



Editing Pipeline and Trigger it Manually

In this lab, you're extending an existing AWS CodePipeline to support cross-regional deployments. The initial setup involved deploying a website from an S3 source bucket to a production bucket in the same region. Now, you're adding a new deployment action to deploy the website to an additional S3 bucket located in the North Virginia region.

The steps include creating a new S3 bucket in the North Virginia region, modifying the pipeline to deploy to both the original and new production buckets in parallel, and configuring the pipeline to extract files before deployment. The process includes addressing a potential issue where the pipeline setup might fail due to a missing default artifact store for cross-region actions.

The end goal is to demonstrate the capability of AWS CodePipeline to handle cross-regional deployments, allowing the website to be deployed to multiple regions. This setup is useful for scenarios where you want to have your application available in different geographic locations, improving availability and performance for users in those areas. After testing the deployment, you delete the pipeline as part of the cleanup process.

😊 To begin with this Lab:

1. In the previous labs, we created a simple pipeline and triggered it by uploading a new version of our website to our source bucket. In this lab, we will add a new deployment action to this pipeline and then trigger it manually from the CodePipeline Console. If you remember, we use an S3 bucket as the source location of our pipeline. This bucket is in the same region as our pipeline because this is a requirement of AWS CodePipeline. We have a single source action in our 'Source' stage. In the 'Deploy' stage, we again have a single action. After extracting its contents, this action deploys the 'Source Artifact' to the production bucket. This bucket is also in the same AWS region as our pipeline.
2. However, unlike source buckets, S3 buckets used as deploy locations can be in other AWS regions, allowing cross-regional deployments with your pipeline.
3. So, now in this lab we will create a production bucket in the North Virginia region. Change your region to N. Virginia and then click on Create bucket.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type | [Info](#)

General purpose

Recommended for most use cases and access patterns.
General purpose buckets are the original S3 bucket type.
They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory - New

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name | [Info](#)

us-production-bucket-01

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#) 

