16: Uptime Kuma - Monitoring & Status Pages

Uptime Kuma is a self-hosted monitoring solution built with Node.js. It provides uptime monitoring, incident tracking, status pages, and 90+ notification integrations. Supports HTTP(s), TCP, Ping, DNS, Docker containers, and more. For API documentation, advanced configuration, and development guides, see the official Uptime Kuma repository.

Prerequisites

- **Docker installed** (Chapter 3)
- **Docker Compose** (Chapter 3)
- **Optional: Traefik installed** (Chapter 4) for HTTPS with Let's Encrypt
- **Optional: Domain configured** (Chapter 4.5), e.g., `status.example.com`

Installation via Infinity Tools

Menu Installation

```
APPLICATIONS → Uptime Kuma → Install
```

CLI Installation

```
sudo bash /opt/InfinityTools/Solutions/setup-uptime-kuma.sh --install


# With domain (Traefik mode)
export BS_DOMAIN="status.example.com"
sudo -E bash /opt/InfinityTools/Solutions/setup-uptime-kuma.sh --install
```

Deployment Modes

Traefik Mode (Default)

Uses Traefik for SSL termination and domain routing:

- Automatic Let's Encrypt certificate provisioning
- Domain-based access: `https://status.example.com`
- Security headers configured
- Requires: Traefik running, DNS A record configured

Standalone Mode

Direct access with HTTP or HTTPS (self-signed):

- HTTP: `http://SERVER_IP:3001`
- HTTPS: `https://SERVER_IP:3001` (self-signed cert via nginx proxy)
- Default port: 3001 (configurable)
- No domain required

Architecture

Container

- **uptime-kuma** - Main application (louislam/uptime-kuma:1)
- **uptime-kuma-ssl-proxy** - Nginx SSL proxy (standalone HTTPS mode only)

Data Persistence

- **Data:** `/opt/speedbits/uptime-kuma/data/` (SQLite database, config)
- **SSL:** `/opt/speedbits/uptime-kuma/ssl/` (standalone mode certificates)

Networks

- **Traefik network:** Joins Traefik's proxy network (Traefik mode)
- **kuma-internal:** Isolated bridge network (standalone mode)

Docker Socket Access

Optional read-only access to `/var/run/docker.sock` for Docker container monitoring:

- Enables Docker container monitoring
- Read-only mount (security best practice)
- Configured during installation

Installation Process

Configuration Steps

1. **SSL Mode Selection:** Choose Traefik (default) or Standalone
2. **If Traefik:** Provide domain name
3. **If Standalone:** Specify port (default: 3001) and SSL mode
4. **Docker Monitoring:** Optional enable Docker socket access
5. **Timezone:** Optional timezone configuration (default: UTC)

What Gets Created

- **Directory:** `/opt/speedbits/uptime-kuma`
- **Container:** `uptime-kuma`
- **Docker Compose:** `/opt/speedbits/uptime-kuma/docker-compose.yml`
- **Data Volume:** SQLite database and configuration

Access Methods

Traefik Mode

```
https://status.example.com
```

Direct web access after DNS propagation and SSL certificate generation (30-60 seconds).

Standalone Mode

HTTP:

```
http://SERVER_IP:3001
```

HTTPS:

```
https://SERVER_IP:3001
```

Accept self-signed certificate warning (Advanced → Proceed).

Authentication

First-Time Setup

- **No default credentials** - Admin account must be created on first access
- **Setup wizard:** "Create your admin account" appears on first visit
- **Password requirements:** Minimum 8 characters (12+ recommended)
- ⚠ **CRITICAL:** Write down credentials immediately - no password reset on first setup

Password Reset

```
docker exec -it uptime-kuma npm run reset-password
```

Follow prompts to enter username and new password.

Monitor Types

Supported Protocols

- **HTTP(s)** - Websites, APIs, webhooks
- **TCP** - Port monitoring (SSH, databases, etc.)
- **Ping (ICMP)** - Server availability
- **DNS** - DNS record monitoring
- **Docker Container** - Container health (requires Docker socket)
- **Keyword** - Content-based monitoring
- **SMTP** - Email server monitoring
- **gRPC** - gRPC service monitoring

Monitor Configuration

- **Check interval** - Frequency of checks (default: 60 seconds)
- **Retry attempts** - Number of retries before marking as down
- **Timeout** - Request timeout
- **Expected status codes** - HTTP status codes to consider "up"
- **Keyword detection** - Check for specific text in response

