initMAX

ZABBIX PREMIUM PARTNER

ZABBIX CERTIFIED TRAINER

Webinar

Zabbix Scripting JavaScript and Automation

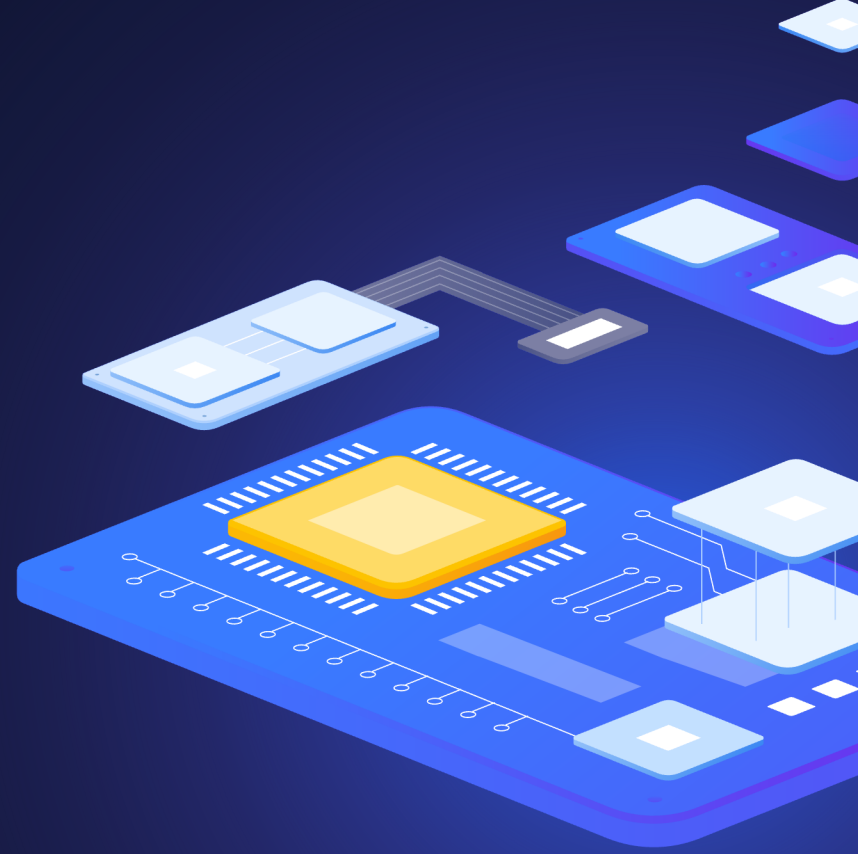
All microphones are muted.

Please ask your questions in Q&A, not in the chat.

Use the chat for discussion, networking, or applause.

1

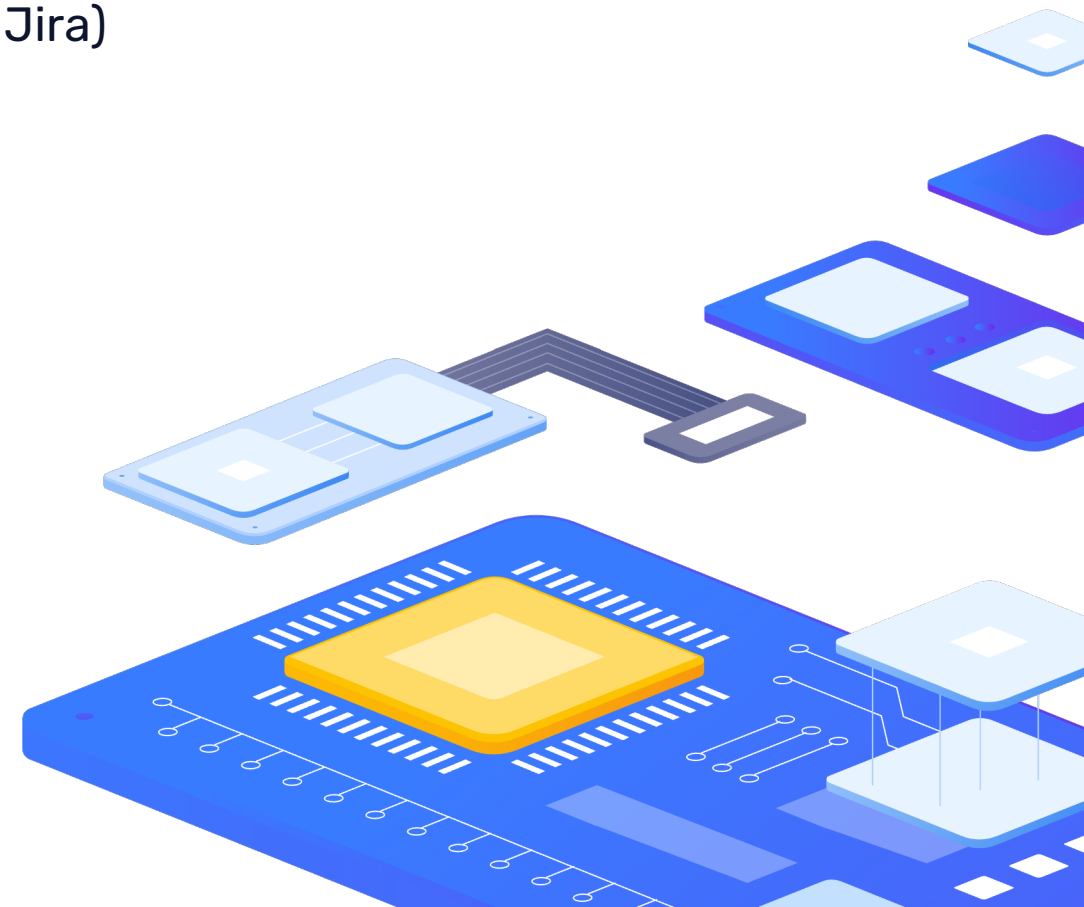
Automation in Zabbix



initMAX

Why Automation in Monitoring

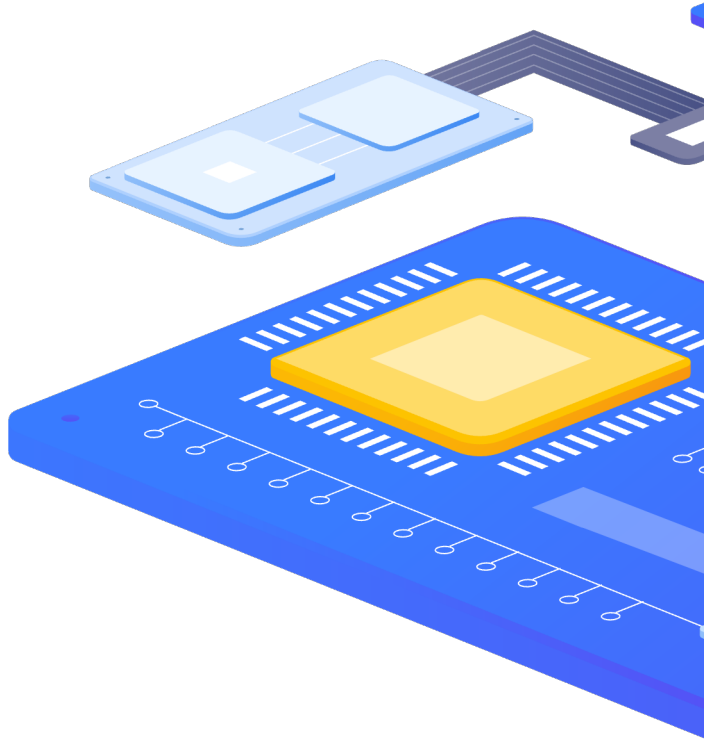
- Manual remediation slows incident resolution
- Tickets often created manually in ITSM tools (ServiceNow, Jira)
- Repetitive operational tasks consume admin time
- Human errors increase operational risk
- Slow response leads to increased Recovery Time



initMAX

How Zabbix enables Automation

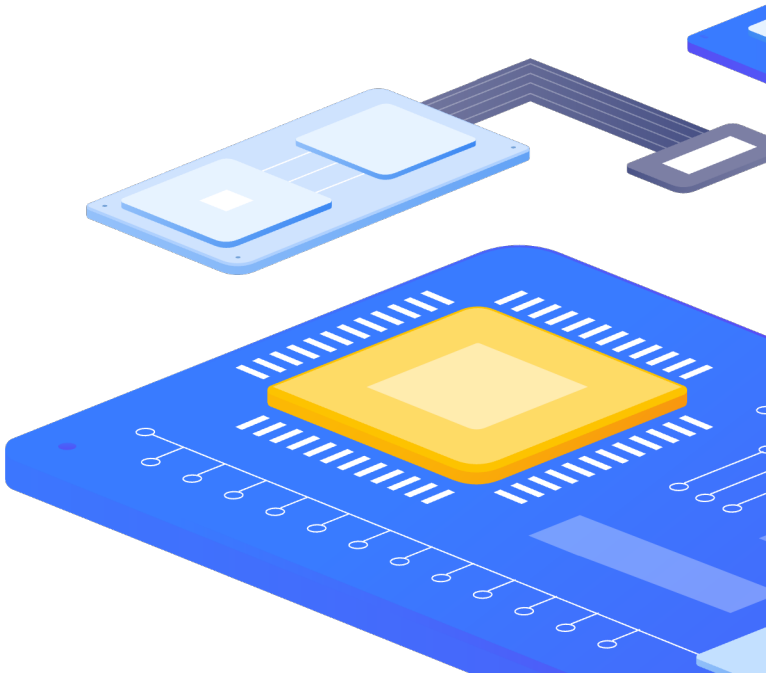
- Zabbix continuously monitors infrastructure metrics
- Triggers detect problems based on defined conditions
- A detected problem generates a Problem Event
- Actions evaluate conditions and execute operations
- Operations can execute scripts, webhooks, or commands
- This allows Zabbix to automatically respond to incidents
- Zabbix is not only a monitoring tool — it is also an automation platform



initMAX

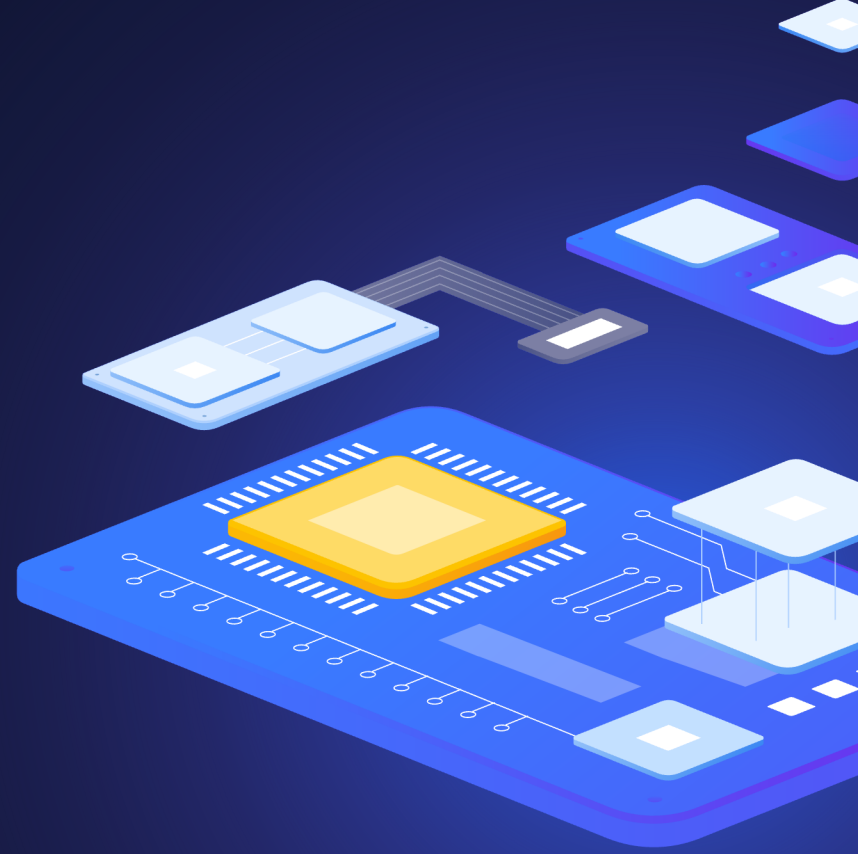
Available Automation and Scripting in Zabbix

- Manual Operations
  - Frontend Scripts
    - Manual Host actions
    - Manual Event actions
- Automatic Remediation
  - Actions (Execute Scripts Automatically)
  - External Integrations (Service Now, Jira, Slack, Custom API)
- Data Preprocessing
- External Scripts



