Red Hat Summit

Connect

Event-driven Ansible Everywhere

Sí, sí, también con NetOps

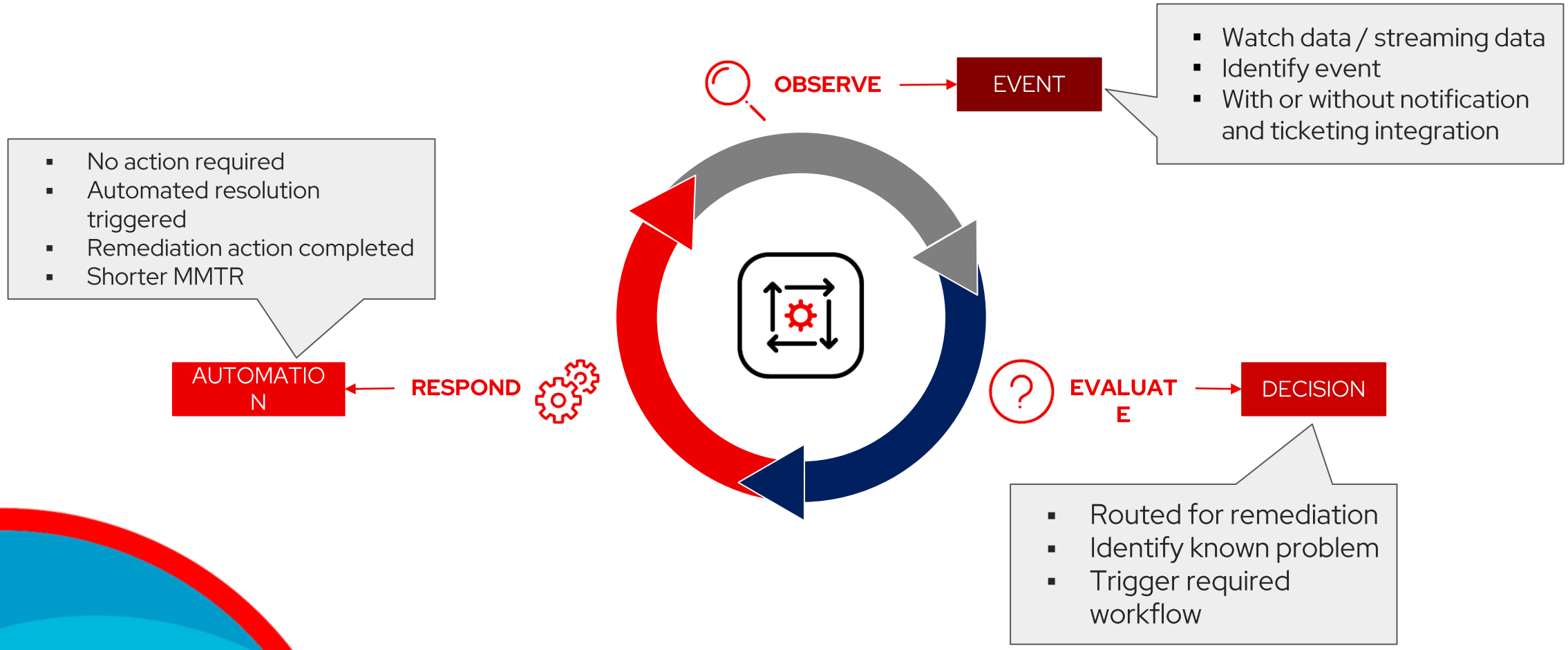
Rafael Minguillón Sánchez
Technical Account Manager

