Menu

Установка

В этом документе представлена вся информация, касающаяся установки ACP .

Обзор

Обзор

Подготовка к установке

Требования

Загрузка

Предварительная обработка узлов

Установка

Установка

Восстановление после катастрофы глобального кластера

Восстановление после катастрофы глобального кластера

Обзор

Этот документ предназначен для системных администраторов с базовыми знаниями Linux и подробно описывает полный процесс установки `global` кластера ACP .

В данном документе часто будут использоваться следующие термины, обратите внимание на их различия:

- `global` **кластер**: Относится к кластеру, отвечающему за основные задачи управления платформой, такие как управление кластером и управление арендаторами. Завершение установки `global` кластера означает завершение установки ACP . После установки можно по необходимости получить доступ к рабочим кластерам или создать их.
- **Платформа**: Обычно означает саму контейнерную платформу — полноценную систему, которая предоставляет среду выполнения приложений и интегрирует функции планирования контейнеров, управления, сетей, хранения и другие. **Платформа** — это общее понятие, охватывающее все компоненты, включая `global` кластер и рабочие кластеры.

> **INFO**
>
> Перед установкой убедитесь, что вы выполнили планирование ресурсов, предварительную подготовку окружения и проверку предварительных условий, чтобы гарантировать соответствие аппаратного обеспечения, сети и ОС каждого узла требованиям. В следующем разделе рассматриваются архитектурные решения платформы, методы установки и объяснение ключевых терминов, что поможет вам лучше понять основные моменты в процессе фактической установки.

Содержание

Метод установки

Объяснение терминов

Метод установки

Процесс установки `global` кластера в основном делится на три этапа:

1. **Этап подготовки**

   - **Проверка предварительных условий**: Убедитесь, что все узлы соответствуют требованиям по аппаратному обеспечению, сети и ОС, таким как версия ядра, архитектура CPU и конфигурация сети.
   - **Загрузка и проверка установочного пакета**: Войдите в Customer Portal для получения последнего установочного пакета и файла подписи, используйте инструмент GPG для проверки целостности и корректности пакета.
   - **Предварительная обработка узлов**: Выполните подготовительные работы на всех узлах.

2. **Этап выполнения**

   - **Загрузка и распаковка установочного пакета**: Загрузите установочный пакет на целевой узел управляющей плоскости (рекомендуемый каталог: `/root/cpaas-install`) и распакуйте установочные ресурсы.
   - **Запуск установщика**: Выполните скрипт установки (например, `bash setup.sh`) на узле управляющей плоскости, выберите сетевой плагин (Kube-OVN или Calico), режим IP-протокола (IPv4/IPv6/dual stack) и настройку VIP в соответствии с реальной средой.
   - **Конфигурация параметров**: Получите доступ к Web UI, предоставляемому установщиком, и последовательно задайте версию Kubernetes, сеть кластера, имя узла, адрес доступа и другие ключевые параметры для завершения установки `global` кластера.

3. **Этап проверки**

- **Проверка состояния системы**: После завершения установки войдите в Web UI платформы, чтобы проверить состояние кластера и работу каждого компонента.
- **Проверка через CLI**: Используйте инструменты командной строки для проверки состояния ресурсов кластера, чтобы убедиться, что все сервисы работают нормально и отсутствуют ошибки или сбои.

Объяснение терминов

- **VIP (Virtual IP)** Виртуальный IP-адрес, используемый для балансировки нагрузки на API Server, Web UI и другие точки доступа `global` кластера с целью обеспечения высокой доступности.

В последующих главах будет подробно описана последовательность операций, параметры конфигурации и методы проверки каждого этапа установки. Пожалуйста, внимательно ознакомьтесь с ними и выполните соответствующую подготовительную работу перед официальной установкой.

Подготовка к установке

Требования

Загрузка

Предварительная обработка узлов

Требования

Перед установкой кластера `global` необходимо подготовить оборудование, сеть и ОС, соответствующие требованиям.

INFO

1. В настоящее время платформа не поддерживает прямую установку кластера `global` в существующую среду Kubernetes. Если в вашей среде уже есть кластер Kubernetes, пожалуйста, сделайте резервную копию данных и очистите среду перед установкой.

2. Если вы планируете использовать глобальное аварийное восстановление кластера (global Cluster Disaster Recovery), сначала ознакомьтесь с разделом global Cluster Disaster Recovery.

Содержание

Планирование ёмкости

Машины

- Базовые требования
- Требования к архитектуре ARM

Поддерживаемые ОС и ядра

Сеть

- Сетевые ресурсы
- Конфигурация сети
- Правила переадресации LoadBalancer

Планирование ёмкости

Перед установкой необходимо выбрать подходящий сценарий установки, исходя из ваших целей и реальных потребностей. Разные сценарии существенно отличаются по конфигурации ресурсов инфраструктуры и требованиям к архитектуре. Ниже приведены рекомендации по планированию для трёх типичных сценариев:

Single Node

Область применения

Подходит для проверки функций платформы, демонстраций или тестирования технической реализуемости. Этот сценарий используется только для проверки основных функций платформы и не предназначен для работы с производственным уровнем нагрузки. Конфигурация ресурсов минимальна.

Требования к конфигурации ресурсов

Параметр	Требования спецификации
Количество узлов	1 (физическая или виртуальная машина)
CPU	≥16 ядер
Оперативная память	≥32ГБ

Описание архитектуры

- **All-in-one**: кластер состоит из одного узла, на котором работают все компоненты управляющей плоскости и приложения.
- **Лёгкая нагрузка**: может запускать только демонстрационные приложения с количеством Pod не более 10.
- **Не для производства**: не поддерживает горизонтальное масштабирование и не обеспечивает непрерывность работы приложений и высокую доступность.

Single Cluster

Область применения

Для стандартизированной поставки ISV (независимыми поставщиками программного обеспечения), кластер `global` одновременно управляет платформой и напрямую запускает приложения ISV без необходимости создавать дополнительные кластеры для рабочих нагрузок.

