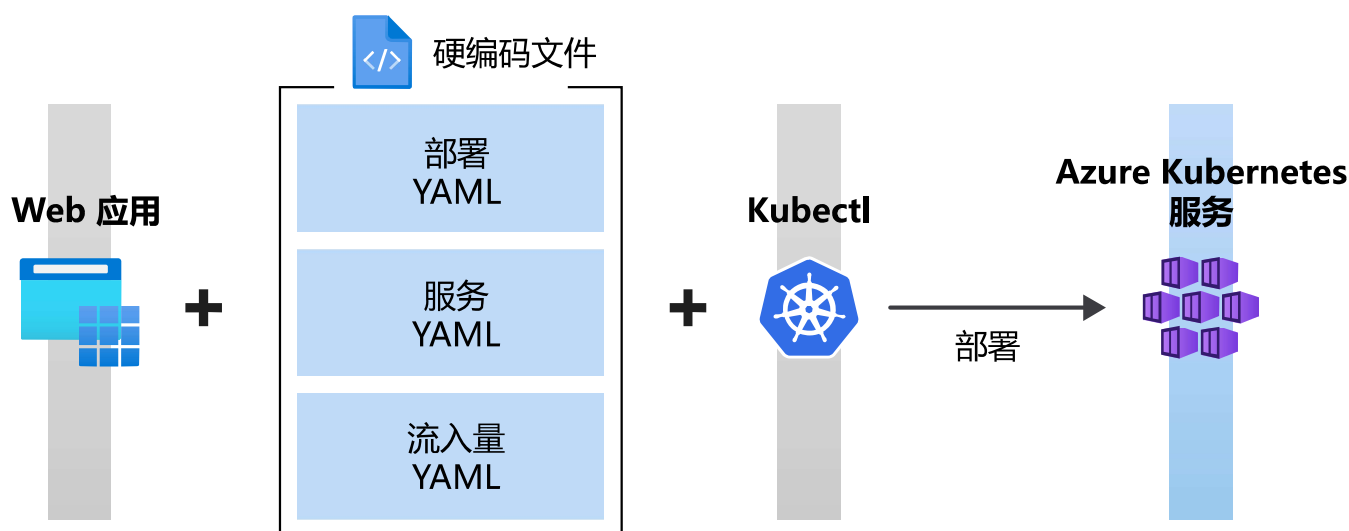


什么是 Helm?

5 分钟

任何应用程序的部署、版本控制和更新通常都需要规划和管理，以确保部署正确版本的软件库和配置设置，使部署的应用程序能够正常运行。

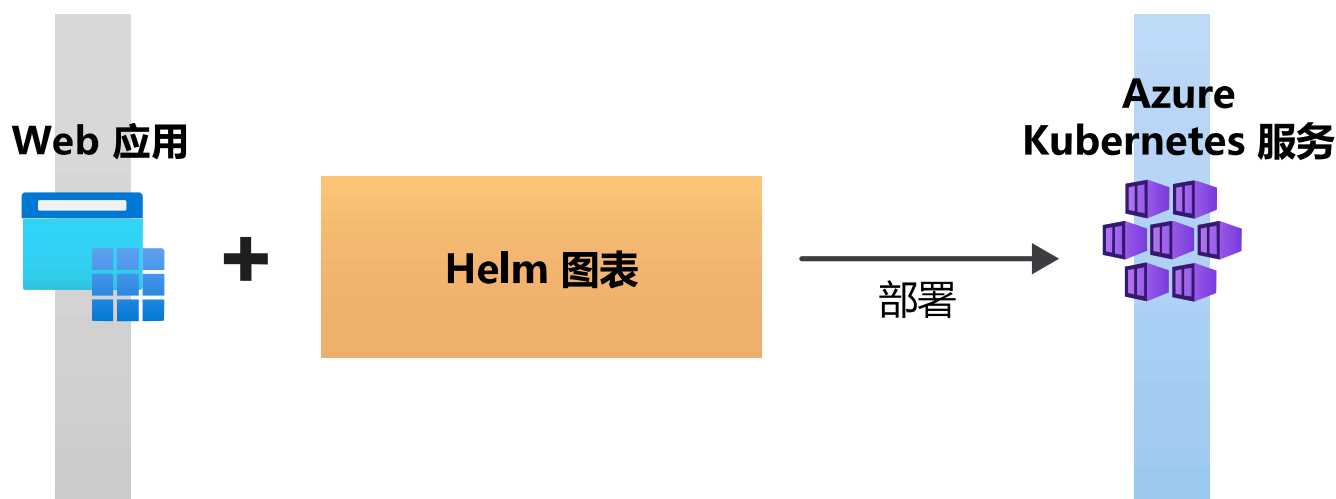


假设你的开发团队决定将无人机跟踪公司网站部署到 Kubernetes。在此过程中，你的团队将创建部署、服务和基于 YAML 的入口文件。每个文件中的信息针对每个目标环境进行硬编码，通过手动方式进行维护。为每个环境维护三个文件很繁琐，且随着应用程序的增长，复杂性会增加。

你可以使用 Helm 来简化应用程序部署过程，并避免对部署变量和设置进行硬编码。

什么是 Helm?

