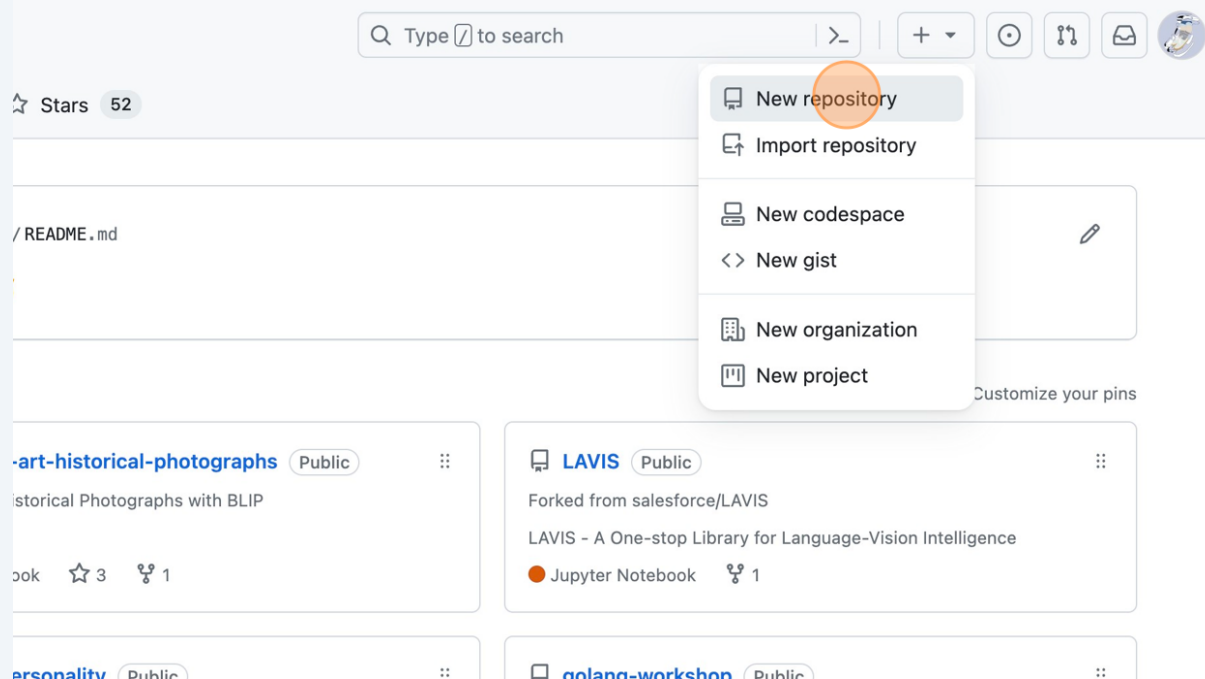


Create a GitHub repository for your portfolio & deploy it.

1 Navigate to your GitHub profile.

2 Create a new repository.



3 Give it a name.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner *

 valeriatisch ▾

Repository name *

/ 

Great repository names are short and memorable. Need inspiration? How about [fluffy-octo-adventure](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

4

a long description for your project. [Learn more about READMEs](#).

from a list of templates. [Learn more about ignoring files](#).

can and can't do with your code. [Learn more about licenses](#).

apps access to this repository

arketplace app.

