

batchRun 用户手册

Product Name : batchRun

Product Version : V1.1

Release Date : 2023.7.17

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

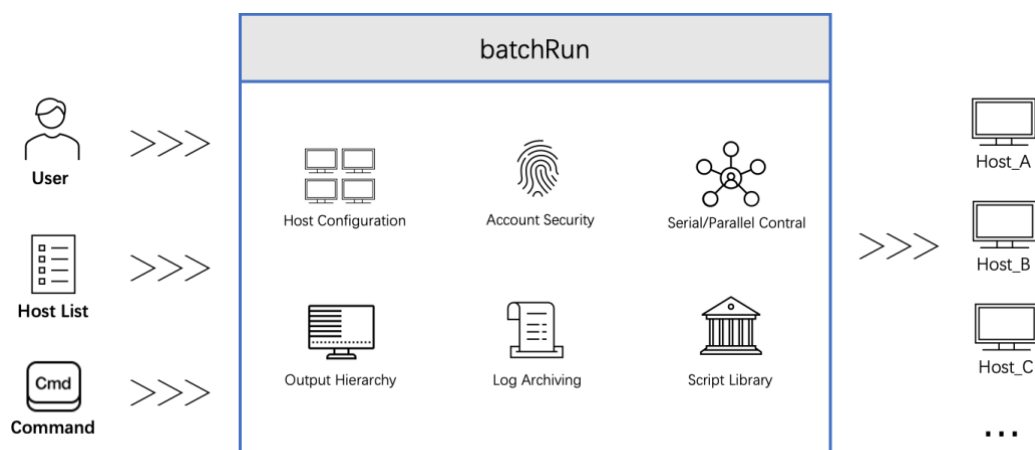
@张静文 (zhangjingwen.silvia@bytedance.com)

目录

一、简介.....	3
二、环境依赖.....	6
2.1 操作系统依赖.....	6
2.2 PYTHON 版本依赖.....	6
2.3 共享存储.....	6
三、工具安装及配置.....	7
3.1 工具下载.....	7
3.2 工具安装.....	8
3.3 工具配置.....	9
3.3.1 config.py.....	9
3.3.2 host.list.....	10
3.3.3 password.encrypted.....	13
四、工具使用.....	14
4.1 工具载入.....	14
4.2 帮助信息.....	14
4.3 功能介绍.....	16
4.3.1 打印 batchRun 版本信息。.....	16
4.3.2 列出预设机器.....	16
4.3.3 指定用户名和密码登录远程机器.....	18
4.3.4 采用内置加密密码登录远程机器.....	18
4.3.5 指定机器执行命令.....	19
4.3.6 指定机器组执行命令.....	19
4.3.7 机器模糊匹配.....	20
4.3.8 串行执行和并行执行.....	20
4.3.9 执行多行命令.....	22
4.3.10 批量配置 ssh 免密.....	22
4.3.11 输出信息层级.....	24
五、辅助工具.....	26
5.1 SAVE_PASSWORD.....	26
5.2 GET_PASSWORD.....	27
5.3 ESSH.....	28
5.4 XSSH.....	28
5.5 SWITCH_ETC_HOSTS.....	30
5.6 PATCH.....	32
六、自带脚本库.....	33
七、技术支持.....	35
附录.....	36
附 1. 变更历史.....	36

一、简介

batchRun 是 ByteDance 开源的一款适用于集成电路设计 Linux 环境的 IT 运维自动化工具，基于 python 语言开发，底层基于 ssh 实现多服务器登录管理，分布式操作，无需客户端。



工具的主要输入包括三类：

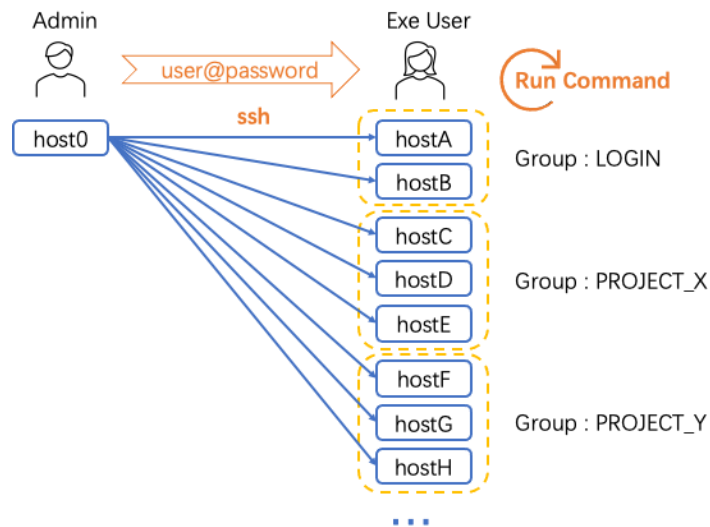
Account/Password：账号密码信息。如果是使用自带的加密账号密码系统，或者 ssh 免密登录，则无需明文指定。

