DevOps for AEM as a Cloud Service

Introduction to Cloud Manager

Student Workbook

Find your journey at learning.adobe.com >

Contents

DevOps for AEM as a Cloud Service

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe. Adobe assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

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

10-14-2020

Introduction

Cloud Manager for Adobe Experience Manager (AEM) is a cloud service that helps customers build, test, and deploy AEM applications hosted by Adobe. Cloud Manager enables customers to manage their custom code deployments on cloud environments with manageable pipeline automation and complete flexibility for their deployments timing or frequency.

Objectives

After completing this module, you will be able to:

- Explain the features and terminologies of Cloud Manager
- Describe Cloud Manager Git
- Get started with Cloud Manager
- Push a release candidate to Cloud Manager
- Describe Cloud Manager API
- Explain aio Cloud Manager and commands

Cloud Manager Features and Terminologies

Cloud Manager enables organizations to self- manage AEM in the cloud. It includes the Continuous Integration/Continuous Deployment (CI/CD)) framework to expedite the delivery of customizations or updates without compromising performance or security.

Features

The features of Cloud Manager are:

- CI/CD Pipeline: Helps speed up the delivery of custom code or updates, such as adding new components on the website. Customers can configure and start their CI/CD pipeline from the Cloud Manager UI. This pipeline helps execute a code scan to ensure that only high-quality applications pass through to the production environment.
- Self service Interface: Enables customers to easily access and manage the cloud environment and CI/CD pipeline for their AEM applications.

Customers define application- specific Key Performance Indicators (KPIs) - peak page views per minute and expected response time for a page load, that ultimately form the basis for measuring a successful deployment . The roles and permissions for different team members can be easily defined.

- Flexible Deployments: Helps deliver experiences according to changing business demands. The code is automatically deployed to an environment based on specific events, such as code commit. You can also schedule code deployments during specified time frames, even outside business hours.
- Autoscaling: Detects the need for additional capacity when the production environment is subject to unusually high load and automatically brings additional capacity online. Cloud Manager triggers the autoscaling provisioning process, sends a notification of the autoscaling event, and brings online the additional capacity within minutes. The additional capacity will be provisioned in the production environment, in the same regions and matching the same system specifications as the running Dispatcher/Publish nodes.

Terminologies used in Cloud Manager

Application: The set of customizations and configurations created by a customer (or their customizer) to adapt the underlying solution for their specific use cases and needs. An application is a logical unit which may be composed of multiple artifacts.

Artifact: A deployable unit. The result of some build process that transforms source code into a single unit. For example a Zip file containing the source code is an artifact.

Artifact Repository: A storage location where customer-specific artifacts are saved and secured.

Environment: A single cluster of virtual machines within a program. For AEM, this is composed of an author instance (optionally with an additional cold standby author instance), zero or more publish instances, one or more dispatcher instances, and a load balancer.

Git Repository: A location where customer-specific source code is stored and made accessible by using the Git protocol.

Instance: A specific virtual server running the AEM solution. Instances represent a single logical unit from a deployment perspective.

Organization: An Adobe construct representing an Enterprise customer. One company may have multiple organizations depending on how they were originally provisioned in Adobe's Identity Management System.

Pipeline: A set of deployment steps that are executed in sequence.

Product: A specific set of functionality within a solution licensed by an organization. Different programs within an organization may be entitled to different sets of products. For example, Sites, Assets of Forms.

Program: A set of environments that support a logical grouping of customer initiatives and usually corresponds to a purchased Service Level Agreements (SLA). Each program has exactly one production environment and may have many non-production environments.

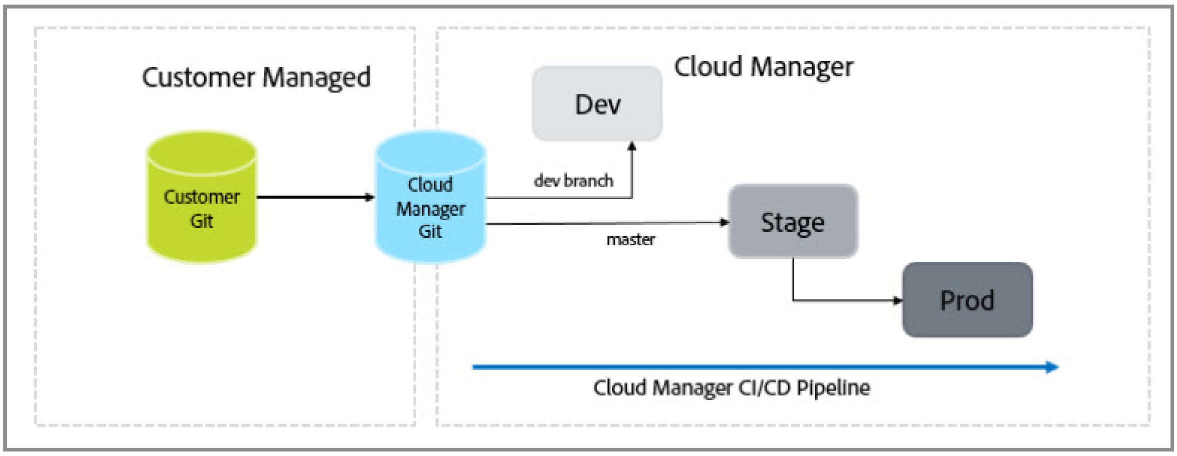
Solution: One of the Adobe Experience Cloud solutions, such as AEM, or Adobe Target, or Adobe Analytics.

Step: A configured instruction set that accomplishes some unit of work, building block of a pipeline.

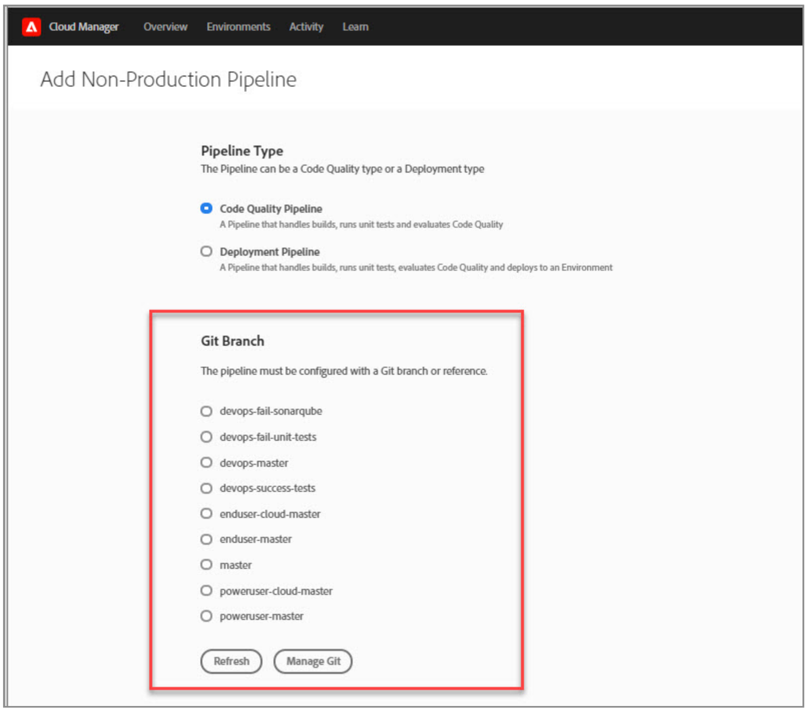
Tip: To learn more about Cloud Manager, see [1]

Cloud Manager Git

Cloud Manager is provisioned with a single git repository for deploying code by using the CI/CD pipelines. Customers can use the Cloud Manager's git repository or integrate an on- premise or customer-managed git repository with Cloud Manager.



