ACESSE O LABORATÓRIO

ABRIR AMBIENTE DO LABORATÓRIO

```
https://labs.cloudflare.com/
```

Guia oficial do exercício

```
https://labs.cloudflare.com/c/guide/app-services-imp#dns
```

Começando

Bem-vindo ao laboratório Implementar: Serviços de Aplicativos. Para realizar os exercícios, utilize a conta Cloudflare e a zona DNS fornecidas para o laboratório.

PRÉ-REQUISITOS

1. Uma conta Cloudflare
2. Uma zona DNS hospedada na Cloudflare com os recursos apropriados

DICA DE NAVEGAÇÃO

Use a lupa na barra superior ou pressione Ctrl + K no Linux e Windows, ou ? + K no Mac, para localizar rapidamente produtos no painel.

Digite o domínio raiz do seu laboratório.
Clique em Aplicar. Só o domínio muda; caminhos e parâmetros são preservados.

DOMÍNIO DO LABORATÓRIO

```
opportunistic-zipline.sxplab.com
```

Aplicar

Vamos começar!

Crie os dois labs antes de começar

Após acessar o ambiente pelo link da página anterior, localize os dois cursos abaixo e selecione Create lab. Os dois ambientes serão usados durante o treinamento.

1 CONFIGURAÇÃO

Implement: Application Services

Lab for Cloudflare University course

Course Code: AS301

CREATE LAB

Use este lab para trabalhar nas configurações de Application Services.

2 TESTES E VALIDAÇÃO

Operate: Cloudflare One

Lab for Cloudflare University course

Course Code: ZT401

CREATE LAB

Use este lab para executar os testes e validar o ambiente Cloudflare One.

COMO OS LABS SERÃO UTILIZADOS

Crie os dois labs antes de iniciar. Use o AS301 para realizar as configurações de Application Services e mantenha o ZT401 disponível para testar e validar as configurações durante os exercícios.

ROTEIRO DO LABORATÓRIO

1. **DNS**
   Configuração e validação de registros

2. **TLS (SSL)**
   Certificados, HTTPS e subdomínios

3. **WAF**
   Managed Rules, Custom Rules e Rate Limiting

4. **Bot Management**
   Modelos e validação de tráfego automatizado

5. **Cache**
   Cache Rules, TTL e status code

6. **Load Balancing**
   Pools, monitores e session affinity

7. **Rules**
   Transform Rules e URL Rewrite

Use os marcadores do PDF para navegar diretamente entre os módulos.

MÓDULO 01

DNS

Configuração e validação de registros

Conteúdo original: páginas 1 a 4

D PATH: Websites > Select your zone

DNS

Tarefa

Para estes exercícios, você usará um aplicativo web em execução no endereço 20.88.188.200 . Como primeiro passo, configure um registro DNS para este aplicativo.

Você pode encontrar mais informações sobre o DNS da Cloudflare em nossa documentação para desenvolvedores de DNS .

Por que

- O DNS permite que você acesse serviços da web usando nomes de domínio fáceis de lembrar em vez de endereços IP.
- Mas, mais importante ainda, a maioria dos Serviços de Aplicativos da Cloudflare opera por meio de proxy reverso , que é habilitado em um registro DNS.
- Isso torna a posse de um registro DNS um pré-requisito necessário para poder utilizar esses serviços.

Passos

1. Configuração de zona aberta

No seu painel do Cloudflare, navegue até **Websites** e selecione a zona com a qual deseja trabalhar durante estes laboratórios.

Se você estiver usando uma conta Cloudflare que lhe foi fornecida, você já deverá ter uma zona provisionada para você.

2. Adicione um registro DNS

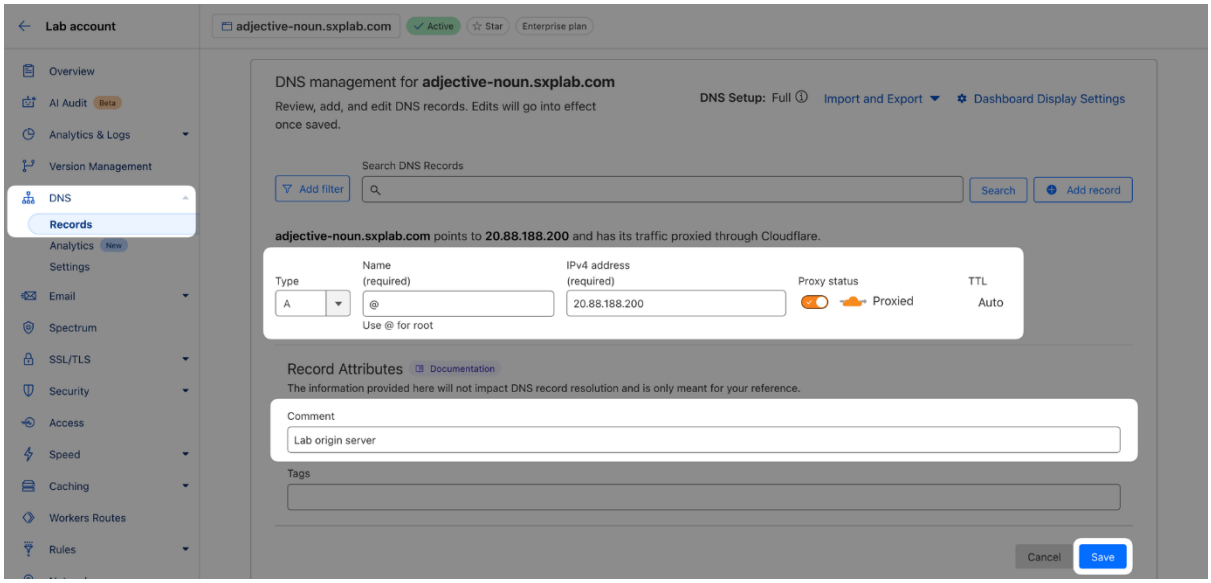
Dentro da sua zona, selecione **DNS ▸ Registros** e clique em **Adicionar registro** .

Dependendo do laboratório que você estiver usando, pode já existirem alguns registros DNS. Você pode ignorá-los por enquanto e prosseguir com a adição de um novo registro seguindo as instruções abaixo.

Crie um novo registro A que aponte a raiz da sua zona para o nosso servidor de origem do laboratório. Certifique-se de manter a opção **"Proxy" ativada** para que o Cloudflare execute o proxy reverso neste domínio.

O Cloudflare também permite a adição de comentários aos registros DNS para facilitar a referência, como um ID de solicitação de alteração. Sinta-se à vontade para adicionar um comentário ao seu registro.

```
Type: A
Name: @
IPv4 address: 20.88.188.200
Proxy: yes
Comment: Lab origin server
```



3. Teste

Abra o terminal e consulte o registro DNS do seu domínio raiz:

Observe que nossos exemplos de laboratório usam "zipline oportunista" (ou "adjetivo-substantivo ") como domínio de exemplo; sempre substitua esses termos pelo domínio real do seu laboratório.

macOS/Linux

```
dig opportunistic-zipline.sxplab.com A
```

```
; <<>> DiG 9.10.6 <<>> +noquestion +nocomments adjective-noun.sxplab.com A
;; global options: +cmd
organized-keystone.sxplab.com. 300      IN      A       104.18.24.205
organized-keystone.sxplab.com. 300      IN      A       104.18.25.205
```

Windows

```
nslookup opportunistic-zipline.sxplab.com
```

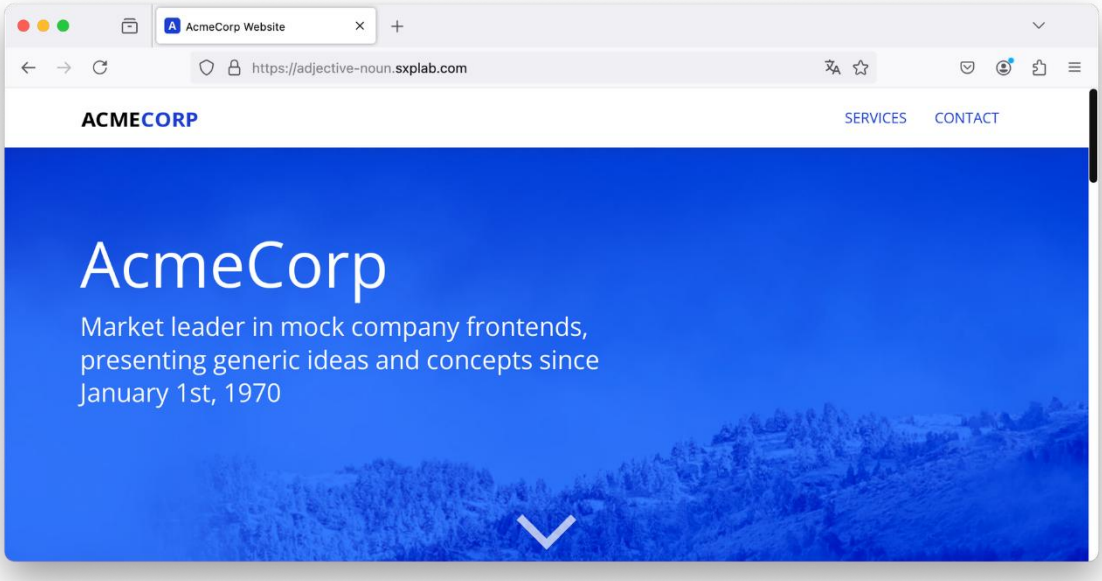
```
Server:         127.0.2.2
Address:        127.0.2.2#53

Non-authoritative answer:
Name:   organized-keystone.sxplab.com
Address: 104.18.24.205
Name:   organized-keystone.sxplab.com
Address: 104.18.25.205
```

Embora o servidor de origem 20.88.188.200 esteja em execução no Azure, você deverá ver dois endereços IP do Cloudflare retornados (você pode verificar a organização proprietária através do whois.com).

Isso ocorre porque o Cloudflare atua como um proxy reverso para todas as solicitações ao novo domínio — ele encerra a sessão HTTP do usuário e, em seguida, estabelece uma nova sessão com a origem.

Por fim, teste se você consegue acessar o aplicativo web usando o registro DNS que acabou de criar. Abra seu navegador e digite o nome do seu domínio na barra de endereços. Você deverá ver a página inicial do nosso aplicativo de laboratório.



Resumo

Nesta seção, configuramos os registros DNS necessários para concluir os exercícios de laboratório.

Em seguida, vamos dar uma olhada no TLS (SSL).

MÓDULO 02

TLS (SSL)

Certificados, HTTPS e subdomínios

Conteúdo original: páginas 5 a 11

TLS (SSL)

Tarefa

Inspecione o certificado SSL universal padrão que a Cloudflare criou para o seu domínio, crie um subdomínio que não esteja coberto por ele e solicite um certificado avançado para esse subdomínio.

Você pode encontrar mais informações sobre o DNS da Cloudflare em nossa documentação para desenvolvedores de SSL .

Por que

- A Cloudflare emite automaticamente um certificado universal para todos os domínios adicionados à Cloudflare.
- Este certificado abrange o domínio raiz e um nível de subdomínios por meio de um certificado wildcard
- E se você precisar de uma estrutura de domínio mais complexa? Nesses casos, você pode solicitar um Certificado Avançado.
- Os Certificados Avançados permitem a personalização de configurações adicionais, como os cipher suites

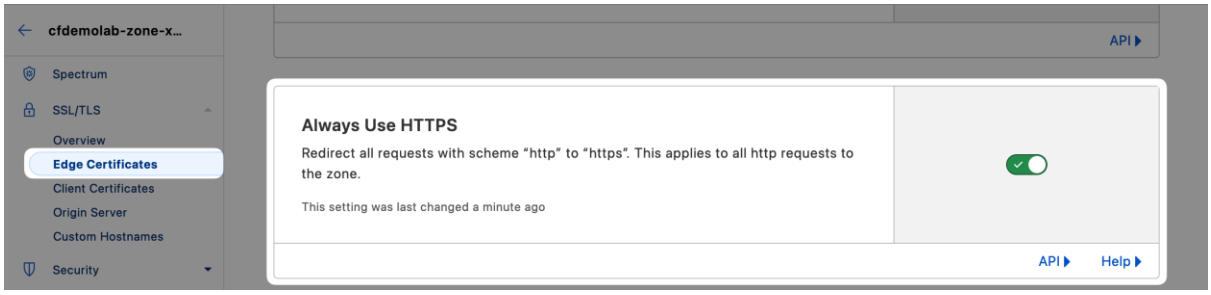
Passos

Antes de prosseguir com este laboratório, certifique-se de desativar quaisquer serviços em execução em sua máquina que possam interceptar o tráfego HTTPS (VPNs, Cloudflare WARP, etc.).

1a. Use sempre HTTPS

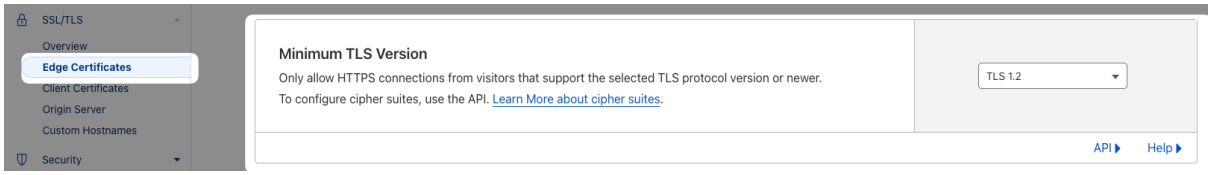
Antes de começarmos a analisar os certificados TLS, vamos impor o uso de HTTPS em nossa zona. Isso ajudará a evitar solicitações acidentais via HTTP que não utilizariam TLS.

Abra sua zona no painel do Cloudflare e selecione **SSL/TLS ▸ Certificados de borda** . Aqui, ative a opção **Sempre usar HTTPS** .



1b. Alterar a versão mínima do TLS

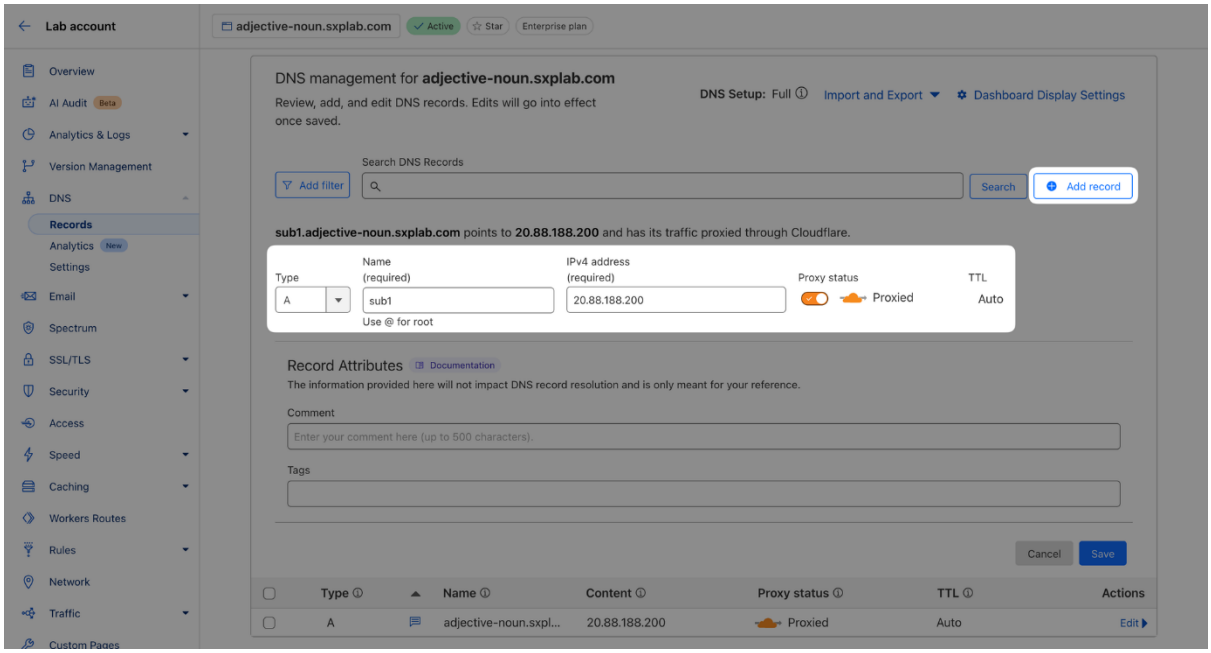
O TLS 1.0 é a configuração de zona padrão. Isso é mantido para fins de compatibilidade, mas é recomendável alterá-lo na maioria dos casos. Role um pouco para baixo a partir da configuração anterior e defina **a Versão mínima do TLS** para **TLS 1.2.**



2. Adicionar um subdomínio

Em seguida, acesse **DNS ▸ Registros** e crie um subdomínio de nível 1; neste exemplo, vamos chamá-lo de sub1.

Para os propósitos deste exercício, não importa realmente para onde esse subdomínio aponte. Por uma questão de simplicidade, vamos usar o mesmo servidor de origem para o qual o domínio raiz aponta: 20.88.188.200.



Acesse esse novo subdomínio (por exemplo, https://sub1.opportunistic-zipline.sxplab.com) no seu navegador e o aplicativo web deverá abrir normalmente, pois está coberto pelo certificado universal.

```
https://sub1.opportunistic-zipline.sxplab.com
```

Se você tiver o OpenSSL instalado, poderá consultar os detalhes do certificado de domínio com o seguinte comando:

```
openssl s_client -connect sub1.opportunistic-zipline.sxplab.com:443 | openssl x509 -noout -ext
subjectAltName
```

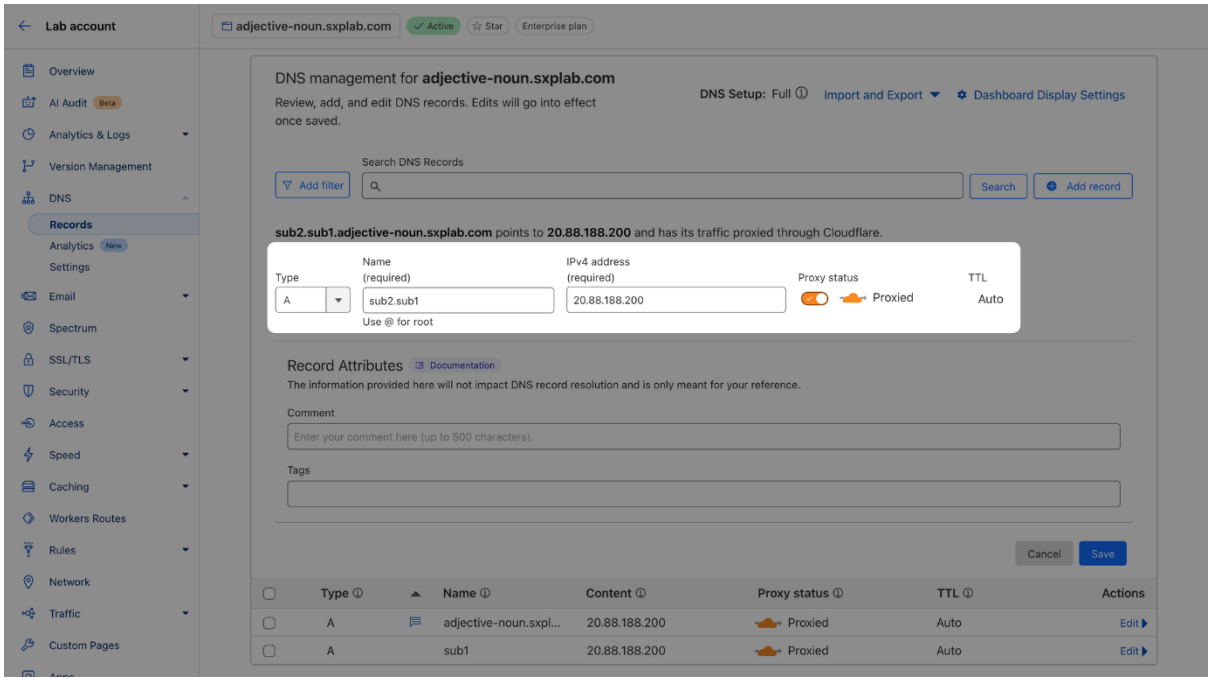
```
depth=2 C=US, O=Google Trust Services LLC, CN=GTS Root R4
verify return:1
depth=1 C=US, O=Google Trust Services, CN=WE1
verify return:1
depth=0 CN=adjective-noun.sxplab.com
verify return:1
X509v3 Subject Alternative Name:
    DNS:adjective-noun.sxplab.com, DNS:*.adjective-noun.sxplab.com
```

Se o resultado não corresponder ao acima, certifique-se de que sua VPN ou Cloudflare WARP esteja desativada.

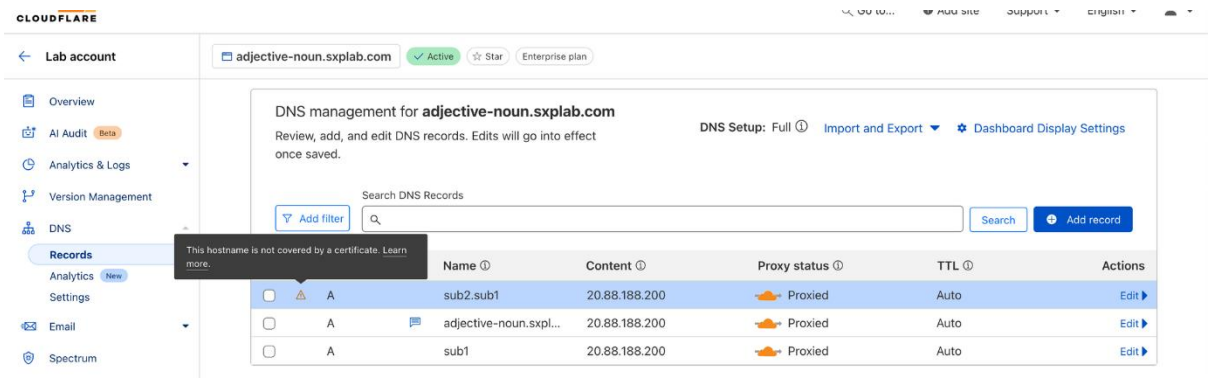
Para relembrar, este é o certificado universal que a Cloudflare emitiu automaticamente para o seu domínio raiz e seus subdomínios diretos quando o domínio foi criado. Anote a entidade emissora do certificado, neste caso, o Google.

