VK WorkSpace

Мессенджер и ВКС

Инструкция по поддержке репликации данных

Портал с документацией: https://biz.mail.ru/docs/on-premises/

Оглавление

Назначение документа 3

Дополнительная документация 3

Проблемы репликации сервисов и контроллера 4

- У сервиса типа KUST реплика не получает данных из мастера 4
- У сервиса типа Tarantool нарушена репликация данных 5
- У сервиса типа MySQL проблемы с топологией кластера 6
- Проблемы с репликацией контроллера 8

Проблемы с ролями контроллера 9

Назначение документа

Для обеспечения отказоустойчивости Мессенджер и ВКС реализованы механизмы копирования всех сервисов и данных. При выходе из строя мастера (отсутствие сети, ошибки в коде) нагрузка переключается на реплику, и Мессенджер и ВКС продолжает работать «бесшовно». Для эксплуатации такой инфраструктуры обязательным условием является репликация данных между основной (main) и резервной (slave) частью бэкенда, чтобы в обеих частях были одинаковые данные в любой момент времени.

В документе описано, как отслеживать проблемы репликации данных и устранять их. Документ предназначен для использования администраторами организации.

Дополнительная документация

Инструкция по подключению к внешней системе мониторинга — в документе описана настройка интеграции с внешними системами мониторинга — VictoriaMetrics и Prometheus.

Мониторинг Мессенджер и ВКС — в документе описана параметры инсталляции Мессенджер и ВКС, которые необходимо контролировать.

Проблемы репликации сервисов и контроллера

Чтобы узнать о проблемах репликации, используйте систему мониторинга согласно документации — она будет мгновенно присылать алерты в случае обнаружения проблем. Ниже описаны примеры алертов конкретных ошибок разрыва репликации и шагов для восстановления репликации данных.

> **Внимание**
>
> Проблемы репликации могут приводить к потерям данных. Их нужно решать в кратчайшие сроки.

У сервиса типа KUST реплика не получает данных из мастера

Алерт:

```
2 snmpexec_mon_cox_1_status - - - replfailed (first: 2023-05-10/10:08:27, last:
2023-05-10/11:34:02, total: 3406)
```

Ниже рассмотрим, как устранить проблему репликации на примере сервиса Cox, инстанс 1:

1. Посмотрите карту сервиса при помощи команды:

   ```
   ic map cox
   ```

2. Найдите в карте slave для нужного инстанса:

   ```
   node   cox-1     (2147483648   buckets)
       srv   cox.vkt-4vm-879-standart-01.cox-1      10.32.0.1:2510   alive   main
       srv   cox.vkt-4vm-879-standart-02.cox-1     10.32.96.0:2510   alive   slave
   ```

3. Подключитесь к slave-ноде (в нашем примере это нода vkt-4vm-879-standart-02) и выполните команду:

   ```
   clean_kust_replica.sh cox-1
   ```

4. В выводе команды будет видно, что происходит удаление ряда файлов. Это нормально. В конце вывод должен быть таким:

   ```
   INFO: starting cox-1
   INFO: successfully started cox-1
   ```

Если вывод отличается, обратитесь в техническую поддержку.

У сервиса типа Tarantool нарушена репликация данных

Алерт:

```
2 tarantool_badger_3_repl_master1 slave_lag=0.01s;500.0;1000.0 slave_lag=0.01s;500.0;1000.0
replication status stopped(!!)

2 tarantool_badger_3_replication - - problems on upstreams:1
```

Ниже рассмотрим, как устранить проблему репликации на примере сервиса Badger, инстанс 3:

1. Посмотрите карту проблемного сервиса:

   ```
   ic map badger
   ```

2. Найдите в карте slave для нужного инстанса для ноды, на которой запускали скрипт mon.sh. В нашем примере это нода vkt-4vm-675-standart-01, сервис Badger, инстанс 3:

   ```
   node   badger-3      (268435456   buckets)
       srv   badger.vkt-4vm-675-standart-01.badger-3      10.32.0.1:2502   alive   slave
       srv   badger.vkt-4vm-675-standart-02.badger-3     10.32.48.0:2502   alive   main
   ```

3. Подключитесь к slave-ноде и выполните команду:

   ```
   clean_tnt_replica.sh badger-3
   ```

   Дождитесь завершения процесса. Вывод в конце должен быть следующего вида:

   ```
   INFO: starting tarantool instance tarantool@badger-3
   INFO: successfully started tarantool@badger-3
   INFO: waiting 10 seconds
   INFO: starting IPROS instance badger-3
   INFO: successfully started badger-3
   ```

4. Если вывод команды **НЕ** такой, то дальше для текущего сервиса ничего **НЕ** делаем. Обратитесь в техническую поддержку.

   Если получен нужный вывод, через 5 минут на той же ноде, где выполняли шаг 3, выполните команду:

   ```
   ic flip <полное имя slave-ноды из шага 2>
   ```

   В выводе команды должно быть «started» или «OK».

   Если возникла ошибка, повторите выполнение команды через 5 минут. Если ошибка снова повторилась, обратитесь в техническую поддержку.

5. Если шаг 4 выполнился успешно, через 5 минут на той же ноде, где выполняли шаг 3, выполните команду:

```
ic map badger
```

и проверьте, что slave стал main, а main стал slave.

6. Если пункт 4 выполнился успешно, подключитесь по SSH к бывшей main-ноде и выполните команду:

```
systemctl restart tarantool@badger-3 badger-3
```

У сервиса типа MySQL проблемы с топологией кластера

Алерт:

```
2 orch_topo_check - orch topology is BAD: see 'im_utils --check-orch' for details
```

Чтобы проверить, есть ли проблема с репликацией у MySQL, посмотрите детали по топологии MySQL — выполните команду **/usr/local/bin/orch_topo.sh** на любой ноде.

Штатный вывод должен быть таким:

```
files-mysql-mysql-cluster-im-db-mysql-0.mysql.files:3306 (files-mysql-mysql-cluster-im-db-
mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306 (files-mysql-mysql-cluster-im-db-
mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
keycloak-mysql-mysql-cluster-im-db-mysql-0.mysql.keycloak:3306 (keycloak-mysql-mysql-cluster-
im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ keycloak-mysql-mysql-cluster-im-db-mysql-1.mysql.keycloak:3306 (keycloak-mysql-mysql-
cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
store-mysql-mysql-cluster-im-db-mysql-0.mysql.store:3306 (store-mysql-mysql-cluster-im-db-
mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ store-mysql-mysql-cluster-im-db-mysql-1.mysql.store:3306 (store-mysql-mysql-cluster-im-db-
mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
```

Инстансы без символов "+" в начале являются мастерами, с "+" — рабочими репликами.

Если в выводе видим, что запущено запущено 2 мастера:

```
files-mysql-mysql-cluster-im-db-mysql-0.mysql.files:3306 (files-mysql-mysql-cluster-im-db-
mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306 (files-mysql-mysql-cluster-im-db-
mysql-1) [null,nonreplicating,8.0.27-18,ro,ROW,>>,GTID:errant]
keycloak-mysql-mysql-cluster-im-db-mysql-0.mysql.keycloak:3306 (keycloak-mysql-mysql-cluster-
im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ keycloak-mysql-mysql-cluster-im-db-mysql-1.mysql.keycloak:3306 (keycloak-mysql-mysql-
cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
store-mysql-mysql-cluster-im-db-mysql-0.mysql.store:3306 (store-mysql-mysql-cluster-im-db-
mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ store-mysql-mysql-cluster-im-db-mysql-1.mysql.store:3306 (store-mysql-mysql-cluster-im-db-
mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
```

