

INTRODUCTION TO GIT



Pavan Badarinath



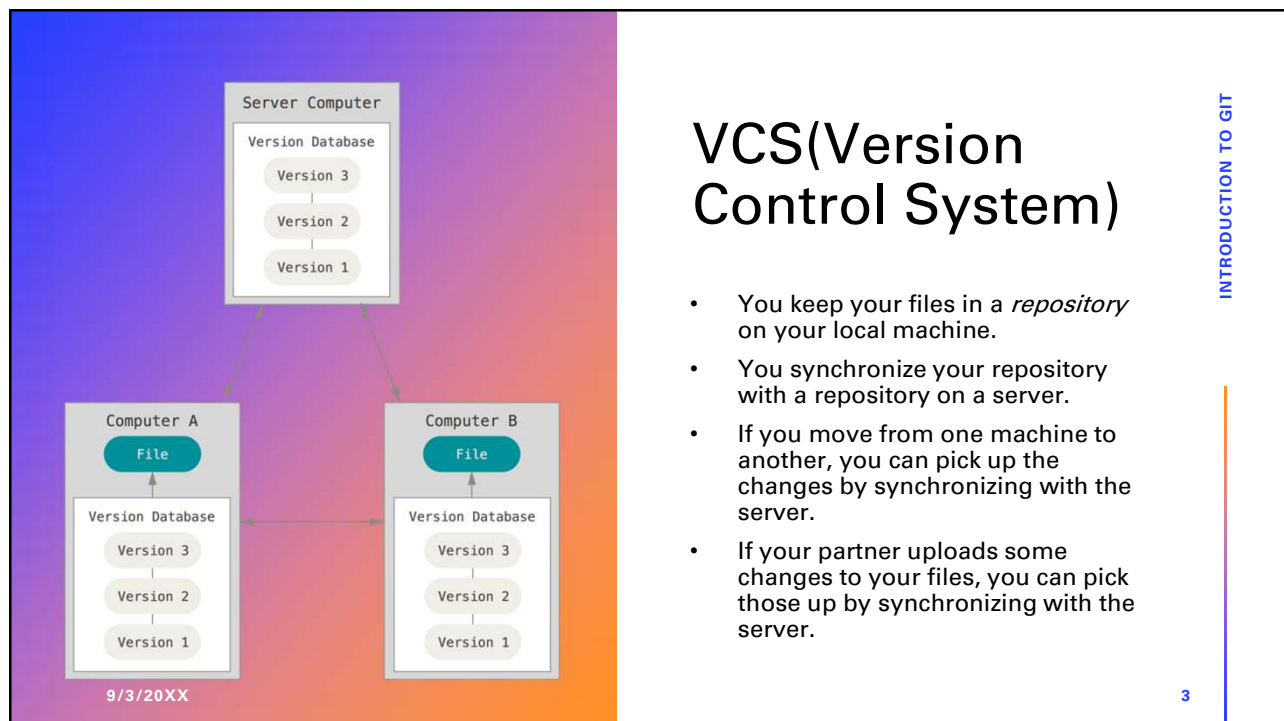
AGENDA

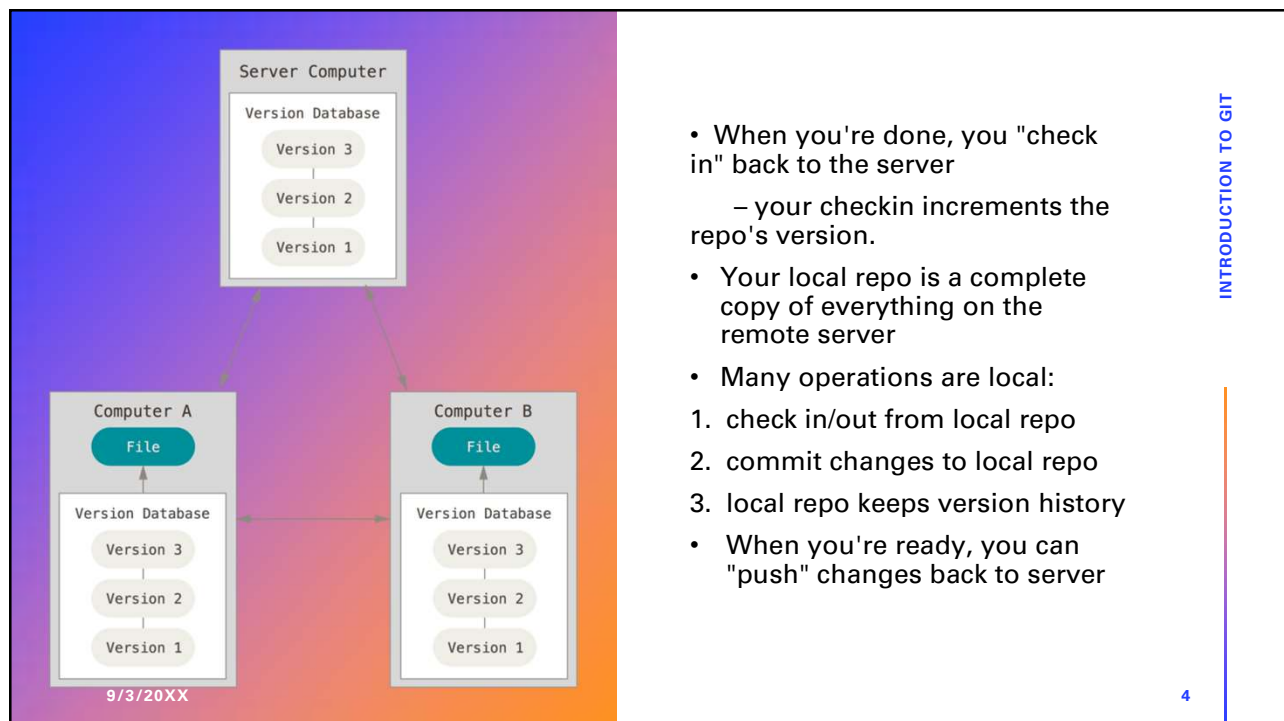
- VCS(Version Control System)
- What is GIT?
- GIT Installation
- Setting Up Repositories
- GIT most common commands
- GIT branching
- Merging

9/3/20XX

INTRODUCTION TO GIT

2





- When you're done, you "check in" back to the server
 - your checkin increments the repo's version.
- Your local repo is a complete copy of everything on the remote server
- Many operations are local:
 1. check in/out from local repo
 2. commit changes to local repo
 3. local repo keeps version history
- When you're ready, you can "push" changes back to server



