Кластеры

Обзор

Обзор

Неподвижная инфраструктура

Неподвижная инфраструктура

Управление узлами

Обзор

Ноды являются основными строительны
могут быть как виртуальными машинам
содержит необходимые компоненты дл
Container Runtime.

Добавление узлов в локальны

Администраторы платформы могут доба
под управлением платформы.

Управлени

Поддержка об
узлов.

Мониторин

Просмотр данн

Управляемые кластеры

обзор **Импорт кластеров** **Регистраци**

Инициализация кластера в пу **Как сделать**

Инициализация кластера в публичном об

Создание локального кластера

Создание локального кластера

Хостингованная контрольная плоскость

Хостингованная контрольная плоскость

Планирование узлов кластера

Планирование узлов кластера

Шифрование etcd

Шифрование etcd

Как сделать

Добавление внешнего адреса

Оптимизация производительны менеджера

Обновлени

Обзор кластеров

Платформа поддерживает несколько моделей управления Kubernetes-кластерами в зависимости от того, как предоставляется базовая инфраструктура и как развернут control plane.

Содержание

Инфраструктура, предоставляемая платформой

Описание:

В этой модели платформа предоставляет как машины, так и операционные системы узлов. Все узлы используют **Immutable OS**, что обеспечивает согласованное, декларативное и легко восстанавливаемое состояние инфраструктуры. Эта модель обеспечивает полную автоматизацию на протяжении всего жизненного цикла кластера — от предоставления ресурсов до масштабирования и обновлений.

Примеры Immutable OS:

К распространенным примерам Immutable OS относятся **Fedora CoreOS**, **Flatcar Linux** и **openSUSE MicroOS**. В настоящее время платформа поддерживает **MicroOS** для управления неизменяемыми узлами.

Роли:

Компонент	Управляется
Машины / узлы	Платформа
ОС узла	Платформа (только Immutable OS)
Kubernetes	Платформа

Инфраструктура, предоставляемая пользователем

Описание:

В этой модели пользователь предоставляет заранее подготовленные физические или виртуальные машины. Платформа устанавливает и управляет Kubernetes на этих узлах, а управление ОС узлов — включая предоставление, установку исправлений или замену — остается под контролем пользователя.

Эта модель предназначена для организаций, у которых уже есть устоявшиеся процедуры или инструменты автоматизации для управления инфраструктурой или операционными системами.

Роли:

Компонент	Управляется
Машины / узлы	Пользователь
ОС узла	Пользователь
Kubernetes	Платформа

Hosted Control Plane (HCP)

Описание:

Hosted Control Plane (HCP) — это модель развертывания, в которой несколько кластеров используют один control plane, размещенный в выделенном управляющем кластере. Общими являются только компоненты control plane — worker-узлы по-прежнему предоставляются в соответствии с одной из двух моделей инфраструктуры выше (либо платформой, либо пользователем).

Характеристики:

- Снижает потребление ресурсов control plane.
- Поддерживает смешанные модели: worker-узлы могут быть неизменяемыми или предоставляться пользователем.
- Идеально подходит для крупных bare-metal-окружений или сред с ограниченными ресурсами.

Подключенные кластеры

Платформа также поддерживает подключение и управление существующими Kubernetes-кластерами, будь то кластеры в публичном облаке или дистрибутивы Kubernetes, соответствующие требованиям CNCF.

Kubernetes в публичном облаке

- Подключается к управляемым сервисам Kubernetes, таким как EKS, AKS и GKE, через облачно-специфичные провайдеры (например, *Alauda Container Platform EKS Provider*).
- Учетные данные облака можно безопасно хранить в платформе.
- Позволяет создавать и управлять кластерами в публичном облаке напрямую из платформы.

Kubernetes, соответствующий требованиям CNCF

- Подключает любой существующий Kubernetes-кластер, соответствующий стандартам CNCF.
- Поддерживает унифицированную видимость, управление политиками и мониторинг во всех средах.
- См. Матрицу поддержки Kubernetes.

Подключение через Tunnel

- Когда **Global cluster** не может напрямую получить доступ к **Workload cluster**, **Tunnel Server** (со стороны global) и **Tunnel Agent** (со стороны workload) устанавливают защищенное соединение.
- Подходит для отключенных или сетевых сред с ограничениями.

Выбор подходящей модели

Сценарий	Инфраструктура предоставляется	ОС узла управляется	Kubernetes управляется	Уровень автоматиз
Инфраструктура, предоставляемая платформой	Платформа	Платформа (только Immutable OS)	Платформа	Полный

Сценарий	Инфраструктура предоставляется	ОС узла управляется	Kubernetes управляется	Уровень автоматиз
Инфраструктура, предоставляемая пользователем	Пользователь	Пользователь	Платформа	Частичный
Hosted Control Plane (HCP)	Платформа	Общие узлы (Платформа)	Платформа	Частичный
Подключенный кластер (облако или CNCF)	Внешний провайдер	Внешний провайдер	Частично / внешняя	Минимальн

Совместимость версий

При импорте или подключении существующих кластеров проверьте версию Kubernetes на соответствие текущей политике совместимости ACP.

ACP 4.3 и более поздние версии

- ACP 4.3 добавляет поддержку Kubernetes 1.34 для сценариев кластеров, управляемых платформой.
- При обновлении до ACP 4.3 workload-кластеры должны оставаться в пределах совместимого диапазона версий 1.34, 1.33, 1.32 и 1.31 до обновления кластера `global`.
- Для сторонних кластеров ACP 4.3 принимает для управления версии Kubernetes в диапазоне `>=1.19.0 <1.35.0`.
- В документации продукта по-прежнему перечисляются только те версии Kubernetes, которые прошли проверку продукта для поддержки сторонних кластеров и базовой линии Extend по умолчанию.
- Проверка продукта для базовой линии Extend охватывает следующие области возможностей:
  - Установка и использование Operators

  - Установка и использование Cluster Plugins
  - Логирование на основе ClickHouse
  - Мониторинг на основе VictoriaMetrics
- Это не означает, что все конкретные Operators или Cluster Plugins охвачены проверкой продукта.
- Для конкретных Operators или Cluster Plugins, выходящих за рамки этой базовой линии, см. соответствующую документацию продукта или обратитесь в техническую поддержку.
- В ACP 4.3 и более поздних версиях workload-кластеры больше не должны находиться на единственном последнем совместимом minor-релизе Kubernetes перед обновлением кластера `global`.

ACP 4.2 и более ранние версии

- Обновите workload-кластеры до последней документированной совместимой версии Kubernetes перед обновлением кластера `global`.
- Используйте Матрицу поддержки Kubernetes в качестве основного источника для сопоставления документированных версий.

О неподвижной инфраструктуре

Неподвижная инфраструктура использует неподвижную операционную систему для развертывания Kubernetes кластеров. В отличие от традиционных кластеров на базе ОС, все конфигурации узлов встроены в образы и остаются неизменными после развертывания. Обновления кластера и изменения конфигурации выполняются путем замены узлов новыми образами, что обеспечивает согласованность, надежность и упрощает операции на протяжении всего жизненного цикла кластера.

Note

Поскольку выпуски Immutable Infrastructure осуществляются в ином режиме, чем у Alauda Container Platform, документация Immutable Infrastructure теперь доступна в виде отдельного набора по адресу Immutable Infrastructure ↗。

Управление узлами

Обзор

Ноды являются основными строительны
могут быть как виртуальными машинам
содержит необходимые компоненты для
Container Runtime.

Добавление узлов в локальны

Администраторы платформы могут доба
под управлением платформы.

Управлени

Поддержка обн
узлов.

Мониторин

Просмотр данн

Обзор

Ноды являются основными строительными блоками кластера. Ноды, добавляемые в кластер, могут быть как виртуальными машинами, так и физическими серверами. Каждый нод содержит необходимые компоненты для запуска Pod'ов, включая Kubelet, Kube-proxy и Container Runtime.

Пользователи с правами управления платформой могут управлять нодами в рамках кластеров.

Примечание: Добавление нод в импортированные кластеры или удаление нод из импортированных кластеров не поддерживается.

Содержание

Типы нод

- **Ноды управляющей плоскости (Control Plane Nodes)**: Отвечают за запуск компонентов кластера, таких как kube-apiserver, kube-scheduler, kube-controller-manager, etcd, контейнерная сеть и некоторые компоненты управления платформой.

  - При разрешении развертывания приложений на нодах управляющей плоскости, эти ноды могут также выполнять функции вычислительных нод.
  - Необходимо добавить минимум 1 нод управляющей плоскости. Настройка 2 нод управляющей плоскости не поддерживается. При наличии 3 и более нод управляющей плоскости кластер становится высокодоступным (для высокодоступных кластеров рекомендуется использовать нечетное количество нод, предпочтительно 3 или 5).
  - При количестве нод управляющей плоскости 3 и более, кластер обладает возможностями многократного резервирования и считается высокодоступным.
- **Вычислительные ноды (Compute Nodes)**: Отвечают за размещение бизнес Pod'ов, работающих в кластере. Количество вычислительных нод, необходимое в кластере, обычно планируется исходя из объема бизнеса.

Проверка доступности Linux-нода

Если необходимо построить on-premises кластер, пожалуйста, сначала ознакомьтесь с cluster check, чтобы убедиться, что все конфигурации нод соответствуют требованиям. Все предварительные условия должны быть выполнены, иначе развертывание кластера может завершиться неудачей.

Поддерживаемые операционные системы и модели CPU

Пожалуйста, ознакомьтесь с разделом Supported Operating Systems and CPU Models.

Добавление узлов в локальные кластеры

Когда необходимо масштабировать кластер или заменить аномальные узлы новыми, вы можете добавить узлы управляющей плоскости и вычислительные узлы в существующие **локальные** рабочие кластеры на платформе.

Содержание

Ограничения и условия

- Узлы, которые будут добавлены в кластер, должны быть подготовлены заранее. Пожалуйста, ознакомьтесь с Node Availability Check Reference для подготовки и проверки узлов, которые будут добавлены в кластер. Убедитесь, что все условия выполнены, иначе развертывание кластера может завершиться неудачей.

- Аппаратная архитектура добавляемых узлов должна совпадать с архитектурой кластера.
- Чтобы избежать непредсказуемых ошибок, тип операционной системы добавляемых узлов должен быть одинаковым с другими узлами кластера.
- SSH-порты и данные для аутентификации узлов, добавляемых в одном диалоге **Add Node**, должны быть едиными.
- **Руководство по планированию кластера**: в кластере должен быть как минимум 1 узел управляющей плоскости. Установка ровно 2 узлов управляющей плоскости не поддерживается. При наличии 3 и более узлов управляющей плоскости кластер становится высокодоступным (для высокодоступных кластеров рекомендуется использовать нечетное количество узлов, предпочтительно 3 или 5). **Примечание**: это требование применяется только при добавлении или изменении мощности управляющей плоскости; вы можете безопасно добавлять рабочие/вычислительные узлы без необходимости добавлять узлы управляющей плоскости.
- Узел может принадлежать только одному кластеру. Узлы, которые будут добавлены, не должны быть заняты другими кластерами.

Предварительные условия

- Если глобальный кластер не может напрямую получить доступ к узлам, которые будут добавлены, через SSH и требуется доступ через прокси, подготовьте прокси-сервис заранее. В настоящее время поддерживается только SOCKS5 прокси.

Процедура

1. В левой навигационной панели нажмите **Clusters** > **Clusters**.
2. Нажмите на ***название кластера*** типа **On-Premises**, в который хотите добавить узлы.
3. Во вкладке **Nodes** нажмите **Add Node**.
4. Ознакомьтесь с Node Configuration Parameters для настройки соответствующих параметров.
5. Нажмите **Add** для проверки доступности узлов.
   После успешной проверки начнется добавление узлов, и их статус изменится на

Adding.

Последующие действия

Просмотр прогресса выполнения

На странице списка узлов вы можете просмотреть информацию о добавленных узлах. Для узлов со статусом **Adding** доступен просмотр прогресса выполнения.

Процедура

1. Нажмите **View Execution Progress** справа от узлов со статусом **Adding**.
2. В появившемся диалоговом окне прогресса выполнения можно просмотреть состояние выполнения узла (`status.conditions`).
   Совет: если определенный тип находится в процессе выполнения или имеет состояние ошибки с причиной, вы можете получить подробную информацию о причине (`status.conditions.reason`), наведя курсор на соответствующую причину (отображается синим текстом).

Повторное добавление неудачных узлов

Если после добавления узлов некоторые из них не были добавлены успешно, над списком узлов появится уведомление. Нажмите кнопку **Re-add** в этом уведомлении, чтобы повторно добавить неудачные узлы.

Управление узлами

Содержание

Обновление меток узлов

Метки↗ — это пары ключ-значение, прикрепленные к узлам, которые могут определять атрибуты узлов. После установки меток для узлов вы можете легко фильтровать или выбирать узлы по меткам. Например: направлять Pods для планирования на определённые узлы.

Поддерживается обновление меток узлов для узлов в нормальном состоянии, добавление или удаление пользовательских меток узлов.

Процедура

1. В левой навигационной панели нажмите **Cluster Management** > **Clusters**.
2. Нажмите на ***название кластера***, в котором находится узел, метки которого нужно обновить.
3. Во вкладке **Nodes** справа у узла, метки которого нужно обновить, нажмите **Update Node Labels**.
4. Добавьте, измените или удалите метки узла.
5. Нажмите **OK**.
   После успешного обновления меток узлов изменится количество меток. Вы можете просмотреть всю информацию о метках узла в пункте **Node Labels** в информационной панели **Node**.

Остановка/Возобновление планирования на узле

Устанавливая состояние планирования узлов, вы можете контролировать, разрешено ли новым Pod'ам в кластере планироваться на этот узел.

- **Stop Scheduling**: Новым Pod'ам запрещено планироваться на узел, но уже запущенные Pod'ы на узле не затрагиваются.
- **Resume Scheduling**: Новым Pod'ам разрешено планироваться на узел.

Процедура

1. В левой навигационной панели нажмите **Clusters** > **Clusters**.
2. Нажмите на ***название кластера***, в котором находится узел, для которого нужно остановить или возобновить планирование.
3. Во вкладке **Nodes** справа у узла нажмите **Stop Scheduling/Resume Scheduling** для установки состояния планирования.
4. Нажмите **OK**.

Выселение Pod'ов

Выселите все Pod'ы, кроме управляемых DaemonSet, с узлов в нормальном состоянии на другие узлы кластера и установите узел в состояние, при котором планирование на него запрещено.

Примечание: Данные локально хранящихся Pod'ов будут потеряны после выселения. Пожалуйста, действуйте осторожно.

Процедура

1. В левой навигационной панели нажмите **Cluster Management** > **Clusters**.
2. Нажмите на ***название кластера***, в котором находится узел, с которого нужно выселить Pod'ы.
3. Во вкладке **Nodes** нажмите на ***название узла***, с которого нужно выселить Pod'ы.
4. В правом верхнем углу нажмите **Actions** > **Evict Pods**.
5. Ознакомьтесь с информацией о Pod'ах для выселения, затем нажмите **Evict**.

Установка Taints

Установите информацию о taint для узлов в нормальном состоянии.

Taints — это свойство узлов, позволяющее узлам отказываться запускать определённые типы Pod'ов или даже выселять Pod'ы. Taints работают вместе с tolerations на Pod'ах, чтобы предотвратить назначение Pod'ов на неподходящие узлы. На каждый узел можно

наложить один или несколько taint, и Pod'ы, не способные терпеть эти taint, не будут приняты узлом.

Например: если у узла обнаружена загрузка памяти на уровне 91%, не рекомендуется продолжать планирование новых Pod'ов на этот узел. Можно установить для него taint. После установки taint Kubernetes не будет планировать Pod'ы на этот узел.

Подробнее...↗

Процедура

1. В левой навигационной панели нажмите **Cluster Management** > **Clusters**.
2. Нажмите на ***название кластера***, в котором находится узел, для которого нужно установить taint.
3. Во вкладке **Nodes** справа у узла нажмите **Set Taints**.
4. В соответствии с описанием ниже установите ключ, значение и эффект taint. Можно добавить несколько taint для узла.
   Атрибуты taint состоят из `key=value [effect]`.
   `key=value` используется для сопоставления с tolerations Pod'ов. Taint указывает, что узел загрязнён `key=value`, и планирование Pod'ов на этот узел не разрешено или должно избегаться, если только Pod не может терпеть (Tolerations) этот taint `key=value`.
   effect — эффект taint, с тремя вариантами:
   - **NoSchedule**: Планирование не разрешено, уже запланированные ресурсы не затрагиваются.
   - **PreferNoSchedule**: Рекомендуется не планировать.
   - **NoExecute**: Планирование не разрешено, и уже запланированные ресурсы будут удалены после `tolerationSeconds`.
5. Нажмите **OK**.

Управление метками и taint

Платформа поддерживает пакетную установку меток и taint для узлов.

Ограничения и условия

- Перед установкой меток устройств необходимо сначала развернуть device plugins в кластере, например NVIDIA GPU MPS device plugin, NVIDIA GPU device plugin, GPU Manager device plugin и др.
  Совет: Метки устройств фактически являются метками узлов. Для удобства платформа классифицирует метки узлов, от которых зависят device plugins, как метки устройств для быстрой настройки.

Процедура

1. В левой навигационной панели нажмите **Clusters** > **Clusters**.
2. Нажмите на ***название кластера***, в котором нужно управлять метками и taint.
3. Во вкладке **Nodes** выберите несколько узлов, которыми хотите управлять, и нажмите кнопку **Label and Taint Management**.
   Совет: Вы можете ввести интересующие метки узлов в поле поиска на странице списка узлов, чтобы быстро отфильтровать список узлов для управления метками и taint.
4. В разделе **Batch Operations** добавьте и заполните операции, которые хотите выполнить, затем нажмите OK для отправки пакетных операций в кластер.

   - **Node Labels**: Можно **добавить/обновить** указанные метки для выбранных узлов или **удалить** указанные метки. При выборе удаления платформа отфильтрует все списки меток на выбранных узлах. Если значение установлено в **Any**, это означает удаление меток на всех узлах, содержащих указанный ключ метки.
   - **Taints**: Можно **добавить/обновить** указанные taint для выбранных узлов или **удалить** указанные taint. При выборе удаления платформа отфильтрует все списки taint на выбранных узлах. Если значение установлено в **Any**, это означает удаление taint на всех узлах, содержащих указанный ключ taint.
   - **Device Labels**: Можно установить устройства, которые хотите использовать для выбранных узлов, где список устройств поступает от device plugins, развернутых в этом кластере.

Включение/отключение переключателя виртуализации

Когда узлы в on-premises кластере являются физическими машинами, вы можете управлять разрешением Kubernetes планировать виртуальные машины (VMI, VirtualMachineInstance) на узел, включая или отключая переключатель виртуализации узла.

Когда переключатель включён, новым виртуальным машинам разрешается планироваться на физический узел; когда переключатель отключён, новым виртуальным машинам запрещается планироваться на физический узел, но это не влияет на виртуальные машины, уже запущенные на узле.

Совет: По связанным операциям и мерам предосторожности смотрите Prepare Virtualization Environment.

Удаление узлов on-premises кластера

Поддерживается удаление узлов в кластерах типа on-premises. Например: удаление неработающих узлов в on-premises кластерах.

Ограничения и условия

- Удаление узлов в импортированных кластерах не поддерживается.
- Если в кластере только один узел управляющей плоскости, удаление этого узла не поддерживается.

Процедура

1. В левой навигационной панели нажмите **Cluster Management** > **Clusters**.
2. Нажмите на ***название кластера*** типа **On-Premises**, в котором находится узел для удаления.
3. Во вкладке **Nodes** справа у узла нажмите **Delete**.

Совет: Если после удаления Linux-узла необходимо очистить ресурсы на узле, нажмите **Download Cleanup Script** внизу диалога, чтобы скачать скрипт очистки локально. После успешного удаления узла войдите на узел и выполните скрипт очистки.

4. Введите имя узла, затем нажмите **Delete**.

Мониторинг узлов

Просмотр данных мониторинга узла на странице сведений об узле.

TIP

- Когда в кластере более 1 узла, вы можете нажать на ***текущее имя узла*** в области пути ресурса на странице сведений об узле, чтобы развернуть выпадающий список узлов, затем нажать для выбора узла и быстрого переключения на страницу сведений другого узла.
- Когда для кластера настроены компоненты мониторинга, вы можете просматривать данные мониторинга узла, включая статус работы ресурсов, использование ресурсов и статистику тенденций ресурсов.

Содержание

Процедура

1. В левой навигационной панели нажмите **Clusters** > **Clusters**.
2. Нажмите на ***имя кластера***, в котором находится целевой узел.

3. Во вкладке **Nodes** нажмите на целевое ***имя узла***.
4. Нажмите на вкладку **Monitoring**, чтобы перейти на страницу отображения данных мониторинга узла и просмотреть соответствующие данные мониторинга узла.

> **TIP**
>
> - Наведите курсор на карточку и нажмите на значок **Details**, чтобы просмотреть выражения PromQL; нажмите на значок **Export**, чтобы экспортировать выражения PromQL для всех графиков на текущей странице.
> - Когда в кластере более 1 узла, вы можете нажать на ***текущее имя узла*** в области пути ресурса на странице сведений об узле, чтобы развернуть выпадающий список узлов, затем нажать для выбора узла и быстрого переключения на страницу сведений другого узла.

> **TIP**
>
> В области отображения статистики пространства хранения, когда у узла более 4 разделов хранения:
>
> - В круговой диаграмме общего использования разделов отдельно отображаются 3 раздела с наибольшим использованием, а оставшиеся разделы показаны как **Others** с отображением их общего использования при наведении курсора;
> - В столбчатой диаграмме использования разделов отдельно отображаются 3 раздела с наибольшим использованием, а оставшиеся разделы показаны как **Others** с отображением их общего использования и индивидуальных показателей при наведении курсора на столбцы.

Статистика тенденций мониторинга описана в следующей таблице.

Параметр	**Описание**
CPU	**Уровень использования**, **уровень запросов** и **уровень лимитов** CPU за указанный период времени.

Параметр	Описание
	Уровень использования = использование CPU всеми подами на узле / общий CPU узла. **Примечание**: Если в определённый период наблюдается резкий рост уровня использования CPU узла, необходимо сначала определить процесс, потребляющий наибольшее количество ресурсов CPU. Например, для Java-приложений с пользовательским кодом утечки памяти или бесконечные циклы могут вызывать высокое использование CPU. **Уровень запросов** = запросы CPU всех подов на узле / общий CPU узла. **Примечание**: Если в определённый период наблюдается резкий рост уровня запросов CPU узла, это может быть связано с неправильными настройками коэффициента over-subscription кластера или чрезмерно высокими значениями запросов для подов, работающих на узле, что может привести к перерасходу ресурсов. **Уровень лимитов** = лимиты CPU всех подов на узле / общий CPU узла. **Примечание**: Если в определённый период наблюдается резкий рост уровня лимитов CPU узла, это указывает на слишком высокие значения лимитов для подов, работающих на узле, что может привести к перерасходу ресурсов CPU.
Memory	**Уровень использования**, **уровень запросов** и **уровень лимитов** памяти за указанный период времени. **Уровень использования** = использование памяти всеми подами на узле / общий объём памяти узла. Память является одним из важных компонентов сервера и служит связующим звеном для коммуникации CPU. Поэтому производительность памяти существенно влияет на работу машины. При запуске программ загрузка данных, параллелизм потоков и буферизация ввода-вывода зависят от памяти. Доступный объём памяти определяет, могут ли программы

Параметр	Описание
	работать нормально и как именно. **Уровень запросов** = запросы памяти всех подов на узле / общий объём памяти узла. **Примечание**: Если в определённый период наблюдается резкий рост уровня запросов памяти узла, это может быть связано с неправильными настройками коэффициента over-subscription кластера или чрезмерно высокими значениями запросов для подов, работающих на узле, что может привести к перерасходу ресурсов. **Уровень лимитов** = лимиты памяти всех подов на узле / общий объём памяти узла. **Примечание**: Если в определённый период наблюдается резкий рост уровня лимитов памяти узла, это указывает на слишком высокие значения лимитов для подов, работающих на узле, что может привести к перерасходу ресурсов памяти.
Storage	**Уровень использования пространства** и **уровень использования inode** за указанный период времени. **Уровень использования пространства** = использованное пространство хранения / общее пространство хранения. Мониторинг исторических данных использования дискового пространства позволяет оценить использование диска за заданный период времени. При высоком использовании диска можно освободить место, очистив ненужные образы или контейнеры. **Уровень использования inode** = использованные inode / общее количество inode. **Примечание**: Каждый файл должен иметь inode для хранения метаданных файла, таких как создатель файла и дата создания. Inode также занимают место на диске, и большое количество мелких файлов кэша может привести к исчерпанию ресурсов inode. Кроме того, при исчерпании inode, но при наличии свободного места на диске, создание новых файлов становится невозможным.

