

Self-healing with Checkmk and Event-Driven Ansible

How to resolve issues automatically

About me

- René Koch
- Self-employed consultant for:
 - Red Hat Ansible (Automation Platform)
 - Red Hat Enterprise Linux
 - Red Hat Satellite
 - Red Hat Identity Management (IPA)
- Experienced monitoring user (Nagios, Icinga, Checkmk)



About me

- René Koch
 - rkoch@rk-it.at
 - +43 660 / 464 0 464
 - <https://www.linkedin.com/in/rk-it-at>
 - <https://github.com/rk-it-at>
 - <https://github.com/scrat14>



Agenda

- Monitoring: Short introduction
- What is Event-Driven Ansible (EDA)?
- Event-Driven Workflow
- Live Demonstration
- Use cases and best practices

Monitoring: Typical workflow

- 🕒 2005: Received email alerts from Nagios 2 for issues with Solaris machines
- 🧩 Manual workflow:
 - 📧 Read email
 - 🗝️ Log in to the system
 - 🔍 Check if issue still exists
 - 🛠️ Fix the issue
 - 🔄 **Repeat the same procedure over and over again**
- 🕒 2026: Still the same workflow?

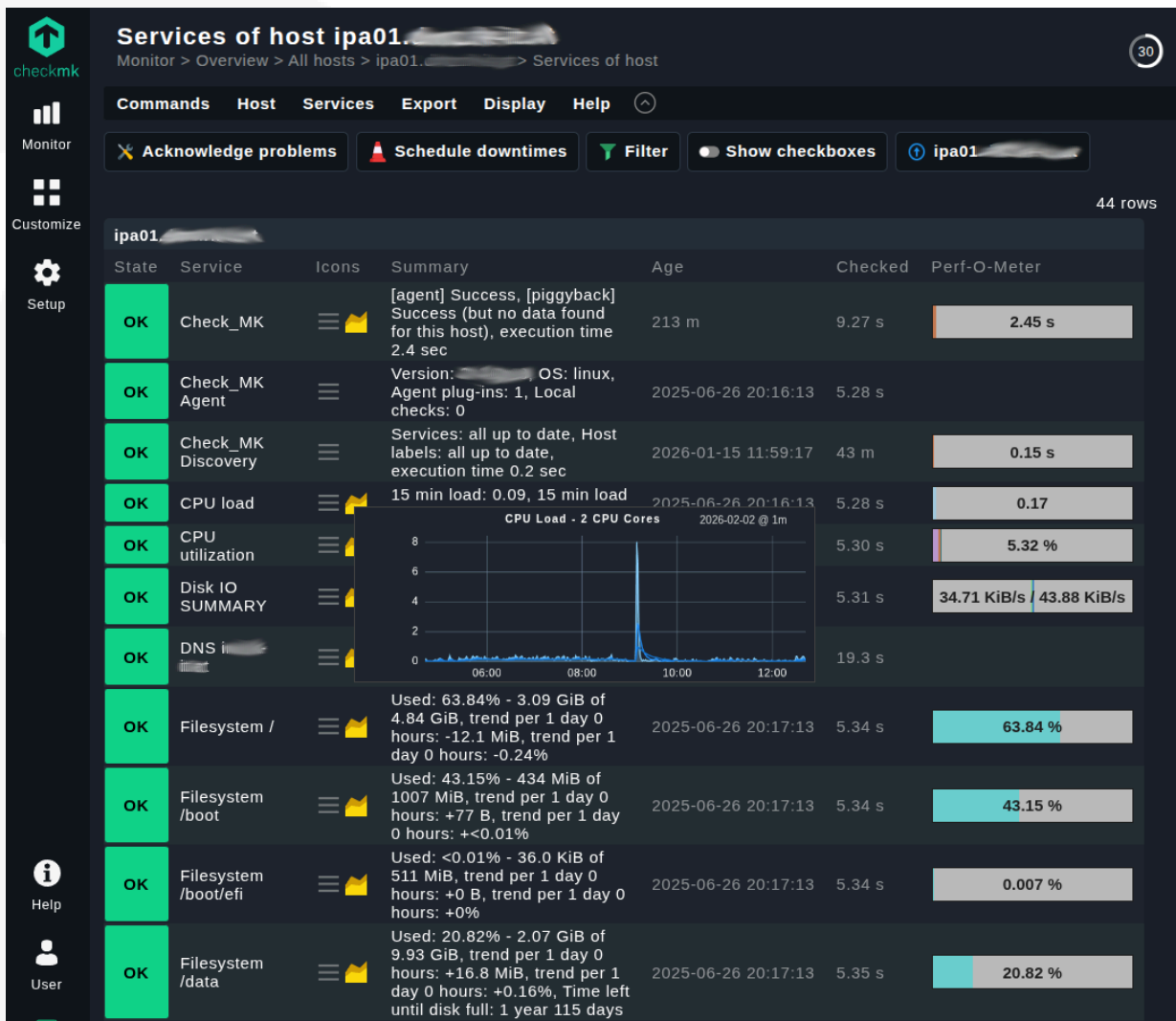


Automation happens when
one person meets a problem
they never want to solve again