Alfonso Fernández
Technical Support Engineer

Red Hat

Event-Driven Ansible

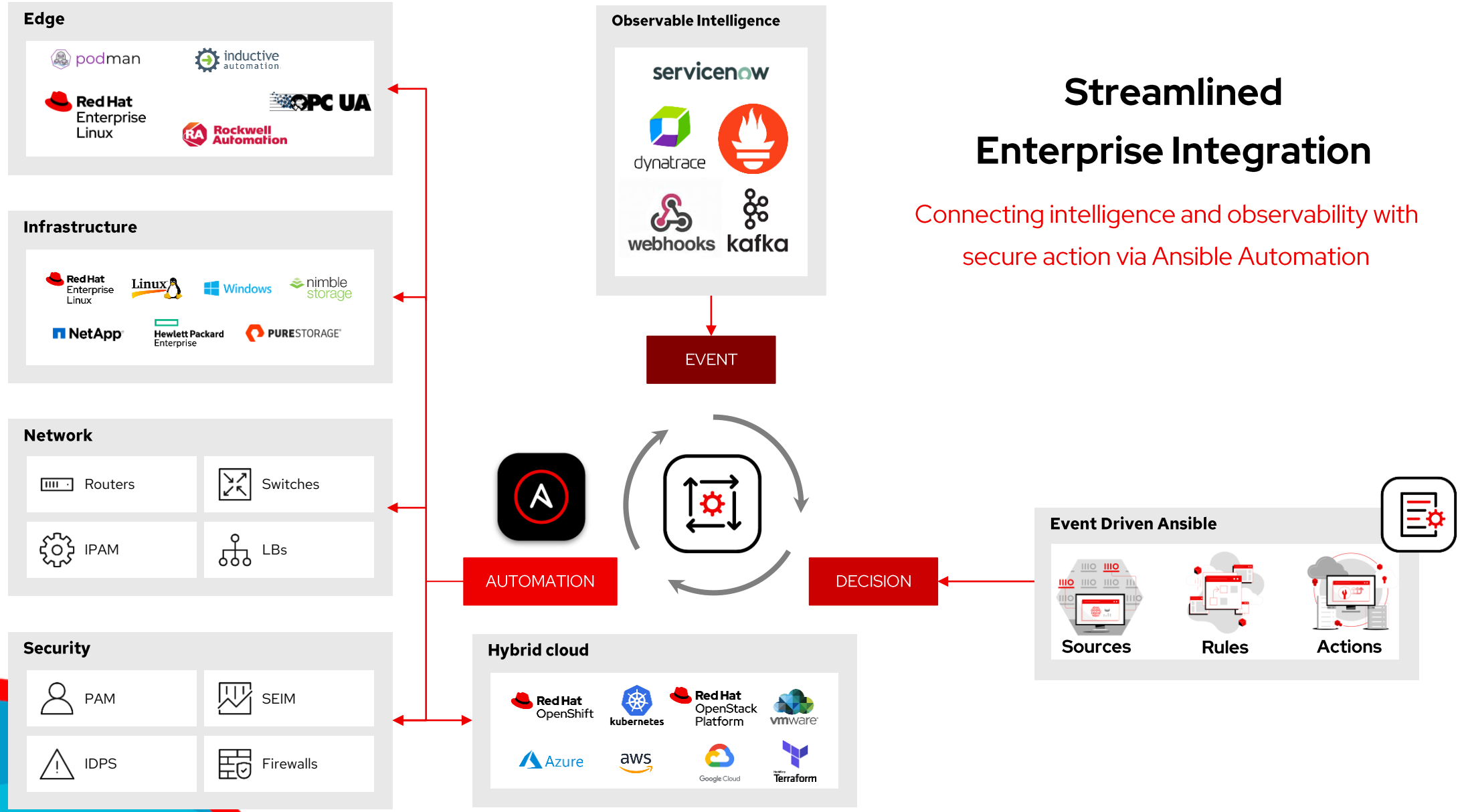
Automation Supporting Mission Critical Workloads



Red Hat

Streamlined Enterprise Integration

Connecting intelligence and observability with secure action via Ansible Automation



Red Hat

Ansible Rulebooks

Simple declarative decisions through rules

- **Events are processed by a rules engine**
  - Rules trigger based on conditions and actions can be carried out by the rules engine
  - Rules are organized into Ansible Rulebooks
  - Ansible rules can apply to events occurring on specific hosts or groups
- **Conditional management of actions to events**
  - Simple YAML structure for logical conditions
  - Events can trigger different types of actions:
    - Run Ansible Playbooks
    - Run Modules
    - Post new events to the event handler
- **YAML-like format familiarity**
  - Current Ansible users quickly learn and use Rulebook writing

```
- name: Automatic Remediation of a web server
  hosts: all
  sources:
    - name: listen for alerts
      ansible.eda.alertmanager:
        host: 0.0.0.0
        port: 8000
  rules:
    - name: restart web server
      condition: event.alert.labels.job == "fastapi"
and event.alert.status == "firing"
      action:
        run_job_template:
            name: "[JT] Restart Web Server"
```

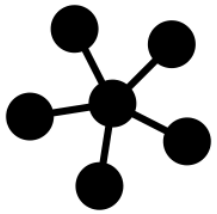
Red Hat

Model-Driven Telemetry

Push-model monitoring and real-time operational statistics

Streamed from network devices. No polling required

Uses **Reliable** transport protocols. E.g. gRPC over HTTP



Powerful **YANG** data models vs weak SNMP MIBs. Can use Netconf and Restconf

Red Hat

Ansible Utils

An Ansible Collection to ease the management, manipulation, and validation of data

