



Serve SPA in AWS

(S3, CLOUDFRONT, SERVERLESS)

Agenda

SERVICES OVERVIEW

AWS SIMPLE STORAGE SERVICE (S3)

AWS CLOUDFRONT

SERVERLESS FRAMEWORK OVERVIEW

HOW TO DEPLOY SPA

MANUAL DEPLOYMENT TO S3

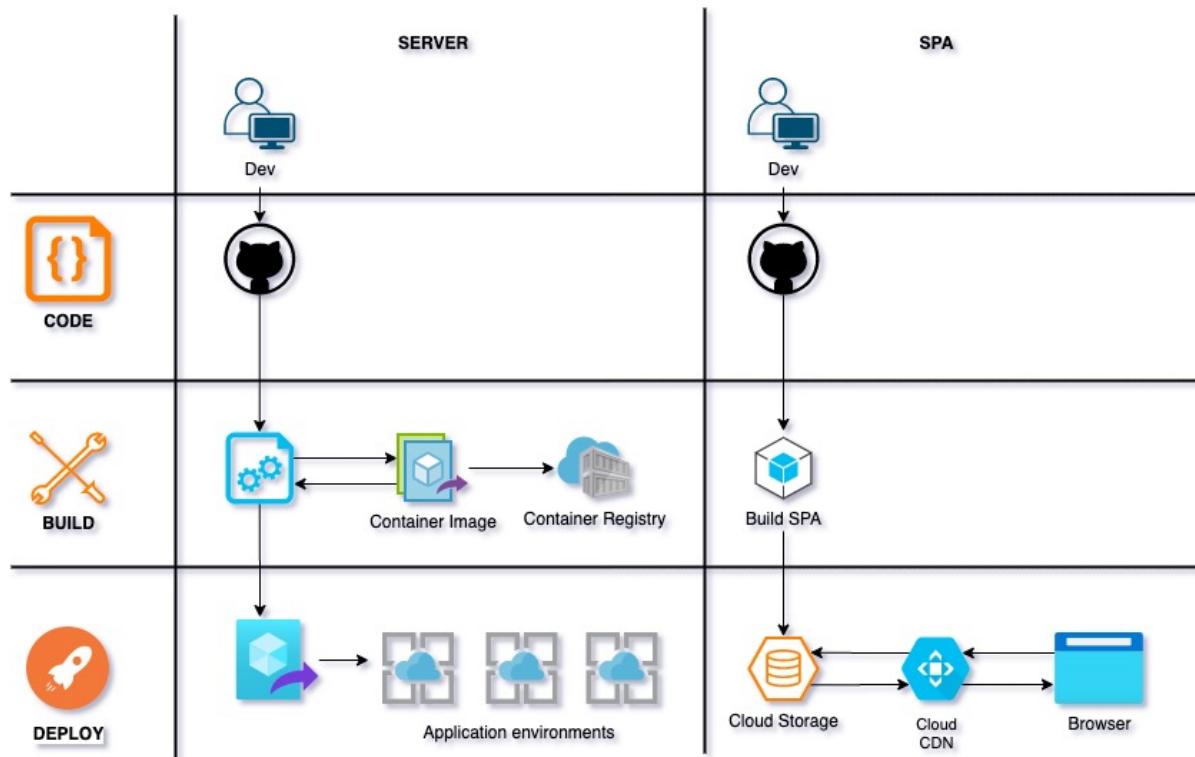
AUTOMATED DEPLOYMENT TO S3 WITH SERVERLESS FINCH

MANUAL CLOUDFRONT DISTRIBUTION CREATION

SERVERLESS & CLOUDFRONT

HOW TO DESTROY INFRASTRUCTURE

Server vs SPA deployment





SERVICES OVERVIEW



AWS SIMPLE STORAGE SERVICE (S3)*

NOTE:

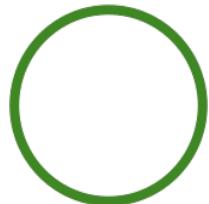
- * - basic complexity level of topic
- ** - intermediate
- *** - advanced

Object-based storage service

Use cases:

- Backup & Recovery;
- Data Archiving;
- Big Data Analytics;
- Static Website Hosting;

S3 basic definitions



Objects contain your data. They are like files.

Objects may consist of:

- **Key** is the name of the object;
- **Value** is the data itself made up of a sequence of bytes;
- **Version ID** when versioning enabled, the version of object;
- **Metadata** additional information attached to the object



Buckets hold objects.

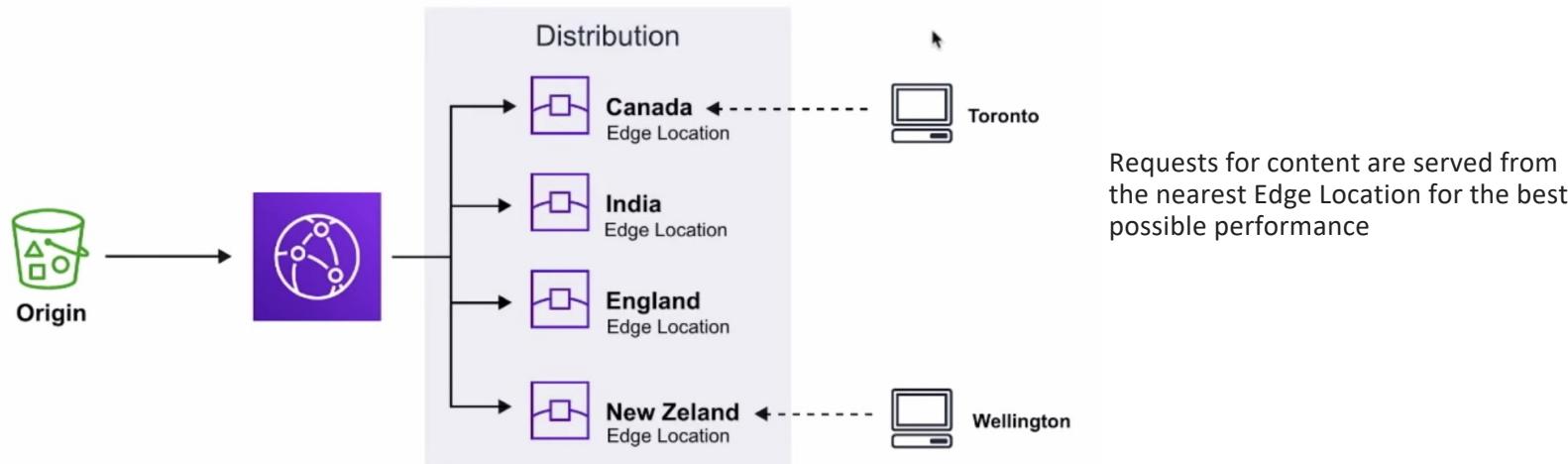
Buckets can also have folders which in turn hold objects.



AWS CLOUDFRONT**

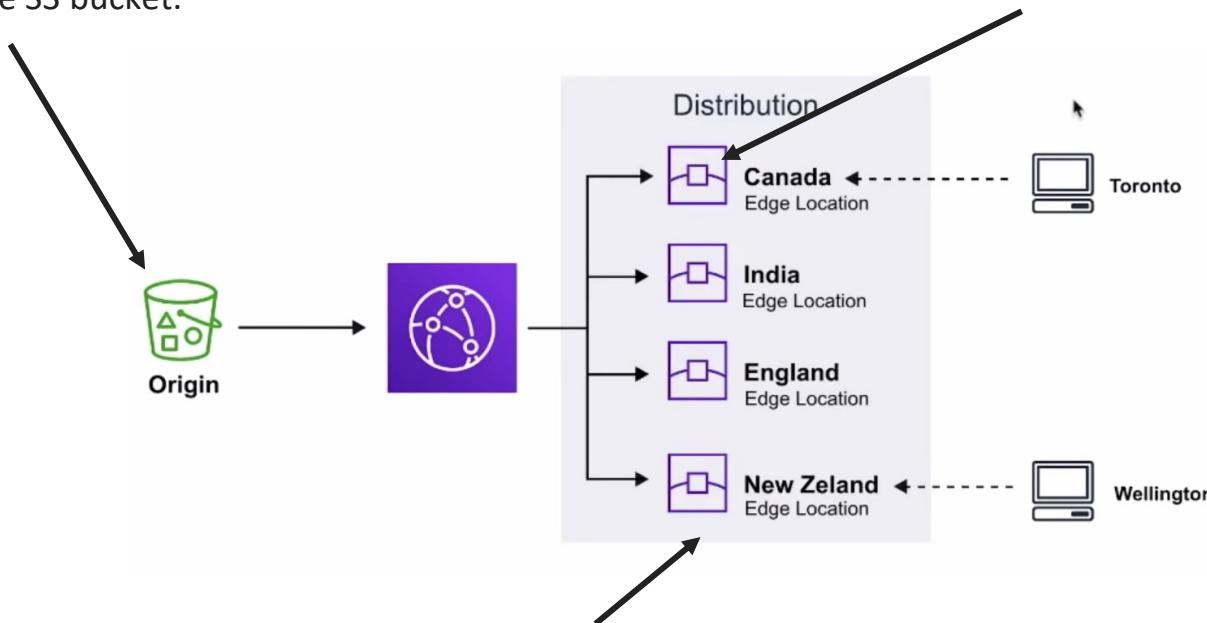
AWS CloudFront

Content delivery network (CDN) which allows you to create cached copies of your website at various Edge Locations around the world.



