**monitorViewer用户手册**

**Product Name :** monitorViewer

**Product Version :** V1.0

**Release Date :** 2024.11.01

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

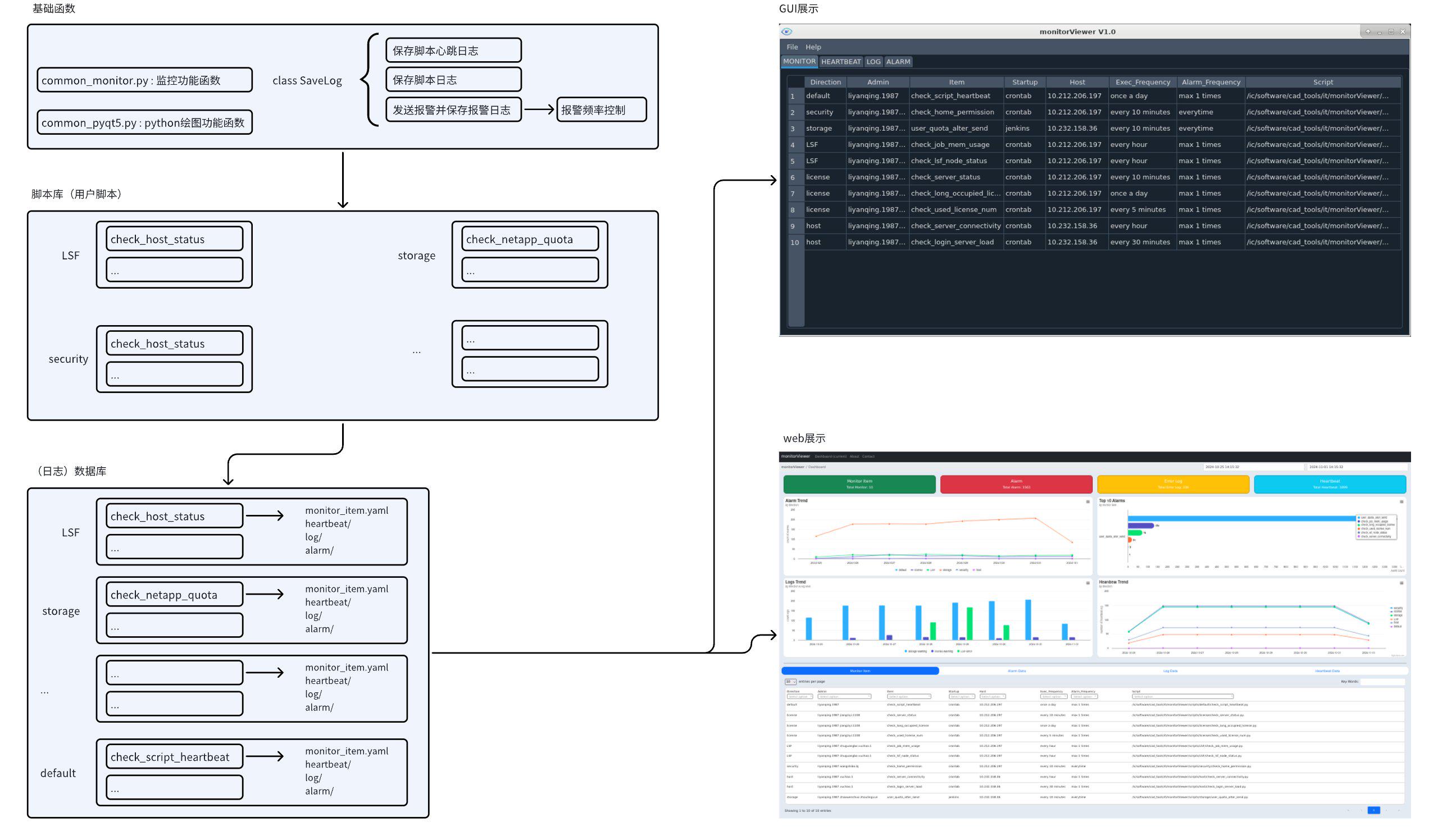
**一、简介**

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的监控，Zabbix等开源监控工具本身可以监控服务器、存储、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，需要依赖自研监控脚本实现。

