**1. Build**

* **What it is:**  
  The process of taking source code and turning it into executable artifacts (binaries, Docker images, packages, etc.).
* **Activities:**
  + Compile code
  + Run unit tests
  + Package dependencies
  + Produce build artifacts (for example, .jar, .dll, container image)
* **Goal:**  
  Ensure the codebase is valid, consistent, and ready to be shipped.
* **Example:**  
  In a CI pipeline, mvn package (Java) or dotnet publish (C#) runs, producing a deployable artifact.

**2. Release**

* **What it is:**  
  The process of taking a specific build artifact and preparing it for delivery into one or more environments (staging, QA, production).
* **Activities:**
  + Versioning the build (for example, v1.2.0)
  + Storing artifacts in a release repository (Nexus, Artifactory, GitHub Packages)
  + Defining release notes, approvals, and change management
  + Coordinating which build goes to which environment
* **Goal:**  
  Provide a controlled, traceable package that can be deployed consistently.
* **Example:**  
  A release pipeline in Azure DevOps takes the build artifact and promotes it through Dev → QA → Prod with approvals.

**3. Deployment**

* **What it is:**  
  The act of actually running the release artifact in a target environment (cloud, VM, Kubernetes, App Service, etc.).
* **Activities:**
  + Copying binaries or containers to servers
  + Running database migrations
  + Configuring environment variables and secrets
  + Starting services and verifying health checks
* **Goal:**  
  Make the software available to users in a stable, working state.
* **Example:**  
  Deploying a Docker image to an Azure Kubernetes Service (AKS) cluster or pushing a web app to Azure App Service.

**Comparison Table**

| **Stage** | **Focus** | **Output** | **Example Tooling** |
| --- | --- | --- | --- |
| Build | Compile and package code | Build artifact (binary, image) | Maven, Gradle, MSBuild, Docker build |
| Release | Version and prepare for delivery | Tagged artifact and release notes | Azure DevOps Release, GitHub Releases, Octopus Deploy |
| Deploy | Run in target environment | Live running application | Helm, Ansible, Terraform, Kubernetes, Azure App Service |

**Why DevOps separates them**

* **Build once, deploy many times:** You don’t want to rebuild code for each environment. Instead, you build once, release that artifact, and deploy it consistently across Dev, QA, and Prod.
* **Traceability:** You can always answer: *Which build is running in production?*
* **Automation:** CI/CD pipelines automate all three, but keep them conceptually distinct.

In short:

* Build = make it.
* Release = package and version it.
* Deploy = run it.