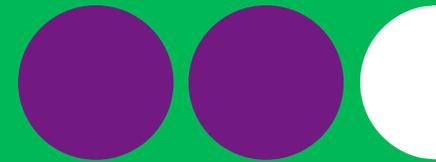


МЕГАФОН



СОЗДАНИЕ НА-КЛАСТЕРА ZABBIX PROXY

КРАСНОДАР, ФЕВРАЛЬ 2021

Содержание

1

Текущая инфраструктура сервиса

2

Описание возможных вариантов реализации

3

Описание финального варианта

4

Распределение нагрузки по группе хостов

5

Переключение прокси (switchover/failover)

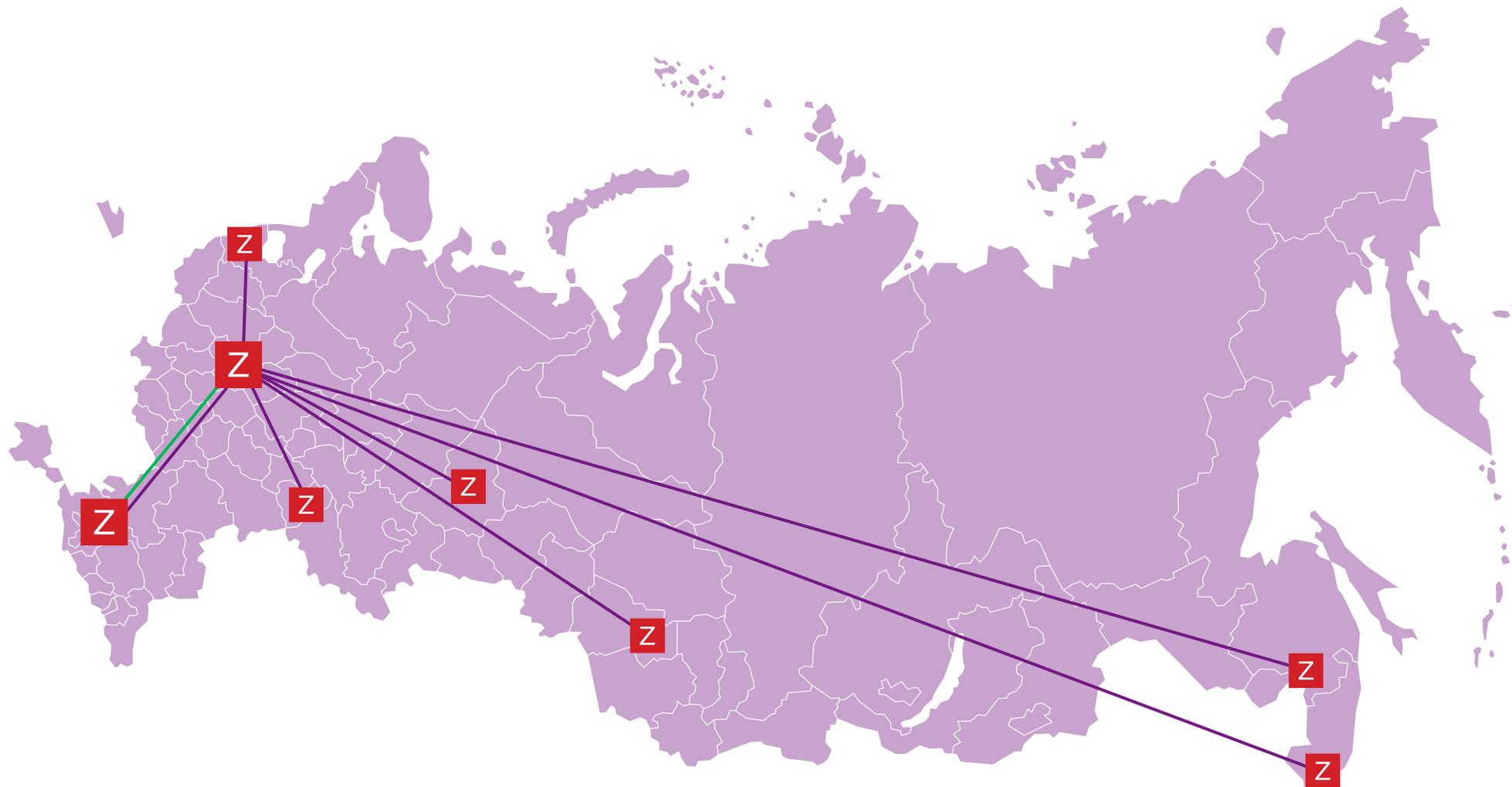
1

ТЕКУЩАЯ ИНФРАСТРУКТУРА СЕРВИСА

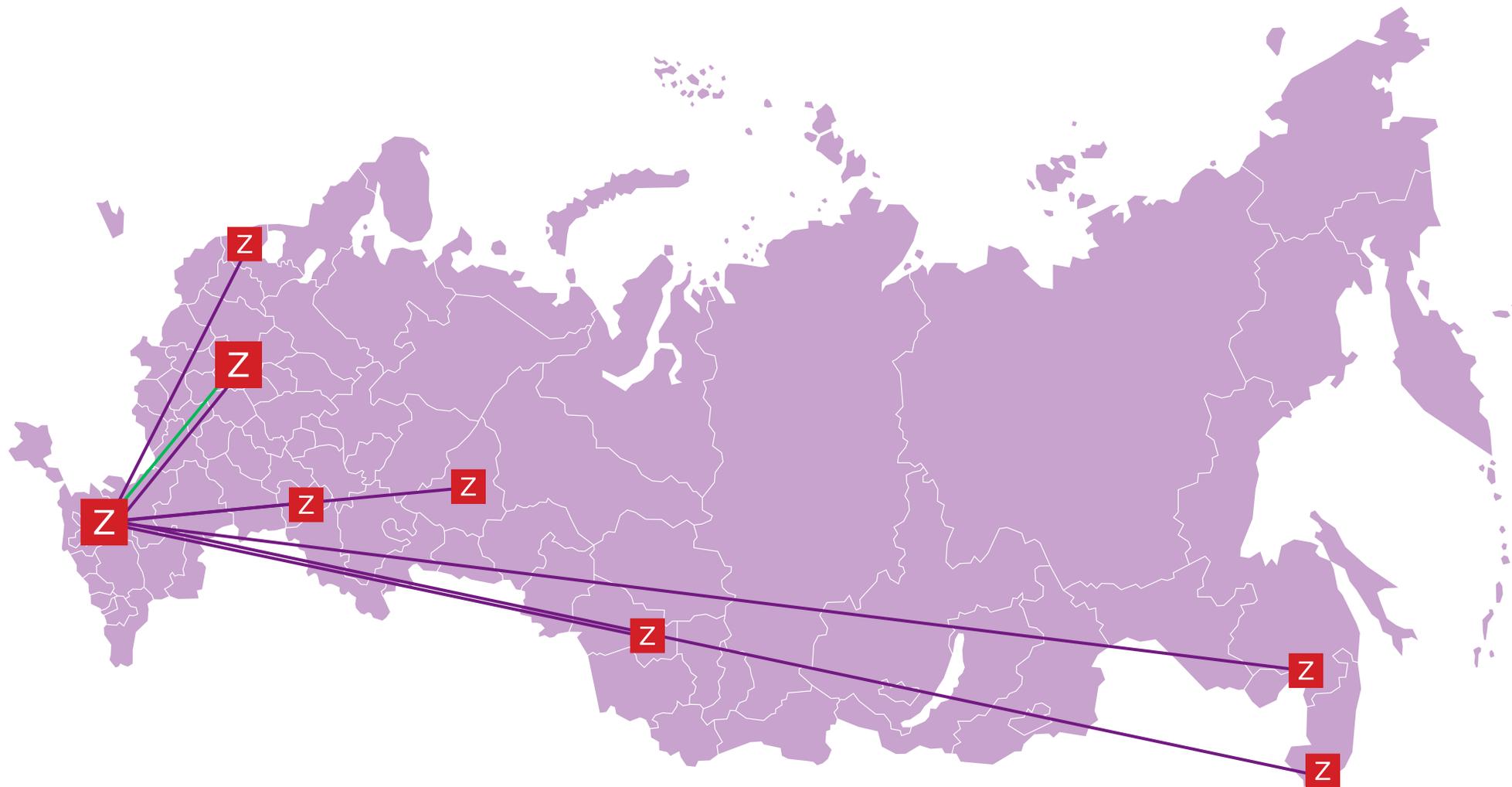
Текущая инфраструктура сервиса мониторинга



Текущая инфраструктура сервиса мониторинга



