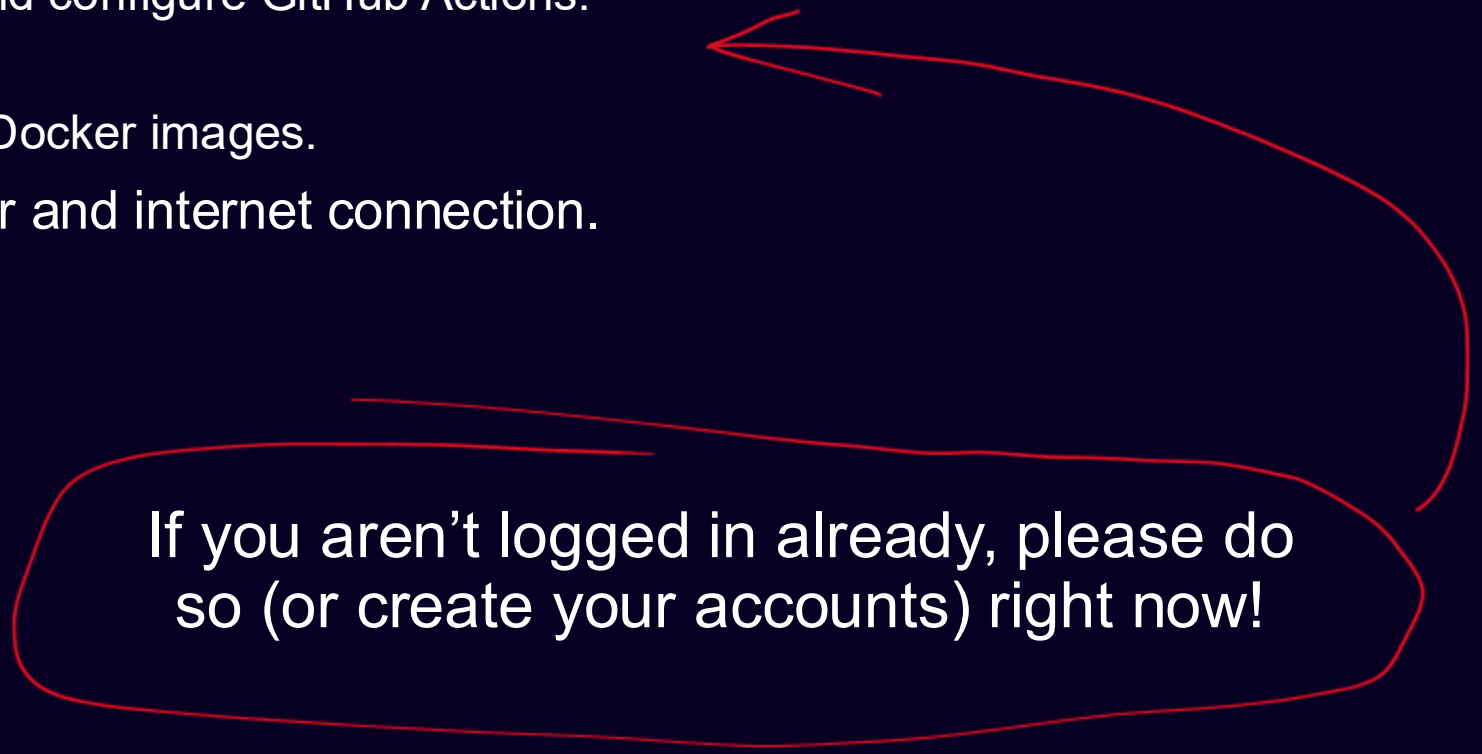
arm

Building Multi-Architectural Docker Images via GitHub Actions

Avin Zarlez (She/Her)

Requirements

- GitHub account
  - To fork the repository and configure GitHub Actions.
- Docker Hub account
  - To create and manage Docker images.
- Access to a web browser and internet connection.



Overview – GitHub Actions

- Create custom workflows for continuous integration (CI) and continuous delivery/deployment (CD)
  - Automate any task, such as building, testing, deploying
  - Check your code before merging
  - Written in easy-to-understand YAML

- Standard Runners
  - Free to all public GitHub repositories
    - Limited number of free minutes for private GitHub repositories, then a small per minute fee.
  - Arm64 runners now available for Linux, Windows and MacOS!
- Large Runners
  - More powerful compute for more intense workflows
  - Per minute fee that scales based on how powerful of compute
- Self Hosted Runners
  - Build it on YOUR computer!
  - Free* (as long as you provide the computer)

Overview – Docker

- “Works on my machine!” – “Okay, let’s ship your machine”
  - Package applications into a container
  - Consistent environment across platforms
  - Streamlined deployment

- Dockerfiles
  - Step by step instructions for how to create the container
  - Define base operating system, install dependencies, add compiled code, etc.

- Sorted by “tags”
  - Such as “latest”, “unstable” or certain version numbers
- Can include multiple images in one tag
  - Automatically use the correct image for your platform

