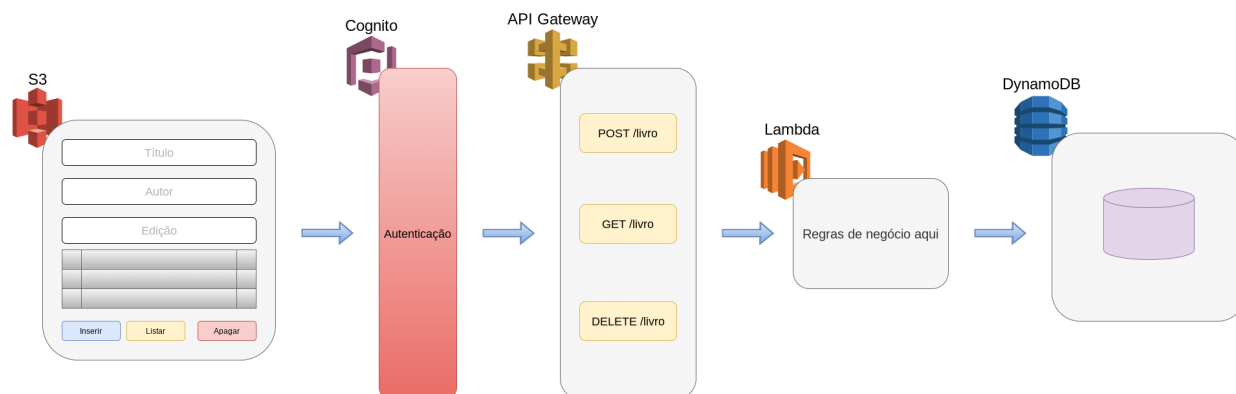


1 Introdução

Neste material, prosseguimos com o desenvolvimento da solução retratada a seguir.



2 Desenvolvimento

Lembre-se que temos interesse na seguinte API (coleção de endpoints).

POST /livros (cadastrar um livro novo)

GET /livros (obter todos os livros)

GET /livros/{id} (obter um livro pelo seu id)

PUT /livros/{id} (atualizar um livro pelo seu id)

DELETE /livros/{id} (apagar um livro pelo seu id)

Neste material, vamos implantar uma versão da aplicação Front End que utiliza apenas os endpoints

GET /livros (obter todos os livros)

GET /livros/{id} (obter um livro pelo seu id)

A implantação do Front End será realizada utilizando-se o **AWS S3**. Veja algumas características sobre ele.

Modelo de Dados de Objeto: O S3 armazena dados como objetos dentro de "**buckets**". Um objeto consiste em dados, um ID exclusivo e metadados.

Durabilidade e Disponibilidade: O S3 oferece uma durabilidade de 99,999999999% (11 9's) e garante 99,99% de disponibilidade durante um ano.

Classes de Armazenamento: O S3 oferece várias classes de armazenamento, incluindo STANDARD, INTELLIGENT_TIERING, ONEZONE_IA, GLACIER e GLACIER_DEEP_ARCHIVE. Cada classe tem seu próprio preço e uso recomendado com base na frequência de acesso e no período de retenção dos dados.

Veja uma descrição sobre as classes de armazenamento.

S3 Standard (STANDARD):

Uso: Projetado para uso geral e armazenamento de dados frequentemente acessados.

Durabilidade: 99,999999999% (11 9's) ao longo de um ano.

Disponibilidade: 99,99% ao longo de um ano.

Recuperação: Recuperação em tempo real.

Custo: Geralmente maior do que os demais.

S3 Intelligent-Tiering (INTELLIGENT_TIERING):

Uso: Ideal para dados com padrões de acesso desconhecidos ou que mudam com o tempo.

Durabilidade: 99,999999999%.

Disponibilidade: 99,9%.

Recuperação: Em tempo real.

Custo: Taxas de monitoramento e automação são aplicadas, mas geralmente é mais barato do que o STANDARD para dados com padrões de acesso variáveis.

S3 Standard-Infrequent Access (STANDARD_IA):

Uso: Para dados menos acessados, mas que ainda precisam de recuperação rápida quando necessários.

Durabilidade: 99,999999999%.

Disponibilidade: 99,9%.

Recuperação: Em tempo real.

Custo: Menor custo de armazenamento por GB em comparação com o STANDARD, mas com taxas de recuperação.

S3 One Zone-Infrequent Access (ONEZONE_IA):

Uso: Para dados que podem ser recriados e são acessados com menos frequência, mas que ainda precisam de recuperação rápida.

Durabilidade: 99,999999999%, mas armazenados em apenas uma zona de disponibilidade, portanto, menos resiliência a falhas em comparação com outras classes.

Disponibilidade: 99,5%.

Recuperação: Em tempo real.

Custo: Menor do que o STANDARD_IA, pois utiliza apenas uma zona.

S3 Glacier (GLACIER):

Uso: Arquivamento de dados de longo prazo que podem tolerar tempo de recuperação de algumas horas.

Durabilidade: 99,999999999%.

Disponibilidade: Não é imediata; a recuperação leva de alguns minutos a várias horas.

Recuperação: De minutos a horas, dependendo do nível de recuperação escolhido.

Custo: Muito mais baixo do que classes de armazenamento em tempo real, mas com taxas de recuperação.

S3 Glacier Deep Archive (GLACIER_DEEP_ARCHIVE):

Uso: Arquivamento de dados de longo prazo que são acessados muito raramente.

Durabilidade: 99,999999999%.

Disponibilidade: Não é imediata; a recuperação leva cerca de 12 horas.

Recuperação: Em torno de 12 horas.

Custo: A classe de armazenamento mais barata no S3, mas com taxas de recuperação.

Estas classes permitem que os usuários otimizem custos, mantendo a durabilidade e disponibilidade necessárias para seus dados. Ao escolher uma classe de armazenamento, é importante considerar a frequência de acesso, o tempo de retenção desejado e o orçamento disponível.

Modelo de Segurança: Você pode controlar o acesso aos buckets e objetos usando a AWS Identity and Access Management (IAM), controlar o acesso público e usar políticas de bucket. Além disso, oferece recursos de criptografia para proteger os dados em trânsito e em repouso.

Transfer Acceleration: Esta funcionalidade usa a rede global da Amazon CloudFront para acelerar os uploads e downloads de objetos para e do S3. Como veremos, funciona como um **CDN**.

Versionamento: Permite preservar, recuperar e restaurar todas as versões de todos os objetos em um bucket. Isso é útil para recuperação de desastres e histórico.

Eventos: É possível configurar notificações para serem acionadas em resposta a determinados eventos no S3, como a criação ou exclusão de objetos.

Replicação: Você pode configurar a replicação automática e assíncrona de objetos para um bucket diferente, potencialmente em uma região AWS diferente.

Hospedagem de Sites Estáticos: Os buckets do S3 podem ser configurados para hospedar sites estáticos sem a necessidade de servidores web tradicionais.

2.1 Obtendo a aplicação Front End A aplicação Front se encontra no seguinte repositório no Github

https://github.com/professorbossini/pessoal_react_aws_livros

Crie uma pasta vazia para você e vincule o VS Code a ela clicando em File >> Open Folder. Depois disso, abra um terminal interno dele com Terminal >> New Terminal. No terminal, use

git clone https://github.com/professorbossini/pessoal_react_aws_livros.git .

para clonar o repositório. Para testar a aplicação localmente, comece baixando as dependências com

```
npm install
```

E execute com

```
npm start
```

Observe que a aplicação já “vem” com uma URL de Back End fixa. A ideia é que você possa encaixar a sua URL ali para testar o seu Back End. Clique em Obter todos para visualizar a lista de livros atual.

Cuidado. É possível que a API cujo link de acesso vem fixo no código não esteja disponível quando você tentar fazer o teste.

