Apigee^{TM}

Apigee Edge for Private Cloud

v4.16.05

Sept. 26, 2016

External Authentication Configuration

Apigee$^{\text{(TM)}}$ and the Apigee logo are trademarks or registered trademarks of Apigee Corp. or its subsidiaries. All other trademarks are the property of their respective owners. All specifications are subject to change without notice.

THE CONTENTS OF THIS PUBLICATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT OF INTELLECTUAL PROPERTY.

APIGEE CORPORATION SHALL NOT UNDER ANY CIRCUMSTANCES BE LIABLE TO ANY PERSON FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, DAMAGES RESULTING FROM THE USE OF OR RELIANCE ON THE INFORMATION IN THIS PUBLICATION, LOSS OF PROFITS, REVENUE OR DATA, EVEN IF APIGEE CORPORATION HAS BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Contact Information

INDIA

No.17/2, 2B Cross, 7th Main, 2 & 3
Floor, Off 80 Feet Road, 3rd Block
Koramangala, Bangalore 560034
Call +91 80 67696800
www.apigee.com

USA

10 Almaden Boulevard,
16th Floor, San Jose
CA 95113
Call +1 (408) 343-7300
www.apigee.com

UK

3 Sheldon Square
London W2 6HY

Call: +44 (0) 750 123 2390
www.apigee.com/

Contents

Introduction

This document explains how to integrate an external directory service into an existing Apigee Edge Private Cloud installation. This feature is designed to work with any directory service that supports LDAP, such as Active Directory, OpenLDAP, and others. All the steps are included here to get Apigee Edge working with your LDAP service.

An external LDAP solution allows system administrators to manage user credentials from a centralized directory management service, external to systems like Apigee Edge that use them. The feature described in this document supports both direct and indirect binding authentication.

Audience

This document assumes that you are an Apigee Edge for Private Cloud global system administrator and that you have an account the external directory service.

Overview

By default, Apigee Edge uses an internal OpenLDAP instance to store credentials that are used for user **authentication**. However, you can configure Edge to use an **external authentication LDAP service** instead of the internal one. The procedure for this external configuration is explained in this document.

Edge also stores role-based access **authorization credentials** in a separate, internal LDAP instance. Whether or not you configure an external authentication service, authorization credentials are **always** stored in this internal LDAP instance. The procedure for adding users that exist in the external LDAP system to the Edge authorization LDAP are explained in this document.

Note that *authentication* refers to validating a user's identity, while *authorization* refers to verifying the level of permission an authenticated user is granted to use Apigee Edge features. See What you need to know about Edge authentication and authorization.

What you need to know about Edge authentication and authorization

It's useful to understand the difference between authentication and authorization and how Apigee Edge manages these two activities.

About authentication

Users who access Apigee Edge either through the UI or APIs must be authenticated. By default, Edge user credentials for authentication are stored in an internal OpenLDAP instance. Typically, users must register or be asked to register for an Apigee account, and at that time they supply their username, email address, password credentials, and other metadata. This information is stored in and managed by the authentication LDAP.

However, if you wish to use an external LDAP to manage user credentials on behalf of Edge, you can do so by configuring Edge to use the external LDAP system instead of the internal one. When an external LDAP is configured, user credentials are validated against that external store, as explained in this document.

About authorization

Edge organization administrators can grant specific permissions to users to interact with Apigee Edge entities like API proxies, products, caches, deployments, and so on. Permissions are granted through the assignment of roles to users. Edge includes several built-in roles, and, if needed, org administrators can define custom roles. For example, a user can be granted authorization (through a role) to create and update API proxies, but not to deploy them to a production environment.

The key credential used by the Edge authorization system is the user's email address. This credential (along with some other metadata) is always stored in Edge's internal authorization LDAP. This LDAP is entirely separate from the authentication LDAP (whether internal or external).

Users who are authenticated through an external LDAP must also be manually provisioned into the authorization LDAP system. Details are explained in this document.

Note: User passwords from the external LDAP system are never stored/cached in the internal authorization system.

For more background on authorization and RBAC, see "Managing organization users" and "Assigning roles" in the main Apigee Edge documentation. Also, refer to "Organization and Environment Maintenance" in the *Apigee Edge for Private Cloud Operations Guide*.

For a deeper view, see also Understanding edge authentication and authorization flows in the Appendix.

Understanding direct and indirect binding authentication

The external authorization feature supports both **direct** and **indirect** binding authentication through the external LDAP system.

Summary: Indirect binding authentication requires a search on the external LDAP for credentials that match the email address, username, or other ID supplied by the user at login. With direct binding authentication, no search is performed--credentials are sent to and validated by the LDAP service directly. Direct binding authentication is considered to be more efficient because there is no searching involved.

About indirect binding authentication

With indirect binding authentication, the user enters a credential, such as an email address, username, or some other attribute, and Edge searches authentication system for this credential/value. If the search result is successful, the system extracts the LDAP DN from the search results and uses it with a provided password to authenticate the user.

The key point to know is that indirect binding authentication requires the caller (e.g., Apigee Edge) to provide external LDAP admin credentials so that Edge can "log in" to the external LDAP and perform the search. You must provide these credentials in an Edge configuration file, which is described later in this document. Steps are also described for encrypting the password credential.

About direct binding authentication

With direct binding authentication, Edge sends credentials entered by a user directly to the external authentication system. In this case, no search is performed on the external system. Either the provided credentials succeed or they fail (e.g., if the user is not present in the external LDAP or if the password is incorrect, the login will fail).

Direct binding authentication does not require you to configure admin credentials for the external auth system in Apigee Edge (as with indirect binding authentication); however, there is a simple configuration step that you must perform, which is described later in this document.

Enabling external authentication

This section explains how to obtain, install, and configure the components required to integrate an external LDAP service into Apigee Edge for user authentication.

1) Prerequisites
2) Configuring the `management-server.properties` file
3) Testing the installation

Prerequisites

- You must have an Apigee Edge for Private Cloud 4.16.01 installation.
- You must have global system administrator credentials on Apigee Edge for Private Cloud to perform this installation.
- You need to know the root directory of your Apigee Edge for Private Cloud installation. The default root directory is `/opt`.
- You must add your **Edge global system administrator credentials** to the external LDAP. Remember that by default, the sysadmin credentials are stored in the Edge internal LDAP. Once you switch to the **external** LDAP, your sysadmin credentials will be authenticated there instead. Therefore, you must provision the credentials to the external system **before** enabling external authentication in Edge.

