Integrating Cequence Bot Defense SaaS with Fastly

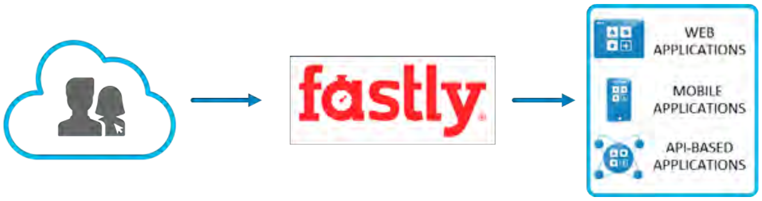
Contents

About Cequence Bot Defense SaaS and Fastly

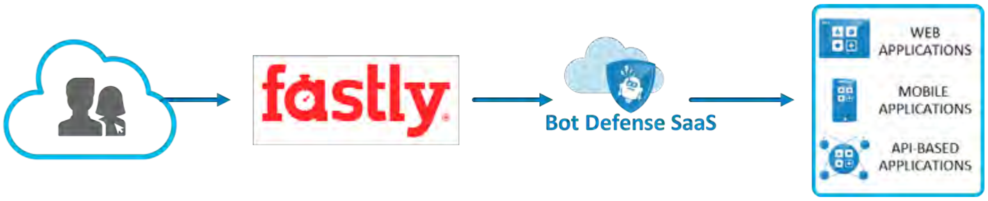
Bot Defense SaaS uses an agentless, ML-based approach to eliminate avenues of fraud caused by account takeovers and API business logic abuse.

When integrated with Fastly, traffic is directed to Bot Defense SaaS where it is analyzed by the CQAI ML-based automation indicators to determine malicious or benign intent. CQAI findings are then used to enforce policy or exported via a REST-based API to an existing component of your security infrastructure.

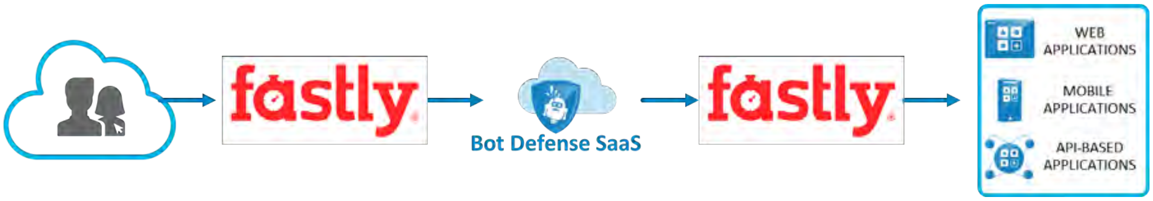
Traffic flow without Bot Defense SaaS:



Traffic flow with Bot Defense SaaS (option 1):



Traffic flow with Bot Defense SaaS in a loopback architecture (option 2):

