SCOM Pulse Administration Guide

Deployment, access, settings, SCOM administration, backup and recovery

Product	**SCOM Pulse**
Document version	**1.3**
Verified against the live application	**16 September 2026**
Audience	**SCOM administrators and SCOM Pulse administrators**

Verified against a live SCOM Pulse deployment at /Pulse on a SCOM 2025 management group, signed in as a SCOM Administrator. Server-side procedures that could not be performed from a workstation are marked unverified and listed in Documentation-Verification.md.

Contents

1 About this guide 5

1.1 The boundary 5

1.2 Who this guide is for 5

1.3 How this guide was verified, and what that means for you 5

2 Architecture, prerequisites and supported platforms 7

2.1 Components 7

2.2 Prerequisites 8

2.3 Identity, ports and transport 8

2.4 Storage layout 9

2.5 Supported SCOM versions 10

2.6 Optional: direct SQL reads from the Operations Manager database 10

2.7 Optional: the data warehouse 11

3 Installing and configuring SCOM Pulse 12

3.1 Before you start 12

3.2 Install 12

3.3 What setup does 13

3.4 Verifying an installation 14

3.5 First sign-in 14

3.6 The SCOM Administrators role and its default member 14

3.6a The SCOM Pulse management pack 15

3.7 Licence and trial 16

3.8 Initial verification 17

4 The access model 18

4.1 One model 19

Roles made from the Delegated administrator profile 21

4.2 Several roles: what you see and where you can act 22

4.3 The role data Pulse is working from 23

4.4 Access diagnostics 24

4.5 What each surface requires 25

Where Pulse asks for more than SCOM does, on purpose 26

4.6 Emergency admin exceptions 27

5 Settings **28**
5.1 Access 28
5.2 Shared dashboards 29
5.3 Sync & data 30
Operations Manager data source 32
Performance charts 33
Data warehouse 34
Objects & groups 35
5.4 Site & display 36
5.5 Service & storage 37
6 Agents and management servers **39**
6.1 The agent inventory 39
UNIX/Linux computers 43
Adding UNIX/Linux computers 44
Upgrading UNIX/Linux agents 47
Moving, removing and uninstalling UNIX/Linux computers 48
6.2 Discovering computers 51
6.3 Installing agents 53
6.4 Management servers 55
7 Management packs and groups **56**
7.1 The management pack list 56
7.2 Creating an unsealed pack 58
7.3 Groups 59
8 Channels, subscribers and subscriptions **63**
9 Run As, resource pools, connectors and user roles **66**
9.1 Run As accounts and profiles 66
9.2 Resource pools and connectors 67
9.3 SCOM user roles 69
10 Watchers **71**
10.1 The three recipes 71
10.2 Choosing the watcher computer 72
10.3 Creating a watcher 72
11 The SCOM operational database **73**

11.1 Storage usage ... 73
11.2 Retention and grooming ... 74
12 The SCOM data warehouse ... 76
12.1 Read the header before the numbers ... 76
12.2 Datasets, aggregations and retention ... 77
12.3 The retention preset ... 79
12.4 Permissions and capability ... 80
13 Sharing and audit ... 81
13.1 Published links ... 81
13.2 The audit trail ... 82
14 Operations, backup and recovery ... 84
14.1 Daily health ... 84
14.2 Backups ... 84
14.3 Integrity ... 85
14.4 Restore ... 85
14.5 Upgrades ... 86
14.6 Repair and uninstall ... 86
14.7 Troubleshooting and an escalation bundle ... 87
15 Boundaries and checklists ... 88
15.1 What still needs the Operations console ... 88
15.2 Daily ... 88
15.3 Weekly ... 88
15.4 Before an upgrade ... 89
15.5 Restore validation ... 89
16 Glossary and configuration reference ... 90
16.1 Glossary ... 90
16.2 Paths ... 91
16.3 Commands ... 91
16.4 Application paths ... 91

1 About this guide

This guide covers two different jobs that happen to share a screen: administering **SCOM Pulse itself**, and administering **System Center Operations Manager through Pulse**. Knowing which of the two you are doing tells you where a setting lives, who may change it, and what breaks if it is wrong.

1.1 The boundary

	Administering Pulse	Administering SCOM through Pulse
Where	**Settings** (`/Pulse/Settings`)	**Administration** (`/Pulse/Admin`)
What it changes	Pulse's own behaviour: how often it polls, what it caches, its backups and audit trail	Real Operations Manager objects: agents, management packs, groups, channels, subscriptions, Run As distribution, user roles, retention
Stored in	Pulse's local SQLite database	The SCOM operational database and management packs
Visible to the Operations console	No	Yes — these are SCOM's own objects
Undo	Change the setting back	Whatever SCOM's own undo is, if any

A third area sits between them: Pulse's **cache**, its **backups** and its **audit trail**. They are Pulse's, but they are the record of what you did to SCOM.

1.2 Who this guide is for

You administer Operations Manager, or you administer the Pulse deployment, or both. You are assumed to know SCOM: management packs, sealed and unsealed, user roles and profiles, grooming, the data warehouse. You are not assumed to have installed Pulse before.

1.3 How this guide was verified, and what that means for you

Every screen in this guide was opened in a live Pulse deployment and captured as it was found. Forms were opened and cancelled. **No administrative change was executed while writing it**: nothing was imported, removed, groomed, published, installed or submitted. Where a procedure ends in a write, the steps and expected result come from the application's own text and from the reviewed source behaviour, and the procedure is marked as not executed.

Three kinds of claim are distinguished throughout:

- **Observed** — seen in the live deployment. Values quoted from it are that deployment's, not product defaults, unless the application itself states a default.
- **From the product reference** — behaviour documented in the reviewed code but not exercised here.
- **Unverified** — could not be confirmed from a workstation browser, typically because it needs access to the Pulse server console.

`Documentation-Verification.md`, delivered alongside this guide, lists all of it.

> **Important — Which version you are running**
>
> This guide documents **SCOM Pulse 1.3**, and deliberately does not name a patch version, because the product states its own and a manual that repeated it would only go stale. Since 1.2.12, **Settings → Service & storage → Service & cache** carries a **Version** row: the website's own build, and beside it whether the Pulse service reports the same one. A mismatch is called out there in red, because the two share a database and an action-signing key and an interrupted upgrade can leave one of them behind. `PulseSetup.exe -verify` and Apps & features remain the authority on what is installed on a server, and are worth checking when the two halves disagree. Internal 3.x numbers are a development series and are not public release versions.

2 Architecture, prerequisites and supported platforms

2.1 Components

A SCOM Pulse deployment is three things on one Windows server:

Component	What it does
Pulse.Web	An ASP.NET MVC 5 application in IIS. Renders every page, reads the local cache, and submits actions
SCOMPulse service	A Windows service. Syncs from SCOM through the Operations Manager SDK and executes queued actions
SQLite database	`pulse.db` on local disk. Holds the cache, settings, dashboards, preferences, the action queue and the audit trail

The web tier and the service **share the database file**, and the service — not the web tier — talks to SCOM.

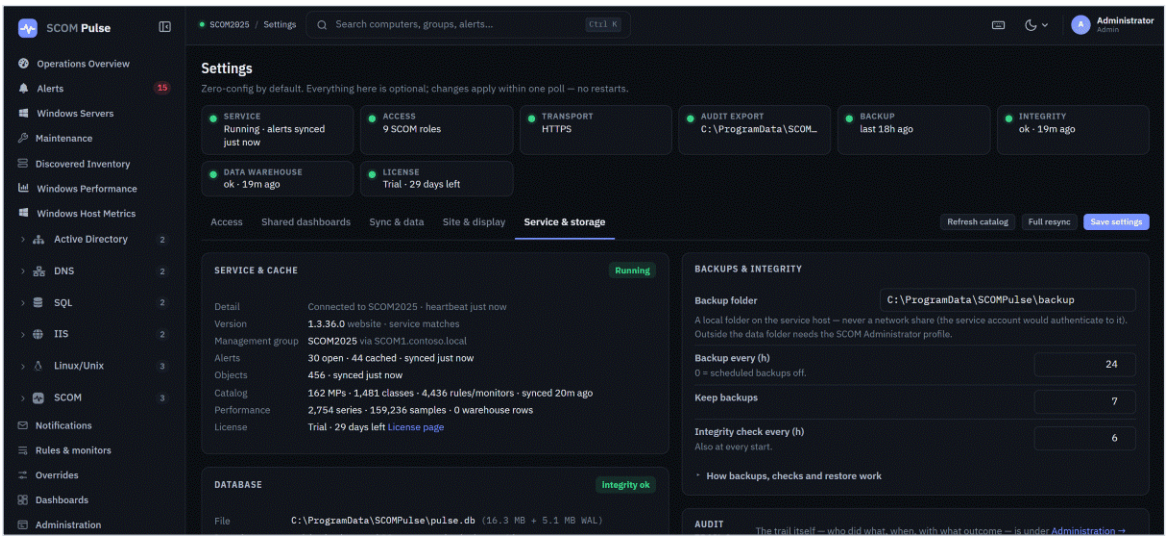


Figure 1 Settings → Service & storage: the service's live state, the management group it is connected to, what it has cached, and the database file with its integrity status.

> **Caution — Unsupported layouts**
>
> Putting the web tier and the service on **different machines**, hosting the SQLite database on a **UNC path**, or running a **web farm** against one SQLite file are all unsupported. SQLite's locking assumes a local disk and a single host.

2.2 Prerequisites

- **Windows Server 2016 or later.**
- **.NET Framework 4.6.2 or later.**
- **IIS** with ASP.NET 4.x, **Windows Authentication** and the supporting web features.
- **A dedicated Windows server, or a SCOM management server.** Pulse ships the Operations Manager client libraries it needs, so a dedicated server does not need the Operations console. Where a management server or the console is installed on the host, the service uses that installation's SCOM client, which matches the management group; on a server with neither it uses the SCOM 2019 client it ships with ("2.5 Supported SCOM versions").
- A **domain service account** for the SCOMPulse service. For full functionality the installation guidance calls for that account to be a member of **Operations Manager Administrators**. A lesser account limits what Pulse can do.
- **Recommended:** read access for that account to the **Operations Manager database**, so Pulse can read it directly ("2.6 Optional: direct SQL reads from the Operations Manager database"). Without it Pulse reads everything through the SDK.

> **Important — The service account is not the end-user authorisation**
>
> Everything Pulse does in SCOM is done by the **service account**; Pulse separately records which signed-in user asked for it. Those are two different checks. Giving the service account broad rights does not give your operators broad rights — their access is decided by "4 The access model" — and it does mean the audit trail, not SCOM's *Modified By*, is where you find out who did something.

2.3 Identity, ports and transport

Users are identified by **Windows Authentication**. There is no local account store and no password for Pulse to hold. There is no sign-in page either, unless you turn on the preview one for HTTPS installs (*Installation Guide*, *Sign people in without a password box*).

Binding and port depend on what already exists in IIS on that server — an existing site, an existing application, a certificate you already have. There is no universal Pulse port and no guarantee of a

particular virtual directory. The deployment reviewed for this guide is published at `/Pulse`; that is one deployment's choice, not a product constant.

> **Caution — Plain HTTP withholds a feature and exposes sign-in**
>
> Pulse's own Settings page states the transport. The deployment used for this guide is served over **HTTPS** and says so (Figure 7); a site on plain HTTP reads *"plain HTTP — not encrypted; Configure SCOM Pulse → pick a certificate"*.
>
> **One feature is withheld outright on HTTP**: running a task under a per-run execution account. The dialog offers only the task's configured account and says *"Open SCOM Pulse over HTTPS to supply task credentials."* If your operators need that, a certificate is not optional. Windows Authentication over unencrypted HTTP exposes the negotiation to anyone on the path. Use a trusted certificate. The configuration tool offers a suitable existing trusted certificate where it finds one; a self-signed certificate is a lab option, not a production answer. Do not disable authentication or override corporate browser policy to make access work.

2.4 Storage layout

	Typical path
Install root	`C:\Program Files\SCOM Pulse\` with `Service`, `Setup` and `Web` beneath it
Data root	`C:\ProgramData\SCOMPulse`
Database	`C:\ProgramData\SCOMPulse\pulse.db` (plus `-wal` and `-shm` sidecars)
Backups	`C:\ProgramData\SCOMPulse\backup\`
Audit export	`C:\ProgramData\SCOMPulse\audit\pulse-audit.jsonl`
Setup log	`C:\ProgramData\SCOMPulse\setup.log`
Action signing key	`action.key`, beside the database

> **Important — Confirm the paths on your server; do not assume the defaults**
>
> A configured data directory, or the `PULSE_DB` environment variable, moves the database. The reviewed deployment reported the defaults above on its Settings page — read yours there rather than assuming.

Both the service identity and the IIS application pool identity need **Modify** access to the data folder. `action.key` signs queued actions: protect it, and understand that a different or regenerated key makes already-queued actions fail closed and need resubmitting.

2.5 Supported SCOM versions

Pulse is intended to work with **Operations Manager 2016 UR7 or later, 2019, 2022 and 2025**. **SCOM 2012 R2 is not supported** — the SDK differs there in the areas Pulse uses for subscriptions and Run As distribution.

Which SCOM client the service loads depends on the host, not on the management group: a management server or the Operations console on the same host supplies its own client, which matches the management group; a server with neither uses the SCOM 2019 UR1 client that Pulse ships. The service reports which management group it is connected to under **Settings** → **Service & storage**.

> **Note — Read this as support intent, and separately as what was exercised**
>
> Intended compatibility and exercised combinations are not the same thing. For this release, **SCOM 2022 UR1** is the version that was exercised end to end: installed from the MSI on a Windows Server 2025 host with no SCOM components, connected through the shipped client, with user roles — Read-only administrator and Delegated administrator roles included — read identically to the SCOM 2022 client. The deployment used for this guide's figures runs on a **SCOM 2025** management server with that server's own client; it was used to read and capture screens, not to qualify installation, writes or role permissions on SCOM 2025. **SCOM 2016** and **SCOM 2019** management groups were not exercised. Treat every combination other than SCOM 2022 as intended compatibility that has not been qualified for this release.

2.6 Optional: direct SQL reads from the Operations Manager database

Pulse always connects to SCOM through the **SDK**, and every write, and role discovery, goes that way. For **reads** it can also go straight to the **Operations Manager (operational) database**: alerts, knowledge, objects and their properties, groups, monitor state and history, dependencies and relations, events, alert history, live performance samples and the Administration inventory. Direct reads are faster and put less load on the management server. The SDK stays in place as the fallback: a reader whose SQL query fails drops back to the SDK and retries SQL a minute later.

- **Where the database is.** Pulse asks SCOM for the operational database's server and name, the same way it finds the warehouse, so there is nothing to type. Every read checks that the database belongs to the management group Pulse is connected to.
- **Who connects.** The **service account**, with Windows integrated security, from the Pulse server only. The web tier never opens a SQL connection, and the signed-in user's identity is not used.
- **What it needs.** Read access to the operational database, typically `db_datareader`. The SCOM Data Access Service account usually has this already. No `db_owner`, `sysadmin`, write or schema rights are needed.
- **Network.** TCP to the operational database's SQL instance (normally **1433**) from the Pulse server.
- **Encryption.** The connection is always encrypted. By default Pulse trusts the SQL Server's certificate, because operational databases commonly present a self-signed or privately issued one. For strict validation set `SqlReads.TrustServerCertificate` to `false` in `PulseService.exe.config`; a SQL certificate the server does not trust then turns direct reads off.

Configure checks access to both databases as the service account and chooses for you: direct SQL reads **On** if the operational database answers, **Off** (SDK reads) if not; the warehouse **On** if it answers, **Off** if not. It never changes the live performance source. After that the choice is yours, under **Settings → Sync & data** ("5.3 Sync & data"), and a later repair or Configure run keeps what you chose.

2.7 Optional: the data warehouse

Pulse can read the SCOM **data warehouse** for long historical chart ranges and for warehouse administration. The warehouse location is **discovered from SCOM's management group settings** — it is not a server field you type into Pulse. Read-only chart use and warehouse administration are different capabilities with different permissions; see "12 The SCOM data warehouse".

3 Installing and configuring SCOM Pulse

> **Documentation gap — IIS, service and log screens are not pictured**
>
> Everything in this chapter happens on the Pulse server console. The installer's own pages are pictured in the *Installation Guide*; IIS Manager, the Windows service, `setup.log` and the command-line tools are described in words. The procedures were exercised on a lab server, not on the deployment this guide's figures come from.

3.1 Before you start

1. Confirm the prerequisites in "2 Architecture, prerequisites and supported platforms". The host can be a dedicated Windows server or a management server; if your Operations Manager Administrators role still holds only `BUILTIN\Administrators`, read "3.6 The SCOM Administrators role and its default member" first.
2. Create or identify the **domain service account**, and add it to **Operations Manager Administrators** if you want full functionality. Setup offers the SCOM Data Access Service account, which already qualifies.
3. Decide the **data directory**, and make sure it is on local disk with room for the database and its backups.
4. Decide the **binding**: whether Pulse joins an existing website as `/Pulse` or gets its own, and which certificate.
5. Check the **SCOM user roles**. They are the whole of who can use Pulse ("4 The access model"): the people who will use it must already be in one, and nothing else needs preparing.

3.2 Install

One wizard installs SCOM Pulse *and* configures it; there is no second tool to run on a first install.

1. Run the **SCOM Pulse MSI** as an administrator and accept the licence and installation folder.
2. On **Connect to SCOM**, confirm the management server and service account, enter the password (a gMSA needs none), and optionally paste a licence key. **Advanced** covers the data folder, host name, website placement, transport and ports.
3. **Next** validates the details before anything is installed; **Install** runs the thirteen configuration steps below and waits until Pulse has reached SCOM; **Finish** opens the site.

The *Installation Guide* walks the wizard field by field, with figures, and covers unattended installation. The

separate **Configure SCOM Pulse** tool in the Start Menu is for later changes and repair ("14 Operations, backup and recovery").

