安装

本文档将提供有关安装 ACP 的所有信息。

概览

安装准备

安装

global 集群灾难恢复

global 集群灾难恢复

概览

通过本指南，您将完成 **ACP Core** 的安装。 如果您需要了解 **ACP Core** 的概念，请参阅 Architecture。

安装 **ACP Core** 指的是部署 `global` 集群的过程。

安装完成后，您可以 创建新的工作负载集群 或 连接现有集群，并安装额外的 **Extensions** 以增强平台功能。

> **INFO**
>
> 安装前，请确保已完成容量规划、环境预处理及先决条件检查，确保每个节点的硬件、网络和操作系统符合要求。 以下内容涵盖平台架构设计、安装方法及关键术语说明，帮助您在实际安装过程中把握核心要点。

目录

安装方法

`global` 集群的安装过程主要分为三个阶段：

1. 准备阶段

- 先决条件检查：确保所有节点的硬件、网络和操作系统满足要求，如内核版本、CPU 架构和网络配置。
- 安装包下载：登录 灵雀云 Customer Portal 获取最新安装包。
- 节点预处理：完成所有节点的预处理工作。

2. 执行阶段

- 安装包上传与解压：将安装包上传至目标控制平面节点（推荐目录：`/root/cpaas-install`），并解压安装资源。
- 启动安装程序：在控制平面节点执行安装脚本（如 `bash setup.sh`），根据实际环境选择 IP 协议模式（IPv4/IPv6/双栈）及 VIP 配置。
- 参数配置：访问安装程序提供的 Web UI，依次设置 Kubernetes 版本、集群网络、节点名称、访问地址等关键参数，完成 `global` 集群的安装。

3. 验证阶段

- 系统状态检查：安装完成后，登录平台 Web UI 检查集群状态及各组件运行状况。
- **CLI** 验证：使用命令行工具检查集群资源状态，确保所有服务正常运行，无异常或失败。

后续章节将详细说明各安装阶段的操作步骤、配置参数及验证方法。请仔细阅读并完成相应准备工作后，再进行正式安装。

附录 — 灵雀云 Customer Portal

灵雀云 **Customer Portal** 是 灵雀云 统一的客户服务与交付平台，提供所有产品相关资源和支持服务的集中访问入口。它是客户、合作伙伴及交付团队获取软件包、文档、支持及许可证管理的官方入口，确保安全且一致的访问体验。

目的与概述

灵雀云 Customer Portal 整合了从安装配置到维护支持的全生命周期产品管理，将所有关键资源汇聚于一处，确保每次部署均基于经过验证的软件版本和官方技术指导。

主要功能

- 产品下载
  提供经过验证的安装及升级包，确保部署与最新支持的产品版本保持一致。
- 知识库
  提供全面的产品文档、技术文章、故障排查指南及最佳实践，协助安装、配置和运维。
- 支持工单
  支持用户在线提交、跟踪和管理支持请求，确保及时解决问题并全程可见支持进度。
- 应用 **Marketplace**
  提供官方及第三方扩展的精选集合，可安装以扩展或定制平台功能。
- 许可证管理
  支持软件许可证的申请、激活及续期，确保所有环境中的许可证使用可追溯且合规。

使用指导

在开始安装或升级操作前，用户应使用授权账号登录 灵雀云 **Customer Portal**，下载所需安装包并核实许可证状态。 对于客户交付及生产环境，灵雀云 Customer Portal 上发布的版本和文档应始终视为部署和维护的 官方基线。

安装准备

前提条件

在安装 `global` 集群之前，您需要准备符合要求的硬件、网络和操作系统环境。

INFO

1. 平台当前不支持在已有 Kubernetes 环境中直接安装 `global` 集群。如果您的环境中已有 Kubernetes 集群，请先备份数据并清理环境后再进行安装。
2. 如果计划使用 global Cluster 灾备功能，请先阅读 global Cluster 灾备。
3. 确保所有新节点满足 节点要求。
4. 磁盘的性能和容量也必须满足 磁盘配置要求。

目录

资源规划

本节提供安装 ACP 前的资源规划指南。
请根据您的环境和使用需求选择合适的部署场景，并相应准备资源。

> **INFO**
>
> 以下建议仅涵盖成功安装 **Global cluster** 所需的最低资源。
>
> 它们不包括部署在 Global cluster 上的任何额外扩展或组件所需的资源。
>
> 有关各扩展的详细需求，请参阅对应组件文档。

部署架构

> **WARNING**
>
> 对于 ARM 架构（如鲲鹏 920），建议配置提升至 x86 最低配置的 **2** 倍，但不少于 **1.5** 倍。
>
> 例如：x86 需要 8 核 16GB，则 ARM 至少应达到 12 核 24GB，推荐配置为 16 核 32GB。

