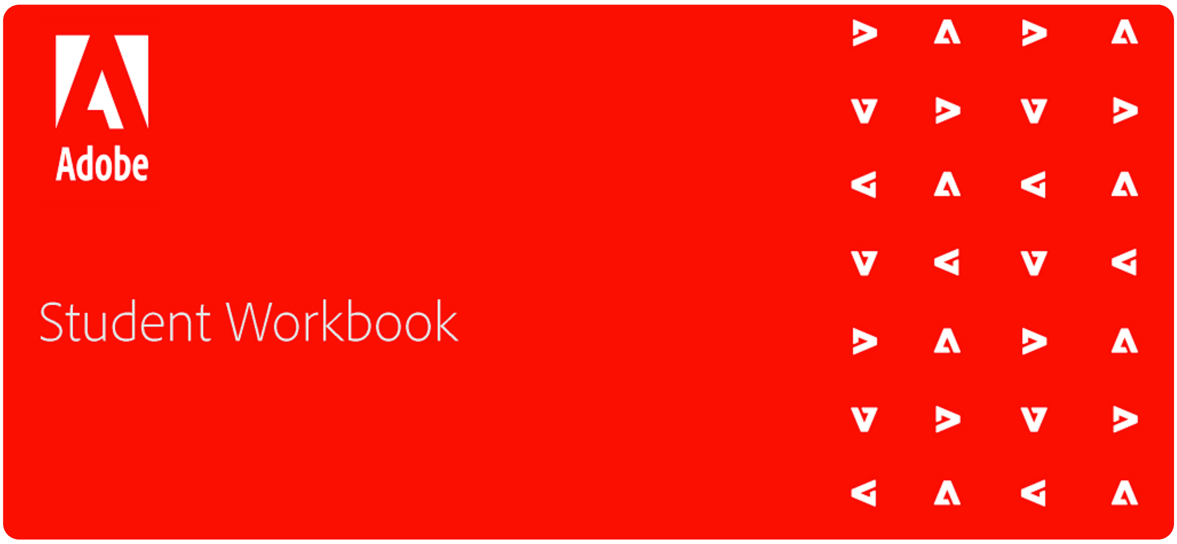
DevOps for AEM as a Cloud Service

Work with Cloud Manager Environments



Adobe

Student Workbook

Find your journey at learning.adobe.com >

Contents

DevOps for AEM as a Cloud Service

Adobe, 345 Park Avenue, San Jose, California 95110, USA.

10-14-2020

Introduction

With Cloud Manager, you can deliver content quickly and efficiently by leveraging the built-in Content Delivery Network (CDN). It helps deploy custom code to the cloud environments and transforms locally assembled content packages into an artifact conforming to the Sling Feature Model.

Objectives

After completing this module, you will be able to:

- Explain the different environments types and their capabilities
- Explain Adobe Experience Manager (AEM) logs
- Download AEM logs
- Explain Continuous Integration (CI)/Continuous Deployment (CD) pipelines
- Create an auto-executing Code Quality Pipeline
- Access and explore the Developer console
- Explain how content packages are deployed through Package Manager
- Deploy content packages that contain immutable content and content packages that contain mutable content

Cloud Manager Environments 1

An environment is an AEM application deployment within a program. For AEM, an environment consists of:

- An author service (optionally with an additional cold standby author service)
- Zero or more publish services

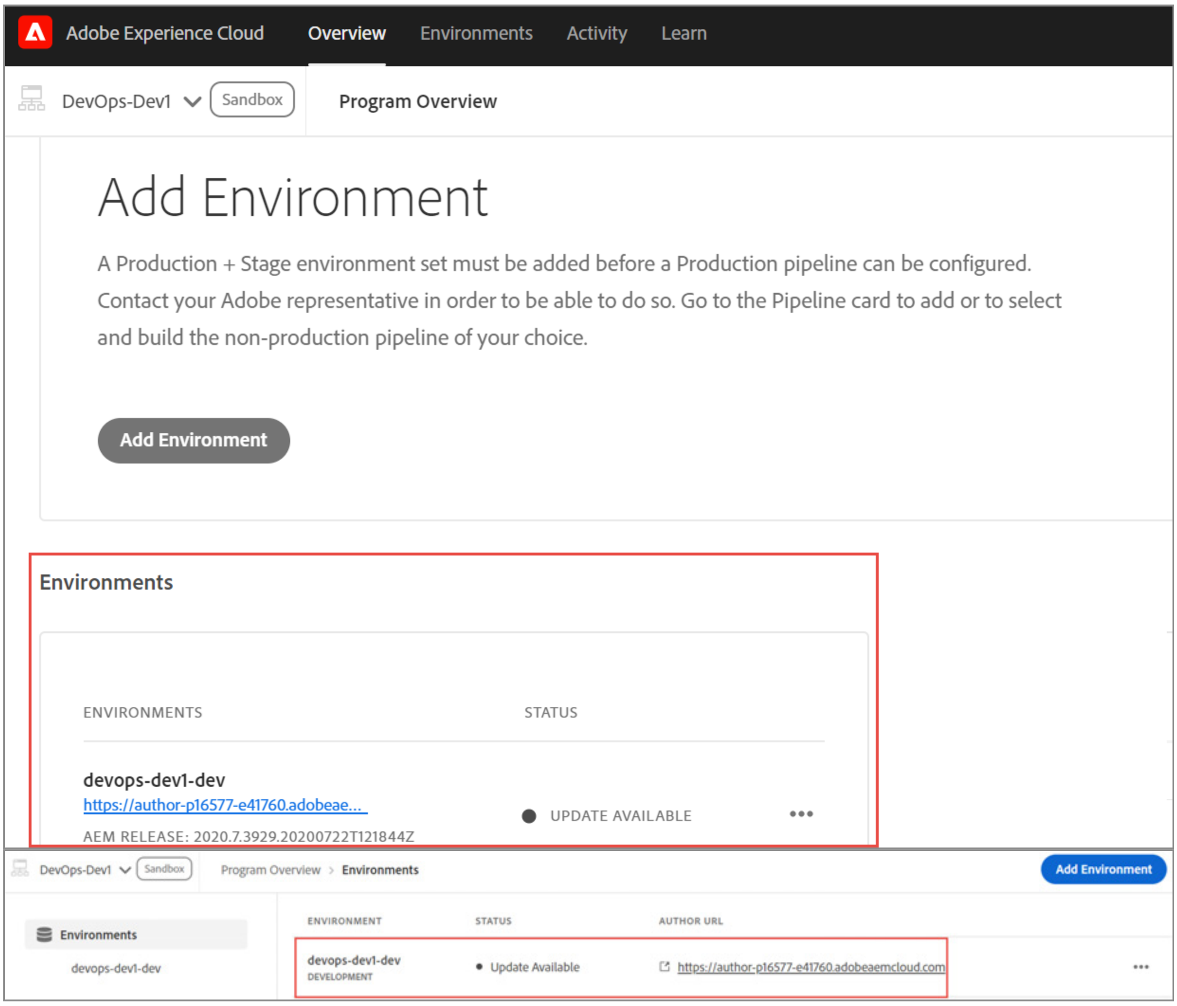
Cloud Manager has different environments to support different roles and can be engaged using different CI/CD pipelines. Cloud Manager environments typically have one Production environment and one Stage environment.

A user with the requisite permissions can create a:

- Production and Stage Environment: Is available as a duo and used for testing and production purposes
- Development Environment: Is used for development and testing purposes and is associated with non-production pipelines only

Note: Users cannot create only Stage environment or only Production environment. A Development environment is auto-created in a Sandbox program and will be configured to include Sites and Assets solutions.

After you log on to Cloud Manager by using your credentials, you can manage different environments from the **Program Overview** and **Environments** screen, as shown:



Add Environment

A Production + Stage environment set must be added before a Production pipeline can be configured. Contact your Adobe representative in order to be able to do so. Go to the Pipeline card to add or to select and build the non-production pipeline of your choice.

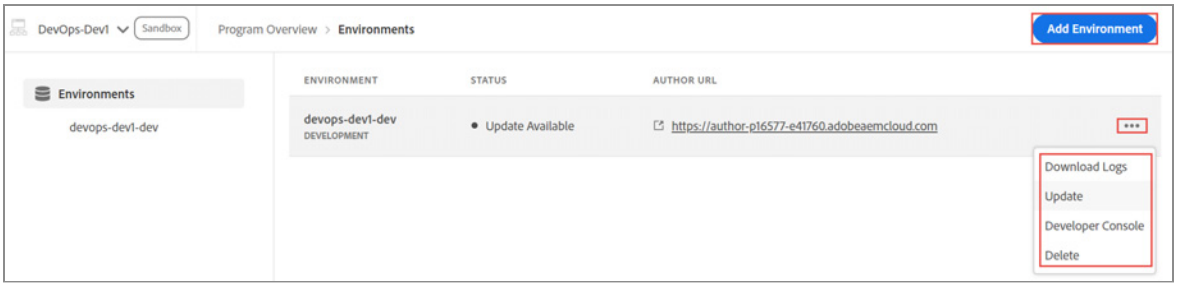
Environments

ENVIRONMENTS	STATUS
devops-dev1-dev https://author-p16577-e41760.adobeae... AEM RELEASE: 2020.7.3929.20200722T121844Z	● UPDATE AVAILABLE

ENVIRONMENT	STATUS	AUTHOR URL
devops-dev1-dev DEVELOPMENT	● Update Available	https://author-p16577-e41760.adobeaemcloud.com

The below table describes the actions that you can perform on environments from the **Program Overview** and **Environments** screen, as shown:

Action	Description
Add Environment	Adds a new environment to the program
Download Logs	Access a list of available log files for the environment
Update	Updates the pipeline associated with the environment to the latest release and execute the pipeline
Developer Console	Generates system status, resolve package dependencies and helps debugging
Delete	Deletes the selected environment



Capabilities of Environments 2

The table below lists the capabilities of environments:

Capability	Local machine	AEM as a Cloud Service					
	AEM SDK Quickstart from Software Distribution	**Dev author**	**Dev publish**	**Stage author**	**Stage publish**	**Prod author**	**Prod publish**
Local admin user	Yes	No	No	No	No	No	No
CRXDE Lite	Yes	Yes - limited	NA	No	No	No	No
Publishing Author > Publish	- Classic UI Replication - Agents Manual	- Distribution Agents - Auto-configured	No	- Distribution Agents - Auto-configured	No	- Distribution Agents - Auto-configured	No
Package Manager UI	Yes	Yes – mutable content only	NA	Yes – mutable content only	NA	Yes – mutable content only	NA
Asset Processing	DAM Update Workflow	Asset Compute	None	Asset Compute	None	Asset Compute	None
i18n dictionary translation UI	Yes (author)	No	N/A	No	N/A	No	N/A
Page Editor Developer Mode	Yes (author)	No	N/A	No	N/A	No	N/A
Create custom rollout configs	Yes (author)	No	N/A	No	N/A	No	N/A
SSL Configuration	Manual through UI	Auto-configured	Auto-configured	Auto-configured	Auto-configured	Auto-configured	Auto-configured

AEM Logging 2

AEM as a Cloud Service is a platform for customers to include custom code to create unique experiences for their customer base. Logging is a critical function to debug and understand code execution on local development, and cloud environments, particularly on AEM as a Cloud Service's Development environments.

