Uptime Kuma

Bulk Monitor Import Guide

Python Script for Automated HTTPS Monitor Creation

Complete Step-by-Step Guide
This guide provides detailed instructions for bulk importing HTTPS monitors into Uptime Kuma using a Python script and CSV file. The script automatically handles duplicate detection, sets heartbeat intervals, and enables certificate expiry notifications.

Document Version: 1.0
Last Updated: February 08, 2026

Table of Contents

1. Prerequisites

Before beginning this guide, ensure you have the following:

- Uptime Kuma already installed and running (see Uptime_Kuma_Installation_Guide.pdf)
- HTTPS enabled with Nginx reverse proxy (see Uptime_Kuma_HTTPS_Setup_Guide.pdf)
- SSH access to your Ubuntu server
- Uptime Kuma user account with username and password
- Python 3.8+ and pip installed on Ubuntu
- List of HTTPS URLs you want to monitor

> **Note:** This guide assumes you have already completed the Uptime Kuma installation and HTTPS configuration from the previous guides.

2. Initial Setup (One-Time)

These steps only need to be performed once to set up the bulk import environment.

Install Python 3 and pip:

```
sudo apt update
sudo apt install python3 python3-pip python3-venv -y
```

Verify installation:

```
python3 --version

pip3 --version
```

Create working directory:

```
mkdir ~/uptime-kuma-import
cd ~/uptime-kuma-import
```

Create Python virtual environment:

```
python3 -m venv kuma-venv
source kuma-venv/bin/activate
```

You should see `(kuma-venv)` appear at the beginning of your prompt.

Install uptime-kuma-api library:

```
pip install uptime-kuma-api
```

Create the import script:

nano import_monitors.py

Paste the complete script from Appendix B. Update the USERNAME and PASSWORD with your Uptime Kuma credentials.

Create the CSV file:

nano monitors.csv

See Appendix A for CSV file format and examples.

3. Regular Workflow (From Fresh SSH Login)

Use this workflow every time you want to add new monitors from a fresh SSH session.

1. SSH into your Ubuntu server:

```
ssh user@your-server
```

2. Navigate to the import directory:

```
cd ~/uptime-kuma-import
```

3. Activate the virtual environment:

> **CRITICAL:** You must activate the virtual environment in every new SSH session.

```
source kuma-venv/bin/activate
```

You will see `(kuma-venv)` appear at your prompt.

4. Edit the CSV file:

```
nano monitors.csv
```

Add new monitors to the CSV file. You can keep all existing monitors in the file - the script automatically handles duplicates and will only add new ones.

Example CSV content:

```
name,url
Google,https://www.google.com/
GitHub,https://github.com/
YourNewSite,https://yournewsite.com/
```

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

5. Run the import:

```
python import_monitors.py
```

The script will display which monitors were added or skipped:

```
Added monitor: YourNewSite
Skipping Google: URL already exists
Skipping GitHub: URL already exists
Import complete.
```

6. (Optional) Deactivate virtual environment:

```
deactivate
```

4. Important Notes

CSV File Management:

The monitors.csv file can contain ALL of your monitors, including ones that have already been imported. There is no need to delete monitors from the CSV when adding new ones. The script's duplicate detection logic will automatically skip any monitors that already exist in Uptime Kuma (based on URL).

Duplicate Detection:

- Duplicates are detected by URL, not by name
- If a URL already exists, it will be skipped regardless of the name
- You can safely run the import multiple times
- Only new URLs will be added to Uptime Kuma

Monitor Settings:

- **Heartbeat Interval:** 86400 seconds (24 hours)
- **Retry Interval:** 60 seconds
- **Max Retries:** 3 attempts
- **Certificate Expiry Notifications:** Enabled
- **TLS Validation:** Enabled (certificates must be valid)

> **Note:** To modify these settings, edit the api.add_monitor() parameters in the script.

5. Management Commands

Useful commands for managing the import environment:

Task	Command
Activate virtual environment	`source ~/uptime-kuma-import/kuma-venv/bin/activate`
Deactivate virtual environment	`deactivate`
Edit import script	`nano ~/uptime-kuma-import/import_monitors.py`
Edit CSV file	`nano ~/uptime-kuma-import/monitors.csv`
Run import	`python import_monitors.py`
View CSV file	`cat ~/uptime-kuma-import/monitors.csv`
Check Python version	`python3 --version`
Check installed packages	`pip list`

6. File Locations

File/Directory	Purpose
`~/uptime-kuma-import/`	`Working directory for import scripts`
`~/uptime-kuma-import/kuma-venv/`	`Python virtual environment`
`~/uptime-kuma-import/import_monitors.py`	`Import script`
`~/uptime-kuma-import/monitors.csv`	`Monitor list CSV file`

Conclusion

You have successfully set up bulk monitor import for Uptime Kuma. You can now efficiently manage large numbers of HTTPS monitors using a simple CSV file.

