OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование

Не забыл включить запись

Меня хорошо видно && слышно?

Ставьте плюсы, если все хорошо

Напишите в чат, если есть проблемы

Правила вебинара

- Активно участвуем
- Задаем вопросы в чат или голосом
- Off-topic обсуждаем в Slack #канал группы или #general
- Вопросы вижу в чате, могу ответить не сразу

etcd

Маршрут вебинара

- Кратко о etcd
- Кластер etcd
- Реанимация кластера etcd

Цели занятия

После занятия вы сможете:

1. Познакомиться с распределенным хранилищем etcd
2. Понять чем etcd отличается от своих аналогов
3. Научиться разворачивать и администрировать кластер etcd

Смысл

Зачем вам это уметь:

1. Чтобы понимать основные особенности использования хранилища etcd
2. Чтобы наиболее эффективно использовать кластер etcd в вашей инфраструктуре

etcd

etcd

Вопрос к аудитории: "Знаете ли вы о etcd?"

etcd

etcd - это распределенное хранилище данных вида "ключ-значение"

Особенности:

- хранение небольших объемов метаданных в виде ключей относительно небольшого размера
- полная репликация между нодами и высокая степень доступности
- все данные пишутся на диск, in-memory отсутствует
- использует для работы алгоритм консесуса RAFT
- написан на Go, кроссплатформенный, имеет небольшой размер и большое сообщество

etcd

Преимущества:

- простой API интерфейс http + json
- иерархическая структура хранения данных по аналогии с файловой системой
- возможность отслеживание изменений (watch) и реакция на них (в этом качестве используется в Kubernetes)
- распределенные блокировки
- транзакции
- B-tree индексы для ключей

etcd

Отвечает требованиям ACID:

- Атомарность: операция либо завершается полностью, либо не завершается вовсе
- Консистентность: независимо от того к какому серверу обращается клиент он всегда получит одинаковые данные или события в том же порядке
- Изоляция: реализует самый высокий уровень изоляции Serializable
- Долговечность: все операции которые получили статус “выполненных” будут сохранены и не могут быть потеряны

etcd

Таблица сравнения etcd с аналогичными продуктами:

https://etcd.io/docs/v3.3.12/learning/why/

Алгоритм Raft

Алгоритм консенсуса Raft

Особенности:

- чёткое разделение фаз (декомпозиция задачи управления кластером на несколько, слабо связанных, подзадач)
- явно выделенный лидер (алгоритм предполагает, что в кластере всегда существует явно выделенный лидер)
- протоколы работы не могут содержать пропусков (записи добавляются строго последовательно)
- изменение размера кластера (Raft позволяет легко менять конфигурацию кластера, не останавливая его работы)

Алгоритм Raft

Ограничения алгоритма Raft:

- механизм согласования в кластере - консенсус
- количество нод в кластере должно быть равно (n / 2) + 1
- все сообщения на запись отправляются на ноду-лидер
- каждый узел кластера хранит полную копию данных ноды-лидера
- чтение может проходить на любом узле кластера
- прежде чем сохранить данные большинство узлов должны подтвердить вставку
- если умирает нода-лидер, кластер ждет определенное время и начинает голосование за нового лидера, все сообщения в это время помещаются в специальную очередь до выбора нового лидера

Алгоритм RAFT

Алгоритм консенсуса Raft который использует etcd имеет ряд ограничений:

- etcd полностью исключает воможность split-brain/multi-master так как для консенсуса нужно иметь большинство живых узлов это же может привести к ситуации полного развала кластера ввиду недостатка кворума

Ваши вопросы?

