**Guided Exercise: Initializing a Git Repository**

In this exercise, you will use VS Code to push your project source code to a remote Git repository.

**Outcomes**

You should be able to:

* Install Git.
* Initialize a local folder as a Git repository.
* Stage a file in a Git repository.
* Commit staged files to a local Git repository.
* Push commits in a local Git repository to a remote repository.
* You have access to a Linux (Debian or Fedora-based), macOS, or Windows system and the required permissions to install software on that system.
* Visual Studio Code (VS Code) is installed on your system.

**Procedure 1.2. Steps**

1. Download and install Git.
   * Linux Installation.
     + Open a new command line terminal.
     + To install Git on Ubuntu and Debian systems, use the following command:

yourname@yourhost:~$ sudo apt install git

The command may prompt for your password to install the package.

* + - To install Git on Fedora and Red Hat Enterprise Linux 8 systems, use the following command:

[yourname@yourhost ~]$ sudo dnf install git

The command may prompt for your password to install the package.

* + macOS Installation.
    - Git is installed by default on the latest macOS versions. To verify the Git installation, open a new command line terminal and enter the following command:
    - $ git --version

git version 2.22.0

* + Windows Installation.
    - In a browser on your Windows system, navigate to <https://git-scm.com/download/win> and save the executable file to your system.
    - In Windows Explorer, navigate to the downloaded file. Double-click the file to start the setup wizard. If prompted, click Yes to allow the installer to make changes to your system.
    - Click Next to accept the license agreement.
    - Click Next to accept the default installation location for Git. If a window displays a warning about the installation location, click Yes to continue the installation of Git to that location.
    - Click Next to accept the installation of the default set of components.
    - Click Next to accept the default Start Menu Folder.
    - Select Use Visual Studio Code as Git’s default editor from the editor list to use VS Code as the default editor. Click Next.
    - At the Adjusting your PATH environment prompt, click Next.
    - Make an appropriate choice for the HTTPS transport back-end. If you are unsure of which option to select, then accept the default selection. Click Next.
    - At the Configuring the line-ending conversions prompt, accept the default selection and click Next.
    - Click Next to accept the default terminal emulator settings.
    - At the Configuring extra options prompt, click Next to accept the defaults.
    - Click Install to accept the default experimental features and begin installation. Wait for installation to complete, and then proceed to the next step.
    - Click Finish to exit the setup wizard.

1. Use VS Code to test your Git installation. Configure your Git installation identity with your GitHub credentials.
   * Open VS Code.
   * Click Terminal → New Terminal to open an integrated terminal.
   * Execute git --version in the integrated terminal to test the installation of Git. The command prints the version of the Git installation on your system.

**Note**

VS Code depends on the configuration options selected during the Git installation process. If the git --version command fails in the integrated terminal, try restarting VS Code. Then, repeat this step to check the installation of Git.

* + In a browser, navigate to <https://github.com>. If you do not have a GitHub account, then create one. Log in to GitHub.
  + In the VS Code integrated terminal, execute git config --global user.name *yourgituser*, replacing *yourgituser* with your GitHub user name.
  + In the VS Code integrated terminal, execute git config --global user.email *user@example.com*, replacing *user@example.com* with the email address associated with your GitHub account.

|  |
| --- |
| https://rol.redhat.com/rol/static/static_file_cache/do101-4.2/devenv/git-version-in-terminal-windows.png |

* + Figure 1.22: The VS Code integrated terminal.
  + **Note**
  + Git requires your GitHub user name and password for certain transactions with remote repositories.
  + On Windows systems, Git manages these credentials by default. You are only prompted for credentials the first time you connect to a remote repository.
  + By default on Linux and macOS systems, Git does not manage your remote repository credentials. Git prompts for your credentials each time you connect to GitHub.
  + To cache your credentials on Linux or macOS systems, execute the following command from a system terminal:
  + $> git config --global credential.helper cache

1. Enable the Always Show Providers source control management option in VS Code.
   * Access the Command Palette (View → Command Palette…​) and type settings. Select Preferences: Open Settings (UI) from the list of options.
   * When the Settings window displays, click User → Features → SCM.
   * VS Code displays Source Control Management (SCM) options for VS Code. Select Always Show Providers.

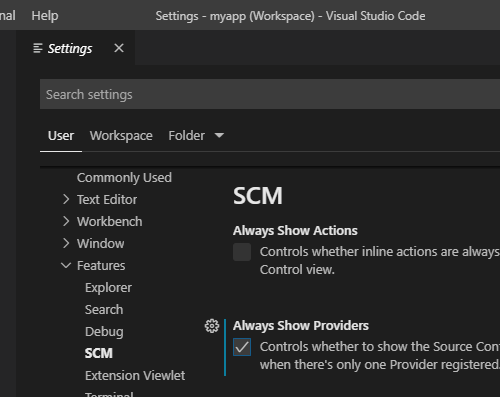


Figure 1.23: Option to always show the Source Control Providers list in the Source Control view of VS Code.

* + Close the Settings tab.

1. Ensure that you have a hello-nodejs project folder in your VS Code workspace. If you already have a hello-nodejs project folder in your VS Code workspace from a previous exercise, then skip this step.
   * Download the following zip file to your system:

<https://github.com/RedHatTraining/DO101-apps/releases/download/v0.1/hello-nodejs.zip>.

Unzip the file, which creates a hello-nodejs folder on your system. The hello-nodejs folder contains a single file, app.js. Note the location of the hello-nodejs folder. You use this folder in a later step.

