Git documentation process

This document provides a guide to editing our documentation using git locally, together with the hpc.aau.dk Github repo

Clone the Github repo

Navigate to the location on your local machine where you want the repo to be stored.

```
cd /path_where_you_want_the_repo/
```

Then clone the hpc.aau.dk repo.

```
git clone https://github.com/aau-claaudia/hpc/
```

If returning to an older repo on your machine, check that you have the most up to date version of the main branch

```
git checkout main
git pull
```

- You should also check what branches you have locally and clean up or delete any unused branches.

Create a new branch and switch over to it

```
git checkout -b newbranchname
```

Alternatively, you can do this after using the longhand option

```
git branch newbranchname
git checkout newbranchname
```

For now everything will exist on only on a local branch.

Check which branch you are on and what changes have been made

```
git branch
```

You can now edit the files that you wish to work on in the branch.

Prepare to commit and commit the changes to the online branch.

Check the status of the branch you are in

```
git status
```

This should display all new files and all files that have been modified in the branch and not yet committed.

The files that you wish to include in your commit should then be added.

Add the files that should be included in the commit

This can be done in bulk in 2 ways, for the current directory using

```
git add .
```

or, for the entire working directory using

```
git add -all
```

Alternatively individual files can be added to the commit by specifying them as follows:

```
git add docs/guides/openstack_guides/delete_instance_launch_from_volume.md
git add mkdocs.yml
git add docs/index.md
```

Commit the changes with a message to indicate the nature of changes

```
git commit -m "New guide for deleting instances and restarting from volume"
```

Push the branch and changes

Once the branch has been created, remember to push it to the web and set the origin to the web based version.

This will allow you to see the branch on the web.

```
git push --set-upstream origin newbranchname
```

Github.com: Merging changes with the main branch

Create, check and merge a pull request

Log into Github and navigate to the hpc page repo.

Next there are _____ steps to complete the merge

1. Create a pull request
2. Check the changes in the pull request
3. Merge the pull request
4. Delete the branch on Github
5. Delete the branch locally
6. Prune remote links to deleted branches

1. Create a pull request

In the Github interface, inside the hpc page repo:

1. Click on the pull request tab
2. Click **New pull request**
3. Select the branch you want to merge into the main branch.

   The changes will be displayed in a code view at the bottom of the screen.

4. Click **Create pull request**

   Provide a title if you wish to change the title from the commit message, and notes about the changes. Current internal processes are that pull requests do not require a review before merging.

2. Check the changes in the pull request

1. Open the pull request tab again.
2. Open the pull request
3. Check the changes in the pull request > If there was a review process it would happen here.

3. Merge the pull request

4. Click "Merge pull request"

4. Delete the branch on Github

Once the pull request has been merged, there is a button available to delete the branch.

If you miss this opportunity, then you find and delete the branch from the repo.

1. Return to the hpc page repo.
2. Click on the branches button under the Code tab.
3. Find the branch that you have just merged, and click the trashcan icon to the right.

5. Delete the branch locally

Once all changes have been merged, the branch can be deleted

This should be done on github first, and then locally.

```
git branch -d branch_name
```

6. Prune remote links to deleted branches

The prune command can then be used to remove any remote links to deleted branches. I.e. if you have a local branch that is set to track an online branch (i.e. upstream) then those links can be removed via pruning.

The following commands should all be run from another branch

```
git checkout main
```

```
git fetch --prune
```

```
git branch -d branchname
```

Additional common commandline tools

Rename a branch

```
git branch -m old_branch_name new_branch_name
```

Force delete a branch locally

```
git branch -D branchname
```