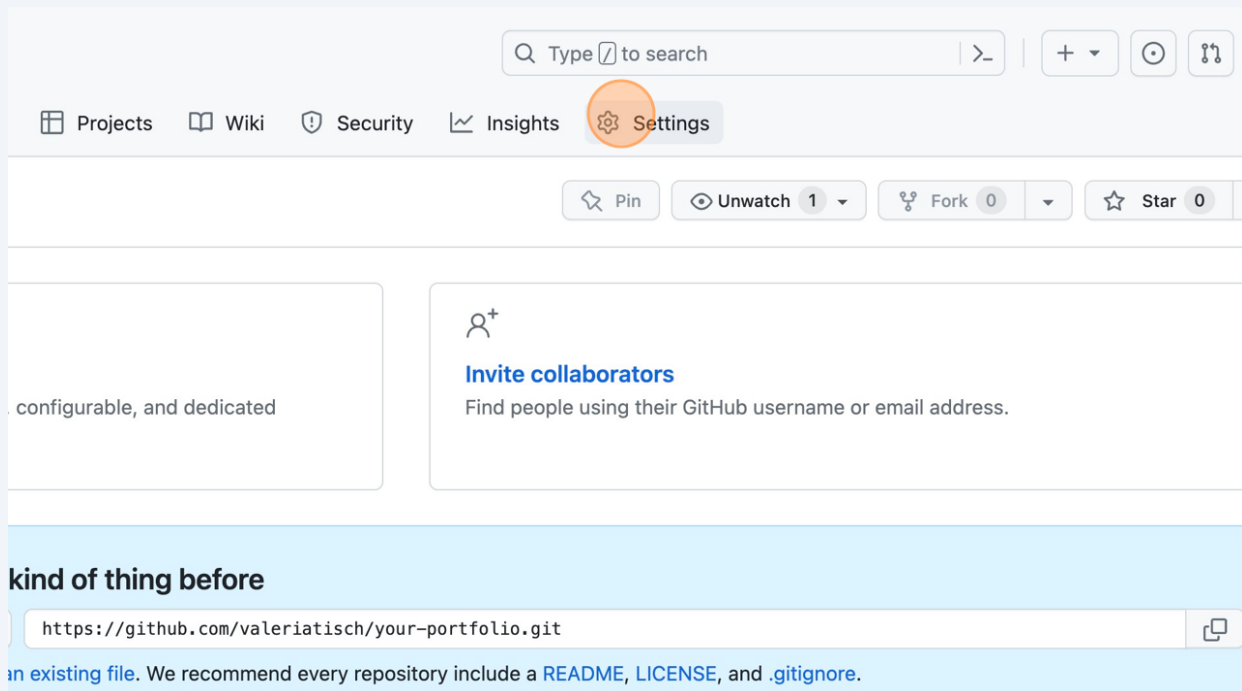
Build

ploy in a fast, consistent, and secure manner

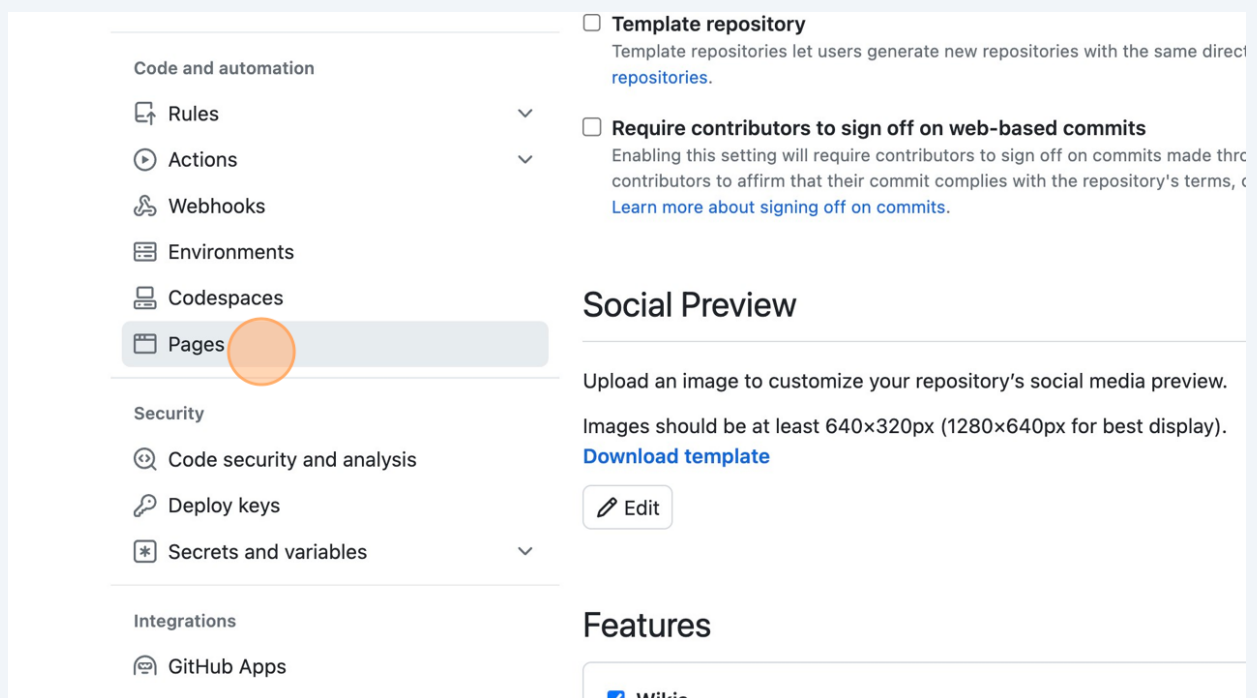
> repository in your personal account.

