HANDS-ON WALKTHROUGH · SPLUNK S2S

Is every receiver accepting, and is my forwarding tier keeping up?

One tenant watches the Splunk-to-Splunk receiving side of a deployment — the indexers, the intermediate forwarders and the pools they form — from the telemetry Splunk already writes: nothing to install, nothing new to ingest.

Captured live on TrackMe 2.4.22 · a four-peer indexer cluster on two sites, a heavy forwarder tier, twenty universal forwarders

THE PROBLEM

Forwarding fails in the middle — and the senders only see a stall

A receiver that stops listening, an intermediate that pauses its output, a queue that backs up: the forwarders retry, data waits on disk, nobody sees which tier is the cause.

Receivers that refuse

Listening ports stopped on blocked queues, a member silent while its pool reports, a pool short of members.

Back pressure

A queue of the pipeline saturated: which stage cannot keep up, and whether the cause is downstream, local or a surge.

An intermediate tier in trouble

Output paused, every sender behind it held; a single destination where it usually load balances.

What a Splunk S2S tenant does

A dedicated tenant type for the S2S receiving side of one Splunk deployment. TrackMe reads what Splunk already writes — the receivers' and intermediates' connection lines, queues and output, their splunkd.log and health report — with bounded searches every five minutes, and raises findings from twenty-two built-in policies on receivers, intermediates, pools and destinations. It only reads: nothing is installed or changed, no forwarder is ever called.

THE MODEL

Roles from signatures, pools from your clusters, judged every five minutes

1 Source

metrics.log, splunkd.log, health report and _introspection — on this search tier or through a remote account.

2 Discover

Roles from signatures; indexer clusters (one pool per site) and intermediate families become pools; yours by pattern, output group or hand.

3 Measure

Every 5 minutes: inbound, senders, posture, queues and the bottleneck, output, churn, CPU and memory.

4 Decide

22 built-in policies on receivers, intermediates, pools and destinations, plus custom SPL policies.

5 Act

Finding groups, notifications, findings actions, the AI Agent, Topology Studio.

Back pressure traced to its origin

A queue that fills, then the queues before it, until the receiver pauses its senders or stops listening: TrackMe reads every queue of the path and names the bottleneck — the stage after the furthest-downstream saturated queue — and, on an intermediate, the origin: downstream, local or an upstream surge.

Planned work is not an incident: while an indexer cluster is in maintenance mode or a rolling restart, its pool's findings wait.

THE PLAN

What we are going to do — one tenant, four parts

PART 1

Setup

- The Splunk S2S card
- Discovery and the manager account
- Pools: automatic, then yours

PART 2

Pools, hosts & data path

- The Pools tab and Blocked now
- Distribution and pipeline
- The data path

PART 3

Findings

- A heavy forwarder stops
- How it is evaluated
- The AI Agent

PART 4

Operate

- The policy catalogue
- Traffic ML models
- Collection health

Every screen is the product as shipped, on a lab of twenty universal forwarders, four heavy forwarders and four indexers.

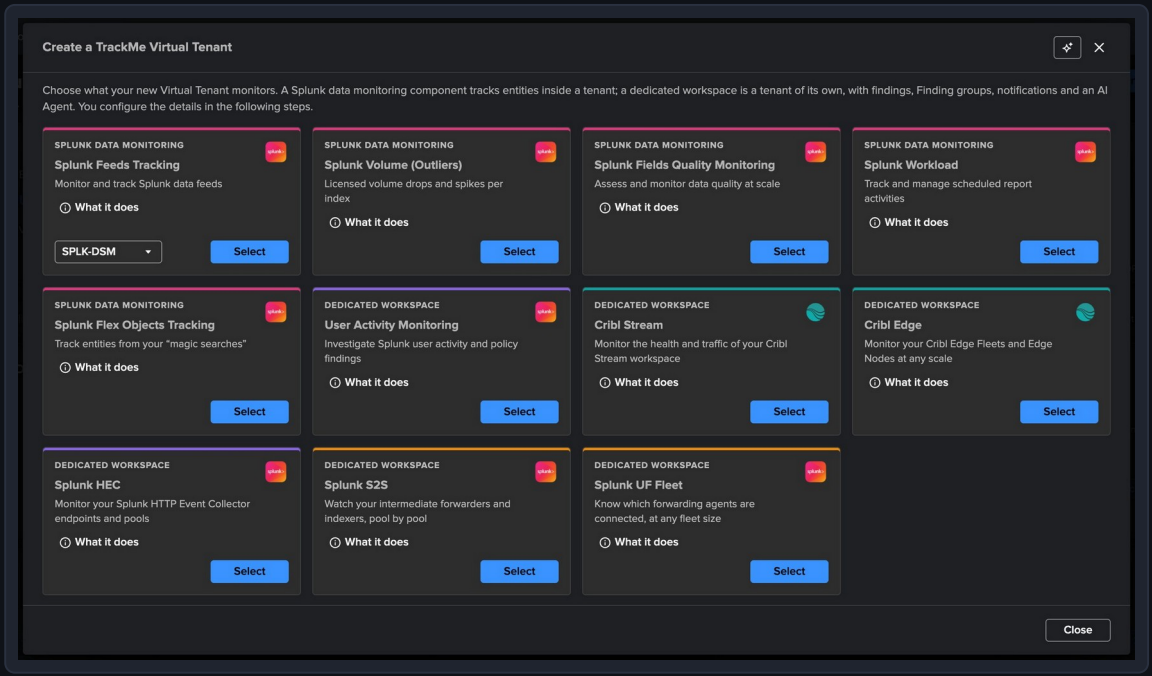
01

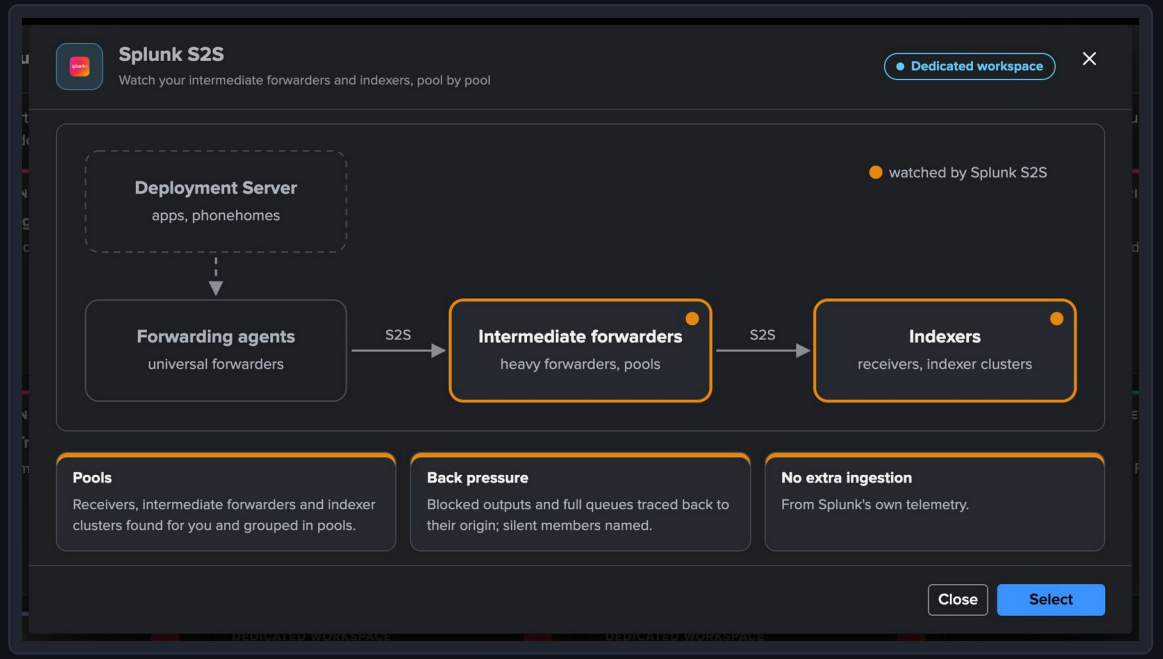
PART 1 · SETUP

One source, pools found for you, one to make your own

The Splunk S2S card, the discovery of the receivers and the indexer cluster, the manager account, the automatic pools and the heavy forwarder tier configured.

A dedicated tenant — pick Splunk S2S





1. **The Splunk S2S card**
   Watch your intermediate forwarders and indexers, pool by pool.