Helm 是 Kubernetes 的包管理器，它将应用程序的所有资源和部署信息组合到单个部署包中。



可以采用与在 Windows 上使用 Windows 程序包管理器、在 Linux 上使用高级包工具 (apt) 或在 macOS 上使用 Homebrew 安装应用程序时相同的方式来看待 Helm。你指定要安装、更新或删除的应用程序的名称，Helm 则负责处理该过程。

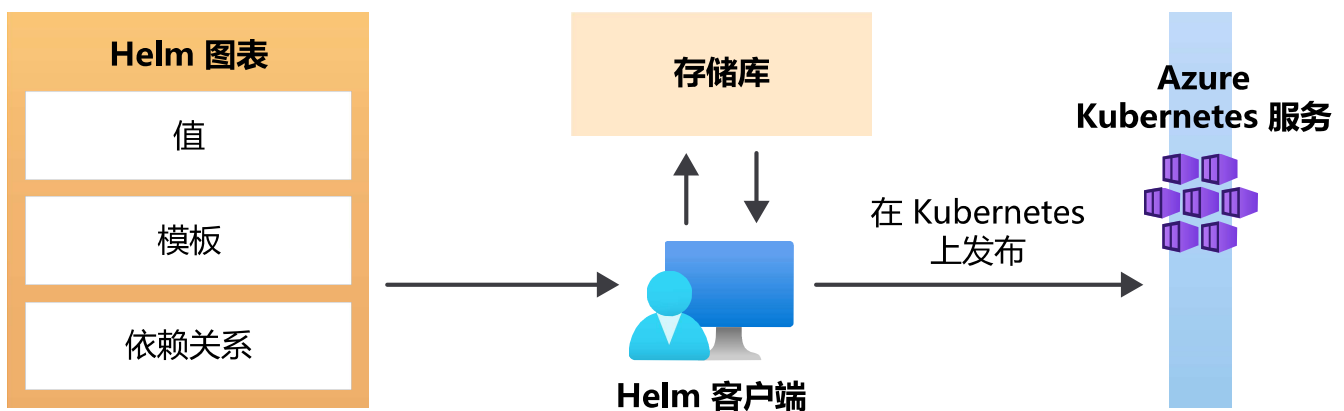
但是，不会将你限制为一次仅安装一个应用。Helm 允许你创建模板化、人工可读的 YAML 脚本文件来管理应用程序的部署。可以使用这些模板文件指定所有必需的依赖项、配置映射和用于成功管理应用程序部署的机密。

Helm 使用四个组件来管理 Kubernetes 群集上的应用程序部署。

- Helm 客户端
- Helm 图表
- Helm 发布
- Helm 存储库

什么是 Helm 客户端？

Helm 客户端是客户端安装的二进制文件，负责创建和提交部署 Kubernetes 应用程序所需的清单文件。客户端负责用户与 Kubernetes 群集之间的交互。



Helm 客户端适用于所有主要操作系统，安装在客户端电脑上。在 Azure 中，Helm 客户端预安装在 Cloud Shell 中，支持 Kubernetes 的所有安全性、标识和授权功能。

此模块和其他 Learn 模块始终假定使用 Helm v3。可以运行 `helm version` 命令来检查客户端设备上运行的 Helm 版本。如果运行的是 Helm v2，请记住，此模块中的某些命令参数将有所不同。

什么是 Helm 图表？

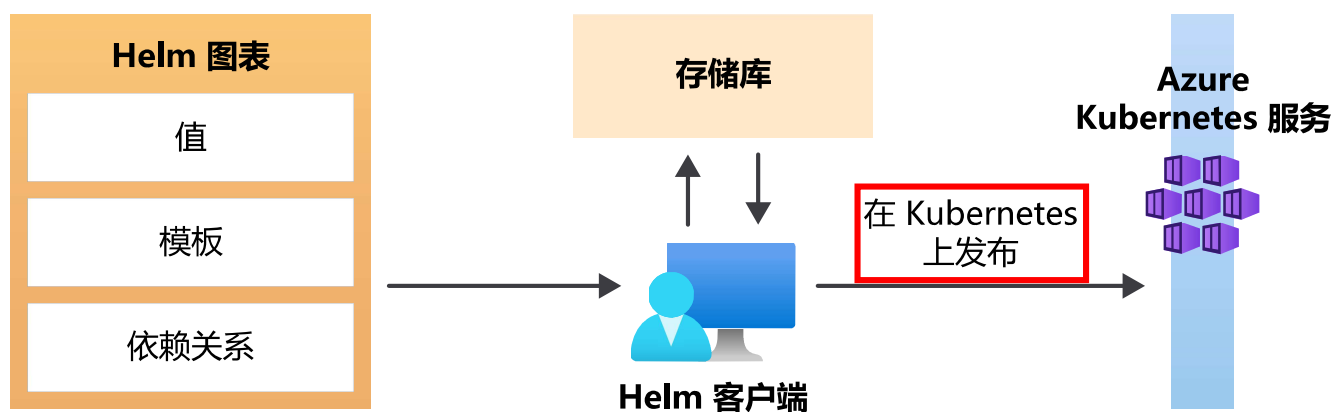
Helm 图表是描述相关的一组 Kubernetes 资源的模板化部署包。它包含为了让应用程序在 Kubernetes 群集上运行而为其生成和部署清单文件所需的所有信息。

