

Установка

В этом документе представлена вся информация, касающаяся установки ACP.

Обзор

[Обзор](#)

Подготовка к установке

[Предварительные требования](#)

[Загрузка](#)

[Предварит](#)

Установка

[Установка](#)

Восстановление после катастрофы для глобального кластера

Восстановление после катастрофы для глобального кластера

Обзор

Следуя этому руководству, вы завершите установку **ACP Core**.

Если вам нужно понять концепцию **ACP Core**, обратитесь к разделу [Architecture](#).

Установка **ACP Core** подразумевает процесс развертывания кластера `global`.

После установки вы сможете **создавать новые рабочие кластеры** или **подключать существующие**, а также устанавливать дополнительные **Extensions** для расширения возможностей платформы.

INFO

Перед установкой убедитесь, что вы выполнили планирование емкости, предварительную подготовку среды и проверку предварительных условий, чтобы гарантировать соответствие аппаратного обеспечения, сети и ОС каждого узла требованиям.

В следующем содержании рассматриваются архитектура платформы, методы установки и объяснения ключевых терминов, которые помогут вам усвоить основные моменты в процессе фактической установки.

Содержание

Метод установки

Приложение — Customer Portal Alauda

Назначение и обзор

Ключевые возможности

Руководство по использованию

Метод установки

Процесс установки кластера `global` 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 сетевая конфигурация.
- **Загрузка установочного пакета:** Войдите в Customer Portal Alauda, чтобы получить последний установочный пакет.
- **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 режим IP-протокола (IPv4/IPv6/dual stack) и настройку VIP в соответствии с реальной средой.
- **Конфигурация параметров:** Через Web UI, предоставленный установщиком, последовательно задайте версию Kubernetes, сеть кластера, имя узла, адрес доступа и другие ключевые параметры для завершения установки кластера `global`.

3. Этап проверки

- **Проверка состояния системы:** После завершения установки войдите в Web UI платформы для проверки состояния кластера и работы 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.

В последующих главах будут подробно описаны операции, параметры конфигурации и методы проверки на каждом этапе установки. Пожалуйста, внимательно ознакомьтесь и

выполните соответствующую подготовительную работу перед официальной установкой.

Приложение — Customer Portal Alauda

Customer Portal Alauda — это единая платформа обслуживания клиентов и доставки Alauda, предоставляющая централизованный доступ ко всем ресурсам и службам поддержки, связанным с продуктом. Она служит официальной точкой входа для клиентов, партнеров и команд доставки для получения программных пакетов, документации, поддержки и управления лицензиями в безопасном и согласованном формате.

Назначение и обзор

Customer Portal Alauda упрощает полный жизненный цикл продукта — от установки и настройки до обслуживания и поддержки — объединяя все необходимые ресурсы в одной платформе. Это гарантирует, что каждое развертывание основано на проверенных версиях программного обеспечения и официальных технических рекомендациях.

Ключевые возможности

- **Загрузка продуктов**

Обеспечивает доступ к проверенным установочным и обновляющим пакетам, гарантируя, что развертывания соответствуют последним поддерживаемым версиям продукта.

- **База знаний**

Предлагает полную документацию по продукту, технические статьи, руководства по устранению неполадок и лучшие практики для помощи в установке, настройке и эксплуатации.

- **Тикеты поддержки**

Позволяет пользователям создавать, отслеживать и управлять запросами в службу поддержки напрямую онлайн, обеспечивая своевременное решение проблем и полную прозрачность процесса поддержки.

- **Маркетплейс приложений**

Предоставляет тщательно подобранную коллекцию официальных и сторонних расширений, которые можно установить для расширения или настройки возможностей платформы.

- **Управление лицензиями**

Поддерживает подачу заявок, активацию и продление программных лицензий, обеспечивая отслеживаемое и соответствующее использование лицензий во всех средах.

