

Mastering GitHub: A Deep Dive

Key takeaways:

- **Git fundamentals:** Understanding the history and concepts of Git and GitHub.
- **Getting started:** Setting up Git, creating an account, and making initial commits.
- **Branching and merging:** Working on different branches, using git push and pull, and resolving conflicts.
- **Version control best practices:** Utilizing the .gitignore file to exclude unwanted files.
- **Collaboration:** Leveraging GitHub's issue section, forking and cloning repositories, and sending pull requests.
- **Deployment:** Hosting static sites on GitHub and exploring VPS and cPanel.
- **CI/CD introduction:** Gaining a foundational understanding of CI/CD pipelines.

I'm excited to apply these valuable skills to future projects and collaborate effectively with others on GitHub.

Git: Imagine Git as a time machine for your code. It saves snapshots of your work at different points, so you can go back and undo changes if something goes wrong. It's like having a backup of your code that you can access whenever you need it.

GitHub: Think of GitHub as a social network for developers. It's a platform where you can share your code with others, collaborate on projects, and learn from the work of other developers. It's like a library for code, where you can find and contribute to different projects.

What is Version Control System? [VCS]

A version control system is a software tool that helps you track and manage changes to your files over time.

It's like a time machine for your work, allowing you to go back to previous versions of your files if you need to.

Here are some key benefits of using a version control system:

- **Undo changes:** If you make a mistake, you can easily revert to a previous version of your work.
- **Collaborate with others:** Multiple people can work on the same project simultaneously without overwriting each other's changes.
- **Track changes:** You can see who made changes to a file and when, making it easier to identify the source of errors.

Prepared by: [Bishal Rijal](#)

By any chance, if you're interested in Digital Marketing and MERN- [Digital Bishal](#)

- **Create backups:** Version control systems automatically create backups of your work, so you don't have to worry about losing your data.

Popular version control systems include Git, SVN, and Mercurial.

Some basic commands of GitHub

echo "# Git-and-GitHub" >> README.md

git init

git add. [*. Stands for all*]

git commit -m "<Your Commit Message>"

git branch -M main

git remote add origin

<https://github.com/bishaldsrija08/Git-and-GitHub.git>

git push -u origin main

Branching

Branches in GitHub are like parallel universes of your project. They allow you to work on different features, bug fixes, or experimental ideas without affecting the main, stable version of your code. Each branch is a separate copy of the project, so you can make changes to it without worrying about breaking the main version.

Key benefits of using branches:

- **Isolation:** Work on new features or bug fixes without affecting the main codebase.
- **Collaboration:** Multiple developers can work on different branches simultaneously.
- **Experimentation:** Test new ideas without risk to the main project.
- **Rollback:** If a change goes wrong, you can easily switch back to a previous branch.

Common branch types:

- **Main branch:** The primary branch of your project, typically representing the stable version.
- **Feature branches:** Created for developing new features or enhancements.
- **Bugfix branches:** Created to address specific bugs or issues.
- **Release branches:** Created to prepare a new version of the project for release.

By effectively using branches, you can manage your project more efficiently, reduce the risk of errors, and collaborate more effectively with your team.

Command to create a new branch:

Git branch <branch name>

Command to switch branch:

Command to pull from branches

git pull origin <branch-name>

Command to merge from branches

git merge <branch-name>

Command to ignore already pushed file

git rm -r --cached

git rm -r --cached <file name>

Forking, Cloning, Changing, and Sending a Pull Request

Step 1: Fork the Repository

1. **Navigate to the repository** you want to contribute to on GitHub.
2. **Click the "Fork" button** in the top-right corner. This creates a copy of the repository under your own account.

Step 2: Clone the Forked Repository

1. **Copy the HTTPS clone URL** of your forked repository.
2. **Open a terminal** and navigate to the directory where you want to clone the repository.
3. **Use Git to clone the repository:**

git clone <your-forked-repository-url>

Step 3: Create a New Branch

1. **Navigate to the cloned repository** in your terminal.

