

Topology at Scale

Building network topology with Zabbix: CDP/LLDP discovery, external streaming, and the performance work behind it — Andy Wojnarek, ATS Group



What We're Covering

Part 1 — Network Topology Maps & External Data Streaming

How we operationalized CDP/LLDP-based discovery in Zabbix, and stream it out to third party platforms.

Part 2 — Performance Modifications

Codebase changes that assist with performance and a capacity — work in progress.

Topology & Streaming

Connector

*

Name

topology_network_connector

Protocol

Zabbix Streaming Protocol v1.0

Data type

Item values

Events

*

URL

URL HERE

Tag filter

And/Or

Or

gpe_site

Equals

DataSite1

Remove

source

Exists

Remove

source

Does not contain

INVENTORY

Remove

sysName

Exists

Remove

sysName

Does not contain

INVENTORY

Remove

sysName

Does not contain

UNKNOWN

Remove

__field__

Exists

Remove

__name__

Exists

Remove

Add

*

Type of information

☒ Numeric (unsigned)

☐ Character

☐ Text

☒ Numeric (float)

☐ Log

☐ Binary

HTTP authentication

Bearer

*

Bearer token

XXXXXX

Advanced configuration

Description

Enabled

☒

Update

Clone

Delete

Cancel



OCTOBER 7 ■ 9, 2026
RIGA ■ LATVIA

What is CDP and LLDP?

Layer 2 discovery protocols that build a live map of how the network is physically wired

- What they are
 - **CDP (Cisco Discovery Protocol)**: Cisco's own neighbor-discovery protocol, and the default on Cisco gear
 - **LLDP (Link Layer Discovery Protocol, IEEE 802.1AB)**: the open standard that works across vendors (Cisco, Juniper, Arista, HPE, Dell, IBM Power/VIOS, VMware, and more)
- How they build topology
 - Each device sends small advertisements out every active port (every 60s for CDP, 30s for LLDP)
 - Only **directly connected neighbors** receive them. The frames aren't forwarded, so each device learns only its one-hop peers
 - Each device keeps a **neighbor table** that says "my port X connects to device Y, port Z"
 - Put together, all the neighbor tables form an accurate, **physical (Layer 2) map** of the network

CDP Cache Info #151.132: cdpCacheAddressType ?

CDP Cache Info #166.133: cdpCacheAddressType ?

CDP Cache Info #167.131: cdpCacheAddressType ?

CDP Cache Info #176.128: cdpCacheAddressType ?

CDP Cache Info #177.134: cdpCacheAddressType ?

CDP Cache Info #181.130: cdpCacheAddressType ?

CDP Cache Info #184.129: cdpCacheAddressType ?

CDP Cache Info #227.120: cdpCacheAddressType ?

CDP Interface Info #2: cdpInterfaceEnable ?

CDP Interface Info #3: cdpInterfaceEnable ?

CDP Interface Info #4: cdpInterfaceEnable ?

CDP Interface Info #5: cdpInterfaceEnable ?

CDP Interface Info #6: cdpInterfaceEnable ?

CDP Interface Info #7: cdpInterfaceEnable ?

CDP Interface Info #8: cdpInterfaceEnable ?

CDP Interface Info #9: cdpInterfaceEnable ?



01

SNMP Walk

Every device in scope gets walked for CDP/LLDP and neighbor data — no shortcuts.

02

Discovery

Walk results feed Zabbix low-level discovery to generate the right items per device.

03

Dependent Items & Filtering

Conditional logic decides what matters — if we see this neighbor type, we do that.

[illegible]

SNMP's Rough Edges

SNMPv2 & VLANs

Pulling per-VLAN data means separate calls with different community-string parameters for each VLAN — still a work in progress.

Naming Isn't Consistent

CDP/LLDP report a device by name or by port alias inconsistently — one OID table exposes a flag we use to pick the right one.

Normalizing Across Vendors

CPU and memory OIDs differ by vendor, and polling frequency has to split by throughput — high-speed OIDs vs. low-speed OIDs.

One Template, or Many?

- Still Exploring: one template to rule them all, or separate templates for neighbor data and for performance/alerting — for maintainability.
- Right now, we run just one template for everything.
- The catch: when an alerting template rides on top of the vendor template for ICMP checks, item IDs collide.

Discovery rules

[All templates](#) / [GPE Network Collection by ATS - Ca...](#) / [Discovery list](#) / [SNMP Device CPU Info \(avg\) Discov...](#) / [Item prototypes 2](#) / [Trigger prototypes](#) / [Graph prototypes 1](#) / [Host prototypes](#)

[Discovery rule](#) / [Preprocessing 2](#) / [LLD macros](#) / [Filters 1](#) / [Overrides](#)

* Name

Type

* Key

* Master item

* Delete lost resources ?

* Disable lost resources ?

Description

Enabled ☒

Where the Real Noise Comes From

It's Not CDP/LLDP

The high-volume noise isn't neighbor data — it's performance data.

Routing Tables, for Firewalls

HW, SW, and multi-hop routing tables get pulled in too — firewalls need them since they don't typically run CDP/LLDP.

Specific Counters, Not Everything

We track certain error rates deliberately, not blanket NIC-level error counters.