или запущена реплика без мастера (статус null, nonreplicating):

```
files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306 (files-mysql-mysql-cluster-im-db-
mysql-1) [null,nonreplicating,8.0.27-18,ro,ROW,>>,GTID:errant]
```

```
keycloak-mysql-mysql-cluster-im-db-mysql-0.mysql.keycloak:3306 (keycloak-mysql-mysql-cluster-
im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ keycloak-mysql-mysql-cluster-im-db-mysql-1.mysql.keycloak:3306 (keycloak-mysql-mysql-
cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
store-mysql-mysql-cluster-im-db-mysql-0.mysql.store:3306 (store-mysql-mysql-cluster-im-db-
mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ store-mysql-mysql-cluster-im-db-mysql-1.mysql.store:3306 (store-mysql-mysql-cluster-im-db-
mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
```

То нужно обратиться в техническую поддержку.

Если в выводе видно некорректно работающую реплику (с символом "-" в начале строки):

```
files-mysql-mysql-cluster-im-db-mysql-0.mysql.files:3306 (files-mysql-mysql-cluster-im-db-
mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
- files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306 (files-mysql-mysql-cluster-im-db-
mysql-1) [null,nonreplicating,8.0.27-18,ro,ROW,>>,GTID:errant]
```

то имеются проблемы с репликацией, которые надо устранить.

Для этого:

1. Выполните на любой ноде команду:

   ```
   im_utils --reup-mysql-replica <replica_name>
   ```

   где — имя реплики, которую нужно перереплицировать с мастера, например, files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306.

   Вывод должен быть таким:

   ```
   # im_utils --reup-mysql-replica files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306
   25 Feb 25 15:29:31 854622 | WARN  | pkg/weave/report.go:134 > assuming weave is ready |
   25 Feb 25 15:29:31 854622 | INFO  | pkg/weave/weave.go:140 > weave is ready |
   25 Feb 25 15:29:31 854622 | WARN  | pkg/weave/report.go:134 > assuming weave is ready |
   25 Feb 25 15:29:31 854622 | INFO  | cmd/utils/main.go:497 > deleting pvc files:data-files-
   mysql-mysql-cluster-im-db-mysql-1 |
   25 Feb 25 15:29:31 854622 | INFO  | cmd/utils/main.go:499 > deleting pod files:files-mysql-
   mysql-cluster-im-db-mysql-1 |
   ```

2. Дождитесь запуска нового пода. Чтобы вывести список подов, на любой из master-нод выполните команду:

   ```
   kubectl -n <namespace> get po
   ```

   где namespace — имя Kubernetes namespace, в котором запущена база. Определить namespace можно из вывода выше после фразы «deleting pvc» или «deleting pod».

   Дождитесь, когда у нового пода появится READY 5/5:

   ```
   NAME                                    READY   STATUS    RESTARTS   AGE
   files-mysql-mysql-cluster-im-db-mysql-1   5/5     Running   0          9m
   ```

3. Выполните команду **/usr/local/bin/orch_topo.sh** и проверьте, что больше нет ошибки «nonreplicating». Корректный вывод выглядит так:

```
files-mysql-mysql-cluster-im-db-mysql-0.mysql.files:3306 (files-mysql-mysql-cluster-im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306 (files-mysql-mysql-cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
keycloak-mysql-mysql-cluster-im-db-mysql-0.mysql.keycloak:3306 (keycloak-mysql-mysql-cluster-im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ keycloak-mysql-mysql-cluster-im-db-mysql-1.mysql.keycloak:3306 (keycloak-mysql-mysql-cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
store-mysql-mysql-cluster-im-db-mysql-0.mysql.store:3306 (store-mysql-mysql-cluster-im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ store-mysql-mysql-cluster-im-db-mysql-1.mysql.store:3306 (store-mysql-mysql-cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
```

Чтобы проверить, есть ли на реплике MySQL транзакции, которых нет на мастере, выполните команду **/usr/local/bin/orch_topo.sh** на любой ноде. Если в выводе есть статус «errant», то такие транзакции есть. Например:

```
files-mysql-mysql-cluster-im-db-mysql-0.mysql.files:3306 (files-mysql-mysql-cluster-im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
+ files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306 (files-mysql-mysql-cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID:errant]
```

В данном примере проблемный инстанс — files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306.

