EEM – Event Manager Applets

EEM Applets bieten Möglichkeiten zur "Programmierung" in der Form
→ wenn sich etwas ereignet: **event**
→ dann führe etwas aus: **action**

Dabei können auch Variablen, String Funktionen, IF Abfragen, WHILE Schleifen, und weiter Funktionalitäten verwendet werden – hier NUR wichtigste Funktionen.

Syntax

(config)# **event manager applet** *applet-name*
(config-applet)# **event** *event*
(config-applet)# **action** *label action*

action *label*
label ist frei wählbar – best practise: **strukturierte Nummerierung** verwenden
ACHTUNG: Reihenfolge ist alphanumerisch (also z.B. 10 vor 2)

```
!
event manager applet TEST
  event ...
  action 1.0 …
  action 1.1 …
  action 1.2 ...
  action 2.0 …
  action 2.1 …
!
```

show run | section event manager
show event manager history events
debug event manager [**action** *applet-name* | **all**]

Übersicht wichtiger Variablen, event und action Kommandos auf den folgenden Seiten.

© 2023 - r.pohlmann@ping.de

Variablen

Zugriff mit ***$varname***

Vordefinierte Variablen	
_exit_status	EEM „Ergebnis" - soll CLI ausgeführt werden – z.B. bei IF Abfragen • 1 = ok (program runs) • 0 = not ok
_cli_result	Ausgaben von **action** *label* **cli command**
_string_result	Ergebnisse von **action** *label* **string** Funktionen
_regexp_result	Ergebinisse von **action** label **regexp** Funktion
Eigene Variablen .. siehe auch action's	
Dauerhaft	**EEM Environment** (config)# **event manager environment** *varname* "*wert*" ! event manager environment EGAL "wurst" event manager applet TESTVAR event none action 1.0 puts $EGAL ! # event manager run TESTVAR # show event manager environment
Temporär	**EEM Applet action set** (config-applet)# **action** *label* **set** *varname* "*wert*" ! event manger applet TESTVAR2 event none action 1.0 set $AUCHEGAL "auch wurst" action 1.1 puts $AUCHEGAL ! # event manager run TESTVAR2

© 2023 - r.pohlmann@ping.de

event's - Gebräuchliche Auslöser (Trigger)

<table>
<tr><td>none
kein Trigger</td><td>falls das Applet manuell ausgeführt werden soll
event manager run applet-name</td></tr>
<tr><td>cli pattern
Trigger: CLI Eingabe</td><td>event cli pattern reg-exp
[sync { no | yes }] [skip { no | yes }]
[occurs anz] [period sec]

sync no skip no (default)
• EEM Applet wird ausgeführt, CLI Command wird ausgeführt
sync no skip yes
• EEM Applet wird ausgeführt, CLI Command wird NICHT ausgeführt
sync yes
• %EEM: Argument "skip" should not exist when sync is "yes"
• EEM Applet wird ZUERST ausgeführt, CLI Command DANACH falls:
◦ skip no (default)
◦ _exit_status = 1</td></tr>
<tr><td>syslog pattern
Trigger: LOG Meldung</td><td>event syslog pattern reg-exp
[occurs anz] [period sec] [priority-level { all | <0-7>}]</td></tr>
<tr><td>track
Trigger: Track State</td><td>event track track-ID state { up | down | any }</td></tr>
<tr><td rowspan="2">timer
Trigger: Time</td><td>event timer { absolute time sec | countdown time sec.msec |
cron cron-entry | watchdog sec } [name timer-name]

Einmal ausführen in 5 h
event timer absolute time 18000 name five-hours
Einmal ausführen in 6 m und 6 sec
event timer countdown time 360.006 name six-minutes
Alle 5 h ausführen
event timer watchdog time 18000</td></tr>
<tr><td>cron-entry
• minute <0-59>
• hour<0-23>
• day-of-month <1-31>
• month <1-12>
• day-of-week <0-6>; 0 = Sunday
• egal <*>

