Uptime Kuma

HTTPS Setup Guide

Nginx Reverse Proxy with Internal AD Certificate Authority

Complete Step-by-Step Guide
This guide provides detailed instructions for securing Uptime Kuma with HTTPS using Nginx as a reverse proxy and certificates from your internal Active Directory Certificate Authority.

Document Version: 1.0
Last Updated: February 07, 2026

Table of Contents

Prerequisites

Before beginning this guide, ensure you have the following:

- Uptime Kuma already running in Docker
- Ubuntu server with sudo/root access
- DNS record pointing to your server (e.g., uptime.yourdomain.com)
- Access to your Windows Active Directory Certificate Authority
- Basic understanding of Linux command line
- Ports 80 and 443 available on your server

> **Important:** This guide assumes Uptime Kuma is already installed and running in Docker. All commands should be run on your Ubuntu server unless otherwise specified.

Step 1: Verify Uptime Kuma Docker Configuration

First, we need to ensure that Uptime Kuma is properly exposing port 3001 to the host system.

Check your running container:

```
docker ps
```

Look for `0.0.0.0:3001->3001/tcp` or `:::3001->3001/tcp` in the PORTS column.

If port 3001 is NOT exposed:

You need to stop and recreate the container with port exposure:

```
# Stop existing container
docker stop uptime-kuma
docker rm uptime-kuma

# Start with port exposed
docker run -d \
  --name uptime-kuma \
  --restart=always \
  -p 3001:3001 \
  -v uptime-kuma:/app/data \
  louislam/uptime-kuma:1
```

Verify Uptime Kuma is accessible:

```
curl http://localhost:3001
```

You should see HTML output from Uptime Kuma.

Step 2: Generate Private Key and CSR with SAN

Modern browsers require Subject Alternative Names (SAN) in certificates. We'll create a CSR with SAN support.

> **Important:** Subject Alternative Names (SAN) are required by modern browsers. A certificate with only Common Name will trigger security warnings.

Create SSL directory:

```
sudo mkdir -p /etc/nginx/ssl
cd /etc/nginx/ssl
```

Generate private key:

```
sudo openssl genrsa -out uptime-kuma.key 2048
```

Create CSR configuration file with SAN:

```
sudo nano /etc/nginx/ssl/csr.conf
```

Paste the following (replace uptime.yourdomain.com with your actual FQDN):

```
[req]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn
req_extensions = v3_req

[dn]
C=US
ST=VA
L=YourCity
O=YourCompany
OU=IT
CN=uptime.yourdomain.com

[v3_req]
subjectAltName = @alt_names

[alt_names]
DNS.1 = uptime.yourdomain.com
```

Save and exit (Ctrl+X, Y, Enter).

Generate CSR with SAN:

```
sudo openssl req -new -key uptime-kuma.key \
  -out uptime-kuma.csr \
  -config csr.conf
```

Verify the CSR includes SAN:

```
sudo openssl req -text -noout -verify -in uptime-kuma.csr | grep -A 1 'Subject Alternative Name'
```

You should see your DNS name listed under Subject Alternative Name.

Step 3: Submit CSR to AD Certificate Authority

Display your CSR:

```
sudo cat uptime-kuma.csr
```

Copy the entire output including the BEGIN and END lines.

Method 1: Web Enrollment (Most Common)

1. Open browser to: `https://your-ca-server/certsrv`
2. Click "Request a certificate"
3. Click "Advanced certificate request"
4. Click "Submit a certificate request by using a base-64-encoded..."
5. Paste your CSR into the text box
6. Select "Web Server" from the Certificate Template dropdown
7. Click Submit
8. Download the certificate (Base 64 encoded format)

Method 2: Windows Command Line

Copy the CSR file to a Windows machine with CA access:

```
certreq -submit -attrib "CertificateTemplate:WebServer" uptime-kuma.csr
```

Step 4: Download and Install Certificate

After your CA approves the request, download the certificate and copy it to your Ubuntu server.

From your workstation:

```
scp uptime-kuma.cer user@ubuntu-server-ip:/tmp/
```

On Ubuntu server:

```
sudo mv /tmp/uptime-kuma.cer /etc/nginx/ssl/uptime-kuma.crt
```

Step 5: Set File Permissions

Secure your certificate files with proper permissions:

```
sudo chmod 600 /etc/nginx/ssl/uptime-kuma.key
sudo chmod 644 /etc/nginx/ssl/uptime-kuma.crt
```

Verify both files are in place:

```
ls -la /etc/nginx/ssl/
```

You should see two files:
• uptime-kuma.key (private key) - permissions: 600
• uptime-kuma.crt (your certificate) - permissions: 644

> **Note:** For testing purposes, we're not including the CA certificate bundle. This means OCSP stapling will be disabled. For production deployments, you may want to add the CA bundle.

