



Deploying a Web Application to AWS S3 and CloudFront

Open Lab | Digital Summit '18

Miracle Innovation Labs

Miracle Software Systems, Inc.

Deploying a Web Application to AWS S3 and CloudFront

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

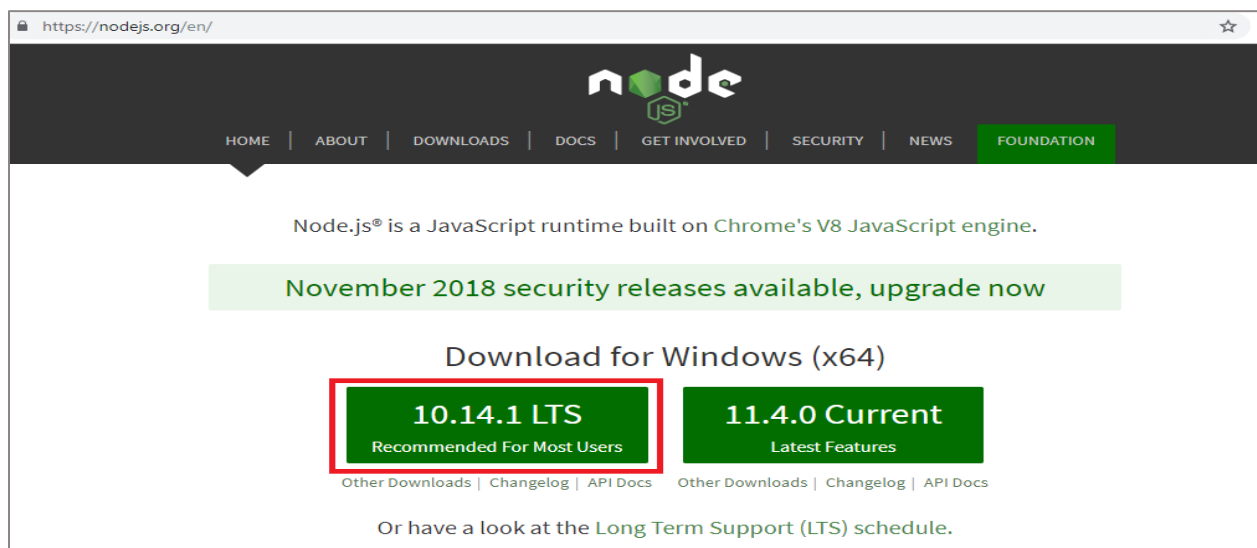
So, let us get started!

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.

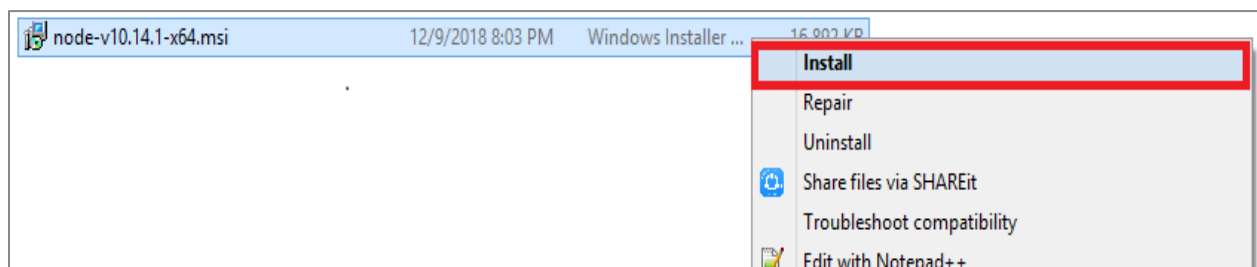
Step #1 | Installation of Node JS

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

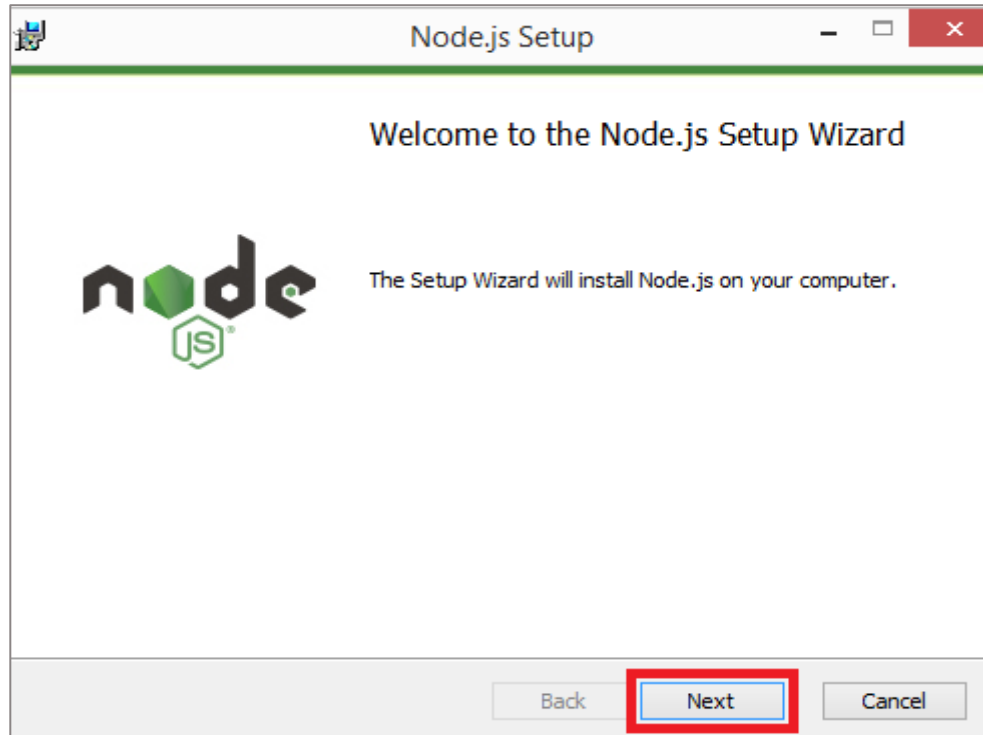
Click on **10.14.1 LTS** to download **Node JS** msi file.



