14.4.2 Installing Git

Click one of the buttons to take you to that part of the video.

Installing Git 0:00-0:23

In this demo we'll explore using Git, which is a distributed version control system that allows multiple users to collaborate on projects, such as a software project that requires many engineers to work together on the same code.

For this demo, I am using OpenSUSE, so OpenSUSE uses YaST, so, to install Git, we'll need to use ‘sudo', and then ‘zypper' and ‘install git'.

Install git 0:16-1:14

We'll need to enter the root password. And then installation will begin to retrieve the repositories in the information that it needs.

Now, note that long with git there be 26 packages installed. There is ‘git' and ‘git-core' and all of the dependencies that they depend upon. In addition, you'll see that there's eight recommended packages being installed, and some other packages are available, but not installed, if you want to use some of the git advanced features.

We'll go ahead and hit ‘y' here to continue to install all of these.

And now the installation is complete.

We'll go ahead and clear the screen so that we can see a little better at the top.

Create or Initialize a git Repository 1:15-2:58

On my local machine I need a place to keep my project files, so I'm going to go ahead and make a directory called projects where I can keep my project files that I'll be working on. And then I'm going to change into that directory.

In that directory, the first thing we'll do is explore the git init command. git init creates an empty git repository or it can also reinitialize an existing one.

To see how this works, let's create a folder in here called ‘my_local_repo'. And we'll change into that directory also.

Let's go ahead and run an ‘ls -a' to verify that that folder is actually empty. Now let's go ahead and create a couple of files in the folder. We'll create a file called localfile1. And we'll create another file called localfile2.

Now let's verify the creation of those files with an ls again ‘ls -a', and you can see that the two files that we created are there in our local repo folder.

Now suppose we want to convert this ‘my_local_repo' folder into a local git repository so that we can begin diversion controller files.

This is done with the git init command. By running ‘git init' in this folder, it will take this exact folder and turn it into a local git repo. It's done that now.

And now we can do once again and ‘ls -a' and you'll see that there's a new folder in this directory called git. This git folder is what makes the folder a git repository.

It's where the index and the local repo information are stored.

View git Repository 2:59-3:18

We can also view the configuration information of the repository using a command called ‘git config -l'. This git config -l command shows that the basics of the repository are created but there isn't a lot of additional information that you would see if it was a remote repository.

Remote git Repositories 3:19-3:35

You can also use git init with the ‘--bare' option, that's b-a-r-e, to create a remote repository. The remote repository is bare because it doesn't need a working directory.

A remote repository is required if you're going to collaborate with others on a project.

Add Files to the Repository 3:36-4:11

With the local repository, we can add files to the index. With ‘git status', we can see that the two files we created have not yet been added to the index. So, let's go ahead and add those. ‘git add localfile1'. And we'll also go ahead and add ‘localfile2'.

At this point, if we run ‘git status' again, you'll see that the files are now green, they been added to the index so there're staged and ready to be committed.

Commit Files 4:12-5:03

Let's go ahead and run ‘git commit' and git commit we'll add those files to our local repository.

When you run git commit it opens up a VI window that gives you the chance to put a message that you want to add to the commit. So, we'll add a message here, "Adding our first files", and then you can hit escape, a colon, and a ‘wq' to write and quit.

That finishes the commit and if we do a ‘git status' again, it tells us that on this branch master there's nothing to commit. Our working tree is clean.

We'll also note that the git commit command suggested that maybe we would want to configure an email address and some additional information for the commit, that may give us some better information for the repo.

Summary 5:04-5:09

That's it for this demo.

In this demo, we installed git.

We configured a local repository and committed the first files to the project.
