Катастрофоустойчивость по схеме 2 ЦОД + witness

Инструкция для администраторов

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

Оглавление

Назначение документа

В документе рассматривается катастрофоустойчивая конфигурация из двух георазпределенных ЦОД и дополнительного сервера, не находящегося в этих двух ЦОД. Документ будет полезен системным инженерам и архитекторам.

Термины и определения

Witness — дополнительный сервер для размещения сервисов, использующих алгоритм Raft для выбора мастера.

Дисковая пара — под дисковой парой подразумеваются связанные разделы дисков, которые размещены на двух разных гипервизорах. Для повышения отказоустойчивости на дисковую пару записываются одни и те же данные.

Overlord — супервизор баз данных.

Statusservice — хранит метаданные о том, к какому bucketservice обращаться за данными, занимается восстановлением и балансировкой данных кластера.

Группа (в Zepto) — пара серверов с одинаковыми данными.

Bucketservice — работает с файлами, которые содержат данные: читает и пишет записи в бакет и из него. Ходит в другие bucketservice и bucketcompactor.

Bucketcompactor — осуществляет отложенное удаление ненужных данных, запись и чтение файла при миграции.

Бакет — файл, в котором хранятся данные.

Дополнительная документация

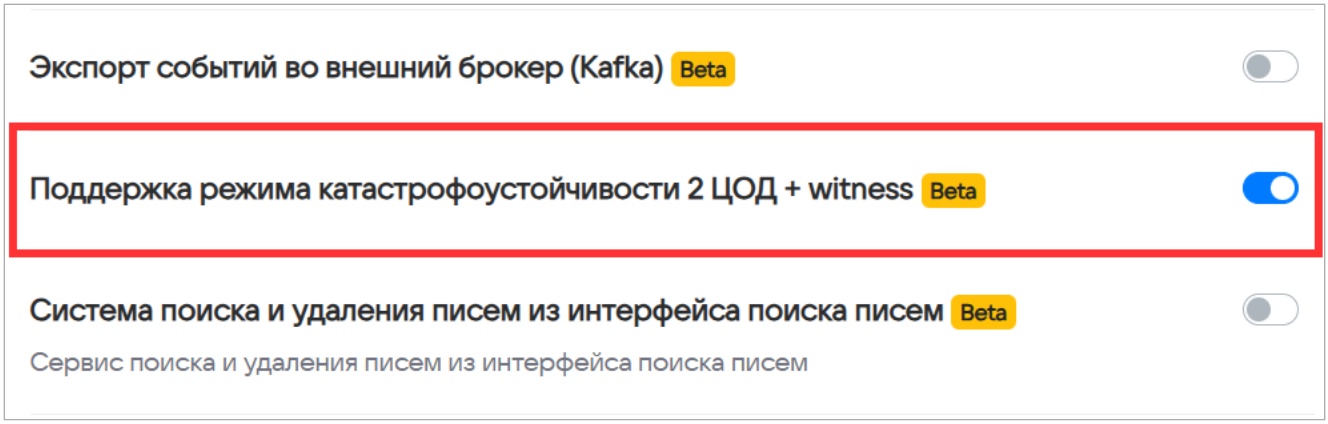
Кластерная установка VK WorkSpace — подробная инструкция по кластерной установке VK WorkSpace на 8 машин.

Катастрофоустойчивость в VK WorkSpace — описание принципов достижения катастрофоустойчивости.

Включение и базовая настройка

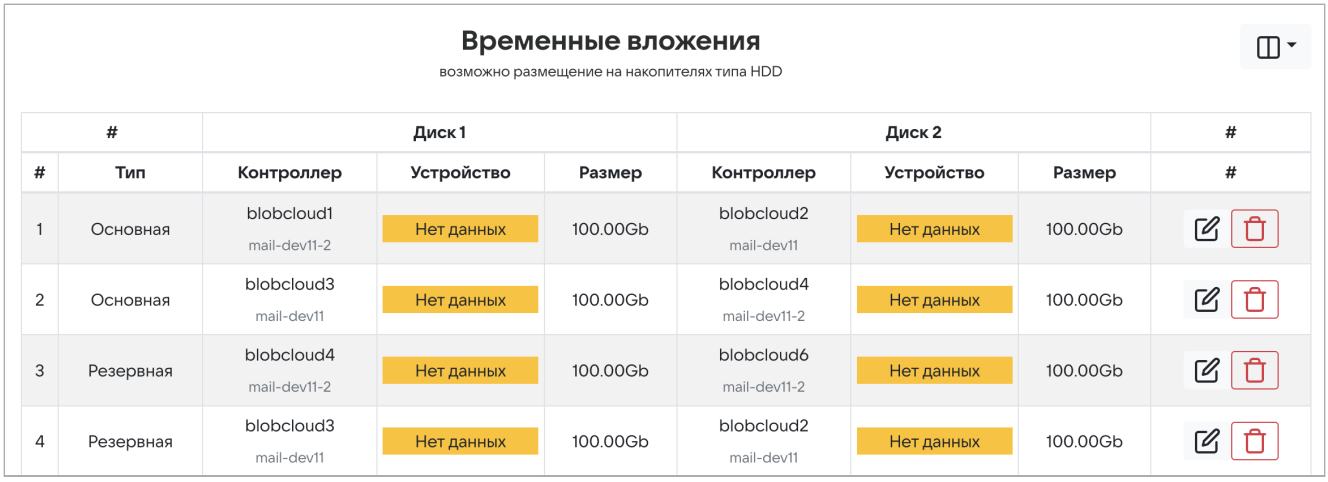
1. Перейдите в веб-интерфейс установщика VK WorkSpace по адресу `http://<company_domain>:8888`.

