

batchRun 用户手册

Product Name : batchRun

Product Version : V1.2

Release Date : 2024.8.23

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

@张静文 (zhangjingwen.silvia@bytedance.com)

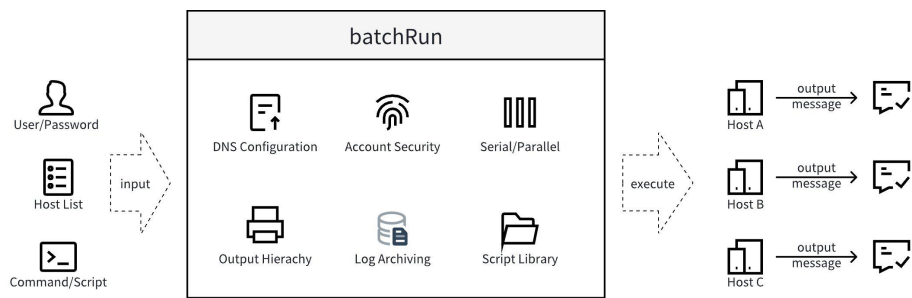
目录

一、简介.....	3
二、环境依赖	4
2.1 操作系统依赖.....	4
2.2 PYTHON 版本依赖.....	4
三、工具安装及配置.....	5
3.1 工具下载.....	5
3.2 工具安装.....	6
3.3 工具配置.....	7
3.3.1 config/config.py.....	7
3.3.2 config/host.list.....	8
3.3.3 config/password.encrypted.....	11
四、工具使用	12
4.1 工具载入.....	12
4.2 帮助信息.....	12
4.3 功能介绍.....	14
4.3.1 打印 batchRun 版本信息。.....	14
4.3.2 列出指定机器.....	15
4.3.3 指定用户名和密码登录远程机器.....	15
4.3.4 采用内置加密密码登录远程机器.....	16
4.3.5 指定机器执行命令.....	17
4.3.6 指定机器组执行命令.....	18
4.3.7 执行命令 (command)	19
4.3.8 执行脚本 (script)	19
4.3.9 串行执行和并行执行.....	20
4.3.10 超时时间 (timeout)	21
4.3.11 输出信息层级.....	21
4.3.12 输出文件.....	23
4.4 日志功能.....	23
五、辅助工具	25
5.1 TOOLS/SWITCH_ETC_HOSTS	25
5.2 SAVE_PASSWORD.....	28
5.3 SAMPLE_HOST_INFO	29
5.4 PATCH.....	31
六、技术支持	33
附录.....	34
附 1. 变更历史.....	34

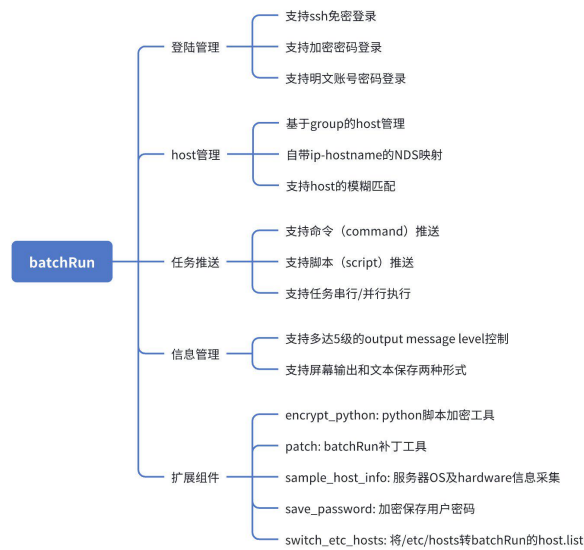
一、简介

batchRun 是 ByteDance 开源的一款适用于 Linux 环境的 IT 运维自动化工具，跟 pssh 或者 ansible 类似，可以用于多服务器的任务推送、应用部署、配置管理及信息获取等。

