



Introduction to Version Control

by Nitin Kumar

1. Git Setup <https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html>

```
nitin@ubuntu:~/Desktop$ sudo apt install git-all
[sudo] password for nitin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils cvs cvspds dh-elpa-helper
  elpa-dash elpa-github elpa-git-commit elpa-let-alist elpa-magit
  elpa-magit-popup elpa-treepy elpa-with-editor emacs emacs-bin-common
  emacs-common emacs-el emacs-gtk git git-cvs git-daemon-run git-doc git-el
  git-email git-gui git-man git-mediawiki git-svn gitk gitweb gsfonts
  imagemagick-6-common libalgorithm-c3-perl libapri libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libb-hooks-endofscope-perl
  libb-hooks-op-check-perl libbgi-fast-perl libbgi-pm-perl libbclass-c3-perl
  libbclass-cs-perl libbclass-data-inheritable-perl libbclass-inspector-perl
  libbclass-method-modifiers-perl libbclass-singleton-perl
  libbclass-xsaccessor-perl libbcommon-sense-perl libbdata-optlist-perl
  libdatetime-format-builder-perl libdatetime-format-iso8601-perl
  libdatetime-format-strptime-perl libdatetime-format-locale-perl libdatetime-perl
  libdatetime-timezone-perl libdbd-sqlite3-perl libdbi-perl
  libdevel-callchecker-perl libdevel-sqlite3-perl libdevel-lexalias-perl
  libdevel-stacktrace-perl libdigest-bubblebabble-perl libdigest-hmac-perl
  libdynaloader-functions-perl libemail-valid-perl liberror-perl
  libeval-closure-perl libexception-class-perl libfcgi-perl libfftw3-double3
  libfile-sharedir-perl libjson-perl libjson-xs-perl liblqr-1-0 liblua5.2-0
  libm17n-0 libmagickcore-6.q16-6 libmagickwand-6.q16-6 libmediawiki-api-perl
  libmodule-implementation-perl libmodule-runtime-perl libmro-compat-perl
  libnamespace-autoclean-perl libnamespace-clean-perl libnet-dns-perl
  libnet-dns-sec-perl libnet-domain-tld-perl libnet-ip-perl libnet-libidn-perl
  libnetf0 libpackage-stash-perl libpackage-stash-xs-perl libpadwalker-perl
  libparams-classify-perl libparams-util-perl libparams-validate-perl
  libparams-validationcompiler-perl libperl4-corelibs-perl libreadonly-perl
  libref-util-perl libref-util-xs-perl librole-tiny-perl libserf-1.1
  libspecio-perl libsub-exporter-perl libsub-exporter-progressive-perl
  libsub-identify-perl libsub-install-perl libsub-name-perl libsub-quote-perl
  libsvn-perl libsvn1 libtcl8.6 libterm-readkey-perl libtk8.6
  libtypes-serialiser-perl libutf8proc2 libvariable-magic-perl libxstring-perl
  libyaml-libyaml-perl libyaml-perl m17n-db runit tcl tcl8.6 tk tk8.6
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom mksh rcs
  mailutils emacs-common-non-dfsg ncurses-term meld mediawiki subversion
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libfftw3-bin libfftw3-dev m17n-docs libmagickcore-6.q16-6-extra
  libscalar-number-perl libtest-fatal-perl libyaml-shell-perl gawk
  tcl-tclreadline
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils cvs cvspds dh-elpa-helper
  elpa-dash elpa-github elpa-git-commit elpa-let-alist elpa-magit
  elpa-magit-popup elpa-treepy elpa-with-editor emacs emacs-bin-common
  emacs-common emacs-el emacs-gtk git git-cvs git-daemon-run git-doc git-el
  git-email git-gui git-man git-mediawiki git-svn gitk gitweb gsfonts
  imagemagick-6-common libalgorithm-c3-perl libapri libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libb-hooks-endofscope-perl
  libb-hooks-op-check-perl libbgi-fast-perl libbgi-pm-perl libbclass-c3-perl
  libbclass-cs-perl libbclass-data-inheritable-perl libbclass-inspector-perl
  libbclass-method-modifiers-perl libbclass-singleton-perl
  libbclass-xsaccessor-perl libbcommon-sense-perl libbdata-optlist-perl
  libdatetime-format-builder-perl libdatetime-format-iso8601-perl
  libdatetime-format-strptime-perl libdatetime-format-locale-perl libdatetime-perl
  libdatetime-timezone-perl libdbd-sqlite3-perl libdbi-perl
  libdevel-callchecker-perl libdevel-sqlite3-perl libdevel-lexalias-perl
  libdevel-stacktrace-perl libdigest-bubblebabble-perl libdigest-hmac-perl
  libdynaloader-functions-perl libemail-valid-perl liberror-perl
  libeval-closure-perl libexception-class-perl libfcgi-perl libfftw3-double3
  libfile-sharedir-perl libjson-perl libjson-xs-perl liblqr-1-0 liblua5.2-0
  libm17n-0 libmagickcore-6.q16-6 libmagickwand-6.q16-6 libmediawiki-api-perl
  libmodule-implementation-perl libmodule-runtime-perl libmro-compat-perl
  libnamespace-autoclean-perl libnamespace-clean-perl libnet-dns-perl
  libnet-dns-sec-perl libnet-domain-tld-perl libnet-ip-perl libnet-libidn-perl
  libnetf0 libpackage-stash-perl libpackage-stash-xs-perl libpadwalker-perl
  libparams-classify-perl libparams-util-perl libparams-validate-perl
  libparams-validationcompiler-perl libperl4-corelibs-perl libreadonly-perl
  libref-util-perl libref-util-xs-perl librole-tiny-perl libserf-1.1
  libspecio-perl libsub-exporter-perl libsub-exporter-progressive-perl
  libsub-identify-perl libsub-install-perl libsub-name-perl libsub-quote-perl
  libsvn-perl libsvn1 libtcl8.6 libterm-readkey-perl libtk8.6
  libtypes-serialiser-perl libutf8proc2 libvariable-magic-perl libxstring-perl
  libyaml-libyaml-perl libyaml-perl m17n-db runit tcl tcl8.6 tk tk8.6
```