安装前，您必须确定哪种部署架构最适合您的使用场景。
ACP 支持以下三种常见部署架构：

- 多集群
  如果需要集中管理多个 Kubernetes 集群，请选择此架构。
  在此模式下，ACP 由一个 **Global cluster** 和多个 业务集群 组成。
  不建议在 Global cluster 上运行非平台工作负载，以免影响平台稳定性和性能。
- 单集群
  如果计划只安装一个集群并直接在其上运行工作负载，请选择此架构。
  在此模式下，Global cluster 同时作为业务集群，因此相比多集群模式中纯 Global-only 部署需要 更多资源。
- 单节点

  > **WARNING**

此架构仅用于测试或概念验证，禁止用于生产环境。

单节点

下表列出了以 单节点 模式安装 ACP 的最低硬件要求。

资源	最低要求
CPU	12 核
内存	24GB
存储	存储容量

单集群

在此模式下，Global cluster 既作为控制平面，也作为业务集群。
Global cluster 的控制平面节点数必须为 3 个。

总资源需求由两部分组成：

- Global cluster 本身的基础资源
- 在同一集群上运行工作负载的额外资源

高可用 Global cluster 所需资源如下：

资源	最低要求
CPU	8 核
内存	16GB
存储	存储容量

如需估算工作负载所需的额外资源，请参阅
评估业务集群资源

多集群

管理多个业务集群时，Global cluster 的资源使用会随着管理集群数量成比例增加。
额外开销主要来自集群注册、监控和控制平面同步。

根据管理集群数量估算 Global cluster 所需资源，请参阅 评估 **Global cluster** 资源

网络

安装前，请确保准备好所需的网络资源。

如果环境中有硬件负载均衡器，推荐使用；如果没有，可以启用 `Self-built VIP`，该方案通过 keepalived 提供软件负载均衡。

注意： `Self-built VIP` 不支持域名配置。

网络资源

资源	是否必需	数量	说明
`global` VIP	必需	1	供集群内节点访问 kube-apiserver 使用，配置于负载均衡设备以保证高可用。 该 IP 也可作为平台 Web UI 的访问地址。 与 `global` 集群处于同一网络的业务集群也可通过此 IP 访问 `global` 集群。
外部 IP	可选	按需	当存在不与 `global` 集群处于同一网络的业务集群（如混合云场景）时，必须提供。其他网络的业务集群通过此 IP 访问 `global` 集群。 该 IP 需配置于负载均衡设备以保证高可用。 该 IP 也可作为平台 Web UI 的访问地址。
域名	推荐	按需	推荐用于集群端点和平台访问地址。请提前提供并确保域名解析正确。 使用域名的好处在于，如果集群安装后需要更换 VIP，可通过更新 DNS 记录轻松完成，且不会影响集

资源	是否必需	数量	说明
			群运行。 以下情况必须使用域名： • `global` 集群需要支持 IPv6 访问； • 计划为 `global` 集群实施灾备方案。
证书	可选	按需	建议使用受信任证书以避免浏览器安全警告；若未提供，安装程序将生成自签名证书，但使用 HTTPS 时可能存在安全风险。

> **注意**
>
> 如果平台需要配置多个访问地址（例如内外网地址），请根据上表提前准备相应的 IP 地址或域名。
> 您可以在安装参数中配置，或安装后根据产品文档进行添加。

网络配置

类型	需求说明
网络带宽	集群内带宽必须 ≥1 Gbps（推荐 10 Gbps）。 跨集群带宽必须 ≥100 Mbps（推荐 1 Gbps）。 带宽不足可能严重影响数据查询性能。
网络延迟	集群内延迟必须 ≤10 ms。 跨集群延迟必须 ≤100 ms（推荐 ≤30 ms）。
网络策略	请参考 LoadBalancer 转发规则 确保必要端口已开放。
IP 地址段	`global` 集群节点应避免使用 172.17-18 网段。如已使用，请调整 nerdctl 配置（添加 bip 参数）以避免冲突。

LoadBalancer 转发规则

该规则用于确保 `global` 集群能正常接收来自 LoadBalancer 的流量。请根据下表检查网络策略，确保相关端口已开放。

源 **IP**	协议	目标 **IP**	目标端口	说明
`global` VIP，外部 IP	TCP	所有控制平面节点 IP	443	通过 HTTPS 协议为平台 Web UI、镜像仓库和 Kubernetes API Server 提供访问服务。默认端口为 `443`。如需使用自定义 HTTPS 端口，请执行以下操作： • 将端口转发规则中的目标端口替换为自定义端口号。 • 后续在 Web UI 安装参数中填写自定义端口号。
`global` VIP，外部 IP	TCP	所有控制平面节点 IP	6443	该端口为集群内节点访问 Kubernetes API Server 提供服务。
`global` VIP，外部 IP	TCP	所有控制平面节点 IP	11443	该端口为集群内节点访问镜像仓库提供服务。 注意：如果计划使用外部镜像仓库替代 `global` 集群默认镜像仓库，则无需配置此端口。

提示

- 建议在 LoadBalancer 上配置健康检查以监控端口状态。
- 如果计划为 `global` 集群实施灾备方案，需要为所有控制平面节点开放端口 `2379`，用于主备集群间 ETCD 数据同步。