Source: https://github.com/ansible/workshops/blob/devel/decks/ansible_rhel.pdf

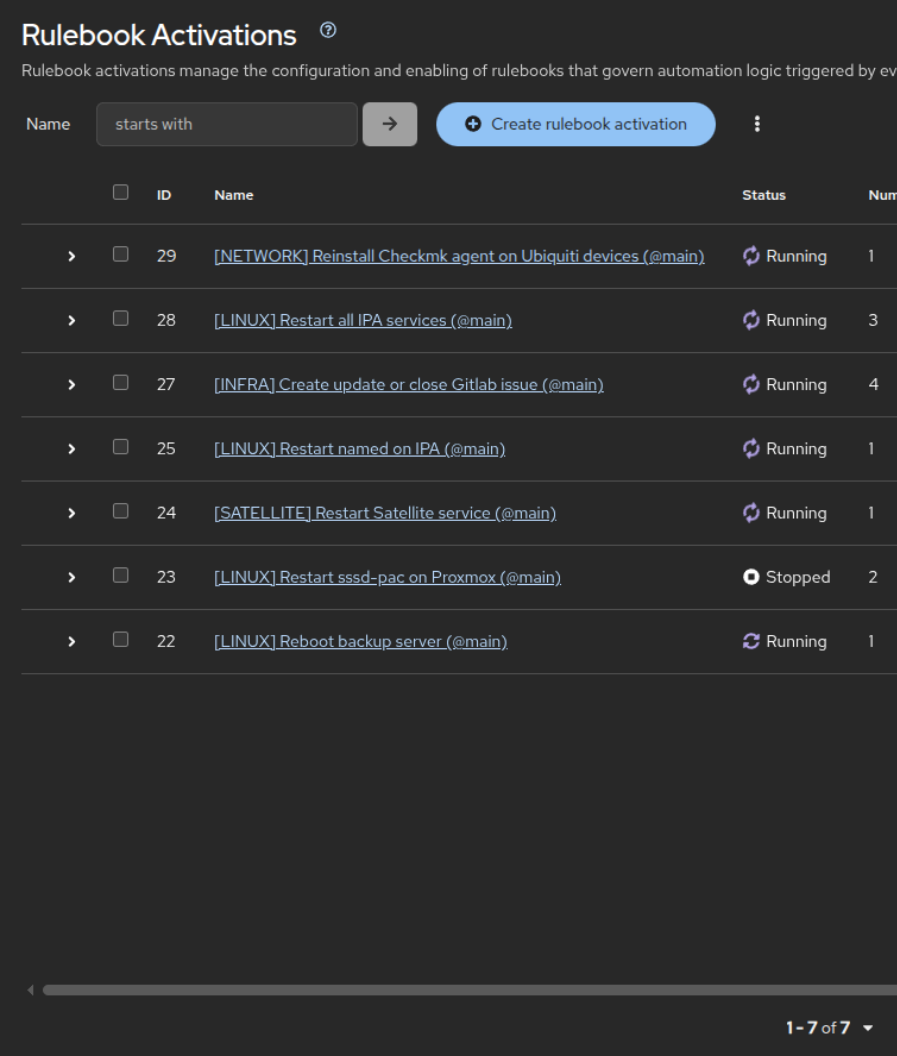
What is Checkmk?

- **Monitoring platform** for infrastructure, applications and services
- Provides **agent- and agentless** checks, dashboards, and alerting
- Built for **scale** with distributed monitoring and automation support
- Helps **detect, analyze, and remediate** issues faster



What is Event-Driven Ansible?

- EDA is automation that reacts to events, not schedules
- Events can come from monitoring, webhooks, message queues, logs or cloud services
- Rules decide **when** to run Ansible actions
- Goal: **faster response** and **consistent remediation**



Name	ID	Name	Status	Num
starts with				
	29	[NETWORK] Reinstall Checkmk agent on Ubiquiti devices (@main)	Running	1
	28	[LINUX] Restart all IPA services (@main)	Running	3
	27	[INFRA] Create update or close Gitlab issue (@main)	Running	4
	25	[LINUX] Restart named on IPA (@main)	Running	1
	24	[SATELLITE] Restart Satellite service (@main)	Running	1
	23	[LINUX] Restart sssd-pac on Proxmox (@main)	Stopped	2
	22	[LINUX] Reboot backup server (@main)	Running	1

What Is an "Event" (vs a Source Action)?

