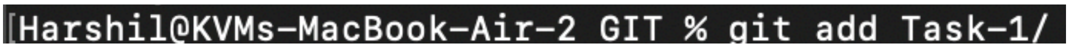
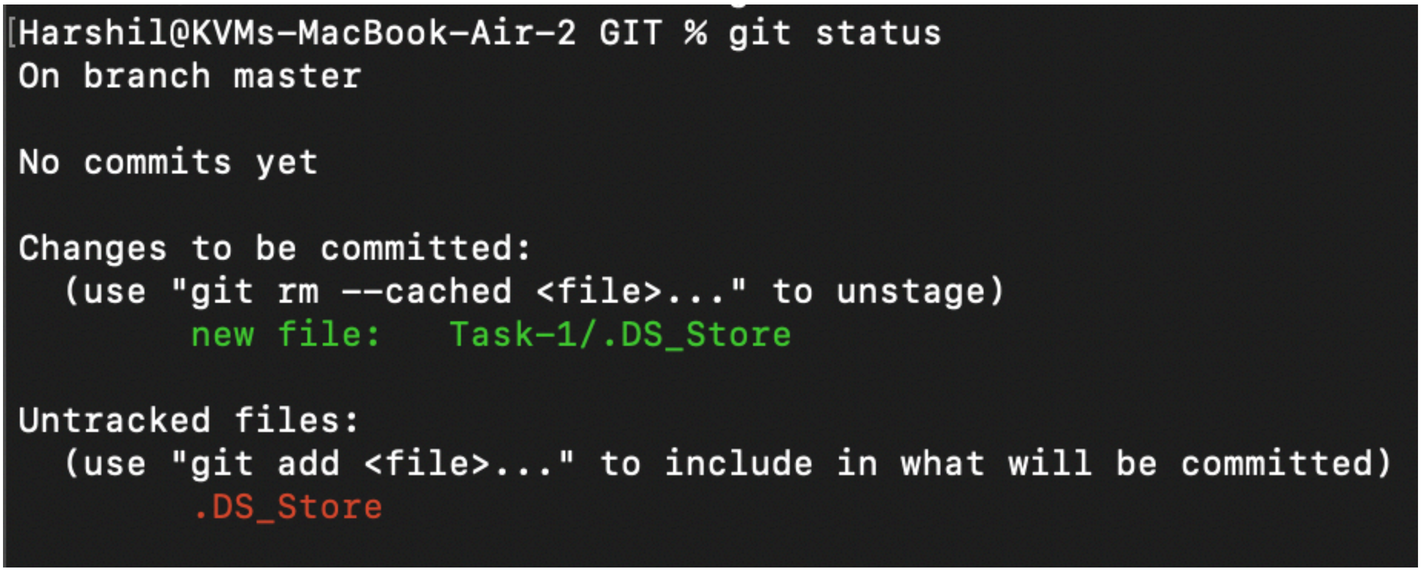
GIT Commands for Task 1

Git add



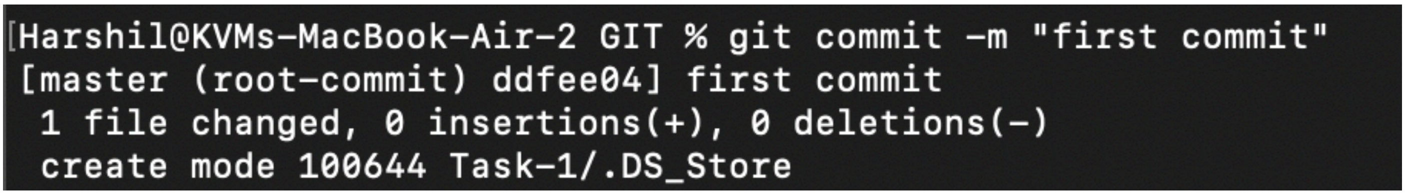
```
[Harshil@KVMs-MacBook-Air-2 GIT % git add Task-1/
```

