Installing, Configuring, and Running the Open edX Platform

Release

October 05, 2016

This guide provides instructions for using your own instance of the Open edX platform and associated applications.

This document applies to the most recent version of the Open edX Platform; that is, it applies to the *master* branch of the edX Platform.

This document also contains instructions for installing Open edX releases. The most recent release of Open edX is *Dogwood*.

For Your Information

1.1 Change Log

Date	Change
16 December 2015	Updated the default value for the `ENABLE_INSTRUCTOR_LEGACY_DASHBOARD` feature flag in the *Index of*
8 December 2015	Added the *Changing the Way Open edX Looks* section.
1 December 2015	Added the *Prohibiting Submission of Specified File Types* section.
20 November 2015	Updated the manual command that can be used to *test an enabled SAML provider*.
9 November 2015	Added *Index of EdX Feature Flags* page.
4 November 2015	Added the *Installing an XBlock* topic.
26 October 2015	Added the *Using Elastic MapReduce on AWS* topic.
23 October 2015	Updated all topics in *Configuring the Open edX Platform* to instruct users to modify the `lms.env.json` and
	Removed the instructions to install the Google Drive XBlock, as it is now included by default in the Open edX
20 October 2015	Added the *Add Keys to the LMS Configuration File* topic.
5 October 2015	Added the *Enabling Social Sharing of Courses and Certificates* section.
2 October 2015	Added the *Options for LTI Authentication and User Provisioning* section.
15 September 2015	Added the *Configuring an edX Instance as an LTI Tool Provider* section.
	Added the *Installing the Open edX Analytics Developer Stack* section.
10 August 2015	Added the *Open edX Cypress Release* section.
	Added the *Setting Up the YouTube API Key* section.
	Added the *Configuring ORA2 to Upload Files to Alternative Storage Systems* section.
6 Aug 2015	Added the *Enabling Third Party Authentication* section.
5 Aug 2015	Added the *Enabling Third Party Authentication with edX Edge* section.
16 June 2015	Added the *Enabling Custom Courses* section.
8 June 2015	Added *Enabling Open edX Search*.
	Added *Enabling Certificates*.
	Added *Enabling Badging*.
	Updated the *Setting Up the Open edX Mobile Applications* section to include configuration for push notification
28 May 2015	Added *Enabling Course and Video Licensing*.
02 Mar 2015	Updated the *Other edX Resources* to include information about the *The edX Partner Portal* and the *The Open ed*
24 Feb 2015	Updated for the *Open edX Birch Release*.
	Added the section *Configuring the Open edX Platform*.
20 Jan 2015	Added the section *Options for Installing edX Insights*.
14 Jan 2015	Added the section *Setting Up the Open edX Mobile Applications*.
07 Jun 2014	Added the section *Installing Open edX Fullstack*.
21 May 2014	The initial release of this guide, with the sections *Installing the Open edX Developer Stack* and *Running the Op*

1.2 Read Me

The *Installing and Configuring the Open edX Platform* documentation is created using RST files and Sphinx. As a member of the community, you can help update and revise this documentation project on GitHub:

    https://github.com/edx/edx-documentation/tree/master/en_us/install_operations/source

To suggest a revision, follow the GitHub Flow: fork the project, make changes in your fork, and submit a pull request back to the original project.

1.3 Other edX Resources

Course teams, researchers, developers, learners: the edX community includes groups with a range of reasons for using the platform and objectives to accomplish. To help members of each group learn about what edX offers, reach goals, and solve problems, edX provides a variety of information resources.

To help you find what you need, browse the edX offerings in the following categories.

- *The edX Partner Portal*
- *The Open edX Portal*
- *System Status*
- *Resources for Course Teams*
- *Resources for Researchers*
- *Resources for Developers*
- *Resources for Open edX*
- *Resources for Learners*

All members of the edX community are encouraged to make use of any of the resources described in this preface. We welcome your feedback on these edX information resources. Contact the edX documentation team at docs@edx.org.

1.3.1 The edX Partner Portal

The edX Partner Portal is the destination for partners to learn, connect, and collaborate with one another. Partners can explore rich resources and share success stories and best practices while staying up-to-date with important news and updates.

To use the edX Partner Portal, you must register and request verification as an edX partner. If you are an edX partner and have not used the edX Partner Portal, follow these steps.

1. Visit partners.edx.org, and select **Create New Account**.

2. Select **Request Partner Access**, then fill in your personal details.

3. Select **Create New Account**. You will receive a confirmation email with your account access within 24 hours.

After you create an account, you can sign up to receive email updates about edX releases, news from the product team, and other announcements. For more information, see *Release Announcements by Email*.

Course Team Support in the edX Partner Portal

EdX partner course teams can get technical support in the edX Partner Portal. To access technical support, submit a support ticket, or review any support tickets you have created, go to partners.edx.org and select **Course Staff Support** at the top of the page. This option is available on every page in the Partner Portal.

1.3.2 The Open edX Portal

The Open edX Portal is the destination for all edX users to learn about the edX roadmap, as well as hosting, extending the edX platform, and contributing to Open edX. In addition, the Open edX Portal provides product announcements, the Open edX blog, and other rich community resources.

All users can view content on the Open edX Portal without creating an account and logging in.

To comment on blog posts or the edX roadmap, or subscribe to email updates, you must create an account and log in. If you do not have an account, follow these steps.

1. Visit open.edx.org/user/register.

2. Fill in your personal details.

3. Select **Create New Account**. You are then logged in to the Open edX Portal.

Release Announcements by Email

To receive and share product and release announcements by email, you can subscribe to announcements on one of the edX portal sites.

1. Create an account on the Open edX Portal or the edX Partner Portal as described above.

2. Select **Community** and then **Announcements**.

3. Under **Subscriptions**, select the different types of announcements that you want to receive through email. You might need to scroll down to see these options.

4. Select **Save**.

You will now receive email messages when new announcements of the types you selected are posted.

1.3.3 System Status

For system-related notifications from the edX operations team, including outages and the status of error reports. On Twitter, you can follow @edxstatus.

Current system status and the uptime percentages for edX servers, along with the Twitter feed, are published on the edX Status web page.

1.3.4 Resources for Course Teams

Course teams include faculty, instructional designers, course staff, discussion moderators, and others who contribute to the creation and delivery of courses on edx.org or edX Edge.

edX101: Overview of Creating a Course

The edX101 course was built in Studio and is available for enrollment on edx.org. This course takes one to two hours to complete, and is designed to provide a high-level overview of the course creation and delivery process. It also highlights the extensive capabilities of the edX platform.

StudioX: Creating a Course with edX Studio

After you complete edX101, StudioX provides more detail about using Studio to create a course, add different types of content, and configure your course to provide an optimal on-line learning experience.

Documentation

Documentation for course teams is available on the docs.edx.org web page.

- Building and Running an edX Course is a comprehensive guide with concepts and procedures to help you build a course in edX Studio, and then use the Learning Management System (LMS) to run a course.

  When you are working in edX Studio, you can access relevant sections of this guide by selecting **Help** on any page.

- Using edX Insights describes the metrics, visualizations, and downloadable .csv files that course teams can use to gain information about student background and activity.

- edX Release Notes summarize the changes in each new version of deployed software.

- edX Open Learning XML Guide provides guidelines for building edX courses with Open Learning XML (OLX). Note that this guide is currently an alpha version.

These guides open in your web browser. The left side of each page includes a **Search docs** field and links to the contents of that guide. To open or save a PDF version, select **v: latest** at the lower right of the page, then select **PDF**.

Note: If you use the Safari browser, be aware that it does not support the search feature for the HTML versions of the edX guides. This is a known limitation.

Email

To receive and share information by email, course team members can:

- Subscribe to announcements and other new topics in the edX Partner Portal or the Open edX Portal. For information about how to subscribe, see *Release Announcements through the Open edX Portal*.

- Join the openedx-studio Google group to ask questions and participate in discussions with peers at other edX partner organizations and edX staffers.

Wikis and Web Sites

The edX product team maintains public product roadmaps on *the Open edX Portal* and *the edX Partner Portal*.

The edX Partner Support site for edX partners hosts discussions that are monitored by edX staff.

1.3.5 Resources for Researchers

Data for the courses on edx.org and edX Edge is available to the "data czars" at our partner institutions, and then used by database experts, statisticians, educational investigators, and others for educational research.

Documentation

The edX Research Guide is available on the docs.edx.org web page.

This guide opens in your web browser, with a **Search docs** field and links to that guide's contents on the left side of each page. To open or save a PDF version, select **v: latest** at the lower right of the page, and then select **PDF**.

Note: If you use the Safari browser, be aware that it does not support the search feature for the HTML versions of the edX guides. This is a known limitation.

Email

To receive and share information by email, researchers can join the openedx-analytics Google group to ask questions and participate in discussions with peers at other edX partner organizations and edX staffers.

Wikis

The edX Analytics team maintains the Open edX Analytics wiki, which includes links to periodic release notes and other resources for researchers.

The edx-tools wiki lists publicly shared tools for working with the edX platform, including scripts for data analysis and reporting.

1.3.6 Resources for Developers

Software engineers, system administrators, and translators work on extending and localizing the code for the edX platform.

Documentation

Documentation for developers is available on the docs.edx.org web page.

- The edX Platform Developer's Guide includes guidelines for contributing to Open edX, options for extending the Open edX platform, using the edX public sandboxes, instrumenting analytics, and testing.

- Installing, Configuring, and Running the edX Platform provides procedures for getting an edX developer stack (Devstack) and production stack (Fullstack) operational.

- Open edX XBlock Tutorial guides developers through the process of creating an XBlock, and explains the concepts and anatomy of XBlocks.

- Open edX XBlock API Guide provides reference information about the XBlock API.

- edX Open Learning XML Guide provides guidelines for building edX courses with Open Learning XML (OLX). Note that this guide is currently an alpha version.

- edX Data Analytics API provides reference information for using the data analytics API to build applications to view and analyze learner activity in your course.

- edX Platform APIs provide reference information for building applications to view course information and videos and work with user and enrollment data.

Note: If you use the Safari browser, be aware that it does not support the search feature for the HTML versions of the edX guides. This is a known limitation.

GitHub

These are the main edX repositories on GitHub.

- The edx/edx-platform repo contains the code for the edX platform.
- The edx/edx-analytics-dashboard repo contains the code for edX Insights.
- The edx/configuration repo contains scripts to set up and operate the edX platform.

Additional repositories are used for other projects. Our contributor agreement, contributor guidelines and coding conventions, and other resources are available in these repositories.

Email and Slack

To receive and share information by email, developers can join these Google groups to ask questions and participate in discussions with peers and edX staffers.

- For conversations about the code in Open edX, join edx-code.
- For conversations about running Open edX, join openedx-ops.
- For conversations about globalization and translation, join openedx-translation.

Additional Google groups are occasionally formed for individual projects.

Note: Please do not report security issues in public. If you have a concern, please email security@edx.org.

For real-time conversation about Open edX, the edX open source community managers run a Slack team. To join the team, get an invitation to Slack and follow the instructions.

Wikis and Web Sites

The Open edX Portal is the entry point for new contributors.

The edX Engineering team maintains an open Confluence wiki, which provides insights into the plans, projects, and questions that the edX Open Source team is working on with the community.

The pull request dashboard is a visualization of the count and age of the pull requests (PRs) assigned to teams at edX. Select the bars in this chart to get more information about the PRs.

The edx-tools wiki lists publicly shared tools for working with the edX platform, including scripts and helper utilities.

1.3.7 Resources for Open edX

Hosting providers, platform extenders, core contributors, and course staff all use Open edX. EdX provides release-specific documentation, as well as the latest version of all guides, for Open edX users. The following documentation is available.

- Open edX Release Notes provides information on the contents of Open edX releases.
- Building and Running an Open edX Course is a comprehensive guide with concepts and procedures to help you build a course in Studio, and then use the Learning Management System (LMS) to run a course.

  When you are working in Studio, you can access relevant sections of this guide by selecting **Help** on any page.

- Open edX Learner's Guide helps students use the Open edX LMS to take courses. This guide is available on the docs.edx.org web page. Because learners are currently only guided to this resource through the courseware, we encourage course teams to provide learners with links to this guide as needed in course updates or discussions.

- Installing, Configuring, and Running the edX Platform provides information about installing and using Devstack and Fullstack.

- The edX Platform Developer's Guide includes guidelines for contributing to Open edX, options for extending the Open edX platform, using the edX public sandboxes, instrumenting analytics, and testing.

- Open edX XBlock Tutorial guides developers through the process of creating an XBlock, and explains the concepts and anatomy of XBlocks.

- Open edX XBlock API Guide provides reference information on the XBlock API.

- EdX Open Learning XML Guide provides guidelines for building edX courses with Open Learning XML (OLX). Note that this guide is currently an alpha version.

- EdX Data Analytics API provides reference information for using the data analytics API to build applications to view and analyze learner activity in your course.

- EdX Platform APIs provide reference information for building applications to view course information and videos and work with user and enrollment data.

Note: If you use the Safari browser, be aware that it does not support the search feature for the HTML versions of the edX guides. This is a known limitation.

1.3.8 Resources for Learners

Documentation

The EdX Learner's Guide and the Open edX Learner's Guide are available on the docs.edx.org web page. Because learners are currently only guided to this resource through the courseware, we encourage course teams to provide learners with links to these guides as needed in course updates or discussions.

In a Course

All edX courses have a discussion forum where you can ask questions and interact with other students and with the course team: select **Discussion**. Many courses also offer a wiki for additional resources and materials: select **Wiki**.

Other resources might also be available, such as a course-specific Facebook page or Twitter feed, or opportunities for Google Hangouts. Be sure to check the **Course Info** page for your course as well as the **Discussion** and **Wiki** pages.

From time to time, the course team might send email messages to all students. While you can opt out of these messages, doing so means that you can miss important or time-sensitive information. To change your preferences for course email, select **edX** or **edX edge** at the top of any page. On your dashboard of current courses, locate the course and then select **Email Settings**.

From edX

To help you get started with the edX learning experience, edX offers a course (of course!). You can find the edX Demo course on the edX web site. EdX also maintains a list of frequently asked questions and answers.

If you still have questions or suggestions, you can get help from the edX support team: select **Contact** at the bottom of any edX web page or send an email message to info@edx.org.

For opportunities to meet others who are interested in edX courses, check the edX Global Community meetup group.

1.4 edX Browser Support

The edX Platform runs on the following browsers.

- Chrome
- Safari
- Firefox
- Internet Explorer

Note: If you use the Safari browser, be aware that it does not support the search feature for the edX documentation. This is a known limitation.

The edX Platform is routinely tested and verified on the current and previous version of each of these browsers. We generally encourage the use of and fully support only the latest version.

This information is updated as new major operating system and browser versions are released. All point releases are supported unless noted; occasional exceptions are based on specific bug fixes or feature updates.

1.4.1 edX Learning Management System

The following table shows operating system and browser support for the edX learning management system (LMS), which learners and course teams use to interact with course content.

	Chrome	Safari	Firefox	IE 11	IE 10
Windows 8	Yes	N/A	Yes	Yes	Yes
Mac OSX Mavericks or Yosemite	Yes	Yes	Yes	N/A	N/A

For more information about the LMS, see Building and Running an edX Course.

1.4.2 edX Studio

The following table shows operating system and browser support for edX Studio, which course teams use to build a course.

	Chrome	Safari	Firefox	IE 11	IE 10
Windows 8	Yes	N/A	Yes	Provisional	Provisional
Mac OSX Mavericks or Yosemite	Yes	Yes	Yes	N/A	N/A

For more information about Studio, see Building and Running an edX Course.

1.4.3 edX Insights

The following table shows operating system and browser support for edX Insights, which course teams use to review and download data about their courses and learners.

	Chrome	Safari	Firefox	IE 11	IE 10
Windows 8	Yes	N/A	Yes	Provisional	Provisional
Mac OSX Mavericks or Yosemite	Yes	Yes	Yes	N/A	N/A

For more information about edX Insights, see Using edX Insights.

Open edX Dogwood Release

This section describes how to install the Open edX Dogwood release.

- *What's Included in Dogwood*
- *What is the Dogwood Git Tag?*
- *Installing the Dogwood Release*
- *Upgrading from Cypress to Dogwood*

2.1 What's Included in Dogwood

The Open edX Dogwood release contains several new features for learners, course teams, and developers. See the release notes for the Open edX Dogwood Release for more details.

2.2 What is the Dogwood Git Tag?

The Git tag for the Dogwood release is `named-release/dogwood`. You use this tag to identify the version of Open edX code that is the Dogwood release.

The following Open edX Git repositories have the Git tag `named-release/dogwood`.

- edx-platform

- configuration

- cs_comments_service

- xqueue

- XBlock

- notifier

- edx-ora2

- edx-documentation

- edx-certificates

- edx-analytics-data-api-client

- edx-analytics-configuration

- edx-analytics-dashboard
- edx-analytics-data-api
- edx-analytics-pipeline

2.3 Installing the Dogwood Release

You can install the Open edX Dogwood release using *Devstack* or *Fullstack*.

Review the prerequisites and instructions for each option, and then choose the option that best meets your needs. Ensure that you install the required software to run the edX platform.

If you are upgrading from the Cypress release, see *Upgrading from Cypress to Dogwood*.

For new installations, follow these steps.

1. *Download the Vagrant Box* or *Download the BitTorrent File*.

   > **Caution:** The Vagrant boxes have a large file size (between 4 and 5 Gb). If you have a slow or unreliable Internet connection, use BitTorrent to download the Vagrant box you need.

2. *Set the OPENEDX_RELEASE Environment Variable*.

3. *Install the Vagrant Box*.

2.3.1 Download the Vagrant Box

If you have a fast and reliable Internet connection, you can download the Vagrant box directly or by running `vagrant up` when you install *Devstack* or *Fullstack*.

See the Open edX Releases Wiki page to access the latest Vagrant boxes.

See Vagrant's documentation on boxes for more information.

2.3.2 Download the BitTorrent File

You can also download the BitTorrent file for the option you selected. BitTorrent is recommended if you have a slow or unreliable data connection. You then use the BitTorrent file to download the Vagrant box. If the Internet connection is temporarily lost while you are downloading the Vagrant box through BitTorrent, you can later continue the download without data loss or corruption.

See the Open edX Releases Wiki page to access the latest Vagrant boxes.

See BitTorrent for more information.

If you download the Vagrant box through BitTorrent, you must add the box to Vagrant before continuing with the installation process.

- For Devstack installations, run the following command.

```
$ vagrant box add /{path-to-downloaded-box}/{name-of-vagrant-box} --name dogwood-devstack
```

- For Fullstack installations, run the following command.

```
$ vagrant box add /{path-to-downloaded-box}/{name-of-vagrant-box} --name dogwood-fullstack
```

2.3.3 Set the OPENEDX_RELEASE Environment Variable

Before installing the Vagrant box, you must set the value of the OPENEDX_RELEASE environment variable to the Git tag for the Dogwood release. To do so, use the Linux export command.

```
export OPENEDX_RELEASE="named-release/dogwood"
```

2.3.4 Install the Vagrant Box

When you have completed the previous steps, install the Dogwood release by following the installation instructions for *Devstack* or *Fullstack*.

2.4 Upgrading from Cypress to Dogwood

You can upgrade an Open edX instance that is running the Cypress release to the Dogwood release. EdX provides the migrate.sh script if you have a simple Cypress installation and want to upgrade it automatically. If you have a more complex or customized installation, you may need to upgrade manually.

2.4.1 Automatic Upgrading

The migrate.sh script is in the configuration repository, available here.

Note: The upgrade scripts provided are verified only for upgrading instances running the Cypress release. If you are running any other version of the Open edX Platform, the upgrade scripts might not work.

Caution: Before upgrading your Open edX instance, back up all data and configuration files. Then verify that you can restore your Open edX instance from the backup files.

On the computer or virtual machine that is running the Cypress release of Open edX, run the upgrade script for your type of installation:

- For Devstack, run ./migrate.sh -c devstack -t named-release/dogwood.
- For Fullstack, run ./migrate.sh -c fullstack -t named-release/dogwood.
- You can also run ./migrate.sh -h to see which other options the script accepts.

The script creates a temporary directory in which it upgrades Open edX, then cleans up extra files and directories when it finishes running.

After upgrading Open edX to the Dogwood release, start the LMS and Studio and verify that course content and data was migrated correctly.

2.4.2 Upgrade Process Overview

This is an overview of what happens during an upgrade from Cypress to Dogwood. The migrate.sh script implements this process. You may need to understand this process if your installation is customized in some way, or if you need to diagnose problems during the upgrade.

Upgrading Cypress to Dogwood is more involved than most Open edX release upgrades.

- Dogwood upgrades the Django framework from version 1.4 to 1.8, which changed the database migration tool from South to Django. When upgrading from Cypress to Dogwood, it's important to take special care with the database migrations.

- Dogwood upgrades Python from 2.7.3 to 2.7.10. This means virtualenvs have to be recreated.

The upgrade from Cypress to Dogwood includes these steps.

1. Applies a forum migration described on the Open edX wiki to support teams discussion filtering.

2. Updates edx-platform to the `release-2015-11-09` tag. This is the last released version that used Django 1.4.

3. Recreates the virtualenvs to use Python 2.7.10 instead of 2.7.3.

4. Migrates the database. This makes the database current with the last 1.4 code.

5. Uninstalls South so that it won't interfere with the new Django migrations.

6. Updates edx-platform to the `dogwood-first-18` tag. This is the first version of the code that used Django 1.8.

7. Applies all the initial Django migrations. This gets your database ready to use the new Django 1.8 migration mechanism.

