



Deploying a Web Application to AWS S3 and Cloud Front

Open Lab | Digital Summit 2019



Deploying a Web Application to AWS S3 and Cloud Front

Introduction

The goal of this document is to create S3 bucket, configure CloudFront which is a content delivery network provided by AWS. In this document we will install and configure all necessary requirements for deploying and accessing a sample web application using CloudFront.

This guide was prepared by **Miracle's Innovation Labs!**

Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the lab (Both PC and MAC are compatible). The following pre-requisites will help you to make the Hands-on Lab experience easier.

- Active email ID for registering with AWS
- Download and Install Node JS

Technology Involved

- AWS Account
- Git
- Node JS
- Angular CLI

Lab Steps

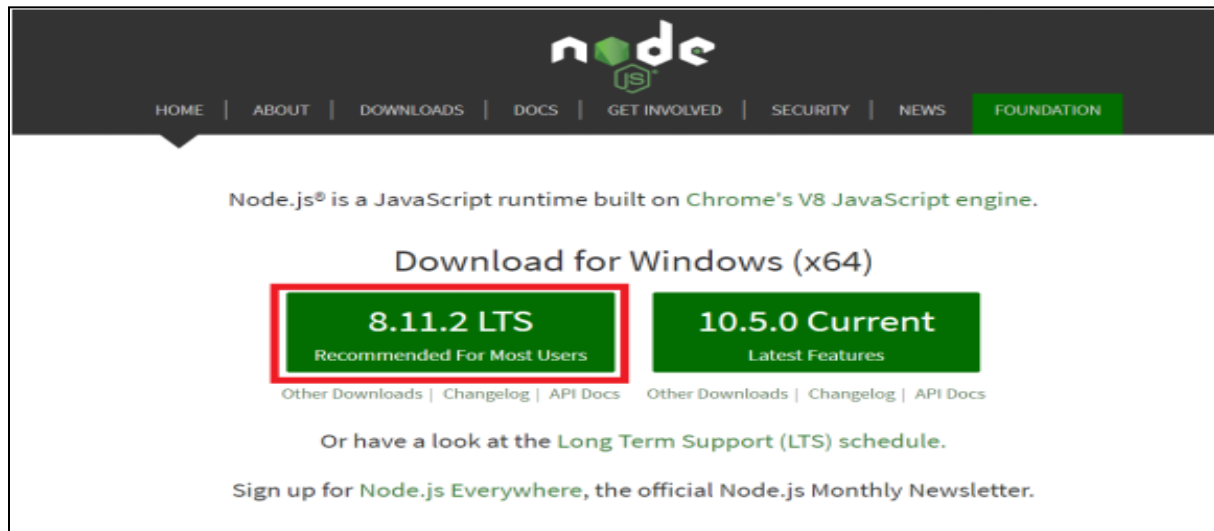
So, let us get started!

In this document, we will install Node JS and Angular CLI which is required to run the application. Also, we will create S3 bucket which is provided by AWS as storage service, configure CloudFront with the S3 bucket and fill the required fields such as granting permissions to access S3 bucket and protocol policies. Once the configurations are done, we will show you how to access the deployed application through CloudFront.

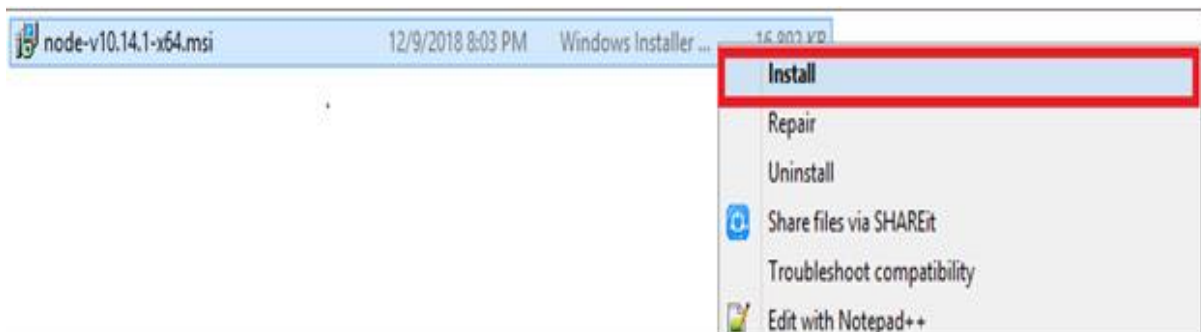
Step #1 | Installation of Node JS

Open your browser and hit the following URL: <https://nodejs.org/en/>

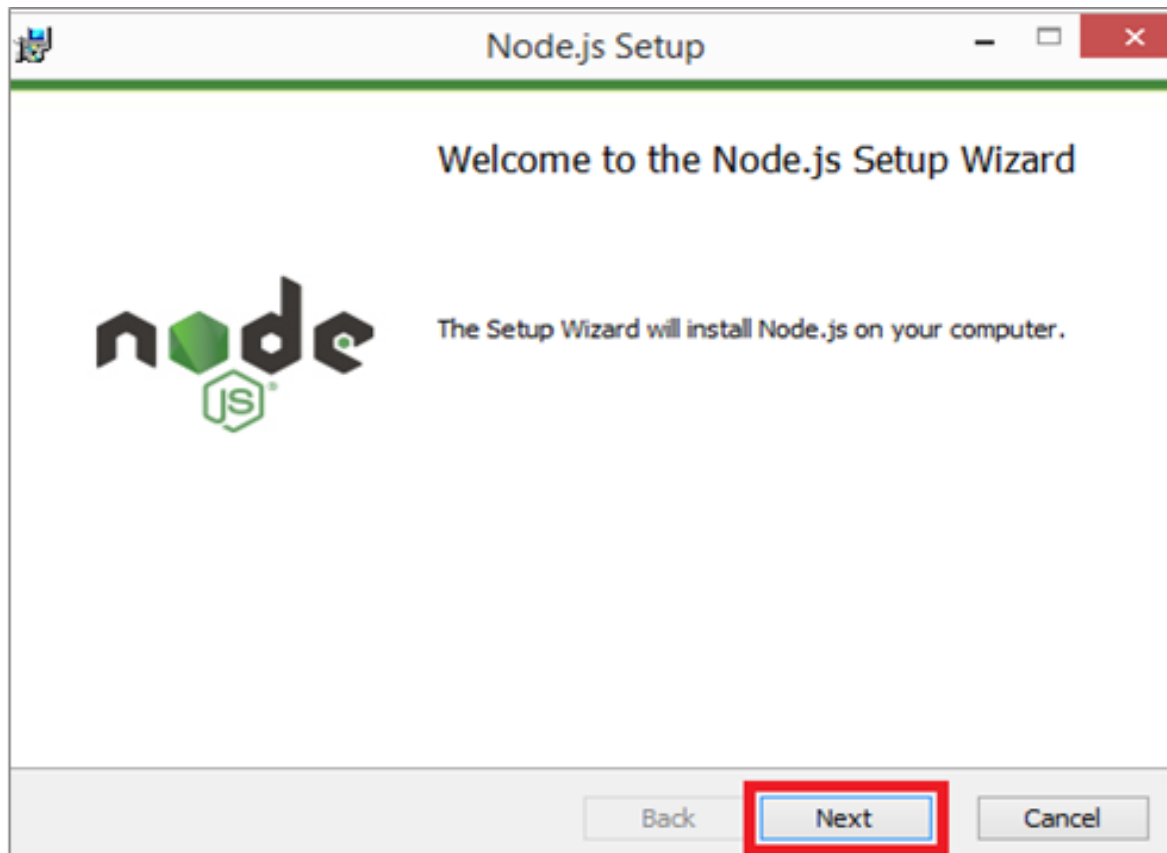
Click on **8.11.2 LTS** to download **Node JS** msi file.



