DevOps for AEM as a Cloud Service

Getting Started with a Maven Project in Git

Adobe

Student Workbook

Find your journey at learning.adobe.com >

Contents

DevOps for AEM as a Cloud Service

10-14-2020

Introduction

When multiple software developers working parallel in a project, you need to ensure that there are no code conflicts. You also need to ensure data integrity and support for distributed, non- linear workflows. Git is a distributed Source Code Management(SCM) for tracking changes in source code during software development. It is designed for coordinating work among programmers and track changes in any set of files.

Adobe Cloud Manager comes provisioned with a single git repository that is used to deploy code using Cloud Manager's CI/CD pipelines. Customers can use the Cloud Manager's Git repository out of the box. Customers also have the option of integrating an on-premise or customer-managed git repository with Cloud Manager.

Objectives

After completing this module, you will be able to:

- Describe the Git repository
- View the branches in the Git repository
- Checkout a branch
- Commit changes to a branch
- Install a mvn project

Git Repository

Git is a program that tracks changes made to files. Once installed, Git can be initialized on a project to create a Git repository. The purpose of Git is to manage a project or set of files as they change over time.

A Git repository is the .git/ folder inside a project. The repository tracks all changes made to the files in your project, building a history over time. There are two types of GIT repositories, local and remote. You can use the GIT local repository on your computer to make changes, and Git remote repository is a common repository that all team members use to exchange their changes.

Git Commands

Below is the list of some basic Git commands:

1. **git init**: Create a new local repository
2. **git clone /path/to/repository** : Check out a repository
3. **git add** : Add files
4. **git add *** : Add one or more files to staging
5. **git commit -m "Commit message"** : Commit changes to the repository
6. **git commit -a**: Commit any files you have added with git add, and also commit any files you have changed.
7. **git push origin master**: Send changes to the master branch of your remote repository
8. **git status** : List which files are staged, unstaged, and untracked
9. **git remote add origin** : Connect to a remote repository
10. **git branch**: Lists all the local branches in the current repository
11. **git checkout** : Switch from one branch to another

Note: For more details on Git commands, [1].

Exercise 1: Work with Git repository

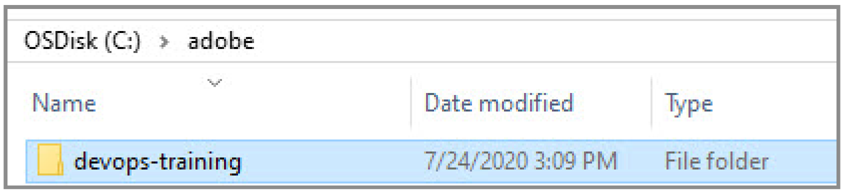
Scenario: In this exercise, you will view the branches in your repository, checkout the starter-project branch, create and fetch the dev branch.

In a real project, you will need the clone to company repository to your local environment in order to develop any hotfixes, upgrades, or features. In this class, the Git project is provided to you in the **Exercise_Files** and there is no need to clone the repository locally.

Prerequisites:

- Git command line is installed
- The devops-training project (provided by your instructor)

1. Extract the **Exercise_Files.zip** containing the **devops-training** project folder. If you are using **ReadyTech**, place this folder in **C:/adobe/**. If you are developing locally, you can use a similar location, as shown:



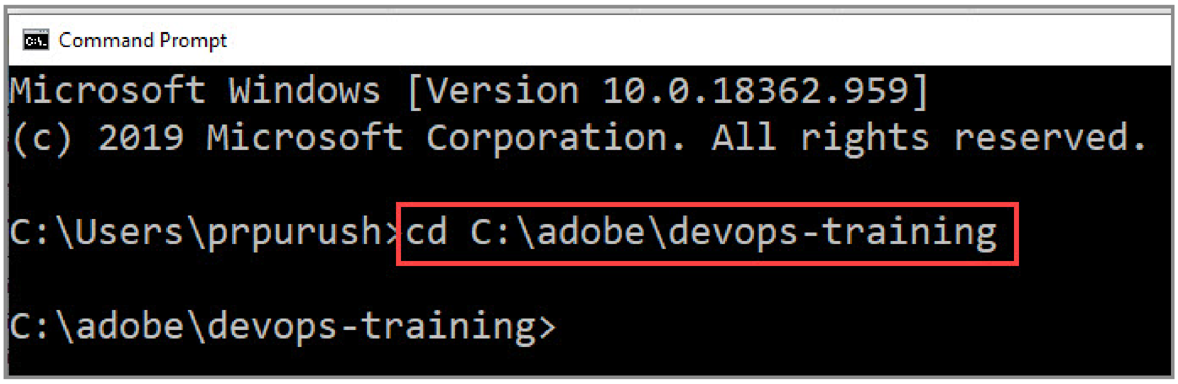
OSDisk (C:) › adobe

Name	Date modified	Type
devops-training	7/24/2020 3:09 PM	File folder

Note: To learn more about cloning a repository you can look at the online documents.[2]

2. Open a **Command Prompt** window and navigate to your project location, as shown:

```
cd c:\adobe\devops-training
```

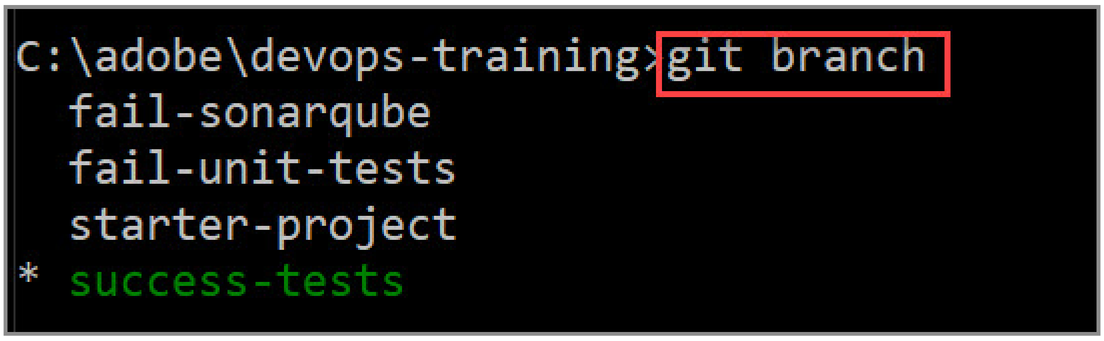


```
Microsoft Windows [Version 10.0.18362.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\prpurush>cd C:\adobe\devops-training

C:\adobe\devops-training>
```

Note: Your project location might be different.

3. View the branches by executing the following command:

```
git branch
```

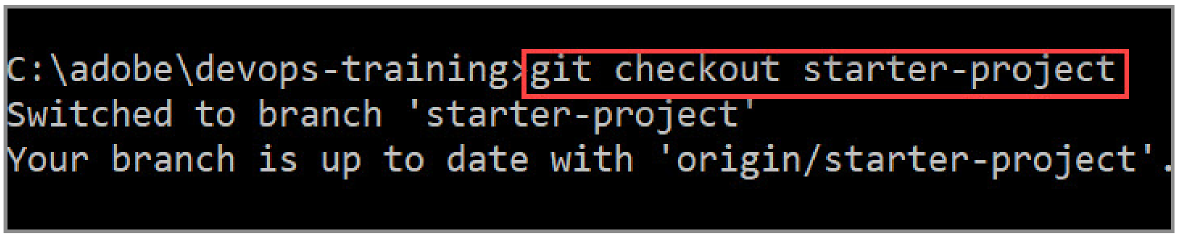


```
C:\adobe\devops-training>git branch
  fail-sonarqube
  fail-unit-tests
  starter-project
* success-tests
```

The branches are listed.