Tag Noise



TAGS

[atIfIndex](#) 2 [atNetAddress](#) 2 [atPhysAddress](#) 2 [cdpCacheAddress](#) 2 [cdpCacheAddressType](#) 2 [cdpCacheApplianceID](#) 2 [cdpCacheCapabilities](#) 2 [cdpCacheDeviceId](#) 2 [cdpCacheDeviceIndex](#) 2 [cdpCacheDevicePort](#) 2 [cdpCacheDuplex](#) 2 [cdpCacheIfIndex](#) 2 [cdpCacheMTU](#) 2 [cdpCacheNativeVLAN](#) 2 [cdpCachePhysLocation](#) 2 [cdpCachePlatform](#) 2 [cdpCachePowerConsumption](#) 2 [cdpCachePrimaryMgmtAddr](#) 2 [cdpCachePrimaryMgmtAddrType](#) 2 [cdpCacheSecondaryMgmtAddr](#) 2 [cdpCacheSecondaryMgmtAddrType](#) 2 [cdpCacheSysName](#) 2 [cdpCacheSysObjectID](#) 2 [cdpCacheVersion](#) 2 [cdpCacheVlanID](#) 2 [cdpCacheVTPMgmtDomain](#) 2 [cdplInterfaceEnable](#) 169 [cdplInterfaceGroup](#) 169 [cdplInterfaceIfIndex](#) 169 [cdplInterfaceName](#) 169 [cdplInterfacePort](#) 169 [cefcFanTrayOperStatus](#) 18 [cefcFRUPowerAdminStatus](#) 18 [cefcFRUPowerName](#) 18 [cefcFRUPowerOperStatus](#) 18 [cefcModuleAdminStatus](#) 3 [cefcModuleOperStatus](#) 3 [cefcModuleResetReason](#) 3 [cefcModuleResetReasonDescription](#) 3 [cefcModuleStateChangeReasonDescr](#) 3 [cefcVmModuleOperStatus](#) 3 [cempMemPoolName](#) 12 [cempMemPoolType](#) 12 [ciscoEnvMonFanState](#) 18 [ciscoEnvMonFanStatusDescr](#) 18 [ciscoEnvMonFanStatusIndex](#) 18 [ciscoEnvMonSupplyState](#) 6 [ciscoEnvMonSupplyStatusDescr](#) 6 [ciscoEnvMonSupplyStatusIndex](#) 6 [ciscoEnvMonTemperatureState](#) 18 [ciscoEnvMonTemperatureStatusDescr](#) 18 [ciscoEnvMonTemperatureThreshold](#) 18 [ciscoMemoryPoolName](#) 12 [clagAggPortListInterfaceIndexList](#) 1 [clagAggPortListPorts](#) 1 [component](#) 3470 [defaultSerialNumberOid](#) 7 [description](#) 4925 [domainName](#) 7 [dot1dBasePort](#) 22 [dot1dBasePortIfIndex](#) 22 [dot1dTpFdbAddress](#) 1 [dot1dTpFdbPort](#) 1 [dot1dTpFdbStatus](#) 1 [dot3StatsDuplexStatus](#) 3011 [entLogicalContextEngineID](#) 7 [entLogicalContextName](#) 7 [entLogicalDescr](#) 7 [entLogicalIndex](#) 7 [entLogicalTAddress](#) 7 [entLogicalTDomain](#) 7 [entLogicalType](#) 7 [entPhysicalAlias](#) 296 [entPhysicalAssetID](#) 247 [entPhysicalClass](#) 247 [entPhysicalContainedIn](#) 247 [entPhysicalDescr](#) 247 [entPhysicalFirmwareRev](#) 247 [entPhysicalHardwareRev](#) 247 [entPhysicalIndex](#) 247 [entPhysicalsFRU](#) 247 [entPhysicalMfgDate](#) 247 [entPhysicalMfgName](#) 247 [entPhysicalModelName](#) 247 [entPhysicalName](#) 296 [entPhysicalParentRelPos](#) 247 [entPhysicalSerialNum](#) 247 [entPhysicalSoftwareRev](#) 