



# **CLOUD FRONT DISTRIBUTION**

**Prepared by Lalit Kumar**

 [linkedin.com/in/lalit192977](https://www.linkedin.com/in/lalit192977)

## What is AWS CloudFront?

CloudFront is a **Content Delivery Network (CDN)** service provided by AWS. Its main job is to deliver **content (websites, images, videos, APIs, files)** to users **faster, securely, and with low latency**.

👉 Imagine your server is in the US, but a user is opening your site from India.

**Without CloudFront** → the request goes all the way to the US → slow response.

**With CloudFront** → AWS stores (caches) your content in **Edge Locations (global data centers)**. The Indian user gets content from the nearest edge (like Mumbai or Singapore) → much faster.

## How CloudFront Works (Simple Flow)

1. User requests your website → goes to CloudFront.
2. CloudFront checks:
  - If content is cached at the nearest edge → it sends immediately.
  - If not → it fetches from your Origin (S3, EC2, Load Balancer, API Gateway, etc.), caches it, then sends to the user.
3. Next user request from that region → gets super-fast response from the cache.

## Common Use Cases

1. Static Website Hosting – Store files in S3 + serve via CloudFront for global reach.
2. Video Streaming – Used by platforms like Netflix/Prime for smooth streaming.
3. Dynamic Content Acceleration – Speeds up APIs, dashboards, logins.

4. E-Commerce – Faster loading of product images, catalogs, and checkout pages.
5. Software Downloads – Ideal for large file delivery (games, apps, OS updates).
6. Security – Integrates with AWS WAF & Shield for protection against DDoS and attacks.

## **Why Use CloudFront? (Benefits)**

-  Global Edge Network → users get content from the nearest AWS location.
-  High Performance → cached data makes repeat requests super fast.
-  Cost Effective → reduces load on your servers, saving bandwidth + compute cost.
-  Secure → HTTPS, custom SSL, WAF, geo-restrictions, DDoS protection.
-  Scalable → handles sudden traffic spikes automatically.

## **Where to Use CloudFront?**

- When your website/app serves a global audience.
- For high-traffic sites where you don't want to overload your servers.
- On image-heavy / video-heavy platforms.
- For E-commerce websites needing fast + secure checkout experience.
- To accelerate APIs used in mobile or SaaS apps.

## Let's do practice

Lets starts with the ec2 instance server, give name to the instance and choose AMI here we choose ubuntu

The screenshot shows the AWS Lambda console with the 'Create new function' wizard. The first step, 'Select runtime environment', is completed. The second step, 'Configure triggers', is currently selected. In the 'Triggers' section, there is a table with one row:

Type	Name	Last triggered	Actions
Amazon CloudWatch Logs	CloudWatchLogsLogGroupTrigger	2023-09-15 14:45:20 UTC	Remove

Below the table, there is a note: 'This function will receive logs from the specified log group. You can also trigger this function from other sources like CloudWatch Metrics or S3.'

From the network settings create a security group for this server allowing http (port-80) and ssh (port-22)

The screenshot shows the AWS Lambda console with the 'Create new function' wizard. The second step, 'Configure triggers', is completed. The third step, 'Configure function', is currently selected. In the 'Function configuration' section, the 'Runtime' is set to 'Node.js 18.x'. Under 'Environment variables', there is a table with one row:

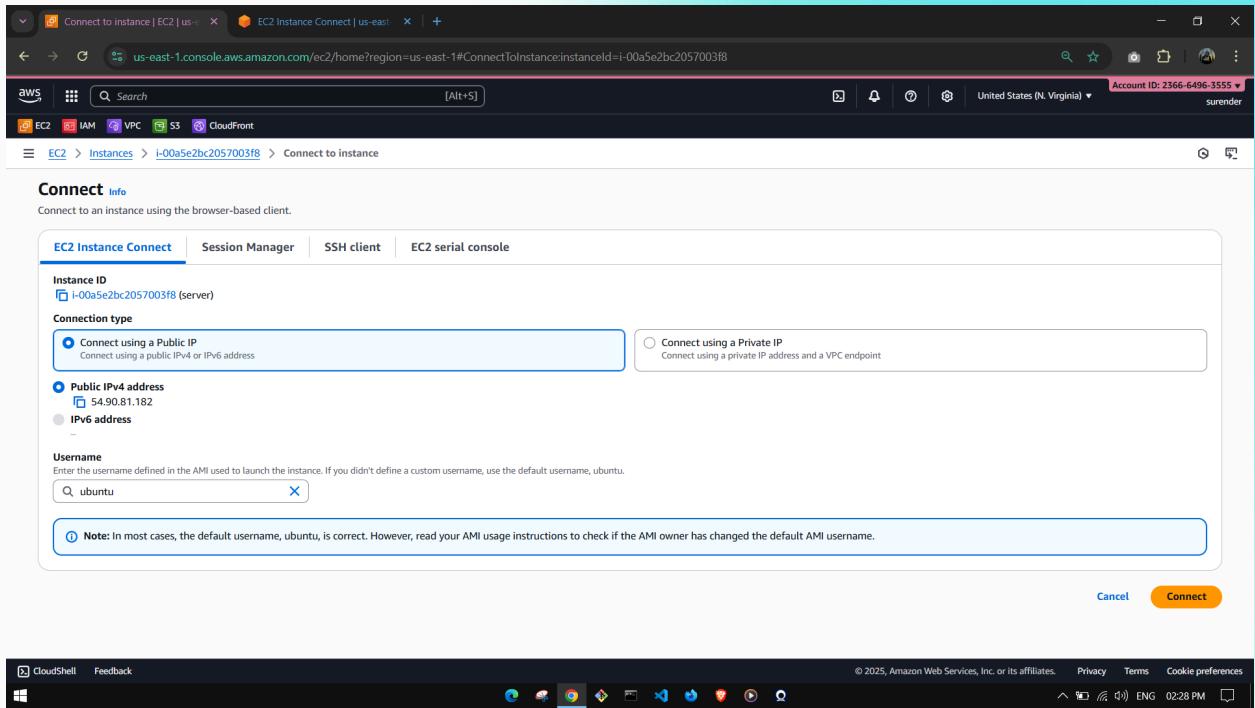
Name	Value
NAME	lambda

Under 'Handler', the value is 'index.handler'. In the 'Role' section, a role named 'lambda-role' is selected. In the 'VPC settings' section, the VPC 'vpc-09c481682000e8b9f' is selected. In the 'Network settings' section, the subnet 'No preference (Default subnet in any availability zone)' is selected. Under 'Auto-assign public IP', 'Enable' is selected. In the 'Firewall (security groups)' section, the 'Create security group' button is selected. A note says: 'We'll create a new security group called 'lambda-wizard-3' with the following rules:'

- Allow SSH traffic from Anywhere (0.0.0.0/0)
- Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server

At the bottom, there is a note: 'A rule with source of 0.0.0.0/0 allows all IP addresses to access your instance. We recommend setting security group rules to restrict access to your instance.'

