

Documentation Qrator Labs

Network architecture, connecting services and working with the Qrator Labs dashboard

Contents

01	Overview	5
1.1	Qrator Labs network architecture	6
	TECHNOLOGIES	
1.2	Technology overview	8
1.3	Bot protection	9
1.4	CDN	10
1.5	Redirection with cookie installation	12
1.6	HTTP reverse proxy	16
1.7	Reverse TCP proxy	20
1.8	Reverse WebSocket proxy	23
1.9	Load balancing	25
1.10	Ingress filtering	29
1.11	HTTP redirects	33
1.12	Secure DNS server	36
02	Connection to Qrator Labs	38
2.1	An overview of connection methods	39
2.2	Bot protection	40
2.3	Reverse HTTP Proxy	44
2.4	Ingress filtering	47
2.5	Reverse TCP Proxy	49
2.6	Reverse WebSocket Proxy	52
2.7	Tunnels	54
2.8	Qrator.CDN	58
2.9	Sending log files	61
2.10	BGP	73
2.11	Secure DNS server	75

03	Dashboard	77
3.1	Interface overview	78
	LISTS OF RESOURCES	
3.2	Domains	83
3.3	Services	86
3.4	Prefixes	89
	QRATOR	
3.5	Statistics	92
3.6	Services	97
3.7	IP addresses	105
3.8	Ports and protocols	107
3.9	TLS	113
3.10	IP lists	116
3.11	Tunnels	119
3.12	Filter configuration	123
3.13	HTTP redirects	128
3.14	Error pages	132
3.15	Notifications	135
	BOT PROTECTION	
3.16	Overview	138
3.17	Locations	145
3.18	Visitors	150
3.19	Signatures	156
	CDN	
3.20	CDN	161
3.21	CDN settings	164
3.22	Utilities	172
3.23	Certificate storage	174
	ACCESS RIGHTS	
3.24	API tokens	182
3.25	Users	185
3.26	Groups	189
3.27	Notifications	195

3.28	My Profile	196
3.29	SSO	199

04	Help and support	206
-----------	-------------------------	------------

4.1	Additional HTTP headers	207
4.2	Terraform	209

01

Overview

Qrator Labs network architecture

Technologies

1.1 OVERVIEW

Qrator Labs network architecture

The Qrator Labs distributed network is designed such that customers whose traffic passes through it can expect a high SLA even under heavy loads.

The network consists of several *traffic scrubbing centers*, which form one autonomous system (AS). Each center can operate independently of the others: incoming user traffic is analysed and, if deemed legitimate, is forwarded to the customer's network.

BGP Anycast technology is used to direct traffic to the nearest traffic scrubbing center. This means that any traffic scrubbing center can respond to a user request directed to the same IP address from the Qrator Labs network.

Also, through the use of BGP Anycast, routers through which user traffic on the Internet passes, prefer a route to an available traffic scrubbing center that is topologically closest to the user. It is with this center that the user will be able to exchange traffic with as quickly as possible.

Each traffic scrubbing center has a direct link to Tier-1 backbone providers. This ensures that the Qrator Labs network remains available even in case of failures at individual service providers.

**NOTE**

Today the Qrator Labs network includes 14 traffic scrubbing centers in North America, Europe, Middle East, Asia, Russia and CIS. The total network capacity is over 3000 Gbps.

The average Qrator Labs customers' application availability in 2021 was 99,9875%. If availability in any month is less than the SLA promised in the tariff, the customer does not pay for that month.

Receiving traffic from the user

As part of the initial setup, the client, assisted by the Qrator Labs tech support team, makes the necessary settings to ensure that all traffic from users goes to the customer's network not directly, but via traffic scrubbing centers.

The specific steps required to configure traffic reception depend on the selected connection method. These steps are described in the relevant documentation.

Traffic analysis

The purpose of Qrator Labs is to send only *legitimate* user traffic (non-attack traffic) to the customer's network. To do this, Qrator Labs traffic scrubbing centers analyze traffic at all levels of the OSI/ISO network model.

To detect attacks, Qrator Labs uses behavioral, heuristic, and signature algorithms. Also, Qrator Labs automatically adjusts filtering methods using traffic pattern without DDoS attacks for training. This enables even application-layer attacks (L7 by the OSI/ISO model) that resemble the actions of regular users to be detected with high accuracy, and also allows you to distinguish DDoS attacks from increased visitor interest in a site.

**NOTE**

The number of false positives during an attack does not exceed 5%.

The customer can access statistics on detected attacks in their personal dashboard. Attack statistics can also be retrieved via [API](#). This allows integrating Qrator Labs with a monitoring system, the security operation center (SOC) or security information and event management (SIEM) to receive notifications in a customer-friendly format.

The customer can set up blacklists and whitelists that would influence on how traffic is handled. This can be done both in the personal dashboard and via API.

Sending traffic to the customer

Various technologies can be used to communicate with the customer's hardware, depending on what technical capabilities the customer has and what set of services he needs.

For more information about the different ways of setting up a channel between Qrator Labs and the customer, see [Connection](#).

For many of the services provided by Qrator Labs, it is possible to balance or reserve connections to the customer's network. In this case, each traffic scrubbing center can independently (from the others) monitor the status of each client upstream to dynamically change the list of connections under use. For example, if a tunnel to one of the upstreams becomes unavailable from a particular region, the traffic scrubbing center in that region can automatically switch to a tunnel to another upstream to keep the resource available for users.

Technology overview

- Bot protection
- Technology: CDN
- Redirection with cookie installation
- HTTP reverse proxy
- Technology: Reverse TCP proxy
- Technology: Reverse WebSocket proxy
- Load balancing
- Technology: Ingress filtering
- Technology: Secure DNS server

Bot protection

Bots are a threat to many sites. This term refers to many varieties of tools used by competitors and attackers to make automated requests. Bots can range from small scripts to sophisticated software systems that use various techniques to masquerade as ordinary users' browsers. Although bots rarely create a high load on the site, they can perform unauthorized actions, threaten the security of user data and disrupt the functioning of a website.

The primary threats that bots may pose are:

- **Data scraping.**

As a site owner, you may not want your competitors' bots to automatically download and process information hosted on your site. Therefore, such bots may not threaten the performance of your site, but may pose a threat to your business.

Automatically traversing pages and extracting data from them is called *scraping*. The Qrator Labs solution makes scraping difficult by checking users devices' digital fingerprints using JavaScript code. By default, the protection works transparently for legitimate users with minimal added latency and does not resort to captchas or other distractions that would degrade user experience.

To combat data scraping, the following are used:

- [Redirection with cookie installation](#),
- Reverse HTTP proxy.

- **Bruteforce.**

Bots are often created to learn some information by brute forcing. For example, bots can make a large number of login attempts by trying different passwords. In the same way, secret links, promo codes and other parameters that you do not want to make public can be picked up. Such activity is called *brute forcing*.

Qrator Labs detects brute force attacks and specifically blocks traffic from those public IP addresses from where such requests come. This allows you to limit bot activity with minimal damage to legitimate users.

A reverse HTTP proxy is used to combat brute-force attacks.

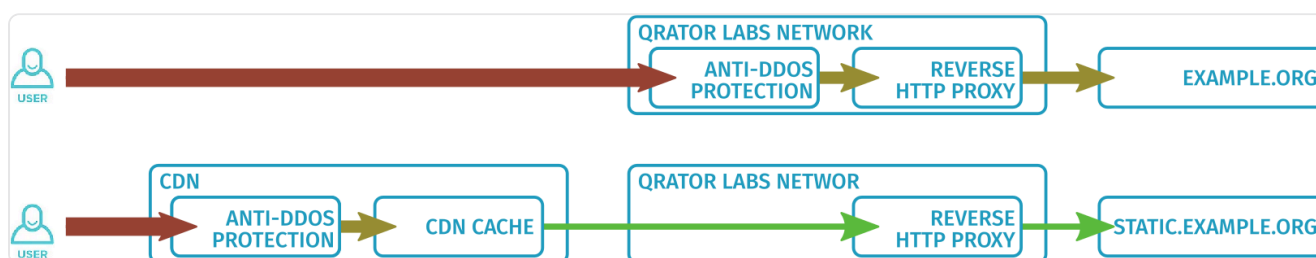
To enable bot protection on your account, please contact the Qrator Labs support specialists. Once the service is activated, you'll be able to configure the anti-bot policy in your dashboard.

1.4 OVERVIEW / TECHNOLOGIES

Technology: CDN

Many sites use third-party CDN (Content Delivery Network) services that store static content on multiple geographically distributed servers. When a user requests content, the request is processed by the CDN server closest to the user. This approach is particularly actively used on sites that operate with large traffic volume, e.g. sites providing multimedia content. The use of CDN speeds up the site for the user and also reduces the load on the server, where the content is hosted.

A common practice is to host all of a site's static content on a separate domain, which will be served using CDN, while the site's main domain will run without it.



If the server where the content is hosted is under Qrator Labs protection (i.e. is an upstream), then additional security settings are required to allow traffic from the CDN network. The CDN is not just outside the customer's network, but also outside the [Qrator Labs network](#). Therefore, requests from CDN to this upstream go the same way as requests from an abstract user. However, their number and nature can be noticeably different. If configured incorrectly, such requests can trigger DDoS attack false positives in the Qrator Labs network. This will immediately affect all users who receive content through CDN.

Qrator Labs offers its own service [Qrator.CDN](#), as well as integrations with many popular CDN service providers. Regardless of the specific provider, the general principle is to correctly identify requests from CDNs and always consider them as legitimate.

The specific way to determine requests depends on which provider's CDN you're using. G2O request signature authentication is used for some providers, while for others, a whitelist-based IP address checking is used. There are also different ways to configure other CDN settings: the settings for [Qrator.CDN](#) are available directly in a [special dashboard section](#) or are configured by the Qrator Labs technical support team, while for most providers you have to configure them at the provider's site, separately from Qrator Labs.

Working with Nginx, Akamai and G-Core

For customers using CDNs from [Nginx](#), [Akamai](#) or [G-Core](#), Qrator Labs automatically adds the IP addresses of the respective networks to attack protection exceptions. So, all traffic passed by CDN is guaranteed to reach the upstream.

All settings are made on the website of the corresponding CDN. Qrator IP should be specified as the upstream for CDN.

Working via G2O

Many CDN providers support request authentication protocol G2O. Under this protocol, two HTTP headers are added to each request. One contains certain additional data about the request, including a randomly generated request ID. The second header contains a signature based on that data. The signature uses a key, one part of which is known to the Qrator Labs network and is used for verification. This way the Qrator Labs network can verify that the request was sent from the CDN.

All the settings are done on the website of the respective CDN. Qrator IP should be specified as the CDN upstream.

Redirection with cookie installation

Redirection with cookie installation is a way to verify the user of a website or mobile application. It is used mainly for [bot protection](#).

The first time a user visits a site, the user's application (e.g., browser or mobile app) receives a [401 Unauthorized](#) status and a page with JavaScript code that collects data about the digital fingerprint of the user's environment and sends it to the Qrator Labs server as a response. If this server detects the environment as a legitimate browser, it assigns a tracking cookie to the user. The user can then browse the site for a while without additional validation, with Qrator Labs automatically extending the validity of the cookie each time. If, however, the Qrator Labs server identifies the user as a bot, it does not set a cookie, but returns an [403 Forbidden](#) error instead of page content.

For different addresses, you can set how Qrator Labs responds to a user without a tracking cookie: either responding with [401 Unauthorized](#) with an environment check code, or responding with [403 Forbidden](#).

The advantage of a digital fingerprint-based solution is that the bot's activity is stopped instantly, preventing it from completing even the first request. This distinguishes this approach from blocking after activity analysis has been completed, which is possible only after several requests.



WARNING

HTTPS resources can be protected using this mechanism only under certain conditions:

1. Forced enabling of HTTPS must be done using [reverse HTTP proxy redirects](#).

HTTP cookies are not available when working via HTTPS and vice versa. Therefore, for bot protection to work correctly, it is necessary that the redirection using the correct protocol is performed before the analysis of the digital fingerprint and installation of cookies.

2. HTTPS proxying with key disclosure must be configured.

Otherwise, the Qrator Labs reverse HTTP proxy will not be able to check or set HTTP headers, which is necessary to handle cookies.

Using captcha

In addition to the standard non-interactive check, in certain cases the Qrator Labs server can also verify the user's legitimacy with a captcha. This check is disabled by default, but it can be enabled after a request to the technical support team.

The captcha will be shown not to all users, but only to those for which the Qrator Labs server could not come to a definitive verdict based on the digital fingerprint analysis.

Using the captcha check helps further minimize the amount of false positive results of the check as a whole. After successfully solving a captcha, the user is considered legitimate and receives a tracking cookie, just like the users whose digital fingerprint was considered legitimate without showing a captcha.

Customizing the HTTP 403 response

If a user that did not pass the check makes a request to an URL for which the “Accept Only with Cookies” rule is set in the [Bot protection → Locations](#) section, the Qrator Labs server sends them a response with the 403 Forbidden code. By default, a basic HTML page is used as the response body, repeating the response code.

You may customize the response in a number of ways.

- **Add more information for the user on the 403 Forbidden page**

Qrator Labs returns an HTML page for the 403 Forbidden status when the `Accept` request header contains the `text/html` type or the header is not set. This page will mostly be seen by legitimate users for which the check gave a false positive result.

Depending on your website's specifics, you may use the page to recommend such users to enable JavaScript, contact you or visit another section of the website.

To make Qrator Labs use your version of the page, upload a `403.html` file to the [Error pages](#) section of the personal dashboard.

- **Provide a JSON version of the response**

Qrator Labs can return a JSON response for the 403 Forbidden status when the `Accept` request header contains the `application/json` type and does not contain the `text/html` type. Such behavior is especially convenient for single-page or mobile applications, which can themselves show an error message to the user, based on the server response.

The response can contain an identifier to help the application distinguish the Qrator Labs server response from an upstream response, as well as some additional information, such as the request timestamp or the string from the `Referer` header.

To enable this behavior and discuss the contents of the JSON response, contact the Qrator Labs technical support specialists.

- **Check the HTTP headers when parsing the response**

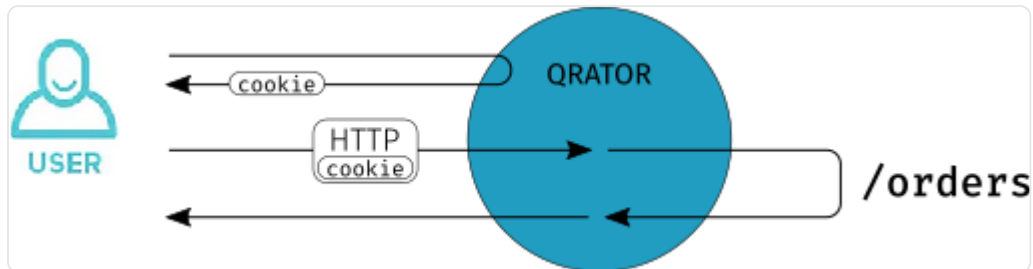
The HTTP headers are another way to inform a single-page or mobile app about which server sent the response: the upstream or the Qrator Labs server. A single header is often enough for that — one that will be present in the Qrator Labs server's response but guaranteed to be absent from an upstream's response.

To enable adding HTTP headers and discuss their names, contact the Qrator Labs technical support specialists.

Integration examples

Protecting your website using cookies

Assume that all pages on your site are meant to be viewed in a browser. In that case, you can set up Qrator Labs protection so that both validation and issuance of a tracking cookie can take place on any page.



The user starts browsing from any page and on the first visit, Qrator Labs performs a validation and sets a cookie. If the cookie expires, the next time the site is accessed, it will be validated again. At the same time, bots will fail the validation and get an `401 Unauthorized` error on any page.

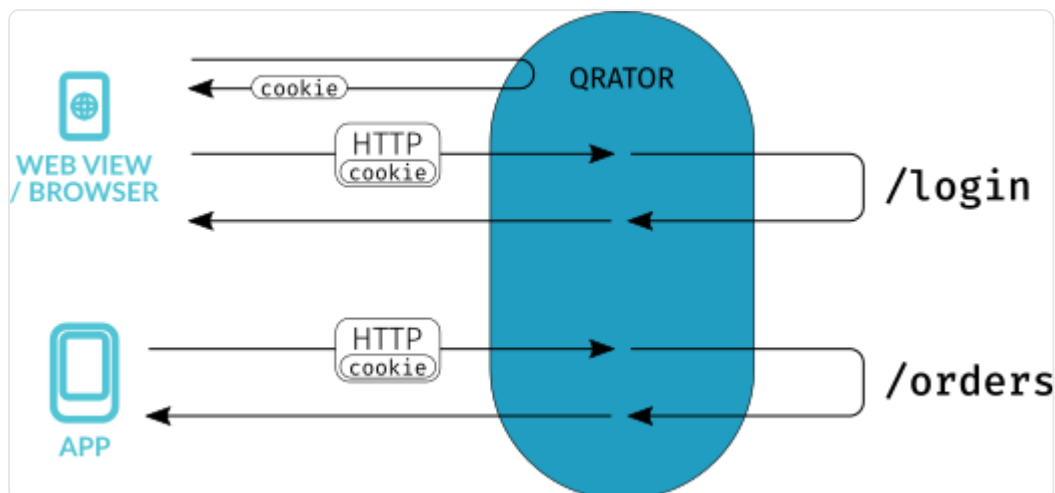
This approach is as transparent to users as possible, but it requires that they always use JavaScript-enabled browsers.

Mobile API protection using cookie

Suppose you have an application that makes requests to your API gateway and can process data received from it, but cannot process an HTML page with JavaScript code. This is a common case, e.g. for mobile apps.

In such a situation, it's recommended to install and check the tracking cookie on different pages:

1. Add a URL to the quick reference card intended to be visited from the browser environment.
2. Enable fingerprint verification with subsequent issuance of a tracking cookie for that page in your Qrator Labs dashboard.
3. In your application client code, add access to this URL through a browser. For example, the application might display the HTML page from the server in the WebView component as a loading screen, and then retrieve the received cookie from the WebView repository. If you are using `captcha`, make sure that the user will be able to enter characters in the text field.
4. The application must use the received cookie in subsequent requests to the API gateway. When the cookie expires, you must direct the user to the page again.



Mobile API protection using hash

For some mobile apps, the [cookie protection method] might not be suitable, as it requires a tracking cookie to be received via a web page.

For such apps, you can use hash verification, which uses HTTP request headers and a secret key known only to you and Qrator Labs:

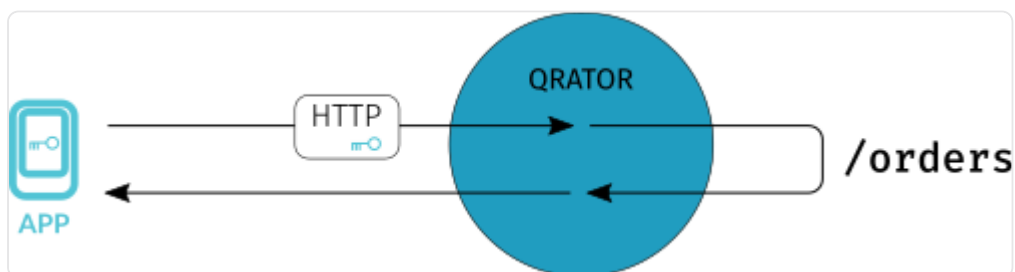
1. In your Qrator Labs dashboard, set the “Accept Only with Cookies” rule for all pages that should be protected.

With this setting, the Qrator Labs server will consider visitors with either the correct cookie or the correct hash in the HTTP headers as legitimate.

2. At the app building phase, add an obfuscated key to it.

Before each request to the API gateway, use that key to calculate the header hash, and add a header with that hash to the request.

Bots that don't have the key won't be able to add the correct hash to requests. Qrator Labs will not send such requests to the API gateway.



HTTP reverse proxy

A significant part of the services provided by Qrator Labs are based on the *reverse proxy* technology. This means that requests destined for your server go to the reverse proxy first, which then forwards them to your server, acting as a client. The same goes for the responses: the server sends them not directly to the user, but to the reverse proxy, and it forwards them to the user.

The server that the reverse proxy accesses is called the *upstream*.

This architecture allows Qrator Labs to solve the following challenges:

- Detect and stop DDoS attacks;
- Block known malicious requests;
- Block [requests from bots](#);
- Perform [balancing](#), distributing load between several upstreams that serves the same service;
- Establish [persistent connections](#) with upstream to save resources.

Reverse HTTP proxy works both with incoming unencrypted traffic via [HTTP](#) and with encrypted traffic via [HTTPS](#). Upstream connections can also be made using both HTTP and HTTPS.

See also: [How to connect reverse HTTP proxy](#).

HTTP proxying

If the site uses HTTP, then in order for the reverse proxy to start handling traffic, you only need to change the DNS A records for the domain.

The user's device contacts the DNS server and gets the IP address of the reverse proxy. After this, direct interaction of the user goes only with Qrator Labs servers. Requests that are not part of a DDoS attack are forwarded to the upstream, with the [X-Forwarded-For](#) header added to them, to which the user's real IP address is added, as well as other [additional HTTP headers](#).

The following situations are an exception:

- If you have added IP addresses to the [blacklist](#), then requests from users from such addresses will not be forwarded to upstream. You can manage the blacklist from your Qrator Labs dashboard.
- If you have configured [HTTP redirects](#), the Qrator Labs reverse proxy will return a response to the corresponding requests on its own, without forwarding the request to the upstream.

HTTPS proxying

For HTTPS sites, the technology works on [the same principles](#) as for HTTP sites: each request from a user is analyzed, and if it came from a legitimate user, the reverse proxy forwards it to the upstream.

But since HTTPS traffic is encrypted, the reverse proxy must decrypt it before processing the request. This requires a TLS certificate and the associated chain of trust and private key. This data is placed in the *certificate storage* and is used to [establish encrypted connections](#) between users and the reverse proxy. Connections between the reverse proxy and the upstream can be set up using both [HTTPS](#) and HTTP.

You can add data to the certificate storage in two ways:

- [Configuring automatic certificate issuance and renewal.](#)
- [Manually uploading files via the dashboard or API.](#)

Automatic certificate issuance

Qrator Labs uses a free service [Let's Encrypt](#) to automatically issue certificates. You can order this certificate [via your dashboard](#), specifying the list of domain names for which the certificate should be issued.



NOTE

Issuing certificates valid for an unlimited number of subdomains (wildcard) is not supported.

Each Let's Encrypt certificate is valid for 90 days after issuance. A few days before this period expires, Qrator Labs automatically sends a certificate reissue request and installs it instead of the outdated one. If automatic renewal is not possible, Qrator Labs will notify you.

To verify domain ownership for Let's Encrypt, the reverse proxy returns a specific token at a specific URL ([HTTP-01 challenge](#)). Note that the Let's Encrypt service will only be able to verify the presence of this token when the DNS A records already point to the reverse proxy's IP address. It is not possible to automatically issue a certificate before changing DNS records. So, after changing DNS records and before the certificate is issued, your resource will not be available to users via HTTPS. If you need it to be available immediately after changing DNS records and updating caches, it is recommended that you obtain and [manually upload the certificate to the storage](#) in advance.

Manually adding files to the certificate storage

You can manually add the certificate and its associated files to the certificate store by first issuing the certificate in any way. For example, you can add the same certificate that was used for the domain before connecting to Qrator Labs.

Each certificate must be accompanied by a chain of trust and a private key.

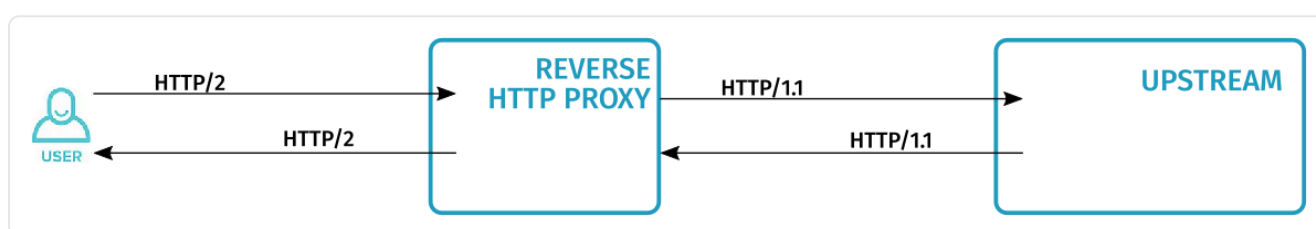
Data can be added either [via dashboard](#) or via API [certrequest_upload](#), [certrequest_install](#) and [sni_link_add](#). X.509 certificates in PEM format are supported. Supported container formats are `pem`, `cer`, `crt`, `der`, `p7b`, `p7c`, `p12`.

HTTP/2 proxying

By default, the Qrator Labs reverse proxy only works over HTTP/1.1, but on your request, technical support specialists can enable support for [HTTP/2](#) to connect to users. This version of the protocol can in some cases greatly speed up resource loading for users whose browsers also support HTTP/2. The use of HTTP/2 is only possible when [using HTTPS](#).

One of the key differences between HTTP/1.1 and HTTP/2 is the ability to load multiple resources simultaneously. When using HTTP/1.1, to request a new resource, you either need to wait until the previous one is finished loading or create a new TCP session. Creating each new TCP session incurs certain costs, and the total number of such sessions in the browser is usually limited. With HTTP/2, the browser can send new requests within an already created TCP session; this approach is called *multiplexing*.

The connection between the Qrator Labs reverse proxy and the upstream is in any case over HTTP/1.1, but due to the large connection pool, the reverse proxy is capable of sending requests in parallel. See [Persistent connections with upstream](#) section below.



Establishing an encrypted connection

When dealing with HTTPS traffic, reverse proxy:

- Supports the following encryption algorithms (cipher suites) for connections to users:

TLS versions	Cipher suites
TLS 1.0, 1.1, 1.2	ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-AES128-SHA256 ECDHE-ECDSA-AES128-SHA ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-ECDSA-CHACHA20-POLY1305 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-CHACHA20-POLY1305 AES128-GCM-SHA256 AES128-SHA256 AES128-SHA DES-CBC3-SHA
TLS 1.3	TLS_AES_128_GCM_SHA256 TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256

**NOTE**

Qrator Labs technical support can change the list of allowed algorithms at your request.

- Supports all possible algorithms and all versions of TLS for upstream connections. The upstream's authenticity is not verified.

**WARNING**

The legacy SSL protocol is not supported for either user or upstream connections.

Speed optimization

Qrator Labs reverse HTTP proxy allows you to optimize your connection performance through several techniques:

- [Persistent connections with upstream](#)
- [TLS offloading](#)
- [{#T}](#)

Persistent connections with upstream

When processing an HTTP request from a website visitor, significant time is spent establishing a TCP transport session. If HTTPS is used, each new TCP session is followed by a *TLS handshake* and generation of session key which also takes considerable time.

A reverse proxy can establish TCP sessions and perform TLS handshakes in advance, forming a dynamic pool of sessions ready for instant payloads. The reverse proxy uses a session from this pool to process each request. One session can be sequentially used to process many requests.

When a user connection is made [via HTTP/2](#) and the user requests multiple resources within the same connection, the Qrator Labs reverse proxy directs those requests to the upstream through different connections from the pool. This way the user can effectively use the HTTP/2 capabilities without having to configure full-fledged HTTP/2 support from the upstream side.

TLS offloading

Using HTTPS implies encrypting all traffic with a session key, which can lead to slower processing of each request than in HTTP protocol. This is especially noticeable on sites with high traffic.

Reverse proxy can work in *TLS offloading* mode. This means that a connection with the upstream can use HTTP, even if the user accesses the resource via HTTPS. The reverse proxy takes all the load associated with TLS and the upstream consumes fewer resources to process each request. This configuration often results in faster resource performance.

Technology: Reverse TCP proxy

Reverse TCP proxy redirects traffic from the user to the client's upstream and vice versa.

Reverse TCP proxy is similar in principle to reverse HTTP proxy, but operates at the transport layer of the OSI/ISO network model and parses TCP traffic. Malicious TCP traffic is cut off and does not reach the upstream. So, Qrator Labs reverse TCP proxy protects sites from DDoS attacks at the transport layer (L4).

TCP proxy can also be used to protect against DDoS attacks at the application layer (L7), acting as an alternative to reverse HTTP proxy. To do this, you need to configure [automatic sending of logs](#) from your server to the Qrator Labs server.



WARNING

Note that this is not equivalent to full application-layer traffic processing (TCP proxy does not work at that level), so you cannot use such a proxy for web application filter or anti-bot protection. These products work strictly at the application layer and require an HTTP reverse proxy.

Therefore, the reverse TCP proxy is suitable for handling traffic in many situations where [reverse HTTP proxy](#) or [tunnel](#) cannot be used, for example:

- When the server needs to see user TCP segment headers;
- When it's necessary to protect a resource running on protocols other than HTTP and HTTPS from DDoS attacks;
- When it's necessary to establish a "point-to-point" connection with a server running Windows.

Qrator Labs reverse TCP proxy supports operation both in [Proxy Protocol mode](#) for upstreams supporting this protocol and in [Raw TCP Proxy mode](#) for any upstreams.

Proxy Protocol mode

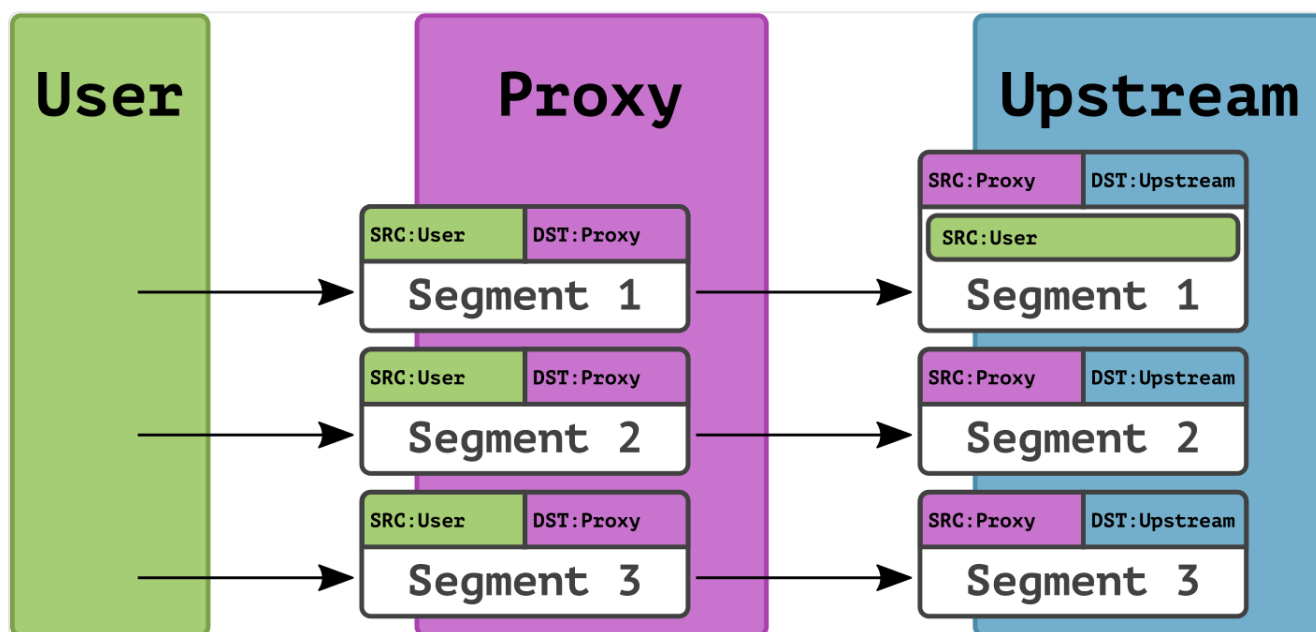
If your upstream software supports [Proxy Protocol](#), the reverse TCP proxy can send it both the contents of user TCP segments and information about the IP addresses they came from. The upstream should also send return traffic using Proxy Protocol through Qrator Labs network from where they will be redirected to the user.

**EXAMPLE**

Proxy Protocol support is built into financial broker trading platform MetaTrader, so a reverse TCP proxy can protect the trading gateways on that platform against DDoS attacks. In this case, instead of receiving TCP traffic from users directly, the MetaTrader gateway in the broker's network only receives it from the reverse TCP proxy in the Qrator Labs network. Illegitimate traffic from users is not sent to the broker's network and does not threaten the gateway performance.

Under Proxy Protocol, at the beginning of each TCP connection, the upstream receives a header with information about the IP addresses of the sender and receiver of the original segment. All other data can be obtained directly from the forwarded segment. It means that after proper header processing, the software on the upstream side can implement the same functions as when receiving TCP traffic directly without reverse proxy, e.g. logging sender's IP address or performing other actions.

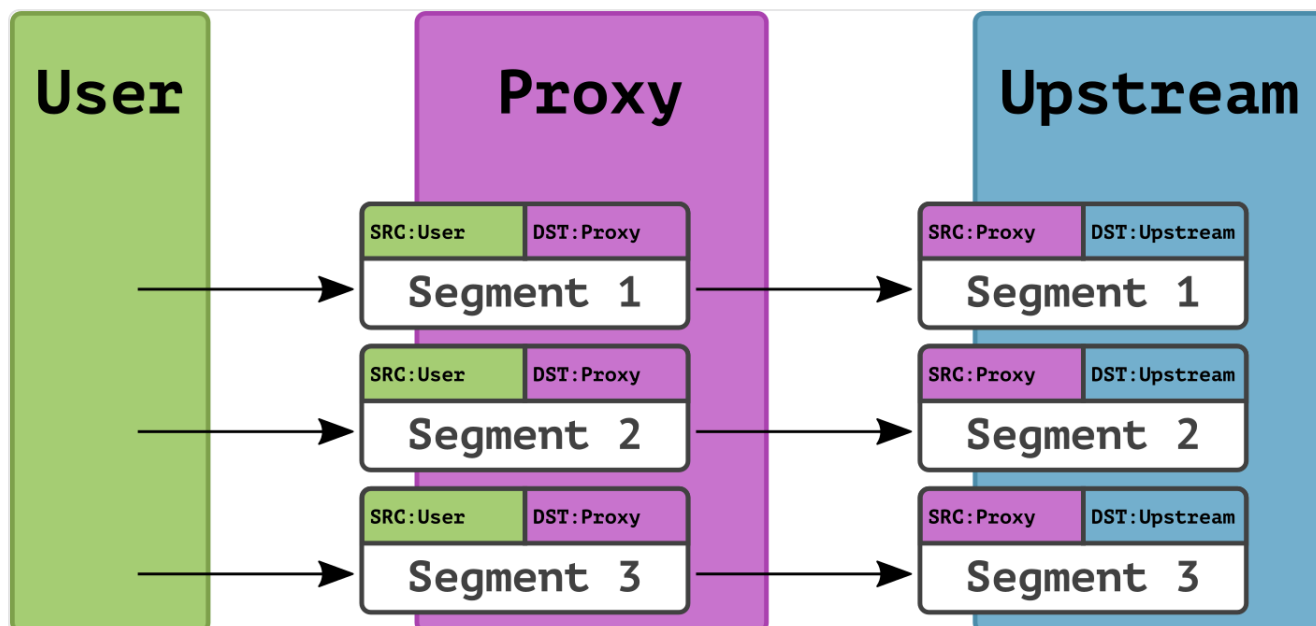
In the figure, the upstream substitutes a separately received header for each segment received within that connection, thus recreating all the information sent by the user:



For information on how to connect a reverse TCP proxy in Proxy Protocol mode, see [relevant section](#).

Raw TCP Proxy mode

To protect upstreams that do not support Proxy Protocol, use a reverse TCP proxy in Raw TCP Proxy mode. In this mode, having received a TCP segment from a user, the reverse TCP proxy simply proxies this segment, acting as the sender of the segment for the upstream. No additional information about the original sender is passed to the upstream.



When using Raw TCP Proxy, the upstream cannot get information about the user's IP address, unless it is provided by a higher-layer protocol. That's why this mode is mostly used for protocols that provide for a way of authentication that doesn't rely on the sender's IP address. For example, with Raw TCP Proxy, you can protect against DDoS attacks that are aimed at the SSH server behind an upstream.

The main advantage of Raw TCP Proxy over Proxy Protocol is that you don't need to configure it on the upstream side. The upstream accepts legitimate TCP segments from the Qrator Labs network just like it would accept them directly from users.

Qrator Labs Raw TCP Proxy protects resources from DDoS attacks no higher than the transport layer (L4).

For information on how to connect Raw TCP proxy, see [relevant section](#).

Technology: Reverse WebSocket proxy

Many modern web applications use not only HTTP and HTTPS, but also WebSocket ([RFC 6455](#)) for data transfer. For example, a web application frontend can work through HTTPS, and a video stream can be loaded through WebSocket. WebSocket upstreams are often located on the same IP addresses or networks as HTTP upstreams, so unprotected WebSocket traffic can make it easy for an attacker to attack your network even when HTTP traffic is fully protected.

Therefore, clients developing such multi-protocol web applications should ensure that all application traffic passes through the Qrator Labs network. To do this, you should use a combination of [reverse HTTP proxy](#) and *reverse WebSocket proxy*.

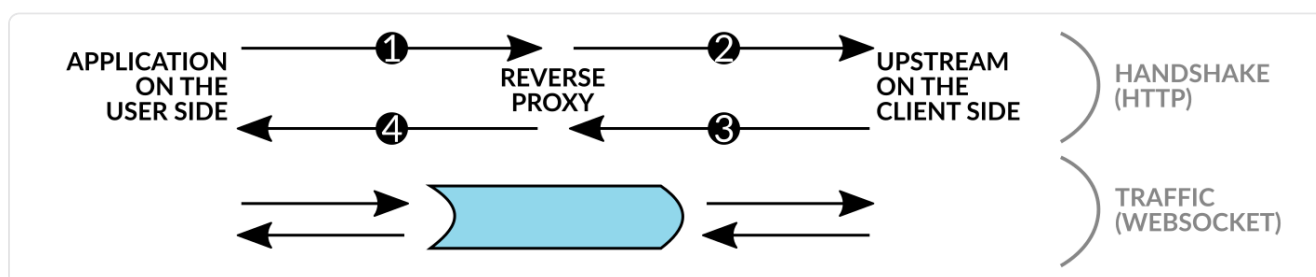
For information on how to start using the Qrator Labs reverse WebSocket proxy, see [Connection: Reverse WebSocket Proxy](#).

How it works

Part of the WebSocket protocol is the handshake, which the user and the upstream must complete before data transfer can begin. The handshake itself uses the HTTP protocol. After the handshake, both parties can send WebSocket traffic to each other until the connection is closed by one of the parties.

The reverse WebSocket proxy proxies the user request and the server response, with additional HTTP headers added to the request. The reverse proxy then accepts and proxies WebSocket traffic both ways through the connections established during handshake.

It is the responsibility of the application to parse and process the data transferred after the handshake. The reverse proxy does not parse or modify WebSocket traffic, which allows it to be used regardless of the format of the data being sent.



Below is a detailed description of the handshake procedure involving a reverse WebSocket proxy.

1. The user sends an HTTP [101 Switching Protocols](#) request.

Upon receiving this request, the reverse proxy checks if the number of requests from the user's IP address exceeds the [client-specified limit](#). If the limit is not exceeded yet, the reverse proxy selects upstream for this user based on his IP address. See [Balancing](#).

2. An HTTP `101 Switching Protocols` request is proxied to the selected upstream with the headers the user specified in his request as well as `additional HTTP headers`.

**WARNING**

By default, the reverse proxy is configured to remove headers containing an underscore `_` before passing the request to the upstream. If you need to keep such headers in the request, contact Qrator Labs technical support to change this setting.

3. The reverse proxy waits 5 seconds for a response from the upstream. If the upstream does not respond, the reverse proxy returns to step 2 and attempts to contact the client's other upstreams. Read more about this in [Balancing](#).
 - If no response is received from any upstream, the reverse proxy returns an HTTP `400 Bad Request` response to the user and the attempt to start the WebSocket protocol is abandoned.
 - If the upstream responds with an error message with any status, connections are dropped and the attempt to start the WebSocket protocol is abandoned.
 - If the upstream responds with a handshake with an HTTP `101 Switching Protocols` status and correct headers, the reverse proxy proxies this response to the user unchanged. In this case, the connection between the reverse proxy and the upstream is considered successfully established.
4. The user checks the status and headers of the received response. This completes the handshake, and the WebSocket connection between the user and the reverse proxy is considered successfully established. The HTTP protocol is no longer used for this connection.

All further information transmission from the user to the upstream and back takes place through the reverse proxy within the already established connection pair.

The reverse proxy does not analyze the forwarded data and does not change its content during transmission, which allows it to work correctly regardless of the format of the transmitted data.

When one party closes the connection to the reverse proxy, the reverse proxy automatically closes the connection to the other party.

Load balancing

If your resource is serviced by several servers, you can set up several *upstreams* corresponding to these servers in your dashboard. In this case, Qrator Labs reverse proxy can be used not only to protect against illegitimate traffic, but also to balance legitimate traffic between upstreams.

Balancing is done independently on each of the Qrator Labs traffic scrubbing centers, which are located in different regions of the world.

You can configure load balancing in the “Services” section of your dashboard.

Switching between upstream lists

In your dashboard, you can set up one or two upstream lists (by default, only one list is enabled). The first list is considered the main list, and the second, the backup list.

