opEvents Realtime Events

> Advanced level feature: skills with Apache or Nginx web server configuration and SSL are required.
>
> See the 'Current Issues' section below for details of problems you may encounter if you enable this feature.

In opEvents 3.3.0, we have introduced realtime events to push updates as they happen to the GUI.

Currently this is an opt-in feature and requires setting `"opevents_realtime_gui"` to `"true"`, in the file `conf/opCommon.json` and restarting the server.

```
sudo vi /usr/local/omk/conf/opCommon.json
```

OR

```
sudo /usr/local/nmis9/admin/patch_config.pl -r /usr/local/omk/conf/opCommon.json opevents_realtime_gui
```

if `false` run below commands to set to `true` and restart the server

```
sudo /usr/local/nmis9/admin/patch_config.pl /usr/local/omk/conf/opCommon.json /opevents
/opevents_realtime_gui=true

sudo service omkd restart
```

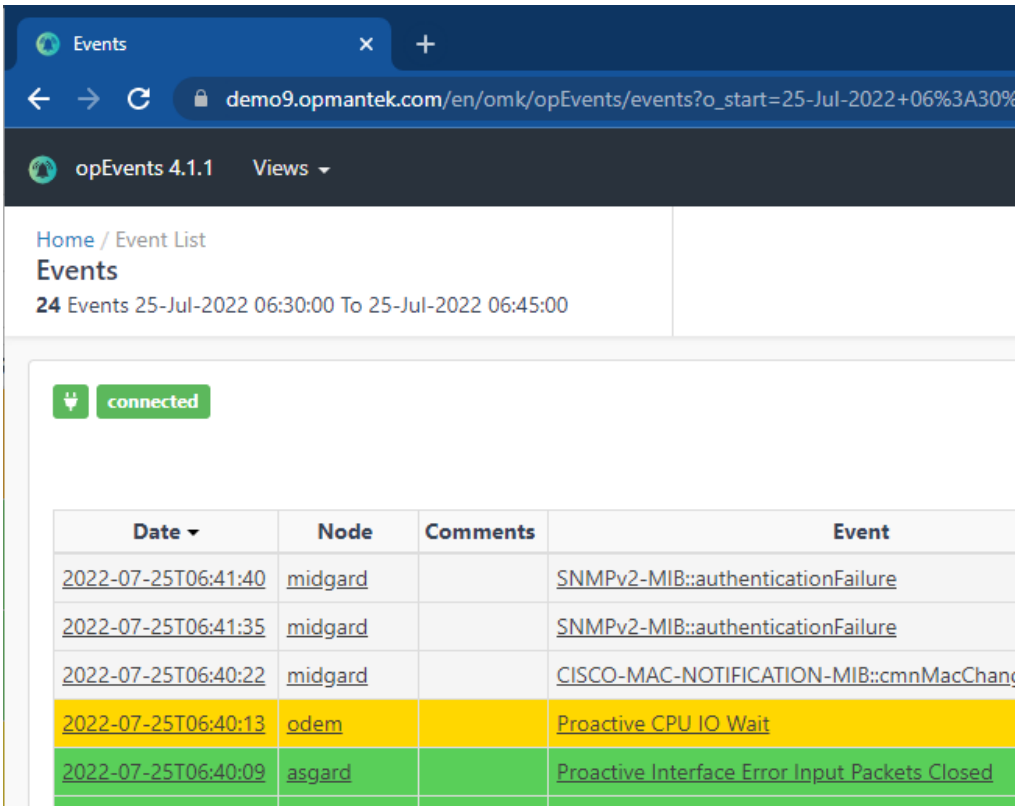
If you are using https for opEvents, you will also have to configure your Apache server and restart it, see below for details.

When a new event is created, either after parsing or via create event the event details are pushed onto a queue.

Updates to the event as it passes through the EventAction pipeline and the key or keys "priority" "acknowledged" "action_checked" "notes" "status" changes, these updates will be pushed into the queue.

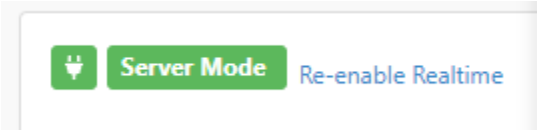
- When an event is updated we match the changed keys to then produce the web socket event to update the GUI, some users may have custom keys in the event and may be calling updateEvent through plugins r
  - Using '*opevents_realtime_push_on_key*' you can add extra event keys to this array to make sure the GUI updates on changes.