Маршрут вебинара

- Кратко о etcd
- Кластер etcd
- Реанимация кластера etcd

Кластер etcd

Кластер etcd

Особенности:

- минимально отказоустойчивый кластер можно собрать из 3 нод
- допустимое количество вышедших из строя нод можно посмотреть в таблице: https://etcd.io/docs/v2/admin_guide/#optimal-cluster-size
- теоретически количество нод не ограничено, но надо помнить о том, что любое изменение данных согласуют все ноды кластера
- задержка записи-чтения в свою очередь приводит к нестабильной работе кластера и постоянным переизбраниям мастера

Кластер etcd

Ограничение на размер запроса:

- etcd спроектирована в расчете на небольшие размеры хранимых ключей.
- большие запросы будут работать, однако это может увеличить задержку ответа
- размер запроса по-умолчанию - 1,5 мб, его можно поменять с помощью флага --max-request-bytes

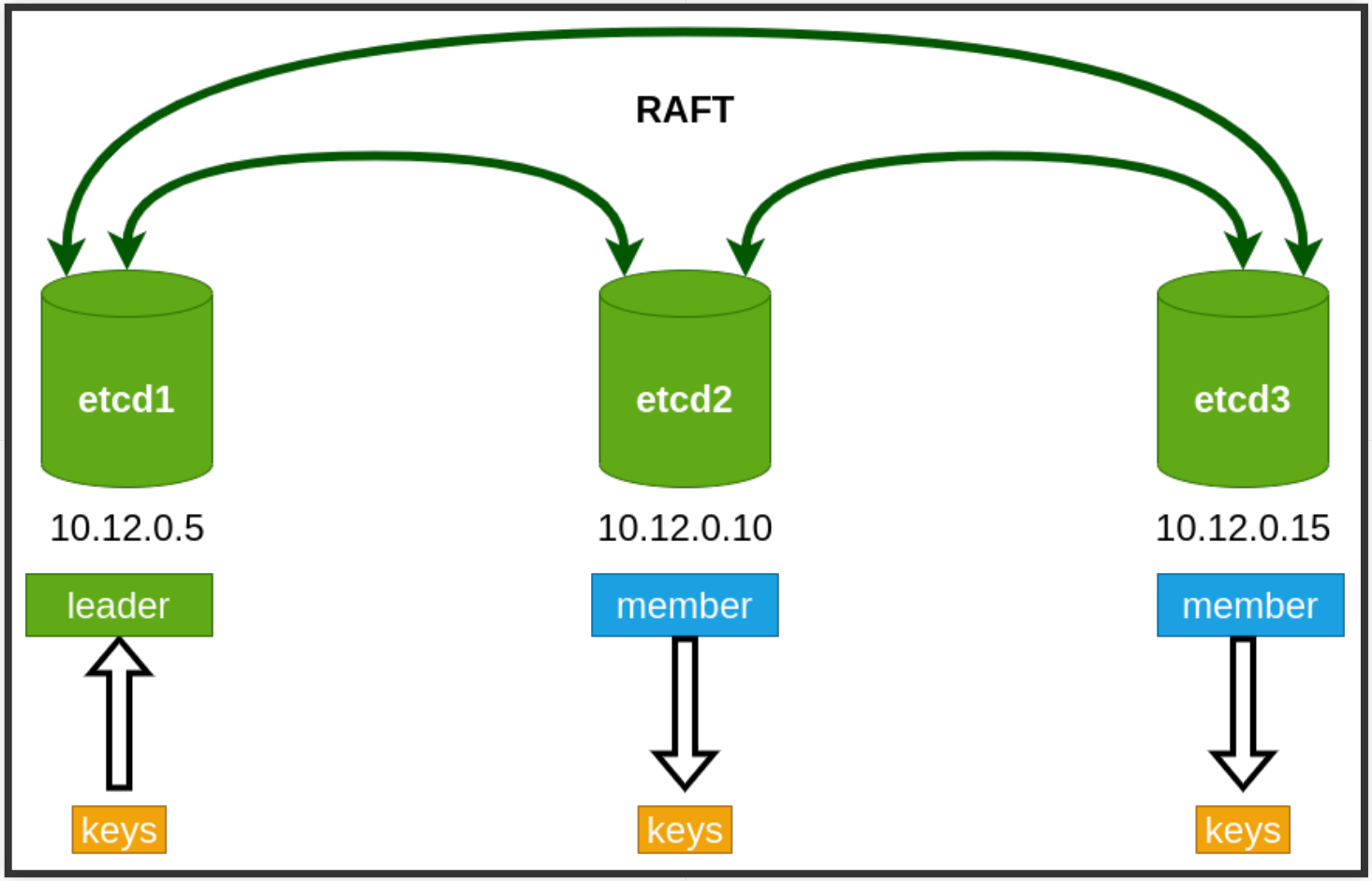
Кластер etcd

Ограничение на размер хранилища:

- размер хранилища по-умолчанию - 2 Гб
- можно расширить с помощью флага --quota-backend-bytes
- рекомендуемый размер хранилища для нормальной работы кластера - 8 Гб
- больше делать не рекомендуется из-за процессов сжатия и дефрагментации

Кластер etcd

Схема тестового стенда:



Кластер etcd

Установка etcd:

```
yum install etcd
```

Конфигурация ноды etcd1 /etc/etcd/etcd.conf:

```
ETCD_NAME="etcd1"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ADVERTISE_CLIENT_URLS="http://10.12.0.5:2379"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://10.12.0.5:2380"
ETCD_INITIAL_CLUSTER_TOKEN="TestCluster"
ETCD_INITIAL_CLUSTER="etcd1=http://10.12.0.5:2380,etcd2=http://10
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_DATA_DIR="/var/lib/etcd"
ETCD_ELECTION_TIMEOUT="5000"
ETCD_HEARTBEAT_INTERVAL="1000"
```

Кластер etcd

Конфигурация ноды etcd2 /etc/etcd/etcd.conf:

```