4. If you are not in the **starter-project** branch, switch to the **starter-project** branch by executing the following command:

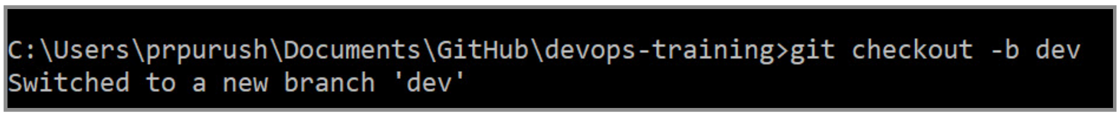
```
git checkout starter-project
```



```
C:\adobe\devops-training>git checkout starter-project
Switched to branch 'starter-project'
Your branch is up to date with 'origin/starter-project'.
```

5. To start development, create a new **dev** branch based on the **starter-project** branch by executing the following command:

```
git checkout -b dev
```

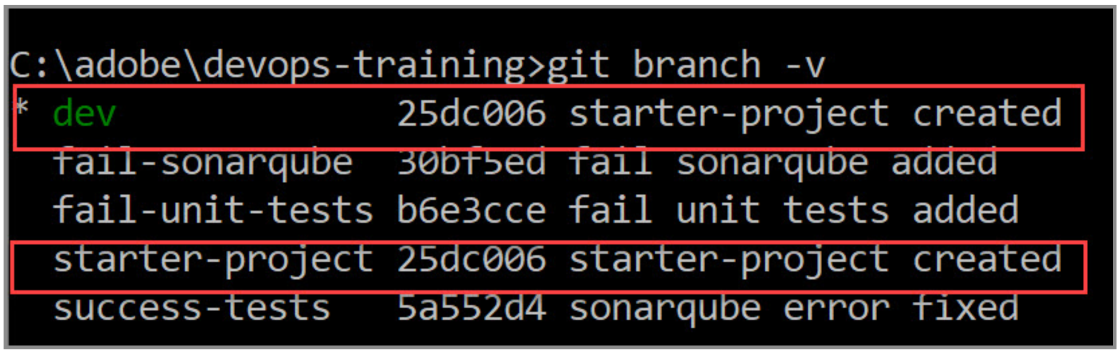


```
C:\Users\prpurush\Documents\GitHub\devops-training>git checkout -b dev
Switched to a new branch 'dev'
```

Your new branch **dev** is created.

6. To verify you have created a **dev** branch based on the **starter-project**, execute the following command:

```
git branch -v
```



```
C:\adobe\devops-training>git branch -v
* dev             25dc006 starter-project created
  fail-sonarqube  30bf5ed fail sonarqube added
  fail-unit-tests b6e3cce fail unit tests added
  starter-project 25dc006 starter-project created
  success-tests   5a552d4 sonarqube error fixed
```

The branches are listed.

Note: Notice the last commit is the same for dev and starter-project.

Exercise 2: Install the DevOps training project

Scenario: When starting a new AEM project, you will typically start from the latest AEM archetype and initialize a git repository. If you are starting to develop from an existing project, you will need to check out your code from your company repository. Once your git project is local, you can make local changes and push them into your company repository. Because AEM has new builds daily, it is a good practice to refresh the build number of the SDK in your POM files to match the build of the local SDK.

In this exercise you will make changes to your project, install your project into AEM, and commit your changes to git.

Prerequisites:

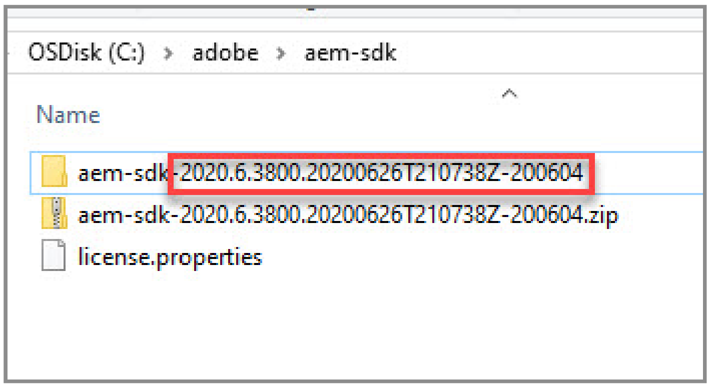
- Maven installed
- AEM running on port 4502

