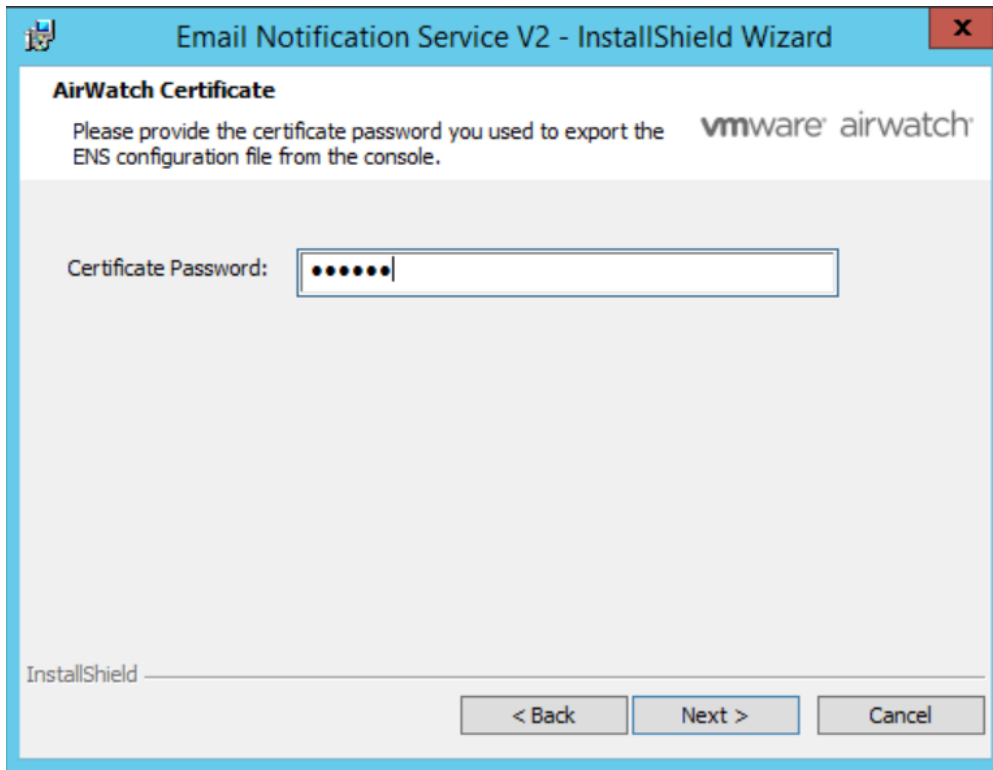
- 3 Select the **I accept the terms in the license agreement** check box and then click **Next**.



- 4 Click **Next** to install the components at the default location. If you want to install the components at a custom location, click **Change** and browse and select your location.

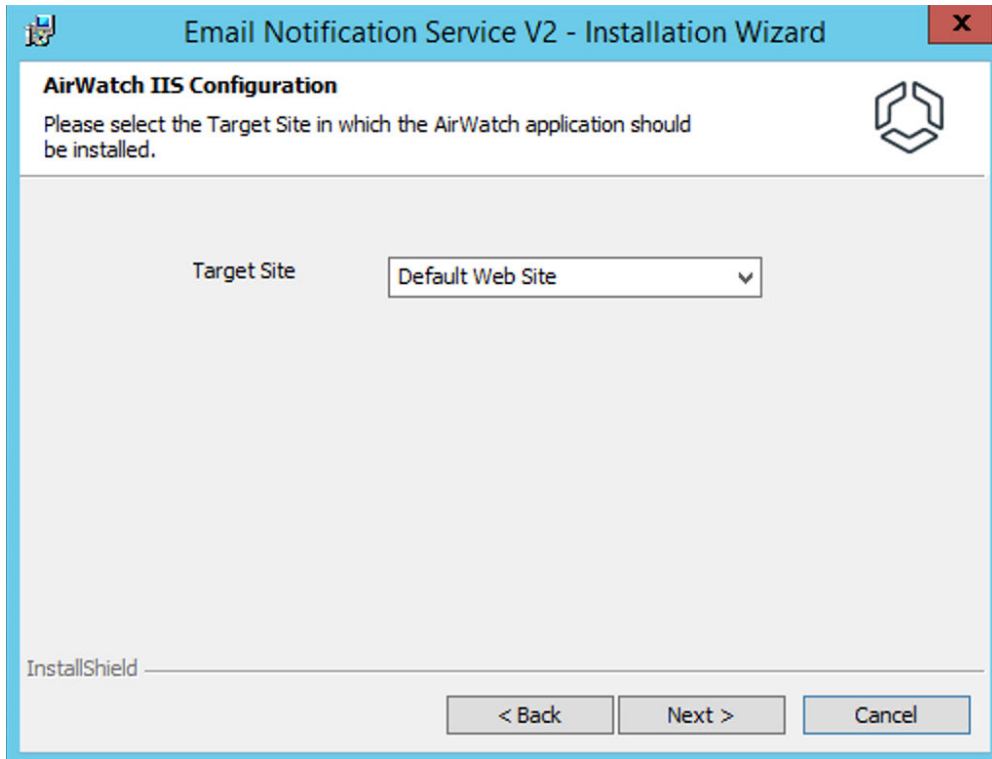


- 5 Click **Browse** and locate the **config.xml** file and then click **Next**.
- 6 Click **Certificate Password** text box and enter the certificate password you provided when you downloaded the configuration file from the Workspace ONE UEM console, and then click **Next**.



- 7 (Optional) On the **AirWatch CNS Email Proxy Configuration** window, provide the following information:
 - a Check **Enable CNS Proxy** to configure the CNS proxy. Enter the hostname/IP address and the proxy port of the the server.
 - b Select the authentication type:
 - Anonymous - user name and password is not required
 - Basic/Windows - Enter user name and password.

- 8 Select the target site on the Airwatch IIS configuration window.



Email Notification Service V2 - Installation Wizard

AirWatch IIS Configuration

Please select the Target Site in which the AirWatch application should be installed.

Target Site: Default Web Site

InstallShield

< Back Next > Cancel

9 On the Database Server window, enter the following information:

- a Browse to select the database server where the database is located. Enter the IP address or host name of the server if the server is not listed.
- b Select Windows authentication or server authentication based on your authentication configuration. If you choose server authentication, enter the login ID and password.
- c Enter the name of the database in the **Name of the database catalog** text box and click **Next**.
 - If the database has already been created, browse and select the existing database.
 - If there is no existing database, enter a name for the new database, and the installer will create and publish the database.
 - You can configure using a single database configuration or with SQL AlwaysOn. The below figure shows the the single database configuration.

The below diagram shows the configuration using SQL Server AlwaysOn.

Note If you are using SQL Server AlwaysOn, you can configure the availability group Listener URL here.

The screenshot shows the 'Database Server' step of the 'Email Notification Service V2 - InstallShield Wizard'. The window has a blue title bar with the text 'Email Notification Service V2 - InstallShield Wizard' and a red close button. The main content area is white with a blue header bar containing the title 'Database Server' and a VMware logo. Below the header, the text 'Select database server and authentication method to install/upgrade the database' is displayed. An 'IMPORTANT NOTE: Please backup the Database that you are targeting.' is shown in bold. The 'Database server that you are installing to:' section has a dropdown menu showing 'ENS-Static.bsb02.org' and a 'Browse...' button. The 'Run installer using:' section has two radio buttons: 'Windows authentication credentials of current user' (unselected) and 'Server authentication using the Login ID and password below' (selected). Below the selected option, there are text boxes for 'Login ID:' (containing 'sqladmin') and 'Password:' (containing masked characters). The 'Name of database catalog:' section has a text box containing 'ENS_Clustered_Static_IP' and a 'Browse...' button. At the bottom, there is an 'InstallShield' label and three buttons: '< Back', 'Next >', and 'Cancel'.

- 10 Click **OK** to confirm and then click **Install** to start the installation.

The screenshot shows the 'Ready to Install the Program' step of the 'Email Notification Service V2 - InstallShield Wizard'. The window has a blue title bar with the text 'Email Notification Service V2 - InstallShield Wizard' and a red close button. The main content area is white with a blue header bar containing the title 'Ready to Install the Program' and the VMware Airwatch logo. Below the header, the text 'The wizard is ready to begin installation.' is displayed. A section with the text 'Click Install to begin the installation.' and 'If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.' is shown. At the bottom, there is an 'InstallShield' label and three buttons: '< Back', 'Install', and 'Cancel'.

- 11 Click **Finish** to complete the installation.

After the installation is complete, an API token is displayed in a text file.

12 Copy the API token.

Note This API token is required when configuring the Boxer application UEM console. Use this value for the *ENSAPIToken* field.

Upgrade ENS2

You can upgrade from an older version of ENS2 to the latest version.

You must have the latest version of the installer on your system. Download the latest version of ENS2 installer from the Software section of the [My Workspace ONE portal](#).

The instructions to upgrade to the latest version of ENS2 are the same as the ENS2 installation instructions. See [Install and Upgrade Email Notification Service 2](#).

Configure Workspace ONE Boxer for On-Premises

After you have installed the ENS2, you must configure the ENS2 related settings for Workspace ONE Boxer on the Workspace ONE UEM console.

Prerequisites

The API token and ENS2 server URL are required to activate the ENS service using Workspace ONE UEM console.

Procedure

- 1 Select the required organization group.
- 2 Select **APPS & BOOKS** and then select the **Public** tab.
- 3 Select **VMware Boxer**.
- 4 Select **Edit** on the upper right corner of the page and then select the **Assignment** tab.
- 5 In the **Application Configuration (Optional)** section, add the required keys. The details of the required keys to be added are listed in the [ENS2 Application Configuration Keys for Boxer](#) topic.
- 6 Select **Save & Publish** and then select **Publish** on the next page. To verify the settings, see [Verify VMware Boxer Settings](#).

ENS2 Application Configuration Keys for Boxer

Configure the ENS2 with the application configuration values. You can configure settings for the ENS2 using the configuration key and configuration value provided by the Workspace ONE UEM.

The following table lists the application configuration keys and the configuration values for ENS2.

Configuration Key	Value Type	Configuration Value	Description
ENSLinkAddress	String	<p>Supported format: https://ens.getboxer.com/api/ens Replace <code>ens.getboxer.com</code> with the resolved name or IP provided by VMware based on your region.</p> <p>Sample link address:</p> <ul style="list-style-type: none"> ■ For AMER - https://ens.getboxer.com/api/ens ■ For APAC - https://ens-apj.getboxer.com/api/ens ■ For EMEA - https://ens-eu.getboxer.com/api/ens ■ For UK - https://ens-uk.getboxer.com/api/ens 	<p>The URL address of the ENS server. Provide the address for the ENS2 system for your users to connect.</p> <p>For Cloud customers, the address must be https://ens.getboxer.com/api/ens (or any of the ENS Cloud URLs or API endpoints).</p> <p>For on-premise customers, the address must be in the following format: https://mycompany.com/MailNotificationService/api/ens. Here, mycompany.com is the IP or domain name of your ENS server.</p>
ENSAPIToken	String	Sample API token - da848cc9340034843ecdjdad11048461q	VMware provides the API token to activate the ENS service. For the on-premise installation, the on-premise installer creates this token.
AccountNotifyPush	Boolean	False - disable (default) True - enable	Enables ENS for the account.
EWSUrl	String	<p>Supported Format: https://[external_email_server_domain]/EWS/Exchange.asmx Sample EWS URL:</p> <ul style="list-style-type: none"> ■ https://e.mail.com/EWS/Exchange.asmx ■ https://seg.dom.com/EWS/Exchange.asmx 	Enables manual configuration of Exchange Web Services (EWS) endpoint when the autodiscovery is disabled in your Exchange environment.
PolicyLimitNotificationText	Integer	<p>0 - sets notification to sender, subject, and preview.</p> <p>1 - sets notification to sender and subject (default).</p> <p>2 - sets notification to sender.</p> <p>3 - sets notification to a generic message (new message).</p> <p>4 - sets notification to none (only the badge is updated).</p>	To configure the ENS notification policy for Workspace ONE Boxer, add the following key value pair. When configured, Workspace ONE Boxer immediately resubscribes to the ENSv2 and notification policy is updated as per the set key value.

Migrating from ENS On-Premises Server to Cloud Server

This topic describes the information required to migrate from the ENS on-premises server to the cloud server.

Before you begin, ensure that the cloud ENS can access the Exchange server. See [Email Notification Service for Cloud](#) for more details. When you migrate from the on-premise server to the cloud server, you must update the following Boxer profile configuration:

- Update the **ENSLinkAddress** to the appropriate cloud URL.
- Update the **ENSAPIToken** to the one provided for cloud.

When all the users migrate to the cloud server, ENS on-premise servers can be shut down. During migration, the users can unregister from the on-premise ENS server and migrate to the cloud ENS server.

Configuring SEG as the EWS Proxy for ENS

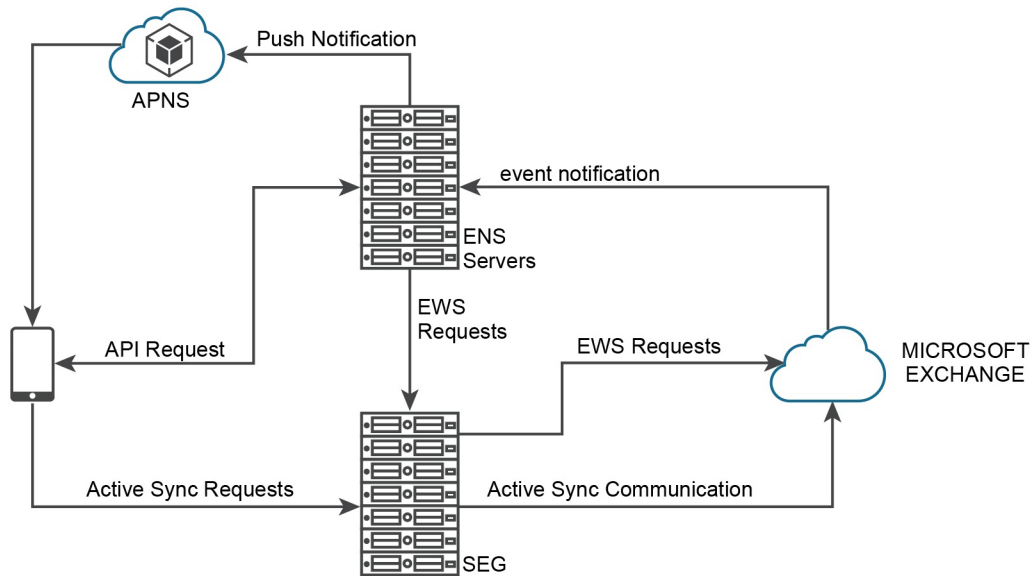
Monitor compliance of the client with the ENS2 environment so that ENS2 together with SEG V2 can block or unblock a client depending on the compliance criteria of the client.

Background

Currently, when a mobile device is enterprise wiped or removed from the Workspace ONE UEM console, the client unregisters from the ENS2 environment. For example, when an enterprise wipe command is sent to iOS Boxer the device tries to unregister until it is successful. However, this is not an ideal scenario as there is a dependency on the device to unregister from the ENS2 environment.

Integration with SEG V2

The SEG V2 protects the email configuration of the client and enables MEM functionality by monitoring the compliance of the device against the configuration in the Workspace ONE UEM console. With the integration of ENS2 and SEG V2, you can block request to a device and control the client, based on the compliance criteria specified in the Workspace ONE UEM console. The following is a high-level diagram showing the interaction between ENS2 and Exchange with SEG V2 as the proxy.



In addition to the compliance scenario, you can use SEG V2 as a proxy when the Exchange Web Service (EWS) endpoint is not publicly available. The EWS proxy allows devices to subscribe to the EWS subscriptions through the SEG V2 server instead of publicly exposing the EWS endpoint.

SEG V2 supports both cloud and on-premises ENS deployments. SEG V2 listens to the EWS traffic from ENS using the EWS endpoints. SEG applies the MEM compliance policies on the incoming requests and proxies the requests to Exchange. See, [Configure ENS2 with SEG](#).

Supported Exchange Web Service Authentication Methods for SEG Proxy

The Exchange Active Sync (EAS) authentication method used with Boxer must match the EWS authentication method as ENS implicitly uses the authentication method used by Boxer. SEG as EWS proxy supports basic authentication, certificate-based authentication (CBA) with KCD, and modern authentication (OAuth) types and does not support the New Technology LAN Manager (NTLM) authentication type.

Certificate-based authentication using KCD is supported. If your deployment utilizes CBA using KCD, SEG acquires the Kerberos token (from KCD) required for the Exchange authentication. The authentication method for EAS and Exchange Web Service (EWS) protocol must match for SEG to work correctly.

For more information, see the *Configure SEG V2 Compliance for Email Notification Service* topic in the *Secure Email Gateway (SEG) V2* guide.

Supported Servers for Exchange Web Service and ActiveSync

If you have different fully qualified domain name (FQDN) for Exchange Web Service (EWS) and ActiveSync endpoints, it is recommended you upgrade to SEG version 2.12 or later. In this SEG version, you can provide a different hostname and uncomment the `ews.email.server.host.and.port=https://example.com:443` property for EWS flows.

Note If you provide a different hostname, SEG still uses the `server.timeout`, `ignoreSslErrorsWithExch`, and other settings from the EAS email server configuration provided in the MEM configuration for the email server client. If the EWS server is using self-signed certificate then you need to add the self-signed certificate in the Java trustStore before the SEG installation or you need to rerun the SEG installer.

For SEG versions before 2.12, the only option available is to have two different MEM configuration and two different SEG servers to proxy traffic. One SEG can serve one email server address or FQDN. However, if EWS and ActiveSync endpoints are hosted on the same email server address or FQDN, same SEG server can proxy both EWS and ActiveSync traffic.

Configure ENS2 with SEG

The following procedure describes the steps to configure ENS2 with SEG.

Procedure

- 1 Navigate to **SEG > Configuration**.
- 2 Select the `application.properties` file and edit the file.
- 3 Select the `enable.boxer.ens.ews.proxy` value and update the value to `enable.boxer.ens.ews.proxy=true`.
- 4 Restart the SEG service. SEG receives the `/EWS` and `/ews` endpoints for traffic from the ENS.

Configure SEG for Authentication

If you are using basic authentication only, and the EWS endpoint is configured to allow NTLM authentication, ensure the SEG version is 2.9.0.1 and validate the `remove.unsupported.auth` configuration in SEG using the following procedure:

Procedure

- 1 Navigate to **SEG > Configuration** folder using file explorer.
- 2 Select the `application.properties` file and edit the file.
- 3 Check if the `remove.unsupported.auth.for.ews` value is true if NTLM authentication is enabled on Exchange, as SEG does not support NTLM connection persistence. If you do not see an entry for `remove.unsupported.auth.for.ews` then the SEG version is not 2.9.0.1. Ensure the SEG version is 2.9.0.1.
- 4 Verify the SEG version and save the file.

Results

In the SEG application.properties, flag the remove.unsupported.auth.for.ews=true value to remove the unsupported www-authentication header from the EWS response to the ENS through SEG. The NTLM and the Negotiate headers are removed from the EWS response. The NTLM header as a persistent connection is not supported by SEG. The Negotiate www-authenticate header is removed in the absence of a valid client certificate, that is, when the userPrincipalname (UPN) is null. In the absence of Kerberos authentication, the Negotiate header can be considered as NTLM authentication.

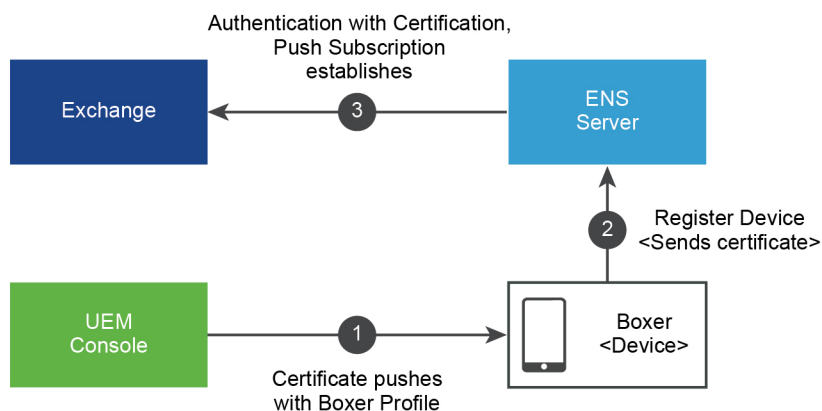
Note If you enable both basic and Kerberos authentication and the client fails to present a valid client certificate, then the SEG removes the Negotiate header and requests you to authenticate using basic authentication. In such scenarios, the client is enforced to use basic authentication only. If the client does not have the basic authentication configured then the client fails to receive a successful response. When the client presents a valid certificate, the SEG generates a Kerberos token and proceeds with the Negotiate authentication.