1

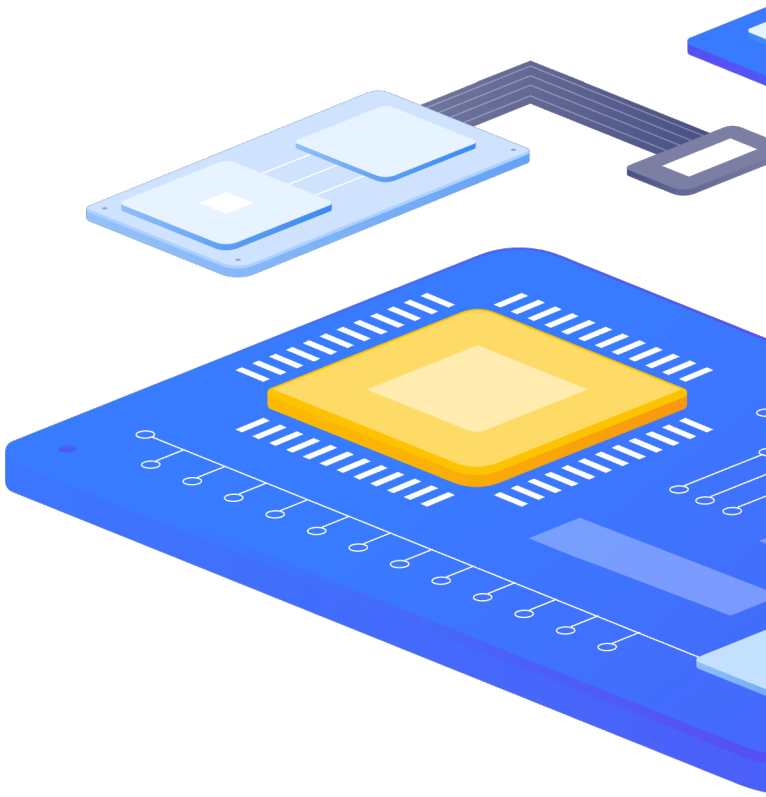
Frontend Scripting



initMAX

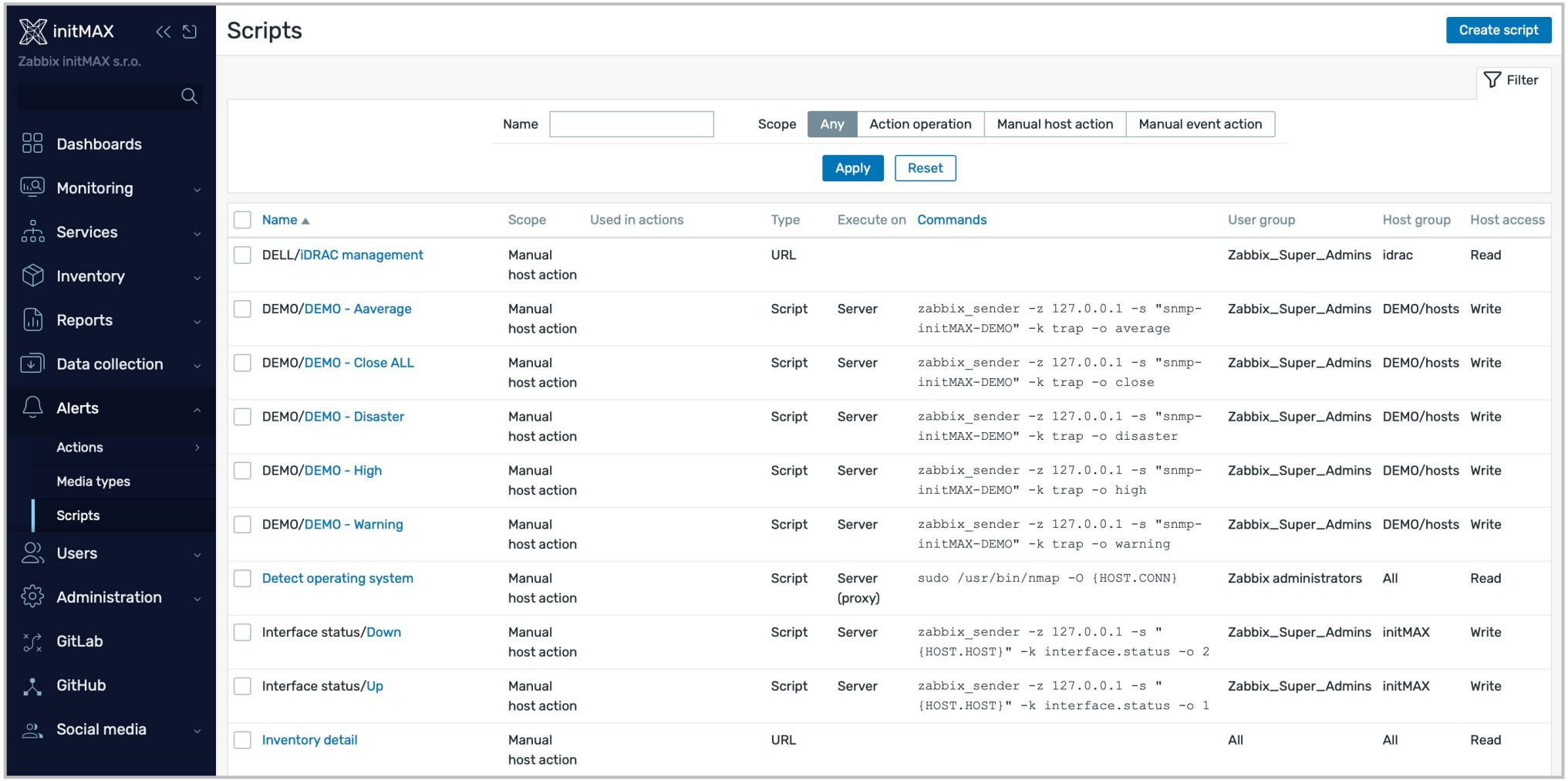
What is a Frontend script?

- Frontend scripts are executed directly from the Zabbix frontend
- They allow administrators to perform operational tasks directly from:
  - Host View
  - Problem View
- Typical Use Cases include
  - Manual remediation of a problem
  - Running troubleshooting commands
  - Performing operational tasks (restart services, diagnostics, etc.)



initMAX

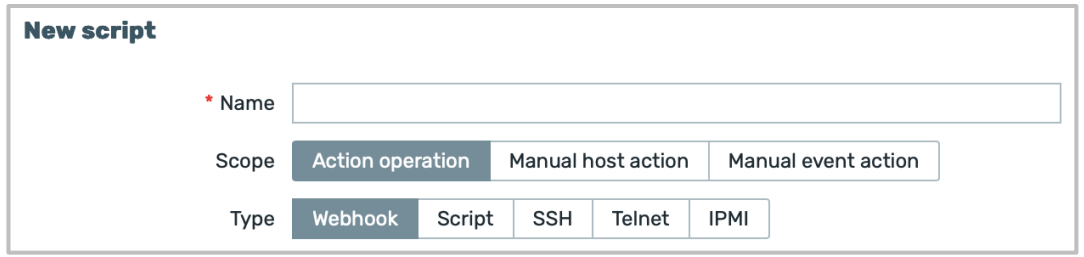
Frontend Scripts

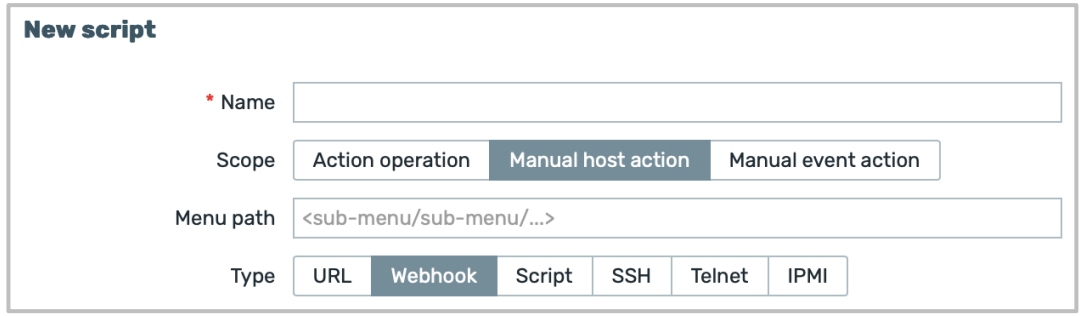


initMAX

Script execution scope

- Frontend scripts can be executed in different contexts
- Action Operation
  - Script executed automatically as part of an Action
- Manual Host action
  - Script executed manually on a Host
- Manual Event action
  - Script executed manually for a specific event

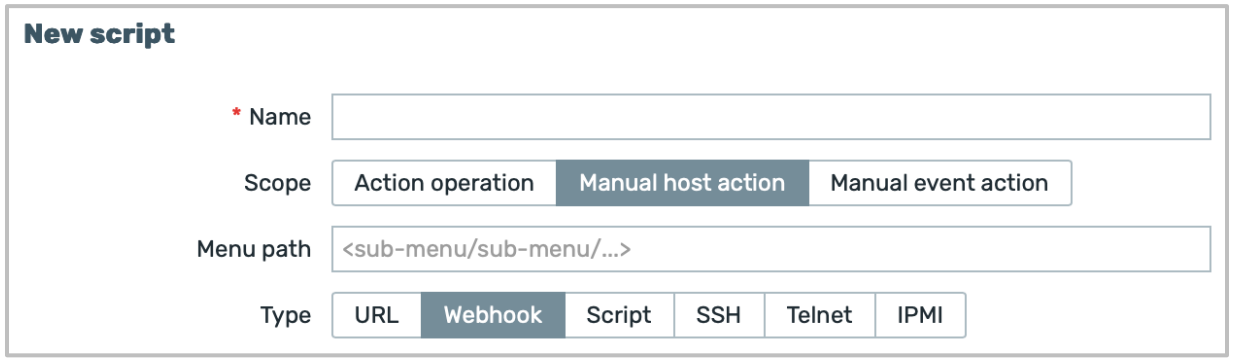




initMAX

Manual Host action

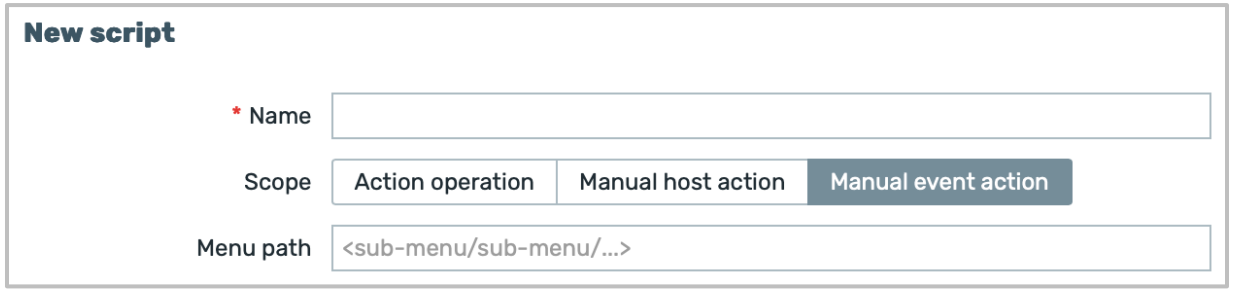
- Scripts executed manually on a specific host
- Typical Use Cases
  - Restart a host or a service
  - Collect diagnostic information (logs, system status)
  - Run operational scripts (cleanup, configuration checks)
  - Identify responsible team or system owner



initMAX

Manual Event action

- Scripts executed manually from the Problem/Event view
- Used when a problem is detected and an operator needs to take action.
- Typical Use Cases
  - Restart a failed service
  - Create a ticket in ITSM System
  - Run troubleshooting and diagnostic scripts
  - Trigger remediation workflows



initMAX

Frontend Script Macro usage

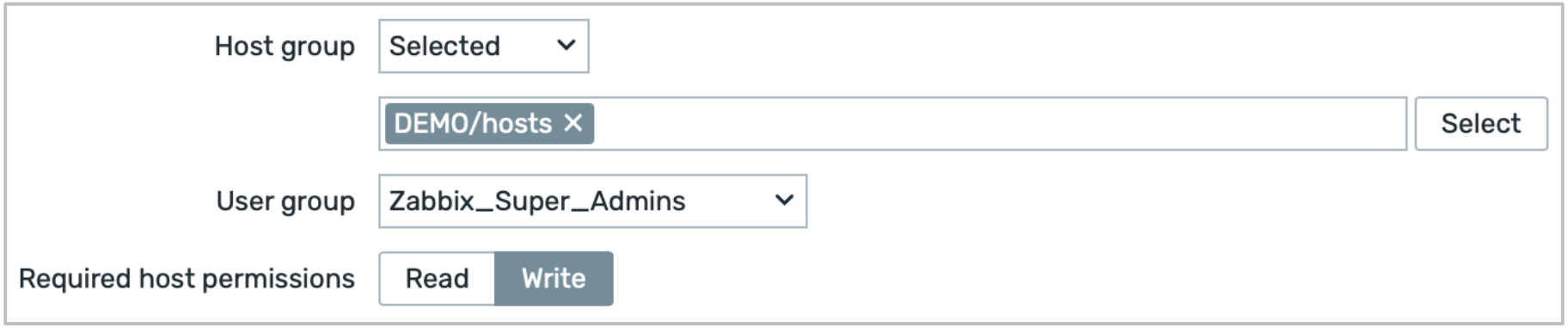
- Supported Macros
- https://www.zabbix.com/documentation/current/en/manual/appendix/macros/supported_by_location
- Available macros **differ** between Zabbix versions