8. Updates edx-platform to Dogwood.

9. Runs Django database migrations.

10. Runs two management commands to update records in the database: `generate_course_overview` and `post_cohort_membership_fix`.

11. Runs the forum migration again. This is to process any discussion topics that may have been created during the running of the script.

Similar steps are followed to upgrade other repositories such as xqueue.

Open edX Cypress Release

This section describes how to install the Open edX Cypress release.

- *What's Included in Cypress*
- *What is the Cypress Git Tag?*
- *Installing the Cypress Release*
- *Upgrading from Birch to Cypress*

3.1 What's Included in Cypress

The Open edX Cypress release contains several new features for learners, course teams, and developers. See the Open edX Release Notes for more details.

Note: There are several new features in the Cypress release that are available, but not enabled by default in new installations. For details, see the following topics.

- *Enabling Open edX Search*

- *Enabling Badging*

- *Enabling Custom Courses*

- *Enabling Course and Video Licensing*

3.2 What is the Cypress Git Tag?

The Git tag for the Cypress release is `named-release/cypress`. You use this tag to identify the version of Open edX code that is the Cypress release.

The following Open edX Git repositories have the Git tag `named-release/cypress`.

- edx-platform

- configuration

- cs_comments_service

- notifier

- edx-certificates

- xqueue

- edx-documentation

- edx-ora2

- XBlock

3.3 Installing the Cypress Release

You can install the Open edX Cypress release using *Devstack* or *Fullstack*.

Review the prerequisites and instructions for each option, and then choose the option that best meets your needs. Ensure that you install the required software to run the edX Platform.

If you are upgrading from the Birch release, see *Upgrading from Birch to Cypress*.

For new installations, follow these steps.

1. *Download the Vagrant Box* or *Download the BitTorrent File*.

> **Caution:** The Vagrant boxes have a large file size (about 2.5GB). If you have a slow or unreliable Internet connection, use BitTorrent to download the Vagrant box you need.

2. *Set the OPENEDX_RELEASE Environment Variable*.

3. *Install the Vagrant Box*.

3.3.1 Download the Vagrant Box

If you have a fast and reliable Internet connection, you can download the Vagrant box directly or by running `vagrant up` when installing *Devstack* or *Fullstack*.

See the Open edX Releases Wiki page to access the latest Vagrant boxes.

See Vagrant's documentation on boxes for more information.

3.3.2 Download the BitTorrent File

You can also download the BitTorrent file for the option you selected. BitTorrent is recommended if you have a slow or unreliable data connection. You then use the BitTorrent file to download the Vagrant box. If the Internet connection is temporarily lost while you are downloading the Vagrant box through BitTorrent, you can later continue the download without data loss or corruption.

See the Open edX Releases Wiki page to access the latest Vagrant boxes.

See BitTorrent for more information.

If you download the Vagrant box through BitTorrent, you must add the box to Vagrant before continuing with the installation process.

- For Devstack installations, run the following command.

```
$ vagrant box add /path-to-downloaded-box/name-of-vagrant-box --name
   cypress-devstack
```

- For Fullstack installations, run the following command.

```
$ vagrant box add /path-to-downloaded-box/name-of-vagrant-box --name
    cypress-fullstack
```

3.3.3 Set the OPENEDX_RELEASE Environment Variable

Before installing the Vagrant box, you must set the value of the OPENEDX_RELEASE environment variable to the Git tag for the Cypress release. Use the Linux export command.

```
export OPENEDX_RELEASE="named-release/cypress"
```

3.3.4 Install the Vagrant Box

When you have completed the previous steps, install the Cypress release by following the installation instructions for *Devstack* or *Fullstack*.

3.4 Upgrading from Birch to Cypress

You can upgrade an Open edX instance that is running the Birch release to the Cypress release, by using the migrate.sh script in the configuration repository, available here.

Note: The upgrade scripts provided are verified only for upgrading instances running the Birch release. If you are running any other version of the Open edX Platform, the upgrade scripts might not work.

Caution: Before upgrading your Open edX instance, back up all data and configuration files. Then verify that you can restore your Open edX instance from the backup files.

On the computer or virtual machine that is running the Birch release of Open edX, run the upgrade script for your type of installation:

- For Devstack, run ./migrate.sh -c devstack.
- For Fullstack, run ./migrate.sh -c fullstack.
- You can also run ./migrate.sh -h to see which other options the script accepts.

The script creates a temporary directory in which it upgrades Open edX, then cleans up extra files and directories when it finishes running.

After upgrading Open edX to the Cypress release, start the LMS and Studio and verify that course content and data was migrated correctly.

Open edX Birch Release

This section describes how to install the Open edX Birch release.

Note: Now that the Open edX Cypress release is available, edX no longer supports the Birch release.

4.1 What's Included in Birch

The Open edX Birch release contains several new features for students, course teams, and developers. See the Open edX Release Notes for more details.

Note: There are several new features in the Birch release that are available, but not configured in new installations. For details, see the following topics.

- *Enabling Course Prerequisites*
- *Enabling Entrance Exams*

4.2 What is the Birch Git Tag?

The Git tag for the Birch release is **named-release/birch.2**. You use this tag to identify the version of Open edX code that is the Birch release.

The following Open edX Git repositories have the Git tag **named-release/birch.2**:

- edx-platform

- configuration

- cs_comments_service

- notifier

- edx-certificates
- xqueue
- edx-documentation
- edx-ora2
- XBlock

4.3 Installing the Birch Release

You can install the Open edX Birch release using *Devstack* or *Fullstack*.

Review the prerequisites and instructions for each option, then choose the option that best meets your needs. Ensure you install the required software to run the edX Platform.

If you are upgrading from the Aspen release, see *Upgrading from Aspen to Birch*.

For new installations, follow the steps below.

1. *Download the Vagrant Box* or *Download the BitTorrent File*.

   > **Caution:** The Vagrant boxes have a large file size (about 2.5GB). If you have a slow or unreliable Internet connection, use BitTorrent to download the Vagrant box you need.

2. *Set the OPENEDX_RELEASE Environment Variable*.

3. *Install the Vagrant Box*.

4.3.1 Download the Vagrant Box

If you have a fast and reliable Internet connection, you can download the Vagrant box directly or by running `vagrant up` when installing *Devstack* or *Fullstack*.

See the Open edX Releases Wiki page to access the latest Vagrant boxes.

See Vagrant's documentation on boxes for more information.

4.3.2 Download the BitTorrent File

You can also download the BitTorrent file for the option you selected. BitTorrent is recommended if you have a slow or unreliable data connection. You then use the BitTorrent file to download the Vagrant box. If the Internet connection is temporarily lost while you are downloading the Vagrant box through BitTorrent, you can later continue the download without data loss or corruption.

See the Open edX Releases Wiki page to access the latest Vagrant boxes.

See BitTorrent for more information.

If you download the Vagrant box through BitTorrent, you must add the box to Vagrant before continuing with the installation process.

- For Devstack installations, run the following command.

```
$ vagrant box add /path-to-downloaded-box/vagrant-images-birch-2-devstack.box --name birch-d
```

- For Fullstack installations, run the following command.

```
$ vagrant box add /path-to-downloaded-box/vagrant-images-birch-2-fullstack.box --name birch-
```

4.3.3 Set the OPENEDX_RELEASE Environment Variable

Before installing the Vagrant box, you must set the value of the *OPENEDX_RELEASE* environment variable to the Git tag for the Birch release:

```
export OPENEDX_RELEASE="named-release/birch.2"
```

4.3.4 Install the Vagrant Box

When you have completed the previous steps, install the Birch release by following the installation instructions for *Devstack* or *Fullstack*.

4.4 Upgrading from Aspen to Birch

You can upgrade your Open edX instance that is running the Aspen release to the Birch release, using the `migrate.sh` script in the configuration repository, available here.

Note: The upgrade scripts provided are verified only for upgrading instances running the Aspen release. If you are running any other version of the Open edX Platform, the upgrade scripts might not work.

> **Caution:** Before upgrading your Open edX instance, back up all data and configuration files. Then verify that you can restore your Open edX instance from the backup files.

On the computer or virtual machine running the Aspen release of Open edX, run the upgrade script for your type of installation:

- For Devstack, run `./migrate.sh -t named-release/birch.2 -c devstack`.
- For Fullstack, run `./migrate.sh -t named-release/birch.2 -c fullstack`.
- You can also run `./migrate.sh -h` to see which other options the script accepts.

The script creates a temporary directory in which it upgrades Open edX, then cleans up extra files and directories when it finishes running.

After upgrading Open edX to the Birch release, run the edX Platform and verify that course content and data was migrated correctly.

Open edX Platform Installation Options

This section describes the Open edX installation options.

- *Open edX Developer Stack*
- *Open edX Fullstack*

5.1 Open edX Developer Stack

The Open edX Developer Stack, known as **Devstack**, is a Vagrant instance designed for local development.

Devstack is in the edx configuration repository on GitHub.

This guide includes the following sections about Devstack:

- *Installing the Open edX Developer Stack*
- *Running the Open edX Developer Stack*

Additional sections are planned for future versions of this guide.

See the edx configuration repository wiki for information from edX and the Open edX community about Devstack and other installation and configuration options. This wiki contains pages with more information about Devstack.

- Devstack wiki
- Developing on Devstack

Given the large number of dependencies needed to develop extensions to Open edX Insights, a separate development environment is available to support analytics development. This guide includes the following section about the analytics devstack:

- *Installing the Open edX Analytics Developer Stack*

5.2 Open edX Fullstack

Open edX Fullstack, known as **Fullstack**, is a Vagrant instance designed for deploying all edX services on a single server.

Fullstack is in the edx configuration repository on GitHub.

This guide includes *Installing Open edX Fullstack*.

See the edx configuration repository wiki for information from edX and the Open edX community on Fullstack and other installation and configuration options.

5.2.1 Ubuntu 12.04 64

You can install Fullstack on a single Ubuntu 12.04 64-bit server.

Ubuntu information is planned for future versions of this guide.

See the edx configuration repository wiki for information from edX and the Open edX community about Ubuntu and other installation and configuration options.

Installing the Open edX Developer Stack

This section describes how to install the Open edX Developer Stack.

6.1 Overview

The Open edX Developer Stack, known as **Devstack**, is a Vagrant instance designed for local development.

Devstack uses the same system requirements as *Open edX Fullstack*. This allows you to discover and fix system configuration issues early in development.

Devstack simplifies certain production settings to make development more convenient. For example, nginx and gunicorn are disabled in Devstack; Devstack uses Django's runserver instead.

See the Vagrant documentation for more information.

6.2 Components

Devstack includes the following edX components.

- The Learning Management System (LMS)

- edX Studio

- Discussion Forums

- Open Response Assessments (ORA)

- EdX Search

Devstack also includes a demonstration edX course.

6.3 Knowledge Prerequisites

To use Devstack, you should meet the following knowledge requirements.

- Understand basic terminal usage. If you are using a Mac computer, see Introduction to the Mac OS X Command Line. If you are using a Windows computer, see Windows Command Line Reference.

- Understand Vagrant commands. See the Vagrant Getting Started guide for more information.

6.4 Software Prerequisites

To install and run Devstack, you must first install the following required software.

- VirtualBox 4.3.12 or higher.

- Vagrant 1.6.5 or higher.

- A Network File System (NFS) client, if your operating system does not include one. Devstack uses VirtualBox Guest Editions to share folders through NFS.

6.5 Install DevStack

To install Devstack directly from the command line, follow the instructions below. You can also install DevStack using a Torrent file, as explained in the next section.

Before beginning the installation, ensure that you have the administrator password for your local computer. The administrator password is needed so that NFS can be set up to allow users to access code directories directly from your computer.

1. Ensure the `nfsd` client is running.

2. Create the `devstack` directory and navigate to it in the command prompt.

```
mkdir devstack
cd devstack
```

3. Download the Devstack Vagrant file.

```
curl -L https://raw.github.com/edx/configuration/master/vagrant/release/devstack/Vagrantfile > V
```

4. Install the Vagrant `vbguest` plugin.

```
vagrant plugin install vagrant-vbguest
```

5. Create the Devstack virtual machine.

```
vagrant up
```

The first time you create the Devstack virtual machine, Vagrant downloads the base box, which has a file size of about 4GB. If you destroy and recreate the virtual machine, Vagrant re-uses the box it downloaded. See Vagrant's documentation on boxes for more information.

6. When prompted, enter the administrator password for your local computer.

When you have completed these steps, see *Running the Open edX Developer Stack* to begin using Devstack.

For help with the Devstack installation, see *Troubleshooting the Devstack Installation*.

6.6 Install Devstack using the Torrent file

You can use BitTorrent to download the box file. Follow all of the instructions in *Install DevStack*, but instead of downloading the Vagrant file with curl, do this:

1. Download the Torrent file for the latest Open edX release. See the Open edX Releases Wiki page to find out what the latest Open edX release is, and where to download the Torrent file.

2. When you have the file on your computer, add the virtual machine using the following command.

```
vagrant box add box-name path-to-box-file
```

6.7 Troubleshooting the Devstack Installation

In some cases, you see an error when you attempt to create the Devstack virtual machine (`vagrant up`). For example:

```
mount.nfs:  mount to NFS server '192.168.33.1:/path/to/edx-platform' failed:
timed out, giving up
```

This error situation arises because Vagrant uses a host-only network in Virtualbox to communicate with your computer. If a network does not exist, one is created on `vagrant up`. If this network is created with the VPN up, it will not work. You must recreate the network with the VPN down.

To resolve the error, follow these steps.

1. Stop the VPN.

2. Type `vagrant halt`.

3. Open Virtualbox.

4. Navigate to **Preferences > Network > Host-only Networks** and remove the most-recently-created host-only network.

5. Type `vagrant up`.

Running the Open edX Developer Stack

This section describes how to run the Open edX Developer Stack.

7.1 Connect to the Devstack Virtual Machine

1. To connect to the Devstack virtual machine, use the SSH command from the *devstack* directory.

```
vagrant ssh
```

2. To use Devstack and perform any of the tasks described in this section, you must connect as the user **edxapp**.

```
sudo su edxapp
```

This command loads the edxapp environment from the file `/edx/app/edxapp/edxapp_env`. This puts `venv python` and `rbenv ruby` in your search path.

This command also sets the current working directory to the edx-platform repository (`/edx/app/edxapp/edx-platform`).

7.2 Set Up Ability to Preview Units (Mac/Linux Only)

If you are installing Devstack on a Linux or Macintosh computer, in order to use the preview feature in edX Studio, you must add the following line to the `etc/hosts` file.

```
192.168.33.10 preview.localhost
```

7.3 Customize the Source Code Location

You can customize the location of the edX source code that gets cloned when you provision Devstack. You may want to do this to have Devstack work with source code that already exists on your computer.

By default, the source code location is the directory in which you run `vagrant up`. To change this location, set the `VAGRANT_MOUNT_BASE` environment variable to set the base directory for the edx-platform and cs_comments_service source code directories.

7.4 Run the LMS on Devstack

When you run the LMS on Devstack, the command updates requirements and compiles assets, unless you use the `fast` option.

The command uses the file `lms/envs/devstack.py`. This file overrides production settings for the LMS.

To run the LMS on Devstack, follow these steps.

1. *Connect to the Devstack Virtual Machine*.

2. Run the following command.

```
paver devstack lms
```

Or, to start the LMS without updating requirements and compiling assets, use the `fast` option.

```
paver devstack lms --fast
```

The LMS starts.

3. Open the LMS in your browser at `http://localhost:8000/`.

Vagrant forwards port 8000 to the LMS server running in the virtual machine.

7.5 Run Studio on Devstack

When you run Studio on Devstack, the command updates requirements and compiles assets, unless you use the `fast` option.

You run Studio on Devstack with the file `cms/envs/devstack.py`. This file overrides production settings for Studio.

To run Studio on Devstack, follow these steps.

1. *Connect to the Devstack Virtual Machine*.

2. Run the following command.

```
paver devstack studio
```

Or, to start Studio without updating requirements and compiling assets, use the `fast` option.

```
paver devstack studio --fast
```

Studio starts.

3. Open Studio in your browser at `http://localhost:8001/`.

Vagrant forwards port 8001 to the Studio server running in the virtual machine.

7.5.1 View Available Studio Commands

To view all available commands for Studio, enter the following command.

```
./manage.py cms -h --settings=devstack
```

7.6 Run Discussion Forums on Devstack

To run discussion forums on Devstack, follow these steps.

1. *Connect to the Devstack Virtual Machine*.

2. Switch to the discussion forum account by entering the following command.

```
sudo su forum
```

3. Update Ruby requirements.

```
bundle install
```

Note: If you get a message for entering a password to install the bundled RubyGems to the system, you can safely exit by entering `control+c` on a Macintosh or `Ctrl+C` on Windows. The RubyGems will still be installed correctly for the forum user.

---

4. Start the discussion forums server.

```
ruby app.rb -p 18080
```

The discussions forum server starts. You can access the discussion forums API at `http://localhost:18080/`.

7.7 Default Accounts on Devstack

When you install Devstack, the following accounts are created.

Account	Description
staff@example.com	An LMS and Studio user with course creation and editing permissions. This user is a course team member with the Admin role, which gives rights to work with the demonstration course in Studio, the LMS, and Insights.
verified@example.com	A student account that you can use to access the LMS for testing verified certificates.
audit@example.com	A student account that you can use to access the LMS for testing course auditing.
honor@example.com	A student account that you can use to access the LMS for testing honor code certificates.

The password for all of these accounts is `edx`.

Installing the Open edX Analytics Developer Stack

This section describes how to install and run the Open edX Analytics Developer Stack.

- *Overview*
- *Components*
- *Install the Software Prerequisites*
- *Install the Analytics Devstack*
- *Using the Analytics Devstack*

8.1 Overview

The Open edX Analytics Developer stack, known as the Analytics Devstack, is a modified version of the *Open edX Developer Stack*.

It provides all of the services and tools needed to modify the Open edX Analytics Pipeline, Data API, and Insights projects.

8.2 Components

The Analytics Devstack includes the following additional components.

- edX Analytics Data API

- edX Insights

The Analytics Devstack also includes all of the components needed to run the Open edX Analytics Pipeline, which is the primary ETL (extract, transform, and load) tool that extracts and analyzes data from the other Open edX services.

8.3 Install the Software Prerequisites

To install and run the Analytics Devstack, you must first install the following required software.

- VirtualBox 4.3.12 or higher.

- Vagrant 1.6.5 or higher.

- A Network File System (NFS) client, if your operating system does not already include one. Devstack uses VirtualBox Guest Editions to share folders through NFS.

Additionally, the Open edX Analytics Pipeline includes a tool that is used to deploy itself. To make use of these tools, follow these steps.

1. Clone the repository on your host, not on the Virtual Machine.

```
$ git clone https://github.com/edx/edx-analytics-pipeline
```

2. Install the project dependencies into a virtualenv on your host.

```
$ cd edx-analytics-pipeline
$ virtualenv venv
$ source venv/bin/activate
$ make bootstrap
```

The system is now ready to start running tasks on the Analytics Devstack using the `remote-task` tool.

8.4 Install the Analytics Devstack

To install the Analytics Devstack extensions directly from the command line, follow these steps.

1. Create the `analyticstack` directory and navigate to it in the command prompt.

```
$ mkdir analyticstack
$ cd analyticstack
```

2. Download the Analytics Devstack Vagrant file.

```
$ curl -L https://raw.github.com/edx/configuration/master/vagrant/release/analyticstack/Vagrantf
```

3. Create the Analytics Devstack virtual machine.

```
$ vagrant up
```

8.5 Using the Analytics Devstack

8.5.1 Run the Open edX LMS

1. Log in to the Analytics Devstack.

```
$ vagrant ssh
```

2. Switch to the `edxapp` user.

```
$ sudo su edxapp
```

3. Start the LMS.

```
$ paver devstack lms
```

8.5.2 Run the Open edX Analytics Data API

1. Log in to the Analytics Devstack.

```
$ vagrant ssh
```

2. Switch to the `analytics_api` user.

```
$ sudo su analytics_api
```

3. Start the Data API.

```
$ ~/venvs/analytics_api/bin/python ~/analytics_api/manage.py runserver 0.0.0.0:8100 --insecure
```

8.5.3 Run Open edX Insights

1. Log in to the Analytics Devstack.

```
$ vagrant ssh
```

2. Switch to the `insights` user.

```
$ sudo su insights
```

3. Enable features that are disabled by default.

```
$ ~/venvs/insights/bin/python ~/edx_analytics_dashboard/manage.py switch display_verified_enroll
$ ~/venvs/insights/bin/python ~/edx_analytics_dashboard/manage.py switch enable_course_api on --
```

4. Start Insights.

```
$ ~/venvs/insights/bin/python ~/edx_analytics_dashboard/manage.py runserver 0.0.0.0:8110 --insec
```

5. Open the URL `http://127.0.0.1:8110` in a browser on the host.

Important: Be sure to use the IP address `127.0.0.1` instead of `localhost`. Using `localhost` will prevent you from logging in.