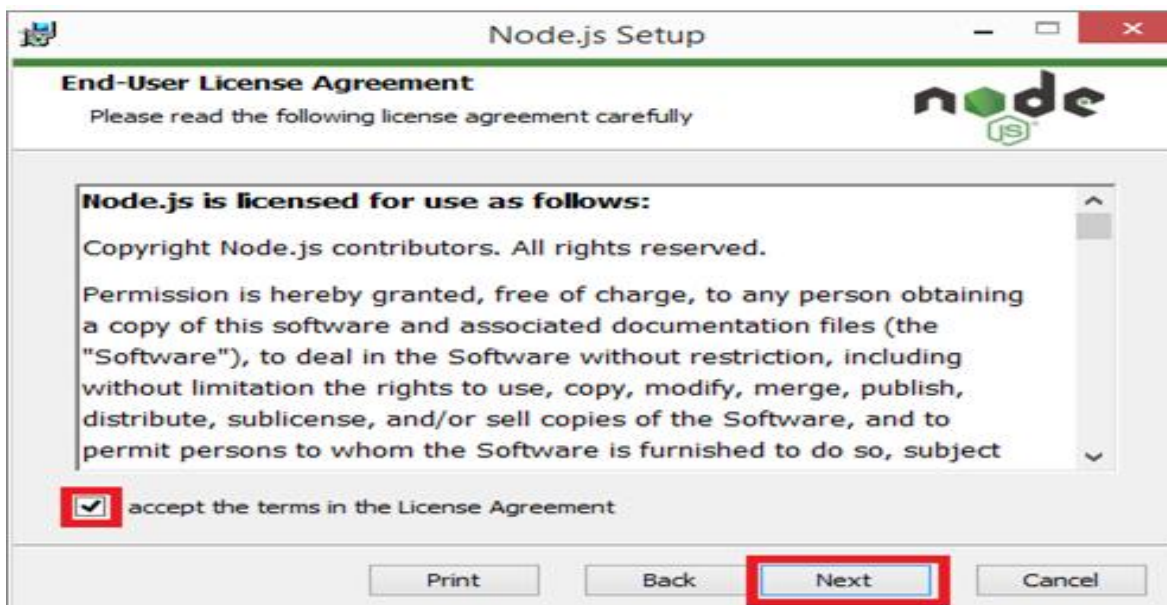
Right click on the installer and click on **Install**.



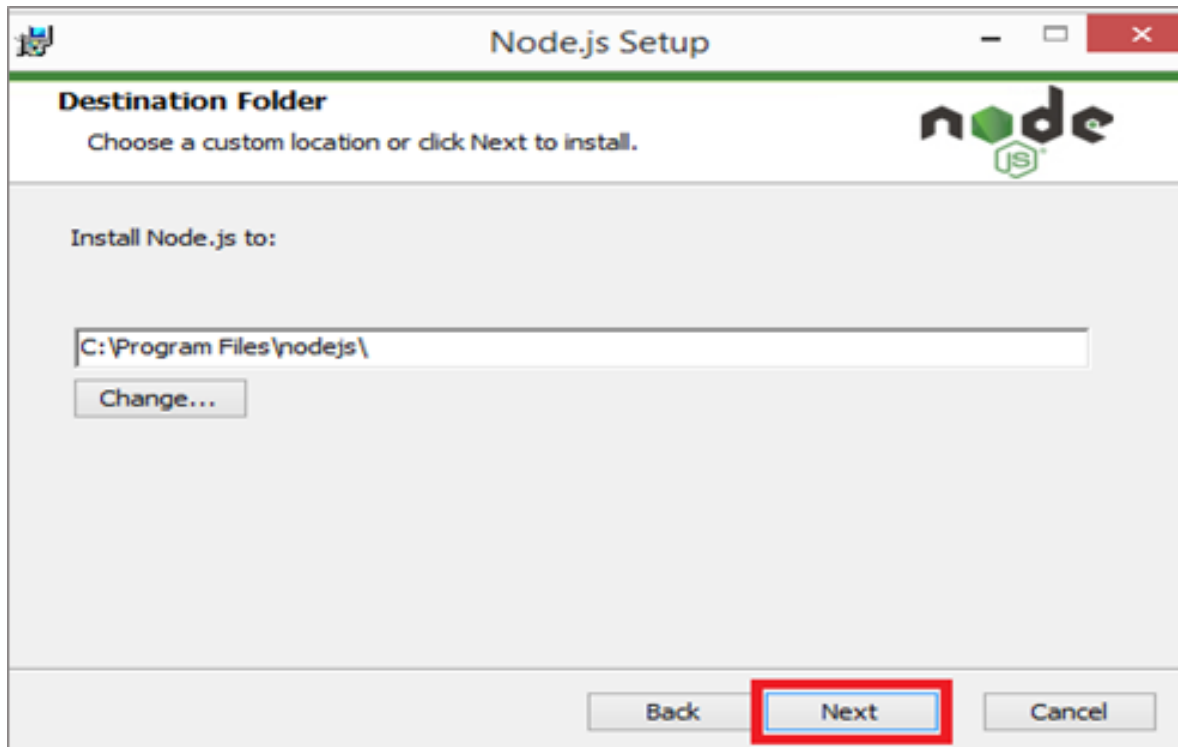
After selecting Install option the setup page is displayed, click on **Next**.



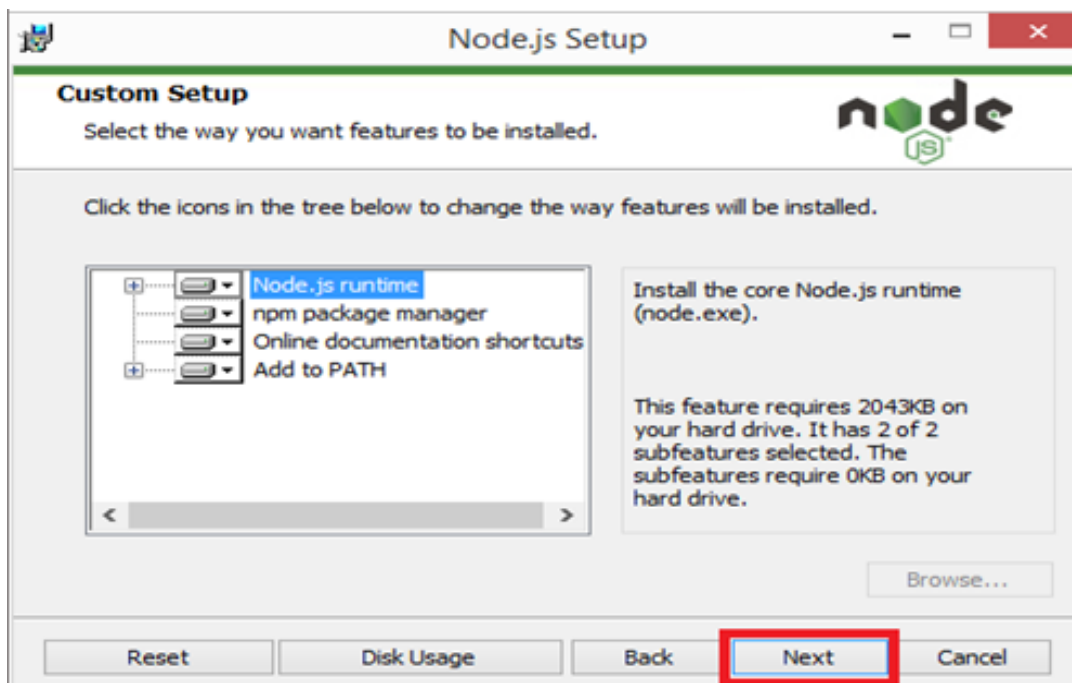
Accept the terms and click on Next.