Git status



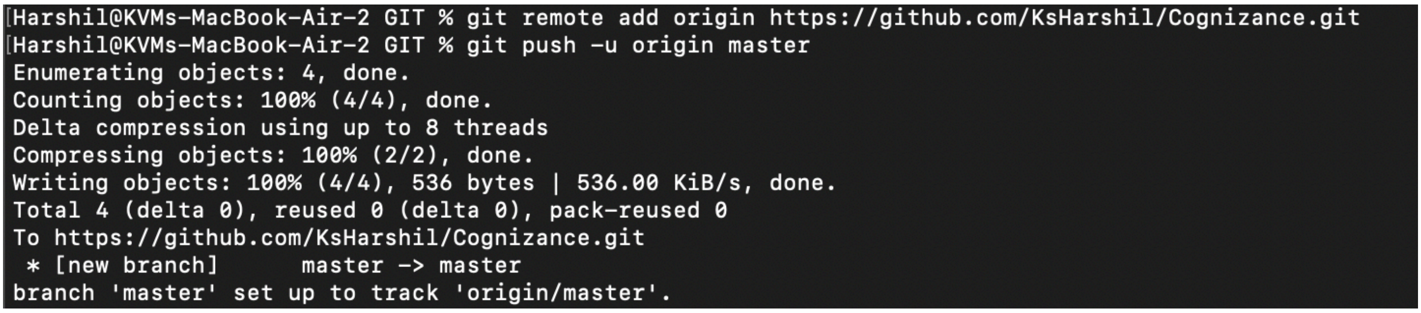
```
[Harshil@KVMs-MacBook-Air-2 GIT % git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Task-1/.DS_Store

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

Git commit



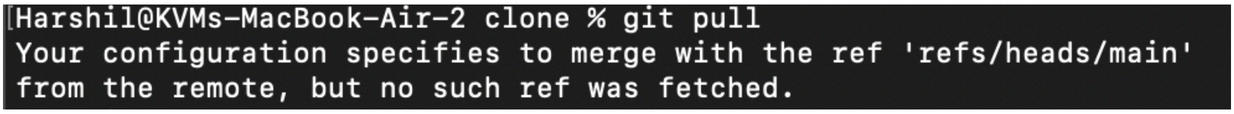
```
[Harshil@KVMs-MacBook-Air-2 GIT % git commit -m "first commit"
[master (root-commit) ddfee04] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Task-1/.DS_Store
```

Git push



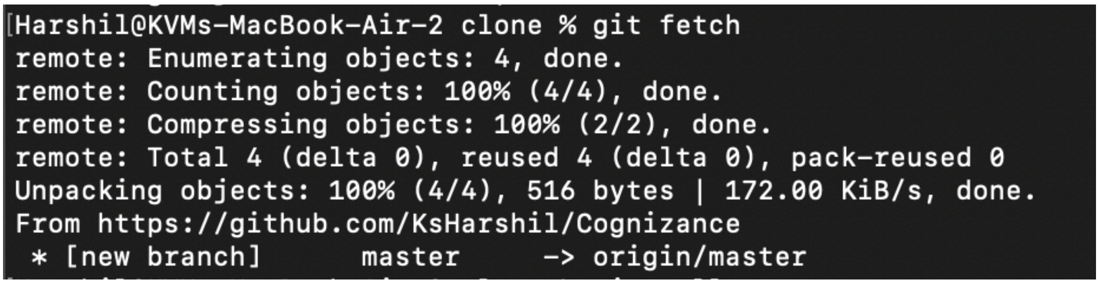
```
[Harshil@KVMs-MacBook-Air-2 GIT % git remote add origin https://github.com/KsHarshil/Cognizance.git
[Harshil@KVMs-MacBook-Air-2 GIT % git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 536 bytes | 536.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KsHarshil/Cognizance.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Git pull



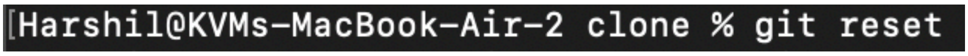
```
[Harshil@KVMs-MacBook-Air-2 clone % git pull
Your configuration specifies to merge with the ref 'refs/heads/main'
from the remote, but no such ref was fetched.
```

Git fetch



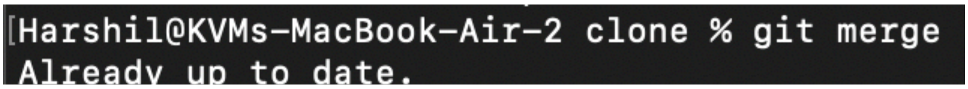
```
[Harshil@KVMs-MacBook-Air-2 clone % git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 516 bytes | 172.00 KiB/s, done.
From https://github.com/KsHarshil/Cognizance
 * [new branch]      master     -> origin/master
```

Git reset



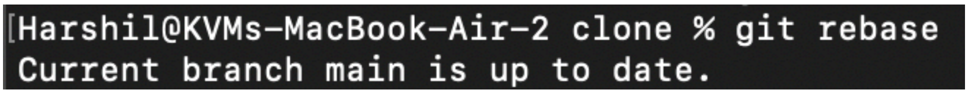
```
[Harshil@KVMs-MacBook-Air-2 clone % git reset
```