8.5.4 Run the Open edX Analytics Pipeline

1. In the Devstack LMS, register a new user and enroll in the demo course.

2. Navigate to the courseware and submit answers to a few problems.

3. Navigate to the location where edx-analytics-pipeline project was cloned on the host.

```
$ cd edx-analytics-pipeline
```

4. Run the enrollment task.

```
$ export WHEEL_URL=http://edx-wheelhouse.s3-website-us-east-1.amazonaws.com/Ubuntu/precise
# On Mac OS X replace the date command below with $(date -v+1d +%Y-%m-%d)
$ remote-task --vagrant-path <path to `analyticstack`> --remote-name devstack --override-config
    ImportEnrollmentsIntoMysql --local-scheduler --interval-end $(date +%Y-%m-%d -d "tomorrow") -
```

5. Run the answer distribution task.

```
$ export WHEEL_URL=http://edx-wheelhouse.s3-website-us-east-1.amazonaws.com/Ubuntu/precise
$ export UNIQUE_NAME=$(date +%Y-%m-%dT%H_%M_%SZ)
$ remote-task --vagrant-path <path to `analyticstack`> --remote-name devstack --override-config
    AnswerDistributionWorkflow --local-scheduler \
        --src hdfs://localhost:9000/data/ \
        --include '*tracking.log*' \
        --dest hdfs://localhost:9000/edx-analytics-pipeline/output/answer_distribution_raw/$UNIQUE
        --name $UNIQUE_NAME \
        --output-root hdfs://localhost:9000/edx-analytics-pipeline/output/answer_distribution/ \
        --marker hdfs://localhost:9000/edx-analytics-pipeline/output/answer_distribution_raw/$UNIQ
        --n-reduce-tasks 1
```

Installing Open edX Fullstack

This section describes how to install the Open edX Fullstack.

9.1 Overview

Open edX Fullstack is a Vagrant instance designed for deploying all Open edX services on a single server.

See the Vagrant documentation for more information.

9.2 Components

Open edX Fullstack includes the following edX components.

- The Learning Management System (LMS).

- EdX Studio.

- XQueue, the queuing server that uses RabbitMQ for external graders.

- Discussion Forums.

- Open Response Assessor (ORA).

- Discern, the machine-learning-based automated textual classification API service.

- Ease, a library for the classification of textual content.

9.3 Knowledge Prerequisites

To use Fullstack, you should meeting the following knowledge requirements.

- Understand basic terminal usage. If you are using a Mac computer, see Introduction to the Mac OS X Command Line. If you are using a Windows computer, see Windows Command Line Reference.

- Understand Vagrant commands. See the Vagrant Getting Started guide for more information.

9.4 Software Prerequisites

To install and run Open edX Fullstack, you must first install the following software.

- VirtualBox 4.3.12 or higher

- Vagrant 1.6.5 or higher

- A Network File System (NFS) client, if your operating system does not include one. Fullstack uses VirtualBox Guest Editions to share folders through NFS.

9.5 Install Open edX Fullstack

To install Open edX Fullstack directly from the command line, follow the instructions below.

Before beginning the installation, ensure that you have your local computer's administrator's password. The password is needed so that NFS can be set up to allow users to access code directories directly from your computer.

1. Ensure the `nfsd` client is running.

2. Create the `fullstack` directory and navigate to it in the command prompt.

```
mkdir fullstack
cd fullstack
```

3. Download the fullstack Vagrant file.

```
curl -L https://raw.githubusercontent.com/edx/configuration/master/vagrant/release/fullstack/Vag
```

4. Install the Vagrant `hostsupdater` plugin.

```
vagrant plugin install vagrant-hostsupdater
```

5. Create the Fullstack virtual machine.

```
vagrant up
```

The first time you create the Fullstack virtual machine, Vagrant downloads the base box, which has a file size of about 4GB. If you destroy and recreate the virtual machine, Vagrant re-uses the box it downloaded. See Vagrant's documentation on boxes for more information.

6. When prompted, enter the administrator password for your local computer.

9.6 Browser Login to Open edX Fullstack

In your browser, go to `preview.localhost`, which is an alias entry for 192.168.33.10 that was created in your `/etc/hosts` file.

The latest version of fullstack is preloaded with the demonstration course and the same set of *default user accounts* that is created for Devstack.

Changing the Way Open edX Looks

Almost everyone who runs Open edX wants to change the way it looks and feels, to uniquely identify their website and courses. This section describes how to create and use themes to change how Open edX looks, without changing how it works.

10.1 What is Theming?

Any application, including Open edX, can be loosely divided into two parts: the user interface ("how it looks"), and the application logic ("how it works"). For example, when considering a multiple choice problem in Open edX, the user interface consists of how the options are laid out on the page, how the option selectors look, how the submit button is labelled, what sort of fonts and colors are used to display the text, and so on. The application logic consists of how Open edX determines if the student's answer is correct or incorrect, and how it records that information to be displayed in the future.

Theming consists of changing the user interface without changing the application logic. When setting up an Open edX website, the website operator often wants to use their own logo, modify the color scheme, change links in the header and footer for SEO (search engine optimization) purposes, and so on. However, although the user interface may look different, the application logic must remain the same so that Open edX continues to work properly. A well-designed theme preserves the general layout and structure of the user interface, so that users of the website still find it familiar and easy to use. Be careful about making sweeping changes to the user interface without warning: your users will be very confused!

From a technical perspective, theming consists of overriding core templates and core assets with themed templates and themed assets.

- Templates are text files that transform raw data (like a student's history of answered problems) into HTML pages (like a graph of the student's score in the course over time).

- Assets are images, CSS, JavaScript, and any other files that are referenced by the HTML files created by the templates. For example, a logo image is an asset.

The Open edX platform already comes with a set of templates and assets that are used in the default user interface; these are known as core templates and core assets. To create a theme, you simply replace some or all of these core templates with your own themed templates, as well as replacing some or all of the core assets with your own themed assets.

There is no limit to the number of themes you can have installed on your Open edX servers, but only one can be active at a time. Multitenancy (the ability to have different themes active in different contexts) is not yet supported.

10.1.1 What Can Be Themed?

Currently, the LMS and Studio can be themed. All other parts of the Open edX software do not yet support theming. This includes Insights, XBlocks, certificates, notification emails, and so on.

Within the LMS and Studio, only a few templates and assets are currently supported for theming. All templates and assets *can* be themed, but unsupported templates and assets may be renamed or used differently in the future, and these changes may occur without notice. The following pages describe the currently supported templates and assets.

- *Supported Themable Templates*
- *Supported Themable Assets*
- *Supported Themable Sass Variables*

10.2 Creating a Theme

To make a theme, you start by creating a directory with an identifying name for your theme. It is a good idea to put a README file into this directory with a description of what your theme is for, and to put the entire directory under version control.

In this example, the theme is named `my-theme`.

```
my-theme
-- README.rst
```

Next, you create a subdirectory for the Open edX component that you want your theme to apply to. One theme can apply to multiple components: LMS, Studio, certificates, emails, and so on. Add a subdirectory for each component that your theme affects. For the moment, only the LMS component and the Studio component are supported.

For this example, you make a `lms` directory for the LMS component.

```
my-theme
-- README.rst
-- lms
```

Most themes override one or more assets, such as overriding the logo in the LMS. To override an asset in the LMS, make a `static` directory inside the `lms` directory, and add an asset with the same name and directory path as the asset you want to override.

For example, to override the logo in the LMS, the name and path is `images/logo.png`.

```
my-theme
-- README.rst
-- lms
    -- static
        -- images
            -- logo.png
```

You can override templates, as well. To override a template in the LMS, make a `templates` directory in the `lms` directory, and add a template with the same name and directory path as the template you want to override. At the moment, only Mako templates are supported.

For example, to override the header of the LMS, the name and path is simply `header.html`.

```
my-theme
-- README.rst
-- lms
    -- static
```

```
|    -- images
|        -- logo.png
-- templates
    -- header.html
```

You can continue to add templates and assets to the `my-theme` directory. For more information, see *Supported Themable Templates* and *Supported Themable Assets*.

You can also theme CSS files using Sass variables. For more information, see *Sass Theming* and *Supported Themable Sass Variables*.

When you have finished creating your theme, *activate your theme*.

10.3 Sass Theming

CSS files are treated as assets, and like all assets in Open edX, they can be replaced with themed versions. However, the Open edX project uses the Sass framework for writing and organizing CSS files, and this gives theme authors more flexibility with how they modify styling on Open edX. Instead of replacing the CSS files outright, theme authors can override specific Sass variables to modify parts of the CSS files without replacing all of them.

Note: The Sass/CSS Styleguide, created by the edX UX (user experience) team, includes guidance for syntax preferences and suggestions for documenting variables. You don't have to make your Sass overrides conform to this styleguide, but your theme may be easier to maintain if it does.

To override Sass variables, create a `sass` directory inside of the `static` directory in your theme. If you are overriding Sass for the LMS, in a theme named `my-theme`, the directory structure looks like the following example.

```
my-theme
-- lms
    -- static
        -- sass
```

Inside of the `sass` directory, you can create Sass files with names that match the core Sass files in the platform. For example, most of the styling in the LMS comes from the `lms/static/sass/lms-main.scss` Sass file, so to override styling in that file, you must create a file named `lms-main.scss` in your theme's `sass` directory.

```
my-theme
-- lms
    -- static
        -- sass
            -- lms-main.scss
```

Within that file, you must define whatever themable Sass variables you want, and then import the core Sass file that you wish to override. Your theme's `lms-main.scss` might look like this example.

```
// Make the background red
$header-bg: rgb(250,0,0);

// Import the core Sass file
@import 'lms/static/sass/lms-main';
```

Note: When overriding variables in Sass, your variables must be defined **before** you import the core template, otherwise they will not work properly.

10.3.1 Organizing Sass with Partials

Sass partials are a widely-used way of organizing Sass and reducing repetition in your styling. The Open edX core Sass files use partials extensively, and edX recommends that you do so, as well. During the Sass compilation process, Open edX produces multiple different CSS files, including special versions of these files for languages with right-to-left alphabets. In order to override Sass variables for all of these CSS files without needing to repeat your overrides multiple times, edX recommends that you use Sass partials to define your overrides once, and import those overrides wherever they are needed.

In the previous example, you overrode the `$header-bg` variable in the `lms-main.scss` file only. To make these overrides also apply when Open edX users are using a right-to-left language, make the same overrides in the `lms-main-rtl.scss` file, as well. To override the `lms-main-rtl.scss` file, you add a Sass file with that name to your `my-theme` Sass directory. As a result, your `my-theme` directory now includes the following subdirectories and files.

```
my-theme
-- lms
    -- static
        -- sass
            -- lms-main.scss
            -- lms-main-rtl.scss
```

Without using partials, the contents of the `lms-main-rtl.scss` file might look like this example.

```
// Make the background red
$header-bg: rgb(250,0,0);

// Import the core Sass file
@import 'lms/static/sass/lms-main-rtl';
```

But that means you are defining the `$header-bg` variable in two different places. If you ever want to change the background color again in the theme, you will need to remember to do it twice. This becomes much more difficult to keep track of as the number of Sass overrides in your theme grows.

Instead, move this definition out into a partial. Add another Sass file to your theme's `sass` directory, and name it `_overrides.scss`. Be sure that the name starts with an underscore!

```
my-theme
-- lms
    -- static
        -- sass
            -- _overrides.scss
            -- lms-main.scss
            -- lms-main-rtl.scss
```

In this `_overrides.scss` file, you can define only the overrides that should apply to both of the other files, like this example.

```
// Make the background red
$header-bg: rgb(250,0,0);
```

Now modify both of the other Sass files to import this partial. The contents of `lms-main.scss` should now look like this example.

```
// Import our theme overrides
@import 'overrides';

// Import the core Sass file
@import 'lms/static/sass/lms-main';
```

The contents of `lms-main-rtl.scss` should look like this example.

```scss
// Import our theme overrides
@import 'overrides';

// Import the core Sass file
@import 'lms/static/sass/lms-main-rtl';
```

Note: The Sass partial must be in a name that starts with an underscore: `_overrides.scss`. However, when using `@import` in the other Sass files in your theme, you should **not** start with an underscore: `@import 'overrides';`. For more information, see the documentation for Sass partials.

This way, both files will load the contents of your `_overrides.scss` file, and you only need to modify your overrides in one place.

10.3.2 Adding New Sass Rules

In addition to overriding Sass variables in your theme, you can also define new Sass rules. These rules are added to any other rules defined in the core Sass files, rather than replacing the existing Sass rules. To add a new Sass rule, you define it in the themed Sass file, or one of the Sass partials that your themed Sass file imports. For example, your `_overrides.scss` could contain the following content.

```scss
// Override a Sass variable
$header-bg: rgb(250,0,0);

// Define a new Sass rule
body ul li {
    font-size: 1000px;
}
```

You still need to import the core Sass file in your themed `lms-main.scss` file; otherwise, your Sass rules will replace all of the core Sass rules instead of adding to them.

10.3.3 Next Steps

When you have finished creating your theme, *activate your theme*.

10.4 Activating a Theme

To activate a theme, you must first install your theme onto the same server that is running Open edX. If you are using devstack or fullstack, you must be sure that the theme is present on the Vagrant virtual machine. If you are installing Open edX onto a server using the configuration scripts, you must update the configuration scripts to install your theme onto the server, as well. It is up to you where to install the theme on the server, but a good default location is `/edx/app/themes/my-theme`.

Once the theme is installed on the server, you can activate it by setting the `COMPREHENSIVE_THEME_DIR` Ansible variable to the full path of your theme's directory. You can do that by adding the following value to your `/edx/app/edx_ansible/server-vars.yml` file:

```yaml
COMPREHENSIVE_THEME_DIR: "/edx/app/themes/my-theme"
```

Where `/edx/app/themes/my-theme` is the path to where you have installed the theme on your server.

To propogate this variable to the `lms.env.json` file, you run Ansible.

```
sudo /edx/bin/update edx-platform master
```

You will then need to restart the LMS in order for it to pick up this value from the `lms.env.json` file.

Test that your theme is activated by visiting the LMS in your browser, and verifying that the changes in your theme appear on your Open edX installation.

10.5 Supported Themable Assets

In theory, all core assets in the LMS and Studio can be overriden by themed assets. In practice, most of the system assumes that core assets are not currently overridden by themed assets, and you might see unexpected behavior if you override certain assets. So far, the list of supported themable assets is limited to the following set.

Component	Asset	Description
LMS	`images/logo.png`	The logo for the LMS. Displayed in the upper left corner of the LMS.
Studio	`images/studio-logo.png`	The logo for Studio. Displayed in the upper left corner of Studio.
LMS	`images/profiles/default_30.png` `images/profiles/default_50.png` `images/profiles/default_120.png` `images/profiles/default_500.png`	Default profile images for learner profiles. The default image displays for learners until they upload their own profile images. These images are named based on their sizes: 30px by 30px, 50px by 50px, 120px by 120px, and 500px by 500px, respectively.

10.6 Supported Themable Templates

In theory, all core Mako templates in the LMS and Studio can be overriden by themed templates. In practice, most of the system assumes that core templates will *not* be overridden by themed templates, and you might see unexpected behavior if you override certain templates. So far, the list of supported themable templates is limited to the following set.

Component	Template	Description
LMS	`header.html`	The page header for the LMS. This template is used for the top of the `<body>` element on the page, not for the `<head>` element of the page. EdX recommends that you start by copying the contents of the `navigation.html` core template, and modifying it as necessary.
LMS	`footer.html`	The page footer for the LMS. EdX recommends that you start by copying the contents of the `footer.html` core template, and modifying it as necessary.

Note: Currently, you can only override templates that are loaded by the Mako templating system. You can override Mako templates and Underscore templates that are loaded using the `<%static:include%>` Mako template tag. You cannot override Django templates.

10.7 Supported Themable Sass Variables

There are currently no supported themable Sass variables.

Sass variables in the LMS and Studio *can* be overriden, but the meaning and usage of these variables are subject to change, so they are not considered to be "supported". If you choose to override these Sass variables anyway, you can find these variables defined in a `_variables.scss` Sass partial in the edx-platform repository.

Component	File
LMS	`lms/static/sass/base/_variables.scss`
Studio	`cms/static/sass/_variables.scss`

You can only override variables that use the Sass `!default` flag. For more information, see the documentation for Sass variable defaults.

10.8 Theming Glossary

- A **theme** is a collection of templates and assets that override the look and feel of an Open edX installation. It consists of a directory containing those templates and assets in a specific hierarchy.

- A **core template** is a template that exists in one of the Open edX codebases. For example, the HTML files in the `lms/templates` directory of the `edx-platform` repository are core templates.

- A **core asset** is an image, CSS file, or JavaScript file that exists in one of the Open edX codebases. For example, the image files in the `lms/static/images` directory of the `edx-platform` repository are core assets.

- A **themed template** is a template that exists in a theme. Typically, themed templates are organized and named in a way that parallels a core template, so that the themed template will override the core template. For example, a themed template named `my-theme-directory/lms/templates/header.html` will override a core template named `lms/templates/header.html` in the `edx-platform` repository.

- A **themed asset** is an image, CSS file, or JavaScript file that exists in a theme. Typically, themed assets are organized and named in a way that parallels a core asset, so that the themed asset will override the core template. For example, a themed asset named `my-theme-directory/lms/static/images/logo.png` will override a core asset named `lms/static/images/logo.png` in the `edx-platform` repository.

Configuring the Open edX Platform

The following sections provide information about Open edX Platform configuration options.

11.1 Guidelines for Updating the Open edX Platform

When you update the Open edX Platform, you should not change configuration files on a running server. Doing so can result in unpredictable problems.

If you need to change settings on a running server, take the following steps.

1. Provision a new server that matches the running server.

2. Make configuration changes on the new server.

3. Start the new server.

4. Reroute traffic from the old server to the new server.

5. Decommission the old server.

11.2 Enabling Course Prerequisites

This topic describes how to enable course prerequisites in your instance of Open edX.

- *Overview*
- *Configure the Milestones Application*
- *Enable Prerequisite Courses in Studio and the Learning Management System*

11.2.1 Overview

Course teams can set prerequisites for a course. Learners must complete the prerequisite courses before participating in the course.

To use this feature on your instance of Open edX, you must configure the Milestones application, then enable prerequisites in Studio and the Learning Management System.

For information about prerequisites, see the *Building and Running an Open edX Course* and *Open edX Learner's* guides.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.2.2 Configure the Milestones Application

1. Set the value of MILESTONES_APP in the lms.env.json and cms.env.json files to True.

   ```
   # Milestones application flag
   'MILESTONES_APP': True,
   ```

2. Save the the lms.env.json and cms.env.json files.

3. Run database migrations.

11.2.3 Enable Prerequisite Courses in Studio and the Learning Management System

To enable prerequisite courses, you modify the lms.env.json and cms.env.json files, which are located one level above the edx-platform directory.

1. Set the value of ENABLE_PREREQUISITE_COURSES in the lms.env.json and cms.env.json files to True.

   ```
   # Prerequisite courses feature flag
   'ENABLE_PREREQUISITE_COURSES': True,
   ```

2. Save the the lms.env.json and cms.env.json files.

11.3 Enabling Entrance Exams

This topic describes how to enable entrance exams in your instance of Open edX.

- *Overview*
- *Configure the Milestones Application*
- *Enable Entrance Exams in Studio and the Learning Management System*

11.3.1 Overview

Course teams can create an entrance exam for the course. Learners must pass the entrance exam before participating in the course.

To enable this feature on your instance of Open edX, you must enable entrance exams in Studio and the Learning Management System.

For information about entrance exams, see the *Building and Running an Open edX Course* and *Open edX Learner's* guides.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.3.2 Configure the Milestones Application

1. Set the value of `MILESTONES_APP` in the `lms.env.json` and `cms.env.json` files to `True`.

```
# Milestones application flag
'MILESTONES_APP': True,
```

2. Save the the `lms.env.json` and `cms.env.json` files.

3. Run database migrations.

11.3.3 Enable Entrance Exams in Studio and the Learning Management System

To enable entrance exams, you modify the `lms.env.json` and `cms.env.json` files, which are located one level above the `edx-platform` directory.

1. Set the value of `ENTRANCE_EXAMS` in the `lms.env.json` and `cms.env.json` files to `True`.

```
# Entrance exams feature flag
'ENTRANCE_EXAMS': True,
```

2. Save the `lms.env.json` and `cms.env.json` files.

11.4 Enabling Course and Video Licensing

This topic describes how to enable licensing in your instance of Open edX.

- *Overview*
- *Enable Licensing in Studio*

11.4.1 Overview

Course teams can specify licensing options for course content as well as for each video in a course.

Course teams can select one of the following license options.

- All Rights Reserved

- Creative Commons

By specifying the license, course teams communicate to learners whether and how they can reuse course content.

To enable this feature on your instance of Open edX, you must enable licensing in both Studio and the Learning Management System.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.4.2 Enable Licensing in Studio

To enable licensing, you modify the `lms.env.json` and `cms.env.json` files, which are located one level above the `edx-platform` directory.

1. In the `lms.env.json` and `cms.env.json` files, in the `FEATURES` dictionary, add `'LICENSING':True`:

```
FEATURES = {
    'LICENSING': True,
    . . .
```

2. Save the `lms.env.json` and `cms.env.json` files.

11.5 Enabling Open edX Search

This section describes how to enable search in your instance of Open edX.

- *Overview*
- *Search Engines and edX Search*
- *EdX Search Requirements*
- *Install edX Search*
- *Enable Indexing*
- *Supported Flags*

11.5.1 Overview

EdX Search is a Django application that provides access to search services from within edX Platform applications. Searching is accomplished by creating an index of documents, and then searching within that index for matching information.