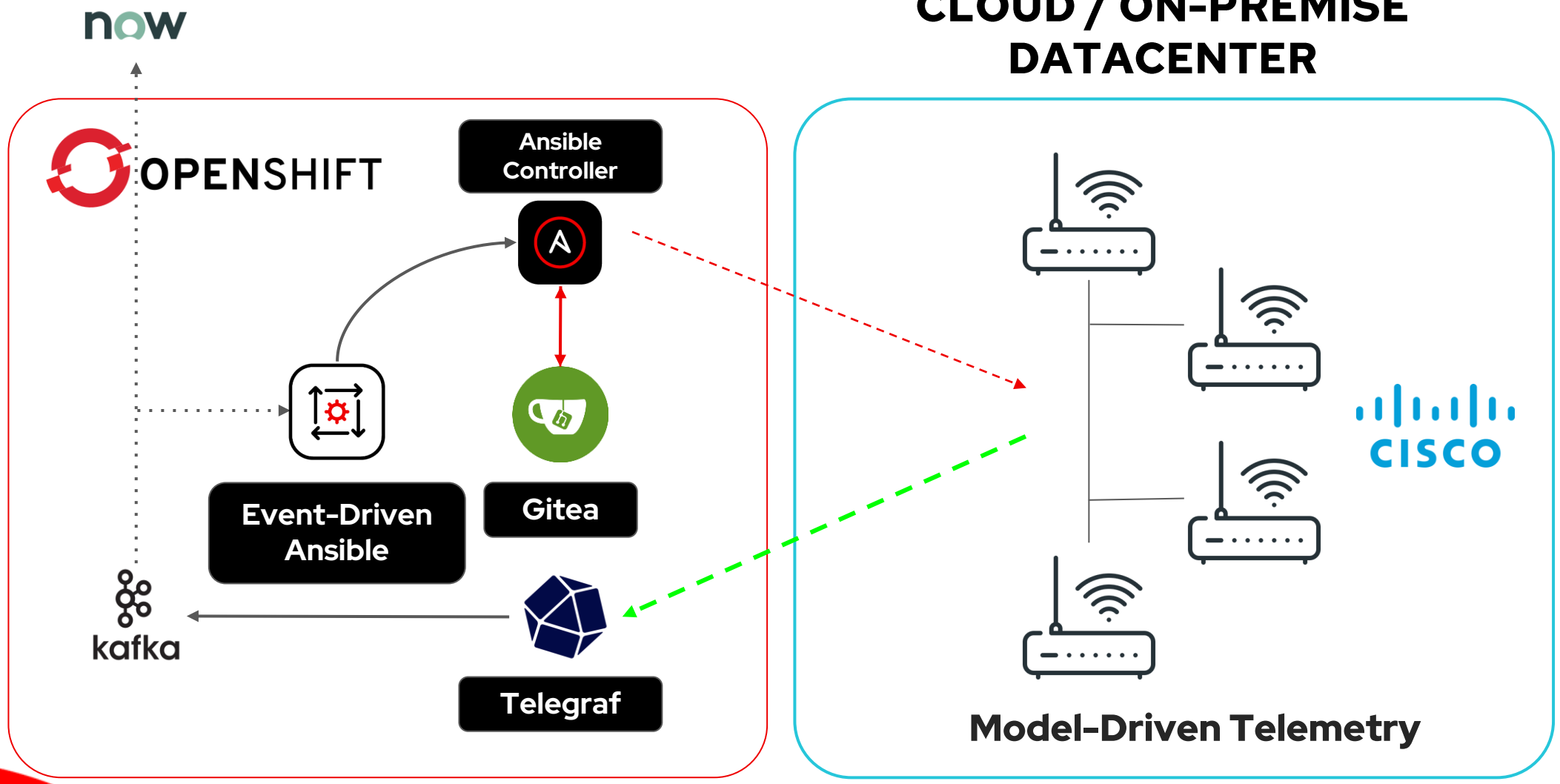
Network Device output

```
# show interfaces
Ethernet1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 022e.dbe8.1375 (bia
022e.dbe8.1375)
  Internet address is 172.18.104.95/16
  Broadcast address is 255.255.255.255
  Address determined by DHCP
  IP MTU 1500 bytes , BW 1000000 kbit
  Full-duplex, 1Gb/s, auto negotiation: on, uni-link:
n/a
  Up 10 hours, 51 minutes, 55 seconds
  Loopback Mode : None
  3 link status changes since last clear
  Last clearing of "show interface" counters never
  5 minutes input rate 950 bps (0.0% with framing
overhead), 1 packets/sec
  5 minutes output rate 858 bps (0.0% with framing
overhead), 1 packets/sec
     19361 packets input, 2964452 bytes
     Received 0 broadcasts, 0 multicast
     0 runts, 0 giants
<rest of output removed for brevity>
```

