# Improving Network Observability with Telemetry using gNMIc and Prometheus

edgeuno

AS 7195

# Telemetry

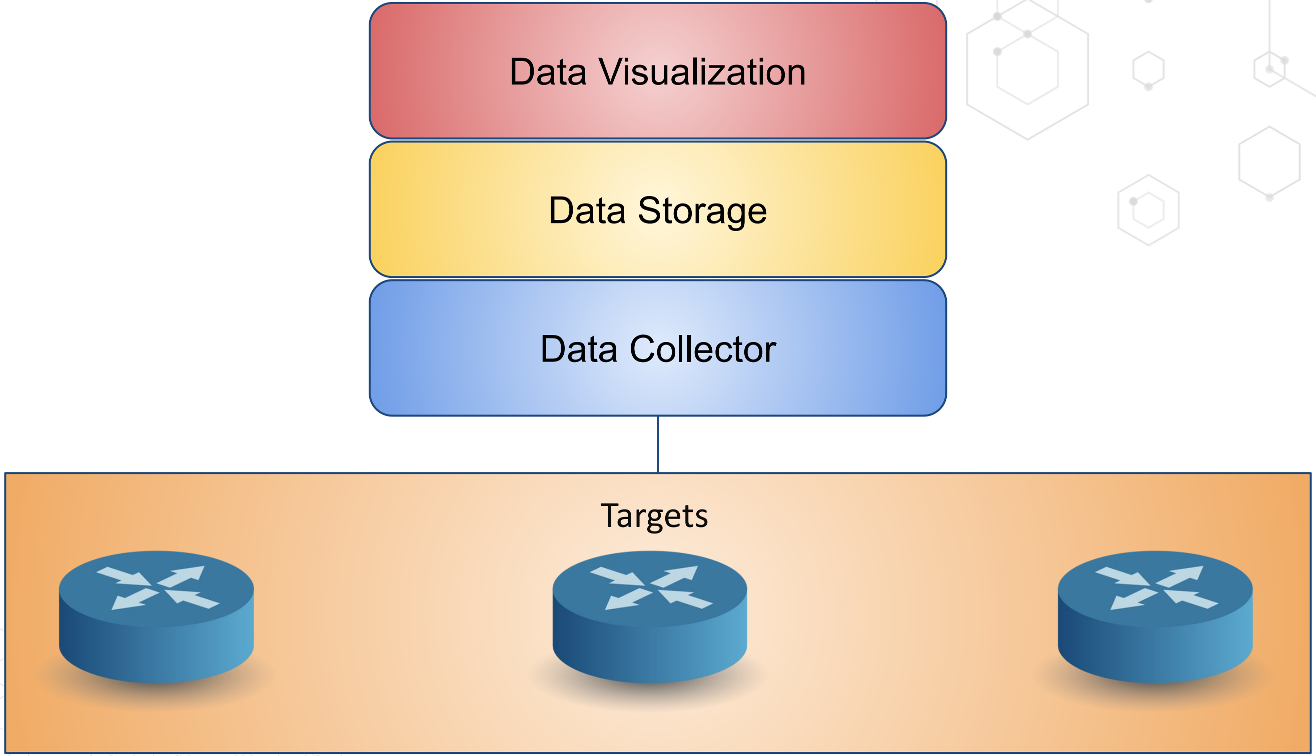
- **Telemetry Concept**
  - Data collecting, processing, and displaying
  - Real-Time
  - Highly used in many environments

- **Telemetry on Network (RFC 9232)**
  - SLA Compliance
  - Network Optimization
  - Event-Tracking

# SNMP x Telemetry

| SNMP | Telemetry |
|---|---|
| UDP | gRPC, HTTP/2 |
| Polling-based | Streaming-based |
| Latency due poller time | Close to real-time data collection |
| Widely standardized, but less modern. | Accepted by newer devices |