Logging in AEM as a Cloud Service has two logical sets:

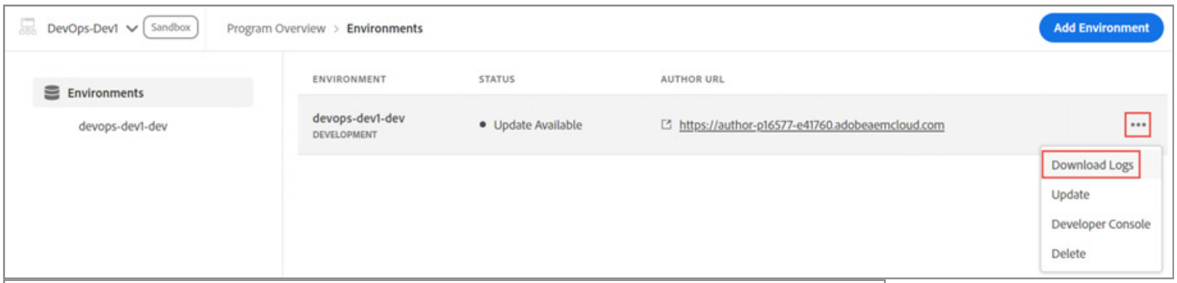
- AEM logging in, which performs logging in at the AEM application level
- Apache HTTPD Web Server or Dispatcher logging, which performs logging of the web server and Dispatcher on the Publish tier

AEM as a Cloud Service's provides access to Java log statements. AEM application developers should follow general Java logging best practices and logging pertinent statements about the execution of custom code. The following table describes the log levels:

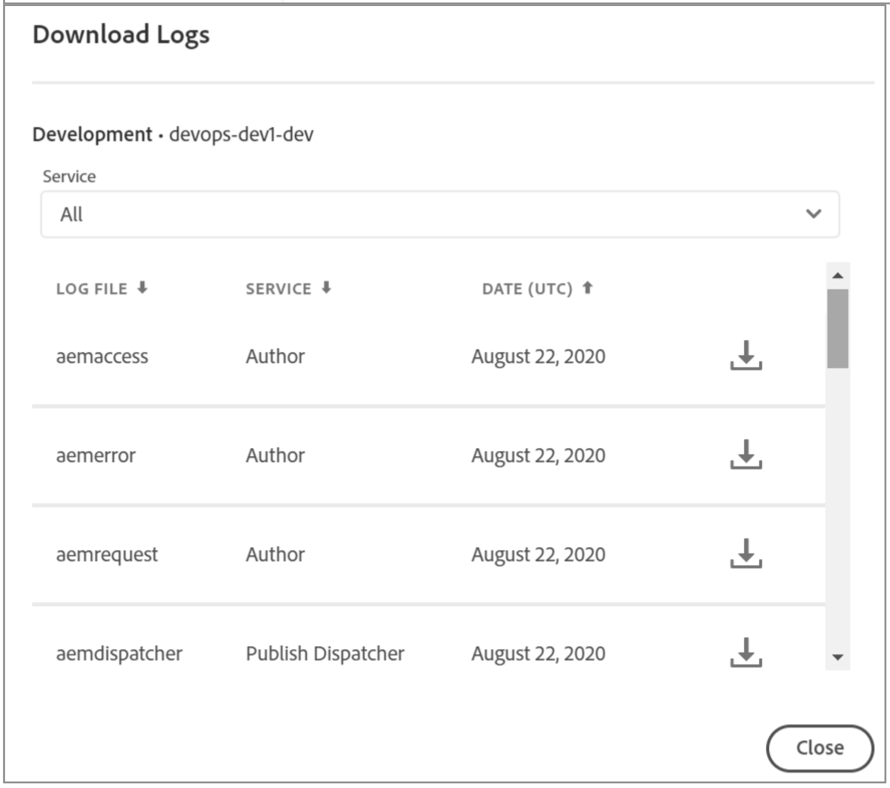
AEM environment	Log level	Description	Log statement availability
Development	DEBUG	Describes what is happening in the application. When DEBUG logging is active, the statements about activities and key parameters that affect processing are logged.	Local development Development
Production	ERROR	Describes the conditions that indicate a failure and need to resolve the error. When ERROR logging is active, only the statements indicating failures are logged.	Local development Development Stage
Stage	WARN	Describes the conditions that have the potential to become errors. When WARN logging is active, only the statements indicating the conditions that are approaching sub-optimality are logged.	Local development Development Stage Production

AEM Log levels are set per environment type through OSGi configuration, which in turn are committed to Git, and deployed through Cloud Manager to AEM as a Cloud Service. Due to this, it is best to keep the log statements consistent and well known for environment types. This ensures the logs available through AEM as Cloud Service are available at the optimal log level without requiring the redeployment of the application with the updated log level configuration.

You can access a list of available log files for the selected environment from the **Environment** screen, as shown:



ENVIRONMENT	STATUS	AUTHOR URL
devops-dev1-dev DEVELOPMENT	Update Available	https://author-p16577-e41760.adobeaemcloud.com



Download Logs

Development • devops-dev1-dev

Service: All

LOG FILE	SERVICE	DATE (UTC)
aemaccess	Author	August 22, 2020
aemerror	Author	August 22, 2020
aemrequest	Author	August 22, 2020
aemdispatcher	Publish Dispatcher	August 22, 2020

On the local Software Development Kit (SDK) installation, after you unpack and start an AEM author service, you can find logs in:

- `crx-quickstart/launchpad/logs`
- `crx-quickstart/server/logs`
- `crx-quickstart/logs`

  The following table defines the different types of logs available on AEM cloud and local SDK installation:

Source	Cloud Log	Local Log	Definition
AEM	aemerror.log	error.log	Error log for AEM
AEM	aemaccess.log	access.log	HTTP access request to the AEM service
AEM	aemrequest.log	request.log	HTTP request or response of the AEM service
Apache log	aemdispatcher.log	dispatcher.log	Dispatcher log for requests, caching, and filtering
Apache log	httpdaccess.log	access.log	HTTP requests to the Apache server
Apache log	httpderror.log	error.log	Error log for Apache

The log format is given below:

Date and time	AEM as a Cloud Service node ID	Log level	Thread	Java class	Log message
29.04.2020 21:50:13.398	[cm-p1234-e5678-aem-author-59555cb5b8-q7l9s]	*DEBUG*	qtp2130572036-1472	com.example.approval.workflow.impl.CustomApprovalWorkflow	No specified approver, defaulting to [Creative Approvers user group]

Exercise 1: Download logs

Scenario: Your organization is using AEM as a Cloud Service for their marketing website. As an administrator, you will login to Cloud Manager and observe different environments associated with the program. You will observe the server logs available on the author service and download today's aemerror.log to your machine.

Prerequisites:

- A program URL
- A program number assigned to you by the instructor
- An AEM environment that has already created for you

To download logs:

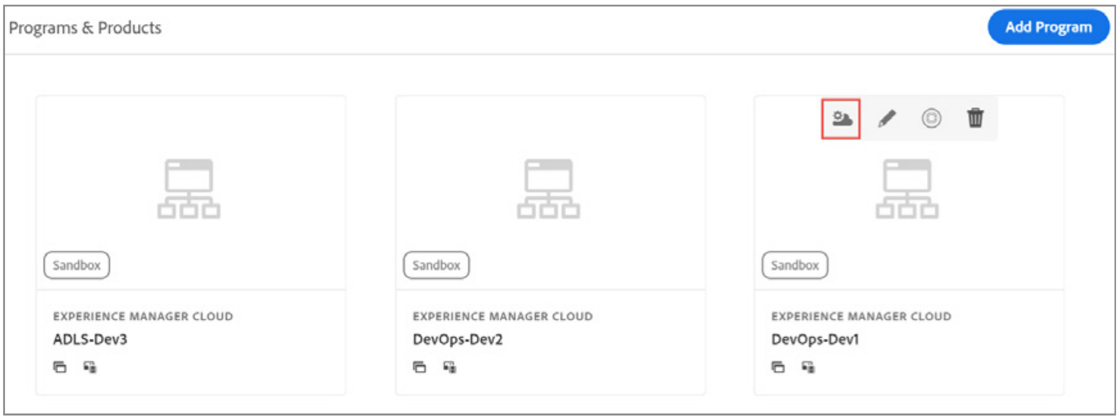
1. Open the https://experience.adobe.com/#/cloud-manager/landing.html page in a browser.

2. Click **Sign In with an Adobe ID**. The **Sign In** page opens.

3. Type the email address provided by your instructor and click **Continue**. The **Select an account** page opens.

4. Select the account specified by the instructor, type the password provided by your instructor and click **Continue**. The **Programs & Products** page with different sandboxes opens.

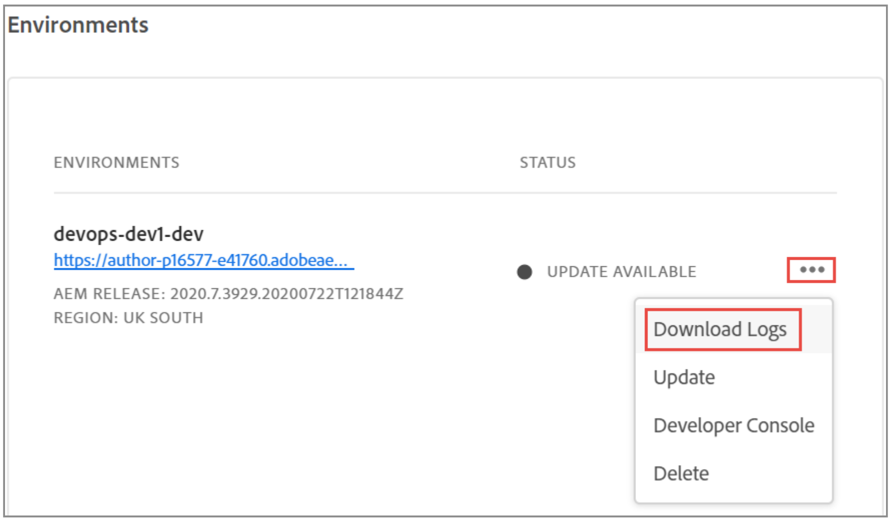
5. Hover over the **Sandbox** specified by the instructor and click the Cloud icon, as shown:



   The **Programs Overview** page opens.

6. In the **Environments** section, observe the different environments associated with your program.

7. Click the ellipsis icon and select **Download Logs** from the drop-down menu, as shown:



Environments

ENVIRONMENTS STATUS

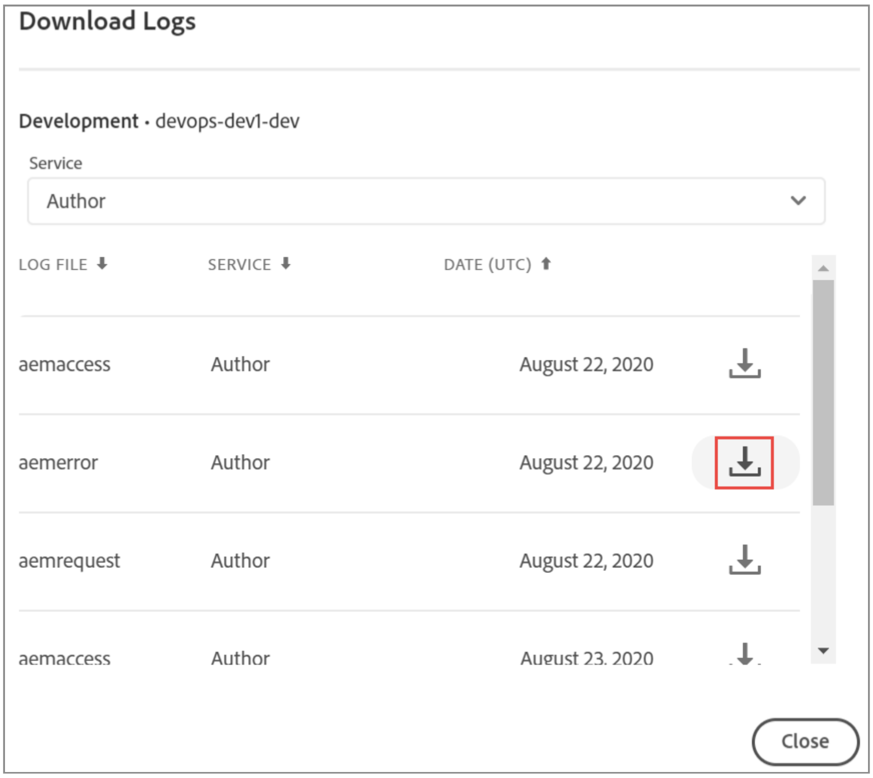
devops-dev1-dev
https://author-p16577-e41760.adobeae...
AEM RELEASE: 2020.7.3929.20200722T121844Z
REGION: UK SOUTH

● UPDATE AVAILABLE •••

The **Download Logs** dialog box opens.

8. Select **Author** from the **Service** drop-down menu. All log files related to the author service appear in the list.

9. Click the Download icon of aemerror, as shown:



Download Logs

Development • devops-dev1-dev

Service: Author

LOG FILE	SERVICE	DATE (UTC)
aemaccess	Author	August 22, 2020
aemerror	Author	August 22, 2020
aemrequest	Author	August 22, 2020
aemaccess	Author	August 23, 2020

The **author_aemerror_.log.gz** file is downloaded to your computer.

Note: The time stamp in the above aemerror log file is for reference and will vary based on the day you download the log file.

10. Click **Close** to close the **Download Logs** dialog box.

CI/CD Pipelines [3]

The CI/CD framework enables implementation teams to quickly test and deliver new or updated code. For example, implementation teams can set up, configure, and start an automated CI/CD pipeline that leverages Adobe coding best practices to perform a thorough code scan and to ensure the highest code quality.

The CI/CD pipeline automates unit and performance testing processes to increase deployment efficiency and proactively identify critical issues that are expensive to fix after deployment. Implementation teams can access a comprehensive code performance report to gain visibility into potential impact on Key performance indicators (KPIs) and critical security validations if the code is deployed to production.

The Cloud Manager has two types of pipeline:

- Production pipeline
- Non-Production pipeline

Production Pipeline

The Production Pipeline is used to build and deploy code through Stage environment to the Production environment and helps decrease time to value.

