

VM Settings

```
[root@main ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4
::1         localhost localhost.localdomain localhost6
192.168.142.140 main.example.com main
[root@main ~]#
[root@main ~]# ip a s ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 q
    link/ether 00:0c:29:49:ba:64 brd ff:ff:ff:ff:ff:ff
    inet 192.168.142.140/24 brd 192.168.142.255 scope 
        valid_lft 1469sec preferred_lft 1469sec
    inet6 fe80::20c:29ff:fe49:ba64/64 scope link
        valid_lft forever preferred_lft forever
[root@main ~]#
[root@main ~]# hostname
main.example.com
[root@main ~]#
[root@main ~]# dnsdomainname
example.com
_
```

Create Linux Users

```
[root@main ~]# groupadd devops
[root@main ~]#
[root@main ~]# useradd -G devops testuser
[root@main ~]# useradd aanya
[root@main ~]# useradd nanu
[root@main ~]#
[root@main ~]# passwd testuser
Changing password for user testuser.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
[root@main ~]#
[root@main ~]# passwd aanya
Changing password for user aanya.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
[root@main ~]#
[root@main ~]# passwd nanu
Changing password for user nanu.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
```

Git Remote Repo Create

```
[testuser@main ~]$ mkdir demo-project.git
[testuser@main ~]$
[testuser@main ~]$ cd demo-project.git/
[testuser@main demo-project.git]$
[testuser@main demo-project.git]$ ls -a
.
..
[testuser@main demo-project.git]$
[testuser@main demo-project.git]$ git --bare init
Initialized empty Git repository in /home/testuser/demo-project.git/
[testuser@main demo-project.git]$
[testuser@main demo-project.git]$ ls -a
.  ..  branches  config  description  HEAD  hooks  info  objects  refs
[testuser@main demo-project.git]$
[testuser@main demo-project.git]$ ls
branches  config  description  HEAD  hooks  info  objects  refs
[testuser@main demo-project.git]$
```

Generate Public/Private Keys

```
[aanya@main ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aanya/.ssh/id_rsa):
Created directory '/home/aanya/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aanya/.ssh/id_rsa.
Your public key has been saved in /home/aanya/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:CDvT7Kx22eKtPtmYe0vDdCLYHU318A0BGnzx+hFnD7E aanya@main.example.com
The key's randomart image is:
+---[RSA 2048]-----+
|      ...=oo..      |
|      oo.= o o      |
|      . o.. = E      |
|      =... . + o     |
|    o+.+.S . . .     |
|      o=+ . . .      |
|      =oX .          |
|      o.%.o          |
|      ..***.         |
+-----[SHA256]-----+
[aanya@main ~]$
[aanya@main ~]$ pwd
/home/aanya
[aanya@main ~]$ _
```

Generate Public/Private Keys

```
[aanya@main ~]$ ssh-copy-id -i .ssh/id_rsa.pub testuser@main.example.com
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_rsa.pub"
The authenticity of host 'main.example.com (192.168.142.140)' can't be established.
ECDSA key fingerprint is SHA256:Psp6PgImy0dgke19pb6bMMaYLcsVfx1UlHdpAftECyQ.
ECDSA key fingerprint is MD5:99:43:a0:54:b3:fc:40:cc:61:30:45:2f:0e:1a:6a:0c.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
lled
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now :
w keys
testuser@main.example.com's password:
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'testuser@main.example.com'"
and check to make sure that only the key(s) you wanted were added.

