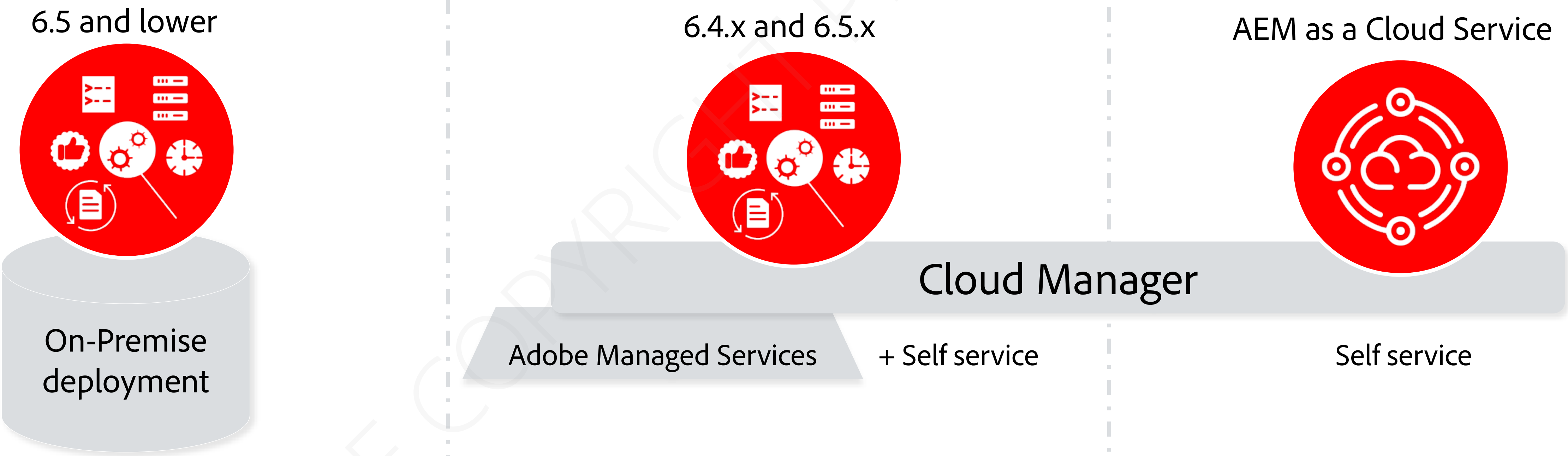
Cloud Manager Basics

Objectives:

- Describe Cloud Manager
- Understand basic Cloud Manager terminology
- Set up Program
- Define Pipeline with existing repository
- Explain custom code quality rules

Deployments for Adobe Experience Manager

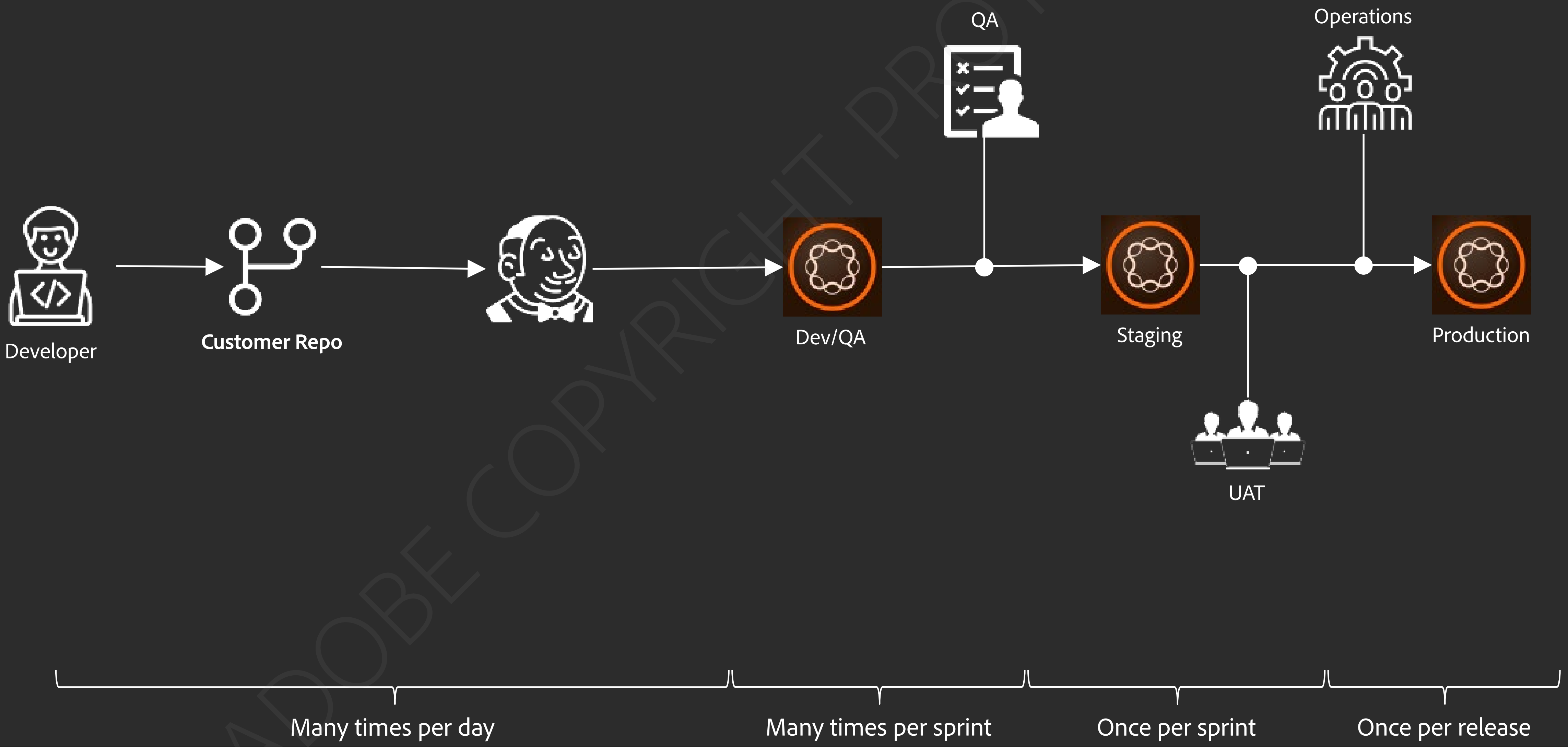


Adobe Experience Manager on a customer/partner managed deployment

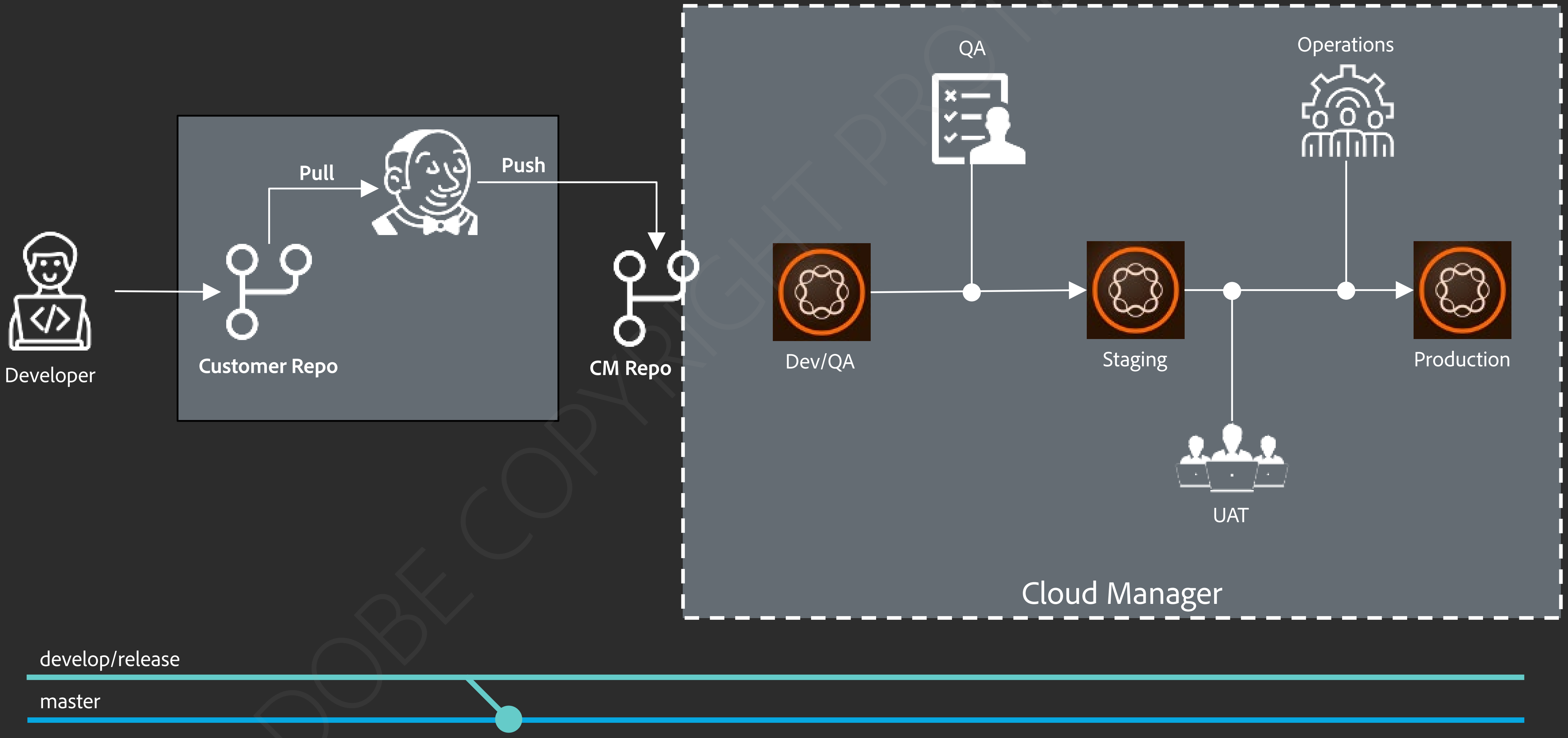
Adobe Experience Manager on Managed Services with Cloud Manager

