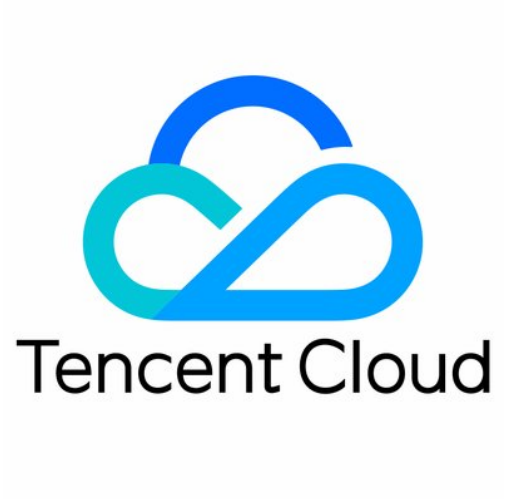
Tencent Cloud EdgeOne

Data Analysis&Log Service

Product Documentation



Contents

Data Analysis&Log Service
- Log Service
  - Overview
  - Real-time Logs
    - Real-time Logs Overview
    - Push to Tencent Cloud CLS
    - Push to AWS S3-Compatible COS
    - Push to HTTP Server
  - Offline Logs
  - Related References
    - Field description
      - L7 Access Logs
      - L4 Proxy Logs
      - Edge Function Running Logs
    - Real-Time Log Push Filter Conditions
    - Custom Log Push Fields
    - Customizing Log Output Formats
- Data Analysis
  - Overview
  - Analytics
  - Web Security Analysis
  - Related References
    - Sampling Statistics
    - How to use filter condition
    - How to Modify Query Time Range
    - How to Export Statistical Data and Reports
- Alarm Service
  - Custom Statistical Metrics

Data Analysis&Log Service
Log Service
Overview

Last updated：2025-04-24 17:07:57

The L4/7 acceleration, web protection, edge function,and other feature modules of EdgeOne global availability zone nodes support recording detailed logs when processing requests for your services. The CLS module collects and aggregates logs from various feature modules, and then provides the logs to users. You can use the log details for troubleshooting, checking the impact of configuration updates, generating monitoring metrics, etc.

Supported Features

Real-time Log Push: Ships the access request logs to your specified destination with low latency, and supports configuration through the console or API. The latency from initiating a request to receiving the logs by the destination is within 5 minutes. It is suitable for scenarios requiring high timeliness such as real-time troubleshooting and monitoring. The recording scopes for various types of request logs are described as follows.

Site Acceleration Logs: Records the domain access logs. By default, only the logs of requests after protection are recorded. Logs of the requests blocked by Anti-DDoS are not recorded.

Note:

The feature of Real-time Logs - Site Acceleration Logs to record full L7 request logs (including L7 protection block logs) is in beta testing. If needed, please contact us.

L4 Proxy Logs: Records the access logs of L4 proxy instances. Only the logs of accesses after protection are recorded. Logs of the accesses blocked by Anti-DDoS are not recorded.

Edge Function Running Logs: Records logs of the edge function execution situation.

Rate Limiting and CC Attack Protection Logs: Only records the request logs that match the security rules of the L7 Protection - Rate Limiting and CC Attack Protection module, no matter whether the requests are blocked or not.

Managed Rule Logs: Only records the request logs that match the security rules of the L7 Protection - Managed Rules module, no matter whether the requests are blocked or not.

Custom Rule Logs: Only records the request logs that match the security rules of the L7 Protection - Custom Rules module, no matter whether the requests are blocked or not.

Bot Management Logs: Only records the request logs that match the security rules of the L7 Protection - Bot Management module, no matter whether the requests are blocked or not.

Offline Logs: By default, the access logs are retained for 30 days. You can obtain the download URL for the log package via the console or API. Usually, the download URL for the log package is available 3 hours after a request is

initiated, and the integrity of the logs within the log package will be guaranteed after 24 hours. It is suitable for scenarios not requiring high timeliness, such as long-term log retention and periodic reconciliation.
Site Acceleration Logs: Records the domain access logs. Only the logs of requests after protection are recorded. Logs of the requests blocked by Anti-DDoS are not recorded.
L4 Proxy Logs: Records the access logs of L4 proxy instances. Only the logs of accesses after protection are recorded. Logs of the accesses blocked by Anti-DDoS are not recorded.
Note:
For real-time and offline log field descriptions, please refer to Field description.

Package Support Differences

Sub-feature	Individual Edition	Basic Edition	Standard Edition	Enterprise Edition
Real-time Log Push	2 Task/Log Types	2 Task/Log Types	3 Task/Log Types	5 Task/Log Types
Offline Logs	Supported. The log retention duration is 30 days.		Supported. The default log retention duration is 31 days, and the maximum log retention duration is 183 days.	

Billing Description

Real-time Log Push

After accessing EdgeOne, you will obtain the real-time log push feature by default, without any additional charges. It's important to note that after you configure real-time log push tasks, the destination of log shipping may also incur charges. For example, after configuring log push to Tencent Cloud CLS, traffic and storage charges may be generated for the Tencent Cloud CLS product. For details, refer to CLS Billing Overview.

Offline Logs

After accessing EdgeOne, you will obtain the offline logs feature by default, without any additional charges.

Real-time Logs
Real-time Logs Overview

Last updated：2025-01-24 15:11:21

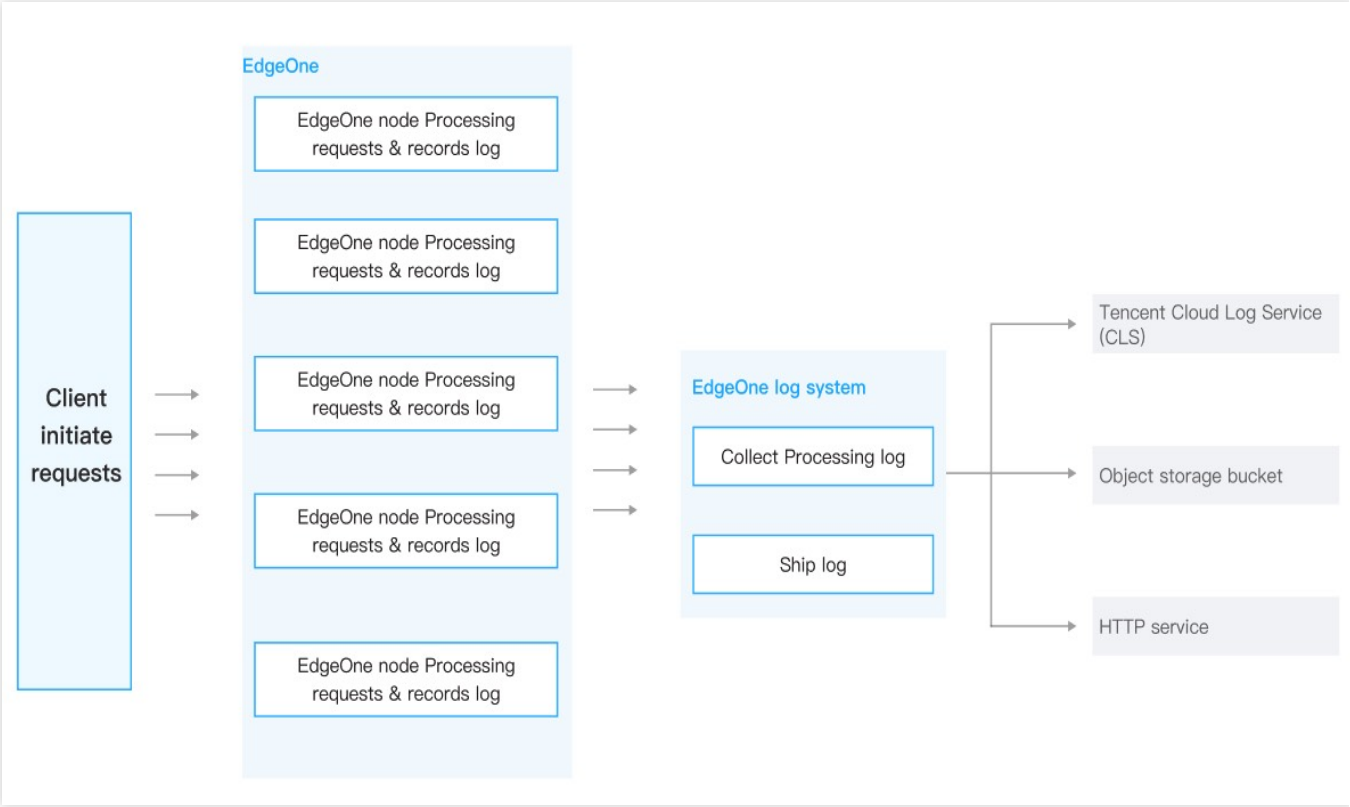
Function Overview

After adding your site to EdgeOne Service, EdgeOne provides you with a wealth of pre-built reports to help you monitor and analyze the operation of your business.In data analysis, you may have more personalized data analysis demands, such as the following data analysis scenarios:

Scenario	Scenario Demands
Deep Data Analysis	Need to specify one or more conditions to find logs that meet the conditions. For example: By specifying the client IP, query the access statistics (access URL, number of accesses, etc.) within a specified time range. Refine the analysis of status code distribution by filtering status codes, time, and URLs. By filtering logs with the action set to "observe", summarize the request header content and other request features carried, and adjust the security policy.
Monitoring Service Metrics	Analyze the quality of EdgeOne Service and the access efficiency of users to detect exceptions in a timely manner. Access efficiency includes overall response time, download speed, origin-pull response time, etc.
Identifying Unauthorized Access	Identify client IPs with behaviors such as unauthorized access by analyzing traffic anomalies, access patterns, and access frequency.
Unified Monitoring of Data from Multiple Cloud Vendors	Build your own data dashboard to monitor application data from multiple cloud vendors.
Storing Logs	Need to reserve full user access logs (including attack interception logs) for more than 30 days.

For the above scenario demands, EdgeOne real-time logging provides the ability to collect and ship logs in real-time, allowing you to ship your logs to Tencent Cloud Log Service (CLS) or your self-built data center, helping you to implement flexible log data retrieval and analysis on your own. Currently, EdgeOne supports shipping logs to the following destinations:

Push to Tencent Cloud CLS: Ship logs to the one-stop log processing service (CLS) provided by Tencent Cloud for further log analysis on CLS.
Push to AWS S3 Compatible COS: Storage buckets compatible with AWS Signature V4 authentication method.
HTTP Service (POST): Ship logs to the specified backend server via HTTP POST requests.



Note：
1. Under normal circumstances, the latency of log delivery is within 5 minutes. To ensure the real-time performance of log delivery, EdgeOne sends logs to the designated destination in batches based on a fixed number of logs or a fixed time period. The default policy prioritizes by the number of logs, 1000 logs per batch; when the number of logs is less than 1000, but the time interval since the last push is 5 seconds, it will trigger a second push.
2. For real-time log fields, refer to Field description.

Billing and Quota Description

Referer to Package Support Differences, Billing Instructions.

Push to Tencent Cloud CLS

Last updated：2025-01-24 15:06:29

EdgeOne Real-Time Log Push supports pushing logs to Tencent Cloud Log Service (CLS). You can configure it through the console or API. For more information about CLS, refer to CLS Product Documentation.

Prerequisites

1. Log in to the Tencent Cloud CLS console and activate CLS.
2. If you wish to use a sub-account for CLS-related operations, refer to CLS Permission Management Guide, to complete sub-account authorization and ensure that the sub-account has related read and write permissions for CLS log sets and log topics.

Note:

You must authorize EdgeOne to access your log sets and log topics through the service role `TEO_QCSLinkedRoleInRealTimeLogCLS`. EdgeOne will use the service role to query log sets and log topics, modify index configurations, and push logs.

Directions

1. Log in to the EdgeOne console and click **Site List** in the left sidebar. Then click on the **site** to be configured in the site list, to enter the site details page.
2. On the site details page, click **Log Service** > **Real-time Logs**.
3. On the real-time logs page, click **Create Push Task**.
4. On the log source selection page, enter a task name, select a log type, service area, and domain name/L4 proxy instance requiring log push, and click **Next**.
5. On the push content definition page:

(Required) Check the log fields to be pushed from the predefined field list.

(Optional) Add a custom log field, which supports extracting specified field names from the request headers, response headers, and Cookie headers.

(Optional) Configure the log push filter conditions. Full logs are pushed by default.

(Optional) In advanced configuration, set the sampling ratio. By default, sampling is not enabled and 100% of logs are pushed to the destination.

(Optional) In advanced configuration, set the log output format. The default format is JSON Lines.

Note:

For pushing logs to CLS, only the JSON format can be selected. Prefixes, suffixes, and separators are not effective.

6. On the destination selection page, select **CLS** and click **Next**.
7. On the destination information page, select the region, log set, and log topic for the destination log set.

Note:

Due to the write operation permission restrictions of Serverless Cloud Function (SCF) on the CLS log topic, during configuration of the EdgeOne real-time log shipping destination, **do not select** the SCF Default Log Topic or other cloud product default log topics to avoid shipping failure or inability to search logs.

8. Click **Push**.
9. In the pop-up window, select the index configuration method. It is recommended to click **Quick Index Configuration**. EdgeOne will create a key-value index for the previously selected log topic. You can also configure the index yourself in the CLS Console. Note that if Key-Value Index is not enabled, you will fail to search logs.

Note:

When the log volume is too large, and the log topic auto-split feature of CLS is disabled or the partition value has reached its limit, CLS will restrict the log push request frequency, which may result in log data loss. To avoid such issues, refer to CLS Log Topic for configuration.

Push to AWS S3-Compatible COS

Last updated：2025-01-24 14:57:43

EdgeOne Real-time Log Push can be configured through the console or API to push logs to AWS S3 Signature Version 4 Authentication Algorithm compatible COS, such as:
Tencent Cloud COS
AWS S3
Google Cloud Storage
IBM Cloud Object Storage
Linode Object Storage
Oracle Cloud Object Storage, etc

Operation step

1. Access the EdgeOne console, locate the left-hand menu, and select the **Site List**. Within this listing, click on the S**ite** requiring configuration to enter its detailed page.
2. On the Site Details Page, click **Log Service** > **Real-time Logs.**
3. On the Real-time Logs page, click **Create shipping task.**
4. On Select the log source Page, fill in the Task name, select the Log type Service area, the Domain name/Layer 4 Proxy Instance/edge function instance for which the logs need to be pushed, and click **Next**.

Note:
Currently, only Site Acceleration Logs、L4 Proxy Logs and edge function runtime logs (in beta) are supported to be pushed to S3 Compatible COS.

5. On the Define delivery content page:
(Required) Check the log fields to be pushed from the Predefined field list;
(Optional) Add a Custom Log Field, supporting the extraction of specified field names from the Request Header, Response Header, and Cookie Header;
(Optional) Configure the Push Log Filter Criteria, the default is to push full logs;
(Optional) In advanced configuration, set the sampling ratio. By default, sampling is not enabled, and 100% of logs are pushed to the destination.
(Optional) In advanced configuration, set the log output format. The default format is JSON Lines.
6. On the Select a destination address page, select **S3 compatible** , click **Next** .
7. On the Destination Information Page, enter the related destination and parameter information.

Parameter name	Description
Endpoint URL	URL without bucket name or path, e.g.: `https://cos.ap-nanjing.myqcloud.com` .

bucket region	Region where the bucket is located, for example: `ap-nanjing` .
Bucket	Bucket name and log storage directory, for example: `your_bucket_name/EO-logs/` . Whether or not you end the directory with `/` , it will be correctly parsed and processed.
File compression	After checking, gzip will be used to compress log files.
Secret ID	Access Key ID used to access the bucket.
Secret key	Secret key used to access the bucket.

8. Click **Ship**.
9. After issuing the real-time log push task, EdgeOne will push a test file to the target bucket directory to verify connectivity, for example, `1699874755_edgeone_push_test.txt` , with the file content being the fixed string "test".

