HANDS-ON WALKTHROUGH · SPLUNK UF FLEET

Which forwarding agents are connected — at any fleet size?

One tenant follows the forwarding agents of a Splunk deployment and the Deployment Servers that manage them, from what Splunk already writes: the receivers' own lines, the agents' logs, the Deployment Server indexes.

Captured live on TrackMe 2.4.22 · twenty universal forwarders on four sites and five UF generations, a Deployment Server cluster

THE PROBLEM

A forwarder that stops sending says nothing — until the data is missed

Thousands of agents, a few Deployment Servers, and no single place that says which agent stopped, why, and whether it is one agent or a whole site.

Agents that go quiet

A disconnected agent, an expected one missing, a whole group gone at once.

Agents in trouble

Output blocked, destinations failing, throttled, restarting in a loop, app deployments failing.

Deployment Servers

Silent, phonehomes below their usual, deployment failures, restart storms, configuration errors.

What a Splunk UF Fleet tenant does

A dedicated tenant type for the fleet of one Splunk deployment. An agent is connected when a receiver saw its connection in the connected window: the data path itself, read from the receivers' own lines. TrackMe never polls an agent, never calls its REST API, never installs anything on it; twenty-eight built-in policies judge the agents, the fleet and its groups, the Deployment Servers and the destinations.

THE MODEL

The receivers describe the fleet — the agents are never polled

1 Source

Receivers' connection lines, agents' own logs, Deployment Server indexes (Splunk 9.2+).

2 Directory

GUID first, host name as display; enrichment from your CMDB; groups.

3 Measure

Every 5 minutes: connected set, agents' troubles, Deployment Server pace and failures.

4 Decide

28 built-in policies on agents, groups, the fleet, Deployment Servers and destinations.

5 Act

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

Agent states

Connected, disconnected, dormant after 7 days (out of every count), removal candidate after 30 days: an agent's state comes from its last connection line, never from a heartbeat you must deploy.

Built for any fleet size: aggregations over indexed terms and metric dimensions, never a search per agent; no per-agent history written.

THE PLAN

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

PART 1

Setup

- The Splunk UF Fleet card
- Deployment Server API accounts
- Scope and discovery

PART 2

Configure

- Enrichment from a CMDB
- Groups and expected agents
- Notifications

PART 3

The fleet

- Agents, one agent up close
- Deployment Servers
- The policies

PART 4

Findings

- A site goes dark
- An expected agent missing
- The AI Agent

Every screen is the product as shipped, on twenty universal forwarders named site-service-number.

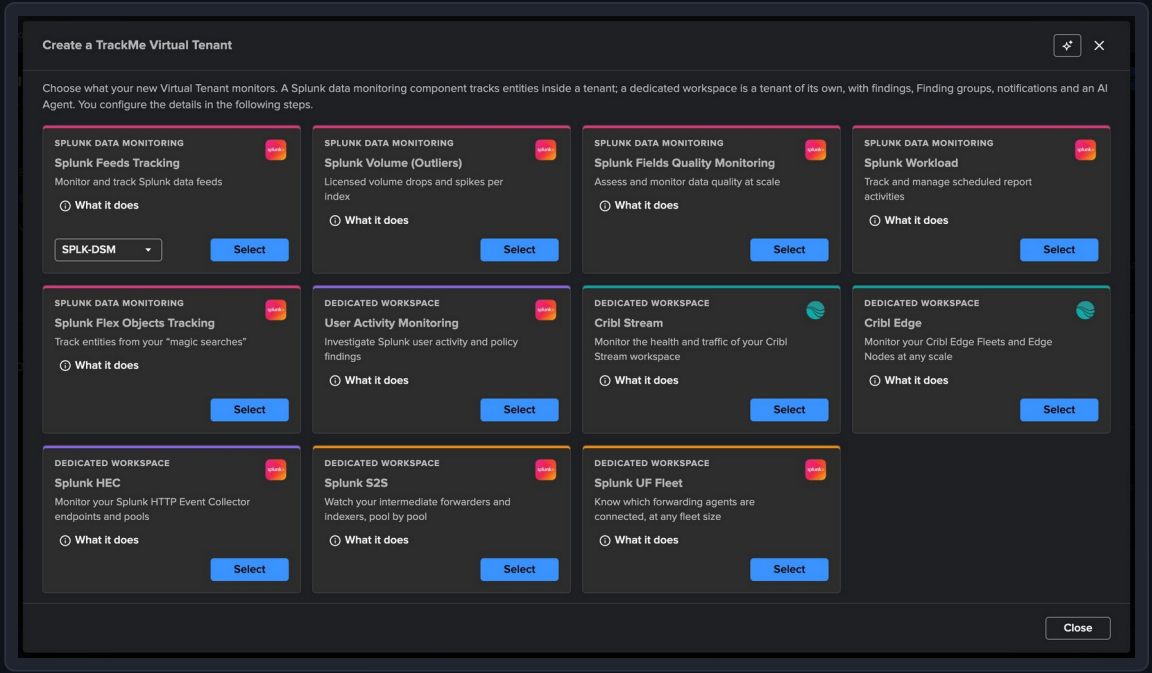
01

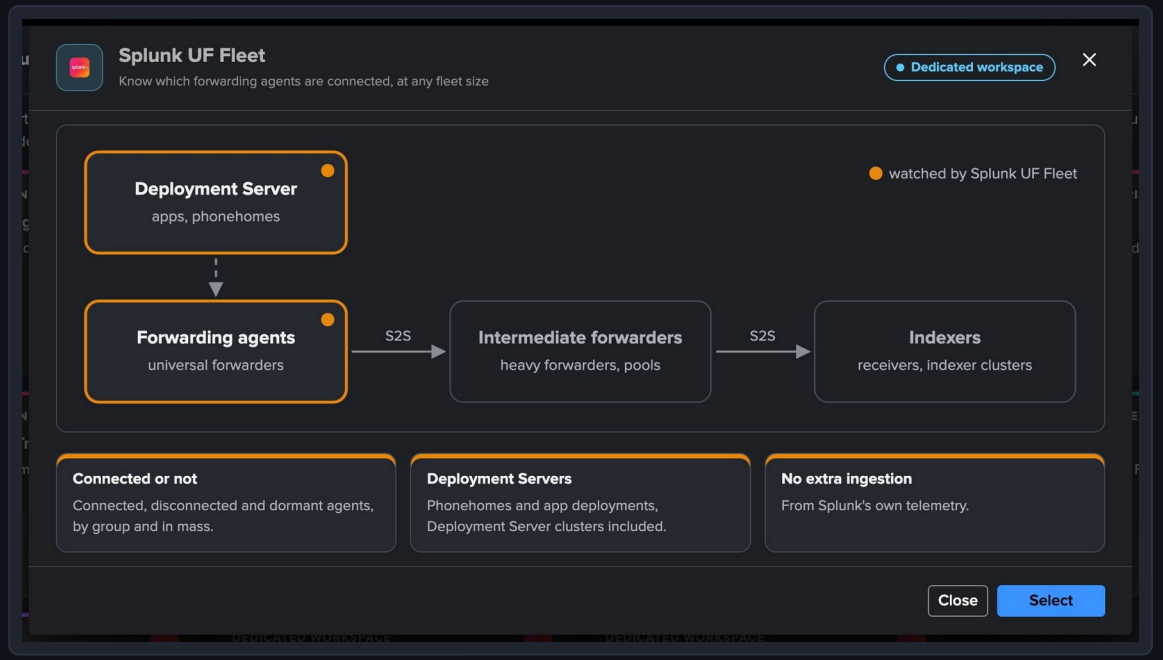
PART 1 · SETUP

One source, two Deployment Servers, one scope

The Splunk UF Fleet card, the Deployment Server API accounts checked as you add them, the scope of the fleet and the agent discovery.