Non-Production Pipeline

The Non-Production pipelines execute the build and code quality steps. It consists of two categories, the Code Quality pipeline and the Deployment pipeline. In Code Quality pipeline, the code from a Git branch is used to build and is evaluated against Cloud Manager's code quality scan. The Deployment pipeline supports the automated deployment of code from the Git repository to any non-production environment.

Configuring the CI/CD Pipeline [4]

To configure the pipeline, you must:

- Define the trigger that will start the pipeline
- Define the parameters controlling the deployment
- Configure the performance test parameters

You can set up the Production pipeline from the Deployment Manager. You cannot set up a Production pipeline until a program creation is complete, Git repository has at least one branch, and a Production and Stage environment set is created.

Before you start to deploy your code, you must configure your pipeline settings from Cloud Manager.

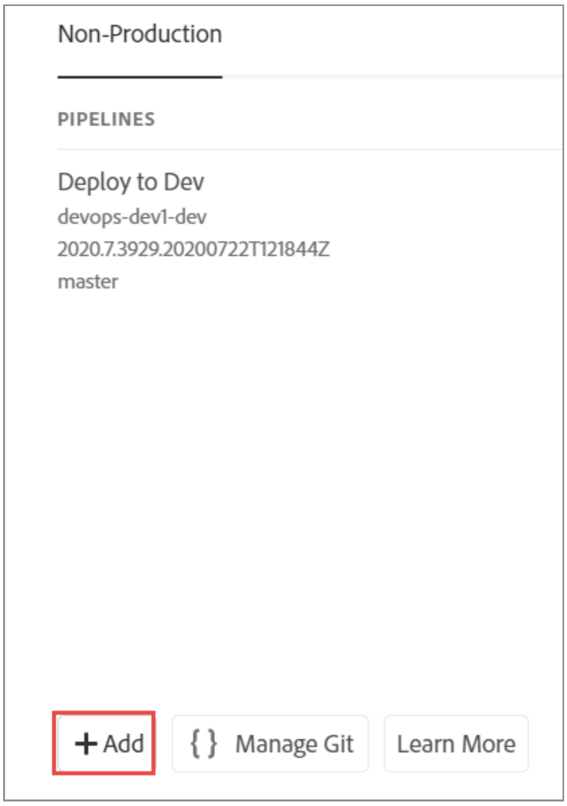
Exercise 2: Create an auto-executing Code Quality pipeline

Scenario: Depending on the build process, Cloud Manager offers a wide variety of pipeline configurations to start the pipeline manually or automatically. In this exercise, you will create a Code Quality pipeline that will auto execute whenever the changes are pushed to Cloud Manager from the Git branch.

To create an auto-executing Code Quality pipeline:

1. Ensure you are in the program assigned by your instructor.

2. On the **Program Overview** page, under the **Pipelines** section, click **Add** at the bottom, as shown:



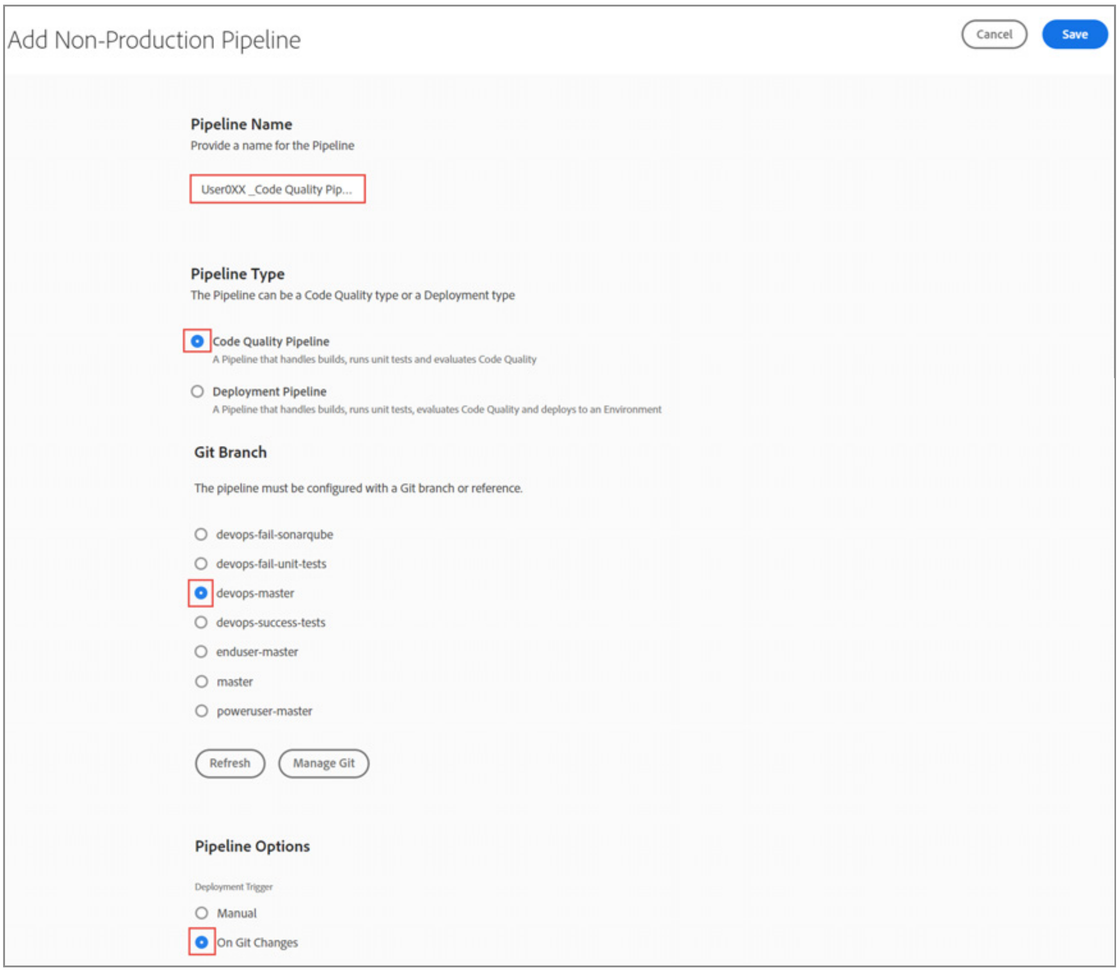
Non-Production

PIPELINES

Deploy to Dev
devops-dev1-dev
2020.7.3929.20200722T121844Z
master

The **Add Non-Production Pipeline** wizard opens.

3. Perform the following actions in the wizard, as shown:
   a. Type **User0XX_Pipeline** in the **Pipeline Name** box.
   b. Select **Code Quality Pipeline** from **Pipeline Type**.
   c. Select **devops-master** from **Git Branch**.
   d. Select **On Git Changes** from **Pipeline Options**.



Pipeline Name
Provide a name for the Pipeline
User0XX _Code Quality Pip...

Pipeline Type
The Pipeline can be a Code Quality type or a Deployment type

● Code Quality Pipeline
A Pipeline that handles builds, runs unit tests and evaluates Code Quality

○ Deployment Pipeline
A Pipeline that handles builds, runs unit tests, evaluates Code Quality and deploys to an Environment

Git Branch
The pipeline must be configured with a Git branch or reference.

○ devops-fail-sonarqube
○ devops-fail-unit-tests
● devops-master
○ devops-success-tests
○ enduser-master
○ master
○ poweruser-master

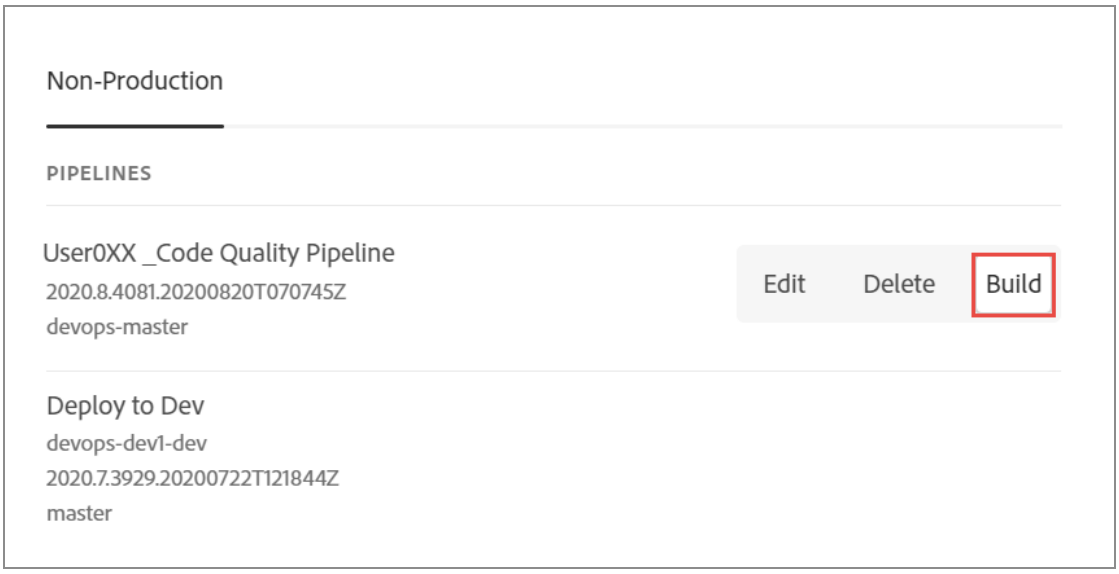
Pipeline Options
Deployment Trigger
○ Manual
● On Git Changes

4. Click **Save**. The **Pipeline was set up successfully** message appears at the top of the **Program Overview** page.

The Code Quality Pipeline will now auto execute every time the code is added to the devops-master branch. At any given time, you can manually start the pipeline too.

5. Notice that the **User0XX_Code Quality Pipeline** is added to the **Non-Production** section.

6. To manually start the pipeline, hover over **User0XX_Pipeline** and click **Build**, as shown:



Non-Production

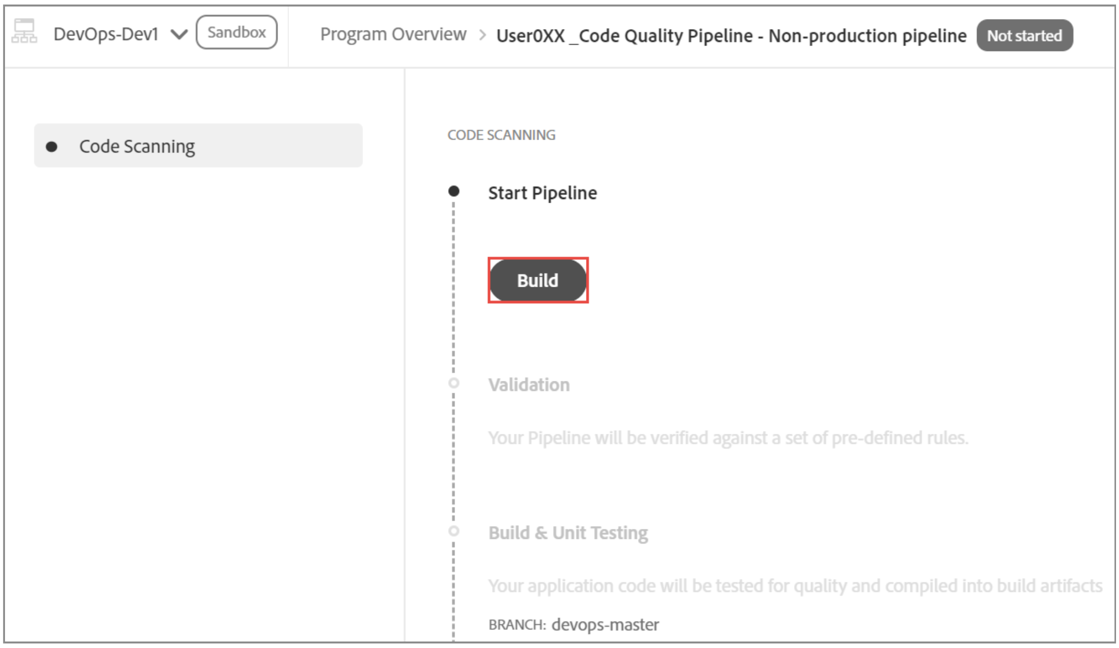
PIPELINES

User0XX _Code Quality Pipeline
2020.8.4081.20200820T070745Z
devops-master

Deploy to Dev
devops-dev1-dev
2020.7.3929.20200722T121844Z
master

The **Code Scanning** section opens.

