 **Working Plan of updating OTT service for supporting Multiple IP**

Date: 05/31/2018

1. **Outline**

* Working Information

|  |  |
| --- | --- |
| **Purpose** | Update GSDM, GLB and VOD support Multiple IPs |
| **Date** |  |
| **Estimated Working Time** | 2 days |
| **Supervisor / Operator** |  |

|  |  |
| --- | --- |
| **Target System(s)** | GSDM: 2 servers, LSM Center: 6 servers, VOD Center: 38 servers, VOD Frozen: 2 servers. |
| **Component(s)** | CiGSDMServer, CiGLBServer |
| **Component version**  **(optional)** |  |
| **Expected effect** | Service will not available during restart |

1. **Working Description**

* Working order should be following below steps:

1. Update VODServer config: update all or some of VOD servers.
2. Update GLBServer binary+ GLBServer config: update all servers
3. Update GSDMServer binary+GSDMServer config: update all servers.

* Note: Viettel can update some of GLB and some of GSDM but the 2 IPs function will be useless.

1. **Procedure**

|  |
| --- |
| **Update VOD configuration** |
| ##########Backup config file###########  **[root@VSM]#** cd /castis/bin/CiVODServer/  **[root@VSM]#** cp CiVODServer.cfg CiVODServer.cfg.`date +%F`  ##########Comment the green line and add red line###########  **[root@VSM]#** vim CiVODServer.cfg  # Listen & Connect  IP\_Address=${bond1\_ip}  Control\_IP\_Address=${bondo\_IP}  #MarkupListen\_Multi\_IPPort=18080  MarkupListen\_Multi\_IPPort=${bond1\_IP}:18080,**${bond1\_VIP}:18080**  SCPServer\_Listen\_IPPort=886  PingPServer\_Listen\_IPPort=889  **[root@VSM]#** killall CiVODServer |