Текущая инфраструктура сервиса мониторинга



Текущая инфраструктура сервиса мониторинга

Zabbix Proxy

Количество: 28 серверов

Тип сервера: VM

OS: RHEL 7.6

CPU: 12 cores

RAM: 32 Gb

HDD: 300 Gb (SAN, HDD)

Сервер СУБД (Oracle)

Количество нод в кластере: 2

Тип сервера: HW

OS: RHEL 7.6

CPU: 80 cores

RAM: 512 Gb

HDD: 20 Tb (SAN, SSD)

Zabbix Server

Количество: 2 сервера

Тип сервера: VM

OS: RHEL 7.6

CPU: 16 cores

RAM: 64 Gb

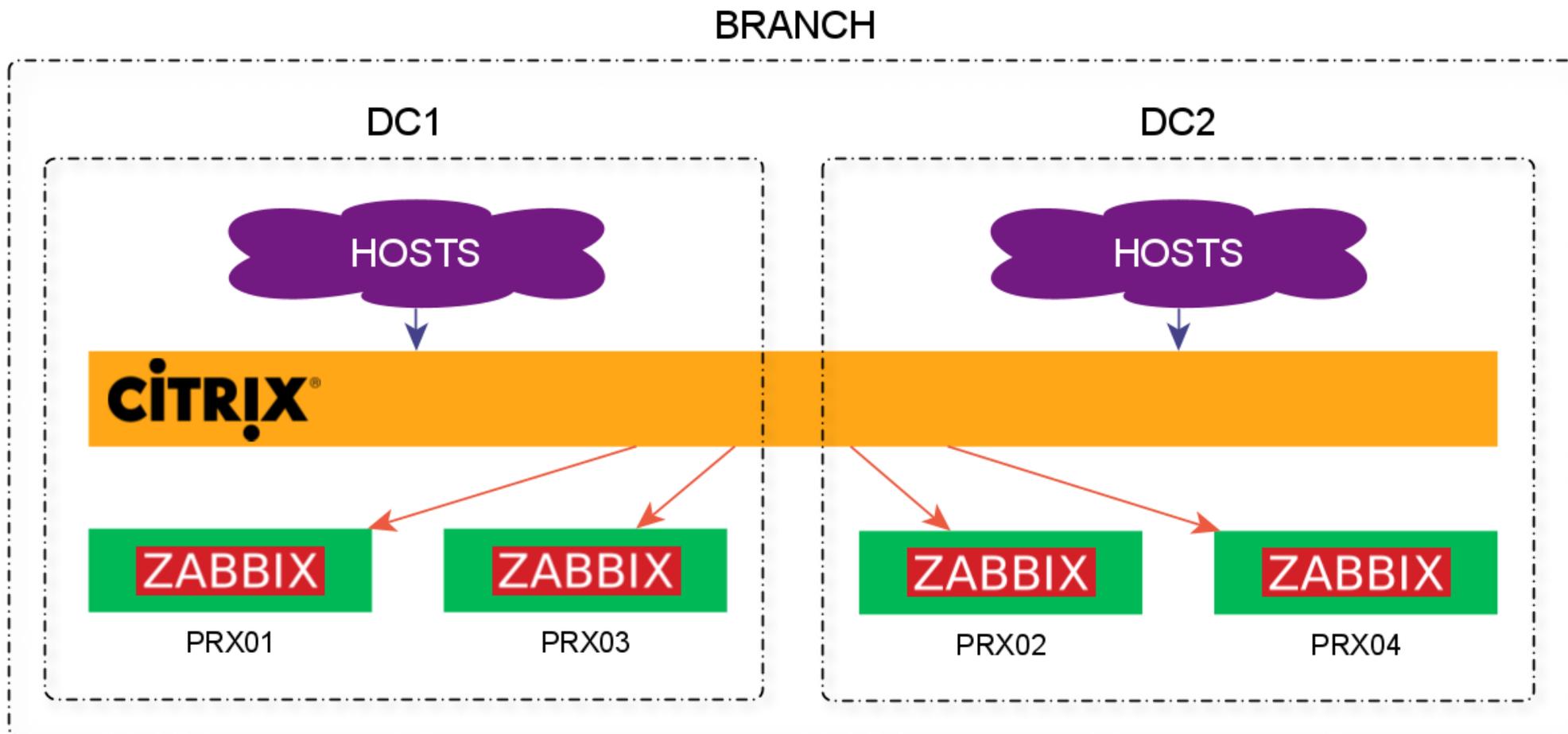
HDD: 1.2 Tb (SAN, HDD)