Core Components

Origin The location where all of original files are located. For example S3 bucket.



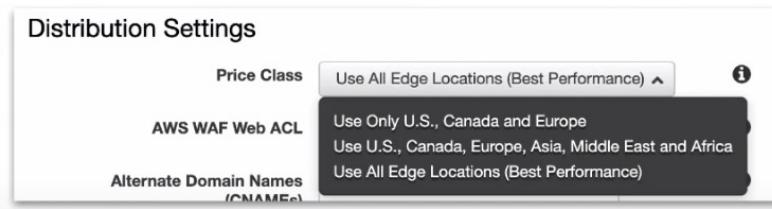
Edge Location The location where web content will be cached. This is different than AWS Region or AZ.

Distribution A collection of Edge Locations which defines how cached content should behave.

Cloudfront Distributions

A **distribution** is a collection of Edge Locations. You specify the Origin(e.g. S3)

It replicates copies based on your Price Class



There are 2 types of Distributions:

- 1) Web (for websites)
- 2) RTMP (for streaming media)

Behaviour

Redirect to HTTPs, restrict HTTP methods, Restrict Viewer Access, set TTLs

Invalidations

You can manually invalidate cache on specific files via Invalidations

Error Pages

You can serve up custom error pages eg. 404

Restrictions

You can use **Geo Restriction** to blacklist or whitelist some countries



SERVERLESS FRAMEWORK OVERVIEW**

What is the Serverless framework

serverless  framework

Do more with less. Serverless.

[HTTPS://WWW.SERVERLESS.COM/](https://www.serverless.com/)

 41,471
Github Stars

 15,907,086
Downloads

 Technology
Partner

"I'm still a huge fan of @Docker but
man this @goserverless stuff is
purely amazing!"

@themccallister
Lead Engineer, Pixel & Tonic



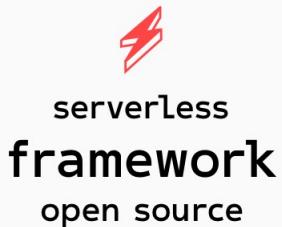
The future is serverless.

Software is eating the world. Unfortunately, most businesses still can't deliver software successfully, much less do so at the pace needed to stay competitive. For those who wish to keep up, let alone lead, software delivery & operation must be radically simplified.

This is what the Serverless Architecture offers — It's built on next-generation public cloud services that auto-scale and charge only when used. When scale, capacity planning & cost management are automated, the result is software that's easier to build, maintain, and often up to 99% cheaper.

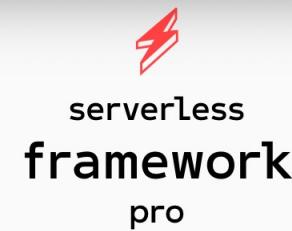
But, Serverless Architectures are new and therefore require a shift in how we previously thought about architectures & workflows. Our goal at Serverless Inc. is to give developers, teams and orgs all of the tools they need to build and operate serverless applications, in one simple, powerful & elegant experience.

Serverless framework: plans

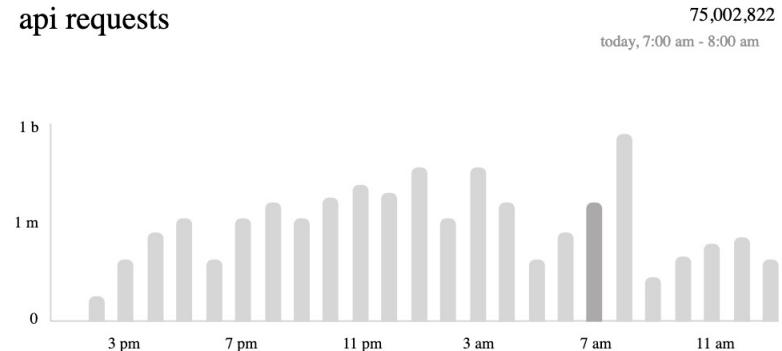


Easy YAML + CLI development and deployment to AWS,
Azure, Google Cloud, Knative & more.

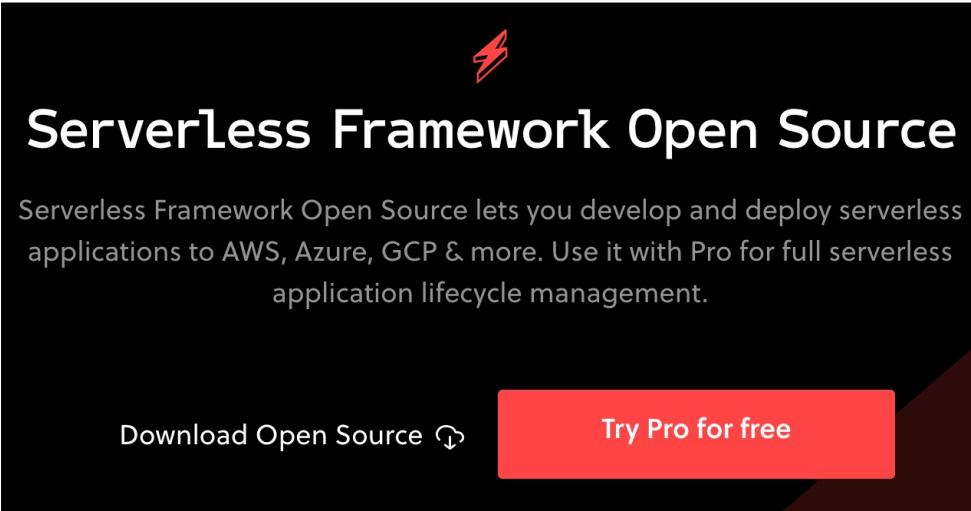
```
1   name: my-rest-api
2
3     functions:
4
5       submit:
6
7         handler: index.submit
8
9         runtime: nodejs10.x
10
11        events:
12
13          -http:
14            path: /submit
15            method: post
```



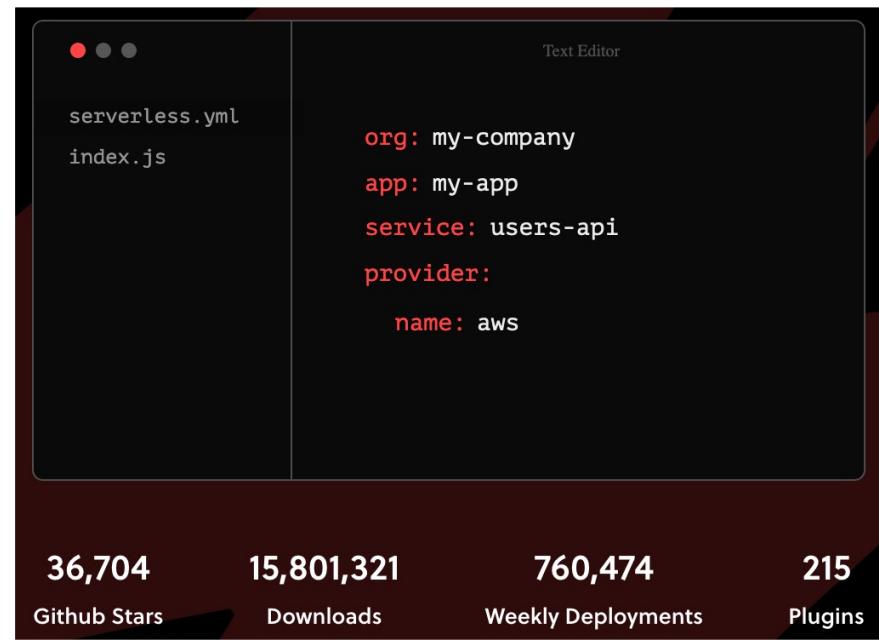
Zero-configuration debugging, CI/CD, troubleshooting,
alerts, safeguards & more.



Serverless framework: Open Source



The landing page for the Serverless Framework Open Source. It features a black header with a red lightning bolt icon. Below the header is a large white title "ServerLess Framework Open Source". A subtext explains the framework's purpose: "Serverless Framework Open Source lets you develop and deploy serverless applications to AWS, Azure, GCP & more. Use it with Pro for full serverless application lifecycle management." At the bottom left is a "Download Open Source" button with a download icon, and at the bottom right is a red "Try Pro for free" button.



A screenshot of the Serverless Framework interface. On the left, there's a dark sidebar with three dots and two files listed: "serverless.yml" and "index.js". On the right, the main area shows deployment details for an "aws" provider:

```
org: my-company
app: my-app
service: users-api
provider:
  name: aws
```

At the bottom, there are four metrics: "36,704 Github Stars", "15,801,321 Downloads", "760,474 Weekly Deployments", and "215 Plugins".

[HTTPS://WWW.SERVERLESS.COM/OPEN-SOURCE](https://www.serverless.com/open-source)

Serverless framework: plugins

Plugins

Extend The Serverless Framework

Top plugins selected and curated by our team

★ 4.3K  23M

Serverless Offline

Emulate AWS λ and API Gateway locally when developing your Serverless project

[view on github](#)

★ 1.6K  16M

Serverless Webpack

Serverless plugin to bundle your lambdas with Webpack

[view on github](#)

★ 0.3K  11M

Serverless Prune Plugin

Deletes old versions of functions from AWS, preserving recent and aliased versions

[view on github](#)

★ 0.8K  9.4M

Serverless Domain Manager

Serverless plugin for managing custom domains with API Gateways.

