

Мессенджер и ВКС

**Инструкция по обновлению кластера
Мессенджера и ВКС (на версию 25.3 и выше с
версии 24.9 и выше)**

Назначение документа	3
Описание процедуры обновления	3
Топология кластера Мессенджера	3
План обновления	4
Перед обновлением	5
Обновление до версии 25.3	5
Требования	6
Сохранение статуса обновления	6
Настройка	6
Обновление	7
Запуск обновления	7
Поэтапное обновление	7
Автоматическое обновление	8
Описание шагов процесса обновления	8
Подготовительные шаги перед обновлением (pre_upgrade)	8
Обновление стороны B (side_b)	10
Обновление стороны A кроме первой ноды (side_a_except_first_node)	12
Обновление стороны A (первая нода)	13
Завершающие шаги после обновления (post_upgrade)	14
Решение типовых проблем при обновлении	14
Ошибки при переключении нагрузки	15
Не собирается кластер сервисного etcd (внутри K8s)	15
Не применяется Helmwave	16
Ошибки состояния оркестратора/MySQL	17
Ошибки mirantool migration	17
Дополнительная информация по работе обновления	17
Если настроена интеграция с Почтой VK WorkSpace	18

Назначение документа

В инструкции описано обновление кластера Мессенджер и ВКС на версию 25.3 и выше с версии 24.9 и выше.

Если последнее обновление Мессенджер и ВКС проводилось более трех версий назад, [свяжитесь с технической поддержкой](#) для помощи с обновлением.

Внимание

Начиная с версии 25.2 и выше сервис Keycloak переместился с поддомена "di." на "kc.". Для его корректной работы необходимо:

1. Настроить DNS-зону [по инструкции](#).
2. Если в инсталляции включена SSO-аутентификация, удалите запись из таблицы `stdb idp_configuration` и заведите новую [по инструкции](#).

Описание процедуры обновления

Обновление Мессенджера состоит из последовательного обновления групп узлов кластера. Каждая виртуальная машина кластера содержит в себе два диска — `root` и `data`. `Root`-диск содержит в себе ОС, ПО Мессенджера и некоторый набор прикладного ПО. `Data`-диск содержит в себе данные сервисов Мессенджера и лог-файлы.

Обновление каждого узла совершается через замену `root`-диска на новую версию с последующей настройкой. При замене `root`-диски IP-адреса виртуальных машин должны сохраняться. `Data`-диск также сохраняется.

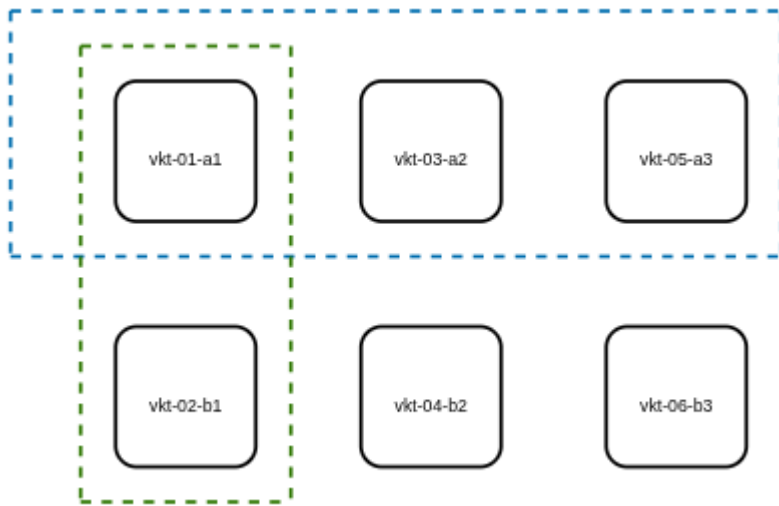
Во время обновления инструмент обновления запросит копирование архива `update.<ver>.tar.gz` на каждый сервер. Этот файл поставляется вместе с дистрибутивом Мессенджера.

После замены диска нужно добавить SSH-ключи для доступа с виртуальной машины, где запускается обновление. Если ключи сохраняются в файле `known_hosts`, нужно будет оттуда удалить старые (или заменить).