File Name Description

Logs will be stored in the specified bucket directory in the format `{{UploadTime}}_{{Random}}.log` , and will be archived by date (UTC+00:00) into folders. For example: `20230331/20230331T185917Z_2aadf5ce.log` . When gzip compression is enabled, the file name will be `20230331/20230331T185917Z_2aadf5ce.log.gz` .
UploadTime: The upload time of the log file, using ISO-8601 Format, UTC+00:00 Time Zone.
Random: Random characters used to identify different files when there are multiple log files for the same upload time due to large log volume.

Push to HTTP Server

Last updated：2025-01-24 14:56:05

EdgeOne Real-time Log Push supports sending logs to a Custom Interface address, and you can configure it through the console or API. EdgeOne can use an HTTP POST request to call the backend interface address you provide, transferring logs in the HTTP Body to the server you specify.

Operation step

1. Access the EdgeOne console, locate the left-hand menu, and select the **Site List**. Within this listing, click on the **Site** requiring configuration to enter its detailed page.
2. On the Site Details Page, click **Log Service** > **Real-time Logs.**
3. On the Real-time Logs page, click **Create shipping task.**
4. On Select the log source Page, fill in the Task name, select the Log type Service area, the Domain name/Layer 4 Proxy Instance/edge function instance for which the logs need to be pushed, and click **Next**.
5. On the Define delivery content page:
(Required) Check the log fields to be pushed from the Predefined field list;
(Optional) Add a Custom Log Field, supporting the extraction of specified field names from the Request Header, Response Header, and Cookie Header;
(Optional) Configure the Push Log Filter Criteria, the default is to push full logs;
(Optional) In advanced configuration, set the sampling ratio. By default, sampling is not enabled, and 100% of logs are pushed to the destination.
(Optional) In advanced configuration, set the log output format. The default format is JSON Lines.
6. On the Select a destination address page, select **HTTP (POST)**, click **Next** .
7. On the Destination Information Page, enter the related destination and parameter information.

Parameter name	Description
Address	Enter your log receiving interface address, for example: `https://www.example.com/edgeone-logs`
File compression	To reduce the size of log content and save on traffic costs, you can enable content compression by checking **Use Compress log files with gzip**. EdgeOne will use the gzip format to compress and then transmit the logs, and it will add the HTTP header `Content-Encoding: gzip` to indicate the compression format.
Origin authentication	When Encryption Authentication is selected, the push logs will carry authentication information for the origin server to verify, ensuring the security of the data source identity. For the authentication algorithm, see: Authentication Algorithm Reference.

Custom HTTP header	Add the HTTP headers that need to be carried when EdgeOne initiates a request. For example: Identify the log source as EdgeOne by adding the header `log-source: EdgeOne`. Obtain the number of log entries pushed in each POST request by adding the header `BatchSize: ${batchSize}` **Note** If the header name you fill in is a default header carried by EdgeOne log push, such as Content-Type, the header value you specify will override the default value.

8. Click **Ship**.

9. During the configuration phase of the Real-time Log Push Task, test data will be sent to the interface address to verify interface connectivity. The data format is as follows:

```
{
    "ClientState": "CH-AH",
    "EdgeResponseTime": 366,
    "RequestID": "13515444256055847385",
    "ClientRegion": "CN",
    "RemotePort": 443,
    "RequestHost": "www.tencent.com",
    "RequestMethod": "GET",
    "RequestUrlQueryString": "-",
    "RequestUrl": "/en-us/about.html",
    "RequestProtocol": "HTTP/2.0",
    "EdgeServerID": "336d5ebc5436534e61d16e63ddfca327-d41d8cd98f00b204e9800998ecf84
    "RequestTime": "2022-07-01T02:37:13Z",
    "EdgeCacheStatus": "-",
    "EdgeResponseBytes": 39430,
    "EdgeResponseStatusCode": 200,
    "ClientIP": "0.0.0.0",
    "RequestReferer": "https://www.tencent.com/",
    "RequestUA": "Mozilla/5.0 (iPhone; CPU iPhone OS 15_5 like Mac OS X) AppleWebKi
    "EdgeServerIP": "0.0.0.0",
    "RequestRange": "0-100/200",
    "EdgeInternalTime": 334,
    "RequestBytes": 237
}
```

Related References

Server-side Log Parsing Code Example

When origin server authentication is not enabled, you can refer to the following Python code to parse the log content in the request body on the server side.

```
# Import modules from the Python standard library
import time  # Used to get the current time
import gzip  # Used to handle Gzip compressed data


# Import HTTPServer and BaseHTTPRequestHandler classes from the http.server module
from http.server import HTTPServer, BaseHTTPRequestHandler
import json  # Used to handle JSON data format


# Define a class that inherits from BaseHTTPRequestHandler, used to handle HTTP req
class Resquest(BaseHTTPRequestHandler):
    # Override the do_POST method, which is called when the server receives a POST
    def do_POST(self):
        # Print request header information
        print(str(self.headers))
        # Print the command from the HTTP request (e.g., POST)
        print(self.command)
        # Read the request body content, determining the length to read as specifie
        req_datas = self.rfile.read(int(self.headers['content-length']))
        try:
            # Attempt to decode the request body content and print it
            print(req_datas.decode())
        except Exception as e:
            # If an exception occurs during decoding, print the exception informati
            print(e)
            # Check if the request header contains Content-Encoding: gzip, if so, d
            if self.headers['Content-Encoding'] == 'gzip':
                data = gzip.decompress(req_datas)
                # Print the decompressed gzip content
                print('---------------decompress gzip content----------------------
                print(data.decode())
        # Check if the request path is '/edgeone-logs', if not, return a 404 error
        if self.path != '/edgeone-logs':
            self.send_error(404, "Page not Found!")
            return
        # If the request path is correct, prepare the response data
        data = {
            'result_code': '1',
            'result_desc': 'Success',
            'timestamp': int(time.time())  # Respond with the current timestamp
        }
        # Send an HTTP response status code 200, indicating the request was success
        self.send_response(200)
```

```
        # Set the response header Content-type to application/json
        self.send_header('Content-type', 'application/json')
        # End the sending of response headers
        self.end_headers()
        # Write the response data in JSON format to the response body
        self.wfile.write(json.dumps(data).encode('utf-8'))


# Check if the current script is running as the main program
if __name__ == '__main__':
    # Define the server listening address and port. You can replace 9002 with your
    host = ('', 9002)
    # Create an HTTPServer object, passing in the listening address, port, and the
    server = HTTPServer(host, Resquest)
    # Print server startup information
    print("Starting server, listen at: %s:%s" % host)
    # Start the server and keep it running until externally interrupted
    server.serve_forever()
```

Request authentication algorithm

If you select **Encryption Signature** for origin server authentication in the Push Destination Information, you can enter the SecretId and SecretKey configured by yourself in Definition. EdgeOne will add `auth_key` and `access_key` to the request URL. The details of the signature algorithm are as follows:

1. **Request URL Composition**

As shown below, the request URL will carry `auth_key` and `access_key` after `?` .

```
http://DomainName[:port]/[uri]?auth_key=timestamp-rand-md5hash&access_key=SecretId
```

Parameter description:

timestamp: The current request time, using Unix second-level 10-digit timestamp.

rand: Random number.

access_key: Used to identify the identity of the interface requester, that is, the SecretId configured by your definition.

SecretKey: Fixed length of 32 characters, that is, the SecretKey configured by your definition.

uri: Resource identifier, for example: `/access_log/post` .

md5hash: md5hash = `md5sum(string_to_sign)` , where string_to_sign = `"uri-timestamp-rand-SecretKey"` . A verification string calculated through the MD5 algorithm, consisting of digits 0-9 and lowercase letters a-z, with a fixed length of 32 characters.

2. **Computational Example**

Assume the parameters are filled as:

API Address: `https://www.example.com/access_log/post`

SecretId = `YourID`

SecretKey = `YourKey`

uri = `/access_log/post`

timestamp = `1571587200`

rand = `0`

```
string_to_sign = "/access_log/post-1571587200-0-YourKey"
```

Based on this string, calculate:

```
md5hash=md5sum("/access_log/post-1571587200-0-YourKey")=1f7ffa7bff8f06bbfbe2ace0f14
```

Then the final URL for the push request is:

```
https://www.example.com/cdnlog/post?auth_key=1571587200-0-1f7ffa7bff8f06bbfbe2ace0f
```

After the server receives the push request, it extracts the value of `auth_key` . Split the value of auth_key to obtain `timestamp` , `rand` , and `md5hash` . First, check whether the `timestamp` has expired; the recommended expiration time is `300s` . Then assemble the encrypted string based on the aforementioned rules and use the SecretKey to construct the string to be encrypted. After encryption, compare it with the `auth_key` value in the `md5hash` . If they match, the authentication is successful.

3. **Server-side Code Example for Parsing Authentication Requests**

Python

Golang

```
import hashlib

from flask import Flask, request

app = Flask(__name__)


def get_rsp(msg, result={}, code=0):
    return {
        "respCode": code,
        "respMsg": msg,
        "result": result
    }


def get_secret_key(access_key):
    return "secret_key"


@app.route("/access_log/post", methods=['POST'])
def access_log():
```

```
    if request.method == 'POST':
        if request.content_type.startswith('application/json'):
            current_time_ts, rand_num, md5hash = request.args.get("auth_key").split
             # Determine if the request time is within the validity period
            if time.time() - int(current_time_ts) > 300:
                return get_rsp(msg="The request is out of time", code=-1)

            access_key = request.args.get("access_key")
            # Retrieve the secret_key using access_key (SecretId)
            secret_key = get_secret_key(access_key)
            raw_str = "%s-%s-%s-%s" % (request.path, current_time_ts, rand_num, sec
            auth_md5hash = hashlib.md5(raw_str.encode("utf-8")).hexdigest()
            if auth_md5hash == md5hash:
                # Authentication successful
                if request.headers['content-encoding'] == 'gzip':
                    # Decompress data
                    pass
                # Data processing
            return get_rsp("ok")
        return get_rsp(msg="Please use content_type by application/json", code=-1)
    return get_rsp(msg="The request method not find, method == %s" % request.method

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8888, debug=True)
```

```
package main

import (
    "context"
    "crypto/md5"
    "fmt"
    "log"
    "net/http"
    "os"
    "os/signal"
    "strings"
    "syscall"
)

func main() {
    mux := http.NewServeMux()
    mux.Handle("/access_log/post", &logHandler{})

    server := &http.Server{
        Addr:    ":5000",
```

```
        Handler: mux,
    }

    // Create system signal receiver
    done := make(chan os.Signal)
    signal.Notify(done, os.Interrupt, syscall.SIGINT, syscall.SIGTERM)
    go func() {
        <-done

        if err := server.Shutdown(context.Background()); err != nil {
            log.Fatal("Shutdown server:", err)
        }
    }()

    err := server.ListenAndServe()
    if err != nil {
        if err == http.ErrServerClosed {
            log.Print("Server closed under request")
        } else {
            log.Fatal("Server closed unexpected")
        }
    }
}

type logHandler struct{}

func (*logHandler) ServeHTTP(w http.ResponseWriter, r *http.Request) {
    if r.Method == "POST" {
        query := r.URL.Query()
        authKey := query.Get("auth_key")
        accessKey := query.Get("access_key") // access_key is your provided SecretI
        authKeys := strings.Split(authKey, "-")
        if len(authKeys) == 3 {
            currentTimeTs := authKeys[0]

            // Perform timestamp validity judgment
            RandNum := authKeys[1]
            md5Hash := authKeys[2]
            secretKey := getSecretKey(accessKey)
            authStr := fmt.Sprintf("%s-%s-%s-%s", "/access_log/post", currentTimeTs
            data := []byte(authStr)
            has := md5.Sum(data)
            authMd5 := fmt.Sprintf("%x", has) // Convert to string for comparison
            if authMd5 == md5Hash {
                // TODO Authentication successful
                if r.Header.Get("Content-Encoding") == "gzip" {
                    // Decompress data
```

```
                }
                // Data processing
            }
        } else {
            // Exception Handling
        }
    }
}

// Retrieve SecretKey
func getSecretKey(accessKey string) string {
    if accessKey != "" {
        // Retrieve Secret_Key using Access_key (SecretId)
        return "secret_key"
    }
    return ""
}
```

Offline Logs

Last updated：2024-09-05 14:19:38

Feature Overview

To facilitate user access analysis by customers, EdgeOne packages access logs on an hourly basis, and provides download services.

Offline Log Format

Logs are stored in JSON Lines format by default. Each JSON line represents a single log.
Log packages are compressed in .gz format with gzip. Due to the directory system limitations in MacOS, double-click for decompression may cause an error. In this case, you can navigate to the directory containing the logs, and use the following Terminal command for decompression.

```
gunzip {your_file_name}.log
```

Log Packaging Rules

Logs are packaged on an hourly basis by default. If there are no requests for accessing your business within an hour, no log packages will be generated for the time interval.
Since EdgeOne nodes are distributed globally, the storage time of offline logs (the time in log package filenames) is set to UTC +00:00 by default in order to synchronize all time zones.
Offline logs are collected from various EdgeOne nodes so that they are different in latency. Generally, logs can be queried and downloaded after a delay of around 3 hours. The log packages will increase continuously and typically stabilize after around 24 hours.

Log Retention Period

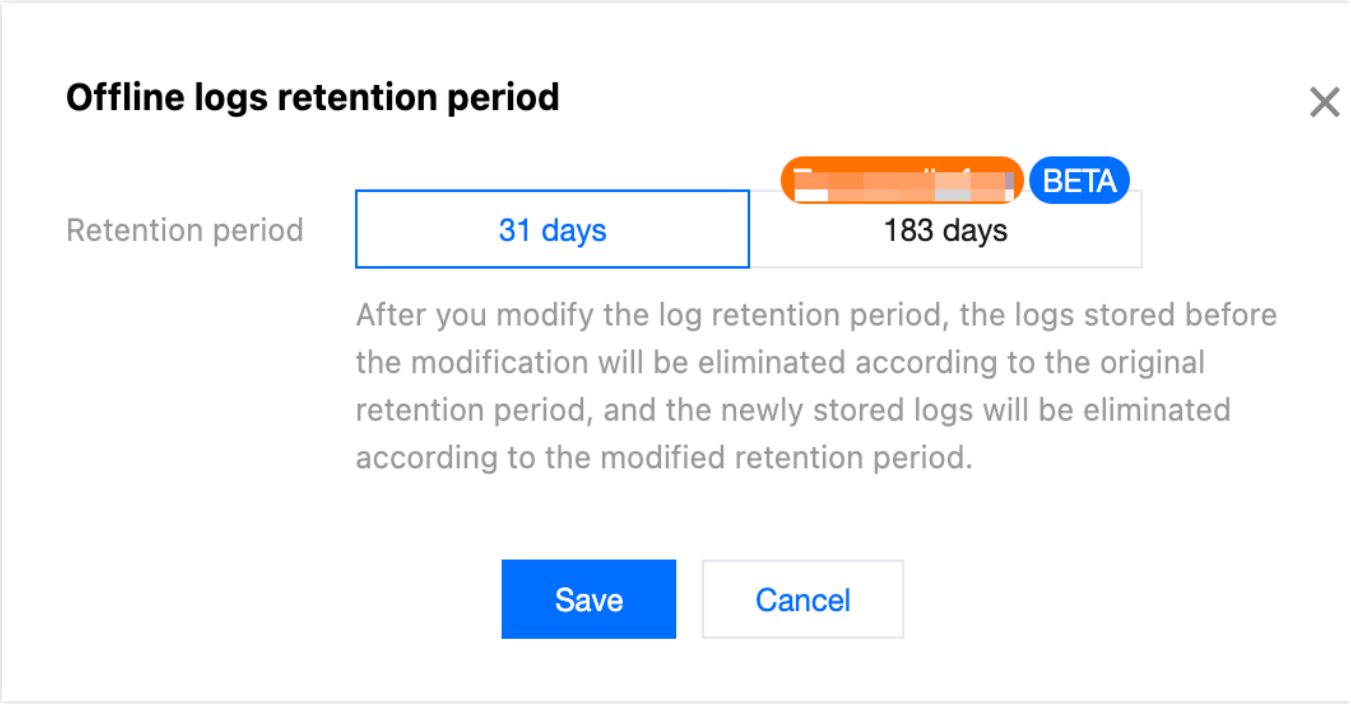
The default retention period of offline logs is 31 days. For sites with Standard and Enterprise plans, the retention period of offline logs can be adjusted to 183 days to meet relevant compliance requirements. For details, see Plan Support Differences.