Параметр	Описание
System Load	Средняя загрузка CPU за 1, 5 и 15 минут. Значение представляет собой отношение общего числа процессов, которые в данный момент выполняются CPU или ожидают выполнения CPU, к максимальному числу процессов, которые CPU может выполнить, что является важным показателем занятости/простоя системы. **Примечание**: Если кривые за 1, 5 и 15 минут схожи в течение определённого периода, это указывает на относительно стабильную загрузку CPU кластера. Если значение за 1 минуту значительно выше значения за 15 минут в определённый период или точку времени, это указывает на рост нагрузки за последнюю минуту и требует дальнейшего наблюдения. Если значение за 1 минуту превышает количество CPU, это может свидетельствовать о перегрузке системы. Необходимо провести дополнительный анализ причин. Если значение за 1 минуту значительно ниже значения за 15 минут в определённый период или точку времени, это указывает на снижение нагрузки за последнюю минуту после высокого уровня нагрузки в предыдущие 15 минут.
Disk Throughput	Пропускная способность диска за указанный период времени — скорость передачи данных диском, где передаваемые данные — это сумма прочитанных и записанных данных.
Disk IOPS	IOPS диска за указанный период времени — сумма операций чтения и записи в секунду, представляющая собой показатель производительности количества операций чтения и записи в секунду диска.
Network Traffic Rate	Скорость входящего и исходящего сетевого трафика за указанный период времени, учитываемая по физическому сетевому интерфейсу узла.

Параметр	Описание
Network Packet Rate (packets/sec)	Скорость приёма и отправки сетевых пакетов за указанный период времени, учитываемая по физическому сетевому интерфейсу узла.

Управляемые кластеры

обзор

обзор

Импорт кластеров

Обзор

Импорт стандартного кластер

Импорт кла

Импорт кластера Amazon EKS

Импорт кластера GKE

Импорт кла

Импорт сущест

Импорт кластера Azure AKS

Импорт кластера Alibaba Cloud ACK

Импорт кластера Tencent Cloud TKE

Регистрация кластера

Регистрация кластера

Инициализация кластера в публичном облаке

Инициализация сети

Инициализация сети кластера в публичн

Инициализация хранилища

Инициализация хранилища кластера в публичном облаке.

Как сделать

Настройка сети для импортир

Получение информации о имі

Доверие не

тев

ых д

Настройка сбора аудита для импортированных стандартных кластеров Kubernetes

Включите аудит Kubernetes API сервера в импортированных стандартных кластерах Kubernetes, чтобы платформа могла собирать данные аудита.

overview

Платформа поддерживает управление существующими стандартными кластерами Kubernetes, OpenShift, Amazon EKS (Elastic Kubernetes Service) и кластерами Huawei Cloud CCE (Cloud Container Engine).

Содержание

Что такое управляемый кластер?

Управляемый кластер — это объединение существующих кластеров в централизованную платформу для единого управления. Это позволяет предприятиям объединять различные типы кластеров — включая стандартные кластеры Kubernetes и некоторые публичные облачные кластеры — под одной контрольной плоскостью. Централизованное управление повышает масштабируемость, доступность и удобство сопровождения, обеспечивая более эффективное использование вычислительных ресурсов и оптимизацию облачной среды. Вы можете подключить кластеры к платформе через **Access a cluster** или **Register a cluster**.

В чем разница между двумя способами подключения?

Они отличаются только способом подключения; повседневные операции при этом остаются одинаковыми.

- **Import a cluster**: Платформа сначала получает информацию о целевом кластере, а затем активно отправляет ему инструкции по доступу. Используя эти данные, платформа устанавливает стабильное соединение для централизованного мониторинга и управления, что помогает администраторам контролировать среду и обеспечивать эффективное и безопасное использование ресурсов.
- **Register a cluster**: В целевом кластере разворачивается обратный прокси, который инициирует запрос на регистрацию на платформе. Кластер с помощью CLI автоматически устанавливает туннель и безопасно взаимодействует с платформой. Поскольку детали кластера не раскрываются, повышается безопасность, а процесс становится проще и эффективнее.

Импорт кластеров

Обзор

Импорт стандартного кластер

Импорт кла

Импорт кластера Amazon EKS

Импорт кластера GKE

Импорт кла

Импорт сущест

Импорт кластера Azure AKS

Импорт кластера Alibaba Cloud ACK

Импорт кластера Tencent Cloud TKE

Обзор

Выберите провайдера для подключения существующего управляемого кластера к платформе.

- Standard Kubernetes
- OpenShift
- AWS EKS
- Google GKE
- Azure AKS
- Alibaba Cloud ACK
- Tencent Cloud TKE

Import Standard Kubernetes Cluster

Поддерживается интеграция стандартных нативных кластеров Kubernetes, развернутых с помощью **kubeadm**, в платформу для единого управления.

Содержание

Терминология

Термин	Описание
Managed Kubernetes Cluster	Тип кластера Kubernetes, предоставляемого облачными провайдерами, где узлы Master и их компоненты управляются провайдером. Пользователи не могут войти или управлять Master-узлами.
Unmanaged Kubernetes Cluster	В отличие от этого, некоторые облачные провайдеры предоставляют кластеры, где пользователи управляют Master-узлами, например Alibaba Cloud ACK Dedicated Edition или Tencent Cloud TKE Independent Cluster.

Предварительные требования

- Kubernetes и связанные компоненты в кластере должны соответствовать требованиям по версиям и параметрам.
- Если используется runtime Containerd, перед интеграцией обновите конфигурацию Containerd, чтобы обеспечить успешное развертывание распределённого хранилища.

Примечания

По умолчанию платформа мониторит трафик NIC, соответствующий шаблону `eth.*|en.*|wl.*|ww.*`. Если у вашей сетевой карты другое именование, обновите конфигурацию после интеграции согласно Сбор данных сети с сетевых карт с пользовательскими именами.

Получение адреса реестра

- Чтобы использовать реестр, развернутый платформой при установке **глобального кластера**, выполните на узле глобального управления:

```
if [ "$(kubectl get productbase -o jsonpath='{.items[].spec.registry.preferPlatformURL}')" = 'false' ]; then
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.data.registryAddress}')
else
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.data.platformURL}' | awk -F // '{print $NF}')
fi
echo "Registry address: $REGISTRY"
```

- Чтобы использовать **внешний реестр**, задайте **REGISTRY** вручную:

```
REGISTRY=<external-registry-address>  # например, registry.example.cn:60080 или 192.168.134.43
echo "Registry address: $REGISTRY"
```

Проверка необходимости дополнительной настройки реестра

1. Выполните проверку поддержки HTTPS с доверенным сертификатом CA:

```
REGISTRY=<registry-address-from-previous-step>

if curl -s -o /dev/null --retry 3 --retry-delay 5 -- "https://${REGISTRY}/v2/"; then
    echo 'Pass: Registry uses a trusted CA certificate. No extra config needed.'
else
    echo 'Fail: Registry does not support HTTPS or uses an untrusted certificate. Follow "Trust Insecure Registry".'
fi
```

2. Если проверка не пройдена, смотрите Как доверять небезопасному реестру?

Получение информации о кластере

См. Как получить информацию о кластере?.

Интеграция кластера

1. В левом меню перейдите в **Cluster Management** > **Clusters**.
2. Нажмите **Import Cluster**.
3. Настройте параметры следующим образом:

Параметр	Описание
Registry	Реестр, в котором хранятся необходимые образы компонентов платформы. Варианты: **Platform Default** (настроен при глобальной установке), **Private Registry** (требуются адрес, порт, имя пользователя, пароль), **Public Registry** (требуется обновление облачных учетных данных).
Cluster Info	Можно ввести вручную или распарсить из файла KubeConfig. Обязательные поля: **Cluster Address**, **CA Certificate** (Base64-декодированный при ручном вводе) и **Authentication** (токен или клиентский сертификат с правами cluster-admin).

4. Нажмите **Check Connectivity**. Платформа проверит сетевой доступ и автоматически определит тип кластера.
5. Если проверка успешна, нажмите **Import** для завершения.
   *Прогресс можно отслеживать через диалог **execution progress** (status.conditions).*
   После интеграции кластер отображается в списке как здоровый.

Сетевая конфигурация

Обеспечьте сетевое взаимодействие между глобальным кластером и импортированным кластером.

Конфигурация после импорта

Если необходимо, чтобы платформа собирала данные аудита с импортированного стандартного кластера Kubernetes, настройте аудит логов API сервера Kubernetes в кластере после импорта. См. Как настроить сбор аудита для импортированных стандартных кластеров Kubernetes?.

FAQ

Почему кнопка «Add Node» отключена?

Для управляемых и неуправляемых кластеров добавление узлов через UI платформы не поддерживается. Добавляйте узлы напрямую или через провайдера.

Какие сертификаты поддерживаются?

1. **Сертификаты Kubernetes**: можно просматривать только сертификаты API Server; остальные сертификаты не поддерживаются и не будут автоматически обновляться.
2. **Сертификаты компонентов платформы**: доступны для просмотра и автоматического обновления.

Какие функции не поддерживаются?

- **Управляемые кластеры**: аудит логи недоступны.
- **Управляемые кластеры**: мониторинг ETCD, Scheduler, Controller Manager не поддерживается (доступны только метрики API Server).
- **Все кластеры**: сертификаты, кроме API Server, не поддерживаются.

Как исправить сбои развертывания распределённого хранилища из-за runtime Containerd?

При использовании Containerd развертывание распределённого хранилища не удаётся, если не настроить Containerd на **всех узлах**:

1. Отредактируйте `/etc/systemd/system/containerd.service`, установите `LimitNOFILE=1048576`.
2. Выполните `systemctl daemon-reload`.
3. Перезапустите Containerd: `systemctl restart containerd`.
4. На управляющих узлах перезапустите поды распределённого хранилища:

```
kubectl delete pod --all -n rook-ceph
```

Импорт кластера OpenShift

Поддерживается интеграция развернутых кластеров OpenShift в платформу для единого управления.

Содержание

Предварительные требования

- Версия Kubernetes и параметры кластера должны соответствовать Стандартным требованиям кластера Kubernetes.
- Для интеграции необходимы команды `kubectl`. Установите CLI-инструмент на bastion-хост, который имеет доступ к кластеру.
- Для обеспечения мониторинга в реальном времени таких метрик, как узлы, рабочие нагрузки (Deployment, StatefulSet, DaemonSet), Pods и контейнеры, убедитесь, что **Prometheus** уже развернут в целевом кластере.

Получение адреса реестра

- Для использования реестра, развернутого платформой при установке **глобального кластера**, выполните следующую команду на глобальном управляющем узле:

```
if [ "$(kubectl get productbase -o jsonpath='{.items[].spec.registry.prefe
rPlatformURL}')" = 'false' ]; then
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.
data.registryAddress}')
else
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.
data.platformURL}' | awk -F // '{print $NF}')
fi
echo "Registry address is: $REGISTRY"
```

- Для использования **внешнего реестра** задайте переменную **REGISTRY** вручную:

```
REGISTRY=<external-registry-address>  # например, registry.example.cn:60
80 или 192.168.134.43
echo "Registry address is: $REGISTRY"
```

Проверка необходимости дополнительной настройки реестра

1. Выполните команду, чтобы проверить, поддерживает ли реестр HTTPS и использует ли доверенный сертификат CA:

```
REGISTRY=<registry-address-from-previous-step>

if curl -s -o /dev/null --retry 3 --retry-delay 5 -- "https://${REGISTRY}/v2/"; then
    echo 'Pass: Registry uses a trusted CA certificate. No extra config needed.'
else
    echo 'Fail: Registry does not support HTTPS or uses an untrusted certificate. Follow "Trust Insecure Registry".'
fi
```

2. Если проверка не пройдена, выполните следующие шаги.

Доверие к небезопасному реестру

1. **Войдите на все узлы кластера OCP.**

2. На каждом узле настройте параметры реестра:

```
sudo -i
sudo chattr -i /

sudo mkdir -p /etc/systemd/system/crio.service.d/
cat | sudo tee /etc/systemd/system/crio.service.d/99-registry-cpaas-system.conf << 'EOF'
[Service]
ExecStart=
ExecStart=/usr/bin/crio \
          --insecure-registry='<registry-address>' \  # например, registry.example.cn:60080 или 192.168.134.43
          $CRIO_CONFIG_OPTIONS \
          $CRIO_RUNTIME_OPTIONS \
          $CRIO_STORAGE_OPTIONS \
          $CRIO_NETWORK_OPTIONS \
          $CRIO_METRICS_OPTIONS
EOF
```

3. Перезапустите `crio`:

```
sudo systemctl daemon-reload && sudo systemctl restart crio
```

Настройка DNS для кластера

Измените ConfigMap CoreDNS в глобальном кластере для настройки DNS.

1. С bastion-хоста получите базовый домен кластера OCP:

```
oc get dns cluster -o jsonpath='{.spec.baseDomain}'
```

   Пример вывода:

```
ocp.example.com
```

2. Войдите в консоль управления платформой, переключитесь на **глобальный** кластер, затем перейдите в **Управление кластерами** > **Управление ресурсами**.
3. Отредактируйте ConfigMap `cpaas-coredns` в пространстве имен `kube-system`. Добавьте новый блок с использованием базового домена OCP и адреса DNS-сервера (из `/etc/resolv.conf` на узле кластера).
   Пример:

```
Corefile: |
ocp.example.com:1053 {
    log
    forward . 192.168.31.220
}
.:1053 {
    log
    forward . 192.168.31.220
}
```

Получение информации о кластере

Выберите один из вариантов:

Метод 1 (рекомендуется): Получение файла KubeConfig

1. На bastion-хосте найдите файл `kubeconfig` и убедитесь, что он содержит контекст администратора.
2. Скопируйте файл kubeconfig с bastion-хоста на локальную машину:

```
scp root@<bastion-ip>:</path/to/kubeconfig> <local-path>
```

Метод 2: Использование токена, адреса API-сервера и сертификата CA

См. Как получить информацию о кластере?.

Импорт кластера

1. В левом меню перейдите в **Управление кластерами** > **Кластеры**.
2. Нажмите **Импортировать кластер**.
3. Настройте параметры:

Параметр	Описание
Registry	Реестр, хранящий образы компонентов платформы.**По умолчанию платформы**: реестр, настроенный при глобальной установке.**Частный реестр**: требуется адрес реестра, порт, имя пользователя и пароль.**Публичный реестр**: требуется обновить учетные данные облака.

Параметр	Описание
Cluster Info	Либо загрузите файл KubeConfig, либо введите вручную.**Адрес кластера**: адрес API-сервера.**Сертификат CA**: декодированный сертификат CA в Base64.**Аутентификация**: токен или клиентский сертификат с правами cluster-admin.

4. Нажмите **Проверить подключение**.
5. Если проверка успешна, нажмите **Импортировать**. Прогресс отображается в журнале выполнения. После импорта кластер отображается в списке со статусом «здоров».

Сетевая конфигурация

Обеспечьте сетевое соединение между глобальным кластером и импортированным кластером. См. Сетевая конфигурация для импортированных кластеров.

Развертывание дополнений

После успешной интеграции перейдите в **Marketplace** для развертывания необходимых дополнений, таких как мониторинг, сбор логов и хранение логов.

Перед развертыванием сбора логов убедитесь, что на `/var/cpaas/` свободно более 50 ГБ:

```
df -h /var/cpaas
```

Обновление политики аудита

Вы можете изменить политику аудита (`spec.audit.profile`) кластера:

- **Default**: логирует метаданные запросов на чтение/запись (при создании OAuth access token логируется тело).

- **WriteRequestBodies**: логирует метаданные всех запросов и тела запросов на запись.
- **AllRequestBodies**: логирует метаданные и тела всех запросов.

Для чувствительных ресурсов (например, Secrets, Routes, OAuthClient) логируются только метаданные.

Обновление выполняется командой:

```
oc edit apiserver cluster
```

FAQ

Почему кнопка «Добавить узел» отключена?

Добавление узлов через UI платформы не поддерживается. Используйте метод поставщика.

Какие сертификаты поддерживаются?

1. **Сертификаты Kubernetes**: виден только сертификат API Server, автоматическая ротация отсутствует.
2. **Сертификаты компонентов платформы**: видны и автоматически ротируются.

Какие функции не поддерживаются для кластеров OpenShift?

- Сбор данных аудита.
- Мониторинг ETCD, Scheduler, Controller Manager (доступны только метрики API Server).
- Сертификаты, кроме API Server.

Импорт кластера Amazon EKS

Подключите существующий кластер Amazon EKS (Elastic Kubernetes Service) к платформе для единого управления.

Содержание

Предварительные требования

- Версия Kubernetes и настройки кластера соответствуют требованиям, описанным в разделе Version compatibility for importing standard Kubernetes clusters.
- Реестр образов должен поддерживать HTTPS и предоставлять действительный TLS-сертификат, выданный публичным центром сертификации.

Подготовка окружения

Для соблюдения практик безопасности AWS EKS выполните следующие шаги в AWS CloudShell.

1. Убедитесь в наличии сетевого подключения к AWS Management Console.
2. Найдите `cloudshell`, затем откройте CloudShell↗.
3. Проверьте, что выбранный регион совпадает с регионом целевого кластера; при необходимости переключитесь.
4. После готовности CloudShell очистите терминал и выполните:

```
# Вывести список кластеров в текущем регионе и проверить права доступа
aws eks list-clusters

# <region-code> — регион кластера, например, us-west-1
# <my-cluster> — имя кластера из предыдущего вывода
aws eks update-kubeconfig --region <region-code> --name <my-cluster>

# Файл kubeconfig сохраняется в "${HOME}/.kube/config"
# Сохраните его содержимое в файл, затем загрузите на платформу для раз
бора
cat "${HOME}/.kube/config"
```

5. Окружение готово. Для последующих шагов, таких как **Получение информации о кластере** и **Импорт кластера**, выполняйте команды для целевого кластера из CloudShell.

Получение информации о кластере

Получение токена для импорта

KubeConfig из кластеров публичного облака нельзя использовать напрямую для импорта.

Обратитесь к разделу How do I get cluster information? для получения токена импорта кластера.

Импорт кластера

1. В левой навигации перейдите в **Cluster Management** > **Clusters**.
2. Нажмите **Import Cluster**.
3. Настройте параметры следующим образом.

Параметр	Описание
Image registry	Реестр, в котором хранятся образы компонентов платформы, необходимые для кластера. **- Platform default**: реестр, настроенный при развертывании глобального кластера. **- Private registry**: заранее подготовленный реестр с необходимыми образами. Укажите **адрес приватного реестра**, **порт**, **имя пользователя** и **пароль**. **- Public registry**: публичный интернет-реестр. Перед использованием получите учетные данные, как описано в разделе Update public registry cloud credentials.
Cluster information	**Совет**: загрузите файл kubeconfig и позвольте платформе автоматически его разобрать. **Cluster endpoint**: внешний адрес API-сервера, предоставляемый целевым кластером. **CA certificate**: сертификат центра сертификации кластера. **Authentication**: используйте **токен**, созданный на предыдущем шаге, с правами **cluster administrator**.

4. Нажмите **Check connectivity** для проверки сетевого подключения и автоматического определения типа кластера. Определенный тип отобразится в виде бейджа в правом верхнем углу формы.

5. После успешной проверки подключения нажмите **Import**, затем подтвердите действие.

   Советы:

   - Для кластеров в состоянии **Importing** нажмите на иконку деталей, чтобы просмотреть ход выполнения в диалоге **Execution progress** (`status.conditions`).
   - После успешного импорта в списке кластеров отображается ключевая информация. Статус кластера — Normal, операции с кластером доступны.

Сетевая конфигурация

Убедитесь, что глобальный кластер и импортированный кластер имеют сетевое соединение. См. Network Configuration for Imported Clusters.

Следующие шаги

Инициализация Ingress и хранилища

Если вам необходимы возможности Ingress и хранилища, ознакомьтесь с разделами Initialize Ingress for AWS EKS и Initialize storage for AWS EKS.

FAQ

Кнопка Add Node отключена после импорта. Как добавить узлы?

Добавление узлов через UI платформы не поддерживается. Пожалуйста, добавляйте узлы через провайдера вашего кластера.

Какие сертификаты поддерживаются управлением сертификатами для импортированных кластеров?

1. **Сертификаты Kubernetes**: доступен только просмотр сертификата API-сервера. Другие сертификаты Kubernetes не видны и не обновляются автоматически.
2. **Сертификаты компонентов платформы**: видны в платформе и поддерживают автоматическое обновление.

Какие функции не поддерживаются для импортированных AWS EKS кластеров?

- Данные аудита недоступны.
- Метрики ETCD, Scheduler и Controller Manager не поддерживаются; доступен лишь поднабор графиков API-сервера.
- Детали сертификатов, кроме сертификата Kubernetes API-сервера, недоступны.

Import GKE Cluster

Платформа поддерживает импорт кластеров Google GKE.

Содержание

Требования

- Версия Kubernetes и компоненты в кластере соответствуют требованиям по версиям для импорта кластеров публичных облаков.
- Убедитесь, что тип кластера — стандартный, и у аккаунта есть права на обслуживание управляющей плоскости. Кластеры Autopilot в настоящее время не поддерживаются.
- Репозиторий образов должен поддерживать доступ по HTTPS и предоставлять действительный TLS-сертификат, подтверждённый публичным центром сертификации.

Подготовка рабочей среды

Для соблюдения стандартов безопасности GKE следующие действия необходимо выполнять с помощью Cloud Shell.

1. Убедитесь в наличии сетевого соединения с Google.
2. Перейдите на страницу **Clusters** page↗ в разделе Kubernetes Engine; найдите импортируемый кластер, откройте его детали и нажмите кнопку **Connect**.
3. В появившемся диалоговом окне скопируйте команду для настройки прав доступа kubectl и нажмите кнопку **Run in Cloud Shell**.
4. Дождитесь готовности Cloud Shell, очистите командную строку, вставьте скопированное содержимое и выполните его.
5. Среда готова. Все последующие команды, выполняемые в среде импортируемого кластера для таких шагов, как **Получение информации о кластере** и **Импорт кластера**, должны выполняться в Cloud Shell.

Получение информации о кластере

Получение адреса API Server и CA-сертификата целевого кластера

1. Перейдите на страницу **Clusters** page↗ в разделе Kubernetes Engine и откройте страницу деталей целевого кластера.