ETCD_NAME="etcd2"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ADVERTISE_CLIENT_URLS="http://10.12.0.10:2379"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://10.12.0.10:2380"
ETCD_INITIAL_CLUSTER_TOKEN="TestCluster"
ETCD_INITIAL_CLUSTER="etcd1=http://10.12.0.5:2380,etcd2=http://10
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_DATA_DIR="/var/lib/etcd"
ETCD_ELECTION_TIMEOUT="5000"
ETCD_HEARTBEAT_INTERVAL="1000"
```

Кластер etcd

Конфигурация ноды etcd3 /etc/etcd/etcd.conf:

```
ETCD_NAME="etcd3"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ADVERTISE_CLIENT_URLS="http://10.12.0.15:2379"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://10.12.0.15:2380"
ETCD_INITIAL_CLUSTER_TOKEN="TestCluster"
ETCD_INITIAL_CLUSTER="etcd1=http://10.12.0.5:2380,etcd2=http://10
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_DATA_DIR="/var/lib/etcd"
ETCD_ELECTION_TIMEOUT="5000"
ETCD_HEARTBEAT_INTERVAL="1000"
```

Работа с кластером

Экспорт переменной с версией API etcd, чтобы не объявлять ее каждый раз:

```
export ETCDCTL_API=3
```

Посмотреть статус текущей ноды:

```
etcdctl --write-out=table endpoint status
```

Посмотреть статусы нод кластера:

```
etcdctl member list
```

Работа с кластером

Удалить участника:

```
etcdctl member remove <member_id>
```

Добавить участника в кластер:

```
etcdctl member add etcd4 --peer-urls=http://10.12.0.20:2380

ETCD_NAME="etcd4"
ETCD_INITIAL_CLUSTER="etcd3=http://10.12.0.15:2380,etcd2=http://1
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://10.12.0.20:2380"
ETCD_INITIAL_CLUSTER_STATE="existing"
```

Работа с кластером

Добавить участника в кластер:

Объявляем конфигурацию в переменных на ноде нового участника кластера:

```
export ETCD_NAME="etcd4"
export ETCD_INITIAL_CLUSTER="etcd3=http://10.12.0.15:2380,etcd2=h
export ETCD_INITIAL_CLUSTER_STATE=existing
```