When you install the Open edX developer stack, edX search is enabled by default. You must enable this feature to use it with Open edX full stack.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.5.2 Search Engines and edX Search

By default, edX Search uses MockSearchEngine for testing and ElasticSearch Engine for production. You can configure edX Search to use a different search engine.

MockSearchEngine

MockSearchEngine is a simple implementation using a JSON file for index storage. It has no specific requirements, but it does not scale well and should only be used for testing.

ElasticSearchEngine

ElasticSearchEngine is a ElasticSearch back-end implementation. It uses same ElasticSearch version that is already part of Open edX Platform. The current version is v0.90.13, and Django Elasticsearch is 0.4.5.

11.5.3 EdX Search Requirements

EdX Search requires the following applications.

- Django (edX Platform version)
- pyMongo (edX Platform version)
- pytz
- Django elasticsearch (0.4.5)

11.5.4 Install edX Search

EdX Search is included in Open edX Platform GitHub requirements and is installed automatically when you install the Open edX Platform.

For existing installations, you must install edX Search manually.

To install edX Search, make sure you are logged to your server as the `edxapp` user and are located in `edx-platform` directory.

If you are not, run the following before continuing:

```
sudo su edxapp -s /bin/bash cd ~source edxapp_env
```

Then install edX Search using one of the three following options.

Option 1 – Add Requirement

Add the GitHub link to edx-search to the `requirements/edx/github.txt` file.

```
-e git+https://github.com/edx/edx-search.git@ae459ead41962c656ce794619f58cdae46eb7896#egg=edx-search
```

Then reinstall GitHub requirements.

```
pip install -r requirements/edx/github.txt
```

Option 2 – Install Locally

Checkout the `edx-search` GitHub repository.

Then in the `edx-search` directory, run the following command.

```
pip install -e ./
```

Option 3 – Install from GitHub

Run `pip` with a GitHub link.

```
pip install -e git+https://github.com/edx/edx-search.
  git@ae459ead41962c656ce794619f58cdae46eb7896
```

11.5.5 Enable Indexing

You enable course indexing by setting the `ENABLE_COURSEWARE_INDEX` flag.

You enable library indexing by setting the `ENABLE_LIBRARY_INDEX` flag.

Indexing is done from Studio as a Celery task. Every publish event triggers the reindex procedure.

You can also reindex the course manually through the **Reindex** button in the **Course Overview** page.

Which Data Gets Indexed

Which data gets indexed is determined by the module `index_dictionary()` function implementation. Modules supporting this method are `Sequence`, `Vertical`, `Video`, and `HTML Block`. You can add support to any module type.

Course metadata, including the name, description, and start and end dates are also indexed.

11.5.6 Supported Flags

The following flags are supported in the CMS and LMS applications.

CMS

- `ENABLE_COURSEWARE_INDEX`: Enables and disables courseware content and course info indexing.

- `ENABLE_LIBRARY_INDEX`: Enables and disables library content indexing.

- `SEARCH_ENGINE`: Sets the search engine to use. There are two predefined values.

    - `"search.elastic.ElasticSearchEngine"`

    - `"search.tests.mock_search_engine.MockSearchEngine"`

- `ELASTIC_FIELD_MAPPINGS`: Sets any additional field mappings that elastic search should be aware of. For example, the following code includes the course start date.

```
ELASTIC_FIELD_MAPPINGS = {
  "start_date": {
    "type": "date"
  }
}
```

LMS

- `ENABLE_COURSEWARE_SEARCH`: Enables and disables Courseware Search feature (in course searching).

- `ENABLE_DASHBOARD_SEARCH`: Enables and disables Dashboard Search feature (in enrolled courses searching).

- `ENABLE_COURSE_DISCOVERY`: Enables and disables Course Discovery feature (over courses searching and facet filtering).

- `COURSE_DISCOVERY_FILTERS`: If provided, overrides the list of facets that are used in the Course Discovery feature to filter the results. By default, all facets will be displayed. The list of available facets includes:

    - Course organization: `"org"`

- Course type: `"modes"`
  - Course language: `"language"`
- `SEARCH_ENGINE`: Sets the search engine to use. The following values are predefined.
  - `"search.elastic.ElasticSearchEngine"`
  - `"search.tests.mock_search_engine.MockSearchEngine"`
- `SEARCH_INITIALIZER`: Used to set custom SearchInitializer. SearchInitializer provides an extension to achieve masquerade and other presearch environmental settings.
  - default: `SearchInitializer`
  - LMS implementation: `lms.lib.courseware_search.lms_search_initializer.LmsSearchInitialize`
- `SEARCH_RESULT_PROCESSOR`: Used to set custom SearchResultProcessor. SearchResultProcessor does post processing and data manipulation on a result set returned by SearchEngine.
  - default: `SearchResultProcessor`
  - LMS implementation: `lms.lib.courseware_search.lms_result_processor.LmsSearchResultProces`
- `SEARCH_FILTER_GENERATOR`: Used to set custom SearchFilterGenerator. SearchFilterGenerator sets filters defined by current active user. Basic implementation sets only course start date filter.
  - default: `SearchFilterGenerator`
  - LMS implementation: `lms.lib.courseware_search.lms_filter_generator.LmsSearchFilterGenera`

11.6 Enabling Certificates

This topic describes how to enable certificates in your instance of Open edX.

- *Overview*
- *Enable Certificates in Studio and the Learning Management System*
- *Configure Certificates for Your Open edX Instance*
- *Customize Certificate Templates For Your Organization*
- *Configure Certificates Within Each Course*
- *Generate Certificates For a Course*

11.6.1 Overview

Organizations and course teams can now generate certificates for learners who have completed courses. Learners can view, print, or share their certificates.

For information about certificates, see the *Building and Running an Open edX Course* and *Open edX Learner's* guides.

To enable this feature on your instance of Open edX, you must enable the feature flag in both Studio and the Learning Management System and perform the configuration tasks described in this topic.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.6.2 Enable Certificates in Studio and the Learning Management System

To enable certificates, you modify the `lms.env.json` and `cms.env.json` files, which are located one level above the `edx-platform` directory.

1. In the `lms.env.json` and `cms.env.json` files, set the value of `CERTIFICATES_HTML_VIEW` within the `FEATURES` object to `true`.

```
"FEATURES": {
    ...
    'CERTIFICATES_HTML_VIEW': true,
    ...
}
```

2. Save the `lms.env.json` and `cms.env.json` files.

3. If it does not exist already, create the folder `/tmp/certificates` owned by the user and group `www-data`. Depending on your configuration, this folder might not survive reboots, and so might need to be created by a script.

4. Run database migrations.

11.6.3 Configure Certificates for Your Open edX Instance

1. Access the Django Administration website for your instance of Open edX. To do this, go to `https://<host name of your Open edX instance>/admin`. For example, this might be `https://YourOrganization.com/admin`.

2. Under **Site Administration > Certificates**, add an HTML View Configuration, and select **Enabled**.

3. Modify the configuration parameters. You must set the following certificates-related parameters for your Open edX instance.

   - `platform_name`
   - `company_about_url`
   - `company_privacy_url`
   - `company_tos_url`
   - `company_verified_certificate_url`
   - `logo_src`
   - `logo_url`

   For each course mode, such as "honor" or "verified", define `certificate_type`, `certificate_title` and `document_body_class_append`. The mode name should match your course mode name exactly.

```
{
    "default": {
        "accomplishment_class_append": "accomplishment-certificate",
        "platform_name": "YourPlatformName",
        "company_about_url":"http://www.YourOrganization.com/about-us",
        "company_privacy_url": "http://www.YourOrganization.com/our-privacy-policy",
        "company_tos_url": "http://www.YourOrganization.com/our-terms-service",
        "company_verified_certificate_url": "http://www.YourOrganization.com/about_verified_cert
        "logo_src": "/static/certificates/images/our_logo.svg",
        "logo_url": "www.YourOrganization.com"
    },
    "honor": {
```

```
            "certificate_type": "honor",
            "certificate_title": "Honor Certificate",
            "document_body_class_append": "is-honorcode"
        },
        "verified": {
            "certificate_type": "verified",
            "certificate_title": "Verified Certificate",
            "document_body_class_append": "is-idverified"
        },
        "base": {
            "certificate_type": "base",
            "certificate_title": "Certificate of Achievement",
            "document_body_class_append": "is-base"
        },
        "distinguished": {
            "certificate_type": "distinguished",
            "certificate_title": "Distinguished Certificate of Achievement",
            "document_body_class_append": "is-distinguished"
        }
    }
```

4. Save the configuration parameters and exit the Django Administration website.

5. Restart the Studio and Learning Management System processes so that the updated environment configurations are loaded.

11.6.4 Customize Certificate Templates For Your Organization

Set up the templates for certificates that your organization will issue. Base templates are included, but you must ensure that they are customized for your organization. For example, you can change the images that appear on certificates for each course mode that your organization supports, as well as fonts and colors that are used on certificates.

Assets for HTML certificates exist in the following locations.

- `lms/templates/certificates` - this folder contains .html files for certificates. The file `valid.html` is an example of a certificate file. Files with names that start with an underscore, such as `_certificate_footer.html`, are partial files that can be referenced in the main certificate .html files

- `lms/static/certificates` - subfolders of this folder contain assets used in creating certificates, such as images, fonts, and sass/css files.

Note: The organization logo on a certificate is uploaded in Studio. For details, see Setting Up Course Certificates in *Building and Running an Open edX Course*.

11.6.5 Configure Certificates Within Each Course

Within Studio, course team members with the Admin role can create and edit a certificate configuration that is used to generate certificates for their course, including adding signatories and images for organization logo and signature images for signatories. For details, Setting Up Course Certificates in *Building and Running an Open edX Course*.

11.6.6 Generate Certificates For a Course

To generate certificates for a course, run the `manage.py` script with the following settings. When the script finishes running, grades are calculated for learners who are enrolled in the course, and certificates are generated for eligible learners.

1. Obtain the course ID for the course for which you are generating certificates. When you view course content in your browser, the course ID appears as part of the URL. For example, in the URL `http://www.edx.org/course/course-v1:edX+demoX_Demo_2015`, the course ID is `course-v1:edX+demoX_Demo_2015`. For some courses, the course ID contains slashes. For example, `edX/Demox/Demo_2014`.

2. Run `manage.py` with the following settings, replacing `{CourseID}` with the actual course ID. Do not include beginning or trailing slashes.

   ```
   ./manage.py lms --settings=aws ungenerated_certs -c {CourseID}
   ```

   For example,

   ```
   ./manage.py lms --settings=aws ungenerated_certs -c course-v1:edX+demoX_Demo_2015.
   ```

   Note: If the LMS is running on a server that does not have https support (such as a locally run fullstack for testing) you will need to use the `--insecure` flag so that the certificate generation service contacts the LMS on http instead of on https.

   ---

3. View the certificate generation status for a course using `gen_cert_report`. An example follows.

   ```
   ./manage.py lms --settings=aws gen_cert_report -c course-v1:edX+demoX_Demo_2015.
   ```

11.7 Enabling Badging

This topic describes how to enable badging in your instance of Open edX.

- *Overview*
- *Make Sure that Certificates are Enabled*
- *Install Badgr Server*
- *Specify a Badge Issuer for Your Organization*
- *Enable Badges in Studio and the Learning Management System*
- *Configure Badges and Badge Images for Your Open edX Instance*
- *Enable Badges Within Each Course*

11.7.1 Overview

Badges provide a way for learners to share their course achievements. For courses that have badges enabled, learners receive a badge at the same time as they receive a course certificate, and have the option of sharing their badges to a badging site such as Mozilla Backpack.

Open edX supports Open Badges, an open standard originally developed by the Mozilla Foundation. For more information about Open Badges, see http://openbadges.org/.

Enabling the badges feature on your instance of Open edX involves the following set up and configuration tasks.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.7.2 Make Sure that Certificates are Enabled

Badge generation depends on certificate generation. Badges are automatically generated when a certificate is generated for a learner. Make sure certificates are enabled on your Open edX instance. For details, see *Enabling Certificates*.

11.7.3 Install Badgr Server

Badgr Server provides an API for issuing Open Badges. Follow the instructions at https://github.com/concentricsky/badgr-server to install and run Badgr Server.

Important: You must install Badgr Server at a publicly accessible IP address, to allow the Open edX LMS and services such as Mozilla Backpack to contact Badgr Server.

11.7.4 Specify a Badge Issuer for Your Organization

Log in to your installation of Badgr Server and add an issuer of Open Badges for your organization.

For more information about issuing Open Badges, see the Issuing Badges topic on the Mozilla wiki.

11.7.5 Enable Badges in Studio and the Learning Management System

To enable badges, you modify the `lms.env.json` and `cms.env.json` files, which are located one level above the `edx-platform` directory.

1. In the `lms.env.json` and `cms.env.json` files, set the value of `ENABLE_OPENBADGES` to `True`.

```
# Enable OpenBadge support. See the BADGR_* settings later in this file.
'ENABLE_OPENBADGES': True,
```

2. In `lms.env.json`, set the values for the following parameters.

   - `BADGR_API_TOKEN`: A string containing the API token for the Badgr superuser account. Obtain the token from the /v1/user/auth-token page while logged in to the API as the superuser.

   - `BADGR_BASE_URL`: A string containing the base URL for Badgr Server. The Badgr Server must be installed at a publicly accessible IP address.

   - `BADGR_ISSUER_SLUG`: A string that is the slug for the Badgr issuer. The slug can be obtained from the URL of the Badgr Server page that displays the issuer. For example, in the URL `http://exampleserver.com/issuer/test-issuer`, the issuer slug is `test-issuer`.

```
############## Badgr OpenBadges generation ##############

BADGR_API_TOKEN = None
# Do not add the trailing slash here.
BADGR_BASE_URL = "http://localhost:8005"
BADGR_ISSUER_SLUG = "test-issuer"
```

3. Save the `lms.env.json` and `cms.env.json` files.

4. Run database migrations.

5. Restart the Studio and Learning Management System processes so that the updated environment configurations are loaded.

11.7.6 Configure Badges and Badge Images for Your Open edX Instance

---

Important: Default images are supplied for badges. You must replace the default images with your organization's own badge images before any badges are issued. When the first badge is issued for a given course, badge images are uploaded to Badgr Server. All badges issued in future for this course will use the original badge image, even if you subsequently change badge images in the Django Administration badge image configuration.

---

1. Access the Django Administration website for your instance of Open edX. To do this, go to `https://<host name of your Open edX instance>/admin`. For example, this might be `https://YourOrganization.org/admin`.

2. Under **Site Administration > Certificates**, define a Badge Image Configuration for each course mode on your platform for which you want to issue badges. For example, "honor" and "verified".

3. For each badge image configuration, set these parameters.

   • Course Mode

   • Icon: The badge image to use for the specified course mode.

   Important: Be sure to replace the default badge images with your organization's own badge images before any badges are issued.

   ---

4. Optionally, define a default image for any course modes that do not have an explicitly specified badge image. Select **Default** in the badge image configuration.

   Note: You can specify only one default badge image.

   ---

5. Save each configuration parameter and exit the Django Administration website.

11.7.7 Enable Badges Within Each Course

Badge issuing is enabled by default for all courses, but can be turned off or on again using an advanced setting in Studio. For details, see Enable or Disable Badges for Your Course in *Building and Running an Open edX Course*.

11.8 Enabling Custom Courses

To enable designated users to create custom courses (CCX) on your instance of Open edX, you must configure the `server-vars.yml` file in the edX platform.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

---

1. Stop the LMS server.

---

2. Create or update the file `/edx/app/edx_ansible/server-vars.yml` to include the CCX feature flag.

```
EDXAPP_FEATURES:
  CUSTOM_COURSES_EDX: true
```

3. Run the command `/edx/bin/update`.

```
sudo /edx/bin/update edx-platform <your-branch-name>
```

4. Restart the LMS server.

11.9 Enabling Social Sharing of Courses and Certificates

This section describes how to configure Open edX so that learners can share their certificates, and so course teams can enable learners to share their courses on social media.

- *Overview*
- *Configure Social Sharing*
- *Enable Custom Course URLs*

11.9.1 Overview

You can enable learners to share courses and certificates that they earn on social media sites such as Facebook and Twitter.

To use this feature on your instance of Open edX, you must configure social sharing settings.

Optionally, you can also enable course teams to set custom URLs for social sharing. If a course team sets a custom course URL, posts to the social sharing site can include a link back to that URL. If you do not enable custom course URLS, a link to the course About page in the LMS is used.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.9.2 Configure Social Sharing

To enable social sharing icons for courses, you modify the `lms.env.json` file, which is located one level above the `edx-platform` directory.

1. In the `lms.env.json` file, modify the `SOCIAL_SHARING_SETTINGS` dictionary as needed.

```
SOCIAL_SHARING_SETTINGS = {
    'CUSTOM_COURSE_URLS': True,
    'DASHBOARD_FACEBOOK': True,
    'CERTIFICATE_FACEBOOK': True,
    'CERTIFICATE_FACEBOOK_TEXT': None,
    'CERTIFICATE_TWITTER': True,
    'CERTIFICATE_TWITTER_TEXT': None,
    'DASHBOARD_TWITTER': True,
    'DASHBOARD_TWITTER_TEXT': None
}
```

(a) For each social sharing icon that you want to enable, set the value of the setting to `True`.

(b) If you set `DASHBOARD_TWITTER` or `CERTIFICATE_TWITTER` to `True`, you can also specify default text that learners will see in the Twitter sharing dialog and that can be included in their tweet. Set the default text in the `DASHBOARD_TWITTER_TEXT` and `CERTIFICATE_TWITTER_TEXT` values. Learners can edit this text before they select the **Share with Twitter** button in the LMS.

(c) If you set `CUSTOM_COURSE_URLS` to `True`, you must *Enable Custom Course URLs*.

2. Configure the `SOCIAL_MEDIA_FOOTER_NAMES` array to the order of links you want learners to see in the footer.

```
SOCIAL_MEDIA_FOOTER_NAMES = [
    "facebook",
    "twitter",
    "youtube",
    "linkedin",
    "google_plus",
    "reddit",
]
```

3. Configure the `SOCIAL_MEDIA_FOOTER_DISPLAY` dictionary to define how you want social media icons to be displayed. For each social media icon you enable, you define a `title`, `icon`, and `action`.

```
"facebook": {
    "title": _("Facebook"),
    "icon": "fa-facebook-square",
    "action": _("Like {platform_name} on Facebook")
},
"twitter": {
    "title": _("Twitter"),
    "icon": "fa-twitter",
    "action": _("Follow {platform_name} on Twitter")
},
"linkedin": {
    "title": _("LinkedIn"),
    "icon": "fa-linkedin-square",
    "action": _("Follow {platform_name} on LinkedIn")
}
}
```

4. Save the `lms.env.json` file.

11.9.3 Enable Custom Course URLs

In addition to enabling the social sharing icons, you can allow course teams to provide a custom URL for social sharing sites to link back to.

You must set the `CUSTOM_COURSE_URLS` parameter to `True` in both the `lms.env.json` and `cms.env.json` files. In the `cms.env.json` file, this parameter is the only social sharing setting.

```
SOCIAL_SHARING_SETTINGS = {
    'CUSTOM_COURSE_URLS': True
}
```

When finished, save the `lms.env.json` and `cms.env.json` files.

Set a Custom URL for a Course

When you enable custom course URLs in your instance of Open edX, course teams can then set custom URLs for their courses.

In Studio **Advanced Settings**, the course team specifies the custom course URL in the **Social Media Sharing URL** setting.

This URL is provided to the social sharing site for linking back to a course location. This URL is used only if you have enabled custom URLs in your instance of Open edX.

Note: If custom URLs are enabled but a course team does not provide a value in the **Social Media Sharing URL** advanced setting in Studio, social sharing icons are not visible in the LMS for that course.

11.10 Enabling Third Party Authentication

To enhance sign in options for your users, you can enable third party authentication between institutional authentication systems and your implementation of the edX platform. After you enable third party authentication, users can register and sign in to your Open edX site with their campus or institutional credentials.

11.10.1 Supported Identity Providers

In an exchange of authentication and authorization data, an identity provider securely asserts the identity and access rights of a set of users. Your implementation of the Open edX platform is the service provider that allows the users access on the basis of credentials sent by an identity provider.

For example, your Open edX installation hosts the courses of three different institutions. When you configure the open edX installation to be a service provider, and configure each of the three institutions to be identity providers, you permit learners who have valid user credentials at any of those institutions to access the Open edX site.

You can enable third party authentication between your Open edX system and identity providers that use the SAML 2.0 (Security Assertion Markup Language, version 2.0) standard for authentication. For more information, see *Integrating with a SAML Identity Provider*.

At an Open edX installation, you begin by *enabling the third party authentication feature* for your installation.

At an institution that has a partner membership with edX, you can *enable third party authentication* between your institutional authentication systems and the edX Edge site.

If you are using *edX as an LTI tool provider* to a external learning management system or application, you can set up an authentication workflow between your Open edX system and the system that is the LTI tool consumer. For more information, see *Options for LTI Authentication and User Provisioning* and *Configuring Open edX User Authentication for LTI*.

11.10.2 Enable the Third Party Authentication Feature

By default, third party authentication is not enabled on edX installations. To enable this feature, follow these steps.