2. Адрес API Server находится в разделе **External endpoints**.
3. Для получения CA-сертификата используйте один из следующих способов в Cloud Shell:

   Способ A: Получить CA-сертификат из kubeconfig:

   ```
   gcloud container clusters get-credentials <cluster-name> --zone <zone>
   kubectl config view --raw -o jsonpath='{.clusters[0].cluster.certificat
   e-authority-data}' | base64 -d
   ```

   Способ B: Получить CA-сертификат напрямую из кластера:

   ```
   gcloud container clusters describe <cluster-name> --zone <zone> --forma
   t='get(masterAuth.clusterCaCertificate)' | base64 -d
   ```

   Примечание: Сертификат необходимо декодировать из Base64 перед вставкой в форму импорта.

Получение токена целевого кластера

Файл KubeConfig публичных облачных кластеров нельзя использовать напрямую для импорта.

Пожалуйста, обратитесь к FAQ Как получить информацию о кластере? для получения токена целевого кластера.

Импорт кластера

1. В левой навигационной панели нажмите **Clusters** > **Clusters**.
2. Нажмите **Manage Cluster** > **Import Cluster**.
3. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Image Repository	Репозиторий для хранения образов компонентов платформы, необходимых кластеру. **- Platform Default**: Репозиторий образов, настроенный при глобальном развертывании. **- Private Repository**: Предварительно созданный репозиторий, хранящий необходимые компоненты платформы. Требуется ввод **Адреса приватного репозитория образов**, **Порта**, **Имени пользователя** и **Пароля** для доступа к репозиторию. **- Public Repository**: Использование публичных сервисов репозиториев образов в интернете. Перед использованием необходимо получить права аутентификации репозитория согласно инструкции Обновление учетных данных публичного репозитория.
Cluster Information	**Информация о кластере**: Включает токен целевого кластера, адрес API Server и CA-сертификат целевого кластера. **Cluster Address**: Адрес доступа, по которому целевой кластер предоставляет API Server для доступа платформы к API Server кластера. **CA Certificate**: CA-сертификат целевого кластера. **Примечание**: При ручном вводе необходимо использовать сертификат, декодированный из Base64. **Authentication Method**: Метод аутентификации целевого кластера, требуется использовать **token** (токен) с **правами управления кластером**, созданный на предыдущем шаге.

4. Нажмите **Check Connectivity** для проверки сетевого соединения с целевым кластером и автоматического определения типа кластера, который будет отображён в виде бейджа в правом верхнем углу формы.
5. После успешной проверки соединения нажмите **Import** и подтвердите.

TIP

- Нажмите на иконку **Details** справа от кластера со статусом **Importing**, чтобы просмотреть ход выполнения (status.conditions) в всплывающем окне **Execution Progress**.

- После успешного импорта кластера вы сможете видеть ключевую информацию о кластере в списке, статус кластера будет отображаться как нормальный, и вы сможете выполнять операции, связанные с кластером.

Настройка сети

Обеспечьте сетевое соединение между глобальным кластером и импортируемым кластером. См. Настройка сети для импортируемых кластеров.

Действия после импорта

Инициализация Ingress и хранилища

После импорта кластера, если необходимо использовать функции Ingress и связанные с хранилищем, пожалуйста, обратитесь к Конфигурация Ingress контроллера Google GKE и Конфигурация хранилища Google GKE.

Часто задаваемые вопросы

Как добавить узлы, если кнопка «Add Node» неактивна после импорта кластера?

Добавление узлов через интерфейс платформы не поддерживается. Пожалуйста, обратитесь к поставщику кластера для добавления узлов.

Какие сертификаты поддерживаются функционалом управления сертификатами для импортируемых кластеров?

1. **Сертификаты Kubernetes**: Все импортируемые кластеры поддерживают только просмотр информации о сертификате APIServer в интерфейсе управления сертификатами платформы. Другие сертификаты Kubernetes не отображаются, автоматическая ротация не поддерживается.
2. **Сертификаты компонентов платформы**: Все импортируемые кластеры могут просматривать информацию о сертификатах компонентов платформы в интерфейсе управления сертификатами и поддерживают автоматическую ротацию.

Импорт кластера Huawei Cloud CCE (публичное облако)

Импортируйте существующий кластер CCE (Cloud Container Engine) (публичное облако) в платформу для единого управления.

Содержание

Предварительные требования

- Версия Kubernetes и параметры кластера соответствуют Стандартным требованиям к версиям компонентов и параметрам Kubernetes.
- Убедитесь, что тип кластера — Huawei Cloud CCE, и у учетной записи есть права на обслуживание управляющей плоскости. Turbo кластеры в настоящее время не поддерживаются.
- Кластеры Huawei Cloud CCE по умолчанию после создания не имеют доступа к ресурсам внешней сети. Перед импортом кластера убедитесь, что импортируемый кластер может получить доступ к адресу платформы.

Получение адреса реестра образов

- Для использования **развернутого на платформе** реестра образов из глобального кластера выполните следующую команду на **контрольном узле глобального кластера** для получения адреса:

```
if [ "$(kubectl get productbase -o jsonpath='{.items[].spec.registry.pr
eferPlatformURL}')" = 'false' ]; then
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath
='{.data.registryAddress}')
else
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath
='{.data.platformURL}' | awk -F \// '{print $NF}')
fi
echo "Image registry address is: $REGISTRY"
```

- Для использования **внешнего реестра образов** вручную задайте переменную **REGISTRY**.

```
REGISTRY=<external image registry address>  # Корректные примеры: regis
try.example.cn:60080 или 192.168.134.43
echo "Image registry address is: $REGISTRY"
```

Определение необходимости дополнительной настройки реестра образов

1. Выполните следующую команду, чтобы определить, поддерживает ли указанный реестр образов HTTPS-доступ и использует ли сертификаты, выданные доверенными CA:

```
REGISTRY=<image registry address obtained from the "Obtain Image Regist
ry Address" section>

if curl -s -o /dev/null --retry 3 --retry-delay 5 -- "https://${REGISTR
Y}/v2/"; then
    echo 'Тест пройден: реестр образов использует сертификаты, выданные
доверенными CA. Выполнение раздела "Trust Insecure Image Registry" не т
ребуется.'
else
    echo 'Тест не пройден: реестр образов не поддерживает HTTPS или сер
тификат не доверенный. Пожалуйста, ознакомьтесь с разделом "Trust Insec
ure Image Registry" для настройки.'
fi
```

2. Если тест не пройден, обратитесь к FAQ Как доверять небезопасному реестру образов?.

Получение информации о кластере

1. Убедитесь в сетевом соединении с консолью Huawei Cloud.
2. Перейдите на страницу управления кластерами ↗ функции `Cloud Container Engine CCE`; найдите импортируемый кластер и нажмите на его имя для перехода на страницу деталей.
3. Как показано на рисунке ниже, перейдите по навигации: `Информация о кластере - Информация о подключении - kubectl - Конфигурация` и скачайте файл KubeConfig.

Получение токена для импорта кластера

Файл KubeConfig публичных облачных кластеров нельзя использовать напрямую для импорта кластера.

Пожалуйста, обратитесь к FAQ Как получить информацию о кластере? для получения токена импорта кластера.

Импорт кластера

1. В левой навигационной панели нажмите **Управление кластерами** > **Кластеры**.
2. Нажмите **Импортировать кластер**.
3. Настройте параметры, связанные с `Image Registry`, согласно следующим инструкциям.

Параметр	Описание
Image Registry	Репозиторий для хранения образов компонентов платформы, необходимых кластеру. **- По умолчанию платформы**: реестр образов, настроенный при развертывании глобального кластера. **- Частный реестр**: заранее подготовленный реестр, хранящий компоненты, необходимые платформе. Необходимо ввести **адрес частного реестра образов**, **порт**, **имя пользователя** и **пароль** для доступа к реестру. **- Публичный реестр**: использование сервисов реестра образов, расположенных в публичной сети. Перед использованием необходимо сначала получить права аутентификации реестра согласно Обновлению учетных данных публичного реестра образов.
Информация о кластере	**Совет**: загрузите файл KubeConfig для автоматического разбора и заполнения платформой. **Адрес кластера**: адрес доступа к API Server, открытый импортируемым кластером, используется платформой для доступа к API Server импортируемого кластера.

Параметр	Описание
	CA сертификат: сертификат CA импортируемого кластера. **Метод аутентификации**: метод аутентификации импортируемого кластера, который требует использования **токена** с **правами управления кластером**, созданного на предыдущем шаге.

4. Нажмите кнопку `Parse KubeConfig File` и отправьте файл KubeConfig, скачанный на предыдущем шаге. Платформа автоматически разберет и заполнит параметры, связанные с `Информацией о кластере`.
5. Нажмите **Проверить подключение**, чтобы проверить сетевое соединение с импортируемым кластером и автоматически определить тип импортируемого кластера. Тип кластера будет отображен в виде бейджа в правом верхнем углу формы.
6. После успешной проверки подключения нажмите **Импортировать** и подтвердите.
   Советы:
   - Нажмите на иконку
     справа от кластера в статусе **Импортируется**, чтобы просмотреть ход выполнения кластера (status.conditions) во всплывающем окне **Ход выполнения**.
   - После успешного импорта кластера вы можете просмотреть ключевую информацию о кластере в списке кластеров. Статус кластера отображается как нормальный, и доступны операции, связанные с кластером.

Настройка сети

Для обеспечения сетевого взаимодействия между глобальным кластером и импортируемым кластером необходимо ознакомиться с Настройкой сети импортированного кластера.

Последующие операции

Инициализация Ingress (входящих правил) и хранилища

После импорта кластера, если требуется использовать Ingress (входящие правила) и функции, связанные с хранилищем, обратитесь к Инициализации Ingress кластера Huawei Cloud CCE и Инициализации хранилища кластера Huawei Cloud CCE.

FAQ

После импорта кластера кнопка добавления узла недоступна. Как добавить узлы?

Добавление узлов через интерфейс платформы не поддерживается. Пожалуйста, обратитесь к поставщику кластера для добавления узлов.

Какие сертификаты поддерживает функция управления сертификатами для импортированных кластеров?

1. **Сертификаты Kubernetes**: все импортированные кластеры поддерживают только просмотр информации о сертификате APIServer в интерфейсе управления сертификатами платформы. Просмотр других сертификатов Kubernetes и автоматическое обновление не поддерживаются.
2. **Сертификаты компонентов платформы**: все импортированные кластеры могут просматривать информацию о сертификатах компонентов платформы в интерфейсе управления сертификатами и поддерживают автоматическое обновление.

Какие другие функции не поддерживаются для импортированных кластеров Huawei Cloud CCE?

- Получение данных аудита не поддерживается.

- Мониторинг, связанный с ETCD, Scheduler и Controller Manager, не поддерживается. Частичный мониторинг APIServer поддерживается.
- Невозможно получить информацию о сертификатах кластера, кроме сертификатов Kubernetes APIServer.

Импорт кластера Azure AKS

Импортируйте существующий кластер Azure AKS в платформу для единого управления.

Содержание

Предварительные требования

- Версия Kubernetes и параметры кластера должны соответствовать Стандартным требованиям к версиям и параметрам компонентов Kubernetes.

  TIP

  - Если узлы AKS не могут получить доступ к глобальному кластеру, обратитесь к FAQ: Как настроить правила группы безопасности внешнего IP узлов AKS.

- Реестр образов должен поддерживать доступ по HTTPS и предоставлять действительный TLS-сертификат, подтверждённый публичным центром сертификации.

Подготовка рабочей среды

Для соответствия стандартам безопасности Azure AKS следующие действия необходимо выполнять с помощью Cloud Shell.

1. Убедитесь в сетевом подключении к Azure Console.
2. Откройте страницу Kubernetes Services ↗, найдите кластер, который хотите импортировать, и перейдите на страницу обзора кластера.
3. Нажмите кнопку `Connect`, откроется всплывающее окно с заголовком `Connect to <import cluster name>`. Следуйте инструкциям для открытия Cloud Shell и настройки рабочей среды.

Получение информации о кластере

Получение токена для импорта кластера

Файл KubeConfig публичных облачных кластеров нельзя использовать напрямую для импорта кластера.

Пожалуйста, обратитесь к FAQ Как получить информацию о кластере? для получения токена импорта кластера.

Импорт кластера

1. В левой навигационной панели нажмите **Cluster Management** > **Clusters**.
2. Нажмите **Import Cluster**.
3. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Image Registry	Реестр, в котором хранятся образы компонентов платформы, необходимых для кластера. **- Platform Default**: Реестр образов, настроенный при развертывании глобального кластера. **- Private Registry**: Предварительно созданный реестр, в котором хранятся образы компонентов, необходимых платформе. Необходимо ввести **Адрес приватного реестра образов**, **Порт**, **Имя пользователя** и **Пароль** для доступа к реестру образов. **- Public Registry**: Использование публичного реестра образов в интернете. Перед использованием необходимо обратиться к Обновлению облачных учетных данных публичного реестра образов для получения прав аутентификации в реестре.
Cluster Information	**Совет**: Пожалуйста, загрузите файл KubeConfig, платформа автоматически распарсит и заполнит информацию. **Cluster Address**: Адрес доступа к API Server, который предоставляет импортируемый кластер, используется платформой для доступа к API Server импортируемого кластера. **CA Certificate**: CA-сертификат импортируемого кластера. **Authentication Method**: Метод аутентификации импортируемого кластера, требуется использовать **Token** с **правами управления кластером**, созданный на предыдущем шаге, для аутентификации.

4. Нажмите **Check Connectivity** для проверки сетевого подключения с импортируемым кластером и автоматического определения типа кластера. Тип кластера будет

отображён в виде бейджа в правом верхнем углу формы.

5. После успешной проверки подключения нажмите **Import** и подтвердите.

> **TIP**
>
> - Нажмите на иконку **Details** справа от кластера со статусом **Importing**, чтобы просмотреть ход выполнения (status.conditions) в всплывающем окне **Execution Progress**.
> - После успешного импорта кластера вы можете просмотреть ключевую информацию о кластере в списке кластеров. Статус кластера будет отображаться как нормальный, и вы сможете выполнять операции, связанные с кластером.

Настройка сети

Убедитесь, что глобальный кластер и импортируемый кластер имеют сетевое соединение. См. Настройка сети для импортируемых кластеров.

Действия после импорта

Инициализация Ingress (входящих правил) и хранилища

После импорта кластера, если необходимо использовать Ingress (входящие правила) и функции, связанные с хранилищем, обратитесь к Инициализации Ingress для кластера Azure AKS и Инициализации хранилища для кластера Azure AKS.

Часто задаваемые вопросы

Как настроить правила группы безопасности внешнего IP узлов AKS

По умолчанию узлы имеют только внутренние IP. Внешний IP настраивается на фронтенд балансировщике нагрузки (LB), который используется по умолчанию для исходящего трафика. Этот LB контролируется основным аккаунтом AKS. Прямое ручное изменение этой конфигурации может привести к проблемам. Вы можете разрешить трафик через **Kubernetes** > **Properties** > **Infrastructure Resource Group** > **Network Security Group** > **Add Outbound/Inbound All Rules**.

Как получить доступ к узлу AKS

Для просмотра логов системных компонентов, таких как Kubelet, CNI и ядро, необходимо сначала выполнить SSH на узел. Рекомендуется использовать плагин `kubectl-node-shell` вместо назначения публичных IP-адресов каждому узлу.

Вариант 1: Использование kubectl node-shell

Official Link ↗

Вариант 2: Использование debug

Official Link ↗

> **NOTE**
>
> Для этого примера требуется версия kubectl 1.25 или выше, которая включает GA-команду `kubectl debug`.

```
kubectl debug node/aks-newadd-41368356-vmss000002 -it --image=mcr.microso
ft.com/dotnet/runtime-deps:6.0
chroot /host
```

Azure ALB с использованием внутреннего балансировщика нагрузки

См. Official Link↗

```yaml
apiVersion: v1
kind: Service
metadata:
  name: internal-app
  namespace: cpaas-system
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-internal: "true"
spec:
  type: LoadBalancer
  ports:
  - name: http-port
    port: 80
    protocol: TCP
  - name: https-port
    port: 443
    protocol: TCP
  selector:
    service.cpaas.io/name: deployment-aks-alb
    service_name: alb2-aks-alb
```

Azure ALB с использованием внешнего балансировщика нагрузки

Разверните высокодоступный ALB с адресом доступа, настроенным как внешний LB.

```
apiVersion: v1
kind: Service
metadata:
  name: azure-alb
  namespace: cpaas-system
spec:
  type: LoadBalancer
  ports:
  - name: http-port
    port: 80
    protocol: TCP
  - name: https-port
    port: 443
    protocol: TCP
  - name: prom-port
    port: 11780
    protocol: TCP
  - name: prom2-port
    port: 11781
    protocol: TCP
  - name: prom3-port
    port: 15012
    protocol: TCP
  selector:
    service_name: alb2-cpaas-system
```

Если он был развернут заранее, вы можете изменить его с помощью следующей команды.

```
kubectl edit helmrequest -n cpaas-system uat-cluster-aks-alb
```

Кнопка добавления узла неактивна после импорта кластера. Как добавить узлы?

Добавление узлов через интерфейс платформы не поддерживается. Пожалуйста, обратитесь к поставщику кластера для добавления узлов.

Какие сертификаты поддерживает функция управления сертификатами для импортируемых кластеров?

1. **Сертификаты Kubernetes**: Все импортируемые кластеры поддерживают только просмотр информации о сертификате APIServer в интерфейсе управления сертификатами платформы. Другие сертификаты Kubernetes не доступны для просмотра, автоматическая ротация не поддерживается.
2. **Сертификаты компонентов платформы**: Все импортируемые кластеры могут просматривать информацию о сертификатах компонентов платформы в интерфейсе управления сертификатами и поддерживают автоматическую ротацию.

Какие другие функции не поддерживаются для импортируемых кластеров AKS?

- Получение данных аудита не поддерживается.
- Мониторинг, связанный с ETCD, Scheduler и Controller Manager, не поддерживается. Частичный мониторинг APIServer поддерживается.
- Информация, связанная с сертификатами кластера, кроме сертификатов Kubernetes APIServer, не может быть получена.

Импорт кластера Alibaba Cloud ACK

Импортируйте существующие управляемые кластеры Alibaba Cloud ACK (Managed Kubernetes) или выделенные кластеры Alibaba Cloud ACK (Dedicated Kubernetes) для единого управления платформой.

TIP

Для получения информации о продуктах управляемых кластеров ACK (Managed Kubernetes) или выделенных кластеров Alibaba Cloud ACK (Dedicated Kubernetes) обратитесь к официальной документации↗.

Содержание

Какие сертификаты поддерживает функция управления сертификатами для импортированных кластеров?

Какие другие функции не поддерживаются для импортированных **управляемых кластеров Alibaba Cloud ACK** и **выделенных кластеров ACK**?

Предварительные требования

- Версия Kubernetes и параметры кластера должны соответствовать требованиям к версиям компонентов и параметрам для импорта стандартных Kubernetes кластеров.

Получение адреса реестра образов

- Чтобы использовать **развернутый платформой** реестр образов из глобального развертывания кластера, выполните следующую команду на **контрольном узле глобального кластера** для получения адреса:

```
if [ "$(kubectl get productbase -o jsonpath='{.items[].spec.registry.prefe
rPlatformURL}')" = 'false' ]; then
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.
data.registryAddress}')
else
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.
data.platformURL}' | awk -F \// '{print $NF}')
fi
echo "Адрес реестра образов: $REGISTRY"
```

- Чтобы использовать **внешний реестр образов**, вручную задайте переменную **REGISTRY**.

```
REGISTRY=<адрес внешнего реестра образов>  # Примеры: registry.example.cn
:60080 или 192.168.134.43
echo "Адрес реестра образов: $REGISTRY"
```

Определение необходимости дополнительной настройки реестра образов

1. Выполните следующую команду, чтобы проверить, поддерживает ли указанный реестр образов HTTPS-доступ и использует ли сертификаты, выданные доверенными CA:

```
REGISTRY=<адрес реестра образов, полученный из раздела "Получение адреса реестра образов">

if curl -s -o /dev/null --retry 3 --retry-delay 5 -- "https://${REGISTRY}/v2/"; then
    echo 'Тест пройден: реестр образов использует сертификаты, выданные доверенными CA. Выполнение раздела "Доверять ненадежному реестру образов" не требуется.'
else
    echo 'Тест не пройден: реестр образов не поддерживает HTTPS или сертификат не доверенный. Пожалуйста, обратитесь к разделу "Доверять ненадежному реестру образов" для настройки.'
fi
```

2. Если тест не пройден, обратитесь к FAQ Как доверять ненадежным реестрам образов?.

Получение KubeConfig

1. Войдите в консоль управления Alibaba Cloud Container Service.
2. В левой навигационной панели консоли нажмите **Clusters**.
3. На странице **Cluster List** нажмите имя целевого кластера или **Details** в столбце **Actions** справа от нужного кластера.
4. На странице **Cluster Information** перейдите на вкладку **Connection Information**, затем нажмите **Generate Temporary KubeConfig**.
5. В диалоговом окне **Temporary KubeConfig** задайте срок действия временных учетных данных и способ доступа к кластеру (включая доступ через публичную сеть и внутреннюю сеть).

6. Нажмите **Generate Temporary KubeConfig**, затем **Copy**, чтобы скопировать содержимое и сохранить его в файл **KubeConfig** на локальном компьютере.
7. После успешного импорта кластера вы можете отозвать временные учетные данные.

Импорт кластера

1. В левой навигационной панели нажмите **Cluster Management** > **Clusters**.
2. Нажмите **Import Cluster**.
3. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Image Registry	Репозиторий для хранения образов компонентов платформы, необходимых кластеру. **- Platform Default**: реестр образов, настроенный при развертывании глобального кластера. **- Private Registry**: заранее подготовленный реестр, в котором хранятся образы компонентов, необходимых платформе. Необходимо ввести **адрес приватного реестра образов**, **порт**, **имя пользователя** и **пароль** для доступа к реестру. **- Public Registry**: использование публичных сервисов реестров образов в интернете. Перед использованием необходимо получить права аутентификации в репозитории, следуя инструкции Обновление учетных данных публичного репозитория.
Cluster Information	**Совет**: можно заполнить вручную или загрузить файл KubeConfig для автоматического разбора и заполнения платформой. **Разбор файла KubeConfig**: после загрузки файла KubeConfig платформа автоматически распарсит и заполнит **Cluster Information**. Вы можете изменить автоматически заполненную информацию. **Cluster Address**: адрес доступа к внешне доступному API Server кластера, используемый платформой для доступа к API Server кластера. **CA Certificate**: сертификат CA кластера. **Примечание**: при ручном вводе необходимо ввести сертификат в Base64-декодированном виде. **Authentication Method**: метод аутентификации для доступа к кластеру. Для аутентификации необходимо использовать **токен** или **сертификатную**

Параметр	Описание
	аутентификацию (клиентский сертификат и ключ) с **правами управления кластером**.

4. Нажмите **Check Connectivity** для проверки сетевого соединения с импортируемым кластером и автоматического определения типа кластера. Тип кластера будет отображен в виде бейджа в правом верхнем углу формы.
5. После успешной проверки соединения нажмите **Import** и подтвердите.

> **TIP**
>
> - Нажмите на иконку **Details** справа от кластера в статусе **Importing**, чтобы просмотреть ход выполнения (status.conditions) в всплывающем окне **Execution Progress**.
> - После успешного импорта кластера вы сможете просмотреть ключевую информацию о кластере в списке кластеров. Статус кластера будет отображаться как нормальный, и вы сможете выполнять операции, связанные с кластером.

