



# Módulo 4: Front-end

## Sesión 2: Despliegue en AWS

**Equipo de desarrolladores:**

Cristian David Ríos MSc  
Daniel Escobar Grisales MSc  
Nestor Rafael Calvo MSc

**Coordinador del proyecto:**

Prof. Dr.-Ing. Juan Rafael Orozco Arroyave



# Hola!

## Mi nombre es Cristian Ríos

Puedes encontrarme como:

 @cdavidrios

 @cdavid-rios

# Agenda

- Servicios que se usarán en el despliegue en AWS
- CodeCommit
  - ¿Qué es CodeCommit?
  - Implementación
- Simple Storage Service (S3)
  - ¿Qué es S3?
  - Implementación
  - URL S3

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines, with some nodes highlighted in blue and others in grey.

1.

# Servicios AWS

# Servicios AWS



## CodeCommit

CodeCommit es un servicio de control de código fuente seguro que aloja repositorios de Git privados en AWS.



## CodePipeline

CodePipeline es un servicio de entrega continua que permite automatizar canalizaciones de lanzamiento.



## Simple Storage Service

S3 es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos y seguridad.



## CloudFront

CloudFront es un servicio de entrega de contenido que distribuye datos, vídeos, aplicaciones y API a clientes de todo el mundo de forma segura.



# 2.

## CodeCommit

# ¿Qué es CodeCommit?

- ◎ CodeCommit es un servicio de control de código fuente seguro, administrado y de alta escalabilidad que aloja repositorios de Git privados en AWS
- ◎ Es compatible con la funcionalidad estándar de Git
- ◎ Será el repositorio donde almacenaremos nuestro código fuente para tener una integración continua



# Creación y enlace con CodeCommit

Inicialmente, es necesario crear un repositorio privado en AWS, para esto es necesario buscar en los servicios CodeCommit y darle la opción “Create repository”

Developer Tools > CodeCommit > Repositories > Create repository

## Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

### Repository settings

Repository name

100 characters maximum. Other limits apply.

Description - optional

1,000 characters maximum

Tags

Add

☐ Enable Amazon CodeGuru Reviewer for Java and Python - optional

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

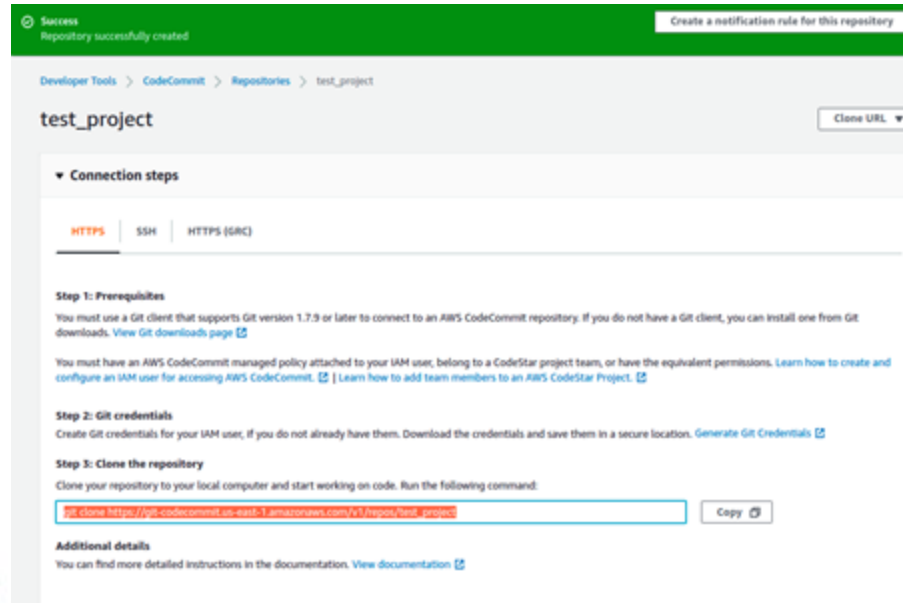
A service-linked role will be created in IAM on your behalf if it does not exist.