3. Adicione um subdomínio a esse subdomínio.

Em seguida, adicione um subdomínio ao subdomínio que você acabou de criar; neste exemplo, vamos chamá-lo de sub2, então o nome será sub2.sub1.



Observe o aviso exibido ao lado deste subdomínio de nível 2. Isso ocorre porque esse subdomínio não está mais coberto pelo certificado universal.



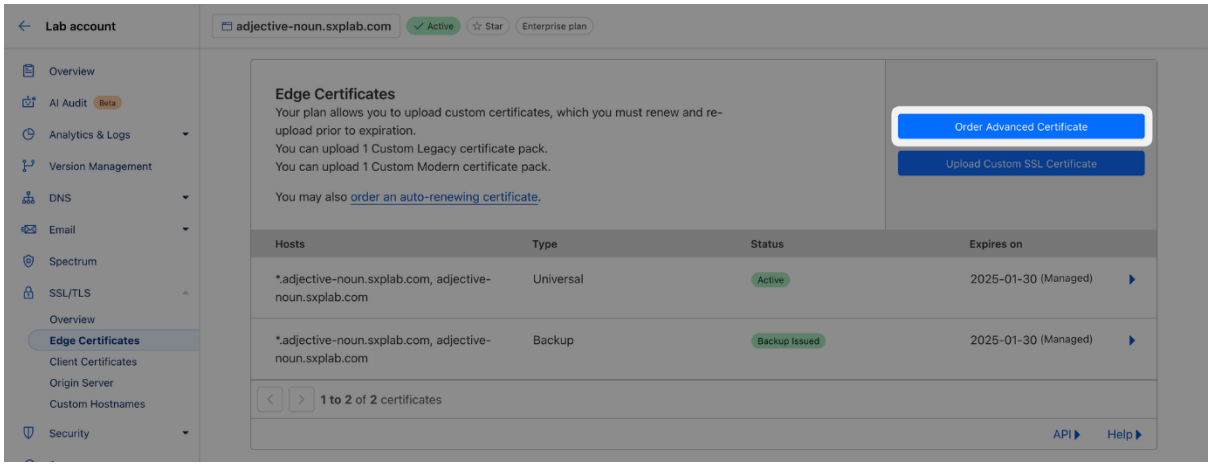
Se você tentar abrir este novo subdomínio em um navegador da web, receberá um erro informando que seu navegador não pode estabelecer uma conexão segura.

```
openssl s_client -connect sub2.sub1.opportunistic-zipline.sxplab.com:443
```

```
CONNECTED(00000006)
40F8F7F201000000:error:0A000410:SSL routines:ssl3_read_bytes:ssl/tls alert handshake
---
no peer certificate available
---
...
```

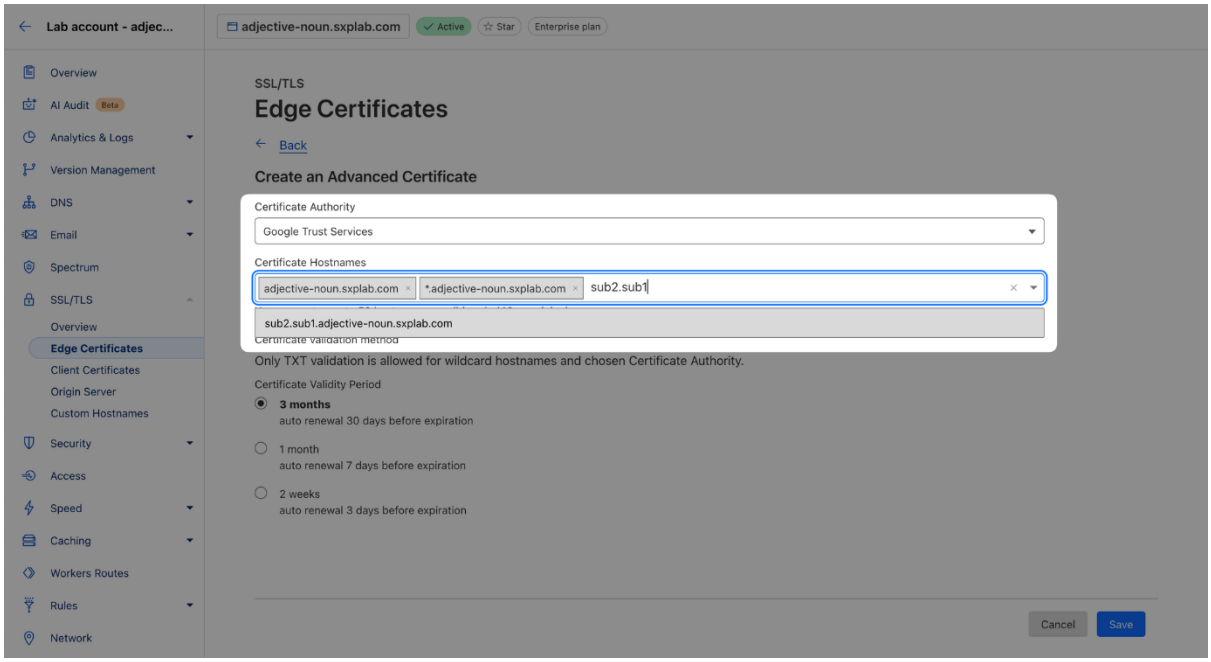
4. Solicitar Certificado Avançado

Acesse SSL/TLS ▸ Certificados de Borda e selecione Solicitar Certificado Avançado .



Você pode usar qualquer Autoridade Certificadora; neste exemplo, usaremos o Google Trust Services.

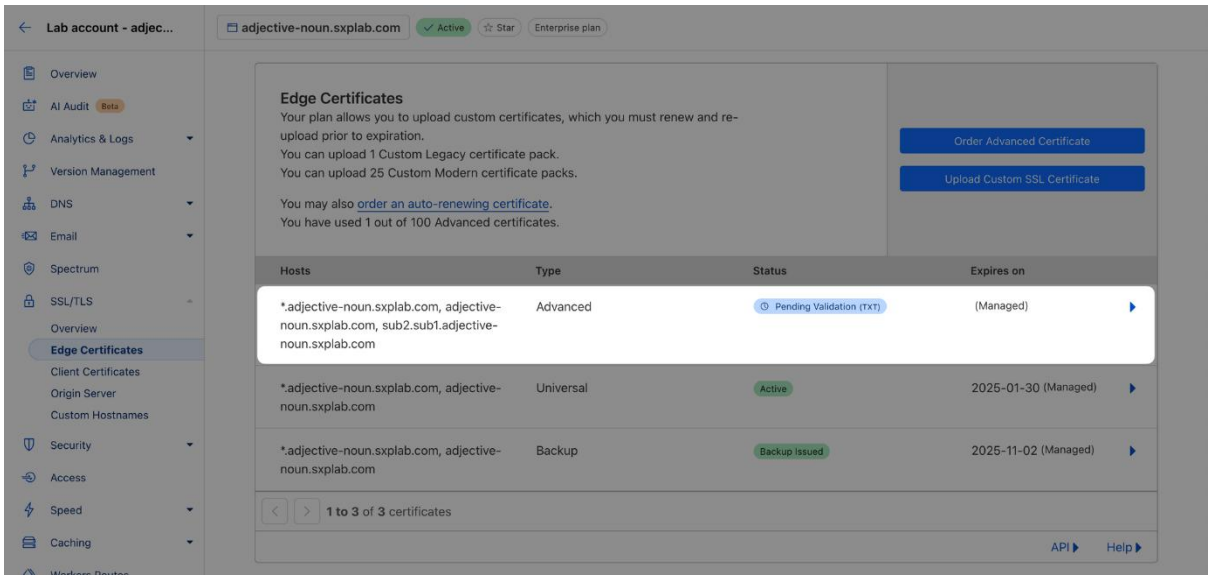
Digite seu subdomínio (ex: sub2.sub1) no campo **"Nomes de host do certificado"** e selecione a opção de preenchimento automático com o nome de domínio completo quando ela aparecer.



Você pode, opcionalmente, remover os nomes de host existentes da lista se desejar que seu certificado cubra apenas este novo subdomínio de nível 2. No entanto, se fizer isso, certifique-se de manter o método de validação como **Validação TXT** .

Contanto que seu domínio esteja totalmente configurado , o Cloudflare cuidará da Validação de Controle de Domínio automaticamente em seu nome ao usar a Validação TXT.

Você deverá ver agora um novo certificado com o status "Validação Pendente".



O processo de validação pode levar alguns minutos para ser concluído, então agora seria um bom momento para pegar uma bebida refrescante de sua preferência.

Após a conclusão do processo de validação, acesse o novo subdomínio novamente com seu navegador e você deverá vê-lo carregar com segurança.

Se você inspecionar o certificado novamente com o OpenSSL, deverá ver o novo subdomínio agora coberto:

```
openssl s_client -connect sub2.sub1.opportunistic-zipline.sxplab.com:443 | openssl x509 -noout -ext
subjectAltName
```

```
depth=2 C=US, O=Google Trust Services LLC, CN=GTS Root R4
verify return:1
depth=1 C=US, O=Google Trust Services, CN=WE1
verify return:1
depth=0 CN=adjective-noun.sxplab.com
verify return:1
X509v3 Subject Alternative Name:
    DNS:adjective-noun.sxplab.com, DNS:*.adjective-noun.sxplab.com, DNS:sub2.sub1.ad
```

Anote novamente a entidade emissora. Selecionamos o Google Trust Services ao solicitar o certificado anteriormente, e essa entidade emissora corresponde à exibida na saída.

A lista de Subject Alternative Name (SAN) será diferente se você remover nomes de host existentes ao solicitar o certificado, mas ela sempre deverá incluir seu subdomínio de nível 2.

Resumo

Nesta seção, entendemos e configuramos certificados TLS (SSL).

Em seguida, vamos analisar o Firewall de Aplicação Web (WAF).

MÓDULO 03

WAF

Managed Rules, Custom Rules e Rate Limiting

Conteúdo original: páginas 12 a 26

WAF

Tarefa

Nesta seção, implantaremos os principais componentes do Cloudflare Application Security, incluindo:

- Conjuntos de regras WAF
- Regras personalizadas
- Limitação de taxa

Por que

O Cloudflare Managed Ruleset é um conjunto abrangente de regras de segurança criado e mantido pela equipe de segurança da Cloudflare. Ele foi projetado para fornecer proteção rápida e eficaz para todos os tipos de aplicativos e é atualizado frequentemente para abranger novas vulnerabilidades e reduzir falsos positivos.

Ao seguir estes exercícios, você aprenderá como implementar esses recursos para proteger seus aplicativos da web.

Passos

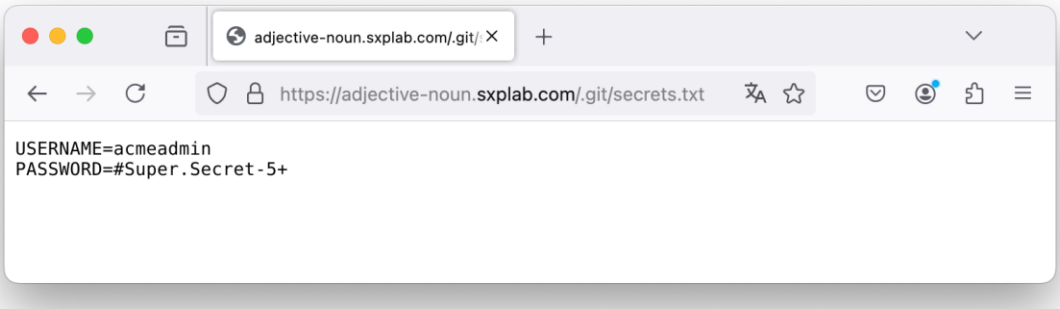
1. O que podemos visualizar antes de implantar o WAF?

O Cloudflare WAF oferece proteção abrangente pronta para uso imediato. Um problema comum em aplicações web é a exposição acidental de arquivos e pastas confidenciais.

Para testar isso, tente navegar até o seguinte endereço /.git/secrets.txt na sua zona Cloudflare:

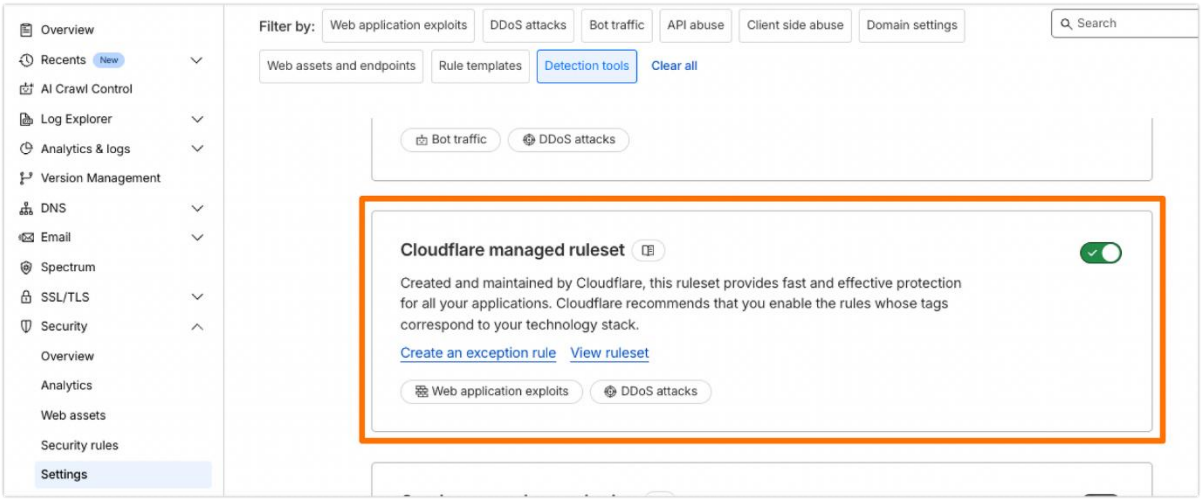
```
https://opportunistic-zipline.sxplab.com/.git/secrets.txt
```

Isso simula o cenário de um repositório de controle de versão Git exposto. Oh não, as credenciais estão expostas! Voltaremos a isso mais tarde.



2. Implante o Cloudflare Managed Ruleset.

No painel de controle do Cloudflare, acesse **Segurança ▸ Configurações ▸ Ferramentas de detecção** e ative o Cloudflare Managed Ruleset

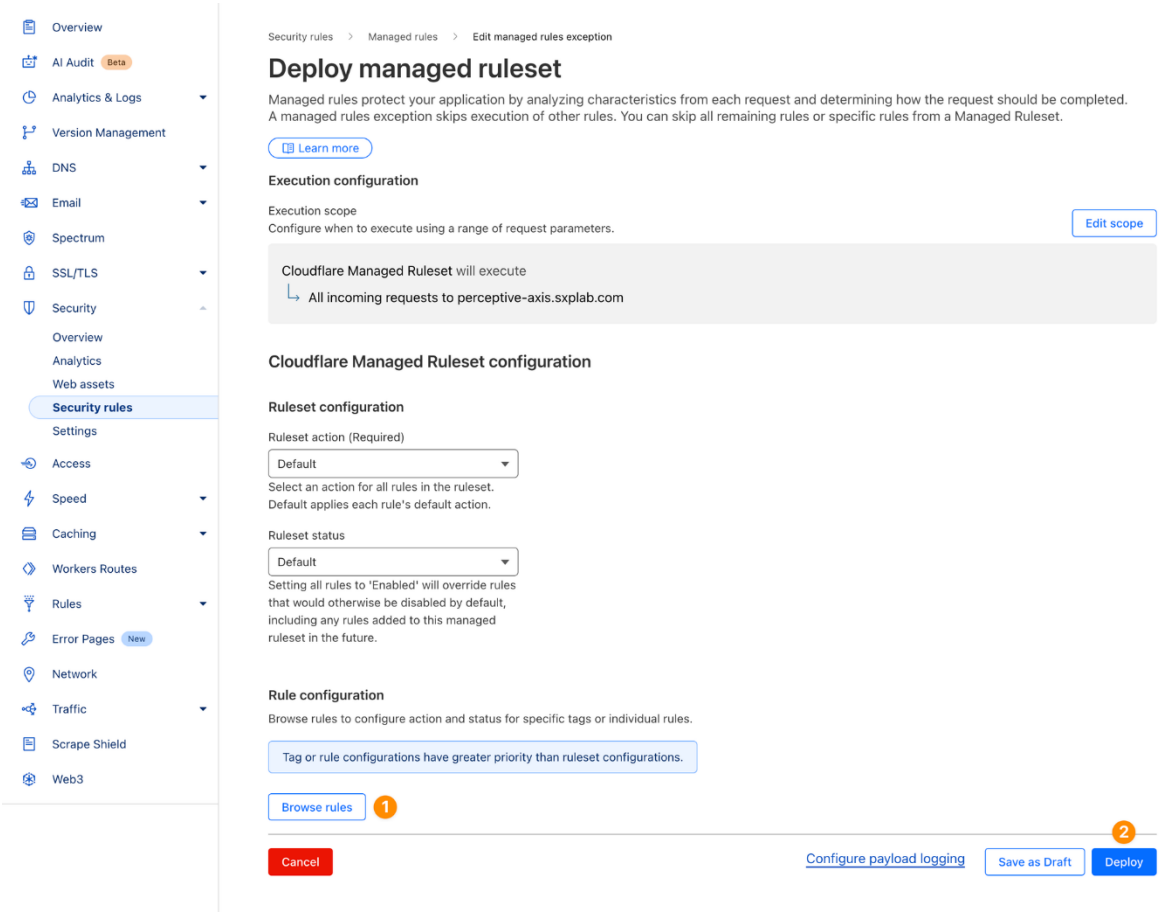


A tela de configurações "Implantar Cloudflare Managed Ruleset será exibida. Nessa tela, você poderá configurar:

- Escopo de execução com expressão de filtro personalizada para aplicar o conjunto de regras apenas a um subconjunto de solicitações recebidas.
- Defina a ação a ser executada (Bloqueio, Managed Challenge, JS Challenge, Registro, Interactive Challenge ou Padrão)
- Navegue e desative regras específicas ou regras com tags específicas.
- Substitua a ação executada por regras individuais ou regras com tags específicas.

Selecione " **Navegar nas regras"** para ver os diferentes tipos de vulnerabilidades mitigadas automaticamente.

Por enquanto, vamos deixar tudo como padrão. Quando estiver pronto, volte para a página de revisão e clique em **Implantar** .



Agora você verá que o conjunto de regras está ativado.

Repita esta ação para também implantar o Cloudflare OWASP Core Ruleset com as configurações padrão. Como antes, sinta-se à vontade para explorar as regras e configurações.

Se você quiser entender melhor os conjuntos de regras e suas várias configurações, consulte a documentação para desenvolvedores do Cloudflare WAF .

3. Teste o Cloudflare Managed Ruleset

No início do exercício, você pôde navegar pela .git pasta na sua zona do Cloudflare, que revelou um arquivo de segredos:

```
https://opportunistic-zipline.sxplab.com/.git/secrets.txt
```

Repita este teste — você verá que simplesmente implantar o conjunto de regras com as configurações padrão impediu esse ataque.

Agora você verá que o conjunto de regras está ativado.

Repita esta ação para também implantar o Cloudflare OWASP Core Ruleset com as configurações padrão. Como antes, sinta-se à vontade para explorar as regras e configurações.

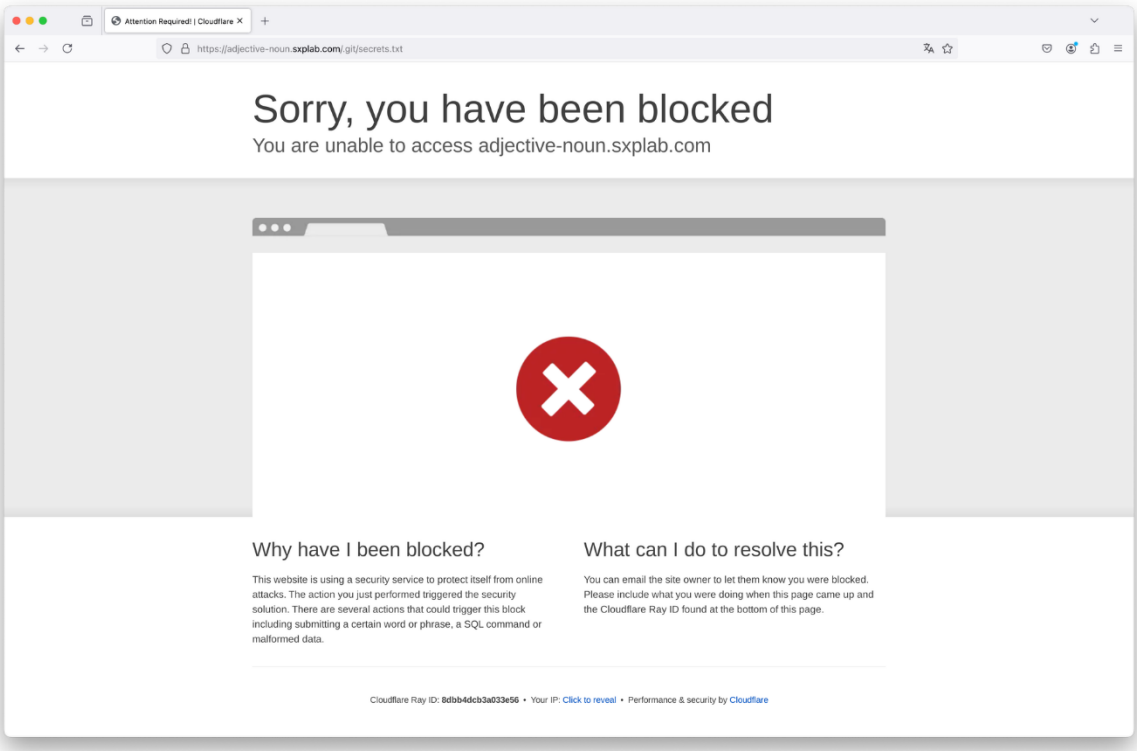
Se você quiser entender melhor os conjuntos de regras e suas várias configurações, consulte a documentação para desenvolvedores do Cloudflare WAF .