Настройка сети

Обеспечьте сетевое взаимодействие между глобальным кластером и импортируемым кластером. См. Настройка сети для импортированных кластеров.

FAQ

Как решить конфликт портов между мониторингом Alibaba Cloud и компонентами мониторинга платформы?

При совместном использовании встроенного мониторинга Alibaba Cloud и компонентов мониторинга платформы возникают конфликты портов. Рекомендуется удалить мониторинг Alibaba Cloud и оставить только мониторинг платформы.

Как использовать доступ через публичную сеть для кластеров Alibaba Cloud?

Если используется доступ через публичную сеть для кластеров Alibaba Cloud, можно привязать публичный IP-адрес в Alibaba Cloud.

После импорта кластера кнопка добавления узлов неактивна. Как добавить узлы?

Ни **управляемые кластеры Alibaba Cloud ACK**, ни **выделенные кластеры ACK** не поддерживают добавление узлов через интерфейс платформы. Пожалуйста, добавляйте узлы через бекенд или обратитесь к поставщику кластера для добавления.

Какие сертификаты поддерживает функция управления сертификатами для импортированных кластеров?

1. **Сертификаты Kubernetes**: все импортированные кластеры поддерживают только просмотр информации о сертификате APIServer в интерфейсе управления сертификатами платформы. Просмотр других сертификатов Kubernetes и автоматическое обновление не поддерживаются.
2. **Сертификаты компонентов платформы**: все импортированные кластеры могут просматривать информацию о сертификатах компонентов платформы в интерфейсе управления сертификатами платформы и поддерживают автоматическое обновление.

Какие другие функции не поддерживаются для импортированных управляемых кластеров Alibaba Cloud ACK и выделенных кластеров ACK?

- **Управляемые кластеры Alibaba Cloud ACK** не поддерживают получение данных аудита.

- **Управляемые кластеры Alibaba Cloud ACK** не поддерживают мониторинг ETCD, Scheduler, Controller Manager, но поддерживают некоторые графики мониторинга APIServer.
- Ни **управляемые кластеры Alibaba Cloud ACK**, ни **выделенные кластеры ACK** не поддерживают получение информации, связанной с сертификатами кластера, кроме сертификатов Kubernetes APIServer.

Импорт кластера Tencent Cloud TKE

Импортируйте существующие выделенные кластеры Tencent Cloud TKE или управляемые кластеры Tencent Cloud TKE в платформу для их централизованного управления.

TIP

Для ознакомления с продуктом выделенных кластеров TKE или управляемых кластеров Tencent Cloud TKE, пожалуйста, обратитесь к официальной документации ↗.

Содержание

Предварительные требования

- Версия Kubernetes и параметры кластера соответствуют требованиям к версиям компонентов и параметрам для импорта стандартных Kubernetes кластеров.
- Реестр образов должен поддерживать доступ по HTTPS и предоставлять действительный TLS-сертификат, выданный публичным центром сертификации.

Получение адреса реестра образов

- Чтобы использовать **развернутый на платформе** реестр образов, настроенный при глобальном развертывании кластера, выполните следующую команду на **контрольном узле глобального кластера** для получения адреса:

```
if [ "$(kubectl get productbase -o jsonpath='{.items[].spec.registry.prefe
rPlatformURL}')" = 'false' ]; then
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.
data.registryAddress}')
else
    REGISTRY=$(kubectl get cm -n kube-public global-info -o jsonpath='{.
data.platformURL}' | awk -F \// '{print $NF}')
fi
echo "Адрес реестра образов: $REGISTRY"
```

- Чтобы использовать **внешний реестр образов**, вручную задайте переменную **REGISTRY**.

```
REGISTRY=<адрес внешнего реестра образов>  # Примеры допустимых значений:
 registry.example.cn:60080 или 192.168.134.43
echo "Адрес реестра образов: $REGISTRY"
```

Определение необходимости дополнительной настройки реестра образов

1. Выполните следующую команду, чтобы определить, поддерживает ли указанный реестр образов доступ по HTTPS и использует ли сертификат, выданный доверенным центром сертификации:

```
REGISTRY=<адрес реестра образов, полученный из раздела "Получение адреса реестра образов">

if curl -s -o /dev/null --retry 3 --retry-delay 5 -- "https://${REGISTRY}/v2/"; then
    echo 'Проверка пройдена: реестр образов использует сертификат, выданный доверенным ЦС. Выполнение раздела "Доверять небезопасному реестру образов" не требуется.'
else
    echo 'Проверка не пройдена: реестр образов не поддерживает HTTPS или сертификат не доверенный. Пожалуйста, обратитесь к разделу "Доверять небезопасному реестру образов" для настройки.'
fi
```

2. Если проверка не пройдена, обратитесь к FAQ Как доверять небезопасному реестру образов?.

Получение KubeConfig

1. Войдите в панель управления Tencent Cloud Container Service.
2. В разделе **Детали кластера** > **Основная информация** просмотрите информацию **Cluster APIServer**.
3. Выберите **Доступ через Интернет** или **Доступ через внутреннюю сеть** в зависимости от реальной сети клиента, затем скачайте **Kubeconfig** и сохраните его на локальном компьютере.

Импорт кластера

1. В левой навигационной панели нажмите **Управление кластерами** > **Кластеры**.
2. Нажмите **Импортировать кластер**.
3. Настройте соответствующие параметры согласно следующим инструкциям.

Параметр	Описание
Реестр образов	Реестр для хранения образов компонентов платформы, необходимых кластеру. **- По умолчанию платформы**: реестр образов, настроенный при глобальном развертывании. **- Частный реестр**: заранее созданный реестр, в котором хранятся образы компонентов, требуемых платформой. Необходимо ввести **адрес частного реестра образов**, **порт**, **имя пользователя** и **пароль** для доступа к реестру. **- Публичный реестр**: использование сервисов реестра образов, расположенных в публичной сети. Перед использованием необходимо сначала обратиться к Обновлению учетных данных публичного реестра для получения прав аутентификации в реестре.
Информация о кластере	**Совет**: можно заполнить вручную или загрузить файл KubeConfig для автоматического парсинга и заполнения платформой. **Парсинг файла KubeConfig**: после загрузки полученного файла KubeConfig платформа автоматически распарсит и заполнит **Информацию о кластере**, при этом вы сможете изменить автоматически заполненные данные. **Адрес кластера**: адрес доступа к внешне доступному API Server кластера, используемый платформой для доступа к API Server кластера. **CA-сертификат**: CA-сертификат кластера. **Примечание**: при ручном вводе необходимо ввести сертификат в формате Base64 без кодирования. **Метод аутентификации**: метод аутентификации для доступа к кластеру, требуется использовать **токен** (Token) или **аутентификацию по сертификату (клиентский сертификат и ключ)** с **правами управления кластером**.

4. Нажмите **Проверить подключение**, чтобы проверить сетевое соединение с импортируемым кластером и автоматически определить тип импортируемого кластера. Тип кластера отобразится в виде бейджа в правом верхнем углу формы.
5. После успешной проверки подключения нажмите **Импортировать** и подтвердите. **Совет**:

   - Нажмите на иконку **Детали** справа от кластера со статусом **Импортируется**, чтобы просмотреть ход выполнения (status.conditions) кластера во всплывающем окне **Ход выполнения**.

- После успешного импорта кластера вы можете просмотреть ключевую информацию о кластере в списке кластеров. Статус кластера будет отображаться как нормальный, и будут доступны операции, связанные с кластером.

Настройка сети

Обеспечьте сетевое соединение между глобальным кластером и импортируемым кластером. Необходимо обратиться к Настройке сети для импорта кластеров.

FAQ

После импорта кластера кнопка «Добавить узел» неактивна. Как добавить узлы?

Как **выделенные кластеры TKE**, так и **управляемые кластеры TKE** не поддерживают добавление узлов через интерфейс платформы. Пожалуйста, добавляйте узлы через бэкенд или обратитесь к провайдеру кластера для их добавления.

Какие сертификаты поддерживает функция управления сертификатами для импортированных кластеров?

1. **Сертификаты Kubernetes**: все импортированные кластеры поддерживают только просмотр информации о сертификате APIServer в интерфейсе управления сертификатами платформы. Просмотр других сертификатов Kubernetes и автоматическое обновление не поддерживаются.
2. **Сертификаты компонентов платформы**: все импортированные кластеры могут просматривать информацию о сертификатах компонентов платформы в интерфейсе управления сертификатами и поддерживают автоматическое обновление.

Какие другие функции не поддерживаются для импортированных управляемых кластеров TKE и выделенных кластеров TKE?

- **Управляемые кластеры TKE** не поддерживают получение данных аудита.
- **Управляемые кластеры TKE** не поддерживают мониторинг, связанный с ETCD, Scheduler, Controller Manager, но поддерживают частичные графики мониторинга APIServer.
- Ни **управляемые кластеры TKE**, ни **выделенные кластеры TKE** не поддерживают получение информации, связанной с сертификатами кластера, за исключением сертификатов Kubernetes APIServer.

Register Cluster

Это метод развертывания сервиса обратного прокси в управляемом кластере, при котором управляемый кластер активно инициирует запросы на регистрацию на платформе.

Содержание

Требования

- В зависимости от типа управляемого кластера версии и параметры Kubernetes и других компонентов в управляемом кластере должны соответствовать Требованиям к версиям и параметрам управляемых кластеров.
- Реестр образов должен поддерживать доступ по HTTPS и предоставлять действительный TLS-сертификат, подтверждённый публичным центром

сертификации. Если это невозможно, обратитесь к FAQ Как доверять небезопасным реестрам образов?

Примечание: **Публичный реестр**, предоставляемый платформой в публичной сети, уже соответствует требованиям доступа по HTTPS. Вам нужно только проверить, поддерживают ли **Платформенный по умолчанию** и **Частный реестр** доступ по HTTPS.

- Если компонент runtime подключаемого кластера — Containerd, необходимо изменить конфигурацию Containerd перед подключением кластера, чтобы обеспечить успешное развертывание распределённого хранилища.

Важные замечания

Мониторинг трафика сетевых карт платформы по умолчанию распознаёт сетевые карты с именами, соответствующими `eth\.\|en\.\|wl\.*\|ww\.*`. Поэтому, если вы используете сетевые карты с другими именами, пожалуйста, обратитесь к документации Сбор данных с сетевых карт с пользовательскими именами для изменения соответствующих ресурсов после подключения кластера, чтобы платформа могла корректно мониторить трафик сетевых карт.

Регистрация кластера

1. В левой навигационной панели нажмите **Clusters** > **Clusters**.
2. Нажмите **Managed Clusters** > **Register Cluster**.
3. Настройте параметры реестра для хранения образов компонентов платформы, необходимых регистрируемому кластеру, согласно следующим инструкциям.

Параметр	Описание
Platform Default	Реестр образов, используемый при развертывании глобальных компонентов.

Параметр	Описание
Private Registry	Внешний реестр образов, который вы настроили заранее. Необходимо ввести **Адрес частного реестра образов**, **Порт**, **Имя пользователя** и **Пароль** для доступа к реестру образов.
Public Registry	Загрузка необходимых образов через публичный реестр образов, предоставляемый платформой. Необходимо обеспечить доступ вашего кластера к публичной сети. Перед использованием необходимо сначала обратиться к Обновлению учётных данных облачного реестра публичной сети для получения прав аутентификации.

4. Нажмите **Create**, получите команду регистрации на странице **Registration Command** и выполните её в регистрируемом кластере.
   Примечание: Команда регистрации действительна в течение 24 часов. Пожалуйста, получите её заново после истечения срока действия.

Просмотр команды регистрации

Вы можете найти кластер, ожидающий регистрации, в списке кластеров и нажать **View Registration Command**. Пожалуйста, выполните регистрацию до истечения срока действия.

FAQ

Как решить проблему с неудачным развертыванием распределённого хранилища, если компонент runtime подключаемого кластера — Containerd?

Если компонент runtime подключаемого кластера — Containerd, развертывание распределённого хранилища завершится неудачей. Для решения этой проблемы необходимо вручную изменить конфигурацию Containerd на **всех узлах** кластера и перезапустить Containerd.

Примечание: Если вы измените конфигурацию Containerd, следуя приведённым ниже шагам до развертывания распределённого хранилища, выполнять шаг четыре не нужно.

1. Войдите на узел кластера и отредактируйте файл `/etc/systemd/system/containerd.service`, изменив значение параметра `LimitNOFILE` на `1048576`.
2. Выполните команду `systemctl daemon-reload` для перезагрузки конфигурации.
3. Выполните команду `systemctl restart containerd` для перезапуска Containerd.
4. Выполните команду `kubectl delete pod --all -n rook-ceph` на управляющем узле кластера, чтобы перезапустить все Pod в пространстве имён rook-ceph и применить изменения конфигурации.

Инициализация кластера в публичном облаке

Инициализация сети

Конфигурация инициализации сети кластера AWS EKS

Содержание

Обзор поддержки

Функция	Статус поддержки	Требования
LoadBalancer Service	Поддерживается	Опционально развернуть AWS Load Balancer Controller. Без этого контроллера возможности LoadBalancer ограничены.
Ingress	Поддерживается	Опционально развернуть AWS Load Balancer Controller. Опционально включить функциональность **Ingress Class** (после включения вы сможете вручную выбирать классы ingress при создании ingress через интерфейс формы).

Предварительные требования

- Подготовьте две подсети с тегом **kubernetes.io/role/elb**. Для общих подсетей добавьте тег **kubernetes.io/cluster/<cluster-name>: shared**. См. Добавление тегов к подсетям.
- Если вы уже создали кластер EKS, импортируйте кластер Amazon EKS.
- Убедитесь, что инструменты kubectl, Helm, AWS CLI и eksctl доступны перед развертыванием AWS Load Balancer Controller.
  Примечание: После установки инструментов настройте данные для входа, используя пользователя, который создал кластер, через AWS CLI, и проверьте корректность установки AWS CLI и eksctl.
- Заранее получите ACCOUNT_ID, **REGION** и **CLUSTER_NAME**, и замените `<ACCOUNT_ID>`, `<REGION>`, `<CLUSTER_NAME>` в документации на реальные значения.
  Примечание: **ACCOUNT_ID** — это идентификатор аккаунта пользователя, создавшего кластер, **REGION** — регион кластера, **CLUSTER_NAME** — имя кластера.
- Обновите и проверьте конфигурационный файл Kubeconfig.

Шаги конфигурации

Развертывание AWS Load Balancer Controller

Примечание: Подробную информацию о развертывании AWS Load Balancer Controller смотрите в официальной документации↗.

Настройка OIDC Provider

Кластеры Kubernetes используют OpenID Connect (OIDC) для управления идентификацией и связаны с URL-адресом издателя OIDC. Чтобы включить AWS Identity в кластере и разрешить IAM роли для Service Accounts, создайте IAM OIDC Provider, связанный с URL издателя OIDC кластера.

Выполните следующую команду в eksctl для настройки OIDC Provider:

```
eksctl utils associate-iam-oidc-provider --region=<REGION> --cluster=<CLU
STER_NAME> --approve
```

Настройка Service Account

Выполните следующие команды для создания IAM политики и создания Service Account с именем `aws-load-balancer-controller`, связанного с IAM ролью:

```
curl -o aws-load-balancer-controller-iam-policy.json https://raw.githubus
ercontent.com/kubernetes-sigs/aws-load-balancer-controller/v2.4.7/docs/in
stall/iam_policy.json
aws iam create-policy \
    --policy-name <CLUSTER_NAME>-AWSLoadBalancerControllerIAMPolicy \
    --policy-document file://aws-load-balancer-controller-iam-policy.json

eksctl create iamserviceaccount \
  --cluster=<CLUSTER_NAME> \
  --namespace=kube-system \
  --name=aws-load-balancer-controller \
  --role-name AmazonEKSLoadBalancerControllerRole \
  --attach-policy-arn=arn:aws:iam::<ACCOUNT_ID>:policy/<CLUSTER_NAME>-AWS
LoadBalancerControllerIAMPolicy \
  --approve
```

Развертывание AWS Load Balancer Controller в кластере

Выполните следующие команды в eksctl для развертывания AWS Load Balancer Controller:

1. Добавьте репозиторий eks-charts:

```
helm repo add eks https://aws.github.io/eks-charts
```

2. Обновите локальный репозиторий:

```
helm repo update eks
```

3. Разверните Helm Chart AWS Load Balancer Controller в кластере:
   Примечание: `aws-load-balancer-controller` — это **Service Account**, созданный в разделе Настройка Service Account.

```
helm install aws-load-balancer-controller eks/aws-load-balancer-control
ler \
  -n kube-system \
  --version=v2.4.7 \
  --set ingressClassConfig.default=true \
  --set clusterName=<CLUSTER_NAME> \
  --set serviceAccount.create=false \
  --set serviceAccount.name=aws-load-balancer-controller
```

Создание Ingress и LoadBalancer сервисов

Вы можете создавать ingress и LoadBalancer сервисы одновременно или выбрать один из них в зависимости от ваших потребностей.

Создание Ingress

1. В **Container Platform** перейдите в раздел **Network** > **Ingress** в левом меню.
2. Нажмите **Create Ingress** и выберите **EKS Ingress Class** для поля **Ingress Class**.
3. Выберите **Protocol**. По умолчанию — **HTTP**. Для **HTTPS** сначала создайте сертификат и выберите его.

4. Переключитесь в режим **YAML** и добавьте следующие аннотации. Подробнее см. в документации по аннотациям ↗:

```
alb.ingress.kubernetes.io/scheme: internet-facing ## Указать публичный доступ
alb.ingress.kubernetes.io/target-type: ip ## Направлять трафик напрямую к pod
```

5. Нажмите **Create**.

Создание LoadBalancer сервиса

1. В **Container Platform** перейдите в раздел **Network** > **Services** в левом меню.
2. Нажмите **Create Service** и выберите **LoadBalancer** для поля **External Access**.
3. Разверните раздел **annotations** и при необходимости заполните аннотации для LoadBalancer сервиса.
4. Нажмите **Create**.

Связанные операции

Проверка установки AWS CLI и eksctl

- Выполните следующую команду. Если она возвращает список кластеров, AWS CLI установлен корректно:

```
aws eks list-clusters
```

- Выполните следующую команду. Если она возвращает список кластеров, eksctl установлен корректно:

```
eksctl get clusters
```

Получение ACCOUNT_ID

Выполните команду `aws sts get-caller-identity` для получения **ACCOUNT_ID**. В ответе `651168850570` — это **ACCOUNT_ID**:

```
{
  "ARN": "arn:aws:iam::651168850570:user/jwshi"
}
```

Конфигурационный файл Kubeconfig

1. Выполните следующую команду для обновления файла Kubeconfig для указанного региона:

   ```
   aws eks --region <REGION>  update-kubeconfig --name <CLUSTER_NAME>
   ```

2. Выполните следующую команду для проверки файла Kubeconfig. Если информация возвращается корректно, конфигурация верна:

   ```
   kubectl get svc -n cpaas-system
   ```

Добавление тегов к подсетям

1. Выполните следующую команду для получения подсетей кластера:

   ```
   eksctl get cluster --name <CLUSTER_NAME>
   ```

2. Выполните следующую команду для получения деталей подсетей:

   ```
   aws ec2 describe-subnets
   ```

3. Выполните следующие команды для добавления тегов к подсетям. Замените `<subnet-id>` на реальные значения. См. Автоматическое обнаружение подсетей ↗:

   - Добавьте тег `kubernetes.io/role/elb` к подсетям:

```
aws ec2 create-tags --resources <subnet-id> --tags Key=kubernetes.io/
role/elb,Value="1"
```

- Добавьте тег `kubernetes.io/cluster/<CLUSTER_NAME>: shared` к общим подсетям:

```
aws ec2 create-tags --resources <subnet-id> --tags Key=kubernetes.io/
cluster/<CLUSTER_NAME>,Value="shared"
```

Создание сертификата

При использовании протокола HTTPS заранее сохраните учетные данные HTTPS сертификата как Secret (типа TLS).

1. В **Container Platform** перейдите в раздел **Configuration** > **Secrets** в левом меню.
2. Нажмите **Create Secret**.
3. Выберите тип **TLS** и импортируйте или заполните поля **Certificate** и **Private Key** по необходимости.
4. Нажмите **Create**.

Дополнительная информация по AWS EKS

Содержание

Терминология

Аббревиатура	Полное название	Описание
eks-clb	Classic Load Balancer	Стандартный балансировщик нагрузки AWS. Имеет проблемы в некоторых ситуациях и не рекомендуется к использованию.
eks-nlb	Network Load Balancer	Балансировщик нагрузки AWS уровня 4, выполняющий балансировку на уровне TCP/UDP,

Аббревиатура	Полное название	Описание
		подходит для сценариев, требующих более низкоуровневого сетевого контроля.
eks-alb	Application Load Balancer	Балансировщик нагрузки AWS уровня 7. По сравнению с eks-nlb, eks-alb может парсить протоколы HTTP/HTTPS и более интеллектуально распределять запросы, подходит для веб-приложений.
aws-lb	AWS Load Balancer	Балансировщик нагрузки, установленный в Kubernetes, который может автоматически создавать eks-nlb и eks-alb на основе LoadBalancer Services и Ingress в Kubernetes для удовлетворения потребностей балансировки нагрузки приложений.
Platform Load Balancer	-	Собственный балансировщик нагрузки платформы уровня 7.
Service Annotations	-	Метаданные, прикрепляемые к объектам в виде пар ключ-значение. Эта дополнительная информация может быть распознана и использована для улучшения и упрощения управления различными аспектами ресурсов Kubernetes. Аннотации могут быть пояснительным текстом без конкретной функциональности, задавать конфигурации или поведение облачного провайдера, либо указывать параметры конфигурации и инструменты. Очень мощная функциональность.

Важные замечания

При создании балансировщиков нагрузки рекомендуется вручную настраивать service annotations, чтобы обеспечить корректное использование aws-lb платформенным балансировщиком нагрузки. Если соответствующие service annotations настроены неправильно, платформа по умолчанию будет использовать eks-clb, который имеет проблемы, связанные с UDP, что может привести к непредвиденным ситуациям.

Использование aws-lb в EKS для обеспечения внешнего доступа к контейнерным сетевым балансировщикам нагрузки

Инструкция по настройке service annotations

1. В соответствующем кластере выполните следующую команду с помощью kubectl, чтобы найти все Pod в namespace kube-system с именами, содержащими "aws-load":

   ```
   kubectl get pod -n kube-system |grep aws-load
   ```

2. Создайте балансировщик нагрузки; подробные шаги и параметры создания см. в разделе создания Load Balancer в AWS EKS Service Annotation Instructions.

   - Если команда не возвращает связанных Pod, значит в кластере не установлен AWS Load Balancer Controller. Service annotations не требуются; создавайте балансировщик напрямую.
   - Если команда возвращает связанные Pod, значит в кластере установлен AWS Load Balancer Controller. При создании балансировщика нагрузки в соответствующем кластере добавьте следующие service annotations. Подробности об аннотациях см. в AWS EKS Service Annotation Instructions:
     - `service.beta.kubernetes.io/aws-load-balancer-type: external //Required`
     - `service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: ip //Required`
     - `service.beta.kubernetes.io/aws-load-balancer-scheme: internet-facing //Optional. Добавьте эту аннотацию, если требуется поддержка публичной сети.`

Метод получения адреса доступа

- При создании контейнерных сетевых балансировщиков нагрузки заполненные service annotations будут установлены на LoadBalancer Service, соответствующий платформенному балансировщику нагрузки.

- В публичных облаках LoadBalancer Service с соответствующими service annotations будет распознаваться облаком и получит назначенный адрес. Платформенный балансировщик нагрузки прочитает этот адрес и установит его в качестве собственного адреса доступа.

