

# monitorViewer user manual

**Product Name** : monitorViewer

**Product Version** : V1.1

**Release Date** : 2024.05.31

**Contact** : @李艳青 (liyanqing.1987@bytedance.com)

@葛贺 (gehe@bytedance.com)

## 目录

一、简介 .....	3
二、环境依赖 .....	5
2.1 操作系统依赖 .....	5
2.2 PYTHON 版本依赖 .....	5
三、工具安装及配置 .....	6
3.1 安装 .....	6
3.2 配置 .....	6
3.3 启动 WEB 服务 .....	8
四、工具使用 .....	9
4.1 开发脚本 .....	9
4.2 脚本启动 .....	11
4.3 日志数据 .....	11
4.4 监控查询 - GUI .....	13
4.5 监控查询 - WEB .....	14
4.6 心跳监测机制 .....	16
附录 .....	17
附 1. 变更历史 .....	17

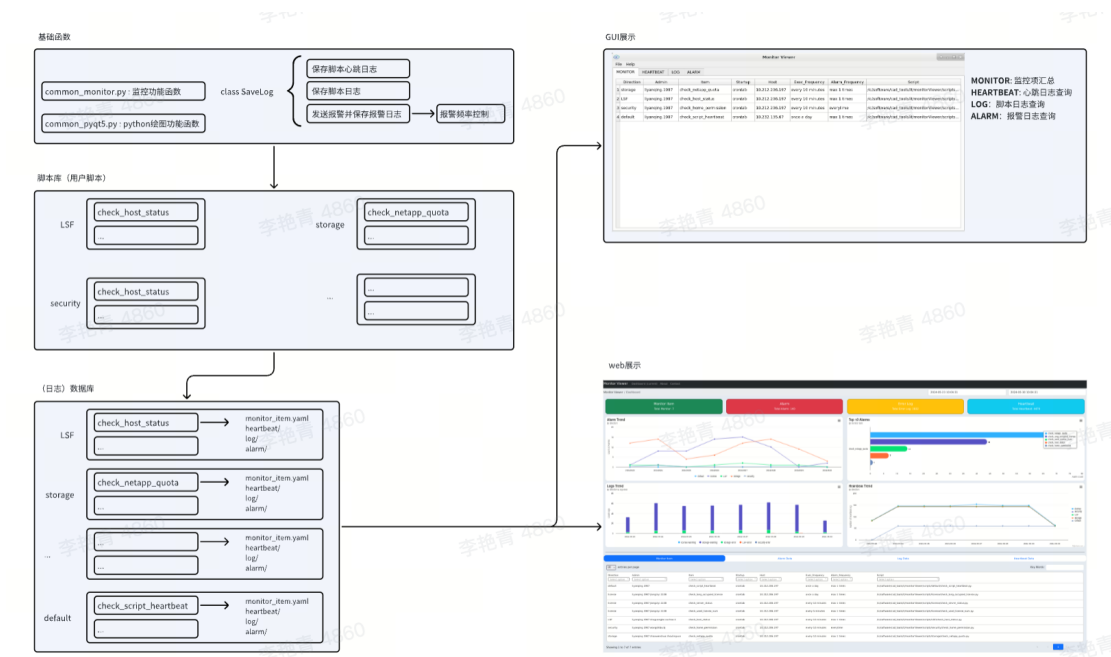
# 一、简介

IT 环境为确保可用性和稳定性，通常需要增加各种系统或者服务器的监控，Zabbix 等开源监控工具本身可以监控服务器、网络、服务等项，但是业务部门往往仍然有大量的定制化监控需求，依赖自研监控脚本实现。

自研监控脚本管理和维护往往面临如下问题：

- 监控方向和监控项众多，监控脚本开发者也不相同，没有统一的开发架构。
- 缺乏一个顶层的前端界面，可以查阅所有的定制监控项及其历史记录。
- 一般的监控脚本缺乏统一的日志系统，无法事后查询。
- 报警方式不统一，频率不易控制，且无记录。
- 监控脚本本身的运行没有监督机制，无法及时发现监控程序的未启动故障。

