**The clean mapping**

1. **CI (Continuous Integration)**
   * **What really happens:** When you push code to GitHub, a **runner VM** (GitHub Actions) pulls your repo, runs tests/linters, and **builds a Docker image from your Dockerfile**.
   * **Key point:** It’s not “transferring your *local* code.” It’s **building from the code you pushed** to GitHub on a clean runner, so builds are reproducible.
2. **CD = Continuous Delivery (artifact ready)**
   * After CI passes, the pipeline **logs in to AWS ECR** and **pushes the built image** there.
   * Now your image is **stored/available** in ECR (the registry).
   * With *Delivery*, you typically **stop here** (or deploy to staging) and wait for **manual approval** to go to prod.
3. **CD = Continuous Deployment (auto ship to prod)**
   * If you enable automatic deploy: the pipeline **updates your ECS service** (or task definition) to use the **new ECR image tag**.
   * **ECS then pulls the image from ECR** and performs a rolling update (with health checks, rollback, etc.).
   * Important nuance: you don’t “send” the image to ECS; **ECS pulls** it from **ECR** when told to deploy.

**Tiny gotchas to remember**

* CI builds from the **repo on GitHub**, not from your **local laptop** (unless you intentionally build locally and push the image yourself).
* **Delivery vs Deployment:** Delivery = artifact is ready (in ECR) and may need **manual approval**; Deployment = **automatic** release to ECS once checks pass.
* ECS needs the **new image tag** (usually the commit SHA) in the **task definition** to know which image to pull.