batchRun 基于 python 语言开发，底层使用 ssh 实现多服务器登录管理，无客户端，配置和使用较为便捷。



batchRun 支持的主要 feature 如下：



二、环境依赖

2.1 操作系统依赖

batchRun 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。

centos/redhat/rocky/ubuntu 都可以运行，主要风险在于系统库版本差异可能会影响部分组件的运行。

2.2 python 版本依赖

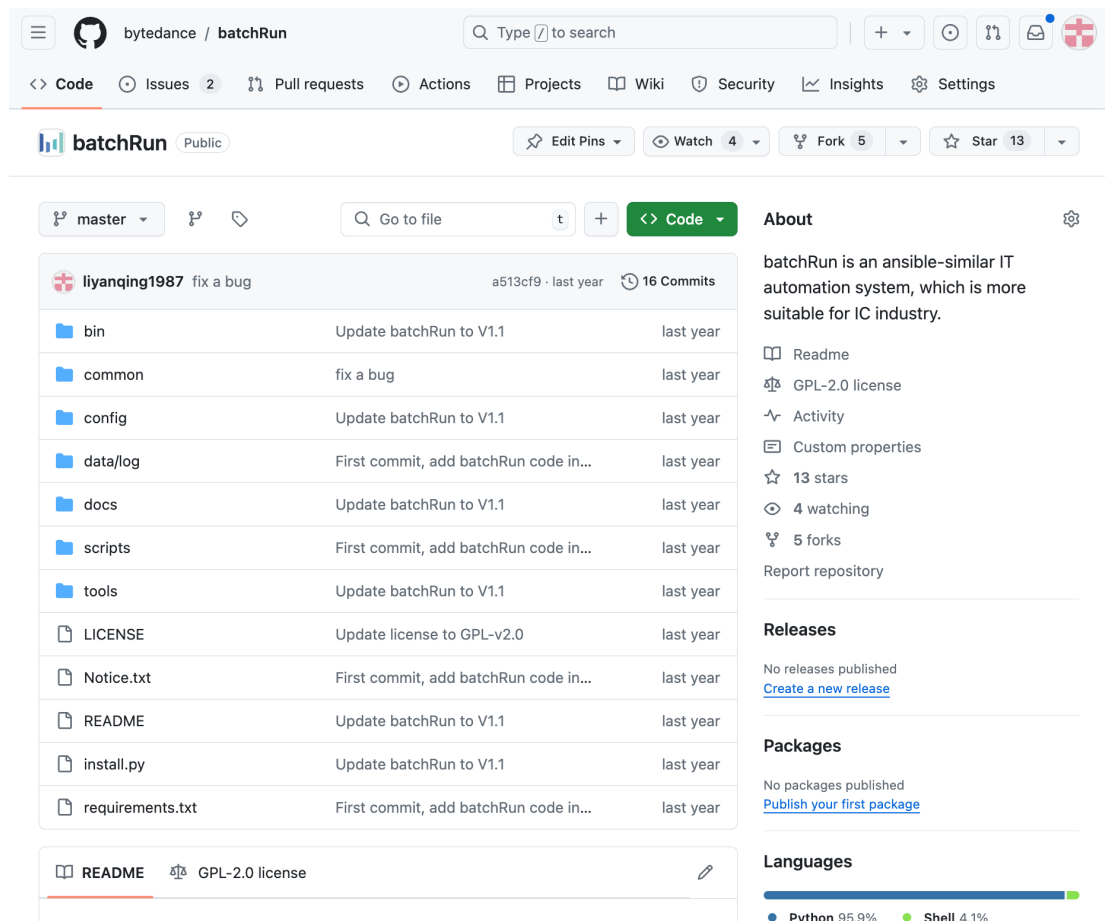
batchRun 基于 python 开发，其开发和测试的 python 版本为 **python3.8.8**。

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

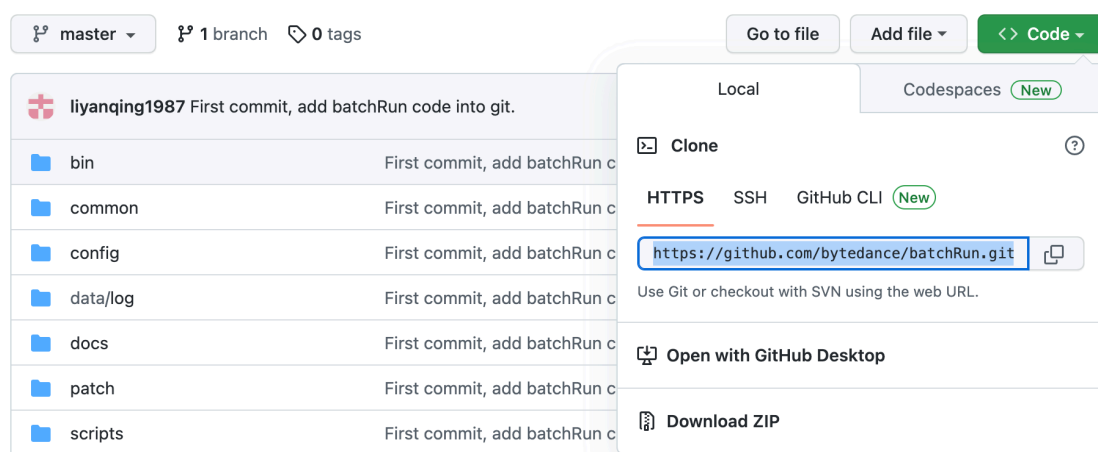
batchRun 的 github 路径位于 <https://github.com/bytedance/batchRun>



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

```
bytedance@C02FT5LHMD6R Downloads % git clone
https://github.com/bytedance/batchRun.git
Cloning into 'batchRun'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 32 (delta 1), reused 32 (delta 1), pack-reused 0
Receiving objects: 100% (32/32), 800.59 KiB | 1.20 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

也可以在 batchRun 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。

batchRun 是 IT 管理员的运维自动化工具，所以建议用 root 账号将安装包拷贝到安装目录（NAS 最佳），并确保当前服务器有权限 ssh 其它被管理机器。

安装包下的文件和目录如下。

```
[root@ic-admin2 batchRun]# ls
bin common config db docs install.py LICENSE Notice.txt
README.md requirements.txt scripts tools
```

确认 python 版本正确，并基于安装包中的 requirements.txt 安装 python 依赖库。（可能需要 root 权限）

```
[root@ic-admin2 batchRun]# which python3
/ic/software/tools/python3/3.8.8/bin/python3
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# pip3 install -r requirements.txt
Looking in indexes: https://bytedpypi.byted.org/simple/
Requirement already satisfied: pexpect==4.8.0 in
/ic/software/tools/python3/3.8.8/lib/python3.8/site-packages (from
-r requirements.txt (line 1)) (4.8.0)
Requirement already satisfied: ptyprocess>=0.5 in
/ic/software/tools/python3/3.8.8/lib/python3.8/site-packages (from
pexpect==4.8.0->-r requirements.txt (line 1)) (0.7.0)
```

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

```
[root@ic-admin2 batchRun]# python3 install.py
>>> Check python version.
    Required python version : (3, 8)
    Current python version : (3, 8)

>>> Generate script "/ic/software/tools/batchRun/bin/batch_run".
>>> Generate script
"/ic/software/tools/batchRun/tools/encrypt_python".
>>> Generate script "/ic/software/tools/batchRun/tools/patch".
>>> Generate script
"/ic/software/tools/batchRun/tools/sample_host_info".
>>> Generate script
"/ic/software/tools/batchRun/tools/save_password".
>>> Generate script
"/ic/software/tools/batchRun/tools/switch_etc_hosts".
>>> Generate config file
"/ic/software/tools/batchRun/config/config.py".