Adobe Experience Manager as a Cloud Service

On-Premise Traditional Deployment Workflow



Cloud Manager Deployment Workflow



Cloud Manager for Adobe Experience Manager

Self Service Portal

Easily access & manage the cloud environment and CI/CD pipeline

CI/CD Pipeline

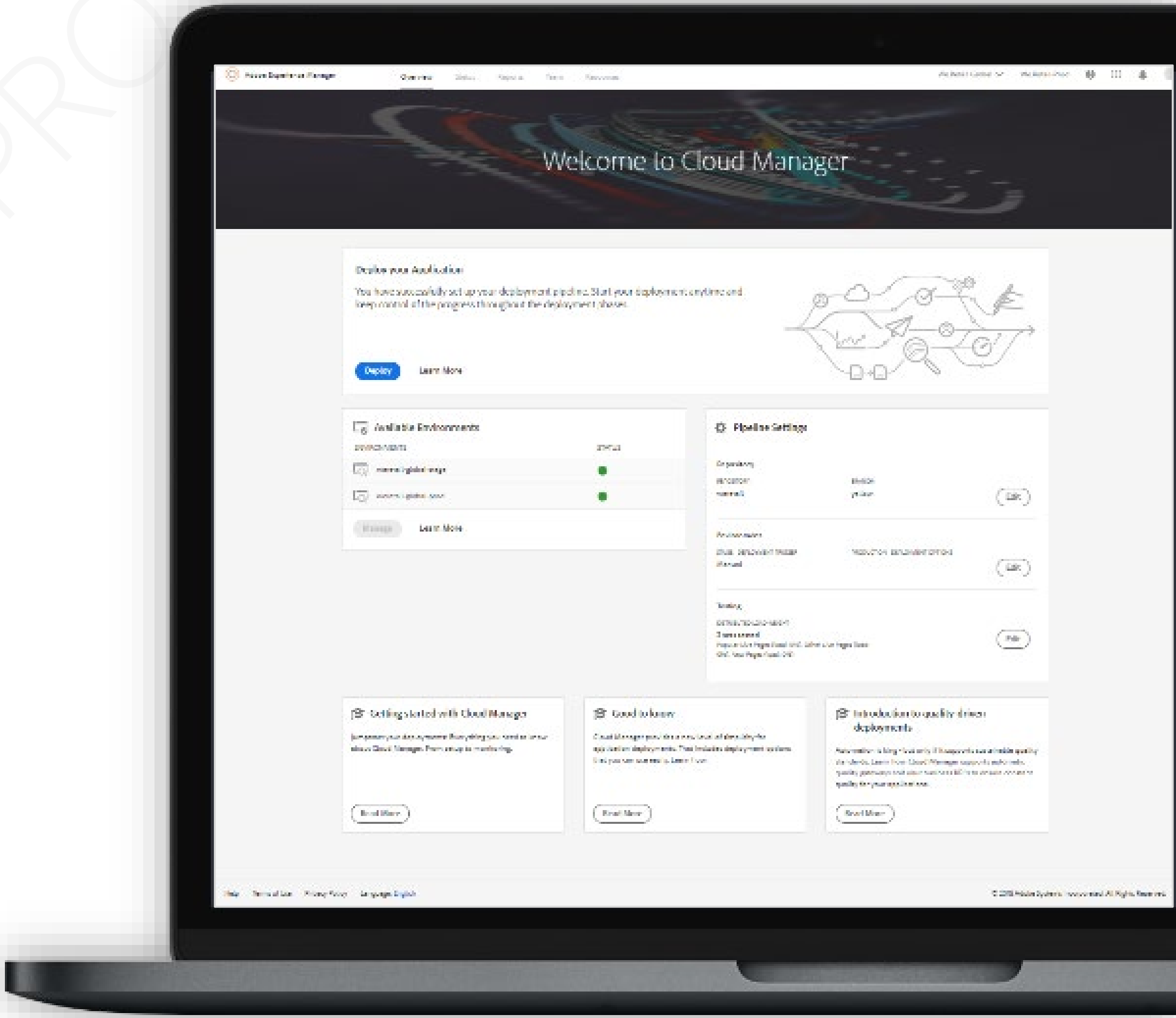
Code inspection, performance testing, and security validation

Flexible Deployments

Schedule code deployments during specified time frames

Auto-scaling

Detects the need for additional capacity during unexpected spikes



Welcome to Cloud Manager

Deploy your Application

You have successfully set up your deployment pipeline. Start your deployment anytime and keep control of the progress throughout the deployment phases.

Cloud Manager Entities

Cloud Manager is the single entry point for the Operations and Developer Personas to create and manage environments for the Adobe Experience Manager application

Programs

Self-service creation of AEM applications (AEM Sites, AEM Assets) with associated code repository

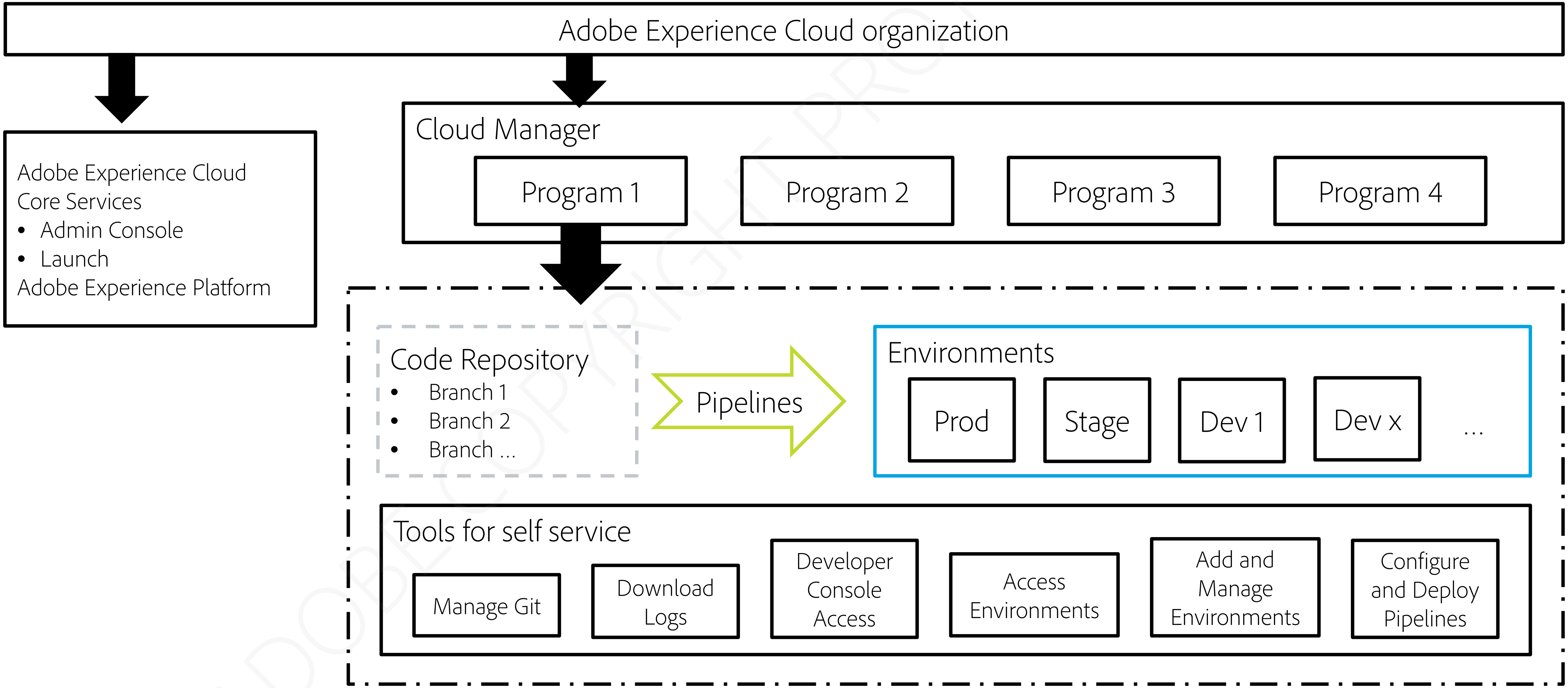
Environments

Self-service creation of AEM environments (production and non-production – for example, demos, POC, training…)

Pipelines

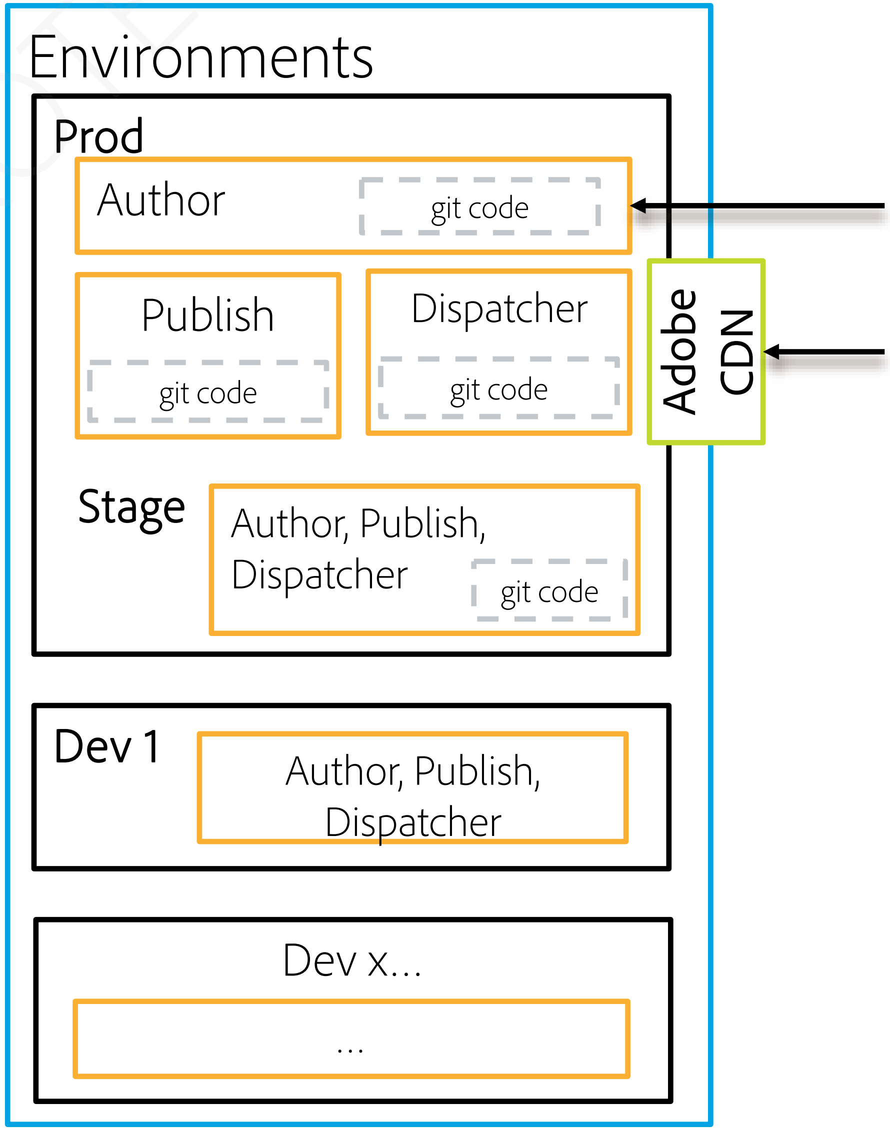
Self-service creation of CI/CD pipelines for best practices