- Git is free and open source distributed system with the emphasis on speed and data integrity.
- No centralized connectivity is needed.
- Powerful and cheap branching with easy to merge.
- Loosing work in your project is very hard.

6

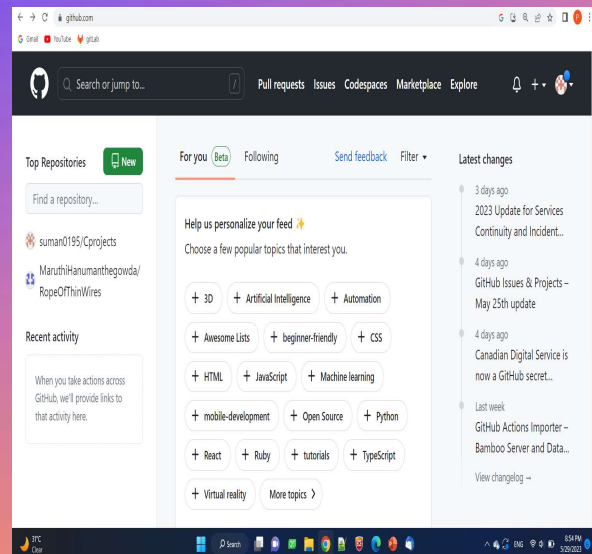
Git Installation:

- Git website: <http://git-scm.com/> : Download git here and install it.
- Free on-line book: <http://git-scm.com/book>
- Reference page for Git: <http://gitref.org/index.html>
- Git tutorial: <http://schacon.github.com/git/gittutorial.html>

7

CREATE GIT REPOSITORY:

- First Click on New button



- **GIVE A NAME TO REPOSITORY.**

WRITE DESCRIPTION FOR THE REPOSITORY.

CHOOSE THE ACCOUNT TYPE PRIVATE OR PUBLIC ACCORDING TO YOU.