Enable Certificate-Based Authentication for ENS

ENS supports certificate-based authentication (CBA) and dual authentication. The dual authentication is a combination of basic authentication and certificate-based authentication. For ENS, you must configure the Boxer application with certificate-based authentication for Exchange server and enable certificate-based authentication for the EWS endpoint. ENS uses the same certificate that the Boxer application receives for the authentication purpose. ENS must ensure that the EWS endpoint can validate the certificates used by the Boxer application.

Prerequisites

Configure Boxer application with CBA and enable CBA for the EWS endpoint. For more information about configuring CBA for Workspace ONE Boxer, see the *Workspace ONE Boxer Admin Guide* documentation.



Procedure

- 1 Push the certificate with Boxer profile from the Workspace ONE UEM console to the Workspace ONE Boxer.

- 2 Register your device with the ENS server and send the certificate from Workspace ONE Boxer.
- 3 Send certificate from ENS to the Exchange server and establish the push subscription.

Configure ENS2 for Certificate-Based Authentication

When you configure ENS2 for Workspace ONE Boxer and want to use Certificate-Based Authentication (CBA) for authentication, you must follow the steps listed in this section for ENS2 to work with CBA.

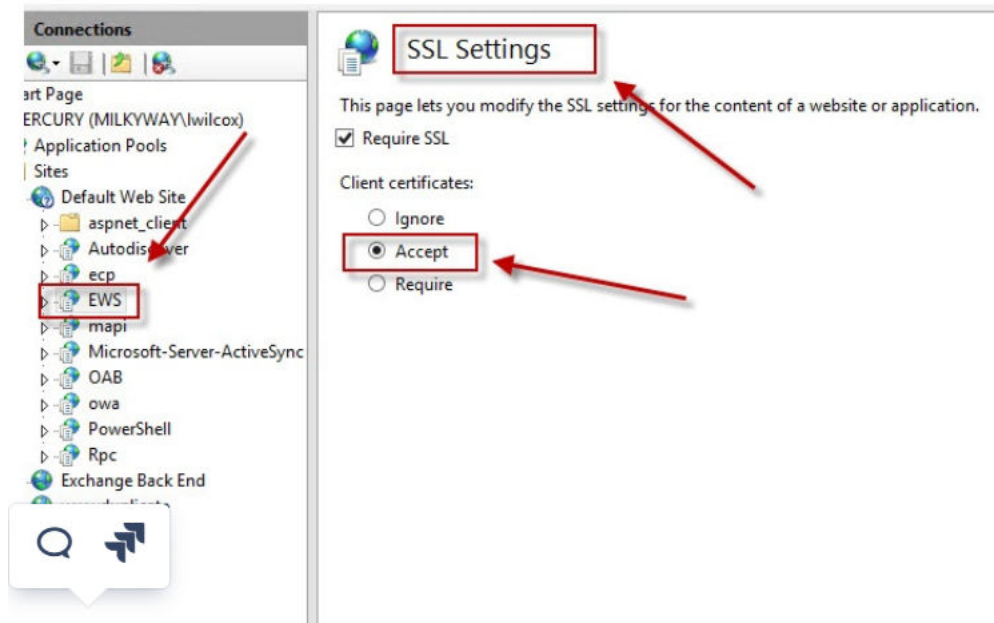
- 1 Configure Workspace ONE Boxer to use CBA. See [Configure Certificate-Based Authentication on the Exchange Server](#).
- 2 Change the appropriate settings to ensure that CBA is supported for the EWS endpoint and for EAS on the on-premise Exchange Server. See [Using Office 365 with ENS2 and Certificate-Based Authentication](#) and [Configure Certificate-Based Authentication on the Exchange Server](#).
- 3 If you are using Secure Email Gateway V2 (SEG V2), see the *Secure Email Gateway V2 guide* for information on the changes that are required on the SEG server.

Configure Certificate-Based Authentication on the Exchange Server

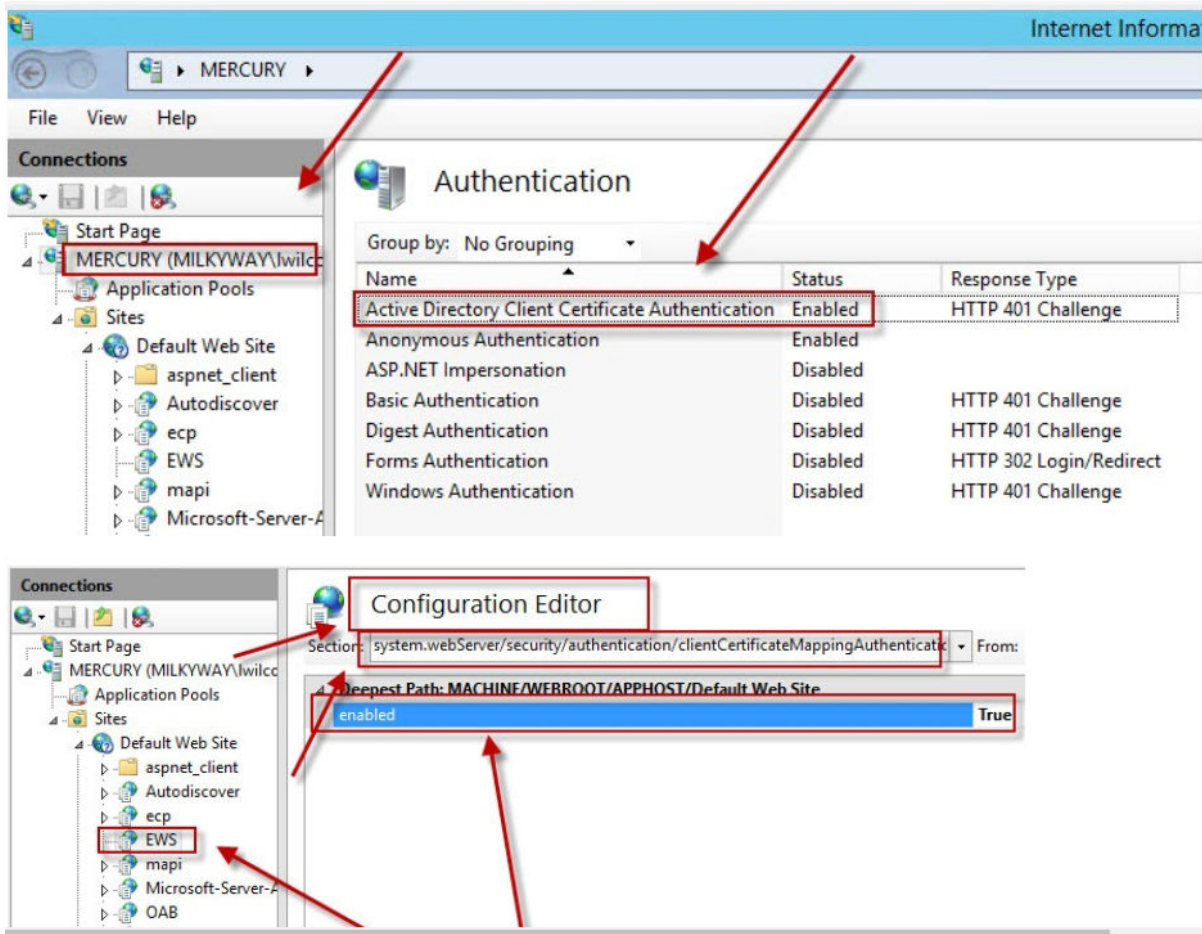
You can enable certificate-based authentication (CBA) for Exchange Active Sync (EAS) on the Exchange Server (for TLS testing) by modifying specific values on the IIS server. Office 365 or Exchange online does not directly support certificate-based authentication. You must set up dual authentication, that is, modern authentication and CBA, to setup certificate-based authentication for Office 365. You must have Active Directory Federation Service (ADFS) setup to do certificate-based authentication. Office 365 authenticates through the modern authentication, and certificate is presented to the ADFS for authentication. On the Boxer profile, modern authentication and certificate-based authentication needs to be enabled that is, AccountUseOAuth must be enabled. See the *Workspace ONE Boxer Admin Guide* documentation for more details.

Procedure

- 1 From the IIS console, navigate to the EWS endpoint and ensure the EWS endpoint accepts the client certificates.



- 2 For client certificates to be allowed on the Exchange server, the Exchange server must have **Active Directory Client Certificate Authentication** installed and enabled in IIS.



Using Office 365 with ENS2 and Certificate-Based Authentication

If you are using Office 365 and want to perform certificate-based authentication (CBA), you must enable certain settings in the Workspace ONE Boxer profile.

Office 365 or Exchange online does not directly support certificate-based authentication. You must set up dual authentication, that is, modern authentication and CBA, to set up certificate-based authentication for Office 365. You must have Active Directory Federation Service (ADFS) set up to perform certificate-based authentication. Office 365 authenticates through the modern authentication and certificate is presented to ADFS for authentication.

You must also enable modern authentication and certificate-based authentication using the *AccountUseOAuth* setting in the Workspace ONE Boxer profile. See the *Workspace ONE Boxer Admin Guide* documentation for more details.

Supported EWS Authentication Methods with Office 365

The following EWS authentication methods are supported with Office 365:

- OAuth 2.0 (Exchange Online only)

- NTLM (Exchange On-premises only)
- Basic (no longer recommended)

Refer to the relevant Microsoft Office 365 documentation for more details.

Troubleshooting ENS

3

This topic lists the various troubleshooting procedures for ENS.

This chapter includes the following topics:

- [ENS2 Resubscription and Badge Count Accuracy Limitations](#)
- [Troubleshooting Accessibility Issues to the ENS Server](#)
- [Troubleshooting the Console Configuration Issues](#)
- [Troubleshooting ENS2 Notification Issues](#)
- [Troubleshooting Connection Issues to the ENS Database](#)
- [Troubleshooting SSL Errors](#)
- [ENS2 Response Code and Error Code Details](#)

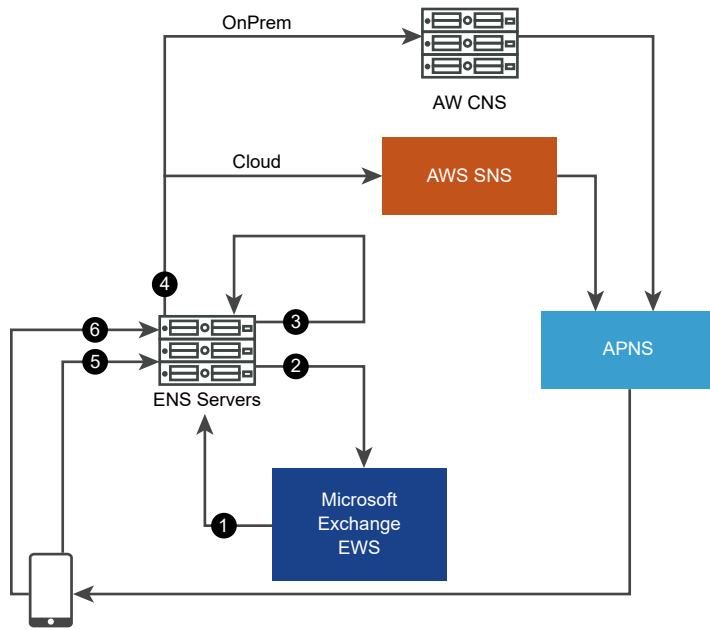
ENS2 Resubscription and Badge Count Accuracy Limitations

The ENS2 uses the Exchange Web Services (EWS) subscription to notify the Boxer application of any changes in an end-users mailbox, including the email notifications. The Boxer application initiates these subscriptions with the ENS and the ENS subscribes a user's account with the EWS.

The EWS is responsible for informing the ENS when there is a change in a user's mailbox. The subscriptions have limited lifetime due to the movement of mailbox, throttling, and so on. The Exchange can drop the EWS push subscriptions which are triggered by the Exchange and the ENS does not have control over the subscription lifetime. The EWS sends notification updates to the Boxer until the EWS subscription is active and alive.

To keep these subscriptions alive, the Boxer application has a check-in mechanism which validates if an EWS subscription is alive. In addition, the ENS2 is listening for status updates from the EWS. If the ENS2 does not receive a status update from the EWS, the ENS2 can send the Boxer a silent push notification to check in with the EWS.

The following figure describes the ENS resubscription process flow.



- 1 The EWS sends a heartbeat signal to the ENS every 15 minutes.
- 2 The ENS sends an acknowledgement to the EWS that the heartbeat signal is received.
- 3 The ENS checks that the heartbeat signal is received every 30 minutes from the EWS.
- 4 If the ENS does not receive a heartbeat signal, the ENS2 sends a silent notification to the Boxer application to initiate the resubscription process.
- 5 The Boxer application initiates a resubscription process on receiving a silent notification.
- 6 The Boxer application proactively checks the EWS subscription status with the ENS server to ensure the continuous delivery of notifications.

The ENS2 requests the Exchange to send heartbeat that a subscription is alive. When the ENS2 detects that heartbeat is not sent, which indicates a drop in subscription. When the ENS2 detects a drop in subscription from the EWS, the ENS2 sends the Boxer a silent notification to initiate the resubscription process. On top of this, when the Boxer application can run in background, the application proactively checks the EWS subscription status with the ENS server to ensure the continuous delivery of notifications.

Therefore, the check-in mechanism used by ENS2 requires intervention from Boxer to renew the EWS subscriptions because the users credentials are required to open the subscription. These credentials are not stored in ENS. The functionality of ENS2 also depends on the Apple Push Notification Service (APNS) to deliver silent notifications to the device.

The following list describes the dependencies of the ENS2 on the EWS and APNS.

- If the Boxer application is active and receives a silent notification, the Boxer application attempts to resubscribe. When the Boxer application receives a silent notification, the Boxer sends a resubscription request to the EWS using the employee credentials.

- The iOS can stop the Boxer process without any warning due to various reasons. In such scenarios, the end users might see Boxer in the **App Scroll** of an iOS device, however, the Boxer process is stopped. The Boxer application has no control over this process and this state is called a **killed** state. If the Boxer application is in a **killed** state when it receives a silent notification, the Boxer application cannot resubscribe due to which the user can experience loss of notifications until the user opens the Boxer application. Opening the Boxer application triggers the ENS subscription again, and the user starts receiving notifications.
- The end user might experience an inaccurate badge count when the time subscription is lost and before the Boxer application resubscribes.

The following list describes the badge count accuracy limitations on the Boxer application:

- **Sync window** - The ENS checks the Inbox folder without the sync period and the Boxer unread messages are within the sync period. So, the users might have unread messages outside the sync window in the Inbox folder. The ENS reports these messages as unread while the user might not see these unread email messages in the Inbox.
- **Boxer application dependency on resubscription** - When the ENS is going through resubscription, the ENS does not receive any notification or badge count. During this period, the ENS does not have the updated badge count.
- **Unmanaged accounts** - When the user has both managed and unmanaged accounts like the Exchange account and Gmail account, the badge counts are not handled correctly.
- **Comparison with Outlook on MAC devices** - The Outlook on MAC devices shows certain emails as read whereas the same emails show unread when opened using Boxer or Outlook for Web Access (OWA). So, the badge count is incorrect when compared with Outlook on MAC devices.

Troubleshooting Accessibility Issues to the ENS Server

This topic describes the steps to troubleshoot the accessibility issues to the ENS server on a cloud ENS installation and on an on-premises ENS installation.

Troubleshooting Accessibility Issues to the ENS Server from a Cloud ENS

Problem: Check if the cloud ENS is accessible from the ENS server and confirm if the ENS server is accessible from the CAS or the Mailbox server.

- 1 Access the following URL in a browser on all CAS or mailbox servers:

<https://{ENS cloud URL}/api/ens/alive>.

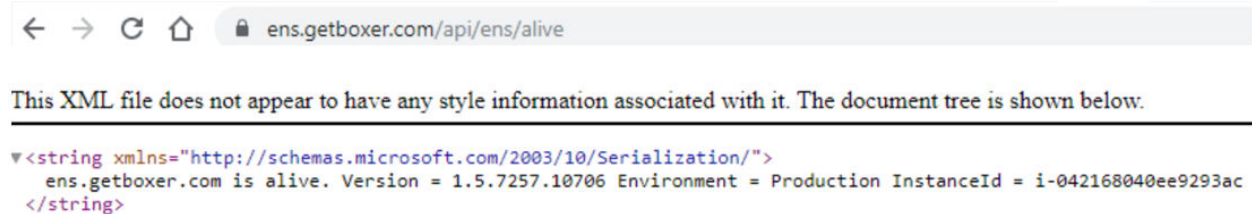
- 2 Select the ENS cloud URL based on your region.

Region	ENS Cloud URL
North America	https://ens.getboxer.com/api/ens
Asia Pacific	https://ens-apj.getboxer.com/api/ens

Region	ENS Cloud URL
European Union (EU)	https://ens-eu.getboxer.com/api/ens
United Kingdom	https://ens-uk.getboxer.com/api/ens

Results:

You must receive the following response:



If you are unable to see a similar response, then whitelist the IP addresses and endpoints and validate the connection to the ENS server. To see the supported ENS2 API endpoints, refer the [ENS Endpoints and IP Whitelist](#).

Troubleshooting Accessibility Issues to the ENS Server from an On-Premises Installation

Problem: Check if the ENS server is accessible on an on-premises setup and is receiving the request. After an on-premises ENS installation, confirm that the ENS is installed and running on the ENS server.

1. Navigate to the following URL in a web browser and check the same server where ENS is installed. The user localhost is mentioned as follows <http://localhost/MailNotificationService/api/ens/alive> and <https://localhost/MailNotificationService/api/ens/alive>. To check from outside the ENS server, see <http://{ENS server public url}/MailNotificationService/api/ens/alive> and <https://{ENS server public url}/MailNotificationService/api/ens/alive>. You must be able to see the following response:



2. Confirm that a certificate is imported and 443 is bound to the website if you have an issue with the https 443 traffic.

Result:

Confirm if the ENS is receiving the request from outside (for example, receiving the request from a browser when you reach the alive endpoint). When verifying the ENS alive endpoint, the IIS logs are generated. The IIS logs are by default stored at the following path: %SystemDrive%\inetpub\logs\LogFiles. If you do not find the logs at the default path, then the logs for your IIS might be stored at a different location. To get the path for the IIS logs, check the following link: [Managing IIS Log File Storage](#).

For other successful ENS traffic, you might see the following log entries in the IIS logs.

```
POST /MailNotificationService/api/ens/getpublickey - 443 - 192.168.2.37 VMwareBoxer/4.12.0+(iPhone;+iOS+11.2.5;+Scale/3.00) - 200 0 0 15
POST /MailNotificationService/api/ens/registerdevicev2 - 443 - 192.168.2.37 VMwareBoxer/4.12.0+(iPhone;+iOS+11.2.5;+Scale/3.00) - 200 0 0 31
POST /mailnotificationService/api/ens/pushnotificationlistener id=1&f=HaUHMJsJulCdvpdOagqcZ2CM0fsm4Q2FcGcdXSD2E17GH3faHfNba6bH1810eKbIyq9yxqohITAgcCAz2MHRG1UUFu2f1V2
POST /MailNotificationService/api/ens/getsubscriptionstatus - 443 - 192.168.2.37 VMwareBoxer/4.12.0+(iPhone;+iOS+11.2.5;+Scale/3.00) - 200 0 0 46

GET /MailNotificationService/api/ens/alive - 443 - 192.168.2.37 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/66.0.3359.139+Safari/537.36 - 200 0 0 0
GET /MailNotificationService/api/ens/alive - 80 - 192.168.2.37 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/66.0.3359.139+Safari/537.36 - 200 0 0 0
```


Test the Exchange Web Services URL

The Exchange Web Services (EWS) subscriptions notify changes in a users' mailbox. Use the [Microsoft's Remote Connectivity Analyzer](#) online tool to test the EWS URL. You can test the EWS URL only if the EWS is configured for the basic authentication and the EWS is publicly available.

- 1 Open the [Microsoft's Remote Connectivity Analyzer](#).
- 2 Select the **Synchronization, Notification, Availability, Automatic Replies** under the **Microsoft Exchange Web Services Connectivity Tests** and click **Next**.
- 3 Enter the **Email address, Domain\User Name (or UPN), Password, and Confirm Password** information.
- 4 Enter the EWS URL manually, if the autodiscovery is not enabled or select the **Use Auto-Discovery to detect server settings** if autodiscovery is enabled.
- 5 Click **Verify** account and perform the test.


