1.Creating a user name for current working directory of the git

```
login as: ec2-user
Authenticating with public key "linuxkey1"
       #_
  ~\_  ####_          Amazon Linux 2
 ~~  \_#####\
 ~~     \###|          AL2 End of Life is 2025-06-30.
 ~~      \#/ ___
  ~~      V~' '->
   ~~~         /       A newer version of Amazon Linux is available!
    ~~._.   _/
       _/ _/           Amazon Linux 2023, GA and supported until 2028-03-15.
     _/m/'                https://aws.amazon.com/linux/amazon-linux-2023/

7 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-29-167 ~]$ sudo -i
[root@ip-172-31-29-167 ~]# mkdir new
[root@ip-172-31-29-167 ~]# cd new
[root@ip-172-31-29-167 new]# git init
-bash: git: command not found
[root@ip-172-31-29-167 new]# yum intall git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No such command: intall. Please use /bin/yum --help
[root@ip-172-31-29-167 new]# yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package git.x86_64 0:2.40.1-1.amzn2.0.1 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.1 for package: git-2.40.1
-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.1 for package: git-2.
40.1-1.amzn2.0.1.x86_64
```
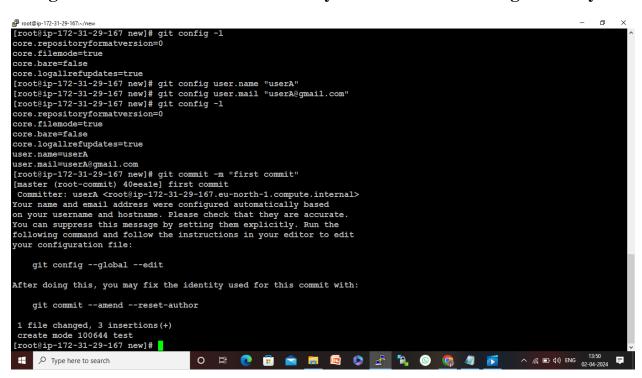
```
[root@ip-172-31-29-167 new]# nano test
[root@ip-172-31-29-167 new]# ls
test
[root@ip-172-31-29-167 new]# git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
[root@ip-172-31-29-167 new]# git add test
[root@ip-172-31-29-167 new]# git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   test

[root@ip-172-31-29-167 new]#
```
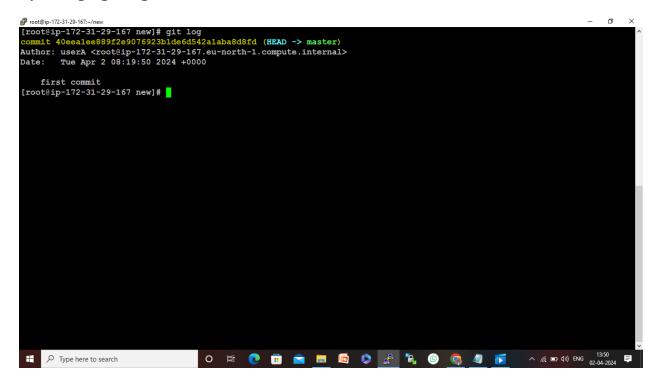
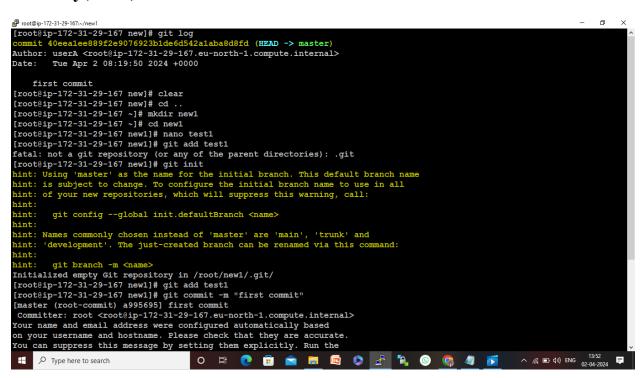**By using git rm –cached command to remove files from the staging area as untracked files.**

Y using "git config user.name " and " git config user.mail" command to configure user name and mail id for only to the current working directory.

First commit shows the configure user for current working directory(new). By using "git log" command.


```
[root@ip-172-31-29-167 new]# git log
commit 40eea1ee889f2e9076923b1de6d542a1aba8d8fd (HEAD -> master)
Author: userA <root@ip-172-31-29-167.eu-north-1.compute.internal>
Date:   Tue Apr 2 08:19:50 2024 +0000

    first commit
[root@ip-172-31-29-167 new]#
```

Creating another directory to check the commit changes by the new working directory(new1).


```
[root@ip-172-31-29-167 new]# git log
commit 40eea1ee889f2e9076923b1de6d542a1aba8d8fd (HEAD -> master)
Author: userA <root@ip-172-31-29-167.eu-north-1.compute.internal>
Date:   Tue Apr 2 08:19:50 2024 +0000

    first commit
[root@ip-172-31-29-167 new]# clear
[root@ip-172-31-29-167 new]# cd ..
[root@ip-172-31-29-167 ~]# mkdir new1
[root@ip-172-31-29-167 ~]# cd new1
[root@ip-172-31-29-167 new1]# nano test1
[root@ip-172-31-29-167 new1]# git add test1
fatal: not a git repository (or any of the parent directories): .git
[root@ip-172-31-29-167 new1]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/new1/.git/
[root@ip-172-31-29-167 new1]# git add test1
[root@ip-172-31-29-167 new1]# git commit -m "first commit"
[master (root-commit) a995695] first commit
 Committer: root <root@ip-172-31-29-167.eu-north-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
```

In the new1 working directory the first commit shows the root for the commit changes.