This exercise includes the following tasks:

1. Update the POM file
2. Install the project into AEM
3. Perform your first commit

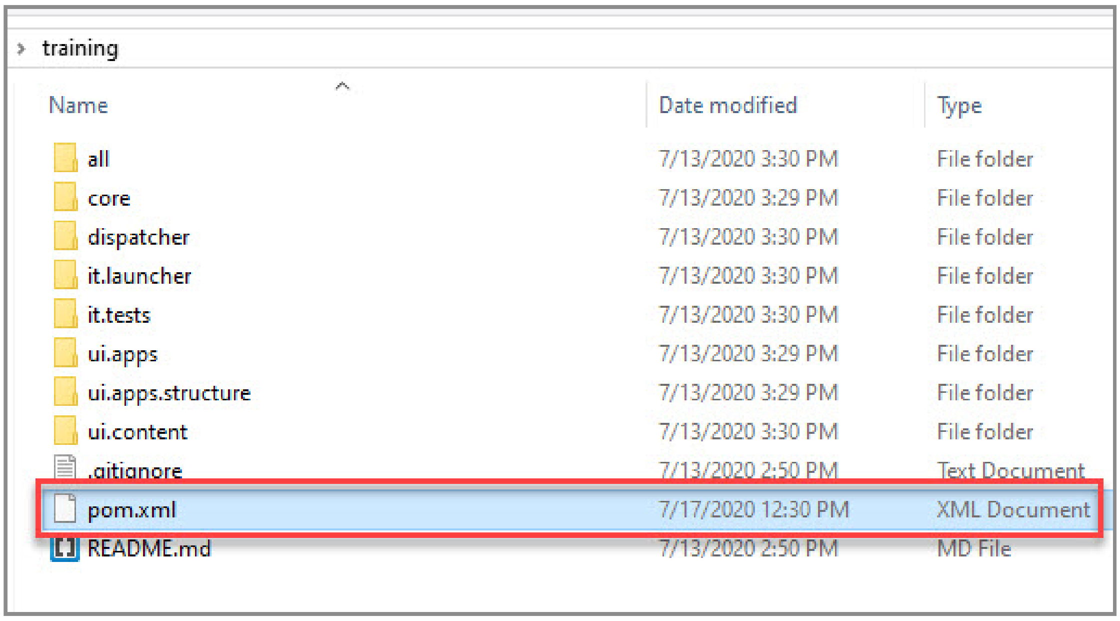
Task 1: Update the POM file

1. Navigate to the location of the AEM SDK quickstart jar. In Readytech navigate to **C:/adobe/aem-sdk/**.
2. Notice the SDK zip contains the build number: aem-sdk-< buildnumber >.
3. Copy the build number, as shown:



```
aem-sdk-2020.6.3800.20200626T210738Z-200604
aem-sdk-2020.6.3800.20200626T210738Z-200604.zip
license.properties
```

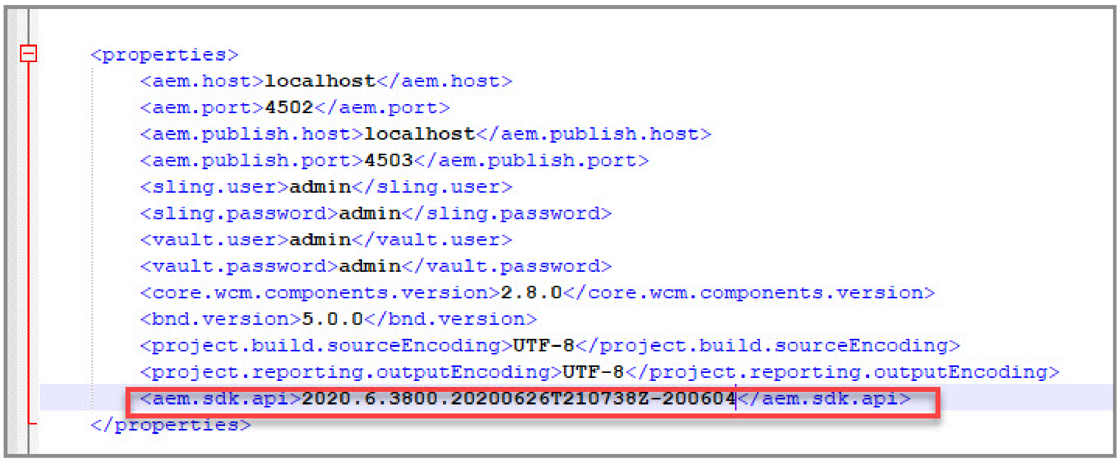
4. Navigate to your newly created Maven project training, as shown:



training

Name	Date modified	Type
all	7/13/2020 3:30 PM	File folder
core	7/13/2020 3:29 PM	File folder
dispatcher	7/13/2020 3:30 PM	File folder
it.launcher	7/13/2020 3:30 PM	File folder
it.tests	7/13/2020 3:30 PM	File folder
ui.apps	7/13/2020 3:29 PM	File folder
ui.apps.structure	7/13/2020 3:29 PM	File folder
ui.content	7/13/2020 3:30 PM	File folder
.gitignore	7/13/2020 2:50 PM	Text Document
pom.xml	7/17/2020 12:30 PM	XML Document
README.md	7/13/2020 2:50 PM	MD File

5. Open the POM file using a text editor (such as Notepad++) and paste the build number that you copied from the SDK zip file in step 3, as shown:



```
<properties>
    <aem.host>localhost</aem.host>
    <aem.port>4502</aem.port>
    <aem.publish.host>localhost</aem.publish.host>
    <aem.publish.port>4503</aem.publish.port>
    <sling.user>admin</sling.user>
    <sling.password>admin</sling.password>
    <vault.user>admin</vault.user>
    <vault.password>admin</vault.password>
    <core.wcm.components.version>2.8.0</core.wcm.components.version>
    <bnd.version>5.0.0</bnd.version>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <aem.sdk.api>2020.6.3800.20200626T210738Z-200604</aem.sdk.api>
</properties>
```

6. Save the file.

Note: To learn more about AEM Archetype, look at the AEM Archetype documentation , [3].

Task 2: Install the mvn project

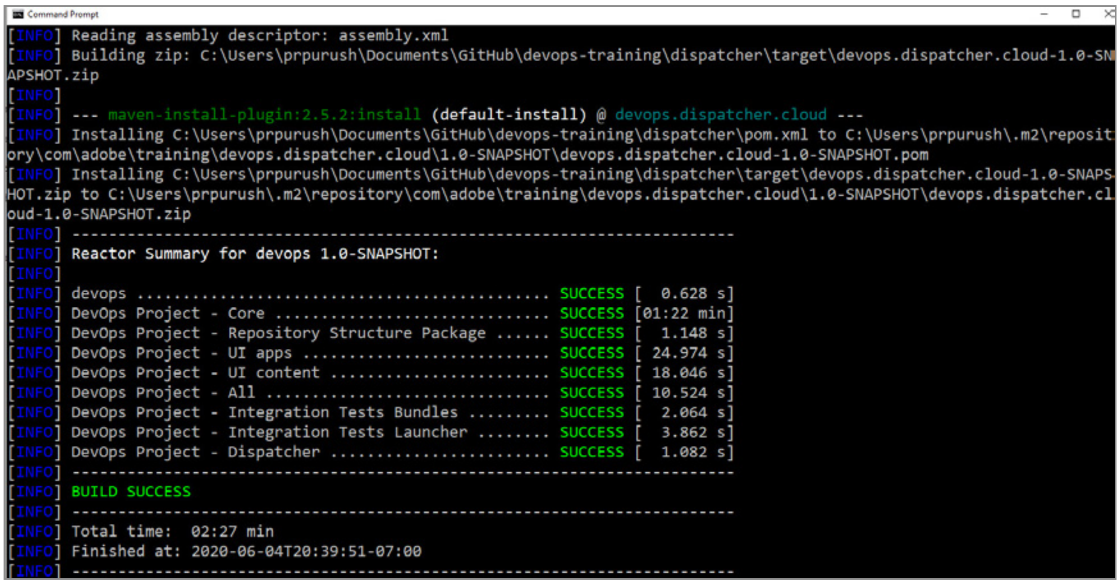
1. Open a **Command Prompt** window and navigate to your repository **devops-training**.