* + Click File → Add Folder to Workspace…​
  + In the file window, navigate to the location of the unzipped hello-nodejs folder. Select the hello-nodejs folder and click Add.

1. Initialize the hello-nodejs project as a Git repository.
   * Access the VS Code Command Palette (View → Command Palette…​).
   * Type initialize. VS Code provides a list of possible commands that match what you type. Select Git: Initialize Repository from the list of Command Palette options.

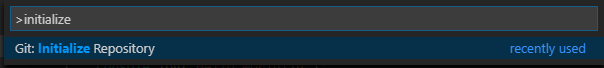


Figure 1.24: Git repository initialization using the Command Palette.

* + Select hello-nodejs from the list of workspace folders.

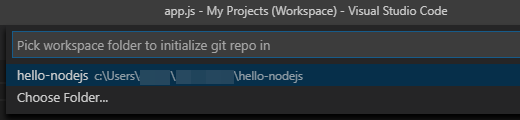


Figure 1.25: Selection prompt to initialize a local Git repository.

1. Create a commit from the app.js file.
   * Click View → SCM to access the Source Control view in the Activity Bar.
   * Hover over the app.js entry under CHANGES. VS Code displays a message that the app.js file is untracked. Click the plus sign for the app.js entry to add the file to the repository.

|  |
| --- |
| https://rol.redhat.com/rol/static/static_file_cache/do101-4.2/devenv/git-sc-changes-windows.png |

* + Figure 1.26: List of changed files.
  + This stages the app.js file for the next commit. The file now appears under the STAGED CHANGES heading.
  + Click in the Message (press Ctrl+Enter to commit) field. Type add initial app.js code in the message field. Click the check mark icon to commit the changes.

|  |
| --- |
| https://rol.redhat.com/rol/static/static_file_cache/do101-4.2/devenv/git-sc-commit-windows.png |

* + Figure 1.27: VS Code commit message box.

1. Create a new GitHub repository to host your project files. Add the GitHub repository as a remote repository for your local hello-nodejs project. Publish your local repository to GitHub.
   * In a browser, navigate to <https://github.com>. If you are not logged in to Github, then log in.
   * Click the + on the upper-right, and then select New repository from the list displayed.

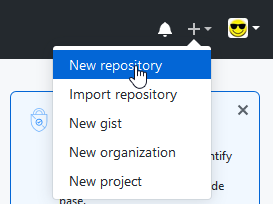


Figure 1.28: Create a new Git repository on GitHub

* + Type hello-nodejs in the Repository name field. By default, the repository is publicly accessible. If you need a private repository, then select the Private check box.

**Warning**

Do not select Initialize the repository with a README. Also, do not add a .gitignore file nor a license to your repository.

Create an empty repository to avoid a merge conflict in a later step.

Click Create Repository to create the new GitHub repository. A summary page provides Git commands for a variety of project initialization scenarios:

|  |
| --- |
| https://rol.redhat.com/rol/static/static_file_cache/do101-4.2/devenv/git-new-repo-summary-windows.png |

Figure 1.29: The summary page for a new GitHub repository

1. Add your new GitHub repository as a remote repository for the hello-nodejs project.
   * In VS Code, type Git: Add in the Command Palette (View → Command Palette…​). Then, select Git: Add Remote from the list of options.

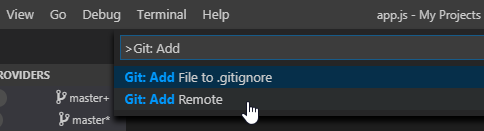


Figure 1.30: The remote URL prompt to add a remote Git repository

* + If you have more that one local Git repository in VS Code, then select hello-nodejs from the list of options.

When prompted for a remote name, enter origin.

**Note**

A Git repository can interact with multiple remote repositories. A remote name of origin is a Git convention to indicate the originating repository for a local Git repository.

* + At the next prompt, enter the HTTPS URL of your hello-nodejs GitHub repository. The URL form is: https://github.com/*yourgituser*/hello-nodejs.

1. Publish your local repository commits to the GitHub repository.
   * Locate the hello-nodejs entry in the SOURCE CONTROL PROVIDERS section, and then click the "Publish Changes" icon.

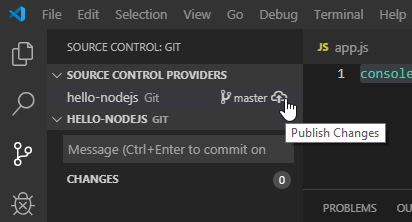


Figure 1.31: Source control view with multiple workspace Git repositories.

The first time VS Code connects to GitHub, a prompt for your GitHub credentials displays. When prompted, provide your GitHub user name and password.

* + If this is your first time publishing commits in VS Code, then an additional prompt displays:

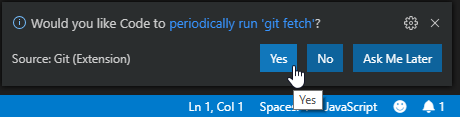


Figure 1.32: VS Code prompt to periodically fetch new commits

Click Yes to configure VS Code to periodically check the remote repository for new commits.

1. In a browser, navigate to https://github.com/*yourgituser*/hello-nodejs, replacing *yourgituser* with your GitHub user name. Verify that your source code is present in your GitHub repository.

|  |
| --- |
| https://rol.redhat.com/rol/static/static_file_cache/do101-4.2/devenv/git-github-verify-windows.png |

1. Figure 1.33: Local project files are present on GitHub
2. To clean up your work, click the Kill Terminal icon to close the integrated terminal window.

This concludes the guided exercise.