2

ОПИСАНИЕ ВОЗМОЖНЫХ ВАРИАНТОВ РЕАЛИЗАЦИИ

Терминация трафика на стороне Citrix Netscaler



Терминация трафика на стороне Citrix Netscaler

Описание тестовой среды

Citrix Netscaler:

Метод балансировки: Persistence.

zabbix_agentd.conf

```
Server=127.0.0.1,prx01,prx02,prx03,prx04
```

```
ServerActive=dns_name_of_gslb
```

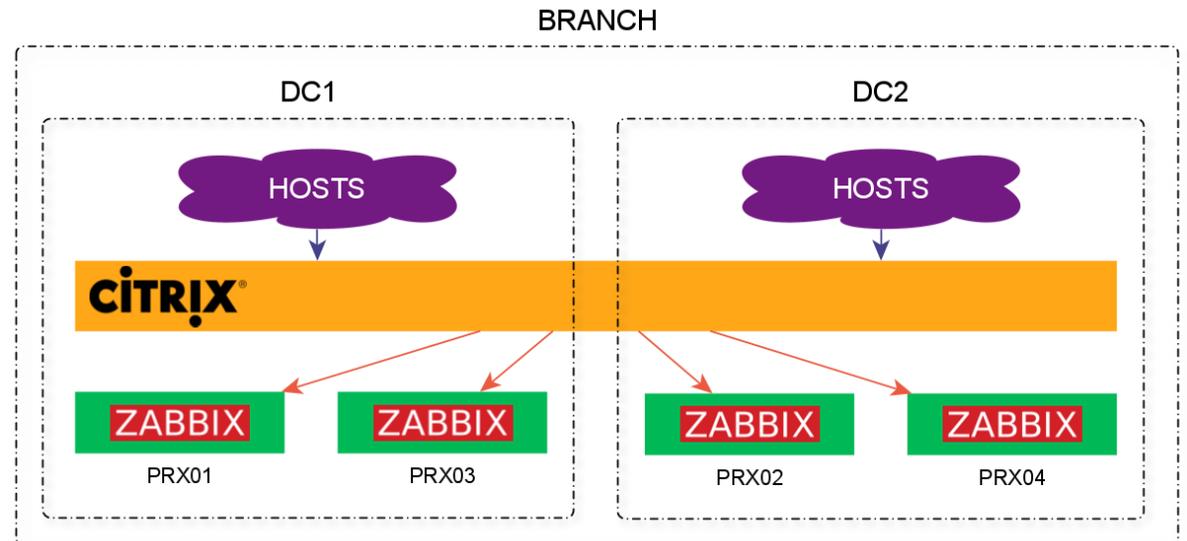
Данный метод предполагает использование авторегистрации.

Решение:

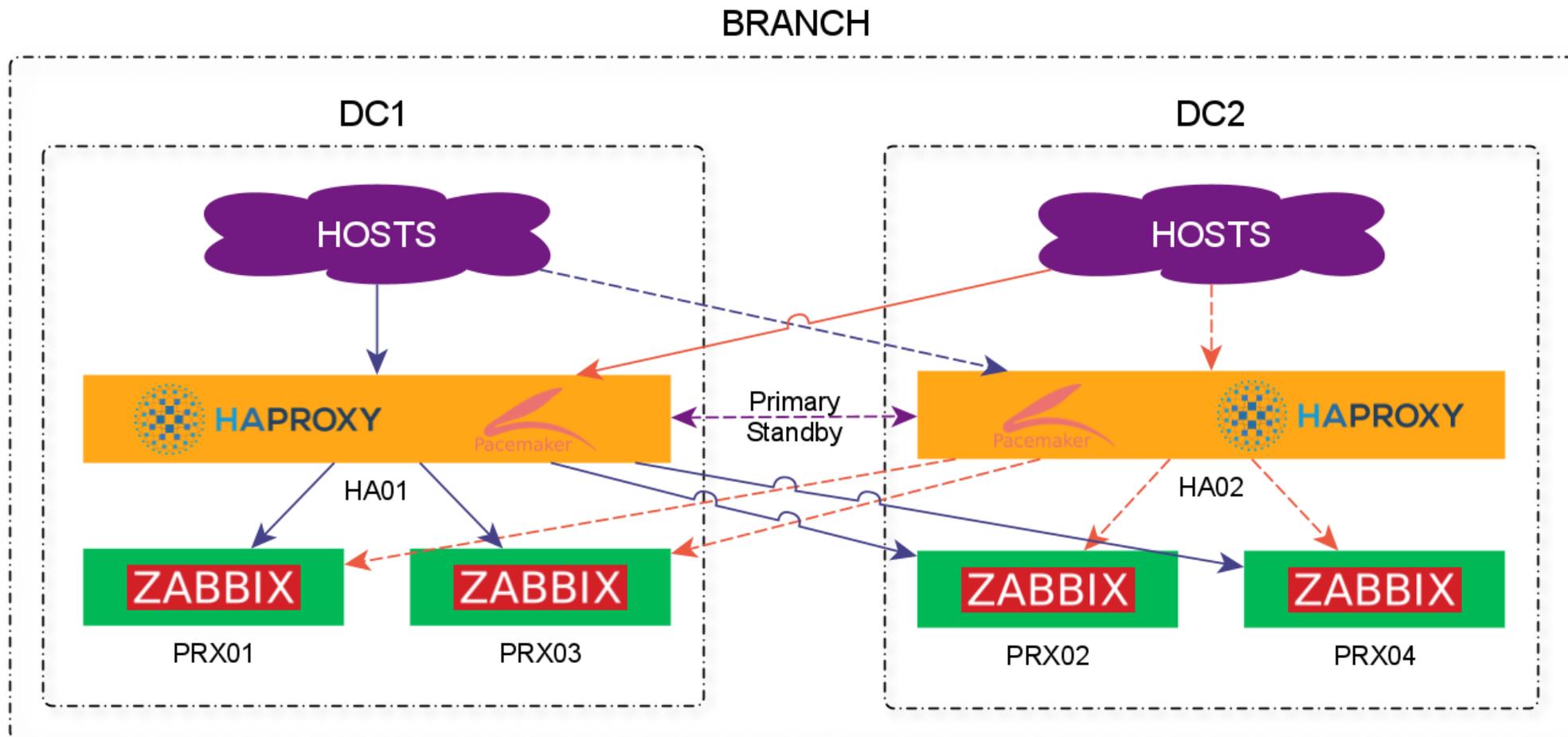
Отрицательное, так как для нужд сервиса мониторинга пришлось бы разворачивать отдельную инфраструктуру сервиса балансировки на базе Citrix Netscaler.