Руководство по использованию

Перед началом установки или обновления пользователи должны войти в **Customer Portal Alauda** с использованием авторизованной учетной записи для загрузки необходимых установочных пакетов и проверки статуса лицензий.

Для производственных и клиентских сред версии и документация, опубликованные в Customer Portal Alauda, всегда должны рассматриваться как **официальная базовая линия** для развертывания и обслуживания.

Подготовка к установке

[Предварительные требования](#)

[Загрузка](#)

[Предварит](#)

Предварительные требования

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

INFO

1. Платформа в настоящее время 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](#).
3. Убедитесь, что все новые узлы соответствуют [Требованиям к узлам](#).
4. Производительность и емкость дисков также должны соответствовать [Требованиям к конфигурации дисков](#).

Содержание

Планирование ресурсов

Архитектуры развертывания

Одиночный узел

Одиночный кластер

Мультикластер

Сеть

Сетевые ресурсы

Сетевая конфигурация

Правила переадресации LoadBalancer

Планирование ресурсов

В этом разделе приведены рекомендации по планированию ресурсов перед установкой АСР. Выберите подходящий сценарий развертывания в зависимости от вашей среды и потребностей, и подготовьте ресурсы соответственно.

INFO

Следующие рекомендации охватывают только минимальные ресурсы, необходимые для успешной установки **глобального кластера**.

Они **не включают** ресурсы, требуемые для любых **дополнительных расширений или компонентов**, развернутых на глобальном кластере.

Для подробных требований к каждому расширению обратитесь к соответствующей документации компонента.

Архитектуры развертывания

WARNING

Для архитектур ARM (например, Kunpeng 920) рекомендуется увеличить конфигурацию в **2 раза** по сравнению с минимальной конфигурацией для x86, но не менее чем в **1,5 раза**.

Например: если для x86 требуется 8 ядер и 16 ГБ памяти, то для ARM должно быть не менее 12 ядер и 24 ГБ, а рекомендуемая конфигурация — 16 ядер и 32 ГБ.

Перед установкой необходимо определить, какая архитектура развертывания лучше всего подходит для вашего случая использования. АСР поддерживает следующие три распространённые архитектуры развертывания:

- **Мультикластерная**

Выберите эту архитектуру, если вам нужно централизованно управлять несколькими кластерами Kubernetes. В этом режиме ACP состоит из одного **глобального кластера** и нескольких **рабочих кластеров**. Запуск рабочих нагрузок, не относящихся к платформе, на глобальном кластере может ухудшить стабильность и производительность платформы и должен быть избегаем.

- **Одиночный кластер**

Выберите эту архитектуру, если планируете установить только один кластер и запускать рабочие нагрузки непосредственно на нём. В этом режиме глобальный кластер также выступает в роли рабочего кластера, поэтому требует **больше ресурсов** по сравнению с чисто глобальной конфигурацией в мультикластерном режиме.

- **Одиночный узел**

WARNING

Эта архитектура предназначена исключительно для тестирования или демонстрационных целей и не должна использоваться в продуктивной среде.

Одиночный узел

В следующей таблице приведены **минимальные требования к оборудованию** для установки ACP в режиме **Одиночного узла**.

Ресурс	Минимальное требование
CPU	12 ядер
Память	24 ГБ
Хранилище	Емкость хранилища

Одиночный кластер

В этом режиме глобальный кластер выполняет функции как управляющей плоскости, так и рабочего кластера. Количество узлов управляющей плоскости глобального

кластера ДОЛЖНО быть 3.

Общее требование к ресурсам состоит из двух частей:

- **Базовые ресурсы** для самого глобального кластера
- **Дополнительные ресурсы** для запуска рабочих нагрузок на том же кластере

Ресурсы, необходимые для **высокодоступного глобального кластера**, следующие:

Ресурс	Минимальное требование
CPU	8 ядер
Память	16 ГБ
Хранилище	Емкость хранилища

Для оценки дополнительных ресурсов, необходимых для ваших рабочих нагрузок, обратитесь к [Оценке ресурсов для рабочего кластера](#)

Мультикластер

При управлении несколькими рабочими кластерами использование ресурсов глобального кластера пропорционально увеличивается с количеством управляемых кластеров. Дополнительные накладные расходы в основном связаны с регистрацией кластеров, мониторингом и синхронизацией управляющей плоскости.