- ⚡ **Event**
 - a *state change* or *signal* that matters (e.g., alert fired, service down)
 - often noisy and hard to filter
 - not every event triggers an action
- 🔄 **Source action**
 - a *routine trigger* (e.g., "on every commit")
 - predefined target/action
- 🩺 Examples:
 - ❌ *Update an AAP project after each commit* (not EDA)
 - ✅ *Send all monitoring alerts to a webhook; EDA decides what to do* (EDA)






Event-Driven Ansible vs. Ansible Playbook

- ⚡ **EDA** keeps a listener running and reacts to events in near real time
- 🧠 **EDA** evaluates event payloads and triggers automation only when rules match
- 📄 **Ansible Playbooks** do not listen for events out of the box
- 🔔 Without EDA, the monitoring source must trigger playbook runs directly
- 🧱 This increases load and complexity on the monitoring system

Event-Driven Ansible vs. AAP Controller

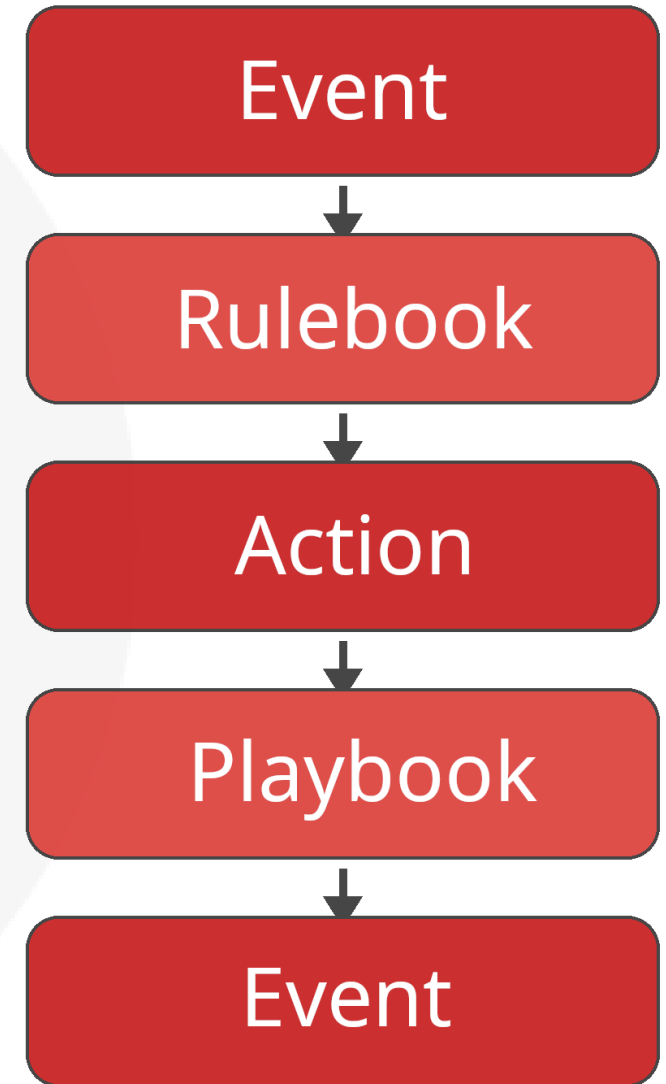
- ⚡ **EDA** keeps an event listener active and can trigger job templates immediately
- 🌐 **AAP Controller** can start jobs via webhook or API, but each job is a full run lifecycle
- 🐢 Job startup overhead (container start, project sync, inventory/collections) adds latency
- 🚦 Job execution may queue behind other jobs, which delays reaction time
- 🎯 Use EDA for fast event decisions; use Controller for governed execution of the actual remediation

Core Building Blocks

-  **Event Sources**: where events originate (ansible.eda plugins for webhooks, Kafka, Altermanager, etc.)
-  **Rulebook**: Rulesets with conditions + actions
-  **Conditions**: Determine if a rule fires
-  **Actions**: run playbooks, run job templates, run modules, etc.
-  **Controller** (optional): central execution and governance

Event-Driven Workflow

1. Event arrives from a source
2. Rulebook evaluates conditions
3. Matching rule triggers an action
4. Action runs playbook or other automation
5. Results can emit **new events** or update systems



Rulebook: Restart named

```
---

- name: Restart named on IPA server
  hosts: all
  gather_facts: false

  sources:
    - name: Listen on port 5000 for Checkmk events
      ansible.eda.webhook:
        port: 5000

  rules:
    - name: Restart named
      condition: >-
        event['payload']['servicename'] == "DNS example.com" and
        event['payload']['servicestate'] == "CRITICAL"
      action:
        run_job_template:
          name: "[LINUX] Restart named on IPA [@production] - Prompt"
          organization: "Default Organization"
          job_args:
            limit: "{{ event.payload.hostname }}"
```