为了解决如上问题，我们开发了 monitorViewer 的定制监控程序框架，并提供了前端查询界面用于监控信息的汇总查询，其基本架构如下。



此处先说明几个概念：

- direction**：业务方向，指 LSF/storage/security/License 等不同管理员维护的业务。
- monitor\_item**：监控项，比如检查 LSF 中机器的状态，就是 LSF 的定制化监控项之一。
- heartbeat**：心跳，指的是程序每次启动的时候都保留一份记录，证明程序启动过。

**log**：程序内部需要保留的信息，可以有不同的信息登记。

**alarm**：报警，针对程序内部发现的严重异常，需要报警并保存报警记录。

## 二、环境依赖

### 2.1 操作系统依赖

monitorViewer 的开发和测试操作系统为 **CentOS Linux release 7.9.2009 (Core)**, 这也是 IC 设计常用的操作系统版本之一。

centos6/centos7/centos8, 及对应的 redhat 版本应该都可以运行, 主要的潜在风险在于系统库版本差异可能会影响部分组件的运行。

建议在 centos7.9 操作系统下使用。

### 2.2 python 版本依赖

monitorViewer 基于 python 开发, 其开发和测试的 python 版本为 **python3.8.8**, 推荐使用 **Anaconda3-2021.05** 以解决库依赖问题。

不同版本的 python 可能会有 python 库版本问题, 按照系统要求安装对应版本的 python 库即可解决。

## 三、工具安装及配置

### 3.1 安装

获取工具安装包，文件目录如下所示。

```
[liyanqing.1987@n232-135-013 monitorViewer]$ ls
bin common config data db docs install.py lib README.md
requirements.txt scripts web
```

确保满足环境依赖后，通过 `python install.py` 的方式安装。

```
[liyanqing.1987@n232-135-013 monitorViewer]$ python3 install.py
>>> Check python version.
      Required python version : (3, 8)
      Current python version : (3, 8)

>>> Generate script
"/ic/data/usr/liyanqing.1987/test/monitorViewer/bin/monitor_viewer".
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/monitorViewer/scripts/gen_monitor_script".
>>> Generate script
"/ic/data/usr/liyanqing.1987/test/monitorViewer/scripts/default/check_script_heartbeat".
>>> Generate config file
"/ic/data/usr/liyanqing.1987/test/monitorViewer/config/config.py".
>>> Generate run web script "web/run.sh".

Done, Please enjoy it.
```

### 3.2 配置

`config/config.py` 是框架配置文件，内容如下。

```
# Specify valid direction and direction_admin.
valid_direction_dic = {
    "default": "liyanqing.1987",
}
```

```
# Specify database path.
db_path = "/ic/data/usr/liyanqing.1987/tools/monitorViewer/db"

# Specify valid message level list, which is used on
SaveLog.save_log argument.
valid_message_level_list = ['Debug', 'Info', 'Warning', 'Error',
'Fatal']

# Specify how to execute alarm command.
send_alarm_command = ""
```

**valid\_direction\_dic**：合法的 direction（业务方向）及其主要负责人。

**db\_path**：数据库（日志库）路径。

**valid\_message\_level\_list**：合法的信息等级，一般不需要修改。

**send\_alarm\_command**：发送报警的命令机器参数（参数值用抽象模式取代）。

下面是修改后应用在业务环境中的一个配置。

```
# Specify valid direction and direction_admin.
valid_direction_dic = {
    "default": "liyanqing.1987",
    "license": "liyanqing.1987",
    "LSF": "liyanqing.1987",
    "storage": "liyanqing.1987 USER_B USER_C",
    "security": "liyanqing.1987 USER_D"
}

# Specify database path.
db_path = "/ic/data/CAD/it/monitorViewer"

# Specify valid message level list, which is used on
SaveLog.save_log argument.
valid_message_level_list = ['Debug', 'Info', 'Warning', 'Error',
'Fatal']

# Specify how to execute alarm command.
send_alarm_command = '/ic/software/cad_tools/bin/send_lark -T
"<TITLE>" -c "<MESSAGE>" -r <RECEIVERS>'
```

其中 `send_alarm_command` 支持不同的报警工具，支持 3 个参数：

**<TITLE>**：报警信息标题。

**<MESSAGE>**：报警信息内容，为字符串，可以包含换行符“\n”。

**<RECEIVERS>**：报警接收人。

### 3.3 启动 web 服务

`web/run.sh` 用于为 `monitorViewer` 启动 web 服务，web 服务可以在物理机/虚拟机上直接启动，基于 `docker/k8s` 是更加推荐的方式。

```
[liyanqing.1987@n232-135-013 monitorViewer]$ cd web
[liyanqing.1987@n232-135-013 web]$ ls
app.py  run.sh  service  static  templates  tools
[liyanqing.1987@n232-135-013 web]$ nohup ./run.sh &
[2] 109186
nohup: ignoring input and appending output to 'nohup.out'
```

