

Aufgabe: CI-CD, github, react, docker hub: Als erstes stellte ich das Repository bereit und dann klonete ich das Projekt in mein Repository. Dann habe ich ein Dockerfile gemacht, welches ich von einem ehemaligen Projekt kopierte.

```
yannickschlaepferb Rename DOCKERFILE to Dockerfile ✓ 3 weeks ago ⌚

20 lines (14 loc) · 456 Bytes

Code Blame Raw Copy Download Edit View Source

1 # Specify the base image
2 FROM node:14-alpine
3
4 # Set the working directory inside the container
5 WORKDIR /app
6
7 # Copy package.json and package-lock.json to the working directory
8 COPY package*.json ./
9
10 # Install dependencies
11 RUN npm install
12
13 # Copy the rest of the application code to the working directory
14 COPY . .
15
16 # Expose the port that the React development server will be running on
17 EXPOSE 3000
18
19 # Start the React development server
20 CMD ["npm", "start"]
```

Dann sah ich mir ein tutorial zu docker registry und github actions an.

YouTube CH github container registry

Build & Push Docker Image to DockerHub Using GitHub Actions..

Effortless Docker Image Deployment: Automating Builds and Pushes to Docker Hub with Github Actions

Cloud Quick Lab 11.700 Abonnenten Mitglied werden Abonnieren 84 4 Teilen Herunterladen

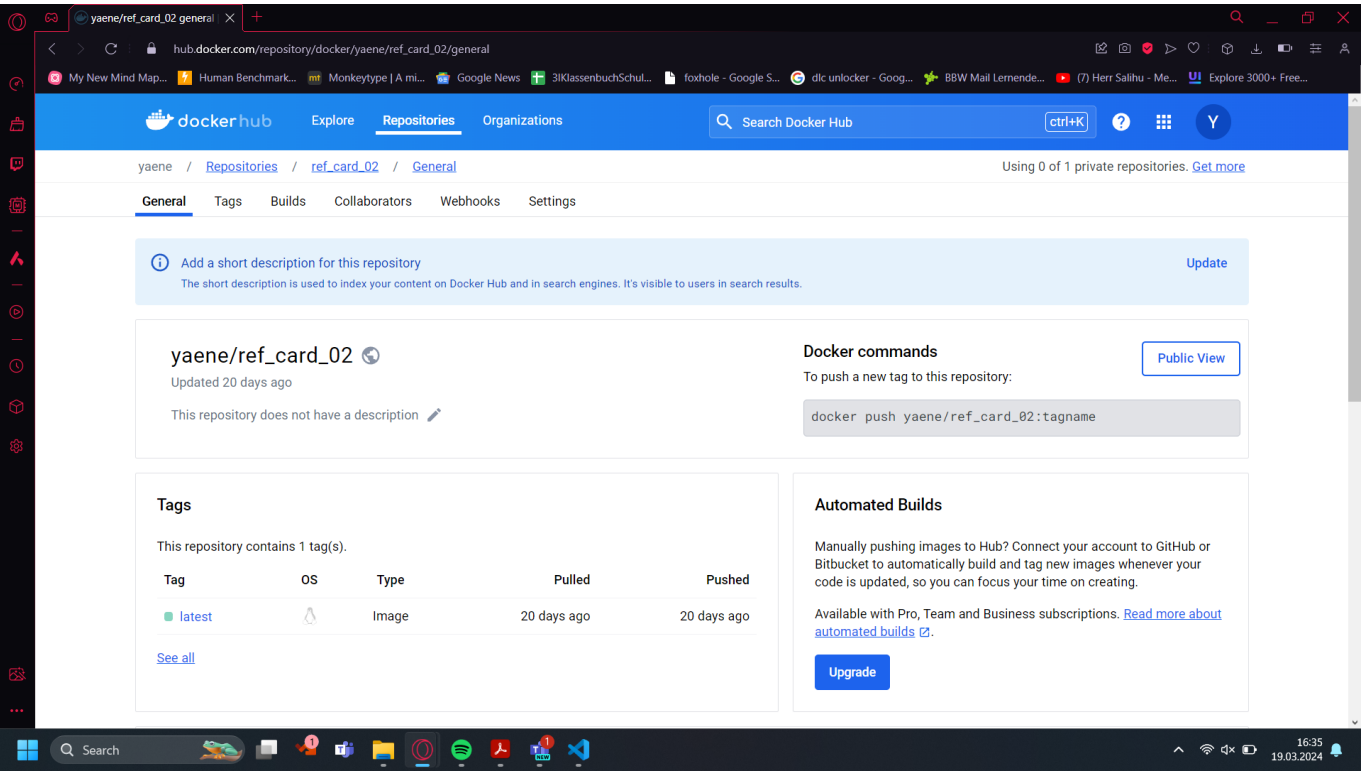
Jetzt LINDOR Elii entdecken Gesponsert · lindt.ch

