Установка

В этом документе представлена вся информация, касающаяся установки ACP.

Обзор

Обзор

Подготовка к установке

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** означает 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 окружения и проверку предварительных условий, чтобы гарантировать соответствие аппаратного обеспечения, сети и ОС каждого узла требованиям. В следующем разделе рассматриваются архитектура платформы, методы установки и объяснение ключевых терминов, чтобы помочь вам усвоить основные моменты в процессе фактической установки.

Содержание

Метод установки

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

1. **Этап подготовки**

   - **Проверка предварительных условий**: Убедитесь, что аппаратное обеспечение, сеть и ОС всех узлов соответствуют требованиям, таким как версия ядра, архитектура CPU и конфигурация сети.
   - **Загрузка установочного пакета**: Войдите в Alauda Customer Portal, чтобы получить последний установочный пакет.
   - **Предварительная обработка узлов**: Выполните подготовительные работы для всех узлов.

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 и выберите режим IP-протокола (IPv4/IPv6/dual stack) и конфигурацию VIP в соответствии с реальной средой.
   - **Настройка 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 кластера `global`.

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.

В последующих главах будут подробно рассмотрены операции, параметры конфигурации и методы проверки каждого этапа установки. Пожалуйста, внимательно ознакомьтесь и выполните соответствующую подготовительную работу перед официальной установкой.

Приложение — Alauda Customer Portal

Alauda Customer Portal — это единая платформа обслуживания клиентов и доставки от 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

< Term name="company" /> Customer Portal 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 функции

- **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 **Alauda Customer Portal** с использованием авторизованной учетной записи для загрузки необходимых установочных пакетов и проверки статуса лицензий. Для клиентских сред доставки и производственных сред версии и документация, опубликованные на Alauda Customer Portal, всегда должны рассматриваться как **официальная базовая линия** для развертывания и обслуживания.

Подготовка к установке

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.
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

В этом разделе приведены рекомендации по планированию ресурсов перед установкой ACP. Выберите подходящий сценарий развертывания в зависимости от вашей среды и потребностей в использовании 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. ACP поддерживает следующие три

распространённые архитектуры развертывания:

- **Мультикластерная**
  Выберите эту архитектуру, если необходимо централизованно управлять несколькими кластерами Kubernetes. В этом режиме ACP состоит из одного **глобального кластера** и нескольких **кластеров рабочих нагрузок**. Запуск рабочих нагрузок, не относящихся к платформе, на глобальном кластере может ухудшить стабильность и производительность платформы и должен быть избегнут.
- **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 кластера рабочих нагрузок, поэтому требует **больше ресурсов** по сравнению с чисто глобальной установкой в режиме мультикластера.
- **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	Ёмкость хранилища

Одиночный кластер

В этом режиме глобальный кластер выполняет функции как управляющей плоскости, так и кластера рабочих нагрузок. 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	Ёмкость хранилища

Для оценки дополнительных ресурсов, необходимых для ваших рабочих нагрузок, обратитесь к разделу **Оценка ресурсов для кластера рабочих нагрузок**

Мультикластер

При управлении несколькими кластерами рабочих нагрузок 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 разделу **Оценка ресурсов для глобального кластера**

Сеть

Перед установкой убедитесь, что необходимые сетевые ресурсы подготовлены.

Если в вашей среде имеется аппаратный LoadBalancer, **рекомендуется** использовать его. Если нет, можно включить `Self-built VIP`, который обеспечивает программное балансирование нагрузки с помощью keepalived.

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

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 веб-интерфейсу платформы. Кластеры рабочих нагрузок в той же сети, что и `global` кластер, также могут обращаться к `global` кластеру через этот IP.
Внешний IP	Опционально	По необходимости	Если есть кластеры рабочих нагрузок, находящиеся в другой сети, например, в гибридном облаке, этот IP обязателен. Кластеры рабочих нагрузок в других сетях обращаются к `global` кластеру через этот IP. Этот IP необходимо настроить в устройстве балансировки нагрузки для обеспечения высокой доступности. Этот IP также может использоваться как адрес

Ресурс	Обязательность	Количество	Описание
			доступа к веб-интерфейсу платформы.
Доменное имя	Рекомендуется	По необходимости	Рекомендуется для Endpoint кластера и адреса доступа к платформе. Пожалуйста, предоставьте его заранее и убедитесь в корректности разрешения доменного имени. Использование доменного имени рекомендуется, поскольку при необходимости изменения VIP после установки кластера вы сможете легко обновить запись DNS без влияния на работу кластера. Доменное имя **обязательно** в следующих случаях: • 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;

Ресурс	Обязательность	Количество	Описание
			если не предоставлен, установщик сгенерирует самоподписанный сертификат, но при использовании HTTPS могут возникать риски безопасности.