# Choosing a Stack



Data Visualization

Data Storage

Data Collector

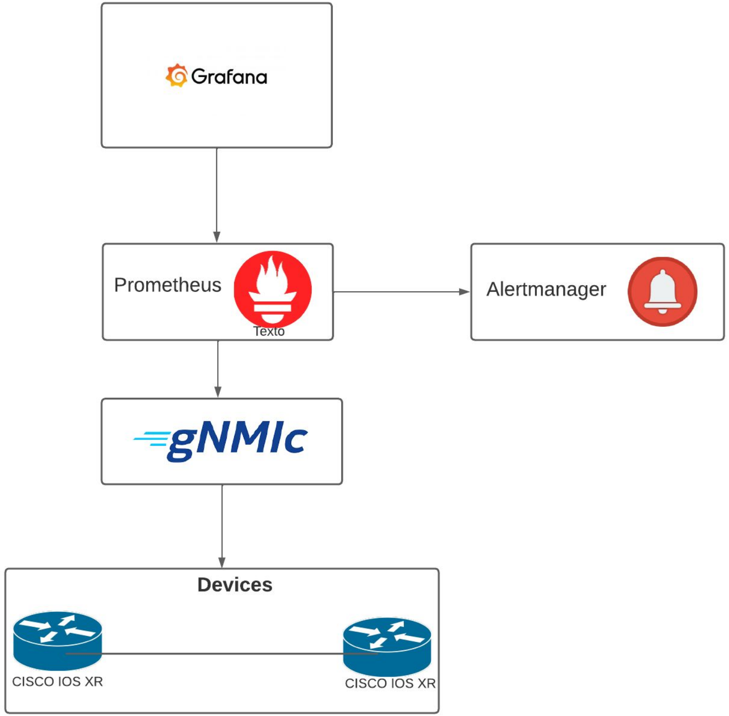
Targets

# Stack with gNMIc and Prometheus

- **Data Collector (gNMIc)**
  - gNMIc is a gNMI client developed as part of the OpenConfig project.
  - It uses HTTP/2 over RPC to manage and monitor network devices.
- **Data Storage (Prometheus)**
  - Prometheus is a monitoring and alerting system that collects and stores metrics as time series data with timestamps.
  - It scrapes data from endpoints and saves samples as numerical measurements, or metrics.
- **Data Visualization (Grafana)**
  - Visualization, monitoring, and analytics platform

# Example of gNMIc Telemetry Stack

- **Targets ( Devices)**
  - [Cisco IOS XR, 7.3.2]
- **Colleting**
  - [gNMIc Single or Cluster]
- **TSDB**
  - [Prometheus]
- **Visualization**
  - [Grafana]



# Device Configuration

- **Configuring devices to respond gNMIc:**
  - Allow protocol gRPC
  - Allow gRPC port in policy (if used)
- **Example**
  - Junos OS :
    - system extension-service request-response grpc
      - define port, max-sessions, ssl
  - Cisco:
    - grpc
      - port, ssl, max-sessions