Done, Please enjoy it.
```

3.3 工具配置

主要的配置文件位于 安装目录下的 config 目录中。

3.3.1 config/config.py

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。

安装后默认配置如下，一般采用默认设置即可。

```
# Specify host list, default is "host.list" on current configure
directory.
host_list = '/ic/software/tools/batchRun/config/host.list'

# Specify the database directory.
db_path = '/ic/software/tools/batchRun/db'

# Default ssh command.
```

```
default_ssh_command = "ssh -o StrictHostKeyChecking=no -t -q"

# Support host_ip fuzzy matching, could be "True" or "False".
fuzzy_match = True

# Define timeout for ssh command, unit is "second".
timeout = 10
```

host_list：指定 host.list 文件位置，一般使用默认值即可。

db_path：指定 batch_run 的数据库路径。

default_ssh_command：指定默认 ssh 的命令，一般使用默认值即可。

fuzzy_match：控制命令参数中 host 模糊匹配的开关，默认打开。

timeout：指定 ssh 命令的超时时间，默认为 10 秒，如果经常执行 long-runtime 的任务，可以适当调大这个值。

3.3.2 config/host.list

host.list 用于定义所有的机器及其分组信息，下面是一个说明。

```
##### Format #####
# [group]
# host_ip1
# host_ip2 ssh_port=<ssh_port2>
# host_ip3 ssh_host=<host_name3>
# host_ip4 ssh_host=<host_name4> ssh_port=<ssh_port4>
# sub_group5
# ~host_ip6
# ~host_name7
# ~sub_group8
#####
```

host.list 配置的基本格式如下：

[group] #组名

host #机器

sub_group #子组

excluded_host #要排除的机器

excluded_sub_group #要排除的子组

3.3.2.1 编写规则

- “group”编写规则

group 为一组机器的组名，一般用大写，不能包含空格，在 host.list 中用“[]”括起来，比如[LOGIN]或者[COMPUTING]。

- “host”编写规则

host 为机器，包含如下 4 种写法。

```
# host_ip1
# host_ip2 ssh_port=<ssh_port2>
# host_ip3 ssh_host=<host_name3>
# host_ip4 ssh_host=<host_name4> ssh_port=<ssh_port4>
```

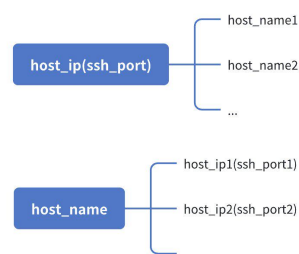
host_ip 是必选项，是 host 的确定标识。

host_name 是可选项，主要用于标识 DNS 映射关系。

ssh_port 是可选项，如果 host_ip 的 ssh 端口不是 22，则需要在此处指定。

此处 host_name 需要注意，不能包含空格。

为保持和/etc/hosts 的兼容，host_ip 和 host_name 是多对多的对应关系，即一个 host_ip 可能对应多个 host_name（但是对应唯一的 ssh_port），一个 host_name 也可能对应多个 host_ip，对应关系如下。



- "sub_group"编写规则

sub_group 为多机器的组名，不能包含空格，必须在 host.list 文件中被定义过。

- “excluded_host”编写规则

可以按照 host_ip 或者 host_name 来排除 host，在前面加上波浪线“~”即可，样式如下。

```
[group]
~host_ip6
~host_name7
```

~host_ip2: [group]中把所有的项按照 host_ip 的维度展开，如果存在 host_ip2，则会被排除。

~host_name3: [group]中把所有的项按照 host_ip 的维度展开，如果某个 host_ip 的机器名为 host_name3，则会被排除。

- “excluded_sub_group”编写规则

可以按照 sub_group 来排除 host，在前面加上波浪线“~”即可，样式如下。

```
[group]
~sub_group8
```

~sub_group4: [group]中把所有的项按照 host_ip 的维度展开，sub_group4 中的 host_ip 会被排除掉。

3.3.2.2 列表生成

config/host.list 可以手工配置，但是在服务器数量较多时较为繁琐，所以更推荐基于格式化的/etc/hosts 文件直接生成，转换方法参照#5.1 switch_etc_hosts 章节。

```
[root@ic-admin2 batchRun]# tools/switch_etc_hosts -i /etc/hosts -o
config/host.list
```

Output File : config/host.list

```
[root@ic-admin2 batchRun]# cat config/host.list
```

```
[LOGIN_IC]
10.232.134.66  ssh_host=n232-134-066
```

...

3.3.3 config/password.encrypted

这个文件由自带工具 `tools/save_password` 生成，用户可以为自己配置加密密码，详情可以参照#5.2 章节。

建议通过 `save_password` 将 `batch_run` 常用用户的密码都存放进去，当前用户已经配置了 `ssh` 免密登录则无需采用这种加密密码登录的方式。

```
[root@ic-admin2 batchRun]# tools/save_password -p ***
[root@ic-admin2 batchRun]# cat db/password/root
root default 1724049499 ed5fa5a1d09c8f4bf3e99c55708be23a
00b2a6273de7bc43a83f9bc352822827
```

四、工具使用

4.1 工具载入

batchRun 的主程序是 batch_run，位于 batchRun 安装目录下的 bin/batch_run，安装后可以直接引用。

如果配置了 modules，则可以通过 module load 的方式引用。

```
[root@ic-admin2 batchRun]# module load cad
[root@ic-admin2 batchRun]# which batch_run
/ic/software/tools/batchRun/bin/batch_run
```

4.2 帮助信息

batch_run 的帮助信息如下。