Конфигурация инициализации сети кластера Huawei Cloud CCE

Содержание

Обзор поддержки

Функция	Статус поддержки	Требования
LoadBalancer Service	Поддержка по умолчанию	Дополнительное развертывание не требуется.

Функция	Статус поддержки	Требования
Ingress	Поддержка по умолчанию	Опционально можно включить функциональность **Ingress Class** (после включения можно вручную выбирать классы ingress при создании ingress через интерфейс формы). Дополнительное развертывание не требуется.

Предварительные требования

Если вы создали кластер CCE, импортируйте кластер CCE (Public Cloud).

Шаги конфигурации

Вы можете создавать ingress и LoadBalancer сервисы одновременно или выбрать один из них в зависимости от ваших потребностей.

Создание Ingress

Существует два способа создания ingress. Рекомендуется **Способ 1: Ручной выбор класса Ingress**.

Примечание: избегайте создания двух ресурсов ingress с одинаковым **путём**.

(Рекомендуемый) Способ 1: Ручной выбор класса Ingress

1. В **Container Platform** нажмите **Network** > **Ingress** в левом навигационном меню.
2. Нажмите **Create Ingress** и выберите **CCE Ingress Class** для поля **Ingress Class**.
3. Выберите **Protocol**. По умолчанию — **HTTP**. Для **HTTPS** сначала создайте сертификат и выберите его.
4. Переключитесь в режим **YAML** и добавьте следующие аннотации в зависимости от типа вашего Ingress Controller по умолчанию. Подробности об аннотациях см. в разделе Использование аннотаций для настройки Load Balancer ↗:

Примечание: Замените **значения** в аннотациях ниже на актуальные значения вашей среды.

Тип Ingress Controller по умолчанию	Аннотации

Shared (Автосоздание)	`kubernetes.io/elb.autocreate: '{"type":"public","bandwidth_name":`
	`{random}","bandwidth_chargemode":"traffic","bandwidth_size":5,"ban`
	`kubernetes.io/elb.class: union`

Shared (Повторное использование)	`kubernetes.io/elb.class: union`
	`kubernetes.io/elb.id: <Load Balancer Instance ID>`
	`kubernetes.io/elb.port: '80'`

Dedicated (Автосоздание)	`kubernetes.io/elb.autocreate: '{"type":"public","bandwidth_name":`
	`Name>","bandwidth_chargemode":"traffic","bandwidth_size":5,"bandwi`
	`["<AZ A>","<AZ B>","<AZ C>"],"elb_virsubnet_ids":["<ELB Virtual Su`
	`ID>"],"l7_flavor_name":"L7_flavor.elb.s1.small","l4_flavor_name":"`
	`kubernetes.io/elb.class: performance`
	`kubernetes.io/elb.port: "80"`

Dedicated (Повторное использование)	`kubernetes.io/elb.class: performance`
	`kubernetes.io/elb.id: <Load Balancer Instance ID>`
	`kubernetes.io/elb.port: "80"`

5. Нажмите **Create**. После создания вы сможете получить доступ к сервисам кластера через ELB.

Способ 2: Использование класса Ingress по умолчанию

1. Создайте YAML-файл IngressClass со следующим содержимым. Подробности см. в разделе Класс Ingress по умолчанию ↗:

```
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  annotations:
    ingressclass.kubernetes.io/is-default-class: "true"
  name: cce
spec:
  controller: alauda/cce
```

2. Сохраните файл и примените его к импортированному кластеру. Замените `<filename.yaml>` на имя вашего YAML-файла:

```
kubectl apply -f <filename.yaml>
```

3. В **Container Platform** нажмите **Network** > **Ingress** в левом навигационном меню.
4. Выберите **Protocol**. По умолчанию — **HTTP**. Для **HTTPS** сначала создайте сертификат и выберите его.
5. Нажмите **Create**. После создания вы сможете получить доступ к сервисам кластера через ELB.

Создание LoadBalancer Service

1. В **Container Platform** нажмите **Network** > **Services** в левом навигационном меню.
2. Нажмите **Create Service** и выберите **LoadBalancer** для **External Access**.
3. Разверните раздел **annotations** и при необходимости заполните аннотации для сервиса LoadBalancer.
4. Нажмите **Create**.

Связанные операции

Создание сертификата

При использовании протокола HTTPS заранее сохраните учетные данные HTTPS-сертификата в Secret (тип TLS).

1. В **Container Platform** нажмите **Configuration** > **Secrets** в левом навигационном меню.
2. Нажмите **Create Secret**.
3. Выберите тип **TLS** и импортируйте или заполните поля **Certificate** и **Private Key** по необходимости.
4. Нажмите **Create**.

Конфигурация инициализации сети кластера Azure AKS

Содержание

Обзор поддержки

Функция	Статус поддержки	Требования
LoadBalancer Service	Поддерживается по умолчанию	Дополнительное развертывание не требуется.
Ingress	Поддерживается	По желанию разверните Ingress Controller. По желанию включите функциональность **Ingress**

Функция	Статус поддержки	Требования
		Class (после включения вы сможете вручную выбирать классы ingress при создании ingress через форму).

Предварительные требования

Если вы создали кластер AKS, импортируйте кластер Azure AKS.

Шаги конфигурации

Развертывание Ingress Controller

AKS использует **режим контейнерной сети** и применяет **Nginx Ingress Controller** для управления балансировщиками нагрузки, при этом предоставляя внешние адреса доступа для виртуальных IP-адресов (VIP) во внутренней сети контейнеров через **LoadBalancer** тип **Services**.

1. Войдите в Microsoft Azure и перейдите к созданному кластеру AKS.
2. В левом меню выберите **Kubernetes Resources** > **Services and Ingresses**.
3. Нажмите **Create**, выберите из выпадающего списка **Ingress (Preview)**, после чего будет предложено и автоматически создан Ingress Controller.
4. Нажмите **Enable** и дождитесь завершения операции.

Создание Ingress и LoadBalancer Services

Вы можете создавать ingress и LoadBalancer сервисы одновременно или выбрать один из них в зависимости от ваших потребностей.

Создание Ingress

1. В **Container Platform** в левом меню выберите **Network** > **Ingress**.

2. Нажмите **Create Ingress** и выберите **webapprouting.kubernetes.azure.com** для **Ingress Class**.
3. Выберите **Protocol**. По умолчанию — **HTTP**. Для **HTTPS** сначала создайте сертификат и выберите его.
4. Нажмите **Create**.

Создание LoadBalancer Service

1. В **Container Platform** в левом меню выберите **Network** > **Services**.
2. Нажмите **Create Service** и выберите **LoadBalancer** для **External Access**.
3. Разверните раздел **annotations** и при необходимости заполните аннотации для LoadBalancer сервиса.
4. Нажмите **Create**.

Связанные операции

Создание сертификата

При использовании протокола HTTPS заранее сохраните учетные данные HTTPS-сертификата в Secret (типа TLS).

1. В **Container Platform** в левом меню выберите **Configuration** > **Secrets**.
2. Нажмите **Create Secret**.
3. Выберите тип **TLS** и импортируйте или заполните поля **Certificate** и **Private Key** по необходимости.
4. Нажмите **Create**.

Конфигурация инициализации сети кластера Google GKE

Содержание

Обзор поддержки

Функция	Статус поддержки	Требования
LoadBalancer Service	Поддержка по умолчанию	Дополнительное развертывание не требуется.

Функция	Статус поддержки	Требования
Ingress	Поддержка по умолчанию	Опционально можно включить функциональность **Ingress Class** (после включения можно вручную выбирать классы ingress при создании ingress через интерфейс формы). Дополнительное развертывание не требуется.

Предварительные требования

Если вы создали кластер GKE, импортируйте кластер GKE.

Шаги конфигурации

Развертывание Ingress Controller

Ручное развертывание не требуется. GKE предоставляет управляемый встроенный контроллер Ingress под названием GKE Ingress. Этот контроллер сопоставляет ресурсы Ingress с Google Cloud Load Balancers для обработки HTTP(S) нагрузок в GKE, что упрощает и автоматизирует конфигурацию.

Создание Ingress и LoadBalancer сервисов

Вы можете создавать ingress и LoadBalancer сервисы одновременно или выбрать один из них в зависимости от ваших потребностей.

Создание Ingress

1. В **Container Platform** нажмите **Network** > **Ingress** в левом навигационном меню.
2. Нажмите **Create Ingress** и выберите **GKE Ingress Class** для **Ingress Class**.
3. Выберите **Protocol**. По умолчанию — **HTTP**. Для **HTTPS** сначала создайте сертификат и выберите его.

4. Нажмите **Create**. Подождите примерно 5 минут, пока платформа GKE автоматически назначит публичный IP-адрес для ingress.
   Примечание: Разным ресурсам ingress будут назначены разные публичные IP-адреса.

Создание LoadBalancer сервиса

1. В **Container Platform** нажмите **Network** > **Services** в левом навигационном меню.
2. Нажмите **Create Service** и выберите **LoadBalancer** для **External Access**.
3. Разверните **annotations** и при необходимости заполните аннотации для LoadBalancer сервиса.
4. Нажмите **Create**.

Связанные операции

Просмотр ресурсов Ingress в Google Cloud

1. Перейдите в **Google Cloud > Kubernetes Engine** и нажмите **Services and Ingress** в левом навигационном меню.
2. Нажмите **INGRESS**.
3. Просмотрите информацию о соответствующих ресурсах Ingress в списке.

Создание сертификата

При использовании протокола HTTPS заранее сохраните учетные данные HTTPS сертификата в Secret (тип TLS).

1. В **Container Platform** нажмите **Configuration** > **Secrets** в левом навигационном меню.
2. Нажмите **Create Secret**.
3. Выберите тип **TLS** и импортируйте или заполните поля **Certificate** и **Private Key** по необходимости.
4. Нажмите **Create**.

Инициализация хранилища

Обзор

- Amazon Elastic Kubernetes Service (Amazon EKS) — это управляемый сервис Kubernetes от Amazon для запуска Kubernetes в облаке AWS и в локальных дата-центрах. В облаке Amazon EKS автоматически управляет доступностью и масштабируемостью узлов управляющей плоскости Kubernetes, отвечающих за планирование контейнеров, управление доступностью приложений, хранение данных кластера и другие критически важные задачи, обеспечивая единое и полностью поддерживаемое решение Kubernetes.
- Huawei Cloud Container Engine (CCE) предоставляет высоконадежные, высокопроизводительные корпоративные сервисы управления контейнерными приложениями, поддерживает нативные приложения и инструменты сообщества Kubernetes, упрощая создание автоматизированных сред выполнения контейнеров в облаке.
- Azure Kubernetes Service (AKS) предоставляет самый быстрый способ начать разработку и развертывание облачно-нативных приложений на Azure, в дата-центрах или на периферии с использованием встроенных конвейеров от кода до облака и защитных механизмов, с единым управлением и контролем для локальных, периферийных и мультиоблачных Kubernetes кластеров.
- Google Kubernetes Engine (GKE) предлагает чрезвычайно масштабируемый, полностью автоматизированный сервис Kubernetes, который можно использовать практически без глубоких знаний Kubernetes. Его преимущества включают повышенную скорость, снижение рисков и общую стоимость владения, встроенные инструменты безопасности и наблюдаемости, а также передовые решения автоскейлинга, способные масштабироваться до 15 000 узлов.

Содержание

Поддержка классов хранилищ

Кластеры AWS EKS

Тип хранилища	Класс хранилища по умолчанию	Создание PVC с режимом доступа RWO	Создание PVC с режимом доступа RWX	Расширение PVC
Файловое	efs-sc	Поддерживается	Поддерживается	Не поддерживается
Блочное	ebs-sc	Поддерживается	Не поддерживается	Поддерживается

Кластеры Huawei Cloud CCE

Тип хранилища	Класс хранилища по умолчанию	Создание PVC с режимом доступа RWO	Создание PVC с режимом доступа RWX	Расширение PVC
Файловое	csi-nas	Не поддерживается	Поддерживается	Поддерживается

Тип хранилища	Класс хранилища по умолчанию	Создание PVC с режимом доступа RWO	Создание PVC с режимом доступа RWX	Расширение PVC
Блочное	csi-disk	Поддерживается	Не поддерживается	Поддерживается

Кластеры Azure AKS

Тип хранилища	Класс хранилища по умолчанию	Создание PVC с режимом доступа RWO	Создание PVC с режимом доступа RWX	Расширение PVC
Файловое	azurefile	Поддерживается	Поддерживается	Поддерживается
Блочное	default	Поддерживается	Не поддерживается	Поддерживается

Кластеры Google GKE

Тип хранилища	Класс хранилища по умолчанию	Создание PVC с режимом доступа RWO	Создание PVC с режимом доступа RWX	Расширение PVC
Файловое	standard-rwx	Поддерживается	Поддерживается	Поддерживается
Блочное	standard-rwo	Поддерживается	Не поддерживается	Поддерживается

Конфигурация инициализации хранилища кластера AWS EKS

Интеграция платформы с AWS EKS и конфигурация инициализации хранилища.

Содержание

Ограничения и лимиты

- Класс хранения файлов по умолчанию efs-sc может не поддерживать изменение разрешений после монтирования, что может привести к некорректной работе некоторых приложений, таких как PostgreSQL и Jenkins.
- Инстансы серии A1 не поддерживаются AMI AL2023, что препятствует корректному развертыванию плагина блочного хранилища EBS (Amazon EBS CSI Driver). Драйвер

EBS CSI имеет GA поддержку мультиархитектуры/ARM, поэтому ограничение связано с поддержкой AMI/инстансов, а не с самим драйвером. Если необходимо использовать классы блочного хранилища EBS, избегайте использования следующих типов инстансов и рассмотрите альтернативы на базе Graviton2/3:

- a1.medium
- a1.large
- a1.xlarge
- a1.2xlarge
- a1.4xlarge

Рекомендуемые альтернативы: Используйте семейства инстансов Graviton2/3, такие как m6g, c6g, r6g, t4g и др., которые обеспечивают лучшую производительность и полную поддержку драйвера EBS CSI.

Предварительные требования

- Убедитесь, что доступны инструменты kubectl↗ и AWS CLI.
- Если кластер EKS уже создан, импортируйте кластер Amazon EKS; если нет — создайте кластер AWS EKS.
- Разверните в кластере EKS плагин файлового хранилища **Amazon EFS CSI Driver** и плагин блочного хранилища **Amazon EBS CSI Driver**.
  Примечание: При использовании файлового хранилища EFS создайте файловое хранилище в регионе EKS и запишите **File System ID** из **File System**.

Шаги конфигурации

Создание классов хранения

1. Перейдите в **Platform Management** и в левом меню выберите **Storage Management** > **Storage Classes**.
2. Нажмите на выпадающий список рядом с **Create Storage Class** > **Create from YAML**.

3. Добавьте следующий контент в YAML-файл для создания классов хранения по умолчанию по мере необходимости. Имя класса хранения по умолчанию для файлового хранилища↗ — **efs-sc**, для блочного хранилища↗ — **ebs-sc**.

   - Файловое хранилище EFS

     Примечание: Замените `<File System ID>` на фактический **File System ID**, например, `fileSystemId: fs-05aef9e1edd309f2b`.

     ```
     kind: StorageClass
     apiVersion: storage.k8s.io/v1
     metadata:
       name: efs-sc
     provisioner: efs.csi.aws.com
     parameters:
       provisioningMode: efs-ap
       fileSystemId: <File System ID>
       directoryPerms: "755"
     ```

   - Блочное хранилище EBS

     ```
     allowVolumeExpansion: true
     apiVersion: storage.k8s.io/v1
     kind: StorageClass
     metadata:
       name: ebs-sc
     provisioner: ebs.csi.aws.com
     reclaimPolicy: Delete
     volumeBindingMode: WaitForFirstConsumer
     ```

4. Нажмите **Create**.

   Примечание: Если классы хранения по умолчанию не соответствуют требованиям, создайте новые классы хранения, следуя указанным выше шагам, и при необходимости измените параметры. См. Доступные параметры класса хранения.

Изменение назначения класса хранения проекту

1. В левом меню выберите **Storage Management** > **Storage Classes**.

2. Нажмите на три точки рядом с классом хранения с именем **efs-sc** или **ebs-sc** > **Update Project**.
3. Выберите нужный метод **Project Assignment** и нажмите **Update** для назначения класса хранения проектам.

Связанные операции

Конфигурация доступных параметров класса хранения

- Доступные параметры файлового хранилища EFS

Параметр	Допустимые значения	Значение по умолчанию	Необязательный	Опис
az		""	true	Испол для кр аккау монти Если испол mount связа для кр аккау монти если случа образ выбир mount кросс

Параметр	Допустимые значения	Значение по умолчанию	Необязательный	Опис
				аккау монти
basePath			true	Путь д созда динам выдел acces Если acces созда корне катал файл систе
directoryPerms			false	Разре катал созда корне катал Point
uid			true	POSI для с корне катал Point
gid			true	POSI для с корне

Параметр	Допустимые значения	Значение по умолчанию	Необязательный	Опис
				катал Point
gidRangeStart		50000	true	Начал диапа group приме при с корне катал point требу задан
gidRangeEnd		7000000	true	Конеч диапа group требу задан
subPathPattern			true	Шабл постр подпу распо кажды point, при динам выдел Може из фикси строк

Параметр	Допустимые значения	Значение по умолчанию	Необязательный	Опис
				огран перем анало перем "subP в чарт subdir provis Опцио парам .PVC. .PVC. .PV.na
ensureUniqueDirectory		true	true	Испол при в динам выдел значе добав шабло subPa чтобь гаран что ас не ука случа один катал **Прим** устан в false если

Параметр	Допустимые значения	Значение по умолчанию	Необязательный	Опис
				что эт желае повед
provisioningMode	efs-ap		false	Тип то насто время подде acces
fileSystemId			false	ID фа систе котор созда point.

- Доступные параметры блочного хранилища EBS

  Примечание: Для параметров производительности различных типов томов см. Amazon EBS Volume Types↗.

Параметр	Допустимые значения	Значение по умолчанию	Описание
"allowAutoIOPSPerGBIncrease"	true, false	false	При значении "true" CSI драйвер увеличивает IOPS тома, если iopsPerGB * <размер тома> слишком низок для соответствия поддерживаемому

Параметр	Допустимые значения	Значение по умолчанию	Описание
			AWS диапазону IOPS. Это гарантирует успешное динамическое выделение даже при слишком малых значениях емкости PVC или iopsPerGB, но может привести к дополнительным расходам из-за более высокого IOPS, чем требуется.
"blockExpress"	true, false	false	Создает тома io2 Block Express, повышая лимиты IOPS для томов io2 до 256000, но тома с IOPS выше 64000 не могут быть смонтированы на инстансах, не поддерживающих io2 Block Express.
"blockSize"			Размер блока, используемый при форматировании

Параметр	Допустимые значения	Значение по умолчанию	Описание
			базовой файловой системы. Применимо только к Linux-узлам с файловыми системами ext2, ext3, ext4 или xfs.
"bytesPerINode"			Количество байт на inode при форматировании базовой файловой системы. Применимо только к Linux-узлам с файловыми системами ext2, ext3 или ext4.
"csi.storage.k8s.io/fstype"	xfs, ext2, ext3, ext4	ext4	Тип файловой системы для форматирования при создании томов. Чувствительно к регистру.
"encrypted"	true, false	false	Требуется ли шифрование

Параметр	Допустимые значения	Значение по умолчанию	Описание
			тома.
"inodeSize"			Размер inode при форматировании базовой файловой системы. Применимо только к Linux-узлам с файловыми системами ext2, ext3, ext4 или xfs. Inode — структуры данных в файловых системах, хранящие метаданные файлов и каталогов.
"iops"			Количество операций ввода-вывода в секунду, применимо к томам IO1, IO2 и GP3.
"iopsPerGB"			Количество операций ввода-вывода в секунду на гигабайт,

Параметр	Допустимые значения	Значение по умолчанию	Описание
			применимо к томам IO1, IO2 и GP3.
"kmsKeyId"			Полный ARN ключа для шифрования томов. Если не указан, AWS использует ключ KMS по умолчанию для региона тома и автоматически генерирует ключ с именем /aws/ebs.
"numberOfINodes"			Количество inode, указанное при форматировании базовой файловой системы. Применимо только к Linux-узлам с файловыми системами ext2, ext3 или ext4.
"throughput"		125	Пропускная способность в МиБ/с.

Параметр	Допустимые значения	Значение по умолчанию	Описание
			Действительно только при указании типа тома gp3. Если не указано, по умолчанию 125 МиБ/с. См. Amazon EBS Volume Types↗.
"type"	io1, io2, gp2, gp3, sc1, st1, standard, sbp1, sbg1	gp3	Тип тома EBS.

Конфигурация инициализации хранилища кластера Huawei Cloud CCE

Интеграция платформы с Huawei Cloud CCE и конфигурация инициализации хранилища.

Содержание

Ограничения и лимиты

Количество PVC в кластере ограничено, а у аккаунта есть квоты на емкость хранилища. Вы можете запросить увеличение через тикеты в службу поддержки.

Предварительные требования

- Если вы создали кластер CCE, импортируйте кластер CCE (Public Cloud).

Шаги конфигурации

1. Перейдите в **Управление платформой** и в левом меню выберите **Управление хранилищем** > **Классы хранилищ**.
2. Нажмите на три точки рядом с классом хранилища с именем **csi-nas** или **csi-disk** > **Обновить проект**.
   Примечание: После импорта кластера CCE создаются классы хранилищ по умолчанию. Для блочного хранилища рекомендуется использовать **csi-disk**, для файлового — **csi-nas**. См. Описание классов хранилищ по умолчанию.
3. Выберите необходимый метод **Назначения проекта** и нажмите **Обновить**, чтобы назначить классы хранилищ **csi-nas** или **csi-disk** проектам.

Описание классов хранилищ по умолчанию

Название класса хранилища	Тип класса хранилища	Описание
(Рекомендуется) csi-disk	Блочное хранилище	Рекомендуется к использованию.
(Рекомендуется) csi-nas	Файловое хранилище	Рекомендуется к использованию. Доступно только в регионах с поддержкой сервиса csi-nas.
csi-disk-topology	Блочное хранилище	Автоматическая топология облачных дисков при размещении узлов в разных AZ.
csi-local	Локальное хранилище	

Название класса хранилища	Тип класса хранилища	Описание
csi-local-topology	Локальное хранилище	
csi-obs	Объектное хранилище	
csi-sfsturbo	Сверхбыстрое файловое хранилище	Сверхбыстрое файловое хранилище не поддерживает динамическое создание персистентных томов.

Распространённые проблемы

Ошибка создания PVC

Возникает следующая ошибка из-за достижения лимита по количеству PVC. Вы можете запросить увеличение через тикеты поддержки:

```
message: "ShareLimitExceeded: Maximum number of shares allowed (10) excee
ded."
```

Просроченный аккаунт

Создание PVC не удаётся из-за просроченных платежей. Пожалуйста, оплатите своевременно:

```
message: "Your account is suspended and resources cannot be used"
```

Конфигурация инициализации хранилища кластера Azure AKS

Интеграция платформы с Azure AKS и конфигурация инициализации хранилища.

Содержание

Ограничения и лимиты

Класс хранения файлов по умолчанию azurefile может не поддерживать изменение разрешений после монтирования, что может привести к некорректной работе некоторых приложений, таких как PostgreSQL и Jenkins.

Предварительные требования

- Если вы создали кластер AKS, импортируйте кластер Azure AKS.

