

Red Hat
Summit

Connect

Ansible and Event-Driven automation for networks

Michal Zdyb,
RHCA, Senior Product Specialist Solution Architect - Ansible

Introduction to Network Automation with Ansible

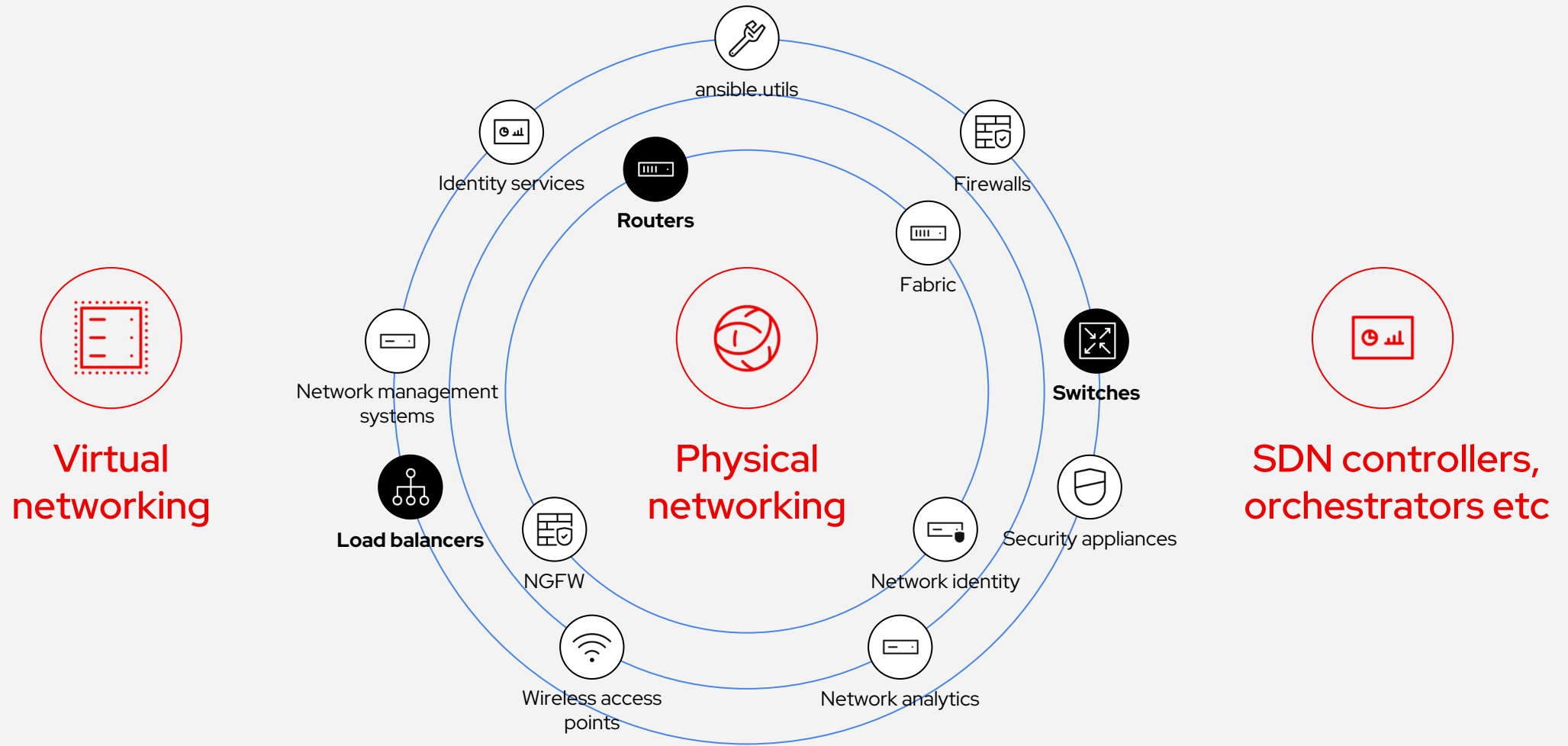
Why automation?