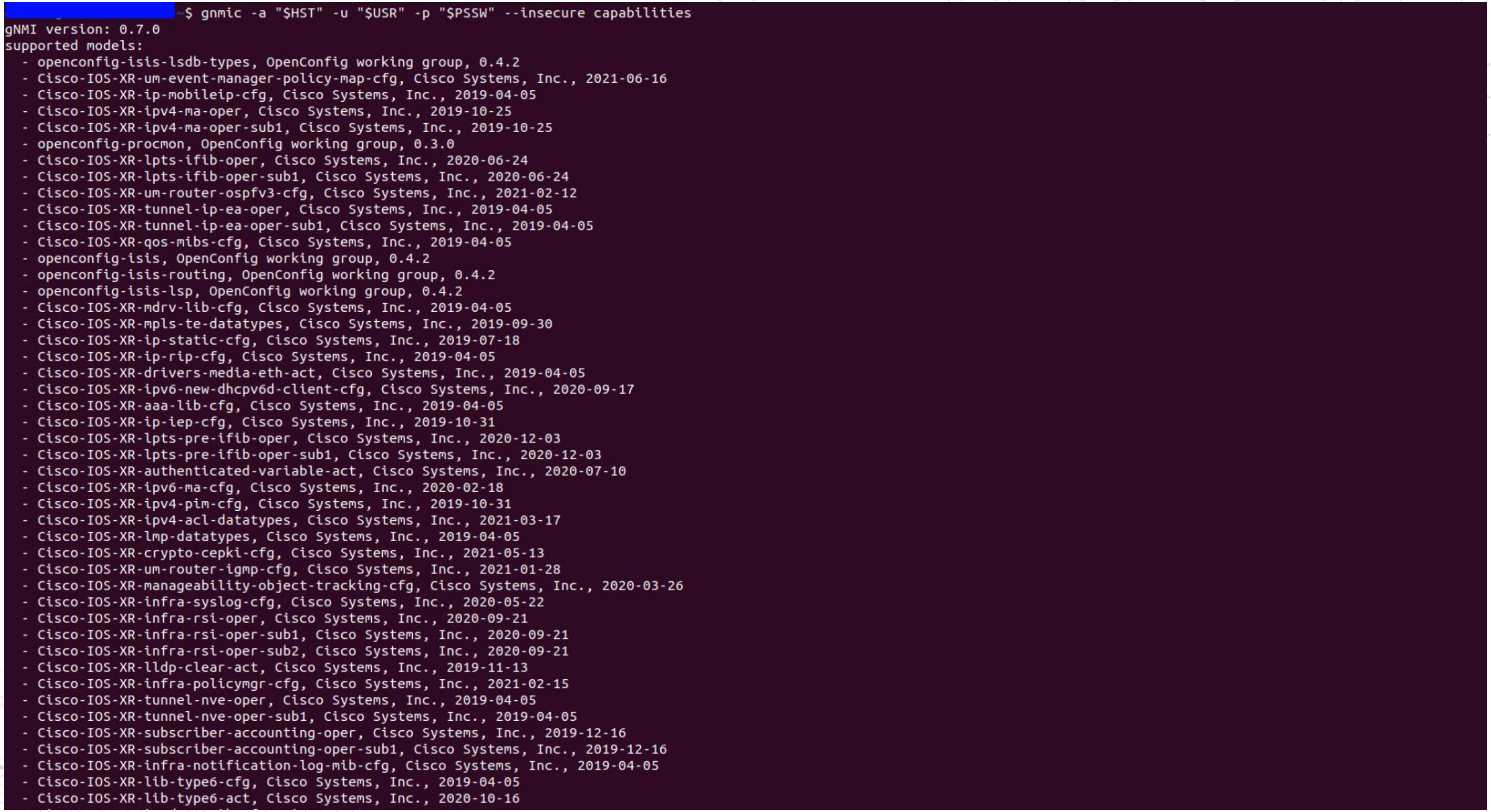
# Using gNMIc

- **gNMIc Linux install:**
  - ○ bash -c "$(curl -sL https://get-gnmic.openconfig.net)"

- **gNMIC with Containerlab**
  - ○ container-based networking labs.