```
[aanya@main ~]$ _
```

Generate Public/Private Keys

```
[aanya@main ~]$ ssh testuser@main.example.com
Last login: Sun Jan 24 14:22:04 2021
[testuser@main ~]$
[testuser@main ~]$ logout
Connection to main.example.com closed.
[aanya@main ~]$
```

Generate Public/Private Keys

```
[nanu@main ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nanu/.ssh/id_rsa):
Created directory '/home/nanu/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nanu/.ssh/id_rsa.
Your public key has been saved in /home/nanu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:jPG8KVAAHxl16t1NFxUGBLaElU7ZkSUmivYSVECNOn8 nanu@main.example.com
The key's randomart image is:
+---[RSA 2048]---+
|  ..++..oB@0BB*=. |
|  ..o.o=+*+o+    |
|  ..+o oo .      |
|  .o 0 o..       |
|  ..+ S .        |
|  .. Eo          |
|  ..o            |
|  .              |
+---[SHA256]-----+
[nanu@main ~]$
[nanu@main ~]$ pwd
/home/nanu
```


Generate Public/Private Keys

```
[nanu@main ~]$ ssh-copy-id -i .ssh/id_rsa.pub testuser@main.example.com
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_rsa.pub"
The authenticity of host 'main.example.com (192.168.142.140)' can't be established.
ECDSA key fingerprint is SHA256:Psp6PgImy0dgke19pb6bMMaYLcsVfx1UlHdpAftECyQ.
ECDSA key fingerprint is MD5:99:43:a0:54:b3:fc:40:cc:61:30:45:2f:0e:1a:6a:0c.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
lled
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now :
w keys
testuser@main.example.com's password:
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'testuser@main.example.com'"
and check to make sure that only the key(s) you wanted were added.

```
[nanu@main ~]$ ssh testuser@main.example.com
Last login: Sun Jan 24 14:34:30 2021 from main.example.com
[testuser@main ~]$
[testuser@main ~]$ logout
Connection to main.example.com closed.
[nanu@main ~]$
```


Create Repo As "Aanya" User

```
[aanya@main ~]$ mkdir aanya-repo
[aanya@main ~]$
[aanya@main ~]$ cd aanya-repo/
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ echo "add content to README" >README
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git status -s
fatal: Not a git repository (or any of the parent directories): .git
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git init
Initialized empty Git repository in /home/aanya/aanya-repo/.git/
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git status -s
?? README
[aanya@main aanya-repo]$
```

Create Repo As “Aanya” User

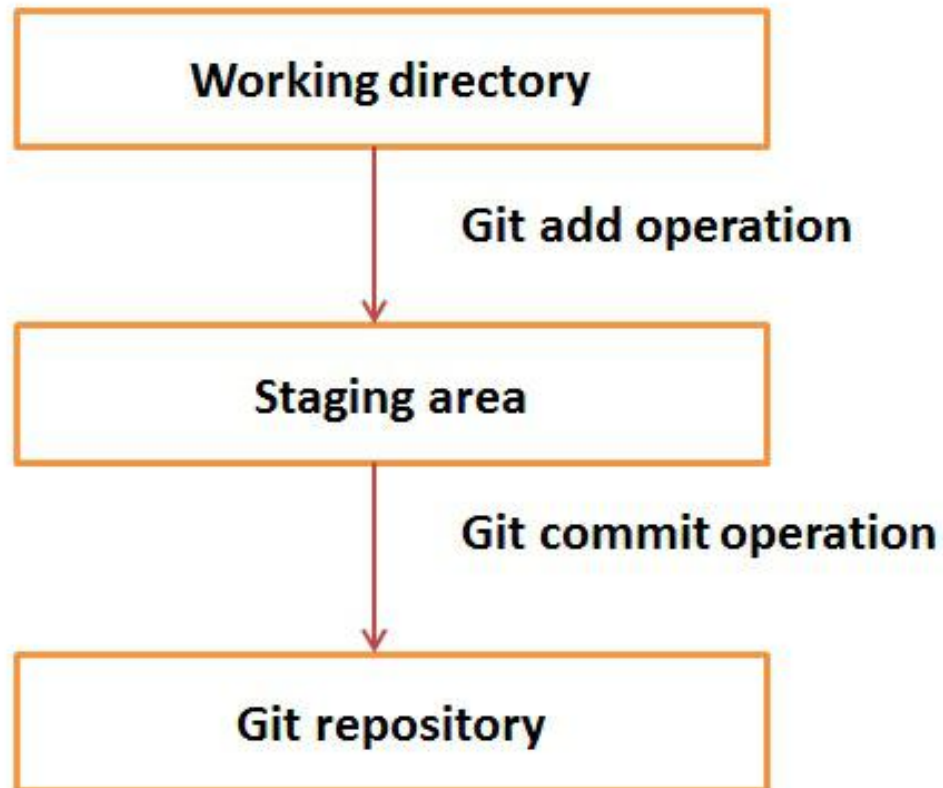
```
[aanya@main aanya-repo]$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git config --global user.name "aanya gupta"
[aanya@main aanya-repo]$ git config --global user.email "aanya@example.com"
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git config --list
user.name=aanya gupta
user.email=aanya@example.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
[aanya@main aanya-repo]$
```