By default, each traffic scrubbing center balances all requests only between upstreams from the main list, while unavailable upstreams (detected via active or passive checks) are temporarily removed from circulation.

Automatic switching to the backup list and back is performed on the basis of [upstream health check](#):

- If all upstreams from the main list have become unavailable for the traffic scrubbing center, the center switches to using the backup list instead of the main one.
- As soon as the checks reveal that at least one upstream from the main list is available again, the traffic scrubbing center switches back to using the main list.

Upstream selection algorithms

Qrator Labs supports two upstream selection algorithms: [Round robin](#) and [IP hash](#). In the dashboard, you can specify which of these algorithms should be used. The selected algorithm will be used both for balancing between the main upstreams, and for balancing between the backup upstreams.

The work of each algorithm can be influenced by specifying an integer weight for the upstream. Depending on the algorithm used, the upstream weight is interpreted differently. Upstreams with zero weight are considered disconnected and do not participate in balancing, they do not receive traffic. If weights are not specified, it is considered that the weight of each upstream is equal to one.

Round robin

The *round robin* algorithm works with the list of upstreams as with a queue in the order in which you placed them in your dashboard. By default, each upstream is used to process one request before the queue moves on to the next upstream. So, the algorithm ensures equal distribution of requests among all upstreams from the corresponding list.

The weight of an upstream determines what fraction of requests will be directed to that upstream.

**EXAMPLE**

If you use the round robin algorithm for two upstreams with weights [1](#) and [2](#), of every three requests, the first will be handled by the first upstream, and the second and third by the second upstream.

IP hash

The *IP hash* algorithm selects an upstream depending on the IP address of the request, with repeated requests from the same IP address passed to the same upstream. This algorithm is useful if one wants all requests from the same user to be handled by the same upstream, e.g. for more efficient use of the cache on each server.

The weight of an upstream determines the probability of selecting that upstream and assigning it to a user.

**EXAMPLE**

If you use the IP hash algorithm for two upstreams with weights [1](#) and [2](#), of every 3 000 first requests from users with unique IPs, on average about 1 000 will be sent to the first upstream and about 2 000 to the second. Distribution of further requests from these users depends on the users themselves and can be very different from the initial one.

Upstream health check

One of the advantages of using multiple upstreams is the ability to increase the fault tolerance of the resource as a whole. If one of the upstreams becomes unavailable or stops processing requests quickly, the remaining upstreams can ensure the availability of the resource.

Upstream availability is tracked separately from each Qrator Labs traffic scrubbing center.

There are two available upstream health check methods:

- [Passive health check](#) occurs during normal requests;
- [Active health check](#) is done regularly with a special request.

Passive health check

In passive health check, the traffic scrubbing center does not make special diagnostic requests to upstreams, but analyzes errors when processing user requests.

Only safe requests are used for analysis (see [Error handling](#)). If more than a third of such requests to an upstream, made within three seconds, ended with errors, then the traffic scrubbing center stops making requests to that particular upstream according to the usual algorithm. Instead no more than one request per three minutes is sent to that upstream, and if the processing of another request succeeds, the upstream is considered available again.

Note that some types of requests are not repeated after unsuccessful submissions; see [Error handling](#). Therefore, users may see errors more often with passive than with active health check.

Active health check

To respond to upstream unavailability before it manifests itself in multiple errors when processing requests from real users, you can use active checking. With active checking, the traffic scrubbing center regularly makes diagnostic requests to a specific URL on each upstream. A decision on whether to use an upstream is made based on the results of such requests.

All active health check parameters are set individually by contacting technical support:

- Which request will the traffic scrubbing center send to an upstream?

The protocol (HTTP or HTTPS), URL and HTTP method are configured. If necessary, you can also specify request parameters e.g., a special token so that the upstream can distinguish diagnostic requests from user requests.

- How often do you need to perform a check?

The shorter the interval between requests, the faster the traffic scrubbing center will be able to detect problems. However, too frequent requests can sometimes affect the upstream load. The optimal interval depends on your upstream performance.

- Which check results are considered successful?

Depending on the characteristics of your resource, either responses with all status codes except server errors (`HTTP 5xx`) or only responses with the `HTTP 200` status code may be considered successful.

Besides, you can specify how many times in a row the check must fail before the traffic scrubbing center considers the upstream to be unavailable and temporarily stops sending user requests to it. Similarly, you can configure under what conditions the upstream is considered available again.

Error handling

After each request to the upstream, the Qrator Labs traffic scrubbing center analyzes whether the request was successful or an error occurred. Errors are displayed in the [“Statistics”](#) section of the dashboard, and also influences the subsequent behavior of this traffic scrubbing center.

Any of the following situations is considered an error:

- Connection to an upstream is lost and the traffic scrubbing center fails to re-establish connection within 15 seconds.
- Upstream failed to respond to a request within 60 seconds or sent an incomplete (incorrect) response.
- Upstream returned a response with the `502 Bad Gateway` or `504 Gateway Timeout` status code.

The error handling procedure is different for *unsafe* and *safe* requests (see [RFC 7231](#)):

- POST, PUT, PATCH, DELETE, CONNECT requests are considered unsafe.

Such requests are not redirected to the same or to another upstream as this may break the logic of the application. The user always gets the first upstream response or, if the upstream fails to respond, a response with a `502 Bad Gateway` or `504 Gateway Timeout` status code depending on the cause of the problem.

- GET, HEAD, OPTIONS, TRACE requests are considered safe.

Before returning an error to the user, the traffic scrubbing center sequentially sends the request to other upstreams from the list and waits for responses from them. If any upstream handles the request without errors, its response will be returned to the user. If the traffic scrubbing center goes through all upstreams without receiving a successful response, it will return the error response from the last upstream.

Meanwhile, if no response was received from the last upstream, then the traffic scrubbing center itself will generate a response with the `502 Bad Gateway` or `504 Gateway Timeout` status code depending on the cause of the problem.

If the user terminates the connection without waiting for a response (e.g., due to timeout), attempts to process the request will stop.

Each error is reflected in the dashboard in the “Analytics” and “Summary” sections. To quickly detect problems with resource availability, contact Qrator Labs technical support specialists, they will set up automatic notifications that suit you, which will be triggered after a certain number of errors.

Technology: Ingress filtering

Protecting transit networks that carry traffic to multiple end-user networks is different from protecting the end-user networks themselves.

- The throughput of a transit network can suffer from DDoS attacks aimed directly at it, as well as from attacks aimed at any of the transit networks.

That said, transit network owners need to keep resources available on all end-user networks as much as possible, even during an attack on some of them.

- Protection of a transit network must rely only on traffic analysis at the network and transport layers, since the end-user networks may use different higher-layer protocols.

Qrator Labs provides ingress filtering to secure transit networks. It allows responding effectively to attacks, either by keeping the resource fully available for each end-user network or by limiting the availability of only specific networks under attack.

To use ingress filtering, the transit network owner specifies *prefixes* corresponding to the protected subnets and configures it in the Qrator Labs dashboard. See [Connection: Ingress filtering](#). This process is transparent to end-user networks, no settings need to be made on their side.

Differences from end-user network protection

The key difference between DDoS attack protection for end-user networks and for transit networks is that threats for transit networks are detected only at the network and transport layers (L3, L4 according to the OSI model) and does not require analysis of outbound traffic.

In many cases, this saves resources and avoids analyzing traffic whose source is known and legitimate. In particular, when serving streaming and other multimedia services, where outbound network traffic can be dozens of times greater than inbound traffic, it often makes sense to pass inbound and outbound traffic through different networks and enable ingress filtering only for the network through which inbound traffic passes.



Traffic accounting

The Qrator Labs network counts and filters incoming traffic for prefix sets configured by the customer. All traffic within one prefix set is divided into three categories, whose traffic is counted independently.

- TCP segments SYN and ACK, which are used to establish new sessions, are counted as *TCP Unverified*. Incorrect segments and segments within unknown sessions fall into this same category.
- TCP segments sent within already established sessions are counted as *TCP Verified*.
- Any other segments, including UDP and ICMP traffic, are counted as *Other*.

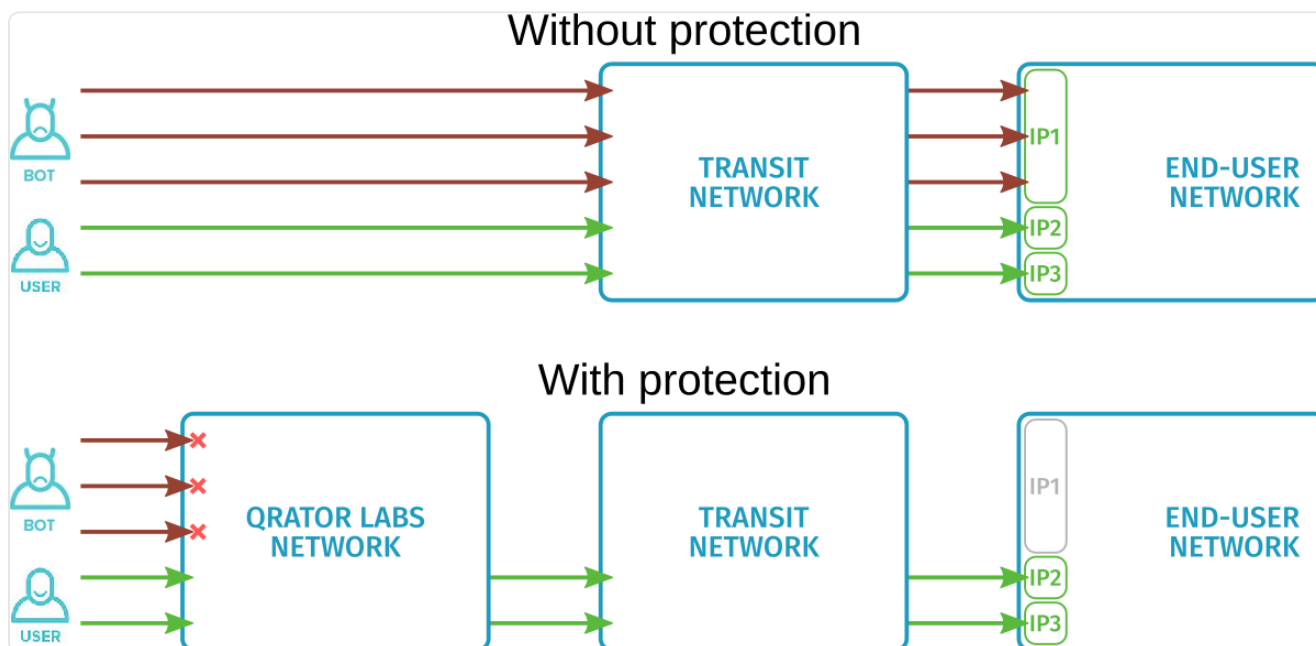
When configuring [filter policies](#), you can set arbitrary limits on each of these traffic categories separately, as well as set a limit on the total traffic volume.

The Qrator Labs network also keeps statistics on distribution of traffic by destination network IP address. This is not related to any limits per se, but plays a role in [blocking busy IP addresses](#).

The segment counter of each category and the total traffic counter are reset once every few seconds (the exact size of the interval may vary depending on load). If any of the counters have exceeded or are approaching the corresponding limit before this period expires, the Qrator Labs network [blocks busy IP addresses](#) or applies the [SYN cookies] mechanism(#syn-cookies) in order to protect the transit network.

Blocking busy IP addresses

This attack-blocking method is applied when the *TCP Verified*, *Other* or total traffic limit is reached (see Traffic Accounting).

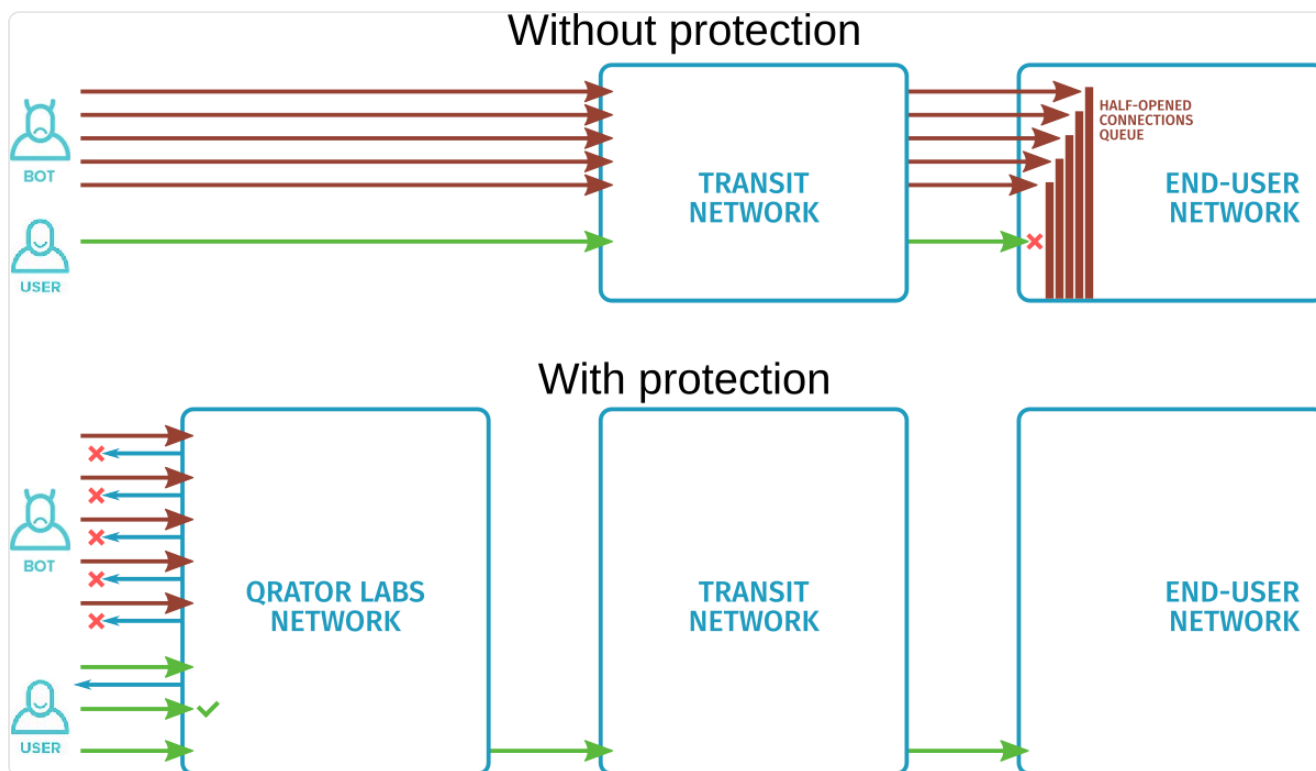


Using traffic distribution statistics for IP addresses on the end-user network, the Qrator Labs network blacklists those IP addresses that have received the most traffic in the last few seconds. The number of temporarily blacklisted addresses is calculated in such a way that when the traffic to these addresses is no longer delivered, the remaining traffic is again no higher than the limit.

This solution reduces the load on the transit network during an attack by temporarily blocking the minimum possible number of IP addresses on end-user networks.

SYN cookies

The SYN cookies mechanism is used when a SYN flood attack is launched against the server and the *TCP Unverified* traffic limit has been reached (see [Traffic Accounting](#)).



In a SYN flood attack, the attacker sends multiple TCP segments to the server, specifying a foreign IP address as the source. The server tries to continue the TCP handshake with this address and never gets a reply. This causes half-opened connection data to remain in the queue for a long time, causing it to overflow and making it impossible to establish a session with legitimate users.

The task of ingress filtering at the moment of such an attack is to deliver TCP segments to the transit network only from legitimate users with real IP addresses. TCP segments from unverified IP addresses are responded to by the Qrator Labs network itself, thus trying to continue the handshake. This response TCP segment is chosen so that the next segment from the user within this handshake can be unambiguously determined whether it received a segment from the Qrator Labs network. IP addresses that pass this check are added to the whitelist. The SYN cookie algorithm is described more fully in [RFC 6013](#).

This process does not use the half-open connection queue, so the volume of traffic from the attacker does not affect the ability of a legitimate user to receive a response TCP segment and complete the handshake.

Upon successful completion of the check, the Qrator Labs network immediately resets the session. Most modern applications are resistant to TCP session resets and they attempt again to establish connection automatically. Since the user's IP address is already on the whitelist, all new segments are sent to the transit network without restriction.

Filtering policies

The customer can define one or more ingress filtering *policies* in his Qrator Labs dashboard. The customer can set the created policy for any set of prefixes within his network or for his entire network.

It is possible to set multiple policies for the same prefix. In this case, once one of the limits in one of the policies is reached, restrictions can be imposed on the IP addresses from the prefix even if limits in the other policies have not yet been reached.

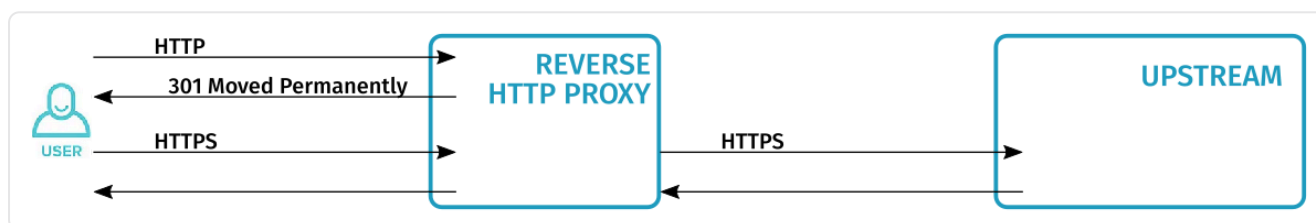
Each policy is a set of [traffic accounting](#) limits. The policy can include separate limits for *TCP Verified*, *TCP Unverified*, *Other* traffic categories, as well as an overall *Total* counter for all traffic. Some of the limits can be left unset by the customer.

Limits can be set both in bits per second and in packets per second. Bits-per-second limits help protect against bandwidth-exhaustion attacks, while packets-per-second limits help defend against those attacks that occupy network hardware with many small segments, e.g., from a TCP SYN flood attack.

Technology: HTTP redirects

Customers using the Qrator Labs [reverse HTTP proxy](#) can set up some kinds of HTTP redirects on the reverse HTTP proxy side. Using this service, requests requiring redirection will be handled directly by the reverse HTTP proxy without reference to the upstream.

One popular use of HTTP redirects is to force HTTPS for all user requests (see [configuration example](#) at the end of the article). In this case, handling HTTP requests on the customer's network can be disabled completely because only HTTPS requests will reach the upstream. This configuration reduces the number of requests sent to the upstream.



WARNING

If you plan to enable bot protection for a domain (see [Redirection with cookie installation](#)), it is important that redirects from HTTP to HTTPS are performed not by the upstream itself, but by Qrator Labs reverse proxy. Otherwise bot protection will not work correctly.

Principle of operation

In the Qrator Labs dashboard, [HTTP redirects](#) with the list of rules can be found in the domain settings. By creating and editing these rules, you can control the behavior of the reverse HTTP proxy. Rules can be added in any order.

When an incoming request arrives, the reverse proxy searches the entire list of rules for the most specific rule that can be applied to the request. This allows you to configure complex combinations of redirect rules. See [Examples](#).

Request	Redirect		
http:// ANY	301 https:// \$hostname \$path ? \$args		
http://example.com	301 https://www.example.com \$path ? \$args		
https://example.com	301 https://www.example.com \$path ? \$args		
http://old.example.com	Disabled		

Each rule has conditions under which it is triggered ("Request"):

- The protocol that the user used in the original request – HTTP or HTTPS,
- The host the request is intended for – a specific string or the "ANY" value corresponding to any host.

If the reverse HTTP proxy finds a rule in the list that matches the current request, it performs the actions described in it ("Redirect").

- The user can be sent an HTTP response code of your choice from the list:
 - 301 Moved Permanently
 - 302 Found
 - 307 Temporary Redirect
 - 308 Permanent Redirect
- When compiling the URL to which the redirect will be made, the host and path you typed or copied from the original request will be used.
- GET arguments may be copied from the original request or deleted.
- Redirection for a given rule can be turned off completely. In this case, requests matching this rule will be forwarded to your network.

Examples

Forced HTTPS enabling

Request	Redirect
http:// ANY	301 https:// ^\$hostname` \$path ? \$args

In this configuration, the reverse HTTP proxy replaces the protocol in all incoming requests from HTTP to HTTPS. It is recommended to use the 301 Moved Permanently code so that the user's browser can cache the response on its side and not perform HTTP requests next time.

Creating exceptions to a rule

Request	Redirect
<code>http://^ANY</code>	301 <code>https://`\$hostname` \$path ? \$args</code>
<code>http://old.example.com</code>	<i>Disabled</i>

This example is similar to [example with forced redirection to HTTPS](#), but it adds a second rule which says that for requests to `old.example.com`, no redirection should be performed. The reverse HTTP proxy prefers a more exact match when matching the request with rules, so the first rule stops being applied for requests to `old.example.com`.

This configuration is useful if you have one or more sites served by your upstream that does not support secure HTTPS, but you want to force this protocol on all other sites.

Redirecting to a new host

Request	Redirect
<code>https://^ANY</code>	301 <code>https://www.example.com/</code>
<code>https://www.example.com</code>	<i>Disabled</i>

Suppose your content is hosted on `wiki.example.com`, `app.example.com` and other `*.example.com` subdomains served by the same upstream, but you need to redirect all visitors from those hosts to the main page on a new main host `www.example.com` after a structural change.

In this situation, you can create two rules as described above. The first rule applies to all HTTPS requests, but the second rule creates an exception to the first rule for the primary host, thus preventing cyclic redirections.

Technology: Secure DNS server

Ensuring that your DNS server is available is an important part of protecting your web service. If an attacker makes the DNS server unavailable, for most users it could mean that your site will be inaccessible or that the application accessing your domain is not working. Besides, information about the unavailability of the DNS server is usually stored for some time in the cache on the user's device or on the recursive DNS server they use, such as Google Public DNS or the recursive DNS server of the Internet service provider. Because of this, users may continue to have problems accessing your site even after the attack on the DNS server has stopped.

To protect against DNS attacks, you can use Qrator Labs *secure DNS server*. Below you will find the [general principle of operation](#) of this service and descriptions of the two possible modes of operation: [secondary DNS server mode](#) and [proxy DNS server mode](#).

Instructions on how to connect the service can be found here: [Connection: Secure DNS server](#).

Principle of operation

A DNS attack is usually in the form of a large number of simultaneous requests to it. The attacker expects that since the cost of processing a DNS request exceeds the cost of creating it, the attacked DNS server will either slow down significantly or stop processing requests from legitimate users altogether. Qrator Labs DNS servers can remain functional during attacks due to the high bandwidth of the Qrator Labs network where the servers are hosted.

When setting up the service for the first time, the customer specifies the Qrator Labs DNS server as the authoritative server for his DNS zone (see [Connection: Secure DNS server](#)). Thus, the user makes requests not to the customer's DNS server, but to one of Qrator Labs DNS servers. These servers translate the data received from the customer's DNS server in one of two ways:

- By transferring the DNS zone (see [Secondary DNS server mode](#))
- By proxying DNS responses (see [Proxy DNS server mode](#))



WARNING

Regardless of the mode you choose, it is recommended to switch to Qrator Labs secure DNS server in advance, without waiting for a possible attack.

Changing the DNS server initiated during an attack often fails to quickly mitigate the damage since both users and recursive DNS servers keep trying to access the former DNS server until the cache lifetime expires.

Secondary DNS server mode

This mode uses the standard zone transfer mechanism, where the customer's DNS server is primary and Qrator Labs DNS server is secondary. The mechanisms of interaction between the primary and secondary DNS servers are described in [RFC 1996](#) and [RFC 5936](#).

The Qrator Labs DNS server queries the primary server for complete information about the zone, including the SOA (start of authority) record, and responds to user requests according to the information received.

By default, the primary server is contacted again after the period specified in the SOA record refresh field has elapsed. The primary server can also send a notify request to the Qrator Labs DNS server, in this case the Qrator Labs DNS server requests full zone information from the customer's DNS server immediately.

If the customer server is unavailable, the Qrator Labs DNS server continues to use data from the last successful update.

Proxy DNS server mode

In this mode, the customer and Qrator Labs DNS servers do not enter into a primary and secondary server relationship and communicate without authorization. Upon receiving a new request from the user, the proxying server redirects it to the customer server unchanged. If the customer server responds to the request, that response is redirected to the user and also stored in the cache.

The default cache expiration time for a negative response (response that no record was found) is 600 seconds (10 minutes), and for a positive response, the TTL is defined in the customer server response. During this time, all users who make exactly the same request to the Qrator Labs DNS server receive a response from the cache. After the TTL expires, the requests are redirected back to the customer server to refresh the cache. If the customer server is unavailable on the next update, the Qrator Labs DNS server continues to use data from the cache.

Thus, at normal times, the proxying Qrator Labs DNS server reduces the load on the customer DNS server by using the cache, and in contingencies does not allow any temporary unavailability of the customer DNS server to cause the negative cache to spread to other DNS servers.

When using this mode, there is no way to force the cache to refresh before the TTL expires.

02

Connection to Qrator Labs

An overview of connection methods

Bot protection

Reverse HTTP Proxy

Ingress filtering

Reverse TCP Proxy

Reverse WebSocket Proxy

Tunnels

Qrator.CDN

Sending log files

BGP

Secure DNS server

2.1 CONNECTION TO QRATOR LABS

An overview of connection methods

- [Connection: Bot Protection](#)
- [Reverse HTTP Proxy](#)
- [Connection: Ingress filtering](#)
- [Connection: reverse TCP Proxy](#)
- [Connection: Reverse WebSocket Proxy](#)
- [Connection: Tunnels](#)
- [Connection: Qrator.CDN](#)
- [Sending log files](#)
- [Connection: BGP](#)
- [Connection: Secure DNS server](#)

2.2 CONNECTION TO QRATOR LABS

Connection: Bot Protection

This article describes the recommended order of primary setup for the [bot protection](#) service.

1. [Configure the request processing rules](#)
2. [Configure exceptions for trusted users](#)
3. [Test the configuration](#)
4. [Incrementally enable bot protection for all users](#)

**BOT PROTECTION SETUP CHECK LIST**

The complete configuration should meet the following conditions:

- In section [Bot protection](#) → [Locations](#):
 - location rules are set,
- In section [Bot protection](#) → [Visitors](#):
 - **Enabled** switch is active,
 - **A/B checks distribution** slider is set at 100%.

1. Configure the request processing rules

In **Bot protection** → **Locations** section [set up the request processing rules](#) according to the structure of your resource. The rules define how the system should treat each incoming user request depending on its HTTP method, host and path values. A possible variant of the ruleset is listed in the [sample config](#).

2. Configure exceptions for trusted users

In **Bot protection** → **Visitors** section you can specify pre-requisites for requests from the users who should be treated as trusted and allowed to proceed without checks and limitations.

Such exceptions can be defined for the user requests:

- [containing specific HTTP header names and values](#),
- [coming from IP addresses located in specific countries](#),
- [coming from specific IP addresses and CIDRs](#).

On your first use of **Bot protection** → **Visitors** you also need to activate “Enabled” switch.

3. Test the configuration



NOTE

Before testing, make sure that **Bot protection** → **Visitors** section have **Enabled** switch activated, and the IP addresses used to make test requests are added to **Force checks For IPs** list.

You can test the correctness of your ruleset without affecting all visitors by using a limited number of test devices. [Add one or several IP addresses to Force checks For IPs list](#) and send test requests from these addresses. Make sure that your ruleset is letting through legitimate users while preventing automated and scripted requests.

We recommend using web browsers such as Google Chrome and console utilities such as curl for configuration tests.

You can find several configuration testing scenarios below.

Example: test access to web pages

This scenario is used to test web pages to which the access is allowed for browsers but not for bots.

- Use your web browser to open the page. It should load correctly and your browser should receive a cookie named `qrator_jsid`.
- Make several requests to the page using curl in your console. Each time the command should return 401 Unauthorized response. The data about all your attempts (bar the first one) should appear on the [Bot protection](#) → [Dashboard](#).
- Make a request to the page using curl in your console and adding the header names and values listed in [Exclude by headers](#) setting. The command should receive the contents of the page from your server without errors.

If your ruleset works incorrectly:

- make sure that in the [request processing rules](#) you have set **Accept & Inject JS Challenge** actions for the corresponding conditions;
- check whether the IP address used for testing is added to [Force checks For IPs](#) list and is NOT added to [Exclude IPs & Networks](#) list;
- if you have recently changed and saved your configuration, wait several minutes for the changes to take effect.

Example: test access to protected API endpoints

This scenario tests the API locations available only to the users who previously completed the checks on one of your web pages and received a tracking cookie.

- Use your web browser to visit any location which has a rule for it with **Accept & Inject JS Challenge** action. The browser should receive a cookie named `qrator_jsid`. After that make a request to the API endpoint which has a rule for it with **Accept With Cookies Only** action. The API request should be completed successfully.
- Delete the `qrator_jsid` cookie from your browser using developer tools and repeat the request to the API endpoint. You should receive a 403 Forbidden error page. The data about your attempt should appear on the [Bot protection → Dashboard](#).
- Make a request to the API location using curl in your console. The command should return 403 Forbidden response. The data about your attempt should appear on the [Bot protection → Dashboard](#).
- Make a request to the API location curl in your console and adding the header names and values listed in [Exclude by headers](#) setting. The command should receive the contents of the page from your server without errors.

If your ruleset works incorrectly:

- make sure that the web page you are accessing meets the conditions for any of the [traffic processing rules](#) that have **Accept & Inject JS Challenge** action, and the API endpoints you are testing meet the conditions for the rules with **Accept With Cookies Only** action;
- check whether the IP address used for testing is added to [Force checks For IPs](#) list and is NOT added to [Exclude IPs & Networks](#) list;
- if you have recently changed and saved your configuration, wait several minutes for the changes to take effect.

Example: test access to unprotected API endpoints

This scenario is for API locations available to any users including all bots.

- Make a request to the API location using curl in your console. The command should receive the response from your server without errors.

If your ruleset works incorrectly:

make sure that in the [request processing rules](#) you have set **Do Nothing** actions for the corresponding conditions;

- check whether the IP address used for testing is added to [Force checks For IPs](#) list and is NOT added to [Exclude IPs & Networks](#) list;
- if you have recently changed and saved your configuration, wait several minutes for the changes to take effect.

4. Incrementally enable bot protection for all users

It is recommended to enable bot protection for your user base within several stages, incrementally increasing the share of users whose requests will be checked. This approach helps minimize the impact of possible

configuration errors on your user base during deployment.

For step-by-step deployment use [A/B checks distribution setting](#). You can start with 5% users checked and keep increasing this value by 5% more each step after a brief observation interval. Keep in mind that applying a new value to this setting may take 5–7 minutes.

Track the changes on **Traffic** and **Bot protection** graphs in [Statistics](#) section after each increment. Usually the increasing share of checked users leads to the corresponding increase of the number of detected bots and decrease of the user traffic passed to the server. Deviations from this pattern may also happen due to large bot networks being detected as well as possible false-positive blockings of legitimate users.

If you observe drastic changes on your graphs, study the detailed data in [Bot protection → Dashboard](#) section to find out possible causes. In case of false-positive blockings it is recommended to cancel the deployment by setting **A/B checks distribution** back to 0%, fix the errors in [Bot protection → Locations](#) and [Bot protection → Visitors](#) sections and start the incremental deployment again.

In case you do not observe any false-positive blockings you can keep increasing **A/B checks distribution** up to 100%.

2.3 CONNECTION TO QRATOR LABS

Reverse HTTP Proxy

In order for the Qrator Labs reverse HTTP proxy to be able to protect the servers serving your resource, you need to ensure that all incoming traffic destined for the resource is always routed to the reverse proxy.

To do this, you need to:

1. Configure DNS such that traffic to the servers goes through the reverse proxy.
2. Change the IP addresses of the servers in case the servers are already under attack.

If you leave the same IP address that was used before, the attacker will be able to send traffic to the server directly, bypassing the Qrator Labs reverse proxy.

3. Set up a firewall on the servers so that only traffic from trusted networks is allowed.

If the server does not drop connections from unknown networks, the attacker will still be able to detect it by brute forcing IP addresses. This allows traffic to be sent directly to the server, bypassing the Qrator Labs reverse proxy.

4. In order to preserve information about the real IP addresses of resource users, it is necessary to configure [processing of the X-Forwarded-For header](#).

DNS configuration

In the DNS zone settings, change the A record for your domain to the reverse proxy IP address that appears as *Qrator IP* in your dashboard. Note that the changes are not immediately visible to other DNS servers and users. The exact time it takes to update an A record depends on its TTL value.

The Qrator Labs network is ready to handle traffic and implement protection as soon as the traffic starts arriving at the reverse proxy.

Changing the current IP address

**WARNING**

This instruction should be followed if you're already under a DDoS attack.

Sometimes an attacker has an archive of old DNS records for your domain. So even after you change your A record to Qrator IP, an attacker may redirect the traffic to the real server IP, bypassing the reverse proxy.

To protect yourself from such an attack vector, change the upstream IP address:

1. Make sure that the DNS A record for the desired domain points to Qrator IP.
2. Get a new IP address from your ISP that has never been used for your site before. Do not publish the new IP address anywhere or use it in your DNS records. If possible, it's better to get an IP address from a different network than the previous one.
3. In your [Qrator Labs dashboard](#), add the new IP address as an upstream address. Delete the previous upstream, whose IP address may be known to an attacker.
4. Stop processing requests to the former IP address.

Qrator Labs will start using the updated upstream list as soon as you save it in your dashboard.

Firewall configuration

If the IP addresses for the upstreams are taken from the same range, an attacker may try to find the actual server IP address by going through the addresses in the same network that the previous IP address belonged to. This happens if you use addresses from the same own network or from the same ISP.

To protect against detection, configure firewall rules such that the server only responds to requests from Qrator Labs networks and from your trusted networks, and drops all other connections. If none of the servers responds to connection attempts from the attacker, the attacker will not be able to know which address to direct the attack to.

Ensure that the list of allowed subnets from where requests will be processed includes the Qrator Labs subnet: `185.94.108.0/24`.

You can also add to this list a trusted office network and some employees' IP addresses if they plan to make requests bypassing the reverse proxy. Connections from all other addresses must be denied at the firewall level.

Depending on the upstream operating system, you can either use only `iptables` rules for plain HTTP, or combine them with rules for `iptables` supporting `conntrack` and `ipset` modules. We recommend using both `conntrack` and `ipset` because it makes the `iptables` rule set smaller (thereby making it easier to maintain) and faster (fewer rules means fewer requests per incoming packet).

You can make sure that the configuration for ports 80 and 443 is correct in the [Utilities](#) → [Vulnerabilities](#) section.

Here are some examples of scripts that do this configuration for `iptables`.



THE EXAMPLES MAY NOT FIT YOUR CONFIGURATION

The following example scripts may not work or have an undesirable effect on your server. It depends on which `iptables` configuration is currently in use. Be careful not to use the scripts in the examples if you are not sure that they are suitable for your case.

Example of `iptables` configuration for plain HTTP

```
#!/bin/sh

QRATOR_NODE="
185.94.108.0/24
"

ADMIN_IPS="
127.0.0.1
"

iptables -N trusted_nodes
for IP in $QRATOR_NODE $ADMIN_IPS; do
    iptables -A trusted_nodes -s $IP -j RETURN
done

iptables -A trusted_nodes -j DROP
iptables -A INPUT -p tcp -m multiport --dports 80,443 -j trusted_nodes
```

Example of `iptables` configuration with `conntrack` and `ipset` support

```
#!/bin/sh

QRATOR_NODE="
185.94.108.0/24
"

ADMIN_IPS="
127.0.0.1
"

ipset -N trusted_nodes hash:net
for IP in $QRATOR_NODE $ADMIN_IPS; do
    ipset -A trusted_nodes $IP
done

iptables -N qrator
iptables -A qrator -m set --match-set trusted_nodes src -j ACCEPT
iptables -A qrator -j DROP

iptables -I INPUT --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp -m multiport --dports 80,443 --state NEW -j qrator
```

2.4 CONNECTION TO QRATOR LABS

Connection: Ingress filtering

To use [ingress filtering](#) from Qrator Labs, you have to be a local Internet registry (LIR) and have your own autonomous system (AS). Connecting this service is possible for network prefixes of at least class C (/24).

To start using Ingress filtering:

- [Add prefixes.](#)
- [Set up a channel to Qrator Labs.](#)
- [Set up filter policies.](#)

Add prefixes

You can add prefixes in two steps.

First, describe the entire address space of the transit network that you want to protect and confirm the right to manage it. To do this, go to “Prefixes” on the “Qrator” tab of your personal dashboard and specify one or more network prefixes. It often makes sense to list directly the networks you manage, and not to divide them into smaller subnets. For each prefix, enter the number of your AS, the origin of the prefix. The prefix must belong to that AS according to the Internet registrar's database. To confirm the action of adding the prefix, Qrator Labs will send an email with a link to the address specified as [tech-c](#) for the corresponding AS.

Then add the confirmed networks or their subnets to “Prefixes” in the “Ingress” tab of the dashboard. At this point, you can use as small subnets of your networks as you like, for which you will be able to configure filtering policies later.

Setting up a channel to Qrator Labs

The channel between your autonomous system and Qrator Labs can be established in a variety of ways, including tunnels, physical junctions, or cross connects in data centers. The possible connection methods depend on the specific providers and data centers with which your transit network is connected.

The recommended connection method is to connect directly from your transit network to Qrator Labs using a dedicated ISP link or data center crossover. Consult the technical support team to determine the best connection method to protect your network.

Note that Ingress filtering works with inbound traffic, and outbound traffic is not required to be routed through Qrator Labs; see [Differences from endpoint protection](#). In most cases, inbound traffic should be routed through the Qrator Labs network to take advantage of ingress filtering and make outbound traffic bypass the Qrator Labs network to ensure maximum speed for it.

Setting up filtering policies

Filtering policies are sets of limits applied to incoming traffic. Filtering policies are configured through the “Ingress” tab of the dashboard. They are in a “many-to-many” relationship with the prefixes you specify. This gives you the flexibility to customize filtering for your needs.



NOTE

One possible approach to setting up filtering policies is to develop multiple filtering policies suitable for end-user networks with different traffic volumes, and assign the desired policies to the prefixes of each network.

When setting limits, it is recommended that you base your limits on the amount of legitimate traffic that passes through your network at normal times. The best way to express these volumes as a set of policies depends on the nature of the services on your end-user networks, the constancy or seasonality of their traffic, and the threat model. You can consult with Qrator Labs technical support experts to find the optimal ingress filtering policies suitable for your transit network.

2.5 CONNECTION TO QRATOR LABS

Connection: reverse TCP Proxy

This article describes how to connect to Qrator Labs reverse TCP proxy in [Proxy Protocol](#) or [Raw TCP Proxy](#) mode.

For more details on the principles of work and differences between these modes, see [Technology: Reverse TCP proxy](#).

Connection in Proxy Protocol mode

Connecting to Qrator Labs using Proxy Protocol allows you to protect traffic against DDoS attacks at the TCP segment level while preserving sender information.

Required software

A TCP segment with a Proxy Protocol header is different from a regular TCP segment and requires additional processing by the receiving side. This can be handled by any upstream software that knows how to handle this protocol. For example, it can be installed on routers or individual servers.

Some specialized software packages have built-in support for Proxy Protocol. For example, the Dovecot mail server or the MetaTrader trading platform can accept a segment using this protocol without additional tools.

An up-to-date list of software supporting the Proxy Protocol can be found [at HAProxy](#).

Note that there are two versions of the protocol. Proxy Protocol v1 uses the text header format, while Proxy Protocol v2 uses the binary format. The overhead for data transmission and processing when using the binary version of the protocol is lower compared to the text version. Some software versions support only one version of the protocol. For example, Proxy Protocol v1 is supported in NGINX 1.5.12 and later versions, but Proxy Protocol v2 is only supported in NGINX 1.13.11 and later versions.

Some software, such as the MetaTrader 5 trading platform, supports the Proxy Protocol without additional configuration.

For information on how to enable the use of Proxy Protocol in some software, please refer to the relevant documentation, e.g.:

- [Accept-proxy parameter in HAProxy](#)
- [Setting up Proxy Protocol in NGINX](#)

Switching between upstreams

By default, if only one upstream is configured, the reverse TCP proxy tries to send all TCP segments from users to that upstream, and if the connection to the upstream fails within 5 seconds, it reports an error to the user.

If you have set up two or more upstreams, in case of an error when connecting to one of them, the reverse TCP proxy will connect to the next upstream on the list. Note that the same upstream check order will be used for each request. This means that if the main upstream fails for a long time, all users' requests will be handled with a delay of at least 5 seconds. So if you know that one of your upstreams is currently unavailable, contact technical support to temporarily remove it from the list of upstreams being used or change their order.

Also, at your request, Qrator Labs technical support can set up active upstream checking to automatically enable and disable each upstream depending on its availability. Active checking is performed by each node in the Qrator Labs network individually and works as follows:

1. As part of each check, the reverse TCP proxy performs a special diagnostic request about the state of each upstream. This request is made to a special healthcheck service, which can be located either on the upstream itself or on any other URL you specify. The request is made using HTTP or HTTPS to a specific URL on the upstream. If necessary, you can specify the HTTP method (GET by default), HTTP headers (including [Host](#)) and the request body to be sent.
2. If upstream fails to connect within a certain time (by default 5 seconds) or returns a response with a certain status (by default any HTTP 5xx status), this is considered an error. If necessary, the reverse TCP proxy can also check the contents of the request body as well as the upstream certificate when connecting over HTTPS.
3. By default, the reverse TCP proxy is configured such that it immediately disconnects the upstream after an error occurs. This means that user traffic will temporarily not be routed to it.