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

4. Turn off the block public access setting and then create your bucket after that change the bucket policy. In the end, enable static website hosting.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::us-production-bucket-01/*"  
    }  
  ]  
}
```

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Edit

Static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)
<http://us-production-bucket-01.s3-website-us-east-1.amazonaws.com>

5. Once it is done go to your code pipeline and click on edit.

Developer Tools > CodePipeline > Pipelines > DemoWebsitePipeline

DemoWebsitePipeline

Pipeline type: V2 Execution mode: QUEUED

Notify Edit Stop execution Clone pipeline Release change

Source Succeeded

Pipeline execution ID: [adb0e7b7-3331-4dd1-b879-2686ec551ea6](#)

6. After that go to the bottom of the page and here you will see that you can edit the stages for both source and deployment. But for now, you need to edit the deployment stage.

The screenshot shows the AWS CloudFormation console interface. At the top, there's a header bar with the AWS logo and navigation links. Below it, the main area is divided into two sections: 'Edit: Source' and 'Edit: Deploy'.
Edit: Source: Contains a single stage named 'Source' with the provider 'Amazon S3'. A 'View details' button is next to it.
Edit: Deploy: Contains a single stage named 'Deploy' with the provider 'Amazon S3'. Below this stage, there's a checkbox labeled 'Configure automatic rollback on stage failure'. At the bottom of the Deploy section is a '+ Add stage' button. In the top right corner of the Deploy section, there's a 'Edit stage' button, which is highlighted with a red box.

7. Here you can see that you have some options to choose from so, you need to choose add action to deploy to both deployment locations in parallel.

This screenshot shows the 'Edit: Deploy' stage configuration screen. At the top, there are buttons for 'Add entry condition', 'Add success condition ▾', 'Add failure condition', 'Cancel', 'Delete', and 'Done'.
Below these are two 'Add action group' buttons. The first group contains a stage named 'Deploy' with the provider 'Amazon S3'. To the right of this stage is a '+ Add action' button, which is highlighted with a red box.
At the bottom left is another '+ Add action group' button, and at the bottom center is a '+ Add stage' button.

8. Then you need to give it an action name and then choose the action provider as S3, the region is N. Virginia and choose your input artifact as your source bucket.
9. After that choose your production bucket which is in N. Virginia and choose extract file before deploying then click on done.

Action name
Choose a name for your action

No more than 100 characters

Action provider

Region

Input artifacts
Choose an input artifact for this action. [Learn more](#)

No more than 100 characters

Bucket

Deployment path - optional

Extract file before deploy
The deployed artifact will be unzipped before deployment.

► Additional configuration

Variable namespace - optional
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. [Learn more](#)

Cancel Done

10. Then you will see that you have another deployment stage. Now you should click on save.

Edit: Deploy

Add entry condition Add success condition ▾ Add failure condition

+ Add action group

Deploy (i) Amazon S3 [edit]	USDeploymentPipeline (i) Amazon S3 [edit]	+ Add action
---	---	---

+ Add action group

Configure automatic rollback on stage failure

+ Add stage

11. While saving your changes, if you encounter a weird error, like a networking error, it results from the missing default artifact store in the second region. For some reason, the Code Pipeline Console started checking it here and stopped creating default artifact stores for the cross-region actions added later. It seems like a temporary issue.

✖ (i) NetworkingError
 Network Failure

✖ (i) NetworkingError
 Network Failure

[Developer Tools](#) > [CodePipeline](#) > [Pipelines](#) > [DemoWebsitePipeline](#) > Edit DemoWebsitePipeline

Editing: DemoWebsitePipeline Delete Cancel Save

12. So, to resolve this issue you can create a new temporary pipeline. Now click on create the pipeline.

13. Here all the steps are the same to create this pipeline, the source is the same, you just need to choose the N. Virginia bucket for the deploy stage that's it.

The screenshot shows the AWS CodePipeline Pipelines page. At the top, there is a search bar and a navigation bar with links for Developer Tools, CodePipeline, Pipelines, and a Create pipeline button. Below the navigation is a table with one row for the 'DemoWebsitePipeline'. The table columns include Name, Latest execution status, Latest source revisions, Latest execution started, and Most recent executions. The pipeline name is 'DemoWebsitePipeline', its latest execution status is 'Succeeded', and it was started '1 hour ago'. There is a 'View details' link next to the status.

14. Here you can see that the pipeline also gets executed successfully.

The screenshot shows the AWS CodePipeline Pipeline execution details page for 'UStempPipeline'. It displays two stages: 'Source' and 'Deploy'. The 'Source' stage is listed as 'Succeeded' with a timestamp of 'Just now'. The 'Deploy' stage is also listed as 'Succeeded' with the same timestamp. Both stages have a 'View details' button. At the bottom right of the page, there are two green checkmark icons.

15. Also, you will see that in S3 that artifact bucket created in N. Virginia region.

The screenshot shows the AWS S3 General purpose buckets page. It lists five buckets: 'codepipeline-eu-west-1-802998226446', 'website-production-bucket', 'website-source-bucket1', 'codepipeline-us-east-1-116094896326', and 'us-production-bucket-01'. Each bucket entry includes a 'Name' column, an 'AWS Region' column (either 'Europe (Ireland)' or 'US East (N. Virginia)'), an 'IAM Access Analyzer' column with a 'View analyzer' link, and a 'Creation date' column. The buckets were created between August 5 and August 6, 2024.

16. Now go back to our previous code pipeline and try to save our content. You can see that our pipeline was saved successfully.

Success

Pipeline was saved successfully.

[Developer Tools](#) > [CodePipeline](#) > [Pipelines](#) > [DemoWebsitePipeline](#)

17. But you will see that it does not ran and we need to run it manually this time.

The screenshot shows the AWS CodePipeline console. At the top, there is a green success banner with the text "Pipeline was saved successfully.". Below the banner, the navigation path is shown: "Developer Tools" > "CodePipeline" > "Pipelines" > "DemoWebsitePipeline". The main content area displays a pipeline execution. The execution status is "Deploy" with a green checkmark icon, labeled "Succeeded". The pipeline execution ID is "adb0e7b7-3331-4dd1-b879-2686ec551ea6". Below the status, there are two columns of details. The left column shows a "Deploy" step using "Amazon S3" as the provider, which succeeded one hour ago. A "View details" button is available for this step. The right column shows a "USDeploymentPipeline" step using "Amazon S3" as the provider, which has not run yet ("Didnt Run") and has "No executions yet". At the bottom of the execution details, the source version ID is listed as "qVCIElZQ3IRI4qn9zxDsEqZ3NjH_U_zG".

18. So, now you need to click on Release change and again on Release.

The screenshot shows the AWS CodePipeline console, specifically the execution details for the pipeline. At the top, there is a row of buttons: "Notify" (with a dropdown arrow), "Edit", "Stop execution", "Clone pipeline", and "Release change". The "Release change" button is highlighted with a red rectangle. This indicates the next step in the process is to click on "Release change" again to trigger the release.

Release change

X

Releasing a change will detect the most recent change in each location configured in your source action(s), and run that change through the pipeline. Do you want to continue?

► Source revision overrides

A source revision is the version of the source artifact with all the changes to your application code for the pipeline execution.

Cancel

Release

19. Below you can see that our sources stage was successful and our deploy stage was also successful only for the same region bucket but the bucket which is in N. Virginia has failed the deployment.

The screenshot shows the AWS CloudWatch Pipeline interface. It displays two stages: 'Source' and 'Deploy'. The 'Source' stage is listed as 'Succeeded' with a green checkmark. It uses 'Amazon S3' as the provider and was completed on 'Aug 6, 2024 7:06 AM (UTC)'. A 'View details' button is available. The 'Deploy' stage is listed as 'Failed' with a red circle icon. It also uses 'Amazon S3' as the provider and was completed on 'Aug 6, 2024 7:06 AM (UTC)'. A 'View details' button is available. At the bottom of the screen, there are three buttons: 'Start rollback', 'Retry stage', and 'Retry failed actions'. The entire screenshot is framed by a light gray border.

20. It failed because the bucket is not in the same region.

Action execution details

X

Action name: USDeploymentPipeline Status: Failed

Summary

Input

Status

 Failed

Last updated

Aug 6, 2024 7:06 AM (UTC)

Action execution ID

 b703a9a7-a017-4a73-b0ed-f238ae83e65a

Error code

Invalid action configuration

Error message

The bucket named us-production-bucket-01 is not located in the eu-west-1 AWS region

Done

21. Once you are done delete your pipeline.