

[How-To] Use Uptime Kuma Behind Reverse Proxy

Purpose

This document aims to show how to configure your reverse proxy configuration for Uptime Kuma as it is a web socket app.

Prerequisites

List of prerequisites:

- Root user or sudo user
- Uptime Kuma Server

Reverse Proxy Configuration

Nginx Reverse Proxy:

For Nginx with SSL:

```
server {  
    listen 443 ssl http2;  
    # Remove '#' in the next line to enable IPv6  
    # listen [::]:443 ssl http2;  
    server_name sub.domain.com;  
    ssl_certificate    /path/to/ssl/cert/crt;  
    ssl_certificate_key /path/to/ssl/key/key;  
    # *See "With SSL (Certbot)" below for details on automating ssl certificates  
  
    location / {  
        proxy_set_header    X-Real-IP $remote_addr;  
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
```

```
proxy_set_header Host $host;
proxy_pass      http://localhost:3001/;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade";
}
}
```

For Nginx with SSL (Certbot):

```
server {
    # If you don't have one yet, you can set up a subdomain with your domain registrar (e.g. Namecheap)
    # Just create a new host record with type='A Record', host='<subdomain>', value='<ip_address>'.

    server_name your_subdomain.your_domain.your_tld;

    location / {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
        proxy_pass      http://localhost:3001/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

# Once that's completed, you can run
# sudo apt install python3-certbot-nginx
# sudo certbot --nginx -d your_domain -d your_subdomain.your_domain -d www.your_domain
# And Certbot will auto-populate this nginx .conf file for you, while also renewing your certificates automatically
in the future.
```

For Nginx without SSL:

```
server {
    listen 80;
    # Remove '#' in the next line to enable IPv6
```

```
# listen [::]:80;

server_name sub.domain.com;

location / {
    proxy_pass http://localhost:3001;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
}

}
```

Apache Reverse Proxy:

For Apache With SSL:

```
<VirtualHost *:443>
    ServerName sub.domain.com
    SSLEngine On
    SSLCertificateFile /path/to/ssl/cert/crt
    SSLCertificateKeyFile /path/to/ssl/key/key
    # Protocol 'h2' is only supported on Apache 2.4.17 or newer.
    Protocols h2 http/1.1
    ProxyPreserveHost on
    ProxyPass / http://localhost:3001/
    RewriteEngine on
    RewriteCond %{HTTP:Upgrade} =websocket
    RewriteRule /(.*) ws://localhost:3001/$1 [P,L]
    RewriteCond %{HTTP:Upgrade} !=websocket
    RewriteRule /(.*) http://localhost:3001/$1 [P,L]
</VirtualHost>
```

For Apache Without SSL:

```
<VirtualHost *:80>
    ServerName sub.domain.com
    ProxyPreserveHost on
    ProxyPass / http://localhost:3001/
    RewriteEngine on
    RewriteCond %{HTTP:Upgrade} websocket [NC]
    RewriteCond %{HTTP:Connection} upgrade [NC]
```

```
RewriteRule ^/?(.*) "ws://localhost:3001/$1" [P,L]
</VirtualHost>
```

Caddy Reverse Proxy:

Caddy Normal:

```
subdomain.domain.com {
    reverse_proxy 127.0.0.1:3001
}
```

Caddy with Docker-Compose:

```
version: '3'
networks:
  default:
    name: 'proxy_network'
services:
  uptime-kuma:
    image: louislam/uptime-kuma:1
    restart: unless-stopped
    volumes:
      - /srv/uptime:/app/data
  labels:
    caddy: status.example.org
    caddy.reverse_proxy: "*" {{upstreams 3001}}"
  caddy:
    image: "lucaslorentz/caddy-docker-proxy:ci-alpine"
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /srv/caddy/:/data
    restart: unless-stopped
    environment:
      - CADDY_INGRESS_NETWORKS=proxy_network
```

HTTPS-Portal Reverse Proxy:

Https Normal:

```
version: '3.3'

services:
  https-portal:
    image: steveltn/https-portal:1
    ports:
      - '80:80'
      - '443:443'
    links:
      - uptime-kuma
    restart: always
    environment:
      DOMAINS: 'status.domain.com -> http://uptime-kuma:3001'
      STAGE: 'production' # Don't use production until staging works
      # FORCE_RENEW: 'true'
      WEBSOCKET: 'true'
    volumes:
      - https-portal-data:/var/lib/https-portal

  uptime-kuma:
    image: louislam/uptime-kuma:1
    container_name: uptime-kuma
    volumes:
      - ./uptime-kuma:/app/data
    ports:
      - 3001:3001

volumes:
  https-portal-data:
```

HAProxy:

No special configuration is required when using HAProxy as a reverse proxy although you may wish to add the `timeout tunnel` option to either the `defaults`, `listen`, or `backend` sections. If using the `timeout tunnel` option, it is also recommended to set `timeout client-fin` to handle instances where the client stops responding.

Read more: <http://cbonte.github.io/haproxy-dconv/2.4/configuration.html#4.2-timeout%20tunnel>

<https://github.com/louislam/uptime-kuma/wiki/Reverse-Proxy#apache>

Revision #1

Created 3 November 2024 20:09:24 by Mike Leffring

Updated 3 November 2024 20:09:49 by Mike Leffring