

Adobe Experience Manager (AEM) Hands-On Training Guide

Project Context: This document captures a detailed walkthrough of practical tasks executed on AEM Author and Publish instances, aligned to the objectives dated **19-3-25**. The tasks cover DAM utilization, component development, image renditions, Sling Models, Package Manager usage, and replication agent configuration.

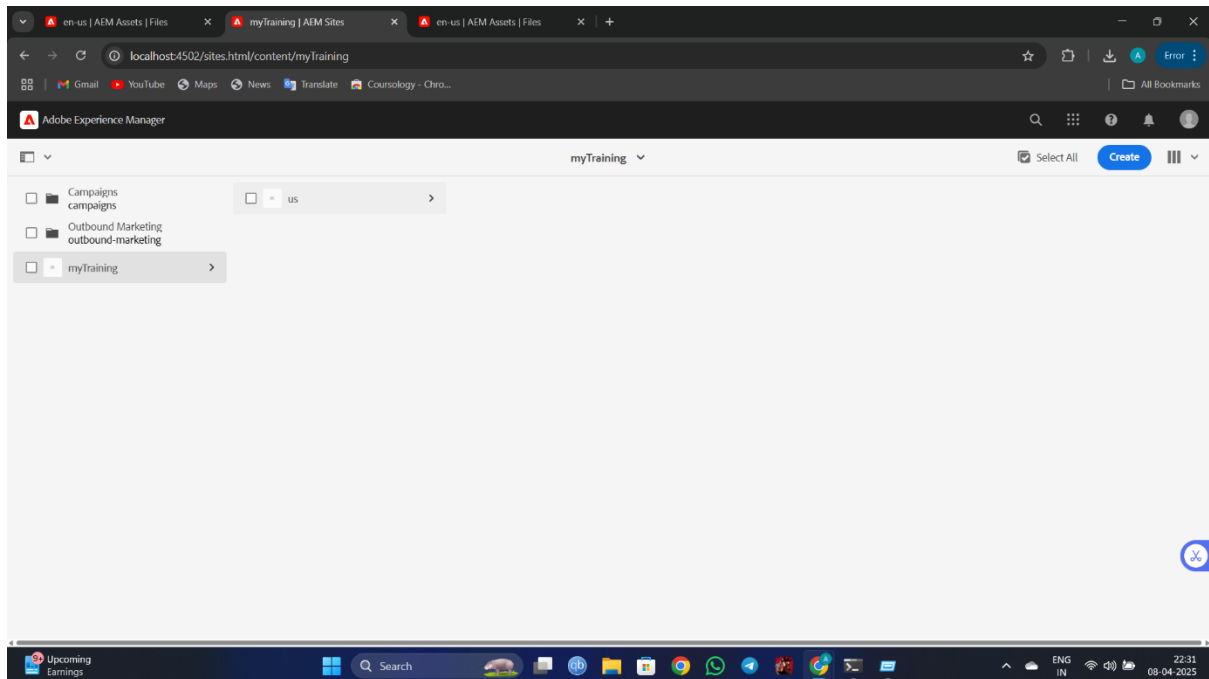
1. What is DAM and Why Do We Use It?

DAM (Digital Asset Management) in AEM refers to a centralized repository used to manage digital assets (e.g., images, videos, documents).

Purpose:

- Efficiently store and retrieve assets
- Enable version control and metadata tagging
- Allow reuse across multiple sites/pages
- Manage renditions and automated workflows

