Обновление

В этом документе представлена вся информация, касающаяся обновления ACP.

Обзор

Подготовка к обновлению

Обновлени

Upgrade Workload Clusters

Обзор

ACP 4.3 использует рабочий процесс на основе Cluster Version Operator (CVO) для обновления кластера.

В Web Console запрос на обновление теперь выполняется в два этапа: сначала выполняется просмотр элементов RPCH, а затем запрос на обновление отправляется на отдельном шаге подтверждения.

При переносе платформы на новую версию ACP Distribution Version обновление обычно выполняется в два этапа:

1. Обновите глобальный tier до целевой версии Distribution Version, следуя проверенной процедуре для глобального кластера, включая подготовку артефактов и предварительные проверки.
2. После того как глобальный tier достигнет целевой версии Distribution Version, обновите workload clusters из поддерживаемой точки входа для workload-cluster и отслеживайте состояние кластера, пока каждый целевой кластер не достигнет той же версии Distribution Version.

Workload cluster можно обновить только до той версии Distribution Version, которую глобальный tier уже достиг. В средах с **global disaster recovery (DR)** это означает, что и резервный, и основной глобальные кластеры должны достичь целевой версии Distribution Version, прежде чем workload clusters будут обновлены до этой версии Distribution Version. Это правило последовательности не отменяет предварительное условие Compatible Versions: до обновления глобального tier до ACP 4.3 workload clusters должны оставаться в диапазоне версий Kubernetes, совместимых с ACP 4.3.

Содержание

Ключевые понятия

Точки входа для обновления

Усиление безопасности после обновления

Связанная документация

Ключевые понятия

- **ClusterVersionShadow (`cvsh`)**: ресурс обновления, используемый для отслеживания текущей версии, целевой версии, результатов предварительных проверок, этапов выполнения и истории.
- **Distribution Version**: версия ACP, которую в данный момент достиг кластер. Workload cluster можно обновить только до той версии Distribution Version, которую глобальный tier уже достиг.
- **Preflight**: проверки валидации, которые выполняются перед тем, как обновление начнет применять целевую версию. Для workload clusters просматривайте результаты preflight в выводе статуса обновления после отправки запроса на обновление.
- **Available upgrade targets**: версии обновления, которые в данный момент доступны для кластера. В Web Console целевая версия для текущего потока обновления определяется платформой.
- **upgrade.sh**: скрипт подготовки, который загружает артефакты и развертывает или обновляет cluster version operator перед продолжением обновления.
- **Global DR environment**: среда, в которой есть и основной глобальный кластер, и резервный глобальный кластер.
- **Primary global cluster**: глобальный кластер, на который в данный момент указывает домен доступа платформы.
- **Standby global cluster**: второй глобальный кластер в паре DR. После failover роли двух кластеров меняются местами.

Точки входа для обновления

- **Global clusters**: следуйте проверенной процедуре на основе `upgrade.sh`. После завершения этапа подготовки запросите обновление в Web Console через двухэтапный поток проверки RPCH, используйте ACP CLI или обновите `ClusterVersionShadow.spec.desiredUpdate` напрямую.
- **Workload clusters**: используйте Web Console через двухэтапный поток проверки RPCH или ACP CLI после того, как целевая версия Distribution Version станет доступна для workload cluster.

Усиление безопасности после обновления

После того как глобальный кластер и все workload clusters достигнут ACP 4.3, выполните усиление безопасности PKCE, следуя инструкции Отключение метода PKCE Plain Method.

После того как глобальный кластер достигнет ACP 4.3, выполните требуемые обновления совместимости плагинов L5 в разделе Обновление глобального кластера.

Связанная документация

- Перед обновлением
- Обновление глобального кластера
- Обновление workload clusters
- Global Cluster Disaster Recovery

Подготовка к обновлению

Поддерживаемые пути обновления:

- С `4.0` → `4.3`
- С `4.1` → `4.3`
- С `4.2` → `4.3`

Перед началом убедитесь, что текущая версия платформы находится в пределах поддерживаемого диапазона обновления.