- 平台默认仅支持 HTTPS。如需支持 HTTP，需为所有控制平面节点开放 HTTP 端口。

下载 Core Package

在安装之前，您需要先下载 **Core Package**。

> **INFO**
>
> 从 灵雀云容器平台 v4.1 开始，如果您同时下载了 **Core Package** 和 **Extensions Packages**，必须先完成 **Core Package** 的安装，才能上传并安装 **Extensions Packages**。

登录 灵雀云 **Customer Portal** 下载 **Core Package**。

提供适用于 **x86**、**ARM** 和 **hybrid** 架构的安装包。hybrid 包含了 x86 和 ARM 的镜像，因此包体积较大。请选择最符合您环境的安装包。

如果您尚未注册账号，请联系技术支持。

目录

从单架构迁移到 **Hybrid**

如果您最初安装的是 x86 或 ARM Core Package，但后续需要支持另一种架构，则必须重新下载 **hybrid Core Package** 并执行以下步骤：

1. 将新下载的 hybrid Core Package 上传到 global 集群的任一控制平面节点。

2. 解压安装包，使用其中的 `upgrade.sh` 脚本将多架构镜像同步到您的镜像仓库：

```
bash upgrade.sh --only-sync-image=true
```

3. 脚本执行完成后，检查 `cluster.platform.tkestack.io` 资源，确认是否存在标签 `cpaas.io/node-arch-constraint`。如果存在，必须将其删除：

```
kubectl get cluster.platform.tkestack.io global -oyaml | grep cpaas.io/
node-arch-constraint
# 如果有输出，则编辑该资源删除该标签；否则可跳过此步骤。
kubectl edit cluster.platform.tkestack.io global   ### 编辑 labels 字段,
删除 cpaas.io/node-arch-constraint
```

节点预处理

在安装 `global` 集群之前，所有节点（控制平面节点和工作节点）必须完成预处理。

目录

支持的操作系统和内核版本

下表列出了支持的操作系统、其验证版本及对应测试的内核版本。

平台对官方支持的版本执行严格的版本匹配策略：

- 操作系统版本 **(x.y.z)**：补丁版本（`z`）可变，但主版本和次版本（`x` 和 `y`）必须严格匹配验证版本。不支持修改 `x` 或 `y`。
- 内核版本 **(x.y.z-build)**：构建后缀（`build`）可变，但核心内核版本（`x.y.z`）必须严格匹配测试版本。不支持修改 `x.y.z`。

INFO

- 仅支持官方操作系统自带的内核版本。如果操作系统、内核版本或 CPU 架构不符合要求，请联系技术支持。
- 麒麟 V10、V10-SP1 和 V10-SP2 存在已知内核问题，可能导致 **NodePort** 网络访问失败，建议升级至 麒麟 **V10-SP3**。

x86

Red Hat Enterprise Linux (RHEL)

- RHEL 7.8: `3.10.0-1127.el7.x86_64`
- RHEL 8.0: `4.18.0-80.el8.x86_64`
- RHEL 8.6: `4.18.0-372.9.1.el8.x86_64`
- RHEL 8.10: `4.18.0-553`
- RHEL 9.6: `5.14.0-570.12.1`