Scripts		
Commands, confirmation text (Type: script, manual **host action**)	Host macros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}[1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Host inventory macros[2] {MANUALINPUT} Username macros User-definable macros	[1] Confirmation text only [2] Supported for Zabbix server and Zabbix proxy
Commands (Type: script, manual **event action**)	Date/time macros Event macros, Cause/symptom event macros, Recovery event macros (if recovery took place)	

initMAX

Manual Action restriction

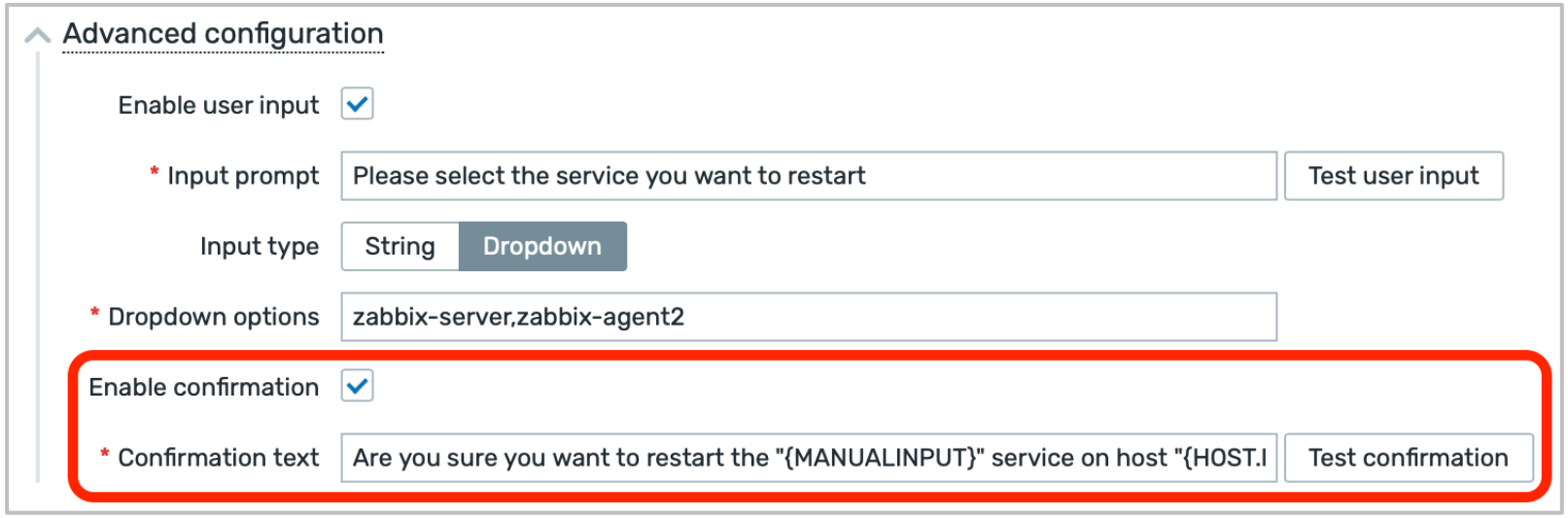
- Frontend scripts can be restricted to control who can execute them and where.
- Restrictions can be configured based on:
  - Selected Host group (Limit scripts on specific infrastructure group)
  - Selected User group (Allow only authorized users to execute scripts)
  - Required Host Permissions (Require specific access levels)

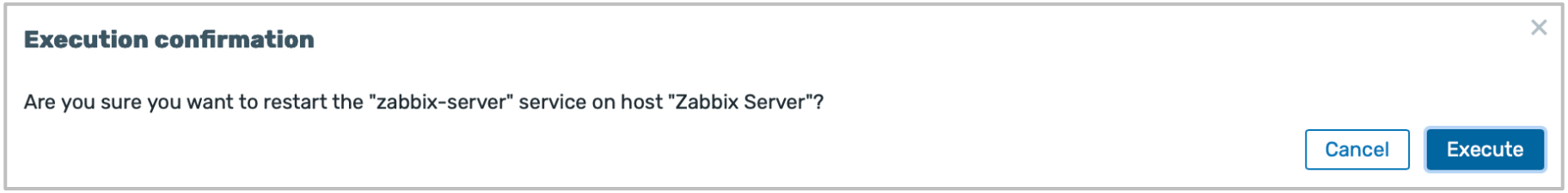


initMAX

Confirmation on script execution

- To prevent accidental execution of critical scripts, Zabbix allows a confirmation dialog before running a script.
- This helps protect against unintended actions.

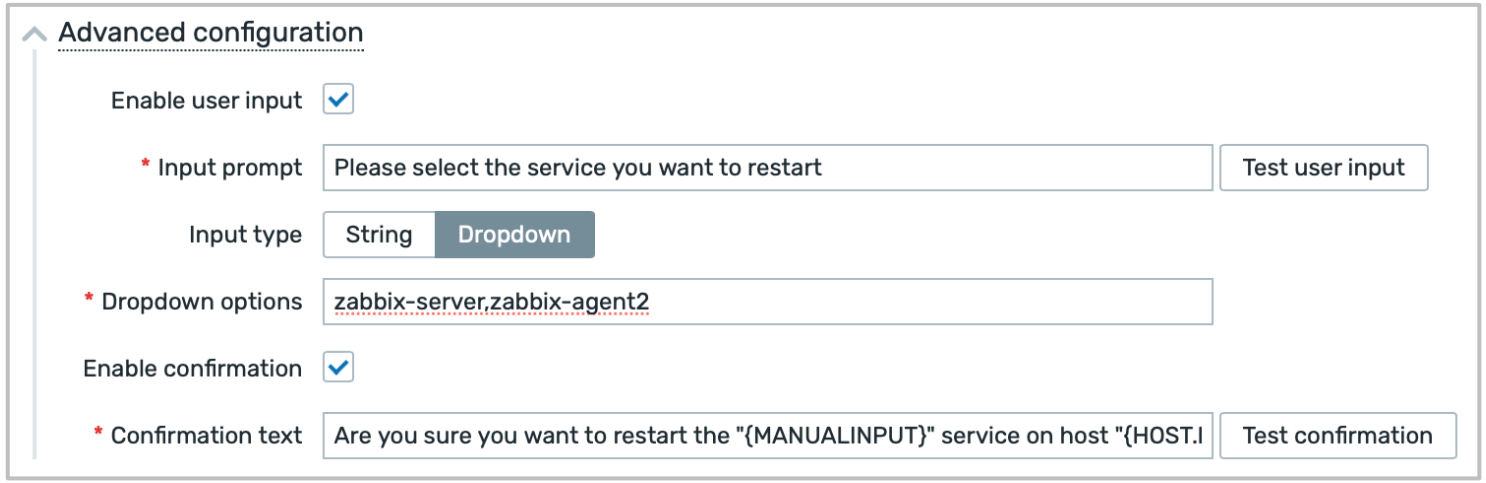


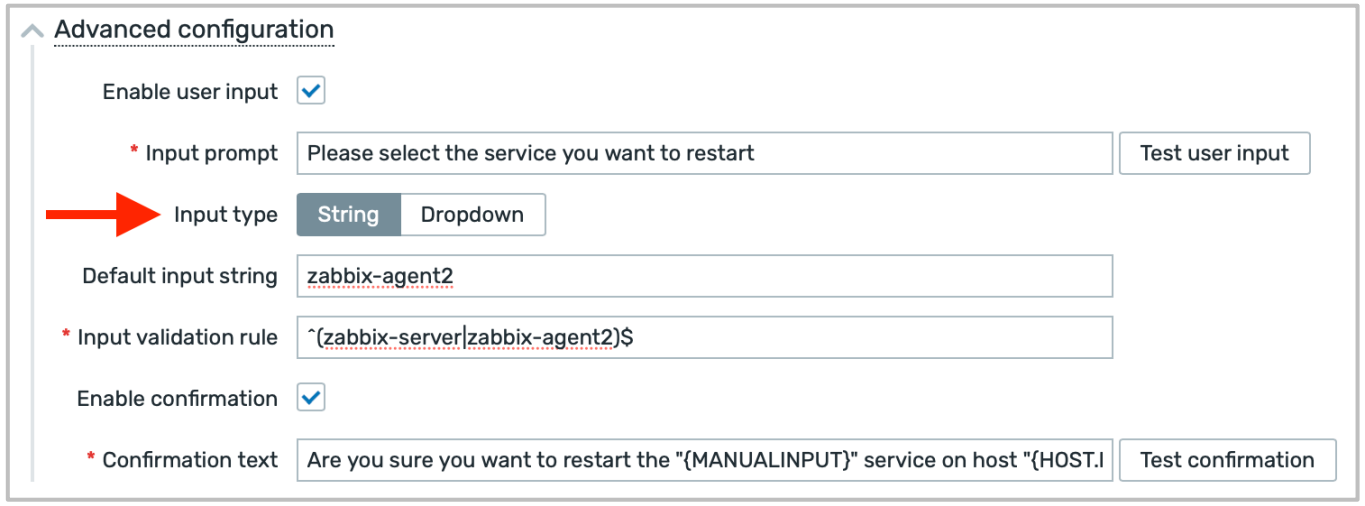


initMAX

Zabbix – Advanced Configuration

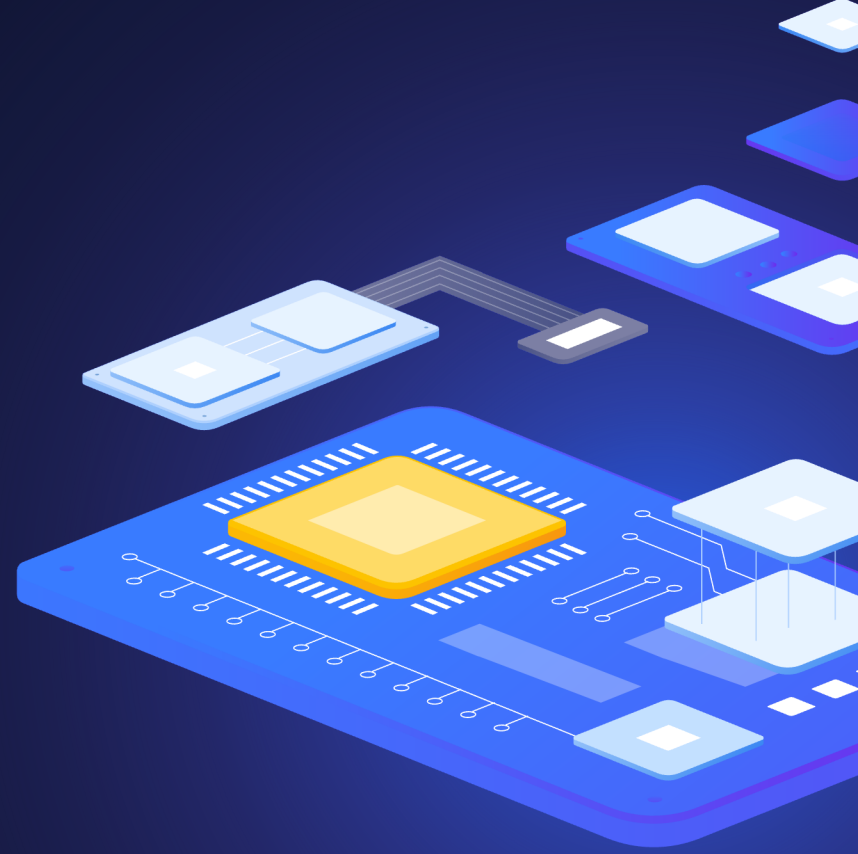
- User input
  - String
  - Dropdown
- Input validation rule
- Safety confirmation dialog