Для оценки ресурсов, необходимых для вашего глобального кластера в зависимости от количества управляемых кластеров, обратитесь к [Оценке ресурсов для глобального кластера](#)

Сеть

Перед установкой убедитесь, что необходимые сетевые ресурсы подготовлены.

Если в вашей среде доступен аппаратный LoadBalancer, рекомендуется использовать его. Если нет, можно включить `Self-built VIP`, который обеспечивает программное

балансирование нагрузки с помощью keeralived.

Примечание: `Self-built VIP` не поддерживает настройку доменных имён.

Сетевые ресурсы

Ресурс	Обязательность	Количество	Описание
<code>global</code> VIP	Обязательно	1	<p>Используется для доступа узлов к kube-apiserver, настраивается в устройстве балансировки нагрузки для обеспечения высокой доступности.</p> <p>Этот IP также может использоваться как адрес доступа к Web UI платформы.</p> <p>Рабочие кластеры в той же сети, что и <code>global</code> кластер, также могут обращаться к <code>global</code> кластеру через этот IP.</p>
Внешний IP	Опционально	По необходимости	<p>Если существуют рабочие кластеры, находящиеся в другой сети, например, в гибридном облаке, этот IP обязателен. Рабочие кластеры в других сетях обращаются к <code>global</code> кластеру через этот IP.</p> <p>Этот IP необходимо настроить в устройстве балансировки нагрузки для обеспечения высокой доступности.</p> <p>Этот IP также может использоваться как адрес доступа к Web UI платформы.</p>

Ресурс	Обязательность	Количество	Описание
Доменное имя	Рекомендуется	По необходимости	<p>Рекомендуется для Cluster Endpoint и адреса доступа к платформе. Пожалуйста, предоставьте его заранее и убедитесь, что разрешение доменного имени работает корректно.</p> <p>Использование доменного имени рекомендуется, поскольку при необходимости изменения VIP после установки кластера вы сможете легко обновить запись DNS без влияния на работу кластера.</p> <p>Доменное имя обязательно в следующих случаях:</p> <ul style="list-style-type: none">• Глобальный кластер должен поддерживать доступ по IPv6;• Планируется аварийное восстановление для глобального кластера.
Сертификат	Опционально	По необходимости	Рекомендуется использовать доверенный сертификат, чтобы избежать предупреждений безопасности в браузере; если не предоставлен, установщик сгенерирует самоподписанный

Ресурс	Обязательность	Количество	Описание
			сертификат, но при использовании HTTPS могут возникать риски безопасности.

ПРИМЕЧАНИЕ

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

Сетевая конфигурация

Тип	Описание требования
Пропускная способность сети	<p>Пропускная способность внутри кластера должна быть ≥ 1 Гбит/с (рекомендуется 10 Гбит/с).</p> <p>Пропускная способность между кластерами должна быть ≥ 100 Мбит/с (рекомендуется 1 Гбит/с).</p> <p>Недостаточная пропускная способность может значительно ухудшить производительность запросов данных.</p>
Задержка сети	<p>Задержка внутри кластера должна быть ≤ 10 мс.</p> <p>Задержка между кластерами должна быть ≤ 100 мс (рекомендуется ≤ 30 мс).</p>
Сетевая политика	<p>Пожалуйста, ознакомьтесь с разделом LoadBalancer Forwarding Rules для обеспечения открытия необходимых портов.</p>
Диапазон IP-адресов	<p>Узлы кластера <code>global</code> должны избегать использования сетевого сегмента 172.17-18. Если он уже используется, пожалуйста, настройте конфигурацию <code>nerdctl</code> (добавьте параметр <code>vip</code>), чтобы избежать конфликтов.</p>

Правила переадресации LoadBalancer

Это правило предназначено для обеспечения нормального приёма трафика кластером `global` через LoadBalancer. Пожалуйста, проверьте сетевую политику согласно таблице ниже, чтобы убедиться, что соответствующие порты открыты.

