I have just logged in to the GitHub website.

Once there, I click on

the green button with the text Create Repository.

When I click on the button,

I am redirected to the create

a new repository screen where I'll

be prompted for who the owner is.

I choose my account as the owner option for this example.

Next, I need to input a repository name.

I type a name called my-first-repo.

Notice that the input field has

a green tick icon beside it.

This is just GitHub letting me know that

this name is available to create the repository.

If it's not, I will see

an X icon and be prompted to rename it.

Now I need to type a value for the description input.

For this, I type practice account for learning Git.

The next option I want you to know about is if

you want the repository to be public or private.

Public just means that anyone

on the Internet can see the repository.

I still have control over who can make changes to it.

It's just available on

the viewable aspect of it on the Internet.

The next option is private,

meaning it's not available for anyone to see.

I can only allow access by

granting people access to the repository.

The next few options are about initialization.

I can initialize a repository with a README file,

a gitignore file, and a license if one is required.

For now, I'm just going to choose

the README file option and

then click the "Create repository" button.

A repo has now been set up and I can see that I have

one single file in the repository called README.md.

Md is just short for markdown,

a popular method for creating documentation because

it's shorthand for creating HTML pages.

This allows me to do things like

creating titles and texts.

I can insert images and various other webpage elements.

Notice that the main branch has

also been created and it's important to

know that every repository you

create will have a single main branch at the start.

This is also known as the main line.

Next, I'm presented with additional button options.

The first is labeled Go to file,

then there is Add file,

which you can use to add a new file from the UI.

Finally, a green button labeled code.

Clicking this button provides me with

a GitHub UI options for cloning down the repository.

First is the option for HTTPS,

which contains the HTTPS URL of

the repository and I can use

this to pull it down by using the git clone command.

Next, there is an option for SSH but to use that,

I have to set up

my SSH keys and assign them to the user accounts.

Finally, I have the GitHub CLI option.

Underneath, notice that there are

additional options for GitHub Desktop

if I would like to use that.

Finally, I can also download

a compressed zip file containing

all the files and folder structures.

For this demo, I will show you how to use HTTPS.

To begin, select the HTTPS option and click on

the "Copy" button to copy the HTTPS URL for cloning.

Now I go to my command line that I will be

using to run the commands to clone the repository.

I'm currently in my home directory.

What I usually like to do is create a directory for

all repositories that I'm working on at the moment.

First, I create a directory using the command, mkdir,

then I type the name of the directory I

want to create, which is projects.

Next, I can cd into that,

and now I can run the commands to

clone the project from the GitHub UI.

To do this, I type the command git clone and

paste the HTTPS URL I copied earlier.

Finally, I press "Enter" on my keyboard.

Notice that I receive a message stating that git

is cloning into the my first repo folder.

It then displays messages about

all the objects that have been received.

It also displays a 100-percent status message,

and then finally, a statement that simply says done.

Now I can list the directory by

running the ls-la command,

which means list all directories.

Notice that I have my repository,

which I named my first repo.

This is the name of the repository

that we set up on GitHub.

Finally, if I enter

inside that folder using the cd command,

I can see a single file, the README.md file.

If I use the ls-la command,

another file is listed,

which is just named.git.

You will learn more about this later when you

explore how to use this for source control.

Some command

Mkdir project

Cd project

Git clone GitHub p

Ls -la : which show the github reposirtry name inside github

Which mean list all repository

rovides you with many options for cloning a repository. With which of the following options do you need to set up SSH keys and assign them to the user accounts?



SSH

Correct

That's correct. GitHub allows you to connect using the Secure Shell Protocol (SSH). This provides you with a secure channel over an unsecured network.