# GITHUB ACTION

<https://github.com/harishneel1/next-js-app>

<https://www.youtube.com/watch?v=r-iLBNaCTDk>

Your code should get automatically deployed to docker on merge

A screenshot of a computer

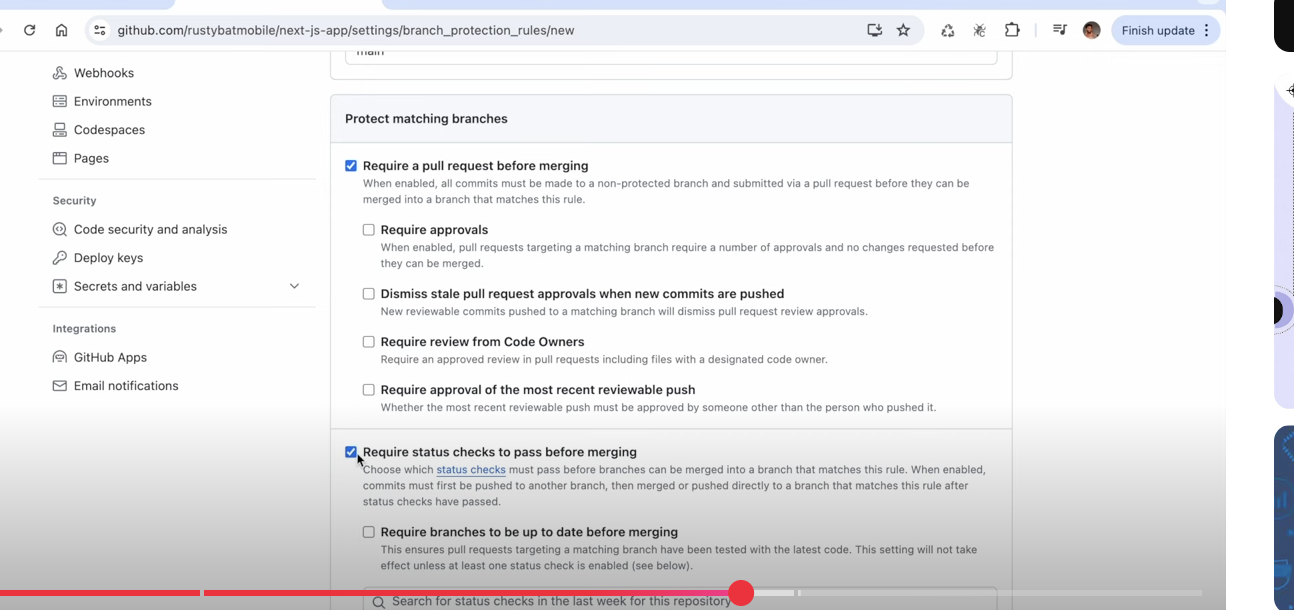
AI-generated content may be incorrect.