Требования к конфигурации ресурсов

План развертывания	Тип узла	Количество	Минимальные характ
Минимальная конфигурация	Узел управляющей плоскости	3	8 ядер 16ГБ + требова
	Узел рабочих нагрузок	0	—
Рекомендуемая конфигурация	Узел управляющей плоскости	3	8 ядер 16ГБ
	Узел рабочих нагрузок	≥2	В соответствии с требо нагрузки приложений I

Описание архитектуры

- **Готов к производству**: кластер `global` одновременно запускает компоненты управляющей плоскости и приложения ISV.
- **Изоляция платформы и приложений**: рекомендуется запускать приложения ISV на выделенных рабочих узлах, чтобы избежать конкуренции за ресурсы с управляющей плоскостью.
- **Масштабирование**: на каждые 50 новых Pod приложений добавляется 1 рабочий узел в соответствии с требованиями ресурсов приложений ISV.

Multi-Cluster

Область применения

Подходит для крупных корпоративных дата-центров, требующих единого управления множеством кластеров рабочих нагрузок в разных облаках и регионах. С увеличением количества кластеров рабочих нагрузок кластер `global` должен динамически масштабировать рабочие узлы и конфигурацию ресурсов для обеспечения высокой доступности и производительности.

Основные функции

- **Гибридное управление**: поддерживает единое управление кластерами рабочих нагрузок в разных облаках и регионах.
- **Эластичное масштабирование**: ресурсы кластера `global` масштабируются линейно с ростом числа подключаемых кластеров рабочих нагрузок.
- **Физическая изоляция**: рекомендуется физически изолировать управляющую и вычислительную плоскости для обеспечения стабильности системы.

Базовая конфигурация ресурсов

Тип узла	Количество	Минимальные характеристики	Рекомендуе
Узел управляющей плоскости	3 или 5	8 ядер 16ГБ	16 ядер 32ГБ
Узел рабочих нагрузок	≥2	12 ядер 24ГБ	24 ядра 48ГБ

Тип узла	Количество	Минимальные характеристики	Рекомендуе

Правила динамического масштабирования

2.1. **Вертикальное масштабирование**: отдельные узлы могут быть поэтапно обновлены (например, 12 ядер 24ГБ → 24 ядра 48ГБ → 48 ядер 96ГБ).

2.2. **Горизонтальное масштабирование**: на каждые 10 подключаемых кластеров рабочих нагрузок рекомендуется увеличить вычислительные ресурсы на 20% (увеличить количество узлов или обновить характеристики узлов).

2.3. **Триггер мониторинга**: при постоянной загрузке CPU/памяти выше 70% необходимо инициировать меры масштабирования.

Описание архитектуры

- Управляющая плоскость состоит из 3 узлов, формирующих кластер ETCD. Рекомендуется включить шифрование TLS и периодические снимки.
- Критичные компоненты платформы рекомендуется развертывать на отдельных рабочих узлах, чтобы избежать конкуренции за ресурсы.
- Рекомендация по аварийному восстановлению: развертывать в нескольких зонах доступности, чтобы рабочие узлы распределялись минимум по 2 физическим доменам отказа.

TIP

1. **Резервирование ресурсов**: в производственной среде рекомендуется резервировать не менее 30% ресурсов для обработки внезапных нагрузок.

2. **Планирование сети**: кластер `global` должен быть развернут в отдельном VPC или VLAN с пропускной способностью ≥1 Гбит/с.

3. **Изоляция хранилища**: для хранения ETCD рекомендуется использовать NVMe SSD, физически изолированный от хранилища приложений.

Машины

INFO

В этом разделе описаны минимальные требования к оборудованию для построения высокодоступного кластера `global`. Если вы завершили планирование ёмкости, подготовьте соответствующие ресурсы согласно разделу Capacity Planning или масштабируйте их по необходимости после установки.

Базовые требования

Для кластера необходимо предоставить не менее **3** физических или виртуальных машин в качестве узлов управляющей плоскости. Минимальная конфигурация для каждого узла следующая:

Категория	Минимальные требования
CPU	≥ 8 ядер, тактовая частота ≥ 2.5 ГГц Без оверпровиженинга; отключить энергосбережение
Память	≥ 16 ГБ Без оверпровиженинга; рекомендуется использовать минимум шестиканальную DDR4
Жёсткий диск	IOPS одного устройства ≥ 2000 Пропускная способность ≥ 200 МБ/с Обязательно использование SSD

Требования к архитектуре ARM

Для архитектур ARM (например, Kunpeng 920) рекомендуется увеличить конфигурацию в **2 раза** по сравнению с минимальной конфигурацией x86, но не менее чем в **1.5 раза**.

Например: если для x86 требуется 8 ядер 16ГБ, то для ARM минимум 12 ядер 24ГБ, а рекомендуемая конфигурация — 16 ядер 32ГБ.

Поддерживаемые ОС и ядра

INFO

1. **Требования к версиям ядра:**

   Эти версии ядра официально выпущены и проверены нашими тестами платформы. В вашем реальном развертывании важно соблюдать **основные номера версий A.B.C**, тогда как последующие минорные версии могут отличаться.

2. **Неподдерживаемые среды:**

   Если ОС, версия ядра или архитектура CPU не соответствуют требованиям, пожалуйста, обратитесь в техническую поддержку.

Red Hat Enterprise Linux (RHEL)

- RHEL 7.8: `3.10.0-1127.el7.x86_64`
- RHEL 8.0 до 8.6: `4.18.0-80.el8.x86_64` до `4.18.0-372.9.1.el8.x86_64`

WARNING

RHEL 7.8 не поддерживает **Calico Vxlan IPv6**.

CentOS

- CentOS 7.6 до 7.9: `3.10.0-1127` до `3.11`

WARNING

CentOS не поддерживает **Calico Vxlan IPv6**.

Ubuntu

- Ubuntu 20.04 LTS: `5.4.0-124-generic`
- Ubuntu 22.04 LTS: `5.15.0-56-generic`

WARNING

Версии Ubuntu HWE (Hardware Enablement) не поддерживаются.

Kylin Linux Advanced Server

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

WARNING

- В версиях Kylin V10, V10-SP1 и V10-SP2 известны проблемы с ядром, которые могут вызывать **сбои сетевого доступа NodePort**; рекомендуется обновиться до **Kylin V10-SP3**.
- Архитектура ARM поддерживает только `Kunpeng 920`. Для других моделей обращайтесь в техническую поддержку.

