Git and GitHub Training

Remote Repositories

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

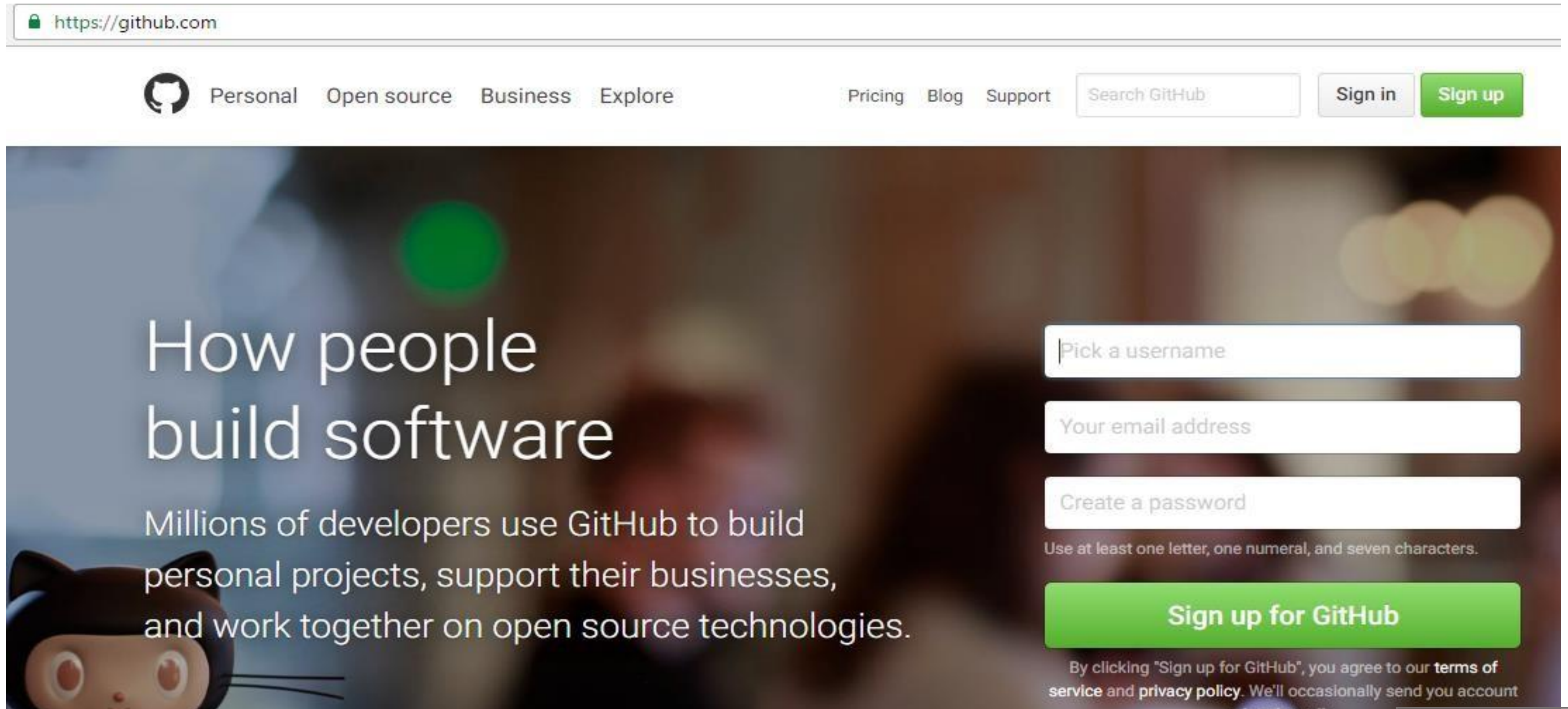
Learning Objectives

By the end of the lesson, you will be able to:

- Explain GitHub and repository using https and ssh
- Define fork, pull requests, and pulling commits
- Collaborate between a local and remote repository
- Implement multiple commits, merge file changes, and track issues
- Define upstream, downstream, and tags

Introduction to GitHub

GitHub: Basics



How people build software

Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.

Pick a username

Your email address

Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

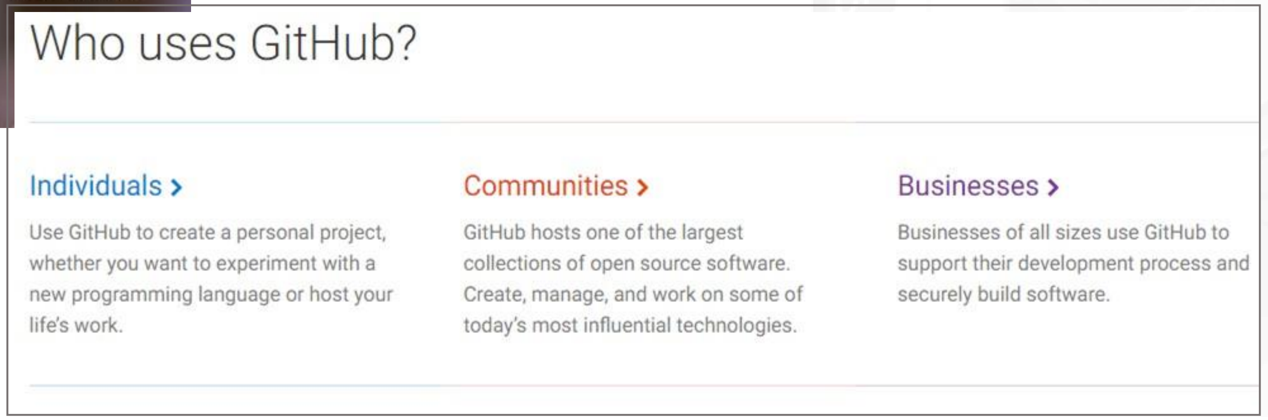
Transparent

Publish Project

Collaboration

Web-based Git repository hosting service

Web interface to upload files



Who uses GitHub?

