Apigee™

Apigee Edge On-Premises Deployment Kit

v4.24.1.0

Nov 26, 2013

Developer Channel Services Installation and Configuration Guide

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

260 Sheridan Avenue
Palo Alto
CA 94306
Call +1 (408) 343-7300
www.apigee.com

UK

One Kingdom Street, 4th Floor
Paddington Central
London W2 6BD
Call: +44 (0) 750 123 2390
www.apigee.com/

Contents

Overview

Apigee Developer Channel Services is a template portal for content and community management. It is based on the open source Drupal (www.drupal.org) project. The default setup allows creating and managing API documentation, forums, and blogs. A built-in test console allows testing of APIs in real time from within the portal.

Apart from content management, Developer Channel Services has various features for community management such as manual/automatic user registration and moderating user comments. Role-Based Access Control (RBAC) model controls the access to features on the Developer Channel Services. For example, you can enable controls to allow registered user to create forum posts, use test consoles, and so on.

Supported Network Topologies

The components of the Apigee Developer Channel Services can be installed in several different network configurations, or *topologies*, including:

- **Single machine** - All Apigee Developer Channel Services components installed on a single machine.
- **Small** - The Drupal MySQL database is installed on a separate machine from the other components.
- **Medium** - Multiple Apache Servers installed with a sync script to keep them updated.
- **Large** - Multiple Apache Servers installed with NFS, GlusterFS, S3, or other type of syncing mechanism.

Note: These four topologies are the only topologies supported by Apigee. If you use a different network topology, Apigee might not be able to support it.

The following figure shows the single-machine topology where all components of the Apigee Developer Channel Services are installed on a single machine:

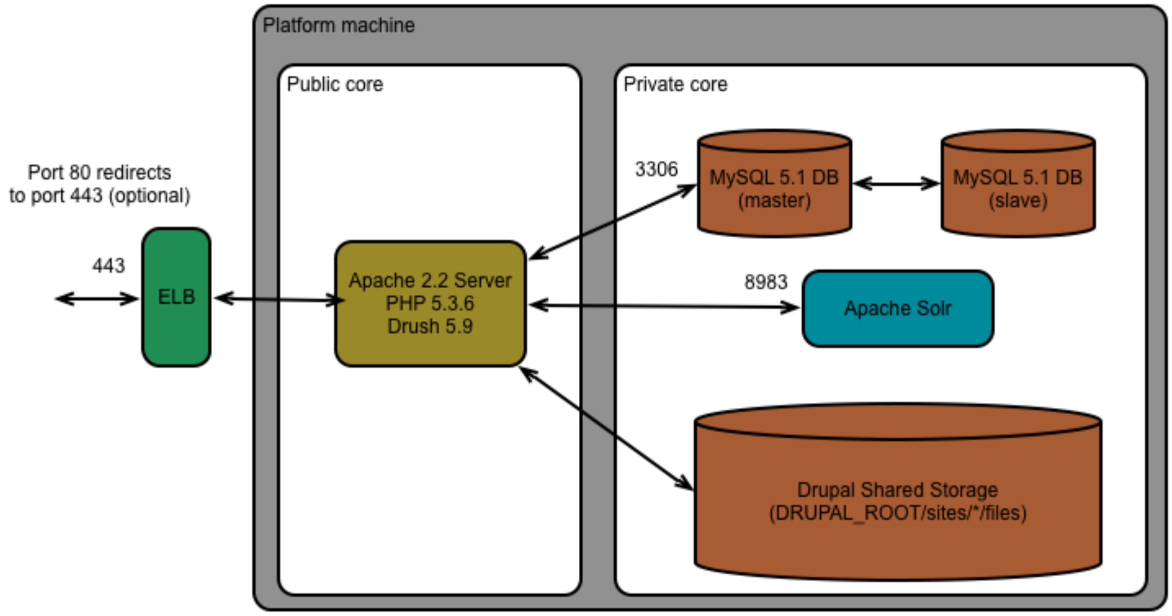


Figure 1: Single-machine topology

In this figure, the Public core contains the components that are publicly accessible. The Private core contains components that are not publicly accessible.

Table 2: Apigee Developer Channel Services components

Component	Description	Installed by
ELB	An Enterprise Load Balancer (ELB).	Your network provider. For example, both Amazon and Rackspace provide Enterprise load balancers for use with their instances.
Apache Server Version 2.2	Depending on your topology, you can have a single Apache server instance (single machine and small) or multiple (medium and large). For medium and large, you must provide a way to synchronize the instances.	Apigee
MySQL 5.1	The database used by Drupal. For the small, medium, and large topologies, the MySQL instance is on a different machine from the other components.	Person doing the install, or connect to an existing installation.

