

## 1.What is DAM and why we use it?

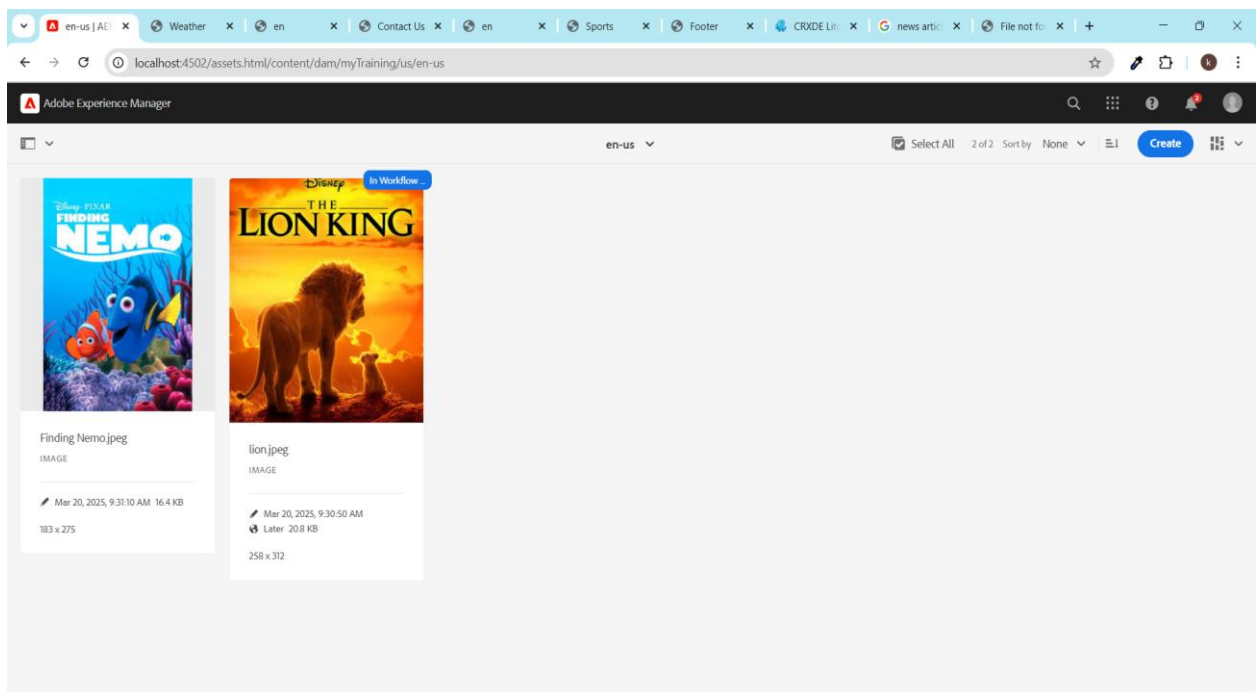
**DAM** (Digital Asset Management) is a system that helps to store, organize, and manage digital assets like images, videos, and documents. In AEM, DAM is available under /content/dam for easy asset management.

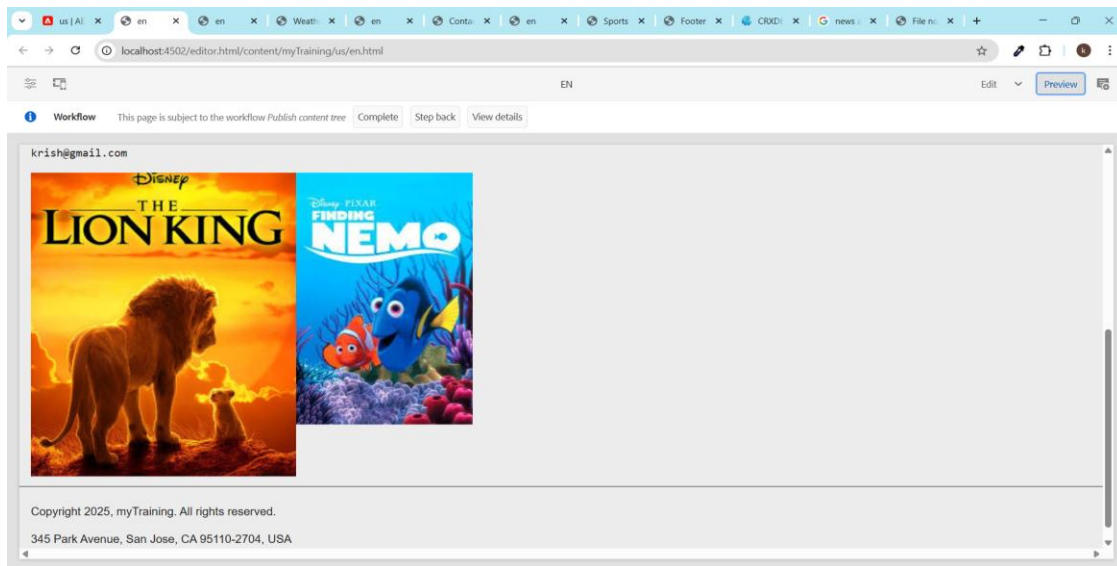
### Why Use DAM?

