



# Cloudflare ACE Training Handbook

This page outlines all labs and assignments. Please make use of this page if you happen to be faster or slower than the training speed.



## Learning Objective:

After completing the training, you can implement **Cloudflare application services** for your customers, on a POC or a production project.

To ensure that, **please make sure to complete all the labs and assignments!**

## 🔌 Table of Contents

[Prerequisite](#)

[Day 1. Implementing Cloudflare](#)

[Day 2. Optimizing Cloudflare](#)

[Day 3. Troubleshooting Cloudflare](#)

[Take the ACE Exam](#)

[Take Home Assignments](#)

## Prerequisite

▼ Click to view

## Day 1. Implementing Cloudflare

### 👉 CNAME Setup Onboarding Labs

▼ A. Get your domain

In this guide, we will use free registrar service at freenom.com. But do feel free to use your own choice of registrar for this step.

1. Go to [freenom.com](#) and create your account.
2. Get a free domain at [services - Register new domain](#)
3. Go to Cloudflare web dashboard.
4. Access [Partners Demo \(Bootcamp\) Account](#)
5. Click [Add a site](#) and register your new domain.



Add the root domain `yourdomain.cf`, not the FQDN `test.yourdomain.cf`

6. Select Enterprise Plan.

7. Finish the guide.

**Complete your nameserver setup**  
julliet.cf is not yet active on Cloudflare.

**1. Log in to your registrar account**  
Determine your registrar via [WHOIS](#).  
Remove these nameservers:

```
ns03.freedom.com  
ns04.freedom.com  
ns02.freedom.com  
ns01.freedom.com
```

**2. Replace with Cloudflare's nameservers**

Nameserver 1  
clarissa.ns.cloudflare.com  
Click to copy

Nameserver 2  
jacob.ns.cloudflare.com  
Click to copy

**Save your changes.**  
Registrars can take 24 hours to process nameserver updates. You will receive an email when your site is active on Cloudflare.

**Quick Actions**

Under Attack Mode  
Show visitors a JavaScript challenge when visiting your site.  
 Off  
Show more

**Domain Registration**

Registrar: Unknown  
Manage domain

**Plan Extensions**

Rate Limiting 100 rules allowed  
Page Rules 125 rules allowed  
Enterprise

**Support Resources**

Guidance & Best Practices: [success@cloudflare.com](#)

This is what you would be seeing after the steps.

▼ B. Account activation on [CNAME setup](#)



We are following the CNAME setup to learn about customer's testing scenario.  
You can also use Cloudflare as your authoritative DNS following full setup.

1. Go to Cloudflare dashboard - Overview.

2. Find [Advanced Options -- Convert to CNAME DNS Setup](#) at the bottom of the screen.

3. Read the warning and click [Convert](#).

The screenshot shows the Cloudflare dashboard for the domain 'julliet.cf'. A red box highlights the 'Record Type' dropdown set to 'TXT' and the 'Content' field containing '626959289-386659963'. Below this, a message states: 'Cloudflare will automatically re-check your TXT record. If your TXT record is correct, your site will be immediately activated. Please allow up to 6 hours for this change to be processed.' To the right, there are sections for 'Quick Actions', 'Domain Registration', and 'Plan Extensions'.

4. Find TXT record for the zone validation, then go to Freenom dashboard.
5. At Freenom, access [My domains - Manage Domain - Manage Freenom DNS](#) and add TXT record you were given.

The screenshot shows the Freenom DNS management interface. A red box highlights the 'Name' field with 'cloudflare-verify.yourdomain.cf' and the 'Type' dropdown set to 'TXT'. The 'Content' field also contains '626959289-386659963'. Below the table, there are buttons for '+ More Records' and 'Save Changes'.

6. Confirm the TXT record propagation at <https://dnschecker.org>. Then confirm Cloudflare account validation.

The screenshot shows the Cloudflare dashboard with a green checkmark message: 'Great news! Cloudflare is now protecting your site'. Below it, a note says 'Data about your site's usage will be here once available.' To the right, there are sections for 'Quick Actions', 'DNS Settings', 'Under Attack Mode', and 'Development Mode'.

Congratulations, account activation success!

#### ▼ C. Health checks — CF ↔ Origin

Always better advised to set up Cloudflare to origin health check to ensure the origin responds to Cloudflare correctly before pushing traffic through Cloudflare proxy.

1. Go to Cloudflare dashboard.
2. Access Traffic - Health Checks.
3. Configure health checks. Today's origin: **35.234.81.115**.

### Create Health Check

Name i

Origin i

Description (optional)

Type i Port i

Interval i Timeout i

Retries i Check regions i  
 Oceania x South East Asia x North East Asia x India x x ▾

---

Notifications i  
 Disable notifications

Health change thresholds  
 Healthy events  Unhealthy events

Health Event notifications  
 ▾

Recipients  
 Remove  
[+ Add notification email](#)

- You should always proceed to next steps after confirming your health check is successful.

Manage Health Checks				
Enable, disable, modify or delete configured Health Checks.				
Status	Name	Failures last 24hr	Enabled	
Healthy	jean-origin-check	0	<input checked="" type="checkbox"/>	<a href="#">Edit   Delete</a>

Like this.

- ▼ D. Activate Cloudflare for staging host first.

1. Go to Cloudflare dashboard - DNS.
2. At Cloudflare DNS, add a DNS record as following:

Type: A  
Name: staging  
IPv4 address: 35.234.81.115  
Proxy status: on (orange-clouded)

A few more steps are required to complete your setup. Hide

✓ Add an A, AAAA, or CNAME record for www so that [www.julliet.cf](https://www.julliet.cf) will resolve.

**DNS management for julliet.cf**

+ Add record  Search DNS Records Advanced

staging.julliet.cf points to 35.234.81.115 and has its traffic proxied through Cloudflare.

Type	Name	IPv4 address	TTL	Proxy status
A	staging	35.234.81.115	Auto	Proxied

Cancel Save

Type	Name	Content	TTL	Proxy status
CNAME	staging	staging.yourdomain.com.cdn.cloudflare.net	3600	Proxied

3. Go to Freenom DNS control panel.
4. At Freenom DNS, add a DNS record as following:

Type: CNAME  
Name: staging  
Value: staging.yourdomain.com.cdn.cloudflare.net  
TTL: 300

Add Records

Name	Type	TTL	Target
staging	CNAME	3600	staging.yourdomain.com.cdn.cloudflare.net

+ More Records Save Changes

5. Confirm the new DNS record propagation at <https://dnschecker.org>.
6. Confirm if your site is accessible at browser and/or terminal.

```
$ curl -svo /dev/null/ http://staging.yourdomain.com 2>&1 | grep 'HTTP'
HTTP/2 200 OK
```

Staging implementation complete. Please test the site to see if the site works as expected.

▼ E. Activate Cloudflare for the production host.

Activate Cloudflare for the **www** host. You can refer to the step d1-d6 if you are not sure how to.

▼ F. Confirm Cloudflare is enabled.