ID	Título	Descrição	Autor

Cadastrar

Atualizar

Remover

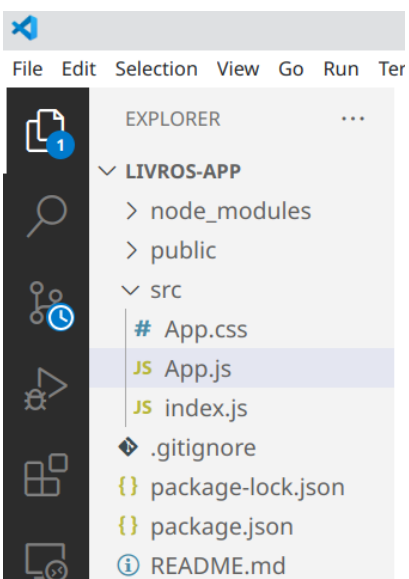
Obter pelo id

Obter todos

<https://ktq5lo2ci.execute-api.us-east-1.amazonaws.com/dev>

Algorithms IV	Cormen
ABC	ABC
Algorithms II	Cormen

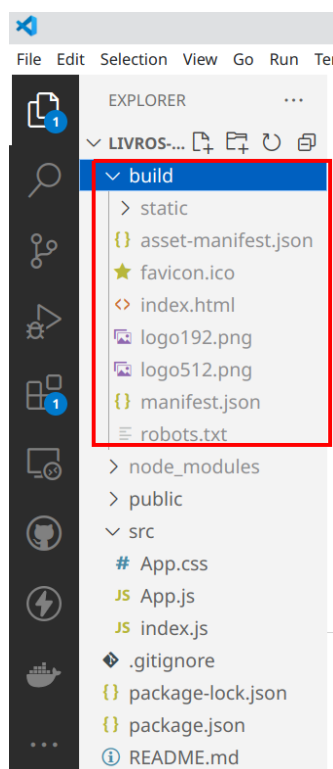
Uma aplicação React executa puramente no Front End. Observe como, em ambiente de desenvolvimento, ela possui uma estrutura própria para esse ambiente.



Ocorre que o navegador apenas “entende” HTML, CSS, Javascript e arquivos de recursos como áudio, vídeo etc. Assim, quando a aplicação está pronta para ser implantada, executamos um script que produz esse conteúdo em função daquilo que foi desenvolvido. No caso de uma aplicação React, o comando é

```
npm run build
```

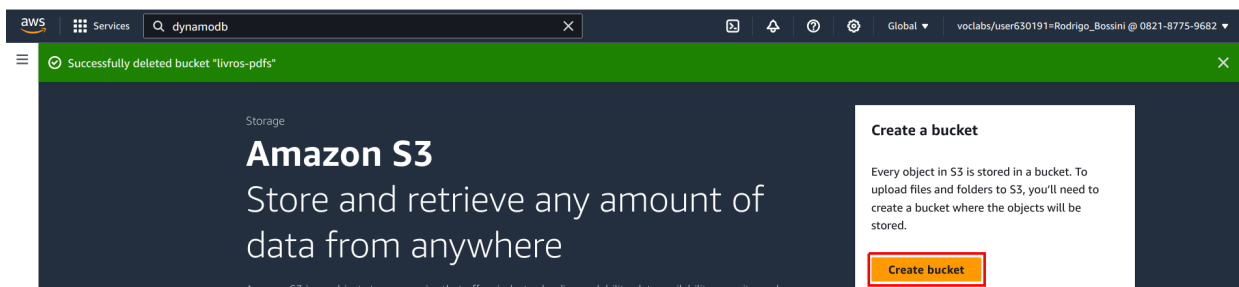
Observe como este comando criou uma pasta chamada build



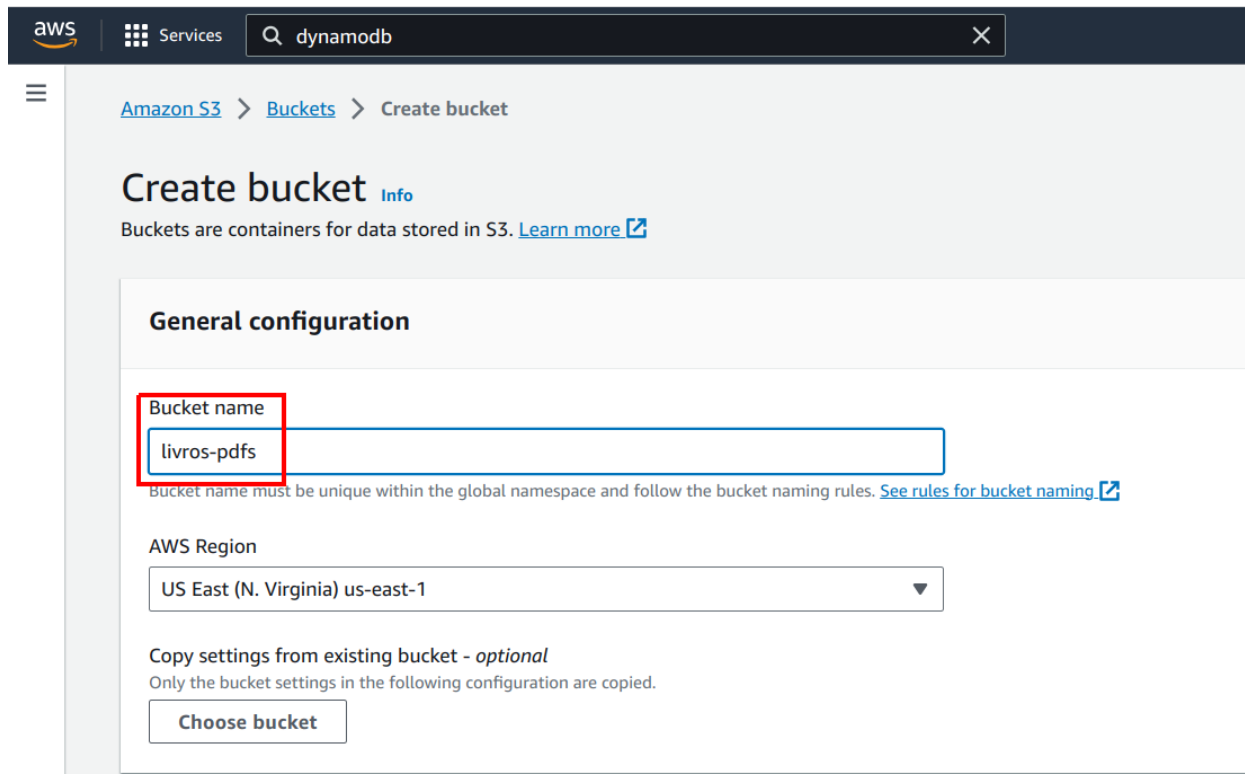
Esta coleção de arquivos é a nossa aplicação Front End. Agora vamos configurar um ambiente no AWS S3 para fazer a sua implantação. Como você verá, será tão simples quanto copiar e colar todos esses arquivos.

2.2 Configurando um Bucket no AWS S3 A fim de implantar a aplicação, vamos criar um Bucket no S3. Um Bucket pode ser utilizado para armazenar arquivos e também pode ser configurado para operar como um servidor web de conteúdo estático.

Na página inicial do S3, clique em **Create bucket**.



Comece escolhendo um nome para o seu Bucket.



The screenshot shows the AWS Management Console 'Create bucket' page. The breadcrumb navigation is 'Amazon S3 > Buckets > Create bucket'. The main heading is 'Create bucket' with an 'Info' link. Below it, a note states 'Buckets are containers for data stored in S3.' with a 'Learn more' link. The 'General configuration' section contains the following fields:

- Bucket name:** A text input field containing 'livros-pdfs'. A red rectangular box highlights this field. Below the input, a note states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [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 'Choose bucket' button.

Ajuste as configurações de permissão de acesso como a seguir. Em particular, quando desmarcamos **Block all public access**, estamos permitindo que o bucket seja acessado externamente. Entretanto, também precisaremos dizer explicitamente como o seu conteúdo pode ser acessado (talvez em modo de leitura ou modo de escrita).

dynamodb

X

ACLs disabled (recommended)

All objects in this bucket are owned by this account.
Access to this bucket and its objects is specified using
only policies.

ACLs enabled

Objects in this bucket can be owned by other AWS
accounts. Access to this bucket and its objects can be
specified using ACLs.

Object Ownership

Bucket owner enforced

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.

Warning icon

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

Mantenha as demais opções com seu valor padrão e clique em **Create bucket**.

Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption [Info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

- ☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)
- ☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- ☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- ☐ Disable
- ☒ Enable

Advanced settings

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

Cancel **Create bucket**

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

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](#)

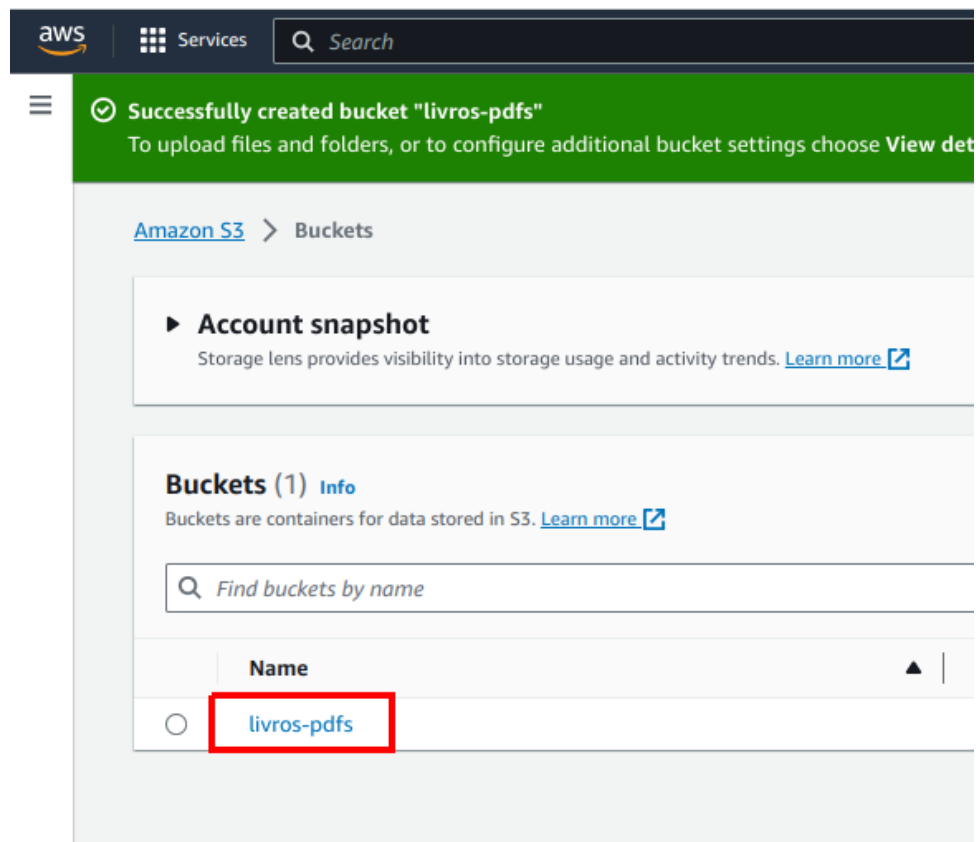
☐ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☐ **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.

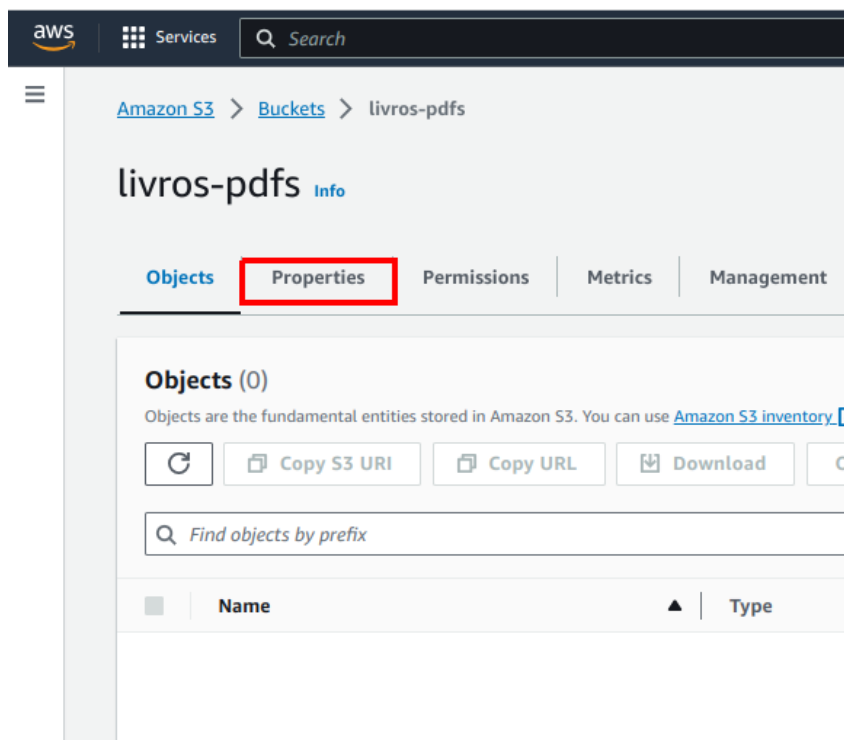
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

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

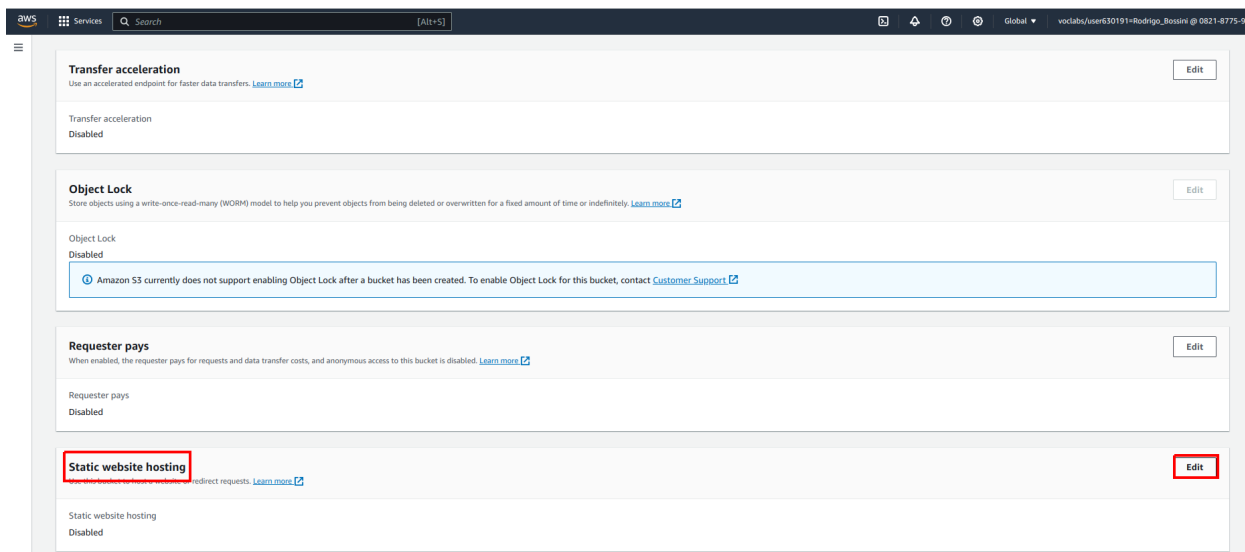
Na tela resultante, clique no nome do seu bucket para ajustar as suas configurações.



No momento, nosso bucket apenas serve para armazenar arquivos. Ele ainda não opera como um servidor Web estático. Vamos ajustar isso, Comece clicando em **Properties**.



Role a página até encontrar a opção **Static website hosting** e clique em **Edit**.



Marque a opção **Enable**. Preencha o campo **Index document** com **index.html** (esse é o nome do arquivo inicial gerado pelo quando fizemos o build da nossa aplicação) e clique em **Save changes**.

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](#)

Index document
The 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](#)

1

JSON Ln 1, Col 1 Errors: 0 Warnings: 0

Cancel **Save changes**

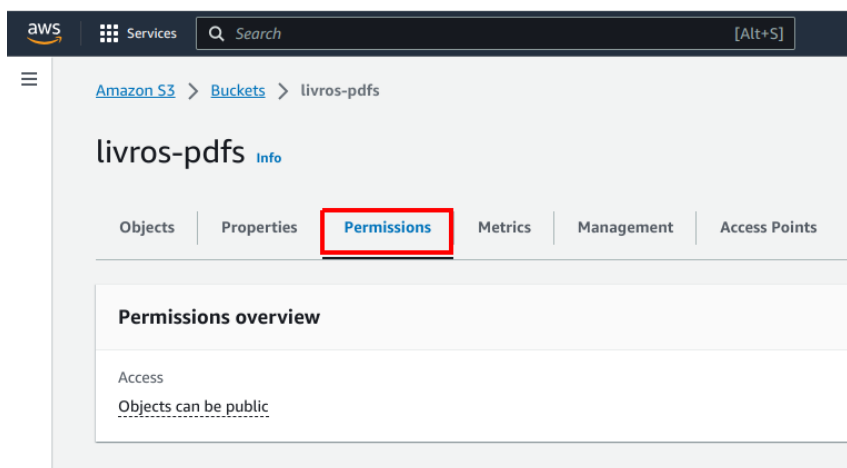
No próximo passo, vamos configurar permissões de acesso aos objetos dentro do Bucket. Veja um trecho da documentação a esse respeito.

“With Amazon S3 bucket policies, you can secure access to objects in your buckets, so that only users with the appropriate permissions can access them. You can even prevent authenticated users without the appropriate permissions from accessing your Amazon S3 resources.”