You can access and manage your Git repository by using Self-Service Git Account Management from the Cloud Manager UI. You can click the Manage Git option to find the URL to your Cloud Manager Git repository, the user name, and the password. This option is visible to users in the Developer or Deployment Manager role.



Add Non-Production Pipeline

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-cloud-master
○ enduser-master
○ master
○ poweruser-cloud-master
○ poweruser-master

Exercise 1: Get started with Cloud Manager

Scenario: When working with AEM as a Cloud Service, you will need to access Cloud Manager to gain access to the author/publish URLs, run pipelines to install your custom code, and manage your AEM Services. In this exercise, you will log into the training organization, access a program, observe empty author/publish service containers, and then run a pipeline on an environment.

Prerequisites:

- Logged into the program assigned to you by the instructor
- An AEM environment already created

This exercise includes the following tasks:

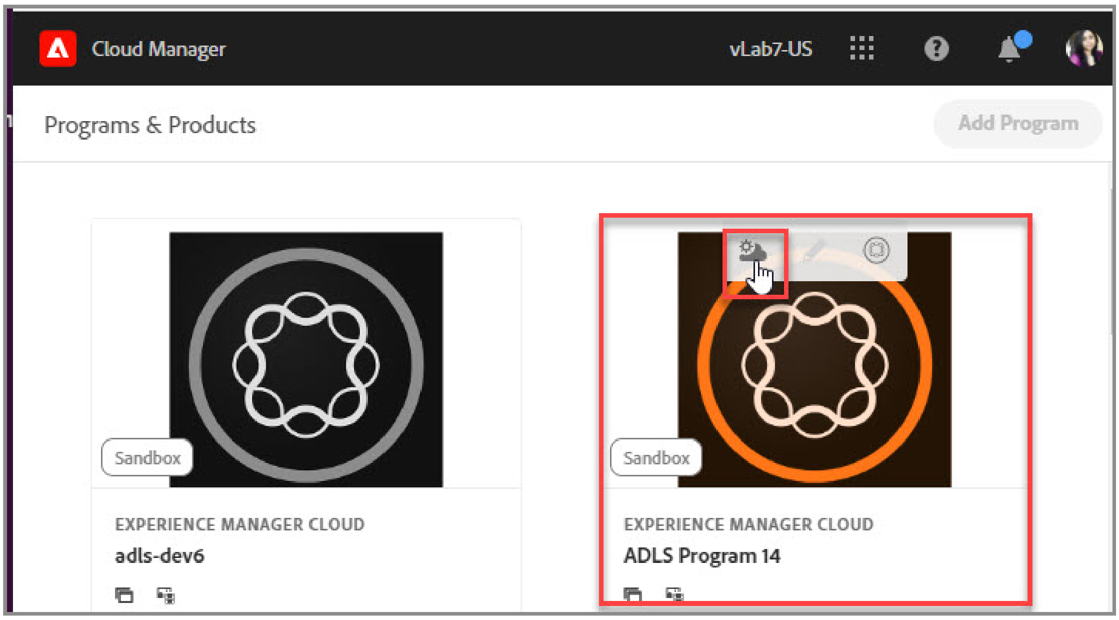
1. Access the AEM author and publish service

2. Run a pipeline

Task 1: Access the AEM author and publish service

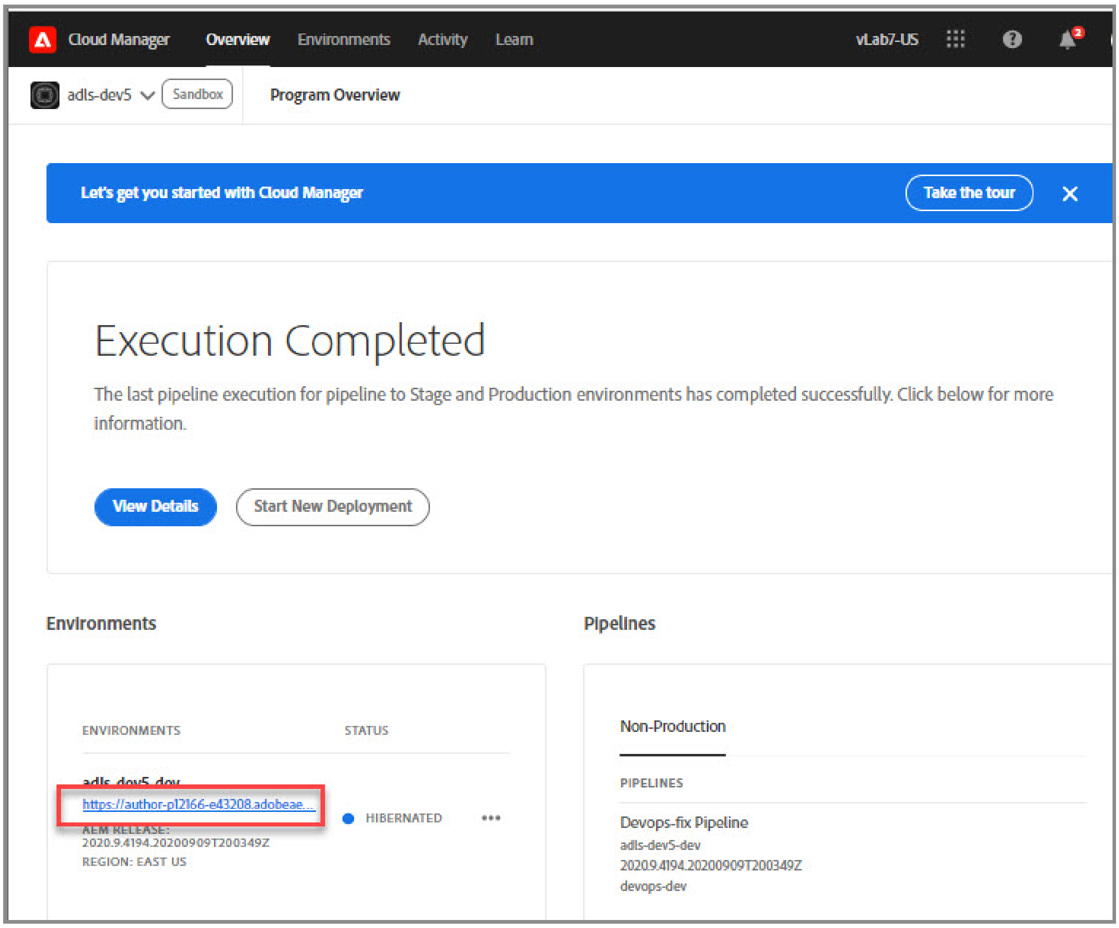
1. Sign in to https://experience.adobe.com/#/cloud-manager/landing.html, if not already signed in.