To protect against false positives, you can configure the reverse TCP proxy to, within a single check, perform several diagnostic requests one after the other immediately or with a slight delay. The reverse TCP proxy will count the number of successful and unsuccessful attempts and will end the check as soon as a certain number of successful attempts or a certain number of unsuccessful attempts have been reached. Upstream will be considered active or inactive depending on which event occurs first.

A full check of all upstreams, including those disconnected, is performed every few seconds. The default interval is 15 seconds, but this can be changed depending on your needs.

Connection setup order

1. In the [Services](#) section of the Qrator Labs personal dashboard, configure a service of the “TCP Proxy” type. Specify the IP addresses and ports of your upstreams that accept the Proxy Protocol traffic.
2. If multiple upstreams are configured, select the necessary [switching parameters](#).
3. If necessary, contact technical support and specify the protocol version that the software on your side supports. By default, Proxy Protocol v2 will be used.
4. If you need to protect against DDoS attacks at the application layer of the OSI/ISO model, make sure that [server logs are transmitted to the Qrator Labs servers](#) automatically for analysis. If this is not done, reverse TCP proxy will not be able to protect against application layer (L7) attacks.
5. Coordinate with technical support on when to enable reverse TCP proxy. Please note that if the upstream at this IP address is currently operating via a reverse HTTP proxy, the site may be temporarily unavailable for about 10 minutes when switching over.

6. At the specified time, enable Proxy Protocol on your side.

Raw TCP Proxy connection

Raw TCP Proxy mode is suitable for cases where it is difficult or impossible to enable Proxy Protocol on the upstream, and information about the sender of TCP segments is not critical for upstream operation.

No additional configuration on the upstream side is required.

2.6 CONNECTION TO QRATOR LABS

Connection: Reverse WebSocket Proxy

The reverse WebSocket proxy allows all WebSocket traffic between the user and the upstream to pass through the Qrator Labs network. For more, see [Technology: Reverse WebSocket proxy](#).

The required [connection configuration order](#) for the reverse WebSocket proxy is described below. You will also find information on setting the [limits](#) and [balancing](#) for this service.

Order of configuration

1. Configure the client side of your application such that WebSocket traffic is routed to a dedicated port.

Qrator Labs uses the same IP address for both HTTP and HTTPS traffic, and for WebSocket traffic. For correct analysis and protection, it is necessary that these two types of traffic come to this IP address on different ports.

**NOTE**

For example, you can configure your application to use port 8080 for WebSocket traffic, while still processing HTTP and HTTPS traffic on standard ports 80 and 443.

The reverse WebSocket proxy responds to any HTTP request other than HTTP [101 Switching Protocols](#) with an HTTP error code [400 Bad Request](#).

2. In the [Services](#) section of the Qrator Labs personal dashboard, configure a service of the “*WebSocket*” type. Specify the IP addresses and ports of your upstreams that accept the WebSocket traffic. They may differ from the upstream IP addresses and ports for [reverse HTTP proxy](#).

When using multiple upstreams at the same time, balancing will be enabled; see [Balancing](#).

3. If necessary, inform the Qrator Labs technical support team of the limits on the number of simultaneous connections from one IP address. Refer to [Limits](#) for more information.
4. Configure the upstream firewall to only allow traffic from trusted networks.

Recommendations for configuring a firewall are given in [the relevant section of the article “Connection: Reverse HTTP Proxy”](#). Note that in the scripts in it you need to replace ports 80 and 443 with the ports on which your upstream accepts requests via WebSocket.

Limits

To make it more difficult to launch DDoS attacks against your application's upstreams, Qrator Labs limits the number of WebSocket connections that users can initiate from a single IP address. When the limits are exhausted, the WebSockets proxy will reject new attempts to establish connections until any of the previously established connections are closed.

The default limits are appropriate for most clients, but can be changed if necessary. For example, if your application needs to access many WebSocket resources for a single user at the same time, inform the technical support team and they will increase the limit accordingly. This will avoid erroneous blocking of legitimate traffic.

If you plan to make frequent requests to the resource from some known IP addresses, provide these IP addresses to technical support to be added to the whitelist. Requests from this list will be subject to increased limits.

**NOTE**

Customers often whitelist their office networks, for example, so that frequent testing of a resource by the customer will not be recognized by the reverse proxy as exceeding the limit.

Balancing

When dealing with multiple upstreams, the reverse WebSocket proxy allocates users among them using the *IP hash* algorithm. This algorithm selects an upstream depending on the request IP address, with repeated connections from the same IP address being established with the same upstream. In practice, this means that all connections initiated by a single user are highly likely to be proxied to the same upstream - which is beneficial, in particular, for more efficient use of the cache on each server.

If an upstream does not respond within 5 seconds when trying to connect, the reverse proxy will attempt to connect to the next upstream in the list. The order of upstreams in the list can be changed by contacting technical support.

A response with HTTP status code `400 Bad Request`, signaling a connection failure, will be sent to the user only after attempts to connect to each upstream in turn fail.

2.7 CONNECTION TO QRATOR LABS

Connection: Tunnels

Tunneling is one way to establish a connection between the Qrator Labs network and your network. This technology allows you to create a virtual connection – *tunnel* – on top of the Internet and pass traffic through it as if the networks are directly connected to each other. Other Qrator Labs technologies and services, such as [reverse HTTP proxies](#) or [BGP connections](#) can be set up to run through the tunnel.

Qrator Labs supports “point-to-point” tunneling using IP/IP or GRE.

If you need to protect against DDoS attacks at the application layer of the OSI/ISO model, make sure that [server logs are transmitted to the Qrator Labs servers](#) automatically for analysis. If this is not done, the Qrator Labs network will not be able to protect against application layer (L7) attacks.

To start using tunneling, take the following three steps:

1. Create a tunnel in your personal dashboard (see “[Tunnels](#)”).
2. Let the tech support know your desired [configuration on the Qrator Labs side](#).
3. [Configure your endpoint](#).

How tunnels work

A tunnel is created between two endpoints connected to the Internet. One is on the Qrator Labs network, the other is on your network.

An endpoint is an entity that routes inbound and outbound tunnel traffic at one end of the tunnel. For example, an endpoint could be a router, a firewall, a virtual machine with Linux as the router, a server directly hosting your resources, and so on.

**NOTE**

An endpoint for the tunnel is created in each of the traffic scrubbing centers in the Qrator Labs network. They all use the same configuration and the same addresses, making them interchangeable. From the customer's perspective, all the endpoints in the Qrator Labs network can be considered as a single endpoint with high fault tolerance.

Two pairs of IP addresses are used when setting up a tunnel: *external* and *internal*.

- Internal addresses are the addresses assigned to the endpoints within a virtual network. For example, an application on your server can send IP packets to an internal address on the Qrator Labs network as if that address were on the same network as the server.

If the endpoint on your side does not support internal IP addresses when configuring “point-to-point” tunnels, you can specify the name of the network interface. See [Customer-side configuration](#).

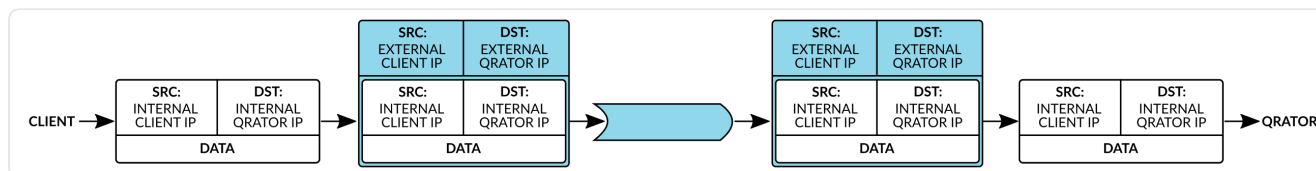
Internal addresses are generated automatically when [adding a tunnel in your personal dashboard](#).

- External addresses are real addresses where endpoints are available on the Internet. So, your endpoint packages an IP packet sent to a Qrator Labs internal IP address into another packet sent to a Qrator Labs external IP address.

The external Qrator IP is the IP address corresponding to the domain for which the tunnel is configured. The customer's external IP address is specified by the customer itself.

All intermediate nodes through which a packet passes in the external network check only the external IP addresses and route the packet according to routing protocols. When a packet arrives at the external IP address of the Qrator Labs-side tunnel, the endpoint discards the external headers and then works with the packet that was originally sent by the application.

So, from the point of view of the customer-side application, the packet is transmitted directly to the Qrator Labs network unchanged, regardless of which route was actually used to reach Qrator Labs.



Packets in the opposite direction are forwarded in the same way.

Configuration on the Qrator Labs side

Port forwarding

To protect the customer against unwanted traffic, the Qrator Labs network passes packets through the tunnel only to ports that the customer has explicitly designated as *open*. For example, web services usually require that only ports 80 and 443 be kept open.

By default, no ports are considered open. To open a port, [create a service of the "TCP" type](#) in the personal dashboard.

Tunnel redundancy

The Qrator Labs network consists of many traffic scrubbing centers in different regions, but by default they all use the same tunnel to communicate with your endpoint. This tunnel is marked as *active tunnel* in the ["Tunnels" section of your personal dashboard](#). You can choose which tunnel you want to make active. The network will not attempt to send traffic to inactive tunnels.

In order to provide redundancy in case of failure and to take full advantage of the Qrator Labs distributed network, it is recommended to set up multiple tunnels to different points and enable automatic switching between them.

Automatic switching works as follows:

1. Once every 120 seconds, each traffic scrubbing center sends a diagnostic request to the internal customer-side tunnel address to check the availability of each tunnel. The request can be made over ICMP, HTTP or HTTPS. Request parameters and response timeout (in milliseconds) are configurable in the dashboard.
2. If a traffic scrubbing center fails to receive a response several times in a row, the tunnel is considered temporarily unavailable, and traffic is directed to the next available tunnel according to the list from the dashboard. Tunnel availability statuses are tracked by each traffic scrubbing center independently of other centers.
3. When the check is successful again, the traffic scrubbing center considers the tunnel available.

So, even if only one tunnel is consistently available from each region, the Qrator Labs network will continue to deliver traffic from any user to your network.

Changing internal IP address

Qrator Labs assigns the same internal IP address to all the tunnels of the same customer. This allows you to use the same configuration on all endpoints on your side. There will be traffic balancing between multiple available tunnels with the same internal IP address.



NOTE

At your request, Qrator Labs technical support can change the internal IP address of the tunnels.

Customer-side configuration

The endpoint at your end of the tunnel can either be the upstream itself or a router serving several of your servers.

In general, setup boils down to the following steps:

1. Create a new tunnel in the system using the IP addresses you have generated in your Qrator Labs dashboard. The system will start accepting incoming packets from this tunnel automatically.
2. Configure the endpoint such that for a packet received through the tunnel, the response packet is also routed to the tunnel. To do this, you can configure a separate routing table (e.g. with VRF), set up policy-based routing (PBR) or use a combination of these methods.

If it is not possible to configure internal IP addresses for the tunnel, use the name of the tunnel interface when configuring traffic routing to the tunnel. This is possible because “point-to-point” tunnels are configured.

Please note that without a proper VRF or PBR setup, your endpoint will not be able to send outbound packets through the tunnel and will only receive inbound traffic. This would make the resource not work correctly. The Qrator Labs tech support team will help you make the proper settings.

The Windows operating system does not support VRF and PBR configuration. If your server is running on Windows, you should connect it to the Internet not directly but through a Linux proxy server or a router.

To avoid packet fragmentation when routing, it is recommended to configure the MTU (maximum transmission unit) and MSS (maximum segment size) taking into account the selected tunneling protocol:

- For IPIP tunnels, set the MTU to be no more than 1480 bytes, MSS to be no more than 1440 bytes.
- For GRE tunnels, set the MTU to be no more than 1476 bytes, MSS to be no more than 1436 bytes.

If inside a tunnel there will be traffic that is also one or more tunnels, the MTU and MSS values must be adjusted accordingly.

2.8 CONNECTION TO QRATOR LABS

Connection: Qrator.CDN

Right in their Qrator Labs personal dashboard, customers can connect and configure the Qrator.CDN service.

Domain and certificate

To use Qrator.CDN, you need to set up your site such that a separate domain could be used for static content. User requests to this domain will be handled by Qrator.CDN, downloading it from your upstream and updating it as needed.

Qrator.CDN automatically creates for each site a domain that can be used for these purposes. However, customers often want static content to be hosted on their own domain, for example, a certain subdomain of the main domain. In this case, a CNAME record must be created in the appropriate DNS zone.

When accessing the domain via HTTPS, the certificate specified by the customer will be used. You can use any certificate pre-loaded in the *certificate storage* in your personal dashboard.

See the dashboard manual: [Add an HTTP host](#), [Add an HTTPS host](#).

Header settings

Although CDN does not modify the body of the content it caches (except for the optional image processing, see below), it is often necessary for CDN to add some sort of HTTP header to a user request or upstream response. Qrator.CDN can add arbitrary headers that are described in your dashboard.

Also, Qrator.CDN can specify the user's source IP address in a special header, whose name you can configure in your dashboard.

See the dashboard manual: [Change HTTP header settings](#).

Blocked and white URL lists

The customer can set up a list of blocked URLs to which the CDN will respond with an error code without requesting content from the upstream. The specific error code is configured by the customer individually for each URL.

The customer can also create a list of “white” URLs, if necessary. In this case, CDN will request content only from URLs on that list and respond with an error code to all others. The default error code is `404 Not Found`, but it can be overridden on a per-page basis using a blocked URLs list.

See the dashboard manual: [Set Request URL policy](#).

Caching settings

Efficient caching is the main task of CDN, and it is important that it is done correctly, taking into account the content structure on the site. Therefore, the customer can configure how CDN will handle HTTP header `Cache-Control` in the dashboard, as well as enable or disable caching of certain types of errors. The customer can also enable CDN to ignore all GET request parameters. For example, this is handy if you rely on randomly generated GET parameters to force the browser to update cached content, but don't want that to result in extra requests from the CDN to upstream.

In some cases, it may be useful to remove some files from the cache after updating upstream content or making significant changes to caching settings. This can be done through your dashboard.

You can ask the Qrator Labs tech support team to force cache and update some content on the CDN (to perform the so-called CDN “warmup”) without waiting for the content to be first requested by users. This is useful if you have a lot of content updated on the source server. Without pre-caching, it may take longer for first time users to download new content because the CDN will not yet have cached content and the cache will fill up slowly.

See the dashboard manual: [Change cache settings](#).

Processing images

For JPEG, PNG, TIFF, WebP and AVIF images (the types are detected based on the `Content-Type` header), Qrator.CDN can perform simple operations according to GET parameters in the user's request.

- `crop` — crop the image.

The GET parameter value must have the format `x{NUMBER},y{NUMBER},w{NUMBER},h{NUMBER}`. The operation returns the contents of a rectangle that begins at the point `(x,y)` from the top left corner and has the size `w×h`.

The example below instructs Qrator.CDN to load `image.png` from the upstream and return its fragment of the size 500×200 pixels, skipping 10 pixels on the left and 15 pixels on the top.

```
https://cdn.example.com/image.png?crop=x10,y15,w500,h200
```

- `resize` — scale the image.

The GET parameter value must have the format `w{NUMBER}` for scaling to the specified width or `h{NUMBER}` for scaling to the specified height. The aspect ratio is kept unchanged when scaling.

If the `resize` operation is used together with `crop`, the scaling is applied to the results of the `crop` operation.

The example below instructs Qrator.CDN to load `image.png` from the upstream and return its copy, reduced or enlarged to the width of 300 pixels.

```
https://cdn.example.com/image.png?resize=w300
```

- `format` — convert the image to a given format.

The GET parameter value can be one of the strings: `jpeg`, `png`, `webp`, `avif`. An optional parameter `q{NUMBER}` (values from 1 to 100 are allowed) specified to the converter how close should the image quality be to the original. The real quality can vary depending on the selected format.

The example below instructs Qrator.CDN to load `image.png` from the upstream and return its copy in the JPEG format with the quality level of 75.

```
https://cdn.example.com/image.png?format=jpeg,q75
```

Note that when given invalid or incomplete set of parameters for an operation, Qrator.CDN will return the original image without modifications.

When using the `crop` or `resize` operation without the `format` operation, Qrator.CDN will return the image as JPEG or WebP (if [Converting images to WebP](#) is enabled).

Converting images to WebP

You can use auto-conversion of images to [WebP](#) to speed up the site. The WebP format provides a smaller file size with comparable image quality.

When auto-conversion is enabled, Qrator.CDN checks the `Accept` header in the request to determine if the user's browser supports WebP. If it does, the user is sent a WebP version of each image. The original versions of images remain in the Qrator.CDN cache. If the user's browser version does not support WebP, Qrator.CDN sends the original version of the image unchanged.

See the dashboard manual: [Configure conversion of the images to WebP](#).

Transport settings

By default, Qrator.CDN serves the static content using the HTTP/1.1 protocol only. You can enable [HTTP/2](#) to provide more efficient traffic through header compression, request multiplexing and some other techniques.

For both HTTP/1.1 and HTTP/2, you can choose the stream compression format to use. Qrator.CDN supports `gzip`, `deflate`, `br` and uncompressed streaming formats. You can switch between them in your dashboard.

See the dashboard manual: [Change transport settings](#).

2.9 CONNECTION TO QRATOR LABS

Sending log files

Sending web server logs to Qrator Labs servers allows you to analyze and clean up HTTP traffic going to these web servers at the application layer (L7 under the OSI/ISO model) in situations where it is not technically possible to analyze HTTP traffic itself.

For instance, it is recommended to configure log sending if you're using [reverse TCP proxy Qrator Labs](#). This proxy analyzes traffic at the transport layer (L4), but, with the help of log analysis, can also protect against some application layer (L7) attacks.

To transmit logs to the Qrator Labs network:

1. [Choose the IP addresses and ports for log sending](#)
2. [Configure the log format.](#)
3. [Configure log collection and sending.](#)

**WARNING**

For analysis, you need the logs of the web server on your network that receives external requests first, regardless of where it routes or proxies them later. In this case, only access logs are needed. It is not necessary to send error logs.

1. Choose the IP addresses and ports for log sending

This step should be done by you and Qrator Labs technical support team. On your side, you need to set which IP address the logs will be sent from. On the Qrator Labs side, the IP address and port of the log listener will be assigned. The assigned IP address may be different from Qrator IP specified in your dashboard.

2. Configure the log format

Configure the server such that access logs are written in a certain format from which the Qrator Labs reverse TCP proxy can extract all the information needed for analysis. Log format requirements may change in the future, in which case Qrator Labs tech support specialists will inform you about the changes beforehand.

Below are the current log formats for popular web servers.

NGINX

Specify the following line in the [log_format](#) option.

```
'2xXXXXXX|:,$bytes_sent|:,$http_accept|:,$http_accept_encoding|:,$http_accept_language|:,$http_cookie|:,$http_host|:,$http_referer|:,$http_user_agent|:,$http_x_forwarded_for|:,$remote_addr|:,$request|:,$request_time|:,$status|:,$time_local|:,$upstream_response_time|:,$http_x_requested_with|:,$http_authorization|:,$remote_port|:,$-|:,$msec|:,$-|'
```

Apache

Specify the following line in [CustomLog](#).

```
'5xXXXXXX|:,$%O|:,$%{Accept}i|:,$%{Accept-Encoding}i|:,$%{Accept-Language}i|:,$%{Cookie}i|:,$%{Host}i|:,$%{Referer}i|:,$%{User-Agent}i|:,$%{X-Forwarded-For}i|:,$%a|:,$%r|:,$%{ms}T|:,$%s|:,$%t|:,$%D|:,$%{X-Requested-With}i|:,$%{Authorization}i|:,$%{remote}p|:,$%{sec}t.%{msec_frac}t|:,$-|:,$-|'
```

F5 BIG-IP

Specify the following line when configuring request logging according to the [Configuring Request Logging](#) guide.

```
'3xXXXXXX|:,$RESPONSE_SIZE|:,$Accept|:,${Accept-Encoding}|:,${Accept-Language}|:,$Cookie|:,$Host|:,$Referer|:,${User-Agent}|:,${X-Forwarded-For}|:,$CLIENT_IP|:,$CLIENT_PORT|:,$HTTP_REQUEST|:,$RESPONSE_MSECS|:,$HTTP_STATCODE|:,$DATE_YYYY/$DATE_MM/$DATE_DD_$TIME_HH24:$TIME_MM:$TIME_SS|:,$HTTP_KEEPALIVE|:,$SERVER_IP|:,$SERVER_PORT|:,$BIGIP_CACHED|:,${X-Requested-With}|:,${Authorization}|:,$CLIENT_PORT|:,$TIME_MSECS|:,$-|:,$-|'
```

HAProxy

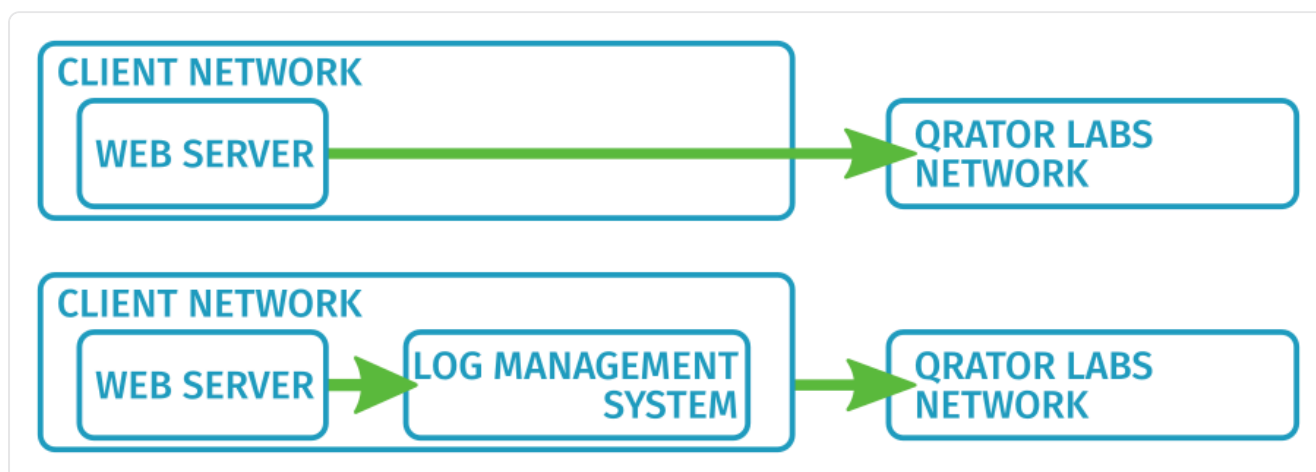
Insert the following snippet into configuration file HAProxy.

```
capture request header Accept len 4096
capture request header Accept-Encoding len 4096
capture request header Accept-Language len 4096
capture request header Cookie len 16384
capture request header Host len 4096
capture request header Referer len 4096
capture request header User-Agent len 4096
capture request header X-Forwarded-For len 4096
capture request header X-Requested-With len 4096
capture request header Authorization len 4096
log-format "4xXXXXXX|:,$%B|:,$%[capture.req.hdr(0)]|:,$%[capture.req.hdr(1)]|:,$%[capture.req.hdr(2)]|:,$%[capture.req.hdr(3)]|:,$%[capture.req.hdr(4)]|:,$%[capture.req.hdr(5)]|:,$%[capture.req.hdr(6)]|:,$%[capture.req.hdr(7)]|:,$%ci|:,$%r|:,$%Ta|:,$%ST|:,$%Tl|:,$%Tr|:,$%[capture.req.hdr(8)]|:,$%[capture.req.hdr(9)]|:,$%cp|:,$%Ts.%ms|:,$-|:,$-|"
```

3. Configure log collection and sending

Depending on the web server you use, you may have different ways of sending logs to Qrator Labs. The easiest way is to [send logs directly](#) from the web server if it supports such. A more universal solution is to [use a log management system](#) for any web server.

Note that the log listener on the Qrator Labs side accepts logs via syslog over the TCP transport protocol. Some web servers only support syslog over UDP or over Unix sockets. In that case, it is recommended to collect them using a logging system that would redirect the logs to Qrator Labs.



Send logs directly

If the web server handling user requests is capable of sending logs to Qrator Labs via the syslog protocol itself, then you only need to configure that web server accordingly. For example, instructions on how to send logs directly from F5 BIG-IP can be found here: [Configuring BIG-IP to log to a remote syslog server](#).

To use the secure TLS to send logs, even if the web server itself does not support this protocol, you can use the optional Stunnel as described here: [SSL Encrypting Syslog with Stunnel](#).

Using log management systems

A logging system is software that can act as an intermediate host between your web server and Qrator Labs log receiver. Examples of such systems for Unix-like operating systems are [rsyslog](#) and [syslog-ng](#); for Windows – [Snare](#). When selecting a logging system, note that Qrator Labs accepts syslog over TCP transport protocol, not UDP.



NOTE

Popular logging systems also support encryption of sent logs using TLS, for example:

- [TLS documentation for syslog-ng](#)
- [TLS documentation for rsyslog](#)

Interaction between the web server and the logging system can be configured in a variety of ways, including reading logs from a text file, over a Unix socket, or over TCP or UDP. The method you choose depends on whether the web server and the logging system are on the same server or on different servers.

You also usually need to configure the web server itself in order to send logs to the right file, socket, or address. Read more in the documentation for the particular web server you are using, e.g:

- [Syslog logging documentation for NGINX](#)
- [Syslog logging documentation for lighttpd](#)



WARNING

When using logging system rsyslog, you must additionally configure the action queues and their discard rules. You can read more here: [Action Queue cfg discarding local messages](#).

rsyslog configuration examples

Below are examples of rsyslog configuration files for HAProxy and NGINX. The required configuration file should be placed in the `/etc/rsyslog.d/` directory and then rsyslog should be restarted. To use TLS, it is also required to substitute the path to [Qrator Labs certificate](#) in the configuration file.

HAProxy without TLS

```
$AddUnixListenSocket /var/lib/haproxy/dev/log
if ($programname startswith 'haproxy') then {
    if ($msg contains '4xFFFFFF') then {
        $template codename-fwd, "<%PRI%>1 %timegenerated:::date-rfc3339% codename
%syslogtag:1:32% - - %msg:::drop-last-lf%\n"
        action(type="omfwd"
            Target="<receiver IP address>"
            Port="<receiver port>"
            protocol="tcp"
            queue.filename="codename-fwd-q1"
            queue.discardSeverity="4"
            queue.discardMark="25000"
            queue.size="30000"
            queue.highWatermark="10000"
            queue.lowWatermark="0"
            queue.timeoutEnqueue="500"
            queue.maxDiskSpace="20m"
            queue.timeoutWorkerthreadShutdown="-1"
            queue.saveOnShutdown="on"
            queue.type="LinkedList"
            Action.ResumeRetryCount="-1"
            template="codename-fwd"
        )
    }
}
```

HAProxy with TLS

```

$AddUnixListenSocket /var/lib/haproxy/dev/log
global(
    DefaultNetstreamDriverCAFile="<path to the receiver certificate file>"
)
if ($programname startswith 'haproxy') then {
    if ($msg contains '4xXXXXXX') then {
        $template codename-fwd, "<%PRI%>1 %timegenerated:::date-rfc3339% codename
%syslogtag:1:32% - - %msg:::drop-last-1f%\n"
        action(type="omfwd"
            Target="<receiver IP address>"
            Port="<receiver port>"
            protocol="tcp"
            queue.filename="codename-fwd-q1"
            queue.discardSeverity="4"
            queue.discardMark="25000"
            queue.size="30000"
            queue.highWatermark="10000"
            queue.lowWatermark="0"
            queue.timeoutEnqueue="500"
            queue.maxDiskSpace="20m"
            queue.timeoutWorkerthreadShutdown="-1"
            queue.saveOnShutdown="on"
            queue.type="LinkedList"
            Action.ResumeRetryCount="-1"
            template="codename-fwd"
        )
    }
}
}

```

NGINX without TLS

```

input(type="imuxsock"
      Socket="/var/log/nginx/nginx.sock")

template(name="codename-fwd" type="string"
         string="<%PRI%>1 %timegenerated:::date-rfc3339% codename %syslogtag:1:32% - -
%msg:::drop-last-lf%\n"
)
if ($programname startswith 'nginx') then {
  if ($msg contains '2FFFFFFF') then {
    action(type="omfwd"
          Target="<receiver IP address>"
          Port="<receiver port>"
          protocol="tcp"
          StreamDriver="gtls"
          StreamDriverMode="1"
          StreamDriverAuthMode="anon"
          queue.filename="codename-fwd-q1"
          queue.discardSeverity="4"
          queue.discardMark="30000"
          queue.size="30000"
          queue.highWatermark="10000"
          queue.lowWatermark="0"
          queue.timeoutEnqueue="500"
          queue.maxDiskSpace="20m"
          queue.timeoutWorkerthreadShutdown="-1"
          queue.saveOnShutdown="on"
          queue.type="LinkedList"
          Action.ResumeRetryCount="-1"
          template="codename-fwd"
    )
    & stop
  }
}

```

NGINX with TLS

```

input(type="imuxsock"
      Socket="/var/log/nginx/nginx.sock")

template(name="codename-fwd" type="string"
         string="<%PRI%>1 %timegenerated:::date-rfc3339% codename %syslogtag:1:32% - -
%msg:::drop-last-lf%\n"
        )
global(
    DefaultNetstreamDriverCAFile="<path to the receiver certificate file>"
)
if ($programname startswith 'nginx') then {
    if ($msg contains '2FFFFFFF') then {
        action(type="omfwd"
              Target="<receiver IP address>"
              Port="<receiver port>"
              protocol="tcp"
              StreamDriver="gtls"
              StreamDriverMode="1"
              StreamDriverAuthMode="anon"
              queue.filename="codename-fwd-q1"
              queue.discardSeverity="4"
              queue.discardMark="30000"
              queue.size="30000"
              queue.highWatermark="10000"
              queue.lowWatermark="0"
              queue.timeoutEnqueue="500"
              queue.maxDiskSpace="20m"
              queue.timeoutWorkerthreadShutdown="-1"
              queue.saveOnShutdown="on"
              queue.type="LinkedList"
              Action.ResumeRetryCount="-1"
              template="codename-fwd"
            )
        & stop
    }
}

```

Qrator Labs certificates

To send data over the TLS transport protocol, you must use a Qrator Labs log receiver certificate.

Different receivers on the Qrator Labs side use different certificates. Contact Qrator Labs tech support to find out which receiver you need to send the logs to. You can also check the authenticity of the CA (certificate authority) that signed these certificates.

Save the required certificate to a file, e.g. `/etc/rsyslog.d/codename-syslog-ca.crt`, and specify this in the logging system configuration path. Configuration files for rsyslog can be found at [rsyslog configuration examples](#).

✓ Certificate for 178.248.239.29

-----BEGIN CERTIFICATE-----

```
MIIFoDCCA4igAwIBAgIU5m2ENhIrrqiVbcdGikZvsaic1ChcwDQYJKoZIhvcNAQEL
BQAwUTEZMBcGA1UEAwQUXJhdG9yIFN5c2xvZyBDQTEUMBGA1UECgwLUXJhdG9y
IExhYnMxHjAcBgkqhkiG9w0BCQEWOD21haWwAcXJhdG9yLm5ldAeFw0yNjA3MDgy
MDQzNDJaFw0zMTA3MDcyMDQzNDJaME8xFDASBgNVBAoMClFyYXRvcjBMWjZMRcw
FQYDVQQDDA4xNzguMjQ4LjIzOS4yOTFEMBwGCsqGSIb3DQEJARYPbWpEbXcmF0
b3IubmV0MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAusYLR0DkJPkf
OhZzaLupR+y01L5SxDka8tW89Mf8mYkpmLPVd42B/8Zw4aqR+YYmsvob8sJYlhv6
zPl1dRocPZ3qXkbvVrj+IrWwZKRvViIN7RS/Ducw+PcDxWA+3Gq7YF02z8nxcn2
YqfIUcsitjQAFxx/UNpCvk3w1gQUSsDh2rPvZZNy1DlmpKy0ArKA8r3qaLqY55vk
qq37cHDSf+ycZxQ4cj8Yxj+yltX7IX7n0cE/s2esb/fzeqB0fVxugZe3Cqumkq+
oI3m0NJsT/F35hu4eFlqK/yRvmwsFqgFBozZ2FuyxStWocYdKyYYIe30X1NQSS2y
1f1FRPwae6MAydn0tHmBRCWdrFK38y7AdmztcFN68qF9iKG5WhOk3sAg39Korlt/
86Ki3Eegml4IkKPx17b0w+fCVIJCN7y2eMMULZVDabJk1Bq0nwZ1f+UEmYXBJ7L
Ys5rr71hpTcjYe2QFjZH6fIRbMwLW5tLY310R1iy9ybDZmirbQVfEXSdokDBZjm
W5UhrnWip00XFek74WffnbGa0mbDUL4iFvfzR1q00/YTB5LK3RzXZesl0kSk3InK
eXhCSTjeGLisxiSSGRlZqcJhA4tHywr0poIZvMeZHoejsTgVVAXhaCSX0NbWj88u
Qbrkl4FqzLJdtaAt3KhCtPfwaZi9QDMCAwEAAANyMHAwHQYDVR00BBYEFpWE0qob
GzPswrrg22lwhjsAtqiHMB8GA1UdIwQYMBaAFGyR/xVNshdETvLwhPDkBYIAEBao
MAkGA1UdEwQCMAAwDgYDVR0PAQH/BAQDAgXgMBMGA1UdJQQMMAoGCCsGAQUFBwMB
MA0GCSqGSIb3DQEBcWUAA4ICAQBskcJw8WrHFFzQ5KFxiyDmxQGFUDGGsq3l4VLE
qIReEW2uLnFaCxvBCzaW40IQ6VqBg5z0T4b1MxA+P7s3VVka6c13epUa6yt0Pl+U
N+0uIdmhpzchEUTNf8VCBHYWhZGJMRa3Bpa7i2PnFAu1eC9kCa0vIvdxajcVn2EA
sSt/P+VBcS9zduY1+woeflwyT896EZ0/Td+WDqG6I81U1z5684tBZKXao9rffioV
WdKUFXd0f1ECilZdj5ADFY7Znu6o0gk3X59Lo43Yp37tfJiIniKSxc1N2LPghU8z
rkxUHa3MK/6nbk/anF6PwAIE2Rm5oCaoK5xNuQU87u903mEIKQS3Ps/x1Qdu+JJZ
/nD7lIM9QFABjpoFcBWR/209c4x275CL8L9yzLxsqJQKX+zpAguXrNUiVPvTI5v
9YsuCrrcdwK90lgrRXFerF14jgAdKIDYk2p0cBHfPyN1m5GQKotp2JwHwCR+sN5G
5v1VVomK5YdE7HXyM7XKBBcDyTqjMcdJVQ0+QAx3mWty23S4ckuwGk5NI5W30LE
/xZhpj0lMtrAoNogDIYu4z31+YNJe1M0Xu5A/f/Tt2MIhVlaPi0Vb43vaEu4mFaT
vm12e9/qPuX3CyBbwLGJBy3DmP5iaCxuBCnNKmawce0sRuFmGgotFlWY2uzP3nKr
G0+f3g==
```

-----END CERTIFICATE-----

✓ Certificate for 178.248.239.30

-----BEGIN CERTIFICATE-----

```
MIIFoDCCA4igAwIBAgIUH3a0xgrPp8ReTBS0xgwMiDo+RzYwDQYJKoZIhvcNAQEL
BQAwUTEZMBcGA1UEAwQUXJhdG9yIFN5c2xvZyBDQTEUMBGA1UECgwLUXJhdG9y
IExhYnMxHjAcBgkqhkiG9w0BCQEWDD21haWwAcXJhdG9yLm5ldAeFw0yNjA3MDgy
MDQzNDRaFw0zMTA3MDcyMDQzNDRaME8xFDASBgNVBAoMClFyYXRvcjBMWjZMRcw
FQYDVQQDDA4xNzguMjQ4LjIzOS4zMDEeMBwGCsqGSIb3DQEJARYPbWFPbEBxcMF0
b3IubmV0MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAusYLR0DkJPkf
OhZzaLupR+y01L5SxDka8tW89Mf8mYkpmLPVd42B/8Zw4aqR+YYmsvob8sJYlhv6
zPl1dRocPZ3qXkbvVRj+IrWwZKRvViIN7RS/Ducw+PcDxWA+3Gq7YF02z8nxcn2
YqfIUcsitjQAFxx/UNpCvk3w1gQUSsDh2rPvZZNy1DlmpKy0ArKA8r3qaLqY55vk
qq37cHDSf+ycZxQ4cj8Yxj+yltX7XIx7n0cE/s2esb/fzeqB0fVxugZe3Cqumkq+
oI3m0NJsT/F35hu4eFlqK/yRvmwsFqgFBozZ2FuyxStWocYdKyYYIe30X1NQSS2y
1f1FRPwae6MAydn0tHmBRCWdrFK38y7AdmztcFN68qF9iKG5WhOk3sAg39Korlt/
86Ki3Eegml4IkKPx17b0W+fCVIJCN7y2eMMULZVDabJk1Bq0nwZ1f+UEmYXBJ7L
Ys5rr71hpTcjYe2QFjZH6FtIRbMwLW5tLY310R1iy9ybDZmirbQVfEXSdokDBZjm
W5UhrnWip00XFek74WffnbGa0mbDUL4iFvfzR1q00/YTB5LK3RzXZesl0kSk3InK
eXhCSTjeGLisxiSSGRlZqcJhA4tHywr0poIZvMeZHoejsTgVVAXhaCSX0NbWj88u
Qbrkl4FqzLJdtaAt3KhCtPfwaZi9QDMCAwEAAANyMHAwHQYDVIR00BBYEFpWE0qob
GzPswrrg22lwhjsAtqiHMB8GA1UdIwQYMBaAFGyR/xVNsHdETvLwhPDkBYlAEBAo
MAKGA1UdEwQCMAAwDgYDVROPAQH/BAQDAgXgMBMGA1UdJQQMMAoGCCsGAQUFBwMB
MA0GCSqGSIb3DQEBwUAAICAQBpf//Kfr6Ya1sNV0aQWB9a3M0lqZ2hmpf0SPi3
AddwWpmXLa50DRpDR6TeV88TPyhyj9AWaA7KXZCdK1a798o4FNYwZRieNN4Q0ugk
gUekvTy9fSjKgAhMSeX0LgVMi5DaKC71mdGDAD1SfBp2irC6ZT6tP0mQHCJ/iqJg
Hmd1KG2+mtwWnhAHLGYLm/yUUhdzToDDUjS2kwVJuFB9uIFP1xZiQz6mZUdrI4ZA
ao2Umdm8u5NJ4Xfy52JrcsPsAK4nptpys8YmoQd2FBfYLS8kkz1S9tziCDejB/3d
CeBAK6RPpzHeVG3c7l3yUAgmLpgmn5diXw6Fyxq7pbTqknI3d6h0TrnAYJGgdfvb
l8N8TBoGjr7Uw4yFbefp4pEcyFD+UalakRoSk4Eb0c6DhsYpGRSST/4wbabbdudo
PbeMMS0vjtihpahz7lWKpGZ0VdDdXUCi/voSQL2mg//R90AS3kCKFYAlV01yaxL6
vv9Cc7uT2Hbv7zMWHDjZS0M3fsUh9/Vd4ygy+D3NoUsmptpsQJy/ffFgaLZWw6e
Gu8C17PfYn6vFD9vP90YYKY8LoghbeMQpXPUx0BGYzzQoLBtusIUVUblRcQDfYvC
QnL8tT739AFEE5oRtjGzaETpKJEQt0xltI3LFh1wvm1CMQo//s0Gx2lXDFBt1gXS
XivWYw==
```