- **gNMIc with Docker / Docker-Compose**
  - ○ image: ghcr.io/openconfig/gnmic

# Testing data collection - Capabilities



```
[redacted]~$ gnmic -a "$HST" -u "$USR" -p "$PSSW" --insecure capabilities
gNMI version: 0.7.0
supported models:
  - openconfig-isis-lsdb-types, OpenConfig working group, 0.4.2
  - Cisco-IOS-XR-um-event-manager-policy-map-cfg, Cisco Systems, Inc., 2021-06-16
  - Cisco-IOS-XR-ip-mobileip-cfg, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-ipv4-ma-oper, Cisco Systems, Inc., 2019-10-25
  - Cisco-IOS-XR-ipv4-ma-oper-sub1, Cisco Systems, Inc., 2019-10-25
  - openconfig-procmon, OpenConfig working group, 0.3.0
  - Cisco-IOS-XR-lpts-ifib-oper, Cisco Systems, Inc., 2020-06-24
  - Cisco-IOS-XR-lpts-ifib-oper-sub1, Cisco Systems, Inc., 2020-06-24
  - Cisco-IOS-XR-um-router-ospfv3-cfg, Cisco Systems, Inc., 2021-02-12
  - Cisco-IOS-XR-tunnel-ip-ea-oper, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-tunnel-ip-ea-oper-sub1, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-qos-mibs-cfg, Cisco Systems, Inc., 2019-04-05
  - openconfig-isis, OpenConfig working group, 0.4.2
  - openconfig-isis-routing, OpenConfig working group, 0.4.2
  - openconfig-isis-lsp, OpenConfig working group, 0.4.2
  - Cisco-IOS-XR-mdrv-lib-cfg, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-mpls-te-datatypes, Cisco Systems, Inc., 2019-09-30
  - Cisco-IOS-XR-ip-static-cfg, Cisco Systems, Inc., 2019-07-18
  - Cisco-IOS-XR-ip-rip-cfg, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-drivers-media-eth-act, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-ipv6-new-dhcpv6d-client-cfg, Cisco Systems, Inc., 2020-09-17
  - Cisco-IOS-XR-aaa-lib-cfg, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-ip-iep-cfg, Cisco Systems, Inc., 2019-10-31
  - Cisco-IOS-XR-lpts-pre-ifib-oper, Cisco Systems, Inc., 2020-12-03
  - Cisco-IOS-XR-lpts-pre-ifib-oper-sub1, Cisco Systems, Inc., 2020-12-03
  - Cisco-IOS-XR-authenticated-variable-act, Cisco Systems, Inc., 2020-07-10
  - Cisco-IOS-XR-ipv6-ma-cfg, Cisco Systems, Inc., 2020-02-18
  - Cisco-IOS-XR-ipv4-pim-cfg, Cisco Systems, Inc., 2019-10-31
  - Cisco-IOS-XR-ipv4-acl-datatypes, Cisco Systems, Inc., 2021-03-17
  - Cisco-IOS-XR-lmp-datatypes, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-crypto-cepki-cfg, Cisco Systems, Inc., 2021-05-13
  - Cisco-IOS-XR-um-router-igmp-cfg, Cisco Systems, Inc., 2021-01-28
  - Cisco-IOS-XR-manageability-object-tracking-cfg, Cisco Systems, Inc., 2020-03-26
  - Cisco-IOS-XR-infra-syslog-cfg, Cisco Systems, Inc., 2020-05-22
  - Cisco-IOS-XR-infra-rsi-oper, Cisco Systems, Inc., 2020-09-21
  - Cisco-IOS-XR-infra-rsi-oper-sub1, Cisco Systems, Inc., 2020-09-21
  - Cisco-IOS-XR-infra-rsi-oper-sub2, Cisco Systems, Inc., 2020-09-21
  - Cisco-IOS-XR-lldp-clear-act, Cisco Systems, Inc., 2019-11-13
  - Cisco-IOS-XR-infra-policymgr-cfg, Cisco Systems, Inc., 2021-02-15
  - Cisco-IOS-XR-tunnel-nve-oper, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-tunnel-nve-oper-sub1, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-subscriber-accounting-oper, Cisco Systems, Inc., 2019-12-16
  - Cisco-IOS-XR-subscriber-accounting-oper-sub1, Cisco Systems, Inc., 2019-12-16
  - Cisco-IOS-XR-infra-notification-log-mib-cfg, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-lib-type6-cfg, Cisco Systems, Inc., 2019-04-05
  - Cisco-IOS-XR-lib-type6-act, Cisco Systems, Inc., 2020-10-16
```