  For example if you have configured and installed Apigee Edge for Private Cloud with global system administrator credentials as...

  username: `edgeuser@mydomain.com`

  password: `Secret123`

  then the user edgeuser@mydomain.com with password `Secret123` must also be present in the external LDAP.
- If you are running a Management Server cluster, note that you must perform all of the steps in this document for each Management Server.

Configuration overview

The main activity you'll perform is configuring the `management-server.properties` file. This activity includes stopping and starting the Edge Management Server, deciding whether you want to use direct or indirect binding, encrypting sensitive credentials, and other related tasks.

In the following sections, we walk you through each step.

Configuring the management-server.properties file

1) **Important:** Decide now whether you intend to use the indirect or direct binding authentication method. This decision will affect some aspects of the configuration. See Understanding direct and indirect binding authentication.
2) **Important:** You must do an additional configuration (described later in this document) under either (or both) of the following circumstances: (a) if you intend to have users log in using usernames that are not email addresses. In this case, your sysadmin user must also authenticate with a username **and/or** (b) if the password for your sysadmin user account in your external LDAP is different from the password you configured when you first installed Apigee Edge for Private Cloud. See Additional configuration required in the event of different sysadmin credentials.
3) **Important:** You must do these config steps on each Apigee Edge Management Server (if you are running more than one).
4) Open `/opt/apigee/customer/application/management-server.properties` in a text editor. If the file does not exist, create it.
5) Add the following line. **Note: Be sure that there are no trailing spaces at the end of the line.**

   ```
   conf_security_authentication.user.store=externalized.authentication
   ```

   This line is required. It adds the external authentication feature to your Apigee Edge for Private Cloud installation.
6) To make this step easy, we have created two well-commented **sample configurations -- one for direct and one for indirect** binding authentication. Go to the sample below for the binding you wish to use, and complete the configuration:

   A. DIRECT BINDING configuration sample

   B. INDIRECT BINDING configuration sample

   Note: For a handy side-by-side view of these two different config options, see also Appendix A. External authentication configuration options for management-server.properties.

A. DIRECT BINDING configuration sample

```
## The first property is always required to enable the external
authorization feature. Do not change it.

conf_security_externalized.authentication.implementation.class=com.apigee.
rbac.impl.LdapAuthenticatorImpl


## Identify the type of binding:
```

```
   # Set to "true" for direct binding
   # Set to "false" for indirect binding.

## Set it to true for DIRECT binding.

conf_security_externalized.authentication.bind.direct.type=true


## The next seven properties are needed regardless of direct or indirect
binding. You need to configure these per your external authentication
installation.

## The IP or domain for your external LDAP instance.

conf_security_externalized.authentication.server.url=ldap://localhost:389


## Your external LDAP server version.

conf_security_externalized.authentication.server.version=3


## The server timeout in milliseconds.

conf_security_externalized.authentication.server.conn.timeout=50000


## Change these baseDN values to match your external LDAP service. This
attribute value will be provided by your external LDAP administrator, and
may have more or fewer dc elements depending on your setup.

conf_security_externalized.authentication.user.store.baseDN=dc=apigee,dc=com


## Do not change this search string. It is used internally.

conf_security_externalized.authentication.user.store.search.query=(&(${userAttribute}=${userId}))


## Identifies the external LDAP property you want to bind against for
Authentication. For example if you are binding against an email address,
this would typically be in the userPrincipalName property in your external
LDAP instance. Alternatively if you are binding against the user’s ID,
this would typically be in the sAMAccountName property:

conf_security_externalized.authentication.user.store.user.attribute=userPrincipalName


## The LDAP attribute where the user email value is stored. For direct
binding, set it to userPrincipalName.

conf_security_externalized.authentication.user.store.user.email.attribute=userPrincipalName
```

```
## ONLY needed for DIRECT binding.
## The direct.bind.user.directDN property defines the string that is used
for the bind against the external authentication service. Ensure it is set
as follows:

conf_security_externalized.authentication.direct.bind.user.directDN=${userDN}
```

B. INDIRECT BINDING configuration sample

```
## Required to enable the external authorization feature. Do not change
it.

conf_security_externalized.authentication.implementation.class=com.apigee.rbac.impl.LdapAuthenticatorImpl


## Identifies the type of binding:
   # Set to "true" for direct binding
   # Set to "false" for indirect binding.

## Set it to false for INDIRECT binding.

conf_security_externalized.authentication.bind.direct.type=false


## The next seven properties are needed regardless of direct or indirect
binding. You need to configure these per your external LDAP installation.

## The IP or domain for your external LDAP instance.

conf_security_externalized.authentication.server.url=ldap://localhost:389


## Replace with your external LDAP server version.

conf_security_externalized.authentication.server.version=3


## Set the server timeout in milliseconds.

conf_security_externalized.authentication.server.conn.timeout=50000


## Change these baseDN values to match your external LDAP service. This
attribute value will be provided by your external LDAP administrator, and
may have more or fewer dc elements depending on your setup.

conf_security_externalized.authentication.user.store.baseDN=dc=apigee,dc=com
```

```
## Do not change this search string. It is used internally.
conf_security_externalized.authentication.user.store.search.query=(&(${userAttribute}=${userId}))

## Identifies the external LDAP property you want to bind against for
Authentication. For example if you are binding against an email address,
this would typically be in the userPrincipalName property in your external
LDAP instance. Alternatively if you are binding against the user's ID,
this would typically be in the sAMAccountName property. See also
Additional configuration required in the event of different sysadmin
credentials.

conf_security_externalized.authentication.user.store.user.attribute=userPrincipalName

## Used by Apigee to perform the Authorization step and currently, Apigee
only supports email address for Authorization. Make sure to set it to the
attribute in your external LDAP that stores the user's email address.
Typically this will be in the userPrincipalName property. See also What
you need to know about Edge authentication and authorization.

conf_security_externalized.authentication.user.store.user.email.attribute=userPrincipalName

## The external LDAP username (for a user with search privileges on the
external LDAP) and password and whether the password is encrypted. You
must also set the attribute externalized.authentication.bind.direct.type
to false.

## The password attribute can be encrypted or in plain text. See Indirect
binding only: Encrypting the external LDAP user's password for encryption
instructions. Set the password.encrypted attribute to "true" if the
password is encrypted. Set it to "false" if the password is in plain text.

conf_security_externalized.authentication.indirect.bind.server.admin.dn=myExtLdapUsername

conf_security_externalized.authentication.indirect.bind.server.admin.password=myExtLdapPassword

conf_security_externalized.authentication.indirect.bind.server.admin.password.encrypted=true
```

7) Restart the Management Server:

```
/opt/apigee/apigee-service/bin/apigee-service edge-management-server restart
```

8) Verify that the server is running:

```
/opt/apigee/apigee-service/bin/apigee-all status
```

Testing the installation

1) Verify that the server is running:

   ```
   /opt/apigee/apigee-service/bin/apigee-all status
   ```

2) Execute this command, providing a set of **Apigee Edge global system admin credentials**. The API call we're going to test can only be executed by an Edge sysadmin.

