

## **Tutorial: Using a Git Repository on Ubuntu**

This tutorial demonstrates how to install the Git client, clone a repository, perform basic Git operations, and handle changes. For learners who don't have write access to the repository, it includes instructions for forking the repository.

<b>Tutorial: Using a Git Repository on Ubuntu</b>	<b>1</b>
<b>Step 1: Install Git on Ubuntu</b>	<b>2</b>
<b>Step 2: Fork the Repository</b>	<b>2</b>
<b>Step 3: Clone the Repository</b>	<b>2</b>
<b>Step 4: Configure Git</b>	<b>3</b>
<b>Step 5: Basic Git Operations</b>	<b>3</b>
<b>Step 6: Update the Repository</b>	<b>3</b>
<b>Step 7: Sync with the Upstream Repository</b>	<b>4</b>
<b>Step 8: Additional Operations</b>	<b>4</b>
<b>Note for Learners</b>	<b>5</b>

## Step 1: Install Git on Ubuntu

### Update the System Packages

Open the terminal and run:

```
sudo apt update
```

1. **Install Git**

Install Git using:

```
sudo apt install git -y
```

2. **Verify Installation**

Confirm that Git is installed:

```
git --version
```

## Step 2: Fork the Repository

Since learners do not have write access to the repository, they need to fork it.

1. **Navigate to the Repository**

Visit <https://github.com/devopscert202/ckacoursenov24> in a web browser.

2. **Fork the Repository**

- Click the **Fork** button at the top-right corner of the repository page.
- The forked repository will now appear in your GitHub account.

## Step 3: Clone the Repository

1. **Copy the Forked Repository URL**

From your GitHub account, copy the HTTPS URL of your forked repository.

### Clone the Repository

Run the following command to clone the repository locally:

```
git clone <your-forked-repository-url>
```

Example:

```
git clone https://github.com/<your-username>/ckacoursenov24.git
```

## 2. Navigate to the Cloned Directory

```
cd ckacoursenov24
```

## Step 4: Configure Git

### Set User Name

```
git config --global user.name "Your Name"
```

### Set User Email

```
git config --global user.email "your-email@example.com"
```

---

## Step 5: Basic Git Operations

### Add a New File

Create a file:

```
echo "This is a test file" > testfile.txt
```

Add the file to staging:

```
git add testfile.txt
```

### Commit Changes

Save the staged changes with a commit message:

```
git commit -m "Added testfile.txt"
```

### Push Changes

Push the changes to your forked repository:

```
git push origin main
```

---

## Step 6: Update the Repository

### Pull Updates from the Forked Repository

Synchronize the local repository with the remote:

```
git pull origin main
```

### **Add Changes to an Existing File**

Edit a file:

```
echo "Additional content" >> testfile.txt
```

Stage and commit the changes:

```
git add testfile.txt
```

```
git commit -m "Updated testfile.txt"
```

Push the changes:

```
git push origin main
```

---

## **Step 7: Sync with the Upstream Repository**

If the original repository (upstream) has changes, you can sync your fork:

### **Add the Upstream Repository**

```
git remote add upstream https://github.com/devopscert202/ckacoursenov24.git
```

#### **1. Fetch Changes from Upstream**

```
git fetch upstream
```

#### **2. Merge Changes**

```
git merge upstream/main
```

#### **3. Push the Changes to Your Fork**

```
git push origin main
```

---

## **Step 8: Additional Operations**

### **View the Git Log**

Display commit history:

```
git log --oneline
```

## Create a Branch

Create and switch to a new branch:

```
git checkout -b feature-branch
```

## Switch Between Branches

```
git checkout main
```

## Delete a File

```
git rm testfile.txt  
git commit -m "Deleted testfile.txt"  
git push origin main
```

---

## Note for Learners

- **Fork First:** Always fork the repository before cloning if you don't have write access.
- **Stay Synced:** Regularly sync your fork with the upstream repository to stay up-to-date.
- **Document Changes:** Add clear commit messages for better tracking.

---

By following this tutorial, you've learned how to manage a Git repository, handle updates, and contribute to a project while collaborating with others.