-----END CERTIFICATE-----

✓ **Certificate for 185.94.108.254**

```
-----BEGIN CERTIFICATE-----
MIIFoDCCA4igAwIBAgIUcPorCucj1+yBchknNH0qa76Ax5EwDQYJKoZIhvcNAQEL
BQAwUTEZMBcGA1UEAwQUXJhdG9yIFN5c2xvZyBDQTEUMBGA1UECgwLUXJhdG9y
IExhYnMxHjAcBgkqhkiG9w0BCQEWDD21haWwAcXJhdG9yLm5ldAeFw0yNjA3MDgy
MDQzNDVaFw0zMTA3MDcyMDQzNDVaME8xFDASBgNVBAoMClFyYXRvcjBMWjZMRcw
FQYDVQQDDA4xODUuOTQuMTA4LjI1NDEeMBwGCSqGSIb3DQEJARYPbWFPbEBxcMF0
b3IubmV0MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAusYLR0DkJPkf
OhZzaLupR+y01L5SxDka8tW89Mf8mYkpmLPVd42B/8Zw4aqR+YYmsvob8sJYlhv6
zPl1dRocPZ3qXkbvVRj+IrWwZKRvViIN7RS/Ducw+PcDxWA+3Gq7YF02z8nxcn2
YqfIUcsitjQAFxx/UNpCvk3w1gQUSsDh2rPvZZNy1DlmpKy0ArKA8r3qaLqY55vk
qq37cHDSf+ycZxQ4cj8Yxj+yltX7XIx7n0cE/s2esb/fzeqB0fVxugZe3Cqumkq+
oI3m0NJsT/F35hu4eFlqK/yRvmwsFqgFBozZ2FuyxStWocYdKyYYIe30X1NQSS2y
1f1FRPwae6MAydn0tHmBRCWdrFK38y7AdmztcFN68qF9iKG5WhOk3sAg39Korlt/
86Ki3Eegml4IkKPx17b0w+fCVIJCN7y2eMMULZVDabJk1Bq0nwZ1f+UEmYXBJ7L
Ys5rr71hpTcjYe2QFjZH6FtIRbMwLW5tLY310R1iy9ybdZmirbQVfEXSdokDBZjm
W5UhrnWip00XFek74WffnbGa0mbDUL4iFvfzR1q00/YTB5LK3RzXZesl0kSk3InK
eXhCSTjeGLisxiSSGRlZqcJhA4tHywr0poIZvMeZHoejsTgvvAxhaCSX0NbWj88u
Qbrkl4FqzLJdtaAt3KhCtPfwaZi9QDMCAwEAANyMHAwHQYDVIR00BBYEFpWE0qob
GzPswrrg22lwhjsAtqiHMB8GA1UdIwQYMBaAFGyR/xVNsHdETvLwhPDkBYlAEBAo
MAKGA1UdEwQCMAAwDgYDVROPAQH/BAQDAgXgMBMGA1UdJQQMMAoGCCSGAQUBwMB
MA0GCSqGSIb3DQEBGwUAA4ICAQB9LYRBwR2uLybwHEiHJ5vNPAv65ZUxZKIaqN5T
VYSfr3R3zpZRzcUYLSicCe6YoTZqavaIAxsmqxDEsK/C9JLOTdw3WdLhbOCxiqzT
7rmx/X300GuM9BdGmg1vti9s7J67Mj+U9q+t9qrQnPmDPQYom9sWnTZ7ft6lwiGy
+QXJAW+I9XrxScU0K0BemdCYTCFHPBEJanYTtBbA+0jtCRw8WZ0cRtmJYiuni1IT
dk/MAHvKppGr0d8vxsXT98TpWqLyyYpdQC7eprIwfbsl+cix6LzN7kw5BoMNd+N
2Y5J9EJNSnDSh1FTkuU2tg4Kzem+15rZzAUBorY+8pQDjlepMqY9e6IC9kthKmf4
Okm5pFX40LkqK2nTLpqF4H9Ias6Io16R0pJ8Gd437MnnE3X+ZqNpnu9fCiajUwjF
4hBCPHBVLHK0fC81s1vZkzggcXq0zY4SVMoFaEbcjRt1UrdRLBik9PldpvJLxAdQ
/LlGEjf9E0Tr6Rrb5qGovKmQGnyujrTMoWeV7AegODFv00SZk3oukmJnLn4BF4E
Kbbqa5tmwJlB+FtUGdRadH8Jk8ur1CwF3TL62uAjbwfdq7yU6dp6g/Vl7tEq7d1C
SolMjcfWnUkFu9f8pcgD6SKNxNRRwlnyQUNfKoat/n4MwnYuN2hJT+jcj4ILf6M
EgK5VA==
-----END CERTIFICATE-----
```

✓ Certificate for 100.64.1.254

-----BEGIN CERTIFICATE-----

```

MIIFnJCCA4agAwIBAgIUeJgBKIAHq2QIM+1t40i+dtltjSwwDQYJKoZIhvcNAQEL
BQAwUTEZMBcGA1UEAwQUXJhdG9yIFN5c2xvZyBDQTEUMBIGA1UECgwLUXJhdG9y
IExhYnMxHjAcBgkqhkiG9w0BCQEWOD21haWwAcXJhdG9yLm5ldAeFw0yNjA3MDgy
MDQzMzZaFw0zMTA3MDcyMDQzMzZaME0xFDASBgNVBAoMC1FyYXRvcjBMWjZMRUw
EwYDVQQDDAwMDAuNjQuMS4yNTQxHjAcBgkqhkiG9w0BCQEWOD21haWwAcXJhdG9y
Lm5ldDCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAggIBALrGC6zg5CT5HzoW
c2i7qUfsjtS+UsQ5GvLVvPTH/JmJKZiz1XeNgf/GcOGqkfmgJrL6G/LCWJYb+sz5
dXUaHD2d615G71UY/iK1lsykb1SIje0Uvw7nMPj3A8VgPtXqu2BTts/J8XLZ9mKn
yFARIrY0ABccf1DaQr5N8NYEFerA4dqz72WTctQ5ZqSsjgK5APK96mi6m0eb5Kqt
+3Bw7H/snGcU0HI/GMY/spbV+1yMe5znBP7NnrG/383qgdH1cboGxtwqrpKvqCN
5tDSbE/xd+YbuHhZaiv8kb5sLBaoBQaM2dhbssUrVqHGHSsmGCHtz19TUEktstX9
RUT8GnujAMnZ6LR5gUQlnaxSt/MuWHZs7XBTEvKhfYihuVoTpN7AIN/SqK5bf/Oi
otxHoIjpeCJCj8Yu2zlvnw1SCQje8tnjDFJwVQ2myZNQatJ8GdX/1BJmFwSey2LO
a6+9YaU3I2HtkBY2R+hbSEWzMJVubS2N9TkdYsvcmw2Zoq20FXxFOaJAwWY5luV
Ia51oqtTfXxp0+F352xmjpmw1C+Ihb380dajjv2EweSyt0c12XrJdJEpNyJyn14
XEk43hi4rMYkrBkZWanCYQLR8sK9KaCgbzHmR6Ho7E4L7wMYWgk19DW1o/PLkG6
5JeBasyyXbwgLDyoQrT38GmYvUAzAgMBAAGjcjBwMB0GA1UdDgQWBbT1hNKqGxsZ
7Fq64NtpcIY7ALaohzAfBgNVHSMEGDAWgBRskf8VTbB3RE7y8ITw5AWJQBAQKDAJ
BgNVHRMEAjAAMA4GA1UdDwEB/wQEAwIF4DATBgNVHSUEDDAKBggrBgEFBQcDATAN
BgkqhkiG9w0BAQsFAAOCAGeAk77DkGwJld0cCF2KJysst2ycm0s4Et0cunM4D96s
pMkbBV5s13shGM6Hq+RQy9rknrQuM4RoP2/ZctIc+Eny8YwMyHkcp7Uog43wDlEO
niYUde9HMgT6XjsGKKmFw2p8si60zIOKQktbMZUn40RBBLwStPwVam/X4IZeKfT0
J5I/0ka6V2bIJGeUxL7PjxtAoTuH+vvsh7EE9ZdNNNTy43B1C4czCoEir55v+S98
GJpUnKV557aQ3GjXiqpZJmM+Z8RmmamLgNhtSrp1Hf6sNKLMITgPZCZP9gEjjcqq
cHpGXFnpPC21aUdjb6ArD8KpRqGGAPWN5adZpkDWV43o3GxY63vrBZUbvLKyt75Q
CN0MIS6cEihouww0U9Q3qab+BrEdsr6oUBplazfMWRZwz6vBuKzh5ncWDvb1NpJ
rIHwGnK9yR2Z2JWHZS5Xh1y9uKBbXtdy0BFECyck/CwmyhwSkeKnnFYGuwIiH/t
pw9qBwbVUQXX1NPAKYndT6k9Js98h+QY0J3JvwqzsIZQGyHM860mG6CxLBqZSGyo
TDhNhCSVcvU1lrKFwx7/+aqKVIloqR0Rk3MzHDVQJ6ZloqogLYcSBT7WH2hNnfu8
XLBbS6htklXPcDiACuZe53sStwWkOxZQf0iMJgUk8R8dPZNkoUuvLgnlj2UN9m5Q
sbM=

```

-----END CERTIFICATE-----

✓ Certificate for CA

```
-----BEGIN CERTIFICATE-----
MIENDCCA52gAwIBAgICBHYwDQYJKoZIhvcNAQELBQAwYQxvCzAJBgNVBAYTA1JVMQ8wDQYDVQQQIEwZbn3Njb3cxZDZANBgNVBAcTBk1vc2NvdzEMMAoGA1UEChMDSEXMQ8wDQYDVQQLEwZRCmF0b3IxFDASBgNVBAMTC1FyYXRvcib0b2RlMR4wHAYJKoZIhvcNAQkBFg9tYWlsQHFyYXRvcib5uZXQwHhcnMjEwNzEyMTUyYmJmM5WhcnMzEwNzEyMTUyYmJmBRMRkwFwYDVQQDDDBBRcmF0b3IgU3l3bG9nIENBMRQwEgYDVQQKDAtRcmF0b3IgTGFIczEeMBwGCSqGSIb3DQEJARYPbWFPbEBxcMf0b3IubmV0MIICiJANBgkqhkiG9w0BAQEFAAOCAG8AMIICGKCAgEAmQmcErDsVsHgz3+C+Q1EEVlMN2HgbnBBlVwu5d0Cp9Em2T5+L2XrvGb6fd1XEEGDi4qi0d6DtCP015NaUCr4su7RVpxw/flzuHcq7DyS0kyjEA0dkaz3L20n1Lx5jjmnC8i+5vdaDhC94NfWdQdkHnTTO47PKB80RkaJTVACZAIa8ab86L02PmL54Y0iTSX+/IUo05SILElZewVJ8fkjnOwmGPiPX0dkP9YAV/6qfoQfa9PVUkPeV9UHNUpw2laN09hAog4UybeIzTV0bwwweiCnlj5TW16zn0JcfTViInn9HUw3EWmbi8YUhv86yTiLjMXHDafhFzQqTDKkg5J4dn7ye0AQzqXUo090jrBC4+fX5RgImJoXMeFS2exSB5f8PxDPBS95LQVuSzzXesuDuZQVJD9T0JT57u/27EB1TwQqMLgqo39RE8iaroduibUjFKAZBn3xxAOMx5EbPK0aliymq82PDBwB69Jdm/i0xSeRIA5r7C/mAjVfcDQxQVz97fJfXek0FChUjHkddgWpHuYqYQreq5sZX218te+86yh5RjRmgN52gb9c4KppJEE04fAXJYhMARuMhuswyCzWLPP8SHI0/QeHD4ifw+FyT8jh/qiFhcIcY48wC2/4I0sfP5OWtUfzfjuZF+g6SSGXhDdKidAWUz2mBa/UsZZsMvsECAwEAAanjMGewHQYDVR00BBYEFgyr/xVNShdETvLwhPDKBYlAEBAoMB8GA1UdIwQYMBAAFAbuy5jX1iZn8znnG0wGytbFKt/XMA8GA1UdEwEB/wQFMAMBAf8wDgYDVROPAQH/BAQDAggGMA0GCSqGSIb3DQEBChUAAGBAI5bpB870oTPMLGE4ZsTSa2LmjYPLz9i0Z0ubzUAZwz7B81RxSdFDz90Y2QNmU0x7Pztzz13IW5XsmtRqOf3Q7jRUWSzBi/JFTEqWALVcGrFixzA9Q0PJ/n0o1FC8hlxmwxA6yoXRnW3ntngAJiziHrrtNKKjILT0Jv4zbk25Jtm

-----END CERTIFICATE-----
```

2.10 CONNECTION TO QRATOR LABS

Connection: BGP

If your organization has its own autonomous system (AS), you can connect to Qrator Labs via BGP and advertise your prefixes. Thus, the Qrator Labs autonomous system will act as a transit AS, through which all traffic associated with the advertised prefixes will pass.

The Qrator Labs network is connected to several Tier 1 ISPs and advertises your prefixes to each of them. Each ISP will choose the route that goes through the Qrator Labs router closest to it so that traffic from anywhere in the world will reach you with minimum latency. Any traffic that Qrator Labs recognizes as an attack will not be sent to your network.

To connect to Qrator Labs via BGP:

1. [Add prefixes.](#)
2. [Set up a channel to Qrator Labs](#)
3. [Set up BGP peering with Qrator Labs.](#)
4. [Describe the service map.](#)
5. [Advertise prefixes.](#)

Adding prefixes

Specify in advance in the [“Prefixes” section of your dashboard](#) each prefix that you are going to advertise and the corresponding number of your AS, which will be the origin for this prefix. The prefix must belong to that AS according to the Internet registrar database.

To confirm the action of adding a prefix, Qrator Labs will send an email with a link to the address listed as [tech-c](#) for the corresponding AS.

Setting up a channel to Qrator Labs

A BGP session requires a “point-to-point” channel between your network and the Qrator Labs network. Virtual tunnels such as GRE, IPIP or MPLS L2 VPN are most often convenient for this. In some cases, other methods are available, such as dedicated channels or physical switching within the same data center.

BGP peering is organized over such a channel. So, selected prefixes from your AS will be advertised only to the Qrator Labs network.

This is important because you need to ensure that all inbound traffic related to the advertised prefixes passes through Qrator Labs. Note that if you advertise prefixes to ISPs directly, an attacker will be able to direct a DDoS attack through another route, bypassing the Qrator Labs network.

Even if the ISP advertises a route to a supernet (less specific route) and Qrator Labs advertises a route to the subnet (more specific route), there would still be the risk of accidental or targeted traffic to the subnet through the ISP, bypassing the Qrator Labs network.

To ensure fault tolerance and reliability, it is recommended to create two or more channels to Qrator Labs. For example, these may be two different GRE tunnels through different providers. For its part, Qrator Labs also pays great attention to redundancy. So even if part of the hardware goes down or fails, AS as a whole will remain available and be able to keep your site available and protected through another channel.

Setting up BGP peering with Qrator Labs

After creating a “point-to-point” channel between your network and the Qrator Labs network, configure your hardware to set up a BGP session over this channel.

For step-by-step instructions, including configuration examples for popular hardware manufacturers, contact Qrator Labs support.

Service map description

When analyzing BGP traffic, traffic scrubbing centers are guided by the information provided by the customer in the “IP addresses” and “Ports and protocols” sections of the dashboard. This information is collectively referred to as *service map* and describes the direction and volume of legitimate traffic expected by the customer.

To start describing traffic, create one or more services in “Services”. Then, for each service, define a list of IP addresses (“IP addresses”), ports and protocols (“Ports and protocols”). All inbound and outbound traffic that does not match what is described in the service map will be discarded.

Advertising prefixes

The immediate start of the advertisement of your prefixes takes place within BGP and does not require action in the personal dashboard. The Qrator Labs network will begin advertising your prefixes as soon as your hardware starts advertising them to the Qrator Labs autonomous system. If you pause the advertising of any prefix, Qrator Labs will immediately suspend its advertising and also stop passing traffic to this prefix.

Note that Qrator Labs does not handle BGP attribute `community` when advertising prefixes.

2.11 CONNECTION TO QRATOR LABS

Connection: Secure DNS server

This article describes how to connect [Secure DNS server](#).

The service is connected in three steps:

1. [Set up communication between your DNS server and the Qrator Labs DNS server](#).
2. [Make sure the Qrator Labs DNS server is responding to your zone requests](#).
3. [Specify the Qrator Labs DNS server as authoritative for your zone](#).

**TIP**

Before you connect, you should, for your DNS server, configure a new IP address that has not been used or published before. In doing so, you can disable access to the server via previous IP address in [step 3](#).

1. Setting up communication between DNS servers

Setting up secondary DNS server communication

Before connecting the service in [secondary DNS server mode](#), make sure that the DNS server you are using allows you to configure zone transfer to secondary authoritative DNS servers. If you are using the domain registrar's or hosting's DNS server, make sure this is enabled in their control panel (e.g: [instructions for GoDaddy](#)).

In the access control list (ACL) of your DNS server, allow zone transfer to the server `ns.qrator.net`. This will allow the Qrator Labs DNS server to act as a secondary DNS server.

Tell the Qrator Labs technical support team the IP address of your primary DNS server. They will configure the secondary DNS server accordingly, and then the secondary DNS server will request a zone transfer from the primary DNS server.

Setting up DNS proxy server communication

To enable [DNS proxy server mode](#), tell the Qrator Labs technical support team the IP address of your DNS server and a list of domains that need to be proxied. Technical support will configure the Qrator Labs DNS server accordingly, and then the server will be able to forward requests for the specified domains to your server.

2. Check configuration

Before you switch to Qrator Labs DNS server, check the new configuration. To do this, send one or more requests about your domains to this server. If the server does not find the record you are looking for, or returns incorrect information, please contact technical support to find out what is causing the problem.

You can use the `nslookup` utility to check, specifying the DNS server as the last argument. For example, a command to get the IP address of the domain www.example.com from the Qrator Labs DNS server would look like this:

```
nslookup www.example.com ns.qrator.net
```

3. Switching to the Qrator Labs DNS server

In the domain registrar's control panel, specify the Qrator Labs DNS server as the authoritative server for your zone, and remove your DNS server from the list.

Depending on the TTL settings on the domain registrar's side, it may take up to 72 hours to update the NS records.



TIP

If you used a new IP address that is different from the one specified in the NS records earlier, then after updating the records, you should disable access to the former address in the firewall settings on your side. Thus, without knowing the new IP address, an attacker will be unable to direct an attack to the primary server. Updating the Qrator Labs DNS server on time is therefore made difficult. This DNS server hiding technique is known as *hidden primary*.

03

Dashboard

Interface overview

Lists of resources

Qrator

Bot protection

CDN

Utilities

Certificate storage

Access rights

Notifications

My Profile

SSO

3.1 DASHBOARD

Overview of the dashboard interface

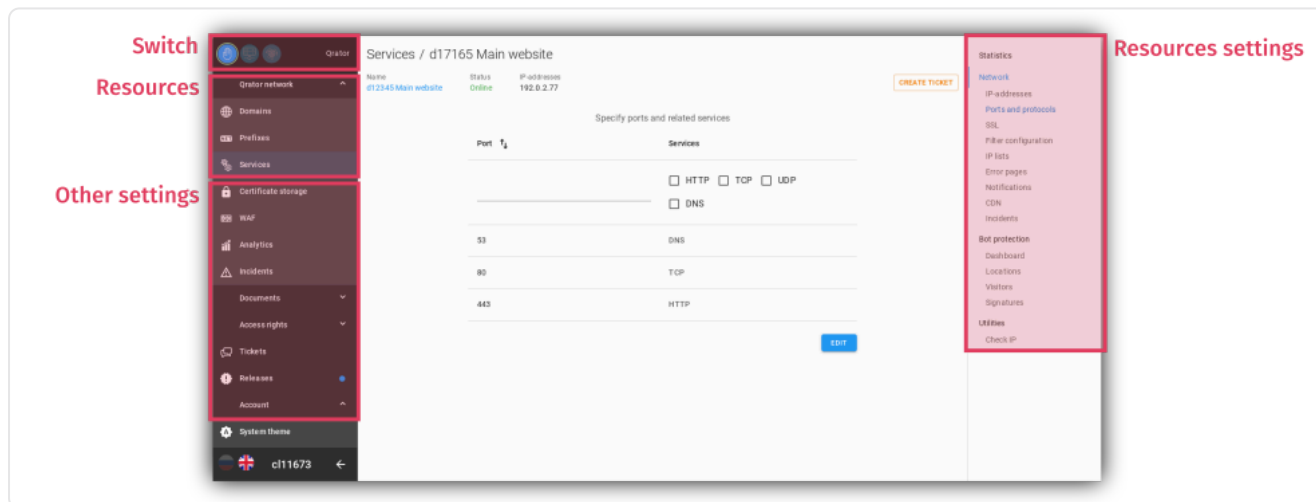
In their Qrator Labs account, customers can configure the services they use, as well as track progress reports for those services.

On this page, you can see a complete list of the different sections of the personal dashboard, divided into several parts:

- Managing the list of [resources](#).
- Managing [resources settings](#).
- Managing [ingress settings](#).
- Managing [other settings](#).

[Resources](#), [Ingress settings](#) and [Other settings](#) are displayed in the *main menu* of the dashboard, located on the left side of the screen. The list of visible elements in the main menu varies depending on the mode selected in the *switch*: **Qrator**, **Ingress**, **WAF** or **CDN**.

[Resources settings](#) are displayed in the *additional menu* of the dashboard, located in the right side of the screen. This menu is displayed only after you've selected a particular resource in the [Domains](#) or [Services](#) section.



Accessibility of some modes and sections of the personal dashboard may depend on what services are enabled for the customer. If you have any questions, please contact Qrator Labs tech support.

Also, some sections or individual settings in them may not be accessible, if the user account is not included into groups with certain access rights. See the description in the [Groups](#) section for more details.

Resources

The key concept for setting up most Qrator Labs services is *resources*. A resource can be, for example, a website or an API gateway that needs to be protected from attacks. For different Qrator Labs services, different types of resources are configured: *domains* or *services*.

To begin working with a service, add the new resource to the list using the sections listed in the table below. Then go to [resources settings](#).

Section	Description
Domains	This is the <i>domains</i> settings for such services as reverse HTTP proxies and CDN integration . For each domain, load balancing is performed independently between the specified upstreams.
Prefixes	Settings for connecting to Qrator Labs via BGP , which must be performed before adding services in Services .
Services	Configuring the <i>service map</i> to work via BGP . Each service describes the type and volume of legitimate traffic for specific IP addresses. This allows the Qrator Labs network to drop traffic that uses ports and protocols not specified by the service.

**NOTE**

The [Prefixes](#) section relating to BGP settings should not be confused with the [Ingress → Prefixes](#) section relating to [ingress settings](#).

Resources settings

Most areas of the personal dashboard contain settings that apply to a specific [resource](#).

To go to resources settings, first go to [Domains](#) or [Services](#) and select the desired resource from the list. After that, a list of sections related to this resource will be displayed in the additional menu.

The set of available sections differs for domains and services.

Section	Description	For domains	For services
Statistics	Statistics of requests, errors and applied traffic blocking.	✓	✓
Services	Upstream and balancing settings for the reverse HTTP proxy , reverse WebSocket proxy and reverse TCP proxy .	✓	
IP addresses	IP addresses to which the Qrator Labs network redirects cleaned traffic when operating via BGP .		✓

Ports and protocols	Type and volume of legitimate traffic expected for the service's IP addresses.		✓
TLS	Settings according to which HTTPS is used for the domain or service.	✓	✓
IP lists	Black and white lists used when determining traffic legitimacy.	✓	✓
Tunnels	Settings related to Qrator Labs connection via tunnels .	✓	
Filter configuration	Settings related to filtering based on HTTP headers, regions and other attributes.	✓	✓
HTTP redirects	Configuration of HTTP redirects working on the reverse HTTP proxy side.	✓	
Error pages	Configuration of pages that will be shown to the user if problems accessing the resource occur.	✓	✓
Notifications	Configuration of notifications that will be shown when specified traffic thresholds are exceeded.	✓	✓
Incidents	Reports on detected attacks and other incidents..	✓	✓
Bot protection	Statistics and settings related to bot protection .	✓	✓
CDN Qrator.CDN → Settings	Statistics and settings related to CDN integration .	✓	
Utilities	Auxiliary utilities for checking the status of services.	✓	✓

Ingress settings

The following sections configure the [Ingress filtering service](#). This service is intended for administrators of transit networks that need to clean incoming traffic to end-user networks. To configure a service, define a list of *prefixes* of end-user networks and a list of *policies* applied to those prefixes.

Dashboard sections related to ingress filtering configuration are hidden by default. To see them, use the switch above the main menu (see [interface description](#)).

Section	Description
Ingress	This section displays both the list of prefixes from Ingress → Prefixes and the list of policies from Ingress → Policies . Also in this section you can add a new prefix or a new policy.
Ingress → Prefixes	List of prefixes of end-user networks whose transit traffic needs to be protected.

Ingress → Policies	List of traffic filtering policies applied to the traffic of specific prefixes.
--------------------	---

Other settings

Some dashboard sections are not related to management of specific services.

All of the sections listed below are accessible through the main menu of the personal dashboard.

Section	Description
Certificate storage	Managing TLS certificates that can be used for reverse HTTP proxy and CDN integration .
WAF	Configuring the Web Application Firewall.
Analytics	Traffic reports.
Incidents	Reports on detected attacks and other incidents.
Reports	Traffic reports.
Finance	Finance reports.
Company	Management of bank details and other data needed to interact with Qrator Labs.
API tokens	Managing the tokens that allow using the Qrator API.
Users, Groups	Management of users that have access to your account, and granting permissions to groups of users.
Tickets	Interaction with Qrator Labs technical support.
Updates	Information about changes in the Qrator Labs dashboard and services.
Notifications	Configuration of receiving the email notifications.
My Profile	Managing your profile in the dashboard.

**NOTE**

The [Incidents](#) section is available both in the main menu and under [resource settings](#) in the advanced menu. Depending on which menu through which the section is opened, it will display either data about all incidents, or data about incidents for a specific resource.

Personal dashboard: Domains



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

Many Qrator Labs services are configured to work with a specific *domain*. You can [see a list of domains](#) connected to your account in the “Domains” section of the dashboard. Here you can also [add new domains](#).

For each added domain, sections with statistics and settings of particular services become available in the personal dashboard. See [Overview of the dashboard interface](#).

View list of domains

The domains available to your account are displayed as a table with the following fields:

- **ID** — Unique domain identifier in Qrator Labs (e.g., `d12345`), and the domain name itself (e.g., `example.com`).

Click on the ID to go to the statistics and settings for that domain.

- ****{{ QPI }}**** — IP address in the Qrator Labs network assigned for use with this domain. It is displayed only for domains that have passed moderation (see [Add domain](#)).
- **Status** — Current status of the service.
 - **Moderation** — Domain is being checked by Qrator Labs employees before being included into the service (see [Add domain](#)).
 - **Online** — Domain is enabled.
 - **Offline** — Domain is disabled.
 - **Updated** — The changed service configuration is being applied.
 - **Error** — An unexpected error occurred, domain may be unavailable.
- **Upstreams** — List of upstreams among which domain traffic is distributed. It is configured in [Upstreams](#).
- **Traffic** - Speed of traffic between users and the Qrator Labs network for a given domain. Each value is calculated as the average speed per second based on data over the last minute.
 -  - inbound traffic.
 -  - outbound traffic.

Next to the table column headers are filter fields you can use to quickly find domains that meet certain conditions.

By default, the table displays no more than 20 domains at a time. If more domains are available for your account, use the < and > buttons below the table to navigate to other parts of the list. You can also select the number of domains to display at a time using the **Rows per Page** field.

Add domain

1. Open the “Domains” section and click **Add domain**.
2. In the dialog box that appears, enter the domain name (e.g., `example.com`). Click **Next**.
3. Specify the upstreams where traffic for the domain will be redirected to. There are two ways to do this:

- **Manually enter IP addresses.**

Enter one or more IP addresses in the **Upstreams** field.

- **Determine IP addresses using DNS.**

Click the **DNS lookup** link. In the dialog box that appears, enter the **Hostname** on which the required upstream is currently available, and click **Find**. The list of addresses from the A record of the domain will be automatically loaded into the dialog box. If necessary, uncheck the addresses that shouldn't be upstream in the new configuration. Then click **Import**.



WARNING

When connecting a domain, it is not recommended to use upstreams that were previously publicly associated with the domain, as this may allow an attacker to bypass the Qrator Labs protection. For more details, see [Changing the current IP address](#).

4. Click **Create**.

The domain will be added to the general list of domains and will be assigned the status **Moderation**. Within a few hours, Qrator Labs specialists will check the application and assign the domain an **Online** status, or they might contact you for further clarifications.

After successful connection, the domain will be assigned a Qrator IP, which you can use to configure services — e.g., for [Reverse HTTP proxy connection](#).

Delete domain

To delete a domain, contact Qrator Labs tech support.

Configuration example

ID ↑	Qrator IP	Status	Upstreams	Traffic ↑↓
		All ▾		
d123 domain1.example.com	185.65.148.91	Online	203.0.113.1 203.0.113.2 203.0.113.3	 64 bps  114.667 bps
d456 domain2.example.com	178.248.233.165	Online	203.0.113.6 203.0.113.	 23.467 bps  47.467 bps
d789 domain3.example.com		Moderation	203.0.113.8 203.0.113.9	

Personal dashboard: Services



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the “Services” section of the dashboard, you can describe what traffic should be considered legitimate when [connecting via BGP](#).

Each service includes:

- one or more IP addresses (“[IP addresses](#)”),
- one or more open ports on which certain network protocols are allowed (“[Ports and protocols](#)”).

The list of services configured in your account is called *service map*. The service map should fully describe the expected legitimate traffic that will pass through the Qrator Labs network. DDoS protection will be automatically configured to detect and prevent attacks against the specified protocols. In addition, all inbound and outbound traffic to IP addresses, ports, and protocols not specified in the service map will be denied.

Before adding services, you must [configure the appropriate prefixes](#).

For each added service, sections with statistics and settings for specific services become available in the personal dashboard. See [Overview of the dashboard interface](#).

View list of services

The services available to your account are displayed as a table with the following fields:

- **ID** — Unique identifier of the service in Qrator Labs (eg., [d12345](#)), and the name of the service (eg., [Main site](#)). Click on the ID to go to the statistics and settings of that service.
- **IP-addresses** — List of IP addresses of the service.
- **Status** — Current status of the service.
 - **Online** — service is enabled.
 - **Offline** — service is disabled.
 - **Updated** — the updated service configuration is being applied.
 - **Error** — unexpected error has occurred, the service might be unavailable. Contact Qrator Labs technical support for details.

- **Traffic** — Speed of traffic between users and the Qrator Labs network for this service. Each value is calculated as average speed per second based on information from the last minute.
 -  — inbound traffic.
 -  — outbound traffic.

Next to the table column headers, there are filter fields that you can use to quickly find services that meet certain conditions.

By default, the table displays no more than 20 services at a time. If more services are available for your account, use the < and > buttons below the table to navigate to other parts of the list. You can also select the number of services to display at one time using the **Rows per page** field.

Add service

To add a new service to the service map:

1. Open the **Services** section and click **Add service**.
2. In the dialog box that appears, enter an arbitrary name for the service (e.g., "Main site").
3. Specify the IP addresses where the service should be available. Each IP address entered must belong to one of the confirmed prefixes in "[Prefixes](#)".

You can specify addresses in two ways:

- **Manually enter IP addresses.**

Enter one or more IP addresses in the **IP Addresses** field.

- **Determine IP addresses using DNS.**

Click the **DNS lookup** link. In the dialog box that appears, enter the **Hostname** on which the required upstream is currently available, and click **Find**. The list of addresses from the A record of the domain will be automatically loaded into the dialog box. If necessary, uncheck the addresses that shouldn't be upstream in the new configuration. Then click **Import**.

4. Click **Confirm**.

After creating a service, you should configure the ports and protocols allowed for it. To do this, click on the ID of the created service, select "[Ports and protocols](#)" and make the necessary settings.

When all necessary settings have been made, [enable the service](#).

Enable service

To enable the service and allow appropriate traffic to pass through:

1. In the **Services** section of the dashboard, select the service.

2. At the top of the page, click **Enable service**.
3. In the dialog box that appears, confirm the action by clicking **Enable**.

The status of the service will show **Updated** for a while and then will switch to **Online**.

Disable service

To disable a service, contact Qrator Labs technical support.

Delete service

To delete a service, contact Qrator Labs technical support.

Configuration example

ID 	IP-addresses	Status	Traffic 
		All 	
d123 Main website	203.0.113.10		
d234 DNS server	203.0.113.1		

Personal dashboard: Prefixes



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the “Prefixes” section of the dashboard, you can manage the list of prefixes used to [connect to Qrator Labs via BGP](#). The prefixes must be configured together with the service map setting in “[Services](#)”.

View list of prefixes

To see all the prefixes available to your account:

1. Open the **Prefixes** section.
2. If necessary, switch to the required tab: **Verified** or **Unverified** (see [Add prefix](#)). The **All prefixes** tab is selected by default.

The prefixes will be listed in a table with the following columns:

- **Prefix** — the unique ID of the prefix in the Qrator Labs system (e.g., [p123](#)), as well as its subnet mask.
- **AS number** — the number of the Autonomous System (AS) from where this prefix originates.
- **Status** — status of this prefix in the system (see description in [Add prefix](#)):
 - **New** — prefix has been added to the list, but the corresponding request has not been submitted for consideration yet.
 - **Processing** — request is under moderation.
 - **Approved** — prefix is approved and may be advertised.
 - **Rejected** — request is rejected.

By default, the table displays no more than 30 prefixes at a time. If you have more prefixes configured for an account, use the < and > buttons below the table to navigate to other parts of the list. You can also choose how many prefixes to display at one time using the **Rows per page** field.

Add prefix

To add a new prefix and verify ownership of it:

1. In the **Prefixes** section of the dashboard, click **Add prefix**.
2. In the **Prefix** field, specify the subnet prefix you are going to advertise over the Qrator Labs network. When entering the prefix, use CIDR notation, for example: `198.51.100.0/24`.
3. In the **AS number** field, specify the number of the autonomous system to which the prefix belongs. This can be done in two ways.

- **Manual entry.**

Enter the autonomous system number in the text field, for example: `64500`.

- **Auto search**

Next to the **Prefix** field, click . A list of one or more autonomous systems to which the prefix belongs will load. Check the box next to your autonomous system number.

If you want to go back to entering the number manually, click the **Manual entry** link.

4. Click **Add**.

A new entry will appear in the prefix list with the data you entered. The **Status** column will display the  icon.

**WARNING**

The next step will initiate prefix validation, after which the entered data cannot be changed. Therefore, make sure that the data is entered correctly.

If there is a mistake, correct it. To do this, click on the prefix in the list and on the page that opens, click **Edit**. In the dialog box that appears, make the necessary changes in the **Prefix** and **AS number** fields and click **Save**.

5. Click on the prefix in the list. On the page that opens, click **Send for moderation**.
6. Qrator Labs will send an email with a link to the address listed as `tech-c` for the specified autonomous system according to the Internet Registrar database. Follow the link in that email to verify ownership of the prefix.

After the ownership of the prefix is successfully verified, the  icon will appear in the **Status** column for that prefix.

If ownership of a prefix has not been verified within the allotted time and the link becomes invalid, the  icon will be displayed for that prefix. You can initiate a re-verification of ownership by following the previous step again

After adding a prefix, configure the services that should be available in it under “[Services](#)”.

Configuration example

ALL PREFIXES (3)				APPROVED (1)	NOT APPROVED (2)
Prefix	AS number	Status ↑↓	Announce ↑↓		
p123 198.51.100.0/24	AS64500				
p456 203.0.113.0/24	AS64500				
p789 233.252.0.0/24	AS65536				

In this example, three prefixes have been added. An ownership verification process was initiated for all three, but only for the `203.0.113.0/24` prefix was it successfully completed. This prefix can now be advertised via BGP on the Qrator Labs network.

Dashboard: Statistics



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups.

For each [domain](#) or [service](#), the “Statistics” section is available, which displays data on inbound traffic. You can switch between different [data categories](#), [periods](#) and [views](#).

You can download the chart as an image by clicking the  button. The image will be downloaded in the same size as it was displayed in the browser at the time of saving.

Select data category

At the top of the page is a drop-down list where you can select a data category to display on the chart. The “Traffic” category is displayed by default.

A category contains several data sets. Each data set is shown in its own color on the chart. You can hide a data set from the chart by clicking on its symbol in the legend.

The following categories are available:

- **Traffic**

Volume of inbound and outbound traffic, bps. Different colors indicate different types of traffic:

- *Inbound* — all traffic from users.
- *Missed* — traffic from users that was not part of the attacks.
- *Outbound* — traffic from upstream to users.
- *Paid* — traffic for which Qrator Labs bills the customer (only in “Chart” view).

- **Traffic (inbound)**

Volume of traffic from users, bits/sec. The *TCP*, *UDP*, *ICMP*, and *Other* protocols (all other protocols) are denoted by different colors.

- **Packets**

Volume of inbound and outbound traffic per second. Different colors indicate different types of traffic:

- *Inbound* — All traffic from users.
- *Missed* — Traffic from users that was not part of the attacks.
- *Outbound* — Traffic from upstream to users.

- **Packets (inbound)**

Number of packets from users per second. Different colors are used for *TCP*, *UDP*, *ICMP*, and *Other* (all other protocols).

- **Requests**

Total number of requests per second (static and dynamic), as well as the number of dynamic requests only.

- Static requests are GET requests with known extensions in the names of requested files (styles, images, video files, etc.).
- All requests except static ones are considered dynamic requests.

- **Response time**

Number of requests per second that upstream responded to within the specified period. Different colors show data for different response periods: *Less than 200ms*, *200-500ms*, *500-700ms*, *700-1000ms*, *1-1.5s*, *1.5-2s*, *2-5s*, *More than 5s*.

- **Errors**

Number of requests per second that the Qrator Labs reverse HTTP proxy responded to with an error code. Data on HTTP 4xx and HTTP 5xx error codes are displayed.

- **Blacklist**

Number of blocked IP addresses from which requests were received. Different colors denote IP addresses from the automatically updated Qrator Labs and WAF blacklists, as well as the customer's own blacklist.

- **Bot protection**

Number of *Checks* and *Blocks* per second conducted under the [Redirect with cookie setting](#).

Select period

You can change the period for which data is displayed on the chart in one of two ways.

- Hold down the left mouse button at the beginning of the section of the chart you are interested in and, without releasing the button, drag to the end of the section. The scale of the chart will automatically change to include data from the selected area only. This zooming can be done several times in succession.

To return to the previous zoom, click . To return to the zoom level used before zooming out, click **Reset zoom**.

- Select one of the preset periods from the drop-down list at the top of the page. The required data will be automatically loaded and displayed on the chart. The chart scale (see previous paragraph) will automatically reset.

The default period is **5 hours**.

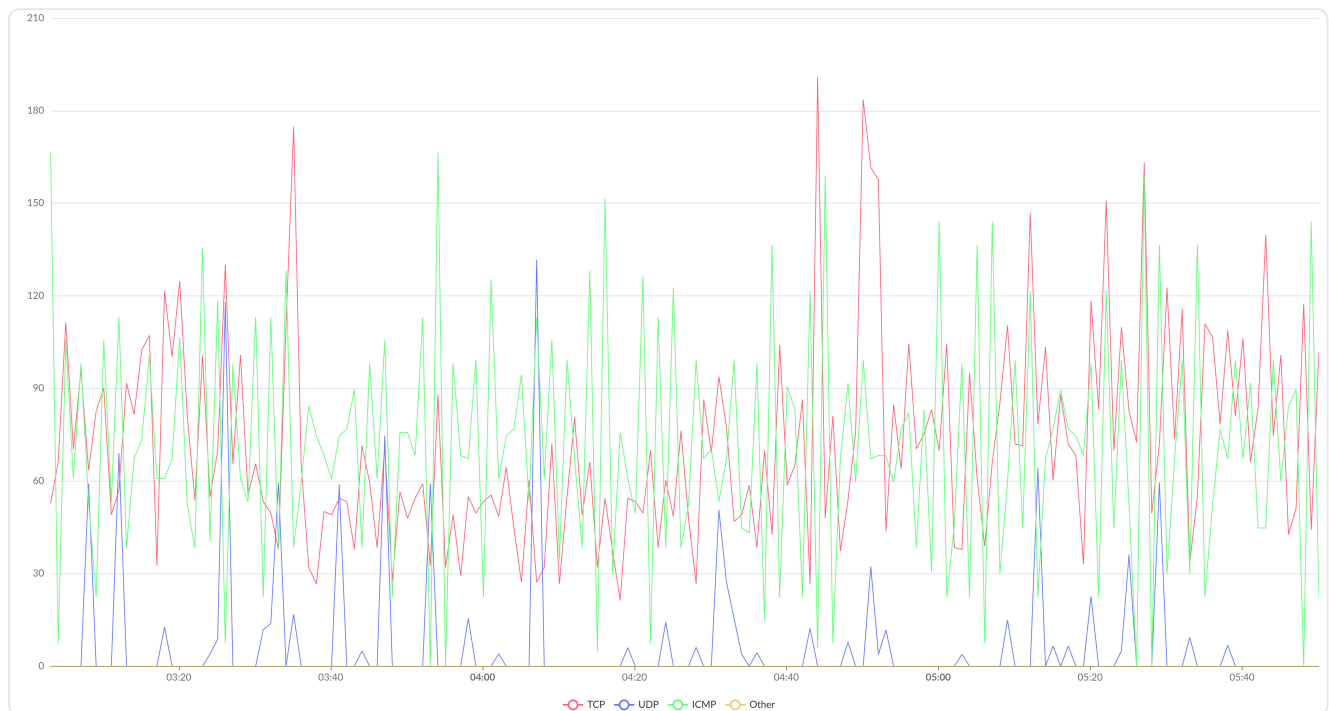
View selection

For all [data categories](#) except “Requests”, a view switch is displayed above the chart. Use it to present the data in one of the following ways:

Chart (default)

Different values are shown as independent lines on the chart.