- Centralized Storage – Keeps all digital assets in one place.
- Metadata Management – Adds tags and descriptions for easy search.
- Automatic Renditions – Creates different image sizes and formats.
- Version Control – Tracks changes and restores previous versions.
- Workflow Support – Helps manage approvals and publishing.

## 2. Create one folder inside our project folder and follow the path:

“/content/dam/myTraining/us/en-us” and upload 2 images and author those images on page using image component as we see today.



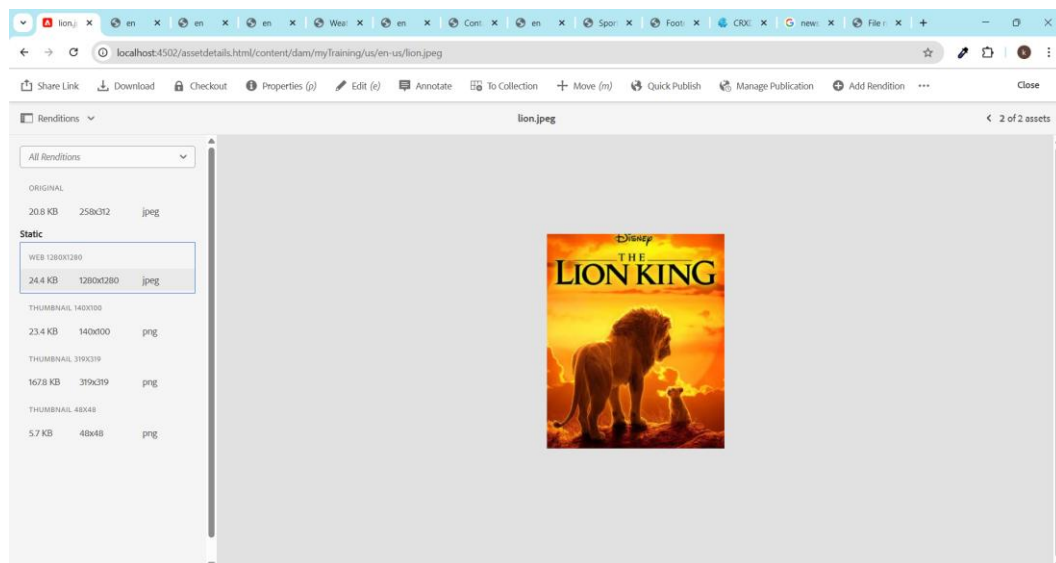


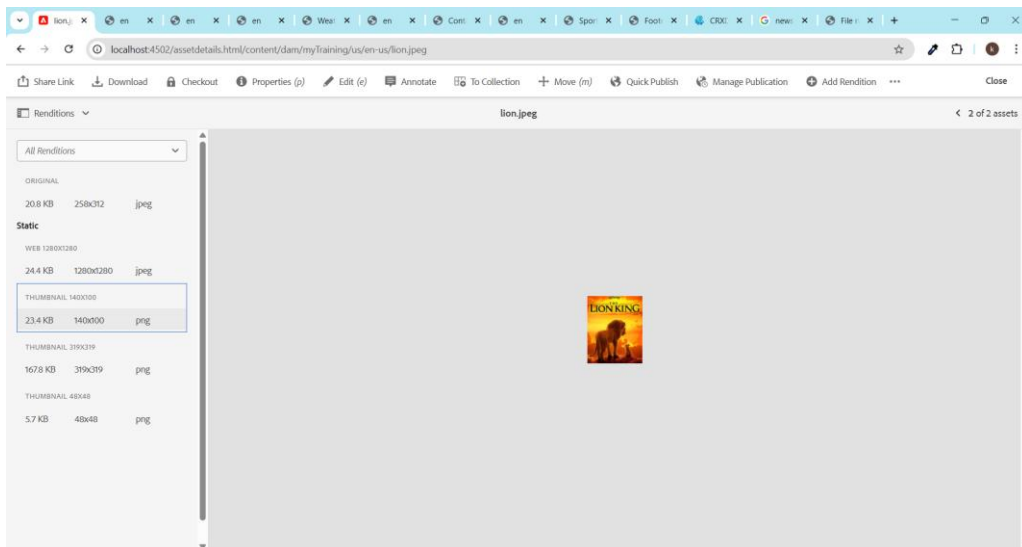
## 2. What are Renditions? Check Renditions for Uploaded Images