Hierarchy of entities in Cloud Manager

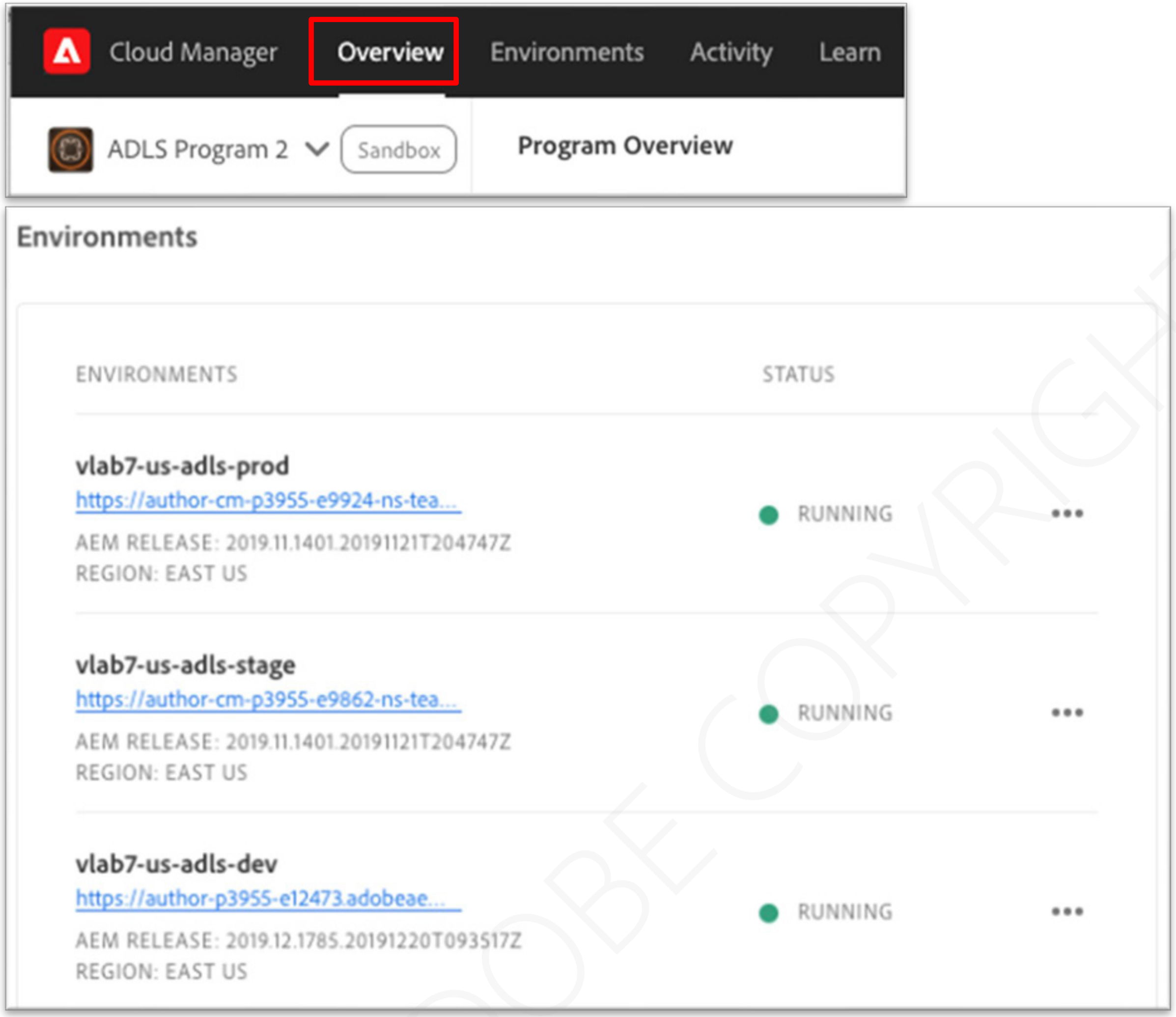


Environments

- 0-many Dev Environments
- 0-1 Stage/Prod Environment set
- Environments Consist of
  - o Author, Publish, and Dispatcher
- Environments are Autoconfigured with
  - o Replication
  - o Cloud native microservices
  - o Sensei Features like Smart Tag and Smart Crop
  - o Adobe IMS for authentication
- Retrievable Logs



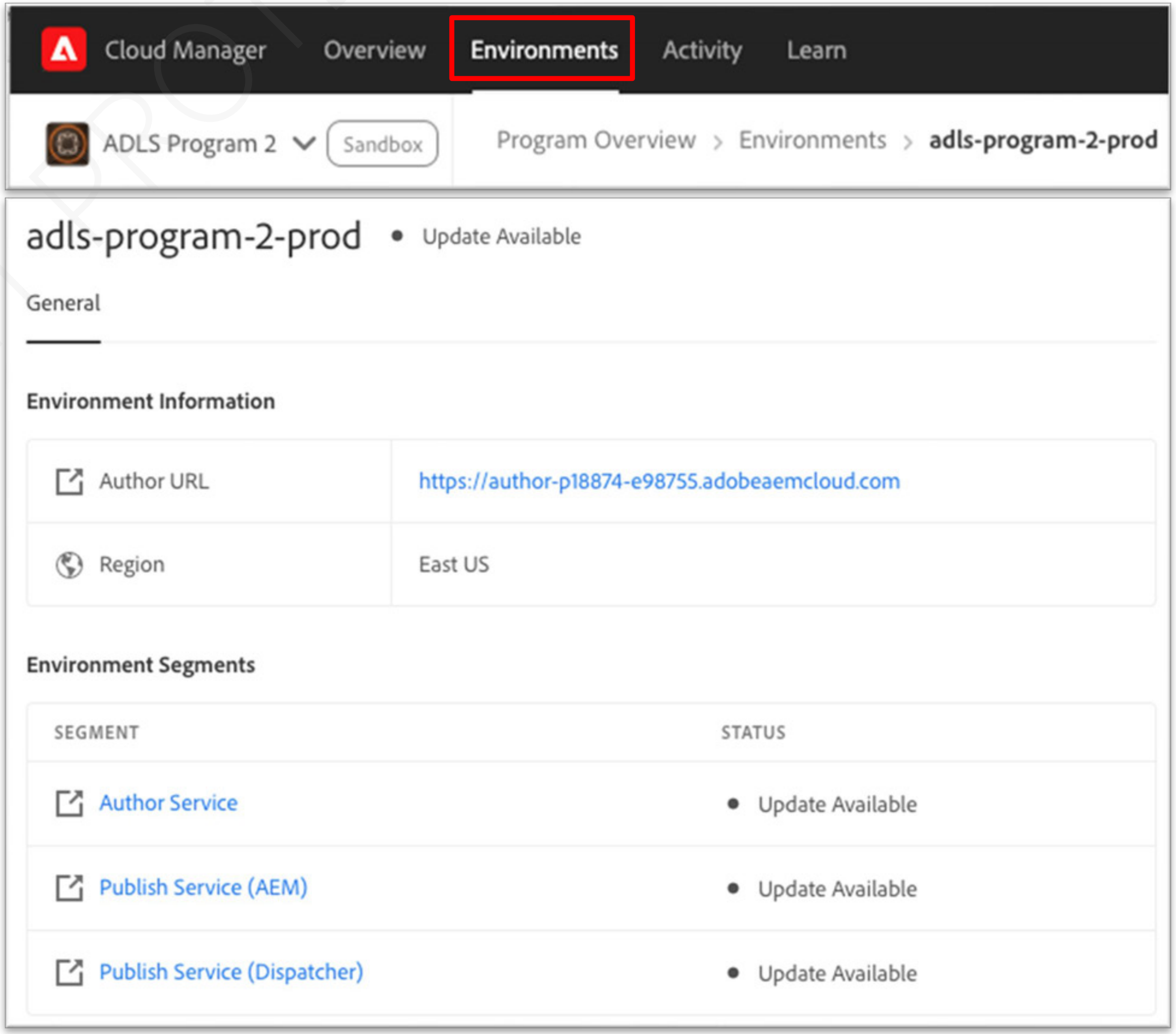
Environments UI



Program Overview

Environments

ENVIRONMENTS	STATUS	
vlab7-us-adls-prod https://author-cm-p3955-e9924-ns-tea... AEM RELEASE: 2019.11.1401.20191121T204747Z REGION: EAST US	● RUNNING	•••
vlab7-us-adls-stage https://author-cm-p3955-e9862-ns-tea... AEM RELEASE: 2019.11.1401.20191121T204747Z REGION: EAST US	● RUNNING	•••
vlab7-us-adls-dev https://author-p3955-e12473.adobeae... AEM RELEASE: 2019.12.1785.20191220T093517Z REGION: EAST US	● RUNNING	•••



Program Overview > Environments > **adls-program-2-prod**

adls-program-2-prod ● Update Available

General

Environment Information

Author URL	https://author-p18874-e98755.adobeaemcloud.com
Region	East US

Environment Segments

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

Environments | Logs

Downloadable Author or Publish Logs:

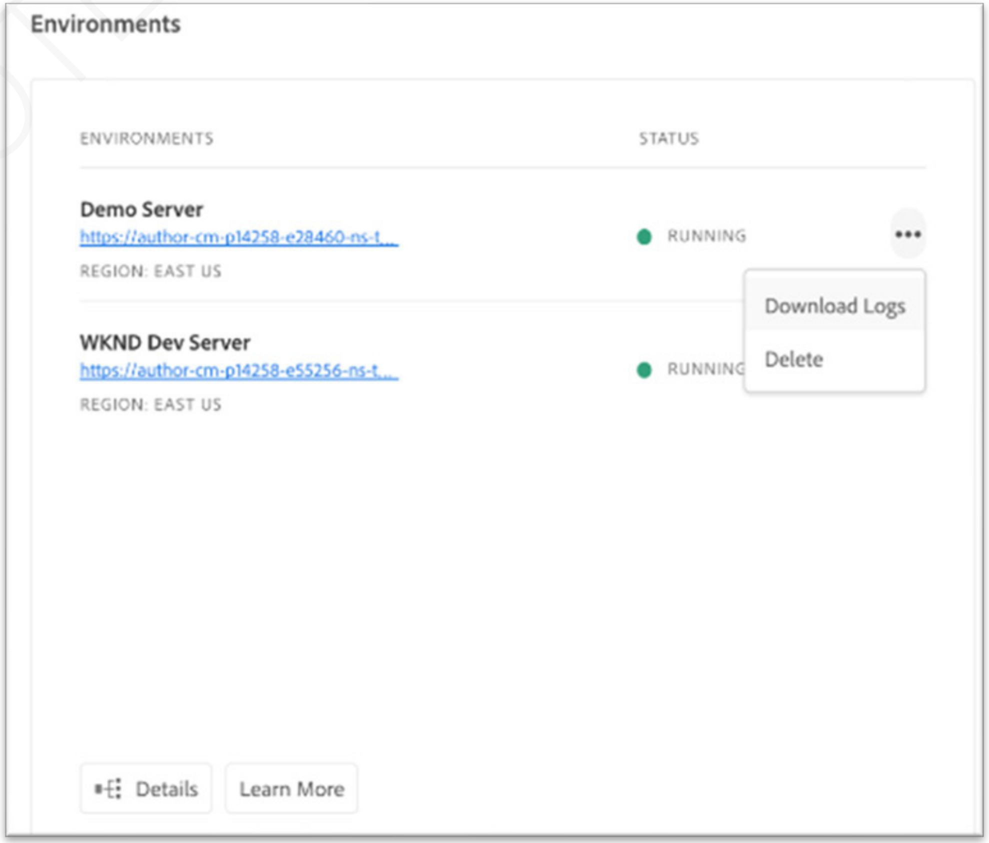
- aemaccess.log
- aemerror.log
- aemrequest.log

Downloadable Dispatcher logs:

- aemdispatcher.log
- httpdaccess.log
- httpderror.log

Logs available through

- Cloud Manager UI
- aio-cli cloudmanager plugin
  - o Download or tail logs



Environments

ENVIRONMENTS	STATUS
Demo Server https://author-cm-p14258-e28460-ns-t... REGION: EAST US	RUNNING
WKND Dev Server https://author-cm-p14258-e55256-ns-t... REGION: EAST US	RUNNING