```
[root@ic-admin2 batchRun]# batch_run -h
usage: batch_run.py [-h] [-H HOSTS [HOSTS ...]] [-G GROUPS
[GROUPS ...]] [-L] [-u USER] [-p PASSWORD] [-c COMMAND
[COMMAND ...]] [-s SCRIPT [SCRIPT ...]] [-P] [-t TIMEOUT] [-l
{0,1,2,3,4}]
                        [-o OUTPUT_FILE] [-v]

optional arguments:
  -h, --help            show this help message and exit
  -H HOSTS [HOSTS ...], --hosts HOSTS [HOSTS ...]
                        Specify host(s) with below format:
                        <host_ip>
                        <host_ip>:<ssh_port>
                        <host_name>
                        <host_name>:<ssh_port>
                        ~<host_ip>
                        ~<host_name>

                        "all | ALL" means all hosts on
                        /ic/software/cad_tools/it/batchRun/config/host.list.
                        "~<host>" means exclud specified host.
  -G GROUPS [GROUPS ...], --groups GROUPS [GROUPS ...]
                        Specify host group(s) with below format:
                        <GROUP>
```

```

~<GROUP>

"all | ALL" means all groups on
/ic/software/cad_tools/it/batchRun/config/host.list.
"~<GROUP>" means exclud hosts on specified
group.
-L, --list          List specified hosts/groups.
-u USER, --user USER  Specify the user name when login into the
host in ssh format.
-p PASSWORD, --password PASSWORD
                    Specify the user password when logging
into the host in ssh format.
-c COMMAND [COMMAND ...], --command COMMAND [COMMAND ...]
                    Specify command you want run on specified
host(s).
-s SCRIPT [SCRIPT ...], --script SCRIPT [SCRIPT ...]
                    Specify script you want run on specified
host(s), will copy script to host /tmp directory.
-P, --parallel      Run command/script parallel on specified
host(s), default is in serial.
-t TIMEOUT, --timeout TIMEOUT
                    Specify ssh timeout, default is 10
seconds.
-l {0,1,2,3,4}, --output_message_level {0,1,2,3,4}
                    Specify output message level, default is
"3".

"0" : print host info;
"1" : print output message;
"2" : print host info and the first line
of the output message;
"3" : print host info and complete output
message;
"4" : print verbose information.
-o OUTPUT_FILE, --output_file OUTPUT_FILE
                    Export output message of command/script to
specified file instead of on the screen.
-v, --version      Show batchRun version information.

```

--help : 打印帮助信息。

--hosts : 指定机器列表，可以是机器 ip，也可以是机器名。如果是“ALL”，则意味着配置文件 host.list 中所有的机器。

--groups : 根据服务器组获取服务器列表。如果是“ALL”，则意味着配置文件 host.list

中所有的机器。

--list : 列出指定的 hosts 或者 groups。

--user : 指定 ssh 登录远程机器时的用户，默认是当前用户。

--password : 指定 ssh 登录远程机器时的用户密码。如果配置了 ssh 免密登录，或者加密密码已经被保存到了 password.encrypted 中，此处无需指定。

--command : 到远程机器上要执行的命令。

--script : 到远程机器上要执行的脚本，执行方式跟 command 一样，但是脚本本身会被拷贝到远程机器上的/tmp 目录下再执行。

--parallel : 指定并行运行模式，默认是串行。如果希望获取有规律的输出信息，建议串行。

--timeout : 指定 ssh 的超时时间，默认为 10 秒。

--output_message_level : 指定执行命令时输出信息的详细程度，分为 0-4 共五个等级。

“0”：只打印机器名。

“1”：只打印（执行命令的）输出信息。

“2”：打印机器名和（执行命令的）输出信息的第一行。

“3”：打印机器名和完成的（执行命令的）输出信息。

“4”：在#3 的基础上打印更详细的 debug 信息（ssh 命令）。

--output_file : 指定输出文件，如指定，会将输出在屏幕上的（执行命令的）输出信息导入到指定的输出文件中。

--version : 打印 batchRun 版本信息。

4.3 功能介绍

4.3.1 打印 batchRun 版本信息。

batch_run --version 可以打印 batchRun 的版本信息，当前版本是 V1.1，是 V1.0 的小升级版本。

```
[root@ic-admin2 batchRun]# bin/batch_run --version
Version : 1.2
Release : 2024.8.23
```

4.3.2 列出指定机器

如果不清楚 batch_run 预设的机器及机器组设置，可以使用“batch_run --list --groups ALL”查看。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --groups ALL
GROUP : [LOGIN_IC]
        10.232.134.66      n232-134-066
GROUP : [LOGIN_EMU]
        10.232.136.12      n232-136-012
GROUP : [CMP_IC]
        10.232.132.142     n232-132-142
        10.232.132.143     n232-132-143
GROUP : [CMP_EMU]
        10.232.136.68      n232-136-068
        10.232.136.69      n232-136-069
GROUP : [UBUNTU]
        10.232.134.196     n232-134-196
GROUP : [ROCKY8]
        10.232.132.70      n232-132-070
GROUP : [LIC]
        10.232.158.3       ic-lic01
GROUP : [VM]
        10.232.158.39      ic-monitor01  8888
```

如果已知机器组名，也可以使用“batch_run --list --groups <GROUP>”查看指定机器组的具体信息。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --groups CMP_IC
GROUP : [CMP_IC]
        10.232.132.142     n232-132-142
        10.232.132.143     n232-132-143
```

4.3.3 指定用户名和密码登录远程机器

如果知道用户名和密码，可以采用指定用户名和密码的方式，以指定帐号的身份登录远程机器执行命令，操作方式为“batch_run --user <USER> --password <PASSWORD> ...”。

下面是一个示例，root 用指定账号密码登录机器并执行命令。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
user liyanqing.1987 --password *** --command whoami

>>> 10.232.134.66
    liyanqing.1987
```