247 [entPhysicalUris](#) 247 [entPhysicalUUID](#) 247 [entPhysicalVendorType](#) 247 [entSensorMeasuredEntity](#) 49 [entSensorPrecision](#) 49 [entSensorScale](#) 49 [entSensorStatus](#) 49 [entSensorType](#) 49 [entSensorValueUpdateRate](#) 49 [fanName](#) 18 [for_gpe](#) 6088 [gpe](#) 6028 [gpe_site](#) 6082 [hostName](#) 7 [ifAdminStatus](#) 4925 [ifAlias](#) 4925 [ifConnectorPresent](#) 4925 [ifDescr](#) 4925 [ifHighSpeed](#) 4925 [ifHighSpeedDisplay](#) 1914 [ifIndex](#) 4925 [ifLinkUpDownTrapEnable](#) 4925 [ifMtu](#) 4925 [ifName](#) 4925 [ifOperStatus](#) 4925 [ifPhysAddress](#) 4925 [ifPromiscuousMode](#) 4925 [ifSpeed](#) 4925 [ifStackHigherLayer](#) 378 [ifStackLowerLayer](#) 378 [ifStackStatus](#) 378 [ifType](#) 4925 [index](#) 511 [inetCidrRouteDest](#) 2 [inetCidrRouteIfIndex](#) 2 [inetCidrRouteNextHop](#) 2 [inetCidrRoutePfxLen](#) 2 [inetCidrRouteProto](#) 2 [inetCidrRouteType](#) 2 [interface](#) 4925 [ipNetToMediaIfIndex](#) 2 [ipNetToMediaNetAddress](#) 2 [ipNetToMediaPhysAddress](#) 2 [ipNetToMediaType](#) 2 [ipNetToPhysicalIfIndex](#) 2 [ipNetToPhysicalNetAddress](#) 2 [ipNetToPhysicalNetAddressType](#) 2 [ipNetToPhysicalNetAddressV4](#) 2 [ipNetToPhysicalNetAddressV6](#) 2 [ipNetToPhysicalPhysAddress](#) 2 [ipNetToPhysicalRowStatus](#) 2 [ipNetToPhysicalState](#) 2 [ipNetToPhysicalType](#) 2 [lldpLocPortDesc](#) 170 [lldpLocPortId](#) 170 [lldpLocPortIdSubtype](#) 170 [lldpLocPortNum](#) 170 [lldpRemChassisId](#) 2 [lldpRemChassisIdSubtype](#) 2 [lldpRemIndex](#) 2 [lldpRemLocalPortNum](#) 2 [lldpRemPortDesc](#) 2 [lldpRemPortId](#) 2 [lldpRemPortIdSubtype](#) 2 [lldpRemSysCapEnabled](#) 2 [lldpRemSysCapSupported](#) 2 [lldpRemSysDesc](#) 2 [lldpRemSysName](#) 2 [lldpRemTimeMark](#) 2 [managementDomainIndex](#) 11 [romID](#) 7 [source](#) 6088 [sysContactDetails](#) 7 [sysDesc](#) 7 [sysLocation](#) 7 [sysName](#) 6088 [sysObjectId](#) 7 [vlanTrunkPortDynamicStatus](#) 3011 [vmPortStatus](#) 3011 [vmVlan](#) 3011 [vmVlanType](#) 3011 [vtpVlanAreHopCount](#) 11 [vtpVlanBridgeNumber](#) 11 [vtpVlanBridgeType](#) 11 [vtpVlanDot10Said](#) 11 [vtpVlanIfIndex](#) 11 [vtpVlanIndex](#) 11 [vtpVlanIsCRFBackup](#) 11 [vtpVlanMtu](#) 11 [vtpVlanName](#) 11 [vtpVlanParentVlan](#) 11 [vtpVlanRingNumber](#) 11 [vtpVlanState](#) 11 [vtpVlanSteHopCount](#) 11 [vtpVlanStpType](#) 11 [vtpVlanTranslationalVlan1](#) 11 [vtpVlanTranslationalVlan2](#) 11 [vtpVlanType](#) 11 [vtpVlanTypeExt](#) 11 [__field__](#) 6082 [__name__](#) 6082