Git merge



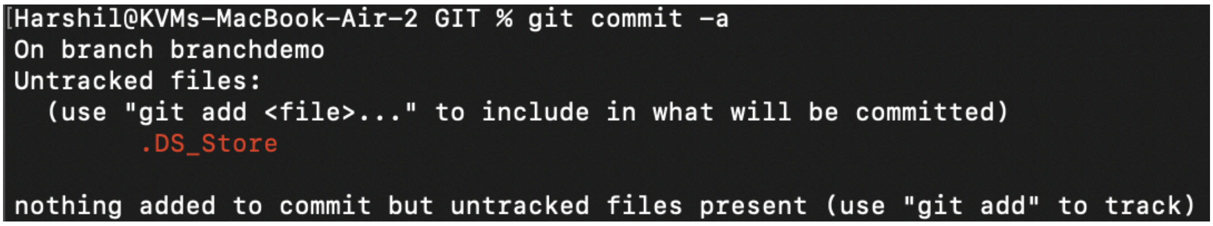
```
[Harshil@KVMs-MacBook-Air-2 clone % git merge
Already up to date.
```

Git rebase



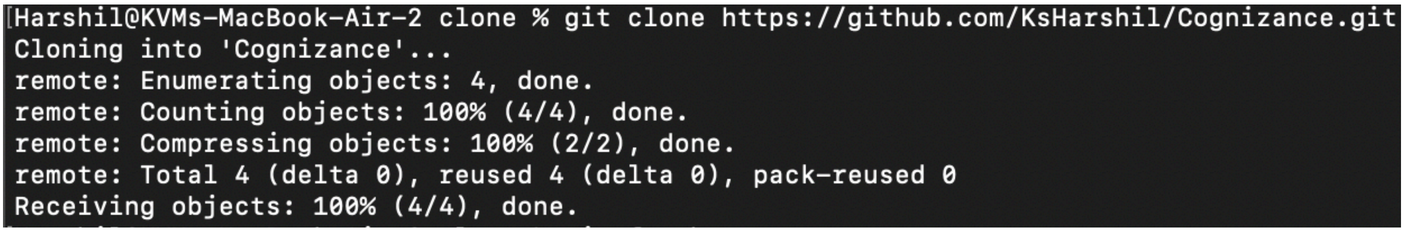
```
[Harshil@KVMs-MacBook-Air-2 clone % git rebase
Current branch main is up to date.
```

Git commit -a



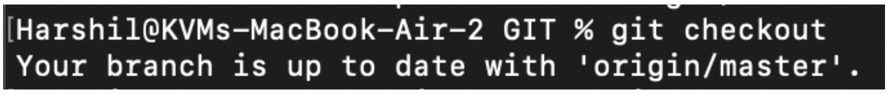
```
[Harshil@KVMs-MacBook-Air-2 GIT % git commit -a
On branch branchdemo
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store

nothing added to commit but untracked files present (use "git add" to track)
```