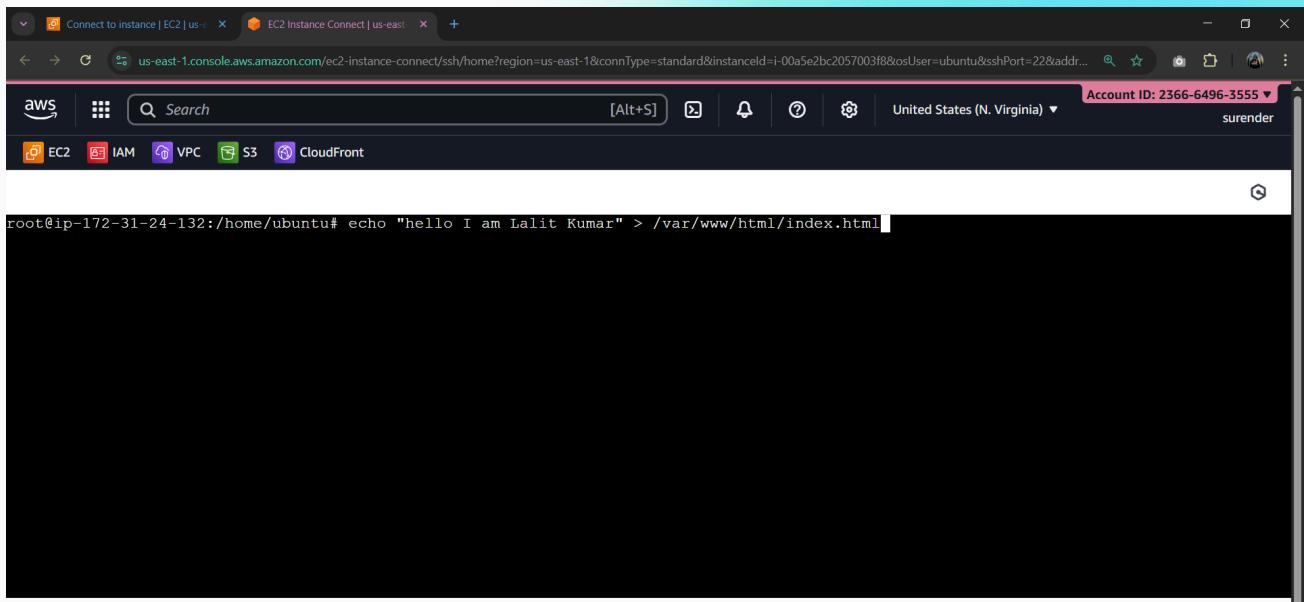
## Connect your server



After connecting it update it and install nginx server through the below commands-

apt update and after that run apt install nginx -y

Now configure your nginx server



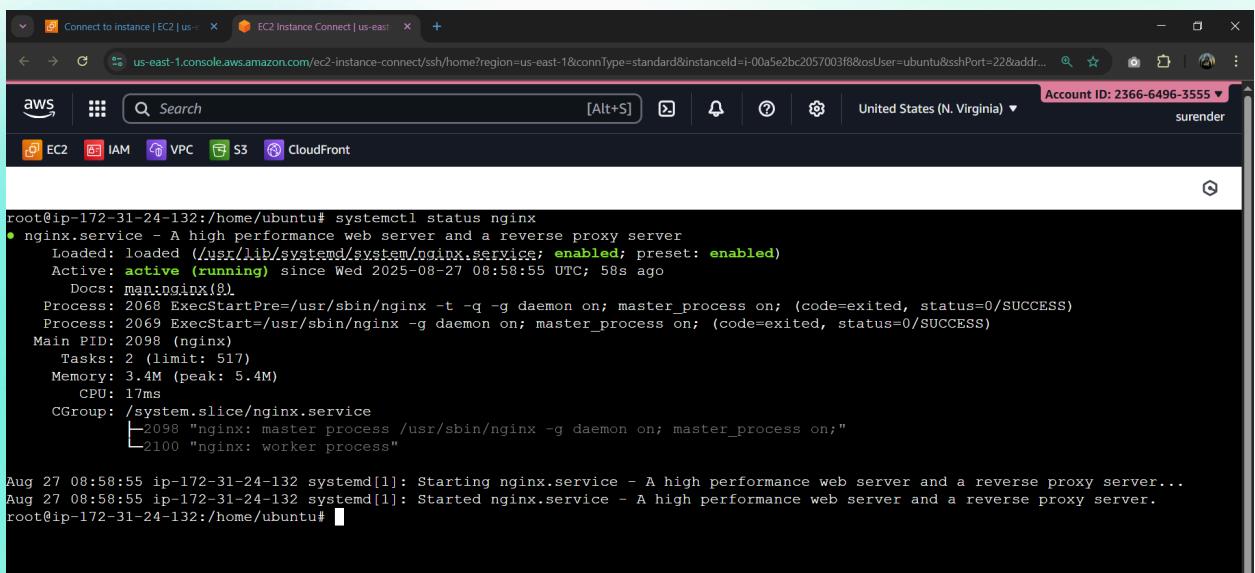
root@ip-172-31-24-132:/home/ubuntu# echo "hello I am Lalit Kumar" > /var/www/html/index.html

i-00a5e2bc2057003f8 (server)

PublicIPs: 54.90.81.182 PrivateIPs: 172.31.24.132

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Check the nginx server is running or not shown in the below image if the nginx server is not running then run it through **systemctl start nginx**



```
root@ip-172-31-24-132:/home/ubuntu# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Wed 2025-08-27 08:58:55 UTC; 58s ago
    Docs: man:nginx(8)
  Process: 2068 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 2069 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 2098 (nginx)
   Tasks: 2 (limit: 517)
  Memory: 3.4M (peak: 5.4M)
    CPU: 17ms
   CGroup: /system.slice/nginx.service
           └─2098 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─2100 "nginx: worker process"

Aug 27 08:58:55 ip-172-31-24-132 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Aug 27 08:58:55 ip-172-31-24-132 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
root@ip-172-31-24-132:/home/ubuntu#
```