2. Initialize a Git Repository

```
nitin@ubuntu:~/Desktop$ git init
Initialized empty Git repository in /home/nitin/Desktop/.git/
nitin@ubuntu:~/Desktop$ git add file.txt
```

3. Add files to the repository

```
nitin@ubuntu:~/Desktop$ git commit -m "first commit"

** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Run git commit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'nitin@ubuntu.(none)')
nitin@ubuntu:~/Desktop$ git config --global user.name "Nitin742"
nitin@ubuntu:~/Desktop$ git config --global user.email "nitin.kumar3@tothenew.com"
nitin@ubuntu:~/Desktop$ git config --list
user.name=Nitin742
user.email=nitin.kumar3@tothenew.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
nitin@ubuntu:~/Desktop$ git commit -m "first commit"
[master (root-commit) 79dfc9a] first commit
1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

4.Unstage 1 file

```
nitin@ubuntu:~/Desktop/Assessment$ git reset HEAD file1.txt  
nitin@ubuntu:~/Desktop/Assessment$
```

5.Commit the file

```
hitin@ubuntu:~/Desktop/Assessment$ git reset HEAD file1.txt
hitin@ubuntu:~/Desktop/Assessment$ git rm file1.txt
rm 'file1.txt'
hitin@ubuntu:~/Desktop/Assessment$ git commit -m "unstaged file"
[main 7bc8d65] unstaged file
1 file changed, 1 deletion(-)
delete mode 100644 file1.txt
hitin@ubuntu:~/Desktop/Assessment$
```


6. Add a remote

```
nitin@ubuntu:~/Desktop/Assessment$ git remote -v
origin  https://github.com/Nitin742/Version-Cont.git (fetch)
origin  https://github.com/Nitin742/Version-Cont.git (push)
nitin@ubuntu:~/Desktop/Assessment$ git remote set-url origin https://github.com/Nitin742/Remote.git
nitin@ubuntu:~/Desktop/Assessment$ git remote -v
origin  https://github.com/Nitin742/Remote.git (fetch)
origin  https://github.com/Nitin742/Remote.git (push)
nitin@ubuntu:~/Desktop/Assessment$
```

7. Undo changes to a particular file