Directions

1. Log in to the EdgeOne console and click **Site List** in the left sidebar. Then click the **site** of your concern in the site list.
2. On the site details page, click **Log Service** > **Offline Logs**.
3. On the offline logs page, click **Settings**


on the right side of the filter field.

4. In the pop-up window, modify the retention period of offline logs and click **Save** . After you modify the log retention period, the logs stored before the modification will be eliminated according to the original retention period, and the newly stored logs will be eliminated according to the modified retention period.



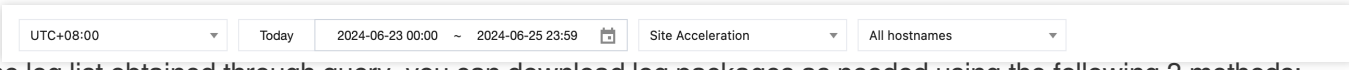
Example: Querying Offline Logs for a Specified Domain Name within a Specified Time Period

Sample Scenario

After you add an acceleration domain name and add `www.example.com` to the EdgeOne service, you shall download all site acceleration logs for `www.example.com` from June 23, 2024 to June 25, 2024 to perform data analysis. You can refer to the following directions.

Directions

1. Log in to the EdgeOne console and click **Site List** in the left sidebar. Then click the **site** of your concern in the site list, to enter the site details page.
2. On the site details page, click **Log Service** > **Offline Logs**.
3. On the offline logs page, select the dates from June 23, 2024 to June 25, 2024 to filter by time range. On the right side, select Site Acceleration Logs as the log type and www.example.com as the domain name. The page will automatically display the query results of logs meeting the conditions after filtering.



4. In the log list obtained through query, you can download log packages as needed using the following 3 methods:
Click **Download** in the **Operation** column, to download the log package for the corresponding domain name/L4 proxy instance and the corresponding time period.
Click **Get Download URLs**, to copy the download link for the corresponding log package.
Select the required log packages, and click **Batch Get Download URLs** to copy the download links for all required log packages in batch.

Related APIs

DownloadL7Logs
DownloadL4Logs

Related References
Field description
L7 Access Logs

Last updated：2024-11-27 11:18:05

The following are detailed field descriptions for L7 Access Logs (Site Acceleration Log, Rate Limiting, CC Attack Protection Log, Custom Rule Log, Bot Management Log, Managed Rule Log).

Note

Real-time Log - Site Acceleration Log records Full L7 Request Log, including the feature of L7 Protection Blocked Log is in beta testing. If needed, please Contact Us.

Rate Limiting, CC Attack Protection Log, Custom Rule Log, and Bot Management Log are projected to be discontinued on July 31, 2024. It is recommended to use the Site Acceleration Log to obtain comprehensive L7 Protection Logs.

Field Description

General Fields

Field Name	Data Type	Description	Does this field support offline logs	Does this field support real-time logs
EdgeEndTime	Timestamp ISO8601	The time to complete the response to the client request. Example value: 2024-10-14T05:13:43Z, denoting 05:13:43, October 14, 2024 (UTC+0), which is equivalent to 13:13:43, October 14, 2024 (UTC+8 (Beijing time)).	×	✓
EdgeFunctionSubrequest	Integer	Indicates whether this log entry belongs to a subrequest initiated by an edge function, with the following values: 1: Subrequest initiated by an edge function.	✓	✓

		0: Subrequest not initiated by an edge function.		
EdgeServerID	String	Unique identifier of the EdgeOne server accessed by the client.	✓	✓
EdgeServerIP	String	IP address of the EdgeOne server obtained through DNS resolution of the Host.	✓	✓
EdgeSeverRegion	String	Country/Region resolved from the IP address of the responding EdgeOne node, in the format as per: ISO 3166-1 alpha-2.	×	✓
LogTime	Timestamp ISO8601	Time the log was generated.Example value: 2024-10-14T05:13:43Z.	×	✓
ParentRequestID	String	If this request is initiated using edge functions, record the parent request's RequestID; otherwise, record as "-".	✓	✓
RequestID	String	Unique ID of the client request.	✓	✓

Client information

Field Name	Data Type	Description	Does this field support offline logs	Does this field support real-time logs
ClientDeviceType	String	Client request device type, values are: TV: Television Tablet: Tablet PC Mobile: Mobile Phone Desktop: Computer Other: Other	×	✓
ClientIP	String	Client IP connecting to EdgeOne nodes.	✓	✓
ClientISP	String	ISP information resolved from	✓	✓

		Client IP. For data within the Chinese mainland, record as the ISP's Chinese name; For data in global availability zones (excluding the Chinese mainland), record as Autonomous System Number (ASN).		
ClientRegion	String	Country/Region resolved from the Client IP. Format standard: ISO 3166-1 alpha-2.	✓	✓
ClientState	String	Subdivision below the country level resolved from the Client IP. Currently supports only data within the Chinese mainland. Format standard: ISO-3166-2.	✓	✓

Request information

Field Name	Data Type	Description	Does this field support offline logs	Does this field support real-time logs
RemotePort	Integer	The EdgeOne node port that establishes a connection with the client under the TCP protocol.	✓	✓
RequestBytes	Integer	Total traffic sent from the client to the EdgeOne node during the request process, based on the size of the request header, request body, and data sent during the SSL handshake. Unit: Byte.	✓	✓
RequestHost	String	Client request host.	✓	✓
RequestMethod	String	HTTP client request method, values are: GET POST HEAD PUT DELETE	✓	✓

		CONNECT OPTIONS TRACE PATCH		
RequestProtocol	String	Client request application layer protocol, values are: HTTP/1.0 HTTP/1.1 HTTP/2.0 HTTP/3 WebSocket	✓	✓
RequestRange	String	Client request Range.	✓	✓
RequestReferer	String	Client request Referer.	✓	✓
RequestSSLProtocol	String	Client SSL(TLS) protocol used. If the value is "-", it means there was no SSL handshake. Possible values are: TLS1.0 TLS1.1 TLS1.2 TLS1.3	×	✓
RequestStatus	String	Client request status. If using the WebSocket protocol, EdgeOne will periodically log it. This field can be used to determine the connection status. Possible values are: 0: not ended 1: Request successfully terminated 2: Under WebSocket protocol, indicates the first log entry of the connection 3: Under WebSocket protocol, indicates a log entry that is neither the first nor the last of the connection	✓	✓
RequestTime	Timestamp ISO8601	Time when the EdgeOne node received the client request, timezone: UTC +00:00.Example value: 2024-10-14T05:13:43Z.	✓	✓
RequestUA	String	Client request User-Agent.	✓	✓

RequestUrl	String	Client request URL Path, excluding query parameters.	✓	✓
RequestUrlQueryString	String	A query string that is carried in the client request URL.	✓	✓

Response information

Field Name	Data Type	Description	Does this field support offline logs	Does this field support real-time logs
EdgeCacheStatus	String	Whether the client request hits the node cache, values include: hit: resource provided by node cache miss: resource can be cached, but provided by origin server dynamic: resource cannot be cached other: unrecognized cache status	✓	✓
EdgeInternalTime	Integer	Time consumption from when EdgeOne receives the client-initiated request to when the first byte is responded to the client; unit: ms.	✓	✓
EdgeResponseBodyBytes	Integer	Response body size returned to the client by the nodes, unit: Byte.	✓	✓
EdgeResponseBytes	Integer	Total traffic returned by the node to the client, based on the size of the response header, response body, and data sent by the EdgeOne node during the SSL handshake. Unit: Byte.	✓	✓
EdgeResponseStatusCode	Integer	Response status code returned to the client by the nodes.	✓	✓
EdgeResponseTime	Integer	Time consumed from when EdgeOne receives the client-	✓	✓

		initiated request to when the client receives the server-side response. Unit: ms.		

Real Server Information

Field Name	Data Type	Description	Does this field support offline logs	Does this field support real-time logs
OriginDNSResponseDuration	Float	Time consumed to receive the DNS Resolution response from the origin server. If there is no origin retrieval, it is recorded as -1. Unit: ms.	×	✓
OriginIP	String	The IP of the origin server accessed for origin retrieval. If there is no origin retrieval, it is recorded as "-".	×	✓
OriginRequestHeaderSendDuration	Float	Time consumed to send the request header to the origin server. It is generally 0. If there is no origin retrieval, it is recorded as -1. Unit: ms.	×	✓
OriginResponseHeaderDuration	Float	Time consumed from sending the request header to the origin server to receiving the response header from the origin server. If there is no origin retrieval, it is recorded as -1. Unit: ms.	×	✓
OriginResponseStatusCode	Integer	origin server Response Status Code, if there is no origin retrieval, record as -1.	×	✓
OriginSSLProtocol	String	SSL protocol version used for the request to the origin server. If there is no origin	×	✓

		retrieval, it is recorded as "-"; possible values: TLS1.0 TLS1.1 TLS1.2 TLS1.3		
OriginTCPHandshakeDuration	Float	Time consumed to complete the TCP handshake when requesting the origin server. If there is no origin retrieval, it is recorded as -1. Unit: ms;**Note: It is 0 when the connection is reused.**	×	✓
OriginTLSHandshakeDuration	Float	Time consumed to complete the TLS handshake when requesting the origin server. If there is no origin retrieval or the origin-pull protocol is HTTP, it is recorded as -1. Unit: ms; **Note: It is 0 when the connection is reused.**	×	✓

Security Protection related fields

Field Name	Data Type	Description	Does this field support offline logs	Does this field support real-time logs
BotCharacteristic	String	EO Bot Intelligent Analysis Engine has identified the characteristics of this request, only available for domains with the Bot Intelligent Analysis feature enabled in Bot Management.	×	✓
BotClassAccountTakeOver	String	Based on recent IP Intelligence Data, the Client IP request poses a risk level for malicious login attacks. The values are: high: High Risk medium: Medium Risk low: Low Risk	×	✓

		-: No historical data or domain has not enabled the Client Reputation feature		
BotClassAttacker	String	Based on recent IP Intelligence Data, the Client IP request poses a risk level for attacks (e.g., DDoS, high-frequency malicious requests, site attacks). The values are: high: High Risk medium: Medium Risk low: Low Risk -: No historical data or domain has not enabled the Client Reputation feature	×	✓
BotClassMaliciousBot	String	Based on recent IP Intelligence Data, the Client IP request poses a risk level for malicious crawlers, volume brushing, and brute force attacks. The values are: high: High Risk medium: Medium Risk low: Low Risk -: No historical data or domain has not enabled the Client Reputation feature	×	✓
BotClassProxy	String	Based on recent IP Intelligence Data, the Client IP request opens a suspicious proxy port and is used as a Network Proxy (including Second-level IP Dialing). The risk levels are: high: High Risk medium: Medium Risk low: Low Risk -: No historical data or domain has not enabled the Client Reputation feature	×	✓
BotClassScanner	String	Based on recent IP Intelligence Data, the Client IP request shows Scanner Behavior of exploiting known vulnerabilities. The risk levels are: high: High Risk medium: Medium Risk low: Low Risk -: No historical data or domain has not enabled the Client Reputation feature	×	✓
BotTag	String	The EO Bot Intelligent Analysis Engine	×	✓

		comprehensively evaluates requests based on factors such as request rate and the IP Intelligence Database, only available for domains with the Bot Intelligent Analysis feature enabled in Bot Management. The values are: evil_bot:Malicious Bot Requests suspect_bot:Suspected Bot Requests good_bot:Normal Bot Request normal:Normal Request -:Unclassified		
JA3Hash	String	Used to analyze the JA3 fingerprint's MD5 hash value for SSL/TLS clients. Provided only for domains with Bot Management enabled.	×	✓
SecurityAction	String	Final disposition action after request hits security rules, with possible values: -:Unknown/Not Hit Monitor:Observation JSChallenge:JavaScript Challenge Deny:Block Allow:Allow BlockIP:IP Ban Redirect:Redirect ReturnCustomPage:Return to Custom Page ManagedChallenge:Hosted Challenge Silence:Silence LongDelay:Response after a long delay ShortDelay:Response after a short delay	×	✓
SecurityModule	String	The name of the security module that finally handles the request, corresponding to `SecurityAction`, possible values include: -:Unknown/Not Hit CustomRule: Web Protection - Custom Rules RateLimitingCustomRule: Web Protection - Rate Limiting Rules ManagedRule: Web Protection - Managed Rules L7DDoS: Web Protection - CC Attack Protection	×	✓

		BotManagement: Bot Management - Basic Bot Management BotClientReputation: Bot Management - Client Profile Analytics BotBehaviorAnalysis: Bot Management - Intelligent Bot Analysis BotCustomRule: Bot Management - Custom Bot Rules BotActiveDetection: Bot Management - Proactive Feature Recognition		
SecurityRuleID	String	ID of the security rule for final request handling, corresponding to SecurityAction.	×	✓

Log Example

Below is an example of a single L7 access log by default. You can customize the EdgeOne log output format according to the specific requirements of the downstream log analysis system. For more details, see Custom Log Output Format.


```
{
    "ClientState": "CN-LN",
    "BotTag": "normal",
    "EdgeSeverRegion": "US",
    "RequestID": "13719873400522703510",
    "RequestMethod": "GET",
    "RequestUrlQueryString": "-",
    "LogTime": "2024-10-13T23:30:39Z",
    "RequestUrl": "/app/",
    "RequestBodyBytes": 0,
    "SecurityRuleID": "-",
    "OriginRequestHeaderSendDuration": 0.001,
    "EdgeResponseTime": 379,
    "ParentRequestID": "-",
    "RequestSSLProtocol": "-",
    "RequestTime": "2024-10-13T23:30:39Z",
    "EdgeResponseStatusCode": 404,
    "ClientIP": "0.0.0.0",
    "BotCharacteristic": "-",
    "SecurityAction": "-",
    "EdgeEndTime": "2024-10-13T23:30:39Z",
    "RequestRange": "-",
    "BotClassScanner": "-",
```

```
    "BotClassProxy": "-",
    "ClientDeviceType": "Desktop",
    "RequestHost": "chatgpt.skyrun.vip",
    "OriginSSLProtocol": "-",
    "EdgeResponseBodyBytes": 548,
    "RequestProtocol": "HTTP/1.1",
    "EdgeServerID": "b3da9837137ad37f8e430b1d6de51dc5-d41d8cd98f00b204e9800998ecf84
    "EdgeCacheStatus": "miss",
    "EdgeFunctionSubrequest": 0,
    "EdgeResponseBytes": 825,
    "OriginTCPHandshakeDuration": 182.485,
    "SecurityModule": "-",
    "EdgeInternalTime": 378,
    "RequestBytes": 769,
    "OriginIP": "0.0.0.0",
    "JA3Hash": "-",
    "OriginResponseHeaderDuration": 182.676,
    "OriginResponseStatusCode": 404,
    "ClientRegion": "US",
    "RemotePort": 80,
    "ClientISP": "AS396982",
    "BotClassMaliciousBot": "-",
    "BotClassAccountTakeOver": "-",
    "OriginDNSResponseDuration": 0.0,
    "RequestReferer": "-",
    "BotClassAttacker": "-",
    "RequestUA": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.3
    "EdgeServerIP": "0.0.0.0",
    "OriginTLSHandshakeDuration": -1,
    "RequestStatus": "1"
}
```

L4 Proxy Logs

Last updated：2024-07-15 09:31:09

The following are detailed descriptions for the fields in L4 proxy logs.