Individuals ›

Use GitHub to create a personal project, whether you want to experiment with a new programming language or host your life's work.

Communities ›

GitHub hosts one of the largest collections of open source software. Create, manage, and work on some of today's most influential technologies.

Businesses ›

Businesses of all sizes use GitHub to support their development process and securely build software.

Getting Started with GitHub

UNASSISTED PRACTICE

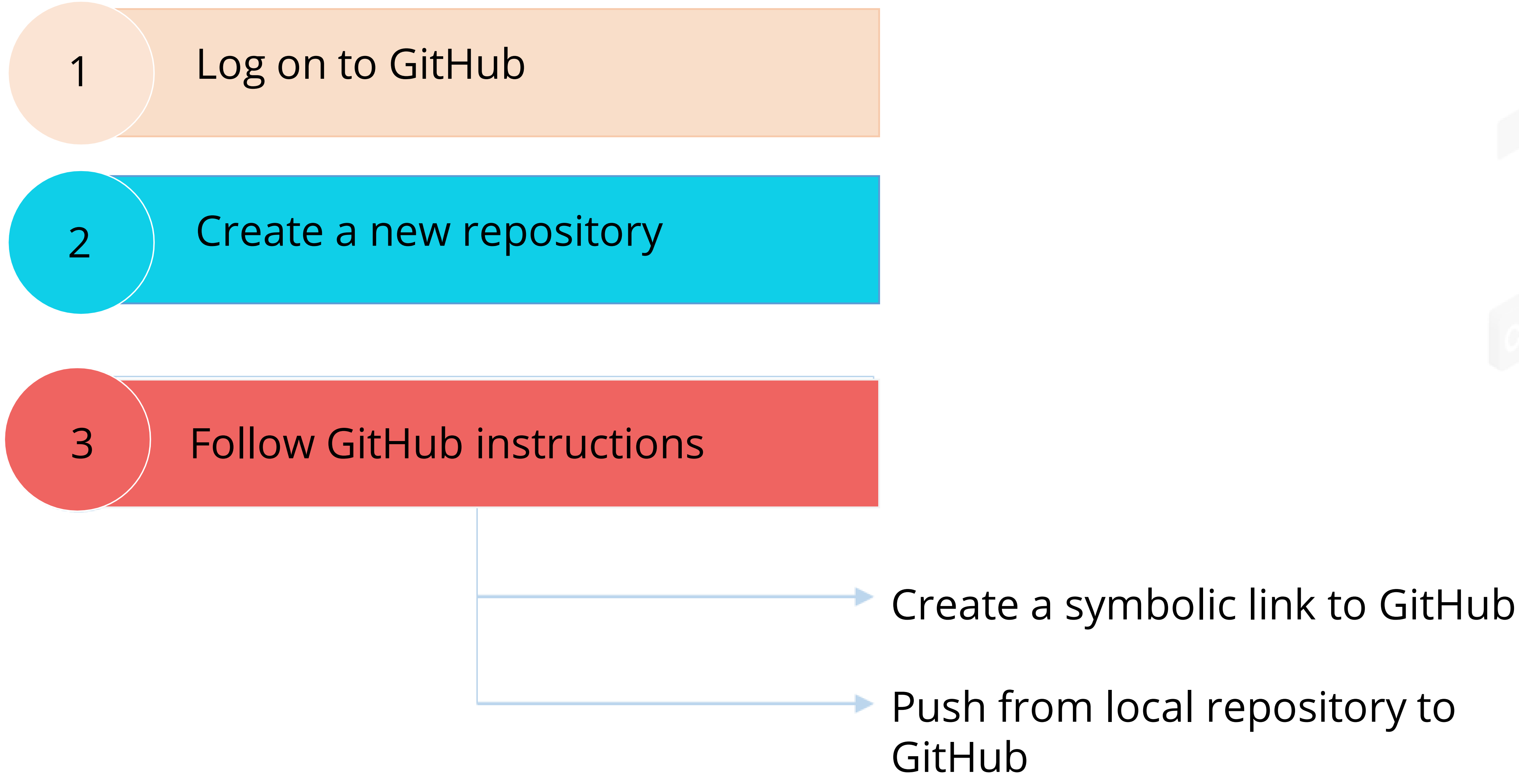
Problem Statement: Trainer has been assigned a class where all the learners are freshers. He has been suggested to do a walk through on day one of the class.

Steps to Perform:

1. Open the website
2. Select the Personal tab
3. Select the Open source tab
4. Select the Business tab
5. Select the Explore tab

GitHub: Using HTTPS

The steps to create a repository in GitHub using HTTPS are:



1 Log on to GitHub

2 Create a new repository

3 Follow GitHub instructions

- Create a symbolic link to GitHub
- Push from local repository to GitHub

Create a Repository in GitHub using HTTPS

Problem Statement: Create public repositories for an open source project. When creating your public repository, make sure to use a credential helper so Git will remember your GitHub username and password every time it talks to GitHub

Steps to Perform:

1. Execute $ git status to fetch the status
2. Execute $ ls -lrt
3. Execute $ git log --online
4. Create a repository
5. Add the GitHub repository
6. Execute $ git remote -v
7. Push the origin master
8. Execute $ git status to fetch the status
9. Execute $ cat
10. Check the commits
11. Match the committed ID

UNASSISTED PRACTICE

GitHub: Using SSH

The steps to create a repository in GitHub using SSH are:

1 Create a local repository

2 Create SSH key (ssh-keygen)

3 Configure GitHub with SSH public key

4 Create a local repository

5 Create a local repository

NOTE

The SSH key helps you create a repository without a username and password.

Create a Repository in GitHub using SSH

Problem Statement: Create public repositories for an open source project. When creating your public repository, make sure to use a secure connection without using username and password.