3.3 What setup does

Setup works through thirteen steps, in the wizard and again whenever **Configure SCOM Pulse** or `PulseSetup.exe -verify` runs, and shows the result of each:

1. **Preflight** — operating system, .NET, and that the Operations Manager client libraries load.
2. **Validate service account.**
3. **IIS features** — IIS, ASP.NET and Windows Authentication, installed if missing.
4. **Log on as a service** for the service account.
5. **Data folder and permissions.**
6. **Action signing key** — `action.key`.
7. **Write configuration files.**
8. **Windows service** registration.
9. **IIS site and application pool** — a site of Pulse's own, or `/Pulse` under the website you chose.
10. **HTTPS binding.**
11. **Database and trial.**
12. **Start service and connect to SCOM**, followed by the **database access checks**: the service, running as its own account, tries the Operations Manager database and the data warehouse and records the results ("2.6 Optional: direct SQL reads from the Operations Manager database"). A database that cannot be reached does not fail the installation; Pulse simply reads through the SDK, or leaves the warehouse off.
13. **Open the site.**

It writes `C:\ProgramData\SCOMPulse\setup.log`. Read that file when a step fails; it is the first place to look, ahead of the Windows event log. If configuration fails during a first install, the installation fails as a whole and takes back what it created — the service, application pool and website — while leaving IIS components, the data folder and the log in place for a retry.

> **Important — Binding choices are not interchangeable**
>
> The tool offers bindings based on the IIS sites and applications that already exist. Choose deliberately, and do not overwrite an existing site that is serving something else. Whatever you choose becomes the prefix in every Pulse URL — `/pulse` in the reviewed deployment, which IIS treats the same as `/Pulse` — and that prefix must be preserved in links, bookmarks and any automation.

3.4 Verifying an installation

`PulseSetup.exe -verify` is the documented, **non-mutating** verification mode. Use it after installing, after repairing, after upgrading, and whenever you need to know what is actually deployed. It is the authority on the installed version; inside Pulse, **Settings** → **Service & storage** shows the website's version and whether the service reports the same one.

3.5 First sign-in

Authorized users go directly to **Operations Overview**, or their chosen home dashboard. There is no welcome screen or setup-completion step. Access comes from existing SCOM user roles ("4 The access model").

If role synchronization is pending or the signed-in account has no monitoring access, Pulse shows an **Access status** page with next steps and a **View my access** link. The `/Pulse/GetStarted` address serves that status page; authorized users visiting it are redirected home.

Choose a personal theme in the top bar. Administrators can optionally change the default under **Settings > Site & display**. Manage team membership through SCOM user roles in the Operations console or **Administration > User roles**.

3.6 The SCOM Administrators role and its default member

A Pulse administrator is a **SCOM Administrator** — nothing else makes one. That puts one SCOM default in your way on a new management group: the Operations Manager Administrators role ships holding only **`BUILTIN\Administrators`**, which is the *management server's* local Administrators group.

A machine-local name only means the same thing on the machine it belongs to. So Pulse counts such a member **only when Pulse itself runs on a management server**. Anywhere else it is ignored: the web server's own local administrators are not people SCOM granted anything to, and Pulse does not treat them as if they were.

Pulse tells you which situation you are in — on the **Access status** page, on **Settings** → **Access**, on **/Pulse/Access**, and in the service step of `PulseSetup.exe` and `-verify`:

What Pulse finds	What it means	What to do
The role holds an Active Directory group or account	Nothing to say, and nothing is shown	—
Only the default member, and Pulse is on a management server	It works: every local administrator of that server is a SCOM administrator, in the console and in Pulse	Microsoft recommends an AD group instead; add one when you can

What Pulse finds	What it means	What to do
Only the default member, and Pulse is **not** on a management server	**Nobody can be verified as a SCOM administrator**, so nobody can administer Pulse	Add an AD group to the role, or install Pulse on a management server
An AD group **and** the default member, Pulse not on a management server	It works through the AD group; the page notes the local entry does not count here	—

To add the group: Operations console → **Administration** → **Security** → **User Roles** → **Operations Manager Administrators**. Then sign out of Windows and back in, and refresh the page.

> **Important — Replacing the default is the right fix, not a workaround**
>
> A named AD group is what Microsoft recommends for the Administrators role in any case: it is auditable, it survives a management server being rebuilt, and it means the same people on every server. If you need to get into Pulse before that change can be made, use an emergency admin exception ("4.6 Emergency admin exceptions").

3.6a The SCOM Pulse management pack

A management pack ships with the product, at `C:\Program Files\SCOM Pulse\ManagementPacks`, beside an `IMPORT.txt`. **Setup does not import it** — an import changes your management group, so it stays your decision, and Pulse runs perfectly well without it.

It monitors **Pulse itself** from SCOM: installed-product discovery, an alert when the service stays down, a bounded restart attempt, and warnings before the licence expires. The licence state reaches it through a status file the product writes beside the database, not through events or the SQLite store.

Exactly one file is laid down — the **sealed** `.mp` when the build had one, the unsealed `.xml` otherwise — so there is never a question of which to import.

> **Important — Sealed versus unsealed decides where your tuning survives**
>
> SCOM stores an override *inside* a management pack. Overrides against an **unsealed** pack live in ours and the next version overwrites them; overrides against a **sealed** one have to go in a pack of your own and therefore survive an upgrade. If you are given the unsealed `.xml`, expect to re-apply tuning after an upgrade, and re-importing the XML discards overrides stored in it.

> **Caution — Uninstalling Pulse does not remove it**
>
> Uninstall does not reach into your management group. Delete the pack from the Operations console if you no longer want it.

3.7 Licence and trial

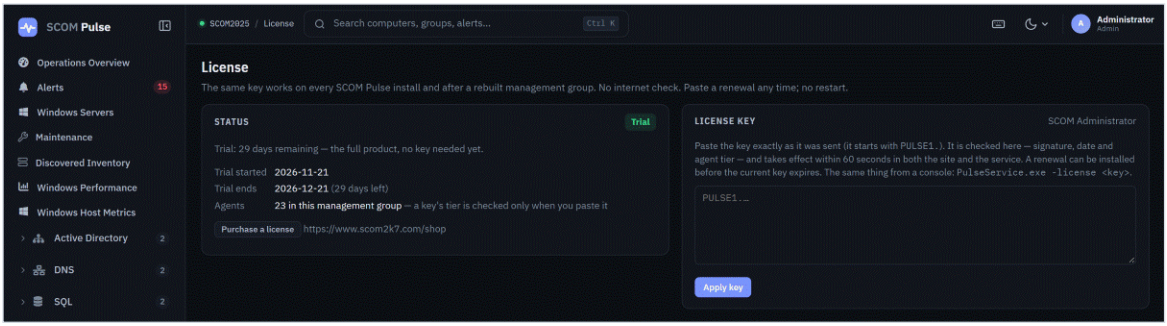


Figure 2 The **License** page. In the reviewed deployment the product was on trial; no key is present, and none is shown here.

The initial trial is documented as **30 days**. The reviewed deployment reported "*Trial: 30 days remaining — the full product, no key needed yet*", with the trial started on 16 September 2026 and ending 16 October 2026.

Applying a key:

1. Open **/Pulse/License**.
2. Paste the key exactly as supplied. Keys begin `PULSE1.`
3. Click **Apply key**.

The key is checked locally — signature, date and agent tier — with **no internet check**, and takes effect within 60 seconds in both the site and the service. The same key works on every install and after a rebuilt management group, and a renewal can be installed before the current key expires. From a console the equivalent is `PulseService.exe -license <key>`.

> **Caution — Never put a licence key in a document, ticket or screenshot**
>
> The key is a secret. This guide's licence figure was captured on a deployment with no key applied. Verify expiry behaviour in your own environment rather than assuming which features keep working after a trial ends — that was not tested here.

3.8 Initial verification

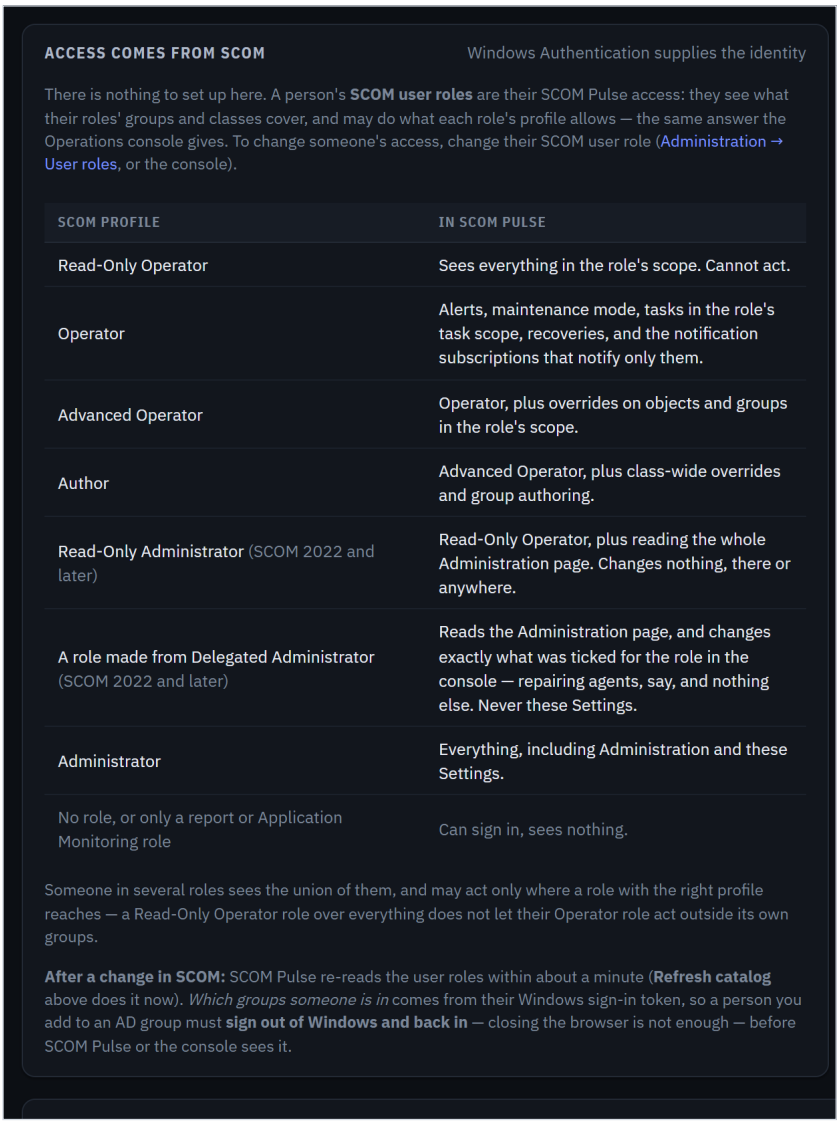
After setup, confirm in this order:

1. `PulseSetup.exe -verify` on the server reports a healthy install.
2. The **SCOMPulse** service is running.
3. **Settings** opens and the status strip shows **SERVICE: Running**, a recent sync, and synchronized SCOM roles (Figure 7).
4. **Settings** → **Service & storage** names the expected management group and shows non-zero alert and object counts (Figure 1).
5. **/Pulse/Access** shows your own account matched to the SCOM user roles you expect ("4.4 Access diagnostics").
6. Reopening Pulse takes you directly to your home dashboard without a setup prompt.

4 The access model

Access to Pulse comes from **SCOM user roles**, and from nothing else. There are no Pulse role lists to fill in and no mode to choose: a person sees what their SCOM roles cover and may do what each role's profile allows — the same answer the Operations console gives them.

4.1 One model



ACCESS COMES FROM SCOM — Windows Authentication supplies the identity

There is nothing to set up here. A person's **SCOM user roles** are their SCOM Pulse access: they see what their roles' groups and classes cover, and may do what each role's profile allows — the same answer the Operations console gives. To change someone's access, change their SCOM user role (Administration → User roles, or the console).

SCOM PROFILE	IN SCOM PULSE
Read-Only Operator	Sees everything in the role's scope. Cannot act.
Operator	Alerts, maintenance mode, tasks in the role's task scope, recoveries, and the notification subscriptions that notify only them.
Advanced Operator	Operator, plus overrides on objects and groups in the role's scope.
Author	Advanced Operator, plus class-wide overrides and group authoring.
Read-Only Administrator (SCOM 2022 and later)	Read-Only Operator, plus reading the whole Administration page. Changes nothing, there or anywhere.
A role made from Delegated Administrator (SCOM 2022 and later)	Reads the Administration page, and changes exactly what was ticked for the role in the console — repairing agents, say, and nothing else. Never these Settings.
Administrator	Everything, including Administration and these Settings.
No role, or only a report or Application Monitoring role	Can sign in, sees nothing.

Someone in several roles sees the union of them, and may act only where a role with the right profile reaches — a Read-Only Operator role over everything does not let their Operator role act outside its own groups.

After a change in SCOM: SCOM Pulse re-reads the user roles within about a minute (**Refresh catalog** above does it now). *Which groups someone is in* comes from their Windows sign-in token, so a person you add to an AD group must **sign out of Windows and back in** — closing the browser is not enough — before SCOM Pulse or the console sees it.

Figure 3 Settings → Access: what access is, profile by profile. There is nothing here to configure.

Your existing SCOM user roles determine access to supported SCOM operations in Pulse.

Everyone signs in with their Windows account. Pulse matches the sign-in against the members of your SCOM user roles, reads each role's profile and its actual permissions from the connected management group, and gives the person exactly that — per role, and within that role's group, class and task scope:

SCOM profile	In SCOM Pulse
Read-only operator	Sees everything in the role's scope — alerts with their history and knowledge, objects with their monitors, events and performance. Cannot act
Operator	Alert actions (any resolution state the management group has), maintenance mode and schedules, tasks in the role's task scope with the role's task-account permissions, diagnostics and recoveries, and the notification subscriptions that notify only them
Advanced operator	Operator, plus overrides on objects and groups in the role's scope
Author	Advanced operator, plus class-wide overrides and group authoring. Reaches **Administration** for the **Groups** tab only
Read-only administrator (SCOM 2022 and later)	Monitors like a Read-only operator, and reads the whole **Administration** page, the **Management servers** page and every notification subscription. Changes nothing anywhere. Does not open **Settings**
A role made from **Delegated administrator** (SCOM 2022 and later)	Reads the **Administration** page, and changes exactly what was ticked for the role in the console's **Create User Role** wizard — see below. Never **Settings**
Administrator	Everything, including **Administration**, **Settings**, the audit trail and the licence
Report Operator, Report Security Administrator, Application Monitoring Operator, or no role at all	Can sign in. Sees nothing

To change somebody's access, change their SCOM user role — in the Operations console or under **Administration → User roles**. That is the whole procedure. There is nothing to migrate and no compatibility mode to select: the model is the same on SCOM 2016, 2019, 2022 and 2025.

SCOM version	Traditional profiles and scoped roles	Read-only administrator	Delegated administrator custom roles
2016	Supported	n/a — the profile does not exist	n/a
2019	Supported	n/a	n/a
2022	Supported	Supported	Supported
2025	Supported	Supported	Supported

How each cell was verified — and which were not exercised against a live management group for this release — is in `SUPPORTED-MATRIX.md`. In short: the traditional profiles and Read-only administrator were verified against a live SCOM 2022 management group; Delegated administrator custom roles were verified against the same management group with one role for each permission category in the console's wizard, and each allowed exactly what was ticked.

Roles made from the Delegated administrator profile

On SCOM 2022 and later the console's **Create User Role** wizard builds a custom role from the Delegated administrator profile by ticking permissions. Pulse honours these, each one separately:

Ticked in the wizard	In SCOM Pulse
Agent Management — Deploy Agents; Repair Agents; Uninstall Agents; Agent Pending Actions; Update Agent Settings	Installing agents; repairing and updating agents, and approving or rejecting pending agents (Repair Agents covers these too); uninstalling agents and deleting them from the management group (Uninstall Agents covers both); pending actions; primary and failover assignment and the proxy setting
Notification Management	Channels, subscribers and subscriptions. Create/edit and delete are separate permissions
Account Management — the Run As items	Create and update Run As accounts, delete Run As accounts, manage Run As profiles
Connector Management	Delete a connector
Global Settings	Operational database grooming ("11 The SCOM operational database")
Management Pack Authoring	Overrides, authoring, and importing, exporting or deleting a management pack
Monitoring Permissions	Executing tasks, diagnostics and recoveries; maintenance mode; managing alerts — each separately. A role with only **Manage Alerts** cannot run tasks or start maintenance

The word *Administrator* in such a role's profile grants nothing by itself: a role with no permission ticked reads the **Administration** page and changes nothing.

> **Important — If a role's permissions cannot be read**
>
> A traditional profile keeps working, because its permission list is fixed. A custom Delegated administrator role grants **nothing** until its permissions can be read — never a guess. `/Pulse/Access` says which of the two happened ("4.4 Access diagnostics").

> **Important — Signing in is not a grant**
>
> Anyone who can authenticate can open Pulse, and that alone shows them nothing. Somebody in no SCOM user role sees "*Your account hasn't been granted monitoring access. Ask your SCOM administrator to assign an appropriate SCOM user role.*" with a **See your access details** link, and lands on the **Access status** page, which says who they are in SCOM.

Consequences worth planning for:

- **SCOM sees the service account.** Pulse carries out every SCOM change as its service account. SCOM's own records show that account; the person's name is in Pulse's audit trail. Pulse decides what a person may ask for — it does not make SCOM see the person.
- **A revoked permission is honoured even for an action already queued.** The request is checked again when it runs, not only when it was made.
- **Roles that cannot be read fail closed.** Until Pulse has read the SCOM user roles — the first minutes of a new installation, or a management group it cannot reach — nobody but an emergency admin exception gets in. An outage cannot silently widen access. People waiting see "*Monitoring access is waiting for SCOM role verification. Refresh after the service synchronizes user roles.*"
- **A change takes two steps to arrive.** Pulse re-reads the roles within about a minute of a change (**Refresh catalog** does it at once). But *which groups somebody is in* comes from their Windows sign-in token, so a person you add to an AD group must **sign out of Windows and back in** — closing the browser is not enough — before Pulse, or the console, sees it.

> **Note — Published links are separate**
>
> A shared dashboard link is a Pulse feature, not a SCOM role. It is read-only, it carries the audience its publisher chose, and it shows what its **publisher's** SCOM roles cover, narrowed to the groups the publisher picked. The person opening it needs no SCOM role, has not acquired one by opening it, and can never act through it.

On screen Pulse summarises a profile in one word — **Viewer** for read-only, **Operator** for Operator, Advanced operator and Author, **Admin** for Administrator. Those words are a label, not a second thing to grant.

4.2 Several roles: what you see and where you can act

Somebody in several SCOM user roles **sees** the union of them, and may **act** only where a role that grants the operation reaches. Permissions are per role and are never added together across roles: the role that grants an operation must also cover the target.