Host List: 执行机器列表。

Command/Script：要执行的命令或者脚本。可以是 shell 命令，也可以是任何编程语言的脚本和工具。

在控制节点上，batchRun 基于 ssh 协议登录要操作的机器，在远程机器上执行指定的命令或脚本，并获取相应的输出信息。

其具体的执行流程如下图所示。



batchRun 的主要目标是 IC 设计环境管理辅助，包括但不限于批量配置操作系统、软件部署、系统信息获取等。所以 batchRun 并不追求 Ansible 之类工具的多平台兼容，而主要关注如何更好满足 IC 设计环境管理需求。

batchRun 的特点如下：



另外，跟 Ansible 相比，batchRun 有两个优点。

- 操作极简

IC IT 管理员的一般工作习惯是，在服务器上执行 shell 命令（或其它脚本）完成配置任务，batchRun 只需要将 命令/脚本 作为参数直接推送执行即可，而 Ansible 往往还需要用户专门配置专门语法格式的 playbook 才行。batchRun 的操作相比较而言更加简单易操作，不需要改变 IT 管理员的使用习惯。

- 易于集成

batchRun 本身就是一个脚本，在配置好 host.list 和 password.encrypted 后，对系统、环境等均无依赖（亦不要求 ssh 免密），非常易于作为插件的形式集成到其它工具内部，用作多机器批量管理和信息获取的手段。

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。

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

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。

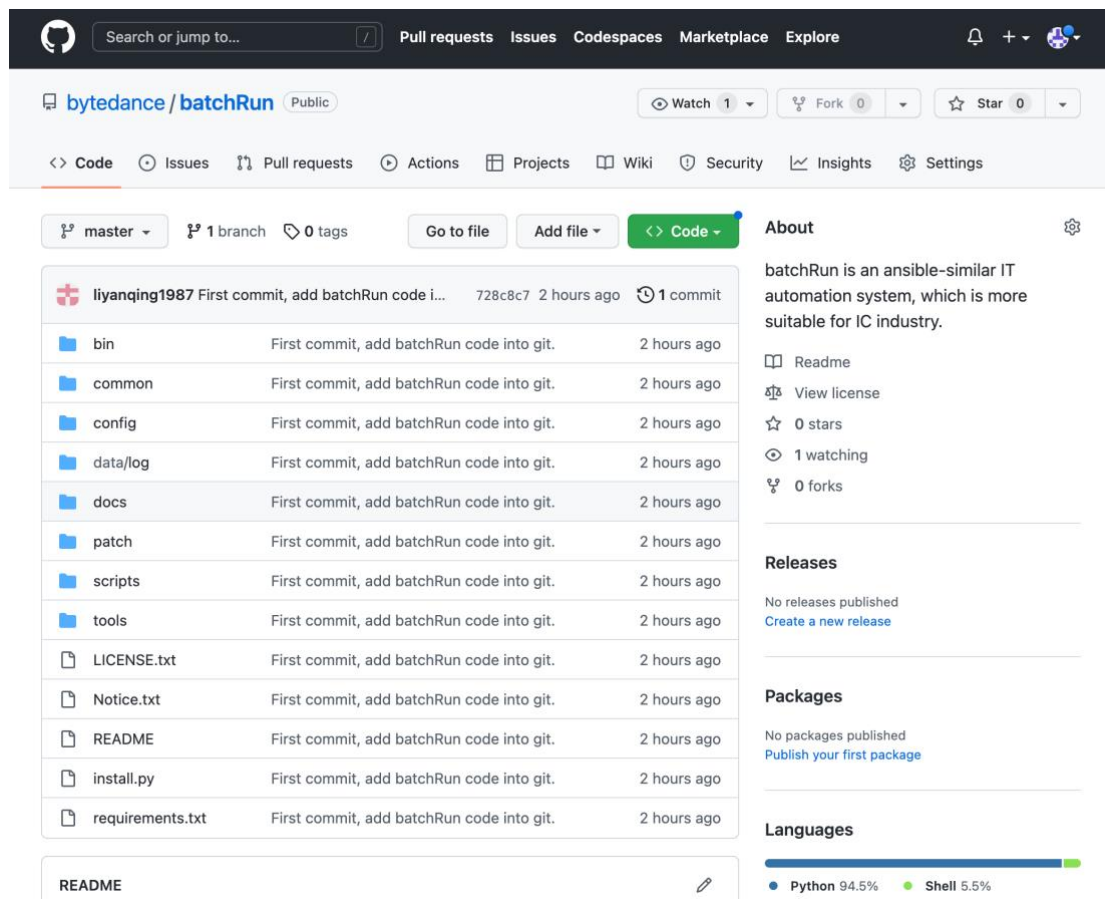
2.3 共享存储

batchRun 本身是分布式运行，并不依赖共享存储，但是 batchRun 自带脚本库，如果用 batchRun 运行自带脚本库中的脚本，为了确保所有执行机器上脚本可访问，需要将 batchRun 安装到共享存储，并在所有执行机上挂载同一个共享存储。