Step 6: Install Nginx

Install Nginx web server:

```
sudo apt update
sudo apt install nginx -y
```

Verify installation:

```
nginx -v
```

Step 7: Create Nginx Configuration

Create the Nginx configuration file for Uptime Kuma:

```
sudo nano /etc/nginx/sites-available/uptime-kuma
```

Paste the following configuration (replace uptime.yourdomain.com with your actual FQDN):

```
# HTTP - Redirect to HTTPS
server {
    listen 80;
    listen [::]:80;
    server_name uptime.yourdomain.com;

    # Redirect all HTTP to HTTPS
    return 301 https://$server_name$request_uri;
}

# HTTPS - Main configuration
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name uptime.yourdomain.com;

    # SSL Certificate Configuration
    ssl_certificate /etc/nginx/ssl/uptime-kuma.crt;
    ssl_certificate_key /etc/nginx/ssl/uptime-kuma.key;
```

```
    # SSL Security Settings
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-S
    ssl_prefer_server_ciphers off;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    # Security Headers
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;

    # Logging
    access_log /var/log/nginx/uptime-kuma-access.log;
    error_log /var/log/nginx/uptime-kuma-error.log;

    # Proxy to Uptime Kuma Docker container
    location / {
        proxy_pass http://localhost:3001;
        proxy_http_version 1.1;

        # WebSocket support (required for Uptime Kuma)
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        # Standard proxy headers
        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;

        # Timeouts
        proxy_read_timeout 86400;
        proxy_connect_timeout 86400;
        proxy_send_timeout 86400;
    }
}
```

Save and exit the editor (Ctrl+X, Y, Enter).

> **Note:** This configuration does NOT include OCSP stapling (which requires the CA bundle). For production use, you may want to add ssl_trusted_certificate, ssl_stapling, and ssl_stapling_verify directives.

Step 8: Enable Site and Test Configuration

Enable the Uptime Kuma site:

```
# Create symbolic link
sudo ln -s /etc/nginx/sites-available/uptime-kuma /etc/nginx/sites-enabled/

# Remove default site (optional)
sudo rm /etc/nginx/sites-enabled/default
```

Test Nginx configuration:

```
sudo nginx -t
```

Expected output:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

If you see any errors, review your configuration file and certificates.

Step 9: Configure Firewall

Open the necessary ports in your firewall:

```
# Allow HTTP and HTTPS
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# Optionally block direct access to Uptime Kuma port
sudo ufw deny 3001/tcp

# Check firewall status
sudo ufw status
```

Note: If UFW is not enabled, you can skip this step or enable it with: `sudo ufw enable`

Warning: If connecting via SSH, allow SSH first: `sudo ufw allow 22/tcp`

Step 10: Start Nginx

Restart Nginx and enable automatic startup:

```
# Restart Nginx
sudo systemctl restart nginx

# Enable on boot
sudo systemctl enable nginx

# Check status
sudo systemctl status nginx
```

The status should show 'active (running)'.

Step 11: Verify Everything is Working

Test SSL connection locally:

```
openssl s_client -connect localhost:443 -servername uptime.yourdomain.com
```

Look for 'Verify return code: 0 (ok)' or certificate chain information.

Verify Nginx is listening on port 443:

```
sudo netstat -tlnp | grep nginx
```

You should see both ports 80 and 443.

Test from a web browser:

1. Open browser to: https://uptime.yourdomain.com
2. You should see the Uptime Kuma interface
3. Check for a padlock icon indicating a valid certificate
4. Click the padlock to view certificate details

Check Nginx logs if issues arise:

```
sudo tail -f /var/log/nginx/error.log
sudo tail -f /var/log/nginx/uptime-kuma-error.log
```

Step 12: Trust CA Certificate on Client Machines

For internal AD CA certificates, client browsers need to trust your CA root certificate.

Windows (Domain-Joined):

Usually automatic via Group Policy. No action required.

Windows (Manual):

1. Download root-ca.cer from your CA
2. Double-click the certificate file
3. Click "Install Certificate"
4. Select "Local Machine"
5. Choose "Place all certificates in the following store"
6. Browse to "Trusted Root Certification Authorities"
7. Complete the wizard

Linux/macOS:

Import the root CA certificate to your system's trust store using distribution-specific tools.