Caution — Seeing an object is not the right to change it

A Read-only operator role over everything, plus an Operator role over the SQL group, lets a person see the whole management group and act on SQL alone. Asked to acknowledge an alert outside it, Pulse answers *"Your SCOM user roles let you see this but not change it: no role of yours with the Operator profile or better covers it."* The same holds one step up: an Advanced operator role over one group does not carry overrides into another role's objects. And it holds for administration: a role that opens the **Administration** page plus a separate Operator role does **not** make someone a notification administrator.

Some operations carry a scope rule of their own. These are SCOM's rules, and Pulse applies them:

- **Agents.** Repairing, uninstalling and changing the settings of an agent are limited to agents inside the granting role's groups. Deploying agents is not group-limited. Approving a pending agent needs a role that is not limited to groups, because a pending agent is in no group yet.
- **Management packs.** Importing or deleting a management pack needs a role that is not limited to certain classes.
- **Notification subscriptions.** A subscription cannot reach further than the groups of the role that grants managing subscriptions. Someone whose granting role is limited to groups must name groups in the subscription.

Caution — Object scope combines group AND class within a role, then OR across roles

Inside one SCOM user role, the group scope and the class scope both apply — an object must satisfy both. Across several roles, the results are combined. Describing it as "the union of all groups and all classes" is wrong and will over-estimate what someone can see. Task scope works the same way, role by role. When the answer matters, read the access diagnostics page for that person rather than reasoning about it.

4.3 The role data Pulse is working from

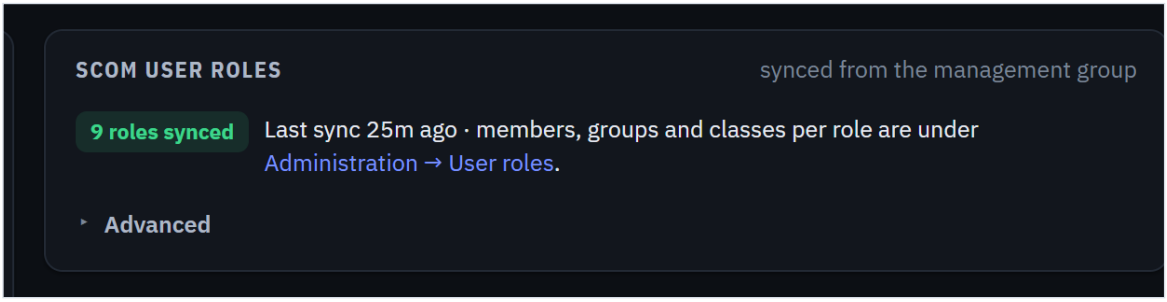


Figure 4 Settings → Access: how many SCOM user roles have been synced, when, and anything Pulse has to say about the Administrators role.

The **SCOM user roles** card shows how many roles Pulse has read and when. The members, groups and classes of each role are under **Administration** → **User roles**. If no roles have been read yet the card says so in red, because until they are nobody can be given access.

The same card carries the finding about the SCOM Administrators role's default member ("3.6 The SCOM Administrators role and its default member"), and an **Advanced** section holding the **role cache duration** and the **emergency admin exceptions** ("4.6 Emergency admin exceptions").

4.4 Access diagnostics

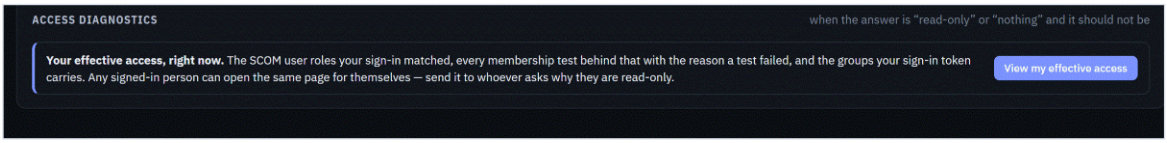


Figure 5 Settings → **Access**: the way to the diagnostics page.

View my effective access opens `/Pulse/Access` for the signed-in account. Any signed-in person can open it for themselves, which makes it the right thing to ask for when somebody reports being read-only, or seeing nothing.

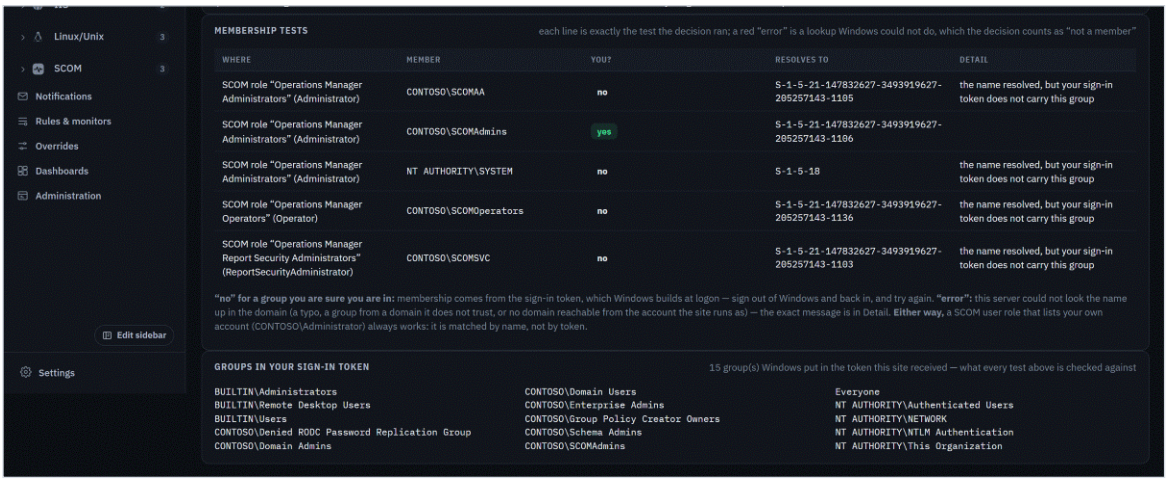


Figure 6 `/Pulse/Access` — the membership tests behind the decision, and the groups the sign-in token carried. Security identifiers are covered; the lab's account and group names are left readable.

The top of the page gives the answer — the SCOM user roles the sign-in matched, the effective profile, and the scope.

Beneath it, **What each of your roles grants** lists every matched role with its profile (a custom role reads *Custom, from Delegated Administrator*) and what it grants in Pulse, noting where that holds within the

role's groups or classes only. A role that does not grant gives one of three different answers, and they mean different things:

- **Nothing here** — the role was read and holds no permission Pulse uses. A Report Operator role reads this way.
- **permissions not read** — the role's permission list could not be read at the last sync. A traditional profile keeps working from its fixed list, and the row says so; a custom Delegated administrator role grants nothing until it can be read.
- **not readable by this SCOM client** — the role uses a profile the SCOM client on this server cannot describe (see the note on the SCOM client in `SUPPORTED-MATRIX.md`). It grants nothing.

The **Membership tests** table after that is every test that actually ran. The four outcomes:

- **yes** — the token carries it, or the role lists the account itself.
- **no**, with "*the name resolved, but your sign-in token does not carry this group*" — the group exists; this sign-in is not in it.
- **error** — the server could not look the name up: a typo, a group in an untrusted domain, or no reachable domain from the site's identity.
- **not counted** — a machine-local member such as `BUILTIN\Administrators`, on a Pulse server that is not a management server ("3.6 The SCOM Administrators role and its default member").

> **Important — Membership comes from the Windows logon token**
>
> Adding someone to a group changes nothing until they **sign out of Windows and back in**. Closing the browser is not enough. A SCOM user role that lists the person's own account always works, because it is matched by name rather than by token.

4.5 What each surface requires

> **Caution — This table is a guide; the application is the authority**
>
> Read it as "what to expect", not as a specification. When a specific action matters, test it with a real account and read the access diagnostics.

Surface	Needs
Any page, read	A SCOM user role with a monitoring profile; content bounded by the roles' scope
Alert actions, maintenance, tasks, diagnostics, recoveries	Operator or better — or a Delegated administrator role with the matching **Monitoring Permissions** item — in a role that covers the object

Surface	Needs
Running a task	Decided by SCOM alone: a role that may run tasks, lists the task in its task scope, covers every target, and allows the chosen account option
`/Pulse/Overrides`	Advanced operator or better; a class-wide override needs Author. Or a Delegated administrator role with **Management Pack Authoring**
Group authoring	Author or better
`/Pulse/Notifications` — the subscriptions that notify only you	Operator or better. Never channels
`/Pulse/Notifications` — everyone's subscriptions, subscribers and channels	Administrator, or a Delegated administrator role with **Notification Management** that is not limited to groups. A Read-only administrator reads them all
`/Pulse/Admin` and `/Pulse/ManagementServers`, read — including the agent inventory, role members, Run As account names and channel settings	Administrator, Read-only administrator or a Delegated administrator role. An Author reaches the **Groups** tab only
A change on `/Pulse/Admin`	Administrator, or a Delegated administrator role with the permission for that change ("4 The access model")
`/Pulse/Settings`, **Administration** → **Audit**, the licence key	Administrator
Emergency exceptions, sharing switches	Administrator, plus an explicit confirmation

There is no Pulse setting that decides who may manage notifications, and none that decides which resolution states a person may set. SCOM decides both. The **Resolution states the menu offers** list under **Settings** → **Sync & data** shortens a menu and is not a permission ("5.3 Sync & data").

> **Note — Tasks with wide reach**
>
> Some agent-management tasks reach a long way — running arbitrary PowerShell, deleting an agent, installing software. Whether someone may run one is decided by their SCOM role like any other task; Pulse no longer asks for the Administrator profile on top. It still asks for an explicit confirmation that the server checks, a typed confirmation for the PowerShell one, and records the run in the audit trail.

Where Pulse asks for more than SCOM does, on purpose

A short list, stated plainly so that it does not surprise you in an access review:

- **User roles.** Creating, editing and deleting a SCOM user role needs the SCOM Administrator profile in Pulse, even though SCOM can delegate it. Pulse carries out changes as its service account, so SCOM

cannot apply its own limits to the delegated administrator — and a role edit is the one change that can widen the editor's own access.

- **Group authoring safeguards.** An Author limited to certain groups may add to a static group only objects their Author role already covers, and may not save dynamic rules on an existing group whose membership decides what their own group-limited role reaches ("7 Management packs and groups").
- **Things SCOM has no permission for.** These need the SCOM Administrator profile: Pulse's **Settings**, the licence, the global audit trail, custom SQL cards, administering other people's shared links, **Refresh inventory** and **Refresh catalog**, the data warehouse connection test, data warehouse retention and **Groom now** (these are changes to the warehouse database, not a SCOM permission), resource pools, and watchers.
- **Roles created in Pulse** can use the five traditional profiles. Read-only administrator and Delegated administrator roles are created in the Operations console, and then honoured by Pulse.

4.6 Emergency admin exceptions

One list in Pulse is **not** a SCOM role: **Emergency admin exceptions**, under **Settings** → **Access** → **Advanced**. An account or group on it is a Pulse administrator whatever SCOM says. It exists for the day SCOM cannot vouch for anybody — the roles cannot be read, or the Administrators role names nobody this server can verify ("3.6 The SCOM Administrators role and its default member").

When nobody can sign in to reach that page, add an entry from the server console, elevated:

```
cd "C:\Program Files\SCOM Pulse\Setup"
PulseSetup.exe -grantEmergency DOMAIN\name
```

It adds that one entry and changes nothing else. It applies to the next page the person opens, and it is recorded in the audit trail under the Windows account that ran it. That is the only way in from the server: local Administrators of the web server get no automatic access.

> **Caution — A standing bypass needs an owner**
>
> Give the list an owner, a review date and an audit expectation; keep it **empty** in normal operation; check it during access reviews. Everything an exception does is recorded as *break-glass* in the audit trail. Someone signed in through the list cannot save a list they are no longer on — it would end their own access with the click — so have a SCOM Administrator remove the entry once SCOM is vouching for people again.

The **role cache duration** beside it (15 minutes in the reviewed deployment) is how long a person's matched roles are reused before they are matched again. A completed role sync refreshes everyone at once, so it only matters when no sync has landed.

5 Settings

Settings is Pulse's own configuration. It is for SCOM Administrators, it applies within one poll without a restart, and it is entirely optional — the product runs with none of it changed.

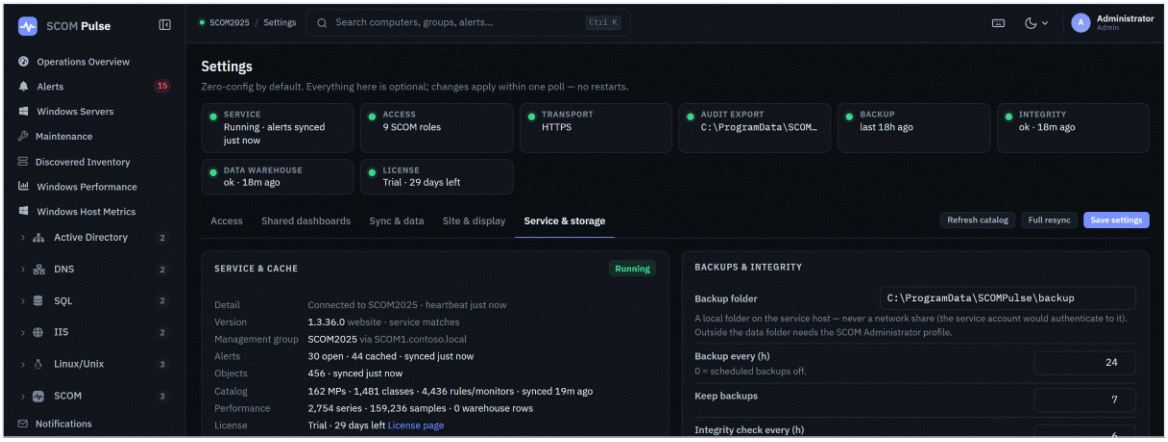


Figure 7 The Settings status strip: service, access, transport, audit export, backup, integrity, warehouse and licence — the fastest health read in the product.

> **Important — Save settings, versus controls that apply immediately**
>
> **Save settings** commits the form. Some controls are not part of it — choosing or removing a logo applies as soon as it is done. **Refresh catalog** and **Full resync** are actions, not settings.

Values quoted below are **what the reviewed deployment held**. Where the application itself states a default — it does for several sharing options — that is called out as a default.

5.1 Access

Covered in "4 The access model". The tab also carries **Signing in without a password box**: the browser policies — Edge's *Configure list of allowed authentication servers* and Chrome's *Authentication server allowlist*, the Windows *Site to Zone Assignment List* and Firefox's two preferences — with the values for the address you opened Pulse with, and **Copy all settings**. Without them, Edge and Chrome usually show a password box to people who open an HTTPS address; the *Installation Guide* (*Sign people in without a password box*) explains why the browser policy is the one to rely on.

5.2 Shared dashboards

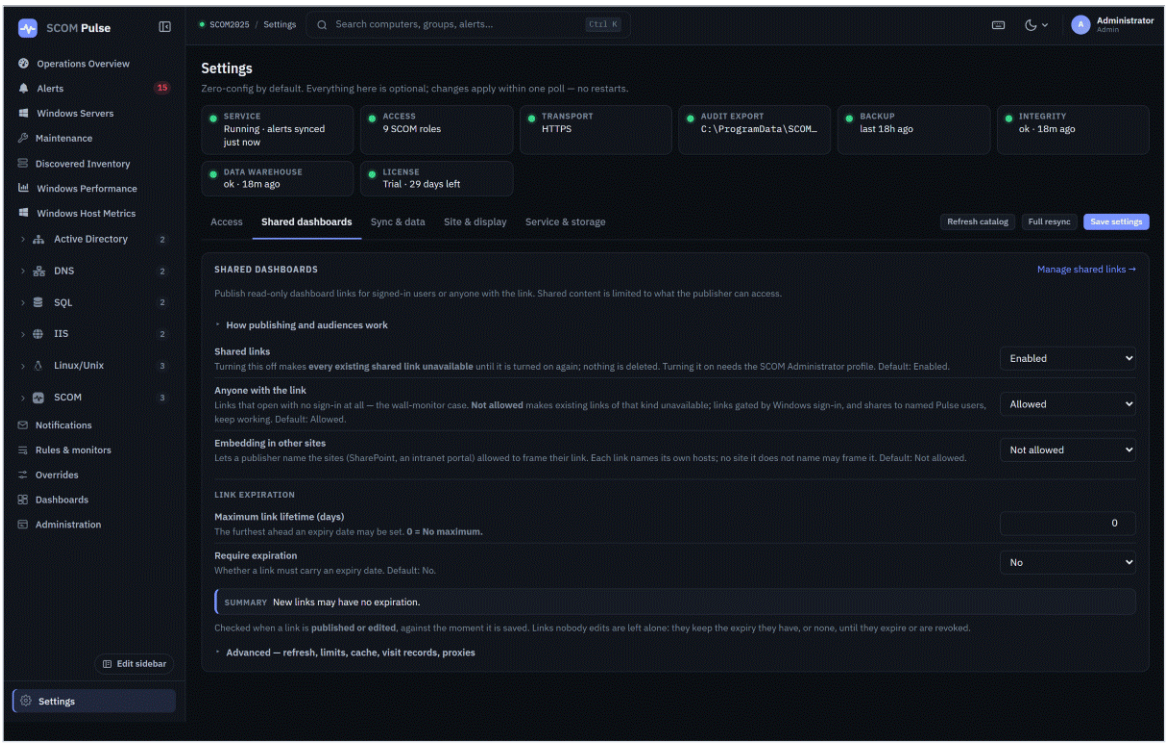


Figure 8 Settings → Shared dashboards: the publication policy for the whole deployment.

Setting	Meaning	Stated default
Shared links	Master switch. Turning it off makes every existing link unavailable until it is turned on again — **nothing is deleted**. Turning it on needs the SCOM Administrator profile	**Enabled**
Anyone with the link	Allows links that open with no sign-in — the wall-monitor case. **Not allowed** makes existing links of that kind unavailable; links gated by Windows sign-in and shares to named Pulse users keep working	**Allowed**
Embedding in other sites	Lets a publisher name the sites allowed to frame their link. Each link names its own hosts	**Not allowed**

Setting	Meaning	Stated default
Maximum link lifetime (days)	Furthest ahead an expiry may be set. `0` = no maximum	`0` in this deployment
Require expiration	Whether a link must carry an expiry date	**No**

The policy is checked **when a link is published or edited**, against the moment it is saved. Links nobody edits keep the expiry they have until they expire or are revoked.