[view on github](#)

★ 0.2K  9.1M

Serverless Pseudo Parameters

Use \${AWS::AccountId} and other cloudformation pseudo parameters in your serverless.yml values

[view on github](#)

★ 1.4K  8.8M

Serverless HTTP

Use your existing middleware framework (e.g. Express, Koa) in AWS Lambda

[view on github](#)

[HTTPS://WWW.SERVERLESS.COM/PLUGINS](https://www.serverless.com/plugins)

Serverless framework: installation

via NPM

Install the serverless CLI via NPM:

```
● ● ●  
npm install -g serverless
```

Note: If you don't already have [Node](#) on your machine, the latest LTS version of NodeJS.

If you don't want to install Node or NPM, you can inst

Install as a standalone binary

MacOS/Linux

To install the latest version, run this command in your terminal:

```
● ● ●  
curl -o- -L https://slss.io/install | bash
```

To install an specific version you may set a VERSION variable, for exa

```
● ● ●  
curl -o- -L https://slss.io/install | VERSION=2.21.1 bash
```

Then, open another terminal window to run the serverless progr

Windows

Install with [Chocolatey](#):

```
● ● ●   
choco install serverless
```

Initial setup

Run the command below and follow the prompts:

```
● ● ●   
serverless
```

The command will guide you to create a new serverless project.

Note: Users in China are presented with a setup centered around the chinese [Tencent](#) provider. If you're based in China and prefer to be presented with steps as outside of China, set the following environment variable: SERVERLESS_PLATFORM_VENDOR=aws

Serverless framework: a new project initiation

Initial setup

Run the command below and follow the prompts:



The command will guide you to create a new serverless project.

Note: Users in China are presented with a setup centered around the provider. If you're based in China and prefer to be presented with step by step instructions, set the environment variable SERVERLESS_PLATFO

```
→ rs-school-app git:(master) serverless

Serverless: No project detected. Do you want to create a new one? Yes
Serverless: What do you want to make? AWS Node.js
Serverless: What do you want to call this project? RsSchoolServerlessApp

Project successfully created in 'RsSchoolServerlessApp' folder.

You can monitor, troubleshoot, and test your new service with a free Serverless account.

Serverless: Would you like to enable this? No
You can run the "serverless" command again if you change your mind later.

Serverless: Would you like to setup a command line <tab> completion? Yes
Serverless: Which Shell do you use ? zsh
Serverless: We will install completion to ~/.zshrc, is it ok ? Yes

Command line <tab> completion was successfully setup. Make sure to reload your SHELL.
You may uninstall it by running: serverless config tabcompletion uninstall
```

Serverless framework: an existing project initiation

```
→ rs-school-app git:(master) ✘ serverless create --template aws-nodejs
Serverless: Generating boilerplate...
```



```
Serverless: Successfully generated boilerplate for template: "aws-nodejs"
```

```
Serverless: NOTE: Please update the "service" property in serverless.yml with your service name
```

AWS Command Line Interface

AWS Command Line Interface

The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI introduces a new set of simple **file commands** for efficient file transfers to and from Amazon S3.

[HTTPS://AWS.AMAZON.COM/CLI](https://aws.amazon.com/cli)

AWS credentials

The screenshot shows the AWS IAM dashboard with a search bar containing "iam". The dashboard provides an overview of IAM resources and best practices.

IAM dashboard:

- Sign-in URL for IAM users in this account:** <https://724443507988.signin.aws.amazon.com/console>
- IAM resources:**
 - Users: 1
 - Groups: 1
 - Identity providers: 0
 - Customer managed policies: 22
- Best practices:**
 - Grant least privilege access: Establishing a principle of least privilege ensures that identities are only permitted to perform the most minimal set of functions necessary to fulfill a specific task, while balancing usability and efficiency.
 - Enable Identity federation: Centrally manage users and access across multiple applications and services. For federation to multiple accounts in your AWS Organization, you can configure your identity source in [AWS Single Sign-on](#).
 - Enable MFA: For extra security, we recommend that you require multi-factor authentication (MFA) for all users.
 - Rotate credentials regularly: Change your own passwords and access keys regularly, and make sure that all users in your account do as well.
 - Enable IAM Access Analyzer: Enable IAM Access Analyzer to analyze public, cross-account, and cross-organization access.
- Learn more about all security best practices.**

Additional information:

- [IAM documentation](#)
- [Videos, IAM release history and additional resources](#)

Tools:

- [Web identity federation playground](#)
- [Policy simulator](#)

Quick links:

- [My access key](#)

AWS Access key ID & Secret access key

The screenshot shows the AWS IAM User Management console. On the left, there's a sidebar with 'Add user' and 'Delete user' buttons. Below that is a search bar and a dropdown for 'User name' set to 'boale'. To the right of the search bar is a 'Groups' button, which has 'AdminGroup' selected. At the top right of the main area, it says 'Showing 1'.

The main area is titled 'Summary' and contains the following details:

- User ARN: arn:aws:iam::724443507988:user/boale
- Path: /
- Creation time: 2020-04-12 18:22 UTC+0300

Below the summary, there are tabs for 'Permissions', 'Groups (1)', 'Tags', 'Security credentials' (which is selected), and 'Access Advisor'.

The 'Sign-in credentials' section shows:

- Summary: Console sign-in link: https://..., MFA is required when signing
- Console password: Enabled (last signed in Today)
- Assigned MFA device: arn:aws:iam::724443507988:mfa/...
- Signing certificates: None

The 'Access keys' section contains a note about using access keys for secure requests and a 'Create access key' button.

A modal window titled 'Create access key' is open in the bottom right. It displays a green 'Success' message: "This is the only time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time." It also includes a 'Download .csv file' button and a table showing the generated keys:

Access key ID	Secret access key
AKIA2RLBSVUKAGHVCF7Z	***** Show

At the bottom right of the modal is a 'Close' button.

AWS-cli credentials setup

```
$ aws configure
```

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

```
Default region name [None]: us-west-2
```

```
Default output format [None]: ENTER
```

```
→ nodejs-aws-fe git:(main) ✘ aws configure
```

```
→ nodejs-aws-fe git:(main) ✘ cat ~/.aws/credentials
```

```
[default]
```

```
aws_access_key_id = [REDACTED]
```

```
aws_secret_access_key = [REDACTED]
```

```
→ nodejs-aws-fe git:(main) ✘ cat ~/.aws/config
```

```
[default]
```

```
output = json
```

```
region = us-east-1
```