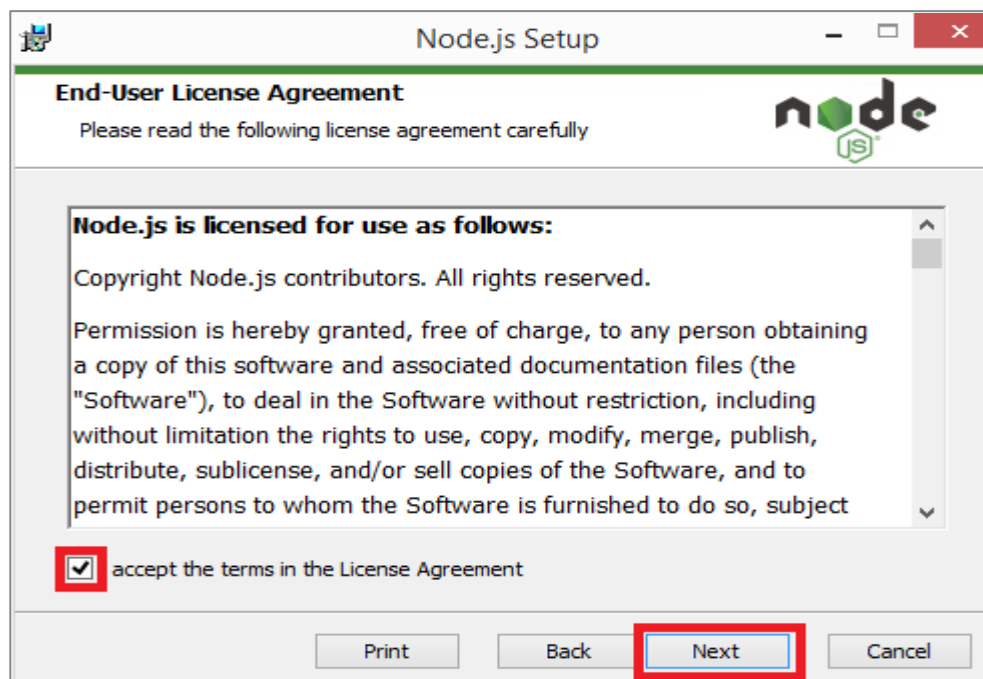
Right click on the installer and click on **Install**.



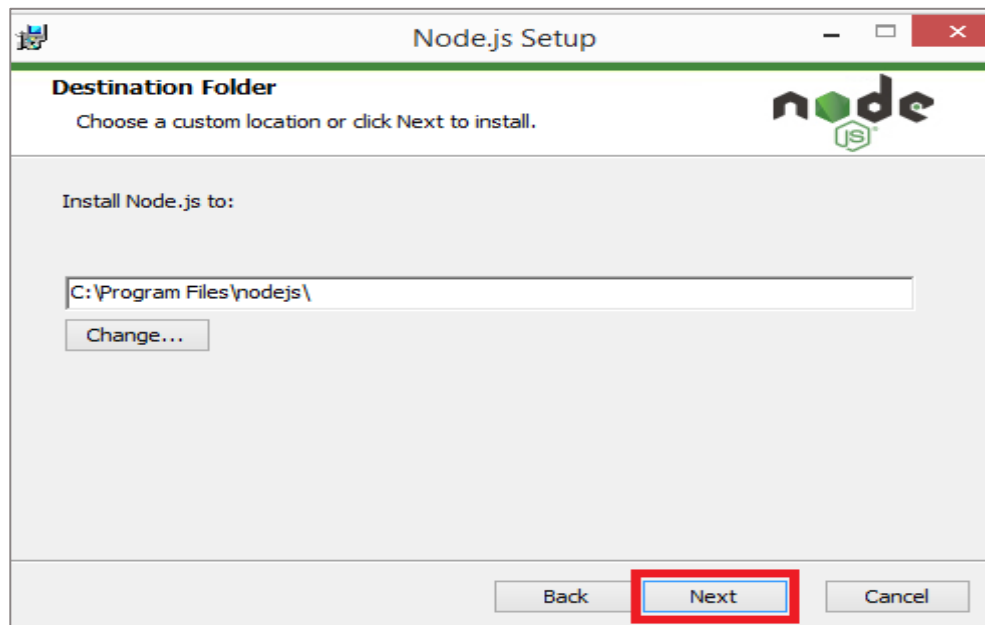
After selecting Install option the setup page is displayed, click on **Next**.