2

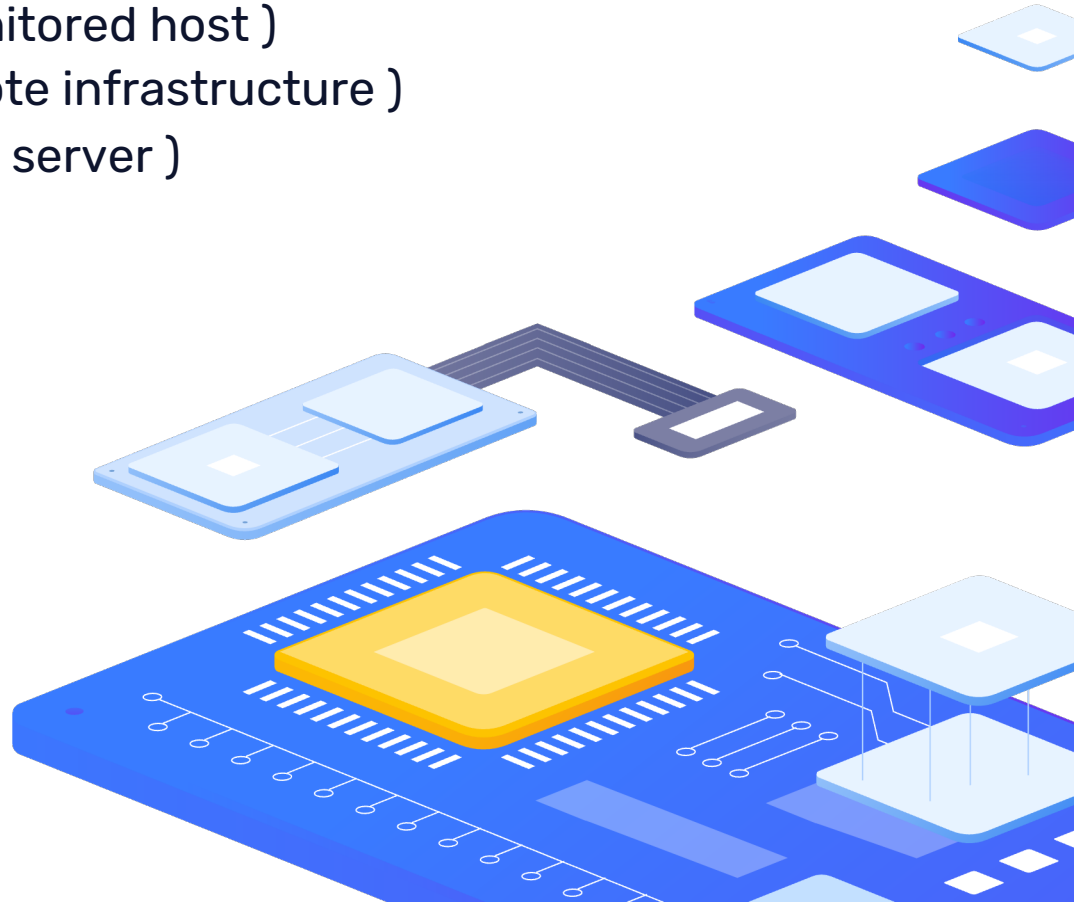
Script Execution Targets



initMAX

Script Execution Target

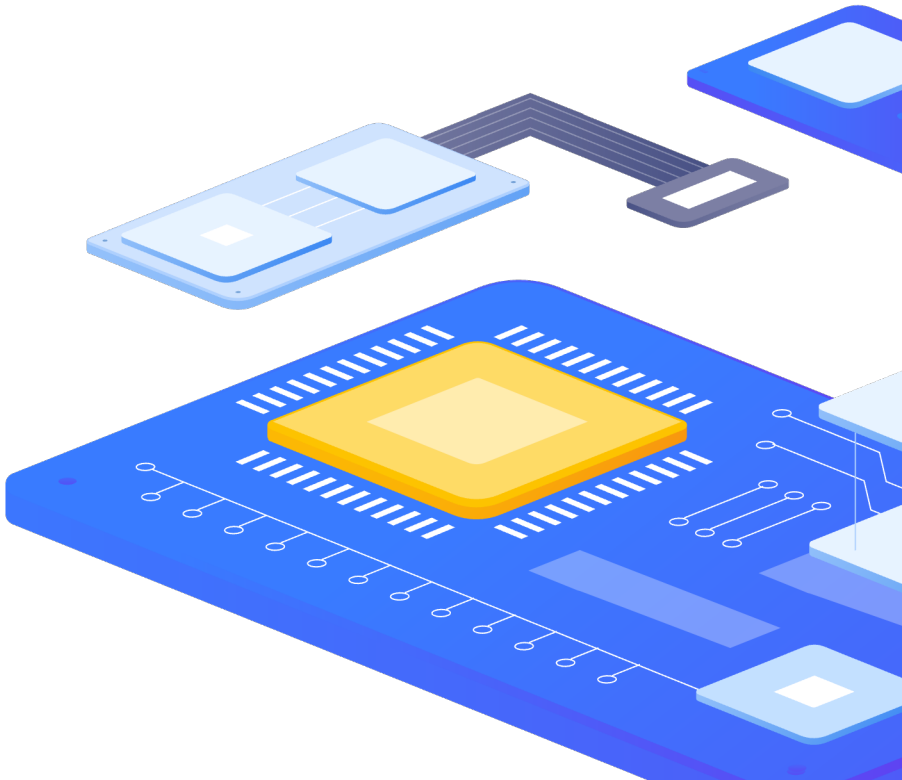
- Frontend scripts can be executed on different components of the Zabbix infrastructure.
- Execute on:
  - Zabbix agent (Execute commands directly on the monitored host)
  - Zabbix server or proxy (Execute scripts closer to remote infrastructure)
  - Zabbix server (Execute scripts centrally on the Zabbix server)



initMAX

Webhook

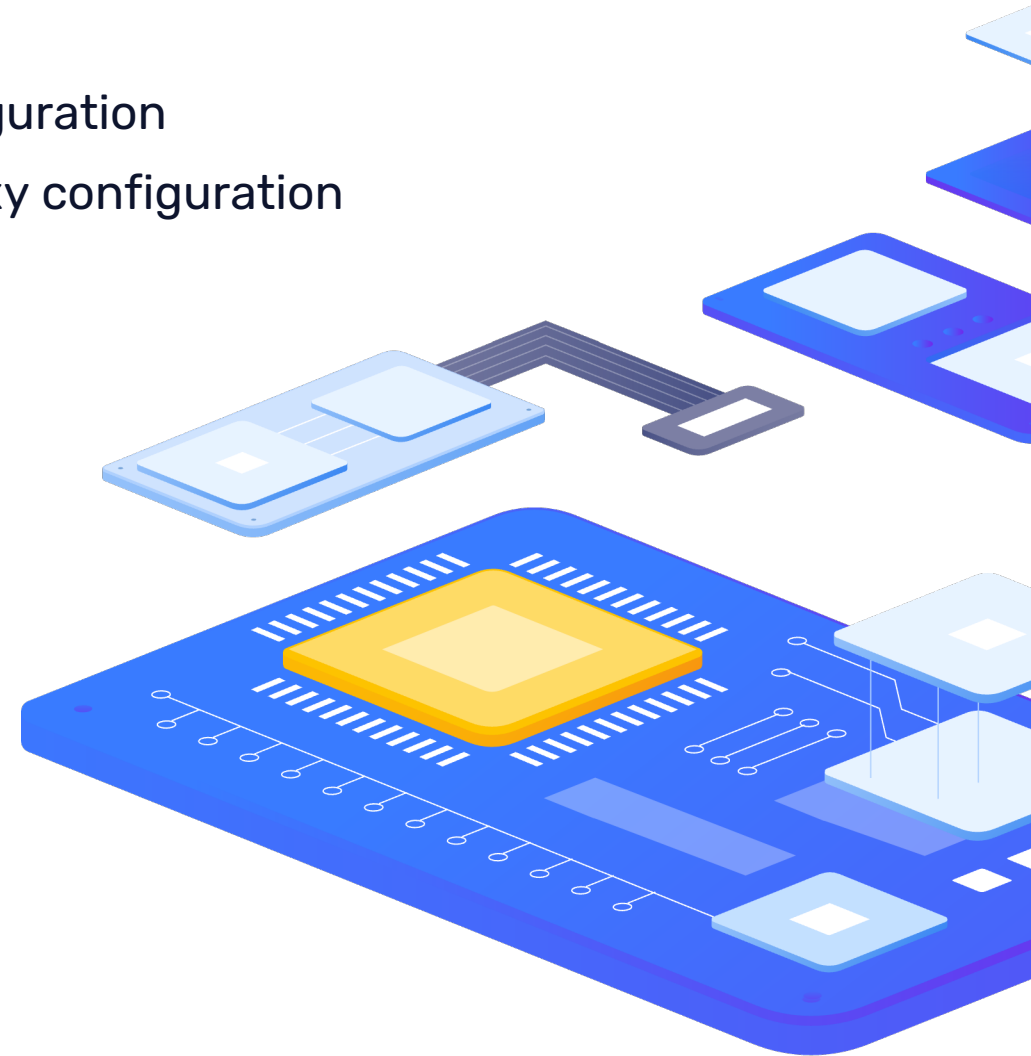
- Webhooks allow Zabbix to communicate with external systems using HTTP APIs.
- Webhooks are typically used for
  - Communication with external systems:
    - Ticketing systems (Jira, request Tracker, OTRS, …)
    - Automation (AWX, Ansible)
    - Inventory systems (CMDB)
    - Callbacks to the Zabbix server
- Webhooks are written in JavaScript
- Webhooks use the built-in HttpRequest() object to call APIs



initMAX

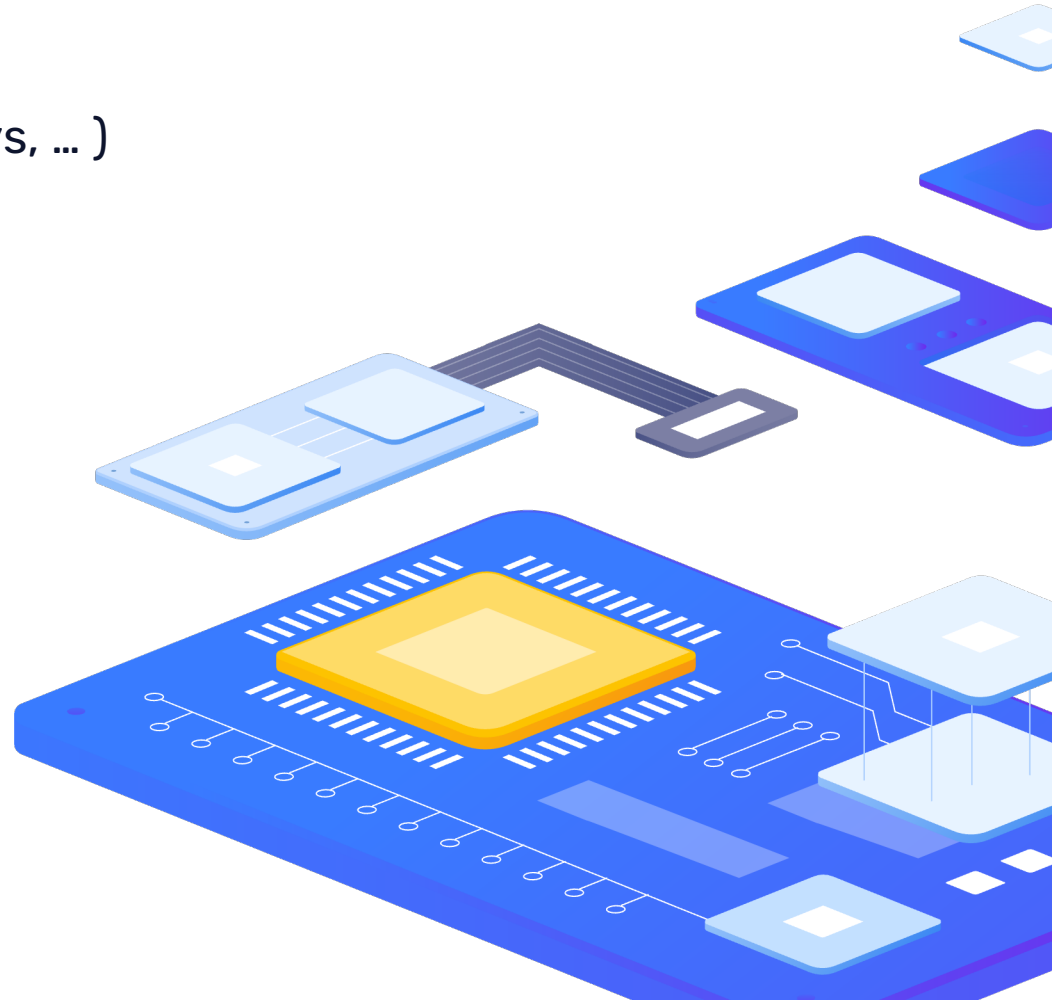
Frontend Script

- Scripts can run on the server, proxy, or agent
- adding an AllowKey=system.run[*] parameter in agent configuration
- setting the EnableRemoteCommands parameter to '1' in proxy configuration
- Target systems:
  - Linux - bash script or command
  - Windows – cmd command or script
- wait / nowait status



SSH / Telnet

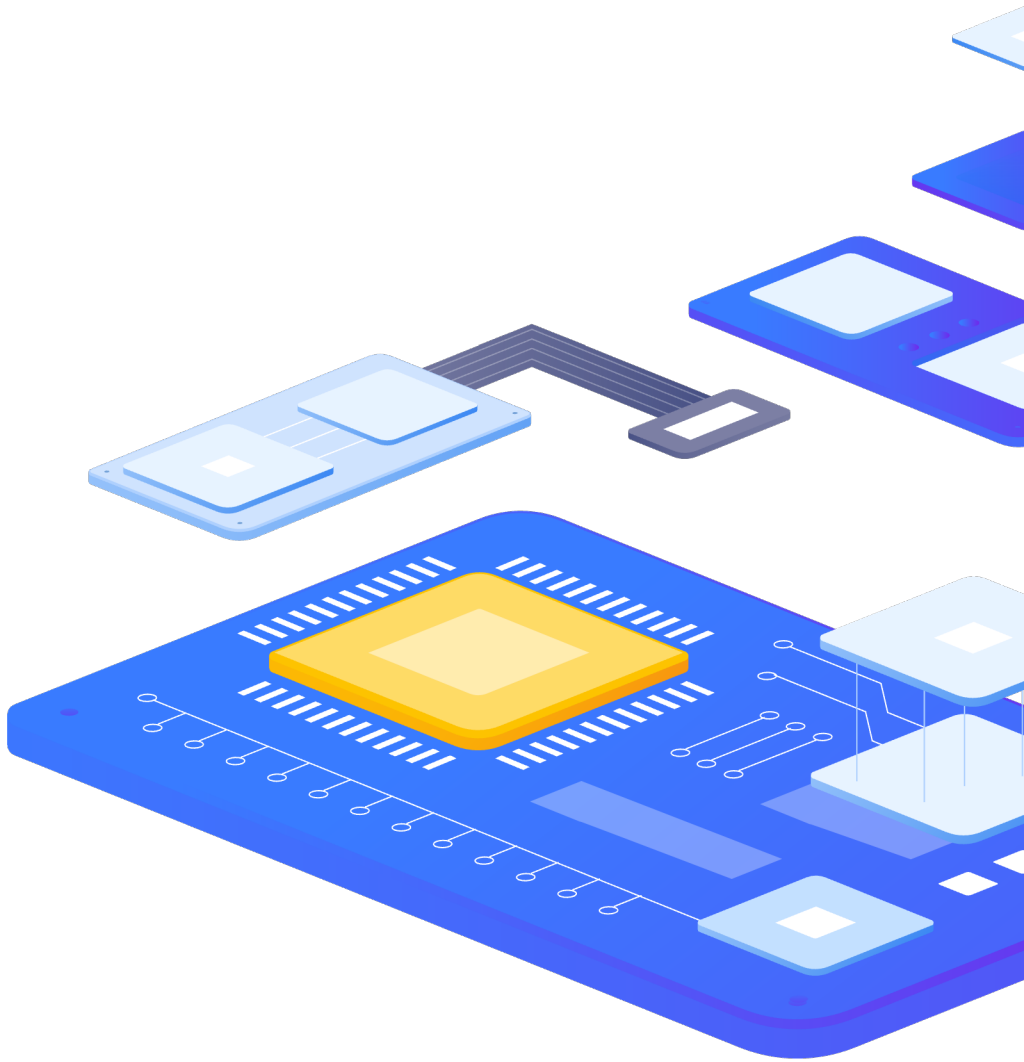
- Remote command execution using the ssh protocol
- Password or key based authentication
- Calls to devices without zabbix agents (switches, disk arrays, …)



initMAX

IPMI

- Remote command execution using the IPMI protocol
- Calls to hardware devices (iDrac, iLo, ..)



