Monitoring a dual homed device with Zabbix
whoami

Brian van Baekel

- Zabbix consultant
- Zabbix trainer
- Network engineer
Introduction
The problem

“Monitor the device in such a way that the monitoring is redundant”

Zabbix server: fully redundant. 2 interfaces per LAN (bonded)
Device: Single interfaces per lan
Network: Separated; no connections, no routers.
The problem

We are monitoring over the blue network.

-> Green network goes down, no problem.
The problem

We are monitoring over the blue network.

-> Blue network goes down, monitoring is gone.
The question

How to tell Zabbix to use the green network, if the blue network is down and vice versa?
The Challenge

Monitoring based on IP address will not work
The fix

DNS + some custom scripts.
The fix

We start monitoring a device using it’s DNS name instead of IP address
The fix

Item must be polled:
1: Request ip of device.xyz
2: device.xyz = 192.0.2.10

Monitoring happens over the blue network.
The fix

What if the blue network is down?

• Create a ‘keepalive’ item on the host with a trigger on it.
• If that trigger goes into problem state, fire an action: change DNS record
The fix

If trigger ‘device keepalive is in problem state, execute remote command:

```
python dnschange.py {blue ip} {green ip}
```

• API call to dns server to edit entry
The fix

Item must be polled:
1: Request ip of device.xyz
2: device.xyz = 198.51.100.10

Monitoring happens over the green network.
Type of DNS records

A record: device.xyz = IP

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Type</th>
<th>Content</th>
<th>Priority</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>xyz</td>
<td>SOA</td>
<td>2020021801 28800 7200 604800 86400</td>
<td></td>
<td>86400</td>
</tr>
<tr>
<td>33</td>
<td>device.xyz</td>
<td>A</td>
<td>192.0.2.10</td>
<td>0</td>
<td>8640</td>
</tr>
</tbody>
</table>

CNAME record: device.xyz = device1.xyz

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Type</th>
<th>Content</th>
<th>Priority</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>xyz</td>
<td>SOA</td>
<td>2020021804 28800 7200 604800 86400</td>
<td></td>
<td>86400</td>
</tr>
<tr>
<td>33</td>
<td>device.xyz</td>
<td>CNAME</td>
<td>device1.xyz</td>
<td>0</td>
<td>8640</td>
</tr>
<tr>
<td>34</td>
<td>device1.xyz</td>
<td>A</td>
<td>192.0.2.10</td>
<td>0</td>
<td>86400</td>
</tr>
<tr>
<td>35</td>
<td>device2.xyz</td>
<td>A</td>
<td>198.51.100.10</td>
<td>0</td>
<td>86400</td>
</tr>
</tbody>
</table>
Zabbix Host config

**A record:** device.xyz = IP

<table>
<thead>
<tr>
<th>Host</th>
<th>Templates</th>
<th>IPMI</th>
<th>Tags</th>
<th>Macros</th>
<th>Inventory</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Host macros</td>
<td>Inherited and host macros</td>
<td></td>
</tr>
<tr>
<td>Macro</td>
<td>Value</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>($IPA)</td>
<td>192.0.2.10</td>
<td>Blue network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>($IPB)</td>
<td>198.100.51.10</td>
<td>Green network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CNAME record:** device.xyz = device1.xyz

<table>
<thead>
<tr>
<th>Host</th>
<th>Templates</th>
<th>IPMI</th>
<th>Tags</th>
<th>Macros</th>
<th>Inventory</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Host macros</td>
<td>Inherited and host macros</td>
<td></td>
</tr>
<tr>
<td>Macro</td>
<td>Value</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>($IPA)</td>
<td>device1.xyz</td>
<td>Blue network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>($IPB)</td>
<td>device2.xyz</td>
<td>Green network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why the different approaches?

A record: device.xyz = IP

✓ Easier to see the Ipaddresses
✓ Easier to understand
✗ Nightmare with DTAP and different subnets

CNAME record: device.xyz = device1.xyz

✓ Bit harder to configure
✓ Bit harder to understand
✗ No problem with DTAP and different subnets
Did we cover everything?

- How about SNMPtraps?

Mismatch!

Received trap:

09:40:49 2020/02/18 ZBXTRAP 192.0.2.10
*truncated*
receivedfrom
*truncated

UDP: [192.0.2.10]:43205->[192.0.2.1]:162
Did we cover everything?

• How about SNMPtraps?

Fix? -> Adjust the default perl script, or build your own
If trap is received, extract the from ip, do a reverse
DNS lookup and parse the dns name

Received trap:

09:40:49 2020/02/18 ZBXTRAP device.xyz
*truncated*
receivedfrom
*truncated

UDP: [192.0.2.10]:43205->[192.0.2.1]:162
Did we cover everything?

- How about SNMPtraps?

**Received trap:**

09:40:49 2020/02/18 ZBXTRAP device.xyz
*truncated*
receivedfrom
*truncated

UDP: [192.0.2.10]:43205->[192.0.2.1]:162

**Match!**

<table>
<thead>
<tr>
<th>Item</th>
<th>Preprocessing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>snmptrap fallback item</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>SNMP trap</td>
</tr>
<tr>
<td><strong>Key</strong></td>
<td>snmptrap.fallback</td>
</tr>
<tr>
<td><strong>Host interface</strong></td>
<td>device.xyz : 161</td>
</tr>
<tr>
<td><strong>Type of information</strong></td>
<td>Numeric (unsigned)</td>
</tr>
</tbody>
</table>
Thank you