A Journey of Discovery
```bash
~$ whois fklaassen
```

Name: Frank Klaassen
Company: Enrise
Team: InfinIT
Job Title: Senior DevOps Engineer
~$ whois fklaassen

- Using Zabbix since 2008
- Introduced or improved existing Zabbix environments
- Long time open source user & contributor
- First time conference speaker
~$ whois enrise

- Enrise develops **high quality software** for the internet
- Founded in 2000
- 50+ people
- **Self-steering** company
~# whois infinit

- Managing internal IT
- Managing various environments on AWS & GCP
- Internal apps/dashboards
- DevOps consultancy
We use Zabbix for monitoring...

- Servers & Services
- Connectivity
- Provisioning

- SSL Certificates
- AWS services
- Application Components
Some facts

- AWS EC2 for Zabbix-server
- AWS RDS for database (MariaDB with partitioning)
- Zabbix-Proxies for external locations
- Zabbix: 4.4.x
- Hosts: 345
- Items: 19123
- Triggers: 5552
How not to create many items
How not to create many items

- Project with a whitelabel domain per customer
- All domains have 3 subdomains each
- SSL monitoring desired for 5 customer environments

That would mean:

5 domains x 3 subdomains = 15 items
How not to create many items

- Quick & dirty...
  You *could* create a template with the required 15 items
- With macros to at least make it *reusable* for the future
All was fine!
The team wanted to disable monitoring for one customer. 

*No problem! Let’s just empty out a few macros.*
Until...

- Oh yeah.. and please **add 3 new customers** as well while you’re at it!

So that would mean the number of items would go from 15 to 12 items and... back to 18.

It’s time for a better solution!
Low Level Discovery
Low Level Discovery

- **Built-in** functionality of Zabbix since 2.0
- **Automatically** creates items, triggers & graphs based on JSON input from Agent or Server
- Getting more advanced each release
  - *Hopefully one day also includes discovering web scenarios (ZBXNEXT-2621)*
What we use Low Level Discovery for

- PHP-FPM pools
- OpenVPN clients
- Gitlab health checks
- Docker containers
- Backup tasks
- Dnsmasq metrics

- SSL Certificates
- PM2 applications
- FreeSWITCH trunks
- Cronjobs
- Application Component Monitoring

But it can actually be used for anything that you can generate JSON for!
Solving the SSL monitoring issue
Without creating items by hand
Solving the SSL monitoring issue

Created a script reading the webserver (Apache, NGINX) configuration returning an array of all vhosts with a certificate

```json
[
  {
    "#SERVERPORT" : 443,
    "#SERVERNAME" : "enrise.com"
  },
  {
    "#SERVERPORT" : 443,
    "#SERVERNAME" : "enrise.nl"
  }
]
```
Solving the SSL monitoring issue
Solving the SSL monitoring issue
Solving the SSL monitoring issue
Solving the SSL monitoring issue
Solving the SSL monitoring issue

<table>
<thead>
<tr>
<th>Severity</th>
<th>Name</th>
<th>Operational data</th>
<th>Expression</th>
<th>Create enabled</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disaster</td>
<td>SSL certificate on [#SERVERNAME] expired</td>
<td></td>
<td>[Template SSL Certificate Auto - Agent:check_certificate: \d{#SERVERNAME},{SSL_PORT}] last()]&lt;1</td>
<td>Yes</td>
<td>ssl-certificate: [#SERVERNAME] ssl-expiry: [ITEM LASTVALUE1]</td>
</tr>
<tr>
<td>High</td>
<td>SSL certificate on [#SERVERNAME] expires in less than 7 days (ITEM LASTVALUE1 remaining)</td>
<td></td>
<td>[Template SSL Certificate Auto - Agent:check_certificate: \d{#SERVERNAME},{SSL_PORT}] last()]&lt;=7</td>
<td>Yes</td>
<td>ssl-certificate: [#SERVERNAME] ssl-expiry: [ITEM LASTVALUE1]</td>
</tr>
<tr>
<td>Average</td>
<td>SSL certificate on [#SERVERNAME] expires in less than 15 days (ITEM LASTVALUE1 remaining)</td>
<td></td>
<td>[Template SSL Certificate Auto - Agent:check_certificate: \d{#SERVERNAME},{SSL_PORT}] last()]&lt;=15</td>
<td>Yes</td>
<td>ssl-certificate: [#SERVERNAME] ssl-expiry: [ITEM LASTVALUE1]</td>
</tr>
<tr>
<td>Warning</td>
<td>SSL certificate on [#SERVERNAME] expires in less than 30 days (ITEM LASTVALUE1 remaining)</td>
<td></td>
<td>[Template SSL Certificate Auto - Agent:check_certificate: \d{#SERVERNAME},{SSL_PORT}] last()]&lt;=30</td>
<td>Yes</td>
<td>ssl-certificate: [#SERVERNAME] ssl-expiry: [ITEM LASTVALUE1]</td>
</tr>
</tbody>
</table>
Solving the SSL monitoring issue