注意： RHEL 7.8 不支持 **Calico Vxlan IPv6**。

CentOS

- CentOS 7.6 到 7.9：`3.10.0-1127` 和 `3.10.0-1160`

注意： CentOS 不支持 **Calico Vxlan IPv6**。

Ubuntu

- Ubuntu 20.04 LTS：`5.4.0-135-generic`
- Ubuntu 22.04 LTS：`5.15.0-56-generic`

注意： 不支持 Ubuntu HWE（硬件启用）版本。

Kylin Linux Advanced Server

- 麒麟 V10 SP3：`4.19.90-52.22.v2207.ky10.x86_64`

ARM

Kylin Linux Advanced Server

- 麒麟 V10 SP3：`4.19.90-52.22.v2207.ky10.aarch64`

注意： ARM 架构仅支持 `Kunpeng 920`，其他型号请联系技术支持。

执行快速配置脚本

ACP 安装包提供了用于快速配置节点的脚本。

解压安装包后，在 `res` 目录下获取 `init.sh` 脚本文件。将脚本文件复制到节点，并确保拥有 `root` 权限。

执行脚本：

```
bash init.sh
```

> 警告
>
> `init.sh` 无法保证以下所有检查均被正确处理，您仍需继续执行以下步骤。

节点检查

以下列出了节点必须完成的所有检查。根据节点的角色，所需检查项会有所不同。例如，部分检查仅适用于控制平面节点。

检查分为两类：

- ✅ 表示必须通过的检查。

- ⚠️ 表示在特定场景下必须满足的检查。请根据说明判断是否满足对应条件，若满足则必须解决。

检查列表如下：

- 操作系统和内核
  - ✅ 机器的 grub 启动配置必须包含参数 `transparent_hugepage=never`。
  - ✅ CentOS 7.x 系统的 grub 启动配置必须包含参数 `cgroup.memory=nokmem`。
  - ✅ 检查内核模块 `ip_vs`、`ip_vs_rr`、`ip_vs_wrr` 和 `ip_vs_sh` 是否已启用。
  - ⚠️ 当内核版本低于 4.19.0（或 RHEL 低于 4.18.0）时，检查内核模块 `nf_conntrack_ipv4` 和（IPv6 时）`nf_conntrack_ipv6` 是否已启用。
  - ⚠️ 若 `global` 集群计划使用 `Kube-OVN` CNI，必须启用内核模块 `geneve` 和 `openvswitch`。
  - ✅ 禁用 apparmor/selinux 和防火墙。
  - ✅ 禁用 `swap`。
- 用户和权限
  - ✅ 节点的 SSH 用户具有 `root` 权限，且可无密码使用 `sudo`。
  - ✅ `/etc/ssh/sshd_config` 中的 `UseDNS` 和 `UsePAM` 参数必须设置为 `no`。
  - ✅ 执行 `systemctl show --property=DefaultTasksMax` 返回 `infinity` 或非常大的值，否则需调整 `/etc/systemd/system.conf`。
- 节点网络
  - ✅ `hostname` 必须符合以下规则：
    - 不超过 36 个字符。
    - 以字母或数字开头和结尾。
    - 仅包含小写字母、数字、`-` 和 `.`，且不能包含 `.-`、`..` 或 `-.`。
  - ✅ `/etc/hosts` 中的 `localhost` 必须解析为 `127.0.0.1`。
  - ✅ `/etc/resolv.conf` 文件必须存在且包含 `nameserver` 配置，但不能包含以 172 开头的地址（需禁用 systemd-resolved）。

- ⚠️ `/etc/resolv.conf` 文件不应配置 search 域（如必须配置，请参见 配置 Search Domain）。
- ✅ 机器的 IP 地址不能是回环、多播、链路本地、全 0 或广播地址。
- ✅ 执行 `ip route` 必须返回默认路由或指向 `0.0.0.0` 的路由。
- ✅ 节点不得占用以下端口：
  - 控制平面节点：`2379`、`2380`、`6443`、`10249` ~ `10256`
  - 安装器所在节点：`8080`、`12080`、`12443`、`16443`、`2379`、`2380`、`6443`、`10249` ~ `10256`
  - 工作节点：`10249` ~ `10256`
- ✅ 若集群使用 **Kube-OVN** 或 **Calico**，确保以下端口未被占用：
  - **Kube-OVN**：`6641`、`6642`
  - **Calico**：`179`
- ⚠️ 确保 nerdctl 需要的网络段 `172.17.x.x` ~ `172.18.x.x` 的 IP 地址未被占用。如该网段 IP 被占用且无法更改，请联系技术支持。

- 软件和目录要求
  - ✅ 必须安装以下工具：`ip`、`ss`、`tar`、`swapoff`、`modprobe`、`sysctl`、`md5sum` 以及 `scp` 或 `sftp`。
  - ⚠️ 若计划使用本地存储 **TopoLVM** 或 **Rook**，需安装 `lvm2`。
  - ✅ 不允许存在 `/etc/systemd/system/kubelet.service` 文件。
  - ✅ `/tmp` 挂载参数中不得包含 `noexec`。
  - ✅ 删除与 `global` 集群组件冲突的软件包（详见 删除冲突软件包）。
  - ✅ 必须删除以下文件（如存在）：
    - `/var/lib/docker`
    - `/var/lib/nerdctl`
    - `/opt/nerdctl/`
    - `/var/lib/containerd`
    - `/var/log/pods`
    - `/var/lib/kubelet/pki`

- 跨节点检查
  - ✅ `global` 集群内节点间不得存在网络防火墙限制。
  - ✅ 集群中每个节点的 `hostname` 必须唯一。
  - ✅ 所有节点的时区必须统一，且时间同步误差 ≤ 10 秒。

附录

删除冲突软件包

安装前，节点上可能已在 docker/nerdctl/containerd 环境中运行应用，或已安装与 `global` 集群冲突的软件。因此，需要检查并卸载冲突的软件包。

> 危险
>
> - 为避免应用中断或数据丢失，请务必确认是否存在冲突软件包。发现冲突时，请制定应用切换方案并备份数据后再卸载。
> - 卸载冲突软件包后，还需检查 `/usr/local/bin/` 等目录中是否存在其他潜在冲突的二进制文件（如与 docker、nerdctl、containerd、runc、podman、容器网络、容器运行时或 Kubernetes 相关的软件）。

以下命令供参考。

CentOS / RedHat

检查：

```
for x in \
    docker docker-client docker-common docker-latest \
    podman-docker podman \
    runc \
    containernetworking-plugins \
    apptainer \
    kubernetes kubernetes-master kubernetes-node kubernetes-client \
    ; do
    rpm -qa | grep -F "$x"
done
```