Create Repo As “Aanya” User

```
[aanya@main aanya-repo]$ git add .
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git status -s
A README
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git commit -m "initial commit"
[master (root-commit) 7504fcad] initial commit
1 file changed, 1 insertion(+)
create mode 100644 README
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git log
commit 7504fcad8cc61180c084734430025542a878831d
Author: aanya gupta <aanya@example.com>
Date: Sun Jan 24 14:55:44 2021 +0530

    initial commit
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git status -s
[aanya@main aanya-repo]$
```

Working Directory & Staging Area



Working Directory & Staging Area



Local repo



Staging area

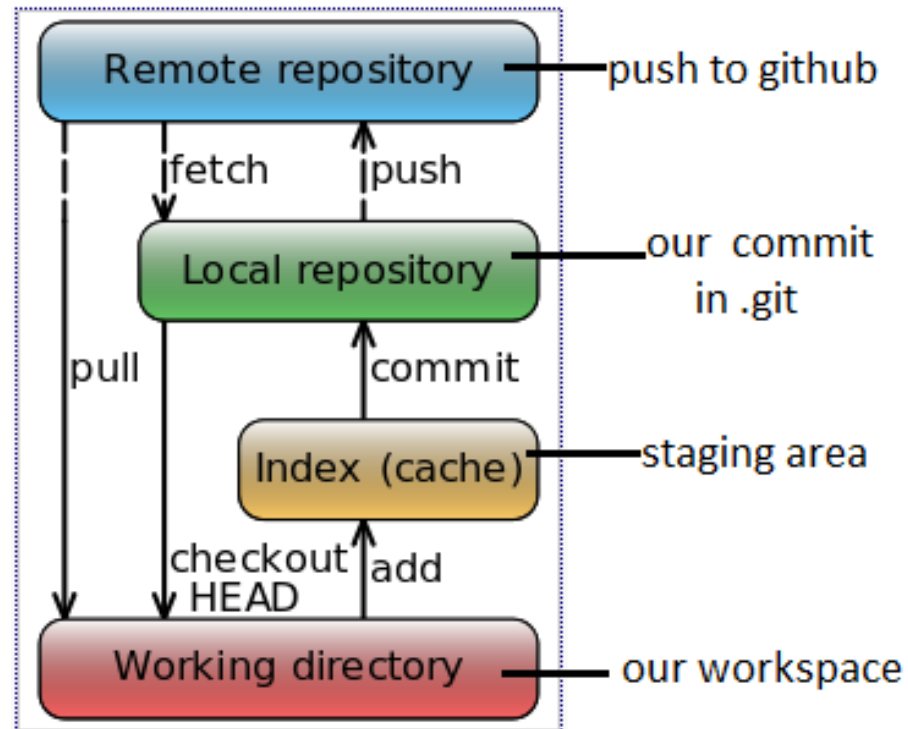


Working area

Push Repo To Remote Git Server

```
[aanya@main aanya-repo]$ git branch
* master
[aanya@main aanya-repo]$ git remote add origin testuser@main.example.com:demo-project.git
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 231 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To testuser@main.example.com:demo-project.git
 * [new branch]      master -> master
[aanya@main aanya-repo]$
```