Notification Integrations

Supported Providers

- **Discord** - Webhook integration
- **Slack** - Webhook integration
- **Telegram** - Bot API
- **Email** - SMTP
- **Apprise** - Self-hosted Apprise (80+ services)
- **Webhooks** - Custom HTTP endpoints
- **90+ providers** - See full list in Uptime Kuma settings

Apprise Integration

If Apprise is installed (Chapter 5), use it for notifications:

- Type: **Apprise (Self-hosted)**
- URL: `http://apprise:8000/notify/{YOUR-KEY}`
- Enables access to all Apprise-supported services

Status Pages

Features

- Public status pages (no authentication required)
- Customizable appearance (colors, logo, theme)
- Monitor selection (choose which monitors to display)
- Incident history
- Uptime statistics
- RSS feed support

Use Cases

- Public service status (like status.github.com)
- Internal team dashboards
- Customer-facing status pages
- Service health transparency

Environment Variables

Uptime Kuma Container

- `TZ` - Timezone (default: UTC)

Data Storage

- SQLite database stored in `/app/data`
- Persisted via Docker volume mount
- Includes monitors, notifications, status pages, user accounts

Security Configuration

Access Security

- Traefik mode uses Let's Encrypt SSL (production-ready)
- Standalone HTTPS uses self-signed certificates (acceptable for internal use)
- Security headers configured (X-Frame-Options, CSP, etc.)
- Docker socket mounted read-only (if enabled)
- Security option: `no-new-privileges:true`

Container Security

- Runs as non-root user
- Read-only Docker socket access (if enabled)
- Network isolation
- Volume mounts for data persistence

Configuration Persistence

- **Data Volume:** `data` persists all configuration
- **SQLite Database:** Stored in data directory
- All monitors, notifications, and settings survive container restarts

Backup & Restore

Backup Strategy

```
# Full backup
cd /opt/speedbits
tar czf uptime-kuma-backup-$(date +%Y%m%d).tar.gz uptime-kuma/

# Using Uptime Kuma built-in backup
# Settings → Backup → Download Backup
```

Restore Process

1. Stop container: `cd /opt/speedbits/uptime-kuma && docker compose down`
2. Restore data: Extract backup to `/opt/speedbits/uptime-kuma/`
3. Start container: `docker compose up -d`

Troubleshooting

Container Not Starting

```
docker logs uptime-kuma
docker ps -a | grep uptime-kuma
```

SSL Certificate Issues

- **Traefik mode:** Check Traefik logs: `docker logs traefik`
- **Traefik mode:** Verify DNS: `dig status.example.com`
- **Standalone mode:** Check nginx proxy logs: `docker logs uptime-kuma-ssl-proxy`

Docker Monitoring Issues

- Verify Docker socket access: `docker exec uptime-kuma ls /var/run/docker.sock`
- Check container permissions
- Verify socket is mounted read-only

Monitor Not Responding

- Check monitor configuration (URL, port, etc.)

- Verify service is actually running
- Check network connectivity from container
- Review monitor logs in Uptime Kuma dashboard

Production Considerations

- **Access Method:** Use Traefik mode for production (trusted SSL)
- **Password Policy:** Enforce strong passwords (12+ characters)
- **2FA:** Enable two-factor authentication (Settings → Security)
- **Backup Strategy:** Implement automated backups
- **Monitoring:** Monitor Uptime Kuma itself (meta-monitoring)
- **Notifications:** Configure multiple notification channels for redundancy
- **Status Pages:** Use public status pages instead of sharing admin access

Integration with Infinity Tools

Uptime Kuma complements Infinity Tools by providing:

- Monitoring for all Infinity Tools applications
- Docker container monitoring for infrastructure
- Status pages for public service transparency
- Notification integration with Apprise

Recommended Monitors:

- Traefik dashboard
- All Infinity Tools application endpoints
- Docker daemon health
- Server resources (via Netdata if installed)

API & Automation

REST API

- Uptime Kuma provides REST API for programmatic access
- API documentation available in web interface
- Useful for automation and integration

Webhooks

- Incoming webhooks for external integrations
- Outgoing webhooks for custom notifications
- Custom payload formatting

Next Steps

Uptime Kuma is now operational. Use it to:

- Monitor all Infinity Tools applications
- Track Docker container health
- Create public status pages
- Set up multi-channel notifications
- Track uptime statistics and incidents

For advanced features, API usage, custom themes, and development guides, refer to the official Uptime Kuma repository.

Revision #1
Created 17 November 2025 16:48:57 by bjoern
Updated 17 November 2025 16:49:41 by bjoern