TAG VALUES

[atIfIndex](#): [253](#) 2

[atNetAddress](#): [0A:1C:50:0B](#) 1 [0A:1C:50:01](#) 1

[atPhysAddress](#): [68 E5 9E 88 BE 7F](#) 1 [AC 7A 56 D9 3F DC](#) 1

[cdpCacheAddress](#): [0A 1C 00 01](#) 2

[cdpCacheAddressType](#): [1](#) 2

[cdpCacheApplianceID](#): [#{CDPCACHEAPPLIANCEID}](#) 2

[cdpCacheCapabilities](#): [00 00 00 29](#) 2

[cdpCacheDeviceId](#): [DBN-5750-MDF-CORE.carhartt.com](#) 2

[cdpCacheDeviceIndex](#): [14](#) 1 [17](#) 1

Pushing SNMP to Its Limits

Bundling for performance is still a work in progress.

Even with bulk requests, a full pass runs about 30 SNMP walks — no performance degradation yet.

One request pulling 30 OIDs at once triggered timeouts no matter what we tried, so we split it into three separate walks.

Often it's the devices, not Zabbix, that can't keep up — we're holding max-repetition count at the default for now.

43 Discoveries



☐ Template	Name ▲	Items	Triggers	Graphs	Hosts	Key	Interval	Type	Status
☐ GPE Network Collection by ATS - Carhartt	SNMP AT Table Info : SNMP AT Table Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.at_table.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP CDP Address Info : SNMP CDP Address Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.cdp_address.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP CDP Cache Info : SNMP CDP Cache Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.cdp_cache.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP CDP Interface Info : SNMP CDP Interface Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.cdp_if.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP CEFS FRUPower Module Info : SNMP CEFS FRUPower Module Info Discovery	Item prototypes 3	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.cefs_frup_module.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP CEFS Module Info : SNMP CEFS Module Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.cefs_module.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP CLAG Agg Port Info : SNMP CLAG Agg Port Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.clag_agg_port.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device CPU Info : SNMP Device CPU Info (avg) Discovery	Item prototypes 2	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_cpu.avg.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device CPU Info : SNMP Device CPU Info (cpm) Discovery	Item prototypes 2	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_cpu.cpm.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device CPU Info : SNMP Device CPU Info (fg) Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_cpu.fgSys.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device Info : SNMP Device Info Discovery	Item prototypes 7	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.device.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device Memory Info : SNMP Device Memory Info (cisco) Discovery	Item prototypes 3	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_mem.cisco.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device Memory Info : SNMP Device Memory Info (cpm) Discovery	Item prototypes 3	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_mem.cemp.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device Memory Info : SNMP Device Memory Info (fg) Discovery	Item prototypes 2	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_mem.fgSys.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device Sensor Info : SNMP Device Sensor Info Discovery	Item prototypes 3	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_sensor.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Device Temperature Info : SNMP Device Temperature Info Discovery	Item prototypes 2	Trigger prototypes	Graph prototypes 1	Host prototypes	gpe.net.device_temp.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Dot1 Base Port Mapping Info : SNMP Dot1 Base Port Mapping Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.dot1base_port_map.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP dot1dTpFdbTable Info : SNMP dot1dTpFdbTable Info Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.dot1dtpfdb.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Fan Info : SNMP Fan Info (cefc) Discovery	Item prototypes 2	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.fan.cefc_snmp_fan.discovery		Dependent item	Enabled
☐ GPE Network Collection by ATS - Carhartt	SNMP Fan Info : SNMP Fan Info (cisco) Discovery	Item prototypes 1	Trigger prototypes	Graph prototypes	Host prototypes	gpe.net.fan.cisco_snmp_fan.discovery		Dependent item	Enabled
		123▶				Displaying 1 to 20 of 43 found			

Before a Single Relationship Can Be Visualized

CDP or LLDP Enabled Everywhere

Neighbor discovery protocols must be turned on across every device in scope — no exceptions.

SNMP OIDs Configured Correctly

Zabbix templates need to walk the right CDP/LLDP MIBs consistently, across every vendor.

Noise Kept Under Control

High-volume neighbor data can flood a monitoring environment if it isn't managed deliberately.

Template Generator

- Allows us to maintain a source of truth for collection
- The output of this data-set is a Zabbix template.
- Allows for the generator data-set to be smaller than a Zabbix template.
- Allows for consistent naming and tagging across technologies.
- Adding new metrics might be 1-3 lines in the generator and builds all the necessary discoveries and item prototypes.