Kaufen

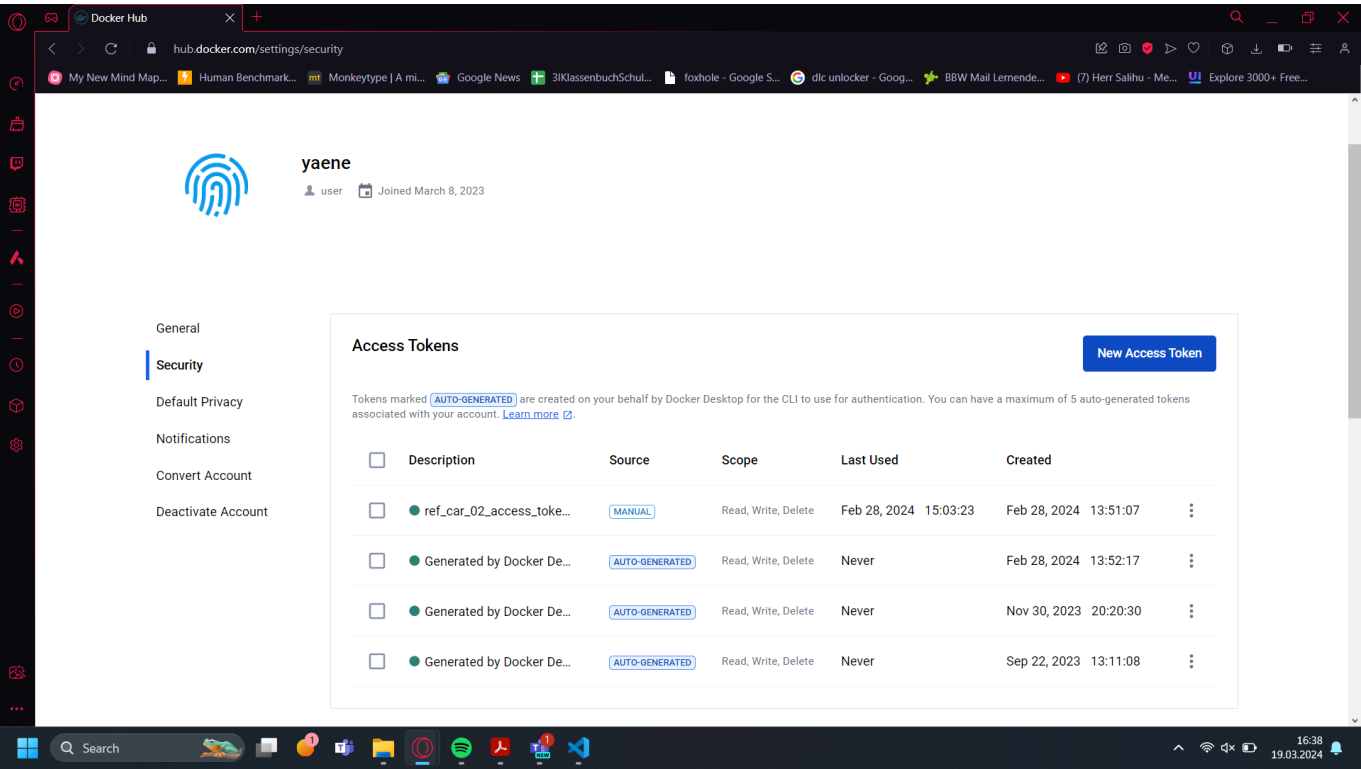
Alle Aus deiner Suche Ähnlich Kürzlich

Lernen Sie, häufige Fehler zu vermeiden Lernen Sie alte und neue Erfolgsgeheimnisse für Ihren Vertriebs Erfolg. Gesponsert · Salesforce

