

服务器实时监控系统用户使用手册

一、系统简介

1.1 核心功能

- 多服务器集中管理：支持同时监控多台 Linux 服务器，统一展示 CPU、内存、磁盘、网络等核心指标
- 实时可视化大屏：16:9 适配投屏，0.5 秒自动刷新数据，趋势图、仪表盘直观呈现运行状态
- 智能告警提醒：资源使用率超阈值时，实时弹出分级告警（黄色警告 / 红色严重），记录完整告警日志
- 零侵入一键部署：无需手动配置被监控节点，通过 SSH 自动部署轻量级 Agent，全程无感知
- 历史数据追溯：支持 1 小时 - 90 天数据查询，支持图表缩放、数据导出，满足运维分析需求
- 高可用采集：Agent 模式故障时自动切换至 Ansible 降级模式，确保监控不中断

1.2 适用场景

- 中小型企业机房服务器集群监控
- 云服务器（阿里云、腾讯云等）远程运维
- 监控中心大屏投屏展示
- 开发 / 测试环境服务器资源监控
- 无专业运维人员的团队快速部署监控方案

二、前置要求

要求类型	具体说明
运行环境	监控中心：Ubuntu 22.04 LTS（推荐）；被监控节点：Ubuntu 20.04+/CentOS 7+
必备工具	Docker 20.10+、Docker Compose 2.0+（系统会自动安装，无需手动配置）
网络要求	

	监控中心与被监控节点网络互通，开放端口： 22（SSH）、3000（前端）、5000（后端）、 9100（Agent）
硬件配置	监控中心：2核4G内存50G磁盘；被监控节点： 1核1G内存20G磁盘（最低配置）
权限要求	监控中心需具备 sudo 权限；被监控节点需提供具有 SSH 登录权限的管理员账号（推荐 root ）

三、一键部署步骤

3.1 环境准备（首次部署）

1. 登录监控中心服务器（Ubuntu 系统），打开终端，执行以下命令安装依赖：

```
# 更新系统软件包（确保依赖安装最新版）
sudo apt-get update && sudo apt-get upgrade -y
# 安装Python、Git、Ansible等核心依赖
sudo apt-get install -y python3.9 python3-pip git ansible sshpass
# 安装Docker与Docker Compose（容器化部署核心工具）
sudo apt-get install -y docker.io docker-compose-plugin
sudo systemctl start docker && sudo systemctl enable docker # 启动并设置开机自启
```

2. 配置 Docker 加速器（可选，提升镜像拉取速度，建议配置）：

```
# 自动创建Docker配置文件并添加国内镜像源
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": [
    "https://docker.mirrors.ustc.edu.cn",
    "https://hub-mirror.c.163.com",
    "https://reg-mirror.qiniu.com"
  ]
}
EOF
# 重启Docker使配置生效
```

```
sudo systemctl daemon-reload && sudo systemctl restart docker  
# 验证加速器配置（可选，查看输出是否包含配置的镜像源）  
docker info | grep -A 3 "Registry Mirrors"
```

[截图 1：Docker 加速器配置验证截图（需显示 3 个镜像源地址）]

3.2 项目部署（核心步骤）

1. 克隆项目代码（替换为实际项目仓库地址）：

```
git clone https://github.com/xxx/server-monitor-system.git  
cd server-monitor-system # 进入项目根目录
```

2. 一键构建并启动服务：

```
# 后台启动服务（首次构建需下载镜像和依赖，耗时5-10分钟，耐心等待）  
docker compose up -d --build  
# 查看服务启动状态（验证是否成功）  
docker compose ps
```

验证服务启动成功：

```
# 查看后端服务日志（确认采集任务启动）  
docker compose logs -f backend
```

3.3 部署失败排查（可选）

若 docker compose ps 显示服务状态为 Exit，执行以下命令查看错误日志：

```
# 查看后端错误日志（核心排查依据）  
docker compose logs backend | grep -i "error"  
# 查看前端错误日志（可选）  
docker compose logs frontend | grep -i "error"
```