Исходный IP	Протокол	Целевой IP	Целевой порт	Описание
<code>global</code> VIP, Внешний IP	TCP	Все IP узлов управляющей плоскости	443	<p>Обеспечивает доступ к Web UI платформы, репозиторию образов и Kubernetes API Server через протокол HTTPS. Порт по умолчанию — <code>443</code>. Если необходимо использовать пользовательский HTTPS-порт, выполните следующие действия:</p> <ul style="list-style-type: none">• Замените целевой порт в правиле переадресации на ваш пользовательский номер порта.• Позже в параметрах установки Web UI

Исходный IP	Протокол	Целевой IP	Целевой порт	Описание
				укажите ваш пользовательский порт.
<code>global</code> VIP, Внешний IP	TCP	Все IP узлов управляющей плоскости	6443	Этот порт обеспечивает доступ к Kubernetes API Server для узлов внутри кластера.
<code>global</code> VIP, Внешний IP	TCP	Все IP узлов управляющей плоскости	11443	Этот порт обеспечивает доступ к репозиторию образов для узлов внутри кластера. Примечание: Если вы планируете использовать внешний репозиторий образов вместо репозитория по умолчанию, предоставляемого кластером <code>global</code> , этот порт настраивать не нужно.

- Рекомендуется настроить проверки состояния на LoadBalancer для мониторинга статуса портов.
- Если планируется реализация аварийного восстановления для 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 Core Package

Перед установкой необходимо скачать **Core Package**.

INFO

Начиная с версии Alauda Container Platform v4.1, если вы загружаете как **Core Package**, так и **Extensions Packages**, необходимо сначала завершить установку **Core Package** перед загрузкой и установкой **Extensions Packages**.

Войдите в **Alauda Customer Portal**, чтобы скачать **Core Package**.

Пакеты доступны для архитектур **x86**, **ARM** и **гибридной**. Гибридный пакет включает образы как для x86, так и для ARM, из-за чего размер пакета больше. Выберите пакет, который лучше всего соответствует вашей среде.

Если у вас нет зарегистрированного аккаунта, обратитесь в техническую поддержку.

Содержание

[Миграция с одноархитектурного на гибридный пакет](#)

Миграция с одноархитектурного на гибридный пакет

Если вы изначально устанавливали Core Package для x86 или ARM, но позже потребуется поддержка другой архитектуры, необходимо повторно скачать **гибридный Core Package** и выполнить следующие шаги:

1. Загрузите вновь скачанный гибридный Core Package на любой узел control plane глобального кластера.
2. Распакуйте пакет и используйте включённый скрипт `upgrade.sh` для синхронизации мультиархитектурных образов с вашим реестром образов:

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

3. После завершения скрипта проверьте ресурс `cluster.platform.tkestack.io`, чтобы убедиться, что метка `cpaas.io/node-arch-constraint` отсутствует. Если она есть, её необходимо удалить:

```
kubectl get cluster.platform.tkestack.io global -oyaml | grep cpaas.io/  
node-arch-constraint  
# Если вывод есть, отредактируйте ресурс, чтобы удалить метку; иначе эт  
от шаг можно пропустить.  
kubectl edit cluster.platform.tkestack.io global ### Отредактируйте п  
оле labels и удалите cpaas.io/node-arch-constraint
```

Предварительная обработка узлов

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

Содержание

Поддерживаемые версии ОС и ядра

x86

ARM

Выполнение скрипта быстрой настройки

Проверки узлов

Приложение

Удаление конфликтующих пакетов

Настройка поискового домена

Поддерживаемые версии ОС и ядра

В следующей таблице перечислены поддерживаемые операционные системы, их проверенные версии и соответствующие протестированные версии ядра.

