



ROITRAINING
MAXIMIZE YOUR TRAINING INVESTMENT™

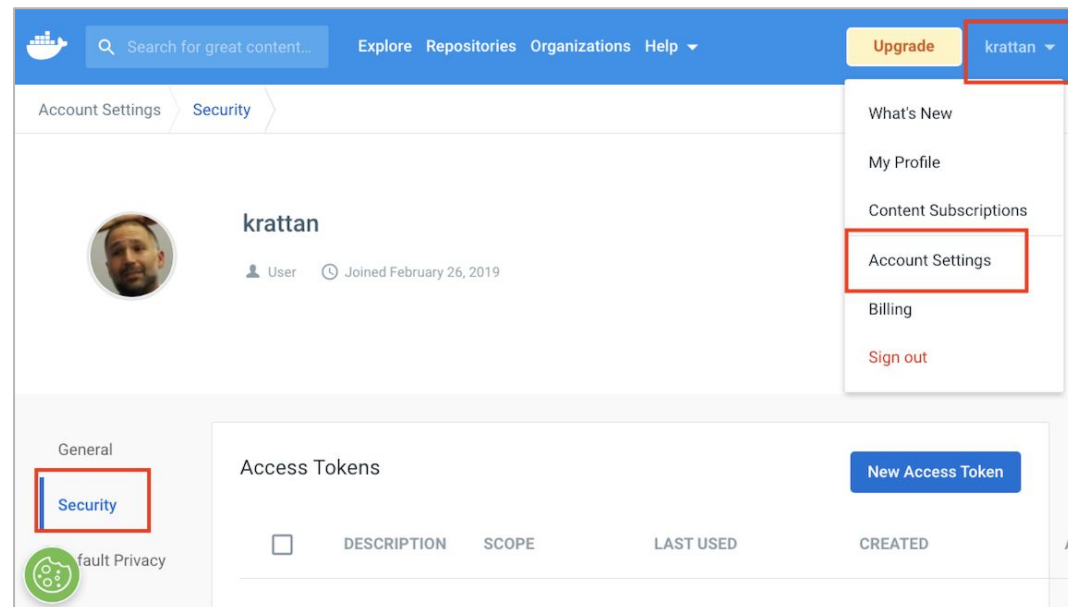
Cheat Sheet: GitHub Actions

Preparing for CI/CD

- In this lab, you will need:
 - A Docker Hub identity
 - A working Dockerfile in your repository
 - Working Unit Tests