ПРИМЕЧАНИЕ

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

Настройка сети

Тип	Описание требования
Пропускная способность сети	Пропускная способность внутри кластера должна быть ≥1 Гбит/с (рекомендуется 10 Гбит/с). Пропускная способность между кластерами должна быть ≥100 Мбит/с (рекомендуется 1 Гбит/с). Недостаточная пропускная способность может значительно ухудшить производительность запросов данных.
Задержка сети	Задержка внутри кластера должна быть ≤10 мс. Задержка между кластерами должна быть ≤100 мс (рекомендуется ≤30 мс).
Сетевая политика	Пожалуйста, ознакомьтесь с разделом Правила переадресации LoadBalancer для обеспечения открытия необходимых портов.
Диапазон IP-адресов	Узлы кластера `global` должны избегать использования сетевого сегмента 172.17-18. Если он уже используется,

Тип	Описание требования
	настройте конфигурацию nerdctl (добавьте параметр bip), чтобы избежать конфликтов.

Правила переадресации LoadBalancer

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

Исходный IP	Протокол	IP назначения	Порт назначения	Описание
`global` VIP, Внешний IP	TCP	Все IP узлов управляющей плоскости	443	Обеспечивает доступ к веб-интерфейсу платформы, репозиторию образов и Kubernetes API Server через протокол HTTPS. Порт по умолчани — `443`. Если требуется использовать нестандартный HTTPS порт, выполните следующие действия: • Замените порт назначения в правиле переадресаци

Исходный IP	Протокол	IP назначения	Порт назначения	Описание
				на ваш пользовательс порт. • Позже в параметрах установки веб-интерфейса укажите ваш пользовательс порт.
`global` VIP, Внешний IP	TCP	Все IP узлов управляющей плоскости	6443	Этот порт обеспечивает доступ к Kubernet API Server для уз внутри кластера.
`global` VIP, Внешний IP	TCP	Все IP узлов управляющей плоскости	11443	Этот порт обеспечивает доступ к репозиторию образов для узло внутри кластера. **Примечание:** Ес вы планируете использовать внешний репозиторий образов вместо репозитория по умолчанию, предоставляемог

Исходный IP	Протокол	IP назначения	Порт назначения	Описание
				кластером `globa` настройка этого порта не требуетс

СОВЕТ

- Рекомендуется настроить проверки работоспособности на LoadBalancer для мониторинга состояния 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

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

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 второстепенные версии (`x` и `y`) должны строго соответствовать проверенным версиям. Изменение `x` или `y` не поддерживается официально.
- **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` не поддерживается официально.

> **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 вызвать **сбои 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

Скрипт `init.sh` не гарантирует, что все перечисленные ниже проверки будут выполнены корректно. Вам 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`.
  - ✅ Проверьте, активированы 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), проверьте, активированы 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 активированы.
  - ✅ Отключите apparmor/selinux и брандмауэр.
  - ✅ Отключите `swap`.
- **Пользователи и права**

- ✅ SSH-пользователь узла должен иметь права `root` и возможность 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 последовательностей `.-`, `..` 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 (отключите systemd-resolved).
  - ⚠️ В файле `/etc/resolv.conf` не должно быть настроек поисковых доменов (если необходимо настроить, смотрите 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, где расположен установщик:** `8080`, `12080`, `12443`, `16443`, `2379`, `2380`, `6443`, `10249` ~ `10256`
    - **Рабочие узлы:** `10249` ~ `10256`
  - ✅ Если в кластере используется **Kube-OVN** или **Calico**, убедитесь, что следующие порты не заняты:

  - **Kube-OVN:** `6641`, `6642`
  - **Calico:** `179`
- ⚠️ Убедитесь, что IP-адреса в сетевом сегменте `172.17.x.x` ~ `172.18.x.x`, необходимых для nerdctl, не заняты. Если IP-адреса в этом сегменте заняты и изменить их нельзя, 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 использование локального хранилища **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 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 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/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 подтвердите наличие конфликтующего программного обеспечения. 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 \
    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-ap
iserver \
    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-ap
iserver \
    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 ku
bernetes-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 ku
bernetes-kubeadm \
    ; do
    yum remove "$x"
done
```

Настройка поискового домена

В Linux ОС файл `/etc/resolv.conf` используется для настройки разрешения доменных имён DNS-клиентом. 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:** длина каждого доменного имени не должна превышать 253 символа.
- **Общая длина:** суммарное 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 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

1 Загрузка и распаковка установочного пакета

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

```
# 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, затем следуйте разделу Extend для загрузки и установки расширений.