因为明文密码会导致敏感信息泄露，所以并不推荐这种方式，建议配置 ssh 免密登录，或者采用内置加密密码的方式登录远程机器。

4.3.4 采用内置加密密码登录远程机器

使用内置工具 save_password 为指定账号保存加密密码。

```
[liyanqing.1987@n232-135-013 batchRun]$ tools/save_password --
password ***
```

其储存的密码为加密样式。

```
[liyanqing.1987@n232-135-013 batchRun]$ cat
db/password/liyanqing.1987
liyanqing.1987 default 1724392876
a2c768667257c39f4bc2a1a4fa1124dc 8d6f948e98fa7ba36cfa07f5d72afafa
```

此时，batch_run 用当前 Linux 登录账号登录远程机器时，可以直接使用这个加密密码，以防止明文导致密码泄露的问题。

```
[liyanqing.1987@n232-135-013 batchRun]$ cat
db/password/liyanqing.1987
liyanqing.1987 default 1724392876
a2c768667257c39f4bc2a1a4fa1124dc 8d6f948e98fa7ba36cfa07f5d72afafa
```

请注意，上面例子中，一定要以密码登陆的方式成为 liyanqing.1987 用户，否则 batch_run 会无法判断当前实际用户是否知悉 liyanqing.1987 账号密码（比如 root 账号 su 成 liyanqing.1987，并不需要知道 liyanqing.1987 的密码），从而导致拒绝执行命

令。

此时即便是没有配置 ssh 免密，也不输入明文密码，仍然可以方便地 ssh 地通过 batchRun 登录到远程服务器上执行任务。

```
[liyanqing.1987@n232-135-013 batchRun]$ bin/batch_run --hosts
10.232.134.66 --command whoami

>>> 10.232.134.66
    liyanqing.1987
```

4.3.5 指定机器执行命令

batch_run 支持用“batch_run --hosts <HOST> <HOST> ...”的方式指定机器，HOST 既可以是 ip 地址，也可以是 hostname（hostname 一定要在/etc/hosts 或者 host.list 中配置过）。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66
n232-136-012 --command hostname

>>> 10.232.134.66
    n232-134-066

>>> n232-136-012 (10.232.136.12)
    n232-136-012
```

同时，batch_run 的--hosts 参数也支持模糊匹配，会把所有匹配的项都列出来。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --hosts 132
[FUZZY MATCH] 132 -> 10.232.132.142
[FUZZY MATCH] 132 -> 10.232.132.143
[FUZZY MATCH] 132 -> 10.232.132.70

GROUP : [CMP_IC]
    10.232.132.142    n232-132-142
    10.232.132.143    n232-132-143
GROUP : [ROCKY8]
    10.232.132.70     n232-132-070
```

不但可以匹配 host ip, 也可以匹配 host name, 还支持^\$等正则匹配符号来界定位置。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --hosts ^n232-134
[FUZZY MATCH] ^n232-134 -> n232-134-066 -> 10.232.134.66
[FUZZY MATCH] ^n232-134 -> n232-134-196 -> 10.232.134.196

GROUP : [LOGIN_IC]
        10.232.134.66      n232-134-066
GROUP : [UBUNTU]
        10.232.134.196    n232-134-196
```

4.3.6 指定机器组执行命令

batch_run 支持用“batch_run --groups <GROUP> <GROUP> ...”的方式指定机器, GROUP 必须在 host.list 中有定义。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --groups CMP_IC
CMP_EMU
GROUP : [CMP_IC]
        10.232.132.142    n232-132-142
        10.232.132.143    n232-132-143
GROUP : [CMP_EMU]
        10.232.136.68     n232-136-068
        10.232.136.69     n232-136-069
```

也可以采用除外的方式排除掉指定的 group 或者 host。

```
[root@ic-admin2 batchRun]# bin/batch_run --list --groups CMP_IC
CMP_EMU --hosts ~10.232.132.143 ~n232-136-069
GROUP : [CMP_IC]
        10.232.132.142    n232-132-142
GROUP : [CMP_EMU]
        10.232.136.68     n232-136-068
```

4.3.7 执行命令 (command)

batch_run 可以在远程机器上执行命令，此处的命令既可以表示系统命令（比如 hostname），也可以表示脚本命令（比如 ~/.run.sh），前提是远程机器上命令是可以直接访问和执行的。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups CMP_IC --command
hostname -output_message_level 2
>>> 10.232.132.142      n232-132-142
>>> 10.232.132.143      n232-132-143
```

4.3.8 执行脚本 (script)

script 跟 command 的主要区别在于，我们认为 command 在远程机器上是一定存在且可执行的，但是 script 则未必（比如 script 是本地脚本），因此需要拷贝到远程机器上再执行。

下面是一个本地脚本，只在本机存在。

```
[root@ic-admin2 batchRun]# cat ~/local_run.sh
#!/bin/bash

echo "I am `whoami`"
echo "This host is `hostname`"
```

如果想在远程机器上执行 local_run.sh，只能将它拷贝到远程机器上再执行。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.132.142 --
script ~/local_run.sh -output_message_level 4

>>> 10.232.132.142
    scp /root/local_run.sh 10.232.132.142:/tmp/local_run.sh
    ssh -o StrictHostKeyChecking=no -t -q root@10.232.132.142
/tmp/local_run.sh
    ==== output ====
    I am root
    This host is n232-132-142
    =====
```