2. Hover over the **Program** your instructor has specified for you and click the Cloud icon, as shown:



The **Program Overview** page opens.

3. On the **Program Overview** page, click the author URL below the **Environments** section, as shown:



Execution Completed

The last pipeline execution for pipeline to Stage and Production environments has completed successfully. Click below for more information.

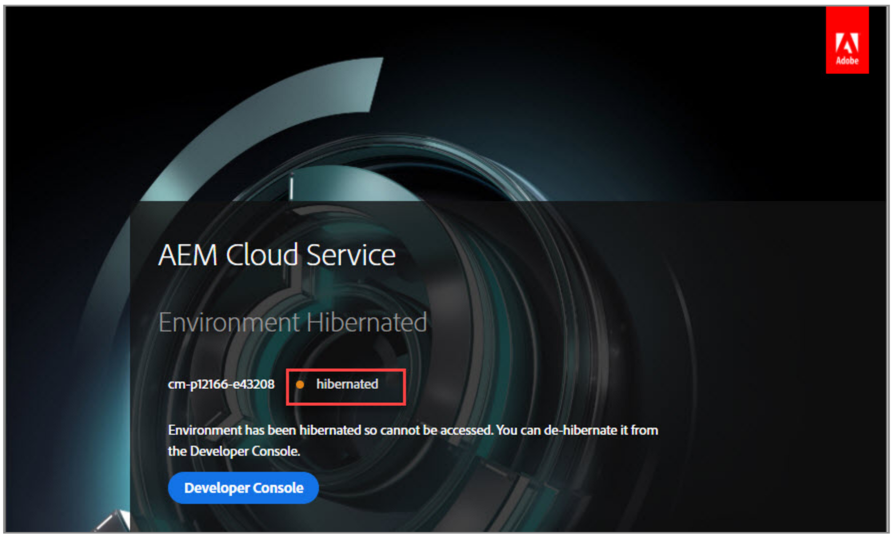
ENVIRONMENTS	STATUS
adls-dev5-dev https://author-p12166-e43208.adobeae... AEM RELEASE: 2020.9.4194.20200909T200349Z REGION: EAST US	● HIBERNATED

Non-Production

PIPELINES

Devops-fix Pipeline
adls-dev5-dev
2020.9.4194.20200909T200349Z
devops-dev

Notice the status is hibernated, as shown:



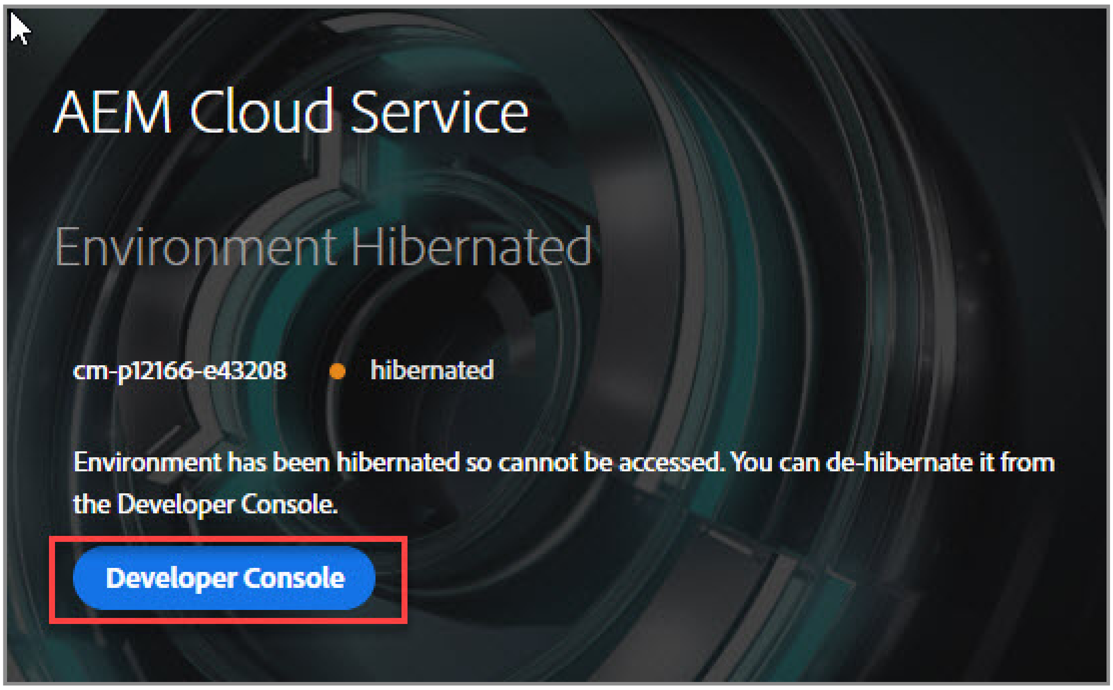
AEM Cloud Service

Environment Hibernated

cm-p12166-e43208 ● hibernated

Environment has been hibernated so cannot be accessed. You can de-hibernate it from the Developer Console.

4. Click on the **Developer Console** button, as shown:



AEM Cloud Service

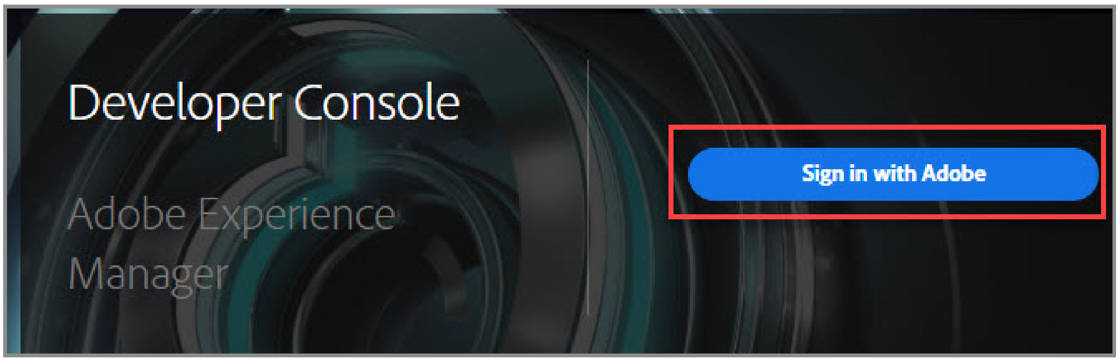
Environment Hibernated

cm-p12166-e43208 ● hibernated

Environment has been hibernated so cannot be accessed. You can de-hibernate it from the Developer Console.

The **Developer Console** page opens.