1. In the `edx/app/edxapp/lms.env.json` file, edit the file so that it includes the following line in the features section.

```
    "FEATURES" : {
        ...
        "ENABLE_COMBINED_LOGIN_REGISTRATION": true,
        "ENABLE_THIRD_PARTY_AUTH": true
    }
```

2. Save the `edx/app/edxapp/lms.env.json` file.

11.10.3 Configuring your Installation as a SAML Service Provider

The first step in configuring your Open edX installation to act as a SAML SP is to create a credential key pair to ensure secure data transfers with identity providers. To complete the configuration procedure, you configure your Open edX installation as a SAML SP, which creates your metadata XML file.

- *Generate Public and Private Keys*
- *Add Keys to the LMS Configuration File*
- *Configure your Installation as a Service Provider*
- *Ensure that the SAML Authentication Backend is Loaded*

After you complete this configuration, you can share your metadata file with SAML identity providers, and configure them to assert identity and access rights for users to your installation. For more information, see *Integrating with a SAML Identity Provider*.

Generate Public and Private Keys

To generate the keys for your Open edX installation, follow these steps.

1. On your local computer or on the server, open Terminal or a Command Prompt and run the following command.

   `openssl req -new -x509 -days 3652 -nodes -out saml.crt -keyout saml.key`

2. Provide information at each prompt.

Two files, `saml.crt` and `saml.key`, are created in the directory where you ran the command.

Add Keys to the LMS Configuration File

Note: Configuration settings added to the `lms.auth.json` file are reset to their default values when you use Ansible to update edx-platform.

To configure your Open edX installation with your public and private SAML keys, follow these steps.

1. Open the `edx/app/edxapp/lms.auth.json` file in your text editor.

2. Edit the file to add a section for the SAML keys. For example, `# SAML KEYS`.

3. In the new section, add the `SOCIAL_AUTH_SAML_SP_PUBLIC_CERT` parameter followed by a colon (:), a space, and the YAML literal style indicator (|).

```
# SAML KEYS
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT: |
```

4. Open the `saml.crt` file, copy its entire contents, and then paste them onto the line after the `SOCIAL_AUTH_SAML_SP_PUBLIC_CERT` parameter.

```
# SAML KEYS
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT: |
-----BEGIN CERTIFICATE-----
SWP6P/C1ypaYkmS...
    ...j9+hjvbBf3szk=
-----END CERTIFICATE-----
```

5. Add the `SOCIAL_AUTH_SAML_SP_PRIVATE_KEY` parameter followed by a colon (:), a space, and the YAML literal style indicator (|).

```
# SAML KEYS
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT: |
-----BEGIN CERTIFICATE-----
SWP6P/C1ypaYkmS...
    ...j9+hjvbBf3szk=
-----END CERTIFICATE-----
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY: |
```

6. Open the `saml.key` file, copy its entire contents, and then paste them onto the line after the `SOCIAL_AUTH_SAML_SP_PRIVATE_KEY` parameter.

```
# SAML KEYS
SOCIAL_AUTH_SAML_SP_PUBLIC_CERT: |
-----BEGIN CERTIFICATE-----
SWP6P/C1ypaYkmS...
    ...j9+hjvbBf3szk=
-----END CERTIFICATE-----
SOCIAL_AUTH_SAML_SP_PRIVATE_KEY: |
-----BEGIN RSA PRIVATE KEY-----
W1icmlkZN+FtM5h...
    ...s/psgLDn38Q==
-----END RSA PRIVATE KEY-----
```

7. Save and close the `lms.auth.json` file.

Configure your Installation as a Service Provider

To configure your Open edX installation as a SAML service provider, follow these steps.

1. Sign in to the Django administration console for your base URL. For example, `http://{your_URL}/admin`.

2. In the **Third_Party_Auth** section, next to **SAML Configuration** select **Add**.

3. Select **Enabled**.

4. Enter the following information.

- **Entity ID**: Enter a URI for the server. To ensure that this value uniquely identifies your site, the naming convention that edX recommends is to include the server's domain name. For example, `http://saml.mydomain.com/`.

- **Organization Info**: Use the format in the example that follows to specify a language and locale code and identifying information for your installation.

```
{
    "en-US": {
```

```
            "url": "http://www.mydomain.com",
            "displayname": "{Complete Name}",
            "name": "{Short Name}"
        }
    }
```

- **Other config str**: Define the security settings for the IdP metadata files. For more information about the security settings, see the Python SAML Toolkit. An example follows.

```
{
    "SECURITY_CONFIG": {
      "signMetadata": false,
      "metadataCacheDuration": ""
    }
}
```

1. Optionally, you can save your public and private keys in the Django administration console. Because this procedure saves your credentials in the database, edX recommends that you use the `lms.auth.json` file instead. For more information, see *Add Keys to the LMS Configuration File*.

2. Select **Save**. You can direct identity providers to `{your LMS URL}/auth/saml/metadata.xml` for your metadata file.

Ensure that the SAML Authentication Backend is Loaded

By default, SAML is included as an approved data format for identity providers. The default configuration of the `/edx/app/edxapp/lms.env.json` file does not explicitly include the `THIRD_PARTY_AUTH_BACKENDS` setting.

If you have customized this file and added the `THIRD_PARTY_AUTH_BACKENDS` setting to it, you might need to verify that the `third_party_auth.saml.SAMLAuthBackend` python-social-auth backend class is specified for it. That backend is required before you can add SAML IdPs.

To verify that the SAML authentication backend is loaded on a devstack or fullstack installation, review the `/edx/app/edxapp/lms.env.json` file.

11.10.4 Integrating with a SAML Identity Provider

You can integrate your Open edX installation with federated identity solutions that use the SAML 2.0 (Security Assertion Markup Language, version 2.0) standard. An example is Shibboleth, a single sign on system that is used by many educational institutions.

- *Exchange Metadata*
- *Add and Enable a SAML Identity Provider*
- *Configuration Options for SAML Identity Providers*
- *Test an Enabled SAML Provider*

Exchange Metadata

SAML metadata is an XML file that contains the information necessary for secure interactions between identity providers and security providers. You send the URL of your metadata file, created when you *configured your installation as a SAML service provider*, to each identity provider that you want to add. Similarly, you obtain the metadata URLs from identity providers before you add and enable them for your installation.

Add and Enable a SAML Identity Provider

To add and enable a SAML 2.0 identity provider, follow these steps.

1. Log in to the Django administration console for your base URL. For example, `http://{your_URL}/admin`.

2. In the **Third_Party_Auth** section, next to **Provider Configuration (SAML IdPs)** select **Add**.

   Note: If you want to change the configuration of an existing provider, next to **Provider Configuration (SAML IdPs)** select **Change**, and then select **Update** for the provider that you want to configure.

   ---

3. Enter the following information for the provider.

   - **Icon class**: Specifies a Font Awesome image for the button that users will select to access the sign in page for this IdP. The **fa-sign- in** icon is used by default. For university or institutional providers, a suggested alternative is **fa-university**.

   - **Name**: The name of the IdP as you want it to appear on the sign in page.

   - **Secondary**: Select this option to include the IdP in an intermediary list of providers that users access from a **Use my institution/campus credentials** button on the sign in page.

   - **Backend name**: The default, **tpa-saml**, is optimized for use with Open edX and works with most SAML providers. Select a different option only if you have added a custom backend that provides additional functionality.

   - **IdP slug**: A short, unique name to identify this IdP in the URL. The slug becomes part of a URL, so the value that you enter cannot include spaces.

   - **Entity ID**: The URI that identifies the IdP. This ID must match the value specified in the metadata XML file.

   - **Metadata source**: The URL of the XML file that contains this provider's metadata.

4. Specify your selections for any of the other, optional configuration options. For more information about these options, see *Configuration Options for SAML Identity Providers*.

5. When you are ready to enable the provider, select **Enabled** at the top of the page. Alternatively, save your configuration settings and enable the provider at another time.

6. Select **Save** or one of the other save options at the bottom of the page.

Next, you can *test an enabled provider*.

Configuration Options for SAML Identity Providers

To customize the registration process for IdP, you make selections for these optional fields on the Add Provider Configuration (SAML IdP) page.

- **Skip Registration Form**: If you select this option, users are not asked to confirm the user account data supplied for them by the IdP (name, email address, and so on). Select this option only for providers that are known to provide accurate user information.

  By default, users review a registration form with the supplied account details.

- **Skip Email Verification**: If you select this option, users are not required to confirm their email addresses, and their accounts are activated immediately upon registration.

  By default, users receive an email message and must select a link in that message to activate their user accounts.

- **User ID Attribute**: Required. This value is used to associate the user's edX account with the campus account. It is not displayed to users.

  By default, uses `userid`, `urn:oid:0.9.2342.19200300.100.1.1`.

- Optional user attributes: You can indicate specific URN values for the following user attributes.

  By default, the registration form includes all of the following attributes if they are sent by the IdP.

  - **Full Name Attribute**: `commonName`, `urn:oid:2.5.4.3`

  - **First Name Attribute**: `givenName`, `urn:oid:2.5.4.42`

  - **Last Name Attribute**: `surname`, `urn:oid:2.5.4.4`

  - **Username Hint Attribute**: `userid`, `urn:oid:0.9.2342.19200300.100.1.1`

  - **Email Attribute**: `mail`, `urn:oid:0.9.2342.19200300.100.1.3`

If the identity provider sends a value that you do not want to be included on the the registration form, you can enter a value such as "DISABLED" or "IGNORE" in that field.
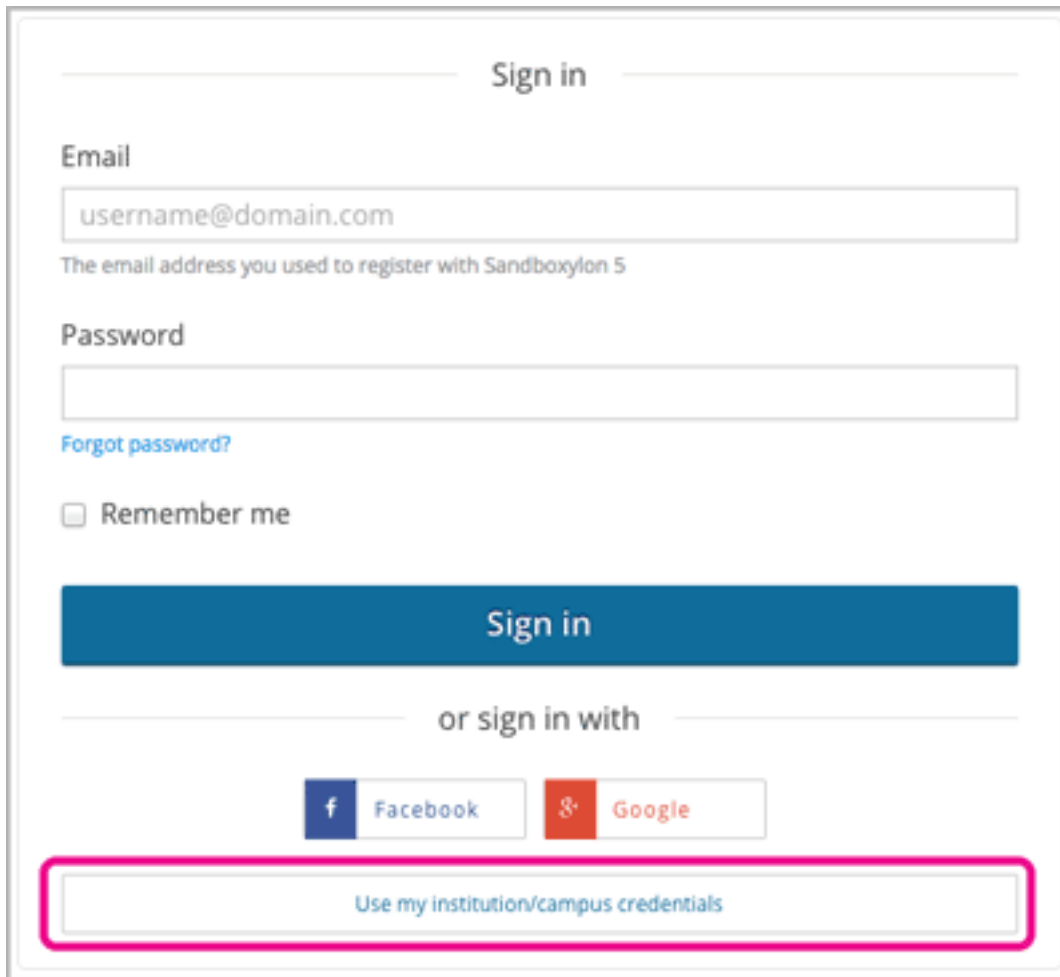
Test an Enabled SAML Provider

To verify the sign in process for an IdP that you have enabled, follow these steps.

1. On the Django administration console, in the **Third_Party_Auth** section, select **Provider Configuration (SAML IdPs)**.

2. Check the icon in the **Metadata ready** column for the IdP. After the provider's metadata is fetched successfully from the URL that you provided as the metadata source, a check mark in a green circle appears and the provider is ready for use immediately.

   You might need to wait 30-60 seconds for the task to complete, and then refresh this page.

   If the check mark does not appear, make sure that celery is configured correctly and is running. You can also manually trigger an update by running the management command `./manage.py lms saml --pull --settings=aws` on Fullstack or `./manage.py lms saml --pull --settings=devstack` on Devstack.

3. For additional information about the data fetched from the IdP, on the Django administration console select **SAML Provider Data**, and then select the provider. The page that opens reports data fetched from the metadata source URL and the date and time it was fetched.

4. To verify that users can use the IdP for sign in, go to the sign in page for your LMS. The page should include the institutional sign in button.

5. Select **Use my institutional/campus credentials**. The list of providers that appears should include the IdP that you enabled.
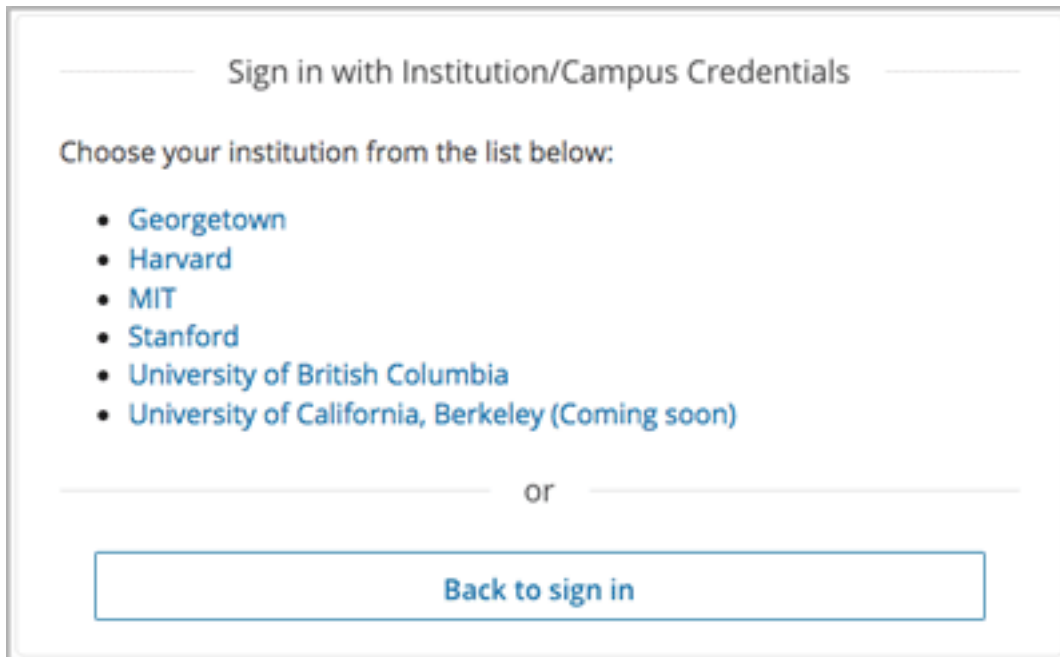
11.10.5 Enabling Third Party Authentication with edX Edge

Institutions that have partner memberships with edX can enable third party authentication between their campus or institutional authentication systems and the edX Edge site. Learners at sites that enable third party authentication can use their campus credentials to authenticate into edX Edge. These procedures require collaboration between members of the DevOps (development operations) or IT teams at your partner institution and edX, facilitated by your edX Partner Manager.

- *Integrating with Shibboleth (SAML) Systems*
    - *Obtain edX Edge SAML Metadata*
    - *Add edX Edge as a Service Provider*
    - *Configure User Attributes*
    - *Send Shibboleth Configuration Data to edX*
    - *Test Edge Registration*

Integrating with Shibboleth (SAML) Systems

SAML 2.0 (Security Assertion Markup Language, version 2.0) is the standard that edX uses for the exchange of authentication and authorization data with institutional partners. Because it is built with SAML, this service is compatible with the Shibboleth single sign on system.

In the exchange of authentication and authorization data, your edX partner institution is an identity provider (IdP) that securely asserts the identity and access rights of a set of users. EdX is the service provider (SP) that allows the users access on the basis of those credentials.

These procedures should be performed by a member of your institution's IT team who is familiar with your institutional Shibboleth system.

Obtain edX Edge SAML Metadata

The service provider SAML metadata for Edge is available in an XML file on the Edge website. This file contains the information necessary for interaction between Edge as service provider and your campus Shibboleth system as identity provider.

Add edX Edge as a Service Provider

On your Shibboleth system, use the SAML metadata obtained from the XML file to add edX Edge to your whitelist of authorized service providers.

For example, you might add the following information to your `$IDP_HOME/conf/relying-party.xml` file.

```
<MetadataProvider xsi:type="FileBackedHTTPMetadataProvider" xmlns="urn:mace:shibboleth:2.0:metadata"
                  id="edxEdgeMetadata"
                  metadataURL="https://edge.edx.org/auth/saml/metadata.xml"
                  backingFile="/tmp/idp-metadata-edx-edge.xml" />
```

For more information about defining a metadata source on a Shibboleth system, see the Shibboleth configuration wiki.

Configure User Attributes

You work with edX to ensure that your identity provider will assert the user information that you want learners to see during initial sign in on edX Edge. When you *send your Shibboleth configuration data* to your institution's edX partner manager, you include any customizations to the attributes that will be asserted.

Selecting User Attributes to Assert By default, the edX platform uses these attributes if your system provides them.

- User identifier: `userid`, `urn:oid:0.9.2342.19200300.100.1.1`. Required. This value is used to associate the user's edX account with the campus account. It is not displayed to users.

- Full name: `commonName`, `urn:oid:2.5.4.3`

- First name: `givenName`, `urn:oid:2.5.4.42`

- Last name: `surname`, `urn:oid:2.5.4.4`

- Suggested username: `userid`, `urn:oid:0.9.2342.19200300.100.1.1`

- Email address: `mail`, `urn:oid:0.9.2342.19200300.100.1.3`

At your request, edX can configure Edge to use only some of these attributes. The only required attribute is the user identifier. If you choose not to provide an attribute, learners are prompted to enter that information themselves during initial sign in.

Specifying Alternative Attributes You can identify different attributes to assert user information than those listed above. For example, you might want to send the suggested username as `eduPersonPrincipalName` instead of `userid`.

Restricting Access with Attribute Assertions You can restrict access to edX Edge to a subset of your users using attributes defined on your Shibboleth system. For example, you might want to allow currently matriculated students to sign in to Edge, but not alumni. The `eduPersonEntitlement` attribute can be used to restrict access in this way.

Note: If you want access to be restricted to certain users, be sure to let edX know what provider assertions to use to determine access rights.

Send Shibboleth Configuration Data to edX

To complete the integration between your Shibboleth system and Edge, send the following information to your institution's edX partner manager.

- Metadata: The URL for your Shibboleth IdP metadata XML file.

- Entity ID: The URI that identifies the Identity Provider. This ID must match the value specified in the metadata XML.