Поддерживаемые версии: 24.9 - 25.4

Топология кластера Мессенджера

Каждый узел Мессенджера принадлежит паре (или шарду) и стороне.



Например, в данной схеме узлы vkt-01-a1, vkt-03-a2 и vkt-05-a3 принадлежат стороне **a**.

Узлы vkt-02-b1, vkt-04-b2 и vkt-06-b3 принадлежат стороне **b**.

Узлы vkt-01-a1 и vkt-02-b1 принадлежат паре **1**.

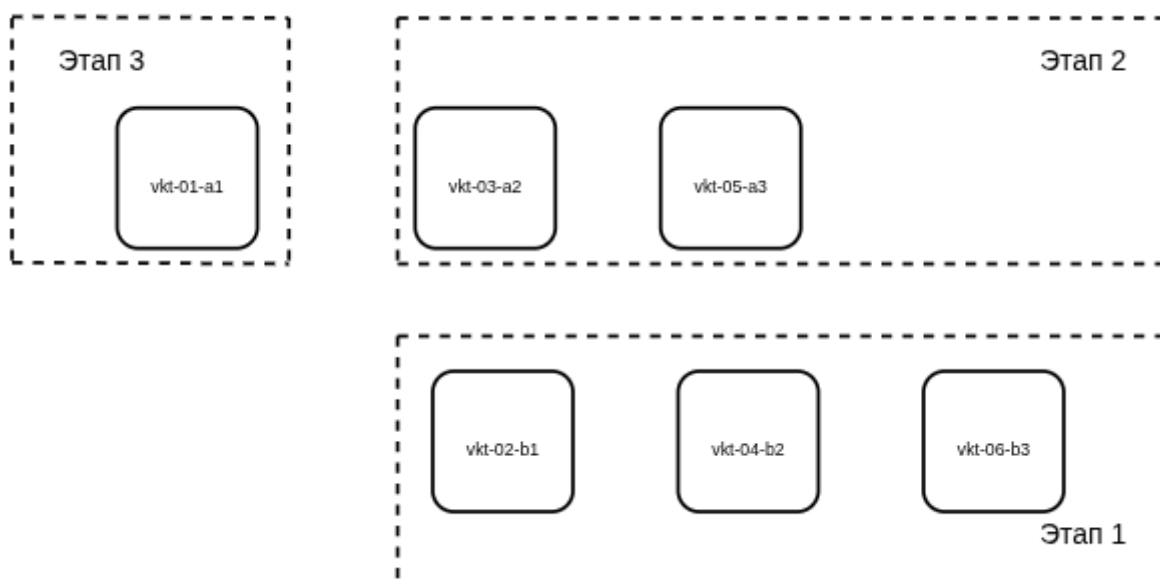
Узлы vkt-03-a2 и vkt-04-b2 принадлежат паре **2**, и т.д.

Принадлежность каждого узла паре и стороне задается при установке, и эту информацию можно посмотреть в конфигурационном файле `/etc/ctfact3.yaml`

План обновления

Обновление происходит в 3 этапа.

План обновления шардированной схемы:



Такая этапность обусловлена наличием сервисов, работающих по протоколу Raft (etcd, Consul), для полноценной работы которых требуется работа более двух из трех инстансов в каждый момент времени.

Перед обновлением

Обновление кластера процедура достаточно сложная, затрагивает все компоненты кластера. Перед обновлением всегда рекомендуется убедиться в наличии рабочих резервных копий всех виртуальных машин и/или данных.

Процедуру обновления всегда рекомендуется сначала проводить на тестовом кластере.

Также нужно иметь список всех кастомных изменений, которые нельзя указать через параметры установщика Мессенджера. Например:

- Модификации конфигурационных файлов сервисов.
- Модификации Helm-чартов или конфигурации Helmwave.
- Модификации системных скриптов (например, времени хранения логов или удаленных серверов syslog).

Так при обновлении Мессенджера структура и расположение конфигурационных файлов может меняться. Это нужно учитывать в инструментах для донесения кастомных правок. В процессе обновления эти изменения нужно будет применить.

После обновления старые версии конфигурационных файлов будут доступны в директории `/mnt/data/myteam.backup/`. Например: `/mnt/data/myteam.backup/usr_local/nginx-im/html/myteam/myteam-config.json`.