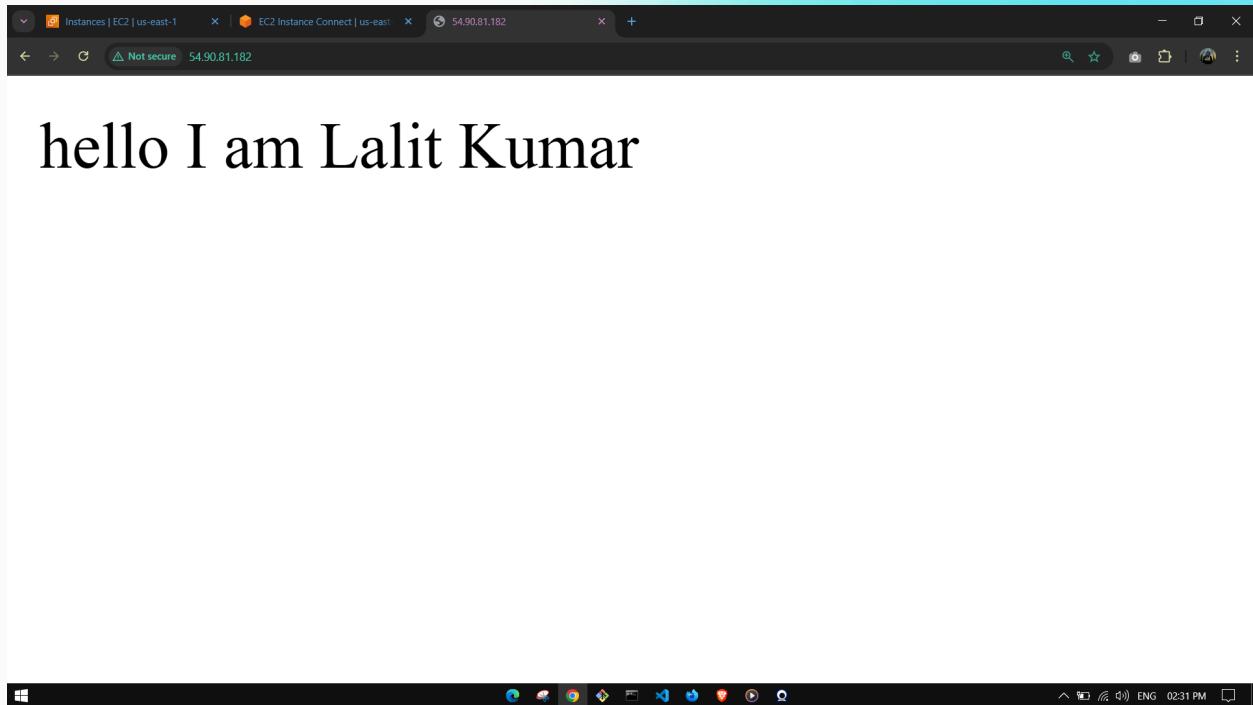
i-00a5e2bc2057003f8 (server)

PublicIPs: 54.90.81.182 PrivateIPs: 172.31.24.132

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now select your instance and copy the public ip address/public dns and paste it in your browser and see your web page is running or not.

Just check here...



# hello I am Lalit Kumar

Now go to cloudfont service and create a cloudfont template, give the name and for now select single website or app from the distribution type to make it simple.

A screenshot of the AWS CloudFront 'Create distribution' wizard. The top navigation bar shows 'Distributions | CloudFront | Global' and the URL 'us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/create'. The sidebar on the left lists steps: Step 1 (Get started, currently selected), Step 2 (Specify origin), Step 3 (Enable security), Step 4 (Get TLS certificate), Step 5 (Review and create). The main content area has a blue header bar with the message: 'We've streamlined the process of creating a CloudFront distribution. Continue here and let us know what you think. Or go to the previous Create Distribution page.' Below this, the 'Get started' section contains a paragraph about connecting websites, apps, files, video streams, and other content to CloudFront. The 'Distribution options' section includes a 'Distribution name' field containing 'first-cloud-front', a 'Description - optional' field, and a 'Distribution type' section with two options: 'Single website or app' (selected) and 'Multi-tenant architecture - New'. The bottom of the screen shows the standard AWS navigation bar with links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Go to your instance and copy your instance DNS

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main content area displays a table titled 'Instances (1/1) Info'. It shows one instance: 'server' (Instance ID: i-00a5e2bc2057003f8, State: Running, Type: t2.nano). Below the table, a detailed view for 'i-00a5e2bc2057003f8 (server)' is shown. The 'Details' tab is selected, displaying information such as Public IPv4 address (54.90.81.182), Instance state (Running), and Private IP DNS name (ip-172-31-24-132.ec2.internal). Other tabs include Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

You have different types of origin types for your cloud distribution, but now select another and paste your ec2 instance DNS as origin and go for the next step.

The screenshot shows the 'Create distribution' page in the AWS CloudFront console, specifically Step 2: 'Specify origin'. On the left, a sidebar lists steps: Get started (selected), Specify origin (current step), Enable security, and Review and create. The main content area is titled 'Specify origin' and 'Origin type'. It explains that CloudFront works with AWS-based origins and origins hosted on other cloud providers. There are five options: Amazon S3, Elastic Load Balancer, API Gateway, Elemental MediaPackage, VPC origin, and Other (selected). The 'Other' option is described as referring to any AWS or non-AWS origin through its publicly resolvable URL. Below this, the 'Origin' section shows a 'Custom origin' input field containing 'ec2-54-90-81-182.compute-1.amazonaws.com'.

## From the settings choose custom origin settings

The screenshot shows the 'Create distribution' step in the AWS CloudFront wizard. Under the 'Settings' tab, the 'Origin path - optional' section has '/path' entered. In the 'Origin settings' section, 'Use recommended origin settings' is selected. In the 'Cache settings' section, 'Use recommended cache settings tailored to serving Custom origin content' is selected. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

Select HTTP only and go for next step

The screenshot shows the 'Create distribution' step in the AWS CloudFront wizard. Under the 'Settings' tab, the 'Protocol' section has 'HTTP only' selected. Other options include 'HTTPS only' and 'Match viewer'. Below the protocol section are fields for 'HTTP port' (set to 80) and 'Connection attempts' (set to 3). At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

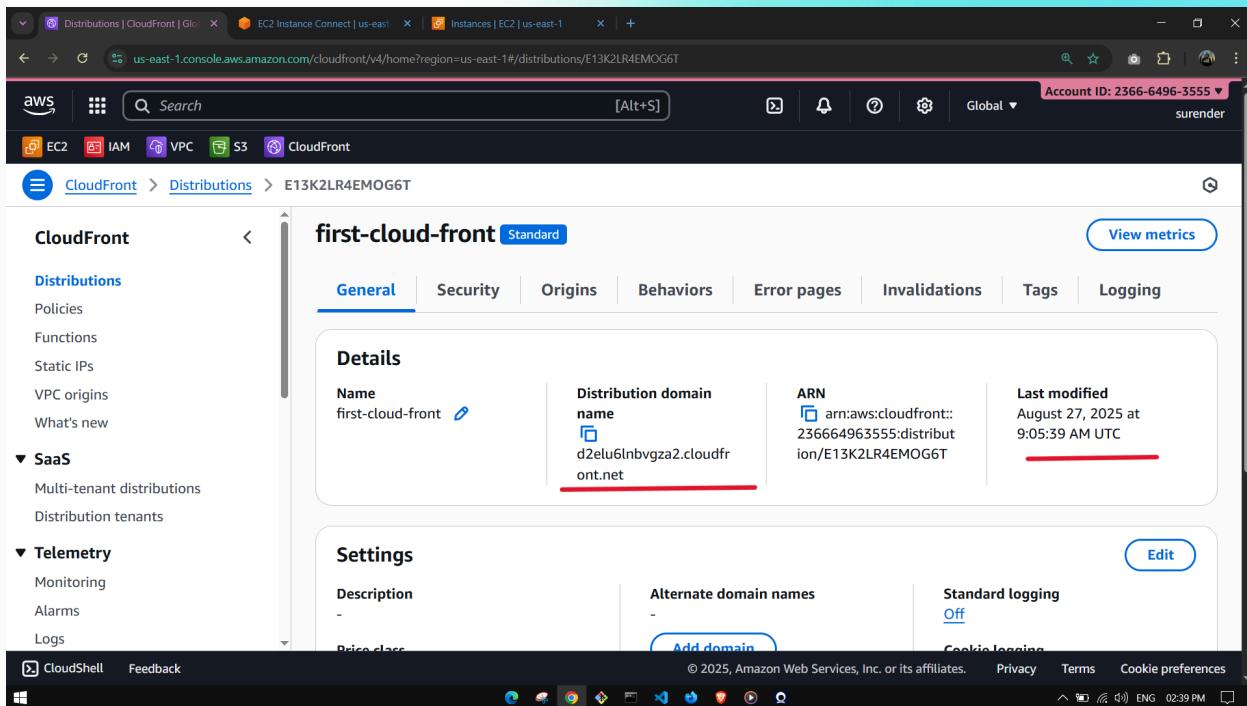
For now just disable the web application firewall (WAF) and go for the next step and create a cloud formation template.

