The best method is to use [screen](http://en.wikipedia.org/wiki/GNU_Screen). Another method is to use [nohup](http://en.wikipedia.org/wiki/Nohup).

nohup is good to use for running process in background when process don't need any user input like httpd server or any other server process like that.

screen is more suitable if proc (process) needs user input. Like install scripts, yes/no prompts.

**Screen** is a "virtual" terminal which you can run from a "real" terminal (actually all terminals today are "virtual" but that is another topic for another day). Screen will keep running even if your ssh session gets disconnected. Any process which you start in a screen session will keep running with that screen session. When you reconnect to the server you can reconnect to the screen session and everything will be as if nothing happened, other than the time which passed.

Screen has a steep learning curve. "[Speaking UNIX: Stayin' alive with Screen](http://www.ibm.com/developerworks/aix/library/au-gnu_screen/)" is a nice tutorial with diagrams explaining the concept.

An alternative to screen is [tmux](http://tmux.github.io/).

**nohup** ignores the [HUP](http://en.wikipedia.org/wiki/SIGHUP) signal. The HUP signal is sent to all child processes when the controlling terminal is closed (for example by a disconnected ssh session). Processes started with nohup will not receive the HUP signal. Although the processes will keep running you can no longer interact with them because they are no longer attached to any terminal.

nohup is useful mainly for long running batch processes which, once started, no longer need any attention.

As an alternative to nohup, the Bash built-in command disown can be used to tell bash to not send the program the HUP signal.