Содержание

Важные замечания

Загрузка пакетов для офлайн-сред

Важные замечания

- Убедитесь, что в каталоге `/cpaas/minio` на узлах управляющей плоскости глобального кластера имеется не менее **120 ГБ** свободного места на диске.
- Перед обновлением глобального уровня до ACP 4.3 все рабочие кластеры должны оставаться в пределах ACP 4.3 **совместимых версий**, описанных в Kubernetes Support Matrix.

- Если какой-либо рабочий кластер находится вне этого совместимого диапазона, сначала обновите этот рабочий кластер до версии, входящей в совместимый диапазон ACP 4.3, прежде чем обновлять глобальный уровень.
- Рабочий кластер можно обновлять только до версии дистрибутива, которую уже достиг глобальный уровень.

Загрузка пакетов для офлайн-сред

С **Alauda Customer Portal** скачайте **ACP Core Package**.

Если вы хотите обновить **Extensions** кластера во время обновления, выполните следующие шаги:

1. Перейдите по пути: [Marketplace - Batch Download - Upgrade - Post-ACP v4.0 Upgrades]
2. Скачайте скрипт `ac-get-app.sh`.
3. Загрузите скрипт на управляющий узел **Global** кластера в вашей среде.
4. Запустите скрипт командой `bash ac-get-app.sh`.
5. После завершения импортируйте сгенерированный файл `apps.yaml` обратно в Alauda Customer Portal для синхронизации списка расширений.

Кроме того, перейдите в раздел **CLI Tools** в **Alauda Customer Portal** и скачайте инструмент `violet`. Этот инструмент необходим для загрузки Extensions. Подробнее о `violet` смотрите в разделе Upload Packages.

> **WARNING**
>
> Если вы обновляетесь **с ACP 4.0 до ACP 4.3** и **Alauda Build of TopoLVM** установлен на любых целевых кластерах, загрузите пакет TopoLVM на эти кластеры перед началом обновления. Этот шаг не требуется при обновлении с ACP 4.1 или ACP 4.2. Вы можете указать несколько целевых кластеров в параметре `--clusters`, разделяя их запятыми.

```
violet push <path/to/directory/only_put_topolvm_plugin_here> \
  --target-catalog-source "platform" \
  --platform-address "https://example.com" \
  --platform-username "<platform_user>" \
  --platform-password "<platform_password>" \
  --clusters "cluster-a,cluster-b"
```

WARNING

Начиная с версии v4.2, мы ввели новый плагин под названием **Alauda Container Platform Log Essentials**. Если ранее вы устанавливали плагин для хранения логов, вам также необходимо загрузить этот плагин перед началом обновления.

Обновление глобального кластера

ACP состоит из **глобального** кластера и одного или нескольких кластеров рабочей нагрузки. Чтобы перевести платформу на новую Distribution Version ACP, сначала обновите глобальный уровень до целевой Distribution Version, а затем обновите кластеры рабочей нагрузки до той же Distribution Version.

ACP 4.3 использует рабочий процесс на основе CVO для обновления кластеров. Типичное обновление `global`-кластера включает подготовку артефактов, предварительные проверки, запрос на обновление и наблюдение за состоянием.

Перед обновлением `global`-кластера до ACP 4.3 убедитесь, что каждый кластер рабочей нагрузки работает на совместимой версии Kubernetes. Для ACP 4.3 совместимыми версиями являются 1.34, 1.33, 1.32 и 1.31. Это предварительное требование отделено от более широкого диапазона управления кластерами сторонних производителей.

Это требование по совместимым версиям применяется независимо от того, используется ли в среде global DR. Global DR изменяет процедуру, используемую для обновления глобального уровня, но не меняет требование о том, что кластеры рабочей нагрузки должны оставаться в пределах совместимого диапазона версий Kubernetes до обновления глобального уровня до целевой Distribution Version.