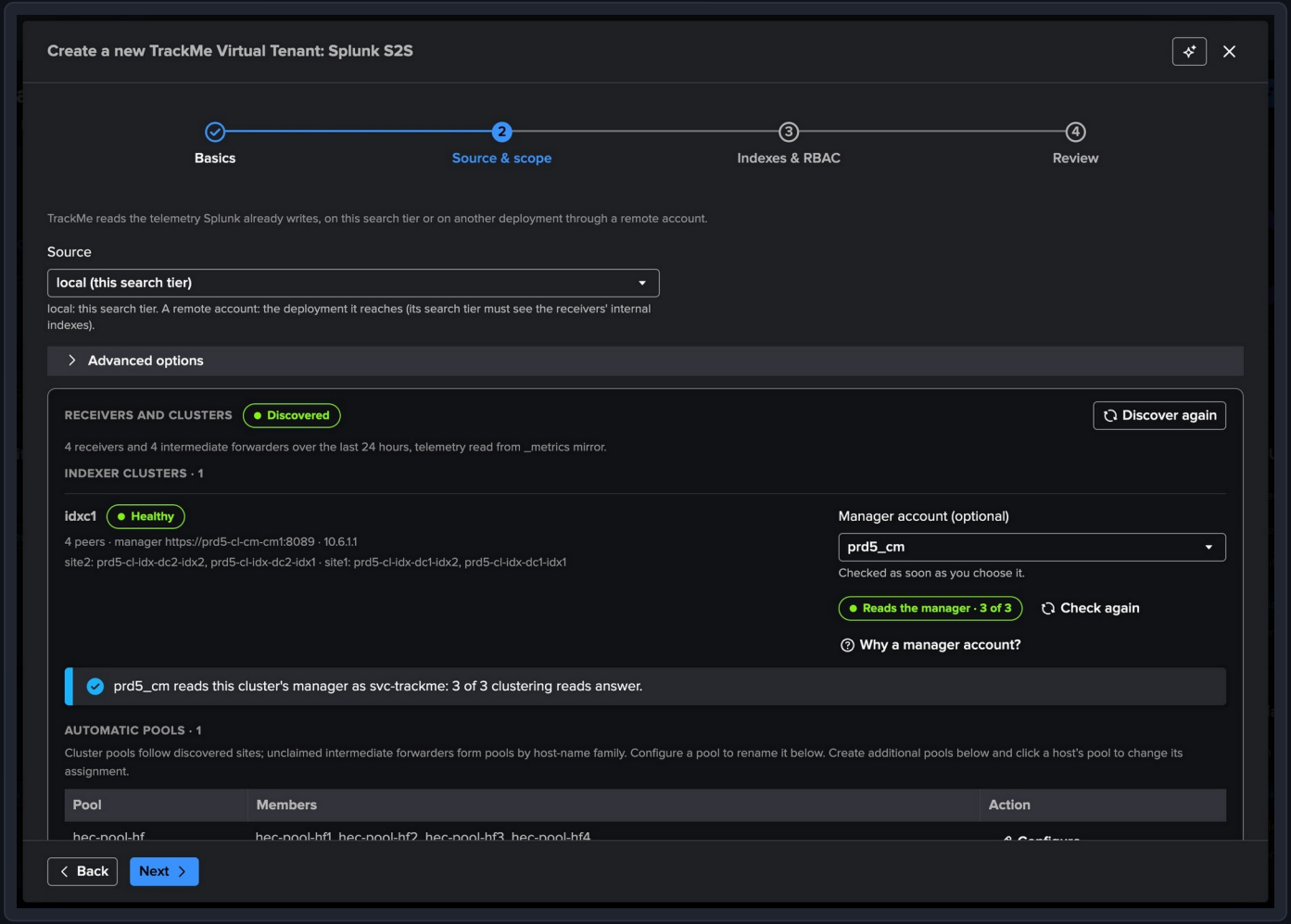
2. **What it does**
   The forwarding path, with the intermediate forwarders and the indexers watched.

3. **Four steps**
   Basics, Source & scope, Indexes & RBAC, Review.

4. **Licence**
   Enterprise, Unlimited, the Unlimited trial or Developer; not Foundation.

PART 1 · SETUP

The discovery finds the indexer cluster — a pool per site, without configuration



1 **Source**

This search tier, or another deployment through a remote account.

2 **Receivers and clusters**

4 receivers and 4 intermediate forwarders over the last 24 hours.

3 **Indexer cluster idxc1**

Healthy, four peers on two sites: one receiver pool per site, its size from the cluster.

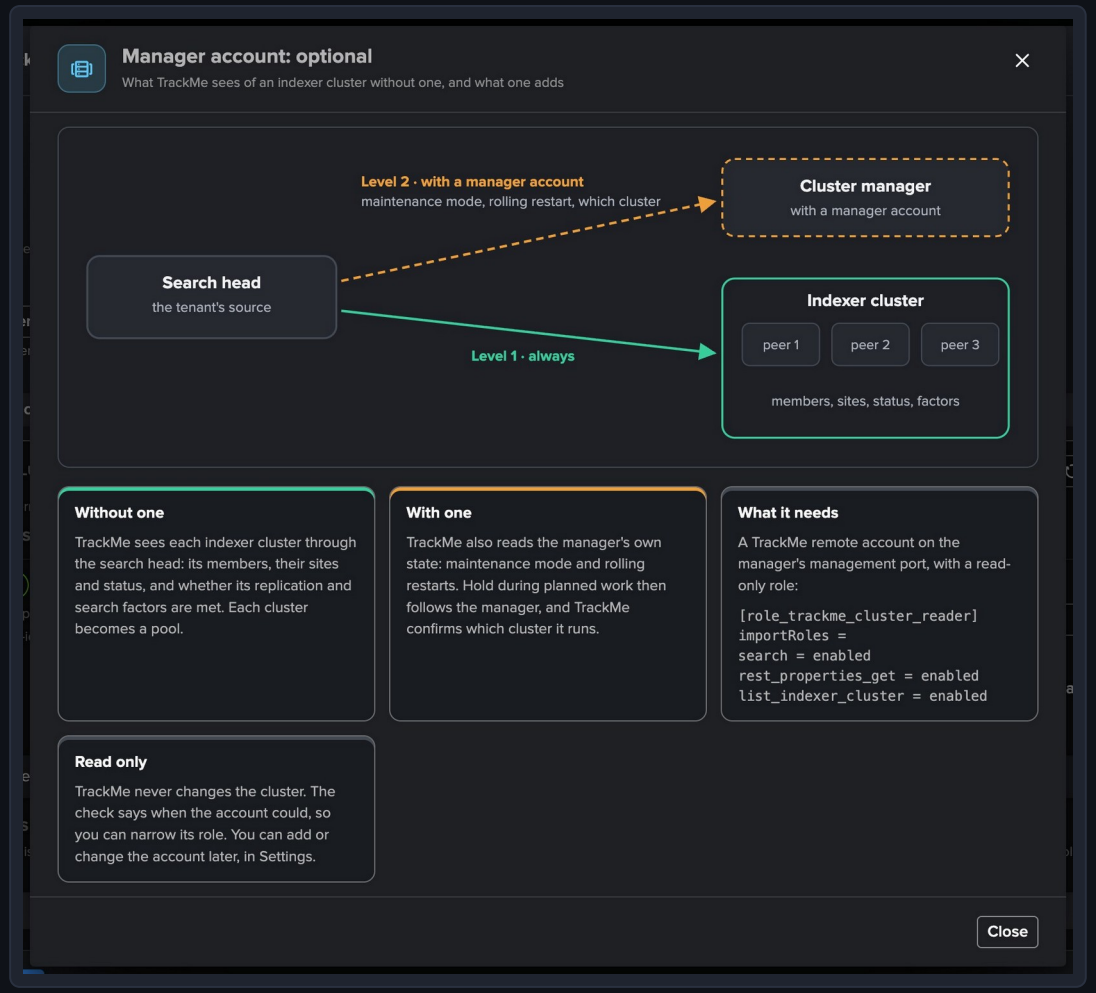
4 **Manager account (optional)**

Checked as soon as it is chosen: reads the manager, 3 of 3.

5 **Least privilege**

A read-only role is enough; an account that could change the cluster gets an advisory.