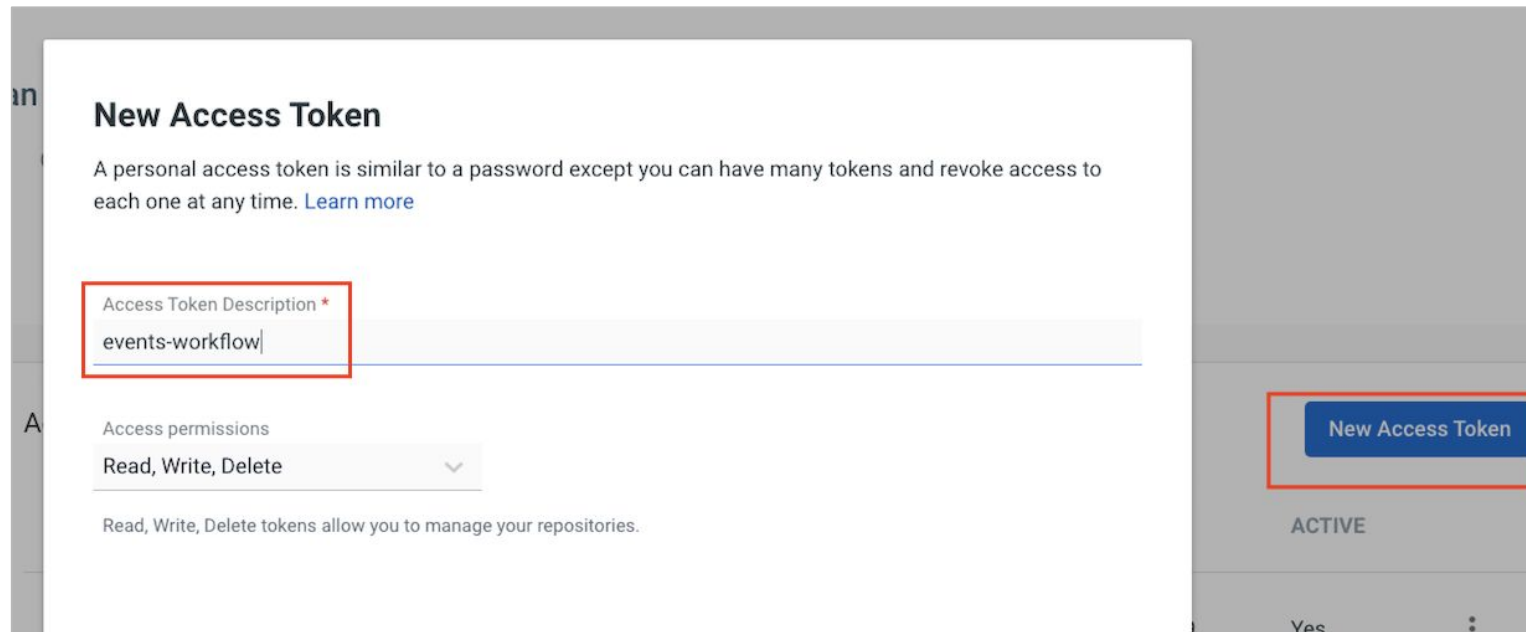
Creating Credentials

- Your pipeline will need a Docker id and an access token in order to work
- The owner of the external GitHub repository will begin by creating the necessary Docker credentials:
 - Go to Docker Hub (<https://hub.docker.com>), sign in, and click your id
 - Select **Account Settings** and then click **Security**



Creating a Docker Token

- Click **New Access Token** and give the token the name **events-workflow**
 - Leave the permissions as Read, Write, Delete
 - Click **Generate**



New Access Token

A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. [Learn more](#)

Access Token Description *

events-workflow

Access permissions

Read, Write, Delete

Read, Write, Delete tokens allow you to manage your repositories.

