Creating a GitHub Pages Website

*Note: Any typing mistakes during this exercise may cause issues, read carefully. Some of these steps are only performed once per repository. It is recommended this exercise is repeated a number of times to become familiar with the interfaces and commands. A repository is deleted from the repository settings page. This allows a restart of this exercise.*

*This exercise may be the template for an assessment with a fixed time limit of 60 minutes*

Contents

[GitHub Account Creation 2](#_Toc149588038)

[Repository Creation 2](#_Toc149588039)

[Website Initialisation 3](#_Toc149588040)

[Generate SSH Key Pair 4](#_Toc149588041)

[SSH private key to the key list 4](#_Toc149588042)

[SSH public key to GitHub 5](#_Toc149588043)

[SSH Clone String 6](#_Toc149588044)

[Cloning Folder Setup 7](#_Toc149588045)

[Clone from GitHub 7](#_Toc149588046)

[Repository parent folder 7](#_Toc149588047)

[Webpages parent folder ./doc 8](#_Toc149588048)

[Create ipaddresses.html 8](#_Toc149588049)

[Commit Locally 9](#_Toc149588050)

[Local git identity 9](#_Toc149588051)

[Push to GitHub Repository 10](#_Toc149588052)

[Publishing Pages 11](#_Toc149588053)

[Enable GitHub Pages 11](#_Toc149588054)

[html.index to use ipaddress.html 12](#_Toc149588055)

[Continuous Deployment 13](#_Toc149588056)

[404 Page 13](#_Toc149588057)

[Update Remote 14](#_Toc149588058)

[Actions / Deployment 14](#_Toc149588059)

[Commit History 15](#_Toc149588060)

[Extras: 16](#_Toc149588061)

[Windows GitHub Authentication 17](#_Toc149588062)

## GitHub Account Creation

1. Use a GitHub Account – existing or create one if necessary at <https://github.com/>

## Repository Creation

1. Create the repository with the repository name provided or choose a relevant name if none is provided. Please note that the repository is part of the URL which is case sensitive – make the repository name lowercase. Rename the repository in settings, but it makes life easier to get it right the first time.

Click the green New button to configure the new repository.

Name the repository, keep it lowercase

Tick the ‘Add a README file’

Under ‘Add .gitignore’ select the ‘Node’ template

Click ‘Create Repository’

Your repository should be generated and the URL appears in the browser address bar.

*Note: the settings page for a repository allows renaming and deletion. You may repeat this exercise many times.*

## Website Initialisation

1. Create the root folder to store the website resources and the index.html file, Call the folder ‘docs’ (set by GitHub) so create /docs/index.html

*Note: Files can be created and deleted in GitHub. However, always remember to commit the changes. Inconsistent updates may cause cloud vs local synchronization issues for pushes.*

In the browser, create a folder structure by creating a new file in the desired folder.

Click Add File->Create New File

then type the folder name ‘doc’ followed immediately by a / and the folder name appear in the path to the left. Then type index.html

Graphical user interface, text, application

Description automatically generated

Paste the following content to the input area for index.html file and then scroll to the bottom to ‘Commit new file’ after changing the relevant parts in RED to match the repository name for this assessment.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>my **REPLACE ME with repository name** </title>

</head>

<body>

<h1>Hello World from  **REPLACE ME with repository name** </h1>

<!-- ipaddresses paste next line -->

<h4 id="id01">AAAA</h4>

<h5 id="id02">AAAA</h5>

</body>

<script>

const date = new Date();

const date1 = date.toLocaleString("en-uk", {

hour12: true, weekday: "short", hour: "2-digit", minute: "2-digit", second: "2-digit",

month: "2-digit", year: "numeric", day: "2-digit" });

document.getElementById("id01").innerHTML = "" + date1;

document.getElementById("id02").innerHTML = "" + Date.now();</script>

</html>

*Note: It is assumed that a Linux based platform is used for the commands below. If using windows most steps should work but there may be variations and/or minor issues (You will need to Google and possibly install s/w).*

## Generate SSH Key Pair

1. Prepare the SSH key pair as they are needed in later steps.

Open a terminal window to the command line prompt to the home folder. Run ssh-keygen command and respond with default parameters, leave the passphrase empty to make life easier later.