**Renditions** are automatically generated variations of an image in different sizes and formats. These help optimize content for different devices and screens.

### Steps to Check Renditions:

1. Go to **AEM Assets** and navigate to **/content/dam/myTraining/us/en-us**.
2. Select an uploaded image and open **Properties**.
3. Click on **Renditions** to see available variations.

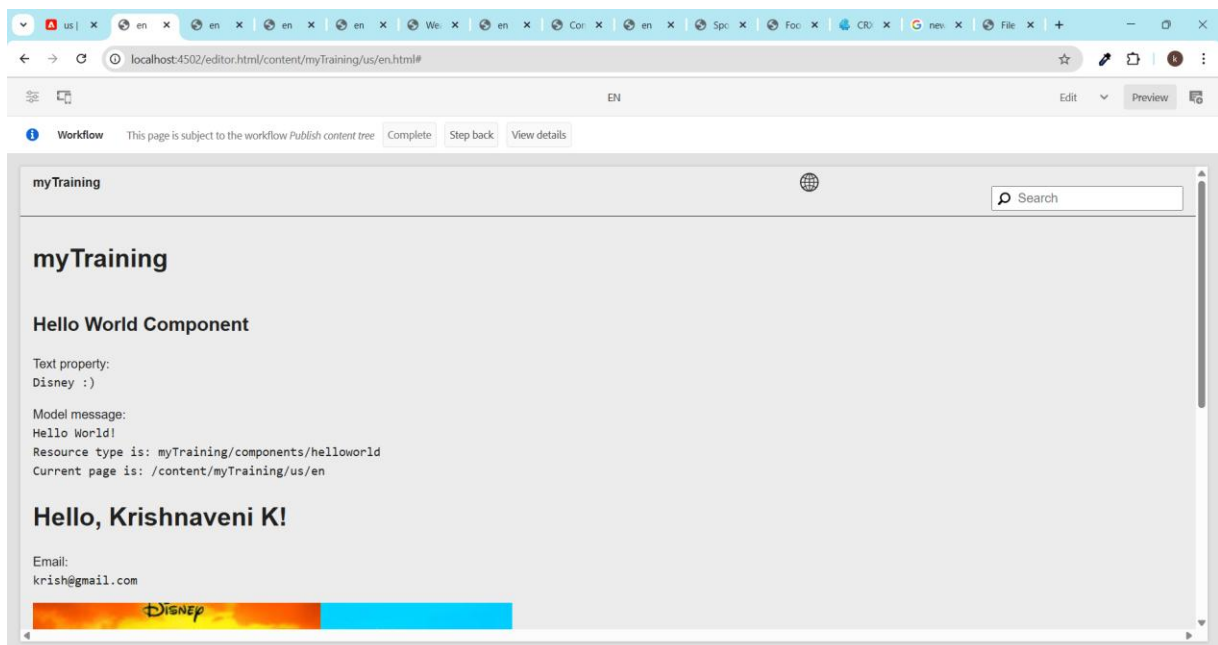




### 3. Add First Name and Last Name Fields in HelloWorld Component Steps:

1. Edit the **HelloWorld Component's** dialog to add two text fields: firstName and lastName.
2. Update the **helloWorld.html** file to print these values using properties:

```
<h1>Hello, ${properties.firstName} ${properties.lastName}!</h1>
```



Added First name : Krishnaveni and Lastname : K as shown in image .