   Important: The identical credentials must exist in your external LDAP account. If not, you need to add them now. Note that the username is usually an email address; however, it depends on how you have configured external authentication, as explained previously in this document.

   ```
   curl -v http://<your-management-server-ip>:8080/v1/o -u <Edge
   Sysadmin Username>
   ```

   For example:

   ```
   curl -v http://192.168.52.100:8080/v1/o -u jdoe@mydomain.com
   ```

3) Enter your password when prompted.

   If the command returns a 200 status and a list of organizations, the configuration is correct. This command verifies that the API call to the Edge Management Server was successfully authenticated through the external LDAP system.

Indirect binding only: Encrypting the external LDAP user's password

If you are using indirect binding, you need to provide an external LDAP username and password in `management-server.properties` that Apigee uses to log into the external LDAP and perform the indirect credential search.

Note: Using plain text passwords in config files may be adequate for testing purposes; however, for production environments, encryption is highly recommended.

The following steps explain how to encrypt your password:

1) Execute the following Java utility, replacing the <YOUR EXTERNAL LDAP PASSWORD> with your actual external LDAP password:

   ```
   java -cp /opt/apigee/edge-
   gateway/lib/thirdparty/*:/opt/apigee/edge-
   gateway/lib/kernel/*:/opt/apigee/edge-gateway/lib/infra/libraries/*
   com.apigee.util.CredentialUtil --password="<YOUR EXTERNAL LDAP
   PASSWORD>"
   ```

2) You will see a newline followed by what looks like a random character string. Copy that string.
3) Edit `/opt/apigee/customer/application/management-server.properties`.
4) Update the below property, replacing `<myAdPassword>` with the string you copied from step 1, above.

   ```
   conf_security_externalized.authentication.indirect.bind.server.admi
   n.password=<myAdPassword>
   ```

5) Be sure the following property is set to true:

   ```
   conf_security_externalized.authentication.indirect.bind.server.admi
   n.password.encrypted=true
   ```

6) Save the file and then start the Management Server:

   ```
   /opt/apigee/apigee-service/bin/apigee-service edge-management-
   server restart
   ```

7) Verify that the server is running:

   ```
   /opt/apigee/apigee-service/bin/apigee-all status
   ```

Testing the installation

See the testing section at the end of Configuring the management-server.properties file, and perform the same test described there.

Configuring TLS/SSL

This section explains how to configure SSL for the external authorization server. For general information, see "About SSL".

1) Install the external LDAP Certificate Services.
2) Obtain the Server Certificate.

   For example: `certutil -ca.cert client.crt`
3) Change to your latest Java version home directory:

   ```
   cd /usr/java/latest
   ```
4) Import the Server Certificate. For example:

   ```
   sudo ./bin/keytool -import -keystore ./jre/lib/security/cacerts -
   file <FULLY-QUALIFIED-PATH-TO-THE-CERT-FILE> -alias <CERT-ALIAS>
   ```

   Where `<CERT-ALIAS>` is optional, but recommended. Replace `<CERT-ALIAS>` with a text name that you can use later to refer to the certificate, for example if you want to delete it.

   Note: The Default Keystore password used by Java is '**changeit**'. If this has been changed already you will need to get your sysadmin to provide the keystore password so you add your certificate.
5) Open `/opt/apigee/customer/application/management-server.properties` in a text editor.
6) Change the `conf_security_externalized.authentication.server.url` property value as follows:

   Old Value : `ldap://localhost:389`
   New Value : `ldaps://localhost:636`
7) Start the Management Server:

   ```
   /opt/apigee/apigee-service/bin/apigee-service edge-management-
   server restart
   ```
8) Verify that the server is running:

   ```
   /opt/apigee/apigee-service/bin/apigee-all status
   ```

Testing the configuration

See the testing section at the end of Configuring the security.properties file, and perform the same test.

Additional configuration required in the event of different sysadmin credentials

When you first installed Apigee Edge, a special kind of user was created called a sysadmin user, and at the same time some additional config files were updated with this user's details. If you configure your external LDAP to authenticate using a non-email address username and / or you have a different password in your external LDAP for this sysadmin user, then you will need to make the changes described in this section.

There are two locations that need to be updated:

- Apigee management UI logs into the Apigee Management Server using credentials that are stored encrypted in a configuration file. This update is required when either/both username or password for your sysadmin user is different.
- Apigee stores the sysadmin username in another file which is used when running various Apigee utility scripts. This update is only required when the username of your sysadmin user is different.

Editing the Edge management UI credential

1. Edit the silent config file that you used to install the Edge UI to set the following properties:

   ```
   ADMIN_EMAIL=newUser
   APIGEE_ADMINPW=newPW
   SMTPHOST=smtp.gmail.com
   SMTPPORT=465
   SMTPUSER=foo@gmail.com
   SMTPPASSWORD=bar
   SMTPSSL=y
   ```

   Note that you have to include the SMTP properties when passing the new password because all properties on the UI are reset.

2. Use the `apigee-setup` utility to reset the password on the Edge UI from the config file:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ui -f configFile
   ```

Testing the configuration

1) Open the management UI in a browser at:

   ```
   http://<management-server-IP>:9000/
   ```

   For example:

```
http://192.168.52.100:9000/
```

2) Log in using the new credentials. If the login succeeds, the configuration is correct.

Editing the Edge sysadmin username store for Apigee utility scripts

1. Edit the silent config file that you used to install the Edge UI to set the following property to change the value of `ADMIN_EMAIL` to the username you will be using for your sysadmin user in your external LDAP:

   ```
   APIGEE_EMAIL=newUser
   ```

2. Use the `apigee-setup` utility to reset the username on all Edge component from the config file:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ui -f configFile
   ```

   You must run this command on all Edge component on all Edge nodes, including: Management Server, Router, Message Processor, Qpid, Postgres.