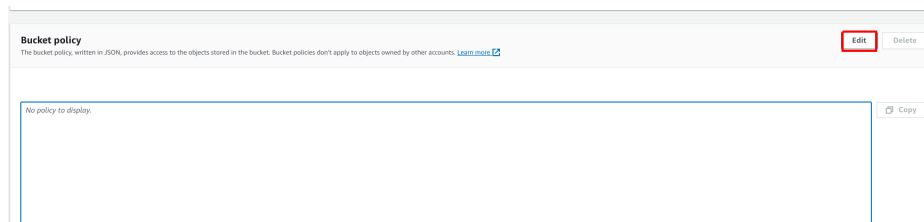
Leia mais em

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies.html>

Neste exemplo, vamos dizer que todos os objetos do bucket têm acesso “somente leitura”. Clique em **Permissions**.



Encontre o campo **Bucket policy** clique em **Edit**.



Agora adicione o seguinte conteúdo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::livros-pdfs/*"
    }
  ]
}
```

Cuidado. Lembre-se de especificar o nome do seu bucket, como destacado acima.

Veja uma explicação para cada chave deste objeto JSON que define a policy.

Version: versão da linguagem de política que estamos utilizando. É importante pois, ao longo do tempo, novas versões podem ser liberadas e seu funcionamento pode ser diferente.

Statement: Fica associado a uma coleção de declarações. Cada declaração é um objeto que caracteriza uma permissão.

Sid: Significa **Statement Id** e é **opcional**. É uma espécie de rótulo que pode te ajudar a lembrar sobre a razão de ser deste statement.

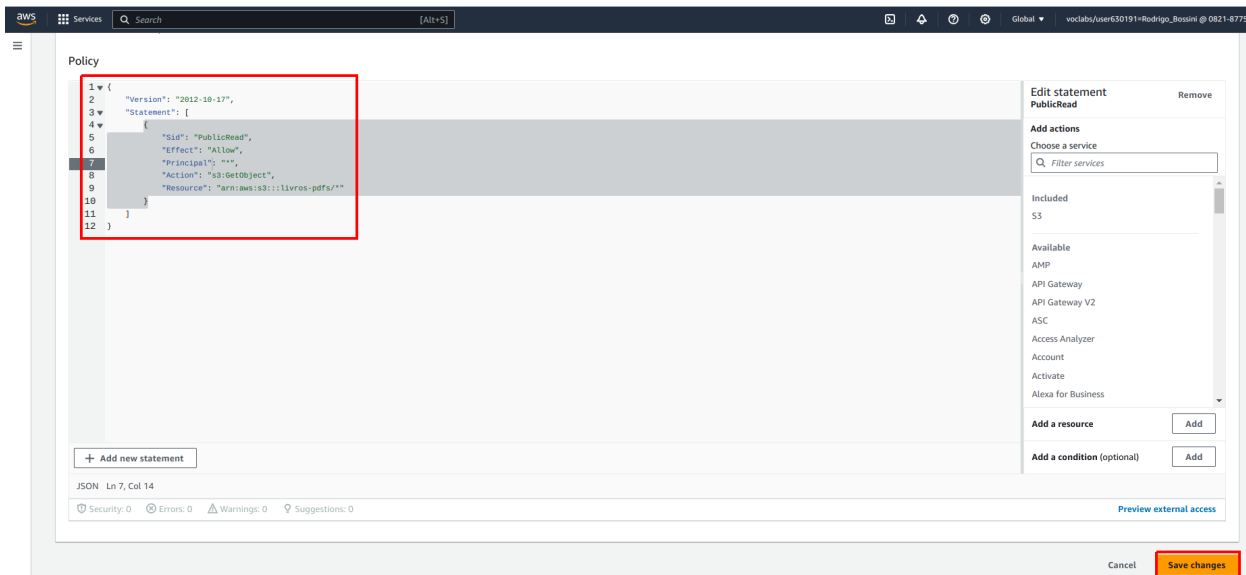
Effect: Especifica se este objeto define permissão ou negação de acesso. Os valores possíveis são **allow** e **deny**.

Principal: Especifica quais usuários estão envolvidos nesta declaração. O asterisco indica todos, autenticados ou não.

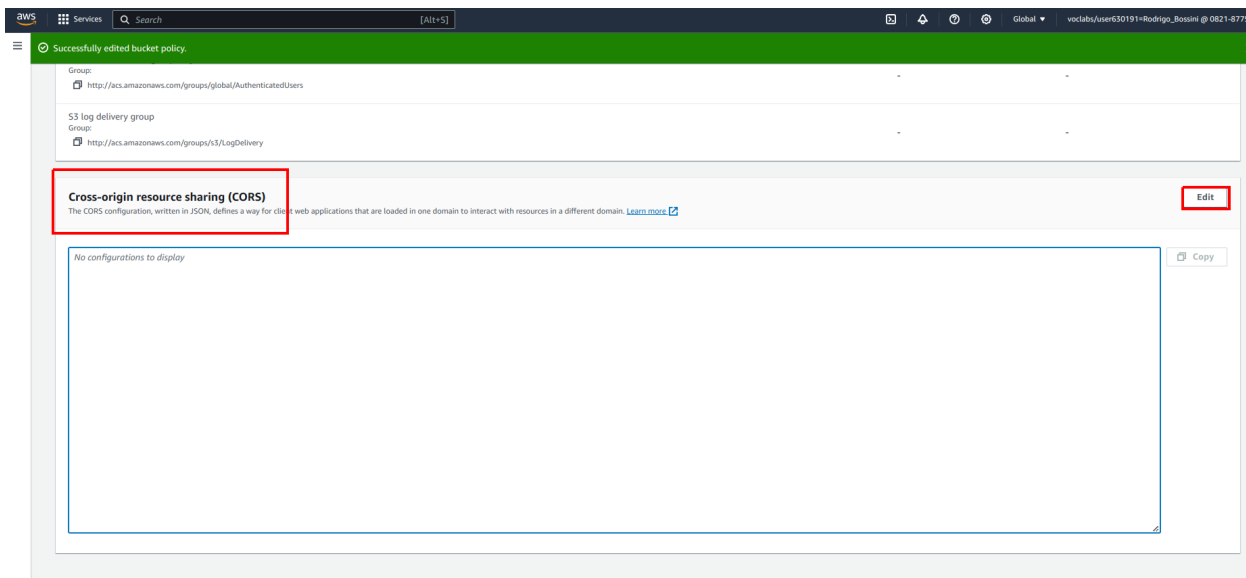
Action: Ação para a qual a permissão está sendo especificada. Valores possíveis são: s3:CreateBucket, s3>DeleteBucket, s3:GetObject, s3>DeleteObject. E assim por diante. Observe que temos ações envolvendo o bucket e ações que envolvem objetos dentro do bucket.

Resource: Qual recurso está envolvido nesta declaração **ARN** vem de **Amazon Resource Name** e a notação arn:aws:s3::: é um padrão Amazon para identificar recursos do S3.

Quando terminar, clique em **Save changes**.



A seguir, role a página e encontre a opção para configuração de CORS. Clique em **Edit**.



Use o seguinte objeto JSON para liberar o acesso sem restrições.

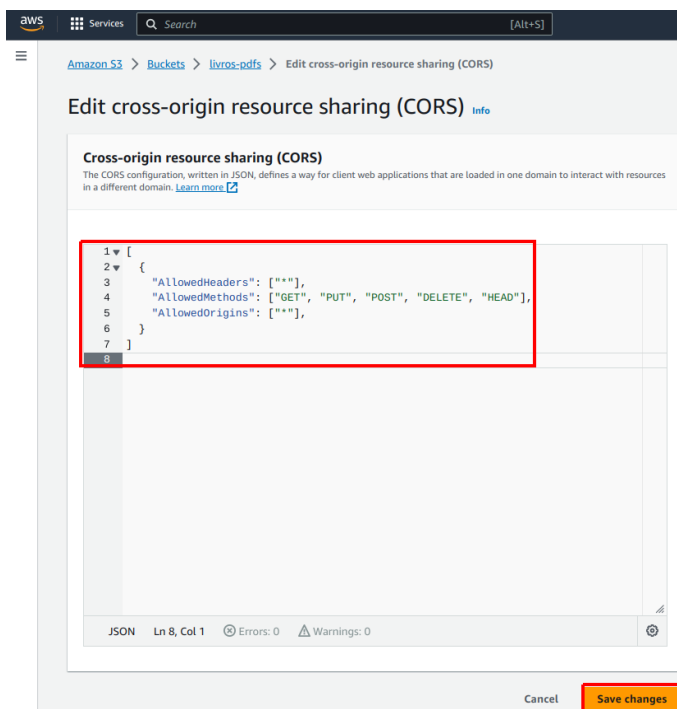
```
[
  {
    "AllowedHeaders": [ "*" ],
    "AllowedMethods": [ "GET", "PUT", "POST", "DELETE", "HEAD" ],
    "AllowedOrigins": [ "*" ]
  }
]
```

Veja uma explicação para cada chave neste objeto JSON. E qualquer caso, o asterisco simboliza “tudo”.

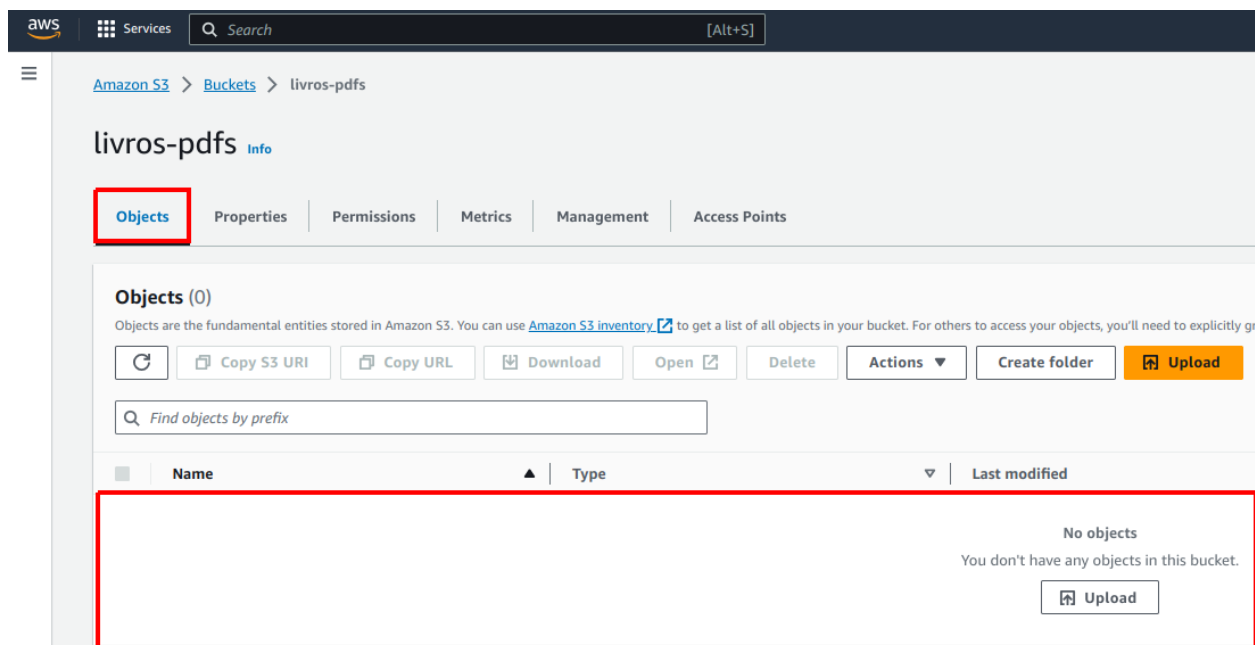
AllowedHeaders: Quais cabeçalhos podem estar presentes na requisição. Valores possíveis são “Authentication”, “Content-type” e assim por diante.