Потенциальные точки отказа в виде DNS:

HOST → DNS → GSLB



Терминация трафика на стороне HAProxy



Терминация трафика на стороне HAProxy

Описание тестовой среды

HAProxy:

Метод балансировки: Source.

Pacemaker:

1. STONITH;
2. two node cluster (no quorum).

zabbix_agentd.conf

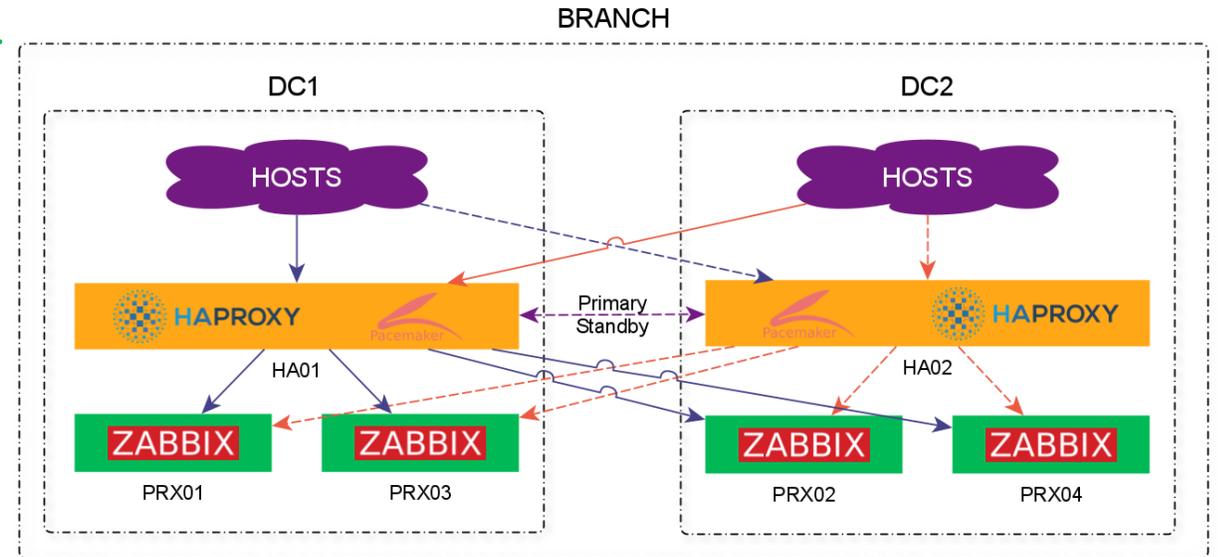
```
Server=127.0.0.1,prx01,prx02,prx03,prx04  
ServerActive=ha01,ha02
```

Данный метод предполагает использование авторегистрации.

Решение:

Отрицательное, так как:

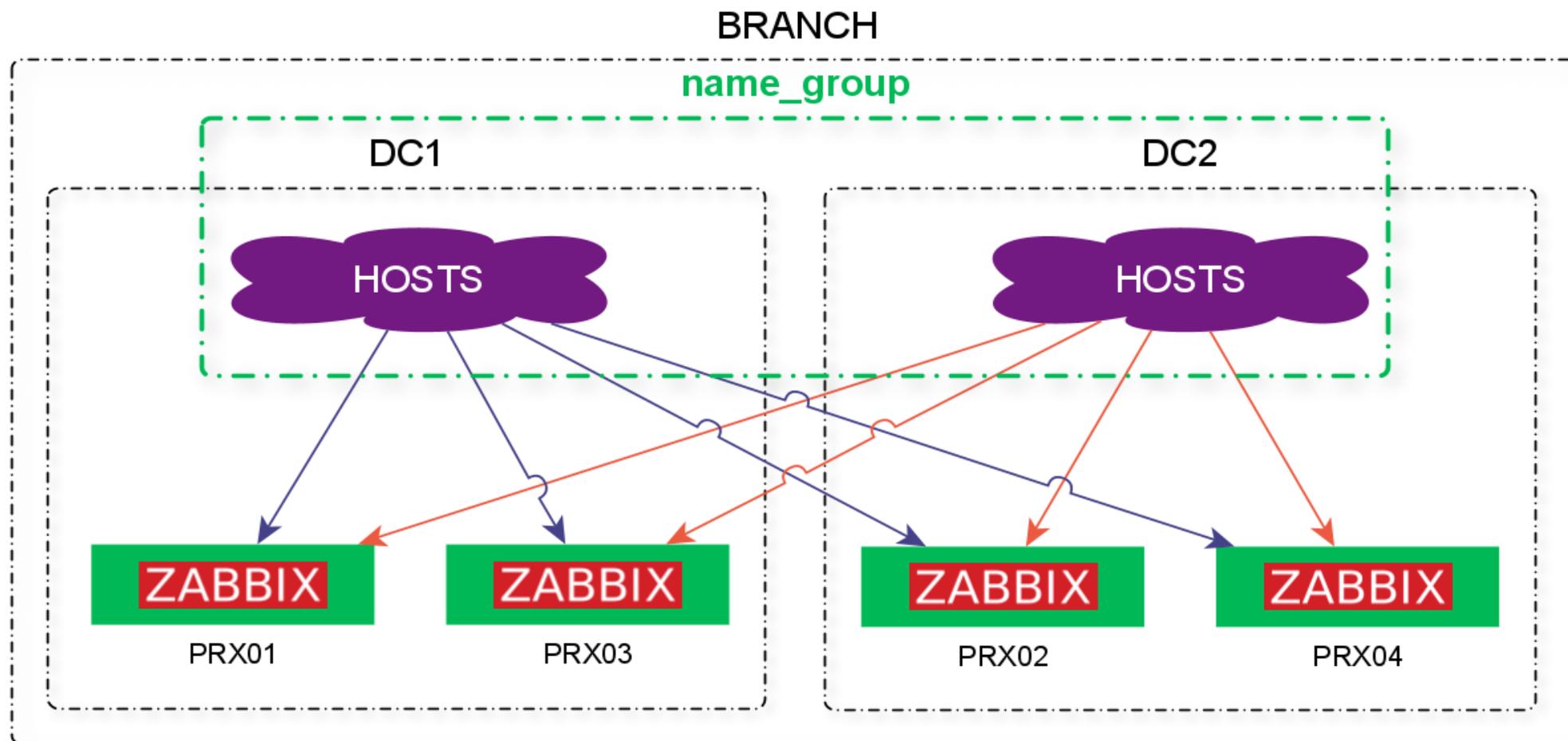
1. в любом случае нужен арбитр;
2. усложнение инфраструктуры;
3. риски геораспределённой среды (разрыв сетевой связанности);
4. не ясно как реплицировать хэши HAProxy.