Why a manager account? Planned work is not an incident



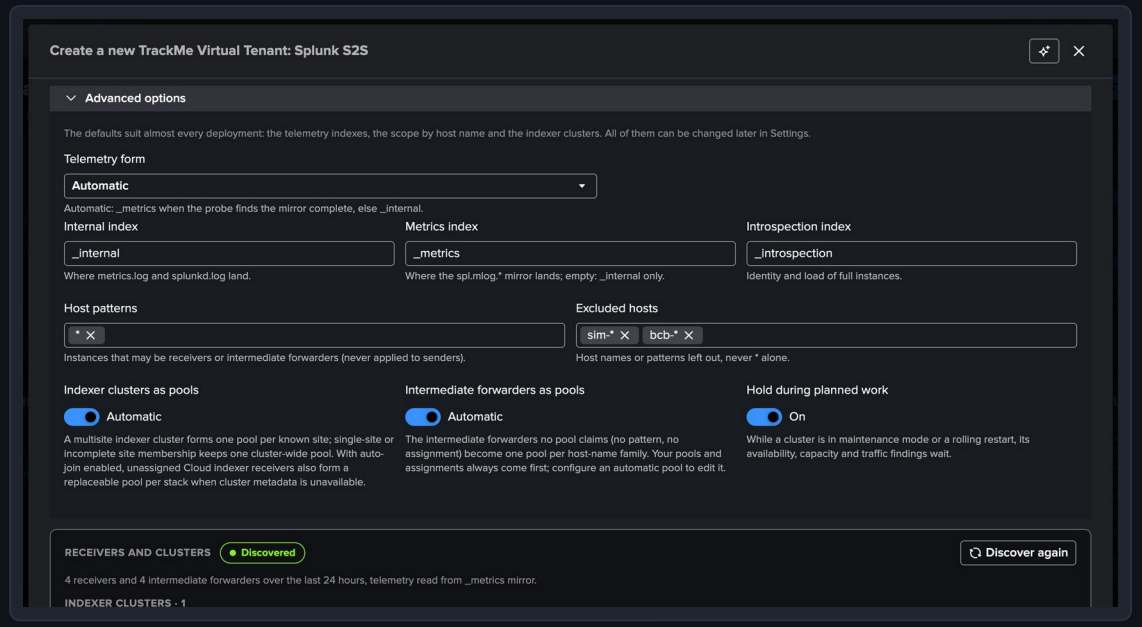
1. **Level 1 · always**
   The search head's view: members, sites, status, factors. Each cluster becomes a pool.

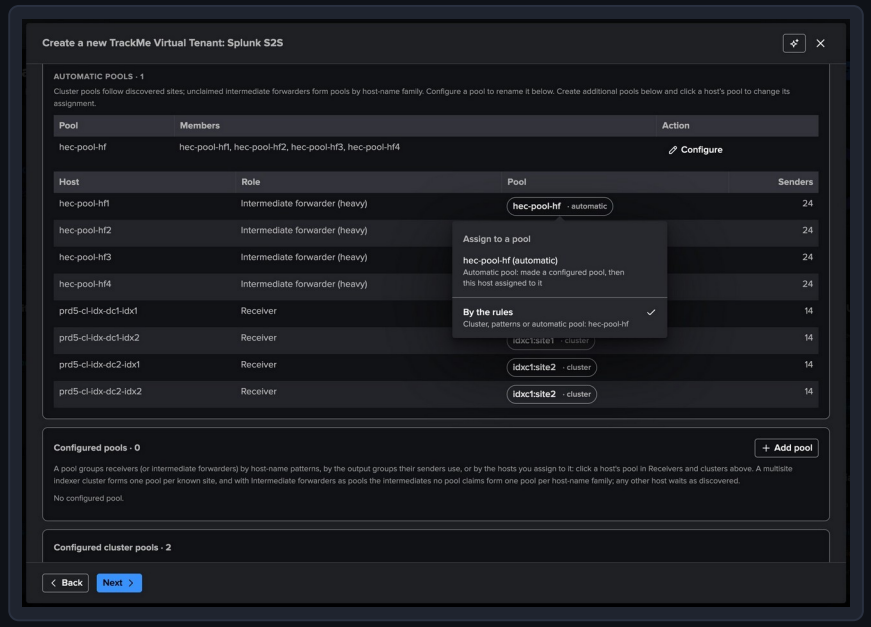
2. **Level 2 · with a manager account**
   Maintenance mode, rolling restart, which cluster.

3. **Hold during planned work**
   Availability, capacity, balance, churn and traffic findings wait during planned work.

4. **Read only**
   TrackMe never changes the cluster; the account can be added later in Settings.

Advanced options and the pools — the defaults suit most deployments





1 Telemetry form

Automatic: the _metrics mirror when complete, else _internal.

2 Host patterns, excluded hosts

Tokens: type a pattern, press Enter. Here sim-* and bcb-* are left out.

3 Automatic pools

Indexer clusters and intermediate forwarders as pools, both Automatic.

4 Pool pills

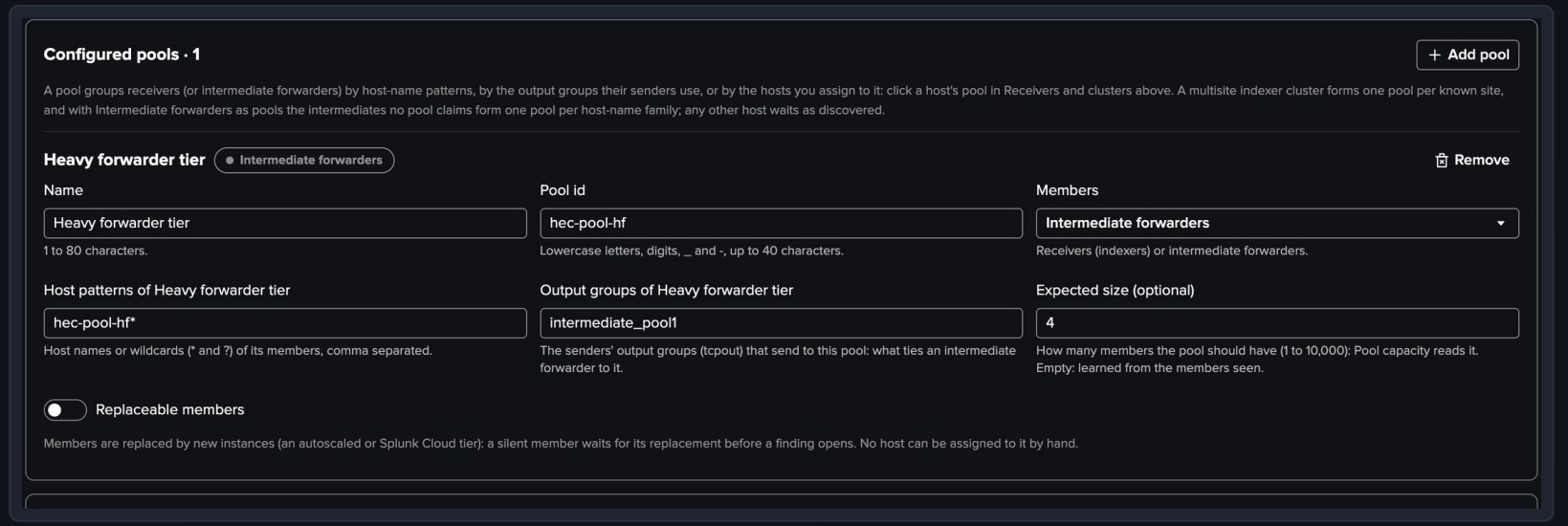
Every host with its pool: click it to assign the host, or By the rules.

5 Same editor as Settings

Automatic pools with Configure, then your configured pools.

PART 1 · SETUP

Configure the automatic pool — the heavy forwarder tier, tied by its output group



1. **Configure**
   The automatic pool hec-pool-hf becomes yours, same id, same history.

2. **Output groups**
   intermediate_pool1: the forwarders' tcpout group that sends to this pool.

3. **Expected size**
   4: Pool capacity compares the members up with it.

4. **Replaceable members**
   For an autoscaled tier: a silent member waits for its replacement.

