Before starting the workshop
Please install **Git** in your system and also create an account in **github.com**

In case of problem, take help with volunteers.

Workshop on **Git** and **GitHub**

AN INITIATION OF KUCC

What is Version Control

- Software for developers to collaborate and maintain history of their work
- Without version control, people often use folders which is highly inefficient

Why use this "Version Control"

- Allow developers to work simultaneously
- Do not allow overwriting each other's changes
- Maintains a history of each version
- Efficient and transparent (what, when and by whom)
- You can always Restore
- With Branching, you can experiment new features

Introduction to Git

- Git is a Version Control Software
- Designed by Linus Torvalds
- Written in C
- Free and Open source

Let's Get Started

Setting up Repository

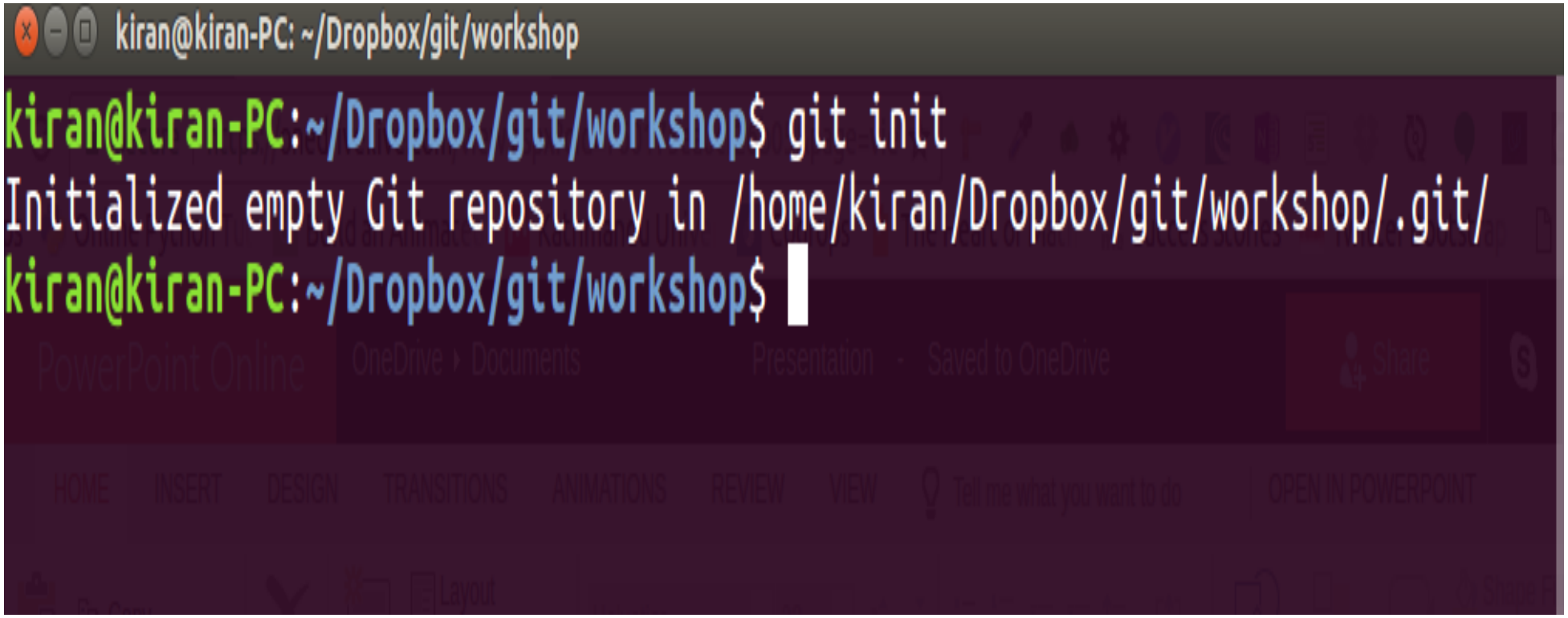
- Git init
- Git config

git init

git init command creates a new Git repository.