4.3.9 串行执行和并行执行

如未指定--parallel 参数，batch_run 的任务默认是串行执行。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups LOGIN_IC CMP_IC
--command 'hostname; touch
/ic/software/tools/batchRun/db/test/`hostname`' --
output_message_level 2
>>> 10.232.134.66      n232-134-066
>>> 10.232.132.142    n232-132-142
>>> 10.232.132.143    n232-132-143
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# ls /ic/software/tools/batchRun/db/test
n232-132-142  n232-132-143  n232-134-066
```

我们看到，脚本在指定机器上串行执行，明确打印 output 信息，并生成了对应文件。

下面我们删掉生成的文件，添加--parallel 参数用并行方式再执行一遍。（此时 output_message_level 被强制为“0”）

```
[root@ic-admin2 batchRun]# rm -f
/ic/software/tools/batchRun/db/test/*
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# bin/batch_run --groups LOGIN_IC CMP_IC
--command 'hostname; touch
/ic/software/tools/batchRun/db/test/`hostname`' --parallel
*Warning*: Switch output_message_level to "0" on parallel mode.
>>> 10.232.134.66
>>> 10.232.132.142
>>> 10.232.132.143
[root@ic-admin2 batchRun]#
[root@ic-admin2 batchRun]# ls /ic/software/tools/batchRun/db/test
n232-132-142  n232-132-143  n232-134-066
```

我们看到文件正常生成，说明脚本在每台机器上都正常执行了。

4.3.10 超时时间 (timeout)

batch_run 底层基于 ssh 登陆远程服务器执行命令，其默认的 timeout 为 10 秒，如果任务的执行时间超过 10 秒，ssh 会报告超时并强制断开连接。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups LOGIN_IC --
command "sleep 15"

>>> 10.232.134.66
    Timeout exceeded.
    ...
    0: re.compile(b'assword:')
    1: EOF
```

执行运行时间超过 10 秒的任务时，可以通过设置更长的 timeout 时间来规避这个问题。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups LOGIN_IC --
command "sleep 15" --timeout 20

>>> 10.232.134.66
[root@ic-admin2 batchRun]#
```

4.3.11 输出信息层级

batch_run 的 output message 共有 5 个层级可选，效果分别如下。

- “0”级输出，仅打印服务器名。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 0
>>> 10.232.134.66
```

- “1”级输出，仅打印执行命令的输出信息，主要是配置--output_file 参数使用。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 1
    total          used          free          shared  buff/cache
available
```

Mem:	1007	172	2	2
832	830			
Swap:	127	61	66	

- “2”级输出，打印服务器名，并显示执行命令的输出信息的第一行。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 2
>>> 10.232.134.66      total      used      free      shared
buff/cache  available
```

- “3”级输出，默认级别，打印服务器名，并显示执行命令的输出信息的全部。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 3

>>> 10.232.134.66
      total      used      free      shared  buff/cache
available
      Mem:      1007      172      2      2
832      830
      Swap:      127      61      66
```

- “4”级输出，在#3的基础上，会打印实际执行的 ssh 命令，以方便用户手工执行并复现结果。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 4

>>> 10.232.134.66
      ssh -o StrictHostKeyChecking=no -t -q root@10.232.134.66 free
      \-g
      ==== output ====
      total      used      free      shared  buff/cache
available
      Mem:      1007      172      2      2
832      830
      Swap:      127      61      66
      =====
```

4.3.12 输出文件

batch_run 的任务执行结果默认输出到屏幕上，但是也可以通过--output_file 参数将其导入指定文件中。

```
[root@ic-admin2 batchRun]# bin/batch_run --hosts 10.232.134.66 --
command "free -g" --output_message_level 1 --output_file
10.232.134.66.mem
[root@ic-admin2 batchRun]# cat 10.232.134.66.mem
```

	total	used	free	shared	buff/cache
available					
Mem:		1007	171	3	2
832	831				
Swap:		127	61	66	

batch_run 还支持自动将--output_file 中的“HOST”字符自动转换为实际的 host 名，这样非常利于多服务器信息采样的信息保存。

```
[root@ic-admin2 batchRun]# bin/batch_run --groups LOGIN_IC CMP_IC
--command "free -g" --output_message_level 1 --output_file
db/host_info/HOST.mem
[root@ic-admin2 batchRun]# ls db/host_info/
10.232.132.142.mem 10.232.132.143.mem 10.232.134.66.mem
[root@ic-admin2 batchRun]# cat db/host_info/10.232.134.66.mem
```

	total	used	free	shared	buff/cache
available					
Mem:		1007	171	3	2
832	831				
Swap:		127	61	66	

4.4 日志功能

出于安全的考虑，batch_run 会自动为所有的操作保留日志记录。

如下是一个示例，获取当前时间，然后执行一条指令。

```
[root@ic-admin2 batchRun]# date +%Y%m%d_%H%M%S; bin/batch_run --
groups CMP_IC --command hostname --output_message_level 2
20240823_162536
>>> 10.232.132.142      n232-132-142
```

```
>>> 10.232.132.143      n232-132-143
```

在日志文件<db_path>/log/<user>/<time_stamp>中会保留当次执行记录。

```
[root@ic-admin2 batchRun]# cat db/log/root/20240823_162536
{"time": "2024-08-23 16:25:36", "user": "root", "login_user":
"root", "command": "/ic/software/tools/batchRun/bin/batch_run.py -
-groups CMP_IC --command hostname --output_message_level 2"}
```

```
>>> 10.232.132.142      n232-132-142
```

```
>>> 10.232.132.143      n232-132-143
```

执行日志的第一行是 json 格式的 python dict, 主要保存了如下信息:

- batch_run 执行时间。
- batch_run 的执行用户。(whomai 获取)
- batch_run 的执行用户对应的登录用户。(who am i 获取)
- batch_run 执行命令。