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>。

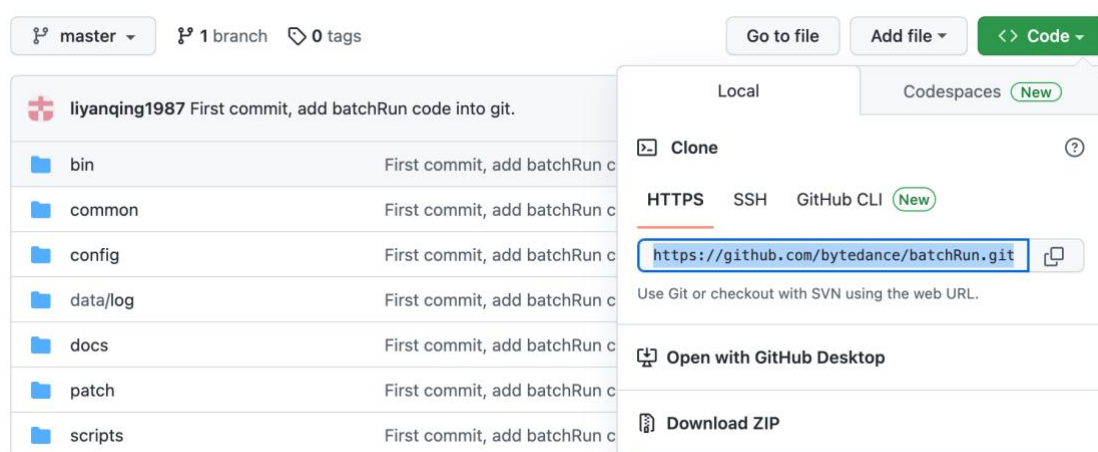


可以采用“git clone <https://github.com/bytedance/batchRun.git>”的方式拉取源代码。

Bash

```
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。

将安装包拷贝到安装目录，并给与合适的目录名。

安装包下的文件和目录如下。

```
Bash
[root@ic-admin1 tools]# cd batchRun/
[root@ic-admin1 batchRun]# ls
bin common config data docs install.py LICENSE Notice.txt
README 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)

```
Bash
[root@ic-admin1 batchRun]# which python3
/ic/software/tools/python3/3.8.8/bin/python3
[root@ic-admin1 batchRun]#
[root@ic-admin1 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。

```
Bash
[root@ic-admin1 batchRun]# python3 install.py
[liyanqing.1987@n232-135-066 batchRun]$ python3 install.py
>>> Check python version.
    Required python version : (3, 8)
    Current  python version : (3, 8)

>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/bin/batch_run".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/encrypt_python".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/get_password".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/patch".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/save_password".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/switch_etc_hosts
".
>>> Generate script
"/ic/data/usr/liyanqing.1987/tools/batchRun/tools/xssh".
>>> Generate config file
"/ic/data/usr/liyanqing.1987/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.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。

```
Bash
# Specify host list, default is "host.list" on current configure
directory.
HOST_LIST =
'/ic/data/usr/liyanqing.1987/tools/batchRun/config/host.list'

# Set log directory.
LOG_DIR = '/ic/data/usr/liyanqing.1987/tools/batchRun/data/log'

# Default ssh command.
DEFAULT_SSH_COMMAND = "ssh -o StrictHostKeyChecking=no"

# 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。

LOG_DIR：指定 batch_run 执行日志文件路径。

DEFAULT_SSH_COMMAND：指定 ssh 操作的命令，一般采用推荐值即可。

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 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。

```
C#
#### Format ####
# [group]
# host_ip1
# host_ip2 <ssh_port2>
# host_ip3 <host_name3>
# host_ip4 <host_name4> <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

下面展开讲一下具体的编写规则。

3.3.2.1 “group”编写规则

group 为多机器的组名，不能包含空格，在 host.list 中用“[]”括起来。

3.3.2.2 “host”编写规则

host 为机器，包含如下 4 种写法。

```
TypeScript
host_ip1
host_ip2 <ssh_port2>
host_ip3 <host_name3>
host_ip4 <host_name4> <ssh_port4>
```

host_ip 是必选项，是 host 的确定标识。

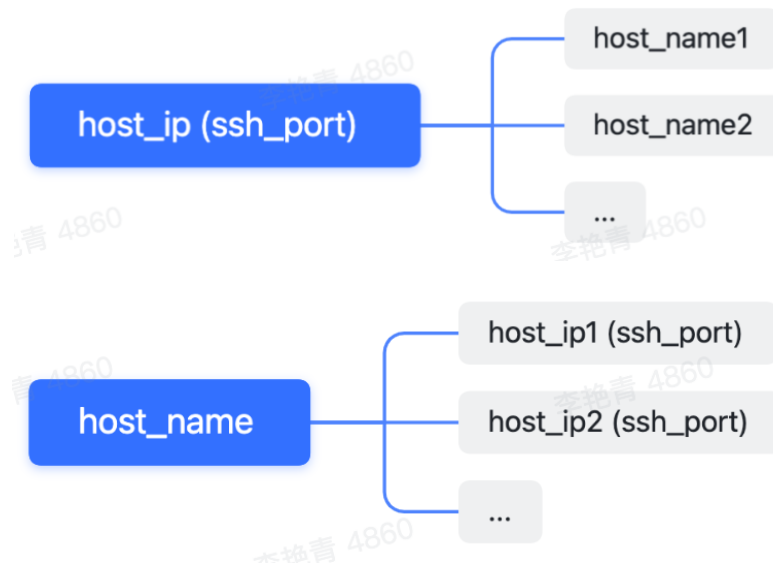
host_name 是可选项，主要用于标识 DNS 映射关系。

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。



3.3.2.3 "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（出现在“[group]”中过）。

3.3.2.4 “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。

```
TypeScript
[group]
host_ip1 host_name1 ssh_port1
sub_group
~host_ip2
~host_name3
```