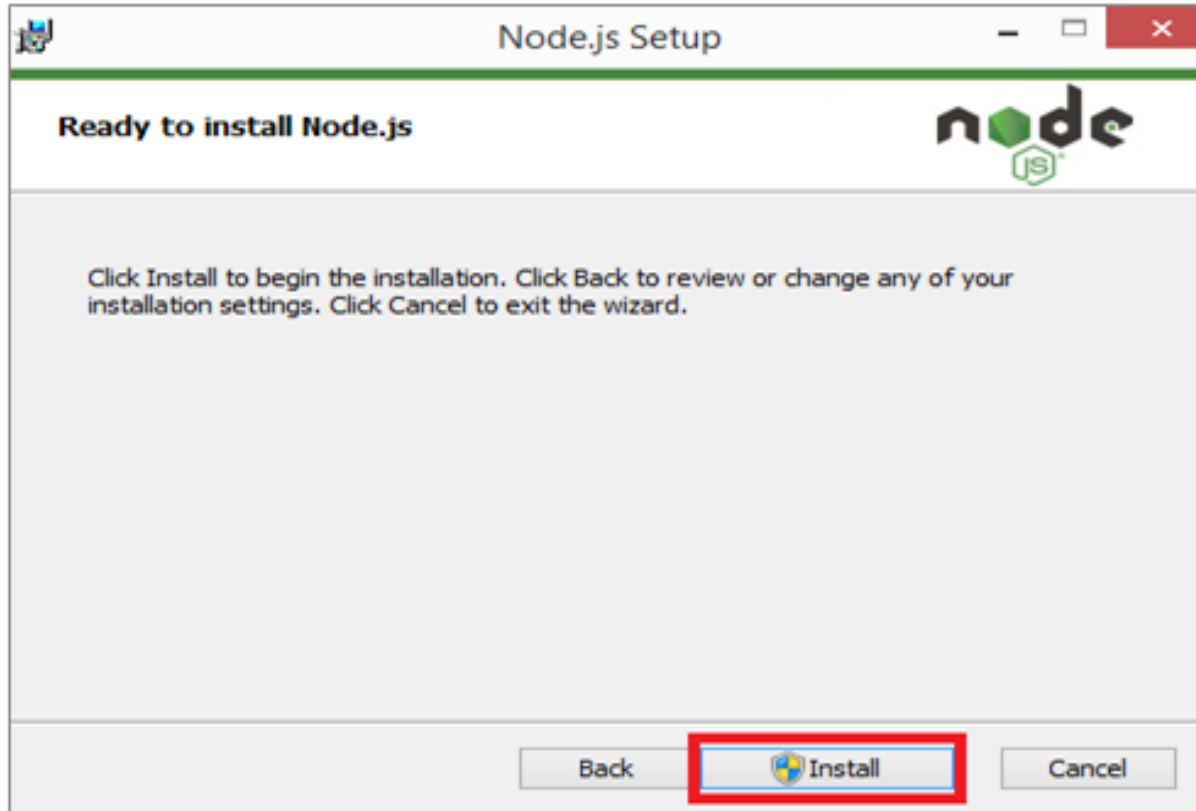
If you want to change the installation path, you can change your desired location. For now we are not changing any path, click on **Next**.



Make sure you select Node.js runtime and click on Next.



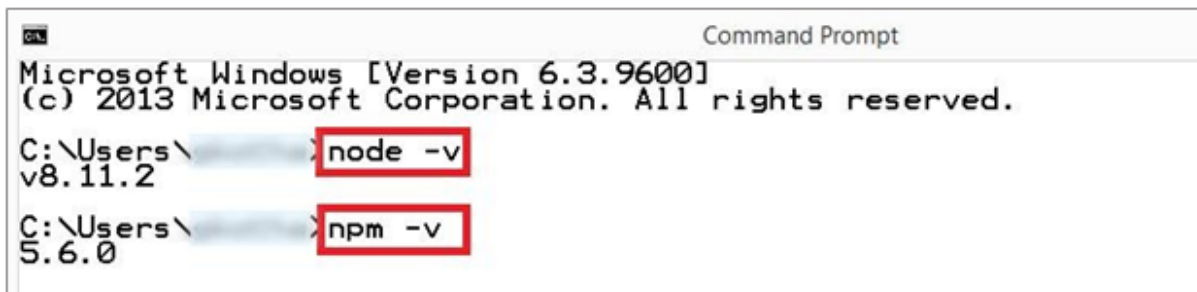
Click on **Install** and then Node JS installation will be initialized.



After completion of installation, click on **Finish**. Open command line interface in your local machine.

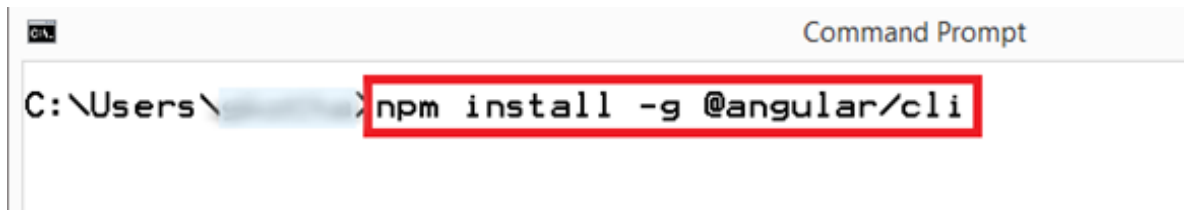
Check whether Node JS is installed: **node -v**

Check npm version: **npm -v**



Step #2 | Installation of Angular CLI

To install Angular CLI, enter command in your command prompt: **npm install -g @angular/cli**

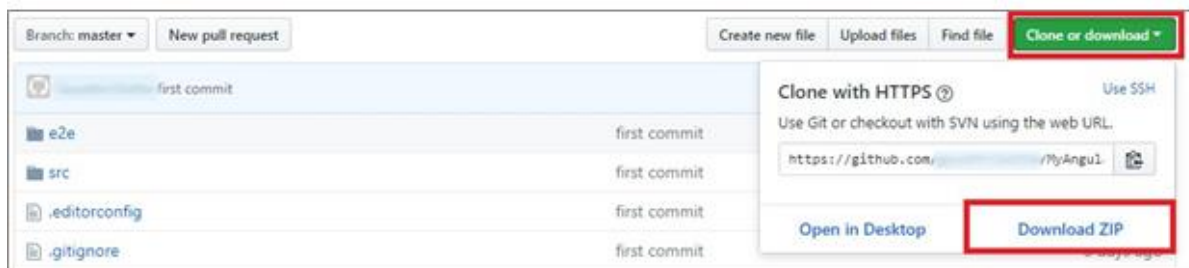


```
C:\Users\> npm install -g @angular/cli
```

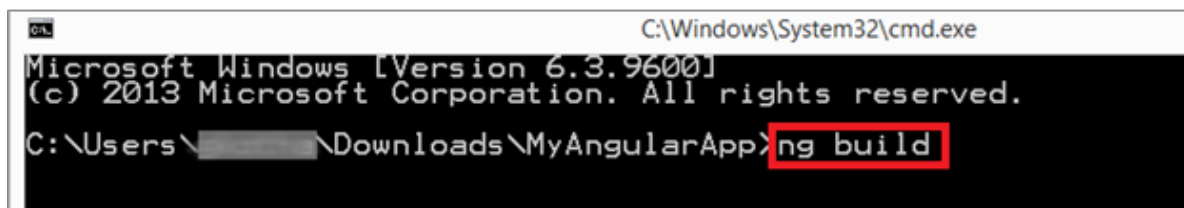
To check Angular CLI version: **ng -v**

Step #3 | Compiling Angular Application

After completion of Angular CLI, open your source code in GitHub and click on Clone or Download and then click on Download ZIP.