initMAX

Zabbix agent configuration

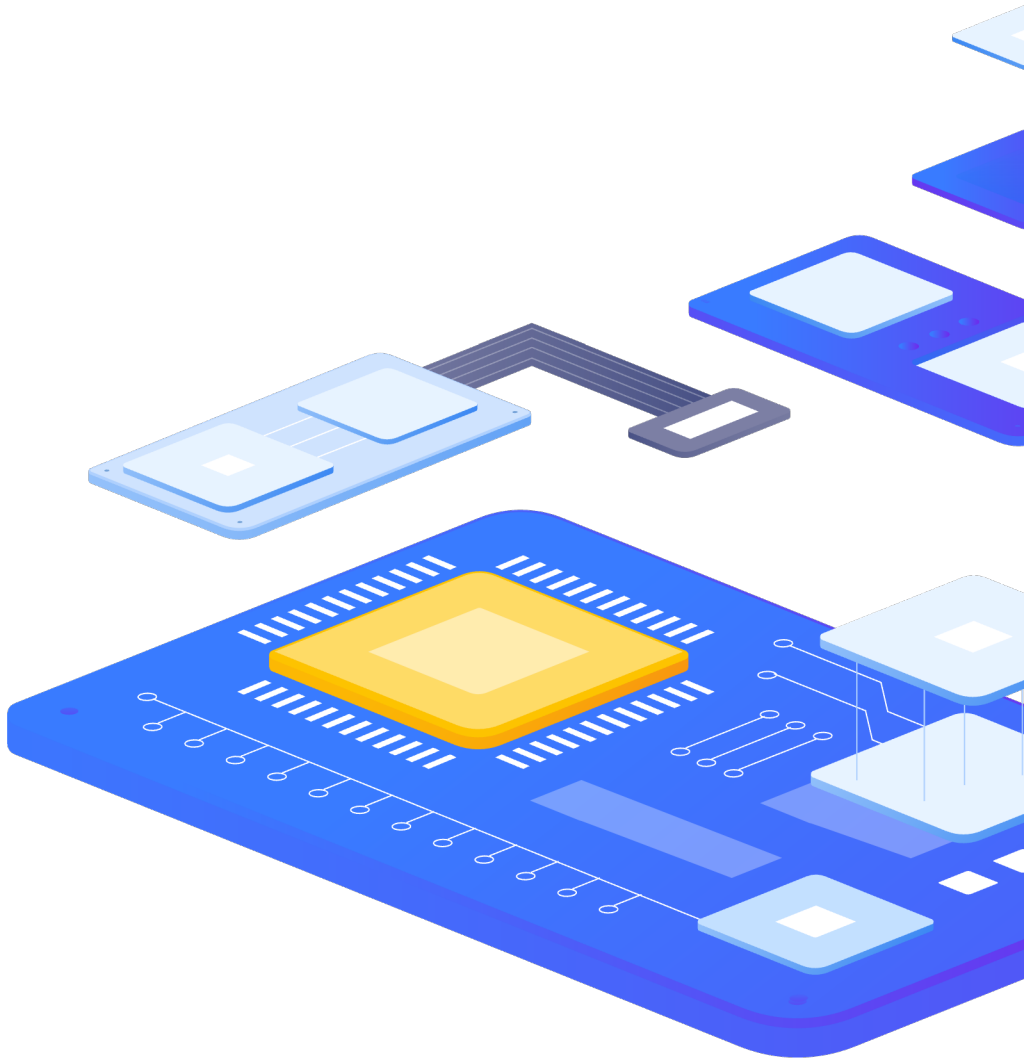
- AllowKey=system.run[shutdown –r –t 0]
- AllowKey =system.run[shutdown –r –t 0,*]
- DenyKey =system.run[*]

- wait / nowait type
  - wait: manual Actions, Item checks
  - nowait: Action operation
- Examples:
  - AllowKey = system.run[shutdown –r –t 0]
  - AllowKey = system.run[shutdown –r –t 0,wait]
  - AllowKey = system.run[shutdown –r –t 0,nowait]
  - AllowKey = system.run[shutdown –r –t 0,*]

- Agent user rights on system?

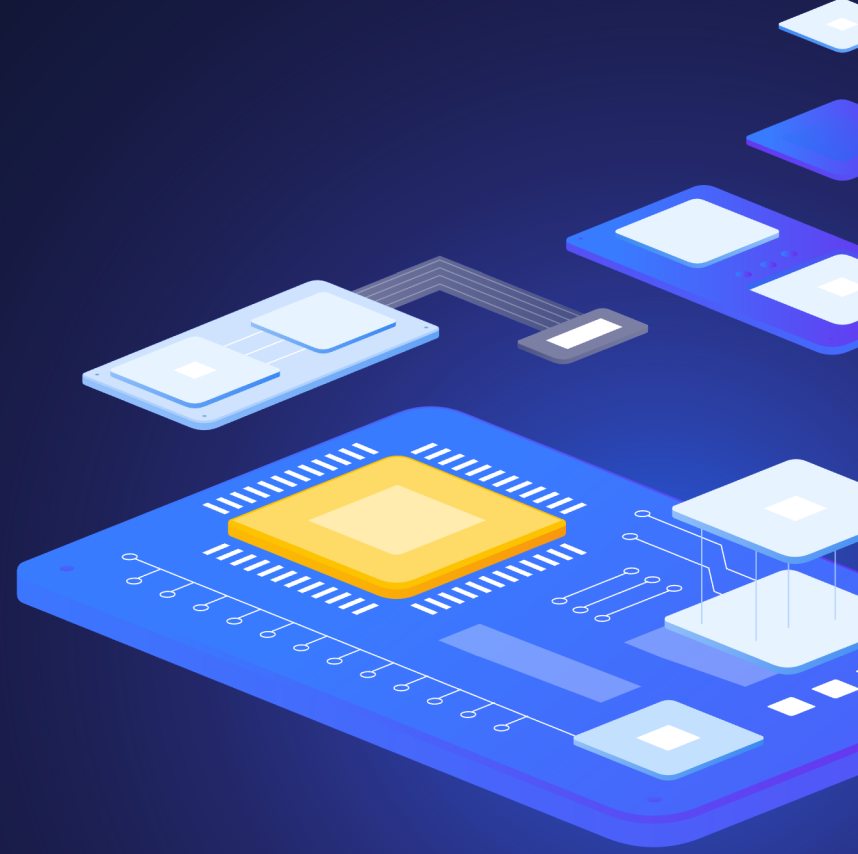
Zabbix version < 5.0

- EnableRemoteCommands



3

JavaScript usage in Zabbix



initMAX

JavaScript capabilities

- Built-In Objects
  - Zabbix

Alias	Alias to
console.log(object)	Zabbix.log(4, JSON.stringify(object))
console.warn(object)	Zabbix.log(3, JSON.stringify(object))
console.error(object)	Zabbix.log(2, JSON.stringify(object))

  - HttpRequest
  - This object encapsulates a cURL handle and allows simple HTTP requests to be made
  - Errors are thrown as exceptions
  - Full List of Objects:
  - https://www.zabbix.com/documentation/current/en/manual/config/items/preprocessing/javascript/javascript_objects

initMAX

JavaScript usage locations

- **Webhooks**
  - Primary Use – Integration with External Systems (ITSM / CHAT / etc)
  - Key object – the built-in HttpRequest() used to send and receive data
  - Context – Typically triggered by Action Operations
- **Preprocessing**
  - Primary Use – Transform raw data into a structured format
  - Capabilities – Advanced JSON/XML Manipulation, Complex math, Transformation
  - Context - Applied to Items or Low-Level Discovery rules to clean up "messy" API responses.
- **Script Items**
  - Primary Use – Custom data collection where standard Zabbix Agent fall short
  - Advantage – Allows for multi-step logic (getting auth.token, then querying data)

initMAX

Zabbix JavaScript Engine

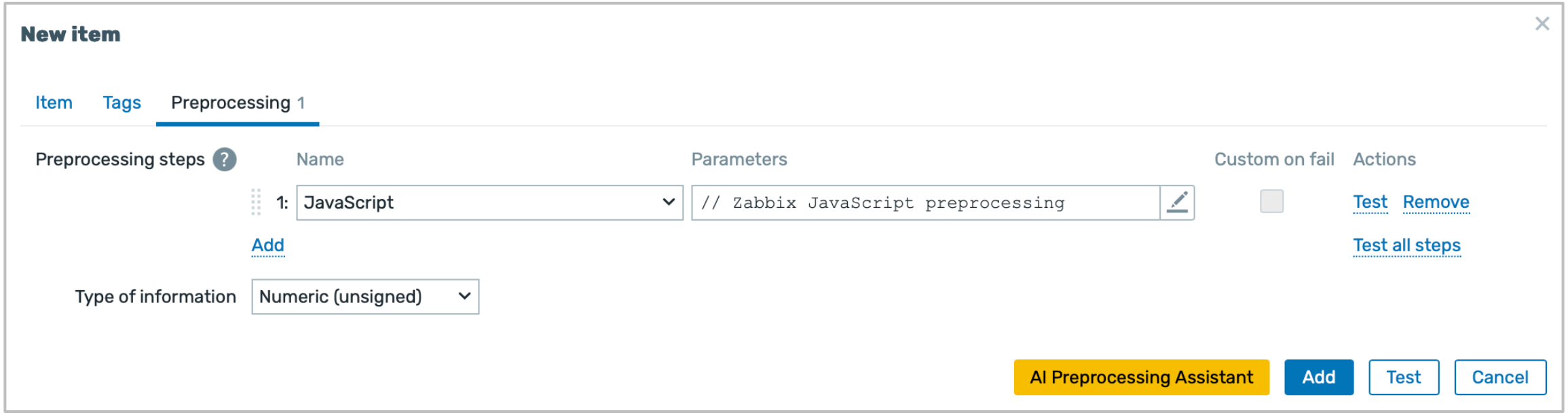
- Sandboxed execution environment
  - Designed for security, the engine does not have direct access to the underlying OS file system or network outside the provided built-in objects
- Built-In specialized objects
  - HttpRequest – Encapsulated cURL handles to perform HTTP requests
  - Zabbix.log – Native logging for debugging
- Error Handling
  - The engine uses standard JavaScript exception handling: errors are thrown as exceptions
- Debugging Capabilities
  - Supports common aliases like console.log()

initMAX

Preprocessing

Custom Script

- JavaScript



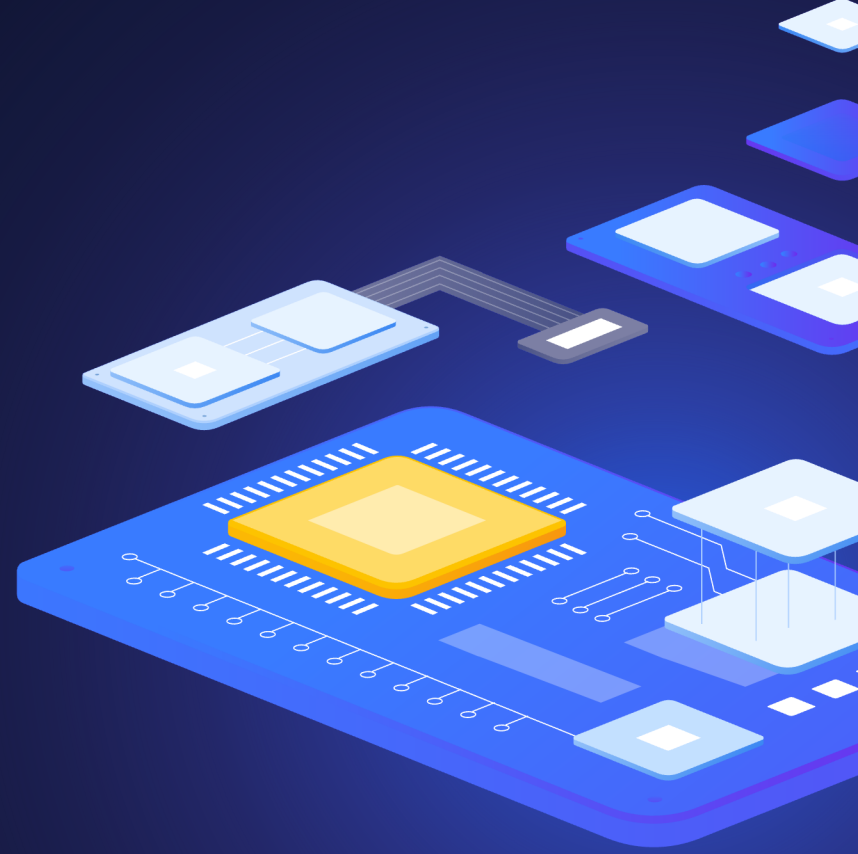
initMAX

Preprocessing JavaScript

- Parsing API Data
  - Convert raw, complex responses from external systems into readable metrics
  - Ideal for scenarios where a single API call returns data for multiple items at once\
- JSON & Structure Manipulation
  - Data Cleanup: Flattening nested JSON objects or filtering out unnecessary arrays before storage.
  - Format Conversion: Transforming XML or unstructured text into clean JSON for easier processing in later steps.
- Value transformations
  - Handles complex math, conditional logic ("if/else"), and string manipulations that standard Zabbix preprocessing steps cannot.
  - Combining multiple received values into a single summary string or calculated metric.