- more work done in less time (increased efficiency)
- reduced errors and risks associated with routine tasks
- more time to focus on high-value initiatives
- becomes strategic decision for organizations
- critical factor of digital transformation

Automation-first culture saves time and money while fostering learning, experimentation and innovation

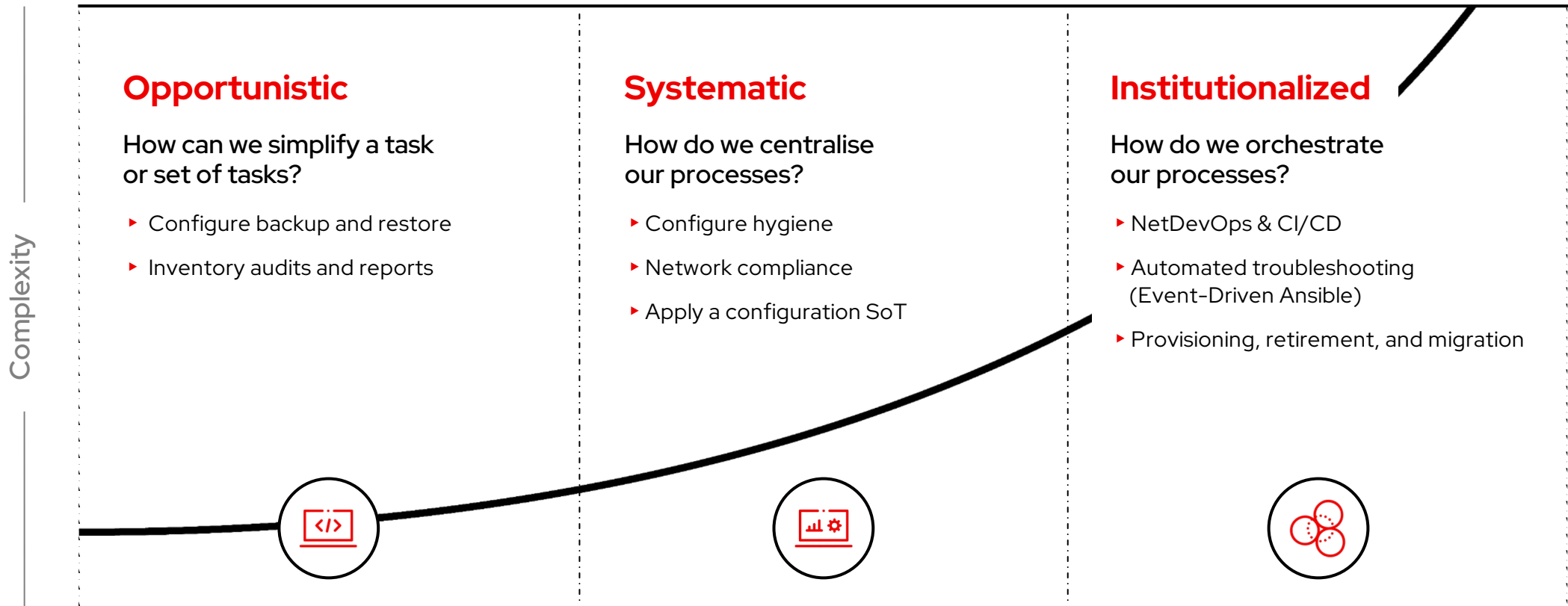


The scope of network automation with Ansible



Network automation journey

Start small, think big



Source of Truth

- Defines the **desired state** of network represented as **structured data**
- Decoupling data from configuration syntax
- Enables **data-driven** network automation
- Popular SoT for networks: Github, Netbox

Example of structured configuration data in yaml files on Github:

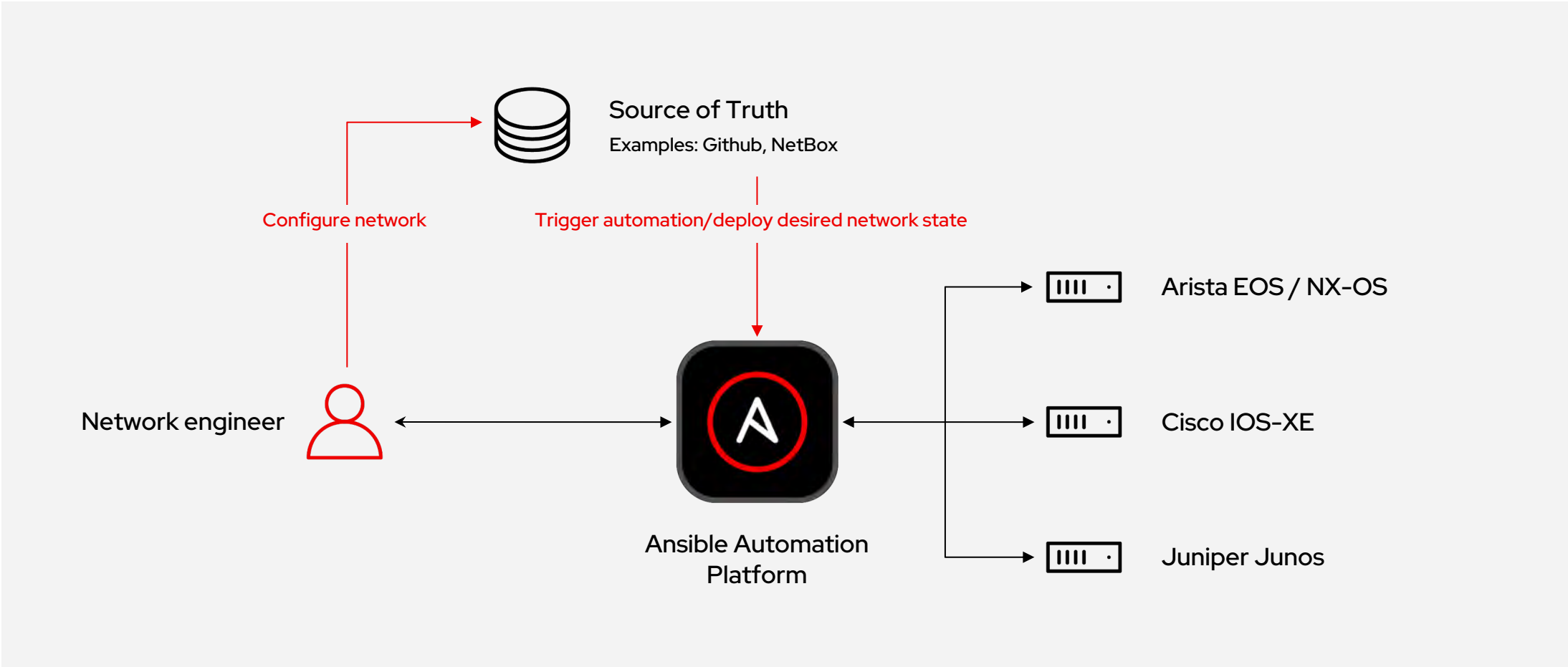
BGP config:

```
bgp:
  as: "64512"
  router_id: 172.16.0.1
  neighbors:
    - ip_address: 10.0.0.2
      remote_as: "64513"
      description: net01-rtr2
  networks:
    - prefix: 192.168.1.0/24
```

Interfaces config:

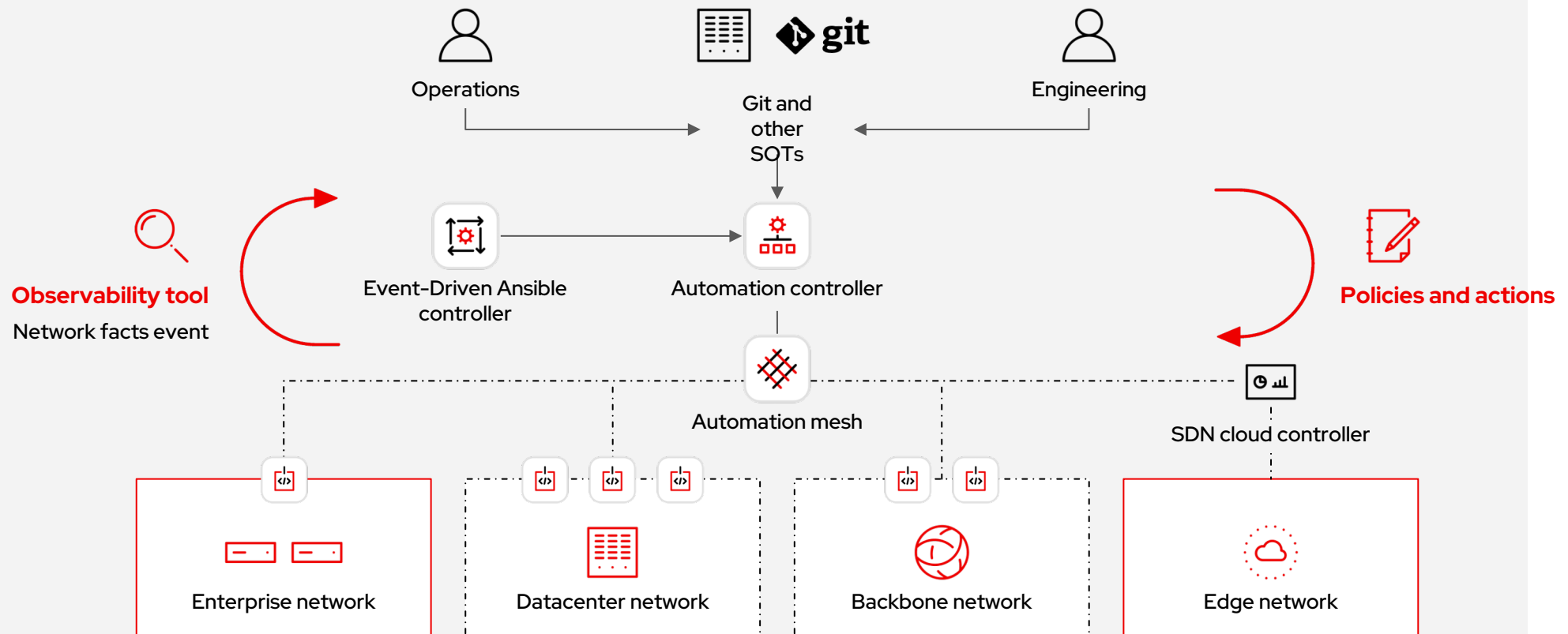
```
interfaces:
- name: Ethernet1
  description: to-net01-rtr2
  ipv4_address: 10.0.0.1/30
  switchport: false
  shutdown: false
- name: Ethernet2
  description: to-net01-client1
  ipv4_address: 192.168.1.1/24
  switchport: false
  shutdown: false
```

Automate configuration management with SoT



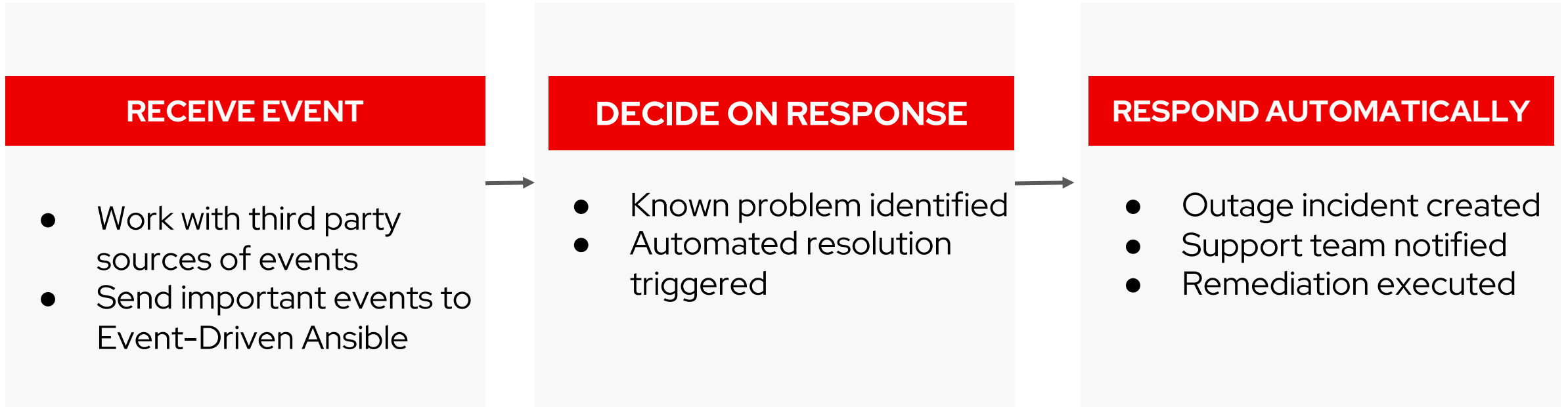
Network operations with Ansible Automation Platform