CHECK THE ADD README FILE

-IT WILL CREATE ONE DEFAULT BRANCH CALLED MAIN.

CLICK ON CREATE REPOSITORY.

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:

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

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:

☒ **Add a README file**
 This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add license
[github templates: New](#)

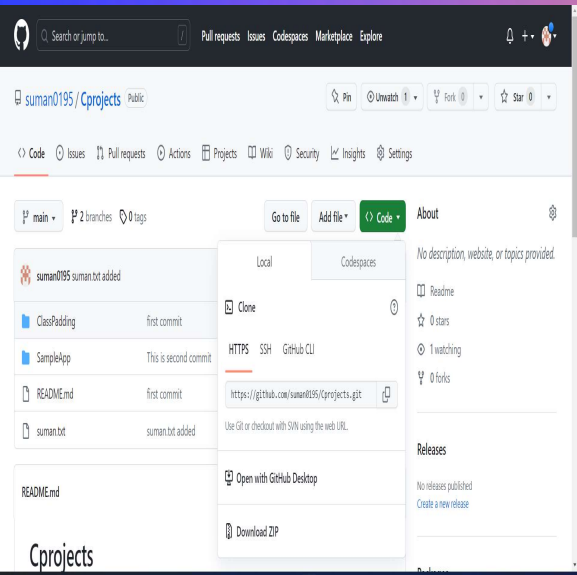
Choose which files to track from a list of templates. [Learn more about creating files.](#)

Choose a license
[License Name](#)

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

You are creating a public repository in your personal account.