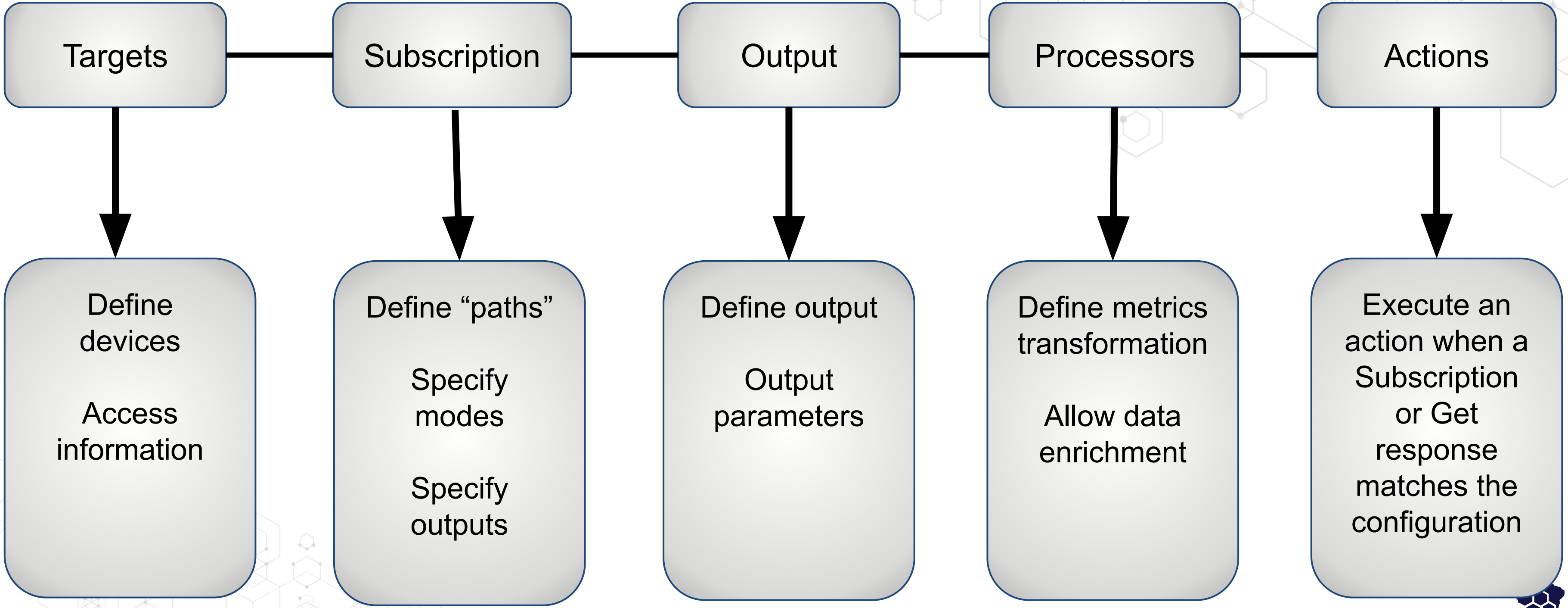
# Building the Docker-Compose

```
prometheus:
    image: prom/prometheus:v2.47.0
    container_name: prometheus
    volumes:
        - ./prometheus/:/etc/prometheus/
        - prometheus-data:/prometheus
    command:
        - '--config.file=/etc/prometheus/prometheus.yaml'
        - '--storage.tsdb.path=/prometheus'
        - '--web.console.libraries=/usr/share/prometheus/console_libraries'
        - '--web.console.templates=/usr/share/prometheus/consoles'
    ports:
        - 9090:9090
```

```
gnmic:
    image: gnmic:latest
    container_name: gnmic1
    volumes:
        - ./gnmic.yaml:/app/gnmic.yaml
    command: "subscribe --config /app/gnmic.yaml"
    ports:
        - 9804:9804
```

```
grafana:
    container_name: grafana
    image: grafana/grafana:latest
    ports:
        - "3001:3000"
```

# gNMIc - Configuration



Targets

Define devices

Access information

Subscription

Define “paths”