AllowedMethods: Quais métodos do protocolo HTTP pode ser usados.

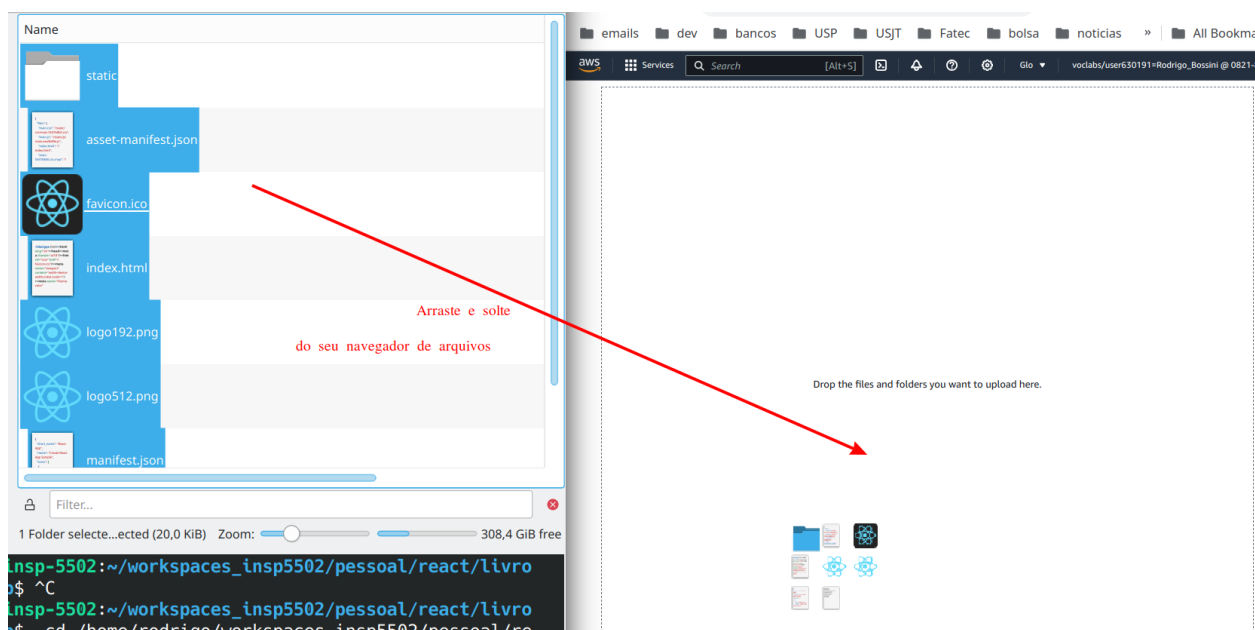
AllowedOrigins: A partir de quais origens requisições podem ser atendidas. Lembrando que uma origem é caracterizada pelo protocolo, host e porta.



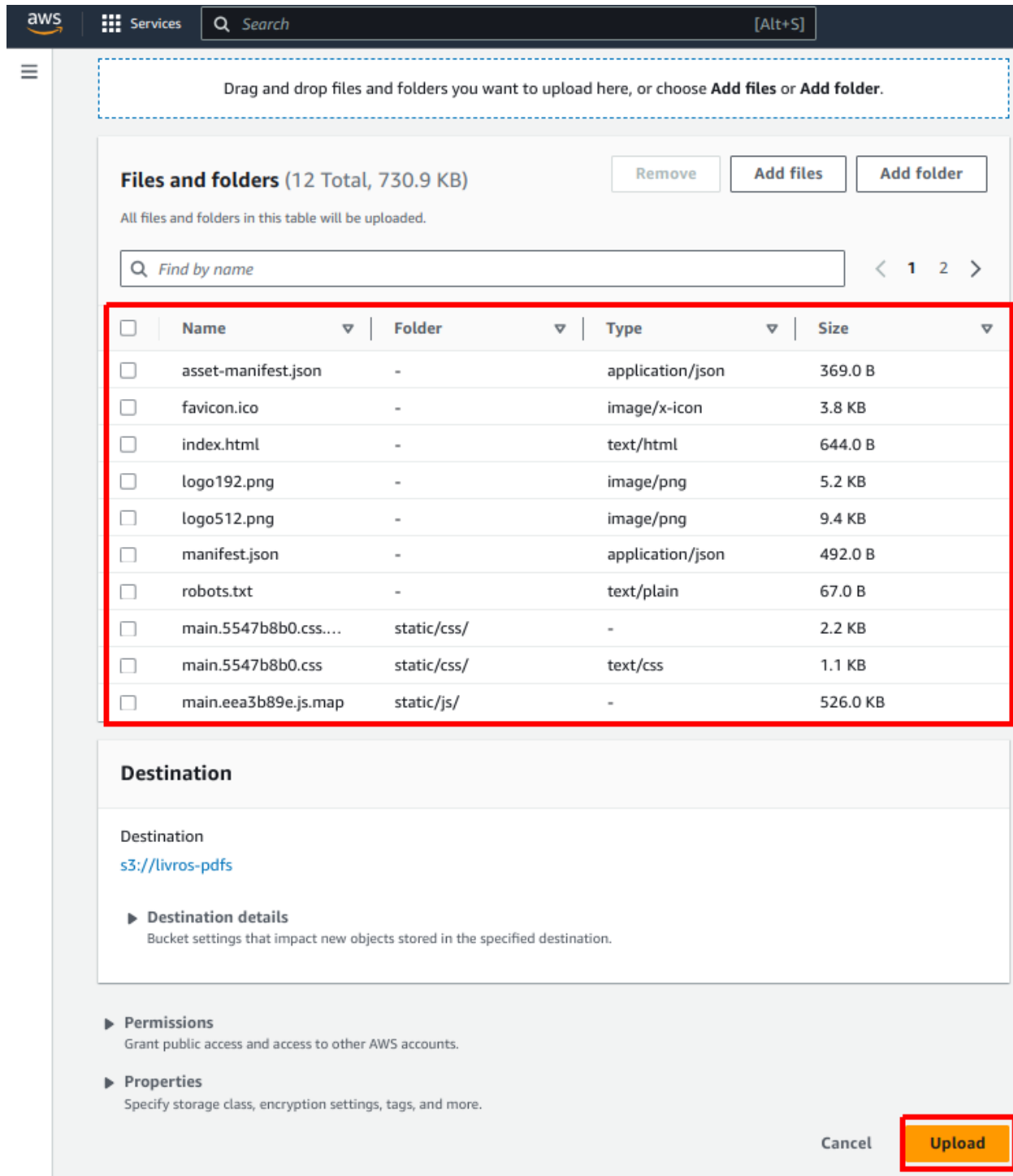
Para fazer o upload dos arquivos da sua aplicação, volte à aba **Objects**. Observe que o campo inferior permite que você arraste e solte arquivos.



Arraste e solte os arquivos da sua pasta **build**. Não arraste a pasta, apenas o seu conteúdo.



Observe que o ambiente mostra a lista de arquivos que vai fazer parte do upload. Clique em **Upload**.



Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (12 Total, 730.9 KB) Remove Add files Add folder

All files and folders in this table will be uploaded.

Find by name < 1 2 >

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	asset-manifest.json	-	application/json	369.0 B
<input type="checkbox"/>	favicon.ico	-	image/x-icon	3.8 KB
<input type="checkbox"/>	index.html	-	text/html	644.0 B
<input type="checkbox"/>	logo192.png	-	image/png	5.2 KB
<input type="checkbox"/>	logo512.png	-	image/png	9.4 KB
<input type="checkbox"/>	manifest.json	-	application/json	492.0 B
<input type="checkbox"/>	robots.txt	-	text/plain	67.0 B
<input type="checkbox"/>	main.5547b8b0.css....	static/css/	-	2.2 KB
<input type="checkbox"/>	main.5547b8b0.css	static/css/	text/css	1.1 KB
<input type="checkbox"/>	main.eea3b89e.js.map	static/js/	-	526.0 KB

Destination

Destination
s3://livros-pdfs

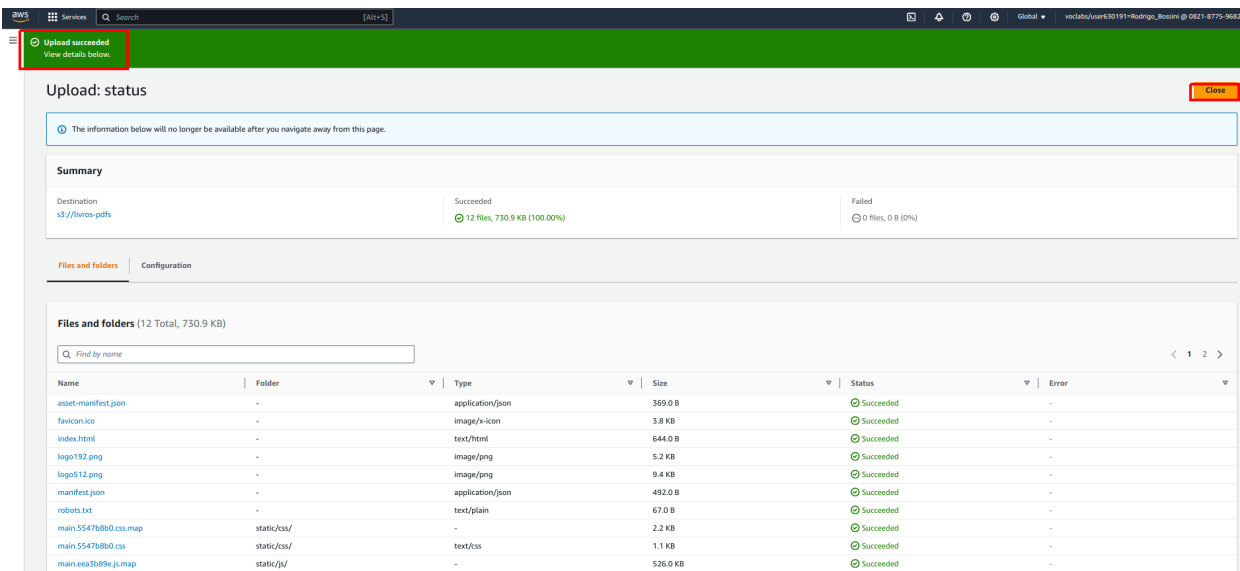
► **Destination details**
Bucket settings that impact new objects stored in the specified destination.

► **Permissions**
Grant public access and access to other AWS accounts.

► **Properties**
Specify storage class, encryption settings, tags, and more.

Cancel Upload

Quando terminar, clique em **Close**.



Upload: status Close

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://livros-pdfs	12 files, 730.9 KB (100.00%)	0 files, 0 B (0%)

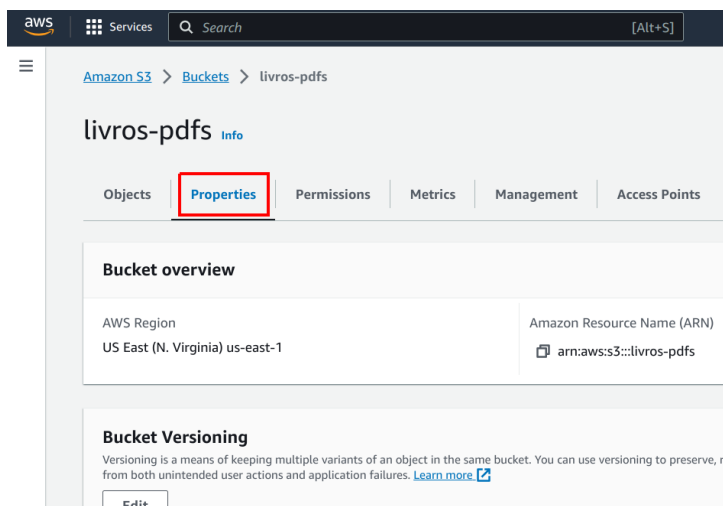
Files and folders | Configuration

Files and folders (12 Total, 730.9 KB)

Find by name

Name	Folder	Type	Size	Status	Error
asset-manifest.json	-	application/json	369.0 B	Succeeded	-
favicon.ico	-	image/x-icon	3.8 KB	Succeeded	-
index.html	-	text/html	644.0 B	Succeeded	-
logo192.png	-	image/png	5.2 KB	Succeeded	-
logo512.png	-	image/png	9.4 KB	Succeeded	-
manifest.json	-	application/json	492.0 B	Succeeded	-
robots.txt	-	text/plain	67.0 B	Succeeded	-