Обновления глобального кластера выполняются по проверенной процедуре на основе `upgrade.sh`, описанной на этой странице. Вы можете запросить обновление глобального кластера из Web Console, обновив `ClusterVersionShadow.spec.desiredUpdate`, или используя ACP CLI с параметром `--cluster=global`. Полный рабочий процесс AC CLI и описание вывода см. в разделе Обновление кластеров. Полный синтаксис команд и флагов см. в Справочнике по административным командам AC CLI.

Если в среде используется **global DR**, следуйте процедуре Global DR. В противном случае следуйте стандартному рабочему процессу ниже.

Содержание

- Стандартный рабочий процесс
  - Подготовьте артефакты обновления
  - Выполните предварительные проверки
  - При необходимости обработайте блокировки preflight
  - Запросите обновление
  - Наблюдайте за выполнением
  - (Условно) Обновите Alauda Service Mesh Essentials
- После обновления
- Процедура Global DR
  - Проверьте среду DR перед обновлением
  - Удалите плагин синхронизации etcd с резервного global-кластера
  - Подготовьте артефакты обновления на обоих global-кластерах
  - Обновите резервный global-кластер
  - Обновите основной global-кластер
  - Переустановите плагин синхронизации etcd и проверьте состояние синхронизации
- Связанная документация

Стандартный рабочий процесс

1 Подготовьте артефакты обновления

Выполните `bash upgrade.sh` из каталога извлеченного core package.

`upgrade.sh` подготавливает ресурсы, необходимые для рабочего процесса на основе CVO, включая:

Тип	Содержимое	Назначение
Образы продукта	`product-image`	Используется для определения целевой версии и образа в `ProductManifest` и CVO.
Образ CVO	`cluster-version-operator`	Используется для развертывания или обновления cluster version operator.
Артефакты плагинов	`plugins/*.tgz`	Используется планом обновления, когда требуются артефакты плагинов.

Поведение registry зависит от того, как настроена среда:

Сценарий	Поведение
Указан `--registry`	Непосредственно используется указанный registry.
`--registry` не указан	Адрес registry считывается из `ProductBase.spec.registry.address`.
Встроенный platform registry	Адрес доступа перестраивается с использованием global VIP.
Внешний registry	Автоматически устанавливается `SKIP_SYNC_IMAGE=true`, и синхронизация образов пропускается.
Требуется загрузка образов, но учетные данные не указаны	`username` и `password` считываются из Secret `cpaas-system/registry-admin`.

Общие параметры:

Параметр	Назначение
`--registry`	Указать адрес целевого registry.

Параметр	Назначение
`--username` / `--password`	Указать учетные данные registry.
`--only-sync-image`	Синхронизировать только образы и артефакты плагинов.
`--skip-sync-image`	Пропустить синхронизацию образов и плагинов.
`--skip-check-artifacts`	Пропустить проверку артефактов.

```
bash upgrade.sh
```

WARNING

- Не переходите к следующему шагу, пока не завершится синхронизация образов и плагинов.
- Используйте `--only-sync-image` только в том случае, если требуется только синхронизация артефактов без дальнейшей подготовки.
- Используйте `--skip-sync-image` только если необходимые образы и артефакты плагинов уже загружены.

2 Выполните предварительные проверки

Выполните preflight перед запросом на обновление:

```
bash upgrade.sh --preflight
```

Preflight возвращает две части:

Вывод	Назначение
`Summary`	Показывает общий результат, текущую версию, целевую версию и целевой образ.
`Checks`	Показывает результат каждого отдельного проверочного пункта.

Набор проверок по умолчанию включает:

- `ResourcePatchUpgradeable`
- `ClusterVersionUpgradeable`
- `VersionUpgradePath`
- `KubernetesVersionSupported`
- `DockerRuntimeUnsupported`
- `ClusterRunning`
- `ClusterModuleStable`
- `ControlPlaneStaticPodsPresent`
- `CustomEtcdBackupCronJobsAbsent`
- `CRIUpgradePodsAbsent`
- `ModuleInfoStable`
- `PlatformLicense`