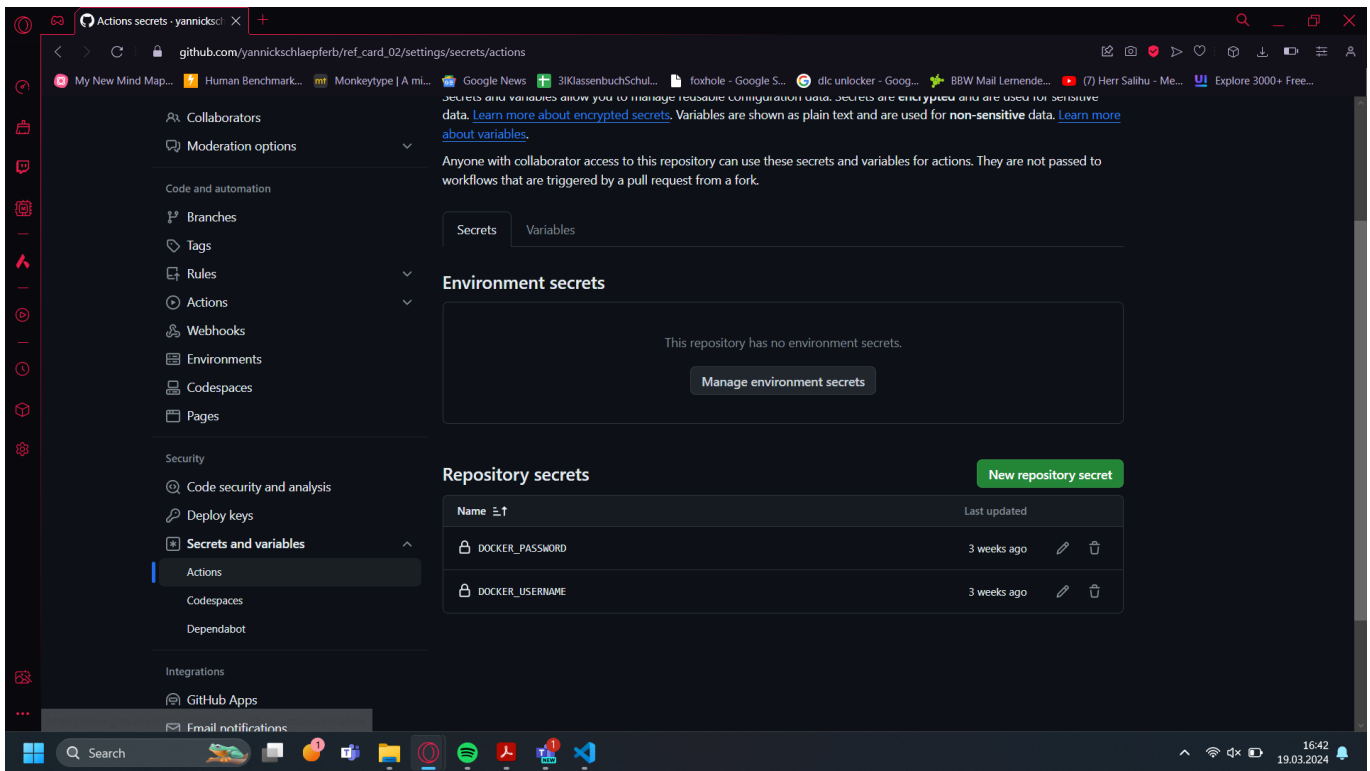
Folglich ging ich vor wie im Tutorial. Ich erstellte ein Repository auf Dockerhub.