- Use your terminal to send below two requests, and compare the difference in HTTP response headers.

```
curl -svo /dev/null/ http://35.234.81.115/
curl -svo /dev/null/ http://staging.yourdomain.com/
curl -svo /dev/null/ https://www.yourdomain.com/
```

- Did you find the difference in response headers? Think about why.
- Are you able to connect to HTTPS to both sites? Think of a reason about why you can or can't.

```
C02SV9HKHF1R:~ jean$ curl -svo /dev/null/ http://35.234.81.115/
*   Trying 35.234.81.115...
* TCP_NODELAY set
* Connected to 35.234.81.115 (35.234.81.115) port 80 (#0)
> GET / HTTP/1.1
> Host: 35.234.81.115
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Mon, 22 Feb 2021 08:28:39 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Last-Modified: Wed, 11 Nov 2020 12:32:31 GMT
< ETag: "119b-5b3d3fd40ca9b"
< Accept-Ranges: bytes
< Content-Length: 4507
< Vary: Accept-Encoding
< Content-Type: text/html
<
{ [1154 bytes data]
* Failed writing body (0 != 1154)
* Closing connection 0
C02SV9HKHF1R:~ jean$
```

Send curl to the origin.

```
C02SV9HKHF1R:~ jean$ curl -svo /dev/null/ http://www.ace-training.cf/
*   Trying 2606:4700::6812:1aa...
* TCP_NODELAY set
* Connected to www.ace-training.cf (2606:4700::6812:1aa) port 80 (#0)
> GET / HTTP/1.1
> Host: www.ace-training.cf
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Date: Mon, 22 Feb 2021 08:29:04 GMT
< Transfer-Encoding: chunked
< Connection: keep-alive
< Cache-Control: max-age=3600
< Expires: Mon, 22 Feb 2021 09:29:04 GMT
< Location: https://www.ace-training.cf/
< cf-request-id: 086a744af300001a62b1a1f0000000001
< X-Content-Type-Options: nosniff
< Server: cloudflare
< CF-RAY: 62575657ea9b1a62-SIN
<
{ [5 bytes data]
* Connection #0 to host www.ace-training.cf left intact
* Closing connection 0
```

Send curl to Cloudflare onboarded domain.

## 👉 DNS, SSL, Onboarding Check Lab

### ▼ DNS Lab

Performance comparison between customer's current DNS and Cloudflare DNS.

Let's learn why we would like to recommend Cloudflare DNS to our customers.

- Go to [DNS Speed Test Tool](#)
- Try with non-Cloudflare domain. e.g. your customer domain

# DNS Speed Test

The DNS hosting speed tool will give you valuable DNS performance information for each level in the DNS tree to assist with performance evaluations and performance troubleshooting.

Enter a host name or domain name:

[Go »](#)

Related Tools: [Zone File Dump](#) [DNS Query Estimator](#) [DNS Lookup](#) [DNS Traversal](#) [Traceroute](#)

## freenom.com

Name Server	A	TXT	SOA	SRV	SPF	CNAME	MX	AAAA
ns03.freenom.com.	83	83	83	83	83	83	83	83
ns01.freenom.com.	68	69	69	68	68	67	68	72
ns04.freenom.com.	83	83	83	83	83	83	83	83
ns02.freenom.com.	69	69	68	67	68	68	68	68
Min. Time(ms)	68	69	68	67	68	67	68	68
Max. Time(ms)	83	83	83	83	83	83	83	83
Avg. Time(ms)	75	76	75	75	75	75	75	76

Example: DNS speed test to freenom.com.

- Try with cloudflare.com

# DNS Speed Test

The DNS hosting speed tool will give you valuable DNS performance information for each level in the DNS tree to assist with performance evaluations and performance troubleshooting.

Enter a host name or domain name:

[Go »](#)

Related Tools: [Zone File Dump](#) [DNS Query Estimator](#) [DNS Lookup](#) [DNS Traversal](#) [Traceroute](#)

## cloudflare.com

Name Server	A	TXT	SOA	SRV	SPF	CNAME	MX	AAAA
ns3.cloudflare.com.	3	1	3	4	4	4	4	3
ns7.cloudflare.com.	3	2	3	3	3	3	3	3
ns6.cloudflare.com.	3	2	3	4	3	3	3	3
ns5.cloudflare.com.	3	3	3	3	3	3	4	4
ns4.cloudflare.com.	3	1	3	3	3	4	3	3
Min. Time(ms)	3	1	3	3	3	3	3	3
Max. Time(ms)	3	3	3	4	4	4	4	4
Avg. Time(ms)	3	1	3	3	3	3	3	3

Example: same to cloudflare.com.



### Would you like to test more?

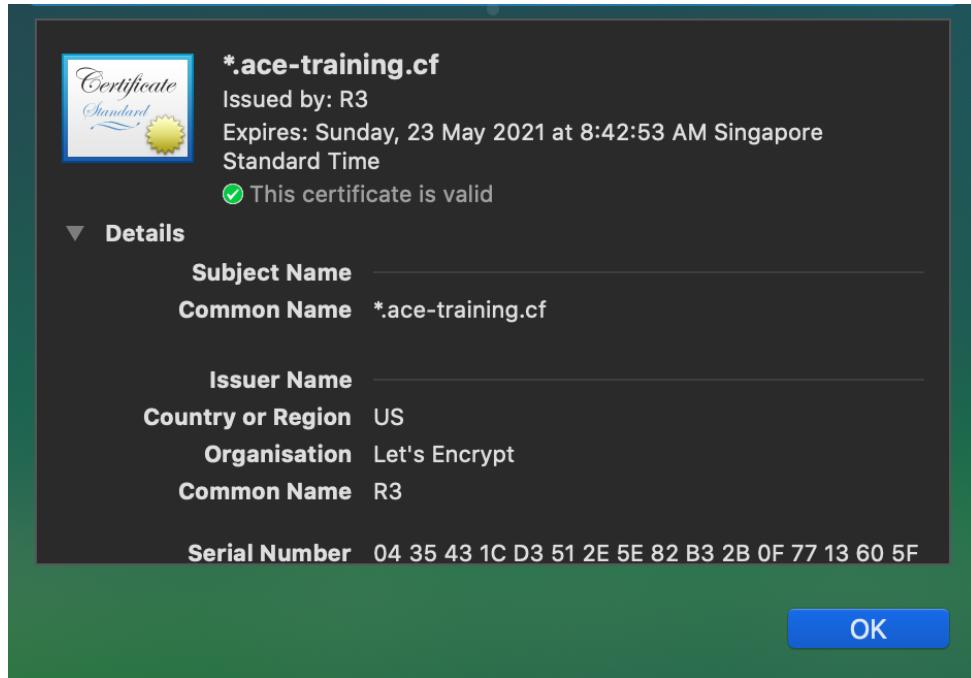
You may want to add a record at Freenom DNS (or your choice of any authoritative DNS) then measure its propagation time vs add a record at Cloudflare DNS (you need a full setup zone for this) then measure its propagation time. You can measure it at [dnschecker.org](#).

## ▼ SSL Lab

Labs — Easy SSL deployment and easy SSL redirection. Cloudflare Certificate is easy to issue and maintain!

### A. Already-available SSL without your action

- At your browser, connect to <https://www.yourdomain.cf/> and see the certificate information.



Confirm HTTPS is already available without your action. See the certificate information.