4

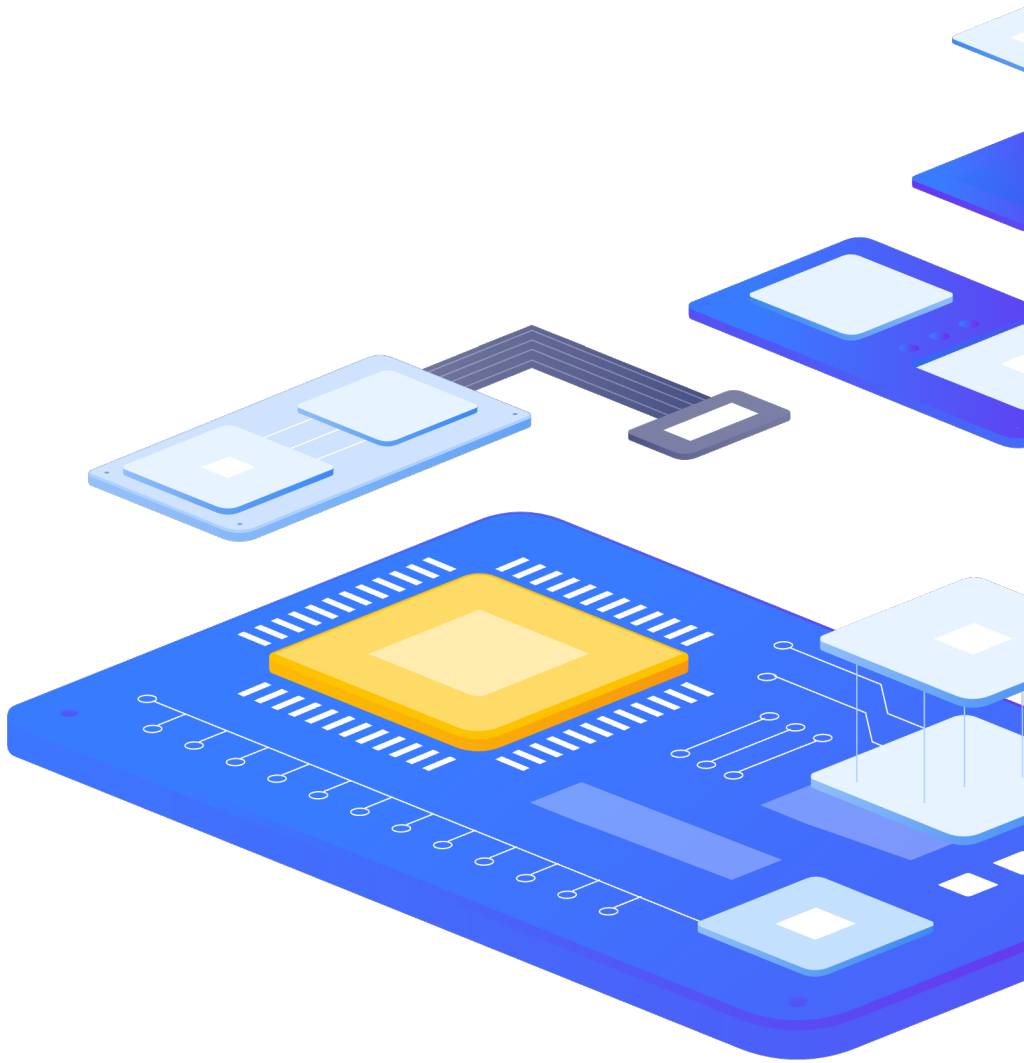
Automation Examples



initMAX

Example 1 - Manual Host action

- Problem:
  - Manually start and stop httpd service
- Solution:
  - Create Host script – stop httpd service
  - Create Host script – start httpd service

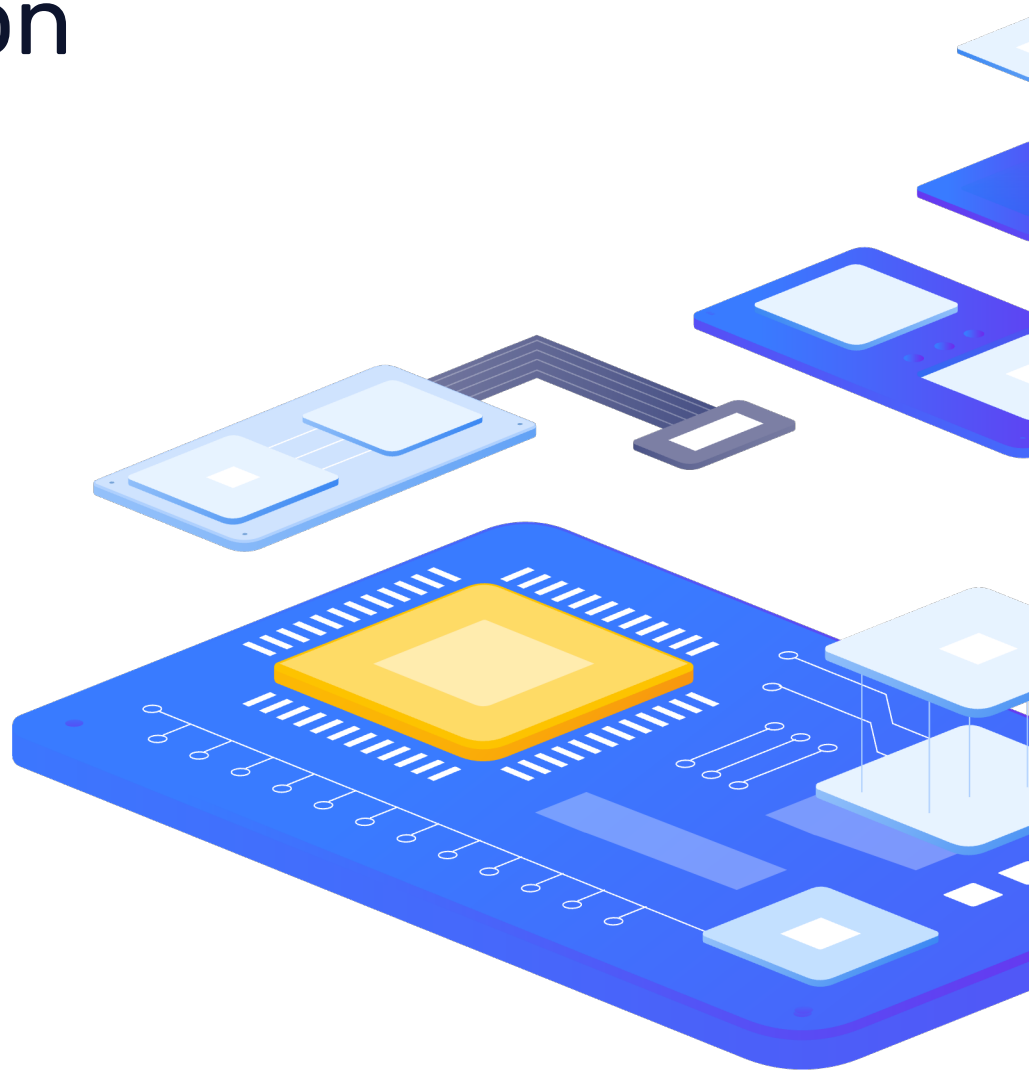


initMAX

Example 2 – Manual Event action

- Problem:
  - Manually create Ticket for problem
- Solution:
  - Create Event script – Webhook

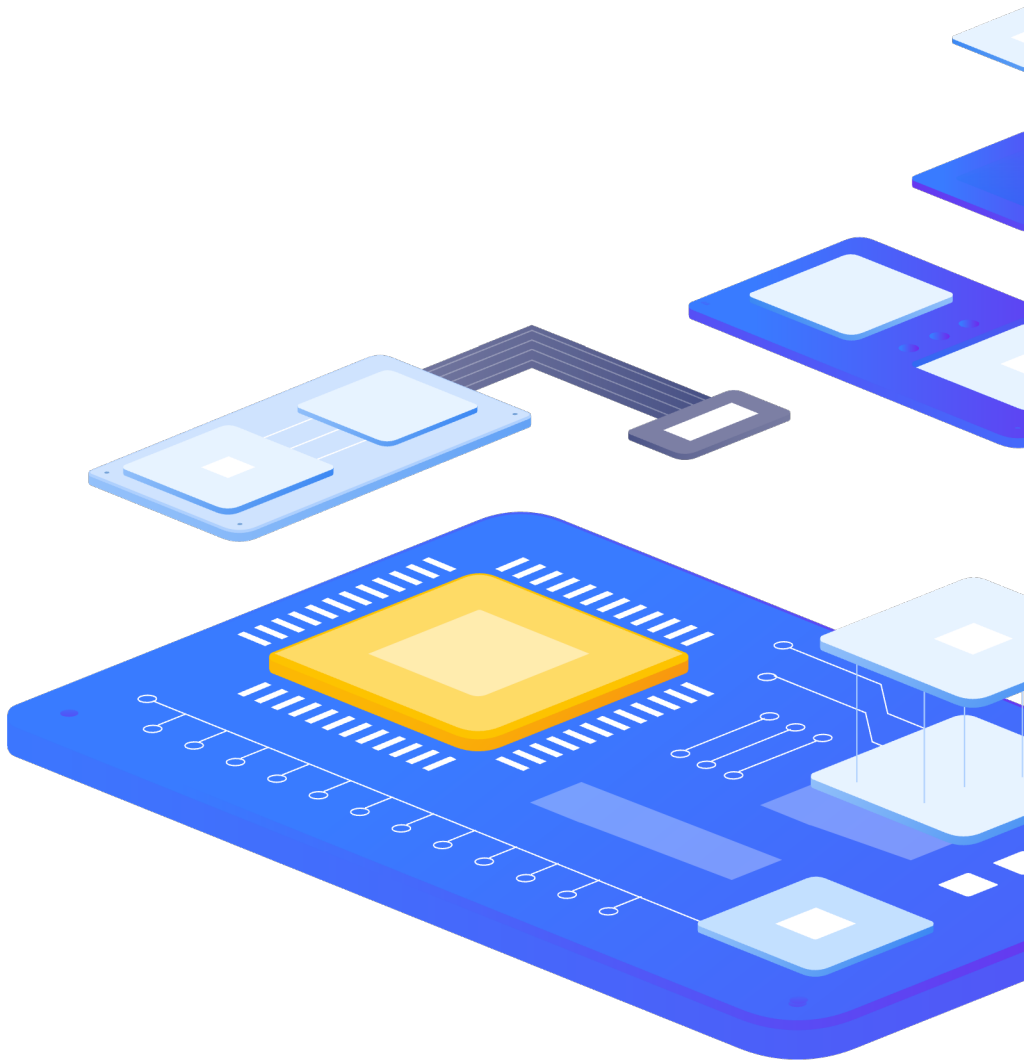
- Problem:
  - Manually start powered-off workstation
- Solution:
  - Create Event script – Wake on LAN



initMAX

Example 3 – Action operation

- Problem:
  - Automatically start stopped service
- Solution:
  - Create Trigger - Httpd service not running
  - Create Action – Execute script
  - Create Script - Start service