Code Repository

1 git repository to 1 Cloud Manager Program

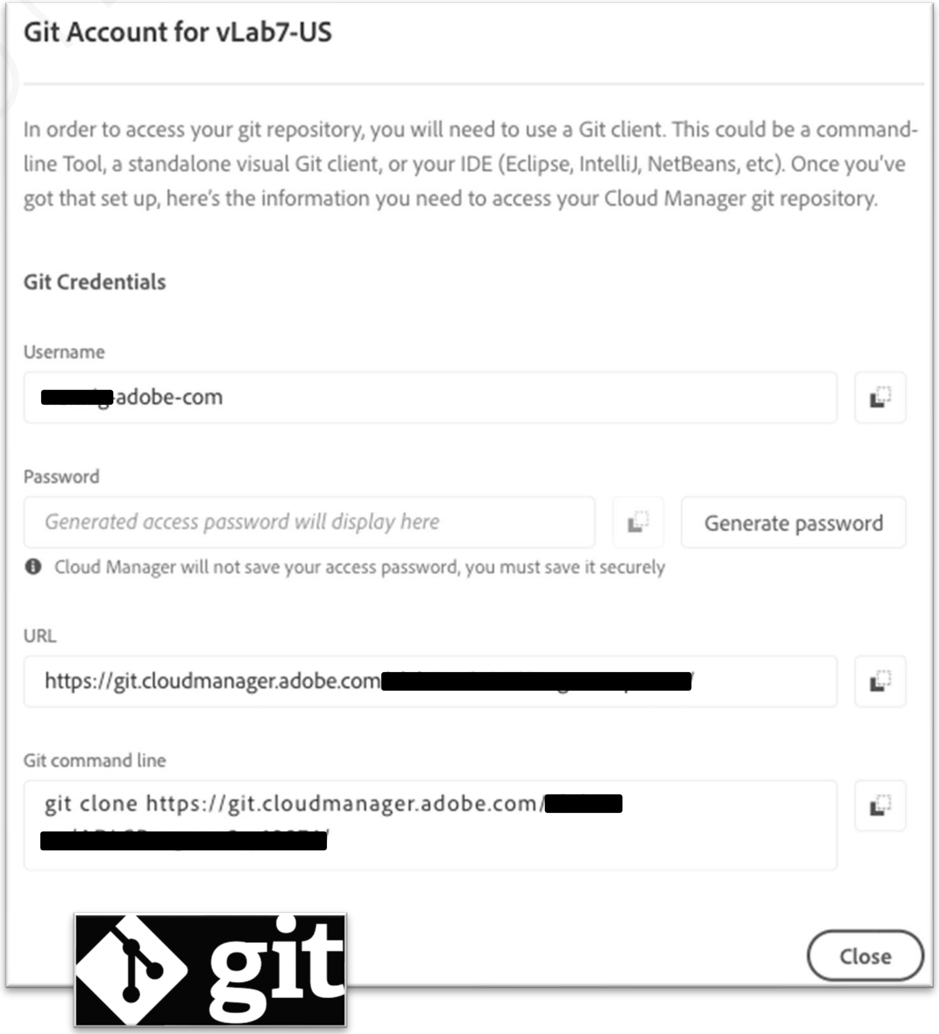
Access git account through a Git client using HTTPS

Different Branches can be used on different pipelines

- feature1 branch > code quality pipeline
- develop branch > QA pipeline
- main branch > production pipeline

Completed Production pipelines

- Auto create a release tag
- Tags can be pulled back to customer repos



Git Account for vLab7-US

In order to access your git repository, you will need to use a Git client. This could be a command-line Tool, a standalone visual Git client, or your IDE (Eclipse, IntelliJ, NetBeans, etc). Once you've got that set up, here's the information you need to access your Cloud Manager git repository.

Git Credentials

Username: [redacted]adobe-com

Password: *Generated access password will display here*

Cloud Manager will not save your access password, you must save it securely

URL: https://git.cloudmanager.adobe.com[redacted]

Git command line:

```
git clone https://git.cloudmanager.adobe.com/[redacted]
```

Pipelines

Goals of pipelines

- Run robust set of tests throughout all stages
- Build an image with a git branch and AEM build
- Deploy the image to an environment (optional)

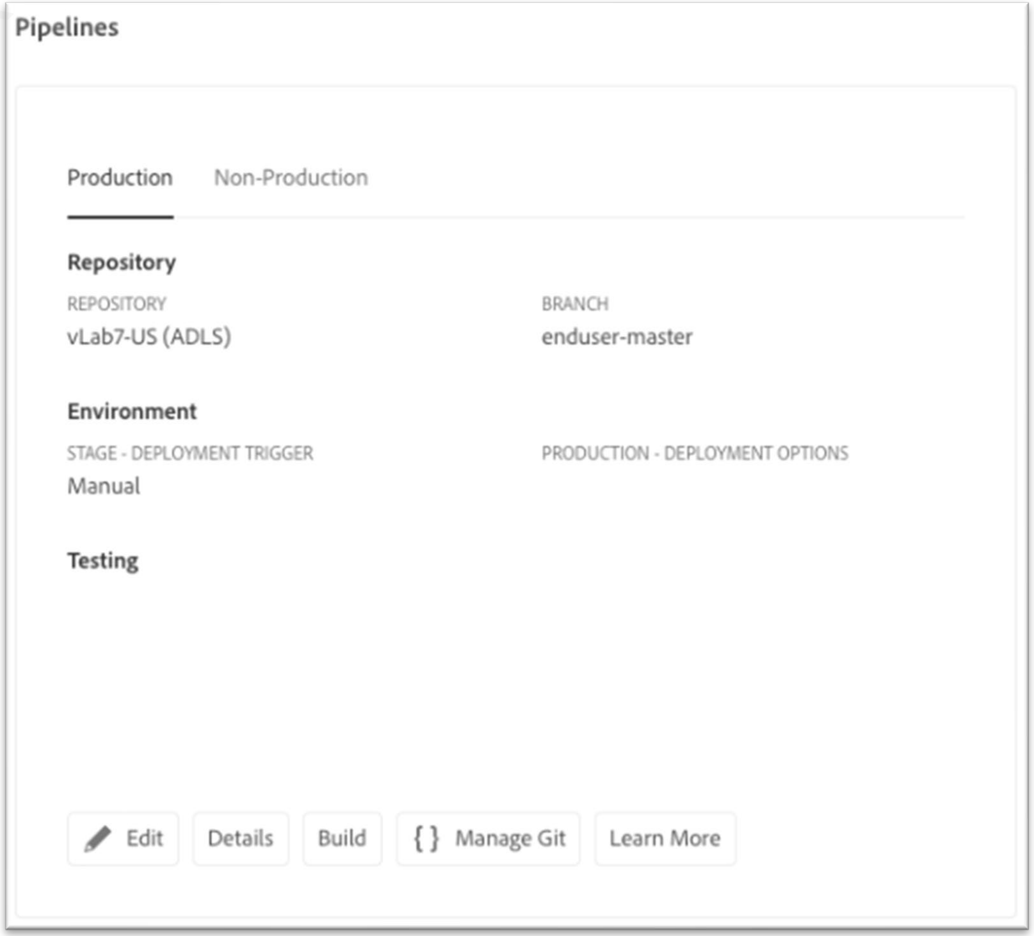
Triggered based On Git Changes or Manual

Configured Failure Behavior (non-critical)

- Ask every time
- Fail Immediately
- Continue Immediately

Types of pipelines

- Non-production code quality
- Non-production deployment
- Production



Pipelines

Production | Non-Production

Repository

REPOSITORY: vLab7-US (ADLS)

BRANCH: enduser-master

Environment

STAGE - DEPLOYMENT TRIGGER: Manual

PRODUCTION - DEPLOYMENT OPTIONS

Testing

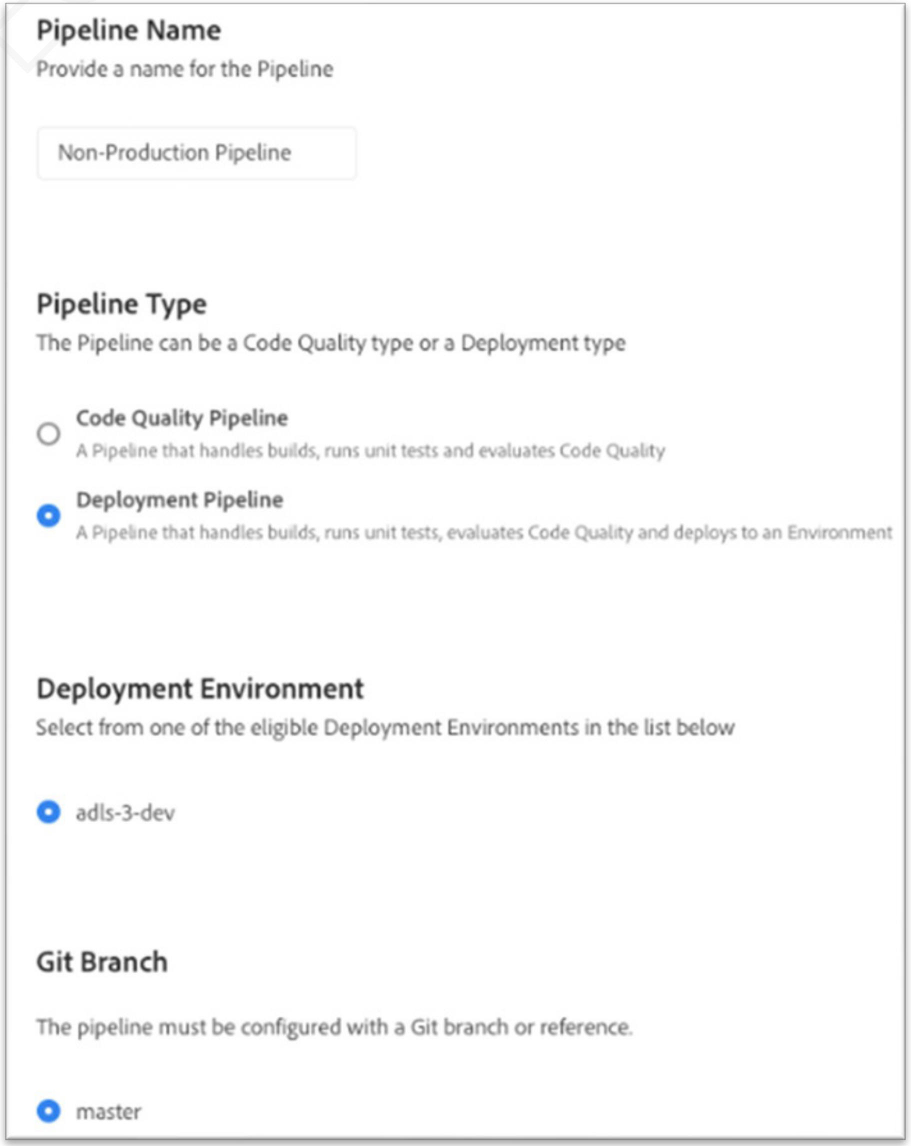
Pipelines | Non-Production Setup

0 to many pipelines

- Types: Code Quality or Deployment
- Deployment Env (not needed for Code Quality)
- Git Branch (code to deploy)
- Pipeline Options (trigger and failure behavior)

Non-Production types

- Code quality: Checks code quality
- Deployment: Checks code quality and deploys to a dev environment



