



Creating a Pipeline between S3 buckets

CI/CD stands for **Continuous Integration** and **Continuous Delivery** (or **Continuous Deployment**). It's a set of practices used in software development to make the process of getting code from development to production smoother and more reliable. Here's a simple breakdown:

Continuous Integration (CI)

1. **Code Integration:** Developers frequently merge their code changes into a shared repository, often several times a day.
2. **Automated Testing:** Each time code is integrated, automated tests run to check if the new changes work well with the existing code. This helps catch bugs early.

In simple terms: Imagine you're working on a group project. Every time you finish a part, you add it to the main project folder. Someone checks if it fits well with everyone else's work, and if it doesn't, you fix it right away. This way, you always know the project is in good shape.

Continuous Delivery (CD)

1. **Automated Release Process:** After passing tests in CI, the code is automatically prepared for release. This can involve packaging the code and ensuring everything is ready for deployment.
2. **Manual Approval (sometimes):** Before the new version of the software goes live, there may be a manual approval step where someone reviews the changes.

In simple terms: Once your part of the project is checked and fits well, it's packaged up nicely and put in a place where it can be easily accessed. Someone might give it a final check before it's officially shared with everyone.

Continuous Deployment (CD)

1. **Automatic Deployment:** In this case, after the code passes all tests and checks, it's automatically deployed to production without manual approval.

In simple terms: Instead of waiting for a final check, the project update is automatically shared with everyone as soon as it's ready.

Overall, CI/CD helps developers quickly and safely deliver new features, updates, and fixes to users. It ensures that the software is always in a working state and that changes can be released frequently with confidence.

1. CI/CD Pipeline

A CI/CD pipeline is a set of automated processes that software goes through from code changes to deployment. The key stages typically include:

- **Source Stage:** This is where the code is stored in a version control system (like Git). When new changes are pushed to the repository, it triggers the CI/CD pipeline.

- **Build Stage:** The code is compiled and built into an executable format. This step might also include packaging the code with its dependencies.
- **Test Stage:** Automated tests are run to validate the code. Tests can include unit tests (checking individual components), integration tests (checking how components work together), and end-to-end tests (checking the whole application).
- **Deploy Stage:** The code is deployed to a production or staging environment. In continuous deployment, this happens automatically; in continuous delivery, there might be a manual approval step.

2. Benefits of CI/CD

- **Faster Time to Market:** By automating the integration, testing, and deployment processes, CI/CD enables teams to release new features and fixes more quickly.
- **Higher Quality:** Automated testing ensures that code changes are thoroughly tested, reducing the likelihood of bugs reaching production.
- **Consistent Delivery:** The automated nature of CI/CD reduces human error and ensures that deployments are consistent and reliable.
- **Improved Collaboration:** Developers can focus on writing code and rely on the pipeline to handle the integration and deployment, fostering a more collaborative environment.

3. Challenges in Implementing CI/CD

- **Initial Setup and Cost:** Setting up a robust CI/CD pipeline requires an initial investment in tools, infrastructure, and training. This can be time-consuming and costly.
- **Maintenance:** The pipeline itself needs to be maintained. As the codebase and team evolve, the pipeline might need updates and optimizations.
- **Test Coverage:** The effectiveness of CI/CD depends heavily on the quality and coverage of automated tests. If tests are incomplete or poorly written, issues can still slip through.
- **Cultural Shift:** CI/CD requires a cultural change within the organization. Teams must embrace automation, frequent releases, and a mindset of continuous improvement.

4. Popular CI/CD Tools

- **Jenkins:** An open-source automation server widely used for building CI/CD pipelines.
- **GitLab CI/CD:** Integrated CI/CD capabilities within the GitLab platform.
- **CircleCI:** A cloud-based CI/CD service that automates the software development process.
- **Travis CI:** A CI/CD service often used for open-source projects.
- **GitHub Actions:** Integrated CI/CD workflows within GitHub.

5. Use Cases

- **Web Development:** CI/CD is commonly used in web development to frequently update web applications with new features and bug fixes.
- **Mobile App Development:** Automating the build, test, and release process for mobile apps ensures a consistent experience for users.
- **Microservices:** In architectures with many small, independent services, CI/CD helps manage the deployment and integration of each service.