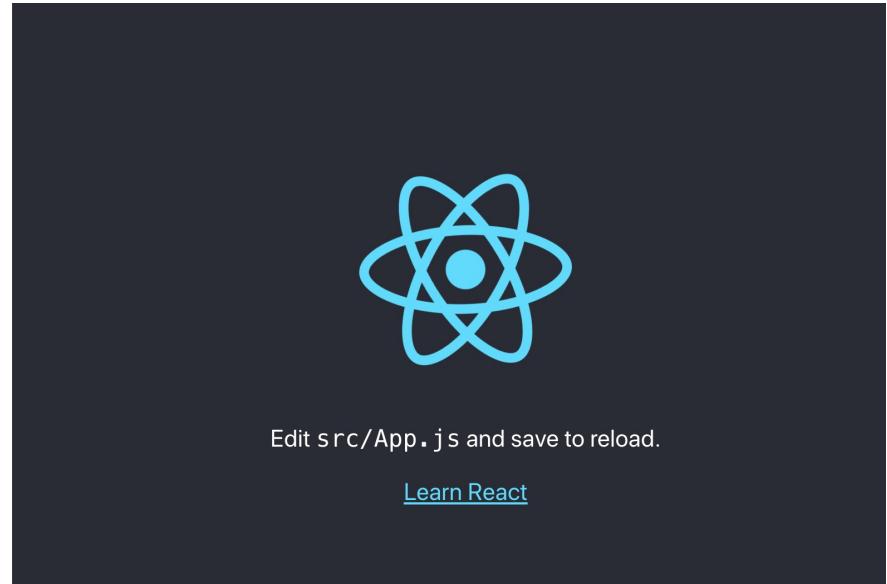


HOW TO DEPLOY SPA

Repository

INGREDIENTS

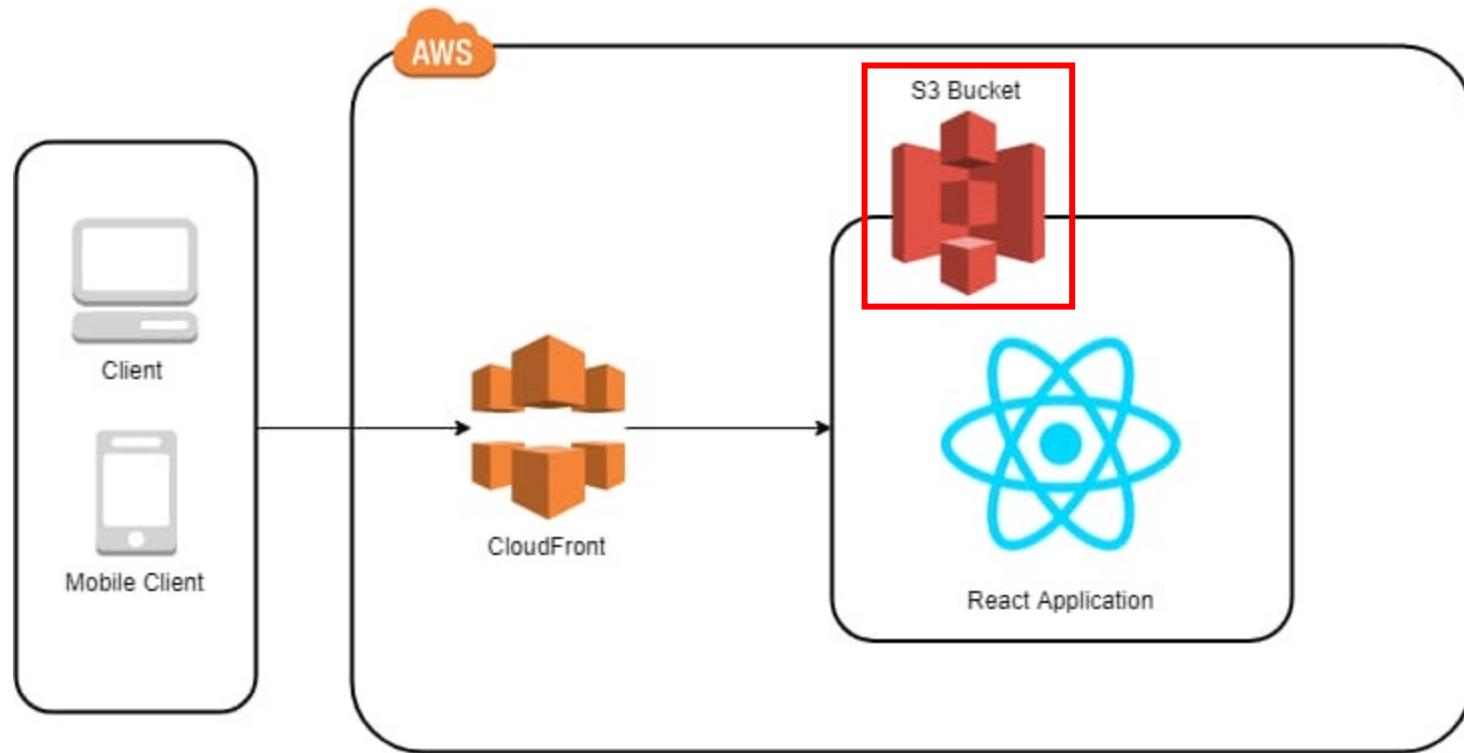
- Application: a simple app by create-react-app
(<https://github.com/facebook/create-react-app>)
- Commands:
 - `npm i -g create-react-app`
 - `npx create-react-app my-app`
 - `cd my-app`
 - `npm start`
 - Open <http://localhost:3000> in browser





MANUAL DEPLOYMENT TO S3*

AWS S3 manual deployment



AWS S3 manual deployment

INGREDIENTS

- Project Source Code
- Script: "build": "npm run build"
- Command: npm run build
- Tool: AWS Simple Storage Service (S3)
- Config: {
 "Version": "2012-10-17",
 "Id": "WebAppBucketPolicy",
 "Statement": [
 {"Sid": "PublicReadForGetBucketObjects",
 "Effect": "Allow",
 "Principal": "*",
 "Action": "s3:GetObject",
 "Resource": "arn:aws:s3:::rs-school-app/*"
 }
]



AWS S3: create a bucket

[Amazon S3](#) > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

app-rs-school

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

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

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

▼ Advanced settings

Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. [Learn more](#)

Disable

Enable

Permanently allows objects in this bucket to be locked. Additional configuration is required after bucket creation to protect objects in this bucket from being deleted or overwritten.

ⓘ Enabling Object Lock automatically enables Bucket Versioning.

Cancel

Create bucket

Bucket settings for Block Public Access

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.

AWS S3: create a bucket

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

Server-side encryption
 Disable
 Enable

▼ Advanced settings

Object Lock
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. [Learn more](#)

Disable
 Enable
Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

Info Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Bucket Versioning.

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

AWS S3 bucket

Amazon S3

▼ Account snapshot

Last updated: Dec 21, 2021 by Storage Lens. Metrics are generated every 24 hours. [Learn more](#)

[View Storage Lens dashboard](#)

Total storage
232.7 MB

Object count
489

Avg. object size
487.3 KB

You can enable advanced metrics in the
"default-account-dashboard"
configuration.

Buckets (10) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)



[Copy ARN](#)

[Empty](#)

[Delete](#)

[Create bucket](#)



rs



1 match

< 1 >

Name

AWS Region

Access

Creation date

app-rs-school

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

Objects can be public

December 22, 2021, 17:07:27 (UTC+02:00)

AWS S3: configure public bucket – AWS Policy Generator

Amazon S3 > app-rs-school > Edit bucket policy

Edit bucket policy Info

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](#)

Bucket ARN
 arn:aws:s3:::app-rs-school

Policy

```
1 ▼ {
  2   "Version": "2012-10-17",
  3   "Statement": [
  4     {
  5       "Sid": "Statement1",
  6       "Principal": "*",
  7       "Effect": "Allow",
  8       "Action": [
  9         "s3:GetObject"
 10      ],
 11       "Resource": [
 12         "arn:aws:s3:::app-rs-school/*"
 13       ]
 14     }
 15   ]
 16 }
```

+ Add new statement

JSON Ln 6, Col 19

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Policy examples **Policy generator**

Edit statement Statement1

1. Add actions
All services > S3
 getOb...

Access level - read or write
 GetObject
 GetObjectAcl
 GetObjectLegalHold
 GetObjectRetention
 GetObjectTagging
 GetObjectTorrent
 GetObjectVersion
 GetObjectVersionAcl
 GetObjectVersionForReplication

2. Add a resource

3. Add a condition (optional)

Cancel **Save changes**

AWS S3: configure public bucket

Amazon S3 > app-rs-school

app-rs-school Info

Publicly accessible

Objects Properties Permissions Metrics Management Access Points

Access control list (ACL)

Grant basic/read/write permissions to other AWS accounts. [Learn more](#)

This bucket has the bucket owner enforced setting applied for Object Ownership
When **bucket owner enforced** is applied, use bucket policies to control access. [Learn more](#)

The console displays combined access grants for duplicate grantees
To see the full list of ACLs, use the Amazon S3 REST API, AWS CLI, or AWS SDKs.

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: decc1de3f40261fe6daaadef96cb2679dd11caf75e43750e685bfe6583492484	List, Write	Read, Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	-	-
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-
S3 log delivery group Group: http://acs.amazonaws.com/groups/s3/LogDelivery	-	-

AWS S3: enable Static website hosting

Amazon S3 > app-rs-school

app-rs-school Info

Publicly accessible

Objects Properties Permissions Metrics Management Access Points

Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) <code>arn:aws:s3:::app-rs-school</code>	Creation date December 22, 2021, 17:07:27 (UTC+02:00)
---	---	--

Static website hosting

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

Static website hosting
Disabled

Amazon S3 > app-rs-school > Edit static website hosting

Edit static website hosting Info

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.
`index.html`

Error document - optional
This is returned when an error occurs.
`index.html`

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

1 |

Cancel **Save changes**

AWS S3: upload files from dist folder (after build)

The screenshot illustrates the process of uploading files from a local directory to an AWS S3 bucket named "app-rs-school".

Left Panel (Bucket Overview):

- Shows the bucket "app-rs-school" is "Publicly accessible".
- Contains tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points.
- The "Objects" section shows 0 objects.
- Includes a toolbar with actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.
- A progress bar indicates an ongoing upload of 214 files (247.8 KB) at 0.57% completion.
- An "Upload" status box shows the total remaining time is 4 minutes.

Right Panel (Upload Progress):

- Shows the "Upload" progress dialog with a list of files being uploaded.
- Details: Total remaining: 214 files; 247.8 KB (0.57%); Estimated time remaining: 4 minutes; Transfer rate: 1.1 kB/s.
- The progress bar is at 3%.
- The "Upload" button is highlighted in orange.
- The "Upload" status box indicates the information will no longer be available after navigating away.
- Summary:** Destination: s3://app-rs-school. Succeeded: 10 files, 8.8 KB (3.43%). Failed: 0 files, 0 B (0%).
- Files and folders:** (224 Total, 256.6 KB) table.
- Destination:** Destination: s3://app-rs-school. 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.

AWS S3 deployment complete!

Amazon S3 > app-rs-school > app/

app/

Objects Properties

Objects (16)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3](#).

[Copy S3 URI](#) [Copy URL](#) [Download](#)

[Find objects by prefix](#)

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	app-navigation.config.js	js
<input type="checkbox"/>	app-navigation.config.js.map	map
<input type="checkbox"/>	app-routing.config.js	js
<input type="checkbox"/>	app-routing.config.js.map	map
<input type="checkbox"/>	app-routing.module.js	js
<input type="checkbox"/>	app-routing.module.js.map	map

C ⚠ Not Secure | rs-school-app.s3-website-us-east-1.amazonaws.com

[Copy S3 URI](#) [Star](#) [Report](#) [YAB](#) [W](#) [⚙️](#)



Edit `src/App.js` and save to reload.

[Learn React](#)



DEPLOY TO S3 WITH SERVERLESS FINCH**

Serverless Finch

Serverless Finch

A Serverless plugin to deploy static website assets to AWS S3.