Um 01:01 am 01 Januar jedes Jahr
event timer cron cron-entry 1 1 1 1 * name Jan1
Um 12:00 Mo - Fr
event timer cron cron-entry 0 12 * * 1-5 name MonFri</td></tr>
</table>

© 2023 - r.pohlmann@ping.de

action's - elementar

puts	schreibt Zeichenfolgen auf das aktive Terminal **action** *label* **puts** [**nonewline**] "*string*"
syslog msg	erzeugt Log-Meldung **action** *label* **syslog** [**priority** *level*] **msg** "*string*" [**facility** *facility*]
cli command	führt CLI Kommando aus (immer mit enable beginnen) **action** *label* **cli command** "*command*"

set	Setzt – temporäre - Variablen **action** *label* **set** *varname* "*sting*"
gets	Liest Eingabe vom aktiven Terminal → Eingabe-String in $*varname* **action** *label* **gets** *varname*

EEM Applet – Temporäre Variablen setzen	Ausgabe
! event manager applet SETVARS event none action 1.0 set myS1 "wurst" action 1.1 set myS2 "auch wurst" action 2.0 puts "$myS1" action 2.1 puts "$myS2" ! # event manager run SETVARS	 wurst auch wurst

EEM Applet – Manuelle Eingabe	Ausgabe
! event manager applet INPUT event none action 1.0 puts nonewline "Eingabe: " action 1.1 gets eingabe action 1.2 puts "$eingabe" ! # event manager run INPUT	 Eingabe: alles wurst alles wurst

© 2023 - r.pohlmann@ping.de

action's advanced

Guidelines

- string → Zeichenfolge
- beinhaltet der string Leerzeichen, muss er in Anführungszeichen gesetzt werden
- string kann auch der Textinhalt einer beliebigen Variable sein
  .. werden bei der Konfiguration auf der CLI automatisch in Anführungszeichen gesetzt
- string Zeichen-Indexierung beginnt mit 0

string	String Funktionen **action** *label* **string** *string-function* Ergebnis der string Funkton in **_string_result** (0|1|string) • **string toupper** *"string"* [*start-index end-index*] → _string_result = Zeichen in Grossbuchstaben • **string tolower** "*string*" [*start-index end-index*] → _string_result = Zeichen in Grossbuchstaben • **string replace** "*string*" *start-index end-index* "*string*" → ersetzt Zeichenfolge: _string_result = neuer string • **string equal** "*string*" "*string*" → wenn gleich: _string_result = "1" • **string match** *"string-pattern"* "*string*" → wenn gleich: _string_result = "1" → pattern NUR: ***** (multiple char) | ? (single char) .. aber wie HINWEIS: für mehr Möglichkeiten .. **action regexp**
regexp	Überprüft, ob pattern in String oder Variable enthalten ist **action** *label* **regexp** "*string-pattern" "string"* [*match-varname*] [*sub-match-varname*] [..] Ergebnis der Funktion in **_regexp_result** (0|1) • *match-varname* der gefundene string kann in eigener Variable gesichert werden • *sub-match-varname* substrings können in insegesamt 3 zusätzlchen Variablen gesichert werden
if, else, end	IF Abfragen **action** *label* **if** "string" { **eq**|**ne**|**ge**|**gt**|**le**|**lt** } "string" **action** *label* **else** **action** *label* **end**
Es gibt noch weitere - hier nicht näher erläuterte - actions, wie z.B. Variablen-Funktionen oder Schleifen (**while** oder **foreach**) ..	