Step 13: Maintenance and Certificate Renewal

Certificates from your AD CA typically expire after 1-2 years. You'll need to renew them before expiration.

Renewal Process:

1. Generate new CSR (or reuse existing private key):
```
cd /etc/nginx/ssl
sudo openssl req -new -key uptime-kuma.key -out uptime-kuma-renewal.csr
```

2. Submit CSR to CA and download new certificate

3. Replace old certificate:
```
sudo cp uptime-kuma-new.crt uptime-kuma.crt
sudo chmod 644 /etc/nginx/ssl/uptime-kuma.crt
```

4. Reload Nginx (no downtime):
```
sudo systemctl reload nginx
```

5. Verify new certificate:
```
openssl s_client -connect localhost:443 -servername uptime.yourdomain.com
```

> **Tip:** Set a calendar reminder 30 days before certificate expiration to ensure timely renewal.

Architecture Overview

Understanding the complete traffic flow:

Component	Protocol	Port	Description
Internet/Users	HTTPS	443	External access point
Nginx	HTTPS→HTTP	443→3001	SSL termination and reverse proxy
Docker Container	HTTP	3001	Uptime Kuma application

Traffic Flow:

1. User browses to https://uptime.yourdomain.com
2. DNS resolves to your Ubuntu server
3. Nginx receives HTTPS request on port 443
4. Nginx performs SSL/TLS handshake using your AD CA certificate
5. Nginx decrypts the traffic (SSL termination)
6. Nginx forwards unencrypted HTTP request to localhost:3001
7. Docker container receives request on port 3001
8. Uptime Kuma processes and responds
9. Response travels back through Nginx (encrypted) to user

Security Best Practices

Certificate Security:
• Private key file (uptime-kuma.key) has 600 permissions (readable only by root)
• Never share or transmit your private key
• Store backups of private key securely and encrypted

Nginx Security:
• Only TLS 1.2 and 1.3 enabled (older protocols disabled)
• Strong cipher suites configured
• Security headers prevent clickjacking and XSS attacks
• HSTS header enforces HTTPS for future connections

Network Security:
• Port 3001 not exposed to external network (only localhost)
• Firewall rules limit access to ports 80 and 443 only
• All traffic between users and server is encrypted

Monitoring:
• Regularly review Nginx access and error logs
• Monitor certificate expiration dates
• Keep Ubuntu, Nginx, and Docker updated
• Set up alerts for service downtime

Quick Reference Commands

Task	Command
Check Nginx status	`sudo systemctl status nginx`
Restart Nginx	`sudo systemctl restart nginx`
Reload Nginx (no downtime)	`sudo systemctl reload nginx`
Test Nginx config	`sudo nginx -t`
View Nginx error log	`sudo tail -f /var/log/nginx/error.log`
Check Docker status	`docker ps`
Restart Uptime Kuma	`docker restart uptime-kuma`
View Docker logs	`docker logs uptime-kuma`
Test SSL certificate	`openssl s_client -connect localhost:443`
Check listening ports	`sudo netstat -tlnp`
Check certificate expiry	`openssl x509 -in /etc/nginx/ssl/uptime-kuma.crt -noout -dates`

Important File Locations

File/Directory	Purpose
`/etc/nginx/ssl/uptime-kuma.key`	Private key (600 permissions)
`/etc/nginx/ssl/uptime-kuma.crt`	SSL certificate (644 permissions)
`/etc/nginx/ssl/csr.conf`	CSR configuration with SAN
`/etc/nginx/sites-available/uptime-kuma`	Nginx configuration file
`/etc/nginx/sites-enabled/uptime-kuma`	Symlink to active config
`/var/log/nginx/uptime-kuma-access.log`	Nginx access logs
`/var/log/nginx/uptime-kuma-error.log`	Nginx error logs
`/var/log/nginx/error.log`	General Nginx error log
`uptime-kuma:/app/data`	Docker volume for Uptime Kuma data

Conclusion

You have successfully configured Uptime Kuma with HTTPS using Nginx as a reverse proxy and certificates from your internal Active Directory Certificate Authority. Your monitoring platform is now accessible securely at `https://uptime.yourdomain.com`.

Key Accomplishments:
• Secured communication with SSL/TLS encryption
• Integrated with your organization's PKI infrastructure
• Implemented industry-standard security headers
• Configured proper certificate validation
• Set up a production-ready reverse proxy

Remember to monitor certificate expiration dates and renew before they expire to maintain uninterrupted secure access to your Uptime Kuma instance.

> For questions, issues, or updates to this guide, please consult your organization's IT department or system administrator.