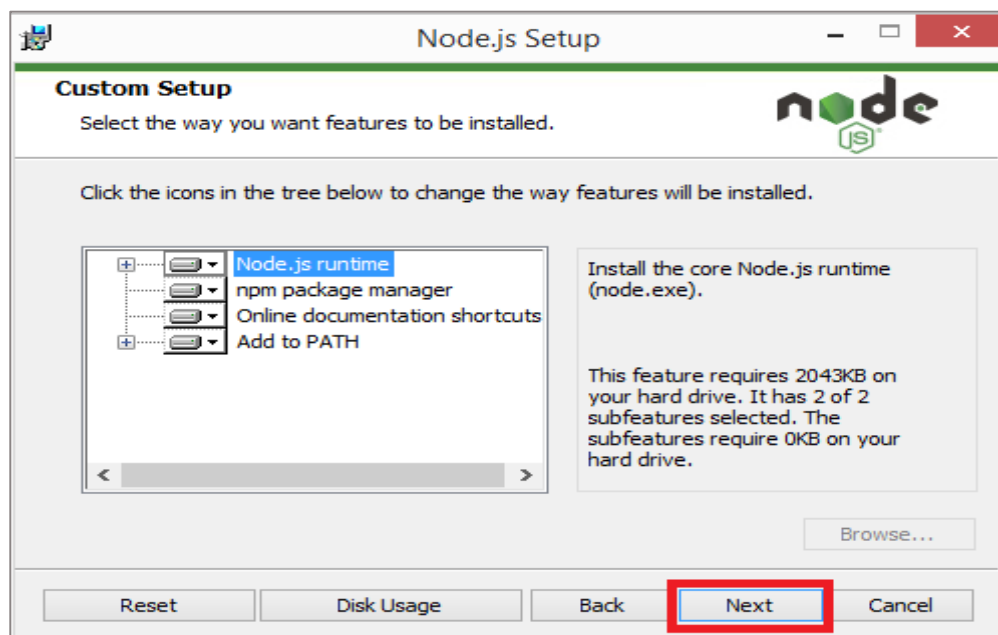
Accept the terms and click on **Next**.



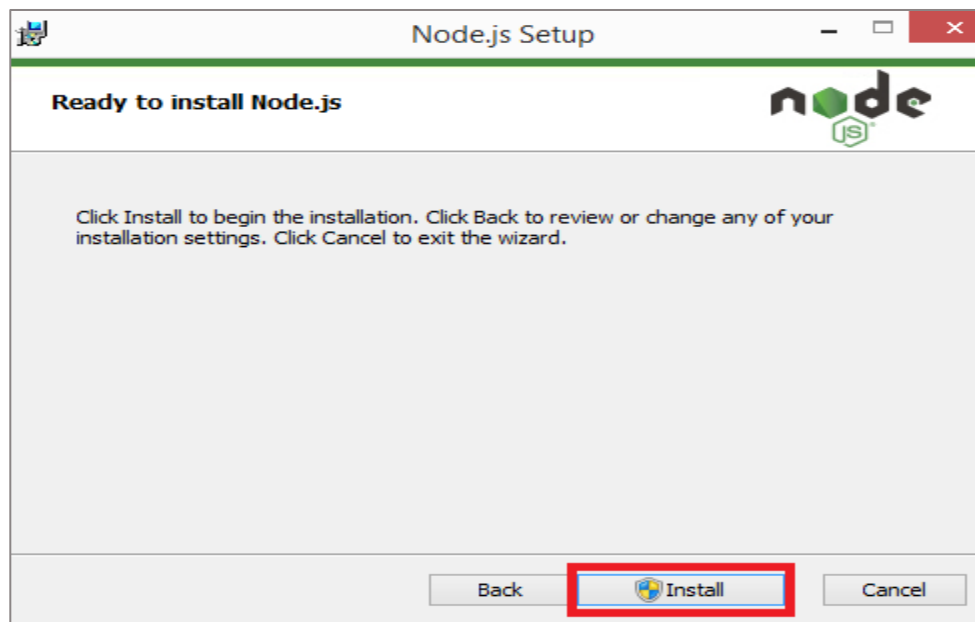
If you want to change the installation path, you can change your desired location. For now we are not changing any path, click on **Next**.



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**

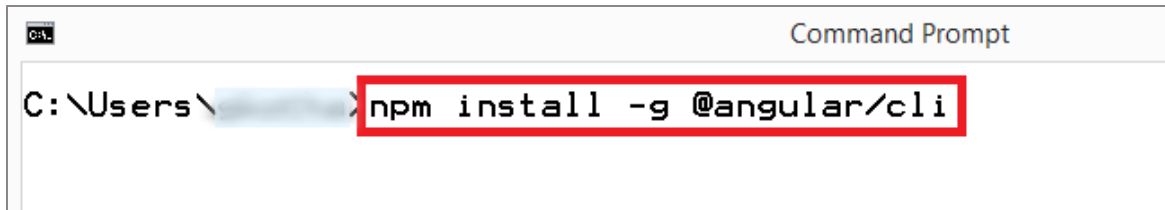
```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\...> node -v
v8.11.2

C:\Users\...> npm -v
5.6.0
```

Step #2 | Installation of Angular CLI

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

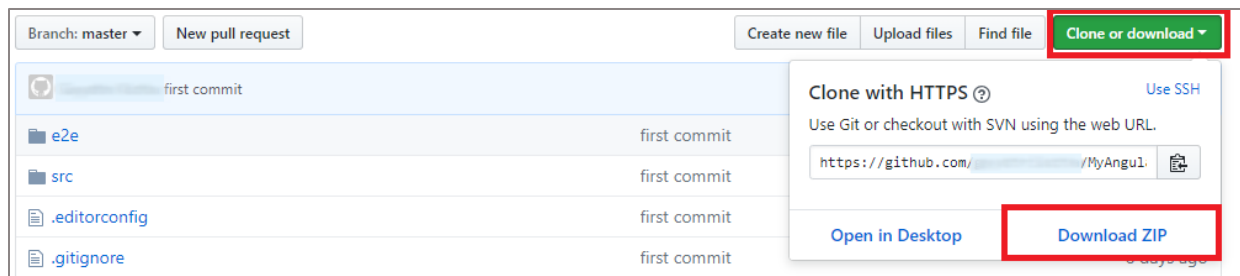


```
C:\Users\ [redacted] > npm install -g @angular/cli
```

