

ssh-batch



Run ssh remote commands in batch mode sequential or in parallel

Sources for ssh-batch

- PUBLIC

The author

<https://github.com/hans-vervaart/ssh-batch>
or

<https://github.com/FBnil/ssh-batch>

The tester

- Installation is easy:
Untar to any directory (we recommend into ~/bin/)

What is an ssh remote command?

- Running a command locally (bash)

```
nilton@debian:~/$ uname -a ; cat /etc/debian_version  
Linux debian 4.19.0-12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64 GNU/Linux  
10.10  
nilton@debian:~/$
```

- Running a command remotely (ssh)

```
nilton@debian:~/$ ssh ansible@rocky 'uname -a ; cat /etc/redhat-release'  
ansible@rocky's password: *****  
Linux rocky 4.18.0-305.el8.x86_64 #1 SMP Thu May 27 16:46:28 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux  
Rocky Linux release 8.4 (Green Obsidian)  
nilton@debian:~/$
```

- Ssh is able to run ad-hoc commands
- But it requires entering the password for each connection

ssh remote command using a scriptfile

- Running script remotely using ssh

```
nilton@debian:~/$ cat cmd.sh
uname -a
cat /etc/redhat-release
```

- nilton@debian:~/\$ **ssh ansible@rocky <cmd.sh**
Pseudo-terminal will not be allocated because stdin is not a terminal.
ansible@rocky's password: ****
Activate the web console with: `systemctl enable --now cockpit.socket`
Linux rocky 4.18.0-305.el8.x86_64 #1 SMP Thu May 27 16:46:28 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
Rocky Linux release 8.4 (Green Obsidian)

```
nilton@debian:~/$ <cmd.sh ssh ansible@rocky
Pseudo-terminal will not be allocated because stdin is not a terminal.
ansible@rocky's password: ****
Activate the web console with: systemctl enable --now cockpit.socket
Linux rocky 4.18.0-305.el8.x86_64 #1 SMP Thu May 27 16:46:28 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
Rocky Linux release 8.4 (Green Obsidian)
```

- We can then capture this output by appending this to the command: **> rocky-output.log**

askpass

- Ssh comes with SSH_ASKPASS to automatically ask for a password.

SSH_ASKPASS

If ssh needs a passphrase, it will read the passphrase from the current terminal if it was run from a terminal. If ssh does not have a terminal associated with it but DISPLAY and SSH_ASKPASS are set, it will execute the program specified by SSH_ASKPASS and open an X11 window to read the passphrase. This is particularly useful when calling ssh from a .xsession or related script. (Note that on some machines it may be necessary to redirect the input from /dev/null to make this work.)

Manual page ssh(1) line 525 (press h for help or q to quit)

- But it does not provide a vault nor a program
- (There is also the ssh_keys method but that requires pushing ssh_keys to each server)