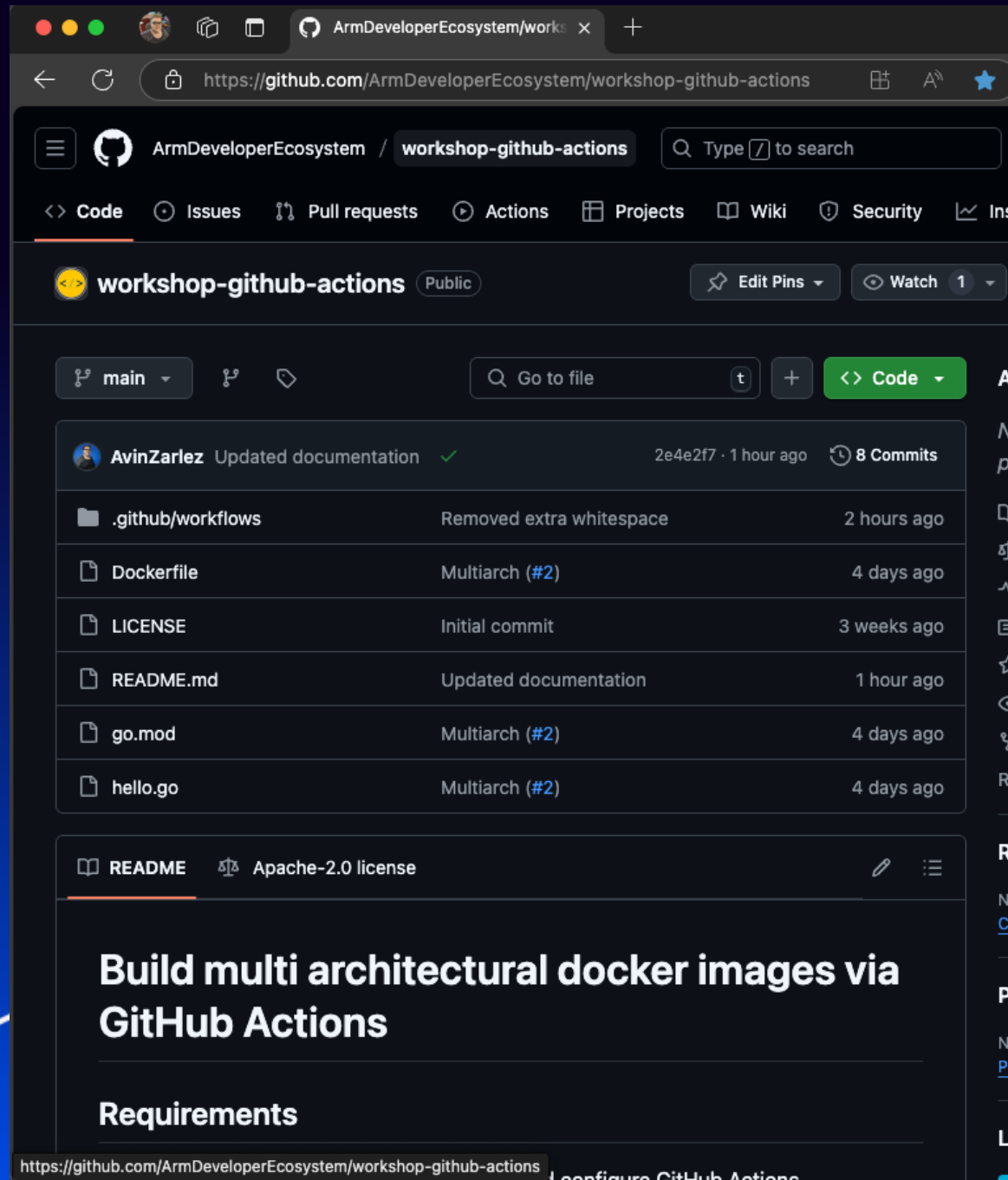
- Will authenticate with a Personal Access Token (PAT)

Let’s go to the workshop GitHub repo

https://github.com/ArmDeveloperEcosystem/workshop-github-actions

Get ready to fork it!

arm



Return to presentation when attendees have:

- ✅ Forked GitHub repo: https://github.com/ArmDeveloperEcosystem/workshop-github-actions
- ✅ Walked through Go application
- ✅ Examined Dockerfile
- ✅ Went to Docker Hub: https://hub.docker.com/
  - Created Personal Access Token
- ✅ Set up repository Secrets and Variables
  - DOCKER_PAT: <your docker personal access token>
  - DOCKER_USER: <your docker username>
  - BASE_OS: ubuntu-24.04
- ✅ Enabled GitHub Actions workflow in forked repository
- ✅ Dispatched build and test workflow
- ✅ Analyzed workflow yaml
  - Triggers
  - Variables and conditions
  - Jobs
  - Matrix strategy
  - Action types

arm

Best Practices

- Use Standard Runners whenever possible!
  - Save money and use a variety of platforms
  - Arm options for Linux, Window and MacOS!
  - https://docs.github.com/en/actions/writing-workflows/choosing-where-your-workflow-runs/choosing-the-runner-for-a-job
- Make your workflows modular
  - Break down into smaller reusable components, don't hard code variables
- Build and test code before accepting PRs
  - Use smaller linting pipelines for all branches
  - Require full tests for merge into shared branches
  - Automate tagging release builds on main branch
- Remove human steps from your process
  - Anything a person must do, automate it. It's worth the time investment

Standard GitHub-hosted runners for public repositories

For public repositories, jobs using the workflow labels shown in the table below will run on virtual machines with the associated specifications. The use of these runners on public repositories is free and unlimited.

Virtual Machine	Processor (CPU)	Memory (RAM)	Storage (SSD)	Architecture	Workflow label
Linux	4	16 GB	14 GB	x64	`ubuntu-latest`, `ubuntu-24.04`, `ubuntu-22.04`, `ubuntu-20.04`
Windows	4	16 GB	14 GB	x64	`windows-latest`, `windows-2025`, `windows-2022`, `windows-2019`
Linux [Public preview]	4	16 GB	14 GB	arm64	`ubuntu-24.04-arm`, `ubuntu-22.04-arm`
Windows [Public preview]	4	16 GB	14 GB	arm64	`windows-11-arm`
macOS	4	14 GB	14 GB	Intel	`macos-13`
macOS	3 (M1)	7 GB	14 GB	arm64	`macos-latest`, `macos-14`, `macos-15`

arm

Merci
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Thank You
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה
ధన్యవాదములు
Köszönöm