What it looks like when it is working:



What does **Re-enable Realtime** mean?

Sometimes, the real time updates cannot be continued, for example if you reverse the sort order of the Date column, so the following is displayed:

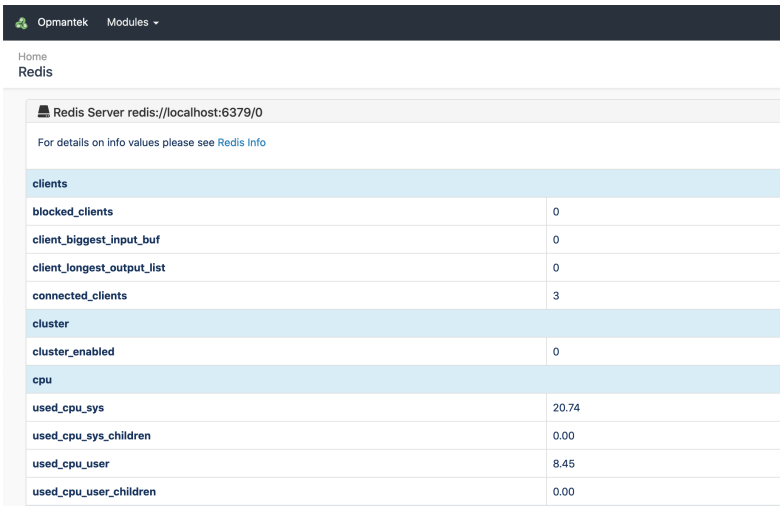


Currently Supported Screens

- Current Events
- Events

Checking Redis Status

Under Help Redis Info you will find debug information about the configured Redis Instance



Redis Issues

For any reason the opEvents log or the Redis status is showing an error please see our page on debugging Redis Redis and Opmantek Applications

Apache Config Changes

HTTPS Realtime Events

When connecting over ssl you will need the web-socket connect to also be secured as the browser cannot run mixed content, secured page and unsecured socket connection. We can use Apache and the optional module proxy_wstunnel to terminate the secured connection and then proxy the connection to the OMKD web server.

The minimum supported Apache Version is 2.4.6, We recommend you use a virtual host and the provided Apache configuration 04omk-proxy.conf under Redhat: /etc/httpd/conf.d/04omk-proxy.conf Debian: /etc/httpd/conf-enabled/04omk-proxy.conf is not currently setup for virtual hosts. Removing the provided 04omk-proxy.conf requires you have basic understanding on editing Apache config. ***NOTE:*** Re-installing, or upgrading opEvents will restore this file, so it will need to be removed once again after a re-install.

Debian 9

Enable `proxy_wstunnel`

Enable these modules to support proxying of the websockets.

```
a2enmod proxy
a2enmod proxy_http
a2enmod proxy_wstunnel
```

Then restart Apache

```
sudo systemctl restart apache2
```

Next you will need to edit /etc/httpd/conf-enabled/04omk-proxy.conf

We need to tell the omk server application the connection is being proxied and the client has connected over https, Find RequestHeader and change from **http** to **https**

```
RequestHeader set X-Forwarded-Proto "https"
```

Above <Location "/omk"> add the following line, if you are using other languages please change "en" to your specified language, or add more entries.

```
ProxyPassMatch ^(\/(en|es)\/omk\/opEvents\/ws\/.*)$  ws://localhost:8042/$1
```

Restart Apache

```
sudo systemctl restart apache2
```

RedHat 7 & Centos 7

```
yum install mod_ssl
```

Enable `proxy_wstunnel`

Edit `/etc/httpd/conf.modules.d/00-proxy.conf`

All modules related to proxying websockets are listed in this configuration file, please uncomment:

```
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_wstunnel modules/mod_proxy_wstunnel.so
```

Then restart Apache

```
sudo systemctl restart httpd
```

`Create a new VirtualHost`

You will need to create a virtual host for proxying web sockets on Redhat, the shipped proxy file our installer / vm ships /etc/httpd/conf.d/04omk-proxy.conf is not compatible and should be removed from Apaches conf.d/ directory.

