tether.

Job Title	DevOps Engineer
Department	Tether Data

Task

Your task is to design a production-quality workflow that covers local development, automated testing, and cross-platform builds, with artifacts published to GitHub Container Registry (GHCR) and GitHub releases for the MLC-LLM codebase.

Repository:

- https://github.com/mlc-ai/mlc-llm
- Build instructions:
  - https://llm.mlc.ai/docs/install/mlc_llm.html#option-2-build-from-source

Deliverables:

1. Multipurpose Docker Image
   - A single Docker image that serves as both:
     - Development environment (interactive shell, source mounted, dev tools installed)
     - Build environment (non-interactive entrypoint for compiling and packaging)
   - The image must be built and pushed automatically to GHCR via your CI pipeline.
2. Automated Tests
   - Tests must run (and gate further stages) in your CI workflow.
3. GitHub Actions CI/CD Pipeline
   - A build pipeline that follows test-driven deployment practices and builds the MLC-LLM python package, and publishes the wheels as a GitHub release for the following platforms:
     - Linux (x64)
     - Windows (x64)
4. Documentation
   - Write extensive documentation with information on prerequisites, dependencies and how to build and run the package & environment.
   - Also document how the GitHub Actions workflow is structured, including triggers, jobs, and publishing steps.

Submission Steps

Once completed, please upload the repo on GitHub and share access with the following users:

- subashsn
- chetasr
- Ignaciolarranaga
- tamer-hassan-tether
- BautistaTether

Please share an email regarding the same with a link to the repo(s) you have submitted.

Best Practices

Pipelines:

- Follow DRY principles (Don't Repeat Yourself) wherever possible to keep code structure modular and reusable.

Documentation:

- Use clear and concise language; do not add filler text to expand on the content.
- Use code snippets and examples wherever possible.
- Make sure to have uniform formatting across the documentation.
- Use diagrams / visuals wherever possible to illustrate flows and concepts.

Evaluation Criteria

- Pipeline readability and structure.
- Best practices for Docker containers.
- Documentation quality (clear to read and understand).