```
nitin@ubuntu:~/Desktop/Assessment$ git log
commit 7bc8d658aeedc8eec957e83b746214ad8c914950 (HEAD -> main)
Author: Nitin742 <nitin.kumar3@tothenew.com>
Date:   Wed Sep 8 10:56:12 2021 -0700

    unstaged file

commit 0fc679f992e5bb4b801f2e3628a42119d827b5b9 (origin/main)
Author: Nitin742 <nitin.kumar3@tothenew.com>
Date:   Wed Sep 8 10:38:25 2021 -0700

    first commit
nitin@ubuntu:~/Desktop/Assessment$ git log --oneline
7bc8d65 (HEAD -> main) unstaged file
0fc679f (origin/main) first commit
nitin@ubuntu:~/Desktop/Assessment$ git reset --hard 0fc679f
HEAD is now at 0fc679f first commit
nitin@ubuntu:~/Desktop/Assessment$ git log --oneline
0fc679f (HEAD -> main, origin/main) first commit
nitin@ubuntu:~/Desktop/Assessment$
```


8. Push changes to Github

```
nitin@ubuntu:~/Desktop/Assessment$ git push -u origin main
Username for 'https://github.com': Nitin742
Password for 'https://Nitin742@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 283 bytes | 14.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/Nitin742/Remote.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
nitin@ubuntu:~/Desktop/Assessment$
```

9.Clone the repository

```
nitin@ubuntu:~/Desktop/Assessment$ git clone https://github.com/Nitin742/Remote.git
Cloning into 'Remote'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 263 bytes | 11.00 KiB/s, done.
nitin@ubuntu:~/Desktop/Assessment$
```

10. Add changes to one of the copies and pull the changes in the other.

```
nitin@ubuntu:~/Desktop/Assessment$ git branch
* main
nitin@ubuntu:~/Desktop/Assessment$ git checkout -b feature
Switched to a new branch 'feature'
nitin@ubuntu:~/Desktop/Assessment$ git branch
* feature
  main
nitin@ubuntu:~/Desktop/Assessment$ ls
file1.txt  file.txt  Remote
nitin@ubuntu:~/Desktop/Assessment$ cat >> file1.txt
hello again
^C
nitin@ubuntu:~/Desktop/Assessment$ git status
On branch feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Remote
    modified:   file1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   file1.txt

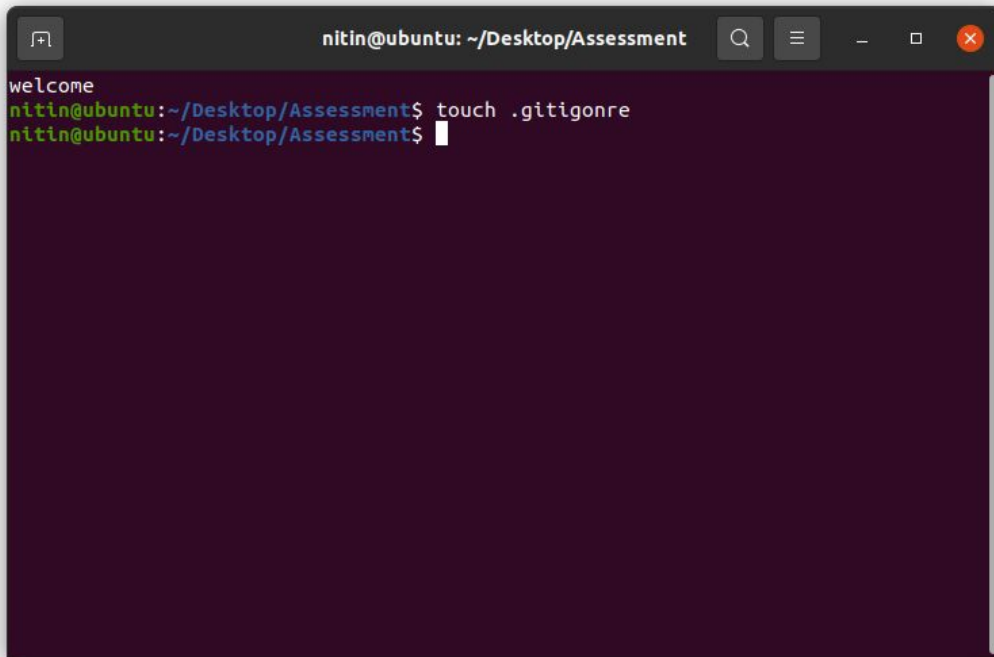
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nitin@ubuntu:~/Desktop/Assessment$ git -am "updated file1 in feature branch"
unknown option: -am
usage: git [-<version>] [-<help>] [-<C <path>] [-<c <name>=<value>]
  [-<exec-path>[=<path>]] [-<html-path>] [-<info-path>]
  [-<p | --paginate | -P | --no-pager>] [-<no-replace-objects>] [-<bare>]
  [-<git-dir=<path>] [-<work-tree=<path>] [-<namespace=<name>]
  <command> [<args>]