Component	Description	Installed by
Apache Solr	The Drupal search server. Apache Solr uses the Apache Lucene search library.	Apigee
Drupa shared storage	The shared storage area used by Drupal for uploaded files, static scripts, and other information.	Apigee
Drush 5.9	The Drupal command line interface.	Person doing the install.
PHP5.3.6	Server-side scripting engine.	Apigee

Shown below is the small topology where the MySQL server resides on a separate machine:

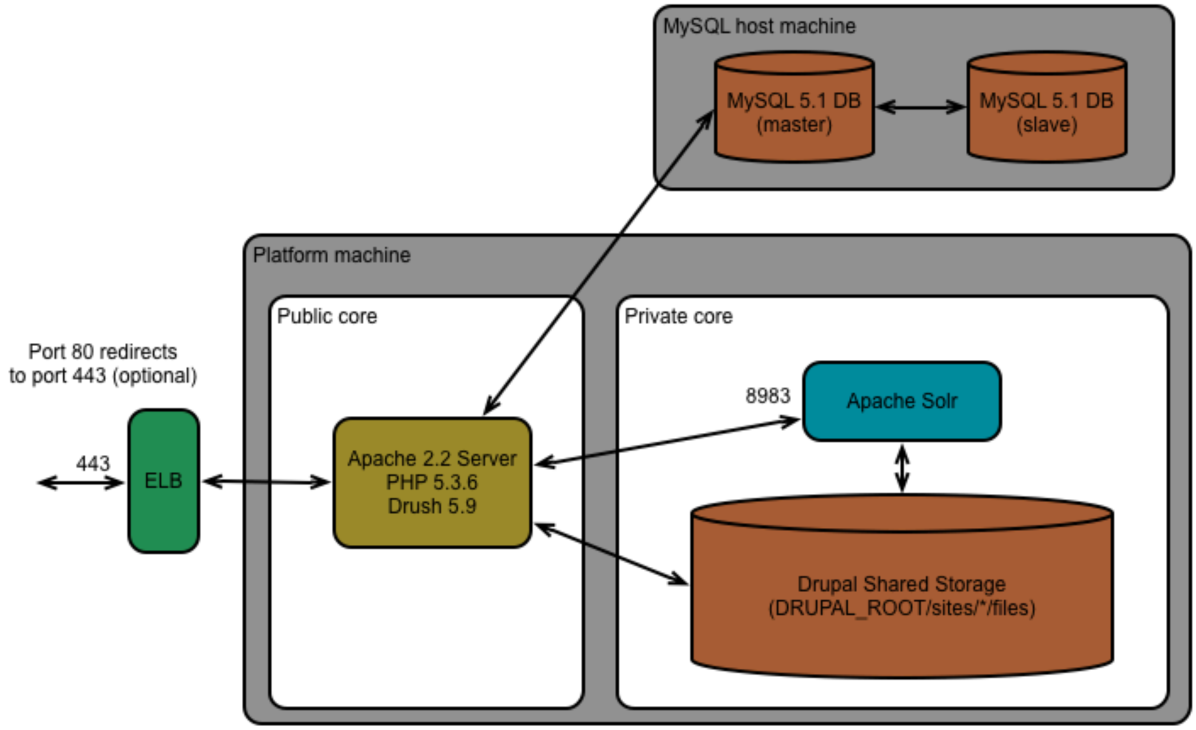

MySQL host machine
MySQL 5.1 DB (master)
MySQL 5.1 DB (slave)
Platform machine
Public core
Private core
Port 80 redirects to port 443 (optional)
443
ELB
Apache 2.2 Server PHP 5.3.6 Drush 5.9
8983
Apache Solr
Drupal Shared Storage (DRUPAL_ROOT/sites/*/files)

Figure 2: Small topology

In the medium topology, you have multiple Apache Server instances:

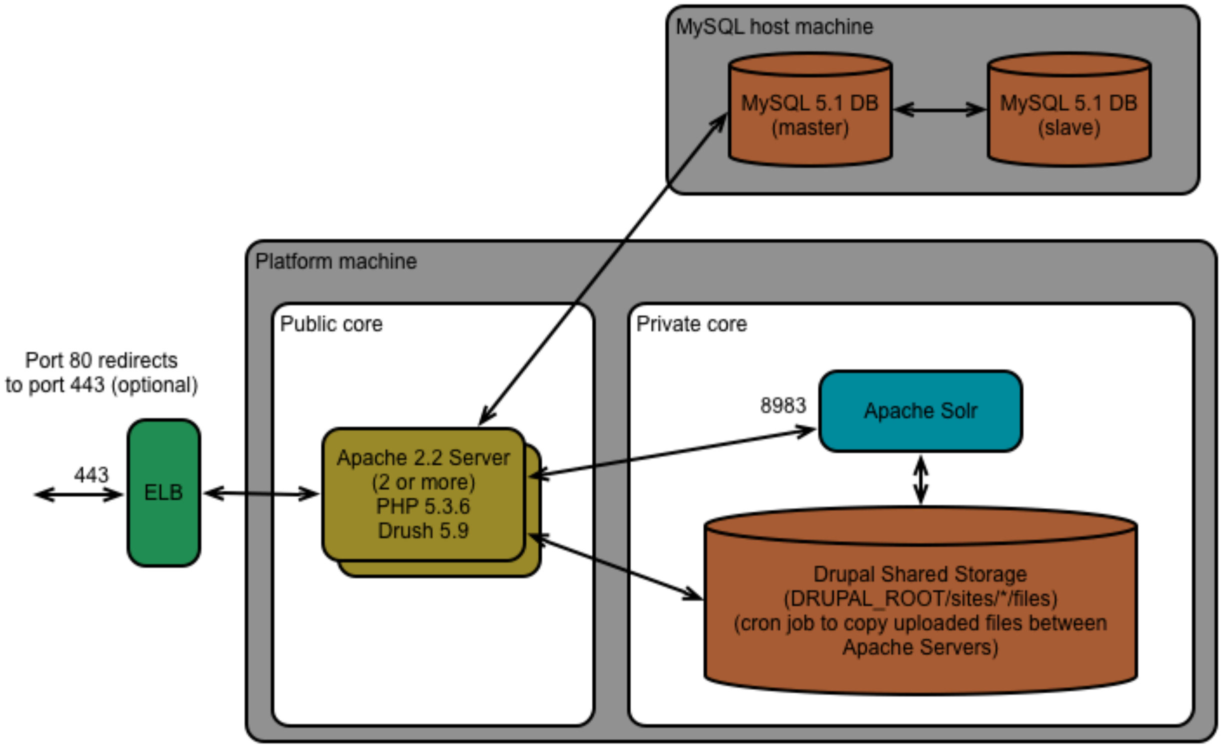

MySQL host machine
MySQL 5.1 DB (master)
MySQL 5.1 DB (slave)
Platform machine
Public core
Private core
Port 80 redirects to port 443 (optional)
443
ELB
Apache 2.2 Server (2 or more) PHP 5.3.6 Drush 5.9
8983
Apache Solr
Drupal Shared Storage (DRUPAL_ROOT/sites/*/files) (cron job to copy uploaded files between Apache Servers)

Figure 3: Medium topology

In the large topology, use NFS, GlusterFS, S3, or other type of syncing mechanism:

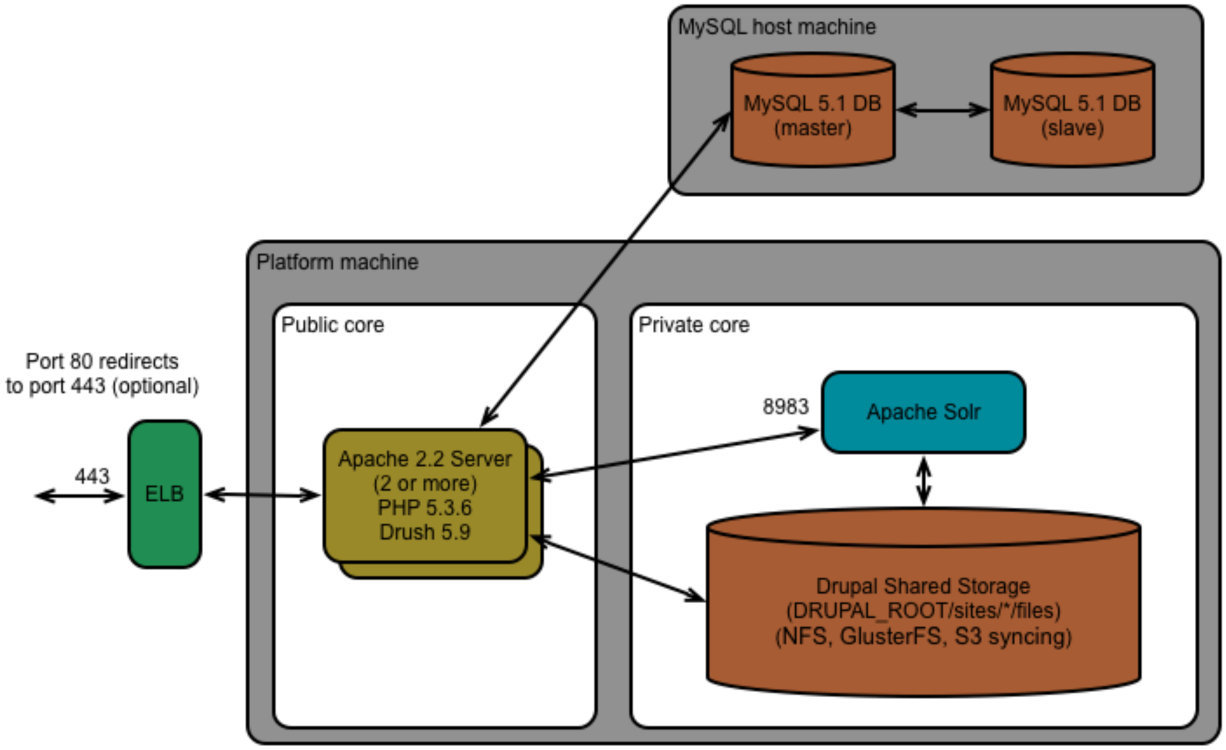

MySQL host machine
MySQL 5.1 DB (master)
MySQL 5.1 DB (slave)
Platform machine
Public core
Private core
Port 80 redirects to port 443 (optional)
443
ELB
Apache 2.2 Server (2 or more) PHP 5.3.6 Drush 5.9
8983
Apache Solr
Drupal Shared Storage (DRUPAL_ROOT/sites/*/files) (NFS, GlusterFS, S3 syncing)

Figure 4: Large topology

Requirements

Following are the hardware and software requirements for installation.

Table 3: Requirements

Hardware	Requirement
Operating system	• CentOS version 6.54 • Licensed copy of Red Hat Enterprise Linux (RHEL) version 6.5. Red Hat requires a license to download and install all required RPMs. • Other OSes might work but are not supported
RAM	1 GB
Hard disk	10 GB
Network interface	Active internet connection

MySQL Database Requirements

If you choose to install the MySQL Server on the same machine as the Developer Channel Services, the single-machine topology, you must create the database and database user. A schema will be created for you.

If you would like to have the database on a separate system, you will need to have an empty database schema already created on that server. You will need to supply the database hostname, database name, and user information to connect the database and create default database tables and data. This user will also be used by Developer Channel Services to connect to the database, and must have the following privileges: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES.

Red Hat Enterprise Linux (RHEL) Requirements

RHEL has extra requirements due to a subscription needed to access software downloads from Red Hat. The server must be able to connect to the Internet to download RPMs via yum. If using RHEL, the server must be registered on the Red Hat Network (RHN) and registered to the server optional channel. The Red Hat requirements are checked during the install and will prompt you with information if the server is not already configured properly.

Additional Requirements

In order to perform the installation, the user installing the software must have root access.

Deployment Architecture Requirements

Developer Channel Services has a single interface with the Apigee Management Server via a REST API in order to store and retrieve information about a user's applications. Developer Channel Services will need to be able to connect to the Management Server via HTTP or HTTPS, depending on your installation.

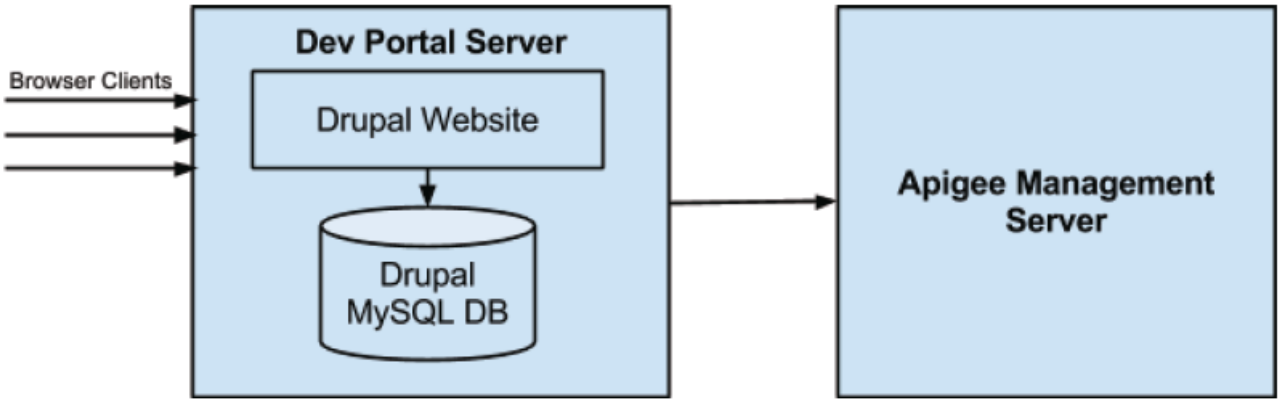


Figure 5: Dev Portal HTTP/S connection flow

Information Required Before You Start the Install

Before starting the install, you must have the following information available:

1. Which platform are you configuring: Red Hat or CentOS? If this is a Red Hat install, the machine must be registered on the Red Hat Network to download RPMs.
2. Do you plan on installing MySQL on the local machine? Some HA installations require MySQL to be on a different machine than the one serving the portal web pages. If this is the case, do not install MySQL locally. If you want a simple install with everything on the same machine, then install MySQL locally.
3. If you intend to access a remote MySQL server, the hostname, port, database name, username, and password of the remote MySQL server. The remote MySQL server should already be configured before you start the installation.
4. What is the fully-qualified domain name of the web server? (This information will be added to /etc/hosts.) This should be an IP address or hostname, such as portalserver.example.com. The default value is localhost.
5. Do you want to use Apigee's default configuration for Apache's virtual host? One virtual host is created with the hostname you've specified above in Step 7.
6. There are three pieces of information that allow your portal to communicate with the Apigee API Management server. This information is as follows:

   a. **URI of the Apigee Management API Endpoint:** This will be either a hostname or an IP address. This is the REST endpoint to which all calls are made to create apps and register developers for app keys. The default endpoint is https://api.enterprise.apigee.com/v1.

b. **Apigee Organization name**: There is a 1-1 relationship between portals and API organizations. You will set up the default organization when you set up the Management API Endpoint. The default value is my-org.

c. **Username and password for the management API endpoint**: In order for the portal to register developers and make requests for keys, the calls themselves to the management API have to be authenticated as an administrator for the organization. The Authentication value is a UN/PW key pair for administrators on the Organization you've created for the management API.

For example: dc_devportal+ORGNAME@apigee.com:MyP@ssw0rd

Installation (with Internet access)

Follow these steps to install Developer Channel Services with an Internet connection.

1. Get the Developer Channel Services bundle

Go to http://community.apigee.com/content/apigee-customer-support and select Login to your Support Portal to request the Developer Channel Services .tar file in one of two ways:

- Request access to the OPDK download directory.
- Request a URL to the Developer Channel Services RPM.

Note: If you do not have an account on the Support Portal, select Login to your Support Portal and then on the sign in page select In a hurry? Raise a support ticket here.

2. Install Developer Channel Services

1. Create a directory for the install files, such as opdk.
2. CD to the directory and download the Developer Channel Services .tar file from the link sent to you by Apigee. The downloaded file uses the naming format:

   DeveloperServices-4.x.y.tar

   where **x.y** corresponds to the version number.
3. Extract the contents of the .tar file by using the command:

   ```
   > tar -xvf <download file>.tar
   ```

   The directory now contains the Developer Channel Services RPM file, and a PDF and RTF file containing the installation instructions. The RPM file uses the following naming convention:

   apigee-drupal-4.x.y.noarch
4. Ensure that you are performing the install on Red Hat Enterprise Linux 6.5 or CentOS 6.54.
5. Ensure that networking is enabled by checking that the ONBOOT property is set to "yes" in /etc/sysconfig/network-scripts/ifcfg-eth0. If you have to change this property, run the

following command:

```
> service network restart
```

6. Enable repos that contain the LAMP software:

    a. For a Red Hat installation, if you have not done so already, register your installation with Red Hat Network by using the command:

    ```
    > subscription-manager register --username=my_username --
    password=my_password --auto-attach
    ```

    Add the server optional channel by using the commands:

    ```
    > yum install yum-utils
    > yum-config-manager --enable rhel-6-server-optional-rpms
    ```

    b. For both CentOS and RedHat, enable the EPEL (Extra Packages for Enterprise Linux) repo:

    ```
    > rpm -
    Uvh  http://dl.iuscommunity.org/pub/ius/stable/Redhat/6/x86_64
    /epel-release-6-5.noarch.rpm
    ```

7. Install the Developer Channel Services RPM:

    ```
    > yum install -y apigee-drupal-4.x.y.noarch.rpm
    ```

    You will see updates downloaded and installed automatically from Red Hat Subscription Management.