- User Attributes: A list of the values that you want your Shibboleth system to assert when users sign in to edX Edge.
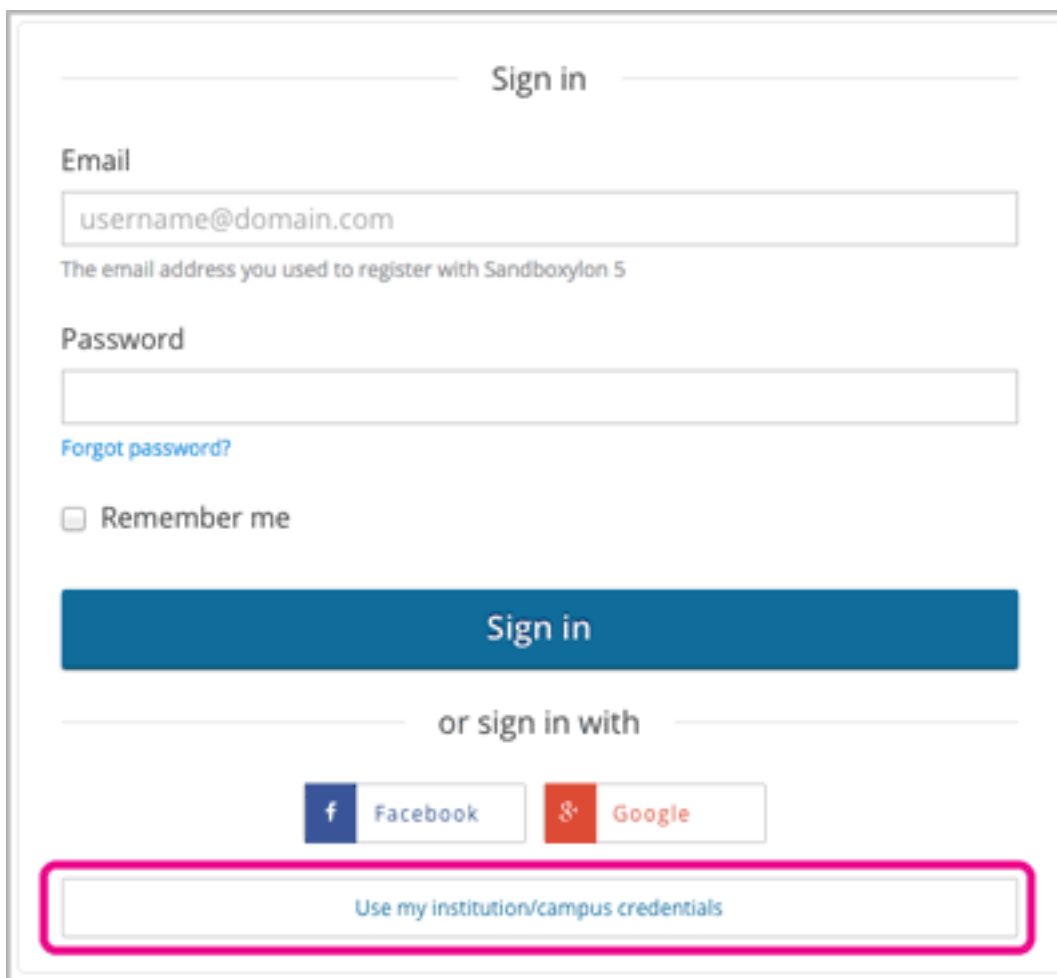
  For more information about how you can work with edX to configure user attributes effectively, see *Configure User Attributes*.

Your edX partner manager notifies you when integration with your IdP is complete.

Test Edge Registration

To verify that users can use their campus or institutional credentials for your IdP to sign in to Edge, follow these steps.

1. Go to the Edge registration page. The page should include the institutional sign in button.

---

2. Select **Use my institutional/campus credentials**. The list of providers that appears should include your IdP.

3. Select your own IdP. The landing page for your Shibboleth system should open.

This section includes information for teams involved in identity management at Open edX installations, including DevOps (development operations) and IT. The *Enabling Third Party Authentication with edX Edge* topic describes procedures for members of the DevOps and IT teams at an edX partner institution.

11.11 Configuring an edX Instance as an LTI Tool Provider

You can configure your edX instance to be a learning tool interoperability (LTI) provider to other systems and applications. You can use this LTI capability to present content from an edX course in any application that is configured to be a consumer of that content. After you enable your edX instance as an LTI tool provider and configure credentials for the tool consumers, course teams can reuse course content from the edX instance in contexts other than the edX LMS.

11.11.1 Enable LTI Provider Functionality

LTI provider functionality is provided in the `lti_provider` app, located in `edx-platform/lms/djangoapps/lti_provider`.

By default, the `lti_provider` app is not used by edX installations. To enable this functionality throughout the platform, follow these steps.

1. In the `edx/app/edxapp/lms.env.json` file, edit the file so that it includes the following line in the features section.

```
"FEATURES" : {
    ...
    "ENABLE_LTI_PROVIDER": true
}
```

2. Save the `edx/app/edxapp/lms.env.json` file.

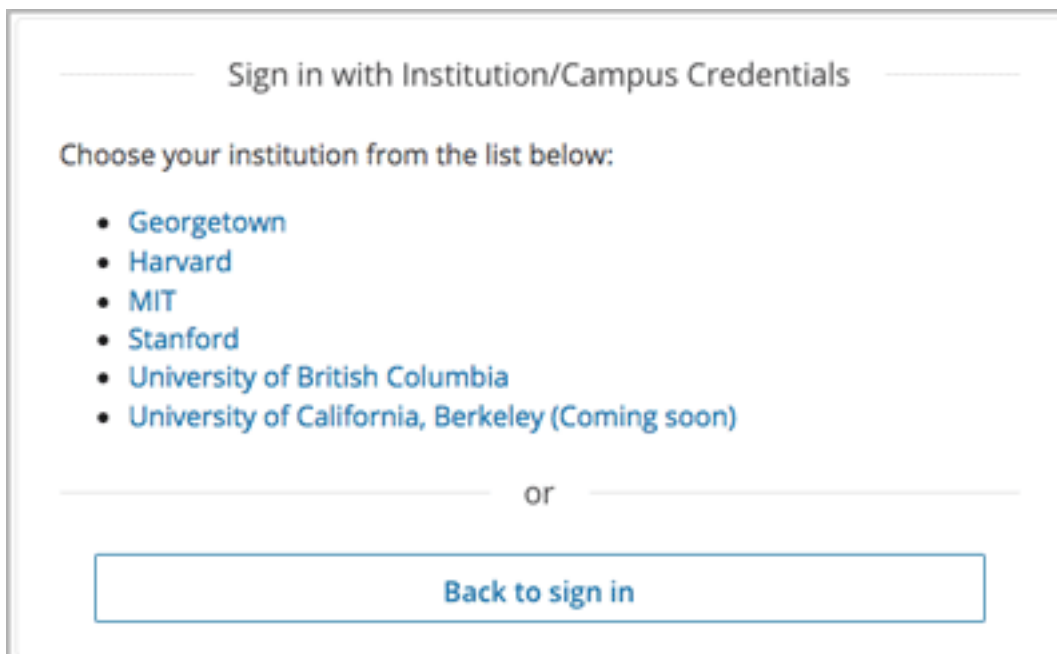3. Run database migrations.

4. Restart the LMS server.

To verify that the LTI provider functionality is enabled, you can check for the presence of the following database tables.

```
lti_provider_gradedassignment
lti_provider_lticonsumer
lti_provider_ltiuser
lti_provider_outcomeservice
```

If these tables are not present, check that the migrations have run properly.

11.11.2 Configuring Credentials for a Tool Consumer

For each external learning management system or application (external LMS) that you want to allow access to your edX instances as an LTI tool consumer, you create OAuth1 credentials, and then configure your edX instance to allow access. Each external LMS that you *configure as a tool consumer* must have separate credentials.

After you complete the configuration of a tool consumer on your edX system, you can add the consumer credentials to your external LMS. For examples of how course teams might set up a course on an external LMS as a consumer of edX course content, see Using Open edX as an LTI Tool Provider in the *Building and Running an edX Course* guide.

Configure the Tool Consumer

To configure an LTI tool consumer to have access to your Open edX installation, follow these steps.

1. Sign in to the Django administration console for your base URL. For example, `http://{your_URL}/admin`.

2. In the **LTI Provider** section, next to **LTI Consumers** select **Add**.

3. Enter the following information.

    - **Consumer Name**: An identifying name for the tool consumer.

    - **Consumer Key**: The console generates a unique key value for this tool consumer. Alternatively, you can use an external application to generate the key, and then enter it here.

    - **Consumer Secret**: The console generates a unique secret value for this tool consumer. Alternatively, you can use an external application to generate the secret, and then enter it here.

Important: Do not supply a value for the **Instance guid** field. The tool consumer generates and supplies a globally unique identifier.

---

4. Select **Save** at the bottom of the page.

11.11.3 Define an Interval for Grade Aggregation (Optional)

When an external LMS links to problem components in a graded edX subsection, the edX system grades the answers to those problems, and then transfers the grades back to the external LMS.

- If the link is to an individual problem component on the edX system, the edX system returns the grade for each learner immediately.

- If the link is to a unit or subsection, you can configure an interval of time for the edX system to delay before returning the grades. The edX system aggregates all of the problems in the unit or subsection that the learner answers during that interval. Aggregating grades can reduce the number of notification messages that learners receive.

By default, the edX system aggregates grades for units and subsections every 15 minutes.

To change the interval for returning aggregated grades, follow these steps.

1. In `edx/app/edxapp/lms.env.json`, change the value for the following parameter.

   `LTI_AGGREGATE_SCORE_PASSBACK_DELAY = 15 * 60`

   You specify a time value in seconds.

2. Save the `/lms/envs/common.py` file.

3. Restart the Learning Management System processes so that the updated environment configurations are loaded.

11.11.4 Options for LTI Authentication and User Provisioning

When you use your Open edX system as an LTI tool provider, data is collected by the Open edX system for all learner activity. Each learner has a user account on the Open edX system that is linked to the user account on the tool consumer system, so that activity, grades, and state can be passed from one system to the other.

The Open edX system supports these user authentication flows for LTI.

- *Anonymous User Authentication*
- *Open edX User Authentication*

Anonymous User Authentication

The first time a learner encounters an Open edX resource in a course, the Open edX content is immediately launched by a POST to the URL. Without requiring any action from the learner, the Open edX system creates a user account and provisions it with a system-generated username, and links it to the tool consumer user account for that learner. Learners never interact with the Open edX system directly.

This authentication flow presents a virtually seamless experience that significantly reduces user error. The Open edX system passes learner data to the tool consumer with no subsequent reconciliation of data between the systems.

After you *configure your edX instance as an LTI tool provider*, no further configuration is needed on your Open edX system for this user authentication flow.

Open edX User Authentication

The first time a learner encounters an Open edX resource in a course, he is prompted to either sign in with existing credentials or create a user account. The Open edX system creates a user account and provisions it with the supplied values, and links it to the tool consumer user account for that learner. The POST to the URL then delivers the Open edX resource in the tool consumer. After the initial sign in or account creation step, learners do not interact with the Open edX system directly.

In this authentication flow, learners knowingly establish or use credentials on the Open edX system. This flow provides a smooth learner experience that can also satisfy legal requirements or privacy concerns.

After you configure your edX instance as an LTI tool provider, you can *configure Open edX user authentication* between your Open edX system and the tool consumer.

11.11.5 Configuring Open edX User Authentication for LTI

Every learner who accesses content on an Open edX system must have a user account. The Open edX system uses the accounts to collect data for learner interactions with courseware.

After you *configure your edX instance as an LTI tool provider*, you can configure Open edX user authentication between your Open edX installation and an LTI tool consumer.

For more information about the authentication flows that are available, see *Options for LTI Authentication and User Provisioning*.

- *Configure Open edX User Authentication for LTI*
- *Test LTI Authentication*

Configure Open edX User Authentication for LTI

To configure Open edX user authentication between your Open edX installation and an LTI tool consumer, follow these steps.

Note: A consumer key and secret are required. The Django administration console provides a hexadecimal string for the secret, but does not provide a hexadecimal string for the key. You must use an external tool to generate the key.

---

1. Sign in to the Django administration console for your base URL. For example, `http://{your_URL}/admin`.

2. In the **Third_Party_Auth** section, next to **Provider Configuration (LTI)** select **Add**.

3. Select **Enabled**.

4. Enter the **Name** of your Open edX system, as you want it to appear on the registration page that is presented to learners who access Open edX content from this LTI tool consumer.

5. To customize the registration process for learners, you can make selections for these optional fields.

   - **Skip Registration Form**: If you select this option, users are not asked to confirm any user account data that is supplied for them by the LTI tool consumer (name, email address, and so on).

     By default, this option is cleared and learners review a registration form with the account details supplied by the tool consumer.

   - **Skip Email Verification**: If you select this option, users are not required to confirm their email addresses, and their accounts are activated immediately upon registration.

     By default, this option is cleared and learners receive an email message and must select a link in that message to activate their user accounts.

6. Enter the following information.

   - **Lti consumer key**: Enter the hexadecimal string of the key.

   - **Lti consumer secret**: The system generates a hexadecimal string value for this field. Alternatively, you can replace it with a secret generated by an external tool.

7. Optionally, change the default value for the **Lti max timestamp age**.

8. Select **Save** at the bottom of the page.

Test LTI Authentication

To verify the sign in process for an LTI provider configuration, follow these steps.

1. Have the LTI consumer key and secret for the LTI provider configuration available. For example, use the Django administration console to open the **Change Provider Configuration (LTI)** page.

2. Use a separate browser window or tab to open the IMS LTI 1.1 Consumer Launch page.

3. As the **Launch URL**, enter your base URL followed by `/auth/login/lti/`. For example, `http://{your_URL}/auth/login/lti/`.

4. Copy the **Lti consumer key** value, and then on the IMS LTI 1.1 Consumer Launch page paste it in as the **Key**.

5. Copy the **Lti consumer secret** value, and then on the IMS LTI 1.1 Consumer Launch page paste it in as the **Secret**.

6. Optionally, change the default values in the **Launch Data** section of the **IMS LTI 1.1 Consumer Launch** page to match the set of values that the tool consumer is configured to supply.

7. To test the workflow for a learner who does not yet have a user account on your Open edX system, follow these steps.

   - Use a separate browser window or tab to make sure that you are signed out of your Open edX LMS.

11.11. Configuring an edX Instance as an LTI Tool Provider 75

- On the **IMS LTI 1.1 Consumer Launch** page, select **Recompute Launch Data** and then select **Press to Launch**.

The page that is configured for delivery to an unauthenticated user loads at the bottom of the page. In the example that follows, the registration page appears (that is, it was not configured to be skipped) and the learner is prompted to complete required fields.

IMS LTI 1.1 Consumer Launch

This is a very simple reference implementaton of the LMS side (i.e. consumer) for IMS LTI 1.1.

Toggle Resource and Launch Data

[Recompute Launch Data] [Reset]

LTI Resource
Launch URL: https:// [_____] /auth/login/lti/
Key: [_____]
Secret: [_____]

Launch Data

[Press to Launch] toggle_debug_data

edX edge   REGISTER   [Sign in]

We couldn't create your account.

- Please enter your Email.
- Please select your Country.

You've successfully signed into [_____]. We just need a little more information before you start learning with edX.

Email *

[username@domain.com]

Full name *

[Jane Q. Public]

Needed for any certificates you may earn

Public username *

[2c8715b735c44075b5aefc15b0a224]

The name that will identify you in your courses - (cannot be changed later)

Country *

[--]

Gender   Year of birth

[--]   [--]

Highest level of education completed

[--]

Mailing address

[   ]

Tell us why you're interested in edX

[   ]

☑ I agree to the edX Terms of Service and Honor Code. *

11.11. Configuring an edX Instance as an LTI Tool Provider   **77**

[Create your account]

* Required field

8. To test the workflow for a learner who already has a user account on your Open edX system, follow these steps.

   - Use a separate browser window or tab to sign in to your Open edX LMS.

   - On the **IMS LTI 1.1 Consumer Launch** page, select **Recompute Launch Data** and then select **Press to Launch**.

   Your Open edX user account is linked to the LTI provider configuration, and your learner dashboard on the Open edX site loads at the bottom of the page. To unlink your user accounts, select the arrow next to your username, and then select **Account**. In the **Connected Accounts** section, select **Unlink** next to the LTI provider configuration name.

For more information and examples of how course teams might set up a course on an external LMS as a consumer of edX course content, see Using Open edX as an LTI Tool Provider in the *Building and Running an edX Course* guide.

11.12 Setting Up the YouTube API Key

This topic describes how to set the YouTube API key for your instance of Open edX.

- *Overview*
- *Get a YouTube API Key*
- *Install the YouTube API Key in Open edX*

11.12.1 Overview

If you intend for courses on your Open edX instance to include videos that are hosted on YouTube, you must get a YouTube API key and set the key in the Open edX Platform.

The Open edX Platform uses the YouTube Data API v3, which requires that the application uses an API key.

11.12.2 Get a YouTube API Key

To get the YouTube API key, follow YouTube's instructions for obtaining authorization credentials. YouTube provides two different options for API keys: server keys or browser keys. You should use a **browser key** for Open edX.

Note: Before proceeding, review *Guidelines for Updating the Open edX Platform*.

11.12.3 Install the YouTube API Key in Open edX

After you obtain a YouTube API key, you must install that key into your Open edX installation. There are two different ways you can do this.

Option 1: Ansible (recommended)

Ansible is the automation system used for installing and updating Open edX. If you set your YouTube API key in Ansible's configuration file, then Ansible will make sure that the YouTube API key remains in place when you update Open edX.

To set your YouTube API key in Ansible's configuration file, complete the following steps.

---

1. Find the configuration repository on your Open edX server. If you are running Devstack or Fullstack, the directory is `/edx/app/edx_ansible/edx_ansible`.

2. In that repository, open the `playbooks/roles/edxapp/defaults/main.yml` file in a text editor.

3. Find the line for the YouTube API key.

   `EDXAPP_YOUTUBE_API_KEY: "PUT_YOUR_API_KEY_HERE"`

   Replace `PUT_YOUR_API_KEY_HERE` with your YouTube API key. Ensure that the YouTube API key is within by quotation marks.

4. Save and close the file.

5. Run Ansible so that it applies your YouTube API key to your Open edX installation. If you are running the Open edX Cypress release, run the following command.

```
    /edx/bin/update edx-platform named-release/cypress
```

Option 2: JSON files

Ansible outputs information to several JSON files used by Open edX. If you prefer not to edit the Ansible configuration, you can edit these files directly.

However, every time you update Open edX, your edits will be overwritten by Ansible. As a result, we recommend setting your YouTube API key in Ansible's configuration instead.

To set your YouTube API key by editing JSON files, complete the following steps.

1. Find the edx-platform repository on your Open edX server. If you are running Devstack or Fullstack, the directory is `/edx/app/edxapp/edx-platform`.

2. In the directory *above* your repository, there should be several JSON files, including `lms.auth.json` and `cms.auth.json`. If you are running Devstack or Fullstack, the directory is `/edx/app/edxapp`.

3. Open the `lms.auth.json` file in your text editor.

4. Find the line for the YouTube API key.

   `"YOUTUBE_API_KEY":  "PUT_YOUR_API_KEY_HERE",`

   Replace `PUT_YOUR_API_KEY_HERE` with your YouTube API key. Verify that the YouTube API key is between the quotation marks.

5. Save and close the file.

6. Open the `cms.auth.json` file and make the same change. If that line does not exist in this file, create it.

7. Save and close the file.

11.13 Configuring Open Response Assessments

You can change the default configuration for the Open Response Assessment (ORA2) application. You can change the default file storage system or change the default set of files that learners are prohibited from submitting.

11.13.1 Configuring ORA2 to Upload Files to Alternative Storage Systems

By default, the Open Response Assessment (ORA2) application stores files that learners upload in an Amazon S3 bucket.

With the Cypress release, you can configure ORA2 to store files in an alternate system. To have learners' files stored in a system other than Amazon S3, follow these steps.

1. In the ORA-2 repository, implement the `BaseBackend` class defined in the base.py file.

   For example, the S3.py file in the same directory is an implementation of `BaseBackend` for Amazon S3. You must implement the equivalent class for the storage system you intend to use.

2. Configure ORA2 to use your alternative storage system by modifying the value of `backend_setting` in init file to point to your implementation of `BaseBackend`.

3. Add code to instantiate the new implementation to the `get_backend()` function in the `init.py` file.

4. Configure ORA2 to use the alternative storage system by modifying the value of `ORA2_FILEUPLOAD_BACKEND` in the Django settings to point to your implementation of `BaseBackend`.

11.13.2 Prohibiting Submission of Specified File Types

Course teams can configure open response assessments so that learners can upload files along with their text responses. During the peer review stage of the assessment, other learners download the submitted file and read the response.

To protect learners from exposure to files with malicious content, the ORA2 application uses a "blacklist" to identify a set of file types that learners are not permitted to upload.

To add or remove file types from the blacklist, follow these steps.

1. In the ORA-2 repository, use an editor to open the submission_mixin.py file.

2. Locate the `FILE_EXT_BLACK_LIST` parameter in the file. By default, this parameter lists the following file types.

```
FILE_EXT_BLACK_LIST = [
    'exe', 'msi', 'app', 'dmg', 'com', 'pif', 'application', 'gadget',
    'msp', 'scr', 'hta', 'cpl', 'msc', 'jar', 'bat', 'cmd', 'vb', 'vbs',
    'jse', 'ws', 'wsf', 'wsc', 'wsh', 'scf', 'lnk', 'inf', 'reg', 'ps1',
    'ps1xml', 'ps2', 'ps2xml', 'psc1', 'psc2', 'msh', 'msh1', 'msh2', 'mshxml',
    'msh1xml', 'msh2xml', 'action', 'apk', 'app', 'bin', 'command', 'csh',
    'ins', 'inx', 'ipa', 'isu', 'job', 'mst', 'osx', 'out', 'paf', 'prg',
    'rgs', 'run', 'sct', 'shb', 'shs', 'u3p', 'vbscript', 'vbe', 'workflow',
    'htm', 'html',
]
```

3. Add or remove values from this list.

4. Save your changes to submission_mixin.py.

5. Restart the Studio (CMS) and Learning Management System (LMS) processes so that your updates are loaded.

For more information and examples of how course teams might set up an open response assessment, see Open Response Assessments in the *Building and Running an Open edX Course* guide.

11.14 Installing an XBlock

The XBlock framework allows developers to expand the Open edX platform by building different learning experiences and deploying them as XBlocks. Before course teams can use an XBlock in courses running on an instance of the Open edX platform, both of the following tasks must be completed.

- A system administrator installs the XBlock in the instance of the Open edX platform.

- Course teams enable the XBlock in the specific courses that will use it.

To install an XBlock, follow these steps.