7. Click **Build**, as shown, to start the pipeline:



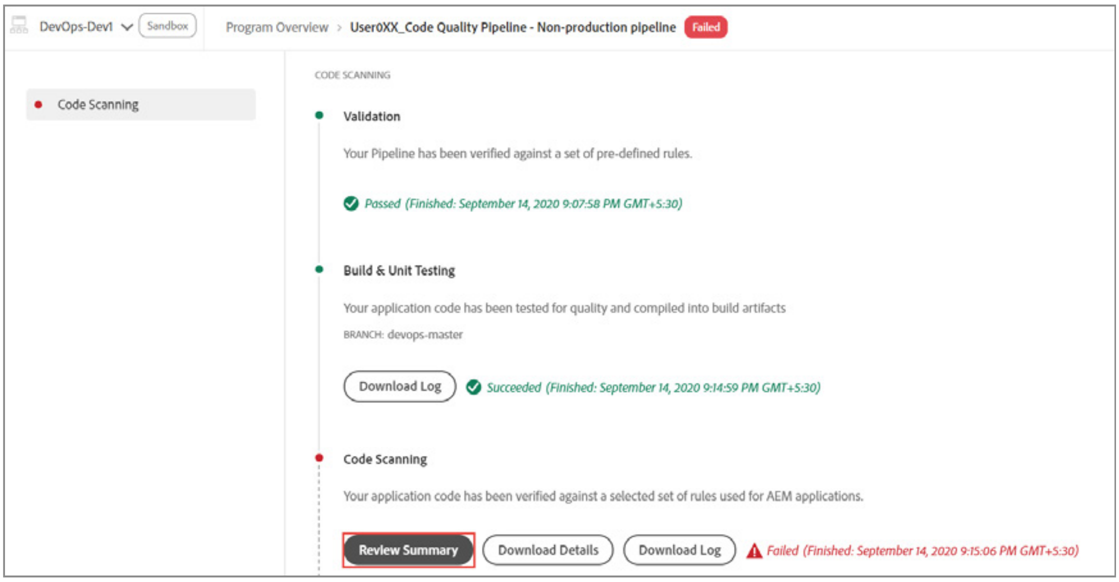
Your Pipeline will be verified against a set of pre-defined rules.

Your application code will be tested for quality and compiled into build artifacts

BRANCH: devops-master

8. After few minutes, notice that the pipeline fails on the **Code Scanning** step.

9. Click **Review Summary**, as shown:



Validation

Your Pipeline has been verified against a set of pre-defined rules.

Passed (Finished: September 14, 2020 9:07:58 PM GMT+5:30)

Build & Unit Testing

Your application code has been tested for quality and compiled into build artifacts

BRANCH: devops-master

Succeeded (Finished: September 14, 2020 9:14:59 PM GMT+5:30)

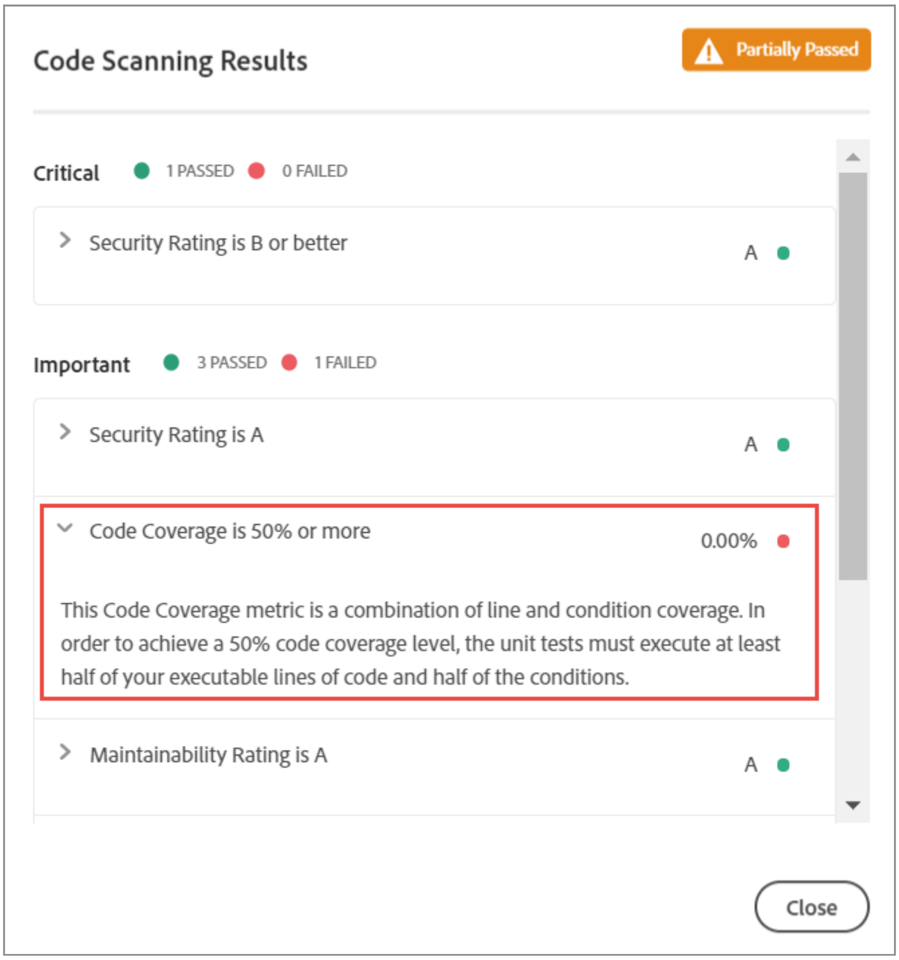
Code Scanning

Your application code has been verified against a selected set of rules used for AEM applications.

Failed (Finished: September 14, 2020 9:15:06 PM GMT+5:30)

The **Code Scanning Results** opens.

10.. Notice that Code Coverage is 0.00%. This is considered as an important issue and, therefore, the pipeline failed, as shown:



Code Scanning Results

Partially Passed

Critical ● 1 PASSED ● 0 FAILED

Security Rating is B or better A ●

Important ● 3 PASSED ● 1 FAILED

Security Rating is A A ●

Code Coverage is 50% or more 0.00% ●

This Code Coverage metric is a combination of line and condition coverage. In order to achieve a 50% code coverage level, the unit tests must execute at least half of your executable lines of code and half of the conditions.

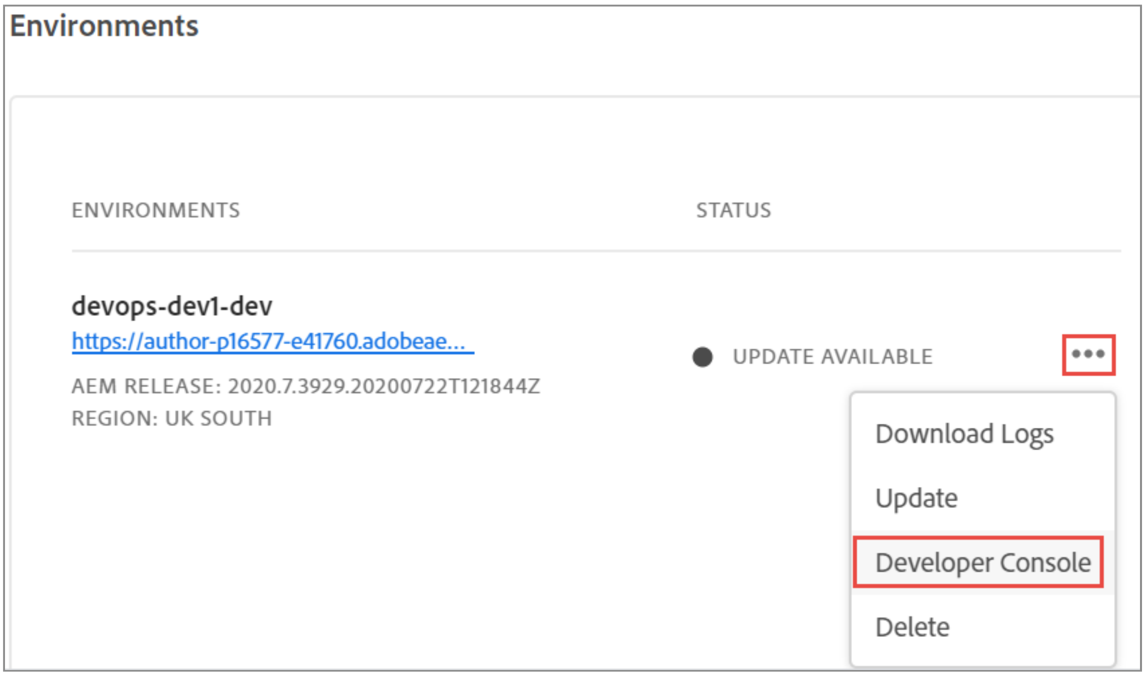
Maintainability Rating is A A ●

Note: Code quality pipelines automatically fail on important and critical issues. Development pipelines (attached to an environment) have the option to continue with important issues if needed. Development pipelines will still fail with critical issues.

11. Click **Close** to close the dialog box.

Developer Console [3]

You can access the **Developer Console** from the drop-down menu on the **Environments** page, as shown:



Environments

ENVIRONMENTS	STATUS
devops-dev1-dev https://author-p16577-e41760.adobeae... AEM RELEASE: 2020.7.3929.20200722T121844Z REGION: UK SOUTH	● UPDATE AVAILABLE

In the **Developer Console**, you can:

- Access to text or Java Script Object Notification (JSON) dumps of data from the Web Console
- Hibernate and de-hibernate the Sandbox Program Environments.
- Generate statuses information that include the state of bundles, components, OSGI configurations, oak indexes, OSGi services, and Sling jobs.
- Resolve package dependencies and servlets.
- Debug and monitor query performance

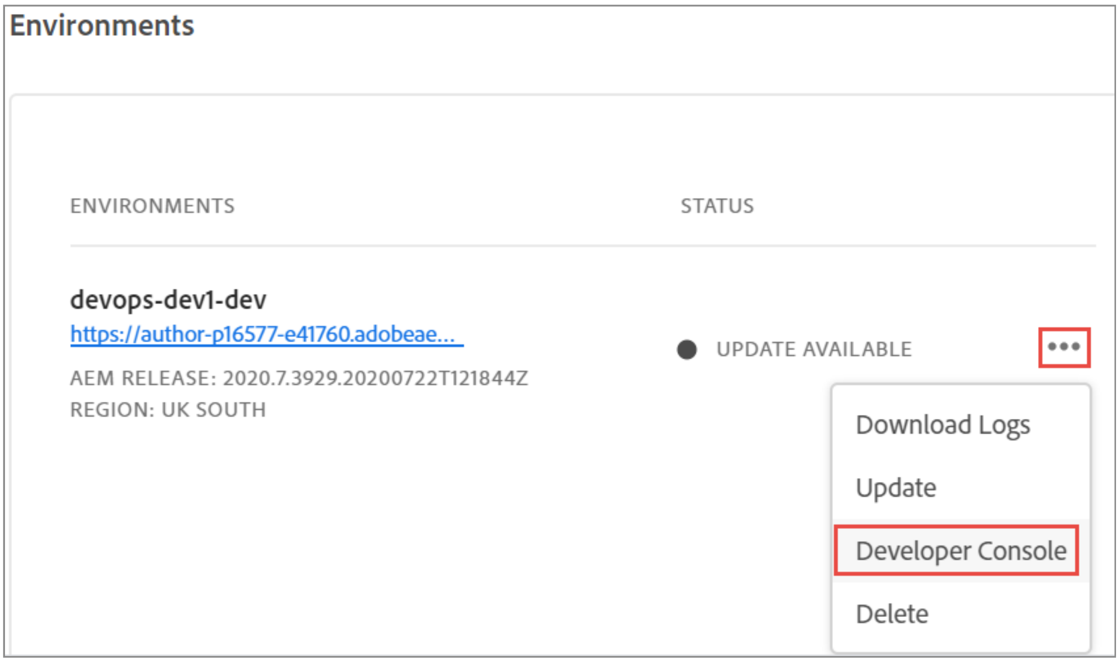
For regular programs, access to the Developer Console is defined by the **Cloud Manager - Developer Role** in the Admin Console, while for sandbox programs, the **Developer Console** is available to any user with a product profile giving them access to AEM as a Cloud Service.

For all programs, **Cloud Manager - Developer Role** is needed for status dumps and users must also be defined in the AEM Users or AEM Administrators Product Profile on both author and publish services to view the status dump data from both services.