8. Enable SELinux outgoing connection policy:

    ```
    > setsebool -P httpd_can_network_connect 1
    ```

9. Add the following lines in red to /etc/sysconfig/iptables:

    ```
    # Firewall configuration written by system-config-firewall
    # Manual customization of this file is not recommended.
    *filter
    :INPUT ACCEPT [0:0]
    :FORWARD ACCEPT [0:0]
    :OUTPUT ACCEPT [0:0]
    -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
    -A INPUT -p icmp -j ACCEPT
    -A INPUT -i lo -j ACCEPT
    -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
    -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
    ```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Then, restart iptables:

```
> sudo service iptables restart
```

10. Start Apache:

```
> service httpd start
```

11. If you want to run MySQL server locally, you must install MySQL, set the root password, and create a MySQL user. Alternatively, you can connect to an existing MySQL installation on a different machine accessible on your network.

To install MySQL locally:

```
> yum install mysql-server
> service mysqld start
```

To set the root password:

```
> /usr/bin/mysqladmin -u root password 'foo'
> /usr/bin/mysqladmin -u root -h 127.0.0.1 -pfoo
```

12. Create a database and optionally a database user for Developer Channel Services to use:

```
> sudo mysql -pPWORD
```

where PWORD is the root password.

If you want to create a new user for the MySQL connection, use the following commands:

```
mysql> CREATE USER 'devportal_user'@'127.0.0.1' IDENTIFIED BY
'devportal_password';
mysql> GRANT ALL PRIVILEGES ON * . * TO
'devportal_user'@'localhost';
```

To verify that the user was created:

```
mysql> select user,host from mysql.user;
```

Create the MySQL database:

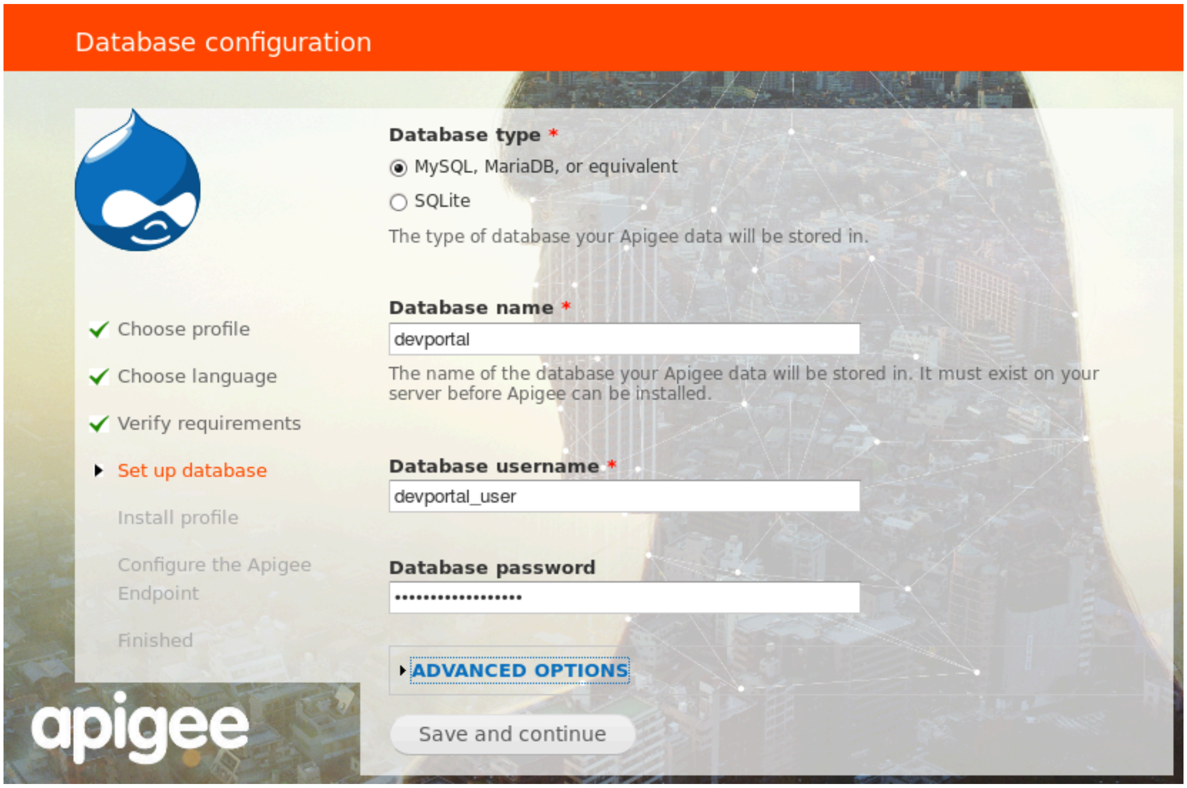
```
mysql> CREATE DATABASE devportal;
mysql> GRANT ALL ON devportal.* TO 'devportal_user'@'%' IDENTIFIED
BY 'devportal_password';
```

If MySQL is on localhost, use 'localhost' instead of '%'.

