DevOps

白皮书

发布日期 2019-06-26

北京凌云雀科技有限公司

目 录

1 概述

1.1 产品背景

一直以来，在传统的企业软件开发模式中，开发、测试和运维人员分别隶属于不同的部门。各部门虽然共享组织目标，但是彼此之间缺乏高效的协作方式，导致在日常 IT 运营和维护的过程中矛盾不断。

在传统的交付周期中，软件开发人员在编写代码后，将软件交给测试团队进行测试，再将最终版本交给运维团队部署。运维团队需要解决代码部署过程中出现的问题，或将代码交还给开发团队来解决遇到的问题，复杂的流程导致软件从开发到交付效率低下且周期延长。由于对运行环境和应用的内部细节缺乏了解，运维人员发布应用时，难以正确选择运行环境和控制发布流程，往往会遇到各种各样的问题，同时花费很多时间。从研发到发布应用是高压力、高风险的活动。原有的开发运维流程，已不适应新时代的科技发展。

在这样的背景下，灵雀云基于 DevOps 理念和为大型客户搭建 DevOps 平台的经验，推出了新一代的 DevOps 平台 Alauda DevOps（以下简称平台或 DevOps）。

DevOps 旨在助力推动研发、测试、运维各小组之间业务的连续性、安全性、敏捷性，更便于适应快节奏的企业发展步伐，轻松应对瞬息万变的市场需求，提高团队协作，扩大创新力度。在软件工程的所有步骤，从整合、测试、发布到部署和基础设施管理，DevOps 大力提倡自动化和监控，缩短开发周期，提高部署频率，与业务目标保持紧密一致。

1.2 产品简介

DevOps 是一款基于容器的云应用平台，支持容器负载统一调度，细粒度的多租户权限。平台为企业提供包含流水线管理，代码仓库纳管，制品管理等服务在内的开箱即用一站式服务，使软件的构建、测试和发布变得更加快捷、频繁和可靠。

平台分为管理视图和业务视图，如图 1-1 所示。

图 1-1 产品架构

通过完整的 DevOps 工具链，深度集成代码仓库、制品仓库、持续集成等类别中的主流工具，实现零成本迁移，快速实践 DevOps。DevOps 强调产品管理、自动化软件交付和基础设施变更的过程。缩短开发周期，增加部署频率，实现更可靠的发布。支撑应用的全生命周期，旨在建立一套快速、频繁、稳定地进行构建、测试、发布软件的文化与环境，与业务目标紧密结合。

通过集成 Jenkins 作为实现 CI/CD 的标准工具，助力实现更敏捷、可靠的应用发布，加快产品迭代速度，使企业更专注于核心业务目标。DevOps 提供方便快捷易落地的解决方案，帮助企业快速构建基于 DevOps 的开发体验，帮助研发团队缩短交付周期、提高研发质量。

1.3 产品特点

基于 DevOps 理念，即重视软件开发人员（Dev）和 IT 运维技术人员（Ops）之间沟通合作的文化和惯例，DevOps 平台对应用研发流程中的多租户、代码、持续集成和交付等方面分别进行了优化。

- 开箱即用

支持一键部署 DevOps 平台，部署平台的同时，用户可选择自动部署哪些 DevOps 工具。部署完成后，用户即可在平台上使用 DevOps 工具。同时，平台的权限和工具的权限保持同步。

- **简单、高效的持续集成和交付**

平台基于常见的用户使用场景提供官方流水线模板，同时，支持将用户根据实际的业务需求自定义的流水线模板导入平台使用。即便不熟悉 CI/CD 工具的用户，仍可直接使用平台提供的多种流水线模板，来创建符合自身业务需求的流水线。

平台支持图形化创建流水线，通过可视化界面引导用户创建流水线，减少误操作。并且支持多分支流水线，免去重复创建流水线、满足多 PR 分支合并到 Master 分支的需求、满足临时提交代码执行流水线的需求。