Шаги конфигурации

1. Перейдите в **Управление платформой** и в левом меню выберите **Управление хранилищем** > **Классы хранилищ**.
2. Нажмите на три точки рядом с классом хранилища с именем **default** или **azurefile** > **Обновить проект**.
   Примечание: После импорта кластера AKS создаются следующие классы хранилищ по умолчанию. Рекомендуется использовать **default** для блочного хранилища и **azurefile** для файлового. См. Описание классов хранилищ по умолчанию.
3. Выберите метод **Назначение проекта** по необходимости и нажмите **Обновить**, чтобы назначить классы хранилищ **default** или **azurefile** проектам.
   Примечание: Если классы хранилищ по умолчанию не соответствуют требованиям, создайте новые классы хранилищ, следуя вышеуказанным шагам, и при необходимости измените параметры. См. Доступные параметры классов хранилищ.

Связанная информация

Описание классов хранилищ по умолчанию

Название класса хранилища	Тип хранилища	Описание
(Рекомендуется) azurefile	Файловое хранилище	Создаёт Azure file shares с использованием стандартного хранилища Azure.
(Рекомендуется) default	Блочное хранилище	Создаёт управляемые диски с использованием Azure StandardSSD.
azurefile-csi	Файловое хранилище	Создаёт Azure file shares с использованием стандартного хранилища Azure.

Название класса хранилища	Тип хранилища	Описание
azurefile-csi-nfs	Файловое хранилище	Создаёт Azure file shares с использованием стандартного хранилища Azure, протокол NFS.
azurefile-csi-premium	Файловое хранилище	Создаёт Azure file shares с использованием премиум-хранилища Azure.
azurefile-premium	Файловое хранилище	Создаёт Azure file shares с использованием премиум-хранилища Azure.
managed	Блочное хранилище	Создаёт управляемые диски с использованием Azure StandardSSD.
managed-csi	Блочное хранилище	Создаёт управляемые диски с использованием локально избыточного хранилища Azure StandardSSD (LRS).
managed-csi-premium	Блочное хранилище	Создаёт управляемые диски с использованием локально избыточного премиум-хранилища Azure (LRS).
managed-premium	Блочное хранилище	Создаёт управляемые диски с использованием премиум-хранилища Azure.

Доступные параметры классов хранилищ

- Для необязательных параметров блочного хранилища и их значений смотрите Create and use volumes with Azure disks in Azure Kubernetes Service (AKS)↗.
- Для необязательных параметров файлового хранилища и их значений смотрите Create and use volumes with Azure Files in Azure Kubernetes Service (AKS)↗.

Конфигурация инициализации хранилища кластера Google GKE

Интеграция платформы с Google GKE и конфигурация инициализации хранилища.

Содержание

Ограничения и лимиты

- Минимальный объем PVC для файлового хранилища по умолчанию составляет 1T. Если при создании указать объем меньше 1T, он автоматически расширится до 1T.

- Файловое хранилище по умолчанию имеет ограничения по емкости. Вы можете запросить расширение через тикеты в службу поддержки.
- Операции создания и удаления файлового хранилища по умолчанию занимают значительное время. Если статус создания длительное время остается активным, проявите терпение.

Требования

- При создании кластеров на платформе Google Cloud в разделе **Cluster > Features** на странице **Other** отметьте опции **Enable Compute Engine Persistent Disk CSI Driver** и **Enable Filestore CSI Driver**.
- Включите **Cloud Filestore API** и **Google Kubernetes Engine API** на платформе Google Cloud. См. Access Filestore instances using the Filestore CSI driver↗.
- Отрегулируйте региональные квоты на файловое хранилище в Google Cloud Platform. См. Resource quotas and limits↗.
- Если кластер GKE уже создан, импортируйте кластер GKE.

Шаги конфигурации

1. Перейдите в **Platform Management** и в левом меню выберите **Storage Management** > **Storage Classes**.
2. Нажмите на три точки рядом с классом хранилища с именем **standard-rwx** или **standard-rwo** > **Update Project**.
   Примечание: После импорта кластера GKE создаются классы хранилища по умолчанию. Для файлового хранилища рекомендуется использовать **standard-rwx**, для блочного — **standard-rwo**. См. Default Storage Class Description.
3. Выберите метод **Project Assignment** по необходимости и нажмите **Update** для назначения классов хранилища **standard-rwx** или **standard-rwo** проектам.
   Примечание: Если классы хранилища по умолчанию не соответствуют требованиям, создайте новые классы хранилища, следуя указанным шагам, и при необходимости измените параметры. См. Available Storage Class Parameters.

Связанная информация

Описание классов хранилища по умолчанию

Название класса хранилища	Тип хранилища	Описание
(Рекомендуется) standard-rwx	Файловое хранилище	Использует Basic HDD Filestore service tier ↗ .
(Рекомендуется) standard-rwo	Блочное хранилище	Использует сбалансированные постоянные диски.
premium-rwx	Файловое хранилище	Использует Basic SSD Filestore service tier ↗ .
premium-rwo	Блочное хранилище	SSD постоянные диски.
enterprise-rwx	Файловое хранилище	Использует Enterprise Filestore tier ↗ .
enterprise-multishare-rwx	Файловое хранилище	Использует Enterprise Filestore tier ↗ . См. Filestore multishares for Google Kubernetes Engine ↗ .

Доступные параметры класса хранилища

- Для опциональных параметров блочного хранилища и их значений см. Storage options ↗ .
- Для опциональных параметров файлового хранилища и их значений см. Service tiers ↗ .

Распространённые проблемы

Ошибка создания PVC класса хранилища файлового типа

- Следующая ошибка возникает, если Cloud Filestore API не включен в проекте или отсутствуют соответствующие разрешения. Для решения см. Prerequisites:

```
failed to provision volume with StorageClass "standard-rwx": rpc error:
code = PermissionDenied desc = googlespi: Error 403: Cloud Filestore AP
I has not been used in project alauda-proj-1234 before or it is disable
d.
...
resion: SERVICE_DISABLED
```

- Следующая ошибка возникает из-за превышения квот хранилища. Для решения см. Prerequisites:

```
failed to provision volume with StorageClass "standard-rwx": rpc error:
code = ResourceExhausted desc = googlespi: Error 429: Quora limit 'Stan
dardStorageGbPerRegion' has been exceeded. Limit 2048 in region asia-ea
st1.
 rateLimitExceeded
```

PVC класса хранилища блочного типа не может корректно привязаться

Следующая ошибка возникает, потому что у CSINode узла отсутствует конфигурация для драйвера pd.csi.storage.gke.io. Решается перезапуском **Compute Engine Persistent Disk CSI Driver**.

Примечание: Обновление этого плагина сделает кластер недоступным. Процесс обновления занимает примерно 5-10 минут.

```
Warning ProvisioningFailed 18m (x14 over 39m) pd.csi.storage.gke.io_gke-5
cb9bddae4d1430eb8ad-01f4-2084-vm_4b4e70bd-e2db-4779-9102-fee83a657ced fai
led to provision volume with StorageClass "standard": error generating ac
cessibility requirements: no available topology found
Normal ExternalProvisioning 4m35s (x143 over 39m) persistentvolume-contro
ller waiting for a volume to be created, either by external provisioner
"pd.csi.storage.gke.io" or manually created by system administrator
Normal Provisioning 3m19s (x18 over 39m) pd.csi.storage.gke.io_gke-5cb9bd
dae4d1430eb8ad-01f4-2084-vm_4b4e70bd-e2db-4779-9102-fee83a657ced External
provisioner is provisioning volume for claim "acp-gke-test/standard"
```

Как сделать

Настройка сети для импортир **Получение информации о ими** **Доверие не**

тев

ых

Настройка сбора аудита для импортированных стандартных кластеров Kubernetes

Включите аудит Kubernetes API сервера в импортированных стандартных кластерах Kubernetes, чтобы платформа могла собирать данные аудита.

Настройка сети для импортируемых кластеров

Содержание

Описание сценария

До импорта кластера обеспечивается только однонаправленная связь, позволяющая глобальному кластеру получать доступ к импортируемому кластеру. После импорта кластера, чтобы обеспечить корректный доступ импортируемого кластера к глобальному кластеру и достичь двунаправленной связи, требуется дополнительная настройка сети для обеспечения нормальной работы функциональных компонентов платформы.

Предварительные требования

Пожалуйста, заранее подготовьте **доменное имя**, **IP-адрес**, на который указывает доменное имя, и соответствующий **действительный сертификат**, к которому импортируемый кластер сможет получить доступ.

Примечание:

- Это доменное имя не должно совпадать с **адресом доступа к платформе**.
- Убедитесь, что порт домена (порт HTTPS, совпадающий с портом адреса доступа к платформе) может перенаправлять трафик на все управляющие узлы глобального кластера.

Добавление информации аннотации для импортируемых кластеров

Для обеспечения корректного доступа компонентов оповещения и сбора логов к платформе необходимо добавить информацию аннотации в импортируемый кластер.

1. В левой навигационной панели нажмите **Cluster Management** > **Clusters**.
2. Нажмите **global**.
3. Нажмите **Operations** > **CLI Tools** и замените соответствующие параметры, используя следующую команду:

   ```
   kubectl annotate --overwrite -n cpaas-system clusters.cluster.x-k8s.io
   <imported cluster name> \
           cpaas.io/platform-url=<prepared domain address, e.g.: https://w
   ww.domain.cn>
   ```

   Пример кода:

   ```
   kubectl annotate --overwrite -n cpaas-system clusters.cluster.x-k8s.io
   cluster-imported \
           cpaas.io/platform-url=https://www.domain.cn
   ```

Как получить информацию об импортируемом кластере?

Содержание

Описание проблемы

Получить конфигурацию, необходимую для подключения к импортируемому кластеру, чтобы платформа могла быть авторизована для доступа и управления им. В этом разделе приведены шаги для получения информации об импортируемом кластере.

Предварительные требования

- Рабочая среда `kubectl`. **Для кластеров в публичных облаках настоятельно рекомендуется использовать CloudShell провайдера.** Если CloudShell недоступен, рекомендуется использовать Linux или macOS.

- Вы получили файл KubeConfig импортируемого кластера и скопировали его в среду, где установлен `kubectl`. **Чтобы избежать работы с неправильной средой, настоятельно рекомендуется использовать один из следующих безвредных подходов:**

  - **Подход с резервным копированием**: Сначала скопируйте существующий kubeconfig в безопасное место: `cp "${HOME}/.kube/config" "${HOME}/.kube/config.backup"`
  - **Изолированный подход**: Установите переменную окружения `KUBECONFIG`, указывающую на импортированный kubeconfig: `export KUBECONFIG="/path/to/imported/kubeconfig"`
  - **Подход с объединением**: Используйте слияние/уплощение kubectl без потери существующих контекстов:

    1. `export KUBECONFIG="/path/to/imported/kubeconfig:${HOME}/.kube/config"`
    2. `kubectl config view --flatten > "${HOME}/.kube/merged.kubeconfig"`
    3. `export KUBECONFIG="${HOME}/.kube/merged.kubeconfig"`

Получение информации о кластере

Получение токена кластера

1. Выполните следующие команды:

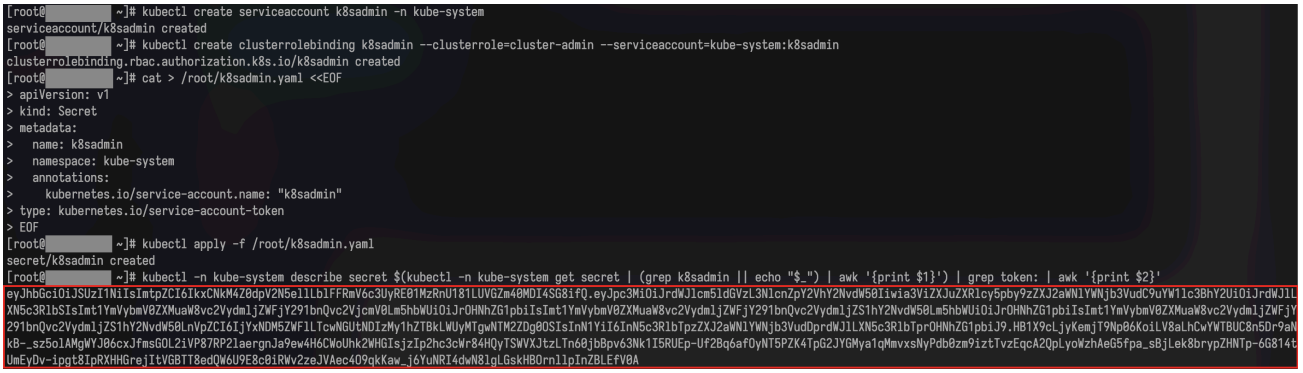
```
# [Важно] Последующие операции поддерживаются только в bash

# Ручное создание секрета, привязка service account и генерация токена без срока действия
kubectl get ns cpaas-system > /dev/null 2>&1 || kubectl create namespace cpaas-system
kubectl create serviceaccount k8sadmin -n cpaas-system
kubectl create clusterrolebinding k8sadmin --clusterrole=cluster-admin --serviceaccount=cpaas-system:k8sadmin

cat | kubectl apply -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: k8sadmin
  namespace: cpaas-system
  annotations:
    kubernetes.io/service-account.name: "k8sadmin"
type: kubernetes.io/service-account-token
EOF

kubectl -n cpaas-system describe secret \
    $(kubectl -n cpaas-system get secret | (grep k8sadmin || echo "$_") | awk '{print $1}') \
    | grep -F 'token:' | awk '{print $2}'
```

WARNING

Эта процедура создаёт учетные данные cluster-admin с предполагаемым отсутствием срока действия.

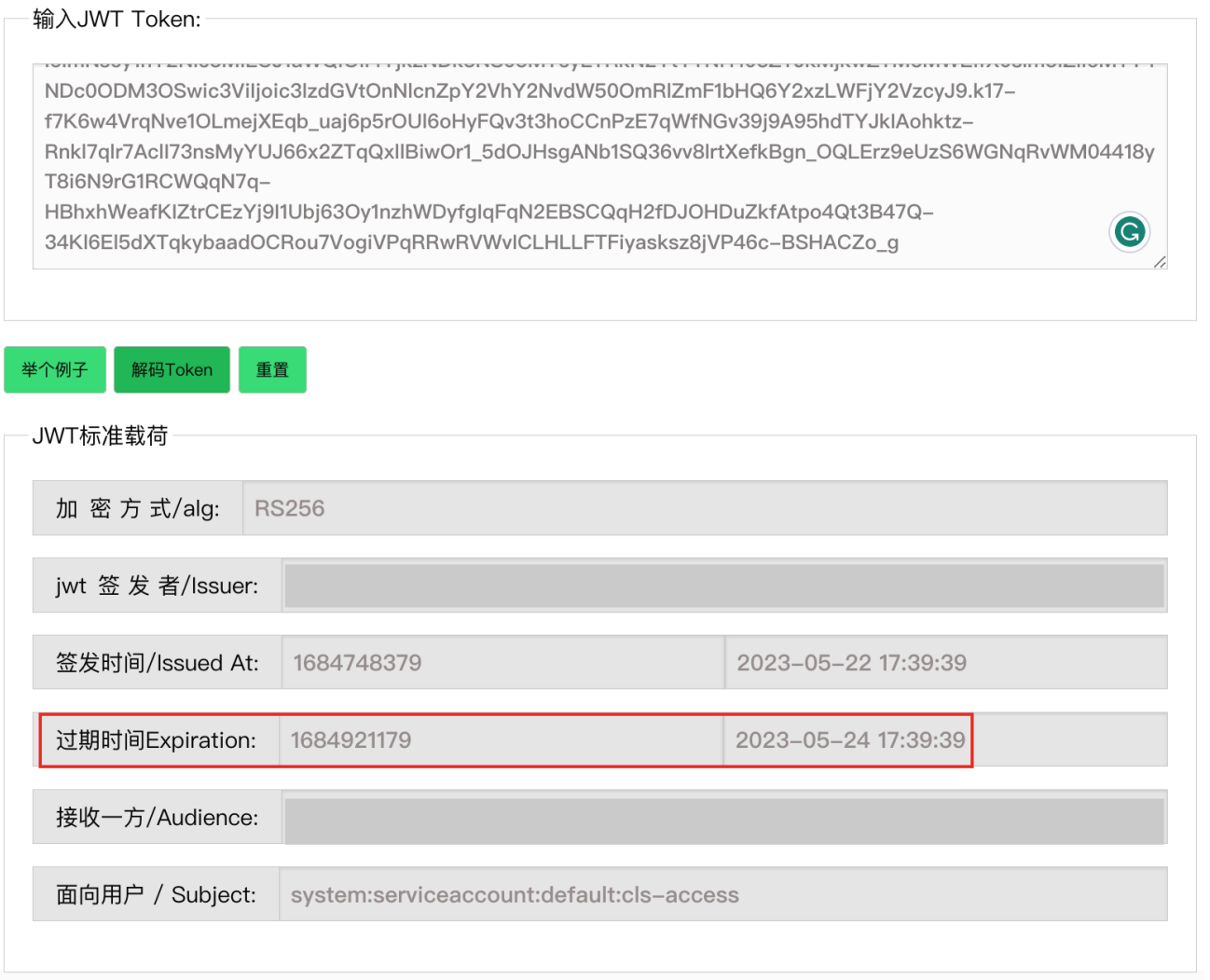
- По возможности используйте RBAC с минимально необходимыми правами, ограниченными нужными ресурсами.
- Храните токен в безопасности; выполняйте ротацию/отзыв при прекращении необходимости.
- Ограничьте круг лиц, имеющих доступ к объектам `Secret` в `cpaas-system`.

2. Пример токена, полученного на предыдущем шаге, показан ниже.



3. Проверьте срок действия токена.

   Используйте любой инструмент, поддерживающий разбор JWT токенов, чтобы проанализировать токен и подтвердить время его истечения. Если в разобранном результате есть поле истечения срока действия (ключ, содержащий "exp", как показано ниже), платформа не сможет управлять импортируемым кластером после этого времени. В этом случае прекратите работу и обратитесь в техническую поддержку.



TIP

Время истечения записано как `"exp": 1684486916,` в исходной полезной нагрузке JWT.

Значение — это UNIX timestamp, который можно преобразовать в UTC время.

Очистка (отзыв доступа) после завершения:

```
kubectl delete clusterrolebinding k8sadmin
kubectl -n cpaas-system delete secret k8sadmin
kubectl -n cpaas-system delete serviceaccount k8sadmin
```

Получение адреса API сервера импортируемого кластера и сертификата CA

TIP

Если вы уже получили адрес API сервера и сертификат CA с помощью функции платформы `Parse KubeConfig File` на странице импортируемого кластера, пропустите этот шаг.

1. Выполните следующие команды:

```
# Просмотр адресов API сервера импортируемого кластера. Адресов может б
ыть несколько; выберите подходящий для вашей среды.
kubectl --kubeconfig "${KUBECONFIG:-${HOME}/.kube/config}" config view
--show-managed-fields=false --flatten --raw -ojsonpath='{$.clusters..cl
uster.server}'
addr_apiserver='<Выбранный адрес API сервера>'

# Получение сертификата CA для указанного выше API сервера
kubectl --kubeconfig "${KUBECONFIG:-${HOME}/.kube/config}" config view
--show-managed-fields=false --flatten --raw \
    -ojsonpath="{$.clusters[?(@.cluster.server == '${addr_apiserve
r}')].cluster.certificate-authority-data}" \
    | { base64 -d 2>/dev/null || base64 -D; }
```

Как доверять небезопасному реестру образов?

Содержание

Описание проблемы

Платформа реестра образов, в которой размещены компоненты, может не предоставлять сервис HTTPS или не иметь действительного TLS-сертификата, выданного публичным центром сертификации. Если вы доверяете этому реестру, настройте ваше контейнерное окружение, выполнив следующие шаги.

Настройка доверия к небезопасному реестру образов

Примечания:

- Все узлы, которые должны использовать образы, включая вновь добавленные, должны быть настроены и на них должен быть перезапущен Containerd.

- Конфигурация немного отличается для Containerd v1.4/v1.5 и v1.6. Следуйте соответствующим шагам для вашей версии.

1. Выполните следующее на каждом узле в импортном кластере:

   - Сделайте резервную копию файла конфигурации

   ```
   mkdir -p '/var/backup-containerd-confs/'
   if ! [ -f /etc/containerd/config.toml ]; then
       echo 'Конфигурация Containerd не найдена. Пожалуйста, проверьте, правильно ли установлен containerd. Если проблема не решается, обратитесь в техническую поддержку.'
       exit 1
   else
       cp /etc/containerd/config.toml /var/backup-containerd-confs/config.toml_$(date +%F_%T)
   fi
   ```

   - Узнайте версию Containerd runtime

   ```
   # Получить версию containerd
   # Сравните эту версию с v1.6. Выберите шаги соответственно
   ctr --version | grep -Eo 'v[0-9]+\.[0-9]+\.[0-9]+'
   ```

Конфигурация Containerd v1.4 v1.5 для небезопасных реестров

2. Выполните следующее на каждом узле в импортном кластере:

   - Отредактируйте файл `/etc/containerd/config.toml`

```
# Пример содержимого для добавления в конфигурационный файл
# Строки в скобках — это секции. Если файл уже содержит секции с таки
ми именами, объедините их содержимое.
[plugins."io.containerd.grpc.v1.cri".registry]
    [plugins."io.containerd.grpc.v1.cri".registry.mirrors]
        [plugins."io.containerd.grpc.v1.cri".registry.mirrors."<regis
try-address>"]
            endpoint = ["https://<registry-address>", "http://<regist
ry-address>"]
        [plugins."io.containerd.grpc.v1.cri".registry.mirrors."192.16
8.134.43"]
            endpoint = ["https://192.168.134.43", "http://192.168.13
4.43"]
    [plugins."io.containerd.grpc.v1.cri".registry.configs]
        [plugins."io.containerd.grpc.v1.cri".registry.configs."<regis
try-address>".tls]
            insecure_skip_verify = true
        [plugins."io.containerd.grpc.v1.cri".registry.configs."192.16
8.134.43".tls]
            insecure_skip_verify = true
```

- Перезапустите Containerd.

```
systemctl daemon-reload && systemctl restart containerd
```

Конфигурация Containerd v1.6 для небезопасных реестров

2. Выполните следующее на каждом узле в импортном кластере:

- Проверьте, существует ли `config_path` в конфигурации.

```
if ! grep -qF 'config_path' /etc/containerd/config.toml; then
    if grep -qE '\[plugins."io.containerd.grpc.v1.cri".registry.(mirrors|configs)(\.|\])' /etc/containerd/config.toml; then
        echo 'Следуйте шагам из раздела "Конфигурация Containerd v1.4 v1.5 для небезопасных реестров".'
    else
        cat >> /etc/containerd/config.toml << 'EOF'
[plugins."io.containerd.grpc.v1.cri".registry]
    config_path = "/etc/containerd/certs.d/"
EOF
    fi
fi

config_path_var=$(grep -F '/etc/containerd/certs.d' /etc/containerd/config.toml)
if [ -z "$config_path_var" ]; then
   echo 'Значение config_path в файле неожиданное. Пожалуйста, проверьте!'
   exit 1
fi
```

- Создайте файл `hosts.toml`.