下方则保存了当次执行所输出的信息。

通过日志检索, 我们可以追溯一些恶意/失误的操作。

五、辅助工具

batchRun 的辅助工具位于 tools 目录下。

5.1 tools/switch_etc_hosts

switch_etc_hosts 用于将格式化的/etc/hosts 文件转换为 batchRun 的 host.list 文件，用法如下。

```
[root@ic-admin2 batchRun]# tools/switch_etc_hosts -h
usage: switch_etc_hosts.py [-h] [-e EXPECTED_IPS
[EXPECTED_IPS ...]] [-E EXCLUDED_IPS [EXCLUDED_IPS ...]] [-i
INPUT_FILE] [-o OUTPUT_FILE] [-a APPEND] [-r] [-t
{batchRun,ansible}]

optional arguments:
  -h, --help                show this help message and exit
  -e EXPECTED_IPS [EXPECTED_IPS ...], --expected_ips EXPECTED_IPS
[EXPECTED_IPS ...]          Specify expected ip(s), support regular
expressions.
  -E EXCLUDED_IPS [EXCLUDED_IPS ...], --excluded_ips EXCLUDED_IPS
[EXCLUDED_IPS ...]          Specify excluded ip(s), support regular
expressions.
  -i INPUT_FILE, --input_file INPUT_FILE
                             Specify input file, default is
"/etc/hosts".
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                             Specify output file, default is
"./host.list".
  -a APPEND, --append APPEND
                             Append configuration file into output
file.
  -r, --rewrite              Rewrite mode, rewrite output file by
force.
  -t {batchRun,ansible}, --tool {batchRun,ansible}
                             Which tool the host list is for, default
is "batchRun".
```

--expected_ips: 指定期望的 ip，也可以用正则匹配的方式指定 ip 段（10.212.206.

*)，如指定，那么 hosts 文件中不在指定列表中 ip 会被抛弃。

--excluded_ips：指定排除的 ip，如指定，那么 hosts 文件中在指定列表中的 ip 会被抛弃。

--input_file：指定输入 hosts 文件来源，默认是“/etc/hosts”。

--output_file：指定输出 host.list 文件路径，默认是“./host.list”。

--append：指定追加文件，会以文本的形式拼接在 output_file 结尾，一般用于指定组嵌套关系。

--rewrite：强制覆盖模式，如指定，output_file 在已存在的情况下会被强制覆盖。

--tool：指定输出 host.list 的格式，默认是 batchRun 所支持的格式，也可以选 ansible 所支持的格式。

/etc/hosts 文件需要符合如下格式才能被正确解析。

```
# GROUP : <group>
<host_ip> <host_name>
<host_ip> <host_name> # SSH_PORT=<port>
```

下面是一个示例 hosts 文件。

```
[root@ic-admin2 batchRun]# cat ~/test/hosts
# GROUP : ETX_IC
10.232.134.66  n232-134-066

# GROUP : ETX_EMU
10.232.136.12  n232-136-012

# GROUP : CMP_IC
10.232.132.142 n232-132-142
10.232.132.143 n232-132-143

# GROUP : CMP_EMU
10.232.136.68  n232-136-068
10.232.136.69  n232-136-069

# GROUP : UBUNTU
10.232.134.196 n232-134-196

# GROUP : ROCKY8
```

```
10.232.132.70    n232-132-070

# GROUP : LIC
10.232.158.3    ic-lic01

# GROUP : VM
10.232.158.39   ic-monitor01 # SSH_PORT=8888
```

我们将如上 hosts 文件转换为 batchRun 的 host.list 文件。(也支持转换为 ansible 格式的 hosts 文件)

```
[root@ic-admin2 batchRun]# tools/switch_etc_hosts --input_file
~/test/hosts --output_file config/host.list -t batchRun

Output File : config/host.list
```

转换后的 batchRun host.list 文件如下。

```
[ETX_IC]
10.232.134.66    ssh_host=n232-134-066

[ETX_EMU]
10.232.136.12    ssh_host=n232-136-012

[CMF_IC]
10.232.132.142    ssh_host=n232-132-142
10.232.132.143    ssh_host=n232-132-143

[CMF_EMU]
10.232.136.68    ssh_host=n232-136-068
10.232.136.69    ssh_host=n232-136-069

[UBUNTU]
10.232.134.196    ssh_host=n232-134-196

[ROCKY8]
10.232.132.70    ssh_host=n232-132-070

[LIC]
10.232.158.3     ssh_host=ic-lic01
```

```
[VM]
10.232.158.39  ssh_host=ic-monitor01  ssh_port=8888
```

因为组定义支持嵌套，所以我们可以预先定义一些组嵌套，并以 `append` 的方式追加到最终的 `host.list` 文件中。

```
[root@ic-admin2 batchRun]# cat ~/test/host.list.append

[IC]
LOGIN_IC
CMP_IC

[EMU]
LOGIN_EMU
CMP_EMU
[root@ic-admin2 batchRun]# tools/switch_etc_hosts --input_file
~/test/hosts --output_file config/host.list --append
~/test/host.list.append

Output File : config/host.list
[root@ic-admin2 batchRun]# tail -n 10 config/host.list
[VM]
10.232.158.39  ssh_host=ic-monitor01  ssh_port=8888

[IC]
LOGIN_IC
CMP_IC

[EMU]
LOGIN_EMU
CMP_EMU
```

5.2 save_password

`save_password` 用于将当前用户密码加密保存。

```
[root@ic-admin2 batchRun]# tools/save_password -h
usage: save_password.py [-h] -p PASSWORD [-H HOST] [-o
OUTPUT_FILE]

optional arguments:
```

```
-h, --help          show this help message and exit
-p PASSWORD, --password PASSWORD
                    Specify user password.
-H HOST, --host HOST Specify the host which user password works
on, default is "default".
-o OUTPUT_FILE, --output_file OUTPUT_FILE
                    Specify the output file, default is
"<db_path>/password/<user>".
```