Results:

If there are no issues, the following success message appears:


Connectivity Test Successful



Test Details

[Start Over](#)
[Run Test Again](#)


 Exchange Web Services synchronization, notification, availability, and Automatic Replies.

Tests of all Exchange Web Services tasks completed successfully.

Additional Details
 Test Steps

[Expand All](#)



If the connectivity test fails for the following reasons, then expand the error to see more information.

You see the following 401 error when the user is unauthorized.

Connectivity Test Failed

Test Details Start Over Run Test Again

Exchange Web Services synchronization, notification, availability, and Automatic Replies.
Not all tests of Exchange Web Services tasks completed.

Additional Details
Elapsed Time: 691 ms.

Test Steps

- Creating a temporary folder to perform synchronization tests.
Failed to create temporary folder for performing tests.
- Additional Details
 - Exception details:
Message: The request failed. The remote server returned an error: (401) Unauthorized.
 - Type: Microsoft.Exchange.WebServices.Data.ServiceRequestException
 - Stack trace:
at Microsoft.Exchange.WebServices.Data.ServiceRequestBase.GetEwsHttpResponse([EwsHttpRequest request])
at Microsoft.Exchange.WebServices.Data.ServiceRequestBase.ValidateAndEmitRequest([EwsHttpRequest& request])
at Microsoft.Exchange.WebServices.Data.MultiResponseServiceRequest`1.Execute()
at Microsoft.Exchange.WebServices.Data.ExchangeService.BindToFolder(FolderId folderId, PropertySet propertySet)
at Microsoft.Exchange.WebServices.Data.ExchangeService.BindToFolder([FolderId folderId, PropertySet propertySet])
at Microsoft.Exchange.Tools.ExRca.Tests.GetOrCreateSyncFolderTest.PerformTestReally()
Exception details:
Message: The remote server returned an error: (401) Unauthorized.
Type: System.Net.WebException
Stack trace:
at System.Net.HttpWebRequest.EndGetResponse([AsyncResult asyncResult])
at System.Threading.Tasks.TaskFactory`1.FromAsyncCoreLogic([AsyncResult iar, Func`2 endFunction, Action`1 endAction, Task`1 promise, Boolean requiresSynchronization])
--- End of stack trace from previous location where exception was thrown ---
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
at Microsoft.Exchange.WebServices.Data.EwsHttpRequest.<ExecuteRequestAsync>d__64.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
at Microsoft.Exchange.WebServices.Data.EwsHttpRequest.Microsoft.Exchange.WebServices.Data.IEwsHttpRequest.GetResponse()
at Microsoft.Exchange.WebServices.Data.ServiceRequestBase.GetEwsHttpResponse([EwsHttpRequest request])
Elapsed Time: 691 ms.

You see the following error when the autodiscovery is not enabled.

Connectivity Test Failed

Test Details Start Over Run Test Again

Exchange Web Services synchronization, notification, availability, and Automatic Replies.
Not all tests of Exchange Web Services tasks completed.

Additional Details

Test Steps

- The Microsoft Connectivity Analyzer is attempting to test Autodiscover for mem1@mem13.ssddevrd.com.
Testing Autodiscover failed.
- Additional Details
Elapsed Time: 503 ms.
- Test Steps
 - Attempting each method of contacting the Autodiscover service.
The Autodiscover service couldn't be contacted successfully by any method.
 - Additional Details
Elapsed Time: 503 ms.
 - Test Steps
 - Attempting to test potential Autodiscover URL https://mem13.ssddevrd.com:443/Autodiscover/Autodiscover.xml
Testing of this potential Autodiscover URL failed.
 - Additional Details
 - Test Steps
 - Attempting to test potential Autodiscover URL https://autodiscover.mem13.ssddevrd.com:443/Autodiscover/Autodiscover.xml
Testing of this potential Autodiscover URL failed.
 - Additional Details
 - Test Steps
 - Attempting to contact the Autodiscover service using the HTTP redirect method.
The attempt to contact Autodiscover using the HTTP Redirect method failed.
 - Additional Details
 - Test Steps
 - Attempting to contact the Autodiscover service using the DNS SRV redirect method.
The Microsoft Connectivity Analyzer failed to contact the Autodiscover service using the DNS SRV redirect method.
 - Additional Details
 - Test Steps

Checking if there is an autodiscover CNAME record in DNS for your domain mem13.ssddevrd.com for Office 365.
Failed to validate autodiscover CNAME record in DNS. If your mailbox isn't in Office 365, you can ignore this warning.
[Tell me more about this issue and how to resolve it](#)

You see the following error when the Remote server cannot be resolved.

Connectivity Test Failed

Test Details Start Over Run Test Again

Exchange Web Services synchronization, notification, availability, and Automatic Replies.
Not all tests of Exchange Web Services tasks completed.

Additional Details

Test Steps

Creating a temporary folder to perform synchronization tests.
Failed to create temporary folder for performing tests.

Additional Details

Exception details:

Message: The request failed. The remote name could not be resolved: 'mail-mem1.ssddevrd.com'

Type: Microsoft.Exchange.WebServices.Data.ServiceRequestException

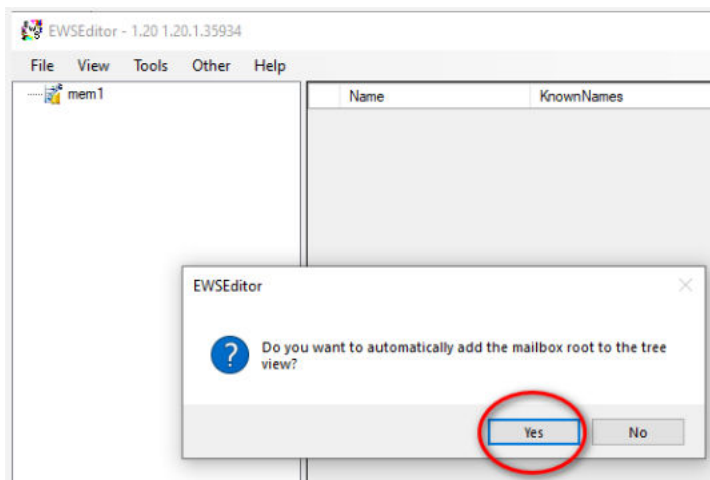
Stack trace:

```
at Microsoft.Exchange.WebServices.Data.ServiceRequestBase.BuildEwsHttpRequest()
at Microsoft.Exchange.WebServices.Data.ServiceRequestBase.ValidateAndEmitRequest(IEwsHttpRequest request)
at Microsoft.Exchange.WebServices.Data.MultiResponseServiceRequest`1.Execute()
at Microsoft.Exchange.WebServices.Data.ExchangeService.BindToFolder(FolderId folderId, PropertySet propertySet)
at Microsoft.Exchange.WebServices.Data.ExchangeService.BindToFolder(TFolderId folderId, PropertySet propertySet)
at Microsoft.Exchange.Tools.ExRca.Tests.GetOrCreateSyncFolderTest.PerformTestReally()
Exception details:
Message: The remote name could not be resolved: 'mail-mem1.ssddevrd.com'
Type: System.Net.WebException
Stack trace:
at System.Net.HttpWebRequest.GetRequestStream(TransportContext& context)
at System.Net.HttpWebRequest.GetRequestStream()
at Microsoft.Exchange.WebServices.Data.ServiceRequestBase.TraceAndEmitRequest(IEwsHttpRequest request, Boolean needSignature, Boolean needTrace)
at Microsoft.Exchange.WebServices.Data.ServiceRequestBase.BuildEwsHttpRequest()
Elapsed Time: 172 ms.
```

Troubleshooting the EWS Accessibility on an On-Premises ENS Installation

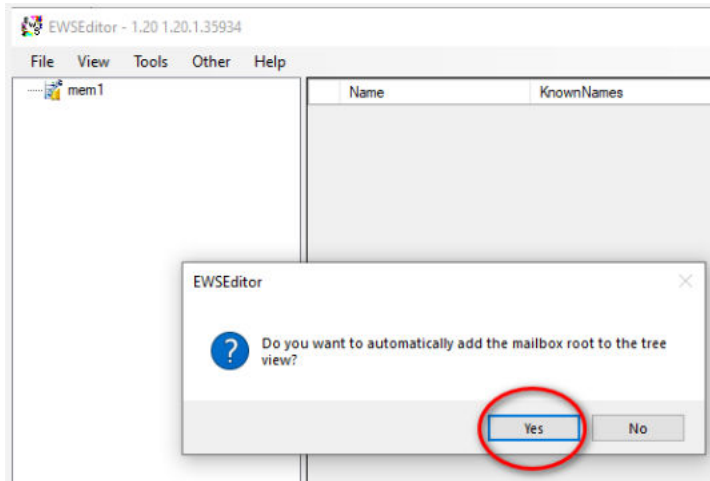
Use the EWSEditor tool to check if the EWS is internal and accessible from an on-premises ENS. The EWSEditor tool works only if you are using basic authentication or Open Authorization (OAuth).

- 1 Download and extract the EWSEditor ZIP file from the [EWSEditor](#).
- 2 Run the **EWSEditor.exe** file.
- 3 Navigate to the **File > New Exchange Service** and enter the **Service URL**, **User Name**, **Password**, and **Domain**.
- 4 Click **OK**. If there is an error in the details entered, then an appropriate error message appears. If the details entered are correct, then the following message appears:



- 5 Click **Yes**.

- 6 Select the device for which you want to check the subscription and right-click on the device. Select **Open Streaming Notifications Viewer**.
- 7 Click **Subscribe** and **Clear Events**.
- 8 To test the notifications, send a test message to the device. If the test is successful, the following screen appears:



Troubleshooting the Console Configuration Issues

You can configure the ENS2 settings using the configuration key and configuration value provided by the Workspace ONE UEM console.

The following image shows the ENS2 settings when configured without EWS URL and with the EWS URL.

ENS2 settings configured without the EWS URL

Configuration Key	Value Type	Configuration Value	
ENSLinkAddress	String	https://a: om/mailnotificator	✗ + Insert Lookup Value
ENSAPIToken	String	17413c 3d88c08	✗ + Insert Lookup Value
AccountNotifyPush	Boolean	True	✗ + Insert Lookup Value
+ Add			

ENS2 settings configured with the EWS URL

Configuration Key	Value Type	Configuration Value	
ENSLinkAddress	String	https://a: m/mailnotificator	✗ + Insert Lookup Value
ENSAPIToken	String	1741 08	✗ + Insert Lookup Value
AccountNotifyPush	Boolean	True	✗ + Insert Lookup Value
EWSUrl	String	https://outlook.office365.com/EWS/Exc	✗ + Insert Lookup Value
+ Add			

The following table lists the Workspace ONE UEM console configuration keys and values for ENS2.

Configuration Key	Value Type	Configuration Value
ENSLinkAddress	String	<p>Specify the URL address of the ENS server.</p> <ul style="list-style-type: none"> ■ For cloud deployments, the URL must be <code>https://ens.getboxer.com/api/ens</code>. Based on your region, VMware provides the resolved name or IP address. ■ For on-premises deployments, the URL must be <code>https://mycompany.com/MailNotificationService/api/ens</code>, where mycompany.com is the IP address or domain name for your ENS server.
ENSAPIToken	String	VMware provides the API token to enable the ENS service. For the on-premises installation, the on-premises installer creates this token.
AccountNotifyPush	Boolean	This value must be True .
EWSUrl	String	<p>Enables manual configuration of the Exchange Web Services (EWS) endpoint when the autodiscovery is disabled in your Exchange environment. Even for deployments where the autodiscovery is enabled, you must prefer to configure this option. The value of this option is your EWS endpoint. For example, <code>https://outlook.office365.com/EWS/Exchange.asmx</code> (for Office 365)</p>

Troubleshooting ENS2 Notification Issues

ENS notifications are applicable only for emails in the Inbox folder. The notifications do not work for Calendar events, sub folders, and so on. This topic describes the steps to troubleshoot the ENS2 notification issues for emails in the Inbox folder.

Public Key Request from the ENS

The Boxer application requests the public key from the ENS. The public key is used to encrypt the user credentials. When the ENS processes the request, the ENS sends the public key and creates a user record in the database against the user ID. In the following sample, the ENS logs for the **GetPublicKeyRequest**, the Boxer application sends the SHA256 hash of the email address as the user ID.

```

2019/10/18 05:54:05.395 WIN-HTCPEDXIUVF 7b21cd56-4c45-4a7c-88d9-a7f225cea3b9 [00000000-00000000]
(5) Debug MailNotificationService.Controllers.EnsController.GetPublicKey User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Processing Get Public key request
for Userid[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d]
2019/10/18 05:54:05.457 WIN-HTCPEDXIUVF 7b21cd56-4c45-4a7c-88d9-a7f225cea3b9 [00000000-00000000]
(5) Debug

```

```
MailNotificationService.BusinessImpl.GetPublicKeyBusiness.ProcessGetPublicKeyRequestAsync User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Key generated for user id
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d]
2019/10/18 05:54:05.457 WIN-HTCPEDXIUVF 7b21cd56-4c45-4a7c-88d9-a7f225cea3b9 [00000000-00000000]
(5) Debug MailNotificationService.Controllers.EnsController.GetPublicKey User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Get Public Key request processed.
HttpStatusCode:[OK] ResponseCode:[UpdateSuccess]
```

The possible error types and solutions that you might see during a **GetPublicKeyRequest** is listed as follows:

Error: Unauthorized Request

If you see the following error when you send a **GetPublicKeyRequest**, then ensure that the provided API token is correct. Verify if the API token is the same at the following instances:

- API token in the ENS logs - API token : [12341*****fasdf]
- The Boxer application configuration in the UEM console. See, the [Workspace ONE Boxer Admin Guide](#) for more information on the Boxer application configuration values.
- API token in the Boxer application logs - Verify the API token in the Boxer application logs.

Error: Unable to add a NULL value into the PublicKey column

Note This section is applicable for an on-premises installation only.

When the available RSA keys in the database are exhausted, you might see the following error. This issue is automatically fixed when the RSAKey tracker service triggers and generates new keys again.

```
2019/10/18 12:20:04.121 WIN-HTCPEDXIUVF b4a42dc8-6896-4243-9a4c-8ed476ae94ab [00000000-00000000]
(5) Debug MailNotificationService.Controllers.EnsController.GetPublicKey User Id
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Processing Get Public key request
for Userid[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d]
2019/10/18 12:20:04.136 WIN-HTCPEDXIUVF b4a42dc8-6896-4243-9a4c-8ed476ae94ab [00000000-00000000]
(5) Debug
MailNotificationService.BusinessImpl.GetPublicKeyBusiness.ProcessGetPublicKeyRequestAsync User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Error: 515, Severity: 16, State:
2, Message: "Cannot insert the value NULL into column 'PublicKey', table 'onpremdev.dbo.UserInfo';
column does not allow nulls. INSERT fails.", Procedure: "UserInfo_Save", Line: 39
2019/10/18 12:20:04.136 WIN-HTCPEDXIUVF b4a42dc8-6896-4243-9a4c-8ed476ae94ab [00000000-00000000]
(5) Debug MailNotificationService.Controllers.EnsController.GetPublicKey User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Get Public Key request processed.
HttpStatusCode:[InternalServerError] ResponseCode:[UpdateFail]
```

Note The RSAKey tracker trigger interval time is 120 minutes. If the number of keys available in the database during the tracker trigger time is less than 250, then the RSAKey tracker starts generating a new batch of RSA keys. By default, the RSAKey tracker generates 500 new keys at a time.

Ensure that the following values are present in the RSAKey tracker configuration file:

```
<add key="numberOfKeysToBeInserted" value="500"/>
<add key="wakeUpIntervalInMins" value="120"/>
<add key="keysThreshold" value="250"/>
```

Error: ENS service communication

When communicating with the ENS service, if you see the following error in the Boxer application logs, then ensure that your device has proper connectivity.

```
2019-11-04T12:23:12Z E [710337] [ENS] An error occurred when communicating with the ENS service:
Error Domain=NSURLErrorDomain Code=-1009 "The Internet connection appears to be offline."
UserInfo={NSUnderlyingError=0x281fead90 {Error Domain=kCFErrorDomainCFNetwork Code=-1009 "The
Internet connection appears to be offline." UserInfo={NSErrorFailingURLStringKey=https://ens-
staging.getboxer.com/api/ens/getpublickey, NSErrorFailingURLKey=https://ens-
staging.getboxer.com/api/ens/getpublickey, _kCFStreamErrorCodeKey=50, _kCFStreamErrorDomainKey=1,
NSLocalizedString=The Internet connection appears to be offline.}},
NSErrorFailingURLStringKey=https://ens-staging.getboxer.com/api/ens/getpublickey,
NSErrorFailingURLKey=https://ens-staging.getboxer.com/api/ens/getpublickey,
_kCFStreamErrorDomainKey=1, _kCFStreamErrorCodeKey=50, NSLocalizedString=The Internet connection
appears to be offline.} at URL: https://ens-staging.getboxer.com/api/ens/getpublickey. Data: .
Response Code: 0
2019-11-04T12:23:12Z E [710337] [ENS] Error registering new account: vmwUser4@awRed.onmicrosoft.com
Error:Error Domain=com.alamofire.serialization.response.error.response Code=-1 "invalid public key"
UserInfo={NSLocalizedString=invalid public key}
2019-11-04T12:23:12Z E [703318] [ENS] Error registering device for push notification
Error:Error Domain=com.alamofire.serialization.response.error.response Code=-1 "invalid public key"
UserInfo={NSLocalizedString=invalid public key}
2019-11-04T12:23:12Z E [726177] - Unexpected error: {
    BXLocalizedStringErrorMessageKey = "Could not update settings for the push notification service";
    BXLocalizedStringErrorKey = "Could not update settings for the push notification service";
    NSLocalizedString = "Could not update settings for the push notification service. ";
    NSLocalizedStringFailureReason = "Failed to update push notification settings. Please contact your
administrator.";
} context: 1
```

Register Device Request

The Boxer application sends a **Register** request to the ENS, a push subscription to the EWS, and a subscribe for notification. If the **GetPublicKey** request is successful, then the Boxer application sends a register request to the ENS with the necessary information required to register a device for notification.

Scenario 1: - If the EWS URL is not configured in the console, then the ENS tries autodiscovery to obtain the EWS URL to subscribe the user.

Configuration Key	Value Type	Configuration Value	
ENSLinkAddress	String	https://a[REDACTED]om/mailnotification	✗ + Insert Lookup Value
ENSAPIToken	String	17413[REDACTED]3d88c08	✗ + Insert Lookup Value
AccountNotifyPush	Boolean	True	✗ + Insert Lookup Value
+ Add			

Scenario 2: - If the EWS URL is configured in the console, then the ENS uses the same EWS URL to subscribe the user.

Configuration Key	Value Type	Configuration Value	
ENSLinkAddress	String	https://a...m/mailnotificator	✖ + Insert Lookup Value
ENSAPIToken	String	1741...08	✖ + Insert Lookup Value
AccountNotifyPush	Boolean	True	✖ + Insert Lookup Value
EWSUrl	String	https://outlook.office365.com/EWS/Exc	✖ + Insert Lookup Value

+ Add

When the subscription is successful, the ENS receives the **[UserSubscribed]** message with the subscription ID as mentioned in the following code snippet.

```
2019/11/05 08:18:49.674 A3 726c4072-5144-4450-848b-821f65174b89 [00000000-00000000] (23)
Info MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id: [1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] User
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] subscribed with subscriptionId
[JwbTbjJwcjE5bWizMDA1Lm5hbXByZDE5LnByb2Qub3V0bG9vay5jb20QAAAAJ6RYazaIoUCfX7KheUsQYUQnw9rIYdcIEAAAAAQ9t
cFCKSZFrT0xLbSCwj4=]
2019/11/05 08:18:49.767 A3 726c4072-5144-4450-848b-821f65174b89 [00000000-00000000] (28)
Debug MailNotificationService.Controllers.EnsController.RegisterDeviceV2 User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Register device request
processed. HttpStatusCode:[OK] ResponseCode:[UserSubscribed]
```

In the Android Boxer logs, you must see the following log entries to confirm a successful registration:

```
-----
ENS SETTINGS
-----
ENS_LINK_ADDRESS = https://ens.getboxer.com/api/ens
ENS_API_TOKEN = 17413*****88c08
POLICY_ACCOUNT_NOTIFY_PUSH = true
EWS_URL =
ENS_STATE = (8 -> Registered)

-----
HEALTH STATUS
-----
App version health status: Green, Current app version: 5.11.0.4, New version: 5.10.0
Sync Health Status: Green, Sync durations in seconds: [0.522, 0.49, 0.416, 0.379, 0.424, 0.368,
0.465, 0.496, 0.565, 1.344], Sync results [OK, OK, OK, OK, OK, OK, OK, OK, OK, OK]
Ens health status: Green , Ens state: Registered
Overall health status: Green
Ens registration for account (id=8) is successful!
```


For the iOS Boxer logs, you must see the following log entries to confirm a successful registration:

```
For normal subscription
2019-11-11T09:31:41Z I [12347] [ENS] Successfully registered account.
```

Note For iOS Boxer logs, open the Boxer application, navigate to the **Boxer Settings**, click the **VMware Secure Email**, and ensure the **Use Push Service** switch is enabled to confirm a successful ENS registration.