平台集成了多种制品仓库，例如：Harbor 等，能够保存应用流程中的中间产物与最终产出物，保障了产出物的阶段性备份和最终交付。

- **容器应用的全生命周期管理**

平台支持 Kubernetes 容器服务，提供高性能可伸缩的容器应用管理能力，支持企业级 Kubernetes 容器化应用的全生命周期管理。例如：应用的部署、组件的管理和扩缩容、日志查看、更新容器等周期管理。

- **安全至上**

✧ **细粒度的多租户管理**

支持对接 LDAP（Lightweight Directory Access Protocol，轻量级目录访问协议）或 OAuth 2.0 协议，纳入企业已有的用户体系，可通过 RBAC 细粒度设置用户权限。

平台的企业级多租户管理能力，大大简化了为角色相同成员分配相同资源权限的重复工作，利于项目内外部的资源隔离，也增强了数据的保密性。

✧ **代码、镜像扫描**

平台支持对代码进行扫描，展示代码扫描结果，帮助开发人员管控代码质量。

✧ **平台审计**

在通过细粒度的多租户实现资源、权限隔离的同时，还支持对平台人员操作的审计，进一步保障了平台的数据安全。

- **基于用户画像的优化设计**

  ✧ **管理视图和业务视图**

  DevOps 针对企业中应用的基础环境维护人员、管理者、研发人员的日常工作平台和习惯的不同，为使用者提供了独有的操作视角。每个视角针对该角色的使用场景进行了优化，同时通过权限控制视角切换，为用户提供更简单、高效的交互体验。

  ✧ **数据可视化、信息可追踪**

  提供 DevOps 各环节数据可视化，通过数据统计，进行目标分析，实现信息的可视化。

- **标准化交付**

  灵雀云借助于多年来服务不同领域头部客户的经验，打造标准化的产品旨在服务于更加广泛的领域和更多的企业客户，助力企业获得持续创新的核心能力。

  同时，标准化交付的 DevOps 可以结合灵雀云的其他产品（例如：Alauda Container Platform、Alauda Service Mesh）提供更为丰富的 PaaS 生态体验，或接入第三方 Kubernetes 容器平台帮助企业快速实现 DevOps 落地。

1.4 系列产品介绍

灵雀云为了满足企业客户的不同需求，提供了除 DevOps 之外的系列产品。包含：Alauda Container Platform（ACP）、Alauda Service Mesh。

用户可单独订阅 ACP 作为容器 PaaS 平台为企业提供企业场景的多集群、多租户管理、智能运维，支持用户基于本平台实践 PaaS 生态集成，也可以同时订阅系列产品和 ACP 组合使用，为大型企业的复杂业务应用管理提供完整的 PaaS 平台体验。

- **Alauda Container Platform（ACP）**

  ACP 拥抱云原生（Cloud Native）技术，通过整合 Docker 容器、Kubernetes 原生架构、微服务等相关新技术和新理念，可实现业务应用从开发、测试，到部署、运维的全

生命周期平台化管理，能有效帮助企业实现数字化转型，提升企业的 IT 交付能力和竞争力。

ACP 作为企业级的云原生容器平台，可服务于不同规模的企业，支持管理各种复杂度的基础设施环境（单台机器或多个异构的数据中心）、组织结构清晰的部门建制和人员团队。

ACP 能够满足企业级应用逐步向容器化、微服务化过渡的广泛需求，支持企业建立一个覆盖内外部各环节和组织结构的私有云平台。提升企业 IT 资源利用率，加快应用迭代速度，降低应用交付成本，实现业务应用的智能运维，从而助力企业获得持续创新的核心能力。

- **Alauda Service Mesh**

Alauda Service Mesh 是基于容器和 Istio 框架的微服务治理平台。平台提供了环境部署、Istio 配置、Istio 监控、应用管理、服务注册发现、服务拓扑、调用链追踪、路由管理、Istio 网关、流量策略、安全策略等服务的开箱即用一站式服务。

Service Mesh 是分布式服务架构，基于 Istio，支持分布式服务发布、服务注册发现、调用链跟踪、智能路由、流量策略、安全策略、Istio 配置、Istio 监控等功能。