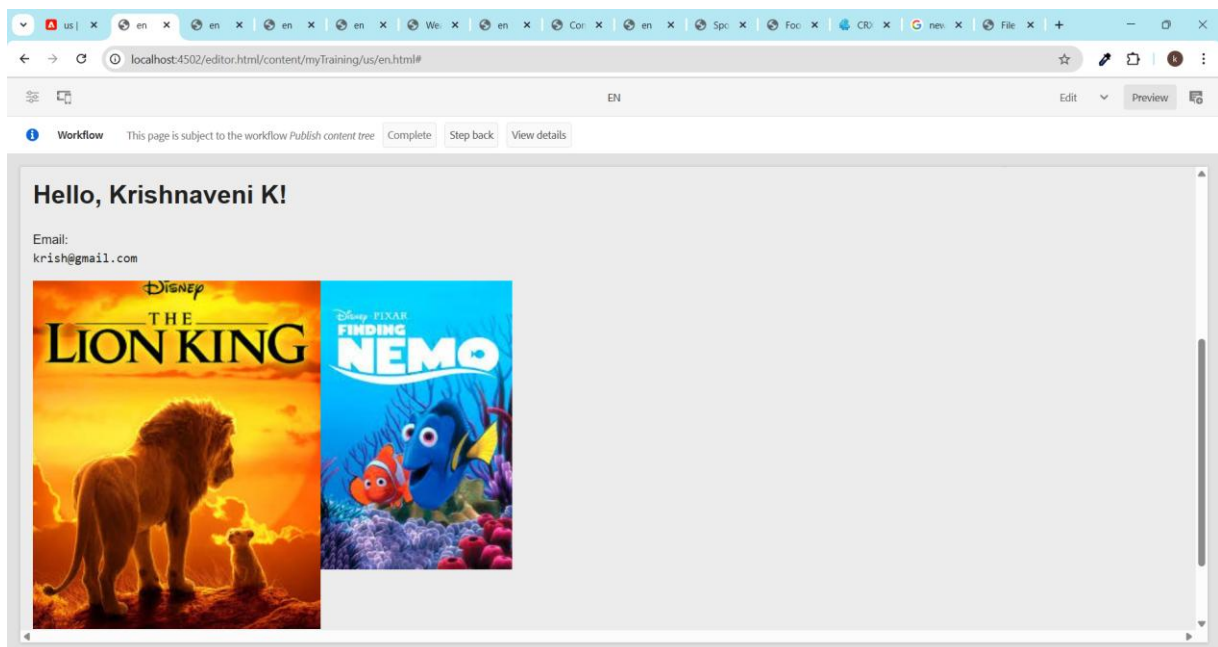
#### 4. Use @ValueMapValue in HelloWorldModel Sling Model