自研监控脚本的开发、管理和维护往往面临如下问题：

* 监控方向和监控项目众多，监控脚本开发者也不相同，缺乏统一的开发架构和基础功能组件。
* 自研监控脚本往往缺乏日志系统，无法事后查询监控和报警记录。
* 自研监控脚本往往缺乏健壮的报警控制方式，报警频率不易控制。
* 自研监控脚本往往缺乏自检测机制，无法及时发现本身未被正常启动的问题。
* 缺乏一个顶层的前端界面，以方便查阅所有的定制监控项及其历史记录。

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/host/storage/license/security等业务。

**monitor\_item**：监控项，比如LSF中的check\_lsf\_node\_status，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**python3.12.7。**

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 工具下载**

monitorViewer的github路径位于 https://github.com/bytedance/monitorViewer

可以采用“git clone https://github.com/bytedance/monitorViewer.git”的方式拉取源代码。

|  |
| --- |
| Bash [bytedance@LQ2KW66YXF ~/Downloads]#git clone https://github.com/bytedance/monitorViewer.git Cloning into 'monitorViewer'... remote: Enumerating objects: 186, done. remote: Counting objects: 100% (186/186), done. remote: Compressing objects: 100% (119/119), done. remote: Total 186 (delta 65), reused 172 (delta 51), pack-reused 0 (from 0) Receiving objects: 100% (186/186), 5.23 MiB | 1.16 MiB/s, done. Resolving deltas: 100% (65/65), done. |

也可以在monitorViewer的github页面上，Code -> Download ZIP的方式拉取代码包。

**3.2 工具安装**

工具安装之前，首先参照第二章“环境依赖”满足batchRun的环境依赖关系。

安装包下的文件和目录如下。

|  |
| --- |
| Bash [root@ic-admin2 monitorViewer]# ls bin common config data db docs install.py lib LICENSE README.md requirements.txt scripts tools web |

确认python版本正确。

|  |
| --- |
| Bash [root@ic-admin2 batchRun]# which python3 /ic/software/tools/python3/3.12.7/bin/python3 |

使用跟python3.12.7对应的pip3，基于安装包中的requirements.txt安装python依赖库。（可能需要root权限）

|  |
| --- |
| Bash [root@ic-admin2 monitorViewer]# pip3 install -r requirements.txt  Looking in indexes: http://bytedpypi.byted.org/simple Requirement already satisfied: Flask==3.0.3 in //ic/software/tools/python3/3.12.7/lib/python3.12/site-packages (from -r requirements.txt (line 1)) (3.0.3) ... |

在安装目录下，使用命令“python3 install.py”安装monitorViewer。

|  |
| --- |
| Bash [root@ic-admin2 monitorViewer]# python3 install.py  >>> Check python version.  Required python version : (3, 12)  Current python version : (3, 12)  >>> Generate script "/ic/software/tools/monitorViewer/bin/monitor\_viewer". >>> Generate script "/ic/software/tools/monitorViewer/scripts/gen\_monitor\_script". >>> Generate script "/ic/software/tools/monitorViewer/scripts/default/check\_script\_heartbeat". >>> Generate script "/ic/software/tools/monitorViewer/tools/patch". >>> Generate config file "/ic/software/tools/monitorViewer/config/config.py". >>> Generate run web script "web/run.sh".  Done, Please enjoy it. |

**3.3 工具配置**

安装目录下的config/config.py用于配置工具的一些基本设置和验证规则。

|  |
| --- |
| Bash # Specify valid direction and direction\_admin. valid\_direction\_dic = {  "default": "root", }  # Specify database path. db\_path = "/ic/software/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**：发送报警的命令及其参数（部分参数项可以用变量取代）。

下面是修改后应用在业务环境中的一个配置。

|  |
| --- |
| Bash # Specify valid direction and direction\_admin. valid\_direction\_dic = {  "default": "liyanqing.1987",  "license": "liyanqing.1987 \*\*\*",  "LSF": "liyanqing.1987 \*\*\* \*\*\*",  "storage": "liyanqing.1987 \*\*\* \*\*\*",  "security": "liyanqing.1987 \*\*\*",  "host": "liyanqing.1987 \*\*\*", }  # Specify database path. db\_path = "/ic/software/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 = "/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。

**3.4 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/虚机上直接启动。

|  |
| --- |
| Bash [root@ic-admin2 monitorViewer]# cd web/ [root@ic-admin2 web]# ls app.py run.sh service static templates tools [root@ic-admin2 web]# nohup ./run.sh & [1] 45773 [root@ic-admin2 web]# 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。

|  |
| --- |
| Bash [root@ic-admin2 web]# cat nohup.out   \* Debug mode: off 2024-11-01 15:16:22,890 | 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://10.232.158.36:5000 2024-11-01 15:16:22,890 | INFO | Press CTRL+C to quit |