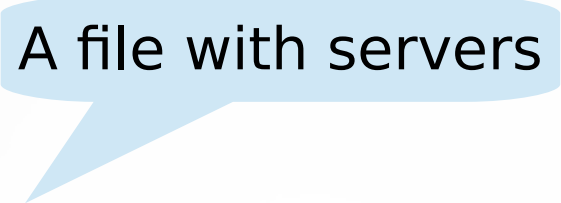
Running without password vault

- Password has to be typed in only once (but must be the same password for all servers)

```
nilton@debian:~/ssh-batch$ ./ssh-batch user@rocky user@centos6 -- cmd1.sh
cmd2.sh --- hello world
# [INFO] Using inline file cmd1.sh
# [INFO] Using inline file cmd2.sh
[ssh_askpass] Enter password:*****
# ssh -o NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no user@rocky
user@rocky's password: <ssh_askpass>
We are using parameter1=hello and parameter2=world in this script.
Linux rocky 4.18.0-305.el8.x86_64 #1 SMP Thu May 27 16:46:28 UTC 2021 x86_64 x86_64
x86_64 GNU/Linux
Rocky Linux release 8.4 (Green Obsidian)
# ssh -o NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no user@centos6
user@centos6's password: <ssh_askpass>
We are using parameter1=hello and parameter2=world in this script.
Linux centos6A.localdomain 2.6.32-754.el6.x86_64 #1 SMP Tue Jun 19 21:26:04 UTC 2018
x86_64 x86_64 x86_64 GNU/Linux
CentOS release 6.10 (Final)
```



```
nilton@debian:~/ssh-batch$ cat cmd1.sh
echo "We are using parameter1=$1 and parameter2=$2 in this script."
nilton@debian:~/ssh-batch$ cat cmd2.sh
uname -a
cat /etc/redhat-release
nilton@debian:~/ssh-batch$ cat ./serverlist
user@rocky
user@centos6
nilton@debian:~/ssh-batch$ ./ssh-batch ./serverlist -- cmd1.sh cmd2.sh --- with parameter
```



A file with servers

- Same result from using a serverlist instead of single servers
- For Jumphosts (ssh proxy servers) use the carrot: proxyuser@proxyserver:22^user@server:22
- Multiple Jumphosts allowed

But what if we have different userids? We will need a password vault for that

Creating a vault

```
nilton@debian:~/ssh-batch$ ./ssh_askpass
```

```
Usage: ssh_askpass [options] [account] ..
```

```
Options:
```

```
--exports
--flush-cache
-h           ← short help
--help      ← long help
--no-set
--no-trap
--update-passwords
--vault-create      ← create our vault
--vault-ignore
--vault-relock
--vault-remove
--vault-show
```



Default password

```
nilton@debian:~/ssh-batch$ ./ssh_askpass --vault-create user ansible root@rocky "
```

```
[user@ssh_askpass] Enter password:****
```

```
[ansible@ssh_askpass] Enter password:*****
```

```
[root@rocky@ssh_askpass] Enter password:****
```

```
[ssh_askpass] Enter password:***** ← default password to fallback to
```

```
[ssh_askpass] Lock vault secret:*
```

```
[ssh_askpass] Check vault secret:*
```

```
[ssh_askpass] New vault content written
```


Now open the vault and run

- Method 1 (universal)

```
nilton@debian:~/ssh-batch$ source ./ssh_askpass --exports  
[bash] Enter password:*
```

```
export SSH_ASKPASS_CACHE=H4sIANEt0mAAA1M2NjI3teCySk23teUCAOW47PoNAAAA-  
export SSH_ASKPASS=/bin/bash  
export DISPLAY=:0  
export SETSID=/usr/bin/setsid
```

```
nilton@debian:~/ssh-batch$ echo $SSH_ASKPASS_CACHE ← do this to verify your vault is open  
H4sIANEt0mAAA1M2NjI3teCySk23teUCAOW47PoNAAAA-
```

- Method 2 (Fbnil's extra openvault script which does the same as above, but is easier to remember)

```
nilton@debian:~/ssh-batch$ . openvault  
[ssh_askpass] Open vault secret:*
```

- Now you can run without entering passwords

```
nilton@debian:~/ssh-batch$ ./ssh-batch ./serverlist -- cmd1.sh cmd2.sh --- with parameter  
# [INFO] Using inline file cmd1.sh  
# [INFO] Using inline file cmd2.sh  
# ssh -o NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no user@rocky  
user@rocky's password: <ssh_askpass> (user)  
We are using parameter1=with and parameter2=parameter in this script.  
Linux rocky 4.18.0-305.el8.x86_64 #1 SMP Thu May 27 16:46:28 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux  
Rocky Linux release 8.4 (Green Obsidian)  
# ssh -o NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no user@centos6  
user@centos6's password: <ssh_askpass> (user)  
We are using parameter1=with and parameter2=parameter in this script.  
Linux centos6A.localdomain 2.6.32-754.el6.x86_64 #1 SMP Tue Jun 19 21:26:04 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux  
CentOS release 6.10 (Final)
```