After downloading source code, set the path in command prompt to that folder and enter the commands **npm install** and **ng build** to download necessary packages and compile the Angular application as shown below. The **dist** folder is created in the source code.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\> ng build
```

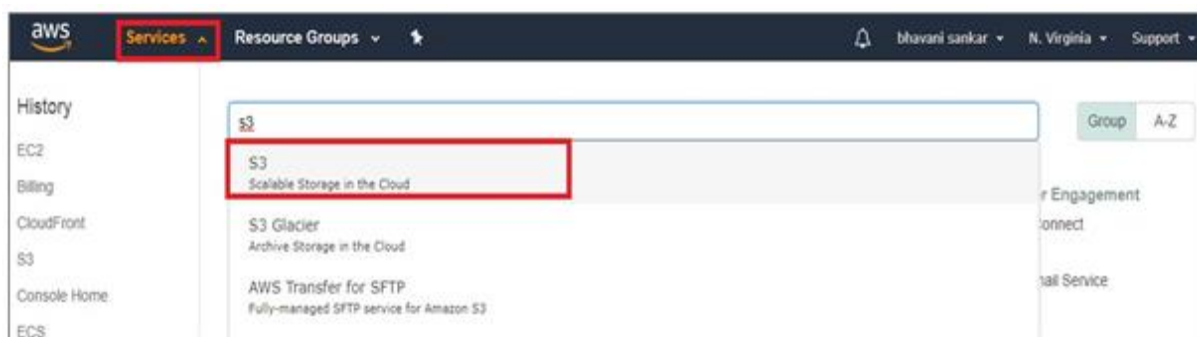
After compiling, **dist** folder will be added as shown below.

.git	12/6/2018 3:03 PM	File folder	
dist	12/9/2018 10:06 PM	File folder	
e2e	12/5/2018 11:11 PM	File folder	
node_modules	12/5/2018 11:17 PM	File folder	
src	12/5/2018 11:11 PM	File folder	
.editorconfig	12/5/2018 11:11 PM	EDITORCONFIG File	1 KB
.gitignore	12/5/2018 11:11 PM	Text Document	1 KB
angular.json	12/5/2018 11:11 PM	JSON File	4 KB
package.json	12/5/2018 11:17 PM	JSON File	2 KB
package-lock.json	12/5/2018 11:17 PM	JSON File	361 KB
README.md	12/5/2018 11:11 PM	MD File	2 KB
tsconfig.json	12/5/2018 11:11 PM	JSON File	1 KB
tslint.json	12/5/2018 11:11 PM	JSON File	3 KB

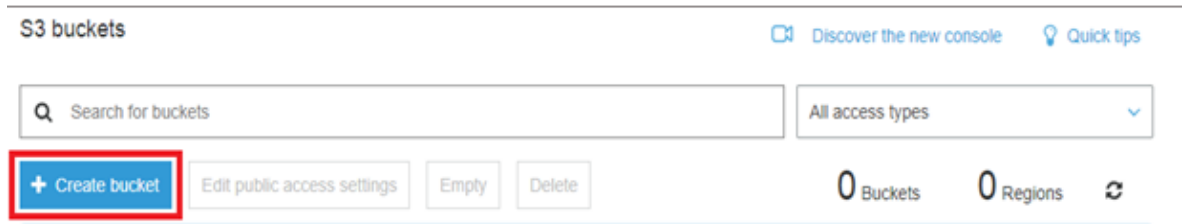
After completion of Node JS, Angular CLI installation and compiling the source code, it's time to create S3 bucket where you need to upload the dist folder files.

Step #4 | Creation of S3 Bucket

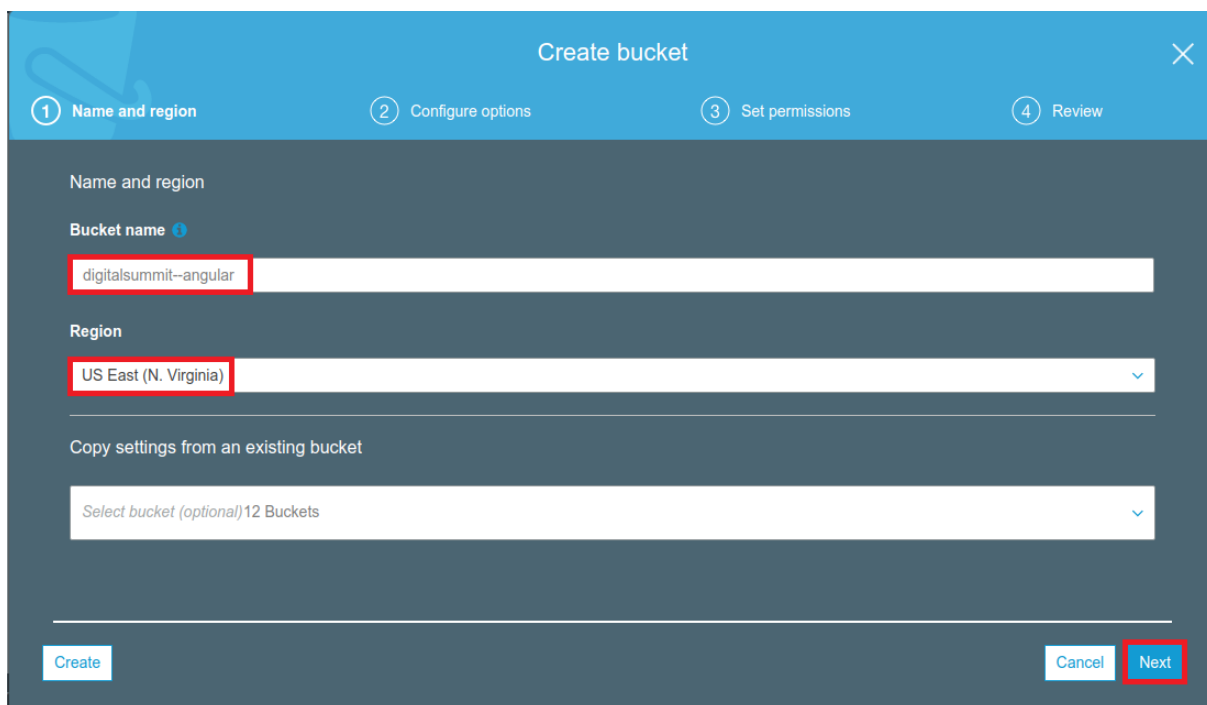
In order to serve the application, first we need to build a compiled version of Angular application using the command `ng build` which will return the compiled version of Angular code. Now, you need to create a bucket for storing this Angular code in S3. For that, search for S3 service in AWS dashboard as follows.



Now you need to create a bucket in that service by selecting Create bucket button as shown below.



Once you click on Create bucket, you need to provide some details like Bucket name and Region and click on Next.



Configure options page will provide options to choose versions, logging all server activities, etc. As of now we can skip these and click on Next as shown below.

The screenshot shows the 'Create bucket' wizard with four steps: 1. Name and region (checked), 2. Configure options (active), 3. Set permissions, and 4. Review. The 'Configure options' section includes 'Versioning' (Keep all versions of an object in the same bucket), 'Server access logging' (Log requests for access to your bucket), 'Tags' (You can use tags to track project costs), and 'Object-level logging' (Record object-level API activity using AWS CloudTrail for an additional cost). The 'Default encryption' section is partially visible. At the bottom right, there are 'Previous' and 'Next' buttons, with the 'Next' button highlighted by a red box.

In Set permissions page, you can configure the public access settings for this bucket. For now leave the default settings and click on Next.