Create repository

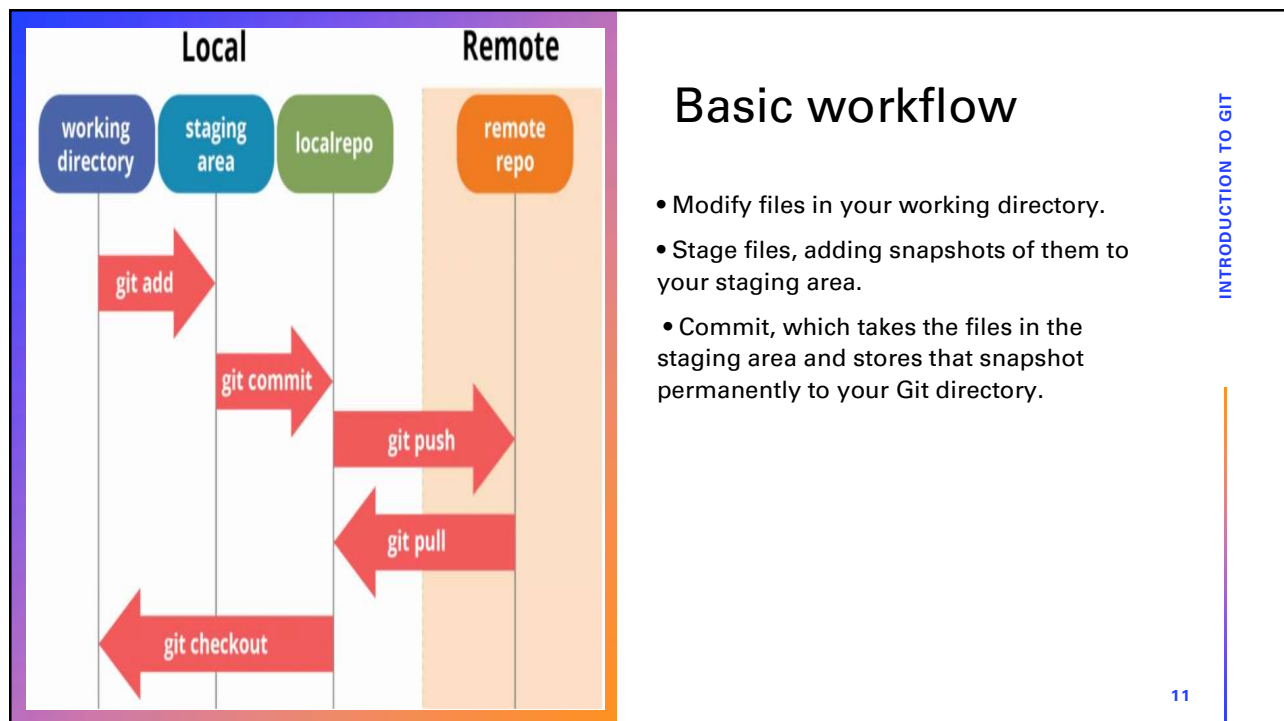


The screenshot shows the GitHub interface for the repository 'suman0195/Cprojects'. The 'Code' button is highlighted, and the dropdown menu is open, displaying the 'Clone' option with the HTTPS URL: `https://github.com/suman0195/Cprojects.git`. The repository's file list on the left includes 'suman0195 suman.txt added', 'ClassPadding', 'SampleApp', 'README.md', and 'suman.txt'.

1.First Click on code
2.Copy the link. following to the HTTPS tab

INTRODUCTION TO GIT

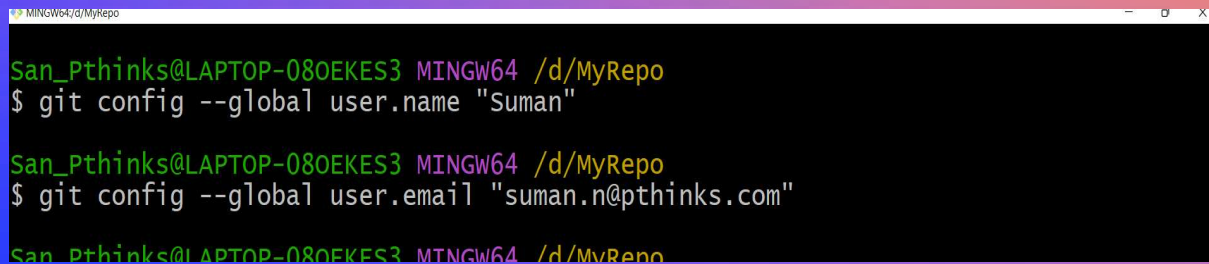
10



GIT COMMANDS

INTRODUCTION TO GIT

- set the name and email for git to use when you commit:
- `git config --global user.name "bugs bunny"`
- `git config --global user.email bugs@gmail.com`
- you can call `git config --list` to verify these are set.

A screenshot of a Windows terminal window titled "MINGW64/d/MyRepo". The prompt is "San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo". The user enters the command "\$ git config --global user.name 'Suman'", followed by "\$ git config --global user.email 'suman.n@pthinks.com'". The prompt is visible again at the bottom.

```
San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo
$ git config --global user.name "Suman"

San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo
$ git config --global user.email "suman.n@pthinks.com"

San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo
```

Git commands

- git clone url [dir]
 - copy a Git repository so you can add to it to an your local repository

MINGW64/d/MyRepo

```
San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo
$ git clone https://github.com/suman0195/Cprojects.git
Cloning into 'Cprojects'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 24 (delta 3), reused 18 (delta 1), pack-reused 0
Receiving objects: 100% (24/24), 6.81 KiB | 3.40 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

13

Git commands

- git add file
-adds file contents to the staging area

MINGW64/d:/MyRepo/Cprojects

```
San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo/Cprojects (main)  
$ git add test/
```

14

Git commands

- git status

-view the status of your files in the working directory and staging area.

```
San_Pthinks@LAPTOP-08OEKES3 MINGW64 /d/MyRepo/Cprojects (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   test/NSISExampleApplication1.class
        new file:   test/NSISExampleApplication1.java
        new file:   test/createInstaller1.nsi
        new file:   test/readme.txt

San_Pthinks@LAPTOP-08OEKES3 MINGW64 /d/MyRepo/Cprojects (main)
$ |
```

Git commands

- git commit
 - records a snapshot of the staging area.

MINGW64/d/MyRepo/Cprojects

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   new file:   test/NSISExampleApplication1.class
#   new file:   test/NSISExampleApplication1.java
#   new file:   test/createInstaller1.nsi
#   new file:   test/readme.txt
#
~
```

16

Git commands

- `git commit -m "message"`
-records a snapshot of the staging area.