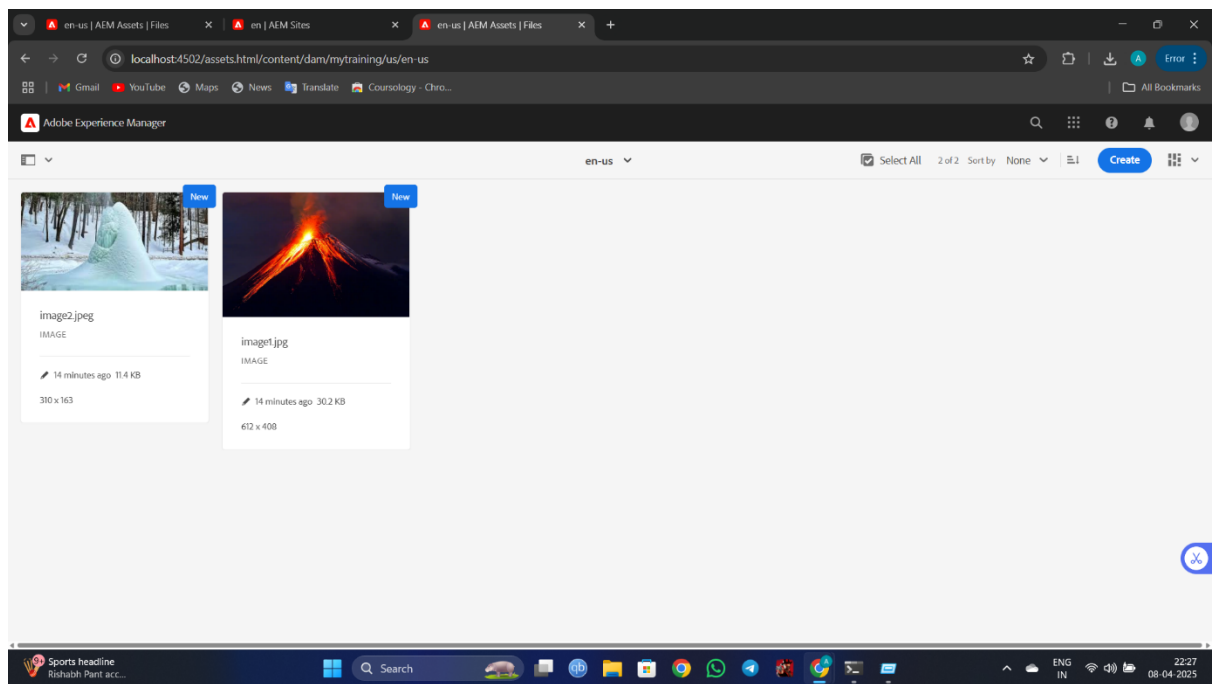
Path: <http://localhost:4502/assets.html/content/dam>



2. Create Folder and Upload Assets to DAM

Step-by-step:

- Navigate to: <http://localhost:4502/assets.html/content/dam>
- Create a folder path: `/content/dam/myTraining/us/en-us`
 - This means: Inside myTraining, create nested folders us/en-us
- Upload **two images** to en-us
- Navigate to your site page:
<http://localhost:4502/sites.html/content/myTraining/us/en>
- Use the **Image Component** to drag and drop the images onto the page.