6. Security Considerations

- **Secure Pipelines:** Ensure that the CI/CD pipeline itself is secure, with proper access controls and monitoring.
- **Vulnerability Scanning:** Integrate security checks, such as vulnerability scans, into the pipeline to identify potential security issues early.

In summary, CI/CD is a powerful practice that enhances the efficiency and reliability of software development and deployment. By automating repetitive tasks and ensuring rigorous testing, CI/CD helps teams deliver high-quality software quickly and consistently.



What are we doing in this Lab?

In this exercise, you are setting up a basic CI/CD pipeline using AWS services. The main goal is to automate the deployment of a static website hosted on Amazon S3. Here's a summary of what you're doing and the end goal:

1. Creating S3 Buckets:

- One bucket serves as the **source repository** where you upload a zip file containing your website's content.
- The other bucket is the **production environment**, where the website is hosted and made publicly accessible.

2. Configuring Buckets:

- Enabling versioning on the source bucket to track changes and manage deployments.
- Setting permissions and enabling static website hosting on the production bucket to make the website publicly viewable.

3. Setting Up a Pipeline in AWS CodePipeline:

- Creating a pipeline that automatically triggers when new files are uploaded to the source bucket.
- Configuring the pipeline to deploy the contents from the source bucket to the production bucket.

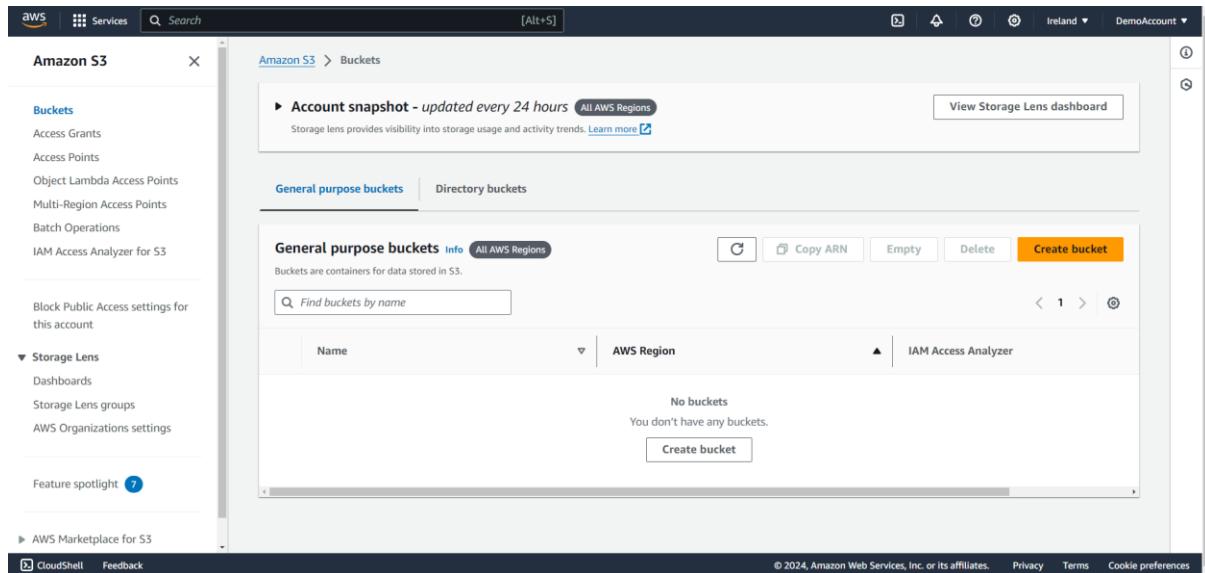
End Goal:

The objective is to automate the deployment process so that whenever you upload a new version of your static website to the source S3 bucket, the CodePipeline is triggered,

automatically deploying the new content to the production bucket. This setup ensures that your website is always up-to-date with the latest changes without manual intervention. The end result is a publicly accessible static website hosted on S3, with an automated process for updating the site content.

To begin with the Lab:

1. In your AWS Console navigate to S3 and create two buckets. So, one bucket will be for the source repository and another bucket for the production environment to host a simple static website.
2. Then, we will build a pipeline on AWS Code Pipeline so that whenever we upload a new version of our website to the source bucket as a zip package, the pipeline will be triggered, and it will deploy the package contents to the production S3 bucket.
3. Now go to your AWS Console and then navigate to S3. Here we are going to create two buckets. So, click on Create Bucket.