[View on Github](#)

downloads 53k/month npm v2.8.0 license MIT

[HTTPS://WWW.SERVERLESS.COM/PLUGINS/SERVERLESS-FINCH](https://www.serverless.com/plugins/serverless-finch)

Serverless finch installation and initiation

Installation

```
npm install --save serverless-finch
```

Usage

First, update your `serverless.yml` by adding the following:

```
1  plugins:
2    - serverless-finch
3
4  custom:
5    client:
6      bucketName: unique-s3-bucketname # (see Configuration Parameters below)
7      # [other configuration parameters] (see Configuration Parameters below)
```

Serverless finch installation and initiation

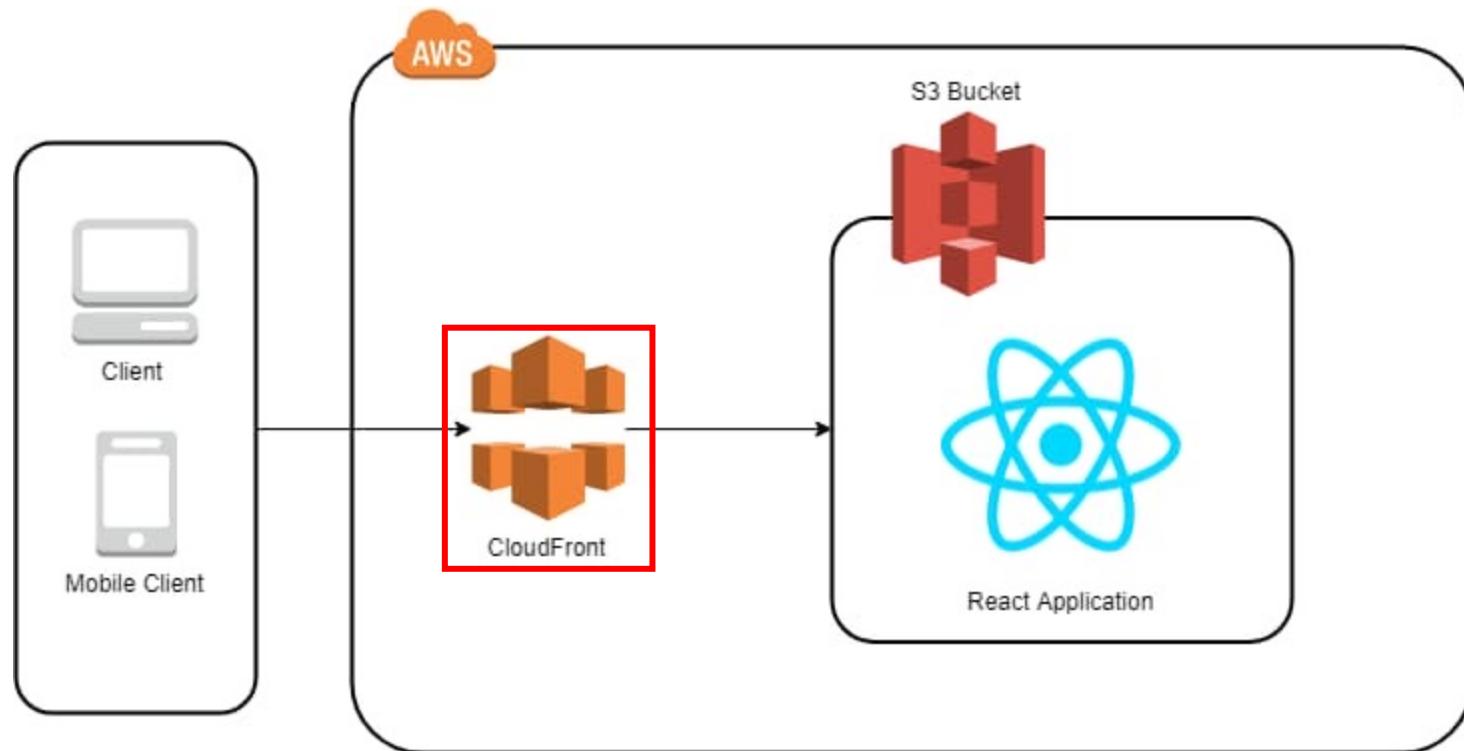
```
package.json x
20 > "client:deploy": "serverless client deploy --no-config-change --no-policy-change --no-cors-change",
21 >   "client:build:deploy": "npm run build && npm run client:deploy"
22 },
23 "eslintConfig": {  
  
> serverless client deploy --no-config-change --no-policy-change --no-cors-change  
  
Serverless: This deployment will:  
Serverless: - Upload all files from 'build' to bucket 'rs-school-app'  
? Do you want to proceed? true  
Serverless: Looking for bucket...  
Serverless: Bucket found...  
Serverless: Deleting all objects from bucket...  
Serverless: Retaining existing bucket configuration...  
Serverless: Retaining existing bucket policy...  
Serverless: Retaining existing tags...  
Serverless: Retaining existing bucket CORS configuration...  
Serverless: Uploading client files to bucket...  
Serverless: Success! Your site should be available at http://rs-school-app.s3-website-us-east-1.amazonaws.com/  
!+ Done in 40.87s.
```





CREATE A CLOUDFRONT DISTRIBUTION**

Create a CloudFront Distribution



Create a CloudFront Distribution

[CloudFront](#) > [Distributions](#) > [create](#)

Create distribution

Origin

Origin domain

Choose an AWS origin, or enter your origin's domain name.

 X

Origin path - optional [Info](#)

Enter a URL path to append to the origin domain name for origin requests.

Name

Enter a name for this origin.

S3 bucket access [Info](#)

Use a CloudFront origin access identity (OAI) to access the S3 bucket.

- Don't use OAI (bucket must allow public access)
- Yes use OAI (bucket can restrict access to only CloudFront)

Add custom header - optional

CloudFront includes this header in all requests that it sends to your origin.

[Add header](#)

Enable Origin Shield [Info](#)

Origin Shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.

- No
- Yes

[► Additional settings](#)

Default cache behavior

Path pattern [Info](#)

Compress objects automatically [Info](#)

- No
- Yes

Viewer

Viewer protocol policy

- HTTP and HTTPS
- Redirect HTTP to HTTPS
- HTTPS only

Allowed HTTP methods

- GET, HEAD
- GET, HEAD, OPTIONS
- GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods

GET and HEAD methods are cached by default.

- OPTIONS

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

- No
- Yes

Create a CloudFront Distribution

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

Cache policy and origin request policy (recommended)

Legacy cache settings

Cache policy

Choose an existing cache policy or create a new one.

CachingOptimized

Default policy when CF compression is enabled

Recommended for S3 origins



[Create policy](#) [View policy](#)

Origin request policy - optional

Choose an existing origin request policy or create a new one.

Select origin policy



[Create policy](#)

Response headers policy - optional

Choose an existing response headers policy or create a new one.

Select response headers



[Create policy](#)

▼ Additional settings

Smooth streaming

Choose No if your origin is configured to use Microsoft IIS for Smooth Streaming.

No

Yes

Field-level encryption [Info](#)

Choose a field-level encryption configuration.

Select a field-level encryption profile



Enable real-time logs [Info](#)

No

Yes

Function associations - *optional* [Info](#)

Choose an edge function to associate with this cache behavior, and the CloudFront event that invokes the function.

	Function type	Function ARN / Name	Include body
Viewer request	No association		
Viewer response	No association		
Origin request	No association		
Origin response	No association		

Settings

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

AWS WAF web ACL - *optional*

Choose the web ACL in AWS WAF to associate with this distribution.

Select a web ACL



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](#).

Create a CloudFront Distribution

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](#) ▾ [C](#)

[Request certificate](#) ?

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

[HTTP/2](#)

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](#)

Create a CloudFront Distribution

EWUM4ARWOFP8L

General Origins Behaviors Error pages Geographic restrictions Invalidations Tags

Details

Distribution domain name daeejxqnyrtf0.cloudfront.net	ARN arn:aws:cloudfront::042421165226:distribution/EWUM4ARWOFP8L	Last modified Deploying
--	--	----------------------------

Settings

Description -	Alternate domain names -	Standard logging Off
Price class Use all edge locations (best performance)	Cookie logging Off	Default root object -
Supported HTTP versions HTTP/2, HTTP/1.1, HTTP/1.0		
AWS WAF -		

