**1. Initialize Git in Your Project**

This sets up your project to use Git version control.

bash

Copy code

git init

* **What it does**: Creates a hidden .git folder in your project, enabling version control for your files.

**2. Create a .gitignore File**

To avoid uploading unnecessary or sensitive files (like node\_modules), create a .gitignore file in your project root.

bash

Copy code

echo "node\_modules/" > .gitignore

* **What it does**: Tells Git to ignore the node\_modules folder, so it won’t be tracked or uploaded.

**3. Add Files to Staging**

bash

Copy code

git add .

* **What it does**: Stages all files (except those ignored by .gitignore) for the next commit. Files must be staged before they can be committed.

**4. Commit Your Changes**

bash

Copy code

git commit -m "Initial commit"

* **What it does**: Saves a snapshot of the staged files with a message describing the changes. The message here is "Initial commit."

**5. Create a Repository on GitHub**

1. Log in to [GitHub](https://github.com).
2. Click the **New** button to create a new repository.
3. Give your repository a name (e.g., MyNodeProject) and click **Create Repository**.

**6. Add the Remote Repository**

Copy the URL from GitHub (e.g., https://github.com/USERNAME/MyNodeProject.git) and link it to your local project:

bash

Copy code

git remote add origin https://github.com/USERNAME/MyNodeProject.git

* **What it does**: Connects your local repository to the remote GitHub repository.

**7. Push Changes to GitHub**

Upload your local files to GitHub:

bash

Copy code

git push -u origin master

* **What it does**:
  + -u sets origin master as the default remote and branch.
  + Pushes your commits to the master branch on GitHub.

**8. Verify the Upload**

Check your GitHub repository to ensure all files (except node\_modules) are uploaded.

**9. Install Dependencies on Other Systems**

When someone clones your repository, they can recreate the node\_modules folder by running:

bash

Copy code

npm install

* **What it does**: Installs all dependencies listed in package.json.

**Meaning of Additional Commands**

* **View Commit History**:

bash

Copy code

git log

* + Shows the history of commits.
* **Check Tags**:

bash

Copy code

git tag

* + Lists all tags (versions) in the project.
* **Push All Tags**:

bash

Copy code

git push --tags

* + Uploads all tags to the remote repository.
* **Delete Old Tag**:

bash

Copy code

git tag -d v1.0

git push origin --delete v1.0

* + Deletes a tag locally and from the remote.
* **Create New Tag**:

bash

Copy code

git tag -a v1.1 -m "Updated release version"

git push origin v1.1

* + Creates a new version tag and pushes it to the remote.

**Summary**

1. git init - Initialize Git.
2. .gitignore - Exclude unnecessary files.
3. git add . - Stage changes.
4. git commit -m "message" - Save a snapshot.
5. git remote add origin <URL> - Link to GitHub.
6. git push -u origin master - Upload files.
7. npm install - Reinstall dependencies elsewhere.