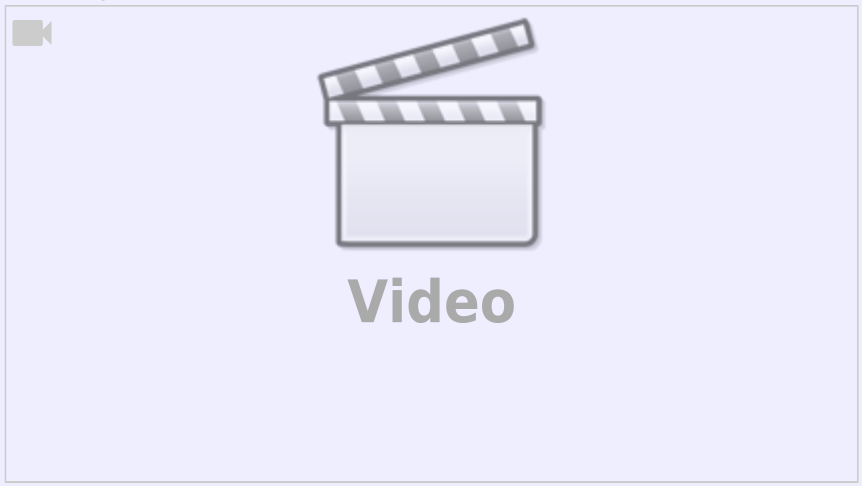
Содержание

Резервирование BRAS

Вебинар по теме:



В СКАТ 8.3+ применяется следующая схема репликации для согласования данных об абонентах на всех fastDPI-серверах: fastPCRF шлет ответы авторизации и CoA-запросы на все серверы, перечисленные в параметрах fdpi_server.

Отправка параметров авторизации производится через персистентную очередь, так что даже если какой-то из серверов fastDPI был отключен на момент отправки данных, при включении он получит все данные за время своего простоя.

Применение данных на fastDPI

При приеме данных авторизации сервер fastDPI видит, его это был запрос или же это ответ на чужой запрос (для этого в пакете есть специальная метка). Если это ответ на свой запрос, данные применяются "по полной": создается DHCP или PPPoE--сессия, если это был запрос DHCP или PPPoE-авторизации, данные запоминаются в UDR. Если же это ответ на чужой запрос, fastDPI просто запоминает "чужие" данные у себя в UDR. Тем самым при отключении основного fastDPI-сервера и переводе нагрузки на резервный, у резервного fastDPI-сервера в UDR уже будут все свойства абонента - его услуги, полисинг, L2-свойства - MAC-адрес, VLAN и пр. То есть UDR у основного и резервного серверов будут по большому счету согласованы.

Резервирование BRAS L2

Резервирование BRAS в режиме L2 предполагает включение двух СКАТ DPI в один широковещательный L2 домен. Один в режиме Master, другой в режиме Slave. Master осуществляет обработку трафика, авторизацию пользователей через PCRF сервер. Slave не

пропускает трафик через себя, интерфейсы DPDK находятся в режиме ожидания трафика (down). Синхронизация информации об абонентах происходит через PCRF сервер. Slave отслеживает доступность и работоспособность Master, при сбое в работе Slave в автоматическом или ручном режиме активирует (up) DPDK интерфейсы и начинает обрабатывать трафик. Пример включения и прохождения трафика представлено на схеме.

Синхронизация базы данных

За синхронизацию отвечает fastPCRF, настройка описана в разделе Репликация данных авторизации.

Настройка Master CKAT DPI

Настройка Slave CKAT DPI

Описание алгоритма

Концепция резервирования CKAT - MASTER-SLAVE (L2-BRAS):

- 1. MASTER 99% времени работает, может быть отключен или может упасть
- 2. MASTER всегда возвращается и вероломно забирает обработку трафика на себя
- 3. SLAVE 99% времени только принимает репликации с MASTER'а и сохраняет их в UDR
- 4. Есть третья сторона, которая переключает трафик на MASTER или SLAVE, в зависимости от ситуации:
- 4.1. MASTER доступен, SLAVE доступен - трафик переключен на MASTER
- 4.2. MASTER доступен, SLAVE не доступен - трафик переключен на MASTER
- 4.3. MASTER не доступен, SLAVE доступен - трафик переключен на SLAVE
- 4.4. MASTER и SLAVE не доступны - трафик переключен на MASTER

Переключение MASTER→SLAVE:

- 1. Третья сторона детектит пропадание MASTER'а м переключает трафик на SLAVE
- 2. Т.к. на 99% UDR SLAVE'а содержит реплицированные данные, то прерывания почти не заметно физически и логически

Bootstrap MASTER'а (SLAVE активен, обрабатывает трафик):

- 1. На MASTER'е сервисы fastdpi+fastpcrf запущены (после загрузки)
- 2. MASTER определяет что SLAVE жив и актуален
- 3. MASTER останавливает свои fastdpi+fastpcrf
- 4. MASTER бекапит UDR на SLAVE и забирает её к себе
- 5. MASTER запускает свои fastdpi+fastpcrf
- 6. Третья сторона детектит появление MASTER'а и переключает трафик на него

Bootstrap MASTER'а (SLAVE не доступен):

- 1. На MASTER'е сервисы fastdpi+fastpcrf запущены (после загрузки)

- 2. MASTER определяет что SLAVE не доступен, считает что UDR на нем актуальнее чем на неработающем SLAVE, продолжает работу штатно
- 3. Третья сторона детектит появление MASTER'а и переключает трафик на него

Bootstrap SLAVE'а (MASTER активен, обрабатывает трафик):

- 1. На SLAVE'е сервисы fastdpi+fastpcrf запущены (после загрузки)
- 2. SLAVE определяет что MASTER жив и актуален
- 3. SLAVE останавливает свои fastdpi+fastpcrf
- 4. SLAVE бекапит UDR на MASTER'е и забирает её к себе
- 5. SLAVE запускает свои fastdpi+fastpcrf
- 6. Начинает реплицировать данные

Bootstrap SLAVE'а (MASTER не доступен):

- 1. На SLAVE'е сервисы fastdpi+fastpcrf запущены (после загрузки)
- 2. SLAVE определяет что MASTER не доступен, считает что UDR на нем актуальнее чем на неработающем MASTER'е, продолжает работу штатно
- 3. Третья сторона детектит появление SLAVE'а и переключает трафик на него