The screenshot shows the AWS CloudFront 'Create distribution' wizard at the 'Enable security' step. On the left, a vertical navigation bar lists four steps: Step 1 (Get started), Step 2 (Specify origin), Step 3 (Enable security), and Step 4 (Review and create). Step 3 is highlighted. The main area is titled 'Enable security' and contains a section for 'Web Application Firewall (WAF)'. It includes two options: 'Enable security protections' (selected) and 'Do not enable security protections'. A note below the first option states: 'Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.' At the bottom right are 'Cancel', 'Previous', and 'Next' buttons. The top of the screen shows the AWS navigation bar and the URL 'us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/create'.

Wait for the deployment it will take some time to deploy your application id different aws edge location so it may take time.

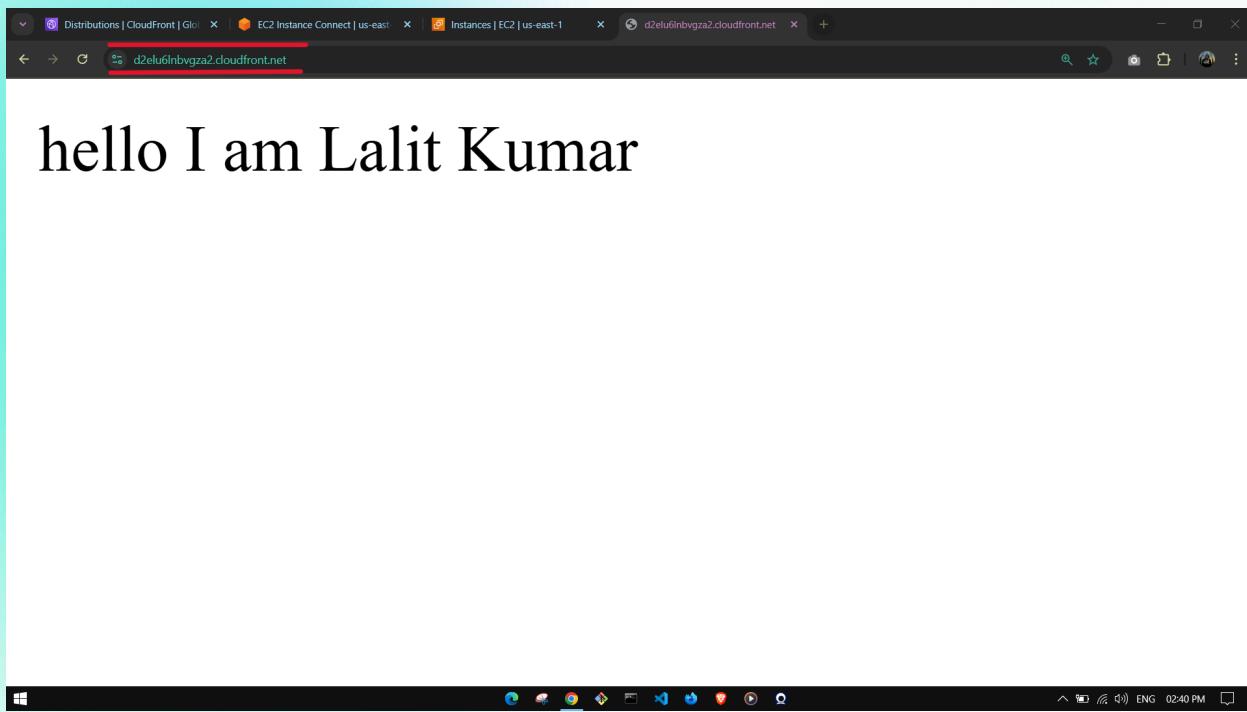
The screenshot shows the AWS CloudFront distribution details page for 'first-cloud-front'. The left sidebar has sections for 'Distributions' (Policies, Functions, Static IPs, VPC origins, What's new), 'SaaS' (Multi-tenant distributions, Distribution tenants), and 'Telemetry' (Monitoring, Alarms, Logs). The main content area shows the distribution 'first-cloud-front' under the 'Standard' configuration. It has tabs for General, Security, Origins, Behaviors, Error pages, Invalidations, Tags, and Logging. The General tab is selected. Under 'Details', the Name is 'first-cloud-front' and the Distribution domain name is 'd2elu6lnvgza2.cloudfront.net'. The ARN is 'arn:aws:cloudfront::236664963555:distribution/E13K2LR4EMOG6T'. The 'Last modified' status is 'Deploying'. Under 'Settings', there are fields for Description (empty), Alternate domain names (with an 'Add domain' button), Standard logging (Off), and Cookie logging. The bottom of the screen shows the AWS navigation bar and the URL 'us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/E13K2LR4EMOG6T'.