在Linux系统中，通过firefox访问http://10.232.158.36:5000，可以查看monitorViewer的web信息看板。

**四、工具使用**

**4.1 开发脚本**

monitorView默认的脚本库位于安装目录下的scripts/下。

|  |
| --- |
| Bash [root@ic-admin2 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，但是更推荐采用monitorViewer自带的gen\_monitor\_script程序生成脚本，已完成SaveLog类的自动初始化。

|  |
| --- |
| Bash [root@ic-admin2 monitorViewer]# mkdir scripts/security [root@ic-admin2 monitorViewer]# cd scripts/security/ [root@ic-admin2 security]# ../gen\_monitor\_script check\_home\_permission.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。

|  |
| --- |
| Bash ... os.environ['MONITOR\_VIEWER\_INSTALL\_PATH'] = '/ic/software/tools/monitorViewer'  sys.path.append(str(os.environ['MONITOR\_VIEWER\_INSTALL\_PATH']) + '/common') import common\_monitor  save\_log\_ins = common\_monitor.SaveLog(  direction='',  monitor\_item='',  script\_path=os.path.abspath(\_\_file\_\_),  script\_auther='root',  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。

|  |
| --- |
| Bash ... os.environ['MONITOR\_VIEWER\_INSTALL\_PATH'] = '/ic/software/cad\_tools/it/monitorViewer'  sys.path.append(str(os.environ['MONITOR\_VIEWER\_INSTALL\_PATH']) + '/common') import common\_monitor  save\_log\_ins = common\_monitor.SaveLog(  direction='security',  monitor\_item='check\_home\_permission',  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='everytime') ... |

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valida\_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。

那么save\_log\_ins又是如何在用户的监控脚本中起到保存日志、发送报警、控制报警频率的功能呢？下面是具体的代码。

|  |
| --- |
| Bash def check\_home\_permission():  """  Check directory permission for all user home dir.  """  alarm\_title = 'Security Alarm: home目录权限被篡改!'   for user\_name in os.listdir('/home'):  if re.match(r'^\.\S+$', user\_name):  continue   home\_dir = '/home/' + str(user\_name)   if os.path.isdir(home\_dir):  home\_dir\_stat\_info = os.stat(home\_dir)  home\_dir\_permission = oct(home\_dir\_stat\_info.st\_mode)[-3:]   if home\_dir\_permission != '700':  # Save log  message = 'For home directory "' + str(home\_dir) + '", permission is modified into "' + str(home\_dir\_permission) + '".'  save\_log\_ins.save\_log(message, message\_level='Error', print\_mode=True)   try:  os.chmod(home\_dir, 0o700)  except Exception as error:  print(error)   # Send alarm  alarm\_receivers = str(user\_name) + ' liyanqing.1987'  alarm\_message = 'Hi ' + str(user\_name) + ', 监控发现' + str(home\_dir) + '目录权限被篡改为' + str(home\_dir\_permission) + ', 因系统安全要求, 现将其强制修改回700, 请勿修改个人home目录权限, 谢谢.'  save\_log\_ins.send\_alarm(message=alarm\_message, alarm\_title=alarm\_title, alarm\_receivers=alarm\_receivers) |

**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。

|  |
| --- |
| Bash # Monitor: security - check\_home\_permission \*/10 \* \* \* \* /ic/software/tools/monitorViewer/scripts/security/check\_home\_permission.py |

当用户的HOME目录权限被修改为非“700”的时候，执行监控程序会触发报警。

|  |
| --- |
| Bash [root@ic-admin2 security]# chmod 777 /home/liyanqing.1987 [root@ic-admin2 security]#  [root@ic-admin2 security]# ./check\_home\_permission.py  \*Error\*: For home directory "/home/liyanqing.1987", permission is modified into "777". |

根据#4.1中的配置，send\_lark脚本会给指定管理员发送如下飞书报警信息。



**4.3 日志数据**

依照上面secutiry/check\_home\_permission的示例，生成的日志数据如下。

|  |
| --- |
| Bash [root@ic-admin2 monitorViewer]# tree db/security/check\_home\_permission db/security/check\_home\_permission ├── alarm │   └── 20241101 ├── heartbeat │   └── 20241101 ├── log │   └── 20241101 └── monitor\_item.yaml  3 directories, 4 files |

