



## CI/CD Lab — Azure Static Web Apps (GUI Only)

### Goal

Build and automatically deploy a simple static site (`index.html`) from GitHub → Azure Static Web App — fully automated with CI/CD.

---

## STEP 1 — Create the GitHub Repository

1. Go to <https://github.com/new>.
  2. **Repository name:** ci-cd-lab
  3. Keep visibility → **Public** (or Private).
  4.  Check “Add a README file”
  5. Click **Create repository**.
- 

## STEP 2 — Add `index.html` (directly in GitHub)

1. In your new repo → click “**Add file**” → “**Create new file**”
2. **File name:**  
`index.html`
3. Paste this code:  
5. `<!DOCTYPE html>`  
6. `<html lang="en">`  
7.  `<head>`  
8.  `<meta charset="UTF-8" />`  
9.  `<meta name="viewport" content="width=device-width, initial-scale=1.0" />`  
10.  `<title>CI/CD Lab – Azure Static Web App V!</title>`  
11.  `<style>`  
12.  `body {`  
13.  `font-family: system-ui, sans-serif;`  
14.  `background: #f5f7fa;`  
15.  `color: #222;`  
16.  `display: flex;`  
17.  `flex-direction: column;`  
18.  `justify-content: center;`  
19.  `align-items: center;`  
20.  `height: 100vh;`  
21.  `}`

```

22.      h1 {
23.          font-size: 2rem;
24.          margin-bottom: 0.5rem;
25.      }
26.      p {
27.          font-size: 1.1rem;
28.      }
29.      footer {
30.          margin-top: 2rem;
31.          font-size: 0.9rem;
32.          color: #555;
33.      }
34.  </style>
35. </head>
36. <body>
37.     <h1>📌 CI/CD Demo __: Azure Static Web App</h1>
38.     <p>This site automatically redeploys on every push to
<code>main</code>. </p>
39.     <footer>
40.         <small><!-- COMMIT_SHA --></small>
41.     </footer>
42. </body>
43. </html>

```

44. Scroll down → click **Commit new file**.

---

## ☰ STEP 3 — Create the Azure Static Web App (GUI)

1. Go to the [Azure Portal](#).
  2. Click **Create a resource** → search **Static Web App** → click **Create**.
  3. Fill in:
    - **Subscription:** your default
    - **Resource Group:** create new → ci-cd-lab-rg
    - **Name:** ci-cd-lab-demo
    - **Hosting plan:** Free
    - **Region:** choose your nearest region
  4. In **Deployment details:**
    - **Source:** GitHub
    - **Organization:** select your GitHub account
    - **Repository:** select ci-cd-lab
    - **Branch:** main
  5. In **Build Details:**
    - **App location:** /
    - **API location:** (leave empty)
    - **Output location:** /
  6. Click **Review + Create** → **Create**.
-

## STEP 4 — Azure Automatically Creates CI/CD Workflow

 Once deployment finishes:

- Azure automatically adds a workflow file in GitHub:
- `.github/workflows/azure-static-web-apps-<env>.yml`
- You can find it under **GitHub → Actions** tab.

That file handles **build + deploy on every push** — no CLI needed.

---

## STEP 5 — Wait for GitHub Action to Run

1. In your repo → go to **Actions** tab.
  2. You'll see a workflow named “**Azure Static Web Apps CI/CD**” running.
  3. Wait ~1 minute → you'll see  **success**.
- 

## STEP 6 — Visit Your Live Website

1. Go back to **Azure Portal** → **Your Static Web App** → **Overview**.
  2. Under “**URL**”, click the link — it looks like:
  3. `https://<random-name>.azurestaticapps.net`
  4. You'll see your deployed site   
*(the commit SHA won't show yet — that comes next)*
- 

## STEP 7 — Inject the Commit SHA (optional enhancement)

You can manually edit the workflow file in GitHub to inject the commit SHA.

1. In GitHub, open `.github/workflows/azure-static-web-apps-<something>.yml`
2. Click **Edit** 
3. Replace its content with this (copy–paste):
  4. `name: Azure Static Web Apps CI/CD`
  5. `on:`
  6. `push:`

```

8.      branches:
9.        - main
10.     pull_request:
11.       types: [opened, synchronize, reopened, closed]
12.     branches:
13.       - main
14.
15.   jobs:
16.     build_and_deploy_job:
17.       if: github.event_name == 'push' || github.event.action == 'closed'
18.       runs-on: ubuntu-latest
19.       name: Build and Deploy Job
20.       steps:
21.         - uses: actions/checkout@v4
22.
23.         - name: Inject commit SHA into HTML
24.           run: |
25.             sed -i "s|<!-- COMMIT_SHA -->|Commit: ${GITHUB_SHA::7}|"
index.html
26.
27.         - name: Build And Deploy
28.           id: builddeploy
29.           uses: Azure/static-web-apps-deploy@v1
30.           with:
31.             azure_static_web_apps_api_token: ${{ secrets.AZURE_STATIC_WEB_APPS_API_TOKEN_ASHY_WATER_035D99E1E }}
32.             repo_token: ${{ secrets.GITHUB_TOKEN }}
33.             action: "upload"
34.             app_location: "/"
35.             api_location: ""
36.             output_location: "/"
37. Click Commit changes.
38. GitHub Action will run again automatically.
39. Refresh your site → the footer will now show
40. Commit: <short-hash>