Сеть

Перед установкой необходимо предварительно настроить следующие сетевые ресурсы. Если аппаратный LoadBalancer предоставить невозможно, установщик поддерживает настройку **haproxy + keepalived** в качестве программного балансировщика нагрузки, однако следует учитывать:

- **Низкая производительность**: программный балансировщик уступает по производительности аппаратному LoadBalancer.
- **Повышенная сложность**: незнание keepalived может привести к недоступности кластера `global`, длительному устранению проблем и серьёзному снижению надёжности платформы.

Сетевые ресурсы

Ресурс	Обязательность	Количество	Описание
`global` VIP	Обязательно	1	Используется для доступа узлов к kube-apiserver, настраивается в устройстве балансировки нагрузки для обеспечения высокой доступности. Этот IP также может использоваться как адрес доступа к Web UI платформы. Кластеры рабочих нагрузок в той же сети, что и `global`, могут обращаться к кластеру `global` через этот IP.
Внешний IP	Опционально	По требованию	Если есть кластеры рабочих нагрузок вне сети `global` (например, в гибридном облаке), необходимо предоставить этот IP. Кластеры в других сетях обращаются к `global` через этот IP. Должен быть настроен в устройстве балансировки нагрузки для обеспечения высокой доступности. Также может использоваться как адрес доступа к Web UI платформы.
Доменное имя	Опционально	По требованию	Если требуется доступ к кластеру `global` или Web UI платформы по доменному

Ресурс	Обязательность	Количество	Описание
			имени, предоставьте его заранее и убедитесь в корректности разрешения домена.
Сертификат	Опционально	По требованию	Рекомендуется использовать доверенный сертификат для избежания предупреждений безопасности браузера; если не предоставлен, установщик сгенерирует самоподписанный сертификат, что может нести риски при использовании HTTPS.

INFO

Доменное имя обязательно в следующих случаях:

1. Кластер `global` должен поддерживать доступ по IPv6.
2. Планируется аварийное восстановление кластера `global`.

NOTE

Если платформе необходимо настроить несколько адресов доступа (например, для внутренней и внешней сети), подготовьте соответствующие IP-адреса или доменные имена заранее согласно таблице выше. Их можно будет указать в параметрах установки или добавить после установки согласно документации продукта.

Конфигурация сети

Тип	Описание требования
Скорость сети	Скорость между кластером `global` и кластерами рабочих нагрузок в одной сети ≥1 Гбит/с (рекомендуется 10 Гбит/с); между сетями ≥100 Мбит/с (рекомендуется 1 Гбит/с). Недостаточная скорость значительно снижает производительность запросов данных.
Задержка сети	Задержка ≤2 мс в одной сети; задержка ≤100 мс (рекомендуется ≤30 мс) между сетями.
Сетевая политика	См. LoadBalancer Forwarding Rules для обеспечения открытия необходимых портов; при использовании Calico CNI необходимо включить протокол **IP-in-IP**.
Диапазон IP-адресов	Узлы кластера `global` должны избегать использования сетевого сегмента 172.16-32. Если он используется, необходимо скорректировать конфигурацию Docker (добавить параметр bip) для предотвращения конфликтов.

Правила переадресации LoadBalancer

Эти правила обеспечивают нормальный приём трафика кластером `global` от LoadBalancer. Проверьте сетевую политику согласно таблице ниже, чтобы убедиться, что соответствующие порты открыты.

Исходный IP	Протокол	IP назначения	Порт назначения	Описани
`global` VIP, External IP	TCP	Все IP узлов управляющей плоскости	443	Обеспечивае доступ к Web платформы, репозиторию образов и Kubernetes A Server через протокол HT По умолчани порт `443`. Ec

Исходный IP	Протокол	IP назначения	Порт назначения	Описани
				требуется использовать нестандартн HTTPS-порт, выполните: • Замените назначени правиле переадрес на ваш по • В парамет установки UI укажите порт.
`global` VIP, External IP	TCP	Все IP узлов управляющей плоскости	6443	Порт для дос к Kubernetes Server изнутр кластера.
`global` VIP, External IP	TCP	Все IP узлов управляющей плоскости	11443	Порт для дос к репозитори образов изну кластера. **Примечание** если планиру использовать внешний репозиторий образов вме стандартного предоставля кластером

Исходный IP	Протокол	IP назначения	Порт назначения	Описани
				`global`, этот настраивать нужно.

TIP

- Рекомендуется настроить проверку состояния (health check) на LoadBalancer для мониторинга статуса портов.
- При планировании аварийного восстановления кластера `global` необходимо открыть порт `2379` для всех узлов управляющей плоскости для синхронизации данных ETCD между основным и аварийным кластерами.
- Платформа по умолчанию поддерживает только HTTPS. Если требуется поддержка HTTP, необходимо открыть HTTP-порт для всех узлов управляющей плоскости.

Загрузка установочного пакета

Перед установкой необходимо скачать и распаковать установочные пакеты. Если вам требуется Extensions Package, сначала распакуйте Core Package, затем Extensions Package в одну и ту же директорию перед началом установки.

Загрузка установочных пакетов

Войдите в **Customer Portal** для загрузки установочных пакетов. Если у вас нет зарегистрированного аккаунта, обратитесь в техническую поддержку.

Доступны два типа установочных пакетов:

- **Core Package**: требуется для всех установок
- **Extensions Package**: опционально, скачивайте только при необходимости дополнительного функционала

Документация по Core Package

Скачайте core package installer-core-v4.x.y-zzzz.tar. Здесь "zzzz" может быть x86, arm или hybrid, в зависимости от выбранной пользователем архитектуры (то же самое применимо ниже). Распакуйте core package с помощью команды tar:

```
tar xvf installer-core-v4.x.y-zzzz.tar
```

Документация по Extensions Package

> Скачайте extensions package installer-extensions-v4.x.y-zzzz.tar. Здесь "zzzz" должен совпадать с core package. Extensions package содержит плагины, которые не включены в core package. Если вы скачали Extensions Package, необходимо распаковать extensions package в ту же директорию перед установкой:
>
> ```
> tar xvf installer-extensions-v4.x.y-zzzz.tar -C <директория распаковки core package>
> ```
>
> Пример: Если core package был распакован в /root, после распаковки там появится директория installer. Следует выполнить следующую команду в директории, где находится installer-extensions-v4.x.y-zzzz.tar: `tar xvf installer-extensions-v4.x.y-zzzz.tar -C /root/`

В руководстве по установке будет подробно описан процесс установки Core Package. Документация по Extensions Package предоставит конкретные инструкции по использованию и настройке расширений.

