

# 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)

## 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 工具下载.....	6
3.2 工具安装.....	7
3.3 工具配置.....	8
3.4 启动 WEB 服务 .....	10
四、工具使用 .....	12
4.1 开发脚本.....	12
4.2 脚本启动.....	15
4.3 日志数据.....	15
4.4 监控查询 - GUI.....	17
4.5 监控查询 - WEB.....	19
4.6 心跳监测机制.....	20
五、技术支持 .....	22
附录.....	23
附 1. 变更历史.....	23

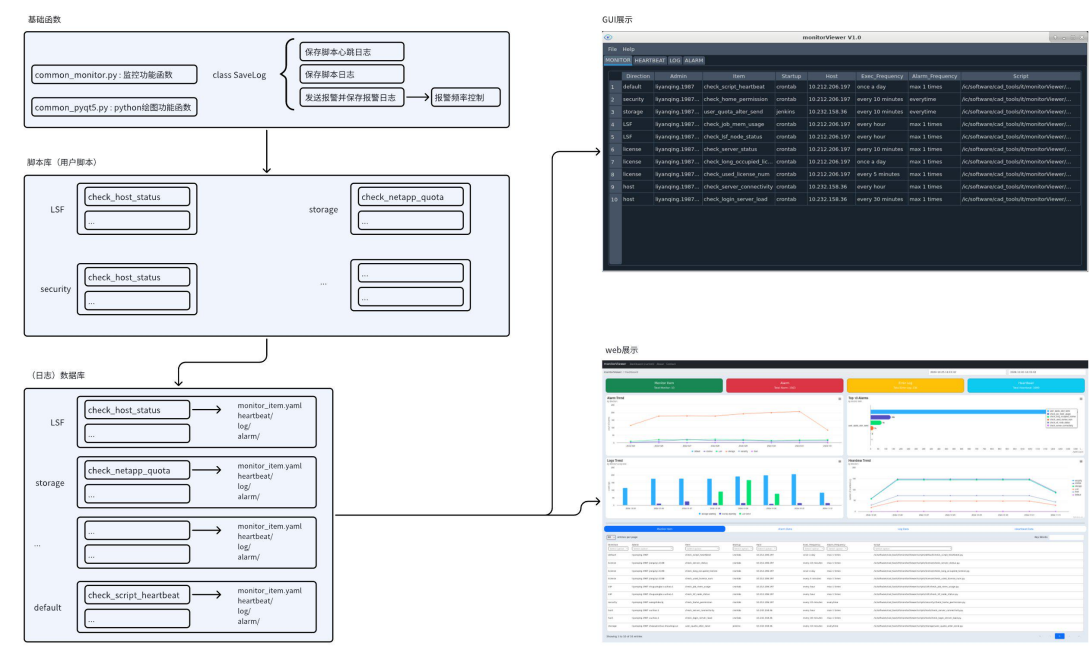
# 一、简介

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

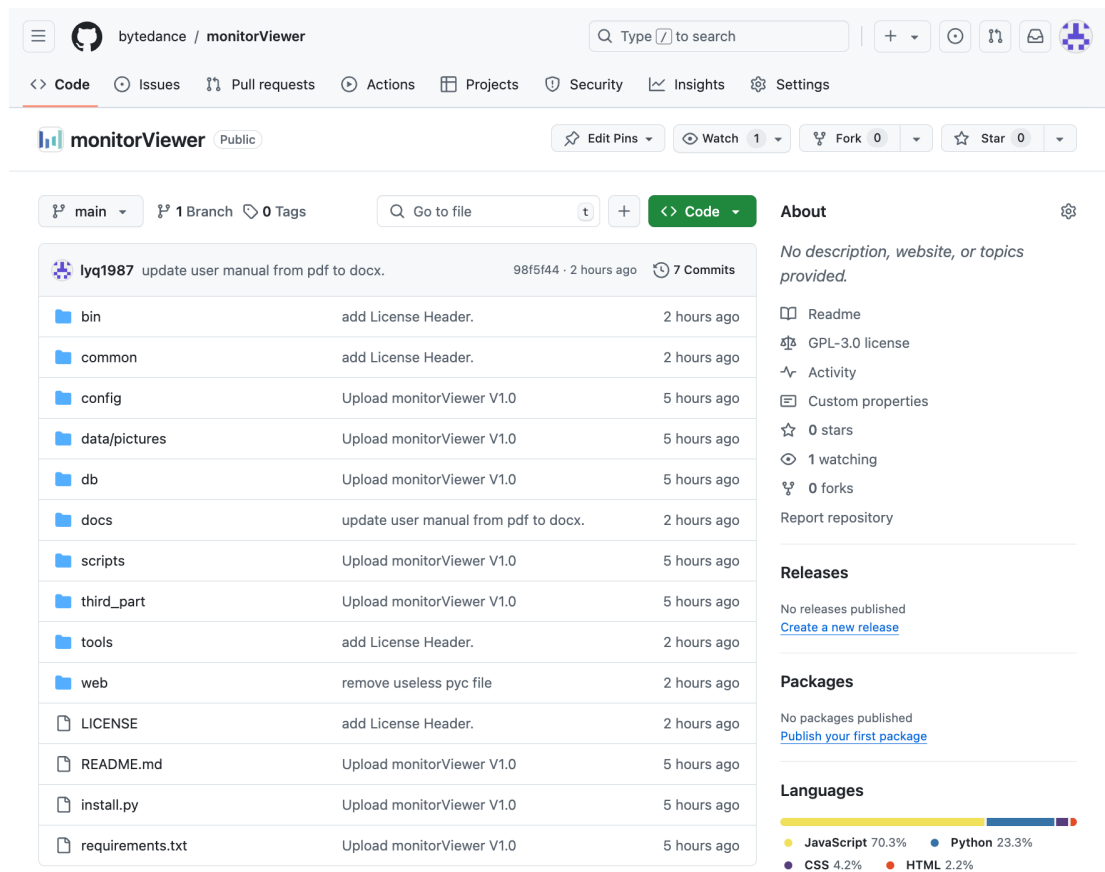
monitorViewer 基于 python 开发, 其开发和测试的 python 版本为 **python3.12.7**。

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

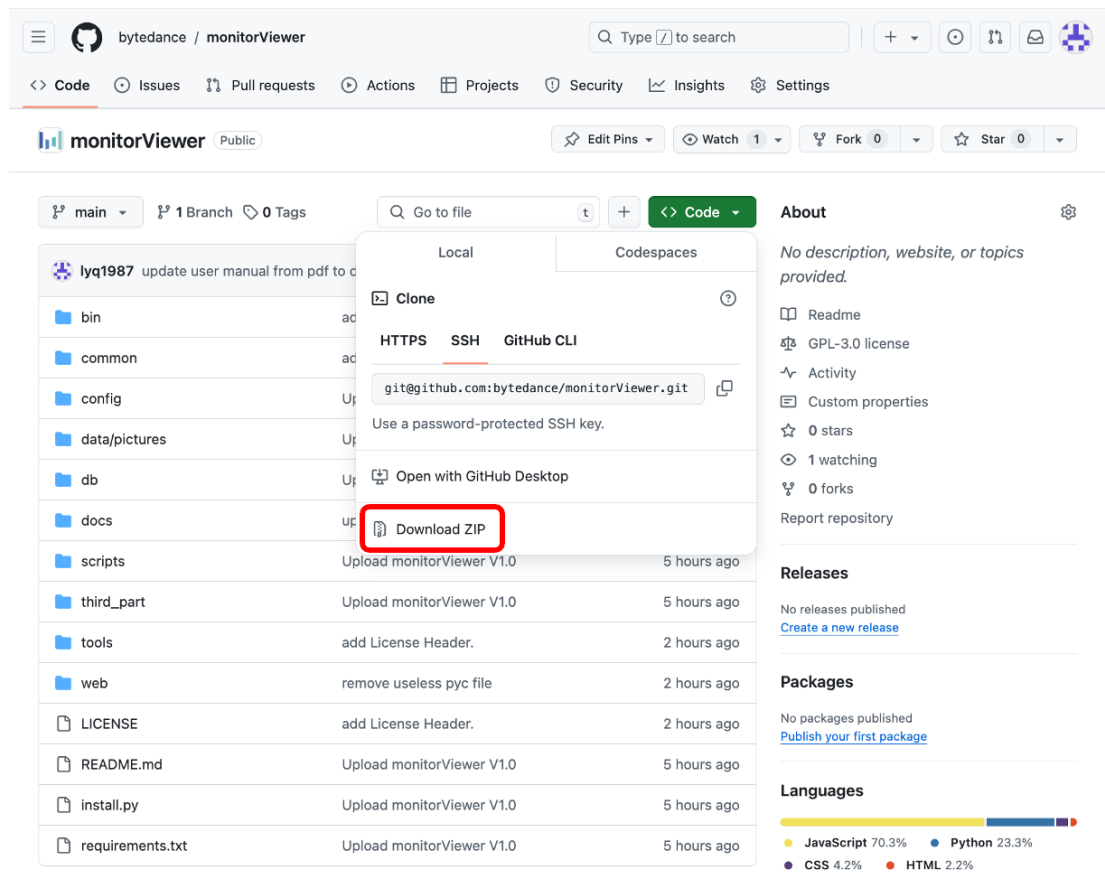
monitorViewer 的 github 路径位于 <https://github.com/bytedance/monitorViewer>