Steps to Perform:

1. Execute $ git status to fetch the status
2. Execute $ ls -lrt
3. Execute $ git log --online
4. Execute $ ssh-keygen
5. Execute $ cat
6. Add SSH key
7. Create a repository
8. Add the GitHub repository
9. Execute $ git remote -v
10. Push the origin master
11. Execute $ git status to fetch the status
12. Check the commits
13. Match the committed ID

UNASSISTED PRACTICE

Overview of Fork and Pull

GitHub: Fork and Pull

Fork

Creates a new repo in your GitHub account

Pull

Updates the repo in your local project by running the commands

GitHub: Fork and Pull

The steps to create a repository in GitHub using fork and pull requests are:

1 Create a fork

2 Clone your fork

3 Modify the code

4 Push your changes

5 Create a pull request

Create a Fork and Pull Request

Problem Statement: The team has been assigned a task as per the client request where they have to create a pull request but don’t have to work on multiple pull requests to the same repository at once.

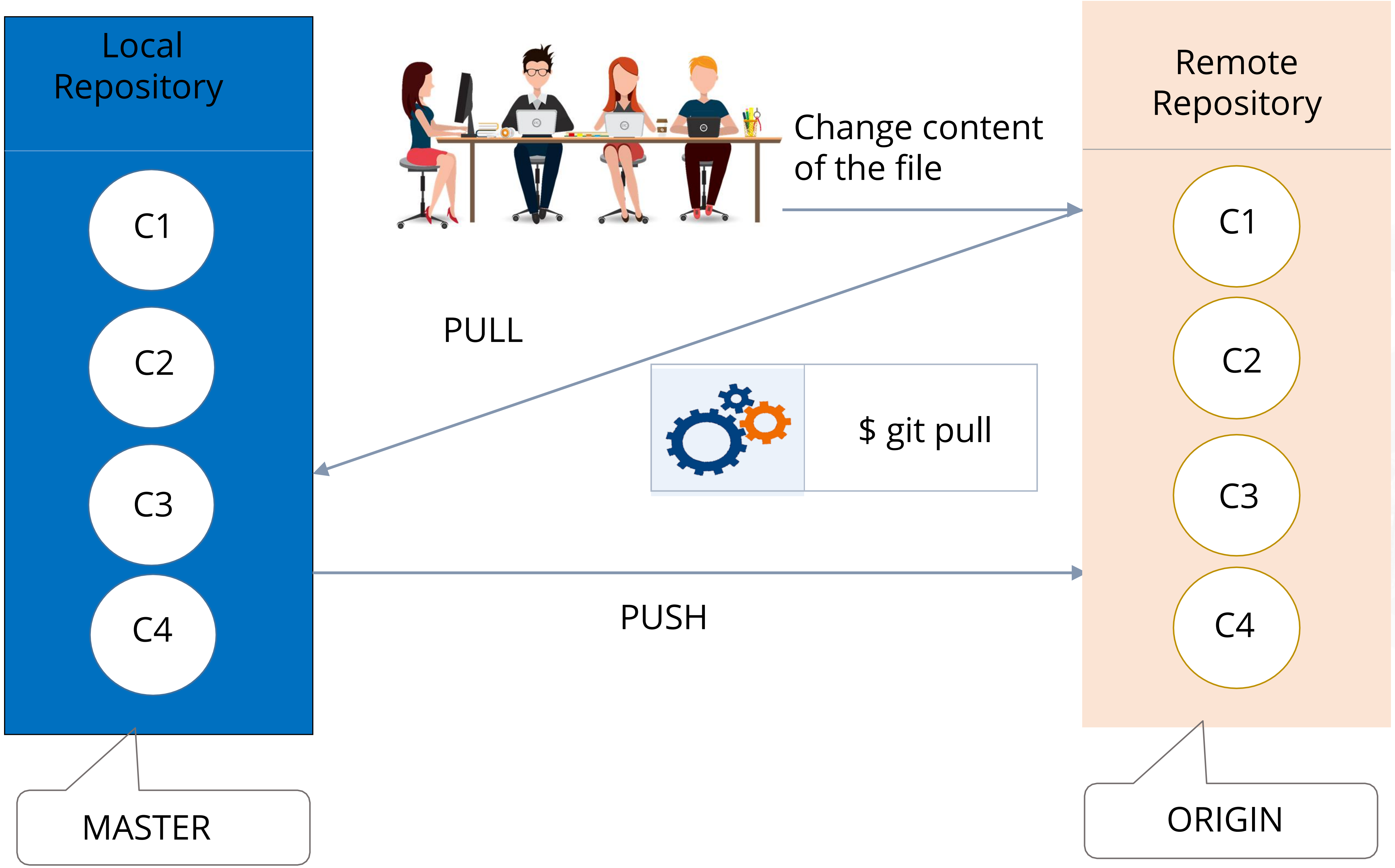
Steps to Perform:

1. Create a Fork
2. Clone your Fork
3. Modify the Code
4. Push your Changes
5. Create a Pull Request

UNASSISTED PRACTICE

Pulling Commits from GitHub

GitHub: Pulling Commits



```
$ git pull
```

Pulling Commits from GitHub

Problem Statement: During scrum the manager has advised the associate of the firm to create a local tracking branch that's associated with a remote branch.

