

DOCUMENTATION ON HOSTING A STATIC WEBSITE ON AMAZON S3 AND CLOUDFRONT



Overview of Amazon S3

Amazon S3, also known as Simple Storage Service, offers scalable object storage within the Amazon Web Services (AWS) platform. It provides developers and businesses with a robust infrastructure for data objects, ensuring security, durability, and high availability. Essentially, S3 functions as a reliable, secure, and cost-effective solution for cloud-based data storage and management. This allows organizations to dedicate resources to innovative application development without the burden of maintaining storage infrastructure.

Leveraging Amazon S3 and CloudFront for static website delivery.

By using S3 for storage and CloudFront for delivery, you get the best of both worlds: S3's ability to handle large amounts of data reliably and CloudFront's network for fast website loading around the globe. Here's a brief overview of the process:

- **Amazon S3:** S3 functions as a cost-optimized, highly dependable storage solution for static website assets. This includes HTML files, CSS, JavaScript, images, and additional resources. Users can leverage S3 buckets specifically configured for static website hosting. This enables direct web page serving from the bucket itself.
- **Cloudfront:** CloudFront acts as a content delivery name (CDN), a geographically distributed network that optimizes web content delivery globally. It achieves this by caching static website assets on edge servers positioned near users. This proximity significantly reduces latency (data transfer delay) and enhances website performance. Additionally, CloudFront allows users to establish a distribution and configure it to utilize an S3 bucket as the source of the content.
- **Integration:** By combining S3's object storage with CloudFront's CDN, users gain a powerful combination. S3 offers the durability and scalability needed for website content, while CloudFront's edge caching network delivers it globally at high speeds. This boosts website reliability, scalability, and performance, ensuring a smooth experience for all visitors.

Purpose of the documentation

By following the documentation, users will be able to:

- **Create an S3 Bucket:** Understand how to create an S3 bucket using the AWS Management Console or AWS CLI, including considerations for naming conventions, region selection, and bucket settings.
- **Upload an HTML Template and Assets:** Learn how to upload their HTML template and any associated assets (CSS, JavaScript, images, etc.) to the S3 bucket, ensuring proper organization and file structure.
- **Enable Static Website Hosting:** Configure the S3 bucket for static website hosting, specifying the default document and error document as per their requirements.
- **Check website Reachability:** Verify the accessibility of the static website by accessing the S3 website endpoint URL provided by AWS.
- **Set up Cloudfront Distribution:** Create a CloudFront distribution to improve website performance and reliability through content delivery network (CDN) caching and distribution. Understand the configuration options, including origin settings, cache behavior, and distribution settings.
- **Associate Cloudfront with S3 Bucket:** Configure CloudFront to serve the static website content from the S3 bucket origin, ensuring proper routing and caching behavior.
- **Test Website Delivery via CloudFront:** Confirm that the static website is being served correctly through the CloudFront distribution by accessing the CloudFront domain name.

Basic Requirements

An AWS account with appropriate permissions.

Before proceeding with the setup of serving a static website with Amazon S3 and CloudFront, ensure that you have the following prerequisites in place:

- **AWS Account:** You must have an AWS account to access and utilize AWS services. If you don't have an account yet, you can sign up for an AWS account on the AWS website.
- **Permissions:** Ensure that your AWS account has appropriate permissions to create and configure S3 buckets, CloudFront distributions, and associated resources. Depending on your organization's policies, you may need administrative or specific IAM (Identity and Access Management) permissions to perform these tasks.
- **Administrator Access:** If you have administrator access to the AWS account, you should have sufficient permissions to create and manage S3 buckets and CloudFront distributions. However, exercise caution when granting administrator privileges, as they provide broad access to AWS resources.
- **IAM Policies:** If you don't have administrator access, ensure that you have been granted IAM permissions that allow you to create and configure S3 buckets and CloudFront

distributions. Your IAM policies should include permissions such as S3:CreateBucket, S3:PutBucketPolicy, cloudfront: Create Distribution, and other necessary actions.

- **Access Keys:** To interact with AWS programmatically using the AWS CLI or SDKs, you'll need to generate access keys for your IAM user account. These access keys consist of an Access Key ID and a Secret Access Key, which are used to authenticate API requests to AWS services.
- **Access Key ID:** A unique identifier for your IAM user or role.
- **Secret Access Key:** A secret key used to authenticate requests made with your Access Key ID.

Ensure that you securely manage your AWS access keys and never share them publicly or expose them in your source code.

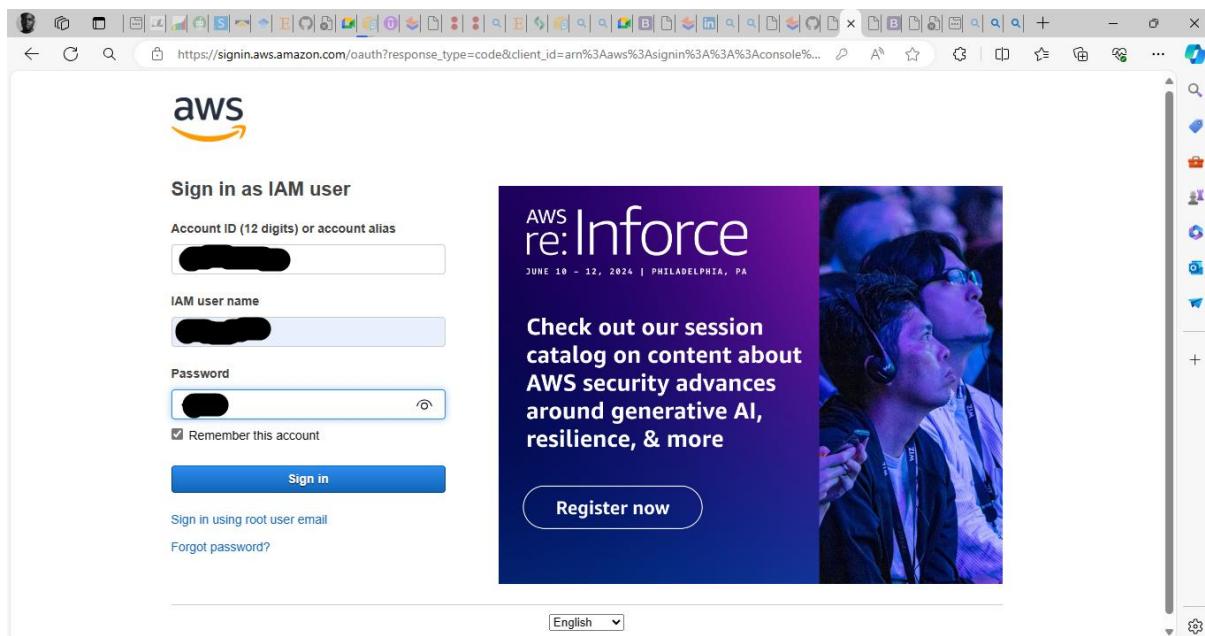
- **Billing Information:** Provide valid billing information for your AWS account to cover any charges incurred by using AWS services. You'll only be charged for the resources you use, and AWS offers a free tier with certain usage limits for new accounts

Setting up the Amazon S3 Bucket

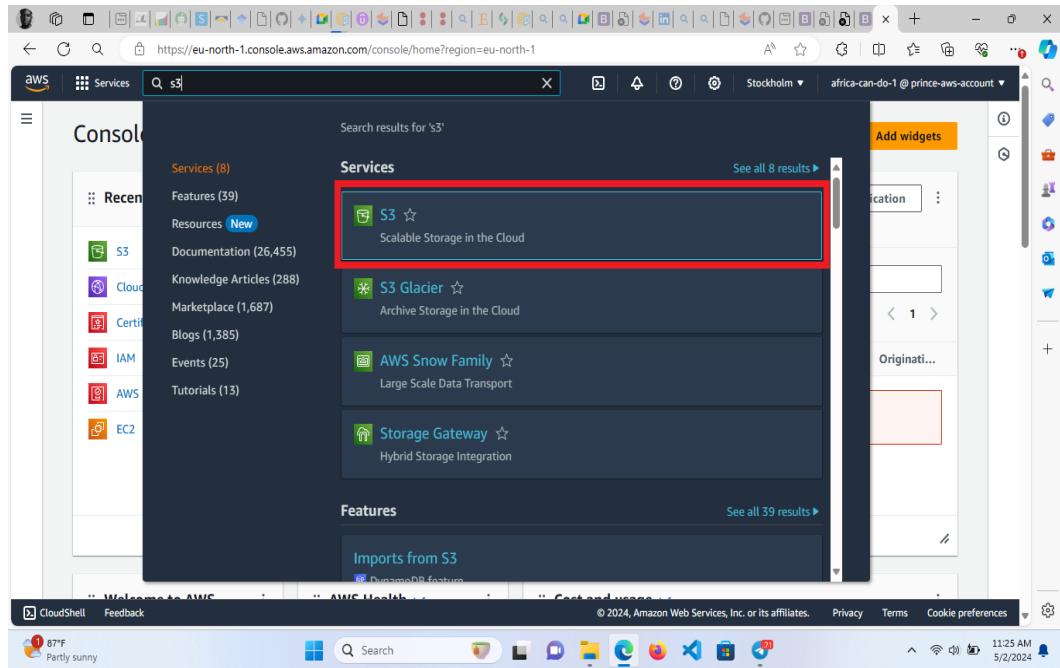
How to create s3 bucket on AWS.

Here's a step-by-step guide:

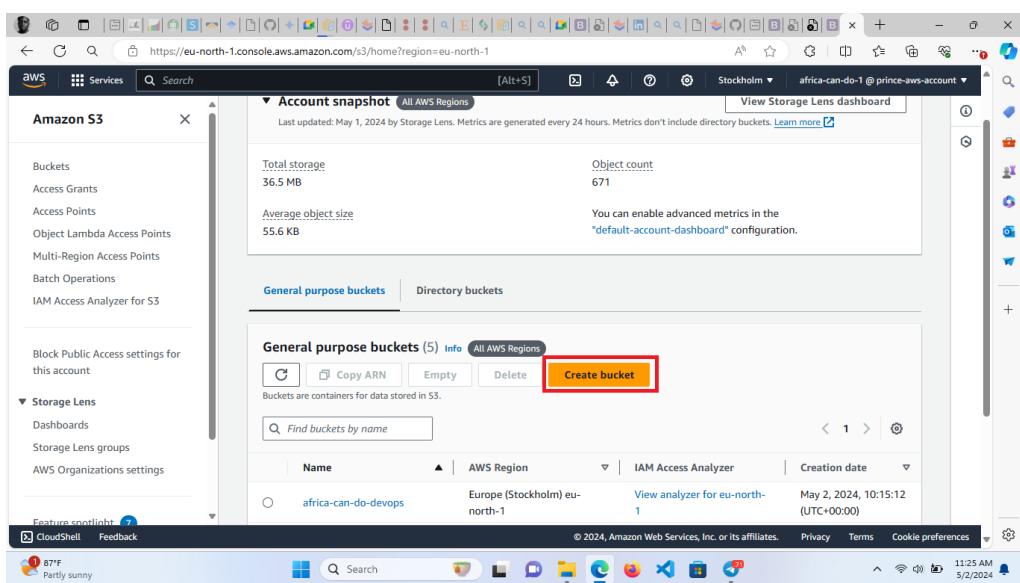
- Sign into the AWS Management Console: Go to the AWS Management Console (<https://aws.amazon.com/console/>) and sign in to your AWS account



- **Navigate to S3:** Once logged in, navigate to the S3 service by either typing "S3" in the search bar at the top or by selecting it from the list of services.