Нужно убедиться в отсутствии текущих проблем в мониторинге.

Во время обновления кластера рекомендуется закрыть все вкладки веб-клиента, поскольку старые версии не сохраняют сессию при обновлении. При активной работе веб-клиента есть вероятность выхода из приложения.

Если в инсталляции Мессенджера и ВКС была вручную настроена трансляция UDP-трафика с белого IP-адреса на серый и/или использование одновременно белого и серого IP, при обновлении с версии 25.3 и выше до 25.4 и выше необходимо перенести настройки из файлов `/usr/local/etc/k8s/helmwave/projects/janus/values/janus.yml` и `/usr/local/etc/k8s/helmwave/projects/voice-relayicq/values/voice-relayicq.yml` в файл `/usr/local/etc/k8s/helmwave/store/calls.yml`. Если этого не сделать, при обновлении будут установлены значения по умолчанию — `false`.

Для применения изменений выполните команду на raft-ноде (1b):

```
HELMWAVE_ENV_NAME=cluster HELMWAVE_USE_LOCAL_REPO_CACHE=true hwup
```

Обновление до версии 25.3

В релизе 25.3 часть сервисов со встроенными хранилищами данных переезжает в K8s. Нужно убедиться в наличии достаточного свободного дискового пространства для миграции.

Объем дополнительного дискового пространства, которое потребуется, можно посмотреть командой:

```
du -hxc /data/{botex,cmini,gbld,cox,fagus}* /data/tarantool/*/
{skywalker,whoost,giraffe,papker}* |fgrep total
```

Требования

- Окружение с Ansible 9.* (версия > 9 не работает со старым Python).
- Установленная коллекция community.general.
- Отсутствие ограничений на количество исходящих соединений SSH. С выбранного окружения должна быть возможность установления количества соединений SSH, равного количеству виртуальных машин в кластере.

Можно установить требуемые компоненты через requirements.txt:

```
pip3 install -r requirements.txt
```

Сохранение статуса обновления

Рекомендуется использовать для каждого обновления отдельную директорию с Ansible.

Инструмент обновления сохраняет факт выполнения шагов обновления в файле `playbooks/roles/upgrade_cluster/vars/state.yaml`. При начале обновления этот файл должен отсутствовать.

В случае остановки обновления из-за возникновения ошибки повторный запуск не будет выполнять уже выполненные шаги. Соответственно, при возникновении ошибки можно перезапустить обновление (после исправления ошибки, если это требуется).

Внимание

По этой причине нельзя использовать одно дерево каталогов инструмента обновления для обновления двух разных инсталляций без очистки файла `state.yaml`.

Настройка

В файле `playbooks/vars/settings.yaml` необходимо указать переменные:

Переменная	Описание
<code>vkt_current_version</code>	"24.9" — текущая версия кластера
<code>vkt_after_version</code>	"25.3" — версия, на которую обновляемся

Переменная	Описание
<code>run_serial</code>	"4" — параллельное выполнение задач (количество узлов). Для 4 виртуальных машин указывается 4, для 8 виртуальных машин — 8 и т.д.)

Также рекомендуется отключить строгую проверку ключей в SSH клиенте на время обновления (параметр `StrictHostKeyChecking` в OpenSSH).

Обновление

Запуск обновления

Для запуска необходимо указать IP-адрес любого хоста из кластера.

Есть 2 способа запуска обновления: поэтапный и автоматический.

Поэтапное обновление

Поэтапный запуск будет выполнять следующие этапы отдельно:

- Подготовка перед обновлением.
- Обновление стороны b.
- Обновление стороны a кроме первой ноды.
- Обновление первой ноды.
- Пост-обновление.

Команды запуска:

```
ansible-playbook -i <any_cluster_ip>, playbooks/upgrade_cluster.yaml --tags pre_upgrade
ansible-playbook -i <any_cluster_ip>, playbooks/upgrade_cluster.yaml --tags side_b
ansible-playbook -i <any_cluster_ip>, playbooks/upgrade_cluster.yaml --tags
side_a_except_first_node
ansible-playbook -i <any_cluster_ip>, playbooks/upgrade_cluster.yaml --tags side_a_first_node
ansible-playbook -i <any_cluster_ip>, playbooks/upgrade_cluster.yaml --tags post_upgrade
```

Где `<any_cluster_ip>` — IP-адрес любой ноды кластера.