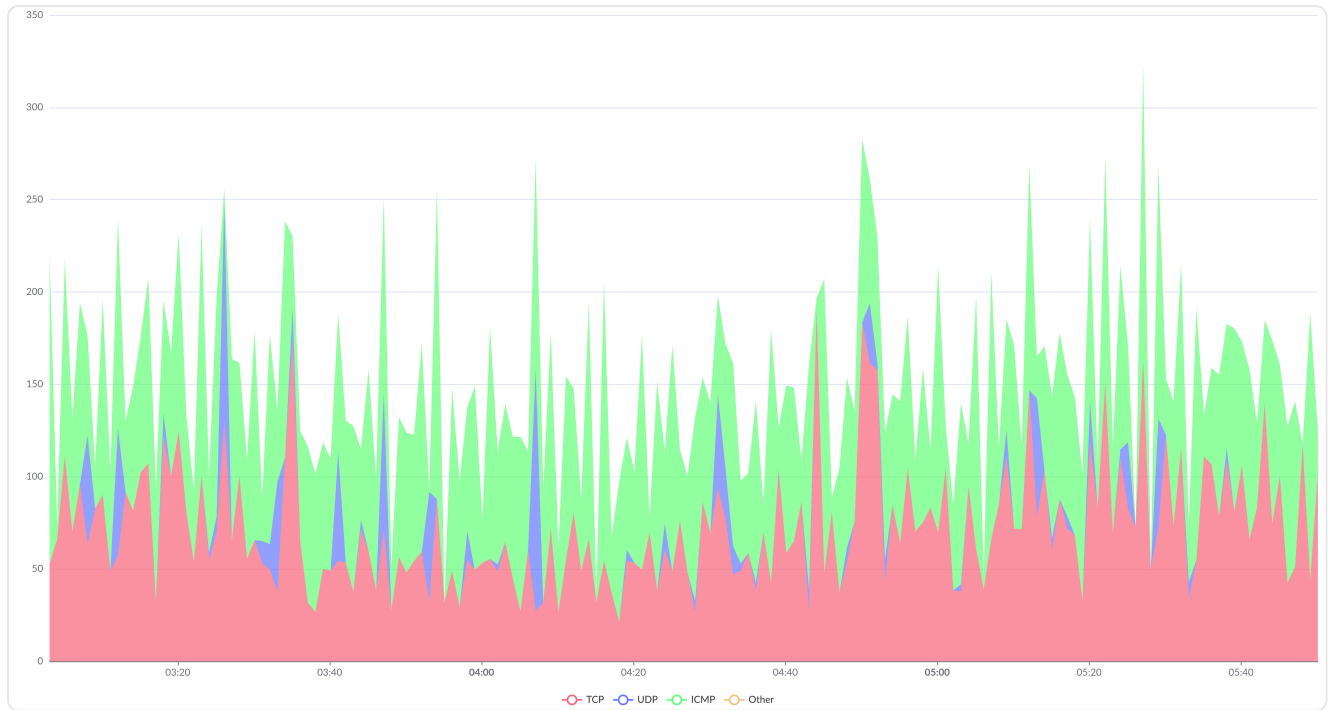
This view type is used if the **Stack** and **Stack 100%** options are disabled.



Bar chart

The volumes of different types of traffic are shown in different colors on top of each other, allowing you to assess the share of each to the total.

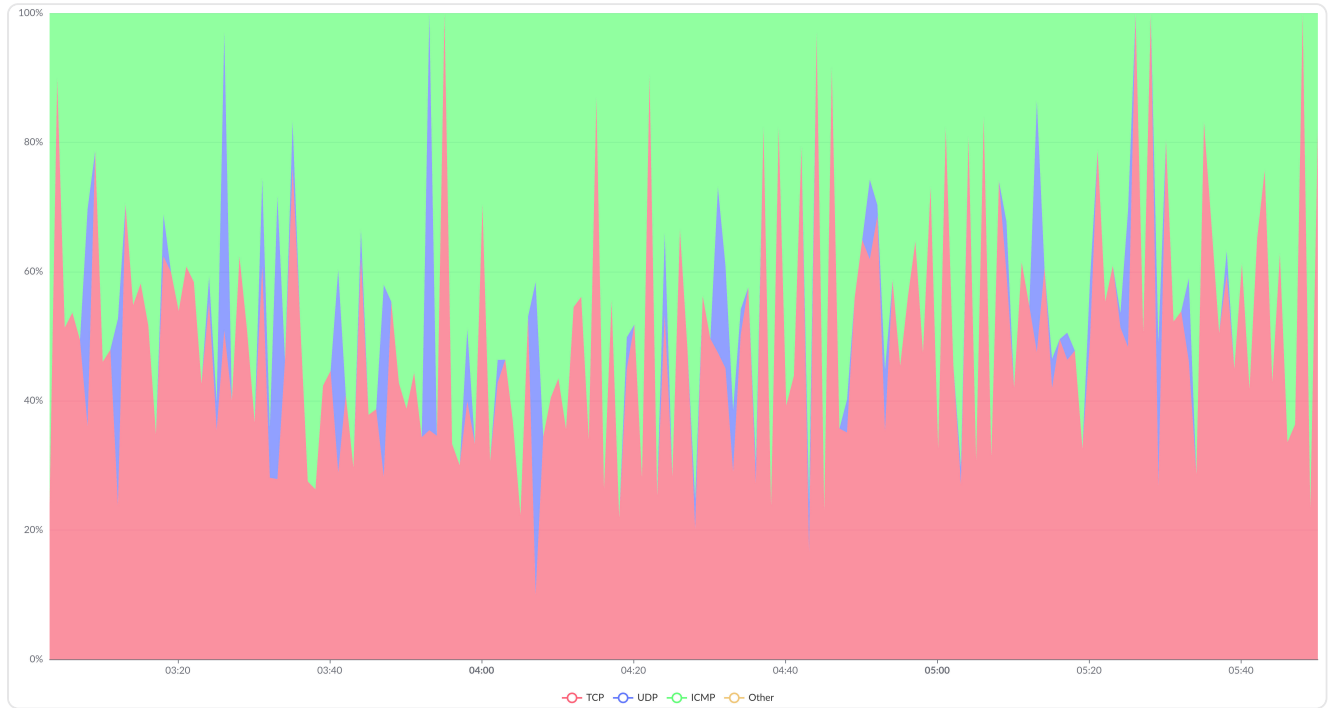
To select this view type, set the **Stack** option.



Stacked bar chart

It is formed similarly to the regular *stacked bar chart*, but the total value at each point in time is taken as 100% and is displayed on the entire height of the chart.

To select this view type, set the **Stack 100%** option.



Personal dashboard: Services



NOTE

This section of the dashboard:

- is accessible only for domains; see [Overview of the dashboard interface](#).
- can be edited only by users with *Technical role* and viewed only by users with *Admin role*. See [Users](#).

In this personal dashboard section, you can manage multiple types of *services* on your Qrator Labs account. The list of services defines how the users' requests will be routed to your upstreams.

Each service represents:

- one port on your Qrator IP for connecting to the users,
- one or more IP addresses and ports for connecting to the upstreams.

The following service types are available:

- **HTTP** — [reverse HTTP proxy](#),
- **WebSocket** — [reverse WebSocket proxy](#),
- **TCP Proxy** — reverse TCP proxy in the [Proxy Protocol](#) mode,
- **TCP** — [tunneling](#).

Note that the “Services” interface is not intended for using from multiple devices or browser tabs simultaneously. If someone has modified the data after you started working in the section, Qrator Labs will display a message about the conflict. In such a case, click **Save** to confirm entering your data (overwriting any changes made by the other user), or click **Cancel** to reset your changes and reload the latest data.

View list of services

To view the list of services configured for the domain:

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **Network** → **Services**.

The upstreams configured the domain will be displayed as a table.

☐							ADD
	Filter						
☐	HTTP	Port: 80	Web Backend 1	Web Backend 2	Web Backend 3		
☐	HTTP	Port: 443	Web Backend 1	Web Backend 2	Web Backend 3	HTTP/2	
☐	WEBSOCKET	Port: 8080	Video Server 1	Video Server 2			

For each service, the following is displayed:

- The service type.
- The port number at the Qrator IP.
- Names or addresses of the upstreams. If more than three upstreams are configured for a service, click to see the full list.
- Symbols representing additional features of the service:
 - – an unencrypted reverse HTTP proxy,
 - – an encrypted reverse HTTP proxy,
 - **HTTP/2** – a reverse HTTP proxy with the HTTP/2 protocol support,
 - – a service with a whitelist-only access.

To quickly find entries, use the filter field above the list. You can enter the following details in the filter:

- Protocol name, e.g., “HTTP” or “WebSocket”
- Open port number, e.g., “80” or “53”
- Upstream's name, IP address, or port number, e.g., “198.51.100.0”.

Configure a HTTP service

A service of the “HTTP” type defines parameters for the [reverse HTTP proxy](#). The proxy accepts users' requests to a one port at the Qrator IP and sends them to one of the upstreams, according to the selected [balancing algorithm](#).

When configuring the service, you can enable the TLS encryption for processing the users' requests or for connecting to the upstreams. Note that to enable TLS for processing the users' requests, you must have at least one certificate added to the [Certificate storage](#) and installed in the [TLS](#) section of the domain settings.

To add or edit a service of the “HTTP” type:

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **Network** → **Services**.
3. Click **Add** for configuring a new service or click next to a previously added service to edit it.

When adding a new service, specify the “HTTP” type.

4. Specify the general settings for the service.

- **Port** — port number to which a user connects.
- **Use TLS** — process requests using HTTPS, not HTTP. This setting is available if the domain has at least one TLS certificate installed.
- **Use HTTP/2** — process request made using the [HTTP/2 protocol](#). This setting is available if the domain has at least one TLS certificate installed, and the **Use TLS** option is enabled.
- **Whitelist-only access** — reset connections from IP addresses not included in the [whitelist](#).

5. Click **Upstream settings**.

A dialog will appear for managing the upstreams lists and balancing settings.

6. In the top part of the dialog, specify the general load balancing settings.

- **Balancing algorithm.** This setting affects how the reverse HTTP proxy would [select upstream](#) where incoming requests from the user would be sent to.
 - *Round-robin:* the reverse proxy will try to ensure even load distribution between enabled upstreams. Different requests from the same user may go to different upstreams.
 - *IP hash:* the reverse proxy will assign an upstream to a user as long as that upstream is available.
- **Use weights.** If this option is enabled, you can specify a weight for each upstream to influence distribution of requests. Upstream weights are interpreted differently depending on the algorithm you use. The effect of weights on the performance of balancing algorithms is described more in [Upstream selection algorithms](#). If the option is disabled, the weights of all upstreams are considered equal.
- **Use TLS** — use the encrypted HTTPS protocol for connecting to the upstreams.
- **Fallback upstreams** If this option is enabled, reverse HTTP proxy will switch from the primary list to the fallback list when upstreams from the primary list are unavailable. The switching principle is described

in [Switching between upstream lists](#). If the option is disabled, fallback upstreams are not used even if the primary ones are unavailable.

7. Configure the upstreams for the service. You can organize independent [primary and backup lists of upstreams](#). Note that there must always be at least one upstream in the primary list — you won't be able to apply the changes otherwise.

To add an upstream, click **Add upstream** next to a list. In a dialog that appears, specify the IP address, the port and, optionally, the name for the upstream. For convenience, you can fill the form with another upstream's data: click , then click on an IP address (to copy the name and IP address without the port) or a port (to copy all the data). If you want to temporarily exclude the upstream from the balancing algorithm, uncheck the **Enabled** checkbox. To finish editing and go back to the previous dialog, click **Add**.

Additionally, you can specify a DNS record in the settings (you must first contact technical support to make this option available).

To move an upstream from one list to another or change its position in the list, drag and drop the corresponding card.

To edit an upstream, click



on its card, make the necessary changes and click **OK**.

To delete an upstream from a list, click



on its card.

8. Click **Save** in the service settings dialog.

If you close the dialog without clicking this button, the changes will be lost.

9. If necessary, add or edit other services before saving the changes. To save the changes, click **Save** in the notification at the bottom of the screen.

Configure a WebSocket service

A service of the *“WebSocket”* type defines parameters for the [reverse WebSocket proxy](#). The proxy accepts users' requests to a one port at the Qrator IP and sends them to one of the upstreams, according to the *IP hash* algorithm.

When configuring the service, you can enable the TLS encryption for processing the users' requests or for connecting to the upstreams. Note that to enable TLS for processing the users' requests, you must have at least one certificate added to the [Certificate storage](#) and installed in the [TLS](#) section of the domain settings.

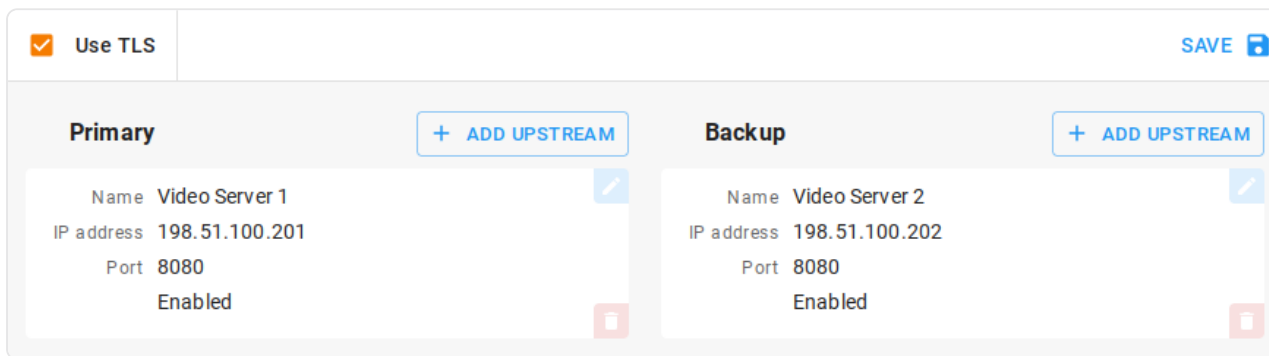
To add or edit a service of the *“WebSocket”* type:

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **Network** → **Services**.
3. Click **Add** for configuring a new service or click  next to a previously added service to edit it.

When adding a new service, specify the *“WebSocket”* type.

4. Specify the general settings for the service.
 - **Port** — port number to which a user connects.
 - **Use TLS** — process requests using HTTPS, not HTTP. This setting is available if the domain has at least one TLS certificate installed.
 - **Whitelist-only access** — reset connections from IP addresses not included in the [whitelist](#).
5. Click **Upstream settings**.

A dialog will appear for managing the upstreams lists.



6. In the top part of the dialog, check the **Use TLS** checkbox if you need to use the encrypted HTTPS protocol for connecting to the upstreams.
7. Configure the upstreams for the service. You can organize independent [primary and backup lists of upstreams](#). Note that there must always be at least one upstream in the primary list — you won't be able to apply the changes otherwise.

To add an upstream, click **Add upstream** next to a list. In a dialog that appears, specify the IP address, the port and, optionally, the name for the upstream. For convenience, you can fill the form with another upstream's data: click , then click on an IP address (to copy the name and IP address without the port) or a port (to copy all the data). If you want to temporarily exclude the upstream from the balancing algorithm, uncheck the **Enabled** checkbox. To finish editing and go back to the previous dialog, click **Add**.

To move an upstream from one list to another or change its position in the list, drag and drop the corresponding card.

To edit an upstream, click



on its card, make the necessary changes and click **OK**.

To delete an upstream from a list, click



on its card.

8. Click **Save** in the service settings dialog.

If you close the dialog without clicking this button, the changes will be lost.

9. If necessary, add or edit other services before saving the changes. To save the changes, click **Save** in the notification at the bottom of the screen.

Configure a TCP Proxy service

A service of the “*TCP Proxy*” type defines parameters for the [reverse TCP proxy in the Proxy Protocol mode](#). The proxy accepts users' requests to a one port at the Qrator IP and sends them to one of the upstreams. The algorithm for switching upstreams can be configured by the Qrator Labs technical support team, see [Switching between upstreams](#).

To add or edit a service of the “*TCP Proxy*” type:

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **Network** → **Services**.
3. Click **Add** for configuring a new service or click next to a previously added service to edit it.

When adding a new service, specify the “*TCP Proxy*” type.

4. Specify the general settings for the service.
 - **Port** — port number to which a user connects.
 - **Whitelist-only access** — reset connections from IP addresses not included in the [whitelist](#).
5. Click **Upstream settings**.

A dialog will appear for managing the upstreams lists.

The screenshot shows a configuration dialog with two main sections: **Primary** and **Backup**. Each section has a blue button labeled "+ ADD UPSTREAM" in the top right corner. Below each section header is a card representing an upstream. The **Primary** card shows: Name: Proxy Protocol Backend, IP address: 1.1.1.150, Port: 3000, and a status of Enabled. The **Backup** card shows: Name: Proxy Protocol Backend, IP address: 1.1.1.151, Port: 3000, and a status of Enabled. Each card has a blue edit icon (pencil) in the top right and a red delete icon (trash can) in the bottom right.

6. Configure the upstreams for the service. You can organize independent [primary and backup lists of upstreams](#). Note that there must always be at least one upstream in the primary list — you won't be able to apply the changes otherwise.

To add an upstream, click **Add upstream** next to a list. In a dialog that appears, specify the IP address, the port and, optionally, the name for the upstream. For convenience, you can fill the form with another upstream's data: click , then click on an IP address (to copy the name and IP address without the port) or a port (to copy all the data). If you want to temporarily exclude the upstream from the balancing algorithm, uncheck the **Enabled** checkbox. To finish editing and go back to the previous dialog, click **Add**.

To move an upstream from one list to another or change its position in the list, drag and drop the corresponding card.

To edit an upstream, click



on its card, make the necessary changes and click **OK**.

To delete an upstream from a list, click



on its card.

7. Click **Save** in the service settings dialog.

If you close the dialog without clicking this button, the changes will be lost.

8. If necessary, add or edit other services before saving the changes. To save the changes, click **Save** in the notification at the bottom of the screen.

Configure a TCP service

A service of the “TCP” type defines port forwarding for [tunneling](#).

To add or edit a service of the “TCP” type:

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **Network** → **Services**.
3. Click **Add** for configuring a new service or click  next to a previously added service to edit it.

When adding a new service, specify the “TCP” type.

4. Specify the settings for the service.
 - **Port** — port number to which a user connects.
 - **Upstream** — the IP address and port number for connecting to the upstream.
 - **Upstream** — the IP address and port number for connecting to the upstream.
 - **Whitelist-only access** — reset connections from IP addresses not included in the [whitelist](#).
5. Click **Save** in the service settings dialog.

If you close the dialog without clicking this button, the changes will be lost.

6. If necessary, add or edit other services before saving the changes. To save the changes, click **Save** in the notification at the bottom of the screen.

Delete a service

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **Network** → **Services**.
3. Find a service and click  next to it.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

Personal dashboard: IP addresses



NOTE

This dashboard section can be accessed only by users with *Admin* or *Technical role*. Settings in this section can be edited only by users with *Technical role*.

In the “IP addresses” section of the dashboard, you can view and edit the list of IP addresses, to which the Qrator network should redirect this service's traffic.

View information about the IP addresses

1. In the **Services** section of the dashboard, select the required service.
2. Select **IP addresses**.

The page will display a list of the service's IP addresses.

Click  above the list to sort the addresses, ascending or descending.

Click  next to an IP address to go to information about it on the [Qrator Radar](#) website.

Edit the list of IP addresses

1. In the **Services** section of the dashboard, select the required service.
2. Select **IP addresses**.
3. In the top right corner, click the **Edit** button.
4. Specify the IP addresses where the service should be available. Each IP address entered must belong to one of the confirmed prefixes in “[Prefixes](#)”.

You can specify addresses in two ways:

- **Manually enter IP addresses.**

Enter one or more IP addresses in the **IP Addresses** field.

- **Determine IP addresses using DNS.**

Click the **DNS lookup** link. In the dialog box that appears, enter the **Hostname** on which the required upstream is currently available, and click **Find**. The list of addresses from the A record of the domain

will be automatically loaded into the dialog box. If necessary, uncheck the addresses that shouldn't be upstream in the new configuration. Then click **Import**.

5. Click **Save**.

Personal dashboard: Ports and protocols



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the “Ports and protocols” section of the personal dashboard, you can specify the directions and volume of legitimate traffic that is expected for the IP addresses of the current [service](#).

How it works

The data in this section is entered as a list of entries. Each entry (except [IP service → ANY](#)) contains information about one network protocol allowed. Depending on the protocol, the entry may also contain additional settings specifying the port allowed for that protocol and the traffic limit applied to it.

All inbound and outbound traffic that is not described in any entry for any of the services will be considered illegitimate and discarded when passing through Qrator Labs data scrubbing centers. See [Connection: BGP](#) for more information about BGP.

The following entry types are available:

- **Application → DNS**

Add such an entry if your DNS server, whose traffic you're protecting with [Secure DNS](#) from Qrator Labs, is running on the specified port. Inbound traffic to the specified port from secondary or caching DNS servers of Qrator Labs as well as response outbound traffic to the specified port will be considered legitimate. Note that adding this entry does not automatically activate the Secure DNS service. Contact Qrator Labs tech support to configure the service correctly.



TIP

If you're not going to use Qrator Labs Secure DNS, but want the traffic to your DNS server to be treated as legitimate, you can, instead of this entry, just add [IP service → TCP](#) or [IP service → UDP](#) with the required port (usually 53).

- **Application → HTTP**

Add such an entry if there is a web server running HTTP or HTTPS on the specified port. Traffic will be analyzed to detect attacks specific to these protocols as described in [HTTP reverse proxy](#).

For this entry, an additional setting **Secure upstream connection** is available. If the web server on the specified port works on HTTPS, enable this setting. Otherwise, leave it disabled.

- **IP service** → (protocol)

Add an entry specifying the particular protocol from the **IP service** section to allow inbound and outbound traffic using the selected protocol. The traffic protocol is determined by headers in the IP packets. All protocols described in [Assigned Internet Protocol Numbers](#) are available.

The **Whitelist-only access** checkbox means that Qrator Labs will reset connections from IP addresses not included in the [whitelist](#).

For any protocol other than TCP, an additional setting — **Speed limit** — is available. The specified limit applies separately to inbound and outbound UDP traffic on the specified port of each IP address of the service. All traffic exceeding the limit will be discarded even if the user IP address is in the [whitelist](#).

For the TCP protocol, the **Speed limit** setting is unavailable, since by default the filtering system only allows the legitimate TCP connections to upstream.

For **IP service** → **TCP**, three modes of processing the inbound traffic are available:

- **Any ports** — open all ports for both inbound and outbound TCP traffic.
- **Specify port** — open one specified port for inbound TCP traffic and allow outbound traffic in response to the legitimate inbound TCP connections.
- **Restricted** — in this mode, the **Any ports** checkbox in the **Outbound traffic** section becomes available: if enabled, Qrator Labs will allow only outbound TCP traffic legitimate on all ports (but not inbound traffic).

For **IP service** → **UDP**, two modes of processing the inbound traffic are available:

- **Any ports** — open all ports for both inbound and outbound UDP traffic.
- **Specify port** — open one specified port for inbound UDP traffic

Additionally for **IP service** → **UDP**, the **Block amplification attacks** settings is available. If it is enabled, Qrator Labs will consider illegitimate all traffic on the ports vulnerable to [Amplification DDoS attacks](#). In particular, this will close the 53 and 123 ports, commonly used for Amplification attacks via the DNS and NTP protocols, respectively. If you expect legitimate traffic on these ports, i.e., your DNS and NTP servers listen to them, we recommend to describe them separately, without disabling the amplification attacks protection for all other UDP traffic.

- **IP service** → **ANY**

Add such an entry to allow a certain amount of inbound and outbound traffic not allowed by other entries. The **Speed limit** setting is available for this entry; it specifies how much unidentified traffic is considered acceptable. The same limit applies separately to unidentified inbound and outbound traffic.

This entry is always applied last. So, only traffic that has not been described by any other entry falls into this category.

**WARNING**

When describing the traffic that should be allowed for your service, it's very easy to forget about some infrequent type of inbound or outbound requests and thus accidentally block some service functionality. We recommend creating an ANY entry to compensate for such an error.

Most often, it makes sense to create an ANY entry with a small limit, based on the assumption that in reality the volume of traffic not described by other entries will be zero. After that, guided by logs and [service statistics](#), add new entries if necessary, describing traffic not previously provided for.

- **Other options** → **Fragmented IP packets**

Add such an entry to allow delivery of fragmented IP packets. This option may be necessary, for example, for some SIP protocols.

For this entry, the **Speed limit** setting is available, specifying exactly how many fragmented packets are considered acceptable. The specified limit applies separately to inbound and outbound traffic.

View list of entries

To view a list of entries:

1. In the **Services** section of the dashboard, select the required service.
2. Select **Ports and protocols**.

The page will display a list of entries, each describing a specific type of traffic allowed; see [How it works](#).

For each entry, the following is displayed:

- Direction of allowed traffic:



- – User-initiated traffic is allowed



- - Upstream-initiated traffic is allowed



- - Both types of traffic are allowed

- Port and protocol of allowed traffic:

- For entries from the **Application** category, the name of the allowed protocol and port number are displayed.

- For entries from the **IP service** and **Other options** categories, the name of the allowed protocol is displayed.
- Speed limit (only for entries for which this setting is available)
- Upstream connection type (only for [Application](#) → [HTTP](#) entries):
 -  — Unencrypted HTTP traffic is allowed
 -  — Encrypted HTTPS traffic is allowed

To quickly find entries, use the filter field above the list. You can enter the following details in the filter:

- Protocol name, e.g. "TCP" or "DNS"
- Open port number, e.g. "80" or "53", or the "ANY" string
- IP number, e.g. "47" for GRE
- Traffic direction, e.g. "input", "output"

Add entry

To add a new entry describing legitimate traffic, take the following steps:

1. Go to [list of ports and protocols](#).
2. At the top of the page, click **Add**.
3. In the drop-down lists in the dialog box that appears, select in sequence the category and type of entry you want to add. For more information on types of entries, see [How it works](#).
4. Depending on the selected entry type, additional settings may become available in the dialog box. In this case, set the required values for the settings.
 - **Port number**

Specify the port number of the upstream on which connection will be initiated. Note that many protocols may use a different port for outbound traffic than the port on which the inbound traffic arrived. In this case, for some protocols, outbound traffic corresponding to legitimate inbound traffic also becomes legitimate. For more details on how different protocols are handled, see [How it works](#).
 - **Speed limit.**

You must set a speed limit for most types of entries. Select the unit of measure for the limit from the drop-down list: **Kbps**, **Mbps** or **Gbps** (the **Gbps** option is not available for some protocols). Then set the value by typing it into the text box or moving the slider.
 - **Secure upstream connection**

This setting is only available for [Application](#) → [HTTP](#) entries. Enabling this setting means that HTTPS traffic should be allowed, not HTTP.
5. Click **Save**.

6. To save your changes, click **Save** in the notification at the bottom of the screen.

Change speed limit

For entries that include a speed limit, you can change this limit at any time after creating the entries.

1. Go to [list of ports and protocols](#).
2. Click the speed limit you want to change.
3. In the dialog box that appears, set the desired speed limit.

Select the unit of measure for the limit in the drop-down list: **Kbps**, **Mbps** or **Gbps** (the **Gbps** option is not available for some protocols). Then set the value by typing it into the text box or moving the slider.

4. Click **Save**.
5. To save your changes, click the **Save** button in the notification at the bottom of the screen.

Delete entry

To delete a single entry:

1. Go to [list of ports and protocols](#).
2. Next to the entry you want to delete, click .
3. To save your changes, click **Save** in the notification at the bottom of the screen.

To delete multiple entries:

1. Go to [list of ports and protocols](#).
2. Mark the entries you want to delete.
3. Click the **Delete** link above the list.
4. To save your changes, click **Save** in the notification at the bottom of the screen.

Configuration example

☐	↓	DNS	PORT: 53			
☐	↓	HTTP	PORT: 80	🔒 🔒		
☐	↓	HTTP	PORT: 443	🔒 🔒		
☐	↑	TCP				
☐	↕	UDP	PORT: ANY	🔄 200 Kbps ✎	🛡️	
☐	↕	ICMP (1)		🔄 200 Kbps ✎		

In this example, the client has specified the following ports and protocols:

- [Application](#) → [DNS](#) on port 53 to support the [Secure DNS server](#) service
- [Application](#) → [HTTP](#) on ports 80 (HTTP) and 443 (HTTPS) to support the [HTTP reverse proxy](#) service
- [IP service](#) → [TCP](#) with inbound traffic disabled and outbound traffic allowed
- [IP service](#) → [UDP](#) with a speed limit of 200 Kbps and Amplification attacks blocking (will not apply to the DNS service described above)
- [IP service](#) → [ICMP](#) with a speed limit of 200 Kbps

Personal dashboard: TLS



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

If a resource has a certificate to which the current user does not have access, then the whole “TLS” section for the resources becomes available to this user in read only mode.

The “TLS” section is available for each [domain](#) or [service](#). Here you can see and change settings on how TLS certificates are used for the domain or service.

Here you can [see installed certificates](#), [view certificate details](#), and [install](#) and [uninstall certificates](#).

Note that in this dashboard section, you can only install certificates you've previously added to the system; you can't add new certificates here. For the difference between installing and adding a certificate and how to add a certificate, see [Certificate storage](#).

View installed certificates

To view a list of certificates configured for a domain or service:

1. Select the domain under **Domains** or service under **Services**.
2. Select **TLS**.

Certificates installed for the selected domain or service will be displayed as a table with the following columns:

- **ID** — Unique numeric identifier of the certificate in the Qrator Labs system. ID can be repeated in several rows of the table, if the same certificate is [installed for different domain names](#).
- **Hostname** — The host (domain name) this certificate should be used for. Can be a specific host name or a “Default Host” string.
- **Subject** — Hosts for which the certificate is valid, as well as a link to [view certificate details](#).
- **Not valid after** — Shows the end of certificate validity period. If the validity period has already expired, this field is highlighted.

The **Hostname** and **Subject** columns have filter fields in the top row. When you enter the required values there, you can filter the certificates displayed in the table.

View certificate details

To view certificate details:

1. Go to [certificate list](#) for the domain or service.
2. Find the row with the required certificate and click the **View more** link in it.

The appearing dialogue will contain the following fields:

- **ID** — the unique numerical identifier for the certificate in the Certificate storage.
- **Type** — the origin of the certificate: *Upload* or *Let's Encrypt*.
- **Subject** — the domain name(s) the certificate is valid for.
- **Not valid before, Not valid after** — the start and the end date of the certificate validity period.

The certificate may contain one or several chains. If there is more than one chain, you can switch between the chains using radio buttons with the corresponding encryption algorithm names on them. For each chain the following data is displayed::

- **Private key encryption** — the cryptographic algorithm used for the private key.
- **Private key size** — the private key size in bytes.
- **SHA-256 Fingerprint, SHA-1 Fingerprint** — the public key fingerprints.
- **Certificate chain** — the whole chain in PEM format. You can copy the full text with the  button.
- **Private key** — the private key in PEM format. You can copy the full text with the  button. This field is not displayed if [the private key is hidden](#).

Install certificate for domain or service

1. Make sure that the certificate you want to install is in the [Certificate storage](#). If there is no such certificate, [release](#) or [upload](#) it.
2. Navigate to [certificate list](#) for the domain or service.
3. Click **Install certificate**.
4. In the dialog box that appears, find the certificate you want to install.

You can use the  button to reveal a more detailed description of a particular certificate. The description includes fields similar to those in the [brief information](#) dialog box in the Certificate storage.

On the row corresponding to the certificate, click **Install**.

5. Specify which domain names the certificate should be used for.
 - If the certificate should be used for one or several known domain names, list them in the text area. Add a comma, a whitespace or use a new line after adding each domain name.
 - If you want to add the certificate before specifying any domain name, or add a domain name which doesn't have any other certificates installed for it, enable **Use as default for any hostnames** option.

Press **Install**.

**WARNING**

If a domain or a service has at least one certificate installed for any specific hostname it should also have a certificate for a default hostname installed.

The changes you made in the configuration will be displayed in the table, but will not be saved automatically.

6. To save your changes, click **Save** in the notification at the bottom of the screen.

Uninstall certificate for domain or service

To uninstall a previously installed certificate:

1. Navigate to [certificate list](#) for the domain or service.
2. Find the certificate you want to uninstall and in the corresponding row in the table, click  .

The changes you made in the configuration will be displayed in the table, but will not be saved automatically.

3. To save your changes, click **Save** in the notification at the bottom of the screen.

The certificate will no longer be used when servicing this domain or service, but will remain available in the [Certificate storage](#). To remove a certificate from the system completely, use the following instructions:
[Personal dashboard: Certificate Storage](#).

Personal dashboard: IP lists



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups.

In this section, you can manage the lists of IP addresses affecting DDoS attack protection for [domain](#) or [services](#).

- Traffic from any *whitelist* IP address is considered legitimate, even when it shows signs of attacks.
- Traffic from any *blacklist* IP address is considered illegitimate, even when it shows no signs of attacks.

View list

To view a whitelist or blacklist for a domain or service:

1. Select the domain in **Domains** or service in **Services**.
2. Select **Network** → **IP lists**.
3. Select one of the tabs: **Whitelist** or **Blacklist**.

The page will show the IP addresses added to the selected list and the comments entered for each of them.

To check for one or more IP addresses in a list, use the filter above the **IP address** column. In this filter field, you can enter both a specific IP address and a subnet in CIDR format. For example, if you enter the subnet `10.1.1.0/24` the table will display all IP addresses in the list from `10.1.1.1` to `10.1.1.254`.

Add an IP address to a list

1. [Go to the list you want](#).
2. Click **Whitelist IP addresses** (or **Block IP addresses** for blacklist).
3. In the dialog box that appears, fill in the following fields:
 - **IP addresses** — One or more IP addresses to be added to the list. If you are adding multiple IP addresses, enter each on a new line.
 - **Comment** (optional) — A text comment for an IP address. You can use this field to distinguish between IP addresses listed for different reasons.

Click **Add**.

**NOTE**

If among the entered IP addresses there are those that are already in the list, their old comments will be replaced by new ones.

Edit a comment for an IP address in the list

To edit a comment for a single IP address:

1. [Go to the list you want.](#)
2. Find the IP address you want and click it.
3. In the dialog box that appears, edit the comment. Click **Save**.

To edit comments for multiple IP addresses at once:

1. [Go to the list you want.](#)
2. Check the boxes next to the IP addresses you want. Click **Set comment**.
3. In the dialog box that appears, edit the comment. Click **Save**.

Delete an IP address from a list

To delete a single IP address:

1. [Go to the list you want.](#)
2. Find the IP address you want and click the  icon on the corresponding row.

To delete multiple IP addresses at once:

1. [Go to the list you want.](#)
2. Check the boxes next to the IP addresses you want. Click **Delete**.

Configuration examples

Whitelist

WHITE LIST **BLACK LIST**

List of trusted IP addresses that are allowed to access, regardless of the presence of signs of attacks in requests. This whitelist is fully controlled by customer [ADD IP ADDRESSES](#)

☐ **Whitelist-only access**
Limit the access to the IP-addresses from whitelist only. Any other IP-address will be blocked

☐	IP address	Comment	
	<u>Find IP address or subnet</u>		
☐	203.0.113.1	office computer	
☐	203.0.113.2	office computer	
☐	203.0.113.3	office computer	
☐	233.252.0.70	personal device	
☐	233.252.0.80	personal device	

Blacklist

WHITE LIST **BLACK LIST**

Add IPs to blacklist to block requests from these addresses. This blacklist is fully controlled by customer [ADD IP ADDRESSES](#)

☐	IP address	Comment	
	<u>Find IP address or subnet</u>		
☐	198.51.100.100	known bot #1	
☐	198.51.100.200	known bot #2	

Personal dashboard: Tunnels



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the “Tunnels” section of the dashboard, you can manage the settings related to [Qrator Labs connection via a tunnel](#).

View tunnels list

To view the tunnels list configured for a domain:

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **Tunnels**.

The tunnels will be displayed as a table with the following columns:

- **Tunnel**

The tunnel mode (*IPIP* or *GRE*) and its unique numeric identifier in Qrator Labs.

If [auto tunnel switching is not enabled](#) for that domain and this tunnel is active, the Tunnel column will display “Active”.

- **External IP addresses**

Real-life addresses where endpoints are available on the Internet.

- **Internal IP addresses.**

Addresses assigned to endpoints within a virtual network.

Add tunnel

1. Go to [tunnel list](#) and click **Add tunnel**.
2. In the dialog box that appears, specify:
 - **Mode** — tunnel mode: *IPIP* or *GRE*,
 - **External IP address** — real address on your side that will send and receive packets.

Click **Create**.

Edit tunnel

1. Go to [tunnel list](#).
2. Place the cursor on the row with the tunnel you want to edit and click the **Edit** link next to its identifier.
3. In the dialog box that appears, edit the following fields:
 - **Mode** — tunnel mode: *IP/IP* or *GRE*,
 - **External IP address** — real address on your side that will send and receive packets.

Click **Save**.

Select active tunnel

You can select an active tunnel only if [auto tunnel switching is not enabled](#) for the domain.

1. Go to [tunnels list](#).
2. Place the cursor on the row with the tunnel you want to make active and click the **Make active** link next to its ID.
3. In the dialog box that appears, click **Confirm**.

Delete tunnel

1. Go to [tunnels list](#).
2. Place the cursor on the row with the tunnel you want to delete and click the **Delete** link next to its ID.
3. In the dialog box that appears, click **Delete** to confirm the action.

Configure auto tunnel switching

By default, Qrator Labs directs traffic only to the [active tunnel](#), but you can enable automatic tunnel switching. In this case, each [data scrubbing center](#) will perform periodic checks on the availability of the active tunnel and, if the check fails, the center will switch to the next tunnel in the list.

To enable auto tunnel switching or change the check settings:

1. Go to [tunnel list](#).
2. Expand the block **Balancing** and click **Edit**.
3. Turn on the **Enabled** switch.

4. Set the check parameters:

- **Check type** - The way in which the data scrubbing center will send a request to the tunnel's IP address: *Ping* or *HTTP*.
- **Timeout** - The number of milliseconds to await a response. No response within this time period is considered an error.
- **Number of attempts** - How many last checks are used to analyze tunnel performance.
- **Successful attempts** - Minimum number of successful attempts from the last **Number of attempts** at which the tunnel is considered to be working correctly.

When selecting the *HTTP* check type, you must also specify the following additional parameters:

- **Hostname** — value for the `Host` header.
- **Path** — path to which HTTP request will be made.
- **Use HTTPS** — if enabled, HTTPS will be used instead of HTTP.

5. Click **Save**.

To disable auto tunnel switching:

1. Go to [tunnels list](#).
2. Expand the block **Balancing** and click **Edit..**
3. Turn off the **Enabled** switch.
4. Click **Save**.

Example

ⓘ Balancer HTTP 172.16.133.165/healthcheck (example.com) Attempts: 2 of 3 Timeout: 5000ms ▼				
+ ADD TUNNEL				
Tunnel	External IP-addresses		Internal IP-addresses	
	Client	Qrator	Client	Qrator
IPIP (ID 800)	198.51.100.101	178.248.233.165	172.16.133.165	172.16.233.165
IPIP (ID 796)	198.51.100.102	178.248.233.165	172.16.133.165	172.16.233.165
GRE (ID 804)	198.51.100.103	178.248.233.165	172.16.133.165	172.16.233.165

In this example, three tunnels are configured: two over IPIP and one over GRE. They have different external IP addresses but use the same internal IP address 172.16.133.165 assigned to them by Qrator Labs. They should

use external IP address 178.248.233.165 ("Qrator IP") and internal IP address 172.16.233.165 to communicate with Qrator Labs.

The client has chosen to configure [auto tunnel switching](#), which is displayed in the **Balancing** block. To check if each of the tunnels is working, the Qrator Labs data scrubbing center makes HTTP request [example.com/healthcheck](#) through them, using internal IP address 172.16.133.165. The data scrubbing center expects to get a successful response within 3 seconds. If two of the three consecutive checks fail, the data scrubbing center switches to another tunnel.

Dashboard: Filter configuration

In the “Filter configuration” of the Qrator Labs personal dashboard, you can configure filtering requests to the selected [domain](#) or [service](#). Filtering can be performed based on [HTTP headers](#), [countries of request origin](#), or [user defined rules](#).

Filter by HTTP headers

The filters allow to automatically form dynamic black and white lists of users' IP addresses based on the [User-Agent](#) and [Referer](#) headers. These dynamic lists control how the Qrator Labs network will process future requests from the same IP addresses, even when the IP addresses are not added manually to [IP lists](#).

Each rule is checked independently of other rules. If at least one rule is triggered for the white list, the requests from the user are considered legitimate. If at least one rule is triggered for the black list, all requests from the user after the current request are considered illegitimate. Other cases are processed by the Qrator Labs network as usual.

The white list rules have higher priority than the black list rules.

Add a rule

1. Select the domain in **Domains** or service in **Services**.
2. Select **Filter configuration**.
3. Next to the **HTTP Headers** section title, enable the checkbox if it is not yet enabled.
4. Next to a list of rules, click **Add rule**.
5. Fill out the following fields:
 - **Header** to test: [User-Agent](#) or [Referer](#) .
 - The test method:
 - **=** (“equals”) — the header must have the exact value entered in the text field.
 - **Contains** — the header value must include the substring entered in the text field.
 - **Is Empty** — the header must not be specified.
 - The string used for the test.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Delete a rule

1. Select the domain in **Domains** or service in **Services**.
2. Select **Filter configuration**.
3. In the **HTTP Headers** section, find the rule you want to delete and click  next to the rule.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Disable filtering by headers

1. Select the domain in **Domains** or service in **Services**.
2. Select **Filter configuration**.
3. Next to the **HTTP Headers** section title, disable the checkbox.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Filter by country

This filter allows you to limit access to the resource for users from specific countries. The source country of the request is determined by the IP address using the [MaxMind GeoIP] database (<https://www.maxmind.com/en/geoip2-services-and-databases>).