4. There are some crucial configurations you need to make for your bucket to be accepted as a source location by your pipeline. First, your source bucket must be in the same AWS region as your pipeline.
5. First, you need to give your bucket a name then scroll down.
6. Now, there is another requirement for S3 bucket source locations. You must enable versioning on your source bucket. This is because when you use an S3 bucket as the source location, Code Pipeline identifies source revisions from their S3 version IDs. After that just go ahead and create your bucket.

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

Europe (Ireland) eu-west-1

Bucket name Info

website-source-bucket1

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

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

- Disable
- Enable

7. Once our source bucket is created now, we are going to create our production bucket.

Amazon S3 > Buckets

► Account snapshot - updated every 24 hours All AWS Regions

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

[General purpose buckets](#) [Directory buckets](#)

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
website-source-bucket1	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	August 5, 2024, 20:13:57 (UTC+05:30)

8. So, unlike source buckets, production buckets can be in any AWS region but for simplicity, we will keep our region same.

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

Europe (Ireland) eu-west-1

Bucket name Info

website-production-bucket

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

9. Our production bucket will be used to host a static website so, we need to disable block public access. Then scroll down to the bottom and create your bucket.

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.

10. Now we need to go inside of our production bucket to update its bucket policy to allow its contents to be publicly readable from the Internet. So go to permission and scroll down to bucket policy click on edit then update the policy.

The screenshot shows the AWS S3 console with the path: Amazon S3 > Buckets > website-production-bucket. The 'Permissions' tab is selected. Under the 'Bucket policy' section, there is a JSON editor containing the following policy:

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

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::YOUR-BUCKET-NAME/*"
      ]
    }
  ]
}
```

11. Now you need to go to its properties and then scroll down to the bottom there you will find static website hosting you need to enable this feature. You can follow the snapshots below to enable this feature.

website-production-bucket Info

Objects Properties Permissions Metrics Management Access Points

Bucket overview

AWS Region
Europe (Ireland) eu-west-1

Amazon Resource Name (ARN)
`arn:aws:s3:::website-production-bucket`

Static website hosting

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

Edit

Static website hosting
Disabled

Static website hosting

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

Static website hosting

- Disable
 Enable

Hosting type

- Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
 Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

ⓘ For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

ⓘ For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document

Specify the home or default page of the website.

index.html

Error document - optional

This is returned when an error occurs.

error.html

12. Now we are going to upload the zip file of our static website onto our S3 source bucket. Below you can see that we have uploaded the static website zip file.

Note: You can get this zip file from the GitHub.

The screenshot shows the AWS S3 console interface. At the top, it displays the path: Amazon S3 > Buckets > website-source-bucket1. Below this, the bucket name 'website-source-bucket1' is shown with an 'info' link. A navigation bar with tabs like Objects, Properties, Permissions, Metrics, Management, and Access Points is visible. Under the Objects tab, there is a table with one item: 'my-website.zip'. The table includes columns for Name, Type, Last modified, Size, and Storage class. The file was last modified on August 5, 2024, at 20:30:52 (UTC+05:30) and has a size of 799.0 B with a Standard storage class.

13. Now you need to search for Code Pipeline in your AWS Console and choose the service accordingly.



14. Here click on Create Pipeline then choose to build a custom Pipeline and click on Next.

The screenshot shows the 'Choose creation option' step of the AWS CodePipeline creation wizard. On the left, a sidebar lists steps: Step 1 (Choose creation option), Step 2 (Choose pipeline settings), Step 3 (Add source stage), Step 4 (Add build stage), Step 5 (Add deploy stage), and Step 6 (Review). The main area is titled 'Choose creation option' with a sub-instruction 'Step 1 of 6'. It contains a section titled 'Creation options' with two choices: 'Create pipeline from template' (radio button not selected) and 'Build custom pipeline' (radio button selected). A note below says 'Build a pipeline from scratch to meet your specific needs.' At the bottom right are 'Cancel' and 'Next' buttons.