Pipeline Name

Provide a name for the Pipeline

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

Deployment Environment

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

● adls-3-dev

Git Branch

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

● master

Pipelines | Production Setup

Can only have 0 to 1 production pipelines

Source Code

- Select a git branch

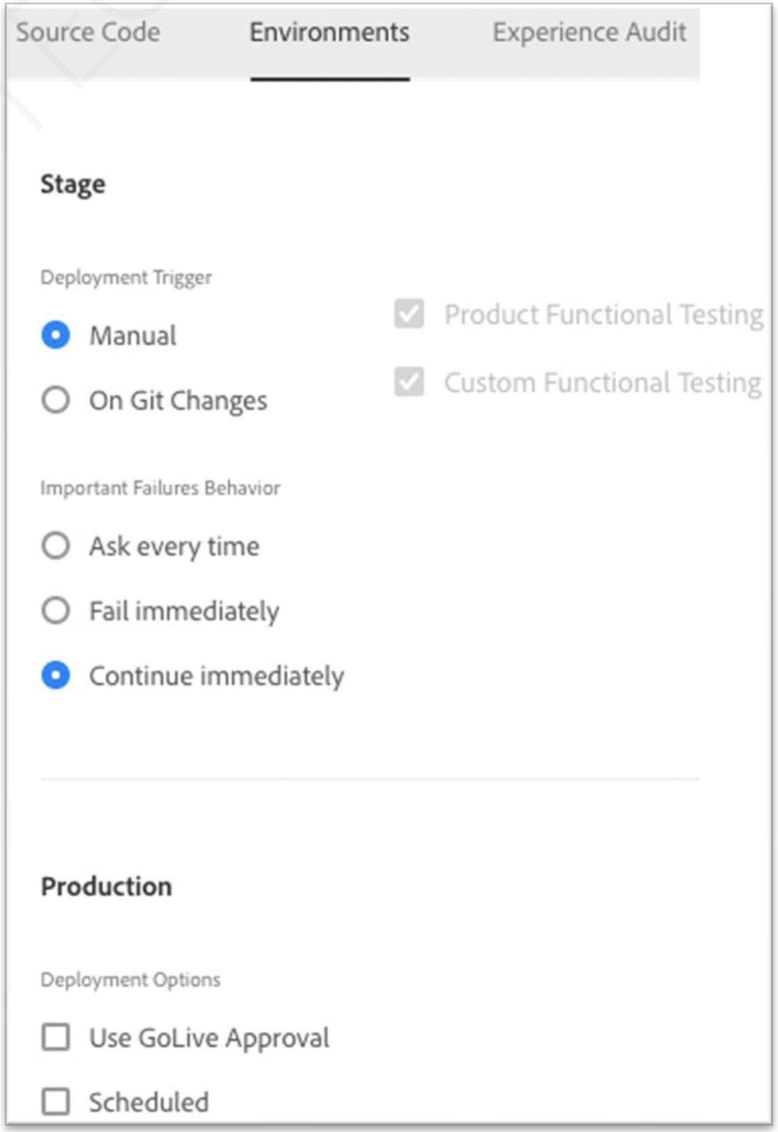
Environments – Stage Options

- Trigger
- failure behavior
- testing decisions

Environments - Production Options

- GoLive Approval
- Scheduled
- CSE Oversight (Managed Services only)

Dispatcher invalidation (Managed Services only)



Stage

Deployment Trigger

● Manual

○ On Git Changes

☑ Product Functional Testing

☑ Custom Functional Testing

Important Failures Behavior

○ Ask every time

○ Fail immediately

● Continue immediately

Production

Deployment Options

□ Use GoLive Approval

□ Scheduled

Pipelines | Stages

Dev and Stage Deployment

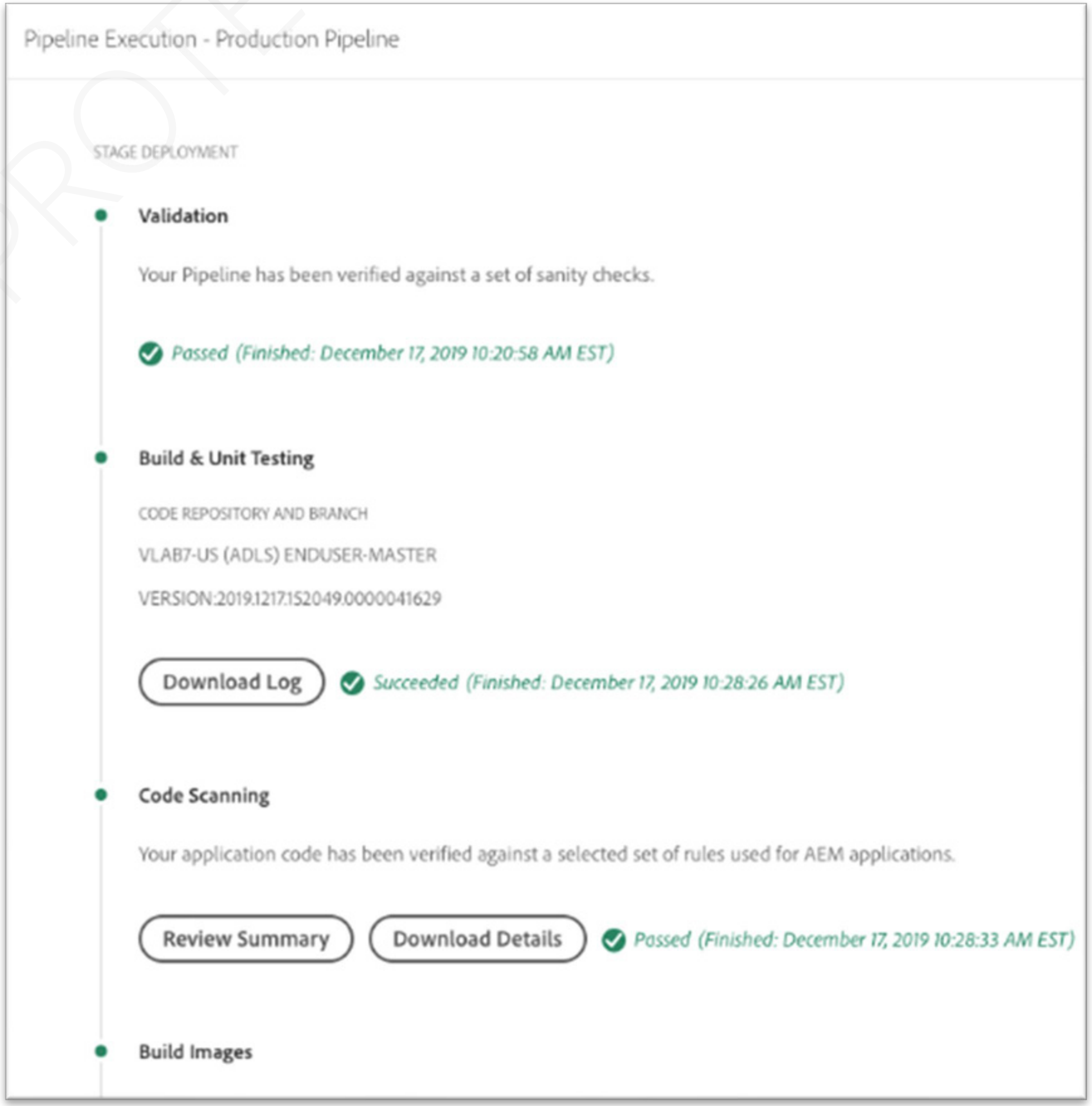
1. Validation (sanity checks)
2. Build & Unit testing
3. Code Scanning
4. Build Images
5. Deploy Environment

Testing (Stage only)

1. Product Functional Testing
2. Custom Functional Testing
3. Custom UI Testing
4. Experience Audit Testing

Prod Deployment (Prod only)

1. Deploy to production



Pipeline Execution - Production Pipeline

STAGE DEPLOYMENT

Validation

Your Pipeline has been verified against a set of sanity checks.

✓ *Passed (Finished: December 17, 2019 10:20:58 AM EST)*

Build & Unit Testing

CODE REPOSITORY AND BRANCH

VLAB7-US (ADLS) ENDUSER-MASTER

VERSION:2019.1217.152049.0000041629

✓ *Succeeded (Finished: December 17, 2019 10:28:26 AM EST)*

Code Scanning

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

✓ *Passed (Finished: December 17, 2019 10:28:33 AM EST)*

Build Images

Testing in Pipelines

All pipelines

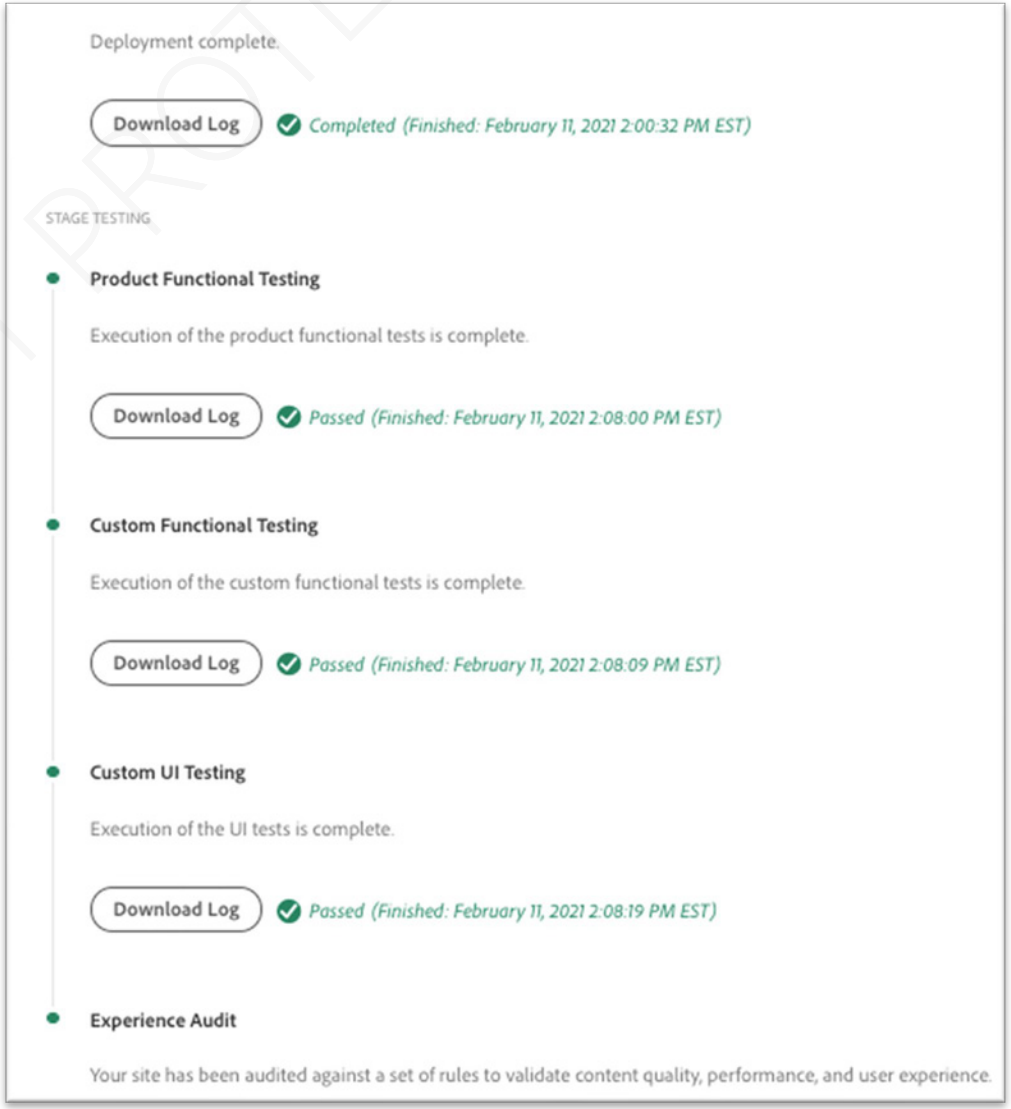
1. Unit Testing
2. Code Quality Testing

Production Pipelines only

1. Functional Testing
2. UI Testing
3. Experience Audit Testing

Tests can be

1. Customers written
2. Adobe written
3. Open-source tools



Deployment complete.