The possible errors and solutions that you might see when you are unable to locate the autodiscover services are listed as follows:

Error: Unable to Locate the Autodiscover Services

If you see the following error, then ensure to enable autodiscovery, check the availability and connectivity of the autodiscovery server using the EWSEditor and the MS remote connectivity analyzer.

```
2019/11/06 07:01:56.207 A3 d252be19-1c5d-4e30-9155-a0ae3a529679 [00000000-00000000] (94)
Warn MailNotificationService.BusinessImpl.SubscriptionBusiness.SubscribeV2Async User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Exception while auto discovery
occured for userId [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586], Exception
Message [The Autodiscover service couldn't be located.] , Exception
[Microsoft.Exchange.WebServices.Data.AutodiscoverLocalException: The Autodiscover service couldn't be
located.
at
Microsoft.Exchange.WebServices.Autodiscover.AutodiscoverService.InternalGetLegacyUserSettings[TSetting
s](String emailAddress, List`1 redirectionEmailAddresses, Int32& currentHop)
at Microsoft.Exchange.WebServices.Autodiscover.AutodiscoverService.GetLegacyUserSettings[TSettings]
(String emailAddress)
at
Microsoft.Exchange.WebServices.Autodiscover.AutodiscoverService.InternalGetLegacyUserSettings(String
emailAddress, List`1 requestedSettings)
at Microsoft.Exchange.WebServices.Data.ExchangeService.GetAutodiscoverUrl(String emailAddress,
ExchangeVersion requestedServerVersion, AutodiscoverRedirectionUrlValidationCallback
validateRedirectionUrlCallback)
at Microsoft.Exchange.WebServices.Data.ExchangeService.AutodiscoverUrl(String emailAddress,
AutodiscoverRedirectionUrlValidationCallback validateRedirectionUrlCallback)
at
MailNotificationService.BusinessImpl.ExchangeServiceBusiness.<GetExchangeServiceViaAutoDiscovery>d__10
.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
at
MailNotificationService.BusinessImpl.ExchangeServiceBusiness.<GetExchangeServiceAsync>d__6.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
at MailNotificationService.BusinessImpl.SubscriptionBusiness.<SubscribeV2Async>d__7.MoveNext(),
Inner Exception [], Autodiscover url used [The Autodiscover service couldn't be located.]
2019/11/06 07:01:56.207 A3 d252be19-1c5d-4e30-9155-a0ae3a529679 [00000000-00000000] (94)
Debug MailNotificationService.Controllers.EnsController.RegisterDeviceV2 User Id:
```



```
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Register device request processed. HttpStatusCode:[Conflict] ResponseCode:[SubscribeAgain]
```

Error: The remote server returned an error (403) Forbidden

If this error occurs during a subscription, then ensure to enter the proper EWS URL in the Boxer application KVP values of the UEM console. The **EWSUrl** used to subscribe must have the complete endpoint specified.

Example of a correct **EWSUrl** - [https://mail-mem13.xyz.com/EWS/exchange.asmx]

Example of an incorrect **EWSUrl** - [https://mail-xyz.com/]

To check the EWS URL availability and connectivity, check the EWSEditor and the MS remote connectivity analyzer.

```
2019/11/06 07:09:54.064 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.Controllers.EnsController.RegisterDeviceV2 User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Processing register device
request for Userid[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586]
2019/11/06 07:09:54.080 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug
MailNotificationService.BusinessImpl.RegisterDeviceBusiness.ProcessRegisterDeviceRequestAsyncV2 User
Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Exchange version sent by boxer
[2]
2019/11/06 07:09:54.080 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeServiceBusiness.GetExchangeServiceAsync User
Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Using client ewsurl,
mailServerUrlMatched : False, deletedEWSUrl: False
2019/11/06 07:09:54.080 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] EWSUrl used to subscribe:
[https://mail-mem13.ssdevrd.com/]
2019/11/06 07:09:54.080 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] User subscribing with
[Basic Auth]
2019/11/06 07:09:54.173 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Warn MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotifications User
Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Service request exception
occured for userId [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586], Inner
exception message [The remote server returned an error: (403) Forbidden.] Going for a retry,
2019/11/06 07:09:54.173 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] EWSUrl used to subscribe:
[https://mail-mem13.ssdevrd.com/]
2019/11/06 07:09:54.173 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] User subscribing with
[Basic Auth]
2019/11/06 07:09:54.205 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Warn MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotifications User
Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Service request exception
occured for userId [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586], Inner
exception message [The remote server returned an error: (403) Forbidden.] Going for a retry,
```

```

2019/11/06 07:09:54.205 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] EWSUrl used to subscribe:
[https://mail-mem13.ssdevrd.com/]
2019/11/06 07:09:54.205 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] User subscribing with
[Basic Auth]
2019/11/06 07:09:54.236 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Warn MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotifications User
Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Service request exception
occured for userId [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586], Inner
exception message [The remote server returned an error: (403) Forbidden.] Going for a retry,
2019/11/06 07:09:54.236 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] EWSUrl used to subscribe:
[https://mail-mem13.ssdevrd.com/]
2019/11/06 07:09:54.236 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] User subscribing with
[Basic Auth]
2019/11/06 07:09:54.251 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Warn MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotifications User
Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Service request exception
occured for userId [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586], Inner
exception message [The remote server returned an error: (403) Forbidden.] Going for a retry,
2019/11/06 07:09:54.251 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Warn MailNotificationService.BusinessImpl.SubscriptionBusiness.SubscribeV2Async User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Service request exception
occured for userId [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586], Inner
exception message [The remote server returned an error: (403) Forbidden.] Going for a retry,
2019/11/06 07:09:54.251 A3 f43eb3d0-e173-49de-9b52-3acb8a1107c4 [00000000-00000000] (98)
Debug MailNotificationService.Controllers.EnsController.RegisterDeviceV2 User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Register device request
processed. HttpStatusCode:[Conflict] ResponseCode:[SubscribeAgain]

```

Sample error logs of Boxer during registration:

```

2019-11-11T09:13:43Z E [9326] [ENS] An error occurred when communicating with the ENS service: Error
Domain=com.alamofire.error.serialization.response Code=-1011 "Request failed: conflict (409)"
UserInfo={NSLocalizedDescription=Request failed: conflict (409), NSErrorFailingURLKey=https://
a3.ssdevrd.com/mailnotificationsservice/api/ens/registerdevicev2,
com.alamofire.serialization.response.error.data={length = 135, bytes = 0x7b227265 7370666e 7365436f
6465223a ... 4f6e5072 656d227d },
com.alamofire.serialization.response.error.response=<NSHTTPURLResponse: 0x282db1fa0> { URL: https://
a3.ssdevrd.com/mailnotificationsservice/api/ens/registerdevicev2 } { Status Code: 409, Headers {
    "Content-Length" = (
        135
    );
    "Content-Type" = (
        "application/json; charset=utf-8"
    );
    Date = (
        "Mon, 11 Nov 2019 09:13:40 GMT"
    );
}
}

```

```

    Server =      (
        "Microsoft-IIS/8.5"
    );
    "X-Powered-By" =      (
        "ASP.NET"
    );
} }} at URL: https://a3.ssdevrd.com/mailnotificationservice/api/ens/registerdevicev2. Data:
{"responseCode":14,"errorMessage":"The Autodiscover service couldn't be
located.", "version":"1.5.7235.6268","environmentType":"OnPrem"}. Response Code: 409
2019-11-11T09:13:43Z  E [9326] [ENS] registerAccountOnENS: Error updating settings or credentials
Error:Error Domain=com.alamofire.error.serialization.response Code=-1011 "Request failed: conflict
(409)" UserInfo={NSLocalizedDescription=Request failed: conflict (409), NSErrorFailingURLKey=https://
a3.ssdevrd.com/mailnotificationservice/api/ens/registerdevicev2,
com.alamofire.serialization.response.error.data={length = 135, bytes = 0x7b227265 7370666e 7365436f
6465223a ... 4f6e5072 656d227d },
com.alamofire.serialization.response.error.response=<NSHTTPURLResponse: 0x282db1fa0> { URL: https://
a3.ssdevrd.com/mailnotificationservice/api/ens/registerdevicev2 } { Status Code: 409, Headers {
    "Content-Length" =      (
        135
    );
    "Content-Type" =      (
        "application/json; charset=utf-8"
    );
    Date =      (
        "Mon, 11 Nov 2019 09:13:40 GMT"
    );
    Server =      (
        "Microsoft-IIS/8.5"
    );
    "X-Powered-By" =      (
        "ASP.NET"
    );
} }}
2019-11-11T09:13:43Z  E [9365] - Unexpected error: {
    BXLocalizedContextMessageErrorKey = "Could not update settings for the push notification service";
    BXLocalizedTitleErrorKey = "Could not update settings for the push notification service";
    NSLocalizedDescription = "Could not update settings for the push notification service. ";
    NSLocalizedFailureReason = "Failed to update push notification settings. Please contact your
administrator.";
}

```

In the sample error logs of Boxer, you can see the following message:

```

{"responseCode":14,"errorMessage":"The Autodiscover service couldn't be
located...

```

In this case, ensure that the autodiscovery URL is reachable from the ENS and the autodiscovery URL is configured correctly using the EWSEditor tool or MS connectivity analyzer tool.

If you are using the **EWSUrl**, ensure that the **EWSUrl** key is configured in the console with a correct value for the **EWSUrl** of their respective Exchange environments. To verify the **EWSUrl** is correct, open a browser, enter the **EWSUrl**, and ensure that you are prompted to enter the credentials.

You can find the error message and response code for different reasons. Based on the error message, you can start troubleshooting the issue.

Error: 403 or 401 error message

EWS must be accessible to the ENS application to subscribe the user for notification. If the EWS is not configured correctly, then you might receive 403 or 401 error. In such cases, refer the following documents:

- [Getting started with the EWS Managed API 2.0](#)
- [Managing access for EWS Managed API 2.0 applications](#)
- [Authentication and EWS in Exchange](#)

Check the type of authentication you have enabled in the EWS. Ensure that the authentication is in parity with what the customer is using for ActiveSync (Basic, OAuth, and CBA). The Boxer application sends the user credentials to the ENS and the ENS uses the same credentials and the same type of authentication to communicate with the EWS.

Note If the ENS can access the Office 365 and the Active Directory Federation Services (ADFS), then ensure that either the ENS IPs are whitelisted on the ADFS or the affected user has no block claim on the ADFS.

If you are using Office 365 and you receive a 401 error from the EWS URL, the reason for the error might be because the client access rules or ADFS claims are configured. In such scenarios, refer the following documents.

- [Client Access Rules in Exchange Online](#)
- [Customizing ADFS Claims Rules for Office 365](#)

In a scenario where the ENS on-premises Exchange with CBA is enabled, you might need to confirm that the client certificate is arriving at the Exchange endpoint. To troubleshoot any errors, see the [#unique_29](#) topic.

Force Register or Re-register on the Boxer application:

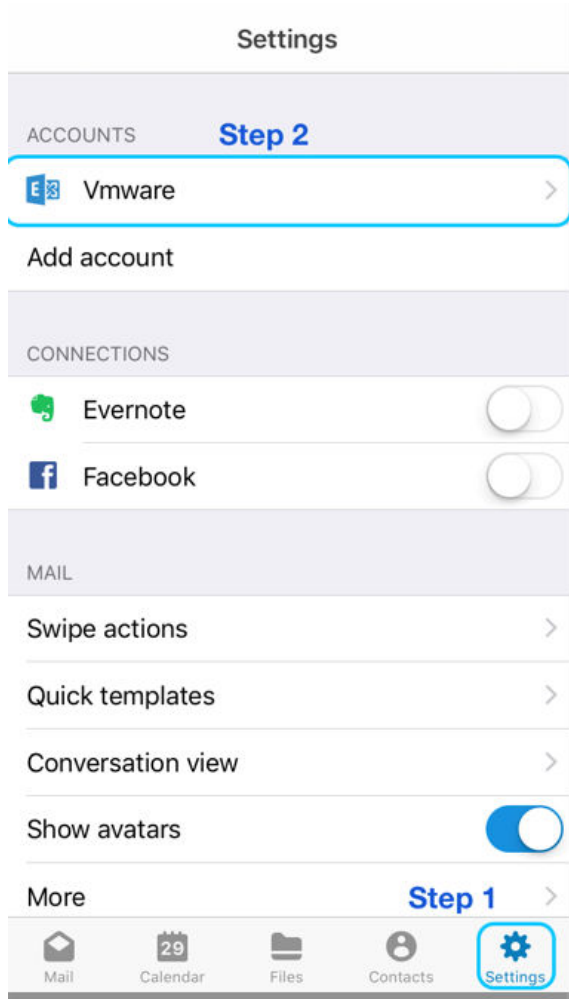
On iOS devices only, you can manually perform a force subscription, in the following cases:

- If there are any changes to the keys in console, then you must approximately wait for 1 hour and check if the users are still receiving the notification. If the users are not receiving notifications, you can proceed to re-register the Boxer application with the ENS2 service.
- If you do not see any register request in the ENS logs from the Boxer application, then assume that the Boxer application has failed to send the register request automatically. Therefore, the ENS tries to re-register the Boxer application with the ENS2 service forcefully.

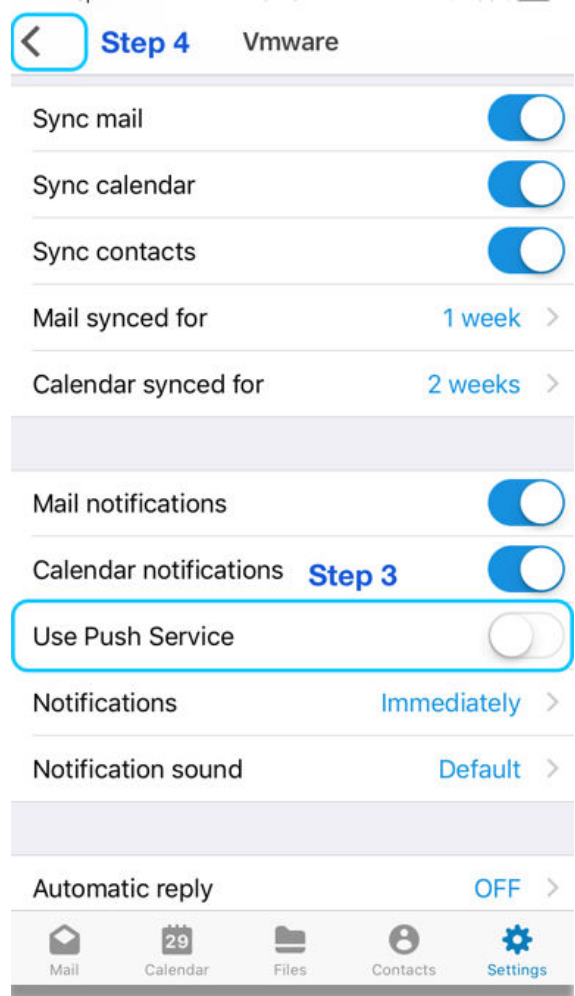
To force register or re-register on the Boxer application, perform the following steps:

- 1 Open the Boxer application and click **Settings**.

- 2 Under the **Accounts** tab, select your ENS-specific account.

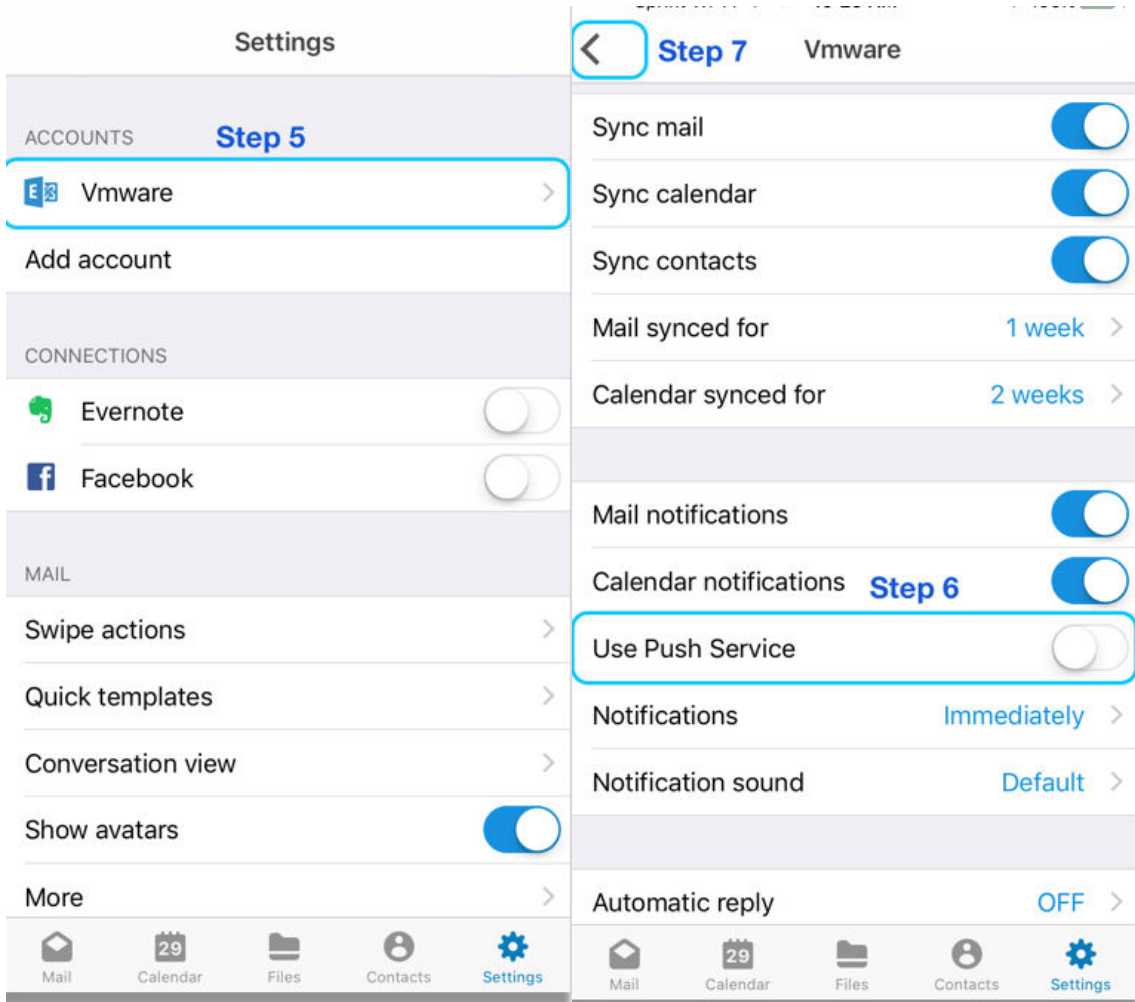


- 3 Turn off the **Use Push Service** option.



- 4 Navigate to the Boxer application **Settings** screen.

- 5 Repeat Step 2 through Step 4 to turn on the **Use Push Service** option.



When you perform either of the steps mentioned, then you can see the force register request in the ENS logs.

To confirm the force subscription in the ENS logs, search for the **ForceSubscription** and you must be able to see the following value: **ForceSubscription : [True]**.

