概览

架构

Kubernetes 支持矩阵

发版日志

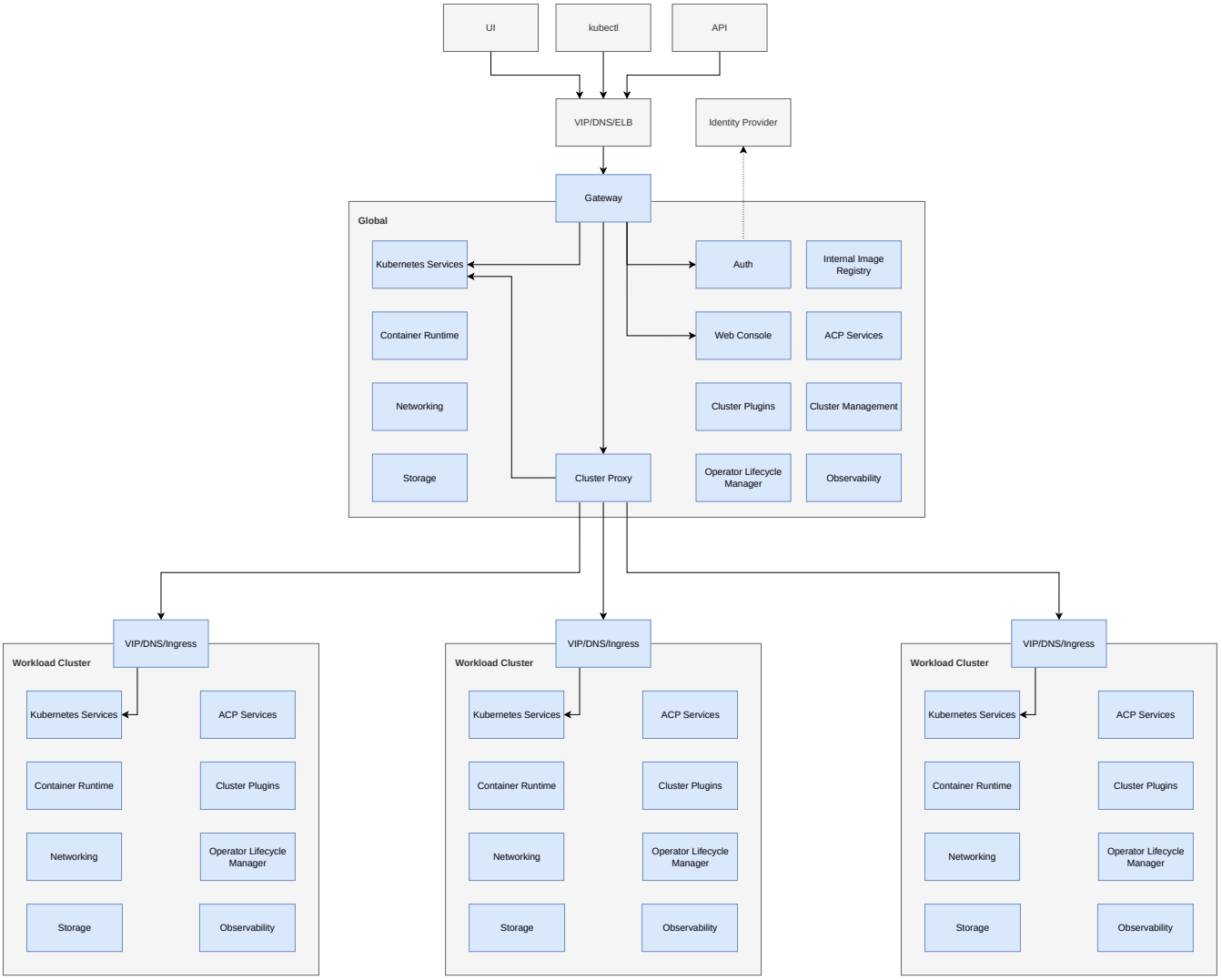
架构

目录

灵雀云容器平台 简介

灵雀云容器平台（ACP）提供了一个企业级的基于 Kubernetes 的平台，使组织能够在混合云和多云环境中一致地构建、部署和管理应用。ACP 集成了核心 Kubernetes 功能，并增强了管理、可观测性和安全服务，提供统一的控制平面和灵活的业务集群。

