

Integrating Cequence Bot Defense SaaS with Amazon CloudFront

Contents

| | |
|---|---|
| About Cequence Bot Defense SaaS and Amazon CloudFront..... | 2 |
| Step 1: Configure Application Availability | 3 |
| Step 2: Configure Bot Defense SaaS Origin | 4 |
| Step 3: Configure Traffic Forwarding to Bot Defense SaaS..... | 5 |

About Cequence Bot Defense SaaS and Amazon CloudFront

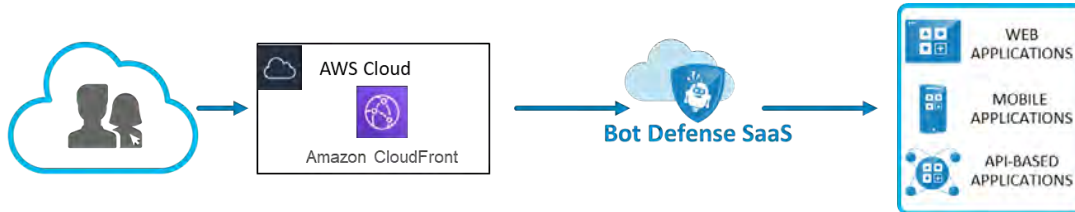
Bot Defense SaaS uses an ML-based approach to eliminate avenues of fraud caused by automated attacks targeted at your web, mobile and API-based applications deployed on AWS. Using a SaaS deployment model reduces the operation efforts associated with deploying Bot Defense to prevent account takeovers and API-based business logic abuse.

[Amazon CloudFront](#), the highly secure and programmable content delivery network (CDN) integrates with Bot Defense SaaS, allowing you to analyze your public-facing application transactions to determine malicious or benign intent. The findings are then used to enforce policy or exported via a REST-based API to an existing component of your security infrastructure.

Traffic flow without Bot Defense SaaS:



Traffic flow with Bot Defense SaaS:



The steps required to integrate Bot Defense SaaS with Amazon CloudFront are relatively straightforward. Selected traffic that terminates on Amazon CloudFront will be routed to Bot Defense SaaS for inspection before it is forwarded to the application origin.

Step 1: Configure Application Availability

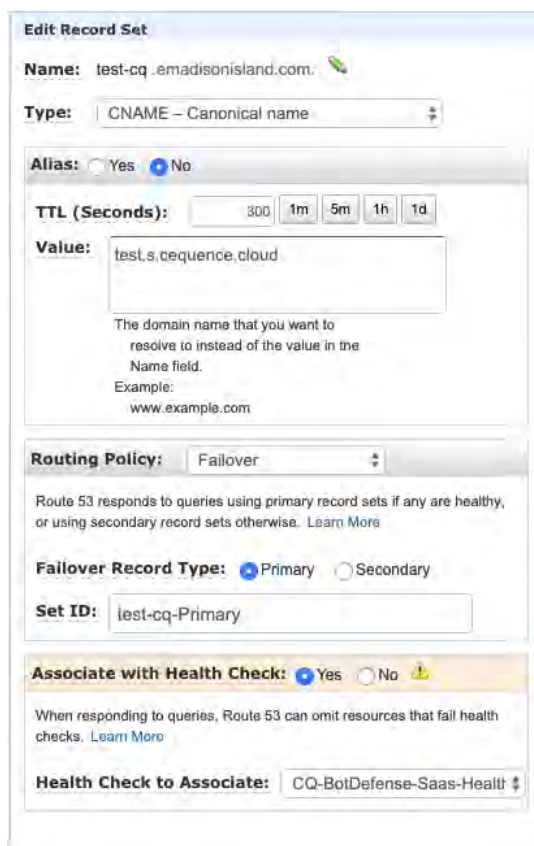
Application availability must be ensured with the addition of Bot Defense SaaS to the traffic flow between Amazon CloudFront and application origin. In the rare event where the Bot Defense SaaS becomes unavailable (determined via a health check), a fail-open must kick in and all application traffic from Amazon CloudFront must get routed directly to the application origin, bypassing Bot Defense SaaS completely. Such a fail-open scenario can be configured with a failover routing policy configuration.