5. Click **Sign in with Adobe**, as shown:

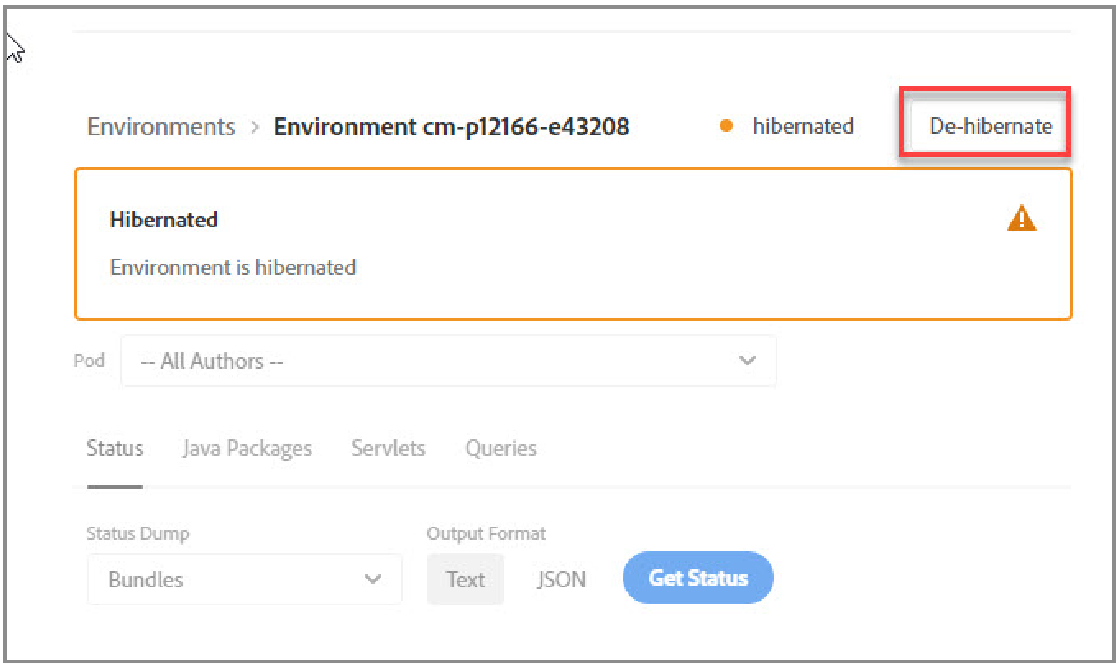


Developer Console

Adobe Experience Manager

The Developer Console > Environments page opens.

6. Click **De-hibernate**, as shown:



Environments > Environment cm-p12166-e43208 ● hibernated

Hibernated

Environment is hibernated

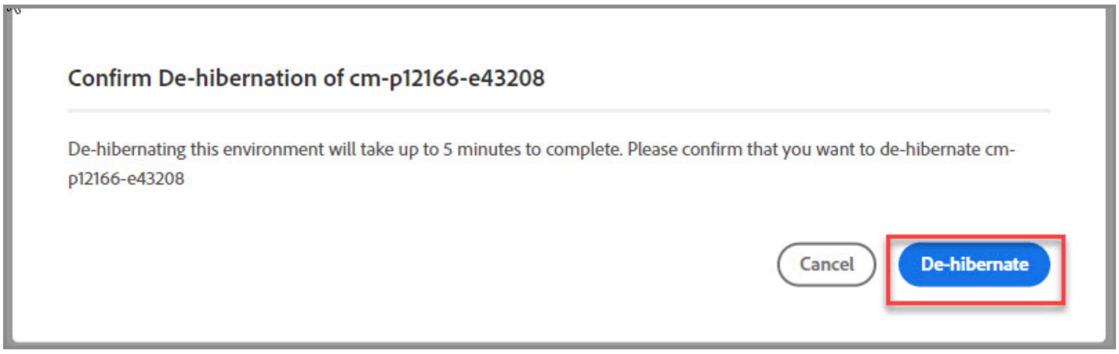
Pod: -- All Authors --

Status Dump: Bundles

Output Format: Text

A confirmation dialog box appears.

7. Click **De-hibernate** again on the Confirm dialog box, as shown:

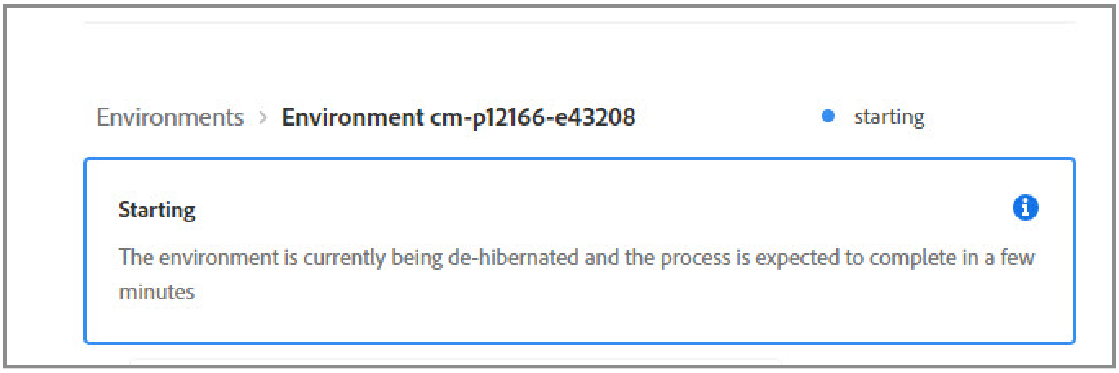


Confirm De-hibernation of cm-p12166-e43208

De-hibernating this environment will take up to 5 minutes to complete. Please confirm that you want to de-hibernate cm-p12166-e43208

A Success message appears confirming the de-hibernation process has started.

8. Notice the status of the Environment is starting, as shown:



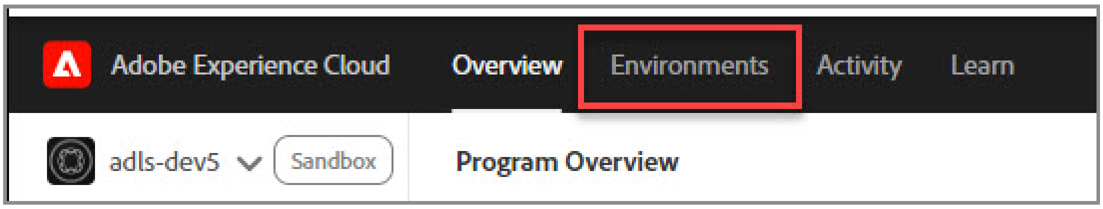
Environments > Environment cm-p12166-e43208 ● starting

Starting

The environment is currently being de-hibernated and the process is expected to complete in a few minutes