*Note: add an identifier to the filename to prevent conflict if generating multiple pairs.*

After entering the following command with your email, take default options by simply hitting [enter].

ssh-keygen -t ed25519 -C "your@email.com"

Text

Description automatically generated

Confirm the generated private and public key files

ls -l ~/.ssh

Graphical user interface, text

Description automatically generated with medium confidence

## SSH private key to the key list (repeat on reboot)

1. Add the private key to the list maintained by ssh-agent and list the keys.

The addition of keys to the agent is transient i.e. they only last only so long as the agent is running. If you kill the agent or restart your computer the keys are lost until you re-add them again.

Note: Private keys must never be shared with the public – keep them a secret and backup externally.

ssh-add ~/.ssh/id\_ed25519



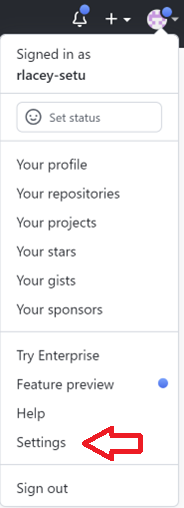
ssh-add -l

*Note: Do not use the same filename for more than one key pair or as this may break the link to the cloud for deployed existing public keys and the relevant SSH steps below will need to be repeated.*

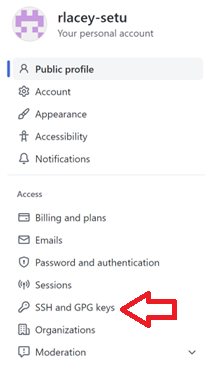
## SSH public key to GitHub

1. Back into the browser at GitHub, from the account menu at the top right select:   
   settings->SSH and GPG keys- followed by New SSH key

**Top right**



**Appears on the left**

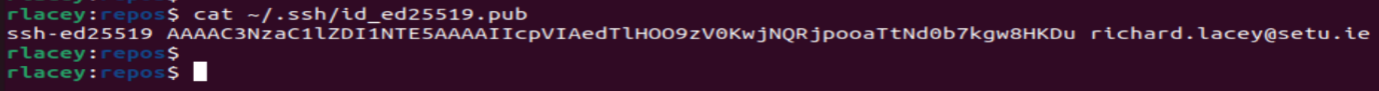


Give the public key a name, typically referencing use/device and copy/paste the text contents of the pub file generated earlier into the Key input and click ‘Add SSH key’ after copying the relevant details.

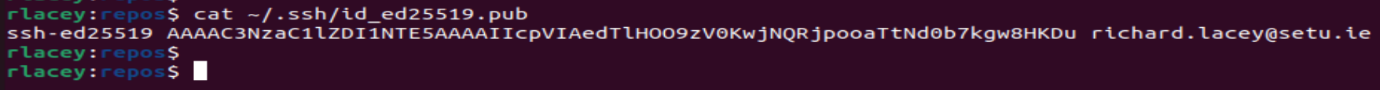
cat ~/.ssh/id\_ed25519.pub

or

nano ~/.ssh/id\_ed25519.pub



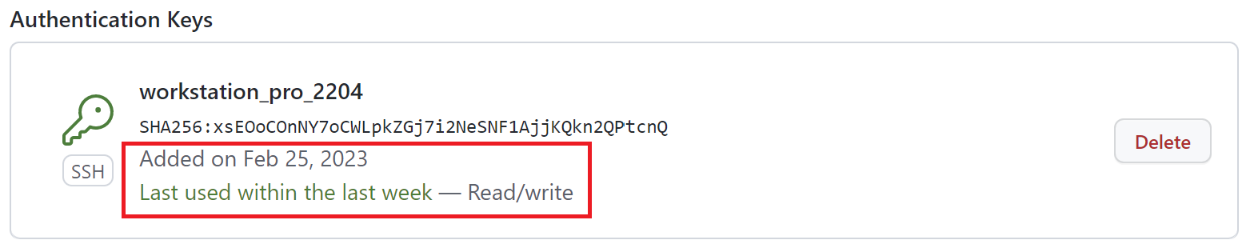
*Note: pasting the public key contents only, getting this wrong causes the SSH to fail later.*



Graphical user interface, text, application, email

Description automatically generated

Result



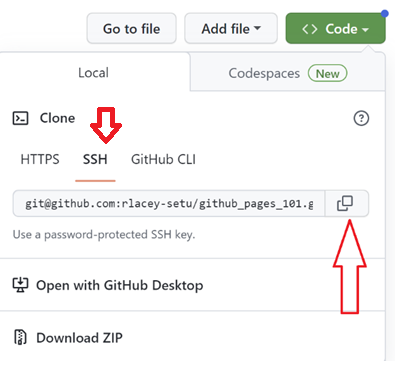
## SSH Clone String

1. On the left click repositories and select the repository used for this exercise.

Clone the repository to the local machine to develop the code.