卸载：

```
for x in \
    docker docker-client docker-common docker-latest \
    podman-docker podman \
    runc \
    containernetworking-plugins \
    apptainer \
    kubernetes kubernetes-master kubernetes-node kubernetes-client \
    ; do
    yum remove "$x"
done
```

Ubuntu

检查：

```
for x in \
    docker.io \
    podman-docker \
    containerd \
    rootlesskit \
    rkt \
    containernetworking-plugins \
    kubernetes \
    ; do
    dpkg-query -l | grep -F "$x"
done

for x in \
    kubernetes-worker \
    kubectl kube-proxy kube-scheduler kube-controller-manager kube-ap
iserver \
    k8s microk8s \
    kubeadm kubelet \
    ; do
    snap list | grep -F "$x"
done
```

卸载：

```
for x in \
    docker.io \
    podman-docker \
    containerd \
    rootlesskit \
    rkt \
    containernetworking-plugins \
    kubernetes \
    ; do
    apt-get purge "$x"
done

for x in \
    kubernetes-worker \
    kubectl kube-proxy kube-scheduler kube-controller-manager kube-ap
iserver \
    k8s microk8s \
    kubeadm kubelet \
    ; do
    snap remove --purge "$x"
done
```

Kylin

检查：

```
for x in \
    docker docker-client docker-common \
    docker-engine docker-proxy docker-runc \
    podman-docker podman \
    containernetworking-plugins \
    apptainer \
    containerd \
    kubernetes kubernetes-master kubernetes-node kubernetes-client ku
bernetes-kubeadm \
    ; do
    rpm -qa | grep -F "$x"
done
```

卸载：

```
for x in \
    docker docker-client docker-common \
    docker-engine docker-proxy docker-runc \
    podman-docker podman \
    containernetworking-plugins \
    apptainer \
    containerd \
    kubernetes kubernetes-master kubernetes-node kubernetes-client ku
bernetes-kubeadm \
    ; do
    yum remove "$x"
done
```

配置 Search Domain

在 Linux 操作系统中，`/etc/resolv.conf` 文件用于配置 DNS 客户端的域名解析设置。`search` 行指定 DNS 查询的域搜索路径。

配置要求

- 域数量： `search` 行中的域数量应小于 `domainCountLimit - 3`（默认 `domainCountLimit` 为 32）。
- 单个域名长度： 每个域名不得超过 253 个字符。
- 总字符长度： 所有域名及空格的总字符数不得超过 `MaxDNSSearchListChar`（默认 2048）。

示例

```
search domain1.com domain2.com domain3.com
```

- 域数量为 3。
- 单个域名长度，如 `domain1.com`，为 11。
- 总字符长度为 35，即 11 + 11 + 11 + 2（两个空格）。

> 警告

- 若 `/etc/resolv.conf` 文件中的 `search` 行不满足上述限制，可能导致 DNS 查询失败或性能下降。
- 修改 `/etc/resolv.conf` 文件前，建议先备份该文件。

安装

本节介绍 `global` 集群的具体安装步骤。

开始安装前，请确保已完成前置检查、安装包下载与校验、节点预处理等准备工作。

目录

流程

1 上传并解压安装包

将 Core Package 安装包上传至 `global` 集群控制平面节点的任意一台机器，并按照以下命令解压：

```
# 假设机器上已存在 /root/cpaas-install 目录
tar -xvf {Path to Core Package File}/{Core Package File Name} -C /root/cpaas-install
cd /root/cpaas-install/installer || exit 1
```

INFO

- 该机器将在 `global` 集群安装完成后成为第一个控制平面节点。
- Core Package 解压后，至少需要 **100GB** 磁盘空间，请确保存储资源充足。
- 如果已下载扩展包，请先完成 ACP Core 安装，再按照扩展上传并安装扩展。

2 启动安装程序

执行以下安装脚本启动安装程序。安装程序启动成功后，命令行终端将输出 Web 控制台访问地址。

等待约 5 分钟后，可使用 PC 上的浏览器访问安装程序提供的 Web 控制台。

```
bash setup.sh
```

WARNING

请确保安装程序所在节点的 IP 地址及端口 8080 可正常访问，以保证安装程序启动成功后 Web 控制台能够顺利访问。

IP 协议族

```
bash setup.sh --ip-family ipv6
```

如果计划创建 Single-stack Network IPv6 的 `global` 集群，启动安装程序时必须显式指定 `--ip-family ipv6` 参数。否则，安装程序默认创建支持 Single-stack Network IPv4 和 Dual-stack Network 的 `global` 集群。

3 参数配置

按照页面引导完成安装参数配置后，确认安装。

参数说明 提供了关键参数的详细描述，请仔细阅读并根据实际需求配置。

4 验证安装成功

安装完成后，页面将显示平台访问 URL。点击 访问 按钮打开平台 Web UI，验证平台是否可访问。

接着，在安装节点执行以下命令验证安装状态：

```
# 检查是否存在失败的 Charts
kubectl get apprelease --all-namespaces
# 检查是否存在非 Running 或 Completed 状态的 Pods
kubectl get pod --all-namespaces | awk '{if ($4 != "Running" && $4 != "Completed")print}' | awk -F'[/ ]+' '{if ($3 != $4)print}'
```

