Labs: List of Tasks

00-Setup

1. Install Tools
2. Setup GitHub

01-Working-With-Repo

1a. Creating a Repository (MCP)
1b. Creating a Repository (WEB)
1c. Creating a Repository (gh)
2. Cloning a Repository
3. Making Changes
4. Committing Changes
5. Pushing Changes

02-Working-With-PR

1. Creating Local Branch
2. Opening Pull Requests
2a. Opening PR (GitHub Web)
2b. Opening PR (GitHub CLI)
2c. Opening PR (GitHub MCP)
3. Working With PR Reviews

August 11, 2025

Nir Geier, Not Committed Yet

Lab 00: Setup

🔨 This lab will guide you through the essential steps to prepare your environment for working with Git and GitHub.

🔨 By the end of this lab, you will have the necessary tools installed and your GitHub account configured.

🔨 After completing this lab, you will be prepared to start working with repositories.

Lab Contents

01. Install Tools:

📒 Learn how to install Git on macOS, Windows, or Linux.

📒 Optionally install the GitHub CLI (`gh`) for advanced GitHub interactions from your terminal.

📒 Configure your Git identity (name and email) for commit attribution.

02. Setup GitHub:

📒 Step-by-step instructions to create and verify your GitHub account

📒 Personalize your GitHub

Installing Tools

About this Guide

This guide will walk you through installing git, creating a GitHub accoun

1. Installing Git

- First, you need to install Git locally.

Installing Git

:fontawesome-brands-git-alt: macOS :fontawesome-

Open Terminal and run:

```
# (Requires [Homebrew](https://brew.sh/))
brew install git
```

Download and install from git-scm.com.

Use your package manager, for example:

```
sudo apt install git
```

2. Installing `gh`

- To work with GitHub from your computer, you can use t
  GitHub directly from your terminal.
- **Click** on the section below to expand it

Setup GitHub

About this Guide

This guide will walk you through setting up your GitHub account and configuring it for the first time.

1. Create GitHub Account

1. Go to https://github.com/.
2. Click **Sign up** in the top-right corner.
3. Enter your email address and click **Continue**.
4. Create a strong password and click **Continue**.
5. Choose a username (this will be your public identity on GitHub).
6. Follow the prompts to verify your account (you may nee solve a puzzle or enter a code sent to your email).
7. Choose your plan (the free plan is sufficient for most us
8. Complete the setup by answering a few questions (opti and click **Complete setup**.

2. Verify Your Email Address

1. Check your email inbox for a message from GitHub.
2. Click the verification link in the email to activate your account.

3. Set Up Your Profile

1. Click your profile icon in the top-right and select **Your p**
2. Click **Edit profile** to add your name, bio, location, and p picture.
3. Save your changes.

Lab 01: Working With Repositories

🔨 This lab guides you through the essential tasks for working with GitHub repositories

🔨 You will learn how to create, clone, modify, commit, and push changes to repositories using different methods and tools.

Lab Contents

Summary

By completing this lab, you will be able to:

:material-check: Create new repositories using multiple methods (MCP, Web, CLI)
:material-check: Clone repositories to your local machine
:material-check: Make and track changes to files

Creating a GitHub Repository Using GitHub MCP

- This guide explains in detail how to create a new GitHub repository using the GitHub Copilot Model Context Protocol (MCP) extension installed in Visual Studio Code.
- The MCP extension allows you to automate repository management tasks, including repository creation, directly from the VS Code interface using natural language prompts.

Prerequisites

- `Visual Studio` Code installed on your computer
- The `GitHub Copilot MCP` extension installed in VS Code
- A `GitHub account` with permission to create repositories

01.01. Install (MCP Extension)

1. Open Visual Studio Code.
2. Go to the Extensions view (`Ctrl+Shift+X` or `Cmd+Shift+X` on Mac).

Creating a GitHub Repository Using Web

- This guide will walk you through the process of creating a new repository on GitHub using the web interface.
- Follow the steps below to set up your project repository quickly and efficiently.

Prerequisites

- A GitHub account (Sign up here if you don't have one)
- A web browser (e.g., Chrome, Firefox, Safari)

---

1. Creating a New Repository

1. Open your web browser and go to https://github.com.
2. Click **Sign in** at the top right corner and enter your credentials.
3. Once logged in, click the `Repositories` tab.
4. Click on the **New** button to create a new repository.

Creating a GitHub Repository Using `gh`

- This guide explains two ways to create a repository on GitHub using GitHub CLI (`gh`).

Prerequisites

- A GitHub account (Sign up here if you don't have one)
- GitHub CLI (`gh`) installed
- Git installed (Download here)
- Authentication with `gh auth login`

---

1. Authenticating with (`gh`)

- **Open your terminal** and authenticate if you haven't already.
- Execute the following and follow the prompts to log in via browser or SSH key.

```
gh auth login
```

```
### Authenticating with GitHub CLI
```

Cloning a GitHub Repository

This guide explains how to clone a repository from GitHub to your local machine using the GitHub website and Git.