Modify the **HelloWorldModel** in `com.myTraining.core.models.HelloWorldModel` by adding email property:

```
@Model(adaptables = Resource.class)
public class HelloWorldModel {

    @ValueMapValue(injectionStrategy = InjectionStrategy.OPTIONAL)
    @Default(values = "No email provided") private
    String email;

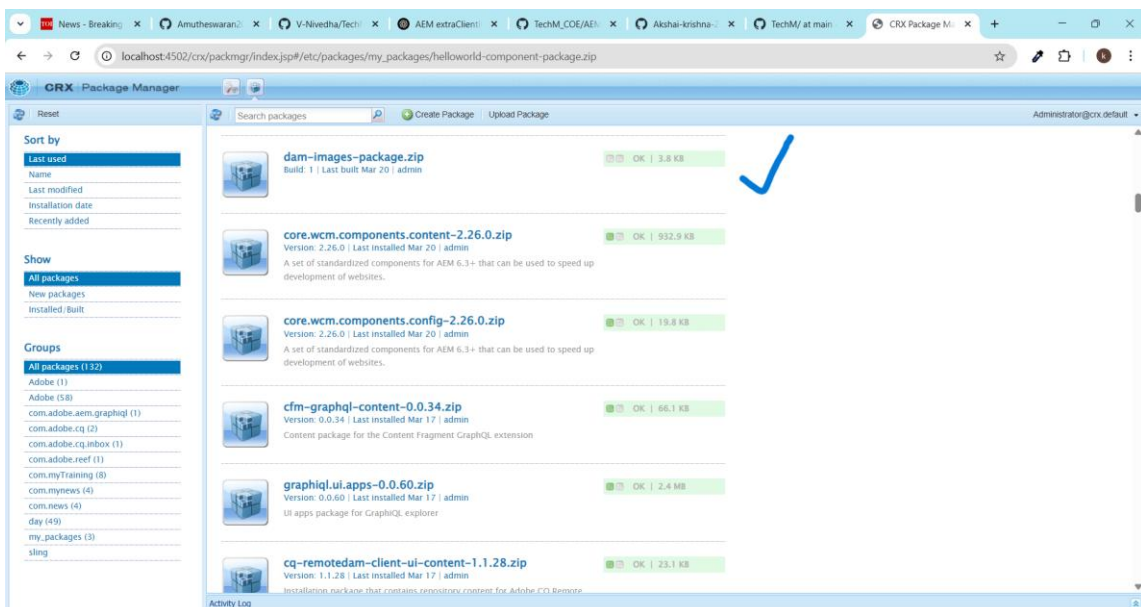
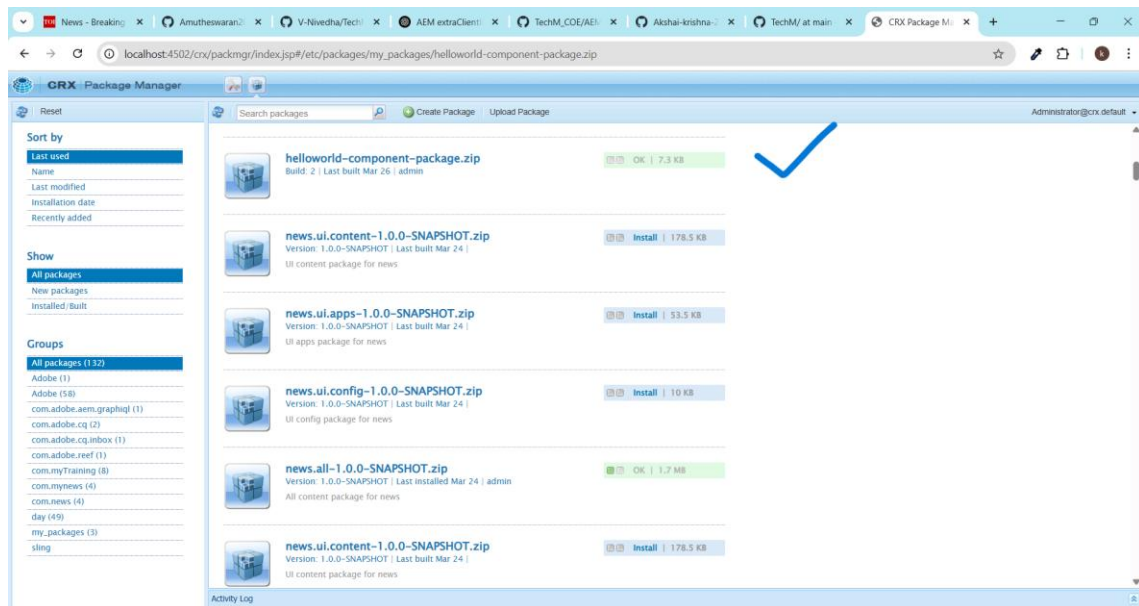
    public String getEmail() {
        return email; }
}
```



Tried **@ValueMapValue** by adding email in the component .

#### 5. Why Use Package Manager? Create Packages for HelloWorld Component & Images

**Package Manager** in AEM is used for **exporting, transferring, and deploying content** between different AEM environments. It helps bundle various resources such as components, templates, pages, and assets into a package that can be easily installed on another AEM instance.