```
[root@ip-172-31-29-167 new1]# git config -l
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
[root@ip-172-31-29-167 new1]# cd ..
[root@ip-172-31-29-167 ~]# cd new
[root@ip-172-31-29-167 new]# git config -l
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
user.name=userA
user.mail=userA@gmail.com
[root@ip-172-31-29-167 new]# git config --global user.mail "userA@gmail.com"
[root@ip-172-31-29-167 new]# git config --global user.name "userA"
[root@ip-172-31-29-167 new]# git config -l
user.mail=userA@gmail.com
user.name=userA
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
user.name=userA
user.mail=userA@gmail.com
[root@ip-172-31-29-167 new]#
```
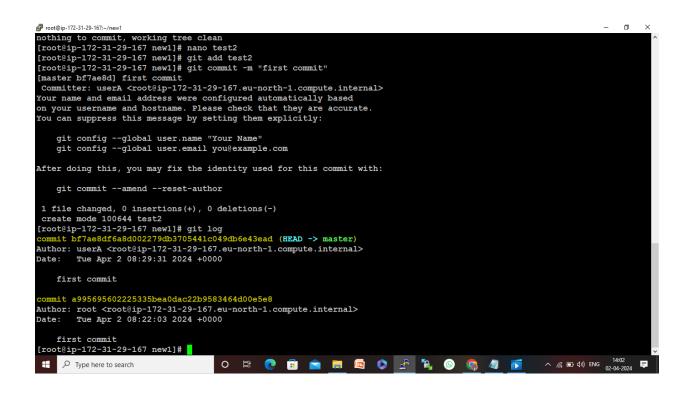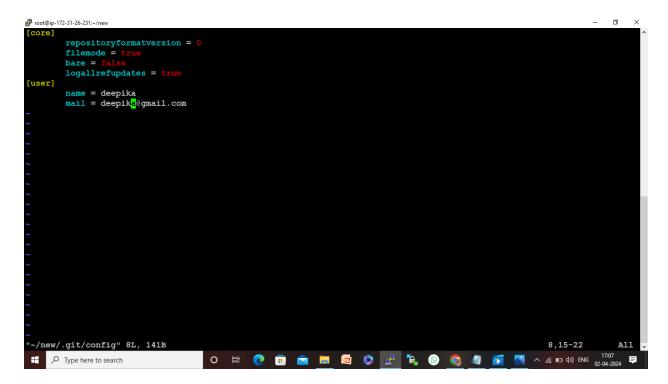
```
nothing to commit, working tree clean
[root@ip-172-31-29-167 new1]# nano test2
[root@ip-172-31-29-167 new1]# git add test2
[root@ip-172-31-29-167 new1]# git commit -m "first commit"
[master bf7ae8d] first commit
 Committer: userA <root@ip-172-31-29-167.eu-north-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

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

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test2
[root@ip-172-31-29-167 new1]# git log
commit bf7ae8df6a8d002279db3705441c049db6e43ead (HEAD -> master)
Author: userA <root@ip-172-31-29-167.eu-north-1.compute.internal>
Date:   Tue Apr 2 08:29:31 2024 +0000

    first commit

commit a995695602225335bea0dac22b9583464d00e5e8
Author: root <root@ip-172-31-29-167.eu-north-1.compute.internal>
Date:   Tue Apr 2 08:22:03 2024 +0000

    first commit
[root@ip-172-31-29-167 new1]#
```
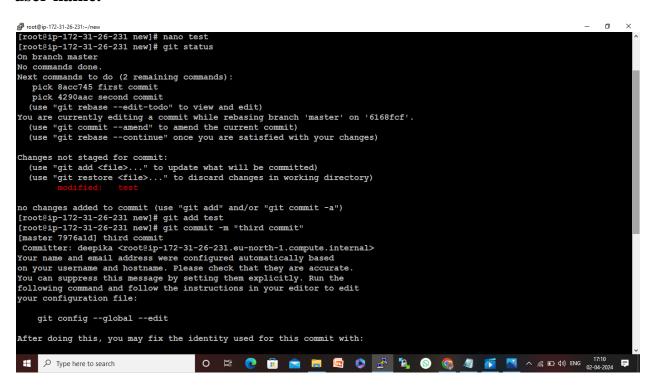
2.To edit or change the configured user name use the command "git config –edit"

Now the user name has been edited.

```
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[user]
        name = deepika
        mail = deepika@gmail.com
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"~/new/.git/config" 8L, 141B                                          8,15-22        All
```

Now make some changes in the source file to commit again to check the edited user name.

```
[root@ip-172-31-26-231 new]# nano test
[root@ip-172-31-26-231 new]# git status
On branch master
No commands done.
Next commands to do (2 remaining commands):
   pick 8acc745 first commit
   pick 4290aac second commit
  (use "git rebase --edit-todo" to view and edit)
You are currently editing a commit while rebasing branch 'master' on '6168fcf'.
  (use "git commit --amend" to amend the current commit)
  (use "git rebase --continue" once you are satisfied with your changes)

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:    test

no changes added to commit (use "git add" and/or "git commit -a")
[root@ip-172-31-26-231 new]# git add test
[root@ip-172-31-26-231 new]# git commit -m "third commit"
[master 7976a1d] third commit
 Committer: deepika <root@ip-172-31-26-231.eu-north-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:
```

Now the new commit shows the edited user name.