2 核心功能介绍

2.1 平台管理

- **用户管理**

平台的 IDP (Identity Provider)配置，支持通过同步 LDAP（Lightweight Directory Access Protocol，轻量级目录访问协议）导入企业已有用户体系；支持 OIDC（OpenId Connect）协议，可使用平台认可的第三方账号登录平台。

- **角色管理**

平台支持基于角色的权限访问控制（RBAC，Role-Based Access Control），根据不同的企业使用场景，系统设置了五大权限角色：平台管理员、平台审计人员、项目管理员、命名空间管理员、命名空间成员。

- **权限管理**

通过给不同的用户分配不同的角色，从而得到不同的权限，简化了权限管理，优化了权限隔离。

2.2 项目管理

平台的项目之间可以灵活的划分出独立且相互隔离的资源空间，每个项目都拥有独立的项目环境，能够代表企业中不同的子公司、部门或项目组。通过项目管理，能够轻松实现项目组之间的资源隔离、租户内的配额管理以及项目下的人员管理。如图 2-1、图 2-2 所示。

项目拥有独立的 DevOps 环境，例如：代码仓库、制品仓库、Jenkins 实例、流水线等。

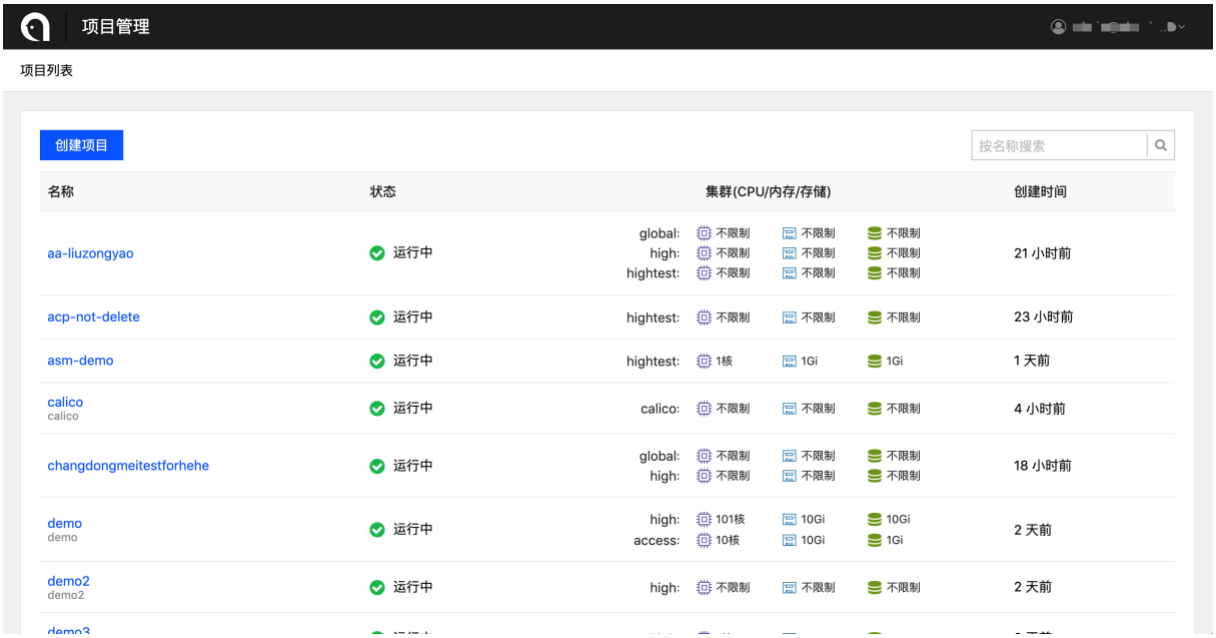

图 2-1 项目列表

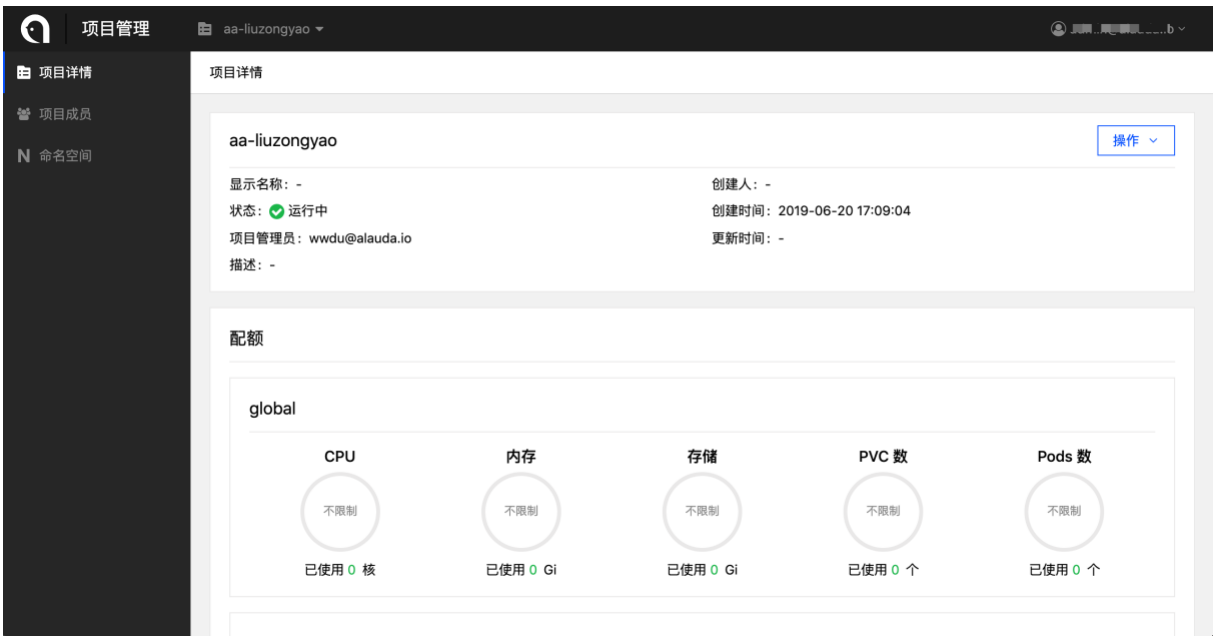

图 2-2 项目管理

2.3 视图拆分

为了进一步提升用户体验，平台根据用户画像，针对企业中应用的基础环境维护人员、管理者、研发人员的日常工作区和习惯，为使用者提供了独有的操作视角，即管理视图和业务视图。每个视图针对该角色的使用场景进行了优化，同时可通过权限控制视图的切换。