~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。

3.3.2.5 “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。

```
TypeScript
[group]
host_ip1 host_name1 ssh_port1
sub_group2
sub_group3
~sub_group4
```

~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。

3.3.3 password.encrypted

这个文件由自带工具 save_password 生成，无需手工配置。

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。

```
Bash
[root@ic-admin1 batchRun]# tools/save_password -u liyanqing.1987 -
p ***
[root@ic-admin1 batchRun]# cat config/password.encrypted
liyanqing.1987 46953b705ab979d47b81b6a5ee8fa0c7
```

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。

Bash

```
[root@ic-admin1 batchRun]# module load cad
[root@ic-admin1 batchRun]# which batch_run
/ic/data/usr/liyanqing.1987/tools/batchRun/bin/batch_run
```

4.2 帮助信息

batch_run 的帮助信息如下。

Bash

```
[root@ic-admin1 batchRun]# bin/batch_run -h
usage: batch_run.py [-h] [-H HOSTS [HOSTS ...]] [-G GROUPS
[GROUPS ...]] [-P] [-u USER] [-p PASSWORD] [-c COMMAND
[COMMAND ...]] [-m MULTI_COMMANDS_FILE] [-t TIMEOUT] [-o
{0,1,2,3,4}]
                        [-l LIST_HOSTS [LIST_HOSTS ...]] [-v]
```

optional arguments:

```
-h, --help            show this help message and exit
-H HOSTS [HOSTS ...], --hosts HOSTS [HOSTS ...]
                        Specify the host(s), could be below
```

format:

```
<host_ip>
<host_ip>:<ssh_port>
<host_name>
<host_name>:<ssh_port>
~<host_ip>
~<host_name>
```

"ALL" means all hosts on

/ic/data/usr/liyanqing.1987/tools/batchRun/config/host.list.

"~<host>" means exclud specified host.

```
[root@ic-admin1 batchRun]# bin/batch_run -h
```

```

usage: batch_run.py [-h] [-H HOSTS [HOSTS ...]] [-G GROUPS
[GROUPS ...]] [-P] [-u USER] [-p PASSWORD] [-c COMMAND
[COMMAND ...]] [-m MULTI_COMMANDS_FILE] [-t TIMEOUT] [-o
{0,1,2,3,4}]
                        [-l LIST_HOSTS [LIST_HOSTS ...]] [-v]

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

                        "ALL" means all hosts on
/ic/data/usr/liyanqing.1987/tools/batchRun/config/host.list.
                        "~<host>" means exclud specified host.
  -o {0,1,2,3,4}, --output_message_level {0,1,2,3,4}
                        Specify output message level, default is
"3".

                        "0" : silence mode;
                        "1" : only show host info;
                        "2" : only show one line output message;
                        "3" : show normal output message;
                        "4" : show verbose output message.
  -l LIST_HOSTS [LIST_HOSTS ...], --list_hosts LIST_HOSTS
[LIST_HOSTS ...]
                        List all or specified-group hosts on
/ic/data/usr/liyanqing.1987/tools/batchRun/config/host.list.
                        The format could be:
                        <GROUP>

                        "ALL" means all hosts on host list file.
                        "<GROUP>" means hosts on specified groups.
  -v, --version          Get batch_run 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。

--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。

--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。可以是系统命令或者指定脚本，也可以是 batchRun 自带脚本库中的脚本。

--multi_commands_files : 如果 ssh 登录到远程机器上需要执行多行命令，可以将多行命令写入到文件中，然后通过--multi_commands_files 来指定多命令文本。

--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 共五个登记，其中“0”意味着完全静默，“4”意味着最详细的信息输出。

--list_hosts : 列出配置文件 host.list 中所有的机器。

--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。

```
TypeScript
[root@ic-admin1 batchRun]# bin/batch_run --version
Version : 1.1
Release Date : 2023.7
```

4.3.2 列出预设机器

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_hosts all”查看。


```

Bash
[root@ic-admin1 batchRun]# bin/batch_run --list_hosts ALL
GROUP : [LOGIN]
        10.232.134.66      n232-134-066
        10.232.135.66      n232-135-066
GROUP : [PROJ_A]
        10.232.133.20
        10.232.133.21      n232-133-021
GROUP : [PROJ_B]
        10.232.133.23      n232-133-023
        10.232.133.23      analog_cmp
        10.232.133.24      n232-133-024
GROUP : [VM]
        10.232.133.66      lsfmaster
        10.232.133.67      lsfmaster
        10.212.206.197     n212-206-197   8080
GROUP : [PHYSICAL]
        LOGIN/
        PROJ_A/
        PROJ_B/
GROUP : [LSF_CMP]
        10.232.133.25      n232-133-025
        10.232.133.26      n232-133-026
        PHYSICAL/
        ~10.232.133.23
        ~n232-133-024
        ~LOGIN/
GROUP : [REL_VM]
        VM/
        ~lsfmaster

```

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

```

Bash
[root@ic-admin1 batchRun]# bin/batch_run --list_hosts LOGIN
GROUP : [LOGIN]
        10.232.134.66      n232-134-066
        10.232.135.66      n232-135-066

```

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 -u <USER> -p <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。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -u liyanqing.1987 -p ***
-H 10.232.134.66 -c 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。

```
Bash
[liyanqing.1987@ic-admin1 batchRun]$ tools/save_password -u
liyanqing.1987 -p ***
```

其储存的密码为加密样式。