To create a failover routing policy, either one of the below solutions can be leveraged:

- Amazon Route 53
- Bot Defense SaaS Traffic Manager (for customers that don't use Amazon Route 53)

The snapshots below show an Amazon Route 53 failover routing policy example, where the DNS Hostname test-cq.emadisonisland.com is pointing at two CNAME records:

1. Bot Defense SaaS origin: test.s.cequence.cloud (set as primary)
2. Sample application origin: origin-www.emadisonisland.com (set as secondary)



The screenshot shows the 'Edit Record Set' interface in the Amazon Route 53 console. The record name is 'test-cq.emadisonisland.com' and the type is 'CNAME - Canonical name'. The 'Alias' option is set to 'No'. The TTL is set to 300 seconds. The 'Value' field contains 'test.s.cequence.cloud'. Below the value field, there is a note: 'The domain name that you want to resolve to instead of the value in the Name field. Example: www.example.com'. The 'Routing Policy' is set to 'Failover'. A descriptive text states: 'Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. Learn More'. The 'Failover Record Type' is set to 'Primary'. The 'Set ID' is 'test-cq-Primary'. The 'Associate with Health Check' option is set to 'Yes'. A note below states: 'When responding to queries, Route 53 can omit resources that fail health checks. Learn More'. The 'Health Check to Associate' is set to 'CQ-BotDefense-SaaS-Health'.

Image 1: Bot Defense SaaS origin set as primary with associated health check

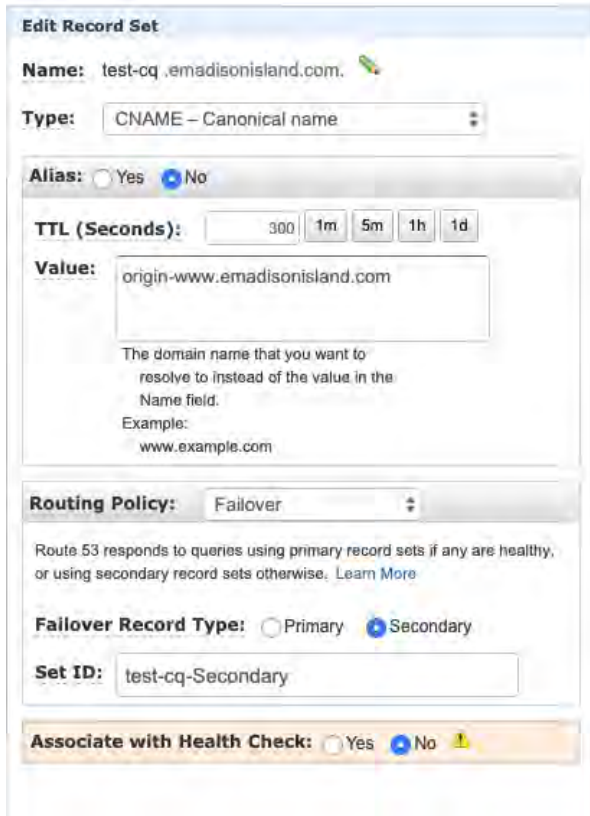


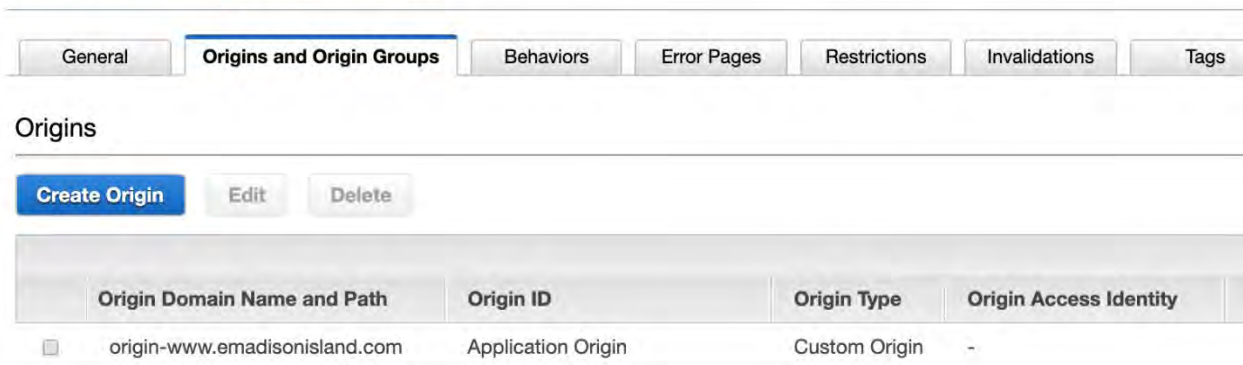
Image 2: Application origin set as secondary

The DNS hostname, test-cq.emadisonisland.com in this example, will be set as the origin hostname for forwarding traffic to Bot Defense SaaS on the Amazon CloudFront configuration.