根据日志提示修复问题后，重新执行部署命令：

```
docker compose down # 停止并清理旧服务  
docker compose up -d --build # 重新构建启动
```

四、系统使用指南

4.1 访问系统

1. 打开任意浏览器（推荐 Chrome、Edge），输入监控中心服务器 IP+3000 端口：

http://监控中心IP:3000

示例：http://192.168.211.149:3000

2. 成功访问后进入系统主页，无需登录即可使用所有功能。

4.2 添加被监控主机（核心操作）

1. 在系统主页点击顶部菜单【主机管理】，进入主机列表页面；
2. 点击页面右上角【添加主机】按钮，弹出配置窗口；
3. 按要求填写被监控主机信息（带 * 为必填项）：

字段名称	示例值	填写说明
IP 地址 *	192.168.211.150	被监控主机的内网 / 公网 IP，确保监控中心能通过该 IP 访问 22 端口（SSH）
SSH 端口 *	22	默认为 22，若被监控主机修改过 SSH 端口（如 2222），需对应填写
登录用户名 *	root	具有管理员权限的 SSH 登录账号（需能执行 sudo 命令，用于部署 Agent）
登录密码 *	12345678	SSH 登录密码（系统采用 Fernet 加密存储，安全无泄露风险）
主机备注	生产环境 - Web 服务器	可选，用于区分主机用途，便于管理

告警阈值配置	默认 (CPU 80% / 内存 85% / 磁盘 90%)	可自定义，超过阈值将触发告警，建议根据主机用途调整
--------	--------------------------------	---------------------------

4. 点击【确认添加】，系统自动执行 3 步操作：

- 测试 SSH 连接：验证账号密码是否正确、网络是否通畅；
- 部署 Mini Agent：通过 SSH 在被监控主机部署轻量级采集服务；
- 启动采集任务：添加完成后 3-5 秒内开始采集指标数据；

5. 刷新主机列表页面，若主机状态显示【在线】，且实时指标正常显示，说明添加成功。

4.3 核心功能使用

4.3.1 实时监控大屏

1. 点击顶部菜单【监控大屏】，进入可视化展示页面；

2. 页面功能说明：

- 顶部概览区：显示在线主机总数、平均 CPU / 内存 / 磁盘使用率、当前活跃告警数；
- 主机选择区：左侧列出所有在线主机，点击可切换查看目标主机的详细指标；
- 指标图表区：中间展示 1 小时内 CPU、内存、磁盘使用率趋势曲线，支持鼠标悬停查看具体数值；
- 告警提示区：右上角实时弹出告警通知，点击可查看告警详情；

3. 操作技巧：

- 全屏模式：按 F11 切换全屏，适合监控中心投屏；
- 刷新控制：页面默认 0.5 秒自动刷新，可点击页面底部【暂停刷新】按钮暂停，便于数据分析；
- 指标切换：点击图表下方的指标名称（CPU / 内存 / 磁盘），可单独显示某类指标趋势。

4.3.2 历史数据查询

1. 点击顶部菜单【历史数据】，进入查询页面；

2. 筛选条件配置：

- 选择主机：从下拉框选择需要查询的被监控主机；
- 选择时间范围：支持预设（1 小时 / 6 小时 / 24 小时 / 7 天 / 30 天）或自定义时间区间（最多支持 90 天）；

3. 点击【查询】，系统展示该时间段内的指标趋势图，支持以下操作：

- 图表缩放：鼠标滚轮放大 / 缩小，拖动图表可查看指定区间；
- 数据导出：点击页面右上角【导出 Excel】，可将历史数据导出为表格文件，用于离线分析；
- 指标切换：单独勾选 CPU / 内存 / 磁盘 / 网络指标，可自定义展示内容。