Workflow – multiple jobs run together

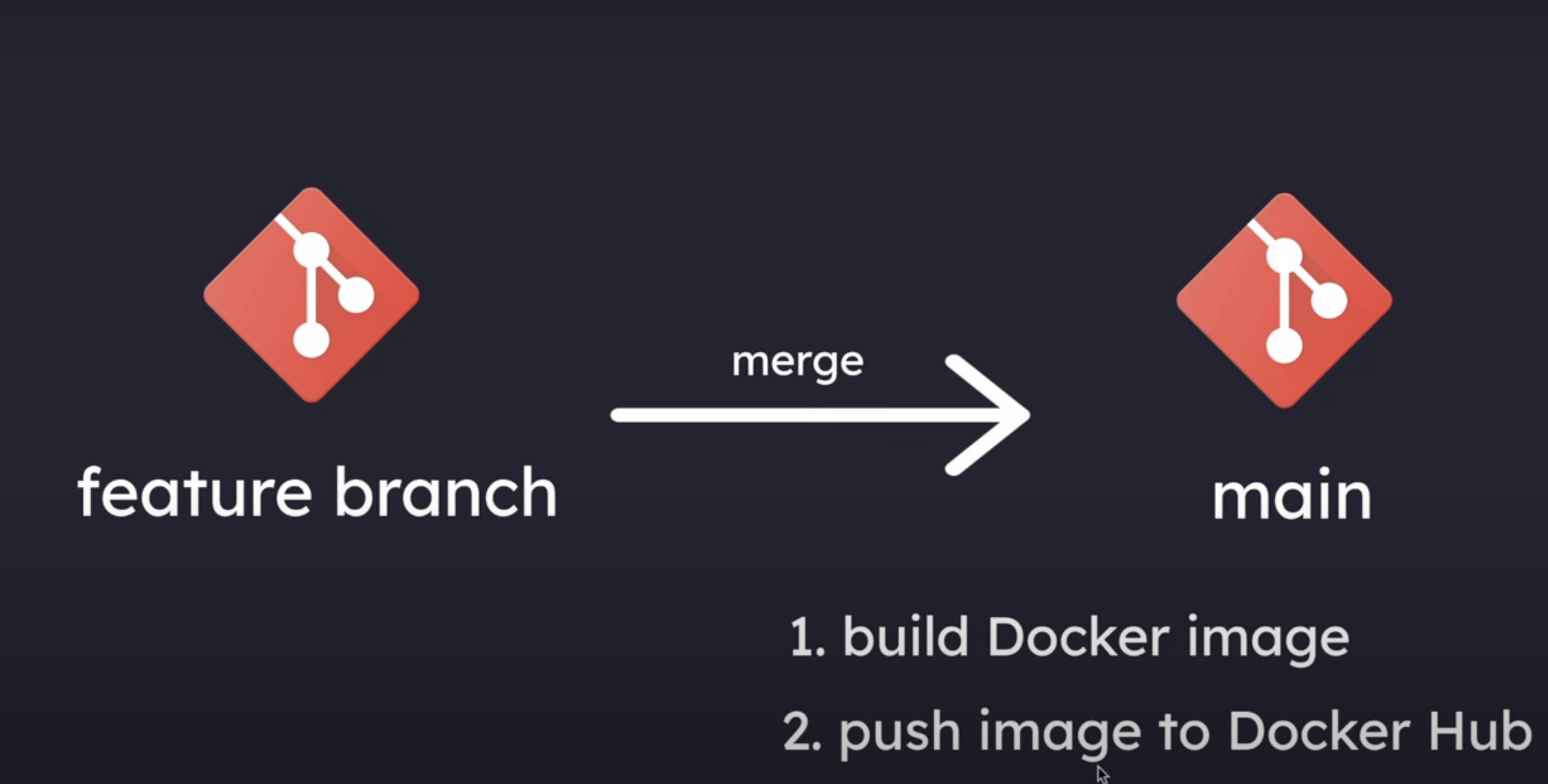
actions/checkout@v4 – checkout to feature branch coz by default it is in main branch

Also add branch protection rule in git to restrict merge if build fails  
A screenshot of a computer