A dedicated tenant — pick Splunk UF Fleet





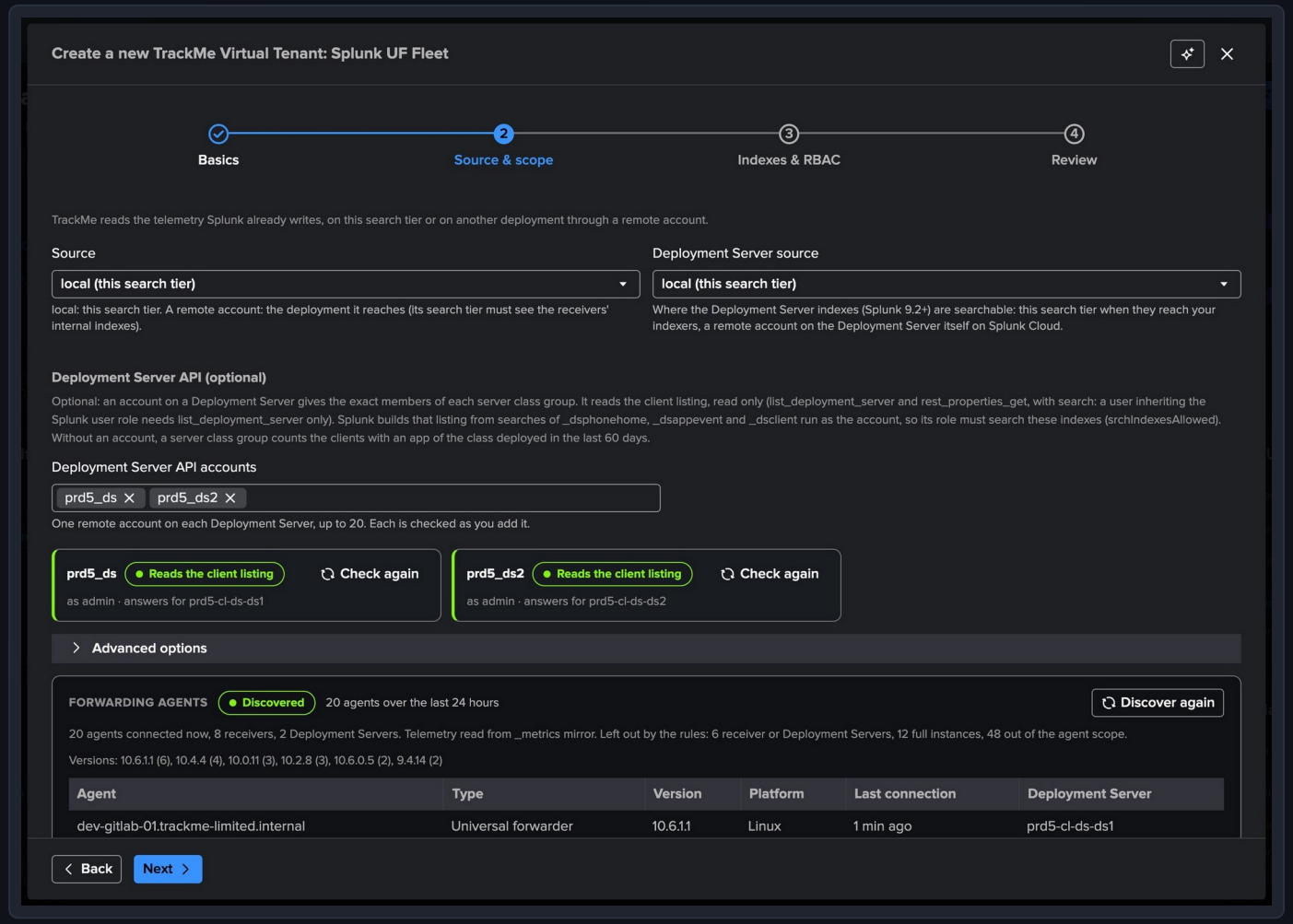
1 **The Splunk UF Fleet card**
Know which forwarding agents are connected, at any fleet size.

2 **What it does**
The Deployment Server and the forwarding agents, watched by Splunk UF Fleet.

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

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

Deployment Server API accounts — checked as you add them



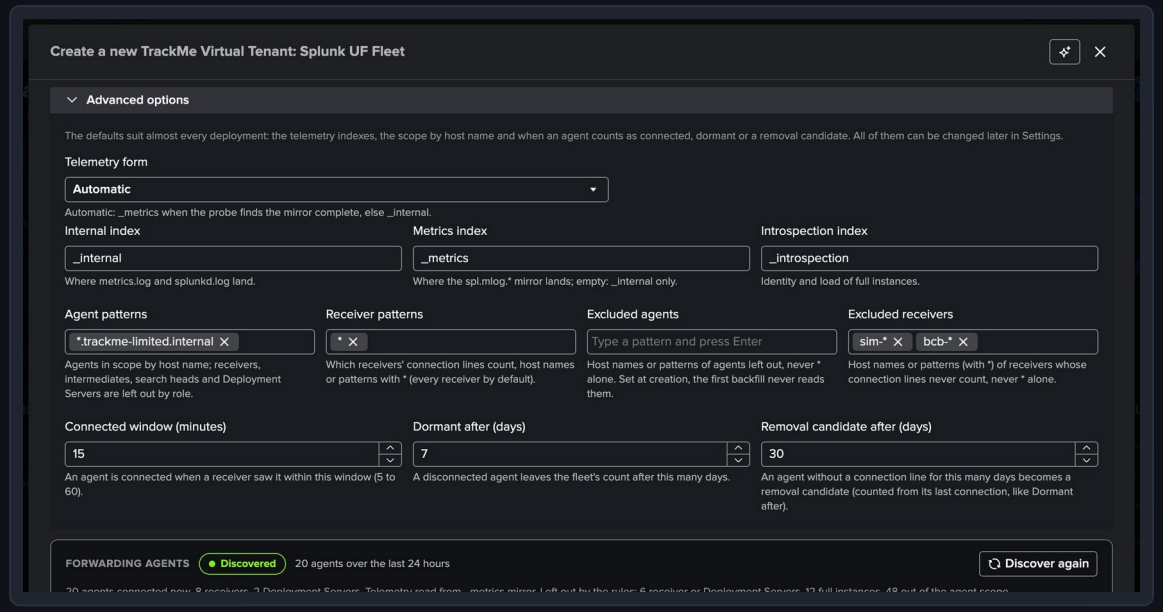
1. **Source**
   Where the receivers' internal logs are searched.

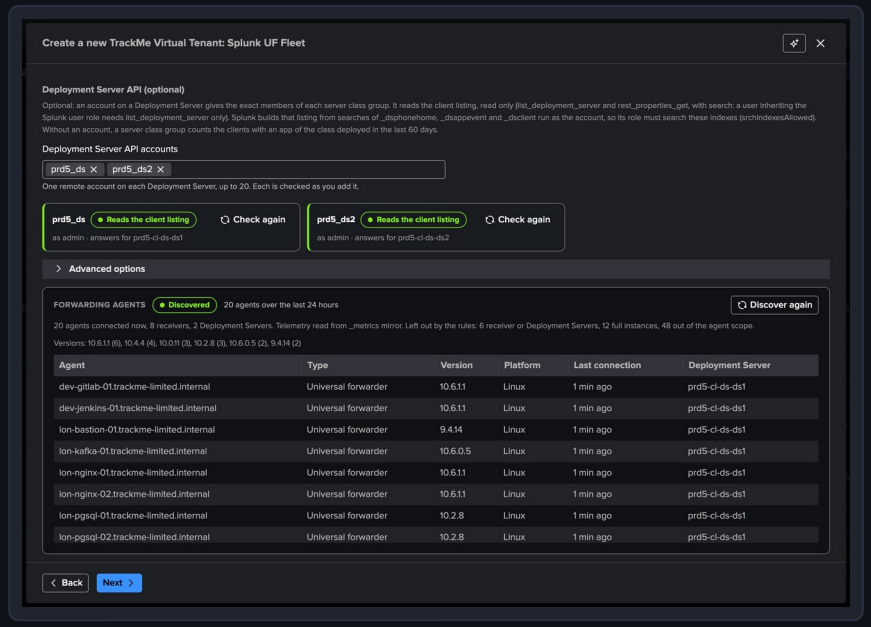
2. **Deployment Server source**
   Where the _ds* indexes are searched; a remote account on the DS on Splunk Cloud.

3. **API accounts (optional)**
   The exact members of each server class group, read only.

4. **Checked at once**
   prd5_ds and prd5_ds2 each read the client listing and answer for their own Deployment Server.

Scope the fleet, then see what the tenant would collect





1 Agent patterns

A token per pattern: the corporate domain, *.trackme-limited.internal.

2 Excluded receivers

sim-* and bcb-*: test receivers sharing the indexes, never read, never agents.

3 The discovery

20 agents, all connected, 8 receivers, 2 Deployment Servers.

4 Left out by the rules

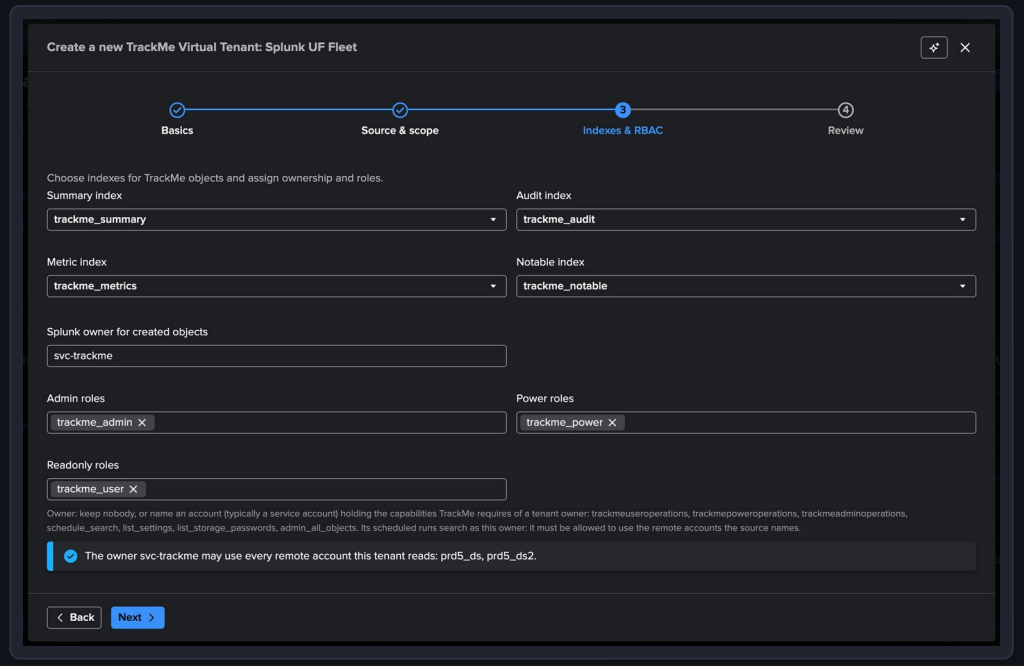
Receivers and Deployment Servers, full instances, agents out of scope.