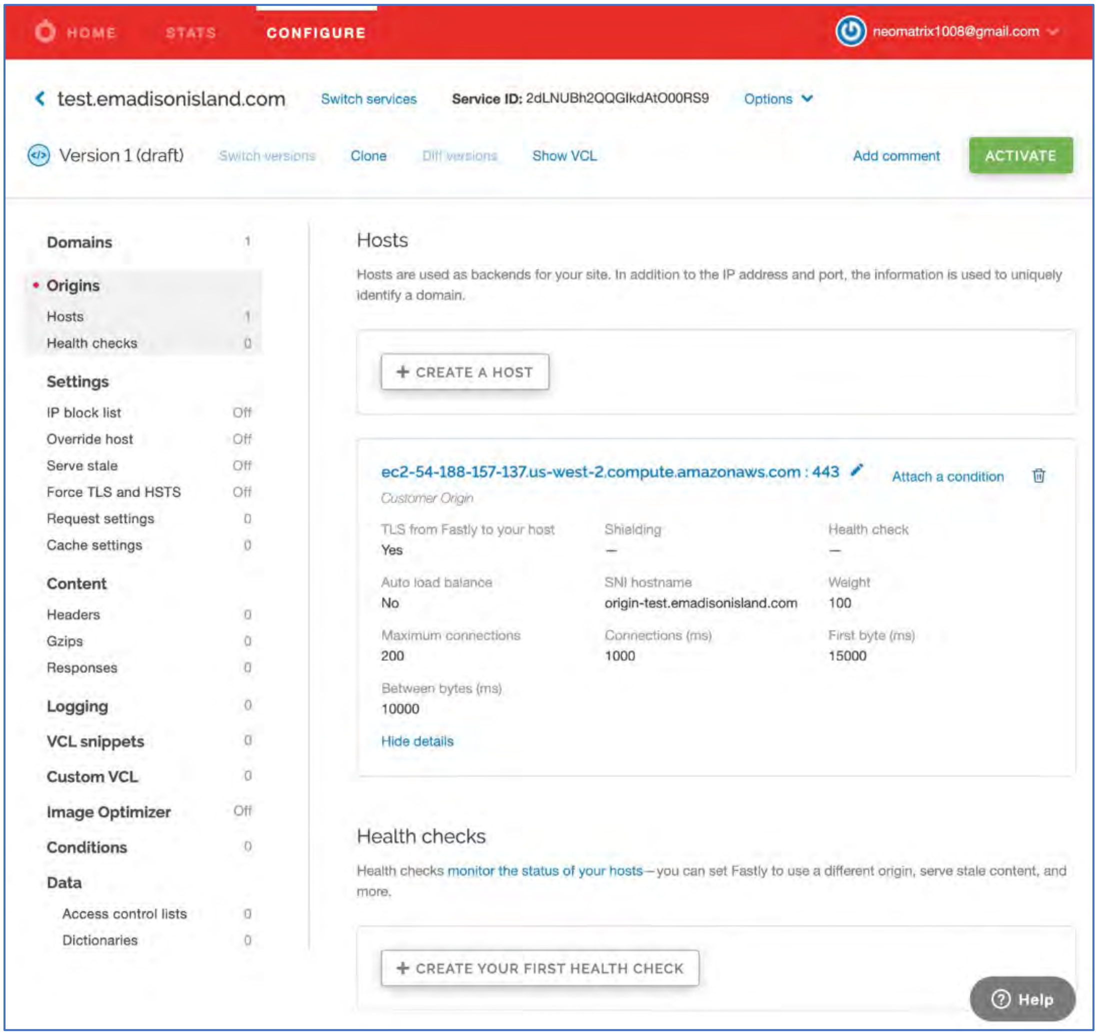


The steps required to integrate Bot Defense SaaS with Fastly are relatively straightforward. All traffic that terminates on Fastly will be routed to Bot Defense SaaS first for inspection and then forwarded to the application origin (option 1) or forwarded back to Fastly from where it will be routed to the application origin (option 2).

Step 1: Configure Bot Defense SaaS Origin

The configuration of Bot Defense SaaS origin and forwarding traffic to it will be explained using an example scenario where:

- Web Application: test.emadisonisland.com
- Application Origin: ec2-54-188-157-137.us-west-2.compute.amazonaws.com
- Bot Defense SaaS Origin: test.s.cequence.cloud



test.emadisonisland.com

Service ID: 2dLNUBh2QQGIkdAtO00RS9

Version 1 (draft)

Hosts

Hosts are used as backends for your site. In addition to the IP address and port, the information is used to uniquely identify a domain.

ec2-54-188-157-137.us-west-2.compute.amazonaws.com : 443

Customer Origin

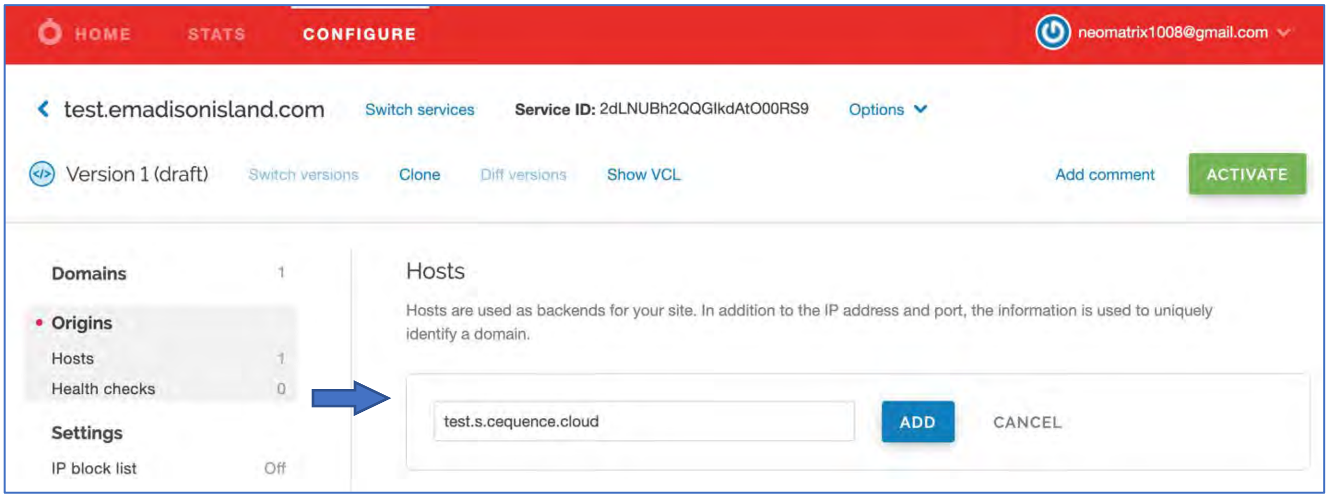
TLS from Fastly to your host	Shielding	Health check
Yes	—	—
Auto load balance	SNI hostname	Weight
No	origin-test.emadisonisland.com	100
Maximum connections	Connections (ms)	First byte (ms)
200	1000	15000
Between bytes (ms)		
10000		

Health checks

Health checks monitor the status of your hosts—you can set Fastly to use a different origin, serve stale content, and more.