3

ОПИСАНИЕ ФИНАЛЬНОГО ВАРИАНТА

Балансировка методами API



Балансировка методами API

Описание

Метод балансировки: Скрипты на стороне сервера Zabbix.

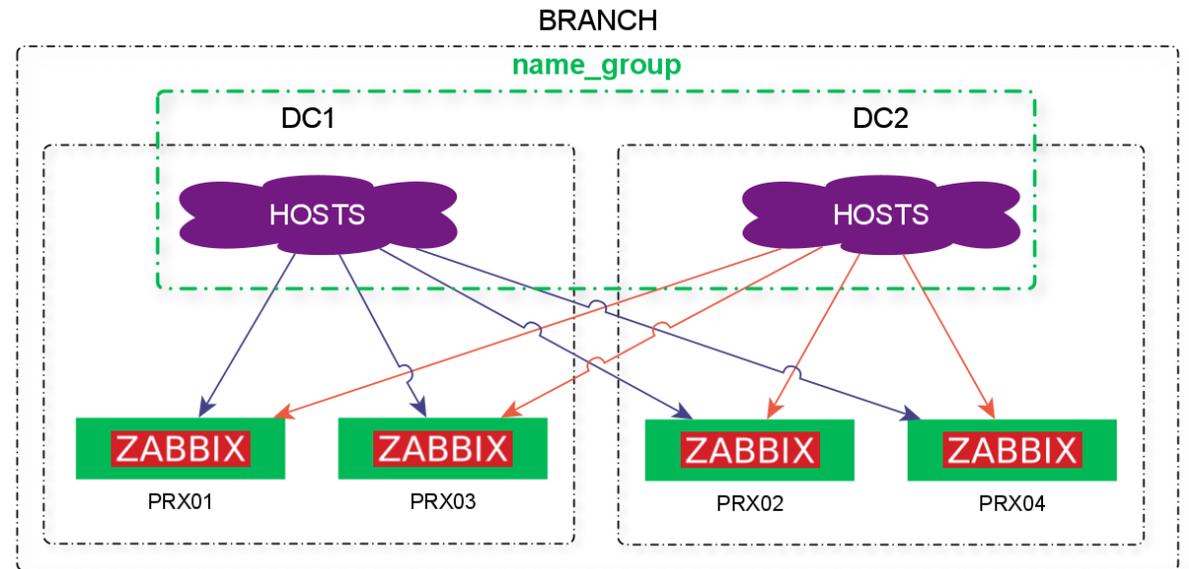
zabbix_agentd.conf

```
Server=127.0.0.1,prx01,prx02,prx03,prx04
```

```
ServerActive=prx01,prx02,prx03,prx04
```

Особенности:

Авторегистрация в этом варианте не может использоваться и должна быть отключена.



Балансировка методами API

Задача: распределить нагрузку на Zabbix прокси по количеству хостов.

Пусть в группе хостов (name_group) содержатся 1011 узлов сети.

Вопрос: необходима ли балансировка? Ответ: **ДА** или **НЕТ**

Входные данные:

1. name_group - (тип данных текст) имя группы хостов;
2. N - (тип данных число) допустимая разница в количестве хостов между конкретными Zabbix прокси. При превышении данного значения, будет запущен механизм по распределению нагрузки. Пусть N=10.

Вспомогательные данные:

1. внутри программы содержится информация о прокси, какие с кем считать взаимосвязанными.

Запуск скрипта в формате: script1.py is_need_balancing name_group N

Выходные данные: программа вернет **TRUE** или **FALSE**, ответ позволит настроить триггер в Zabbix.



Балансировка методами API

Алгоритм работы программы:

1. рассчитать количество хостов, по Zabbix прокси, из группы name_group;
2. находим количественный минимум и максимум содержания хостов на Zabbix прокси.

Формула: $M = \text{MAX} - \text{MIN}$

Если $M > N$: то **TRUE** (необходимо распределение)

Иначе: **FALSE** (распределение не требуется)

ИЛИ хост(ты) расположены на неизвестной Zabbix прокси.