```
Bash
[root@ic-admin1 ~]# cat config/password.encrypted
liyanqing.1987 f63792adf3ac037d46aacfd5b76589f1
```

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。

```
Bash
[liyanqing.1987@ic-admin1 batchRun]$ bin/batch_run -H
10.232.134.66 -c whoami
```

```
>>> 10.232.134.66
      liyanqing.1987
```

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 ic_admin，并不需要知道 ic_admin 的密码），从而导致拒绝执行命令。

4.3.5 指定机器执行命令

batch_run 支持用“batch_run -H <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）。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 n232-135-066 -c hostname

>>> 10.232.134.66
      n232-134-066

>>> n232-135-066 (10.232.135.66)
      n232-135-066
```

4.3.6 指定机器组执行命令

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

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -G PROJ_A PROJ_B -c hostname -o 2

>>> 10.232.133.20      n232-133-020
>>> 10.232.133.21      n232-133-021
>>> 10.232.133.23      n232-133-023
>>> 10.232.133.24      n232-133-024
```

也可以采用除外的方式排除掉指定的 group 或者 host，比如不想要 20 和 21 两台

机器，支持按照 ip/hostname 的方式排除机器。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -G PROJ_A PROJ_B -H
~10.232.133.20 ~n232-133-021 -c hostname -o 2
>>> 10.232.133.23      n232-133-023
>>> 10.232.133.24      n232-133-024
```

4.3.7 机器模糊匹配

同时，batch_run 也支持 ip 地址或者 hostname 的模糊匹配，如果匹配上多项，则多项同时执行。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 66 -c hostname -o 2
[FUZZY MATCH] 66 -> 10.232.134.66
[FUZZY MATCH] 66 -> 10.232.135.66
[FUZZY MATCH] 66 -> 10.232.133.66

>>> 66 (10.232.134.66)      n232-134-066
>>> 66 (10.232.135.66)      n232-135-066
>>> 66 (10.232.133.66)      n232-133-066
[root@ic-admin1 batchRun]#
[root@ic-admin1 batchRun]# bin/batch_run -H lsf -c hostname -o 2
[FUZZY MATCH] lsf -> lsfmaster -> 10.232.133.66
[FUZZY MATCH] lsf -> lsfmaster -> 10.232.133.67

>>> lsf (10.232.133.66)      n232-133-066
>>> lsf (10.232.133.67)      n232-133-067
```

4.3.8 串行执行和并行执行

先创建一个要执行的脚本。

```
Bash
[root@ic-admin1 batchRun]# cat
/home/liyanqing.1987/touch_hostname.sh
#!/bin/bash

cd /home/liyanqing.1987/test
```

```
echo `hostname`  
touch `hostname`
```

默认是串行执行，效果如下。

```
Bash  
[root@ic-admin1 batchRun]# bin/batch_run -G PROJ_A PROJ_B -c  
/home/liyanqing.1987/touch_hostname.sh -o 2  
>>> 10.232.133.20      n232-133-020  
>>> 10.232.133.21      n232-133-021  
>>> 10.232.133.23      n232-133-023  
>>> 10.232.133.24      n232-133-024  
[root@ic-admin1 batchRun]#  
[root@ic-admin1 batchRun]# ls /home/liyanqing.1987/test/  
n232-133-020  n232-133-021  n232-133-023  n232-133-024
```

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，用并行方式再执行一遍。

并行命令为“batch_run -P”。

```
Bash  
[root@ic-admin1 batchRun]# rm -f /home/liyanqing.1987/test/*  
[root@ic-admin1 batchRun]#  
[root@ic-admin1 batchRun]# bin/batch_run -P -G PROJ_A PROJ_B -c  
/home/liyanqing.1987/touch_hostname.sh -o 2  
>>> 10.232.133.23 >>> 10.232.133.20 >>> 10.232.133.21 >>>  
10.232.133.24      n232-133-023  
      n232-133-024  
      n232-133-020  
      n232-133-021  
[root@ic-admin1 batchRun]#  
[root@ic-admin1 batchRun]# ls /home/liyanqing.1987/test/  
n232-133-020  n232-133-021  n232-133-023  n232-133-024
```

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，但是输出信息是乱序的，和机器对应不上，这是由于并行执行的不确定性导致的。

4.3.9 执行多行命令

如果在同一台机器上需要执行多行命令，可以采用参数“batch_run -m <FILE>”来实现。

我们先创建一个文件，写入多行命令。（每一行都是有效命令行）

```
Bash
[root@ic-admin1 batchRun]# cat /home/liyanqing.1987/get_host_info
cat /etc/redhat-release
cat /proc/cpuinfo | grep name | sort | uniq
cat /proc/meminfo | grep MemTotal
```

执行多行命令。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 -m
/home/liyanqing.1987/get_host_info

>>> 10.232.134.66
    CentOS Linux release 7.9.2009 (Core)
    model name           : Intel(R) Xeon(R) Platinum 8260 CPU @
2.40GHz
    MemTotal:            1056265156 kB
```