main.5547b8d0.css.map	static/css/	-	2.2 KB	Succeeded	-
main.5547b8d0.css	static/css/	text/css	1.1 KB	Succeeded	-
main.eea3b89e.js.map	static/js/	-	526.0 KB	Succeeded	-

Clique na aba **Properties** novamente. Agora vamos encontrar o link do nosso app.



Amazon S3 > Buckets > livros-pdfs

livros-pdfs [Info](#)

Objects **Properties** Permissions Metrics Management Access Points

Bucket overview

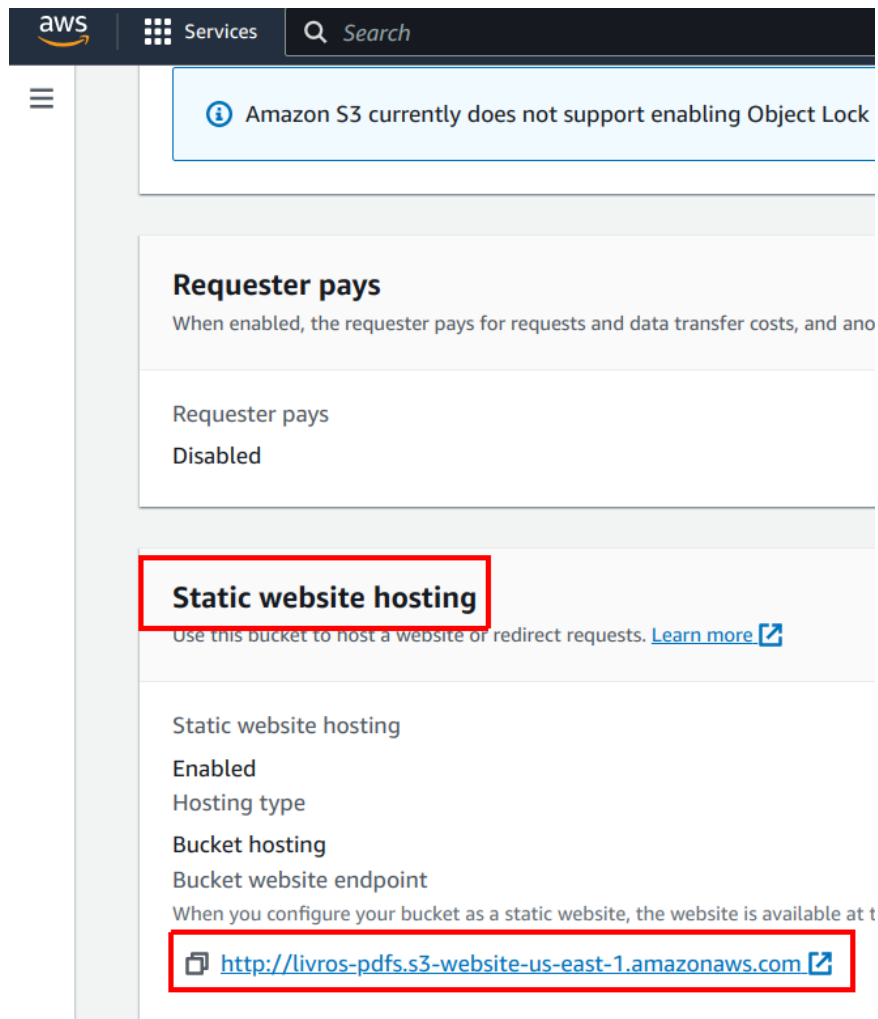
AWS Region	Amazon Resource Name (ARN)
US East (N. Virginia) us-east-1	arn:aws:s3:::livros-pdfs

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, not delete, objects from both unintended user actions and application failures. [Learn more](#)

[Edit](#)

Role a página até encontrá-lo. Geralmente é uma das últimas opções.



Visite o link no seu navegador e veja se deu tudo certo.

React App

livros-pdfs.s3-website-us-east-1.amazonaws.com

ID

Título

Descrição

Autor

Cadastrar

Atualizar

Remover

Obter pelo id

Obter todos

<https://kf95lol2ci.execute-api.us-east-1.amazonaws.com/dev>

Se o Back End estiver em execução, deve ser possível obter a lista completa de livros.

ID
Título
Descrição
Autor
Cadastrar
Atualizar
Remover
Obter pelo id
Obter todos
https://kfq5lol2ci.execute-api.us-east-1.amazonaws.com/dev
Algorithms IV
Cormen
ABC
ABC
Algorithms II

1

Concrete Mathematics

1

Donald Knuth

Cadastrar

Atualizar

Remover

Obter pelo id

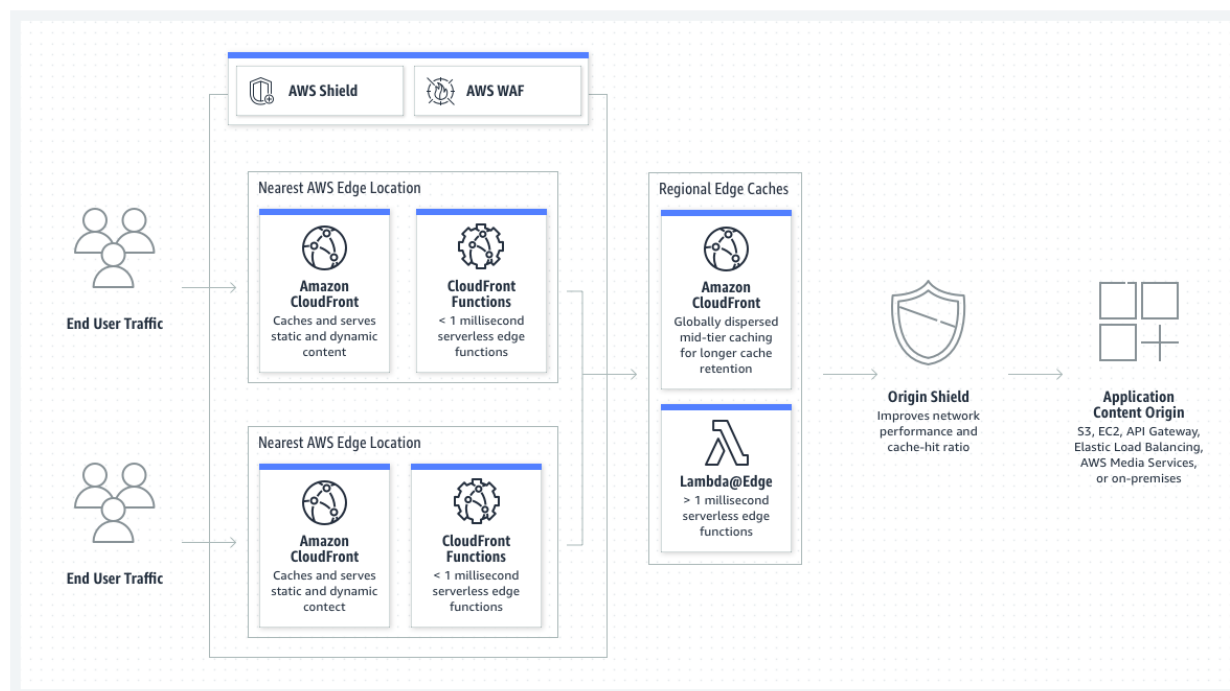
Obter todos

2.3 CDN com AWS CloudFront CDN (Content Delivery Network) é uma coleção de servidores espalhados geograficamente responsáveis por entregar ao cliente determinado conteúdo. A ideia é que todos os servidores sejam capazes de entregar o mesmo conteúdo e que o usuário final seja atendido pelo servidor que estiver mais próximo dele geograficamente, o que tende a diminuir a **latência** (tempo de espera entre a requisição e a resposta). Veja uma figura. A ideia que ela passa é que há diferentes grupos de usuários em diferentes regiões. Cada grupo é atendido por um servidor do CloudFront diferente. Observe que há também outros recursos que podem ser usados, como um Firewall e Caching.

Nota.

O **AWS Shield** que aparece na figura é um serviço de segurança focado em ataques **DDoS (Distributed Denial of Service)**, que são requisições (geralmente milhões ou bilhões por segundo) enviadas ao servidor com o intuito de esgotar seus recursos, inviabilizando o seu funcionamento.