OUTPUT = Consistent Zabbix template across all network vendors

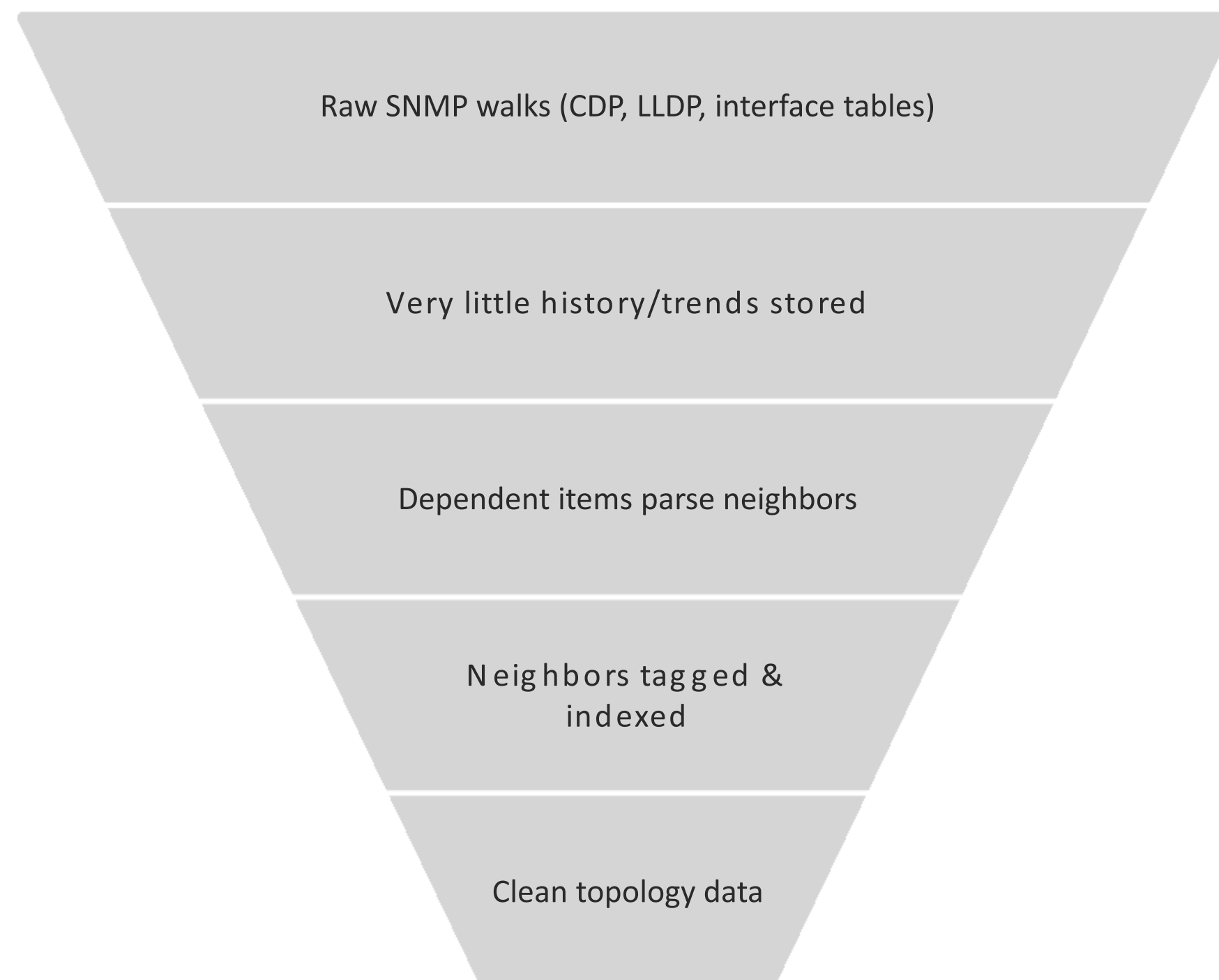
```

1 gam_network_cpu:
2   type: walk_gen
3   name: 'SNMP Device CPU Info'
4   description: 'Device Info.'
5   history: '0'
6   trends: '0'
7   timeout: 300s
8   delay: '{{intervals.trend}}'
9   # snmp_oid: 'walk[1.3.6.1.4.1.9.2.1.57,1.3.6.1.4.1.9.2.1.58,1.3.6.1.4.1.9.9.109.1.1.1.7,1.3.6.1.4.1.9.9.109.1.1.1.8]'
10  snmp_oid: 'walk[1.3.6.1.4.1.12356.101.4.1.3,1.3.6.1.4.1.9.2.1.57,1.3.6.1.4.1.9.2.1.58,1.3.6.1.4.1.9.9.109.1.1.1.7,1.3.6.1.4.1.9.9.109.1.1.1.8]'
11  tags:
12    - tag: component
13      value: cpu
14  key:
15    prefix: 'gpe.net.device_cpu'
16    fetch: 'walk'
17  item_name_prefix: "CPU #{SNMPINDEX}"
18  default_units: '%'
19  gauges:
20    fgSysCpuUsage:
21      oid: 1.3.6.1.4.1.12356.101.4.1.3
22      name: utilization
23    cpmCPUTotal1minRev:
24      oid: 1.3.6.1.4.1.9.9.109.1.1.1.7
25      name: 1m utilization
26    cpmCPUTotal5minRev:
27      oid: 1.3.6.1.4.1.9.9.109.1.1.1.8
28      name: 5m utilization
29    avgBusy1:
30      oid: 1.3.6.1.4.1.9.2.1.57
31      name: 1m utilization
32    avgBusy5:
33      oid: 1.3.6.1.4.1.9.2.1.58
34      name: 5m utilization
35  configs:
36    index:
37      value: '#{SNMPINDEX}'
38  keep:
39    - fgSysCpuUsage
40    - avgBusy5
41    - cpmCPUTotal5minRev
42  discovery_rules:
43    fgSys:
44      name: 'SNMP Device CPU Info (fg)'
45      at_least_one_exists:
46        - fgSysCpuUsage
47      metrics:
48        - fgSysCpuUsage
49    cpm:
50      name: 'SNMP Device CPU Info (cpm)'
51      at_least_one_exists:
52        - cpmCPUTotal1minRev
53      metrics:
54        - cpmCPUTotal1minRev
55        - cpmCPUTotal5minRev
56    avg:
57      name: 'SNMP Device CPU Info (avg)'
58      at_least_one_exists:
59        - avgBusy1
60      metrics:
61        - avgBusy1
62        - avgBusy5

```

Taming the Noise

Our gpesnmp_network template turns raw neighbor data into clean, topology-ready records — without bloating the Zabbix database.



From Pilot to Production

01

Scope with Network Engineers

Identify device populations, confirm CDP/LLDP support, and agree on polling impact upfront.

02

Proceduralized Onboarding

A standard checklist for enabling protocols, applying the template, and validating neighbor data before go-live.

