GitHub Workflows Training–Mini Sandbox Only

Mini V.00 | 06/06/19

Michael Vincerra
michael.vincerra@intel.com

Practice Repo: GitLab Sandbox

Select this link to access the git-workflows-training repo.

git workflows sandbox

This repo offers you a place to learn Git Workflows. The Repo is provided as a safe environment where you’re free to attempt git commands, and make mistakes, without any major consequences.

Clear Linux Documentation

The instructor will guide you **through** using **git workflows sandbox** during training.

When you're ready to contribute to Clear Linux documentation, engage an instructor and use this PDF training document for more comprehensive training and instruction.

Prerequisites

This tutorial assumes that:

- You use SSH in a Linux environment (e.g., Window Subsystem for Linux)
  - Follow this guide: https://document-publishing.gitlab-pages.devtools.intel.com/tcs-template/howtos/tech/set-up-ubuntu-on-windows.html
- You have **GitHub** account
- You may have little to no experience using the command line interface (CLI)

TCS GitLab resources: https://document-publishing.gitlab-pages.devtools.intel.com/tcs-template/howtos/index.html#deploy-with-gitlab

Configure, create SSH key for GitHub

Check if you already have an SSH key.

1. Enter: *ls -la ~/.ssh*
2. Enter: cat ~/.ssh/id_rsa.pub
3. Copy the contents of: *id_rsa.pub*
4. Skip to "Adding a new SSH key to your GitHub account" below.

OTC recommends that you follow Connecting to GitHub with SSH :

- Checking for existing SSH keys
- Generating a new SSH key and adding it to the ssh-agent
- Adding a new SSH key to your GitHub account

Note: You do not need to use a passphrase when setting up SSH.

For Windows OS, we recommend that you install and use Cygwin.

Clone git-worklfows-sandbox; add remote upstream

1. In the CLI, navigate to your home directory.
2. In GitHub, go to: https://github.com/mvincerx/git-workflows-training
3. Select **Fork** button in the far upper right. (Icon shows it's forked to your account.)
4. Go to *your own GitHub account*, navigate to YOUR FORKED repo.
5. Press **Clone or download**.
6. In the dialogue box, select “Use SSH”. Copy the SSH address.
7. In CLI, enter *git clone* + the copied text, and assure [yourusername] appears.

```
git clone git@github.com:[yourusername]/git-workflows-training.git
```

8. Return to the CLI and change directory into the forked repo:

```
$ cd git-workflows-training
```

9. Repeat Step 2. Repeat Step 5 and copy upstream repo’s URL. Enter :

```
git remote add upstream git@github.com:mvincerx/git-workflows-training.git
```

10. Assure there is an **origin** and **upstream** remote: *git remote -v*

create a branch: add, commit, and push.

1. Assure you complete steps in Synchronize before proceeding.
2. While on the master branch in the CLI, create a new branch. Enter:

```
git checkout -b [xx]-[filename]
```

   Branch name = **[xx]-[filename]**

3. While on new branch, make edits to the document in the Editor (e.g., Sublime).
4. In the CLI, enter command to **add** the revised [filename] and commit your changes.

```
git add [relative/path/filename]
```

5. Enter command. In the editor write a descriptive commit message. Save and exit.

```
git commit –s
```

   See: How to write a good commit message

6. Push your new branch. Then complete a Pull Request.

```
git push origin [xx]-[filename]
```

create a pull request (PR)

1. Visit https://github.com/mvincerx/git-workflows-training. In this case, this is ‘upstream/master’.
2. “*Your recently pushed branches* :” will appear as the last branch you pushed.
3. Select the button “Compare & pull request. "Open a pull request” appears.
4. Find the **gear icon** beside Reviewers: Select the Repo Maintainer as reviewer.
5. Review the pull down menus below “Open a pull request.”
6. Assure *compare* *(right)* shows your branch and *base fork* (left) shows upstream.
7. Select the button “Create pull request.”
8. You will receive email notifications on the status of your pull request.
9. The Repo Maintainer will ask for changes or merge your pull request.

Note: Your pull request will be either merged or others may request additional changes. Continue to push commits to your branch until it's approved.

synchronize your fork to upstream repo

After your PR is merged, sync your fork with upstream master.

1. In the CLI, enter:

```
git checkout master
```

2. Pull changes from upstream master to update your local forked copy:

```
git pull --rebase upstream master
```

We recommend using '--rebase' in this command for a cleaner commit history.

3. Push the changes to your local **origin**—to assure that your **forked copy** is up to date.

```
git push origin master
```

Do these steps every time that you start to work. Do this before you start a new branch.

Congratulations! Your forked repo is now up to date.