Zabbix Server-to-Server Replication

Dealing with multi-tenant multi-integration environments
Background: New Customer Project

• Logistics company
• Own IT department
  – Developing and operating their own business software
  – Several hundred employees on their own
• sysfive.com tasks
  – Log Monitoring (Elastic Stack)
  – Application Monitoring (Zabbix)
  – Application Server (Third-Level-Support)
Resulting problems

- **Sysfive.com standard procedures**
  - Full platform operation
  - Centralised monitoring (one dashboard for Ops Team)
  - Integration of Monitoring with other tools

- **Special Customer Project**
  - Selection of services
  - Monitoring “as a service” for other teams
  - Providing monitoring servers at the customers site

- **Problems**
  - Both companies want to “own” the monitoring
  - Limited possibilities for integration with other sysfive Services
Sysfive Infrastructure (so far)
Expansion (rejected)
Expansion using replication
Solution

• Server-to-Server-Replication!
  – Receiver: sysfive Zabbix
  – Sender: customer’s Zabbix
    • Only limited access rights acceptable
  – Replicate all data and necessary configuration
    • Only item/trigger configuration
    • Only limited set of hosts
    • No templates etc, because no configuration done on Receiver

• Conclusion:
  – We need a light-weight replicator that can run as an API read-only user
  – It has to run on the sysfive server that works as a “monitoring aggregator”
Solution
Technical Solution

- **Python Replication Script**
  - Based on “pyzabbix” module
  - Using a simple configuration file for connection + hosts
  - Automatically replicate wanted hosts (items & triggers)

- **Use only one Interface on Data Source**
  - Zabbix REST API

- **Use two interfaces on Receiver**
  - Zabbix REST API for configuration
  - Zabbix Sender API for monitoring data

- **Automatic conversions and creations**
  - All items become Trapper Items
  - Per Host a Replication Monitor Item+Trigger is created
    - All replicated triggers depend on this
Technical Solution

- Running as a cron job
- Uses an flock on the configuration to prevent “overrunning”
- Automatically detects which host configuration have to be replicated
- Local “replica cache” for efficient and fast replication (minimize API calls)
- https://github.com/sysfivecom/zabbix-replicator
Limitations

- Not running as a continuous service
- Not supporting replication of more types (Applications, Templates…)
  - This could be added easily, if you need it, go for it!
- Not automatically detecting changing configurations
  - Need a forced replication of configurations at the moment
Demonstration

- No Real Data available
  - Production system not yet implemented

- Test setup
  - Sysfive.com Zabbix Server as Source
  - Vagrant Zabbix Server as target
  - Replicate 12 Hosts

- Demonstration Tasks
  - Show initial replication
  - Show performance impact
Demonstration – Empty Target
Demonstration – 5min initial replication

<table>
<thead>
<tr>
<th>Name</th>
<th>Applications</th>
<th>Items</th>
<th>Triggers</th>
<th>Graphs</th>
<th>Discovery</th>
<th>Web</th>
<th>Interface</th>
<th>Templates</th>
<th>Status</th>
<th>Availability</th>
<th>Agent encryption</th>
<th>Infs</th>
</tr>
</thead>
<tbody>
<tr>
<td>sshv121</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Discovery</td>
<td>Web</td>
<td>Interface</td>
<td>Templates</td>
<td>Status</td>
<td>Availability</td>
<td>Agent encryption</td>
<td>Infs</td>
</tr>
<tr>
<td>sshv121</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Discovery</td>
<td>Web</td>
<td>Interface</td>
<td>Templates</td>
<td>Status</td>
<td>Availability</td>
<td>Agent encryption</td>
<td>Infs</td>
</tr>
<tr>
<td>sshv122</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Discovery</td>
<td>Web</td>
<td>Interface</td>
<td>Templates</td>
<td>Status</td>
<td>Availability</td>
<td>Agent encryption</td>
<td>Infs</td>
</tr>
<tr>
<td>sshv123</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Discovery</td>
<td>Web</td>
<td>Interface</td>
<td>Templates</td>
<td>Status</td>
<td>Availability</td>
<td>Agent encryption</td>
<td>Infs</td>
</tr>
<tr>
<td>sshv124</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Discovery</td>
<td>Web</td>
<td>Interface</td>
<td>Templates</td>
<td>Status</td>
<td>Availability</td>
<td>Agent encryption</td>
<td>Infs</td>
</tr>
<tr>
<td>sshv125</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Discovery</td>
<td>Web</td>
<td>Interface</td>
<td>Templates</td>
<td>Status</td>
<td>Availability</td>
<td>Agent encryption</td>
<td>Infs</td>
</tr>
<tr>
<td>Zabbix server</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Discovery</td>
<td>Web</td>
<td>Interface</td>
<td>Templates</td>
<td>Status</td>
<td>Availability</td>
<td>Agent encryption</td>
<td>Infs</td>
</tr>
</tbody>
</table>

Template App Zabbix Server, Template OS Linux, Template App Zabbix Agent
Demonstration – 1h history per item
Demonstration – Target Server Load
Demonstration – Source Server Load
Demonstration - Conclusion

• Initial Replication
  – Very High Load on target
  – Significant load increase on source

• Ongoing Replication
  – Small load on both servers
    • Increases with number of hosts, no large scale solution
  – Comparable to one or two users opening the dashboard
## Before vs After

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two different dashboards to maintain</td>
<td>One central dashboard</td>
</tr>
<tr>
<td>Limited automation integration</td>
<td>Full workflow integration</td>
</tr>
<tr>
<td>2x the work for user/notification setup</td>
<td>One central user/notification setup</td>
</tr>
<tr>
<td>Split data ownership</td>
<td>Shared data ownership</td>
</tr>
<tr>
<td>Some notifications impossible because of company limits (SMS)</td>
<td>All notifications working</td>
</tr>
</tbody>
</table>
• Thank you for your attention!