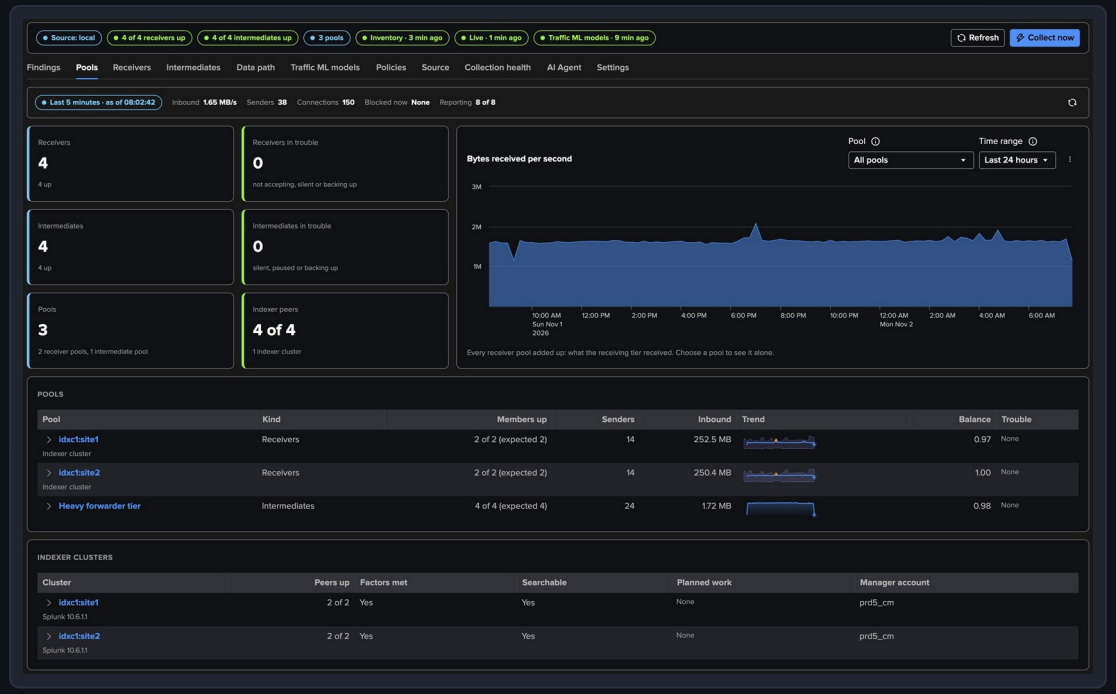
02

PART 2 · POOLS, HOSTS & DATA PATH

Every pool, every host, every hop

The Pools tab, how a pool spreads its load, a receiver's pipeline queue by queue, and the data path from sender types to receivers.

The Pools tab — the last five minutes, then every pool



1 Status strip

Receivers and intermediates up, the pools, the Inventory, Live and Models jobs.

2 Last 5 minutes

Inbound per second, senders, connections, blocked now, hosts reporting.

3 Pools

idxc1:site1, idxc1:site2 and the Heavy forwarder tier: members against expected, senders, inbound, balance.

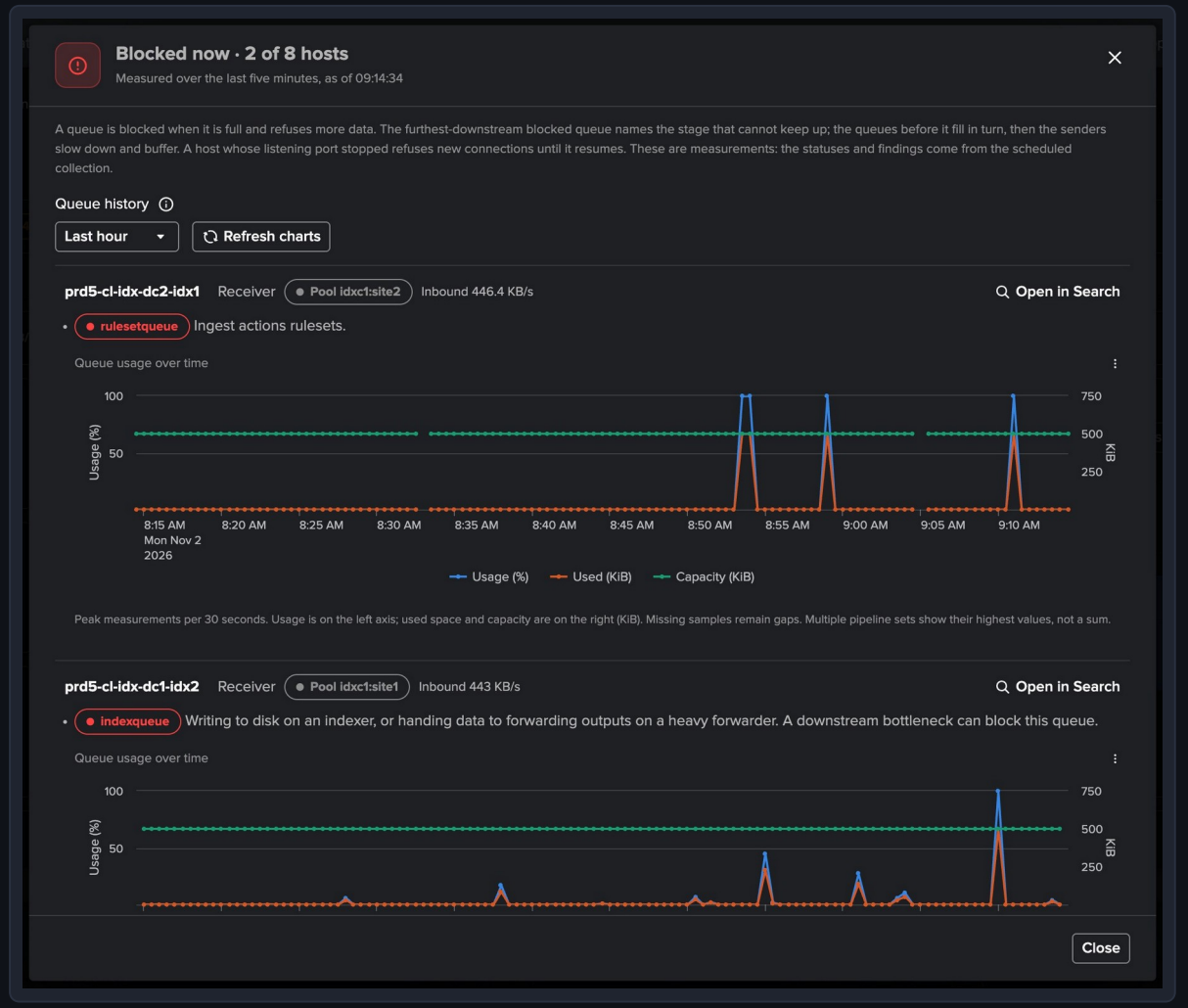
4 Indexer clusters

One row per site: peers up, factors met, searchable, planned work, manager account.

5 Breakdowns

The Splunk versions of the hosts, the intermediate types.

Blocked now — which host, which queue, and its history



1. **Blocked now · 2 of 8 hosts**
   The last five minutes strip: the pill opens what is blocked.

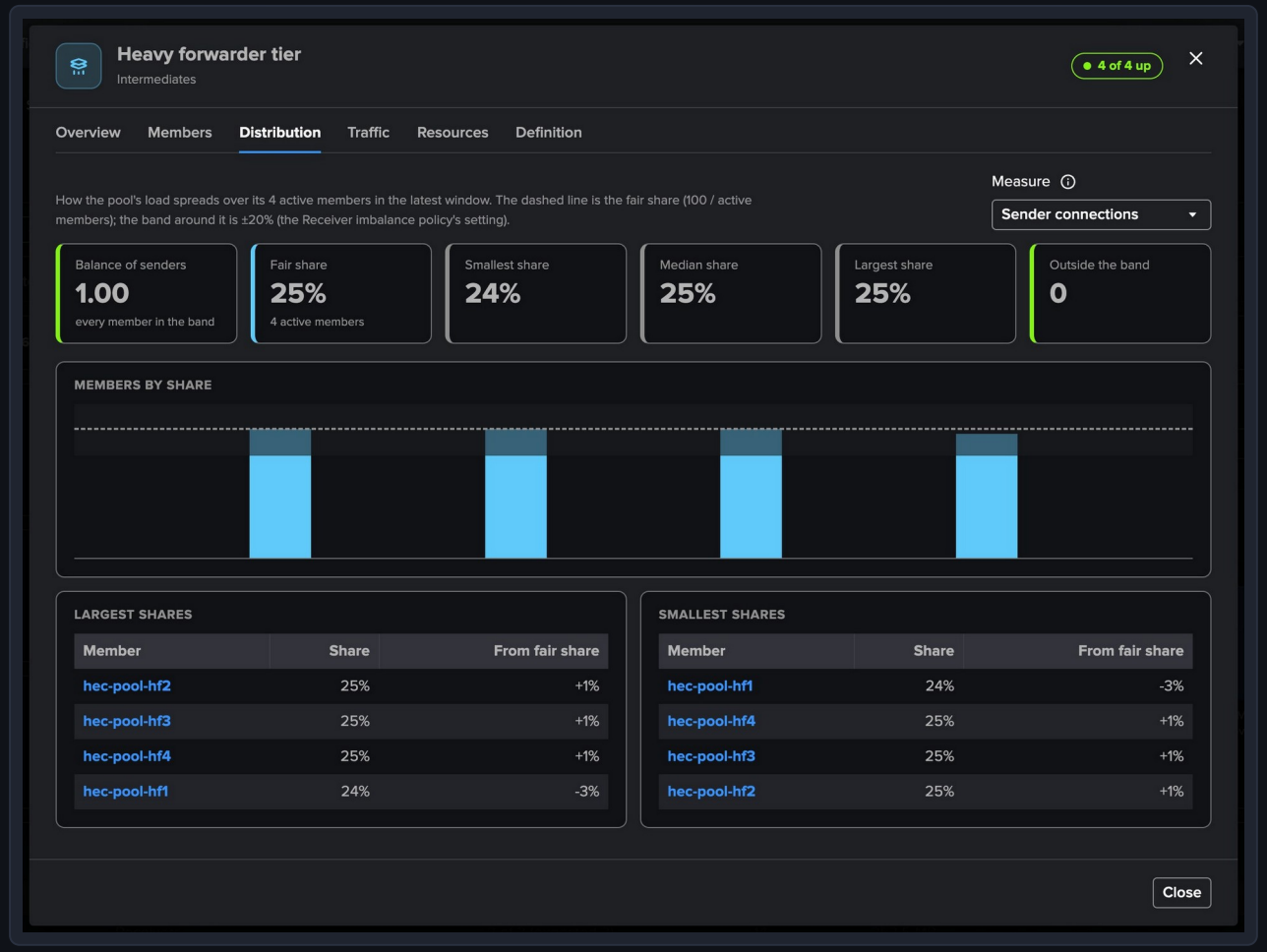
2. **Each host**
   Its role, its pool, its inbound, and Open in Search on its queues.