Git clone



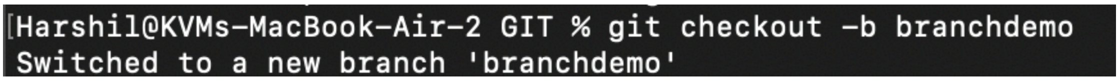
```
[Harshil@KVMs-MacBook-Air-2 clone % git clone https://github.com/KsHarshil/Cognizance.git
Cloning into 'Cognizance'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Git checkout



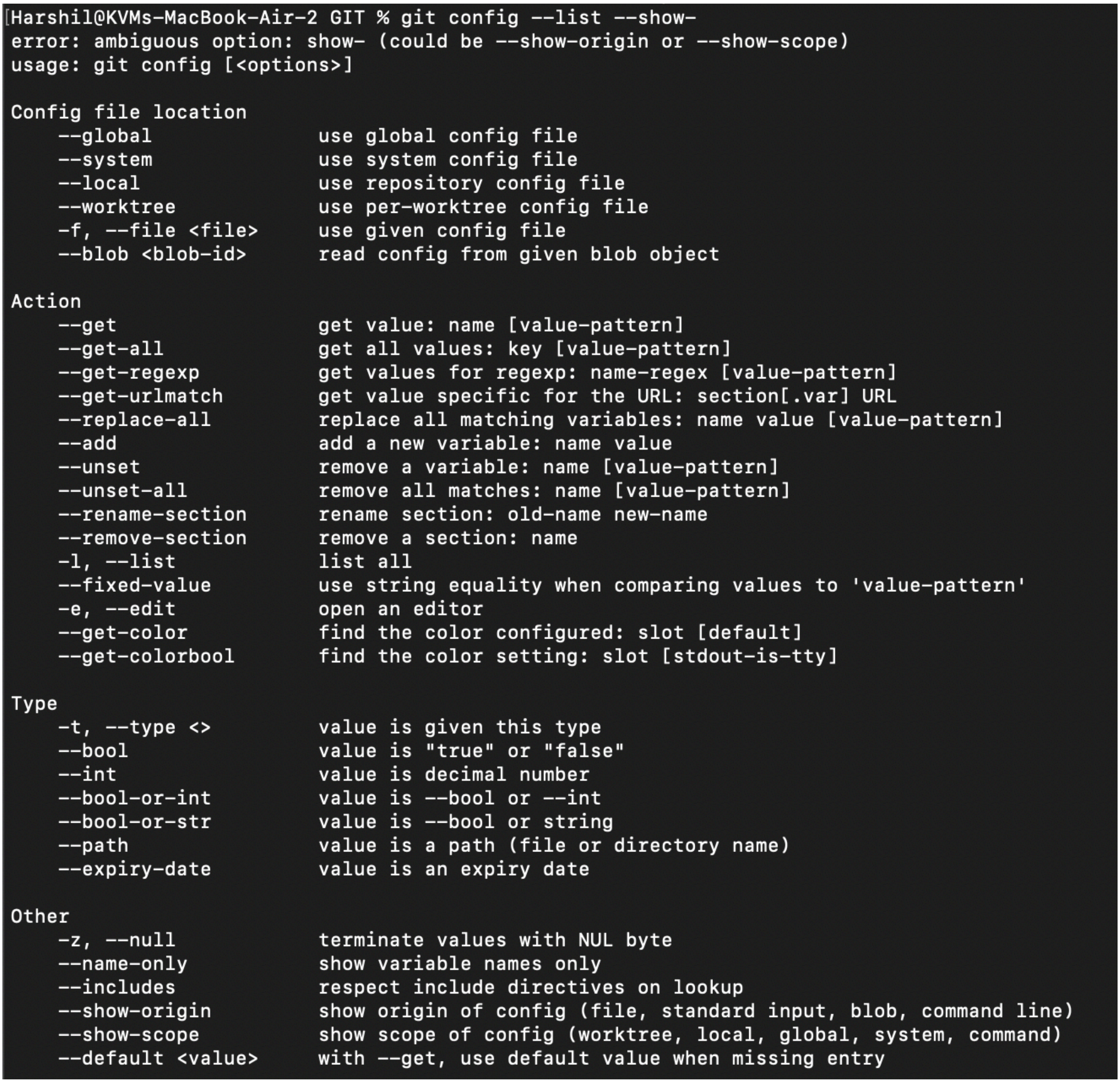
```
[Harshil@KVMs-MacBook-Air-2 GIT % git checkout
 Your branch is up to date with 'origin/master'.
```

Git checkout -b



```
[Harshil@KVMs-MacBook-Air-2 GIT % git checkout -b branchdemo
 Switched to a new branch 'branchdemo'
```

git config --list --show-



```
[Harshil@KVMs-MacBook-Air-2 GIT % git config --list --show-
 error: ambiguous option: show- (could be --show-origin or --show-scope)
 usage: git config [<options>]

 Config file location
     --global              use global config file
     --system              use system config file
     --local               use repository config file
     --worktree            use per-worktree config file
     -f, --file <file>     use given config file
     --blob <blob-id>      read config from given blob object

 Action
     --get                 get value: name [value-pattern]
     --get-all             get all values: key [value-pattern]
     --get-regexp          get values for regexp: name-regex [value-pattern]
     --get-urlmatch        get value specific for the URL: section[.var] URL
     --replace-all         replace all matching variables: name value [value-pattern]
     --add                 add a new variable: name value
     --unset               remove a variable: name [value-pattern]
     --unset-all           remove all matches: name [value-pattern]
     --rename-section      rename section: old-name new-name
     --remove-section      remove a section: name
     -l, --list            list all
     --fixed-value         use string equality when comparing values to 'value-pattern'
     -e, --edit            open an editor
     --get-color           find the color configured: slot [default]
     --get-colorbool       find the color setting: slot [stdout-is-tty]

 Type
     -t, --type <>         value is given this type
     --bool                value is "true" or "false"
     --int                 value is decimal number
     --bool-or-int         value is --bool or --int
     --bool-or-str         value is --bool or string
     --path                value is a path (file or directory name)
     --expiry-date         value is an expiry date

 Other
     -z, --null            terminate values with NUL byte
     --name-only           show variable names only
     --includes            respect include directives on lookup
     --show-origin         show origin of config (file, standard input, blob, command line)
     --show-scope          show scope of config (worktree, local, global, system, command)
     --default <value>     with --get, use default value when missing entry
```