The screenshot shows the 'Create bucket' wizard with four steps: 1. Name and region (checked), 2. Configure options (checked), 3. Set permissions (active), and 4. Review. The 'Set permissions' section includes 'Public access settings for this bucket' (Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. Learn more), 'Manage public access control lists (ACLs) for this bucket' (Block new public ACLs and uploading public objects (Recommended), Remove public access granted through public ACLs (Recommended)), 'Manage public bucket policies for this bucket' (Block new public bucket policies (Recommended), Block public and cross-account access if bucket has public policies (Recommended)), and 'Manage system permissions' (Do not grant Amazon S3 Log Delivery group write access to this bucket). At the bottom right, there are 'Previous' and 'Next' buttons, with the 'Next' button highlighted by a red box.

Review the configuration details and click on **Create bucket**.

Create bucket

1 Name and region 2 Configure options 3 Set permissions 4 Review

Bucket name digitalsummit-angular Region US East (N. Virginia)

Options Edit

Versioning	Disabled
Server access logging	Disabled
Tagging	0 Tags
Object-level logging	Disabled
Default encryption	None
CloudWatch request metrics	Disabled
Object lock	Disabled

Permissions Edit

Block new public ACLs and uploading public objects True

Previous Create bucket

After completion of creating bucket, it will show you the bucket name in the list of buckets that you have. Select the bucket that you have created.

S3 buckets [Discover the console](#)

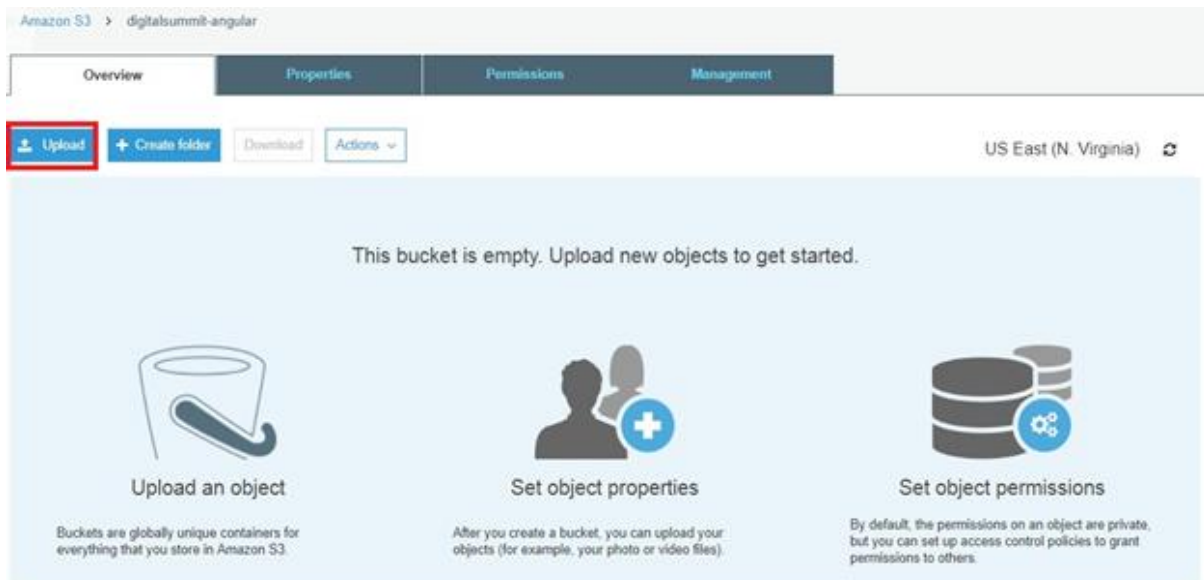
Q digital All access types

+ Create bucket Edit public access settings Empty Delete 1 Buckets 1 Regions

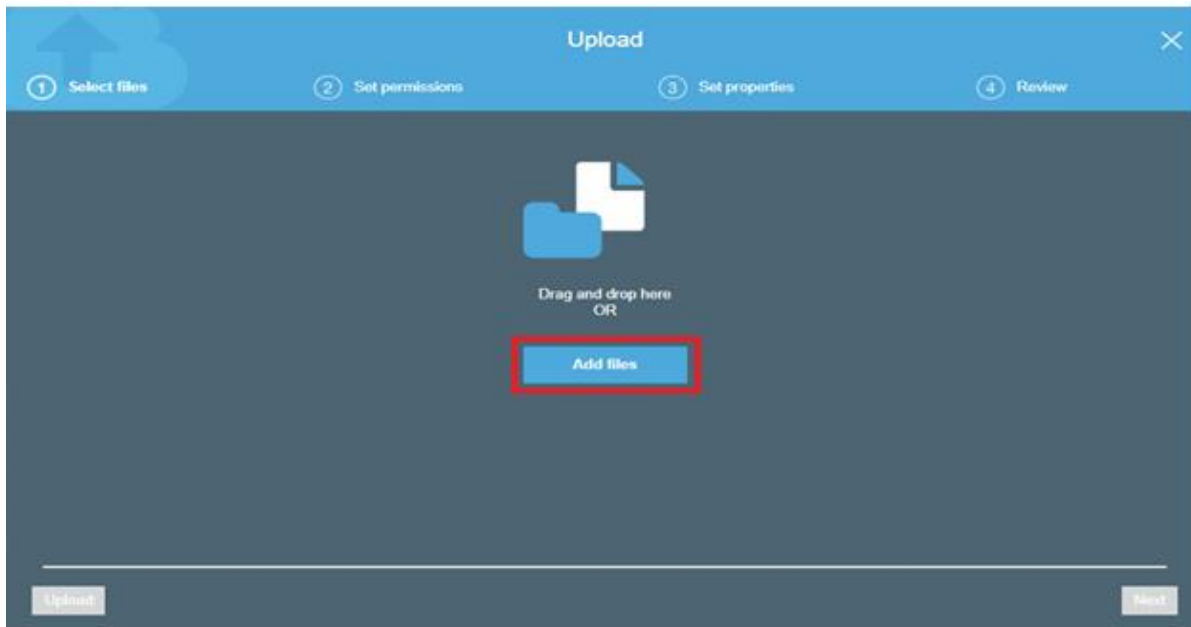
Bucket name	Access	Region	Date created
digitalsummit-angular	Bucket and objects not public	US East (N. Virginia)	Dec 4, 2019 2:21:10 PM GMT+0530

Step #5 | Uploading Objects to S3 Bucket

Your S3 bucket is empty and your S3 dashboard will be displayed as shown below. Click on Upload to add all build files from your local machine to S3 dashboard.

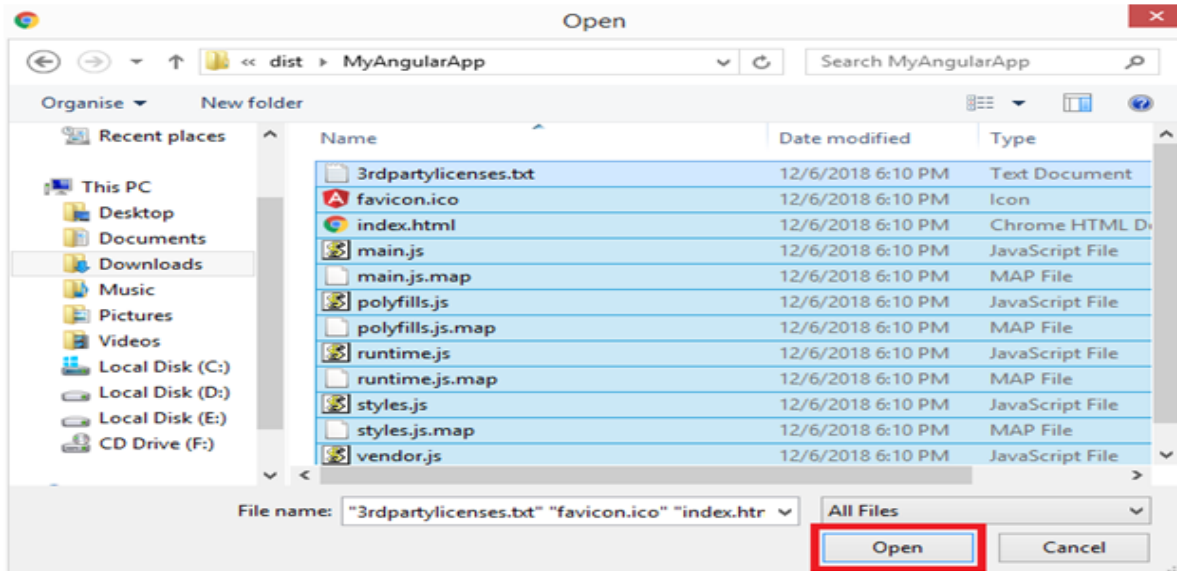


Click on Add files as below

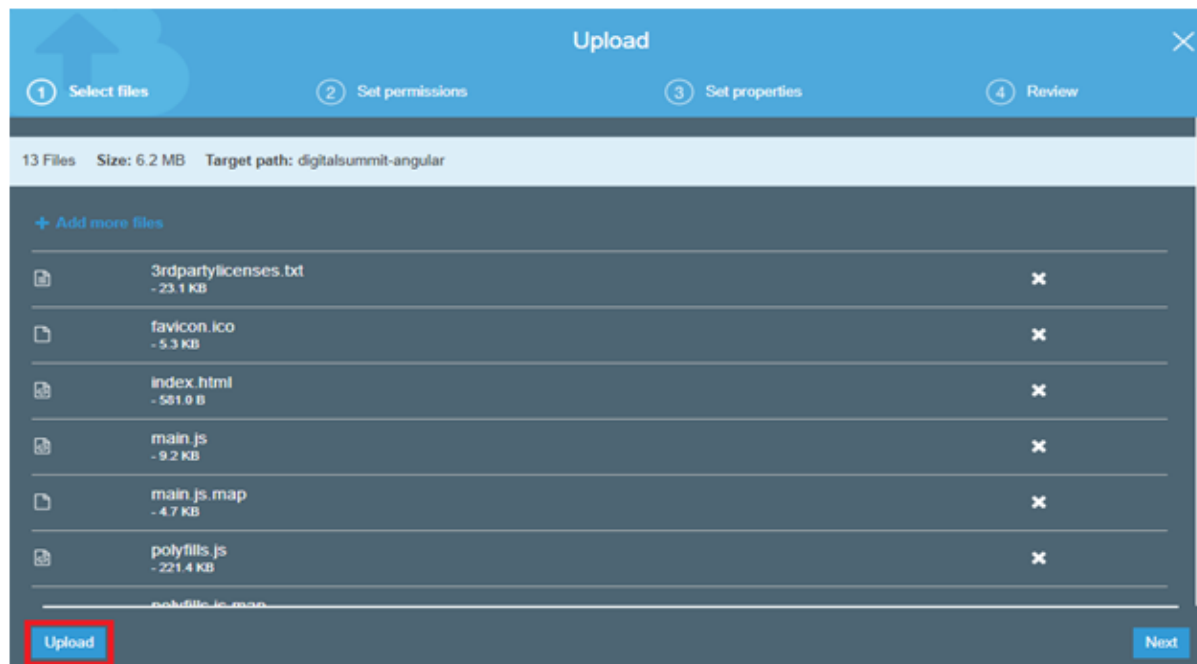


Select all files from your **dist** folder and click on **Open**.

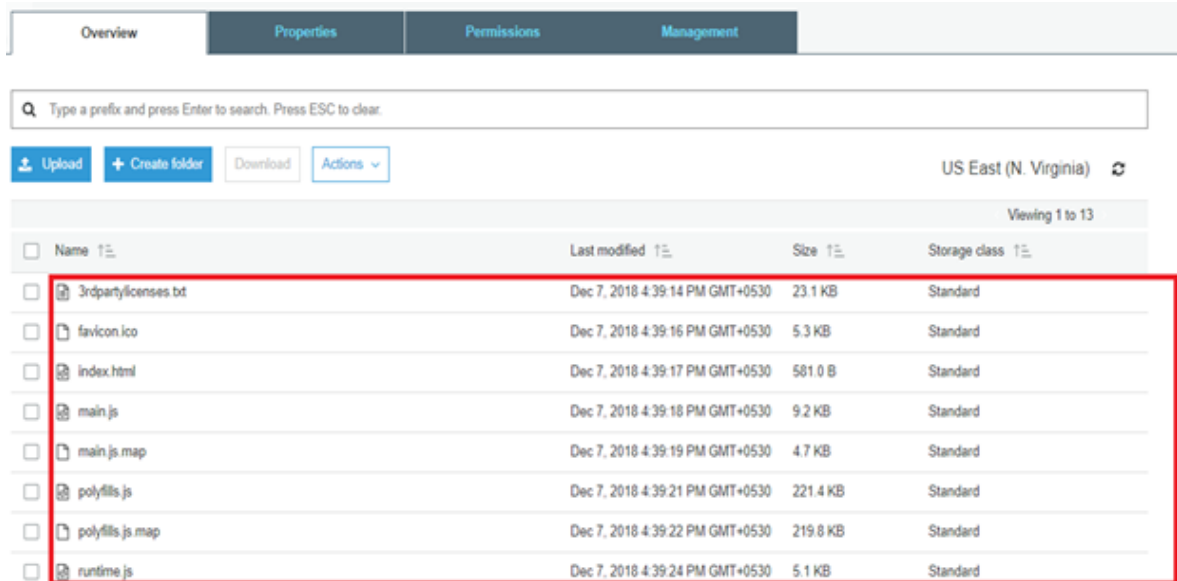
Note - After performing **ng build**, you will get dist folder. Open dist folder and select all the files as shown below.



Click on **Upload**.



Now the build files got uploaded as below.

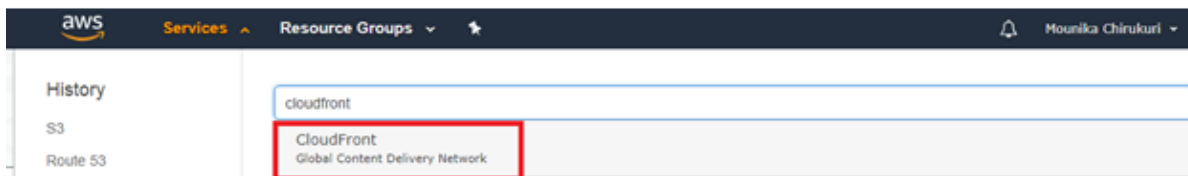


Overview				
Properties				
Permissions				
Management				
Q Type a prefix and press Enter to search. Press ESC to clear.				
Upload Create folder Download Actions				
US East (N. Virginia)				
Viewing 1 to 13				
Name	Last modified	Size	Storage class	
3rdpartylicenses.txt	Dec 7, 2018 4:39:14 PM GMT+0530	23.1 KB	Standard	
favicon.ico	Dec 7, 2018 4:39:16 PM GMT+0530	5.3 KB	Standard	
index.html	Dec 7, 2018 4:39:17 PM GMT+0530	581.0 B	Standard	
main.js	Dec 7, 2018 4:39:18 PM GMT+0530	9.2 KB	Standard	
main.js.map	Dec 7, 2018 4:39:19 PM GMT+0530	4.7 KB	Standard	
polyfills.js	Dec 7, 2018 4:39:21 PM GMT+0530	221.4 KB	Standard	
polyfills.js.map	Dec 7, 2018 4:39:22 PM GMT+0530	219.8 KB	Standard	
runtime.js	Dec 7, 2018 4:39:24 PM GMT+0530	5.1 KB	Standard	

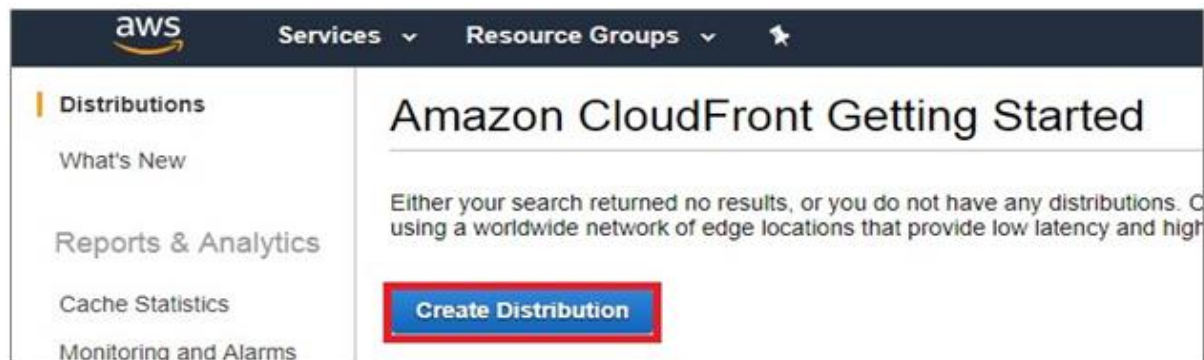
Once you are good with configuring the S3, it's time to configure CDN.

Step #6 | Creating Distribution in CloudFront

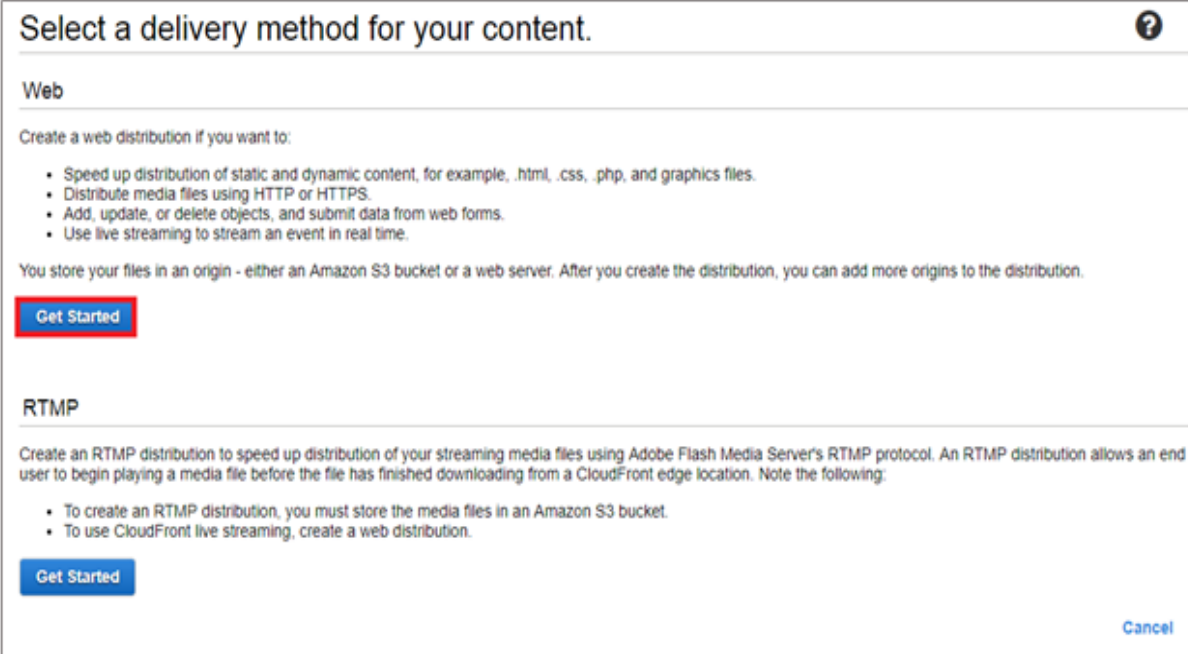
In order to configure CloudFront, search for CloudFront service in AWS dashboard and click on CloudFront.