Exercise 3: Access and explore the Developer Console

To access the Developer Console:

1. Ensure you are in the program assigned by your instructor.
2. On the **Program Overview** page, under the **Environments** section, click the ellipsis icon and select **Developer Console** from the drop-down menu, as shown:



Environments

ENVIRONMENTS | STATUS

devops-dev1-dev
https://author-p16577-e41760.adobeae...
AEM RELEASE: 2020.7.3929.20200722T121844Z
REGION: UK SOUTH

● UPDATE AVAILABLE

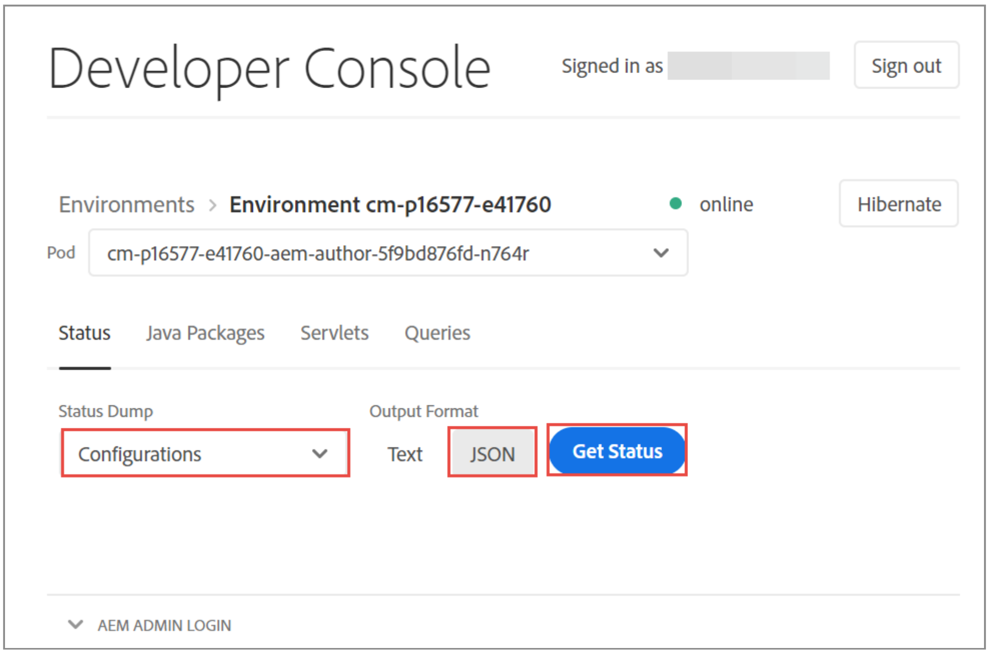
The **Developer Console** page opens.

3. Notice that the **Pod** option helps specify the AEM pod to get the data output for the selected service.

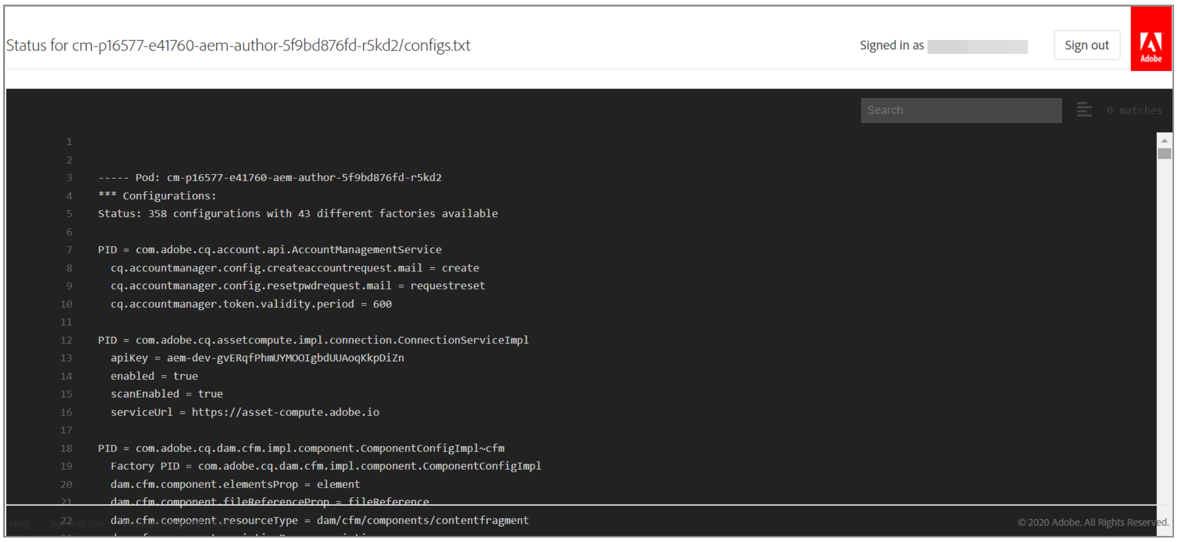
Note: Depending on the traffic to your AEM service, you can view more or less pods available for each author or publish service.

4. Select an author service from the **Pod** drop-down menu.

5. On the **Status** tab, perform the following actions, as shown:
   a. Select **Configurations** from the **Status Dump** drop-down menu.
   b. Select **JSON** from **Output Format**.
   c. Click **Get Status**.



The Status for selected author service for example, **Status for cm-p16577-e41760-aem-author-5f9bd876fd-r5kd2/configs.json**, as shown, opens on a new browser tab:

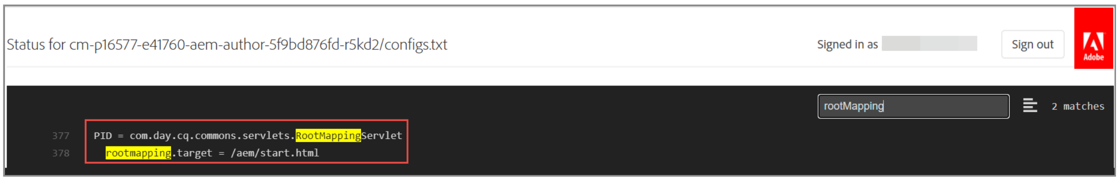

Status for cm-p16577-e41760-aem-author-5f9bd876fd-r5kd2/configs.txt
Signed in as [redacted]
Sign out
Adobe
Search
0 matches
Help
Terms of Use
Privacy Policy and Cookies
© 2020 Adobe. All Rights Reserved.

```


----- Pod: cm-p16577-e41760-aem-author-5f9bd876fd-r5kd2
*** Configurations:
Status: 358 configurations with 43 different factories available

PID = com.adobe.cq.account.api.AccountManagementService
  cq.accountmanager.config.createaccountrequest.mail = create
  cq.accountmanager.config.resetpwdrequest.mail = requestreset
  cq.accountmanager.token.validity.period = 600

PID = com.adobe.cq.assetcompute.impl.connection.ConnectionServiceImpl
  apiKey = aem-dev-gvERqfPhmUYMOOIgbdUUAoqKkpDiZn
  enabled = true
  scanEnabled = true
  serviceUrl = https://asset-compute.adobe.io

PID = com.adobe.cq.dam.cfm.impl.component.ComponentConfigImpl~cfm
  Factory PID = com.adobe.cq.dam.cfm.impl.component.ComponentConfigImpl
  dam.cfm.component.elementsProp = element
  dam.cfm.component.fileReferenceProp = fileReference
  dam.cfm.component.resourceType = dam/cfm/components/contentfragment
```

The json output contains the set of all OSGi configurations for the AEM Author service. Because you have already run through the CI/CD pipeline that contains the completed code for this devOps training, the OSGi configurations output should contain the rootMapping config you created in a previous module.

6. In the json output, type **rootMapping** in the **Search** box and press the Enter key.

7. Observe the custom value from our maven project, as shown:



Status for cm-p16577-e41760-aem-author-5f9bd876fd-r5kd2/configs.txt

```
PID = com.day.cq.commons.servlets.RootMappingServlet
  rootmapping.target = /aem/start.html
```

Deploy Content Packages Through Cloud Manager 5

The Sling OSGi Feature Model or the Sling Feature Model replaces the Sling Provisioning Model as the primary way of defining and assembling Sling applications. This feature enables you to add new annotations to Sling Models. These annotations define how the Model can be exported as a different Java object or serialized into a different format such as JSON. Apache Sling provides a Jackson JSON exporter to cover the most common case of exporting Sling Models as JSON objects for consumption by programmatic web consumers such as other web services and JavaScript applications.

You can work with Sling Models when developing with AEM. When developing an AEM project, you can define a model object (a Java object) and map that object to Sling resources.

The packages in Package Manager on Cloud environments include *Content Package to Feature Model* (*cp2fm*). The transformed packages will not have metadata. You cannot interact with, which means you cannot download, replicate, or open the package.

Content packages written for AEM as a Cloud Service application must have a clean separation between immutable and mutable content. Cloud Manager will enforce that clean separation by failing the pipeline build if mutable and immutable content is mixed in a single content package.

Apache Sling Content Package to Feature Model Converter

The cp2fm tool is used to:

- Extract OSGi bundles and OSGi configurations
- Scan nested content-packages from an input content-package
- Create one (or more) Apache Sling Feature model files
- Deploy the extracted OSGi bundles in a directory.

Immutable Content Packages

All content and code persisted in the immutable repository must be checked into the Github repository and deployed through Cloud Manager. Code is **never** deployed directly to a running AEM instance. This ensures that the code running for a given release in any Cloud environment is identical. This eliminates the risk of unintentional code variation on production.

Exercise 4: Observe and attempt to upload immutable content packages

Scenario: An administrator wants to verify whether they can upload immutable content by using the Package Manager. This exercise includes the following tasks:

1. Observe immutable content packages

2. Attempt to upload an immutable content package

Prerequisites:

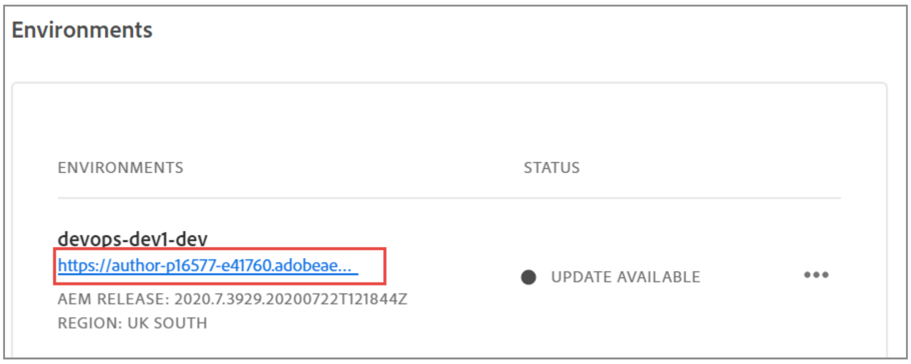
- AEM Cloud Service development environment with a defined associated pipeline
- A valid and running author service
- WKND application deployed to the author service

Task 1: Observe immutable content packages

In this task, you will observe the content packages that are uploaded as part of the WKND application.

1. Ensure you are in the program assigned by your instructor.

2. On the **Program Overview** page, under the **Environments** section, click the author service, as shown:



Environments

ENVIRONMENTS	STATUS
devops-dev1-dev https://author-p16577-e41760.adobeae... AEM RELEASE: 2020.7.3929.20200722T121844Z REGION: UK SOUTH	● UPDATE AVAILABLE

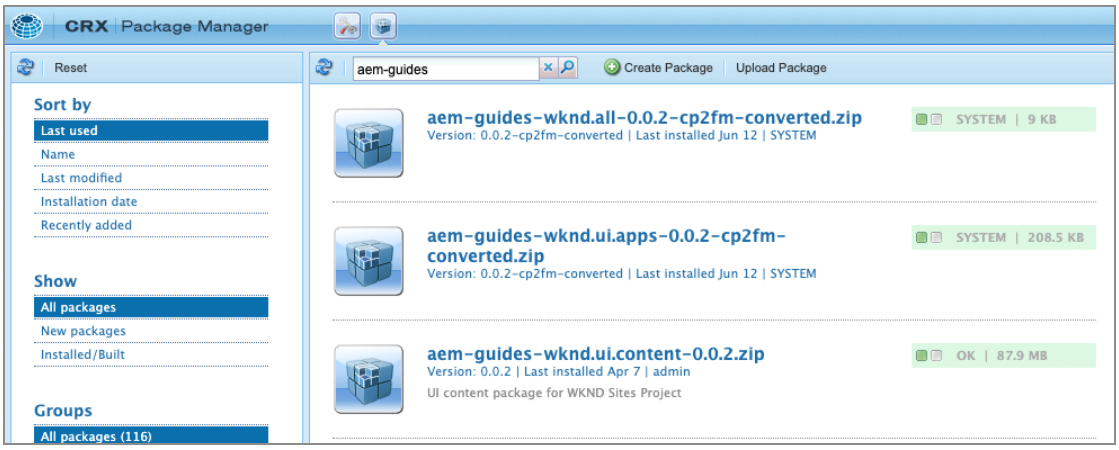
The **AEM Sign In** page opens.

3. Click **Sign in with Adobe**.

> **Note:** You will be signed into the AEM service with the Student ID you were assigned. Because you created the environment, you are automatically assigned the AEM-Administrator rights.