Testing the configuration

1. Verify that you can access the central POD. On the Management Server, run the following CURL command:

   ```
   > curl -u sysAdminEmail:password
   http://localhost:8080/v1/servers?pod=central
   ```

   You should see output in the form:

   ```
   [ {
    "internalIP" : "192.168.1.11",
    "isUp" : true,
    "pod" : "central",
    "reachable" : true,
    "region" : "dc-1",
    "tags" : {
      "property" : [ ]
    },
    "type" : [ "application-datastore", "scheduler-datastore",
   "management-server", "auth-datastore", "apimodel-datastore", "user-
   settings-datastore", "audit-datastore" ],
    "uUID" : "d4bc87c6-2baf-4575-98aa-88c37b260469"
   }, {
    "externalHostName" : "localhost",
   ```

```
      "externalIP" : "192.168.1.11",
      "internalHostName" : "localhost",
      "internalIP" : "192.168.1.11",
      "isUp" : true,
      "pod" : "central",
      "reachable" : true,
      "region" : "dc-1",
      "tags" : {
        "property" : [ {
          "name" : "started.at",
          "value" : "1454691312854"
        }, ... ]
      },
      "type" : [ "qpid-server" ],
 "uUID" : "9681202c-8c6e-4da1-b59b-23e3ef092f34"
} ]
```

Turning external authentication off

Perform these steps if you want to turn off external authentication and revert to using the internal authentication LDAP in Apigee Edge.

Important: You must do the following steps on each Apigee Edge Management Server.

1) Open `/opt/apigee/customer/application/management-server.properties` in a text editor.
2) Set the `conf_security_authentication.user.store` property to `ldap`. **Note: Be sure that there are no trailing spaces at the end of the line.**

   ```
   conf_security_authentication.user.store=ldap
   ```
3) **OPTIONALLY, only applicable if you were using a non-email address username or a different password in your external LDAP for your sysadmin user:**
   a. Follow the steps you previously followed in Additional configuration required in the event of different sysadmin credentials, above, but substituting the external LDAP username with your Apigee Edge sysadmin user's email address.
4) Start the Management Server:

   ```
   /opt/apigee/apigee-service/bin/apigee-service edge-management-
   server restart
   ```
5) Verify that the server is running:

   ```
   /opt/apigee/apigee-service/bin/apigee-all status
   ```

6) **Important:** An Edge organization administrator must take the following actions after external authentication is turned off:

   - Make sure there are no users in Apigee Edge that should not be there. You need to manually remove those users.
   - Communicate to users that because the external authentication has been turned off, they need to either start using whatever their original password was (if they remember) or complete a "forgot password" process in order to log in.

External Role Mapping

External Role Mapping lets you map your own groups or roles to role-based access control (RBAC) roles and groups created on Apigee Edge.

Prerequisites

- You must be an Apigee Private Cloud system administrator with global system admin credentials to perform this configuration.
- You need to know the root directory of your Apigee Edge Private Cloud installation. The default root directory is `/opt`. If you chose a different root directory during the Apigee Edge Private Cloud installation, use that instead of `/opt` as you follow these instructions.
- Obtain the required JAR files from Apigee.

Ensure users are registered on Edge and in your directory service

When using role mapping, all users who access Edge must exist in both your external directory service and in the Edge user repository. That means when you add a user to your external directory service, you must also add that some user to the Apigee user repository.

For example, user **a01@company.com** exists in your external directory group **'apiadmin'**. You then want to map user **a01@company.com** to the **orgadmin** role in Edge. Therefore, user **a01@company.com** must first be added to the **orgadmin** group on Edge.

See Creating global users for more on creating Edge users and assigning them to roles.

Default configuration

External role mapping is **disabled** by default.

Enabling External Role Mapping

1. Before you can complete the following configuration, you must create a Java class that implements the `ExternalRoleMapperService` interface. For details about this implementation, see About the ExternalRoleMapperImpl sample implementation.

2. Log into your Apigee Edge Management Server and then stop the Management Server:

   ```
   /opt/apigee/apigee-service/bin/apigee-service edge-management-
   server stop
   ```

3. Check the status of the servers. Be sure the Management Server is stopped/not running:

   ```
   /opt/apigee/apigee-service/bin/apigee-all status
   ```

4. Open `/opt/apigee/customer/application/management-server.properties` in a text editor.
5. Edit the `management-server.properties` file with the following settings:

   ```
   conf_security_authentication.user.store=externalized.authentication
   conf_security_externalized.authentication.role.mapper.enabled=true
   conf_security_externalized.authentication.role.mapper.implementatio
   n.class=com.customer.authorization.impl.ExternalRoleMapperImpl
   ```

   Important: The implementation class and package name referenced above (`ExternalRoleMapperImpl`) is only an example -- it is a class that you must implement and that you can name the class and package whatever you wish. For details about implementing this class, see About the ExternalRoleMapperImpl sample implementation.

6. Save the `management-server.properties` file.
7. Start the Management Server:

   ```
   /opt/apigee/apigee-service/bin/apigee-service edge-management-
   server start
   ```

8. Verify that the server is running:

   ```
   /opt/apigee/apigee-service/bin/apigee-all status
   ```

Disabling External Authorization

To disable external authorization:

1. Change the authentication user store to `ldap`:

   ```
   conf_security_authentication.user.store=ldap
   ```

2. Set this property to `false`:

   ```
   conf_security_externalized.authentication.role.mapper.enabled=false
   ```

3. Restart the Management Server:

```
/opt/apigee/apigee-service/bin/apigee-service edge-management-
server restart
```

About the ExternalRoleMapperImpl sample implementation

In the `management-server.properties` config file described previously in Enabling External Role Mapping, note the this line:

```
conf_security_externalized.authentication.role.mapper.implementation.clas
s=com.customer.authorization.impl.ExternalRoleMapperImpl
```

This class implements the `ExternalRoleMapperService` interface, and is required. You need to create your own implementation of this class that reflects your respective groups. When finished, place the compiled class in a JAR and put that JAR in `/<install_dir>/apigee/edge-gateway/lib/infra/libraries`.