3 При необходимости обработайте блокировки preflight

Если `ResourcePatchUpgradeable` завершается с `reason=UnexemptResourcePatches`, проверьте блокирующий `ResourcePatch` и добавьте требуемую аннотацию исключения:

```
kubectl -n cpaas-system get cvsh global \
  -o jsonpath='{range .status.preflight.checks[?(@.name=="ResourcePat
chUpgradeable")]}{.state}{"\t"}{.reason}{"\t"}{.message}{"\n"}{end}'

kubectl get resourcepatches <rp-name> -o yaml
```

Ключ аннотации по умолчанию: `config.cpaas.io/exempt-for-ver`.

```
kubectl annotate resourcepatches <rp-name> \
  config.cpaas.io/exempt-for-ver=4.3.0 \
  --overwrite
```

Если для временного устранения неполадок требуется отключить определенные проверки, настройте `cpaas-system/cvo-config`:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cvo-config
  namespace: cpaas-system
data:
  preflight: |
    disabled:
      - ResourcePatchUpgradeable
      - VersionUpgradePath
```

4 Запросите обновление

После завершения этапа подготовки выберите одну из следующих точек входа:

- Используйте Web Console после того, как целевая версия станет доступна для кластера.
- Непосредственно измените `ClusterVersionShadow.spec.desiredUpdate`, если необходимо работать с базовым ресурсом CVO.
- Используйте ACP CLI, чтобы явно запросить обновление для `global`.

Если вы используете Web Console, запрос выполняется в два шага:

- На **Step 1** просмотрите список RPCH.
- Нажмите **Acknowledge**, чтобы перейти к **Step 2**.
- На **Step 2** просмотрите **Current Version** и **Target Version**. На этом этапе страница не отображает список плагинов или панель предупреждений.
- Целевая версия определяется подготовленными артефактами обновления и не может быть выбрана вручную в Web Console.
- Нажмите **Start Upgrade**.
- Подтвердите действие в диалоговом окне.
- После подтверждения страница покажет, что запрос на обновление отправлен, а действие перейдет в состояние выполнения.

Пример `kubectl`:

```
kubectl patch cvsh global -n cpaas-system --type merge -p '{
  "spec": {
    "desiredUpdate": {
      "version": "4.3.0"
    }
  }
}'
```

Также можно отредактировать ресурс напрямую:

```
kubectl edit cvsh global -n cpaas-system
```

Минимальная конфигурация:

```
spec:
  desiredUpdate:
    version: 4.3.0
```

Пример ACP CLI:

```
# Запросить обновление до самой высокой версии, которая сейчас опубли
кована в availableUpdates
ac adm upgrade --cluster=global --to-latest

# Запросить обновление до конкретной целевой версии
ac adm upgrade --cluster=global --to=4.3.0

# Показать summary, preflight и ход выполнения этапов для обновления
global-кластера
ac adm upgrade status --cluster=global
```

5 Наблюдайте за выполнением

Используйте следующую команду для просмотра общего состояния:

```
kubectl get cvsh -n cpaas-system
```

Важные поля состояния:

Поле	Назначение
`status.conditions`	Точка входа в общее состояние.
`status.preflight.observedAt`	Время последнего запуска preflight.
`status.preflight.checks`	Подробный результат каждого элемента preflight.
`status.current`	Текущая примененная версия и образ.
`status.desired`	Целевая версия и образ, которые находятся в процессе согласования.
`status.history`	История обновлений, новые записи в начале.
`status.stages`	Этапы обновления и состояние выполнения для каждого этапа.

Сначала обратите внимание на следующие условия:

Условие	Интерпретация
`PreflightReady`	`True` означает, что preflight успешно пройден.
`Ready`	`True` означает, что кластер достиг целевой версии.
`Reconciling`	`True` означает, что обновление все еще выполняется.
`Stalled`	`True` означает, что обновление заблокировано и требует вмешательства.

Полезная диагностика:

```
kubectl -n cpaas-system get cvsh global \
  -o jsonpath='{range .status.conditions[*]}{.type}{"\t"}{.status}
{"\t"}{.reason}{"\t"}{.message}{"\n"}{end}'

kubectl -n cpaas-system get cvsh global \
  -o jsonpath='{.status.preflight.observedAt}{"\n"}{range .status.pre
flight.checks[*]}{.name}{"\t"}{.policy}{"\t"}{.state}{"\t"}{.reason}
{"\t"}{.message}{"\n"}{end}'

kubectl -n cpaas-system get cvsh global \
  -o jsonpath='{range .status.history[*]}{.version}{"\t"}{.state}
{"\t"}{.startedTime}{"\t"}{.completionTime}{"\n"}{end}'
```

6 (Условно) Обновите Alauda Service Mesh Essentials

Если установлен **Service Mesh v1**, перед обновлением кластеров рабочей нагрузки обратитесь к документации Alauda Service Mesh Essentials Cluster Plugin ↗ .

После обновления

- Обновить Alauda AI ↗
- Обновить Alauda DevOps ↗

- После того как все кластеры рабочей нагрузки также достигнут ACP 4.3, завершите усиление PKCE, выполнив инструкцию Отключение PKCE Plain Method.
- ACP 4.3 устраняет проблему аутентификации API, из-за которой к некоторым API ранее можно было обращаться без аутентификации. После того как глобальный кластер достигнет ACP 4.3, обновите следующие L5-плагины до версий, совместимых с ACP v4.3. В противном случае их UI-страницы могут не открываться:
  - `Alauda DevOps v3`
  - `Alauda AI Essentials`
  - `Alauda Hyperflux`
  - `Alauda Container Platform Data Services Essentials`
- После обновления плагинов убедитесь, что UI-страница каждого из перечисленных плагинов открывается успешно.

Процедура Global DR

Используйте эту процедуру, если в среде есть основной global-кластер и резервный global-кластер. Специфичные для DR шаги ниже выполняются дополнительно к стандартному рабочему процессу CVO.

1 Проверьте среду DR перед обновлением

Выполните обычные процедуры проверки global DR, чтобы убедиться, что данные в **резервном global-кластере** согласованы с **основным global-кластером**. Дополнительную информацию о топологии DR и рабочем процессе синхронизации см. в Disaster Recovery для глобального кластера.

Если обнаружены несоответствия, обратитесь в техническую поддержку, прежде чем продолжать.

На **обоих** global-кластерах выполните следующую команду, чтобы убедиться, что никакие узлы `Machine` не находятся в состоянии не-выполнения:

```
kubectl get machines.platform.tkestack.io
```

Если такие узлы есть, устраните проблему перед продолжением.

2 Удалите плагин синхронизации etcd с резервного global-кластера

1. Откройте Web Console **резервного global-кластера** по его IP или VIP.
2. Переключитесь в представление **Administrator**.
3. Перейдите в **Marketplace > Cluster Plugins** и выберите кластер `global`.
4. Найдите **Alauda Container Platform etcd Synchronizer** и удалите его.
5. Дождитесь завершения удаления, прежде чем продолжать.

3 Подготовьте артефакты обновления на обоих global-кластерах

Выполните пункт **Подготовьте артефакты обновления** из стандартного рабочего процесса на **резервном global-кластере** и **основном global-кластере**.

Используйте одинаковый режим подготовки на обоих кластерах.

4 Обновите резервный global-кластер

Если вы будете использовать Web Console на резервном global-кластере, убедитесь, что `ProductBase` резервного кластера включает резервный VIP в `spec.alternativeURLs`:

```yaml
apiVersion: product.alauda.io/v1alpha2
kind: ProductBase
metadata:
  name: base
spec:
  alternativeURLs:
    - https://<standby-cluster-vip>
```

После завершения подготовки выполните оставшиеся шаги стандартного рабочего процесса на **резервном global-кластере**:

1. **Выполните предварительные проверки**
2. **Запросите обновление**
3. **Наблюдайте за выполнением** до тех пор, пока резервный global-кластер не достигнет целевой версии

5 Обновите основной global-кластер

После того как резервный global-кластер достигнет целевой версии, выполните оставшиеся шаги стандартного рабочего процесса на **основном global-кластере**:

1. **Выполните предварительные проверки**
2. **Запросите обновление**
3. **Наблюдайте за выполнением** до тех пор, пока основной global-кластер не достигнет целевой версии

6 Переустановите плагин синхронизации etcd и проверьте состояние синхронизации

Перед переустановкой плагина убедитесь, что порт `2379` корректно перенаправляется с обоих VIP global-кластера на их узлы control plane, если используется такой режим перенаправления. Перенаправление порта через load balancer не требуется, если резервный global-кластер может напрямую обращаться к активному global-кластеру.

Чтобы переустановить плагин:

1. Откройте Web Console **резервного global-кластера** через его VIP и переключитесь в представление **Administrator**.
2. Перейдите в **Marketplace > Cluster Plugins** и выберите кластер `global`.
3. Найдите **Alauda Container Platform etcd Synchronizer**, нажмите **Install** и настройте необходимые параметры.

При настройке плагина:

- Если порт `2379` не перенаправляется через load balancer, правильно задайте **Active Global Cluster ETCD Endpoints**.

- Используйте значение по умолчанию для **Data Check Interval**.
- Оставьте **Print detail logs** отключенным, если только вы не выполняете устранение неполадок.

Убедитесь, что sync Pod запущен на резервном global-кластере:

```
kubectl get po -n cpaas-system -l app=etcd-sync
etcd_sync_pod=$(kubectl get po -n cpaas-system -l app=etcd-sync -o js
onpath='{.items[0].metadata.name}')
kubectl logs -n cpaas-system "$etcd_sync_pod" | grep -i "Start Sync u
pdate"
```

После появления `Start Sync update` пересоздайте один из Pod, чтобы запустить синхронизацию ресурсов с зависимостями ownerReference:

```
etcd_sync_pod=$(kubectl get po -n cpaas-system -l app=etcd-sync -o js
onpath='{.items[0].metadata.name}')
kubectl delete po -n cpaas-system "$etcd_sync_pod"
```

Проверьте состояние синхронизации:

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

Интерпретация вывода:

- `LOCAL ETCD missed keys`: ключи существуют в основном global-кластере, но отсутствуют в резервном. Это часто решается после перезапуска одного Pod `etcd-sync`.

> - `LOCAL ETCD surplus keys`: ключи существуют в резервном global-кластере, но отсутствуют в основном. Перед удалением обсудите их с вашей операционной командой.

Связанная документация

- Обзор
- Перед обновлением
- Обновить кластеры рабочей нагрузки
- Обновление кластеров
- Disaster Recovery для глобального кластера

Upgrade Workload Clusters

Кластеры workload могут быть обновлены после того, как глобальный tier уже достиг целевой версии распределения ACP. Глобальный tier не нужно обновлять повторно, если он уже находится на этой версии распределения.

ACP 4.3 использует тот же рабочий процесс на основе CVO для workload clusters, что и для кластера `global`: подтвердите prerequisites, выполните preflight checks, отправьте запрос на обновление и отслеживайте выполнение. В этом рабочем процессе есть два дополнительных правила:

- Если платформа использует **global DR**, и standby, и primary global clusters должны достичь целевой версии Distribution Version, прежде чем любой workload cluster будет обновлен до этой версии Distribution Version.
- Целевая версия обычно становится доступна для workload cluster только после того, как global tier уже достиг той же версии Distribution Version.

Содержание

Workflow

Confirm workload-cluster prerequisites

Run preflight checks

Request the upgrade

Observe progress

After All Workload Clusters Reach ACP 4.3

Related Documentation

Workflow

1 Confirm workload-cluster prerequisites

Перед отправкой запроса на обновление убедитесь, что:

- Версия workload cluster ниже текущей версии кластера `global`.
- Global tier уже достиг целевой версии ACP Distribution Version.
- В средах global DR и standby, и primary global clusters уже достигли этой целевой версии Distribution Version.

2 Run preflight checks

Выполните preflight перед отправкой запроса на обновление:

```
bash upgrade.sh --cluster=<cluster> --preflight
```

Preflight возвращает две части:

Output	Purpose
`Summary`	Shows the overall result, current version, desired version, and desired image.
`Checks`	Shows the result of each individual validation item.

Если preflight не проходит, не отправляйте запрос на обновление, пока все блокирующие элементы не будут устранены.

Паттерны обработки блокировок preflight см. в Handle preflight blocks when needed.

3 Request the upgrade

После успешного прохождения preflight выберите один из следующих способов:

- Запустите обновление из workload cluster в Web Console.
- Используйте ACP CLI, если ваша текущая сессия ACP может получить доступ к целевому workload cluster. Чтобы избежать неоднозначности, укажите целевой кластер явно с помощью `--cluster`.

Если вы впервые работаете с ACP CLI, см. Getting Started with ACP CLI. Для выбора session и cluster см. Managing CLI Profiles.

Если вы используете Web Console, запрос выполняется в два этапа:

- На **Step 1** проверьте список RPCH.
- Нажмите **Acknowledge**, чтобы перейти к **Step 2**.
- На **Step 2** проверьте **Current Version** и **Target Version**. На этом этапе страница не отображает список plugin или warning panel.
- Целевая версия — это текущая версия кластера `global`, и ее нельзя выбрать вручную в Web Console.
- Нажмите **Start Upgrade**.
- После отправки запроса на странице отображается, что запрос на обновление был отправлен, а действие переходит в состояние in-progress.

Пример ACP CLI:

```
# Request upgrade to the highest version currently published in avail
ableUpdates
ac adm upgrade --cluster=<cluster> --to-latest

# Request upgrade to a specific target version
ac adm upgrade --cluster=<cluster> --to=4.3.0
```

ACP CLI отправляет запрошенную целевую версию для указанного workload cluster. Возможность продолжения обновления по-прежнему определяется доступными целями обновления кластера и проверками preflight в upgrade controller. ACP CLI не используется для обновления кластера `global`.

Полный рабочий процесс обновления в AC CLI (подготовка metadata, available updates, режимы запроса и интерпретация статуса) см. в Upgrading Clusters. Полный синтаксис команд и флагов см. в AC CLI Administrator Command Reference.

4 Observe progress

После отправки запроса на обновление используйте `cvsh.status`, чтобы отслеживать текущую версию, целевую версию, результаты preflight, stages и history:

```
kubectl get cvsh <cluster> -n cpaas-system
kubectl get cvsh <cluster> -n cpaas-system -o yaml
```

Если текущий контекст ACP указывает на целевой workload cluster, вы также можете проверить статус, отображаемый CLI:

```
# Show summary, preflight, and stage progress for the target cluster upgrade
ac adm upgrade status
```

Подробную семантику вывода `ac adm upgrade status` (интерпретацию preflight и stages) см. в Upgrading Clusters.

При устранении неполадок сначала проверьте conditions, сведения preflight и history:

```
kubectl -n cpaas-system get cvsh <cluster> \
  -o jsonpath='{range .status.conditions[*]}{.type}{"\t"}{.status}{"\t"}{.reason}{"\t"}{.message}{"\n"}{end}'

kubectl -n cpaas-system get cvsh <cluster> \
  -o jsonpath='{.status.preflight.observedAt}{"\n"}{range .status.preflight.checks[*]}{.name}{"\t"}{.policy}{"\t"}{.state}{"\t"}{.reason}{"\t"}{.message}{"\n"}{end}'

kubectl -n cpaas-system get cvsh <cluster> \
  -o jsonpath='{range .status.history[*]}{.version}{"\t"}{.state}{"\t"}{.startedTime}{"\t"}{.completionTime}{"\n"}{end}'
```

After All Workload Clusters Reach ACP 4.3

После того как все workload clusters достигнут ACP 4.3, завершите усиление безопасности PKCE, следуя инструкции Disabling the PKCE Plain Method.

Related Documentation

- Overview
- Upgrade the global cluster
- Upgrading Clusters