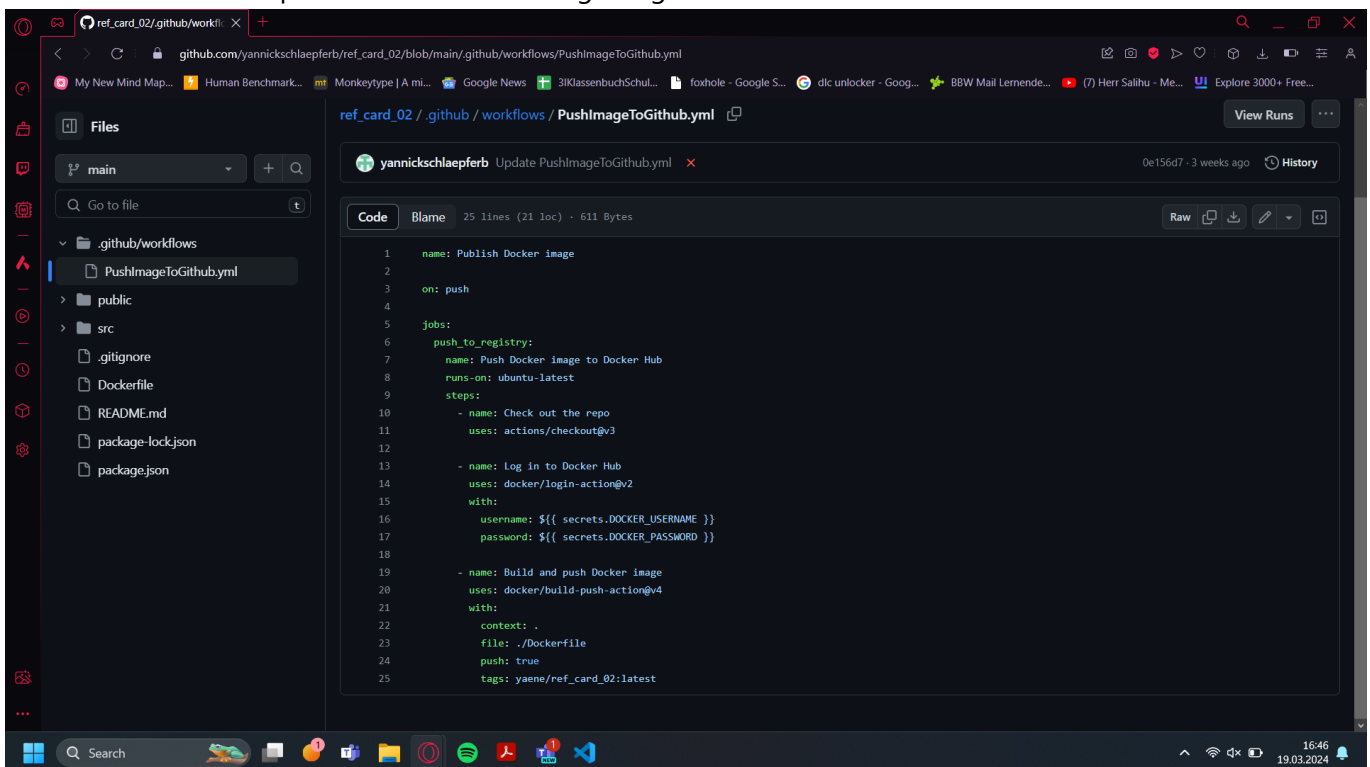
Danach ging ich auf mein Profil, dann auf Security und generierte einen Access Token.