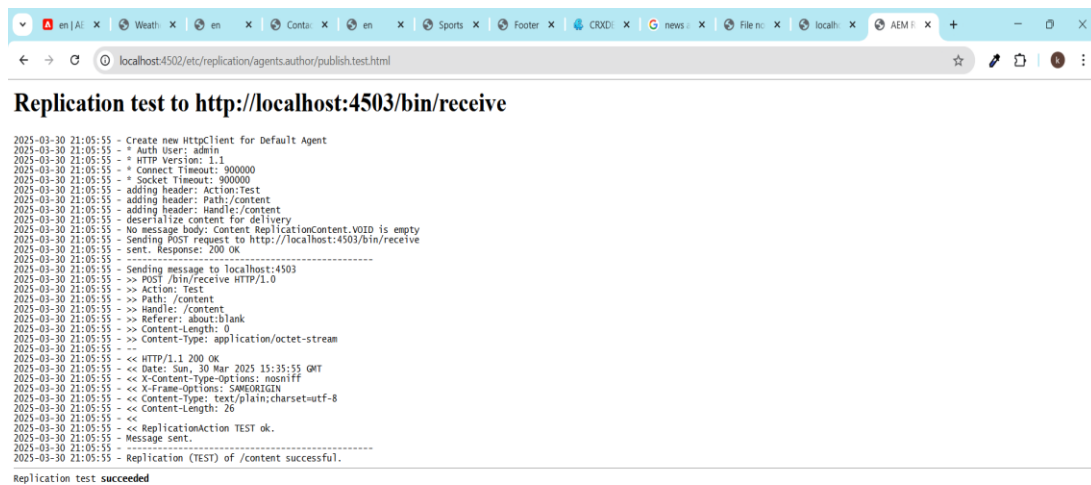
## 6. Configure Replication Agent & Publish Page from Author to Publish (4502 → 4503)

**Replication Agent** ensures that content from the **author instance (4502)** is published to the **publish instance (4503)**.

### Steps to Configure:

1. Go to **AEM Tools** → **Replication** → **Agents on Author**.
2. Click on **Default Agent (publish)** and configure it with the **publish server details** (e.g., <http://localhost:4503>).

### 3. Save and test the replication connection.



The screenshot shows a web browser window with the address bar displaying `localhost:4502/etc/replication/agents.author/publish.test.html`. The page title is "Replication test to http://localhost:4503/bin/receive". The main content area displays a detailed log of the replication test process, including the creation of an HttpClient, adding headers, sending a POST request, and receiving a 200 OK response. The log concludes with "Replication test succeeded".

```
2025-03-30 21:05:55 - Create new HttpClient for Default Agent
2025-03-30 21:05:55 - * Auth User: admin
2025-03-30 21:05:55 - * HTTP Version: 1.1
2025-03-30 21:05:55 - * Connect Timeout: 900000
2025-03-30 21:05:55 - * Socket Timeout: 900000
2025-03-30 21:05:55 - adding header: Action:Test
2025-03-30 21:05:55 - adding header: Path:/content
2025-03-30 21:05:55 - adding header: Handle:/content
2025-03-30 21:05:55 - deserialize content for delivery
2025-03-30 21:05:55 - No message body: Content: ReplicationContent.VOID is empty
2025-03-30 21:05:55 - Sending POST request to http://localhost:4503/bin/receive
2025-03-30 21:05:55 - sent. Response: 200 OK
2025-03-30 21:05:55 - -----
2025-03-30 21:05:55 - Sending message to localhost:4503
2025-03-30 21:05:55 - >> POST /bin/receive HTTP/1.0
2025-03-30 21:05:55 - >> Action: Test
2025-03-30 21:05:55 - >> Path: /content
2025-03-30 21:05:55 - >> Handle: /content
2025-03-30 21:05:55 - >> Referer: about:blank
2025-03-30 21:05:55 - >> Content-Length: 0
2025-03-30 21:05:55 - >> Content-Type: application/octet-stream
2025-03-30 21:05:55 - --
2025-03-30 21:05:55 - << HTTP/1.1 200 OK
2025-03-30 21:05:55 - << Date: Sun, 30 Mar 2025 15:35:55 GMT
2025-03-30 21:05:55 - << X-Content-Type-Options: nosniff
2025-03-30 21:05:55 - << X-Frame-Options: SAMEORIGIN
2025-03-30 21:05:55 - << Content-Type: text/plain;charset=utf-8
2025-03-30 21:05:55 - << Content-Length: 26
2025-03-30 21:05:55 - <<
2025-03-30 21:05:55 - << ReplicationAction TEST ok.
2025-03-30 21:05:55 - Message sent.
2025-03-30 21:05:55 - -----
2025-03-30 21:05:55 - Replication (TEST) of /content successful.

Replication test succeeded
```

### Steps to Publish a Page:

1. Open the AEM **author** instance (**4502**).
2. Navigate to the page and click **Publish Page**.
3. Check the published page in the **publish** instance (**4503**) by accessing its URL.