After the deployment copy the DNS and paste it in your browser



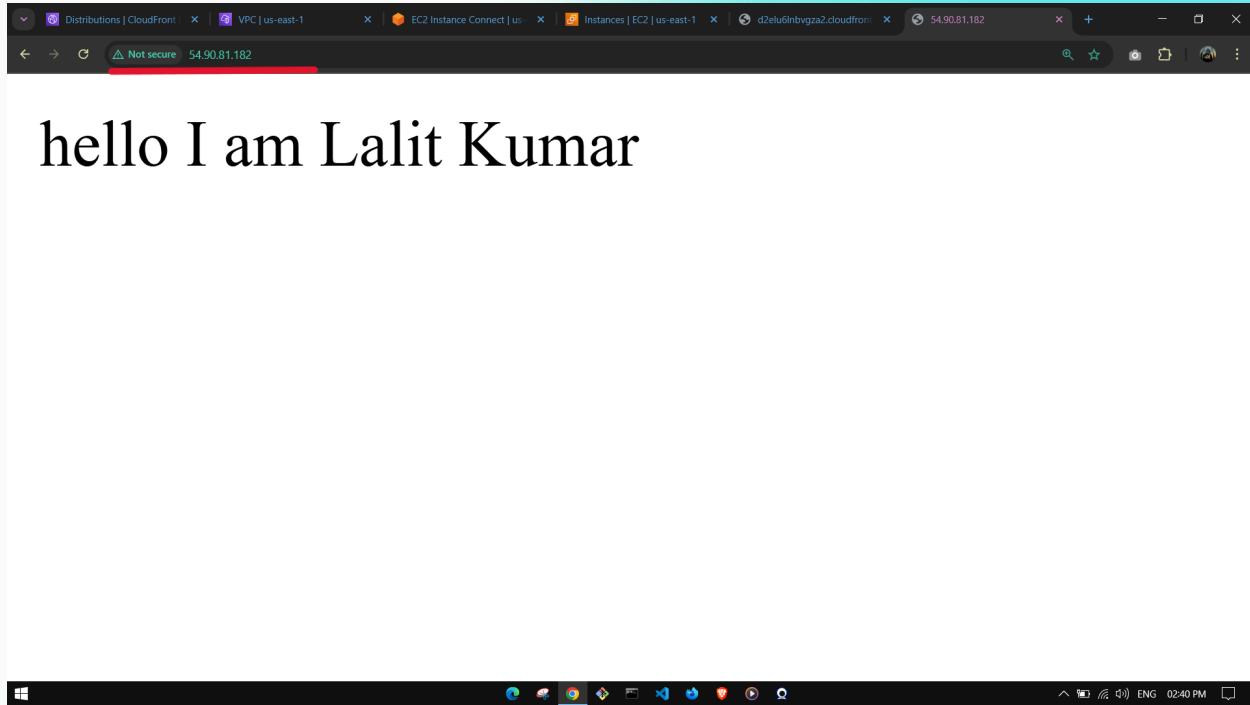
The screenshot shows the AWS CloudFront console with the distribution 'first-cloud-front' selected. The 'General' tab is active. In the 'Details' section, the 'Distribution domain name' is listed as 'd2elu6lnbvgza2.cloudfront.net'. The ARN is shown as 'arn:aws:cloudfront::236664963555:distribution/E13K2LR4EMOG6T'. The 'Last modified' timestamp is 'August 27, 2025 at 9:05:39 AM UTC'. The 'Settings' section shows 'Standard logging' is turned 'Off'. The left sidebar includes links for Policies, Functions, Static IPs, VPC origins, and SaaS.

Here you can see your web-application is accessible through cloudfront you created.



The screenshot shows a web browser window with the URL 'd2elu6lnbvgza2.cloudfront.net' in the address bar. The page content is 'hello I am Lalit Kumar'. The browser interface includes a toolbar with icons for back, forward, search, and refresh, and a status bar at the bottom indicating 'ENG 02:40 PM'.

But there is a problem: it is also accessible through your public ip address of your server instance. It is not a good practice to be accessible through the server ip address so let's change it and make it secure.



Now go to vpc service and copy the prefix-id of the cloudfront origin facing. It stores all the ip addresses of the edge locations of the cloudfront and aws manages itself.

The screenshot shows the AWS VPC Managed Prefix Lists console. On the left, a navigation pane includes options like Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, **Managed prefix lists** (which is selected and highlighted with a red box), NAT gateways, Peering connections, and Route servers. The main area displays a table titled "Managed prefix lists (1/1) Info". The table has columns for Prefix list ID, Prefix list name, Max entries, and Address family. One row is selected, showing "pl-3b927c52" and "com.amazonaws.global.cloudfront.origin-facing" under the "Prefix list name" column. The "Address family" column shows "IPv4" for all rows. Below the table, a modal window titled "pl-3b927c52 - com.amazonaws.global.cloudfront.origin-facing" shows the "Details" section with fields for Prefix list name, Prefix list ID, Version, Max entries, State, State message, and Owner ID.

Go to the server security group and configure some settings in the inbound rule.

The screenshot shows the AWS EC2 Instances console. On the left, a navigation pane includes EC2, Instances (selected and highlighted with a red box), Images, and Elastic Block Store. The main area displays a table titled "Instances (1/1) Info". The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One instance, "server" (i-00a5e2bc2057003f8), is selected and highlighted with a red box. The "Instance state" column shows "Running" and the "Availability Zone" column shows "us-east-1b". Below the table, a modal window titled "i-00a5e2bc2057003f8 (server)" shows the "Security" tab selected. Other tabs include Details, Status and alarms, Monitoring, Security (selected), Networking, Storage, and Tags. Under the Security tab, sections for IAM Role and Security groups are visible. The IAM Role section shows "Owner ID" and "236664963555". The Security groups section shows "sg-00a91d343f45bf245 (launch-wizard-3)".

Click on edit inbound rules.

**sg-00a91d343f45bf245 - launch-wizard-3**

**Inbound rules (2)**

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-05cb06b65d590212b	IPv4	HTTP	TCP	80
-	sgr-008dcab870e54f6f1	IPv4	SSH	TCP	22

Remove your old http rule now add new and paste cloudfront prefix-id which you copied earlier in the source of http rule.

**Inbound rules**

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-02e919417f7cfdde9	HTTP	TCP	80	Cust... <input type="text" value="pl-3b927c52"/>	pl-3b927c52
-	SSH	TCP	22	Any... <input type="text" value="0.0.0.0/0"/>	0.0.0.0/0