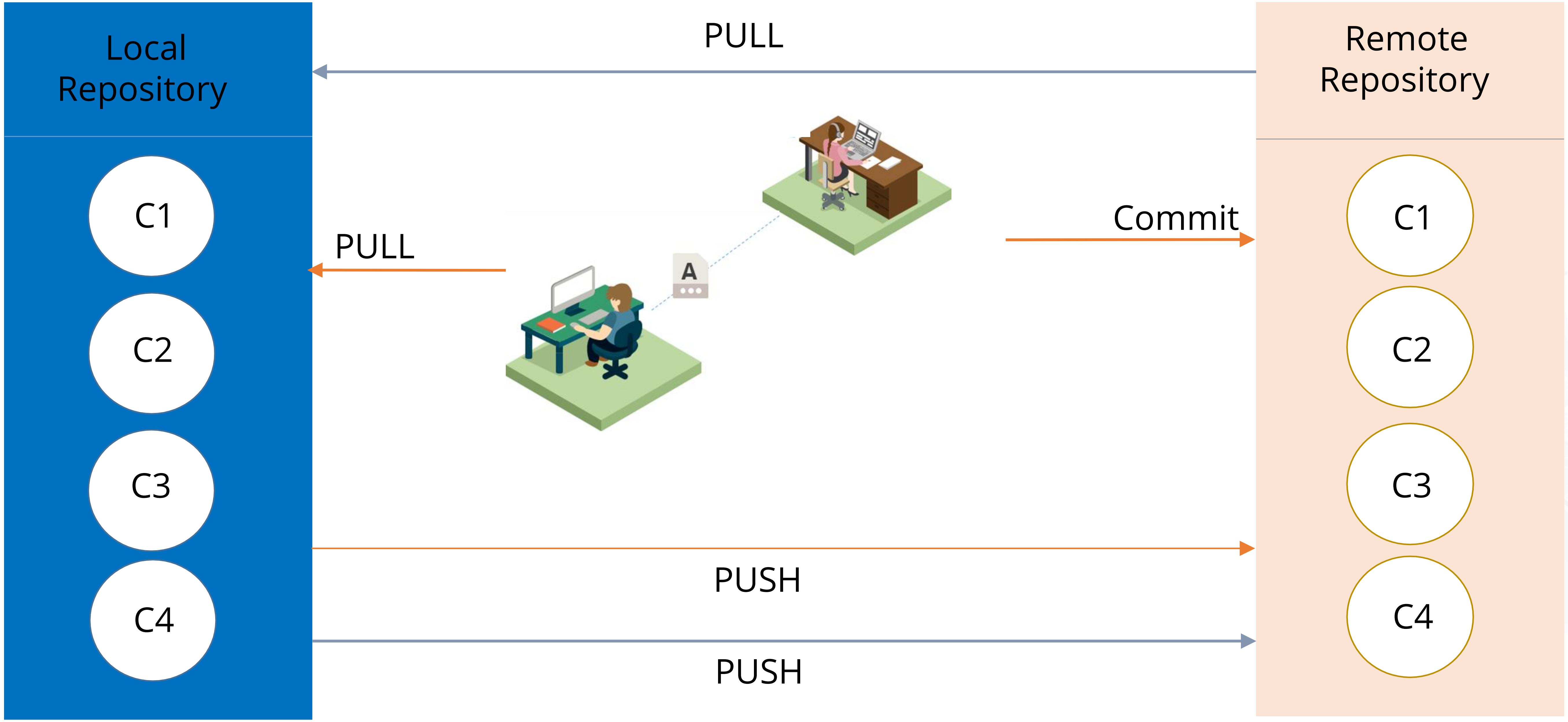
Steps to Perform:

1. Execute $ git status
2. Make changes in the file to create a new repo
3. Check commits
4. Execute $ git log –online
5. Execute $ git status
6. Push the origin master
7. Execute $ git status
8. Execute $ git log –online

UNASSISTED PRACTICE

Collaboration Between a Local and Remote Repository

How to Collaborate?



Collaboration between Local and Remote Repository

Problem Statement: The team wants the make a copy of someone else's GitHub repo in their GitHub account.

Steps to Perform:

1. Make changes on index.html
2. Execute $ git status
3. Execute $ git log –online
4. Pull the origin master
5. Execute $ cat
6. Execute $ vi index.html
7. Execute $ git status
8. Add $ git add and $ git commit -m
9. Add $ git commit -m