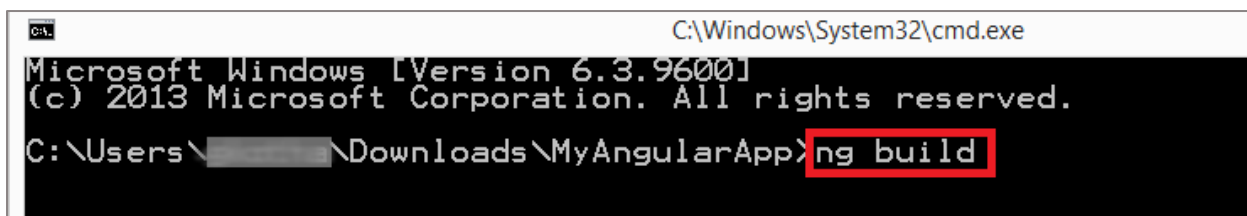
To check Angular CLI version : **ng -v**

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 command **ng build** to 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\ [redacted] \Downloads\MyAngularApp> ng build
```

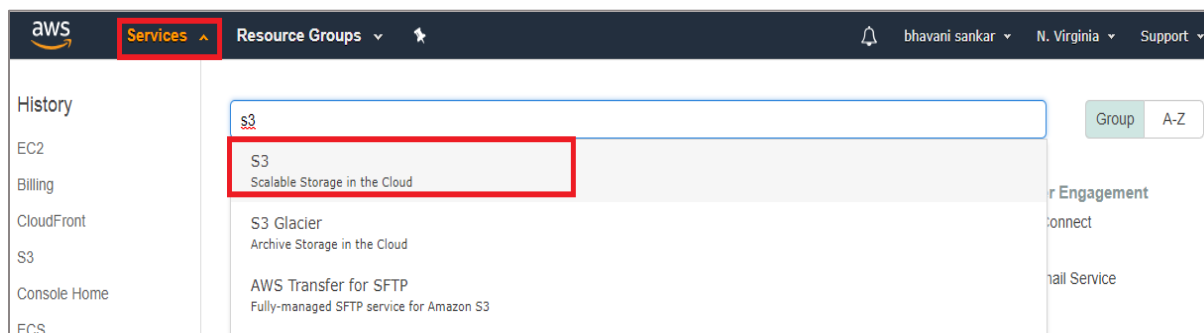
After compiling, **dist** folder will be added as shown below.

.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

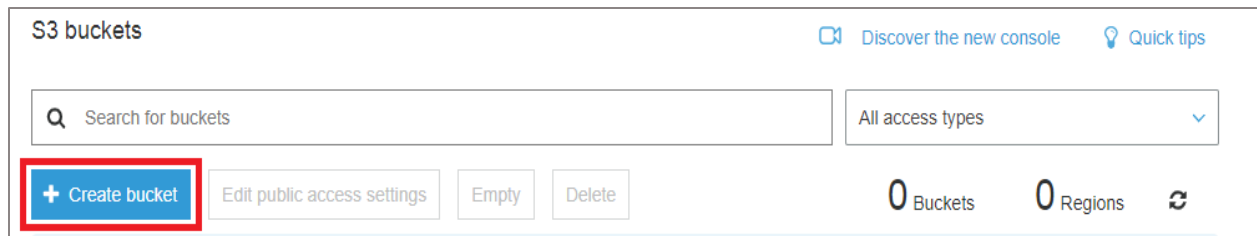
After completion of Node JS, Angular CLI installation and compiling the source code, it's time to create S3 bucket where you need to upload the dist folder files.

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.



S3 buckets

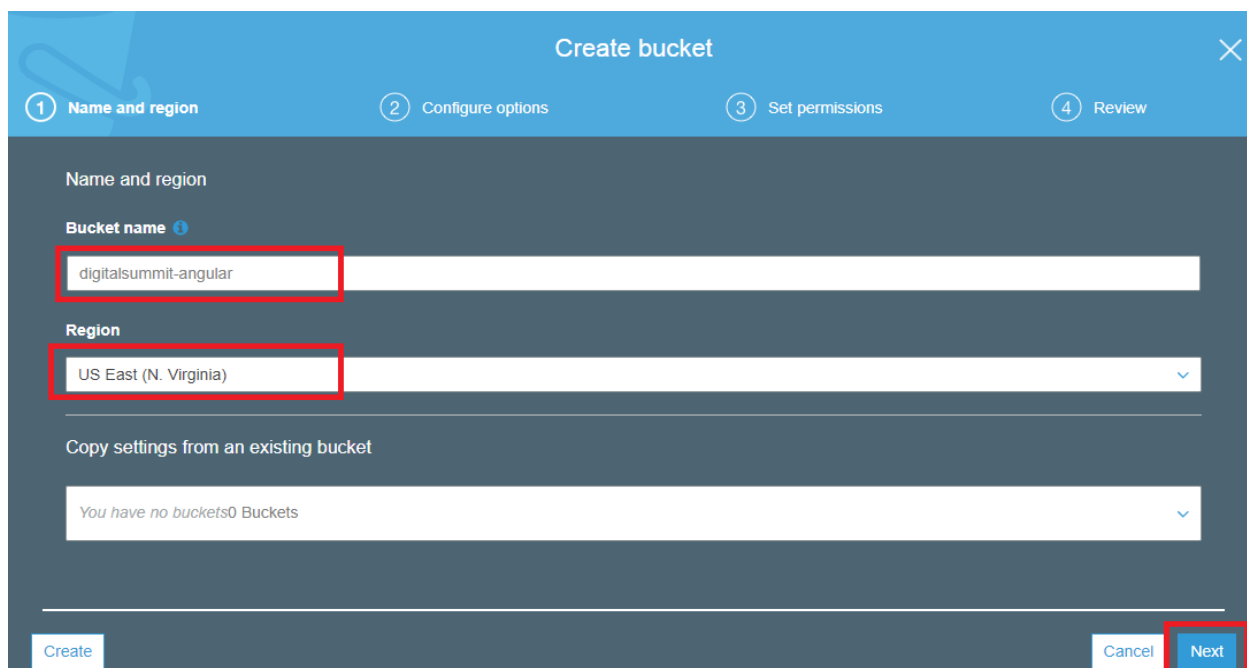
Discover the new console Quick tips

Search for buckets All access types

+ Create bucket Edit public access settings Empty Delete

0 Buckets 0 Regions

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



Create bucket

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

Name and region

Bucket name **digitalsummit-angular**

Region **US East (N. Virginia)**

Copy settings from an existing bucket

You have no buckets 0 Buckets