3. **Each blocked queue**
   What that stage does: indexqueue writes to disk, rulesetqueue applies ingest actions.

4. **Queue history**
   Usage, used and capacity, from 15 minutes to 24 hours.

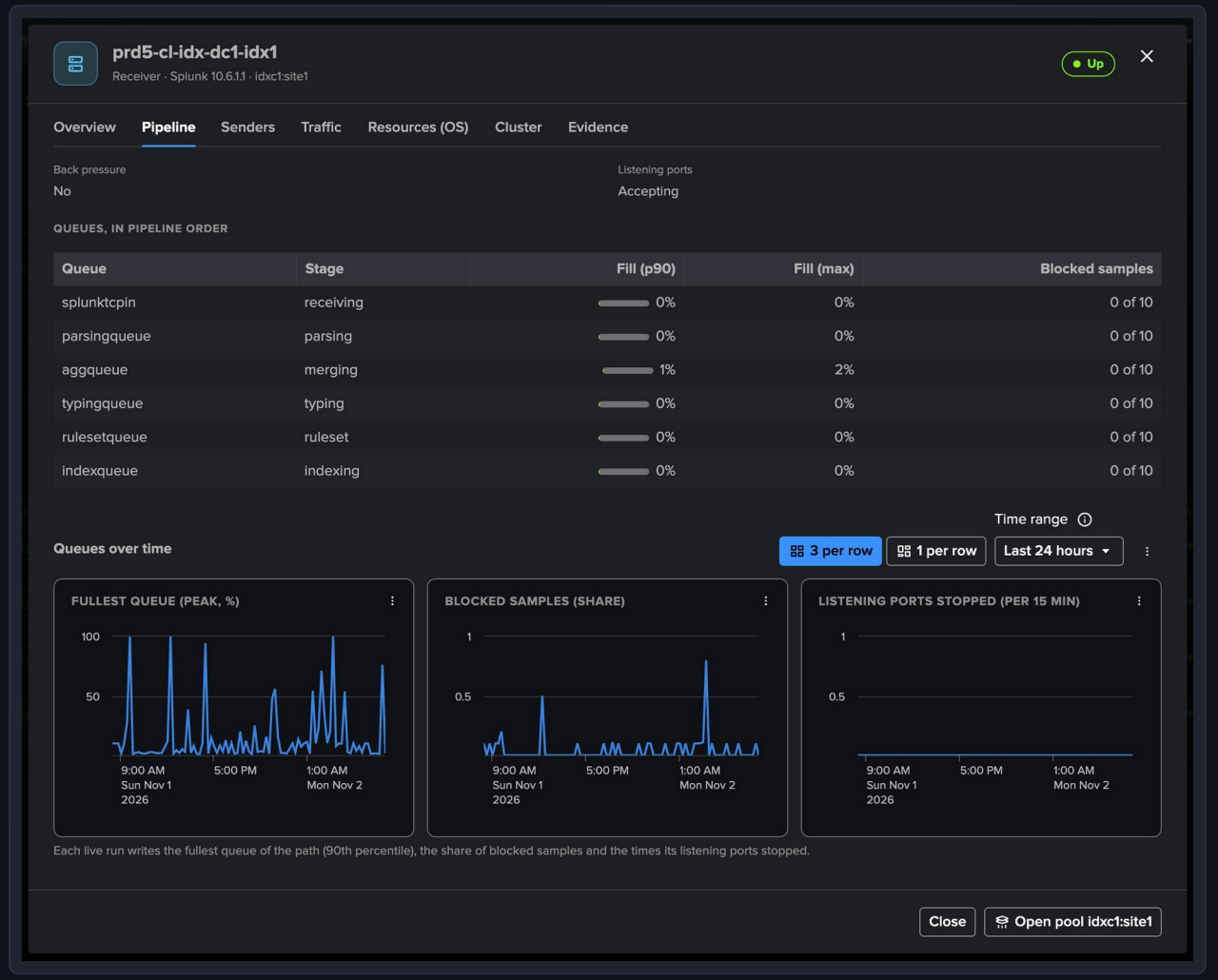
5. **A measurement**
   Short peaks show here; statuses and findings come from the collection.

How the pool spreads its load — the fair share and the band



1. **Fair share**
   25 % for the four heavy forwarders: the dashed line.
2. **The band**
   ±20 %, the Receiver imbalance policy's setting.
3. **Balance of senders**
   Jain's fairness index: 1.00, every member in the band.
4. **Measure**
   Sender connections (the policy's own measure) or volume.
5. **Largest and smallest**
   Each member's share and its distance from the fair share.

A receiver's pipeline — queue by queue, in order



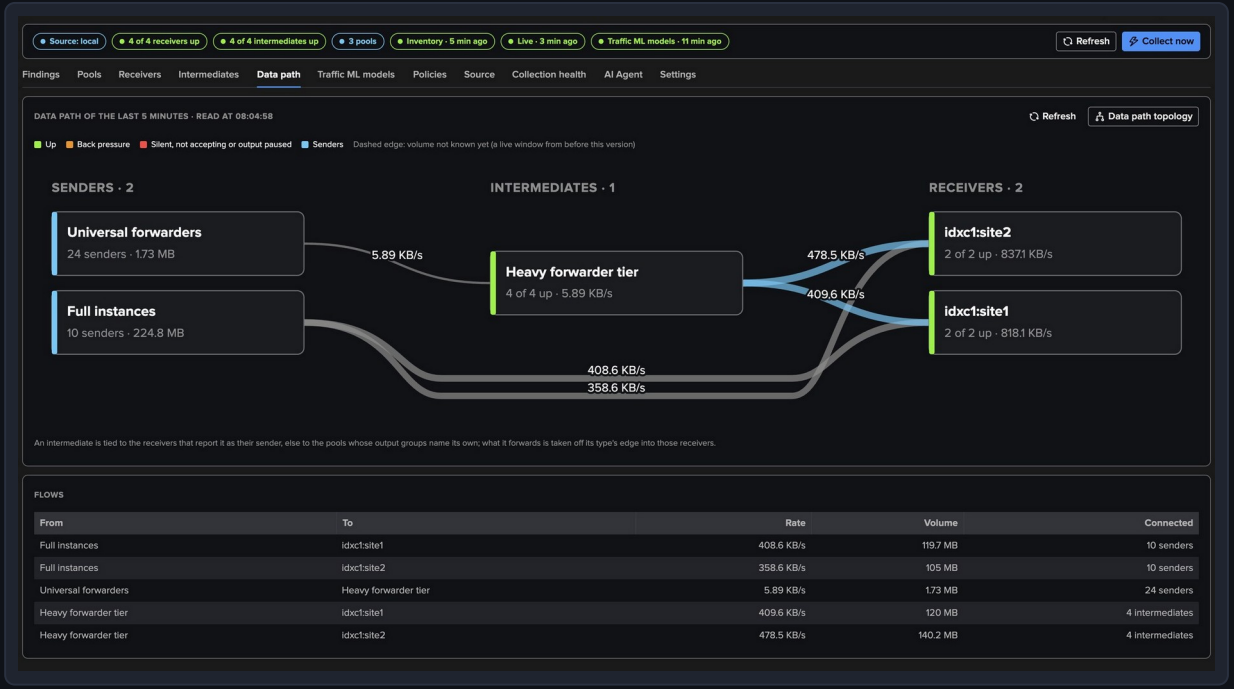
1. **Back pressure, listening ports**
   No back pressure; accepting its senders.