1. Obtain the Github location and commit for the XBlock.

2. Run `pip` with a Github link to the XBlock.

   An example that installs the Oppia XBlock follows.

```
pip install git+https://github.com/oppia/xblock.git@9f6b95b7eb7dbabb96b77198a3202604f96adf65#egg
```

The course teams that want to include components that use the XBlock can then enable the XBlock for their courses. To do so, they add the name specified in the XBlock's `setup.py` file to each course's advanced module list. For more information, see Enabling Additional Exercises and Tools.

Installing and Configuring edX Insights

The following topics provide information about running edX Insights and its dependencies in a production environment.

12.1 Options for Installing edX Insights

This topic is intended for those who are interested in running edX Insights and its dependencies in a production environment. This topic does not provide complete installation procedures for edX Insights. It presents the introductory material that is available now.

- *Overview*
- *What You Should Know Before You Start*
- *Planning Your Deployment*
- *Example Deployments*

12.1.1 Overview

Course teams use edX Insights to access data gathered from active courses. In edX Insights, course teams can display charts, summary statistics, and data tables.

The Learning Management System (LMS) gathers data about learner activity. This data is aggregated by the edX Analytics Pipeline. The aggregated data is exposed by the edX Analytics Data API. EdX Insights reads the data from the edX Analytics Data API and presents the data to course team members.

Architecture Diagram
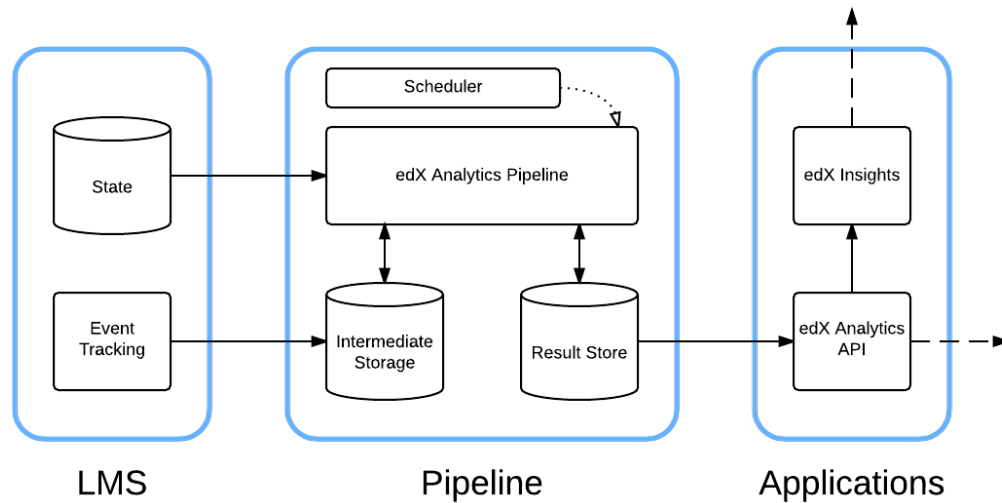
Components

- *LMS*
- *edX Analytics Pipeline*
- *Scheduler*
- *edX Analytics Data API*
- *edX Insights*

LMS

The LMS records learner actions in tracking log files. The standard `logrotate` utility periodically compresses and copies these files into a file system that can be read by the edX Analytics Pipeline. The LMS also captures a lot of information in a MySQL database. The edX Analytics Pipeline connects directly to this database to extract information about learners.

edX Analytics Pipeline

The edX Analytics Pipeline reads the MySQL database used by the LMS as well as the tracking log files produced by the LMS. The data is processed and the resulting summary data is published to the result store. The result store is a MySQL database.

Requirements

- Hadoop version 1.0.3 or higher

- Hive version 0.11.0.2 or higher

- Sqoop version 1.4.5

- Python 2.7

- Either Debian version 6.0 or higher, or Ubuntu version 12.04 or higher

- A MySQL server version 5.6 or higher

Scheduler

The Scheduler schedules the execution of data computation tasks. Data computation tasks are run by the edX Analytics Pipeline. Data computation tasks are used to update parts of the result store.

edX Analytics Data API

The ref:*opendataapi:edX Analytics Data API<edX Data Analytics API Overview>* provides an HTTP interface for accessing data in the result store. Typically, the data in the result store is updated periodically by the edX Analytics Pipeline.

Requirements   Python 2.7

edX Insights

EdX Insights uses the edX Analytics Data API to present data to users. Users access the data using a supported web browser. EdX Insights communicates directly with the LMS to authenticate users, authorize users, and read course structure information.

Requirements   Python 2.7

12.1.2 What You Should Know Before You Start

To install edX Insights and deploy the edX Analytics Pipeline, you must understand the following concepts.

- Basic terminal usage.

- How the LMS has been deployed and configured.

- Basic computer network terminology.

- YAML file format.

If you plan to use Amazon Web Services, an understanding of AWS terminology is also required.

12.1.3 Planning Your Deployment

All edX Analytics services are designed to be relocatable. This means that they do not require a particular configuration of virtual servers. You are free to choose how the services should be distributed among the resources you have available.

Hadoop

Most of the computation performed by the edX Analytics Pipeline is implemented as Map Reduce jobs that must be executed by a Hadoop cluster. You can scale your Hadoop cluster based on your current and projected data sizes. Hadoop clusters can be scaled vertically and horizontally as your data grows. For very small installations of Open edX, a single virtual server should be sufficiently powerful to process your data.

Amazon's Elastic MapReduce (EMR) service offers preconfigured Hadoop clusters. If you are able to use Amazon Web Services (AWS), use of this service is recommended. Proper installation and configuration of Hadoop can be time consuming. For more information, see *Using Elastic MapReduce on AWS*.

Additionally, vendors such as Cloudera and MapR offer simplified Hadoop administration experiences.

Hadoop is a distributed system that consists of several different services. It is worth noting that they are Java services and require a non-trivial amount of memory to run. The high memory requirement might prevent you from running all services on the same virtual server if it does not have enough memory available.

edX Applications

The edX Analytics Data API responds to a small number of requests every time a page is loaded in edX Insights. Small installations can probably host both services on the same virtual server. Larger installations should consider hosting the services on more than one virtual server. A load balancer is recommended for each service that requires more than one virtual server.

Result Store

The results of computations performed by the edX Analytics Pipeline are stored in a MySQL database. Even small installations should use a different MySQL server than the one used by the LMS. The edX Analytics Pipeline's write patterns to the result store are more I/O intensive than usual. Placing both databases on the same server can degrade the performance of the LMS.

Scheduler

Scheduling executions of the edX Analytics Pipeline can be accomplished in many different ways. Any tool that can periodically execute shell commands should work. The simplest tool that can perform this task is cron. Jenkins is also a good candidate.

12.1.4 Example Deployments

- *Small Scale Using Elastic MapReduce*
- *Large Scale Using Elastic MapReduce*
- *Large Scale Without Using Elastic MapReduce*

Small Scale Using Elastic MapReduce

A small deployment might consist of a single master node and a single core node. The Scheduler is deployed to the master node and periodically executes the edX Analytics Data Pipeline on this server. Additionally, the edX Analytics API, edX Insights and result store are deployed to the master node. These services run continuously. For more information, see *Using Elastic MapReduce on AWS*.

Large Scale Using Elastic MapReduce

A large scale deployment consists of a single master node, several core nodes, and many task nodes deployed into a public subnet of a Virtual Private Cloud.
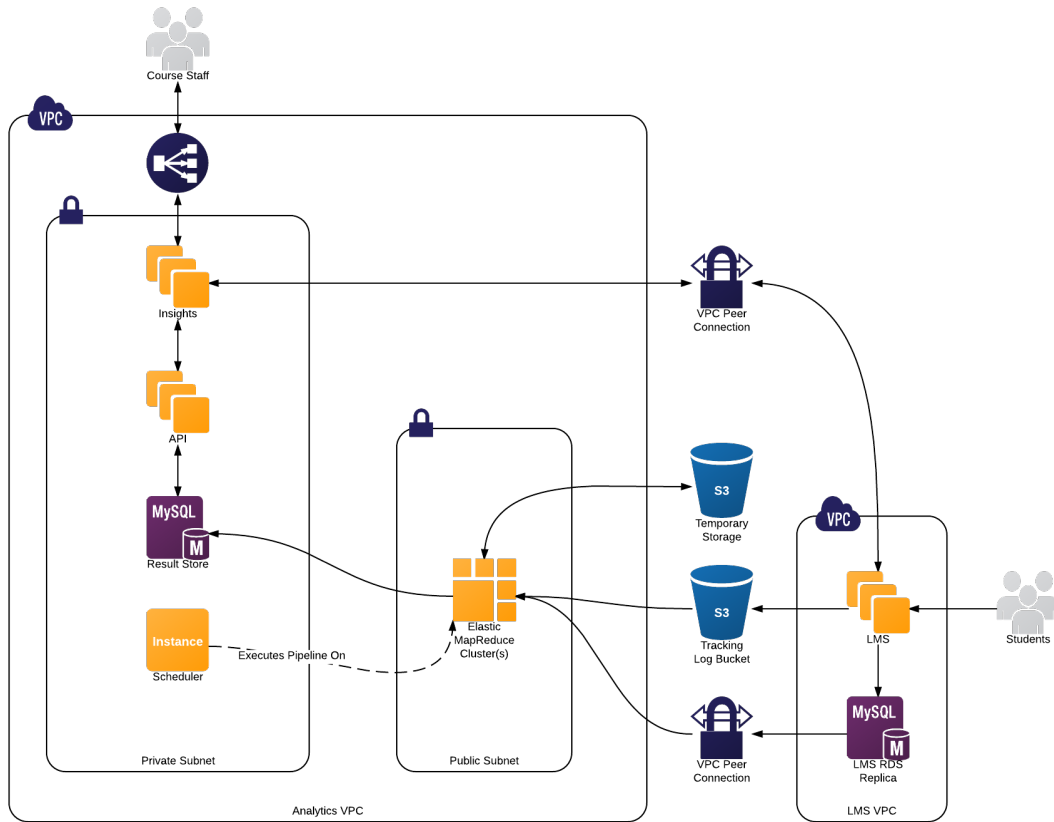
The edX Analytics API and edX Insights are each deployed into an auto-scaling group behind an Elastic Load Balancer which terminates SSL connections and distributes the load among the application servers. The application servers are deployed into a private subnet of the Virtual Private Cloud. A single virtual server is deployed into a private subnet to host the Scheduler. The Relational Database Service is used to deploy a MySQL server into a private subnet. The MySQL database will be used as the result store. For more information, see *Using Elastic MapReduce on AWS*.

Large Scale Without Using Elastic MapReduce

A large deployment that does not use Elastic MapReduce requires additional configuration steps to establish a properly configured environment. A Hadoop cluster is deployed. The master node is considered the node that is running the Job Tracker service for Hadoop 1.X deployments or the Resource Manager service for Hadoop 2.X deployments. Hive and Sqoop are deployed to the master node. Several servers are deployed outside of the Hadoop cluster that host the remainder of the infrastructure. The edX Analytics API and edX Insights services are each deployed to at least one server. The Scheduler is deployed to another server. A MySQL database is deployed to a server that is configured to host a relational database.

12.2 Using Elastic MapReduce on AWS

This topic provides an overview of the components and services that you set up and configure in a deployment that uses Amazon EMR. Work to prepare complete installation procedures for edX Insights is in progress.

> - *Virtual Private Cloud*
> - *Identity and Access Management*
> - *SSH Credentials*
> - *Elastic Compute Cloud*
> - *Relational Database Service*
> - *Scheduler Service*
> - *Simple Storage Solution Buckets*
> - *MySQL Connector Library*
> - *Cluster Configuration File*
> - *EMR Cluster*

For more information about AWS, including detailed procedures, see the AWS Documentation.

Important: The tasks described by this topic rely on the use of third-party services and software. Because the services and software are subject to change by their owners, the information provided here is intended as a guideline and not as exact procedures.

12.2.1 Virtual Private Cloud

Create a Virtual Private Cloud (VPC) with at least one public subnet. A limitation of EMR running in a VPC is that clusters must be deployed into public subnets of VPCs. An existing VPC can be used if one is already available for use.

EdX recommends that you configure the VPC to have at least one private subnet in addition to the required public subnet. A private subnet is not required. For more information, see the example configuration scenario in the *Amazon Virtual Private Cloud User Guide* published by AWS.

An example configuration that includes only a single public subnet can also be found in the *Amazon Virtual Private Cloud User Guide*.

Other Considerations

- To take advantage of price fluctuations in the spot pricing market , you can also deploy several public subnets in different availability zones.

- Consider that the clusters deployed using EMR will need network connectivity to a read replica of the LMS database. EdX recommends that you use the same VPC as the LMS if possible.

- While it is possible to run all of these services outside of a VPC, edX does not recommend doing so.

12.2.2 Identity and Access Management

For Amazon Identity and Access Management (IAM), create an IAM role for use by the EMR cluster. Assign all of the Elastic Compute Cloud (EC2) nodes in the cluster to this role. An option is to consider copying the contents of the default EC2 role for Amazon EMR used by AWS.

Make sure that an IAM user with administrative privileges is available for use.

12.2.3 SSH Credentials

Generate a secure SSH key that you use to access all AWS resources. For example, run the following command.

```
ssh-keygen -t rsa -C "your_email@example.com"
```

Upload the public key from the SSH key pair to AWS and assign a key pair name to it.

12.2.4 Elastic Compute Cloud

Ensure that at least one Amazon Elastic Compute Cloud (EC2) instance is available to host the various services. Depending on the scale of your deployment, additional instances might be necessary.

Make sure to use the secure key pair name to deploy all of the EC2 instances.

12.2.5 Relational Database Service

Deploy a MySQL 5.6 Relational Database Service (RDS) instance. This RDS instance is the result store.

Connectivity

Ensure that there is connectivity between the EC2 instance that hosts the edX Analytics API and your RDS instance.

Also, ensure that instances deployed into the public subnet of the VPC can connect to your RDS instance.

Users

Create at least the following two users on the RDS instance.

- The "pipeline" user must have permission to create databases, tables, and indexes, and issue arbitrary Data Manipulation Language (DML) commands.
- The "api" user must have read-only access.

Configure passwords for both of these users.

Other Considerations

Configure the RDS instance to use "utf8_bin" collation by default for columns in all databases and tables.

12.2.6 Scheduler Service

Establish an SSH connection to the EC2 instance within the VPC that will run the scheduler service. Then, issue the following commands from a shell running on that instance.

1. Check out the sources files from edx-analytics-configuration.

```
git clone https://github.com/edx/edx-analytics-configuration.git
```

2. Configure the shell to use the AWS credentials of the administrative AWS user.

```
        export AWS_ACCESS_KEY_ID=<access key ID goes here>
        export AWS_SECRET_ACCESS_KEY=<secret access key goes here>
```

3. Install the AWS Command Line Interface.

```
        pip install awscli
```

12.2.7 Simple Storage Solution Buckets

Create a Simple Storage Solution (S3) bucket to hold all Hadoop logs from the EMR cluster.

```
    aws s3 mb s3://<your logging bucket name here>
```

Then, create an S3 bucket to hold secure configuration files and initialization scripts.

```
    aws s3 mb s3://<your configuration bucket name here>
```

12.2.8 MySQL Connector Library

Download the MySQL connector library from Oracle. After the download is complete, you then upload it to S3.

```
    aws s3 cp /tmp/mysql-connector-java-5.1.*.tar.gz s3://<your configuration bucket name here>/
```

The `edx-analytics-configuration/batch/bootstrap/install-sqoop` script references a specific version of the MySQL connector library. Update this `install-sqoop` script to point to the correct version of the library in the S3 bucket. You must update the script before you continue.

Then, upload the contents of the `edx-analytics-configuration/batch/bootstrap/` directory into your configuration bucket.

```
    aws s3 sync edx-analytics-configuration/batch/bootstrap/ s3://<your configuration bucket name he
```

12.2.9 Cluster Configuration File

Create a cluster configuration file to specify the parameters for the EMR cluster. Review the parameters that follow, and change them to specify your desired configuration. For example, review the `core` and `task` mappings and change the values for `bidprice` and `type` to meet your needs.

Then, save this file to a temporary location such as `/tmp/cluster.yml`.

```
    {
        name: <your cluster name here>,
        keypair_name: <your keypair name here>,
        vpc_subnet_id: <your VPC public subnet ID here>,
        log_uri: "s3://<your logging bucket name here>",
        instance_groups: {
            master: {
                num_instances: 1,
                type: m1.medium,
                market: ON_DEMAND,
            },
            core: {
                num_instances: 2,
                type: m1.medium,
                market: SPOT,
```

```
            bidprice: 0.03
        },
        task: {
            num_instances: 1,
            type: m1.medium,
            market: SPOT,
            bidprice: 0.03
        }
    },
    bootstrap_actions: {
        security: {
            path: "s3://<your configuration bucket name here>/security.sh"
        },
        jobtrackerconfig: {
            path: "s3://elasticmapreduce/bootstrap-actions/configure-hadoop",
            args: [
                -m, "mapred.jobtracker.completeuserjobs.maximum=5",
                -m, "mapred.job.tracker.retiredjobs.cache.size=50",
                -m, "mapred.job.shuffle.input.buffer.percent=0.20"
            ]
        }
    },
    steps: [
        { type: hive_install },
        {
            type: script,
            name: Install Sqoop,
            step_args: [
                "s3://<your configuration bucket name here>/install-sqoop",
                "s3://<your configuration bucket name here>"
            ]
        }
    ],
    user_info: []
}
```

12.2.10 EMR Cluster

Deploy the EMR cluster.

```
EXTRA_VARS="@/tmp/cluster.yml" make provision.emr
```

Example Output

```
pip install -q -r requirements.txt

ansible-playbook --connection local -i 'localhost,' batch/provision.yml -e "$EXTRA_VARS"

PLAY [Provision cluster] ********************************************************

TASK: [provision EMR cluster] **************************************************
changed: [localhost]

TASK: [add master to group] ***************************************************
ok: [localhost]
```

```
TASK: [display master IP address] *********************************************
ok: [localhost] => {
    "msg": "10.0.1.236"
}

TASK: [display job flow ID] **************************************************
ok: [localhost] => {
    "msg": "j-29UUJVM8P1NPY"
}

PLAY [Configure SSH access to cluster] ****************************************

TASK: [user | debug var=user_info] *******************************************
ok: [10.0.1.236] => {
    "item": "",
    "user_info": []
}

TASK: [user | create the edxadmin group] *************************************
changed: [10.0.1.236]

TASK: [user | ensure sudoers.d is read] **************************************
changed: [10.0.1.236]

TASK: [user | grant full sudo access to the edxadmin group] ******************
changed: [10.0.1.236]

TASK: [user | create the users] **********************************************
skipping: [10.0.1.236]

TASK: [user | create .ssh directory] *****************************************
skipping: [10.0.1.236]

TASK: [user | assign admin role to admin users] ******************************
skipping: [10.0.1.236]

TASK: [user | copy github key[s] to .ssh/authorized_keys2] *******************
skipping: [10.0.1.236]

TASK: [user | copy additional authorized keys] *******************************
skipping: [10.0.1.236]

TASK: [user | create bashrc file for normal users] ***************************
skipping: [10.0.1.236]

TASK: [user | create .profile for all users] *********************************
skipping: [10.0.1.236]

TASK: [user | modify shell for restricted users] *****************************
skipping: [10.0.1.236]

TASK: [user | create bashrc file for restricted users] ***********************
skipping: [10.0.1.236]

TASK: [user | create sudoers file from template] *****************************
changed: [10.0.1.236]

TASK: [user | change home directory ownership to root for restricted users] ***
```

```
     skipping: [10.0.1.236]

     TASK: [user | create ~/bin directory] ****************************************
     skipping: [10.0.1.236]

     TASK: [user | create allowed command links] **********************************
     skipping: [10.0.1.236]

     PLAY RECAP *******************************************************************
     10.0.1.236                 : ok=0    changed=4    unreachable=0    failed=0
     localhost                  : ok=4    changed=1    unreachable=0    failed=0
```

Additional Tasks

To complete the EMR configuration, additional configuration and automation procedures are required, including scheduling jobs and automating log duplication. For more information, see the edX Analytics Installation wiki page.

Work to prepare complete installation procedures for edX Insights is in progress. This section presents the introductory, overview material that is available now.

Setting Up the Open edX Mobile Applications

This section is intended for those are who are building Open edX mobile applications and customizing an Open edX installation to support their use.

- *Accessing the Source Code*
- *Configuring Mobile Application Features*
- *Configuring Video Modules for Mobile*
- *Enabling Push Notifications*

13.1 Accessing the Source Code

There are currently two edX mobile applications, one for iOS and one for Android. You can find the source code and additional documentation for each application here.

- iOS: http://github.com/edx/edx-app-ios

- Android: http://github.com/edx/edx-app-android

13.2 Configuring Mobile Application Features

For the mobile API and authentication to work correctly with the edX mobile applications, you enable them in the `lms.env.json` file. You then complete the setup for authentication by creating OAuth clients and adding their IDs to the custom configuration file of each mobile application.

13.2.1 Enable Mobile Application Features

Note: Configuration settings added to the `lms.env.json` file are reset to their default values when you use Ansible to update edx-platform.

To enable the mobile application features, follow these steps.

1. In the `edx/app/edxapp/lms.env.json` file, add the following lines to the features section.

```
    "FEATURES" : {
        ...
        "ENABLE_MOBILE_REST_API": true,
        "ENABLE_OAUTH2_PROVIDER": true,
        "ENABLE_COMBINED_LOGIN_REGISTRATION": true
    }
```

If you are running in a non-SSL environment, you can set `"OAUTH_ENFORCE_SECURE": false`. This configuration setting should never be disabled in anything other than a development environment.