Create Cancel **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), 'Manage public access control lists (ACLs) for this bucket' (Block new public ACLs and uploading public objects, Remove public access granted through public ACLs), 'Manage public bucket policies for this bucket' (Block new public bucket policies, Block public and cross-account access if bucket has public policies), 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

✓ Name and region

✓ Configure options

✓ Set permissions

4 Review

Bucket name

digitalsummit-angular

Region

US East (N. Virginia)

Options

Versioning

Disabled

Server access logging

Disabled

Tagging

0 Tags

Object-level logging

Disabled

Default encryption

None

CloudWatch request metrics

Disabled

Object lock

Disabled

Permissions

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 new console

Quick tips

Q digital

All access types

+ Create bucket

Edit public access settings

Empty

Delete

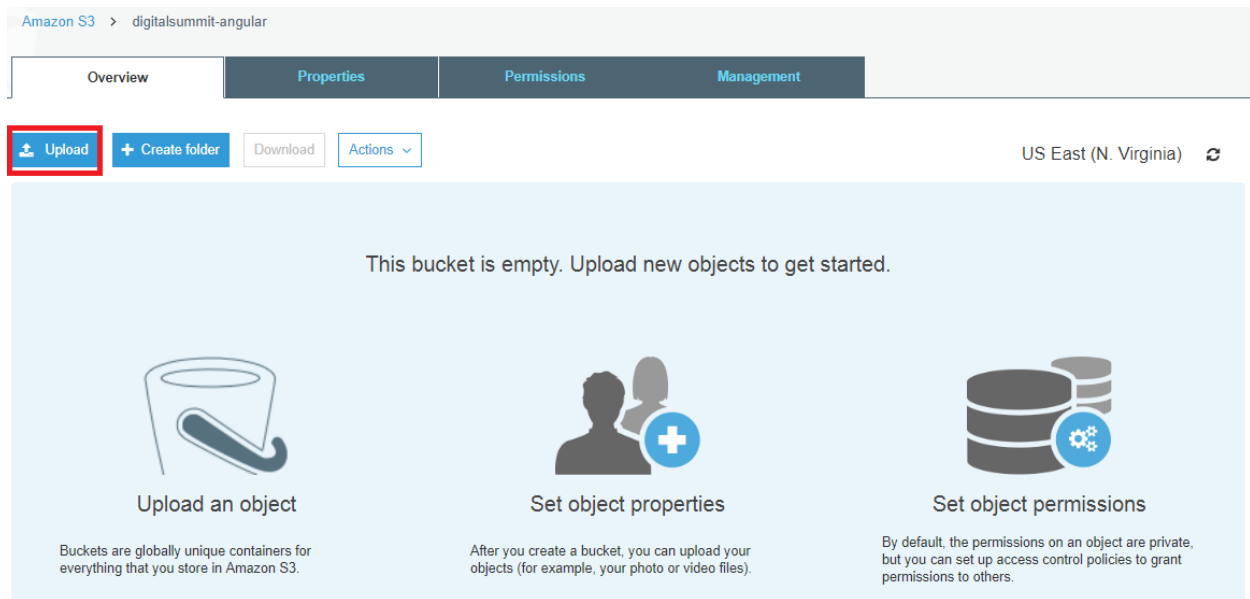
1 Buckets

1 Regions

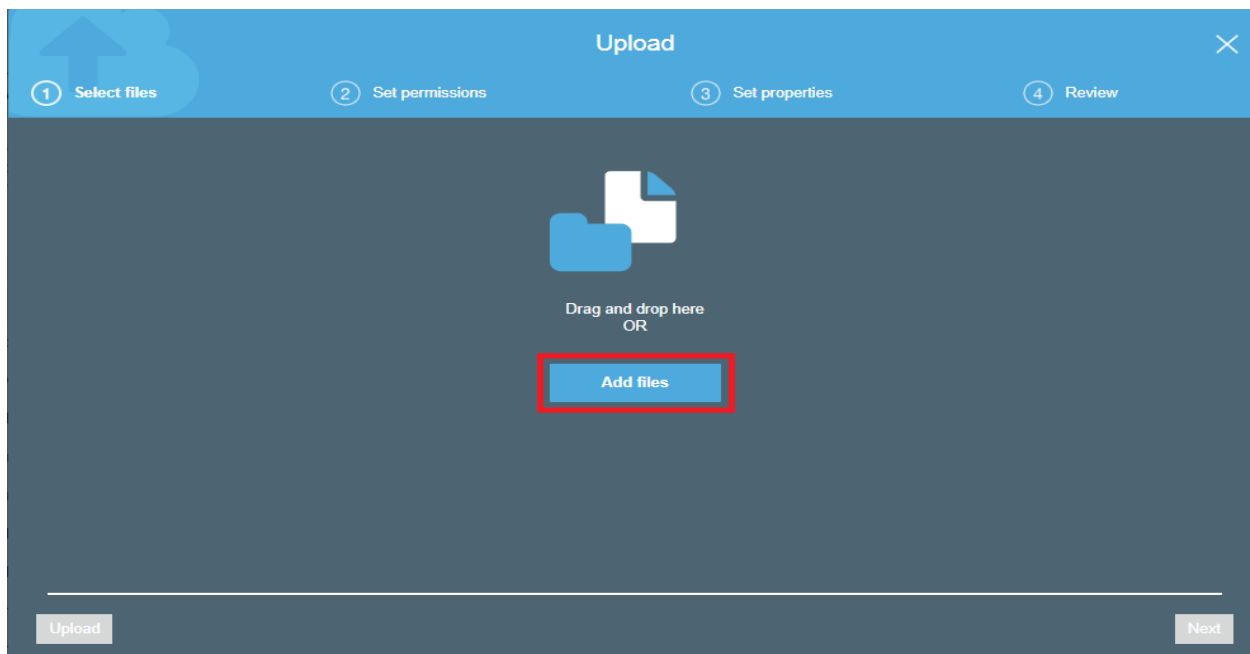
<input type="checkbox"/>	Bucket name	Access	Region	Date created
<input type="checkbox"/>	digitalsummit-angular	Bucket and objects not public	US East (N. Virginia)	Dec 6, 2018 8:42:42 PM GMT+0530

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.

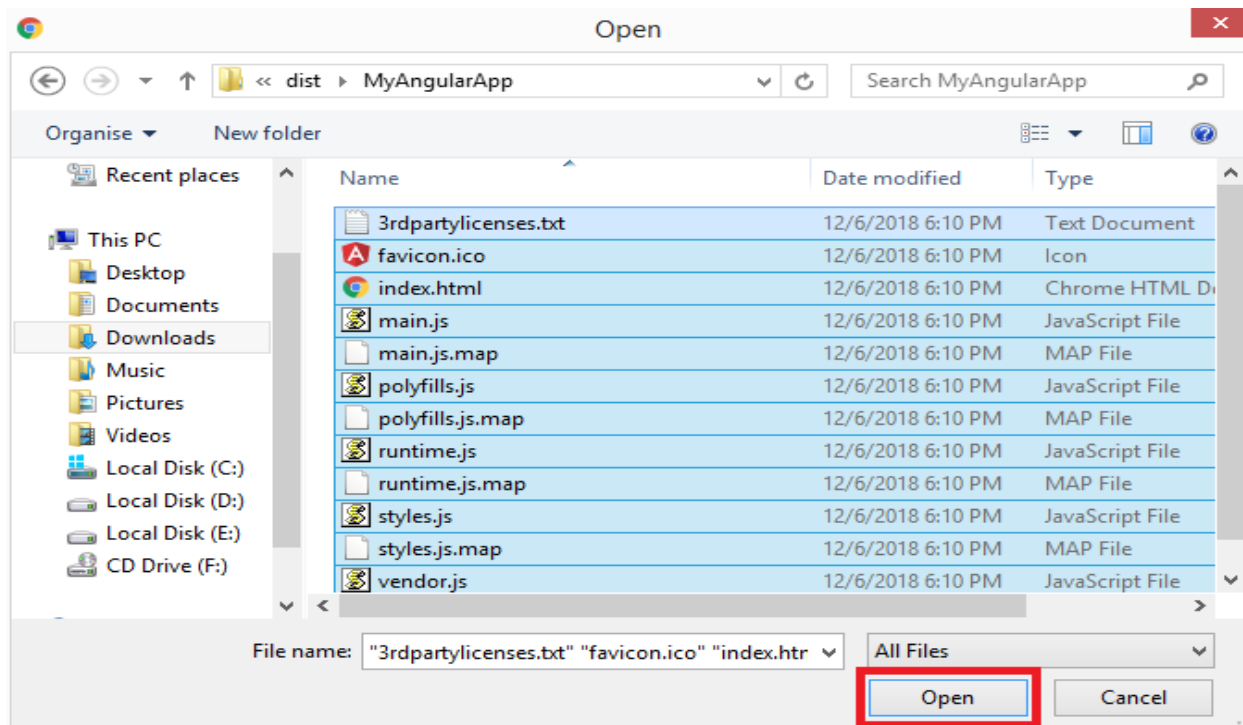


Click on **Add files** as below.

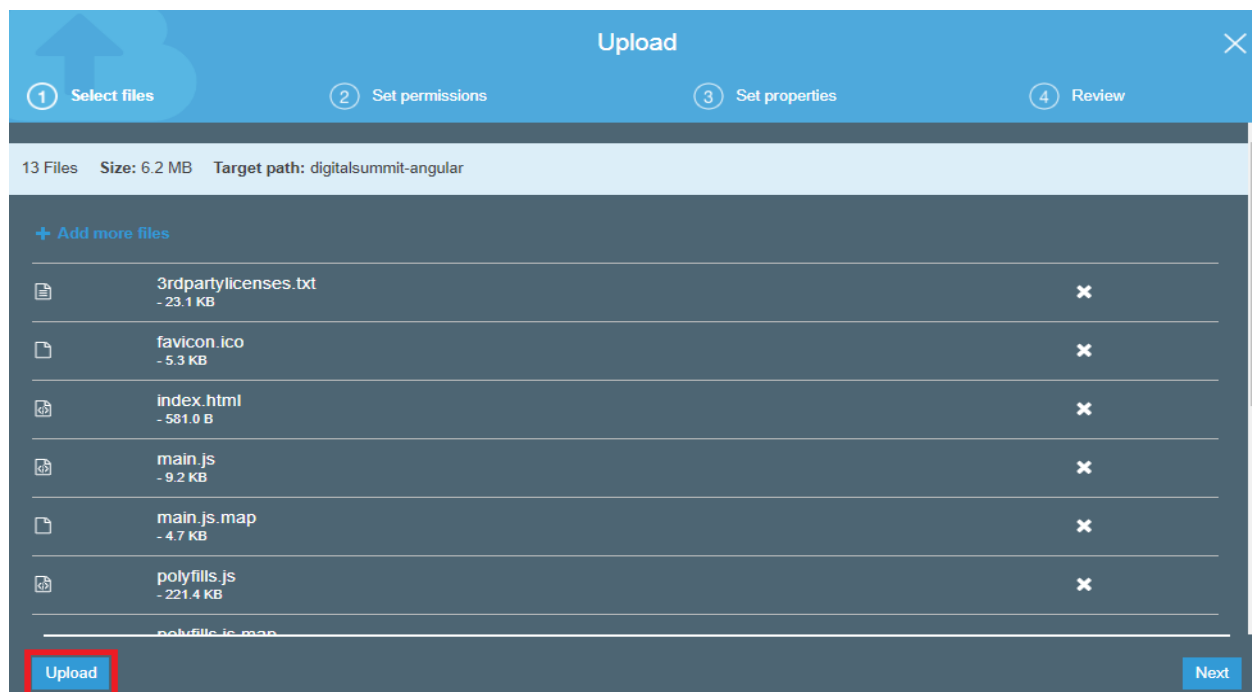


Select all files from your **dist** folder and click on **Open**.