Registration Status Events

If the registration is successful, then the Exchange sends a status event to the ENS periodically against each subscription ID, to confirm the subscription. The ENS then sends an acknowledgment for each of the subscription IDs back to the Exchange.

```
2019/11/05 08:57:31.413 A3 1eb9186b-9370-45de-a172-0e452586f398 [00000000-00000000] (58)
Debug MailNotificationService.BusinessImpl.ExchangeNotificationParser.ScanEventNotificationAsync
User Id:[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Received [StatusEvent]
for subscription:
[JwbTbjJwcjE5bWIZMDA1Lm5hbXByZDE5LnByb2Qub3V0bG9vay5jb20QAAAA14H5dKboFUm1kJ8ZNBKkJILRTBjMYdcIEAAAAQ9t
cFCKSZFrT0xLbSCwj4=]
2019/11/05 08:57:31.413 A3 1eb9186b-9370-45de-a172-0e452586f398 [00000000-00000000] (58)
Debug MailNotificationService.BusinessImpl.PushNotificationBusiness.HandleExchangeEvents User Id:
```

```
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Status event received for user:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d]
```

If the ENS receives the status event for the old subscription ID, then the ENS responds to the Exchange with an unsubscribe response as shown in the following logs.

```
2019/11/05 08:49:20.123 A3 d2adec8a-73d7-48f2-ba14-abbd917844cd [00000000-00000000] (54)
Info MailNotificationService.BusinessImpl.ExchangeNotificationParser.ScanEventNotificationAsync
User Id:[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] This
JwBtbjJwcjE5bWIZMDA1Lm5hbXBZDE5LnByb2Qub3V0bG9vay5jb20QAAAAJ6RYazaIoUCfX7KheUsQYUQnw9rIYdcIEAAAAAQ9tc
FCKSZFrT0xLbSCwj4= is old subscription for user
1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d, sending unsubscribe response
2019/11/05 08:49:20.123 A3 d2adec8a-73d7-48f2-ba14-abbd917844cd [00000000-00000000] (54)
Debug MailNotificationService.BusinessImpl.ExchangeNotificationParser.ScanEventNotificationAsync
User Id:[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Sent Unsubscribe response
to EWS successfully for subscriptionId:
[JwBtbjJwcjE5bWIZMDA1Lm5hbXBZDE5LnByb2Qub3V0bG9vay5jb20QAAAAJ6RYazaIoUCfX7KheUsQYUQnw9rIYdcIEAAAAAQ9t
cFCKSZFrT0xLbSCwj4=]
2019/11/05 08:49:20.123 A3 d2adec8a-73d7-48f2-ba14-abbd917844cd [00000000-00000000] (54)
Debug
MailNotificationService.BusinessImpl.PushNotificationBusiness.ProcessPushNotificationV2Async User
Id:[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d]
ProcessNotificationResponse.IsUnSubscribeResponse is true
```

For more information on the status frequency, see the [StatusFrequency](#) topic.

ENS must receive the status events from the Exchange immediately after a subscription is successful. If the ENS is not receiving the status events, then check the following troubleshooting methods to verify the communication between the Exchange server and the ENS.

Error: Status event not received

If you do not see any status events in the ENS logs after a successful subscription, then check the communication between the Exchange server and the ENS. Access the following URLs in the browser on the CAS or the mailbox servers to check the communication between the Exchange and the ENS.

- For on-premises ENS deployments, use the `https://{ENS URL}/MailNotificationService/api/ens/alive`.
- For cloud ENS deployments, use the `https://{ENS URL}/api/ens/alive`. For example, <https://ens.getboxer.com/api/ens/alive>. Select the ENS cloud URL based on your region.

You must be able to see the following result when you browse the specified URLs from the browser.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">
  ens.getboxer.com is alive. Version = 1.5.7227.9937 Environment = Production InstanceId = i-04676f24928463e31
</string>
```

- For on-premises ENS deployments, use the `https://{ENS URL}/MailNotificationService/api/ens/pushnotificationlistener`.
- For cloud ENS deployments, use the `https://{ENS URL}/api/ens/pushnotificationlistener`. Select the ENS cloud URL based on your region.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="utf-8" ?>
<Error>
  <Message>
    The requested resource does not support http method 'GET'.
  </Message>
</Error>
```

Note When browsing the URLs, if you see any SSL error, then proceed to import the ENS certificate in the MMC of the server.

New Mail Event and Fetch Mail

When a device is successfully registered and the communication between the ENS and the Exchange is working correctly, the Exchange starts sending new mail events to the ENS whenever a new mail is received on the subscribed user mailbox. If the payloads of the created events contain an unread count, then the ENS uses the unread count, else the ENS gets the unread count from the EWS.

```
2019/11/05 09:39:56.608 A3 9f08ed6d-0726-430c-8440-9c396443c7ca [00000000-00000000] (74)
Debug MailNotificationService.BusinessImpl.ExchangeNotificationParser.ScanEventNotificationAsync
User Id: [1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Received [CreatedEvent]
for subscription:
[JwbTbjJwcjE5bWIZMDA1Lm5hbXBzZDE5LnByb2Qub3V0bG9vay5jb20QAAAAA14H5dKboFUm1kJ8ZNBKkJILRTBjMYdcIEAAAAAQ9t
cFCKSZFrTOxLbSCwj4=]
2019/11/05 09:39:56.639 A3 9f08ed6d-0726-430c-8440-9c396443c7ca [00000000-00000000] (74)
Debug MailNotificationService.BusinessImpl.UnreadCountExchangeBusiness.GetUnReadCountV2 User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] EWSUrl used to get unread count:
[https://outlook.office365.com/EWS/Exchange.asmx]
2019/11/05 09:39:56.889 A3 9f08ed6d-0726-430c-8440-9c396443c7ca [00000000-00000000] (74)
Info MailNotificationService.BusinessImpl.PushNotificationBusiness.HandleNewMailEvent User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Received new mail event for user
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] with BADGE count [893]
```

Whenever the ENS receives a new mail event, the ENS fetches the mail information from the Exchange. The possible errors and solutions that you might see during a fetch mail request is listed as follows:

Error: Stuck in EWSUrl used to sync email: [https://outlook.office365.com/EWS/Exchange.asmx] steps

When a mail event is received from the Exchange, the ENS tries to fetch all the information from the mail. If you are unable to see any ENS logs such as the **Fetched email**, then check the respective EWS logs in the Exchange. You can obtain the corresponding EWS logs using the client request ID or the activity ID.

Fetch New Mail Request

Sample client request ID or the activity ID: 03ea7f36-f72f-4322-8413-0dcd81c4ac78

Note You can get the client request ID or the activity ID in the third column of the ENS logs. Copy that ID and search for the client request ID or the activity ID in the EWS logs.

ENS sends a push notification request to the CNS or the SNS

When the new mail information is fetched from the Exchange, the ENS composes and sends a notification payload to the CNS (for on-premises) or the SNS (for cloud).

Sample of sending a notification payload to the CNS (for on-premises)

```
2019/11/05 09:48:42.675 A3 fedf9a1d-6cc8-4607-acad-ae006766292a [0000000-0000000] (82)
Info MailNotificationService.BusinessImpl.NotificationsProcessor.AddNotificationToBatch User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] About to Post Notification for
user : [1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] and Device Id : [1]
2019/11/05 09:48:42.690 A3 fedf9a1d-6cc8-4607-acad-ae006766292a [0000000-0000000] (82)
Info MailNotificationService.BusinessImpl.NotificationsProcessor.AddNotificationToBatch User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] About to Post Notification for
user : [1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] and Device Id : [5]
2019/11/05 09:48:47.699 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Info MailNotificationService.BusinessImpl.CNSHelper.ComposeAPNSPushNotification User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Total unread count retrieved
[894] for user [1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d]
2019/11/05 09:48:47.699 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Debug MailNotificationService.BusinessImpl.CNSHelper.ComposeAPNSPushNotification User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Sending to :: User :
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d], DeviceId : [1], DeviceLogId : [],
Message : messageId:
[AAMKAGMxYjUzZDA0LTl5NDItNDUyNi1hZDMzLWlXMMRiNDgyYzIzZQBGAAAAA0x2petA5rS4RDQM8RjW1TBwDnJcIsAp4/
S4beDDAIaXMhAAAAAEMAADnJcIsAp4/S4beDDAIaXMhAAGszQatAAA=]
2019/11/05 09:48:47.699 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Info MailNotificationService.BusinessImpl.CNSHelper.ComposeAPNSPushNotification User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Total unread count retrieved
[894] for user [1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d]
2019/11/05 09:48:47.699 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Debug MailNotificationService.BusinessImpl.CNSHelper.ComposeAPNSPushNotification User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Sending to :: User :
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d], DeviceId : [5], DeviceLogId :
[61F9BB13-863C-444C-A300-4F888383ACDD-534-0000000CE599EDE0], Message : messageId:
[AAMKAGMxYjUzZDA0LTl5NDItNDUyNi1hZDMzLWlXMMRiNDgyYzIzZQBGAAAAA0x2petA5rS4RDQM8RjW1TBwDnJcIsAp4/
S4beDDAIaXMhAAAAAEMAADnJcIsAp4/S4beDDAIaXMhAAGszQatAAA=]
2019/11/05 09:48:47.699 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Debug MailNotificationService.BusinessImpl.CNSHelper.CreateWebRequest User Id:[no-user-id] CNS
Url : [https://cns.awmdm.com/nws/notify/apns]
2019/11/05 09:48:47.699 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Debug MailNotificationService.BusinessImpl.CertificateHelper.ComputeCmsSignature User Id:[no-user-
id] Signing URL [/nws/notify/apns] with Cert [CN=AW Cloud Notification - aTest]
2019/11/05 09:48:48.558 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Debug MailNotificationService.BusinessImpl.CNSHelper.ReadResponse User Id:[no-user-id] Response
{"status":"success","errorReason":null}
2019/11/05 09:48:48.558 A3 7e45c693-511b-4c19-ae7c-305e5f8f9f0e [0000000-0000000] (8)
Info MailNotificationService.BusinessImpl.CNSHelper.ReadResponse User Id:[no-user-id]
ResponseCode OK
```

Sample of sending a notification payload to the SNS (for cloud)

```
2019-09-06 12:11:51.5380|INFO|
MailNotificationService.BusinessImpl.NotificationsProcessor.AddNotificationToBatch|b1d8e164-c3fb-4f67-
baa6-002dd3719c4e|User Id:[35045e4062200ca81c92d5b03928a7e86383ef8e9436d512187a711a4b18e94f] About to
```

```
Post Notification for user [35045e4062200ca81c92d5b03928a7e86383ef8e9436d512187a711a4b18e94f]
2019-09-06 12:11:52.5537|INFO|MailNotificationService.BusinessImpl.AmazonSNSHelper.PostNotifications|
67d3c6f0-a197-4af4-958c-260eedbf567|User Id:
[35045e4062200ca81c92d5b03928a7e86383ef8e9436d512187a711a4b18e94f] Sending notification via SNS
2019-09-06 12:11:52.5692|INFO|
MailNotificationService.BusinessImpl.AmazonSNSHelper.PushNotificationViaSNS|67d3c6f0-
a197-4af4-958c-260eedbf567|User Id:
[35045e4062200ca81c92d5b03928a7e86383ef8e9436d512187a711a4b18e94f] Notification successfully sent via
SNS for [424716]
```

To confirm if your Android device is receiving notifications from the ENS, enable the Boxer application passcode and restart the device after a successful registration. You might see a notification, that is, a banner containing the email address configured. On the banner notification if you cannot perform actions such as, Delete, Reply, and Read option then, the notification is a push notification that is sent from the ENS and not locally from the Boxer application itself. If the notification banner contains notification actions such as Delete, Reply, Read, and so on, then the notification is a local notification from the Boxer application and not a push notification from the ENS.

The possible errors and solutions that you might see during a push notification request is listed as follows:

Error: The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel.

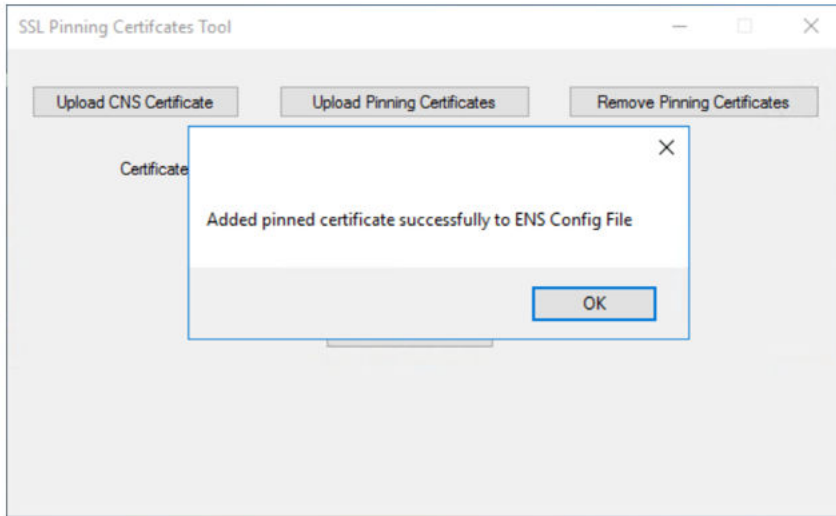
```
2019/11/06 09:03:48.218 A3 aa57f568-6871-42cc-8b8d-39c77a15af41 [00000000-00000000] (40)
Error MailNotificationService.BusinessImpl.CNSHelper.ReadResponse User Id:[no-user-id] Failed To
Post to CNS [https://cns.awmdm.com/nws/notify/apns] Error: [The underlying connection was closed:
Could not establish trust relationship for the SSL/TLS secure channel.] Response: []
```

Note Ensure you have followed the steps as mentioned in the [Configure CNS and Download Email Notification Service Configuration Files](#) topic.

If the issue still persists, download the latest public CNS certificate from <http://resources.workspaceone.com/view/2hjxzvgkxyf8n738hy7x/en> and perform the following steps:

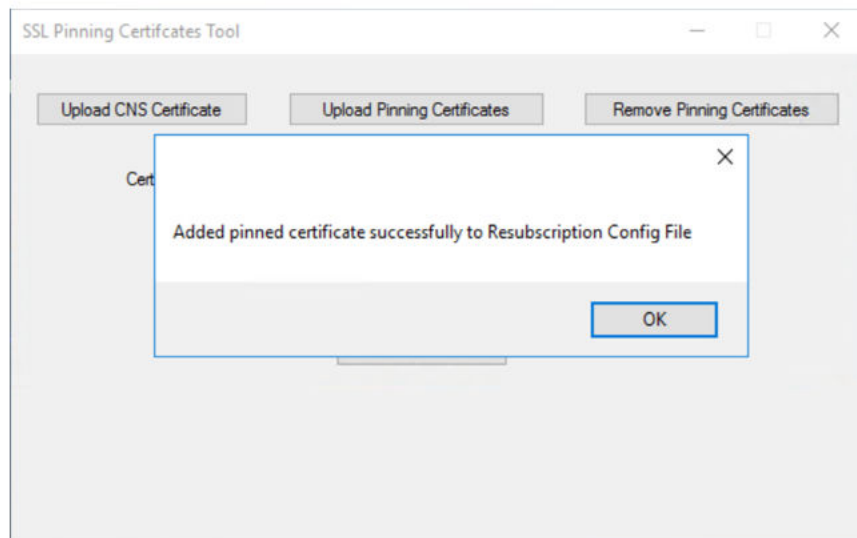
- 1 Click the **SSLPinningCertTool** shortcut present in the ENS server or click <ENS_INSTALL_DIR>\Email Notification Service\Tools\SSLPinningCertTool\SSLPinningCertTool.exe.
- 2 Click the **Upload CNS Certificate** button.

- 3 Select the certificate to be uploaded and click **Submit**. If the following screen appears, then the certificate is successfully added.



Note After uploading the SSL pinning certificate on the ENS, the tool adds the public key of the certificate to the ENS configuration. When the ENS posts payload to the CNS, the certificate validation is done against the newly added certificate public key.

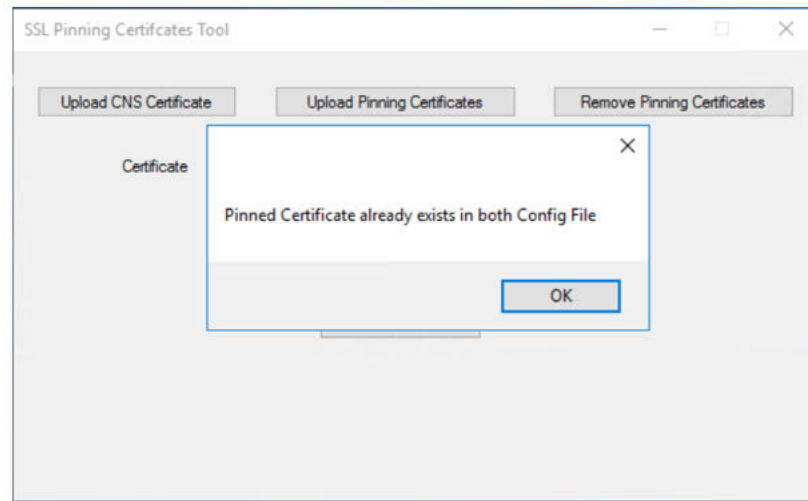
- 4 If the following screen appears, the certificate is successfully added to the resubscription



configuration file.

Note After uploading the SSL pinning certificate, the tool adds the public key of the certificate to the resubscription configuration file. For the resubscription mechanism, after payload (silent notification) to the CNS, the certificate validation is done against the newly added certificate public key.

- 5 If the certificate is already present in both the configuration files, then you are prompted with the



following message.

Note The upload pinning certificate occurs as follows:

- The tool tries to upload the certificate to the ENS configuration file only if the provided certificate is not present in the ENS configuration file. If the given certificate is already present, then the tool does not prompt any message and continues to upload the same certificate to the resubscription configuration file.
- The tool tries to add a certificate to the resubscription configuration file only if the provided certificate is not present in the resubscription configuration file. If the given certificate is present, then the tool does not prompt any message to the user.

- 6 If the certificate is added to the resubscription configuration file, then navigate to **Services** and restart the **AirWatch Resubscription Mechanism** service.

Error: The remote server returned an error: (401) Unauthorized.

Sample error log:

```
2019/11/06 09:25:13.688 A3 6c041e00-c909-45ff-b340-283844376c06 [00000000-00000000] (6)
Error MailNotificationService.BusinessImpl.CNSHelper.ReadResponse User Id:[no-user-id] Failed To
Post to CNS [https://cns.awmdm.com/nws/notify/apns] Error: [The remote server returned an error:
(401) Unauthorized.] Response: [{"code":2007,"message":"Unable to verify if the signer cert as
trusted. The associated request id is 154e9542-b695-497b-9896-a8fd9cb13e84."}]
```

If you see a 401 error while posting a notification and the UEM console is on-premises, then navigate to **System > Advanced > Secure Channel Certificate** and select the **Download CNS Secure Channel Certificate Installer**. You can also open a Zendesk ticket with the **SaasOps > CNS Upload Request** category. To install the certificate on the CNS server, send a request to the VMware Support team.

Error: ENS has posted notification to CNS/SNS successfully, but we don't see any notification on the device.

This error occurs due to the APNS or the GCM token issue. To verify the APNS or the GCM tokens, perform the following steps:

- 1 Log in to the Workspace ONE UEM console and navigate to the organization group where the device is enrolled.
- 2 Navigate to the **Devices > List View** and select the device.
- 3 Click the **SEND > PUSH NOTIFICATION** and select the application as Boxer from the drop-down.
- 4 Enter the **Message Body** and click **SEND**. After you click **SEND**, you must be able to see the notification on the device if the APNS token is correct.

Unregistered ENS Logs

The Boxer application sends an unregister request to the ENS in the following scenarios:

- When a device account is removed from the Boxer application
- When a device is deleted from the Workspace ONE UEM console.
- During an enterprise wipe from the Workspace ONE UEM console.
- Toggle off the push notification button in the Boxer application settings.

Sample of unregistered ENS logs:

```
2019/11/06 10:33:23.976 A3 2bd0af6a-ba08-479e-a606-b1326281902c [00000000-00000000] (53)
Debug MailNotificationService.Controllers.EnsController.Unregister User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Processing Unregister request.
UserId:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586]
2019/11/06 10:33:24.054 A3 2bd0af6a-ba08-479e-a606-b1326281902c [00000000-00000000] (55)
Debug MailNotificationService.BusinessImpl.UnregisterBusiness.ProcessUnregisterRequestAsync User
Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Device Unregistered for user:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586]
2019/11/06 10:33:24.054 A3 2bd0af6a-ba08-479e-a606-b1326281902c [00000000-00000000] (55)
Debug MailNotificationService.Controllers.EnsController.Unregister User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Unregister request processed.
HttpStatusCode:[OK] ResponseCode:[DeviceUnregistered]
```

When the ENS receives an unregister request, the ENS processes the request and sends an unsubscribe request to the Exchange and deletes the records from the database. The possible errors and solutions that you might see when you unregister is listed as follows:

Error: 401 error Unauthorized

The following logs are seen when the Boxer application sends an unregister request with a wrong API token. You can confirm the API token comparing the API token logged in the ENS logs and present in the Boxer application logs.