该架构遵循中心辐射（**hub-and-spoke**）模型，由一个 `global` 集群和多个业务集群组成。此设计既提供了集中治理，又允许独立的工作负载执行和可扩展性。



核心架构组件

Global Cluster

`global` 集群作为 ACP 的集中管理和控制中心，提供平台范围的服务，如认证、策略管理、集群生命周期操作和可观测性。它也是多集群管理的核心枢纽，支持跨集群功能。

关键组件包括：

- **Gateway**
  作为平台的主要入口，管理来自 UI、CLI（kubectl）和自动化工具的 API 请求，并将其路由到相应的后端服务。
- **Authentication and Authorization (Auth)**
  集成外部 Identity Providers (**IdPs**)，提供单点登录（SSO）和基于 RBAC 的访问控制。

- **Web Console**
  提供基于 Web 的 ACP 界面，通过 Gateway 访问平台 API。
- **Cluster Management**
  负责业务集群的注册、配置和生命周期管理。
- **ACP Services**
- **Operator Lifecycle Manager (OLM)** 和集群插件
  管理 operator 和集群扩展的安装、更新及生命周期。
- **Internal Image Registry**
  提供开箱即用的集成容器镜像仓库，支持基于角色的访问控制。
- **Observability**
  为 `global` 集群和业务集群提供集中式日志、指标和追踪。
- **Cluster Proxy**
  实现 `global` 集群与业务集群之间的安全通信。

Workload Cluster

业务集群是由 `global` 集群管理的基于 Kubernetes 的环境。每个业务集群运行隔离的应用工作负载，并继承来自中央控制平面的治理和配置。

外部集成

- **Identity Provider (IdP)**
  支持通过标准协议（OIDC、SAML）进行联合认证，实现统一的用户管理。
- **API** 和 **CLI** 访问
  用户可通过 RESTful API、Web 控制台或命令行工具（如 `kubectl` 和 `ac`）与 ACP 交互。
- 负载均衡器 **(VIP/DNS/SLB)**
  为 Gateway 及 `global` 和业务集群的入口端点提供高可用性和流量分发。

可扩展性与高可用性

ACP 设计支持水平扩展和高可用性：

- 各组件均可冗余部署，消除单点故障。
- `global` 集群支持管理数十至数百个业务集群。
- 业务集群可根据工作负载需求独立扩展。
- 通过 VIP/DNS/Ingress 实现无缝路由和故障切换。

功能视角

灵雀云容器平台（ACP）的完整功能由**ACP Core**和基于两大技术栈的扩展组成：**Operator** 和 **Cluster Plugin**。

- **ACP Core**
  ACP 的最小交付单元，提供核心能力，如集群管理、容器编排、项目和用户管理。
  - 满足最高安全标准
  - 提供最大稳定性
  - 支持最长生命周期
- 扩展
  Operator 和 Cluster Plugin 两个技术栈中的扩展可分为：
  - **Aligned** – 生命周期策略包含多个维护流，与 ACP 保持一致。
  - **Agnostic** – 生命周期策略包含多个维护流，独立于 ACP 发布。

  更多扩展详情，请参见 Extend。

技术视角

平台组件运行时
所有平台组件均作为容器运行在 Kubernetes 管理集群（即 `global` 集群）中。

高可用架构

- `global` 集群通常由至少三个控制平面节点和多个工作节点组成
- etcd 的高可用性是集群 HA 的核心；详情见*关键组件高可用机制*
- 负载均衡可由外部负载均衡器或集群内自建 VIP 提供