4.3.3 告警记录管理

1. 点击顶部菜单【告警记录】，进入告警列表页面；

2. 功能说明：

- 列表展示：显示所有告警记录（包括未解决和已解决），包含主机 IP、告警类型、阈值、当前值、告警时间；
- 状态筛选：可通过页面顶部【未解决告警】【已解决告警】标签筛选查看；
- 标记已解决：告警恢复后（指标低于阈值），可点击操作列【标记已解决】，更新告警状态；
- 日志导出：点击【导出告警日志】，可将所有记录导出为 Excel 文件，用于运维审计。

4.3.4 主机管理操作

1. 编辑主机信息：在主机列表页面，点击目标主机操作列【编辑】，可修改 SSH 端口、登录密码、告警阈值等信息；
2. 删除主机：点击【删除】，系统将停止该主机的采集任务，并删除所有相关指标和告警记录（不可恢复，请谨慎操作）；
3. 手动刷新状态：若主机状态显示异常，可点击页面顶部【刷新所有主机状态】按钮，手动触发状态检测；
4. 批量操作：勾选多个主机，可执行批量删除、批量刷新状态操作（适合多主机管理场景）。

五、常见问题排查

问题现象	可能原因	解决方案
浏览器无法访问系统（提示“无法连接”）	1. 监控中心服务器防火墙未开放 3000/5000 端口；2. frontend/backend 服务未启动	1. 开放端口： <code>sudo ufw allow 3000 && sudo ufw allow 5000-</code> ；2. 查看服务状态： <code>docker</code>

		<code>compose ps</code> , 若未启动执行 <code>: docker compose restart</code>
添加主机后状态一直显示“初始化中”	1. SSH 连接失败（账号密码错误 / 端口错误）；2. 被监控主机 22 端口未开放；3. 被监控主机无管理员权限	1. 验证 SSH 连接：ssh 用户名@主机IP -p 端口，确保能正常登录；2. 开放 22 端口： <code>- sudo ufw allow 22</code> ；3. 确认账号具有 sudo 权限
主机状态显示“在线”但指标不刷新	1. Agent 服务未启动；2. 被监控主机 9100 端口未开放	1. 登录被监控主机，重启 Agent： <code>sudo systemctl restart network-agent</code> ；2. 开放 9100 端口： <code>sudo ufw allow 9100</code>
监控大屏无告警提示（指标已超阈值）	1. 告警阈值设置过高；2. 告警功能未启用	1. 编辑主机，降低告警阈值；2. 确认系统未关闭告警：页面底部【告警开关】处于开启状态
Docker 加速器配置后不生效	1. JSON 格式错误；2. 未重启 Docker 服务	1. 校验 JSON 格式：访问 https://jsonlint.com/ ，粘贴/etc/docker/daemon.json 内容校验；2. 重启 Docker： <code>sudo systemctl daemon-reload && sudo systemctl restart docker</code>
部署时提示“permission denied”（权限不足）	未使用 sudo 命令，或当前用户无 Docker 操作权限	1. 所有部署命令前添加 sudo；2. 赋予当前用户 Docker 权限： <code>sudo usermod -aG docker \$USER</code> ，注销后重新登录

六、注意事项

1. 数据安全：系统加密存储 SSH 密码，建议定期更换被监控主机的 SSH 密码，进一步提升安全性；
2. 资源占用：监控中心每监控 10 台主机，约占用 0.5G 内存和 5% CPU，若需监控更多主机（超过 20 台），请升级监控中心硬件配置（推荐 4 核 8G 内存）；
3. 数据保留：监控数据默认保留 90 天，90 天后自动清理过期数据，若需长期保留，可修改项目配置文件中的DATA_RETENTION_DAYS 参数（需重启服务生效）；

4. 数据备份：数据库文件存储在项目目录下的`./data`文件夹，建议定期备份该文件夹（如每周备份 1 次），避免数据丢失；
5. 系统更新：若需更新系统版本，可通过`git pull`拉取最新代码后，重新执行`docker compose up -d --build`，无需重新配置主机信息（数据持久化存储）；
6. 被监控主机迁移：若被监控主机 IP 变更，需在【主机管理】中编辑该主机的 IP 地址，否则将无法正常采集数据。

| (注：文档部分内容可能由 AI 生成)