10.Add $ git status

11.Add $ git log --online

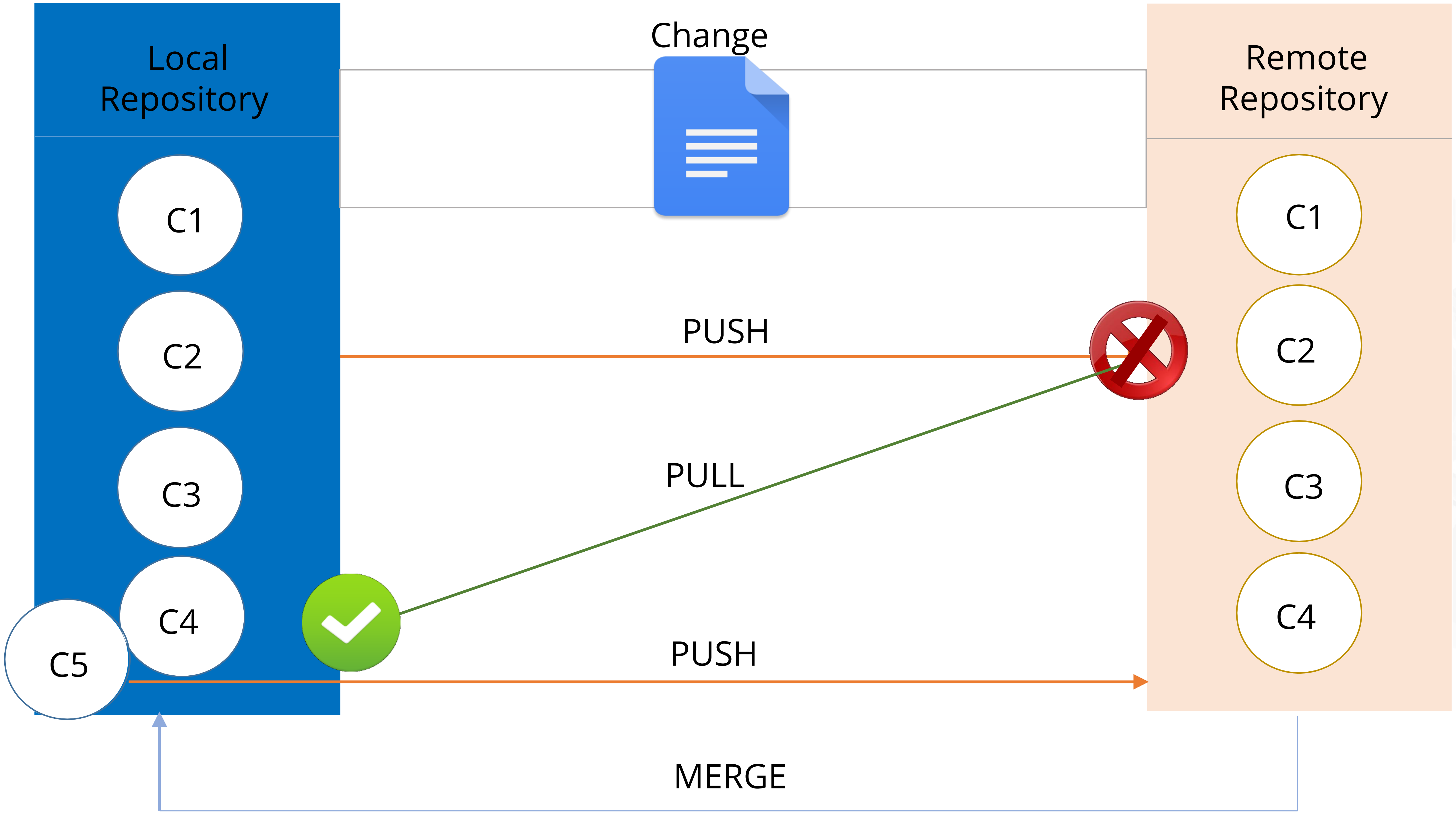
12.Push the last commit

13.Push the origin master

UNASSISTED PRACTICE

Managing Multiple Commits in Git

How does Git manage changes?



Managing Multiple Commits in Git

Problem Statement: The web designing team wants to create a page about writing Git commit messages for the team practices website.

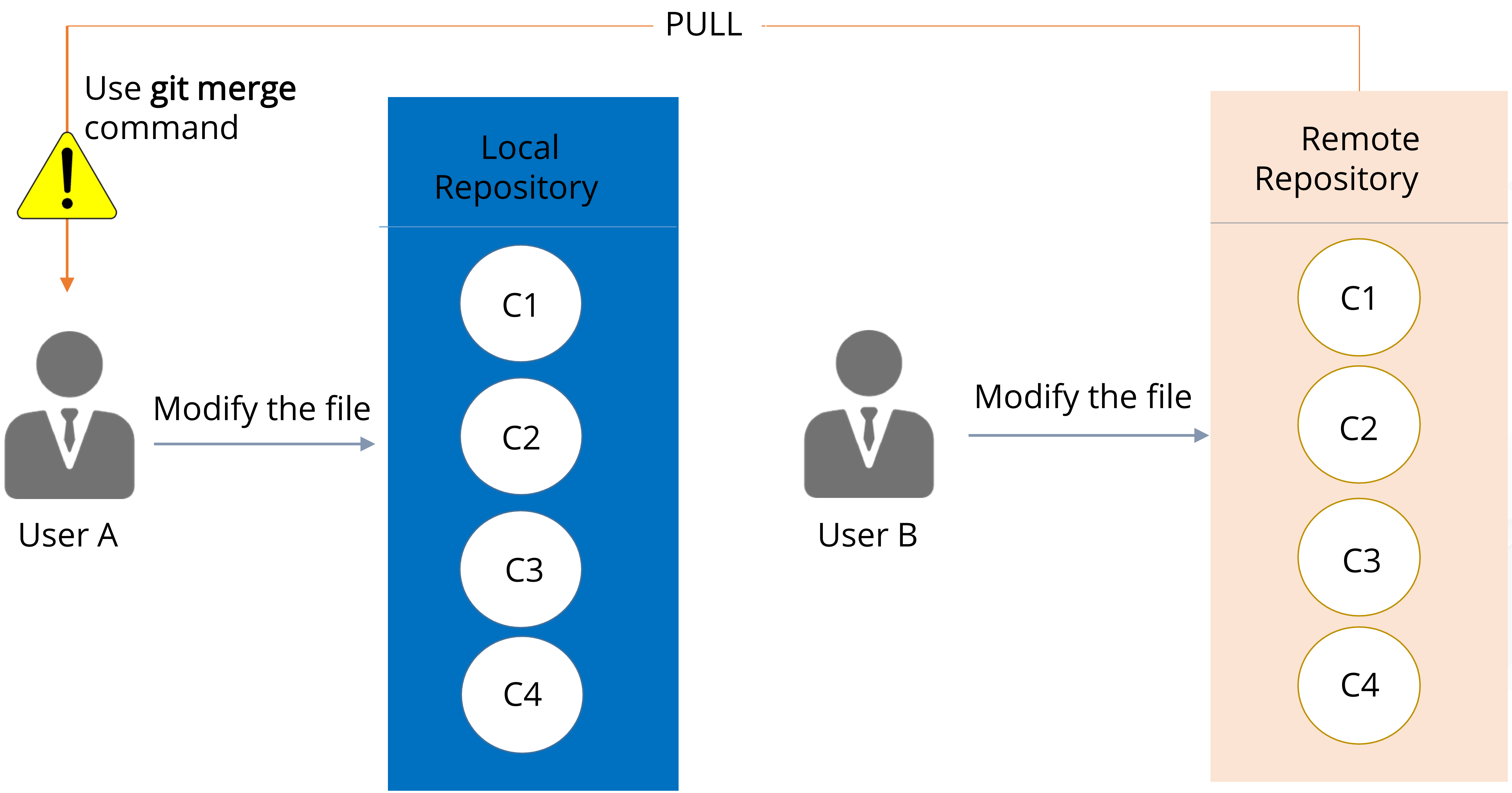
Steps to Perform:

1. Execute $ git status
2. Execute $ git log –online
3. Execute $ vi index.html
4. Add extra header
5. Execute $ git status
6. Add $ git add and $ git commit -m
7. Execute $ git log --online
8. Edit file on the remote repository
9. Push the origin master
10. View the changes in default editor
11. Execute $ git log –online
12. Open file in vi editor
13. Execute $ cat
14. Push the origin master
15. Execute $ git log –online

UNASSISTED PRACTICE

Merging File Changes in Git

Scenario of Merging File Changes in Git



Merging File Changes in Git

Problem Statement: You have assigned a task to combine separate changes to an original in Git.

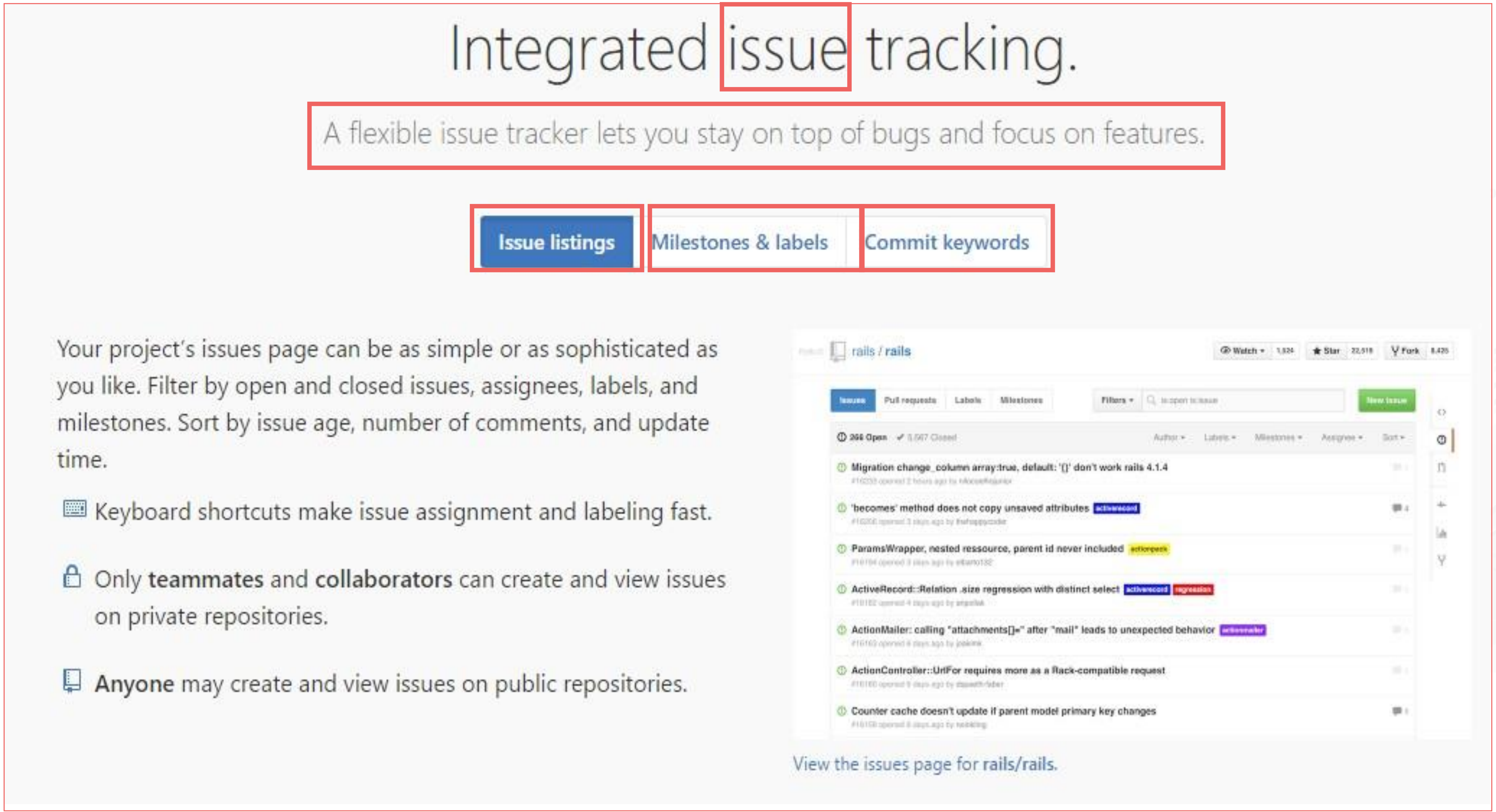
Steps to Perform:

1. Execute $ git log --online
2. Edit the index file
3. Make changes to index file
4. Execute $ git status
5. Execute git add and git commit
6. Execute $ git log –online
7. Push and pull the origin master
8. Edit the index file
9. Execute $ git status
10. Execute git add and commit
11. Execute $ git log --online
12. Push the origin master

UNASSISTED PRACTICE

Issue Tracking in GitHub

Creating and Tracking Issues



Integrated issue tracking.

A flexible issue tracker lets you stay on top of bugs and focus on features.

Your project's issues page can be as simple or as sophisticated as you like. Filter by open and closed issues, assignees, labels, and milestones. Sort by issue age, number of comments, and update time.

⌨ Keyboard shortcuts make issue assignment and labeling fast.