You should understand how a virtual host works, please see https://httpd.apache.org/docs/2.4/vhosts/examples.html

A basic example with config to use serve opEvents over SSL and proxy the Webscockets, create a new file in /etc/httpd/conf.d/omkd_ssl.conf

Apache will listen on port 443, serve SSL, proxy the websockets and main application to the OMKD web server listening on localhost 8042

Apache will also redirect requests from 80 to 443 to make sure no users can access the application without SSL

```
<VirtualHost *:443>
        ServerName example.opmantek.com

        SSLEngine on
        SSLProxyEngine On
        ProxyRequests Off

        SSLCertificateFile    /etc/ssl/certs/example/cert.pem
        SSLCertificateKeyFile /etc/ssl/certs/example/privkey.pem
        SSLCertificateChainFile /etc/ssl/certs/example/fullchain.pem

        RequestHeader set X-Forwarded-Proto "https"

                # Proxy the websocket connection and rewrite the header
        RewriteEngine On
        RewriteCond %{REQUEST_URI} ^/en/omk/opEvents/ws/(.*)
        RewriteRule /(.*) ws://localhost:8042/en/omk/opEvents/ws/%1 [P,L]

        # Proxy the rest of the application
        ProxyPass /en/omk http://localhost:8042/en/omk
        ProxyPass /es/omk http://localhost:8042/es/omk
        ProxyPass /omk http://localhost:8042/omk
        ProxyPassReverse / http://localhost:8042/
                ErrorDocument 503 '<html><head><meta http-equiv="refresh" content="60"><
/head><body><h1>Temporary Service Interruption</h1>The requested OMK page should be back soon. This page will
automatically reload in 60 seconds.</body></html>'

</VirtualHost>
<VirtualHost *:80>
    ServerName example.opmantek.com
    Redirect 301 / https://example.opmantek.com/
</VirtualHost>
```

Settings which you will need to modify from the example

Name	Value	Example	Apache Docs
ServerName	FQDN of the server which users will refer to it by	monit-prod.opmantek.com	https://httpd.apache.org/docs/2.4/vhosts/name-based.html
SSLCertificateFile	Server PEM-encoded X.509 certificate data file or token identifie	/etc/ssl/certs/example/cert.pem	https://httpd.apache.org/docs/current/mod/mod_ssl.html#sslcertificatefile
SSLCertificateKeyFile	Server PEM-encoded private key file	/etc/ssl/certs/example/privkey.pem	https://httpd.apache.org/docs/current/mod/mod_ssl.html#sslcertificatekeyfile
SSLCertificateChainFile	(Before apache 2.4.8) File of PEM-encoded Server CA Certificates	/etc/ssl/certs/example/fullchain.pem	https://httpd.apache.org/docs/current/mod/mod_ssl.html#sslcertificatekeyfile
Redirect 301	HTTPS url of the server which your users refer to by	https://example.opmantek.com/	

`Testing the config`

Then restart Apache

```
sudo systemctl restart httpd
```

Nginx Config Changes

We now support Nginx 1.18.0 and above and this can be used if you wish to switch to nginx over apache regardless of your linux distribution.

Ubuntu 20.04

This configuration is also to ensure you can proxy websocket connections for ubuntu 20.04 and over, Ubuntu does not support the required apache2 version needed for opevents realtime gui so if you wish to enable this feature and use Ubuntu you will need to follow these steps.

```
sudo apt-get install nginx
sudo apt install fcgiwrap
```

In /etc/nginx/sites-available/, create the main configuration file:

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}

server {
    listen 80;
    server_name your_server_name;

    location / {
        if ($host != localhost) {
            rewrite ^(.*)$ https://$host$request_uri permanent;
        }
    }

    location /nmis9 {
        alias /usr/local/nmis9/htdocs;
        index index.html;
    }

    location = /nmis9/ {
        rewrite ^ /cgi-nmis9/nmiscgi.pl permanent;
    }

    location /menu9/ {
        alias /usr/local/nmis9/menu/;
    }

    location /cgi-nmis9/ {
        alias /usr/local/nmis9/cgi-bin/;
        include fastcgi_params;
        fastcgi_pass unix:/var/run/fcgiwrap.socket;
        fastcgi_param SCRIPT_FILENAME $request_filename;
    }
}

server {
    listen 443 ssl http2;
    server_name your_server_name;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Proto https;
    ssl_certificate /path/to/ssl_cert;
    ssl_certificate_key /path/to/ssl_key;

    include common_nmis_locations;

    location = / {
        return 301 $scheme://$host/omk;
    }

    location ~ ^/(en|es)/omk/opEvents/events/.* {
        include common_proxy_headers;
        proxy_pass http://localhost:8042;
    }