Run in parallel (Many more options, see: **./ssh-batch --help**)

- Ssh-batch can run in parallel by using --bg-log-dir (the directory where it can save the output into).

```
nilton@debian:~/ssh-batch$ mkdir ./output
```

```
nilton@debian:~/ssh-batch$ ./ssh-batch --bg-log-dir ./output ./serverlist -- cmd1.sh cmd2.sh --- with parameter
```

- # [INFO] Using inline file cmd1.sh
[INFO] Using inline file cmd2.sh
[ssh_askpass] Open vault secret:*
ssh -o NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no user@rocky
Output send to ./output/rocky
ssh -o NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no user@centos6
Output send to ./output/centos6
Background sessions running
Waiting for 0/2 sessions to finish (09:59)
- Output

```
nilton@debian:~/ssh-batch$ ls output/  
centos6 rocky
```

```
nilton@debian:~/ssh-batch$ cat output/rocky
```

```
user@rocky's password: <ssh_askpass> (user)
```

```
We are using parameter1=with and parameter2=parameter in this script.
```

```
Linux rocky 4.18.0-305.el8.x86_64 #1 SMP Thu May 27 16:46:28 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux  
Rocky Linux release 8.4 (Green Obsidian)
```

./ssh-batch --help

Usage: ssh-batch [options] hosts [[- adhoc-cmds]][-- inline-files]][--- args]

The hosts argument can be a single host name or multiple hostnames.

Multiple hostnames are multiple arguments. Arguments including a / are text files containing hostnames. Tip: Hostname files can be nested. A hostname can contain a port (hostname:port), which will be passed as:

ssh -p port hostname

--no-info

No info messages, only ssh command and host output.

--no-ssh-askpass

Disable ssh_askpass usage (when ssh-keys are used).

--quiet

Only host output.

--ssh-option <value>

Options to be passed to the ssh command.

--tags

Add tags; inline names surrounded by { } in the output.

Jumphosts can be added with a ^ as concatenation character to the hostname (jumphost^hostname). Multiple jumphosts are supported.

Options:

--all-hosts

Enables the hostnames with a 'soft' comment (- or %).

--bg-disable

Disable background runs, when --bg-log-dir is provided.

--bg-log-dir <value>

Background logs directory and enable background runs.

--bg-session-throttle <value>

Maximum background runs.

--bg-timeout <value>

The wait for all background runs time-out.

-C

Requests ssh to use gzip compression for all transmitted data.

--exit-status

Add the EXIT_STATUS of the inline to the output.

-F <value>

Specifies an alternative per-user ssh configuration file.

--hostnames

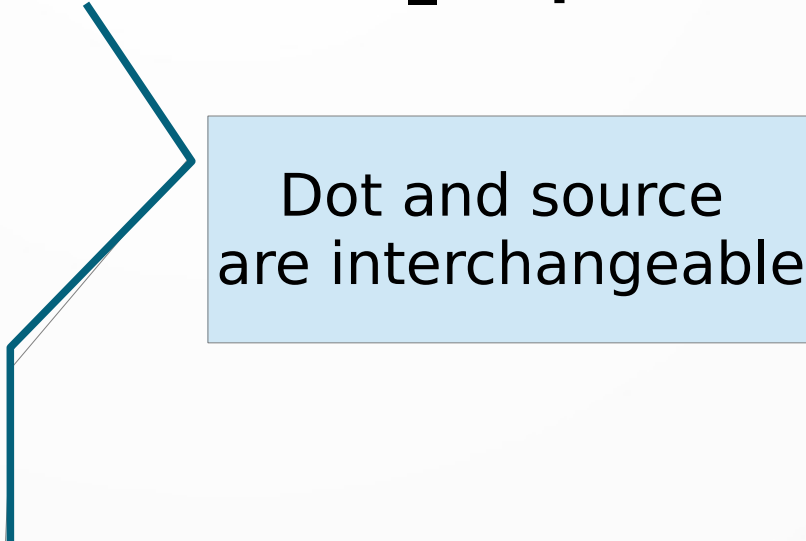
Add the hostname to the output.