```

---

## STEP 8 — Test CI/CD

1. Edit `index.html` (add or change text).
  2. Commit your change.
  3. GitHub → Actions → workflow runs again automatically.
  4. Refresh your live site → new version appears!
- 

## RESULT

 **Full CI/CD pipeline — no CLI, no terminal.**

Every time you edit files on GitHub → Azure redeploys automatically.  
You can see the commit ID, and you can add preview slots later if needed.

---

---

## **Bonus Step — Enable Preview Environments for Pull Requests**

Preview environments let you deploy **every Pull Request (PR)** to a **temporary live URL** (like <https://pr-7-yourappname.azurestaticapps.net>) automatically.  
This is great for testing features before merging.

---

### **Step-by-Step (All GUI)**

1. **Go to your Azure Static Web App**
  - In the [Azure Portal](#), search for and open your **Static Web App** resource.
2. **Navigate to "Configuration" → "General" → "Preview environments"**
  - Look for the option: **Enable preview environments**
  - Toggle it **ON**
3. **Save** your changes.
4. **Now go to your GitHub repo**
  - Click **Pull Requests** → **New Pull Request**
  - Choose a test branch (you can create one directly in the GitHub UI by editing a file → "Create a new branch").
5. **Submit the Pull Request**
  - Once submitted, go to the **Checks tab** or your **GitHub Actions tab**.
  - You'll see a new workflow run called **Azure Static Web Apps CI/CD**.
6. **Wait for deployment (1–2 minutes)**
  - After completion, scroll down in the PR → you'll see a **comment from Azure Static Web Apps** like:
  - Your app is available at:
  - <https://pr-7-yourappname.azurestaticapps.net>
  -  Click it to see your preview deployment!

---

### **Validation**

<b>Check</b>	<b>Expected Result</b>
PR opened	A new deployment starts automatically

Check	Expected Result
PR page comment	Shows unique preview URL
Visiting URL	Shows your app with the new change
Closing or merging PR	Preview environment automatically deleted

---

## Troubleshooting

Issue	Cause	Fix
No preview environment created	Feature disabled	Make sure “Enable preview environments” is turned ON in Azure portal
GitHub workflow failed	YAML syntax or missing token	Check your workflow YAML under <code>.github/workflows/azure-static-web-apps.yml</code> for errors
Preview URL not found	Build not finished	Wait 1–2 minutes, then refresh PR
Preview not deleted after merge	Azure delay	Wait a few minutes or manually delete old previews in Azure portal

Excellent — you’ve finished the CI/CD setup, so the next professional step is to build a **troubleshooting & validation manual**. Below is a **complete diagnostic and validation checklist**, written like a real DevOps playbook, made 100% for **GUI (no CLI)**.

---

## CI/CD Lab — Troubleshooting & Validation Manual (GUI-Only)

### Table of Contents

1.  Quick Validation Steps
2.  Detailed Troubleshooting by Category
  - o A. GitHub Workflow Issues
  - o B. Azure Deployment Issues
  - o C. Website Not Updating
  - o D. Secret / Token Problems
  - o E. Build Path / Artifact Issues
3.  Validation Scenarios

- 
4. 🚧 Recovery & Reset Steps
  5. 🔍 Best Practices for Long-Term Stability
- 

## ✓ 1. Quick Validation Steps

After each deployment, verify these **five key indicators**:

Step	What to Check	Where	Expected
1	<b>GitHub Action triggered</b>	GitHub → Actions tab	“Azure Static Web Apps CI/CD” job runs automatically
2	<b>Workflow succeeded</b>	Inside the Action run	Green ✅ “Completed successfully”
3	<b>Deployment listed in Azure</b>	Azure → Static Web App → “Environments”	One environment named <i>production</i>
4	<b>Site updated</b>	Visit your <code>.azurestaticapps.net</code> URL	Matches latest HTML commit
5	<b>Commit SHA visible (if added)</b>	Footer of site	Shows short hash like Commit: <code>a1b2c3d</code>

If any of these fail — continue below ⏵

---

## ⌚ 2. Detailed Troubleshooting

### ✖ A. GitHub Workflow Issues

- ◊ *Problem 1: Workflow didn't start after push*

**Possible causes:**

- Workflow file not in `.github/workflows/`
- Branch name mismatch (main vs master)
- GitHub Actions disabled

**Fix:**

1. Go to **Repository** → **Settings** → **Actions** → **General**
  2. Ensure “Allow all actions and reusable workflows” is enabled.
  3. Verify the file name follows:  
4. `.github/workflows/azure-static-web-apps-xxxx.yml`
  5. Ensure the trigger branch matches your default branch (e.g. `main`).
- 