    location /en/omk/opCharts/events/log {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/en/omk/opCharts/events/log;
    }

    location /en/omk/opEvents/ws/test {
```

```
        include common_proxy_headers;
        proxy_pass http://localhost:8042;
    }

    location /en/omk/opEvents/ws/events {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/en/omk/opEvents/ws/events;
    }

    location /es/omk/opCharts/events/log {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/en/omk/opCharts/events/log;
    }

    location /es/omk/opEvents/ws/test {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/en/omk/opEvents/ws/test;
    }

    location /es/omk/opEvents/ws/events {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/en/omk/opEvents/ws/events;
    }

    location /omk {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/omk;
        error_page 503 '<html><head><meta http-equiv="refresh" content="60"></head><body><h1>Temporary Service
Interruption</h1>The requested OMK page should be back soon. This page will automatically reload in 60 seconds.<
/body></html>';
    }

    location /omk.json {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/omk.json;
    }

    location /es {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/es;
        error_page 503 '<html><head><meta http-equiv="refresh" content="60"></head><body><h1>Temporary Service
Interruption</h1>The requested OMK page should be back soon. This page will automatically reload in 60 seconds.<
/body></html>';
    }

    location /en {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/en;
        error_page 503 '<html><head><meta http-equiv="refresh" content="60"></head><body><h1>Temporary Service
Interruption</h1>The requested OMK page should be back soon. This page will automatically reload in 60 seconds.<
/body></html>';
    }

    location /pt {
        include common_proxy_headers;
        proxy_pass http://localhost:8042/pt;
        error_page 503 '<html><head><meta http-equiv="refresh" content="60"></head><body><h1>Temporary Service
Interruption</h1>The requested OMK page should be back soon. This page will automatically reload in 60 seconds.<
/body></html>';
    }
}
```

At the end of the first server block for port 80, please check **fastcgi_pass unix:/var/run/fcgiwrap.socket;**

and make sure that this is the correct path your fcgi.socket, when you install fcgiwrap the path to fcgiwrap.socket will differ depending on your distribution:

Ubuntu/Debian: */var/run/fcgiwrap.socket*
CentOS/RHEL: */usr/lib/systemd/system/fcgiwrap.socket*

Settings which you will need to modify from the example

Name	Value	Example	Apache Docs
ServerName	FQDN of the server which users will refer to it by	monit-prod.opmantek.com	https://httpd.apache.org/docs/2.4/vhosts/name-based.html
SSLCertificateKeyFile	Server PEM-encoded private key file	/etc/ssl/certs/example/privkey.pem	https://httpd.apache.org/docs/current/mod/mod_ssl.html#sslcertificatekeyfile
SSLCertificateChainFile	(Before apache 2.4.8) File of PEM-encoded Server CA Certificates	/etc/ssl/certs/example/fullchain.pem	https://httpd.apache.org/docs/current/mod/mod_ssl.html#sslcertificatekeyfile
fcgiwrap.socket	allows you to set up a socket for communication between a web server and `fcgiwrap` to handle FastCGI requests	/var/run/fcgiwrap.socket	

Next create two configuration files in the main nginx directory: /etc/nginx. One of these configs will be called **common_proxy_headers** and will contain:

```
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection $connection_upgrade;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Host $host;
proxy_set_header X-Forwarded-Port $server_port;
```

The other will be common_nmis_locations and will contain:

```
    location /nmis9 {
        alias /usr/local/nmis9/htdocs;
        index index.html;
    }