 Create repository

5 Go to the "Settings"



6 Click "Pages"



7 Select "GitHub Actions Beta"

General

Access

- Collaborators
- Moderation options

Code and automation

- Rules
- Actions
- Webhooks
- Environments
- Codespaces
- Pages**

Security

- Code security and analysis
- Deploy keys

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages.

Build and deployment

Source

Deploy from a branch

- GitHub Actions (Beta)**
Best for using frameworks and customizing your build process
- ✓ Deploy from a branch
Classic Pages experience

Visibility **GITHUB ENTERPRISE**

With a GitHub Enterprise account, you can restrict access to your GitHub Pages privately published site can only be accessed by people with read access to the repository. You can use privately published sites to share your internal documentation or

add content to your repository by pushing source for your site.

8 Configure for a Static HTML.

Configure

Configure

Jekyll
GitHub Actions
Package a Jekyll site.

Static HTML
By GitHub Actions
Deploy static files in a repository without a build.

Configure

flow details will appear here once your site has been deployed. [View workflow runs.](#)

Custom domain

Custom domain
Custom domains allow you to serve your site from a domain other than valeriatisch.github.io. [Learn more about configuring custom domains.](#)

9 Type 'static' for path.

```
27     environment:
28       name: github-pages
29       url: ${ steps.deployment.outputs.page_url }
30     runs-on: ubuntu-latest
31     steps:
32       - name: Checkout
33         uses: actions/checkout@v3
34       - name: Setup Pages
35         uses: actions/configure-pages@v3
36       - name: Upload artifact
37         uses: actions/upload-pages-artifact@v2
38         with:
39           # Upload entire repository
40           path: '.'
41       - name: Deploy to GitHub Pages
42         id: deployment
43         uses: actions/deploy-pages@v2
44
```

Use **Control + Shift + m** to toggle the **tab** key moving focus. Alternatively, use **esc** then **tab** to move to the next interactive element on the page.

Use **Control + Space** or **Option + Space** to trigger autocomplete in most situations.

10 Commit changes...

> + ↻ 🔗 📧 🚀

Wiki Security Insights Settings

Cancel changes Commit changes...

Spaces 2 No wrap

s tab

o GitHub Pages

Marketplace Documentation

Featured Actions

Setup Node.js environment 3.1k

By actions

Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH

Setup Go environment 1.2k

By actions

11 Click "Go to default branch"

[.github](#) / workflows / 



Error loading page

An unexpected error occurred. Try [reloading the page](#).

Go to default branch

12



valeriatisch / **your-portfolio**

Q Type  to s

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

 Code

main

+ Q

Q Go to file

t

>  .github/workflows

[your-portfolio](#) 