Step 2: Configure Bot Defense SaaS Origin

In this step, configure the Bot Defense SaaS as a new origin:

[CloudFront Distributions](#) > E2PX47ABVW95AL



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

Origins

Create Origin Edit Delete

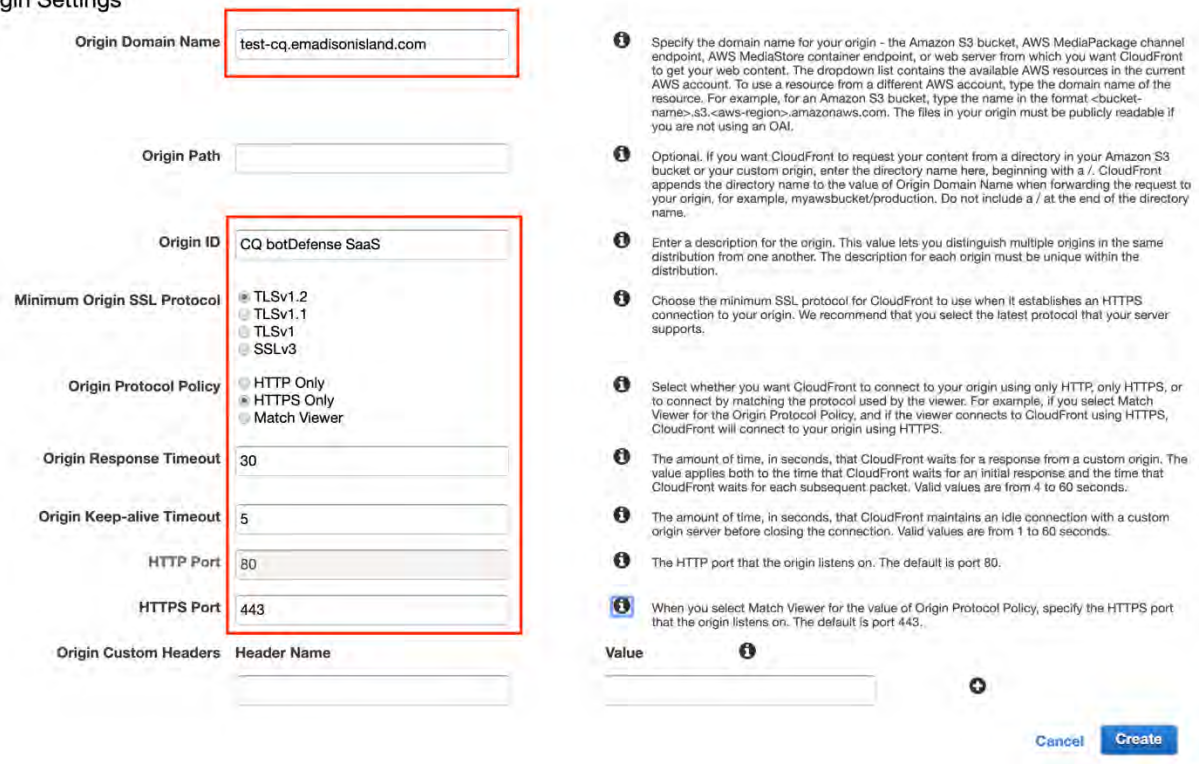
| Origin Domain Name and Path | Origin ID | Origin Type | Origin Access Identity |
|--|--------------------|---------------|------------------------|
| <input type="checkbox"/> origin-www.emadisonisland.com | Application Origin | Custom Origin | - |

Image 3: The existing application origin configured with Amazon CloudFront

- Go to the Origins and Origin Groups tab and click on Create Origin where you will create and define the new origin settings for Bot Defense SaaS. Change the origin settings configuration as shown in image 2 on the following page.
- In the example, the Origin Domain Name field is where the origin hostname for Bot Defense SaaS created in Step 1 will need to be entered
- All the other settings should be configured exactly as shown in the screenshot. Once complete, click Create.
- This will complete the creation of Bot Defense SaaS origin on the existing Amazon CloudFront distribution.