Image 1: Customer application origin set as the only host

Navigate to the **Origins > Hosts** section to add the Bot Defense SaaS Origin on an existing Fastly configuration.



test.emadisonisland.com

Service ID: 2dLNUBh2QQGIkdAtO00RS9

Version 1 (draft)

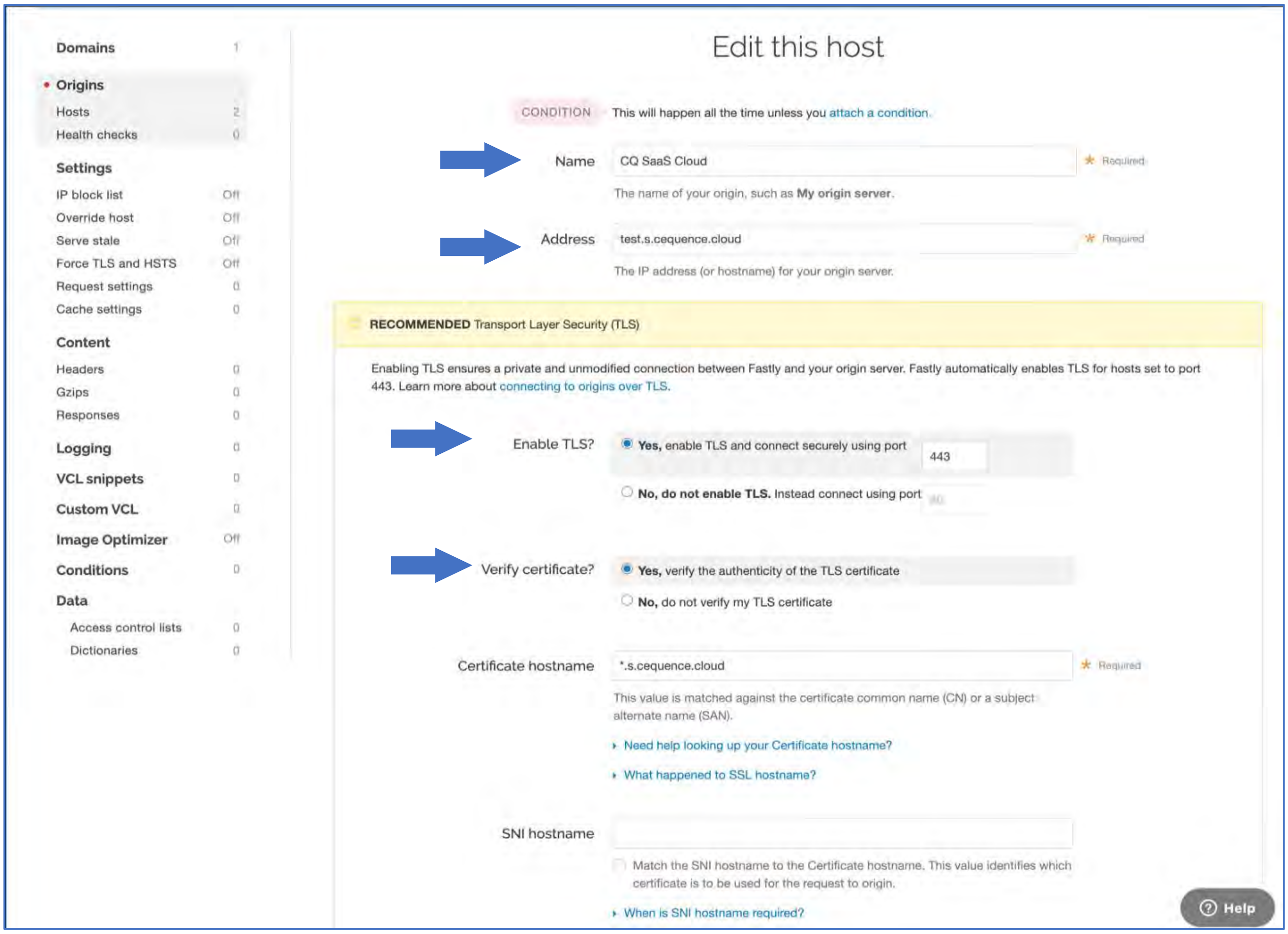
Hosts

Hosts are used as backends for your site. In addition to the IP address and port, the information is used to uniquely identify a domain.

test.s.cequence.cloud

Image 2: Add Cequence provided Bot Defense SaaS origin hostname

Fill out the Host Details as shown below and leave the other options as defaults:



Edit this host

CONDITION This will happen all the time unless you attach a condition.

Name: CQ SaaS Cloud ★ Required

The name of your origin, such as **My origin server**.

Address: test.s.cequence.cloud ★ Required

The IP address (or hostname) for your origin server.

RECOMMENDED Transport Layer Security (TLS)

Enabling TLS ensures a private and unmodified connection between Fastly and your origin server. Fastly automatically enables TLS for hosts set to port 443. Learn more about connecting to origins over TLS.

Enable TLS?
● **Yes,** enable TLS and connect securely using port 443
○ **No, do not enable TLS.** Instead connect using port 80

Verify certificate?
● **Yes,** verify the authenticity of the TLS certificate
○ **No,** do not verify my TLS certificate

Certificate hostname: *.s.cequence.cloud ★ Required

This value is matched against the certificate common name (CN) or a subject alternate name (SAN).

‣ Need help looking up your Certificate hostname?
‣ What happened to SSL hostname?