Платформа строго соблюдает политику соответствия версий для официальной поддержки:

- **Версия ОС (x.y.z):** патч-версии (`z`) могут отличаться, но основные и минорные версии (`x` и `y`) должны строго соответствовать проверенным версиям. Изменение

`x` или `y` официально не поддерживается.

- **Версия ядра (x.y.z-build):** суффикс сборки (`build`) может отличаться, но основная версия ядра (`x.y.z`) должна строго соответствовать протестированным версиям. Изменение `x.y.z` официально не поддерживается.

INFO

- Поддерживается только версия ядра, поставляемая с официальной операционной системой. Если ОС, версия ядра или архитектура CPU не соответствуют требованиям, обратитесь в техническую поддержку.
- В Kylin V10, V10-SP1 и V10-SP2 известны проблемы с ядром, которые могут вызывать сбой сетевого доступа **NodePort**, рекомендуется обновиться до **Kylin 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`

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

CentOS

- CentOS 7.6 до 7.9: `3.10.0-1127` и `3.10.0-1160`

Примечание: CentOS не поддерживает **Calico Vxlan IPv6**.

Ubuntu

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

Примечание: версии Ubuntu HWE (Hardware Enablement) не поддерживаются.

Kylin Linux Advanced Server

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

ARM

Kylin Linux Advanced Server

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

Примечание: архитектура ARM поддерживает только `Kunpeng 920`. Для других моделей обратитесь в техническую поддержку.

Выполнение скрипта быстрой настройки

Установочный пакет ACP предоставляет скрипт для быстрой настройки узлов.

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

Выполните скрипт:

```
bash init.sh
```



ВНИМАНИЕ

`init.sh` не гарантирует, что все перечисленные ниже проверки будут корректно выполнены. Вам всё равно необходимо продолжить выполнение следующих шагов.

Проверки узлов








Ниже перечислены все проверки, 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, которую необходимо пройти.
-  Обозначает проверку, которую нужно выполнить в определённых сценариях. Пожалуйста, определите, выполняются ли соответствующие условия согласно инструкциям. Если да, необходимо их устранить.

Список проверок:

• ОС и ядро

-  В конфигурации загрузчика grub машины должен присутствовать параметр `transparent_hugepage=never`.
-  В конфигурации загрузчика grub системы CentOS 7.x должен присутствовать параметр `cgroup.memory=nokmem`.
-  Проверить, включены ли модули ядра `ip_vs`, `ip_vs_rr`, `ip_vs_wrr` и `ip_vs_sh`.
-  Если версия ядра ниже 4.19.0 (или RHEL ниже 4.18.0), проверить, включены ли модули ядра `nf_conntrack_ipv4` и (для IPv6) `nf_conntrack_ipv6`.
-  Если в кластере `global` планируется использовать CNI `Kube-OVN`, модули ядра `geneve` и `openvswitch` должны быть включены.
-  Отключить `apparmor/selinux` и `файрвол`.
-  Отключить `swap`.

• Пользователи и права

-  SSH-пользователь узла имеет права `root` и может использовать `sudo` без пароля.
-  Параметры `UseDNS` и `UsePAM` в `/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 `..`, `..` или `-. .`.
- В `/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 (необходимо отключить `systemd-resolved`).
- В файле `/etc/resolv.conf` не должны настраиваться поисковые домены (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 расположен установщик:** `8080`, `12080`, `12443`, `16443`, `2379`, `2380`, `6443`, `10249` ~ `10256`
 - **Рабочие узлы:** `10249` ~ `10256`
- Если кластер использует **Kube-OVN** или **Calico**, убедитесь, что следующие порты не заняты:
 - **Kube-OVN:** `6641`, `6642`
 - **Calico:** `179`