Usage

- git init



git config

Used to set Git configuration values

Usage

- git config user.email "email"
- git config user.name "name"

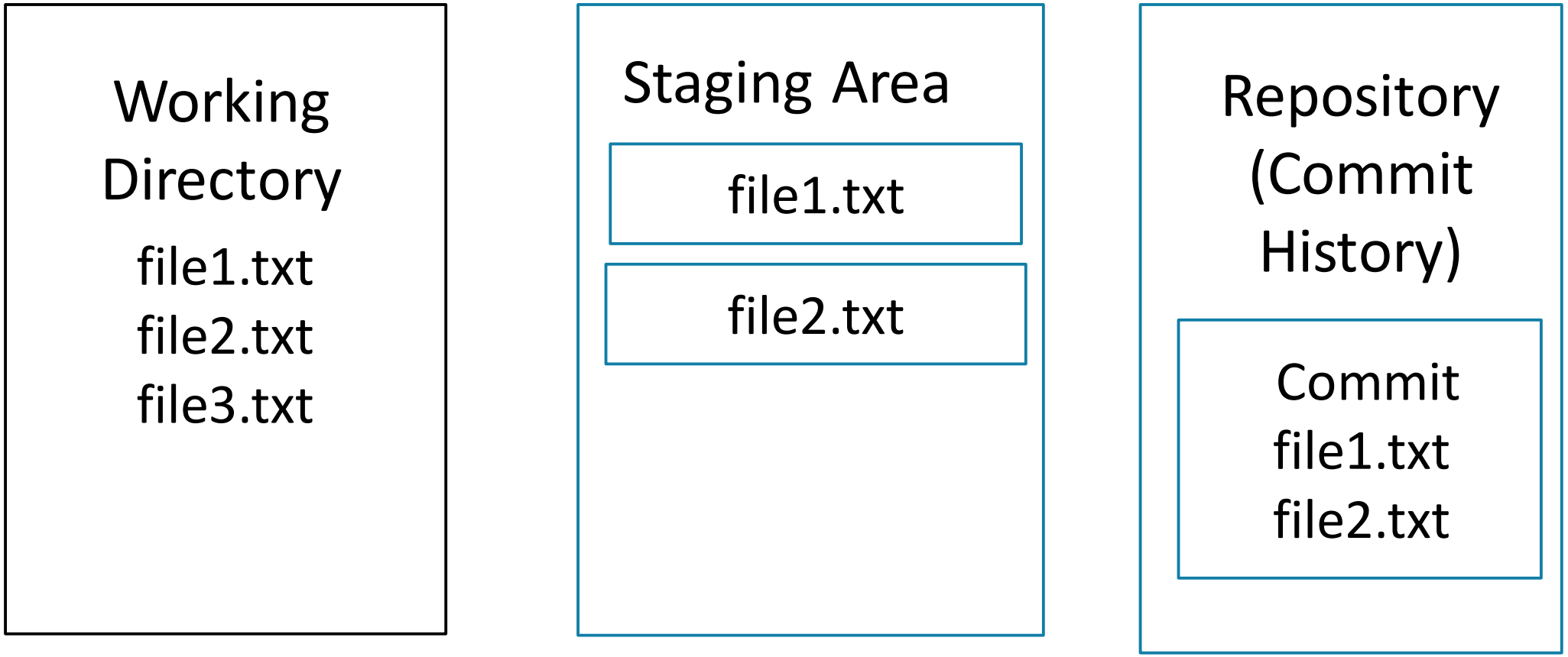
```
kiran@kiran-PC:~/Dropbox/git/workshop$ git config --global user.name "Kiran Kuma
r Chaudhary"
kiran@kiran-PC:~/Dropbox/git/workshop$ git config --global user.email "hellokira
n2020@gmail.com"
kiran@kiran-PC:~/Dropbox/git/workshop$
```