An **Advanced** section holds refresh, rate limits, cache, visit records and proxies. In the reviewed deployment: default refresh **30 s**, **300** requests per minute per viewer, card cache **10 s**, visit records kept **90 days**, pause-when-publisher-away **0** (off), and no trusted proxies.

> **Caution — Set trusted proxies only if links genuinely arrive through a reverse proxy**
>
> The setting exists so that viewer addresses are attributed correctly. Populating it when there is no proxy lets a client spoof its address in the visit records.

5.3 Sync & data

The tab holds five cards: **Alerts** and **Operations Manager data source** on the left; **Objects & groups**, **Performance charts** and **Data warehouse** on the right.

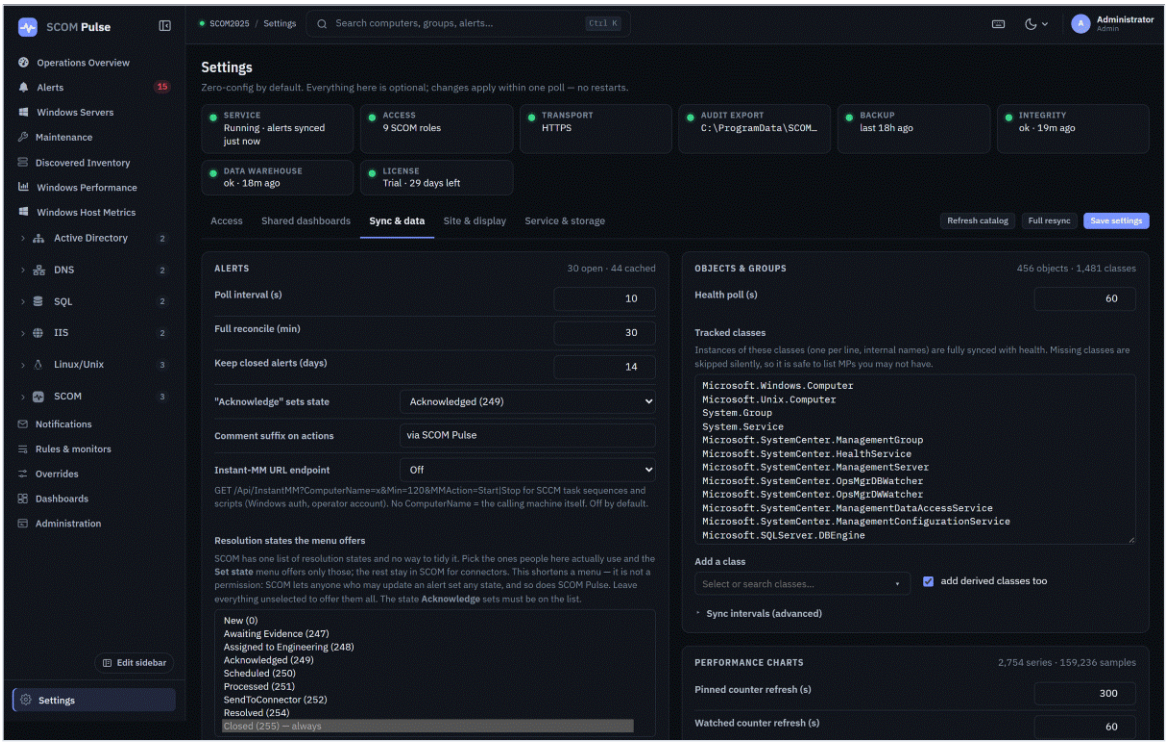


Figure 9 Settings → Sync & data: alert polling, the state that **Acknowledge** sets, the Instant-MM endpoint and the resolution-state shortlist on the left, with **Objects & groups** beside them.

Alerts. Poll interval **10 s**; full reconcile **30 min**; keep closed alerts **14 days**. A closed alert is kept for that many days after it closed — or after it was last changed, if that is later — and that is as far back as an object page's **Recently closed** alerts reach: a period longer than it is offered there as unavailable, with a line saying how long closed alerts are kept and, for administrators, a link to this setting.

"Acknowledge" sets state maps the Acknowledge button to one of SCOM's resolution states — **Acknowledged (249)** here. Change it and every operator's Acknowledge changes with it.

Comment suffix on actions — "*via SCOM Pulse*" — is appended to comments Pulse writes into SCOM, so a reader of the Operations console can tell where a comment came from.

Resolution states the menu offers publishes a shortlist. SCOM has one list of states and no way to tidy it; picking a shortlist here means the **Set state** menu offers only those, for everyone. It shortens a menu — **it is not a permission**: anyone whose role may update alerts can set any state the management group has, as SCOM allows. **Closed (255)** is always offered, and the state that **Acknowledge** maps to must be on the list. Selecting nothing offers every state.

Caution — Instant-MM URL endpoint

`GET /Api/InstantMM?ComputerName=x&Min=120&MMAction=Start|Stop` is **Off by default** and was **Off** in the reviewed deployment. It is a real action on a GET: calling it starts or stops maintenance. It is Windows-authenticated and runs as an operator account, and with no `ComputerName` it targets the calling machine. `Min=` accepts up to 1,051,200 minutes — 730 days, two years, the Operations console's own limit — and a larger value is clamped to it. Requests a browser labels as coming from another site, or as embedded in another page, are refused; requests without those labels, as scripts send them, are accepted with Windows sign-in so existing scripts keep working. Enable it only for automation that needs it — SCCM task sequences and scripts — and never open it in a browser to see what it does. Treat **200** as completed and **202** as still queued or running.

Operations Manager data source

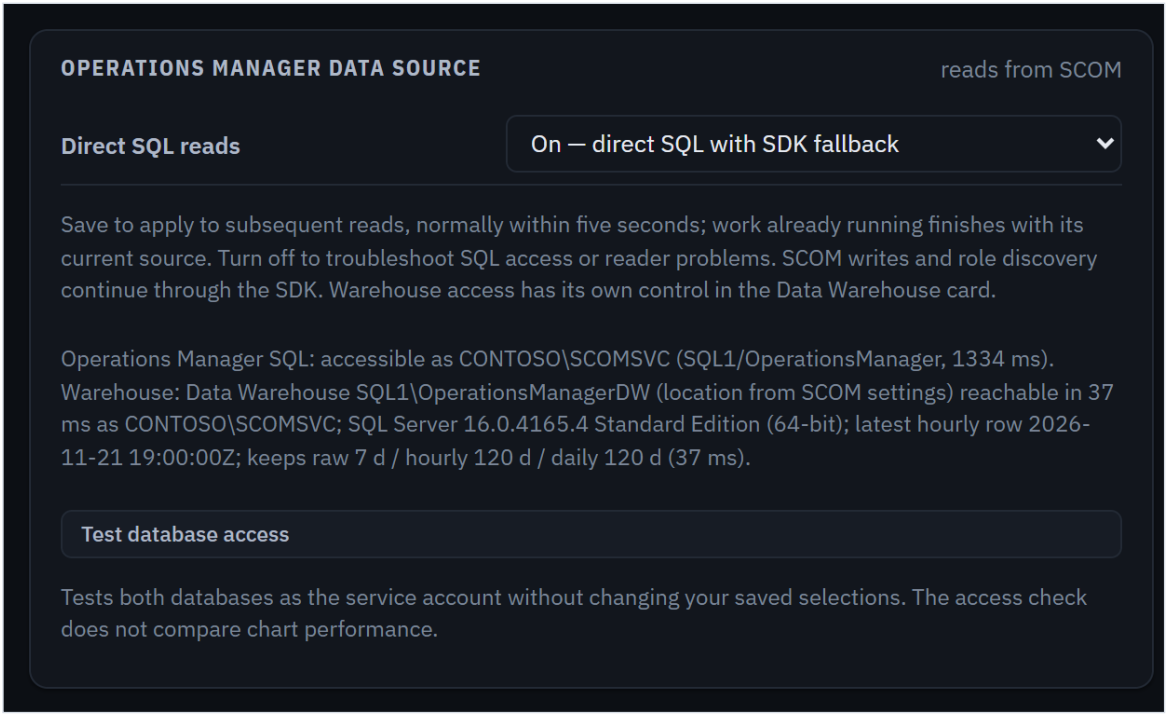


Figure 10 Settings → Sync & data → Operations Manager data source: direct SQL reads on, and the result of the last database access check.

This card decides where Pulse **reads** Operations Manager data from ("2.6 Optional: direct SQL reads from the Operations Manager database").

Direct SQL reads has two choices:

- **On — direct SQL with SDK fallback.** Pulse reads alerts, objects, health, performance samples, history, the Administration inventory and an object's effective configuration straight from the Operations Manager database. A read that fails falls back to the SDK, and SQL is tried again a minute later. The **Effective configuration** tab gains most: with direct SQL reads it covers everything an object contains, names the override behind each value, and answers in about a second rather than several seconds per object.
- **Off — SDK reads.** Everything is read through the management server, as in earlier versions.

Save, and the choice applies to subsequent reads, normally within five seconds; work already running finishes with the source it started with. No service restart is needed. **SCOM writes and role discovery always go through the SDK**, whichever you choose. The warehouse has its own control in the Data warehouse card.

Two other values can appear in the list. **Access check pending — using SDK** means Configure has asked the service to test access and it has not answered yet. **Existing service configuration** appears on an installation upgraded from a version that predates this setting; Pulse keeps whatever the service configuration file says until you choose On or Off.

Under the choice, the card shows the result of the last **database access check** — in the reviewed deployment "*Operations Manager SQL: accessible as CONTOSO\SCOMSVC (SQL1/OperationsManager, 1311 ms)*", followed by the warehouse result. **Test database access** runs the check again, as the service account, against both databases. It never changes your saved choices, and it does not measure chart performance.

> **Important — Turn it off to troubleshoot, not to fix**
>
> If pages show data that looks wrong or stale and you suspect the SQL readers, set **Direct SQL reads** to **Off**, save, and compare. If the problem goes away, the service account's SQL access or the database is where to look: run **Test database access** and read its message. Pulse is fully functional on the SDK, only slower on large estates.

Changing either database source — this card, **Enabled** and **Performance source** in the warehouse card — needs the SCOM Administrator profile — which everyone who can open **Settings** has.

Performance charts

Pinned counter refresh **300 s**; watched counter refresh **60 s**; keep samples **7 days**; default chart window **24 h**.

Pinned counter refresh is how often the service re-reads pinned counters. **Watched counter refresh** is the faster cadence for counters on a chart somebody has open (looked at in the last 15 minutes) — a

board left open for troubleshooting follows SCOM's own sampling, which for most counters is every 5 minutes, so a value below 60 s rarely shows more. A counter nobody is watching stops being refreshed, so its cache ages; opening its chart again has the service refresh it within a few seconds, and the chart shows its old line blurred under an *Updating from SCOM…* label until the new samples land. The web tier wakes the service the moment it draws a stale chart (`Perf.WakeRequestedUtc`; the service also wakes itself every 3 s), and the pass reads the most recently opened counters first and writes them before the rest, at most 50 a pass — so the chart being looked at is current within about a second and a page with hundreds of counters fills in behind it over a minute or so. Direct SQL reads make each pass one query rather than one SDK call per counter. Counters nobody has ever opened are not read at all until someone asks.

Data warehouse

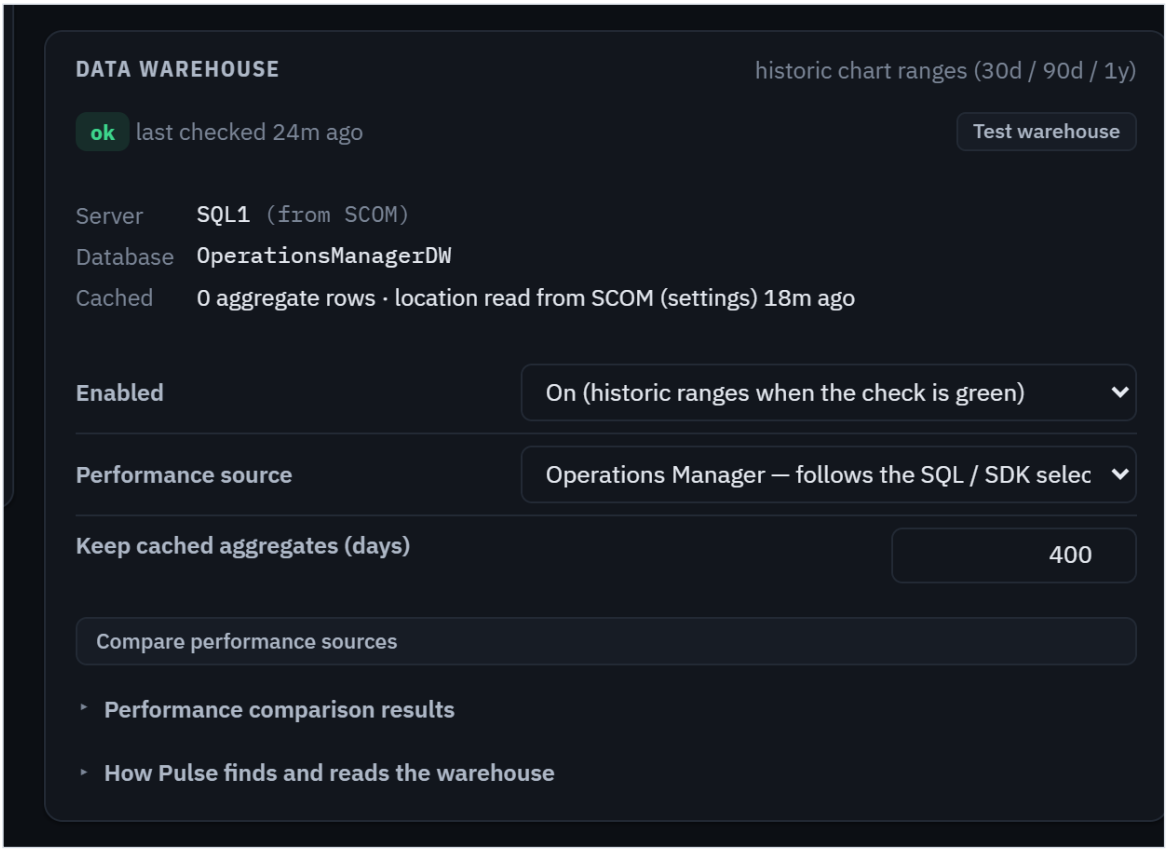


Figure 11 Settings → Sync & data → Data warehouse: the discovered warehouse, the live performance source, and **Compare performance sources**.

The **Server** and **Database** are **read from SCOM**, not typed in — the page shows them as discovered (`SQL1` / `OperationsManagerDW` here) with a **Test warehouse** button. **Enabled** turns on historic ranges (30 d / 90 d / 1 y) when the check is green. After a fresh Configure it may read **Access check pending — off until verified** until the service has tested access.

Performance source chooses where **live** samples come from:

- **Operations Manager — follows the SQL / SDK selection** (default). Live samples are read from the Operations Manager database when direct SQL reads are on, or through the SDK when they are off.
- **Data Warehouse first** — bulk reads of raw samples from `Perf.vPerfRaw`. Raw rows reach the warehouse later than the operational database, and rules that do not write to the warehouse are not there at all.

Configure never changes this choice. A quick access test cannot tell you which source is faster or fresher for your charts, and that is what **Compare performance sources** is for.

Compare performance sources reads a small sample — up to 12 counters Pulse already collects — from both the Operations Manager source and the warehouse, as the service account: three alternating reads of the last hour and of a settled 24-hour window. It reports speed, whether the values match, and how far behind the warehouse is, under **Performance comparison results**. It only ever *recommends*: it does not change the performance source or anything you have saved. It runs separately from sync, needs the SCOM Administrator profile, and can be repeated after five minutes. **Refresh result** checks on a comparison still running.

> **Note — Read a recommendation for what it is**
>
> The comparison recommends the warehouse only if its data matched in every read and it was clearly faster in both windows. Missing or late warehouse data rules it out, however fast it is. It is a sample of a dozen counters, not a benchmark of your estate, and on a small management group both sources are usually fast enough that Operations Manager — the fresher of the two — is the right answer.

Objects & groups

Health poll **60 s**. **Tracked classes** lists the classes whose instances are fully synced with health, one internal class name per line, with **add derived classes too** ticked in this deployment. Missing classes are skipped silently, so it is safe to list packs you may not have. Advanced sync intervals: full object sync **15 min**, **25** groups per cycle, watched group refresh **5 min**, other group refresh **60 min**, catalog refresh **60 min**.

> **Important — Tracking more classes costs sync time**
>
> Every tracked class is more to read on every cycle. Add classes because a dashboard or a scope needs them, not speculatively.

5.4 Site & display

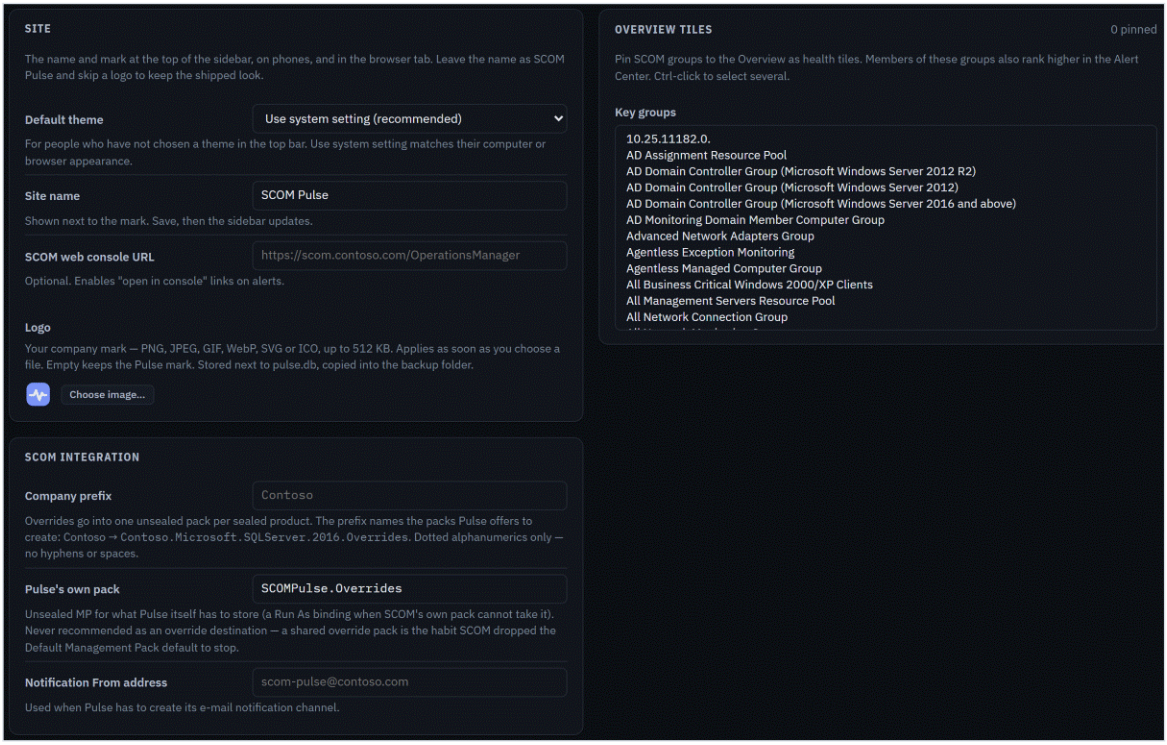


Figure 12 Settings → Site & display: branding, the SCOM console link, the company prefix that names new override packs, Pulse's own pack, and the Overview health tiles.