Step 4: Make Your Changes

1. **Edit the files** in your local copy of the repository to make your changes.
2. **Commit your changes:**

```
git add <files>
```

```
git commit -m "Your commit message"
```

Replace <files> with the specific files you've modified.

Step 5: Push Your Changes to Your Fork

1. **Push your changes** to your forked repository:

```
git push origin <branch-name>
```

Step 6: Create a Pull Request

1. Navigate to your forked repository on GitHub.
2. Click the "Compare & pull request" button.
3. Review the changes and write a clear description of your contribution.
4. Click the "Create pull request" button.

Your pull request will be sent to the original repository's maintainers for review. If approved, your changes will be merged into the main branch.

Hosting a Static Site on GitHub Pages

Prepared by: [Bishal Rijal](#)

By any chance, if you're interested in Digital Marketing and MERN- [Digital Bishal](#)
GitHub Pages is a free service that allows you to host static websites directly from your GitHub repositories. Here's a step-by-step guide on how to host your static site:

1. Create a GitHub Repository:

- Sign in to your GitHub account.
- Click on the "New repository" button.
- Give your repository a name (e.g., "<your-username>.github.io") and a description.
- Choose the "Public" option to make your repository visible to everyone.
- Click "Create repository."

2. Create Your Static Site:

- Create your static site files (HTML, CSS, JavaScript) and place them in the root directory of your repository.

3. Wait for Deployment:

- GitHub will automatically build your site and deploy it to the <https://<your-username>.github.io/<your-repository-name>> URL.

5. Access Your Site:

- Once the deployment is complete, you can visit your site using the URL provided by GitHub.

Additional Tips:

- **Custom Domain:** If you want to use a custom domain for your site, you can set it up in your repository's settings under "GitHub Pages".

By following these steps, you can easily host your static site on GitHub Pages and share it with the world.

VPS hosting

VPS hosting is a type of web hosting where a physical server is divided into multiple virtual servers, each with its own dedicated resources like CPU, RAM, and storage. This means you have a private space on the server, giving you more control and flexibility compared to shared hosting.

Key benefits of VPS hosting:

- **Dedicated resources:** No resource sharing with other users, ensuring consistent performance.
- **Root access:** Full control over your virtual server, allowing you to install any software or applications you need.
- **Scalability:** You can easily upgrade your resources as your website or application grows.
- **Cost-effective:** More affordable than a dedicated server, but offers better performance than shared hosting.

Ideal for:

- High-traffic websites

Prepared by: [Bishal Rijal](#)

By any chance, if you're interested in Digital Marketing and MERN- [Digital Bishal](#)

- E-commerce stores
 - Resource-intensive applications
 - Developers and businesses that need more control over their hosting environment.
-

C-Panel hosting

C-Panel hosting is a popular type of web hosting that provides a user-friendly control panel for managing your website. It offers a wide range of features and tools that make it easy to create, manage, and maintain your website.

Key features of C-Panel hosting:

- **Easy interface:** Easy-to-use control panel with a drag-and-drop interface.
- **Website builder:** Built-in tools to create websites without coding knowledge.
- **Email management:** Manage your email accounts, create mailing lists, and set up email forwarding.
- **Database management:** Create and manage databases (MySQL, PostgreSQL).
- **Security features:** Protect your website from threats with features like SSL certificates, malware scanning, and firewalls.
- **One-click installations:** Install popular applications like WordPress, Joomla, and Drupal with just a few clicks.

Bishal Rijal

GitHub: <https://github.com/bishaldsrija08>

Prepared by: [Bishal Rijal](#)

By any chance, if you're interested in Digital Marketing and MERN- [Digital Bishal](#)

Ideal for:

- Small businesses
- Individuals
- Beginners who need a simple and easy-to-use platform for managing their websites.

C-Panel hosting is a great choice for those who want a hassle-free way to manage their website. With its user-friendly interface and wide range of features, it can help you get your website up and running quickly and easily.