请求路由

- 客户端请求首先经过负载均衡器或自建 VIP
- 请求转发至运行在指定入口节点（或配置为控制平面节点）的 **ALB**（平台默认 Kubernetes Ingress Gateway）
- ALB 根据配置规则将流量路由到目标组件 Pod

副本策略

- 核心组件至少运行两个副本
- 关键组件（如 registry、MinIO、ALB）运行三个副本

容错与自愈

- 通过 kubelet、kube-controller-manager、kube-scheduler、kube-proxy、ALB 等组件协作实现
- 包括健康检查、故障切换和流量重定向

数据存储与恢复

- 控制平面配置和平台状态以 Kubernetes 资源形式存储于 etcd
- 在灾难性故障时，可通过 etcd 快照进行恢复

主备灾备

- 两个独立的 `global` 集群：主集群 和 备集群
- 灾备机制基于主集群到备集群的 etcd 数据实时同步
- 主集群故障不可用时，服务可快速切换至备集群

关键组件高可用机制

etcd

- 部署在三个（或五个）控制平面节点上
- 使用 RAFT 协议进行领导选举和数据复制
- 三节点部署可容忍最多一个节点故障；五节点部署可容忍最多两个节点故障

- 支持本地和远程 S3 快照备份

监控组件

- **Prometheus**：多实例，结合 Thanos Query 实现去重和跨地域冗余
- **VictoriaMetrics**：集群模式，包含分布式 VMStorage、VMInsert 和 VMSelect 组件

日志组件

- **Nevermore** 收集日志和审计数据
- **Kafka / Elasticsearch / Razor / Lanaya** 以分布式和多副本模式部署

网络组件（**CNI**）

- **Kube-OVN / Calico / Flannel**：通过无状态 DaemonSet 或三副本控制平面组件实现 HA

ALB

- Operator 以三副本部署，启用领导选举
- 实例级健康检查和负载均衡

自建 **VIP**

- 基于 Keepalived 的高可用虚拟 IP
- 支持心跳检测和主备切换

Harbor

- 基于 ALB 的负载均衡
- PostgreSQL 采用 Patroni 实现 HA
- Redis 采用哨兵模式
- 无状态服务多副本部署

Registry 和 **MinIO**

- Registry 以三副本部署
- MinIO 采用分布式模式，支持纠删码、数据冗余和自动恢复

Kubernetes 支持矩阵

本文档提供了 ACP 的 Kubernetes 版本支持矩阵。该信息在创建集群、升级 ACP 以及管理第三方集群时至关重要。

目录

概述

ACP 支持多个 Kubernetes 版本，覆盖不同的 ACP 版本。了解支持的版本对于以下操作非常重要：

- 创建集群 – 确定在配置新集群时可使用的 Kubernetes 版本
- 升级 **ACP** – 确保所有工作负载集群满足文档中兼容版本的要求后，再升级 global 集群
- 管理第三方集群 – 验证公有云或 CNCF 兼容的 Kubernetes 集群是否在支持的管理范围内

版本支持矩阵

下表展示了各 ACP 版本对应的 Kubernetes 版本支持情况。

> **INFO**
>
> 表中列出了 ACP 的小版本号，不区分补丁版本。补丁版本仅包含错误修复和安全更新，因此同一小版本内所有补丁版本的 Kubernetes 小版本保持一致。
>
> 从 **ACP 4.1** 开始，每个 ACP 版本仅支持 一个 **Kubernetes** 版本 用于集群创建。 这确保了新集群的一致性并简化了升级路径。

ACP 版本	支持创建集群的版本	兼容版本
ACP 4.3	1.34	1.34, 1.33, 1.32, 1.31
ACP 4.2	1.33	1.33, 1.32, 1.31, 1.30
ACP 4.1	1.32	1.32, 1.31, 1.30, 1.29
ACP 4.0	1.31, 1.30, 1.29, 1.28	1.31, 1.30, 1.29, 1.28