15. Now you just need to give your pipeline a unique name and keep other settings to default then scroll down to bottom.

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Pipeline type

 You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode

Choose the execution mode for your pipeline. This determines how the pipeline is run.

Superseded

A more recent execution can overtake an older one. This is the default.

Queued (Pipeline type V2 required)

Executions are processed one by one in the order that they are queued.

Parallel (Pipeline type V2 required)

Executions don't wait for other runs to complete before starting or finishing.

Service role

New service role

Create a service role in your account

Existing service role

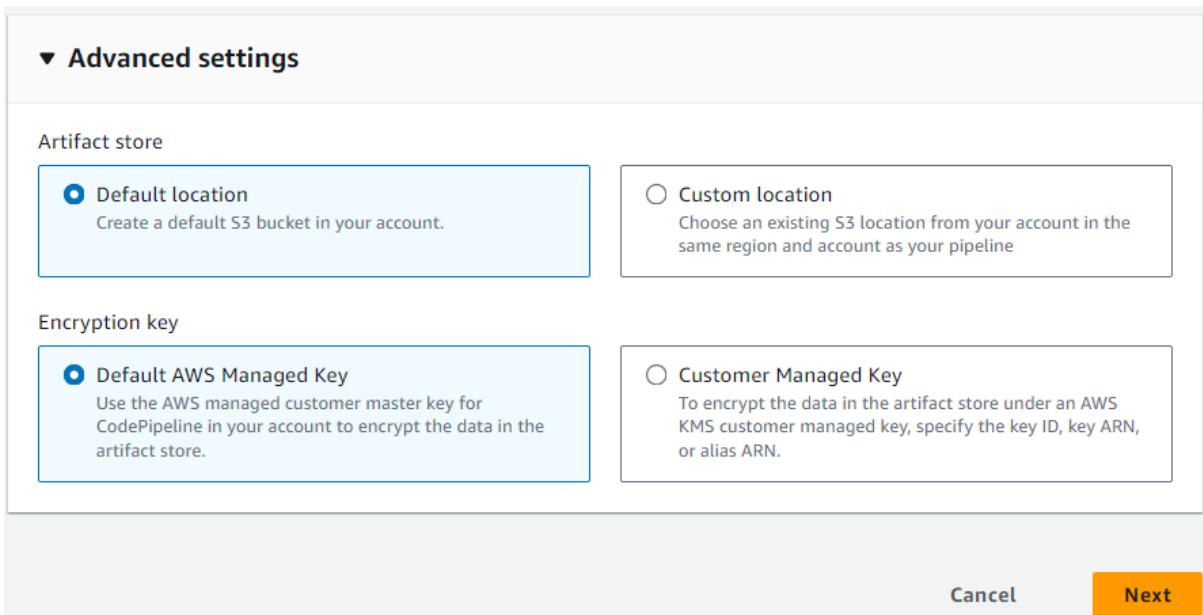
Choose an existing service role from your account

Role name

Type your service role name

Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

16. Then you need to expand the advanced settings options and here you will see that it will create an S3 bucket in our account to store artifacts. Also, we can give a custom location but let's leave it to default for now. Same for the Encryption key. Click on next.



17. Now you need to add a source provider, so choose Amazon S3 and then choose your source bucket. Then you need to give the name of your zip file in the S3 object key. Keep the change detection option to Amazon Cloud watch events and click on next.

Add source stage Info

Step 2 of 5

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

Amazon S3

Bucket
website-source-bucket1

S3 object key
my-website.zip

Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Cancel Previous Next

18. In the next option it is asking us to choose any build provider but we don't need any at this stage. So, choose the skip build stage and move to the next step.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

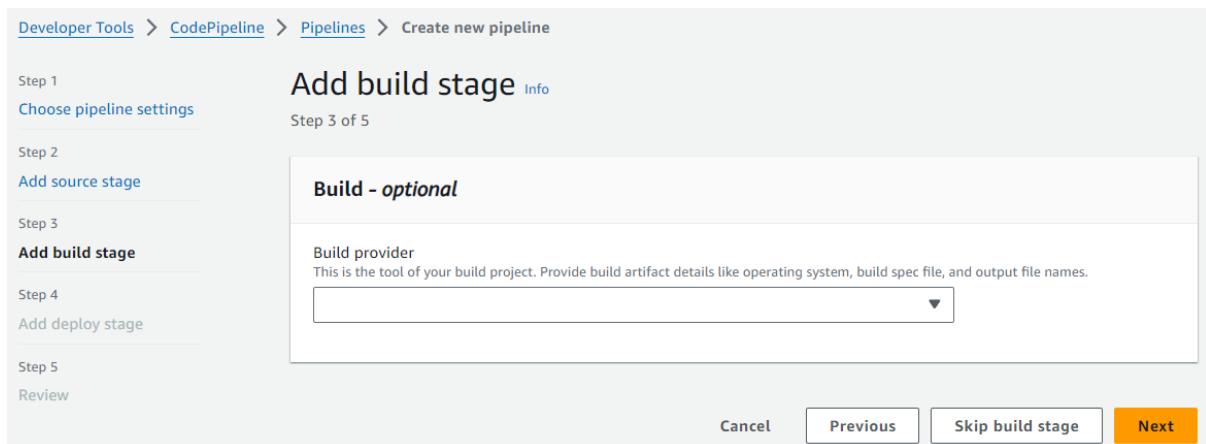
Add build stage Info

Step 3 of 5

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Cancel Previous Skip build stage Next



19. Now we need to choose a deploy provider, for that, you need to choose Amazon S3 and in the region choose the same region where you are creating the pipeline because we are not changing our region for now.
20. Then you need to choose the bucket and choose Extract file before deploying, click on next.
21. After moving to the review page create your pipeline.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

Amazon S3

Region
Europe (Ireland)

Bucket
website-production-bucket

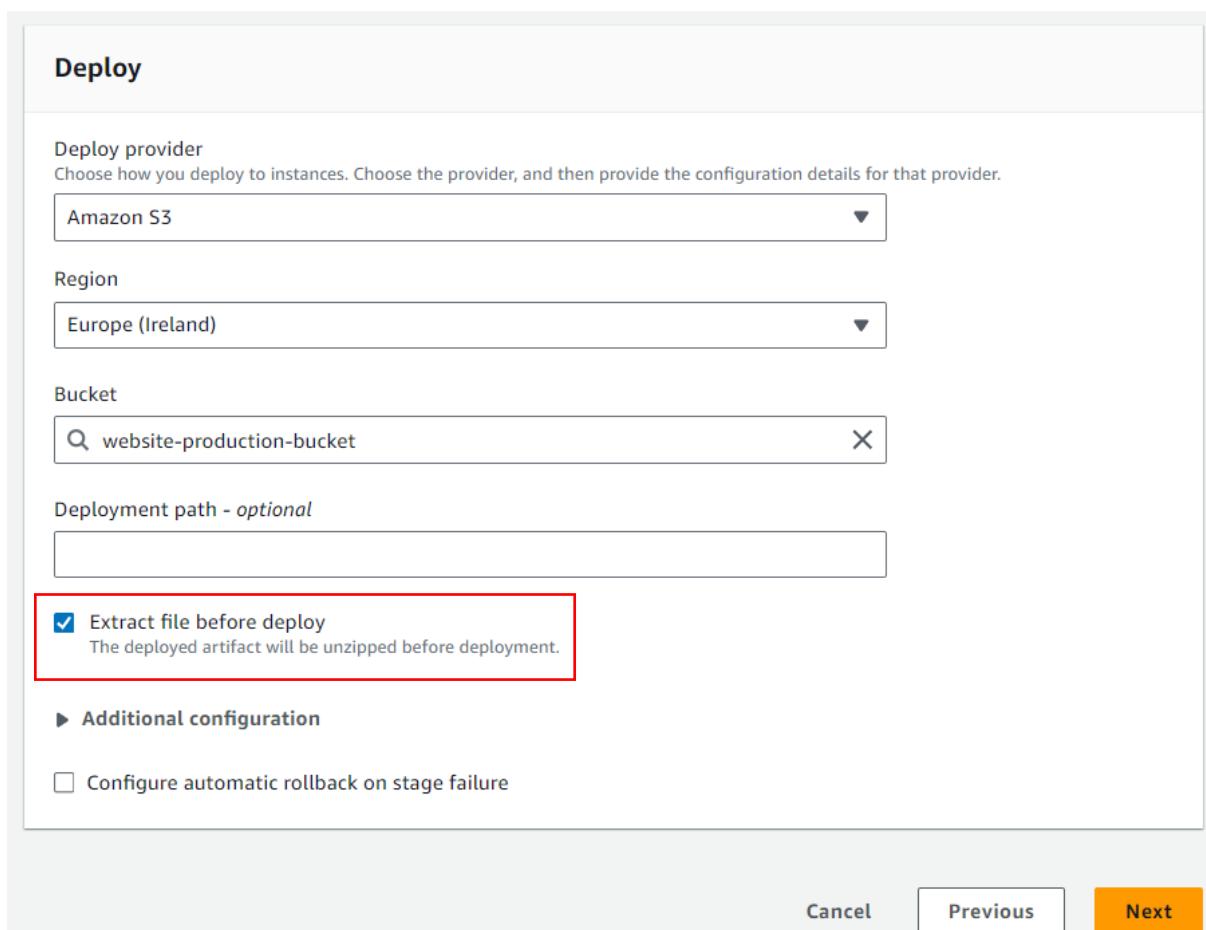
Deployment path - optional