Warning: If you apply a patch to Edge, or if you upgrade Edge to a later version, you must re-copy the JAR file to `/<instal_dir>/apigee/edge-gateway/lib/infra/libraries`.

You can name the class and package whatever you wish as long as it implements `ExternalRoleMapperService`, is accessible in your classpath, and is referenced correctly in the `management-server.properties` config file.

Below is a well-commented sample implementation of an `ExternalRoleMapperImpl` class. To compile this class, you must reference the following JAR file included with Edge:

`/<install_dir>/apigee/edge-gateway/lib/infra/libraries/authentication-1.0.0.jar`

Note: For better readability, we recommend that you copy the code into a text editor or IDE with wider margins.

```
package com.apigee.authenticate;


import com.apigee.authentication.ConfigBean;

import com.apigee.authentication.ConnectionException;

import com.apigee.authentication.ExternalRoleMapperService;

import com.apigee.authentication.NameSpace;

import com.apigee.authentication.NameSpacedRole;

import com.apigee.authorization.namespace.OrganizationNamespace;

import com.apigee.authorization.namespace.SystemNamespace;

import java.util.Collection;

import java.util.HashSet;

import javax.naming.NamingEnumeration;

import javax.naming.NamingException;
import javax.naming.directory.Attributes;
import javax.naming.directory.DirContext;
import javax.naming.directory.InitialDirContext;
import javax.naming.directory.SearchControls;
import javax.naming.directory.SearchResult;

/** *
 * Sample Implementation constructed with dummy roles with required namespaces.
 *
 * Created by GopiAlagar on 6/12/15.
 */

public class ExternalRoleMapperImpl implements ExternalRoleMapperService {
      InitialDirContext dirContext = null;

      @Override
      public void stop() throws Exception {
      }

      /**
       *
       * This method would be implemented by the customer, Below is the basic
       * example.
       *
       * If User has sysadmin role then it's expected to set SystemNameSpace
       * along with the
       * res\quested NameSpace. Otherwise role's requestedNameSpace to be set
       * for the NameSpacedRole.
       *
       * Collection<NameSpacedRole> results = new HashSet<NameSpacedRole>();
       *
       * NameSpacedRole sysNameSpace = new NameSpacedRole("sysadmin",
       * SystemNamespace.get());
       *
```

```
     * String orgName =
     * ((OrganizationNamespace)requestedNameSpace).getOrganization();
     *
     * NameSpacedRole orgNameSpace = new NameSpacedRole ("orgadmin",
     * requestedNameSpace);
     *
     * results.add(sysNameSpace);
     *
     * results.add(orgNameSpace);
     */

    public Collection<NameSpacedRole> getUserRoles(String userName,
                String password, NameSpace requestedNameSpace) {
        /*
         * There are 3 actions performed in the below implementation
         *
         * 1. Authenticate Given User against ADS
         *
         * 2. Fetch the internal groups from the ADS
         *
         * 3. Map the internal group into the apigee-edge roles
         */

        /*************************************************************/
        /******************* Authenticate Given User *******************/
        /*************************************************************/

        // Customer Specific Implementation will override this method
        // implementation.

        // Obtain dnName for given username or email address.
        String dnName = ImplementDnameLookupLogic();

        if (dnName == null) {
```

```
                  System.out.println("Error ");
            }

            DirContext dirContext = null;

            Collection<NameSpacedRole> results = new HashSet<NameSpacedRole>();

            try {
                  // Verify the credentials for a given username or dnName
                  // and password in order to create a directory context.
                  dirContext = ImplementDirectoryContextCreationLogic();

                  /****************************************************/
                  /********** Fetch internal groups *******************/
                  /****************************************************/

                  String groupDN = "OU=Groups,DC=corp,DC=wacapps,DC=net";
                  SearchControls controls = new SearchControls();
                  controls.setSearchScope(SearchControls.ONELEVEL_SCOPE);
                  NamingEnumeration<SearchResult> groups =
dirContext.search(groupDN,
                              "(objectClass=*)", new Object[] { "", "" },
controls);

                  if (groups.hasMoreElements()) {
                        while (groups.hasMoreElements()) {
                              SearchResult searchResult = groups.nextElement();
                              Attributes attributes =
searchResult.getAttributes();
                              String groupName =
attributes.get("name").get().toString();


                             /**************************************/
                             /** Map the internal group into the ***/
                             /** apigee-edge roles               ***/
                             /**************************************/


                              if (groupName.equals("BusDev")) {
                                    results.add(new
NameSpacedRole("businessAdmin",
                                                SystemNamespace.get()));


                              } else if (groupName.equals("DevSupport")) {
                                    results.add(new
NameSpacedRole("devOpsAdmin",
                                                SystemNamespace.get()));


                              } else if (groupName.equals("Engineering")) {
                                    if (requestedNameSpace instanceof
OrganizationNamespace) {
                                          String orgName =
((OrganizationNamespace) requestedNameSpace)
                                                      .getOrganization();
                                          results.add(new
NameSpacedRole("orgadmin",
                                                      new
OrganizationNamespace(orgName)));
                                    }


                              } else if (groupName.equals("Operations")
                                          || groupName.equals("IT")) {


                                    results.add(new NameSpacedRole("sysadmin",
                                                SystemNamespace.get()));


                              } else if (groupName.equals("Marketing")) {


                                    results.add(new
NameSpacedRole("marketAdmin",
                                                SystemNamespace.get()));
                              } else {

                                    results.add(new NameSpacedRole("readOnly",
                                                SystemNamespace.get()));
                              }
                        }

                  } else {

                        /*                       *
                         * In case of no group found or exception found we throw
empty
                         * roles.
                         */

                        System.out.println(" !!!!! NO  GROUPS FOUND !!!!!");
                  }

            } catch (Exception ex) {
                  ex.printStackTrace();
                  System.out.println("Error in authenticating User: {}"
                              + new Object[] { userName });
            } finally {
                  // Customer implementation to close
                  // ActiveDirectory/LDAP context.
            }

            return results;

      }

      @Override
      public void start(ConfigBean arg0) throws ConnectionException {
            try {
                  // Create InitialDirContext.
                  // Create a directory context based on the
                  // system admin user credentials.
                  dirContext = ImplementDirContextCreationLogicForSysAdmin();
            } catch (NamingException e) {
                  // TODO Auto-generated catch block
                  throw new ConnectionException(e);
            }
      }
}
```

About authorization

Edge organization administrators can grant specific permissions to users to interact with Apigee Edge entities like API proxies, products, caches, deployments, and so on. Permissions are granted through the assignment of roles to users. Edge includes several built-in roles, and, if needed, org administrators can define custom roles. For example, a user can be granted authorization (through a role) to create and update API proxies, but not to deploy them to a production environment.