- **⚠** Убедитесь, что IP-адреса в сетевом сегменте `172.17.x.x` ~ `172.18.x.x`, необходимых для `nerdctl`, не заняты. Если IP в этом сегменте заняты и их нельзя изменить, обратитесь в техническую поддержку.
- **Требования к программному обеспечению и каталогам:**
 - **✓** Должны быть установлены: `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`.
 - **✓** Удалить пакеты, конфликтующие с компонентами кластера `global` (см. [Удаление конфликтующих пакетов](#)).
 - **✓** Следующие файлы и каталоги должны быть удалены, если они существуют:
 - `/var/lib/docker`
 - `/var/lib/nerdctl`
 - `/opt/nerdctl/`
 - `/var/lib/containerd`
 - `/var/log/pods`
 - `/var/lib/kubelet/pki`
- **Межузловые проверки**
 - **✓** Между узлами кластера `global` не должно быть ограничений сетевого файрвола.
 - **✓** `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 окружении `docker/nerdctl/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`, `nerdctl`, `containerd`, `runc`, `podman`, сетями контейнеров, runtime 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 \  
  aptainer \  
  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 \  
  aptainer \  
  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-api  
server \  
  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-api
server \
  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 \
  aptainer \
  containerd \
  kubernetes kubernetes-master kubernetes-node kubernetes-client kub
ernetes-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 \  
  aptainer \  
  containerd \  
  kubernetes kubernetes-master kubernetes-node kubernetes-client kub  
ernetes-kubeadm \  
; do  
  yum remove "$x"  
done
```

Настройка поискового домена

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

Требования к настройке

- **Количество доменов:** количество доменов в строке `search` должно быть меньше `domainCountLimit - 3` (по умолчанию `domainCountLimit` равен 32).
- **Длина одного домена:** каждое доменное имя не должно превышать 253 символа.
- **Общая длина:** общее количество символов всех доменных имён и пробелов не должно 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 резервную копию файла.

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

IP Family

Конфигурация параметров

Проверка успешной установки

Установка плагина Product Docs

Описание параметров

Очистка установщика

Дополнительные ресурсы

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 уже загрузили расширения, сначала завершите установку ACP Core, а затем следуйте инструкции в разделе [Extend](#) для загрузки и установки расширений.

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 корректный доступ к веб-консоли после успешного

запуска установщика.

IP Family

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

Если вы планируете создать кластер `global` с Single-stack Network IPv6, необходимо явно указать параметр `--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 их в соответствии с реальными требованиями.

4 Проверка успешной установки

После завершения установки на странице будет отображён URL доступа к платформе. Нажмите кнопку **Access** для открытия Web UI платформы и проверьте доступность платформы.

Далее выполните следующие команды на узле установки для проверки статуса установки:

```
# Проверка наличия неудачных Charts
kubectl get apprelease --all-namespaces
# Проверка наличия Pod, не находящихся в статусе Running или Complete
d
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** предоставляет 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 404.

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

Описание параметров

Параметр	Описание
Kubernetes Version	<p>Все опциональные версии тщательно протестированы на стабильность и совместимость.</p> <p>Рекомендация: Выбирайте последнюю версию для оптимальных возможностей и поддержки.</p>
Cluster Network Protocol	<p>Поддерживает три режима: IPv4 single stack, IPv6 single stack, IPv4/IPv6 dual stack.</p> <p>Примечание: Если выбран режим dual stack, убедитесь, что у всех узлов корректно настроены IPv6-адреса; после установки изменить сетевой протокол нельзя.</p>
Cluster Endpoint	<p>Введите заранее подготовленное доменное имя.</p> <p>Если доменное имя отсутствует, введите заранее подготовленный <code>global VIP</code>.</p> <p><code>Self-built VIP</code> по умолчанию отключён, включайте его только</p>

если у вас нет предоставленного LoadBalancer.

При использовании `Self-built VIP` должны быть выполнены следующие условия:

- Доступен рабочий VRID;
- Сетевая инфраструктура хоста поддерживает протокол VRRP;
- Все узлы управляющей плоскости и VIP находятся в одной подсети.