Specify modes

Specify outputs

Output

Define output

Output parameters

Processors

Define metrics transformation

Allow data enrichment

Actions

Execute an action when a Subscription or Get response matches the configuration

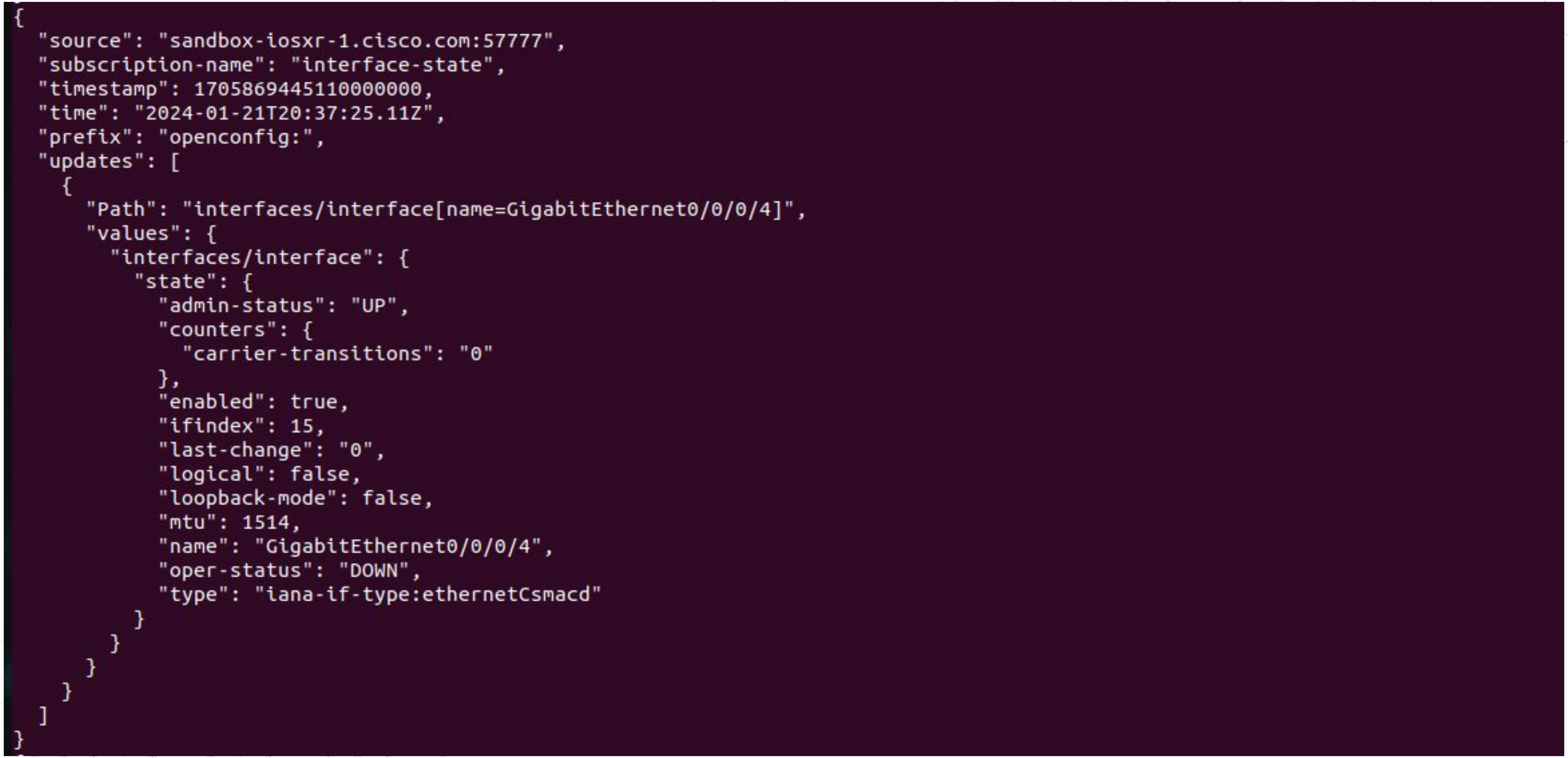
# gNMIc - Configuration Example

- **Targets and Subscriptions**
  - Targets specify the devices and access information.
  - Subscriptions specify the data collection mode, outputs, and paths.

```
targets:
  cisco-IOS-XR:
    address: sandbox-iosxr-1.cisco.com:57777
    username: ${USERNAME}
    password: ${PASSWORD}
    subscriptions:
     - interface-state
```

```
subscriptions:
  interface-state:
     paths:
        - /interfaces/interface
     mode: STREAM
     stream-mode: on-change
     outputs:
           - prom-output
```

# Output example:



```
{
  "source": "sandbox-iosxr-1.cisco.com:57777",
  "subscription-name": "interface-state",
  "timestamp": 1705869445110000000,
  "time": "2024-01-21T20:37:25.11Z",
  "prefix": "openconfig:",
  "updates": [
    {
      "Path": "interfaces/interface[name=GigabitEthernet0/0/0/4]",
      "values": {
        "interfaces/interface": {
          "state": {
            "admin-status": "UP",
            "counters": {
              "carrier-transitions": "0"
            },
            "enabled": true,
            "ifindex": 15,
            "last-change": "0",
            "logical": false,
            "loopback-mode": false,
            "mtu": 1514,
            "name": "GigabitEthernet0/0/0/4",
            "oper-status": "DOWN",
            "type": "iana-if-type:ethernetCsmacd"
          }
        }
      }
    }
  ]
}
```