如果远程机器挂载了共享目录，可以用脚本的方式取代多行模式。但是在远程机器没有共享目录的情况下，是没办法执行指定脚本的，只能以多行模式来实现同一效果。

4.3.10 批量配置 ssh 免密

batchRun 自有脚本库位于安装目录下的 scripts 目录，可以放入自有脚本。“nopassword_ssh.sh”就是一个自带脚本，可以用于共享存储条件下配置服务器间用户 ssh 免密登录。

```
Bash
[liyanqing.1987@ic-admin1 batchRun]$ bin/batch_run -G LOGIN -c
nopassword_ssh.sh

>>> 10.232.134.66
```

```

Generating public/private rsa key pair.
Your identification has been saved in id_rsa.
Your public key has been saved in id_rsa.pub.
The key fingerprint is:
SHA256:v7EWmdNpG9BwzALdKLJLy4y1UxYow2Ah4nlwu5uu1R0
liyanqing.1987@n232-134-066
The key's randomart image is:
+---[RSA 2048]-----+
|o.++ . .o =      |
|oo= + o . = =     |
| o *   + . =      |
|  o . *   . .     |
|   . X ES  = .    |
|    B * ..= =     |
|   + o .  o+ o    |
|  o         .+.    |
| ...        .o     |
+-----[SHA256]-----+

>>> 10.232.135.66
Generating public/private rsa key pair.
Your identification has been saved in id_rsa.
Your public key has been saved in id_rsa.pub.
The key fingerprint is:
SHA256:9+DmzgkSMT954JT5KX4vGTFOlMhrwJQankGqigXRxCU
liyanqing.1987@n232-135-066
The key's randomart image is:
+---[RSA 2048]-----+
| +E..            |
|. +.      o      |
| +   .o =.       |
|o o o o *o+o.    |
|.o B ..=S *o     |
|. = o oo.*.o     |
|o.   o..o +o.    |
|o      .. *oo    |
|                |
|                |
+-----[SHA256]-----+

```

配置完毕后，即可在指定服务器间免密登录。

```

Bash
[liyanqing.1987@ic-admin1 batchRun]$ ssh 10.232.134.66

```

```
Last login: Mon Mar 20 10:09:22 2023 from 10.232.135.66
[liyanqing.1987@n232-134-066 ~]$
```

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，静默模式，无任何输出。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 -c
"lsb_release -a" -o 0
[root@ic-admin1 batchRun]#
```

- “1”级输出，极简模式，仅显示执行的机器。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 -c
"lsb_release -a" -o 1
>>> 10.232.134.66
[root@ic-admin1 batchRun]#
```

- “2”级输出，当行模式，仅显示 output message 的第一行，且显示到机器信息的同一行。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 -c
"lsb_release -a" -o 2
>>> 10.232.134.66      LSB Version:          :core-4.1-amd64:core-
4.1-noarch:cxx-4.1-amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-
4.1-noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-
amd64:printing-4.1-noarch
[root@ic-admin1 batchRun]#
```

- “3”级输出，正常模式，正常显示 output message。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 -c
```



```
"lsb_release -a" -o 3
```

```
>>> 10.232.134.66
```

```
    LSB Version:          :core-4.1-amd64:core-4.1-noarch:cxx-4.1-  
amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-  
noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-  
amd64:printing-4.1-noarch  
    Distributor ID:      CentOS  
    Description:         CentOS Linux release 7.9.2009 (Core)  
    Release:             7.9.2009  
    Codename:            Core
```

- “4”级输出，详细模式，除正常 output message 外，会打印额外的 debug 信息。

Bash

```
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 -c  
"lsb_release -a" -o 4
```

```
>>> 10.232.134.66
```

```
/ic/data/usr/liyanqing.1987/tools/batchRun/tools/run_ssh_command.p  
y -c "ssh -o StrictHostKeyChecking=no root@10.232.134.66  
lsb_release \-a" -H 10.232.134.66 -p *** -t 10  
==== output ====  
    LSB Version:          :core-4.1-amd64:core-4.1-noarch:cxx-4.1-  
amd64:cxx-4.1-noarch:desktop-4.1-amd64:desktop-4.1-  
noarch:languages-4.1-amd64:languages-4.1-noarch:printing-4.1-  
amd64:printing-4.1-noarch  
    Distributor ID:      CentOS  
    Description:         CentOS Linux release 7.9.2009 (Core)  
    Release:             7.9.2009  
    Codename:            Core  
=====
```

默认的输出级别是“3”。

五、辅助工具

出于安全和便捷性的考虑，batch_run 自带一些辅助工具。

5.1 save_password

用于加密用于密码并保存。

Bash

```
[liyanqing.1987@ic-admin1 batchRun]$ tools/save_password -h
usage: save_password.py [-h] [-P PASSWORD_FILE] [-u USER] -p
PASSWORD [-H HOST]
```

optional arguments:

```
-h, --help                show this help message and exit
-P PASSWORD_FILE, --password_file PASSWORD_FILE
                        Specify the user password file, default is
                        "/ic/data/usr/liyanqing.1987/tools/batchRun/config/password.encrypted".
-u USER, --user USER    Specify user name, default is current user.
-p PASSWORD, --password PASSWORD
                        Specify user password.
-H HOST, --host HOST    Specify the host which user password is
                        for, default is for all hosts.
```