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

```
[bytedance@LQ2KW66YXF ~/Downloads/github]#git clone
https://github.com/bytedance/monitorViewer.git
Cloning into 'monitorViewer'...
remote: Enumerating objects: 106, done.
remote: Counting objects: 100% (106/106), done.
remote: Compressing objects: 100% (81/81), done.
remote: Total 106 (delta 16), reused 103 (delta 13), pack-reused 0
(from 0)
Receiving objects: 100% (106/106), 3.83 MiB | 1.39 MiB/s, done.
Resolving deltas: 100% (16/16), 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。

```
[root@ic-admin2 tools]# cd monitorViewer
[root@ic-admin2 monitorViewer]# ls -p
bin/  common/  config/  data/  db/  docs/  install.py  LICENSE
README.md  requirements.txt  scripts/  third_part/  tools/  web/
```

确认 python3 和 pip3 版本正确。

```
[root@ic-admin2 monitorViewer]# which python3
/ic/software/tools/python3/3.12.7/bin/python3
[root@ic-admin2 monitorViewer]#
[root@ic-admin2 monitorViewer]# which pip3
```

```
/ic/software/tools/python3/3.12.7/bin/pip3
```

使用跟 python3 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）

```
[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。

```
[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_hear
rtbeat".
>>> 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。

```
# 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）。

请注意，这个参数建议配置，否则无法发送报警，但是不同公司允许从 IC 环境向办公网环境发送报警的命令并不相同，请用户根据自己的情况配置。

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 *** ***",
    "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。

```
[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。

```
[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 默认脚本库位于安装目录下的 scripts/下。

```
[root@ic-admin2 monitorViewer]# ls -p 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。

```
[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。

```
...  
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。

```
...
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 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。

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。

```
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。

```
# 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。

```
[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。

### Security Alarm: home目录权限被篡改!

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.

## 4.3 日志数据

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

```
[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。

```
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。

```
{"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。

```
{"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。

```
{"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。

```
[root@ic-admin2 monitorViewer]# bin/monitor_viewer  
*Error*: Heartbeat check script have been stoped abnormally for  
over a day.  
*Info*: Execute heartbeat check script  
"/ic/software/tools/monitorViewer/scripts/default/check_script_hea  
rtbeat" ...
```

第一次执行它会报一个心跳检查异常的警告, 后续正常规律运行监控项就不会再报了, 可以忽略。

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。

monitorViewer V1.0								
File Help								
MONITOR HEARTBEAT LOG ALARM								
	Direction	Admin	Item	Startup	Host	Exec_Frequency	Alarm_Frequency	Script
1	default	liyanqing.1987...	check_script_heartbeat	crontab	10.212.206.197	once a day	max 1 times	/ic/software/cad_tools/it/monitorViewer/...
2	security	liyanqing.1987...	check_home_permission	crontab	10.212.206.197	every 10 minutes	everytime	/ic/software/cad_tools/it/monitorViewer/...
3	storage	liyanqing.1987...	user_quota_alter_send	jenkins	10.232.158.36	every 10 minutes	everytime	/ic/software/cad_tools/it/monitorViewer/...
4	LSF	liyanqing.1987...	check_job_mem_usage	crontab	10.212.206.197	every hour	max 1 times	/ic/software/cad_tools/it/monitorViewer/...
5	LSF	liyanqing.1987...	check_lsf_node_status	crontab	10.212.206.197	every hour	max 1 times	/ic/software/cad_tools/it/monitorViewer/...
6	license	liyanqing.1987...	check_server_status	crontab	10.212.206.197	every 10 minutes	max 1 times	/ic/software/cad_tools/it/monitorViewer/...
7	license	liyanqing.1987...	check_long_occupied_lic...	crontab	10.212.206.197	once a day	max 1 times	/ic/software/cad_tools/it/monitorViewer/...
8	license	liyanqing.1987...	check_used_license_num	crontab	10.212.206.197	every 5 minutes	max 1 times	/ic/software/cad_tools/it/monitorViewer/...
9	host	liyanqing.1987...	check_server_connectivity	crontab	10.232.158.36	every hour	max 1 times	/ic/software/cad_tools/it/monitorViewer/...
10	host	liyanqing.1987...	check_login_server_load	crontab	10.232.158.36	every 30 minutes	max 1 times	/ic/software/cad_tools/it/monitorViewer/...