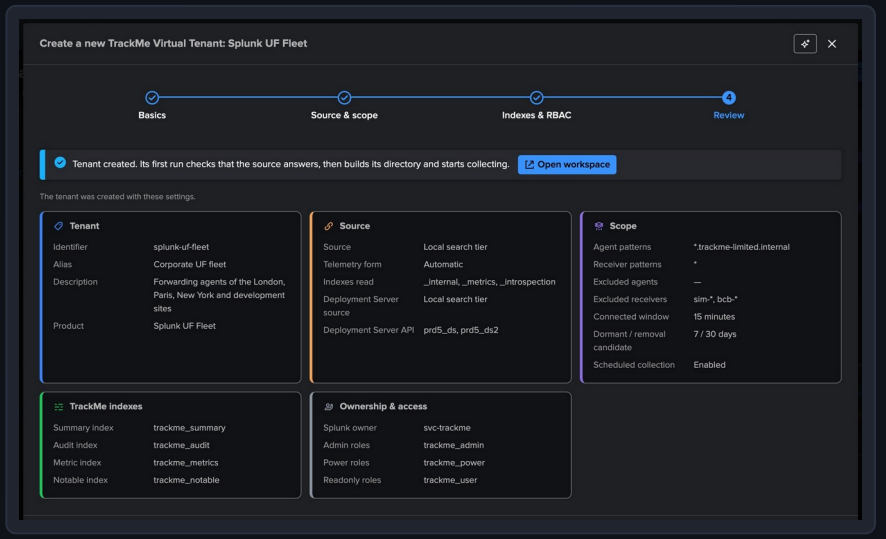
5 Five UF generations

9.4.14 to 10.6.1.1, every agent with its Deployment Server.

PART 1 · SETUP

A least-privilege owner — checked before the tenant exists





1 Tenant owner

svc-trackme, a least-privilege service account: the scheduled runs search as it.

2 Remote accounts

The wizard checks the owner may use every account the tenant reads.

3 Done

The first live run checks the source, then builds the directory.

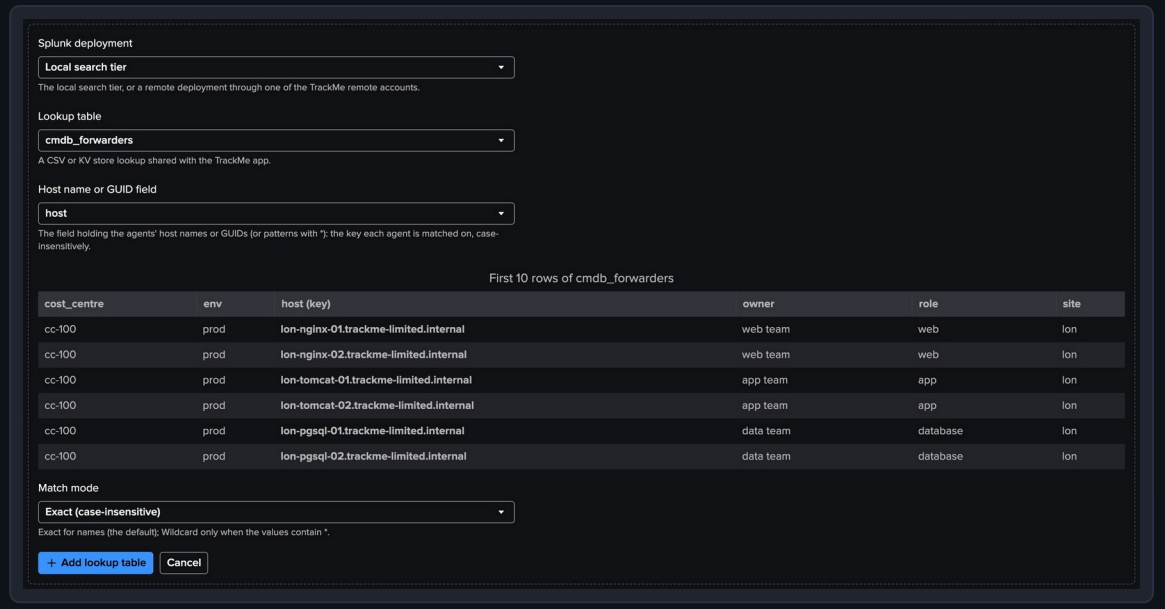
02

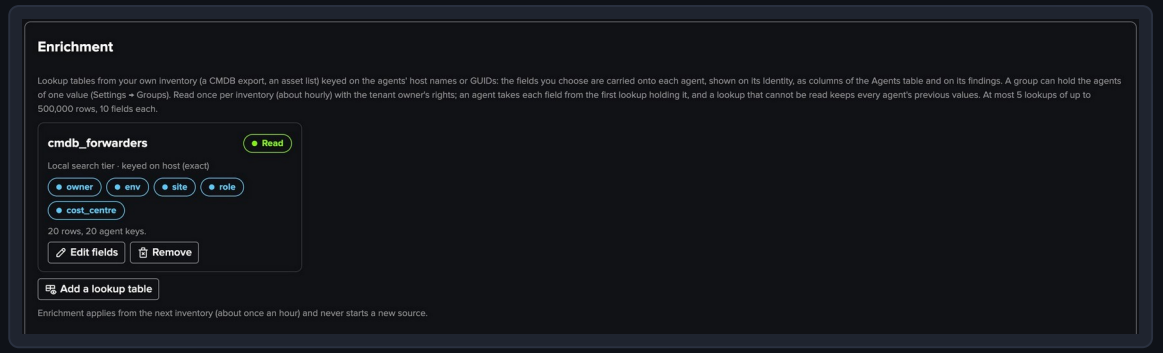
PART 2 · CONFIGURE

From a list of agents to a fleet you can read

Enrichment from your CMDB, groups proposed from the Deployment Servers and the enrichment, the expected agents, and the notifications.

Enrichment — your CMDB on every agent





1 **A lookup from your inventory**

cmdb_forwarders, keyed on the host name, exact or wildcard.

2 **Fields to carry**

owner, env, site, role, cost_centre: the first three are Agents columns.