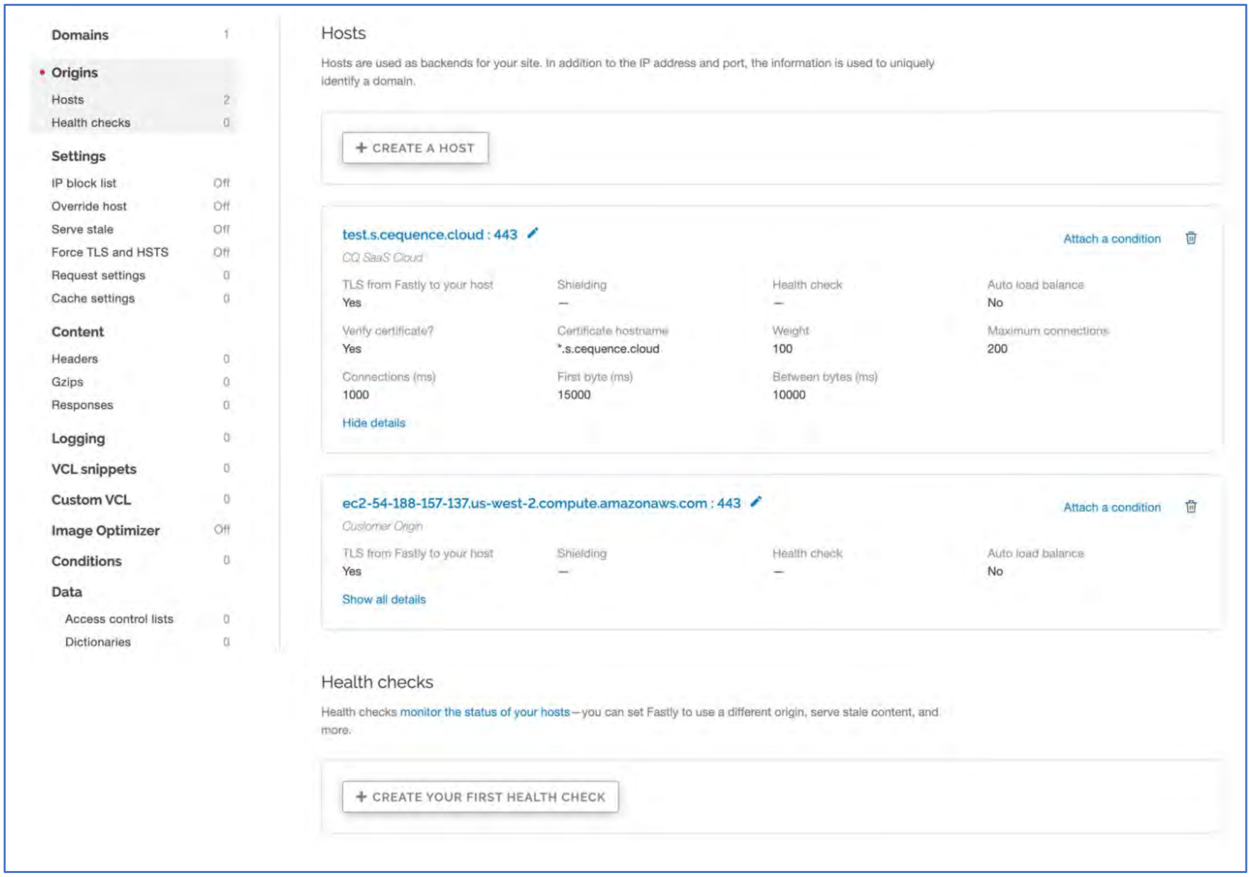
SNI hostname:

□ Match the SNI hostname to the Certificate hostname. This value identifies which certificate is to be used for the request to origin.

‣ When is SNI hostname required?

Image 3: Configure Bot Defense SaaS origin host

The hosts, both for the customer's Application Origin as well as for Bot Defense SaaS Origin will be shown as below:



Hosts

Hosts are used as backends for your site. In addition to the IP address and port, the information is used to uniquely identify a domain.

test.s.cequence.cloud : 443

CQ SaaS Cloud

TLS from Fastly to your host	Shielding	Health check	Auto load balance
Yes	—	—	No
Verify certificate?	**Certificate hostname**	**Weight**	**Maximum connections**
Yes	*.s.cequence.cloud	100	200
Connections (ms)	**First byte (ms)**	**Between bytes (ms)**	
1000	15000	10000	

ec2-54-188-157-137.us-west-2.compute.amazonaws.com : 443

Customer Origin

TLS from Fastly to your host	Shielding	Health check	Auto load balance
Yes	—	—	No

Health checks

Health checks monitor the status of your hosts—you can set Fastly to use a different origin, serve stale content, and more.