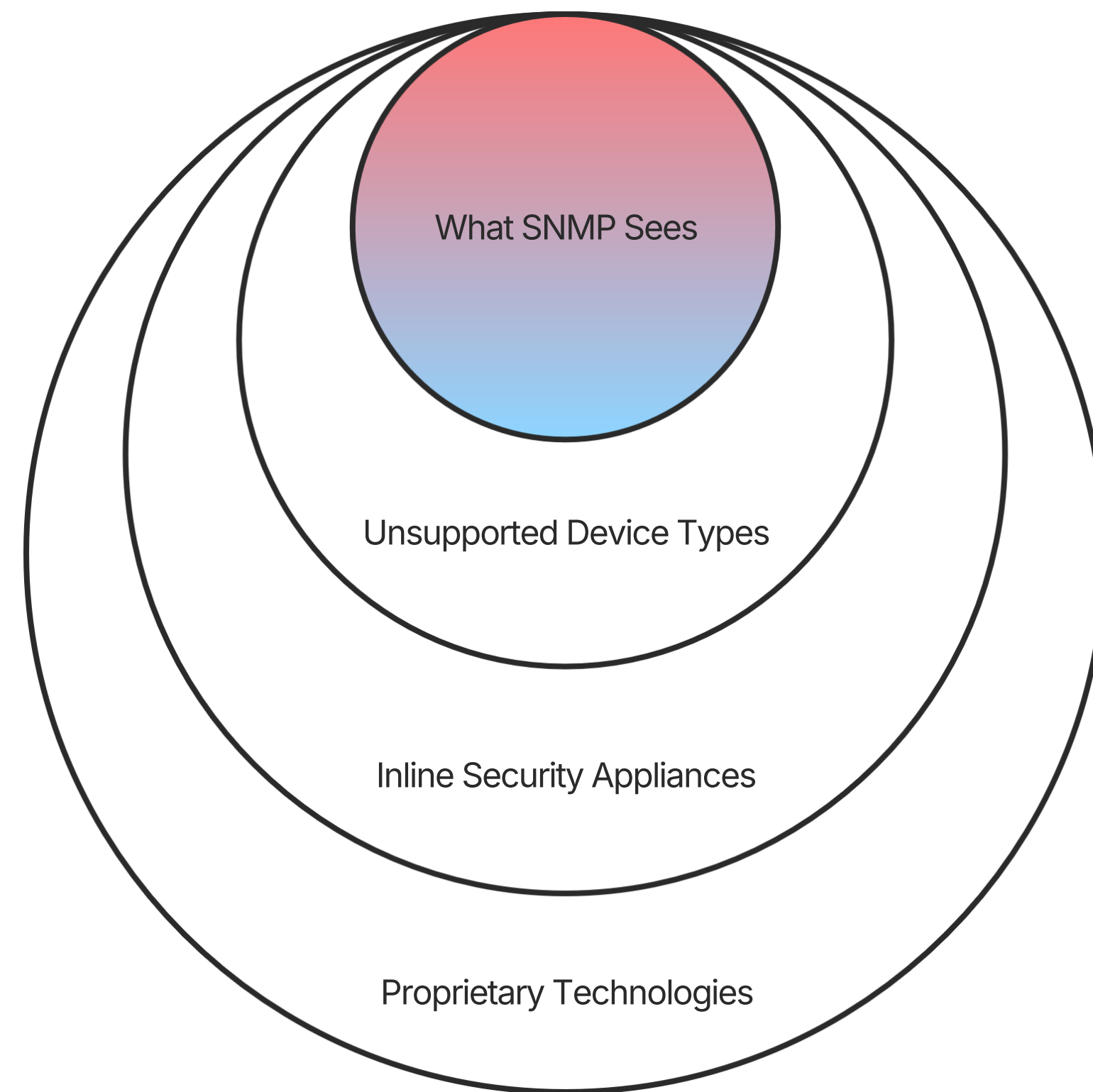
03

Continuous Validation

Ongoing checks for new devices, protocol drift, and template coverage gaps.

What Zabbix Can't See

CDP/LLDP and SNMP get you most of the way — a complete map still has to account for what monitoring can't fully see.



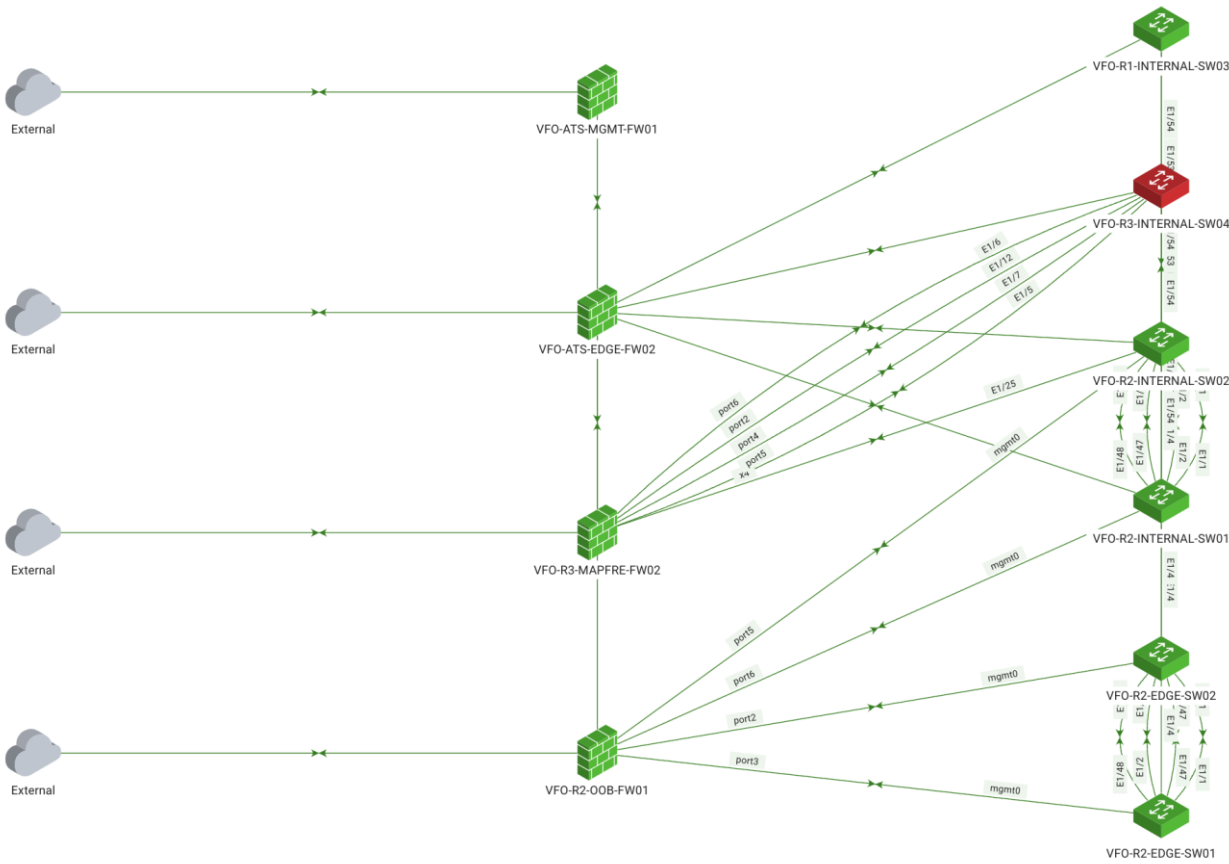
Use Case: From Zabbix to Topology Platform