Note:

In a long TCP connection scenario, EdgeOne records logs periodically and the last log entry is recorded when the connection ends. You can judge the connection status by checking whether the `DisconnetReason` field is empty. Additionally, you can use the `SessionID` to identify the connection. The logs with the same `SessionID` record the actions of the same connection.

Under the L4 Proxy Logs type, real-time logs and offline logs record the same fields.

Field Name	Data Type	Description
ClientRealIP	String	Real IP address of the client.
ClientRegion	String	2-letter country/region code of the client, compliant with ISO-3166 alpha-2 standard.
ConnectTimeStamp	Timestamp ISO8601	Connection time, UTC +0 time zone by default.
DisconnetReason	String	Disconnection reason. If not disconnected during the current log cycle, the value is -. The format is **Direction: Reason**. Valid values for the direction include: up: origin server direction down: client direction Valid values for the reason include: net_exception_peer_error: Read/Write peer error. net_exception_peer_close: The peer has closed connection. create_peer_channel_exception: Failed to create a channel to the next hop. channel_eof_exception: Channel has ended (when the request ends, the node ending the request will send channel_eof to the adjacent node, informing it that the request has ended). net_exception_closed: Connection has closed. net_exception_timeout: Read/Write timed out.
DisconnetTimeStamp	Timestamp ISO8601	Disconnection time, UTC +0 time zone by default. If not disconnected during the current log cycle, the value is -.
EdgeIP	String	IP address of the accessed EdgeOne server.
ForwardPort	Integer	Customer-configured forwarding port.

ForwardProtocol	String	Customer-configured forwarding protocol TCP/UDP.
LogTimeStamp	Timestamp ISO8601	Log generation time, UTC +0 timezone by default.
ReceivedBytes	Integer	Outbound traffic generated from the recording time of the previous log entry to the recording time of this log entry, in bytes.
SentBytes	Integer	Inbound traffic generated from the recording time of the previous log entry to the recording time of this log entry, in bytes.
ServiceID	String	Unique identifier ID of the L4 proxy service.
SessionID	String	Unique identifier ID of the TCP connection or UDP session.

Edge Function Running Logs

Last updated：2025-04-22 15:26:02

Field Description

Field Name	Data Type	Description
EdgeFunctionName	String	Name of the triggered edge function.
RequestHost	String	Host of the client request (the domain name of the main request).
RequestID	String	Unique ID of the client request (UUID of the main request).
EventTimestamp	Integer	Timestamp of the function trigger. Format: UNIX, accurate to ms.
Logs	Array[object]	Specified information printed by Users in the JS code using the console.log() method.
Outcome	String	Whether the function execution is successful, values are: ok: no exceptions or errors exception: There are exceptions or errors. For error details, please refer to the Exceptions field.
Exceptions	Array[object]	Details of exceptions and errors encountered during function execution.
CpuTime	Integer	CPU time consumed, unit: microsecond.
WallTime	Integer	Wall clock time, the time elapsed from function trigger to completion, unit: microsecond.

Log Examples

Below is an example of a single edge function runtime log by default. You can customize the EdgeOne log output format according to the specific requirements of the downstream log analysis system. For more details, see Customizing Log Output Format.

```
{
    "WallTime": 125221,
    "RequestID": 8254903099116445190,
    "EventTimestamp": 1727251229247,
```

```
    "EdgeFunctionName": "testtracelog-zone-2rdnikn6zjck-251255070",
    "Outcome": "ok",
    "RequestHost": "www.example.com",
    "Exceptions": [],
    "Logs": [
        {
            "level": "INFO",
            "message": "test log from function",
            "timestamp": 1727251229247
        }
    ],
    "CpuTime": 547
}
```

Real-Time Log Push Filter Conditions

Last updated：2024-07-15 09:31:09

Real-time Log Push supports configuring the filter conditions to help you filter out specific types of logs and reduce the volume of downstream log processing. The following are the supported log fields and comparison operators.

Note

Currently, only Real-time Logs - **Site Acceleration Logs** support configuring the log push filter conditions.

The Real-time Log Push Filter Conditions feature is in beta testing. If needed, please contact us.

Supported Log Fields

Field Name	Data Type	Description
SecurityAction	String	Final handling action after a request matches the security rules. Valid values include: -: unknown/not matched Monitor: observation JSChallenge: JavaScript challenge Deny: block Allow: pass BlockIP: IP banning Redirect: redirect ReturnCustomPage: returning custom pages ManagedChallenge: managed challenge Silence: Silence LongDelay: response after a long delay ShortDelay: response after a short delay
SecurityModule	String	Name of the security module finally handling the request, corresponding to `SecurityAction`. Valid values include: -: unknown/not matched CustomRule: Web Protection - Custom Rules RateLimitingCustomRule: Web Protection - Rate Limiting Rules ManagedRule: Web Protection - Managed Rules L7DDoS: Web Protection - CC Attack Protection BotManagement: Bot Management - Bot Basic Management BotClientReputation: Bot Management - Client Reputation BotBehaviorAnalysis: Bot Management - Bot Intelligent Analysis BotCustomRule: Bot Management - Custom Bot Rules

		BotActiveDetection: Bot Management - Proactive Feature Recognition
EdgeResponseStatusCode	Integer	Response status code returned to the client by the node.
OriginResponseStatusCode	Integer	Response status code of the origin server. If there is no origin-pull, it is recorded as -1.

Supported Comparison Operators

Comparison Operator Name	Supporting the Data Type or Not	
	String	Integer
Equals (matching any value in the list)	✓	✓
Greater than	×	✓
Less than	×	✓
Greater than or equal to	×	✓
Less than or equal to	×	✓

Example: Filtering out Logs with HTTP Status Codes of 4xx/5xx

Sample Scenario

In a large e-commerce platform's IT Ops team, you are responsible for monitoring and analyzing real-time logs of the website. Due to the high volume of site visits and the enormous amount of log data, you wish to reduce unnecessary log data push by setting up filtering rules, thus avoiding unnecessary burden on the analysis platform. For instance, you can perform configuration to push only the access logs with HTTP status codes of 4xx/5xx, which usually indicate some kind of error. In this way, you can focus on logs that may point to user experience issues or system failures requiring immediate attention. You can follow the directions below for configuration.

Directions

1. Log in to the EdgeOne console and click **Site List** in the left sidebar. Then click on the **site** to be configured in the site list, to enter the site details page.
2. On the site details page, click **Log Service** > **Real-time Logs**.
3. On the real-time logs page, click **Create Push Task**.

4. On the log source selection page, enter a task name, select a log type, service area, and domain name/L4 proxy instance requiring log push, and click **Next**.

5. On the push content definition page, configure the log push range.

5.1 Select **Filtered logs**.

5.2 Enter the filtering conditions, as shown in the figure below:

6. After configuring the destination, click **Push**, confirm the related cost tips in the pop-up window, and click **Confirm Creation** to save the configuration.

Custom Log Push Fields

Last updated：2025-01-24 14:38:41

If you need to push HTTP request headers, HTTP response headers, Cookie values, or certain field values in the HTTP request body, you can accurately record these information in the logs using the custom log field feature.

Use Restrictions

Currently, only **real-time logs - site acceleration logs** support adding custom fields;
In the same real-time log push task, custom field names must not be duplicated;
A maximum of 200 Custom Definition fields can be configured;
Field names are case-sensitive and must exactly match the original field names in HTTP actions;
When the field type is selected as request header, response header, or Cookie, the field name can be 1 to 100 characters long, starting with a letter, consisting of letters, digits, and hyphens (-), and ending with a letter or digit;
When the field type is selected as request body, you can use Google RE2 regular expressions to extract the specified content;
For a single request body type field, the maximum length of the regular expression is 4 KB;
For a single request body type field, the maximum length of the extracted content is 1,000 bytes. Contents exceeding this limit will be truncated and discarded;
Up to 5 request body type custom fields can be added for a single real-time log push task.

Example 1: Logging the value of a specified response header

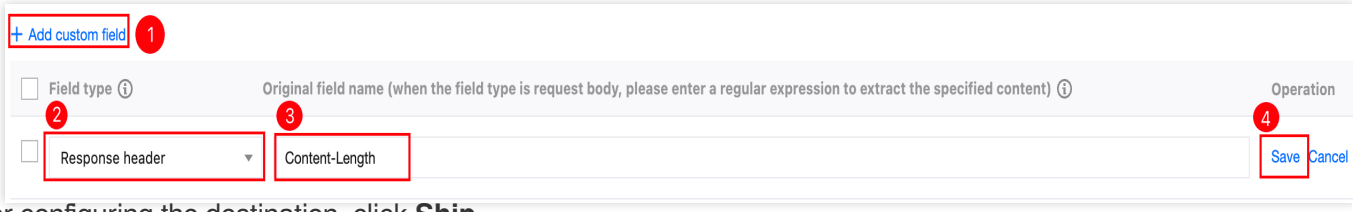
Sample Scenario

In some business scenarios, understanding the size of the response body is crucial for monitoring network traffic and optimizing performance. To achieve this, custom definition log fields can be configured to record the value of the `Content-Length` header for each response.

Operation step

1. Access the EdgeOne console, locate the left-hand menu, and select the **Site List**. Within this listing, click on the **Site** requiring configuration.
2. On the Site Details Page, click **Log Service** > **Real-time Logs.**
3. On the Real-time Logs page, click **Create shipping task.**
4. On Select the log source Page, fill in the Task name, select the Log type Service area, the Domain name/Layer 4 Proxy Instance for which the logs need to be pushed, and click **Next**.

5. On the Define delivery content page, click **Add custom field**.
5.1 Select the field type as Response Header;
5.2 Enter the field name as `Content-Length` ;
5.3 click **Save**.



6. After configuring the destination, click **Ship**.

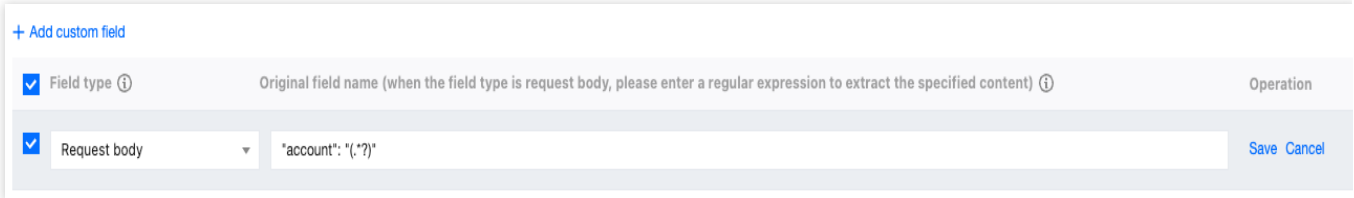
Example 2: Logging Specified Content from the Request Body

Sample Scenario

Assume that you need to extract the `account` field from a POST request's request body JSON object for analyzing the access behaviors of different user accounts.

Operation Step

1. Log in to the EdgeOne console, and on the left sidebar, click **Site List** . In the site list, click the **site** to be configured.
2. On the Site Details page, click **Log Service** > **Real-time Logs**.
3. On the Real-time Logs page, click ++**Create shipping task**++.
4. On the Select the log source page, fill in the task name, select the log type, service area, and the domain name for which the logs to be pushed, and then click **Next** .
5. On the Define Push Content page, click **Add custom field** .
5.1 Select the field type as Request body.
5.2 Fill in the field name as `"account": "(.*?)"` .
5.3 Click **Save** .



6. After configuring the destination, click **Push**.
7. Send a test request to check whether the log recording behavior meets expectations. The test command is as follows:

```
  curl -X POST https://www.example.com -H "Content-Type: application/json" -d
'{"account": "user123", "password": "pass456"}'
```

8. The received logs will include the `RequestBodyCustom1` field, presented as a key-value pair in the example. If you add multiple request body type custom fields in a single real-time log push task, the received logs will sequentially include `RequestBodyCustom1` , `RequestBodyCustom2` , and so on.

```
{"RequestBodyCustom1":"\\"account\\": \\"user123\\""}
```

References

If you would like to understand the meanings of various HTTP request and response headers to decide whether to log them, please refer to HTTP Standard Header Explanation.

Customizing Log Output Formats

Last updated：2024-09-24 18:10:15

Overview

Users can customize log output formats, including selecting different logging styles (such as JSON Lines or CSV) and defining the prefix and suffix of batches and single logs, as well as the delimiter between log records or fields. The default format for real-time and offline logs is JSON Lines.

Note:

Currently, only real-time logs - **site acceleration logs** support configuring the log output format.

Configuration Item

Log output format: Preset output format type for log shipping.

JSON Lines: Fields in a single log are presented as key-value pairs.

CSV: Only field values are presented in a single log, without displaying field names.

Batch prefix & suffix: Users can define prefixes and suffixes for log batches. A batch refers to a single log push request. Each log shipping batch may contain multiple log records.

Prefix: A string added before each log shipping batch.

Suffix: A string appended after each log shipping batch.

Single log record prefix & suffix:

Prefix: A string added before each log record.

Suffix: A string appended after each log record.

Note:

When log sampling or filtering is not enabled, for domain name business, one HTTP request corresponds to one log record.

Log record delimiter: A string inserted between log records as a delimiter.

Field delimiter: A string inserted between fields within a single log record as a delimiter.

Sample Configuration

The following are log samples corresponding to different log output formats.

JSON Lines

Configuration Example

Configuration Item	Value
Log output format	JSON Lines
Single log record prefix	{
Single log record suffix	}
Log record delimiter	\\n
Field delimiter	,

Log Output Sample

```
{"SecurityAction":"Deny","RequestID":"1494104494197154881","RequestTime":"2024-08-
{"SecurityAction":"Deny","RequestID":"1494104594197154882","RequestTime":"2024-08-
```

CSV

Configuration Example

Configuration Item	Value
Log output format	csv
Log record delimiter	\\n
Field delimiter	,

Log Output Sample

```
Deny,1494104494197154881,2024-08-12T08:12:15Z,1.1.1.1
Deny,1494104594197154882,2024-08-12T08:12:30Z,2.2.2.2
```

JSON Lines Variants

JSON Array Format Configuration Example

Configuration Item	Value
Log output format	JSON Lines
Batch prefix	[

Batch suffix	]
Single log record prefix	{
Single log record suffix	}
Log record delimiter	,
Field delimiter	,

JSON Array Format Log Output Sample

```
[
    {"SecurityAction":"Deny","RequestID":"14941044941971548881","RequestTime":"2024
    {"SecurityAction":"Deny","RequestID":"14941045941971548882","RequestTime":"2024
    {"SecurityAction":"Allow","RequestID":"14941046941971548883","RequestTime":"202
]
```

Embedded JSON Object Format Configuration Example

Configuration Item	Value
Log output format	JSON Lines
Batch prefix	{"events":[
Batch suffix	]}
Single log record prefix	{"info":{
Single log record suffix	}}
Log record delimiter	,
Field delimiter	,

Embedded JSON Object Format Log Output Sample

```
{"events": [
        {"info": {"SecurityAction":"Deny","RequestID":"14941044941971548881","Reque
        {"info": {"SecurityAction":"Deny","RequestID":"14941045941971548882","Reque
]}
```

CSV Variants

CSV with Header Format Configuration Example

Configuration Item	Value
Log output format	csv
Batch prefix	SecurityAction,RequestID,RequestTime,ClientIP\\n
Log record delimiter	\\n
Field delimiter	,

CSV with Header Format Log Output Sample

```
SecurityAction,RequestID,RequestTime,ClientIP
Deny,14941044941971548881,2024-08-12T08:12:15Z,1.1.1.1
Deny,14941045941971548882,2024-08-12T08:12:30Z,2.2.2.2
Allow,14941046941971548883,2024-08-12T08:12:45Z,3.3.3.3
```

TSV Format Configuration Example

Configuration Item	Value
Log output format	csv
Log record delimiter	\\n
Field delimiter	\\t

TSV Format Log Output Sample

```
Deny  14941044941971548881  2024-08-12T08:12:15Z  1.1.1.1
Deny  14941045941971548882  2024-08-12T08:12:30Z  2.2.2.2
Allow  14941046941971548883  2024-08-12T08:12:45Z  3.3.3.3
```

Data Analysis Overview

Last updated：2024-08-26 09:58:16

Tencent Cloud EdgeOne security acceleration platform analyzes access log data and provides various data metrics in the data analysis page for you to understand your business data from multiple dimensions.