- 管理视图

管理视图从管理者的角度，对平台进行管理，例如：集成工具、为项目绑定工具、管理凭据、为项目配置流水线模板仓库等，如图 2-3 所示。

平台管理员和具有查看管理视图权限的用户，可以进入管理视图。

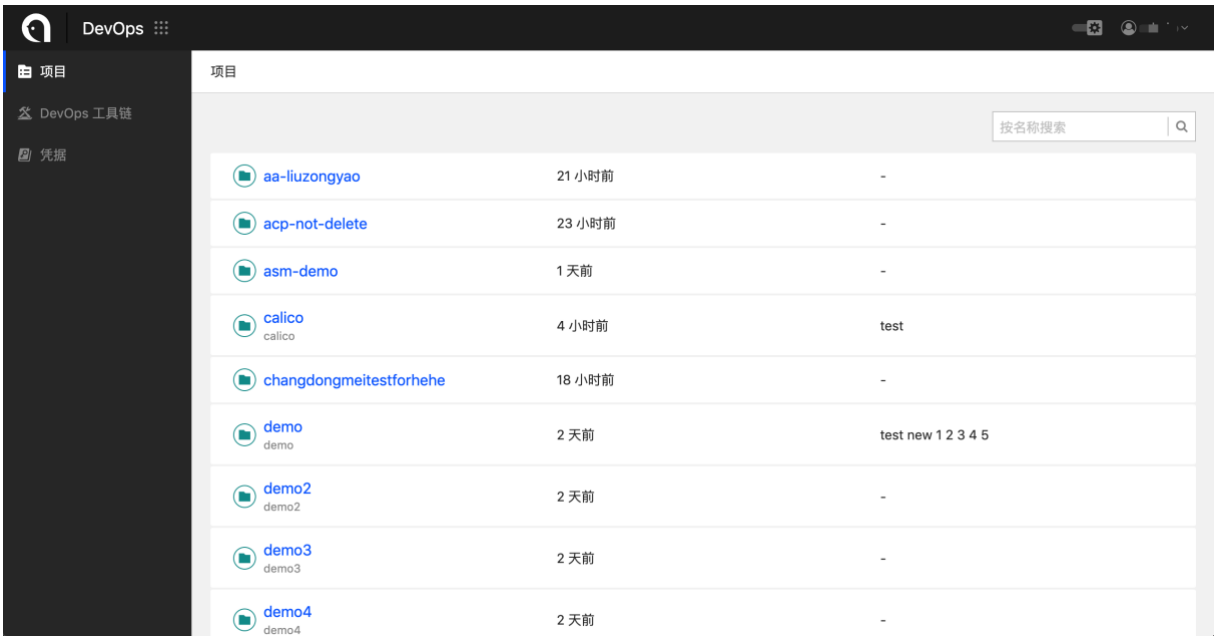

图 2-3 管理视图

- **业务视图**

业务视图主要面向项目管理员以及一般的开发、测试人员。

业务视图从项目成员的角度，运行业务，例如：创建并执行流水线、查看代码质量、扫描镜像等，如图 2-4、图 2-5 所示。

图 2-4 业务视图-项目列表

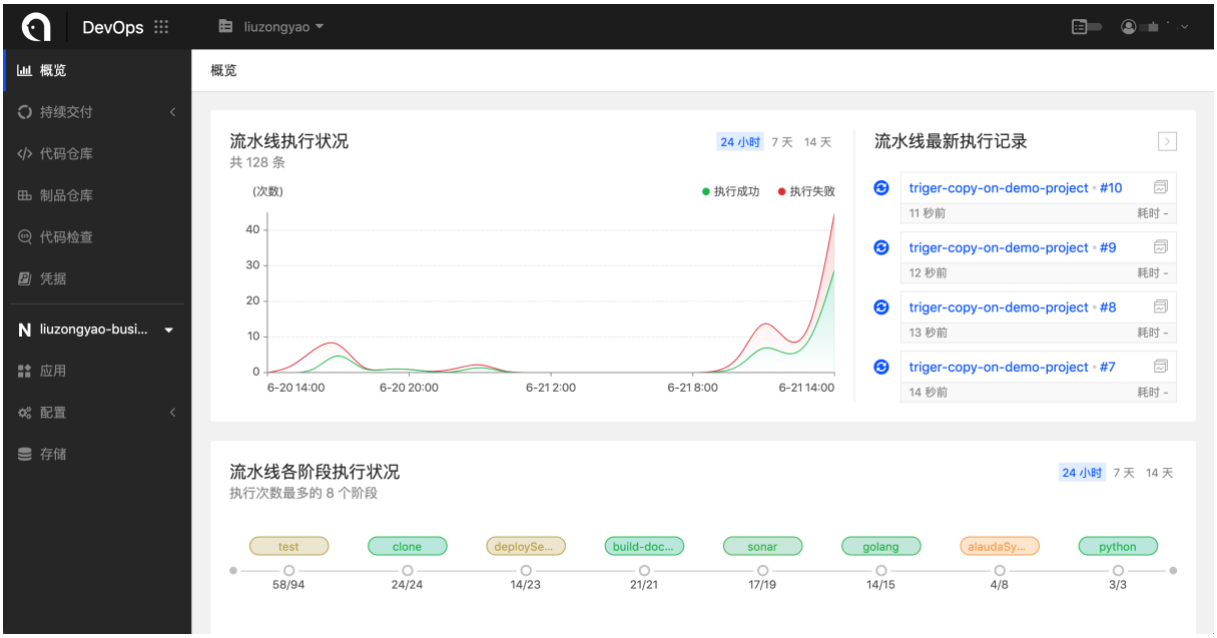

图 2-5 业务视图-概览

2.4 工具链管理

DevOps 强调产品管理、自动化软件交付和基础设施变更的过程，旨在建立一套快速、频繁、稳定地进行构建、测试、发布软件的文化与环境，与业务目标紧密结合，填补开发端和运维端之间的信息鸿沟，提倡自动化和监控，改善团队之间的协作关系。DevOps 可以缩短开发周期，增加部署频率，实现更可靠的发布。 在平台上集成 DevOps 工具可以省去反

复输入的工作量，提高操作效率，方便统一管理，平台集成了通用的工具链，在用户视图下和管理视图下都有工具链选项以供不同权限的用户使用。

- **集成 DevOps 工具链**

  ✧ **代码仓库**

  平台支持绑定多种代码仓库，例如： GitHub、Bitbucket、GitLab 公有和私有仓库、码云等。同时，支持多种代码库管理服务，例如：通过平台查看代码仓库信息等。

  将代码集成到 DevOps 平台，针对不同的项目，分配可用的代码仓库。在整个研发流程中，让用户专注于业务本身，不再关注代码仓库地址、凭据等信息，通过简单的选择即可快速、便捷的使用代码仓库。

  ✧ **持续集成**

  平台支持集成 Jenkins。集成 Jenkins 后，我们可以通过编排流水线，做多语言的持续构建，持续集成， 自动化的代码与镜像安全管理、镜像同步等等在 DevOps 上集成并绑定 Jenkins 服务，实现服务的持续集成和升级。