*Note: Cloning via HTTPS (read only) will cause a problem with an attempt to push updates later due to HTTPS push deprecation. Do not use this option for this exercise.*

Copy the contents of the SSH tab



## Cloning Folder Setup

1. Return to the terminal/console window on the local machine. Move to the parent folder for the repositories, (adjust as relevant) for example:

cd ~

mkdir repos

cd repos

Text

Description automatically generated

## Clone from GitHub

1. Clone the repository from the GitHub cloud- this requires that ‘git’ is installed on the device (install if necessary)

Type git clone and paste the SSH string copied from GitHub SSH (**YOUR** copied string)

git clone [git@github.com:rlacey-setu/github\_pages\_101.git](mailto:git@github.com:rlacey-setu/github_pages_101.git)

Note: Trouble shoot any error message and recover i.e. earlier step(s) may have issues.

Text

Description automatically generated

## Repository parent folder

1. Change into the repository folder just created and list the contents

cd github\_pages\_101

ls



## Webpages parent folder ./doc

1. Move into the ‘docs’ folder which is the parent folder for the website pages containing index.html, list the files and show the contents of the index.html created in the browser earlier.

cd docs

ls

cat index.html

Text

Description automatically generated

## Create ipaddresses.html

1. Create the file ipaddresses.html in the web pages root folder with the commands below.

Use the relevant command to output the IP addresses of the machine to the file.

**Note: install net-tools or equivalent, if required and this may be o/s specific.**

Note: use **ifconfig** on Linux or **ipconfig** for Mac or Microsoft (need own windows solution)

sudo apt install net-tools may not be required

**ifconfig** | grep -ioE "inet(6?)\s[a-z0-9:.]\*" | sed -e 's/$/<p>/' > temp.html

cat temp.html | grep -viE "(127.0.0.1 |::1) " > ipaddresses.html

echo $(date +"%d-%m-%Y %H:%M:%S %Z %s") **>>** ipaddresses.html

cat /etc/machine-id **>>** ipaddresses.html

cat ipaddresses.html

Text

Description automatically generated

**Note: This file is designed to ensure uniqueness with the file ipaddresses.html**

## Commit Locally

1. This file is new to the project and now has to be brought through steps to be committed into the repository so that it is tracked properly for future edits.

Enter the commands below and read the feedback.

git status

Text

Description automatically generated

git add .

Text

Description automatically generated

git status

Text

Description automatically generated

git commit -m 'created ipaddress.html'

An error will occur if not previously authenticated to GitHub from this machine, this is expected the first time for a repo (depending on local setup)

Text

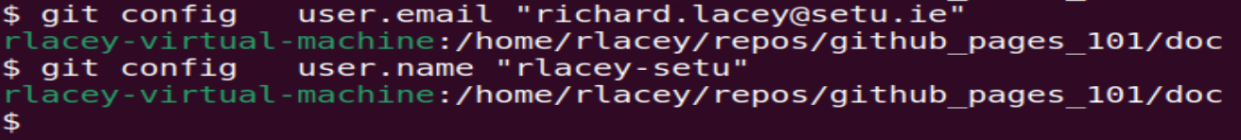
Description automatically generated

### Local git identity

If this happens, do not set the global identity, take the local option as follows.

git config user.email "[you@example.com](mailto:you@example.com)"

git config user.name "Your Name"



Now retry the commit:

git commit -m 'created ipaddress.html'