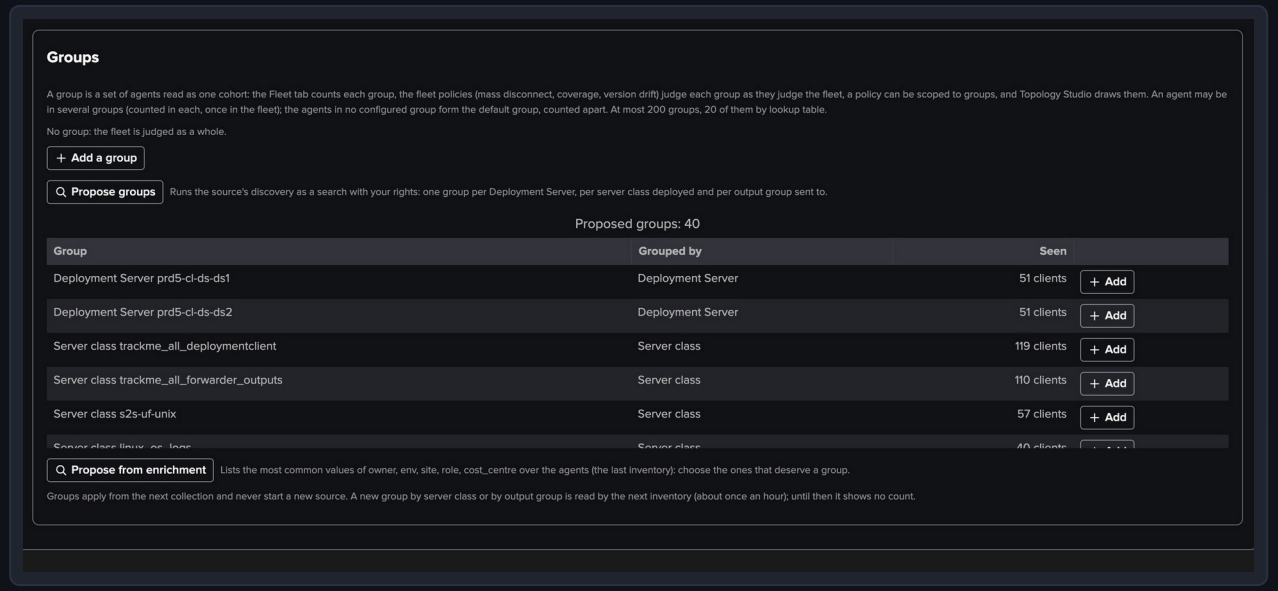
3 **On the findings**

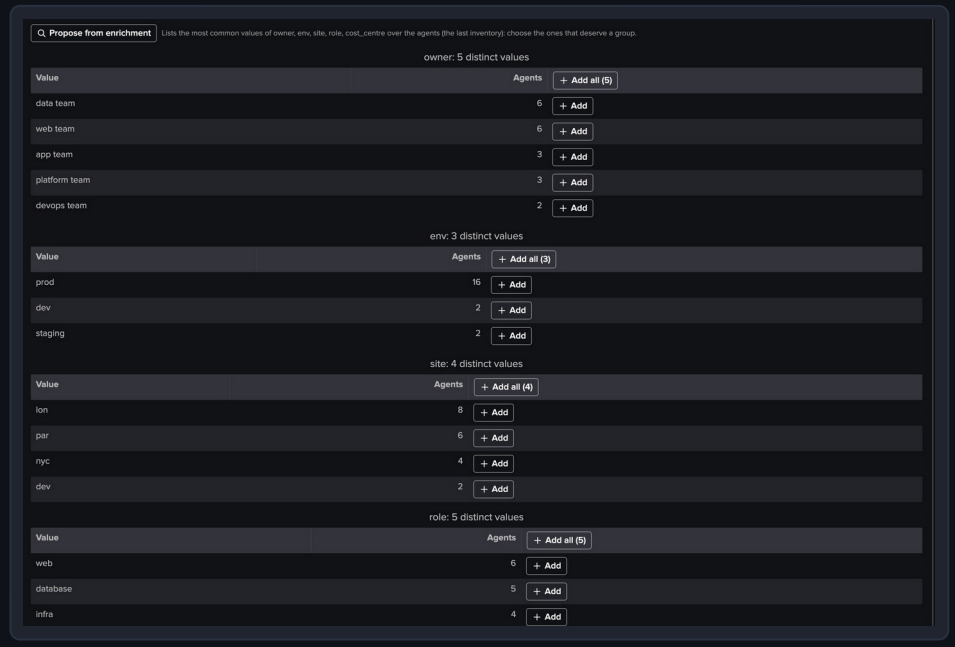
Notifications, assignment and the AI Agent read them.

4 **Read hourly**

As the tenant owner: Read, 20 rows and 20 agent keys; an unreadable lookup keeps the previous values.

Groups — proposed from your Deployment Servers and your CMDB





1 Propose groups

One per Deployment Server, server class deployed, output group sent to.

2 Sites and roles

The server classes site_* and role_* added, then renamed.

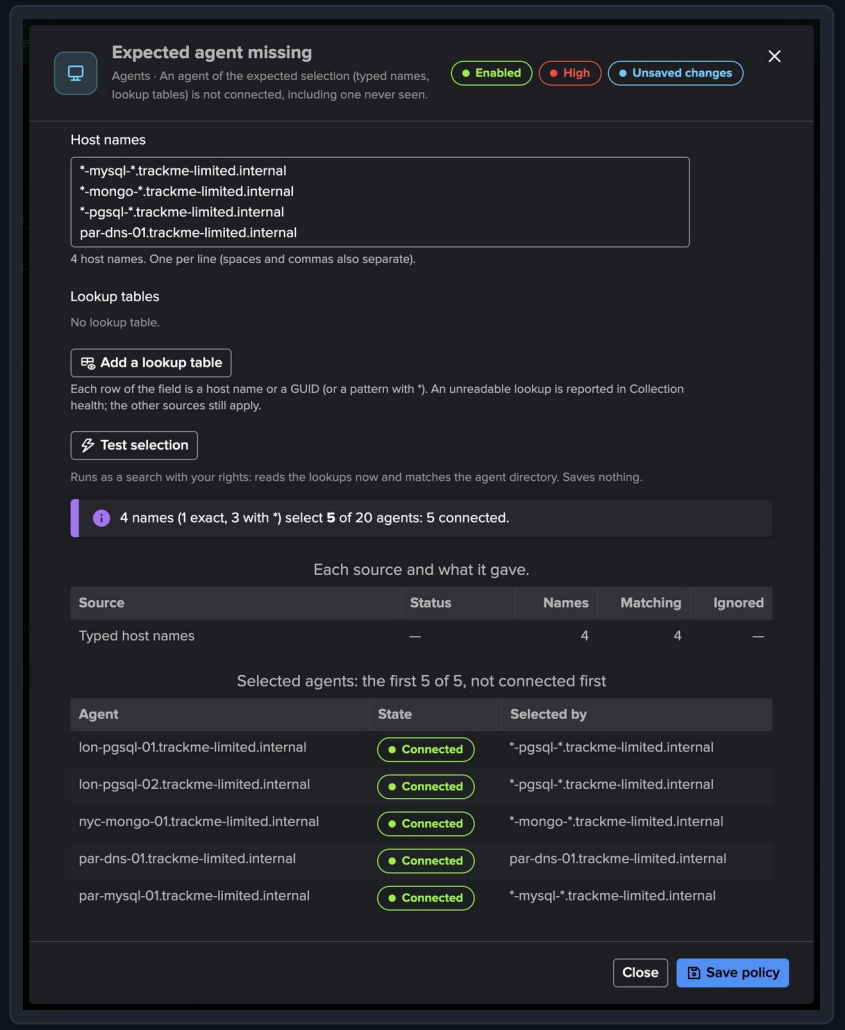
3 Propose from enrichment

The most common values of each field: env prod, staging, dev.

4 Judged as cohorts

Mass disconnect, coverage and version drift judge every group.

The agents that must always stay connected



1. **Typed names and patterns**
   The database servers and the DNS: *-pgsql-*, *-mysql-*, *-mongo-*, par-dns-01.

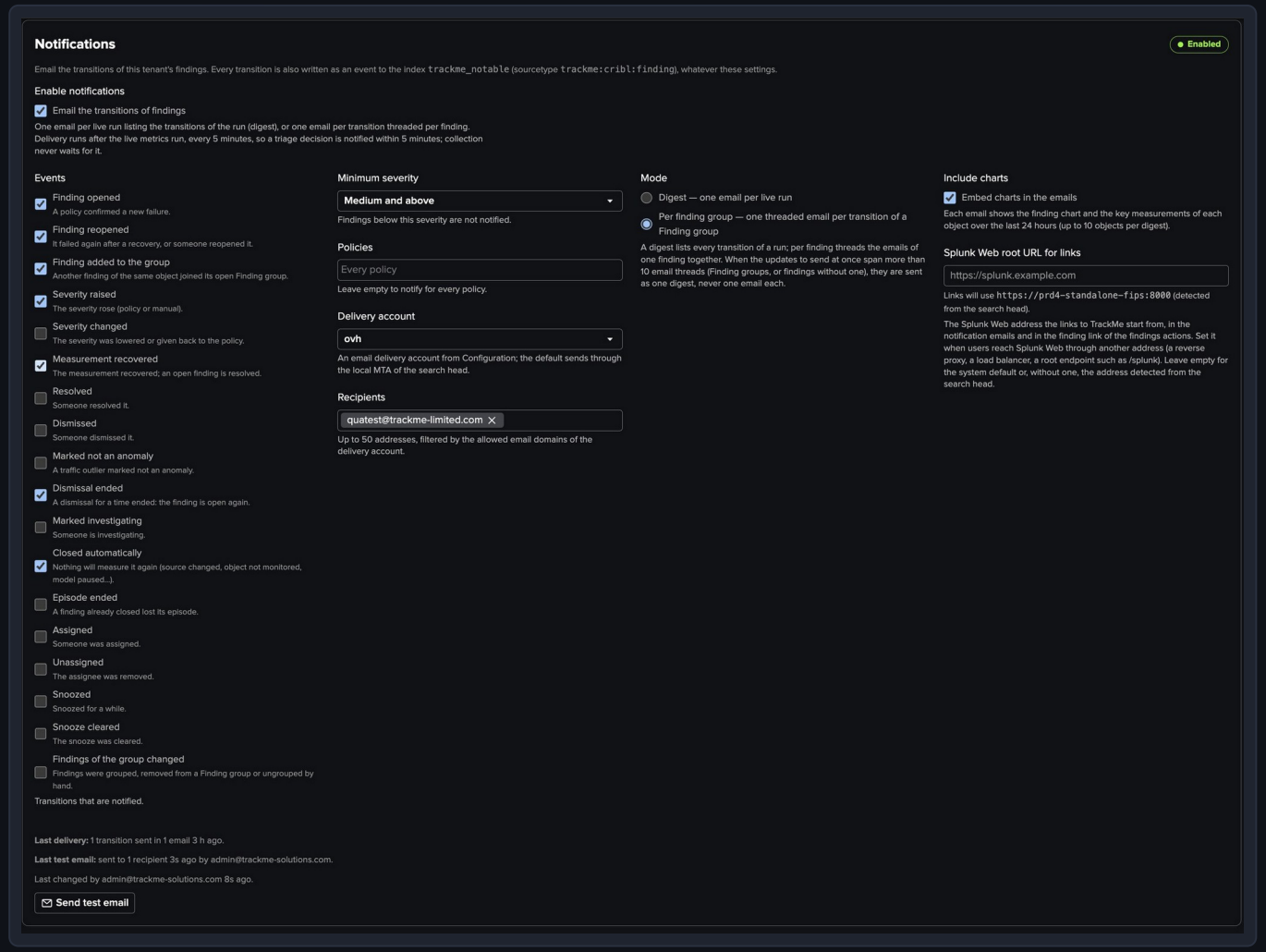
2. **Lookup tables**
   Up to 20, read again every 15 minutes as the tenant owner.