Site. Default theme — **Use system setting (recommended)**, **Light** or **Dark** — sets the starting appearance "*For people who have not chosen a theme in the top bar.*" This setting is optional. A personal choice in the top bar always wins, and the value is rendered before the stylesheet loads so a dark-by-default site does not open white and blink.

Site name (`SCOM Pulse`) and an optional **Logo** — PNG, JPEG, GIF, WebP, SVG or ICO up to 512 KB, applied as soon as it is chosen, stored beside `pulse.db` and copied into the backup folder. **SCOM web console URL** is optional and enables *open in console* links on alerts.

> **Note — A published link can pin its own appearance**
>
> Shared views get the same three-way control, unless the link forces a theme — in which case the control is hidden and the choice is refused, so a link published to look a particular way keeps looking that way.

SCOM integration. This is where override authoring policy lives:

- **Company prefix** — overrides go into one unsealed pack per sealed product, so uninstalling that product means removing one override pack. The prefix names the packs Pulse offers to create: `Contoso` → `Contoso.Microsoft.SQLServer.2016.Overrides`. Without a prefix nothing is offered for creation and the dialog asks for a pack — an existing one, or **New pack…**.
- **Pulse's own pack** — the unsealed pack for what Pulse itself has to store (a Run As binding when SCOM's own pack cannot take it). `SCOMPulse.Overrides` here. It is **never recommended** as an override destination and is no longer created for one; where it already exists it stays selectable under **See more — all unsealed packs**, like any unsealed pack. Overrides already in it keep working — move them to per-product packs when convenient.
- **Notification From address** — used when Pulse has to create its own e-mail notification channel.

Overview tiles. Pin SCOM groups to appear as health tiles on the Overview. Members of pinned groups also rank higher in the Alert Center, so this setting quietly affects Smart rank for everyone.

5.5 Service & storage

Figure 1 shows the service and cache block: service state, management group, and what is cached — alerts, objects, catalog, performance series — plus the database file, its size, its integrity result and the verified backup.

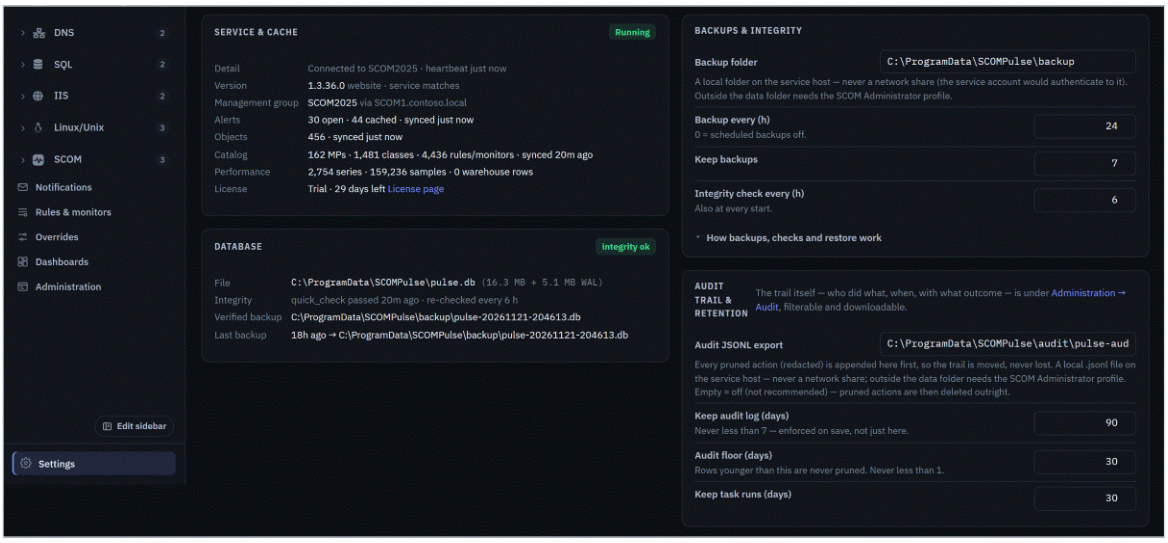


Figure 13 Settings → Service & storage: the backup and integrity schedule, and audit retention.

Backups & integrity. Backup folder `C:\ProgramData\SCOMPulse\backup`; backup every **24 h**; keep **7**; integrity check every **6 h** and at every start.

> **Caution — The backup folder must be local**
>
> The service would have to authenticate to a network share, so a UNC path is refused. A folder outside the data directory needs the SCOM Administrator profile.

Audit trail & retention. Audit JSONL export `C:\ProgramData\SCOMPulse\audit\pulse-audit.jsonl`; keep audit log **90 days**; audit floor **30 days**; keep task runs **30 days**.

Every pruned action is appended to the JSONL file, redacted, **before** it is pruned — so the trail is moved, not lost. Leaving the path empty turns that off and pruned actions are deleted outright; the page marks that "not recommended". **Keep audit log** is never less than **7** days and **Audit floor** never less than **1**; both are enforced on save.

6 Agents and management servers

The **Administration** pane is Operations Manager's, read from Pulse's cache. Its header says so: "*The console's Administration pane, read from the cache (synced 46m ago). Writes go through the service and are audited.*" **Refresh inventory** re-reads it.

The pane is read by the people SCOM lets read its Administration space: the Administrator and Read-only administrator profiles and Delegated administrator roles. The header tells the person where they stand: a Read-only administrator sees **Read-only for you**, and a Delegated administrator role sees **Partly read-only for you** — what the role does not grant is shown without its buttons. Each change needs the SCOM permission for it ("4 The access model").

6.1 The agent inventory

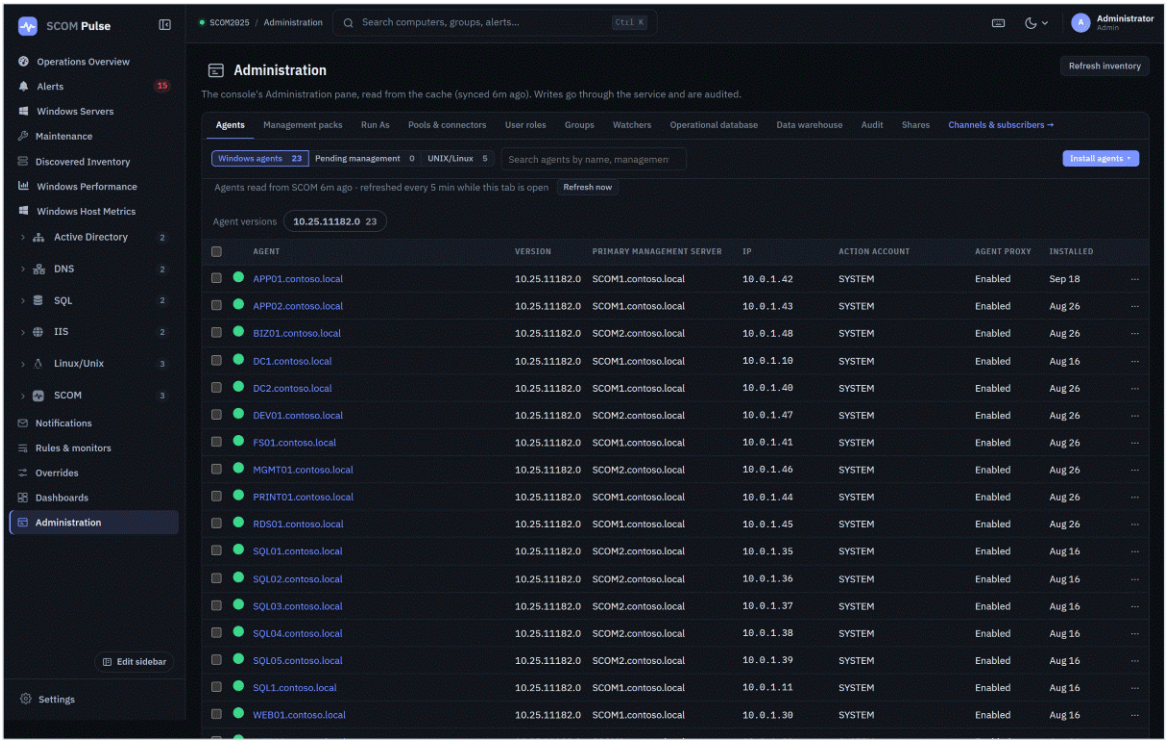


Figure 14 Administration → Agents. Health, version, primary management server, IP, action account, proxy setting and install date, above a note that the list refreshes itself every five minutes.

Columns: **HEALTH**, **AGENT**, **VERSION**, **PRIMARY MANAGEMENT SERVER**, **IP**, **ACTION ACCOUNT**, **AGENT PROXY**, **INSTALLED**, and a ⋯ menu per row. The footer states the scope — "*Showing 2 of 2*".

How current the list is. A line under the list switch says when the agents were read from SCOM — *Agents read from SCOM 2m ago* — and **Refresh now** beside it reads them again at once, with the pending list and the UNIX/Linux computers. The list is also read from SCOM when you open the tab and what SCOM Pulse holds is more than a minute old, and every five minutes while the tab stays open, so reloading the page shows the agents as SCOM has them now. An agent that has just joined the management group — approved from Pending management, or installed from here — is listed before SCOM knows its health, version or address; SCOM Pulse reads it again every half minute until SCOM does, and the list follows by itself. **Refresh inventory**, at the top of the page, reads the whole Administration pane, and moves the *synced … ago* under the title.

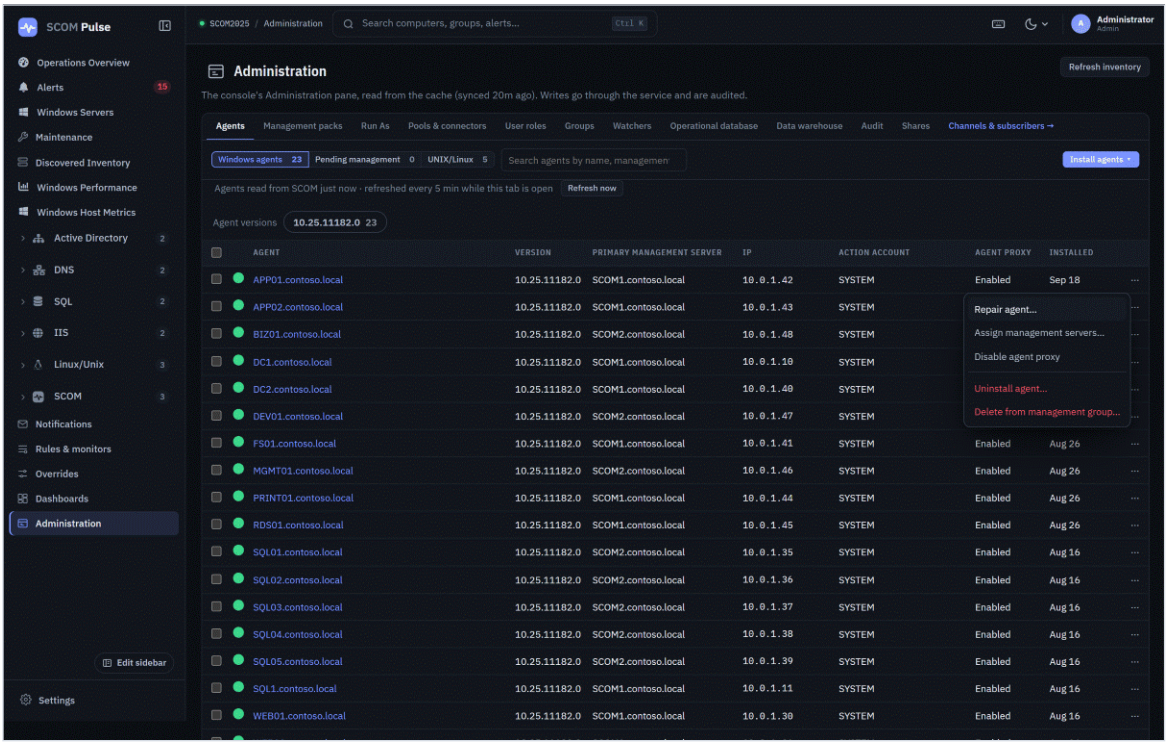


Figure 15 The per-agent ⋯ menu, holding the lifecycle actions available for one agent.

Per-agent actions cover repair, management-server assignment, the proxy setting and uninstall.

Windows agents, Pending management and UNIX/Linux. The tab holds the lists the Operations console keeps apart under Device Management, behind a switch at the left of the toolbar: **Windows agents** — the inventory above, the console's *Agent Managed* — **Pending management**, the agents

waiting for a decision, and **UNIX/Linux**, described below. Each button carries its count, and the **Agents** tab itself shows *N pending* while anything is waiting. A pending entry shows its **TYPE**, **COMPUTER**, **MANAGEMENT SERVER** and when SCOM **LAST MODIFIED** it, with **Approve** for a manual installation, **Update agent...** for an update SCOM has queued, and **Reject** to clear the entry; click a column heading to sort.

Install agents . Every installation starts from this one button at the top right, on all three lists: **Install Windows agents...** (6.3), **Install UNIX/Linux agents...** (below) and **Find Windows computers...**, which lists the computers in a domain and installs nothing (6.2). An entry your SCOM role does not allow, or that this server cannot run — UNIX/Linux work needs the Operations console on the SCOM Pulse server — stays in the menu, greyed out, with the reason under it.

Updating many agents at once. After an update rollup, SCOM queues an *UpdateAgent* entry for every agent it could not update on its own, and an *UpdateFailed* entry where it tried and failed. Each of those entries has a tick box; the box in the header ticks every one that is showing, so a search — a management server's name, a site's naming prefix — narrows what it ticks. **Update selected...** then runs the console's Update Agents task for all of them as **one job**: each agent is updated through its own management server, with its own progress and result, using the management server's action account or an account you supply. SCOM clears an entry itself when that agent's update succeeds; one whose update fails stays in Pending management, and the job keeps the management server's own error text for it. A row's result is the management server's own account of the update, not the status of the task that carried it: SCOM marks the task succeeded even when the update was refused on the computer, and such a row reads *Failed* here, with the reason — *Access denied*, for one. A row that completed says which updates were installed, or that the management server reported none. Up to 200 agents per job; five or more ask for a confirmation first. What is ticked is always among the entries showing: changing the search unticks what it hides, so nothing is updated that is not on screen. If an entry has gone, or its computer is not in the agent inventory, when you start, the whole request is refused and says which — refresh and select again.

A job of many agents is handed to SCOM one management server at a time, and what has been handed over is SCOM's to finish. **Cancel** stops the agents not yet sent and asks SCOM to stop the rest — SCOM honours that only for an update that has not started. If one management server refuses its agents, those agents fail with its reason and the others go on. If the SCOM Pulse service restarts, or loses its connection to SCOM, part-way, the updates already handed over keep running and the job follows them again when it can (for up to two hours); the agents it had not reached are marked as not sent, to be updated again. An agent whose own management server cannot be found is not sent through another one.

Approving many agents at once. An entry waiting for approval — a manually installed agent, type *ManualApproval* — has a tick box too, for anyone whose SCOM role holds the Agent Pending Actions permission. Tick the ones you want, or search and use the box in the header to tick every one that is showing, and choose **Approve selected...**. The dialog names every computer; **Approve** then approves them together, as one entry in the audit trail with a result for each computer. One that SCOM does not approve stays in Pending management with the reason, and the others are approved all the same; the

notice that reports the result has a **details** link listing every computer. Up to 200 agents at a time; five or more ask for a confirmation first. When manual installations and agent updates are ticked together, the bar offers **Approve N…** and **Update N…**, and each acts only on its own kind. If an entry has gone, or is not waiting for approval, when you start, the whole request is refused and says which — refresh and select again. **Reject** stays one entry at a time, and a failed installation or repair has no tick box: it is neither approved nor updated.

Whether manually installed agents arrive here at all. Above the list, Pending management says what the management group does with a new agent that was installed on a computer by hand: it *rejects* them, they *wait here* until someone approves or rejects them, or they are *approved automatically*. Where they are rejected, their requests never reach this list however many are asking, so this line is the first thing to read when an agent you expect is missing. A management server or gateway that was given its own setting in the Operations console, and answers differently from the management group, is named beside it.

Anyone whose SCOM role holds the Global Settings permission has **Change…** on that line, with the three choices. It is the management group's setting, the same one as **Administration** → **Settings** → **Security** in the Operations console. A management server or gateway with its own setting keeps it whatever is chosen here — the dialog names each one — and that is still changed on the server in the console. The change is confirmed, written to the audit trail, and shown as SCOM reports it afterwards; if someone else changed the setting while the dialog was open, nothing is changed and you are told. An agent that was rejected before a change can appear after it; allow about an hour. With *Approve automatically* each new manually installed agent is added with nobody reviewing it. An agent still has to authenticate to the management server, as every agent does, but choose this only where adding them unreviewed is acceptable.

Uninstalling an agent, and deleting one from the management group. A Windows agent's row menu (□), and the bar that appears when rows are ticked, offer two ways to stop managing a computer:

- **Uninstall agent…** removes the agent from the computer through its management server and then stops managing it. Use it for a computer that is running.
- **Delete from management group…** — the Operations console's *Delete* — stops managing the computer and does nothing on it: the computer leaves the Windows agents list and is no longer monitored, its history stays until groomed, and its agent stays installed. Use it for a computer that no longer exists or cannot be reached, where an uninstall can only fail. No account is asked for.