Image 4: Bot Defense SaaS origin and customer application origin hosts

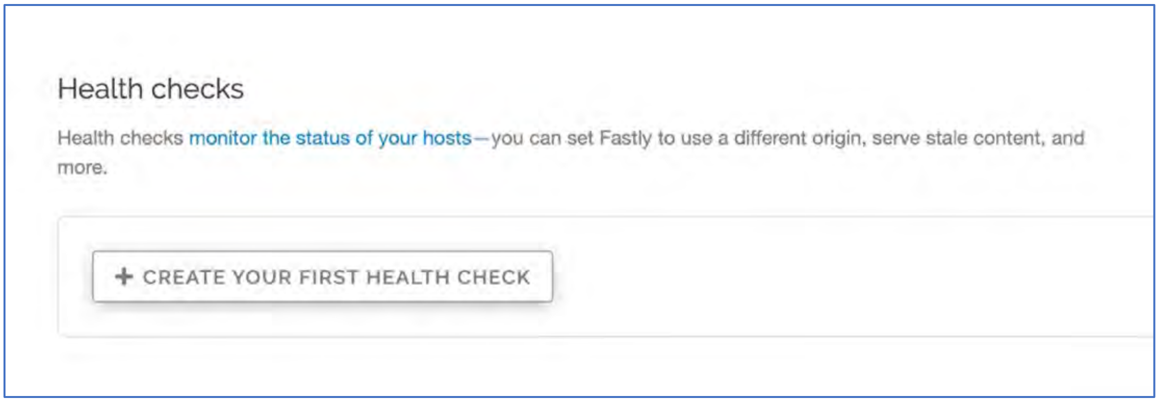
Step 2: Configure Application Availability

Application availability must be ensured with the addition of Bot Defense SaaS to the traffic flow between Fastly 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 Fastly must get routed directly to the Application Origin, bypassing Bot Defense SaaS completely.

Fastly offers the capabilities to set up a fail-open configuration using health checks.

To create the health check for the Bot Defense SaaS origin, navigate to Origins > Hosts and create a health check under the Health checks section.

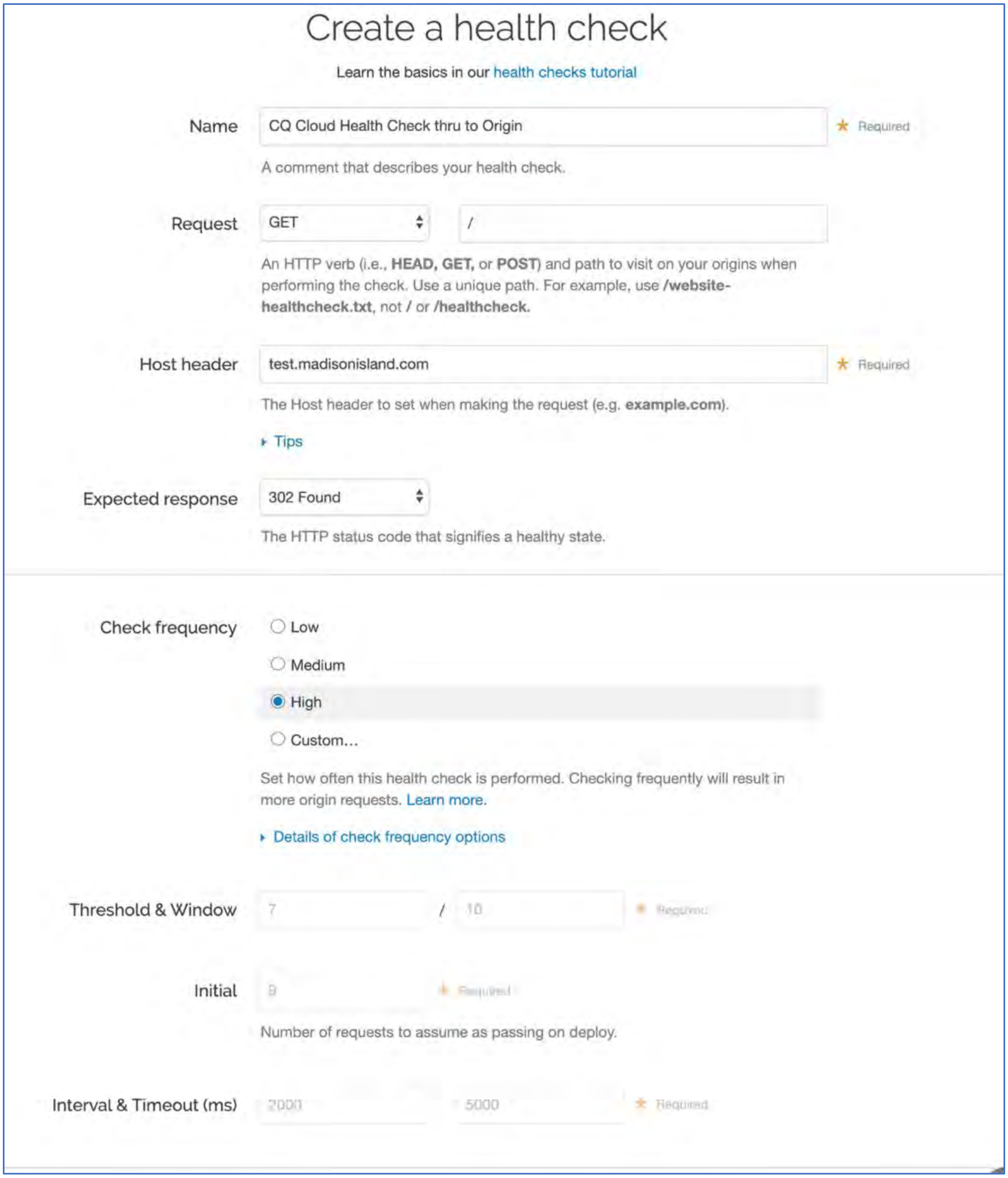

Health checks

Health checks monitor the status of your hosts—you can set Fastly to use a different origin, serve stale content, and more.

Image 5: Health checks section

The Health Check needs to have the Host header field present in order to allow Bot Defense SaaS to forward the Health Check traffic onto the Application Origin.