O AWS Web Application Firewall opera na camada de aplicação. Ele nos protege de ataques como **XSS (Cross Site-Scripting)**, ocorre quando uma página incorpora trechos de outro ambiente (outra página, outra origem etc) e este contém código Javascript malicioso que executa sem o usuário saber e **SQL Injection**, que funciona da mesma forma, porém é código SQL incluído numa string que pode ser interpretado e executado pelo servidor.



Veja algumas vantagens que o AWS CloudFront traz.

Desempenho: tempo de latência reduzido, pela possibilidade de um dos servidores estar mais próximo do usuário final.

Redundância e disponibilidade: como existem muitos servidores, a falha de um não compromete o funcionamento do sistema como um todo.

Distribuição de carga: como existem muitos servidores, nenhum deles fica sobrecarregado.

Atualização e manutenção de conteúdo: Quando o conteúdo a ser entregue ao usuário final tiver de ser atualizado, o desenvolvedor pode fazê-lo em um único ponto. O AWS CloudFront possui um mecanismo de “invalidação”, que instrui os servidores a obterem nova cópia atualizada junto à origem do recurso.

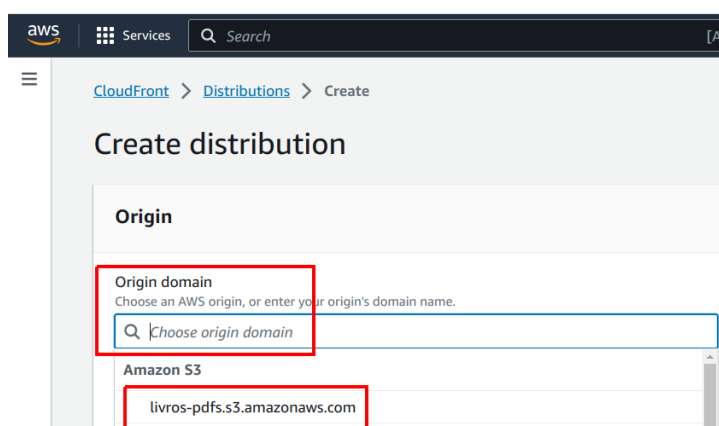
É comum que serviços de streaming, como o Netflix e o Amazon Prime utilizem serviços de CDN. Frameworks CSS, como o Bootstrap, também costumam ser disponibilizados por meio de uma CDN.

Para utilizar o AWS CloudFront, vamos criar uma **distribuição**. Trata-se de uma coleção de configurações em que especificamos a origem dos recursos (o bucket S3 neste caso) e outras coisas, como questões de segurança e outras funcionalidades oferecidas por ele. Comece visitando a sua página no console.

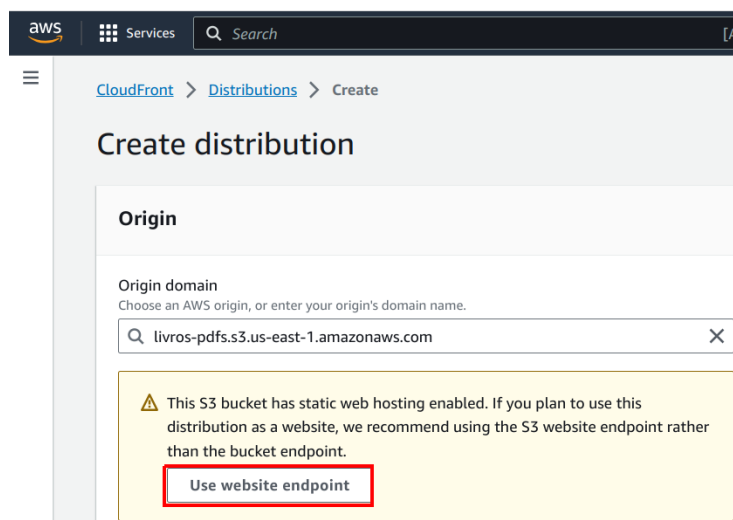
Comece clicando em **Create distribution**. Talvez a sua tela seja um pouco diferente mas você deverá ver esta opção.



Clique no campo **Origin domain** e escolha seu bucket S3.

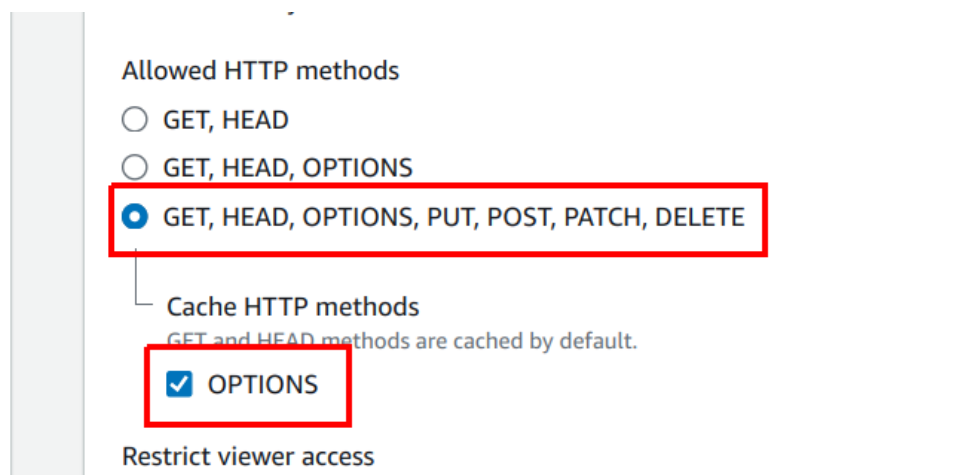


Caso ela apareça, clique para utilizar a recomendação da Amazon, que sugere que utilizemos a URL do site e não aquela direta do bucket S3.



Há algumas razões técnicas sutis para isso sobre as quais não falaremos no momento.

Vá rolando a página. Quando chegar em **Allowed HTTP methods**, faça os seguintes ajustes.



Nota. Quando fazemos cache do método options, estamos dizendo que o navegador pode fazer uma única requisição OPTIONS e, nas próximas, o resultado já estará em cache, sem ter de ser “recalculado”, ou seja, obtido junto ao servidor.