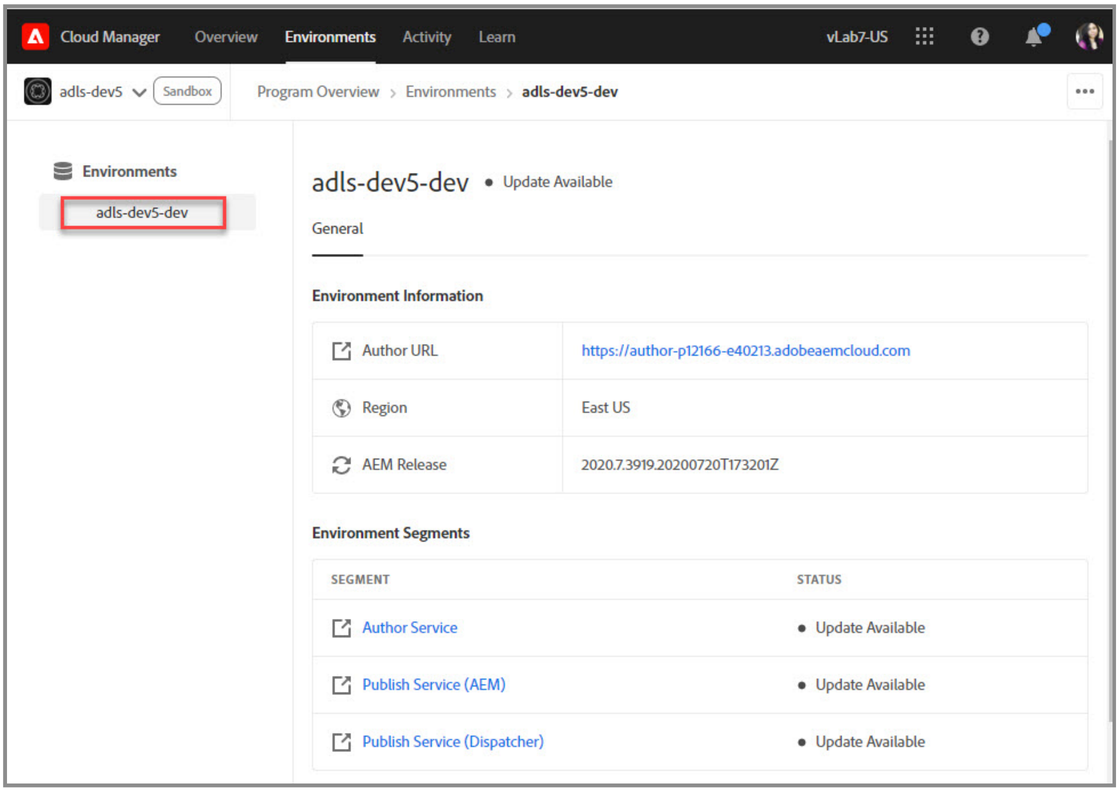
This process might take approximately 30-40 minutes to complete.

9. Go back to **Cloud manager**. In the black bar at the top, click **Environments**, as shown:



The **Environments** details are displayed.

10. Click **Environments** > **adls-< Program >**, as shown:



adls-dev5-dev • Update Available

Environment Information

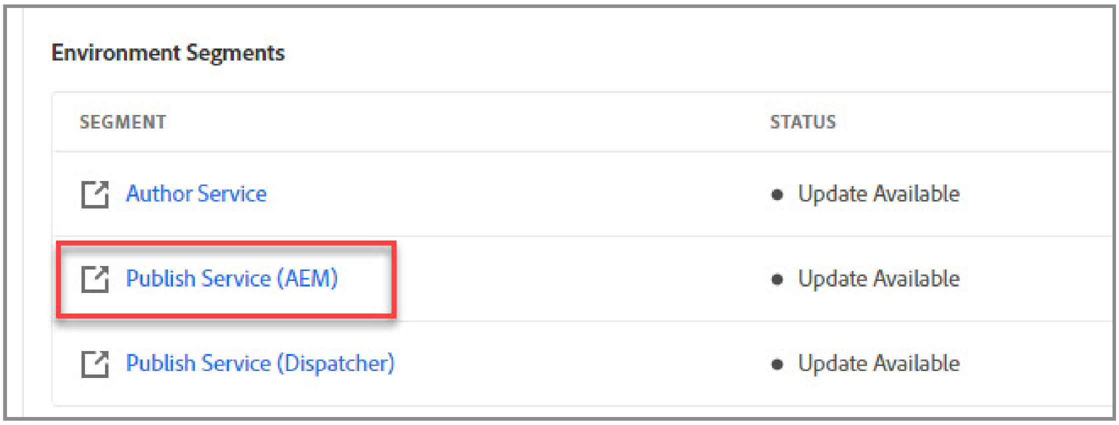
Author URL	https://author-p12166-e40213.adobeaemcloud.com
Region	East US
AEM Release	2020.7.3919.20200720T173201Z

Environment Segments

SEGMENT	STATUS
Author Service	● Update Available
Publish Service (AEM)	● Update Available
Publish Service (Dispatcher)	● Update Available

The services currently running within your Program are displayed.

11. Click **Publish Service (AEM)**, as shown:



Environment Segments

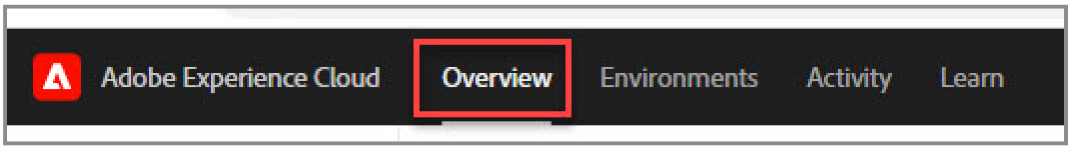
SEGMENT	STATUS
Author Service	● Update Available
Publish Service (AEM)	● Update Available
Publish Service (Dispatcher)	● Update Available

The **Publish Service** opens. Note that you may receive a 404 error because there is no website on the **Publish Service** yet.

Note: The publish service can only be accessed through the Dispatcher web server. (Learn more in the dispatcher module). Because of this, **Publish Service (AEM)** and **Publish Service (Dispatcher)** resolve to the same URL.