ACP 4.3 说明

- ACP 4.3 为平台管理的集群场景新增了对 Kubernetes 1.34 的支持。
- 升级到 ACP 4.3 时，工作负载集群兼容版本为 1.34、1.33、1.32 和 1.31。
- 这意味着从 ACP 4.0 升级到 ACP 4.3 的环境可以在升级 global 集群的同时，保持工作负载集群运行在 Kubernetes 1.31 至 1.34 版本。

第三方集群管理范围

- 对于第三方集群，ACP 4.3 接受的 Kubernetes 版本范围为 `>=1.19.0 <1.35.0`。
- 此管理范围与“兼容版本”列不同，后者是升级 ACP global 集群的权威前提条件。
- 产品文档继续仅列出通过产品验证的第三方集群支持的 Kubernetes 版本及默认 Extend 基线。
- Extend 基线的产品验证涵盖以下能力领域：

  - Operator 的安装与使用
  - Cluster Plugin 的安装与使用
  - 基于 ClickHouse 的日志
  - 基于 VictoriaMetrics 的监控
- 这并不意味着所有特定的 Operator 或 Cluster Plugin 都包含在产品验证范围内。
- 对于基线之外的特定 Operator 或 Cluster Plugin，请参考相关产品文档或联系技术支持。

升级要求

对于 ACP 4.3 及以后版本，工作负载集群只需保持在文档中兼容版本范围内，即可升级 ACP global 集群。对于 ACP 4.3，即 Kubernetes 1.31 至 1.34。

在 ACP 4.2 及更早版本中，所有工作负载集群必须在升级 ACP global 集群之前，升级到兼容版本列表中的最新 Kubernetes 版本。

发版日志

目录

已修复问题

已知问题

4.3.0

功能与增强

支持 **Kubernetes 1.34**

ACP 4.3 为平台托管集群场景增加了对 **Kubernetes 1.34** 的支持。

对于升级到 ACP 4.3，业务集群的兼容版本为 1.34、1.33、1.32 和 1.31。该兼容版本要求决定 `global` 集群是否可以升级，并且与第三方集群管理范围是分开的。

更多信息，请参见 Kubernetes 支持矩阵。

基于 **CVO** 的集群升级工作流

ACP 4.3 为 `global` 集群和业务集群引入了基于 Cluster Version Operator (CVO) 的升级工作流。

主要能力包括：

- 使用 `bash upgrade.sh` 准备升级制品和升级控制器
- 在执行前运行预检检查
- 通过 Web Console 或更新 `ClusterVersionShadow.spec.desiredUpdate` 发起升级
- 从 `cvsh.status` 查看条件、预检结果、阶段和历史记录

ACP CLI 还引入了面向升级的管理员命令，例如 `ac adm upgrade`、`ac adm upgrade status`、`--to-latest`、`--to` 和 `--allow-explicit-upgrade`，用于从当前上下文请求和排查业务集群升级问题。

操作指导请参见 升级。

独立集群插件升级

ACP 4.3 为使用 `Aligned` 或 `Agnostic` 生命周期的集群插件增加了独立升级支持。

现在 集群插件 页面会显示插件生命周期，符合条件的插件可以从列表页或详情页单独升级。`Core` 插件仍然跟随集群升级。

基于 **MicroOS** 的华为 **DCS** 上的 **global** 集群

ACP 4.3 允许管理员在华为 DCS 上创建基于 MicroOS 的不可变基础设施 `global` 集群。这将不可变运维模式从业务集群扩展到了 DCS 上的平台安装场景。

更多信息，请参见 关于不可变基础设施。

不可变基础设施中的华为云 **Stack** 支持

ACP 4.3 为华为云 Stack（HCS）增加了不可变基础设施支持。HCS provider 文档现在在不可变基础设施文档集中涵盖了 provider 概览、安装、集群创建、节点管理、集群升级和 provider API。

更多信息，请参见 关于不可变基础设施。

4.3 版本周期中的 **VMware vSphere** 支持

ACP 4.3 开始为 VMware vSphere 引入不可变基础设施支持。相关 provider 工作现在已纳入不可变基础设施文档集，而公开安装细节和最终的插件命名仍在持续发布中。