🔒 Only **teammates** and **collaborators** can create and view issues on private repositories.

Anyone may create and view issues on public repositories.

Migration change_column array:true, default: '{}' don't work rails 4.1.4
#16233 opened 2 hours ago by [illegible]

'becomes' method does not copy unsaved attributes activerecord 4
#16206 opened 3 days ago by [illegible]

ParamsWrapper, nested ressource, parent id never included actionpack
#16194 opened 3 days ago by [illegible]

ActiveRecord::Relation .size regression with distinct select activerecord regression
#16182 opened 4 days ago by [illegible]

ActionMailer: calling "attachments[]=" after "mail" leads to unexpected behavior actionmailer
#16163 opened 6 days ago by [illegible]

ActionController::UrlFor requires more as a Rack-compatible request
#16160 opened 5 days ago by [illegible]

Counter cache doesn't update if parent model primary key changes 1
#16158 opened 6 days ago by [illegible]

View the issues page for rails/rails.

Creating and Tracking Issues

Problem Statement: You have been assigned a task to do a walkthrough of creating and tracking issues.

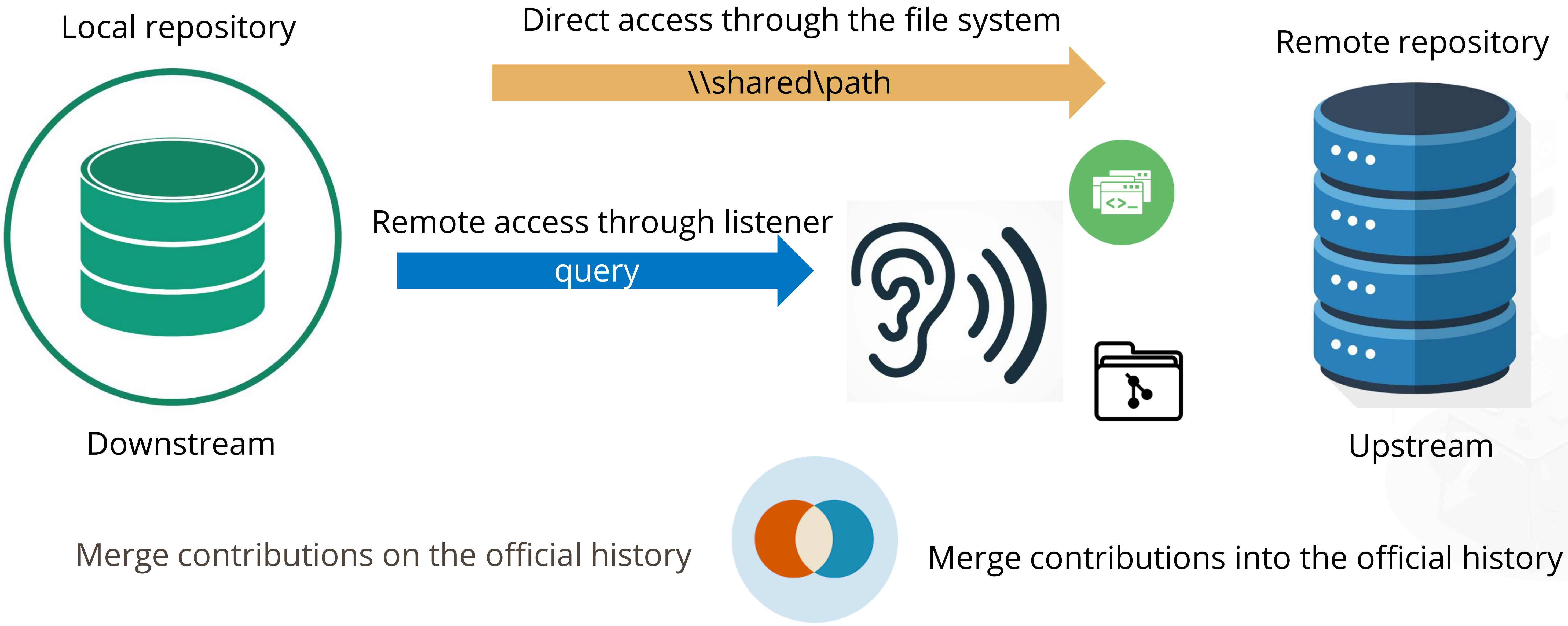
Steps to Perform:

1. Select issues tab
2. Create new label
3. Create milestone
4. Create a new issue
5. Comment on the issue created

UNASSISTED PRACTICE

Overview of Upstream and Downstream

Downstream and Upstream



Local repository

Direct access through the file system

\\shared\path

Remote repository

Remote access through listener

query

Downstream

Upstream

Merge contributions on the official history

Merge contributions into the official history

Getting Started with Git Upstream

Problem Statement: Your team were forking projects but missed out sending it back to the parent repository which means you're at risk for losing track of them. You have to find a solution to make sure contributors are drawing from the same place.

Steps to Perform:

1. Verify the setup of remote repo
2. Add upstream with remote
3. Verify remote added correctly
4. Fetch upstream
5. Use merge or rebase

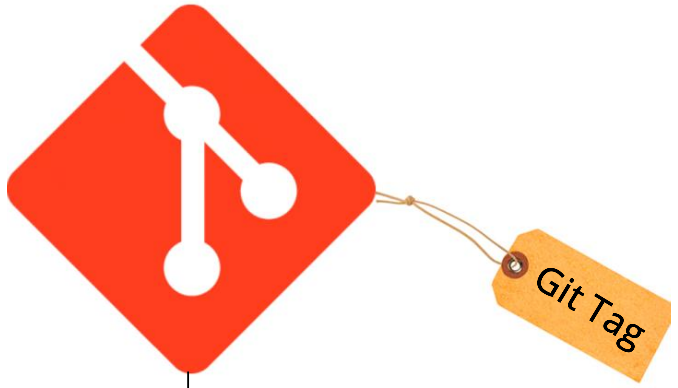
UNASSISTED PRACTICE

FULL STACK

Creating and Verifying Tags

Introduction to Tags

- Used to label and mark a specific commit in the history
- Indicate release versions with the release name