Working Directory & Staging Area



Clone Repo As “Nanu” User

```
[nanu@main ~]$ mkdir nanu-repo
[nanu@main ~]$
[nanu@main ~]$ cd nanu-repo/
[nanu@main nanu-repo]$
[nanu@main nanu-repo]$ git clone testuser@main.example.com:demo-project.git
Cloning into 'demo-project'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
[nanu@main nanu-repo]$
[nanu@main nanu-repo]$ ls
demo-project
[nanu@main nanu-repo]$ cd demo-project/
[nanu@main demo-project]$
[nanu@main demo-project]$ ls
README
[nanu@main demo-project]$ cat README
add content to README
[nanu@main demo-project]$
```

Clone Repo As “Nanu” User

```
[nanu@main demo-project]$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=testuser@main.example.com:demo-project.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
[nanu@main demo-project]$
[nanu@main demo-project]$ git config --global user.name "nanu gupta"
[nanu@main demo-project]$
[nanu@main demo-project]$ git config --global user.email "nanu@example.com"
[nanu@main demo-project]$
[nanu@main demo-project]$ git config --list
user.name=nanu gupta
user.email=nanu@example.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=testuser@main.example.com:demo-project.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
[nanu@main demo-project]$
```

Clone Repo As “Nanu” User

```
[nanu@main demo-project]$ git branch
* master
[nanu@main demo-project]$
[nanu@main demo-project]$ git log
commit 7504fcad8cc61180c084734430025542a878831d
Author: aanya gupta <aanya@example.com>
Date:   Sun Jan 24 14:55:44 2021 +0530
```

initial commit

```
[nanu@main demo-project]$
[nanu@main demo-project]$ git diff
[nanu@main demo-project]$
[nanu@main demo-project]$ git show
commit 7504fcad8cc61180c084734430025542a878831d
Author: aanya gupta <aanya@example.com>
Date:   Sun Jan 24 14:55:44 2021 +0530
```

initial commit

```
diff --git a/README b/README
new file mode 100644
index 00000000..3457d3c
--- /dev/null
+++ b/README
@@ -0,0 +1 @@
+add content to README
[nanu@main demo-project]$ _
```

Add Files As “Nanu” User

```
[nanu@main demo-project]$ pwd
/home/nanu/nanu-repo/demo-project
[nanu@main demo-project]$
[nanu@main demo-project]$ cat >fruits.c
Apple
[nanu@main demo-project]$ cat fruits.c
Apple
[nanu@main demo-project]$
[nanu@main demo-project]$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       fruits.c
nothing added to commit but untracked files present (use "git add" to track)
[nanu@main demo-project]$
```

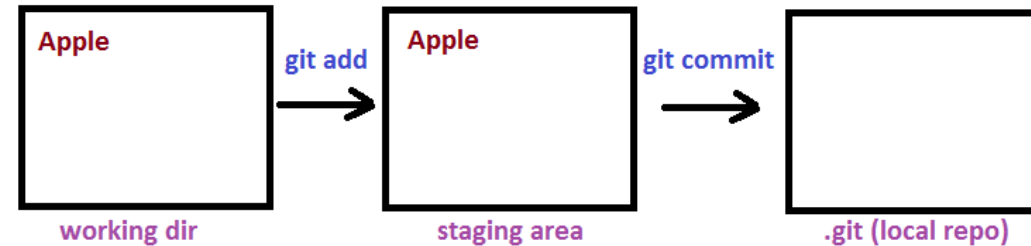
Snap shot of the file **fruits.c** in 3 different versions



Stage & Commit

```
[nanu@main demo-project]$ git add fruits.c
[nanu@main demo-project]$
[nanu@main demo-project]$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   fruits.c
#
[nanu@main demo-project]$
[nanu@main demo-project]$ git commit -m "created a file fruits.c"
[master f640743] created a file fruits.c
 1 file changed, 1 insertion(+)
 create mode 100644 fruits.c
[nanu@main demo-project]$
```