  Если предыдущая команда вывела `Следуйте шагам из раздела "Конфигурация Containerd v1.4 v1.5 для небезопасных реестров".`, смотрите *Конфигурация Containerd v1.4 v1.5 для небезопасных реестров*.

```
REGISTRY='<адрес реестра, полученный в разделе "Get the registry address">'

mkdir -p "/etc/containerd/certs.d/$REGISTRY/"
cat > "/etc/containerd/certs.d/$REGISTRY/hosts.toml" << EOF
server = "$REGISTRY"
[host."http://$REGISTRY"]
  capabilities = ["pull", "resolve", "push"]
  skip_verify = true
[host."https://$REGISTRY"]
  capabilities = ["pull", "resolve", "push"]
  skip_verify = true
EOF
```

- Перезапустите Containerd.

```
systemctl daemon-reload && systemctl restart containerd
```

Сбор сетевых данных с сетевых карт с пользовательскими именами

Содержание

Описание сценария

После создания workload-кластера платформа мониторинга по умолчанию распознает имена сетевых карт, соответствующие шаблонам `eth.*|en.*|wl.*|ww.*`. Для сетевых карт с пользовательскими именами данные о сетевом трафике на странице мониторинга отображаться не будут. Для решения этой задачи платформа поддерживает изменение соответствующих параметров ресурса для ручного сбора данных о трафике сетевых карт.

Процедура

1. Выполните вход на управляющий узел глобального кластера и выполните следующие команды с помощью kubectl.

2. Сначала найдите имя ресурса moduleinfo, соответствующего workload-кластеру в глобальном кластере:

```
kubectl get moduleinfo | grep -E 'prometheus|victoriametrics'
```

Пример вывода:

```
global-6448ef7f7e5e3924c1629fad826372e7     global     prometheus
prometheus                          Running   v3.15.0-zz231204040711-9d
1fc12474c2   v3.15.0-zz231204040711-9d1fc12474c2   v3.15.0-zz2312040407
11-9d1fc12474c2
ovn-0954f21f0359720e8c115804376b3e7e        ovn        prometheus
prometheus                          Running   v3.15.0-zz231204040711-9d
1fc12474c2   v3.15.0-zz231204040711-9d1fc12474c2   v3.15.0-zz2312040407
11-9d1fc12474c2
```

3. Отредактируйте ресурс moduleinfo workload-кластера, заменив `ovn-0954f21f0359720e8c115804376b3e7e` на имя ресурса moduleinfo workload-кластера, найденное на предыдущем шаге:

```
kubectl edit moduleinfo ovn-0954f21f0359720e8c115804376b3e7e
```

4. Добавьте поле valuesOverride и измените поле и регулярное выражение согласно комментариям:

```
spec:
  valuesOverride: # Если этого поля нет, необходимо добавить valuesOver
ride и следующие параметры внутри spec
    ait/chart-cpaas-monitor:
      ovn: # Замените на имя workload-кластера
        indicator:
          networkDevice: eth.*|em.*|en.*|wl.*|ww.*|[A-Z].*i|custom_inte
rface # Замените custom_interface на пользовательское регулярное выраже
ние для корректного сопоставления имени сетевой карты
```

5. Подождите 10 минут, затем проверьте сетевые графики на странице мониторинга узла, чтобы убедиться, что изменения вступили в силу.

Как настроить сбор аудита для импортированных стандартных кластеров Kubernetes?

Содержание

Описание сценария

После импорта стандартного кластера Kubernetes в платформу необходимо включить аудит Kubernetes API сервера на кластере, прежде чем платформа сможет собирать данные аудита с этого кластера.

Этот документ применим к стандартным кластерам Kubernetes, узлы управляющей плоскости которых находятся под вашим управлением, например, кластерам на базе kubeadm. Он не применим к управляемым облачным кластерам Kubernetes, где вы не можете войти или изменить узлы управляющей плоскости.

Предварительные условия

- Стандартный кластер Kubernetes уже импортирован в платформу.
- Вы можете войти на каждый узел управляющей плоскости в кластере.
- Кластер использует стандартный путь манифеста статического Pod API сервера в стиле kubeadm: `/etc/kubernetes/manifests/kube-apiserver.yaml`.

Процедура

1. Создайте локальный файл `policy.yaml` для политики аудита.
   Установите `apiVersion` в соответствии с версией Kubernetes:

   - Kubernetes до версии 1.24: `audit.k8s.io/v1beta1`
   - Kubernetes 1.24 и выше: `audit.k8s.io/v1`

   Используйте следующее содержимое:

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
  - "RequestReceived"
rules:
  - level: None
    users:
      - system:kube-controller-manager
      - system:kube-scheduler
      - system:serviceaccount:kube-system:endpoint-controller
    verbs: ["get", "update"]
    namespaces: ["kube-system"]
    resources:
      - group: ""
        resources: ["endpoints"]
  - level: None
    nonResourceURLs:
      - /healthz*
      - /version
      - /swagger*
  - level: None
    resources:
      - group: ""
        resources: ["events"]
  - level: None
    resources:
      - group: "devops.alauda.io"
  - level: None
    verbs: ["get", "list", "watch"]
  - level: None
    namespaces:
      - kube-system
      - cpaas-system
      - alauda-system
      - istio-system
      - kube-node-lease
    resources:
      - group: "coordination.k8s.io"
        resources: ["leases"]
  - level: None
    resources:
      - group: "authorization.k8s.io"
        resources: ["subjectaccessreviews", "selfsubjectaccessreviews"]
```

```
    - group: "authentication.k8s.io"
      resources: ["tokenreviews"]
  - level: Metadata
    resources:
    - group: ""
      resources: ["secrets", "configmaps"]
  - level: RequestResponse
    resources:
    - group: ""
    - group: "aiops.alauda.io"
    - group: "apps"
    - group: "app.k8s.io"
    - group: "authentication.istio.io"
    - group: "auth.alauda.io"
    - group: "autoscaling"
    - group: "asm.alauda.io"
    - group: "clusterregistry.k8s.io"
    - group: "crd.alauda.io"
    - group: "infrastructure.alauda.io"
    - group: "monitoring.coreos.com"
    - group: "networking.istio.io"
    - group: "networking.k8s.io"
    - group: "portal.alauda.io"
    - group: "rbac.authorization.k8s.io"
    - group: "storage.k8s.io"
    - group: "tke.cloud.tencent.com"
    - group: "devopsx.alauda.io"
    - group: "core.katanomi.dev"
    - group: "deliveries.katanomi.dev"
    - group: "integrations.katanomi.dev"
    - group: "builds.katanomi.dev"
    - group: "operators.katanomi.dev"
    - group: "tekton.dev"
    - group: "operator.tekton.dev"
    - group: "eventing.knative.dev"
    - group: "flows.knative.dev"
    - group: "messaging.knative.dev"
    - group: "operator.knative.dev"
    - group: "sources.knative.dev"
    - group: "operator.devops.alauda.io"
  - level: Metadata
```

TIP

Если версия кластера ниже 1.24, измените только поле `apiVersion` на `audit.k8s.io/v1beta1`. Остальное содержимое политики остается без изменений.

2. Загрузите `policy.yaml` в каталог `/etc/kubernetes/audit/` на каждом узле управляющей плоскости.

> **WARNING**
>
> - Если в кластере несколько узлов управляющей плоскости, загрузите файл на каждый из них.
> - Создайте каталог вручную, если он не существует: `/etc/kubernetes/audit/`

3. Обновите файл `/etc/kubernetes/manifests/kube-apiserver.yaml` на каждом узле управляющей плоскости.
   Добавьте или обновите следующие флаги, связанные с аудитом, в `spec.containers[].command`:

Флаг	Обязательно	Описание
`--audit-policy-file`	Да	Должен быть установлен в `/etc/kubernetes/audit/policy.yaml`.
`--audit-log-format`	Да	Должен быть установлен в `json`.
`--audit-log-path`	Да	Должен быть установлен в `/etc/kubernetes/audit/audit.log`.
`--audit-log-mode`	Нет	Рекомендуемое значение: `batch`.
`--audit-log-maxsize`	Нет	Максимальный размер файла журнала аудита в МиБ. Рекомендуемое значение: `200`.

Флаг	Обязательно	Описание
`--audit-log-maxbackup`	Нет	Количество сохраняемых файлов журнала аудита. Рекомендуемое значение: `2`.

Пример:

```
- --audit-log-format=json
- --audit-log-maxbackup=2
- --audit-log-maxsize=200
- --audit-log-mode=batch
- --audit-log-path=/etc/kubernetes/audit/audit.log
- --audit-policy-file=/etc/kubernetes/audit/policy.yaml
```

4. Добавьте конфигурацию монтирования каталога аудита в тот же файл `kube-apiserver.yaml`.

   Добавьте следующий элемент в `spec.containers[].volumeMounts`:

```
- mountPath: /etc/kubernetes/audit
  name: k8s-audit
```

   Добавьте следующий элемент в `spec.volumes`:

```
- hostPath:
    path: /etc/kubernetes/audit
    type: DirectoryOrCreate
  name: k8s-audit
```

> **WARNING**
>
> - Обновляйте манифест на каждом узле управляющей плоскости, если их несколько.
> - Значение `volumeMounts[].name` должно совпадать с соответствующим значением `volumes[].name`.
> - Не изменяйте путь монтирования `/etc/kubernetes/audit`.

5. Сохраните файл и проверьте, что конфигурация вступила в силу.

   Проверьте, создан ли файл `/etc/kubernetes/audit/audit.log` на каждом узле управляющей плоскости. Если файл существует и содержит записи аудита, конфигурация работает.

```
ls -l /etc/kubernetes/audit/audit.log
tail -n 20 /etc/kubernetes/audit/audit.log
```

Создание локального кластера

Содержание

Предварительные требования

Требования к узлам

1. Если вы скачали установочный пакет для одной архитектуры с Download Installation Package, убедитесь, что архитектура ваших узлов совпадает с архитектурой пакета. В противном случае узлы не запустятся из-за отсутствия образов, специфичных для архитектуры.
2. Проверьте, что операционная система и ядро узлов поддерживаются. Подробнее см. Supported OS and Kernels.
3. Выполните проверки доступности узлов. Конкретные пункты проверки смотрите в разделе Node Preprocessing > Node Checks.
4. Если IP-адреса узлов недоступны напрямую по SSH, предоставьте для узлов SOCKS5-прокси. Кластер `global` будет обращаться к узлам через этот прокси-сервис.

Балансировка нагрузки

Для производственных сред требуется балансировщик нагрузки для узлов управляющей плоскости кластера для обеспечения высокой доступности.

Если в вашей среде есть аппаратный LoadBalancer, его использование **рекомендуется**. Если нет, можно включить `Self-built VIP`, который обеспечивает программную балансировку нагрузки с помощью keepalived.

Рекомендация: При использовании аппаратного LoadBalancer рекомендуется настраивать Cluster Endpoint с доменным именем. Если VIP аппаратного LoadBalancer изменится после установки кластера, можно обновить DNS-запись вместо перенастройки кластера.

Примечание: `Self-built VIP` не поддерживает настройку доменного имени.

Балансировщик нагрузки должен корректно перенаправлять трафик на порты `6443`, `11780` и `11781` на всех узлах управляющей плоскости кластера.

Если в вашем кластере только один узел управляющей плоскости и вы используете IP этого узла в параметре Cluster Endpoint, то впоследствии масштабировать кластер с одного узла до высокодоступного многозвенного нельзя. Поэтому рекомендуется предоставить балансировщик нагрузки даже для кластеров с одним узлом.

При включении `Self-built VIP` необходимо подготовить:

1. Доступный VRID
2. Хостовую сеть с поддержкой протокола VRRP
3. Все узлы управляющей плоскости и VIP должны находиться в одной подсети, при этом VIP должен отличаться от IP любого узла.

Подключение кластера `global` и рабочих кластеров

Платформа требует взаимного доступа между кластером `global` и рабочими кластерами. Если они находятся в разных сетях, необходимо:

1. Обеспечить `External Access` для рабочего кластера, чтобы кластер `global` мог к нему обращаться. Сетевые требования должны гарантировать доступ `global` к портам `6443`, `11780` и `11781` на всех узлах управляющей плоскости.
2. Добавить дополнительный адрес в `global`, доступный для рабочего кластера. При создании рабочего кластера добавьте этот адрес в аннотации кластера с ключом `cpaas.io/platform-url` и значением, равным публичному адресу доступа к `global`.

Реестр образов

Образы кластера поддерживают варианты: встроенный в платформу, приватный репозиторий и публичный репозиторий.

- **Встроенный в платформу**: Используется реестр образов, предоставляемый кластером `global`. Если кластер не может получить доступ к `global`, смотрите Add External Address for Built-in Registry.
- **Приватный репозиторий**: Используется ваш собственный реестр образов. Для подробностей о загрузке необходимых образов в ваш реестр обратитесь в техническую поддержку.
- **Публичный репозиторий**: Используется публичный реестр образов платформы. Перед использованием завершите Updating Public Repository Credentials.

Сетевое взаимодействие контейнеров

Если вы планируете использовать Kube-OVN Underlay для вашего кластера, обратитесь к Preparing Kube-OVN Underlay Physical Network.

Процедура создания

1. Перейдите в представление **Administrator** и в левом навигационном меню выберите **Clusters/Clusters**.
2. Нажмите **Create Cluster**.
3. Настройте следующие разделы согласно инструкциям ниже: Basic Info, Container Network, Node Settings и Extended Parameters.

Basic Info

Параметр	Описание
Kubernetes Version	Все доступные версии тщательно протестированы на стабильность и совместимость. **Рекомендация:** Выберите последнюю версию для оптимального функционала и поддержки.
Container Runtime	По умолчанию предоставляется контейнерный рантайм Containerd.

Cluster Network Protocol	Поддерживает три режима: IPv4 single stack, IPv6 single stack, IPv4/IPv6 dual stack. **Примечание:** При выборе dual stack убедитесь, что у всех узлов корректно настроены IPv6-адреса; сетевой протокол нельзя изменить после установки.
Cluster Endpoint	`IP Address / Domain`: Введите заранее подготовленное доменное имя или VIP, если доменное имя отсутствует. `Self-built VIP`: По умолчанию отключено. Включайте только если балансировщик нагрузки не предоставлен. При включении установщик автоматически развернет `keepalived` для поддержки программной балансировки нагрузки. `External Access`: Введите внешний доступный адрес, подготовленный для кластера, если он находится в другой сетевой среде, нежели `global` кластер.

Container Network

Kube-OVN

Корпоративная система оркестрации сетей контейнеров Cloud Native Kubernetes, разработанная Alauda. Она переносит зрелые сетевые возможности из области OpenStack в Kubernetes, поддерживает управление сетями в мультиоблачной среде, интеграцию с традиционной сетевой архитектурой и инфраструктурой, а также сценарии развертывания на периферийных кластерах, значительно повышая

безопасность, эффективность управления и производительность сетей контейнеров Kubernetes.

Параметр	Описание
Subnet	Также известна как Cluster CIDR, представляет собой **подсеть по умолчанию**. После создания кластера можно добавлять дополнительные подсети.
Transmit Mode	**Overlay**: Виртуальная сеть, абстрагированная над инфраструктурой, не потребляющая физические сетевые ресурсы. При создании Overlay-подсети по умолчанию все Overlay-подсети кластера используют одинаковую конфигурацию cluster NIC и node NIC. **Underlay**: Этот режим передачи опирается на физические сетевые устройства. Он может напрямую выделять физические сетевые адреса для Pod, обеспечивая лучшую производительность и связь с физической сетью. Узлы в Underlay-подсети должны иметь несколько NIC, а NIC, используемый для мостовой сети, должен использоваться исключительно для Underlay и не должен нести другой трафик, например SSH. При создании Underlay-подсети по умолчанию cluster NIC фактически является NIC по умолчанию для мостовой сети, а node NIC — это конфигурация NIC узла в мостовой сети. • **Default Gateway**: Адрес шлюза физической сети, являющийся шлюзом для сегмента Cluster CIDR (должен находиться в диапазоне адресов Cluster CIDR). • **VLAN ID**: Идентификатор виртуальной локальной сети (номер VLAN), например, `0`. • **Reserved IPs**: Укажите зарезервированные IP-адреса, которые не будут автоматически выделяться, например, IP-адреса в подсети, уже используемые другими устройствами.

Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети по умолчанию.
Join CIDR	В режиме передачи Overlay это диапазон IP-адресов для связи между узлами и Pod. Не должен пересекаться с подсетью по умолчанию или Service CIDR.

Calico

Calico — решение уровня 3 для сетей, обеспечивающее безопасное сетевое соединение для контейнеров.

Параметр	Описание
Default Subnet	Также известна как Cluster CIDR, представляет собой **подсеть по умолчанию**. После создания кластера можно добавлять дополнительные подсети.
Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети по умолчанию.

Flannel

Flannel предоставляет плоскую сетевую среду для всех контейнеров в кластере, обеспечивая контейнерам, созданным на разных узлах, уникальный виртуальный IP-адрес по всему кластеру. Подсеть Pod равномерно делится между узлами кластера согласно маске, и Pod на каждом узле получают IP-адреса из сегмента, выделенного этому узлу. Это повышает эффективность коммуникации между контейнерами без необходимости учитывать вопросы трансляции IP.

Параметр	Описание
Cluster CIDR	Диапазон IP-адресов, используемый для Pod, создаваемых при запуске кластера. Поддерживается настройка максимального количества IP-адресов, которые могут быть выделены Pod на каждом узле в текущей контейнерной сети. **Примечание**: Платформа автоматически рассчитает максимальное количество узлов, которое может вместить кластер, исходя из указанной конфигурации, и отобразит это в подсказке под полем ввода. **Важно**: После создания кластера сетевая конфигурация изменить нельзя, поэтому планируйте сеть внимательно.
Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети контейнеров.

Custom

Если необходимо установить другие сетевые плагины, выберите режим **Custom**. Вы сможете вручную установить сетевые плагины после успешного создания кластера.

Параметр	Описание
Cluster CIDR	Диапазон IP-адресов, используемый для Pod, создаваемых при запуске кластера.

Service CIDR	Диапазон IP-адресов, используемый Kubernetes Service типа ClusterIP. Не должен пересекаться с диапазоном подсети контейнеров.

Node Settings

Параметр	Описание
Network Interface Card	Имя сетевого интерфейса хоста, используемого сетевым плагином кластера. **Примечание**: • При выборе режима передачи Underlay для подсети Kube-OVN по умолчанию необходимо указать имя сетевого интерфейса, который будет использоваться как NIC по умолчанию для мостовой сети. - По умолчанию мониторинг трафика сетевых интерфейсов платформы распознаёт трафик на интерфейсах с именами, начинающимися на `eth.|en.|wl.|ww.`. Если вы используете интерфейсы с другими именами, после добавления кластера обратитесь к Collect Network Data from Custom-Named Network Interfaces для корректировки ресурсов и обеспечения правильного мониторинга трафика.
Node Name	Можно выбрать использование IP-адреса узла или имени хоста в качестве имени узла на платформе. **Примечание**: При выборе имени хоста убедитесь, что имена хостов узлов, добавляемых в кластер, уникальны.

Nodes	**Добавьте узлы** в кластер или **Восстановите из черновика** временно сохранённую информацию об узлах. Подробные описания параметров добавления узлов приведены ниже.
Monitoring Type	Поддерживаются **Prometheus** и **VictoriaMetrics**. При выборе **VictoriaMetrics** необходимо настроить **Deploy Type**: **- Deploy VictoriaMetrics**: Разворачивает все связанные компоненты, включая **VMStorage**, **VMAlert**, **VMAgent** и др. **- Deploy VictoriaMetrics Agent**: Разворачивает только компонент сбора логов **VMAgent**. При таком способе развертывания необходимо связать с уже развернутым экземпляром VictoriaMetrics на другом кластере платформы для предоставления мониторинга.
Monitoring Nodes	Выберите узлы для развертывания компонентов мониторинга кластера. Поддерживается выбор вычислительных узлов и узлов управляющей плоскости, допускающих развертывание приложений. Чтобы не влиять на производительность кластера, рекомендуется отдавать приоритет вычислительным узлам. После успешного создания кластера компоненты мониторинга с типом хранения **Local Volume** будут развернуты на выбранных узлах.

Параметры добавления узлов

Параметр	Описание
Type	**Control Plane Node**: Отвечает за запуск компонентов kube-apiserver, kube-scheduler, kube-controller-manager, etcd, контейнерной сети и

	некоторых компонентов управления платформой в кластере. При включенной опции **Application Deployable** узлы управляющей плоскости могут также использоваться как вычислительные узлы. **Worker Node**: Отвечает за размещение бизнес-подов, работающих в кластере.
IPv4 Address	IPv4-адрес узла. Для кластеров, созданных в режиме внутренней сети, укажите **приватный IP** узла.
IPv6 Address	Актуально при включенном dual stack IPv4/IPv6. IPv6-адрес узла.
Application Deployable	Актуально при **Node Type** = **Control Plane Node**. Разрешить ли развертывание бизнес-приложений на данном узле управляющей плоскости, то есть планировать бизнес-поды на этот узел.
Display Name	Отображаемое имя узла.
SSH Connection IP	IP-адрес, по которому можно подключиться к узлу через SSH. Если вы можете войти на узел с помощью команды `ssh <username>@<IPv4-адрес узла>`, этот параметр не обязателен; в противном случае укажите публичный IP или внешний NAT IP узла, чтобы кластер `global` и прокси могли подключаться к узлу по этому IP.
Network Interface Card	Укажите имя сетевого интерфейса, используемого узлом. Приоритет эффективности конфигурации сетевого интерфейса следующий

	(слева направо, по убыванию): **Kube-OVN Underlay**: Node NIC > Cluster NIC **Kube-OVN Overlay**: Node NIC > Cluster NIC > NIC, соответствующий маршруту по умолчанию узла **Calico**: Cluster NIC > NIC, соответствующий маршруту по умолчанию узла **Flannel**: Cluster NIC > NIC, соответствующий маршруту по умолчанию узла
Associated Bridge Network	**Примечание**: При создании кластера настройка мостовой сети не поддерживается; этот параметр доступен только при **добавлении узлов** в кластер, в котором уже созданы Underlay-подсети. Выберите существующую Add Bridge Network. Если не хотите использовать NIC по умолчанию мостовой сети, можно отдельно настроить node NIC.
SSH Port	Номер порта SSH-сервиса, например, `22`.
SSH Username	Имя пользователя SSH с правами root, например, `root`.

Proxy	Использовать ли прокси для доступа к SSH-порту узла. Если кластер `global` не может напрямую подключиться к добавляемому узлу по SSH (например, кластер `global` и рабочий кластер находятся в разных подсетях; IP узла — внутренний, недоступный напрямую для `global`), необходимо включить этот переключатель и настроить параметры прокси. После настройки прокси доступ к узлу и развертывание будут осуществляться через прокси. **Примечание**: В настоящее время поддерживается только SOCKS5-прокси. **Access URL**: Адрес прокси-сервера, например, `192.168.1.1:1080`. **Username**: Имя пользователя для доступа к прокси-серверу. **Password**: Пароль для доступа к прокси-серверу.
SSH Authentication	Метод аутентификации и соответствующая информация для входа на добавляемый узел. Варианты: **Password**: Требуется имя пользователя с правами root и соответствующий **SSH пароль**. **Key**: Требуется **приватный ключ** с правами root и **пароль к приватному ключу**.
Save Draft	Сохраняет текущие настройки в диалоге как черновик и закрывает окно **Add Node**.

Не покидая страницу **Create Cluster**, можно выбрать **Restore from draft** для открытия диалога **Add Node** и восстановления сохранённых настроек.

Примечание: Восстанавливаются самые последние сохранённые данные черновика.

Extended Parameters

Примечание:

- За исключением обязательных настроек, не рекомендуется задавать расширенные параметры, так как некорректные настройки могут сделать кластер недоступным, и их нельзя будет изменить после создания кластера.
- Если введённый **Key** совпадает с ключом параметра по умолчанию, он переопределит стандартную конфигурацию.

Процедура

1. Нажмите **Extended Parameters** для раскрытия области настройки расширенных параметров. Можно опционально задать следующие параметры для кластера:

Параметр	Описание
Kubelet Parameters	`kubeletExtraArgs`, дополнительные параметры конфигурации для Kubelet. **Примечание**: При вводе параметра **Node IP Count** в **Container Network** автоматически создаётся параметр **Kubelet Parameter** с ключом `max-pods` и значением **Node IP Count**. Он задаёт максимальное количество Pod, которые могут работать на любом узле кластера. Этот параметр не отображается в интерфейсе.

	Добавление новой пары ключ-значение `max-pods: максимальное количество запускаемых Pod` в разделе **Kubelet Parameters** переопределит значение по умолчанию. Разрешены любые положительные целые числа, но рекомендуется использовать значение по умолчанию (Node IP Count) или не превышать `256`.
Controller Manager Parameters	`controllerManagerExtraArgs`, дополнительные параметры конфигурации для Controller Manager.
Scheduler Parameters	`schedulerExtraArgs`, дополнительные параметры конфигурации для Scheduler.
APIServer Parameters	`apiServerExtraArgs`, дополнительные параметры конфигурации для APIServer.
APIServer URL	`publicAlternativeNames`, адреса доступа к APIServer, указанные в сертификате. Можно вводить только IP или доменные имена, максимум 253 символа.
Cluster Annotations	Аннотации кластера — метаданные в виде пар ключ-значение, которые позволяют компонентам платформы или бизнес-компонентам получать соответствующую информацию о характеристиках кластера.

4. Нажмите **Create**. Вы вернётесь на страницу списка кластеров, где создаваемый кластер будет иметь статус **Creating**.

Действия после создания

Просмотр прогресса создания

На странице списка кластеров отображается список созданных кластеров. Для кластеров в состоянии **Creating** можно проверить ход выполнения.

Процедура

1. Нажмите маленькую иконку **View Execution Progress** справа от статуса кластера.
2. В появившемся диалоговом окне можно просмотреть прогресс выполнения кластера (status.conditions).
   Совет: Если определённый тип находится в процессе или в состоянии ошибки с указанием причины, наведите курсор на причину (отображается синим текстом), чтобы увидеть подробную информацию о причине (status.conditions.reason).

Ассоциация с проектами

После создания кластера его можно добавить в проекты в представлении управления проектами.

О хостингованной контрольной плоскости

Hosted Control Plane (HCP) — это облегчённая модель управления мультикластером, которая отделяет контрольную плоскость от рабочих узлов. Контрольная плоскость каждого кластера контейнеризована и размещена внутри управляющего кластера, что снижает потребление ресурсов, ускоряет создание и обновление кластеров, а также улучшает масштабируемость при работе с несколькими кластерами.

> **Note**
>
> Поскольку выпуски Hosted Control Plane осуществляются в ином режиме, чем у Alauda Container Platform, документация Hosted Control Plane теперь доступна в виде отдельного набора по адресу Hosted Control Plane ↗ 。

Планирование узлов кластера

Кластер использует метки ролей узлов Kubernetes `node-role.kubernetes.io/<role>` для назначения различных ролей узлам. Для удобства описания мы будем называть этот тип меток метками ролей.

По умолчанию кластер содержит два типа узлов: узлы управляющей плоскости (control plane) и рабочие узлы (worker nodes), которые используются для размещения рабочих нагрузок управляющей плоскости и приложений соответственно.

В кластере:

- Узлы управляющей плоскости помечены меткой роли `node-role.kubernetes.io/control-plane`.

  > **Примечание:**
  > До версии Kubernetes v1.24 сообщество также использовало метку `node-role.kubernetes.io/master` для обозначения узлов управляющей плоскости. Для обратной совместимости обе метки считаются действительными для идентификации узлов управляющей плоскости.

- Рабочие узлы по умолчанию не имеют меток ролей. Однако при необходимости вы можете явно назначить рабочему узлу метку роли `node-role.kubernetes.io/worker`.

Помимо этих стандартных меток ролей, вы также можете определить пользовательские метки ролей на рабочих узлах для их дальнейшей классификации по функциональным типам. Например:

- Вы можете добавить метку роли `node-role.kubernetes.io/infra`, чтобы обозначить узел как infra-узел, предназначенный для размещения инфраструктурных

компонентов.

- Вы можете добавить метку роли `node-role.kubernetes.io/log`, чтобы обозначить узел как log-узел, специализированный для размещения компонентов логирования.

В этом документе описывается процесс создания infra-узлов и узлов с пользовательскими ролями, а также миграция рабочих нагрузок на эти узлы.

Содержание

Создание Infra-узлов в неиммутабельном кластере

По умолчанию кластер включает только узлы управляющей плоскости и рабочие узлы. Если вы хотите выделить определённые рабочие узлы как infra-узлы, предназначенные для размещения инфраструктурных компонентов, необходимо вручную добавить соответствующую метку роли и taint этим узлам.

> **Примечание:**
> Операции, описанные в этом разделе, применимы только к неиммутабельным кластерам. То есть следующие операции не поддерживаются в облачных кластерах (например, управляемых кластерах EKS, развернутых через Cluster Plugin `Alauda Container Platform EKS Provider`), сторонних кластерах или кластерах, где узлы используют иммутабельную ОС.

Добавление Infra-узлов

Шаг 1: Добавьте метку роли Infra к ресурсам Node