Внимание

Обратите внимание на запятую после IP-адреса

Автоматическое обновление

Не рекомендуется для обновления production-среды.

Автоматический запуск будет последовательно выполнять все шаги обновления до завершения:

```
ansible-playbook -i any_cluster_ip, playbooks/upgrade_cluster.yaml
```

Где <any_cluster_ip> – IP-адрес любой ноды кластера.

Внимание

Обратите внимание на запятую после IP-адреса

Описание шагов процесса обновления

Ниже представлено подробное описание каждого этапа обновления. Данные этапы будут последовательно выполняться при применений плейбуков выше.

Подготовительные шаги перед обновлением (pre_upgrade)

Шаг	Команда	Где запускается
Проверка mon.sh на наличие критических проблем	mon.sh	Все виртуальные машины
Проверка списка установленных чартов на наличие проблем (релизов не в статусу deployed)	helm list -A -a	Пара 1, сторона а
Проверка состояния кластеров MySQL	orch_topo.sh	Пара 1, сторона а
Проверка списка подов K8s на наличие проблем	kubectl get pods -A	Пара 1, сторона а
Переключение нагрузки на сторону B	im_utils -drain-ipros im_utils -drain-orch	Пара 1, сторона b
Проверка наличия файла update__<версия на которую обновляемся>.tar.gz	ls /tmp/update_<version>.tar.gz	Сторона а

Шаг	Команда	Где запускается
Обновление версии K3s	update_k3s.sh	Сторона а, последовательно
Создание вспомогательных директорий	mkdir -p /oap/icq/logs/ go.files.icq.com/siem chown gofiles:gofiles /oap/icq/ logs/go.files.icq.com/siem	Сторона а
Переключение нагрузки на сторону А	im_utils -drain-ipros im_utils -drain-orch	Пара 1, сторона b
Проверка наличия файла update_<версия на которую обновляемся>.tar.gz	ls /tmp/update_<version>.tar.gz	Сторона b
Обновление версии K3s	update_k3s.sh	Сторона b, последовательно
Создание вспомогательных директорий	mkdir -p /oap/icq/logs/ go.files.icq.com/siem chown gofiles:gofiles /oap/icq/ logs/go.files.icq.com/siem	Сторона b
Обновление пакетов gic, tnt и im_deployer (при необходимости) Пакеты в директории playbooks/ roles/mirantool_migration/files/rpms/	rpm -Uvh *.rpm	Пара 1, сторона b
Сохранение IPROS-карт для сервисов типа imantool, мигрируемых в K8s. 25.3: Skywalker, Whoost, Giraffe, Parker	/usr/local/bin/gic utils export- map --map=<service> --export- dir=/mnt/data/ mirantool_backup/ctrl	
Остановка сервисов аварийного переключения контроллера IPROS и VShard	systemctl stop {ctrl,polls} {sentinel,failover} systemctl disable {ctrl,polls}	Все виртуальные машины
Установка аннотации im/white-ip на все ноды кластера	kubectl annotate node <node> im/white-ip=<external_ip>	Все виртуальные машины
Применение патчей для K3s Скрипты в директории: playbooks/	bash patch_registry_selector.sh	Пара 1, сторона b

Шаг	Команда	Где запускается
roles/upgrade_cluster/files/ pre_upgrade_kube_patches/		

Обновление стороны B (side_b)

Шаг	Команда	Где запускается
Проверка /etc/ctfact3.yaml на наличие нужных групп (apigwv2, ingress)	-	Сторона b
Переключения мастера контроллера IPROS на сторону A	/usr/local/bin/im_utils --switch-ctrlr	Пара 2
Переключение нагрузки на сторону A	im_utils --drain-ipros im_utils --drain-orch im_utils --drain-k8s	Пара 1, сторона b
Запуска pre-upgrade.sh	pre-upgrade.sh	Сторона b
Остановка виртуальных машин на стороне b	init 0	Сторона b
Замена root-дисков на новую версию на стороне b	-	Сторона b
Применение патчей Скрипты в директории: playbooks/ roles/upgrade_cluster/files/ pre_upgrade_kube_patches/ bash patch_registry_selector.sh	bash update_registry_selector.sh	Сторона b
Подготовка root новых дисков к запуску сервисов	/usr/local/bin/im_deployer --update --prepare-root	Сторона b
Запуск первого этапа настройки новых дисков	/usr/local/bin/im_deployer --update --stage1	Сторона b
		Сторона b