valeriatisch Create static.yml

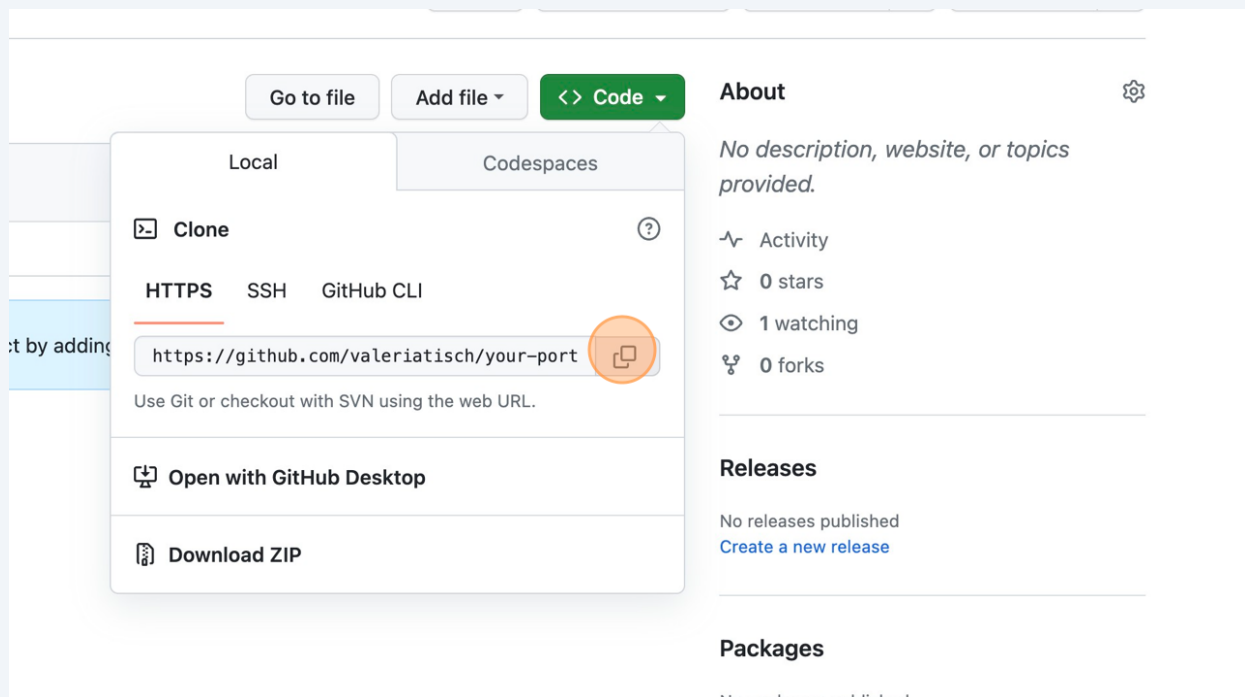
Name

Last commit mes

 .github/workflows

Create static.yr

13 Clone your repository.



Next Steps 🚀

1. Clone the Repository 📦

Open your terminal and type:

```
git clone https://github.com/username/your-portfolio.git
```

Change directory to the cloned repository:

```
cd your-portfolio
```

2. Copy Files 📁

Copy the everything from `golang-web-development/exercises/01_personal_portfolio/`:

```
cp -r ../golang-web-development/exercises/01_personal_portfolio/* .
```

3. Implement TODOs 🖥️

Open the repo in VSCode and implement the tasks.

4. Test Locally ✍️

Once you've made your changes, test your code locally by running your application.

```
go run main.go
```

A server should start and you should be able to view your portfolio in your local web browser under `localhost:8080`.

5. Commit and Push Changes 📁

Once you're satisfied with your changes and have tested everything locally, you can commit and push your changes to the remote repository.

You can check the status of your changes:

```
git status
```

Next, stage all the changes:

```
git add .
```

Now, commit the changes with a descriptive message:

```
git commit -m "implemented portfolio"
```

Finally, push your changes to the remote repository:

```
git push origin main
```


Get your link.

1

If your deployment is still running, here will be an orange dot. If it was successful, there will be this green mark. Click on it.

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with links for Issues, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, there's a search bar and buttons for Unwatch (1) and Fork (0). The main content area shows a list of commits. The first commit is highlighted with an orange circle containing a green checkmark, indicating a successful deployment. The commit message is "only publish static folder" and it was made "3 hours ago". To the right of the commit list, there's a sidebar with sections for "About" (No description, we provided.), "Activity" (0 stars, 1 watching, 0 forks), and "Releases" (No releases published).

0 tags

Go to file Add file > <> Code

re 11c9d75 3 hours ago 4 commits

only publish static folder	3 hours ago
restructure	3 hours ago
restructure	3 hours ago
restructure	3 hours ago

review of your project. Add a README

About
No description, we provided.

Activity
0 stars
1 watching
0 forks

Releases
No releases published

2 Click "Details"

The screenshot shows a GitHub repository page. At the top, there's a search bar and buttons for 'Unwatch', 'Fork', and 'Code'. Below the repository name, there are buttons for 'Go to file', 'Add file', and 'Code'. A section titled 'All checks have passed' shows a successful deployment check: 'Deploy static content to Pages / deploy (push)'. A 'Details' link is highlighted with an orange circle. To the right, there's an 'About' section with 'No description provided' and statistics like '0 stars', '1 watching', and '0 forks'. At the bottom, there's a 'Releases' section with 'No releases published' and a 'Packages' section.

3 Open "Complete job".

The screenshot shows a GitHub Actions workflow run details page. On the left, there's a sidebar with 'Run details', 'Usage', and 'Workflow file'. The main area shows a list of steps in a workflow, each with a checkmark icon. The steps are: 'Set up job', 'Checkout', 'Setup Pages', 'Upload artifact', 'Deploy to GitHub Pages', 'Post Checkout', and 'Complete job'. The 'Complete job' step is highlighted with an orange circle.

4

Click the link generated for your app, something like "https://username.github.io/your-portfolio/"

