Deploy the Kaltura Video Connector for AEM

Content updated: 12/08/2025

This article is designated for administrators.

About

This article explains how to deploy the **Kaltura Video Connector for Adobe Experience Manager (AEM)**.

The connector is delivered as a standard AEM package and is designed for **AEM as a Cloud Service**. Deployment is typically performed by an AEM administrator, developer, or implementation partner as part of the project's build and deployment process.

This article covers:

- Installing the Kaltura connector package
- Deploying through standard AEM workflows
- Required post-deployment infrastructure configuration

After deployment, you must complete connector setup in AEM before authors can use the Kaltura Video component. For those steps, see Configure Kaltura settings for an AEM site.

Deployment options

The connector is delivered as an installable AEM package (`.zip`). You can deploy it using one of the following methods.

Install using Maven (recommended)

This method integrates the connector into your project build and CI/CD pipeline.

1. Install the package to your local Maven repository

Since AEM Cloud Service cannot access external local files during the build process, third-party packages must be hosted within a project-level repository. This ensures the package is available to the Maven reactor during the Cloud Manager deployment pipeline.

Initialize the Local Repository

Before referencing the package in your project's POM, you must install the `.all` package

Copyright © 2025 Kaltura Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this document constitutes acceptance of the Kaltura Terms of Use and Privacy Policy.

into your local Maven repository (`.m2`).

Run the following command, ensuring the -Dfile path points to your local package location (e.g., /localrepo) within your project structure:

```
mvn install:install-file \
  -Dfile=/path/to/your/package/aem-kalturaaemintegration-project.all-0.0.1-SNAPSHOT.zip \
  -DgroupId=com.kaltura \
  -DartifactId=aem-kalturaaemintegration-project.all \
  -Dversion=0.0.1-SNAPSHOT \
  -Dpackaging=zip
  -DlocalRepositoryPath=./localrepo
```

This converts the .zip file into a Maven-readable format.

Configure the Project Object Model (POM)

To make Maven aware of this internal storage, register the directory as a repository in your **root pom.xml**. Insert the following configuration immediately after the `<description>` tag:

```
<repositories>
   <repository>
      <id>local-vendor</id>
      <url>file://${project.basedir}/localrepo</url>
   </repository>
</repositories>
```

2. Add the dependency to your project

To include the Kaltura integration in your project build, update the all/pom.xml file with the following configurations.

```
<dependency>
   <groupId>com.kaltura</groupId>
   <artifactId>aem-kalturaaemintegration-project.all</artifactId>
   <version>0.0.1-SNAPSHOT</version>
   <type>zip</type>
</dependency>
```

3. Embed the package during the build

To ensure the package is embedded and deployed correctly during the build process, add the following **<embedded>** configuration to the **filevault-package-maven-plugin** inside the **<build>** section:

Copyright © 2025 Kaltura Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this document constitutes acceptance of the Kaltura Terms of Use and Privacy Policy.

```
<plugin>
   <groupId>org.apache.jackrabbit</groupId>
   <artifactId>filevault-package-maven-plugin</artifactId>
   <configuration>
      <embeddeds>
         <embedded>
            <groupId>com.kaltura</groupId>
            <artifactId>aem-kalturaaemintegration-project.all</artifactId>
            <type>zip</type>
            <target>/apps/{your-project-identifier}-packages/application/install</target>
         </embedded>
      </embeddeds>
   </configuration>
</plugin>
```

4. Build and deploy

Once the configurations are updated, deploy the project to your local AEM instance using the following Maven command:

```
mvn clean install -PautoInstallSinglePackage
```

Install using AEM Package Manager

This method involves uploading the compiled .zip artifact directly to the AEM CRX repository.

1. Access the package manager

Navigate to the AEM Package Manager via the global navigation or directly at the following URL:

`{AEM Author URL}/crx/packmgr/index.jsp`

2. Upload the package

1. Select **Upload Package**.
2. In the dialog, browse to the local directory containing the Kaltura integration file: `aem-kalturaaemintegration-project.all-0.0.1-SNAPSHOT.zip`
3. Click **OK** to upload the package to the repository.

3. Install the package

1. Locate the uploaded package in the list (typically under the `com.kaltura` group).
2. Click **Install**.
3. Confirm the installation in the dialog box.

Once the activity log displays "Package installed", the connector is deployed and Kaltura components and services will be available for use.

Copyright © 2025 Kaltura Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this document constitutes acceptance of the Kaltura Terms of Use and Privacy Policy.

Post-deployment requirements

Dispatcher configuration (if applicable)

The following configuration is required to ensure proper communication between the Kaltura integration and the AEM Publish environment when accessed through a web browser.

>
> This step is only required if your environment utilizes an AEM Dispatcher (e.g., Stage, Production, or a local Dispatcher SDK). If you are testing directly on a local Author or Publish instance (e.g., `localhost:4502` or `localhost:4503`), this configuration can be bypassed.

Allow the X-Referer-Path request header

To support tracking and routing, the Dispatcher must allow the **X-Referer-Path** header to pass through. By default, custom headers are blocked and must be explicitly whitelisted.

1. Open the following file in your Dispatcher configuration:

   ```
   dispatcher/src/conf.dispatcher.d/clientheaders/clientheaders.any
   ```

2. Add the header to the allowed list:

   ```
   $include "./default_clientheaders.any"


   # Custom headers for Kaltura integration
   "X-Referer-Path"
   ```

3. Commit the change to your Git repository and deploy it using your Cloud Manager pipeline.

Verify deployment

After deployment is complete:

- Confirm that the Kaltura Cloud Service is available in AEM
- Confirm that the Kaltura Video component appears in the AEM component list

At this stage, the connector is successfully deployed.

Copyright © 2025 Kaltura Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this document constitutes acceptance of the Kaltura Terms of Use and Privacy Policy.

To complete setup and enable authoring, continue with Setup guide for the Kaltura connector for AEM (Adobe Experience Manager).

Copyright © 2025 Kaltura Inc. All Rights Reserved. Designated trademarks and brands are the property of their respective owners. Use of this document constitutes acceptance of the Kaltura Terms of Use and Privacy Policy.