更多信息，请参见 关于不可变基础设施。

新的 **Web Console Preview** 入口

ACP Core 现在提供下一代 Web Console 体验所需的顶部导航锚点。当 Alauda Container Platform Web Console Base 安装在 `global` 集群上时，**Container Platform** 和 **Administrator** 视图中的用户可以通过一个单独的浏览器标签页中的 **Preview Next-Gen Console** 入口打开新控制台。

该体验旨在支持渐进式迁移，并可与 global 集群上的 Web Console Base 插件以及业务集群上的 Web Console Collector 插件配合使用。

containerd 2.0 基线

ACP 4.3 将平台运行时基线升级到 containerd 2.0。对于依赖自定义 containerd 配置的环境，请在升级前检查相关运行时运维流程。

扩展第三方集群管理范围

对于第三方集群，ACP 4.3 现在接受范围为 `>=1.19.0 <1.35.0` 的 Kubernetes 版本。

该管理范围与用于判断 `global` 集群是否可升级的兼容 Kubernetes 版本是分开的。

产品文档会继续仅发布已通过产品验证、适用于第三方集群支持和默认 Extend 基线的 Kubernetes 版本。

Extend 基线的产品验证覆盖以下能力领域：

- 安装和使用 Operators
- 安装和使用集群插件
- 基于 ClickHouse 的日志
- 基于 VictoriaMetrics 的监控

这并不意味着所有具体的 Operators 或集群插件都经过了产品验证。

对于超出该基线的具体 Operators 或集群插件，请参考相关产品文档或联系技术支持。

更多信息，请参见 Kubernetes 支持矩阵 和 导入标准 Kubernetes 集群。

监控插件配置扩展

ACP 4.3 扩展了监控插件的配置选项，使监控部署更容易适配 infra-node 放置和不同的存储布局。

对于使用 VictoriaMetrics 的 ACP Monitoring，管理员现在可以：

- 配置插件级别的节点选择器和容忍度，将工作负载放置到专用 infra 节点上
- 当 `Storage Type` 为 `LocalVolume` 时，为 VictoriaMetrics 配置数据存储目录
- 移除 VictoriaMetrics 部署此前的三节点限制

对于使用 Prometheus 的 ACP Monitoring，管理员现在可以配置插件级别的节点选择器和容忍度，从而通过插件配置将监控工作负载调度到专用 infra 节点。

> **WARNING**
>
> 如果你之前使用 patch 资源或基于 override 的自定义来分别定义节点选择器或容忍度，升级到 ACP 4.3 后需要更新插件配置。在更新后的插件配置生效后，必须移除相关的 patch 资源或 override 设置。

操作指导请参见 安装 和 为监控规划 Infra 节点。

StatefulSet 跨集群应用灾难恢复方案

本版本为有状态应用引入了跨集群灾难恢复能力。该方案基于 Active-Passive 双中心架构，结合 **Alauda Build of VolSync** 异步数据同步和 **GitOps** 配置分发，实现分钟级 RTO 故障切换。

主要亮点：

- 主集群处理所有 `read/write` 流量；备用集群通过周期性 rsync 快照维护热数据副本（RPO > 0）。
- 支持三种运维场景：计划迁移、紧急故障切换和故障回切。
- 备用集群默认以 `replicas=0` 运行；存储和计算资源保持冷备用状态，不承载业务流量。
- 适用于不要求严格零数据丢失（RPO = 0）的工作负载。对于金融或事务型核心应用，请改用原生数据库复制。

更多详情，请参见：有状态应用的跨集群应用灾难恢复 ↗

Alauda Container Platform Registry - 镜像管理增强

本版本引入了 `ac images` 和 `ac adm prune images` 命令，使得可以通过命令行对 Registry 镜像进行全生命周期管理。

- `ac get images`：列出 Registry 中的镜像。结果范围限定为当前用户有权限的命名空间，支持按命名空间过滤以及多种输出格式（`table`、`json`、`yaml`、`wide`）。
- `ac delete images`：按 Registry 路径删除一个或多个镜像标签。内置命名空间权限检查；默认以 dry-run 模式运行以预览影响，并且需要 `--confirm` 才会执行实际删除。