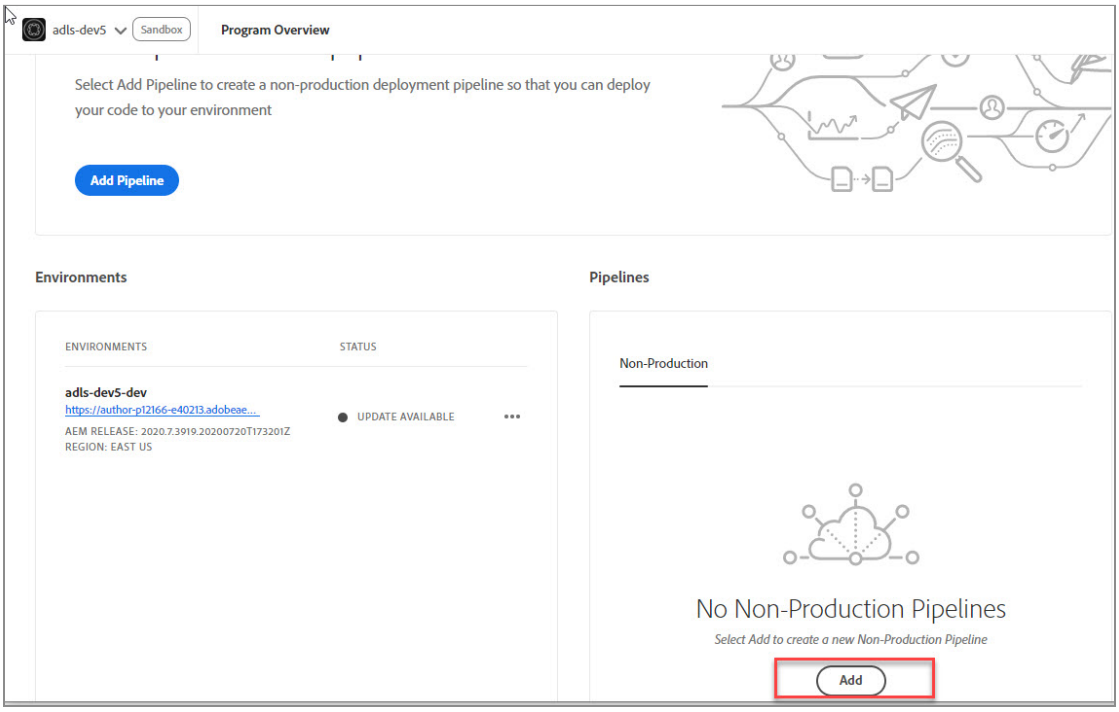
Task 2: Run a pipeline

1. On the browser where Cloud Manager is running, click **Overview** at the top, as shown:



The **Program Overview** page opens.

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



Select Add Pipeline to create a non-production deployment pipeline so that you can deploy your code to your environment

adls-dev5-dev
https://author-p12166-e40213.adobeae...
AEM RELEASE: 2020.7.3919.20200720T173201Z
REGION: EAST US
● UPDATE AVAILABLE

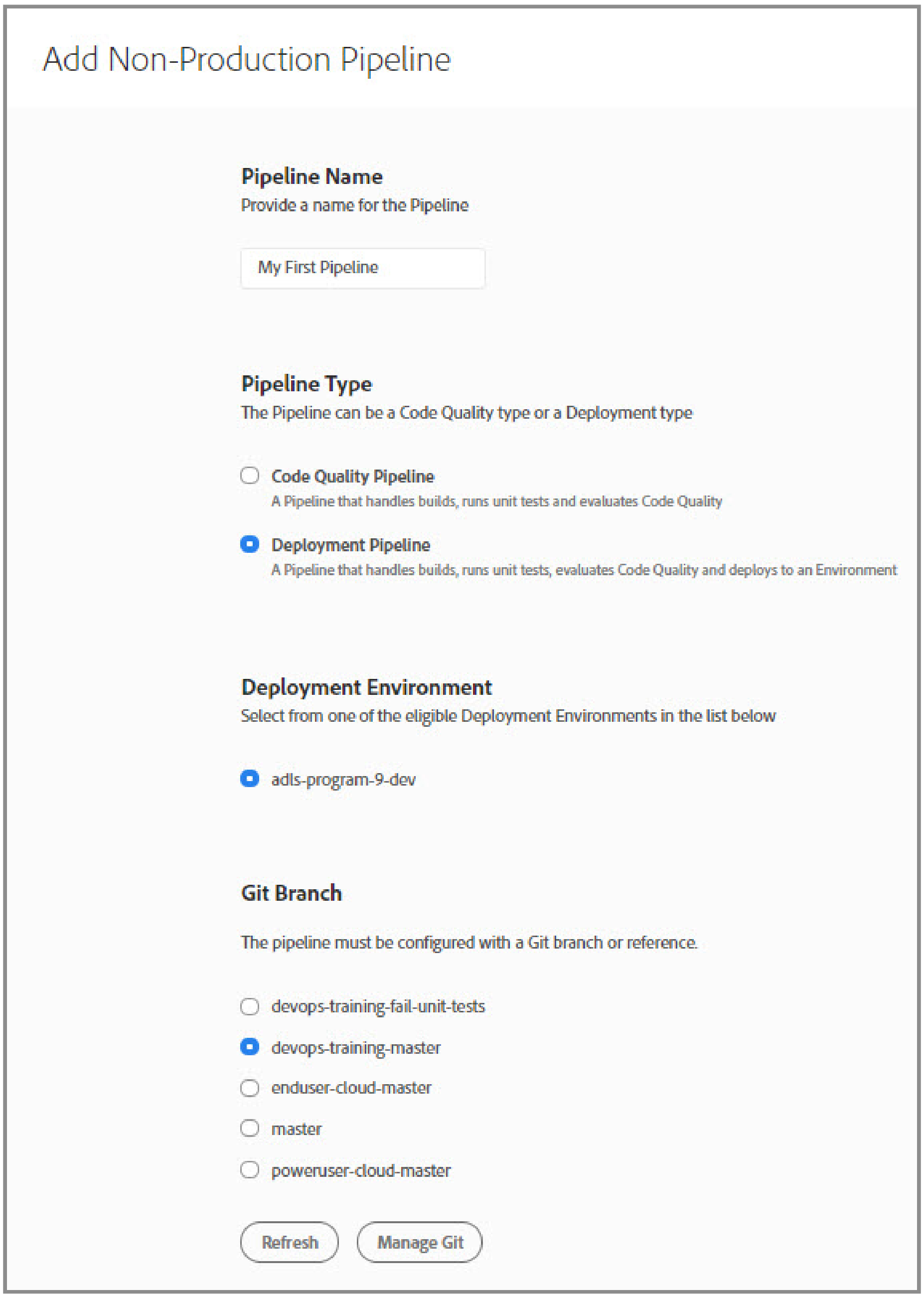
No Non-Production Pipelines

Select Add to create a new Non-Production Pipeline

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

3. On the **Add Non-Production Pipeline** screen, enter the following details, as shown:

- Pipeline Name: **My First Pipeline**
- Pipeline Type: **Deployment Pipeline**
- Deployment Environment: (select the only option)
- Git Branch: **devops-training-master**
- Pipeline Options: Keep Defaults



Add Non-Production Pipeline

Pipeline Name

Provide a name for the Pipeline

My First Pipeline

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

Deployment Environment

Select from one of the eligible Deployment Environments in the list below

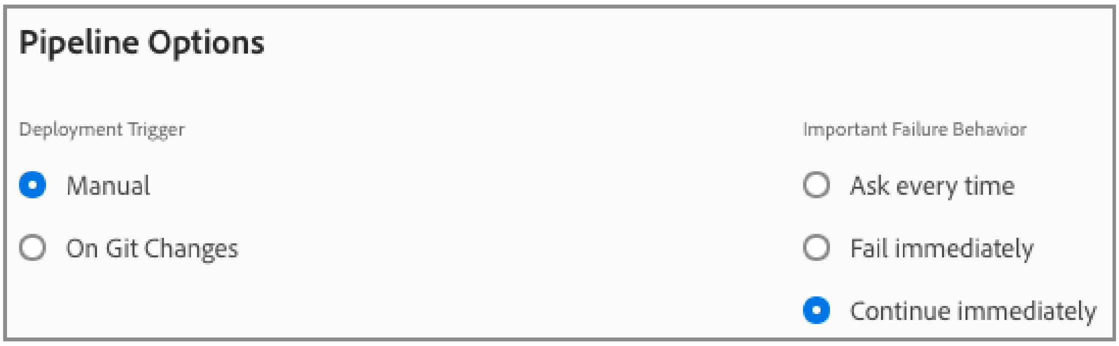
● adls-program-9-dev

Git Branch

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

○ devops-training-fail-unit-tests
● devops-training-master
○ enduser-cloud-master
○ master
○ poweruser-cloud-master