Совет: Для одноузловых развертываний в сценариях ознакомления можно напрямую указать IP узла. В этом случае нет необходимости включать `Self-built VIP` или готовить сетевые ресурсы, такие как `global VIP`.

Platform Access

Address

Если нет необходимости различать **Cluster Endpoint** и **Platform Access Address**, введите тот же адрес, что и в **Cluster Endpoint**.

Если различие необходимо, например, когда 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 разделе **Advanced Settings** (не рекомендуется).

Примечание: В следующих случаях необходимо ввести доменное имя:

- 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 разделе **Other Settings** > **Other Platform Access Addresses** на следующем шаге. Либо

после установки добавить их в управлении платформой согласно руководству пользователя.

Certificate

Платформа по умолчанию предоставляет самоподписанные сертификаты для поддержки HTTPS.
Если требуется использовать собственный сертификат, можно загрузить существующий.

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 находятся в разных подсетях, иначе возможны 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 нагрузок;

- Исключаются рабочие нагрузки типа DaemonSet.

Узел управляющей плоскости:

- Поддерживается добавление 1 или 3 узлов управляющей плоскости (3 для конфигурации высокой доступности);
- Если включён `Platform Exclusive`, параметр `Deployable Applications` принудительно отключается, и узлы управляющей плоскости запускают только компоненты платформы;
- Если `Platform Exclusive` отключён, можно выбрать, включать ли `Deployable Applications`, позволяя узлам управляющей плоскости запускать рабочие нагрузки приложений.

Add Node

Рабочий узел:

- Если включён `Platform Exclusive`, параметр `Deployable Applications` принудительно отключается;
- Если `Platform Exclusive` отключён, параметр `Deployable Applications` принудительно включается.

При использовании Kube-OVN можно указать сетевую карту узла, введя имя шлюза.

Если проверка доступности узла не прошла, отрегулируйте параметры согласно подсказкам на странице и попробуйте добавить узел снова.

Очистка установщика

Обычно установщик удаляется автоматически после установки. Если установщик не удалился автоматически в течение 30 минут после установки, выполните следующую команду на узле, где расположен установщик, для принудительного удаления контейнера установщика:

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

Дополнительные ресурсы

- [Загрузка и установка расширений](#)

Восстановление после катастрофы для глобального кластера

Содержание

Обзор

Поддерживаемые сценарии катастроф

Неподдерживаемые сценарии катастроф

Примечания

Обзор процесса

Требуемые ресурсы

Процедура

Шаг 1: Установка основного кластера

Шаг 2: Установка резервного кластера

Шаг 3: Включение синхронизации etcd

Процесс восстановления после катастрофы

Регулярные проверки

Загрузка пакетов

Обзор

Это 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 платформы при сбое кластера `global` в рамках этого решения разворачиваются два кластера `global`: основной кластер (Primary Cluster) и резервный кластер (Standby Cluster).

Механизм восстановления после катастрофы основан на синхронизации данных etcd в реальном времени с основного кластера на резервный. Если основной кластер становится недоступен из-за сбоя, сервисы могут быстро переключиться на резервный кластер.

Поддерживаемые сценарии катастроф

- Неисправимый системный сбой основного кластера, делающий его непригодным к работе;
- 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, вызванная сбоями системы 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

- Это решение синхронизирует только данные etcd кластера `global`; данные реестра, chartmuseum и других компонентов не включены;