Click on **Create Distribution** as shown in below to create a distribution service.



Once you click on Create Distribution, it will provide you with two options where one is for distributing the content to a web service and the other for streaming media. Among these two, select **Web** service as shown below.



Select a delivery method for your content. ?

Web

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin - either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

Get Started

RTMP

Create an RTMP distribution to speed up distribution of your streaming media files using Adobe Flash Media Server's RTMP protocol. An RTMP distribution allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location. Note the following:

- To create an RTMP distribution, you must store the media files in an Amazon S3 bucket.
- To use CloudFront live streaming, create a web distribution.

Get Started

Cancel

Once you select the web service, it will show the list of options where you need to select S3 bucket name for the origin domain name. Please provide the remaining details as shown below. Now, you need to provide the origin name where bucket lies, as a good practice and restrict access to the bucket where only CloudFront can access the application. Also select the option to update the bucket policy as shown in the picture.

Now, configure the application to allow both HTTP and HTTPS unless we have trusted certificates which can be configured.

Default Cache Behavior Settings

Path Pattern	Default (*)	
Viewer Protocol Policy	<input checked="" type="radio"/> HTTP and HTTPS <input type="radio"/> Redirect HTTP to HTTPS <input type="radio"/> HTTPS Only	
Allowed HTTP Methods	<input checked="" type="radio"/> GET, HEAD <input type="radio"/> GET, HEAD, OPTIONS <input type="radio"/> GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE	
Field-level Encryption Config	<input type="text"/>	
Cached HTTP Methods	GET, HEAD (Cached by default)	
Cache Based on Selected Request Headers	<input type="text"/> None (Improves Caching) Learn More	
Object Caching	<input checked="" type="radio"/> Use Origin Cache Headers <input type="radio"/> Customize Learn More	

Create Distribution

Origin Settings

Origin Domain Name	digitalsummit-angular.s3.amazonaws.co							
Origin Path	<input type="text"/>							
Origin ID	S3-digitalsummit-angular							
Restrict Bucket Access	<input checked="" type="radio"/> Yes <input type="radio"/> No							
Origin Access Identity	<input checked="" type="radio"/> Create a New Identity <input type="radio"/> Use an Existing Identity							
Comment	access-identity-digitalsummit-angular.s3							
Grant Read Permissions on Bucket	<input checked="" type="checkbox"/> Yes, Update Bucket Policy <input type="checkbox"/> No, I Will Update Permissions							
Origin Custom Headers	<table> <thead> <tr> <th>Header Name</th> <th>Value</th> <th></th> </tr> </thead> <tbody> <tr> <td><input type="text"/></td> <td><input type="text"/></td> <td></td> </tr> </tbody> </table>	Header Name	Value		<input type="text"/>	<input type="text"/>		
Header Name	Value							
<input type="text"/>	<input type="text"/>							