Applicable Scenarios

Scenario	Specific Demand
Daily monitoring and inspection	By observing the trends and distribution of various data metrics of acceleration domain names/L4 proxy instances, continuously monitor whether EdgeOne has high latency or failures.
Troubleshooting analysis	By analyzing access logs, understand the path and content of the user's access to locate and troubleshoot issues.
Business data insight	By analyzing and mining client data, understand user profiles.

Function Details

Data analysis function	Function introduction
Metric analysis	Provides aggregated metric data on the traffic, bandwidth, number of requests, etc., and supports viewing the access region distribution, cache hit ratio, security protection status, and status codes, to help you understand the operational status of various businesses that have accessed EdgeOne.
Web security analysis	Analyzes the access logs that match the Web security rules, to obtain the attack surface data related to your business, including attack sources and attack methods. This helps you better understand the attack situation and formulate more effective security policies. You can also directly view the sample logs to understand the attack request context.

Analytics

Last updated：2025-04-22 15:20:16

Overview

Analytics is a powerful data analysis service provided by EdgeOne. It aims to help you get an in-depth understanding of the business operation and security status. Through real-time monitoring and analysis on key metrics, you can quickly identify problems, optimize configurations, and enhance the stability and security of your service.

Supported Metrics

Analytics allows you to customize the display of data metrics, including the following operations:
1. **Display and sort metrics**: You can select certain metrics in **metric settings** for display, and sort the metrics by dragging and dropping to determine the display order of the metrics on the dashboard.
2. **Select time ranges** : You can modify the query time range to view data within last 30 minutes, last hour, today, and so on. The time span for a single filter should not exceed 31 days.
3. **Set filters** : You can use filter conditions to view specific data filtered by domain name, status code, country/region, and so on.
Note:
1. Support for historical time ranges and filters may vary depending on your plan version. For details, see Comparison of EdgeOne Plans.
2. Data analyzed by some dimensions (such as client IP, status code, URL path, and Referer) is retained for only 30 days. For example, when you query data of a specified client IP one month ago, the data may not be found.
3. In order to enhance user experience, EdgeOne has incorporated sampling into its data analysis to ensure that queries maintain accuracy and timeliness even when dealing with large volumes of data.
Analytics supports the following metrics:

For Domain Name Service

L7 Access Traffic: The statistics of traffic transmitted between the client and EdgeOne. After clicking **EdgeOne Response Traffic**, you can view the data such as access region distribution, host, client IP, Referer, URL path, resource type, status code, client browser type, client device type, and client operating system ranking.
L7 Access Bandwidth: The statistics of bandwidth transmitted between the client and EdgeOne.
Note:
Under different time granularities, the calculation method for bandwidth metrics varies. This method applies to L7 access bandwidth, L7 origin bandwidth, and L4 access bandwidth metrics.

1-minute granularity: Total traffic within 1 minute * 8 / 60 seconds.
5-minute granularity: Total traffic within 5 minutes * 8 / 300 seconds.
1-hour granularity: The maximum value among all 5-minute granularity bandwidth points.
1-day granularity: The maximum value among all 5-minute granularity bandwidth points.
L7 Access Requests: The statistics of the number of client requests received by EdgeOne.
L7 Security Policy Hits: The statistics of the number of requests matching the EdgeOne Web Security rules. After clicking a certain type of security rules (e.g., custom rules), you can view more detailed data such as matched rule ranking, client IP ranking, URL path ranking, client distribution, and recent events.
L7 Origin Traffic: The statistics of traffic transmitted between EdgeOne and the origin server.
L7 Origin Bandwidth: The statistics of bandwidth transmitted between EdgeOne and the origin server.
L7 Origin Requests: The statistics of the number of requests initiated by EdgeOne to the origin server.
Origin Offload Rate: The percentage of EdgeOne response traffic that was served without a corresponding origin fetch. The calculation formula is: `Origin Offload Rate = 1 - (Origin Response Traffic / EdgeOne Response Traffic)` .
Note:
L7 origin-related metrics and origin offload rate metric display features are currently in beta testing. If you are interested in using these features, please contact us.
L7 Edge Response Time:
Average L7 edge response time: The average duration from when EdgeOne receives a client request to when it responds with the complete file, excluding TCP connection establishment and TLS handshake duration. The statistical formula is: `sum of response time for all requests / total access request count` .
Average L7 edge first byte response time: The average duration from when EdgeOne receives a client request to when it responds with the first byte, excluding TCP connection establishment and TLS handshake duration. The statistical formula is: `sum of first byte response time for all requests / total access request count` .
DNS Queries: The number of resolution requests received by EdgeOne DNS. Only supports the data of zones accessing EdgeOne in NS mode.

Edge Function Metrics

Edge Function Invocation Count:
Total Invocation Count: The number of times an edge function is triggered. An edge function is triggered when a client request meets the triggering rules or when the default domain name of the edge function is requested.
Successful Invocation Count: The number of times an edge function execution is successful.
Failed Invocation Count: The number of times an edge function execution fails.
Edge Function CPU Time: The duration of CPU usage during the execution of an edge function. The statistical presentation includes average values and percentiles (P50, P90, P99).
Edge Function Wall Clock Time: The elapsed time from the start to the completion of an edge function execution. Wall clock time includes the actual execution time of the function as well as any potential waiting time (such as I/O

operations). The statistical presentation includes average values and percentiles (P50, P90, P99).

Note:

1. The percentiles P50, P90, and P99 are calculated as follows:

P50 (Median): Indicates that 50% of the data values are less than or equal to this value.

P90: Indicates that 90% of the data values are less than or equal to this value.

P99: Indicates that 99% of the data values are less than or equal to this value.

2. Please note that when viewing percentile statistical data, you need to specify a specific edge function instance through filtering conditions.

For TCP/UDP Application Service

L4 Access Traffic: The statistics of traffic transmitted between the client and EdgeOne.

L4 Access Bandwidth: The statistics of bandwidth transmitted between the client and EdgeOne.

L4 Concurrent Connections: The number of transport layer connections simultaneously established between the client and EdgeOne. Supports viewing the regional distribution of concurrent connections.

Note:

Under different time granularities, the calculation method for the concurrent connection metric varies.

1-minute granularity: The total number of active connections within 1 minute.

5-minute granularity: The maximum value among all 1-minute granularity concurrent connection points.

1-hour granularity: The maximum value among all 1-minute granularity concurrent connection points.

1-day granularity: The maximum value among all 1-minute granularity concurrent connection points.

For L3/4 DDoS Protection

L3/4 DDoS Protection Bandwidth: The protection bandwidth for network and transport layer DDoS attacks, packet rate, and number of attack events. Supports viewing the protocol ranking of the protection traffic and packet quantity, as well as the historical attacks, attack source distribution, and attack type ranking.

For EdgeOne Shield

EdgeOne Shield Response Traffic: The response traffic of the EdgeOne Shield service.

EdgeOne Shield Responded Requests: The number of requests responded by the EdgeOne Shield service.

Note:

The EdgeOne Shield feature is in beta testing. If you want to use it, contact us.

Web Security Analysis

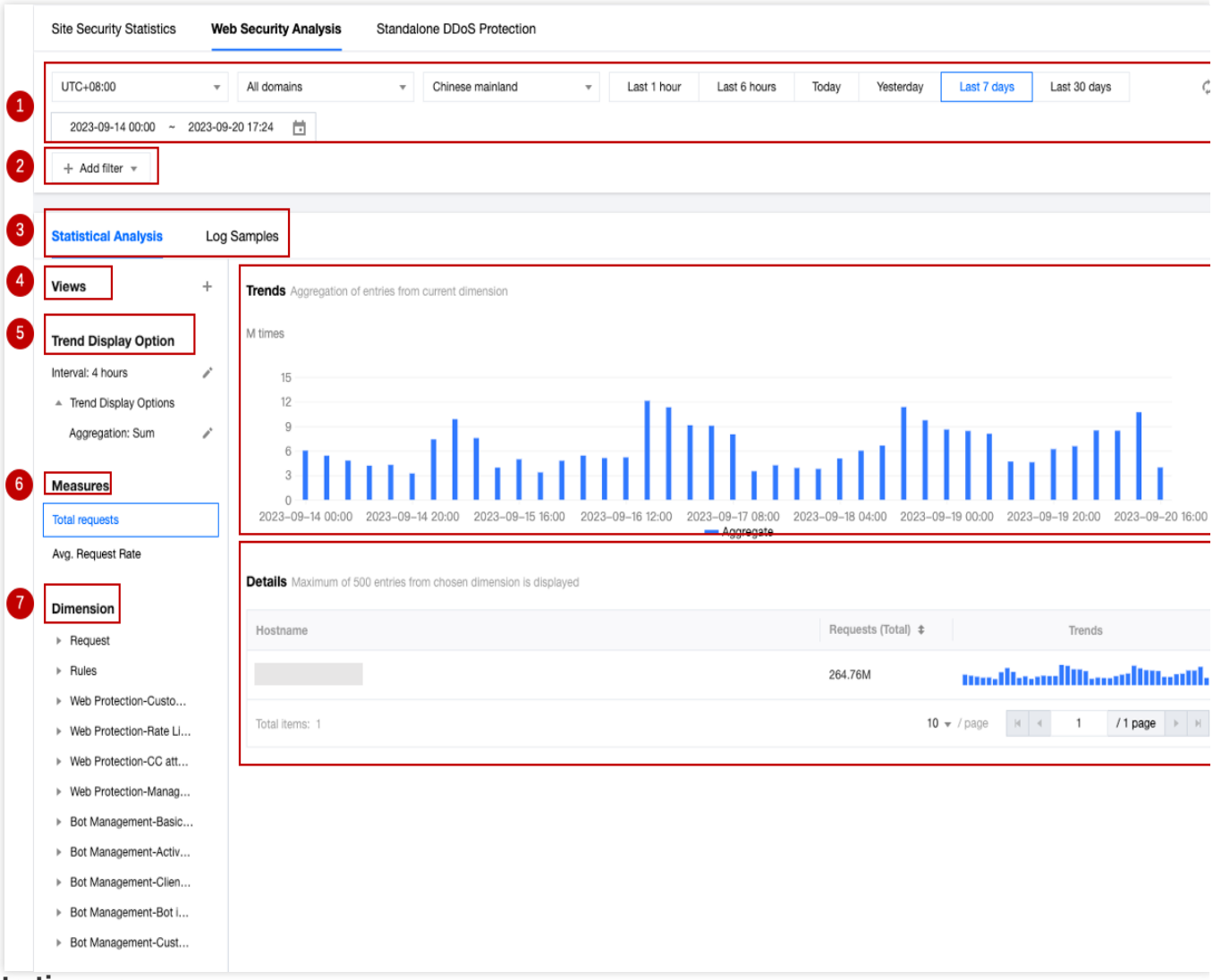
Last updated：2024-12-16 12:08:27

Overview

Web security analysis provides fine-grained analysis tools for security events, offering reference for you to formulate or adjust security policies. You can not only view the statistical analysis and distribution trends of recent security events in dozens of dimensions, but also further understand the specific content and detailed information of an event by viewing sample logs. Web security analysis offers multiple analytical dimensions for request data under EdgeOne web protection, helping you develop efficient security policies.

Supported capabilities

Note:
1. In a security event, a single request may hit multiple security rules. When filtering or selecting statistical dimensions, please distinguish between the rule's disposal method and the request's disposal result. For example: A request hits multiple rules with the disposal method set to observe, and also hits a rule with the disposal method set to intercept, resulting in the final disposal result of the request being intercepted.
2. To optimize user experience, Sampling Statistics technology has been introduced in EdgeOne data analysis to ensure that the accuracy and timeliness of queries can be maintained even when large amounts of data are processed.



1. Data time range

By adjusting the query time range, you can query the security events of a specific time period.

Note:

For the query time range supported by different version plans, please refer to the Comparison of EdgeOne Plans.

2. Add filter

Supports filtering Web security data by request features, rule ID, and other dimensions. For the filter items supported by Web security analysis, please refer to How to use filter conditions.

Note:

1. A single request may hit multiple rules, so when using rule ID filtering, the statistical details and trend distribution of other rules hit simultaneously will be displayed.

2. You can click on the feature value you want to filter in the statistical details to quickly add it to the filter.

3. Analysis dimensions

Statistical analysis: Helps you display the ranking of indicators by the selected dimension, discover abnormal access volume and abnormal access trends. For example: When you choose to display by User-Agent header dimension, you can view the distribution of accessed devices and access indicator trends, thus identifying devices with abnormal access volume and suspicious access behavior with uniform speed cycle.
Sample logs: Help you further view the details of security events and determine whether the security policy hit by the request meets expectations. For example: You can view the managed rules hit by the request and the field content matched by the managed rules through sample logs, which will help you determine whether it is a false intercept and adjust the security policy accordingly.

4. Common views

You can save the current view options as a common view for quick access later according to your needs. You can name the view, which will save the current trend display options, statistical indicators, and statistical dimension information.

5. Trend display statistical method

Note:
When adjusting the data filter time range, the data granularity will be adjusted accordingly to ensure an appropriate trend chart display.
You can adjust the trend chart display options as needed:
Data granularity: The data statistics duration corresponding to each column in the trend chart.
Aggregation method: The calculation method of the data corresponding to each column in the trend chart.
Sum: Displays the sum of all indicators of the statistical items in the selected dimension filtered data within that time period. For example: In the statistical period corresponding to a column in the trend chart, there are 6000 requests, and the column displays data as 6000.
Average value: Displays the average value of all indicators of the statistical items in the selected dimension filtered data within that time period. For example: When displaying statistical data by Host dimension, the data contains 5 Host data, and in the statistical period corresponding to a column in the trend chart, there are 6000 requests, then the column displays data as 6000 / 5 = 1200.
Maximum value: Displays the maximum data item within the time period after the data is split by the selected dimension.
99th percentile value: Displays the minimum value of the data items greater than 99% in the selected dimension split data within that time period, i.e., this value is greater than 99% of the other statistical item indicator values.
99.9th percentile value: Displays the minimum value of the data items greater than 99.9% in the selected dimension split data within that time period, i.e., this value is greater than 99.9% of the other statistical item indicator values.

6. Statistical indicators

You can choose to display the number of requests or the average request rate indicator to display the required statistical features (such as rate features or request number features).

Number of requests: Displays the total number of requests by the current statistical dimension, used to distinguish the characteristics of visitors with a large number of requests. For example: Analyzing by request Host dimension can distinguish the concentrated business domain names.
Average request rate: Calculates the average request rate by the current statistical dimension, used to distinguish the characteristics of visitors with high access frequency. For example: Analyzing by User-Agent header dimension can distinguish the device types with abnormal access frequency.

7. Statistical dimensions

Web security analysis provides the following analysis dimension categories, and you can adjust the statistical objects and grouping methods according to the selected dimensions:
Statistical dimensions classified by request attributes include:
Client IP: Counts the number of requests from different client IPs.
Client IP (XFF header priority): Counts the number of requests from different client IPs. If the client accesses through a Web proxy, the IP of the most recent hop in the XFF header will be counted.
User-Agent: Counts requests from different device types (distinguished by HTTP User-Agent header).
Request URL: Counts requests accessing different URLs (including access paths and query parameters).
Hostname: Counts requests accessing different domains (distinguished by HTTP header Hostname).
Request Referer: Counts requests accessing resources using different referencing methods (distinguished by HTTP Referer header).
Statistical dimensions classified by rule attributes include:
Category: Counts requests hitting different security modules (such as custom rules, managed rules, etc.).
Rule ID: Counts requests hitting different rules.
Note:
1. You can use the rule ID option in the rule classification to merge and display requests hitting all security protection rules.
2. You can also use the rule ID option in the specific security feature classification to view only the situation of hitting rules in that module. For example: Count requests by the rule ID of the Web Protection custom rules hit.
3. Different version plans support different statistical dimensions, please refer to the Comparison of EdgeOne Plans for details.
You can also choose other analysis options provided by the protection features, such as the hit field of managed rules, the bot label of bot intelligent analysis, etc., to perform statistical analysis.