Запускаем etcd с набором параметров:

```
etcd --listen-client-urls http://10.12.0.20:2379 --advertise-clie
```

Работа с кластером

Добавить участника в кластер:

Добавляем конфиг для запуска сервиса etcd:

```
ETCD_NAME="etcd4"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_ADVERTISE_CLIENT_URLS="http://10.12.0.20:2379"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://10.12.0.20:2380"
ETCD_INITIAL_CLUSTER_TOKEN="TesCluster"
ETCD_INITIAL_CLUSTER="etcd3=http://10.12.0.15:2380,etcd2=http://1
ETCD_INITIAL_CLUSTER_STATE="existing"
ETCD_DATA_DIR="/var/lib/etcd"
ETCD_ELECTION_TIMEOUT="5000"
ETCD_HEARTBEAT_INTERVAL="1000"
```

Работа с кластером

Добавить участника в кластер:

Меняем владельца каталога с файлами базы:

```
chown -R etcd:etcd /var/lib/etcd
```

Запускаем сервис etcd:

```
systemctl start etcd
```

Обслуживание кластера etcd

Сжатие - освобождение дискового пространства

Особенности:

- в etcd существует единый счетчик ревизий, который начинается с нуля
- ревизия - последняя версия измененного ключа
- сжатие по сути удаляет историю пространства ключей (ревизии)

Обслуживание кластера etcd

Сжатие - освобождение дискового пространства

- флаг для включения автоматического периодического сжатия:

```
--auto-compaction-retention=1
```

- запуск сжатия вручную, с указанием версии ревизии для всех ключей:

```
etcdctl compact 3
```

Обслуживание кластера etcd

Дефрагментация - удаляет незаполненные места в базе данных

Запуск дефрагментации в кластере:

```
etcdctl defrag-cluster
```

- одного сжатия данных недостаточно для высвобождения дискового пространства, так как база данных фрагментирована после сжатия
- при сжатии устаревшие данные просто удаляются, оставляя пробелы в базе данных, которые по-прежнему приводят к использованию дискового пространства

Обслуживание кластера etcd

Дефрагментация - удаляет незаполненные места в базе данных

- дефрагментация на работающем инстансе кластера блокирует чтение и запись, пока состояние кластера не будет перестроено
- чем больше размер базы - тем дольше проходит дефрагментация
- можно запустить для отдельной ноды, можно - для всего кластера
- в случае если дефрагментация превзойдет по времени тайм-аут выбора лидера - произойдет переизбрание лидера

Обслуживание кластера etcd

Снапшоты - бэкап ключевого пространства базы данных в файл

Создать снапшот:

```
etcdctl snapshot save /var/tmp/etcd.backup
```

Просмотр состояния файла снапшота:

```
etcdctl --write-out=table snapshot status /var/tmp/etcd.backup
```

Ваши вопросы?

Маршрут вебинара

- Кратко о etcd
- Кластер etcd
- Реанимация кластера etcd

Реанимация кластера etcd

Реанимация кластера etcd

Примерная схема падения кластера:

- любая причина, оказавшая воздействие на большинство участников
- количества оставшихся участников кластера недостаточно для кворума по выборам лидера
- оставшиеся без лидера участники перестают отдавать ключи