If you have a purchased domain, you can enter the domain name in **Alternate Domain Names** field to access your application with that domain name. For now, we are leaving that field.

Price Class

Use All Edge Locations (Best Performance) ▾ ⓘ

AWS WAF Web ACL

None ▾ ⓘ

Alternate Domain Names
(CNAMEs)

ⓘ

SSL Certificate

* Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as https://d111111abcdef8.cloudfront.net/logo.jpg). Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.

Custom SSL Certificate (example.com).

Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.jpg. You can use a certificate stored in AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.

ⓘ

Request or Import a Certificate with ACM

[Learn more about using custom SSL/TLS certificates with CloudFront.](#)
[Learn more about using ACM.](#)

Once that is done, we need to provide the default landing page for the application as shown in the configuration and then click on **Create Distribution**.

Supported HTTP Versions

☒ HTTP/2, HTTP/1.1, HTTP/1.0

☐ HTTP/1.1, HTTP/1.0

Default Root Object

index.html

Logging

☐ On

☒ Off

Bucket for Logs

Log Prefix

Cookie Logging

☐ On

☒ Off

Enable IPv6

☒

[Learn more](#)

Comment

Distribution State

☒ Enabled

☐ Disabled

Cancel

Back

Create Distribution

Once the distribution is created, it will direct you to the page where our distribution details are available. Initially, distribution will be in progress status. It will take around 15-20 minutes to complete the distribution.

<div> Create Distribution Distribution Settings Delete Enable Disable </div>							
<div> Viewing : Any Delivery Method Any State </div>							
	Delivery Metho	ID	Domain Name	Origin	CNAMEs	Status	State
<input type="checkbox"/>	Web	E1F9G3HWBGSDVK	dga08vjkmss59.cloudfront.net	webinartestui.s3.a	-	Deployed	Enabled
<input type="checkbox"/>	Web	E275KGDZSHD34	d262c3ntpg8ys4.cloudfront.net	document-hub.mir	-	Deployed	Enabled
<input type="checkbox"/>	Web	E37Z3TE1U5SPW0	dr368bysylem3.cloudfront.net	miracle-demo.s3.a	-	Deployed	Enabled
<input type="checkbox"/>	Web	E39L9984GTHYNX	d3oyvsf9wf0mfi.cloudfront.net	digitalsummit--ang	-	In Progress	Enabled
<input type="checkbox"/>	Web	E3DXGH8BPAAUG7	dg4txwkuz2odo.cloudfront.net	dev.msscontracts.t	-	Deployed	Enabled
<input type="checkbox"/>	Web	E3TPY6B0HX2AF0	d32zzauijeokl5.cloudfront.net	miracle-automator	-	Deployed	Enabled
<input type="checkbox"/>	Web	EFGZ5DUDGZJ92	d3a3b8ivpekwrz.cloudfront.net	demo.americold.cc	-	Deployed	Enabled
<input type="checkbox"/>	Web	EMWUTMTHFSXVL	d2bjc88qdmnbjk.cloudfront.net	weatherbot19.s3.a	-	Deployed	Enabled

Once the distribution is successful, the status will be changed to **Deployed** as shown below.

<div> Create Distribution Distribution Settings Delete Enable Disable </div>							
<div> Viewing : Any Delivery Method Any State </div>							
	Delivery Metho	ID	Domain Name	Origin	CNAMEs	Status	State
<input type="checkbox"/>	Web	E1F9G3HWBGSDVK	dga08vjkmss59.cloudfront.net	webinartestui.s3.a	-	Deployed	Enabled
<input type="checkbox"/>	Web	E275KGDZSHD34	d262c3ntpg8ys4.cloudfront.net	document-hub.mir	-	Deployed	Enabled
<input type="checkbox"/>	Web	E37Z3TE1U5SPW0	dr368bysylem3.cloudfront.net	miracle-demo.s3.a	-	Deployed	Enabled
<input checked="" type="checkbox"/>	Web	E39L9984GTHYNX	d3oyvsf9wf0mfi.cloudfront.net	digitalsummit--ang	-	Deployed	Enabled
