CICD Pipeline Explanation

Here are the steps I undertook to create my CICD pipeline for the production branch:

1. I first set up a digital ocean server, this is an ubuntu 22.04LTS server that will in future steps pull my code automatically from the repository, build it, test it, and deploy it.
2. I created SSH keys so that I can pull stuff from my private repository and added them to my digital ocean server.
3. I then set up my development environment in Ubuntu Server. This involved downloading node, angular and all the other dependencies for my web application, setting up the databases needed in postgresql and installing the links programming language.
4. I then installed nginx, which will be used to route traffic coming from trylinks.com and trylinks.com/api to different ports, one for the frontend and one for the backend.
5. I created a special linux user with limited privileges that will be responsible for executing the CICD pipeline via github actions.
6. I followed the Github Tutorial under my repository -> settings -> actions -> runners, in order to add a self hosted runner to my digital ocean server.
7. I created my github actions workflow, which you can view under .github/workflows in the repo, which upon a merge request or commits to the production branch executes.
8. I set up a service on the server so that the github action listener is always active and restarts when needed.
9. I installed pm2 which is used to run node applications and is a process manager with a built in load balancer.
10. I created a pm2 process labeled as 0 which runs the node application.
11. I created a bunch of github secrets and variables that are needed to run the application under repo -> settings -> environments -> production -> [3 secrets, 3 variables].
12. Now, when the github workflow is activated, the following steps occur

    -> We checkout the production branch
    -> NVM is used to select a certain node version
    -> We setup our paths as needed by the webapp
    -> cd into the client folder (angular app)
    -> run npm i to install the dependencies
    -> run unit tests on FE
    -> run npm build in order to build the angular application
    -> cd into the server folder (npm app)
    -> run nmp i to install the dependencies
    -> run pm2 stop 0 to stop the previous server instance
    -> run pm2 start 0 to start this server instance with the new code addition
    -> save the pm2 configuration
    -> restart nginx
    -> run e2e tests