    location = /nmis9/ {
        rewrite ^ /cgi-nmis9/nmiscgi.pl permanent;
    }

    location /menu9/ {
        alias /usr/local/nmis9/menu/;
    }

    location /cgi-nmis9/ {
        alias /usr/local/nmis9/cgi-bin/;
        include fastcgi_params;
        fastcgi_pass unix:/var/run/fcgiwrap.socket;
        fastcgi_param SCRIPT_FILENAME $request_filename;
    }
```

*note, you do not need the common_nmis_locations and can include this block into the 443 server block if you wish, this ensures no duplicated entry's though and is a more santitized configuration. You **MUST** include the proxy directives as a seperate configuration file, as any incorrect order or misconfiguration of proxy derectives can easily break wss:// headers in nginx.

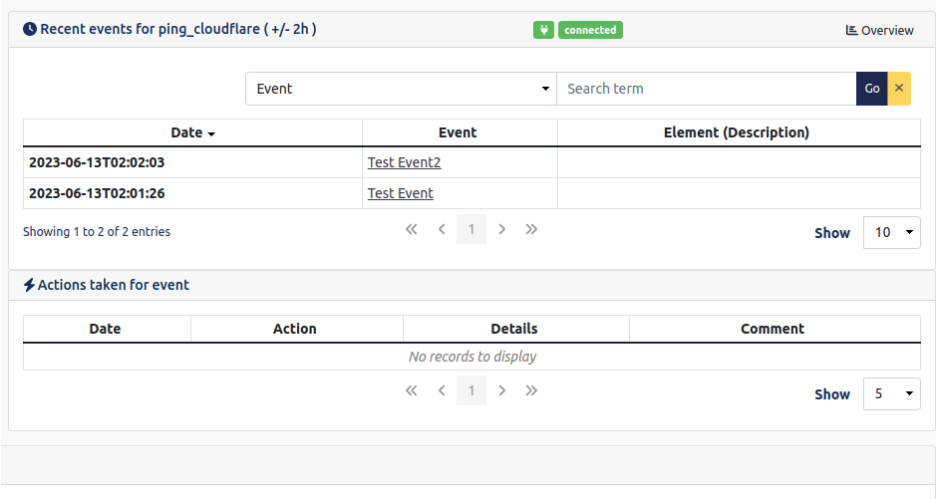
Create a symbolic link from /etc/nginx/sites-available/your_config to /etc/nginx/sites-enabled for the master configuration file

```
ln -s /etc/nginx/sites-available/your_config /etc/nginx/sites-enabled/your_config
```

Restart nginx

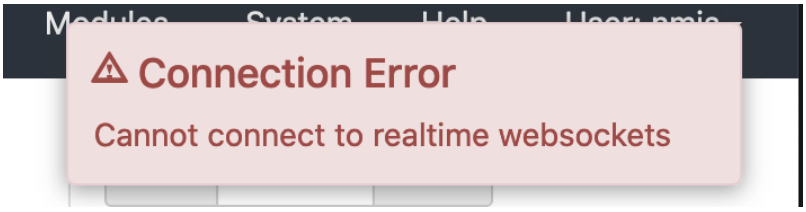
```
sudo systemctl restart nginx
or
sudo service nginx restart
```

And test realtime events connects and works

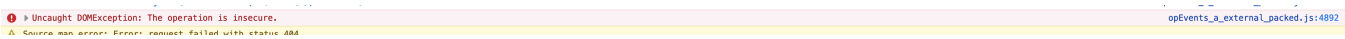


Debugging Web Socket connections

If you see this error it means the browser cannot connect to the webserver, at this current point you will need to use the Browsers developments tools to access the Javascript Console



Uncaught DomException: The operation is insecure

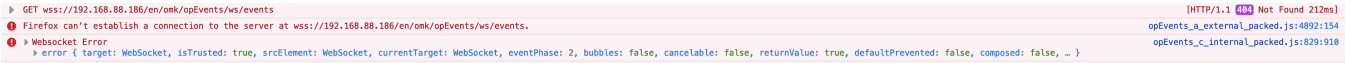


You are serving opEvents over ssl but the websocket is trying to to connect over http, this is disallowed from happening because of browser security policies.

You will need to find your Apache Config and make sure this header is being sent to the omkd web server

```
RequestHeader set X-Forwarded-Proto "https"
```

Get /en/omk/opEvents/ws/events 404 Not Found



The websocket is trying to connect securely but its not being proxied correctly, please see the above section Proxy the websocket

Check Apache has proxy_wstunnel_module loaded

```
httpd  -t -D DUMP_MODULES | grep proxy_wstunnel_module
```

Current Issues

The following are known issues encountered when enabling this feature and will be addressed in a future release:

1. Event partial updates over websockets eg {id:abc, acknowledged:1} have no group check, meaning if you have a user with only group access of "DataCenter" their browsers websocket would receive partial updates from events their group permission would not allow access to. As the users browsers doesn't have the original event the partial update will not be shown to the user.
2. Node Context Links are not rendered with realtime events.
3. If you enable the `opevents_realtime_gui` without configuring Apache correctly, the search on the Events form will not work, you need to configure Apache.

See Also

Events Pane in the Node View - enabling websocket when using SSL or TLS