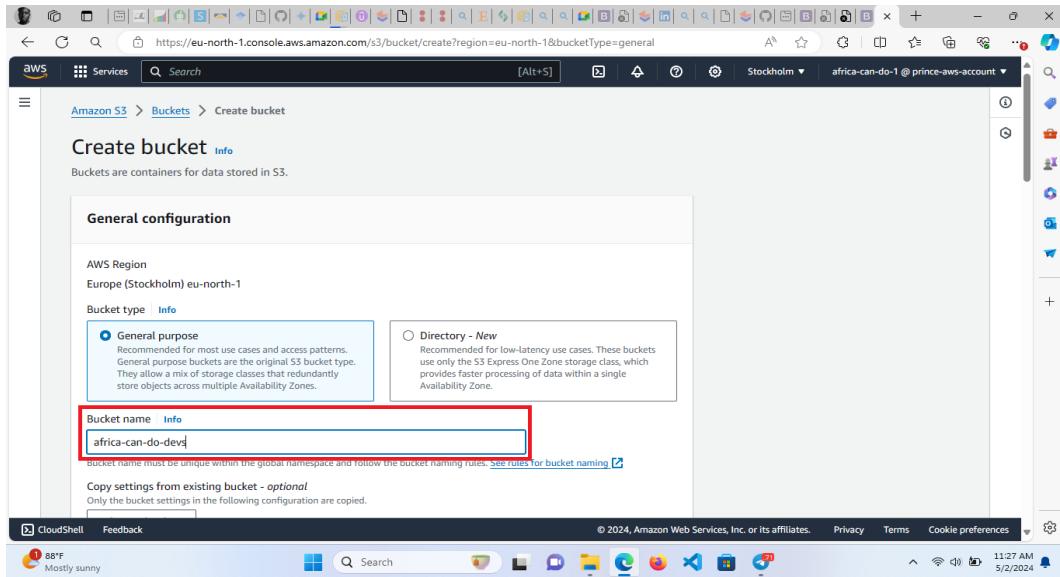


Create Bucket: Click on the "Create bucket" button.



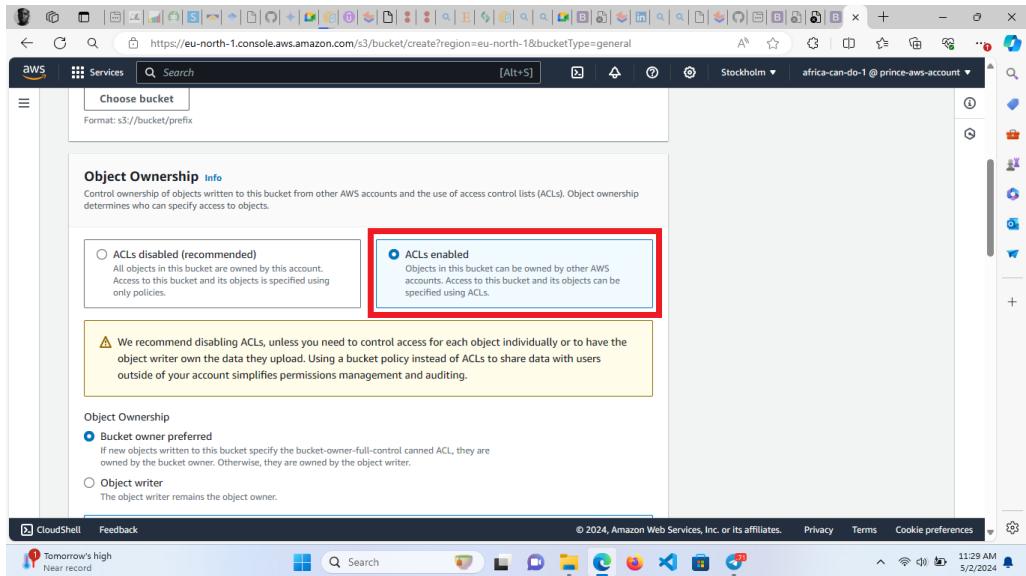
- **Bucket Configuration:**

- **Bucket Name:** Enter a unique name for your bucket. This name must be globally unique across all existing bucket names in AWS because the bucket name becomes part of the bucket URL (e.g., <https://s3.amazonaws.com/bucket-name>).

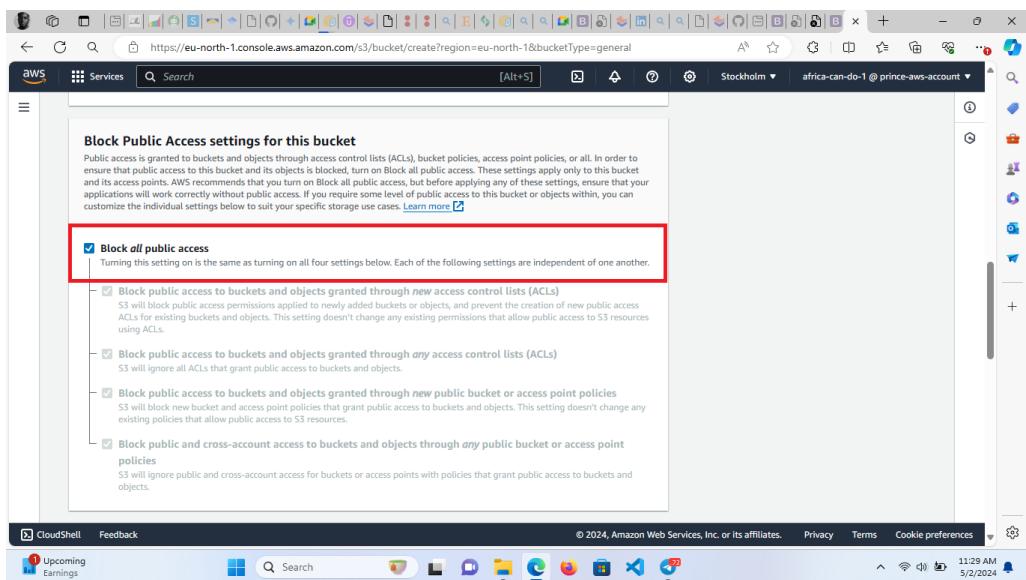


- **Region:** Choose the AWS region where you want to create your bucket. Select the region that's closest to your users to minimize latency.
- **Configure options:** You can configure additional options like versioning, logging, tags, encryption, etc., based on your requirements. These are optional.
- **Set Permissions:**

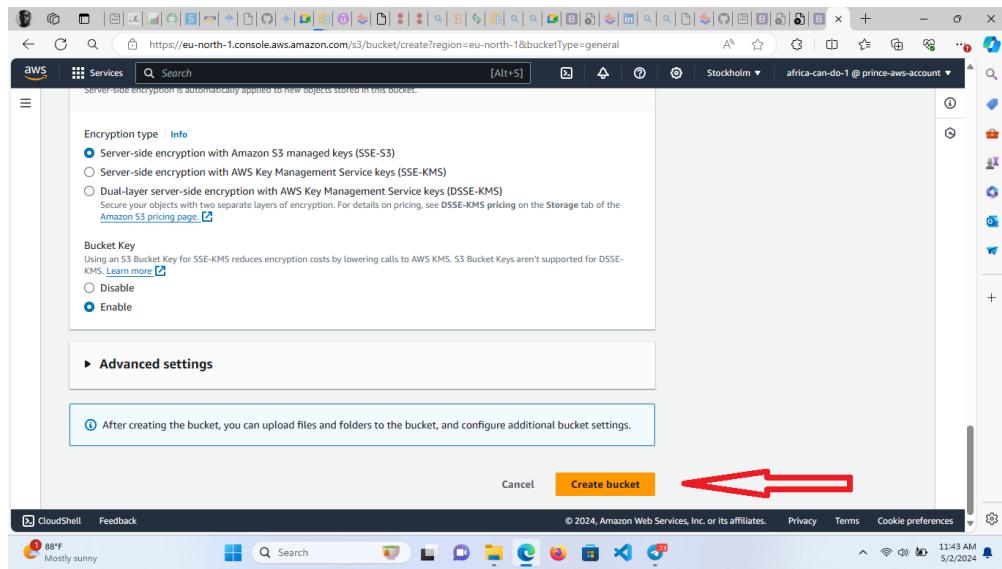
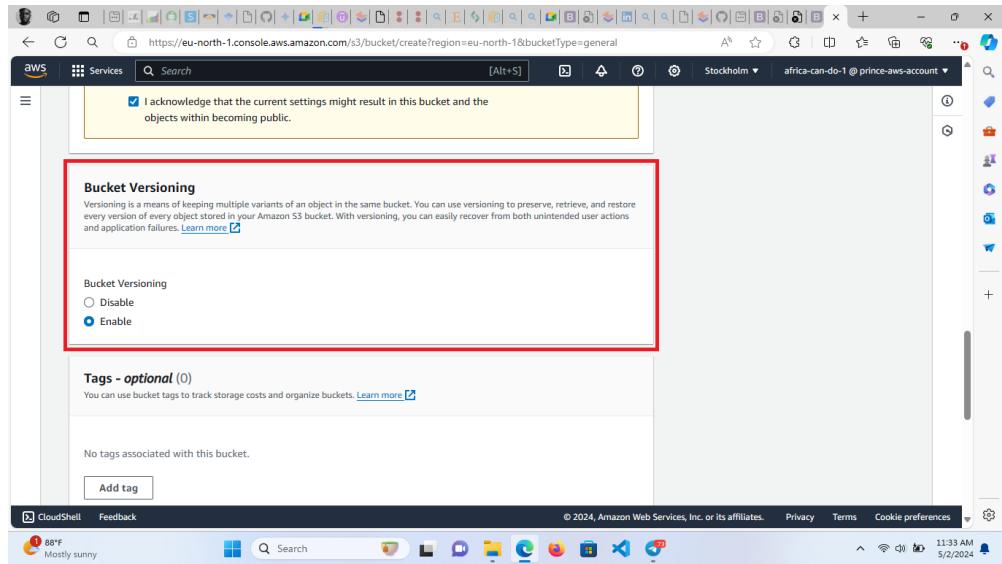
You can configure permissions for the bucket. By default, only the bucket owner has access to the bucket and its contents. You can grant additional permissions to other AWS accounts or make the bucket publicly accessible if needed.



- Uncheck this box



- **Review:** Review your settings and configurations to ensure everything is as desired.



- **Create Bucket:** Click on the "Create bucket" button to create your S3 bucket.

Once the bucket is created, you can start uploading files, configuring access policies, enabling versioning, setting up lifecycle rules, and much more based on your specific use case and requirements.

Uploading website content (HTML, CSS, JavaScript, images, etc.)

Steps to follow to upload website contents to an S3 bucket on AWS:

- **Prepare your Website Contents:**

Ensure that your website files (HTML, CSS, JavaScript, images, etc.) are organized and ready to be uploaded.

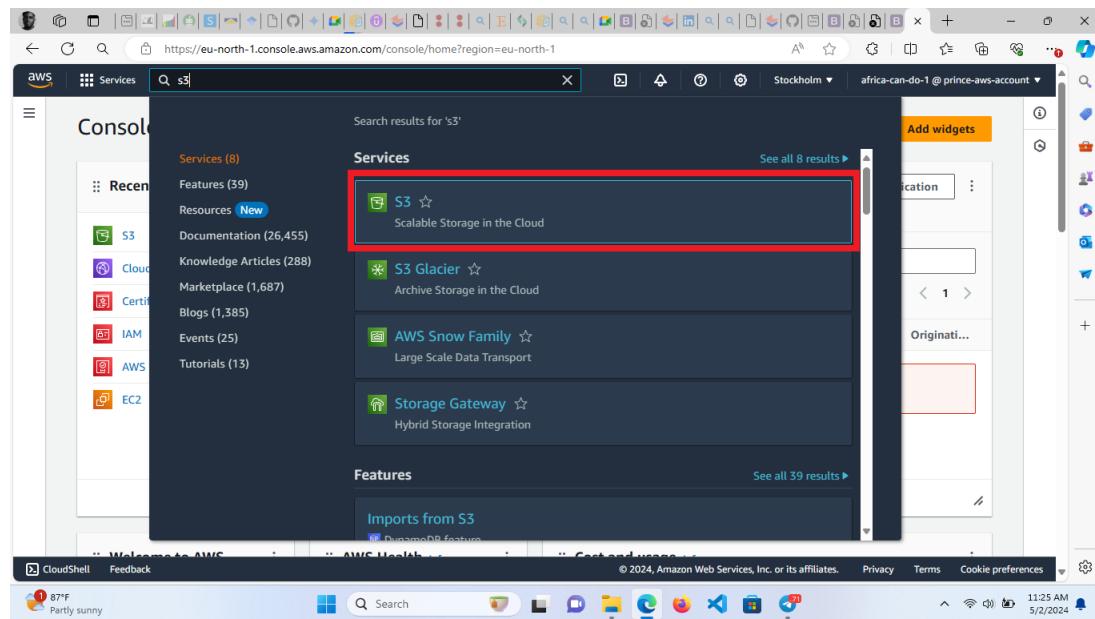
Make sure your website's main HTML file (e.g., **index.html**) is present, as this will be the default page served by S3.