Text

Description automatically generated

## Push to GitHub Repository

1. Push the latest version of the repo to the cloud so they are synchronized.

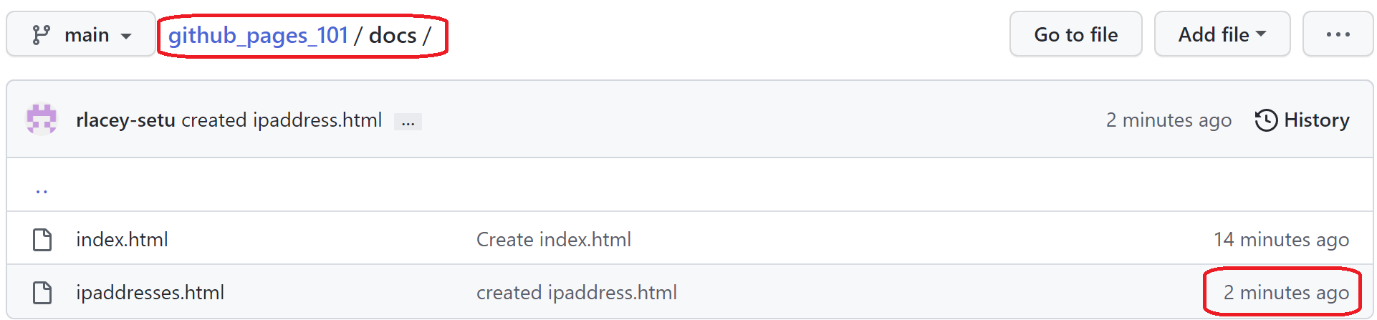
git push origin main

Text

Description automatically generated

Having successfully, pushed local changes to the GitHub cloud repository. Confirm this in the browser.

*Note: After this push, any browser based edits /changes to the code will cause a synchronisation issue which may lead to git complexities beyond the scope here.*



*Note: the push worked due to the previous setup of the SSH steps. A single key pair may be used for multiple repositories from the same device as long as the private key is available on the key list.*

## Publishing Pages

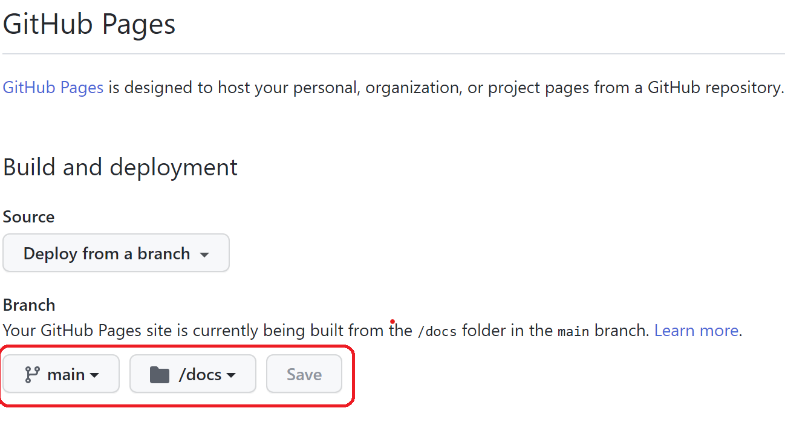
1. Using the menu bar along the top of the page, move from the Code view to the Settings view and click ‘Pages’ on the left.

Graphical user interface, text, application, email

Description automatically generated

## Enable GitHub Pages

1. Enable GitHub Pages by identifying the branch, in this case ‘main’ and the source folder /docs and click ‘Save’



The GitHub Pages website URL is https://{userid}.github.io/{reponame} in this case

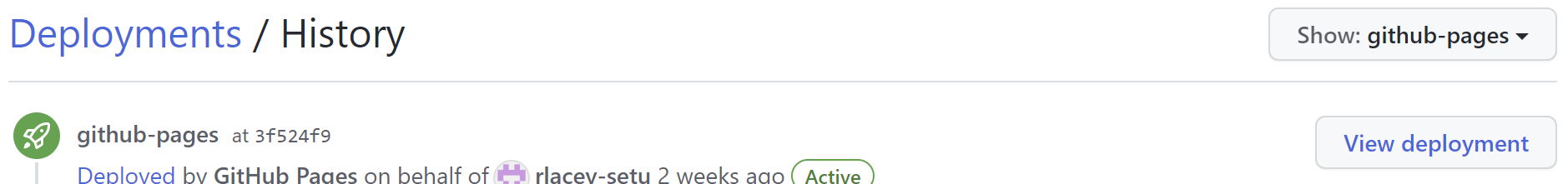
<https://rlacey-setu.github.io/github_pages_101>

You may also access the website (and get the URL) via the bottom right of the repository home page by clicking the ‘github-pages’ link and then click ‘View deployment’.