CloudFront > Distributions

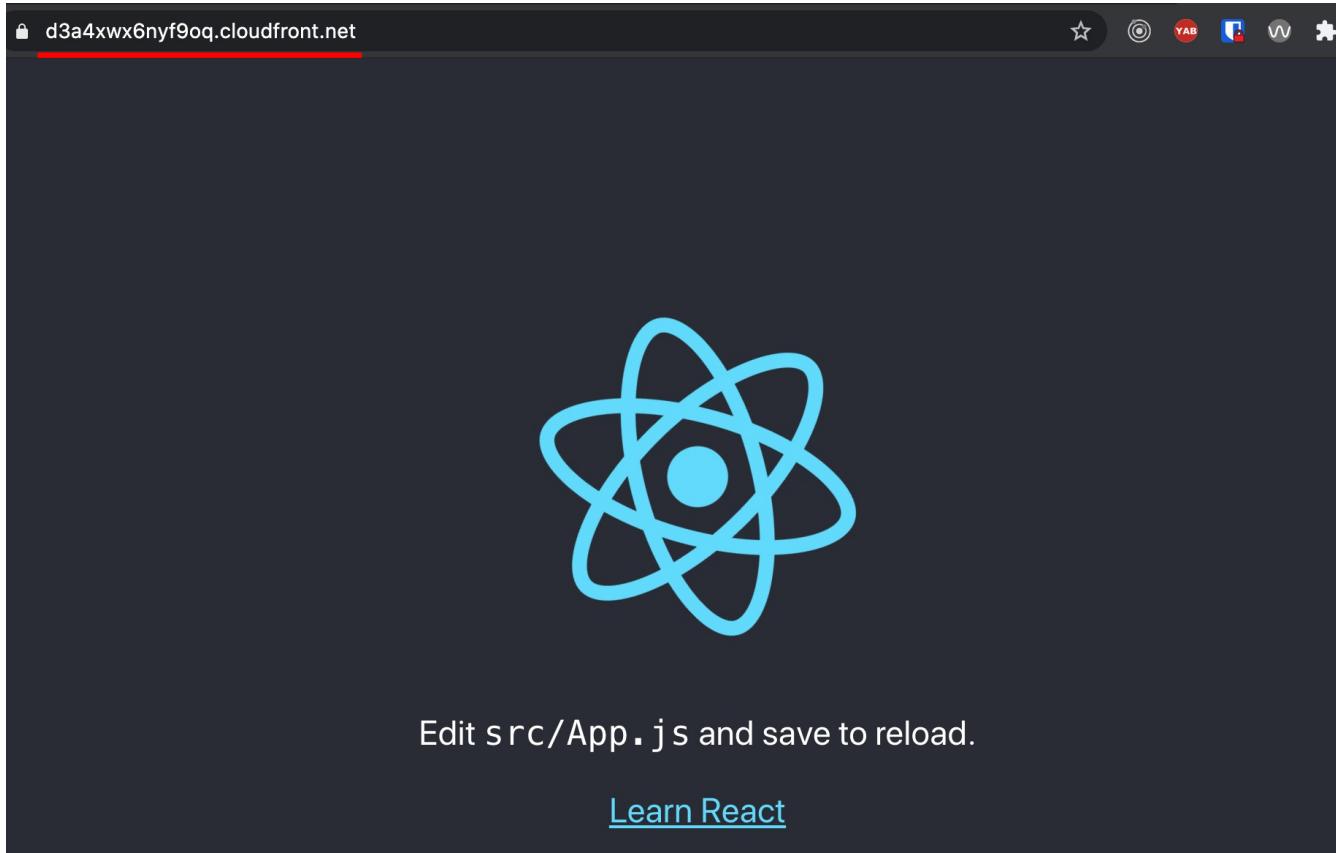
Distributions (2) [Info](#)

[Enable](#) [Disable](#) [Delete](#) [Create distribution](#)

[1](#)

<input type="checkbox"/>	ID	Description	Domain name	Alternate domain names	Origins
<input type="checkbox"/>	EWUM4ARWOFP8L	-	daeejxqnyrtf0.cloud...	-	app-rs-school.s3.us-eas...

Create a CloudFront Distribution



Create a CloudFront Distribution

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](#)

[Policy examples](#) [Policy generator](#)

Bucket ARN
 arn:aws:s3:::app-rs-school

Policy

```
1▼ {
2    "Version": "2012-10-17",
3    "Statement": [
4        {
5            "Sid": "Statement1",
6            "Effect": "Allow",
7            "Principal": "*",
8            "Action": "s3:GetObject",
9            "Resource": "arn:aws:s3:::app-rs-school/*"
10        },
11        {
12            "Sid": "Statement2",
13            "Principal": {
14                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity E3JDUTI53YD8675TD"
15            },
16            "Effect": "Allow",
17            "Action": [
18                "s3:GetObject"
19            ],
20            "Resource": [
21                "arn:aws:s3:::app-rs-school/*"
22            ]
23        }
24    ]
25 }
```

[+ Add new statement](#)

Edit statement Statement2 [Remove](#)

1. Add actions

All services > S3

getOb [X](#)

Access level - read or write

GetObject [i](#)

GetObjectAcl [i](#)

GetObjectLegalHold [i](#)

GetObjectRetention [i](#)

GetObjectTagging [i](#)

GetObjectTorrent [i](#)

GetObjectVersion [i](#)

GetObjectVersionAcl [i](#)

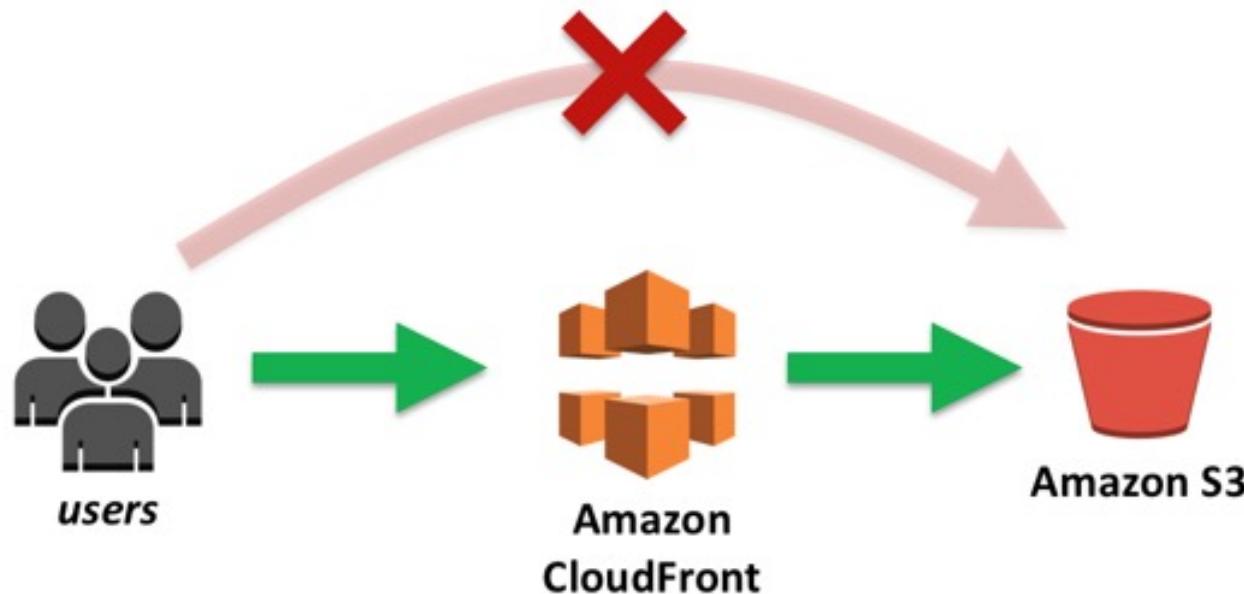
GetObjectVersionForReplication [i](#)

GetObjectVersionTacking [i](#)

2. Add a resource [Add](#)

3. Add a condition (optional) [Add](#)

Create a CloudFront Distribution



Create a CloudFront Distribution

CloudFront > Distributions > EWUM4ARWOFP8L

EWUM4ARWOFP8L

General | Origins | Behaviors | Error pages | Geographic restrictions | **Invalidations** | Tags

Invalidations

Filter invalidations by property or value

[View details](#)

[Copy to new](#)

Create invalidation

Invalidation ID

▼

Status

▼

Date created

<

1

>

⚙

No invalidations

You don't have any invalidation:

[Create invalidation](#)

CloudFront > Distributions > EWUM4ARWOFP8L > Create invalidation

Create invalidation

Object paths

Add object paths

Add the path for each object that you want to remove from the CloudFront cache. You can use wildcards (*).

```
/*
```

ⓘ To add object paths individually, use the standard editor.

[Cancel](#)

Create invalidation

Create a CloudFront Distribution

⌚ Successfully created invalidation I1MQ53SNX8V4HR. ×

CloudFront > Distributions > EWUM4ARWOFP8L > I1MQ53SNX8V4HR

Invalidation details	
Date created December 22, 2021 at 3:47:11 PM UTC	Object paths /*
Status ○ In progress	Copy to new

CloudFront > Distributions > EWUM4ARWOFP8L

EWUM4ARWOFP8L

General | Origins | Behaviors | Error pages | Geographic restrictions | **Invalidations** | Tags

Invalidations		
<input type="text"/> Filter invalidations by property or value		
Invalidation ID	Status	Date created
I1MQ53SNX8V4HR	Completed	December 22, 2021 at 3:47:11 PM UTC

View detailsCopy to newCreate invalidation



SERVERLESS & CLOUDFRONT***

Serverless Cloudfont setup



The screenshot shows a code editor interface with the following details:

- Project View:** On the left, the project structure is displayed under "rs-school-app". It includes ".serverless", "build", "node_modules" (highlighted in yellow), "public", and a folder named "serverless-single-page-app-plugin". This last folder is highlighted with a red box.
- Code Editor:** On the right, two tabs are open: "package.json" and "index.js". The "package.json" tab is active and shows the following JSON code:

```
1 {  
2     "name": "serverless-single-page-app-plugin",  
3     "version": "1.0.1",  
4     "description": "A plugin to simplify deploying Single Page Application using S3 and CloudFront",  
5     "author": "",  
6     "license": "MIT"  
7 }  
8 }
```