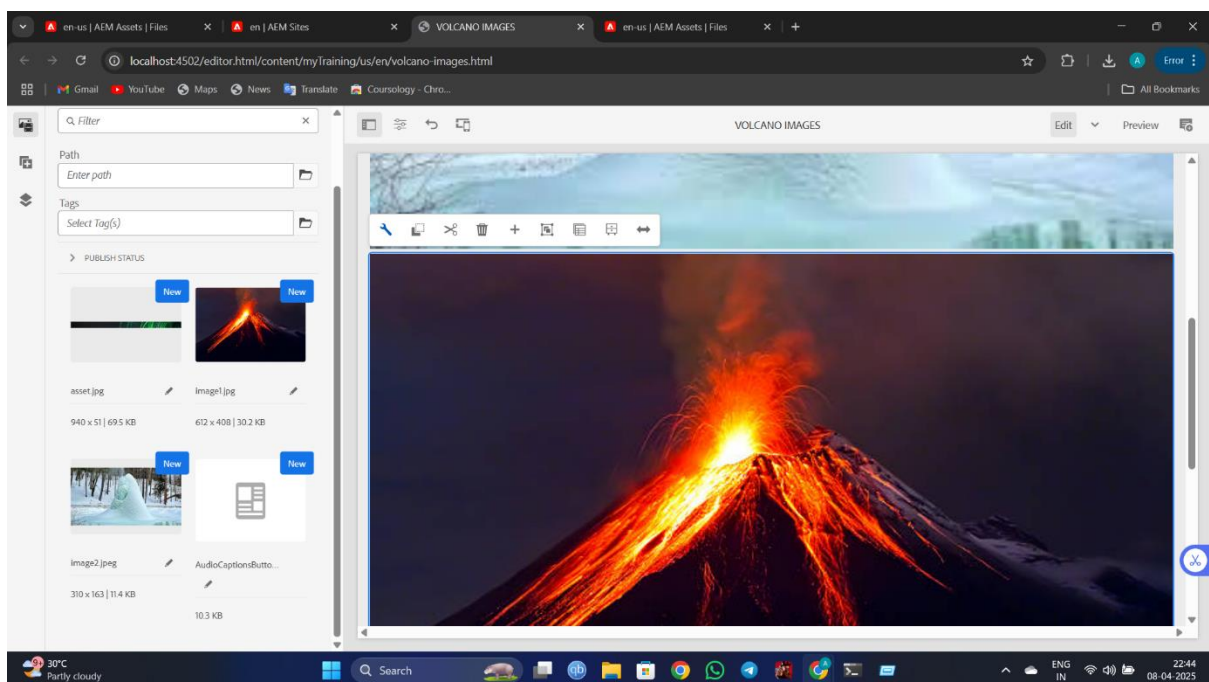
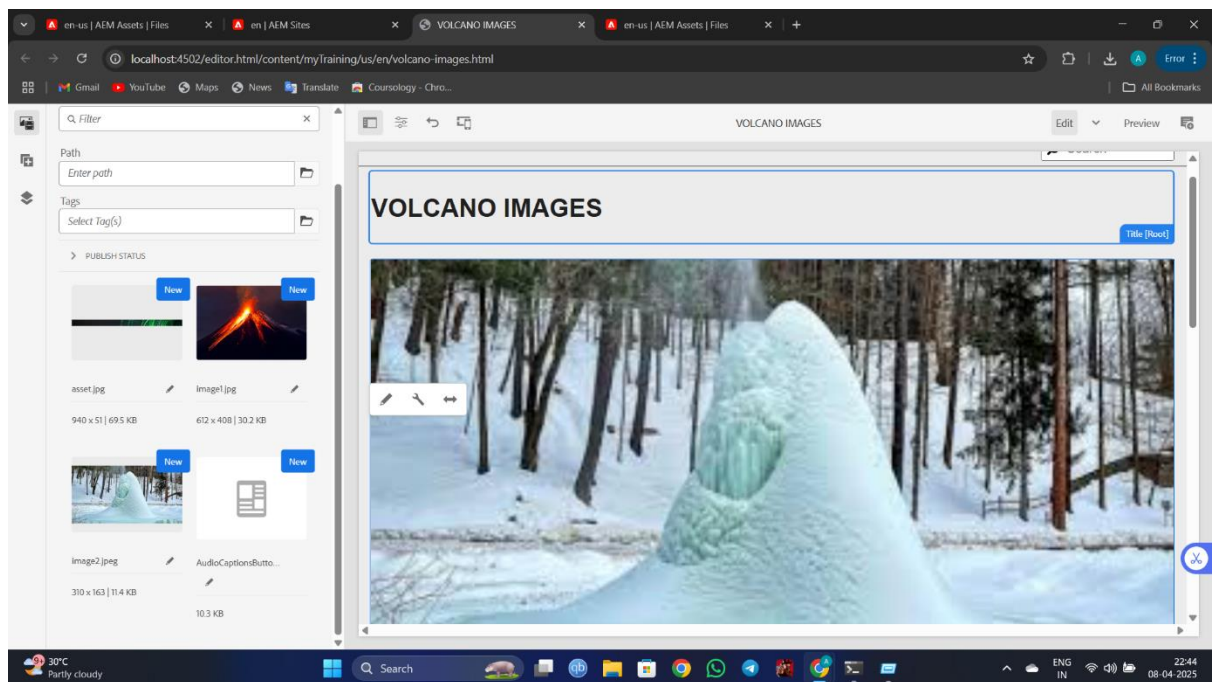
3. What are Renditions?