-P 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。

-u USER，指定用户名，默认是当前用户。

-p PASSWORD，指定用户密码。

-H HOST，指定账号密码生效的机器，默认为所有机器。

下面是一个例子。

TypeScript

```
[liyanqing.1987@ic-admin1 batchRun]$ tools/save_password -u
liyanqing.1987 -p Lishiyu@2017
[liyanqing.1987@ic-admin1 batchRun]$ tools/save_password -u
ic_admin -p abc -H 10.232.134.66
```

```
[liyanqing.1987@ic-admin1 batchRun]$  
[liyanqing.1987@ic-admin1 batchRun]$ cat config/password.encrypted  
liyanqing.1987 f63792adf3ac037d46aacfd5b76589f1  
ic_admin e23f3107f61ea703c5d3319e9c59f668 10.232.134.66
```

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。

5.2 get_password

用于查看自己保存的密码，注意，只能查看用户自己的密码。

```
Bash  
[liyanqing.1987@ic-admin1 batchRun]$ tools/get_password -h  
usage: get_password.py [-h] [-P PASSWORD_FILE] [-u USER] [-H HOST]  
  
optional arguments:  
  -h, --help                show this help message and exit  
  -P PASSWORD_FILE, --password_file PASSWORD_FILE  
                           Specify the user password file, default is  
                           "/ic/data/usr/liyanqing.1987/tools/batchRun/config/password.encrypted".  
  -u USER, --user USER     Specify the user name, default is current  
                           user.  
  -H HOST, --host HOST      Specify the host which user password is  
                           for, default is for all hosts.
```

-P 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。

-u USER，指定用户名，默认是当前用户。

-H HOST，指定账号密码生效的机器，默认是所有机器。

下面是一个例子。

```
TypeScript  
[ic_admin@n232-135-066 batchRun]$ tools/get_password -u  
liyanqing.1987
```

```
[ic_admin@n232-135-066 batchRun]$ tools/get_password -u ic_admin  
  
[ic_admin@n232-135-066 batchRun]$ tools/get_password -u ic_admin -  
H 10.232.134.66  
abc
```

ic_admin 尝试获取 liyanqing.1987 的密码，会返回空（因为没有权限，只能获取自身密码）。

ic_admin 尝试获取自己在所有机器上的密码，会返回空（因为没有存储）。

ic_admin 尝试获取自己在 10.232.134.66 上的密码，会返回 abc（之前存储值）。

5.3 essh

采用 expect 的方式无密码登录指定机器，用法如下。

```
TypeScript  
essh <host_ip>
```

如果用 save_password 保存过加密密码，登录机器不依赖用户的 ssh 免密登录，也不需要输入密码，比较方便。

```
Bash  
[liyanqing.1987@ic-admin1 batchRun]$ tools/essh 10.232.134.66  
spawn ssh -XY -o StrictHostKeyChecking=no -p 22 10.232.134.66  
[liyanqing.1987@n232-134-066 ~]$
```

5.4 xssh

采用 xterm 的方式无密码登录指定机器，开启模糊匹配的前提下，机器支持简写，用法如下。

```
TypeScript  
[liyanqing.1987@ic-admin1 batchRun]$ tools/xssh -h  
Usage: xssh host <port>
```

如果用 `save_password` 保存过加密密码，登录机器不依赖用户的 `ssh` 免密登录，也不需要输入密码，而且还支持 `host` 简写，非常方便，强烈推荐！

Bash

```
[liyanqing.1987@n232-134-194 batchRun]$ tools/xssh 26  
[FUZZY MATCH] 26 -> 10.232.133.26
```

```
xterm -e "/ic/data/usr/liyanqing.1987/tools/batchRun/tools/essh  
10.232.133.26 22" &
```



如果有多个匹配的 `hosts`，会列出供用户选择。

C#

```
[liyanqing.1987@n232-134-194 batchRun]$ tools/xssh 66  
[FUZZY MATCH] 66 -> 10.232.134.66  
[FUZZY MATCH] 66 -> 10.232.135.66  
[FUZZY MATCH] 66 -> 10.232.133.66
```

Below are possible hosts:

```
[0] 10.232.134.66  
[1] 10.232.135.66  
[2] 10.232.133.66  
[3] None of above hosts
```

Please choice one (number): 2

```
xterm -e "/ic/data/usr/liyanqing.1987/tools/batchRun/tools/essh  
10.232.133.66 22" &
```

```
liyanqing.1987@n232-133-066:~  
spawn ssh -XY -o StrictHostKeyChecking=no -p 22 10.232.133.66  
Warning: No xauth data; using fake authentication data for X11 forwarding.  
Last login: Mon Jul 17 14:39:47 2023 from 10.232.135.66  
[liyanqing.1987@n232-133-066 ~]$
```

5.5 switch_etc_hosts

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

```
TypeScript  
[liyanqing.1987@ic-admin1 batchRun]$ tools/switch_etc_hosts -h  
usage: switch_etc_hosts.py [-h] [-i INPUT_FILE] [-o OUTPUT_FILE]  
  
optional arguments:  
  -h, --help                show this help message and exit  
  -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".
```