Hosts	Type	Status	Expires on
*.ace-training.cf, ace-training.cf	Universal	Active	2021-05-23 (Managed)

**Review Universal Certificate for \*.ace-training.cf, ace-training.cf**

The certificates in the pack listed below are managed and auto-renewed by Cloudflare.

Certificate	Expiration
SHA 2 RSA	2021-05-23 (Managed by Cloudflare)

**Certificate Validity Period**: 3 months  
**Validation method**: TXT  
**Certificate Authority**: Let's Encrypt

Learn how to verify which certificate is shown in the browser.

## B. Try the following commonly used SSL setting.

- Automatically redirect `http://` requests to `https://`
- Order an `Advanced Certificate` with your preferred custom hostnames, validity period, certificate authority.
- Upload your own certificate in the dashboard.

### ▼ Onboarding Checks

Let's have some fun with basic commands you can use for Cloudflare.

## ▼ dig

Dig is command line tool similar to nslookup that is used to run DNS queries and check DNS records for a given domain/website.

- Try it with your domain.

```
$ dig yourdomain.cf
```

```
[C02SV9HKHF1R:~ jean$ dig www.ace-training.cf

; <>> DiG 9.10.6 <>> www.ace-training.cf
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61048
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;www.ace-training.cf.           IN      A

;; ANSWER SECTION:
www.ace-training.cf.    300      IN      A          104.18.0.170
www.ace-training.cf.    300      IN      A          104.18.1.170

;; Query time: 86 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Mon Feb 22 23:57:53 +08 2021
;; MSG SIZE  rcvd: 80
```

## ▼ curl

cURL is a command line tool used to transport data using the URL syntax.

- Try it with your domain.

```
$ curl -svo /dev/null/ https://www.yourdomain.cf/
```

```
[C02SV9HKHF1R:~ jean$ curl -svo /dev/null/ https://www.ace-training.cf
* Trying 2606:4700::6812:aa...
* TCP_NODELAY set
* Connected to www.ace-training.cf (2606:4700::6812:aa) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/cert.pem
*   CApth: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
} [233 bytes data]
* TLSv1.2 (IN), TLS handshake, Server hello (2):
{ [100 bytes data]
* TLSv1.2 (IN), TLS handshake, Certificate (11):
{ [2267 bytes data]
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
{ [115 bytes data]
* TLSv1.2 (IN), TLS handshake, Server finished (14):
{ [4 bytes data]
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
} [37 bytes data]
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
} [1 bytes data]
* TLSv1.2 (IN), TLS handshake, Finished (20):
{ [16 bytes data]
* TLSv1.2 (IN), TLS change cipher, Change cipher spec (1):
{ [1 bytes data]
* TLSv1.2 (IN), TLS handshake, Finished (20):
{ [16 bytes data]
* SSL connection using TLSv1.2 / ECDHE-ECDSA-CHACHA20-POLY1305
* ALPN, server accepted to use h2
* Server certificate:
*   subject: CN=www.ace-training.cf
*   start date: Feb 22 01:39:55 2021 GMT
*   expire date: May 23 01:39:55 2021 GMT
*   subjectAltName: host "www.ace-training.cf" matched cert's "www.ace-training.cf"
*   issuer: C=US; O=Let's Encrypt; CN=R3
*   SSL certificate verify ok.
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
* Using Stream ID: 1 (easy handle 0x7fbf10009a00)
> GET / HTTP/2
> Host: www.ace-training.cf
> User-Agent: curl/7.64.1
> Accept: */
>
* Connection state changed (MAX_CONCURRENT_STREAMS == 256)!
< HTTP/2 200
< date: Mon, 22 Feb 2021 16:02:45 GMT
< content-type: text/html
< set-cookie: __cfduid=d2846ae48086e8ac16045a69c9459e86e1614009765; expires=Wed, 24-Mar-21 16:02:45 GMT; path=/; domain=.ace-training.cf; HttpOnly; SameSite=Lax; Secure
< last-modified: Wed, 11 Nov 2020 12:32:31 GMT
< vary: Accept-Encoding
< cf-cache-status: MISS
< expires: Tue, 22 Feb 2022 16:02:45 GMT
< cache-control: public, max-age=31536000
< cf-request-id: 086c13a50d0000cbf8a8289000000001
< expect-ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
< strict-transport-security: max-age=15552000; includeSubDomains; preload
< x-content-type-options: nosniff
< server: cloudflare
< cf-ray: 6259eee80ccccbf8-SIN
<
{ [585 bytes data]
* Failed writing body (0 != 585)
* stopped the pause stream!
* Connection #0 to host www.ace-training.cf left intact
* Closing connection 0
```

- Try this `curl` and see the difference.

```
$ curl -svo /dev/null/ http://www.yourdomain.cf/ --connect-to ::35.234.81.115
```

```
[C02SV9HKHF1R:~ jean$ curl -svo /dev/null/ http://www.ace-training.cf --connect-to ::35.234.81.115
* Connecting to hostname: 35.234.81.115
*   Trying 35.234.81.115...
* TCP_NODELAY set
* Connected to 35.234.81.115 (35.234.81.115) port 80 (#0)
> GET / HTTP/1.1
> Host: www.ace-training.cf
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Mon, 22 Feb 2021 16:16:43 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Last-Modified: Wed, 11 Nov 2020 12:32:31 GMT
< ETag: "119b-5b3d3fd40ca9b"
< Accept-Ranges: bytes
< Content-Length: 4507
< Vary: Accept-Encoding
< Content-Type: text/html
<
{ [4507 bytes data]
* Failed writing body (0 != 4507)
* Closing connection 0
```

This option will check the origin response directly.

#### ▼ mtr

MTR/Traceroute is Network based command line tools used to measure performance/latency on a particular path to a given host/destination.

- Try it with your domain.

```
$ sudo mtr www.yourdomain.cf
```

```
[C02SV9HKHF1R:~ jean$ sudo mtr www.ace-training.cf
```

My traceroute [v0.94]								
Keys:	Help	Display mode	Restart statistics	Order of fields	quit	Packets	Pings	
Host						Loss%	Snt	Last
1.	2406:3003:2004:1bf8:5aef:68ff:febe:29					0.0%	26	1.8
2.	2406:3003:1004::2					4.0%	26	21.6
3.	2406:3003:1:41::1					0.0%	26	6.4
4.	2406:3003:1:11::1					0.0%	25	32.8
5.	2400:cb00:35:3::a29e:a0e6					44.0%	25	5.1
6.	2606:4700::6812:aa					0.0%	25	12.2
								8.9
								1.1
								42.4
								10.1
								2.9
								37.3
								10.5
								4.4
								34.5
								8.1
								21.4
								89.3
								60.6
								15.4
								7.4

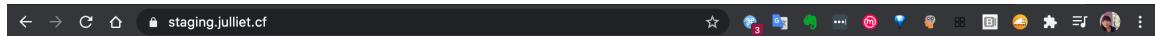
## Day 2. Optimizing Cloudflare

#### 👉 Onboarding Best Practice — Labs

#### ▼ Custom Error Pages

Configure example custom error pages. Let's see why we need "Custom Error Pages" for enterprise customers.