Graphical user interface, text

Description automatically generated

Followed by clicking the “View deployment” button.



1. The site is published automatically by GitHub after updates, in may take a short period of time to go live. Confirm this

Graphical user interface, text, application, email

Description automatically generated

## html.index to use ipaddress.html

1. Back to your local repo docs folder, edit the index.html file and insert the html below into the structure after the this comment <!-- ipaddresses past next line --> in the body element. Use a suitable text editor.

<br>  
<div id="list">  
 <iframe width="400" height="185" src="./ipaddresses.html"   
 frameborder="0" >  
 </iframe>  
</div>

1. Now commit any file updates with a message and push the change(s) back to the cloud repository.

git status

git add .

git commit -m "ipaddresses index.html"  
git push origin main

Text

Description automatically generated

## Continuous Deployment

1. Confirm the changes have gone live almost immediately by refreshing the public URL of the site after 20 to 60 seconds (network dependent).

Graphical user interface, text, application

Description automatically generated

## 404 Page

1. Paste the following content to the 404.html that is to be created in the docs folder of the local repository and save the file.

<!DOCTYPE html>  
<html lang="en">  
 <head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>my 404</title>  
 </head>  
 <body>  
 <h3>Sorry page does not exist </h3>  
 <h1 id="id01">AAAA</h1>  
 <h1 id="id02">AAAA</h1>  
 </body>  
<script>  
 const date = new Date();  
 const date1 = date.toLocaleString("en-uk", {  
 hour12: true, weekday: "short", hour: "2-digit", minute: "2-digit", second: "2-digit",  
 month: "2-digit", year: "numeric", day: "2-digit" });  
 document.getElementById("id01").innerHTML = "" + date1;  
 document.getElementById("id02").innerHTML = "" + Date.now();  
</script>  
</html>

README.md

1. Paste the following text into the README.md file found in the parent folder of /docs.

GitHub Pages Exercise

===========

Involves

\* git

\* GitHub

\* SSH

## Update Remote

1. Push the latest changes to the remote repository.

git status

git add .

git commit -m "404 and README.md"  
git push origin main

Wait about 20 to 60 seconds and check that the updates are live by refreshing the browser view. (The wait is network dependent, assuming no mistakes made)

Confirm the README.md upload by viewing and generate a 404 error on the website by appending /aaa (any gibberish) to the website URL address.

*Note: some of the tasks are repeated as a teaching exercise, in reality may files can be created and edited before the update steps are performed to update the repository locally and in the cloud.*

*The intention here is to introduce some git / GitHub / GitHub Pages basics, there are many more features available with these applications.*

Actions / Deployment

1. A timeline of actions/deployments and revisit files to see changes in a push.

Actions-> (it’s a menu item for the repo)

https://github.com/rlacey-setu/GitHub\_Pages\_101/actions

Actions->Deployments

<https://github.com/rlacey-setu/GitHub_Pages_101/deployments>

Graphical user interface, text, application, email

Description automatically generated

## Commit History

1. View the commit history:

in the browser

Menu Bar < > Code -> Commits (on the right under green Code button)

Graphical user interface, application

Description automatically generated

at the command line

git log

Text

Description automatically generated

## Extras:

About remote repositories

[https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls](https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories" \l "cloning-with-https-urls)

Websites for you and your projects.

<https://pages.github.com/>

Creating a GitHub Pages site

<https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site>

Configuring a publishing source for your GitHub Pages site

<https://docs.github.com/en/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site>

Creating a custom 404 page for your GitHub Pages site

<https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-custom-404-page-for-your-github-pages-site>

Generating a new SSH key and adding it to the ssh-agent

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Creating and Using SSH Keys in Windows

<https://www.purdue.edu/science/scienceit/ssh-keys-windows.html>



## Windows GitHub Authentication

Graphical user interface, text

Description automatically generated

This error appears when not authenticated against the cloud account.

On Windows a authentication dialog may popup.

Supply a password for the username entered earlier, a popup should appear (check Linux)

Graphical user interface, application

Description automatically generated

Select “Sign in with your browser”

Graphical user interface, text, application, chat or text message

Description automatically generated

Click ‘Authorize’ and entered the current password for this GitHub account.