|  |
| --- |
| **Update GLB Server** |
| ##########Add VIP monitoring script for LSM#########  **[root@VSM]#** mkdir /castis/bin/monitor\_vip\_script  **[root@VSM]#** mv /home/vt\_admin/monitor\_vip.sh /castis/bin/monitor\_vip\_script  **Detail of script:**  #!/bin/bash  ## VARIABLES  # Monitor NIC  monitorBond=”bond1:1”  monitorIp=**${bond1:1 IP}**  # VIP NIC  bondName="bond1:2"  ip=**${bond1:2 IP}**  netMask=**${bond1 netmask}**  # Run every n sec  runInterval="3"  logFolder="/castis/log/monitor\_vip\_log"  mkdir -p ${logFolder}  ## SCRIPT  while true; do  nicUp=`ifconfig ${monitorBond} | grep ${monitorIp}`  vipUp=`ifconfig ${bondName} | grep ${ip}`  if [[ ${#nicUp} -gt 0 ]]; then  if [[ ${#vipUp} -eq 0 ]]; then  ifconfig ${bondName} ${ip} netmask ${netMask} 2>&1 >/dev/null  checkVIP=`ifconfig ${bondName} | grep ${ip}`  # Check if VIP successfully created.  if [[ ${#checkVIP} -gt 0 ]]; then  echo "[`date +"%Y-%m-%d-%H:%M:%S"`] VIP ${bondName} ${ip} is successfully created !"  else  echo "[`date +"%Y-%m-%d-%H:%M:%S"`] VIP ${bondName} ${ip} FAIL to create !"  fi  fi  else  if [[ ${#vipUp} -gt 0 ]]; then  ifconfig ${bondName} down 2>&1 >/dev/null  echo "[`date +"%Y-%m-%d-%H:%M:%S"`] Service down on this server ==> ${bondName} shutdown !"  fi  fi  sleep ${runInterval}  done >> ${logFolder}/$(date +%d-%m-%Y)\_monitor\_vip.log  **[root@VSM]#** mv /home/vt\_admin/startInstance.sh /castis/bin/monitor\_vip\_script  **Detail of Script:**  #!/bin/bash  ## VARIABLES  workingFolder="/castis/bin/monitor\_vip\_script"  logFolder="/castis/log/monitor\_vip\_log"  programPath="${workingFolder}/monitor\_vip.sh"  mkdir -p ${logFolder}  mkdir -p ${workingFolder}  ## SCRIPT CHECK IF PROCESS RUNNING  if [[ -f ${programPath} ]]; then  if [[ ! -f ${workingFolder}/.monitor\_vip.pid ]]; then  ${programPath} &  echo $! > ${workingFolder}/.monitor\_vip.pid  echo "[`date +"%Y-%m-%d-%H:%M:%S"`] PID file not found. Start processs!" >> ${logFolder}/$(date +%d-%m-%Y)\_monitor\_vip.log  else  processId=`cat ${workingFolder}/.monitor\_vip.pid`  checkRunning=`ps -p ${processId} | wc -l`  if [[ ${checkRunning} -eq 1 ]]; then  echo "[`date +"%Y-%m-%d-%H:%M:%S"`] VIP monitoring not running. Start processs!" >> ${logFolder}/$(date +%d-%m-%Y)\_monitor\_vip.log  ${programPath} &  echo $! > ${workingFolder}/.monitor\_vip.pid  fi  fi  else  echo "[`date +"%Y-%m-%d-%H:%M:%S"`] Monitor VIP program NOT FOUND. Exit !" >> ${logFolder}/$(date +%d-%m-%Y)\_monitor\_vip.log  fi  **[root@VSM]#** chmod +x /castis/bin/monitor\_vip\_script/\*.sh  **[root@VSM]#** echo “\* \* \* \* \* root /castis/bin/monitor\_vip\_script/startInstance.sh” >> /etc/crontab  ##########Copy binary###########  **[root@VSM]#**  cp /home/vt\_admin/CiGLBServer-2.4.7.QR1-el6.x86\_64 /castis/bin/CiGLBServer/  ##########Modify config of GLBServer. if VODServer use old config, ${VOD\_bond1\_IP} and ${VOD\_bond1:1\_VIP} should be same when modify proxy.cfg ###########  **[root@VSM]#** cd /castis/bin/CiGLBServer/  **[root@VSM]#** vim proxy.cfg  VERSION 1.0  #VOD\_IP\_Change ${ServiceIP} ${ChangeIP}  VOD1\_IP\_Change ${VOD1\_bond1\_IP} **${VOD1\_bond1:1\_VIP}**  VOD2\_IP\_Change ${VOD2\_bond1\_IP} **${VOD2\_bond1:1\_VIP}**  **......**  **[root@VSM]#** cp CiGLBServer.cfg CiGLBServer.cfg.`date +%F`  ##########Add new config ###########  **[root@VSM]#** vim CiGLBServer.cfg  #New Config for Multiple IP GLB  UseGLBProxy=1  NumberOfMarkupProxy=1  MarkupProxyConfigFileName0=/castis/bin/CiGLBServer/proxy.cfg  MarkupProxyListen\_Multi\_IPPort0=**${LSM\_bond1:2\_VIP}:**18080  ##########Make config###########  **[root@VSM]#** ln -sf CiGLBServer-2.4.7.QR1-el6.x86\_64 CiGLBServer  **[root@VSM]#** killall CiGLBServer |

|  |
| --- |
| **Update GSDM-OTT** |
| ##########Copy binary###########  **[root@VSM]#**  cp /home/vt\_admin/CiGSDMServer-1.10.3-jar-with-dependencies.jar /castis/bin/CiGSDMServer/  ##########Modify config of GLBServer. if GLBServer use old config, ${VOD\_bond1\_IP} and ${VOD\_bond1:1\_VIP} should be same when modify proxy.cfg ###########  **[root@VSM]#** cd /castis/bin/CiGSDMServer/  ##########create 3party file###########  **[root@VSM]#** vim 3rdParty.cfg  ViettelOnMe\_OTT=regionInfo\_Onme.cfg  ViettelOnMe\_OTT\_Android=regionInfo\_Onme.cfg  ViettelOnMe\_OTT\_iOS=regionInfo\_Onme.cfg  ViettelTV\_OTT=regionInfo\_Onme.cfg  ##########create regionInfo\_Onme and update new VIP of LSM###########  **[root@VSM]#** cp regionInfor.cfg regionInfo\_Onme.cfg  **[root@VSM]#** vim regionInfo\_Onme.cfg  The\_Number\_Of\_Region=65  glbAddress\_0=A076,${GLB\_VIP}:18080,${GLB\_VIP}:18080  glbAddress\_1=B056, ,${GLB\_VIP}:18080,${GLB\_VIP}:18080  ……  **[root@VSM]#** cp gsdm.cfg gsdm.cfg`date +%F`  ##########Add new config ###########  **[root@VSM]#** vim gsdm.cfg  # use3rdParty config  **use3rdPartyConfig=1**  ##########Make config###########  **[root@VSM]#** ln -sf CiGSDMServer-1.10.3-jar-with-dependencies.jar CiGSDMServer  **[root@VSM]#** service gsdm restart |