HEARTBEAT 页：

用来查看监控项的心跳日志。

monitorViewer V1.0

FileHelp

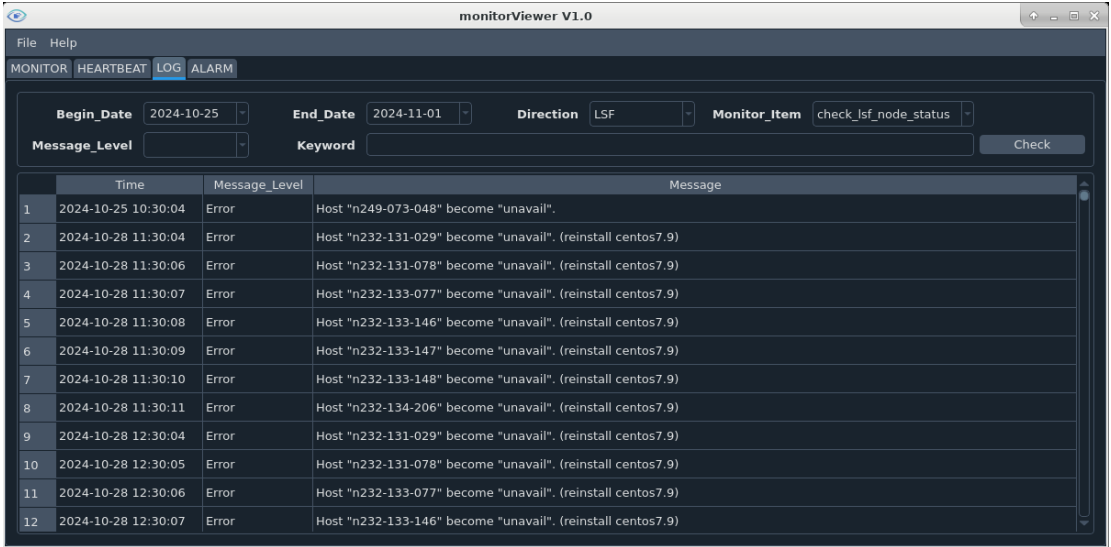
MONITORHEARTBEATLOGALARM

Begin\_Date2024-10-25End\_Date2024-11-01DirectionLSFMonitor\_Itemcheck\_lsf\_node\_statusCheck

	Time	User	Host	Script
1	2024-10-25 00:30:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
2	2024-10-25 01:30:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
3	2024-10-25 02:30:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
4	2024-10-25 03:30:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
5	2024-10-25 04:30:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
6	2024-10-25 05:30:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
7	2024-10-25 06:30:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
8	2024-10-25 07:30:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
9	2024-10-25 08:30:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
10	2024-10-25 09:30:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
11	2024-10-25 10:30:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
12	2024-10-25 11:30:03	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py
13	2024-10-25 12:30:02	ic_admin	10.212.206.197	/ic/software/cad_tools/it/monitorViewer/scripts/LSF/check_lsf_node_status.py