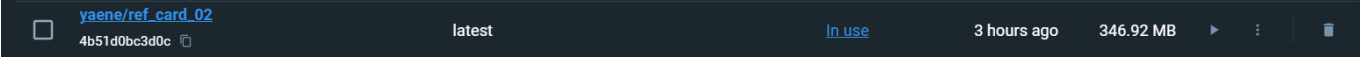
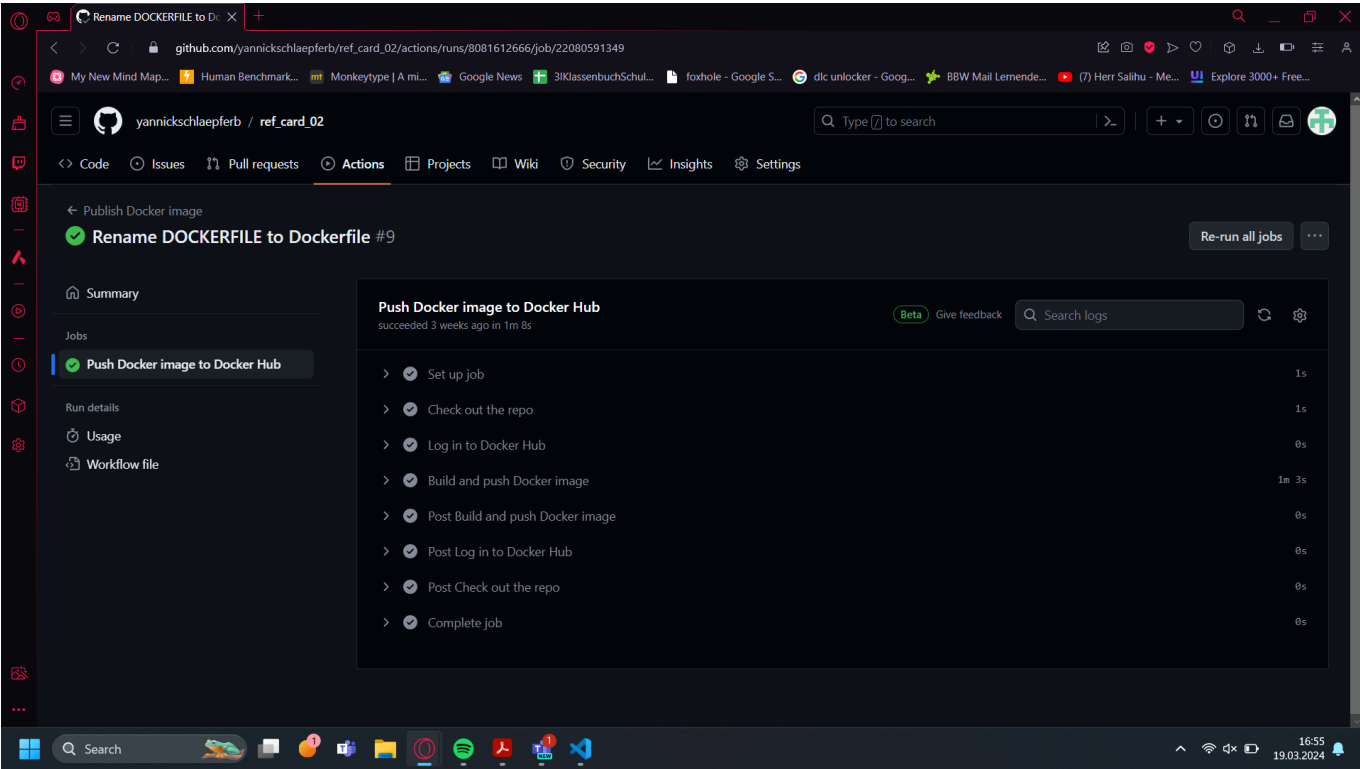
Zunächst ging ich auf Github, Einstellungen und dann Secrets. Dort erstellte ich die Secrets Docker_Password und Docker_Username.