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

► Additional configuration

Configure automatic rollback on stage failure

Cancel Previous Next



22. As you see, there are two large boxes named 'Source' and 'Deploy', which are connected with an arrow. These are the stages of our pipeline. When you create a pipeline, it is triggered automatically.

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded
Pipeline execution ID: [b7e8a30d-7cc4-4336-83cd-54e0bede7dbb](#)

Source
Amazon S3

Succeeded - 1 minute ago

[View details](#)

Source: Amazon S3 version id: mx_uAsp.FkApGUxCgiWYXIDVIRVs641h

[Disable transition](#)

Deploy Succeeded
Pipeline execution ID: [b7e8a30d-7cc4-4336-83cd-54e0bede7dbb](#)

Deploy
Amazon S3

Succeeded - 1 minute ago

[View details](#)

Source: Amazon S3 version id: mx_uAsp.FkApGUxCgiWYXIDVIRVs641h

23. If you return to S3, here you will see that we have a new bucket which is created by the code pipeline.

Amazon S3 > Buckets

► Account snapshot - updated every 24 hours (All AWS Regions)
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

General purpose buckets [Directory buckets](#)

General purpose buckets (3) [Info](#) All AWS Regions

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
codepipeline-eu-west-1-465484690132	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	August 5, 2024, 20:47:20 (UTC+05:30)
website-production-bucket	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	August 5, 2024, 20:19:38 (UTC+05:30)
website-source-bucket1	Europe (Ireland) eu-west-1	View analyzer for eu-west-1	August 5, 2024, 20:13:57 (UTC+05:30)

24. Now go inside our production bucket, you will see that we have both of our HTML files.

Amazon S3 > Buckets > website-production-bucket

website-production-bucket [Info](#)

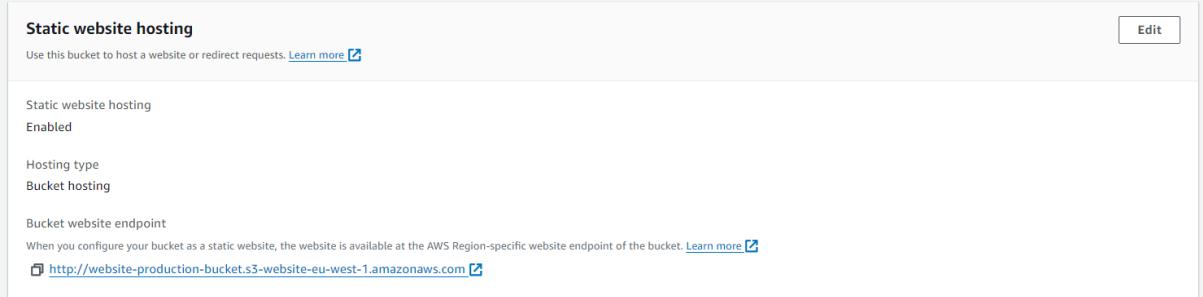
Objects Properties Permissions Metrics Management Access Points

Objects (2) [Info](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
error.html	html	August 5, 2024, 20:47:32 (UTC+05:30)	428.0 B	Standard
index.html	html	August 5, 2024, 20:47:32 (UTC+05:30)	405.0 B	Standard

25. Now go to properties and scroll down to the bottom here you will see the link to visit your website click on it.



The screenshot shows the 'Static website hosting' configuration for a bucket named 'website-production-bucket'. It includes sections for 'Static website hosting' (Enabled), 'Hosting type' (Bucket hosting), and 'Bucket website endpoint' (http://website-production-bucket.s3-website-eu-west-1.amazonaws.com). An 'Edit' button is visible in the top right corner.

26. Below you can see that our website is working fine and it was deployed successfully.

Welcome to Our Website!

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

Website Version: 1.0