- Для удобства устранения неполадок и управления рекомендуется называть узлы в стиле `standby-global-m1`, чтобы указывать, к какому кластеру принадлежит узел (Primary или Standby).
- 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 установочный пакет с поддержкой двух архитектур;
- Следующие пространства имён исключены из синхронизации etcd. Если в этих пространствах создаются ресурсы, пользователи должны выполнять их резервное копирование вручную:

```
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;
- Если в основном кластере развернут **Alauda DevOps Eventing v3** (knative-operator) и его экземпляры, аналогичные компоненты должны быть предварительно установлены в резервном кластере.

Обзор процесса

1. Подготовить единое доменное имя для доступа к платформе;

2. Указать домен на VIP основного кластера и установить **Primary Cluster**;
3. Временно переключить разрешение DNS на резервный VIP 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 после катастрофы.

Требуемые ресурсы

- Единое доменное имя, которое 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 портах `80`, `443`, `6443`, `2379` и `11443` на управляющие узлы за соответствующим VIP.

Процедура

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 такими же при установке резервного кластера.

- ДОЛЖЕН быть предварительно настроен **Load Balancer, предоставляемый пользователем**, для маршрутизации трафика, направленного на виртуальный IP. Опция **Self-built VIP** НЕ доступна.
- 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** НЕ доступна.
- При установке **Image Repository** в **Platform Deployment** поля **Username** и **Password** НЕ должны быть пустыми; поле **IP/Domain** ДОЛЖНО быть установлено на домен, используемый как **Platform Access Address**.
- Порты **HTTP Port** и **HTTPS Port** для **Platform Access Address** ДОЛЖНЫ быть 80 и 443 соответственно.
- На второй странице установки (Шаг: **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 укажите 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 на IP управляющих узлов ре
зервного кластера
do
  ssh "<user>@$i" "sudo mkdir -p /etc/kubernetes/"
  scp /etc/kubernetes/encryption-provider.conf "<user>@$i:/tmp/encrypti
on-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/en
ryption-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 быть такими же, как в **основном кластере**:

- 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 `ПРИМЕЧАНИЯМ ПО УСТАНОВКЕ DR (Окружение восстановления после катастрофы)` из Шага 1.

Обратитесь к следующей документации для завершения установки:

- [Подготовка к установке](#)
- [Установка](#)

Шаг 3: Включение синхронизации etcd

1. При необходимости настройте балансировщик нагрузки для переадресации порта `2379` на управляющие узлы соответствующего кластера. Поддерживается ТОЛЬКО TCP-режим; переадресация на уровне L7 не поддерживается.

INFO

Переадресация порта через балансировщик нагрузки не обязательна. Если из резервного кластера доступен прямой доступ к активному глобальному кластеру, укажите адреса etcd через **Active Global Cluster ETCD Endpoints**.

2. Зайдите в веб-консоль **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:
 - Если порт `2379` не переадресуется через балансировщик, необходимо корректно настроить **Active Global Cluster ETCD Endpoints**;
 - Используйте значение по умолчанию для **Data Check Interval**;
 - Оставьте переключатель **Print detail logs** выключенным, если не требуется отладка.

Проверьте, что Pod синхронизации запущен в 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 "Start Sync update", пересоздайте один из pod'ов, чтобы повторно инициировать синхронизацию 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=$(kubectl get svc -n cpaas-system etcd-sync-monitor -o jsonpat
h='{.spec.clusterIP}')
ipv6_regex="^[0-9a-fA-F:]+$"
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
```

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,vms
elect,vminsert)'
```

Процесс восстановления после катастрофы

1. При необходимости перезапустите Elasticsearch в резервном кластере:

```
# Скопируйте installer/res/package-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 переадресацию порта `2379` с обоих VIP;
5. Переключите DNS домена платформы на резервный VIP, который теперь становится основным кластером;
6. Проверьте разрешение DNS:

```
kubectl exec -it -n cpaas-system deployments/sentry -- nslookup <platform access domain>
# Если разрешение некорректно, перезапустите pod'ы coredns и повторите попытку 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
```

- Если кластеры рабочих нагрузок отправляют данные мониторинга в основной, перезапустите warlock в кластере рабочих нагрузок:

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

- На исходном основном кластере повторите шаги [Включения синхронизации etcd](#), чтобы преобразовать его в новый резервный кластер.

Регулярные проверки

Регулярно проверяйте статус синхронизации в 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

Подробности по подкоманде `violet push` см. в разделе [Upload Packages](#).