Create Origin

Origin Settings



Origin Domain Name test-cq.emadisonisland.com

Origin Path

Origin ID CQ botDefense SaaS

Minimum Origin SSL Protocol

- TLSv1.2
- TLSv1.1
- TLSv1
- SSLv3

Origin Protocol Policy

- HTTP Only
- HTTPS Only
- Match Viewer

Origin Response Timeout 30

Origin Keep-alive Timeout 5

HTTP Port 80

HTTPS Port 443

Origin Custom Headers

| Header Name | Value |
|-------------|-------|
| | |

Informational Text:

- Specify the domain name for your origin - the Amazon S3 bucket, AWS MediaPackage channel endpoint, AWS MediaStore container endpoint, or web server from which you want CloudFront to get your web content. The dropdown list contains the available AWS resources in the current AWS account. To use a resource from a different AWS account, type the domain name of the resource. For example, for an Amazon S3 bucket, type the name in the format <bucket-name>.s3.<aws-region>.amazonaws.com. The files in your origin must be publicly readable if you are not using an OAI.
- Optional. If you want CloudFront to request your content from a directory in your Amazon S3 bucket or your custom origin, enter the directory name here, beginning with a /. CloudFront appends the directory name to the value of Origin Domain Name when forwarding the request to your origin, for example, myawsbucket/production. Do not include a / at the end of the directory name.
- Enter a description for the origin. This value lets you distinguish multiple origins in the same distribution from one another. The description for each origin must be unique within the distribution.
- Choose the minimum SSL protocol for CloudFront to use when it establishes an HTTPS connection to your origin. We recommend that you select the latest protocol that your server supports.
- Select whether you want CloudFront to connect to your origin using only HTTP, only HTTPS, or to connect by matching the protocol used by the viewer. For example, if you select Match Viewer for the Origin Protocol Policy, and if the viewer connects to CloudFront using HTTPS, CloudFront will connect to your origin using HTTPS.
- The amount of time, in seconds, that CloudFront waits for a response from a custom origin. The value applies both to the time that CloudFront waits for an initial response and the time that CloudFront waits for each subsequent packet. Valid values are from 4 to 60 seconds.
- The amount of time, in seconds, that CloudFront maintains an idle connection with a custom origin server before closing the connection. Valid values are from 1 to 60 seconds.
- The HTTP port that the origin listens on. The default is port 80.
- When you select Match Viewer for the value of Origin Protocol Policy, specify the HTTPS port that the origin listens on. The default is port 443.

Buttons: Cancel, Create

Image 4: Behavior modification

Step 3: Configure Traffic Forwarding to Bot Defense SaaS

To configure forwarding of all application traffic to Bot Defense SaaS origin as shown in the previous step, we will need to make Bot Defense SaaS the default origin.

- Go to the **Behaviors** tab and select the Origin that has the **Path Pattern** of **Default (*)**
- In the example screenshot - Behaviors: screenshot (a), the existing customer application origin **Application Origin** is set with the **Path Pattern** of **Default (*)**

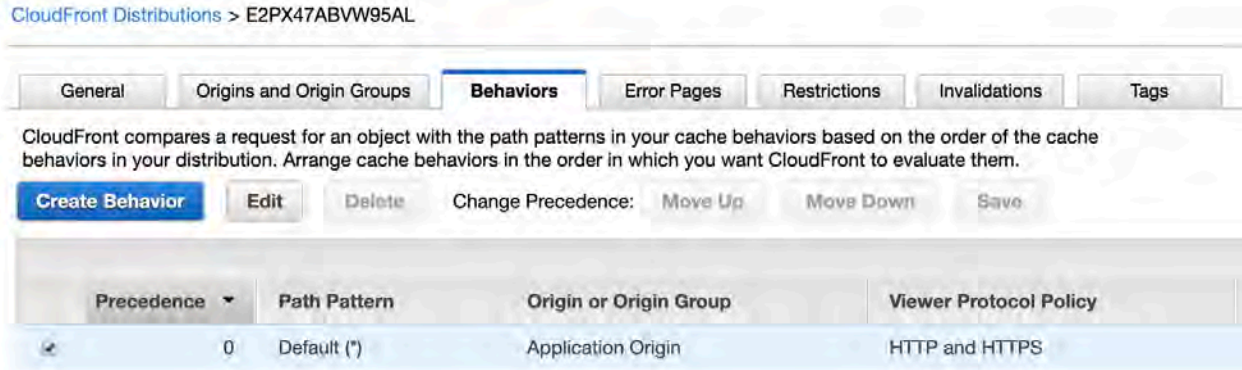


Image 5: Modifying CloudFront behavior (a)

- Click **Edit** to update the default behavior settings
- Select **Bot Defense SaaS** in the **Origin or Origin Group**, as shown in image 7.
- Click **Yes, Edit**
- As shown in Image 6, the **Behaviors** tab will now be updated and reflect **Bot Defense SaaS** as the Origin with **Path Pattern** of **Default (*)**.