Helm 图表包含用来描述图表的多个文件和文件夹。某些组件是必需的，某些是可选的。你选择包括的内容取决于应用配置要求。下面是包含以粗体显示的必需项的文件和文件夹的列表。

文件/文件夹	描述
<code>Chart.yaml</code>	一个 YAML 文件，其中包含有关图表的信息。
<code>values.yaml</code>	图表的默认配置值。
<code>templates/</code>	包含图表的部署模板的文件夹。
<code>LICENSE</code>	包含图表许可证的纯文本文件。
<code>README.md</code>	一个 Markdown 文件，其中包含有关如何使用图表的说明。
<code>values.schema.json</code> **	一个架构文件，用于在 <code>values.yaml</code> 文件上应用结构。
<code>charts/</code>	一个文件夹，其中包含主图表的所有子图表。
<code>crds/</code>	自定义资源定义。
<code>templates/Notes.txt</code>	包含模板用法说明的文本文件

什么是 Helm 发布？

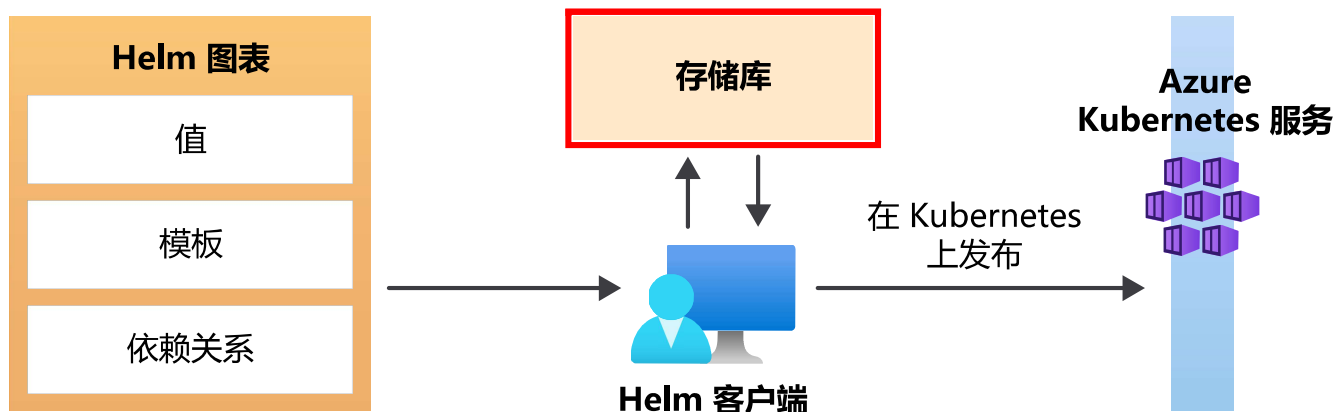
Helm 发布是使用图表部署的应用程序或应用程序组。每次安装图表时，都会在群集上创建应用程序的新实例。每个实例都有一个发布名称，可用于与特定的应用程序实例进行交互。



例如，假设你已使用图表在 Kubernetes 群集上安装了两个 Nginx 实例。然后，你决定升级第一个 Nginx 实例，但不升级第二个。由于两个发布不同，因此你可以升级第一个发布而不会影响第二个发布。

什么是 Helm 存储库？

Helm 存储库是一个专用的 HTTP 服务器，它存储有关 Helm 图表的信息。该服务器提供了一个文件来描述图表以及在何处下载每个图表。



Helm 项目承载了许多公共图表，并且存在你可以从中重复使用图表的许多存储库。Helm 存储库简化了 Helm 包的可发现性和可重用性。

使用 Helm 的好处

Helm 带来了许多好处，可简化应用程序部署并提高云原生应用程序的开发和部署生命周期中的工作效率。使用 Helm，你可以获得具有以下特点的应用程序发布：

- 可重复
- 可靠
- 可在多个复杂环境中管理
- 跨不同开发团队重复使用。

Helm 图表使用由所设置的输入值参数化的打包模板逻辑来标准化应用程序的部署。此模板驱动的包设计提供了一个不受环境影响的方法，用于部署和共享云原生应用程序。