--password_file: 用于指定保存账号密码信息的文件，一般无需指定，采用默认值即可。

--host: 指定账号密码生效的机器，默认为所有机器。

--password: 指定用户明文密码。

下面是一个例子。

```
[root@ic-admin2 batchRun]# tools/save_password --password ***
[root@ic-admin2 batchRun]# cat db/password/root
root default 1724049499 ed5fa5a1d09c8f4bf3e99c55708be23a
00b2a6273de7bc43a83f9bc352822827
```

用户密码被加密保存后，在工具引用时可以直接读取，无需明文输入，从而保证了密码安全性。

5.3 sample_host_info

sample_host_info 用于批量采集多服务器的 OS 和 hardware 信息，并保存为 json 文件备用。

其底层是借助了 batch_run 的 --output_file 功能，所以会接受一些 batch_run 的基本参数并传递给 batch_run 执行，然后将 batch_run 的输出信息整理后重新保存。

```
[root@ic-admin2 batchRun]# tools/sample_host_info -h
usage: sample_host_info.py [-h] [-H HOSTS [HOSTS ...]] [-G GROUPS
[GROUPS ...]] [-u USER] [-p PASSWORD] [-o OUTPUT_DIR]

optional arguments:
```

```
-h, --help          show this help message and exit
-H HOSTS [HOSTS ...], --hosts HOSTS [HOSTS ...]
                    Specify the host(s) for batch_run.
-G GROUPS [GROUPS ...], --groups GROUPS [GROUPS ...]
                    Specify host group(s) for batch_run.
-u USER, --user USER Specify the user name when connectting
host as.
-p PASSWORD, --password PASSWORD
                    Specify the user password when connectting
host with.
-o OUTPUT_DIR, --output_dir OUTPUT_DIR
                    Where to save temporary file and
host_info.json, default is current directory.
```

--host: 指定 host, “ALL”意味着 host.list 中所有的服务器。

--groups: 指定 group, “ALL”意味着 host.list 中所有的服务器, 于--host 参数二选一设置即可。

--user: 指定 user, 非必须。

--password: 指定 password, 非必须。

--output_dir: 指定输出路径, 默认为当前路径。

下面是一个例子。

```
[root@ic-admin2 batchRun]# tools/sample_host_info --hosts ALL -o
db/host_info
>>> Clean up db/host_info ...
    rm -rf *.info host_list.json host_info.json
>>> Sampling host list information ...
    /ic/software/tools/batchRun/bin/batch_run -H ALL -L -o
db/host_info/host_list.json
* Host(s) info has been saved into "db/host_info/host_list.json".
>>> Sampling host os/cpu/mem information ...
    /ic/software/tools/batchRun/bin/batch_run -H ALL -c
"lsb_release -a; lscpu; free -g" -P -l 1 -o db/host_info/HOST.info
>>> Collecting host information ...
    Host info has been saved to file
"db/host_info/host_info.json".
```

生成的 host_info.json 文件内容如下。

```
{
  "10.232.136.69": {
    "groups": [
      "CMP_EMU"
    ],
    "host_ip": "10.232.136.69",
    "host_name": "n232-136-069",
    "server_type": "physical",
    "os": "CentOS Linux release 7.9.2009 (Core)",
    "cpu_architecture": "x86_64",
    "cpu_thread": 64,
    "thread_per_core": 1,
    "cpu_model": "Intel(R) Xeon(R) Platinum 8336C CPU @
2.30GHz",
    "cpu_frequency": 3.0,
    "cpu_frequency_unit": "GHz",
    "mem_size": 2015,
    "mem_size_unit": "GB",
    "swap_size": 1023,
    "swap_size_unit": "GB"
  },
  ...
}
```

5.4 patch

patch 是帮助 batchRun 打补丁的工具，其帮助信息如下。

```
[root@ic-admin2 batchRun]# tools/patch -h
usage: patch.py [-h] [-p PATCH_PATH]

optional arguments:
  -h, --help            show this help message and exit
  -p PATCH_PATH, --patch_path PATCH_PATH
                        Specify patch path (new install package
path).
```

--patch_path: 指定补丁包（也就是新的安装包）路径。

一般而言，batchRun 的版本变更，主要是新增 python 脚本，或者是现有 python 脚本内容变更，针对这种变更的安装包，patch 可以自动将新的安装包变更更新到当前安装包。

下面是一个示例。

```
[root@ic-admin2 batchRun]# tools/patch -p ~/batchRun-master
Install Path : /ic/data/usr/liyanqing.1987/tools/batchRun
Patch path : /home/liyanqing.1987/batchRun-master

*Warning*: current install path name is "batchRun", but patch path
name is "batchRun-master".
Do you want to continue? (y|n) y

> Copying python file "/home/liyanqing.1987/batchRun-
master/bin/batch_run.py" into
"/ic/data/usr/liyanqing.1987/tools/batchRun/bin/batch_run.py".
> Copying python file "/home/liyanqing.1987/batchRun-
master/common/common.py" into
"/ic/data/usr/liyanqing.1987/tools/batchRun/common/common.py".
> Copying python file "/home/liyanqing.1987/batchRun-
master/tools/encrypt_python.py" into
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/encrypt_python.p
y"
```


六、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/licenseMonitor”，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
2022.12	1.0	发布第一个正式 release 版本
2023.7	1.1	增加对 host_ip 和 host_name 的多对多映射关系的支持。 去除对 LSF queue 机器获取的支持。
2024.8	1.2	增加服务器信息采样功能。