Key Benefits:
• Automated monitor creation from CSV files
• Duplicate detection prevents re-adding existing monitors
• Certificate expiry notifications enabled automatically
• Consistent configuration across all monitors
• Easy to maintain - keep all monitors in one CSV file

Remember to activate the virtual environment (`source kuma-venv/bin/activate`) before running the import script in each new SSH session.

> For questions or issues with the import process, verify that Uptime Kuma is running, check your credentials in the script, and ensure the CSV file format is correct.

Appendix A: CSV File Format

The CSV file must have two columns: name and url

Required Format:

```
name,url
MonitorName1,https://example1.com/
MonitorName2,https://example2.com/
MonitorName3,https://example3.com/
```

Column Requirements:

Column	Required	Description	Example
name	Yes	Display name in Uptime Kuma	Google
url	Yes	Must be HTTPS URL	https://www.google.com/

Important: Only HTTPS URLs are processed. HTTP URLs will be automatically skipped.

Appendix B: Import Script

Complete Python script for importing monitors. Save this as import_monitors.py in the ~/uptime-kuma-import directory.

```
"""
Script Name: import_monitors.py
Description: Bulk import HTTPS monitors into Uptime Kuma from a CSV file.
             Automatically handles duplicates by URL, sets heartbeat interval,
             and enables certificate expiry notifications.
Author: Auto-generated example
Requirements:
    - Python 3.8+
    - uptime-kuma-api Python library
    - monitors.csv file in the same directory
Usage:
    1. Create a virtual environment:
       python3 -m venv kuma-venv
       source kuma-venv/bin/activate
    2. Install required library:
       pip install uptime-kuma-api
    3. Edit configuration below (KUMA_URL, USERNAME, PASSWORD, CSV_FILE)
    4. Run:
       python import_monitors.py
Notes:
    - Duplicate detection is by URL.
    - Only HTTPS URLs are processed.
"""

import csv
from uptime_kuma_api import UptimeKumaApi

# ===== Configuration =====
KUMA_URL = "http://127.0.0.1:3001"  # URL to your Uptime Kuma instance
USERNAME = "adminuser"               # Uptime Kuma username
PASSWORD = "adminpassword"           # Uptime Kuma password
CSV_FILE = "monitors.csv"           # CSV file containing monitors
```

```
# ===== Connect to Uptime Kuma =====
api = UptimeKumaApi(KUMA_URL)
api.login(USERNAME, PASSWORD)

# ===== Fetch existing monitors to handle duplicates =====
existing_monitors = api.get_monitors()
existing_urls = {m["url"] for m in existing_monitors}

# ===== Read CSV and create monitors =====
with open(CSV_FILE, newline="", encoding="utf-8") as csvfile:
    reader = csv.DictReader(csvfile)

    for row in reader:
        name = row["name"].strip()
        url = row["url"].strip()

        # Only process HTTPS URLs for certificate expiry monitoring
        if not url.lower().startswith("https://"):
            print(f"Skipping {name}: URL is not HTTPS")
            continue

        # Skip duplicates
        if url in existing_urls:
            print(f"Skipping {name}: URL already exists")
            continue

        try:
            # ===== Create HTTPS monitor =====
            api.add_monitor(
                type="http",
                name=name,
                url=url,
                interval=86400,           # Heartbeat interval in seconds (24 hours)
                retryInterval=60,
                maxretries=3,
                expiryNotification=True,  # Enable certificate expiry alerts
                ignoreTls=False           # Validate certificates
            )

            print(f"Added monitor: {name}")

            # Add URL to set to prevent future duplicates in same run
            existing_urls.add(url)

        except Exception as e:
            print(f"Failed to add {name}: {e}")

# ===== Disconnect =====
api.disconnect()
print("Import complete.")
```

Configuration: Update KUMA_URL (if not using localhost), USERNAME, and PASSWORD with your actual Uptime Kuma credentials.