ENS logs: API token : [12341*****fasdf]

Boxer application logs: ensapitoken: 17413*****88c08

Sample of UnauthorizedRequest log:

```

2019/11/06 10:38:20.413 KAVINASH-W03 cd790dc0-ca7e-4f3d-b468-3c5181c34063 [0000000-0000000]
(31) Warn MailNotificationService.BusinessImpl.ApiKeyRepository.ValidateAsync User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] ApiKey header present [True],
Value Empty/Null: [False] API key dictionary has keys:[True] Key: [12341:fasdf]
2019/11/06 10:38:20.424 cd790dc0-ca7e-4f3d-b468-3c5181c34063 [0000000-0000000] (31) Debug
MailNotificationService.Controllers.EnsController.Unregister user Id
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] API token :
[12341*****fasdf]
2019/11/06 10:38:20.444 cd790dc0-ca7e-4f3d-b468-3c5181c34063 [0000000-0000000] (31) Warn
MailNotificationService.Controllers.EnsController.Unregister User Id:
[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Error Code:'23' Error message:
'UnauthorizedRequest'
Stack Trace: at MailNotificationService.Controllers.EnsController.<Unregister>d__21.MoveNext() in
C:\Stash\MailNotificationService\Controllers\EnsController.cs:line 926

```

Badge Update for ENS Logs

Badge update is only supported for iOS devices. The badge notification starts displaying only after the badge receives the first notification from ENS. The badge count is not seen in Boxer immediately after the badge counter is configured and subscribed.

Sample of badge update ENS logs:

```

2019/11/11 12:27:55.416 A3 04f06dcb-a721-4a90-a2ff-2be8e007f533 [0000000-0000000] (52)
Debug MailNotificationService.BusinessImpl.ExchangeNotificationParser.ScanEventNotificationAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Received [ModifiedEvent]
for subscription:
[EgBleGNoMjAxMy5tZW0xMy5vcmcQAAAAjo0Q0qTL7hk2FF7QvXOHC1BLv0sChZtcIEAAAACanmwmX5x50pwfUW+dfdrQ=]
2019/11/11 12:27:55.525 A3 04f06dcb-a721-4a90-a2ff-2be8e007f533 [0000000-0000000] (52)
Info MailNotificationService.BusinessImpl.PushNotificationBusiness.HandleMoveModifiedEventAsync
User Id:[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] -BADGE UPDATE- [5422]
previous BADGE count is [5422] Received modified event for user
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586]
2019/11/11 12:27:55.525 A3 04f06dcb-a721-4a90-a2ff-2be8e007f533 [0000000-0000000] (52)
Info MailNotificationService.BusinessImpl.NotificationsProcessor.AddNotificationToBatch User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] About to Post Notification for
user : [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] and Device Id : [9]
2019/11/11 12:28:00.531 A3 d7893c08-3a06-46a1-a8a7-45361572b573 [0000000-0000000] (16)
Info MailNotificationService.BusinessImpl.CNSHelper.ComposeAPNSPushNotification User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Total unread count retrieved
[5422] for user [20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586]
2019/11/11 12:28:00.531 A3 d7893c08-3a06-46a1-a8a7-45361572b573 [0000000-0000000] (16)
Debug MailNotificationService.BusinessImpl.CNSHelper.ComposeAPNSPushNotification User Id:
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586] Sending to :: User :
[20943ad3f74ef04b3a2394b968cb46cc498f54994bdec0b3520d965e35356586], DeviceId : [9], DeviceLogId : [],
Message : messageId: []
2019/11/11 12:28:00.531 A3 d7893c08-3a06-46a1-a8a7-45361572b573 [0000000-0000000] (16)
Debug MailNotificationService.BusinessImpl.CNSHelper.CreateWebRequest User Id:[no-user-id] CNS
Url : [https://cns.awmdm.com/nws/notify/apns]
2019/11/11 12:28:00.531 A3 d7893c08-3a06-46a1-a8a7-45361572b573 [0000000-0000000] (16)
Debug MailNotificationService.BusinessImpl.CertificateHelper.ComputeCmsSignature User Id:[no-user-
id] Signing URL [/nws/notify/apns] with Cert [CN=AW Cloud Notification - aTest]

```

```

2019/11/11 12:28:00.748 A3 d7893c08-3a06-46a1-a8a7-45361572b573 [00000000-00000000] (16)
Debug MailNotificationService.BusinessImpl.CNSHelper.ReadResponse User Id:[no-user-id] Response
{"status":"success","errorReason":null}
2019/11/11 12:28:00.748 A3 d7893c08-3a06-46a1-a8a7-45361572b573 [00000000-00000000] (16)
Info MailNotificationService.BusinessImpl.CNSHelper.ReadResponse User Id:[no-user-id]
ResponseCode OK

```

Understanding ENS Logs

The ENS logs contain information about registration, subscriptions, notifications, and the CNS or the APNS delivery status. For the on-premises ENS, you can find the ENS2 logs files at: %ENS Installed Directory%\Logs\Email Notification Service. For example, the ENS2 log file can be at: C:\AirWatch\Logs\Email Notification Service. The name of the log file is *ENS.log*.

Sample ENS2 log file:

```

2019/11/05 09:39:56.608 A3 9f08ed6d-0726-430c-8440-9c396443c7ca [00000000-00000000] (74) Debug

MailNotificationService.BusinessImpl.ExchangeNotificationParser.ScanEventNotificationAsync

User Id:[1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Received [CreatedEvent]
for subscription:
[JwBtbjJwcjE5bWIZMDA1Lm5hbXBzZDE5LnByb2Qub3V0bG9vay5jb20QAAAA14H5dKboFUm1kJ8ZNBKkJILRTBjMYdcIEAAAAA9t
cFCKSZFrT0xLbSCwj4=]

```

The following table provides a description of a sample ENS2 log file.

Log Format	Value
Date	2019/11/05 09:39:56.608. The date is mentioned in the UTC format.
machinename	A3
ActivityId	9f08ed6d-0726-430c-8440-9c396443c7ca
threadid	(74)
logLevel	Debug

Log Format	Value
Logger	MailNotificationService.BusinessImpl.ExchangeNotificationParser.ScanEventNotificationAsync
Message	UserId: [1743604ea20cda831dc7aea285e7fdc011ca233caf0fa7d5d926916622dd182d] Received [CreatedEvent] for subscription: [JwBtbjJwcjE5bWizMDA1Lm5hbXBzZDE5LnByb2Qub3V0bG9vay5jb20QAAAAI4H5dKboFUm1kJ8ZNBKKJILRTBJMYdcIEAAAAAQ9tcFCKSZFrTOxLbSCwj4=]

Note In the logs, you can find the user name or email address in the alphanumeric format and not in the plain text format. For example, the user ID is mentioned as an alphanumeric string such as, 4e9dc715faba719b266fe90f866caf8e377c08984cd1fd005bac72c7eba4db02. This string is a hash value that is calculated from the email address.

You can use the [SHA-256 hash calculator](#) to translate any email address to a hash value. You can then use the hash value to search logs for any user.

To obtain the logs for the cloud ENS, use the ENS2 Self-Help website based on your region.

- For the US region - <https://enshelp.getboxer.com>
- For the EU region - <https://enshelp-eu.getboxer.com>
- For the APJ region - <https://enshelp-apj.getboxer.com>
- For the UK region - <https://enshelp-uk.getboxer.com>

Troubleshooting the ENS2 SEG Errors

This section describes the troubleshooting steps you might have to perform due to communication errors between the ENS2 and Exchange with SEGv2 as the proxy.

The following steps describe the interaction between the ENS2 and Exchange with SEGv2 as the proxy.

- 1 Boxer application requests a public key from the ENS.
- 2 Boxer application encrypts the user credentials using the public key and sends a subscription request to the ENS.
- 3 ENS requests a subscription to the Exchange server using the SEG URL which also contains the encrypted credentials. The ENS also sends a client certificate. If the client certificate is configured on the Boxer application profile, then the authentication received from the Boxer profile is sent. For certificate-based authentication (CBA), when a register device request is sent to the cloud ENS server, the ENS routes the request to the SEG with the certificate information. The SEG follows the same token retrieval process similar to the ActiveSync request.
- 4 SEG forwards the subscription request to the Exchange to complete the subscription. The same authentication method configured in the Boxer application profile is used for subscription. The ENS server callback URL is used to subscribe.
- 5 The Exchange server receives an email.

- 6 The Exchange server notifies the ENS callback URL of the subscriber to inform that a new email has arrived, hence update the email client with the notification. The ENS fetches the details of the email from the SEG.
- 7 The ENS server requests the CNS or SNS to send notification to the Boxer application or the device of the subscriber.
- 8 The CNS or the SNS server contacts the Apple Push Notifications (APNs for iOS devices) or GCM or FCM (for Android devices).
- 9 The APNS or GCM server pushes the email notification to the device.

Using the transaction ID received in the **ews-transaction log**, you can search the **ews-proxy.log**. For example, if the transaction ID is **544ef2b7-9ca3-4009-b116-8a9f6513f2c7** then search for **544ef2b7**.

When you see 200 in the ENS transaction log, you can confirm if the notifications are going through the CNS communication.

The **Ews-transaction log** sample.

```
Time, LogLevel, Thread Id, Message, HTTP-Method, Remote-Host, X-Forwarded-For, SEG TransactionId,
Request-DeviceId, EnsDevices, EmailServerResponseStatus, SegResponseStatus, EmailRequestBodySize,
EmailResponseBodySize, TimeTakenByKerberosService(ms), TimeTakenBySeg(ms),
TimeTakenByEmailServer(ms), BeginningOfRequest
2018-12-03 17:20:15.696, DEBUG, (vert.x-eventloop-thread-0), Responding back to
ENS, POST, 192.168.2.34, null, 544ef2b7-9ca3-4009-
b116-8a9f6513f2c7, 6C30D0304E7A4EE795494DEB0F465B72, "6C30D0304E7A4EE795494DEB0F465B72:200", 200, 200, 1243
, 1147, 2547, 0, 16, 1543875613133
2018-12-03 17:20:18.274, DEBUG, (vert.x-eventloop-thread-0), Responding back to
ENS, POST, 192.168.2.34, null, d77ae46b-2d38-46b5-9548-3bcc25a1bf03, 6C30D0304E7A4EE795494DEB0F465B72, "6C30
D0304E7A4EE795494DEB0F465B72:200", 200, 200, 673, 1806, 2547, 0, 16, 1543875615711
2018-12-03 17:20:41.430, DEBUG, (vert.x-eventloop-thread-0), Responding back to
ENS, POST, 192.168.2.34, null, b639bdee-0cfa-42b5-82ea-0629ab1d586a, 6C30D0304E7A4EE795494DEB0F465B72, "6C30
D0304E7A4EE795494DEB0F465B72:200", 200, 200, 1632, 2464, 2562, 0, 47, 1543875638821
2018-12-03 17:21:16.462, DEBUG, (vert.x-eventloop-thread-1), Responding back to
ENS, POST, 192.168.2.34, null, ed0db6c1-9dd4-420a-83d3-
e746cb17445c, 82B15D853CC14CA3989020257158BFC1, "82B15D853CC14CA3989020257158BFC1:200", 200, 200, 1632, 3028
, 2563, 0, 47, 1543875673852
2018-12-03 17:21:26.493, DEBUG, (vert.x-eventloop-thread-1), Responding back to
ENS, POST, 192.168.2.34, null, 425fc495-4ae1-4c26-abc5-
c30f34a376cf, 82B15D853CC14CA3989020257158BFC1, "82B15D853CC14CA3989020257158BFC1:200", 200, 200, 673, 1815,
2547, 16, 15, 1543875683915
2018-12-03 17:22:46.649, DEBUG, (vert.x-eventloop-thread-1), Responding back to
ENS, POST, 192.168.2.34, null, ba0b13ad-b341-43e3-a4a9-
d1a79c5330e0, 82B15D853CC14CA3989020257158BFC1, "82B15D853CC14CA3989020257158BFC1:200", 200, 200, 1632, 3028
, 2547, 15, 32, 1543875764055
2018-12-03 17:23:01.649, DEBUG, (vert.x-eventloop-thread-1), Responding back to
ENS, POST, 192.168.2.34, null, 262cc2b2-8ae4-4ea7-b062-
da2b2eb42a68, 82B15D853CC14CA3989020257158BFC1, "82B15D853CC14CA3989020257158BFC1:200", 200, 200, 673, 1815,
2547, 0, 15, 1543875779087
2018-12-03 17:26:47.353, DEBUG, (vert.x-eventloop-thread-3), Responding back to
ENS, POST, 192.168.2.34, null, c7e5a6c9-
b1b0-4739-9132-49470306882c, 6C30D0304E7A4EE795494DEB0F465B72, "6C30D0304E7A4EE795494DEB0F465B72:200", 20
0, 200, 673, 1806, 2547, 0, 94, 1543876004712
2018-12-03 17:26:51.884, DEBUG, (vert.x-eventloop-thread-3), Responding back to
```

```

ENS,POST,192.168.2.34,null,d5cf2470-d818-45f6-ab0e-
dd68599d4aa8,6C30D0304E7A4EE795494DEB0F465B72,"6C30D0304E7A4EE795494DEB0F465B72:200",200,200,673,1806,
2547,0,15,1543876009322
2018-12-03 22:06:55.421, DEBUG, (vert.x-eventloop-thread-2), Responding back to
ENS,POST,192.168.2.34,null,93f7f097-bda5-417a-
ac67-5667b4088c84,6C30D0304E7A4EE795494DEB0F465B72,"6C30D0304E7A4EE795494DEB0F465B72:200",200,200,673,
1806,12000,16,234,1543892803171
2018-12-03 22:07:00.031, DEBUG, (vert.x-eventloop-thread-0), Responding back to
ENS,POST,192.168.2.34,null,d10c08a4-49cd-4240-bcc0-
ba9bb81f74f0,82B15D853CC14CA3989020257158BFC1,"82B15D853CC14CA3989020257158BFC1:200",200,200,673,1815,
11969,0,188,1543892807874
2018-12-04 10:31:33.786, DEBUG, (vert.x-eventloop-thread-2), Responding back to
ENS,POST,192.168.2.34,null,3844719b-73c6-4b77-91d8-8a7d8b9a97c0,82B15D853CC14CA3989020257158BFC1,"82B1
5D853CC14CA3989020257158BFC1:200",200,200,1632,3028,2563,15,516,1543937490692

```

The **Ews-transaction log** sample filtered using the **544ef2b7**.

```

2018-12-03 17:20:13.133 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsRequestReadHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - Incoming EWS request, Path: /EWS/Exchange.asmx. Headers are
2018-12-03 17:20:13.133 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsHelper] - 544ef2b7-9ca3-4009-
b116-8a9f6513f2c7 - Collected ENS devices: [6C30D0304E7A4EE795494DEB0F465B72]
2018-12-03 17:20:13.133 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsRequestReadHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - Getting device policy for request device
6C30D0304E7A4EE795494DEB0F465B72
2018-12-03 17:20:13.133 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsComplianceCheckHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - Device list: [6C30D0304E7A4EE795494DEB0F465B72]
2018-12-03 17:20:13.133 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsComplianceCheckHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - Checking compliance for device 6C30D0304E7A4EE795494DEB0F465B72
2018-12-03 17:20:13.133 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsComplianceCheckHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - Device 6C30D0304E7A4EE795494DEB0F465B72 is compliant
2018-12-03 17:20:13.133 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsRequestProxyHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 KCD authentication is (true), upn is TUSER1@MILKYWAY.LOCAL.
2018-12-03 17:20:15.680 DEBUG (pool-7-thread-5) [c.a.s.e.h.EwsRequestProxyHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - Successfully got kerberos token for UPN TUSER1@MILKYWAY.LOCAL
- token length 2024
2018-12-03 17:20:15.680 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsRequestProxyHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - Proxying request to EWS
2018-12-03 17:20:15.680 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsRequestProxyHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - EWS client request headers:
2018-12-03 17:20:15.696 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsRequestProxyHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - EWS client response headers:
2018-12-03 17:20:15.696 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsHelper] - 544ef2b7-9ca3-4009-
b116-8a9f6513f2c7 - Response headers from SEG to ENS:
X-AW-SEG-TRANSACTION-ID : 544ef2b7-9ca3-4009-b116-8a9f6513f2c7
2018-12-03 17:20:15.696 DEBUG (vert.x-eventloop-thread-0) [c.a.s.e.h.EwsRequestProxyHandler] -
544ef2b7-9ca3-4009-b116-8a9f6513f2c7 - EWS response status 200, length 1147

```

The possible errors and solutions you might see during an interaction between the ENS2 and Exchange with SEGv2 as the proxy is listed as follows:

Error: 404 / https://[segURL]/EWS/Exchange.asmx is not found

If you see this error in the ENS logs, then ensure you have enabled the EWS proxy in the SEG server. If you have not enabled the EWS proxy in the SEG server then perform the following steps.

- 1 Navigate to the **SEG > Config** folder using the File explorer.
- 2 Select the **application.properties** file and edit the file.
- 3 Select the **enable.boxer.ens.ews.proxy** value and update the value to **enable.boxer.ens.ews.proxy=true**.
- 4 Save the file.
- 5 Restart the **VMware AirWatch Secure Email Gateway** service.

Sample of the **application.properties** file.

```
#####
#####
##### Start - HTTP endpoint path for SEG active-sync, syncML and REST API.
#####
#####
#####
# SEG HTTP server context path. This should be same as the context path of Email/Exchange server as
Device won't know
# if it's sending request to email server or SEG Proxy. This value generally don't change but we want
to give
# the ability to the Admin to change it, if needed in some exceptional cases.
# Right now Vertx doesn't support "ignore-case" on path, and also doesn't allow mounting sub-routers
on RegEx.
# For now we're trying to avoid using RegEx anyway - https://groups.google.com/forum/#!topic/vertx/
ck95b4juj4A
activesync.context.paths=/Microsoft-Server-ActiveSync,/microsoft-server-activesync# Context path when
SEG works as EWS proxy for ENS. EWS endpoint will be disabled by default.
enable.boxer.ens.ews.proxy=true
ews.proxy.context.paths=/EWS,/ews

# Flag used to remove unsupported www-authenticate header such as NTLM and Negotiate (in absense of
certificate) from EWS response to ENS.

remove.unsupported.auth.for.ews=true
```

Error: 401 - Please check the authentication type enabled in exchange (EWS endpoint)

If you see this error in the ENS logs, then the SEGV2 does not support the NTLM authentication. If both the Basic and NTLM authentication mechanisms are enabled for the EWS endpoint, then the SEGV2 version prior to version 2.9.0.1 cannot prefer Basic authentication over the unsupported NTLM authentication.

This results in the ENS attempting the NTLM-based authentication for requests through the SEG, that eventually causes 401 error responses as observed in the **ews-transaction.log**. If the user is unable to disable the NTLM authentication mechanism for the EWS endpoint, and is using any lower version of the SEG, then setup the KCD authentication for the ENS-SEG integration to work correctly.

If you connect directly to the EWS endpoint on the SEGv2 proxy through the `https://[segURL]/EWS/Exchange.asmx` URL, you might receive a 400 error message unless you connect using a permitted device.

Error: The request was aborted: Could not create SSL/TLS secure channel

In the ENS logs, if you see the following error during the registration process, then the error might be due to a cipher mismatch.

```
2019-12-05 15:33:40.5081|DEBUG|
MailNotificationService.BusinessImpl.ExchangeRetriesHandler.SubscribeForNotificationsAsync|
3ed2219d-42f2-4a2a-b857-ab7639ad1858|User Id:
[af03aa8bb3cae692442ec673b207f5666e0762bf3ca62cbaaa61c4208cd7bd] EWSUrl used to subscribe: [https://
uag.testdomain.com/ews/exchange.asmx]
2019-12-05 15:33:40.5550|WARN|
MailNotificationService.BusinessImpl.SubscriptionBusiness.SubscribeV2Async|3ed2219d-42f2-4a2a-b857-
ab7639ad1858|User Id:[af03aa8bb3cae692442ec673b207f5666e0762bf3ca62cbaaa61c4208cd7bd] Service
request exception occurred for userId
[af03aa8bb3cae692442ec673b207f5666e0762bf3ca62cbaaa61c4208cd7bd], Inner exception message [The
request was aborted: Could not create SSL/TLS secure channel.].
```

To fix the cipher mismatch error, perform the following steps:

- 1 Run a TCP dump on the UAG or SEG. Check the reason for the handshake failure, using the following commands. See the *Troubleshooting Firewall and Connection Issues* section in the *Deploying and Configuring VMware Unified Access Gateway* guide.

```
/etc/vmware/gss-support/install.sh
```

```
tcpdump -i any -n -v tcp port any -w /tmp/vmware/capture.pcap
```

- 2 Open the TCP dump logs using the Wireshark or any supported application. Filter the logs based on the IP source and IP destination and check for the client hello request as shown in the following log.