Шаг	Команда	Где запускается
Запуск второго этапа настройки новых дисков	<code>/usr/local/bin/im_deployer -update -stage2</code>	
Получение токенов для ботов	<code>/usr/local/bin/im_deployer -bots</code>	Сторона b
Включение в нагрузку стороны b в K8s	<code>kubectl uncordon <host></code>	Пара 1, сторона b
Обновление Helm-чартов	<code>/usr/local/bin/im_deployer -update -helmwave</code> <code>/usr/local/bin/im_deployer -helmwave -hw-project bootstrap-cluster -hw-once -update</code>	Пара 1, сторона b
Перезагрузка Nginx	<code>nginx.sh reload</code>	Все виртуальные машины
Проверка статуса репликации сервисов Kust, смигрированных в K8s. 25.3: Botex, Cmini, Gbld-mchat, Gbld-st, Cox, Fagus	<code>/usr/local/bin/gic utils check-map -map <service></code>	Пара 1, сторона b
Остановка сервисов mirantool для миграции в K8s. 25.3: Skywalker, Woost, Piraffe, papker	<code>systemctl stop tarantool@<service></code>	Сторона a
Запуск миграции сервисов	<code>/usr/local/bin/micli migrate-map -backup-dir-path=/mnt/data/mirantool_backup/ctlr <service></code>	Пара 1, сторона b
Проверка статуса репликации мигрируемых сервисов	<code>/usr/local/bin/tnt -ipros-map=<service> -lookup-address-in-weave-enabled=false check -wait=true replication</code>	Пара 1, сторона b
Проверка статуса смигрированных сервисов	<code>/usr/local/bin/gic utils check-map -map=<service></code>	Пара 1, сторона b

Обновление стороны А кроме первой ноды (side_a_except_first_node)

Шаг	Команда	Где запускается
Проверка /etc/ctfact3.yaml на наличие нужных групп (apigwv2, ingress)	-	Сторона а (кроме 1)
Переключения мастера контроллера IPROS на сторону В	/usr/local/bin/im_utils -switch-ctrl <new_master>	Пара 2
Переключение нагрузки на сторону В	im_utils -drain-ipros im_utils -drain-orch im_utils -drain-k8s	Пара 1, сторона b
Запуска pre-upgrade.sh	pre-upgrade.sh	Сторона а (кроме 1)
Остановка виртуальных машин на стороне b	init 0	Сторона а (кроме 1)
Замена root-дисков на новую версию на стороне b	-	Сторона а (кроме 1)
Применение патчей. Скрипты в директории: playbooks/ roles/ upgrade_cluster/files/ pre_upgrade_kube_patches/ bash patch_registry_selector.sh	bash update_registry_selector.sh	Сторона а (кроме 1)
Подготовка root новых дисков к запуску сервисов	/usr/local/bin/im_deployer -update -prepare-root	Сторона а (кроме 1)
Запуск первого этапа настройки новых дисков	/usr/local/bin/im_deployer -update -stage1	Сторона а (кроме 1)
Запуск второго этапа настройки новых дисков	/usr/local/bin/im_deployer -update -stage2	Сторона а (кроме 1)
Получение токенов для ботов	/usr/local/bin/im_deployer -bots	Сторона а (кроме 1)

Шаг	Команда	Где запускается
Включение в нагрузку стороны b в K8s	kubectl uncordon <host>	Сторона а (кроме 1)
Перезагрузка Nginx	nginx.sh reload	Все VM

Обновление стороны А (первая нода)