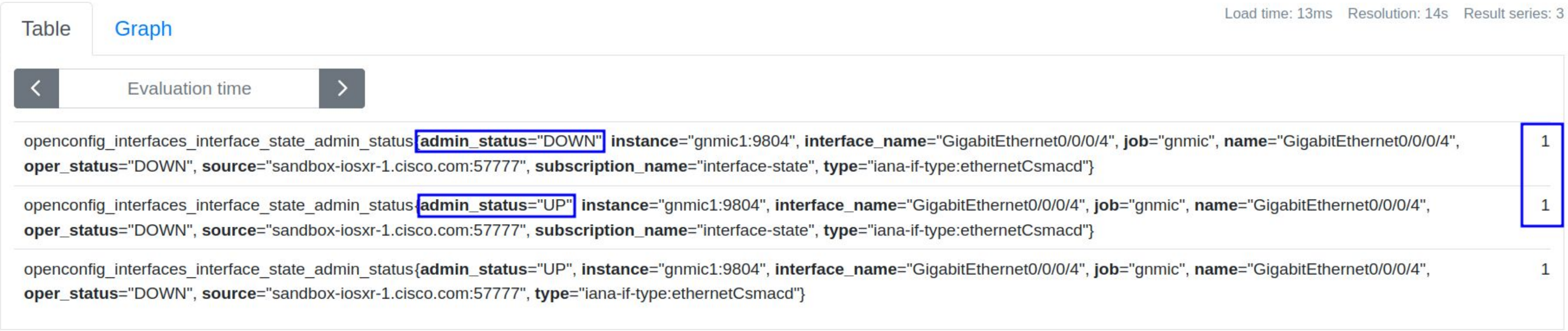
# gNMIc - Configuration Example

- **Outputs**
  - gNMIc outputs allow the user to store the collected metrics.

```
outputs:
  prom-output:
      type: prometheus_write
      url: http://prometheus:9090/api/v1/write
      debug: true
      event-processors:
      - delete-data
      - convert-timestamp
```

```
processors:
  convert-timestamp:
      event-override-ts:
           precision: ns
```

# Collecting Data - Prometheus



Load time: 13ms Resolution: 14s Result series: 3

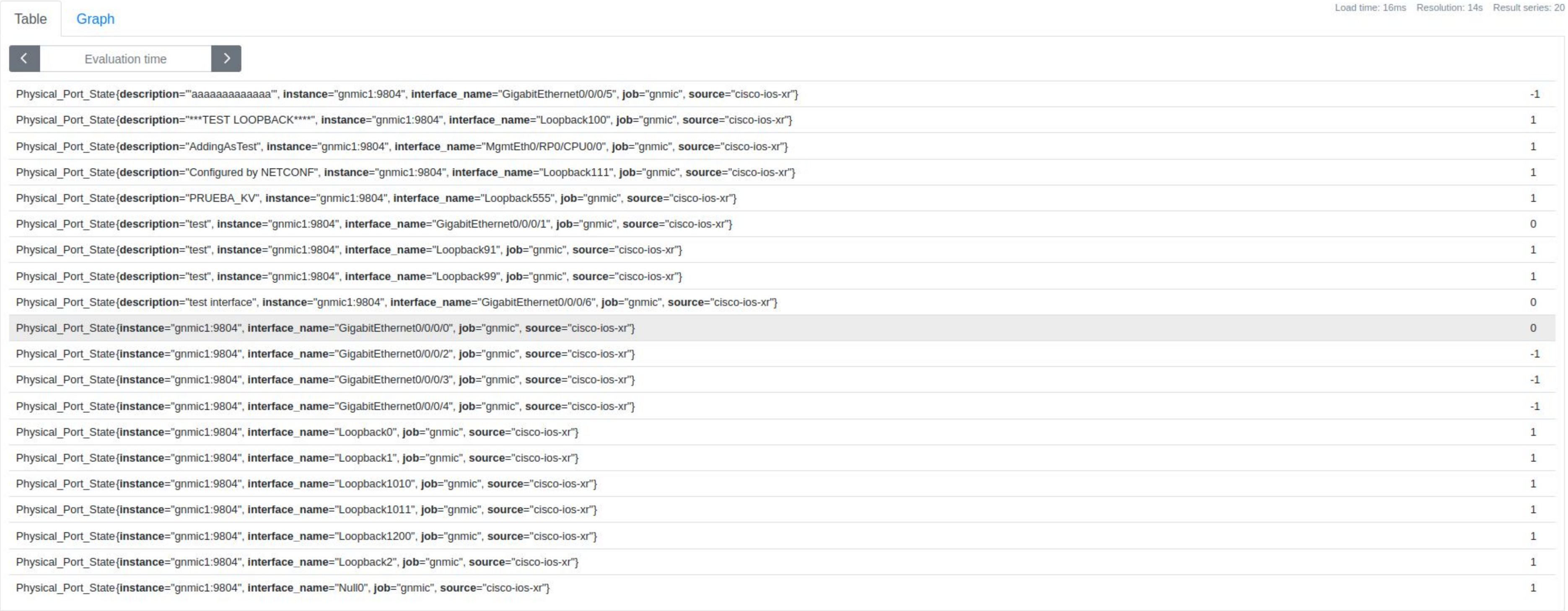
```
openconfig_interfaces_interface_state_admin_status{admin_status="DOWN", instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/4", job="gnmic", name="GigabitEthernet0/0/0/4",   1
oper_status="DOWN", source="sandbox-iosxr-1.cisco.com:57777", subscription_name="interface-state", type="iana-if-type:ethernetCsmacd"}

openconfig_interfaces_interface_state_admin_status{admin_status="UP", instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/4", job="gnmic", name="GigabitEthernet0/0/0/4",   1
oper_status="DOWN", source="sandbox-iosxr-1.cisco.com:57777", subscription_name="interface-state", type="iana-if-type:ethernetCsmacd"}

openconfig_interfaces_interface_state_admin_status{admin_status="UP", instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/4", job="gnmic", name="GigabitEthernet0/0/0/4",   1
oper_status="DOWN", source="sandbox-iosxr-1.cisco.com:57777", type="iana-if-type:ethernetCsmacd"}
```