- Items are created in Zabbix

<table>
<thead>
<tr>
<th>Name</th>
<th>Triggers</th>
<th>Key</th>
<th>Interval</th>
<th>History</th>
<th>Trends</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL Vhosts: SSL certificate validity - errise.com (Local)</td>
<td>Triggers 4</td>
<td>check_certificate[-d,errise.com,${SSL_PORT}]</td>
<td>4h</td>
<td>1w</td>
<td>30d</td>
<td>Zabbix agent (active)</td>
</tr>
<tr>
<td>SSL Vhosts: SSL certificate validity - errise.nl (Local)</td>
<td>Triggers 4</td>
<td>check_certificate[-d,errise.nl,${SSL_PORT}]</td>
<td>4h</td>
<td>1w</td>
<td>30d</td>
<td>Zabbix agent (active)</td>
</tr>
</tbody>
</table>
Solving the SSL monitoring issue

- And a set of triggers to notify us a certificate is about to expire

<table>
<thead>
<tr>
<th>Severity</th>
<th>Value</th>
<th>Name</th>
<th>Operational data</th>
<th>Expression</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disaster</td>
<td>OK</td>
<td>SSL Vhosts: SSL certificate on enrise.com expired</td>
<td></td>
<td>{web01.enrise.com.check_certificate[d.enrise.com,${SSL_PORT}],last()}&lt;1</td>
<td>Enabled</td>
</tr>
</tbody>
</table>
| High     | OK    | SSL Vhosts: SSL certificate on enrise.com expires in less than 7 days  
((ITEM.LASTVALUE1) remaining)  
Depends on: web01.enrise.com: SSL certificate on enrise.com expired |                                                                                  | {web01.enrise.com.check_certificate[d.enrise.com,${SSL_PORT}],last()}<=7            | Enabled|
| Average  | OK    | SSL Vhosts: SSL certificate on enrise.com expires in less than 15 days  
((ITEM.LASTVALUE1) remaining)  
Depends on: web01.enrise.com: SSL certificate on enrise.com expires in less than 7 days  
((ITEM.LASTVALUE1) remaining) |                                                                                  | {web01.enrise.com.check_certificate[d.enrise.com,${SSL_PORT}],last()}<=15           | Enabled|
| Warning  | OK    | SSL Vhosts: SSL certificate on enrise.com expires in less than 30 days  
((ITEM.LASTVALUE1) remaining)  
Depends on: web01.enrise.com: SSL certificate on enrise.com expires in less than 15 days  
((ITEM.LASTVALUE1) remaining) |                                                                                  | {web01.enrise.com.check_certificate[d.enrise.com,${SSL_PORT}],last()}<=30           | Enabled|
Monitoring OpenVPN Clients

Without creating items by hand
Monitoring OpenVPN Clients

- Users generate their own configuration via our VPN Portal
- UserParameter script on the VPN instance returns a list of users to Zabbix Discovery
- Zabbix creates the desired items (traffic in/out & client IP)