2 Запуск установщика

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

Через примерно 5 минут вы сможете использовать браузер на вашем ПК для доступа к веб-консоли, предоставленной 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 будет поддерживать 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 подробные описания ключевых параметров. Пожалуйста, внимательно ознакомьтесь и настройте их в соответствии с реальными потребностями.

4 Проверка успешной установки

После завершения установки на странице будет отображён URL доступа к платформе. Нажмите кнопку **Access**, чтобы открыть веб-интерфейс платформы и проверить 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** 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 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	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 Endpoint	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 предоставленного LoadBalancer. 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 развертывания на одном узле 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 Endpoint** и **Platform Access Address**, введите тот же адрес, что и для **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, активируйте его 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, вы можете добавить их в **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 сеть контейнеров и сеть 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 могут быть настроены как `Platform Exclusive`, то есть запускать только компоненты платформы, обеспечивая изоляцию платформенных и прикладных 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`, параметр `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 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, чтобы принудительно удалить контейнер установщика:

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

Дополнительные ресурсы

- Загрузка и установка расширений

Восстановление после катастрофы для глобального кластера

Содержание

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 этом решении развертываются 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 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, вызванная сбоями системы хранения (вне области синхронизации etcd);

Роли **Primary Cluster** и **Standby Cluster** являются относительными: кластер, обслуживающий платформу в данный момент, считается Primary Cluster (DNS указывает на него), а резервный кластер — Standby Cluster. 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, пользователям необходимо 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 из них;
- Если в основном кластере развернут **Alauda DevOps Eventing v3** (knative-operator) и его экземпляры, то те же компоненты должны быть предварительно развернуты в резервном кластере.

Обзор процесса

1. Подготовить единое доменное имя для доступа к платформе;
2. Указать домен на VIP **Primary Cluster** 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 Standby Cluster;
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.

Необходимые 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` к управляющим узлам за соответствующим VIP.

Процедура

Шаг 1: Установка Primary Cluster

> **ПРИМЕЧАНИЯ ПО УСТАНОВКЕ DR (Окружение восстановления после катастрофы)**
>
> При установке основного кластера окружения DR,

- В первую очередь необходимо задокументировать все параметры, установленные при следовании руководству веб-интерфейса установки. Некоторые опции должны быть одинаковыми при установке резервного кластера.
- Необходимо предварительно настроить **Load Balancer с пользовательской конфигурацией** 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 запросить 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 Standby Cluster

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 IP на управляющие узлы 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 окружения DR 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. При необходимости настройте балансировщик нагрузки для перенаправления порта `2379` на управляющие узлы соответствующего кластера. Поддерживается ТОЛЬКО режим TCP; перенаправление на уровне L7 не поддерживается.

   > **INFO**
   >
   > Перенаправление порта через балансировщик нагрузки не обязательно. Если резервный кластер имеет 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 перенаправляется через балансировщик, необходимо корректно указать **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 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=$(kubectl get svc -n cpaas-system etcd-sync-monitor -o jsonpat
h='{.spec.clusterIP}')
ipv6_regex="^[0-9a-fA-F:]+$"
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/packaged-scripts/for-upgrade/ensure-asm-temp
late.sh в /root:
# НЕ пропускайте этот шаг

# при необходимости переключитесь на пользователя root
sudo -i

# проверьте, установлен ли Log Storage для Elasticsearch в глобальном к
ластере
_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

    # Перезапуск 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 Primary Cluster;
6. Проверьте разрешение DNS:

```
kubectl exec -it -n cpaas-system deployments/sentry -- nslookup <platfo
rm access domain>
# Если разрешение некорректно, перезапустите pod-ы coredns и повторяйте
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,v
mselect,vminsert)'
```

- cluster-transformer:

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

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

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

10. На исходном Primary Cluster повторите шаги из раздела Включение синхронизации etcd, чтобы превратить 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='{.s
pec.clusterIP}')/check
```

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

Загрузка пакетов

Подробности о подкоманде `violet push` см. в разделе Upload Packages.