13. Edit /etc/php.ini and set the date.timezone to the correct timezone name. For example, on the American East Coast, set this to `America/New_York`. In California, set it to `America/Los_Angeles`. It must be a valid Unix timezone name with no spaces.

14. With Apache started, navigate to the server URL in a browser: http://localhost. Typically, you will have already configured a hostname and registered it with your DNS server so that you do not have to use http://localhost.

    You are redirected to the Apigee profile Database configuration page:



Database configuration

Database type *

● MySQL, MariaDB, or equivalent

○ SQLite

The type of database your Apigee data will be stored in.

Database name *

devportal

The name of the database your Apigee data will be stored in. It must exist on your server before Apigee can be installed.

Database username *

devportal_user

Database password

••••••••••••••••••

Use this page to configure the connection to a local instance of MySQL. If MySQL is on a remote machine, expand the Advanced Options section and enter the connection information to that machine, then Select Save and continue.

On the Configure the Apigee Endpoint page, enter the following information, and then select Save:

Dev Portal Org:
Dev Portal Endpoint URL:
Endpoint Authenticated User:
Authenticated Users' Password:

For more information on these values, see Information Required Before You Start the Install.

You can now log into your portal. For more information on configuring your portal, see the Apigee documentation at:

http://apigee.com/docs/developer-channel/content/what-developer-portal

3. Install Drush

Drush is the command line interface to Drupal. Apigee requires that you install Drush 5.9, which is available for download at https://github.com/drush-ops/drush. That site lets you download Drush in various forms (ZIP, tar, GitHub clone) and also includes installation instructions.

Note: Drush requires PHP, so you must install Developer Channel Services, which installs PHP, before installing Drush.

You can also use the following procedure to install Drush from a command line:

```
1. yum install php-pear
2. pear channel-update pear.php.net
3. pear upgrade -f -a PEAR Console_Color Console_Table
4. pear config-set auto_discover 1
5. pear channel-discover pear.drush.org
6. pear config-set preferred_state stable
7. pear install -a drush/drush-5.9.0
8. drush dl registry_rebuild
```

Note that you might see warnings from step 3. You can ignore those warnings.

4. Create admin users

Use drush to create administrative users for your Developer Channel Services installation. CD to your drupal_root/sites/default directory and enter the following commands, replacing the commands in “< >” with your values:

```
> cd <root directory of drupal install>/sites/default

> drush user-create <username> --mail=”<email-address>”
--password=”<password>”

> drush user-add-role Administrator <username>
```

Configuring SSL with Load Balancers

For better performance, load balancers are sometimes configured to perform SSL termination. With SSL termination, load balancers decrypt messages sent over https and forward the messages to backend servers over http. That saves backend servers the overhead of decrypting https messages themselves.

If load balancers forward unencrypted http messages to servers in the same data center, security is not an issue. However, if load balancers forward messages over http to servers outside the data center, such as your Apigee developer portal, the messages are unencrypted, which opens a security hole.

If your developer portal sits behind load balancers that are using SSL termination, and you want all traffic served over https, the website pages will need to contain https links only and you will need to add the following code to your developer portal sites/default/settings.php file. Because the load balancer does not automatically transform the content of the HTML pages, the code ensures that all links passed to the client start with https://.

To configure SSL with load balancers, add the following lines to the sites/default/settings.php file:

```
$can_detect_ssl = FALSE;

if (isset($_SERVER['HTTP_X_FORWARDED_PROTO']) ||
(isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == 'on')) {

 $can_detect_ssl = TRUE;

}

if (isset($_SERVER['HTTP_X_FORWARDED_PROTO']) &&
strtolower($_SERVER['HTTP_X_FORWARDED_PROTO']) == 'https') {

 $_SERVER['HTTPS'] = 'on';

}

if ($can_detect_ssl && $_SERVER['HTTPS'] != 'on') {

 header('Location: https://' . $_SERVER['SERVER_NAME'] .
$_SERVER['REQUEST_URI']);

 exit;

}
```

Configuring Monetization Services with Developer Channel Services

As an API provider, you can take advantage of the Monetization Services features in the developer portal or integrate Monetization Services features into your own developer portal. The Developer Channel Services can include Monetization Services-related content that an API provider publishes for viewing by a developer, such as a catalog of available API packages and rate plans for each package.