1. At Cloudflare dashboard, go to `TLS/SSL - Overview` and select `Strict (SSL only origin pull)`
2. At the browser, visit your website: `https://www.yourdomain.cf/`



## Error 526

Ray ID: 6260d676bf3d0197 • 2021-02-23 12:09:24 UTC

Invalid SSL certificate



You

Browser  
Working



Singapore

Cloudflare  
Working



staging.julliet.cf

Host  
Error

### What happened?

The origin web server does not have a valid SSL certificate.

### What can I do?

If you're a visitor of this website:

Please try again in a few minutes.

Think about why you see this error.

3. At the browser, access <http://www.ace-training.cf/error.html>
4. At Cloudflare dashboard, visit [Custom Pages - 500 Class Errors](#), and try publishing the page: <http://www.ace-training.cf/error.html>
5. Give Cloudflare some time to provision, then visit your website again: <https://www.yourdomain.cf/> and confirm the difference.
6. Once you're finished with testing, please roll back the SSL setting to [full](#).



Why configuring custom error pages is a must?

When something happens, what do you think the customer wants to show - their branded page or Cloudflare branded error page?

### ▼ Configure Alerts

Configure necessary alerts to your email address. Get alerted by email in the event of: Origin server failure, DDoS attack

1. At Cloudflare dashboard, go to [Account Home - Notification](#).
2. Try setting [HTTP DDoS Attack Alert](#), [Layer 3/4 DDoS Attack Alert](#) and [Passive Origin Monitoring](#) to your work email.
3. Once alert is set, you will get alerted on HTTP DDoS events, and origin health (even if no active health check) based on limited set of passive error codes. (e.g. 521)



#### Note:

- Alert only works after you have added the notifications;
- Layer 3/4 DDoS Attack Alerts is currently available for Magic Transit, Spectrum customers only;
- Customizing alerting threshold is not supported yet

### ▼ Is the Origin Correctly Secured?

In this lab we will scan the origin server and understand the implementation best practice. You should be permitted to scan the origin server when you run this for the customer.

1. Try access <https://pentest-tools.com/network-vulnerability-scanning/tcp-port-scanner-online-nmap>
2. Enter the test origin address `35.234.81.115` and run a `light scan`. Check the result.
3. If you have, check your own test origin server or a customer's origin server.
4. Do you think this origin is opening the web port to ANY? Were you able to confirm?
5. Even if the origin is set like this, if it's behind Cloudflare, is it immune to DDoS and security threats?

Found 4 open ports (1 host)

□ 35.234.81.115					
> 115.81.234.35.bc.googleusercontent.com					
Port Number	State	Service Name	Service Product	Service Version	Service Extra Info
● 22	open	ssh	OpenSSH	7.2p2 Ubuntu 4ubuntu2.8	Ubuntu Linux; protocol 2.0
● 80	open	http	Apache httpd	2.4.18	(Ubuntu)
● 81	open	http	Apache httpd	2.4.46	(Unix)
● 443	open	https	Apache httpd	2.4.18	(Ubuntu)
● 3306	closed	mysql			
● 3389	closed	ms-wbt-server			
● 5000	closed	upnp			
● 8080	closed	http			
● 8443	closed	https-alt			

What should you advise to the owner of this server?



## ▼ WAF

Let's do a quick pentest before and after WAF configuration, check the logs

1. Access Cloudflare dashboard - [Firewall - Managed Rules](#). Confirm WAF is [OFF](#).

2. Access [https://www.yourdomain.cf/file.php?cmd=echo\(shell\\_exec\(%22ls%20/etc/var%22\)\)](https://www.yourdomain.cf/file.php?cmd=echo(shell_exec(%22ls%20/etc/var%22)))

```
$ curl -svo /dev/null/ "https://www.yourdomain.cf/file.php?cmd=echo(shell_exec(%22ls%20/etc/var%22))"
```

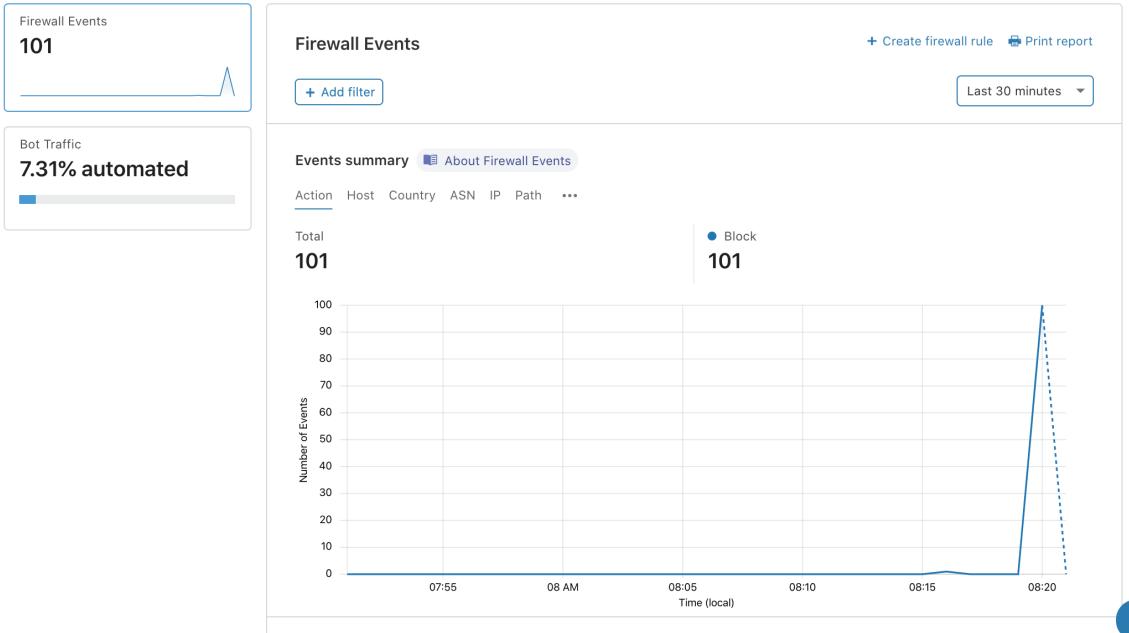
3. Turn the WAF [on](#).

4. Re-try the exploit.

5. (Optional) Try many times

```
$ for i in {1..100}; do curl -svo /dev/null/ -H "exploit: true" "https://www.yourdomain.cf/file.php?cmd=echo(shell_exec(%22ls%20/etc/var%22))" 2>&1 | grep "< HTTP"; done;
```

6. Find the blocked logs at Cloudflare Firewall dashboard.



## ▼ Security Level

Let's understand the Security Level, JS Challenge everyone with "I'm Under Attack Mode"

1. At Cloudflare dashboard, adjust your test domain's security level to [I'm Under Attack Mode](#).

2. At your browser, try access the domain and check the result and status codes.

3. At your terminal, try access the domain and check the result and status codes.
4. After the exercise, **turn off the IUAM** (change it to medium/high) so the next exercise won't be disturbed.



## Checking your browser before accessing julliet.cf.

This process is automatic. Your browser will redirect to your requested content shortly.

Please allow up to 5 seconds...

DDoS protection by [Cloudflare](#)

Ray ID: 62651d311c0b18fc

Make sure you know what is this for.

Do you think the security level should be on if the customer has a public API endpoint?

### ▼ Rate Limiting

Rate Limiting is a must add-on for enterprise clients who are worried about DDoS attack.