Renditions are auto-generated image variants of uploaded assets in AEM, used for responsive design and performance optimization.

To check renditions:

- Go to:
<http://localhost:4502/assets.html/content/dam/myTraining/us/en-us>
- Click on an image → Select **Renditions** tab

- View thumbnails, web versions, and custom sizes



4. Update HelloWorld Component with First Name and Last Name

Steps:

- Navigate to: /apps/myTraining/components/helloworld/dialog

- Add the following structure:

tabs

└─ items

└─ general

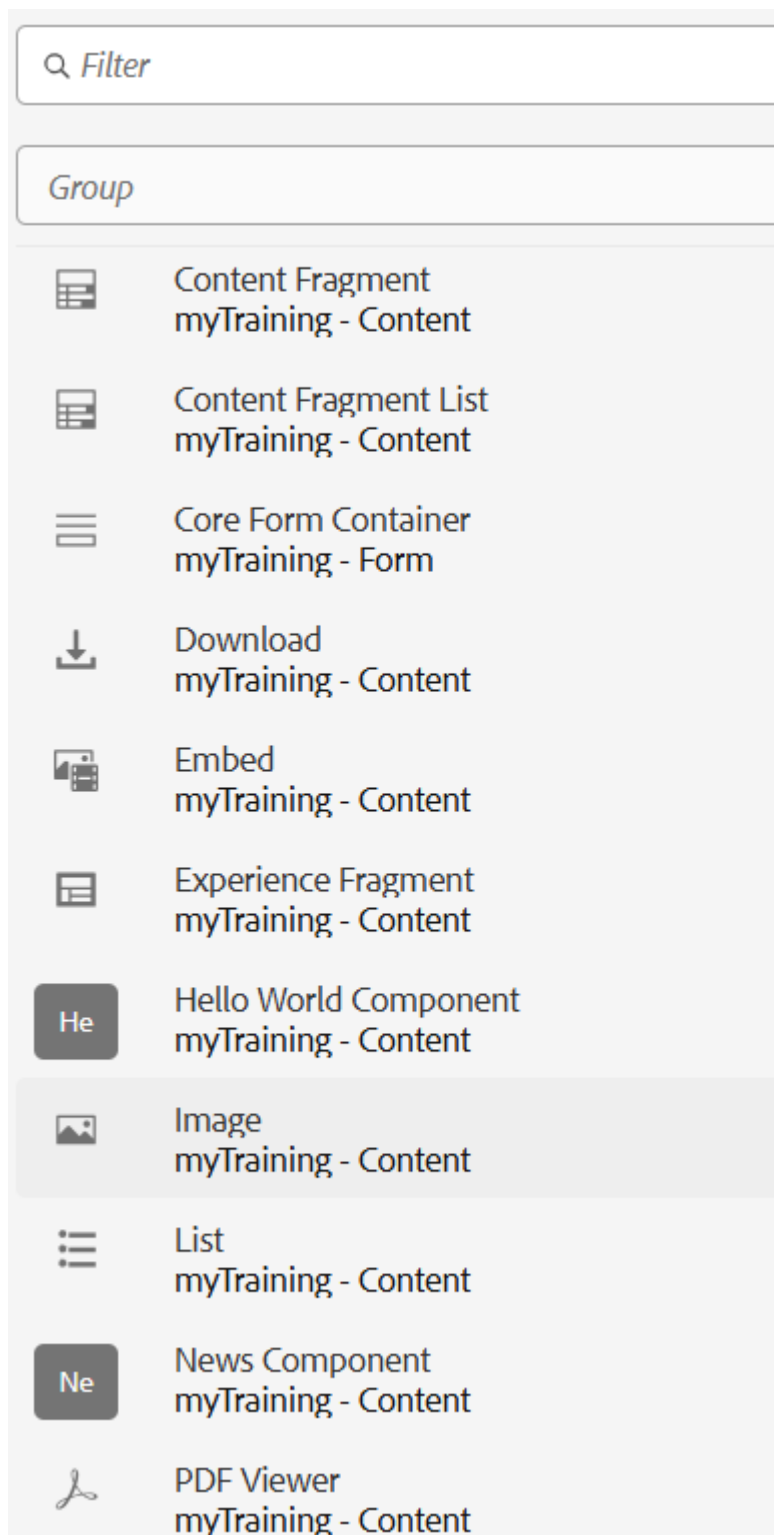
└─ items

└─ firstname (Textfield)

└─ lastname (Textfield)

- Use `granite/ui/components/coral/foundation/form/textfield` for both fields

- Save changes



5. Use @ValueMapValue in HelloWorldModel

Update the Sling Model:

```
@Model(adaptables = Resource.class)
```

```
public class HelloWorldModel {
```

```
    @ValueMapValue
```

```
    private String firstname;
```

```
    @ValueMapValue
```

```
    private String lastname;
```

```
    public String getFirstname() {
```

```
        return firstname;
```

```
    }
```

```
    public String getLastname() {
```

```
        return lastname;
```

```
    }
```

```
}
```

Update helloworld.html:

```
<div>
```

```
    <p>First Name: ${properties.firstname}</p>
```

```
    <p>Last Name: ${properties.lastname}</p>
```