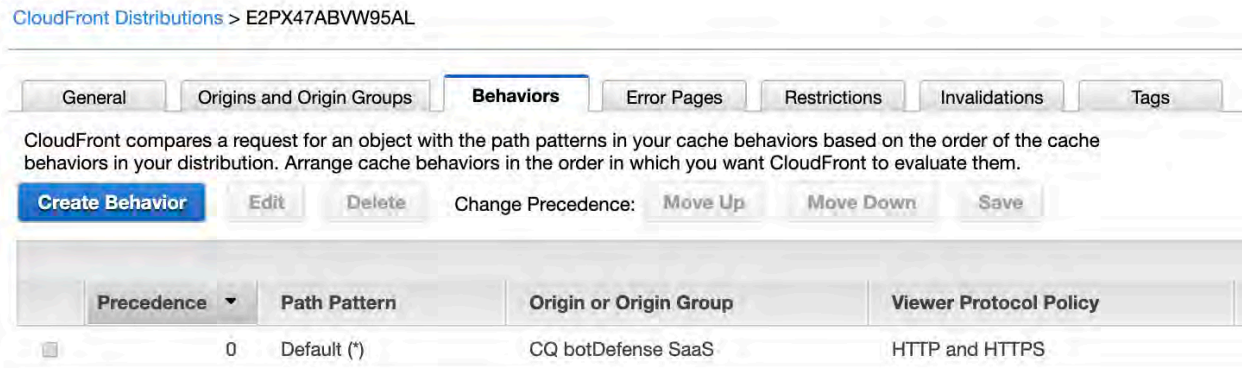


Image 6: Modifying CloudFront behavior (b)

Edit Behavior

Default Cache Behavior Settings

| | | | | | | | | |
|--|--|--------------------------|---------------------|--------------|-------------------|--|--------------------------|--|
| Path Pattern | Default (*) | i | | | | | | |
| Origin or Origin Group | Application Origin | i | | | | | | |
| Viewer Protocol Policy | <div style="border: 1px solid black; padding: 2px;"> CQ botDefense SaaS Application Origin </div> | i | | | | | | |
| Allowed HTTP Methods | <input checked="" type="radio"/> GET, HEAD <input type="radio"/> GET, HEAD, OPTIONS <input type="radio"/> GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE | i | | | | | | |
| Field-level Encryption Config | | i | | | | | | |
| Cached HTTP Methods | GET, HEAD (Cached by default) | i | | | | | | |
| Cache Based on Selected Request Headers | None (Improves Caching) Learn More | i | | | | | | |
| Object Caching | <input checked="" type="radio"/> Use Origin Cache Headers <input type="radio"/> Customize Learn More | i | | | | | | |
| Minimum TTL | 0 | i | | | | | | |
| Maximum TTL | 31536000 | i | | | | | | |
| Default TTL | 86400 | i | | | | | | |
| Forward Cookies | None (Improves Caching) | i | | | | | | |
| Query String Forwarding and Caching | None (Improves Caching) | i | | | | | | |
| Smooth Streaming | <input type="radio"/> Yes <input checked="" type="radio"/> No | i | | | | | | |
| Restrict Viewer Access (Use Signed URLs or Signed Cookies) | <input type="radio"/> Yes <input checked="" type="radio"/> No | i | | | | | | |
| Compress Objects Automatically | <input type="radio"/> Yes <input checked="" type="radio"/> No Learn More | i | | | | | | |
| Lambda Function Associations | | i | | | | | | |
| | <table border="0"> <tr> <td>CloudFront Event</td> <td>Lambda Function ARN</td> <td>Include Body</td> </tr> <tr> <td>Select Event Type</td> <td></td> <td><input type="checkbox"/></td> </tr> </table> Learn More | CloudFront Event | Lambda Function ARN | Include Body | Select Event Type | | <input type="checkbox"/> | |
| CloudFront Event | Lambda Function ARN | Include Body | | | | | | |
| Select Event Type | | <input type="checkbox"/> | | | | | | |

[Cancel](#) [Yes, Edit](#)

Image 7: Modifying default cache behavior