Saving Changes

git add

git commit

Choosing what changes to Commit



git add

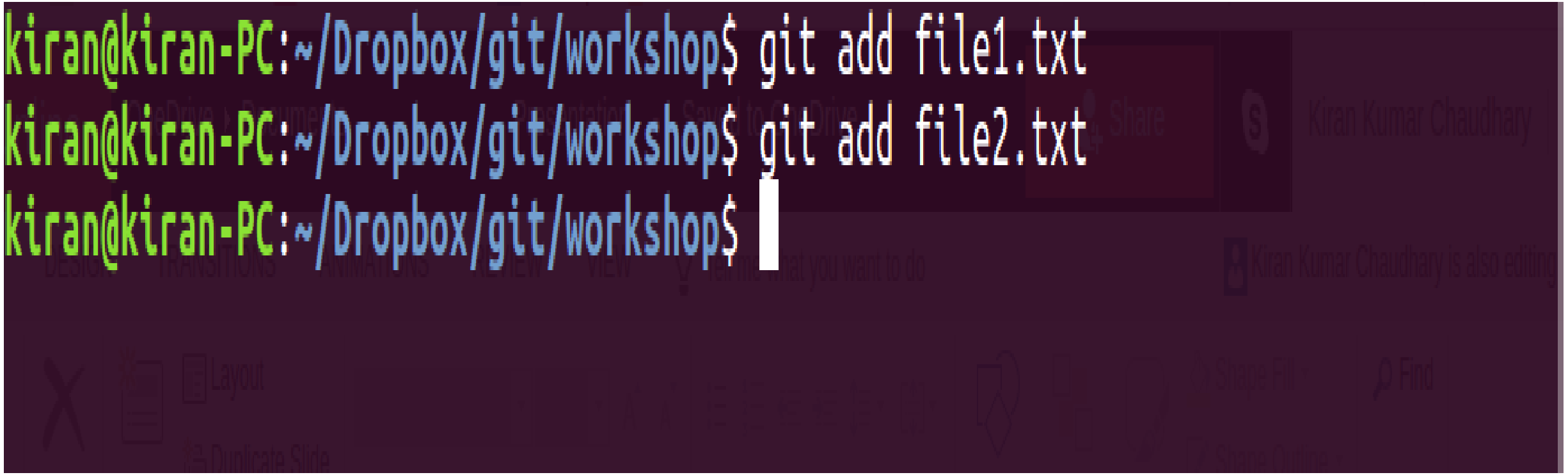
adds a change in the working directory to the staging area

doesn't really affect the repository

Usage

- git add <file>
- git add .

Let see in Action



git commit

commits the staged snapshot to the project history(repository)

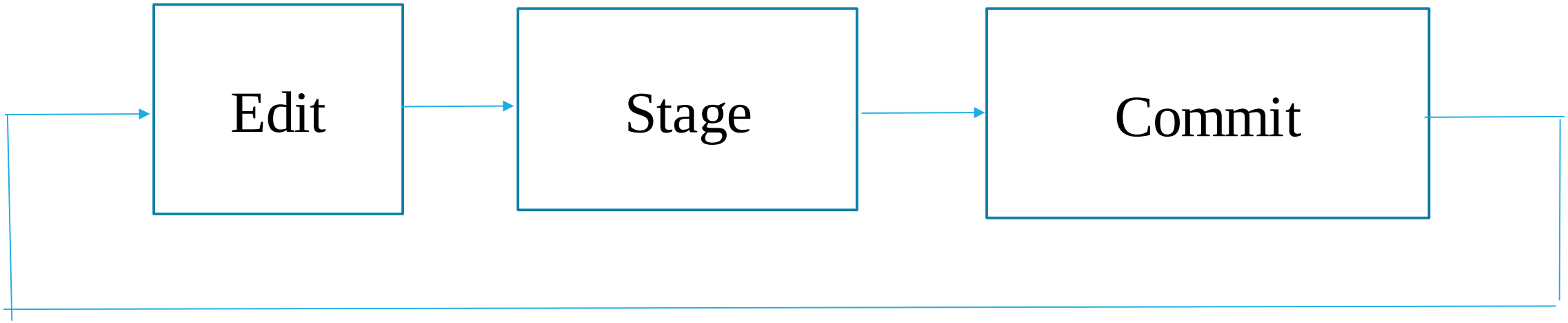
Usage

- git commit –m "commit message"

In Action

```
kiran@kiran-PC:~/Dropbox/git/workshop$ git commit -m "file1.txt and file2.txt ad
ded."
[master (root-commit) 2261a95] file1.txt and file2.txt added.
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
kiran@kiran-PC:~/Dropbox/git/workshop$
```