Source of Truth to operate with consistency and control



Introduction to Event-Driven Ansible

A typical event driven automation process



WORK ACROSS MULTI-VENDOR IT OPERATIONS

Work flexibly and well with multi-vendor monitoring and other solutions across the event driven architecture with appropriate approvals, controls and awareness

Key building blocks in Event-Driven Ansible

Simple, powerful, agentless



Sources

All the sources of event data you want to use



Rules

What you will create using Event-Driven Ansible®



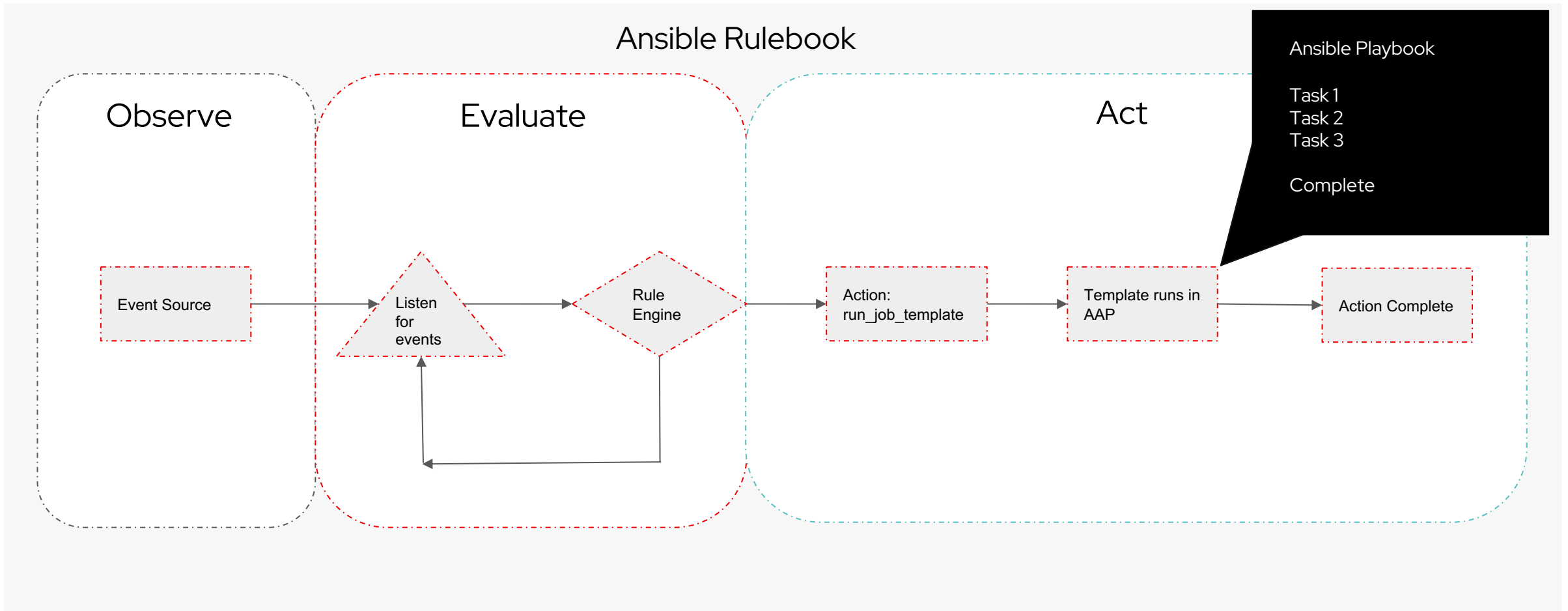
Actions

When a condition or event is met, the Ansible Rulebook executes

Ansible Rulebooks contain the source of the event, as well as the instructions on what steps to perform when a certain condition is met—and it is all very flexible.

Event-Driven Ansible

Rulebook vs Playbook



Rulebooks

Ansible Playbook has many plays, Ansible Rulebook has many rulesets

▶ Rulebooks comprise of rulesets

- ▶ Rulebooks can contain multiple Rulesets
- ▶ Rules trigger based on conditions and actions can be carried out by the rules engine
- ▶ Multiple sources can be defined in a Rulebook
- ▶ Rulebooks can have a similar structure to a Playbook with multiple plays.

```
- name: My ruleset 01
hosts: all
sources:
  - name: Range
    ansible.eda.range:
      limit: 5
rules:
  - name: First rule
    condition: event.i == 1
    action:
      debug:

- name: My ruleset 02
hosts: all
sources:
  - name: Kafka
    ansible.eda.kafka:
      host: 192.168.122.110
      port: 9092
      topic: network-events
rules:
  - name: First rule
    condition: event.interface.oper-status == "DOWN"
    action:
      run_job_template:
        - name: "Network EDA - interface recovery"
```

Rulebook Activation

Activated Automation

Rulebook Activations

Rulebook activations are rulebooks that have been activated to run.

ID	Name	Status
36	Automatic network issues recovery	Running

Rulebook Activated

Waiting for Events

Once Rulebooks are activated and running they are listening for events

Rulebook activation log:

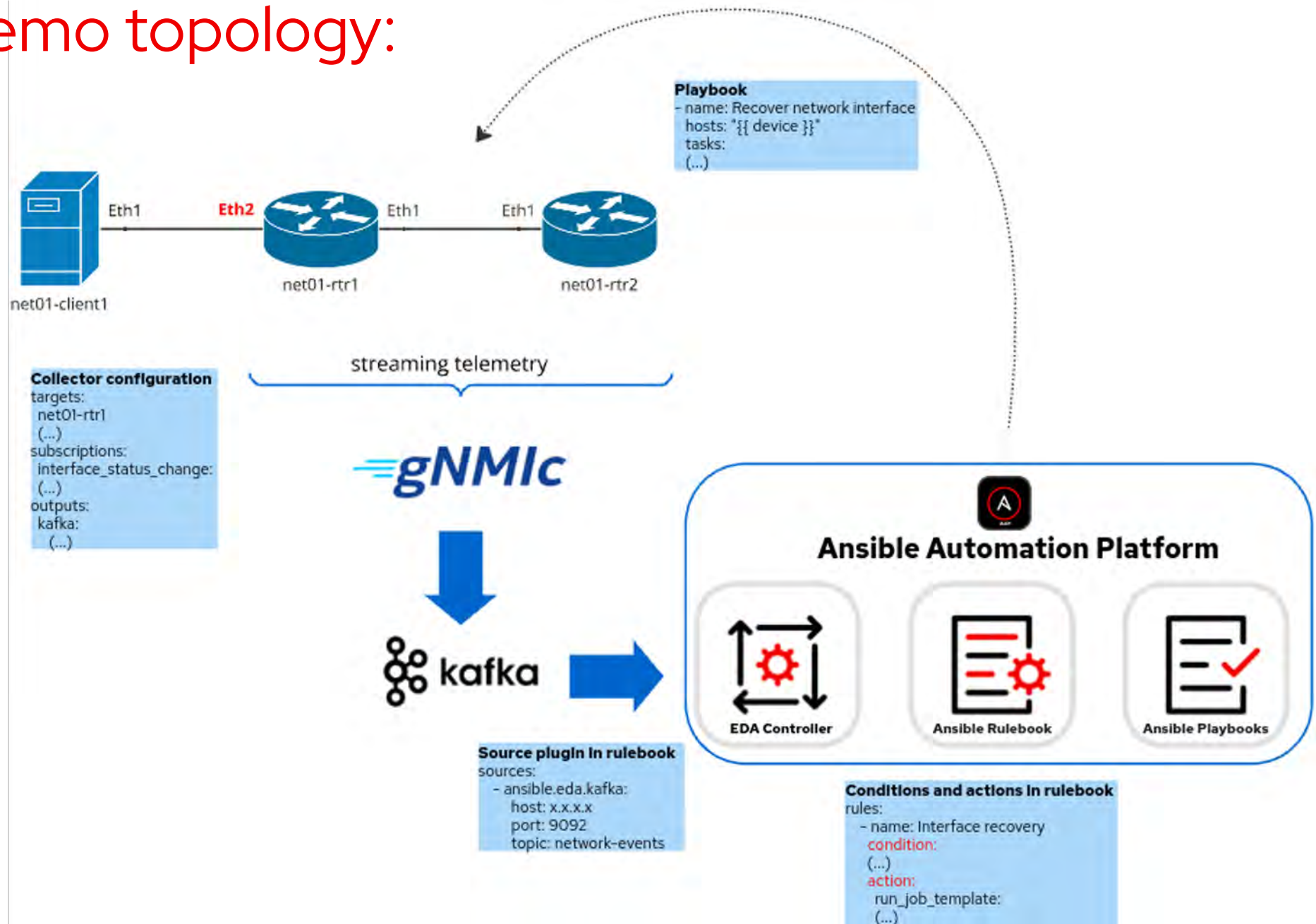
```
2024-10-02 11:20:31,238 - ansible_rulebook.rule_set_runner - INFO -  
Waiting for actions on events from Automatic network issues recovery  
2024-10-02 11:20:31,238 - ansible_rulebook.rule_set_runner - INFO -  
Waiting for events, ruleset: Automatic network issues recovery
```

Demo

Ansible and Event-Driven Automation for networks

<https://github.com/mzdyb/event-driven-ansible-for-networks>

Demo topology:



Demo steps:

In the demo we are observing automatic reaction to the following network events:

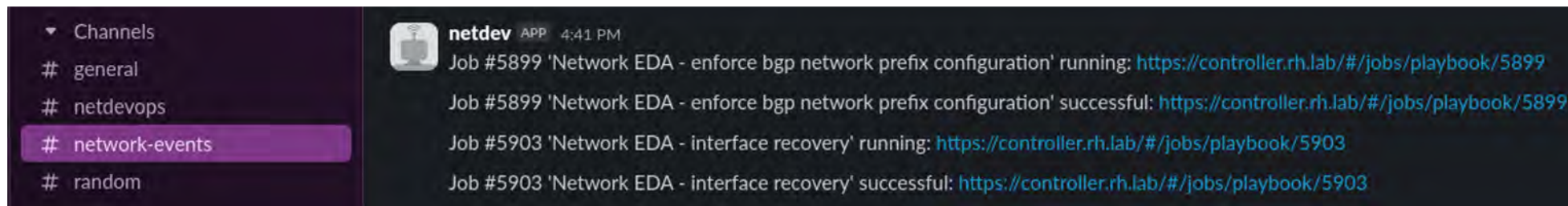
1. Operational state change of port Ethernet2 in router net01-rtr1. It is simulated by interface *shutdown* command

interface Ethernet2
shutdown  "bounce" interface Ethernet2

1. Any configuration change to *network* commands in 'router bgp' context on router net01-rtr1

router bgp 64512
(...)
network 192.168.1.0/24  apply configuration from SoT

Using Notification feature AAP sends information about automatic reaction to each event in real-time to Slack channel:



Red Hat
Summit

Connect

Thank you



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



twitter.com/RedHat