1. **Testcase**

***4.1 Check full flow GSDM , GLB***

- Play Content on ViettelTV app -> capture packet, check log -> Use new GLB and VOD VIP IP.

- Play Content on OnMe app -> capture packet, check log -> Use normal GLB and VOD IP.

***4.2 Check Dynamic API update on GSDM:***

- Update 3rdParty.cfg config using PUT API:

+ Method: PUT

+ API: /api/v1/3rdParty

+ Header: Content-Type: application/json, Accept: application/json

+ Body:

{

"thirdPartyConfig": [{

"serviceProvider": "OnMe",

"configName": "regionInfo\_onme.cfg"

}]

}

+ Response:

{

"resultCode": 200,

"message": "Update 3rdParty.cfg info successfully!"

}

***4.3 Check Dynamic API update regionId.cfg:***

- Update regionInfo.cfg config using PUT API:

+ Method: PUT

+ API: /api/v1/region?serviceProvider=${value}

+ Parameter:

* **serviceProvider** is optional
* If serviceProvider not in request API -> update regionInfo.cfg
* If serviceProvider in request API -> update the regionInfo config got mapped with serviceProvider in 3rdParty.cfg

Example:

**# cat 3rdParty.cfg**

OnMe\_OTT=regionInfo\_onme.cfg

# curl -XPUT <http://127.0.0.1:18081/api/v1/region?serviceProvider=OnMe_OTT> [-d@"region.json](mailto:-d@%22region.json)"

=> This request will update **regionInfo\_onme.cfg** file.

+ Header: Content-Type: application/json, Accept: application/json

+ Body:

{

"regionConfig": [{

"regionId": "A076",

"glbAddress": ["10.60.70.199:18080", "10.60.70.199:18080"]

}, {

"regionId": "B056",

"glbAddress": ["10.60.70.199:18080", "10.60.70.199:18080"]

}, {

"regionId": "B062",

"glbAddress": ["10.60.70.199:18080", "10.60.70.199:18080"]

}, {

"regionId": "B075",

"glbAddress": ["10.60.70.199:18080", "10.60.70.199:18080"]

}]

}

+ Response:

{

"resultCode": 200,

"message": "Update Region table info successfully!"

}

***4.4 Check fail over of LSM VIP IP:***

- If **bond1:1** up on 1 server -> Up **bond1:2**

- If **bond1:1** down on 1 server -> Down **bond1:2**

1. **Rollback**

|  |
| --- |
| **Restore config** |
| ##########Restore VOD###########  **[root@VSM]#** cd /castis/bin/CiVODServer  **[root@VSM]#** ln -sf CiVODServer.cfg.`date +%F` CiVODServer.cfg  **[root@VSM]#** killall CiVODServer  ##########Restore CiGSDM###########  **[root@VSM]#** cd /castis/bin/CiGSDMServer/  **[root@VSM]#** ln -sf gsdm.cfg`date +%F` gsdm.cfg  **[root@VSM]#** ln -sf CiGSDMServer-1.9.13.QR4.jar CiGSDMServer  **[root@VSM]#** service gsdm restart  ##########Restore GLB##########  **[root@VSM]#** cd /castis/bin/CiGLBServer  **[root@VSM]#** ln -sf CiGLBServer-2.4.6.QR1-el6.x86\_64 CiGLBServer  **[root@VSM]#** ln -sf CiGLBServer.cfg.`date +%F` CiGLBServer.cfg  **[root@VSM]#** killall CiGLBServer  **[root@VSM]#** vim /etc/crontab  Comment the last added line:  ….  **# \* \* \* \* \* root /castis/bin/monitor\_vip\_script/startInstance.sh**  **[root@VSM]#** killall monitor\_vip.sh |