2. Нажмите на кнопку  в правом верхнем углу и выберите пункт **Продукты**.
3. Перейдите на вкладку **Почта**.
4. В списке продуктов включите опцию **Поддержка режима катастрофоустойчивости 2 ЦОД + witness**.



5. Нажмите на кнопку **Сохранить**.
6. Перейдите в раздел **Хранилища**.
7. Для каждого типа хранилищ добавьте минимум по две резервных пары. Каждая пара должна быть размещена в одном ЦОД.

   Пример корректного распределения дисковых пар:



Временные вложения

возможно размещение на накопителях типа HDD

#		Диск 1			Диск 2			#
#	Тип	Контроллер	Устройство	Размер	Контроллер	Устройство	Размер	#
1	Основная	blobcloud1 mail-dev11-2	Нет данных	100.00Gb	blobcloud2 mail-dev11	Нет данных	100.00Gb	
2	Основная	blobcloud3 mail-dev11	Нет данных	100.00Gb	blobcloud4 mail-dev11-2	Нет данных	100.00Gb	
3	Резервная	blobcloud4 mail-dev11-2	Нет данных	100.00Gb	blobcloud6 mail-dev11-2	Нет данных	100.00Gb	
4	Резервная	blobcloud3 mail-dev11	Нет данных	100.00Gb	blobcloud2 mail-dev11	Нет данных	100.00Gb	

> **Внимание**
>
> Основные дисковые пары разнесены по разным дата-центрам, в то время как резервные находятся на одном.

8. Запустите автоустановку и дождитесь ее окончания.
9. Создайте на гипервизорах, отведенных под фронты, по одному контейнеру **monstor**. Их должно быть не меньше двух, они должны быть разнесены по разным дата-центрам.
10. Запустите автоустановку для применения настроек.

Как работает управление данными

Катастрофоустойчивость достигается с помощью сервиса monstor. Каждые 30 секунд он опрашивает основные пары на предмет функционирования. Время можно настроить в конфигурационном файле. Если все основные пары недоступны, вся нагрузка переключается на резервные пары. При этом monstor продолжает мониторить основные пары.

Когда одна из основных пар становится целиком доступной, производится **move out** — это процедура переноса данных с резервной пары на основную. Затем резервная пара снова переводится в режим только для чтения. Так данные не теряются.

Устройство конфигурационного файла monstor

Monstor управляет работой всех типов хранилищ. Конфигурационный файл monstor по умолчанию находится по пути: `/opt/mailOnPremise/dockerVolumes/monstor1/conf/config.yaml`.

Сначала monstor подключается к **mailetcd** и пытается стать мастером. Затем он проверяет каждый тип хранилищ на наличие резервных пар. Если резервные пары присутствуют в конфигурационном файле, то monstor будет следить за этим хранилищем.

Подключение к mailetcd

Часть конфигурационного файла, отвечающая за подключение monstor к mailetcd:

```
mailetcd:
  endpoints:
    - 'mailetcd1.qdit:2379'
  timeout: 5s
  prefix: "/mailonpremise/monstor/"
  requestTimeout: 5s
  lockTimeout: 5s
  dialKeepAliveTime: 5s
  dialKeepAliveTimeout: 5s
  sessionTTLSec: 5
```

Проверка доступности дисковых пар

Как monstor должен проверять доступность хранилищ:

```
storages:
  recheckTimeout: 30s # Как часто проверять доступность
  retryConnect: 3 # Количество попыток подключения к БД Tarantool хранилищ mailcloud и
```

```
blobcloud
  retryConnectTimeout: 5s # Таймаут между попытками
```

Подключение к blobcloud и mailcloud (stz)

Каждые 30 секунд (или сколько указано в разделе `storages`) monstor пытается загрузить файл на основную пару. Если пара работает, то monstor переходит в режим ожидания до следующей проверки. Также monstor работает с остальными основными парами.

Если хоть один инстанс из пары возвращает ошибку, monstor переходит в **pairdb** и помечает пару доступной только для чтения. monstor не прекращает следить за этой парой и, если она начинает отвечать, то возвращает ей права на запись.

Когда ни одна из основных пар не отвечает, monstor обращается к pairdb и снимает у резервных пар метку «только для чтения». За основными парами monstor продолжает следить. Если хоть одна основная пара начинает отвечать, инициируется move out с резервной пары на основную.