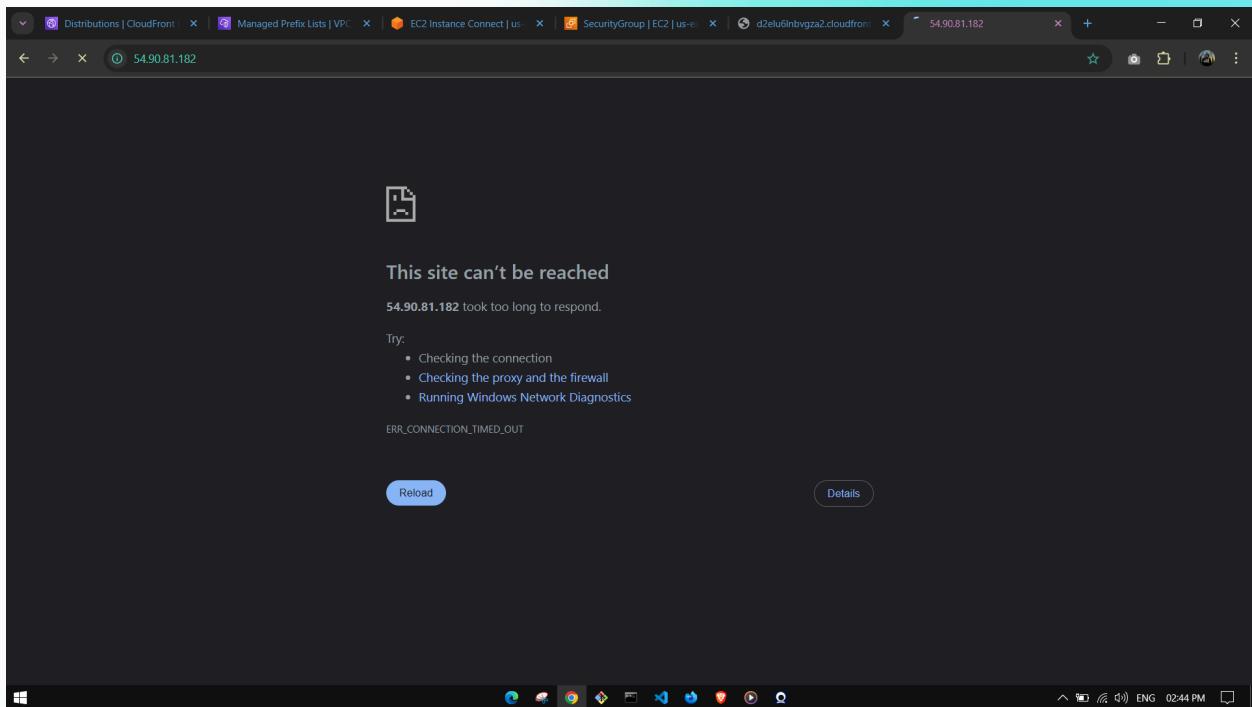
**Info**

**Cancel** **Preview changes** **Save rules**

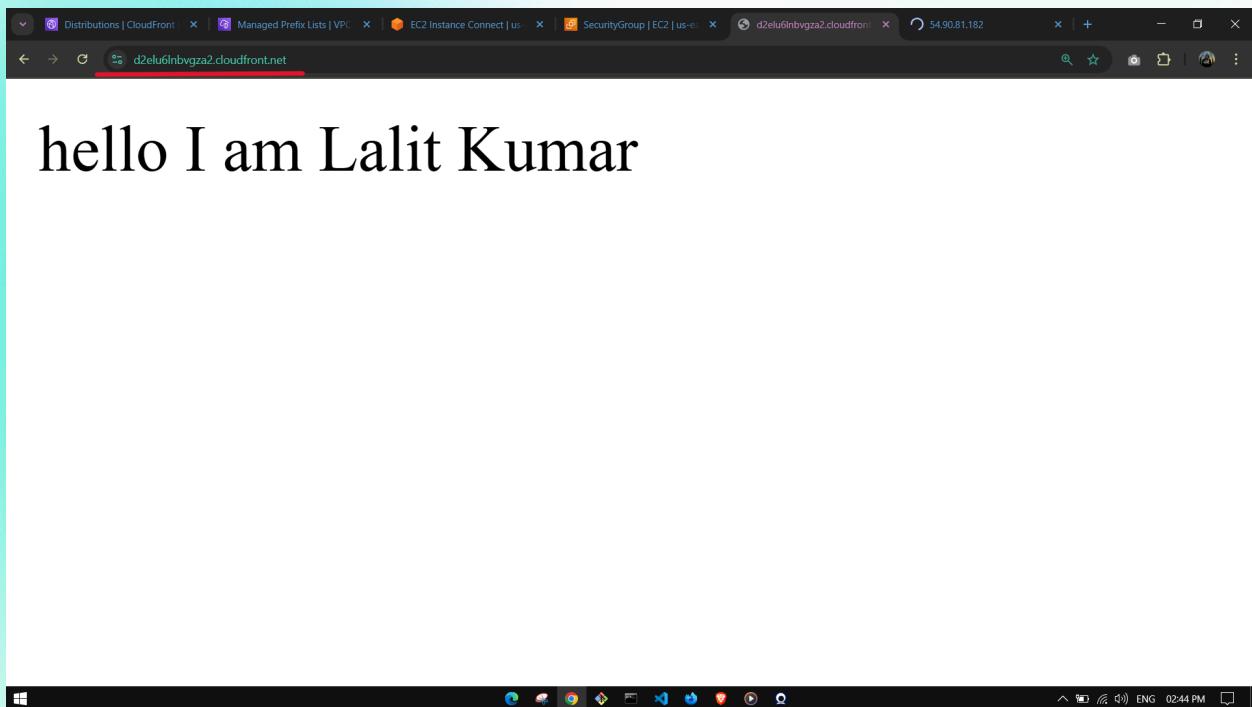
**Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

**When you reference a prefix list in a security group rule, the maximum number of entries for the prefix lists counts against the quota for the number of entries for the security group. For example, if you create a prefix list with 20 maximum entries and you reference that prefix list in a security group rule, this counts as 20 security group rules.**

Now try to access your web-application through the ip address of your server instance, probably you will not be able to access it.



But it is accessible through your cloudfront



Now go to s3 service and create a s3 bucket for multiple origin in cloudfront

The screenshot shows the 'Create bucket' page in the AWS S3 console. Under 'General configuration', the 'Bucket type' section has 'General purpose' selected (radio button is checked). The 'Bucket name' field contains 'my-testing-bucket'. Below the bucket name, a note states: 'Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More'.

**General configuration**

**AWS Region**  
US East (N. Virginia) us-east-1

**Bucket type** [Info](#)

**General purpose**  
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

**Directory**  
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

**Bucket name** [Info](#)  
my-testing-bucket

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

**Copy settings from existing bucket - optional**  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

Format: s3://bucket/prefix

The screenshot shows the 'Create bucket' page in the AWS S3 console, continuing from the previous step. Under 'Advanced settings', there is a note: 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.'

**Encryption type** [Info](#)  
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

**Server-side encryption with Amazon S3 managed keys (SSE-S3)**

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

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

Disable

Enable

**Advanced settings**

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

[Cancel](#) [Create bucket](#)

After creating the bucket just create a folder inside your bucket

Create a folder with the name images as shown in the below image

The screenshot shows the AWS S3 console interface. The user is navigating through the 'Amazon S3 > Buckets > my-testing-buckettt > Create folder' path. A modal window titled 'Folder' is open, with the 'Folder name' field containing 'images'. Below the field, a note states 'Folder names can't contain "/". See rules for naming.' Under the 'Server-side encryption' section, it says 'Server-side encryption protects data at rest.' and 'The following encryption settings apply only to the folder object and not to sub-folder objects.' There are two options: 'Don't specify an encryption key' (selected) and 'Specify an encryption key'. A warning message at the bottom states: '⚠ If your bucket policy requires objects to be encrypted with a specific encryption key, you must specify the same encryption key when you create a folder. Otherwise, folder creation will fail.' At the top right of the modal, there are 'Cancel' and 'Create folder' buttons.

After creating the folder inside the bucket upload an image to the images folder of s3 bucket

The screenshot shows the AWS S3 console interface. The user is navigating through the 'Amazon S3 > Buckets > my-testing-buckettt > Upload' path. A modal window titled 'Upload objects - S3 bucket my...' is open. In the 'Files and folders' section, there is one item: 'Screenshot (151...) image/png 434.7 KB'. Below this, the 'Destination' section shows the destination as 's3://my-testing-buckettt'. The 'Destination details' section notes that 'Bucket settings that impact new objects stored in the specified destination.' The 'Permissions' and 'Properties' sections are also visible. At the top right of the modal, there are 'Cancel' and 'Upload' buttons.