initMAX

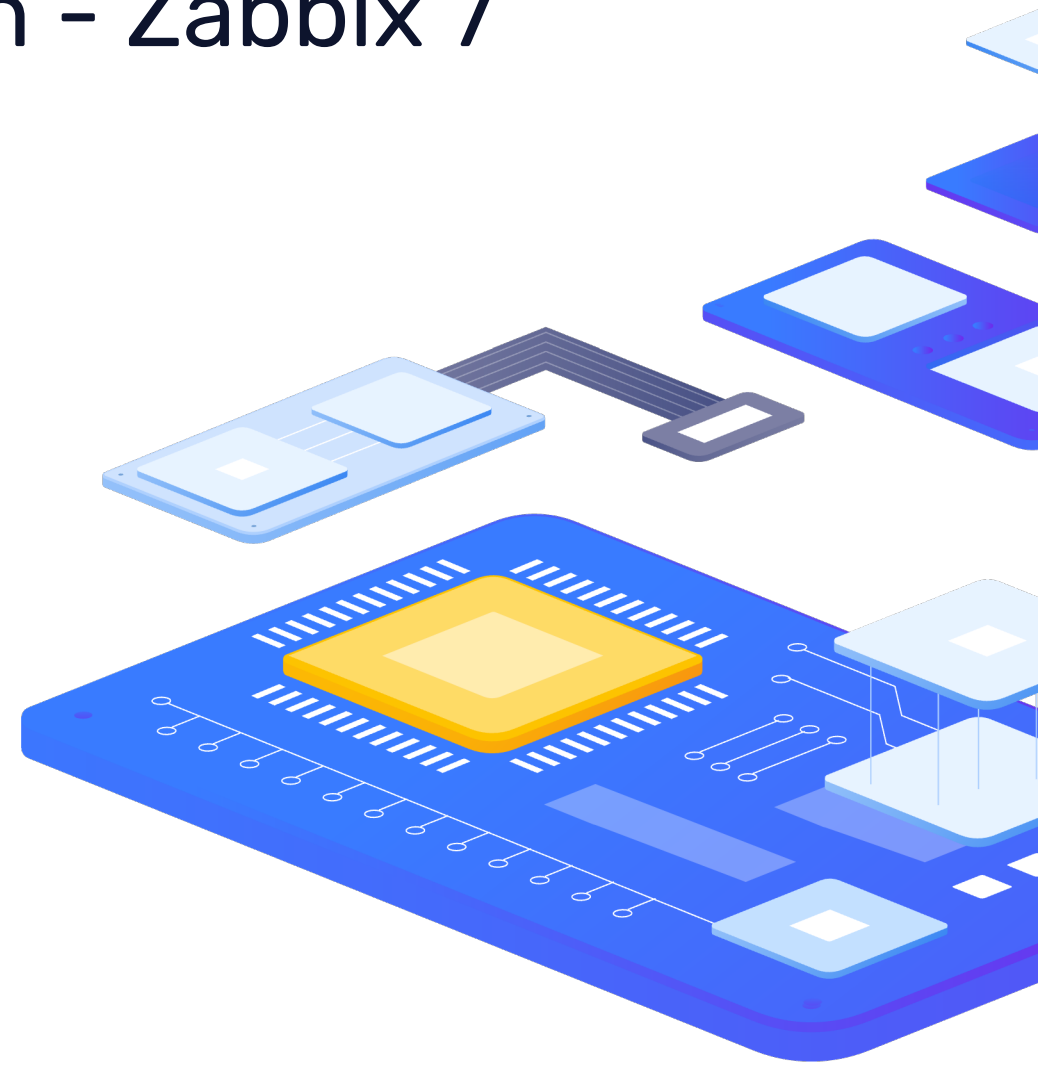
Example 4 – DNS HOST Name settings

- Problem:
  - Agent autoregistration updates workstation interface settings
  - Dynamic IP addresses
- Solution:
  - Change host interface setting based on hostname

initMAX

Example 5 – Manual Host action - Zabbix 7

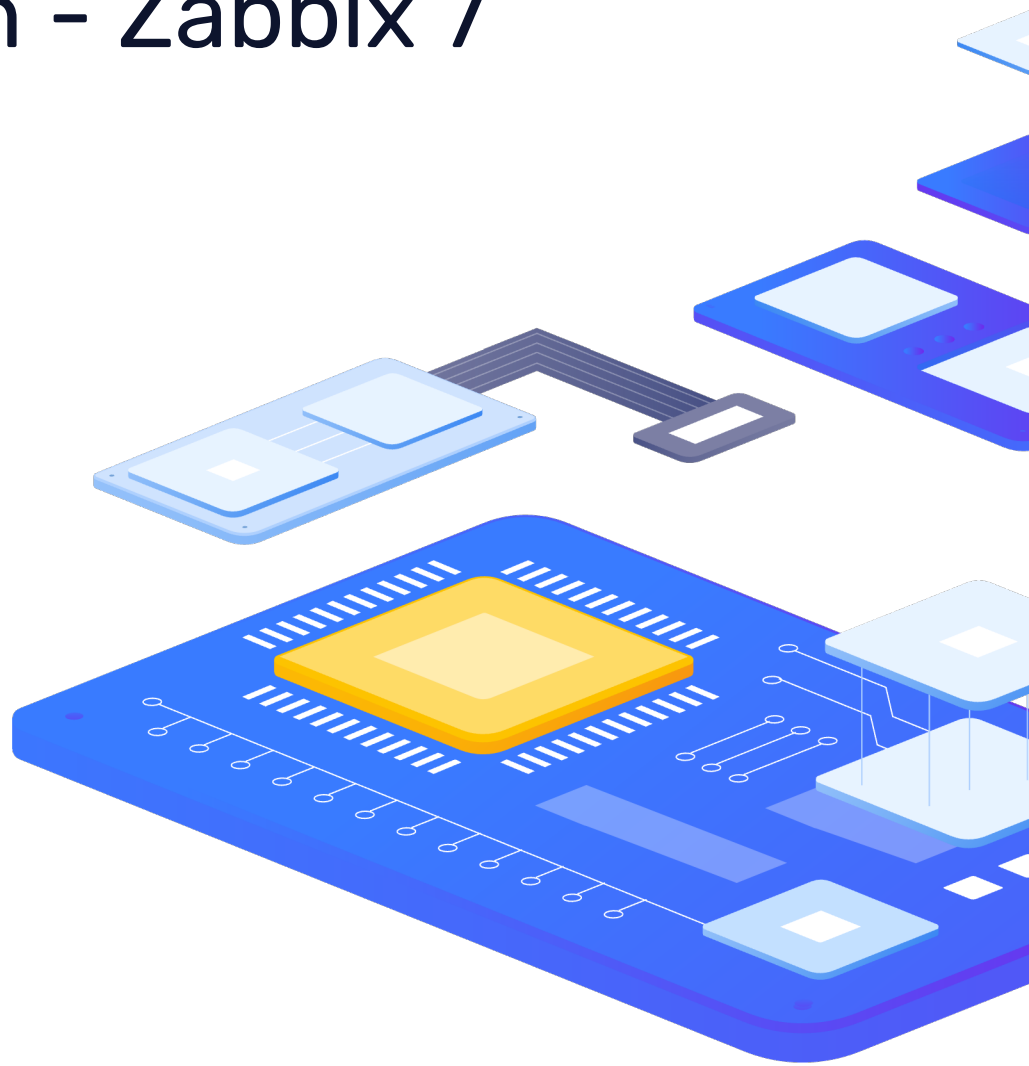
- Problem:
  - Allow to restart list of services
- Solution:
  - Agent script with dropdown menu



initMAX

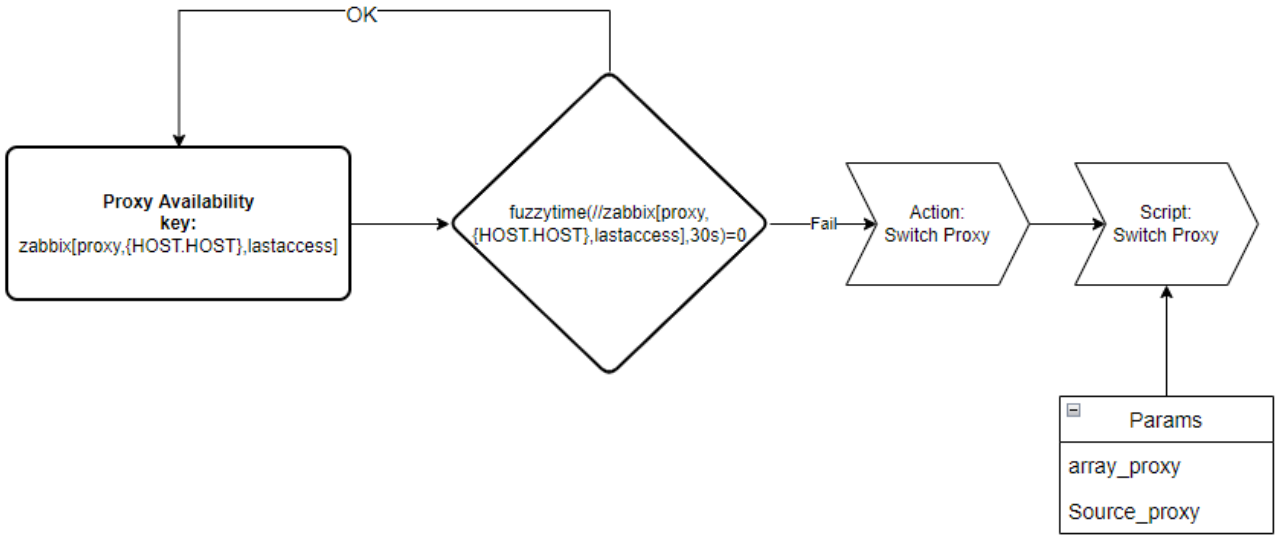
Example 6 – Manual Host action - Zabbix 7

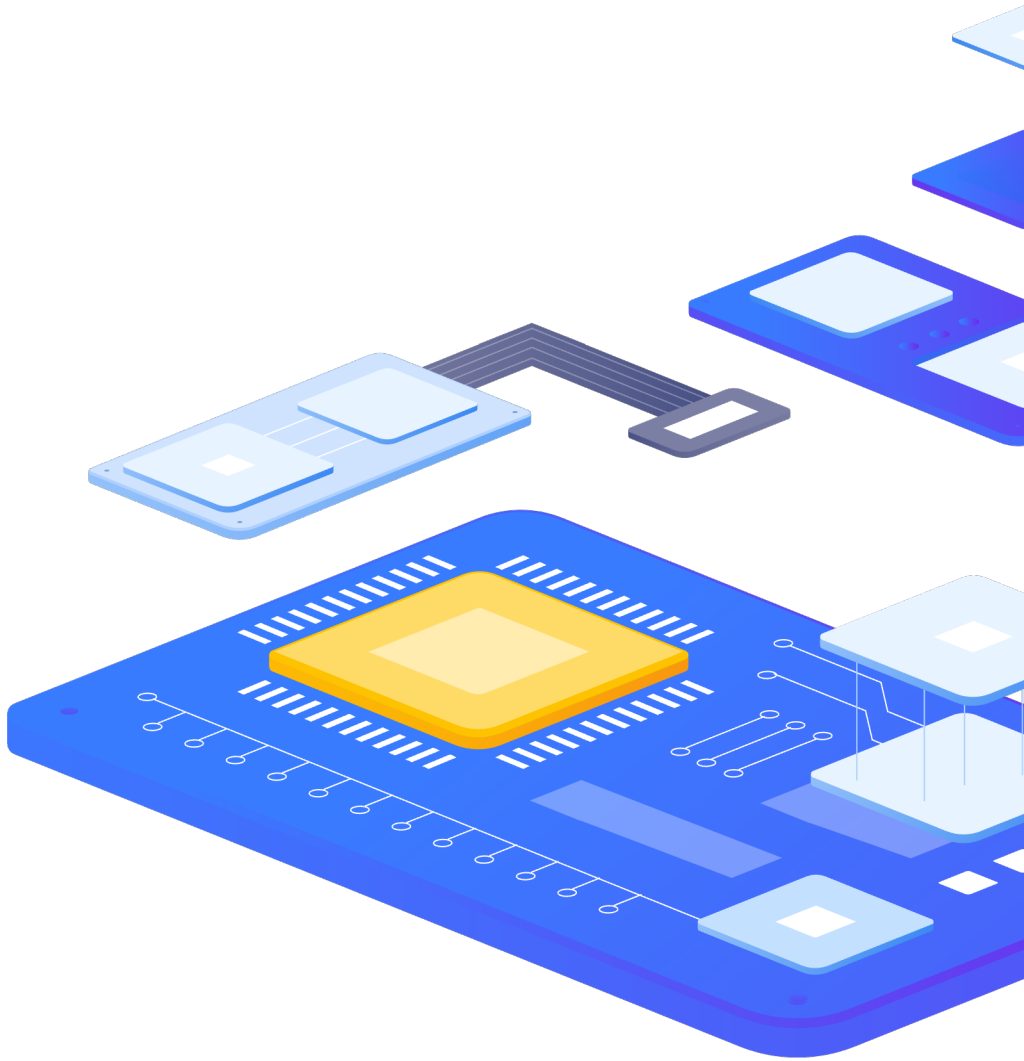
- Problem:
  - Manual data input
- Solution:
  - Webhook script with API call history.push



Example 7 – Proxy failover

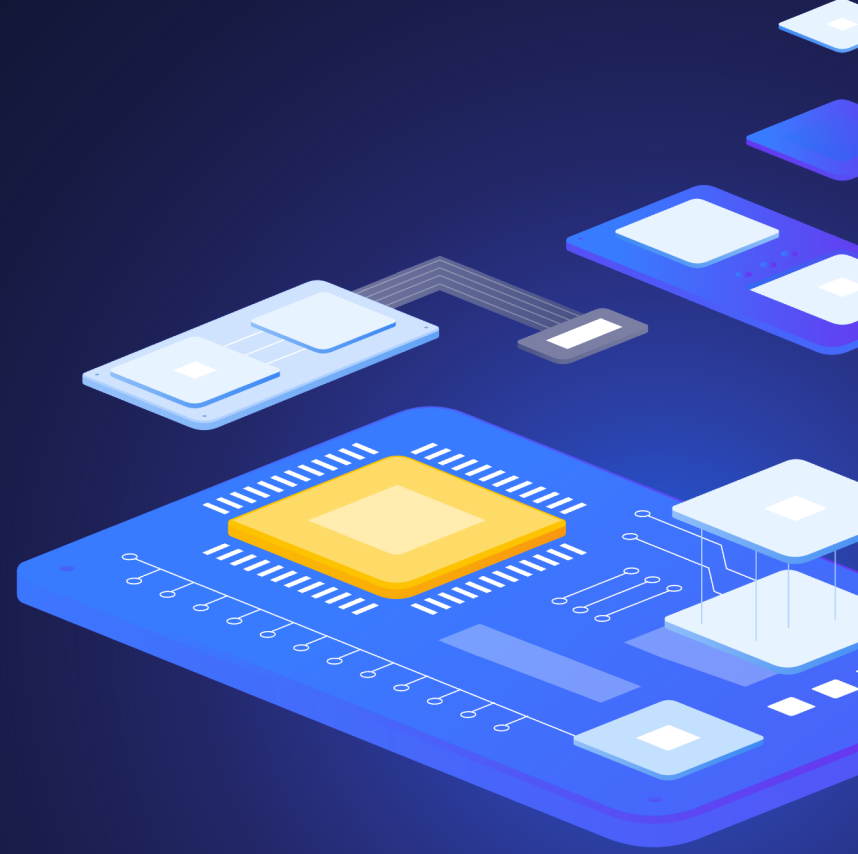
- Problem:
  - HA for Zabbix proxy needed by customer for SNMP devices
- Solution:
  - Use two Zabbix proxies as a “cluster”
  - Script for Failover based on availability of proxy