We will do a quick test before and after Rate Limiting configuration, check the logs

- At your terminal, run this command to send 200 requests to the site:

```
for i in {1..200}; do curl -svo /dev/null/ -H "requestflood: true" "https://www.yourdomain.cf/" 2>&1 | grep "< HTTP"; done;
```

- Confirm your requests are served with status code **200**.
- At Cloudflare dashboard, create a Rate Limiting rule.

## Create a Rate Limiting Rule

X

**Rule Settings**

Rule Name  X

If Traffic Matching the URL

\*

from the same IP address exceeds  requests per  X

[Advanced Criteria ▾](#)

**Method(s) i**

X

---

**HTTP Response Header(s) i**

X

[+ Add header response field](#)

Also apply rate limit to cached assets

---

**Origin Response code(s) i**

---

**NAT i**

Support users behind NAT

---

Then  matching traffic from that visitor for  X

When "Block" is set, when the threshold is exceeded, the Client will receive a "429" error page until the Block time has expired.

[Advanced Response ▾](#)

---

[Bypass ▾](#)

---

Cancel Save as Draft Save and Deploy

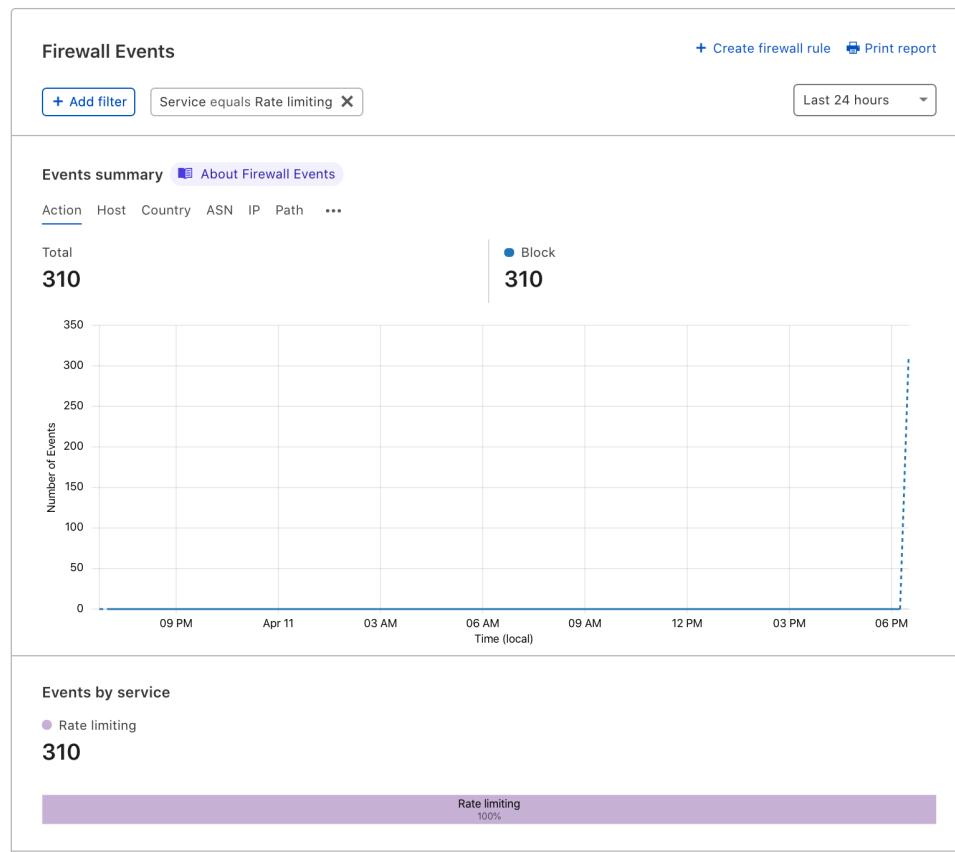
- Run the same command again.

```
for i in {1..200}; do curl -svo /dev/null/ -H "requestflood: true" "https://www.yourdomain.cf/" 2>&1 | grep "< HTTP"; done;
```

- See what happens to your flood requests.

```
< HTTP/2 200
< HTTP/2 429
```

- At Cloudflare dashboard, check the logs created just now.



## ▼ Firewall Rules

Using Firewall Rules, you can flexibly customize your organization's security rules to meet the needs. Let's test with below examples.

1. Try set a rule to only allow one client IP `1.2.3.4` when people access

`http(s)://yourtestdomain.cf/admin.html`.

2. Try set a rule to give `JS challenge` to everyone when the request to `http(s)://yourtestdomain.cf/*` is `NOT` coming from your country.

When incoming requests match...

```

graph TD
    IP[IP Address] -- equals --> IPVal[e.g. 192.0.2.0]
    IP --- And1[And]
    Country[Country] -- equals --> CountryVal[Select]
    Country --- And2[And]
    Hostname[Hostname] -- equals --> HostnameVal[e.g. example.com]
    Hostname --- And3[And]
    UserAgent[User Agent] -- equals --> UserAgentVal[e.g. Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6)...]
    UserAgent --- Or[Or]
    And1 --- And2
    And2 --- And3
    And3 --- Or
  
```

Expression Preview: `(ip.src eq and ip.geoip.country eq "" and http.host eq "" and http.user_agent eq "")`

Then...  
Choose an action  
Block Test rule

Try to create the rules

## 👉 Performance Labs

### ▼ Synthetic Performance Test

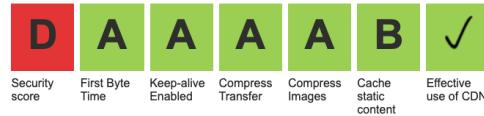
Learn how to run synthetic performance tests, and think about how you can make your Cloudflare site faster than you see.

1. Access <https://webpagetest.org>
2. Run a test to <https://www.yourdomain.cf/> (choose a node you want; e.g. Singapore EC2)
3. Open another tab with the same URL webpagetest.org, run another test to the sample site  
— <https://optimized.ace-training.cf/>  
This is the same page proxying to the same origin. Use the same location with above (e.g. Singapore EC2)
4. Compare two results. Content is same. Is the result similar or different? Let's think about why.

