Содержание

BRAS L2 backup

Backing up of BRAS in L2 mode involves the connecting up of two Stingray Service Gateways in one L2 broadcast domain. One of them in Master mode and the other in Slave one. Master SSG carries out traffic processing along with users authorization through the PCRF server. Slave does not pass traffic through itself, dpdk interfaces are in the traffic standby mode (down). Subscribers information is synchronized through the PCRF server. Slave monitors the availability and performance of the Master and when the last fails, Slave will activate (up) dpdk interfaces and start to process traffic automatically or manually. An example of DPI connection and routes the traffic passes through it are presented in the diagram below.

Database synchronization

FastPCRF is responsible for synchronization, its configuration is described in the section Replication of authorization data.

Configuring the SSG Master mode

Configuring the SSG Slave mode

Algorithm description

Stingray Service Gateway backup concept - MASTER-SLAVE (L2-BRAS):

- 1. MASTER is running 99% of the time, it can be disabled or may fail
- 2. When being recovered MASTER always treacherously proceed to process the traffic
- 3. SLAVE just accepts replications from MASTER and saves them in UDR in 99% of the time
- 4. There is a third party that switches traffic to MASTER or to SLAVE, depending on the current situation:
- 4.1. MASTER is available, SLAVE is available, then the traffic will be switched to MASTER
- 4.2. MASTER is available, SLAVE isn't, then the traffic will be switched to MASTER
- 4.3. MASTER is not available, SLAVE is available, then the traffic will be switched to SLAVE
- 4.4. MASTER and SLAVE are not available, then the traffic will be switched to MASTER

MASTER→ SLAVE toggling:

- 1. The third party detects that MASTER becomes unavailable and switches all the traffic to SLAVE
- 2. Delays when switching are barely perceptible (physically and logically) due to 99% SLAVE's UDR contains replicated data

Bootstrap MASTER'a (SLAVE is active and process traffic):

- 1. MASTER has the fastdpi+fastpcrf services running and enabled (they were started on boot)
- 2. MASTER detects that SLAVE is active and stores relevant data

- 3. MASTER stops its fastdpi + fastpcrf services
- 4. MASTER backups UDR on SLAVE and takes it back
- 5. MASTER starts its fastdpi + fastpcrf
- 6. A third party detects that the MASTER becomes available and switches the traffic to it

Bootstrap MASTER'a (SLAVE is anavailable):

- 1. MASTER has the fastdpi+fastpcrf services running and enabled (they were started on boot)
- 2. MASTER determines that SLAVE is not available, considers that UDR it holds is more relevant than the one located on SLAVE, continues to work normally
- 3. A third party detects that the MASTER becomes available and switches the traffic to it

Bootstrap SLAVE'a (MASTER is active and process traffic):

- 1. SLAVE has the fastdpi+fastpcrf services running and enabled (they were started on boot)
- 2. SLAVE detects that MASTER is active and stores relevant data
- 3. SLAVE stops its fastdpi + fastpcrf services
- 4. SLAVE backups UDR on MASTER and takes it back
- 5. SLAVE starts its fastdpi + fastpcrf
- 6. SLAVE starts to replicate data

Bootstrap SLAVE'a (MASTER is unavailable):

- 1. SLAVE has the fastdpi+fastpcrf services running and enabled (they were started on boot)
- 2. SLAVE detects that MASTER is unavailable, considers that UDR it holds is more relevant than the one located on the currently unavailable MASTER, continues to work normally
- 3. A third party detects that SLAVE becomes available and switches the traffic to it