2. Run the following command and press Enter. The project starts to build.

```
mvn clean install -Padobe-public -PautoInstallSinglePackage
```

Note: The project build may take a few minutes.

3. Verify the build is success, as shown:



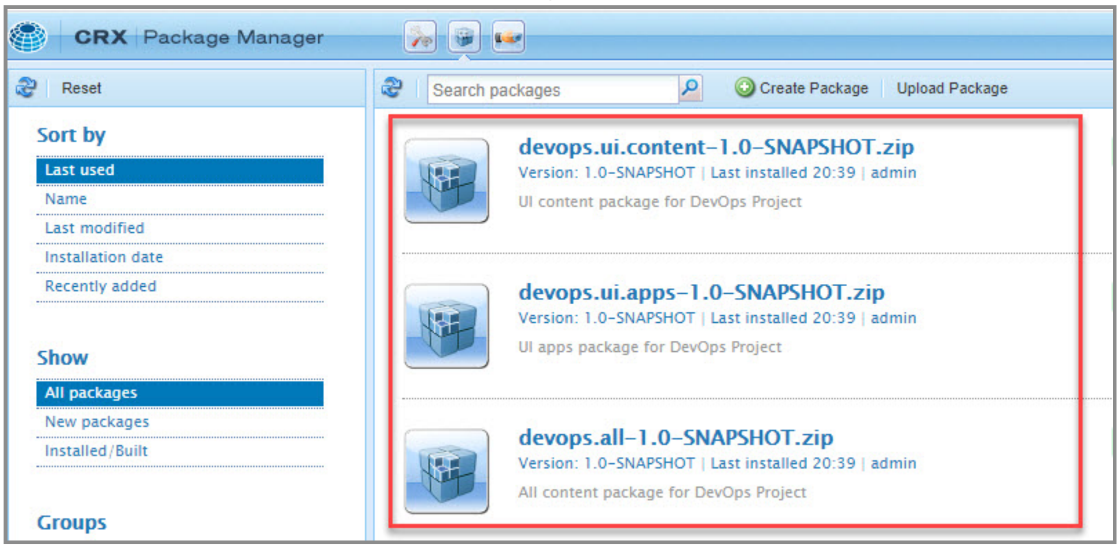
```
Command Prompt
[INFO] Reading assembly descriptor: assembly.xml
[INFO] Building zip: C:\Users\prpurush\Documents\GitHub\devops-training\dispatcher\target\devops.dispatcher.cloud-1.0-SN
APSHOT.zip
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ devops.dispatcher.cloud ---
[INFO] Installing C:\Users\prpurush\Documents\GitHub\devops-training\dispatcher\pom.xml to C:\Users\prpurush\.m2\reposit
ory\com\adobe\training\devops.dispatcher.cloud\1.0-SNAPSHOT\devops.dispatcher.cloud-1.0-SNAPSHOT.pom
[INFO] Installing C:\Users\prpurush\Documents\GitHub\devops-training\dispatcher\target\devops.dispatcher.cloud-1.0-SNAPS
HOT.zip to C:\Users\prpurush\.m2\repository\com\adobe\training\devops.dispatcher.cloud\1.0-SNAPSHOT\devops.dispatcher.cl
oud-1.0-SNAPSHOT.zip
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary for devops 1.0-SNAPSHOT:
[INFO]
[INFO] devops ............................................. SUCCESS [  0.628 s]
[INFO] DevOps Project - Core .............................. SUCCESS [01:22 min]
[INFO] DevOps Project - Repository Structure Package ...... SUCCESS [  1.148 s]
[INFO] DevOps Project - UI apps ........................... SUCCESS [ 24.974 s]
[INFO] DevOps Project - UI content ........................ SUCCESS [ 18.046 s]
[INFO] DevOps Project - All ............................... SUCCESS [ 10.524 s]
[INFO] DevOps Project - Integration Tests Bundles ......... SUCCESS [  2.064 s]
[INFO] DevOps Project - Integration Tests Launcher ........ SUCCESS [  3.862 s]
[INFO] DevOps Project - Dispatcher ........................ SUCCESS [  1.082 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  02:27 min
[INFO] Finished at: 2020-06-04T20:39:51-07:00
[INFO] ------------------------------------------------------------------------
```