Snap shot of the file **fruits.c** in 3 different versions



Snap shot of the file **fruits.c** in 3 different versions



Stage & Commit

```
[nanu@main demo-project]$ git log
commit f640743b742443097a7e3c1fa9a68b336195ff0c
Author: nanu gupta <nanu@example.com>
Date:    Sun Jan 24 15:52:00 2021 +0530
```

created a file fruits.c

```
commit 7504fcad8cc61180c084734430025542a878831d
Author: aanya gupta <aanya@example.com>
Date:    Sun Jan 24 14:55:44 2021 +0530
```

initial commit

```
[nanu@main demo-project]$
[nanu@main demo-project]$ git log --oneline
f640743 created a file fruits.c
7504fca initial commit
[nanu@main demo-project]$
```

Stage & Commit

```
[nanu@main demo-project]$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#   (use "git push" to publish your local commits)
#
nothing to commit, working directory clean
[nanu@main demo-project]$
```


Add More Data To File

```
[nanu@main demo-project]$ cat fruits.c
Apple
[nanu@main demo-project]$ cat >>fruits.c
Pine Apple
[nanu@main demo-project]$ cat fruits.c
Apple
Pine Apple
[nanu@main demo-project]$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#   (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 checkout -- <file>..." to discard changes in working directory)
#
#       modified:   fruits.c
#
no changes added to commit (use "git add" and/or "git commit -a")
[nanu@main demo-project]$
```

Snap shot of the file **fruits.c** in 3 different versions



Snap shot of the file **fruits.c** in 3 different versions



Add File To Staging Area

```
[nanu@main demo-project]$ git add fruits.c
[nanu@main demo-project]$
[nanu@main demo-project]$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   fruits.c
#
[nanu@main demo-project]$
```

Snap shot of the file **fruits.c** in 3 different versions



Add More Data To File

```
[nanu@main demo-project]$ cat fruits.c
```

```
Apple
```

```
Pine Apple
```

```
[nanu@main demo-project]$ cat >>fruits.c
```

```
Custard Apple
```

```
[nanu@main demo-project]$ cat fruits.c
```

```
Apple
```

```
Pine Apple
```

```
Custard Apple
```

```
[nanu@main demo-project]$ git status
```

```
# On branch master
```

```
# Your branch is ahead of 'origin/master' by 1 commit.
```

```
#   (use "git push" to publish your local commits)
```

```
#
```

```
# Changes to be committed:
```

```
#   (use "git reset HEAD <file>..." to unstage)
```

```
#
```

```
#       modified:   fruits.c
```

```
#
```

```
# Changes not staged for commit:
```

```
#   (use "git add <file>..." to update what will be committed)
```

```
#   (use "git checkout -- <file>..." to discard changes in working directory)
```

```
#
```

```
#       modified:   fruits.c
```

```
#
```