3. Teste o Cloudflare Managed Ruleset

No início do exercício, você pôde navegar pela .git pasta na sua zona do Cloudflare, que revelou um arquivo de segredos:

```
https://opportunistic-zipline.sxplab.com/.git/secrets.txt
```

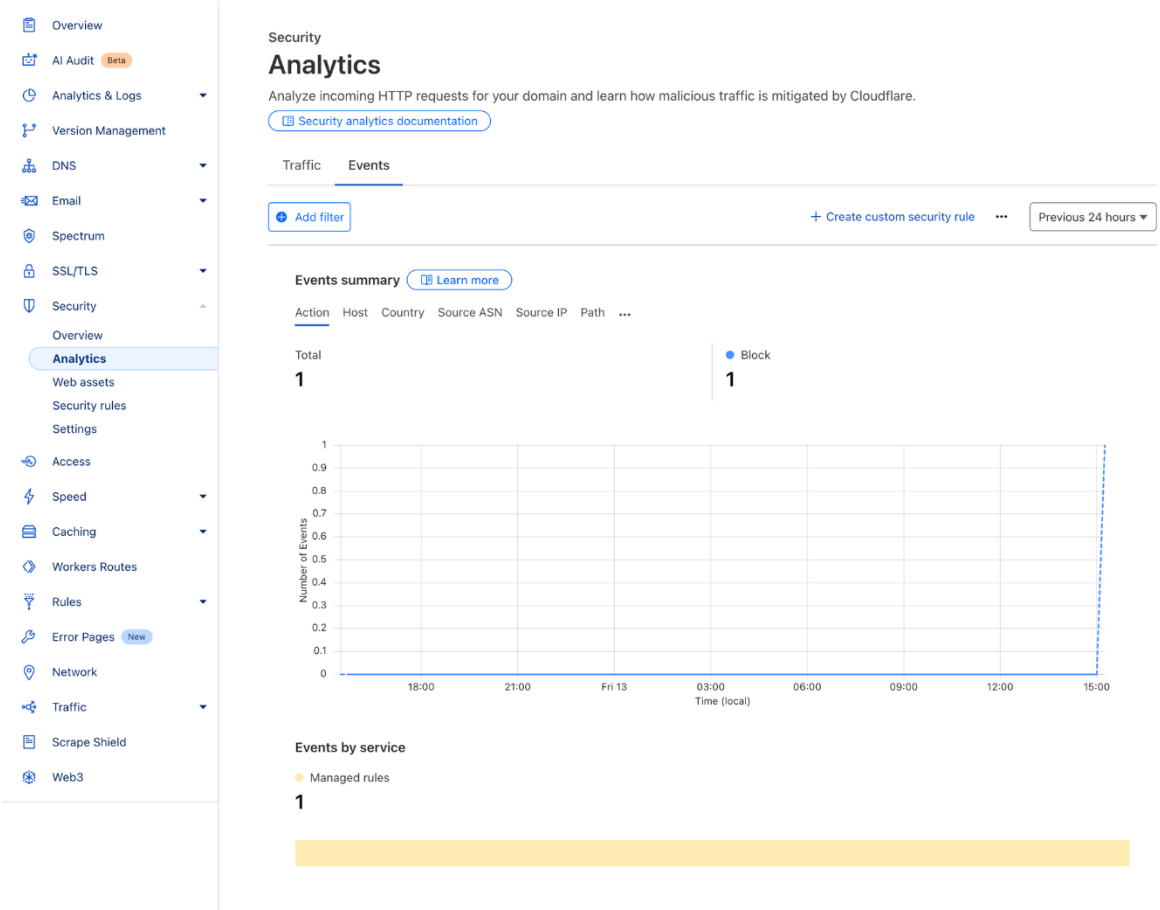
Repita este teste — você verá que simplesmente implantar o conjunto de regras com as configurações padrão impediu esse ataque.



4. Analise o bloqueio

Acesse **Segurança > Análises** e selecione a guia **Eventos** .

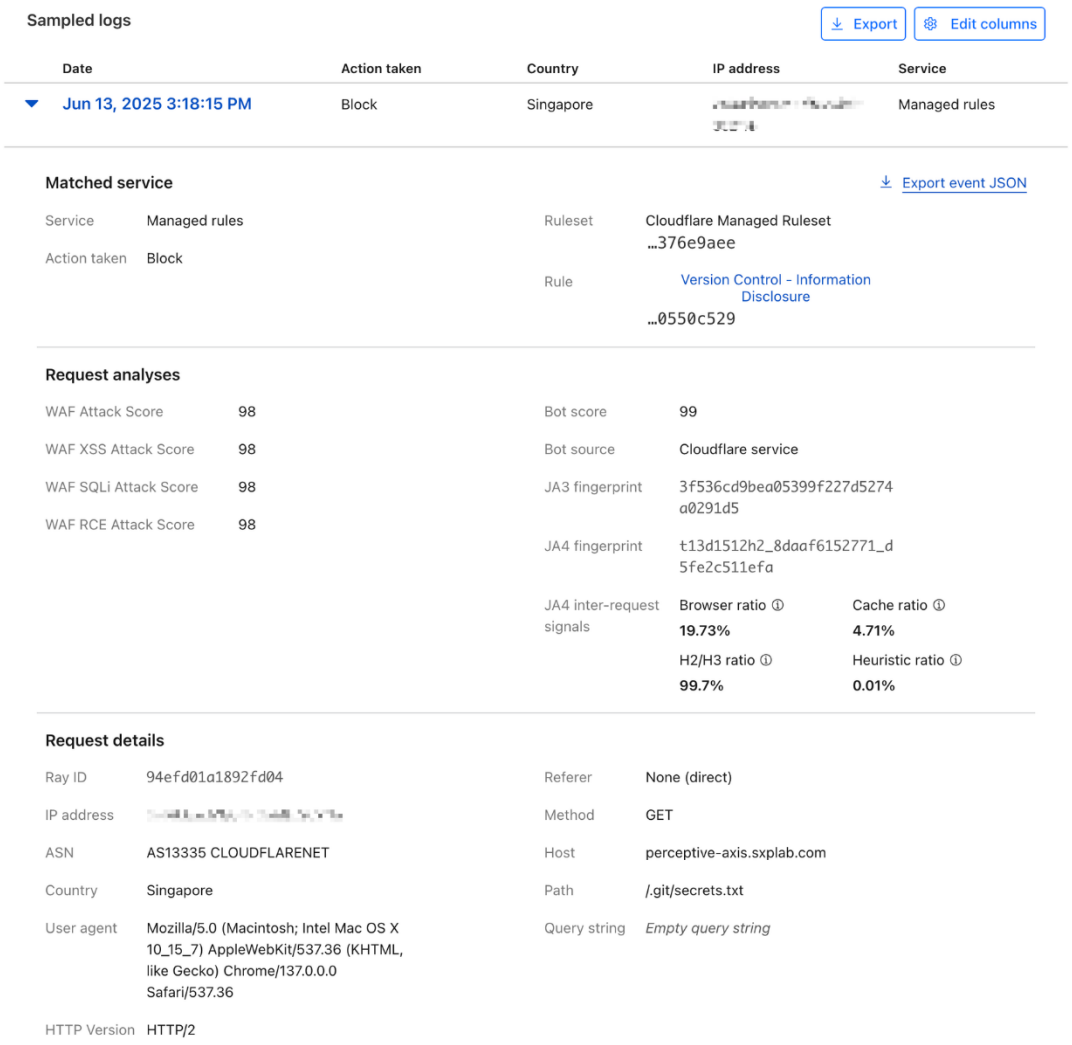
Você verá alguns bloqueios recentes, causados pelas etapas do exercício anterior (pode levar um minuto para que seu evento mais recente apareça):



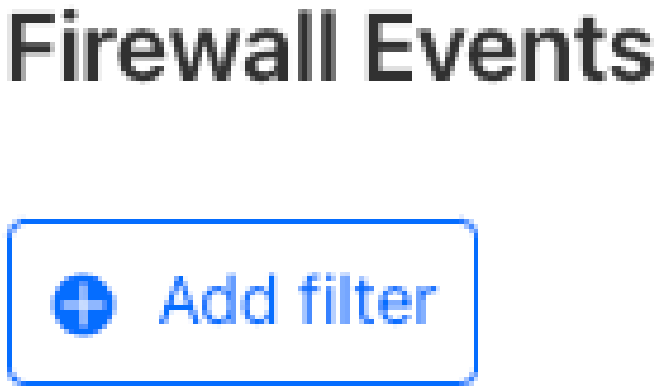
Dedique algum tempo a analisar as várias opções disponíveis. Observe como é possível visualizar o serviço associado ao bloqueio

Deslize a tela para baixo para ver mais detalhes sobre os eventos. Observe que você pode ver o que acionou a regra, o endereço IP de origem e assim por diante.

Por fim, expanda uma das linhas no **Log de Atividades** — isso fornece ainda mais detalhes. Observe como você pode ver o Conjunto de Regras e o ID da Regra Específica, juntamente com informações mais detalhadas sobre a solicitação. Isso é muito útil para analisar os gatilhos do WAF.



Tente adicionar um filtro que corresponda apenas a esta regra para que você possa ver todos os eventos associados a esta regra de exemplo.



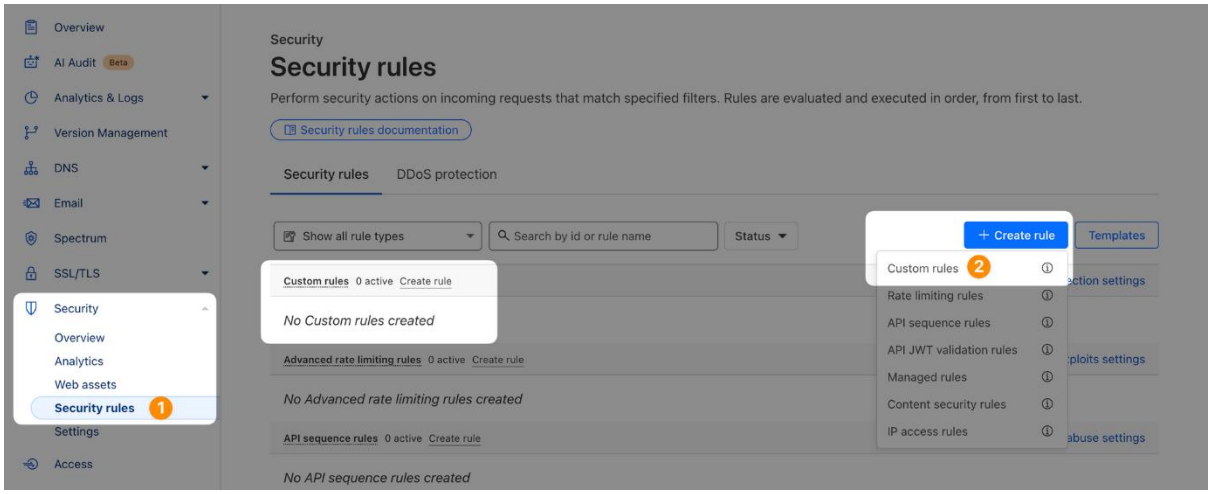
5. Crie uma regra personalizada

Antes de prosseguir com esta seção, recomendamos que você dê uma olhada rápida na documentação dos operadores de regras do Cloudflare, caso ainda não esteja familiarizado.

As regras personalizadas permitem controlar o tráfego de entrada, filtrando as solicitações para uma zona específica. Você pode executar ações como Bloquear ou Managed Challengeem solicitações de entrada, de acordo com as regras definidas por você.

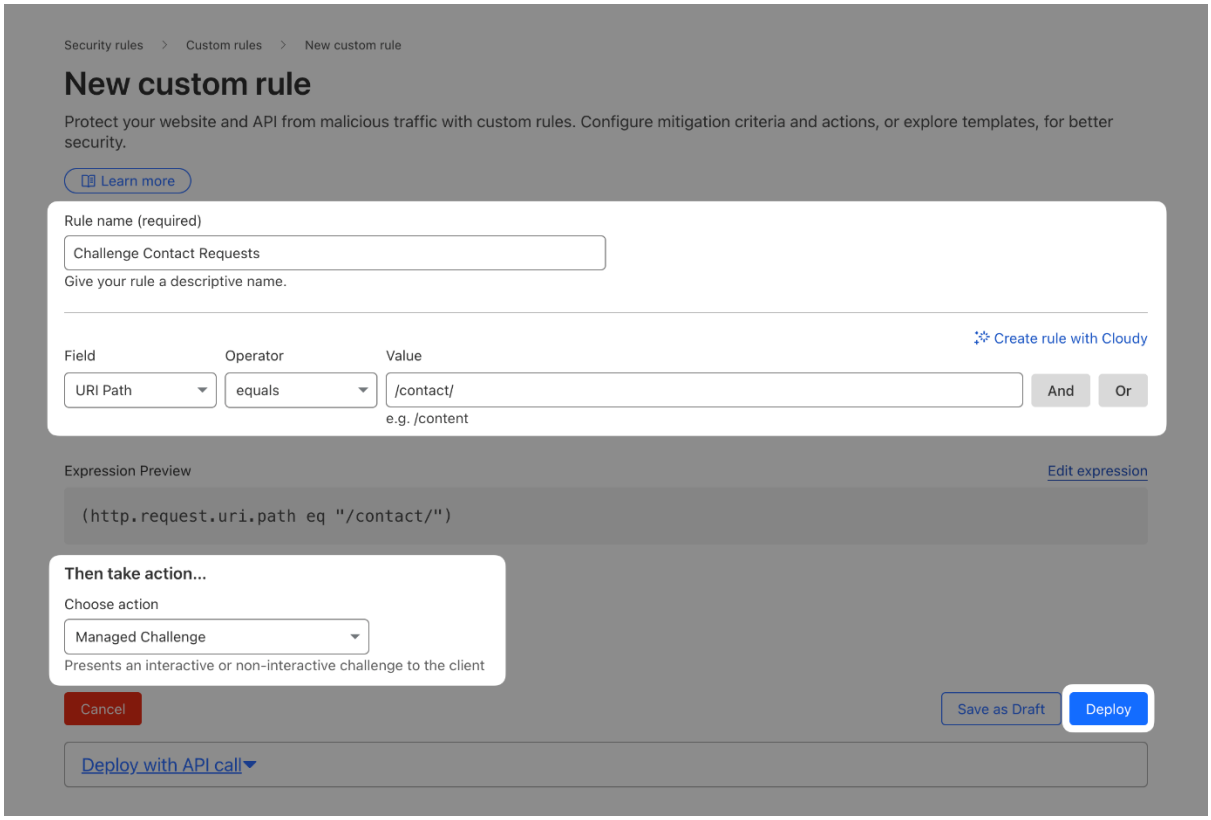
Nesta tarefa, criaremos uma regra personalizada simples para apresentar um Managed Challenge aos usuários que tentam acessar a página de contato da AcmeCorp.

No painel de controle do Cloudflare, acesse **Segurança ▸ Regras de segurança ▸ Criar regra ▸ Regras personalizadas** .



- Insira um nome descritivo para a regra, como "Desafiar solicitações de contato".
- Defina o **campo** como **Caminho URI.**
- Defina o **operador** como **igual a**
- Defina o **valor** para **/contact/**
- Na lista suspensa **Escolher ação** , selecione **Managed Challenge**

Sua configuração deve ser semelhante a esta:

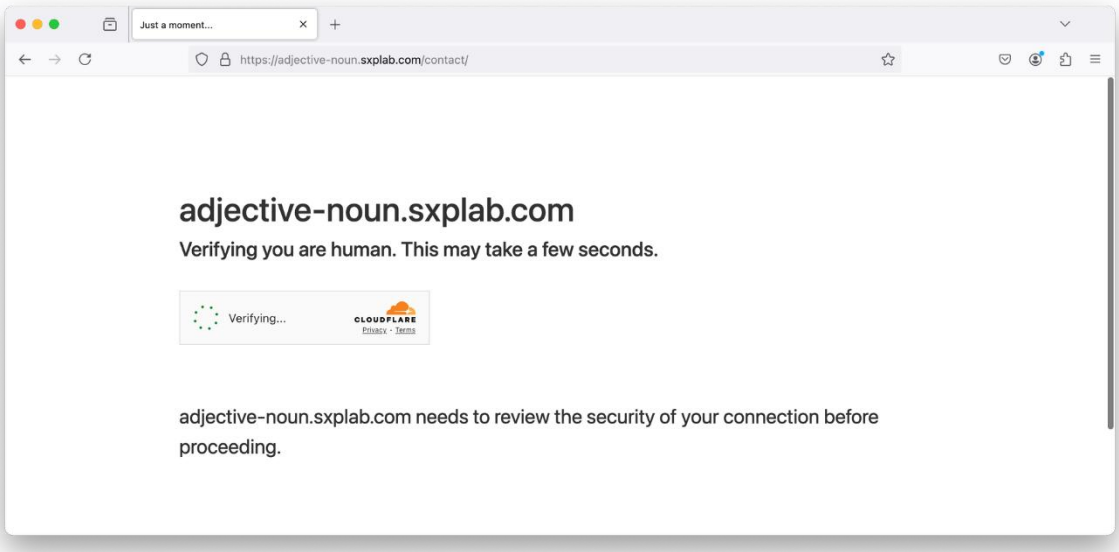


- **Implante** a regra

Sua regra agora está ativa e qualquer solicitação à página de contato do seu site acionará um Managed Challenge

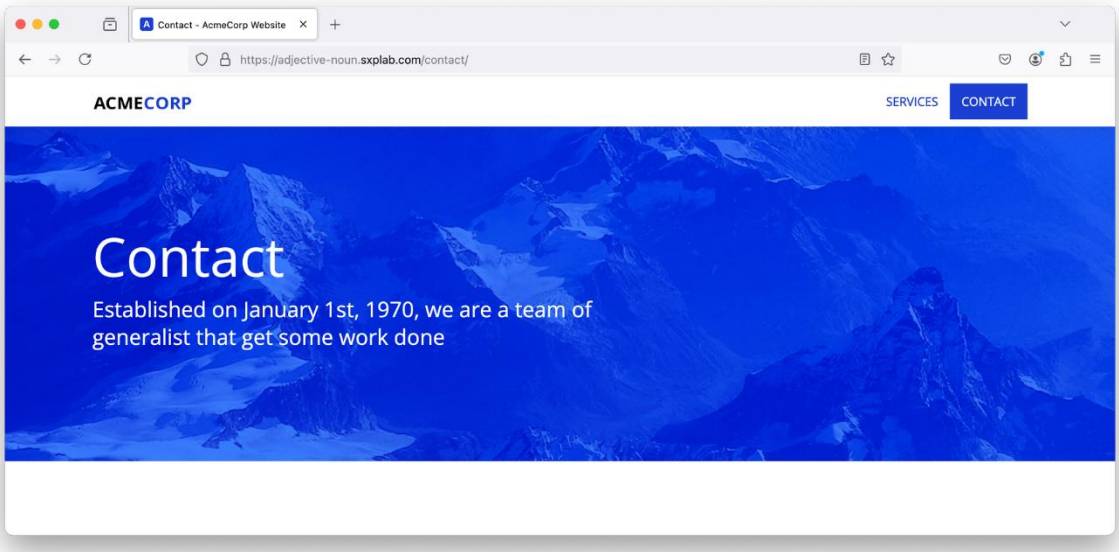
Acesse sua página de Contato - você deverá ver o Managed Challenge

Se você não visualizar o Managed Challenge tente uma guia anônima/privada ou um navegador diferente.



Normalmente, um navegador moderno e um usuário real serão redirecionados para a página solicitada. Se, por algum motivo, você for solicitado a verificar que é humano, marque a caixa.

Você verá agora a página de contato da AcmeCorp.