Workflow



Inspecting a Repository

git status

git log

git diff

git status

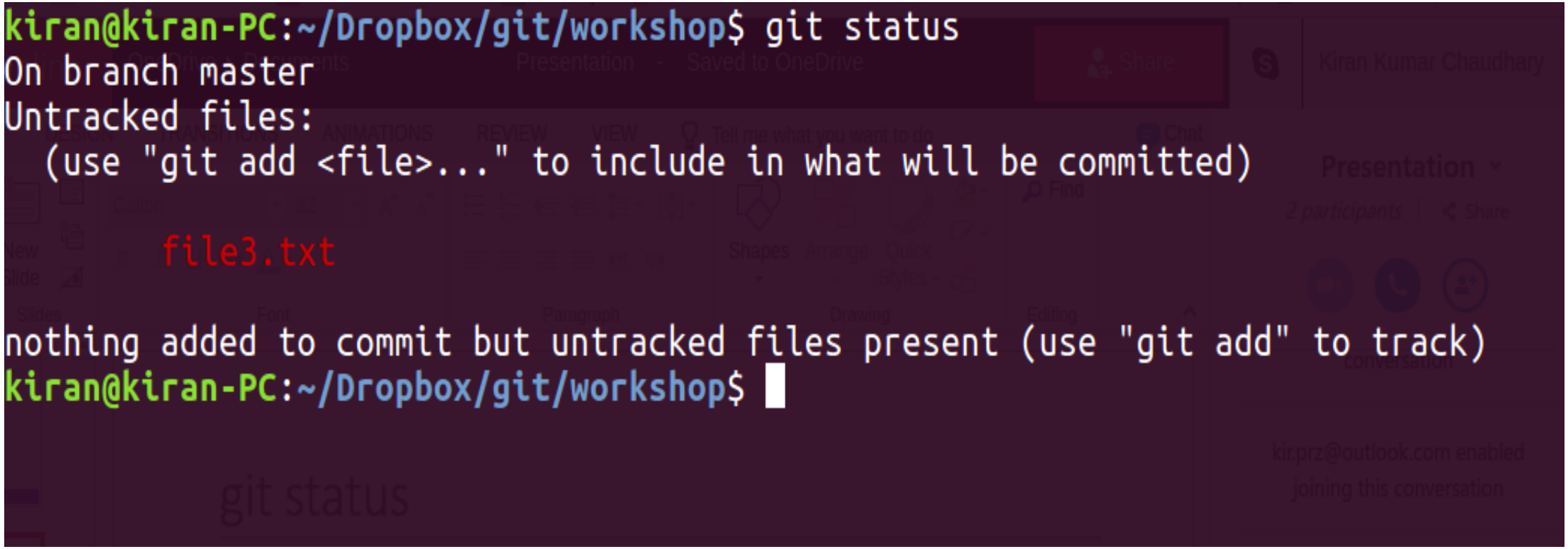
displays the state of the working directory and the staging area

see staged, unstaged, and untracked files by Git

Usage

- git status

In Action


kiran@kiran-PC:~/Dropbox/git/workshop$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file3.txt
nothing added to commit but untracked files present (use "git add" to track)
kiran@kiran-PC:~/Dropbox/git/workshop$

git log

displays committed snapshots

Usage

- git log

In Action

```
kiran@kiran-PC:~/Dropbox/git/workshop$ git log
commit c7bc18aa3713895686664f025cf6d9cc925c62d5e
Author: Kiran Kumar Chaudhary <hellokiran2020@gmail.com>
Date:   Thu Jun 1 23:18:34 2017 +0545

    line 1 added in file1.txt

commit 2261a95087bcb10f8e2dabaf6b11b85c3d6767ef
Author: Kiran Kumar Chaudhary <hellokiran2020@gmail.com>
Date:   Thu Jun 1 23:05:35 2017 +0545

    file1.txt and file2.txt added.
kiran@kiran-PC:~/Dropbox/git/workshop$
```

