ChatGPT

If you want to keep the deployments to the new account repository **manual and separate**, you can configure a different workflow that triggers only via a custom command or on-demand. This setup ensures that actions in the testing repository do not automatically propagate to the new account repository.

Revised Workflow for Manual Push and Deployment

Create a new workflow in your testing repository specifically for deploying to the new account repository. This workflow will be triggered only when you manually run a command or use the GitHub Actions UI.

Workflow File: `.github/workflows/manual-deploy.yml`

```yaml
name: Manual Deploy to New Account Repository

on:
  workflow_dispatch: # Allows manual triggering via GitHub Actions UI
    inputs:
      branch:
        description: 'Branch to deploy (main or gh-pages)'
        required: true
        default: 'main'

jobs:
  deploy_to_main:
    if: ${{ github.event.inputs.branch == 'main' }}
    name: Push Changes to Main Branch in New Repository
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Code
        uses: actions/checkout@v3

      - name: Push to Main Branch in New Repository
        env:
          TOKEN: ${{ secrets.NEW_ACCOUNT_PAT }}
        run: |
          git remote add production https://${TOKEN}@github.com/<new-account>/<new-
repo>.git
          git push production main:main

  deploy_to_gh_pages:
    if: ${{ github.event.inputs.branch == 'gh-pages' }}
    name: Deploy Dist Folder to gh-pages Branch in New Repository
    runs-on: ubuntu-latest

    steps:
      - name: Checkout Code
        uses: actions/checkout@v3

      - name: Install Dependencies
        run: npm install

      - name: Build Project
        run: npm run build

      - name: Deploy Dist Folder to gh-pages
        env:
          TOKEN: ${{ secrets.NEW_ACCOUNT_PAT }}
        run: |
          git remote add production https://${TOKEN}@github.com/<new-account>/<new-
repo>.git
          git fetch production
          git switch --orphan gh-pages
          git rm -rf .
          cp -r dist/* .
          git add .
          git commit -m "Deploy to gh-pages"
          git push --force production gh-pages
```

Key Features of the Workflow:

1. **Trigger Manually**:

   - The `workflow_dispatch` event allows you to trigger this workflow manually via the GitHub Actions UI or the GitHub CLI.
2. **Branch-Specific Deployment**:
   - You can specify which branch (`main` or `gh-pages`) to deploy via an input parameter when triggering the workflow.
3. **Two Separate Jobs**:
   - `deploy_to_main`: Pushes the `main` branch to the `main` branch in the new account repository.
   - `deploy_to_gh_pages`: Builds the project and deploys the `dist` folder to the `gh-pages` branch in the new account repository.

How to Trigger the Workflow Manually:

1. Go to the **Actions** tab in your testing repository.
2. Select the **Manual Deploy to New Account Repository** workflow.
3. Click **Run Workflow**.
4. Choose the branch to deploy (`main` or `gh-pages`) and confirm.

Command-Line Trigger (Optional)

If you'd like to trigger the workflow via the GitHub CLI, use:

```bash
gh workflow run manual-deploy.yml --ref main --field branch=main
```

Replace `branch=main` with `branch=gh-pages` to deploy the `dist` folder.

Summary

This setup ensures:

- **Separation of workflows**: Deployments to the new account repository only happen when you explicitly trigger the workflow.
- **Flexibility**: You can manually deploy the `main` branch or `dist` folder (`gh-pages`) to the new repository as needed.
- **No automatic triggers**: Actions in the testing repository won't affect the new account repository unless you manually initiate the workflow.

Let me know if you need further customization!