Реанимация кластера etcd

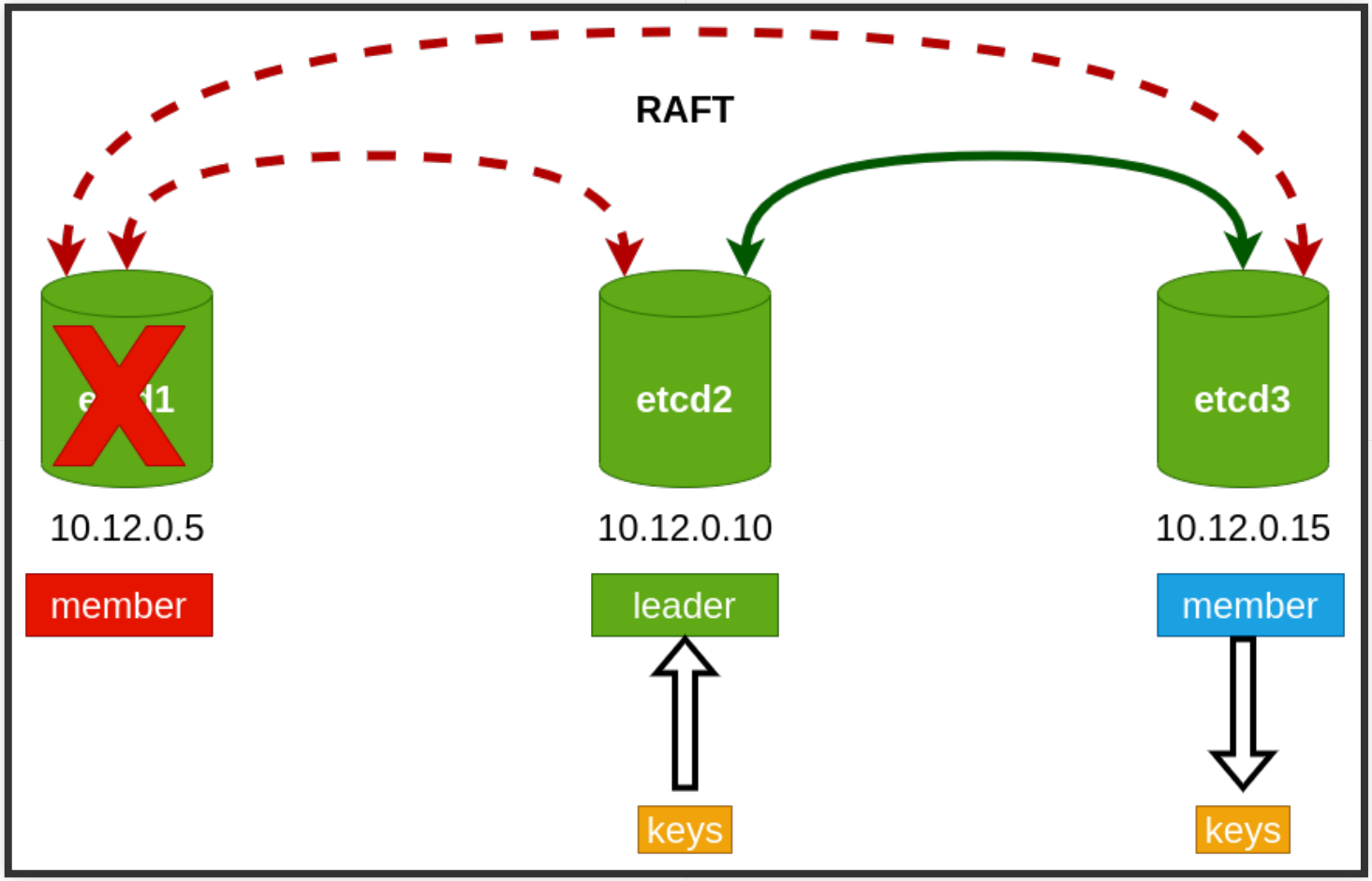
Ситуация №1: потеря минимума участников, оставшихся достаточно для кворума

Решение проблемы:

1. Восстановление сервиса на вышедших из строя нодах
2. Введение в строй новых нод