- **Sign in to the AWS Management Console:**

Go to the AWS Management Console (<https://aws.amazon.com/console/>) and sign in to your AWS account.

- **Navigate to S3:**

Navigate to the S3 service by typing "S3" in the search bar at the top or by selecting it from the list of services.



- **Select your Bucket:**

Click on the name of the bucket where you want to upload your website contents.

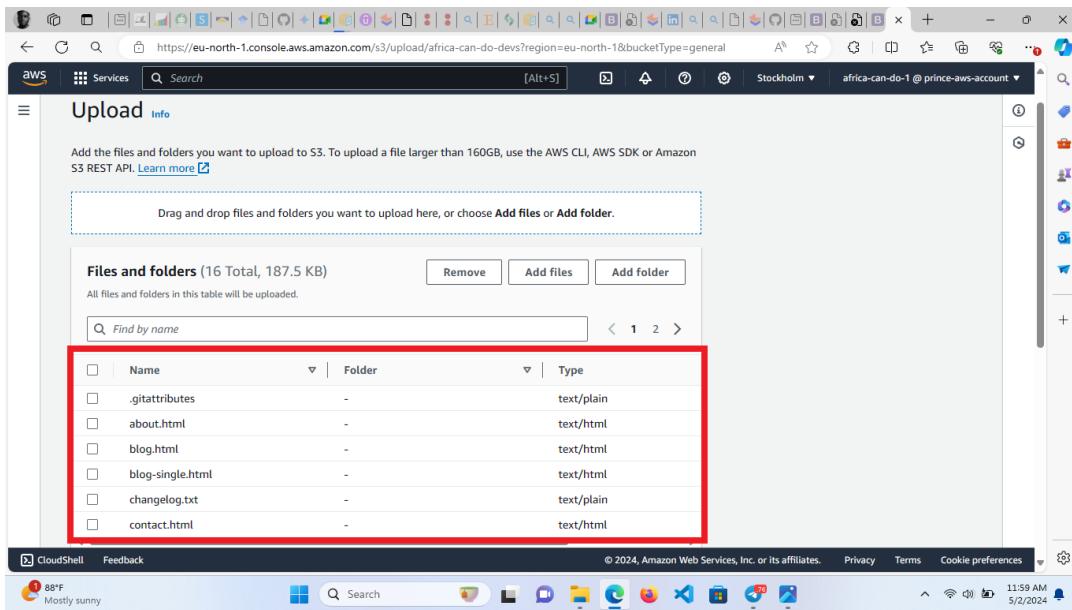
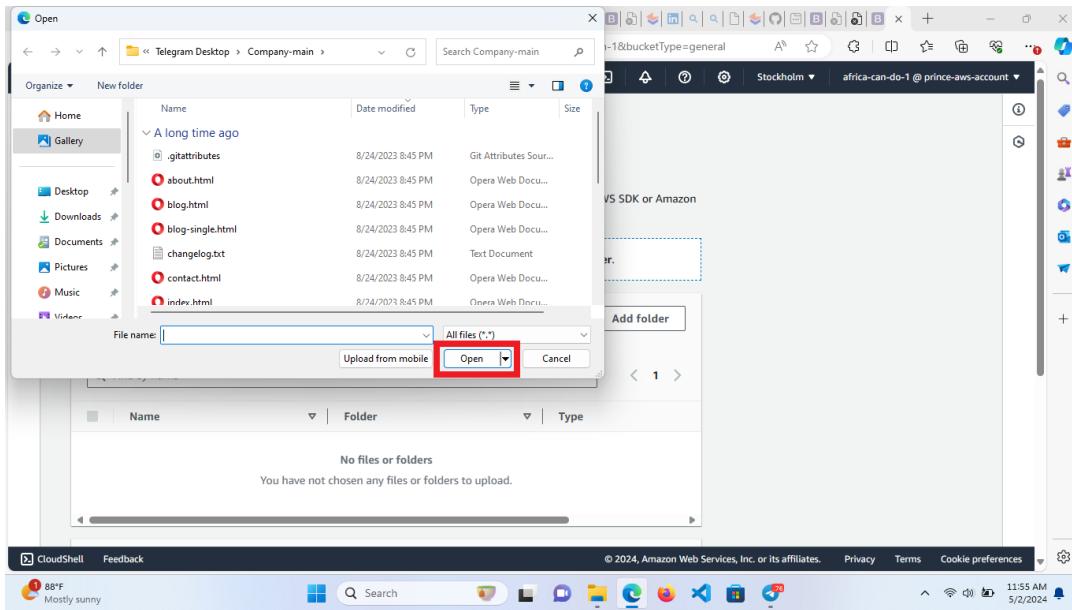
The screenshot shows the AWS S3 console with the URL <https://eu-north-1.console.aws.amazon.com/s3/buckets?region=eu-north-1&bucketType=general>. A green banner at the top says "Successfully created bucket 'africa-can-do-devs'. To upload files and folders, or to configure additional bucket settings, choose View details." Below the banner, there are tabs for "General purpose buckets" and "Directory buckets". Under "General purpose buckets", there is a table with columns: Name, AWS Region, IAM Access Analyzer, and Creation date. The table lists four buckets: "africa-can-do-devs" (selected), "africa-can-do-devops", "africa-can-do-qa", and "elasticbeanstalk-us-east-1-001526952227". The "africa-can-do-devs" row is highlighted with a red box. At the bottom of the table, there are buttons for "Create bucket", "Copy ARN", "Empty", and "Delete". The status bar at the bottom right shows "11:48 AM 5/2/2024".

- **Upload Website Files:**

Click on the "Upload" button.

The screenshot shows the AWS S3 bucket "africa-can-do-devs" with the URL <https://eu-north-1.console.aws.amazon.com/s3/buckets/africa-can-do-devs?region=eu-north-1&bucketType=general&tabc=objects>. The page title is "Amazon S3 > Buckets > africa-can-do-devs". The "Objects" tab is selected. The main area shows a table with one row: "No objects". Below the table, it says "You don't have any objects in this bucket." At the bottom of the table, there is a "Upload" button with a red arrow pointing to it. Other buttons in the row include "Actions", "Create folder", "Copy S3 URI", "Copy URL", "Download", "Open", "Delete", and "Upload". The status bar at the bottom right shows "11:49 AM 5/2/2024".

Select the files and folders from your local machine that you want to upload to the bucket.



Alternatively, you can drag and drop files directly into the browser window.

The screenshot shows the AWS S3 console interface. A modal window titled "Uploading" is displayed, indicating "Total remaining: 156 files: 18.2 MB(100.00%)". Below the modal is a table titled "Files and folders (157 Total, 18.2 MB)" showing a list of files with their names, folders, types, sizes, statuses, and errors. The status column for all files shows "Pending". At the bottom of the screen, a taskbar displays various application icons.

Name	Folder	Type	Size	Status	Error
norframewor...	assets/vend...	text/javascript	2.02 KB	Pending	-
swiper-bun...	assets/vend...	text/css	16.1 KB	Pending	-
swiper-bun...	assets/vend...	text/javascript	140.3 KB	Pending	-
swiper-bun...	assets/vend...	-	528.0 KB	Pending	-
remixicon.css	assets/vend...	text/css	107.8 KB	Pending	-
remixicon.eot	assets/vend...	-	393.8 KB	Pending	-
remixicon.less	assets/vend...	-	107.9 KB	Pending	-
remixicon.svg	assets/vend...	image/svg+	1.1 MB	Pending	-

Upload them and make sure it is successful.

The screenshot shows the AWS S3 console interface after the upload has completed successfully. A modal window titled "Upload succeeded" is displayed, with the message "View details below." Below the modal is a table titled "Files and folders (157 Total, 18.2 MB)" showing the same list of files as before, but now all have a green "Succeeded" status in the status column. At the bottom of the screen, a taskbar displays various application icons.

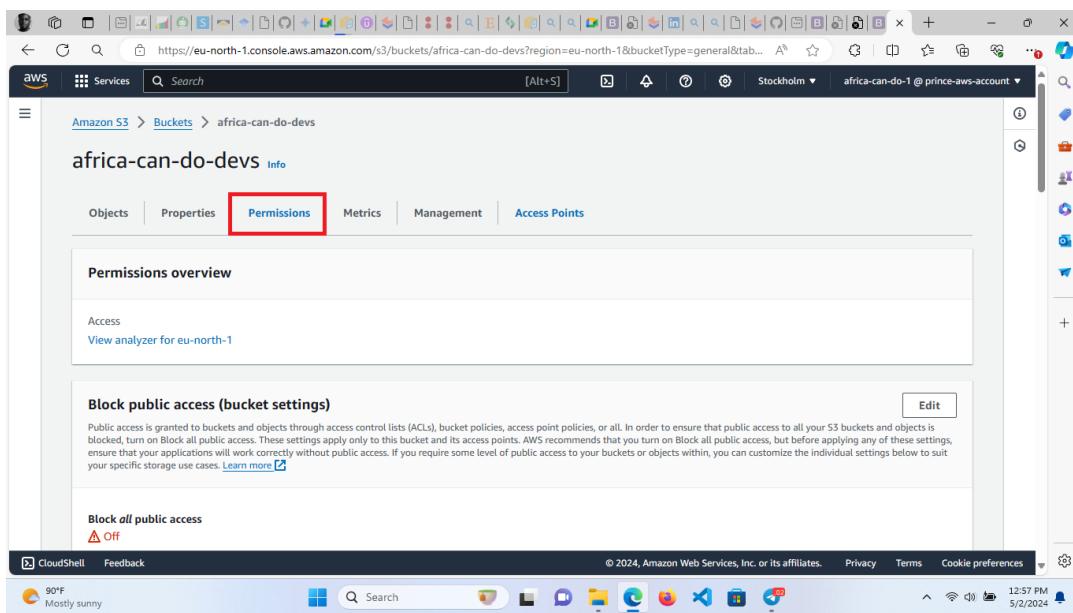
Name	Folder	Type	Size	Status	Error
norframewor...	assets/vend...	text/javascript	2.02 KB	Succeeded	-
Readme.txt	forms/	text/plain	176.0 B	Succeeded	-
noframewor...	assets/vend...	text/javascript	20.6 KB	Succeeded	-
swiper-bun...	assets/vend...	text/css	16.1 KB	Succeeded	-
swiper-bun...	assets/vend...	text/javascript	140.3 KB	Succeeded	-
swiper-bun...	assets/vend...	-	528.0 KB	Succeeded	-
remixicon.css	assets/vend...	text/css	107.8 KB	Succeeded	-
remixicon.eot	assets/vend...	-	393.8 KB	Succeeded	-
remixicon.less	assets/vend...	-	107.9 KB	Succeeded	-
remixicon.svg	assets/vend...	image/svg+	1.1 MB	Succeeded	-

Your folder will be created

- **Set Permissions:**

By default, all objects uploaded to an S3 bucket are private. If you want your website files to be publicly accessible, you need to adjust the permissions accordingly.