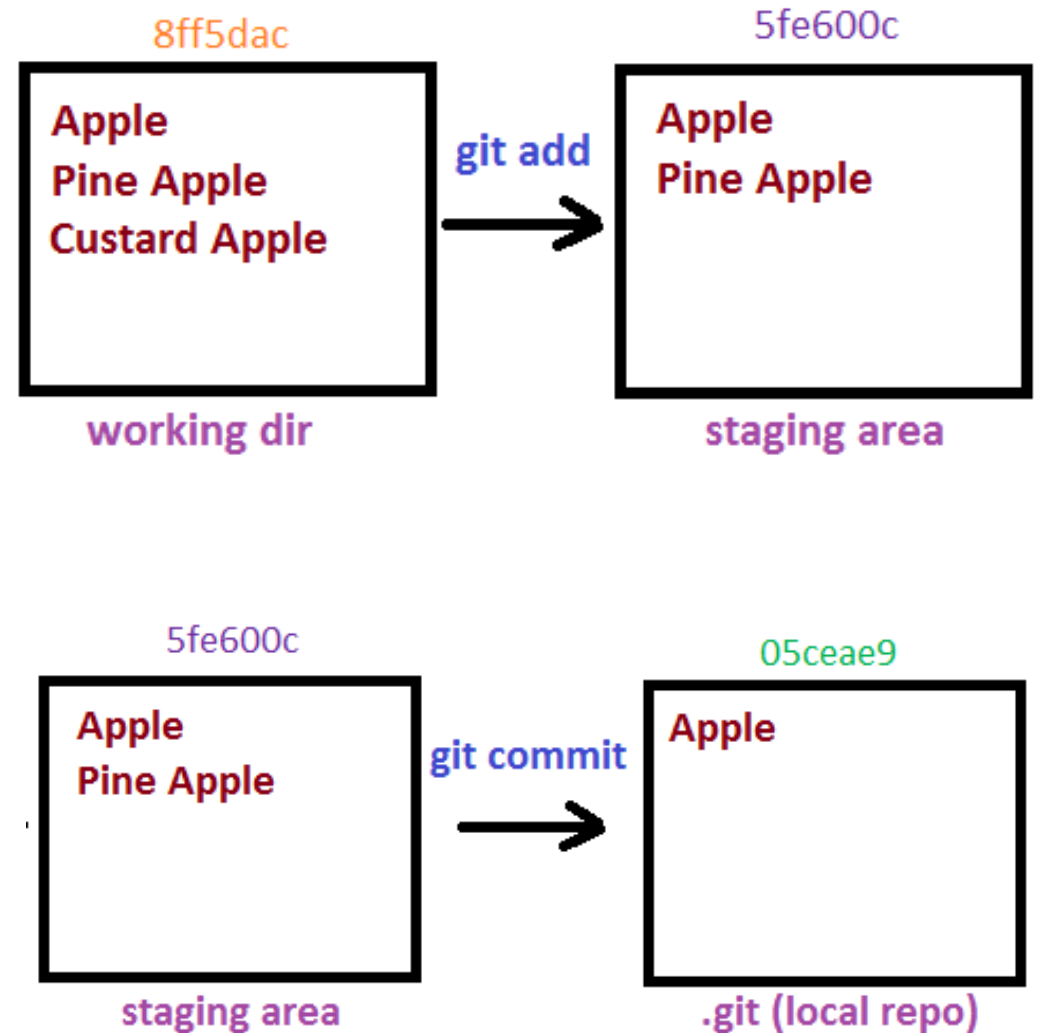
```
[nanu@main demo-project]$ _
```

Snap shot of the file **fruits.c** in 3 different versions



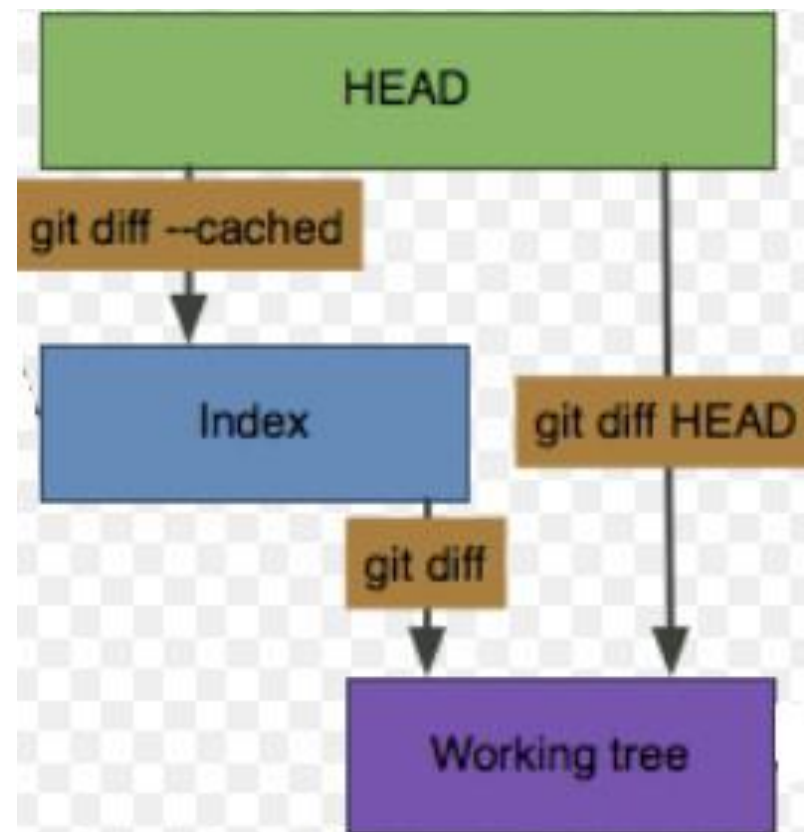
Git Diff

```
[nanu@main demo-project]$ git diff
diff --git a/fruits.c b/fruits.c
index 5fe600c..8ff5dac 100644
--- a/fruits.c
+++ b/fruits.c
@@ -1,2 +1,3 @@
 Apple
 Pine Apple
+Custard Apple
[nanu@main demo-project]$
[nanu@main demo-project]$ git diff --cached
diff --git a/fruits.c b/fruits.c
index 05ceae9..5fe600c 100644
--- a/fruits.c
+++ b/fruits.c
@@ -1 +1,2 @@
 Apple
+Pine Apple
[nanu@main demo-project]$
```



Git Diff

```
[nanu@main demo-project]$ git diff HEAD
diff --git a/fruits.c b/fruits.c
index 05ceae9..8ff5dac 100644
--- a/fruits.c
+++ b/fruits.c
@@ -1,3 @@
 Apple
+Pine Apple
+Custard Apple
[nanu@main demo-project]$
```



Git Push

```
[nanu@main demo-project]$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:
```

```
git config --global push.default matching
```

To squelch this message and adopt the new behavior now, use:

```
git config --global push.default simple
```

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

```
Counting objects: 4, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 281 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To testuser@main.example.com:demo-project.git
  7504fca..f640743  master -> master
[nanu@main demo-project]$
```

Git Status & Log

```
[nanu@main demo-project]$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   fruits.c