Cancel Create



# Creación y enlace con CodeCommit

Luego, se debe clonar el repositorio en su computador para tener una comunicación directa desde su computador a AWS.



The screenshot displays the AWS CodeCommit console for a repository named 'test\_project'. At the top, a green banner indicates 'Success: Repository successfully created' with a button to 'Create a notification rule for this repository'. Below this, the breadcrumb trail shows 'Developer Tools > CodeCommit > Repositories > test\_project'. The repository name 'test\_project' is prominently displayed with a 'Clone URL' button. The 'Connection steps' section is expanded, showing three tabs: 'HTTPS' (selected), 'SSH', and 'HTTPS (SRC)'. Under 'Step 1: Prerequisites', it states that a Git client version 1.7.9 or later is required and provides a link to 'View Git downloads page'. It also mentions the need for an AWS CodeCommit managed policy and provides a link to 'Learn how to add team members to an AWS CodeStar Project'. 'Step 2: Git credentials' instructs users to create credentials for their IAM user and provides a link to 'Generate Git Credentials'. 'Step 3: Clone the repository' provides the command to clone the repository and includes a text box with the URL 'git clone https://git-codecommit.us-east-1.amazonaws.com/v3/repos/test-project' and a 'Copy' button. Finally, the 'Additional details' section provides a link to 'View documentation'.

Success  
Repository successfully created

Create a notification rule for this repository

Developer Tools > CodeCommit > Repositories > test\_project

test\_project Clone URL

▼ Connection steps

HTTPS SSH HTTPS (SRC)

**Step 1: Prerequisites**  
You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. [View Git downloads page](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. [Learn how to create and configure an IAM user for accessing AWS CodeCommit](#) | [Learn how to add team members to an AWS CodeStar Project](#)

**Step 2: Git credentials**  
Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. [Generate Git Credentials](#)

**Step 3: Clone the repository**  
Clone your repository to your local computer and start working on code. Run the following command:

`git clone https://git-codecommit.us-east-1.amazonaws.com/v3/repos/test-project` Copy

**Additional details**  
You can find more detailed instructions in the documentation. [View documentation](#)

# Creación y enlace con CodeCommit

Además de su código fuente, es importante añadir un archivo que nos permitirá compilar la aplicación usando el servicio de CodePipeline de AWS. El archivo debe tener el nombre de “buildspec.yml” y debe contener el código mostrado a continuación:

```
buildspec.yml x
buildspec.yml
1 # Specifies what build spec version this file is.
2 # This helps AWS CodePipeline parse the file correctly.
3 # Keep this at 0.2
4 version: 0.2
5 # We can listen for specific phases and execute commands per phase.
6 phases:
7   # The build server won't have access to our node modules folder
8   # This is because we have it inside of our .gitignore file
9   # To give our build server access, we can simply run "npm install"
10  pre_build:
11    commands:
12      - npm install
13  # Now we want to actually build our React app
14  build:
15    commands:
16      - npm run build
17  # Artifacts will specify what files will be uploaded to s3
18  # This will include all files within the "build" folder
19  artifacts:
20    files:
21      - '**/*'
22  discard-paths: no
23  base-directory: build
24
```

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines, with some nodes highlighted in blue and others in grey.

