

## 2. 持续交付 (CD)

持续交付 (CD) 的主要逻辑是：将“在 CI 流程中构建的程序镜像”从镜像仓库自动发布到具体的基础架构环境（如测试/生产 Kubernetes 集群）。

实现 CD 的工具主要有 GitLab Runner (CD Pipeline)、Helm (Kubernetes 软件包管理工具) 等。

CD 的核心是：通过输入的各种用户参数（如 yamI 文件、环境配置参数等）自动生成具体的发布指令（如 Helm 指令），并根据参数中设置的信息来配置程序的具体运行环境。

可持续交付 (CD) 的基本运行流程如图 9-2 所示。



图 9-2

本节描述了 CI/CD 的基本概念及流程，后面节将依据这些内容来构建具体的 DevOps 发布系统。

## 9.2 了解 DevOps 发布系统的设计流程

本章所设计的 DevOps 发布系统，主要是利用 GitLab 提供的 CI 机制，来实现在代码发生提交或合并等事件时自动触发预设的 CI/CD 流程。具体的系统结构如图 9-3 所示。

- CI 阶段主要包括基本的代码编译、构建和打包，并将打包好的应用 Docker 镜像发布至镜像仓库中。
- CD 阶段则是从镜像仓库拉取应用 Docker 镜像，并根据设置的 CD 流程将应用发布至指定的 Kubernetes 集群中。
- DevOps 发布系统主要由 GitLab、Harbor 镜像仓库及 Kubernetes 集群组成。其中，GitLab 主要承担代码版本管理，以及 CI/CD 流程的定义和触发；Harbor 负责应用 Docker