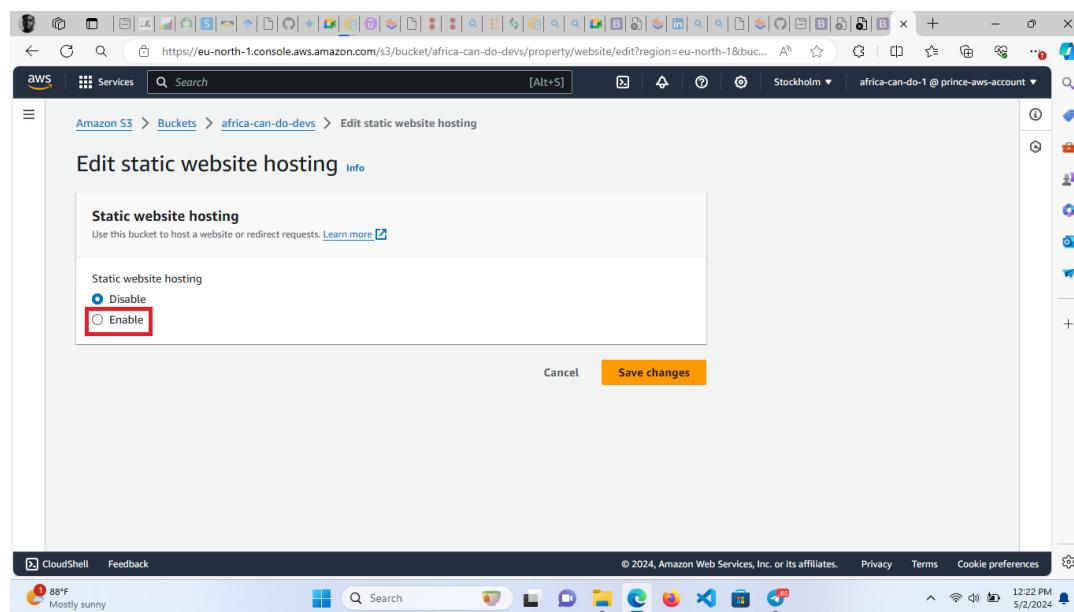
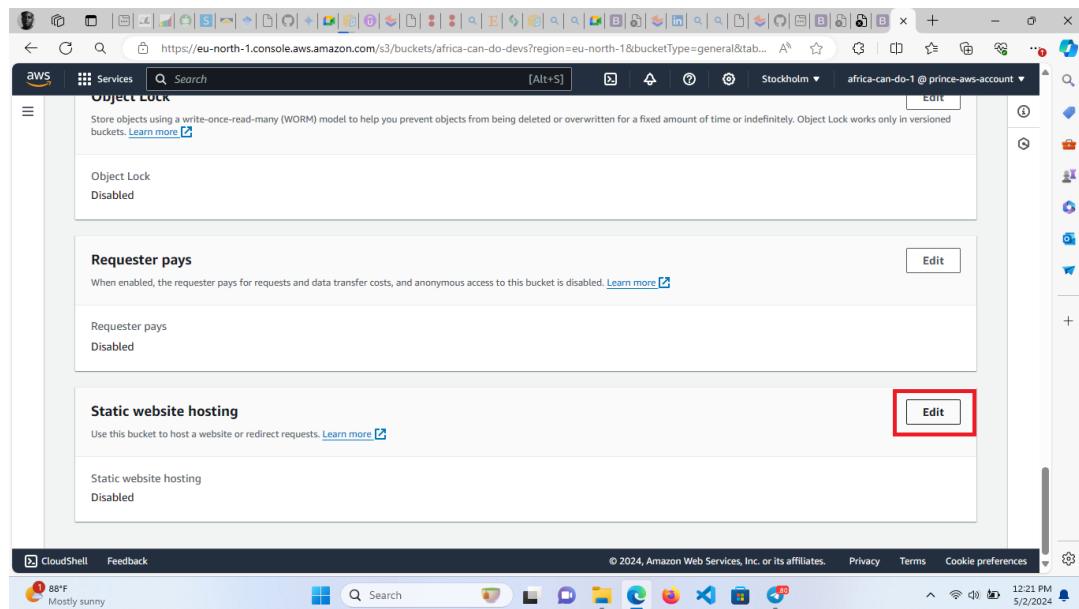
Select the uploaded files and folders, click on the "Actions" dropdown, and choose "Make Public" or adjust permissions manually in the "Permissions" tab.



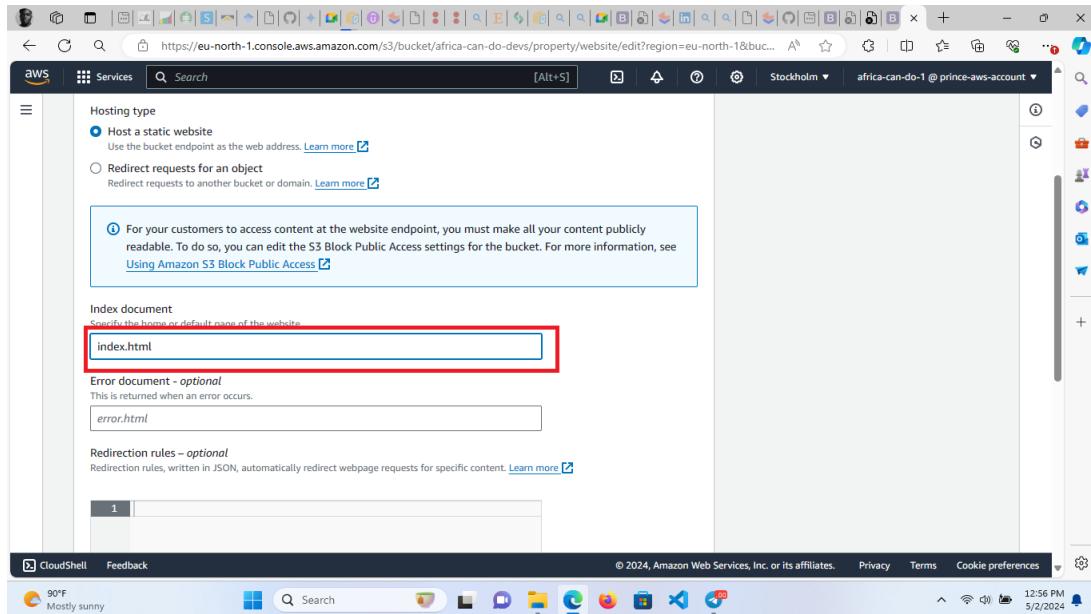
Click on “Make public

- **Enable Static Website Hosting:**

Once your files are uploaded, navigate to the "Properties" tab of your bucket. Scroll down to the "Static website hosting" card and click on “Edit” button. Enable “Static website hosting



Specify the index document (e.g., **index.html**) and error document if needed

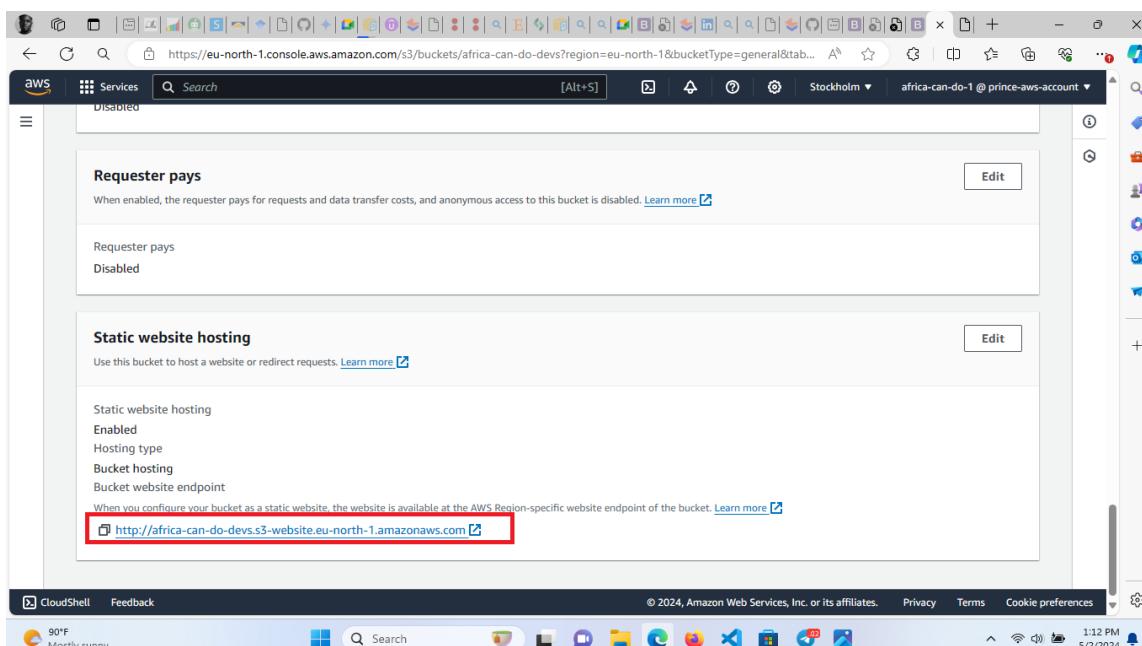


The screenshot shows the AWS S3 console for editing a bucket's properties. Under the 'Static website hosting' section, the 'Index document' field is set to 'index.html' and is highlighted with a red box. Below it, the 'Error document - optional' field contains 'error.html'. Other sections like 'Requester pays' and 'Static website hosting' are also visible.

Click "Save changes."

- **Access your website:**

After enabling static website hosting, AWS will provide you with a website endpoint URL (e.g., <http://your-bucket-name.s3-website-eu-north-1.amazonaws.com>). You can use this URL to access your website.



The screenshot shows the AWS S3 console for editing a bucket's properties. Under the 'Static website hosting' section, the 'Bucket website endpoint' field is set to 'http://africa-can-do-devs.s3-website.eu-north-1.amazonaws.com' and is highlighted with a red box. Other sections like 'Requester pays' and 'Static website hosting' are also visible.

- **Configure DNS (Optional):**

If you want to use a custom domain for your website, you can configure DNS settings to point to the S3 website endpoint. This typically involves creating a CNAME record pointing to the S3 endpoint in your DNS provider's dashboard.

- **Testing:**

Test your website by accessing it through the provided S3 website endpoint or your custom domain (**if configured**).

Once these steps are completed, your website contents will be hosted on Amazon S3 and accessible to users via the internet. Make sure to keep your website files updated by uploading new versions as needed.

Configuring bucket properties for website hosting.

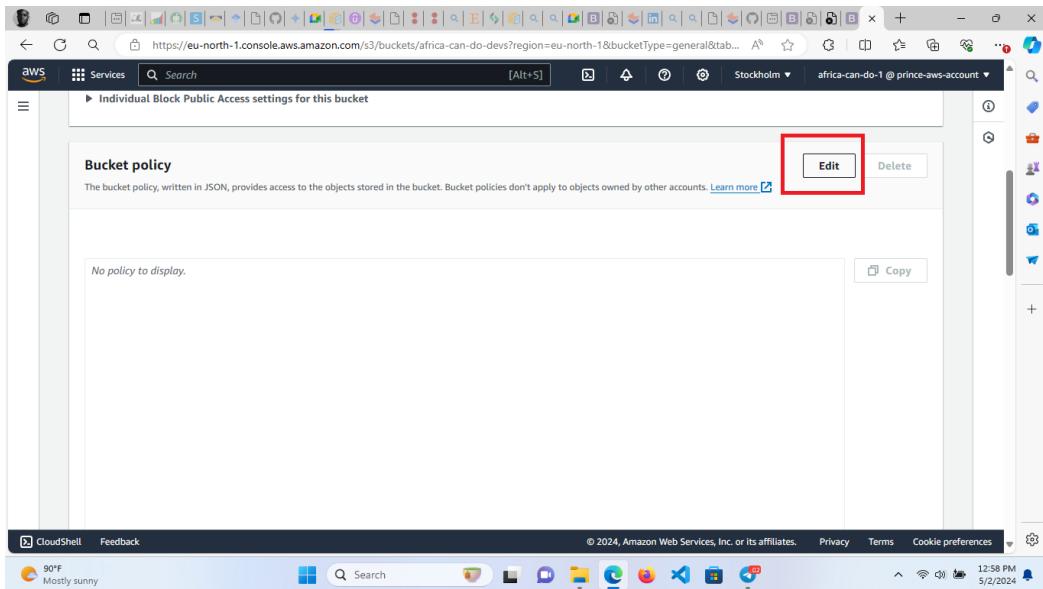
Configuring bucket properties for website hosting in Amazon S3 involves several steps to ensure that the bucket is properly set up to serve static website content. Here's a guide on how to do it:

- **Navigate to S3 Management Console:** Log in to your AWS Management Console and navigate to the Amazon S3 service.
- **Select the Bucket:** From the list of buckets, select the bucket you want to configure for website hosting.
- **Access Bucket Properties:** Once you've selected the bucket, click on the "Properties" tab in the top-right corner to access the bucket properties.
- **Enable Static Website Hosting:** Scroll down to the "Static website hosting" section and click on the "Edit" button.