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



Click on **Upload**.



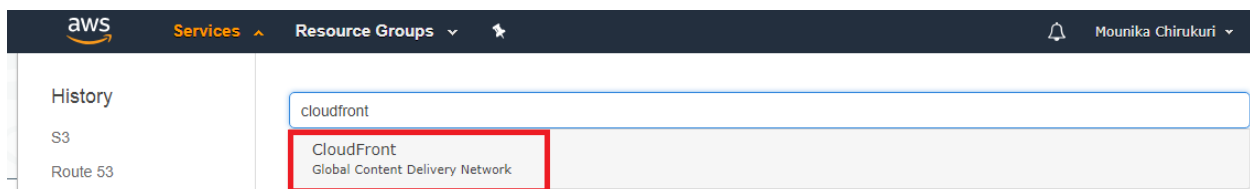
Now the build files got uploaded as below.

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) Refresh				
Viewing 1 to 13				
<input type="checkbox"/> Name	Last modified	Size	Storage class	
<input type="checkbox"/> 3rdpartylicenses.txt	Dec 7, 2018 4:39:14 PM GMT+0530	23.1 KB	Standard	
<input type="checkbox"/> favicon.ico	Dec 7, 2018 4:39:16 PM GMT+0530	5.3 KB	Standard	
<input type="checkbox"/> index.html	Dec 7, 2018 4:39:17 PM GMT+0530	581.0 B	Standard	
<input type="checkbox"/> main.js	Dec 7, 2018 4:39:18 PM GMT+0530	9.2 KB	Standard	
<input type="checkbox"/> main.js.map	Dec 7, 2018 4:39:19 PM GMT+0530	4.7 KB	Standard	
<input type="checkbox"/> polyfills.js	Dec 7, 2018 4:39:21 PM GMT+0530	221.4 KB	Standard	
<input type="checkbox"/> polyfills.js.map	Dec 7, 2018 4:39:22 PM GMT+0530	219.8 KB	Standard	
<input type="checkbox"/> runtime.js	Dec 7, 2018 4:39:24 PM GMT+0530	5.1 KB	Standard	

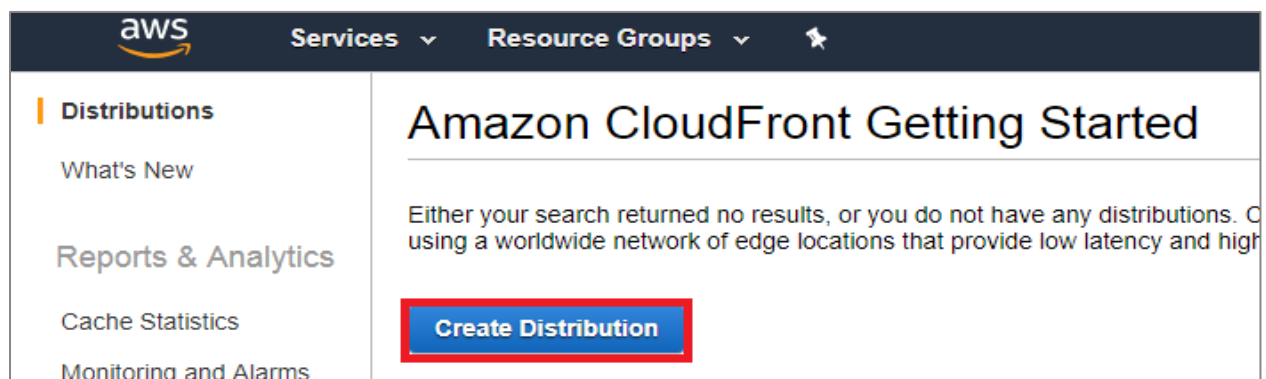
Once you are good with configuring the S3, it's time to configure CDN.

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.

Create Distribution

Origin Settings

Origin Domain Name

digitalsummit-angular.s3.amazonaws.com

Origin Path