Предварительная обработка узлов

Перед установкой кластера `global` все узлы (узлы управляющей плоскости и рабочие узлы) должны пройти предварительную обработку.

Содержание

Выполнение скрипта быстрой настройки

Проверки узлов

Приложение

- Удаление конфликтующих пакетов
- Настройка поискового домена

Выполнение скрипта быстрой настройки

В установочном пакете ACP предоставлен скрипт для быстрой настройки узлов.

Распакуйте установочный пакет, чтобы получить файл скрипта `init.sh` в каталоге `res`. Скопируйте файл скрипта на узлы и убедитесь, что у вас есть права `root`.

Выполните скрипт:

```
bash init.sh
```

ВНИМАНИЕ

Скрипт `init.sh` не гарантирует корректную обработку всех перечисленных ниже проверок.

Вам всё равно необходимо продолжить выполнение следующих шагов.

Проверки узлов

Ниже приведён список всех проверок, которые должны быть выполнены на узлах. В зависимости от роли узла требуемые проверки могут отличаться. Например, некоторые проверки применимы только к узлам управляющей плоскости.

Проверки разделены на две категории:

- ✅ Обозначает проверку, которая должна быть успешно пройдена.
- ⚠️ Обозначает проверку, которая должна быть выполнена в определённых сценариях. Пожалуйста, определите, выполнены ли соответствующие условия согласно инструкциям. Если да, необходимо их устранить.

Список проверок:

- **ОС и ядро**
  - ✅ В конфигурации загрузчика grub машины должен быть параметр `transparent_hugepage=never`.
  - ✅ В конфигурации загрузчика grub системы CentOS 7.x должен быть параметр `cgroup.memory=nokmem`.
  - ✅ Проверьте, что модули ядра `ip_vs`, `ip_vs_rr`, `ip_vs_wrr` и `ip_vs_sh` загружены.
  - ⚠️ Если версия ядра ниже 4.19.0 (или RHEL ниже 4.18.0), проверьте, что модули ядра `nf_conntrack_ipv4` и (для IPv6) `nf_conntrack_ipv6` загружены.
  - ⚠️ Если в кластере `global` планируется использовать CNI `Kube-OVN`, модули ядра `geneve` и `openvswitch` должны быть загружены.
  - ✅ Отключите apparmor/selinux и брандмауэр.
  - ✅ Отключите `swap`.
- **Пользователи и права**

- ✅ SSH-пользователь узла имеет права `root` и может использовать `sudo` без пароля.
- ✅ Параметры `UseDNS` и `UsePAM` в файле `/etc/ssh/sshd_config` должны быть установлены в `no`.
- ✅ Выполнение `systemctl show --property=DefaultTasksMax` должно возвращать `infinity` или очень большое значение; в противном случае отредактируйте `/etc/systemd/system.conf`.

- **Сеть узла**
  - ✅ `hostname` должен соответствовать следующим правилам:
    - Не более 36 символов.
    - Начинается и заканчивается буквой или цифрой.
    - Содержит только строчные буквы, цифры, `-` и `.`, но не содержит последовательностей `.-`, `..` или `-.`.
  - ✅ В файле `/etc/hosts` `localhost` должен разрешаться в `127.0.0.1`.
  - ✅ Файл `/etc/resolv.conf` должен существовать и содержать конфигурации `nameserver`, но не должен содержать адреса, начинающиеся с 172 (отключите systemd-resolved).
  - ⚠️ В файле `/etc/resolv.conf` не должно быть настроек поисковых доменов (если необходимо настроить, см. Настройка поискового домена).
  - ✅ IP-адрес машины не должен быть петлевым, многоадресным, локальным по ссылке, all-0 или широковещательным.
  - ✅ Выполнение `ip route` должно возвращать маршрут по умолчанию или маршрут, указывающий на `0.0.0.0`.
  - ✅ Узлы не должны занимать следующие порты:
    - **Узлы управляющей плоскости:** `2379`, `2380`, `6443`, `10249` ~ `10256`
    - **Узел, на котором находится установщик:** `8080`, `12080`, `12443`, `16443`, `2379`, `2380`, `6443`, `10249` ~ `10256`
    - **Рабочие узлы:** `10249` ~ `10256`
  - ✅ Если в кластере `global` используется **Kube-OVN** или **Calico**, убедитесь, что следующие порты не заняты:

    - **Kube-OVN:** `6641`, `6642`
    - **Calico:** `179`
  - ⚠️ Убедитесь, что IP-адреса в сетевом сегменте `172.16.x.x` ~ `172.32.x.x`, необходимых Docker, не заняты. Если IP-адреса в этом сегменте заняты и их нельзя изменить, обратитесь в техническую поддержку.
- **Требования к программному обеспечению и каталогам:**
  - ✅ Должны быть установлены следующие утилиты: `ip`, `ss`, `tar`, `swapoff`, `modprobe`, `sysctl`, `md5sum`, а также `scp` или `sftp`.
  - ⚠️ Если планируется использовать локальное хранилище **TopoLVM** или **Rook**, необходимо установить `lvm2`.
  - ✅ Файл `/etc/systemd/system/kubelet.service` не должен существовать.
  - ✅ Параметры монтирования `/tmp` не должны содержать `noexec`.
  - ✅ Удалите пакеты, конфликтующие с компонентами кластера `global` (см. Удаление конфликтующих пакетов).
  - ✅ Следующие каталоги должны быть удалены, если они существуют:
    - `/var/lib/docker`
    - `/var/lib/containerd`
    - `/var/log/pods`
    - `/var/lib/kubelet/pki`
- **Проверки между узлами**
  - ✅ Между узлами кластера `global` не должно быть ограничений сетевого брандмауэра.
  - ✅ `hostname` каждого узла в кластере должен быть уникальным.
  - ✅ Все узлы должны иметь одинаковый часовой пояс, а ошибка синхронизации времени должна быть ≤ 10 секунд.

Приложение

Удаление конфликтующих пакетов

Перед установкой на узлах могут уже работать приложения в средах docker/containerd или может быть установлено программное обеспечение, конфликтующее с кластером `global`. Поэтому необходимо проверить и удалить конфликтующие пакеты.