Чтобы исправить эту ошибку:

1. Выполните на любой ноде команду:

   ```
   im_utils --orch-fix-errants <replica>
   ```

   где replica — имя реплики, на которой нужно исправить errant-транзакции, например, files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306.

2. Через 2 минуты выполните команду **/usr/local/bin/orch_topo.sh.** Проверьте, что в выводе больше нет ошибок. Корректный вывод выглядит так:

   ```
   files-mysql-mysql-cluster-im-db-mysql-0.mysql.files:3306 (files-mysql-mysql-cluster-im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
   + files-mysql-mysql-cluster-im-db-mysql-1.mysql.files:3306 (files-mysql-mysql-cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
   keycloak-mysql-mysql-cluster-im-db-mysql-0.mysql.keycloak:3306 (keycloak-mysql-mysql-cluster-im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
   + keycloak-mysql-mysql-cluster-im-db-mysql-1.mysql.keycloak:3306 (keycloak-mysql-mysql-cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
   store-mysql-mysql-cluster-im-db-mysql-0.mysql.store:3306 (store-mysql-mysql-cluster-im-db-mysql-0)   [0s,ok,8.0.27-18,rw,ROW,>>,GTID]
   + store-mysql-mysql-cluster-im-db-mysql-1.mysql.store:3306 (store-mysql-mysql-cluster-im-db-mysql-1) [0s,ok,8.0.27-18,ro,ROW,>>,GTID]
   ```

Проблемы с репликацией контроллера

Алерт:

```
2 ctlr_topo_check - ctlr topology is BAD:  ctlr has no active replicas, check ctlr logs on
vkt04
```

В данном случае алерт указывает на проблему с репликой на ноде vkt04 сервиса Ctlr. Убедитесь, что мастер сервиса Ctlr запущен и работает на ноде vkt03:

```
vkt03# echo ipros_cluster_show |nc $(fgrep compot_bind /usr/local/etc/ctlr-1.conf |cut -d' ' -
f2 |cut -d':' -f1) 4020
vkt03.weave.local      10.32.88.0:2410  self main
vkt04.weave.local      10.32.96.0:2410  dead
```

В выводе ожидается строка вида:

```
vkt03.weave.local      10.32.88.0:2410  self main
```

Она означает, что мастер контроллера запущен на ноде vkt03.

Чтобы восстановить репликацию на vkt04, выполните:

```
vkt04# cp -rp /data/ctlr-1 /data/ctlr-1.$(date +%Y%m%d_%H%M)
vkt04# /usr/local/bin/im_deployer --ctlr-repl --update
vkt04# systemctl disable ctlr-1
```

После выполнения команды выше проверьте топологию контроллера:

```
vkt03# /usr/lib/check_mk_agent/local/ctlr_topo_check.sh
0 ctlr_topo_check - ctlr topology is OK

vkt03# echo ipros_cluster_show |nc $(fgrep compot_bind /usr/local/etc/ctlr-1.conf |cut -d' ' -
f2 |cut -d':' -f1) 4020
vkt03.weave.local      10.32.88.0:2410 self main
vkt04.weave.local      10.32.96.0:2410 alive
```

В выводе обе ноды контроллера должны иметь статус «self main» или «alive».

Проблемы с ролями контроллера

В кластере Мессенджер и ВКС штатно работает 2 инстанса контроллера. Один из этих контроллеров имеет роль мастера, второй — реплики. Роль контроллера задается в конфигурационном файле **/usr/loca/etc/ctlr-1.conf**, например:

```
vkt03# fgrep ipros_cluster /usr/local/etc/ctlr-1.conf
ipros_clustered true
ipros_cluster_add vkt04.weave.local 10.32.96.0:2410
ipros_cluster_add vkt03.weave.local 10.32.32.0:2410
ipros_cluster_main vkt03.weave.local
```

В данном случае мастером является инстанс контроллера, запущенный на vkt03.