Note that the geo-filter is ignored when the user's IP address is in the [black or white IP list](#).

Enable filtering by country

1. Select the domain in **Domains** or service in **Services**.
2. Select **Filter configuration**.
3. Next to the **Geo-filters** section title, enable the checkbox if it is not yet enabled.
4. In the dropdown list, select the filtering mode:
 - **Allow requests** only from the specified countries.
 - **Deny requests** from the specified countries.

In both cases, the first request from a user will be considered legitimate, but after it, the IP address will be blacklisted if it does not match the condition. If the reverse HTTPS proxy is used without key disclosure, the IP address will be blacklisted after [receiving the logs](#) for the first request, which may cause an additional delay.

5. Edit the list of countries. The countries in the list are represented by codes according to [ISO 3166] standard (https://en.wikipedia.org/wiki/ISO_3166), for example, the code "FR" is used for France.

- To add a country to the list, click its code in the dropdown list. To search for the desired country, type one or both of the letters of its code in the text field.
 - To delete a country from the list, click the X mark next to its code.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Disable filtering by country

1. Select the domain in **Domains** or service in **Services**.
2. Select **Filter configuration**.
3. Next to the **Geo-filters** section title, disable the checkbox.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

User defined rules

In the **User defined rules** section, you can configure conditions for blacklisting or whitelisting IP addresses based on a large amount of properties from both requests and responses (see the list below). You can also combine the conditions into more complex rules using the **AND**, **OR**, and **NOT** logical operations.

The conditions are being checked at the moment of sending the response from the upstream to the user. If the request and the response match the conditions, the selected action will be applied to the future requests from the same IP address.

1. Select the domain in **Domains** or service in **Services**.
2. Select **Filter configuration**.
3. Next to the **User defined rules** section title, enable the checkbox if it is not yet enabled.
4. Configure the list of rules.

To add a rule, click **Add rule** at the bottom of the page. To delete a rule, click  next to the rule.

5. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Available actions for rules are:

- **Whitelist** — send the response to the current request, ignoring other filters and the black list.
- **Blacklist** — send the response to the current request, but blacklist the IP address for the future requests.
- **RateLimit** — limit the requests frequency according to the [Leaky bucket](#) algorithm. When this action is selected, you can specify the **Rate** and **Burst** values for the algorithm.

Available condition types:

- nested condition groups:
 - **AND** — the condition is matched when all the nested conditions are matched.
 - **OR** — the condition is matched when at least one nested condition is matched.
 - **NOT** — the condition is matched when the nested condition is not matched.
- request property tests:
 - **Method** — the request's HTTP method.
 - **Path** — the path to which the request was made.
 - **Query** — the string containing the GET arguments.
 - **Request Uri** — the full request URI. Includes the protocol, **Host**, **Path**, and **Query**.
 - **Host** — the host to which the request was made.
 - **Header** — the value of an arbitrary HTTP header from the request. The header name is entered in the first text field.
 - **Cookie** — the value of an arbitrary cookie from the request. The cookie name is entered in the first text field.
 - **User Agent** — the value of the `User-Agent` header from the request.
 - **Referer** — the value of the `Referer` header from the request.
 - **Content Length** — the value of the `Content-Length` header from the request.
 - **Geo** — the source country of the request. You can select multiple countries in the dropdown list. The source country of the request is determined by the IP address using the [MaxMind GeoIP] database (<https://www.maxmind.com/en/geoip2-services-and-databases>).
- response property tests:
 - **Status** — the response's HTTP status code.
 - **Duration** — the time spent waiting for the upstream response.
 - **Response Header** — the value of a HTTP header from the response. Only the `Location` and `Content-Type` headers are supported. The header is selected in the first field.

Different operations are available for different properties. For example, for any text property, you can select `!=` (test for an exact match), `startsWith` (test for a prefix match), etc. For a numeric property, you can select `>` (strictly greater than), `>=` (greater than or equal), etc. For properties related to HTTP headers and Cookies, the operations include `exist` and `notExists` which test the existence of the header.

Most operations require parameters, such as the string to compare to. All parameters must be filled.

Configuration example

The screenshot displays the Qrator Labs dashboard's filter configuration interface. It is organized into three main sections: HTTP-headers, GEO-FILTERS, and USER DEFINED RULES.

- HTTP-headers:** This section is divided into 'Deny rules' and 'Whitelist rules'.
 - Deny rules:** A rule is configured where the 'Referer' header 'CONTAINS' the substring 'https://example.com/'.
 - Whitelist rules:** A rule is configured where the 'User-Agent' header 'equals' the value 'CustomTestTool'.
- GEO-FILTERS:** The 'Allow requests' section is set to 'only from' the country 'RU' (Russia). A note states: 'All requests from other countries will be rejected by Qrator'. A link 'Copy countries' is provided.
- USER DEFINED RULES:** A single rule is listed, which is 'Enabled'.
 - Action:** 'RateLimit'.
 - Rate, rps:** A slider is set to 1.
 - Burst:** A slider is set to 5.
 - Conditions:** The rule is defined by an 'AND' combination of three conditions:
 - 'USER-AGENT' 'notExist'.
 - 'PATH' 'is' '/'. (Note: the image shows '!'=, likely a typo for 'is')
 - 'DURATION' '>' '1000' ms.

In this example:

- The client detected a big amount of automated requests that had the substring `https://example.com/` in their `Referer` header, so they decided to automatically blacklist all IP addresses that send such requests.
- The client added a rule for automatic whitelisting all IP addresses from which they get requests with the value `CustomTestTool` in the `User-Agent` header. For example, if the client configures their testing tools so that they use this header, it will prevent false positives when Qrator Labs will analyze the test requests.
- The client only works with users from Russia, so the IP addresses from other countries are automatically blacklisted.
- The client decided to blacklist IP addresses if they sent requests to the main page without the `User-Agent` requests and processing them took more than 1 second.

Personal dashboard: HTTP redirects



NOTE

This dashboard section can be accessed only by users with *Technical role*.

In this dashboard section, you can manage the rules guiding how [reverse HTTP proxy](#) redirects part of a user's HTTP requests.

For details, see [Technologies: HTTP redirects](#).

View list of rules

To see the list of rules configured for a particular domain:

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **HTTP redirects**.

The rules are displayed in a table with the following columns:

- **Request** - URL pattern for which this rule applies.

If the rule uses a specific host, then the request in this column is displayed in the format

`<protocol>://<host>`, e.g. `http://example.com`.

If the rule should work for any host, the “ANY” value is displayed instead of the host, e.g. `http://ANY`.

- **Redirect** – HTTP response code, as well as URL to which the redirect is made.

The response code and URL are displayed on the same line separated by a space, e.g. `301`

`http://example.com/`.

If the **Transmit as is** options were enabled for some request components (see [Add rule](#), the string will be displayed with special variables whose values will be substituted when processing each specific request:

- `$hostname` - Host from user's request,
- `$path` - Path from user's request,
- `$args` - GET arguments from user's request.

For example: `301 http://$hostname$path?$args`.

If redirection is disabled within this rule, this column will show **Disabled**.

Add rule

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **HTTP redirects**.
3. Click **Add**.
4. In the **Request** section, specify the request parameters for which the rule will be triggered.
 - **Scheme** — request protocol (*HTTP* or *HTTPS*).
 - **Hostname** — the domain name to which the request is made. If the “FQDN” option is selected, then in the text field below you must enter a specific domain name for which the requests will be checked. If the “ANY” option is selected, the rule will not depend on the domain name in the request.
5. In the **Redirect** section, specify the actions that the reverse HTTP proxy should perform over the request.
 - **Enabled** - enable request processing. If this option is disabled, requests will be sent to upstream without any changes.
 - **Scheme** - request protocol (*HTTP* or *HTTPS*).
 - **Response code** - HTTP response code with which the user will be redirected. You can choose from the following codes:
 - 301 Moved Permanently
 - 302 Found
 - 307 Temporary Redirect
 - 308 Permanent Redirect
 - **Hostname** — the domain name that will be used in the URL for redirection.

To use the domain name from the original request unchanged each time, set the **Transmit as is** option.
 - **Path** — the path that will be used in URL redirection.

To use the path from the original request unchanged each time, set the **Transmit as is** option.
 - **Request parameters** — whether to use GET parameters from the original request.

To use GET parameters from the original request unchanged each time, set the **Transmit as is** option. If the option is not set, all GET parameters will be removed from the request.
6. Click **Save**.
7. To save your changes, click **Save** in the notification at the bottom of the screen.

Edit rule

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **HTTP redirects**.
3. Next to the rule you want to edit, click **Edit**.
4. Make the necessary changes to the rule.

Fields in the rule editing form are identical to the fields in the [add rule form](#).
5. Click **Save**.
6. To save your changes, click **Save** in the notification at the bottom of the screen.

Delete rule

1. In the **Domains** section of the dashboard, select the particular domain.
2. Select **HTTP redirects**.
3. Next to the rule you want to edit, click **Delete**.
4. Confirm the action by clicking **Delete** in the dialog box that appears.
5. To save your changes, click **Save** in the notification at the bottom of the screen.

Configuration example

Request	Redirect		
http:// ANY	301 https:// \$hostname \$path ? \$args		
http://example.com	301 https://www.example.com \$path ? \$args		
https://example.com	301 https://www.example.com \$path ? \$args		
http://old.example.com	Disabled		

In the above example, the reverse HTTP proxy is used to protect a site using the `example.com` host and its subdomains. It is assumed that all sections of the site, except for the section on the `old.example.com` subdomain, support HTTPS.

The reverse HTTP proxy will execute the rules as follows:

- When working with HTTP:
 - In requests to `example.com`, the protocol is replaced by HTTPS and the domain is replaced by `www.example.com` with the path and arguments preserved (rule 2).
 - Requests to `old.example.com` are sent to the upstream unchanged (rule 4).

- In other requests, the protocol is replaced by HTTPS, keeping the domain, path and arguments (rule 1).
- When working with HTTPS:
 - In requests to `example.com`, the domain is replaced by `www.example.com` with the path and arguments preserved (rule 3).
 - Other requests are sent to the upstream unchanged (implicit rule by default).

**NOTE**

Other redirect configuration examples for typical scenarios can be found in [Examples](#) in “Technologies: HTTP redirects”.

Personal dashboard: Error pages



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the “Error pages” section of the dashboard, you can upload your own pages, which the [Qrator Labs reverse HTTP proxy](#) will display when errors 403 Forbidden, 502 Bad Gateway, 503 Service Unavailable or 504 Gateway Timeout occur. All settings are valid only for the current [domain](#) or [service](#).

How it works

For the reverse HTTP proxy to use a page as an error page, it must be uploaded as a file with the `html` extension.

To show the same error page for all error types, use the `index.html` filename. To show unique pages for different error types, use the error code in the filename, for example: `502.html`.

If you need to show localized error pages to users that prefer different languages, specify the language in the filename, for example: `502.en.html`. The language will be detected based on the `Accept-Language` header from the user's browser, for example: `en`, `ru`, `es`.

If additional images or other files are needed to display the pages, you can upload them in this section, too. When including the files into the error page code, start the URLs with `/qrerror/`, for example: `/qrerror/favicon.ico`, `/qrerror/images/logo.png`. Resources located on other URLs may be unavailable when the error occurs.

Resources that are executed in the user's browser (JavaScript code, Flash or Silverlight elements) can be used. Resources that are executed on the server side is not supported for error pages.

Note that the uploaded error pages are only shown when the reverse HTTP proxy yields an error by itself (see [Redirection with cookie installation](#), [Load balancing](#)). Errors yielded by the protected server are shown without modification.

**EXAMPLE**

Let's assume that all upstreams failed to respond to a request, and, following the error processing algorithm, the user now needs to be shown the 504 Gateway Timeout error message. Let's also assume that the user has English language selected in their browser.

In this case, the reverse HTTP proxy will go through the options in the following order:

`504.en.html`, `504.html`, `index.en.html`, `index.html`, the default page. The user will be shown the first page it finds.

View error pages

1. Select the domain in **Domains** or service in **Services**.
2. Select **Error pages**.
3. Click on the URL next to one of error codes.

A page will open in a new browser tab that users will see when the corresponding error occurs.

Replace error pages

1. Select the domain in **Domains** or service in **Services**.
2. Select **Error pages**.
3. Click **Edit**.
4. In the file management dialog box that appears, create the necessary file structure.

All pages must be in the root folder, and their names must follow the rules described above in [How it works](#). If additional images or other files are needed to display the pages, you can upload them in this dialog, too.

The following buttons are available in the file management dialog box:

- **Create folder** — Create a new folder in the current folder.
- **Upload files** — Upload one or more files into the current folder. Files to be uploaded are selected using the system dialog box.
- **Upload archive** — Upload a Zip archive and unpack it to the root folder. The archive to be uploaded is selected using the system dialog box. Only Zip archives are supported.
- **Download all** — Download Zip archive with the full contents of the root folder.
- **Delete all** — Delete all the contents of the root folder permanently.

Click any file name to open its preview in a new tab, or click the button to copy the preview link. Note that preview by this link is available only after logging into your account.

There is also a menu next to each file with the following items:

-  **Download** - Download the selected file.
-  **Delete** - Delete the selected file permanently.

All changes made in this dialog box are saved automatically and become visible to users after a few minutes. To close the dialog box, click **Apply**.

Configuration example

			    
File name	File size	Last modified	
 images		2022-08-08 09:14	
 502.html 	306.00 bytes	2022-08-08 09:14	
 503.html 	200.00 bytes	2022-08-08 09:14	
 504.html 	201.00 bytes	2023-03-24 17:57	
Root			
CANCEL			APPLY

In this example, three error pages are loaded. Also loaded is the `images` folder with the images needed for these pages.

Personal dashboard: Notifications (resource configuration)



NOTE

This page describes notification settings for a resource. To learn about configuring receiving notification for a certain user account, see [Notifications \(profile configuration\)](#).

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the “Notifications” section of the dashboard, you can configure thresholds which, when exceeded, the client receives an email notification. All settings are valid only for the current [domain](#) or [service](#).

The following notifications are supported:

- **Inbound traffic** — Total inbound bandwidth (before filtering).
- **Outbound traffic** — Total outbound bandwidth.
- **Missed traffic** — Bandwidth of inbound traffic that was passed to the client.
- **Blocked IP addresses** — Number of blocked IP addresses in the Qrator Labs automatic blacklists at the same time.

A notification is sent if the average value of any of the indicators over the last minute exceeds the specified threshold.

Enable notifications or change the notification threshold for an indicator

1. Select the domain under **Domains** or service under **Services**.
2. Select **Notifications**.
3. Click **Edit**.
4. Find the indicator you want and edit the numeric value of the threshold by entering from your keyboard or using the slider.
 - In the **Inbound**, **Outbound** and **Missed traffic** categories, the default unit of measurement is Mbps. To enter values greater than 999 Mbps, you must use the Gbps unit of measurement. To switch between Mbit/s and Gbit/s, click on the unit of measure.
 - In the **Blocked IP addresses** category, the unit of measure is one IP address.

5. Click **Save**.

Disable notifications for a category

1. Select the domain under **Domains** or service under **Services**.
2. Select **Notifications**.
3. Click **Edit**.
4. Find the indicator you want and put 0 as its threshold value.

The  image will appear next to the category.

5. Click **Save**.

Configuration example



Category	Threshold Value	Unit	Notifications
Input traffic	2	Gbps	Enabled
Output traffic	500	Mbps	Enabled
Passed traffic	0		Disabled (Bell icon)
Banned IPs	2000	IPs	Enabled

In this example, the client will receive a notification in each of the following cases:

- If average traffic bandwidth over the last minute was 2 Gbps,
- If average outbound bandwidth over the last minute was 500 Mbit/s,
- If the automatic blacklist over the last minute had an average of 2000 IP addresses.

Notifications on the bandwidth of traffic passed to the customer's network are disabled in this example.

Personal dashboard: Bot protection



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups.

The section “Bot protection → Dashboard” personal dashboard Qrator Labs displays data on blocked requests to the current [domain](#) or [service](#).

The section consists of several widgets:

- [events graph](#),
- [widget with key metrics](#),
- [widgets with data on blocked requests](#).

You can upload data [for different periods] ([#select-period](#)). By default, the data on the page is [updated automatically](#). Uploaded data can be [exported as an image or table](#).

View data

Events graph

At the top of the page there is a graph that displays events related to bot protection. Different datasets on the graph represent different types of events. You can hide a data set from the graph by clicking on its designation in the legend.



NOTE

Events corresponding to the 401 Unauthorized code are registered only after the user has made the next request. For example, the event **Check procedure failed** (401 Unauthorized) is registered simultaneously with the corresponding event **Blocked due to failed check procedure** (403 Forbidden). Hence the event data corresponding to the 401 Unauthorized code may appear on the graph with a slight delay.

Below is a list of event types displayed on the graph. The HTTP response codes sent to users are shown in parentheses.

- **User check ignored** (401 Unauthorized)

The user received a page with a JavaScript verification code, but made another request instead of executing it.

This may indicate that the user is a bot or uses a browser without the ability to execute JavaScript code.

- **User check shows a bot** (401 Unauthorized)
Browser bot blocked (403 Forbidden)

The user received a page with a JavaScript verification code, sent a digital fingerprint of the environment and was identified as a bot.

- **Check procedure failed** (401 Unauthorized)
Blocked due to failed check procedure (403 Forbidden)

The user received a page with a JavaScript verification code, sent a digital fingerprint of the environment and was identified as a legitimate user. They were given a tracking cookie. Later they turned to the address that requires a tracking cookie, but did not provide it.

This can be caused by an unexpected disconnection on the user's side, when a tracking cookie is received after verification. Most often, in a situation like that, the user can return to the verification page and successfully pass the verification again.

- **Invalid cookie/token** (403 Forbidden)

The user contacted an address that requires a tracking cookie or token, but did not provide them.

This may indicate that the user is a bot, or that the user has not visited the verification page. If you use [hash-based mobile API protection](#), this error may also occur if your mobile application sends API requests with missing or incorrect header tokens.

Key metrics

This widget contains brief information about the number of requests processed during the selected period.

- **Total blocked bot requests** — total number of illegitimate requests.
- **Requests failed to validate** — the number of illegitimate requests in which the user did not execute the JavaScript verification code issued to them before they visited the page (action *"Accept and Show Challenge"* in [request processing rules](#)).
- **Requests failed to present a cookie** — the number of illegitimate requests in which the user did not provide a tracking cookie on the page that requires it (action *"Accept only with Cookies"* in [request processing rules](#)).
- **Bad fingerprint requests** — the number of illegitimate requests in which the user executed the JavaScript verification code issued to them before they visited the page (action *"Accept and Show Challenge"* in [request processing rules](#)), and was identified as a bot.

All the indicators, except **Total blocked bot requests**, are displayed as an absolute value and as a percentage of the total number of requests. Please note that some requests may fall into more than one category, so the sum of the three values may exceed the value of **Total blocked bot requests** (100%).

Data on blocked requests

These widgets contain information on the most common values of some parameters of requests blocked during the selected period. For each of the values, the number of requests with this value is specified.

- **Top 10 countries** — countries from which requests were made. The data is displayed as a bar chart. To see the number of requests from a country, point the mouse to the corresponding column. The source country of the request is determined by the IP address using the [MaxMind GeoIP] database (<https://www.maxmind.com/en/geoip2-services-and-databases>).
- **Top 10 IP addresses** — IP addresses from which requests were made.
- **Top 10 URIs** — URI paths to which requests were made.
- **Top 10 referer** — values of the HTTP header `Referer` from requests.
- **Top 10 User agents** — values of the HTTP header `User-Agent` from requests.

**TIP**

To view more detailed information on all the blocked requests for the selected period use [data export](#).

Period selection

All widgets in the section display data for a certain period (by default **1 day**). You can change this period in one of two ways.

- Hold down the left mouse button at the beginning of the [events graph] section you are interested in (#events-graph) and, without releasing the button, pull to the end of the section. The scale of the graph will automatically change so as to include data only from the selected area. Such zoom out can be performed successively several times.

Each time the chart is updated all the other widgets in the section are automatically updated and display the data for the corresponding period.

To return to the previous period, click the button . To return to the default period, press the **Reset Zoom** button.

- Select one of the preset periods from the drop-down list at the top of the page. The necessary data will be automatically loaded and displayed on the chart and other widgets.

Data update

By default, all the data in this section of the personal dashboard is automatically updated without reloading the page every **2 minutes**. To change this interval, open the drop-down list next to the button  at the top of the page and select the desired interval. To disable auto-update completely, select the option *“Off”* in the drop-down list.

To update the data forcedly, click the button  .

Export

Export graph

To save an [events graph](#) for a selected period select the menu item **Export** → **Export graph (.png)** at the top of the page. The image will be downloaded in the same size as it was displayed in the browser at the time of saving.

Exporting requests

To download the full data on all the blocked requests for a selected period select the menu item **Export** → **Export requests (.csv.gz)** at the top of the page.



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups.

The section “Bot protection → Dashboard” personal dashboard Qrator Labs displays data on blocked requests to the current [domain](#) or [service](#).

The section consists of several widgets:

- [events graph](#),
- [widget with key metrics](#),
- [widgets with data on blocked requests](#).

You can upload data [for different periods] ([#select-period](#)). By default, the data on the page is [updated automatically](#). Uploaded data can be [exported as an image or table](#).

View data

Events graph

At the top of the page there is a graph that displays events related to bot protection. Different datasets on the graph represent different types of events. You can hide a data set from the graph by clicking on its designation in the legend.

**NOTE**

Events corresponding to the 401 Unauthorized code are registered only after the user has made the next request. For example, the event **Check procedure failed** (401 Unauthorized) is registered simultaneously with the corresponding event **Blocked due to failed check procedure** (403 Forbidden). Hence the event data corresponding to the 401 Unauthorized code may appear on the graph with a slight delay.

Below is a list of event types displayed on the graph. The HTTP response codes sent to users are shown in parentheses.

- **User check ignored** (401 Unauthorized)

The user received a page with a JavaScript verification code, but made another request instead of executing it.

This may indicate that the user is a bot or uses a browser without the ability to execute JavaScript code.

- **User check shows a bot** (401 Unauthorized)
Browser bot blocked (403 Forbidden)

The user received a page with a JavaScript verification code, sent a digital fingerprint of the environment and was identified as a bot.

- **Check procedure failed** (401 Unauthorized)
Blocked due to failed check procedure (403 Forbidden)

The user received a page with a JavaScript verification code, sent a digital fingerprint of the environment and was identified as a legitimate user. They were given a tracking cookie. Later they turned to the address that requires a tracking cookie, but did not provide it.

This can be caused by an unexpected disconnection on the user's side, when a tracking cookie is received after verification. Most often, in a situation like that, the user can return to the verification page and successfully pass the verification again.

- **Invalid cookie/token** (403 Forbidden)

The user contacted an address that requires a tracking cookie or token, but did not provide them.

This may indicate that the user is a bot, or that the user has not visited the verification page. If you use [hash-based mobile API protection](#), this error may also occur if your mobile application sends API requests with missing or incorrect header tokens.

Key metrics

This widget contains brief information about the number of requests processed during the selected period.

- **Total blocked bot requests** — total number of illegitimate requests.
- **Requests failed to validate** — the number of illegitimate requests in which the user did not execute the JavaScript verification code issued to them before they visited the page (action *"Accept and Show*

Challenge" in [request processing rules](#)).

- **Requests failed to present a cookie** — the number of illegitimate requests in which the user did not provide a tracking cookie on the page that requires it (action *"Accept only with Cookies"* in [request processing rules](#)).
- **Bad fingerprint requests** — the number of illegitimate requests in which the user executed the JavaScript verification code issued to them before they visited the page (action *"Accept and Show Challenge"* in [request processing rules](#)), and was identified as a bot.

All the indicators, except **Total blocked bot requests**, are displayed as an absolute value and as a percentage of the total number of requests. Please note that some requests may fall into more than one category, so the sum of the three values may exceed the value of **Total blocked bot requests** (100%).

Data on blocked requests

These widgets contain information on the most common values of some parameters of requests blocked during the selected period. For each of the values, the number of requests with this value is specified.

- **Top 10 countries** — countries from which requests were made. The data is displayed as a bar chart. To see the number of requests from a country, point the mouse to the corresponding column. The source country of the request is determined by the IP address using the [MaxMind GeoIP] database (<https://www.maxmind.com/en/geoip2-services-and-databases>).
- **Top 10 IP addresses** — IP addresses from which requests were made.
- **Top 10 URIs** — URI paths to which requests were made.
- **Top 10 referer** — values of the HTTP header [Referer](#) from requests.
- **Top 10 User agents** — values of the HTTP header [User-Agent](#) from requests.



TIP

To view more detailed information on all the blocked requests for the selected period use [data export](#).

Period selection

All widgets in the section display data for a certain period (by default **1 day**). You can change this period in one of two ways.

- Hold down the left mouse button at the beginning of the [events graph] section you are interested in (#events-graph) and, without releasing the button, pull to the end of the section. The scale of the graph will automatically change so as to include data only from the selected area. Such zoom out can be performed successively several times.

Each time the chart is updated all the other widgets in the section are automatically updated and display the data for the corresponding period.

To return to the previous period, click the button . To return to the default period, press the **Reset Zoom** button.

- Select one of the preset periods from the drop-down list at the top of the page. The necessary data will be automatically loaded and displayed on the chart and other widgets.

Data update

By default, all the data in this section of the personal dashboard is automatically updated without reloading the page every **2 minutes**. To change this interval, open the drop-down list next to the button  at the top of the page and select the desired interval. To disable auto-update completely, select the option *"Off"* in the drop-down list.

To update the data forcedly, click the button .

Export

Export a graph

To save an [events graph](#) for a selected period select the menu item **Export** → **Export graph (.png)** at the top of the page. The image will be downloaded in the same size as it was displayed in the browser at the time of saving.

Exporting requests

To download the full data on all the blocked requests for a selected period select the menu item **Export** → **Export requests (.csv.gz)** at the top of the page.

Personal dashboard: Bot protection → Locations



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the section “Bot protection → Locations” of the personal dashboard Qrator Labs, you can enable, disable and change the processing rules applied to requests on the current [domain](#) or [service](#).

Each rule consists of a condition according to which a request is checked, and an action that is performed if the condition is met. No more than one rule applies to a single request.

When a user request arrives at the Qrator Labs server, the server matches it to the rules in the order in which they are stored in the personal dashboard, starting from the top. Upon the first successful match, the action specified in the corresponding rule is performed. If the matches for all the rules fail, then an implicit default rule is applied with the action **Do Nothing**.

Add a host group

If your domain or service uses a large number of hosts, it may be convenient to combine some of them into host groups. [Request processing rules](#) can refer to groups of hosts instead of individual hosts, which simplifies the structure of the rules as a whole.

One host can be a member of several groups at the same time.



EXAMPLE

Let's assume that part of your site's content is located on subdomains that indicate the language used: [ru.example.com](#), [en.example.com](#), [es.example.com](#) etc. In this case, you can create a host group with the name “Language versions of the site”, which includes these subdomains, and refer to it in the rules.

If later on you add a new language and a new subdomain to the site that would be subject to the same rules, just add the subdomain to this group.

To add or change a host group:

1. Select the desired domain in the **Domains** section or the service in the **Services** section.

2. Select the menu item **Bot protection** → **Locations**.
3. In the **Host-groups** section, click on the link **Add host-group**.
4. In the dialog that opens, fill in the fields:
 - **Name** — the unique name of the group.
 - **Hostnames** — the list of hosts in the group, one host per line.
5. Click the button **Save**.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Edit a host group

1. Select the target domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Locations**.
3. In the **Host-groups** section, click the **Edit** link next to the name of the group you want to change.
4. In the dialog that opens, fill in the fields:
 - **Name** — the unique name of the group.
 - **Hostnames** — the list of hosts in a group, one host per line.
5. Click the button **Save**.
6. To save the changes, click the button **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Delete a host group

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Locations**.
3. Make sure that the host group to be deleted is not used in the request processing rules.

If necessary, [edit](#) or [delete](#) the rules referring to this host group.
4. In the **Host-groups** section, click the **Delete** link next to the name of the group you want to change.
5. Confirm the action by clicking **Delete** in the dialog that appears.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Add or change a request processing rule

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Locations**.
3. Find the rule that you want to change, or the place where you want to add a new rule.
 - To change an existing rule, click the  icon next to the rule.
 - To add a new rule, click the link **Add a rule** in the correct place. For example, to make the new rule the highest priority, click the link above the first existing rule.

The dialog with the rule settings will open.

4. Enter the name of the rule in the field **Name**. (This step can be skipped.)
5. In the **IF** section, describe the condition under which the rule is triggered. The condition consists of three parts that check different components of an HTTP request. The verification method is selected independently for each of the parts.
 - **Host** — the host to which the request is being made. For example: `example.com`.

Ways to check the host:

- **Host group** — the host must belong to a `host group` selected from the drop-down list.
 - **Any value** — the host can be any value.
 - **Equal** — the host must strictly match the string entered in the text field.
 - **Starts with** — the host must start with the string entered in the text field.
 - **Ends with** — the host must end with the string entered in the text field.
 - **Contains** — the host must contain the string entered in the text field.
- **Method** — HTTP request method. For example: `GET`.

Ways to test the method:

- **Any value** — the method can be any value.
 - **Equal** — the method must strictly match the method selected from the drop-down list.
- **Path** — the path to which the request is made. For example: `/`.

Ways to check the path:

- **Any value** — the path can be any value.
- **Equal** — the path must strictly match the string entered in the text field.
- **Starts with** — the path must start with the string entered in the text field.
- **Ends with** — the path must end with a string entered in the text field.
- **Contains** — the path must contain the string entered in the text field.

When selecting the verification methods **Equal**, **Starts with**, **Ends with** or **Contains**, an additional popup window appears with the checkbox **One of**. Check off this checkbox to specify several possible values with which the component from the request will be compared. Leave the checkbox unchecked and press the **Esc** key to specify only one value.

6. In the **THEN** section, select what should happen when the rule is triggered.

- **Accept & Inject JS Challenge** — the user will receive the content of the site only if there is a correct cookie. If the cookie is missing or incorrect, the user will receive a page with a JavaScript verification code. After successful verification, the user will receive a cookie, and the page will be automatically reloaded.
- **Accept With Cookies Only** — the user will receive the content of the site only if there is a correct cookie. If the cookie is missing or incorrect, the user will receive a response with the status 403 Forbidden.
- **Do Nothing** — the user will receive the content of the site without additional checks.
- **CDN Origin** — a special cookie for CDN will be sent to the user. Use this option for locations where static content is cached using third-party CDN services.

Please note that the action **Accept and Show Challenge** is intended only for processing GET requests and may not work correctly with other types of requests. The rest of the actions work with any kind of requests.

7. Click the button **Save**.

8. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Change the order of the rules

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Locations**.
3. Hold the left mouse button on the  icon next to the rule you want to move, and, without releasing the button, drag it to the desired position in the list.

If necessary, repeat this for other rules that need to be moved.

4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Delete a rule

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Locations**.
3. Next to the rule you want to delete, click the icon .

- Confirm the action by clicking **Delete rule** in the dialog that appears.
- To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Configuration example

The screenshot displays the configuration interface for location rules. At the top, there is a toggle for 'Enabled' which is turned on. Below this, there are sections for 'Host-groups' and 'Regional versions'. The 'Host-groups' section lists 'Web app backend' with its associated domains. The 'Regional versions' section lists regional domains. Below these, there are four location rules defined:

- 1. Cookie endpoint before backend**: This rule has an 'IF' condition where the 'HOST' is 'Web app backend', 'METHOD' is 'GET', and 'PATH' is '/get-cookie'. The 'THEN' action is 'Accept & Inject JS Challenge'.
- 2. Restrict access to backend to valid users**: This rule has an 'IF' condition where the 'HOST' is 'Web app backend', 'METHOD' is 'ANY VALUE', and 'PATH' is 'ANY VALUE'. The 'THEN' action is 'Accept With Cookies Only'.
- 3. Available for everyone**: This rule has an 'IF' condition where the 'HOST' is 'Regional versions', 'METHOD' is 'GET', and 'PATH' is '/manifest.json' or '/robots.txt'. The 'THEN' action is 'Do Nothing'.
- 4. Static content**: This rule has an 'IF' condition where the 'HOST' is 'static.example.com', 'METHOD' is 'GET', and 'PATH' is 'ANY VALUE'. The 'THEN' action is 'CDN Origin'.

In this example, the client has configured the protection of a single-page web application. The web application runs on hosts from the group “Regional versions”, and interacts with the backend that runs on hosts from the group “Web application backend”.

The web interface is available to users only if they pass the challenge (rule 2). An exception is made for several files that can be requested by external services and applications (rule 1).

The backend is available only after the browser has been verified for the web interface and has received a tracking cookie (rule 3).

Static content is served with a special cookie for external CDN providers (rule 4).

Dashboard: Bot protection → Visitors



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

The section “Bot protection → Visitors” of the dashboard Qrator Labs defines the categories of users for whom traffic processing rules are applied for bot protection purposes.

The settings in this section allow you to:

- [enable](#) or [disable](#) bot protection,
- configure forced checks [by IP address](#),
- configure exceptions [by request headers](#), [by country](#) and [by IP addresses](#),
- configure partial activation of the service with an [A/B testing setup for verifications](#).

Enable bot protection

1. Select the desired domain under **Domains** or the desired service under **Services**.
2. Select the menu item **Bot protection → Visitors**.
3. Tick off the **Enabled** option.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Disable bot protection

1. Select the desired domain under **Domains** or the desired service under **Services**.
2. Select the menu item **Bot protection → Visitors**.
3. Unset the **Enabled** option.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The bot protection will be disabled within 5–7 minutes.

Configure forced checks by IP address

This setting allows you to specify a list of IP addresses from which requests will always be matched against the rules, regardless of the setting [A/B checks distribution](#). This is useful for testing protection rules from a limited number of devices before fully enabling them for all users. For more information, see description of the recommended procedure for enabling the service in the article [Connection: Bot Protection](#).

1. Select the desired domain under **Domains** or the desired service under **Services**.
2. Select the menu item **Bot protection** → **Visitors**.
3. Under **Force checks For IPs**, click **Edit**.
4. In the dialog that appears, enter the list of IP addresses for which mandatory checks must be enabled. Separate IP addresses with line breaks.
5. Click the **Save** button.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Exclude users from checks by request headers

If this setting is configured, the system will consider all requests with certain HTTP header name-value combinations legitimate.



EXAMPLE

Let's assume that you use your own bot to make requests as part of functional or load testing. Configure the bot so that it adds an HTTP header with a specific name and value to each request. Add this name and value to the exceptions as described below. After that, the bot will be able to receive content from the upstream without passing the verification.

1. Select the desired domain under **Domains** or the desired service under **Services**.
2. Select the menu item **Bot protection** → **Visitors**.
3. In the section **Exclude by headers** add the new exception or select the exception to be changed.
 - To add a new exception, click **Add exception**.
 - To change an existing exception, click the  icon next to it.
4. In the dialog that appear, describe the condition upon which the user will be considered legitimate. In order to do that, add header name, value verification method and the string to be used for verification.

Fill out the following fields:

- **Header name** — name of the header to check.
- Verification method:
 - **Contains** — the value of the header must contain the string in the text field.
 - **Ends with** — the value of the header must end with the string in the text field.
 - **Equal** — the value of the header must be equal to the string in the text field.
 - **Starts with** — the value of the header must start with the string in the text field.

When selecting the verification method, an additional popup window appears with a checkbox labeled **One of**. Check this checkbox to specify several possible values at once with which the component from the request will be compared. Leave the checkbox unchecked and press the **Esc** key to specify only one value.

- Value used for verification.

If the checkbox **One of** was checked, you can input several values in the text field, separated by line breaks.

**WARNING**

Do not add as exceptions header values that bot creators can find in the documentation or other public sources. It is recommended to use generated random strings known only to your team.

5. Click **Save**.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Delete exception by header

1. Select the desired domain under **Domains** or the desired service under **Services**.
2. Select the menu item **Bot protection** → **Visitors**.
3. Under **Exclude by headers** click  next to the exception to be deleted.
4. Confirm the action by clicking **Delete** in the dialog that appears.
5. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Configure Exclude countries

If this setting is configured, the bot protection will consider all requests from certain countries legitimate. The source country of the request is determined by the IP address using the [MaxMind GeoIP] database (<https://www.maxmind.com/en/geoip2-services-and-databases>).

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Visitors**.
3. In the **Exclude countries** section, edit the list of countries. The countries in the list are represented by codes according to [ISO 3166] standard (https://en.wikipedia.org/wiki/ISO_3166), for example, the code "FR" is used for France.
 - To add a country to the list, click its code in the dropdown list. To search for the desired country, type one or both of the letters of its code in the text field.
 - To delete a country from the list, click the X mark next to its code.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Configure IP and subnet exceptions

If this setting is configured, the bot protection will consider all requests from certain IP addresses legitimate. For example, this way you can allow automated tools from the networks of your company or its partners.

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Visitors**.
3. In the **Exclude IPs & Networks** section, click the **Edit** link.
4. In the dialog that appears, enter the list of IP addresses and subnets in CIDR notation, separating them with line breaks.
5. Click the **Save** button.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Configure A/B checks distribution

This setting determines for which percentage of users the bot protection is activated. This is useful for step-by-step activation of the bot protection service, see [the corresponding step in the connection instructions](#).

**NOTE**

A request from a user will be analyzed by the bot protection if:

- the “Enable” option in the section **Bot protection** → **Locations** is activated,
- the “Enable” in the section **Bot protection** → **Visitors** is activated,
- any of the three conditions is met:
 - **A/B checks distribution** is set to 100%,
 - **A/B checks distribution** is set to some value, and the user's IP address is included in the sample for verification,
 - the user's IP address is in the [list for forced checks] (#edit-force-checks).

To change the A/B checks distribution:

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Visitors**.
3. In the section **A/B checks distribution**, move the slider to the desired value from 0% to 100%.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Configuration example

Enabled ? ☒

Exclude by headers ?

Force checks For IPs

203.0.113.61, 203.0.113.62, 203.0.113.63

[Edit](#)

TESTSECRET
EQUAL

g3xYPZKA6wNLhhXCgXsY6fITzN3h71gG1h9cU1mYepbrZ05j6SSIVcLYSL36xIOf
B3KX4KrvktKnKti3c15Lk.JgdEBnhoseaxJVHKLl55L0YjNRv1K36t83YA3iwTJLG
IzpsXv7Tg8wrpZnfiKROTxW4lw4G4D21O.JzKegdPyrB6npaH7z8atulsGOYsEg

[ADD EXCEPTION](#) +

Exclude Countries ?

Choose a country/countries

JP ×
PK ×

Exclude IPs & Networks ?

198.51.100.0/24, 233.252.0.0/24

[Edit](#)

A/B checks distribution ?

20%

of traffic will be affected by Anti-bot

In this example, bot protection will be applied:

- to requests from 20% of visitors,
- to requests from IP addresses 203.0.113.61, 203.0.113.62, 203.0.113.63.

The following shall be deemed legitimate requests that do not require verification:

- requests with one of the three specified cookie values `Test Secret`,
- requests from IP addresses from Japan and Pakistan,
- requests from IP addresses from subnets 198.51.100.0/24, 233.252.0.0/24.

Personal dashboard: Bot protection → Signatures



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

In the section “Bot protection → Signatures” of the personal dashboard Qrator Labs you can manage signatures designed to verify requests from mobile applications to the API. The operation scheme of such protection is described in the section [Mobile API protection using hash](#).

View the list of signatures

To view the list of signatures configured for a domain or service:

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection → Signatures**.

The signatures are displayed in a table with the following columns:

- **Status** — signature status:
 -  — the signature is enabled, requests with this signature are allowed.
 -  — the signature is disabled, requests with this signature are prohibited.
- **Comment** — short description of the signature.
- **Revoked versions** — outdated versions of the mobile application, requests from which are prohibited.

Enable signature verification

To allow requests from users providing a signature instead of a [tracking cookie](#):

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection → Signatures**.
3. At the top of the page, activate the **Enabled** option.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Disable signature verification

To prohibit requests from users providing a signature instead of a [tracking cookie](#):

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Signatures**.
3. At the top of the page, deactivate the **Enabled** option.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Add a signature

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Signatures**.
3. Click the button **Add signature**.
4. In the dialog that opens, fill in the fields:
 - **Enable** — if this option is disabled, requests with this signature will be prohibited.
 - Application version signature key settings:
 - **App version header** — the title of the header in which the mobile application must transmit data about its version.
 - **Secret key** — the key that is used to calculate the value of the salt.
 - **Salt method** — salt value calculation method:
 - **md5_lowercase** — MD5 hash represented as a lowercase string.
 - **md5_uppercase** — MD5 hash represented as an uppercase string.

In both cases, the hash is calculated from the string formed by concatenating the **Private key** and the value of the **Version header**.

- **Revoked versions** — prohibited values of the **Version header**.

You can specify up to 200 revoked versions. A request with any of these versions will be considered unsigned, and its processing will be done according to the standard [locations](#)-based algorithm.

- Signature mechanism settings:
 - **Hash Header** — the name of the header in which the mobile application must transmit the signature of the request.