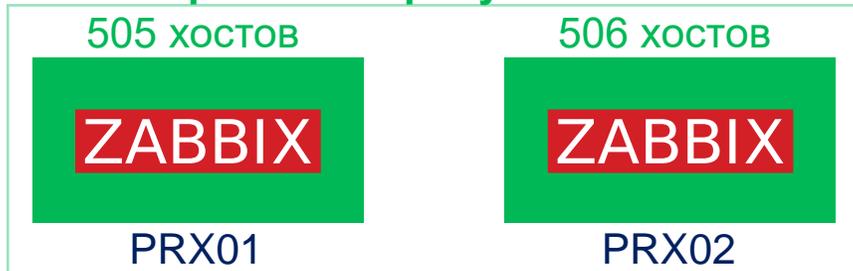
4

РАСПРЕДЕЛЕНИЕ НАГРУЗКИ ПО ГРУППЕ ХОСТОВ

Распределение нагрузки по группе хостов

Примеры распределения хостов по Zabbix прокси: **prx01** и **prx02**

Балансировка не требуется



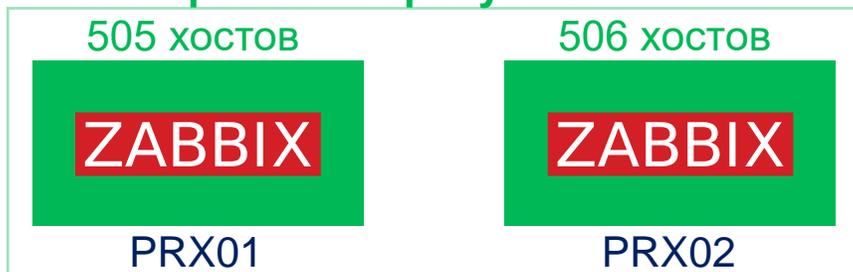
$M = 1, N = 10$



Распределение нагрузки по группе хостов

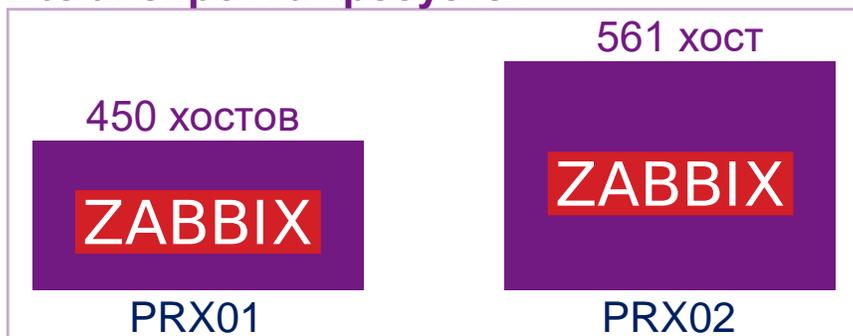
Примеры распределения хостов по Zabbix прокси: prx01 и prx02

Балансировка не требуется



$M = 1, N = 10$

Балансировка требуется



$M = 111, N = 10$



Распределение нагрузки по группе хостов

Примеры распределения хостов по Zabbix прокси: **prx01**, **prx02** и **prx03**, плюс есть хосты на неизвестной скрипту прокси

Балансировка не требуется



Распределение нагрузки по группе хостов

Примеры распределения хостов по Zabbix прокси: **prx01**, **prx02** и **prx03**, плюс есть хосты на неизвестной скрипту прокси

Балансировка не требуется



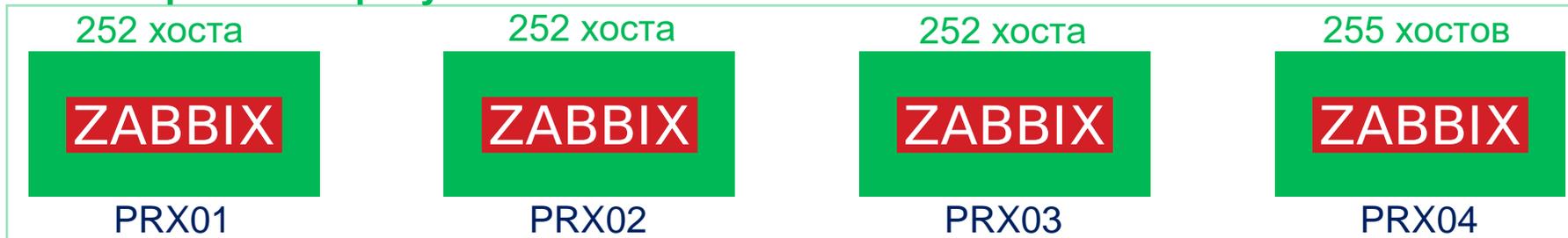
Балансировка требуется



Распределение нагрузки по группе хостов

Примеры распределения хостов по Zabbix прокси: prx01, prx02, prx03 и prx04

Балансировка не требуется



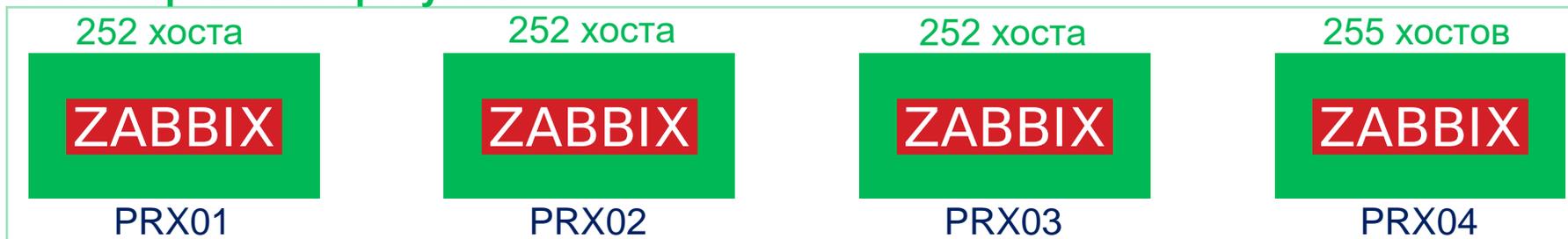
$M = 3, N = 10$



Распределение нагрузки по группе хостов

Примеры распределения хостов по Zabbix прокси: prx01, prx02, prx03 и prx04

Балансировка не требуется



$M = 3, N = 10$

Балансировка требуется



$M = 71, N = 10$



Описание скрипта, который распределяет нагрузку

Условие:

- 1.нет активного события Switchover/Failover (см. описание скрипта по переключению Zabbix прокси);
- 2.проверка по балансировке = **TRUE**.

Если все условия выполняются, скрипт по распределению нагрузки может быть запущен, в формате:

```
script1.py balancing name_group
```

Алгоритм работы программы:

Расчет переменной (LIMIT) - это допустимое значение количества хостов на одну прокси:

LIMIT = Количество хостов // Количество Zabbix прокси (деление без остатка)

$$LIMIT = len(hosts) // len(prxs)$$

Если остаток **>0**, тогда переменную LIMIT Увеличиваем на **+1**

$$if (len(hosts) \% len(prxs)) > 0: LIMIT += 1$$

Алгоритм проходит по каждой Zabbix прокси и отсчитывает допустимое значение **<=LIMIT**, остальное отправляет в буфер.

Для хостов из буфера будет применен метод Zabbix API (host.massupdate), что позволит распределить нагрузку на менее нагруженную прокси.



