Git Tutorial Transcripts

**Sid:** [00:00:00] Today I'm going to be going over a basic git tutorial for all of you, and we're just going to look into like, working with the LBFD repository, but this will work with any hide repositories as well. So, first thing you need to have is, um, you need to have VS Code installed, and you need to have git bash and just github installed, um, if you don't know how to do that, go to their website and there should be a download.

Um, this is All of the git sort of commands that you can work with. Um, for the most part we're only going to be working with really the basics. And we're not going to work with like the merging and the pulling. Uh, but yeah. So first thing we're going to look into is git clone, which is here. You're going to [00:01:00] need to want to like clone a repository.

So this is my, um, teams repository. So what we're going to do is we're going to copy this link. Copy. And. This can be in any terminal. I'm just doing it in WSL because, um, I don't have the repository in WSL. So, so, in my tutorial, uh, folder, I just say git clone. And then I paste in what I had. And then it's going to start cloning.

Did it really fast because I already did that. If I ls, you can see that I have the, um,

I have the folder, or at least the project. At least the front end project. Um, and then I can see all the files here. Um, if you have VS Code, you can just do code. [00:02:00] And this will open up a window. Um, I don't think you guys will see the window. Capture.

There we go. So this is our front end, sort of, repository, and I can open up a terminal here as well. I don't like working with this one, but just for the sake of easiness, I guess we can just work here. It's not. There's not much to do here, but they actually have really nice source control here. So, we have the stalker file, and I'm just going to add a con, and it's like, uh, This needs to be in WSL, because you can't run the stalker file, um, in Windows, because of Nginx.[00:03:00]

Um, now that I made a change, um, saying, it wants me to, like, stage my changes. Uh, staging your changes is kind of like, Getting them ready for a save. Um, I can also discard this. Which is just like, you can press this button, do the, you can stage it, and then it's ready to like commit. Committing is essentially saving, um, your changes.

So,

but before you can really start working in changes, Best to work in a different branch. So if I press git. Right, it says that I'm on the main branch. And usually we don't want to be on here. Uh, usually you want to create your own branch. And then, um, and then publish your branch. And then, um, somebody else will look into [00:04:00] your branch.

And then do a pull request or whatever. So, let's work on our own branch. We'll just call it std, because that's me. Or we'll just call it toTor. It's comments, actually. A lot better. And then if I press git branch, it says I'm still in main. Um, so I need to switch to git checkout. Um,

and then it says I've switched to branch comments. And if I write in git branch again, we'll see that I am on comments. Um, your VS Code may not look like this, but Uh, just for ease of viewing. The star will show what branch you're working with. Um, now that we've switched branch, it's telling us to publish our branch.

We'll do that after we've made changes in our branch. So, [00:05:00] we'll add a name, jvm, wsl, uh, for run. So to create the image for the dockerfile, we need it in, we need it in wsl. I didn't have this comment, I don't know why, but might as well add it here. Now that we've, I guess, put this in here, um, Oh, it's asking me to install.

Oh, we can do that later. The next thing is, obviously, um, We want to try and commit our, uh, changes, so, We can, obviously, just stage our changes and then publish it, right? Uh, that's like three button clicks. But what, uh, what else we can do, Is my git reference. We can say git, uh, commit, or git add, and git status.

So, Something we need to look into is like, since we've made a [00:06:00] change, Oh, we've modified, it's a, writing git status will show us what we've modified. And we can also add, um, file, which is what we did here. And now it's modified and it's ready to be committed, git commit. Um, I guess I should show you guys how, um, this would work as well.

So, pressing the buttons here, I say git commit. Oh, I need to add by user name. Okay, give me a second, I gotta add by user name.

Unpaused. I added my user name here, just, um, said, write me, yes, and here, and now I was able to commit. Uh, and it's asking for, like, a message. Um, GitHub won't allow you to, uh, I guess, commit [00:07:00] without a message, so So, just like talk about what you did, um, in this commit, so, I added, say, added comments, uh, to the Dockerfile.

Uh, so that people know, uh, to run it with WSL or Linux.

And, we can press this, accept commit message, and, nice! We have our, um, sort of comments committed. We

can also publish our branch. We can do, um, a git push. And then, um, whatever, wherever you want to push it to. Or we can just say, publish.[00:08:00]

And, and now it's been published. Um, let me switch windows for you guys. Uh, Brave, I think? So within Brave, as you can see, Comments had a recent push. And then, we're going to look into a git info request. Out of title. So this is something that you're going to have to do if you made a, uh, a publish. And then you're going to have to ask somebody in your team to sort of, um, review your code.

Um, at least this is what we do in LBFE. Maybe it's different in other people's, uh, workflow. But what we like to do is, uh, any, uh, any publishing has to get pushed. So, but, I'm not going to publish this, uh, but I can create a pull request. And it's saying, oh, you need, merging is [00:09:00] blocked. And you need to review it.

Um, I can merge without because I am the, um, I own the thing, but I'm not gonna do it. But, um, after, uh, if I do, uh, do it, it's gonna add it to our main branch. So, here's main. Here's the comments, uh, thing that I added. Let's see, it says we're one commit ahead. And hopefully, um, that explains the most basic ideas of git.

There's also other things that you might want to look into. Like Restore, um, to look at all of the, if you want to like go back to a certain commit, so let's say you messed up somehow and your production code is just really bad for some reason, um, Git Restore will help you with that, um, [00:10:00] so make sure you look at all of these and also ask for, ask help from your, uh, advisors on how to work with Git, hopefully this helps you just make the basic, uh, workflow easier.

Anyways, thank you.