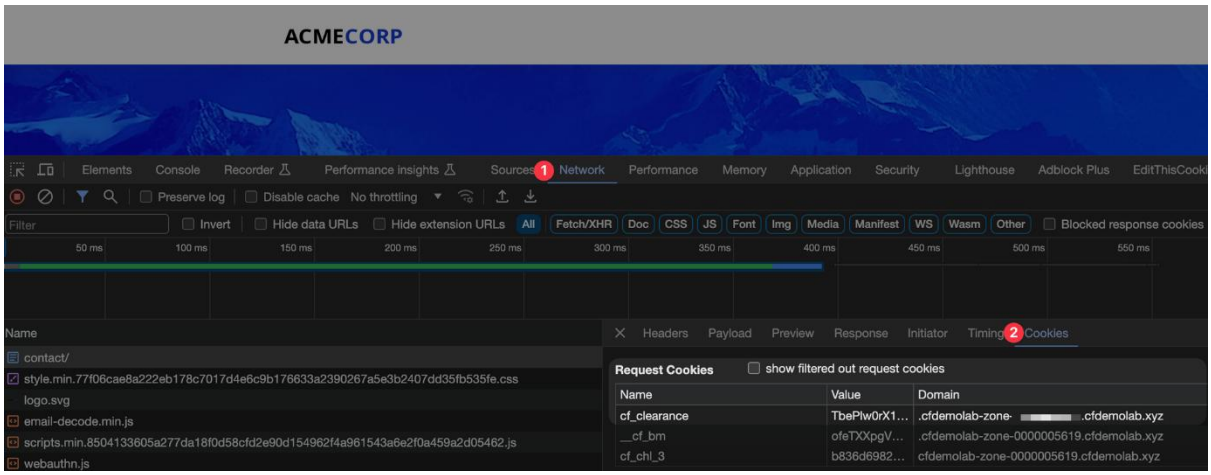


Tente a mesma requisição com curl. O que você vê? Por quê?

Antes de encerrarmos esta seção, abra o console do desenvolvedor. Vamos dar uma olhada no que acontece no navegador quando você supera um desafio.

No Console do Desenvolvedor, acesse a guia **Rede** e selecione **Cookies.**

Quando um desafio é resolvido, o cookie **cf_clearance** é enviado ao navegador na resposta. Isso impede desafios repetidos, mas também pode significar que você não verá o desafio quando esperar! Consulte a dica anterior sobre como usar uma sessão de navegação anônima ou um navegador diferente para acionar o Managed Challenge pois você ainda não terá o cookie nessa sessão.

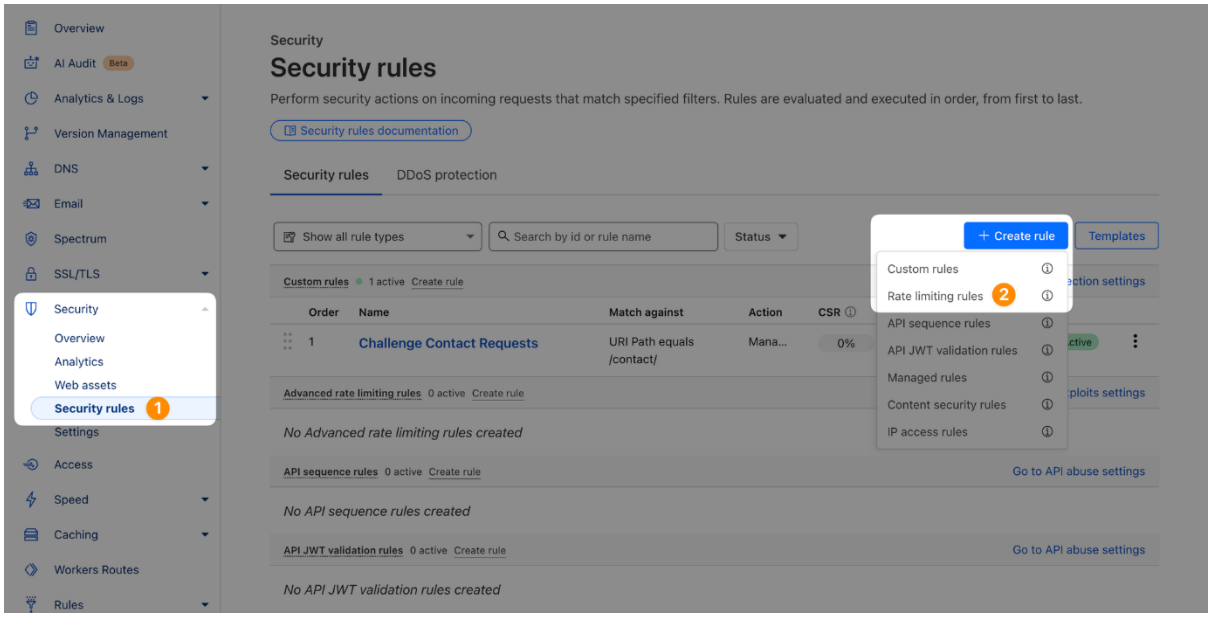


Se desejar saber mais sobre os diferentes casos de uso para regras personalizadas, consulte Casos de uso comuns para regras personalizadas .

6. Crie uma regra de limitação de taxa

Agora que implantamos o Cloudflare Managed Ruleset e as Regras Personalizadas, vamos reforçar ainda mais o acesso ao aplicativo AcmeCorp com a **Limitação de Taxa Avançada.**

No painel de controle do Cloudflare, acesse **Segurança ▸ Regras de segurança ▸ Criar regra** e selecione **Rate Limiting Rules**.



Sua tarefa é limitar as tentativas de requisição a 5 por minuto para evitar ataques de força bruta.

Utilize os seguintes critérios:

- Defina o **campo** como **Caminho URI.**
- Defina o **operador** como **"Contém".**
- Defina o **valor** para **/services/safes/**
- Definir **características** para **IP**
- Quando a taxa exceder... **5** solicitações em um período **de 1 minuto**
- Defina a **ação** como **Bloquear** com **o Tipo de Resposta** definido como **padrão.**
- Selecione **"Bloquear" para a duração escolhida** como comportamento e defina **1 minuto** como valor.

Sua regra finalizada deve ter esta aparência:

Security rules > Advanced rate limiting rules > New advanced rate limiting rule

New advanced rate limiting rule

Protect your website and API from malicious traffic with rate limiting rules. Configure mitigation criteria and actions for better security.

Learn more

Rule name (required)

Rate Limiting

Give your rule a descriptive name.

Create rule with Cloudy

Field	Operator	Value		
URI Path	contains	/services/safes/	And	Or

e.g. /content

Expression Preview — Edit expression

```
(http.request.uri.path contains "/services/safes/")
```

Cache status

- [x] Also apply rate limiting to cached assets

With the same characteristics...

IP — And

- [ ] Use custom counting expression

When rate exceeds...

Requests (required)	Period (required)
5	1 minute

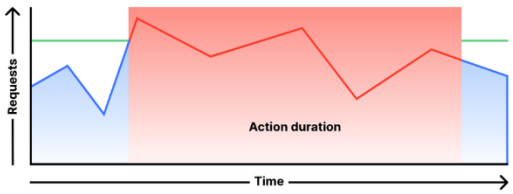
Then take action...

Choose action	With response type	With response code
Block	Default Cloudflare rate limiting response	429

Blocks matching requests and stops evaluating other rules

With the following behaviour

- (•) **Block for the selected duration**
  Once the rate is exceeded, the WAF will block all requests during the selected duration before the counter resets.
  1 minute
- ( ) Throttle requests over the maximum configured rate
  The WAF will block any requests exceeding the configured rate, and the remaining requests will be allowed.



Cancel — Save as Draft — Deploy

Deploy with API call

Implemente sua regra.

Você pode estar se perguntando por que usamos `&&` contains como operador neste exemplo e equals no exemplo anterior. Isso serve para destacar os diferentes operadores, mas também contains pode ser mais flexível para cenários como diretórios ou páginas com nomes semelhantes, ou em casos (como no site da AcmeCorp) em que barras invertidas de teste são adicionadas por meio de um redirecionamento. O equals operador `&&` é explícito; portanto, se você definisse uma política para `/`, /contact mas o aplicativo se conectasse a `/contact//`, a regra não seria acionada se o usuário navegasse para `/` /contact/, mas se a política usasse `&&` contains, então seria.

Agora vamos testar a regra que você acabou de criar.

Abra seu terminal e execute um loop usando curl ou powershell (substitua opportunistic-ziplinepelo domínio do seu laboratório):

curl

```
for i in {1..10}; do curl -svo /dev/null "https://opportunistic-zipline.sxplab.com/services/safes/" 2>&1
| grep "< HTTP.* "; done
```

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

PowerShell

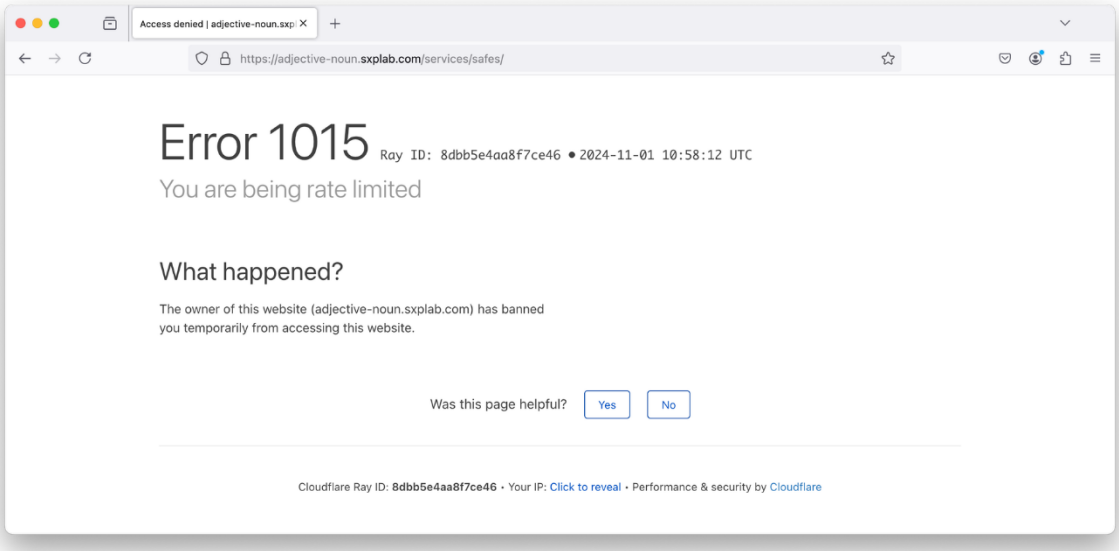
```
for (($i = 0); $i -lt 10; $i++) { try { $Response = Invoke-WebRequest -Uri
"https://opportunistic-zipline.sxplab.com/services/safes/"; $StatusCode = $Response.StatusCode } catch {
$StatusCode = $_.Exception.Response.StatusCode.value__ }; "#$i - $(Get-Date -Format "yyyy-MM-dd HH:mm:ss") -
$StatusCode" }
```

```
#0 - 2023-10-30 08:50:25 - 200
#1 - 2023-10-30 08:50:25 - 200
#2 - 2023-10-30 08:50:25 - 200
#3 - 2023-10-30 08:50:25 - 200
#4 - 2023-10-30 08:50:26 - 200
#5 - 2023-10-30 08:50:26 - 429
#6 - 2023-10-30 08:50:26 - 429
#7 - 2023-10-30 08:50:26 - 429
#8 - 2023-10-30 08:50:26 - 429
#9 - 2023-10-30 08:50:26 - 429
```

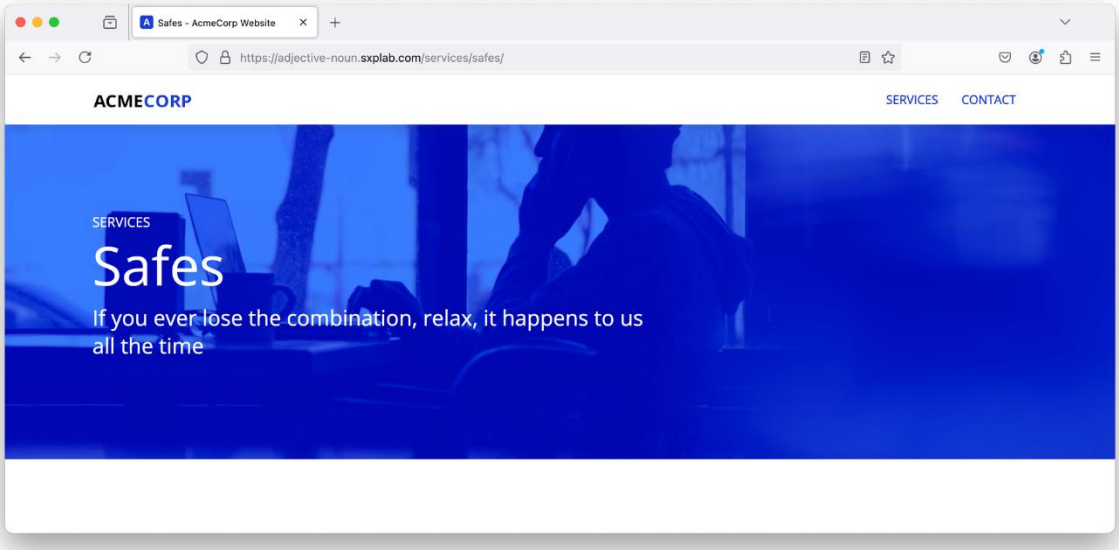
Observe como vemos algumas respostas 200, seguidas pelo HTTP 429, à medida que suas solicitações são limitadas pela taxa de requisições.

Você vai notar que não recebe um erro HTTP 429 em outras páginas.

Se você tiver algum problema ao usar o curl, basta usar um navegador e atualizar a página rapidamente, pois o limite é bastante baixo.



Aguarde 1 minuto e tente novamente. Agora você poderá acessar a página novamente.



Este exemplo é razoavelmente simples, no entanto, você pode ler mais sobre as opções de configuração na documentação avançada de limitação de taxa .

Que outras regras você pode criar? Sinta-se à vontade para experimentar.

Resumo

Nesta seção, configuramos o WAF, juntamente com regras personalizadas e limitação de taxa avançada.

MÓDULO 04

BOT MANAGEMENT

Modelos e validação de tráfego automatizado

Conteúdo original: páginas 27 a 30

Gerenciamento de Bots

Tarefa

Implante modelos de regras do Cloudflare Bot Management WAF.

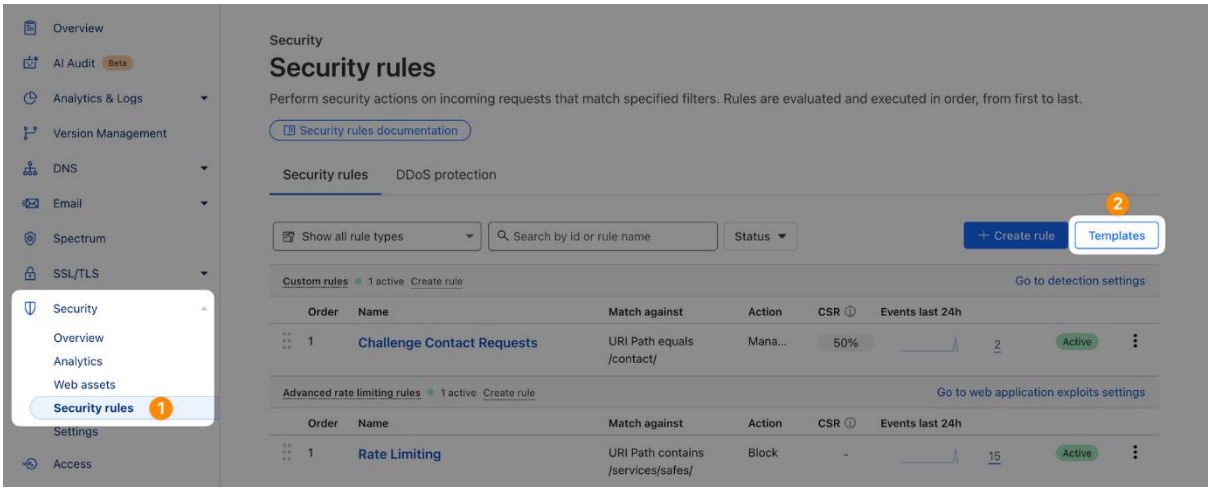
Por que

- Bloqueia ataques automatizados que drenam recursos e receita por meio de credential stuffing, inventory hoarding e content scraping
- Atribui uma pontuação de bot a cada solicitação usando aprendizado de máquina em 25 milhões de propriedades da Internet, distinguindo com precisão bots legítimos de bots maliciosos.
- Implementa-se instantaneamente através de modelos de regras WAF como "Definir Bots" e "Bots Prováveis" - basta ativar e definir uma ação de resposta.

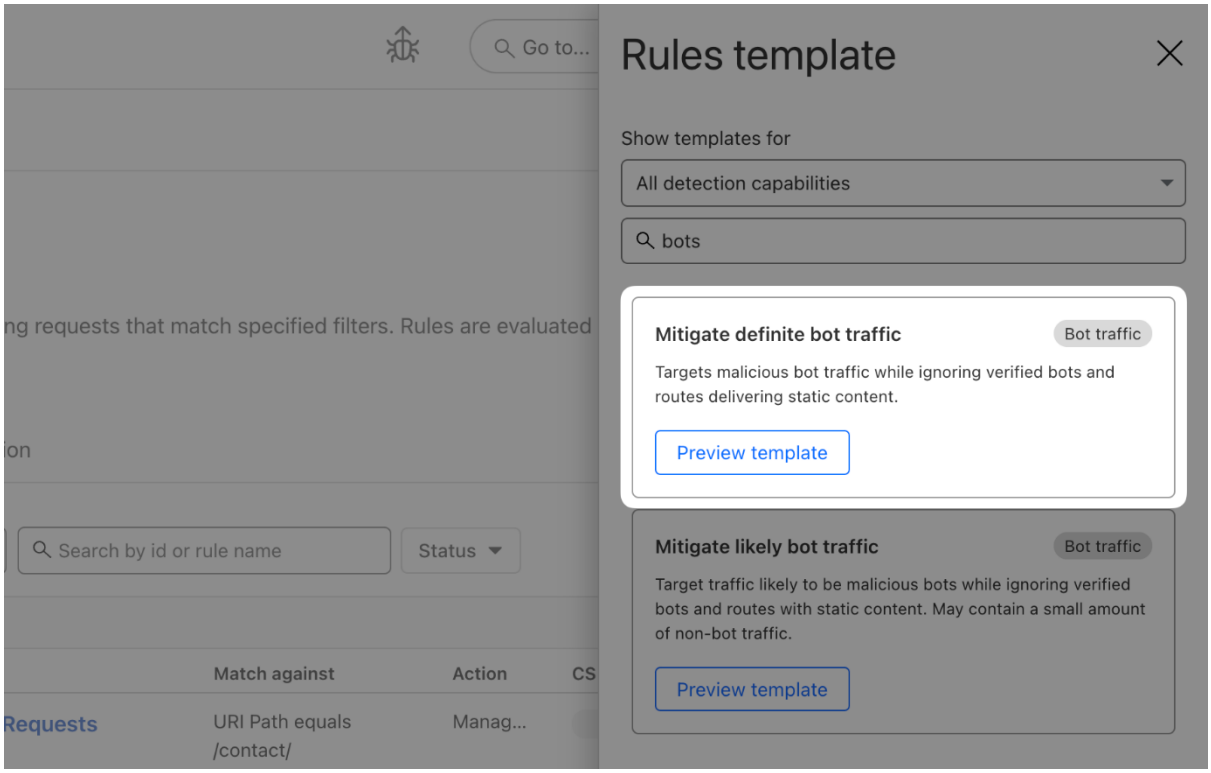
Passos

1. Implantar regra de modelo para bots específicos

No painel de controle do Cloudflare, acesse **Segurança ▸ Regras de segurança ▸ Modelos** .



Na **seção "Regras",** procure por modelos com a palavra "bots" e selecione o modelo " **Mitigar tráfego de bots definido"** clicando em **"Visualizar modelo"** .

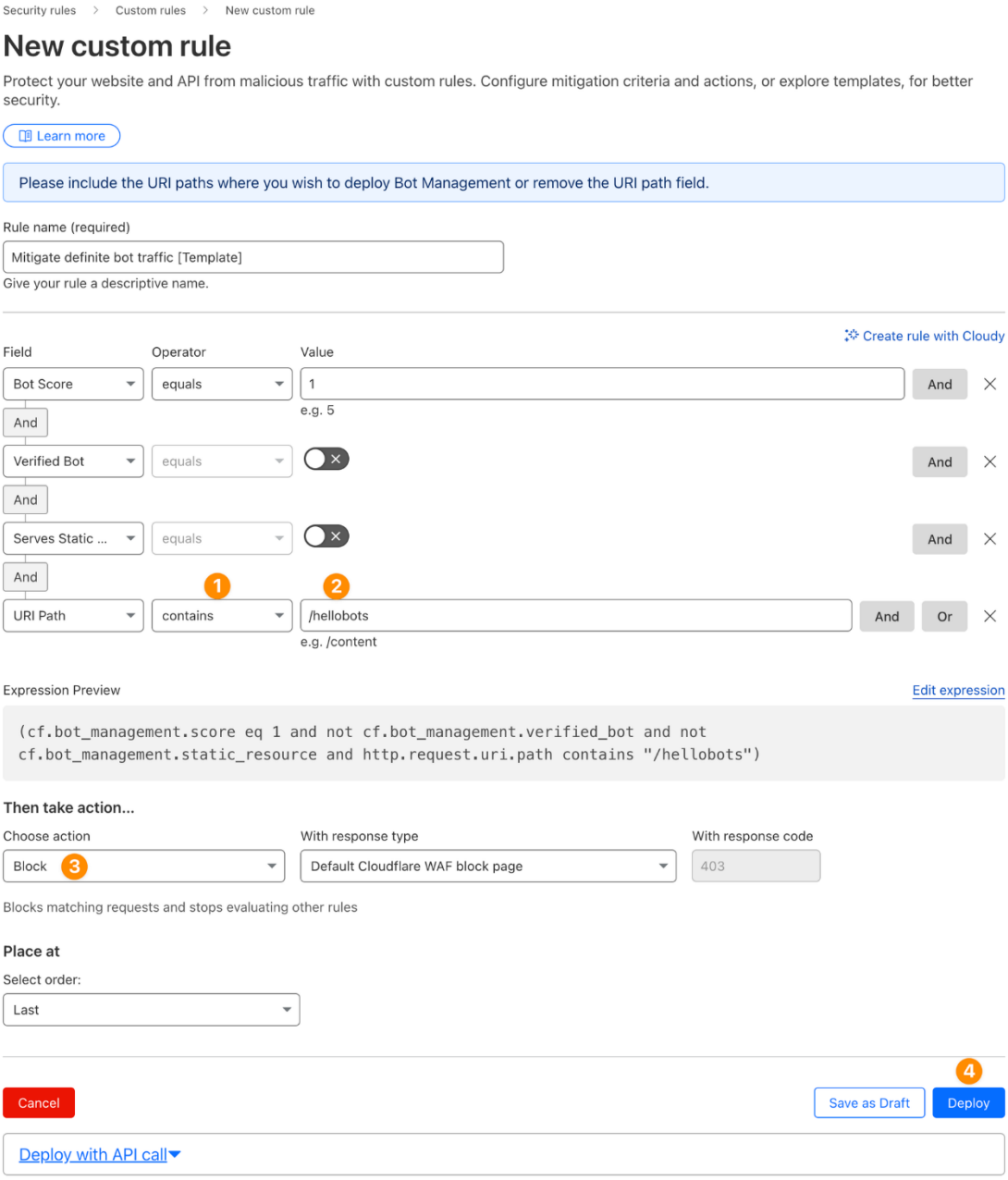


Isso abrirá a caixa de diálogo familiar de regras personalizadas com uma regra já configurada!