New Access Token

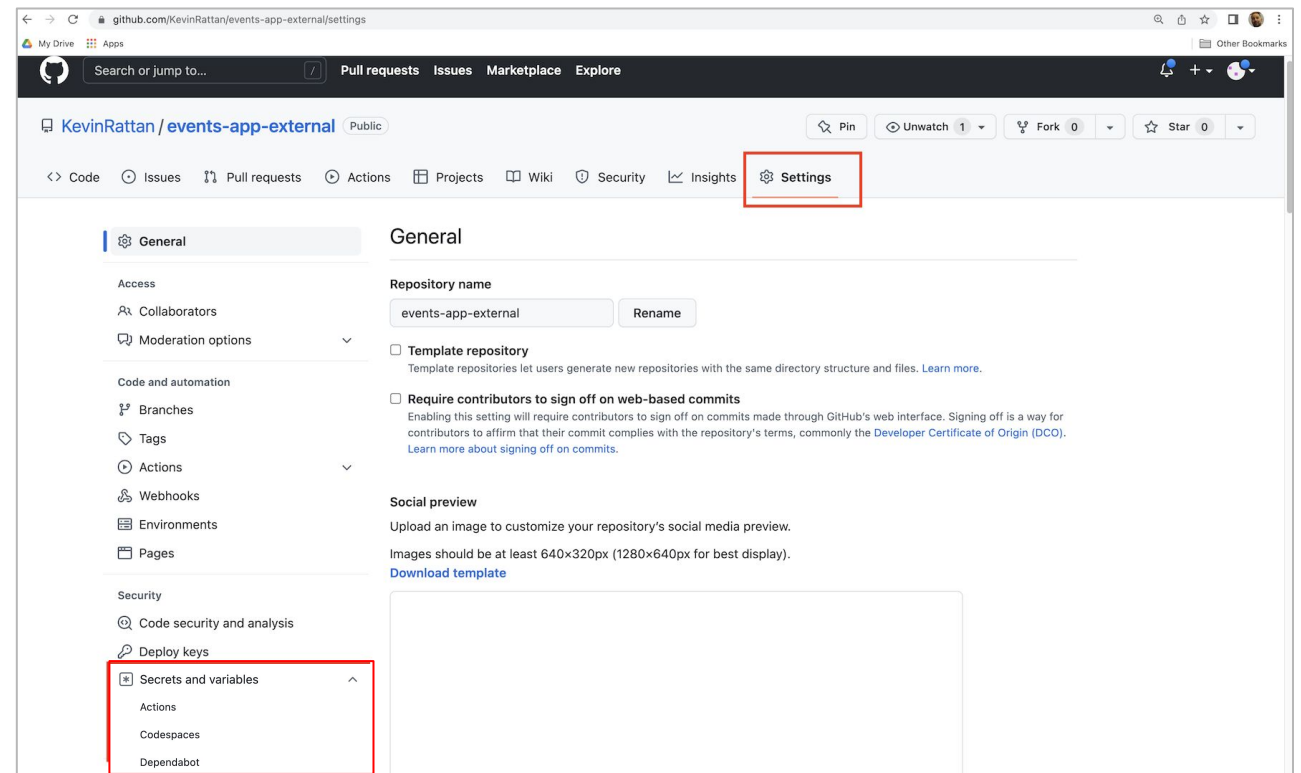
ACTIVE

Yes

- The token will be displayed
 - **COPY** the token and save it on your computer

Creating GitHub Secrets

- Go to GitHub (<https://github.com>)
- Navigate to your **events-app-external** repository in your browser
- Click the **Settings** link and then expand **Secrets and variables**
- Select **Actions** from the **Secrets and Variables** options



Creating Secrets

- Click **New Repository Secret**, and provide the following values:
 - **Name** : **DOCKER_HUB_USERNAME**
 - **Secret** : *your own docker username*
- Click **Add secret**
- Click **New Repository Secret** again, and create another secret:
 - **Name** : **DOCKER_HUB_ACCESS_TOKEN**
 - **Secret** : *the Docker Hub token you saved earlier*
- Click **Add secret**

Actions secrets / New secret

Name *

DOCKER_HUB_USERNAME

Secret *

your-username-here

Add secret

Editing the Pipeline

- You have now prepared all the credentials. The next step is to create the pipeline.
- A start point for your pipeline is available
`cd ~/sources`
`edit pipeline.yaml`
- Open **pipeline.yaml** in the Cloud Shell editor
- Examine the file contents and make sure that the secret names and image name match the ones you entered in your GitHub project
- Then select and copy the entire contents of the file

Creating a Pipeline

- Return to your project in GitHub and click **Actions** tab menu
- On the next page, click the **Set up a workflow yourself** link
- In the **Edit new file** text area, paste the file you just copied
- Then click **Start commit** and then **Commit new file**
- Click **Actions**
 - Your workflow should have started
- Wait for your workflow to complete successfully and confirm that a new image has been uploaded to Docker Hub
- If the workflow fails, examine the run history and fix any underlying issues

(Optional) Create Your Own Pipeline

If you have more time...

- Do the same thing using your internal repository
 - You will need:
 - To modify the starting pipeline
 - Your Docker Hub identity
 - A working Dockerfile in your repository
 - Working Unit Tests

Success!

- **Congratulations!** You have created a continuous integration pipeline using a GitHub Action.