2. **Queues in pipeline order**
   splunktcpin, parsing, aggregation, typing, ruleset, indexing.

3. **Fill and blocked samples**
   90th-percentile and peak fill, blocked share of the samples.

4. **Over time**
   The fullest queue over the last 24 hours.

The data path — sender types, intermediates, receivers



1 Three columns

Sender types, intermediates, receivers; a pool as one node.

2 Edges by volume

Width and KB/s labels: 24 universal forwarders into the Heavy forwarder tier.

3 Tied by output group

An intermediate is tied to the receivers that report it, else by its pool's output groups.

4 Read now

The newest window, read with your rights; nothing saved.

5 Data path topology

Saves it to Topology Studio on request.

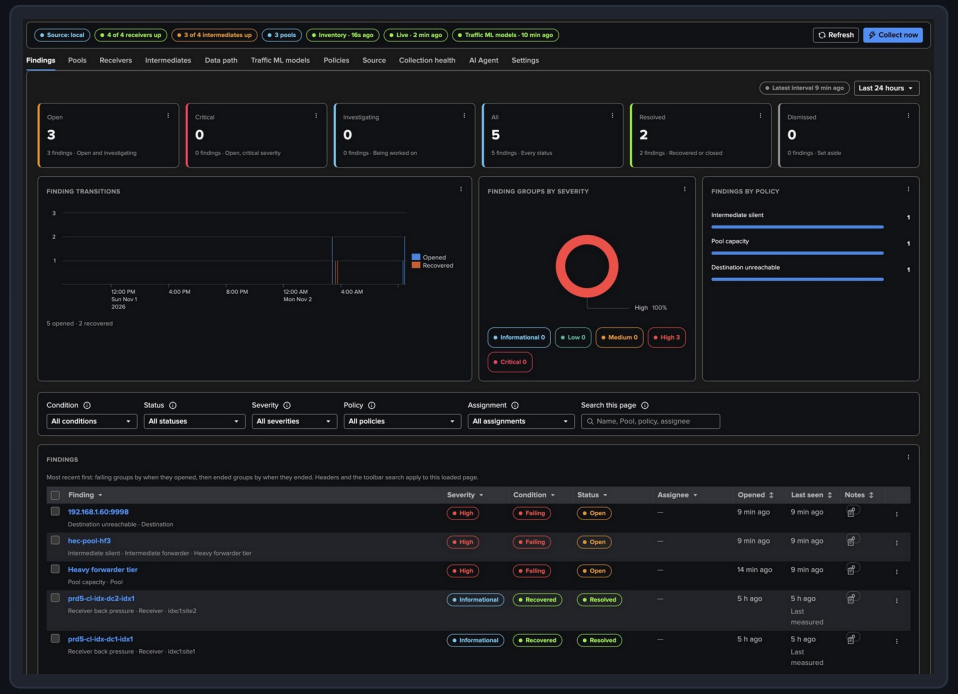
03

PART 3 · FINDINGS

A heavy forwarder stops

One member of the heavy forwarder tier stopped on purpose: what opens, how each finding is evaluated, and the Splunk S2S AI Agent.

What opens when a heavy forwarder stops



1 **Intermediate silent**

No connection line while the other members of its pool report.

2 **Pool capacity**

Three members up against four expected.

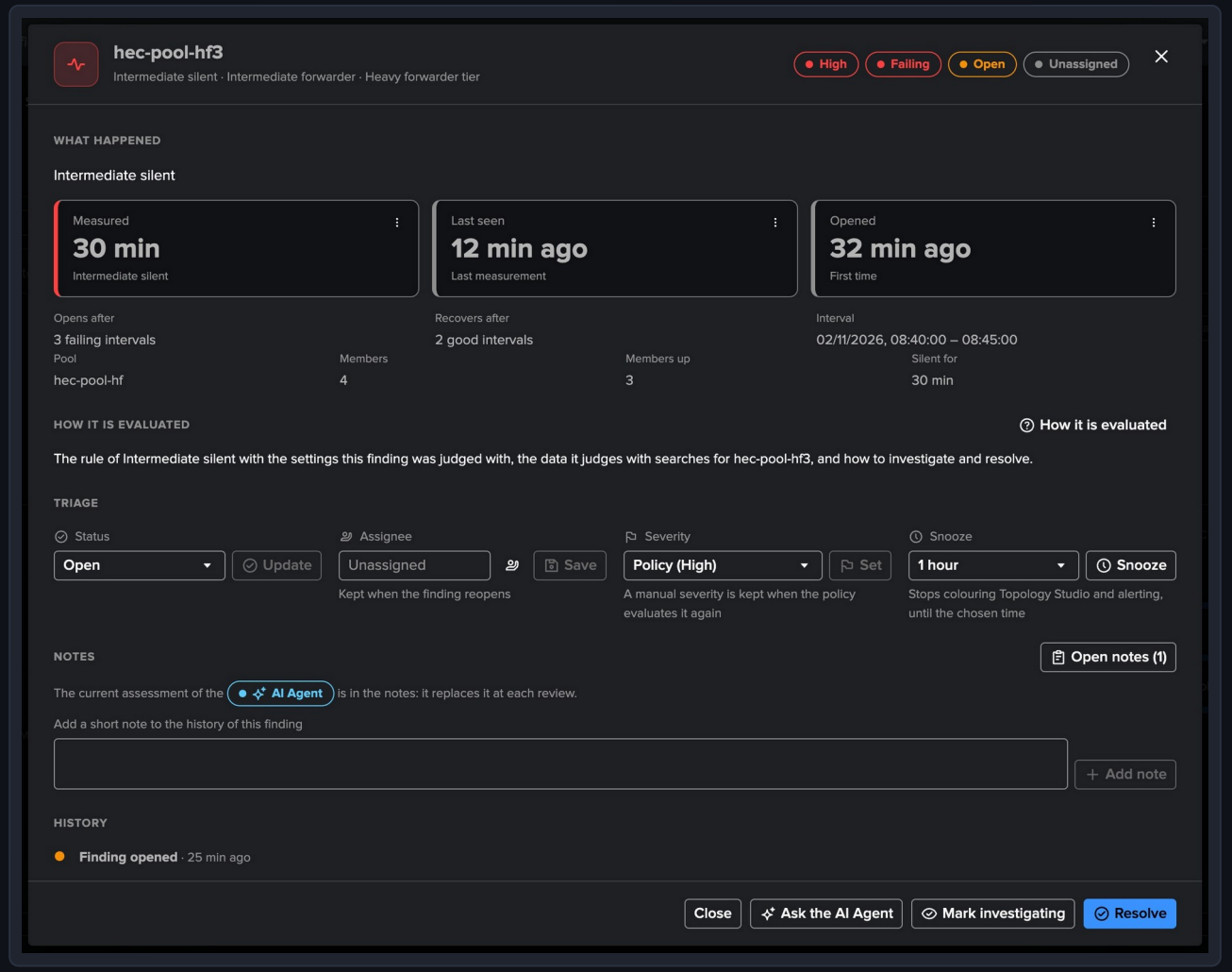
3 **Destination unreachable**

Twenty senders fail to reach its port: one finding, never twenty.

4 **Triage**

Investigate, snooze, dismiss, resolve; assign; merge into one group.