从 `nohup.out` 中可以获取 web 服务器的访问地址。

```
[liyanqing.1987@n232-135-013 web]$ cat nohup.out
* Debug mode: off
2024-06-03 14:56:24,088 | INFO | WARNING: This is a development
server. Do not use it in a production deployment. Use a production
WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.232.135.13:5000
2024-06-03 14:56:24,088 | INFO | Press CTRL+C to quit
```

在 Linux 系统中，通过 `firefox` 访问 `http://10.232.135.13:5000`，可以获取 `monitorViewer` 相关的信息。



## 四、工具使用

### 4.1 开发脚本

monitorView 默认脚本库位于安装目录下的 scripts/下。

```
[liyanqing.1987@n232-135-013 monitorViewer]$ ls scripts/  
default  gen_monitor_script  gen_monitor_script.py
```

当然，用户也可以把脚本放到任意自己喜欢的路径，但是极力推荐采用 monitorViewer 自带的 gen\_monitor\_script 程序生成脚本，已完成 SaveLog 类的自动初始化。

```
[liyanqing.1987@n232-135-013 monitorViewer]$ mkdir scripts/LSF  
[liyanqing.1987@n232-135-013 monitorViewer]$ cd scripts/LSF/  
[liyanqing.1987@n232-135-013 LSF]$ ../gen_monitor_script  
check_host_status.py
```

这样生成的 python 程序已经完成了 saveLog 的初始化，check\_host\_status.py 脚本中 SaveLog 初始化的核心代码如下。

```
...  
os.environ['MONITOR_VIEWER_INSTALL_PATH'] =  
'/ic/data/usr/liyanqing.1987/tools/monitorViewer'  
  
sys.path.append(os.environ['MONITOR_VIEWER_INSTALL_PATH'])  
from common import common_monitor  
  
save_log_ins = common_monitor.SaveLog(  
    direction='',  
    monitor_item='',  
    script_path=os.path.abspath(__file__),  
    script_auther='liyanqing.1987',  
    script_startup_method='',  
    script_execute_frequency='',  
    alarm_receivers='',  
    alarm_frequency='everytime')  
...
```

这个初始化过程只填充了一些默认值，我们还需要根据实际情况补充/修改一下，修改后的内容如下。

```
...
os.environ['LSFMONITOR_INSTALL_PATH'] =
    '/ic/software/cad_tools/it/lsfMonitor/monitor'
os.environ['MONITOR_VIEWER_INSTALL_PATH'] =
    '/ic/software/cad_tools/it/monitorViewer'

sys.path.append(os.environ['LSFMONITOR_INSTALL_PATH'])
from common import common_lsf
sys.path.append(str(os.environ['MONITOR_VIEWER_INSTALL_PATH']) +
    '/common')
import common_monitor

save_log_ins = common_monitor.SaveLog(
    direction='LSF',
    monitor_item='check_host_status',
    script_path=os.path.abspath(__file__),
    script_auther='liyanqing.1987',
    script_startup_method='crontab',
    script_execute_frequency='every 10 minutes',
    alarm_receivers='liyanqing.1987',
    alarm_frequency='max 1 times')
...
```

SaveLog 的主要参数说明如下：

**direction**：指定业务方向，默认是“default”，可以修改，需要保证填写的项已经在 config/config.py 中的 valid\_direction\_dic 中的 key 上定义。

**monitor\_item**：监控项目描述，填写一串不能带空格的字符串，单词间可以用下划线连接。

**script\_path**：脚本路径，默认就是当前脚本，非特殊情况请不要修改默认值“\_\_file\_\_”。

**script\_auther**：脚本作者，默认是当前用户。

**script\_startup\_method**：脚本启动方式，按时情况填写，一般是“crontab”或者“Jenkins”居多。

**script\_execute\_frequency**：脚本启动频率，按实际情况填写描述，没有格式要求。

**alarm\_receivers**：如果发送报警，指定默认的报警接收人，默认是 config/config.py 中的 valid\_direction\_dic 中定义的 direction\_admin，如果是多人，用空格隔开。

**alarm\_frequency**：报警频率，当前仅支持“everytime”和“max <n> times”，前者指每次执行都报警，后者指每天最多报警 n 次，比如“max 3 times”指同一报警（同样的报警内容和报警接收人）每天最多发送 3 次。