其中，monitor\_item.yaml存放此监控项的基本信息。

|  |
| --- |
| Bash alarm\_frequency: everytime alarm\_receivers: liyanqing.1987 direction: security direction\_admin: liyanqing.1987 monitor\_item: check\_home\_permission script\_auther: liyanqing.1987 script\_execute\_frequency: every 10 minutes script\_path: /ic/software/tools/monitorViewer/scripts/security/check\_home\_permission.py script\_startup\_host: 10.232.158.36 script\_startup\_method: crontab |

heartbeat目录按天存放程序启动的心跳日志，单行都是json格式。

|  |
| --- |
| Bash {"time": "2024-11-01 16:56:36", "user": "root", "host": "10.232.158.36", "script": "/ic/software/tools/monitorViewer/scripts/security/check\_home\_permission.py"} ... |

log目录按天存放程序的输出日志。

|  |
| --- |
| Bash {"time": "2024-11-01 16:56:36", "message\_level": "Error", "message": "For home directory \"/home/liyanqing.1987\", permission is modified into \"777\"."} ... |

alarm目录按天存放程序的报警日志。

|  |
| --- |
| Bash {"time": "2024-11-01 16:56:39", "md5": "e700d9b1070b02290eaf0c06d9f66e85", "receivers": "liyanqing.1987 liyanqing.1987 wangshibo.bj", "send\_alarm\_result": "PASSED", "message": "Hi liyanqing.1987, 监控发现/home/liyanqing.1987目录权限被篡改为777, 因系统安全要求, 现将其强制修改回700, 请勿修改个人home目录权限, 谢谢."} ... |