Go to the origin tab and see you have only one origin which is your ec2 instance (server).

The screenshot shows the AWS CloudFront Origins page for a distribution named 'first-cloud-front'. The 'Origins' tab is selected. There is one origin listed: 'ec2-54-90-81-182.cor' with 'Origin type' set to 'EC2'. Below the origins section is an 'Origin groups' section which is currently empty.

Create another origin for your s3 bucket and paste DNS of your bucket here

The screenshot shows the 'Create origin' page for the same distribution. In the 'Settings' section, the 'Origin domain' field contains 'my-testing-buckettt.s3.us-east-1.amazonaws.com'. The 'Origin path - optional' field is empty. In the 'Name' section, the name 'my-testing-buckettt.s3.us-east-1.amazonaws.com' is entered. Under 'Origin access', the 'Public' option is selected. At the bottom, there are 'Save' and 'Cancel' buttons.

You don't have need to make your bucket public, select the setting which are highlighted in the below image and create **origin access control** to allow access the s3 objects.

The screenshot shows the AWS CloudFront 'Create origin' configuration page. In the 'Origin access' section, the 'Origin access control settings (recommended)' option is selected. A red box highlights this selection. Below it, a dropdown menu for 'Select an origin access control' is open, showing 'This field cannot be empty'. A red box highlights this error message.

Just create an OAC or you can use what you created before.

The screenshot shows the 'Create new OAC' dialog box. The 'Name' field contains 'my-testing-buckettt.s3.us-east-1.amazonaws.com'. The 'Signing behavior' section has 'Sign requests (recommended)' selected. A red box highlights this selection. The 'Origin type' dropdown is set to 'S3'. A red box highlights this selection. The 'Create' button is visible at the bottom right.

Select your OAC which you created right now and copy the policy to be paste in the s3 bucket policy later.

The screenshot shows the AWS CloudFront 'Create origin' configuration page. On the left, there's a sidebar with navigation links like 'Distributions', 'Policies', 'Functions', etc. The main area has a 'Name' input field containing 'my-testing-buckettt.s3.us-east-1.amazonaws.com'. Under 'Origin access', the 'Origin access control settings (recommended)' option is selected. In the 'Origin access control' section, a dropdown menu shows 'my-testing-buckettt.s3.us-east-1.amazonaws.com' and a 'Create new OAC' button. A callout box highlights the 'Copy policy' button next to a note about CloudFront policy statements. At the bottom, there are 'Go to S3 bucket permissions' and 'CloudShell' buttons.

Paste the policy which you copy earlier to the s3 bucket policy.

The screenshot shows the AWS S3 'Edit bucket policy' configuration page. The left sidebar shows 'Amazon S3 > Buckets > my-testing-buckettt > Edit bucket policy'. The main area is titled 'Policy' and contains a JSON policy document. The policy document includes statements for allowing CloudFront access to the bucket. To the right, there's a 'Select a statement' dropdown and a '+ Add new statement' button.

```
1 {  
2     "Version": "2008-10-17",  
3     "Id": "PolicyForCloudFrontPrivateContent",  
4     "Statement": [  
5         {  
6             "Sid": "AllowCloudFrontServicePrincipal",  
7             "Effect": "Allow",  
8             "Principal": {  
9                 "Service": "cloudfront.amazonaws.com"  
10            },  
11            "Action": "s3:GetObject",  
12            "Resource": "arn:aws:s3:::my-testing-buckettt/*",  
13            "Condition": {  
14                "StringEquals": {  
15                    "AWS:SourceArn": "arn:aws:cloudfront::236664963555:distribution/E13K2LR4EMOG6T"  
16                }  
17            }  
18        }  
19    ]  
20}
```

Go to the behaviour tab of your cloudfront and create another behaviour for your application or you want to redirect your traffic for a specific path.

The screenshot shows the AWS CloudFront Behaviors page for a distribution named 'first-cloud-front'. The 'Behaviors' tab is selected. There is one behavior listed: 'Default (\*)' with the path pattern '/'. The behavior points to the origin 'ec2-54-90-81-...'. Other tabs include General, Security, Origins, Error pages, Invalidations, Tags, and Logging. A 'Create behavior' button is visible at the top right of the behaviors table.

Create a path and select origin as your ec2 instance and create a behaviour for this path.

The screenshot shows the 'Create behavior' page for a distribution named 'E13K2LR4EMOG6T'. The 'Path pattern' field contains '/api/'. The 'Origin and origin groups' dropdown is set to 'ec2-54-90-81-182.compute-1.amazonaws.com-metr1du40xp'. Under 'Viewer', the 'Viewer protocol policy' is set to 'HTTP and HTTPS'. The 'Allowed HTTP methods' section is visible at the bottom. The 'Settings' section includes options for compressing objects automatically ('Yes' is selected) and a 'Compress objects automatically' checkbox.

The screenshot shows the AWS CloudFront 'Create behavior' configuration page. At the top, there are tabs for 'CloudFront', 'Distributions', and 'E13K2LR4EMOG6T'. Below the tabs, there is a search bar and a navigation breadcrumb: 'CloudFront > Distributions > E13K2LR4EMOG6T > Create behavior'. A dropdown menu labeled 'Select response headers' is open, showing 'Create response headers policy'. Below this, a section titled 'Additional settings' is partially visible.

**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	

Buttons at the bottom right include 'Cancel' and 'Create behavior' (highlighted in orange).

Check the path of your cloudfront with `/api` and you will see your behaviour works successfully.

The screenshot shows a browser window displaying the URL `d2elu6lnbvqza2.cloudfront.net/api/index.html`. The page content is "Hello I am Lalit Kumar". The browser's address bar also shows other tabs related to AWS services like CloudFront, S3, and EC2.

Go to the behaviour tab and you will see your have two behaviours default and **/apt/\***

Precedence	Path pattern	Origin or origin group	Viewer protocol	Cache policy name	Origin request policy	Response header
0	/api/*	ec2-54-90-81-18...	HTTP and HTTPS	Managed-CachingDisabled	Managed-AllViewer	-
1	Default (*)	ec2-54-90-81-18...	Redirect HTTP to ...	UseOriginCacheControl	Managed-AllViewer	-

Create a path **/images/\*** for redirect to your bucket and select your bucket as your origin also choose the settings which are shown in the below image and create it.

Now you can see clearly there are three behaviours and let's test them.