Lightweight tag

- Points directly to a specific commit in the history
- Can only add tagger's name

Annotated tag

- Points directly to a specific commit
- Can add comments, signature, date, tagger's name, and email ID

Different Commands in Tags

Command	Explanation
-a/--annotate	Create an unsigned and annotated tag object
-s/--sign	Create a GPG-signed tag using the default email address key
--no-sign	Override **tag.gpgSign** configuration variable that is set to force each and every tag to be signed
-u <keyid>/--local-user=<keyid>	Create a GPG-signed tag, using the given key
-f/--force	Replace an existing tag with the given name (instead of failing)
-d/--delete	Delete existing tags with the given names
-v/--verify	Verify the GPG signature of the given tag names

Create and Delete Tag

Problem Statement: Write a command to create and delete a tag.

Steps to Perform:

1. Create a tag
2. Delete a tag

Key Takeaways

- Github provides web-based git repository hosting service.
- Pull updates the repo on local system and Fork helps to create repo on a GitHub account.
- A local repository can be connected to one or more remote repositories.
- Files can be committed,merged, and used to track issues.
- Tag is used to label and mark a specific commit in the history.

Knowledge Check

Knowledge Check

1

Git would reject pushing changes to the remote repository under which of the following circumstances?

a. When the remote repository is in sync with the local repository

b. When the same file has been modified both at the local and remote repositories

c. When you use HTTPS protocol instead of SSH protocol

d. When certain files in the remote repository are not in sync with the local repository

Knowledge Check

1

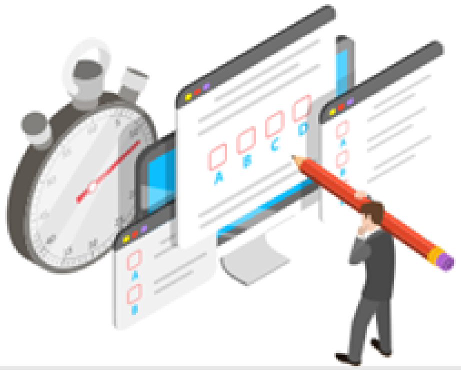
Git would reject pushing changes to the remote repository under which of the following circumstances?

a. When the remote repository is in sync with the local repository

b. When the same file has been modified both at the local and remote repositories

c. When you use HTTPS protocol instead of SSH protocol

d. When certain files in the remote repository are not in sync with the local repository



The correct answer is b

Git would reject pushing changes to the remote repository when the same file has been modified both at the local and remote repositories.

Knowledge Check

2

Which of the following statements is TRUE about GitHub?

a. GitHub works only with servers hosted in the public cloud

b. GitHub works only with servers hosted on your on-premise servers

c. GitHub works with both servers hosted on public or private servers

d. GitHub has only command line interface deployed on your workstation

Knowledge Check

2

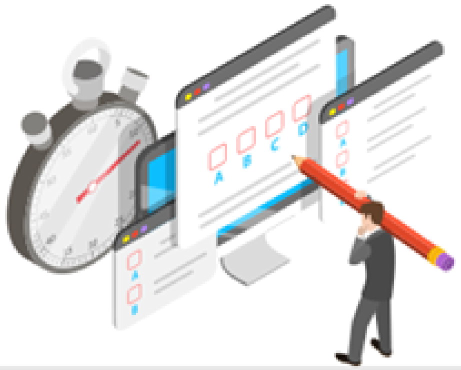
Which of the following statements is TRUE about GitHub?

a. GitHub works only with servers hosted in the public cloud

b. GitHub works only with servers hosted on your on-premise servers

c. GitHub works with both servers hosted on public or private servers

d. GitHub has only command line interface deployed on your workstation



The correct answer is **c**

GitHub works with both servers hosted on public or private servers.

Knowledge Check

3

Which of the following statements is true when you push several changes from the local repository to the remote repository?

a. Git consolidates all commits from the local repository and creates one commit on the remote repository

b. Git would consolidate all changes at each file level and only add one commit per modified file in the remote repository

c. Git would only copy the latest commit of each modified file into the remote repository

d. Git would copy each commit from the local repository to the remote repository so that the entire commit history is available at the remote repository

Knowledge Check

3

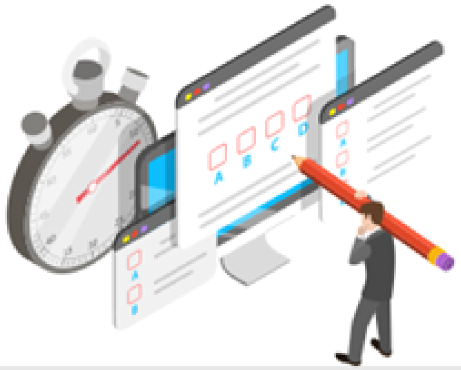
Which of the following statements is true when you push several changes from the local repository to the remote repository?

a. Git consolidates all commits from the local repository and creates one commit on the remote repository

b. Git would consolidate all changes at each file level and only add one commit per modified file in the remote repository

c. Git would only copy the latest commit of each modified file into the remote repository

d. Git would copy each commit from the local repository to the remote repository so that the entire commit history is available at the remote repository



The correct answer is d

When you push several changes from the local repository to the remote repository, Git would copy each commit from the local repository to the remote repository so that the entire commit history is available at the remote repository.

Remote Repositories

Duration: 30 mins.

Problem Statement:
Fork upstream repo which will cover concepts like creation, issue, pull, and merge.

LESSON-END PROJECT