A Success message appears.

4. Verify you are logged on to your AEM author service on port 4502 (http://localhost:4502).

5. Navigate to **Tools** > **Deployment** > **Packages** in your AEM author service. The **Package Manager** opens.

6. Verify the three newly installed devops packages, as shown:



devops.ui.content-1.0-SNAPSHOT.zip
Version: 1.0-SNAPSHOT | Last installed 20:39 | admin
UI content package for DevOps Project

devops.ui.apps-1.0-SNAPSHOT.zip
Version: 1.0-SNAPSHOT | Last installed 20:39 | admin
UI apps package for DevOps Project

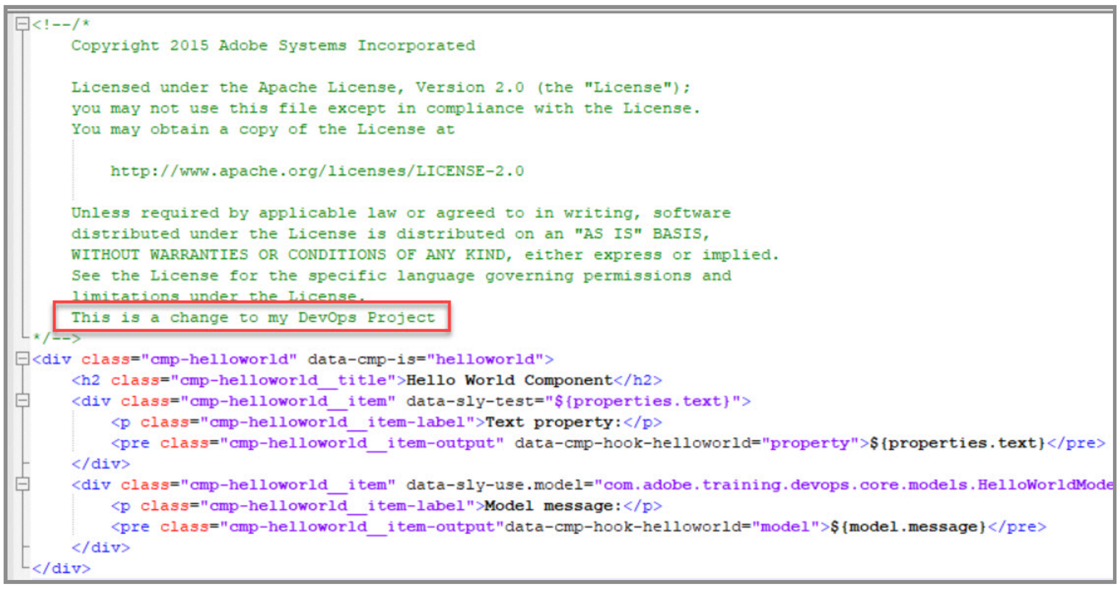
devops.all-1.0-SNAPSHOT.zip
Version: 1.0-SNAPSHOT | Last installed 20:39 | admin
All content package for DevOps Project

Note: To learn more about Development Tools, [4].