The key credential used by the Edge authorization system is the user's email address. This credential (along with some other metadata) is always stored in Edge's internal authorization LDAP. This LDAP is entirely separate from the authentication LDAP (whether internal or external).

For more background on authorization and RBAC, see "Managing organization users" and "Assigning roles" in the main Apigee Edge documentation. Also, refer to "Organization and Environment Maintenance" in the Apigee Edge Private Cloud Operations Guide.

Appendix

A. External authentication configuration options for management-server.properties

The following table provides a comparison view of `management-server.properties` attributes required for direct and indirect binding for external authentication. Direct and indirect binding are described in Understanding direct and indirect binding authentication.

Note, in the following table, values are provided in between " ". When editing the management-server.properties file, include the value between the quotes (" ") but do not include the actual quotes.

Property	DIRECT bind	INDIRECT bind
conf_security_externalized.authentication.implementation.class=com.apigee.rbac.impl.LdapAuthenticatorImpl		
	This property is always required to enable the external authorization feature. Do not change it.	
conf_security_externalized.authentication.bind.direct.type=		
	Set to "**true**".	Set to "**false**".
conf_security_externalized.authentication.direct.bind.user.directDN=		
	If the username is an email address, set to "**${userDN}**". If the username is an ID, set to "**CN=${userDN},CN=Users,DC=apigee,DC=com**", replacing the CN=Users, DC=apigee,DC=com with appropriate values for your external LDAP.	Not required, comment out.
conf_security_externalized.authentication.indirect.bind.server.admin.dn=		
	Not required, comment out.	Set to the username/email address of a user with search privileges on the external LDAP.
conf_security_externalized.authentication.indirect.bind.server.admin.password=		

	Not required, comment out.	Set to the password for the above user.
`conf_security_externalized.authentication.indirect.bind.server.admin.password.encrypted=`		
	Not required, comment out.	Set to "**`false`**" if using a plain-text password (NOT RECOMMENDED) Set to "**`true`**" if using an encrypted password (RECOMMENDED)
`conf_security_externalized.authentication.server.url=`		
	Set to "**`ldap://localhost:389`**", replacing "`localhost`" with the IP or domain for your external LDAP instance.	
`conf_security_externalized.authentication.server.version=`		
	Set to your external LDAP server version, e.g. "3".	
`conf_security_externalized.authentication.server.conn.timeout=`		
	Set to a timeout (number in milliseconds) that is appropriate for your external LDAP.	
`conf_security_externalized.authentication.user.store.baseDN=`		
	Set to the baseDN value to match your external LDAP service. This value will be provided by your external LDAP administrator. E.g. in Apigee we might use "**`DC=apigee,DC=com`**"	
`conf_security_externalized.authentication.user.store.search.query=(&(${userAttribute}=${userId}))`		
	Do not change this search string. It is used internally.	
`conf_security_externalized.authentication.user.store.user.attribute=`		
	This identifies the external LDAP property you want to bind against. Set to whichever property contains the username in the format that your users use to log into Apigee Edge. For example: If users will log in with an email address and that credential is stored in "**`userPrincipalName`**", set above to "**`userPrincipalName`**".	

	If users will log in with an ID and that is stored in "**sAMAccountName**", set above to "**sAMAccountName**".
conf_security_externalized.authentication.user.store.user.email.attribute=	
	This is the LDAP attribute where the user email value is stored. This is typically "**userPrincipalName**" but set this to whichever property in your external LDAP contains the user's email address that is provisioned into Apigee's internal authorization LDAP.

B. Understanding Edge authentication and authorization flows

This section explains how authentication and authorization work on Apigee Edge. This information may provide useful context when you configure an external LDAP with Apigee Edge.

The authentication and authorization flows depend whether a user authenticates through the management UI or through the APIs.