Prerequisites

- Git installed on your computer (Download Git)
- A GitHub account

1: Find the Repository

1. Go to https://github.com and log in.
2. Navigate to the repository you want to clone.

2: Copy the Repository URL

1. Click the green **Code** button on the repository page.
2. Choose the desired protocol:
3. **HTTPS** (recommended for most users)

Making Changes in a GitHub Repository

This guide explains how to make changes to files in your local copy of a GitHub repository.

Prerequisites

- A cloned copy of the repository on your computer
- A text editor or IDE

Step 1: Open the Repository

1. Open your terminal and navigate to the repository folder:

   ```
   cd <repository-name>
   ```

2. Open the project in your preferred editor (e.g., VS Code, Atom, Sublime Text).

Committing Changes in a GitHub Repository

This guide explains how to commit your changes to a local GitHub repository using Git.

Prerequisites

- Changes made to files in your local repository
- Git installed on your computer

Step 1: Stage Your Changes

1. Open your terminal and navigate to the repository folder.
2. To stage all changes, run:

   ```
   git add .
   ```

   Or, to stage specific files:

   ```
   git add <filename>
   ```

Pushing Changes to GitHub

This guide explains how to push your local commits to a remote GitHub repository.

Prerequisites

- Committed changes in your local repository
- Remote repository set up (e.g., on GitHub)
- Git installed on your computer

Step 1: Check Remote Repository

1. In your terminal, run:

   ```
   git remote -v
   ```

2. Ensure the correct remote URL is set (usually named `origin`).

Lab 02: Working with Pull Requests

This lab covers the complete workflow of working with pull requests in GitHub, from creating branches to managing the review and approval process.

Lab Overview

This lab is divided into three main sections, each covering different aspects of the pull request workflow:

01-Creating-Local-Branch.md

Learn how to create local branches and push them to remote repositories: - Creating and managing local branches - Pushing branches to remote repositories - Branch naming conventions and best practices - Keeping branches updated - Common troubleshooting scenarios

Creating Local Branch and Pushing to Remote Repository

This guide explains how to create a local branch and push it to the remote repository. Working with branches is essential for collaborative development and maintaining a clean project history.

Prerequisites

- Git installed on your computer
- A cloned repository on your local machine
- Proper authentication setup with GitHub (SSH keys or HTTPS)

---

1. Check Current Branch Status

Before creating a new branch, check your current branch and repository status:

```
# Check current branch
git branch
```

Opening Pull Requests: Complete Guide

This guide provides a comprehensive overview of creating pull requests using three different methods. Each method has its own detailed guide for in-depth learning.

Overview

Pull requests are a fundamental part of collaborative software development. They allow you to propose changes, discuss them with your team, and merge them into the main codebase after review. This lab covers three primary methods for creating pull requests:

1. **GitHub Web Interface** - Visual, user-friendly approach
2. **GitHub CLI** - Command-line efficiency for developers
3. **GitHub MCP Extension** - AI-powered natural language approach

---

Prerequisites

- A repository with at least one branch containing changes
- Proper authentication setup with GitHub
- Basic understanding of Git branching concepts

Opening Pull Requests via GitHub Web Interface

This guide explains how to create pull requests using the GitHub web interface. This is the most visual and user-friendly method for creating pull requests.

Prerequisites

- A repository with at least one branch containing changes
- Proper authentication setup with GitHub
- Your feature branch has been pushed to the remote repository

---

1: Navigate to Repository

1. Go to your repository on GitHub.com
2. Ensure your feature branch has been pushed to the remote repository

---

Opening Pull Requests via GitHub CLI

This guide explains how to create and manage pull requests using the GitHub CLI (gh command). This method is perfect for developers who prefer command-line workflows.

Prerequisites

- A repository with at least one branch containing changes
- GitHub CLI installed (`gh` command)
- Proper authentication setup with GitHub CLI

1: Install and Authenticate GitHub CLI

```
# Install GitHub CLI (if not
already installed)
# macOS
brew install gh

# Windows
winget install --id GitHub.cli

# Linux (Ubuntu/Debian)
sudo apt install gh
```

Opening Pull Requests via GitHub MCP Extension

This guide explains how to create and manage pull requests using the GitHub Model Context Protocol (MCP) extension in Visual Studio Code. This AI-powered method allows you to create PRs using natural language commands.

Prerequisites

- Visual Studio Code installed
- GitHub Copilot MCP extension installed and configured
- A repository with at least one branch containing changes
- Proper authentication setup with GitHub

1: Install and Setup GitHub MCP Extension

Installation

1. Open Visual Studio Code

Working with Pull Requests: Adding Reviewers, Review Process & Approval

This guide covers the complete pull request review workflow, including adding code reviewers, conducting reviews, and managing the approval process using GitHub Web, GitHub CLI, and GitHub MCP.

Prerequisites

- An open pull request in a GitHub repository
- Proper permissions to request reviews and approve changes
- Understanding of the codebase and review requirements