Deleting is confirmed every time, for one computer as for many, and the confirmation names each computer. An agent left running on a deleted computer keeps trying to report, and what SCOM does with it then follows the management group's setting for manually installed agents — so for a running computer, uninstall. When a selected computer is shown as available in the latest inventory, the confirmation says so. A management server's or gateway's own agent is refused, and a selection that contains one is refused whole. An agent that agentless-managed computers or network devices are

monitored through is not deleted until they are moved to another proxy agent or no longer monitored; its row says how many there are.

A delete runs as a job with a result for each computer, and a computer counts as *deleted* only when SCOM, asked again, no longer lists it; one that had already gone is reported as *Already gone*. A computer whose delete was sent and which SCOM did not then report as gone is *Not confirmed* — whether SCOM answered with an error, was slow, or could not be asked — with the instruction to refresh the inventory later: once the computer is no longer listed it was deleted, and if it stays listed it can be deleted again. A delete is sent at most once for a computer in a job: if the job is cancelled, or the SCOM Pulse service restarts or loses its connection to SCOM part-way, nothing is sent again, and the computers not yet reached say that nothing was done for them. To manage a deleted computer again, install or repair its agent. When an uninstall fails because the computer did not answer, its row points here.

Both are granted by the **Uninstall Agents** permission of a SCOM user role ("4 The access model"); a role limited to groups uninstalls and deletes only the agents inside them. Up to 200 agents per job.

Agent versions. Above the **Windows agents** list a line counts the agents at each version, newest first — *10.22.10118.0 58 · 10.19.10014.0 4*. Click a version to list those agents; click it again to clear. The counts are of your whole list, whatever is being searched for, so the line shows at a glance how far an update rollup has reached.

The one search box acts on whichever list is showing, and every count follows it — "*0 of 62*" beside "*3 of 45*" — so a search for new servers that only matches computers waiting for approval says so from either list, with a **Show them** button to change over. `/Pulse/Admin?tab=agents&view=pending` opens the pending list directly, and `&view=unix` the UNIX/Linux one. Someone whose SCOM role is limited to groups is not shown pending entries ("4 The access model"), so their switch has no **Pending management**.

UNIX/Linux computers

UNIX/Linux lists the UNIX and Linux computers the management group manages — the console's *UNIX/Linux Computers* — with **COMPUTER**, **IP**, **OPERATING SYSTEM**, **AGENT VERSION**, **RESOURCE POOL** and **SSH PORT**, and one column the console does not have: **HEARTBEAT**.

Heartbeat, with the reason. SCOM keeps five checks on every UNIX/Linux computer: the heartbeat itself, the agent's certificate, the name the management server resolves, WinRM authentication on the management server, and the Run As account. The console shows them one computer at a time, in Health Explorer. This column reads them for every computer, so the list says *Healthy*, *Not monitored yet* (a computer added in the last few minutes) or *Failing* with what SCOM names as the cause: **name resolution**, **WinRM authentication on the management server**, **certificate**, **Run As account**, or **no heartbeat** when the heartbeat check is failing and none of the others names a cause. Failing computers are listed first, with how long they have been failing; **Heartbeat failing** shows only those, with their number. Pointing at a cell explains the check and where to look.

> **Note — "No heartbeat" does not mean the computer is down**
>
> When only the heartbeat check is red, SCOM has not said why. A computer that is switched off reads this way, and so does one whose agent is stopped, one behind a firewall, and — as seen in testing — one that refuses the Run As account. **Diagnose** tells them apart.

Diagnose. Tick one or more computers and choose **Diagnose...**. Pulse runs the Operations console's own discovery against each one, through the resource pool that manages it, and reports per computer **Nothing wrong** or **Problem found** with three things: what is wrong in a sentence, SCOM's own message, and what to check next — for example "*The agent answered but refused the account*", "*The agent's certificate names a different host*", or "*The management server could not resolve the name*". This is the check an administrator otherwise re-runs the console's Discovery Wizard for.

Diagnose uses the **Run As accounts** SCOM already holds for each computer; nothing is typed and nothing secret is stored. It installs, signs and changes nothing. A computer whose agent does not answer with a signed certificate is also tried over SSH, exactly as the Discovery Wizard does, unless you tick **Check the agent only**. Up to 200 computers can be checked in one run, the run continues if you close the dialog, and it is recorded in the audit trail under your name. It is offered to members of the **Operations Manager Administrators** role, the people the console offers its Discovery Wizard to.

> **Important — UNIX/Linux agent work needs the Operations console on the SCOM Pulse server**
>
> Diagnose, discovery, Manage, Sign, the update check, Upgrade, Change resource pool, Remove and Uninstall run Microsoft's own UNIX/Linux discovery and deployment, which are installed with the **Operations console** and not with SCOM Pulse. On a Pulse server without the console the list and the heartbeat column work as described and those actions are unavailable, with a line saying why. Install the console on the Pulse server — at the same update rollup as the management group — and it becomes available within about ten minutes, or at once when the SCOM Pulse service restarts. A console at a different update level from the management group's UNIX/Linux management packs is reported in the result, because some answers cannot then be read.

Adding UNIX/Linux computers

Install agents □ → **Install UNIX/Linux agents...** is the console's Discovery Wizard in two steps with a review between them.

1. Discover. List the computers — names or IPv4 addresses, one per line, up to 200 — and choose the **resource pool** that will manage them (the pools already managing UNIX/Linux computers are listed first), the **SSH port**, and what to look at: **All computers**, or **Only computers that already have the agent installed**. Discovery installs and changes nothing. For a computer with no working agent, *All computers*

signs in over SSH and runs SCOM's detection script to see what the computer is; the second choice sends nothing over SSH.

2. Review. The result lists every computer with one of three outcomes:

Outcome	Meaning
Can be managed	With what managing it will do: *install the agent, sign its certificate and add the computer*; *the agent is installed — sign its certificate and add the computer*; or *add the computer*.
Already managed	The computer is in the management group — also when it was typed as an address or a short name and SCOM holds it under its full name. Nothing is done to it.
Cannot be managed	With the diagnosis Diagnose gives: what is wrong, SCOM's own message, and what to check.

3. Manage. Tick the computers you want and choose **Manage selected...**. The dialog repeats, per computer, what will be done. Each computer is discovered again first, and one that is no longer as you reviewed it — an agent installed or removed since, or the computer managed by someone else in the meantime — is left alone and reported. Progress is shown per computer as SCOM reports it: copying the agent, installing, signing the certificate, adding the computer. A computer that was added appears in the UNIX/Linux list once SCOM has distributed its monitoring, usually within a few minutes. Five or more computers ask for a confirmation first.

> **Note — Cancelling does not undo**
>
> A step SCOM has already started on a computer runs to its end. A cancelled or interrupted job can therefore leave an agent installed on a computer that is not managed yet. Run discovery again: it shows that state, and Manage finishes from it.

Credentials. Each of these jobs — discovery, Manage, and Sign certificate again below — runs with one of two kinds of credentials, as in the console:

- **The Run As accounts SCOM holds** (the default). Nothing is typed and nothing secret passes through SCOM Pulse. The accounts are the ones your *UNIX/Linux Agent Maintenance Account* and *UNIX/Linux Action Account* profiles resolve for the computer.
- **Other credentials, typed now.** A user name with a **password** or an **SSH key**, and how the account becomes root: it *is privileged*, it *uses sudo*, or it *uses su* (with the root password). Typed credentials need SCOM Pulse to be open over **HTTPS**; over HTTP the choice is not offered. They are sent once to the SCOM Pulse service, held in memory only, for at most ten minutes, handed to SCOM for the one job, and never written to the Pulse database, the audit trail or a log. The audit trail records that typed

credentials were used and for which user — never the credentials. Manage after a discovery that used typed credentials uses the same ones without asking again while the page stays open; after a reload it asks for them again.

Important — SSH keys: check the host accepts the key before relying on one

SCOM reads a **PuTTY private key file (.ppk), format 2, RSA** — an OpenSSH key, a PuTTY format 3 key or an Ed25519 key is refused with the reason and how to convert it. SCOM signs in with the older **ssh-rsa (SHA-1)** signature, which current releases refuse unless the host is set to accept it. On **Ubuntu 22.04** and its relatives that is `PubkeyAcceptedAlgorithms +ssh-rsa` in the SSH service's configuration. On **RHEL 9** and its relatives (Rocky, Alma, Oracle Linux 9) that setting alone is not enough: SHA-1 has to be allowed in the system's crypto policy (`update-crypto-policies --set DEFAULT:SHA1`, then restart the SSH service) — a decision for whoever owns the host's security baseline, and the reason a password with sudo is often the simpler choice there. A refused key is reported by SCOM as "*Too many authentication failures*", the same words a wrong password gets, or as a certificate that could not be read over SSH. A **passphrase** that does not open the key is said at once, before any job is started — where that can be known for certain: a passphrase or a key comment with characters outside plain ASCII is not checked, and SCOM is left to try the key. A password does not have this limit. With a key you can also name an **account for checking the agent** : the agent is checked with a user name and password, and without one SCOM checks with its own Run As account for the computer.

Note — Every management server in the resource pool takes part

SCOM runs each step of a discovery or an install on whichever management server of the pool is free. A computer's name therefore has to resolve — name to address and address to name — on **every** member of the pool, and each member needs the pool's certificates and the Run As accounts distributed to it. A name that resolves on one member only shows up as "*could not resolve*" or "*could not find by name*" on some runs and not others.

Sign certificate again. Tick managed computers and choose **Sign certificate again...** when Diagnose says the agent's certificate is not signed by this resource pool — after an agent was reinstalled, or its certificate regenerated. The pool signs the certificate, and Pulse then checks that the agent answers with it. Nothing else about the computer changes: not its resource pool, not its agent. A certificate that names a different host is not cured by signing; Diagnose says which it is.

Discover and install in one step. For a list you already trust — the servers of a rollout you planned, a batch re-added after a rebuild — the Discover dialog offers, under **After discovery**, *Manage every computer that can be managed, without a review*. The job then discovers each computer and goes straight on: the agent is installed where there is none, its certificate is signed and the computer is added to the pool you chose. A computer SCOM already manages, and one that cannot be managed as discovery finds it, is reported and left alone, exactly as in the two-step path. Nothing is shown to you before it is done, so

this choice always asks for a confirmation, whatever the number of computers, and it looks at every computer in full — *Only computers that already have the agent installed* does not apply to it. It needs the right to manage as well as the right to discover.

Discovery, Manage and Sign are offered to members of the **Operations Manager Administrators** role and need the Operations console on the SCOM Pulse server, as Diagnose does.

Upgrading UNIX/Linux agents

After an update rollup, the management group holds newer UNIX/Linux agents once the updated management packs are imported. The UNIX/Linux list shows which computers that applies to, and upgrades them.

Check for updates. Beside the filters the list says when it last asked — *Agent updates checked 5m ago*, or *not checked yet* — with **Check now**. The check asks each resource pool which agent version it holds for each computer's platform; nothing is changed on any computer, and it takes a few seconds. Run it after importing new UNIX/Linux management packs. A computer whose agent is older than the one SCOM holds then reads *1.9.1-0 → 1.9.2-0* in the **AGENT VERSION** column, and **Update available** filters the list to those computers and counts them. A computer SCOM holds no agent for reads *no agent held*, and *N not compared* beside the filters lists the computers that could not be compared at all — they are not known to be current. If a resource pool does not answer the check, what was known for its computers is kept and the result says so. One check runs at a time; asking again while one is running waits for that one.

Upgrade agent… Tick computers and choose **Upgrade agent…**. The dialog lists each computer that shows an update, with the version it has and the version it will get, and offers to include the ticked computers that show none. What the list shows comes from SCOM's record of each agent's version, which lags behind an agent changed outside SCOM; the job does not rely on it. It **asks each agent which version it is** before doing anything, and upgrades only the ones older than the agent SCOM holds: one that is current, or ahead, is reported as *Already current* and left alone, and one that cannot be asked — its certificate is not accepted, the account is refused — is not upgraded, with the reason. After an upgrade the agent is asked again, and the computer counts as *Upgraded* only when it answers with the new version. That answer is the whole of it: if SCOM's own last check fails — on a management server that does not accept the agent's certificate, say — and the agent nevertheless answers with the new version, the computer is *Upgraded*; if it cannot be asked, the row says the upgrade ran and that its result is not known, with why. Progress is shown per computer — copying the new agent, upgrading, checking. The agent restarts during its upgrade, so monitoring of that computer pauses for the moment that takes. Up to 200 computers per job; five or more ask for a confirmation first. When the job ends, the list and its update counts are read again.

Upgrade takes the same two kinds of credentials as the jobs above. With Run As accounts it uses the account in the *UNIX/Linux Agent Maintenance Account* profile, which has to be privileged on the computer.

Important — After importing new UNIX/Linux management packs, give the agent a day — or check the kit folder

A new management pack brings a new agent, and **Check now** shows the update within minutes. But SCOM puts the new agent into the folder its management servers share (`Server\AgentManagement\UnixAgents\DownloadedKits` on each management server) only **once a day**, or when a management server's health service restarts. Until every management server in the resource pool has it there, an install or an upgrade can fail with "*The agent could not be copied*" — one server names the file by a path that exists only on itself, and another does the copying. If you need the upgrade the same day, look in that folder on each management server of the pool first.

Note — An agent older than the computer's OpenSSL

"*The agent SCOM holds for this platform does not run on this computer: it does not support the computer's OpenSSL (3.x)*" means the management group's UNIX/Linux management packs are older than the operating system — the agent of the original SCOM 2022 packs knows OpenSSL 1.0 and 1.1 only, and RHEL 9 or Ubuntu 22.04 have OpenSSL 3. Nothing was installed and nothing is wrong with the computer or the account. Import the packs from a current update rollup, then discover again.

Note — If SCOM holds no agent for a computer

"*SCOM has no agent to upgrade this computer with*" means the management pack for that computer's platform is not imported at the new level, or its agent kit has not arrived on the management servers of the resource pool. Import the management packs from the update rollup, wait for the kits to be distributed, and use **Check now**.

Checking for updates and upgrading are offered to members of the **Operations Manager Administrators** role and need the Operations console on the SCOM Pulse server.

Moving, removing and uninstalling UNIX/Linux computers

More □, in the bar that appears when computers are ticked, holds the rest of a managed computer's life. Each runs as a job with per-computer progress and result, up to 200 computers at a time, and is recorded in the audit trail under your name. Each computer's result is recorded as soon as its own step is over, so a job that is cancelled part-way has already said what happened to the computers it reached. Cancelling one of these jobs needs the same right as starting it.

Action	What it does	On the computer	Credentials
Change resource pool…	Moves the computer to another resource pool; that pool's management servers take over its monitoring.	Nothing	None
Remove from management group…	SCOM stops managing the computer. It leaves the UNIX/Linux list and is no longer monitored.	Nothing — the agent stays installed and running	None
Uninstall agent…	Uninstalls the agent, and SCOM stops managing the computer.	The agent is removed	Run As accounts, or typed

A computer counts as *Moved*, *Removed* or *Uninstalled* only when SCOM's own list, read again after the step, shows the change. When SCOM accepts a step and its list does not show it, the row says so instead of reporting success. When the job ends, the UNIX/Linux list is read again.

Change resource pool. Choose the pool to move to; the pools already managing UNIX/Linux computers are listed first. A computer the chosen pool already manages is reported as *Already there* and left alone. Five or more computers ask for a confirmation first.

> **Important — Check the new pool before you move computers to it**
>
> Every management server in the new pool has to resolve the computer's name, reach it on port 1270 and accept its certificate — the servers of a pool accept each other's signatures only once their certificates have been exchanged (`scxcertconfig -export` / `-import`) — and the UNIX/Linux Run As accounts have to be distributed to the pool. If one of these is missing the computer stops heartbeating after the move. **Diagnose** then says which it is, and moving the computer back restores it.

Remove from management group. Use it for a computer that is gone or cannot be reached any more, or one you want to stop monitoring while leaving its agent in place. Nothing is sent to the computer, so no credentials are asked for. History stays until groomed. To manage the computer again later, use **Install agents** → **Install UNIX/Linux agents…**: the agent that is still there is found, and the computer is added without a new install. Removing always asks for a confirmation.

Uninstall agent. Signs in to each computer, uninstalls the agent and then stops managing the computer. It takes the same two kinds of credentials as the jobs above; with Run As accounts it uses the account in the *UNIX/Linux Agent Maintenance Account* profile, which has to be privileged on the computer. A computer whose uninstall fails is normally still managed, and its row says why — a refused sign-in, a name that does not resolve, an SSH port that cannot be reached — and whether SCOM still lists the computer: an uninstall takes the agent off and then stops managing the computer, and it can fail between the two. If

the agent is gone and the computer is still listed, or the computer itself is gone, use *Remove from management group*. Uninstalling always asks for a confirmation.

> **Note — Cancelling does not undo**
>
> An uninstall SCOM has started on a computer runs to its end. A computer added or moved in the last few minutes may be refused with "*SCOM could not run this through the resource pool that manages it*": SCOM has not placed it on a management server yet. Try again shortly.

Change resource pool, Remove and Uninstall are offered to members of the **Operations Manager Administrators** role and need the Operations console on the SCOM Pulse server.

The list keeps itself current. While the Agents tab is open and the browser window is in front, Pulse re-reads the agent inventory from SCOM every five minutes and says so under the list — *Auto-refresh every 5 min while this tab is visible*. A refresh keeps your search, sort and selection, and reloads only the pages you have already scrolled through, and it waits while a dialog is open, so nothing moves under a wizard you are part way through. It stops when you move to another tab or another window, so nothing is read on your behalf while you are not looking. **Refresh inventory** is still there when you want it now: it re-reads in place, never reloading the page out from under you.

> **Caution — Selection is bounded to the rows that are loaded**
>
> As with the alert grid, a header checkbox selects **loaded** rows, not every agent in the management group. Before a bulk uninstall or reassignment, compare the selection count with the footer count and with what you intend.

> **Important — What agent lifecycle Pulse does not do**
>
> Windows agent **upgrade** and UR patching, gateway deployment and management-server deployment are outside these workflows. Management-server and gateway agents are protected from uninstall here — as they should be, since uninstalling one is not an agent operation.