In the example below, the health check traffic is sent to the URI of "/" along with the respective Host Header and a 302 response is expected in order to indicate a success.



Create a health check

Learn the basics in our health checks tutorial

Name: CQ Cloud Health Check thru to Origin ★ Required

A comment that describes your health check.

Request: GET /

An HTTP verb (i.e., **HEAD, GET,** or **POST**) and path to visit on your origins when performing the check. Use a unique path. For example, use **/website-healthcheck.txt**, not **/** or **/healthcheck.**

Host header: test.madisonisland.com ★ Required

The Host header to set when making the request (e.g. **example.com**).

▸ Tips

Expected response: 302 Found

The HTTP status code that signifies a healthy state.

Check frequency:

○ Low

○ Medium

● High

○ Custom…

Set how often this health check is performed. Checking frequently will result in more origin requests. Learn more.

▸ Details of check frequency options

Threshold & Window: 7 / 10 ★ Required

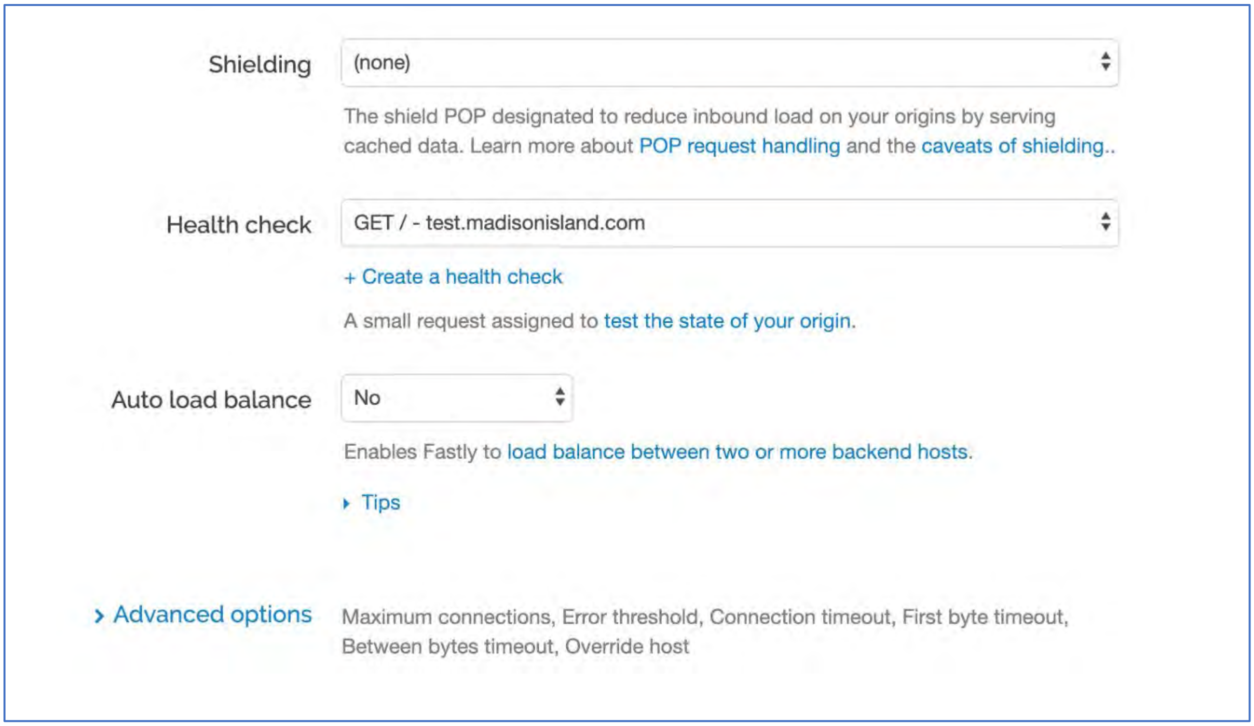
Initial: 9 ★ Required

Number of requests to assume as passing on deploy.

Interval & Timeout (ms): 2000 5000 ★ Required

Image 6a: Health check configuration for Bot Defense SaaS origin

Once the Health Check configuration is created, edit the Bot Defense SaaS Host configuration (test.s.cequence.cloud, in our case) and assign to it the Health Check that was created.



Shielding: (none)

The shield POP designated to reduce inbound load on your origins by serving cached data. Learn more about POP request handling and the caveats of shielding..

Health check: GET / - test.madisonisland.com

+ Create a health check

A small request assigned to test the state of your origin.