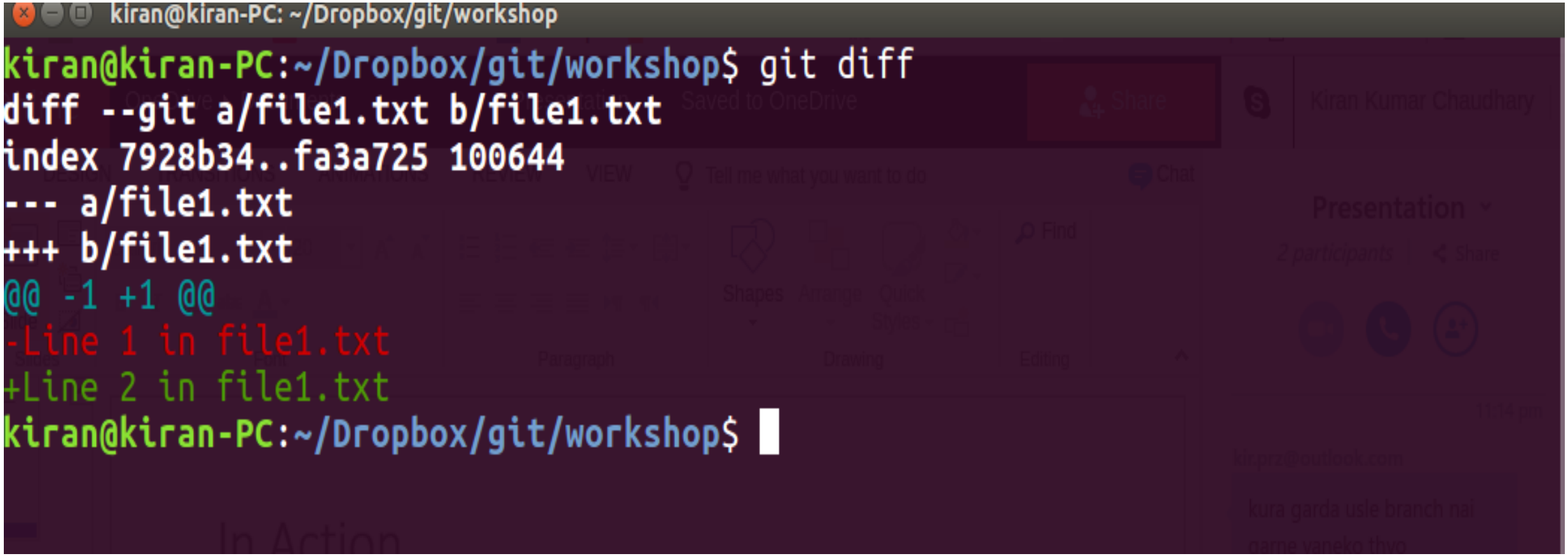
git diff

compare files a/b

Usage

- git diff

In Action

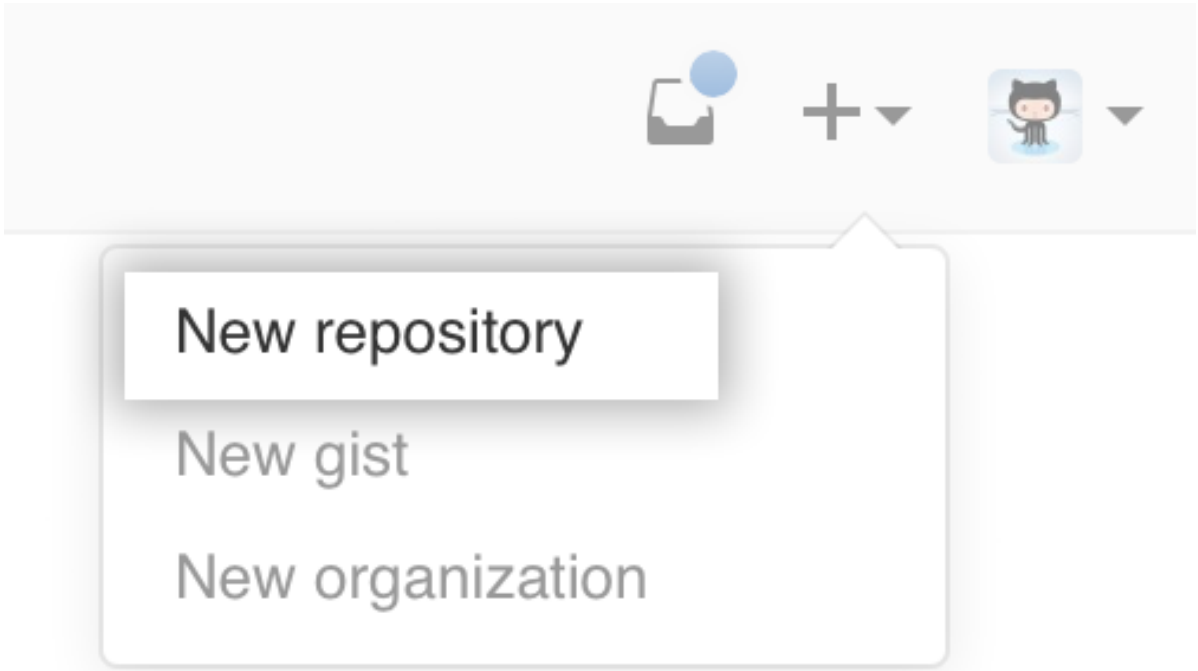

kiran@kiran-PC: ~/Dropbox/git/workshop
kiran@kiran-PC:~/Dropbox/git/workshop$ git diff
diff --git a/file1.txt b/file1.txt
index 7928b34..fa3a725 100644
--- a/file1.txt
+++ b/file1.txt
@@ -1 +1 @@
-Line 1 in file1.txt
+Line 2 in file1.txt
kiran@kiran-PC:~/Dropbox/git/workshop$