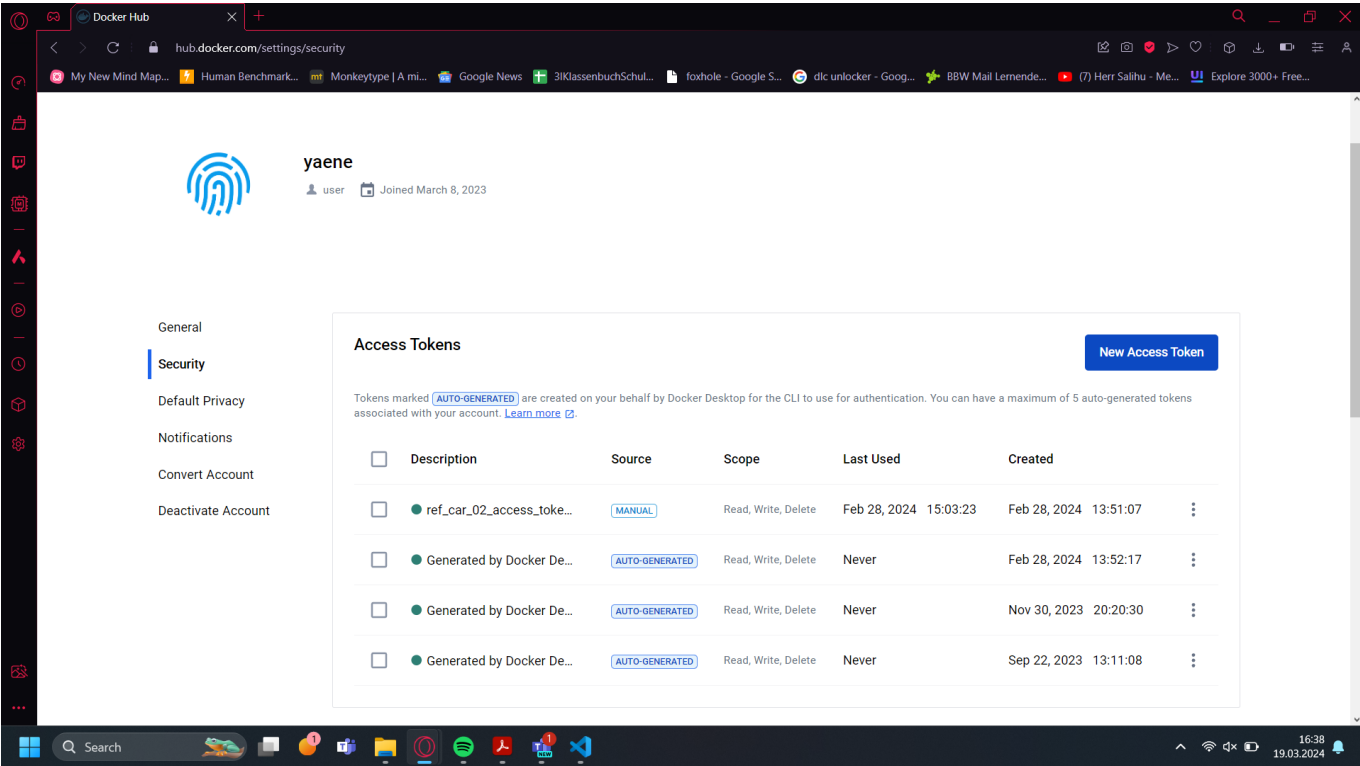
Dann machte ich die .github/workflows Ordner. Darin erstellte ich das PushImageToGithub.yml und kopierte den Inhalt vom Tutorial. Ich kürzte die einen Sachen die ich nicht brauchte und benannte username und password, auch den filepath und tags zu was entsprach. Ich hatte eine andere Ordnerstruktur und tagnamen. Aufgrund der anderen Ordnerstruktur konnte ich nicht den path vom Tutorial übernehmen und musste fünf mal committen und erproben wie es der richtige weg zum dockerfile ist.



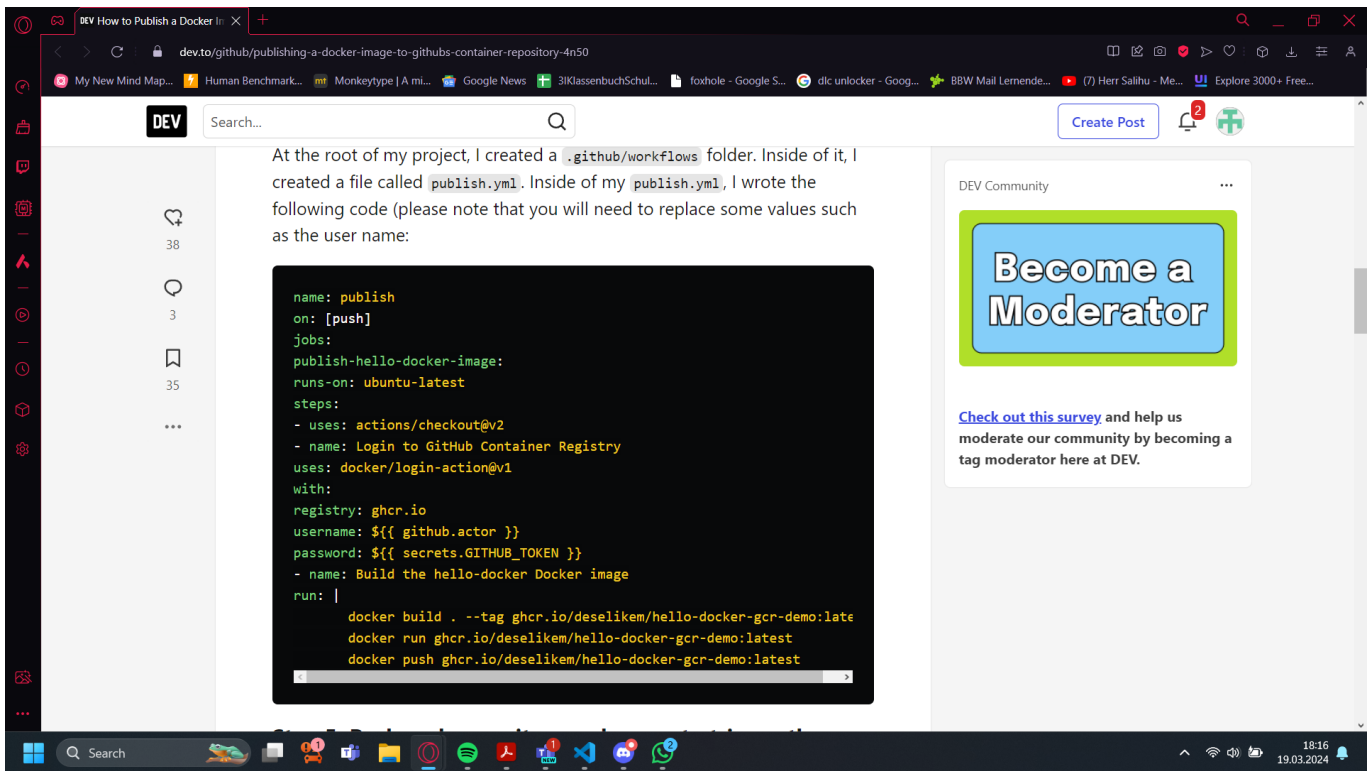
Zuvor stockten die Actions schon nur bei dem Dockerfile, nach dieser Änderung hat es jedoch alles glatt durchgeführt.