Configure Hosting Options:

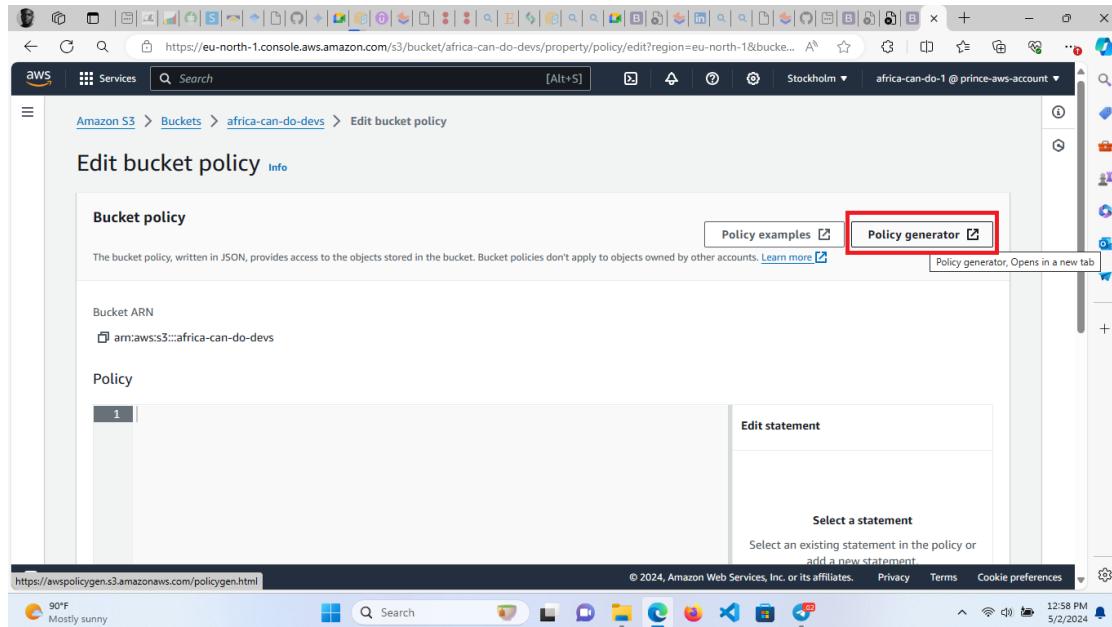
- **Use this bucket to host a website:** Select this option to enable website hosting for the bucket.
- **Index document:** Specify the name of the default HTML file that should be served when visitors access the root URL of your website (e.g., index.html).
- **Error document:** Optionally, specify a custom error document to display when errors occur (e.g., error.html).
- **Save Changes:** After configuring the hosting options, click on the "Save changes" button to apply the settings.
- **Set Bucket Policy (Optional):** To allow public access to your website content, you may need to configure a bucket policy.

- Click on the "Permissions" tab and then select "Bucket Policy."

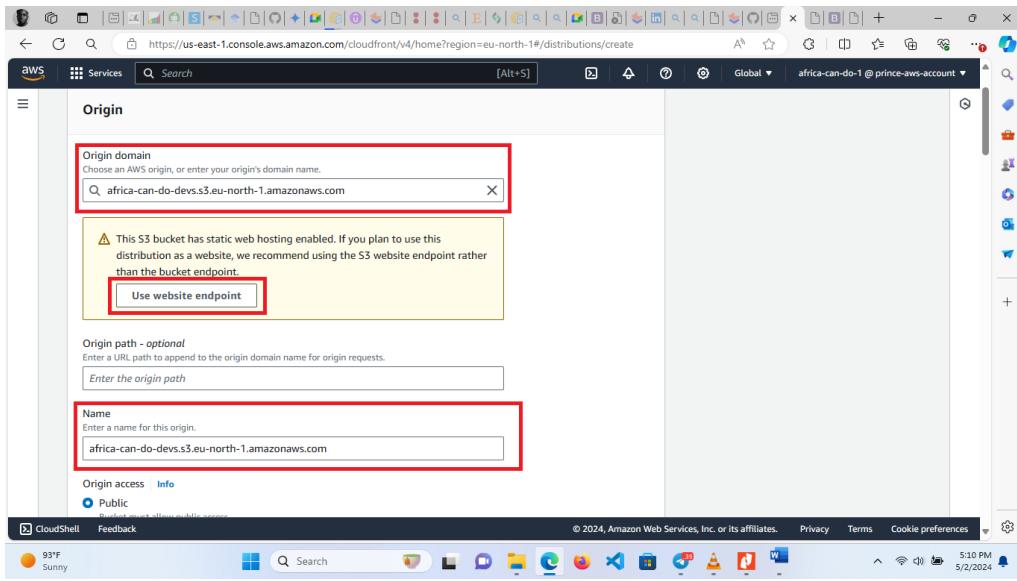


- You can then create a policy that grants read access to everyone (*) for the s3:GetBucketPolicy action on objects within the bucket. Here's an example bucket policy

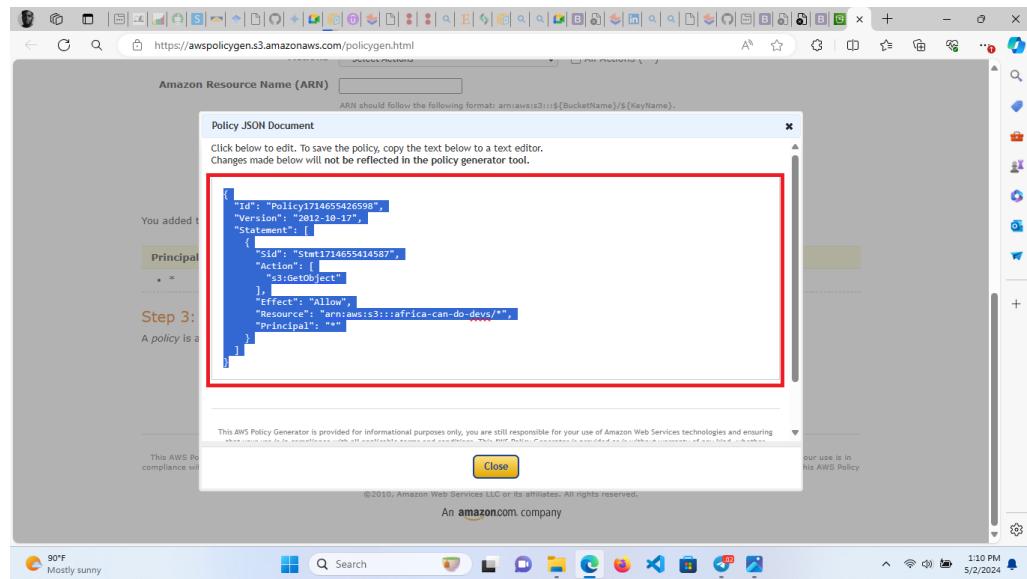
Generate a bucket Policy



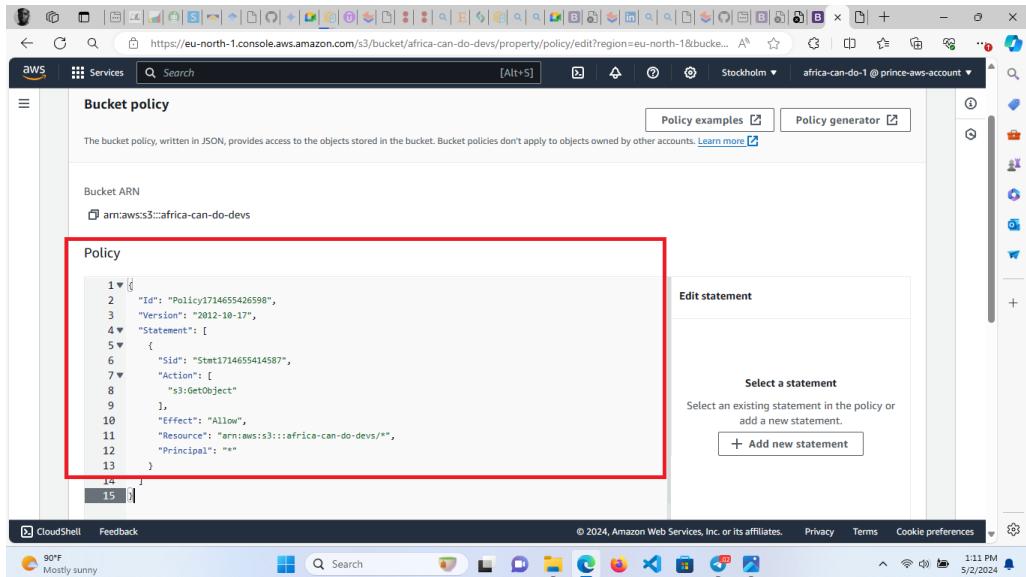
Click on Generate Policy after setting the preference for your policy



- The policy JSON Document is generated, make sure to copy the documents.

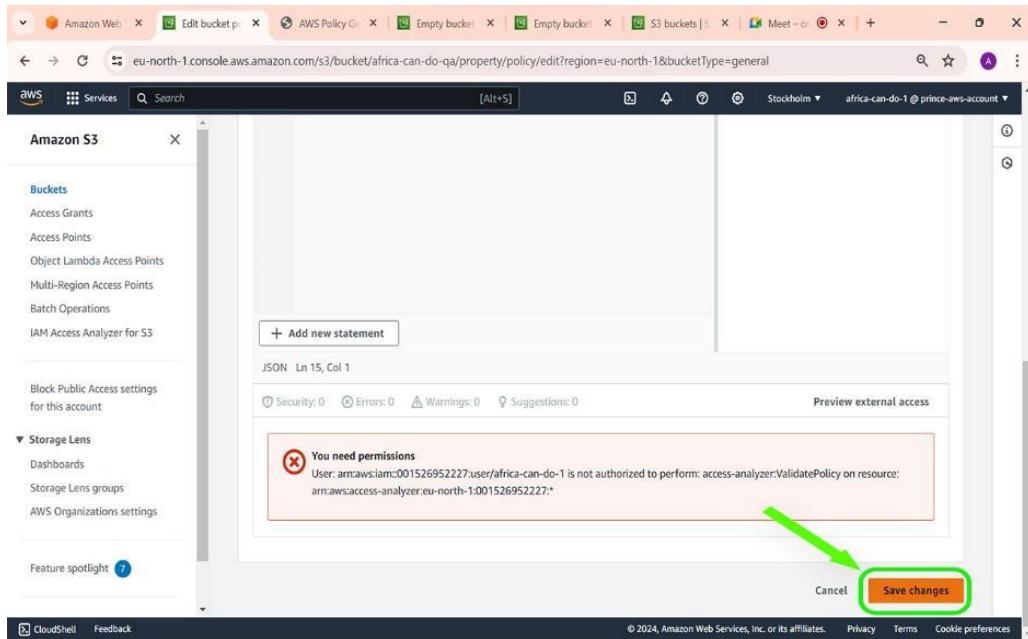


Go back and paste the policy you have copied in the policy container.



The screenshot shows the AWS Bucket Policy editor interface. In the center, there is a large text area labeled "Policy" containing a JSON policy document. The policy document grants "Allow" access to all objects in the bucket to the principal "arn:aws:s3:::africa-can-do-devs". The entire JSON block is highlighted with a red box. To the right of the policy area, there is a sidebar titled "Edit statement" with a sub-section "Select a statement" and a button "+ Add new statement". At the bottom of the main area, there is a "CloudShell" button and a feedback link. The status bar at the bottom right shows the date and time: "111 PM 5/2/2024".