8. Statistical trend chart

The statistical trend chart will display the corresponding aggregated data bar chart according to your trend display options and filter conditions.

9. Statistical details

Displays the request feature values of different dimensions and their corresponding indicators according to your statistical dimension and statistical indicator options. For example: When the number of `requests indicator` and `User-Agent` analysis dimension are selected, the statistical details section will display the number of requests for different client device types (User-Agent header values), displayed in descending order of the number of requests, and the request trends of each device type.

Analysis example

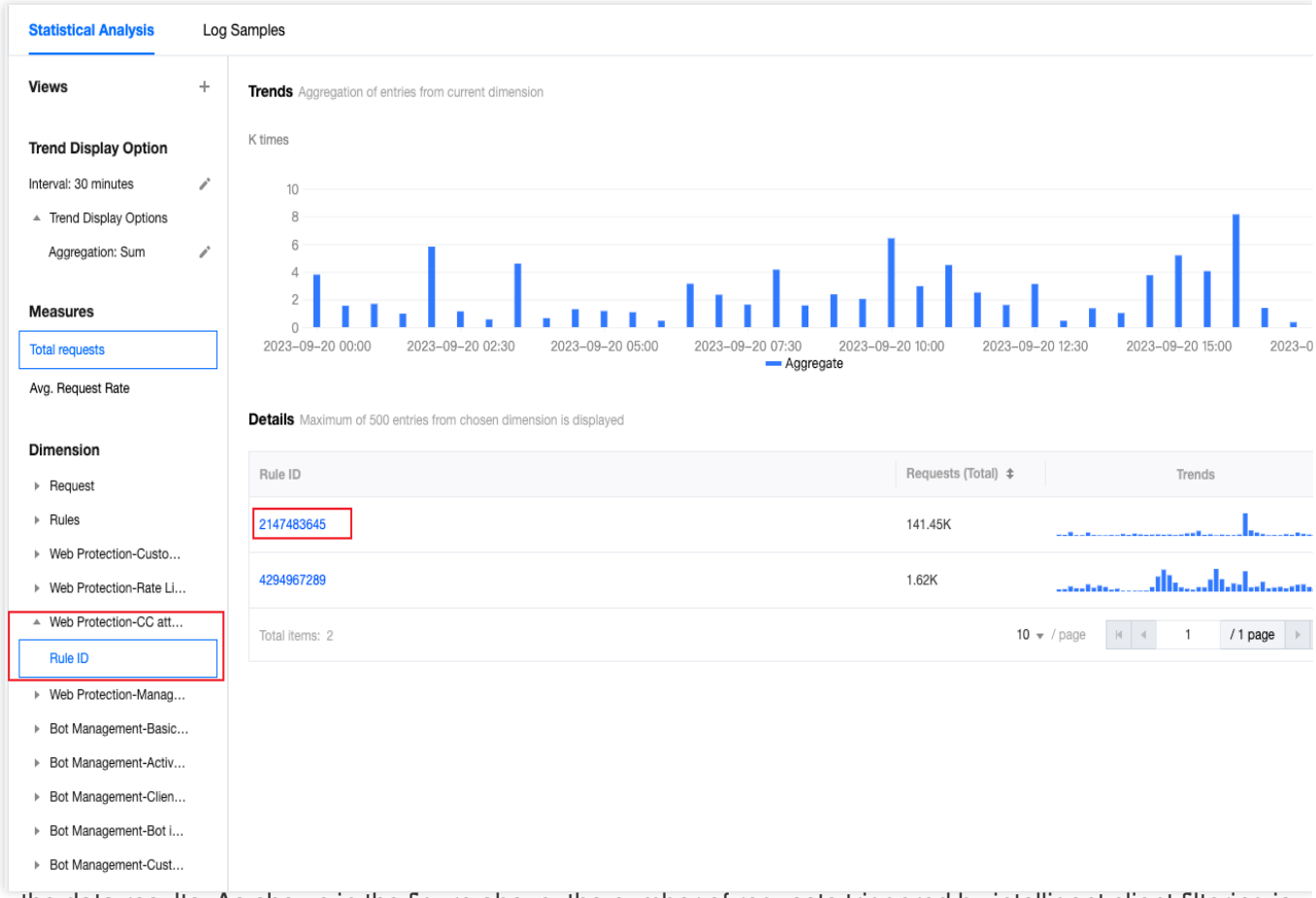
Scenario 1: Analyze the request trend of CC attack defense in the past 1 day

Scenario example

Suppose your site `example.com` finds a suspicious surge in access volume, hitting the CC attack defense rule. To analyze whether all requests hitting CC attack defense in the past 1 day are normal requests, you can follow the steps below for analysis.

Directions

1. Log in to the EdgeOne console, click on the site list in the left menu bar, click on the site to be configured in the site list, and enter the site details page.
2. In the site details page, click on **Data Analysis > Security Protection**, and enter the site security overview analysis page by default. Click on **Web Security Analysis** at the top.
3. Filter and view the domain name, time range, and aggregation conditions of the site to be analyzed. In this scenario, you can select the time range within the past 1 day.
4. In the statistical analysis, click on **Web Protection-CC Attack Defense > Rule ID**.



5. View the data results. As shown in the figure above, the number of requests triggered by intelligent client filtering is very high (Rule ID: 4294967293). You can click on the rule ID to add it to the filter. Then click on Request > User Agent in the left statistical dimensions to view the summary information of all User Agent headers hitting the rule. You can judge whether the User Agent value meets your normal client expectations. You can also continue to add other statistical dimensions in the statistical dimensions, such as Client IP and Request URL, to further narrow down the filter range.

Scenario 2: Analyze whether there are abnormal requests in suspicious bot requests within the last 1 day

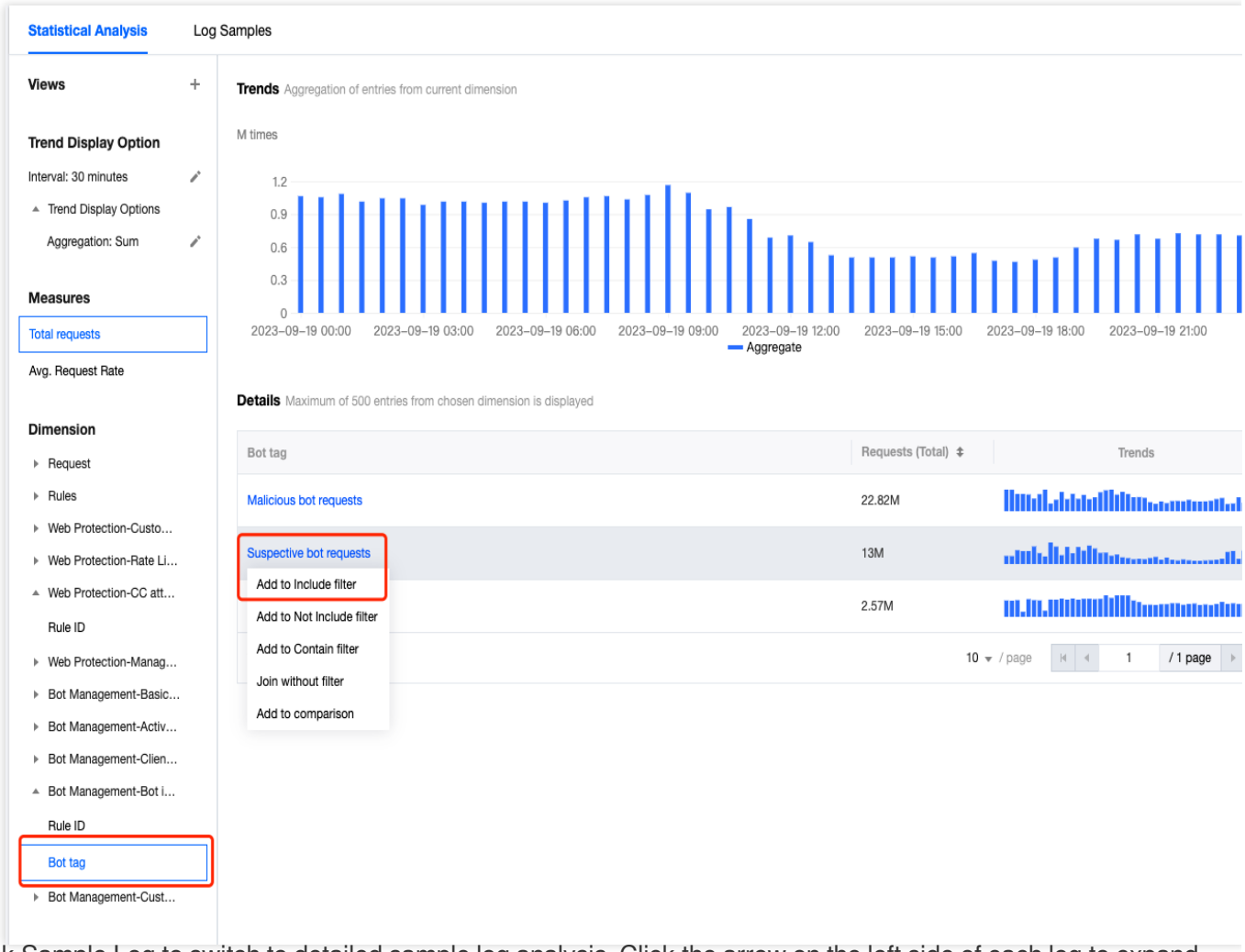
Scenario Example

Suppose your site `example.com` has recently been frequently visited by suspicious bots, and you need to analyze whether all suspicious bot request accesses in the past 1 day are normal requests. You can refer to the following steps for analysis.

Directions

1. Log in to EdgeOne console. In the left sidebar, click **Web Security Analysis**.
2. Filter and view the domain name, time range, and aggregation conditions of the site to be analyzed. In this scenario, you can select the time range within the past 1 day.
3. In the statistical analysis, click **Bot Management-Bot Intelligent Analysis > Bot Tag**.

4. Query the data results, and in the statistical details, you can see the request times of the corresponding bot tags. In this scenario, you can click **Suspective Bot Requests > Add Equal Filter** for further analysis. After adding the filter condition, you can also continue to add other statistical dimensions in the statistical dimension, such as User-Agent to further narrow the filter range.



5. Click Sample Log to switch to detailed sample log analysis. Click the arrow on the left side of each log to expand and view the detailed request header and hit rules situation to determine whether the request is a normal request.

Related References
Sampling Statistics

Last updated：2025-01-15 10:57:06

The EdgeOne data analysis module helps users to analyze traffic characteristics through in-depth analysis of the massive log data continuously recorded by EdgeOne products. In order to optimize the user experience, the sampling statistics technology is introduced in the EdgeOne data analysis to ensure accurate and timely query even when large amounts of data are processed.

What Is Sampling Statistics?

In data analysis, sampling refers to selecting a representative subset from all the data for analysis in order to extract valuable information. For example, in social surveys, researchers cannot survey everyone, so they will select a portion of the population as a representative sample and use the answers of these samples to reflect the tendencies of the entire population.

When Will EdgeOne Apply Sampling Statistics?

EdgeOne employs the dynamic sampling technology to adapt to different users' log data volumes, so as to ensure the accuracy and efficiency of data analysis. In the following data query scenarios, the data displayed on the EdgeOne related pages may be sampled.
When querying L7 access-related metrics on the Metric Analysis page with filters such as status code, ISP, province, TLS version, URL path, Referer, resource type, device type, browser type, system type, IP version, client IP address and User-Agent. This is because when users query the overall traffic, we will provide users with pre-aggregated statistical tables to help users quickly obtain accurate statistical results. However, when users need to perform drill-down analysis on certain specific dimensions, the query will switch to a massive multidimensional statistical table. At this point, a sampling mechanism is needed to reduce the amount of underlying data scanning volume and provide users with a fast query experience.
When querying L7 protection-related metrics on the Metric Analysis page, or conducting **Statistical Analysis** or viewing **Sample Log** on the **Web Security Analysis** page. If a large-scale CC attack occurs within the query time range, the data you see may also be sampled. In this case, there may be circumstances where the log corresponding to a specific request ID cannot be retrieved.
Note:
Note that EdgeOne will continuously optimize and adjust the sampling policy based on the scale of platform log data and users' actual needs. If you have any questions about the data analysis query results provided by EdgeOne, feel

free to contact us at any time.

Does It Affect the Use of EdgeOne?

The sampling statistics technology is only applied to the data analysis module and will not affect other service configurations such as site acceleration, L4 proxy, or security protection. Through the sampling statistics technology, EdgeOne can provide you with statistical analysis results more quickly, helping you obtain query results on the page while improving query efficiency. This ensures that even in the face of massive data, EdgeOne can maintain query response speed and accuracy.

How Do I Query Full Data?

If your business requires in-depth analysis of full log data, we recommend you use EdgeOne's real-time log push feature. The real-time log push feature can transfer detailed and complete log data to your designated log analysis system (such as Tencent Cloud CLS, third-party log solution, or self-built ELK stack). By obtaining full data, you can perform precise data processing. Through the real-time log feature, you can ensure that more accurate data analysis results can be obtained in scenarios requiring higher data precision, thereby providing more accurate data support for your business decisions.

Learning More

Working Principles of Sampling Statistics

Sampling Policy

EdgeOne adopts a dynamic grading policy. This policy will periodically analyze your domain name request volume and the corresponding query performance to determine whether your domain name meets the sampling conditions. When the sampling system determines that your domain name meets the sampling conditions, it will select an appropriate sampling grade for you from the four sampling ratios of 10%, 1%, 0.1%, and 0.01% based on the request volume during the determination period. The trigger rules for each sampling ratio are as follows:
10%: The daily average request volume is more than 10 million times;
1%: The daily average request volume is more than 100 million times;
0.1%: The daily average request volume is more than 1 billion times;
0.01%: The daily average request volume is more than 10 billion times.
After sampling is triggered, your sampling grade is not fixed. If your domain name request volume continues to increase, EdgeOne will accordingly upgrade your sampling grade and use a lower sampling ratio; if your domain name

request volume continues to decline, EdgeOne will accordingly downgrade your sampling grade, use a higher sampling ratio, or even cancel the sampling mechanism for you.

Data Representativeness

EdgeOne will provide a unique identifier (Request ID) for each of your request logs. The sampling system will perform sampling analysis on your data based on this unique identifier to ensure the randomness of the sampling factor. Based on our tests, when the characteristic you need to analyze accounts for a high proportion in the overall data, sampling analysis can provide you with fast and accurate results. However, we also need to point out that when the characteristic you need to analyze accounts for a low proportion in the overall data, due to the small sample size, the results of the sampling analysis may be too large or too small.

For example, you have a dataset with a volume of 10,000, which includes three URL paths: A, B, and C. Their quantity distributions are 7,000 (70%), 2,900 (29%), and 100 (1%) respectively. Ideally, after 10% sampling, the sample sizes for URL paths A, B, and C will be 700, 290, and 10 respectively. Since the sample size for the URL C is too small, the accuracy of estimating the overall based on this sample will be greatly reduced. At this point, the results of your drill-down analysis on URL C may not meet expectations.

How to use filter condition

Last updated：2025-04-22 15:22:54

EdgeOne data analysis supports two types of filtering conditions:
1. Time filtering condition (required): View the data within the selected time range, for details, please refer to How to modify the query time range.
2. Other filtering conditions: Customize the data filtering according to the filtering options supported by each page. The following is a detailed explanation of this part.