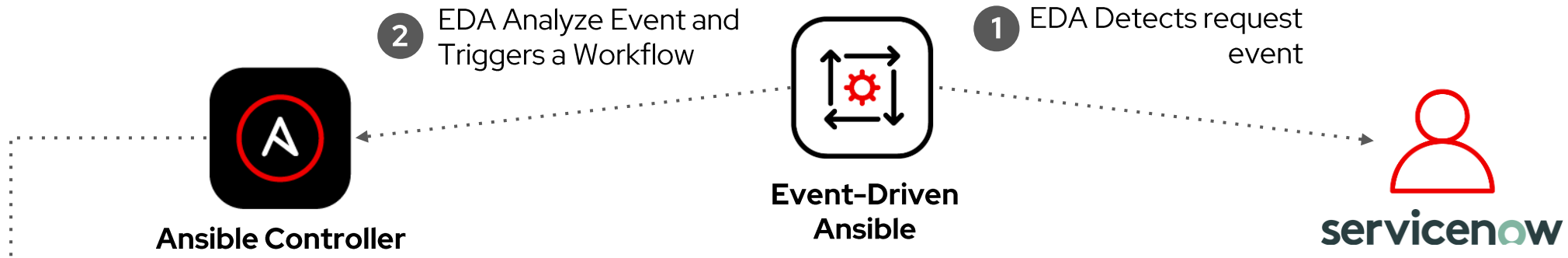
Parsed Data

```
result["parsed"]:
  Ethernet1:
    hardware: Ethernet
    mac_address: 022e.dbe8.1375
    state:
      operating: up
      protocol: up
  Loopback0:
    hardware: Loopback
    state:
      operating: up
      protocol: up
  Tunnel0:
    hardware: Tunnel
    mac_address: ac12.685f.0800
    state:
      operating: up
      protocol: up
```

Red Hat


now
CLOUD / ON-PREMISE
DATACENTER
OPENSHIFT
Ansible
Controller
Event-Driven
Ansible
Gitea
kafka
Telegraf
CISCO
Model-Driven Telemetry

Red Hat



Provision instances

- Deploy in EC2
- Get IPS and FQDNs
- Create Inventory in the Ansible controller

Auto-Discovery

- Get Facts
- Complete ServiceNow CMDB
- Create configuration and push to Git
- Create desired state json and push it to Git

Base line

- Configure Hostname
- Configure Credentials
- Configure Model-Driven Telemetry

Custom configuration

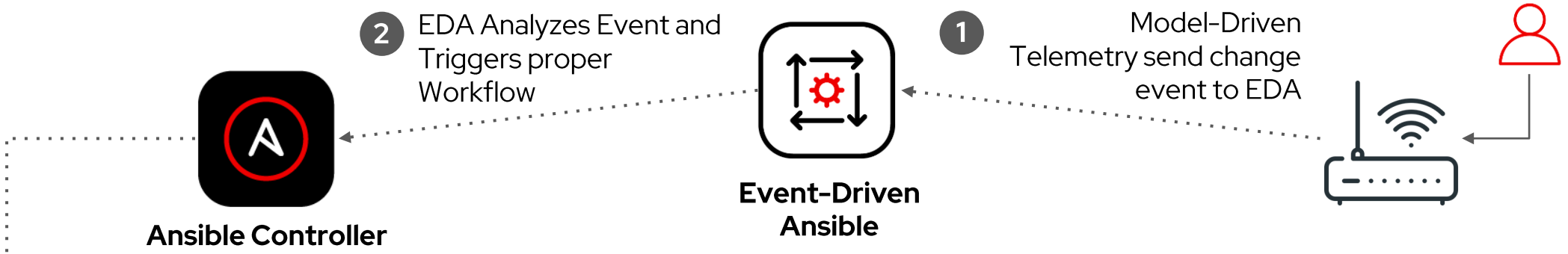
- Download configuration files from Git
- Apply Configurations
- Validate desired State
- Close Request