/etc/hosts 文件需要符合如下格式才能被正确解析。

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

下面是一个示例 hosts 文件。

TypeScript

```
[liyanqing.1987@ic-admin1 batchRun]$ cat ~/hosts
```

```
# GROUP : LOGIN
```

```
10.212.206.194 n212-206-194
```

```
10.212.206.195 n212-206-195
```

```
# GROUP : PROJ_A
```

```
10.212.206.197 n212-206-197 # SSH_PORT=8080
```

```
10.232.134.206 n232-134-206
```

```
# GROUP : PROJ_B
```

```
10.212.206.200 n212-206-200
```

```
10.212.206.201 n212-206-201
```

```
# GROUP : VM
```

```
10.232.159.46 ic-hpc-mon01
```

```
10.212.240.47 ic-hpc-mon02
```

转换后的 host.list 文件如下。

TypeScript

```
[liyanqing.1987@ic-admin1 batchRun]$ tools/switch_etc_hosts -i  
~/hosts -o ./host.list
```

```
Output File : ./host.list
```

```
[liyanqing.1987@ic-admin1 batchRun]$ cat ./host.list
```

```
[LOGIN]
```

```
10.212.206.194 n212-206-194
```

```
10.212.206.195 n212-206-195
```

```
[PROJ_A]
```

```
10.212.206.197 n212-206-197 8080
```

```
10.232.134.206 n232-134-206
```

```
[PROJ_B]
```

```
10.212.206.200 n212-206-200
```

```
10.212.206.201 n212-206-201
```

```
[VM]
```

```
10.232.159.46 ic-hpc-mon01
```

```
10.212.240.47 ic-hpc-mon02
```

5.6 patch

patch 是帮助 batchRun 打补丁的工具，其帮助信息如下。

```
TypeScript
[liyanqing.1987@ic-admin1 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。

```
TypeScript
[liyanqing.1987@ic-admin1 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、自带脚本库

`batch_run` 可以执行系统命令，也可以指定指定路径下的脚本（前提是远程机器挂载了共享目录，可以读到这个脚本），有时候我们为了方便，会把一些常用操作写成脚本来执行（IT 运维自动化），为了更方便地引用这些脚本，可以把它们放到 `batchRun` 的自带脚本库“<INSTALL_PATH>/scripts”下面，这样就可以无路径引用这些脚本。

比如我们在 `batchRun` 安装目录下的 `scripts` 目录中写一个脚本，用于获取服务器的基本信息。

```
Bash
[root@ic-admin1 batchRun]# cat scripts/monitor/get_host_info.sh
#!/bin/bash

host_name=`hostname`
echo "Host : ${host_name}"

cpu_num=`cat /proc/cpuinfo | grep "cpu cores" | uniq | awk '{print $4}'`
echo "Cpu  : ${cpu_num}"

total_mem=`free -h | grep Mem | awk '{print $2}'`
echo "Mem  : ${total_mem}"

uptime=`uptime`
echo "Load : ${uptime}"
```

执行这个自带脚本库中的脚本时，无需指明路径，只写上脚本名即可。

```
Bash
[root@ic-admin1 batchRun]# bin/batch_run -H 10.232.134.66 -c
get_host_info.sh

>>> 10.232.134.66
    Host : n232-134-066
    Cpu  : 24
    Mem  : 1.0T
    Load : 20:09:08 up 127 days, 3:13, 410 users, load average:
0.86, 0.95, 0.89
```

这样我们就可以把如下类型的 IT 运维自动化脚本写入 `batchRun`，然后便捷引用。

- 操作系统初始化。
- 操作系统优化。
- 磁盘挂载。
- 常用服务安装和启动。
- 常用系统配置。

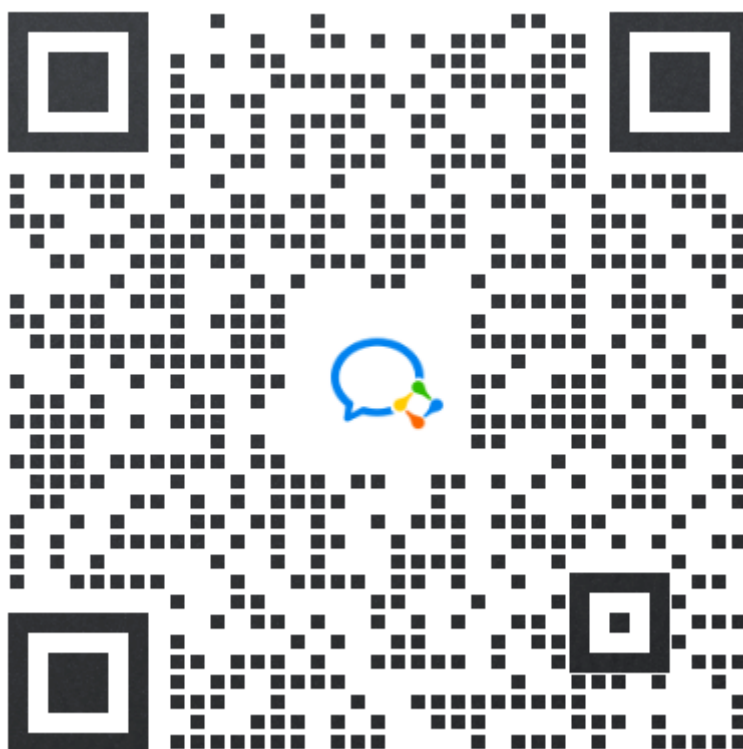
七、技术支持

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。
- 加入官方技术支持群。



附录

附 1. 变更历史

日期	版本	变更描述
2022.12.12	1.0	发布第一个正式 release 版本
2023.7.17	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。