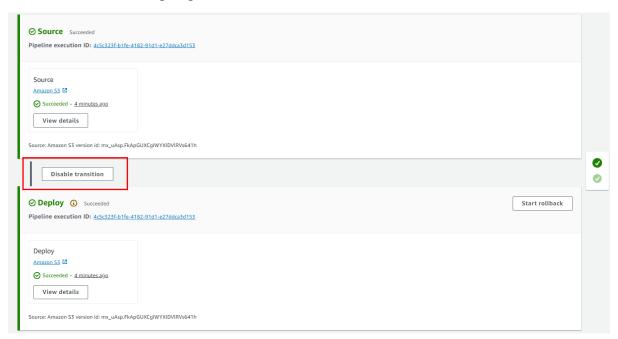
Triggering Pipeline by updating Source

In this lab, you are working with AWS CodePipeline to manage the deployment of a website. The key steps involve updating the website by making changes to an HTML file, creating a zipped folder of the updated content, and uploading it to an S3 bucket. The pipeline is designed to automatically detect new uploads and trigger a deployment. However, to demonstrate control over the deployment process, you temporarily disable the transition between the 'Source' and 'Deploy' stages. This action pauses the pipeline execution, preventing the changes from being deployed immediately. After verifying that the website has not been updated, you re-enable the transition to complete the deployment process.

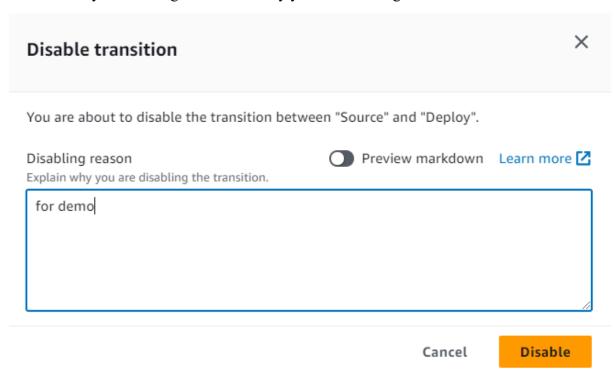
The end goal is to demonstrate how AWS CodePipeline can manage the automated deployment of updates while allowing for manual control over the transition between stages. This ensures that changes are deployed only when you're ready, providing flexibility and control in the deployment process.

😄 To begin with the Lab:

- 1. In the previous lab, we created our first pipeline using S3 buckets as source and deployment locations. AWS CodePipeline executed the pipeline automatically for the first time and deployed our content. But what if we need to update our website? Will our pipeline work automatically when we upload a new version to our source bucket? Well, in this lab, we will provide an example of this.
- 2. Let's talk about transitions on AWS CodePipeline. If you remember from the previous lab, the transitions connect stages. And this arrow represents the transition from our 'Source' stage to the 'Deploy' stage. By default, all transitions are enabled when you create a pipeline. On our pipeline, this means that as soon as the source stage is completed, the pipeline execution continues with the deploy stage. But you can also disable a transition and then re-enable it whenever you want in the future.
- 3. So, now we are going to disable this transition, click on disable.



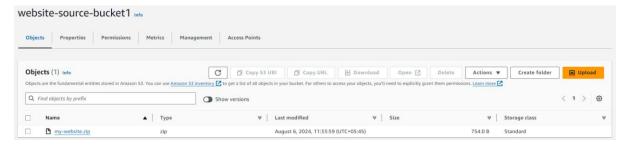
4. Now you need to give a reason why you are disabling transition.



- 5. Now we need to open our HTML file and make some changes to it. Below you can see that in the file we just changed the version of our website and then saved it.
- 6. Then you need to make a zipped folder out of your website folder and upload it again to your S3 bucket.



7. Once it is uploaded then it should trigger our pipeline. So, move to the code pipeline in your console.



8. Below you can see that the source stage was executed almost immediately.



- 9. Because we had disabled the transition, because of that our execution is queued and has not yet been gone to the deployed stage.
- 10. So, this means that our website has not yet been updated; to update we need to enable the transition. But first, let's check our website and as expected you can see that our website has not been updated yet.
- 11. Now we are going to enable the transition.

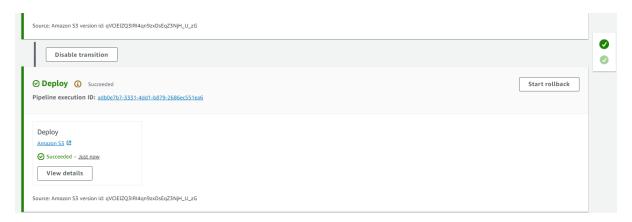


Welcome to Our Website!

This is a simple, static website that we deploy by creating a pipeline on AWS CodePipeline.

Website Version: 1.0

12. Below you can see that our deployment has also been completed, you can also check the version IDs of both source and deploy stages. If they are the same, then it means that our pipeline has been executed successfully.



13. Now, let's check our check website again and you can see that the changes were deployed successfully.

Welcome to Our Website!

This is a simple, static website that we deploy by creating a pipeline on AWS CodePipeline.

Website Version: 2.0