Monstor работает с хранилищами blobcloud и mailcloud (stz) одинаково. Пример конфигурации для blobcloud:

```
  blobcloud:
    mainPairs: # Основные пары
      - pairId: 1
        members:
          - blobcloud1.qdit:8080
          - blobcloud2.qdit:8080
      - pairId: 2
        members:
          - blobcloud3.qdit:8080
          - blobcloud4.qdit:8080
    reservedPairs: # Резервные пары
      - pairId: 3
        members:
        - blobcloud4.qdit:8080
        - blobcloud6.qdit:8080
      - pairId: 4
        members:
        - blobcloud3.qdit:8080
        - blobcloud2.qdit:8080
    pairDb: # Адрес БД хранилища
      addr: 127.100.82.1:3301
```

Параметры подключения к cldst

При работе monstor с cldst выставление режимов только для чтения и move out производится через cld-jqueue и nylon. Через etcd, по префиксу определяются cld-jqueue и nylon. Через nylon выставляются и снимаются метки «только для чтения». Через cld-jqueue ставится задание на move out.

При работе с cldst monstor пытается загрузить файлы также на резервные пары. Если один ЦОД не функционирует, одна из двух резервных пар станет недоступной, поэтому monstor выставит и у резервной пары метку «только для чтения».

Пример конфигурации для cldst:

```
cloudst:
   mainPairs:
     - pairId: 1
       members:
       - s3storage2.qdit:80
       - s3storage1.qdit:80
     - pairId: 4
       members:
       - s3storage7.qdit:80
       - s3storage8.qdit:80
   reservedPairs:
     - pairId: 2
       members:
       - s3storage3.qdit:80
       - s3storage4.qdit:80
     - pairId: 3
       members:
       - s3storage5.qdit:80
       - s3storage6.qdit:80
   jQueue:  # Для активации move out при возвращении в работу основных пар
     etcdPrefix: "/cloud/jqueue"
     timeout: 5s
   nylon: # Чтобы выставлять пары в режим «только для чтения» и возвращать права на запись
     etcdPrefix: "/cloud/nylon"
     timeout: 5s
     etcd:
       endpoints:
         - 'http://mailetcd1.qdit:2379'
       timeout: 5s
```

Параметры подключения к хранилищам zepto

В файле конфигурации указаны параметры подключения для каждого из типов хранилищ zepto:

- zepto_del (stz-del) — файловый индекс удаленных писем.
- zepto_mail (stz-main) — файловый индекс основной информации о письмах.
- zepto_metad (stz-metad) — хранилище файловых деревьев пользователей.
- zepto_opt (stz_opt) — файловый индекс метаинформации о письмах.
- zepto_search (stz-search) — хранилище поисковых сниппетов.
- zepto_skel (stz-skel) — хранилище тел писем.

Из чего состоит хранилище Zepto:

- **stz-*-ss[N]** — контейнер, в котором запущен сервис, хранящий всю информацию о кластере Zepto.
- **stz-*-bm[N]** — контейнер, в котором запущены сервисы: `bucketmaster`, `bucketservice` и `bucketcompactor`.

Раз в `recheckTimeout` monstor делает запрос в `statusserver` и проверяет наличие health-статуса `daemon_maint`. Если в основной паре есть такие сообщения и нет разрыва между ЦОД, monstor попытается сделать основные пары доступными на запись.

Далее monstor проверяет основные пары на доступность. Доступность проверяется наличием хотя бы одного доступного бакета для записи в этом хранилище. Если нет ни одного доступного бакета на запись, резервная пара переключается в режим read-write. Когда основные пары становятся доступными, для резервных пар включается режим move out. Каждый `recheckTimeout` проверяется, есть ли задачи на move out для резервных пар. Когда все задачи будут выполнены, резервные пары перейдут в режим только для чтения.

Раздел в конфигурации monstor, отвечающий за хранилище `stz-search`:

```
 zepto:
    - name: stz-search
      mainPairs:
        - pairId: 3
          members:
          - stz-search-bm3.qdit
          - stz-search-bm3.qdit
      reservedPairs:
        - pairId: 1
          members:
          - stz-search-bm1.qdit
          - stz-search-bm1.qdit
        - pairId: 2
          members:
          - stz-search-bm2.qdit
          - stz-search-bm2.qdit
      statusServers: # Адреса status-серверов, на которых хранится информация о bucket-
мастерах
        -  http://127.11.22.1:11186
        -  http://127.11.22.2:11186
```

Полезные команды

Проверить состояние health-статуса:

```
zeptoctl health
```

Получить список всех групп через API:

```
zeptoctl group get
```

Проверить состояние move out:

```
zeptoctl move status
```

Список задач, который показывает статус выполненных команд по изменению в бакет-серверах RW и RO:

```
zeptoctl tasks show
```

Проверить наличие доступных на запись бакетов:

```
zeptoctl buckets
```

15 апреля 2026 г.