Checkmk Notification Script

```
#!/usr/bin/env bash

HEADER="X-Checkmk-Token"
TOKEN="${NOTIFY_PARAMETER_1}"
URL="${NOTIFY_PARAMETER_2}"

JSON=`cat <<EOF
{
  "hostname": "${NOTIFY_HOSTNAME}",
  "hostoutput": "${NOTIFY_HOSTOUTPUT}",
  "hoststate": "${NOTIFY_HOSTSTATE}",
  "servicename": "${NOTIFY_SERVICEDESC}",
  "serviceoutput": "${NOTIFY_SERVICEOUTPUT}",
  "servicestate": "${NOTIFY_SERVICESTATE}",
  "date": "${NOTIFY_SHORTDATETIME}",
  "type": "${NOTIFY_NOTIFICATIONTYPE}",
  "what": "${NOTIFY_WHAT}"
}
EOF
`

curl -X POST -H "Content-Type: application/json" -H "${HEADER}: ${TOKEN}" -d "${JSON}" ${URL}
exit $?
```

Playbook: Restart named

```
---  
  
- name: Restart named on IPA  
  hosts: all  
  become: true  
  gather_facts: true  
  
  tasks:  
    - name: Restart named  
      ansible.builtin.service:  
        name: named  
        state: restarted
```

Run a Rulebook (CLI)

```
$ ansible-rulebook -r rulebooks/restart_named.yml -i localhost

PLAY [Restart named on IPA] *****






TASK [Gathering Facts] *****
ok: [ipa01.example.com]

TASK [Restart named] *****
changed: [ipa01.example.com]

PLAY RECAP *****
ipa01.example.com : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

! Use the **run_playbook** action (instead of **run_job_template**) when running with ansible-playbook outside Ansible Automation Platform.

Ansible Automation Platform Integration

-  **Projects:** Git repository configuration
-  **Decision Environments:** Container images to run rulebooks
-  **Credentials:** Secrets for Git, Controller, Hub, tokens, etc.
-  **Event Streams:** Entry points for events (mapped to source definition in rulebook)
-  **Rulebook Activations:** Rulebook runs

[Rulebook Activations](#) > Edit [LINUX] Restart named on IPA (@main)

Edit [LINUX] Restart named on IPA (@main)

Name *

[LINUX] Restart named on IPA (@main)

Description

Enter description

Organization * **Project ***

Rene Koch IT Project: Rulebooks Linux (@main)

Rulebook * **Event streams ?**

linux_ipa_restart_named.yml Webhook: Checkmk Notifications ⌕

Credential ? **Decision environment * ?**




AAP: Controller ⌕ Default Decision Environment

Restart policy * ? **Log level * ?**

Always Info

Rulebook activation enabled? ?

☐

Variables ?    **YAML** **JSON**

Options

☐ Skip audit events ?

Save rulebook activation Cancel



Live Demo: Fix DNS issue

Self-Healing Best Practices

- Start with **low-risk** automations
- Use **idempotent** playbooks (if possible)
- Add **guardrails** (approvals, maintenance windows, downtimes)
- Emit **metrics and logs** for auditing






✓ Low risk

✓ Idempotent






✓ Guardrails

✓ Auditing

Challenges with Self-healing

-  Triggering on noisy events (missing filtering)
-  Insufficient monitoring coverage
-  Healing the wrong host (issue caused by a backend dependency)
-  Lack of knowledge or runbooks
-  Triggering during maintenance windows due to missing downtime

Event-Driven Ansible Use Cases

-  **Monitoring alerts:** run remediation playbooks
-  **Infrastructure events:** auto-scale or restart services
-  **Security findings:** isolate hosts or rotate credentials
-  **Ticketing:** enrich and open incidents automatically
-  **Documentation:** update asset database or documentation system

Additional Information





- **Products:**

- Ansible Automation Platform: <https://urlr.me/WcXZgR>
- Checkmk: <https://urlr.me/BPJs98>

- **Product Documentation:**

- Ansible: <https://urlr.me/2x4HfW>
- Automation Decisions: <https://urlr.me/HejEDB>
- Rulebooks: <https://urlr.me/Qb4TsB>

Summary

-  Checkmk **monitors** your IT landscape and **notifies** on state change
-  EDA turns these events into **real-time automation**
-  It complements traditional Ansible by **reacting** instead of **scheduling**
-  Start small, measure impact, and iterate

Thank you!

René Koch

Freelancer

Ansible Anwendertreffen Austria 18.02.2026

Slides: <https://urlr.me/zCPcuK>

