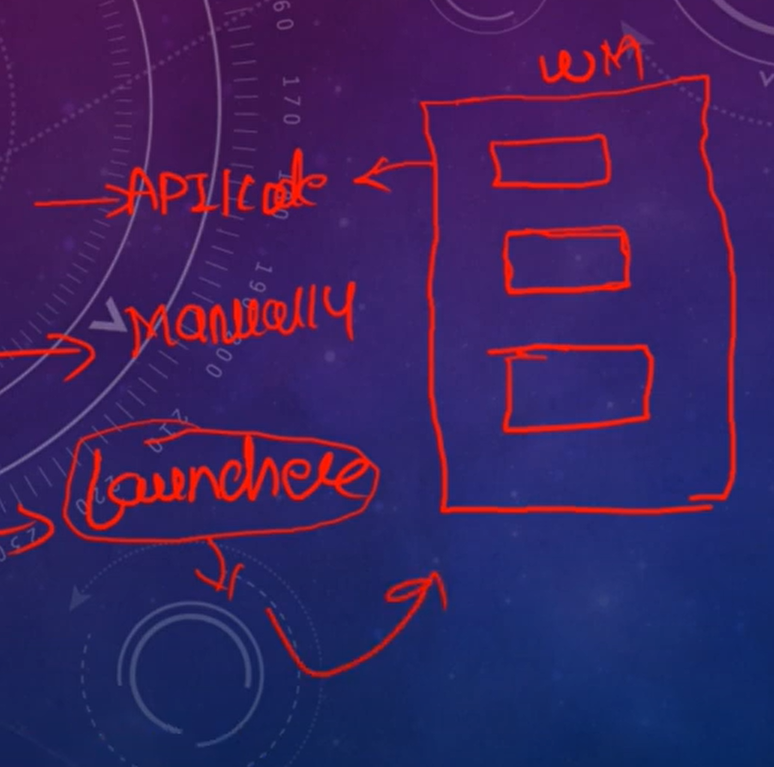
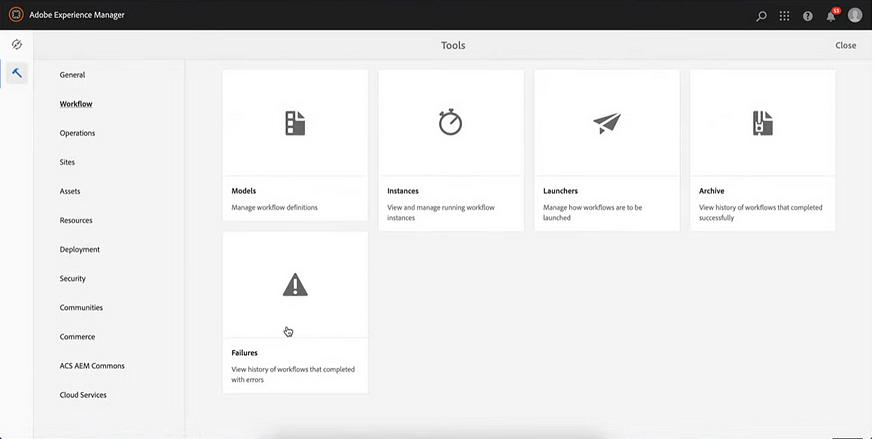
**WORKFLOWS**

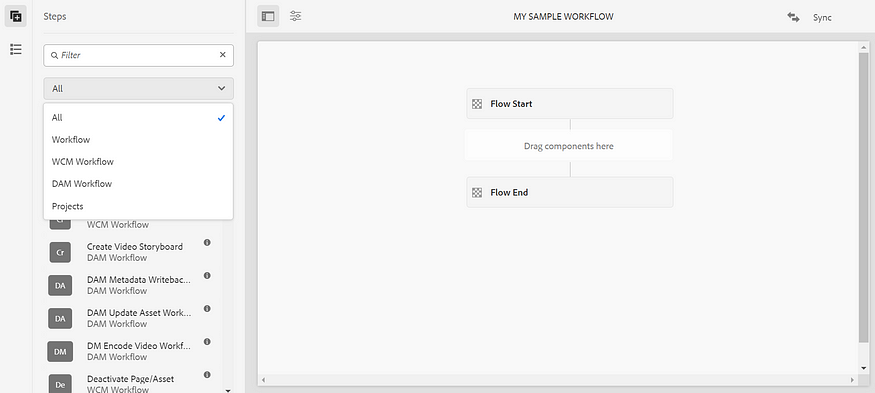
**We can intiate the workflow in 3 different ways using API/Code , Manually , Launchers**

****

AEM Workflows allow you to automate a series of steps that are performed on (one or more) pages and/or assets