4. Navigate to **Tools** > **Deployment** > **Packages**. The **Package Manager** page opens.

5. Type **aem-guides** in the **Search packages** box at the top and press the Enter key. This will find the packages associated with the WKND application, as shown:

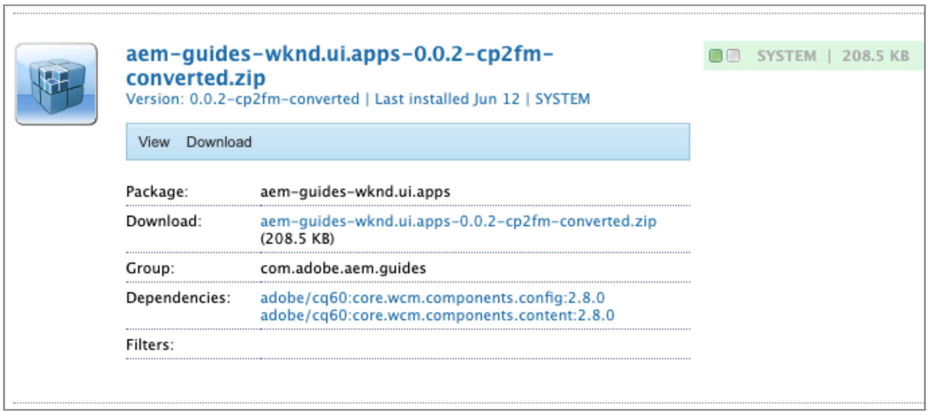


aem-guides-wknd.all-0.0.2-cp2fm-converted.zip
Version: 0.0.2-cp2fm-converted | Last installed Jun 12 | SYSTEM
SYSTEM | 9 KB

aem-guides-wknd.ui.apps-0.0.2-cp2fm-converted.zip
Version: 0.0.2-cp2fm-converted | Last installed Jun 12 | SYSTEM
SYSTEM | 208.5 KB

aem-guides-wknd.ui.content-0.0.2.zip
Version: 0.0.2 | Last installed Apr 7 | admin
UI content package for WKND Sites Project
OK | 87.9 MB

6. Click the **aem-guides-wknd.ui.apps--cp2fm-converted.zip** package, as shown:



aem-guides-wknd.ui.apps-0.0.2-cp2fm-converted.zip
Version: 0.0.2-cp2fm-converted | Last installed Jun 12 | SYSTEM
SYSTEM | 208.5 KB

Package:	aem-guides-wknd.ui.apps
Download:	aem-guides-wknd.ui.apps-0.0.2-cp2fm-converted.zip (208.5 KB)
Group:	com.adobe.aem.guides
Dependencies:	adobe/cq60:core.wcm.components.config:2.8.0 adobe/cq60:core.wcm.components.content:2.8.0
Filters:	

The content package containing immutable content is displayed.

7. Notice the name of the ui.apps package: aem-guides-wknd.ui.apps--cp2fm-converted. When the cp2fm converter converts the immutable package, it appends **cp2-fm-converted** to the name.

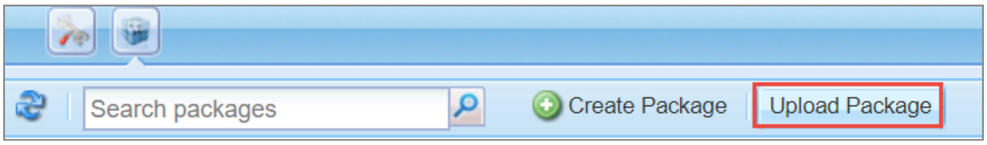
8. Notice that the package does not have any **Filters**.

Task 2: Attempt to upload an immutable content package

In this task, you will verify whether the upload of any immutable content is blocked.

1. Ensure you are in the **Package Manager** page.

2. Click **Upload Package** on the actions bar, as shown:



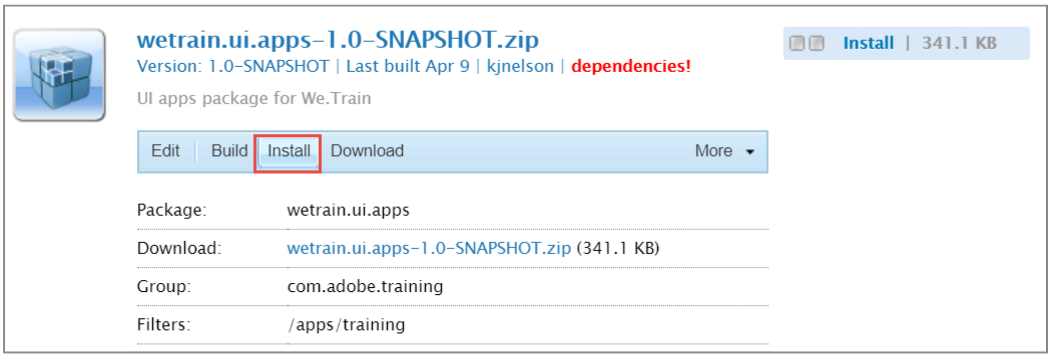
The **Upload Package** dialog box opens.

3. Click **Browse** in the dialog box. The **Open** dialog box opens.

4. On your file system, navigate to the **Exercise_Files/Environments** folder, select **we-train.ui.apps-1.0-SNAPSHOT.zip** and click **Open**. The selected package is added to the **Upload Package** dialog box.

5. Click **OK**. The package is uploaded.

6. Click **Install** on the toolbar, as shown:



wetrain.ui.apps-1.0-SNAPSHOT.zip

Version: 1.0-SNAPSHOT | Last built Apr 9 | kjnelson | dependencies!

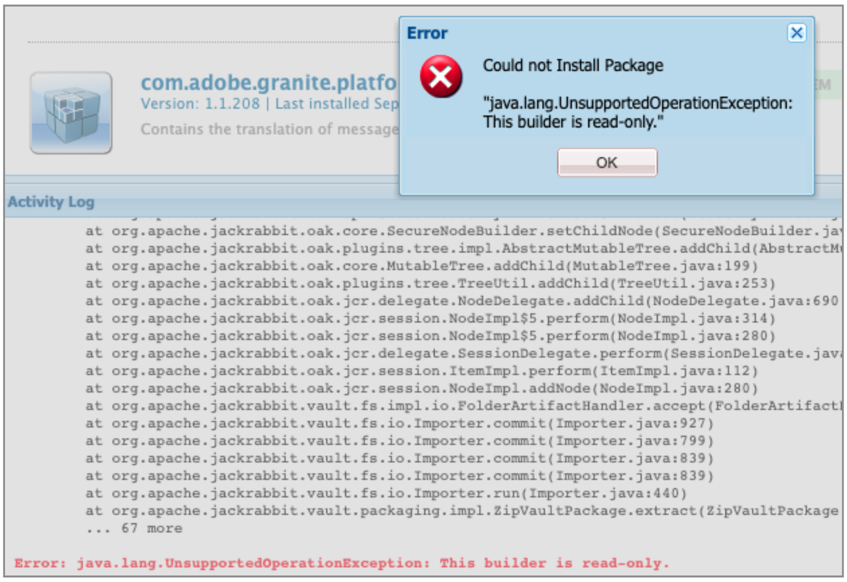
UI apps package for We.Train

Package:	wetrain.ui.apps
Download:	wetrain.ui.apps-1.0-SNAPSHOT.zip (341.1 KB)
Group:	com.adobe.training
Filters:	/apps/training

The **Install Package** dialog box opens.

7. Click **Install**.

8. Notice that an **Error** box with **Could not Install Package** message opens, as shown:



com.adobe.granite.platfo

Version: 1.1.208 | Last installed Sep

Contains the translation of message

Error

Could not Install Package

"java.lang.UnsupportedOperationException: This builder is read-only."

```
        at org.apache.jackrabbit.oak.core.SecureNodeBuilder.setChildNode(SecureNodeBuilder.ja
        at org.apache.jackrabbit.oak.plugins.tree.impl.AbstractMutableTree.addChild(AbstractM
        at org.apache.jackrabbit.oak.core.MutableTree.addChild(MutableTree.java:199)
        at org.apache.jackrabbit.oak.plugins.tree.TreeUtil.addChild(TreeUtil.java:253)
        at org.apache.jackrabbit.oak.jcr.delegate.NodeDelegate.addChild(NodeDelegate.java:690
        at org.apache.jackrabbit.oak.jcr.session.NodeImpl$5.perform(NodeImpl.java:314)
        at org.apache.jackrabbit.oak.jcr.session.NodeImpl$5.perform(NodeImpl.java:280)
        at org.apache.jackrabbit.oak.jcr.delegate.SessionDelegate.perform(SessionDelegate.jav
        at org.apache.jackrabbit.oak.jcr.session.ItemImpl.perform(ItemImpl.java:112)
        at org.apache.jackrabbit.oak.jcr.session.NodeImpl.addNode(NodeImpl.java:280)
        at org.apache.jackrabbit.vault.fs.impl.io.FolderArtifactHandler.accept(FolderArtifact
        at org.apache.jackrabbit.vault.fs.io.Importer.commit(Importer.java:927)
        at org.apache.jackrabbit.vault.fs.io.Importer.commit(Importer.java:799)
        at org.apache.jackrabbit.vault.fs.io.Importer.commit(Importer.java:839)
        at org.apache.jackrabbit.vault.fs.io.Importer.commit(Importer.java:839)
        at org.apache.jackrabbit.vault.fs.io.Importer.run(Importer.java:440)
        at org.apache.jackrabbit.vault.packaging.impl.ZipVaultPackage.extract(ZipVaultPackage
        ... 67 more

Error: java.lang.UnsupportedOperationException: This builder is read-only.
```

The installation fails because the package includes immutable content.

You have verified that the immutable content cannot be deployed on AEM as a Cloud Service using the Package Manager. Immutable content can only be deployed using the Cloud Manager pipelines.

Exercise 5: Create, upload, and install mutable content packages

Scenario: An author, participating in User Acceptance Testing (UAT) wants to validate and test pages, created from their templates and components, in the AEM as a Cloud Service development environment. At the end of the test, the author would like to return the environment to the original state so that the next test can begin.

This exercise includes the following tasks:

1. Create new content

2. Create a content package containing test content and download it

3. Upload a mutable content package of test content

4. Reinstall the WKND mutable content package

Prerequisites:

- AEM Cloud Service development environment with defined associated pipeline
- A valid and running author service
- WKND application deployed to author service

Task 1: Create new content

In this task, you will test the templates and components by creating a new page and editing that page.

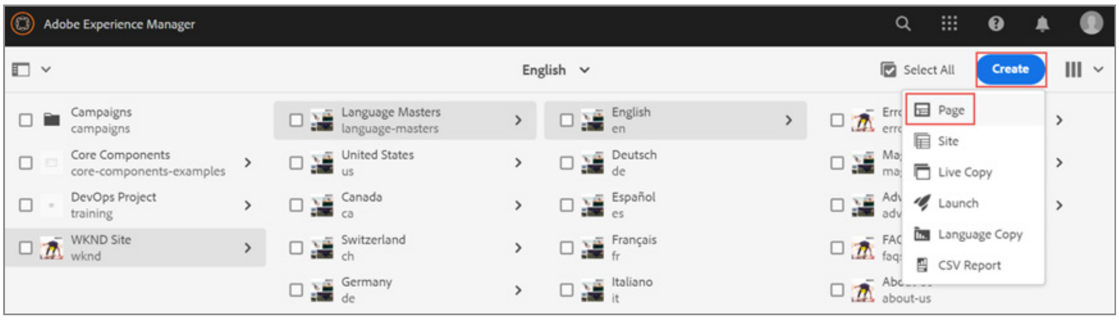
1. Ensure you have logged on to the author service.

2. Click **Adobe Experience Manager** on the header bar. The **Navigation** pane opens.

3. Click **Sites** to open the console.

4. Navigate to **WKND Site** > **Language Masters** > **English**.

5. Click **Create** on the actions bar and select **Page** from the drop-down menu, as shown:



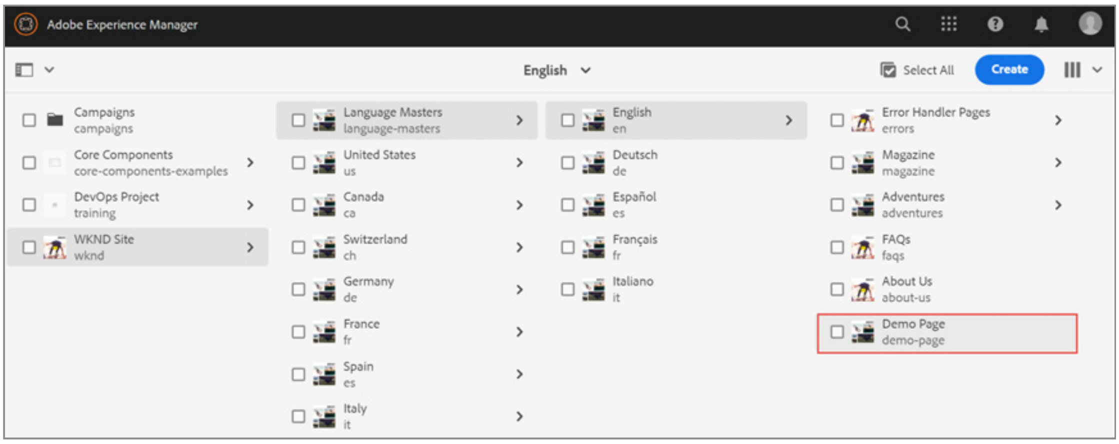
The **Create Page** wizard is displayed.

