

Incinga-Konzept

Hier ein Entwurf für das neue Icinga-Monitoring.

Gateway-Monitoring:

- Vom Icinga-Server aus überwachen:
 - Zunächst IP des Blechs testen. Falls nicht pingbar Email an Kontaktadresse versenden
 - Pingtest Hoster-IP Falls nicht pingbar: Auf Blech einloggen und mit virsh destroy \$vname && virsh start \$vname neu starten
 - Pro Host müsste es separat setzbar sein, was im Falle eines Ausfalls zu tun ist.
 - Teilweise ist Login auf Blech möglich
 - Prüfen wie man VMware-Kisten automatisiert töten kann
 - Bei gespendeten VMs muss teilweise einfach eine Email versendet werden. SMTP-Login dafür kann bereit gestellt werden
 - Falls gesetzt Nat-IP pingen (von außen)
- Vom Gateway selbst per nripe:
 - Pro Domäne die a.b.c.2, a.b.c.3, a.b.c.11 und a.b.c.12 pingen, falls zwei nicht pingbar Wahrscheinlich hängt das Batman Neustart
 - Gleiches für V6: <Präfix>:\$domäne::2, ::3, ::11, ::12 pingen Falls zwei nicht pingbar Neustart
 - (Falls die VM dabei hängen bleibt, greift die erste Regel)
 - Prüfen ob folgende Dienste laufen, und falls nicht automatisch neu starten (protokollieren, wie oft dies passiert ist)
 - bird
 - Ausgabe von birdc show proto überprüfen, ob OSPF „running“ und alle bgp „established“ sind. Falls nicht: Icinga-Service auf 'Warning' setzen. (Wenn auf einem Gateway bind abschmirt, fallen die Routingverbindungen zum Partnergateway mit aus)
 - bird6
 - Ausgabe von birdc6 show proto überprüfen, ob OSPF „running“ und alle bgp „established“ sind. Falls nicht: Icinga-Service auf 'Warning' setzen. (Wenn auf einem Gateway bind abschmirt, fallen die Routingverbindungen zum Partnergateway mit aus)
 - kea
 - collectd
 - py-respond
 - Prüfen, ob ip r s t ffnet und ip -6 r s t ffnet einen default-Eintrag haben, falls nicht Email an hinterlegte Benachrichtigungsemail

Domänen-Monitoring:

Folgende Tests sollen pro Domäne durchgeführt werden. Dazu soll es pro Domäne ein Gluon geben und insgesamt ein oder zwei Testdebians. RAM-technisch können wir leider nicht 65 Debians und 65 Gluons laufen lassen, daher müssen diese der Reihe nach durchgeschaltet werden.

- Vorbereitungsphase:
 - Gluon starten
 - Auf Gluon Broker auf domaenexx-**A**.servers.tld setzen: uci set; /etc/init.d/tunneldigger restart (ohne commit)
 - Netzwerk des Test-Debians an dieses Gluon hängen (Code dafür kann [Matthias Walther](#) bereitstellen, gibt es schon fertig)
 - Vorschlag: Test-Debian über ein zweites Interface (eth1) mit lokaler, genatteter IP über SSH ansprechen. (Ich hatte das mal über seriell probiert, das ist ein ziemlicher Krampf, weil es keine Rückgabewerte gibt und man nicht weiß, wann ein Testskript durchgelaufen ist.)
 - Auf Test-Debian ifdown eth0; ifup eth0
 - Prüfen ob Gluon auf seiner IPV6 vom Debian pingbar ist, prüfen ob Gateway pingbar ist, falls nicht bis zu fünf Minuten warten, manchmal dauert die Tunnelleinwahl so lange
- Testphase-A:
 - ping 8.8.8.8 und zwei drei weitere IPV4s
 - ping auf zwei drei IPV6-Adressen
 - DNS-Test
 - Prüfen ob ein V4-DNS-Server gesetzt ist
 - Prüfen ob ein V6-DNS-Server gesetzt ist
 - Jeden einzelnen anfragen und ein paar Testabfragen machen
 - sowohl knoten.ffms
 - als auch externe wie heise.de, google.de
 - Vom Testdebian: Externe IPV4 prüfen und mit zu erwartender vergleichen, Abweichungen still protokollieren
 - Traceroute zu 8.8.8.8 durchführen und speichern
 - Traceroute zu fester IPV6 durchführen und speichern
- Testphase-B: Gluon auf domaeneXX-**B**.servers.tld umstellen und Tests analog Testphase-A wiederholen
- Nächste Domäne testen

Wie oben beschrieben, sollte das Domänen-Monitoring so aufgebaut sein, dass es sich parallelisieren lässt, damit zwei bis drei Domänen gleichzeitig getestet werden können. Es wäre gut, wenn jede Domäne alle 15-30 min getestet würde. Also je nach Dauer der Tests bräuchte man 5-10 Testmaschinen.