# 2.

## **Simple Storage Service**

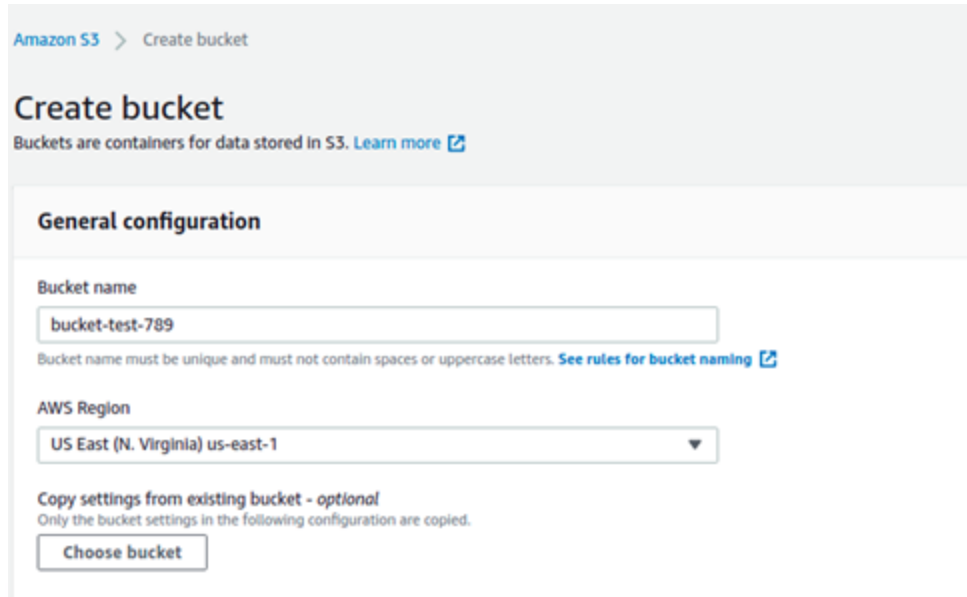
# ¿Qué es Simple Storage Service?

- ◎ Simple Storage Service (S3) es un servicio de almacenamiento estático que ofrece escalabilidad, disponibilidad de datos y seguridad
- ◎ Nosotros usaremos S3 para almacenar y observar el sitio web estático con código fuente almacenado en CodeCommit
- ◎ Para este caso, el servicio de CodePipeline será el encargado de compilar este contenido estático y llevarlo a S3



# Creación y configuración del bucket

Inicialmente, debemos ir al servicio S3 en AWS y luego darle click al botón “Create bucket”. La siguiente figura muestra la configuración general del bucket



The screenshot displays the 'Create bucket' interface in the Amazon S3 console. At the top, the breadcrumb 'Amazon S3 > Create bucket' is visible. The main heading is 'Create bucket', followed by a subtext: 'Buckets are containers for data stored in S3. [Learn more](#)'. The 'General configuration' section is highlighted and contains the following elements:

- Bucket name:** A text input field containing 'bucket-test-789'. Below the field, a note states: 'Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)'.
- AWS Region:** A dropdown menu currently showing 'US East (N. Virginia) us-east-1'.
- Copy settings from existing bucket - optional:** A section with the text 'Only the bucket settings in the following configuration are copied.' and a button labeled 'Choose bucket'.

# Creación y configuración del bucket

Luego, es necesario configurar el acceso para el bucket, en este caso debemos desbloquear el acceso a todo el público.

## Block Public Access settings for 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.

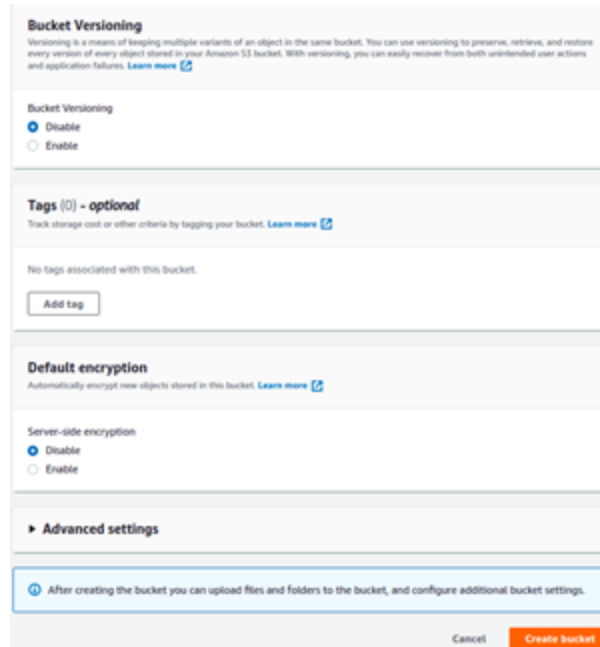


Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

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

# Creación y configuración del bucket

Ahora, configuramos el versionamiento del bucket y cifrado de la información. En nuestro caso, no trabajamos con versiones, ya que estas se manejan desde CodeCommit.