© 2023 - r.pohlmann@ping.de

String Funktionen	Ausgabe
! event manager applet SF event none action 1.0 set myS1 "Wurst" action 1.1 set myS2 "Auch Wurst" action 1.2 puts "myS1: $myS1" action 1.3 puts "myS2: $myS2" !	 myS1: Wurst myS2: Auch Wurst
action 2.0 string toupper "$myS1" action 2.1 puts "_string_result = $_string_result" action 2.2 string tolower "$myS2" action 2.3 puts "_string_result = $_string_result" action 2.4 puts " " action 2.5 puts "myS1 not changed: $myS1" action 2.6 puts "myS2 not changed: $myS2" !	 _string_result = WURST _string_result = auch wurst myS1 not changed: Wurst myS2 not changed: Auch Wurst
action 3.0 string replace "$myS2" 0 3 "ebenso" action 3.1 puts "_string_result = $_string_result" action 3.2 puts "myS2 not changed: $myS2" !	 _string_result = ebenso Wurst myS2 not changed: Auch Wurst
action 4.0 string equal "$myS1" "Wurst" action 4.1 puts "_string_result = $_string_result" action 4.2 string equal "Wurst" "$myS1" action 4.3 puts "_string_result = $_string_result" action 4.4 string equal "Wurst" "$myS2" action 4.5 puts "_string_result = $_string_result" !	 _string_result = 1 _string_result = 1 _string_result = 0
action 5.0 string match "Wurst" "$myS1" action 5.1 puts "_string_result = $_string_result" action 5.2 string match "Wurst" "$myS2" action 5.3 puts "_string_result = $_string_result" action 5.4 string match "* Wurst" "$myS2" action 5.5 puts "_string_result = $_string_result" action 5.6 string match "*uch*ur*" "$myS2" action 5.7 puts "_string_result = $_string_result" !	 _string_result = 1 _string_result = 0 _string_result = 1 _string_result = 1
exit ! # event manager run SF	

© 2023 - r.pohlmann@ping.de

RegExp Funktion und **IF Abfragen**	Erläuterung

!
event manager applet CHECKROUTE
 event none
 action 1.00 cli command "enable"
 action 1.01 cli command "sh ip route | include 10.0.0.2"
!
 action 2.00 regexp ".* 10\.0\.0\.2 .*" $_cli_result
!
 action 3.00 if $_regexp_result eq "1"
 action 3.01 puts "Route to 10.0.0.2 in Routing Table .."
 action 3.02 regexp "O .*" $_cli_result
!
 action 3.03 if $_regexp_result eq "1"
 action 3.04 puts ".. learned via OSPF"
 action 3.05 else
 action 3.06 puts ".. learned via UNKNOWN"
 action 3.07 end
 action 3.08 else
 action 3.09 puts "Route to 10.0.0.2 NOT in Routing Table"
 action 3.10 end
!
 exit
!
 	
# event manager run CHECKROUTE 	
	
	Ausgabe in _cli_result
	
	.. ist 10.0.0.2 in string enthalten
	.. ja → _regexp_result = 1
	.. if (wenn) ja
	  .. Ausgabe
	  .. ist O in string enthalten
	  .. ja → _regexp_result = 1
	  .. wenn ja
	  .. Ausgabe
	  .. ansonsten
	  .. Ausgabe
	  .. ende if
	.. else (ansonsten)
	  .. Ausgabe
	.. ende if

Ausgabe des Applets, wenn Route als OSPF Route in Routing Tabelle existiert

R1# **event manager run CHECKROUTE**
Route to 10.0.0.2 in Routing Table ..
.. learned via OSPF

Ausgabe des Applets, wenn Route in Routing Tabelle existiert, aber nicht von OSPF kommt

R1# **event manager run CHECKROUTE**
Route to 10.0.0.2 in Routing Table ..
.. learned via UNKNOWN

Ausgabe des Applets, wenn Route als OSPF Route in Routing Tabelle existiert

R1# **event manager run CHECKROUTE**
Route to 10.0.0.2 NOT in Routing Table

© 2023 - r.pohlmann@ping.de

EEM Applet Beispiele

event track → action cli, puts	Erläuterung
! track 1 interface loopback 0 line-protocol ! event manager applet LONOSHUT event track 1 state down action 1.0 cli command "enable" action 2.0 cli command "conf t" action 3.0 cli command "interface loopback 0" action 4.0 cli command "no shutdown" action 5.0 cli command "end" action 6.0 puts "IF Loopback 0 activated" !	 tracking auf IF Loopback 0 Wenn tracking state down .. IF Loopback 0 sofort wieder aktivieren .. Nachricht auf Terminal