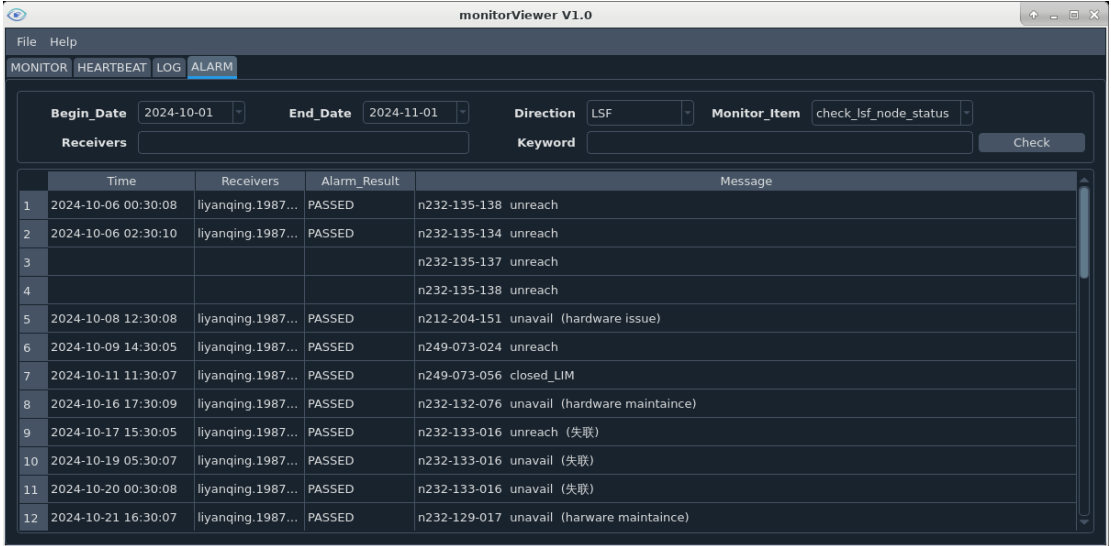
LOG 页：

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。



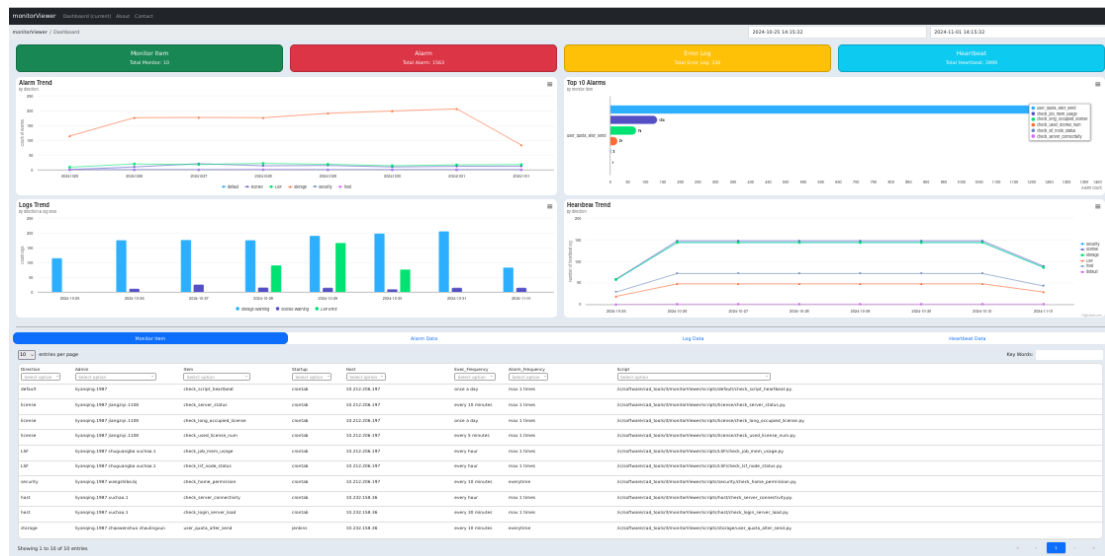
ALARM 页：

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://10.232.158.36:5000>。



其中上半部分是汇总信息。

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

Direction	Monitor Item	Type	Status	Message
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat
check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat	check_script_heartbeat

## 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。

## 五、技术支持

本工具为开源工具，由开源社区维护，可以提供如下类型的技术支持：

- 部署和使用技术指导。
- 接收 bug 反馈并修复。
- 接收功能修改建议。（需审核和排期）

获取技术支持的方式包括：

- 通过 Contact 邮箱联系开发者。
- 添加作者微信“liyanqing\_1987”，注明“真实姓名/公司/monitorViewer”，由作者拉入技术支持群。



# 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	