Role um pouco mais a página e escolha não utilizar o WAF. Ele tem um preço alto e não o utilizaremos no momento.

Web Application Firewall (WAF)

☐ **Enable security protections**
Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.

☒ **Do not enable security protections**
Select this option if your application does not need security protections from AWS WAF.

Observe que você pode escolher regiões do mundo onde deseja ter servidores do CloudFront. Mantenha as opções com seu valor padrão e clique em **Create distribution**.

aws Services Search [Alt+S]

Price class - Info

Choose the price class associated with the maximum price that you want to pay.

☒ Use all edge locations (best performance)

☐ Use only North America and Europe

☐ Use North America, Europe, Asia, Middle East, and Africa

Alternate domain name (CNAME) - optional

Add the custom domain names that you use in URLs for the files served by this distribution.

Add item

To add a list of alternative domain names, use the [bulk editor](#).

Custom SSL certificate - optional

Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

Choose certificate

Request certificate

Supported HTTP versions

Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.

☒ HTTP/2

☐ HTTP/3

Default root object - optional

The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

Standard logging

Get logs of viewer requests delivered to an Amazon S3 bucket.

☒ Off

☐ On

IPv6

☐ Off

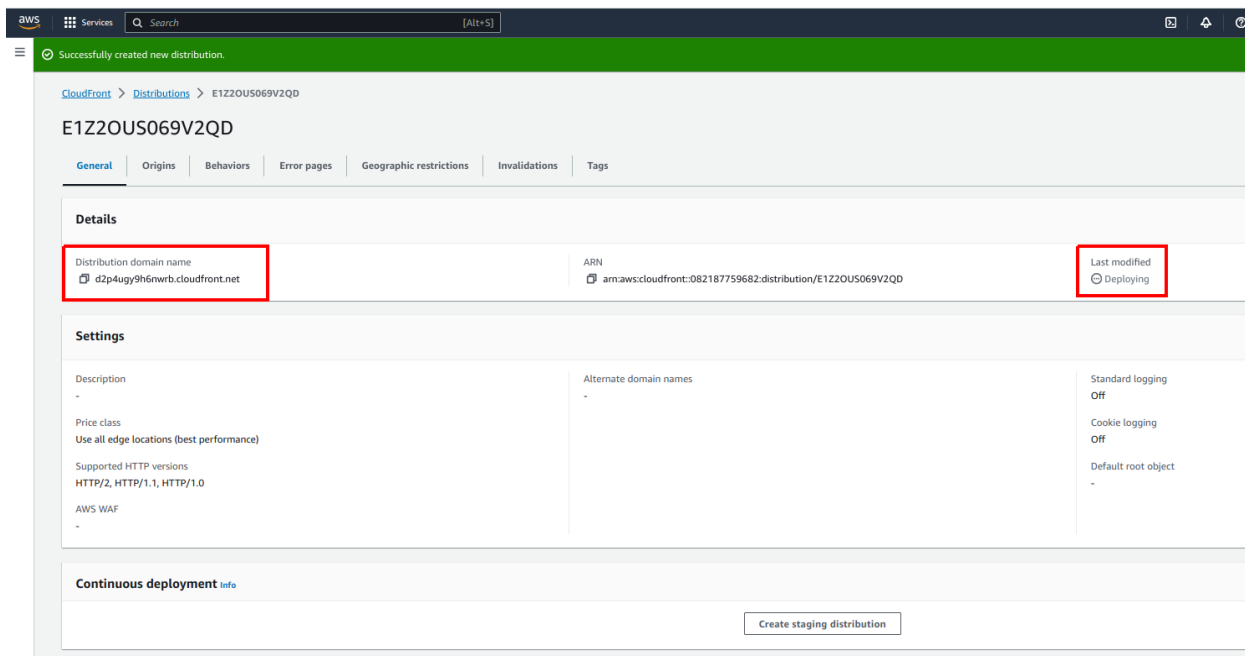
☒ On

Description - optional

Cancel

Create distribution

Na tela a seguir você já terá acesso a uma URL para acesso ao seu site por meio do CloudFront. Seu status deverá ser **deploying**. Pode demorar alguns minutos até que o conteúdo fique acessível.



Aguarde alguns minutos e faça um teste no seu navegador.

React App x +

← → ↻ Not secure d2p4ugy9h6nwrn.cloudfront.net

Referências

[1] Amazon Web Services (AWS) - Cloud Computing Services. 2023. Disponível em <<https://aws.amazon.com/>>. Acesso em setembro de 2023.

[2] PiCloud Launches Serverless Computing Platform To The Public | TechCrunch. 2023. Disponível em <<https://techcrunch.com/2010/07/19/picloud-launches-serverless-computing-platform-to-the-public/>>. Acesso em setembro de 2023.

[3] Serverless Architectures. 2023. Disponível em <<https://martinfowler.com/articles/serverless.html>>. Acesso em setembro de 2023.

[4] Who coined the term 'serverless'?. 2023. Disponível em <<https://www.quora.com/Who-coined-the-term-serverless>>. Acesso em setembro de 2023.