**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

**Tags [0] - optional**  
Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

Add tag

**Default encryption**  
Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption  
☒ Disable  
☐ Enable

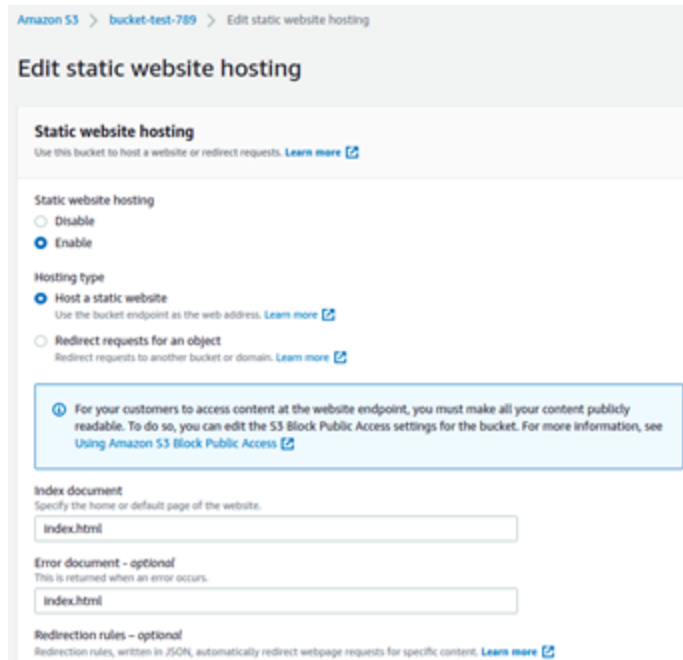
► **Advanced settings**

① After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel Create bucket

# Creación y configuración del bucket

Luego, la figura a continuación muestra la configuración de usar el bucket como host de un contenido estático, por lo tanto esta configuración la debemos habilitar.



Amazon S3 > bucket-test-789 > Edit static website hosting

## Edit static website hosting

**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](#)

**Index document**  
Specify the home or default page of the website.

**Error document - optional**  
This is returned when an error occurs.

**Redirection rules - optional**  
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)



# Creación y configuración del bucket

Finalmente, solo es darle al botón “Create bucket” y ya tenemos nuestro bucket con la configuración inicial de nuestro sitio web

Ahora queda hacer 2 ajustes para permitir el acceso a nuestro contenido. Inicialmente, nos iremos para nuestro bucket y luego para la pestaña “Permissions”, en esta pestaña configuraremos 2 permisos especiales para una correcta visualización de nuestro contenido

El primero está relacionado con las políticas del bucket y el segundo corresponde a la configuración de CORS

# Creación y configuración del bucket

## Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

Edit

Delete

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::bucket-test-789/*"
    }
  ]
}
```

Copy

## Cross-origin resource sharing (CORS)

The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. [Learn more](#)

Edit

```
{
  {
    "AllowedHeaders": [
      ""
    ],
    "AllowedMethods": [
      "GET",
      "POST"
    ],
    "AllowedOrigins": [
      ""
    ],
    "ExposeHeaders": []
  }
}
```

Copy

# URL del bucket

Finalmente, para conocer la URL expuesta por S3 debemos dirigirnos a la pestaña “Properties”. La siguiente figura muestra un pantallazo del ejemplo desarrollado.