4

Best Practices in Script usage



initMAX

Security Best Practices

- Restrict system.run on Zabbix agents
  - Use AllowKey and DenyKey parameters in the agent configuration to whitelist specific commands and block everything else.
  - Avoid using broad wildcards like system.run[*] which can lead to unauthorized remote command execution
  - Ensure proper agent user rights on the system to limit the impact if a command is compromised
- Limit scripts by User groups
- Confirmation dialogs
  - Enable confirmation dialogs for critical tasks like service reboots to prevent "accidental clicks"

initMAX

Performance Best Practices

- Avoid heavy scripts
  - JavaScript execution is sandboxed and limited to a specific memory and timeout window to prevent scripts from hanging the main process.
  - Avoid deep nested loops or processing massive datasets (e.g., multi-megabyte JSON files) within a single preprocessing step.
  - If a script requires significant computation, consider moving that logic to an external script triggered via the agent or server rather than using internal JavaScript
- Use Zabbix proxies
  - Execute scripts closer to the remote infrastructure to reduce latency and offload the CPU burden from the central Zabbix Server.
- Logging
  - excessive use of console.log() or Zabbix.log() can bloat server logs and impact I/O performance.
  - Ensure all debugging code is removed or commented out before moving scripts into production environments

initMAX

Script Maintainability

- Naming Conventions
  - Organize scripts by type and function (e.g., Linux/Service/Restart, API/Jira/CreateTicket) to keep the "Scripts" menu clean.
- Version Control
  - Do not treat the Zabbix Frontend as your only source of truth; maintain your JavaScript and shell scripts in a version control system like GitLab or GitHub.
- Script Documentation
  - Document complex regex expressions and custom JavaScript objects (like HttpRequest) directly within the script body

initMAX

Tips and tricks

Check our wiki and social networks regularly for tips and updates

Tips and tricks on our webpage:

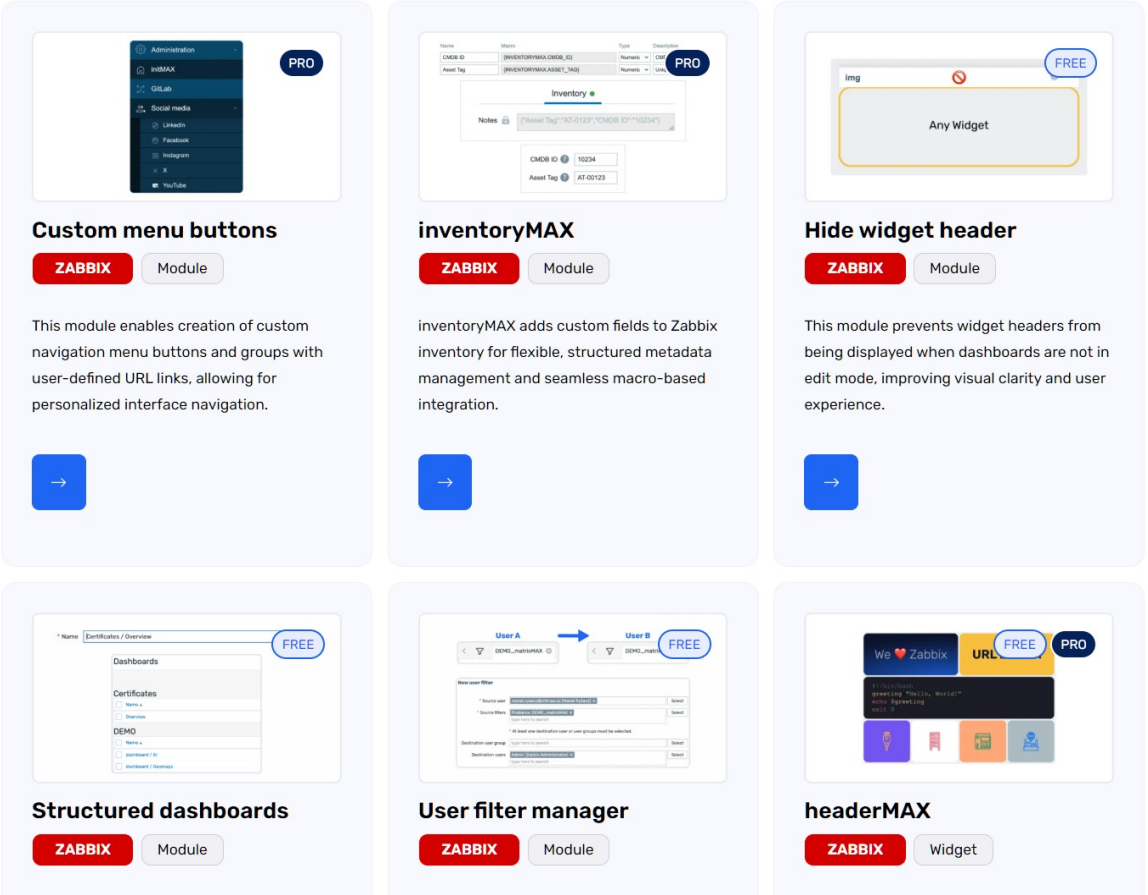
- https://www.initmax.com/wiki/frontend-scripts-and-sudo-in-zabbix/
- https://www.initmax.com/wiki/zabbix-java-gateway-installation-with-tomcat-monitoring/
- https://www.initmax.com/wiki/zabbix-7-0-instructions-for-installation-in-5-minutes/
- https://www.initmax.com/wiki/zabbix-7-0-and-increasing-system-limits/
- https://www.initmax.com/wiki/zabbix-migration-from-mysql-to-postgresql/
- https://www.initmax.com/wiki/how-to-set-up-snmp-trap-in-zabbix/
- https://www.initmax.com/wiki/microsoft-teams-integration-in-five-steps/
- https://www.initmax.com/wiki/reporting-in-zabbix-7-0/

Frontend Scripting

initMAX E-Shop

- Custom visualization widgets
- UX Improvement Modules
- AI Integration with Zabbix
- Both **FREE** and PRO Versions

- initMAX eshop link
  https://www.initmax.com/eshop/



- Video explanation of widgets and modules
  https://www.youtube.com/watch?v=fpW6TR7DQdU&list=PLF7Hh_ikyQDpHiHhXwLtw57OCDF9jn7zL

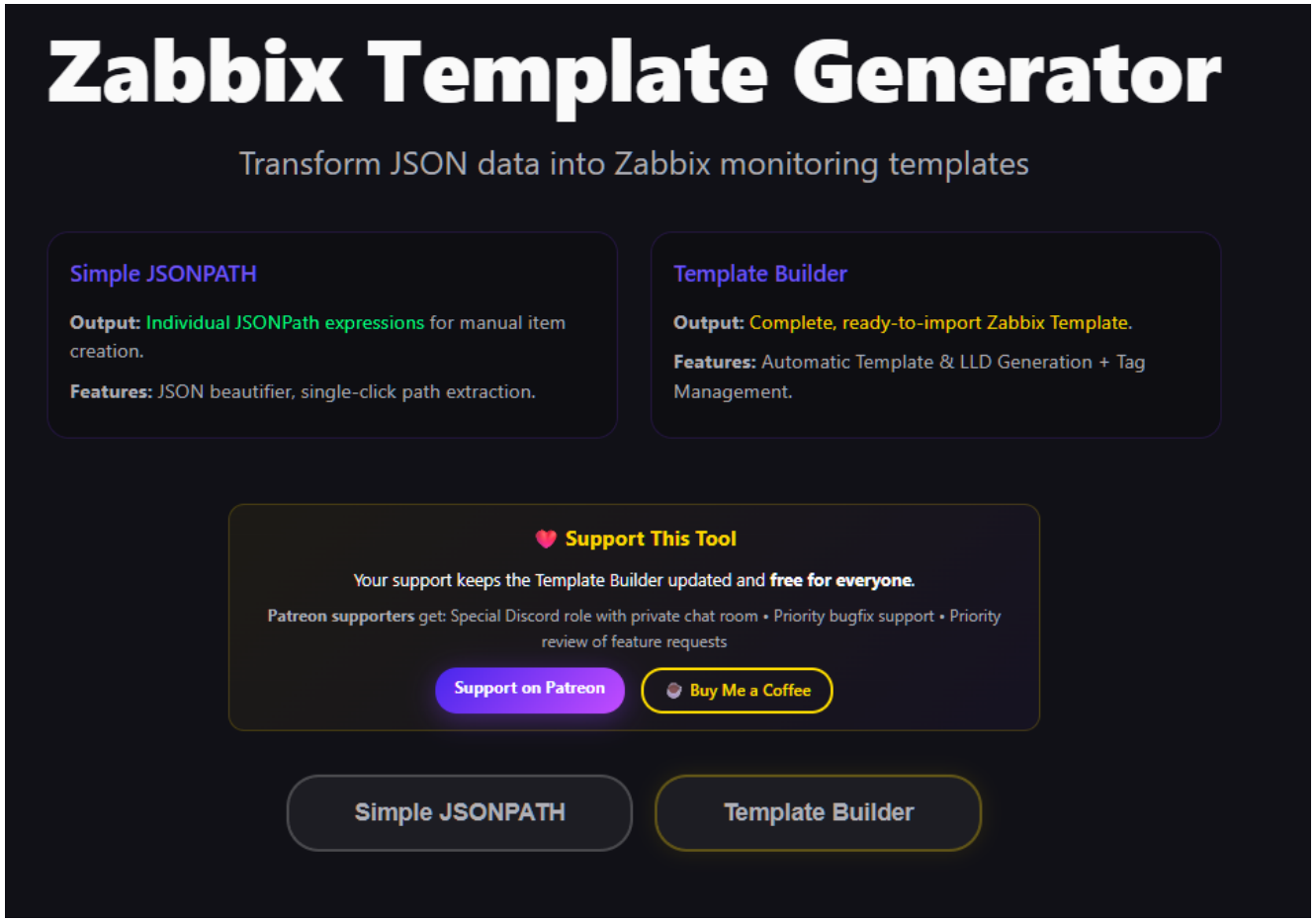
initMAX

Frontend Scripting

dmitrylambert.com

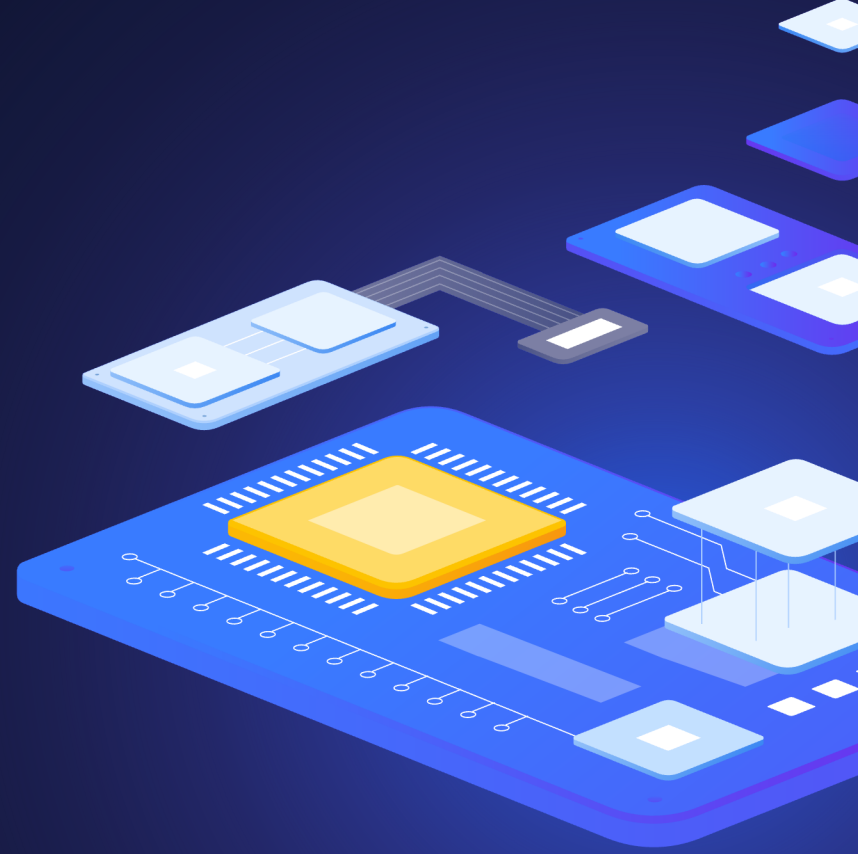
- FREE Template builder out of JSON
- Tag support
- Low Level Discovery support
- Zabbix 7.0 / 7.2 / 7.4 version support
- Extract Simple JSONPATH
- Download ready to apply Template

- www.dmitrylambert.com



initMAX

Questions?



Frontend Scripting

initMAX

Contact us:

Phone:	+420 800 244 442
Web:	https://www.initmax.com
Email:	tomas.hermanek@initmax.com
LinkedIn:	https://www.linkedin.com/company/initmax
Twitter:	https://twitter.com/initmax
Tomáš Heřmánek:	+420 732 447 184