- `ac adm prune images`：管理员命令，用于清理未被任何集群 Pod 引用的镜像 manifest。灵活的清理策略包括保留时长、保留数量、允许列表以及 `--all` 范围。清理后可选择触发 Registry GC。也支持通过 CronJob 执行定时清理。

更多详情，请参见：集群镜像 Registry 清理：管理员手动与定时任务指南↗

Alauda Container Platform Project Application Essential（Alpha）

本版本引入了 **Alauda Container Platform Project Application Essential** 插件，它基于全新的 **Next-Gen Console** 前端框架构建。部署在 `global` 集群上时，它可从以项目为中心的视角提供跨集群应用编排和全生命周期管理，并完全遵循用户权限。

主要亮点：

- 跨集群编排：在单个项目内统一将应用部署到多个成员集群。
- 全生命周期管理：支持 `create`、`update`、`scale`、`rollback`、`delete`，并可实时同步各集群中的应用状态。
- 项目隔离：所有操作都限定在项目边界内，确保项目之间天然隔离。
- 权限感知：严格执行 RBAC 权限，仅显示用户有权访问的资源。

底层网络和 **Egress Gateway** 增强

ACP 4.3 扩展了核心 CNI 网络能力，重点增强了底层网络接入和 egress gateway 操作。

主要增强包括：

- 改进 egress gateway 工作负载的高可用和快速切换设计，降低节点维护或故障切换期间对服务的影响。
- 为 egress gateway Pods 提供资源保护指导和平台支持，帮助降低流量高峰或从节点扩容时节点资源争用的风险。
- 支持为 egress gateway 工作负载配置污点，从而更好地在专用节点上实现放置隔离。
- 支持管理底层 NIC 的 VLAN 子接口。
- 新增对子网资源的 YAML 编辑支持。
- 新增中央网关节点选择器设置支持。
- 新增中央网关场景的子网 CRD 支持。

这些增强使 ACP 在复杂企业网络环境中更具适应性，并简化了从早期暴露模型向基于底层网络设计的迁移。

Gateway API 增强

ACP 4.3 进一步强化了 **Gateway API** 作为平台关键 Layer 7 负载均衡能力的地位。

主要增强包括：

- 支持基于 host-network 的网关部署场景。
- 支持通过 `metalLB + Envoy Gateway proxy + underlay` 暴露服务，使业务流量可以绕开管理网络。
- 支持 Gateway API 的自定义 VIP 地址，帮助在重建或生命周期变化时保持服务暴露地址稳定。

基于 **PVC** 保护的有状态应用灾难恢复

ACP 4.3 为有状态工作负载引入了更强的灾难恢复能力，包括 基于 **PVC** 的灾难恢复，以及为 MinIO 等以存储为后端的应用提供 基于 **VolSync** 的备份和恢复工作流 支持。

这一增强提升了有状态应用的跨集群恢复准备度，并为以存储为主的生产环境提供了更实用的保护路径。

Ceph 存储管理增强

ACP 4.3 改进了存储运维以及基于 Ceph 的工作负载支持。

主要增强包括：

- 新增通过 UI 将磁盘放入不同 Ceph pool 的支持。
- 改进了 Ceph 磁盘替换场景的运维支持。

这些改动提升了日常存储运维能力，并使基于 Ceph 的环境在生产中更易管理。

虚拟化平台增强

ACP 4.3 带来了多项重要的虚拟化相关改进。

主要增强包括：

- 改进 VM 创建和展示工作流。
- 新增虚拟化相关场景对 Astra Linux 的支持。
- 新增虚拟机对多 NIC 和 NIC 热插拔能力的支持。

这些增强改善了虚拟化的易用性，并扩展了企业环境中的来宾工作负载兼容性。

弃用和移除的功能

运营统计功能下线

计量和计费插件现在已经正式可用，并且完整覆盖了此前由运营统计功能提供的能力。因此，平台管理 下的顶层 运营统计 入口将被移除。

- 对于新部署的平台，将不再安装运营统计组件。如果你需要计量或计费能力，请使用 **Cost Management** 插件。
- 对于已升级的平台，运营统计进行的计量采集会在升级后停止，而历史数据仍然可用。如果你需要数据清理或迁移，请提交支持请求。