Пример работы скрипта по распределению нагрузки

Исходная ситуация

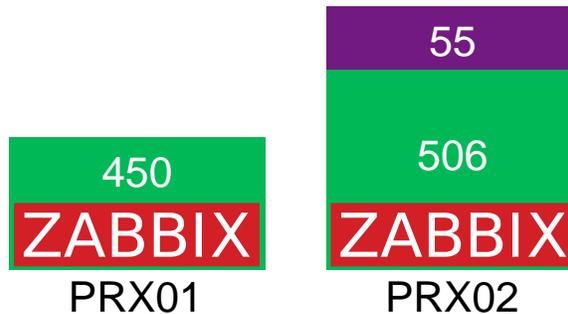


Пример работы скрипта по распределению нагрузки

Исходная ситуация

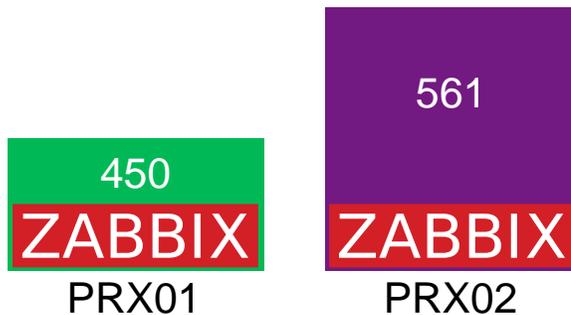


Наполнение буфера



Пример работы скрипта по распределению нагрузки

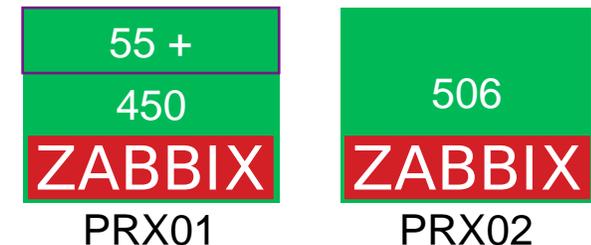
Исходная ситуация



Наполнение буфера



Перенос хостов



Пример работы скрипта по распределению нагрузки



Пример работы скрипта по распределению нагрузки



$$\text{LIMIT} = 1011 / 3 = 337$$



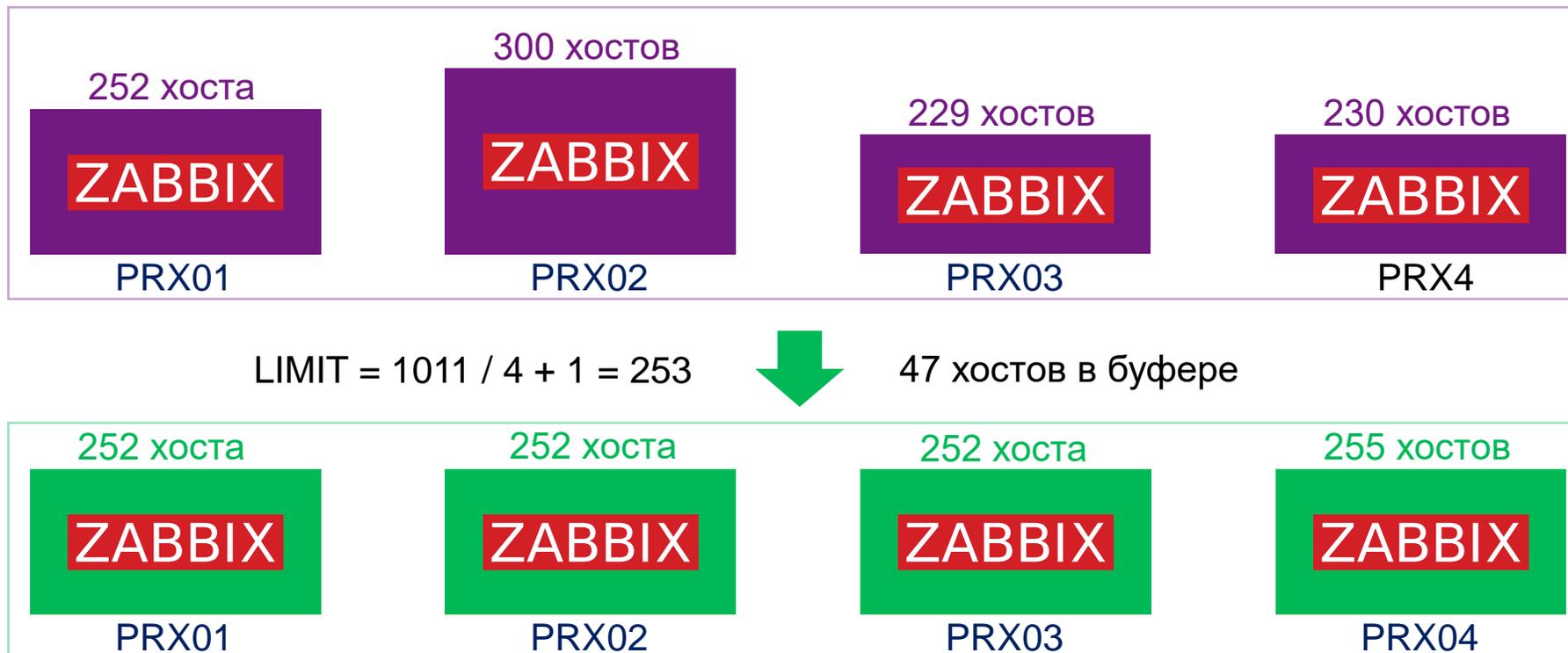
2 хоста в буфере



Пример работы скрипта по распределению нагрузки



Пример работы скрипта по распределению нагрузки



Описание скрипта, который распределяет нагрузку

Минусы и особенности решения

1. Все хосты, подлежащие балансировке, должны состоять в определённой группе. Все вновь создаваемые хосты должны помещаться в эту группу.

2. Неудобство с отсутствием авторегистрации.

Выход из ситуации: бот, занимающийся автоматической постановкой новых хостов на мониторинг по заявкам, проверяет:

- пингуется ли хост по DNS, имя хоста берёт из заявки;
- пингует агента (*agent.ping*);
- проверяет соответствие имени хоста, указанного в заявке, с именем хоста, возвращаемом агентом (*agent.hostname*).

В случае, если все три проверки успешны, бот создает хост методом API.

Если нет - отправляет заявку на рассмотрение специалистами.