- And save the changes



The screenshot shows the AWS S3 Bucket Properties page. On the left, there is a sidebar with options like "Buckets", "Access Grants", "Access Points", etc. The main area displays a JSON policy document with a validation error message: "You need permissions User: arn:aws:iam:001526952227:user:africa-can-do-1 is not authorized to perform: access-analyzer:ValidatePolicy on resource: arn:aws:access-analyzer:eu-north-1:001526952227:+". A green arrow points from the bottom right towards the "Save changes" button, which is highlighted with a green box. The status bar at the bottom right shows the date and time: "111 PM 5/2/2024".

Testing Website Accessibility

Accessing the website using the S3 bucket website endpoint URL.

To access the website using the S3 bucket website endpoint URL, follow these steps:

- **Retrieve the Website Endpoint URL:**

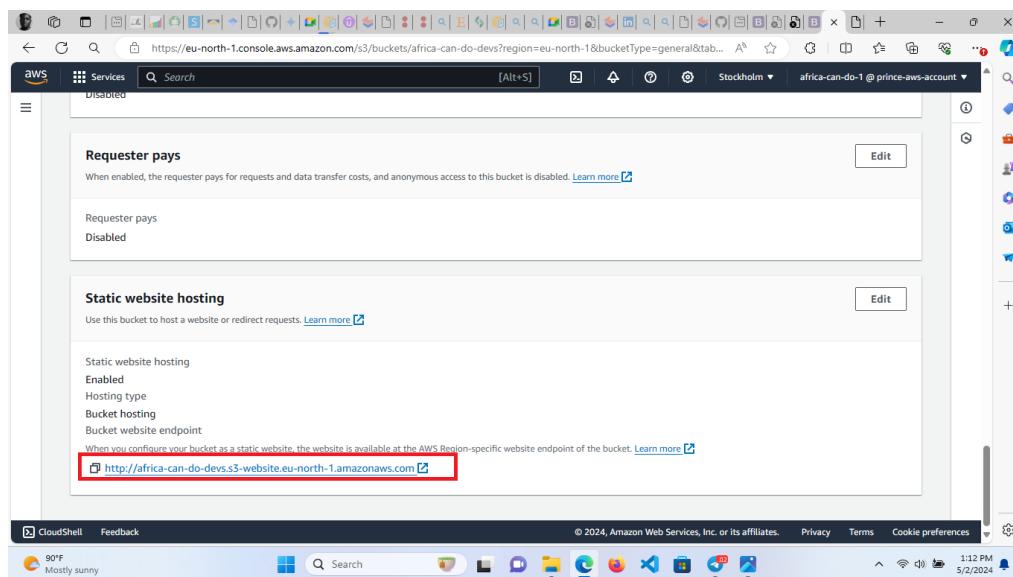
Navigate to the Amazon S3 Management Console.

Select the bucket configured for website hosting.

Go to the "Properties" tab.

Scroll down to the "Static website hosting" section.

Note the "Endpoint" URL provided under the "Hosting a Static Website" section. It typically follows the format: <http://<bucket-name>.s3-website-<AWS-region>.amazonaws.com>.



- **Access the Website:**

Open a web browser.

Enter the endpoint URL obtained in the previous step into the address bar.

Press Enter to navigate to the website.

- **Verify Website Access:**

Once the page loads, verify that the website content is displayed correctly. Navigate through different pages or sections of the website to ensure all content is accessible. Test any interactive features or functionality if applicable. If **Not secure** is visible with the URL of the website, it is normal at this stage and will be solved after serving the site with Cloudfront in the next chapter.

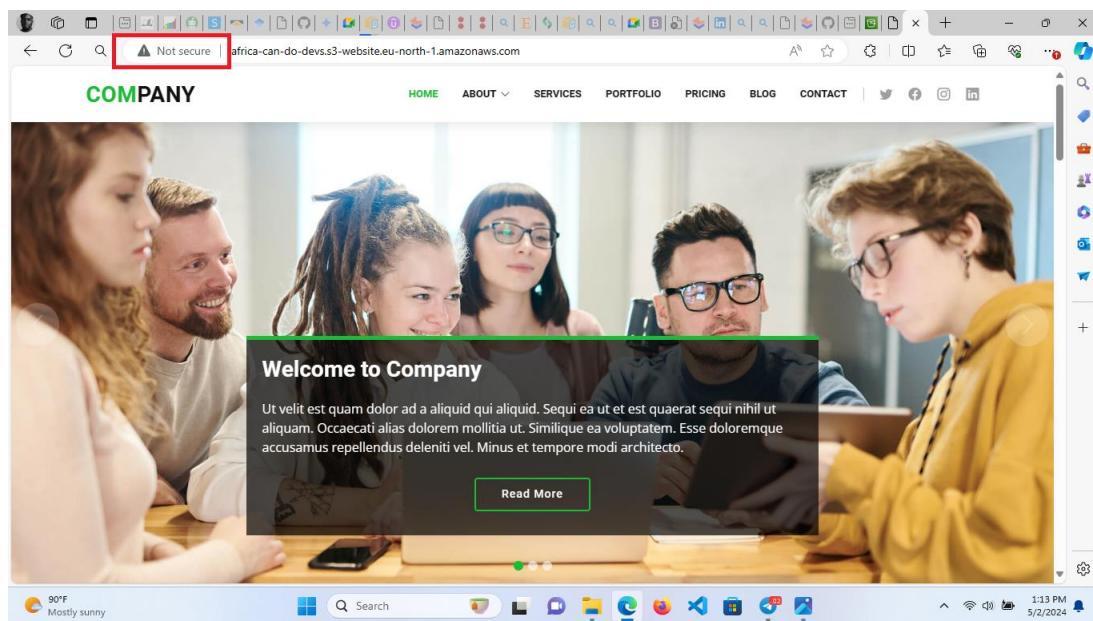
- **Troubleshooting (if necessary):**

If you encounter any errors or issues while accessing the website, double-check the endpoint URL for accuracy.

Ensure that the bucket is configured for static website hosting and that the index document specified in the bucket properties exists.

Verify the permissions of the objects in the bucket to ensure they are publicly accessible if necessary.

Check the browser console for any error messages that may provide insights into the issue. By following these steps, you can easily access the static website hosted on Amazon S3 using the bucket's website endpoint URL. This allows you to verify that the website is accessible to visitors before proceeding with additional configurations or optimizations.



Setting Up CloudFront Distribution

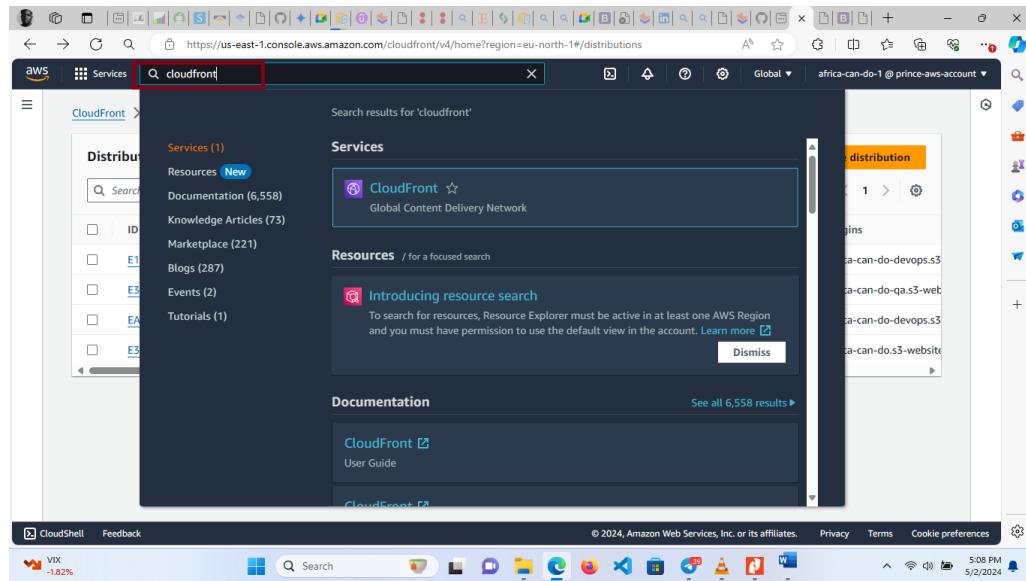
Creating a CloudFront distribution. Here's how you can create a CloudFront distribution:

- **Sign in to the AWS Management Console:**

Go to the AWS Management Console (<https://aws.amazon.com/console/>) and sign in to your AWS account.

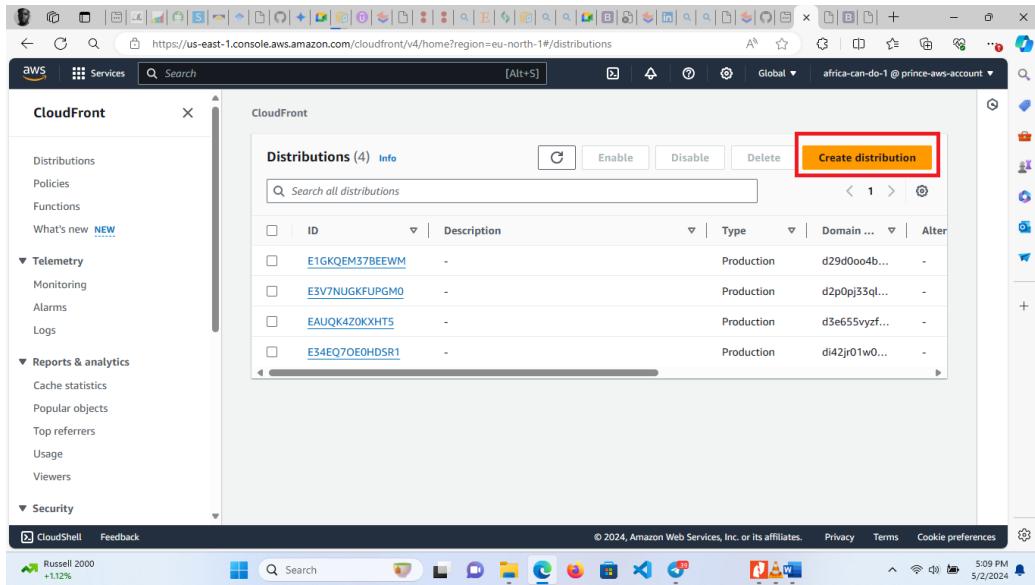
- **Navigate to CloudFront:**

Go to the CloudFront service by typing "CloudFront" in the search bar at the top or by selecting it from the list of services.

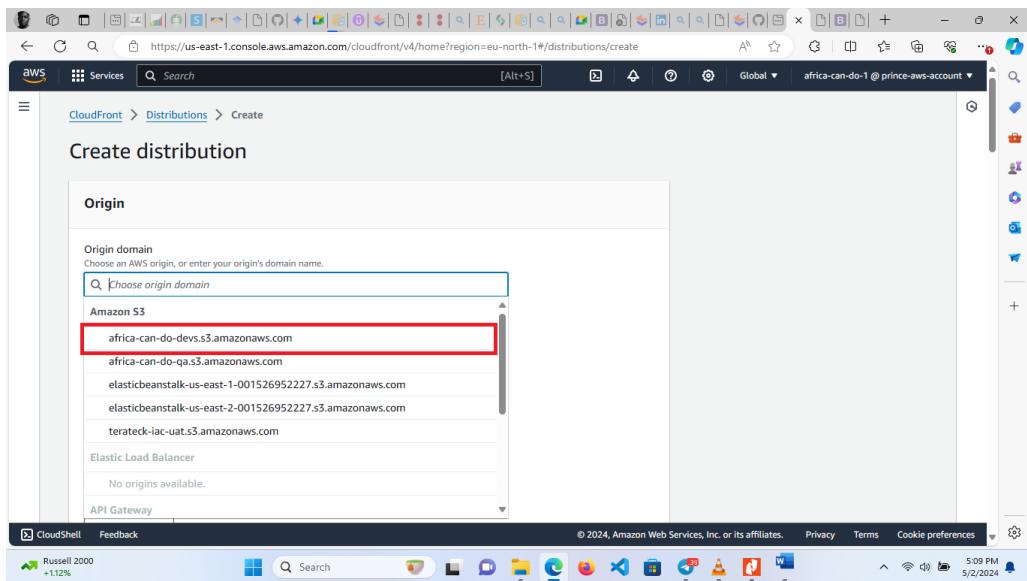


- **Create Distribution:**

In the CloudFront dashboard, click on the "Create Distribution" button.