Auto load balance: No

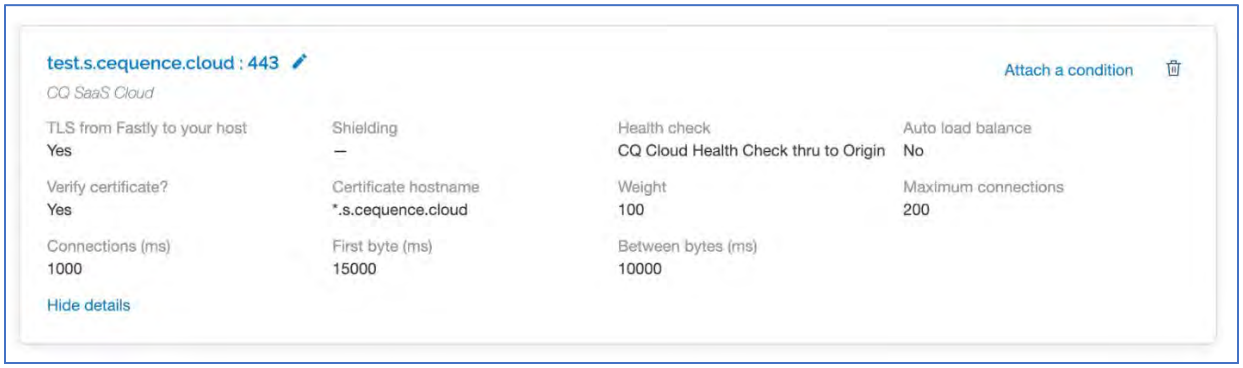
Enables Fastly to load balance between two or more backend hosts.

▸ Tips

› Advanced options: Maximum connections, Error threshold, Connection timeout, First byte timeout, Between bytes timeout, Override host

Image 6b: Health check configuration for Bot Defense SaaS origin

After assignment of the Health check, the Host summary should appear as below:



test.s.cequence.cloud : 443

CQ SaaS Cloud

TLS from Fastly to your host	Shielding	Health check	Auto load balance
Yes	–	CQ Cloud Health Check thru to Origin	No
Verify certificate?	Certificate hostname	Weight	Maximum connections
Yes	*.s.cequence.cloud	100	200
Connections (ms)	First byte (ms)	Between bytes (ms)	
1000	15000	10000	

Hide details

Image 7: Bot Defense SaaS Origin Summary with Health Check Assigned

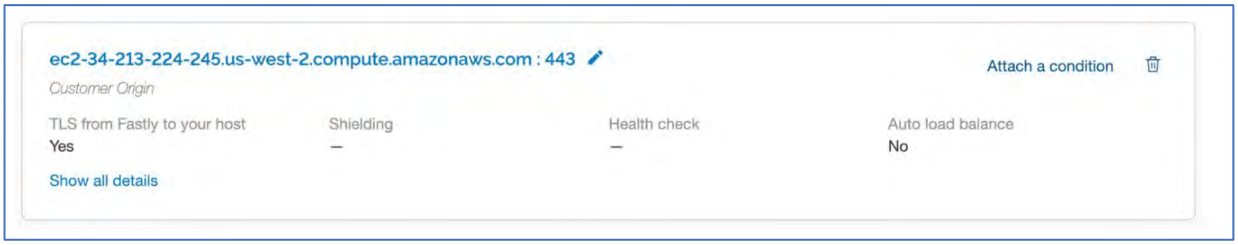
Step 3: Configure Traffic Forwarding to Bot Defense SaaS

In the Fastly configuration, the Host that is configured without attaching any condition is treated as the Default Host for forwarding application traffic.

Since all application traffic from Fastly will first need to be forwarded to Bot Defense SaaS, we do not attach any condition to it.

Instead, a condition will be attached to forward traffic to the customer’s Application Origin. This condition will typically be that of a health check failure to Bot Defense SaaS in order to trigger a fail-open to the Application Origin.

To set this up, click on Attach a condition for the Customer Application Origin Host, and create the condition as shown below:



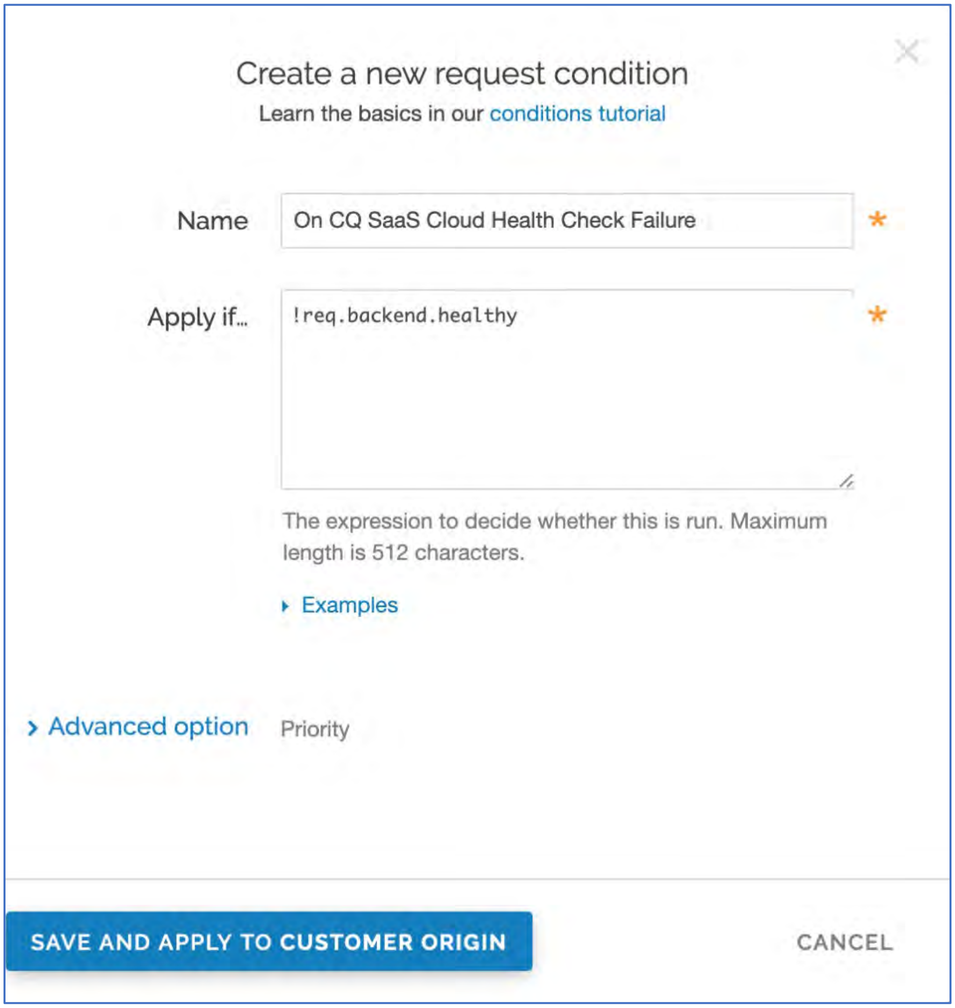
ec2-34-213-224-245.us-west-2.compute.amazonaws.com : 443