Origin ID

S3-digitalsummit-angular

Restrict Bucket Access

☒ Yes

☐ No

Origin Access Identity

☒ Create a New Identity

☐ Use an Existing Identity

Comment

access-identity-digitalsummit-angular.s3.

Grant Read Permissions on Bucket

☒ Yes, Update Bucket Policy

☐ No, I Will Update Permissions

Origin Custom Headers

Header Name	Value

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

☒ HTTP and HTTPS

☐ Redirect HTTP to HTTPS

☐ HTTPS Only

Allowed HTTP Methods

☒ GET, HEAD

☐ GET, HEAD, OPTIONS

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

Field-level Encryption Config

Cached HTTP Methods

GET, HEAD (Cached by default)

Cache Based on Selected Request Headers

None (Improves Caching)

[Learn More](#)

Object Caching

☒ Use Origin Cache Headers

☐ Customize

[Learn More](#)

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 ⓘ

AWS WAF Web ACL ⓘ

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://d1111111abcdef8.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.

ⓘ

[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 ⓘ

Logging ☐ On ⓘ
☒ Off

Bucket for Logs ⓘ

Log Prefix ⓘ

Cookie Logging ☐ On ⓘ
☒ Off

Enable IPv6 ☒ ⓘ
[Learn more](#)

Comment ⓘ

Distribution State ☒ Enabled ⓘ
☐ Disabled

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.

CloudFront Distributions

