README

Author: Huayun Li

#System Resource Monitor

A Python-based system monitoring tool for real-time resource tracking and alerts.

Table of Contents

1. Overview

1.1 Core Features

- Real-time monitoring of CPU, memory, and disk usage
- Configurable alert thresholds and intervals
- Email notifications for threshold violations
- Rich console output with color coding
- Rotating log file support

1.2 System Requirements

- Python 3.9 or higher
- Modern Linux/Unix system or Windows
- Network connection for email alerts

2. Installation

2.1 Quick Start

Bash:

```
$ git clone https://github.com/Thomaslica/632system_monitor.git
$ cd system-monitor
$ pip install -r requirements.txt
$ cp config.yaml.example config.yaml
```

2.2 Dependencies

- psutil>=5.9.0: System monitoring
- PyYAML>=6.0.1: Configuration handling
- rich>=13.3.1: Console formatting

3. Configuration

3.1 Basic Settings (config.yaml)

thresholds:

```
  cpu: 80          # CPU threshold (%)
  memory: 80       # Memory threshold (%)
  disk: 80         # Disk threshold (%)

interval: 300              # Check interval (seconds)
alert_cooldown: 3600       # Alert cooldown (seconds)
```

3.2 Email Configuration

```
email:
  smtp_server: "smtp.gmail.com"
  smtp_port: 587
  sender: "your-email@gmail.com"
  password: "your-app-password"
  recipient: "admin@example.com"
```

3.3 Output Settings

```
output:
  log_level: "INFO"
  console_colors: true
  log_file: true
  log_max_size: 10
  log_backups: 5
```

4. Command Line Usage

4.1 Basic Commands

- Start monitoring:

Bash: $python system_monitor.py

- Custom interval:

Bash: $python system_monitor.py --interval 60

- Enable file logging:

Bash: $python system_monitor.py --log-file

4.2 Command Arguments

- --config PATH: Custom config file
- --interval SECONDS: Check interval
- --log-file: Enable file logging
- --output FILE: Output redirection
- --debug: Debug logging
- --quiet: Minimal output

4.3 Environment Variables

- MONITOR_CONFIG: Config file path
- MONITOR_LOG_LEVEL: Logging level
- MONITOR_ALERT_EMAIL: Alert email

4.4 Exit Codes

- 0: Success
- 1: Configuration error
- 2: Permission error
- 3: Runtime error

5. Output Formats

5.1 Console Output

System Resource Usage Report - 2024-12-26 14:30:45

==========

CPU Usage : 45.2% (Threshold: 80%) - OK

Memory Usage : 72.8% (Threshold: 80%) - OK

Disk Usage : 85.1% (Threshold: 80%) - ALERT

==========

5.2 Log File Format

2024-12-26 14:30:45 - INFO - CPU: 45.2%, Memory: 72.8%, Disk: 85.1%

2024-12-26 14:30:45 - WARNING - Disk usage exceeds threshold

6. Components

6.1 Core Classes

- SystemMonitor: Main monitoring class
    - check_resources(): Resource checks
    - send_alert(): Email notifications
    - get_resource_usage(): Metric collection
- ResourceUsage: Metric data class
    - cpu_percent: CPU usage
    - memory_percent: Memory usage
    - disk_percent: Disk usage
    - timestamp: Measurement time

7. Error Handling

7.1 Common Issues

- Configuration errors
- Permission issues
- Resource access failures
- Network connectivity problems

7.2 Recovery Mechanisms

- Default value fallback
- Automatic retry logic
- Graceful degradation
- Error logging

8. Best Practices

8.1 Production Setup

- Run as system service
- Configure log rotation
- Regular config review
- Monitor redundancy

8.2 Security

- Protect config files
- Use secure SMTP
- Regular password updates
- Minimal permissions

9. Troubleshooting

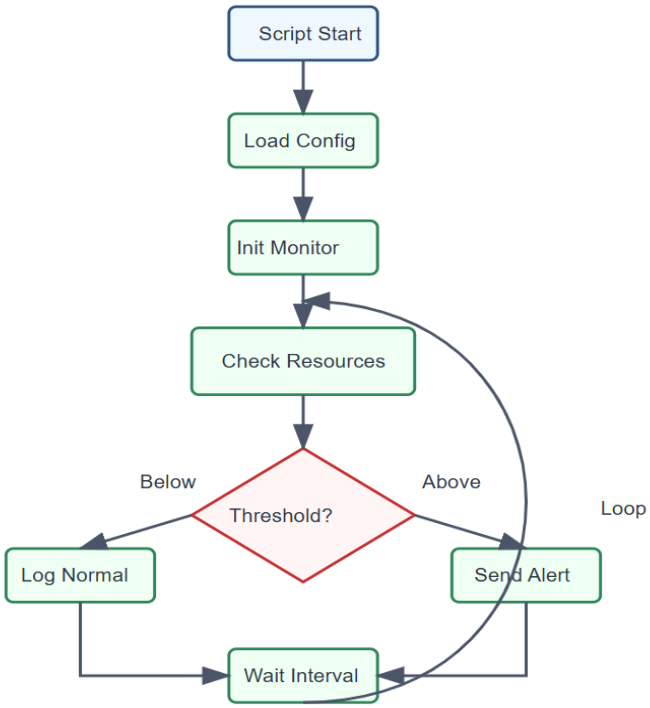
9.1 Startup Issues

- Check Python version
- Verify dependencies
- Test file permissions
- Validate config syntax

9.2 Runtime Problems

- Check log files
- Verify SMTP settings
- Monitor resource usage
- Test network connectivity

10. Workflow



License

MIT License - Free to use and modify