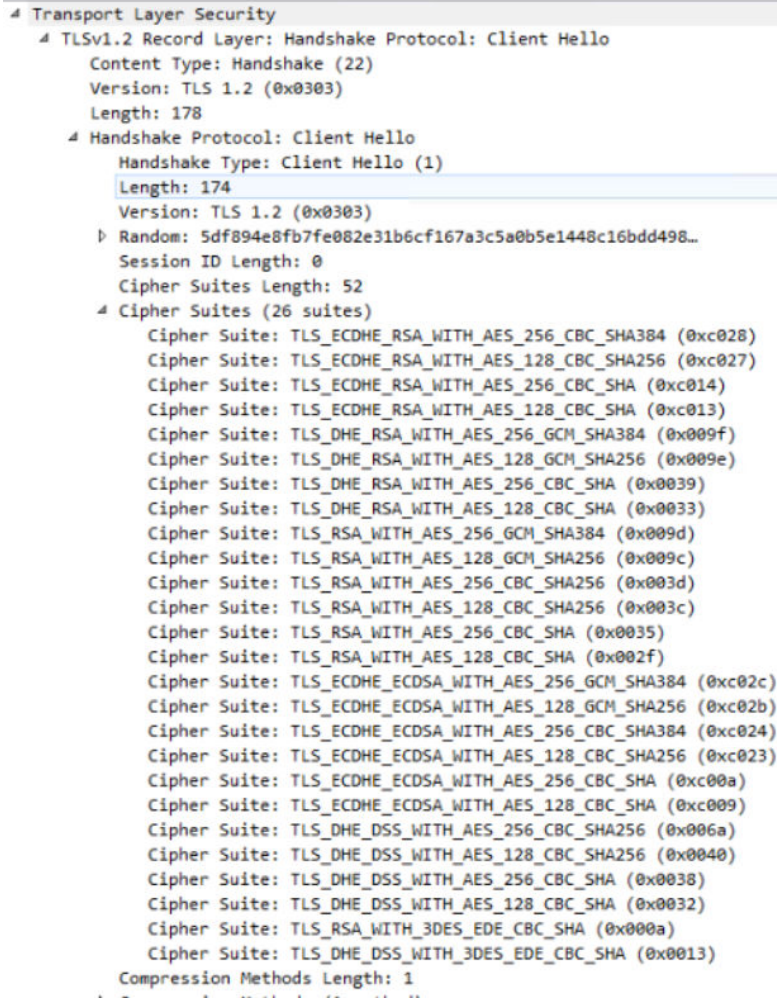
Use the `tls.alert_message.level` filter to search for the SSL error or alert in the Wireshark. Identify the source and destination IP, right click, and select **Follow > Follow → TLS stream**.

Time	Source	Destination	Protocol	Length	Info
20:01:27.497557	127.0.0.1	127.0.0.1	TL		Mark/Unmark Packet Ctrl+M Level: Fatal, Description: Handshake Failure)
20:01:27.497601	172.27.0.87	172.27.0.253	TL		Ignore/Unignore Packet Ctrl+D Level: Fatal, Description: Handshake Failure)
20:01:36.421920	127.0.0.1	127.0.0.1	TL		Set/Unset Time Reference Ctrl+T Level: Fatal, Description: Handshake Failure)
20:01:36.421965	172.27.0.87	172.27.0.253	TL		Time Shift... Ctrl+Shift+T Level: Fatal, Description: Handshake Failure)
20:01:39.856021	127.0.0.1	127.0.0.1	TL		Packet Comment... Ctrl+Alt+C Level: Fatal, Description: Handshake Failure)
20:01:39.856079	172.27.0.87	172.27.0.253	TL		Edit Resolved Name Level: Fatal, Description: Handshake Failure)
20:01:44.484228	127.0.0.1	127.0.0.1	TL		Apply as Filter Level: Fatal, Description: Handshake Failure)
20:01:44.484291	172.27.0.87	172.27.0.253	TL		Prepare a Filter Level: Fatal, Description: Handshake Failure)
20:01:45.912331	127.0.0.1	127.0.0.1	TL		Conversation Filter Level: Fatal, Description: Handshake Failure)
20:01:45.912372	172.27.0.87	172.27.0.253	TL		Colorize Conversation Level: Fatal, Description: Handshake Failure)
20:01:55.657406	127.0.0.1	127.0.0.1	TL		Follow TCP Stream Ctrl+Alt+Shift+T Level: Fatal, Description: Handshake Failure)
20:01:55.657453	172.27.0.87	172.27.0.253	TL		Copy UDP Stream Ctrl+Alt+Shift+U Level: Fatal, Description: Handshake Failure)
					TLS Stream Ctrl+Alt+Shift+S Level: Fatal, Description: Handshake Failure)
					Protocol Preferences HTTP Stream Ctrl+Alt+Shift+H
					Decode As...
					Show Packet in New Window

Time	Source	Destination	Protocol	Length	Info
20:01:27.496707	127.0.0.1	127.0.0.1	TCP	68	3790 → 11443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_
20:01:27.496720	127.0.0.1	127.0.0.1	TCP	68	11443 → 3790 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS
20:01:27.496731	127.0.0.1	127.0.0.1	TCP	56	3790 → 11443 [ACK] Seq=1 Ack=1 Win=29696 Len=0
20:01:27.496786	127.0.0.1	127.0.0.1	TL		TLSv1.2 2... Client Hello
20:01:27.496792	127.0.0.1	127.0.0.1	TCP	56	11443 → 3790 [ACK] Seq=1 Ack=186 Win=30720 Len=0
20:01:27.497557	127.0.0.1	127.0.0.1	TL		TLSv1.2 63 Alert (Level: Fatal, Description: Handshake Failure)
20:01:27.497576	127.0.0.1	127.0.0.1	TCP	56	3790 → 11443 [ACK] Seq=186 Ack=8 Win=29696 Len=0
20:01:27.497645	127.0.0.1	127.0.0.1	TCP	56	11443 → 3790 [FIN, ACK] Seq=8 Ack=186 Win=30720 Len=0
20:01:27.515820	127.0.0.1	127.0.0.1	TCP	56	3790 → 11443 [FIN, ACK] Seq=186 Ack=9 Win=29696 Len=0
20:01:27.515838	127.0.0.1	127.0.0.1	TCP	56	11443 → 3790 [ACK] Seq=9 Ack=187 Win=30720 Len=0

3 Right click and open the **Client Hello** information.


- 4 Click the **Show packet > TLS 1.2 Record Layer > Handshake Protocol : Client Hello > Transport Layer Security > Cipher Suites**. You can see a list of cipher suites that the client ENS is sending to initiate a secure communication as shown in the following image.



- 5 Ensure that the UAG or the SEG server has enabled the ciphers listed in the **Client hello Request**.

Note To check for the enabled cipher suites in the UAG or the SEG server, you can use the [SSL report](#). Enter your SEG or UAG URL and wait for the test to complete. When the test is complete, you might see the following result.


Configuration



Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	No
SSL 3	No
SSL 2	No

For TLS 1.3 tests, we only support RFC 8446.



Cipher Suites

TLS 1.2 (server has no preference)

TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH secp521r1 (eq. 15360 bits RSA) FS WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) WEAK	128
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) WEAK	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp521r1 (eq. 15360 bits RSA) FS WEAK	128

The following table lists all the response codes and messages in the SEG logs.

Response Code	Message	Description
204	No Content	Indicates that the policy data is not loaded in the SEG to run the compliance check on the requesting devices.
403	Forbidden	Indicates that none of the devices listed in the ENS request headers are compliant.
400	Bad Request	Indicates that none of the devices listed in the ENS request header are found in the SEG device policy cache.
5xx		Indicates the server errors.

Troubleshooting Connection Issues to the ENS Database

When installing ENS, use the SQL authentication and not the Windows authentication to access the ENS database. This topic is applicable for ENS on-premise installation only.

Problem

In case you connect to the ENS database using the Windows authentication then you might receive the following error:

```
2018/11/05 19:55:40.800 EUROPA 800000005-0001-ff00-b63f-84710c7967bb [00000000-00000000] (35)
Error MailNotificationService.ProviderImpl.ApiTokensDataHandler.ApiTokensAsync User Id:[ ] Error
While loading the api tokens Exception [Cannot open database "ENS" requested by the login. The login
failed.
Login failed for user 'NT AUTHORITY\LOCAL SERVICE'.] StackTrace[ at
System.Data.SqlClient.SqlInternalConnectionTds..ctor(DbConnectionPoolIdentity identity,
```



```
SqlConnection connectionOptions, SqlCredential credential, Object providerInfo, String
newPassword, SecureString newSecurePassword, Boolean redirectedUserInstance, SqlConnectionString
userConnectionOptions, SessionData reconnectSessionData, DbConnectionPool pool, String accessToken,
Boolean applyTransientFaultHandling)
```

Cause

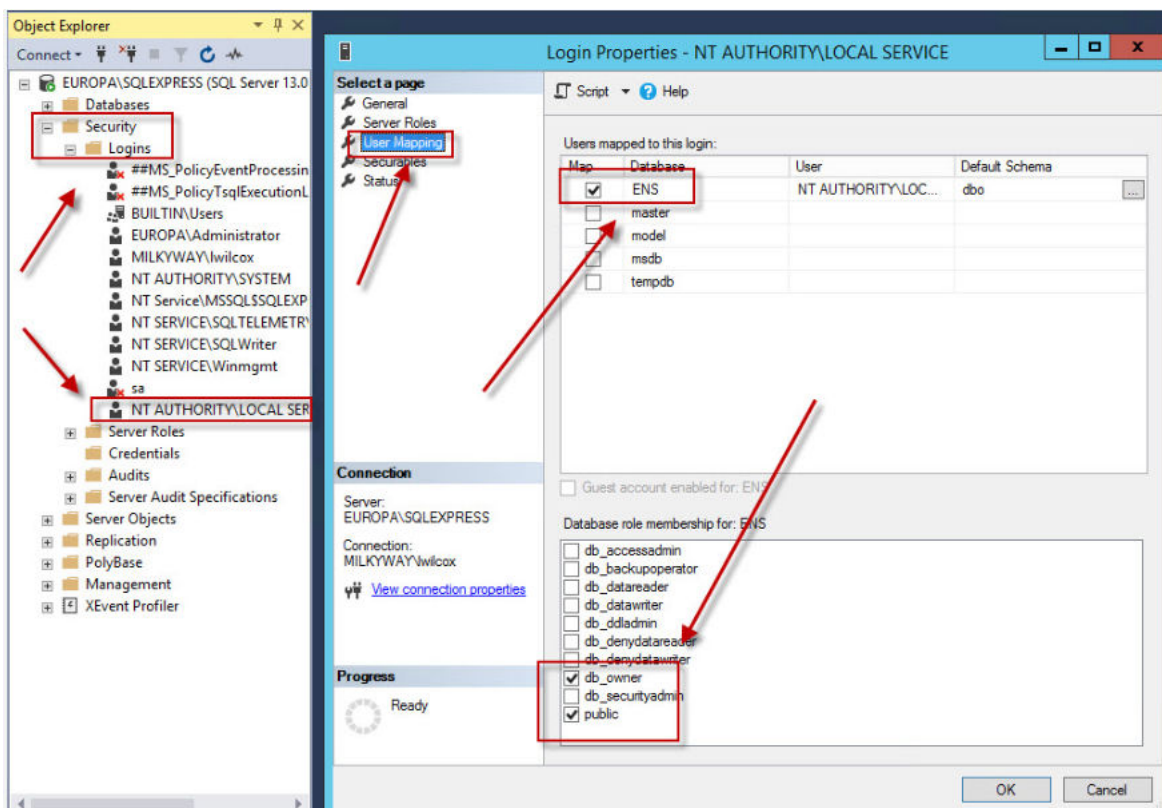
Connecting to the ENS database using the Windows authentication might cause this issue.

Use the SQL authentication to connect to your ENS database. In the solutions steps provided, the **NT AUTHORITY\LOCAL SERVICE** is the name of the user account and the database role membership for the **NT AUTHORITY\LOCAL SERVICE** account must have the **db_owner** and **public** enabled.

To add an SQL account to the ENS database, perform the following steps:

Solution

- 1 Open the SQL Server Management Studio.
- 2 Navigate to **Security > Logins** and add NT AUTHORITY\LOCAL SERVICE.
- 3 Navigate to **Security > Logins > NT AUTHORITY\LOCAL SERVICE > User Mapping > .**



- 4 Select the ENS database and add the required permissions.

Troubleshooting SSL Errors

Use the SSL pinning certificate tool when the notifications are not delivered to the devices. This tool is only used for troubleshooting purpose and is not a mandatory step during installation. The following error

message appears in the ENS logs while posting the notifications to CNS: **The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel.**

The following procedure describes the steps to upload the SSL pinning certificate to the ENS.

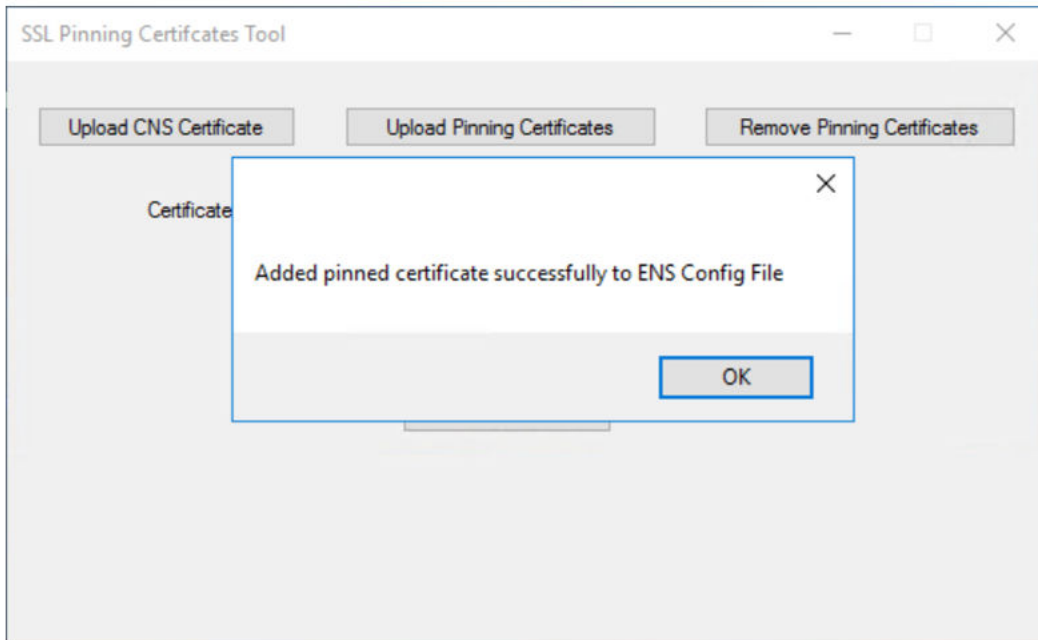
Prerequisites

Download the latest certificate from: <http://resources.workspaceone.com/view/2hjxzvgkxyf8n738hy7x/en>.

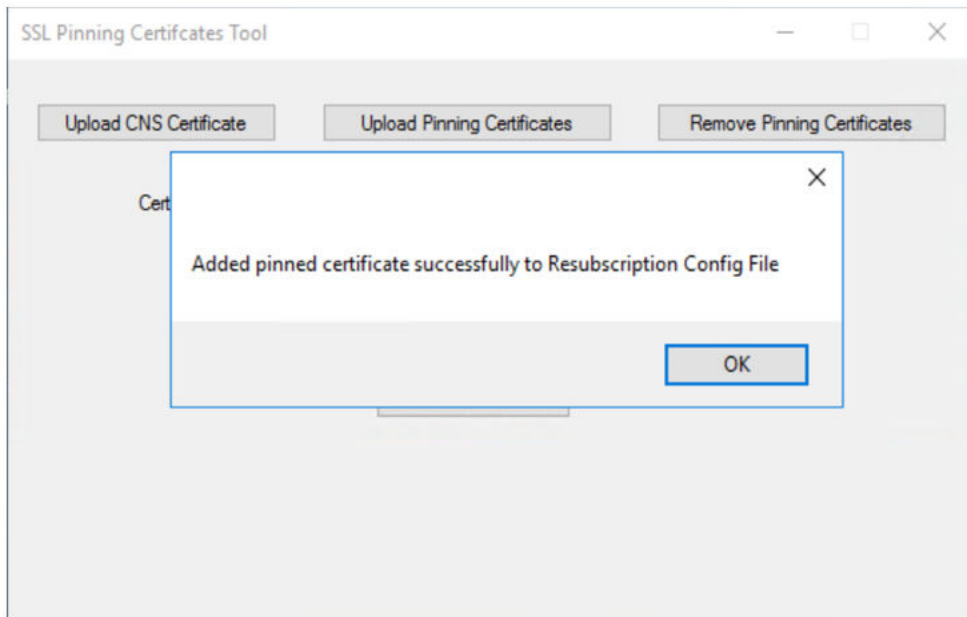
Procedure

- 1 Click the **SSLPinningCertTool** shortcut on the ENS server, or navigate to the <ENS_INSTALL_DIR>\Email Notification Service\Tools\SSLPinningCertTool\SSLPinningCertTool.exe file.
- 2 Click the **Upload CNS Certificate** button.
- 3 Select the certificate to be uploaded and click **Submit**.

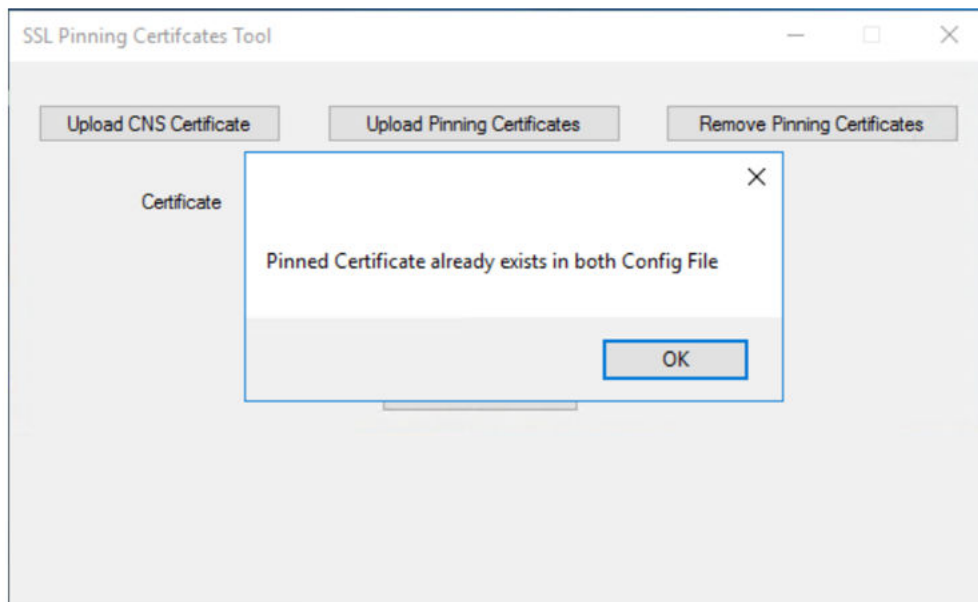
If the following screen appears, then the certificate is successfully added to the ENS configuration file. Click **OK** to continue. After uploading the SSL pinning certificate on the ENS, the tool adds the public key of the certificate to the ENS configuration file. When the ENS posts payload to the CNS, the certificate validation is done against the newly added certificate public key.



If the following screen appears, then the certificate is added successfully to the resubscription configuration file. After uploading the SSL pinning certificate, the tool adds the public key of the certificate to the resubscription configuration file. When the resubscription mechanism posts payload to the CNS, the certificate validation is done against the newly added certificate public key.