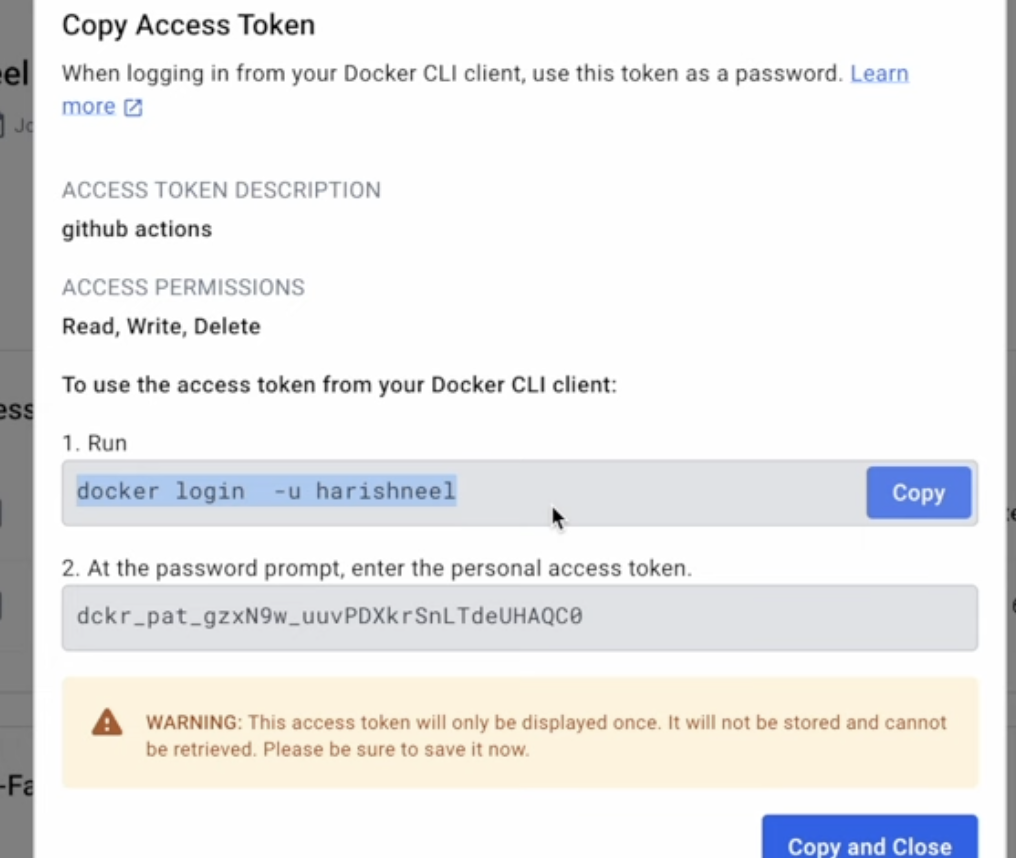
AI-generated content may be incorrect.



Job after merge in deploy.yaml,remove node setup steps coz while building dockerimage,dockerfile has all steps how to setup node etc