更具体的类说明，可以参照安装目录下 common/common\_monitor.py 中 SaveLog 的类/函数说明。

## 4.2 脚本启动

监控脚本一般需要周期性启动，常用的方式是 crontab 或者 Jenkins，不建议把手工启动的脚本加入到这个系统中，会引发心跳检测异常。

下面是一个 crontab 的示例，对应#4.1 中的配置。

```
# Monitor: LSF
*/10 * * * *
/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
```

## 4.3 日志数据

依照上面 LSF/check\_host\_status 的示例，生成的日志数据如下。

```
[root@ic-admin2 check_host_status]# tree
/ic/data/CAD/it/monitorViewer/LSF/check_host_status
/ic/data/CAD/it/monitorViewer/LSF/check_host_status
├── alarm
│   └── 20240129
├── heartbeat
│   └── 20240129
├── log
│   └── 20240129
└── monitor_item.yaml
```

其中，monitor\_item.yaml 存放此监控项的基本信息。

```
alarm_frequency: max 1 times
alarm_receivers: liyanqing.1987
```

```
direction: LSF
direction_admin: liyanqing.1987
monitor_item: check_host_status
script_auther: liyanqing.1987
script_execute_frequency: every 10 minutes
script_path:
/ic/data/usr/liyanqing.1987/tools/monitorViewer/scripts/LSF/check_
host_status.py
script_startup_host: 10.212.206.197
script_startup_method: crontab
```

heartbeat 目录按天存放程序启动的心跳日志。

```
[time=2024-01-29 15:50:03] [user=ic_admin] [host=10.212.206.197]
[script=/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_
host_status.py]
[time=2024-01-29 16:00:03] [user=ic_admin] [host=10.212.206.197]
[script=/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_
host_status.py]
[time=2024-01-29 16:10:03] [user=ic_admin] [host=10.212.206.197]
[script=/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_
host_status.py]
...
```

log 目录按天存放程序的输出日志。

```
[time=2024-01-29 17:10:03] [message_level=Error] Host "n020-064-
049" become "unavail".
[time=2024-01-29 17:10:03] [message_level=Error] Host "n020-064-
162" become "unavail".
[time=2024-01-29 17:10:03] [message_level=Error] Host "n020-064-
163" become "unavail"
...
```

alarm 目录按天存放程序的报警日志。

```
[time=2024-01-29 17:10:06] [md5=704c540721680b4d89decae81fe2e26d]
[receivers=liyanqing.1987] [send_alarm_result=PASSED] n020-064-049
unavail
n020-064-162  unavail
```