Download Log ✓ *Completed (Finished: February 11, 2021 2:00:32 PM EST)*

STAGE TESTING

● **Product Functional Testing**

Execution of the product functional tests is complete.

Download Log ✓ *Passed (Finished: February 11, 2021 2:08:00 PM EST)*

● **Custom Functional Testing**

Execution of the custom functional tests is complete.

Download Log ✓ *Passed (Finished: February 11, 2021 2:08:09 PM EST)*

● **Custom UI Testing**

Execution of the UI tests is complete.

Download Log ✓ *Passed (Finished: February 11, 2021 2:08:19 PM EST)*

● **Experience Audit**

Your site has been audited against a set of rules to validate content quality, performance, and user experience.

Pipelines: All

Testing | Unit

Unit testing is run through Maven

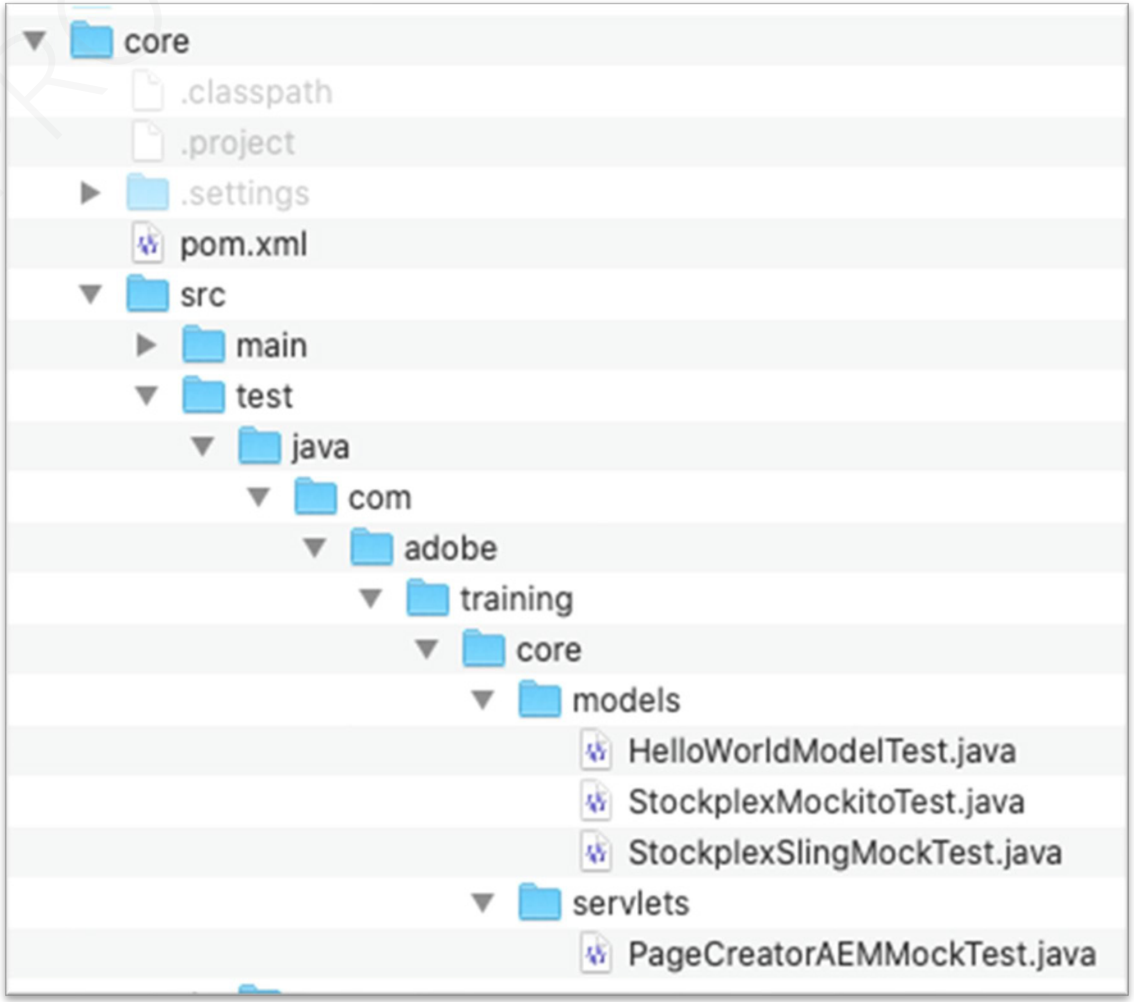
Common mocking frameworks

- Mockito
- Sling Mocks
- AEM Mocks

Unit tests are always run as apart of the Maven build process

Unit tests can be explicitly run locally by using Maven

- `$ mvn test`



```
▼ core
      .classpath
      .project
  ▶ .settings
      pom.xml
  ▼ src
    ▶ main
    ▼ test
      ▼ java
        ▼ com
          ▼ adobe
            ▼ training
              ▼ core
                ▼ models
                    HelloWorldModelTest.java
                    StockplexMockitoTest.java
                    StockplexSlingMockTest.java
                ▼ servlets
                    PageCreatorAEMMockTest.java
```

Pipelines: All

Testing | Code Quality

Analyzed using SonarQube rules

Issues are grouped into 3 categories

- Code Smell (Maintainability domain)
- Bug (Reliability domain)
- Vulnerability (Security domain)

Rules executed:

- **Squid** - Standard SonarQube Java rules
- Sonar Plugins
  - o **AEM** – AEM Rules (By Cognifide)
  - o **CQ** – Cloud Manager Rules (By Adobe)

Pipelines: All

Testing | Code Quality Results

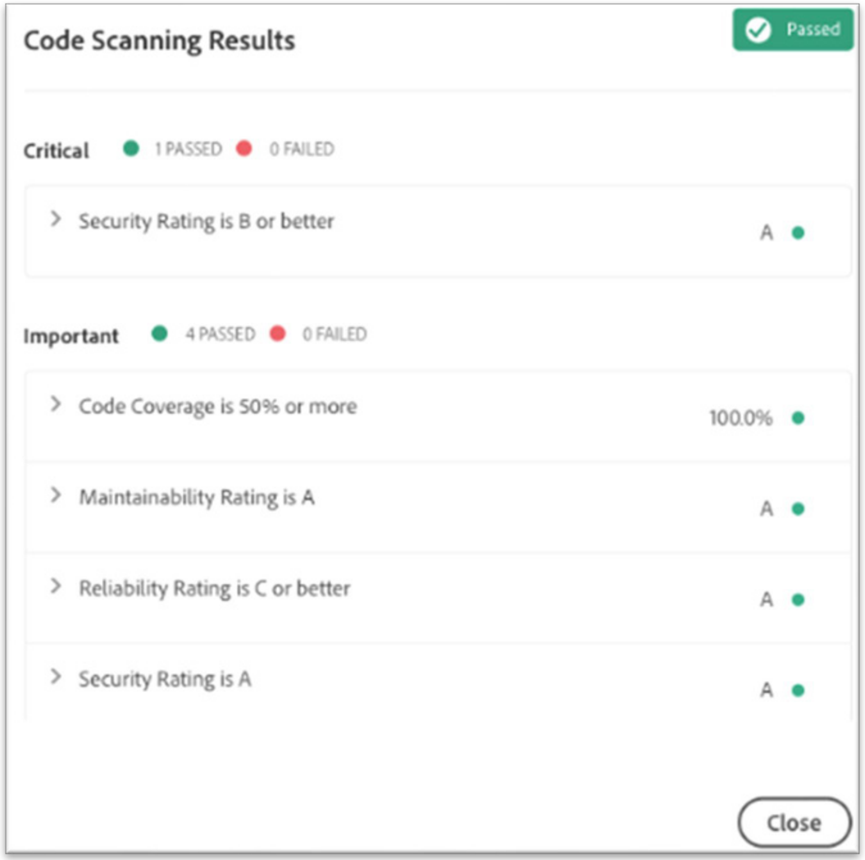
Quality is focused on

- Security Rating
- Reliability
- Maintainability
- Code coverage

Groups are separated into 3 severity levels

- **Critical** – fails the pipeline. Must be fixed to continue
- **Important** – Pauses the pipeline. Can be overridden or ignored
- **Information** – No impact on the pipeline execution

SonarQube produces a CSV report of all issues



Code Scanning Results

Critical ● 1 PASSED ● 0 FAILED

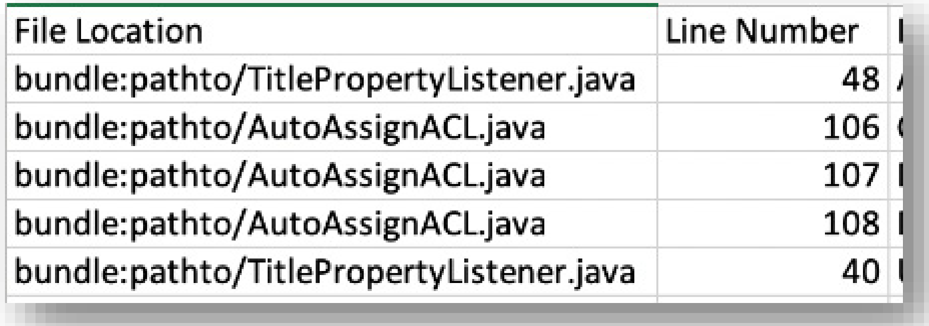
› Security Rating is B or better	A ●

Important ● 4 PASSED ● 0 FAILED

› Code Coverage is 50% or more	100.0% ●
› Maintainability Rating is A	A ●
› Reliability Rating is C or better	A ●
› Security Rating is A	A ●

Pipelines: All

Example CSV output

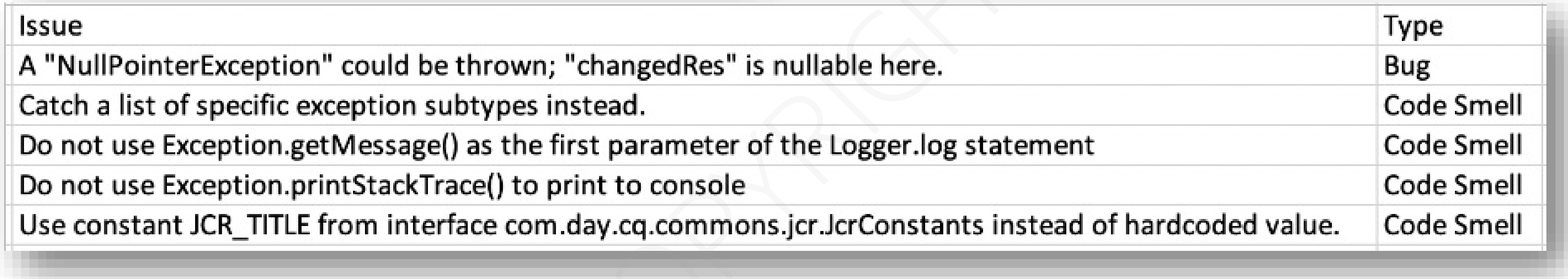


File Location	Line Number
bundle:pathto/TitlePropertyListener.java	48
bundle:pathto/AutoAssignACL.java	106
bundle:pathto/AutoAssignACL.java	107
bundle:pathto/AutoAssignACL.java	108
bundle:pathto/TitlePropertyListener.java	40