4. Under Pipeline Options, enter the following details, as shown:
   - Deployment Trigger: **Manual**
   - Important Failure Behavior: **Continue Immediately**



Pipeline Options

Deployment Trigger

● Manual

○ On Git Changes

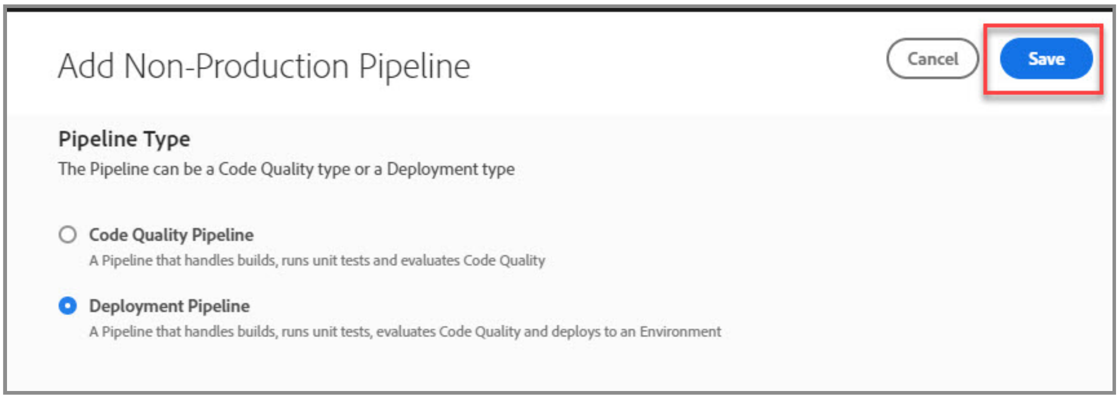
Important Failure Behavior

○ Ask every time

○ Fail immediately

● Continue immediately

5. Click **Save**.



Add Non-Production Pipeline

Pipeline Type

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

○ Code Quality Pipeline

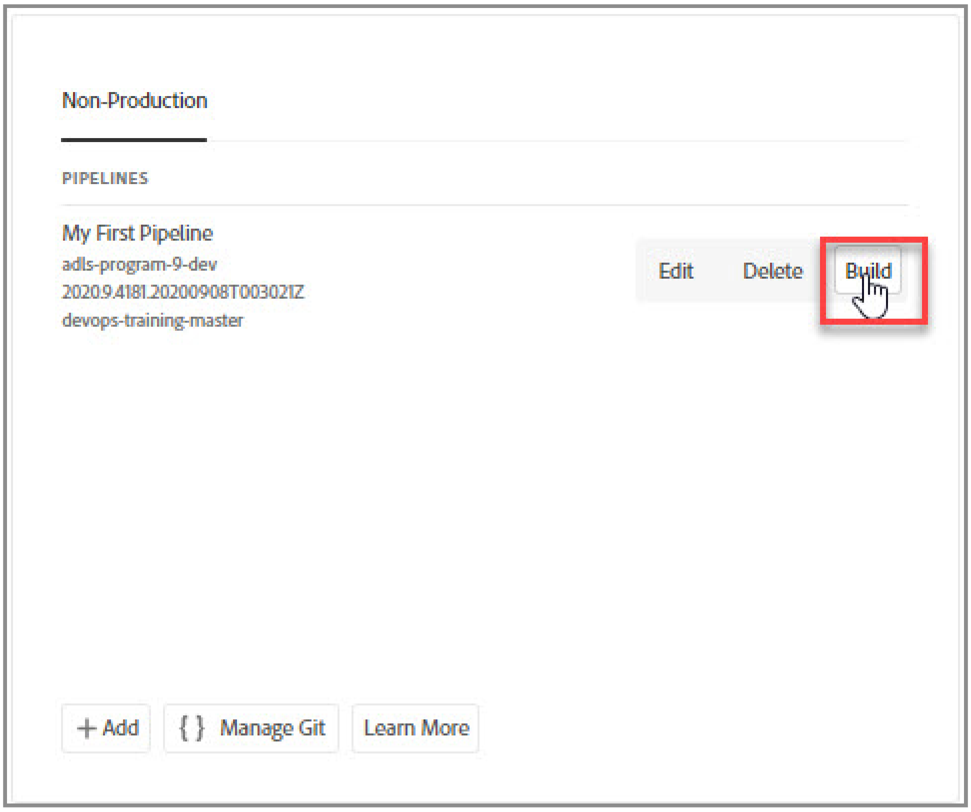
A Pipeline that handles builds, runs unit tests and evaluates Code Quality

● Deployment Pipeline

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

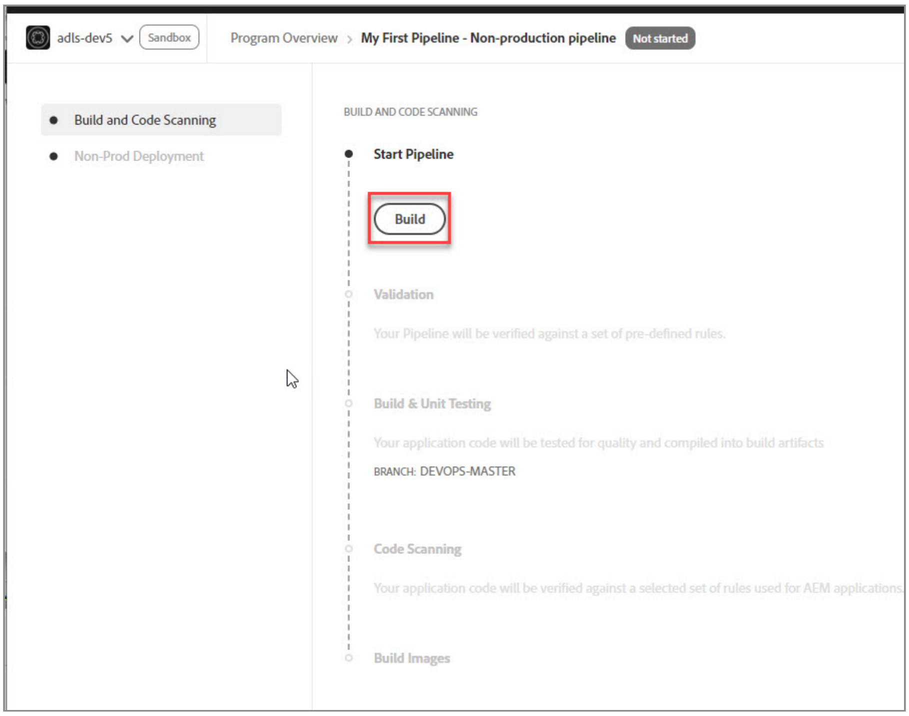
The **Non-Production** screen is displayed.