Let’s do some work

1. Change file2.txt. Add some line.
2. Run "git status" to see which file is modified.
3. Run "git diff" to see the difference between files.
4. Add the changes in Staging Area. Remember How?
5. Commit the Staged Snapshot in Commit history.

GitHub

creating repository



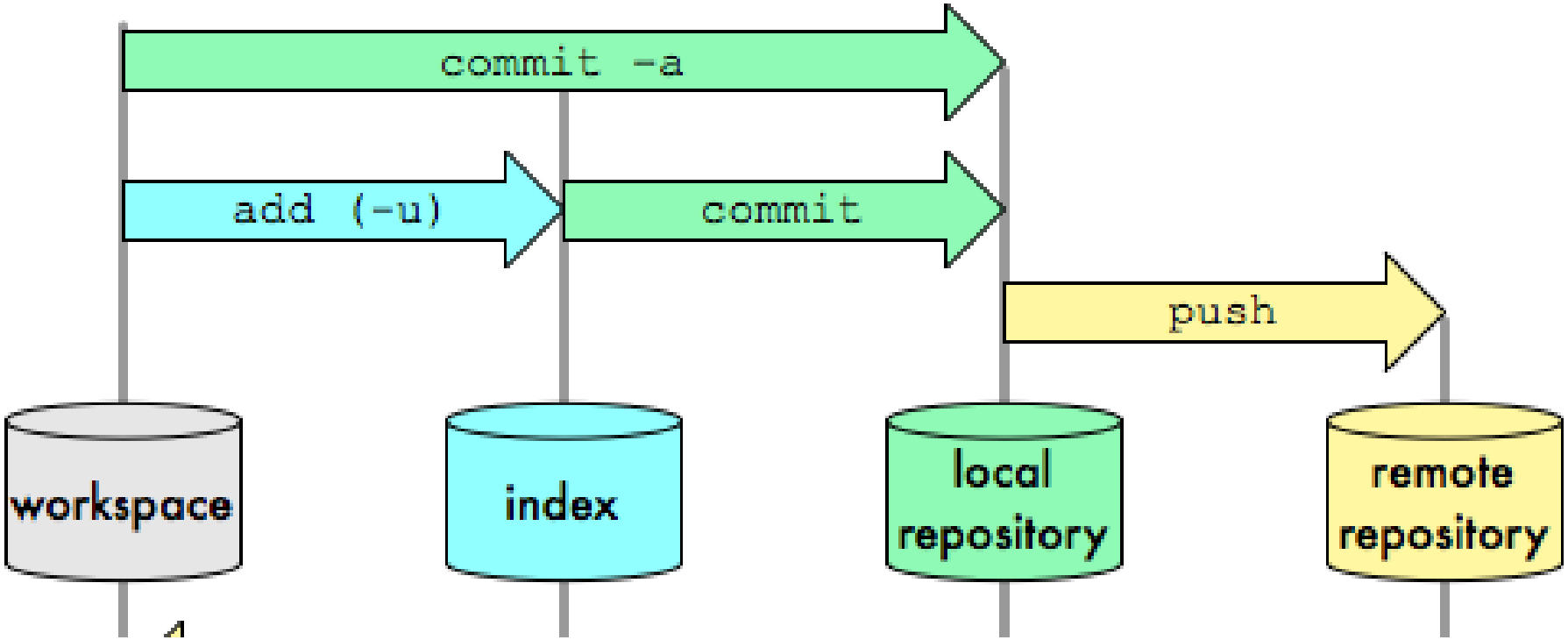
git push

transfer the last commit(s) to a remote server

Usage

- git init
- git add .
- git commit –m "Initial Commit"
- git remote add origin https://github.com/Kiran995/kucc.git
- git push –u origin branch_name

Workflow


commit -a
add (-u)
commit
push
workspace
index
local repository
remote repository

git clone

fetches all remote branches and creates one local branch

Usage

- git clone https://github.com/Kiran995/kucc.git

Git Branches