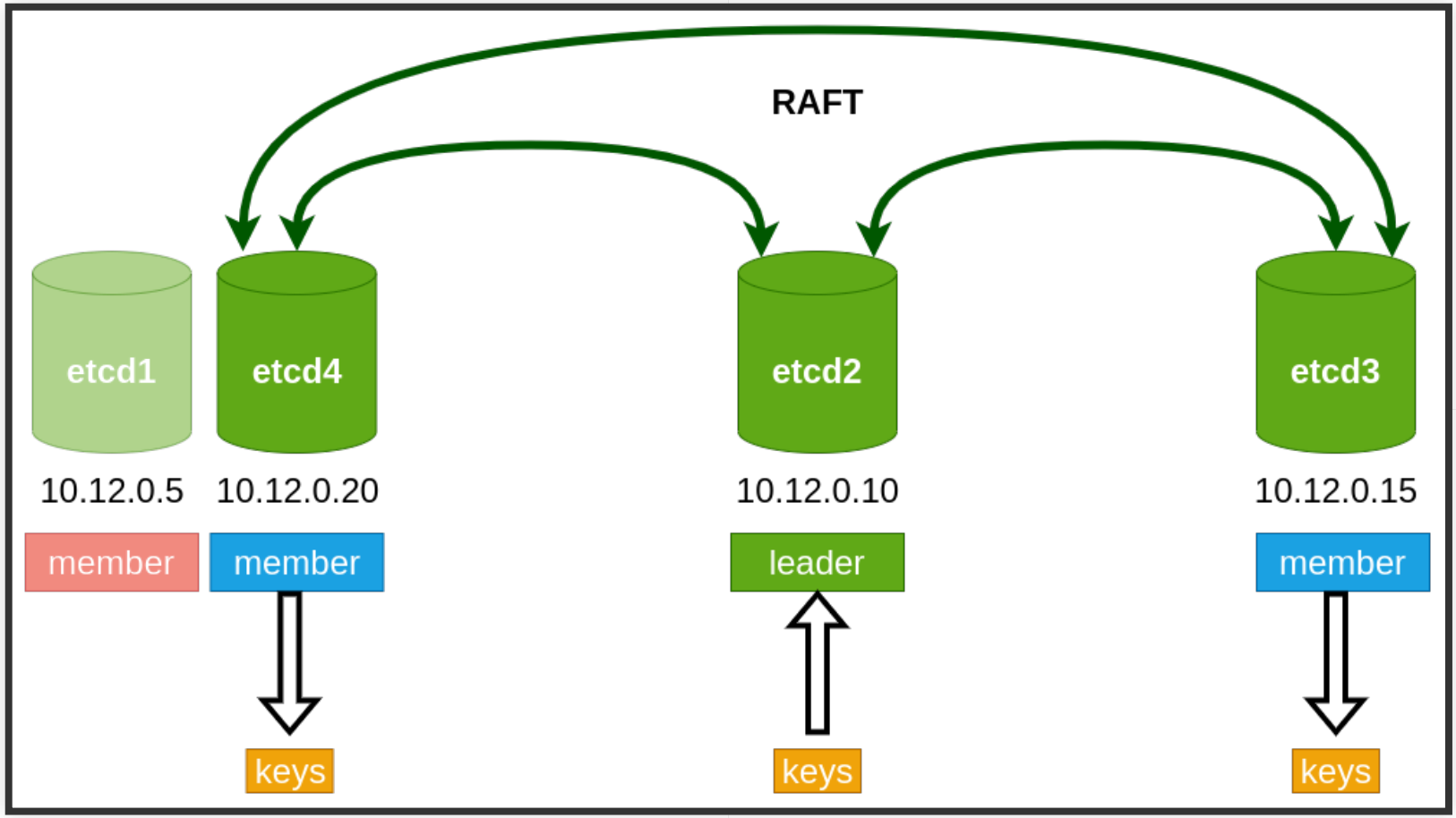
Реанимация кластера etcd

Схема тестового стенда:



Реанимация кластера etcd

Схема тестового стенда:



Реанимация кластера etcd

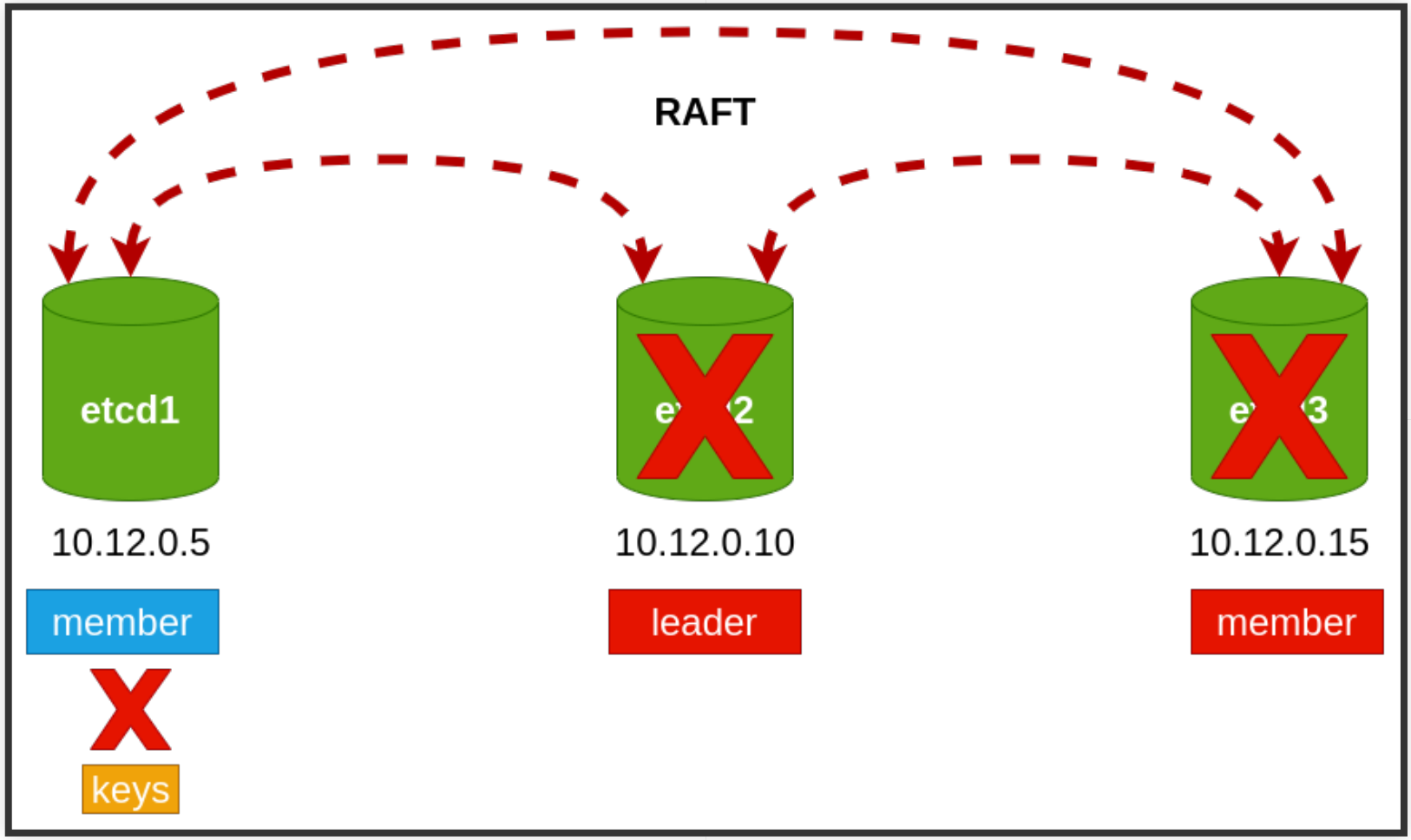
Ситуация №2: потеря максимума участников, оставшихся недостаточно для кворума

Решение проблемы:

1. Снятие снапшота базы на оставшейся ноде
2. Перенос снапшота на новые ноды
3. Формирование нового логического кластера со старыми данными в нем