<table>
<thead>
<tr>
<th>Name</th>
<th>Last check</th>
<th>Last value</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPN Clients (3 Items)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenVPN Client fklassen - Traffic Outgoing</td>
<td>2020-02-14 10:30:27</td>
<td>1.46 Mbytes</td>
<td>+52.28 Kbytes</td>
</tr>
<tr>
<td>OpenVPN Client fklassen - Traffic Incoming</td>
<td>2020-02-14 10:30:27</td>
<td>17.92 Mbytes</td>
<td>+1.86 Mbytes</td>
</tr>
<tr>
<td>OpenVPN Client fklassen - IP</td>
<td>2020-02-14 10:30:27</td>
<td>256.128.512.64</td>
<td></td>
</tr>
</tbody>
</table>
Monitoring OpenVPN Clients

- Tracking bandwidth usage per user
- Tracking accidental usage at office

With automatic Slack notifications
Application Component Monitoring
Application Component Monitoring

- External API’s
- Database connectivity
Application Component Monitoring

Dev Team

- Knows best how to monitor their application
- Does not have or need knowledge of Zabbix
- Wants to make changes without involving Ops

Ops Team

- Has Zabbix knowledge
- Does not (need to) know what components the application has
Application Component Monitoring

Key Endpoint
(/monitoring/keys.json)

```json
{
    "Warning Component": {
        "severity": "warning"
    },
    "Average Component": {
        "severity": "average"
    },
    "High Component": {
        "severity": "high"
    },
    "Disaster Component": {
        "severity": "disaster"
    }
}
```

Status Endpoint
(/monitoring/status.json)

```json
{
    "Warning Component": {
        "statusCode": 0,
        "severity": "warning"
    },
    "Average Component": {
        "statusCode": 0,
        "severity": "average"
    },
    "High Component": {
        "statusCode": 0,
        "severity": "high"
    },
    "Disaster Component": {
        "statusCode": 0,
        "severity": "disaster"
    }
}
```
# Application Component Monitoring

<table>
<thead>
<tr>
<th>Name</th>
<th>Triggers</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components: average: Component status - Average Component</td>
<td>Triggers</td>
<td>aamv2.status[Average Component,average]</td>
</tr>
<tr>
<td>Components: disaster: Component status - Disaster Component</td>
<td>Triggers</td>
<td>aamv2.status[Disaster Component,disaster]</td>
</tr>
<tr>
<td>Components: high: Component status - High Component</td>
<td>Triggers</td>
<td>aamv2.status[High Component,high]</td>
</tr>
<tr>
<td>Components: Information: Component status - Information Component</td>
<td>Triggers</td>
<td>aamv2.status[Information Component,information]</td>
</tr>
<tr>
<td>Components: unclassified: Component status - Unclassified Component</td>
<td>Triggers</td>
<td>aamv2.status[Unclassified Component,unclassified]</td>
</tr>
<tr>
<td>Components: warning: Component status - Warning Component</td>
<td>Triggers</td>
<td>aamv2.status[Warning Component,warning]</td>
</tr>
<tr>
<td>Template Automatic Application Monitoring v2: AAMv2: Components Retrieval</td>
<td>Triggers</td>
<td>aamv2.py&quot;--config&quot;&quot;/etc/zabbix/zabbix_agentd.aam.conf&quot;:&quot;--discovery&quot;:&quot;[HOST:HOST]:&quot;&quot;[$SAAM_KEY_ENDPOINT]&quot;&quot;</td>
</tr>
<tr>
<td>Template Automatic Application Monitoring v2: AAMv2: Status Retrieval</td>
<td>Triggers</td>
<td>aamv2.py&quot;--config&quot;&quot;/etc/zabbix/zabbix_agentd.aam.conf&quot;:&quot;--update-items&quot;:&quot;[HOST:HOST]:&quot;&quot;[$SAAM_STATUS_ENDPOINT]&quot;&quot;</td>
</tr>
</tbody>
</table>
Application Component Monitoring

Opensource! Source is available on GitHub:

https://enri.se/zabbix-appmonitor
Questions?