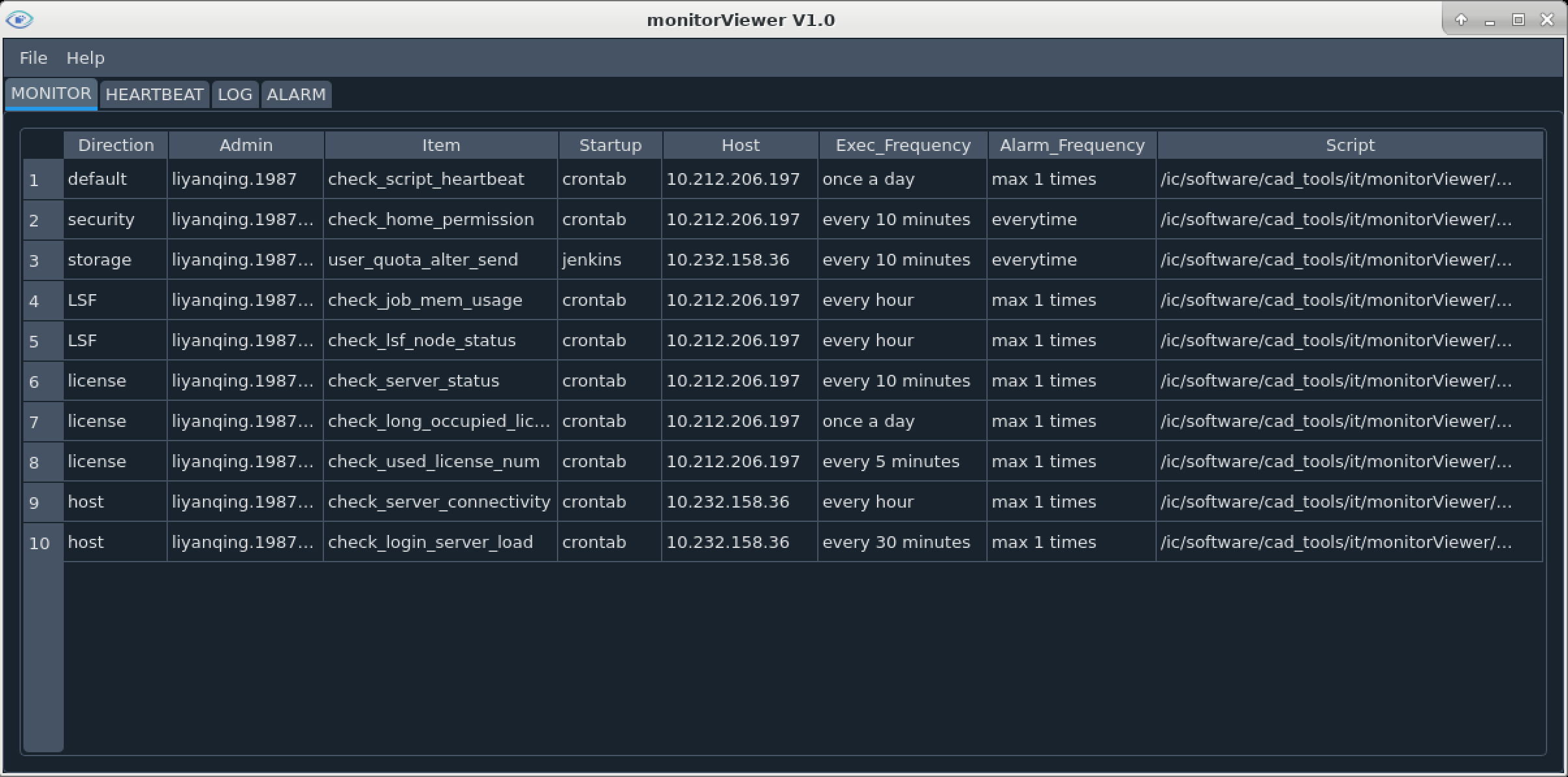
**4.4 监控查询 - GUI**

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\_viewer程序包含MONITOR/HEARTBEAT/LOG/ALARM四个页面。