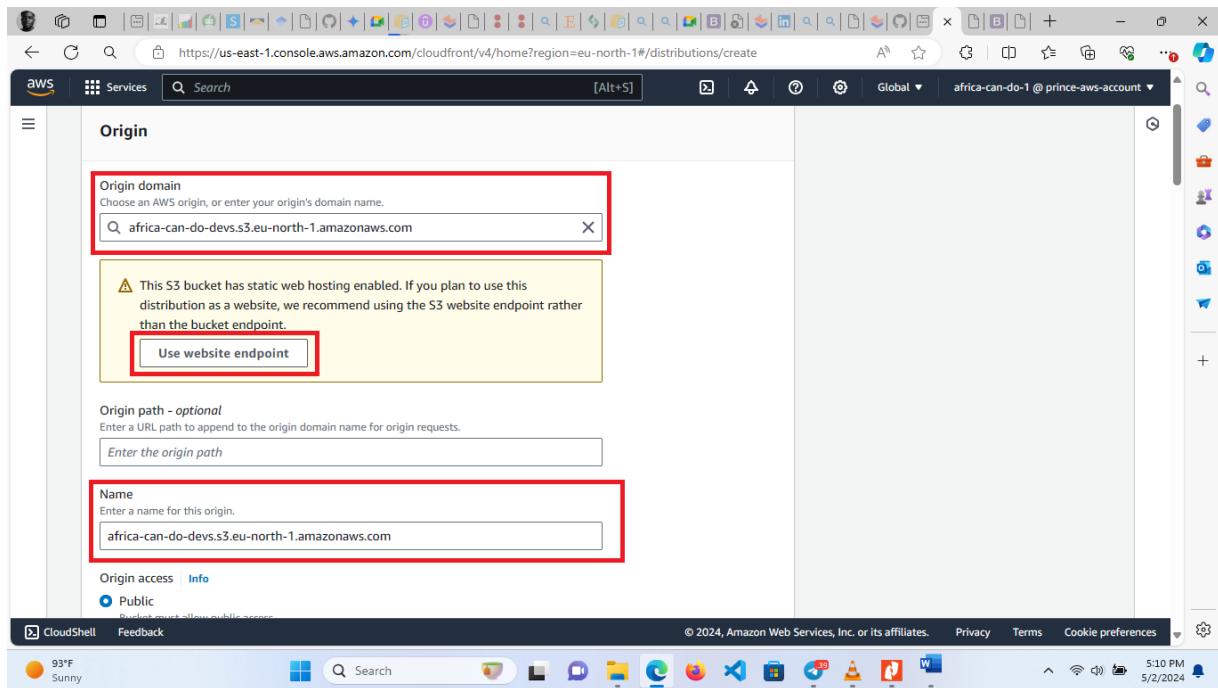


- Select the S3 bucket that the static website is Hosted on



Choose a Delivery Method:

CloudFront supports two delivery methods: Web and RTMP. Choose "Get Started" under the Web section since you're setting up a web distribution



- **Create Distribution:**

Configure the settings for your distribution:

- **Origin Settings:** Select the S3 bucket or custom origin (e.g., an EC2 instance, load balancer, or custom HTTP server) from which CloudFront will fetch your content.
- **Default Cache Behavior Settings:** Configure caching settings, such as viewer protocol policy, cache and origin request policies, allowed HTTP methods, etc.
- **Distribution Settings:** Configure additional settings such as the distribution's price class, SSL certificate, supported HTTP versions, etc.
- **Logging:** Optionally configure CloudFront logging to capture access logs.
- **Tagging:** Optionally add tags to your distribution for easier management.
- **Errors:** Customize error pages if desired.
- Click on the "Create Distribution" button once you've configured all the settings.

- **Wait for Deployment:**

It may take some time for your CloudFront distribution to deploy. You can track the progress in the CloudFront console.

- **Configure DNS (Optional):**

If you want to use a custom domain with your CloudFront distribution, you need to configure DNS settings to point to the CloudFront domain name.

- **Testing:**

Once the distribution is deployed and DNS (if applicable) is configured, test your website or application by accessing it through the CloudFront domain name.

- **Monitor and Manage:**

After your CloudFront distribution is up and running, you can monitor its performance, manage settings, and make adjustments as needed in the CloudFront console.

Configuring the origin settings to point to the S3 bucket.

Configuring the origin settings in Amazon CloudFront to point to the Amazon S3 bucket involves specifying the S3 bucket as the origin from which CloudFront should retrieve content. Here's how to do it:

- **Navigate to CloudFront Console:**

Log in to your AWS Management Console. Go to the CloudFront service.

- **Select Distribution:**

From the CloudFront dashboard, select the distribution that you want to configure to point to the S3 bucket.

- **Edit Distribution:**

Click on the distribution's ID or its domain name to open its details page. In the "General" tab, click on the "Edit" button to modify the distribution settings.

- **Configure Origin Settings:**

Scroll down to the "Origins and Origin Groups" section.

You'll see the existing origin(s) listed. Click on the origin that corresponds to your S3 bucket.

- **Edit Origin Settings:**

In the "Edit Origin" section, ensure that the "Origin Domain Name" field points to the S3 bucket's website endpoint URL.

If the S3 bucket is configured for website hosting, the origin domain name should be in the format <bucket-name>.s3-website-<AWS- region>.amazonaws.com.

- **Specify Origin Path (Optional):**

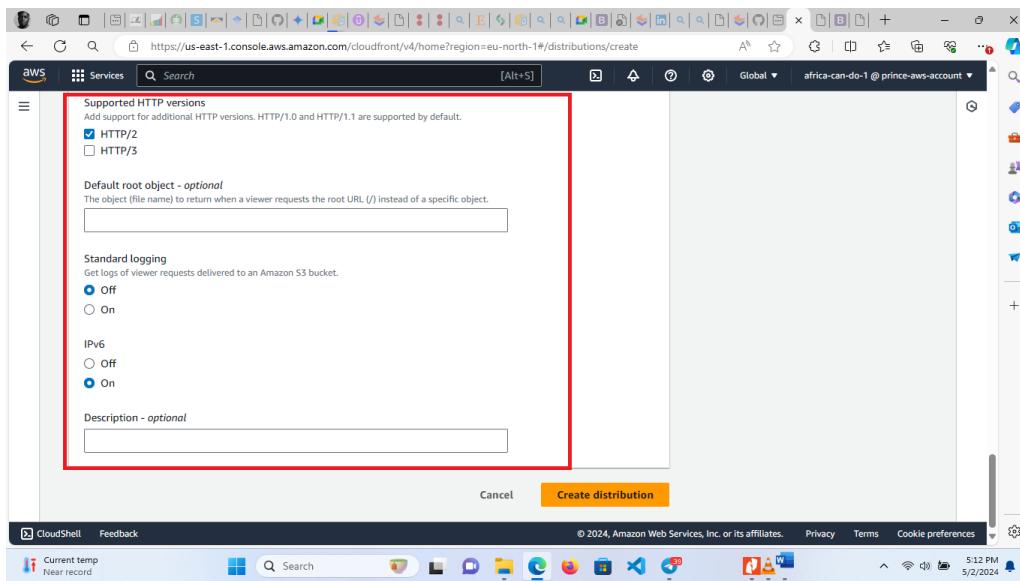
If your website content is stored in a specific folder within the S3 bucket, you can specify the origin path

For example, if your content is stored in a folder named "public" within the bucket, set the "Origin Path" to /public.

- **Configure Origin Protocol Policy:**

Choose the appropriate origin protocol policy based on your website's requirements.

Generally, it's recommended to select "HTTP Only" or "Match Viewer" to ensure compatibility with various client devices and browsers.



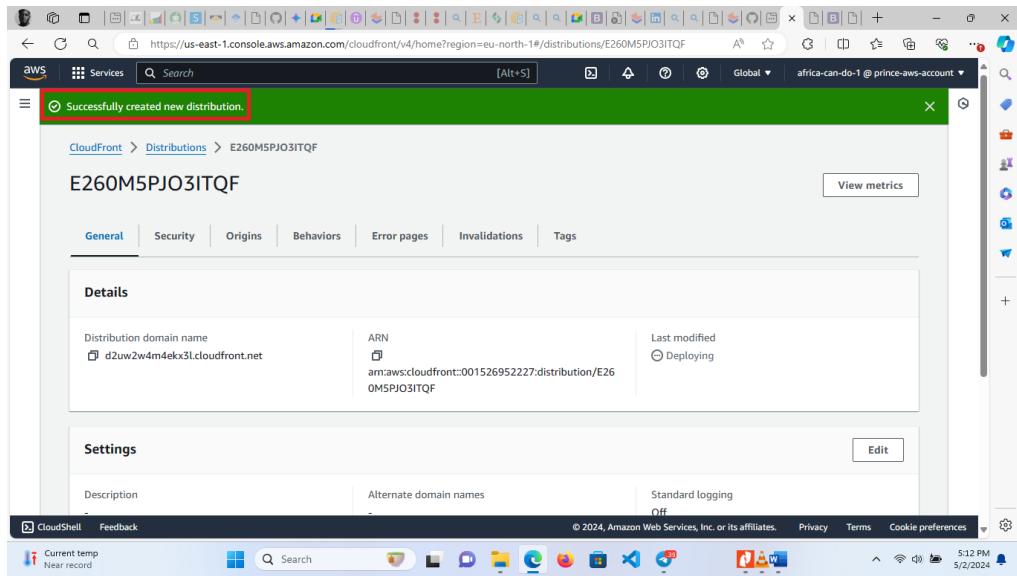
- **Save Changes:**

Once you've configured the origin settings, click on the "Yes, Edit" button to save your changes.

- **Distribution Deployment:**

CloudFront distributions can take several minutes to deploy changes. Wait for the distribution status to change to "Deployed" before proceeding.

By following these steps, you can configure the origin settings in Amazon CloudFront to point to the Amazon S3 bucket, allowing CloudFront to retrieve content from the bucket and serve it to users via its global content delivery network (CDN).



Testing Website Delivery via CloudFront

Accessing the website using the CloudFront domain name.

To access the website using the CloudFront domain name, follow these steps:

- **Retrieve the CloudFront Domain Name:**

Log in to your AWS Management Console.

Go to the CloudFront service.

Select the distribution that you have

configured for your website.

- **Find the CloudFront Domain Name:**

In the distribution details, locate the "Domain Name" field under the "General" tab. This is the domain name that CloudFront assigns to your distribution.

- **Update DNS Records (Optional):**

If you have a custom domain (e.g., www.example.com) that you want to use for your website, you'll need to update your DNS settings to point to the CloudFront domain name.

Log in to your DNS provider's website (e.g., Route 53, GoDaddy, etc.).

Create a CNAME record that points your custom domain to the CloudFront domain name.

- **Access the Website:**

Open a web browser.

Enter the CloudFront domain name into the address bar.

Press Enter to navigate to the website.

- **Verify Website Access:**

Once the page loads, verify that the website content is displayed correctly.

Navigate through different pages or sections of the website to ensure all content is accessible.

Test any interactive features or functionality if applicable.

- **SSL/TLS Certificate Validation:**

If you've configured CloudFront to use HTTPS, ensure that your SSL/TLS certificate is valid and trusted by the browser. You should see a padlock icon in the browser's address bar indicating a secure connection.