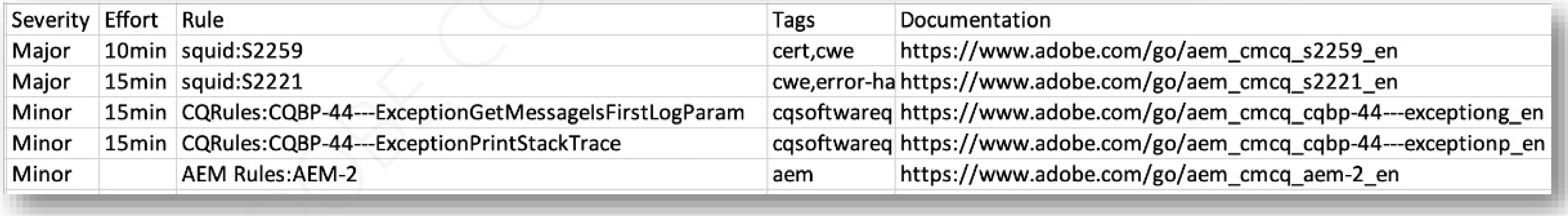
Exact file, line#, what to fix and how long

Full categorization (Type, Severity, Rule, Tags)

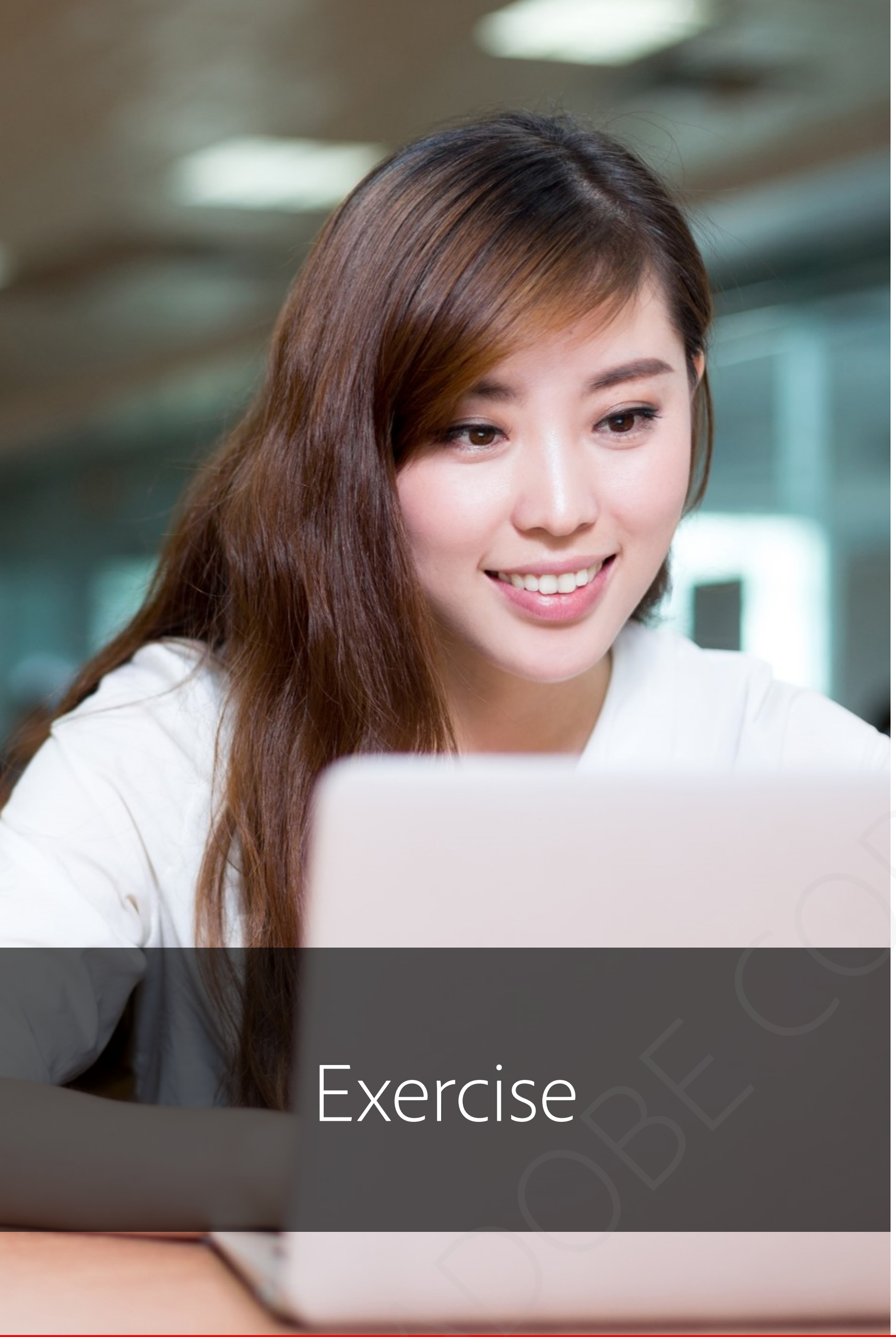
Quick documentation links



Issue	Type
A "NullPointerException" could be thrown; "changedRes" is nullable here.	Bug
Catch a list of specific exception subtypes instead.	Code Smell
Do not use Exception.getMessage() as the first parameter of the Logger.log statement	Code Smell
Do not use Exception.printStackTrace() to print to console	Code Smell
Use constant JCR_TITLE from interface com.day.cq.commons.jcr.JcrConstants instead of hardcoded value.	Code Smell



Severity	Effort	Rule	Tags	Documentation
Major	10min	squid:S2259	cert,cwe	https://www.adobe.com/go/aem_cmcq_s2259_en
Major	15min	squid:S2221	cwe,error-ha	https://www.adobe.com/go/aem_cmcq_s2221_en
Minor	15min	CQRules:CQBP-44---ExceptionGetMessageIsFirstLogParam	cqsoftwareq	https://www.adobe.com/go/aem_cmcq_cqbp-44---exceptiong_en
Minor	15min	CQRules:CQBP-44---ExceptionPrintStackTrace	cqsoftwareq	https://www.adobe.com/go/aem_cmcq_cqbp-44---exceptionp_en
Minor		AEM Rules:AEM-2	aem	https://www.adobe.com/go/aem_cmcq_aem-2_en



Exercise 1: Observe the custom code quality rules

In this exercise, you will observe the custom code quality rules output for a non-compliant Java file.

You will then observe the code quality rules created for code, which is fully compliant with the code rules.

Pipeline: Prod

Testing | Functional and UI

The AEM Archetype can produce maven modules for Functional and UI tests

Product Functional Testing (by Adobe)

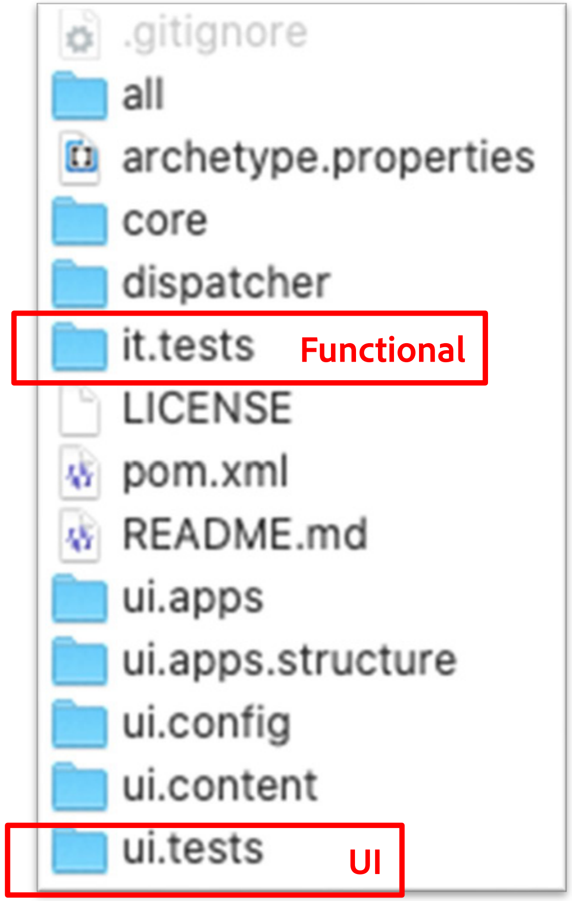
- Set of stable integration tests for AEM. Cannot be skipped

Custom Functional Testing (by Customer)

- Separate testing jar build using a maven module
- Auto-passes if no testing jar is present

UI Testing (By Customer)

- Selenium-based tests packaged into a docker image
- Tests can be written with Java, Node or other frameworks



.gitignore
all
archetype.properties
core
dispatcher
it.tests Functional
LICENSE
pom.xml
README.md
ui.apps
ui.apps.structure
ui.config
ui.content
ui.tests UI

Pipeline: Prod

Testing | Experience Audit

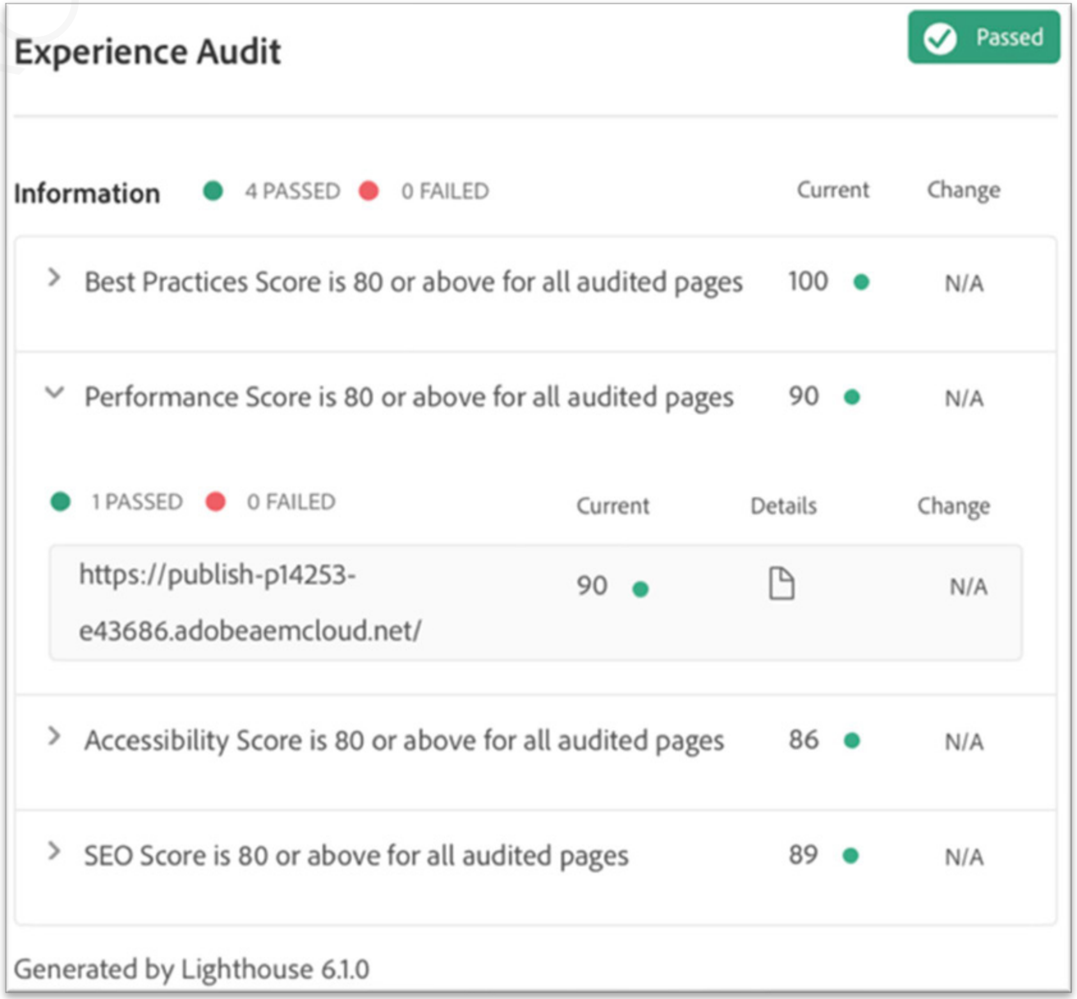
Powered by Google Lighthouse that audits baseline standards for performance, accessibility, best practices and SEO