A finding up close — what happened, triage, history



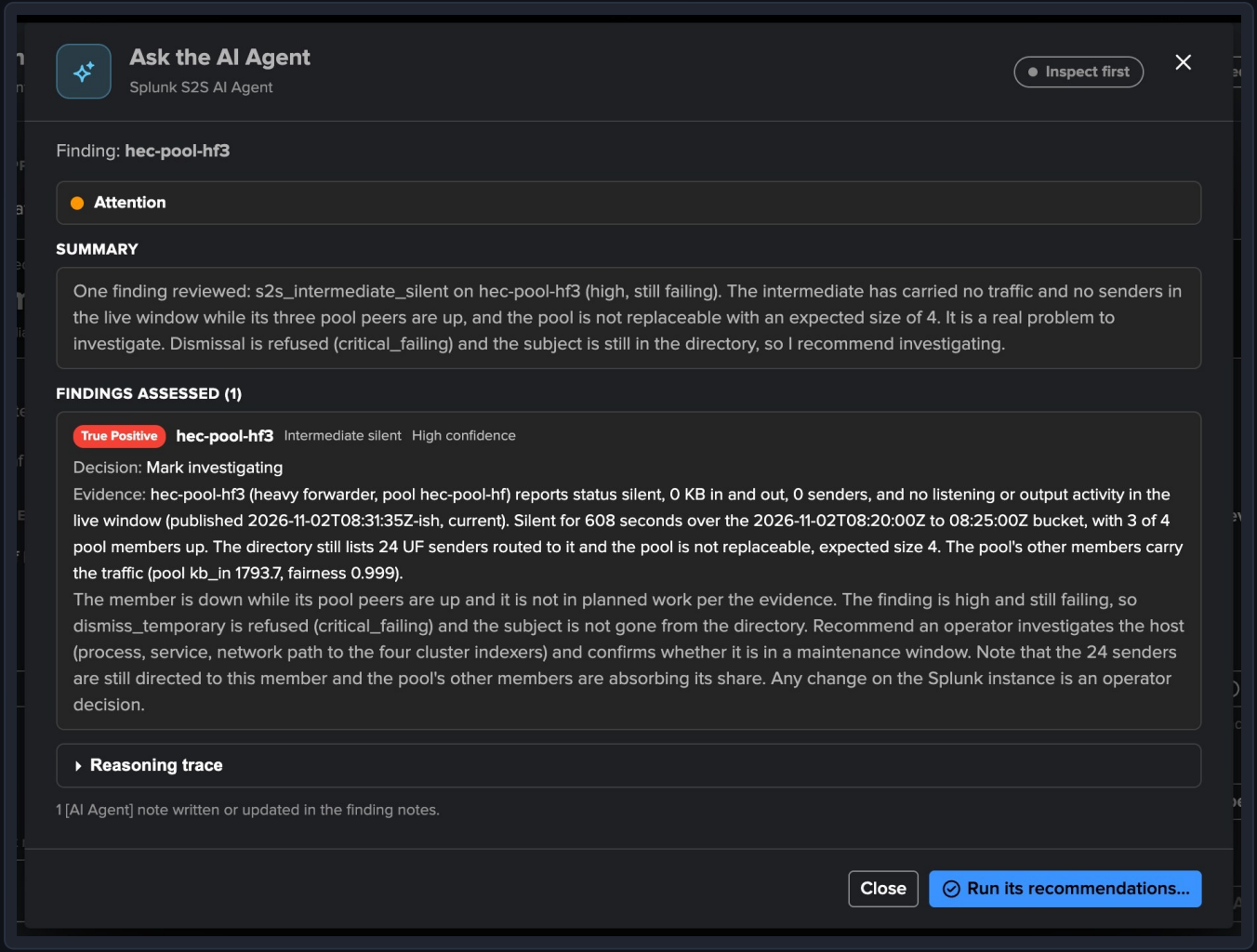
1. **What happened**
   The measurement (silent for 15 minutes), when it opened, its live window.

2. **Triage**
   Status, assignee, severity, snooze and notes, kept when it reopens.

3. **Open resource**
   From the row menu: the host or the pool modal, in place.

4. **How it is evaluated**
   Every policy's rule, data and searches, in the Policies tab.

The Splunk S2S AI Agent — a review on demand or on a schedule



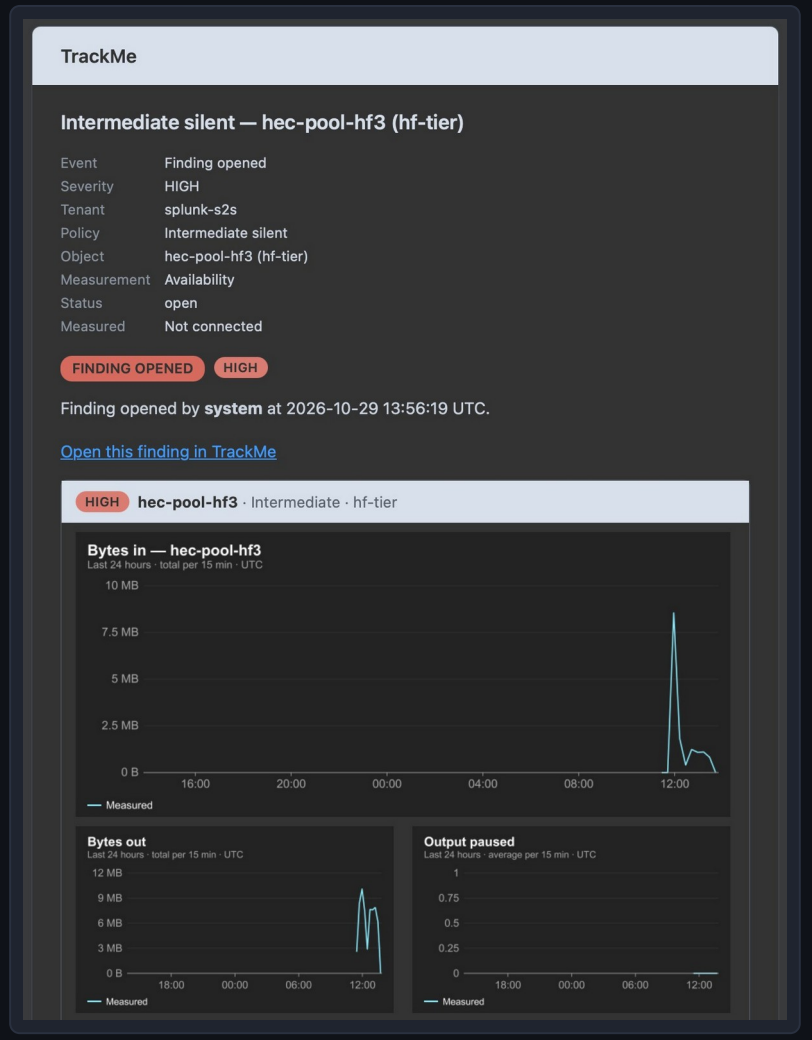
1. **What it reads**
   The host's live row, its pool, the cluster, collection health, past decisions.

2. **Its tools**
   Hosts, pools, senders, data path, traffic models, evidence, trends.

3. **One note per review**
   An [AI Agent] note on the finding.

4. **Act mode**
   Only what the tenant allows; never a change on a Splunk instance.

The email — the finding, its measurement and its charts



1. **Per Finding group or digest**
   One threaded email per transition, or one digest per live run.

2. **What it says**
   Policy, object, measurement, status, and a link to the finding.

3. **Charts embedded**
   The object's last 24 hours: bytes in, bytes out, output paused.

4. **Always recorded**
   Every transition is an event in the notable index.

One stop, one email per object

The same stop emailed the Destination unreachable finding of its receiving port: twenty senders, one email.