- **Troubleshooting (if necessary):**

If you encounter any errors or issues while accessing the website, double-check the CloudFront domain name for accuracy.

Ensure that your DNS records are correctly configured if you're using a custom domain. Check the CloudFront distribution settings for any misconfigurations that may affect website delivery.

By following these steps, you can easily access the static website hosted on Amazon S3 via the CloudFront domain name, providing users with a fast and reliable browsing experience leveraging CloudFront's global content delivery network (CDN).

•

Verifying that the website is being served through CloudFront

To verify that the website is being served via CloudFront, you can use various methods to confirm that the CloudFront distribution is indeed delivering your website content. Here's how you can verify it:

- Check Response Headers:**

Open your web browser and navigate to your website using the CloudFront domain name. Right-click on the webpage and select "Inspect" or "Inspect Element" to open the browser's developer tools.

Go to the "Network" tab.

Refresh the page if necessary.

Look for requests with the domain name of your website. The "cf-cache-status" response header should indicate "hit" or "miss." If it's "hit," it means the content is served from the CloudFront cache.

- View CloudFront Access Logs:**

Access the Amazon S3 bucket where you configured CloudFront to store access logs.

Look for log files with recent timestamps.

Open the log files and check for requests to your website. You should see requests with

CloudFront's domain name in the "cs(Referer)" field.

- CloudFront Monitoring Dashboard:**

Navigate to the CloudFront console.

Select your distribution.

Go to the "Monitoring" tab.

Check the "Requests" metric to see if there's traffic being served by CloudFront.

- **CloudFront Distribution Statistics:**

In the CloudFront console, select your distribution.

Go to the "General" tab.

Review the distribution statistics, such as "Total Requests," "Bytes Transferred," and "Data Transfer Out." You should see an increase in these metrics if your website is being accessed through CloudFront.

- **Origin Server Access Logs:**

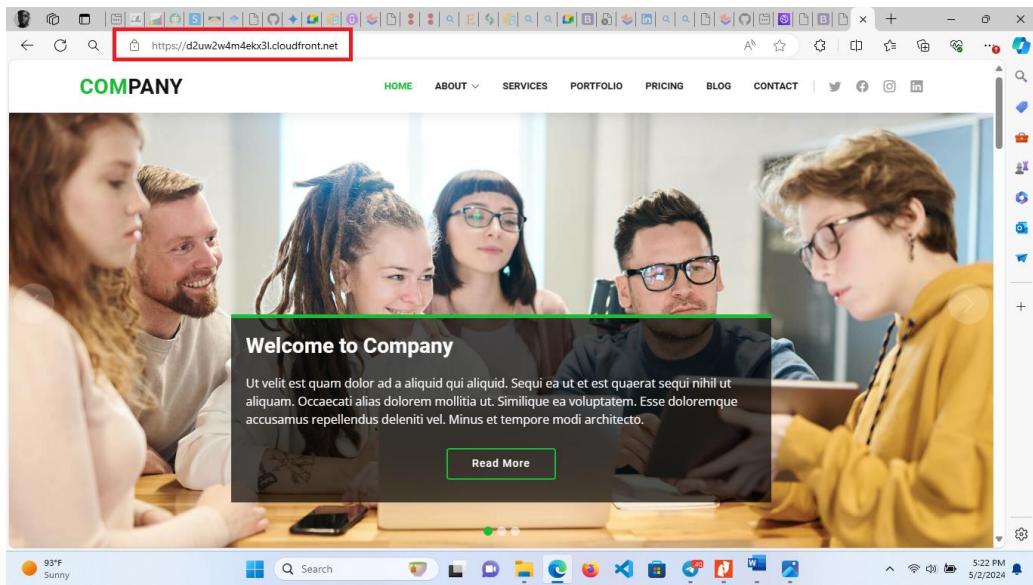
If your origin server (e.g., S3 bucket or custom origin) logs requests, you can check the logs to see if requests are coming from CloudFront IP addresses. This can confirm that CloudFront is serving your website content.

By using these methods, you can verify that your website is indeed being served via CloudFront, ensuring that your users are benefiting from the improved performance and reliability provided by the content delivery network (CDN).

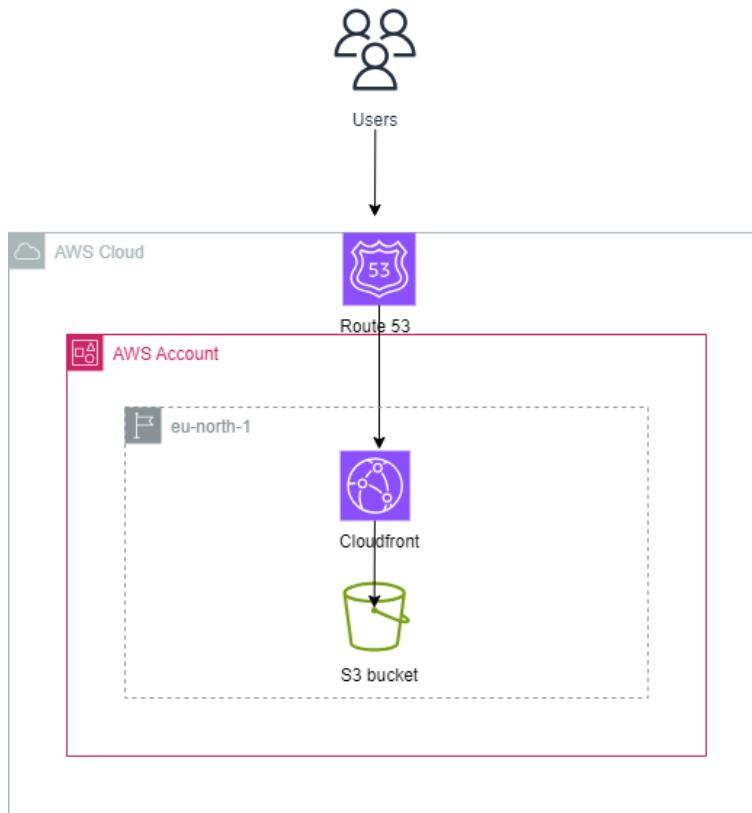
The screenshot shows the AWS CloudFront console with the URL <https://us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=eu-north-1#/distributions/E260M5PJO3ITQF>. The left sidebar has sections for Distributions, Telemetry, Reports & analytics, and Security. The main area shows a distribution named 'E260M5PJO3ITQF'. The 'General' tab is selected. In the 'Details' section, the 'Distribution domain name' field is highlighted with a red box and contains the value 'd2uw2w4m4ekx3l.cloudfront.net'. The 'ARN' field shows 'arn:aws:cloudfront:001526952227:distribution/E260M5PJO3ITQF'. The 'Settings' section includes fields for 'Description' (empty), 'Alternate domain names' (empty), 'Standard logging' (Off), 'Price class' (Use all edge locations (best)), and 'Cookie logging' (Off). A 'View metrics' button is visible at the top right of the main content area.

- **Note:** The Not secured is no longer be identified with URL of the site in the figure below.

The website URL is now secured



Cloud Architecture



DESCRIPTION

S3 Bucket: Store your static website files (HTML, CSS, JavaScript, images, etc.) in an Amazon S3 bucket. Ensure the bucket is configured for static website hosting.

Route 53: Use Route 53, Amazon's DNS service, to manage your domain's DNS settings. You'll create a record set that points to your CloudFront distribution.

CloudFront Distribution: Create a CloudFront distribution to serve your static website content. Configure the distribution to use your S3 bucket as the origin.

Users: Users access your website via their web browsers. When they request a page, their DNS lookup (handled by Route 53) directs them to the CloudFront distribution.

CloudFront Cache: CloudFront caches your website's content at edge locations around the world, improving performance and reducing latency for users.

Authors:

- Prosper Kwaku Galenku
[\(19\) Prosper \(kwaku\) Galenku | LinkedIn](#)
- Asiwome Boateng
[\(19\) Asiwome Boateng | LinkedIn](#)
- Hope Decardi-Nelson
[\(19\) Hope Decardi-Nelson | LinkedIn](#)
- Sorkpor Ruth Eyram
[\(19\) Ruth \(Eyram\) Sorkpor | LinkedIn](#)
- Emmanuel A. Agyei
[\(19\) EMMANUEL A. AGYEI | LinkedIn](#)
- Abu Fati
[\(19\) Fati Abu | LinkedIn](#)
- Mustapha Tahiru
[\(19\) Tahiru Mustapha | LinkedIn](#)
- Prince Yaw Asiedu
[\(19\) Prince Yaw Asiedu | LinkedIn](#)

Hosting a static website using S3 and Cloudfront

Automation Process/ Workflow Method

1. First, Open the S3 on the AWS console and create a new S3 Bucket
2. create an S3 bucket and name it say, "dev.auroraverusschool.org" always good to name the S3 after the domain name you want to use
3. Enable ACLs in bucket creation (ACLs enabled) under the Object Ownership
4. Uncheck the "Block all public access" and check the acknowledgement check box
5. Enable bucket Versioning
6. Click Create Bucket
7. when bucket is created, select the created bucket by clicking on it
8. Policies
 1. Go to permissions, Policies and open the Policy generator
 2. Under type of Policy, select "S3 bucket Policy"
 3. Put * for the Principal section
 4. Under Actions, select GetObject only
 5. Under Amazon Resource Name (ARN), copy your bucket ARN at where the bucket policy and come paste in there and add `/*` (forward slash all) to it
 6. add statement and generate policy,
 7. copy, and paste in the policy and save
9. Go to properties and enable static website hosting at the very end of the section, under "Static Website Hosting"
 1. default index is index.html
 2. save changes
10. now if you visit the bucket link just at the "Static Website Hosting" it will say key not found
11. now go back to the jobs file, the main.yaml file
 1. make sure you are in your own branch, I repeat, make sure you are in your own branch, else create your branch and all works should be pushed to the created to that branch and do not merge to main but to the "dev" branch

2. under "run", change the "s3 bucket link" to the name of the bucket that has been created which is "dev.auroraverusschool.org"

3. rename all sections of the job in the yaml file to the name of the branch to which the code will be pushed to, which is "dev"

12. Create a Cloudfront Distribution

1. Open Cloudfront from the AWS console

2. Select "create distribution"

3. origin : select the bucket that you just created, this one starts with the "dev.auroraverusschool.org.s3"

4. it's a website so select HTTP

5. under Web Application Firewall (WAF), select "Do not enable security protections"

6. leave everything else as it is and create the distribution

7. copy the distribution ID, written big and boldly at the top and paste it in the "distribution ID" section under the "run" command of your jobs, in the main.yaml file

13. edit or make some additions in the index.html file

14. save

15. now add using `git add .`

16. then commit using `git commit -m "comment"``

17. then push using `git push`

18. Now when it is pushed, GO TO GITHUB

1. create the pull request

2. compare your branch to the "dev" branch

3. create the pull request

4. merge

5. check the actions to see the kind of job it will pick

6. it should skip any other job that is not related to the "dev"

19. Now check the S3 bucket and everything should be deployed

20. Create an alternative domain which is the "dev.auroraverusschool.org"

1. go back to cloudfront

2. Under "General", go to "Settings" and select "Edit"

3. under "Alternative domain name", type "dev.auroraverusschool.org"

4. under "Custom SSL certificate", choose the already created certificate "auroraverusschool.org" certificate

5. leave the rest as it is and save changes

21. Now you have to redirect all traffic from HTTP to HTTPS

1. still in the cloudfront, go to "Behaviours" section

2. select or tick your distribution and select "Edit"

3. change "HTTP and HTTPS" to "Redirect HTTP to HTTPS"

4. save changes

22. now go to dev.auroraverusshcool.org and realise it is serving properly

23. Open Route 53

1. check if the record is hitting the right Cloudfront Endpoint, if not, create one or make sure it is

2. if editing or creating, select and edit or create one

3. in the "Record name" section, put "dev" because the ".auroraverusschool.org" is already written for you

4. check on the "Alias"

5. "Route traffic to": select "Alias to Cloudfront Distribution"

6. save and refresh

24. now you can open the devs.auroraverusschool.org, everything should work fine