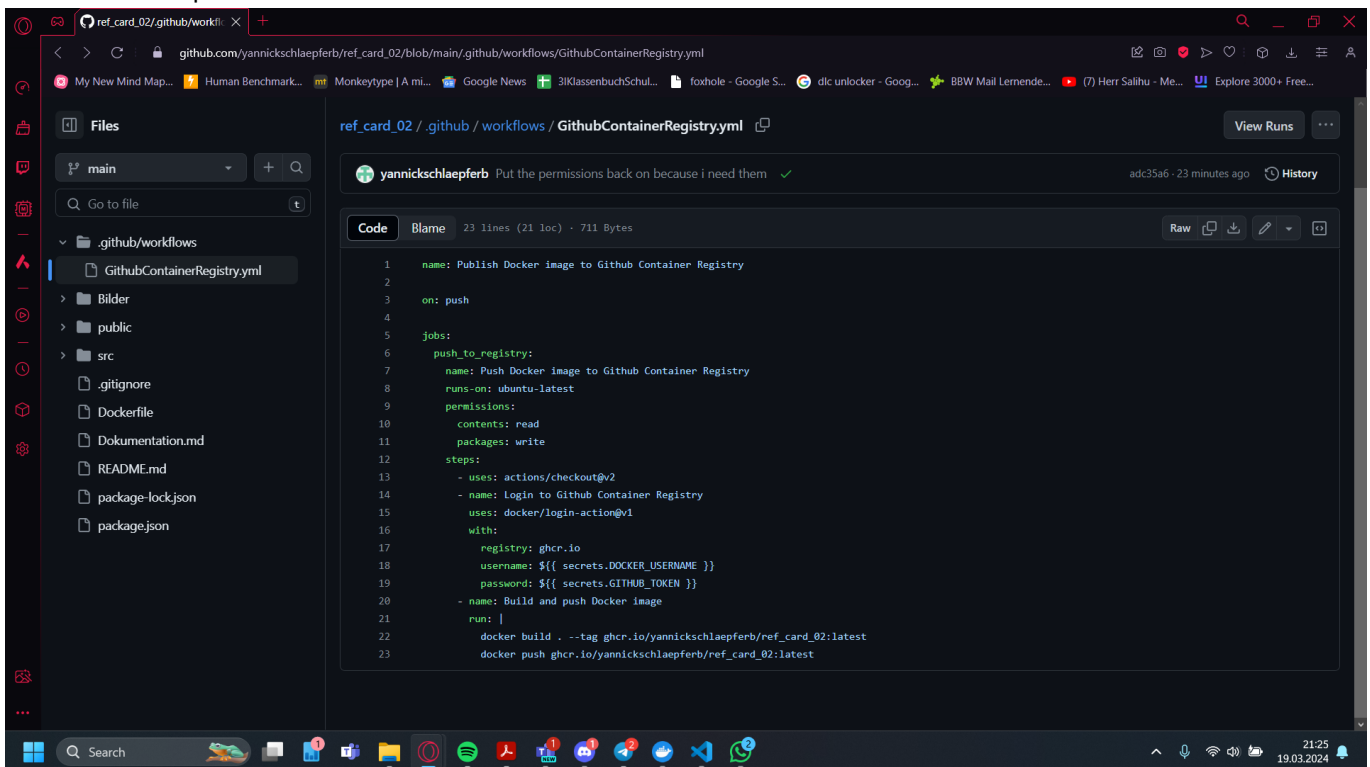
Aufgabe 2: Ich ging auf personalAccessTokens und generierte einen classic Token.