Serverless Cloudfont setup

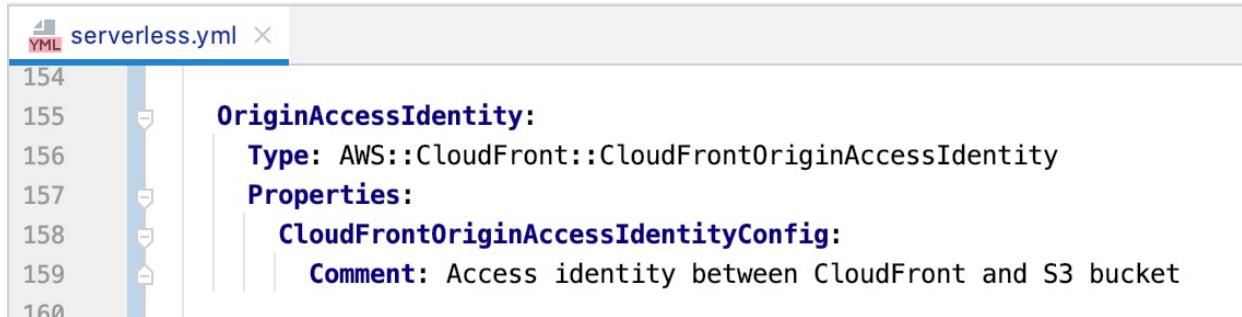
```
YML serverless.yml ×
29   plugins:
30     - serverless-finch
31     - serverless-single-page-app-plugin
32
33   custom:
34     ## Serverless-finch configuration
35     client:
36       bucketName: rs-school-app
37       distributionFolder: build
38       indexDocument: index.html
39       manageResources: false
40     ## Serverless-single-page-app-plugin configuration:
41     s3LocalPath: ${self:custom.client.distributionFolder}/
42     s3BucketName: ${self:custom.client.bucketName}
```

Serverless Cloudfont setup

```
YML serverless.yml ×  
121 # you can add CloudFormation resource templates here  
122 resources:  
123   Resources:  
124     ## Specifying the S3 Bucket  
125       WebAppS3Bucket:  
126         Type: AWS::S3::Bucket  
127         Properties:  
128           BucketName: ${self:custom.s3BucketName}  
129           AccessControl: PublicRead  
130           WebsiteConfiguration:  
131             IndexDocument: index.html  
132             ErrorDocument: index.html  
133             #  
134             VersioningConfiguration:  
135               Status: Enabled
```

```
YML serverless.yml ×  
135 ## Specifying the policies to make sure all files inside the Bucket are available to CloudFront  
136 WebAppS3BucketPolicy:  
137   Type: AWS::S3::BucketPolicy  
138   Properties:  
139     Bucket:  
140       Ref: WebAppS3Bucket  
141     PolicyDocument:  
142       Statement:  
143         - Sid: 'AllowCloudFrontAccessIdentity'  
144           Effect: Allow  
145           Action: s3:GetObject  
146           Resource: arn:aws:s3:::${self:custom.s3BucketName}/*  
147           Principal:  
148             AWS:  
149               Fn::Join:  
150                 - ''  
151                 - - 'arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity'  
152                 - !Ref OriginAccessIdentity
```

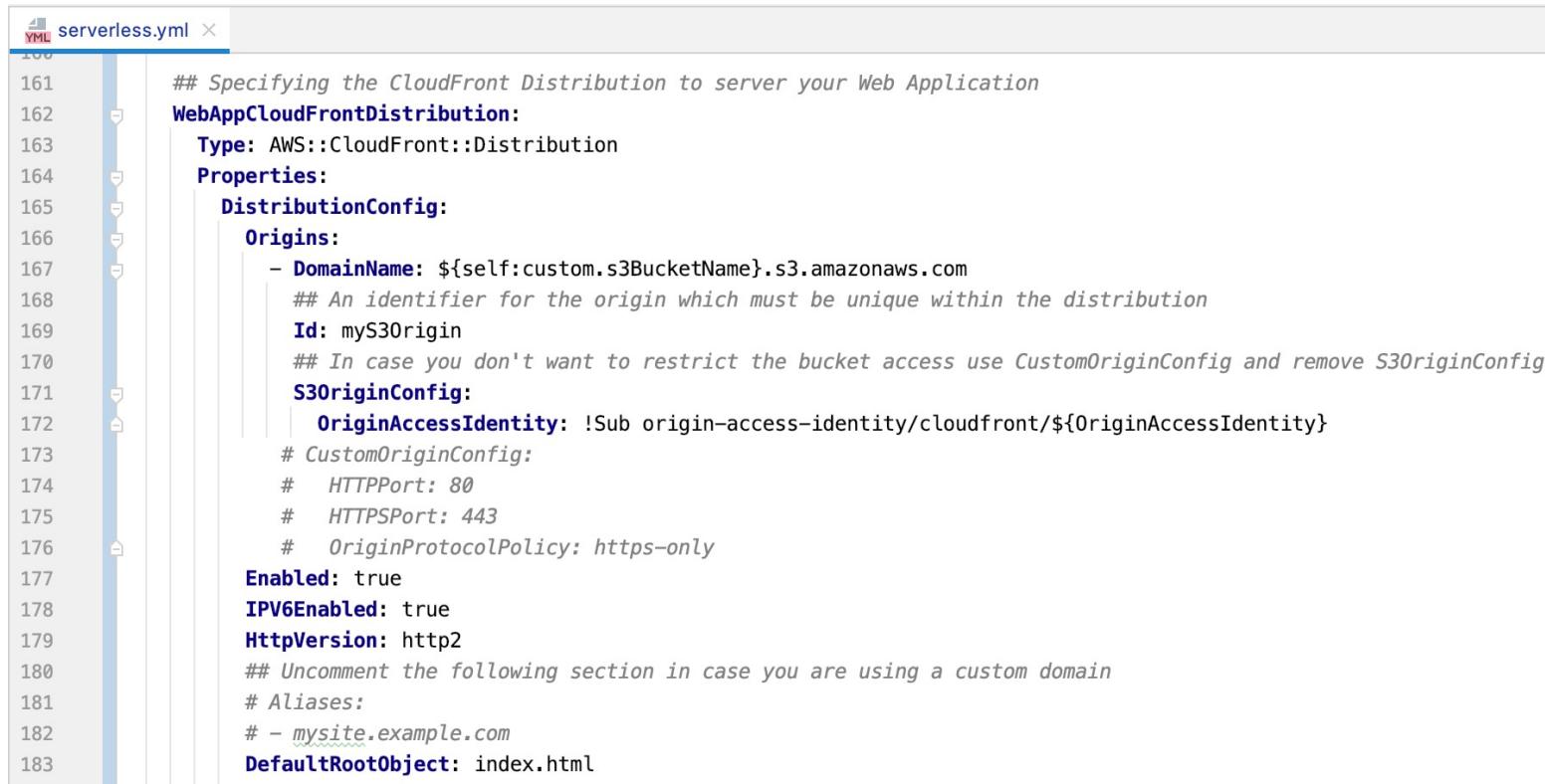
Serverless Cloudfont setup



The screenshot shows a code editor window with a tab labeled "serverless.yml". The file content is as follows:

```
YML serverless.yml ×
154
155     OriginAccessIdentity:
156         Type: AWS::CloudFront::CloudFrontOriginAccessIdentity
157         Properties:
158             CloudFrontOriginAccessIdentityConfig:
159                 Comment: Access identity between CloudFront and S3 bucket
160
```

Serverless Cloudfront setup



The screenshot shows a code editor window with a file named "serverless.yml". The file contains YAML configuration for a CloudFront distribution. The code is color-coded, with blue for keywords like `Type`, `Properties`, etc., and grey for comments. The editor has a vertical scrollbar on the left and a status bar at the bottom.

```
## Specifying the CloudFront Distribution to serve your Web Application
WebAppCloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Origins:
        - DomainName: ${self:custom.s3BucketName}.s3.amazonaws.com
          ## An identifier for the origin which must be unique within the distribution
          Id: myS3Origin
          ## In case you don't want to restrict the bucket access use CustomOriginConfig and remove S3OriginConfig
          S3OriginConfig:
            OriginAccessIdentity: !Sub origin-access-identity/cloudfront/${OriginAccessIdentity}
          # CustomOriginConfig:
          #   HTTPPort: 80
          #   HTTPSPort: 443
          #   OriginProtocolPolicy: https-only
      Enabled: true
      IPV6Enabled: true
      HttpVersion: http2
      ## Uncomment the following section in case you are using a custom domain
      # Aliases:
      # - mysite.example.com
    DefaultRootObject: index.html
```

Serverless Cloudfront setup

```
YML serverless.yml ×
184      ## Since the Single Page App is taking care of the routing we need to make sure ever path is served with index.html
185      ## The only exception are files that actually exist e.h. app.js, reset.css
186  CustomErrorResponses:
187      - ErrorCode: 404
188          ResponseCode: 200
189          ResponsePagePath: /index.html
190  DefaultCacheBehavior:
191      AllowedMethods: [ 'GET', 'HEAD', 'OPTIONS' ]
192      CachedMethods: [ 'GET', 'HEAD', 'OPTIONS' ]
193  ForwardedValues:
194      Headers:
195          - Access-Control-Request-Headers
196          - Access-Control-Request-Method
197          - Origin
198          - Authorization
199      ## Defining if and how the QueryString and Cookies are forwarded to the origin which in this case is S3
200      QueryString: false
201      Cookies:
202          Forward: none
203      ## The origin id defined above
204      TargetOriginId: myS3origin
205      ## The protocol that users can use to access the files in the origin. To allow HTTP use `allow-all`
206      ViewerProtocolPolicy: redirect-to-https
```