event cli pattern → action puts, gets, string, if, set	Erläuterung HINT: statt **string match** hätte hier auch **string equal** verwendet werden können
! event manager applet SHSTART event cli pattern "show startup-config" sync yes action 1.0 puts "HINWEIS: startup-config aktuell ?" action 1.1 puts nonewline "Continue [Y/N]: " action 1.2 gets eingabe action 2.0 string toupper "$eingabe" action 2.1 string match "$_string_result" "Y" action 3.0 if $_string_result eq 1 action 3.1 set _exit_status "1" action 3.2 end !	 CLI Kdo wird noch nicht ausgeführt (sync yes) Eingabe wird ausgelesen .. konveriert in UPPER case .. überprüft ob Ergebnis = Y .. wenn ja _string_result = 1 .. _exit_status = 1 Bei _exit_status = 1 → CLI Kdo wird ausgeführt .. sonst (_exit_status = 0) nicht

event none → action cli	Erläuterung
! event manager applet ADMPING event none action 1.0 cli command "enable" action 1.1 cli command "tclsh flash:/admping.tcl" ! # event manager run ADMPING	 Applet wird manuell gestartet .. führt tcl Script in tcl Shell aus START des Applets
	# flash:/admping.tcl *foreach address {* *192.168.1.1* *10.0.0.1* *10.0.0.2* *} { ping $address* *}*

© 2023 - r.pohlmann@ping.de

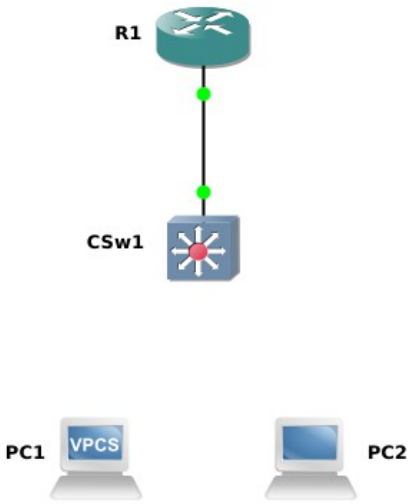
event cli pattern → action cli, syslog	Erläuterung
! event manager environment filename R1.cfg event manager environment tftpserver tftp://10.1.1.100/ ! event manager applet BACKUP-CONFIG event cli pattern "write mem.*" sync yes action 1.0 cli command "enable" action 2.0 cli command "configure terminal" action 3.0 cli command "file prompt quiet" action 4.0 cli command "end" action 5.0 cli command "copy start $tftpserver$filename" action 6.0 cli command "configure terminal action 7.0 cli command "no file prompt quiet" action 8.0 syslog priority 5 msg "startup-config -> TFTP" !	 Variablen setzen Bei Sicherung der Config .. keine interaktive Abfragen .. Backup an TFTP .. wieder interaktive Abfragen .. SYSLOG Level 5 Nachricht

event timer cron → action cli	Ausgabe
! event manager applet LOGGING event timer cron name LOGG cron entry 0 21 * * 0-4 action 1.0 set dst "tftp://10.1.1.1/R1.log" action 1.0 cli command "enable" action 2.0 cli command "show logging | redirect $dst" !	 täglich 21:00 Uhr, So - Do .. Variable setzen .. Log-Buffer Inhalt an TFTP

event timer watchdog → action cli, regexp, if, syslog	Ausgabe
! event manager applet VR event timer watchdog time 3600 action 1.0 cli command "enable" action 1.2 cli command "sh ip route | include 10.0.0.2" action 2.0 regexp "O.*" $_cli_result action 3.0 if $_regexp_result eq "1" action 3.1 syslog msg "Route existent" action 3.2 syslog msg $_cli_result action 3.3 else action 3.4 syslog msg "Route NICHT existent" action 3.5 end !	 jede Stunde .. ist Route in Routing Table .. ist Route OSPF Route .. dann .. sonst

© 2023 - r.pohlmann@ping.de

EEM Lösung für ein besonderes DHCP Server Binding Table Problem

Scenario	
	**Anschluss PCs** → temporär → ausschliesslich abwechselnd, ausdrücklich nicht gemeinsam → wahlweise an Port gi1/0 oder Port gi1/1 → IP Adresse und Default GW via DHCP **CSW1: DHCP Server** → stellt im Pool NUR genau eine IP Adresse aus dem Netzwerk 192.168.1.0/24 für die Vergabe zur Verfügung ! ip dhcp excluded-address 192.168.1.2 192.168.1.254 ! ip dhcp pool VL1 network 192.168.1.0 255.255.255.0 default-router 192.168.1.254 !