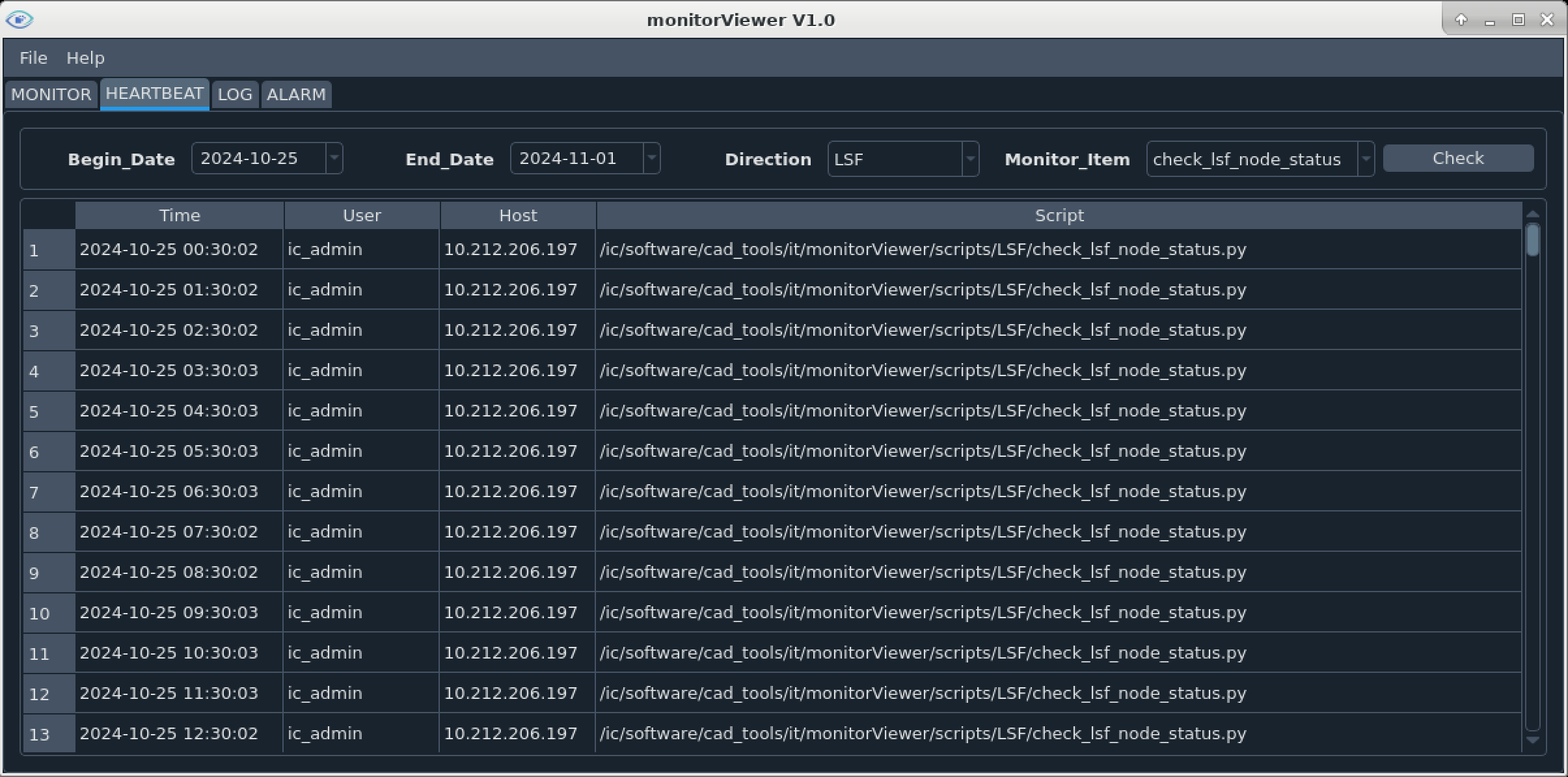
**MONITOR页：**

用来查看所有业务方向所有监控程序的汇总信息。



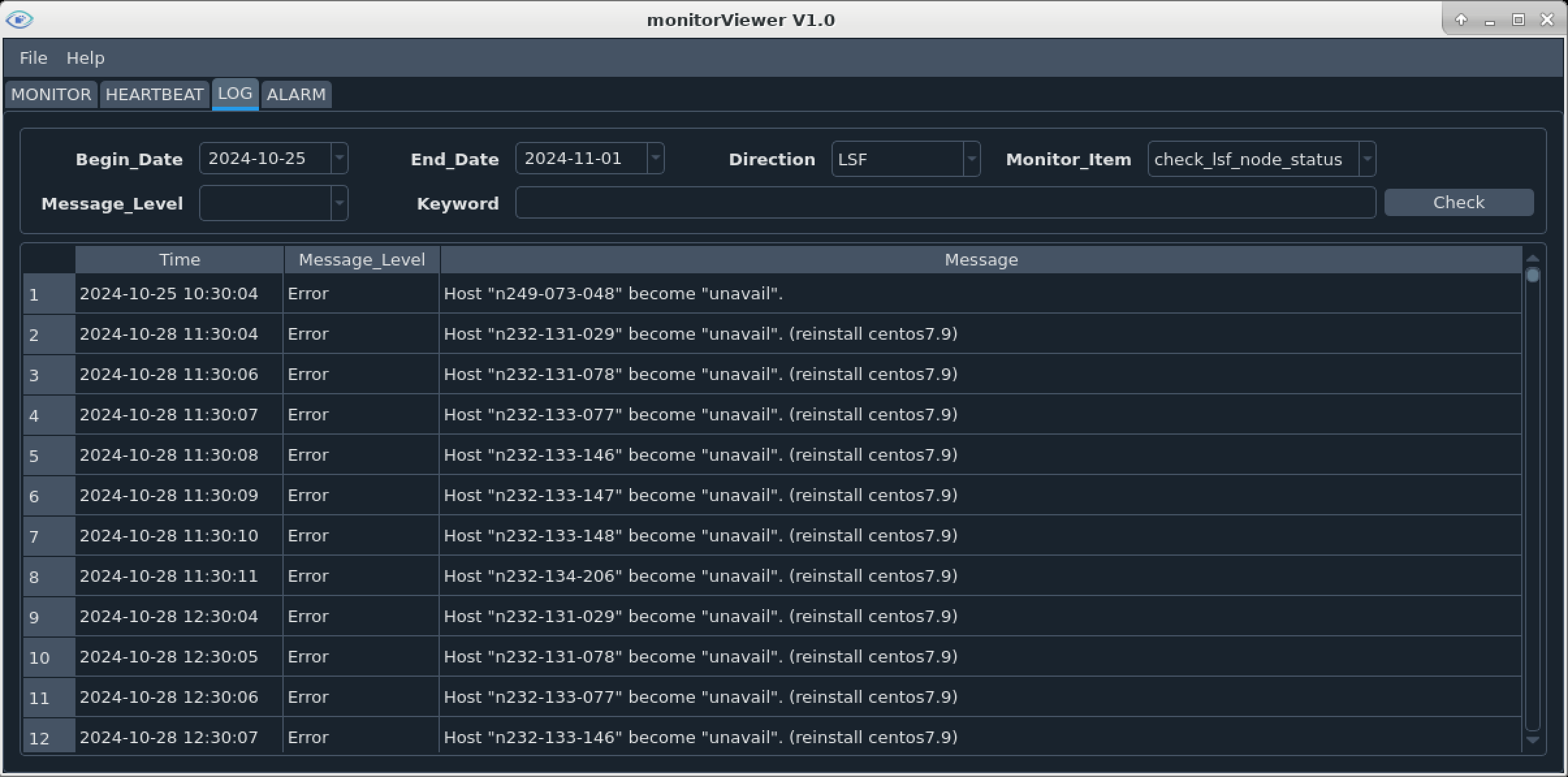
**HEARTBEAT页：**

用来查看监控项的心跳日志。



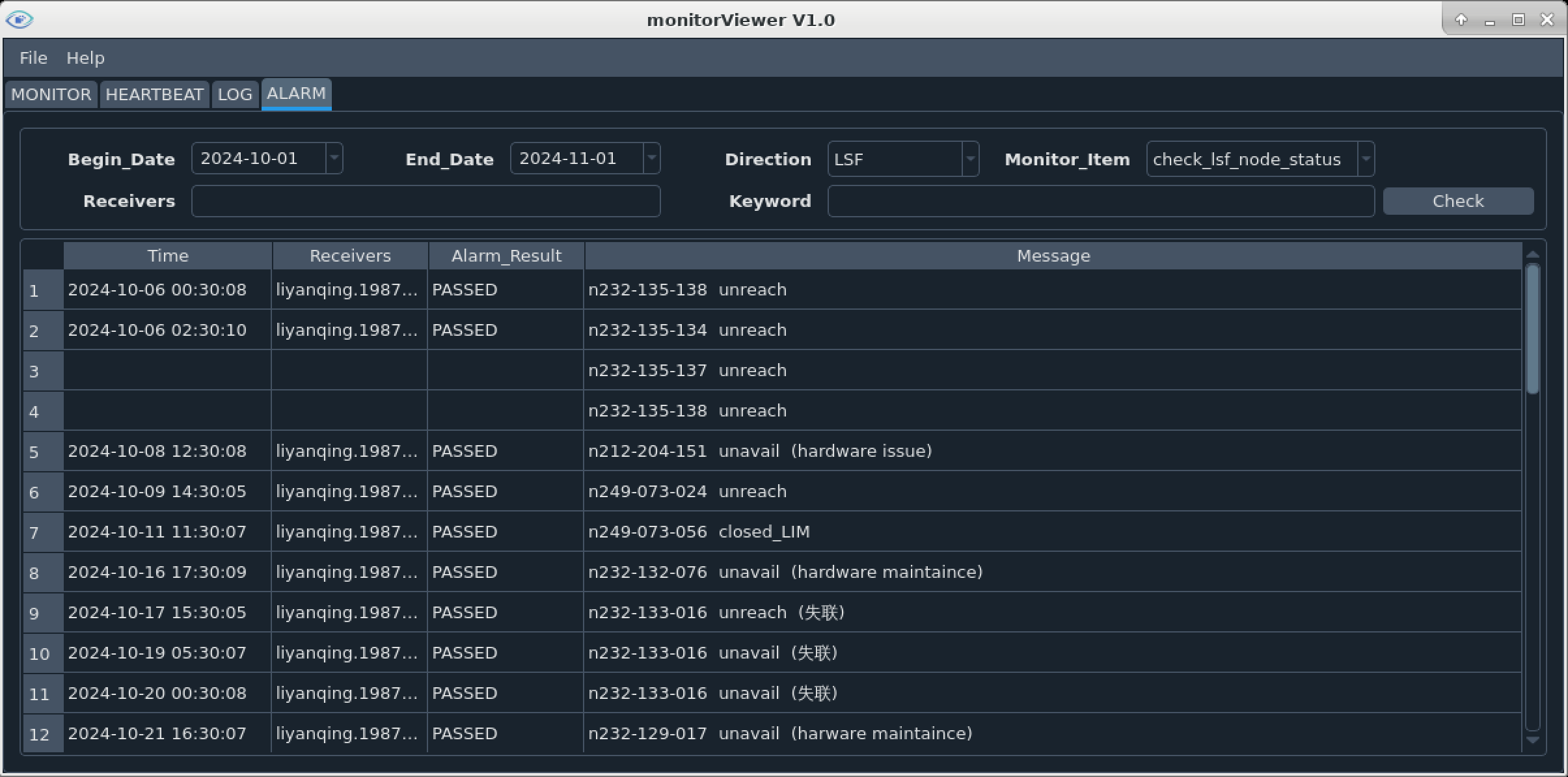