3. **Test selection**
   4 names select 5 of 20 agents, all connected; saves nothing.

4. **Never seen is missing**
   An exact name no agent carries is reported missing at once.

5. **Add to expected**
   From the Agents tab; the selected agents carry a pin.

Notifications — one threaded email per Finding group



1. **Events**
   Opened, reopened, added to the group, severity raised, recovered...

2. **Minimum severity, policies**
   Medium and above, every policy.

3. **Delivery account, recipients**
   An email delivery account and up to 50 addresses.

4. **Mode and charts**
   Per Finding group, threaded; the finding's chart embedded.

03

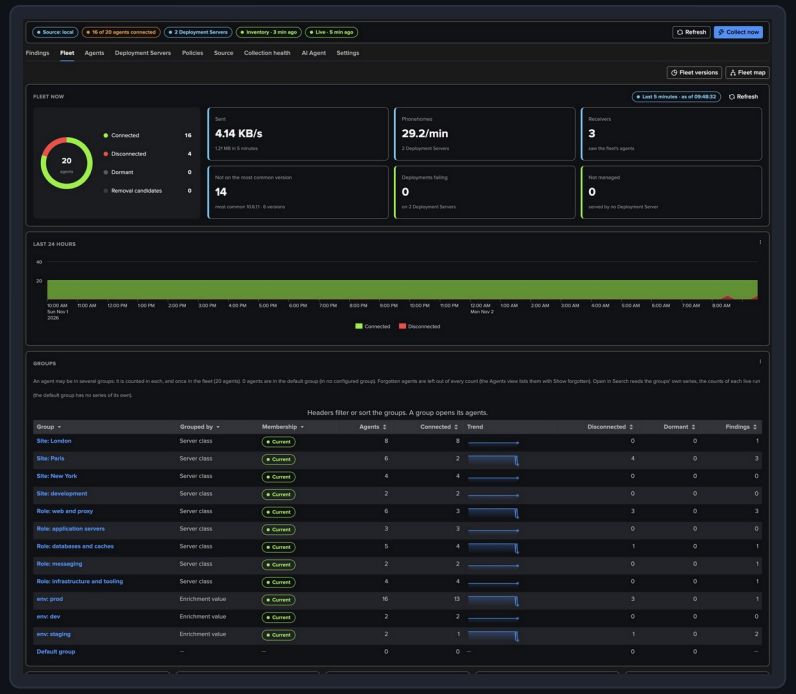
PART 3 · THE FLEET

Every agent, every Deployment Server

The fleet at a glance, the agents and one agent up close, the Deployment Servers and the policies.

PART 3 · FLEET

The fleet — now, the last 24 hours, every group



1 Fleet now

The agents by state; the last five minutes; versions, failing deployments, unmanaged agents.

2 Last 24 hours

Connected and disconnected agents over time.

3 Breakdowns

Versions, platforms, types, agents per Deployment Server.

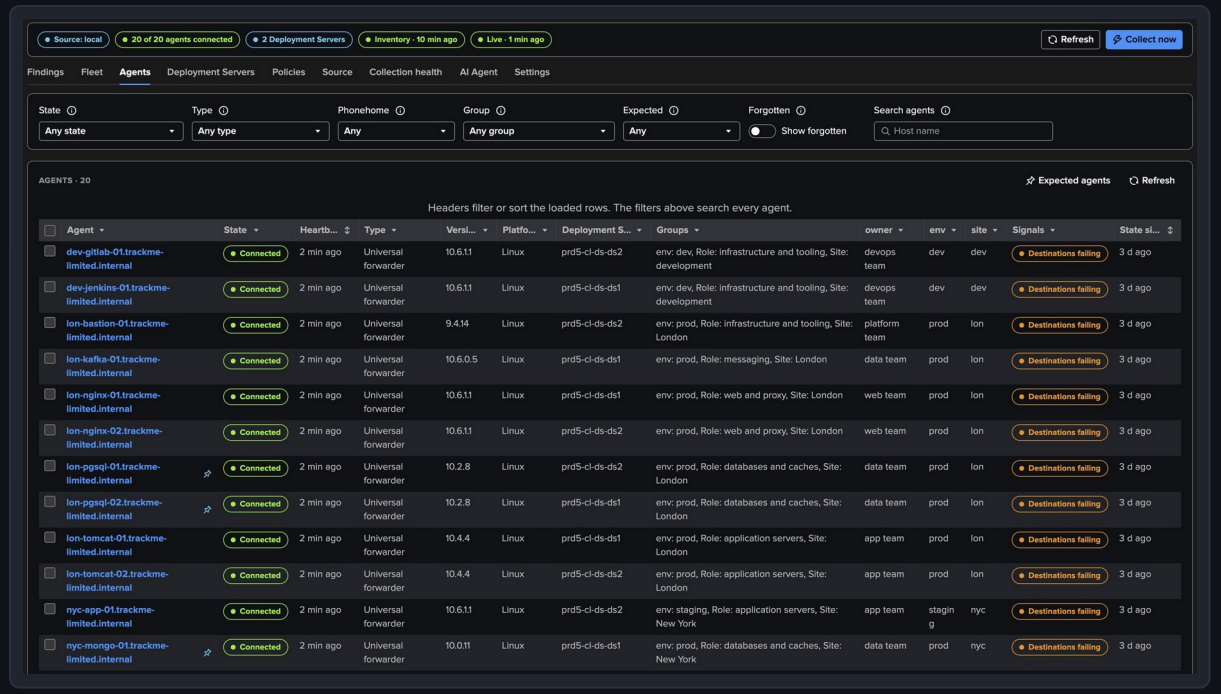
4 Groups

Each group's counts, its trend and its open cohort findings.

5 Fleet map

A generated topology of the fleet, its groups and Deployment Servers.

The agents — with their groups and your CMDB fields



1 **Filters**

State, type, phonehome, group, expected, host name.

2 **Groups and enrichment**

Site and role from the server classes; owner, env, site from the CMDB.

3 **Expected pin**

The PostgreSQL servers are expected agents.