5 安装产品文档插件

> **INFO**
>
> **Alauda Container Platform Product Docs** 插件提供平台内访问产品文档的功能。平台内所有帮助链接均指向该文档。如果未安装此插件，点击平台内帮助链接将导致 404 访问错误。

1. 进入 管理员。
2. 在左侧边栏点击 **Marketplace** > **Cluster Plugins**，选择 `global` 集群。
3. 找到 **Alauda Container Platform Product Docs** 插件，点击 安装。

参数说明

参数	说明

Kubernetes 版本	所有可选版本均经过严格测试，确保稳定性和兼容性。 推荐：选择最新版本以获得最佳功能和支持。
集群网络协议	支持三种模式：IPv4 单栈、IPv6 单栈、IPv4/IPv6 双栈。 注意：若选择双栈模式，需确保所有节点均正确配置 IPv6 地址；设置后网络协议不可更改。
集群访问端点	输入预先准备好的域名。 若无域名，则输入预先准备好的 `global VIP`。 `自建 VIP` 默认禁用，仅在未提供 LoadBalancer 时启用。 使用 `自建 VIP` 时需满足以下条件： • 有可用的 VRID； • 主机网络支持 VRRP 协议； • 所有控制平面节点及 VIP 必须处于同一子网。 提示：在功能体验场景的单节点部署中，可直接输入节点 IP，无需启用 `自建 VIP` 或准备 `global VIP` 等网络资源。
平台访问地址	如果不需要区分 集群访问端点 和 平台访问地址，请输入与 集群访问端点 相同的地址。 若需区分，例如 `global` 集群仅供内网访问，平台需提供外网访问，则输入预先准备好的域名或 `外部 IP`。 平台默认使用 HTTPS 访问，不启用 HTTP。如需启用 HTTP 访问，可在 高级设置 中开启（不推荐）。 注意：以下情况必须输入域名， • 计划为 `global` 集群做灾备； • 平台需支持 IPv6 访问。

	提示：如需配置更多平台访问地址，可在下一步的 其他设置 > 其他平台访问地址 中添加，或安装后根据用户手册在平台管理中添加。
证书	平台默认提供自签名证书以支持 HTTPS 访问。 如需使用自定义证书，可上传已有证书。
镜像仓库	默认使用 `平台部署` 镜像仓库，包含所有组件镜像。 如需使用 `外部` 镜像仓库，请先联系技术支持获取镜像同步方案后再配置。
容器网络	集群默认子网与 Service 网络段不可重叠。 使用 Kube-OVN Overlay 网络时，确保容器网络与主机网络不在同一网段，否则可能导致网络异常。
节点名称	若选择 `主机名作为节点名` ，请确保所有节点主机名唯一。
`global` 集群平台节点隔离	仅在计划在 `global` 集群运行应用工作负载时启用。 启用后： • 节点可设置为 `平台专属` ，即仅运行平台组件，确保平台与应用工作负载隔离； • 排除 DaemonSet 类型的工作负载。
添加节点	控制平面节点： • 支持添加 1 个或 3 个控制平面节点（3 个用于高可用配置）； • 若启用 `平台专属` ，则强制禁用 `可部署应用` ，控制平面节点仅运行平台组件；

- 若未启用 `平台专属`，可选择是否启用 `可部署应用`，允许控制平面节点运行应用工作负载。

工作节点：

- 若启用 `平台专属`，强制禁用 `可部署应用`；
- 若未启用 `平台专属`，强制启用 `可部署应用`。

使用 Kube-OVN 时，可通过输入网关名称指定节点网卡。

若节点可用性检查失败，请根据页面提示调整后重新添加。

安装程序清理

通常安装完成后，安装程序会自动删除。如果安装完成 30 分钟后安装程序未自动删除，请在安装程序所在节点执行以下命令强制删除安装程序容器：

```
nerdctl rm -f minialauda-control-plane
```

其他资源

- 上传并安装扩展

global 集群灾难恢复

目录

Overview

该方案针对 `global` 集群的灾难恢复场景设计。`global` 集群作为平台的控制平面，负责管理其他集群。为确保在 `global` 集群故障时平台服务的持续可用性，本方案部署两个 `global` 集群：主集群和备用集群。

