Heck yes—your JupyterLite site is alive 🎉  
Here’s a clean, repeatable workflow you can keep (and hand to future-you when you forget a step).

**What you’ll have in the repo**

csharp

CopyEdit

my-lite/

├─ content/ # your notebooks (.ipynb)

├─ jupyter-lite.json # sets baseUrl for Pages

├─ .github/workflows/deploy.yml

└─ (everything Actions builds into \_output when it runs)

**1) The GitHub Actions workflow (copy-paste)**

Put this in .github/workflows/deploy.yml:

yaml

CopyEdit

name: Deploy JupyterLite site

on:

push:

branches: [ main ]

permissions:

contents: read

pages: write

id-token: write

concurrency:

group: "pages"

cancel-in-progress: true

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-python@v5

with:

python-version: "3.11"

- run: pip install "jupyterlite==0.6.4" "jupyterlite-pyodide-kernel==0.6.1"

# Build with the correct base url for your repo

- run: jupyter lite build --contents content --base-url /my-lite/

- uses: actions/upload-pages-artifact@v3

with:

path: \_output

deploy:

environment:

name: github-pages

url: ${{ steps.deployment.outputs.page\_url }}

runs-on: ubuntu-latest

needs: build

steps:

- id: deployment

uses: actions/deploy-pages@v4

And keep this at repo root to lock the base URL (already in your repo, but for reference):

json

CopyEdit

// jupyter-lite.json

{

"jupyter-lite-schema-version": 0,

"jupyter-config-data": {

"baseUrl": "/my-lite/"

}

}

**2) What to do in Anaconda (local)**

This part is optional (you don’t have to build locally anymore), but if you want to test locally:

powershell

CopyEdit

# Anaconda Prompt

conda activate base # or your env with Python 3.11+

pip install "jupyterlite==0.6.4" "jupyterlite-pyodide-kernel==0.6.1"

cd C:\Users\testl\my-lite

# quick local build & serve

jupyter lite build --contents content

jupyter lite serve --contents content

# open: http://127.0.0.1:8000/lab/index.html?disableSW=1

**3) What to do in PowerShell (repo & pushes)**

(you already did most of this—below is the canonical sequence)

powershell

CopyEdit

# from PowerShell

cd C:\Users\testl\my-lite

# make sure Git is initialized and on main

git init

git branch -M main

# set the correct remote (no PAT in the URL)

git remote remove origin 2>$null

git remote add origin https://github.com/Priyesh122/my-lite.git

# add files & push

git add .

git commit -m "Deploy JupyterLite with baseUrl"

git push -u origin main

**4) What to do in GitHub (one-time)**

1. Repo → **Settings** → **Pages**
   * Source: **GitHub Actions** (you’re already using it).
2. Repo → **Actions** → ensure workflows are allowed (default is fine).
3. Push to main. Actions will build + deploy automatically.
4. Open: https://priyesh122.github.io/my-lite/lab/index.html?disableSW=1

**5) Common troubleshooting (quick hits)**

**Kernel shows “No Kernel”**

* The site path must match your repo name. You did this with:
  + --base-url /my-lite/ in the build step, and
  + jupyter-lite.json with the same baseUrl.
* Hard refresh or open with service worker disabled:  
  ?disableSW=1 at the end of the URL.
* Clear cached site data (one time): DevTools → Application → Clear storage → Clear site data → reload.

**Wrong remote / push issues**

powershell

CopyEdit

git remote -v

git remote set-url origin https://github.com/Priyesh122/my-lite.git

**“Everything up-to-date” but site didn’t change**

* Check Actions logs: Repo → **Actions** → click your workflow → ensure both **build** and **deploy** are green.
* Confirm artifact was uploaded from \_output (you see it in the build job logs).
* Pages: Repo → **Settings** → **Pages** → see published URL.

**Windows credential hiccups (stuck auth)**

powershell

CopyEdit

# clear saved GitHub credentials

cmdkey /list | findstr /i github

cmdkey /delete:git:https://github.com

# GitHub CLI (if installed) – check auth

gh auth status

gh auth logout

**Confirm your Git identity and config**

powershell

CopyEdit

git config user.name

git config user.email

git config --list --show-origin

**Force a rebuild**

powershell

CopyEdit

# touch a file and push

echo "# redeploy" >> README.md

git add README.md

git commit -m "Force redeploy"

git push

**6) Useful logs & “who’s logged in” checks**

You can’t see “who is viewing the website” from GitHub Pages (no server logs; it’s static hosting). But you can see:

**Git remotes & branches**

powershell

CopyEdit

git remote -v

git branch -vv

**Git credential state (Windows)**

powershell

CopyEdit

cmdkey /list # shows stored Windows credentials (look for git:https://github.com)

git credential-manager diagnose # if Git Credential Manager is installed

**GitHub CLI sessions (optional, if gh installed)**

powershell

CopyEdit

gh auth status

gh auth token # prints token scope (don’t share)

**Actions run history**

* Repo → **Actions** → you can see who pushed, workflow status, and full build logs.

If you want a belt-and-suspenders version of the workflow with a cache to speed up installs, say the word and I’ll drop that in. Otherwise, you’re set: push to main, Actions builds to \_output, deploys to Pages, and the Pyodide kernel loads correctly under /my-lite/.

Ask ChatGPT

Top of Form

Bottom of Form

[**https://chatgpt.com/share/6895cb36-08dc-8000-b8b9-5fa30c1de76d**](https://chatgpt.com/share/6895cb36-08dc-8000-b8b9-5fa30c1de76d)

############################################

cd C:\Users\testl\my-lite

git init

git branch -M main

git remote set-url origin https://Priyesh122:github\_pat\_11BKTAE7Y0GpMdsrHxdwZc\_n7K6nh5rmgq6I1Wbg4cjsuxNU4syOriAygwiNCGVW0Z66L5PJTVxdeCFR513@github.com/Priyesh122/my-lite.git

git add .

git commit -m "Initial JupyterLite build"

git push -u origin main