Эти параметры должны быть согласованы между двумя инстансами контроллера. То есть на vkt04 конфигурационный файл будет выглядеть так же:

```
vkt04# fgrep ipros_cluster /usr/local/etc/ctlr-1.conf
ipros_clustered true
ipros_cluster_add vkt04.weave.local 10.32.96.0:2410
ipros_cluster_add vkt03.weave.local 10.32.32.0:2410
ipros_cluster_main vkt03.weave.local
```

Алерт:

```
2 ctlr_topo_check - ctlr topology is BAD:  ctlr has no active replicas, check ctlr logs on
vkt03
```

может указывать на наличие двух запущенных мастеров сервиса Ctlr.

> **Внимание**
>
> Нельзя допускать ситуации, когда в кластере запущено два инстанса контроллера с ролью мастер — это может приводить к потере данных и к различным нарушениями в работе кластера.

Посмотреть текущую топологию контроллера можно командой:

```
vkt03# echo ipros_cluster_show |nc $(fgrep compot_bind /usr/local/etc/ctlr-1.conf |cut -d' ' -
f2 |cut -d':' -f1) 4020
vkt03.weave.local      10.32.88.0:2410 self main
vkt04.weave.local      10.32.96.0:2410 alive
```

Если по какой-то причине в кластере запущено 2 мастера, вывод будет следующим:

```
vkt03# echo ipros_cluster_show |nc $(fgrep compot_bind /usr/local/etc/ctlr-1.conf |cut -d' ' -
f2 |cut -d':' -f1) 4020
vkt03.weave.local      10.32.88.0:2410 self main
vkt04.weave.local      10.32.96.0:2410 dead

vkt04# echo ipros_cluster_show |nc $(fgrep compot_bind /usr/local/etc/ctlr-1.conf |cut -d' ' -
f2 |cut -d':' -f1) 4020
vkt03.weave.local      10.32.88.0:2410 dead
vkt04.weave.local      10.32.96.0:2410 self main
```

То есть контроллер на vkt03 считает себя мастером и контроллер на vkt04 тоже считает себя мастером. Таким образом, мы обнаружили проблему с ролями контроллеров вручную.

В данной ситуации нужно принять один из контроллеров за мастер, а второй переключить на реплику и перереплицировать (удалить данные реплики и подтянуть их из мастера).

Какой из контроллеров выбирать мастером — зависит от истории инцидента. Правильнее всего выбрать мастером тот контроллер, который штатно получил роль мастера, а за реплику принять контроллер, который ошибочно получил роль мастера (как правило недавно, контроллер получает ошибочную роль в последнюю очередь).

Например, мы принимаем за мастер контроллер на vkt04.

На vkt03 выполните:

```
systemctl stop ctlr-1
cp -rp /data/ctlr-1 /data/ctlr-1.$(date +%Y%m%d_%H%M)
/usr/local/bin/im_deployer --ctlr-repl --update
systemctl disable ctlr-1
```

После выполнения команды проверьте топологию контроллера:

```
vkt03# /usr/lib/check_mk_agent/local/ctlr_topo_check.sh
0 ctlr_topo_check - ctlr topology is OK

vkt03# echo ipros_cluster_show |nc $(fgrep compot_bind /usr/local/etc/ctlr-1.conf |cut -d' ' -
f2 |cut -d':' -f1) 4020
vkt03.weave.local      10.32.88.0:2410 self
vkt04.weave.local      10.32.96.0:2410 alive main
```

Если команды выполняются на vkt03, в выводе нода vkt03 должна быть в статусе «self», а vkt04 — «alive main».

Технический писатель: Белова Ирина

8 августа 2025 г.