Deixe a regra como padrão, mas para não afetar outros exercícios, limite-a a um caminho inexistente:

- Para o último critério **de caminho URI** ...
- Defina o **operador** como **"contém".**
- Defina o **valor** para **/hellobots**
- Defina a **ação** como **Bloquear** com os valores padrão.

Sua configuração deve ser semelhante a esta:



Implante a regra.

2. Regra de teste

Abra seu terminal e inspecione os cabeçalhos HTTP recebidos do endpoint /hellobots usando o seguinte comando (substitua ` opportunistic-zipline <seu_domínio_de_laboratório>` pelo domínio do seu laboratório):

Curl

```
curl -I "https://opportunistic-zipline.sxplab.com/hellobots"
```

```
< HTTP/2 403
< date: Thu, 26 Oct 2023 14:27:56 GMT
< content-type: text/html; charset=UTF-8
< content-length: 4512
< x-frame-options: SAMEORIGIN
< referrer-policy: same-origin
< cache-control: max-age=15
< expires: Thu, 26 Oct 2023 14:28:11 GMT
< set-cookie: __cf_bm=G4vpNBHbkvCNn35LBmEbXittWY_pdlYvqjyZsmvR7JA-1698330476-0-ARnmg
< server: cloudflare
< cf-ray: 81c362047fbe45a0-LHR
```

PowerShell

```
Invoke-WebRequest -Uri "https://opportunistic-zipline.sxplab.com/hellobots" -method HEAD
```

```
Invoke-WebRequest: Response status code does not indicate success: 403 (Forbidden).
```

Tanto curl o comando quanto o comando do PowerShell são reconhecidos como uma ferramenta automatizada/bot, portanto, sua solicitação acionou a regra e o teste foi bloqueado.

3. Analise o bloqueio

No painel do Cloudflare, acesse **Segurança ▸ Análise** e examine as solicitações que acionaram o bloqueio. Você consegue identificar qual regra causou o bloqueio?

Se você não visualizar os eventos imediatamente, ou se preferir visualizar um período de tempo menor, tente selecionar *"Últimos 30 minutos"*.



Resumo

Nesta seção, configuramos o gerenciamento de bots.

Em seguida, realizaremos alguns ajustes na CDN e na configuração do cache.

MÓDULO 05

CACHE

Cache Rules, TTL e status code

Conteúdo original: páginas 31 a 43

Ajuste de cache

Tarefa

Verifique os parâmetros de configuração da CDN e modifique o comportamento do cache com regras de cache. O objetivo desta tarefa é servir uma página (incluindo HTML, que não é armazenado em cache por padrão) a partir do cache. Usaremos isso /services/luggages/ neste exemplo.

Por padrão, o Cloudflare armazena em cache diversas extensões diferentes, a menos que sua origem nos informe o contrário (por exemplo, com um cabeçalho Cache-Control com o valor private). Antes de prosseguir, dê uma olhada rápida em Comportamento Padrão do Cache . Em particular, observe as extensões e as respostas do cache.

Note que o HTML não é armazenado em cache por padrão. Às vezes, é desejável substituir o comportamento padrão do cache para exibir mais conteúdo a partir do cache.

Aviso: Os exemplos abaixo foram criados em um ambiente de laboratório para demonstrar a flexibilidade do cache e das regras de cache do Cloudflare. É muito importante considerar as implicações de alterar as configurações de cache em larga escala, principalmente ao substituir os valores padrão e armazenar conteúdo HTML em cache.

Por que

- Acelere a entrega de conteúdo fornecendo dados de mais de 300 locais de borda globais, reduzindo a carga e a latência na origem, além de diminuir os custos de largura de banda.
- Controle o comportamento do cache com precisão por meio de Regras de Cache — personalize TTLs, chaves de cache e configurações de resposta de origem sem escrever código.
- Forneça conteúdo desatualizado durante erros de origem com períodos de tolerância personalizáveis, garantindo alta disponibilidade mesmo se os sistemas de backend falharem.

Passos

Embora a maioria dos exemplos utilize o curl, também fornecemos exemplos para navegador e PowerShell. Sinta-se à vontade para reaproveitá-los e reutilizá-los nas etapas posteriores, caso prefira essas ferramentas.

1. Analisar o comportamento padrão

Usando curl o PowerShell, vamos verificar os cabeçalhos de resposta HTTP (lembre-se de substituir ` opportunistic-zipline <sua zona>` pela sua zona):

curl

```
curl -I https://opportunistic-zipline.sxplab.com/services/luggages/
```

```
< HTTP/2 200
< date: Thu, 26 Oct 2023 07:42:38 GMT
< content-type: text/html
< last-modified: Thu, 26 Oct 2023 03:55:52 GMT
< x-origin: origin-1
< accept-ranges: bytes
< cf-cache-status: DYNAMIC
< server: cloudflare
< cf-ray: 81c110540d6823ab-LHR
```

PowerShell

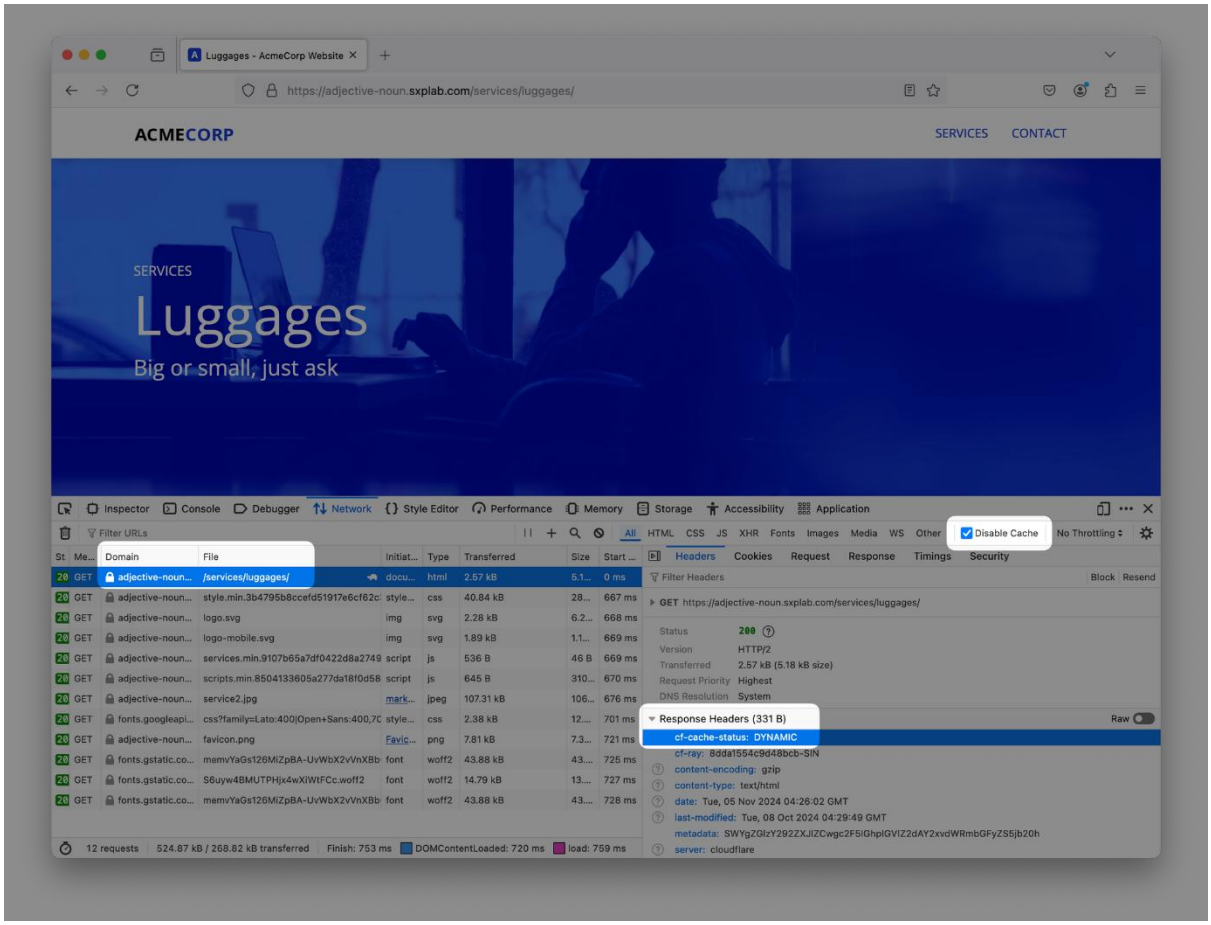
```
Invoke-WebRequest -Uri "https://opportunistic-zipline.sxplab.com/services/luggages/" -Method HEAD
```

```
StatusCode        : 200
StatusDescription : OK
Content           :
RawContent        : HTTP/1.1 200 OK
                    Date: Mon, 30 Oct 2023 09:02:08 GMT
                    Connection: keep-alive
                    X-Origin: origin-1
                    CF-Cache-Status: DYNAMIC
                    Age: 8
                    Cache-Control: public, max-age=14400
                    Accept-Ranges: bytes
                    Expect-CT: max-age=8…
```

O cabeçalho principal que nos interessa aqui é cf-cache-status — observe como o valor é DYNAMIC — isso informa ao Cloudflare que este recurso **não** é considerado elegível para cache e, portanto, a página é obtida diretamente da origem. Alteraremos esse comportamento na próxima tarefa e analisaremos os resultados novamente.

Nota: Se você não tiver o curl, abra uma nova aba no seu navegador e abra **as Ferramentas do Desenvolvedor** (normalmente, isso pode ser acessado pressionando F12).

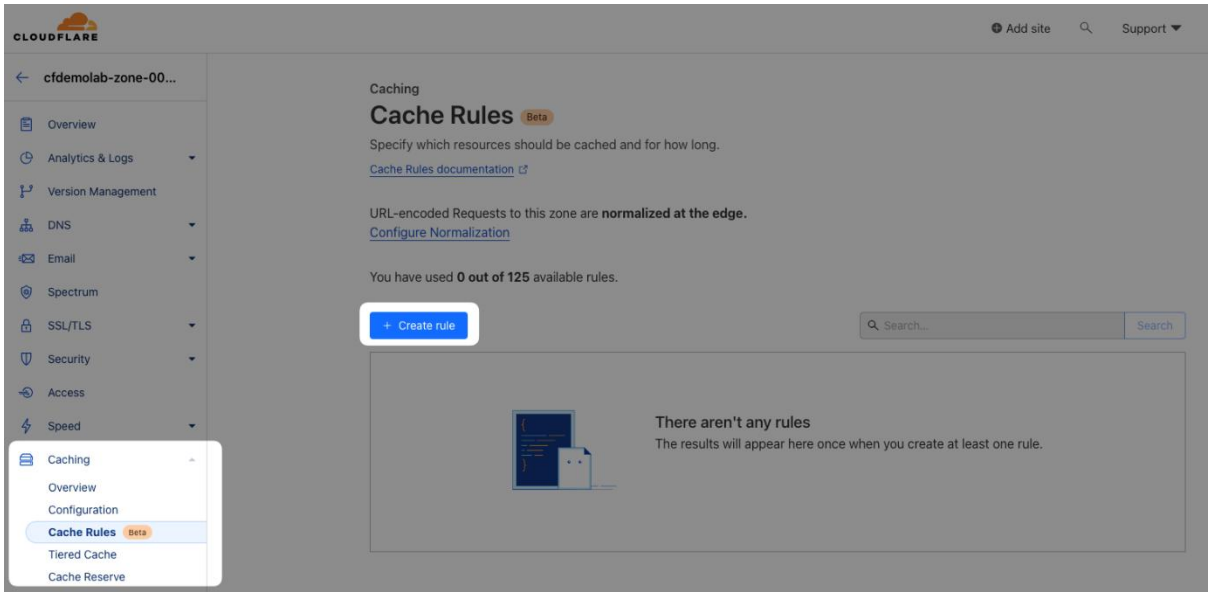
Navegue até sua página /services/luggages/ - então na janela Ferramentas do desenvolvedor em **Rede** selecione a página (1) e então você poderá ver os cabeçalhos (2) - certifique-se de desativar o cache usando a caixa de seleção Disable cache



2. Crie uma regra de cache para /services/luggages

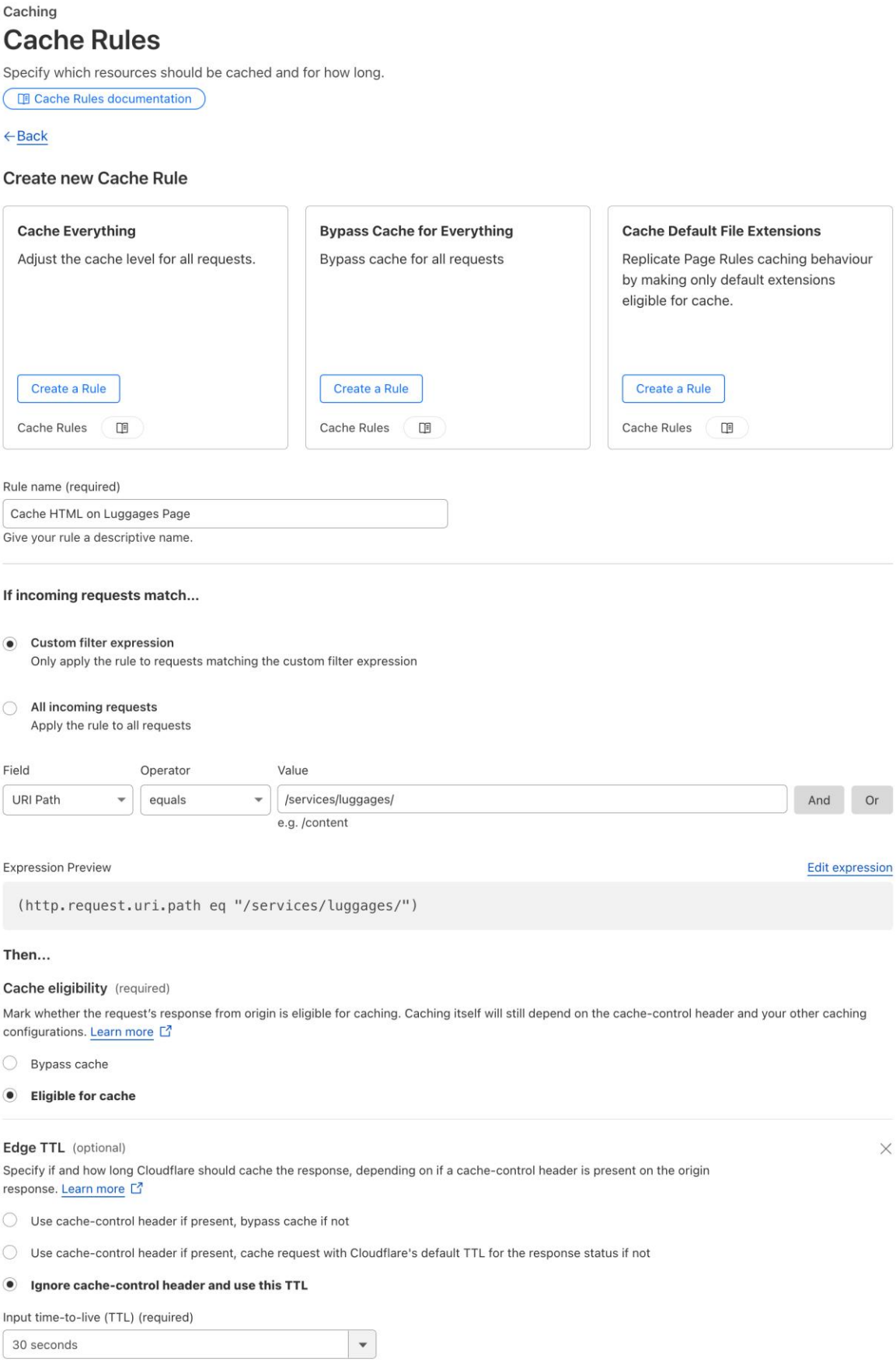
Agora vamos modificar o comportamento padrão do cache para servir o conteúdo HTML a /services/luggages/ partir do cache.

No painel do Cloudflare, acesse **Cache ▸ Regras de cache** e selecione **Criar regra.**



- Insira um nome descritivo para a regra, como "Armazenar HTML em cache na página de bagagens".
- Defina o **campo** como **Caminho URI.**
- Defina o **operador** como **igual a**
- Defina o **valor** para **/services/luggages/**
- Defina **o status do cache** como " **Elegível para cache".**
- Você pode personalizar configurações adicionais de cache, como:
    - Edge TTL: O tempo que o Cloudflare leva para armazenar o recurso em cache na rede do Cloudflare.
    - TTL do navegador: o tempo que o navegador do visitante leva para armazenar o recurso em cache.
    - Chave de cache: personalize o que será incluído na chave de cache.
    - Exibir conteúdo desatualizado: Exibir conteúdo desatualizado enquanto o processo de revalidação ocorre em segundo plano.
    - Respeitar ETags fortes: Respeitar ETags fortes para revalidação de cache.
    - Repasse da página de erro de origem: Permite repassar as páginas de erro da origem.
- À direita de onde você vê **Edge TTL** , clique em **+ Adicionar configuração**
- Selecione **"Ignorar cabeçalho cache-control", use este TTL** e defina a duração para **30 segundos.**

Sua configuração deve ser semelhante a esta:


Caching
Cache Rules
Specify which resources should be cached and for how long.
Cache Rules documentation
←Back
Create new Cache Rule
Cache Everything
Adjust the cache level for all requests.
Create a Rule
Cache Rules
Bypass Cache for Everything
Bypass cache for all requests
Create a Rule
Cache Rules
Cache Default File Extensions
Replicate Page Rules caching behaviour by making only default extensions eligible for cache.
Create a Rule
Cache Rules
Rule name (required)
Cache HTML on Luggages Page
Give your rule a descriptive name.
If incoming requests match...
Custom filter expression
Only apply the rule to requests matching the custom filter expression
All incoming requests
Apply the rule to all requests
Field
Operator
Value
URI Path
equals
/services/luggages/
e.g. /content
And
Or
Expression Preview
Edit expression
(http.request.uri.path eq "/services/luggages/")
Then...
Cache eligibility (required)
Mark whether the request's response from origin is eligible for caching. Caching itself will still depend on the cache-control header and your other caching configurations. Learn more
Bypass cache
Eligible for cache
Edge TTL (optional)
Specify if and how long Cloudflare should cache the response, depending on if a cache-control header is present on the origin response. Learn more
Use cache-control header if present, bypass cache if not
Use cache-control header if present, cache request with Cloudflare's default TTL for the response status if not
Ignore cache-control header and use this TTL
Input time-to-live (TTL) (required)
30 seconds

- Role a página para baixo, deixando os outros valores como padrão (mas fique à vontade para explorá-los!) e clique em **Implantar.**

Isso criará uma regra de cache que tornará todo o conteúdo da /services/luggages/ página elegível para cache e ignorará os cabeçalhos de controle de cache de origem. O conteúdo será armazenado em cache por 30 segundos no servidor de borda da Cloudflare.

3. Regra de cache de teste

Usando curl (ou através do navegador ou dos métodos do PowerShell compartilhados anteriormente), vamos verificar os cabeçalhos de resposta HTTP:

```
curl -I https://opportunistic-zipline.sxplab.com/services/luggages/
```

```
< HTTP/2 200
< date: Thu, 26 Oct 2023 11:02:13 GMT
< content-type: text/html
< last-modified: Thu, 26 Oct 2023 03:55:52 GMT
< x-origin: origin-2
< cf-cache-status: MISS
< expires: Thu, 26 Oct 2023 15:02:13 GMT
< cache-control: public, max-age=14400
< accept-ranges: bytes
< server: cloudflare
< cf-ray: 81c234aabcf976c3-LHR
< cf-team: 1c0d3f3eaa000076c304734400000001
```

Observe como agora você vê cf-cache-status com um valor de MISS

MISS Isso nos informa que o recurso é elegível para cache, mas não estava em cache e, portanto, veio da origem. Execute o comando novamente imediatamente:

```
curl -I "https://opportunistic-zipline.sxplab.com/services/luggages/"
```

Sucesso! Temos um cf-cache-status resultado positivo HIT - a página foi exibida a partir do cache.