Шаг	Команда	Где запускается
Проверка /etc/ctfact3.yaml на наличие нужных групп (apigwv2, ingress)	-	Папа 1, сторона а
Переключение нагрузки на сторону В	im_utils --drain-ipro im_utils --drain-k8s	Пара 1, сторона а
Запуска pre-upgrade.sh	pre-upgrade.sh	Пара 1, сторона а
Остановка виртуальных машин 1 на стороне а	init 0	Пара 1, сторона а
Замена root-дисков на новую версию на виртуальных машин 1 на стороне а	-	Пара 1, сторона а
Применение патчей. Скрипты в директории: playbooks/ roles/upgrade_cluster/files/ disk_patches/	bash update_registry_selector.sh	Пара 1, сторона а
Подготовка root новых дисков к запуску сервисов	/usr/local/bin/im_deployer --update --prepare-root	Пара 1, сторона а
Запуск первого этапа настройки новых дисков	/usr/local/bin/im_deployer --update --stage1	Пара 1, сторона а

Шаг	Команда	Где запускается
Запуск второго этапа настройки новых дисков	<code>/usr/local/bin/im_deployer -update -stage2</code>	Пара 1, сторона а
Получение токенов для ботов	<code>/usr/local/bin/im_deployer -bots</code>	Пара 1, сторона а
Включение в нагрузку виртуальных машин 1 на стороне а	<code>kubectl uncordon <host></code>	Пара 1, сторона а
Перезагрузка Nginx	<code>nginx.sh reload</code>	Все ВМ

Завершающие шаги после обновления (post_upgrade)

Шаг	Команда	Где запускается
Проверка mon.sh на наличие критических проблем	<code>mon.sh</code>	Все виртуальные машины
Проверка списка установленных чартов на наличие проблем (релизов не в статусу deployed)	<code>helm list -A -a</code>	Пара 1, сторона а
Проверка состояния кластеров MySQL	<code>orch_topo.sh</code>	Пара 1, сторона а
Проверка списка подов K8s на наличие проблем	<code>kubectl get pods -A</code>	Пара 1, сторона а

Решение типовых проблем при обновлении

В общем случае, если какой-то шаг завершается с ошибкой, можно попробовать его перезапустить. Если после перезапуска ошибка сохраняется, нужно более детально разбираться с ошибкой.

Ошибки при переключении нагрузки

В случае возникновения ошибок при выполнении `/usr/local/bin/im_utils --drain-ipros`:

1. Проверить мониторинг на наличие ошибок репликации. В случае нахождения подобных ошибок исправить их в соответствии с инструкцией по эксплуатации.
2. Если ошибок репликации нет, то нужно выполнить перезапуск службы, для которой не происходит переключения. Например, если при переключении нагрузки на первой паре возникает ошибка переключения службы Border, нужно перезапустить службу border-1 на виртуальной машине в первой паре:

```
systemctl restart border-1
```

В случае возникновения ошибок при выполнении `/usr/local/bin/im_utils --drain-orch`:

1. Повторить команду.
2. В случае повторения ошибки проверить вывод `/usr/local/bin/orch_topo.sh` на наличие проблем. В случае нахождения проблем исправить их в соответствии с инструкцией по эксплуатации.

Не собирается кластер сервисного etcd (внутри K8s)

Иногда в процессе обновления кластер etcd нормально не собирается — один или более узлов etcd рестартится в цикле.

Ниже шаги по исправлению.

1. Перед работами с etcd выполнить дамп текущего содержимого:

```
etcdctl --endpoints etcd.im-etcd.svc.cluster.local:2379 get --prefix / > etcd.dump
```

Посмотреть `etcd.dump`, должен быть не пустой.

2. Перезапустить все поды etcd:

```
kubectl delete pod -n im-etcd -l app.kubernetes.io/name=etcd
```

Подождать 5 минут, проверить состояние кластера.

3. Если не помогло, то нужно инициировать заново проблемные узлы кластера etcd.

Например если проблема с 1 и 3:

```
kubectl delete pvc -n im-etcd data-etcd-1 & kubectl delete pod -n im-etcd etcd-1  
kubectl delete pvc -n im-etcd data-etcd-3 & kubectl delete pod -n im-etcd etcd-3
```

4. Если не помогло, удалить проблемный узел из кластера и выполнить `rebootstrap`. Например, если не поднимается инстанс etcd-0:

```
kubectl exec -n im-etcd etcd-1 -c etcd -- etcdctl member list # получаем id проблемного инстанса
kubectl exec -n im-etcd etcd-1 -c etcd -- etcdctl member remove <id> # из предыдущей команды
kubectl delete pvc data-etcd-0 -n im-etcd & delete pod etcd-0 -n im-etcd
```

5. Если не помогло, полностью переустановить etcd.