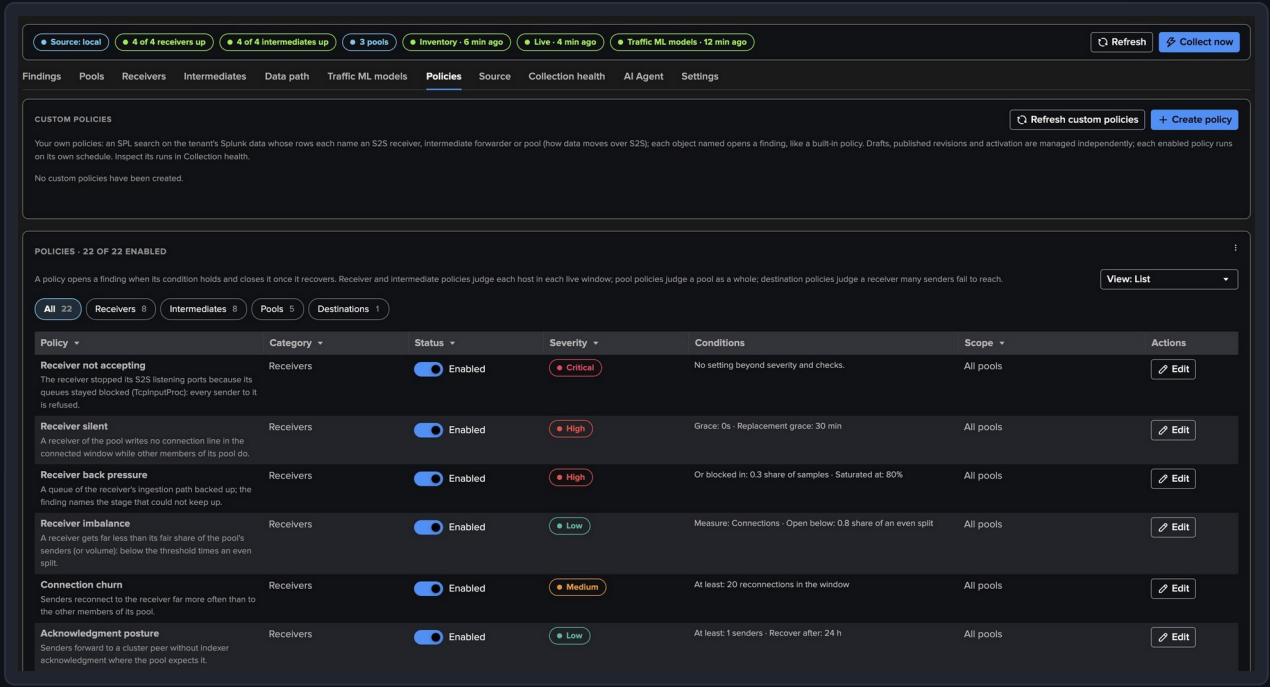
04

PART 4 · OPERATE

Policies, models, collection

Twenty-two policies and how each is evaluated, the traffic ML models of every receiver pool, and collection health.

Twenty-two policies — receivers, intermediates, pools, destinations



1 Receivers

Not accepting, silent, back pressure, imbalance, churn, TLS, acknowledgement, versions.

2 Intermediates

Output blocked, back pressure, silent, destinations, single destination, in / out.

3 Pools

Not receiving, capacity, traffic drop, traffic outlier, back pressure cascade.

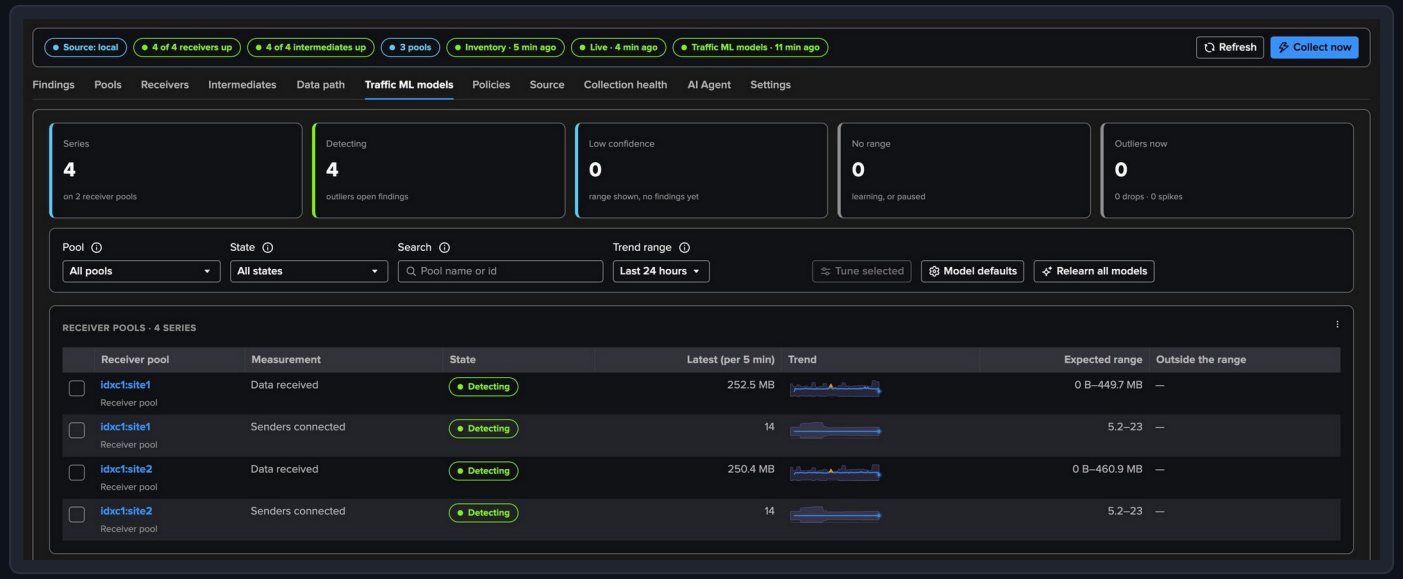
4 Destinations

A receiver many senders fail to reach: one finding, never one per sender.

5 Custom policies

Your own SPL: each row names a receiver, an intermediate or a pool.

Traffic ML models — what each pool usually receives



1 **Two models per pool**

KB received and connected senders, per weekday and hour: four series on the two site pools.

2 **Learned**

From the members' lines over the learning history, 14 days by default.

3 **Pool traffic outlier**

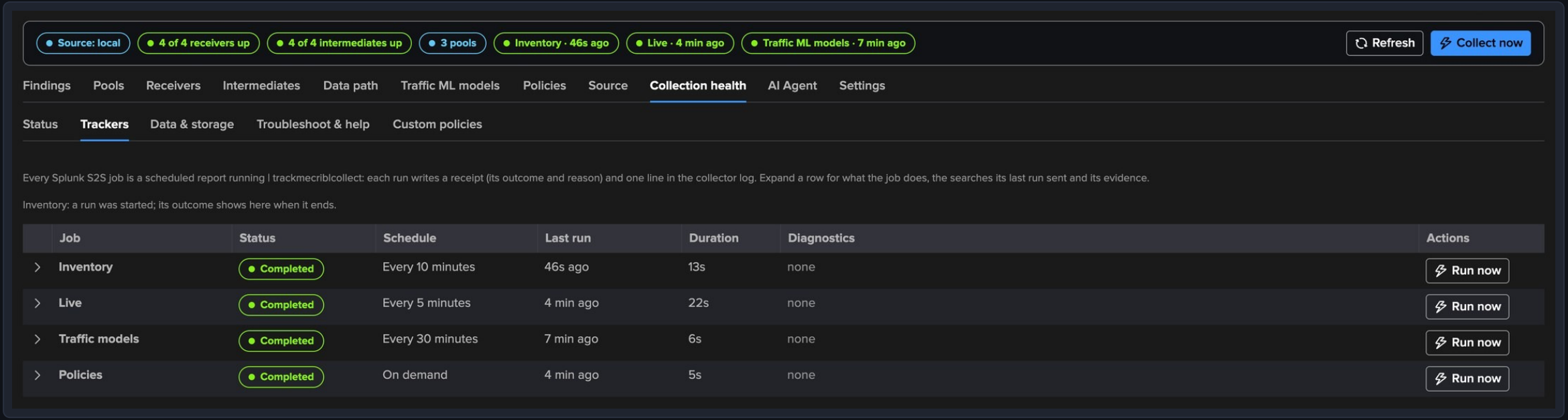
Outside the model's range for this hour, from normal confidence.

4 **Tuning**

Tune model with a simulation, Not an anomaly, Relearn all models.

PART 4 · COLLECTION

Collection health — every job, and why when it does not run



1 **Status**
Every problem with its meaning, source, causes and steps.

2 **Trackers**
Inventory, Live, Traffic models, Policies: last run, searches, Run now; Custom policies apart.

3 **Cluster reads**
cluster_read_refused names the missing list_indexer_cluster.

4 **Configuration Guardian**
The same assessment raises cribl_collection_health.

WHERE WE ARE

One tenant, every receiver, intermediate and pool

Set up

a tenant on the local search tier, an indexer cluster as one pool per site, the heavy forwarder tier

Followed

pools, a pool's distribution, a receiver's pipeline, the data path

Judged

a heavy forwarder silent and its pool short of a member, explained

Operated

twenty-two policies, traffic ML models, collection health

What made it work

Splunk's own telemetry

No agent, no new data, no call to a forwarder: the lines Splunk already writes.

Pools judged as a whole

Members up against the expected size, the share each one receives, planned work held.

Back pressure explained

The bottleneck and its origin, one cascade finding instead of a storm.