Observe o age cabeçalho — vemos que ele está em cache há 19 segundos no exemplo abaixo. Certifique-se de ter esperado pelo menos 30 segundos e tente novamente:

```
curl -I "https://opportunistic-zipline.sxplab.com/services/luggages/"
```

```
< HTTP/2 200
< date: Thu, 26 Oct 2023 11:06:01 GMT
< content-type: text/html
< last-modified: Thu, 26 Oct 2023 03:55:52 GMT
< x-origin: origin-2
< cf-cache-status: REVALIDATED
< expires: Thu, 26 Oct 2023 15:06:01 GMT
< cache-control: public, max-age=14400
< accept-ranges: bytes
< server: cloudflare
< cf-ray: 81c23a3fed9076c3-LHR
< cf-team: 1c0d42bbe9000076c306090400000001
```

O conteúdo já foi acessado REVALIDATED.

Se você tentar novamente, verá outro resultado imediatamente.

Experimente navegar para outras páginas; você notará que o conteúdo cf-cache-status ainda está DYNAMIC em outro lugar.

4. Ampliando o Escopo

Quando usado adequadamente, o armazenamento em cache de conteúdo oferece benefícios substanciais de desempenho e desafoga tarefas dos seus servidores de origem. No último exemplo, criamos uma regra para armazenar em cache todo o conteúdo em /services/luggages/- agora vamos expandir isso para abranger tudo em /services/.

Modifique a regra que você já criou, atualizando os critérios de correspondência da seguinte forma:

- Mantenha o **campo** definido como **Caminho URI.**
- Defina o **operador** como **"contém".**
- Defina o **valor** para **/services/**
- Deixe os outros valores como estão e clique em **Salvar.**

Faça o teste navegando para outra página, por exemplo /services/arrows/ , execute o comando curl pelo menos duas vezes; você verá um cache HIT.

```
curl -I https://opportunistic-zipline.sxplab.com/services/arrows/
```

```
< HTTP/2 200
< date: Thu, 26 Oct 2023 11:17:12 GMT
< content-type: text/html
< last-modified: Thu, 26 Oct 2023 03:55:53 GMT
< x-origin: origin-2
< cf-cache-status: HIT
< age: 3
< expires: Thu, 26 Oct 2023 15:17:12 GMT
< cache-control: public, max-age=14400
< accept-ranges: bytes
< server: cloudflare
< cf-ray: 81c24a9e3fe776c3-LHR
< cf-team: 1c0d4cf6e1000076c30ab6c400000001
```

Até agora, temos nos concentrado em páginas que existem e retornam uma HTTP 200 resposta. Mas o que acontece se tentarmos navegar para uma página que corresponde a uma regra de cache, mas não existe?

Tente acessar qualquer página fictícia dentro de /services/, por exemplo:

```
curl -I https://opportunistic-zipline.sxplab.com/services/doesnotexist/
```

```
< HTTP/2 404
< date: Thu, 26 Oct 2023 11:21:17 GMT
< content-type: text/html
< cf-cache-status: MISS
< expires: Thu, 26 Oct 2023 15:21:17 GMT
< cache-control: public, max-age=14400
< server: cloudflare
< cf-ray: 81c25097ca5a76c3-LHR
< cf-team: 1c0d50b2d7000076c30c6ed400000001
```

Recebemos uma resposta HTTP 404 (Arquivo não encontrado) e cf-cache-status um MISS

Execute o comando novamente:

```
curl -I "https://opportunistic-zipline.sxplab.com/services/doesnotexist/"
```

```
< HTTP/2 404
< date: Thu, 26 Oct 2023 11:22:30 GMT
< content-type: text/html
< cf-cache-status: HIT
< age: 73
< expires: Thu, 26 Oct 2023 15:22:30 GMT
< cache-control: public, max-age=14400
< server: cloudflare
< cf-ray: 81c25261ae4876c3-LHR
< cf-team: 1c0d51d0fd000076c30cee8400000001
```

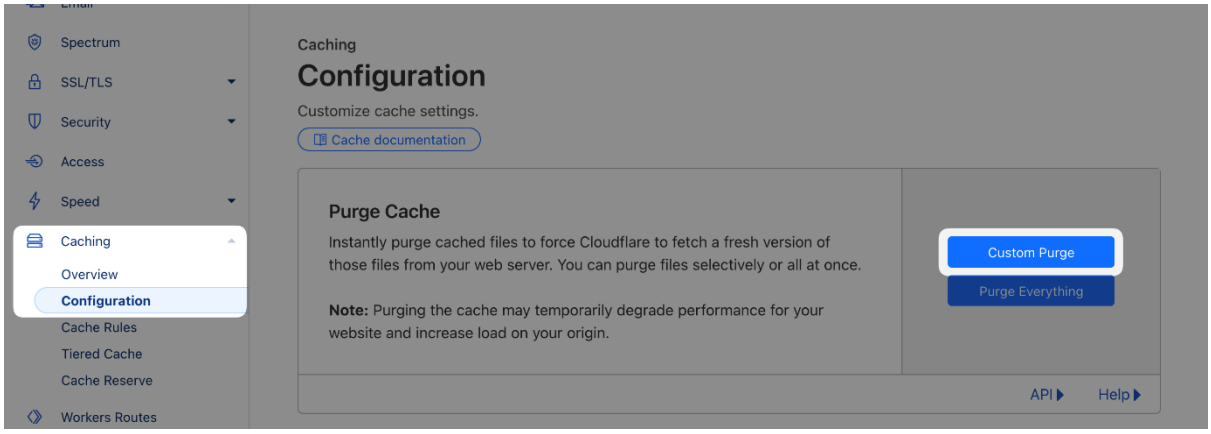
Analise o resultado acima por um momento e tente replicá-lo. O que você percebe de estranho?

- Temos um cache HIT apesar de ser um erro 404.
- O age cabeçalho está retornando um valor maior que o nosso TTL.
- ...por que?

A documentação sobre cache por código de status mostra que o código de erro 404 tem um TTL padrão de 3 minutos (ou 180 segundos).

Para corrigir isso, basta modificarmos nossa regra para substituir o comportamento padrão de um 404 código de resposta.

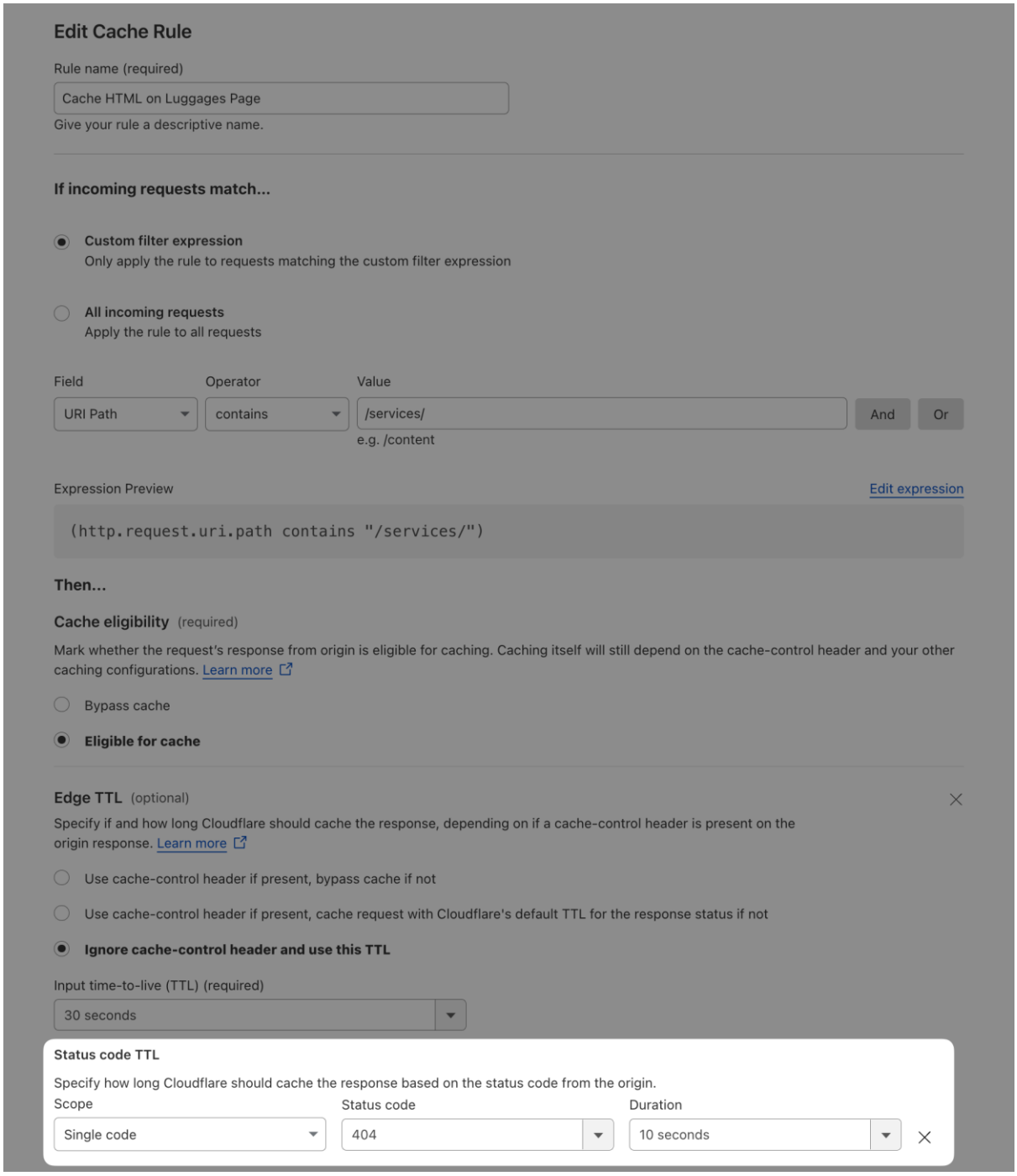
Antes de prosseguirmos, vamos limpar o cache. Caso contrário, você poderá continuar recebendo acessos devido aos testes anteriores. Navegue até **Cache ▸ Configuração ▸ Custom Purge**



Fique à vontade para usar o método que preferir. Caso não saiba quais são esses métodos, consulte a documentação para desenvolvedores sobre limpeza de cache .

Volte às Regras de Cache e modifique sua regra para definir um **TTL (tempo de vida) para o código de status** :

- Na seção Edge TTL que já configuramos, selecione **Adicionar configuração de código de status.**
- Defina o **escopo** para **Código Único**
- Defina o **código de status** para **404.**
- Defina **a duração** para **10 segundos.**



Não faça mais nenhuma alteração e salve sua regra novamente.

Execute o teste novamente:

```
curl -I "https://opportunistic-zipline.sxplab.com/services/doesnotexist/"
```

```
< HTTP/2 404
< date: Thu, 26 Oct 2023 11:45:05 GMT
< content-type: text/html
< cf-cache-status: HIT
< age: 9
< expires: Thu, 26 Oct 2023 15:45:05 GMT
< cache-control: public, max-age=14400
< server: cloudflare
< cf-ray: 81c2737a7d6876c3-LHR
< cf-team: 1c0d668088000076c315b30400000001
```

Neste exemplo, consegui capturar a mensagem age aos 9 segundos. Ao testar, tente executar o comando algumas vezes. Você verá que obterá um status EXPIRED após 10 segundos e, em seguida, HIT novamente:

```
curl -I "https://opportunistic-zipline.sxplab.com/services/doesnotexist/"
```

```
< HTTP/2 404
< date: Thu, 26 Oct 2023 11:46:20 GMT
< content-type: text/html
< cf-cache-status: EXPIRED
< expires: Thu, 26 Oct 2023 15:46:20 GMT
< cache-control: public, max-age=14400
< server: cloudflare
< cf-ray: 81c2754cebc776c3-LHR
< cf-team: 1c0d67a404000076c3161cc400000001

<rerun immediately>

< HTTP/2 404
< date: Thu, 26 Oct 2023 11:46:21 GMT
< content-type: text/html
< cf-cache-status: HIT
< age: 1
< expires: Thu, 26 Oct 2023 15:46:21 GMT
< cache-control: public, max-age=14400
< server: cloudflare
< cf-ray: 81c27555ecfb76c3-LHR
< cf-team: 1c0d67a9ae000076c316293400000001
```

É importante entender como o cache varia de acordo com o código de status, e o cenário de páginas 404 em cache não é incomum!

Resumo

Nesta seção, exploramos como ajustar e otimizar o cache do Cloudflare.

MÓDULO 06

LOAD BALANCING

Pools, monitores e session affinity

Conteúdo original: páginas 44 a 53

Balanceador de carga

Tarefa

Crie um balanceador de carga que distribua as solicitações recebidas entre vários servidores de origem.

Por que

Os balanceadores de carga são componentes essenciais da infraestrutura de rede que fornecem:

- **Otimização de recursos** através da distribuição do tráfego web por vários servidores, maximizando a taxa de transferência e minimizando a latência.
- **Alta disponibilidade** através do roteamento do tráfego apenas para servidores operacionais, ignorando os servidores com falha.
- **Escalabilidade**, facilitando a rápida adaptação às mudanças no volume de tráfego, ao tornar mais fácil a adição de servidores ou poder computacional.

Passos

Como você deve se lembrar, você começou esses laboratórios usando uma origem compartilhada e depois migrou para uma origem dedicada, executada em sua máquina virtual, usando cloudflared como ponto de entrada.

Neste exercício, voltamos a usar a origem compartilhada por um momento. Ela consiste, na verdade, em 2 servidores de origem localizados atrás de um balanceador de carga legado:

Origin 1: 52.191.5.18

Origin 2: 52.170.37.97

Legacy LB: 20.88.188.200 < do not use this LB in this exercise

Neste exercício, você colocará um balanceador de carga Cloudflare na frente dos dois servidores de origem compartilhados, substituindo o balanceador de carga legado.

Cada servidor de origem adiciona o cabeçalho X-Origin à resposta; isso ajudará você a rastrear qual origem processou sua solicitação.

```
curl --head 52.191.5.18
```

```
...
X-Origin: origin-1
...
```

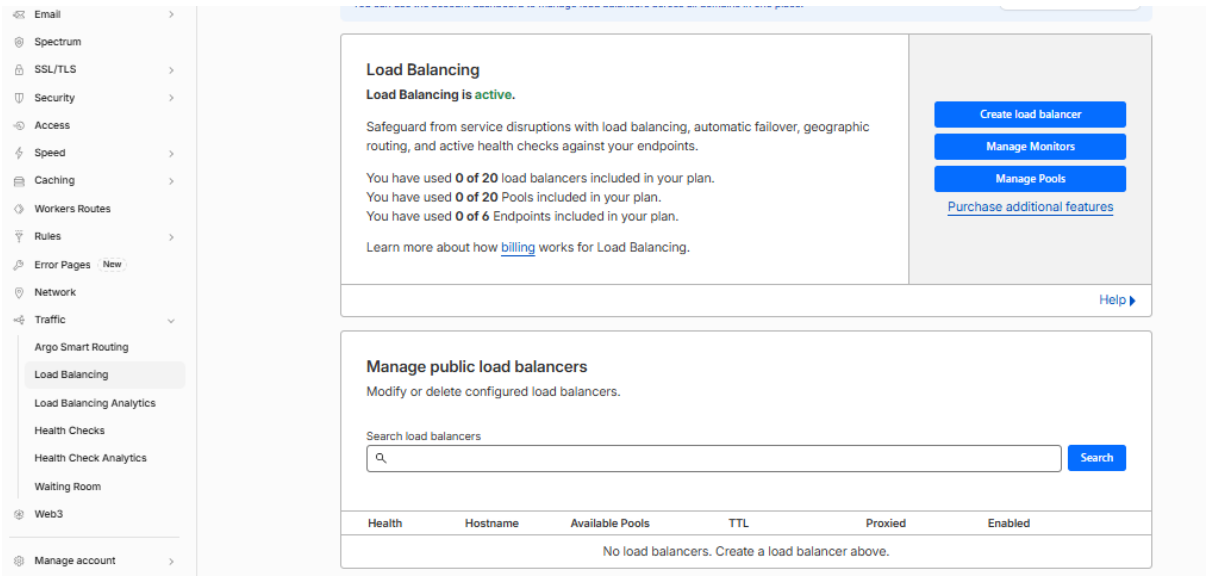
```
curl --head 52.170.37.97
```

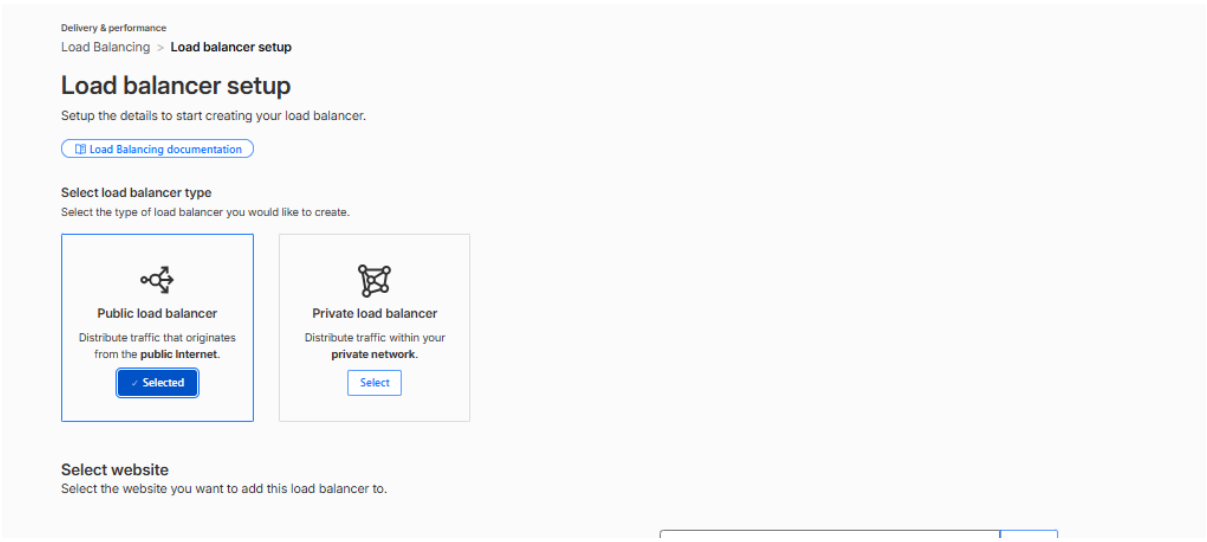
```
...
X-Origin: origin-2
...
```

Vamos criar o balanceador de carga!

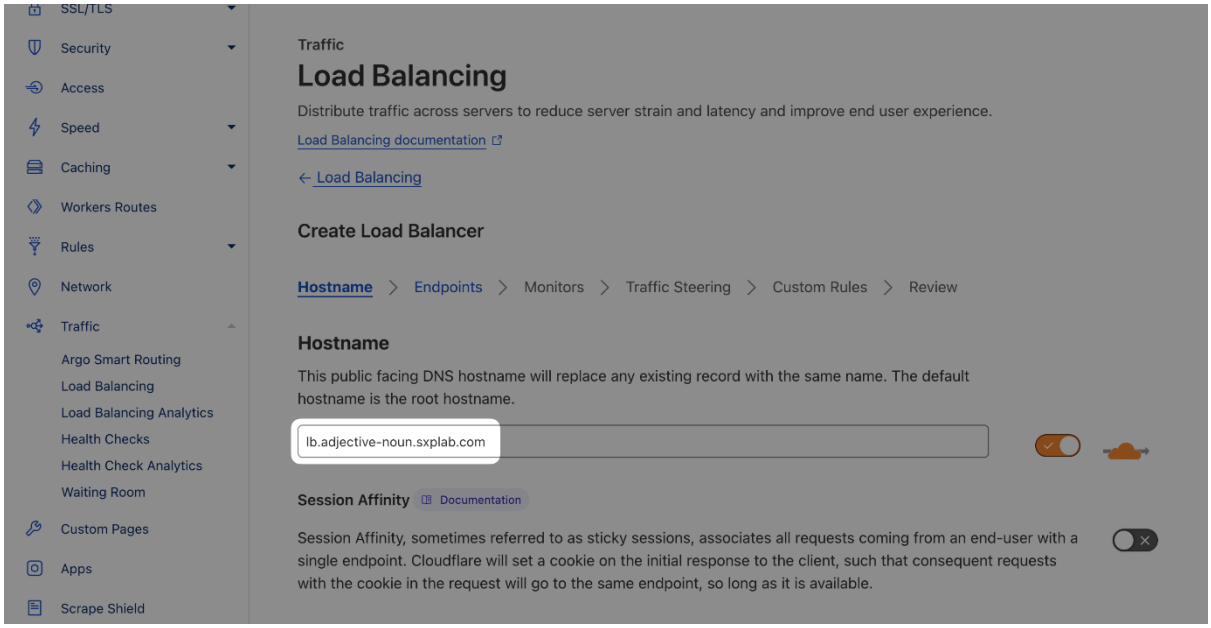
1. Criar balanceador de carga

No seu painel de controle , em **Sites** , abra uma zona e selecione **Tráfego** ▸ **Balanceamento de Carga** . Aqui, crie um novo Balanceador de Carga (LB):





Crie um novo nome de host para o balanceador de carga adicionando um prefixo lb à sua zona existente. Mantenha a afinidade de sessão e o roteamento adaptativo desativados por enquanto.