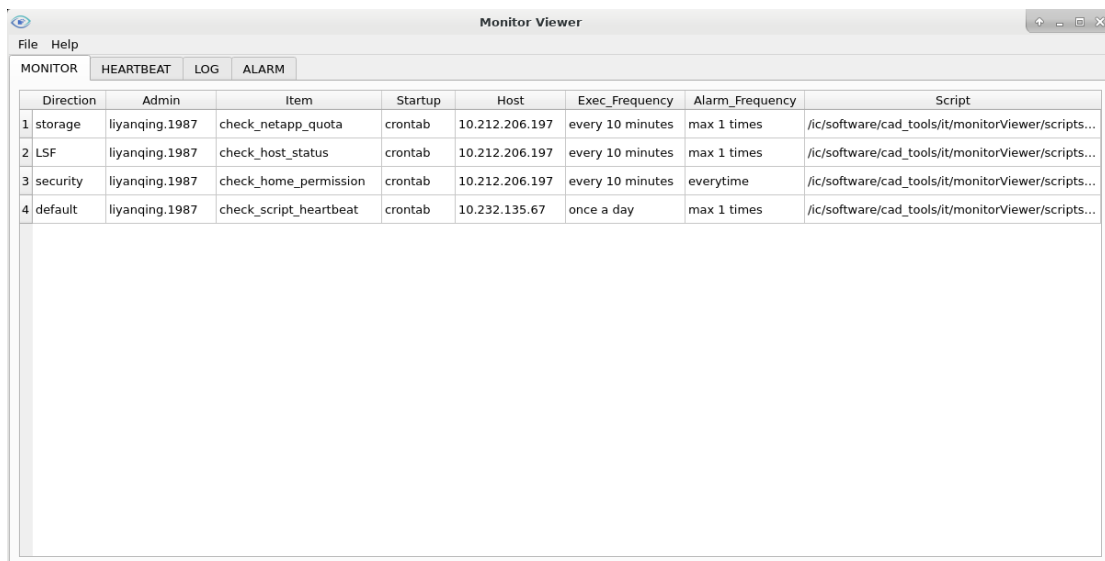
n020-064-163 unavail

...

## 4.4 监控查询 - GUI

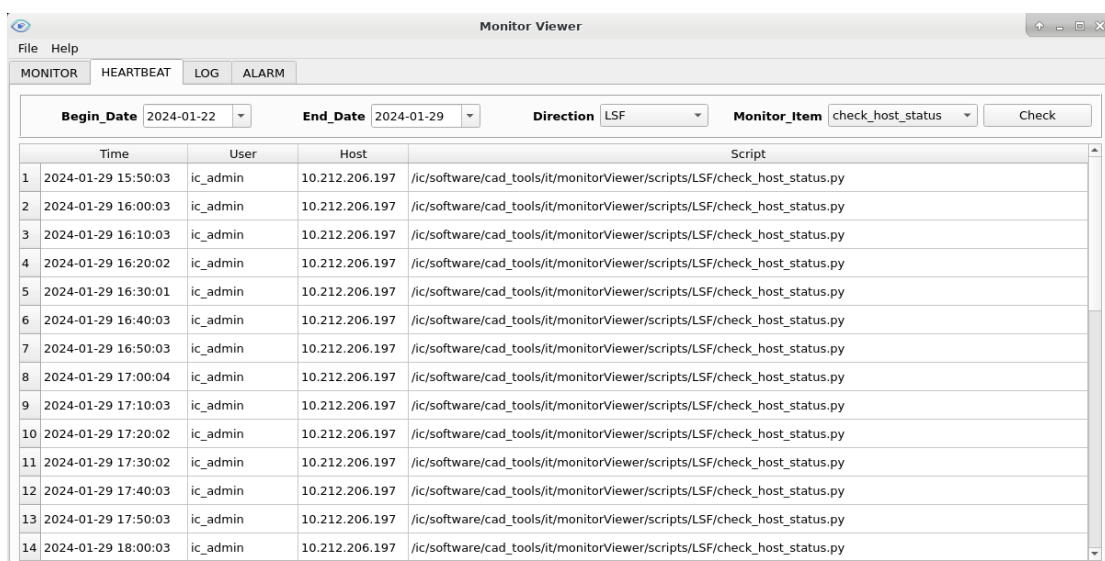
如果已经用这套框架开发了一些定制监控程序并周期性启动，那么可以使用安装目录下的 bin/monitor\_viewer 视窗来查看监控项的各种信息。