Problem

1. Nach erstmaligem Anschluss eines PC's, erhält dieser die einzige IP Adresse aus dem Pool
   CSw1 → Eintrag in die DHCP Binding Table
2. .. dann trennt der PC die temporäre Kabelverbindung wieder
   CSw1 → da der DHCP Server keine DHCP Release Nachricht vom PC erhalten hat (das Kabel wurde entfernt) verbleibt der Eintrag in der DHCP Binding Table
3. Nach erneutem Anschluss eines PC, erhält dieser KEINE IP Adresse mehr vom DHCP Server, da die einzige zu vergebende IP bereits vergeben ist.

Problemlösung mit EEM - event syslog pattern, action cli command, puts, syslog msg

wenn die Ethernet Verbindung zu PC1 oder PC2 auf gi1/0 oder gi1/1 ausfällt

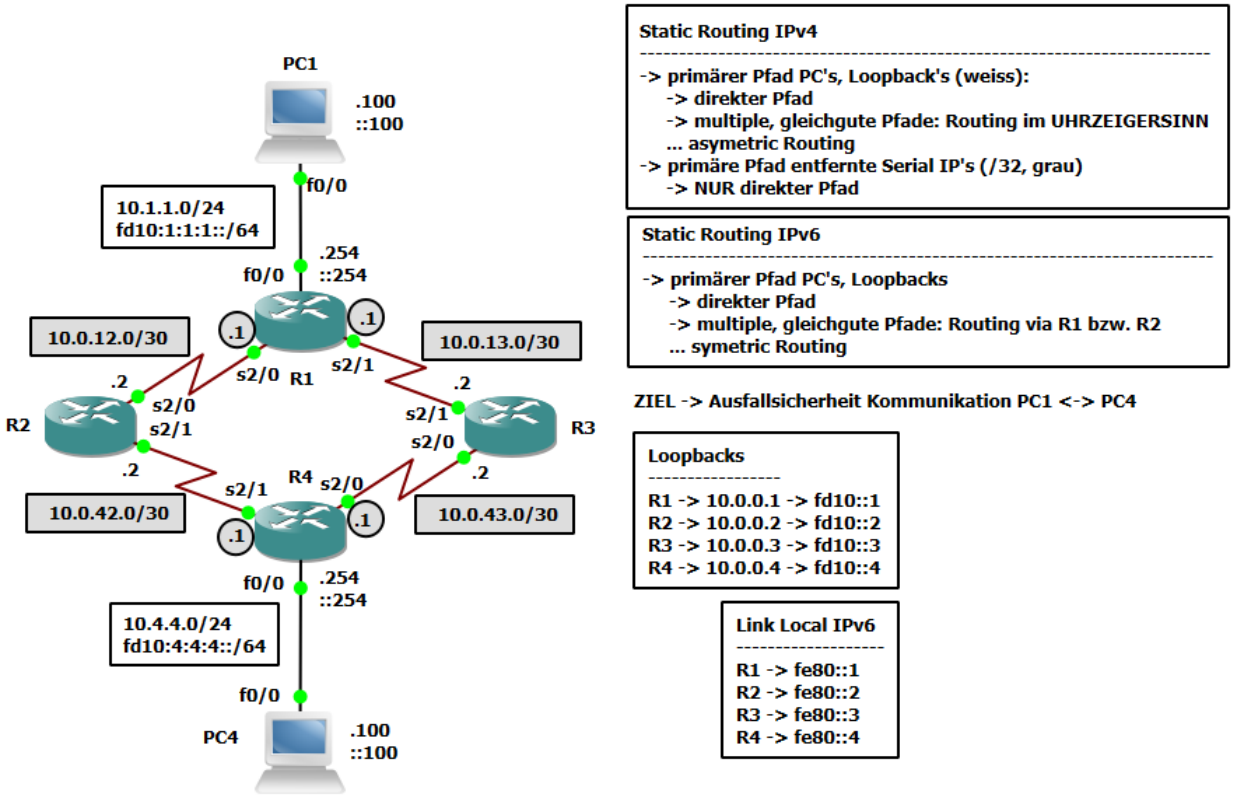
1. schreibe eine Ereignis- und Aktions-Nachricht auf das aktive Terminal
2. lösche die DHCP Binding Table
3. erzeuge eine SYSLOG Nachricht über die Aktion

```
!
event manager applet DHCP_RELEASE
 event syslog pattern ".*LINEPROTO-5.* GigabitEthernet1/[01]+.* down"
 action 1.0 puts "Ethernet Connection to CLIENT LOST"
 action 1.1 puts "Clearing DHCP Binding Table"
 action 2.0 cli command "enable"
 action 2.1 cli command "clear ip dhcp binding *"
 action 2.2 cli command "end"
 action 3.0 syslog msg "IP DHCP Binding Table cleared"
 exit
!
```