2. Criar pool de origem

Na próxima etapa, você será solicitado a adicionar um Pool de Origem. Como ainda não temos um, selecione **Criar um pool** .

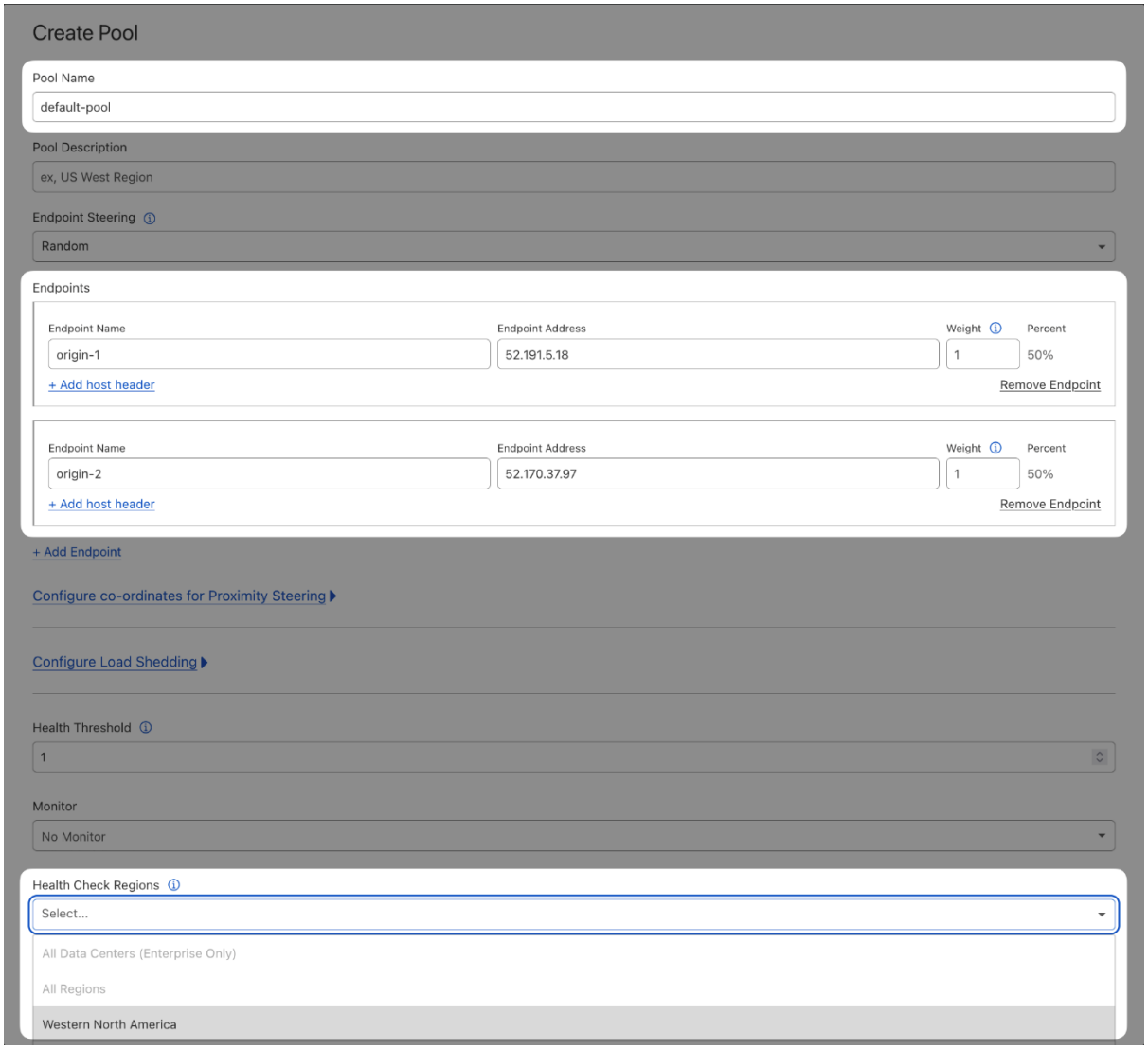
Dê um nome ao seu pool, certifique-se de que **o Endpoint Steering** esteja definido como **Random** e adicione as duas origens a seguir:

origin-1 52.191.5.18 1

origin-2 52.170.37.97 1

Name	IP	Weight
origin-1	52.191.5.18	1
origin-2	52.170.37.97	1

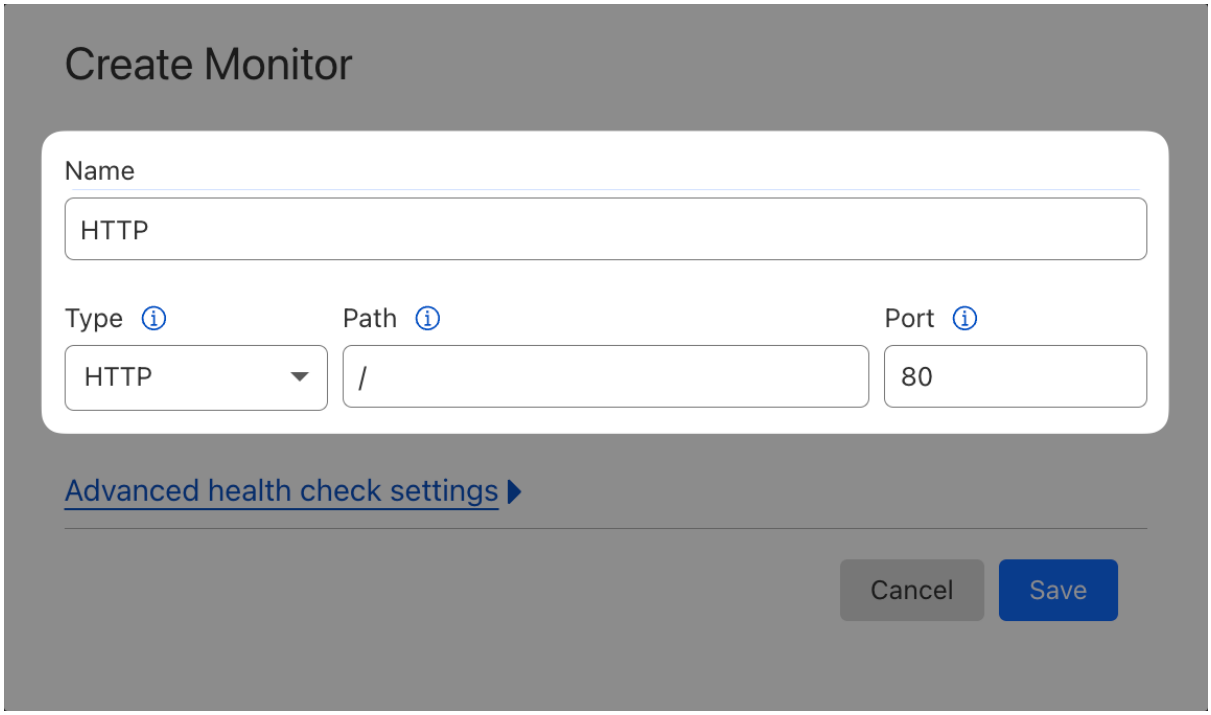
Em seguida, na parte inferior, selecione Regiões de Verificação de Saúde: **América do Norte Ocidental** (essa é a região onde estão localizadas nossas origens).


Create Pool
Pool Name
default-pool
Pool Description
ex, US West Region
Endpoint Steering
Random
Endpoints
Endpoint Name
Endpoint Address
Weight
Percent
origin-1
52.191.5.18
1
50%
+ Add host header
Remove Endpoint
Endpoint Name
Endpoint Address
Weight
Percent
origin-2
52.170.37.97
1
50%
+ Add host header
Remove Endpoint
+ Add Endpoint
Configure co-ordinates for Proximity Steering
Configure Load Shedding
Health Threshold
1
Monitor
No Monitor
Health Check Regions
Select...
All Data Centers (Enterprise Only)
All Regions
Western North America

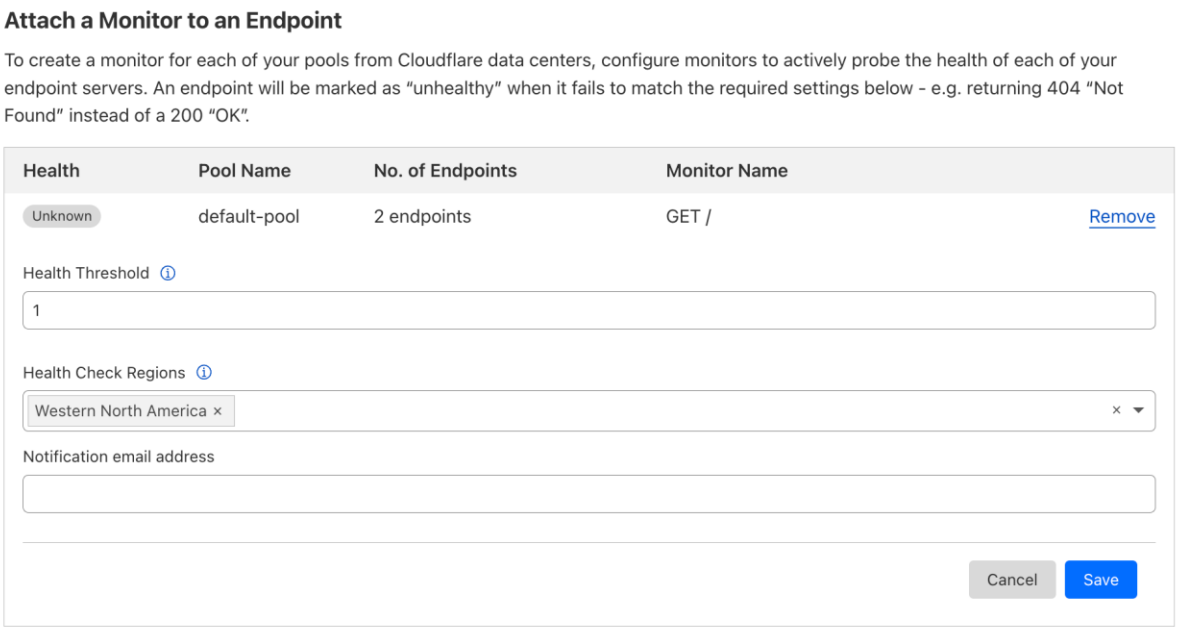
3. Criar monitores

Na página seguinte, selecione **Anexar monitor** e **Criar um monitor** .

Para os fins deste laboratório, vamos criar um monitor HTTP simples.



Ao conectar o monitor, mantenha as configurações padrão e clique em **Salvar** para continuar para a próxima página.



4. Traffic Steering

Mantenha o Traffic Steering **desativado,** pois estamos usando apenas uma pool.

Pule as etapas restantes da configuração e **clique em Salvar e Implantar** no final.

5. Teste

Tente enviar várias solicitações para o seu balanceador de carga e verifique se um servidor de origem aleatório processa cada solicitação. Você pode usar o curl repetidamente:

```
curl -s --head https://lb.opportunistic-zipline.sxplab.com
```

```
...
X-Origin: origin-2
...
```

1ª requisição

```
curl -s --head https://lb.opportunistic-zipline.sxplab.com
```

```
...
X-Origin: origin-1
...
```

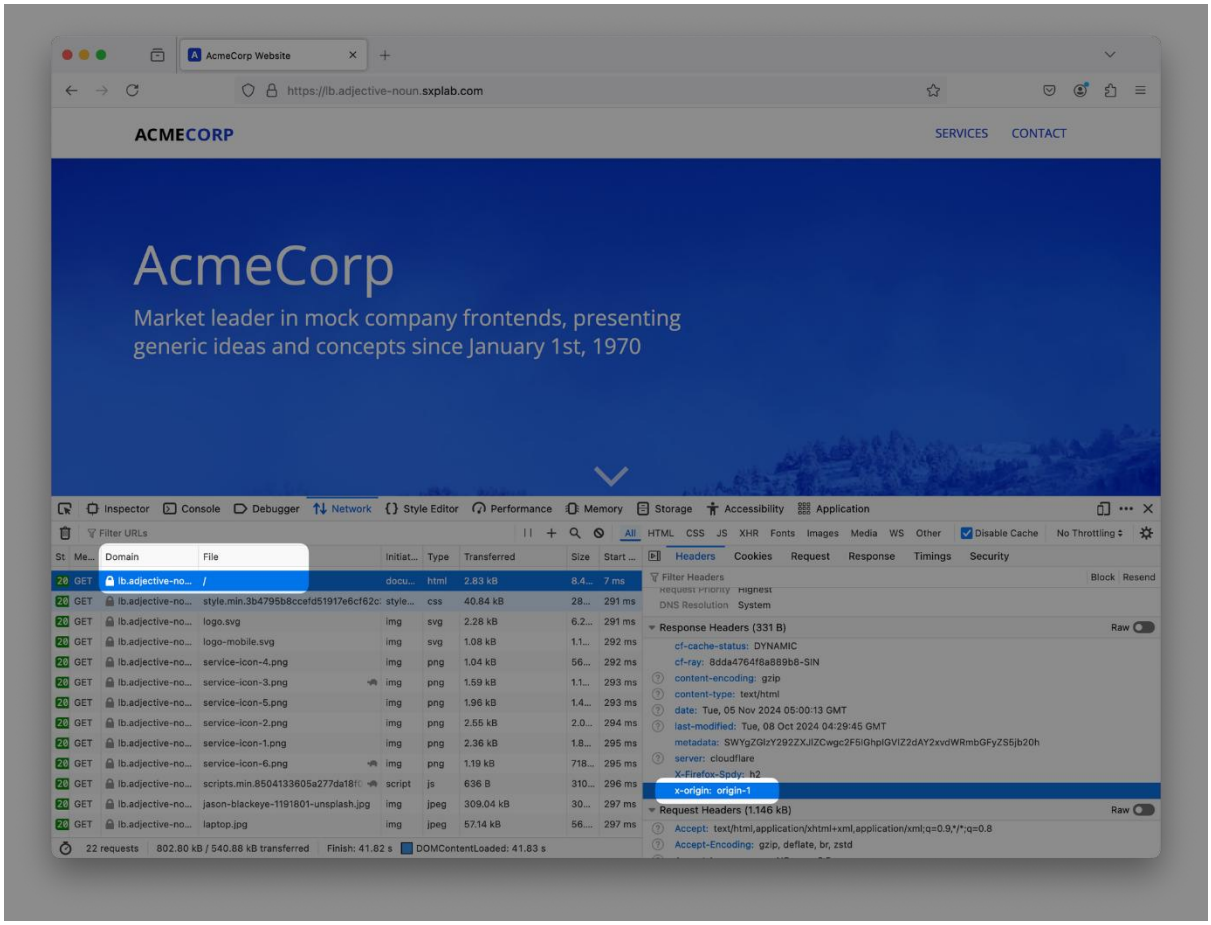
2ª requisição

Ou você pode usar o seguinte script:

```
while true; do curl -s --head https://lb.opportunistic-zipline.sxplab.com | grep -i x-origin; done
```

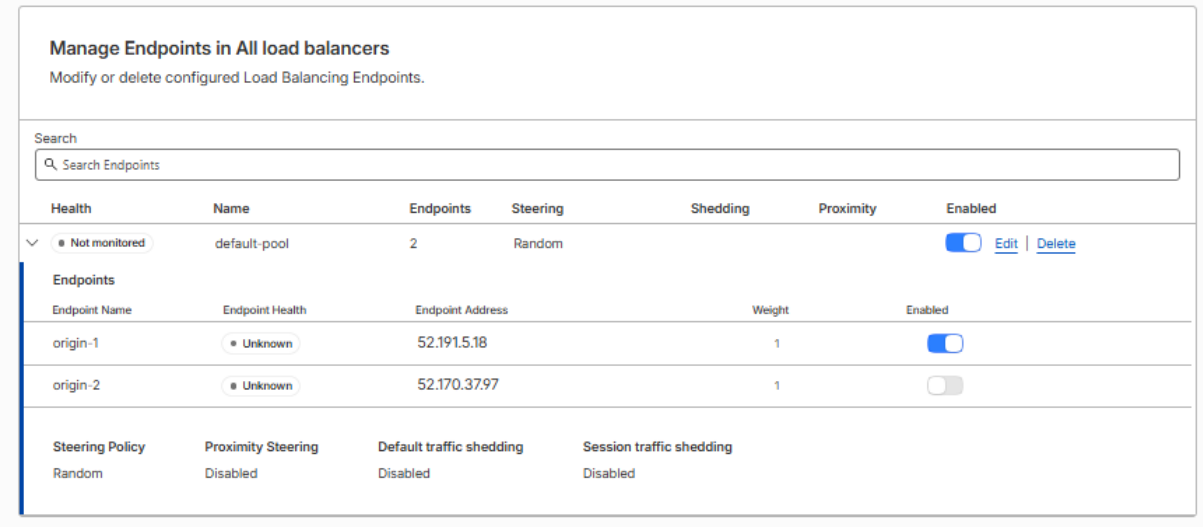
```
X-Origin: origin-2
X-Origin: origin-2
X-Origin: origin-1
X-Origin: origin-1
...
```

Ou, se você não tiver o curl, use as Ferramentas de Desenvolvedor do seu navegador. Recarregue a página algumas vezes e você deverá vê-la sendo servida por uma origem diferente.



6. Desative uma origem

Volte para a página principal de Balanceamento de Carga, selecione **load balancing – pool**, expanda o pool de origem que você criou e desative um dos servidores de origem. Isso simula uma interrupção em uma de suas origens.



7. Reteste

Teste novamente qual origem está atendendo o tráfego. Agora você deverá ver todo o tráfego sendo tratado pelo servidor de origem habilitado restante:

```
while true; do curl -s --head https://lb.organized-keystone.sxplab.com | grep -i x-origin; done
```



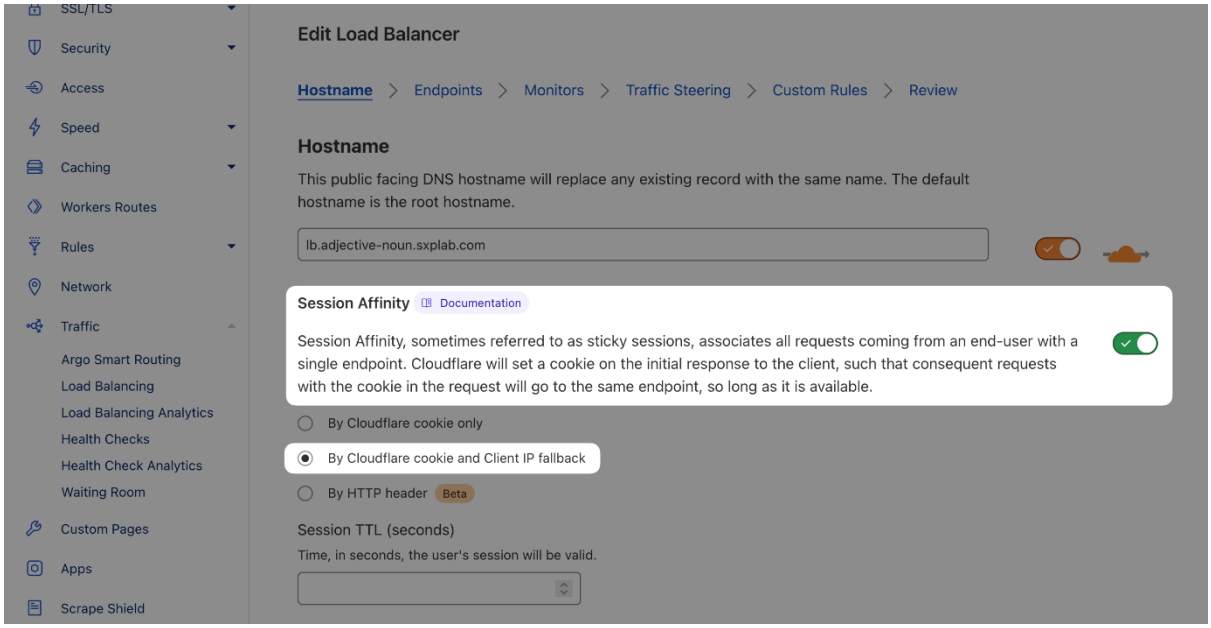
8. Configurar afinidade de sessão

Volte ao seu pool de origem e habilite novamente ambas as origens. Você pode testar novamente se ambas as origens estão processando suas solicitações, em ordem aleatória.

Em seguida, volte para a página principal de Balanceamento de Carga e clique em **Editar** ao lado do balanceador de carga que você criou.

Ative **a afinidade de sessão** e selecione " **Por cookie do Cloudflare" e "Recurso alternativo de IP do cliente"** .

Isso garantirá que todas as solicitações do mesmo navegador (cookie) ou cliente (IP) sejam processadas pelo mesmo servidor de origem. Leia mais aqui .



9. Reteste

Todas as suas solicitações provenientes do mesmo cliente (por exemplo, um navegador específico) agora devem ser atendidas consistentemente pela mesma origem. Como os cookies não são compartilhados entre navegadores, tentar acessar a partir de outro navegador pode resultar em uma origem diferente atendendo à solicitação. Mas, para o mesmo navegador, será sempre a mesma origem.

Resumo

Nesta seção, configuramos o Load Balancer com um pool de origem e um monitor de serviço.