MONITOR 页用来查看所有业务方向所有监控程序的汇总信息。



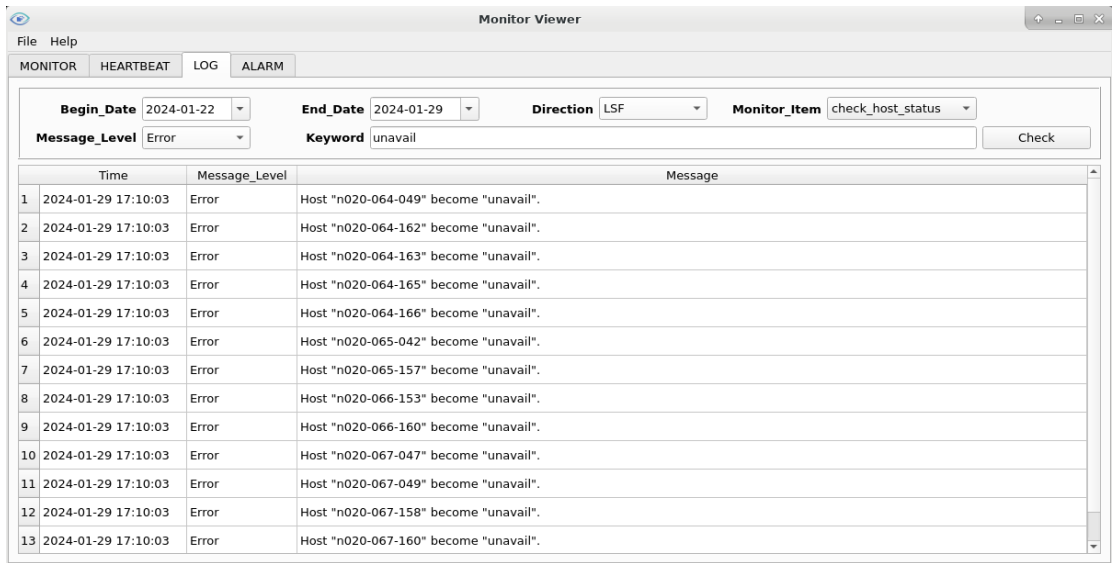
	Direction	Admin	Item	Startup	Host	Exec_Frequency	Alarm_Frequency	Script
1	storage	liyanqing.1987	check_netapp_quota	crontab	10.212.206.197	every 10 minutes	max 1 times	/ic/software/cad_tools/it/monitorViewer/scripts...
2	LSF	liyanqing.1987	check_host_status	crontab	10.212.206.197	every 10 minutes	max 1 times	/ic/software/cad_tools/it/monitorViewer/scripts...
3	security	liyanqing.1987	check_home_permission	crontab	10.212.206.197	every 10 minutes	everytime	/ic/software/cad_tools/it/monitorViewer/scripts...
4	default	liyanqing.1987	check_script_heartbeat	crontab	10.232.135.67	once a day	max 1 times	/ic/software/cad_tools/it/monitorViewer/scripts...

HEARTBEAT 页用来查看监控项的心跳日志。

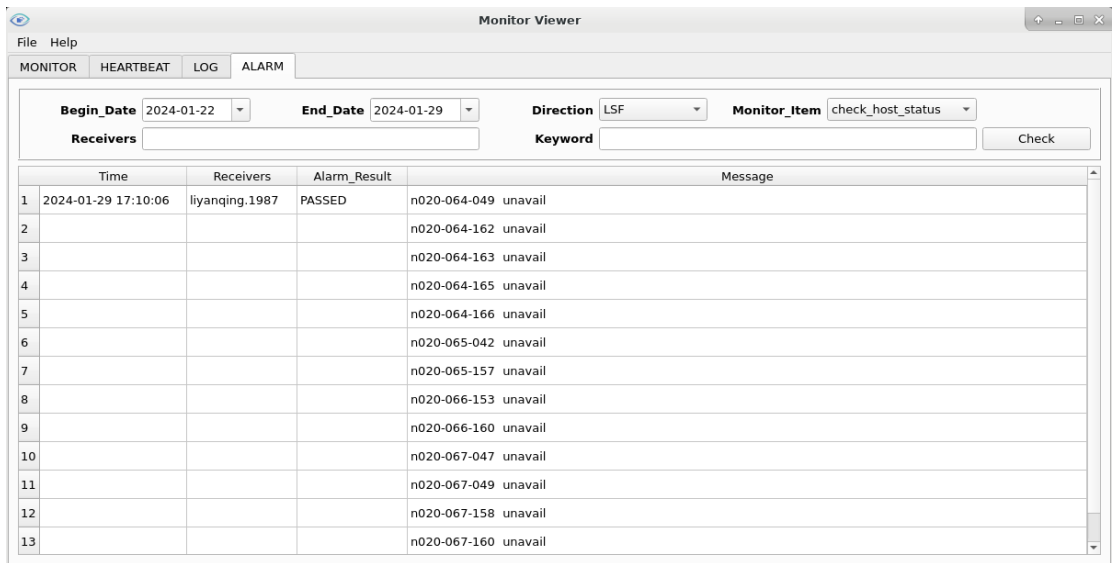


	Time	User	Host	Script
1	2024-01-29 15:50:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
2	2024-01-29 16:00:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
3	2024-01-29 16:10:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
4	2024-01-29 16:20:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
5	2024-01-29 16:30:01	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
6	2024-01-29 16:40:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
7	2024-01-29 16:50:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
8	2024-01-29 17:00:04	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
9	2024-01-29 17:10:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
10	2024-01-29 17:20:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
11	2024-01-29 17:30:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
12	2024-01-29 17:40:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
13	2024-01-29 17:50:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py
14	2024-01-29 18:00:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_host_status.py