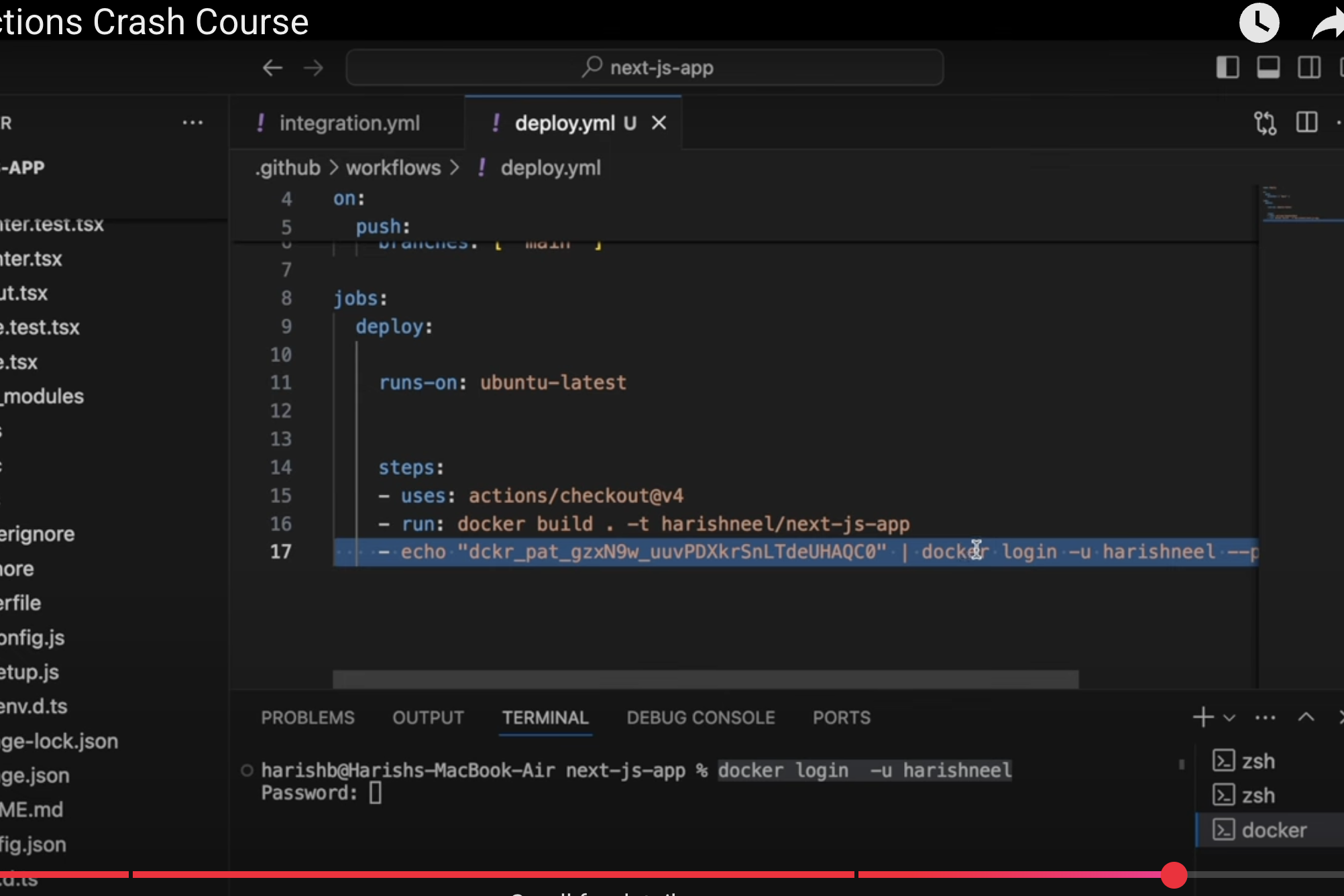


U nedd acees ytoken ito push in docker hub



As user is not rquired top exlicity enter password while in docker hub through job

So use echo in job file to put password before pipe to password stdin argument to 2nd cmd after pipe



Move poassword to github->setting->secrets->action->repository secret

And use same secret in workflow file

A computer screen with text

AI-generated content may be incorrect.

Now u can see docker image will be build once u merge code in githiub action ,deploy jobs will be there

Setting up the Node.js environment in a GitHub Actions workflow file for a build process is necessary because the GitHub-hosted runners are generic virtual machines that do not inherently have a specific Node.js version pre-configured for your project's needs.

Here's a breakdown of why this setup is crucial:

* **Version Control and Consistency:**

Different Node.js projects may require different versions of Node.js. Using the actions/setup-node action allows you to explicitly define the Node.js version needed for your build, ensuring consistency and preventing issues that might arise from using an incompatible or outdated version.

* **PATH Configuration:**

The actions/setup-node action not only installs the specified Node.js version but also correctly adds its binaries to the system's PATH environment variable. This ensures that subsequent steps in your workflow, such as running npm install or npm run build, can correctly locate and execute Node.js and npm commands.

* **Dependency Management:**

Node.js projects heavily rely on npm or Yarn for dependency management. Setting up the Node.js environment ensures that the correct package manager is available and configured to install your project's dependencies before the build process begins.

* **Optimized Caching:**

The actions/setup-node action can leverage GitHub Actions' caching mechanisms to store and reuse downloaded Node.js versions and npm/Yarn caches, significantly speeding up subsequent workflow runs.

In essence, explicitly setting up the Node.js environment in your GitHub workflow provides a controlled and predictable environment for your Node.js builds, guaranteeing that your project runs with the correct tools and configurations every time.

A screenshot of a computer

AI-generated content may be incorrect.

# K8s

When u deploy k8s u get a cluster