nitin@ubuntu:~/Desktop/Assessment$ git add .
nitin@ubuntu:~/Desktop/Assessment$ git commit
[feature b790957] update file1 in feature branch and added git ignore
 3 files changed, 4 insertions(+)
 create mode 100644 .gitignore
 create mode 160000 Remote
nitin@ubuntu:~/Desktop/Assessment$ git log --oneline --graph
* b790957 (HEAD -> feature) update file1 in feature branch and added git ignore
* 0fc679f (origin/main, origin/branch2, main) first commit
nitin@ubuntu:~/Desktop/Assessment$
```

12. Ignore a few files to be checked in

1 .git



```
nitin@ubuntu: ~/Desktop/Assessment
welcome
nitin@ubuntu:~/Desktop/Assessment$ touch .gitignore
nitin@ubuntu:~/Desktop/Assessment$
```

13. Create a new branch.

```
nitin@ubuntu:~/Desktop/Assessment$ git branch branch1
nitin@ubuntu:~/Desktop/Assessment$ git branch
  branch1
* main
nitin@ubuntu:~/Desktop/Assessment$ git branch branch2
nitin@ubuntu:~/Desktop/Assessment$ git checkout branch2
Switched to branch 'branch2'
nitin@ubuntu:~/Desktop/Assessment$ git merge branch1
Already up to date.
nitin@ubuntu:~/Desktop/Assessment$ git push origin branch2
Username for 'https://github.com': Nitin742
Password for 'https://Nitin742@github.com':
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:   https://github.com/Nitin742/Remote/pull/new/branch2
remote:
To https://github.com/Nitin742/Remote.git
 * [new branch]      branch2 -> branch2
```


14.Diverge them with commits

```
nitin@ubuntu:~/Desktop/Assessment$ git branch
* main
nitin@ubuntu:~/Desktop/Assessment$ git checkout -b feature
Switched to a new branch 'feature'
nitin@ubuntu:~/Desktop/Assessment$ git branch
* feature
  main
nitin@ubuntu:~/Desktop/Assessment$ ls
file1.txt  Remote
nitin@ubuntu:~/Desktop/Assessment$ cat >> file1.txt
hello again
^C
nitin@ubuntu:~/Desktop/Assessment$ git status
On branch feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Remote
    modified:   file1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   file1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .gitignore

nitin@ubuntu:~/Desktop/Assessment$ git -am "updated file1 in feature branch"
unknown option: -am
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

nitin@ubuntu:~/Desktop/Assessment$ git add .
nitin@ubuntu:~/Desktop/Assessment$ git commit
[feature b790957] update file1 in feature branch and added git ignore
 3 files changed, 4 insertions(+)
 create mode 100644 .gitignore
 create mode 160000 Remote
nitin@ubuntu:~/Desktop/Assessment$ git log --oneline --graph
* b790957 (HEAD -> feature) update file1 in feature branch and added git ignore
* 0fc679f (origin/main, origin/branch2, main) first commit
nitin@ubuntu:~/Desktop/Assessment$
```


15. Edit the same file at the same line on both branches and commit

```
nitin@ubuntu:~/Desktop/Assessment$ nano file1.txt
nitin@ubuntu:~/Desktop/Assessment$ git commit -am "changed line two in file1.txt on master branch"
[main 877727a] changed line two in file1.txt on master branch
 1 file changed, 1 insertion(+)
nitin@ubuntu:~/Desktop/Assessment$ git checkout feature
Switched to branch 'feature'
nitin@ubuntu:~/Desktop/Assessment$ nano file1.txt
nitin@ubuntu:~/Desktop/Assessment$ git commit -am "changed line two in file1.txt on faeture branch"
[feature 1592cc1] changed line two in file1.txt on faeture branch
 1 file changed, 1 insertion(+), 1 deletion(-)
nitin@ubuntu:~/Desktop/Assessment$
```

16. Try merging and resolve merge conflicts