灾难恢复机制基于主集群到备用集群的 etcd 数据实时同步。当主集群因故障不可用时，服务可快速切换至备用集群。

支持的灾难场景

- 主集群发生不可恢复的系统级故障，导致无法正常运行；
- 托管主集群的物理机或虚拟机故障，导致无法访问；
- 主集群所在位置网络故障，导致服务中断；

不支持的灾难场景

- `global` 集群内部部署的应用故障；
- 存储系统故障导致的数据丢失（etcd 同步范围之外）；

主集群和备用集群的角色是相对的：当前为平台提供服务的集群为主集群（DNS 指向该集群），备用集群为备用集群。故障切换后，角色互换。

注意事项

- 本方案仅同步 `global` 集群的 etcd 数据；不包含 registry、chartmuseum 或其他组件的数据；
- 为便于排查和管理，建议节点命名采用如 `standby-global-m1` 的风格，以区分节点所属集群（主集群或备用集群）。
- 不支持集群内应用数据的灾难恢复；
- 两个集群间需保持稳定的网络连接，以确保 etcd 同步的可靠性；
- 若集群基于异构架构（如 x86 与 ARM），请使用双架构安装包；
- 以下命名空间不参与 etcd 同步，若在这些命名空间创建资源，需用户自行备份：

```
cpaas-system
cert-manager
default
global-credentials
cpaas-system-global-credentials
kube-ovn
kube-public
kube-system
nsx-system
cpaas-solution
kube-node-lease
kubevirt
nativestor-system
operators
```

- 若两个集群均使用内置镜像仓库，容器镜像需分别上传至各集群；
- 若主集群部署了 灵雀云 **DevOps Eventing v3**（knative-operator）及其实例，备用集群需预先部署相同组件。

流程概述

1. 准备统一的域名作为平台访问地址；
2. 将域名指向主集群的 VIP 并安装主集群；
3. 临时将 DNS 解析切换至备用 VIP，安装备用集群；
4. 将主集群的 ETCD 加密密钥复制到备用集群将作为控制平面的节点；
5. 安装并启用 etcd 同步插件；
6. 验证同步状态并定期检查；
7. 发生故障时，将 DNS 切换至备用集群完成灾难恢复。

所需资源

- 统一域名作为 `Platform Access Address`，以及该域名的 TLS 证书和私钥，用于 HTTPS 服务；
- 每个集群专用的虚拟 IP 地址——一个用于主集群，另一个用于备用集群；

- 预先配置负载均衡器，将端口 `80`、`443`、`6443`、`2379` 和 `11443` 的 TCP 流量转发至对应 VIP 后端的控制平面节点。

操作步骤

第 1 步：安装主集群

> **灾难恢复环境安装注意事项**
>
> 安装灾难恢复环境的主集群时，
>
> - 首先，记录安装 Web UI 指南中设置的所有参数。安装备用集群时需保持部分选项一致。
> - 必须预先配置 **User-provisioned** 负载均衡器，将流量路由至虚拟 IP。`Self-built VIP` 选项不可用。
> - `Platform Access Address` 字段必须为域名，`Cluster Endpoint` 必须为虚拟 IP 地址。
> - 两个集群均须配置使用 `An Existing Certificate`（且证书相同），必要时申请合法证书。`Self-signed Certificate` 选项不可用。
> - 当 `Image Repository` 设置为 `Platform Deployment`，`Username` 和 `Password` 字段均不能为空；`IP/Domain` 字段必须设置为作为 `Platform Access Address` 的域名。
> - `Platform Access Address` 的 `HTTP Port` 和 `HTTPS Port` 字段必须分别为 80 和 443。
> - 安装指南第二页（步骤：`Advanced`）中，`Other Platform Access Addresses` 字段必须包含当前集群的虚拟 IP。

请参考以下文档完成安装：

- 安装准备
- 安装指南

第 2 步：安装备用集群

1. 临时将域名指向备用集群的 VIP；

2. 登录主集群的第一个控制平面节点，将 etcd 加密配置复制到备用集群所有控制平面节点：

```
# 假设主集群控制平面节点为 1.1.1.1、2.2.2.2 和 3.3.3.3
# 备用集群控制平面节点为 4.4.4.4、5.5.5.5 和 6.6.6.6
for i in 4.4.4.4 5.5.5.5 6.6.6.6  # 替换为备用集群控制平面节点 IP
do
  ssh "<user>@$i" "sudo mkdir -p /etc/kubernetes/"
  scp /etc/kubernetes/encryption-provider.conf "<user>@$i:/tmp/encryption-provider.conf"
  ssh "<user>@$i" "sudo install -o root -g root -m 600 /tmp/encryption-provider.conf /etc/kubernetes/encryption-provider.conf && rm -f /tmp/encryption-provider.conf"
done
```

3. 按照主集群安装方式安装备用集群

> **安装备用集群注意事项**
>
> 安装灾难恢复环境的备用集群时， 以下选项必须与主集群保持一致：
>
> - `Platform Access Address` 字段；
> - `Certificate` 的所有字段；
> - `Image Repository` 的所有字段；
> - 重要：确保镜像仓库凭据和 ACP 管理员用户与主集群设置一致。
>
> 并且务必遵循第 1 步中的 `灾难恢复环境安装注意事项`。