Supported Operators

Operator	Description
Equal	Query data with the filter item equal to any specified value
Does not equal	Query data with the filter item not equal to any specified value
Contain	Query data with a field such as URL, Referer, and resource type containing a specified string (for example, query `data with URL containing /example`)
Does not contain	Query data with a field such as URL, Referer, and resource type not containing a specified string (for example, query `data with URL not containing /example`)
Starts with	Query data with a field such as URL, Referer, and resource type starting with a specified string
Does not start with	Query data with a field such as URL, Referer, and resource type not starting with a specified string
Ends with	Query data with a field such as URL, Referer, and resource type ending with a specified string
Does not end with	Query data with a field such as URL, Referer, and resource type not ending with a specified string

Relationship Between Multiple Filtering Conditions

The relationship between multiple filtering conditions is "And", and the relationship between multiple values within the same filtering condition is "Or".

For example, adding filtering conditions `Country/Region=Singapore` ; Thailand and `Status Code=404` means querying data that meets the access from Singapore or Thailand clients and the edge response status code is 404.

Supported Filtering Options

Metric Analysis

Site: EdgeOne site.
Host: The host of the client request.
Country/Region: The country or region where the client request comes from.
Status Codes: The status codes used by EdgeOne for responding to the client.
HTTP Protocol Version: The HTTP version used by the client request. Values include:
HTTP/1.0
HTTP/1.1
HTTP/2.0
HTTP/3.0 (QUIC protocol)
WebSocket Over HTTP/1.1 (WebSocket protocol initiated by HTTP/1.1)
ISP: The ISP where the client request comes from. Supports only the data of sites in the Chinese mainland availability zone.
Province: The province where the client request comes from. Supports only the data of sites in the Chinese mainland availability zone.
TLS Version: The TLS protocol version used by the client request. Values include:
TLS 1.0
TLS 1.1
TLS 1.2
TLS 1.3
URL Path: The URL path (path) of the client request. Supports entering multiple values separated by semicolons, such as `/example1;/example2` .
Referer: The Referer of the client request header. Supports entering multiple values separated by semicolons.
Resource Type: The resource type requested by the client. Supports entering multiple values separated by semicolons, such as `.txt;.jpg` .
Device Type: The device type used by the client request, parsed from the User-Agent in the HTTP request header. Values include:
TV: televisions
Tablet: tablet computer
Mobile: mobile phone
Desktop: computer

Other: others

Empty: empty

Browser Type: The browser type used by the client request. Values include:

Firefox

Chrome

Safari

Opera

QQBrowser

LBBrowser

MaxthonBrowser

SouGouBrowser

BIDUBrowser

TaoBrowser

UBrowser

IE

Microsoft Edge

Bot

Empty

Other

System Type: The operating system type used by the client request. Values include:

Empty

Android

IOS

MacOS

Linux

Windows

ChromiumOS

NetBSD

Bot

Other

IP Version: The IP address version used by the client request. Values include:

IPv4

IPv6

HTTP/HTTPS: The HTTP protocol type used by the client request. Values include:

HTTP

HTTPS

Cache Status: The cache status for the client request. Values include:

hit: The request hits the EdgeOne node cache, and the resource is provided by the node cache. Resources that partially hit the cache are also recorded as hit.
miss: The request does not hit the EdgeOne node cache, and the resource is provided by the origin server.
dynamic: The resource requested cannot be cached or is not configured to be cached by the node. The resource is provided by the origin server.
other: Unrecognizable cache status. Requests responded to by edge functions are recorded as other.
Client IP: View only the request data from the specified Client IP. When the operator is equal to/not equal to, multiple values are supported, separated by Enter.
User-Agent: The User-Agent header value of the client request. Multiple values are supported, separated by Enter.
L4 Proxy Forwarding Rules: The specific forwarding rules for the L4 proxy instance.
L4 Proxy Instance: The name of the L4 proxy instance.
DNS Return Code: The DNS resolution response status code. Values include:
NOError: Successful response with no errors.
NXDomain: Non-existent record.
NotImp: Not implemented. The DNS server does not support the request query type. For implemented request query types, see Record Type.
Refused: Refused. The DNS server refuses to perform the specified operation due to policies.
DNS Record: The DNS record type. For values, see Record Type.
DNS Region: The continent where the client request comes from. Currently supports the following options:
Asia
Europe
Africa
Oceania
America
Applied Action: View only the requests that hit the security rules and apply the specified action (excluding release or exception rules). Values include:
Monitor
Rule ID: View only the request data that hits the specified Web protection rule ID.
Request Path: View only the request data for accessing the specified request path.
Client IP: View only the request data from the specified client IP. Supports entering multiple values separated by carriage returns when the operator is `Equal` or `Does not equal` .
Anti-DDoS Instance: View the data of the specified Anti-DDoS (Enterprise) instance.
EdgeOne Shield Space: View the data of the specified EdgeOne Shield space.
Content Identifier: View the data of the specified content identifier. Only L7 access-related metric data is supported. The content identifier feature is in beta testing. If you need to use it, please Contact Us.
Edge Function Name: View data for a specific edge function instance. Only supports metrics related to edge functions. The feature to display edge function statistics is currently in beta testing.

Edge Function Execution Result: View data regarding the execution results of a specific edge function. Only supports metrics related to edge functions. The feature to display edge function statistics is currently in beta testing.

Note

When the metrics L7 access-related metrics, L7 back-to-source metrics, and cache hit rate are selected, the filtering options including L4 Proxy Forwarding Rules, L4 Proxy Instance, DNS Return Code, DNS Record, DNS Region, Applied Action, Rule ID, Anti-DDoS Instance, and EdgeOne Shield Space are not supported.

When the metric L7 Protection Hit Count is selected, only the filtering options Host, Applied Action, Rule ID, and Client IP are supported.

Web Security Analysis

Supports filtering based on request features, rule features, various detailed Web protection rules, and Bot management policy features. The specific filtering options are described as follows:

Site: EdgeOne site.

Client IP: View only the request data from the specified client IP. Supports entering multiple values separated by carriage returns.

Client IP Region: The client IP comes from the specified country or region.

Client IP (prioritizing XFF header): View only the request data from the specified client IP (prioritizing XFF header). Supports entering multiple values separated by carriage returns.

Client IP Region (prioritizing XFF header): The client IP (prioritizing XFF header) comes from the specified country or region.

User-Agent: The User-Agent header information carried in the client request. Supports entering multiple values separated by carriage returns.

Request URL: The URL in the client request (excluding Host and only including the request path and query parameters). Supports entering multiple values separated by carriage returns.

Hostname: The host of the client request. Supports entering multiple values separated by carriage returns.

Referer: The referer carried in the client request. Supports entering multiple values separated by carriage returns.

Applied Action: The final disposal result of the request by the Web protection module. For details, see Action.The disposition result "Unknown" signifies that no predefined disposal methods have been executed. It serves solely as a fallback classification for data statistical processes and can be disregarded in routine analysis.

Request Path (Path): The URL path of the client request (HTTP request path, excluding Host and query parameters). Supports entering multiple values separated by carriage returns.

Request JA3 Fingerprint: JA3 fingerprint calculated based on the relevant parameters for the TLS handshake request of the client. Only supports the data of the domain names with Bot Management enabled.

Request Method (Method): The HTTP method of the client request.

Request ID: Unique identifier of a request, that is, the `Request ID` of the default block page, `{{ EO_REQ_ID }}` of the custom response page, `EO-LOG-UUID` in the EdgeOne default response header, and `RequestID` in the L7 access logs.

Rule Category: View only the request data that hits the specified category of Web protection rules.

Rule ID: View only the request data that hits the specified Web protection rule ID.

How to Modify Query Time Range

Last updated：2024-08-01 21:38:40

The EdgeOne data analysis page supports users to custom filter the time range. The following mainly introduces two ways to filter the time range.

Note:

In order to improve the query efficiency, the granularity of data in different time ranges is as follows:

Time Range ≤ 2 hours: 1 minute.

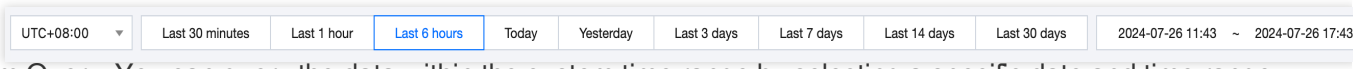
2 hours < Time Range ≤ 48 hours: 5 minutes.

48 hours < Time Range ≤ 7 days: 1 hour.

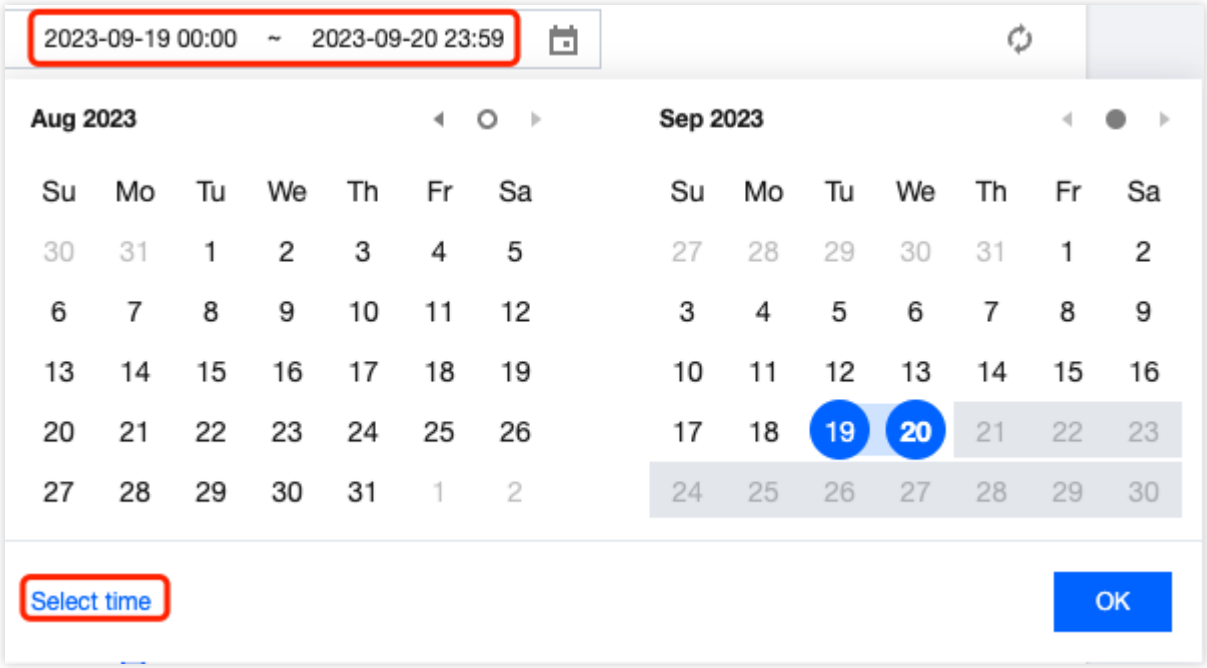
Time Range > 7 days: 1 day.

Method 1: Set the query time range through the filter bar

Quick Query: Quickly query the corresponding time range data by clicking on the buttons such as "Last 30 minutes","Last 1 hours", "Last 6 hours", "Today", "Yesterday","Last 3 days","Last 7 days"



Custom Query: You can query the data within the custom time range by selecting a specific date and time range.

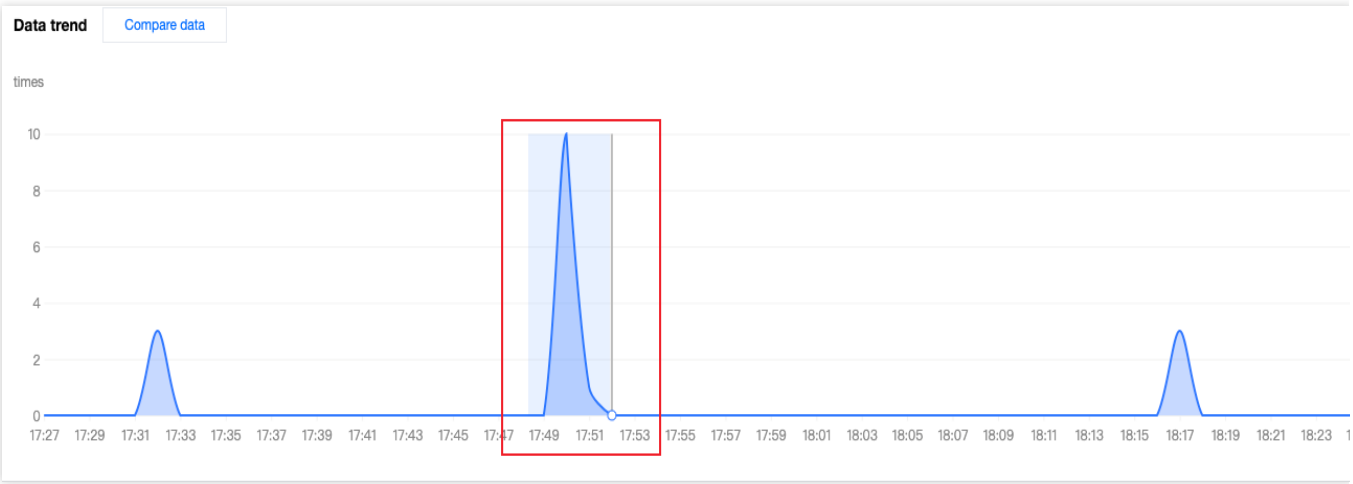


Note:

1. When you select "Last 30 minutes","Last 1 hours", "Last 6 hours", "today", the page will Show the data of the Last 30 minutes,last 1 hour, 6 hours, and the current day (starting from 00:00) and refresh every 5 minutes.
2. The maximum query time range for a single time is 31 days.
3. Due to different Plan versions, different sites may support different data query ranges. For details, please refer to the Plan selection comparison.

Method 2: Select the query time range on the time trend chart

If you want to View the specific time period on the curve, as shown in the figure below, you can select the specific region of the curve by clicking and sliding the mouse on the curve. The time range corresponding to this region will be backfilled to the top filter bar and affect the statistics of other data on the page.



How to Export Statistical Data and Reports

Last updated：2024-08-26 10:01:39

This document describes how to export statistical data and reports from the EdgeOne data analysis page. The specific steps are as follows.