```
nitin@ubuntu:~/Desktop/Assessment$ git merge feature
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.
nitin@ubuntu:~/Desktop/Assessment$ cat file1.txt
hello world  ...!!!!
<<<<<< HEAD
Nitin
=====
hello nitin
hello again
>>>>>> feature
nitin@ubuntu:~/Desktop/Assessment$ nano file1.txt
nitin@ubuntu:~/Desktop/Assessment$ git add .
nitin@ubuntu:~/Desktop/Assessment$ git commit -m "resolve merge conflict"
[main 31cf61b] resolve merge conflict
nitin@ubuntu:~/Desktop/Assessment$
```

17. Stash the changes and pop them

```

nitin@ubuntu:~/Desktop/Assessment$ cat >> file1.txt
making some changes to stash and pop
^C
nitin@ubuntu:~/Desktop/Assessment$ git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
nitin@ubuntu:~/Desktop/Assessment$ git stash
Saved working directory and index state WIP on main: 31cf61b resolve merge conflict
nitin@ubuntu:~/Desktop/Assessment$ git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
nitin@ubuntu:~/Desktop/Assessment$ git stash pop
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (64fad307b229402a4fe6f1ec732241a3a11ce547)
nitin@ubuntu:~/Desktop/Assessment$ git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
nitin@ubuntu:~/Desktop/Assessment$ git commit -am "committing stashed changes"
[main 5e0bdf6] committing stashed changes
1 file changed, 1 insertion(+)
nitin@ubuntu:~/Desktop/Assessment$

```

1. Add the following code to your .bashrc file : color_prompt="yes"

```

parse_git_branch() {
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*/(\1)/'
}
if [ "$color_prompt" = yes ]; then
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\[\033[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt

```

```
76 # test if /etc/colorshrc exists (if not, create it)
77 if [ ! -f /etc/colorshrc ]; then
78     # create /etc/colorshrc
79     alias ls='ls --color=auto'
80     #alias dir='dir --color=auto'
81     #alias vdir='vdir --color=auto'
82
83     alias grep='grep --color=auto'
84     alias fgrep='fgrep --color=auto'
85     alias egrep='egrep --color=auto'
86 fi
87
88 # colored GCC warnings and errors
89 #export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
90
91 # some more ls aliases
92 alias ll='ls -alF'
93 alias la='ls -A'
94 alias l='ls -CF'
95
96 # Add an "alert" alias for long running commands. Use like so:
97 # sleep 10; alert
98 alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error" "${history|tail -n1|sed -e '\''s/^s*[0-9]\+s*//;s/[:&]]\s*alert$/\''}'"
99
100 # Alias definitions.
101 # You may want to put all your additions into a separate file like
102 # ~/.bash_aliases, instead of adding them here directly.
103 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
104
105 if [ -f ~/.bash_aliases ]; then
106     . ~/.bash_aliases
107 fi
108
109 # enable programmable completion features (you don't need to enable
110 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
111 # sources /etc/bash.bashrc).
112 if ! shopt -oq posix; then
113     if [ -f /usr/share/bash-completion/bash_completion ]; then
114         . /usr/share/bash-completion/bash_completion
115     elif [ -f /etc/bash_completion ]; then
116         . /etc/bash_completion
117     fi
118 fi
119
120 # color_prompt="yes"
121 parse_git_branch() {
122     git branch 2> /dev/null | sed -e '/^[/\*]/d' -e 's/* \([^*]\)/(\{1\})/'
123 }
124
125 if [ "$color_prompt" = yes ]; then
126     PS1='\u@h[\[033[00m\]:\[033[01;34m\]\W[\[033[01;31m\] $(parse_git_branch)\[033[00m\]]$ '
127 else
128     PS1='\u@h:\W $(parse_git_branch)$ '
129 fi
130
131 unset color_prompt force_color_prompt
```