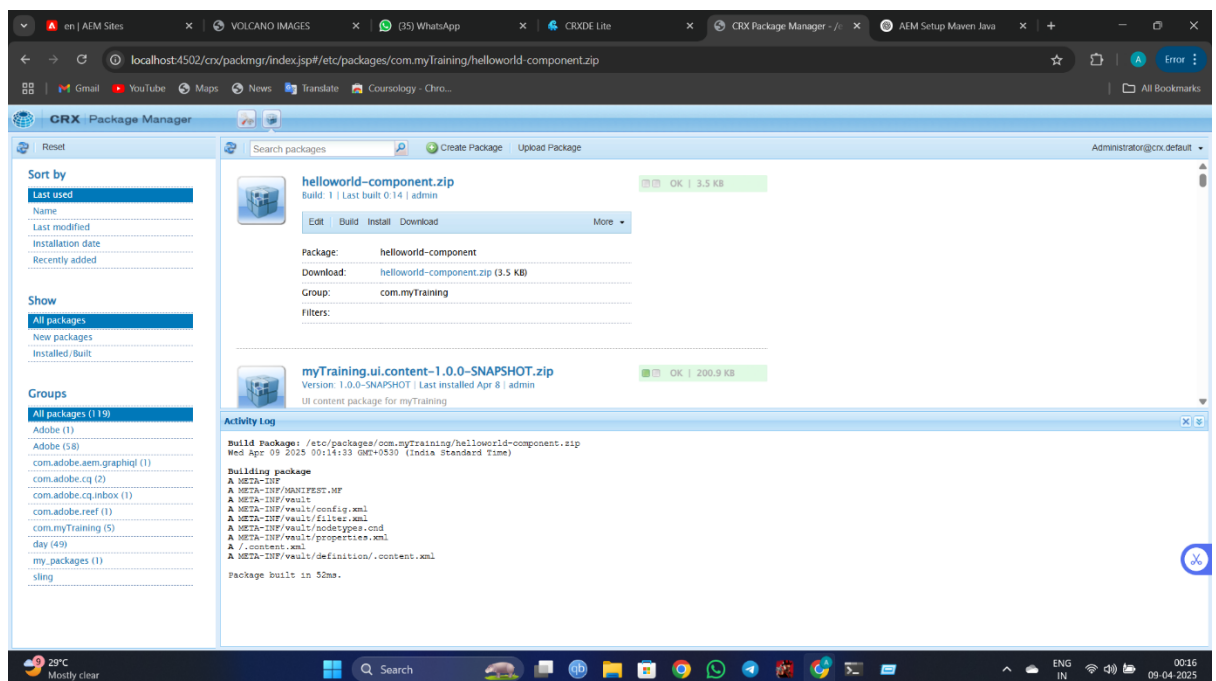
```
</div>
```

6. Package Manager: Create and Deploy Packages

Purpose: To bundle AEM resources (components, content, assets) for transport across instances.

Steps:

1. Navigate to: <http://localhost:4502/crx/packmgr>
2. Click **Create Package**
3. Enter:
 - Group: com.myTraining
 - Name: helloworld-component
 - Version: 1.0.0
4. Add filters:
 - /apps/myTraining/components/helloworld
5. Repeat to create another package:
 - Name: training-assets
 - Filter: /content/dam/myTraining/us/en-us
6. Build both packages and download ZIP files



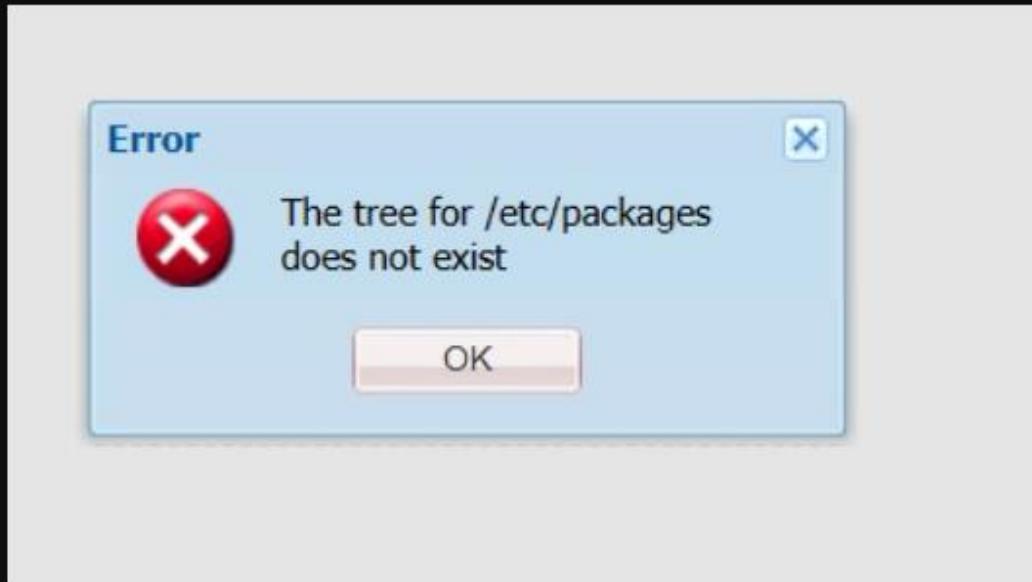
7. Replication Agent Setup (Author to Publish)

Purpose: To publish content/assets from Author (4502) to Publish (4503)

Steps:

1. Start both instances:
 - Author: `java -Xmx6g -jar aem-author-p4502.jar`
 - Publish: `java -Xmx6g -jar aem-publish-p4503.jar`
2. Author: Go to
`http://localhost:4502/etc/replication/agents.author.html`
3. Configure **default agent** or create a new agent:
 - Title: To Publish
 - URL:
`http://localhost:4503/bin/receive?sling:authRequestLogin=1`
 - User: admin, Password: admin
4. Test connection
5. Go to site page → Click **Publish Page**
6. Visit `http://localhost:4503/sites.html/content/myTraining` to verify

Note: If `/etc/packages` path is missing in publish, create it manually using CRXDE Lite or install a patch package.



End of Document

For further assistance, maintain this structure for repeated tasks and adapt component naming and structure according to your project conventions.