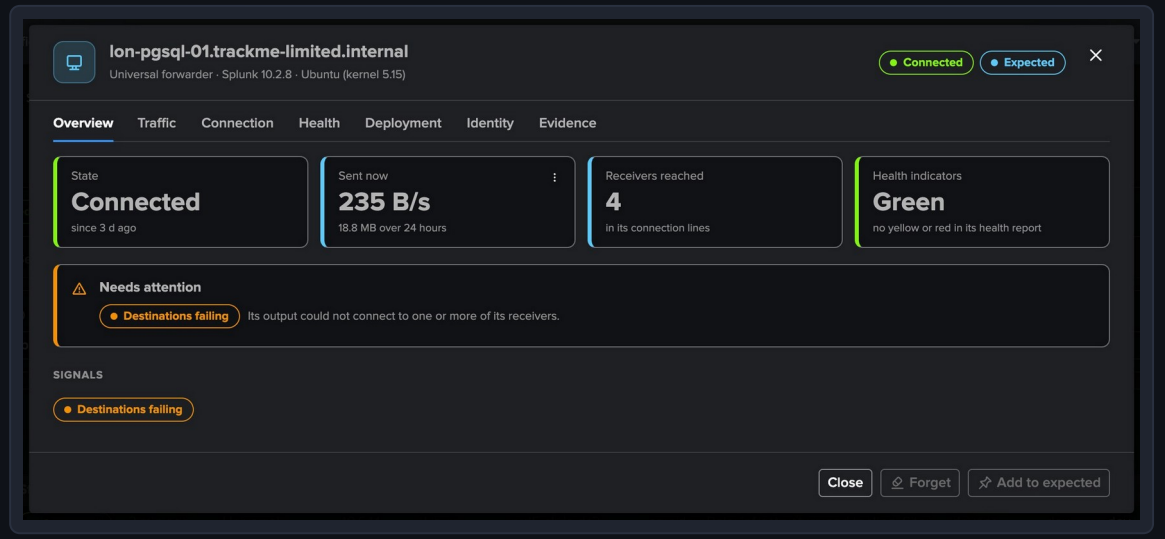
4 **Signals**

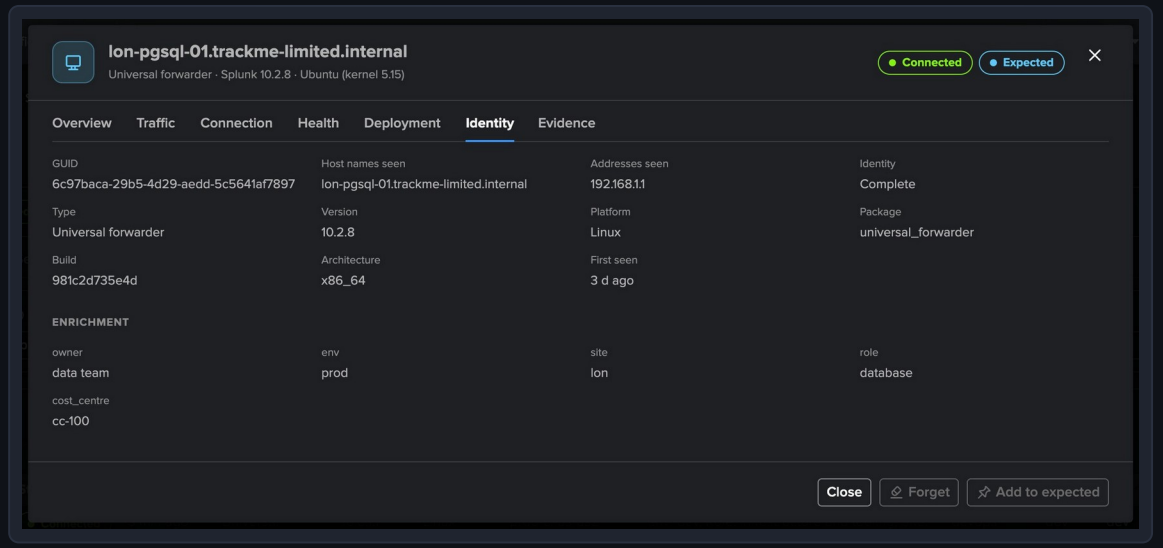
Destinations failing: a heavy forwarder of the receiving tier is stopped.

5 **Actions**

Forget an agent not connected; Add to expected; Show forgotten; rows per page.

One agent up close — state, traffic, identity





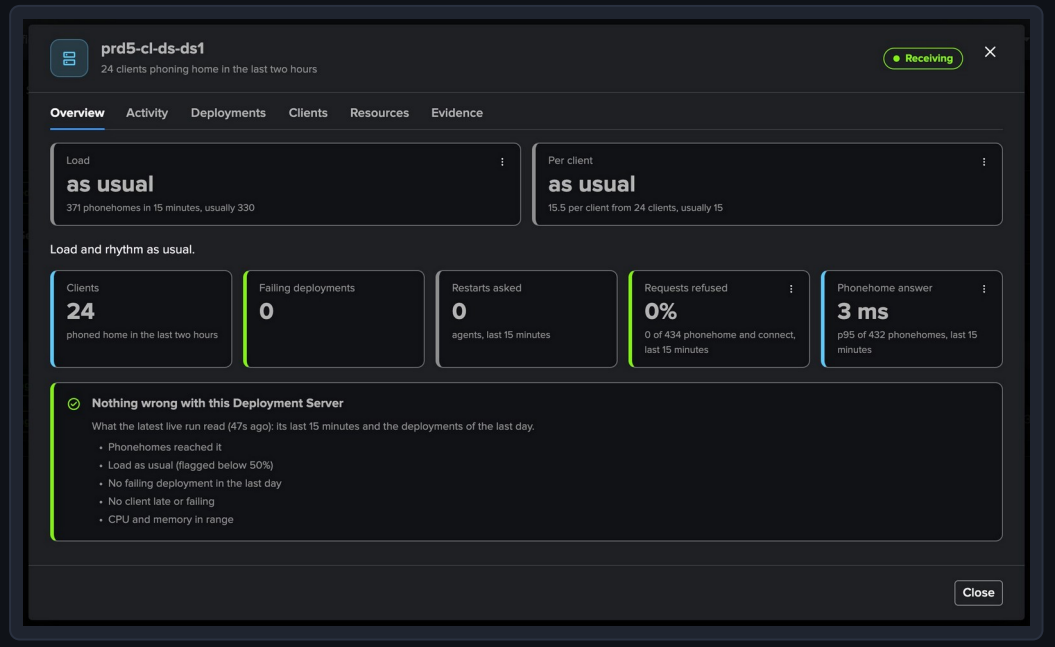
1 **Overview**
Connected, sent now and over 24 hours, receivers reached, health.

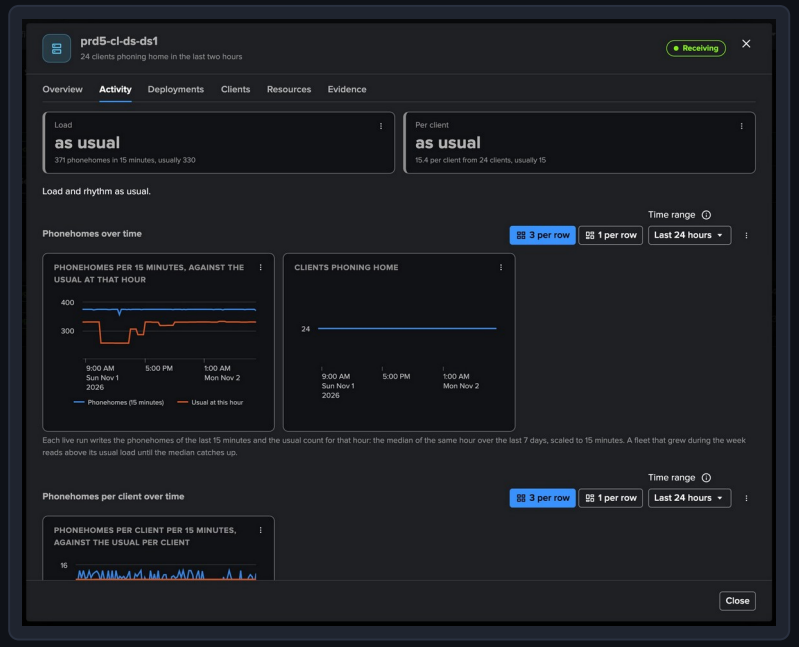
2 **Needs attention**
Its signals and open findings: here Destinations failing.

3 **Identity**
GUID, host names and addresses seen, version, build, platform.

4 **Enrichment**
owner data team, env prod, site lon, role database, cost centre.

Deployment Servers — their pace against their usual





1 Load

Phonehomes against the usual for the hour: as usual.

2 Per client

The clients' rhythm against their usual.