Infrastructure Provision

Provision, discovery and configure in a single workflow

Red Hat

Analyze changes and validate them to get the desired configuration state



Check Operational State

- Download JSON from Git
- Parse and compare with operational state
- Fail if doesn't match

Open ticket

- Open a ticket in ServiceNow

Fix Configuration

- Download Configuration from Git
- Apply Configuration
- Parse and compare with operational state

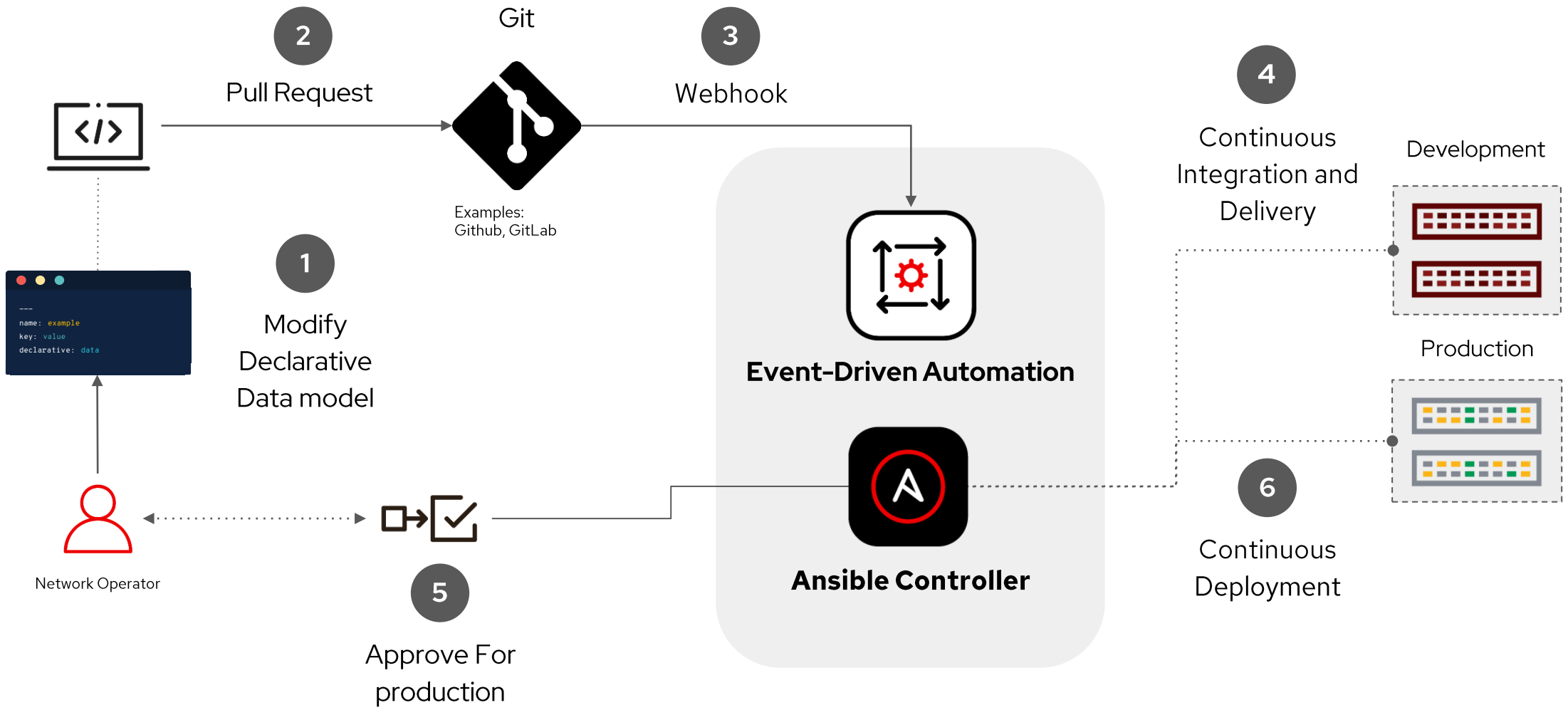
Close ticket

- Fill and close Servicenow ticket

Red Hat

Automated NetOps

Using Gitops and Infrastructure as code
to keep standard configurations



Red Hat

Red Hat Summit

Connect

Thank you

 linkedin.com/company/red-hat

 facebook.com/redhatinc

 youtube.com/user/RedHatVideos

 twitter.com/RedHat

Red Hat