NAME

pssh — parallel ssh program

SYNOPSIS

pssh [**−vAiIP**] [**−h** *hosts_file*] [**−H** [*user*@]*host*[:*port*]] [**−l** *user*] [**−p** *par*] [**−o** *outdir*] [**−e** *errdir*] [**−t** *timeout*] [**−O** *options*] [**−x** *args*] [**−X** *arg*] *command ...*

pssh −I [**−vAiIP**] [**−h** *hosts_file*] [**−H** [*user*@]*host*[:*port*]] [**−l** *user*] [**−p** *par*] [**−o** *outdir*] [**−e** *errdir*] [**−t** *timeout*] [**−O** *options*] [**−x** *args*] [**−X** *arg*] [*command ...*]

DESCRIPTION

pssh is a program for executing ssh in parallel on a number of hosts. It provides features such as sending input to all of the processes, passing a password to ssh, saving output to files, and timing out.

OPTIONS

−h *host_file*
−−hosts *host_file*
Read hosts from the given *host_file*. Lines in the host file are of the form [*user*@]*host*[:*port*] and can include blank lines and comments (lines beginning with "#"). If multiple host files are given (the **−h** option is used more than once), then pssh behaves as though these files were concatenated together. If a host is specified specified multiple times, then pssh will connect the given number of times.

−H [*user*@]*host*[:*port*]
−−host [*user*@]*host*[:*port*]
−H "[*user*@]*host*[:*port*] [[*user*@]*host*[:*port*] ...]"
−−host "[*user*@]*host*[:*port*] [[*user*@]*host*[:*port*] ...]"
Add the given host strings to the list of hosts. This option may be given multiple times, and may be used in conjunction with the **−h** option.

−l *user*
−−user *user*
Use the given username as the default for any host entries that don't specifically specify a user.

−p *parallelism*
−−par *parallelism*
Use the given number as the maximum number of concurrent connections.

−t *timeout*
−−timeout *timeout*
Make connections time out after the given number of seconds. With a value of 0, pssh will not timeout any connections.

−o *outdir*
−−outdir *outdir*
Save standard output to files in the given directory. Filenames are of the form [*user*@]*host*[:*port*][.*num*] where the user and port are only included for hosts that explicitly specify them. The number is a counter that is incremented each time for hosts that are specified more than once.

−e *errdir*
−−errdir *errdir*
Save standard error to files in the given directory. Filenames are of the same form as with the **−o** option.

−x *args*
−−extra-args *args*
Passes a extra SSH command-line arguments (see the **ssh**(1) man page for more information about SSH arguments). This option may be specified multiple times. The arguments are processed to split on whitespace, protect text within quotes, and escape with backslashes. To pass arguments without such processing, use the **−X** option instead.

−X *arg*
−−extra-arg *arg*
Passes a single SSH command-line argument (see the **ssh**(1) man page for more information about SSH arguments). Unlike the **−x** option, no processing is performed on the argument, including word splitting. To pass multiple command-line arguments, use the option once for each argument.

−O *options*
−−options *options*
SSH options in the format used in the SSH configuration file (see the **ssh_config**(5) man page for more information). This option may be specified multiple times.

−A
−−askpass
Prompt for a password and pass it to ssh. The password may be used for either to unlock a key or for password authentication. The password is transferred in a fairly secure manner (e.g., it will not show up in argument lists). However, be aware that a root user on your system could potentially intercept the password.

−i
−−inline
Display standard output and standard error as each host completes.

−v
−−verbose
Include error messages from ssh with the **−i** and \ options.

−I
−−send-input
Read input and send to each ssh process. Since ssh allows a command script to be sent on standard input, the **−I** option may be used in lieu of the command argument.

−P
−−print
Display output as it arrives. This option is of limited usefulness because output from different hosts are interleaved.

EXAMPLE

Connect to host1 and host2, and print "hello, world" from each:

```
pssh -i -H "host1 host2" echo "hello, world"
```

Print "hello, world" from each host specified in the file hosts.txt:

```
pssh -i -h hosts.txt echo "hello, world"
```

Run a command as root with a prompt for the root password:

```
pssh -i -h hosts.txt -A -l root echo hi
```

Run a long command without timing out:

```
pssh -i -h hosts.txt -t 0 sleep 10000
```

If the file hosts.txt has a large number of entries, say 100, then the parallelism option may also be set to 100 to ensure that the commands are run concurrently:

```
pssh -i -h hosts.txt -p 100 -t 0 sleep 10000
```

Run a command without checking or saving host keys:

```
pssh -i -H host1 -H host2 -x "-O StrictHostKeyChecking=no -O UserKnownHostsFile=/dev/null
-O GlobalKnownHostsFile=/dev/null" echo hi
```

EXIT STATUS VALUES

0	Success
1	Miscellaneous error
2	Syntax or usage error
3	At least one process was killed by a signal or timed out.
4	All processes completed, but at least one ssh process reported an error (exit status 255).
5	There were no ssh errors, but at least one remote command had a non-zero exit status.

AUTHORS

Written by Brent N. Chun <bnc@theether.org> and Andrew McNabb <amcnabb@mcnabbs.org>.

http://code.google.com/p/parallel-ssh/

SEE ALSO

ssh(1), **pscp**(1), **prsync**(1), **pslurp**(1), **pnuke**(1)