ОПАСНОСТЬ

- Чтобы избежать прерывания работы приложений или потери данных, обязательно убедитесь в наличии конфликтующих программных пакетов. При обнаружении конфликта разработайте план переключения приложений и сделайте резервную копию данных перед удалением.
- После удаления конфликтующих пакетов необходимо проверить наличие других потенциально конфликтующих бинарных файлов в каталогах, таких как `/usr/local/bin/` (например, программное обеспечение, связанное с docker, containerd, runc, podman, сетями контейнеров, средами выполнения контейнеров или Kubernetes).

Ниже приведены команды для справки.

CentOS / RedHat

Проверка:

```
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
```

Удаление:

```
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

Проверка:

```
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-apiserver \
    k8s microk8s \
    kubeadm kubelet \
    ; do
    snap list | grep -F "$x"
done
```

Удаление:

```
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-apiserver \
    k8s microk8s \
    kubeadm kubelet \
    ; do
    snap remove --purge "$x"
done
```

Kylin

Проверка:

```
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 kubernetes-kubeadm \
    ; do
    rpm -qa | grep -F "$x"
done
```

Удаление:

```
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 kubernetes-kubeadm \
    ; do
    yum remove "$x"
done
```

Настройка поискового домена

В Linux OS файл `/etc/resolv.conf` используется для настройки параметров разрешения доменных имён клиента DNS. Строка `search` задаёт путь поиска доменов для DNS-запросов.

Требования к конфигурации

- **Количество доменов:** Количество доменов в строке `search` должно быть меньше `domainCountLimit - 3` (по умолчанию `domainCountLimit` равно 32).
- **Длина одного домена:** Каждый домен не должен превышать 253 символа.
- **Общая длина:** Общая длина всех доменных имён и пробелов не должна превышать `MaxDNSSearchListChar` (по умолчанию 2048).

Пример

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

- Общее количество доменов — 3.
- Длина одного домена, например `domain1.com`, — 11.
- Общая длина — 35, то есть 11 + 11 + 11 + 2 (два пробела).

ВНИМАНИЕ

- Если строка `search` в файле `/etc/resolv.conf` не соответствует указанным ограничениям, это может привести к сбоям DNS-запросов или снижению производительности.
- Перед изменением файла `/etc/resolv.conf` рекомендуется сделать его резервную копию.

Установка

В этом разделе описаны конкретные шаги по установке кластера `global`.

Перед началом установки убедитесь, что вы выполнили проверку предварительных условий, загрузку и проверку установочного пакета, предварительную обработку узлов и другие подготовительные работы.

Содержание

Процесс

- Загрузка и распаковка установочного пакета
- Запуск установщика
  - Режим сети и IP-семейство
- Конфигурация параметров
- Проверка успешной установки
- Установка плагина Product Docs

Описание параметров

Очистка установщика

Процесс

1 Загрузка и распаковка установочного пакета

Загрузите установочный пакет Core Package на любую машину из узлов управляющей плоскости кластера `global` и распакуйте его согласно следующей команде:

```
# Предполагается, что папка /root/cpaas-install уже существует на машине
tar -xvf {Path to Core Package File}/{Core Package File Name} -C /root/cpaas-install
cd /root/cpaas-install/installer || exit 1
```

INFO

- Эта машина станет первым узлом управляющей плоскости после завершения установки кластера `global`.
- После распаковки Core Package требуется не менее **100 ГБ** свободного места на диске. Пожалуйста, обеспечьте достаточные ресурсы хранения.
- Если вы загрузили Extensions Package, распакуйте его и ознакомьтесь с документацией, включённой в пакет, перед переходом к следующим шагам.

2 Запуск установщика

Выполните следующий скрипт установки для запуска установщика. После успешного запуска установщика в командной строке будет выведен адрес доступа к веб-консоли.

Через примерно 5 минут вы сможете использовать браузер на вашем ПК для доступа к веб-консоли, предоставляемой установщиком.

```
bash setup.sh
```

WARNING

Убедитесь, что IP-адрес и порт 8080 узла, на котором расположен установщик, доступны, чтобы обеспечить беспрепятственный доступ к веб-консоли после успешного запуска установщика.

Режим сети и IP-семейство

```
bash setup.sh --network-mode calico
```

Параметр `--network-mode` влияет на CNI создаваемого установщиком кластера `global`. Если этот параметр не указан, CNI кластера `global` по умолчанию будет Kube-OVN. Если вы хотите использовать Calico в качестве CNI, необходимо явно указать `--network-mode calico`.

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

Если вы планируете создать кластер `global` с Single-stack Network IPv6, необходимо явно указать `--ip-family ipv6` при запуске установщика. Без этого параметра создаваемый установщиком кластер `global` по умолчанию поддерживает Single-stack Network IPv4 и Dual-stack Network.

3 Конфигурация параметров

После завершения настройки параметров установки согласно подсказкам на странице подтвердите установку.

Описание параметров содержит подробное описание ключевых параметров. Пожалуйста, внимательно ознакомьтесь и настройте в соответствии с фактическими потребностями.

4 Проверка успешной установки

После завершения установки на странице будет отображён адрес доступа к платформе. Нажмите кнопку **Access**, чтобы перейти в веб-интерфейс платформы.

В представлении **Platform Management** последовательно выберите **Cluster Management > Clusters** и найдите кластер с именем `global`.

В выпадающем меню справа выберите `CLI Tools` и выполните следующую команду для проверки статуса установки:

```
# Проверка наличия неудачных Charts
kubectl get apprelease --all-namespaces
# Проверка наличия неудачных Pod
kubectl get pod --all-namespaces | awk '{if ($4 != "Running" && $4 !=
"Completed")print}' | awk -F'[/ ]+' '{if ($3 != $4)print}'
```

5 Установка плагина Product Docs

> **INFO**
>
> Плагин **Alauda Container Platform Product Docs** обеспечивает доступ к документации продукта внутри платформы. Все ссылки на помощь в платформе будут вести к этой документации. Если плагин не установлен, при нажатии на ссылки помощи в платформе будет возникать ошибка доступа 404.

1. Перейдите в раздел **Administrator**.
2. В левой боковой панели выберите **Marketplace** > **Cluster Plugins** и выберите кластер `global`.
3. Найдите плагин **Alauda Container Platform Product Docs** и нажмите **Install**.

Описание параметров

Параметр	Описание
Kubernetes Version	Все опциональные версии тщательно протестированы на стабильность и совместимость. **Рекомендация:** Выбирайте последнюю версию для оптимальных функций и поддержки.
Cluster Network Protocol	Поддерживает три режима: IPv4 single stack, IPv6 single stack, IPv4/IPv6 dual stack.

	Примечание: Если выбран режим dual stack, убедитесь, что у всех узлов корректно настроены IPv6-адреса; после установки изменить сетевой протокол нельзя.
Cluster Address	Введите заранее подготовленное доменное имя. Если доменное имя отсутствует, введите заранее подготовленный `global VIP`. `Self-Built VIP` по умолчанию отключён, включайте его только если у вас нет LoadBalancer. После включения установщик автоматически развернёт `keepalived` для обеспечения поддержки программного балансировщика нагрузки. **Примечание:** При использовании `Self-Built VIP` должны быть выполнены следующие условия, • Доступен используемый VRID; • Сетевая инфраструктура хоста поддерживает протокол VRRP; • Все узлы управляющей плоскости и VIP находятся в одной подсети. **Совет:** Для одноузловых развертываний в сценариях ознакомления с функциями можно напрямую указать IP узла. Нет необходимости включать `Self-Built VIP` или готовить сетевые ресурсы, такие как `global VIP`.
Platform Access Address	Если нет необходимости различать **Cluster Address** и **Platform Access Address**, введите тот же адрес, что и в **Cluster Address**. Если различие необходимо, например, если кластер `global` предназначен только для внутреннего доступа, а платформа должна обеспечивать внешний доступ, введите заранее подготовленное доменное имя или `External IP`. По умолчанию платформа использует HTTPS и не включает HTTP. Если требуется включить HTTP, сделайте это в

	Advanced Settings (не рекомендуется). **Примечание:** Доменное имя обязательно в следующих случаях, • Планируется план аварийного восстановления кластера `global`; • Платформа должна поддерживать доступ по IPv6. **Совет:** Если необходимо настроить дополнительные адреса доступа к платформе, их можно добавить в **Other Settings > Other Platform Access Addresses** на следующем шаге. Или после установки добавить их в управлении платформой согласно руководству пользователя.
Certificate	Платформа по умолчанию предоставляет самоподписанные сертификаты для поддержки HTTPS-доступа. Если требуется использовать собственный сертификат, можно загрузить существующий сертификат.
Image Repository	По умолчанию используется репозиторий образов `Platform Deployment`, содержащий образы всех компонентов. Если необходимо использовать `External` репозиторий образов, обратитесь в техническую поддержку для получения плана синхронизации образов перед настройкой.
Container Network	Подсеть по умолчанию и сегмент сети Service кластера не должны пересекаться. При использовании Kube-OVN Overlay сети убедитесь, что контейнерная сеть и сетевая инфраструктура хоста не находятся в одном сетевом сегменте, иначе могут возникнуть сетевые сбои.

Node Name	Если выбран параметр `Host Name as Node Name`, убедитесь, что имена хостов всех узлов уникальны.
`global` **Cluster Platform Node Isolation**	Включайте только если планируете запускать рабочие нагрузки приложений в кластере `global`. **После включения:** • Узлы можно настроить как `Platform Exclusive`, то есть запускать только компоненты платформы, обеспечивая изоляцию платформенных и прикладных рабочих нагрузок; • Исключаются рабочие нагрузки типа DaemonSet.
Add Node	**Узел управляющей плоскости:** • Поддерживается добавление 1 или 3 узлов управляющей плоскости (3 для конфигурации высокой доступности); • Если включён `Platform Exclusive`, `Deployable Applications` принудительно отключается, и узлы управляющей плоскости запускают только компоненты платформы; • Если `Platform Exclusive` отключён, можно выбрать включение или отключение `Deployable Applications`, позволяя узлам управляющей плоскости запускать рабочие нагрузки приложений. **Рабочий узел:** • Если включён `Platform Exclusive`, `Deployable Applications` принудительно отключается; • Если `Platform Exclusive` отключён, `Deployable Applications` принудительно включается. При использовании Kube-OVN можно указать сетевую карту узла, введя имя шлюза. Если проверка доступности узла не пройдена, отрегулируйте параметры согласно подсказкам на странице и попробуйте

	добавить снова.

Очистка установщика

Обычно установщик автоматически удаляется после установки. Если установщик не удалился автоматически через 30 минут после установки, выполните следующую команду на узле, где расположен установщик, чтобы принудительно удалить контейнер установщика:

```
docker rm -f minialauda-control-plane
```

Восстановление после катастрофы глобального кластера

Содержание

Обзор

Поддерживаемые сценарии катастроф

Неподдерживаемые сценарии катастроф

Примечания

Обзор процесса

Требуемые ресурсы

Процесс

Шаг 1: Установка основного кластера

Шаг 2: Установка резервного кластера

Шаг 3: Включение синхронизации etcd

Процесс восстановления после катастрофы

Регулярные проверки

Загрузка пакетов

FAQ

Обзор

Данное решение предназначено для сценариев восстановления после катастрофы, связанных с кластером `global`. Кластер `global` служит управляющей плоскостью платформы и отвечает за управление другими кластерами. Для обеспечения непрерывной доступности сервисов платформы при сбое кластера `global` в этом

решении развертываются два кластера `global`: основной кластер (Primary Cluster) и резервный кластер (Standby Cluster).

Механизм восстановления основан на синхронизации данных etcd в реальном времени с основного кластера на резервный. В случае недоступности основного кластера из-за сбоя, сервисы могут быстро переключиться на резервный кластер.

Поддерживаемые сценарии катастроф

- Неисправимый системный сбой основного кластера, делающий его неработоспособным;
- Сбой физических или виртуальных машин, на которых размещён основной кластер, приводящий к его недоступности;
- Сбой сети в месте расположения основного кластера, вызывающий прерывание сервиса;

Неподдерживаемые сценарии катастроф

- Сбои приложений, развернутых внутри кластера `global`;
- Потеря данных из-за сбоев в системе хранения (вне области синхронизации etcd);

Роли **Primary Cluster** и **Standby Cluster** являются относительными: кластер, обслуживающий платформу в данный момент, считается основным (DNS указывает на него), а резервный — резервным. После переключения роли меняются местами.

Примечания

- Решение синхронизирует только данные etcd кластера `global`; данные реестра, chartmuseum и других компонентов не включены;
- Для удобства отладки и управления рекомендуется называть узлы в стиле `standby-global-m1`, чтобы указывать принадлежность узла к основному или резервному кластеру.
- Восстановление данных приложений внутри кластера не поддерживается;

- Для надежной синхронизации etcd требуется стабильное сетевое соединение между двумя кластерами;
- Если кластеры основаны на гетерогенных архитектурах (например, x86 и ARM), используйте пакет установки с поддержкой двух архитектур;
- Следующие пространства имён исключены из синхронизации etcd. Если в них создаются ресурсы, пользователи должны самостоятельно выполнять их резервное копирование:

```
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
```

- Если оба кластера используют встроенные реестры образов, контейнерные образы необходимо загружать отдельно в каждый из них;
- Если в основном кластере развернут **Alauda DevOps Eventing v3** (knative-operator) и его экземпляры, те же компоненты должны быть предварительно развернуты в резервном кластере.

Обзор процесса

1. Подготовить единое доменное имя для доступа к платформе;
2. Направить домен на VIP основного кластера и установить **Primary Cluster**;

3. Временно переключить разрешение DNS на VIP резервного кластера для установки Standby Cluster;

4. Скопировать ключ шифрования ETCD основного кластера на узлы, которые впоследствии станут управляющими узлами резервного кластера;

5. Установить и включить плагин синхронизации etcd;

6. Проверить статус синхронизации и выполнять регулярные проверки;

7. В случае сбоя переключить DNS на резервный кластер для завершения восстановления.

Требуемые ресурсы

- Единый домен, который будет `Platform Access Address`, а также TLS-сертификат и приватный ключ для обслуживания HTTPS на этом домене;
- Выделенный виртуальный IP-адрес для каждого кластера — один для **Primary Cluster** и другой для Standby Cluster;
  - Предварительно настроить балансировщик нагрузки для маршрутизации TCP-трафика на порты `80`, `443`, `6443`, `2379` и `11443` на управляющие узлы за соответствующим VIP.

Процесс

Шаг 1: Установка основного кластера

ПРИМЕЧАНИЯ ПО УСТАНОВКЕ DR (Окружение восстановления после катастрофы)

При установке основного кластера DR-окружения,

- В первую очередь задокументируйте все параметры, установленные при следовании руководству веб-интерфейса установки. Необходимо сохранить некоторые опции

одинаковыми при установке резервного кластера.

- Необходимо предварительно настроить **пользовательский** Load Balancer для маршрутизации трафика, направленного на виртуальный IP. Опция `Self-built VIP` недоступна.
- Поле `Platform Access Address` ДОЛЖНО содержать домен, а `Cluster Endpoint` — виртуальный IP-адрес.
- Оба кластера ДОЛЖНЫ быть настроены на использование `An Existing Certificate` (одинакового сертификата), при необходимости запросите легитимный сертификат. Опция `Self-signed Certificate` недоступна.
- При установке `Image Repository` в значение `Platform Deployment` поля `Username` и `Password` НЕ ДОЛЖНЫ быть пустыми; поле `IP/Domain` ДОЛЖНО содержать домен, используемый как `Platform Access Address` .
- Порты `HTTP Port` и `HTTPS Port` для `Platform Access Address` ДОЛЖНЫ быть 80 и 443 соответственно.
- На втором шаге установки (Step: `Advanced`) поле `Other Platform Access Addresses` ДОЛЖНО включать виртуальный IP текущего кластера.

Обратитесь к следующей документации для завершения установки:

- Подготовка к установке
- Установка

Шаг 2: Установка резервного кластера

1. Временно направьте доменное имя на VIP резервного кластера;
2. Войдите на первый управляющий узел **Primary Cluster** и скопируйте конфигурацию шифрования etcd на все управляющие узлы резервного кластера:

```
# Предположим, управляющие узлы основного кластера: 1.1.1.1, 2.2.2.2 и 3.3.3.3
# управляющие узлы резервного кластера: 4.4.4.4, 5.5.5.5 и 6.6.6.6
for i in 4.4.4.4 5.5.5.5 6.6.6.6  # Замените IP управляющих узлов резервного кластера
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. Установите резервный кластер так же, как и основной

ПРИМЕЧАНИЯ ПО УСТАНОВКЕ РЕЗЕРВНОГО КЛАСТЕРА

При установке резервного кластера DR-окружения, следующие параметры ДОЛЖНЫ совпадать с параметрами **основного кластера**:

- Поле `Platform Access Address`.
- Все поля `Certificate`.
- Все поля `Image Repository`.
- Важно: убедитесь, что учётные данные репозитория образов и пользователь ACP admin совпадают с теми, что установлены в **Primary Cluster**.

И ОБЯЗАТЕЛЬНО следуйте `NOTES OF DR (Disaster Recovery Environment) INSTALLING` из Шага 1.

Обратитесь к следующей документации для завершения установки:

- Подготовка к установке
- Установка

Шаг 3: Включение синхронизации etcd

1. Настройте балансировщик нагрузки для перенаправления порта `2379` на управляющие узлы соответствующего кластера. Поддерживается ТОЛЬКО TCP-

режим; перенаправление на уровне L7 не поддерживается.

2. Получите доступ к веб-консоли **резервного глобального кластера** через его VIP и переключитесь в режим **Administrator**;
3. Перейдите в **Marketplace > Cluster Plugins**, выберите кластер `global`;
4. Найдите **Alauda Container Platform etcd Synchronizer**, нажмите **Install**, настройте параметры:
   - Используйте интервал синхронизации по умолчанию;
   - Оставьте переключатель логирования выключенным, если не требуется отладка.

Проверьте, что Pod синхронизации запущен на резервном кластере:

```
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"
```

Когда появится сообщение “Start Sync update”, пересоздайте один из Pod для повторного запуска синхронизации ресурсов с зависимостями ownerReference:

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

Проверьте статус синхронизации:

```
mirror_svc=$(kubectl get svc -n cpaas-system etcd-sync-monitor -o
jsonpath='{.spec.clusterIP}')
ipv6_regex="^[0-9a-fA-F:]+$"
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
```

Объяснение вывода:

- `LOCAL ETCD missed keys`: Ключи есть в основном кластере, но отсутствуют в резервном. Часто вызвано GC из-за порядка ресурсов при синхронизации. Перезапустите один Pod etcd-sync для исправления;
- `LOCAL ETCD surplus keys`: Лишние ключи есть только в резервном кластере. Перед удалением этих ключей из резервного кластера согласуйте с командой эксплуатации.