1. Save the `edx/app/edxapp/lms.env.json` file.

2. Restart the server.

13.2.2 Create the OAuth Clients

You create an OAuth client ID and secret that are specific to your installation for use by the mobile applications. The procedure that follows assumes that you create a different client for each of the edX mobile applications.

Before you can create OAuth clients, a superuser must exist on the server. For more information, see edX Managing the Full Stack.

To create your OAuth clients, follow these steps.

1. Log in to the Django administration console for your base URL. For example, `http://{your_URL}/admin`.

2. In the **Oauth2** section, select **Clients**.

3. Select **Add client**. A dialog box opens with the **Client id** and **Client secret** populated for the new client.

4. Enter a **Url** and **Redirect Url** for the first application. For example, `https://{your_URL}/api/mobile/{version}/?app=ios`. While the console requires values for these fields, they are not used by the edX mobile applications. You can enter the same value in both fields.

5. For the **Client type**, select **Public**.

6. Optionally, enter a **User** and an identifying **Name**.

7. Select **Save and add another**.

8. Repeat steps 4-6 for the second application, and then select **Save**.

13.2.3 Configure the Applications with OAuth Client IDs

The procedure that follows assumes that you have a different OAuth client ID for each edX mobile application.

To configure each edX mobile application with its OAuth client ID, you add a setting to its corresponding custom configuration .yaml file. For information about how to set up custom configuration directories and files, see the GitHub repositories for iOS and Android.

1. Obtain the OAuth client id for the first application from your Django administration console. For more information, see *Create the OAuth Clients*.

2. In your custom GitHub configuration directory, edit the .yaml file for the edX mobile application for iOS. For example, edit your `ios.yaml` file.

3. Add the following line to the .yaml file.

```
      OAUTH_CLIENT_ID: '{client_id_for_iOS_app}'
```

4. Save the file.

5. Repeat steps 1-4 for the edX mobile application for Android.

13.3 Configuring Video Modules for Mobile

Courseware videos must be specifically prepared to ensure that they are in mobile accessible formats. Video modules in mobile-available courses should have low resolution encodings that are readily accessible by mobile devices.

After you obtain a low resolution encoding for a video file and post it online, your course team can add its location to the video module that includes the video in the course structure.

- To configure a video module in edX Studio, you edit the video component. On the **Advanced** tab, in the **Video File URLs** field, enter the URL to the mobile-targeted video as the first URL in the list. For more information, see Working with Video Components in *Building and Running an Open edX Course*.

- To configure a video module by editing the course XML, enter the URL to the mobile-targeted video as the first URL in the list of `html5_sources`. For more information, see Video Components in the *edX Open Learning XML Guide*.

13.4 Enabling Push Notifications

You enable push notifications on the server and then configure each of the edX mobile applications. Currently, you can use Parse for notification delivery.

The procedures that follow assume that you have obtained an application ID, REST API key, and client key from Parse.

13.4.1 Enable Push Notification on the Server

To enable the push notification feature, follow these steps.

1. In the `edx/app/edxapp/cms.auth.json` file, add the following lines.

```
PARSE_KEYS = {

  "APPLICATION_ID": "{app_id}",

  "REST_API_KEY": "{API_key}"

}
```

2. Save the `edx/app/edxapp/cms.auth.json` file.

3. Restart the server.

4. Log in to the Django administration console for your Studio URL. For example, `http://studio.{your_URL}/admin`.

5. In the **Contentstore** section, select **Push notification configs**.

6. Select **Add push notification config**.

7. Verify that **Enabled** is selected, and then select **Save**.

13.4.2 Configure Push Notification on the Clients

The procedure that follows assumes that you use a single set of Parse credentials for both of the edX mobile applications.

You add push notification settings to the applicable custom configuration .yaml file. For information about how to set up custom configuration directories and files, see the GitHub repositories for iOS and Android.

1. In your custom GitHub configuration directory, edit the .yaml file that stores configuration settings that are common to both of the edX mobile applications. For example, edit your `shared.yaml` file.

2. Add the following lines to the .yaml file.

```
PARSE:
  NOTIFICATIONS_ENABLED: true
  APPLICATION_ID: {your application id}
  CLIENT_KEY: {your client key}

PUSH_NOTIFICATIONS: true
```

3. Save the file.

Index of EdX Feature Flags

The following list includes most feature flags that are available in the edX platform.

Name	Description	Default Value
ACCESS_REQUIRE_STAFF_FOR_COURSE	Supported	FALSE
ADVANCED_SECURITY	Supported	TRUE
ALLOW_COURSE_STAFF_GRADE_DOWNLOADS	Supported	FALSE
AUTH_USE_CAS	Supported	TRUE
AUTH_USE_CERTIFICATES	Supported	FALSE
AUTH_USE_OPENID	Supported	FALSE
AUTH_USE_OPENID_PROVIDER	Supported	FALSE
AUTH_USE_SHIB	Supported	FALSE
AUTOMATIC_AUTH_FOR_TESTING	Development	FALSE
AUTOMATIC_VERIFY_STUDENT_IDENTITY_FOR_TESTING	Development	FALSE
CERTIFICATES_HTML_VIEW	Supported	FALSE
CERTIFICATES_INSTRUCTOR_GENERATION	Supported	FALSE
CUSTOM_COURSES_EDX	Supported	FALSE
DEBUG_LEVEL	Deprecated	0
DISABLE_LOGIN_BUTTON	Supported	FALSE
DISPLAY_ANALYTICS_DEMOGRAPHICS	Deprecated	TRUE
DISPLAY_ANALYTICS_ENROLLMENTS	Deprecated	TRUE
DISPLAY_DEBUG_INFO_TO_STAFF	Supported	TRUE
EMBARGO	Supported	FALSE
ENABLE_CORS_HEADERS	Supported	FALSE
ENABLE_COSMETIC_DISPLAY_PRICE	Supported	FALSE
ENABLE_COURSE_DISCOVERY	Supported	FALSE
ENABLE_COURSEWARE_SEARCH	Supported	FALSE
ENABLE_CREDIT_API	Supported	FALSE
ENABLE_DASHBOARD_SEARCH	Supported	FALSE
ENABLE_DISABLING_XBLOCK_TYPES	Supported	TRUE
ENABLE_DISCUSSION_EMAIL_DIGEST	Supported	FALSE
ENABLE_DISCUSSION_HOME_PANEL	Supported	FALSE
ENABLE_DISCUSSION_SERVICE	Supported	TRUE
ENABLE_DJANGO_ADMIN_SITE	Supported	TRUE
ENABLE_EDXNOTES	Supported	FALSE
ENABLE_FEEDBACK_SUBMISSION	Supported	FALSE
ENABLE_INSTRUCTOR_ANALYTICS	Deprecated	FALSE
ENABLE_INSTRUCTOR_BACKGROUND_TASKS	Supported	TRUE
	Continued on next page	

Table 14.1 – continued from previous page

Name	Description	Default Value
ENABLE_INSTRUCTOR_EMAIL	Supported	TRUE
ENABLE_INSTRUCTOR_LEGACY_DASHBOARD	Deprecated	FALSE
ENABLE_LMS_MIGRATION	Deprecated	FALSE
ENABLE_MANUAL_GIT_RELOAD	Supported	FALSE
ENABLE_MAX_FAILED_LOGIN_ATTEMPTS	Supported	TRUE
ENABLE_MAX_SCORE_CACHE	Supported	TRUE
ENABLE_MKTG_SITE	Supported	FALSE
ENABLE_MOBILE_REST_API	Supported	FALSE
ENABLE_OAUTH2_PROVIDER	Supported	FALSE
ENABLE_ONLOAD_BEACON	Supported	FALSE
ENABLE_OPENBADGES	Supported	FALSE
ENABLE_PAID_COURSE_REGISTRATION	Supported	FALSE
ENABLE_PREREQUISITE_COURSES	Supported	FALSE
ENABLE_PSYCHOMETRICS	Deprecated	FALSE
ENABLE_RENDER_XBLOCK_API	Removed	FALSE
ENABLE_S3_GRADE_DOWNLOADS	Supported	FALSE
ENABLE_SHOPPING_CART	Supported	FALSE
ENABLE_STUDENT_HISTORY_VIEW	Supported	TRUE
ENABLE_STUDENT_NOTES	Supported	TRUE
ENABLE_TEAMS	Supported	FALSE
ENABLE_TEXTBOOK	Supported	TRUE
ENABLE_THIRD_PARTY_AUTH	Supported	FALSE
ENABLE_VIDEO_BEACON	Supported	FALSE
ENABLE_VIDEO_BUMPER	Supported	FALSE
ENABLE_XBLOCK_VIEW_ENDPOINT	Supported	FALSE
ENFORCE_PASSWORD_POLICY	Supported	TRUE
ENTRANCE_EXAMS	Supported	FALSE
FORCE_UNIVERSITY_DOMAIN	Deprecated	FALSE
INDIVIDUAL_DUE_DATES	Supported	FALSE
LICENSING	Supported	FALSE
LOG_POSTPAY_CALLBACKS	Supported	TRUE
MAX_ENROLLMENT_INSTR_BUTTONS	Supported	200
MILESTONES_APP	Supported	FALSE
MODE_CREATION_FOR_TESTING	Development	FALSE
PREVENT_CONCURRENT_LOGINS	Supported	TRUE
REQUIRE_COURSE_EMAIL_AUTH	Supported	TRUE
RUN_AS_ANALYTICS_SERVER_ENABLED	Deprecated	FALSE
SEGMENT_IO_LMS	Deprecated	FALSE
SHIB_DISABLE_TOS	Supported	FALSE
SHOW_BUMPER_PERIODICITY	Supported	7 * 24 * 3600
STORE_BILLING_INFO	Supported	FALSE
SUBDOMAIN_BRANDING	Deprecated	FALSE
SUBDOMAIN_COURSE_LISTINGS	Deprecated	FALSE
USE_DJANGO_PIPELINE	Supported	TRUE
USE_MICROSITES	Supported	FALSE

Glossary

15.1 A

A/B Test

> See *Content Experiment*.

About Page

> The course page that provides potential students with a course summary, prerequisites, a course video and image, and important dates.

> For more information, see The Course Summary Page.

Accessible Label

> The descriptive, identifying name that you supply when you add a problem component to your course. All problems require accessible labels.

> For more information, see Creating Exercises and Tools.

Advanced Editor

> An XML-only editor in a problem component that allows you to can create and edit any type of problem. For more information, see The Advanced Editor.

Assignment Type

> The category of graded student work, such as homework, exams, and exercises.

> For more information, see Establishing a Grading Policy.

15.2 C

Capa Problem

> Any of the problem types implemented in the edX platform by the `capa_module` XBlock. Examples range from text input, drag and drop, and math expression input problem types to circuit schematic builder, custom JavaScript, and chemical equation problem types.

> Other assessment methods are also available, and implemented using other XBlocks. An open response assessment is an example of a non-capa problem type.

Certificate

A document issued to an enrolled student who successfully completes a course. Not all edX courses offer certificates, and not all students enroll as certificate candidates.

Chapter

See *Section*.

Checkbox Problem

A problem that prompts the student to select one or more options from a list of possible answers. For more information, see Checkbox Problem.

Chemical Equation Response Problem

A problem that allows the student to enter chemical equations as answers. For more information, see Chemical Equation Problem.

Circuit Schematic Builder Problem

A problem that allows the student to construct a schematic answer (such as an electronics circuit) on an interactive grid.

For more information, see Circuit Schematic Builder Problem.

Closed Captions

See *Transcript*.

Cohort

A group of students who participate in a class together. Students who are in the same cohort group can communicate and share experiences in private discussions.

Cohorts are an optional feature of courses on the edX platform. For information about how you enable the cohort feature, set up cohorts, and assign students to them, see Using Cohorts in Your Courses.

Component

The part of a unit that contains your actual course content. A unit can contain one or more components. For more information, see Developing Course Components.

Content Experiment

You can define alternative course content to be delivered to different, randomly assigned groups of students. Also known as A/B or split testing, you use content experiments to compare the performance of students who have been exposed to different versions of the content. For more information, see Creating Content Experiments.

Content Library

See *Library*.

Content-Specific Discussion Topic

A category within the course discussion that appears at a defined point in the course to encourage questions and conversations. To add a content-specific discussion topic to your course, you add a discussion component to a unit. Students cannot contribute to a content-specific discussion topic until the release date of the section that contains it.

For more information, see Working with Discussion Components and Creating Discussion Topics for Your Course.

Course Catalog

The page that lists all courses offered in the edX learning management system.

Course Handouts

Course handouts are files you make available to students in the Course Info page.

For more information, see Add Course Handouts.

Course Info Page

The page that opens first every time students access your course. You can post announcements on the Course Info page. The Course Handouts sidebar appears in the right pane of this page.

Course Run

The term or time frame in which a specific offering of your course takes place. You set the course run when you create your course. For more information, see Create a New Course.

Courseware

The page where students access the primary instructional materials for your course. Sections, subsections, units, and components are all accessed from the Courseware page.

Course-Wide Discussion Topic

Optional categories that you create to guide how students find and share information in the course discussion. Examples of course-wide discussion topics include Announcements and Frequently Asked Questions. Students can contribute to these topics as soon as your course starts.

For more information, see Creating Discussion Topics for Your Course.

Custom Response Problem

A custom response problem evaluates text responses from students using an embedded Python script. These problems are also called "write-your-own- grader" problems. For more information, see Write-Your-Own-Grader Problem.

15.3 D

Data Czar

A data czar is the single representative at a partner institution who is responsible for receiving course data from edX, and transferring it securely to researchers and other interested parties after it is received.

For more information, see the edX Research Guide.

Discussion

The set of topics defined to promote course-wide or unit-specific dialog. Students use the discussion topics to communicate with each other and the course team in threaded exchanges.

For more information, see Managing Course Discussions.

Discussion Component

Discussion topics that course teams add directly to units. For example, a video component can be followed by a discussion component so that students can discuss the video content without having to leave the page. When you add a discussion component to a unit, you create a content-specific discussion topic.

For more information, see Working with Discussion Components.

Dropdown Problem

A problem that asks students to choose from a collection of answer options, presented as a drop-down list. For more information, see Dropdown Problem.

15.4 E

edX101

An online course about how to create online courses. The intended audience for edX101 is faculty and university administrators.

edX Edge

Edge is a less restricted site than edX.org. While only edX employees and consortium members can create and post content on edX.org, any users with course creator permissions for Edge can create courses with Studio on studio.edge.edx.org, then view the courses on the learning management system at edge.edx.org.

edX Studio

The edX tool that you use to build your courses.

For more information, see What is Studio?.

Exercises

Practice or practical problems interspersed in edX course content to keep the learner engaged. Exercises are also an important measure of teaching effectiveness and learner comprehension.

Export

A tool in edX Studio that you use to export your course or library for backup purposes, or so that you can edit the course or library directly in XML format. See also *Import*.

For more information, see Export a Course or Export a Library.

15.5 F

Forum

See *Discussion*.

15.6 G

Grade Range

Thresholds that specify how numerical scores are associated with grades, and the score a student must obtain to pass a course.

For more information, see Set the Grade Range.

Grading Rubric

See *Rubric*.

15.7 H

HTML Component

A type of component that you can use to add and format text for your course. An HTML component can contain text, lists, links, and images.

For more information, see Working with HTML Components.

15.8 I

Image Mapped Input Problem

A problem that presents an image and accepts clicks on the image as an answer.

For more information, see Image Mapped Input Problem.

Import

A tool in edX Studio that you use to load a course or library in XML format into your existing course or library. When you use the Import tool, Studio replaces all of your existing course or library content with the content from the imported course or library. See also *Export*.

For more information, see Import a Course or Import a Library.

15.9 K

Keyword

A variable in a bulk email message. When you send the message, a value that is specific to the each recipient is substituted for the keyword.

15.10 L

Label

See *Accessible Label*.

LaTeX

A document markup language and document preparation system for the TeX typesetting program.

In edX Studio, you can import LaTeX code.

You can also create a problem written in LaTeX.

Learning Management System (LMS)

The platform that students use to view courses, and that course team members use to manage learner enrollment, assign team member privileges, moderate discussions, and access data while the course is running.

Learning Sequence

The horizontal navigation bar that appears at the top of the **Courseware** page in the LMS. The learning sequence contains an icon for each unit in the selected subsection. When a learner moves the cursor over one of these icons, the names of each component in that unit appear.

Left Pane

The navigation frame that appears at the left side of the **Courseware** page in the LMS. The left pane shows the sections in the course. When you click a section, the section expands to show subsections.

Library

A pool of components for use in randomized assignments that can be shared across multiple courses from your organization. Course teams configure randomized content blocks in course outlines to reference a specific library and randomly provide a specified number of problems from that library to each student.

For more information, see Libraries Overview.

Live Mode

A view that allows the course team to review all published units as students see them, regardless of the release dates of the section and subsection that contain the units.

For more information, see View Your Live Course.

LON-CAPA

The LearningOnline Network with Computer-Assisted Personalized Approach e-learning platform. The structure of capa problem types in the edX platform is based on the LON-CAPA assessment system, although they are not compatible.

See *Capa Problems*.

15.11 M

Math Expression Input Problem

A problem that requires students to enter a mathematical expression as text, such as e=m*c^2.

For more information, see Entering Mathematical and Scientific Expressions.

MathJax

A LaTeX-like language that you use to write equations. Studio uses MathJax to render text input such as x^2 and sqrt(x^2-4) as "beautiful math."

For more information, see A Brief Introduction to MathJax in Studio.

Module

An item of course content, created in an XBlock, that appears on the **Courseware** page in the edX learning management system. Examples of modules include videos, HTML-formatted text, and problems.

Module is also used to refer to the structural components that organize course content. Sections, subsections, and units are modules; in fact, the course itself is a top-level module that contains all of the other course content as children.

Multiple Choice Problem

A problem that asks students to select one answer from a list of options.

For more information, see Multiple Choice Problem.

15.12 N

Numerical Input Problem

A problem that asks students to enter numbers or specific and relatively simple mathematical expressions.

For more information, see Numerical Input Problem.

15.13 O

Open Response Assessment

A type of assignment that allows learners to answer with text, as in a short essay and, optionally, an image or other file. Learners then evaluate each others' work by comparing each response to a rubric created by the course team.

These assignments can also include a self assessment, in which learners compare their own responses to the rubric.

For more information, see Open Response Assessments.

15.14 P

Pages

Pages organize course materials into categories that students select in the learning management system. Pages provide access to the courseware and to tools and uploaded files that supplement the course. Each page appears in your course's navigation bar.

For more information, see Adding Pages to a Course.

Partner Manager

Each EdX partner institution has an edX partner manager. The partner manager is the primary contact for the institution's course teams.

Pre-Roll Video

A short video file that plays before the video component selected by the learner. Pre-roll videos play automatically, on an infrequent schedule.

For more information, see Adding a Pre-Roll Video.

Preview Mode

A view that allows you to see all the units of your course as students see them, regardless of the unit status and regardless of whether the release dates have passed.

For more information, see Preview Course Content.

Problem Component

A component that allows you to add interactive, automatically graded exercises to your course content. You can create many different types of problems.

For more information, see Working with Problem Components.

Progress Page

The page in the learning management system that shows students their scores on graded assignments in the course.

15.15 Q

Question

A question is a type of contribution that you can make to a course discussion topic to bring attention to an issue that the discussion moderation team or other students can resolve.

For more information, see Managing Course Discussions.

15.16 R

Rubric

A list of the items that a student's response should cover in an open response assessment.

For more information, see Rubric.

15.17 S

Section

The topmost category in your course outline. A section can represent a time period or another organizing principle for course content. A section contains one or more subsections.

For more information, see Developing Course Sections.

Sequential

See *Subsection*.

Short Course Description

The description of your course that appears on the edX Course List page.

For more information, see Describe Your Course.

Simple Editor

The graphical user interface in a problem component that contains formatting buttons and is available for some problem types. For more information, see The Studio View of a Problem.

Split Test

See *Content Experiment*.

Subsection

A division in the course outline that represents a topic in your course, such as a lesson or another organizing principle. Subsections are defined inside sections and contain units.

For more information, see Developing Course Subsections.

15.18 T

Text Input Problem

A problem that asks the student to enter a line of text, which is then checked against a specified expected answer.

For more information, see Text Input Problem.

Transcript

A text version of the content of a video. You can make video transcripts available to students.

For more information, see Working with Video Components.

15.19 U

Unit

A unit is a division in the course outline that represents a lesson. Learners view all of the content in a unit on a single page.

For more information, see Developing Course Units.

15.20 V

Vertical

See *Unit*.

Video Component

A component that you can use to add recorded videos to your course.

For more information, see Working with Video Components.

15.21 W

Wiki

The page in each edX course that allows both students and members of the course team to add, modify, or delete content.

Students can use the wiki to share links, notes, and other helpful information with each other.

For more information, see Hide or Show the Course Wiki Page.

15.22 XYZ

XBlock

EdX's component architecture for writing courseware components: XBlocks are the components that deliver course content to learners.

Third parties can create components as web applications that can run within the edX learning management system.

XSeries

A set of related courses in a specific subject. Learners qualify for an XSeries certificate when they pass all of the courses in the XSeries.

For more information, see https://www.edx.org/xseries.