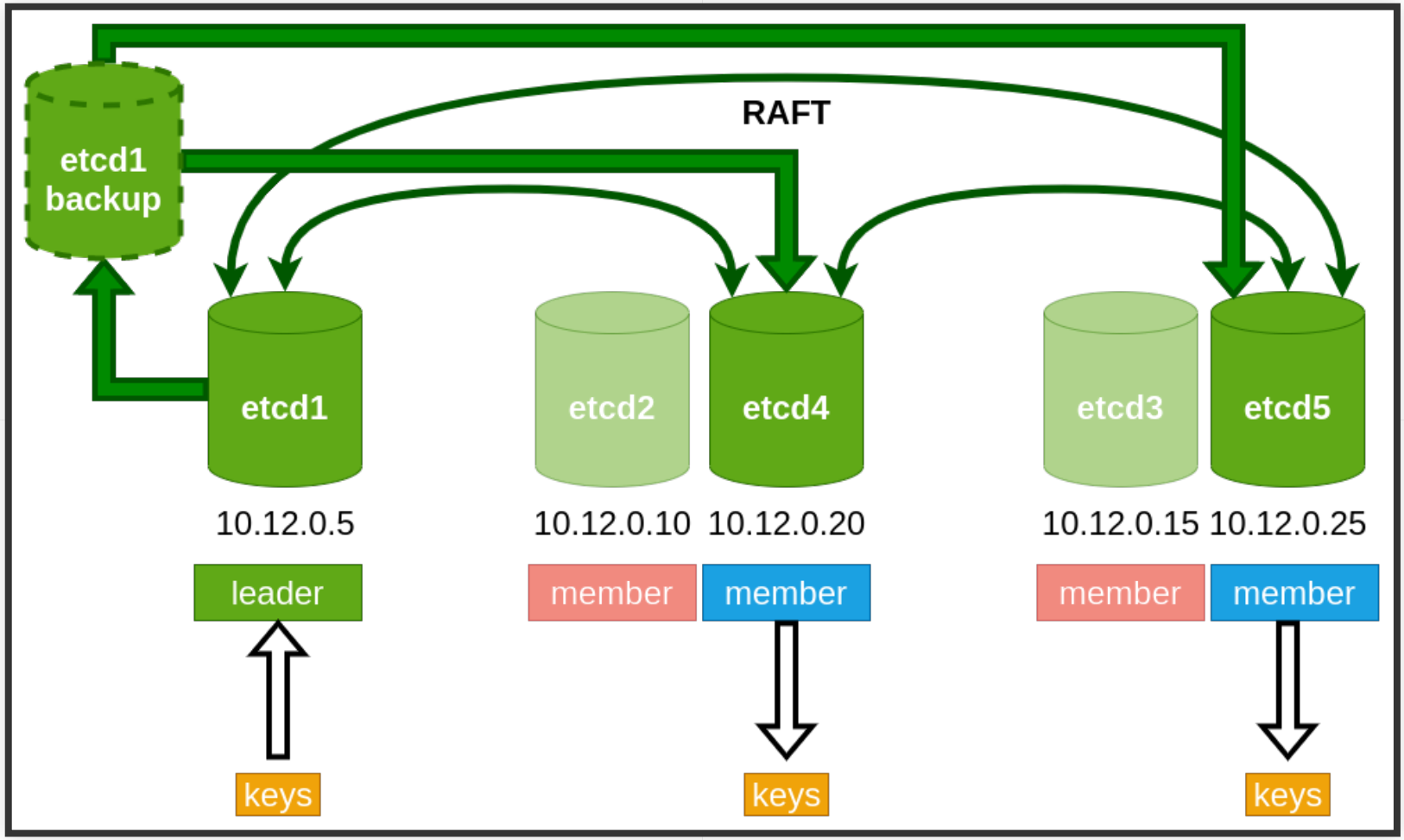
Реанимация кластера etcd

Схема тестового стенда:



Реанимация кластера etcd

Схема тестового стенда:



Ваши вопросы?

Заполните, пожалуйста, опрос о занятии по ссылке в чате

Приходите на следующие вебинары

Спасибо за внимание!