3. Мы балансируем хосты, а не vps! (Но, справедливости ради, ни NetScaler, ни HAProxy vps тоже не учитывают.)

Прокси	Число хостов	Число vps
PRX01	100	1000
PRX02	101	12000



5

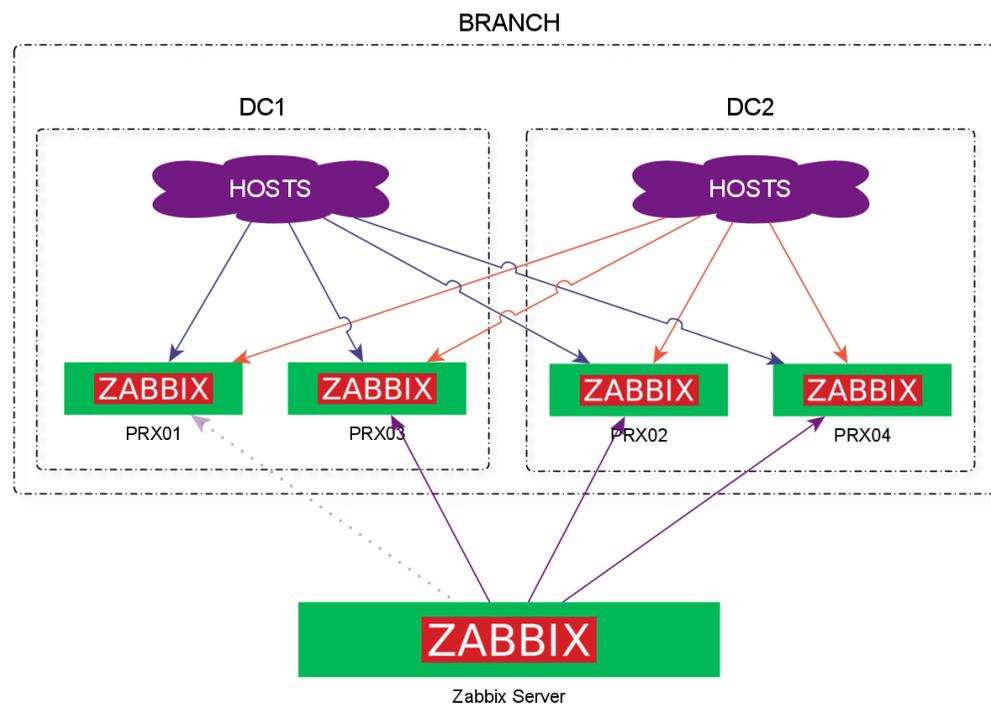
ПЕРЕКЛЮЧЕНИЕ ПРОКСИ (SWITCHOVER/FAILOVER)

ПЕРЕКЛЮЧЕНИЕ ПРОКСИ (SWITCHOVER/FAILOVER)

Задача: в случае недоступности одной из Zabbix прокси, перенести ее хосты на другую.

Условие:

1. Zabbix сервер перестал видеть Zabbix прокси по порту 10051 (*net.tcp.service[tcp,prx_IP,10051]*);
2. последнее сообщение, полученное Zabbix сервером от Zabbix прокси, получено N секунд назад (*zabbix[proxy,prx_name,lastaccess]*)



ПЕРЕКЛЮЧЕНИЕ ПРОКСИ (SWITCHOVER/FAILOVER)

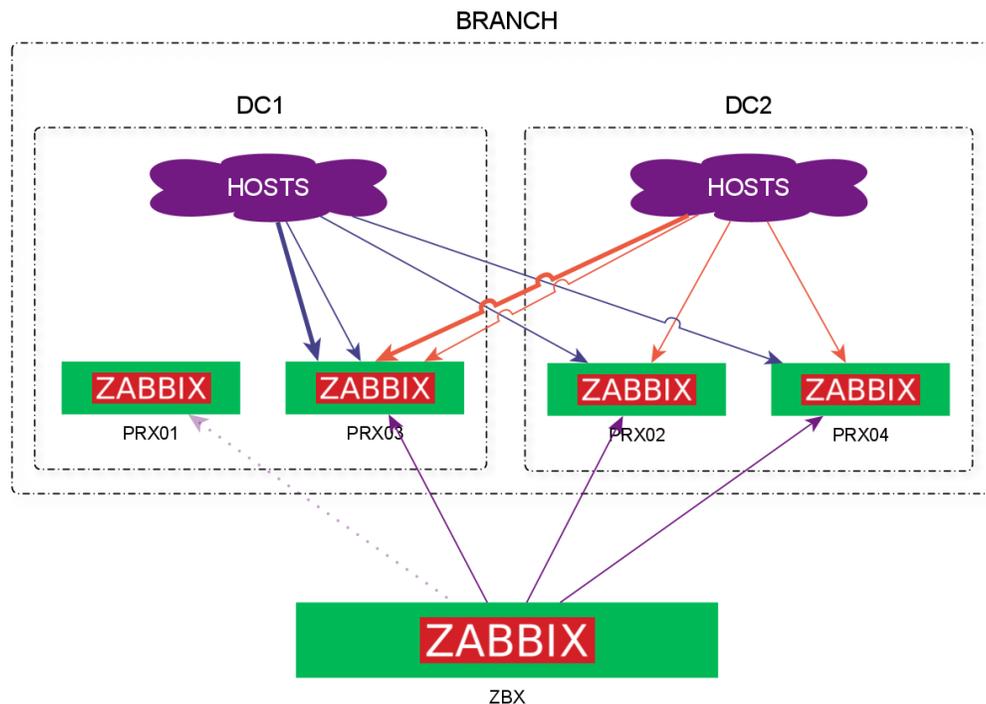
Действие:

Все хосты с недоступной Zabbix прокси переключаются на оставшуюся на площадке.

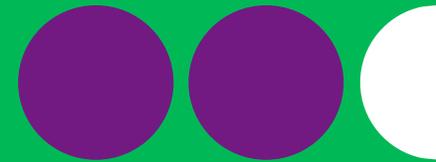
Примечание:

Триггер по распределению нагрузки на Zabbix прокси зависит от триггера по недоступности любой из Zabbix прокси. То есть пока одна из Zabbix прокси недоступна, распределение нагрузки никогда не запустится.

Скрипт в формате: script2.py prx01



МЕГАФОН



СПАСИБО ЗА ВНИМАНИЕ

Прожирова Светлана, Диденко Алексей