## Web Page Performance Test for

<https://www.ace-training.cf>

From: Melbourne, Australia - Azure - Chrome - Cable  
24/02/2021, 08:56:31



[Summary](#) [Details](#) [Performance Review](#) [Content Breakdown](#) [Domains](#) [Processing Breakdown](#) [Screenshot](#) [Image Analysis](#) [Request Map](#)

First View only  
Test runs: 3  
[Re-run the test](#)

[View JSON result](#)  
[Raw page data - Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

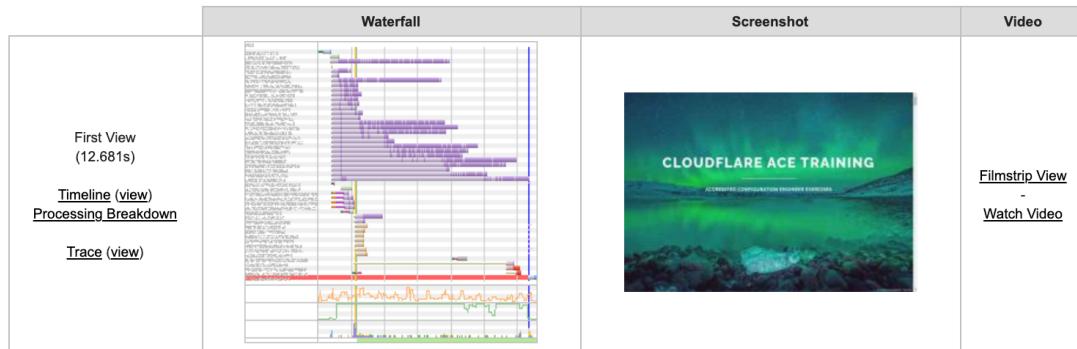
### Performance Results (Median Run - SpeedIndex)

	Web Vitals						Document Complete			Fully Loaded				
	First Byte	Start Render	First Contentful Paint	Speed Index	Largest Contentful Paint	Cumulative Layout Shift	Total Blocking Time	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View (Run 3)	0.179s	1.200s	1.220s	8.688s	1.973s	0.055	0.000s	11.796s	48	6,571 KB	11.878s	49	6,572 KB	\$\$\$\$\$

[Plot Full Results](#)

## Test Results

Run 1:



Think about why the difference between [this](#) and [that](#)



**Challenge:** Make your domain as fast as [optimized.ace-training.cf](https://optimized.ace-training.cf) too!

### ▼ Granular Settings

Understand how you can customize Cloudflare cache settings. Try creating a caching rule for all [HTML pages](#) under the domain.

- At the Edge:
  - If `status code == 200(OK)`, cache `1 minute`
  - If `status code == 403(FORBIDDEN)` or `404(NOT FOUND)`, cache `10 minutes`
- At client browser: Recommend cache for `30 seconds`



You can make rules at [Page Rules](#)

## 💡 Optimization Summary

▼ Click to view

#### Optimization (Fine-tuning) Checklist

Aa Category	<input checked="" type="checkbox"/>	Checklist
<u>Onboarding</u>	<input type="checkbox"/>	Set up health checks
<u>Untitled</u>	<input type="checkbox"/>	Set up necessary notification email
<u>Untitled</u>	<input type="checkbox"/>	Use custom error pages
<u>Untitled</u>	<input type="checkbox"/>	Consider Cloudflare dashboard 2FA or SSO
<u>Untitled</u>	<input type="checkbox"/>	Origin to allow-list Cloudflare IPs
<u>Untitled</u>	<input type="checkbox"/>	Review CF headers/cookies, restore the originating IP if needed.
<u>Untitled</u>	<input type="checkbox"/>	Enable CF Logpush (or Logpull)
<u>Security</u>	<input type="checkbox"/>	Avoid HTTP, redirect/rewrite to HTTPS
<u>Untitled</u>	<input type="checkbox"/>	Hide origin IP/port
<u>Untitled</u>	<input type="checkbox"/>	Turn on WAF
<u>Untitled</u>	<input type="checkbox"/>	Use Security Level at Med or High, IUAM if needed.
<u>Untitled</u>	<input type="checkbox"/>	Set Rate Limiting Rules
<u>Untitled</u>	<input type="checkbox"/>	Get bots visibility and control them
<u>Untitled</u>	<input type="checkbox"/>	Deploy their security needs at Cloudflare Firewall Rules
<u>Performance</u>	<input type="checkbox"/>	Cache as much as possible
<u>Untitled</u>	<input type="checkbox"/>	Maximise use of optimization features
<u>Untitled</u>	<input type="checkbox"/>	Use Brotli compression
<u>Untitled</u>	<input type="checkbox"/>	Use Argo Smart Routing
<u>Untitled</u>	<input type="checkbox"/>	Use newer TLS and better technologies
<u>Untitled</u>	<input type="checkbox"/>	Keep the customer's staging subdomain so you can always use it for origin compatibility test vs. CF optimization feature

## Day 3. Troubleshooting Cloudflare

### 🛠 Troubleshooting Labs

▼ Turn on the Log

Cloudflare logs is enterprise only, and it has all the necessary information you need for 99% of troubleshooting. While Cloudflare recommends Logpush over Logpull (remember this on customer project), we will be using Logpull for this lab because not everyone has access to cloud-based storage.

- Follow this doc <https://developers.cloudflare.com/logs/logpull-api/enabling-log-retention> to enable Log retention. You can use Postman or Terminal.

```
1 ✓ [
2   "errors": [],
3   "messages": [],
4   "result": {
5     "flag": true
6   },
7   "success": true
8 ]
```

Confirm "flag": true, "success": true

### ▼ Use curl to compare responses

Use cURL resolve override to compare the behaviour between `response-coming-via-Cloudflare` vs `response-directly-from-origin`.

1. curl -svo /dev/null/ "http://www.yourdomain.cf"
  
2. curl -svo /dev/null/ "http://www.yourdomain.cf" --resolve www.yourdomain.cf:80:35.234.81.115
  
3. Think about in what situation you can make use of these two commands.

### ▼ Use cdn-cgi page

When a host is on Cloudflare there's a way to find metadata easily.

1. At Cloudflare dashboard, **grey-cloud** `staging.yourdomain.cf`
2. At the browser, visit `http://www.yourdomain.cf/cdn-cgi/trace/`
3. At another tab of the browser, visit `http://staging.yourdomain.cf/cdn-cgi/trace/`

### ▼ Get HAR file

Generate **HAR** file to show the status code and a page you see. If the error is harder to replicate, it will help the Cloudflare support team to understand what you are seeing, or what your customer is seeing.

1. At your browser, open developers tool first, access `http://www.yourdomain.cf/`
2. Right-click the mouse and find "Save as HAR with Content". Save it in your local.

The screenshot shows the Network tab in the Chrome DevTools. A red arrow points to the 'Record' button at the top left. Another red arrow points to the 'Save as HAR with Content' option in a context menu that has been opened over a list of captured requests.

3. Open [Google HAR Analyzer](#), open the HAR file you have just saved.

#### ▼ Retrieve Logs with RayID

`RayID` is Cloudflare's identifier of a request. Use RayID to retrieve detailed information of a request.

1. rayId: Find one RayID you would like to inspect
2. zoneName: `yourdomain.cf`
3. Guidance and example: [here](#)
4. Hint: [Start from a ready-made query](#)
5. Log fields and its meaning: <https://developers.cloudflare.com/logs/log-fields>

The screenshot shows the POSTMAN interface. A GET request is made to `https://api.cloudflare.com/client/v4/zones/90bfaa7f734549f7b1527666e6b379a/logs/rayids/6:...`. The response status is 200 OK, with a time of 2.42 s and a size of 2.12 KB. The log fields are displayed in the body:

```

10 "ClientRequestClass": "noRecord",
11 "ClientRequestBytes": 2897,
12 "ClientRequestHost": "www.ace-training.cf",
13 "ClientRequestMethod": "GET",
14 "ClientRequestPath": "/",
15 "ClientRequestProtocol": "HTTP/2",
16 "ClientRequestReferer": "",
17 "ClientRequestURI": "/",
18 "ClientRequestUserAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)",
19 "ClientSSLCipher": "AEAD-AES128-GCM-SHA256",
20 "ClientSSLProtocol": "TLSv1.3",
21 "ClientSrcPort": 50796,
22 "ClientXRequestedWith": "",
23 "EdgeColoCode": "SIN",
24 "EdgeColoID": 35,
...