Zabbix collects and parses CDP/LLDP neighbor data over SNMP, tags it, and prepares structured neighbor records — without holding it in the core database.

☐	Name ▲	Triggers	Key	Interval	History	Trends	Type
☐	*** Hardware operating system		gpe.net.hw.os	5m	31d		SNMP agent
☐	*** Hardware serial number		gpe.net.hw.serialnumber	5m	31d		SNMP agent
☐	*** Hardware system location		gpe.net.hw.location	5m	31d		SNMP agent
☐	*** SNMP AT Table Info		gpe.net.at_table.walk	5m	0		SNMP agent
☐	*** SNMP CDP Address Info		gpe.net.cdp_address.walk	5m	0		SNMP agent
☐	*** SNMP CDP Cache Info		gpe.net.cdp_cache.walk	5m	0		SNMP agent
☐	*** SNMP CDP Interface Info		gpe.net.cdp_if.walk	5m	0		SNMP agent
☐	*** SNMP CEFS FRUPower Module Info		gpe.net.cefs_frup_module.walk	5m	0		SNMP agent
☐	*** SNMP CEFS Module Info		gpe.net.cefs_module.walk	5m	0		SNMP agent
☐	*** SNMP CLAG Agg Port Info		gpe.net.clag_agg_port.walk	5m	0		SNMP agent
☐	*** SNMP Device CPU Info		gpe.net.device_cpu.walk	5m	0		SNMP agent
☐	*** SNMP Device Info		gpe.net.device.walk	5m	0		SNMP agent
☐	*** SNMP Device Memory Info		gpe.net.device_mem.walk	5m	0		SNMP agent
☐	*** SNMP Device Sensor Info		gpe.net.device_sensor.walk	5m	0		SNMP agent
☐	*** SNMP Device Temperature Info		gpe.net.device_temp.walk	5m	0		SNMP agent
☐	*** SNMP Dot1 Base Port Mapping Info		gpe.net.dot1base_port_map.walk	5m	0		SNMP agent
☐	*** SNMP dot1dTpFdbTable Info		gpe.net.dot1dtpfdb.walk	5m	0		SNMP agent
☐	*** SNMP Fan Info		gpe.net.fan.walk	5m	0		SNMP agent
☐	*** SNMP IF Receive Address Info		gpe.net.if_recv_address.walk	5m	0		SNMP agent
☐	*** SNMP Interface Info		gpe.net.if.walk	5m	0		SNMP agent



Neighbor and interface data is streamed out through Zabbix's external data streaming capability, straight into Galileo — where it becomes live, queryable topology maps.



Inside Zabbix

Collection & normalization

Galileo

(galileosuite.com)
Topology Visualization + Capacity planning

Performance Modifications



Performance-Oriented Codebase Changes

Proxy Postgres + TimescaleDB

- <https://github.com/zabbix/zabbix/pull/155>
 - Add partitioning by row id to schema for proxy_history table (no compression)
 - Update housekeeper to drop full partitions rather than issue DELETE
- <https://github.com/zabbix/zabbix-docker/pull/1748>
 - Adds Postgres + TimescaleDB option to container image build framework

HTTP Streaming

- All data can be streamed regardless of retention
- Use Case: Stream data to a third-party that you do not care to store in your Zabbix DB; allows for collection framework - not the storage.