LOG 页用来查看监控项的输出日志，同时支持按照 Message\_Level 和 Keyword 来检索日志。

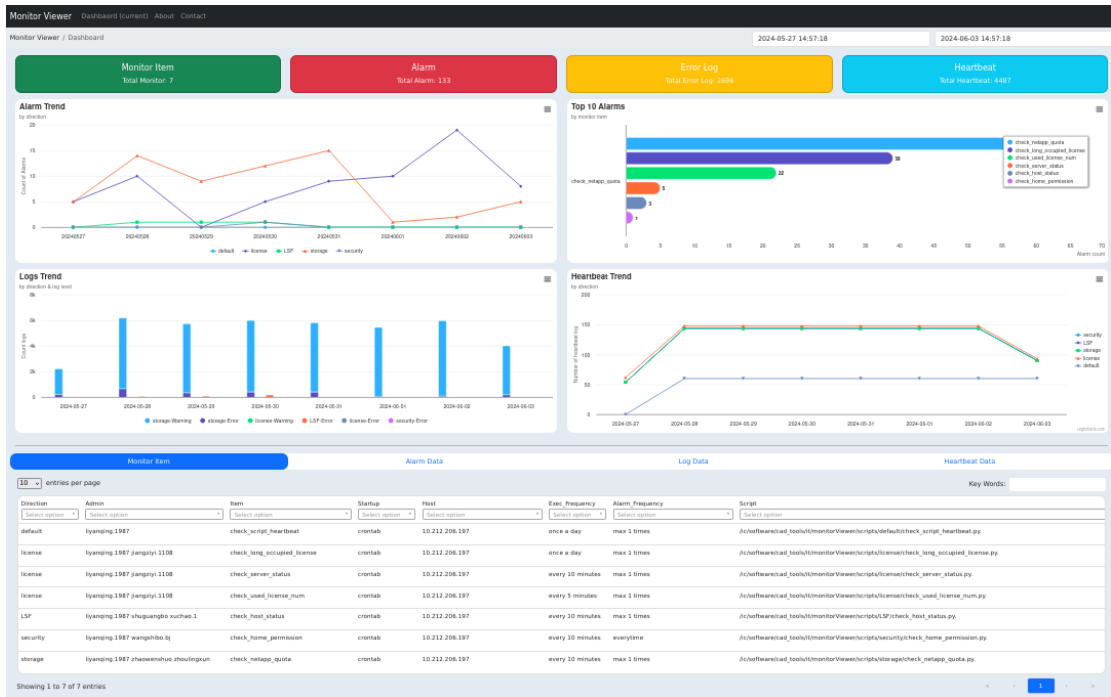


ALARM 页用来查看监控项的报警日志，同时支持按照 Receivers 和 Keyword 来检索日志。

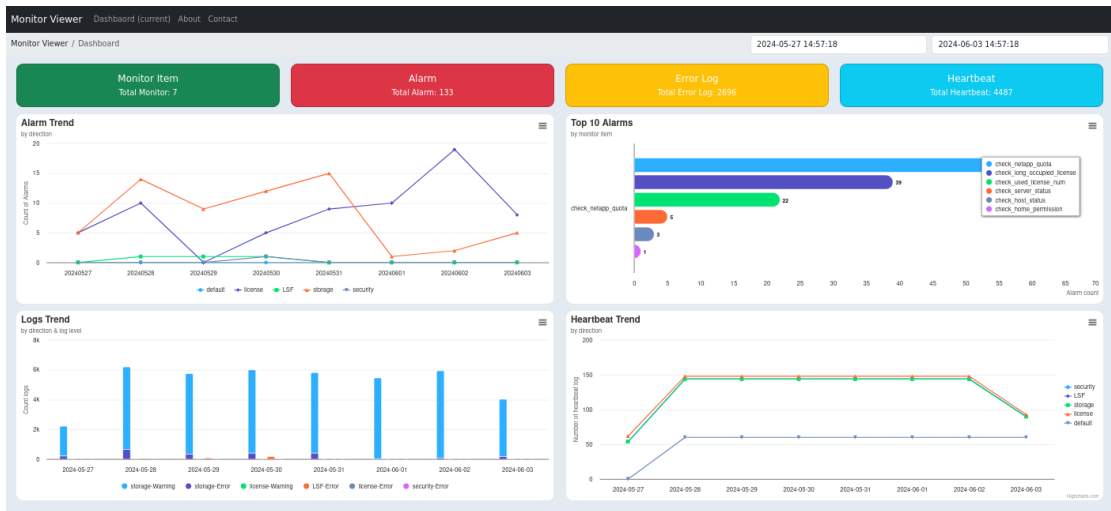


## 4.5 监控查询 - web

还可以采用 web 的方式查看监控信息，如#3.3 所示，在 firefox 中打开网址 <http://10.232.135.13:5000>。



其中上半部分是汇总信息。



下半部分可以在 Monitor Item / Alarm Data / Log Data / Heartbeat Data 之间切换，通过 Direction/Monitor\_Item 等下拉菜单选择，可以展示自己想看的历史记录。

Monitor Item

Alarm Data

Log Data

Heartbeat Data

10 entries per page

Key Words:

Direction	Monitor Item	Time	Receivers	Alarm Result	Message
storage	check_netapp_quota	Select option	Select option	Select option	Select option
storage	check_netapp_quota	2024-05-27 17:50:11	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: v-c.qianzhang.ic; volume: adas; 磁盘限制: 500GB; 磁盘用量: 500.2GB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-27 18:00:12	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: tianyangyu; volume: HOME; 磁盘限制: 5GB; 磁盘用量: 5GB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-27 18:20:09	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: xinjiaci; volume: vivian; 磁盘限制: 2TB; 磁盘用量: 2TB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-27 22:40:10	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: duhanxiao.4213; volume: orca; 磁盘限制: 1TB; 磁盘用量: 1TB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-27 23:00:10	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: wangxiao.87; volume: vivian_sta; 磁盘限制: 1TB; 磁盘用量: 1TB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-28 00:00:08	