```

Easier and quicker with [POSTMAN](#).

## ▼ Get All Logs for the Last 1 Hour

Try to use timestamp to download all logs of a Cloudflare ENT zone within specific time range.

1. zoneName: `yourdomain.cf`
2. time\_start: Use this format `2021-02-25T03:00:00Z`
3. time\_end: Use this format `2021-02-25T04:00:00Z`
4. Guidance and example: [here](#)
5. Hint: [Start from a readymade query](#)
6. Log fields and its meaning: <https://developers.cloudflare.com/logs/log-fields>

```
1  se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
2  se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
3  se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
4  se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
5  lse,"ClientASN":43948,"ClientCountry":"se","ClientDeviceType":"desktop","ClientIP":"185.39.146.214","ClientIPClass":"monitoringService"
6  se,"ClientASN":43948,"ClientCountry":"se","ClientDeviceType":"desktop","ClientIP":"94.247.174.83","ClientIPClass":"monitoringService"
7  lse,"ClientASN":36351,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"169.51.2.22","ClientIPClass":"monitoringService"
8  lse,"ClientASN":19148,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"23.83.129.219","ClientIPClass":"monitoringService"
9  lse,"ClientASN":30083,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"148.72.170.233","ClientIPClass":"monitoringService"
10 false,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
11 se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
12 :false,"ClientASN":32489,"ClientCountry":"ca","ClientDeviceType":"desktop","ClientIP":"184.75.210.90","ClientIPClass":"monitoringService"
13 se,"ClientASN":60781,"ClientCountry":"nl","ClientDeviceType":"desktop","ClientIP":"94.75.211.73","ClientIPClass":"monitoringService"
14 se,"ClientASN":53340,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"76.164.234.170","ClientIPClass":"monitoringService"
15 lse,"ClientASN":24961,"ClientCountry":"de","ClientDeviceType":"desktop","ClientIP":"46.20.45.18","ClientIPClass":"monitoringService"
16 se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
17 false,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
18 lse,"ClientASN":199081,"ClientCountry":"gr","ClientDeviceType":"desktop","ClientIP":"185.70.76.23","ClientIPClass":"monitoringService"
19 se,"ClientASN":199081,"ClientCountry":"gr","ClientDeviceType":"desktop","ClientIP":"185.70.76.23","ClientIPClass":"monitoringService"
20 :false,"ClientASN":32489,"ClientCountry":"ca","ClientDeviceType":"desktop","ClientIP":"184.75.208.210","ClientIPClass":"monitoringService"
21 false,"ClientASN":32489,"ClientCountry":"ca","ClientDeviceType":"desktop","ClientIP":"184.75.210.90","ClientIPClass":"monitoringService"
22 false,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
23 se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
24 se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
25 se,"ClientASN":49367,"ClientCountry":"es","ClientDeviceType":"desktop","ClientIP":"95.141.37.2","ClientIPClass":"monitoringService"
26 se,"ClientASN":8100,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"104.129.30.18","ClientIPClass":"monitoringService"
27 :false,"ClientASN":394380,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"172.241.112.86","ClientIPClass":"monitoringService"
28 lse,"ClientASN":17090,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"162.208.48.94","ClientIPClass":"monitoringService"
29 se,"ClientASN":60781,"ClientCountry":"nl","ClientDeviceType":"desktop","ClientIP":"94.75.211.73","ClientIPClass":"monitoringService"
30 se,"ClientASN":17090,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"76.72.172.208","ClientIPClass":"monitoringService"
31 lse,"ClientASN":30083,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"148.72.171.17","ClientIPClass":"monitoringService"
32 se,"ClientASN":132892,"ClientCountry":"us","ClientDeviceType":"desktop","ClientIP":"2a06:98c0:3600::103","ClientIPClass":"noRecord"
```

Easier and quicker with [POSTMAN](#).

## ▼ Traceroute from Cloudflare colo

If the feature is enabled, you can use API to run traceroute from specific Cloudflare data center to the origin server.

1. Access <https://api.cloudflare.com/#diagnostics-properties>
2. Try running a diagnostics to: `35.234.81.115` from: `SIN02`.
3. Hint: Account ID can be found in dashboard overview or at the address line.

The screenshot shows the POSTMAN application interface. At the top, it displays the URL: <https://api.cloudflare.com/client/v4/accounts/ad837c08906f93cce77fe21cdce2ddb5/diagnostics>. Below the URL, there are tabs for Params, Authorization, Headers (13), Body (green dot), Pre-request Script, Tests, Settings, Cookies, and Code. The Body tab is selected. The response status is 200 OK, with a time of 20.42 s and a size of 3.7 KB. The response body is displayed in JSON format:

```

1  [
2   "result": [
3     {
4       "target": "35.234.81.115",
5       "colos": [
6         {
7           "colo": {
8             "name": "sin02",
9             "city": "Singapore, SG"
10            },
11            "traceroute_time_ms": 20041,
12            "target_summary": {
13              "asn": "",
14              "ip": "35.234.81.115",
15              "name": "35.234.81.115",
16              "packet_count": 0,
17              "mean_rtt_ms": 0,
18              "std_dev_rtt_ms": 0
19            }
20          }
21        ]
22      }
23    ]
24  ]