Once you create a workflow model- this is like a blueprint for the workflow

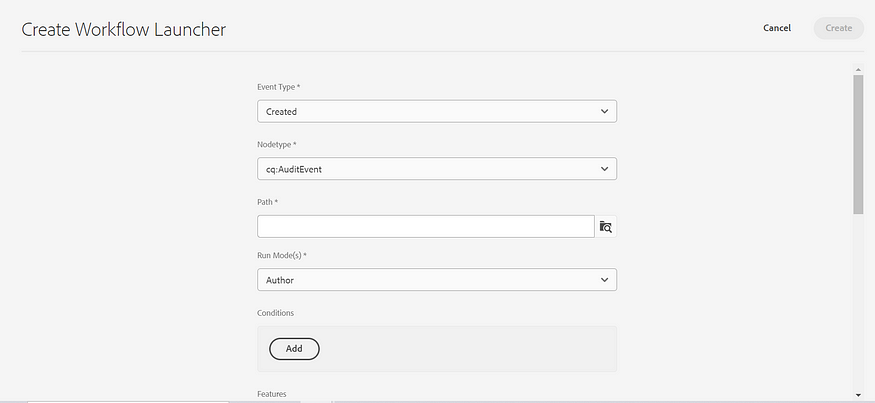


Note that, the model always has a start and a end node.

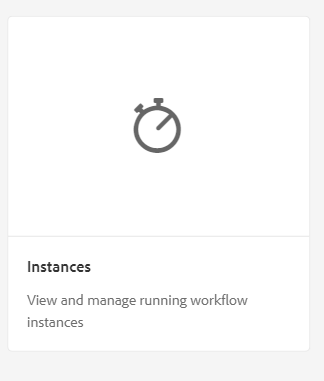
Once you create a workflow, you'll have to create a launcher to automate the trigger for the workflow-

**Required properties:**

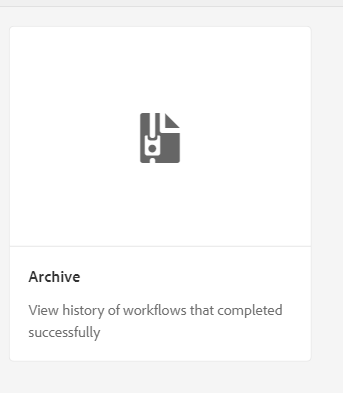
* *Event Type:* It contains the type of event to trigger the workflow like created, modified and removed.
* *Node Type:*It is used to mention the primary type of your node.
* *Path:*Defines the path on which the workflow is activated(payload).
* *Run Modes:*It contains the run modes on which the workflow needs to be run- author, publish or both.
* *Workflow Model:*To select the workflow model from the list.



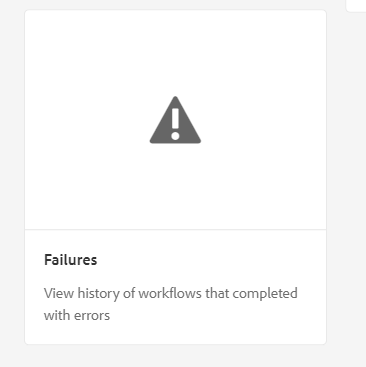
Once the workflow is triggered, you will find the workflow instance in



 completed workflows in



If your workflow fails it will show the same in



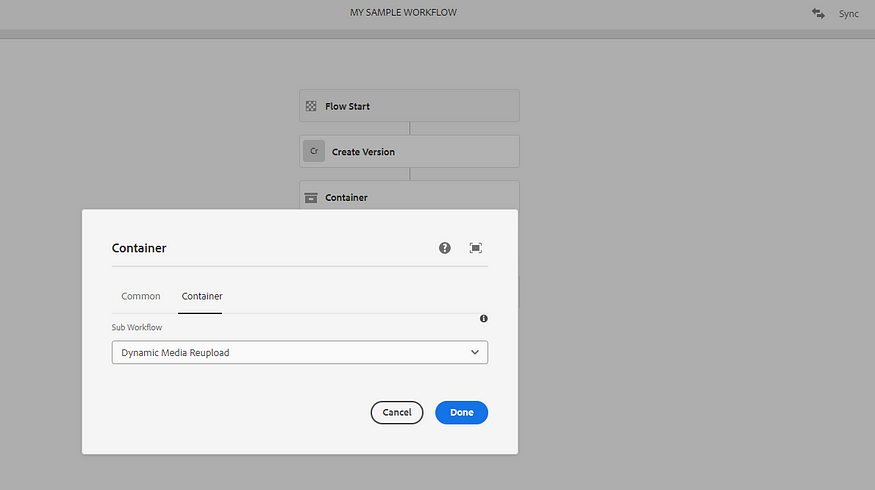
**Note:**

 workflow models and launchers(Model designs) are stored under /conf, but workflow running instances(Runtime models) are stored in /var.

* Process Step: Used to execute a service or ECMA script at runtime. This is generally used when we want our application to execute a certain logic.
* Participant Step: Used to assign an ID of the user for the generated work item.
* Dynamic Participant Step: Used to selects the ID of the user that is assigned the work item via service or ECMA script.
* Decision Step: This contains the AND and OR operations for the workflow via routing expressions.

Payload: Defines the resource on which the workflow is performed eg: page, asset etc.

We also can call another workflow inside a workflow using container step



We can check the versions changed or not after executing the workflow