Если установлены следующие компоненты, перезапустите их сервисы:

- Alauda Container Platform Log Storage для Elasticsearch:

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

- Alauda Container Platform Monitoring для VictoriaMetrics:

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

Процесс восстановления после катастрофы

1. При необходимости перезапустите Elasticsearch на резервном кластере:

```
# Скопируйте installer/res/packaged-scripts/for-upgrade/ensure-asm-template.sh в
/root:
# НЕ пропускайте этот шаг

# при необходимости переключитесь на пользователя root
sudo -i

# проверьте, установлен ли Log Storage для Elasticsearch в глобальном кластере
_es_pods=$(kubectl get po -n cpaas-system | grep cpaas-elasticsearch | awk '{print
$1}')
if [[ -n "${_es_pods}" ]]; then
    # Если скрипт вернул ошибку 401, перезапустите Elasticsearch
    # затем выполните скрипт для повторной проверки кластера
    bash /root/ensure-asm-template.sh

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

2. Проверьте согласованность данных в резервном кластере (та же проверка, что и в Шаге 3);

3. Удалите плагин синхронизации etcd;

4. Уберите перенаправление порта `2379` с обоих VIP;

5. Переключите DNS домена платформы на VIP резервного кластера, который теперь становится основным;

6. Проверьте разрешение DNS:

```
kubectl exec -it -n cpaas-system deployments/sentry -- nslookup <platform access
domain>
# Если разрешение некорректно, перезапустите Pod coredns и повторяйте попытки
до успеха
```

7. Очистите кэш браузера и зайдите на страницу платформы, чтобы убедиться, что она отражает бывший резервный кластер;

8. Перезапустите следующие сервисы (если установлены):

- Alauda Container Platform Log Storage для Elasticsearch:

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

- Alauda Container Platform Monitoring для VictoriaMetrics:

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

- cluster-transformer:

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

9. Если рабочие кластеры отправляют данные мониторинга в основной, перезапустите warlock в рабочем кластере:

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

10. На исходном основном кластере повторите шаги Включения синхронизации etcd, чтобы превратить его в новый резервный кластер.

Регулярные проверки

Регулярно проверяйте статус синхронизации на резервном кластере:

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

Если обнаружены отсутствующие или лишние ключи, следуйте инструкциям в выводе для их устранения.

Загрузка пакетов

При использовании `violet` для загрузки пакетов в резервный кластер необходимо указывать параметр `--dest-repo` с VIP резервного кластера. Если этот параметр опущен, пакет будет загружен в репозиторий образов основного кластера, что помешает резервному кластеру установить или обновить соответствующее расширение.

FAQ

- Инструкция на случай, если ключ шифрования ETCD резервного кластера не был синхронизирован с ключом **основного кластера** до установки резервного кластера.

1. Получите ключ шифрования ETCD на любом управляющем узле резервного кластера:

```
ssh <user>@<STANDBY cluster control plane node> sudo cat /etc/kubernetes/encryption-provider.conf
```

Он должен выглядеть так:

```
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
  - resources:
    - secrets
    providers:
    - aescbc:
        keys:
        - name: key1
          secret: MTE0NTE0MTkxOTgxMA==
```

2. Создайте резервную копию ключа шифрования ETCD на любом управляющем узле основного кластера (например, 1.1.1.1):

```
# Войдите на управляющий узел основного кластера 1.1.1.1
ssh "<user>@1.1.1.1"
sudo install -o root -g root -m 600 /etc/kubernetes/encryption-provider.conf
/etc/kubernetes/encryption-provider.conf.bak
```

3. Объедините ключ шифрования ETCD резервного кластера в файл `/etc/kubernetes/encryption-provider.conf` на узле `1.1.1.1`, убедившись, что имена ключей уникальны. Например, если ключ основного кластера называется `key1`, переименуйте ключ резервного в `key2`:

```
apiVersion: apiserver.config.k8s.io/v1
kind: EncryptionConfiguration
resources:
  - resources:
    - secrets
    providers:
    - aescbc:
        keys:
        - name: key1
          secret: My4xNDE1OTI2NTM1ODk3
        - name: key2
          secret: MTE0NTE0MTkxOTgxMA==
```

4. Убедитесь, что новый файл `/etc/kubernetes/encryption-provider.conf` перезаписывает ВСЕ реплики на управляющих узлах обоих кластеров:

```
# Предположим, управляющие узлы основного кластера: 1.1.1.1, 2.2.2.2 и 3.3.3.3
# управляющие узлы резервного кластера: 4.4.4.4, 5.5.5.5 и 6.6.6.6

# Поскольку 1.1.1.1 уже настроен на использование обоих ключей ETCD,
# войдите на узел 1.1.1.1 и выполните следующие команды:
for i in \
    2.2.2.2 3.3.3.3 \
    4.4.4.4 5.5.5.5 6.6.6.6 \
; do
    scp /etc/kubernetes/encryption-provider.conf "<user>@${i}:/tmp/encryption-
provider.conf"
    ssh "<user>@${i}" '
#!/bin/bash
set -euo pipefail

sudo install -o root -g root -m 600 \
    /etc/kubernetes/encryption-provider.conf /etc/kubernetes/encryption-
provider.conf.bak
sudo install -o root -g root -m 600 \
    /tmp/encryption-provider.conf            /etc/kubernetes/encryption-provider.conf
sudo rm -f /tmp/encryption-provider.conf

_pod_name="kube-apiserver"
_pod_id=$(sudo crictl ps --name "${_pod_name}" --no-trunc --quiet)
if [[ -z "${_pod_id}" ]]; then
    echo "FATAL: не найден pod `kube-apiserver` на узле $(hostname)"
    exit 1
fi
sudo crictl rm --force "${_pod_id}"
sudo systemctl restart kubelet.service
'
done
```

5. Перезапустите kube-apiserver на узле 1.1.1.1

```
_pod_name="kube-apiserver"
_pod_id=$(sudo crictl ps --name "${_pod_name}" --no-trunc --quiet)
if [[ -z "${_pod_id}" ]]; then
    echo "FATAL: не найден pod `kube-apiserver` на узле $(hostname)"
    exit 1
fi
sudo crictl rm --force "${_pod_id}"
sudo systemctl restart kubelet.service
```