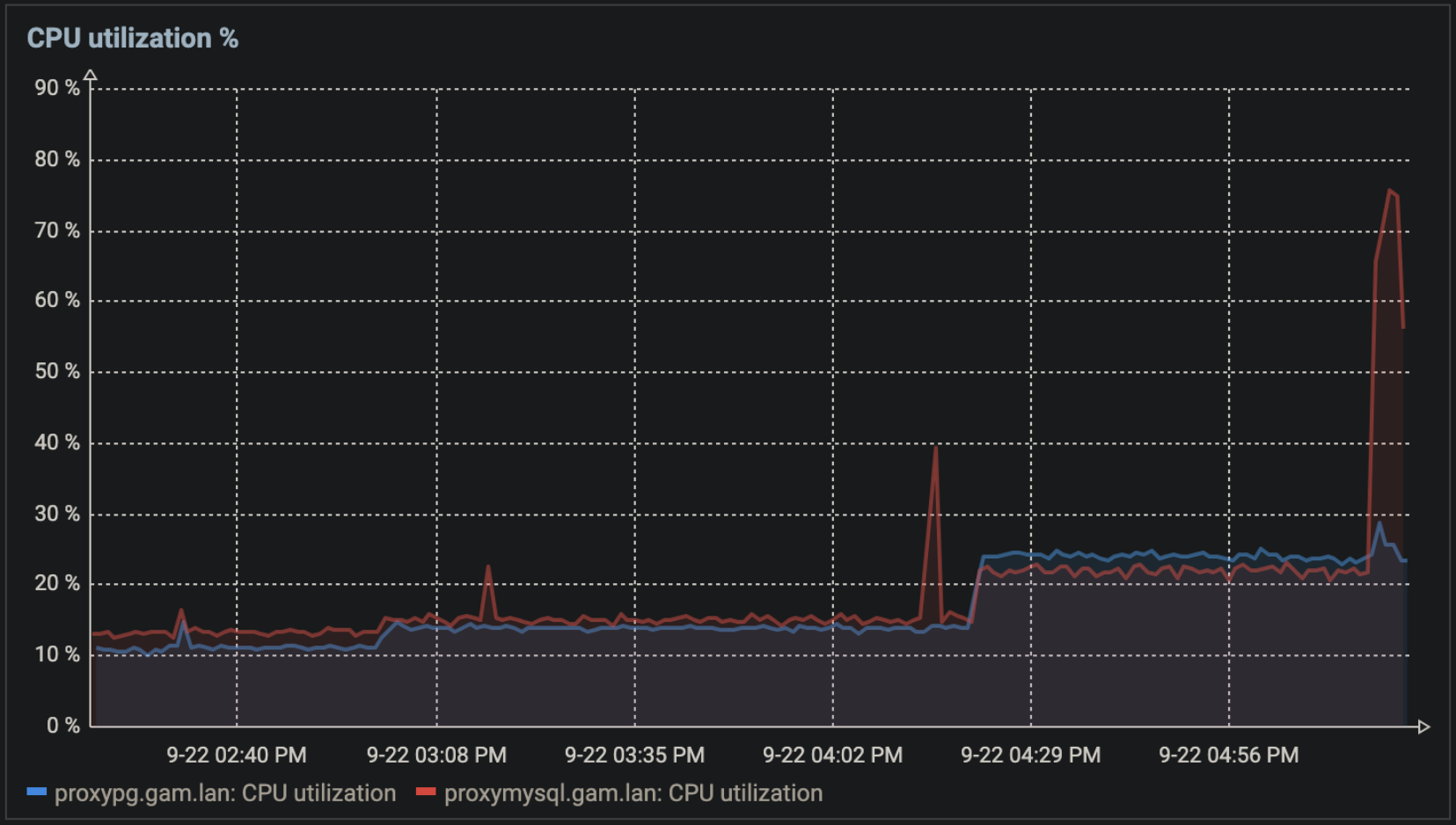
Expanded Server Stats

- New server-side instrumentation counts every collected value unconditionally — previously it only tracked collection when a VPS Limit was configured. That's exposed as collected_total alongside the existing written_total and limit — so zero-retention streamed data still shows up in total throughput even though it never lands in written_total, and operators can see how close the server is running to its configured VPS Limit.
- Written Num vs Collected Num

Performance

17:15 to 17:18	PG (.88)	MySQL (.89)
Housekeeper busy	one blip of 6.2% at 17:16 (a few seconds)	46% → 100% → 100% → 35%: the delete ran for about 3 minutes
History syncer (new writes)	steady at 4-5%	2% → 37% → 70% : writes are backing up behind the delete
History write cache	0.1%	spiked to 12.8% at 17:17, back to 0
CPU	24% → 29% → 26%	22% → 66% → 70% → 76% , still 75% at 17:19
iowait	0.6 → 2.9	0.3 → 11 → 25
Memory	flat at 38.9%	64% → 71% → 77% → 93%

- We ran the identical load against both proxies at once: 1,000 values/sec from 15:00 to 16:21 (about 1h20m), then 5,000 values/sec from 16:21 to 18:51 (2.5h). Each proxy's housekeeper runs once an hour and deletes the values it has already forwarded to the server, so each pass had to clear about an hour of data.
- At 1,000/sec the MySQL proxy took 23 seconds to delete 3.6M rows at 16:15, where PG took 0.3 seconds for 4.0M. At 5,000/sec MySQL's 17:15 pass pegged its housekeeper for about 3 minutes and pushed memory to 93%. On the 18:18 pass it stopped accepting data entirely from 18:21 to 18:23 and dropped about 90,000 values. The PG proxy handled every pass with no visible impact."



Key Takeaways

Takeaway 1

Groundwork isn't optional. It's the project.

Takeaway 2

Operationalize the onboarding process.

Takeaway 3

High-throughput data needs its own path.

Thank You

Questions?

andy.wojnarek@theatsgroup.com

<https://www.theatsgroup.com/zabbix>