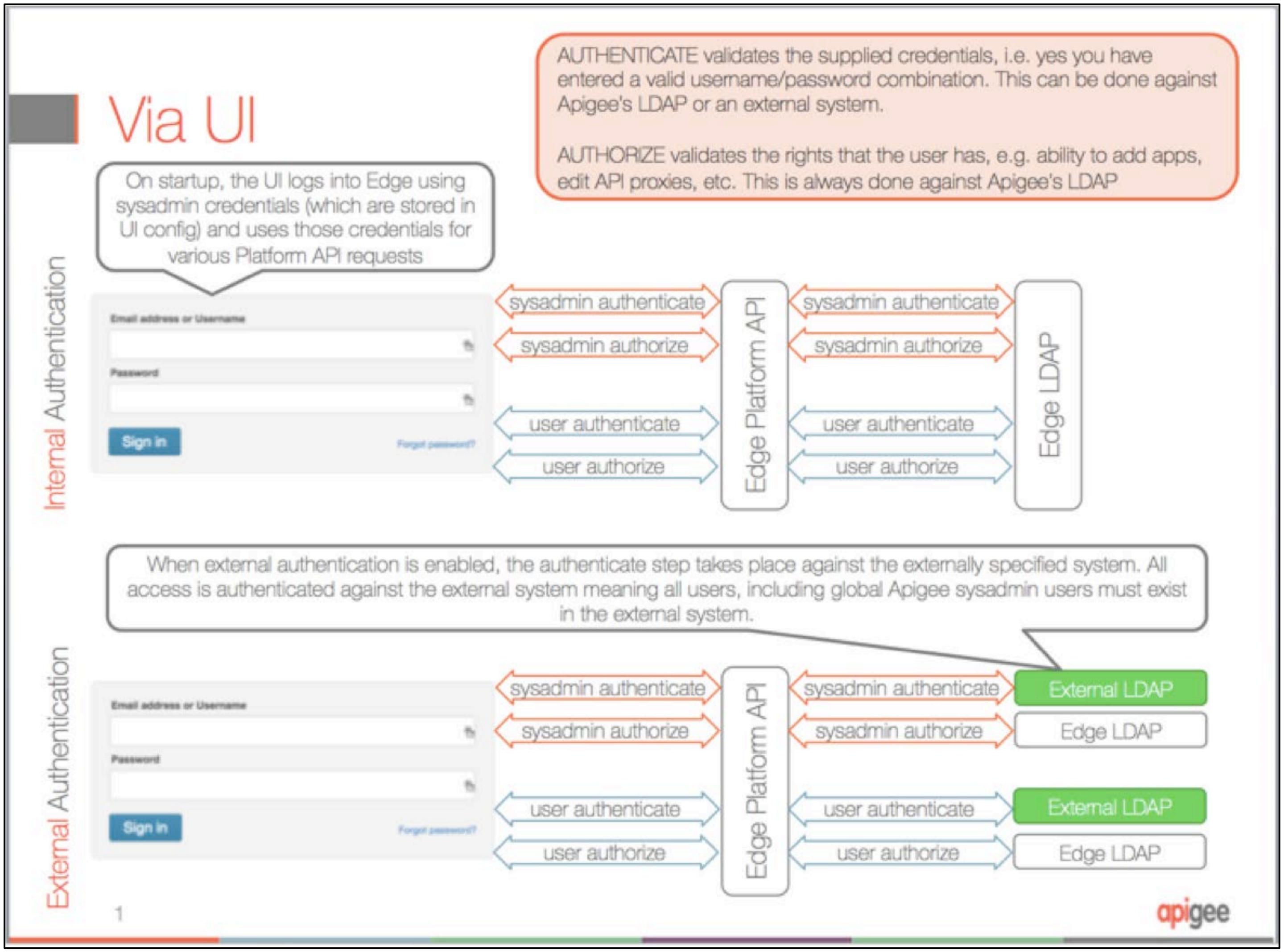
When logging in through the UI

When you log in to Edge through the UI, Edge performs a separate login step to the Apigee Management Server using the Edge global system administrator credentials.

The following UI login steps are illustrated in Figure 1:

1) The user enters login credentials in the login UI.
2) Edge logs in to the Management Server using the global system admin credentials.
3) The global system admin credentials are authenticated and authorized. The UI uses these credentials to make certain platform API requests.
    a. If external authentication is enabled, the credentials are authenticated against the external LDAP, otherwise, the internal Edge LDAP is used.
    b. Authorization is always performed against the internal LDAP.
4) The credentials entered by the user are authenticated and authorized.
    a. If external authentication is enabled, the credentials are authenticated against the external LDAP, otherwise, the internal Edge LDAP is used.
    b. Authorization is always performed against the internal LDAP.

Figure 1: Authorization and authentication via the Edge UI



Via UI

AUTHENTICATE validates the supplied credentials, i.e. yes you have entered a valid username/password combination. This can be done against Apigee's LDAP or an external system.

AUTHORIZE validates the rights that the user has, e.g. ability to add apps, edit API proxies, etc. This is always done against Apigee's LDAP

On startup, the UI logs into Edge using sysadmin credentials (which are stored in UI config) and uses those credentials for various Platform API requests

When external authentication is enabled, the authenticate step takes place against the externally specified system. All access is authenticated against the external system meaning all users, including global Apigee sysadmin users must exist in the external system.

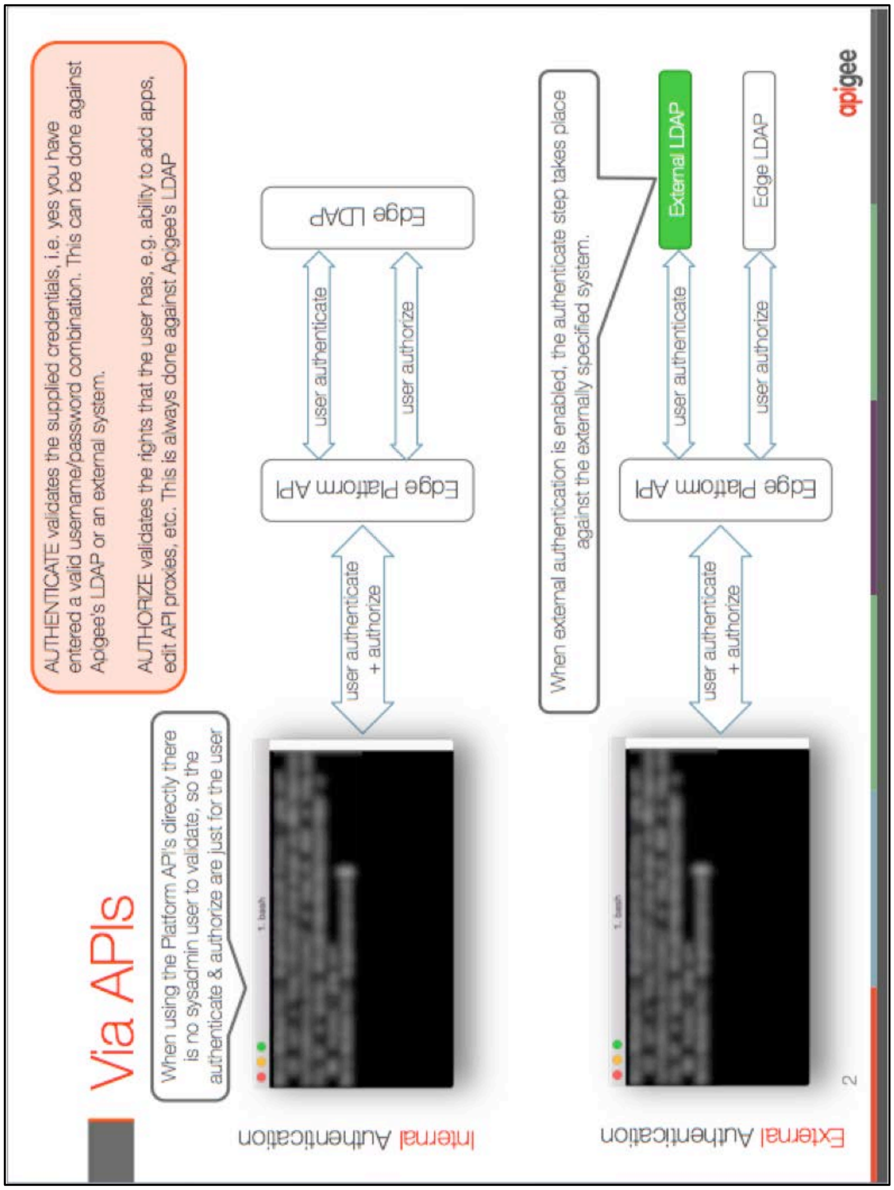
When logging in through APIs

When you log in to Edge through the an API, only the credentials entered with the API are used. Unlike with UI login, a separate login with system admin credentials is not needed.