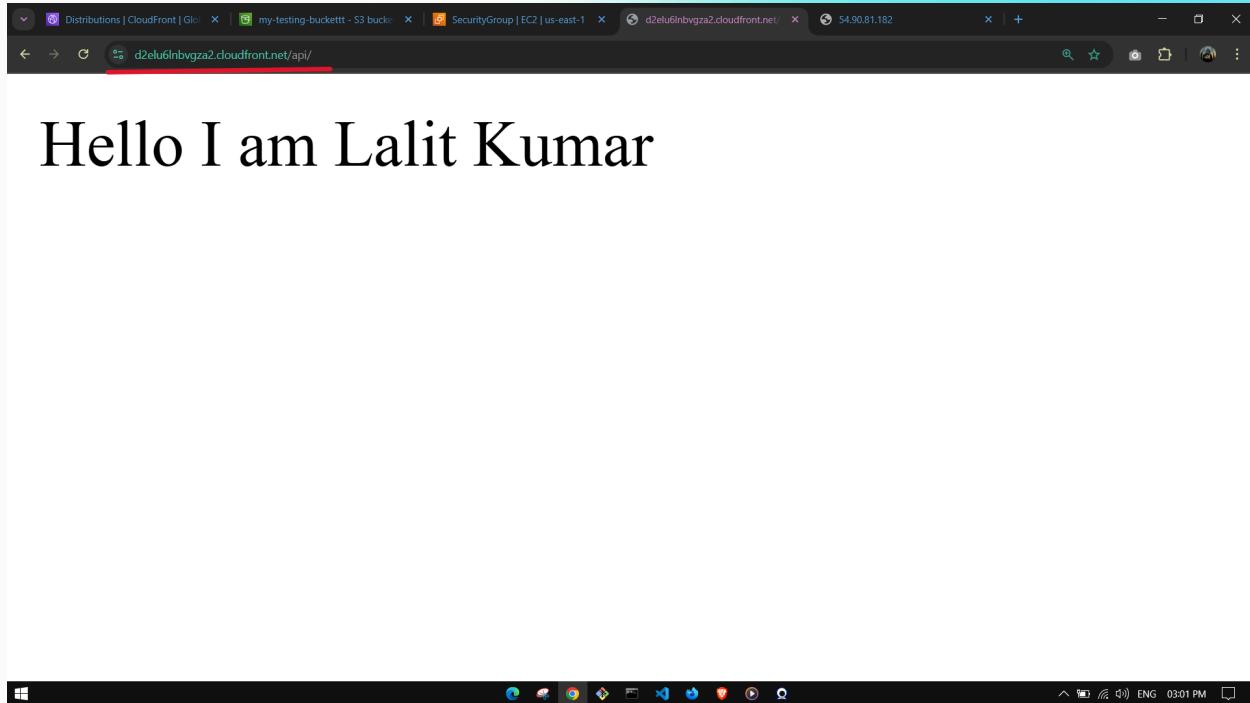
The screenshot shows the AWS CloudFront Behaviors configuration page. At the top, there are tabs for General, Security, Origins, Behaviors (which is selected), Error pages, Invalidations, Tags, and Logging. Below the tabs, there is a table titled "Behaviors (3)". The table has columns for Preced..., Path pattern, Origin or origin group, Viewer protocol policy, Cache policy name, Origin request policy na..., and Response headers policy.... The rows show the following configurations:

Preced...	Path pattern	Origin or origin group	Viewer protocol policy	Cache policy name	Origin request policy na...	Response headers policy...
0	/api/*	ec2-54-90-81-182.comp...	HTTP and HTTPS	Managed-CachingDisabled	Managed-AllViewer	-
1	/images/*	my-testing-buckettt.s3.u...	Redirect HTTP to HTTPS	Managed-CachingOptimized	-	-
2	Default (*)	ec2-54-90-81-182.comp...	Redirect HTTP to HTTPS	UseOriginCacheControlHeader	Managed-AllViewer	-

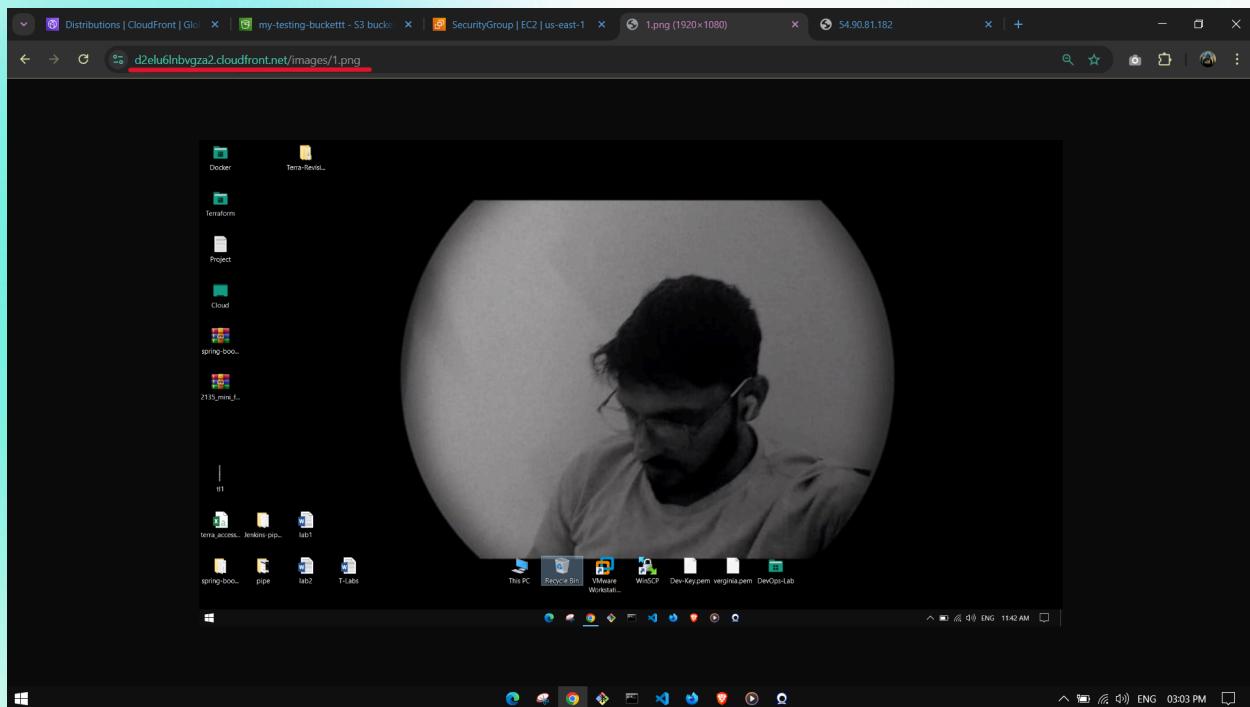
Default one

The screenshot shows a web browser window displaying the URL [d2elu6lnbgza2.cloudfront.net](https://d2elu6lnbgza2.cloudfront.net). The page content is the text "hello I am Lalit Kumar". The browser's address bar also shows the full URL and the CloudFront distribution ID.

Behaviour **/api/\***



Behaviour **/images/\*** you will have the images age accessible through the cloudfront even if the bucket is private. Even you can use bucket for other purposes like static website hosting, etc



## **Let's Connect :-**

**Lalit Kumar**

Follow: [www.linkedin.com/in/lalit192977](https://www.linkedin.com/in/lalit192977)

Profile: <https://bento.me/lalit192977>