Following are the steps to configure Monetization Services using Drush (a command line shell and scripting interface for Drupal):

1. Setup organization id/name, Management Server endpoint and credentials.

   - Set organization id/name

     ```
     $ drush vset devconnect_monetization_org [org]
     ```

For example,

```
$ drush vset devconnect_monetization_org sampleorg
```

- Set endpoint (endpoint must end with a slash):

```
$ drush vset devconnect_monetization_endpoint [endpoint URL]
```

For example,

```
$ drush vset devconnect_monetization_endpoint
https://mgmt.provider.com/v1/
```

- Set endpoint credentials

```
$ drush vset devconnect_monetization_curlauth [user:password]
```

For example,

```
$ drush vset devconnect_monetization_curlauth admin:secret1234
```

2. Install Monetization Service modules

```
$ drush pm-enable devconnect_monetization
```

```
$ drush pm-enable devconnect_monetization_payment
```

3. Clear cache

```
# drush cc all
```

Prepaid Billing

Monetization Services supports both prepaid and postpaid billing. A prepaid developer pays in advance for the use of your API products. Funds are deducted from a prepaid developer's balance when the API product is used. If you want prepaid billing type, you need to enable the payment method.

Note: By default, Monetization Services is pre-integrated with *Worldpay* as payment processor. If you want to use different processor, the flow can be different.

Following are the steps to enable payment method using Developer Channel Services:

1. Log into your developer portal using administrator username and password
2. Navigate to Store → Configuration → Payment Methods and click “Payment via Worldpay”
3. In Actions list select “Enable payment method: Payment via Worldpay”
4. Fill form with below valid fields and leave other fields intact, then click Save:

   INSTALLATION ID

   SITE ID

   SECRET KEY

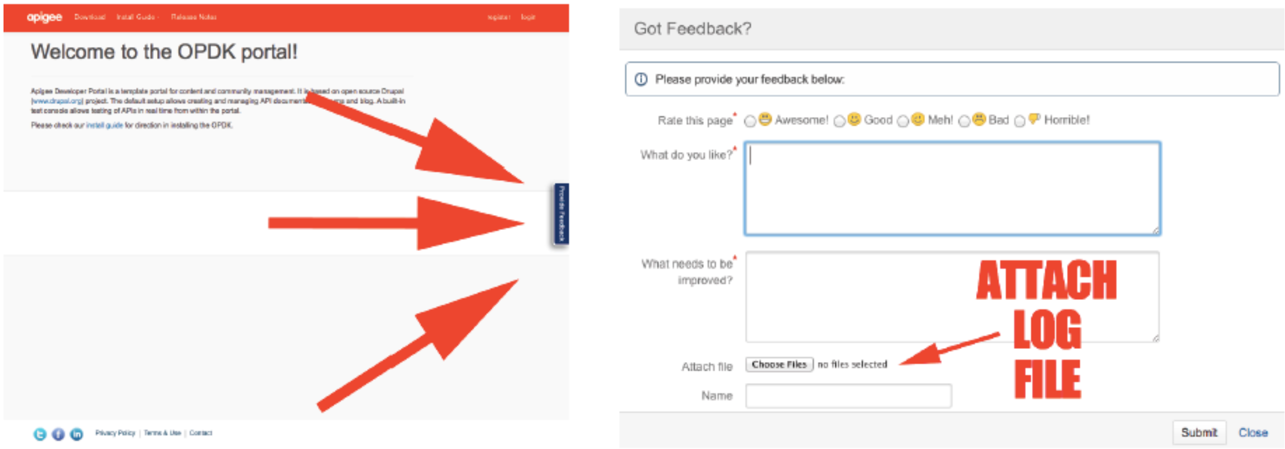
   Note: As an API provider, you must obtain above details from your payment provider.

Contact us

For the most up-to-date information and to report bugs, please refer to the on-premises Developer Channel Services website.

https://opdk.apiportal.apigee.com

If the installer exits prematurely because of an error, it will give you a log file. Please submit this log file with any bugs you report.



Welcome to the OPDK portal!

Apigee Developer Portal is a template portal for content and community management. It is based on open source Drupal (www.drupal.org) project. The default setup allows creating and managing API documentation, forums and blog. A built-in test console allows testing of APIs in real time from within the portal.

Please check our install guide for direction in installing the OPDK.

Got Feedback?

Please provide your feedback below:

Rate this page* ○ Awesome! ○ Good ○ Meh! ○ Bad ○ Horrible!

What do you like?*

What needs to be improved?*

Attach file no files selected

Name

Figure 6: Attaching a log file to a feedback report

260 Sheridan Ave. Palo Alto
CA 94306
USA

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

One Kingdom Street, 4th Floor
Paddington Central
London W2 6BD
UK

www.apigee.com