6.2 Discovering computers

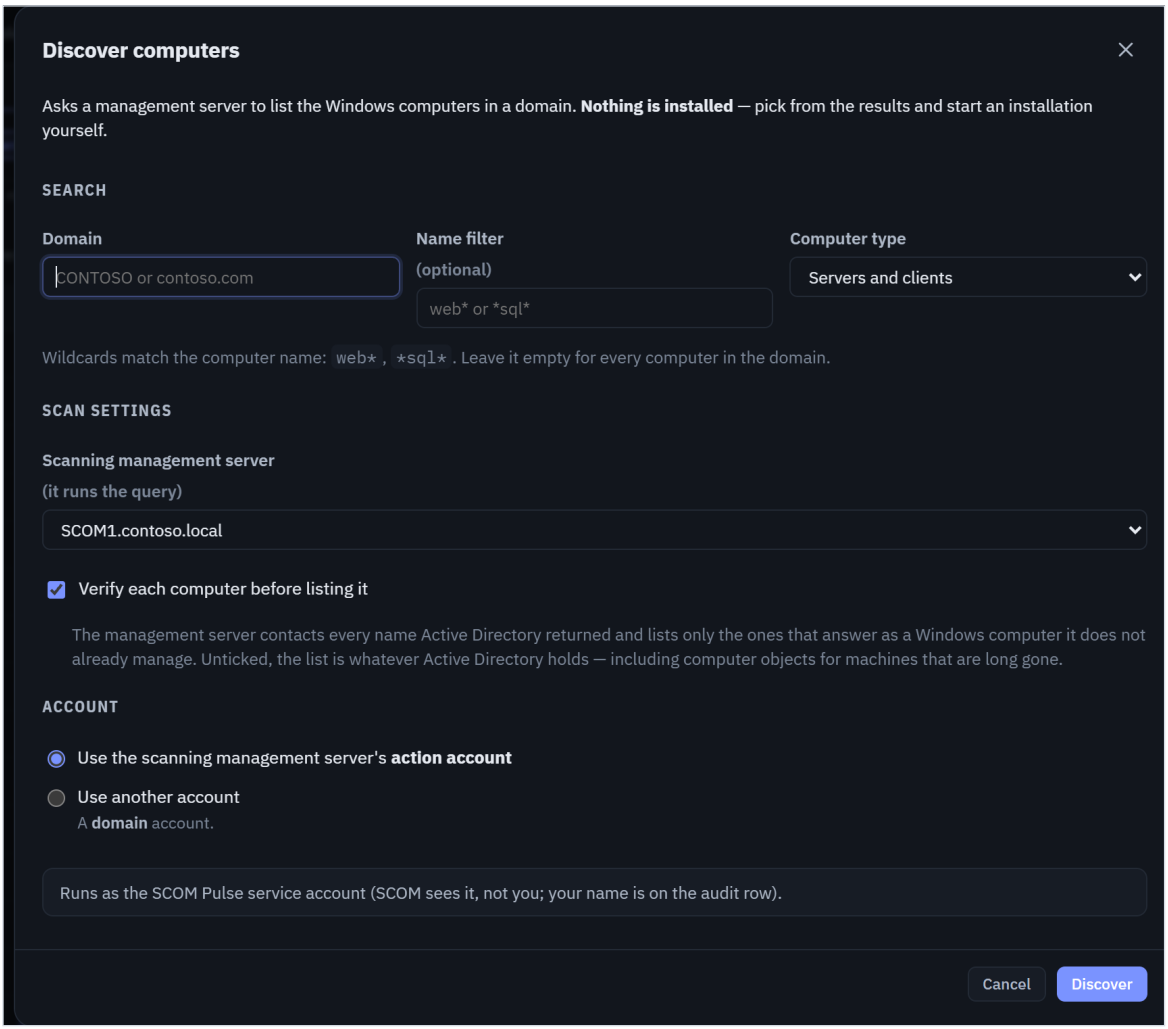


Figure 16 The **Discover computers** dialog, unsaved. Nothing is installed by a discovery.

Purpose — ask a management server to list the Windows computers in a domain. Open it with **Install agents** → **Find Windows computers…** on the Agents tab. "*Nothing is installed — pick from the results and start an installation yourself.*"

1. **SEARCH** — set **Domain**, an optional **Name filter** (wildcards match the computer name: `web*`, `*sql*`; empty means every computer in the domain), and **Computer type**: **Servers and clients**, **Servers only** or **Clients only**.
2. **SCAN SETTINGS** — choose the **Scanning management server**, which runs the query.
3. Decide **Verify each computer before listing it**. Ticked, the management server contacts every name Active Directory returned and lists only the ones that answer as a Windows computer it does not already manage. Unticked, you get whatever AD holds — including computer objects for machines that are long gone.
4. **ACCOUNT** — use the scanning management server's **action account**, or supply another domain account.
5. **Discover**.

> **Important — Who SCOM thinks is asking**
>
> *"Runs as the SCOM Pulse service account (SCOM sees it, not you; your name is on the audit row)."* That sentence, printed on the dialog, is the whole identity model in one line.

6.3 Installing agents

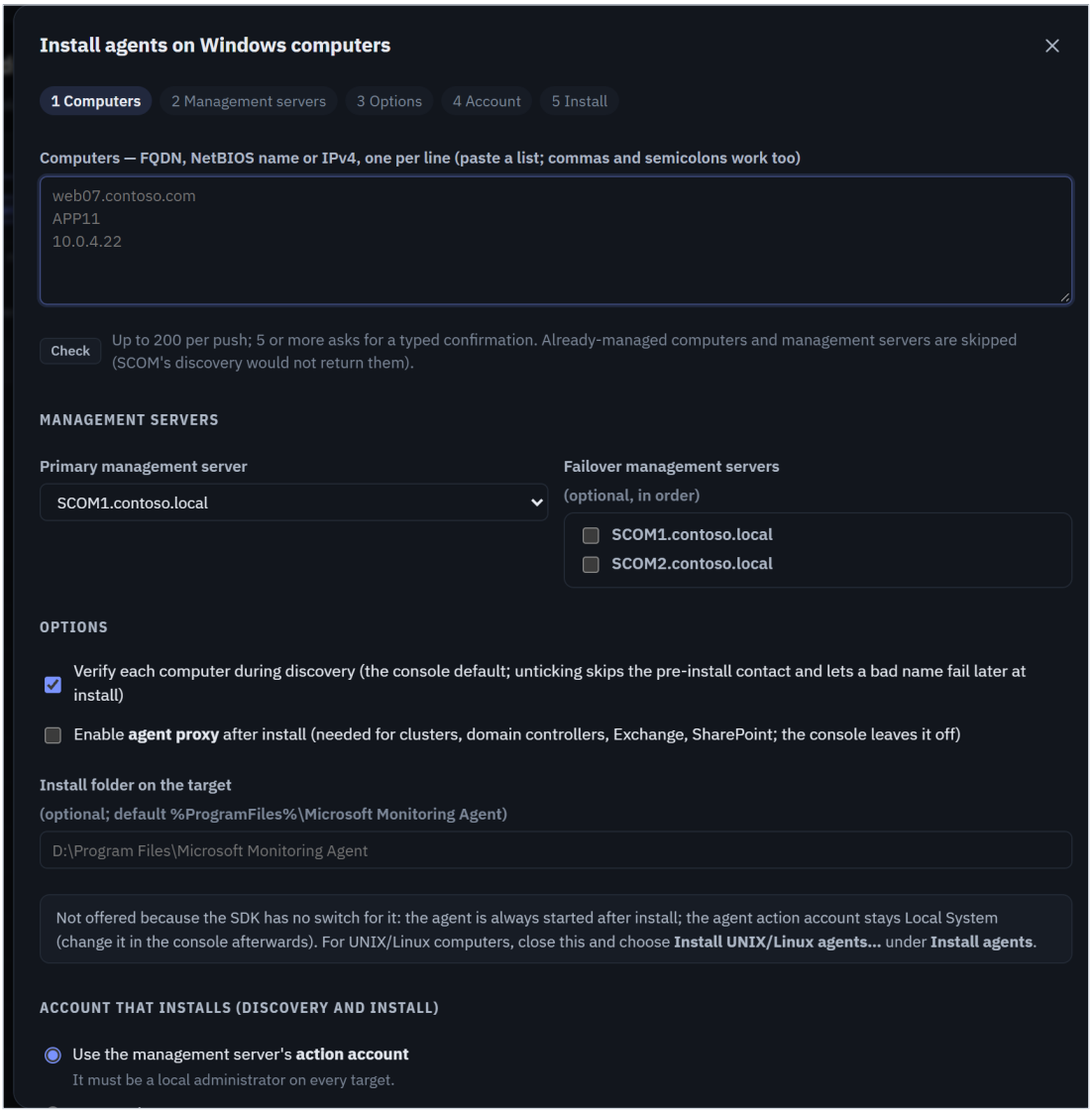


Figure 17 The **Install agents on Windows computers** wizard, unsaved, showing all five steps.

> **Required access — The SCOM Administrator profile, or a Delegated administrator role with Deploy Agents**
>
> Installing an agent changes a remote server.

Open the wizard with **Install agents** □ → **Install Windows agents…** on the Agents tab.

1. **Computers** — FQDN, NetBIOS name or IPv4, one per line; commas and semicolons work too. Use **Check** before continuing. Limits: up to **200 per push**, and **5 or more asks for a typed confirmation**. Already-managed computers and management servers are skipped, because SCOM's discovery would not return them.
2. **Management servers** — a **Primary management server** and optional **Failover management servers** in order.
3. **Options** — **Verify each computer during discovery** (the console default; unticking skips the pre-install contact and lets a bad name fail later at install); **Enable agent proxy after install**, needed for clusters, domain controllers, Exchange and SharePoint, which the console leaves off; and an optional **Install folder on the target** (default `%ProgramFiles%\Microsoft Monitoring Agent`).
4. **Account** — the management server's **action account**, which must be a local administrator on every target, or another domain account that is.
5. **Install**.

The dialog is explicit about what it cannot offer, because the SDK has no switch for it: the agent is always started after install, the agent action account stays Local System (change it in the console afterwards), and UNIX/Linux computers are a console job.

> **Caution — Push credentials, and a different credential entirely**
>
> When you supply an account for a push, it is handed to the service through a local named pipe and held in memory — it is **not** stored in the action payload. That is specific to agent pushes. Setup has a **separate**, machine-encrypted credential mechanism for the service account during upgrades, with a documented opt-out. Do not conflate the two, and never expose either.

Expected result — a job appears with progress and diagnostics, and per-target results. Confirm from the agent inventory afterwards, not from the job alone.

> **Documentation gap — Agent installation was not executed**
>
> No agent was installed, repaired, reassigned or uninstalled while writing this guide. The steps and limits above come from the dialog's own text and the product reference; job progress, diagnostics and result screens were not captured. See `Documentation-Verification.md`.

6.4 Management servers

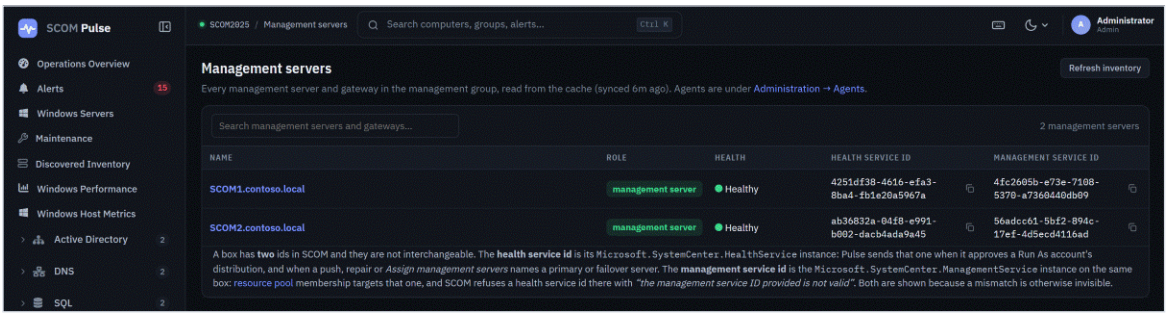


Figure 18 /Pulse/ManagementServers — a separate page, not an Administration tab.

Reached from **Administration** → **Pools & connectors**, this page lists every management server and gateway with its **ROLE**, **HEALTH**, **HEALTH SERVICE ID** and **MANAGEMENT SERVICE ID**.

> **Important — Two ids, not interchangeable**
>
> A box has two ids in SCOM. The **health service id** is its `Microsoft.SystemCenter.HealthService` instance: Pulse sends that one when approving a Run As account's distribution, and when a push, repair or **Assign management servers** names a primary or failover server. The **management service id** is the `Microsoft.SystemCenter.ManagementService` instance on the same box: resource pool membership targets that one, and SCOM refuses a health service id there with "*the management service ID provided is not valid*". Both are shown because a mismatch is otherwise invisible.

7 Management packs and groups

7.1 The management pack list

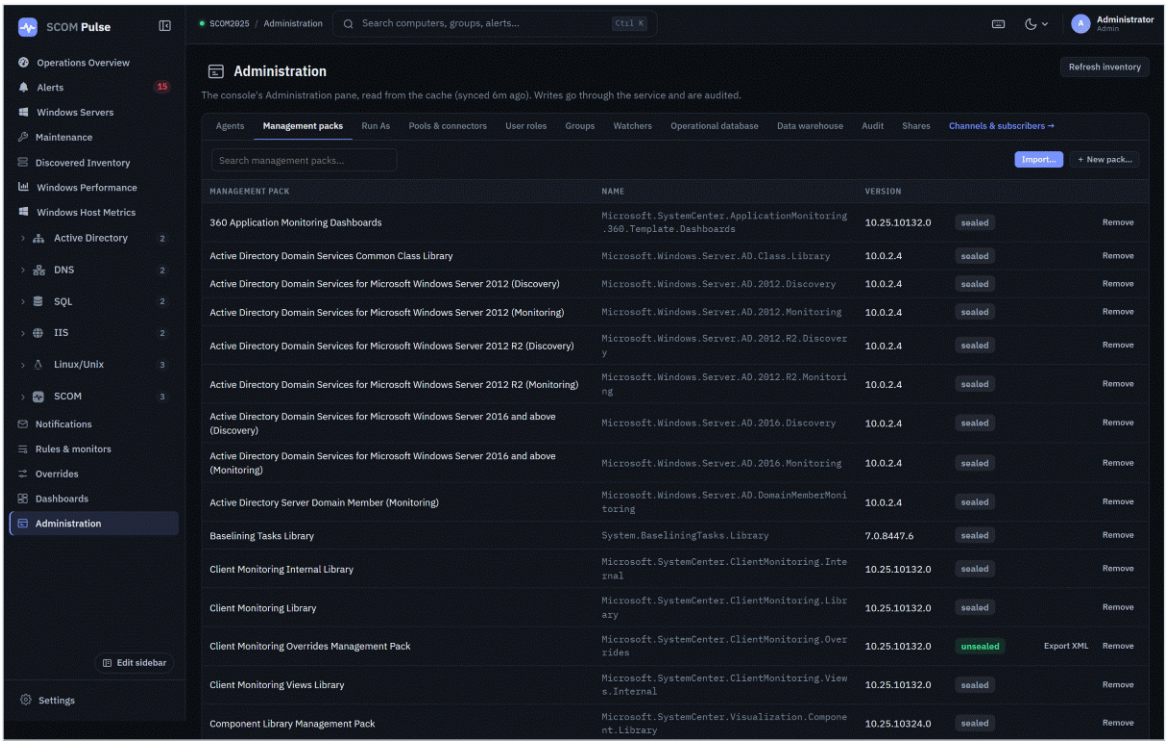


Figure 19 Administration → Management packs, with **Import...** and □ **New pack...** above the list.

Each row gives the display name, the internal name, the version, whether the pack is **sealed** or **unsealed**, and the actions available. Unsealed packs offer **Export XML**; every pack offers **Remove**.

Import... accepts `.xml`, `.mp` and `.mpb`. It opens the operating system's file chooser rather than an in-page dialog.

> **Important — Export is unsealed-only**
>
> Pulse exports **unsealed** packs as XML. Exporting a sealed pack is not a Pulse operation. For a sealed pack, keep the original vendor file.

Caution — Removing a pack removes what it defines

A pack carries classes, discoveries, rules, monitors, views and overrides. Removing it removes all of them, along with the monitoring they provide, and SCOM will refuse where another pack depends on it. Resolve dependencies deliberately — do not force your way through by removing dependants you have not thought about.