```
kubectl label nodes 192.168.143.133 node-role.kubernetes.io/infra="" --ov
erwrite
```

Эта команда добавляет метку роли infra к узлу 192.168.143.133: `node-role.kubernetes.io/infra: ""`, указывая, что узел является infra-узлом.

Шаг 2: Добавьте taint к ресурсам Node

Добавьте taint, чтобы предотвратить планирование других рабочих нагрузок на infra-узел.

```
kubectl taint nodes 192.168.143.133 node-role.kubernetes.io/infra=reserve
d:NoSchedule
```

Эта команда добавляет taint `node-role.kubernetes.io/infra=reserved:NoSchedule` к узлу 192.168.143.133, указывая, что на этот узел могут быть запланированы только приложения, которые терпят этот taint.

Шаг 3: Проверьте метку и taint

Проверьте, назначены ли узлу метка роли infra и taint:

```
# kubectl describe node 192.168.143.133
Name:               192.168.143.133
Roles:              infra
Labels:             node-role.kubernetes.io/infra=""
                    ...
Taints:             node-role.kubernetes.io/infra=reserved:NoSchedule
```

Вывод показывает, что узел 192.168.143.133 настроен как infra-узел и имеет taint `node-role.kubernetes.io/infra=reserved:NoSchedule`.

Миграция Pod на Infra-узлы

Если вы хотите запланировать конкретный Pod на infra-узлы, необходимо выполнить следующие настройки:

- nodeSelector, ориентированный на метку роли infra.
- Соответствующие tolerations для taint infra-узла.

Ниже приведён пример манифеста Deployment, настроенного для запуска на infra-узле.

```
apiVersion: apps/v1
kind: Pod
metadata:
  name: infra-pod-demo
  namespace: default
spec:
  ...
  nodeSelector:
    node-role.kubernetes.io/infra: ""
  tolerations:
  - effect: NoSchedule
    key: node-role.kubernetes.io/infra
    value: reserved
    operator: Equal
  ...
```

nodeSelector гарантирует, что Pod будет запланирован только на узлах с меткой `node-role.kubernetes.io/infra: ""`, а toleration позволяет Pod терпеть taint `node-role.kubernetes.io/infra=reserved:NoSchedule`.

С этими настройками Pod будет запланирован на infra-узле.

> **Примечание:**
> Перемещение Pod, установленных через OLM Operators или Cluster Plugins, на infra-узел не всегда возможно. Возможность перемещения таких Pod зависит от конфигурации каждого Operator или Cluster Plugin.

Планирование пользовательских узлов

Помимо infra-узлов, вы можете захотеть выделить рабочие узлы для других специализированных задач — например, для размещения компонентов логирования, сервисов хранения или агентов мониторинга.

Это можно сделать, назначив дополнительные пользовательские метки ролей и соответствующие taint рабочим узлам, эффективно превращая их в узлы с пользовательскими ролями.

Общие шаги для определения узлов с пользовательскими ролями

Процесс аналогичен созданию infra-узлов.

Шаг 1: Добавьте пользовательскую метку роли

```
kubectl label nodes <node> node-role.kubernetes.io/<role>="" --overwrite
```

Замените <role> на желаемое имя роли, например monitoring, storage или log.

Шаг 2: Добавьте соответствующий taint

```
kubectl taint nodes <node> node-role.kubernetes.io/<role>=<value>:NoSchedule
```

Замените <role> на имя вашей пользовательской роли, а <value> — на осмысленное описание, например reserved или dedicated. Это значение необязательно, но полезно для документации и ясности.

Шаг 3: Проверьте конфигурацию

```
kubectl describe node <node>
```

Убедитесь, что поля Labels и Taints отражают вашу конфигурацию пользовательской роли.

Пример: создание узла, выделенного для компонентов логирования

Если вы хотите создать узел специально для установки компонентов логирования, можно добавить роль log. В этом случае создайте log-узел следующим образом.

Шаг 1: Добавьте метку роли Log

```
kubectl label nodes 192.168.143.133 node-role.kubernetes.io/log="" --overwrite
```

Эта метка указывает, что узел предназначен для рабочих нагрузок, связанных с логированием.

Шаг 2: Добавьте taint к узлу

```
kubectl taint nodes 192.168.143.133 node-role.kubernetes.io/log=reserved:NoSchedule
```

Этот taint предотвращает развертывание незапланированных рабочих нагрузок на узле.

Шаг 3: Проверьте метку и taint

```
Name:               192.168.143.133
Roles:              log
Labels:             node-role.kubernetes.io/log=reserved
                    ...
Taints:             node-role.kubernetes.io/log=reserved:NoSchedule
```

Это подтверждает, что узел успешно настроен как log-узел с соответствующей меткой и taint.

Следуя приведённым рекомендациям, вы сможете эффективно разделять узлы Kubernetes по их назначению, улучшать изоляцию рабочих нагрузок и обеспечивать развертывание конкретных компонентов на соответствующим образом настроенных узлах.

Шифрование etcd

Это руководство поможет вам установить, понять и управлять etcd Encryption Manager в ACP для автоматизации ротации ключей шифрования данных etcd в ваших кластерах.

Оно обеспечивает шифрование конфиденциальных данных, хранящихся в etcd, таких как secrets и configmaps, с использованием надежного алгоритма, повышая безопасность вашего кластера.

Содержание

Установка

Смотрите Cluster Plugin для инструкций по установке.

> **Примечание:**

> - В настоящее время поддерживаются:
>   - On-Premises кластеры
>   - DCS кластеры
> - Не поддерживается:
>   - `global cluster`

Как это работает

После установки в namespace `kube-system` разворачивается контроллер `etcd-encryption-manager`, который:

- Периодически выполняет ротацию ключей шифрования данных etcd.
- Сохраняет 8 последних ключей для обеспечения совместимости с откатом.
- Обновляет конфигурации шифрования на всех управляющих узлах.
- Запускает горячую перезагрузку новых ключей в `kube-apiserver`.
- Автоматически мигрирует ресурсы для повторного шифрования данных новыми ключами.

Стабильность кластера сохраняется на протяжении всех этих операций.

Конфигурация по умолчанию

Параметр	Значение
Шифруемые ресурсы	secrets, configmaps
Алгоритм шифрования	256-битный AES-GCM
Интервал ротации	168 часов (7 дней)

Руководство по эксплуатации

Файлы конфигурации

Путь	Содержание
`/etc/kubernetes/encryption-provider.conf`	Текущая конфигурация шифрования
`/etc/kubernetes/encryption-provider-history.bak`	Исторические записи ключей (для восстановления)
`/etc/kubernetes/encryption-provider-bak/`	Истекшие версии конфигураций шифрования

Проверка статуса

Выполните следующую команду для проверки текущего статуса ротации:

```
kubectl get EtcdEncryptionConfig default -o yaml
```

Пример вывода:

```yaml
apiVersion: cluster.alauda.io/v1alpha1
kind: EtcdEncryptionConfig
metadata:
  name: default
spec:
  resources:
    - secrets
    - configmaps
  rotationInterval: 168h0m0s
  type: aesgcm
status:
  deployStatus:
    192.168.100.1:
      revision: 3
      state: Success
    192.168.100.2:
      revision: 3
      state: Success
    192.168.100.3:
      revision: 3
      state: Success
  migration:
    completeTimestamp: "2025-05-27T05:47:01Z"
    resources:
      - secrets
      - configmaps
    revision: 3
    state: Success
  revision: 3
```

Как сделать

Добавление внешнего адреса

Оптимизация производительны менеджера

Обновлени

Добавление внешнего адреса для встроенного реестра

Содержание

Overview

Когда кластер `global` использует реестр `Platform Built-in`, рабочие кластеры обычно также используют этот реестр для загрузки образов. Реестр обслуживает не только компоненты внутри кластера `global`, но и должен быть доступен узлам рабочих кластеров.

В некоторых случаях узлы рабочих кластеров не могут напрямую получить доступ к адресу реестра кластера `global` — например, когда кластер `global` находится в частном дата-центре, а рабочие кластеры — в публичных облаках или на периферии.

В этом руководстве описывается, как настроить внешний доступный адрес для реестра платформы по умолчанию, чтобы рабочие кластеры могли загружать образы.

Prerequisites

Перед началом подготовьте следующее:

- Доменное имя, доступное для узлов рабочих кластеров
- IP-адрес, на который указывает доменное имя
- Действительный SSL-сертификат для доменного имени

> **WARNING**
>
> - Доменное имя должно отличаться от адреса доступа к платформе
> - Убедитесь, что IP-адрес домена может перенаправлять трафик на все узлы управляющей плоскости кластера `global`

Procedure

Настройка сертификата и правил маршрутизации для реестра платформы

1. Скопируйте действительный сертификат домена на любой узел управляющей плоскости кластера `global`
2. Создайте TLS-секрет, содержащий сертификат домена:

   ```
   kubectl create secret tls registry-address.tls --cert=<certificate-file
   name> --key=<key-filename> -n kube-system
   ```

   Пример:

   ```
   kubectl create secret tls registry-address.tls --cert=custom.crt --key=
   custom.key -n kube-system
   ```

Примечание: После создания сертификата следите за сроком действия секрета **registry-address.tls** в пространстве имён **kube-system** кластера `global`. Замените сертификат до его истечения.

3. Создайте ingress-правила на любом узле управляющей плоскости кластера `global`:

```
REGISTRY_DOMAIN_NAME=<www.registry.com> # Замените на ваше доступное до
менное имя
cat << EOF | kubectl create -f -
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    nginx.ingress.kubernetes.io/backend-protocol: HTTPS
  name: registry-address
  namespace: kube-system
  labels:
    service_name: registry
spec:
  rules:
    - host: $REGISTRY_DOMAIN_NAME
      http:
        paths:
          - backend:
              service:
                name: registry
                port:
                  number: 443
            path: /v2/
            pathType: ImplementationSpecific
          - backend:
              service:
                name: registry
                port:
                  number: 443
            path: /v2/_catalog
            pathType: ImplementationSpecific
          - backend:
              service:
                name: registry
                port:
                  number: 443
            path: /v2/.+/tags/list
            pathType: ImplementationSpecific
          - backend:
              service:
                name: registry
                port:
                  number: 443
```

```
            path: /v2/.+/manifests/[A-Za-z0-9_+.-:]+
            pathType: ImplementationSpecific
          - backend:
              service:
                name: registry
                port:
                  number: 443
            path: /v2/.+/blobls/[A-Za-z0-9-:]+
            pathType: ImplementationSpecific
          - backend:
              service:
                name: registry
                port:
                  number: 443
            path: /v2/.+/blobls/uploads/[A-Za-z0-9-:]+
            pathType: ImplementationSpecific
          - backend:
              service:
                name: registry
                port:
                  number: 443
            path: /auth/token
            pathType: ImplementationSpecific
  tls:
    - secretName: registry-address.tls
      hosts:
        - $REGISTRY_DOMAIN_NAME
EOF
```

Ответ, похожий на `... created`, означает успешное создание ingress.

4. Проверьте, существует ли ресурс Service для реестра:

```
kubectl -n kube-system get svc | grep registry
```

Если Service отсутствует, создайте его с помощью:

```
cat << EOF | kubectl create -f -
apiVersion: v1
kind: Service
metadata:
  labels:
    name: registry
    service_name: registry
  name: registry
  namespace: kube-system
spec:
  ports:
    - protocol: TCP
      port: 443
      targetPort: 60080
  selector:
    component: registry
  type: ClusterIP
EOF
```

5. Проверьте конфигурацию, загрузив образ из реестра с использованием доменного имени:

```
crictl pull <registry-domain-name>/automation/qaimages:helloworld
```

Или

```
podman pull <registry-domain-name>/automation/qaimages:helloworld
```

Оптимизация производительности Pod с помощью политик менеджера

Это руководство предоставляет администраторам Kubernetes кластера практическое, готовое к применению руководство по включению и проверке **CPU ManagerPolicy**, **Memory ManagerPolicy** и **Topology ManagerPolicy**. Согласовывая закрепление CPU, NUMA-аффинность и выравнивание топологии, вы можете обеспечить стабильную задержку и улучшенную производительность для критически важных нагрузок.

Содержание

Область применения и предварительные требования

Роли и права

- Требуется доступ в окно обслуживания, административные права `kubectl` и SSH-доступ к узлам.

Требования к нагрузке

- Для достижения выделенного CPU и NUMA-аффинности Pods должны работать в классе **Guaranteed QoS**: requests = limits и CPU указаны целыми ядрами (например, 2, 4).

Не рассматривается

- HugePages не рассматриваются. Если вам нужна поддержка HugePages, обратитесь в службу поддержки.

Быстрый старт: пример конфигурации Kubelet

Добавьте следующий фрагмент в `/var/lib/kubelet/config.yaml`, подстроив значения под вашу среду:

```
# ── CPU ManagerPolicy ──
cpuManagerPolicy: "static"              # Опции: none | static
cpuManagerPolicyOptions:
  full-pcpus-only: "true"               # Рекомендуется: выделять только
целые ядра
cpuManagerReconcilePeriod: "5s"
reservedSystemCPUs: ""                  # например, "0-1" для резервирова
ния конкретных CPU для системы

# ── Memory ManagerPolicy ──
memoryManagerPolicy: "Static"           # Опции: none | Static
reservedMemory:
  - numaNode: 0
    limits:
      memory: "2048Mi"
  - numaNode: 1
    limits:
      memory: "2048Mi"

# ── Topology ManagerPolicy ──
topologyManagerPolicy: "single-numa-node"     # Опции: none | best-effort
| restricted | single-numa-node
topologyManagerScope: "pod"                   # Опции: container | pod
```

Примечания:

- `full-pcpus-only: "true"` улучшает согласованность задержек.
- `topologyManagerScope: pod` гарантирует, что контейнеры внутри одного Pod выравниваются по общей NUMA-топологии.
- `reservedMemory` должен рассчитываться на основе конфигурации kubelet и порогов эвакуации (см. следующий раздел).

Как рассчитать `reservedMemory`

Формула:

```
R_total = kubeReserved(memory) + systemReserved(memory) + evictionHard(me
mory.available)
```

Сумма `reservedMemory` по всем NUMA-узлам должна равняться `R_total`.

Шаги (для N NUMA-узлов):

1. Рассчитать `R_total` (в Mi).
2. Вычислить деление и остаток:
   - base = floor(R_total / N)
   - rem = R_total − base × N
3. Назначить значения:
   - NUMA-узел 0 = base + rem
   - Остальные NUMA-узлы = base

Пример (2 NUMA-узла):

- kubeReserved=512Mi, systemReserved=512Mi, evictionHard=100Mi → R_total = 1124Mi
- base = 562, rem = 0

```yaml
reservedMemory:
- numaNode: 0
  limits:
    memory: "562Mi"
- numaNode: 1
  limits:
    memory: "562Mi"
```

Применение конфигурации

Для каждого узла:

1. **Отключение и эвакуация**

```
kubectl cordon <node>
kubectl drain <node> --ignore-daemonsets --delete-emptydir-data
```

2. **Остановка Kubelet и очистка состояния**

```
sudo systemctl stop kubelet
sudo rm -f /var/lib/kubelet/cpu_manager_state
sudo rm -f /var/lib/kubelet/memory_manager_state
```

3. **Перезапуск Kubelet**

```
sudo systemctl daemon-reload
sudo systemctl start kubelet
```

4. **Повторное включение узла**

```
kubectl uncordon <node>
```

- Для DaemonSets и системных Pod перезапустите или удалите Pod явно.

5. **Проверка восстановления**

```
kubectl get nodes
kubectl get pods -A -o wide | grep <node>
```

Проверка

Состояние CPU ManagerPolicy

```
sudo cat /var/lib/kubelet/cpu_manager_state | jq .
```

Проверьте:

- `.policyName` = `"static"`
- `.defaultCpuSet` содержит невыделенные CPU

- `.entries` показывают назначение контейнеров на CPU

Состояние Memory ManagerPolicy

```
sudo cat /var/lib/kubelet/memory_manager_state | jq .
```

Проверьте:

- `.policyName` = `"Static"`
- Сумма зарезервированной памяти соответствует `R_total`
- Pods с Guaranteed назначены на NUMA-узлы согласно политике `single-numa-node`

Основные политики и поведение

CPU ManagerPolicy

- Цель: выделять эксклюзивные физические CPU для Pods с Guaranteed QoS
- Конфигурация: `cpuManagerPolicy: static`, `full-pcpus-only: "true"`
- Поведение: применяется только к Pods с Guaranteed; Burstable/BestEffort не затрагиваются

Memory ManagerPolicy

- Цель: резервировать и выравнивать память на уровне NUMA-узлов
- Конфигурация: `memoryManagerPolicy: "Static"`, `reservedMemory`
- Поведение: лучше всего работает совместно с Topology ManagerPolicy для выравнивания

Topology ManagerPolicy

- Цель: выравнивать CPU, память и устройства на одном NUMA-узле
- Конфигурация: `topologyManagerPolicy: single-numa-node`, `topologyManagerScope: pod`
- Режимы: best-effort, restricted, single-numa-node (строгий)

Терминология

- **NUMA node**: домен с неоднородным доступом к памяти (Non-Uniform Memory Access)
- **CPU pinning**: привязка контейнеров к выделенным CPU
- **NUMA affinity**: предпочтение памяти из того же NUMA-узла, что и CPU
- **Topology alignment**: совместное размещение CPU, памяти и устройств на одном NUMA-узле
- **Guaranteed Pod**: requests = limits; CPU указаны целыми ядрами

Обновление учетных данных публичного репозитория

Содержание

Overview

`Public Repository` — это сервис реестра образов, предоставляемый платформой и доступный в публичном интернете. Если вы хотите, чтобы ваши кластеры использовали `Public Repository` в качестве реестра образов, необходимо обновить встроенные облачные учетные данные `public-registry-credential`. Это гарантирует, что ваша платформа имеет разрешение на загрузку образов из публичного реестра.

Procedure

1. Войдите в **Alauda Customer Portal** и скачайте файл аутентификации вашей организации из раздела **Enterprise Management**, расположенного в выпадающем меню **User Information** в правом верхнем углу.

2. Перейдите в **Clusters** > **Cloud Credential** в левой навигационной панели консоли **Administrator**.
3. Найдите облачные учетные данные с именем `public-registry-credential` и выберите **Update** в выпадающем меню справа.
4. В разделе **Upload Public Repository Address** загрузите файл аутентификации, который вы скачали из **Alauda Customer Portal**.
5. Нажмите **Update** для применения изменений.