```

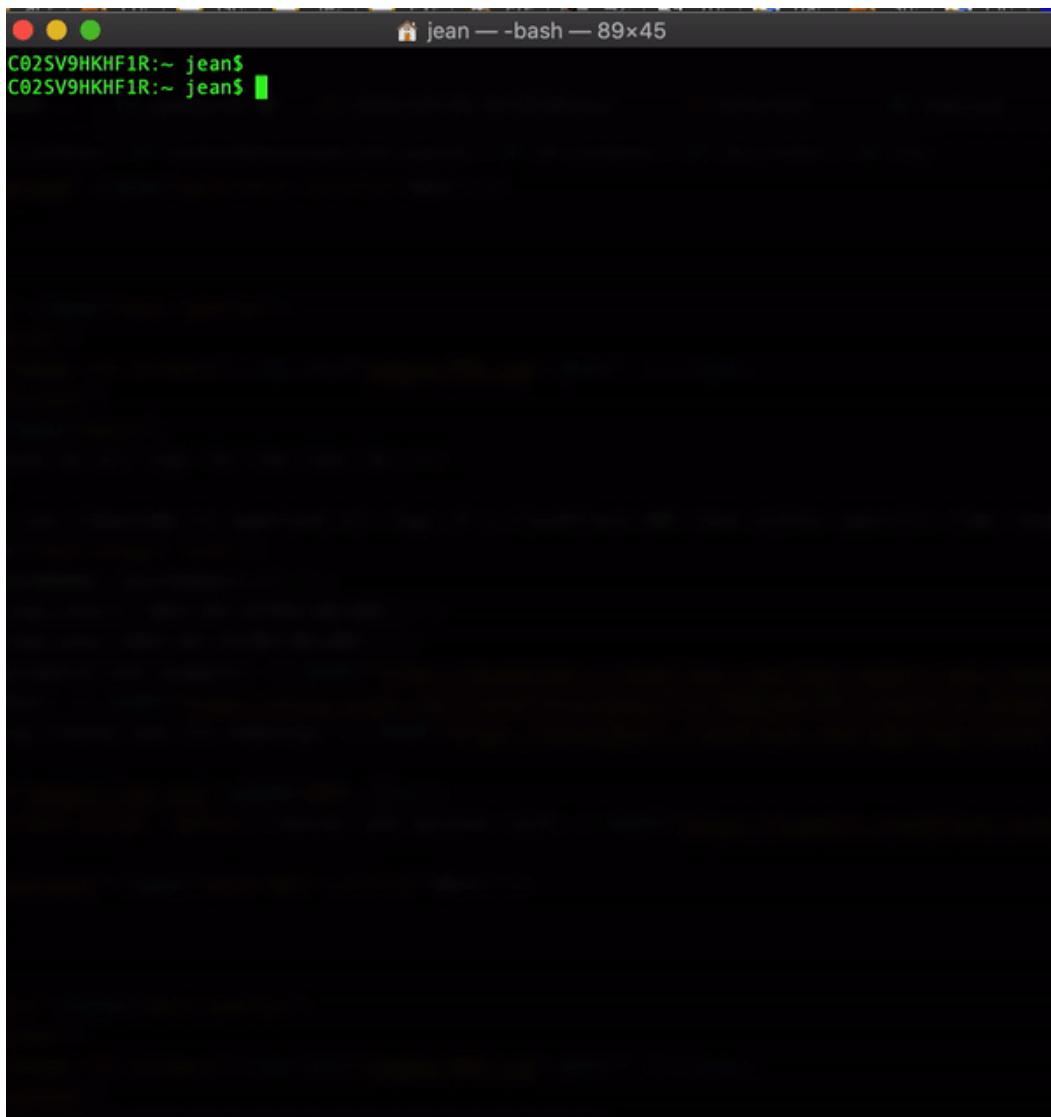
Easier and quicker with [POSTMAN](#).

## ▼ Play more with curl

Directly send request to origin, repeat cURL 50 times to confirm intermittent error, send specific headers, etc.

```
for i in {1..50}; do curl -svo /dev/null/ "https://www.cloudflare.com?x=${i}" 2>&1 | grep -Ei "< HTTP|< Date|< CF-Cache-Status|< CF-RAY|< Server"; printf "\n\n"; done;
```

```
curl -o /dev/null -D- -k -s -w '%{url_effective} CODE:%{http_code} \n -- DNS:%{time_namelookup} \n - - Connect:%{time_connect} \n -- TTFB:%{time_starttransfer} \n -- Time_Pretransfer:%{time_pretransfer} \n -- Time_Redirect:%{time_redirect} \n -- Time_Start_Transfer:%{time_starttransfer} \n -- TOTAL Time:%{time_total}\n -- Size_download:%{size_download} \n' "https://cloudflare.com"
```



## 💥 Troubleshooting Summary

▼ Click to view

- **Initial t/s investigation checkpoints:**
  1. Health checks
  2. CF-Origin response codes cross check (cURL)
  3. Confirm logs of the corresponding RayID
  4. Change logs (audit logs) of the problematic time
  5. Check [cloudflarestatus.com](https://cloudflarestatus.com)
  6. Save cdn-cgi and HAR
- **Still can't find the root cause? Send SOS to the below contact:**

1. Customer issue: [entsupport@cloudflare.com](mailto:entsupport@cloudflare.com)
2. Partner issue: [partnersupport@cloudflare.com](mailto:partnersupport@cloudflare.com)
- 3.💡 IMPORTANT: Share the initial t/s result. It will save time!
- 4.💡 IMPORTANT: Need to contact support from the right legitimate email.

## Take the ACE Exam

Once you completed the training, take the ACE certification at the [Cloudflare partner portal](#).

Successful candidates who passed ACE can take further steps and attend Accredited Services Architect coursework.

## Take Home Assignments

**Do you want to get hands-on yourself and ensure you are comfortable with Cloudflare implementation?**

Please complete the below technical assignments. Please contact your partner SE (APAC: [jean@cloudflare.com](mailto:jean@cloudflare.com)) if you have any questions!

▼ [Click to view the assignment](#)

Aa Assignment	≡ Requirements	≡ Reference	≡ Verification
<a href="#"><u>Onboarding</u></a>	Implement Cloudflare so the following hosts are accessible via Cloudflare: <a href="https://web.yourdomain.com/">https://web.yourdomain.com/</a> <a href="https://api.yourdomain.com/">https://api.yourdomain.com/</a> <a href="https://admin.yourdomain.com/">https://admin.yourdomain.com/</a> , and <a href="ssh.yourdomain.com">ssh.yourdomain.com</a> You can use full setup or CNAME setup, as you like.		
<a href="#"><u>Untitled</u></a>	Protect the web services at <a href="80/443">80/443</a> , and SSH service at <a href="22">22</a> .		
<a href="#"><u>Untitled</u></a>	Make sure nmap test says the origin doesn't have <a href="22/80/443">22/80/443</a> opened.		
<a href="#"><u>Untitled</u></a>	Make sure an email alert will be sent when the origin is not reachable from Cloudflare.		
<a href="#"><u>Untitled</u></a>	Make sure <a href="web.yourdomain.com">web.yourdomain.com</a> has the following HA set up: All visitors will be routed to the primary origin. But once the primary origin is not reachable, it will fall back to the backup origin ( <a href="35.234.81.115">35.234.81.115</a> )		
<a href="#"><u>Security</u></a>	Any requests over HTTP needs to be redirected to HTTPS.		
<a href="#"><u>Untitled</u></a>	Make sure there's no path of HTTP traffic flows unencrypted.		
<a href="#"><u>Untitled</u></a>	Make sure you use Cloudflare's WAF and IP Intelligence.		

Aa Assignment	≡ Requirements	≡ Reference	≡ Verification
<u>Untitled</u>	Make sure you block anyone who sends more than 600 requests in a minute. Blocking time is 5 minutes.		
<u>Untitled</u>	Site Rule: <code>admin.yourdomain.com</code> should NOT allow any requests without the following request header 'admin: true'		
<u>Untitled</u>	Site Rule: <code>admin.yourdomain.com</code> should NOT allow any external access other than my own IP		
<u>Untitled</u>	Site Rule: <code>api.yourdomain.com</code> should NOT use Cloudflare's WAF and IP intelligence.		
<u>Untitled</u>	Site Rule: Anyone who accesses <code>web.yourdomain.com/country.html</code> from anywhere but your country or residence, should firstly pass JS Challenge.		
<u>Performance</u>	Check the current cache TTL and change it to 7 days.		
<u>Untitled</u>	Cache the front page for 1 minute.		
<u>Untitled</u>	Optimize the content as much as possible. Compress the images and codes.		
<u>Untitled</u>	What other settings do you need to enable to make the site faster? Find and enable them. (Hint: <a href="#">Refer to optimization training</a> )		
<u>API and Troubleshooting</u>	Set the WAF in "Detection-only" mode.		
<u>Untitled</u>	Set up Logpush for HTTP Request, and Spectrum Events.		
<u>Untitled</u>			