Don't forget to close/delete the vault once you are done (or close the terminal)

- Method 1 (universal)

```
nilton@debian:~/ssh-batch$ source ./ssh_askpass --flush-cache
```

```
unset SSH_ASKPASS_CACHE  
unset SSH_ASKPASS  
unset SETSID
```



Dot and source
are interchangeable

- Method 2 (Fbnil's)

```
nilton@debian:~/ssh-batch$ . closevault
```

- Dont forget to verify the vault is closed (variable is empty again):

```
nilton@debian:~/ssh-batch$ echo $SSH_ASKPASS_CACHE
```

~/.ssh/

- All configuration files of ssh-batch are written to ~/.ssh/ for example:

```
nilton@debian:~/.ssh$ ls -l
```

```
total 36
```

```
-rw----- 1 nilton nilton    0 jun 22 21:35 askpass
-rw----- 1 nilton nilton   33 jun 22 19:13 askpass.algorithm
-rw----- 1 nilton nilton  350 jun 22 21:34 askpass.vault
-rw-r--r-- 1 nilton nilton  291 jun 21 17:49 config
-rw----- 1 nilton nilton  464 jun 20 00:54 id_ed25519
-rw-r--r-- 1 nilton nilton   98 jun 20 00:54 id_ed25519.pub
-rw-r--r-- 1 nilton nilton 4113 jun 21 17:49 known_hosts
```

- askpass.vault is your encrypted password vault
- askpass is your configuration file to group servernames for same pwd
- askpass.algorithm is created only in the Fbnil fork and allows you to change your encryption before you create your vault (just run ./ssh_askpass without parameters to create the files)
- If your vault is faulty, just delete the askpass* files (but not the other files!)

Using scp

- scp-batch has not been written yet, meanwhile open the vault, then run:

```
nilton@debian:~/ssh-batch$ echo $(setsid scp -o  
NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no ./poem.txt rocky:/  
tmp)
```

```
ansible@rocky's password: <ssh_askpass> (ansible)
```

- For parallel scp we use **xargs -P \$nr_of_procs**, like so:
- nilton@debian:~/ssh-batch\$ cat serverlist
user@rocky
user@centos6
- nilton@debian:~/ssh-batch\$ <serverlist xargs -d '\n' -P 2 -I {}
/bin/bash -c "setsid scp ./poem.txt {}: /tmp"
user@rocky's password: <ssh_askpass> (user)
user@centos's password: <ssh_askpass> (user)

Using rsync

- rsync-batch has not been written yet, meanwhile open the vault, then run:

```
nilton@debian:~/ssh-batch$ setsid rsync -avz -e  
"ssh -p 5022 -o NumberOfPasswordPrompts=1 -o  
StrictHostKeyChecking=no -o  
UserKnownHostsFile=/dev/null" --progress  
./ssh-batch user@centos6:/tmp
```

Using rsync with Jumphost

- `nilton@debian:~/ssh-batch$ setsid rsync -avz -e "ssh -p 5022 -o NumberOfPasswordPrompts=1 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null" --progress ./ssh-batch user@centos6:/tmp`
- `rsync -azv -e 'ssh -J jumpuser@jumphost:22' local-file-path targetuser@targethost:target-save-location`
- `setsid rsync -r -e 'ssh -J ansible@192.168.1.68' --port=5022 ./roses ansible@192.168.1.68`
-

Need more structure? Take a look at Evidencer that can use ssh-batch and provides a structure.

Unable to write a script that ssh-batch can convert to inline? (for example: it uses HEREDOC syntax). Use rundeer
(it can be done with base64 encoded files though)

Need to place files everywhere? Wait for scp-batch (or use rundeer)