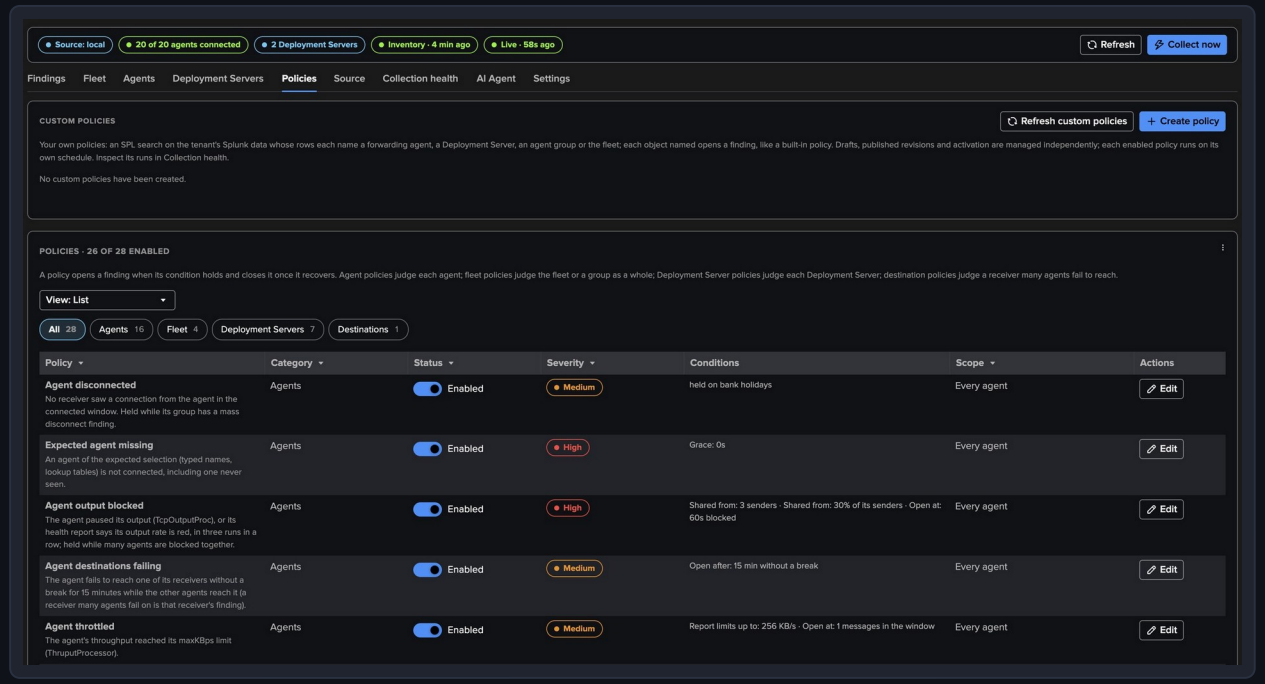
3 Deployments and restarts

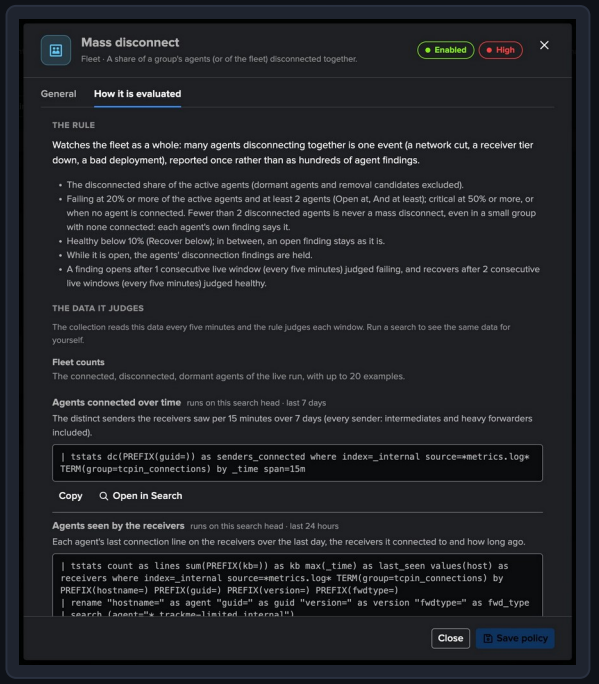
Failing deployments, restarts asked.

4 Access log

Requests refused and the 95th percentile of the phonehome answer.

Twenty-eight policies — and how each is evaluated





1 Agents

Disconnected, expected missing, output, destinations, throttle, restarts, identity, versions, deployment.

2 Fleet

Mass disconnect, coverage, version drift, cloned images.

3 Deployment Servers

Silent, phonehome drop, failures, restart storm, errors, requests, response.

4 How it is evaluated

The rule, the data with ready-to-run searches, how to resolve.

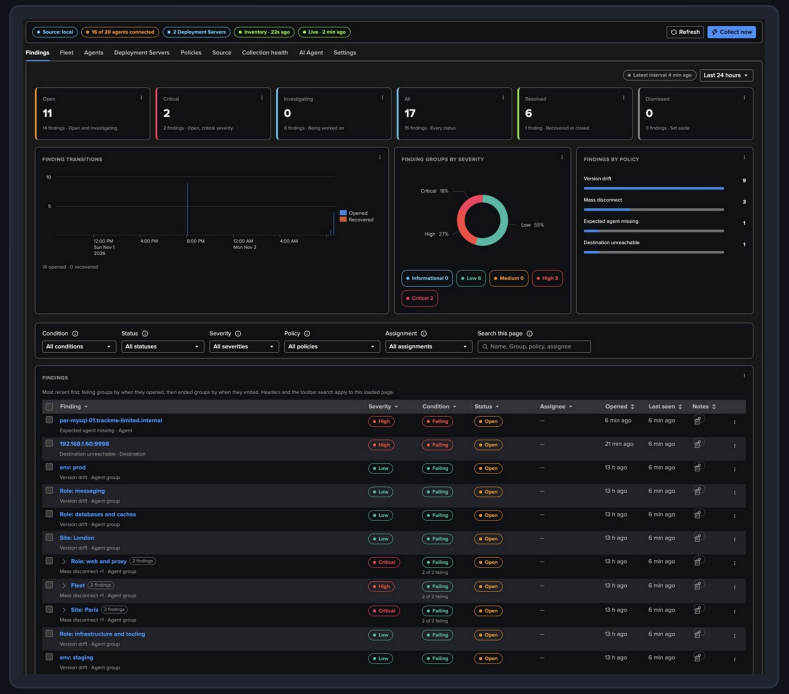
04

PART 4 · FINDINGS

A site goes dark

Agents of one site stopped on purpose: a mass disconnect per cohort, an expected agent missing, and the Splunk UF Fleet AI Agent.

A mass disconnect per cohort — not four agent findings



1 **Site: Paris**

Four of six agents disconnected: critical at 50 % and more.

2 **Role and fleet**

Web and proxy (critical) and the fleet (20 %, high): each cohort it hits.

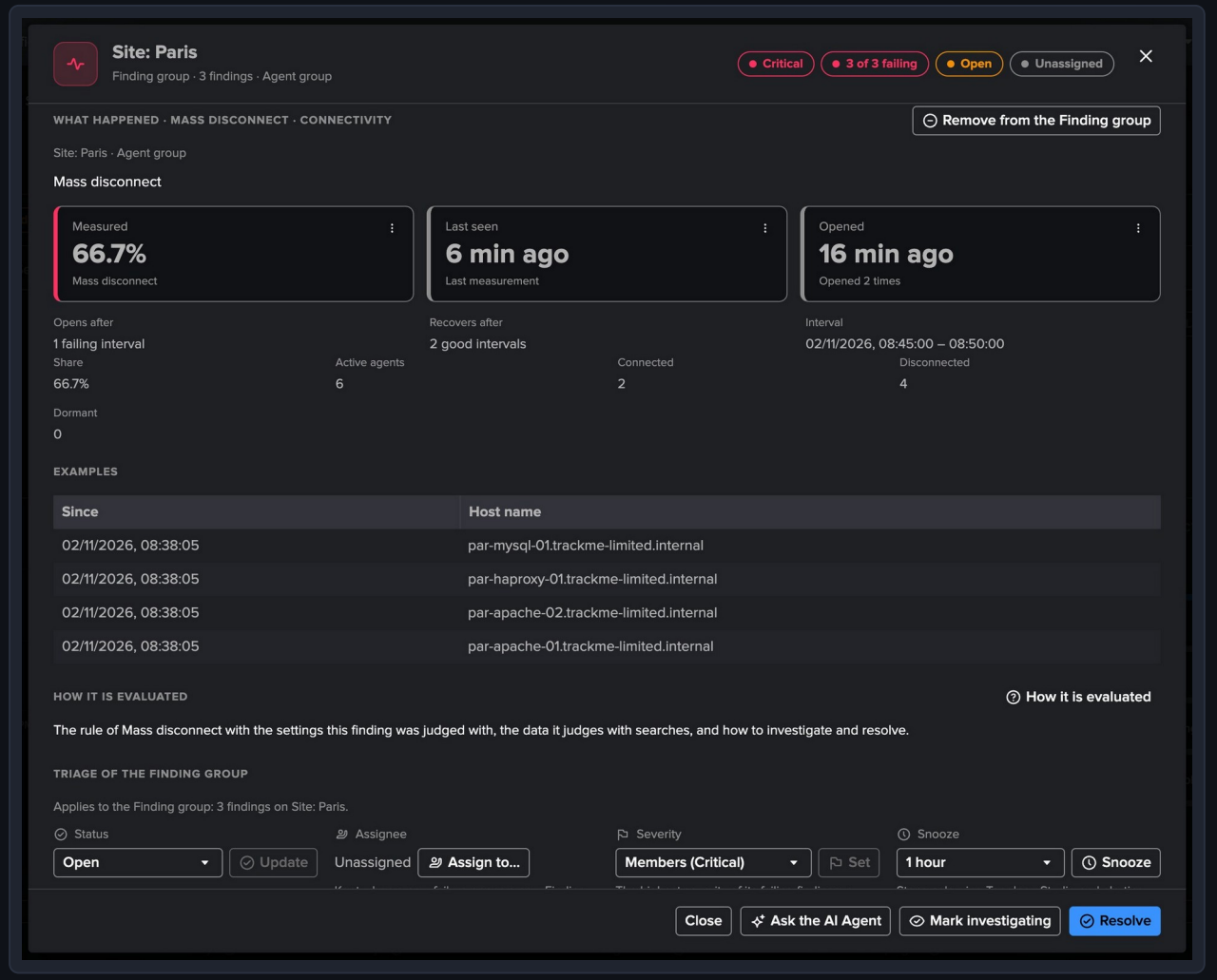
3 **One Finding group per cohort**

Each Mass disconnect joins its cohort's group, beside its Version drift.

4 **Expected agent missing**

par-mysql-01: an expected agent is never held; the agents' own findings are.