Task 3: Perform your first commit

1. In your File Explorer, navigate to **C:\adobe\devops-training\ui.apps\src\main\content\jcr_root\apps\training\components\helloworld** on your local desktop.

2. Open the file **helloworld.html** using a text editor and add a comment to the file, as shown:

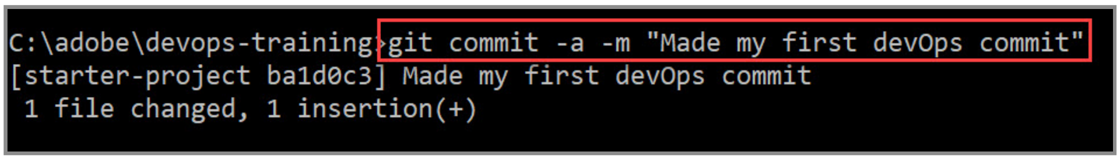


```
<!--/*
    Copyright 2015 Adobe Systems Incorporated

    Licensed under the Apache License, Version 2.0 (the "License");
    you may not use this file except in compliance with the License.
    You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
    This is a change to my DevOps Project
*/-->
<div class="cmp-helloworld" data-cmp-is="helloworld">
    <h2 class="cmp-helloworld__title">Hello World Component</h2>
    <div class="cmp-helloworld__item" data-sly-test="${properties.text}">
        <p class="cmp-helloworld__item-label">Text property:</p>
        <pre class="cmp-helloworld__item-output" data-cmp-hook-helloworld="property">${properties.text}</pre>
    </div>
    <div class="cmp-helloworld__item" data-sly-use.model="com.adobe.training.devops.core.models.HelloWorldMode
        <p class="cmp-helloworld__item-label">Model message:</p>
        <pre class="cmp-helloworld__item-output"data-cmp-hook-helloworld="model">${model.message}</pre>
    </div>
</div>
```

3. Save the changes (**File**>**Save**).

4. Commit the changes into your dev branch by executing the following command:

```
git commit -a -m “Made my first devOps commit”
```

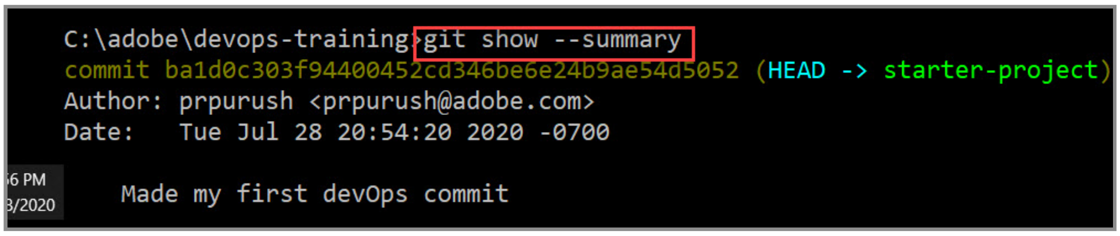


```
C:\adobe\devops-training>git commit -a -m "Made my first devOps commit"
[starter-project ba1d0c3] Made my first devOps commit
 1 file changed, 1 insertion(+)
```

Note: The above command will commit all the files that have been changed.

5. View the details of your commit by executing the following command:

```
git show --summary
```



```
C:\adobe\devops-training>git show --summary
commit ba1d0c303f94400452cd346be6e24b9ae54d5052 (HEAD -> starter-project)
Author: prpurush <prpurush@adobe.com>
Date:   Tue Jul 28 20:54:20 2020 -0700

    Made my first devOps commit
```

The summary is listed.

References

1. Git Commands: https://guides.github.com/introduction/git-handbook/ ↩

2. Git Clone: https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository ↩

3. AEM Archetype: https://docs.adobe.com/content/help/en/experience-manager-core-components/using/developing/archetype/overview.html ↩

4. Development Tools for AEM Projects: https://docs.adobe.com/content/help/en/experience-manager-learn/cloud-service/local-development-environment-set-up/development-tools.html ↩