© 2023 - r.pohlmann@ping.de

IPv6 Ausfallsicherheit mit IPSLA Tracking, PBR und EEM

.. für eine Ausfallsicherheit der Kommunikation zwischen den PC's



IPv4 → durch **IP SLA Tracking** und **PBR** (kein EEM notwendig)
Innerhalb der Route Map wird mit **verify-availiability** der Track für den primaren next-hop eingebunden. Ist der Track Status "down" wird der zweite next-hop Eintrag verwendet.

R1	R4

!
ip sla 4
 icmp-echo 10.0.43.1
  frequency 5
ip sla schedule 4 start now life forever
track 4 ip sla 4 reachability
!
ip access-list extended IPv4-PBR
 permit ip 10.1.1.0 0.0.0.255 10.4.4.0 0.0.0.255
!
route-map IPv4-PBR permit 10
 match ip address IPv4-PBR
 set ip next-hop verify-availability 10.0.13.2 1 track 4
 set ip next-hop 10.0.12.2
!
int fa 0/0
 ip policy route-map IPv4-PBR
!
ip route 10.4.4.0 255.255.255.0 10.0.13.2
ip route 10.0.43.1 255.255.255.255 10.0.12.2
!
	!
	ip sla 1
	 icmp-echo 10.0.12.1
	  frequency 5
	ip sla schedule 1 start now life forever
	track 1 ip sla 1 reachability
	!
	ip access-list extended IPv4-PBR
	 permit ip 10.4.4.0 0.0.0.255 10.1.1.0 0.0.0.255
	!
	route-map IPv4-PBR permit 10
	 match ip address IPv4-PBR
	 set ip next-hop verify-availability 10.0.42.2 1 track 1
	 set ip next-hop 10.0.43.2
	!
	int fa 0/0
	 ip policy route-map IPv4-PBR
	!
	ip route 10.1.1.0 255.255.255.0 10.0.42.2
	ip route 10.0.12.1 255.255.255.0 10.0.42.2
	!

ACHTUNG: Das Kommando **verify-availiability** NICHT für IPv6 zur Verfügung.

© 2023 - r.pohlmann@ping.de

IPv6 → durch **IP SLA Tracking** und **PBR** und **EEM**

Das EEM Applet aktiviert PBR am incoming interface wenn der IP SLA Track den Status down zurückgibt (event track → action cli command, syslog msg)