Serverless Cloudfront setup

```
serverless.yml ×

206      Compress: true
207      DefaultTTL: 0
208      ## The certificate to use when viewers use HTTPS to request objects.
209      ViewerCertificate:
210          CloudFrontDefaultCertificate: 'true'
211      ## Uncomment the following section in case you want to enable logging for CloudFront requests
212      # Logging:
213      #   IncludeCookies: 'false'
214      #   Bucket: mylogs.s3.amazonaws.com
215      #   Prefix: myprefix
216
217      ## In order to print out the hosted domain via `serverless info` we need to define the DomainName output for CloudFormation
218      Outputs:
219          WebAppS3BucketOutput:
220              Value:
221                  'Ref': WebAppS3Bucket
222          WebAppCloudFrontDistributionOutput:
223              Value:
224                  'Fn::GetAtt': [ WebAppCloudFrontDistribution, DomainName ]
```

Serverless Cloudfont setup

```
package.json ×

21 ► "client:deploy": "sls client deploy --no-config-change --no-policy-change --no-cors-change",
22 ► "client:build:deploy": "npm run build && npm run client:deploy",
23 ► "cloudfont:setup": "sls deploy",
24 ► "cloudfont:domainInfo": "sls domainInfo",
25 ► "cloudfont:invalidateCache": "sls invalidateCloudFrontCache",
26 ► "cloudfont:build:deploy": "npm run client:build:deploy && npm run cloudfont:invalidateCache",
27 ► "cloudfont:update:build:deploy": "npm run cloudfont:setup && npm run cloudfont:build:deploy"
28 },
```

Serverless Cloudfront setup

```
+ rs-school-app git:(master) ✘ npm run cloudfront:update:build:deploy

> rs-school-app@0.1.0 cloudfront:update:build:deploy /Users/oleksandr_bondarenko/Projects/rs-school/rs-school-app
> npm run cloudfront:setup && npm run cloudfront:build:deploy

> rs-school-app@0.1.0 cloudfront:setup /Users/oleksandr_bondarenko/Projects/rs-school/rs-school-app
> sls deploy

Serverless: Packaging service...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Validating template...
Serverless: Updating Stack...
Service Information
service: rs-school-app
stage: dev
region: us-east-1
stack: rs-school-app-dev
resources: 6
api keys:
  None
endpoints:
  None
functions:
  None
layers:
  None
```

Serverless Cloudfont setup

```
> rs-school-app@0.1.0 client:build:deploy /Users/oleksandr_bondarenko/Projects/rs-school/rs-school-app
> npm run build && npm run client:deploy
```

```
> rs-school-app@0.1.0 build /Users/oleksandr_bondarenko/Projects/rs-school/rs-school-app
> react-scripts build
```

Creating an optimized production build...

Compiled successfully.

File sizes after gzip:

39.39 KB	build/static/js/ 2.6fd5d4d1.chunk.js
777 B	build/static/js/ runtime-main.40fb83ce.js
628 B (-4 B)	build/static/js/ main.778242cf.chunk.js
547 B	build/static/css/ main.5f361e03.chunk.css

The project was built assuming it is hosted at [/](#).

You can control this with the [homepage](#) field in your [package.json](#).

The [build](#) folder is ready to be deployed.

Serverless Cloudfront setup

```
> rs-school-app@0.1.0 client:deploy /Users/oleksandr_bondarenko/Projects/rs-school/rs-school-app
> sls client deploy --no-config-change --no-policy-change --no-cors-change

Serverless: This deployment will:
Serverless: - Upload all files from 'build' to bucket 'rs-school-app'
? Do you want to proceed? true
Serverless: Looking for bucket...
Serverless: Bucket found...
Serverless: Deleting all objects from bucket...
Serverless: Retaining existing bucket configuration...
Serverless: Retaining existing bucket policy...
Serverless: Retaining existing tags...
Serverless: Retaining existing bucket CORS configuration...
Serverless: Uploading client files to bucket...
Serverless: Success! Your site should be available at http://rs-school-app.s3-website-us-east-1.amazonaws.com/
```

```
> rs-school-app@0.1.0 cloudfront:invalidateCache /Users/oleksandr_bondarenko/Projects/rs-school/rs-school-app
> sls invalidateCloudFrontCache

Serverless: Web App Domain: d8rf70pc3i3ck.cloudfront.net
Serverless: Invalidating CloudFront distribution with id: E1WSEDHYC5Z450
Serverless: Successfully invalidated CloudFront cache
```



**WRAPPING UP.
HOW TO DESTROY INFRASTRUCTURE****

Destroy Infrastructure

Amazon S3

▼ Account snapshot

Last updated: Dec 21, 2021 by Storage Lens. Metrics are generated every 24 hours. [Learn more](#)

[View Storage Lens dashboard](#)

Total storage	Object count	Avg. object size	You can enable advanced metrics in the "default-account-dashboard" configuration.
232.7 MB	489	487.3 KB	

Buckets (10) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

[C](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

<input type="text"/> rs	X	1 match	
Name	AWS Region	Access	Creation date
app-rs-school	US East (N. Virginia) us-east-1	⚠️ Public	December 22, 2021, 17:07:27 (UTC+02:00)

Destroy Infrastructure

Buckets (10) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

rs X 1 match

Name	AWS Region	Access	Creation date
app-rs-school	US East (N. Virginia) us-east-1	⚠ Public	December 22, 2021, 17:07:27 (UTC+02:00)

Successfully deleted bucket "app-rs-school"

Amazon S3

▼ Account snapshot

Last updated: Dec 21, 2021 by Storage Lens. Metrics are generated every 24 hours. [Learn more](#)

Total storage 232.7 MB Object count 489 Avg. object size 487.3 KB You can enable advanced metrics in the "default-account-dashboard" configuration.

View Storage Lens dashboard

Buckets (9) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

rs X 0 matches < 1 > ⚙️

Name	AWS Region	Access	Creation date
No matches			

We can't find a match.

[Clear filters](#)

Amazon S3 > app-rs-school > Empty bucket

Empty bucket [Info](#)

⚠ • Emptying the bucket deletes all objects in the bucket and cannot be undone.
• Objects added to the bucket while the empty bucket action is in progress might be deleted.
• To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

[Learn more](#)

ⓘ If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket. [Learn more](#) [Go to lifecycle rule configuration](#)

Permanently delete all objects in bucket "app-rs-school"?

To confirm deletion, type **permanently delete** in the text input field.

Cancel Empty

Amazon S3 > app-rs-school > Delete bucket

Delete bucket [Info](#)

⚠ • Deleting a bucket cannot be undone.
• Bucket names are unique. If you delete a bucket, another AWS user can use the name.

[Learn more](#)

Delete bucket "app-rs-school"?

To confirm deletion, enter the name of the bucket in the text input field.

Cancel Delete bucket

Destroy Infrastructure

CloudFront > Distributions

Distributions (2) Info					C	Enable	Disable	Delete	Create distribution
	ID	Description	Domain name	Alternate domain names				Origins	
<input checked="" type="checkbox"/>	EWUM4ARWOPF8L	-	daeejxqnyrtf0.cloud...	-				app-rs-school.s3.us-east-1.amazonaws.com	

Are you sure you want to disable this 1 distribution? When it's disabled, the distribution is offline and cannot respond to requests. You can enable the distribution later to restore it.

EWUM4ARWOPF8L

[Cancel](#) [Disable](#)

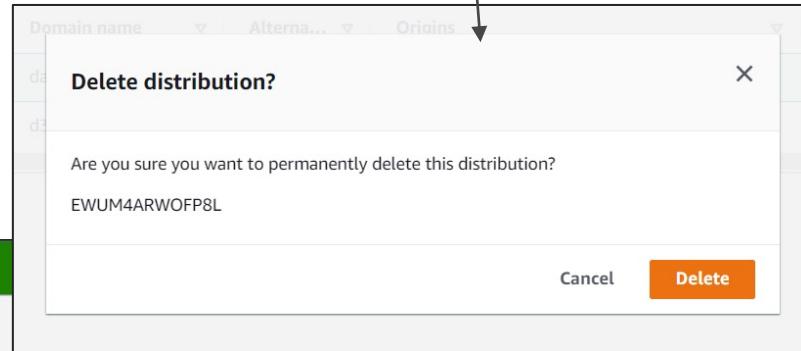
Destroy Infrastructure

CloudFront > Distributions

Distributions (2) [Info](#)

Search all distributions

ID	Description	Domain name	Alternate domain names	Origins	Status	Last modified
<input checked="" type="checkbox"/> EWUM4ARWOFP8L	-	daeejxqnyrtf0.cloudfront.net	-	app-rs-school.s3.us-east-1.amazonaws.com	Disabled	December 22, 2021 at 3:57:48 PM UTC



Deleted distribution: EWUM4ARWOFP8L.

CloudFront > Distributions

Distributions (1) [Info](#)

Search all distributions

ID	Description	Domain name	Alternate domain names	Origins	Status	Last modified

Destroy Infrastructure

CloudFormation > Stacks

Stacks (1)

Delete (highlighted with a red underline)

Filter by stack name

Active View nested

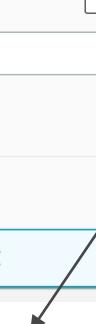
Stack name Status Created time Description

rs-school-app-dev	✓ UPDATE_COMPLETE	2020-10-09 13:36:19 UTC+0300	The AWS CloudFormation template for t...
-------------------	-------------------	------------------------------	--

Delete rs-school-app-dev?

Deleting this stack will delete all stack resources. Resources will be deleted according to their DeletionPolicy. [Learn more](#)

Cancel Delete stack





THANK YOU