6. In the **Template** section, select the **Landing Page Template** and click **Next**. The **Create Page** Properties screen opens.

7. Type **Demo Page** in the **Title** box and click **Create**. The **Success** dialog box opens.

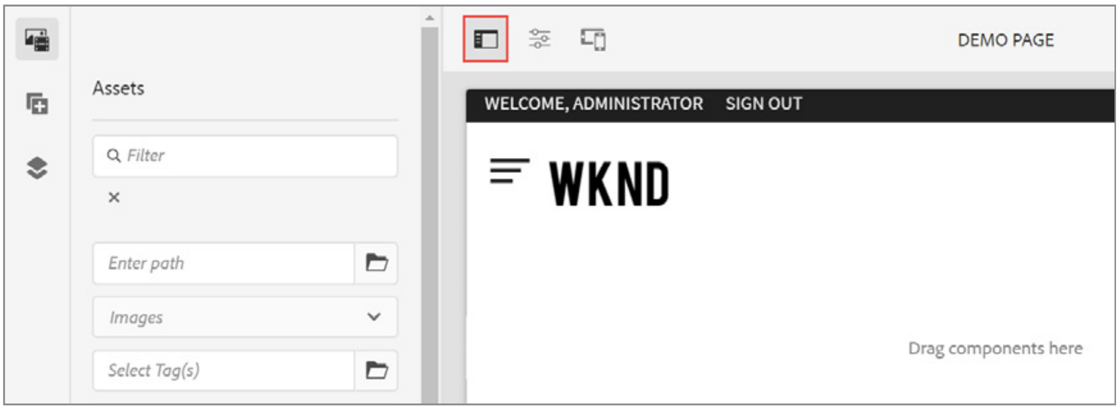
8. Click **Done**. The **Sites** console opens.

9. Verify that the **Demo Page** has been created in the correct place, as shown:

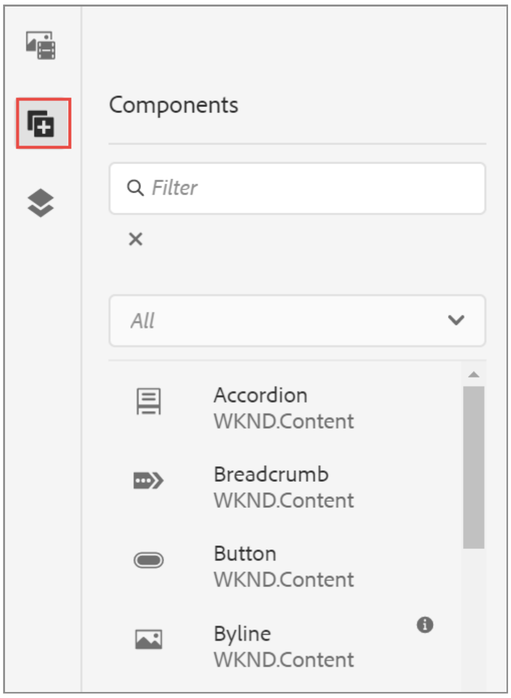


10. Select the **Demo Page** and click **Edit** on the actions bar. The page opens on a new browser tab in **Edit** mode.

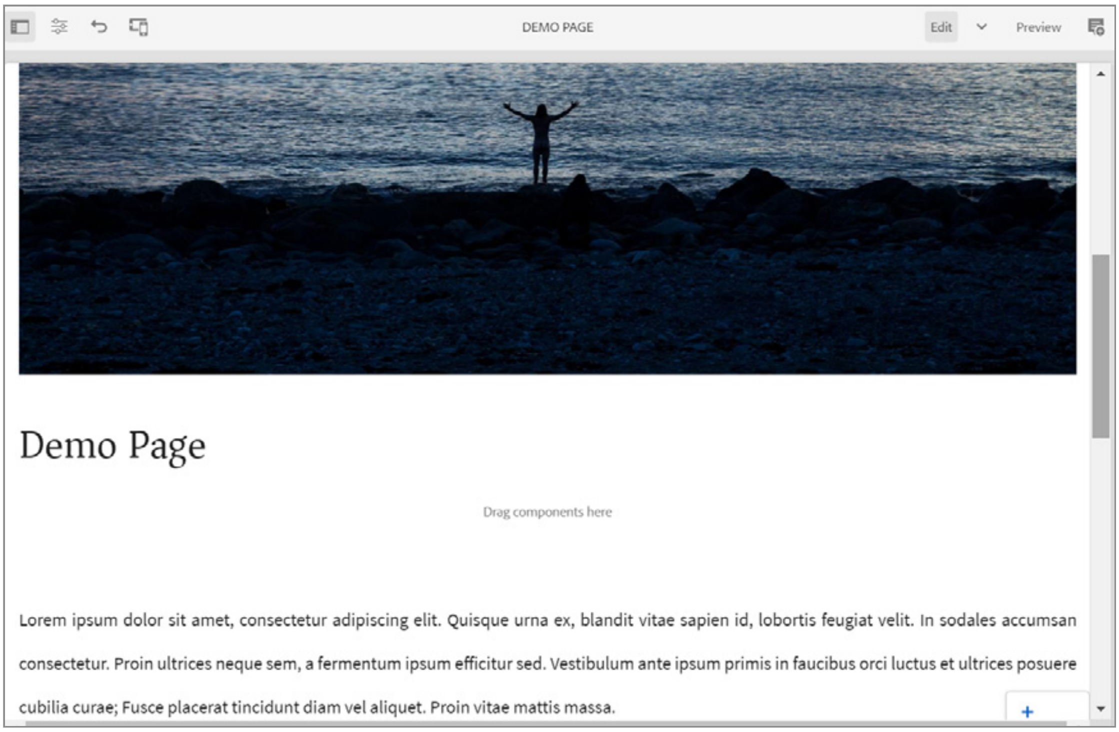
11. Click the Toggle Side Panel icon on the page toolbar to open the panel, as shown:



12. Click the Components icon to open the panel, as shown:



13. Drag few components to add content to the page. Your page should look similar to the one shown in the below screenshot:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque urna ex, blandit vitae sapien id, lobortis feugiat velit. In sodales accumsan consectetur. Proin ultrices neque sem, a fermentum ipsum efficitur sed. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Fusce placerat tincidunt diam vel aliquet. Proin vitae mattis massa.

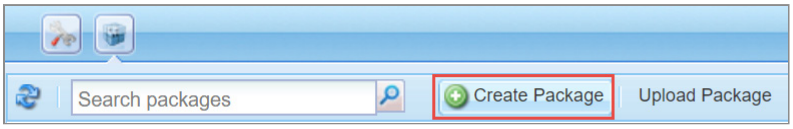
Task 2: Create a content package containing test content and download it

In this task, you will create a content package with the test content and download that package for documentation purposes.

1. Click **Adobe Experience Manager** on the header bar. The **Navigation** pane opens.

2. Navigate to **Tools** > **Deployment** > **Packages**. The **Package Manager** page opens.

3. Click the **Create Package** button on the actions bar, as shown:

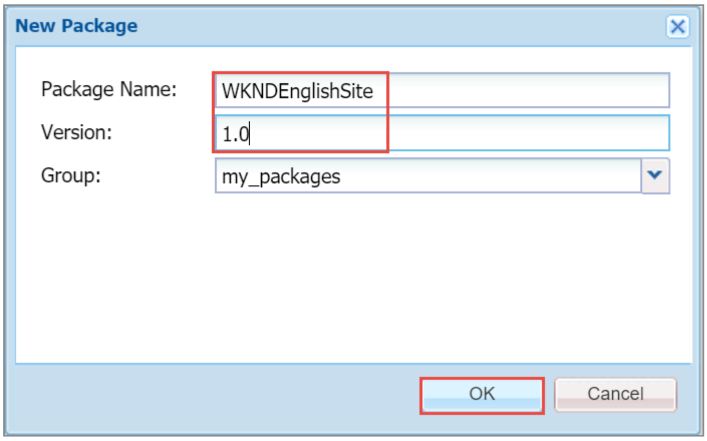


The **New Package** dialog box opens.

4. Type the following information, as shown, and click **OK**.

a. **Package Name**: **WKNDEnglishSite**

b. **Version:** 1.0



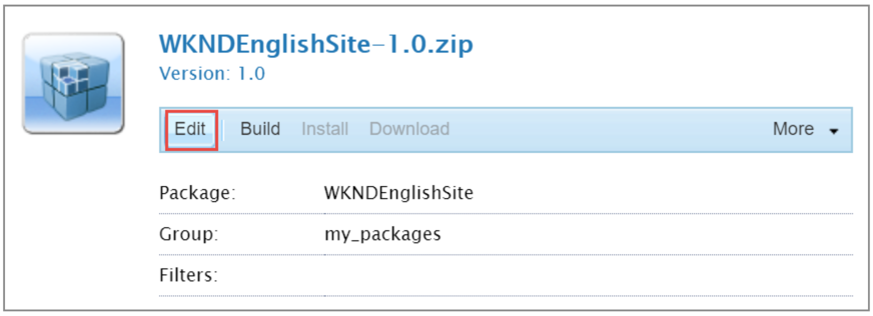
New Package

Package Name: WKNDEnglishSite

Version: 1.0

Group: my_packages

5. Click the **Edit** button on the actions bar, as shown:



WKNDEnglishSite-1.0.zip

Version: 1.0

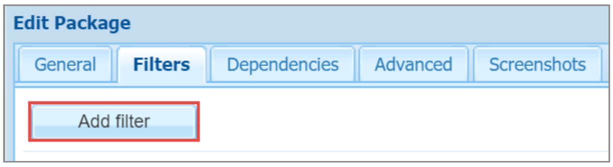
Package: WKNDEnglishSite

Group: my_packages

Filters:

The **Edit Package** dialog box opens.

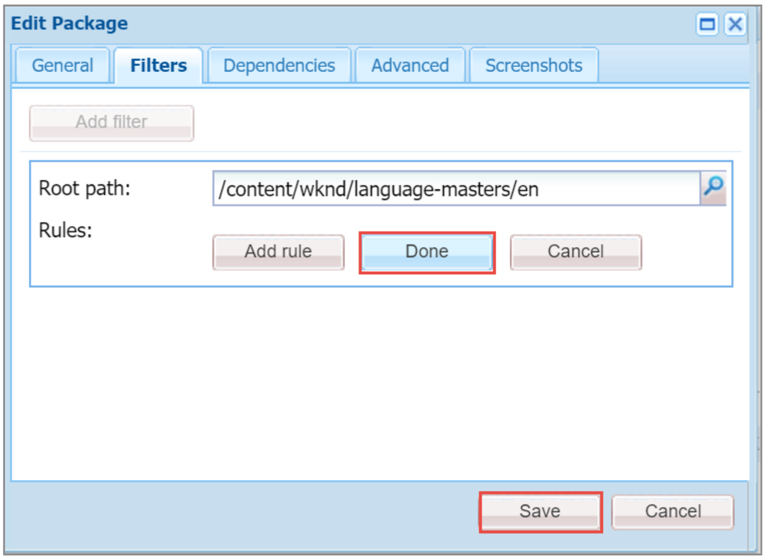
6. Click the **Filters** tab and click the **Add filter** button, as shown:



Edit Package

7. Add **/content/wknd/language-masters/en** in the **Root Path** box.

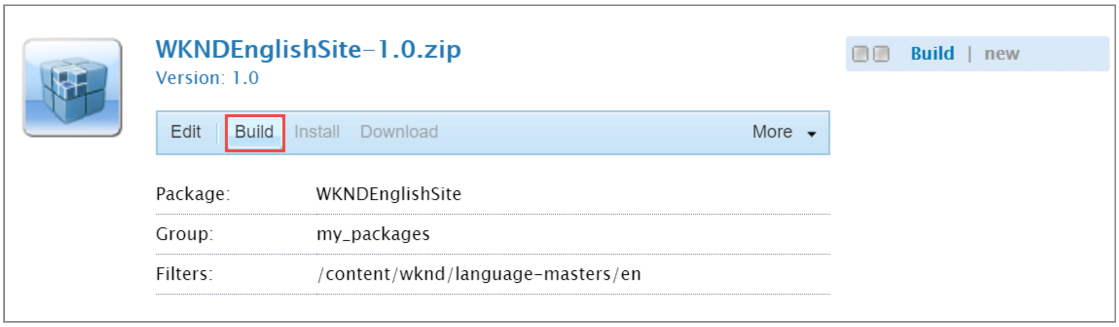
8. Click **Done** and **Save** to save the filter and the package definition, as shown:



Root path: /content/wknd/language-masters/en

Rules:

9. Click **Build** on the actions bar, as shown:

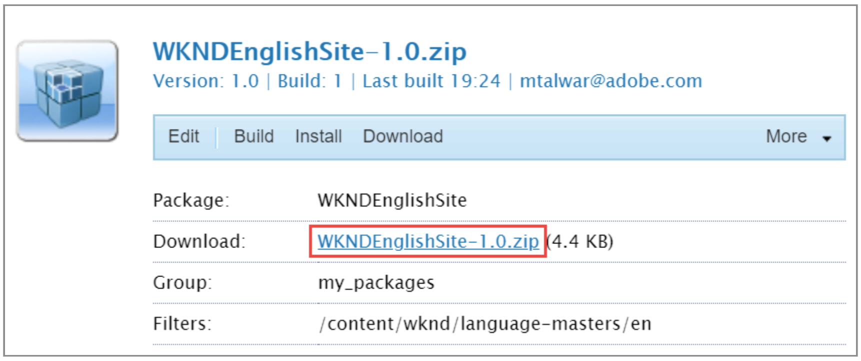


Package:	WKNDEnglishSite
Group:	my_packages
Filters:	/content/wknd/language-masters/en

A confirmation dialog box opens.