**LOG页：**

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。



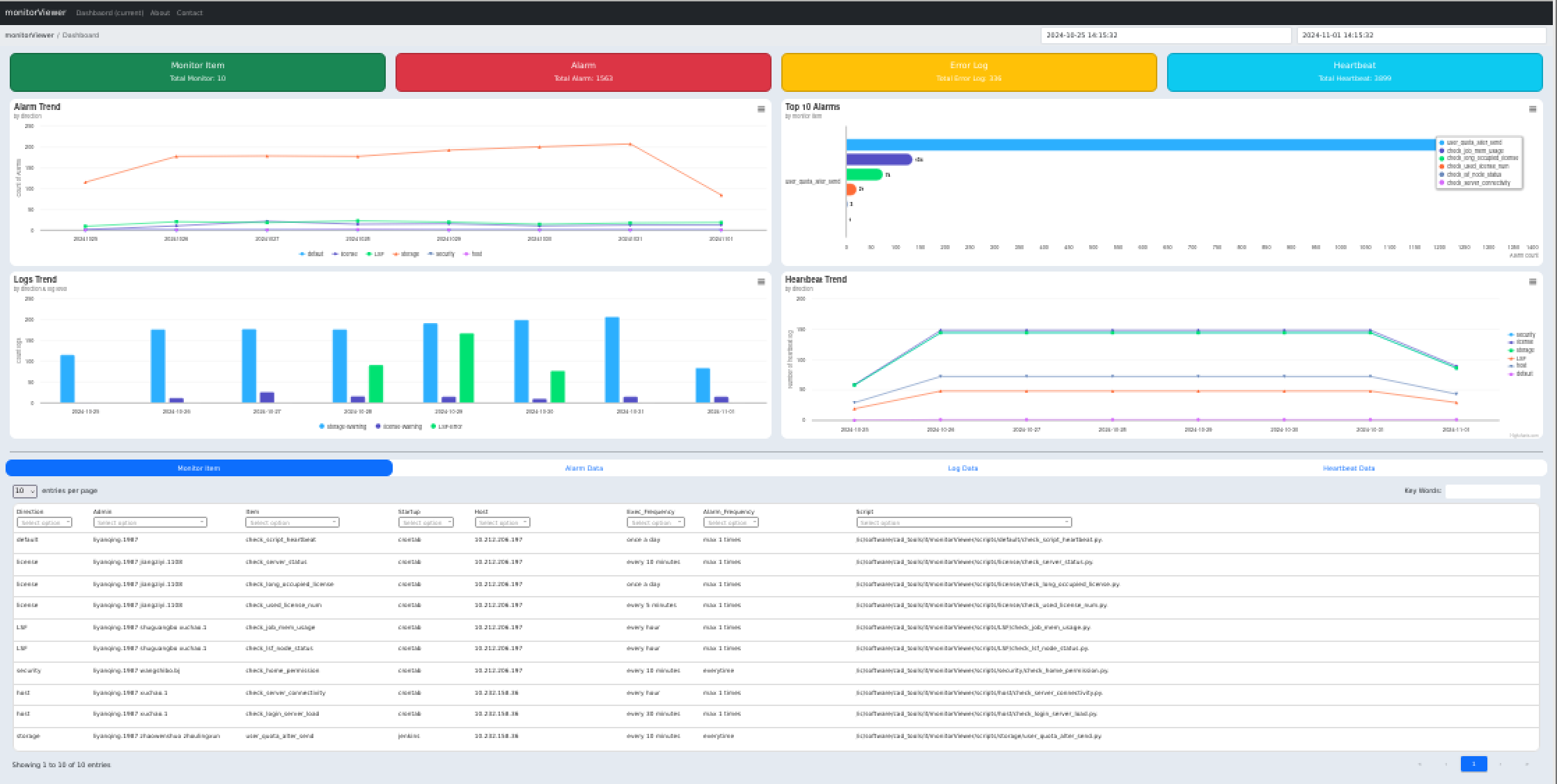
**ALARM页：**

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://10.232.158.36:5000。



其中上半部分是汇总信息。

下半部分可以在Monitor Item / Alarm Data / Log Data / Heatbeat 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。



**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** |
| 2024.11 | 1.0 | 发布第一个正式release版本 |  |
|  |  |  |  |