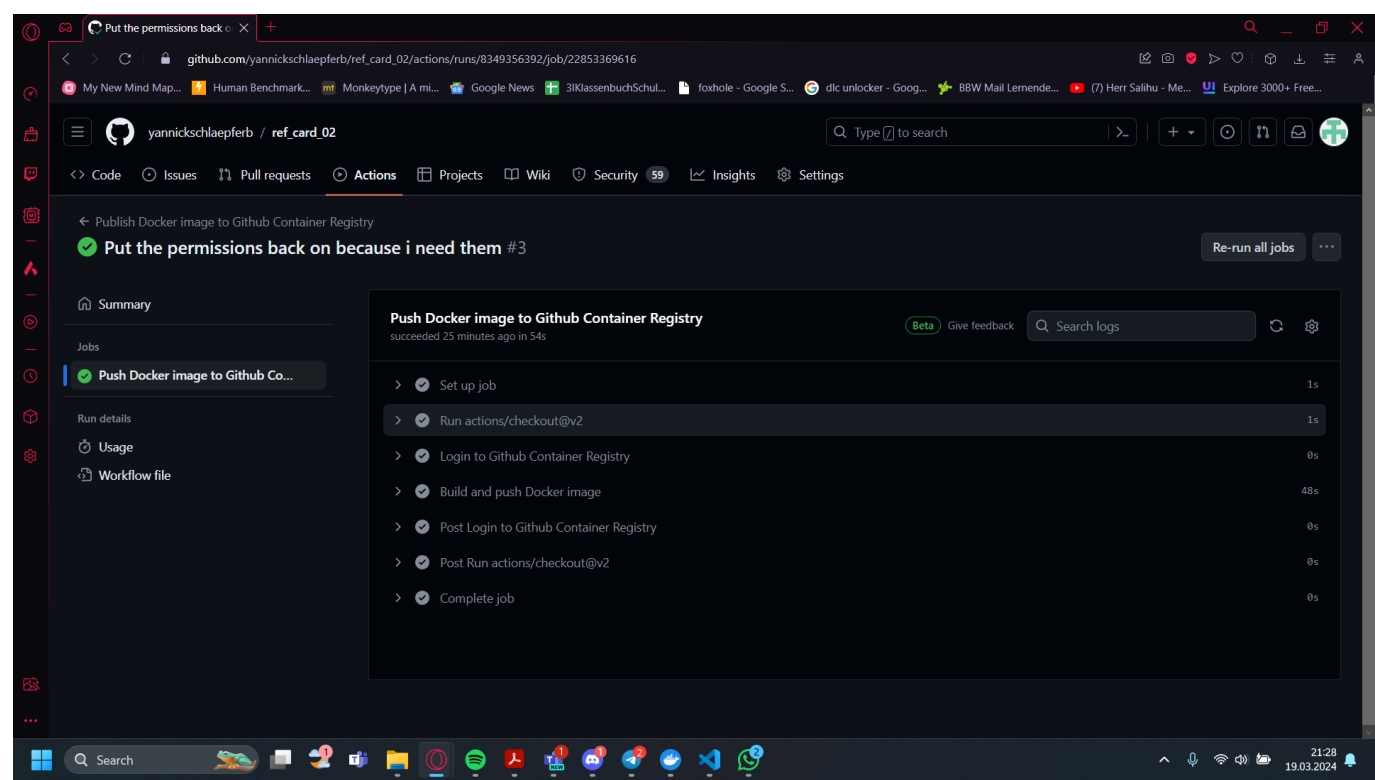
Danach übernahm ich den Code von diesem Tutorial



und fügte ihn bei meinem `.yml` file ein. Dabei änderte ich den username zu meinem github secret key Namen und änderte auch repository und username im run command. Damit ging es noch nicht und ich fügte noch read and write permissions hinzu.



Danach war der Job erfolgreich.



und das package war da