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   fruits.c
#
```

```
[nanu@main demo-project]$
[nanu@main demo-project]$ git log --oneline
f640743 created a file fruits.c
7504fca initial commit
[nanu@main demo-project]$
```

```
[nanu@main demo-project]$ git log --oneline
f640743 created a file fruits.c
7504fca initial commit
[nanu@main demo-project]$ git show f640743
commit f640743b742443097a7e3c1fa9a68b336195ff0c
Author: nanu gupta <nanu@example.com>
Date:   Sun Jan 24 15:52:00 2021 +0530
```

```
    created a file fruits.c
```

```
diff --git a/fruits.c b/fruits.c
new file mode 100644
index 00000000..05ceae9
--- /dev/null
+++ b/fruits.c
@@ -0,0 +1 @@
+Apple
[nanu@main demo-project]$
```


Git Pull

```
[aanya@main aanya-repo]$ git remote
origin
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ git remote -v
origin  testuser@main.example.com:demo-project.git (fetch)
origin  testuser@main.example.com:demo-project.git (push)
[aanya@main aanya-repo]$
```

Git Pull

```
[aanya@main aanya-repo]$ git pull origin master
From main.example.com:demo-project
 * branch                master      -> FETCH_HEAD
Updating 7504fca..f640743
Fast-forward
 fruits.c | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 fruits.c
[aanya@main aanya-repo]$
[aanya@main aanya-repo]$ ls
fruits.c  README
[aanya@main aanya-repo]$
```

Using Github

CentOS Linux 7 (Core)

Kernel 3.10.0-1160.6.1.el7.x86_64 on an x86_64

main login: vipin

Password:

[vipin@main ~]\$

[vipin@main ~]\$ mkdir vipin-repo

[vipin@main ~]\$

[vipin@main ~]\$ cd vipin-repo/

[vipin@main vipin-repo]\$

[vipin@main vipin-repo]\$ cat >colors.java

Red

[vipin@main vipin-repo]\$ git init

Initialized empty Git repository in /home/vipin/vipin-repo/.git/

[vipin@main vipin-repo]\$

[vipin@main vipin-repo]\$ git config --global user.name "vipin gupta"

[vipin@main vipin-repo]\$

[vipin@main vipin-repo]\$ git config --global user.email "vipin2411@gmail.com"

[vipin@main vipin-repo]\$

Using Github

```
[vipin@main vipin-repo]$ git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       colors.java
nothing added to commit but untracked files present (use "git add" to track)
[vipin@main vipin-repo]$
[vipin@main vipin-repo]$ git status -s
?? colors.java
[vipin@main vipin-repo]$ git add colors.java
[vipin@main vipin-repo]$
[vipin@main vipin-repo]$ git status -s
A  colors.java
[vipin@main vipin-repo]$
[vipin@main vipin-repo]$ git commit -m "first commit"
[master (root-commit) c406d72] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 colors.java
[vipin@main vipin-repo]$
```

Using Github

```
[vipin@main vipin-repo]$ git log  
commit c406d72f9e2e447eef7b84db77ce354986dfe015  
Author: vipin gupta <vipin2411@gmail.com>  
Date:   Sun Jan 24 18:14:18 2021 +0530
```

```
    first commit  
[vipin@main vipin-repo]$
```

Using Github

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *

Repository name *

 vipin2411 / demorepo 

Great repository names are short and memorable. Need inspiration? How about [glowing-fortnight](#)?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more](#).
- ☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more](#).
- ☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

Using Github

vipin2411 / demorepo

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop

or

HTTPS

SSH

<https://github.com/vipin2411/demorepo.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# demorepo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/vipin2411/demorepo.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/vipin2411/demorepo.git
git branch -M main
git push -u origin main
```


Using Github

```
[vipin@main vipin-repo]$ git branch
* master
[vipin@main vipin-repo]$
[vipin@main vipin-repo]$ git branch -M main
[vipin@main vipin-repo]$
[vipin@main vipin-repo]$ git branch
* main
[vipin@main vipin-repo]$
[vipin@main vipin-repo]$ git remote add origin https://github.com/vipin2411/demorepo.git
[vipin@main vipin-repo]$
[vipin@main vipin-repo]$ git push -u origin main
Username for 'https://github.com': vipin2411
Password for 'https://vipin2411@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 216 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/vipin2411/demorepo.git
 * [new branch]      main -> main
Branch main set up to track remote branch main from origin.
[vipin@main vipin-repo]$
```

Using Github

The screenshot shows a web browser window with the address bar displaying `https://github.com/vipin2411/demorepo`. The page content includes a light blue header with a navigation bar containing links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, there is a section for the repository `vipin2411 / demorepo`. This section includes a dropdown for the `main` branch, indicating 1 branch and 0 tags. To the right are buttons for 'Go to file', 'Add file', and a green 'Code' button. Below this, a commit history table shows a single commit by `vipin2411` titled 'first commit' with hash `c406d72`, made 1 hour ago. A file `colors.java` is listed as part of this commit. At the bottom, a light blue box prompts the user to 'Add a README' to help people understand the project.



Using the Hello World guide, you'll start a branch, write comments, and open a pull request

[Read the guide](#)

`vipin2411 / demorepo`

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

`main` 1 branch 0 tags [Go to file](#) [Add file](#) [Code](#)

 <code>vipin2411</code> first commit	<code>c406d72</code> 1 hour ago	🕒 1 commit
 <code>colors.java</code>	first commit	1 hour ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

Git Work Flow (Life Cycle)

