Cloudflare Accredited Configuration Engineer+

Technical Assignment

Version: Nov 2021
Content Owner: Chrisanthy Carlane
Technical Reviewer: Meng Xin, Jamal Boutkabout

Exam Information

Dear Cloudflare Partners,

Thank you for taking part in Cloudflare Accredited Configuration Engineer+ Practical Examination!

This practical examination is intended to evaluate your technical knowledge in implementing, optimizing and troubleshooting Cloudflare core solutions.
Bonus points will be awarded for completing optional section.

Exam Preparation:
Review ACE course material and lab guide.

You are required to complete the assignment with 75% passing score. Follow these instructions to answer your assignment:

- Create a doc/PDF file, label the file using this format: [Your Name] - [Your Organization] - ACE.PDF
- Number your answers, so that they correspond to the questions
- Explain your answer, if relevant, include a screenshot of your configuration
- Save your completed document in a shared folder (publicly available URL with Read permission)
- Go to **Test** section and paste the URL to the PDF doc at the answer section.
- Upon receiving your submission, respective Cloudflare PSE will review your assignment and mark the exam as Passed/Failed.

Reach out to cloudflareuniversity@cloudflare.com if you have any questions.

1. Account Creation (5 points)

You should have access to a Cloudflare Enterprise plan account with your official email address.

Origin Server:
Create a simple website, running on 2 origin web servers on a platform of your choosing. This could be in AWS, Google Cloud, Digital Ocean etc.
The web server must run an endpoint that returns all HTTP request headers in the body of the HTTP response.
The endpoint can be something that you have written yourself (e.g. in html, php, jsp, aspx, etc.).

Include an image and video file as part of your website content and place them under /static folder.

2. DNS & SSL/TLS configuration (5 points)

You are in charge of onboarding OrangeCloud Technology (OCT) applications to Cloudflare.

OCT is managing their authoritative DNS and wish to keep it that way.
OCT application team would like to use Cloudflare for 2 applications:

web.<domain.com>
app.<domain.com>

Once onboarded with Cloudflare:

A. Deploy a ECDSA origin cert on the origin and secure the communication between Cloudflare and the origin using SSL Full (Strict) mode

B. Describe the difference between SSL mode Flexible, Full and Full (strict)

C. Bonus (1 point): deploy a custom vanity NS server on your zone

3. Firewall (15 points)

A. Enable both the Cloudflare Managed Rulesets and the OWASP ruleset on the entire zone. Set OWASP threshold to low, with PL4 Paranoia level
B. Create a Firewall Rule to block request to app.<domain.com>/admin from any IP in Singapore, unless that request contains a cookie: "cf_bypass=true"
C. Create a Firewall Rule to allow requests going to the app.<domain.com>/admin to bypass WAF Managed Rulesets and Rate Limiting

4. Performance (30 points)

A. Ensure that all static html pages are cached at Cloudflare edge for 1 day
B. Ensure that all static files under /static directory should be cached at Cloudflare edge for 1 hour and in browser for 4 hours
C. Turn on webp conversion for pictures on Cloudflare and do a size comparison of both version of the picture, with and without webp
D. Show the steps to purge all cache under the folder app.<domain.com>/static
E. Upload an mp4 file to your webserver and create Page Rules according to the scenario below:
    a. *<your url>/<file>.mp4*
    b. *<your url>/<file>.mp4?a=123*
    c. *<your url>/<file>.mp4?a=123&b=234*
    d. *<your url>/<file>.mp4?a=234&b=234*

    1) Create one Page Rule to cache all of them
    2) Create one Page Rule to cache all of them and hit the same cache (i.e. after requesting the first URL, the second, third and fourth URL should be a cache hit)
    3) Bonus (3 points): Create one Page Rules where a and b is cached and hit the same cache, c and d is cached and hit the same cache. Show how you test to show that it hits the same cache (Hint: custom cache key)

Paste each configuration screenshot in your answer document.

5. User Administration & Log Analytics (5 points)

User Administration:

A. Show the screenshot where you can check which account member changed any Cloudflare configuration
B. Show the screenshot where you personalize any block and error pages for your domain

Logs and Analytics

C. Show the screenshot to set up Logpush jobs to any Logpush target for HTTP requests, Audit Log and Firewall events
D. Show the screenshot of Postman or curl to run API call and output to turn on Logpull for your zone.
E. Show the screenshot of Postman or curl to run API call to download your log for the previous 1 hour using Logpull, while redacting your authentication key

6. Client API (15 points)

A. Using Postman, GET the JSON response of the zone settings of your zone
B. Create an API token that can be used to purge cache at your domain only, valid for 1 week and can be executed from a device coming from your network only
C. Run the API call to purge all the cache from Cloudflare edge using the above API token

Add-on Product

7. Argo Smart Routing/Tiered Caching (Optional: 2 points)

A. Turn on Argo Smart routing and show the analytics of improved performance
B. Turn on Tiered Caching and set it to Smart Tiered Caching Topology

8. Rate Limiting (Optional: 4 points)

A. Create a policy to block any request going to https://api.<yourdomain.com> for 5 minutes if the request from the same IP Address exceeds 100 times

B. Customize the blocking message to this text message: “Please try again after 5 minutes”

9. Spectrum (Optional: 5 points)

A. Put an ssh server behind the Spectrum and change the edge port to 2022 (instead of 22).
B. Show login screen to that ssh server using the configured Spectrum hostname

10. Load Balancing (Optional: 10 points)

Set up a Cloudflare Load Balancing between 2 instances under lb.<domain.com> with these specifications:

a. Each instance will reside in different pool, name it pool1 and pool, with pool1 as the fallback pool
b. Configure a mechanism to check every 10 minutes if https://lb.<yourdomain>/monitor.txt is not accessible on port 443 within 10 seconds
c. Add a traffic steering to route traffic to the fastest pool based on measured latency from health checks

CLOUDFLARE

END

Fill in the link to your documentation in the Exam section