请参考以下文档完成安装：

- 安装准备
- 安装指南

第 3 步：启用 **etcd** 同步

1. 如适用，配置负载均衡器将端口 `2379` 转发至对应集群的控制平面节点。仅支持 TCP 模式，不支持 L7 转发。

> **INFO**

通过负载均衡器转发端口非必需。 若备用集群可直接访问活动 global 集群，需通过 **Active Global Cluster ETCD Endpoints** 指定 etcd 地址。

2. 使用备用 global 集群的 VIP 访问 Web 控制台，切换至 **Administrator** 视图；
3. 进入 **Marketplace > Cluster Plugins**，选择 `global` 集群；
4. 找到 灵雀云容器平台 **etcd Synchronizer**，点击 **Install**，配置参数：
   - 若未通过负载均衡器转发端口 `2379`，需正确配置 **Active Global Cluster ETCD Endpoints**；
   - 使用默认的 **Data Check Interval**；
   - 除非排查问题，否则保持 **Print detail logs** 关闭。

验证同步 Pod 是否在备用集群运行：

```
kubectl get po -n cpaas-system -l app=etcd-sync
kubectl logs -n cpaas-system $(kubectl get po -n cpaas-system -l app=etcd
-sync --no-headers | head -1) | grep -i "Start Sync update"
```

当出现 “Start Sync update” 后，重建其中一个 Pod 以重新触发带有 ownerReference 依赖的资源同步：

```
kubectl delete po -n cpaas-system $(kubectl get po -n cpaas-system -l app
=etcd-sync --no-headers | head -1)
```

检查同步状态：

```
mirror_svc=$(kubectl get svc -n cpaas-system etcd-sync-monitor -o jsonpat
h='{.spec.clusterIP}')
ipv6_regex="^[0-9a-fA-F:]+$"
if [[ $mirror_svc =~ $ipv6_regex ]]; then
  export mirror_new_svc="[$mirror_svc]"
else
  export mirror_new_svc=$mirror_svc
fi
curl $mirror_new_svc/check
```

输出说明：

- `LOCAL ETCD missed keys`：主集群存在但备用集群缺失的键，通常因同步时资源顺序导致 GC。重启一个 etcd-sync Pod 可修复；
- `LOCAL ETCD surplus keys`：备用集群独有的多余键，删除前请与运维确认。

若安装了以下组件，重启其服务：

- 灵雀云容器平台 Elasticsearch 日志存储：

```
kubectl delete po -n cpaas-system -l service_name=cpaas-elasticsearch
```

- 灵雀云容器平台 VictoriaMetrics 监控：

```
kubectl delete po -n cpaas-system -l 'service_name in (alertmanager,vmselect,vminsert)'
```

灾难恢复流程

1. 如有必要，重启备用集群的 Elasticsearch：

```
# 将 installer/res/packaged-scripts/for-upgrade/ensure-asm-template.sh
复制到 /root：
# 请勿跳过此步骤

# 如有需要，切换至 root 用户
sudo -i

# 检查 global 集群是否安装 Elasticsearch 日志存储
_es_pods=$(kubectl get po -n cpaas-system | grep cpaas-elasticsearch |
awk '{print $1}')
if [[ -n "${_es_pods}" ]]; then
    # 若脚本返回 401 错误，重启 Elasticsearch
    # 然后执行脚本再次检查集群
    bash /root/ensure-asm-template.sh

    # 重启 Elasticsearch
    xargs -r -t -- kubectl delete po -n cpaas-system <<< "${_es_pods}"
fi
```

2. 验证备用集群数据一致性（同 第 3 步 检查）；
3. 卸载 etcd 同步插件；
4. 取消两个 VIP 的端口 `2379` 转发；
5. 将平台域名 DNS 切换至备用 VIP，备用集群成为新的主集群；
6. 验证 DNS 解析：

```
kubectl exec -it -n cpaas-system deployments/sentry -- nslookup <platfo
rm access domain>
# 若解析失败，重启 coredns Pod 并重试，直至成功
```

7. 清理浏览器缓存，访问平台页面确认显示为原备用集群内容；
8. 重启以下服务（如已安装）：

- 灵雀云容器平台 Elasticsearch 日志存储：

```
kubectl delete po -n cpaas-system -l service_name=cpaas-elasticsearch
```

- 灵雀云容器平台 VictoriaMetrics 监控：

```
kubectl delete po -n cpaas-system -l 'service_name in (alertmanager,v
mselect,vminsert)'
```

- cluster-transformer：

```
kubectl delete po -n cpaas-system -l service_name=cluster-transformer
```

9. 若业务集群向主集群发送监控数据，重启业务集群中的 warlock：

```
kubectl delete po -n cpaas-system -l service_name=warlock
```

10. 在原主集群上重复执行 启用 etcd 同步 步骤，将其转为新的备用集群。

日常检查

定期检查备用集群的同步状态：

```
curl $(kubectl get svc -n cpaas-system etcd-sync-monitor -o jsonpath='{.s
pec.clusterIP}')/check
```

若发现缺失或多余键，按照输出提示进行处理。

上架软件包

有关 `violet push` 子命令的详细信息，请参见 Upload Packages。