MINGW64/d/MyRepo/Cprojects

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is up to date with 'origin/main'.
#
# Changes to be committed:
#   new file:   test/NSISExampleApplication1.class
#   new file:   test/NSISExampleApplication1.java
#   new file:   test/createInstaller1.nsi
#   new file:   test/readme.txt
#
~
```

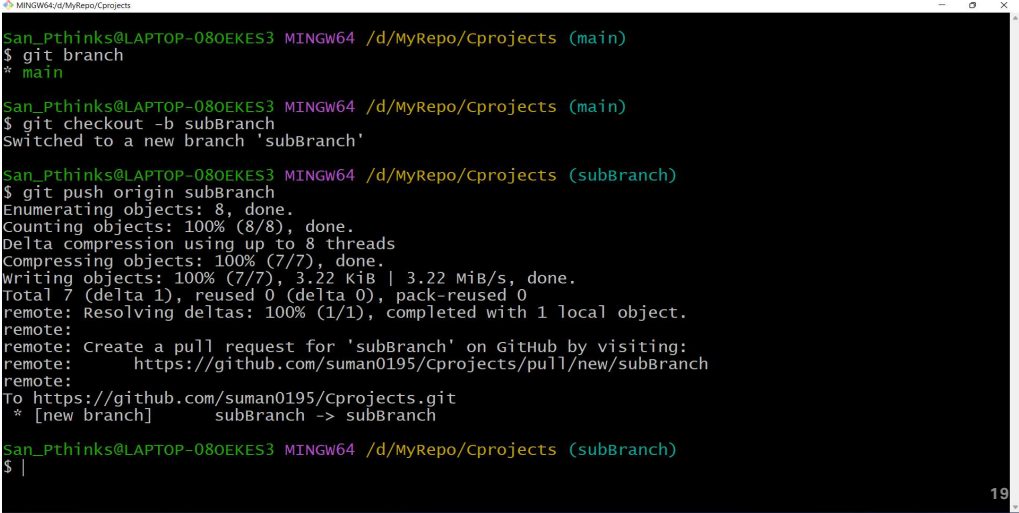
17

Branching

- To create a new local branch:
 - git branch name
- To list all local branches: (* = current branch)
 - git branch
- To switch to a given local branch
 - git checkout branchname
- To create and switch to a given local branch:
 - git checkout -b branchname
- To merge changes from a branch into the local master:
 - git checkout master
 - git merge branchname

18

- Push your local changes to the remote repo.
 - git push origin branchname



```
San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo/Cprojects (main)
$ git branch
* main

San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo/Cprojects (main)
$ git checkout -b subBranch
Switched to a new branch 'subBranch'

San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo/Cprojects (subBranch)
$ git push origin subBranch
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 3.22 KiB | 3.22 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'subBranch' on GitHub by visiting:
remote:   https://github.com/suman0195/cprojects/pull/new/subBranch
remote:
To https://github.com/suman0195/cprojects.git
 * [new branch]      subBranch -> subBranch

San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/MyRepo/Cprojects (subBranch)
$ |
```

- Pull from remote repo to get most recent changes. To fetch the most recent updates from the remote repo into your local repo, and put them into your working directory
 - git pull origin master

```

San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/new (main)
$ git pull origin main
From https://github.com/suman0195/Cprojects
* branch      main      -> FETCH_HEAD
14f4f8c..f44dc23  main    -> origin/main
Updating 14f4f8c..f44dc23
Fast-forward
 SampleApp/Sample.cpp      | 25 +++++
 SampleApp/Sample.h        | 25 +++++
 SampleApp/SampleApp.pro    | 25 +++++
 SampleApp/SampleApp.pro.user | 282 +++++++++++++++++++++++++++++++++++++
 SampleApp/main.cpp         | 25 +++++
 SampleApp/main.qml         | 17 +
 SampleApp/qml.qrc          | 5 +
 suman.txt                  | 1 +
 8 files changed, 405 insertions(+)
 create mode 100644 SampleApp/Sample.cpp
 create mode 100644 SampleApp/Sample.h
 create mode 100644 SampleApp/SampleApp.pro
 create mode 100644 SampleApp/SampleApp.pro.user
 create mode 100644 SampleApp/main.cpp
 create mode 100644 SampleApp/main.qml
 create mode 100644 SampleApp/qml.qrc
 create mode 100644 suman.txt
San_Pthinks@LAPTOP-080EKES3 MINGW64 /d/new (main)
$

```

20