Results Produced:

- Aggregate level metrics for average page score
- Page specific level metrics available
- Scores of individual test results
- Result history informs website improvement

Aggregate scores explain

- The page(s) have improved on the selected test
- The page(s) have regressed on the selected test
- No Change
- No previous score available to compare to



Experience Audit ✓ Passed

Information ● 4 PASSED ● 0 FAILED	Current	Change
› Best Practices Score is 80 or above for all audited pages	100 ●	N/A
˅ Performance Score is 80 or above for all audited pages	90 ●	N/A

● 1 PASSED ● 0 FAILED	Current	Details	Change
https://publish-p14253-e43686.adobeaemcloud.net/	90 ●	🗋	N/A

	Current	Change
› Accessibility Score is 80 or above for all audited pages	86 ●	N/A
› SEO Score is 80 or above for all audited pages	89 ●	N/A

Generated by Lighthouse 6.1.0

Adobe I/O

Adobe I/O is a single-entry point for communication with all Adobe solutions

- Easy, secure, and fast way to access Adobe's APIs
- Can be accessed via UI or command line
- Provides events and an API gateway

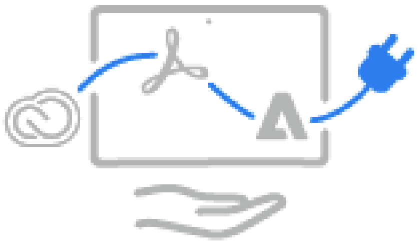
AEM uses Adobe I/O to connect to Sensei, Launch, Target, Analytics, etc

All solutions connect to AEM with 2 primary steps:

- IMS configuration through Adobe IO
- Cloud Configuration in AEM for the solution

Integrated

Serverless



Flexible

Adobe I/O | Cloud Manager

Adobe I/O provides a command line tool called aio-cli

Cloud Manager offers a plugin for automation purposes

- npm install @adobe/aio-cli-plugin-cloudmanager

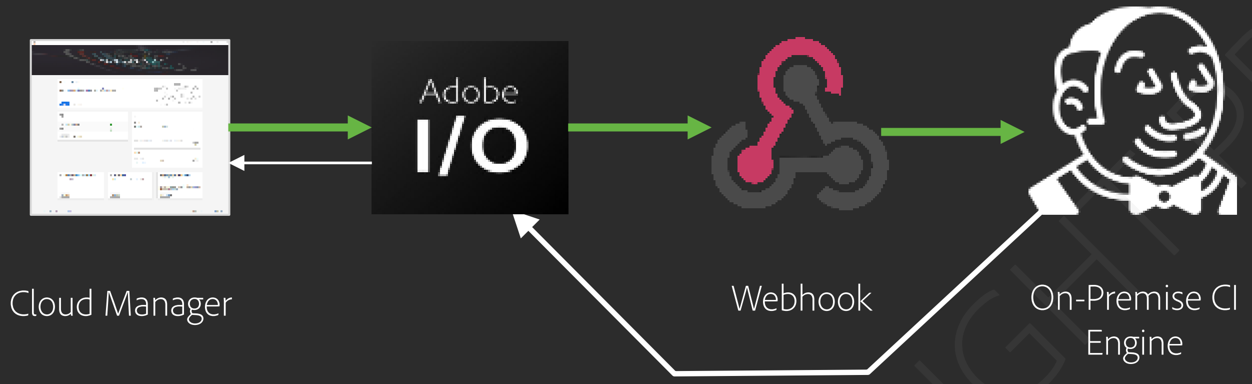
Common commands available

- List programs, environments, pipelines
- Check pipeline status
- Create/Delete/Advance a pipeline
- Developer console and log access
- Set environment variables

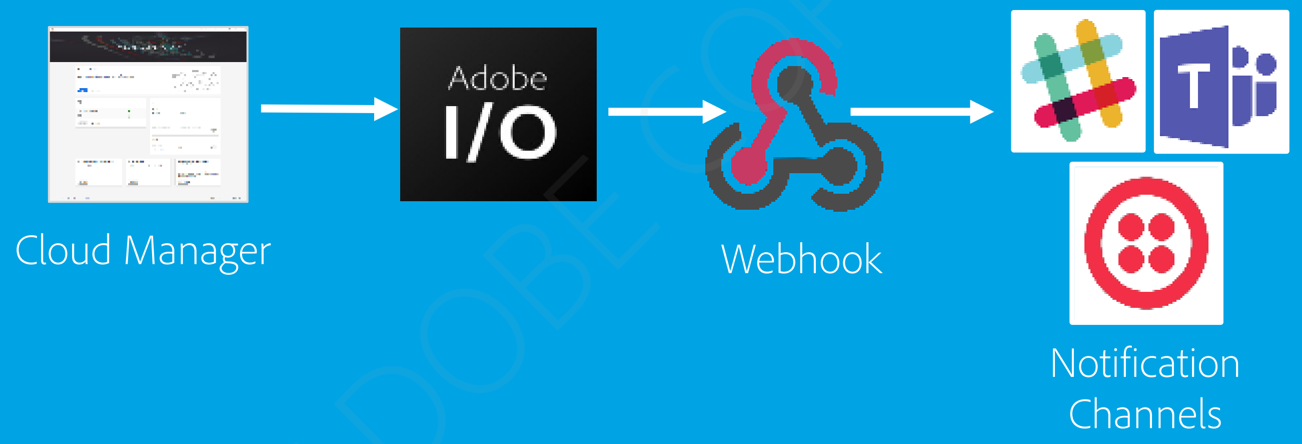
adobe / aio-cli-plugin-cloudmanager

Adobe IO | Endless Possibilities

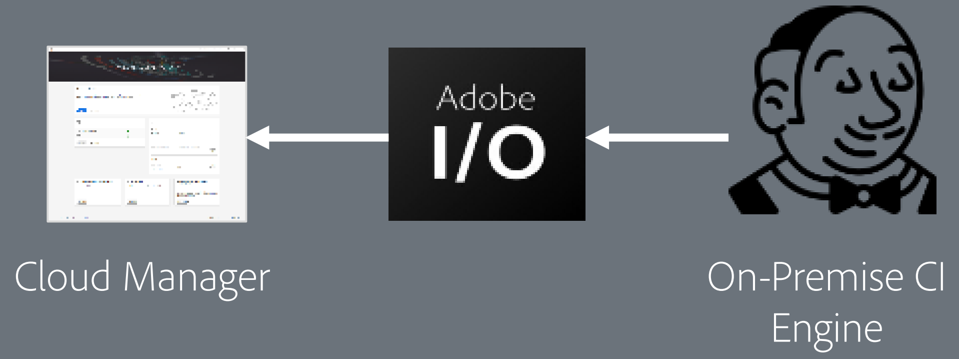
Run external tests



Events notifying external systems

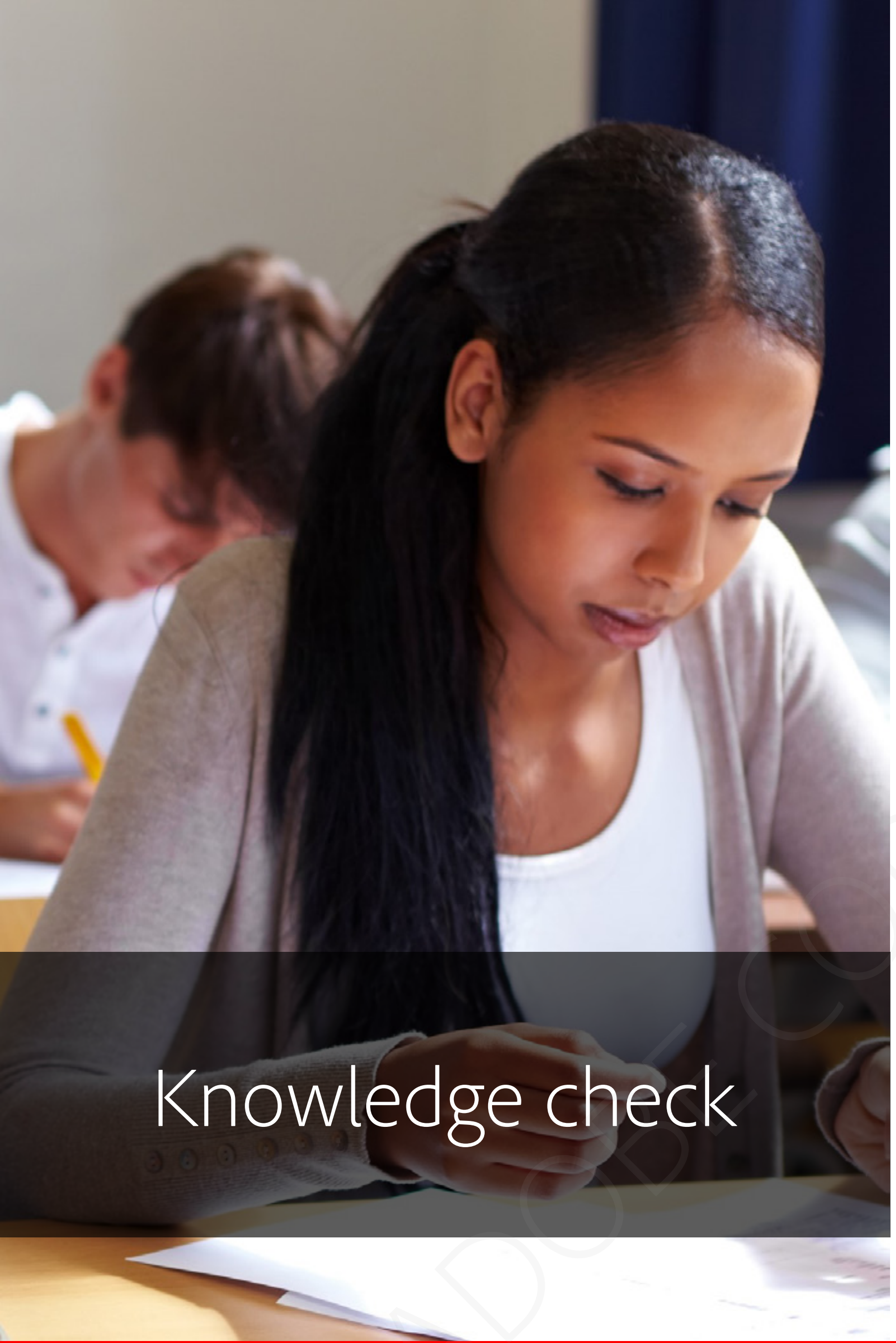


Start Pipeline



Key takeaways

- Cloud Manager enables organizations to self-manage Experience Manager in the cloud
- Key Features
  - Self Service Interface
  - CI/CD Pipeline
  - Flexible Deployments
  - Autoscaling
- Continuous Integration Pipeline: Promote code changes and eliminate manual processes remotely



Knowledge check

Which of the following are part of Cloud Manager Environment? Select the three correct answers.

A. Author Instance ✓

B. LDAP

C. Publish Instance ✓

D. Dispatcher ✓