A finding up close — what happened, triage, history



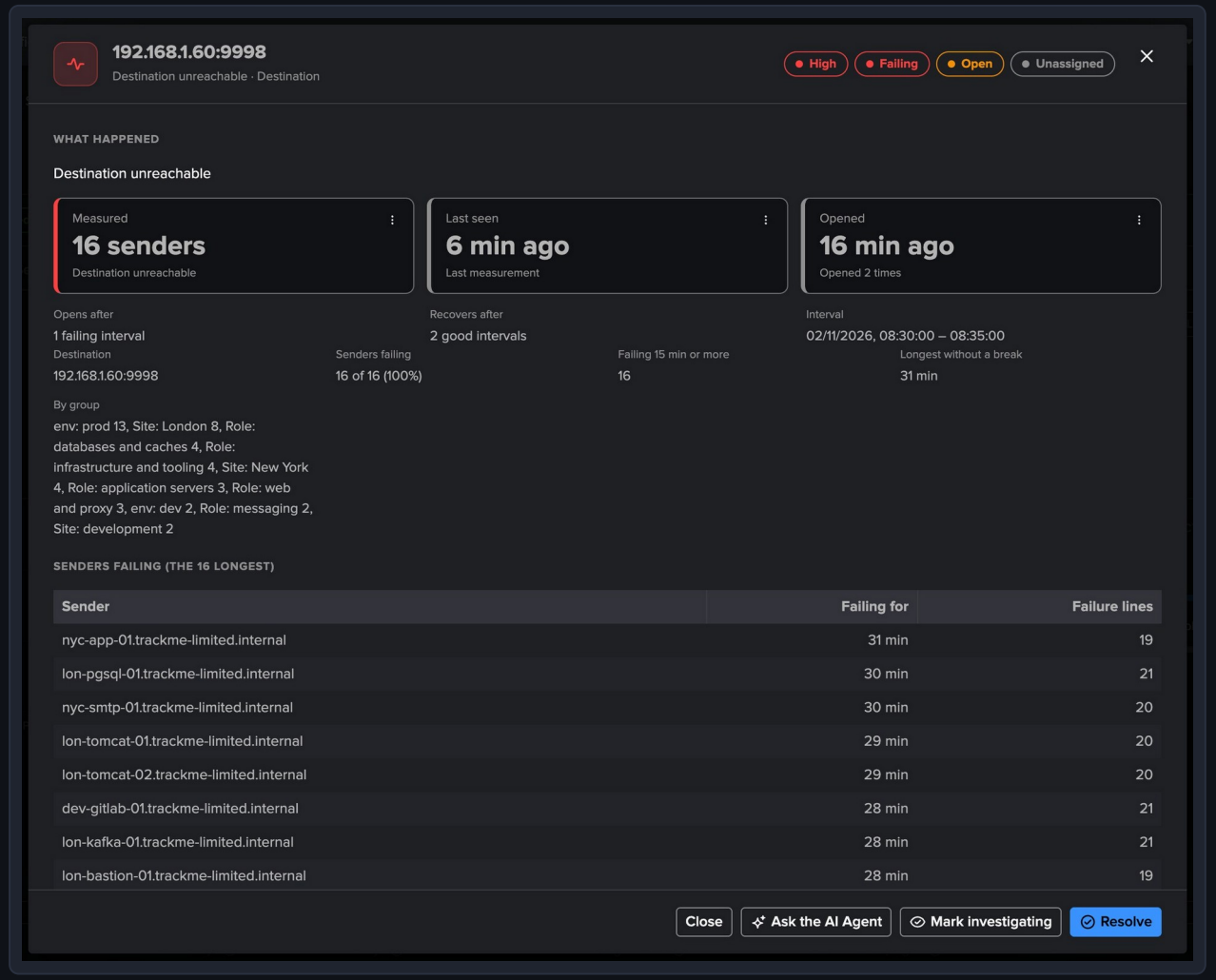
1. **What happened**
   The measurement (66.7 % of the site), when it opened, its live window.

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

3. **Open resource**
   From the row menu: the Fleet tab, an agent or its Deployment Server.

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

A destination many agents fail to reach — one finding, never one per agent



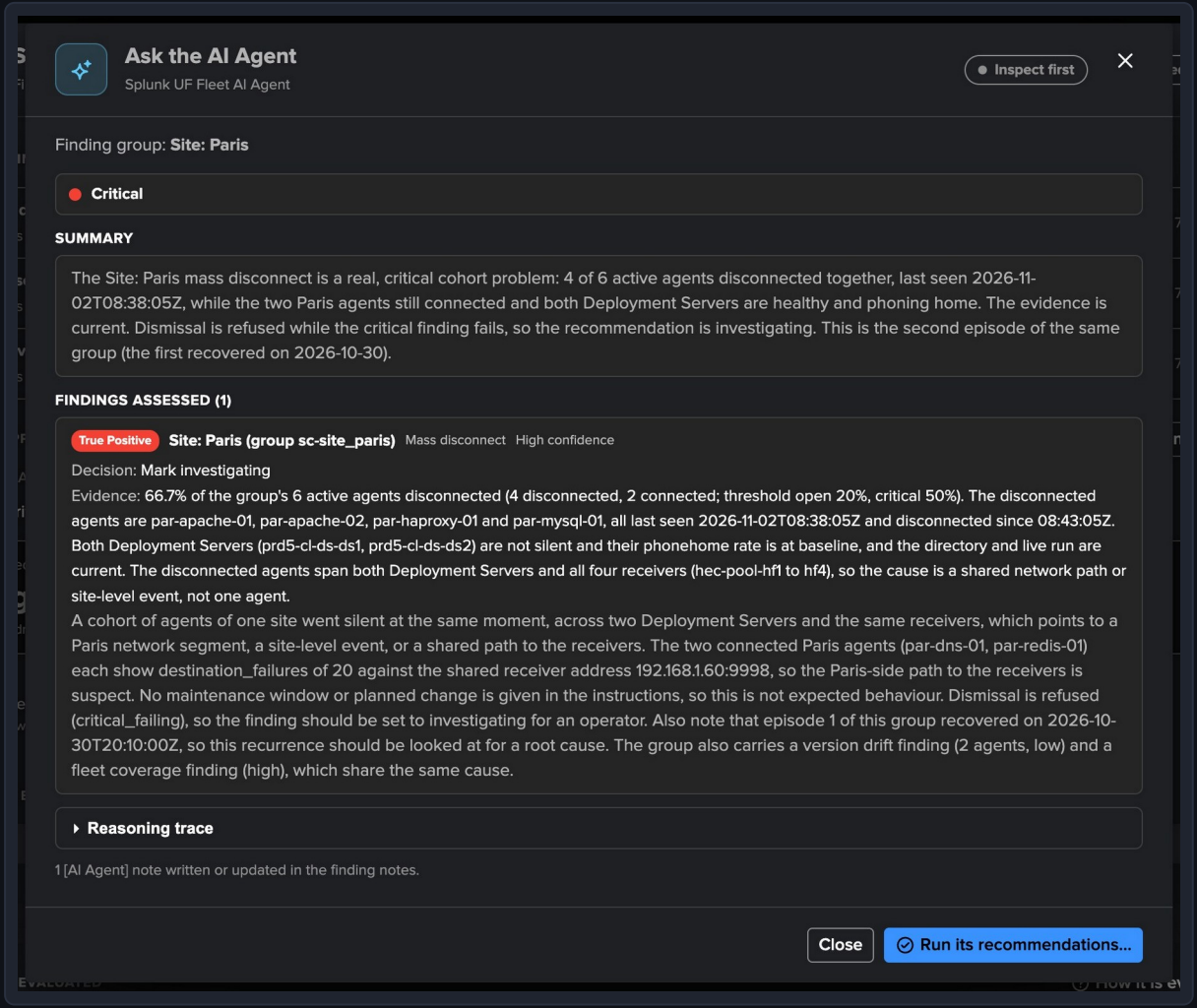
1. **Measured**
   The agents failing to reach it for 15 minutes or more: 16.

2. **Every agent failing**
   How long without a break, and its failure lines.

3. **By group**
   The groups the failing agents belong to: London, New York, the roles.

4. **The cause is the receiver**
   A heavy forwarder of the receiving tier, stopped: see the Splunk S2S tenant.

The Splunk UF Fleet AI Agent — a review on demand or on a schedule



1. **What it reads**
   The agent's directory row, its Deployment Server, the fleet, collection health.

2. **Its tools**
   Fleet overview, agents, Deployment Servers, log evidence, trends.

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

4. **It recommends**
   Never Forgets an agent, never adds an expected one: an operator decides.

WHERE WE ARE

One tenant, every agent and Deployment Server

Set up

a tenant on the local search tier, two Deployment Server API accounts, the corporate scope

Configured

a CMDB on every agent, groups by site, role and environment, expected agents

Followed

the fleet, an agent up close, the Deployment Servers and their pace

Judged

a site gone dark as one finding, an expected agent missing

What made it work

The data path itself

Connected means a receiver saw it: no heartbeat to deploy, no agent polled.

Cohorts, not storms

A group or a fleet disconnecting together is one finding holding its members.

Bounded at any size

Aggregations over indexed terms, never a search per agent.