Exporting Statistical Data

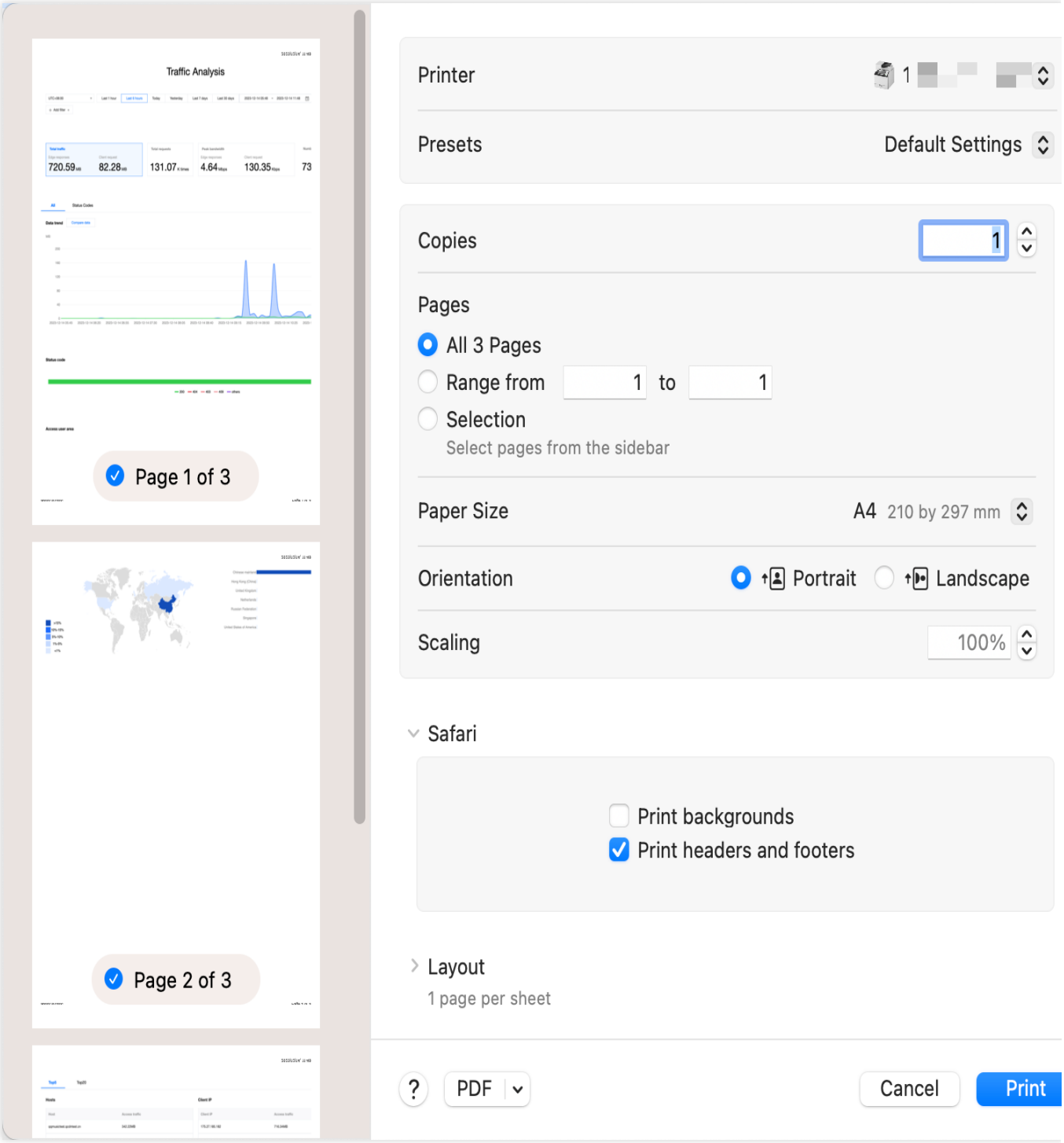
1. Log in to the EdgeOne Console and enter any **Data Analysis** page.
2. Click

to download the corresponding statistical data table. The file format is .csv and the filter conditions on the current page will be applied to the exported data.

Export Report

1. Log in to the EdgeOne Console and enter any **Data Analysis** page.
2. Click on the

located on the top-right corner of the filter bar. EdgeOne will then initiate the browser's print window where you may choose to print or save your report as a PDF. The filter conditions on the current page will be printed in your report at the same time.



Alarm Service
Custom Statistical Metrics

Last updated：2024-12-23 15:24:55

Summary of Features

The feature of custom statistical metrics allows users to meet personalized business monitoring needs through flexible configuration. With this feature, users can precisely customize and track key traffic and performance metrics of websites or APIs, gaining deep business insights. EdgeOne pushes user-defined metrics to the Tencent Cloud Observability Platform (TCOP) and enables users to build customized alarm policies based on these metrics (for example, monitoring access traffic from specific countries or regions), in order to monitor business status in real-time, promptly detect and respond to potential issues, and ensure business continuity and stability.

Restrictions on Use

Supported Scope of Metrics

Base metrics currently supported are as follows:
L7 access traffic
Total traffic
EdgeOne response traffic
Client request traffic
L7 access bandwidth
Total bandwidth
EdgeOne response bandwidth
Client request bandwidth
L7 access requests
Note:
1. The meanings of base metrics can be found in Metric Analysis.
2. Supported filter conditions can be found in How to Use Filter Conditions.

Description of Data Reporting Delay

In the EdgeOne service, data collection and processing is a complex process involving multiple stages. Specifically, EdgeOne collects log information in real-time from nodes around the world. The system then processes the data carefully based on specific filtering conditions set by users in the feature of custom statistical metrics. The processed

data is pushed to the TCOP for further analysis and monitoring by users. As this process covers multiple steps such as data collection, processing, and transmission, there is a certain data processing delay. **Currently, the delay from data collection to the final display on the TCOP is approximately 10 minutes. This means that when an alarm policy is triggered, the alarm notification may be delivered 10-12 minutes after the actual triggering event.**
This delay is acceptable for most monitoring scenarios as it does not significantly impact users' grasp of traffic trends and response to abnormal situations. However, for urgent situations requiring immediate response, users may need to consider this delay and adjust their alarm policies accordingly to ensure that necessary response measures can be taken in a timely manner.

Plan Support Differences

The feature of custom statistical metrics is available only in the Trial, Standard, and Enterprise editions. For the Trial edition, the quota is limited to 10. For a complete comparison of plan support differences, see the Comparison of EdgeOne Plans.

Feature	Individual Edition	Basic Edition	Standard Edition	Enterprise Edition
Custom statistical metrics	Not supported		100/site	100/site

Scenario Examples

Example 1: Monitoring Access Traffic from Specific Countries or Regions

Scenario Description

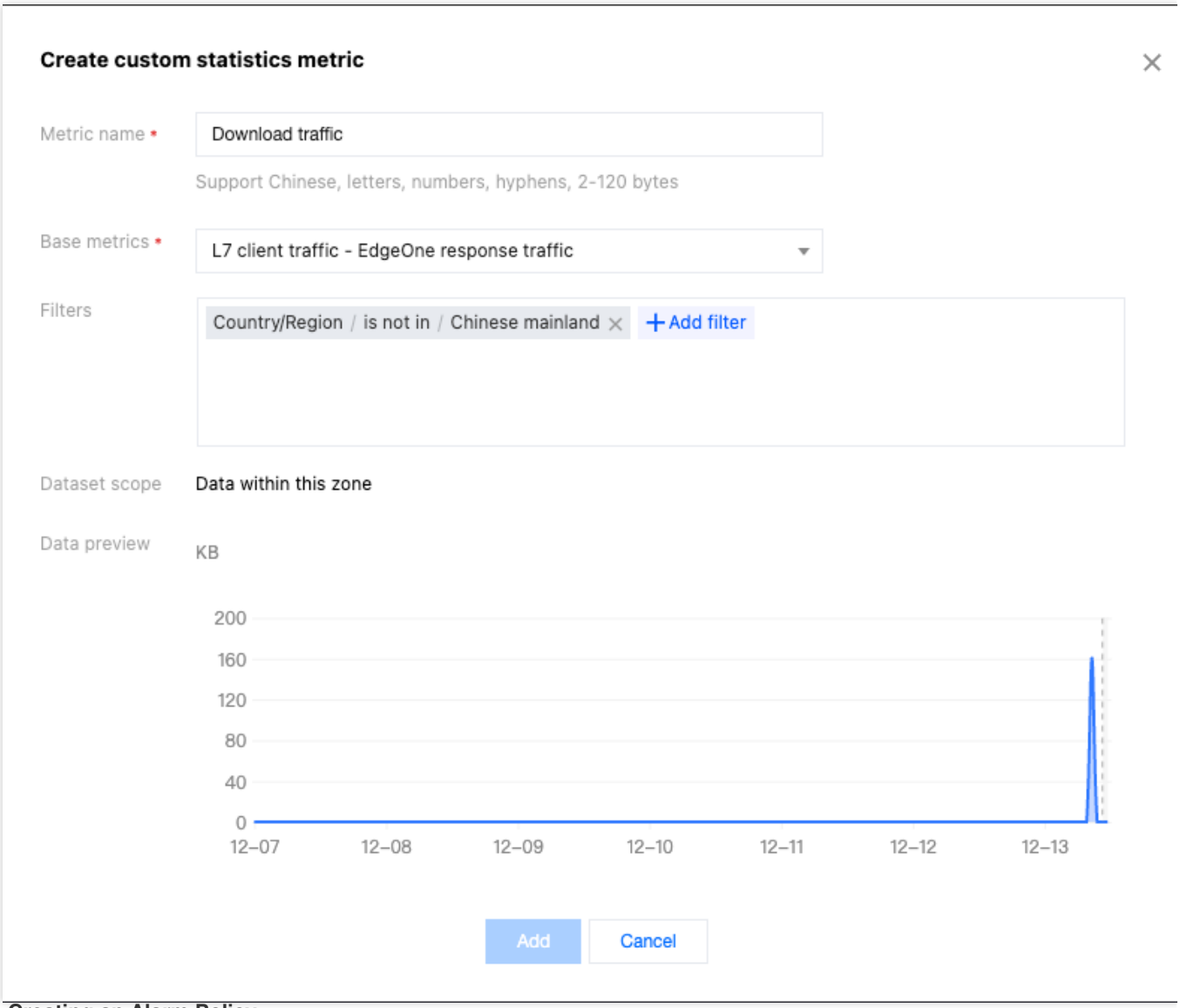
In a globalized business environment, it is crucial for enterprises to monitor traffic from specific countries or regions and configure alarms. This involves not only a quick response to market dynamics but also continuous assurance of service quality. By using the feature of custom statistical metrics, users can achieve detailed monitoring of traffic in key regions and set up alarm mechanisms based on the data to take swift action in case of abnormal traffic.

Directions

Step 1: Creating an EdgeOne Custom Statistical Metric

1. Log in to the Tencent Cloud EdgeOne console, click **Site List** in the left menu bar, and click the **site** to be configured in the site list to go to the site details page.
2. On the site details page, click **Alarm Service** > **Custom Statistical Metrics**.
3. On the custom statistical metrics page, click **Create Custom Statistical Metric**.
4. In the pop-up window, enter the metric name, which supports Chinese characters, letters, digits, and hyphens, with a length of 2 - 120 bytes.

5. Select the base metric **"L7 access traffic - EdgeOne response traffic"**.
6. Set the filter conditions to specify the country/region, such as **"Country/Region is not in Chinese mainland"**. The data preview area will display the data for the past 7 days based on the base metric and filter conditions configured by the user.
7. Click **Add** to save the configuration. The system will start collecting access data under the above conditions and begin pushing it to the TCOP.



Step 2: Creating an Alarm Policy

1. After creating the custom statistical metric, click **Configure Alarm Policy** in the operation column of the custom statistical metric list. The new page will redirect to Tencent Cloud Observability Platform - Alarm Management - Alarm Configuration - Create Policy, and the corresponding EO custom statistical metric will be automatically selected as the alarm object.
2. Enter the **alarm policy name**.
3. Configure alarm triggering conditions based on actual business needs.
4. Click **Next: Configure Alarm Notification**.

4.1 Confirm whether the **preset notification template** meets expectations. If a custom notification template is needed, see Creating Notification Template.
4.2 After selecting the required notification template, click **Complete** to save the configuration.

Example 2: Excluding the Impact of Specific User-Agents on Monitoring Alarms

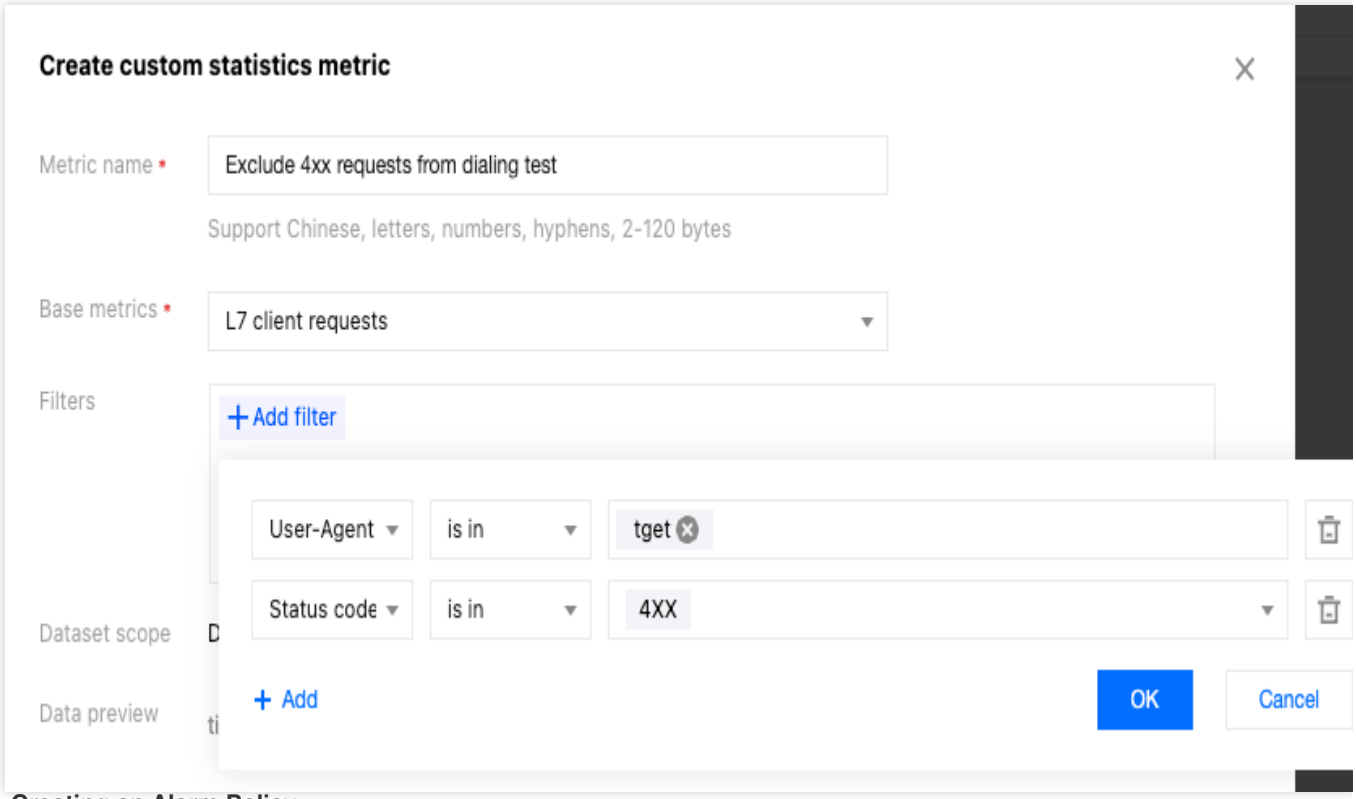
Scenario Description

When website traffic is monitored, it is sometimes necessary to exclude access from certain User-Agents, such as crawlers or testing tools, to avoid the access affecting normal business analysis and alarm triggering. By customizing statistical metrics, you can create monitoring metrics that exclude specific User-Agents and configure corresponding alarm policies.

Directions

Step 1: Creating an EdgeOne Custom Statistical Metric

1. Log in to the Tencent Cloud EdgeOne console, click **Site List** in the left menu bar, and click the **site** to be configured in the site list to go to the site details page.
2. On the site details page, click **Alarm Service** > **Custom Statistical Metrics**.
3. On the custom statistical metrics page, click **Create Custom Statistical Metric**.
4. In the pop-up window, enter the metric name, which supports Chinese characters, letters, digits, and hyphens, with a length of 2 - 120 bytes.
5. Select the base metric **"L7 access requests"**.
6. Set the filter conditions, such as **"User-Agent is not in `tget` " and "Status code is in `4xx` "**. The data preview area will display the data for the past 7 days based on the base metric and filter conditions configured by the user.
7. Click **Add** to save the configuration. The system will start collecting access data under the above conditions and begin pushing it to the TCOP.



Step 2: Creating an Alarm Policy

1. After creating the custom statistical metric, click **Configure Alarm Policy** in the operation column of the custom statistical metric list. The new page will redirect to **Tencent Cloud Observability Platform** > **Alarm Management** > **Alarm Configuration** > **Create Policy**, and the corresponding EO custom statistical metric will be automatically selected as the alarm object.
2. Enter the **alarm policy name**.
3. Configure alarm triggering conditions based on actual business needs.
4. Click **Next: Configure Alarm Notification**.
4.1 Confirm whether the **preset notification template** meets expectations. If a custom notification template is needed, see Creating Notification Template.
4.2 After selecting the required notification template, click **Complete** to save the configuration.