  ✧ **制品仓库**

  平台集成了多种制品仓库来管理镜像，实现镜像的存储、管理等，例如：Harbor Registry、Docker Registry。Docker Registry。同时，Harbor Registry 在 Docker Registry 的基础上，添加了一些必需的功能特性，例如安全、标识和管理等。

  ✧ **代码扫描**

  平台支持集成代码扫描工具，例如：质量管理平台 SonarQube，可帮助用户管理代码质量，对代码质量做自动化分析和管理。

  同时，可在平台的业务视图中查看代码的扫描和统计分析结果。

- **分配工具链**

  集成的工具链可通过绑定项目，灵活的分配给具体的项目，方便项目使用。

  一个工具可以分配给多个项目使用，一个项目也可以被分配一个或多个工具。统一由具备管理权限的角色对工具进行分配和管理，增加项目使用工具的便利性的同时，方便平台统一管理和维护工具链。

- **凭据**

在用户视图中，不同权限的平台使用者可以管理用于认证 DevOps 工具的凭据，分为租户凭据和用户凭据两种。

租户凭据：租户管理员在集成 DevOps 工具时，可使用租户凭据；在订阅工具，且希望使用自己的凭据时，也可使用租户凭据。

用户凭据：用户在创建流水线时，若使用的代码仓库、镜像仓库等工具为未集成的工具实例，则需要手动输入仓库地址，若此工具需要凭据，则需要用到用户凭据。

2.5 持续交付

- **流水线模板**

流水线模板包括官方模板和自定义模板。管理员在管理视图上，可设置要对接的流水线模板仓库，平台用户在用户视图中，通过流水线模板可快速方便地创建 Jenkins 流水线，达到降低 DevOps 学习成本，提高 Jenkins 使用效率的效果。

平台支持系统提供的流水线模板，例如：Java、Golang、Python 构建和部署、同步镜像、构建并更新应用等。不熟悉 CI/CD 工具，仍可直接使用平台提供的多种流水线模板，来创建符合自身业务需求的流水线。满足更多不同企业的工作场景和特有的业务需求，大幅度提高工作效率。