R1	R4
! ip sla 64 icmp-echo FD10::4 source-interface Loopback0 frequency 5 ip sla schedule 64 life forever start-time now track 64 ip sla 64 reachability ! ipv6 access-list PBR permit ipv6 FD10:1:1:1::/64 FD10:4:4:4::/64 ! route-map PBR permit 10 match ipv6 address PBR set interface Serial2/1 set ipv6 next-hop FE80::3 ! event manager applet PBR event track 64 state down action 1.0 cli command "enable" action 1.1 cli command "conf t" action 1.2 cli command "interface fa0/0" action 1.3 cli command "ipv6 policy route-map PBR" action 1.4 cli command "end" action 2.0 puts "PBR Routing -> enabled" action 2.1 syslog msg "PBR Routing -> enabled" ! no event manager applet NO-PBR event track 64 state up action 1.0 cli command "enable" action 1.1 cli command "conf t" action 1.2 cli command "interface fa0/0" action 1.3 cli command "no ipv6 policy route-map PBR" action 1.4 cli command "end" action 2.0 puts "PBR Routing -> disabled" action 2.1 syslog msg "PBR Routing -> disabled" ! ipv6 route fd10:4:4:4::/64 serial 2/0 fe80::2 ipv6 route fd10::4 serial 2/0 fe80::2 !	! ip sla 61 icmp-echo FD10::1 source-interface Loopback0 frequency 5 ip sla schedule 61 life forever start-time now track 61 ip sla 61 reachability ! ipv6 access-list PBR permit ipv6 FD10:4:4:4::/64 FD10:1:1:1::/64 ! route-map PBR permit 10 match ipv6 address PBR set interface Serial2/0 set ipv6 next-hop FE80::3 ! event manager applet PBR event track 61 state down action 1.0 cli command "enable" action 1.1 cli command "conf t" action 1.2 cli command "interface fa0/0" action 1.3 cli command "ipv6 policy route-map PBR" action 1.4 cli command "end" action 2.0 puts "PBR Routing -> enabled" action 2.1 syslog msg "PBR Routing -> enabled" ! event manager applet NO-PBR event track 61 state up action 1.0 cli command "enable" action 1.1 cli command "conf t" action 1.2 cli command "interface fa0/0" action 1.3 cli command "no ipv6 policy route-map PBR" action 1.4 cli command "end" action 2.0 puts "PBR Routing -> disabled" action 2.1 syslog msg "PBR Routing -> disabled" ! ipv6 route fd10:1:1:1::/64 serial 2/1 fe80::2 ipv6 route fd10::1 serial 2/1 fe80::2 !

© 2023 - r.pohlmann@ping.de

IPv6 First Hop Redundancy mit Router Preference und EEM

.. mit Tracking: geht natürlich auch mit HSRPv2 oder VRRPv3 ;).

Router R1 und R2 haben über das Interface fa 0/0 jeweils eine Verbindung in das IPv6 Netzwerk fd88:8:8:8::/64.

R1 ist primäres Default GW → **ipv6 nd router-preference high**.

Tracking mit EEM
event track → action cli command, puts

.. falls die "externe" Verbindung (über fa 1/0) ausfällt, soll R2 als primäres Default GW verwendet werden:
→ **router-preference** set to **low**

R1	R2
! track 1 int fa 1/0 line-protocol ! int fa 0/0 descr -> LAN ipv6 address fe80:88::1 link-local ipv6 address fd88:8:8:8::1/64 ipv6 nd router-preference high ! int ser 0/0 descr -> EXTERN ipv6 address fe80:88::1 link-local ipv6 address 2000:ba:ff:1::1/64 ! event manager applet ND-RPrefLOW event track 1 state down action 1.0 cli command "enable" action 1.1 cli command "conf t" action 1.2 cli command "interface fastethernet0/0" action 1.3 cli command "ipv6 nd router-preference low" action 1.4 cli command "end" action 1.5 puts "ND Preference set to LOW" event manager applet ND-RPrefHIGH event track 1 state up action 1.0 cli command "enable" action 1.1 cli command "conf t" action 1.2 cli command "interface fastethernet0/0" action 1.3 cli command "ipv6 nd router-preference high" action 1.4 cli command "end" action 1.5 puts "ND Preference set to HIGH" !	! int fa 0/0 descr -> LAN ipv6 address fe80:88::2 link-local ipv6 address fd88:8:8:8::2/64 ! int ser 0/0 descr -> EXTERN ipv6 address fe80:88::2 link-local ipv6 address 2000:ba:ff:2::2/64 !

© 2023 - r.pohlmann@ping.de