- **Value method** — the method of forming the string from which the hash is calculated.
 - **salt_schema_host_uri_timestamp** — hash from a string including salt, full URI and timestamp.
 - **salt_schema_host_path_timestamp** — hash from a string including salt, URI without the GET query string, and a timestamp.

Additional values can be added to the string, see the field **Additional headers** below.

- **Hash method** — signature calculation method:
 - **md5_lowercase** — MD5 hash, represented as a lowercase string.
 - **md5_uppercase** — MD5 hash, represented as an uppercase string.
- **Additional headers** — the names of the headers whose values will be added to the string from which the hash will be calculated.

The values will be added in the order specified in this field.

- Settings of the signature validity time interval:

- **Timestamp header** — the name of the header in which the mobile application must transmit the time the request was sent.
- **Timestamp format** — the field that affects the time transmission format and the way it is checked:
 - **unix_msec** — the header must contain the request sending timestamp in milliseconds.
 - **unix_sec** — the header must contain the request sending timestamp in seconds.
 - **any** — the timestamp format is not checked.

When **unix_msec** and **unix_sec** formats are selected, the reverse HTTP proxy checks the received timestamp with the current time. If they differ by more than the specified **Interval in seconds**, the request will be considered unsigned, and its processing will be done according to the standard [locations](#)-based algorithm.

- **Timestamp drift** — the acceptable difference between the received timestamp and the current time, see the description of the field **Time format**. The field is not displayed if the time format **any** is selected.

- **Comment** (optional) — a short description of the signature.

5. Click the **Add** button.

6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

**WARNING**

Adding signatures to the list does not automatically enable signature verification for site visitors. See [Enable signature verification](#).

Edit signature

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Signatures**.
3. Find the desired signature and click  next to it.
4. In the dialog that opens, edit the required fields.

The fields of the signature editing form are identical to the fields of the [signature addition form](#).

5. Click the **Save** button.
6. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Enable specific signatures

To enable one or more signatures and allow requests that use signatures:

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Signatures**.
3. Check off the boxes next to the necessary signatures. Click the **Enable** button.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.



WARNING

Enabling signatures in the list does not automatically enable signature verification for site visitors. See [Enable signature verification](#).

Disable specific signatures

To disable one or more signatures and prohibit requests that use them:

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Signatures**.
3. Check off the boxes next to the necessary signatures. Click the button **Disable**.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Delete one signature

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Signatures**.
3. Find the desired signature and click  next to it.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Delete multiple signatures

1. Select the desired domain in the **Domains** section or the service in the **Services** section.
2. Select the menu item **Bot protection** → **Signatures**.
3. Check off the boxes next to the necessary signatures. Click the **Delete** button.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within 5–7 minutes.

Personal dashboard: CDN



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups.

Under “CDN” in the Qrator Labs dashboard, you can enable and configure [CDN integration](#) for the current domain or service.

Enable integration

If CDN integration is not enabled for a domain or service, then buttons for configuring such integration are available in this section.

The required settings depend on which CDN provider you need to configure integration with.

Enable Qrator.CDN

1. Select the domain in **Domains**.
2. Select the menu item **CDN**.
3. Click the link **Qrator.CDN**. In the dialog box that appears, click **Turn on Qrator.CDN**.
4. The dialog will list the host (domain name) that Qrator.CDN will allocate for files for your domain or service. This host is the main one and is available regardless of the settings. Additionally, you can connect any number of other hosts.

Enter the hosts in the **CDN Hostnames** field, separating them with line breaks. Click **Next**.



NOTE

You will be able to [change the list of CDN hosts](#) later.

5. Go to the control panel of your domain name registrar. For each of the hosts that CDN should respond to, create a CNAME record pointing to the primary CDN host provided by Qrator.CDN. For your convenience, an example CNAME record will be shown in the last step in Qrator.CDN setup dialog.

If necessary, wait for the DNS cache to update after the change has been made.

Enable Ngenix integration

1. Select the domain under **Domains** or service under **Services**.

2. Select **CDN**.
3. Click the **Nginx** link. In the dialog box that appears, click **Enable Nginx**.

Enable Akamai integration

1. Select the domain under **Domains** or service under **Services**.
2. Select **CDN**.
3. Click the **Akamai** link. In the dialog box that appears, click **Enable Akamai**.
4. Specify the parameters required to [interact with CDN via G2O](#). These parameters should be obtained from the Akamai website.
 - **Data header** — the name of the header containing the request data.
 - **Sign header** — the name of the header containing the request signature.
 - **Password** — the key used for signature verification.
 - **Algo** — the cryptographic algorithm used for signing the data.

Click **Save**.



NOTE

Click **Configure later** to skip this step. You will be able to [edit G2O settings](#) later.

Enable G2O integration

1. Select the domain under **Domains** or service under **Services**.
2. Select **CDN**.
3. Click the **G2O** link.
4. Specify the parameters required for [interacting with CDN via G2O](#). These parameters should be obtained from the website of the CDN you are using.
 - **Data header** — the name of the header containing the request data.
 - **Sign header** — the name of the header containing the request signature.
 - **Password** — the key used for signature verification.
 - **Algo** — the cryptographic algorithm used for signing the data.

Click **Save**.

Enable G-Core integration

1. Select the domain under **Domains** or service under **Services**.
2. Select **CDN**.
3. Click the **G-Core** link. In the dialog box that appears, click **Enable G-Core**.

Edit G2O settings

If you use integration with Akamai or with other CDNs [running on G2O](#), you can change the protocol settings in your personal dashboard.

1. Select the domain under **Domains** or service under **Services**.
2. Select **CDN**.
3. Click the link **Edit** (or **Edit G2O settings**).
4. In the dialog box that appears, specify the G2O parameters. These parameters should be obtained from the website of the CDN you are using.
 - **Data header** — the name of the header containing the request data.
 - **Sign header** — the name of the header containing the request signature.
 - **Password** — the key used for signature verification.
 - **Algo** — the cryptographic algorithm used for signing the data.

Click **Save**.



NOTE

When using Akamai CDN, you can completely disable G2O communication by clicking **Reset G2O**. However, [Akamai IP addresses will remain on the Qrator Labs whitelist](#).

Configuration example

Data header	X-G2O-Auth-Data
Sign header	X-G2O-Auth-Sign
Password 	*****
Algo	sha256

[EDIT G2O SETTINGS](#)

Disable integration

1. Select the domain under **Domains** or service under **Services**.
2. Select **CDN** → **Settings**.
3. Click **Disable**.
4. In the confirmation dialog box that appears, click **Disable**.

Dashboard: Qrator.CDN → Settings



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups. Editing the settings is available only to users in the *Technical* group.

This page describes the settings available after switching to the **CDN** mode in the switch on the top left.

For convenience, the default **Qrator** mode also provides the **Qrator.CDN** section in a domain's settings, but some settings may be unavailable there.

Change HTTP header settings

This block contains settings related to processing HTTP headers in requests and responses sent and received by the CDN server.

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **Headers**.
3. Change the settings.

- **Upstream headers, Client headers**

Sets of HTTP headers appended by CDN. **Upstream headers** defines the headers added to each user request when sending it to the upstream. **Client headers** defines the headers added to each response returned to the user.

Click the **Edit** button next to the name of the header set you want to edit. For each header enter **Name** and **Value** separately. Both name and value should be non-empty. In order to add another header click **Add row**. In order to delete a header click **Delete**. Click **Save** to apply the changes.

- **Access-Control-Allow-Origin**

A list of regular expressions defining the acceptable values of the [Origin](#) header in user requests. For each [Origin](#) value matching one of the regular expressions CDN will add the same value in the [Access-Control-Allow-Origin](#) header of the response. Thus the browser will be allowed to use the files received from CDN on websites listed in the [Origin](#) header.

Click **Edit**. Enter a valid regular expression on each line. If the expression is valid you will see a  icon next to it. To add another regular expression click **Add row**. To remove a regular expression click **Delete**. Click **Save** to apply the changes.

**TIP**

You can check the correctness of a regular expression by clicking the icon. In the appearing widget you can input an arbitrary string to check whether it matches the regular expression you put in earlier or not. If the check is successful you will see the icon near the string, otherwise you will see the icon. In order to close the widget and go back to editing the regular expression, click outside it or press **Esc** key.

- **Client IP header**

If you set this parameter, CDN will pass a user's IP address to the upstream in an HTTP header with the given name.

Click **Edit**. In the appearing text area enter a valid HTTP header name. Click **Save** to apply the changes.

Example

Upstream headers		Client headers	
Headers which will be added to every request sent by Qrator.CDN to upstream		Headers which will be added to every response sent by Qrator.CDN to client	
Name	Value	Name	Value
X-Sent-Via-CDN	1	X-Cached-By-CDN	1
Access-Control-Allow-Origin If "Origin" header from the client matches one of specified regexp, then Qrator.CDN will add Access-Control-Allow-Origin header in the response equals to received "Origin" header value		Client IP header Header name where real client IP will be passed	
/^http://example\.com(:80)?\$/		X-Client-IP	
/^https://example\.com(:443)?\$/			

View list of hosts

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **SNI/Hostnames**.

The hosts that you have set up for Qrator.CDN are displayed in a table with the following columns:

- **ID** (HTTPS only) — unique numeric identifier of the certificate in Qrator Labs. Click on the identifier to see the full certificate details in the [Certificate storage](#).
- **Hostname** — the hostname at which Qrator.CDN is available.
- **Subject** (HTTPS only) — the [Subject](#) field of the installed certificate. Click **Show details** in the table cell to see short certificate details.
- **Not valid after** (HTTPS only) — certificate expiration date. If the certificate has already expired, this field will be highlighted in color.

Next to the table column headers are filter fields you can use to quickly find domains with certain values in their **Hostname** and **Subject**.

Example

Installed certificates					ADD
ID ↑↓	Hostname ↑↓	Subject ↑↓	Not valid after ↑		
-	cdn1.example.com	Without certificate	-		
89882	cdn2.example.com	CN=cdn2.example.com Show details	Nov 17, 2033 7:31 PM		
89883	cdn3.example.com	CN=cdn3.example.com Show details	Nov 17, 2033 7:31 PM		
90121	cdn4.example.com	CN=cdn4.example.com Show details	Nov 18, 2033 9:34 PM		
90121	cdn5.example.com	CN=cdn4.example.com Show details	Nov 18, 2033 9:34 PM		
Rows per page All ▾ 1-5 of 5 < >					

Add an HTTP host

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **SNI/Hostnames**.
3. Click the **Add** button.
4. In the dialog that appears, click **Install** next to the item “Without certificate”.
5. In the text field that appears, enter one or more hosts at which Qrator.CDN should be available.
6. Click the **Install** button.

A row will be added to the table for each of the hosts entered.

7. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

8. Access your domain name registrar's control panel. For each of the hostnames CDN must respond by create a CNAME record pointing at the primary hostname that you were given when registering for Qrator.CDN. You can contact the Qrator Labs technical support to find out your primary hostname.

Wait for the DNS cache refresh after saving the changes if needed.

Add an HTTPS host

1. Make sure that a valid certificate for the host you want to add is present in [Certificate storage](#).
2. In the **CDN** mode, go to the **Domains** section and click on a domain.
3. Select the menu item **SNI/Hostnames**.
4. Click the **Add** button.
5. In the dialog that appears, find the certificate for the host that you want to add. If necessary, use the **Hostname**, **Subject** and **Type** filters to search for it.

Click **Install** next to the certificate you need.

6. In the text field that appears, enter or select from the dropdown list one or more hosts for which the certificate should be used.
7. Click the **Install** button.

A row will be added to the table for each of the hosts entered.

8. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

9. Access your domain name registrar's control panel. For each of the hostnames CDN must respond by create a CNAME record pointing at the primary hostname that you were given when registering for Qrator.CDN. You can contact the Qrator Labs technical support to find out your primary hostname.

Wait for the DNS cache refresh after saving the changes if needed.

Delete a host

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **SNI/Hostnames**.
3. Find a host and click  next to it.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Set Request URL policy

By default Qrator.CDN assumes every request it receives should be forwarded to the upstream but in some cases this may be a suboptimal behaviour.

If you are aware that on some URLs the upstream cannot provide the content and the user should receive 404 Not Found or any other error code, it is recommended to describe them using the *request URL policy*. Such configuration allows CDN to return error code responses without requesting the upstream in aforementioned cases.

**NOTE**

CDN only analyzes the path component of the requested URL. For example, if a user requests `/images/logo.png?v2`, the rules of this section will be checked for `/images/logo.png`.

In order to set the policy:

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **Request URL policy**.
3. Edit the settings.

- **White URLs**

A list of regular expressions describing all paths a user may request from CDN. If this parameter is non-empty, for any other URL the 404 Not Found error will be returned (except for **Blocked URLs**).

Click **Edit**. Enter a valid regular expression on each line. If the expression is valid you will see a  icon next to it. To add another regular expression click **Add row**. To remove a regular expression click **Delete**. Click **Save** to apply the changes.

- **Blocked URLs**

A list of reregular expressions describing specific paths for with CDN must return specific error codes.

Click **Edit**. Enter a valid regular expression on each line. If the expression is valid you will see a  icon next to it. In the **Code** field enter the error code CDN should return for the requests matching this regular expression. To add another tuple of a regular expression and an error code click **Add row**. To remove a tuple click **Delete**. Click **Save** to apply the changes.

**TIP**

You can check the correctness of a regular expression by clicking the  icon. In the appearing widget you can input an arbitrary string to check whether it matches the regular expression you put in earlier or not. If the check is successful you will see the  icon near the string, otherwise you will see the  icon. In order to close the widget and go back to editing the regular expression, click outside it or press **Esc** key.

Example

White URLs ✎

If requested URL does not match one of specified regular expressions, Qrator.CDN responses with 404 status code. Leave it empty to disable this option.

<code>/^/images/.*/\$</code>	
<code>/^/videos/.*/\$</code>	
<code>/^/files/.*/\$</code>	

Blocked URLs ✎

If requested URL matches one of specified regular expressions, Qrator.CDN responses with corresponding status code

Change cache settings

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **Cache**.
3. Edit the settings.

- **Cache control**

Defines whether CDN should respect the time to live of the cached files provided by the upstream in `Expires` and `Cache-Control` headers.

- **Client no cache**

If enabled, then no cache headers beside `Cache-control: no-cache` are sent to client.

- **Cache errors**

If an error is received from upstream with one of specified status codes, then the next request for the same resource is possible after defined timeout.

Click **Edit**. Enter the error **Code** and the corresponding **Timeout** to cache it. Timeout value may range from 60 000 to 300 000 ms. Click **Save** to apply the changes.

- **Cache errors (Client IP)**

Identical to **Cache errors** but the cache saved according to these rules will only be used for requests from the same IP address as in the original request.

- **Ignore query parameters**

Ignore all GET parameters when caching. It is convenient when you are using randomly generated GET parameters to force the browser to refresh the cached content, but do not want CDN to make unneeded requests to the upstream server.

- **Clear cache**

You can manually remove one or several files from CDN.

Click **Clear cache**. In the appearing window specify paths to all files you wish to remove from the cache. Each path should start with `/`. After that click **Clear cache** once more. The specified files will be removed from the cache in several minutes.

Example

Cache control If enabled, requests to upstream are governed by the expires, cache-control headers ☒ Enabled	Client no cache If enabled, then no cache headers beside "cache-control: no-cache" are sent to client ☐ Enabled										
Cache errors If an error is received from upstream with one of specified status code, then the next request for the same resource is possible after defined timeout <table border="0"> <thead> <tr> <th>Code</th> <th>Timeout, ms</th> </tr> </thead> <tbody> <tr> <td>500</td> <td>60000</td> </tr> <tr> <td>502</td> <td>120000</td> </tr> </tbody> </table>	Code	Timeout, ms	500	60000	502	120000	Cache errors (Client IP) When an upstream responses with status code equals to specified one, then the response will be cached for current client IP and next request from the client IP will be possible after corresponding timeout <table border="0"> <thead> <tr> <th>Code</th> <th>Timeout, ms</th> </tr> </thead> <tbody> <tr> <td>403</td> <td>60000</td> </tr> </tbody> </table>	Code	Timeout, ms	403	60000
Code	Timeout, ms										
500	60000										
502	120000										
Code	Timeout, ms										
403	60000										
Ignore query parameters Ignore query parameters in cache key calculation. ☒ Ignore	Clear cache Remove files from the CDN cache manually CLEAR CACHE										

Change transport settings

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **Transport**.
3. Edit the settings.

- **HTTP/2 Support**

Enable the HTTP/2 protocol support. If disabled, CDN will use only the HTTP/1.1 protocol.

- **Compression algorithms**

You can enable or disable the support of certain compression algorithms here. By default Qrator.CDN supports `gzip`, `deflate` and `br` formats but you can disable any of them by unticking the checkboxes.

Example

HTTP2 support Set this option to enable Qrator.CDN support both protocols (HTTP/2 and HTTP/1.1) ☒ HTTP2	Compression algorithms Uncheck compression algorithm if you don't want Qrator.CDN to use it ☒ gzip ☒ deflate ☒ br
--	--

Configure conversion of the images to WebP

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **WebP**.

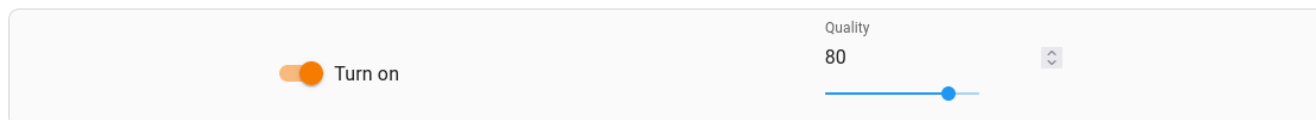
3. Click **Turn on** to enable or disable the autoconversion to the WebP format.

If necessary, change the value for the **Quality** parameter. Lower values of the parameter mean higher effectiveness of the compression but bigger losses in the image quality.

4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Example



Configure HTTP redirects

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **HTTP redirects**.
3. Click **Turn on** to enable or disable the redirects.

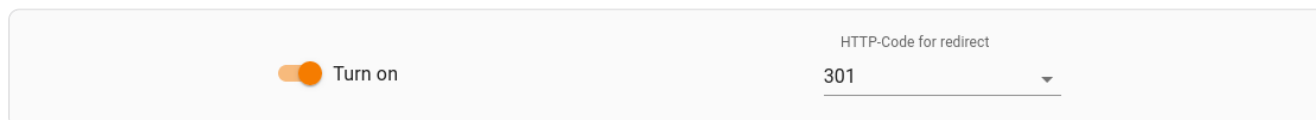
If necessary, change the **HTTP-Code for redirects**. The following codes are available:

- 301 Moved Permanently
- 302 Found
- 307 Temporary Redirect
- 308 Permanent Redirect

4. To save the changes, click **Save** in the notification at the bottom of the screen.

The changes will be applied within a few minutes.

Example



Turn off Qrator.CDN

1. In the **CDN** mode, go to the **Domains** section and click on a domain.
2. Select the menu item **Turn Off**.
3. Click the **Turn off** button.
4. In the appearing window click **Turn off**.

Personal dashboard: Utilities



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups.

The “[Check IP](#)”, “[Trace](#)” and “[Vulnerabilities](#)” utilities in the Qrator Labs dashboard can be useful for diagnosing user problems and checking if the firewall configuration is correct.

Check IP address detail

Using the “IP Check” utility, you can find out how Qrator Labs reverse proxies treat inbound traffic from a given IP address. The utility is available for [domains](#) and [services](#). If you have multiple domains or services configured in your account, the status of the same IP address may be different for them.

To use the utility:

1. Select the domain under **Domains** or service under **Services**.
2. Select **Check IP**. (When working with a domain, this menu item is found in submenu **Utilities**)
3. In the text box, enter the IP address.
4. Click **Search**.

One of the following statuses will be shown next to the form.

- Statuses in which traffic from the IP address is allowed without restriction:
 - *“IP address not blocked”*
 - *“IP address is on the “Whitelist””*
 - *“IP address is upstream”*
- Statuses in which traffic from an IP address is restricted:
 - *“IP address blocked via Qrator API”*
 - *“IP address blocked by WAF”*
 - *“IP address blocked by Qrator filtering system”*
 - *“Blocked based on HTTP error code (456) server response”*

View traceroute to IP address

With the “Traceroute” utility, you can see the route via which packets are sent from each of the [Qrator Labs traffic scrubbing centers] to a given IP. This utility is only available for [domains](#).

To use the utility:

1. Select your domain under **Domains**.
2. Select **Utilities** → **Traceroute**.
3. Select the **IP address** of your upstream from the drop-down list.
4. Specify the **Port**.
5. Click **Run**.

Check for vulnerable ports

The “Vulnerabilities” section displays the results of regular checks of the [upstreams](#) for open ports 80 and 443.

The checks are performed each 4 hours from an IP address outside the Qrator Labs subnet. With a correct configuration, firewall should drop the connection from such an IP address. If any kind of server response was received during the check, this indicates a vulnerability of your network to attacks that are directed to the upstream directly, i.e., not via the Qrator Labs network.

To see the results of a check:

1. Select your domain under **Domains**.
2. Select **Utilities** → **Vulnerabilities**.
3. In the **Date** field, select the date and time you want. By default, the date and time of the latest check are selected.

The results are displayed in a table with the following columns:

- the IP address of the upstream,
- the port checked,
- the results of the check: “*Protected*” or “*Vulnerable*”.

If at least one of your ports is vulnerable, change the configuration of your firewall to eliminate the vulnerability. To do so, follow the [Firewall configuration](#) guide and, if necessary, contact the Qrator Labs technical support.

Personal dashboard: Certificate Storage



NOTE

This dashboard section can be accessed by users with *Admin* or *Technical* role only. The settings can be edited by users with *Technical* role only.

The Certificate Storage contains TLS certificates available for your account, and it allows you to install these certificates for specific domains and services.

Before you start working with certificates

To start using TLS encryption on [domain](#) or [services](#), you need to take two steps:

1. Adding a certificate to the certificate storage

In the “Certificate storage” section of your dashboard, you can *add* a certificate to the system by [generating it with Let's Encrypt](#) or [downloading ready-made files from another source](#). You can do the same with [Qrator API methods](#). The added certificate does not become activated automatically; it becomes available for installation for any of the account's domains and services; see the next step.

2. Installing a certificate for a domain or service

Installing a certificate means setting up a match between the certificate and the specific domain name for which the certificate will be used. A certificate can be installed [in the “Certificate Storage” section, under “TLS” for the domain or service](#). You can also use [Qrator API methods](#).

When installing a certificate, you specify which domain names it will be used for. You can also select the “Default Host” option. A certificate installed with this option will be used when a user makes a request to a domain name for which no certificate is assigned. Each domain or service for which certificates are installed must have one certificate with the “Default Host” option.

View list of certificates

Certificates added to the certificate storage are displayed in the **Active certificates** table. The table has the following columns:

- **ID** — Unique numeric identifier of the certificate in Qrator Labs.
- **Expires** — Certificate expiration date. If the certificate has already expired, this field will be highlighted in color.
- **Type** — Certificate origin: *Loaded* or *Let's Encrypt*.

- **Subject** — Domain names for which the certificate is valid, as well as a link to [view certificate details](#).

If a certificate is installed on one or more resources (domains or services), the following additional information will be shown for it:

- **Resources** - Links to “TLS” for managing installed certificates. Links in different lines lead to the settings of different domains or services.
- **Hostname** - Information about which domain names this certificate is used for. Each line can be a specific domain name or a “*default hostname*” line; see [Before you start working with certificates](#).

Above the table there are filter fields that you can use to find certificates that contain the required values in the **Subject**, **Hostname** and **Resources** fields.

View certificate details

Brief details

To see brief details about a certificate:

1. Go to **Certificate storage**.
2. Find the line with the certificate whose brief details you want to view and click **Show certificate details**.

The appearing dialogue will contain the following fields:

- **ID** — the unique numerical identifier for the certificate in the Certificate storage.
- **Type** — the origin of the certificate: *Upload* or *Let's Encrypt*.
- **Subject** — the domain name(s) the certificate is valid for.
- **Not valid before, Not valid after** — the start and the end date of the certificate validity period.

The certificate may contain one or several chains. If there is more than one chain, you can switch between the chains using radio buttons with the corresponding encryption algorithm names on them. For each chain the following data is displayed::

- **Private key encryption** — the cryptographic algorithm used for the private key.
- **Private key size** — the private key size in bytes.
- **SHA-256 Fingerprint, SHA-1 Fingerprint** — the public key fingerprints.
- **Certificate chain** — the whole chain in PEM format. You can copy the full text with the  button.
- **Private key** — the private key in PEM format. You can copy the full text with the  button. This field is not displayed if [the private key is hidden](#).

Full details

To see complete details about a certificate:

1. Go to **Certificate storage**.
2. Find the line with the certificate whose complete details you want to view and click it.

The appearing dialogue will contain the following fields:

- **ID** — the unique numerical identifier for the certificate in the Certificate storage.
- **Type** — the origin of the certificate: *Upload* or *Let's Encrypt*.
- **Subject** — the domain name(s) the certificate is valid for.
- **Resources** — the ID and name of a domain or a service for which the certificate is installed. Displayed only for [installed certificates](#).
- **Certificate auto-renewal** — shows whether the certificate is renewed automatically or not. Displayed only for Let's Encrypt certificates [issued via Personal dashboard](#).
- **Not valid before, Not valid after** — the start and the end date of the certificate validity period.

The certificate may contain one or several chains. If there is more than one chain, you can switch between the chains using radio buttons with the corresponding encryption algorithm names on them. For each chain the following data is displayed::

- **Private key encryption** — the cryptographic algorithm used for the private key.
- **Private key size** — the private key size in bytes.
- **SHA-256 Fingerprint, SHA-1 Fingerprint** — the public key fingerprints.
- **Certificate chain** — the whole chain in PEM format. You can copy the full text with the  button.
- **Private key** — the private key in PEM format. You can copy the full text with the  button. This field is not displayed if [the private key is hidden](#).

If a certificate is [installed](#) for one or more domains or services, they'll be shown on this page as a list. Next to each domain or service is information showing the domain names the certificate is used for. Each line can be a specific domain name or a *"Default Host"* line. Click on the name of the domain or service to manage all certificates installed for it at *"TLS"*.

Issue Let's Encrypt certificate

Qrator Labs allows you to issue a new certificate for your domain name directly from your dashboard. This is made possible thanks to [Let's Encrypt](#) integration.

1. Make sure that the domain name is already added in the settings of the service you are going to use it for. Otherwise, Qrator Labs will not be able to prove the control over the domain during the challenge for Let's Encrypt.
 - For the [reverse HTTP proxy](#): in the A record in the DNS zone for your domain, specify the Qrator IP, and [add the domain name to the upstreams list](#).

- For [Qrator.CDN](#): create a CNAME record in the DNS zone according to the instructions given during setup, and [add the domain name to the hostnames list](#) as a hostname without certificate.
- For [connecting via BGP](#): configure BGP peering with the Qrator Labs network and [enable the service in the dashboard](#).

2. Go to **Certificate storage** and click **Add certificate**.

On the page that opens, select **Use Let's Encrypt**. Click **Next**.

3. Specify information to complete the [HTTP-01 challenge](#) validation for Let's Encrypt:

- Select a domain or service on your account that already has Qrator Labs reverse HTTP proxy or BGP configured,
- Enter one or more domain names for which a certificate should be issued.

The reverse proxy on the Qrator Labs side will automatically fulfill the necessary requirements to pass the validation. No additional configuration on your part is required for this.

Click the **Generate Certificate** button.

4. Wait for the Let's Encrypt server to complete checking the request. If the process takes a long time, you can close the tab and return to the process later; see [View or delete incomplete actions](#).

If the validation fails, you will see a message describing the error. If necessary, contact Qrator Labs technical support to find out the cause of the problem.

5. Before adding a certificate to the certificate storage, you'll see a confirmation screen where you can view the details about the certificate you're adding.

Information about chain:

- **Not valid before, Not valid after** — Beginning and end of certificate validity period.
- **Chain length** — Number of certificates in the chain, not counting the last one.

Information about each public key:

- **Subject** — Information about the certificate scope.
- **Not valid before, Not valid after** — Beginning and end of key validity period.
- **SHA-1, SHA-256** — Public key fingerprints.

To complete the action, click **Add to certificate storage**.

Note that adding a certificate does not make it automatically installed and active. See details in [Before you start working with certificates](#).

Upload certificate

Use this method if you've already generated a TLS certificate through some service or by yourself.

1. Go to **Certificate storage** and click **Add certificate**.

On the page that opens, the **Upload certificate** tab will be automatically selected.

- Specify the files you want to upload. To do this, click the  icon and select files from the system dialog box or just drag them to the page.

The list of files selected for upload will be shown in a special confirmation dialog box. In this box, you can:

- click **Delete** to cancel uploading a specific file selected in error,
- click **Delete All** to clear the list of files,
- click **+** to select additional files.

Make sure that the file set you provide contains both the public keys or their public key chains and their corresponding private keys.

X.509 certificates in PEM format are supported. Supported container formats are `pem`, `cer`, `crt`, `der`, `p7b`, `p7c`, `p12`. All or part of the file set can be uploaded as an archive in one of the following formats: `tar`, `tar.gz`, `tar.bz2`, `zip`, `7z`.

- Click **Upload** and wait for the files to be processed. If the process takes a long time, you can close the tab and return to the process at a later time; see [View or delete incomplete actions](#).
- If the file set provided contains multiple certificates or multiple possible chains for a single certificate, you will see all available options. (This step is automatically skipped if the file set contains only one certificate and one chain.)

Certificates / Certificate (ID: 1097076)

DELETE

Select certificate to install

PREVIOUS

● ● ● ● ●

NEXT

Subject

CN=example.com

Not valid before

Jul 30, 2022 8:52 AM

Not valid after

Jul 30, 2023 8:52 AM

SHA-1

35:6F:70:BD:D2:0C:36:08:F2:96:89:F3:26:3A:43:2A:A2:78:8A:4C

SHA-256

F0:06:70:CC:BB:43:87:DE:FD:45:5D:6B:3D:51:26:40:F8:97:DF:56:08:3A:BE:A7:27:1C:DA:93:86:8C:53:B8

INSTALL

Select the certificate and chain you want using the **Previous** and **Next** links.

Click **Install** to go to the next step.

5. Before adding a certificate to the certificate storage, you'll see a confirmation screen where you can view the details of the certificate you're adding.

Information about chain:

- **Not valid before, Not valid after** — Beginning and end of certificate validity period.
- **Chain length** - Number of certificates in the chain, not counting the last one.
- **Self-signed** - Shows whether the certificate is [self-signed](#).
- **Trusted CA** - Shows whether the chain contains a certificate signed by a trusted CA.

Information about each public key:

- **Subject** — Information about the certificate scope.
- **Not valid before, Not valid after** — Beginning and end of key validity period.
- **SHA-1, SHA-256** — Public key fingerprints.

To complete the action, click **Add to certificate storage**.

Note that adding a certificate does not make it automatically installed and active. See details in [Before you start working with certificates](#).



NOTE

If the downloaded file set contains multiple certificates, then after adding you can close the dialog box and go back to the previous step to add another certificate from the set to the certificate storage.

View or delete incomplete actions

There are two kinds of incomplete actions that could occur when adding a certificate:

- Pending requests.

When a request to add a certificate has been initiated but not yet completed [Let's Encrypt server check](#) or [Qrator Labs server processing](#), that request is displayed under the **Pending** tab. If there are no requests in your account marked as pending, the tab will not be displayed.

- Processed requests

When the automatic check is already completed, this request is displayed under the **Processed** tab. If there are no requests in your account marked as processed, the tab will not be displayed.

To go to an incomplete action:

1. Go to **Certificate storage**. Open the **Pending** or **Processed** tab.
2. Click the action you're interested in.

Then you can continue adding the certificate from the same place where [Let's Encrypt server check](#) or [Qrator Labs server processing](#) was initiated.

To cancel an action:

1. Go to **Certificate storage**. Open the **Pending** or **Processed** tab.
2. Find the action you want to undo and click  next to it.
3. In the confirmation dialog box, click **Delete**.

Install certificate for a domain or service

As described in [Before you start working with certificates](#), installation requires a certificate already added to the Certificate storage.

To install a certificate:

1. Go to **Certificate storage**.
2. Find the line with the certificate you want to install and click it.
3. Click **Install certificate**.
4. In the dialog box that appears, select **Resource** - the domain or service on your account for which you want to install the certificate.

Click **Next**.

5. Specify which domain names the certificate should be used for.
 - If the certificate should be used for one or several known domain names, list them in the text area. Add a comma, a whitespace or use a new line after adding each domain name.
 - If you want to add the certificate before specifying any domain name, or add a domain name which doesn't have any other certificates installed for it, enable **Use as default for any hostnames** option.

Press **Install**.



WARNING

If a domain or a service has at least one certificate installed for any specific hostname it should also have a certificate for a default hostname installed.

Enable or disable auto-renewal

Qrator Labs can perform auto-renewal for Let's Encrypt certificates that were [issued through the dashboard](#); see [Automatic certificate issuance](#). This feature is enabled by default for each such certificate.

To disable or re-enable this feature:

1. Go to **Certificate storage**.
2. Find the line with the required **Let's Encrypt** certificate and click it.
3. Next to the **Certificate auto-renewal** field, click **Disable auto-renewal** or **Enable auto-renewal**. In the dialog box that appears, confirm your action.

Hide certificate private key

By default, any user included into a group with the "Certificate Section" policy can [view full details of any certificate](#), including its private key. Depending on who is in your team, this may be undesirable from a security point of view. Therefore, Qrator Labs allows you to permanently hide the contents of a particular certificate's private key from your personal dashboard, without affecting the certificate's functionality.

To hide the contents of a private key from your dashboard:

1. Go to **Certificate storage**.
2. Find the line with the certificate whose private key you want to hide and click it.
3. Next to the **Private key** field, click **Protect private key**. In the dialog box that appears, confirm your action.

Delete certificate

1. Go to **Certificate storage**.
2. Find the line with the certificate you want to delete and click it.
3. At the top of the page that opens, click **Delete**.

If the certificate is installed on domains or services, a message will appear telling you that you cannot delete the certificate. In this case, go to the settings of the respective domains or services and [disable the certificate](#), then start deleting it again.

4. In the dialog box that appears, confirm the action by clicking **Delete**.

Dashboard: API tokens



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Administrative* and *Technical* groups.

In the “API tokens” section of the dashboard, you can add or delete tokens for [authorization in the Qrator API](#). Each token can be individually assigned a set of API methods that you will be able to call using it.

View list of tokens

The **API tokens** section displays a table with all the active tokens for the account.

The following columns are available:

- **Token value** — the string that should be passed in a HTTP header for the [authorization in Qrator API](#).

The string is displayed in abbreviated form. The full value of the token can only be retrieved through the interface after its creation.

- **Comment** — the comment, as entered when adding or editing the token. May be empty.
- **Policies** — the number of policies added for the token.

Sometimes, this value corresponds to the number of API methods that can be called using the token. However, in some cases, the actual number of methods can be greater (when using the “*Any client's method*” or “*Any resource's method*” policies).

Add a token

1. In the **API tokens** section, click the **Add API-token** button. A dialog for adding a token will appear.
2. Enter the **Comment** for the token. You can leave the field empty.

It is recommended to use the comment to describe who and for what will use the token. This will allow to easier find the token to [edit it](#) later.

3. Add one or more policies. The set of policies defines which API methods you will be able to use with this token.

To add a policy related to a Qrator API [client method](#):

1. Click **Add policy**. A dialog for adding a policy will appear.

2. In the dropdown list, select a client method name or the *"Any client's method"* policy.
3. Click **Save**.

To add a policy related to a Qrator API [domain method](#) or [service methods](#):

1. Click **Add policy**. A dialog for adding a policy will appear.
2. In the dropdown list, select a resource method name or the *"Any resource's method"* policy.
3. Select one or more resources (domains or services) to which the policy should apply, or check the **Any resource** checkbox.
4. Click **Save**.

To cancel adding a policy, click  next to its name.

5. Click **Add** to finish adding the token.

An added token can be copied by clicking the copy icon. You can copy a token only once during the creation process; after that, it will not be displayed in the interface.

Edit a token

1. In the **API tokens** section, click  next to the token that you want to edit.
2. If necessary, edit the **Comment** for the token.
3. If necessary, edit the list of policies for the token.

To add a policy related to a Qrator API [client method](#):

1. Click **Add policy**. A dialog for adding a policy will appear.
2. In the dropdown list, select a client method name or the *"Any client's method"* policy.
3. Click **Save**.

To add a policy related to a Qrator API [domain method](#) or [service methods](#):

1. Click **Add policy**. A dialog for adding a policy will appear.
2. In the dropdown list, select a resource method name or the *"Any resource's method"* policy.
3. Select one or more resources (domains or services) to which the policy should apply, or check the **Any resource** checkbox.
4. Click **Save**.

To delete a policy, click  next to its name.

5. Click **Edit** to save the changes and close the dialog.

Delete a token

1. In the **API tokens** section, click  next to the token that you want to delete.

2. Confirm the action by clicking **Delete** in the dialog that appears.

Personal dashboard: Users

If you need to provide access to your Qrator Labs account for multiple employees at once, you can create several user accounts. All employees' accounts are linked to the same *account* that was set up for the company. You can manage accounts in the **Users** section of your account.

Different user accounts can be granted different access rights in the dashboard, see the [Groups](#) section description.



NOTE

In the [default configuration](#), only users in the *Access Manager* group can edit other users' accounts, as well as add and remove user accounts.

Changing the password and enabling two-factor authentication for a user account is only available to the said user account itself, see [My profile](#).

View list of users

The **Users** section of the dashboard displays a table with all the account users.

The following columns are available:

- **Name** - Last name and first name of the user.
- **Groups** - List of [groups](#) the user account is included into.
- **Contact details** — User's email address.

View user details

In the **Users** section of the personal dashboard, click on the user's name. The page that opens will display information about that user.

The information fields are divided into three groups.

- **Basic information**
 - **First name** and **Last name** of the user.
 - **Groups** - List of [groups](#) the user account is included into.
- **Contact details**
 - **Email address** specified when [creating the user account](#).

- **Skype** — User's Skype ID for instant communication (if specified).
- **Phone number, Extension number, Mobile phone** - Details for instant communication by phone (if specified).
- **Language settings**
 - **Language** - Dashboard interface language for the user.

Edit user information

1. In the **Users** section of the personal dashboard, click on the user.
2. In the **User information** block, click .
3. Edit user information.
 - **First name** and **Last name** of the user.
 - **Skype** - User's Skype ID for instant communication (optional).
 - **Phone number, Extension number, Mobile phone** - Details for instant communication by phone (optional).
4. Click **Save**.

Edit user groups

1. In the **Users** section of the personal dashboard, click on the particular user.
2. In the **User information** block, click .
3. In the **Groups** section, edit the list.

To add a group to the list, click its name in the dropdown list. To search for the desired group, type its name in the text field.

To delete a group from the list, click the X mark next to its name.
4. Click **Save**.



TIP

If you've changed the groups for the account you are logged in with, refresh the page to fully apply the changes.

Edit user language preferences

1. In the **Users** section of the personal dashboard, click on the particular user.

2. In the **User information** block, click .
3. Set the required settings.
 - **Language** - Dashboard interface language for the user.
 - **Report language** - Language for generating [reports](#) that the user will receive by email. If you don't want the user to be receiving reports, select **Don't send**.
4. Click **Save**.

Create a user account

1. In the **Users** section of the personal dashboard, click **Create user**.
2. In the dialog box that appears, enter the user's details.

The information fields are divided into four groups.

- **Basic information.**
 - **First name** and **Last name** of the user.
- **Contact details**
 - **Email address** - Will be used for logging in, as well as for sending [notifications](#) and [reports](#).
 - **Skype** - User's Skype ID for instant communication (optional).
 - **Phone number, Extension number, Mobile phone** - Details for instant communication by phone (optional).
- **Groups**
 - **Selected groups** — List of [groups](#) the user account should be included into.

To add a group to the list, click its name in the dropdown list. To search for the desired group, type its name in the text field.

To delete a group from the list, click the X mark next to its name.
- **Language settings.**
 - **Language** - Dashboard interface language for the user.

3. Click **Save**.

A message containing a link to set a password will be sent to the user's email address.

Disable two-factor authentication

If one of the users [enabled two-factor authentication](#) in the account, but subsequently loses access to the application generating the temporary codes, then a user with an edit access to "Users/Groups Sections" can

disable two-factor authentication for that user.

**WARNING**

Before taking this action, make sure, using trusted means of communication, that the request to disable two-factor authentication actually comes from the user himself.

1. In the **Users** section of the personal dashboard, click the name of the user you want to disable two-factor authentication for.
2. In the **Account is protected** block, click **Disable**.
3. Enter the code and click **Confirm**.

After that, the user will be able to log in to the account with just username and password. If necessary they will be able to [enable two-factor authentication](#) again by themselves.

Reset user password

1. Under **Users** in your account, click the name of the user you want to reset password for.
2. In the **Password reset** block, click **Reset password**.
3. In the dialog box that appears, confirm the action by clicking **Reset Password**.

A password reset link will be sent to the user by email.

Delete user

1. In the **Users** section of the personal dashboard, click the name of the user you want to delete.
2. In the **Delete user** block, click **Delete account**.
3. In the dialog box that appears, confirm the action by clicking **Delete account**.