Create Distribution		Distribution Settings	Delete	Enable	Disable			Refresh	Settings	Help
Viewing : Any Delivery Method		Any State								
	Delivery Me	ID	Domain Name	Co	Origin	CNAMEs	Status	State		
<input type="checkbox"/>	Web	EYF6UEGOEHVQ3	dvk7r38d9ck9i.cloudfront.net	-	digitalsummit-angular.s3.ama	-	In Progress	Enabled		

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

CloudFront Distributions

Create Distribution

Distribution Settings

Delete

Enable

Disable

Viewing :


Any Delivery Method

Any State

<<

<

View

	Delivery Me	ID	Domain Name	Co	Origin	CNAMEs	Status	State
<input checked="" type="checkbox"/>	 Web	EYF6UEGOEHVQ3	dvk7r38d9ck9i.cloudfront.net	-	digitalsummit-angular.s3.ama	-	Deployed	Enabled

Now you can access your website using domain name as shown below.

CloudFront Distributions

Create Distribution

Distribution Settings

Delete

Enable

Disable

↺

⚙

?

👤

Viewing :

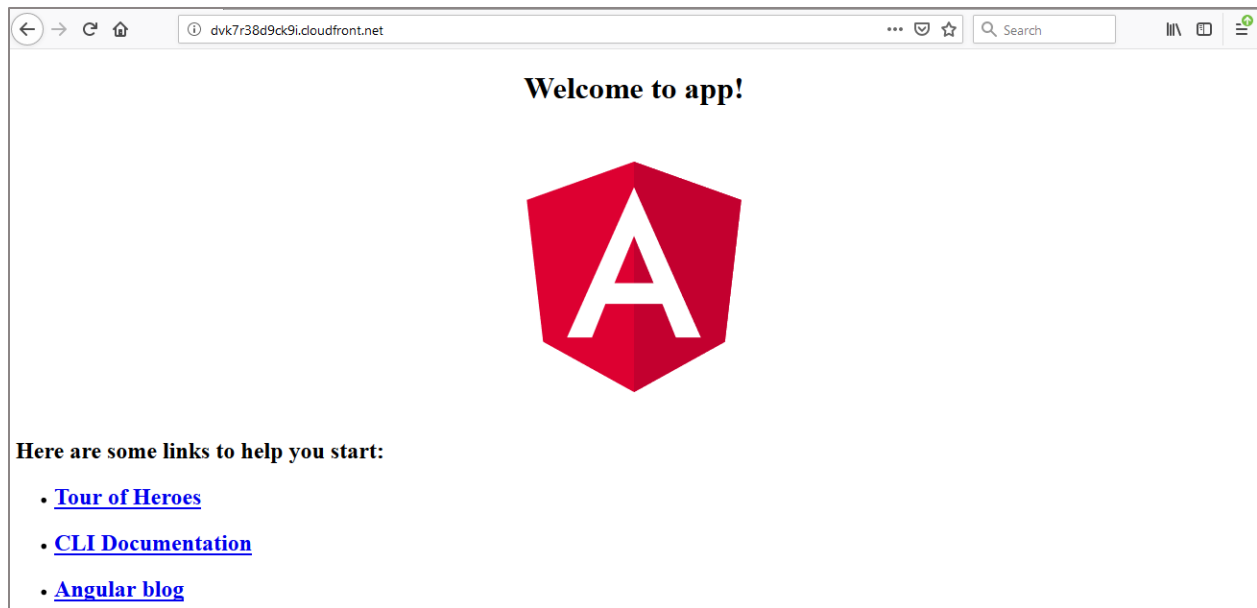
Any Delivery Method

Any State

⏪ < View

	Delivery Me	ID	Domain Name	Co	Origin	CNAMEs	Status	State
<input checked="" type="checkbox"/>	Web	EYF6UEGOEHVQ3	dvk7r38d9ck9i.cloudfront.net	-	digitalsummit-angular.s3.ama	-	Deployed	Enabled

Enter domain name in browser and hit enter. Here is the UI after deployment of sample application.



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