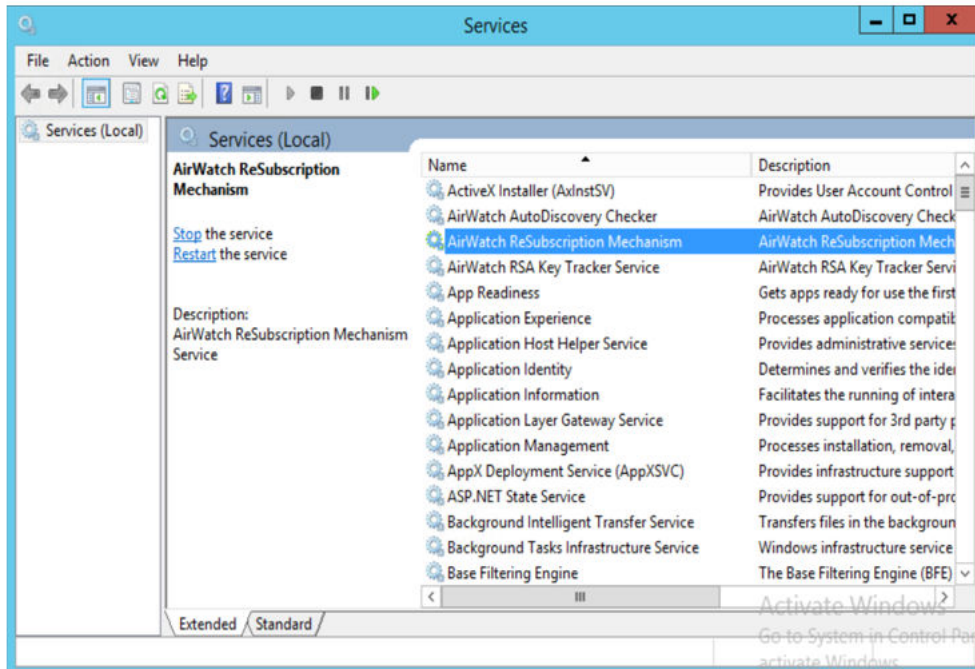
If the certificate is already present in both the configuration files, then the following prompt message appears:



The upload pinning certificate process works as follows:

- The SSL pinning certificate tool tries to upload the certificate to the ENS configuration file only if the provided certificate is not present in the ENS configuration file. If the given certificate is already present, then the tool does not prompt any message and continues to upload the same certificate to the resubscription configuration file.

- The SSL pinning certificate tool tries to add the certificate to the resubscription configuration file only if the provided certificate is not present in the resubscription configuration file. If the given certificate is present, then the tool does not prompt any message to user.
- If the certificate is added to the resubscription configuration file, then restart the **AirWatch Resubscription Mechanism** service in the **Services** tab.



ENS2 Response Code and Error Code Details

If the Boxer application sends a request to the ENS, the ENS processes the request and sends a response with a response code and message to the Boxer application.

The following table lists all the response codes and messages in the Boxer application logs and the ENS logs.

Response Code	Message	Description
14	SubscribeAgain	If a subscription failed, then the ENS sends a subscribe again message to the Boxer application.
8	ErrorSubscribeOrUpdateDb	When you try to add a user or device details to the database during subscription, you might receive this error .
23	UnauthorizedRequest	Authentication failed (API token mismatch) for the request from the Boxer application.
32	Failed (Handled Exception)	Registration failed with a handled exception. For example, the URL is not in the correct format.

Response Code	Message	Description
17	Success	Indicates that the registration is successful.
3	UpdateSuccess	<p>The Boxer application receives this response when the:</p> <ul style="list-style-type: none"> ■ Get the Public Key request is successful and the database is updated accordingly. ■ The synchronization key for the user success and the database is updated accordingly. ■ Any device details updated in database, such as, update device token is successful.
4	UpdateFail	ENS sends this response for multiple reasons. When you receive this response, verify the corresponding ENS logs and troubleshoot based on the message in the logs.
5	TokenDoesNotExist	<p>ENS sends this response when the device record is absent.</p> <p>When you send a force register, (by changing the notification sound in the Boxer application setting) a new device is created on the ENS server.</p>
6	UserAlreadySubscribed	ENS sends this response when a user is already subscribed on the ENS server.
7	UserSubscribed	User subscription is successful.
9	NoRecordExists	<p>ENS sends this response when a user record is absent.</p> <p>When you send a force register, (by changing the notification sound in the Boxer application setting) a new user is created on the ENS server.</p>
15	UserSubscribedNotUpdatedInDb	User subscribed but failed to add device details in the database. In this case, ensure that the connection from the ENS to the database is working correctly or the user has permission to update the database.
16	FailedToGetEwsUrlFromAutoDiscovery	Unable to determine the Exchange version after autodiscovery.
21	EmailFetchfailed	Fetching email information from the EWS failed.
24	DeviceUnregisteredUserUnsubscribed	Unsubscribe successful and the user is unregistered.

Response Code	Message	Description
25	DeviceUnregistered	The device is deleted from the database.
26	DeviceNotRegistered	The device is not registered.
28	UserSubscriptionNotFound	User record does not exist.
29	UserRecordPresentNotSubscribed	User records are present but not subscribed.
30	SubscribedNeedsUpdate	User has already subscribed and must be added to the database.
34	InvalidDecryptedPayload	Payload is encrypted with a wrong public key.
35	EWSUrlMismatch	Unsubscribing with the wrong EWS URL. The EWSUrl for the register request and Exchange service is not the same.
36	InvalidAuthType	Indicates the invalid authentication type.

Troubleshooting ENS with On-Premise Exchange Server

This section describes the steps to confirm that the client certificate is available at the Exchange endpoint. When a customer is using ENS with on-premise Exchange and certificate-based authentication (CBA) enabled, you must confirm that the client certificate is appearing at the Exchange endpoint. Traffic might reach the Exchange CAPI2 logs before the traffic reaches the Exchange IIS.

Cause: The CAPI2 logs are part of the Event Viewer. You can check the CIAP2 logs to confirm whether the client certificate appeared on the Exchange server. The devices are set up to check the client certificate for any SSL sessions for security issue. In this case, the client certificate must authenticate the user.

- 1 Enable CAPI2 logs in the Event Viewer.
- 2 Confirm that the client certificate is appearing correctly.

In the Event Viewer, ensure the extended usage has client authentication listed, check if the root certificate and certificate chain are valid.

Note The following images are for example only, the actual information might be different.

Click on the log line and open up the certificate details as shown in the following image. Ensure the **ExtendedKeyUsage** has **Client Authentication** listed.

```

+ System
- UserData
  - CertGetCertificateChain
    - Certificate
      [ fileRef]      19EC88F3D2BB9FA8E18F0A49CB0322A803FAB4C1.cer
      [ subjectName] lwilcox
    - AdditionalStore
      - Certificate
        [ fileRef]      3640B5724969FDAE60D138DCD4D774593537E5FA.cer
        [ subjectName] milkyway-SUN-CA
      - Certificate
        [ fileRef]      19EC88F3D2BB9FA8E18F0A49CB0322A803FAB4C1.cer
        [ subjectName] lwilcox
    - ExtendedKeyUsage
      - Usage
        [ oid]          1.3.6.1.5.5.7.3.2
        [ name]          Client Authentication

```

Check if the root certificate and the certificate chain is valid.

```

+ System
- UserData
  - X509Objects
    - Certificate
      [ fileRef]      3640B5724969FDAE60D138DCD4D774593537E5FA.cer
      [ subjectName] milkyway-SUN-CA
    - Subject
      CN              milkyway-SUN-CA
      DC              milkyway
      DC              local
    - SubjectKeyID
      [ computed]     false
      [ hash]         12E7AB1A336F2B676498FD2AFE98A0A0EAA517E6
    - SignatureAlgorithm
      [ oid]           1.2.840.113549.1.1.5
      [ hashName]      SHA1
      [ publicKeyName] RSA
    - PublicKeyAlgorithm
      [ oid]           1.2.840.113549.1.1.1
      [ publicKeyName] RSA

```

Ensure the certificate is not revoked.

```

+ System
- UserData
  - CertVerifyRevocation
    - Certificate
      [ fileRef]      19EC88F3D2BB9FA8E18F0A49CB0322A803FAB4C1.cer
      [ subjectName] lwilcox
    - IssuerCertificate
      [ fileRef]      3640B5724969FDAE60D138DCD4D774593537E5FA.cer
      [ subjectName] milkyway-SUN-CA
    - Flags
      [ value]        0
    - AdditionalParameters
      [ timeToUse]     2018-03-14T16:53:04.339Z
      [ currentTime]  2018-03-14T16:53:04.339Z
      [ urlRetrievalTimeout] PT15S
    - RevocationStatus
      [ index]         0
      [ error]         0
      [ reason]        0
      [ actualFreshnessTime] PT1H33M22S
      [ thirdPartyProviderUsed] C:\Windows\System32\cryptnet.dll

```

Frequently Asked Questions

4

This section lists and describes some of the frequently asked questions about the ENS2 functionality.

How are the credentials or authentication tokens handled?

Although the client shares the credentials or tokens with the ENS2 environment upon registration, the credentials or tokens are not saved on the Workspace ONE UEM servers. The Exchange server sends the encrypted authentication information to the Workspace ONE UEM as part of a notification when a new email is available. From that notification (Exchange to the ENS2), the credentials are decrypted and used to make any requests necessary to the Exchange server. The credentials are discarded after performing the necessary requests.

If credentials are not saved, what data does the ENS save? How secure is the ENS?

- Workspace ONE stores a list of devices and a list of public private key-pairs used to decrypt the credentials when the notifications are sent from the Exchange. The database is saved on a Virtual Private Cloud (private subnet) secured using firewall. There is no direct access from the Internet to this subnet. All access is controlled using VPC and firewall rules and only web servers with a single account have access to the database.
- Workspace ONE saves the log files to help debug issues and monitor the system. The log does not contain any private information (PI) of the customers and access is secured using the account permissions.

Where is the ENS hosted? Are there instances configured to serve each region based on data sovereignty laws?

The ENS is hosted in multiple regions. We have various environments spanning the US, Europe, and Asia regions that permit us to abide by data sovereignty rules.

What data is transmitted through the ENS server without being saved? How is it secured?

- User credentials that are encrypted with the RSA encryption.
- Email subject and sender (sent using HTTPS).
- Future functionality: The functionality to control what data (if any) is sent or fetched for the notification. You can also control the data from an email that is used in the notification payload.
- All communication is made through HTTPS.

What is the dependency of ENS on cloud services?

- AWS Simple Notification Service (SNS) is used for managing push notification in the AWS Cloud deployment.
- Cloud Notification Service (CNS) is mandatory for passing notifications to Apple/Android devices for on-premises deployments.
- AWS Relational Database Service (RDS) is used for the data persistence.

When sending requests to the Exchange which user agent does the ENS2 use?

The ENS2 uses the MailNotificationService/v2 (ExchangeServicesClient/15.00.0913.015) user agent. The value '15.00.0913.015' changes as new libraries from Microsoft are released and are updated for using ENS2.

What email folders does ENS2 monitor for incoming messages and actions?

ENS2 only monitors a user's Inbox folder.

How does the ENS server authenticate a device before subscribing (Boxer application) to the notifications?

Each ENS tenant is issued an access token, the device provides the token to access the ENS APIs. In addition, the user credentials are required to create a subscription for a user.

How is the ENS server discovered on the device? Which application is used?

The Boxer application is configured with an ENS endpoint provided by the Workspace ONE console. The Boxer application manages the ENS subscriptions.

How does the application authenticate the ENS server?

The Boxer application uses certificate pinning to validate the ENS endpoint.

How are the public-private keys generated and managed on the ENS server? One key at the time of installation or one key (or key-pair) per mailbox or user ID?

The public or private key-pairs are generated in advance and stored in the ENS database. Each device is assigned a unique key-pair when the device registers with the ENS service.

How many pairs of public-private keys are used for moving credentials from the device?

There is one key-pair for each user that is used to encrypt all sensitive data transmitted from the client.

How are the keys and secrets managed on the ENS server?

Public or private key-pairs, hashed email ID, device ID, partial certificate, APNS token, EWS URL, and subscription ID manage the keys and secrets on the ENS server.

When a device initiates a connection to the ENS server what measures are taken on the client side and the ENS server to prevent against a man-in-the-middle attack?

The device uses TLS pinning to ensure that the device is connected to a valid ENS endpoint.

What security measures are used in the notification subscription flow to ensure that a user credentials cannot be intercepted in transmission?

In the older version of the ENS, the device provides the EWS endpoint used for subscriptions or the Autodiscover dynamically provides the EWS endpoint. In the latest version of the ENS, a set of EWS endpoints and their associated certificate fingerprints is associated with each API token, and the ENS server connects to the pre-configured endpoints validated by their fingerprints.

What data is stored by the ENS locally?

Each ENS server is stateless, apart from the API key which is refreshed every one hour.

What data is stored by the ENS on the SQL server?

The ENS stores the public or private key-pairs, hashed email ID, device ID, partial certificate, APNS token, EWS URL, and subscription ID on the SQL server.

How does the ENS handle connections to the SQL server?

The ENS stores the encrypted connection string in the **web.config** file which is decrypted and used to open a connection with the database.

How are credentials to the SQL server managed and secured by the ENS server?

Credentials are present in the configuration file and are encrypted with the **RsaProtectedConfigurationProvider**.

How does the connection pooling and failover work with redundant SQL servers?

The ENS fully supports SQL Always ON.

In a deployment scenario where the redundant servers are across different data centers, how is the data replicated across the data centers?

The ENS does not provide any explicit support for multiple data centers.

Does VMware have any guidelines for hardening the ENS server?

The standard server hardening procedures only apply. The only requirement is that the server must be accessible through HTTPS.

Does VMware have any guidelines for hardening the SQL Server (that accepts the connections from the DMZ hosted on the ENS server)?

The standard server hardening procedures only apply. The only requirement is that the ENS server can connect to the SQL server endpoint.

Does the ENS server work if the connection from the device is bridged at a reverse-proxy or load-balancer? The connection terminates on the proxy and a spate connection transmits the payload.

The only requirement is that the device can communicate with the ENS endpoint over plain HTTPS. Long-running connections or other special behavior is not required, so a standard proxy might not cause problems.

How are the service account credentials managed on the ENS server?

The ENS2 does not use any service account.

How are the APNS certificates provisioned and handled on the ENS server?

The ENS servers do not directly require APNS certificates. The ENS notifications are routed through the CNS, and the communication between the ENS and CNS are authenticated through the mutual TLS. The CNS certificate is provisioned from the Workspace ONE console and stored in the `web.config` file on the ENS.

How does the ENS construct the webhook URL?

Whenever the ENS receives the request, extract the **requesturi** from the ENS and then use the **requesturi** as the webhook URL. When a request is made to the **registerdevicev2** endpoint, the ENS gets the credentials which are in the encrypted format and the same encrypted data is used and added in the webhook URL query parameters. For the credentials, use the user name and password in the basic authentication, use the **OAuthAccessToken** in the **OAuth**, and use the CBA data in CBA.

How are the user credentials encrypted and encoded for the webhook URL?

Encrypt the user credentials with the asymmetric cryptographic algorithm, that is, RSA with Public-Key Cryptography Standards 1 (PKCS 1) padding using the **BouncyCastle** crypto library. After encrypting the credentials, encode the credentials using, the **HttpUtility.UrlEncode**.

What encryption methods and tools are used to encrypt the user credentials for the webhook URL?

Use the RSA encryption with the PKCS 1 padding algorithm. For more information on PKCS 1, see: <https://github.com/bcg-it/bc-csharp/blob/master/crypto/src/crypto/encodings/Pkcs1Encoding.cs>.

If two users share the same password credentials, then the encrypted password in webhook URL will have the same value?

Since RSA encryption is used, different encrypted payload is obtained even though two inputs or passwords are the same.

Can local caching storage be used for password storage? After using the plaintext password (credential) to fetch the email how is the password purged?

The ENS does not store any caching in the local storage after decrypting the credentials. The ENS synchronizes with the Exchange and the object holding the password is disposed.

How does the flow work when the ENS server decrypts the user credentials (password) to plaintext to fetch the email?

The ENS decrypts the credentials (that are part of the callback URL on which the ENS receives notifications from the Exchange) using the private key of the user and synchronizes with the Exchange to get the email information.

Does the ENS server need a service account for the Exchange server? If yes, what are the required access privileges?

The ENS is explicitly designed to operate with no service accounts.

How does the ENS authenticate with the EWS or the Exchange? How are the credentials managed?

The device initiates all the EWS subscriptions using the user credentials stored on the device. The device encrypts the user credentials with a unique public or private key-pair and calls the subscription endpoint. The ENS service decrypts the credentials and uses the credentials to create the EWS subscription.

The credentials are not stored in memory.

How does the device resubscription to notification function, after a user has changed the password on the Exchange?

When the ENS endpoint notices that a subscription has failed, the ENS sends a silent push notification to the device to inform that the subscription must be recreated. In addition, the device can call a status endpoint on the ENS service to determine if a subscription is active. This permits the device to determine if the device must resubscribe to the ENS at the application start time.

How does the ENS protect against device spoofing to ensure only devices enrolled on the WorkSpace ONE UEM are allowed?

If a strict device compliance is required, the ENS configuration must use a SEG to communicate with the EWS, instead of connecting to the EWS directly. The ENS server includes the device IDs in all calls to the SEG. The SEG validates if the device IDs are compliant before allowing the subscriptions to be created.

Does the ENS check periodically if a device is compromised before sending push notifications to that device?

When the ENS is configured to communicate with the EWS through the SEG, the SEG prevents the ENS service from retrieving notifications for the compromised devices.

How is the Boxer application configured for registering with the ENS for email notifications?

The configuration for using the ENS is provided to the Boxer application through the same channel as all other Boxer configurations.

Assuming there is a URI in the Boxer configuration for making the initial connection to the ENS, provide a sample URI.

An ENS endpoint URL is similar to <https://ens.getboxer.com/api/ens>. Find the list of available cloud ENS endpoints at: [ENS Endpoints and IP Whitelist](#).

How is the webhook URL constructed with the user credentials encrypted encoding inline?

Example of a sample webhook URL: <http://10.89.240.187/mailnotificationsservice/api/ens/pushnotificationlistener?id=4&f=Plaw5DI0czKmhWWowlJnj%2bFsjDPNt0Eplgg5EaBqgiVsrAmlI%2bIXLy9ik8JIUklQsELIefjp7z8jBgSA2nxa4p7Hwxze6jUiT39%2bjaAea8df7rMUN3xjAtJPTb60ifXHULIH%2bjLPIRMEN92zNJGAU50Cj%2fp2fpq>. Here, the **id** is the userinfo pertaining to a single user and the parameter **f** is the filler key containing the encrypted information.

How is the payload of push notification constructed with any enrichment and/or trimming of data pulled from the Exchange by the ENS?

On receiving a webhook from the EWS that a new message has arrived, the ENS server decrypts the credentials in the webhook parameters. The ENS uses the credentials to call back the EWS to collect the details of any new messages. This data is used to form any APNS notifications to be sent. The **PolicyLimitNotificationText** key in the Boxer configuration controls the content of the generated notifications. The content is later passed to the ENS server. The following options are available:

Values	Description
0	Sender, subject, and preview
1	Sender and subject
2	Sender only
3	Generic message (new message)
4	Set the notification to none (only the badge is updated).

What are the crypto libraries and binaries used by the ENS?

The ENS uses the `BouncyCastle.crypto` library. For more information about the `BouncyCastle.crypto` library, see <https://www.nuget.org/packages/BouncyCastle>.

Are the credentials stored in the memory of the ENS servers. If yes, for how long are the credentials stored and what are the mitigating controls to prevent an unauthorized person from accessing this data?

When the ENS receives push notifications from the Exchange with encrypted credentials, the ENS decrypts the credentials in memory and sends a web request to the Exchange server with the credentials to retrieve the subject and summary of the new email. The ENS then sends push notifications through the SNS and discards the credentials from the memory. This process takes less than a minute. Nobody has access to the credentials data as the data is not stored anywhere.

Credentials are discarded after performing the necessary requests. Provide more information on the discarding process?

The ENS runs on the .net which provides garbage collection for unused objects in the memory. The ENS depends on this process to clean up the memory.

Can a third-party SIEM tools be integrated with the ENS? Does VMware support any form of internal monitoring and maintenance of access logs to the ENS to identify suspicious or malicious events?

Currently, the ENS does not have a solution to feed data into the SIEM solution. You can contact your support or account team with your requirements.

Describe the monitoring access level the ENS2 has on the Inbox folder. Is ENS able to view the email details (sender, subject and email body) contained in Inbox folders?

The ENS can only access the sender, subject, and the preview fields. The ENS does not synchronize or fetch the entire Inbox folder. The ENS only fetches one email data at a time and discards after constructing a notification.

Are copies of emails stored on the ENS server or does the ENS server act as a middle-man to pass email details and notifications to the mobile device through the AWS SNS?

Email data is not stored in the ENS server.

Can the OAuth token used to get the mail information (sender, recipients, subject, if mail attachment) be used for the Exchange notification scope only (and to get only this information)? Or does the OAuth token have the permission to read or write emails (and see all content of an email)?

The ENS relies on the **EWS.AccessAsUser.All** permission to gain access to email information using the OAuth token. Microsoft does not provide granular permissions for the EWS access. The

EWS.AccessAsUser.All is the only permission you can provide to gain access to the EWS. The ENS fetches the required information about a new message (sender, recipient, subject, and so on) when the Exchange notifies the ENS through a push notification. The ENS then sends this information to the APNS or FCM and discards the information. The ENS never reads any other information or stores the information on the ENS server.

Both the Boxer application and the ENS share application registration on the Azure AD and the Boxer application uses these permissions.