The following API login steps are illustrated in Figure 2:

1) The user enters login credentials in the login UI.
2) The credentials entered by the user are authenticated and authorized.
3) If external authentication is enabled, the credentials are authenticated against the external LDAP, otherwise, the internal Edge LDAP is used.
4) Authorization is always performed against the internal LDAP.

Figure 2: Authorization and authentication via the Edge APIs



Via APIs

When using the Platform API's directly there is no sysadmin user to validate, so the authenticate & authorize are just for the user

AUTHENTICATE validates the supplied credentials, i.e. yes you have entered a valid username/password combination. This can be done against Apigee's LDAP or an external system.

AUTHORIZE validates the rights that the user has, e.g. ability to add apps, edit API proxies, etc. This is always done against Apigee's LDAP

When external authentication is enabled, the authenticate step takes place against the externally specified system.

C. Managing Edge organization users

This appendix provides a quick overview on how to add users to Edge through both the UI and APIs. As an Edge admin, you'll need to provision users that exist in the external LDAP into Edge. You need to do this because any user who wants to connect to and use the Apigee Edge management APIs and/or management UI must be added in both the external authentication system and in Apigee Edge.

An Edge organization administrator can add users to Edge and assign roles to the users either through the UI. A global system admin can perform these functions through using Edge APIs.

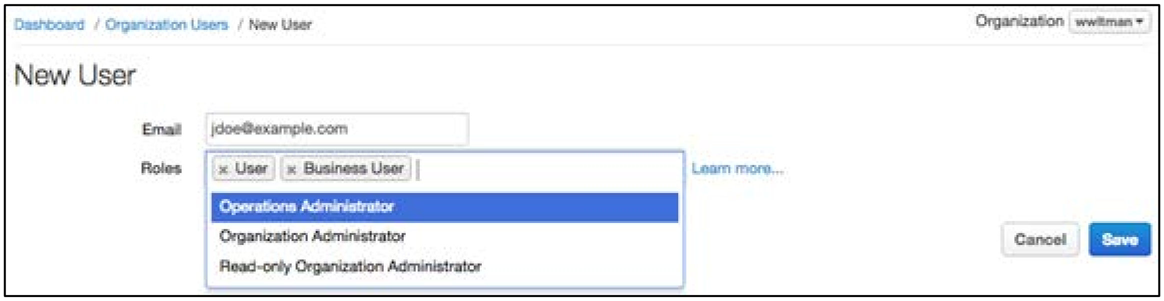
Note: When you add, change, or delete a user in the external authentication system, an Edge organization administrator must also mirror those user changes in Apigee Edge.

Adding users to an organization through the UI

An Edge organization admin selects **Organization User** from the Admin menu to bring up the Add User dialog. The following figure shows the UI for adding a user. Note that you cannot add a user to an Edge org through the UI without assigning a role. The only credential required is the user's email address.

Important: The user you add must also exist in the external LDAP to ensure that the user can be authenticated when logging in to the UI.

Figure 3. Assigning a role to a user



Dashboard / Organization Users / New User

New User

Email: jdoe@example.com

Roles: User, Business User

Adding users through the API

An Edge global admin can add users to Apigee Edge through the APIs.

For more details, see "Organization and Environment Maintenance" in the *Apigee Edge for Private Cloud Operations Guide* for information on adding and configuring users.

Here we demonstrate how to add an organization user and assign a role to the user through the Edge APIs.

1) Extract user information from the external LDAP system. You'll need the email address, first name, and last name. Also, it's recommended that you note which roles the user was assigned -- this information may help when you assign Apigee Edge roles to the user.

2) Execute the following Apigee Edge management API calls using curl or a REST tool like Postman.

Note: Be sure to use your global system administrator credentials when making this call.

```
POST http://<message-processor-ip>:8080/v1/users \
-u <global-sysadmin-username> \
{
  "emailId" : "jdoe@mydomain.com",
  "firstName" : "Jane",
  "lastName" : "Doe",
  "password" : "Mypass123"
}
```

3) **Important:** The **password** specified in the preceding call is required, but it is not used when external authentication is employed. All you need to do is pass in a dummy password. This password will not be used in the external authentication use case (because the authenticated password resides on the external LDAP). However, it will be used if you switch back from external authentication to the default Apigee LDAP.

4) Map a role between the role(s) you noted from the external LDAP system to the roles defined in Apigee Edge. The default Edge roles are:

   - orgadmin (Organization Administrator)
   - user (User)
   - businessuser (Business User)
   - opsadmin (Operations Administrator)

   For more details, see "Managing organization users" and "Assigning roles" in the Apigee Edge documentation. The docs include a detailed reference page for each of the built-in roles. Also, refer to the "Organization and Environment Maintenance" in the Apigee Edge for Private Cloud Operations Guide for information on configuring roles.

5) To do the role mapping, make the following management API call (with curl, Postman, or another tool):

```
POST
http://192.168.56.50:8080/v1/users/rajeshmishra@mydomain.com/userroles \

Authorization : Basic b3Bka0BhcGlnuY29tOkFwaWlZTEyMw== \

{
"role":[{"name":"orgadmin","organization":"org1"}], \

"user":{"emailId":"jdoe@mydomain.com","firstName":"Jane", \
"lastName":"Doe"}
}
```

Testing the configuration

1) Open the management UI in a browser at:

```
http://<management-server-IP>:9000/
```

For example:

```
http://192.168.52.100:9000/
```

2) Log in and go to the **Admin > Organization Users** page.
3) Verify that the user was added.

10 Almaden Boulevard, 16th Floor, San Jose
CA 95113
USA

No. 17/2, 2B Cross, 7th Main,
2 & 3 Floor, Off 80 Feet Road, 3rd Block
Koramangala, Bangalore 560034
INDIA

3 Sheldon Square
London W2 6HY
UK

www.apigee.com