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: yupegda; volume: HOME; 磁盘限制: 5GB; 磁盘用量: 5GB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-28 00:00:10	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: wanghangwei; volume: HOME; 磁盘限制: 5GB; 磁盘用量: 5GB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-28 00:00:11	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: yunfeilong; volume: fpga; 磁盘限制: 100GB; 磁盘用量: 100GB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-28 00:00:13	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: v-c.wzengli.ic; volume: orca; 磁盘限制: 500GB; 磁盘用量: 500.9GB; quota使用率已经达到100%, 请及时清理空间。
storage	check_netapp_quota	2024-05-28 00:00:14	Ilyanqing 1987 zhaowenshuo zhoulingxun	PASSED	用户: duhanxiao.4213; volume: orca; 磁盘限制: 1TB; 磁盘用量: 1TB; quota使用率已经达到100%, 请及时清理空间。

Showing 1 to 10 of 63 entries (filtered from 133 total entries)

## 4.6 心跳监测机制

监控程序常遇到的一个问题是，监控程序可能因为各种原因停止正常运作了，但是用户一直没有发现这个故障，心跳监测机制可以有效地发现这个问题，并及时通知管理员。

monitorViewer 的心跳监控机制实现如下：

- 安装目录下的 `scripts/default/check_script_heartbeat` 程序，可以根据监控日志数据中的 `heartbeat` 数据判断程序的运行状态。
- 如果依据监控项的 `heartbeat` 日志，发现昨天和今天都是缺失的，那么可以判断这个监控程序没有被正常启动。
- `check_script_heartbeat` 发现异常后，及时发送报警给管理员。

`check_script_heartbeat` 本身也通过 `crontab` 或者 `Jenkins` 启动，每天运行一次即可。

那么 `check_script_heartbeat` 万一没有正常运行，又该如何发现呢？  
monitorViewer 有双重保险机制，可以有效地预防这一问题出现。每次 `monitor_viewer` 图形界面启动的时候，`check_script_heartbeat` 都会被手动触发，重新做一次心跳监测。

这样的话，只要这套系统经常被使用，就可以确保所有定制监控程序的心跳监测是持续有效的。



# 附录

## 附 1. 变更历史

日期	版本	变更描述	备注
2024.1.29	1.0	发布第一个正式 release 版本	
2024.5.31	1.1	增加 web 信息展示	