6. Hover over **My First Pipeline** and click **Build**, as shown. The **Build and Code Scanning** screen opens.



My First Pipeline
adls-program-9-dev
2020.9.4181.20200908T003021Z
devops-training-master

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



adls-dev5
Sandbox
Program Overview
My First Pipeline - Non-production pipeline
Not started
Build and Code Scanning
Non-Prod Deployment
BUILD AND CODE SCANNING
Start Pipeline
Build
Validation
Your Pipeline will be verified against a set of pre-defined rules.
Build & Unit Testing
Your application code will be tested for quality and compiled into build artifacts
BRANCH: DEVOPS-MASTER
Code Scanning
Your application code will be verified against a selected set of rules used for AEM applications.
Build Images

A pipeline on an average takes from 35 to 50 minutes to complete. You can look at the completed pipeline later in this class.

Exercise 2: Push a release candidate to Cloud Manager (Optional)

Scenario: In a realistic development scenario, a company would have access to push new code into the Cloud Manager Git. This exercise outlines typical commands you might use to push your to Cloud Manager Git. Although the commands are given, you cannot run these commands on your Maven project because you do not have an authorization token for the training program.

Prerequisites:

- Local Git project
- Credentials to Cloud Manager git

1. Open the Command Line and navigate to your **devops-training** project:

```
cd C:/adobe/devops-training
```

2. To test the dev branch against the cloud manager pipeline, switch to your dev branch:

```
git checkout dev
Switched to branch 'dev'
Your branch is up to date with 'origin/dev'.
```

3. Pull down any changes from the company repository.

```
git fetch origin
git merge origin/dev
```

Note: Pulling down any changes is always a good practice.

4. Now that you have the latest dev branch local, you need to create a remote downstream to the Cloud Manager git:

```
git remote add adobe <https-git-url>
git remote -v
adobe   https://git.cloudmanager.adobe.com/<programGIT> (fetch)
adobe   https://git.cloudmanager.adobe.com/<programGIT> (push)
origin  https://git.corp.acme.com/acme/devops-training.git (fetch)
origin  https://git.corp.acme.com/acme/devops-training.git (push)
```

The remote downstream is created.

5. You can now push your dev branch to the downstream remote repository, as shown:

```
git push adobe dev
Enumerating objects: 140, done.
Counting objects: 100% (128/128), done.
Delta compression using up to 12 threads
Compressing objects: 100% (70/70), done.
Writing objects: 100% (92/92), 7.72 KiB | 1.54 MiB/s, done.
Total 92 (delta 23), reused 0 (delta 0)
remote: Analyzing objects... (92/92) (1877 ms)
remote: Storing packfile... done (87 ms)
remote: Storing index... done (54 ms)
To https://git.cloudmanager.adobe.com/<programURL>
   72afd5c..fbbb448  dev -> dev
```

Tip: To learn more about this process, see [2]

Cloud Manager API

Cloud Manager, part of the Adobe Managed Cloud Services, enables organizations to self-manage Experience Manager environments in the cloud.

The Cloud Manager API enables Cloud Manager customers to interact with the same underlying capabilities exposed through the web UI in a fully programmatic manner. This allows for integration of the Cloud Manager CI/CD pipeline with other systems.

The Cloud Manager API is composed of two components:

- An inbound HTTP-based API which can be used by Cloud Manager customers to read and customize the state of their CI/CD pipelines.
- An outbound event system which enables Cloud Manager customers to receive events when key events happen in their CI/CD pipelines.

Tip: To learn more about inbound API, see [3] and for Event API, see [4]

Cloud Manager API Tutorial

The Cloud Manager API tutorial is a good place to get started with event driven activities with Cloud Manager. You can find the tutorial and more on adobe.io. [5]

Adobe I/O CLI (aio)

The aio platform for Adobe solutions provides command-line functionality for accessing their APIs[6]. The aio provides a core set of endpoints and syntax, and Adobe solutions add their own endpoints for specific functionality.

Adobe I/O Plugin: Cloud Manager

The aio plugin for Cloud Manager offers a host of different commands that enable you to control and manage the CI/CD pipeline and environments in Cloud Manager programs. The plugin to aio can be downloaded on githb[7]. Once authenticated, the plugin offers many different command line options:

- `aio cloudmanager`
- `aio cloudmanager:advance-current-execution PIPELINEID`
- `aio cloudmanager:cancel-current-execution PIPELINEID`
- `aio cloudmanager:delete-environment ENVIRONMENTID`
- `aio cloudmanager:delete-pipeline PIPELINEID`
- `aio cloudmanager:delete-program PROGRAMID`
- `aio cloudmanager:download-logs ENVIRONMENTID SERVICE NAME (DAYS)`
- `aio cloudmanager:get-current-execution PIPELINEID`
- `aio cloudmanager:get-execution-step-details PIPELINEID EXECUTIONID`
- `aio cloudmanager:get-execution-step-log PIPELINEID EXECUTIONID ACTION`
- `aio cloudmanager:get-quality-gate-results PIPELINEID EXECUTIONID ACTION`
- `aio cloudmanager:list-available-log-options ENVIRONMENTID`
- `aio cloudmanager:list-current-executions`
- `aio cloudmanager:list-environment-variables ENVIRONMENTID`
- `aio cloudmanager:list-environments`
- `aio cloudmanager:list-pipeline-variables PIPELINEID`
- `aio cloudmanager:list-pipelines`
- `aio cloudmanager:list-programs`
- `aio cloudmanager:open-developer-console ENVIRONMENTID`
- `aio cloudmanager:set-environment-variables ENVIRONMENTID`
- `aio cloudmanager:set-pipeline-variables PIPELINEID`
- `aio cloudmanager:start-execution PIPELINEID`
- `aio cloudmanager:tail-log ENVIRONMENTID SERVICE NAME`
- `aio cloudmanager:update-pipeline PIPELINEID`

References

1. Cloud Manager: https://docs.adobe.com/content/help/en/experience-manager-cloud-manager/using/introduction-to-cloud-manager.html ↩

2. Cloud Manager Integration: https://docs.adobe.com/content/help/en/experience-manager-cloud-manager/using/managing-code/setup-cloud-manager-git-integration.html ↩

3. Create API Integration: https://www.adobe.io/apis/experiencecloud/cloud-manager/docs.html#!AdobeDocs/cloudmanager-api-docs/master/create-api-integration.md ↩

4. Event API: https://www.adobe.io/apis/experiencecloud/cloud-manager/docs.html#!AdobeDocs/cloudmanager-api-docs/master/create-event-integration.md ↩

5. CLoud Manager API Tutorial: https://www.adobe.io/apis/experiencecloud/cloud-manager/docs.html#!AdobeDocs/cloudmanager-api-docs/master/tutorial/0-setup.md ↩

6. aio-cli Tool: https://github.com/adobe/aio-cli ↩

7. aio-ali cloudmanager: https://github.com/adobe/aio-cli-plugin-cloudmanager ↩