10. Click **Build**. The content package is created.

11. Click the **WKNDEnglishSite-1.0.zip** link to download the package to your **Downloads** folder, as shown:


WKNDEnglishSite-1.0.zip
Version: 1.0 | Build: 1 | Last built 19:24 | mtalwar@adobe.com
Edit
Build
Install
Download
More

Package:	WKNDEnglishSite
Download:	WKNDEnglishSite-1.0.zip (4.4 KB)
Group:	my_packages
Filters:	/content/wknd/language-masters/en

Task 3: Upload a mutable content package of test content

In this task, you will upload the new test content.

1. Ensure you are on the **Package Manager** page.

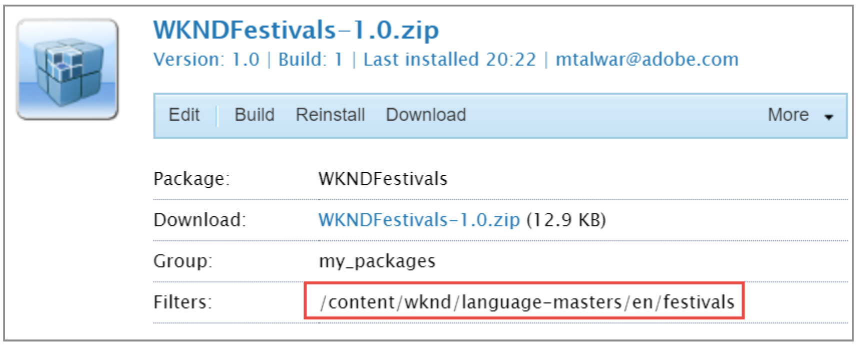
2. Click **Upload Package** on the actions bar. The **Upload Package** dialog box opens.

3. Click **Browse** in the dialog box. The **Open** dialog box opens.

4. On your file system, navigate to the **Exercise_Files/Environments** folder, select **WKNDFestivals-1.0.zip** and click **Open**. The selected package is added to the **Upload Package** dialog box.

5. Click **OK**. The package is uploaded.

6. Notice the filter path: **/content/wknd/language-masters/English/festivals** on the package, as shown:



WKNDFestivals-1.0.zip
Version: 1.0 | Build: 1 | Last installed 20:22 | mtalwar@adobe.com

Package:	WKNDFestivals
Download:	WKNDFestivals-1.0.zip (12.9 KB)
Group:	my_packages
Filters:	/content/wknd/language-masters/en/festivals

The package will upload the Festivals page and its children below the English page.

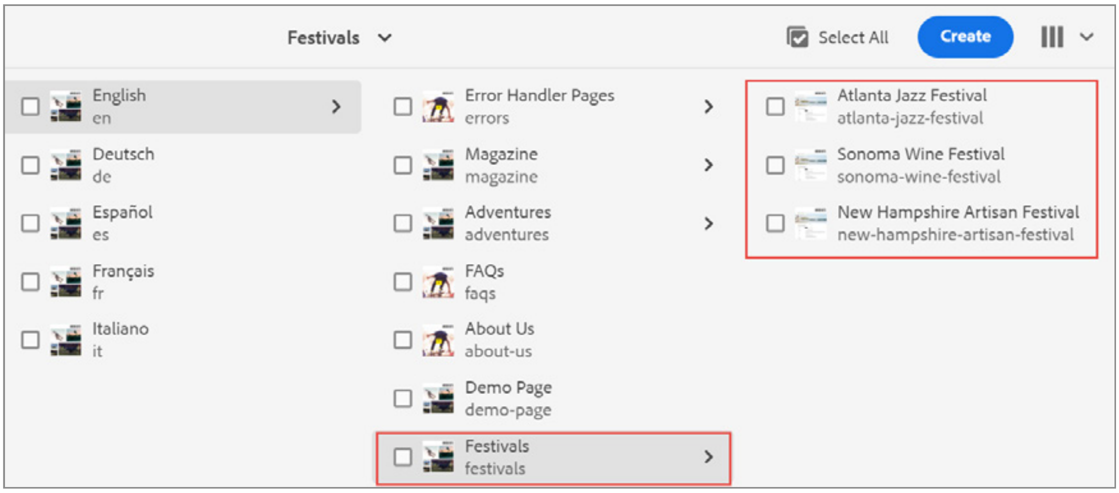
7. Click **Install** on the toolbar. The **Install Package** dialog box opens.

8. Click **Install**. The package is installed.

9. To verify whether the package install created the Festivals page and its children, click **Adobe Experience Manager** on the header bar. The **Navigation** pane opens.

10. Click **Sites** to open the console.

11. Navigate to **WKND Site** > **Language Masters** > **English** > **Festivals**, as shown:



□ English en
□ Deutsch de
□ Español es
□ Français fr
□ Italiano it

□ Error Handler Pages errors
□ Magazine magazine
□ Adventures adventures
□ FAQs faqs
□ About Us about-us
□ Demo Page demo-page
□ Festivals festivals

□ Atlanta Jazz Festival atlanta-jazz-festival
□ Sonoma Wine Festival sonoma-wine-festival
□ New Hampshire Artisan Festival new-hampshire-artisan-festival

Task 4: Reinstall the WKND mutable content package

In this task, you will return the test site to its original state before testing begins.

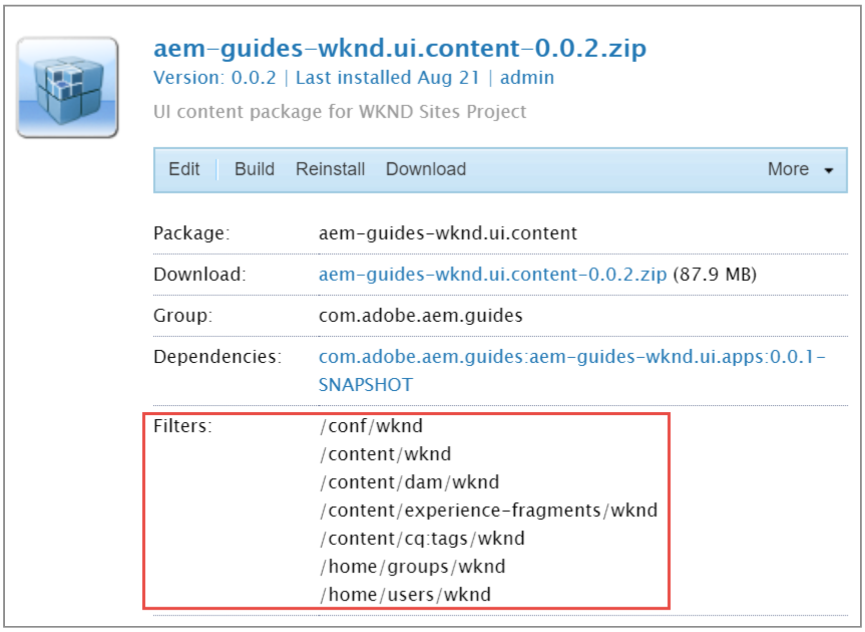
1. Click **Adobe Experience Manager** on the header bar. The **Navigation** pane opens.

2. Navigate to **Tools** > **Deployment** > **Packages**. The **Package Manager** page opens.

3. Type **aem-guides** in the **Search packages** box at the top and press the Enter key. This will find the packages associated with the WKND application

4. Click **aem-guides-wknd.ui.content-0.0.2.zip** to select it.

5. Notice the filters included in the package definition, as shown. The path **/content/wknd** is important, as it is the parent to **/content/wknd/language-masters/English/festivals** and **/content/wknd/language-masters/English/demo-page**:



aem-guides-wknd.ui.content-0.0.2.zip

Version: 0.0.2 | Last installed Aug 21 | admin

UI content package for WKND Sites Project

Package:	aem-guides-wknd.ui.content
Download:	aem-guides-wknd.ui.content-0.0.2.zip (87.9 MB)
Group:	com.adobe.aem.guides
Dependencies:	com.adobe.aem.guides:aem-guides-wknd.ui.apps:0.0.1-SNAPSHOT
Filters:	/conf/wknd /content/wknd /content/dam/wknd /content/experience-fragments/wknd /content/cq:tags/wknd /home/groups/wknd /home/users/wknd

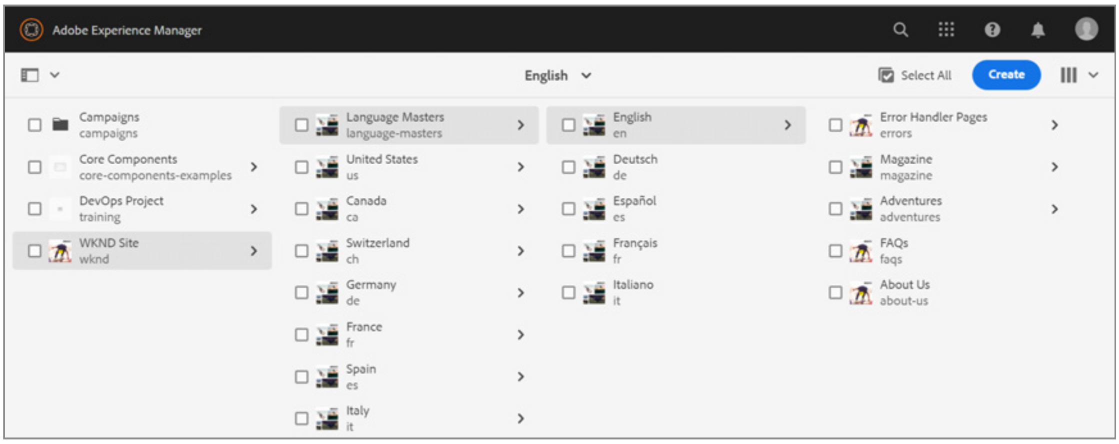
6. Click the **Reinstall** on the actions bar. The **Install Package** dialog box opens.

7. Click **Install** to reinstall the package.

8. To verify whether the package was installed successfully, click **CRX Package Manager** (Back Home icon) to open the **Navigation** pane.
. Click **Sites** to open the console.

9. Navigate to **WKND Site** > **Language Masters** > **English**.

10. Notice that the Demo Page that you created earlier and the Festivals pages are removed, as shown:



Campaigns campaigns
Core Components core-components-examples
DevOps Project training
WKND Site wknd

Language Masters language-masters
United States us
Canada ca
Switzerland ch
Germany de
France fr
Spain es
Italy it

English en
Deutsch de
Español es
Français fr
Italiano it

Error Handler Pages errors
Magazine magazine
Adventures adventures
FAQs faqs
About Us about-us

The original **aem-guides-wknd.ui.content-0.0.2.zip** package did not contain these pages. The reinstall returned the state of the website to the original state.

You have verified that mutable content can be created, packaged, uploaded and downloaded into or from an AEM as a Cloud Service author service.

References

1. Managing Environments https://docs.adobe.com/content/help/en/experience-manager-cloud-service/implementing/using-cloud-manager/manage-environments.html ↩

2. Manage Logs https://docs.adobe.com/content/help/en/experience-manager-cloud-service/implementing/using-cloud-manager/manage-logs.html ↩ ↩

3. Developer Console https://docs.adobe.com/content/help/en/experience-manager-cloud-service/implementing/developing/development-guidelines.html ↩ ↩

4. CI/CD Pipeline https://docs.adobe.com/content/help/en/experience-manager-cloud-service/implementing/using-cloud-manager/configure-pipeline.html ↩

5. Package Manager https://docs.adobe.com/help/en/experience-manager-cloud-service/implementing/deploying/overview.html#deploying-content-packages-via-cloud-manager-and-package-manager ↩