Выполнять на виртуальных машинах пары 1, стороны b. Удалить etcd:

```
helm uninstall -n im-etcd etcd
kubectl delete pvc -n im-etcd data-etcd-0
kubectl delete pvc -n im-etcd data-etcd-1
kubectl delete pvc -n im-etcd data-etcd-2
```

Переустановить etcd:

```
im_deployer --helmwave --update --hw-project etcd --hw-once
```

Подождать, пока все поды поднимутся и перезапустить сервис myteam-admin:

```
kubectl delete pod -n vkteams -l app=myteam-admin
```

6. Вернуть кастомные ключи из etcd.dump, если они там есть. Например настройку DLP:

```
etcdctl --endpoints=etcd.im-etcd.svc.cluster.local:2379 put /vars/services/plumber/development/public/service/DLP/address infowatch.beicq.net:9101
```

Не применяется Helmwave

Основные причины:

1. Ошибки ctfact3.yaml. Подобные ошибки могут возникать при обновлении с очень старых версий. Для устранения проблемы нужно исправить конфигурационный файл /etc/ctfact3.yaml и повторить данный шаг.
2. Выполнение Helmwave завершается по timeout. Если при применении Helmwave были проблемы с сетью или производительностью etcd, то возможны таймауты в etcd, что может привести к ошибкам.

Для исправления:

1. Посмотреть вывод команды:

```
helm list -A -a |fgrep -i pending
```

При наличии в выводе релизов Helm в статусе pending, надо сделать helm rollback на эти релизы.

Например:

```
helm rollback -n vkteams minio
```

2. Запустить повторно Helmwave.

3. При повторении проблемы можно уменьшить параллелизм команды Helmwave — выполнить вручную на второй виртуальной машине (пара 1, сторона b) команды:

```
export HELM_CACHE_HOME=/root/.cache/helm
export HELMWAVE_LOG_FORMAT=text
export HELMWAVE_LOG_COLOR=false
export HELMWAVE_GRAPH_WIDTH=1
export HELMWAVE_AUTO_YML=true
export HELMWAVE_TEMPLATER=gomplate
export HELMWAVE_AUTO_BUILD=true
export HELMWAVE_USE_LOCAL_REPO_CACHE=1
export HELMWAVE_PARALLEL_LIMIT=1
export HELMWAVE_LOG_LEVEL=debug
export IM_DEPLOYER_STAGE=update
export HELMWAVE_ENV_NAME=cluster
helmwave up
```

4. Ошибка в предыдущем состоянии K8s как правило возникает в случае несовместимых ручных изменений в объектах K8s, либо некорректного обновления с предыдущих версий. Механизм исправления зависит от характера поломки, скорее всего нужно привлекать техническую поддержку.

Ошибки состояния оркестратора/MySQL

При возникновении ошибок, связанных с целостностью MySQL, нужно восстановить репликацию согласно [инструкции](#).

Ошибки mirantool migration

В случае возникновения ошибок при выполнении роли mirantool migration [обратитесь в поддержку](#).

Дополнительная информация по работе обновления

Поддержка тегов: Возможность запуска плейбуков по тегам (название тега совпадает с названием `state`).

Динамический inventory: Плейбук генерирует `inventory` на основе файла `/etc/ctfact3.yaml`: `side_a`, `side_b`, `side_a_db`, `side_b_db`, `side_a_raft`, `side_b_raft`

Если настроена интеграция с Почтой VK WorkSpace

Если у вас настроена интеграция с Почтой VK WorkSpace настройте отображение сервисов Почты, Диска и Календаря в клиентском приложении.

Начиная с версии Мессенджер и ВКС 25.4 для корректного отображения сервисов необходимо донести в конфигурационный файл `/usr/local/nginx-im/html/myteam/myteam-config.json` конфигурацию в соответствии с шагами 4 и 5 инструкции <https://biz.mail.ru/docs/on-premises/vk-teams/mail-integration/index.html#10-vk-teams>

 Технический писатель: Белова Ирина

 25 марта 2026 г.