已修复问题

- 修复了 olm-registry pod 持续重启导致 OperatorHub 无法正常使用的问题。该问题由 CIS 合规加固时添加的 `seccompProfile: RuntimeDefault` 安全配置引起，该配置拦截了 CGO 操作所需的 `clone` 系统调用。已调整 seccomp 配置以允许必要的系统调用，同时保持安全合规性。已在 ACP 4.3.0 修复。
- 修复了当集群安装 60+ 个 Operator 时，原生应用创建接口权限校验极慢（10秒以上）的性能问题。已在 ACP 4.3.0 修复。
- 当使用 Alauda Container Platform Cluster Enhancer 提供的 etcd 备份功能时，如果用户配置将 etcd 备份到 S3 存储，插件无法获取 secretRef 中引用的 Secret 对象。原因是插件缺少读取 Secret 的 RBAC 权限，导致 S3 认证信息获取失败。此问题已在 ACP 4.3.0 中修复。
- 当使用 Alauda Container Platform Monitoring for VictoriaMetrics 且多个集群共享同一个 Storage 时，告警策略 cpaas-certificates-rule 存在两个问题：告警触发时无法区分来自哪个集群，以及该策略会监控客户的 secret 而非仅监控平台证书。
- 修复 metis 组件 storage limit 配置太小，在超出限制后导致 metis 容器重启

- 修复了向内置镜像仓库推送包含超多数据层（100+）的容器镜像时失败的问题
- 修复了在业务应用使用自定义 ServiceAccount 的场景下，imagePullSecret 未自动注入导致镜像拉取失败的问题。
- 修复了在 image-registry 的 imagePullSecret 自动轮询通过“新建 Secret + 删除旧 Secret”进行轮转、且历史 Pod 仍引用旧 Secret 并在 Secret 过期后才启动的场景下，Pod 无法拉取镜像的问题。
- 修复了命名空间创建特殊场景下触发的异常。当进入创建命名空间页面的时候，如果页面请求返回比较慢，刚进页面的时候可能会导致没有默认选中的集群信息，从而触发页面其他接口的报错，导致页面项目配额无法正常展示。
- 调整了实时日志组件中的部分文案， Logging has ended => End of logs
- 修复了 windows 上编辑 configmap 的时候，换行符和 mac 上行为不一致的问题。

已知问题

- 使用 violet push 推送 chart package 时，虽然 push 显示成功，但该 package 在 public-charts 仓库中可能无法看到。
  临时解决方案： 重新 push 一次。
- 使用 violet push 推送 chart package 时，虽然 push 显示成功，但该 package 在 public-charts 仓库中可能无法看到。
  临时解决方案： 重新 push 一次。
- 通过 YAML 创建应用时使用 defaultMode 字段导致应用创建失败。
  操作路径：容器平台 → 应用管理 → 应用列表 → 通过 YAML 创建（Create from YAML），当提交的 YAML 文件中包含 defaultMode 字段（通常用于 ConfigMap/Secret 的卷挂载权限配置）时，应用创建会失败并返回校验错误。
  解决方案：创建应用前手动移除 YAML 中所有 defaultMode 声明。
- 当 Helm Chart 中设置了 pre-delete post-delete hook。
  执行删除模板应用，卸载 Chart 时，遇到某些原因导致 hook 执行失败，进而导致应用无法删除。需要排查原因，并优先解决 hook 执行失败的问题。