◊ *Problem 2: Workflow started but failed*

**Possible causes:**

- Token expired or missing
- Invalid app location / output location
- Syntax error in YAML file

**Fix:**

1. Click the **failed job** → expand the step with .
  2. Read the red error text carefully — usually mentions the problem.
  3. Most common fix:
    - Check the `azure_static_web_apps_api_token` secret exists under **Repo** → **Settings** → **Secrets** → **Actions**
    - If missing → recreate it (see Section D below).
- 

## **B. Azure Deployment Issues**

◊ *Problem: Deployment didn't show up or is missing*

1. Go to **Azure Portal** → **Static Web App** → **Environments** tab.
2. Ensure you see “production” listed.
  - If not: the GitHub Action may have failed to deploy.
3. Check **Azure Activity Log** (left sidebar) → filter by “Static Web Apps Deploy”.

If nothing appears, delete and re-link the GitHub repo:

- Azure → Static Web App → **Configuration** → **Deployment Configuration**
  - Click **Disconnect repo**
  - Then click **Connect repo** again → reselect `ci-cd-lab`.
- 

## **C. Website Not Updating**

◊ *Symptoms:*

- You push changes → GitHub Action success  → but old version still appears.

**Possible causes:**

1. **Browser cache / CDN cache**
2. **Preview slot accidentally selected**
3. **App artifact path mismatch**

**Fix sequence:**

1. Open site in **private/incognito** window or add ?refresh=1 to URL.
  2. In Azure Portal:
    - Go to your Static Web App → **Configuration** → ensure:
      - “Enable configuration file changes” = ON
      - “Enable preview environments” = ON (optional)
  3. If still old:
    - Under “Deployment configuration” → choose **Production slot** explicitly.
  4. In GitHub workflow YAML:
    - Ensure `app_location: "/"` and `output_location: "/"`.
- 

## D. Secret / Token Problems

◊ *When needed*

Azure adds the token automatically during setup.

If it's deleted or the workflow was edited manually, you may need to restore it.

**Check:**

1. In GitHub → Settings → Secrets → Actions
  - You should see a key like:  
`AZURE_STATIC_WEB_APPS_API_TOKEN_XXXX`

**If missing:**

1. Go to **Azure Portal** → **Static Web App** → **Overview**
2. Click **Manage deployment token**
3. Copy the token
4. Back in GitHub → **Settings** → **Secrets** → **Actions** → **New repository secret**
  - Name: same as used in your YAML file
  - Value: paste the token

- 
5. Re-run workflow (click “Re-run failed jobs”).

---

## E. Build Path / Artifact Issues

- ◊ *Problem: GitHub Action says “No content found to deploy”*

**Cause:** app\_location or output\_location paths are incorrect.

**Fix:**

1. For simple HTML repo → both should be /
2. Example YAML:
  3. app\_location: "/"
  4. api\_location: ""
  5. output\_location: "/"
6. Commit and push again.

---

## 3. Validation Scenarios

Run these checks after setup or troubleshooting:

#	Test	Expected Result
1	Modify text in <code>index.html</code>	New version visible on site
2	Add comment inside HTML	Redeployment triggers automatically
3	Open Actions tab	One new run per commit
4	Delete workflow file	Deployment stops (confirming dependency)
5	Restore file	Deployment resumes

---

## 4. Recovery & Reset Steps

If your deployment is too corrupted to fix:

### Option 1 — Redeploy Clean

1. Delete your Azure Static Web App (keep RG).
2. Create a **new Static Web App** with same GitHub repo.
3. Azure will automatically recreate a clean workflow file.

## Option 2 — Recreate Repo

1. Backup your `index.html`.
  2. Delete the repo.
  3. Create new repo → repeat setup wizard.
  4. Faster and 100% clean.
- 



## 5. Best Practices for Stability

Category	Recommendation
<b>Branches</b>	Always deploy from <code>main</code>
<b>Commits</b>	Use small commits — easier rollback
<b>Workflows</b>	Don't rename the <code>.yml</code> file manually
<b>Secrets</b>	Don't delete tokens from Azure Portal
<b>Cache</b>	Test with incognito window after each deployment
<b>Backups</b>	Export YAML & <code>index.html</code> once it's working
<b>Validation</b>	Use "Actions → Re-run all jobs" if deployment seems stuck

---



## Final Validation Checklist (Production-Ready)

Check	Status
Repo exists and public	<input type="checkbox"/>
<code>index.html</code> loads from GitHub	<input type="checkbox"/>
Workflow file in <code>.github/workflows/</code>	<input type="checkbox"/>
GitHub Action runs on push	<input type="checkbox"/>

Check	Status
Azure portal shows latest deployment	<input type="checkbox"/>
Site updates instantly after commit	<input type="checkbox"/>
Commit SHA visible (optional)	<input type="checkbox"/>

---