# gNMIc - Configuration Example

- **Processors**
  - gNMIc event processors allow us to transform an event message that will be be written to output.

```
processors:
  delete-data:
      event-delete:
      value-names:
      - ".*openconfig.*"
  delete-tags:
      event-delete:
      tag-names:
      - "^subscription-name"
      - ".*status.*"
```

# Formatted Output



```
Physical_Port_State{description="'aaaaaaaaaaaaa'", instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/5", job="gnmic", source="cisco-ios-xr"}   -1
Physical_Port_State{description="***TEST LOOPBACK****", instance="gnmic1:9804", interface_name="Loopback100", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{description="AddingAsTest", instance="gnmic1:9804", interface_name="MgmtEth0/RP0/CPU0/0", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{description="Configured by NETCONF", instance="gnmic1:9804", interface_name="Loopback111", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{description="PRUEBA_KV", instance="gnmic1:9804", interface_name="Loopback555", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{description="test", instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/1", job="gnmic", source="cisco-ios-xr"}   0
Physical_Port_State{description="test", instance="gnmic1:9804", interface_name="Loopback91", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{description="test", instance="gnmic1:9804", interface_name="Loopback99", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{description="test interface", instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/6", job="gnmic", source="cisco-ios-xr"}   0
Physical_Port_State{instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/0", job="gnmic", source="cisco-ios-xr"}   0
Physical_Port_State{instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/2", job="gnmic", source="cisco-ios-xr"}   -1
Physical_Port_State{instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/3", job="gnmic", source="cisco-ios-xr"}   -1
Physical_Port_State{instance="gnmic1:9804", interface_name="GigabitEthernet0/0/0/4", job="gnmic", source="cisco-ios-xr"}   -1
Physical_Port_State{instance="gnmic1:9804", interface_name="Loopback0", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{instance="gnmic1:9804", interface_name="Loopback1", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{instance="gnmic1:9804", interface_name="Loopback1010", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{instance="gnmic1:9804", interface_name="Loopback1011", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{instance="gnmic1:9804", interface_name="Loopback1200", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{instance="gnmic1:9804", interface_name="Loopback2", job="gnmic", source="cisco-ios-xr"}   1
Physical_Port_State{instance="gnmic1:9804", interface_name="Null0", job="gnmic", source="cisco-ios-xr"}   1
```

# Creating actions

- **Actions**
  - Enables the execution of an action when an event is triggered.

```
processors:
    trigger-alarm:
      event-trigger:
      condition: '.values.port_state == "0"'
      min-occurrences: 1
      max-occurrences: 2
      window: 60s
      async: true
      actions:
           - alarm
```

```
actions:
  alarm:
      type: http
      method: POST
      url:
      headers:
           content-type: application/text
      timeout: 5s
      body: ' Teste'
      debug: false
```

# Conclusion

- **Although the configuration is complex, there are a lot of benefits for your network:**
    - real-time monitoring, less CPU intensive, granularity
    - personalize for your necessities
    - reinforce SLAs

- **Flow technologies can enhance data collection**
    - Flow can help understand “who” whereas gNMI helps understand “what” is impacting your network

# OBRIGADO!

Danilo Rodrigues

edgeuno

AS 7195