Dashboard: Groups

Groups are an essential component of the access rights systems that is used in the Qrator Labs dashboard. To permit a user access to a certain dashboard section, it is necessary to add them to a group, for which the required [policies](#) are set. A user can be a part of multiple groups. Groups can also be related to one another in such a way that one group's policies are automatically added to another one.

Groups are managed in the “Groups” section of the dashboard.



NOTE

In the [default configuration](#), this section of the personal dashboard is available only to users in the *Access Manager* group.



WARNING

When configuring access right, make sure that at least one user is always able to edit settings in “Users” and “Groups”. If necessary, contact the Qrator Labs tech support team to restore the access.

Introduction

Terms

The system of access rights for the dashboard is based on the three terms.

- *User* – a person (usually, an employee of of company that is a Qrator Labs client) that uses their own email address and password for logging into the dashboard. A company account may have multiple users configured. You can view and manage user accounts in the [Users](#) section. Any user can edit their own account in the [My profile](#) section.
- *Policy* – a permission to access a certain dashboard section or [resource](#). Some types of policies allow for more detailed configuration: for example, a right to add new resources does not automatically mean a right to delete them.
- *Group* – a set of users, grouped in order to grant them common access rights to the dashboard. A group implements a “many-to-many” relationship between users and configured policies. It is also possible to [include groups into one another](#).

**EXAMPLE**

One possible approach to managing access rights is to create groups that correspond to different departments of the company: “Tech support”, “Sales”, etc. You can also create a separate group “Senior tech support specialists” and include it into “Tech support” to reuse the configured policies.

Policies

Access rights to the dashboard are granted by adding *policies*, whose names approximately match the names of the dashboard sections. For example, policies exist with names “*Prefixes Section*”, “*Finance Section*”, etc.

For some kinds of policies, an identifier is required of a specific resource to which the access will be granted. For example, when adding a “*Specific resource: Tunnels settings*” policy, a domain identifier is specified. Such a policy does not affect the access to other domains. If necessary, you can add multiple similar policies with different resource identifiers.

When adding any kind of policy, it is also specified which operations the group's users will be allowed to perform in the section. There are four types of operations: *Create*, *Read*, *Update*, *Delete* (some operations may be inapplicable to a policy). For example, if you add a “*Certificates Section*” policy and specify only the *Read* and *Update* operations, then the users of this group will be able to see and update existing TLS certificates but not add or delete them. The exact meaning of each operation may differ slightly from one section to another.

Default configuration

For all Qrator Labs clients, a default configuration is created with four groups, as shown in the table below. The user that is created by default belongs to all four groups, including the “Access Manager” group that allows them to create new users and add them to the existing or new groups.

Group	Access rights
Technical	View and edit settings in “Domains” (all subsections), “Services” (all subsections), “Prefixes”, “Ingress”, “Certificate storage”, “WAF”, “Analytics”, “Tickets”, “API tokens”. View settings in “Reports”, “Users”, “Groups”.
Finance	View and edit settings in “Finance”, “Company”, “Tickets”. View settings in “Users”, “Groups”.
Administrative	View and edit settings in “Finance”, “Company”, “Tickets”. View settings in “Domains” (all subsections), “Services” (all subsections), “Prefixes”, “Ingress”, “Certificate storage”, “WAF”, “Analytics”, “Reports”, “API tokens”, “Users”, “Groups”.
Access Manager	View and edit settings in “Users”, “Groups”.

View list of groups

The **Groups** section of the personal dashboard shows a table of all groups for the current account. For each group, its **Name**, **Number of policies**, **Number of users**, and **Number of groups** are shown.

Next to the **Name** column header is a filter field you can use to quickly find groups with a certain substring in their names.

Add group

1. In the **Groups** section of the dashboard, click the **Add group** button.
2. In the dialog that appears, enter a name for the new group.

Click **Next**.

3. Select policies that will be set for the new group. This step can be skipped by clicking **Next**.

To add each new policy:

1. Click **Add policy**. A policy adding dialog should appear.
2. Select a policy that you want to add to the group.
3. Specify a resource identifier, if the policy requires it (see [Policies](#)).
4. Check the checkboxes next to the names of operations that the users in the group should have.
5. Click **Save** to finish configuring the new policy and close the dialog.

After adding a policy, you can click  , to go back to the dialog and edit the policy. To cancel adding the policy, click  next to its name.

To go to the next step, click **Next**.

6. Select parent groups, from which the new group should inherit policies. This step can be skipped by clicking **Next**.

To add a parent group, click **Add group**. In the dialog that appears, select one or more parent groups in the dropdown and click **Save**. To cancel adding a parent group, click  next to its name.

To go to the next step, click **Next**.

7. Select users that should be included into the new group. This step can be skipped by clicking **Create**.

To add a user, click **Add users**. In the dialog that appears, select one or more parent users in the dropdown and click **Save**. To cancel adding a user, click  next to their name.

To go to the next step, click **Create**.

Rename group

1. In the **Groups** section of the dashboard, click on a group's name that you want to change.
2. Next to the page title, click .

3. In the dialog that appears, enter the new name and click **Change**.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

Add policy

1. In the **Groups** section of the dashboard, click on a group's name.
2. In the **Policies** block, click **Add**. A policy adding dialog should appear.
3. Select a policy that you want to add to the group.
4. Specify a resource identifier, if the policy requires it (see [Policies](#)).
5. Check the checkboxes next to the names of operations that the users in the group should have.
6. Click **Save** to finish configuring the new policy and close the dialog.
7. To save the changes, click **Save** in the notification at the bottom of the screen.

Edit policy

1. In the **Groups** section of the dashboard, click on a group's name.
2. in the **Policies** block, click  next to the name of the group you want to edit.
3. In the dialog that appears, check or uncheck the checkboxes next to the **Create, Read, Delete, Update** operations.
4. Click **Save** to finish configuring the new policy and close the dialog.
5. To save the changes, click **Save** in the notification at the bottom of the screen.

Delete policy

1. In the **Groups** section of the dashboard, click on a group's name.
2. in the **Policies** block, click  next to the name of the group you want to delete.
3. Confirm the action by clicking **Delete** in the dialog that appears.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

Delete multiple policies

1. In the **Groups** section of the dashboard, click on a group's name.
2. Check off the boxes next to the necessary policies. Click the **Delete** button.

3. Confirm the action by clicking **Delete** in the dialog that appears.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

Include a group into other groups

A group can be included into other groups. This means that all the policies configured for the group will also be configured for the other groups it is included into.

To include a group into other groups:

1. In the **Groups** section of the dashboard, click on a child group's name.
2. In the **Included in groups** block, click **Add**.
3. In the dialog that appears, select one or more parent groups in the dropdown and click **Save**.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

Remove a group from other groups

1. In the **Groups** section of the dashboard, click on a child group's name.
2. In the **Included in groups** block, click  next to a parent group.
3. Confirm the action by clicking **Delete** in the dialog that appears.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

Add user to a group

1. In the **Groups** section of the dashboard, click on a child group's name.
2. In the **Users** block, click **Add**.
3. In the dialog that appears, select one or more users in the dropdown and click **Save**.
4. To save the changes, click **Save** in the notification at the bottom of the screen.

Delete a user from a group

1. In the **Groups** section of the dashboard, click on a child group's name.
2. In the **Users** block, click  next to a user's name.
3. Confirm the action by clicking **Delete** in the dialog that appears.
4. To save the changes, click **Save** in the notification at the bottom of the screen.



Delete group

1. In the **Groups** section of the dashboard, click on a child group's name.
2. In the top right corner, click **Delete group**.
3. Confirm the action by clicking **Delete** in the dialog that appears.

Dashboard: Notifications (profile configuration)



NOTE

This page describes notification settings for a specific user account. To learn about configuring notification thresholds for a resource, see [Notifications \(resource configuration\)](#).

In the “Notifications” section of the personal dashboard, you can enable or disable specific types of notifications from Qrator Labs for your user account.

Check or uncheck the appropriate checkboxes to modify the list of notifications you will receive. To save the changes, click **Save** in the notification at the bottom of the screen.

The enabled notifications will be sent to the email address specified in [your profile](#).

Note that the following types of notifications are sent when a problem occurs that can lead to an incorrect operation of your resources:

- “Let’s Encrypt certificate auto-update for one of the resources was disabled”,
- “Traffic thresholds notifications”,
- “Bansize threshold notifications”,
- “Certificate expiration”.

Before disabling these types of notifications, we recommend making sure that they are enabled for at least one other user account.

Personal dashboard: My Profile

In the “My Profile” section of the dashboard, you can manage the settings of the account that you are currently logged in to. Every account has the My Profile section.

View user information

Go to **My Profile**. The page that opens will display information about the user.

The information fields are divided into three groups.

- **Basic information**
 - **First name** and **Last name** of the user.
 - **Account type** - “Regular user” or “Full access”.
 - **Groups** - List of [groups](#) the user account is included into.
- **Contact details**
 - **E-mail address** specified when [creating the user account](#).
 - **Skype** — User's Skype ID for instant communication (if specified).
 - **Phone number, Extension number, Mobile phone** - Details for instant communication by phone (if specified).
- **Language settings**
 - **Language** - Dashboard interface language for the user.
 - **Report language** - Language for generating [reports](#) for the user (if reporting is enabled).

Edit user information

1. In the **My Profile** section, click .
2. Edit the user information.
 - **First name** and **Last name** of the user.
 - **Skype** - User's Skype ID for instant communication (optional).
 - **Phone number, Extension number, Mobile phone** - Details for instant communication by phone (optional).
3. Click **Save**.

Edit user language preferences

1. In the **My Profile** section, click .
2. Set the required settings.
 - **Language** - Dashboard interface language.
 - **Report language** - Language for generating [reports](#) that you will receive by email. If you don't want to be receiving reports, select **Don't send**.
3. Click **Save**.

Enable two-factor authentication

Your Qrator Labs account is password-protected by default. To enhance account security, you can enable *two-factor authentication*. In this case, in order to access your dashboard, you will have to enter not only your email address and password, but also a one-time six-digit code generated on your mobile device. You can use Google Authenticator on [iOS](#) or [Android](#) or another TOTP application to generate the codes.

To enable two-factor authentication:

1. Go to **My Profile**.
2. In the **Protect your account** block, click **Start**.

This will generate and display your private key and QR code.
3. Launch the app on your mobile device. Add an account to it by entering your private key or scanning the QR code from your dashboard.
4. In the dashboard, click **Next**.
5. In the **Pin code** field, enter the code generated by the mobile app and click **Confirm**.

Disable two-factor authentication



NOTE

To disable two-factor authentication, you must have access to a mobile device with an app that generates temporary codes. If this is not possible, [a user with full access can disable two-factor authentication for you](#).

1. Go to **My Profile**.
2. In the **Account is protected** block, click **Disable**.

3. In the **Pin code** field, enter the code generated by the mobile app and click **Confirm**.

Change password



NOTE

To set a new password, you must enter your current password. If you have forgotten your current password, a user with full access can reset your password for you]([users.md#reset-password](#)).

1. Go to **My Profile**.
2. In the **Change password** block, click .
3. In the **Password** field, enter your current password.
4. In the **New password** field, enter your new password. In the next field, enter the new password again.
5. Click **Save**.

Configuring SSO via OAuth 2.0 / OpenID Connect

Introduction

Qrator Labs supports user authentication through identity federation using the OAuth 2.0 and OpenID Connect (OIDC) protocols.

This is a standard mechanism for implementing single sign-on (Single Sign-On, SSO), a technology that allows users to access multiple systems without using separate usernames and passwords for each one.

This approach is called identity federation: user credential information is stored by a trusted identity provider - Identity Provider (IdP), while Qrator Labs acts as the Service Provider (SP) and delegates authentication to the external system.

If your organization already uses a centralized user and access management system, such as Microsoft Active Directory, Keycloak, or Google Workspace, you can use it to authenticate employees in the Qrator Lab's account. In this case:

- there is no need to create separate user accounts in Qrator Labs;
- access management remains within your infrastructure;
- users sign in using their corporate credentials.

SSO workflow

Below is a simplified diagram of user authentication via SSO using OpenID Connect:

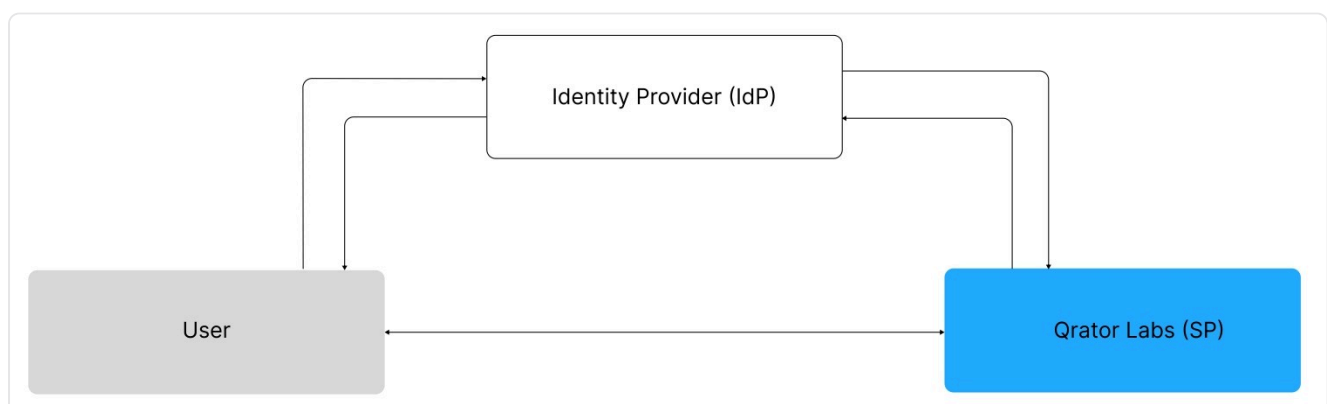


Diagram components

- User - an employee of the organization

- Qrator Labs (SP) - the service providing access
- Identity Provider (IdP) - the authentication system

Authentication flow

1. The user initiates sign-in via SSO;
2. Qrator Labs redirects the user to the IdP;
3. The user authenticates;
4. The IdP returns the user with a token;
5. Qrator Labs verifies the token and obtains the user data;
6. Group mapping is applied;
7. The user is granted access.

Key features

- Authentication takes place entirely on the IdP side;
- Qrator Labs does not access the user's password and connects to the IdP from the IP address 185.65.149.86;
- The interaction is implemented via redirect (through the browser);
- Access is determined based on the attributes sent by the IdP;
- Existing groups will be overwritten with the groups that the IdP sends to Qrator Labs;
- Access to the account for previously created user accounts will be preserved.

Overview

SSO configuration through OpenID Connect is available in the Qrator Labs account.

After configuration:

- users authenticate through the IdP;
- local user accounts are not created;
- access is managed through groups and policies.

How OpenID Connect works

OpenID Connect is a protocol that adds authentication on top of OAuth 2.0.

Prerequisites

Before you start configuration:

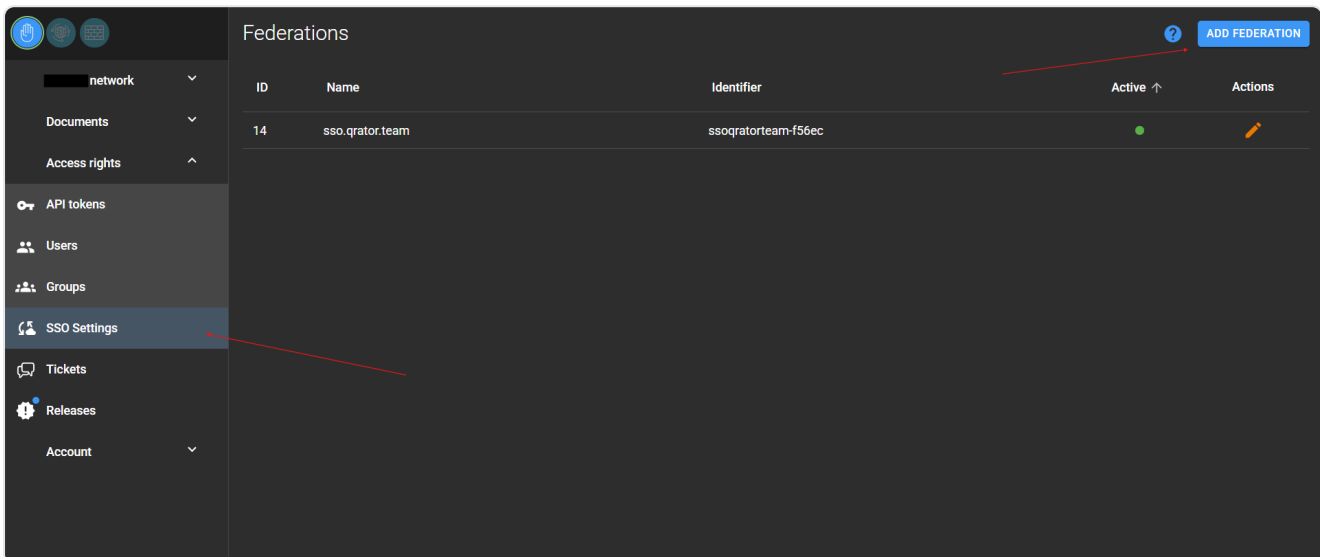
- an Identity Provider with OIDC support has been configured;
- the following are available:
 - Client ID
 - Client Secret

- OIDC endpoints or Discovery URL

Configuring SSO in Qrator Labs

1. Go to SSO settings

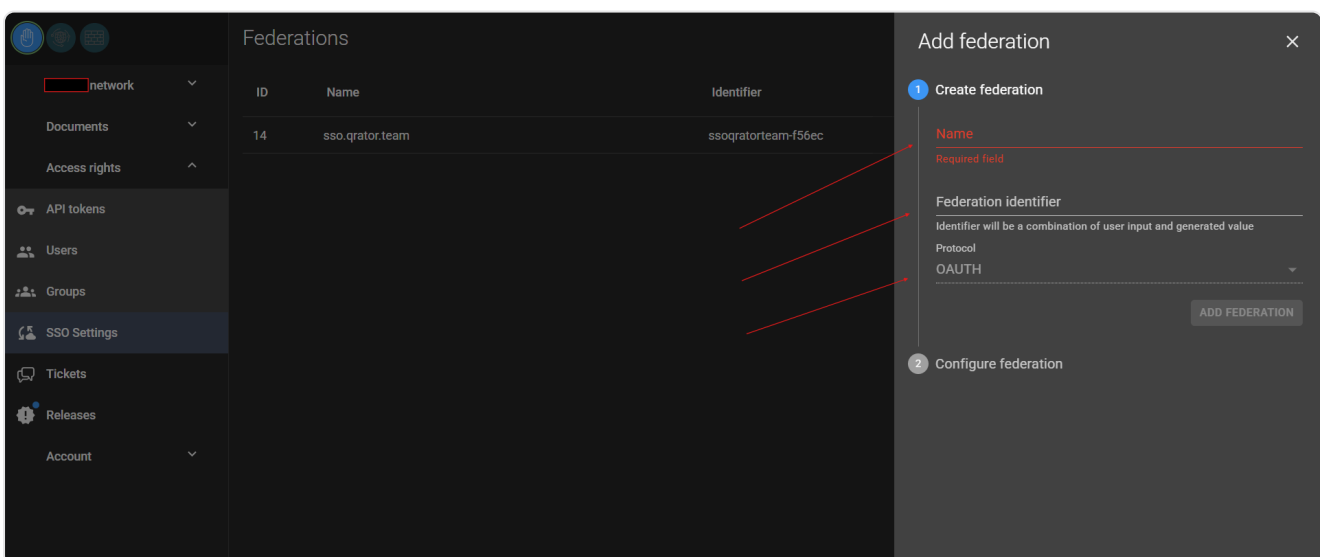
Go to the **SSO Settings** section and click **Add federation**.



2. Create a federation

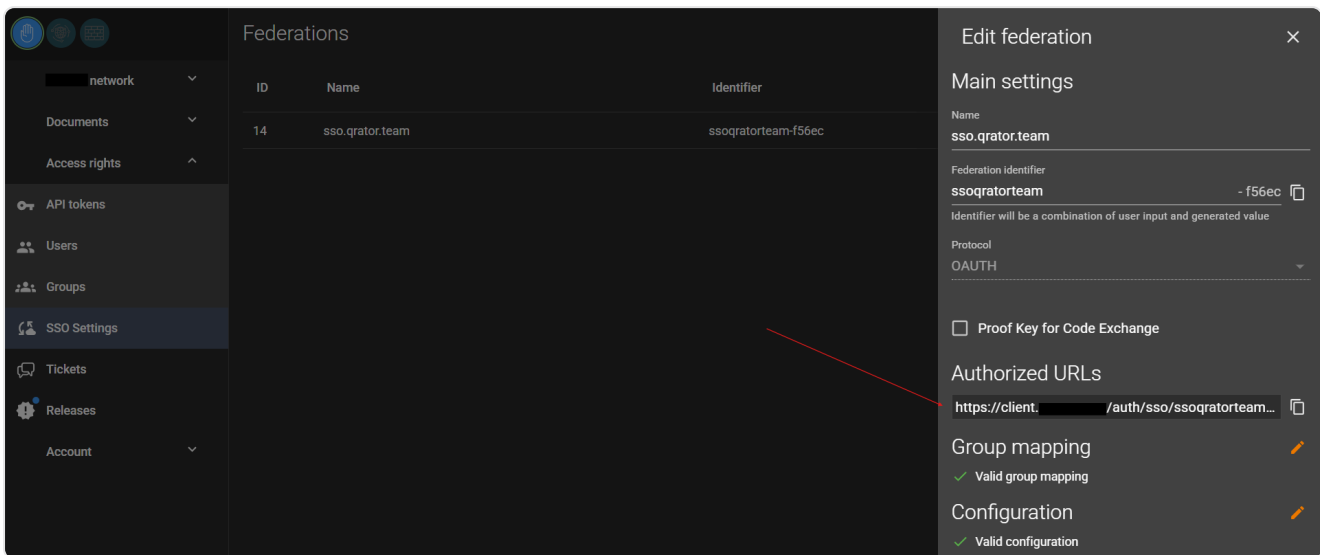
Specify the parameters:

- Name - any value;
- Federation identifier (slug). The identifier will be supplemented with a random suffix;
- Protocol - OAuth.



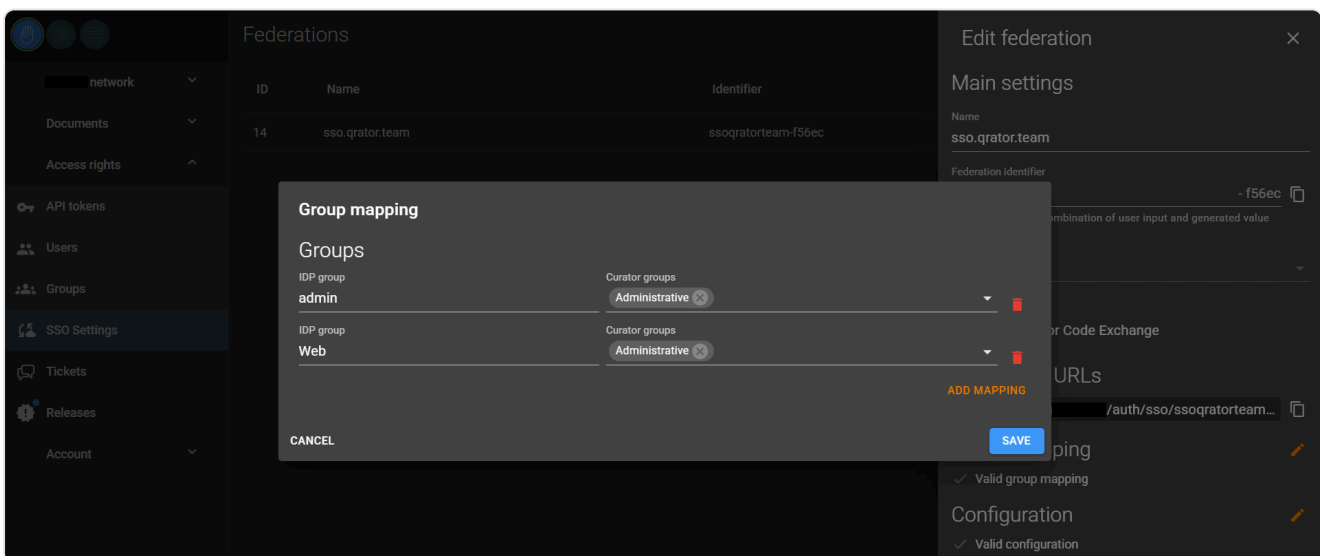
3. Obtain the Authorization URL

After the federation is created, the Authorization URL (Redirect URI) will be displayed. This URL must be specified in your Identity Provider settings.



4. Configure group mapping

Go to the Group binding section and configure the mapping between IdP groups and Qrator Labs groups.

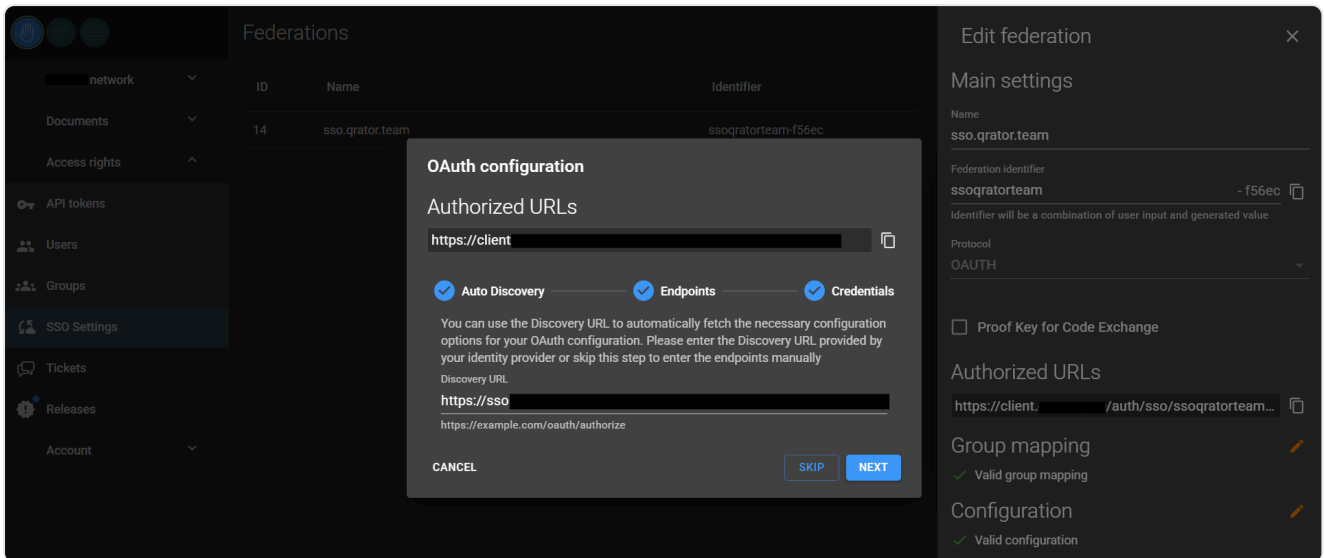


IMPORTANT! Without group binding configured, users signing in via SSO will not receive roles and groups. This step is required for correct operation.

Make sure the mapping includes a group that corresponds to the Qrator Labs group with user management permissions (Access Manager by default). Otherwise, administrators may lose access to user management at their next SSO sign-in.

5. Configure OAuth settings

Go to the **Configuration** section.

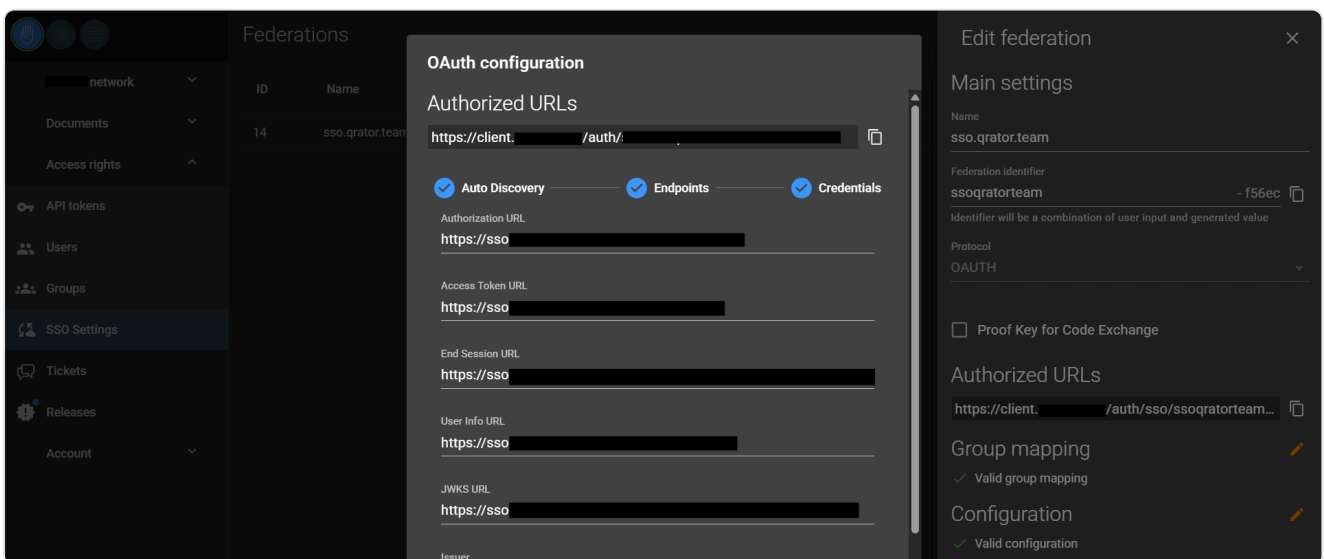


6. Specify endpoints

Specify your Identity Provider endpoints:

- Authorization Endpoint
- Token Endpoint
- UserInfo Endpoint
- JWKS Endpoint
- Logout Endpoint

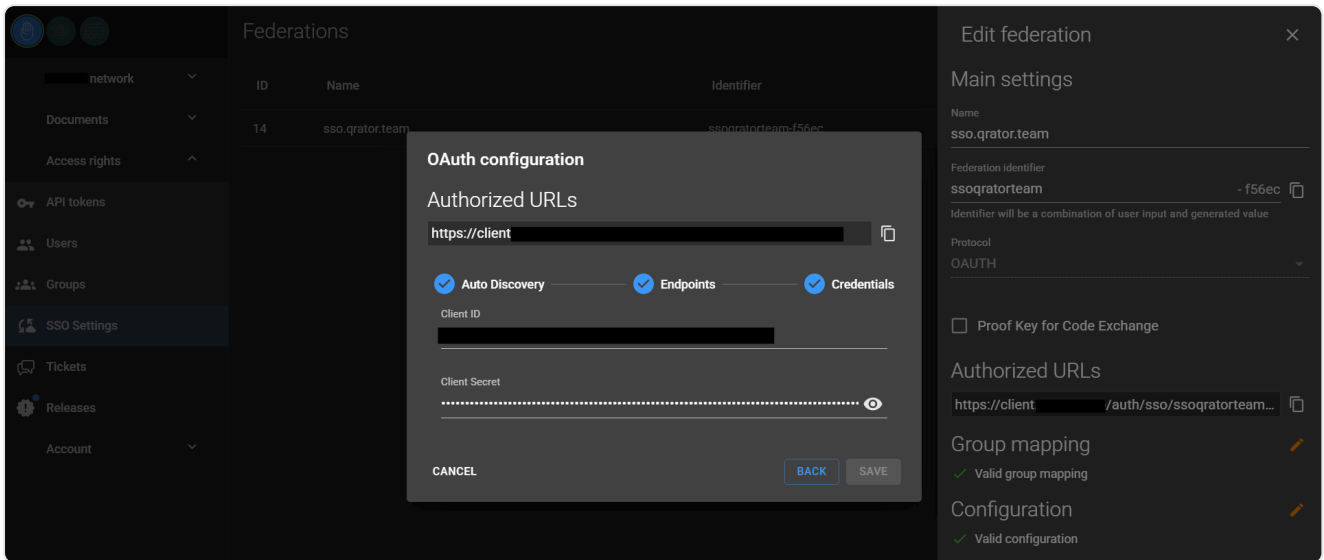
Or use the Discovery URL (`.well-known/openid-configuration`); the parameters will be populated automatically.



7. Specify credentials

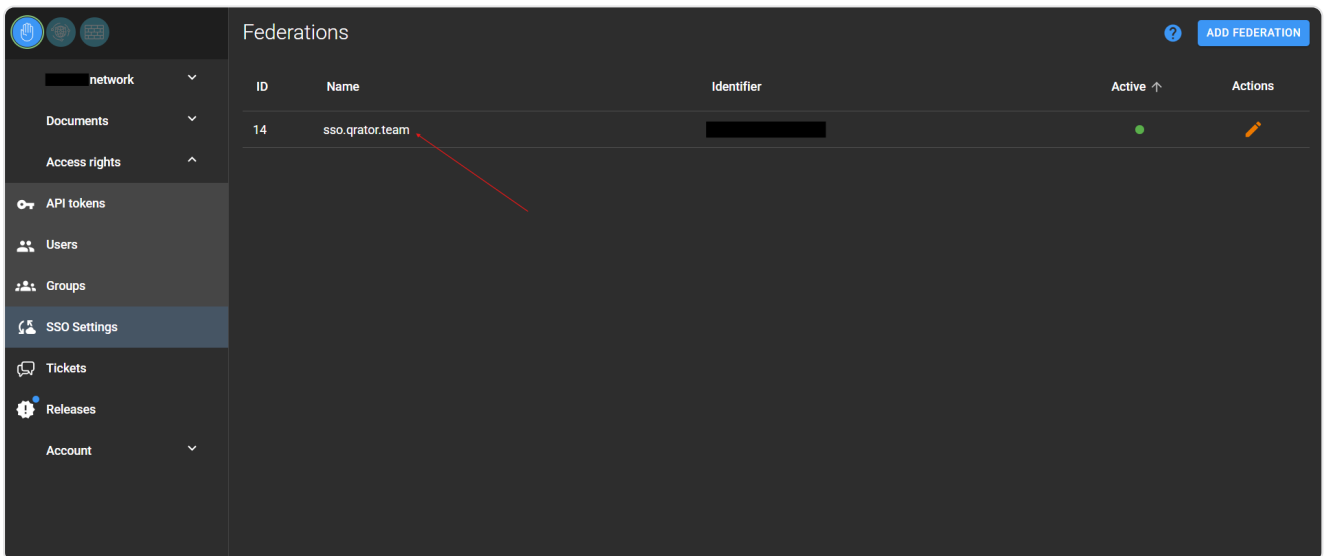
Enter the access parameters:

- Client ID
- Client Secret



8. Complete the configuration

Save the configuration. After that, the federation will appear in the list and become available for use.



Additional information

The account interface includes built-in instructions for configuring SSO.

The screenshot shows the QRATOR LABS dashboard with a sidebar on the left containing navigation links: Curator network, Documents, Access rights, API tokens, Users, Groups, SSO Settings (highlighted), Tickets, Releases, and Account. The main content area is titled 'Federations' and displays a table with one entry, ID 14. A modal window titled 'SSO connection via OAuth federation' is open, providing instructions for setting up SSO. The modal includes a description of single sign-on, a section for creating a federation with a sample callback URL, and a section for OAuth configuration with options for auto-discovery or manual endpoint entry.

Federations

ID
14

SSO connection via OAuth federation

Single sign-on into the client portal through your corporate Identity Provider (Keycloak, Okta, Azure AD, Google Workspace, or any other OAuth 2.0 / OIDC-compatible provider).

1. Create a federation

Enter a name and an identifier for the federation. The identifier will be appended with a random suffix. Protocol — **OAuth**.

Save the Callback URL — you will need it when registering the application in your IdP:

```
https://client[redacted]/auth/sso/<federation-id>/callback
```

2. OAuth configuration

Provide your IdP endpoints using one of two methods:

- Auto Discovery** — enter the Discovery URL (`.well-known/openid-configuration`) and all parameters will be populated automatically.
- Manual** — fill in the fields: `Authorization URL`, `Access Token URL`, `End Session URL`, `User Info URL`, `JWKS URL`, `Issuer`.

Then enter the **Client ID** and **Client Secret** provided by your IdP.

04

Help and support

[Additional HTTP headers](#)

[Terraform](#)

4.1 HELP AND SUPPORT

Additional HTTP headers

Reverse HTTP proxy and reverse WebSocket proxy add additional headers to every HTTP request proxied from the client.

The information contained in these headers can be useful for logging requests and subsequently debugging problems that arise.

- **X-Forwarded-For**

Example of value: 1.2.3.4, 5.6.7.8

Contains the IP address from where the request came to the Qrator Labs reverse proxy, as well as a list of any previous IP addresses added to this header, if any.

If the request that came to the Qrator Labs reverse proxy did not have an **X-Forwarded-For** header, it will be added. Such a header will only contain the IP address of the user from where the request came.

If the request that came to Qrator Labs reverse proxy already had an **X-Forwarded-For** header, the IP address from where the request came will be added to the existing content. So, the header will contain both the user's IP address and the addresses of all the proxies through which the user reached the Qrator Labs reverse proxy. Commas followed by spaces are used as separators.

**NOTE**

If some hardware on your network also modifies the **X-Forwarded-For** header before passing the request to the server, it may be difficult to determine which of the listed IP addresses the request came to Qrator Labs from. If you need to specify this accurately, use the **X-Qrator-IP-Source** header.

- **X-Q-DomId**

Example of value: 11169

Contains numeric ID of the domain that is being proxied. It is the same as the ID specified in the list of domains in the dashboard, but is specified without the letter d in the HTTP header.

- **X-Qrator-IP-Source**

Example of value: 1.2.3.4

Contains IP address of the user from where the request came. Matches the last IP address in the **X-Forwarded-For** header.

- **X-Qrator-RequestID**

Example of value: f4b16a6bf0bcc8bebc394a2824d5d569

Contains a unique ID of the processed request. If any problem you have detected is related to specific requests, you can provide the IDs of these requests to technical support for detailed review.

- **X-Qrator-TCP-Info**

Example of value: 51890, 69493, 10151

Contains information about TCP sessions. Consists of three numbers.

The first number is the sender port from where TCP segments were sent to the Qrator Labs network.

The second and third numbers are RTT and RTTVar values calculated according to [RFC 2988](#).

Connecting the Terraform Provider for Qrator

The Terraform Provider for Qrator allows you to manage Qrator resources through Terraform configurations instead of manually configuring them in the interface. The provider is published in the Terraform Registry as [QratorLabs/qrator](#), supports Terraform 1.0+, and works with domains, services, CDN configurations, and client certificates.

Prepare the Terraform configuration

Add the provider to the `required_providers` block:

```
terraform {
  required_providers {
    qrator = {
      source = "qratorlabs/qrator"
      version = "~> 0.2"
    }
  }
}
```

Configure authentication

The provider requires an API key to access the Qrator API. Authentication can be set either in the provider block or via environment variables. The README provides both options: `api_key` and `endpoint` in the configuration, or `QRATOR_API_KEY` and `QRATOR_ENDPOINT` in the environment.

Example using a provider block:

```
provider "qrator" {
  api_key = var.api_key
  endpoint = "https://api.qrator.net"
}
```

Example using environment variables:

- `export QRATOR_API_KEY="your-api-key"`
- `export QRATOR_ENDPOINT="https://api.qrator.net"`

Initialize Terraform

After adding the provider, perform standard initialization: `terraform init`.

Then, the provider follows the usual Terraform cycle: plan to check for changes and apply to apply the configuration.

What can be managed through the provider

The provider supports the following resource groups:

Domains

- `qrator_domain`
- `qrator_domain_services`
- `qrator_domain_sni`
- `qrator_domain_whitelist`
- `qrator_domain_blacklist`

Services

- `qrator_service`
- `qrator_service_services`
- `qrator_service_sni`
- `qrator_service_whitelist`
- `qrator_service_blacklist`

CDN and Certificates

- `qrator_cdn`
- `qrator_cdn_sni`
- `qrator_client_certificate`

Operation Scenarios

Two main scenarios are supported.

Scenario 1: Create a new configuration

If the domain, service, or CDN configuration has not yet been created, describe them in `.tf` files and run:

1. `terraform plan`
2. `terraform apply`

The process is described in detail in the [documentation](#).

Scenario 2: Connecting existing resources

If resources already exist in Qrator, they can be managed by Terraform without recreating them.

The basic workflow is:

1. Describe the resource in `.tf`;
2. Run `terraform import`;
3. Verify the `terraform plan`;
4. Match the `.tf` configuration to the actual state;
5. Run `terraform apply`.

The process is described in detail in the [documentation](#).

Recommended Workflow

For practical use, the following workflow is recommended:

New resource

1. `terraform init`
2. `terraform plan`
3. `terraform apply`

Existing resource

1. `terraform init`
2. `terraform import`
3. `terraform plan`
4. `terraform apply`

This workflow is required to first synchronize the Terraform state with the current state in Qrator and then transfer the resource to a managed configuration.

Important considerations during operation

- The provider operates through the Qrator API, so a valid API key is required for connection.
- For local provider development, you can use `dev_overrides` in `~/.terraformrc`.
- The repository contains an `examples/` directory with ready-made configuration examples,
- which can be used as a starting point.

Useful links

Official provider [repository](#): README, requirements, authentication, resource list.



Up-to-date documentation — docs.qrator.net