MÓDULO 07

RULES

Transform Rules e URL Rewrite

Conteúdo original: páginas 54 a 60

Regras

Tarefa

Nesta seção, exploraremos recursos que permitem alterações nas configurações e no comportamento, e configuraremos alguns exemplos com Transform Rules

Se você está familiarizado com as Regras de Página, pode estar se perguntando por que elas estão faltando. Embora sejam válidas, elas fazem parte de um recurso antigo que será descontinuado em breve. Como boa prática, use uma das novas opções de Regras que melhor atendam às suas necessidades.

Por que

É comum desejar alterar as configurações padrão e ter configurações personalizadas que melhor se adaptem à sua aplicação. As Regras do Cloudflare permitem que você ajuste solicitações e respostas, configure definições personalizadas e acione ações específicas para solicitações correspondentes.

Existem muitas opções que podem ser configuradas, portanto, sinta-se à vontade para explorar as configurações além destes exercícios. Antes de prosseguir com as tarefas, familiarize-se brevemente com a documentação de Regras acima.

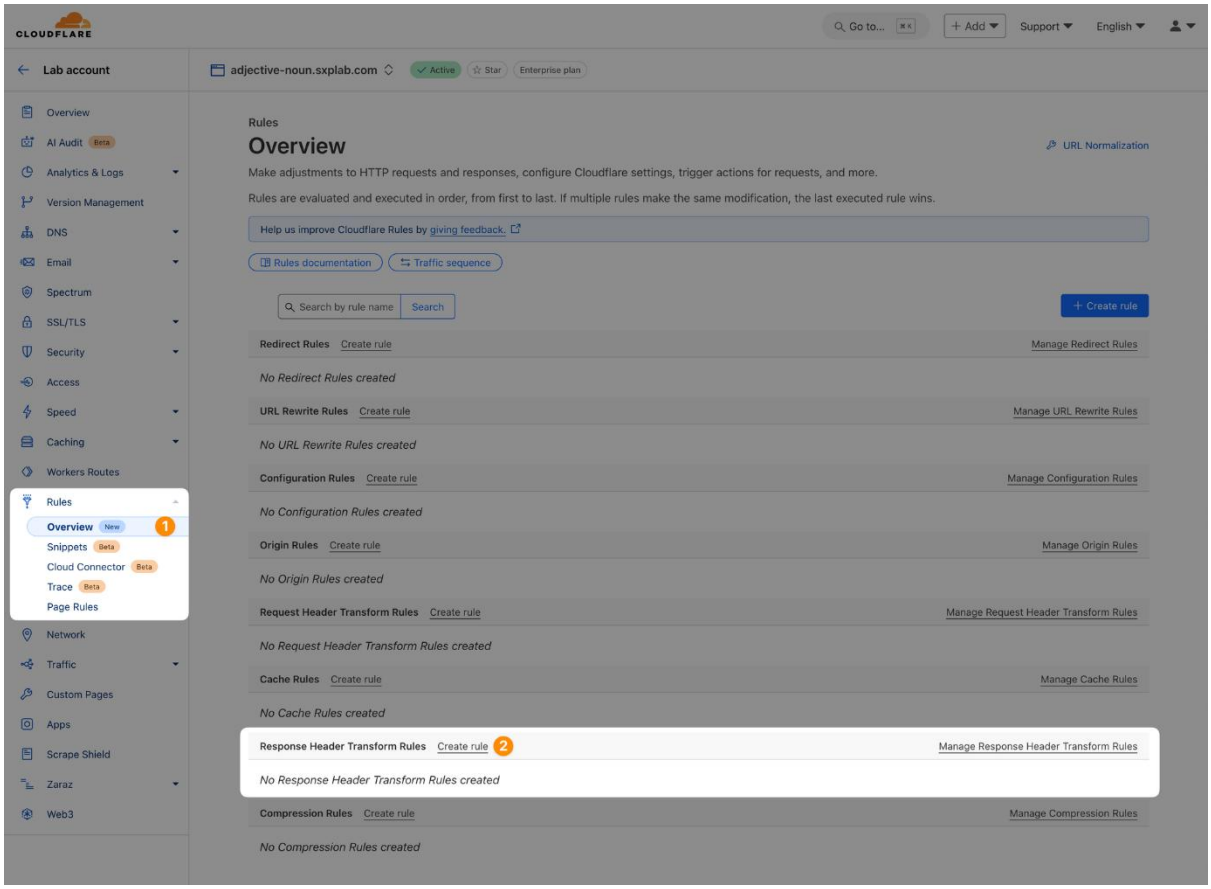
Passos

1. Crie uma regra de transformação que adicione um cabeçalho a todas as respostas.

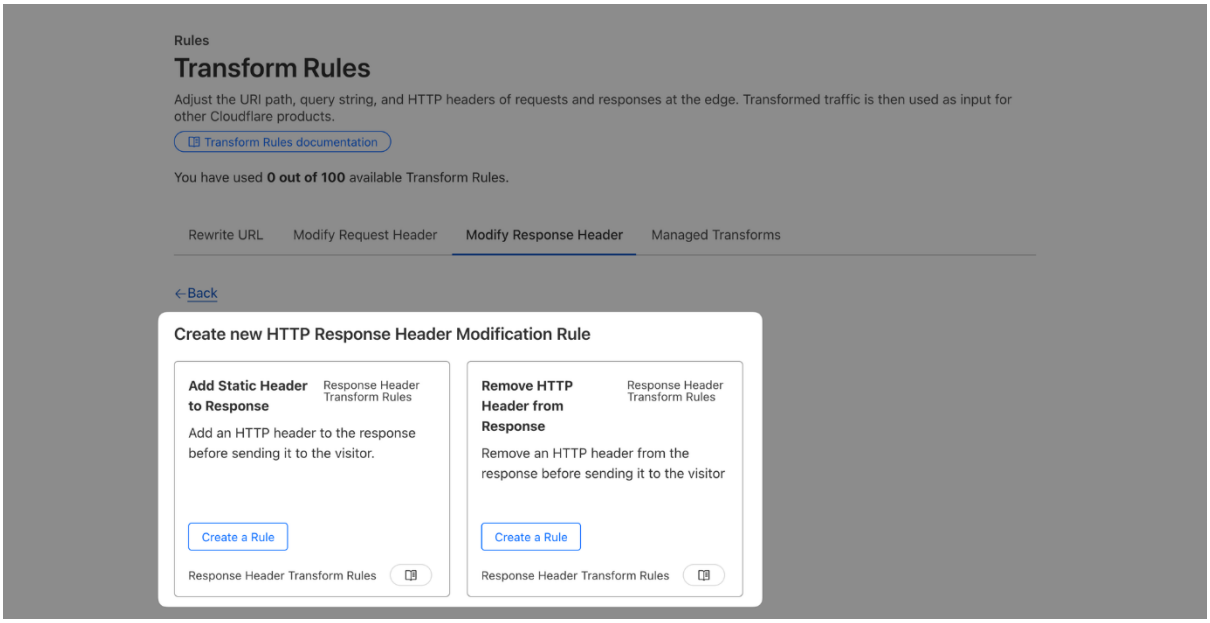
As Transform Rules permitem ajustar o caminho da URI, a string de consulta e os cabeçalhos HTTP de solicitações e respostas na rede global da Cloudflare.

Nesta tarefa, criaremos uma regra para adicionar o cabeçalho de segurança comum X-Frame-Options a todas as respostas. Em seguida, veremos como podemos facilitar isso com Managed Transforms...

No painel do Cloudflare, acesse **Regras ▸ Visão geral** , role para baixo até **Response Header Transform Rules** e clique em **Criar regra**.



Aqui você pode ver alguns modelos predefinidos para casos de uso comuns:

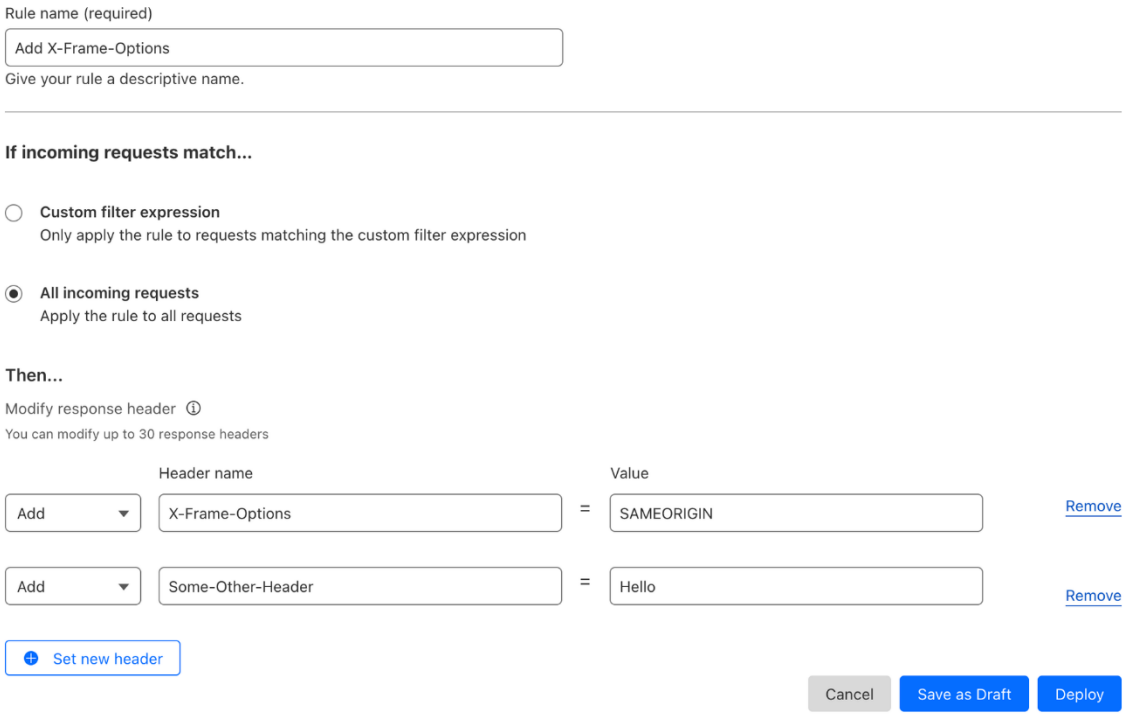


Poderíamos usar o modelo para adicionar um cabeçalho estático, mas, em vez disso, vamos criar uma nova regra do zero. Role a página para baixo até os modelos e crie uma nova regra com os seguintes valores:

- Defina **o nome da regra** para algo descritivo, como "Adicionar opções de quadro X".
- Defina " **Se as solicitações recebidas corresponderem...**" para **"Todas as solicitações recebidas".**
- Defina os critérios de **"Então..."** para...
- **Adicionar**
- Defina o **nome do cabeçalho** como **X-Frame-Options.**
- Defina o **valor** como **SAMEORIGIN**

Fique à vontade para adicionar outros cabeçalhos arbitrários, se desejar.

Sua regra deve ser semelhante a esta:



Clique em **Implantar** .

2. Verificar o cabeçalho em uma resposta

Verifique os cabeçalhos de resposta HTTP usando curl (ou através do método do navegador compartilhado anteriormente):

```
curl -I https://organized-keystone.sxplab.com/services/luggages/
```

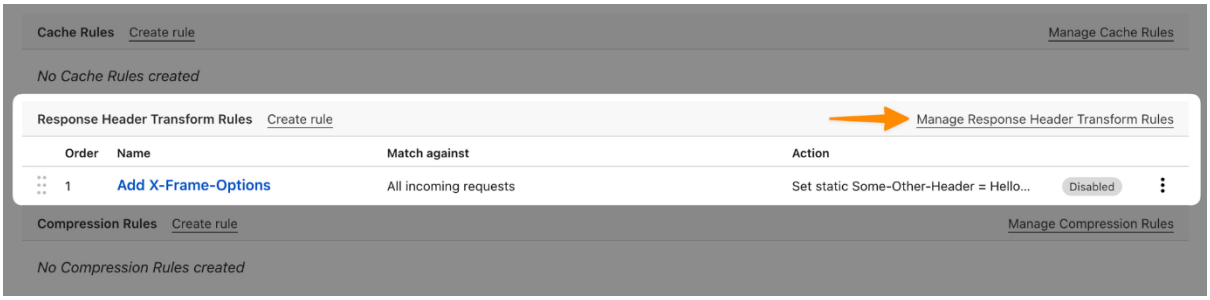
```
HTTP/2 200
date: Fri, 24 Jan 2025 04:36:50 GMT
content-type: text/html
last-modified: Tue, 08 Oct 2024 04:29:49 GMT
x-origin: origin-1
accept-ranges: bytes
cf-cache-status: DYNAMIC
x-frame-options: SAMEORIGIN
some-other-header: Hello
server: cloudflare
```

Você pode ver que o cabeçalho `x-frame-options <header>` e quaisquer outros cabeçalhos que você definiu agora estão presentes.

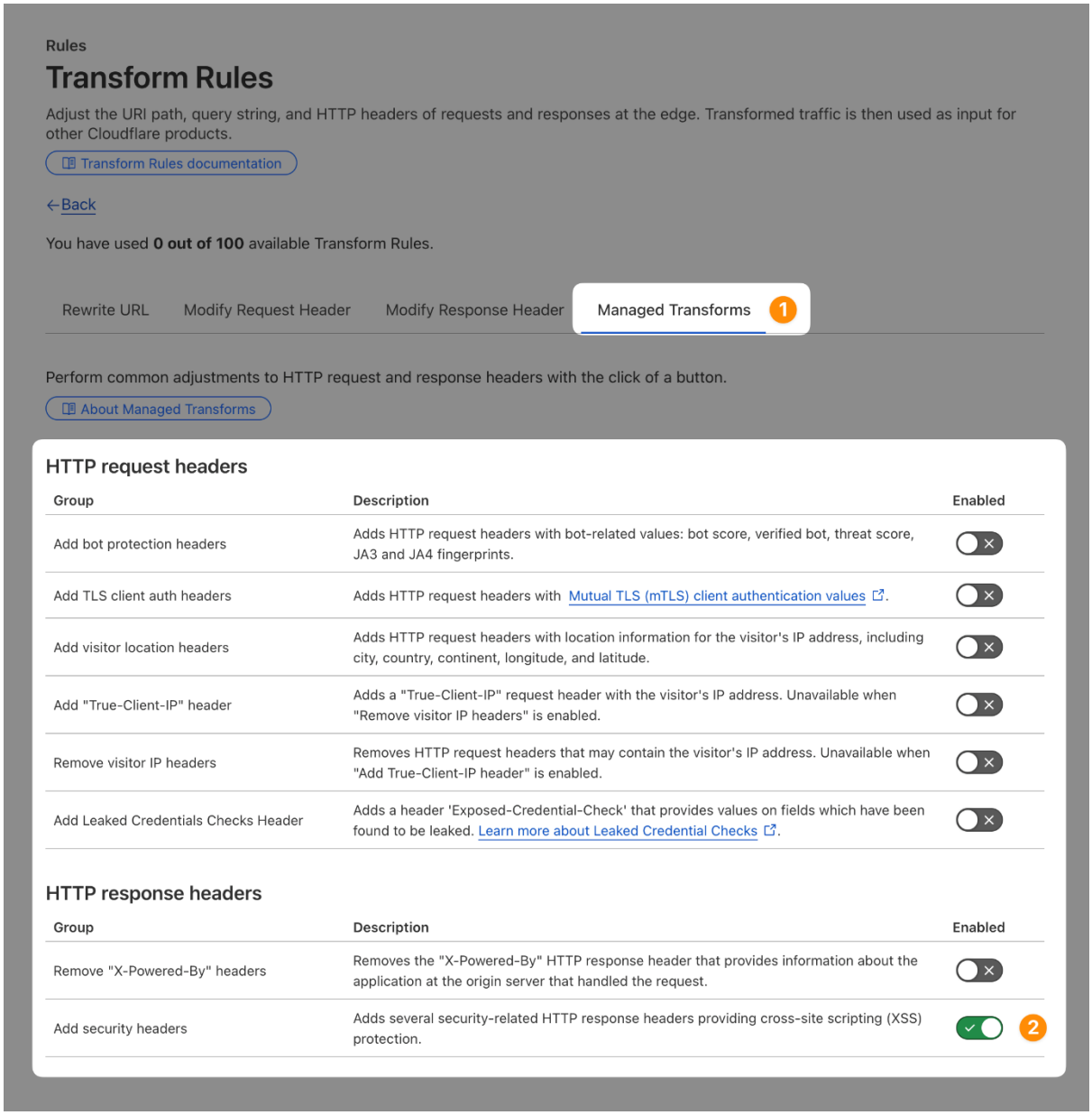
3. Adicione mais cabeçalhos de segurança com Managed Transforms.

Agora que você já sabe como adicionar cabeçalhos, vamos adicionar mais alguns!

Primeiro, **desative** a regra que você acabou de implantar. Em seguida, acesse **Regras ▸ Visão geral** e, em **Response Header Transform Rules**, selecione **Gerenciar Response Header Transform Rules**.



Acesse a guia **Managed Transforms**. Como você pode ver, há várias opções para escolher, com base em cenários comuns. Ative a opção **Adicionar cabeçalhos de segurança** na seção **Cabeçalhos de resposta HTTP**.



Agora repita o comando curl (ou visualize no seu navegador) - observe a presença dos novos cabeçalhos, como x-xss-protection:

```
curl -I "https://organized-keystone.sxplab.com/services/luggages/"
```

```
HTTP/2 200
date: Fri, 24 Jan 2025 04:49:41 GMT
content-type: text/html
last-modified: Tue, 08 Oct 2024 04:29:49 GMT
metadata: SWYgZGlzY292ZXJlZCwgc2F5IGhpIGV1Z2dAY2xvdWRmbGFyZS5jb20h
x-origin: origin-1
accept-ranges: bytes
cf-cache-status: DYNAMIC
expect-ct: max-age=86400, enforce
referrer-policy: same-origin
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
server: cloudflare
...
```

4. Adicione uma regra de reescrita de URL

Às vezes, é desejável exibir uma página diferente daquela que está na URL. Agora, vamos explorar a funcionalidade de reescrita de URL para exibir a /services/luggages página ao fazer a solicitação /services/arrows.

Acesse **Regras ▸ Visão geral** e, em **Regras de reescrita de URL,** selecione **Criar regra** .

Criar e implementar uma nova regra:

- Defina **o nome da regra** como algo descritivo, como "Exibir página de bagagens no URL da Arrows".
- Defina " **Se as solicitações recebidas corresponderem a..."** para **Expressão de filtro personalizada.**
- Defina o **campo** como **Caminho URI.**
- Defina o **operador** como **igual a**
- Defina o **valor** para **/services/arrows/**
- Defina os critérios de **"Então..."** para...
- **Reescrever para...**
- **Estático** services/luggages/

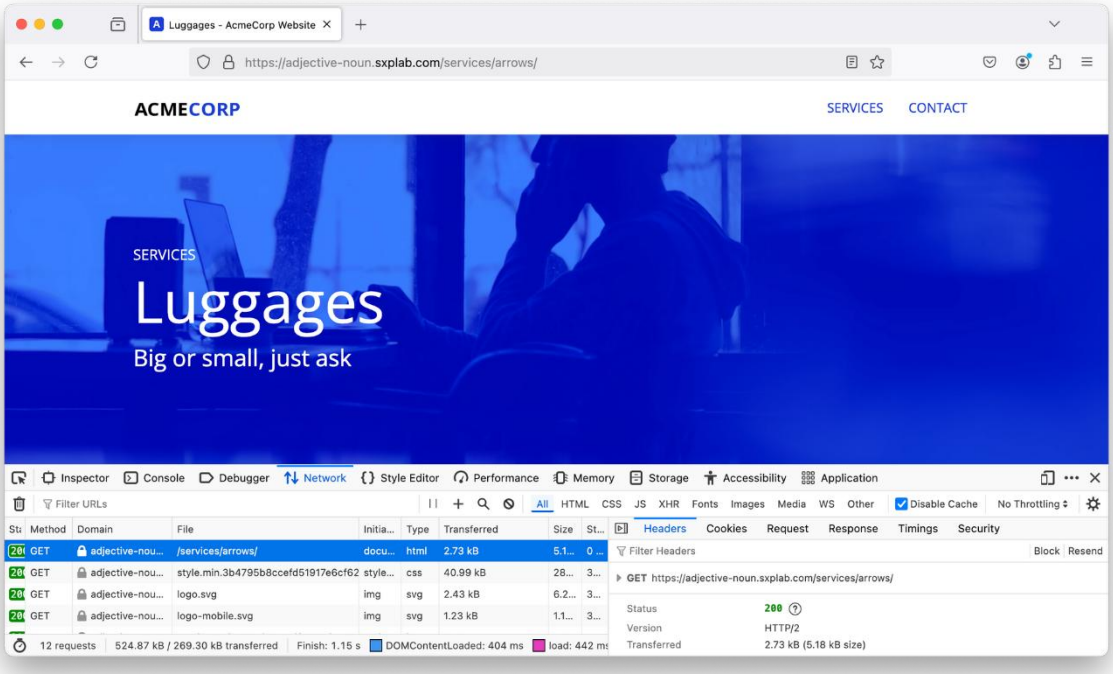
Sua configuração deve ser semelhante a esta:

- Deixe os outros valores como padrão (mas fique à vontade para explorá-los!) e clique em **Implantar.**

Acesse o site da AcmeCorp usando a URL completa:

```
https://opportunistic-zipline.sxplab.com/services/arrows/
```

Observe que a Luggages página é exibida, apesar da URL arrows ter sido mantida. Note também que não se trata de um redirecionamento HTTP, apenas um conteúdo diferente retornado para essa URL.



5. (Opcional) Adicionar mais regras

Explore as várias opções dentro das regras - você consegue criar uma regra para redirecionar /services/calculators/ de volta para a página inicial da AcmeCorp?