Customer Origin

TLS from Fastly to your host	Shielding	Health check	Auto load balance
Yes	—	—	No

Image 7: See Attach a condition

Create a new request condition and Save the configuration.



Create a new request condition

Learn the basics in our conditions tutorial

Name: On CQ SaaS Cloud Health Check Failure

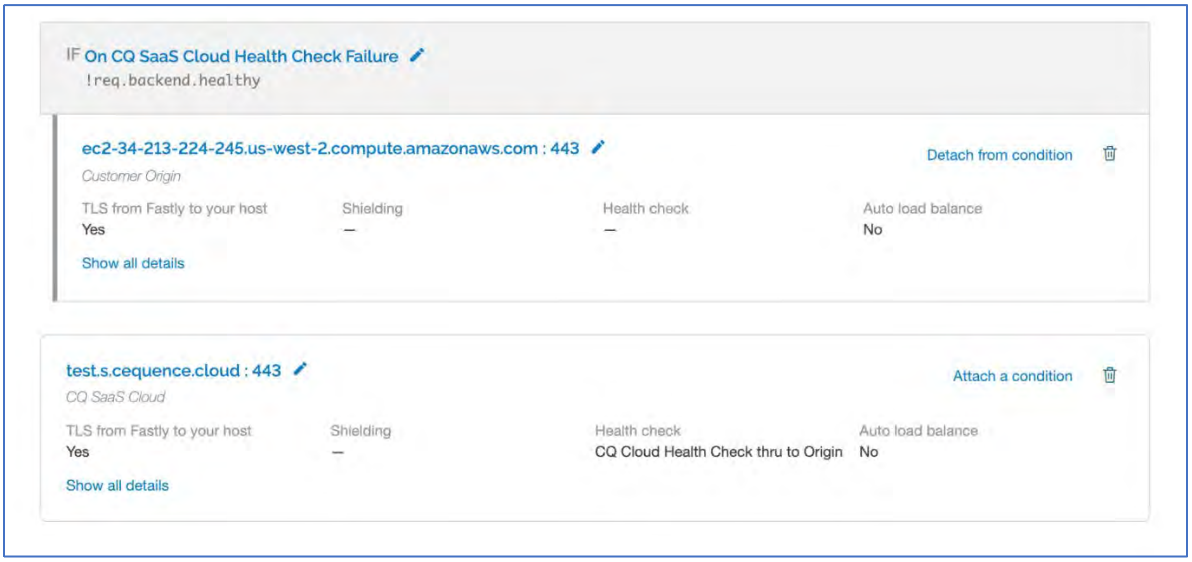
Apply if...:

```
!req.backend.healthy
```

The expression to decide whether this is run. Maximum length is 512 characters.

Image 8: Create a health check fail-open condition

The below image is indicative of the way Origin > Hosts section would appear after Bot Defense SaaS has been configured as the Default Origin and a condition has been attached for forwarding traffic to the Application Origin.



IF On CQ SaaS Cloud Health Check Failure
`!req.backend.healthy`

ec2-34-213-224-245.us-west-2.compute.amazonaws.com : 443
Customer Origin

TLS from Fastly to your host	Shielding	Health check	Auto load balance
Yes	–	–	No

test.s.cequence.cloud : 443
CQ SaaS Cloud

TLS from Fastly to your host	Shielding	Health check	Auto load balance
Yes	–	CQ Cloud Health Check thru to Origin	No

Image 9: Bot Defense SaaS origin and the application origin with the fall-open condition can be seen

Step 4: Configure Traffic Forwarding to Application Origin – Loopback Only

As shown in the loopback architecture traffic flow diagram (option 2 on page 2), Fastly forwards all application traffic to Bot Defense SaaS, by default.

- Bot Defense SaaS then adds a pre-shared key in a specialized request header to all the application traffic it processes and forwards to Fastly.
- When this traffic hits Fastly again, using a conditional placed on the presence of the pre-shared key in the specialized request header added by Bot Defense SaaS, Fastly makes the determination to no longer forward traffic to Bot Defense SaaS, and instead forwards it to the Application Origin.