<input type="checkbox"/>	Web	E3DXGH8BPAAUG7	dg4txwkuz2odo.cloudfront.net	dev.msscontracts.t	-	Deployed	Enabled
<input type="checkbox"/>	Web	E3TPY6B0HX2AF0	d32zzauijeokl5.cloudfront.net	miracle-automator	-	Deployed	Enabled
<input type="checkbox"/>	Web	EFGZ5DUDGZJ92	d3a3b8ivpekwrz.cloudfront.net	demo.americold.cc	-	Deployed	Enabled
<input type="checkbox"/>	Web	EMWUTMTHFSXVL	d2bjc88qdmnbjk.cloudfront.net	weatherbot19.s3.a	-	Deployed	Enabled

Step #7 | Custom Error Page Creation

Select the distribution to update and choose Distribution Settings.

CloudFront Distributions

Create Distribution **Distribution Settings** Delete Enable Disable

Viewing: Any Delivery Method Any State << < Viewing 1 to 8 of 8 Items > >>

Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
<input type="checkbox"/> Web	E1F9G3HWBGSDVK	dga08vjkmss59.clou	-	webinartestu	-	Deployed	Enabled	2019-03-23 01:22 U1
<input type="checkbox"/> Web	E275KGDTZSHD34	d262c3ntpg8ys4.clou	-	document-hu	-	Deployed	Enabled	2019-03-13 20:10 U1
<input type="checkbox"/> Web	E37Z3TE1U5SPW0	dr368bysylem3.clou	-	miracle-dem	-	Deployed	Enabled	2019-05-31 00:07 U1
<input checked="" type="checkbox"/> Web	E39L9984GTHYNX	d3oyvsf9wf0mfi.clou	-	digitalsummi	-	Deployed	Enabled	2019-12-04 14:59 U1
<input type="checkbox"/> Web	E3DXGH8BPAAUG7	dg4txwkuz2odo.clou	-	dev.msscont	-	Deployed	Enabled	2019-06-02 00:05 U1
<input type="checkbox"/> Web	E3TPY6B0HX2AF0	d32zzaujveokl5.clou	-	miracle-autor	-	Deployed	Enabled	2019-06-21 01:02 U1
<input type="checkbox"/> Web	EFGZ5DUDGZJ92	d3a3b8lvpekwrz.clou	-	demo.americ	-	Deployed	Enabled	2019-06-28 19:30 U1
<input type="checkbox"/> Web	EMWUTMTHTFSXVL	d2bjc88qdmnbjk.clou	-	weatherbot1	-	Deployed	Enabled	2019-11-26 21:15 U1

Click on Create Custom Error Response to create a new error response.

Edit Custom Error Response

Custom Error Response Settings

HTTP Error Code **404: Not Found**

Error Caching Minimum TTL (seconds) **30**

Customize Error Response ☒ Yes ☐ No

Response Page Path **/index.html**

HTTP Response Code **200: OK**

Cancel **Yes, Edit**

Customize the error response and provide the path for the landing page to which it should redirect http to https when the error is encountered.

CloudFront Distributions > E39L9984GTHYNX

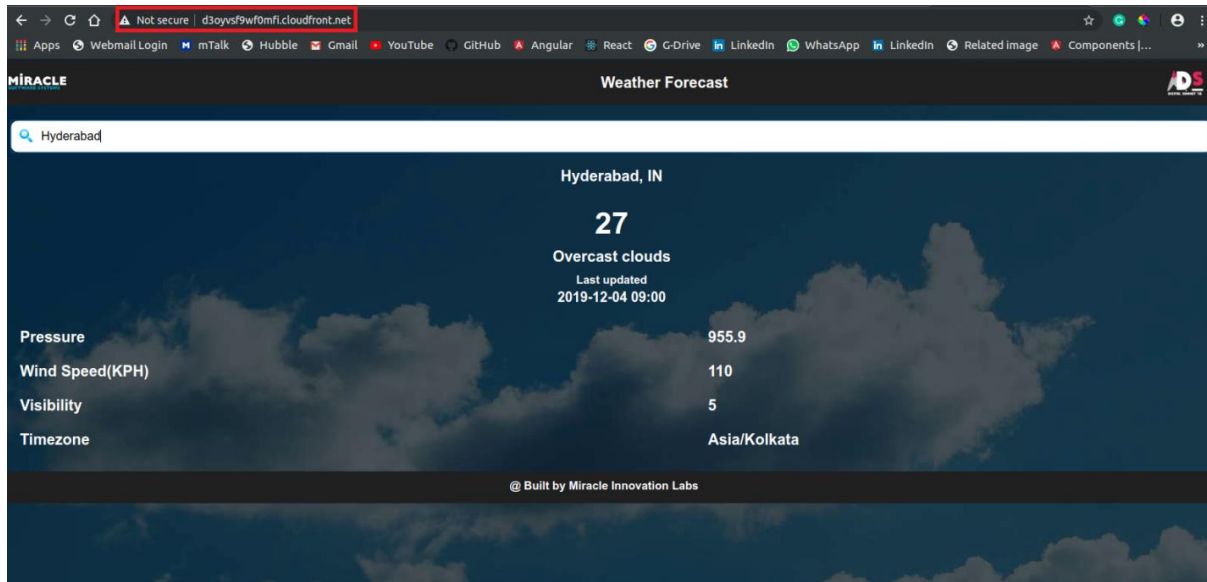
General Origins and Origin Groups Behaviors **Error Pages** Restrictions Invalidations Tags

You can configure CloudFront to respond to requests using a custom error page when your origin returns an HTTP 4xx or 5xx status code. For example, when your custom origin is unavailable and returning 5xx responses, CloudFront can return a static error page that is hosted on Amazon S3. You can also specify a minimum TTL to control how long CloudFront caches errors. For more information, see [Customizing Error Responses](#) in the *Amazon CloudFront Developer Guide*.

Create Custom Error Response Edit Delete

HTTP Error Code	Error Caching Minimum TTL	Response Page Path	HTTP Response Code
No Data			

Hit the CloudFront Endpoint and you can find your application.



For any questions regarding the lab please feel free to reach out to innovation@miraclesoft.com. We hope you enjoyed the session with us!