当系统提供的流水线模板不满足需求时，支持导入并同步自定义模板。

- **流水线**

平台的流水线是基于 Jenkins 进行构建和集成的。流水线是一个自定义的 CI/CD 流水线模式，定义了包含构建、测试和发布的完整构建过程。自动化持续交付流水线涉及到代码管理与集成、部署、发布等环节。当有新的代码提交时，会自动触发流水线。

流水线模板：支持使用系统自带或自定义的模板创建流水线。默认会显示所有的流水线模板，包括系统自带或自定义的模板，并根据模板的标签分组展示，支持分组查找模板。每个模板显示了模板名称、版本号、标签分类、模板包含的流水线任务名称。

多分支流水线：多分支流水线可以自动扫描并过滤指定分支。多分支流水线可以理解为在项目中的所有代码分支的流水线集合，Jenkins 会发现代码中 Jenkinsfile 配置文件，生成对应的分支 job。多分支流水线可以免去重复创建流水线的工作、满足临时

提交的代码分支执行流水线的需求、满足多 Pull Request（PR）分支合并到 Master 的需求。

2.6 容器服务

- **应用**

容器应用是运行在平台上的一组实例。支持以镜像和 YAML 的形式创建应用。通过关联凭据，设置网络和环境变量等来创建应用。

支持 Kubernetes 容器服务，提供高性能可伸缩的容器应用管理能力，支持企业级 Kubernetes 容器化应用的全生命周期管理。例如应用部署、扩缩容、日志查看、更新容器等周期管理。

- **配置**

支持配置字典 (ConfigMap)和保密字典 (Secret)。

配置字典 (ConfigMap)，用于保存配置数据的键值对，可以用来保存单个属性，也可以用来保存配置文件，可以更方便地处理不包含敏感信息的字符串。

保密字典 (Secret)，用于保存敏感信息，例如：密码、token、 SSH key 等，保存数据更安全，使数据免于暴露到镜像或 Pod Spec 中。保密字典支持以下几种类型：Opaque、TLS 认证、SSH 请求认证、用户名和密码、镜像服务。

- **存储**

存储提供添加持久卷声明，请求使用特定大小和访问模式的持久卷。通过数据卷挂载的方式来实现独立于 Docker 容器生命周期的持久化存储。

3 客户价值

DevOps 平台致力于解决企业中软件研发的痛点，使企业更专注于业务本身，能够更快的进行创新并灵活应对不断发展的市场动态，提高竞争力，助力企业以更低的成本实现持续交付的解决方案。

- **成熟的、经过市场验证的 DevOps 解决方案**

  平台提供方便快捷易落地的 DevOps 解决方案，帮助企业快速构建基于容器、DevOps 的开发体验，帮助研发团队缩短交付周期、提高交付效率，降低人力成本，让用户进行符合其企业场景、易落地的 DevOps 解决方案。

- **助力实现完整的 DevOps 实践**

  DevOps 在自动化方面提供全面的控制和协调，涵盖基础设施的整个层次结构，深度集成代码仓库、制品仓库、持续集成等类别中的主流工具，实现流程的自动化和零成本迁移，快速实践 DevOps。

- **降低技术门槛、提升交付质量、快速落地 DevOps**

  开箱即用的 DevOps 工具链，降低 DevOps 实践的技术门槛。简单易用的 CICD，使流水线构建和执行变得简单。支持企业级 Kubernetes 容器化应用的全生命周期管理。

- **将安全融入 DevOps**

  DevSecOps，在应用的整个生命周期内确保安全性。实现安全防护自动化，以保护整体环境和数据；同时实现持续集成/持续交付流程，并确保容器中应用的安全性。实现用户身份和访问控制功能的集中化。支持集成适用于容器的安全性扫描程序。

- **持续提升 IT 运营能力**

以质量和安全为基础支撑保障，覆盖从需求到部署上线的软件全生命周期管理。将线下 IT 生产过程转变为线上高度自动化、可视化的 IT 生产流水线，提升产品研发效率，快速响应业务需求，持续提升 IT 运营能力。