7.2 Creating an unsealed pack

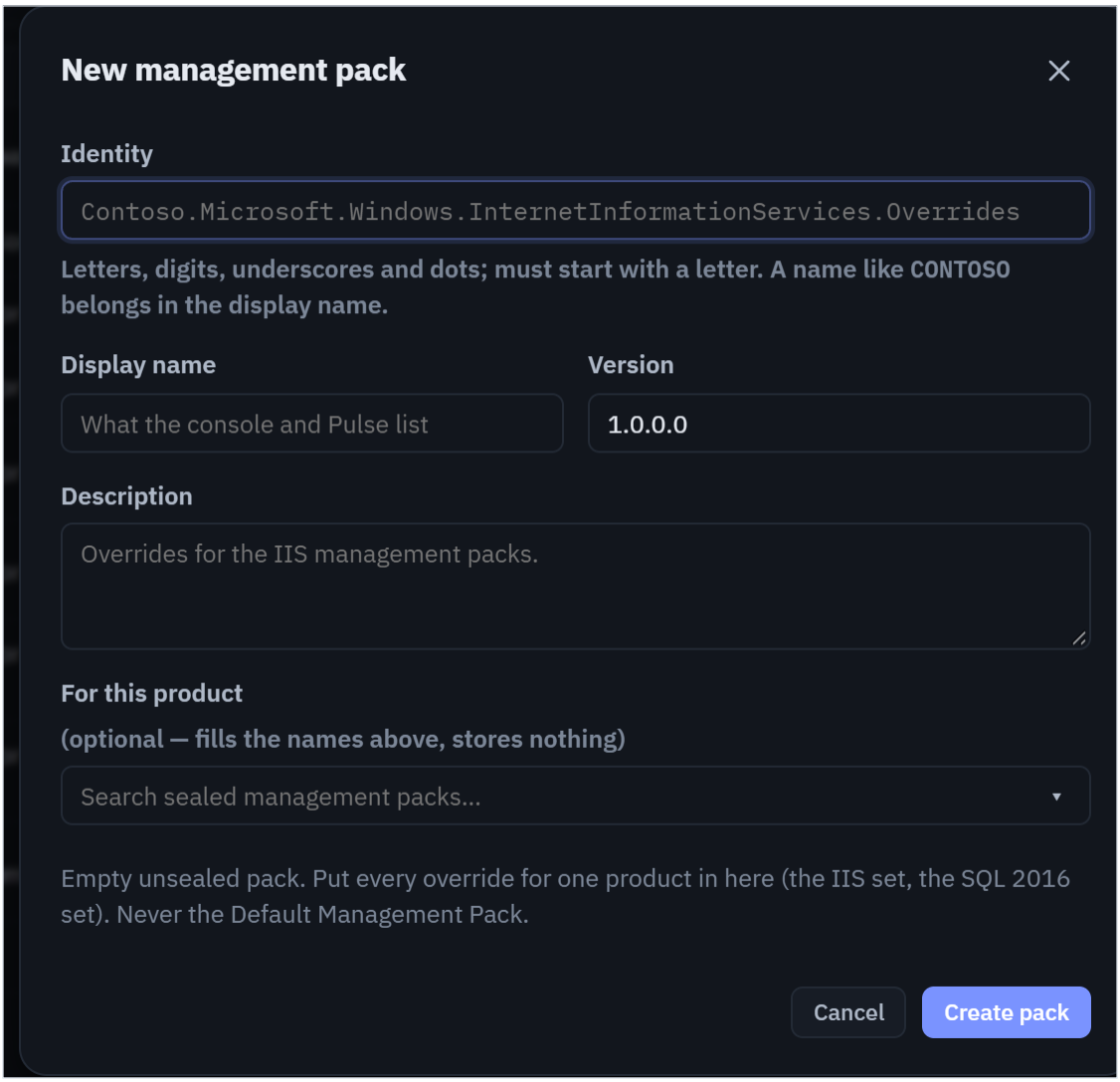


Figure 20 The **New management pack** dialog, unsaved.

1. **Identity** — letters, digits, underscores and dots, starting with a letter. A name like `CONTOSO` belongs in the display name, not here.
2. **Display name**, **Version**, **Description**.
3. **For this product** optionally fills the names above from a product you pick. It stores nothing itself.
4. **Create pack**.

The dialog states the intent plainly: "*Empty unsealed pack. Put every override for one product in here (the IIS set, the SQL 2016 set). Never the Default Management Pack.*"

> **Important — Why the Default Management Pack is refused**
>
> Anything stored in the Default Management Pack cannot be removed without removing everything else in it, and it accumulates references that make other packs undeletable. Pulse refuses it as an authoring destination, and so should you.

7.3 Groups

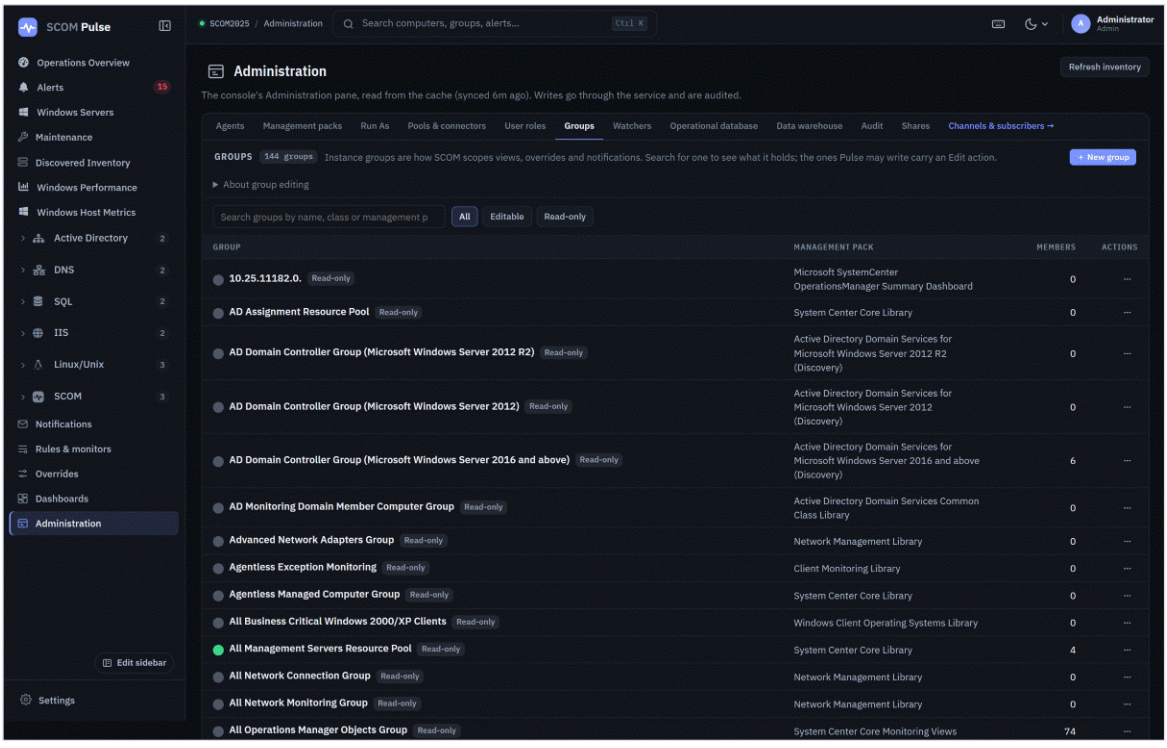


Figure 21 Administration → Groups, filtered by whether Pulse may write to them.

The page explains the scope: "*Instance groups are how SCOM scopes views, overrides and notifications. Search for one to see what it holds; the ones Pulse may write carry an Edit action.*" Filter by **All**, **Editable** or **Read-only**.

Group authoring needs the **Author** profile. An Author who is not an administrator sees **Administration** with this tab alone, and a group carries **Edit** where one of their Author roles covers it — on this tab and on the group's own page alike.

> **Important — Editing a group cannot widen your own access**
>
> A group can decide what a SCOM user role reaches, so Pulse does not let an edit become a way to give yourself more. If your Author role is limited to groups, every member you **add** to a group inside it must already be something that role covers. A **dynamic** group's rules cannot be held to that — a rule matches by object type and property across the whole management group, today and as new objects are discovered — so for a group inside a role of yours that is limited to groups, the rules are read-only and the page says why. An Author or Administrator whose role is not limited to groups can change them. Creating a new group is unaffected: nothing is scoped to it yet.

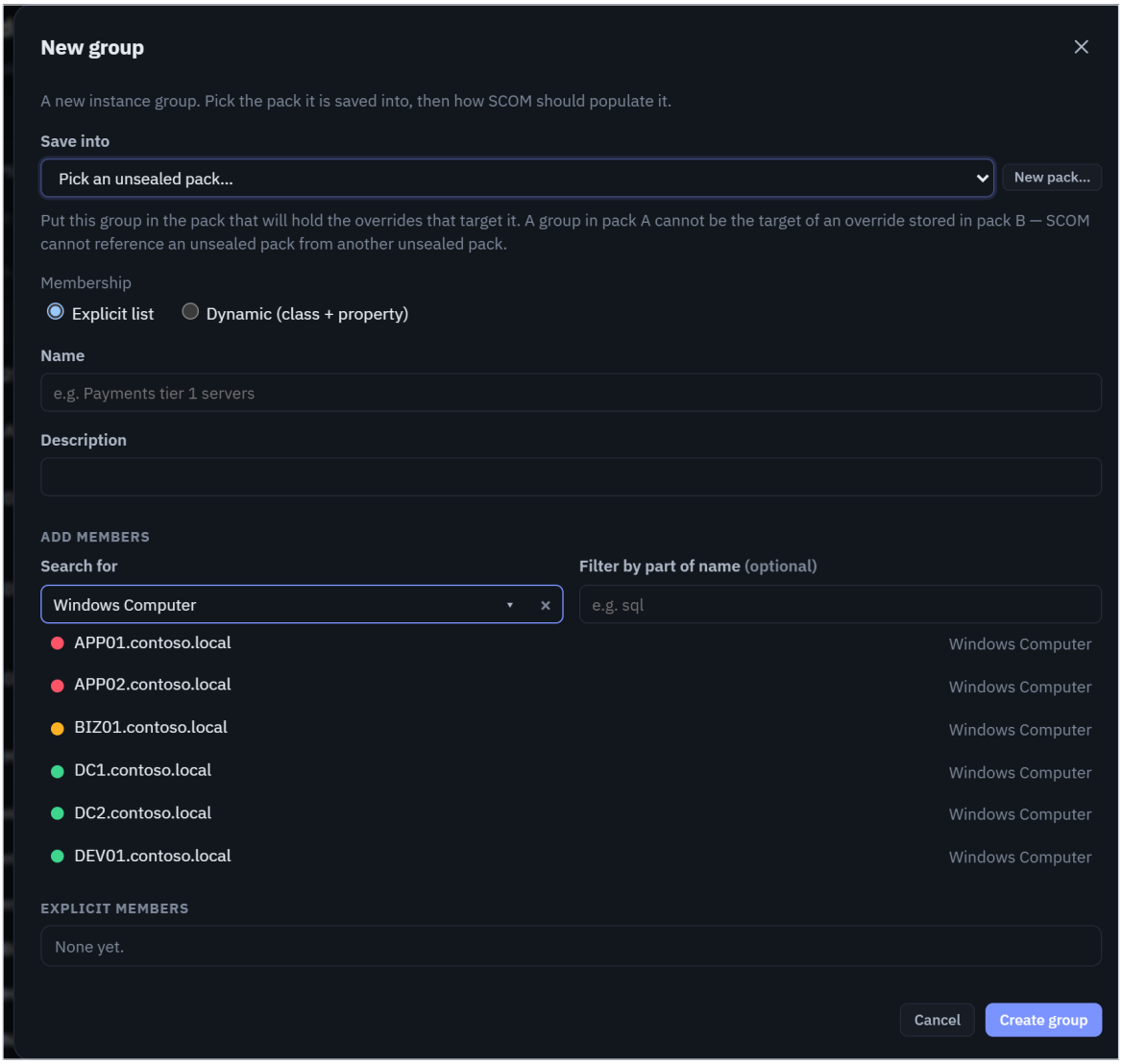


Figure 22 The **New group** dialog, unsaved.

1. **Save into** — pick an **unsealed** pack, or **New pack...**.
2. **Membership** — **Explicit list** or **Dynamic (class + property)**.
3. **Name** and **Description**.
4. Add members, then **Create group**.

> **Caution — Put the group in the pack that will hold the overrides targeting it**
>
> The dialog states the constraint: "*A group in pack A cannot be the target of an override stored in pack B — SCOM cannot reference an unsealed pack from another unsealed pack.*" Getting this wrong means authoring the group twice.

What Pulse will refuse: **empty static groups**, **unsupported formulas**, and sealed or Default-pack destinations. Complex formulas, sealed and default-pack edits, and computer-group creation remain Operations console work. Eligible console-created groups can be edited here.

8 Channels, subscribers and subscriptions

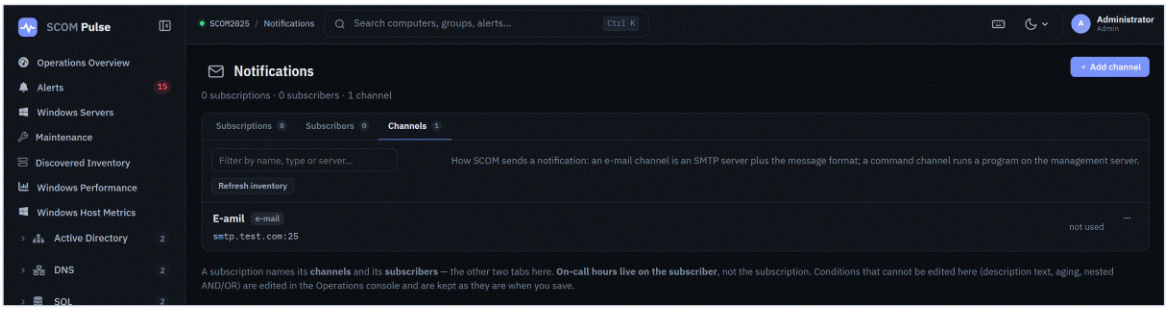


Figure 23 Notifications → Channels: each channel with its type and endpoint, and how many subscriptions send through it. **Subscribers** is the tab beside it.

Channels and subscribers are two tabs of the **Notifications** page, beside the **Subscriptions** that use them — as the Operations console keeps the three together. (Until 1.3.23 they were a tab of Administration; **Channels & subscribers** → in Administration's tab strip, and any old link, lead here.) Each tab is its own SCOM permission: **Channels** and **Subscribers** are shown to a role that reads the Administration space and that list, and the buttons on them to a role that may change it.

CHANNELS is how SCOM sends: "*an e-mail channel is an SMTP server plus the message format; a command channel runs a program on the management server.*" **SUBSCRIBERS** is who is notified, at which addresses, and when they are available — "*An address belongs to a channel — that is how SCOM knows which way to reach them.*" Each list has a filter box. A row opens the item's details; its ⋯ menu holds **Edit…** and **Delete…**. For someone who reads every subscription, each row also says how many subscriptions use it — "*used by 3 subscriptions*", which opens those subscriptions — or "*not used*": what to look at before deleting a channel or a subscriber. **Refresh inventory** reads both lists from SCOM again, for a change made in the Operations console.

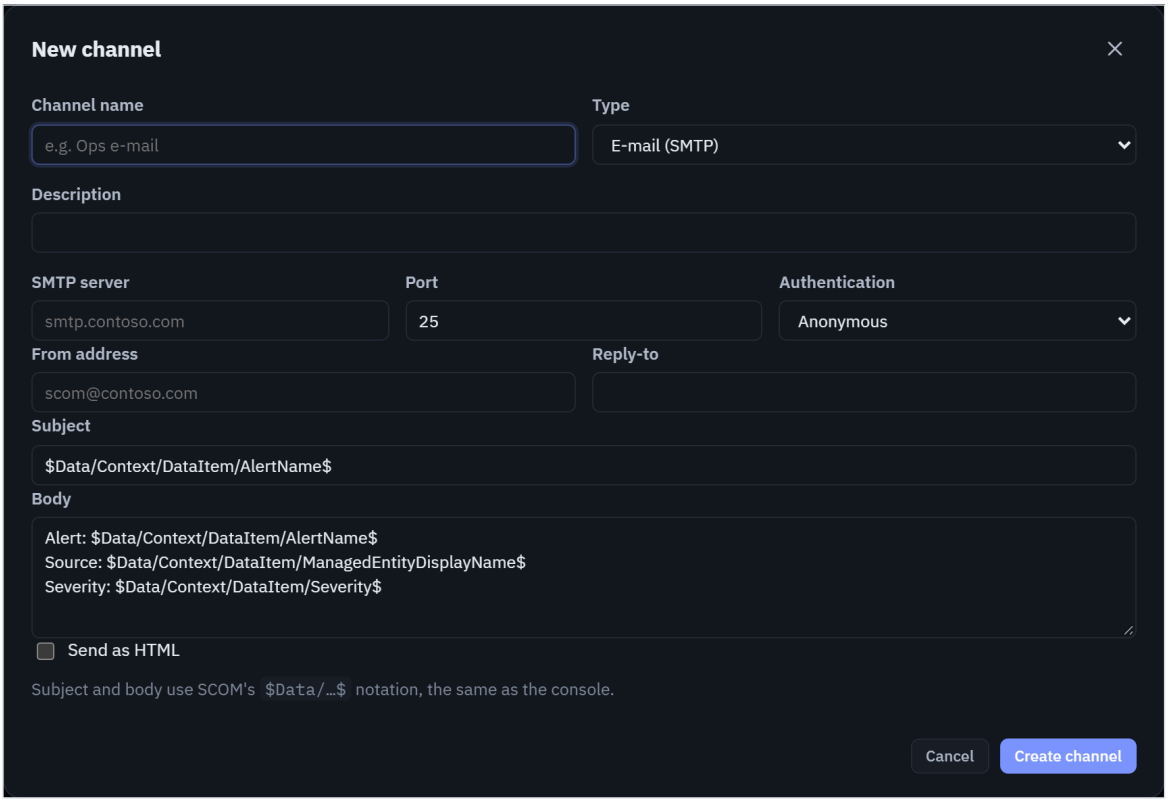


Figure 24 The **New channel** dialog, unsaved, on the e-mail type.

Types: **E-mail (SMTP)**, **SMS**, **Instant message (SIP)** and **Command**. For e-mail: **SMTP server**, **Port**, **Authentication** (**Anonymous** or **Windows**), **From address**, **Reply-to**, **Subject**, **Body** and **Send as HTML**. Subject and body use SCOM's `$Data/…$` notation, exactly as the console does.

A channel is listed under the name the Operations console gives it in its own **Channels** list. Channels of a kind SCOM Pulse does not edit — **Microsoft Teams** — are listed too, can be used in a subscription and can be deleted here; they are created and edited in the Operations console. Teams channels are shown only when the Operations console is installed on the SCOM Pulse server (Installation Guide), and on a server without it SCOM Pulse refuses to save a subscription that uses one — "*This subscription uses a notification channel this server's Operations Manager client cannot read… Nothing was changed.*" — rather than save it without that channel.

Caution — A command channel runs a program on the management server

It executes with the notification account's rights, on the management server, with alert data as arguments. Anyone who can create or edit a command channel can run code there. Treat channel administration as a privileged operation, restrict who holds the SCOM Administrator profile or a Delegated administrator role with **Notification Management**, and review command channels during access reviews.

On-call hours live on the subscriber, not the subscription — SCOM has no schedule on a subscription. Operators select from your channels and subscribers but never see or edit them; their procedure is in the *User Guide*, chapter 12.

Who may do what here is SCOM's decision, and there is no Pulse setting for it. Everyone with the Operator profile or better may manage the subscriptions that notify only themselves. A role that administers notifications — Administrator, or a Delegated administrator role with **Notification Management** that is not limited to groups — manages every subscription, subscriber and channel. A Read-only administrator reads them all.

Important — Do not overwrite console-authored criteria

Pulse supports severity, priority, groups, classes, alert-name-contains, rules and monitors, custom fields and only-new. Criteria beyond that — description text matching, alert aging, nested AND/OR — are edited in the Operations console and are never overwritten from Pulse.

9 Run As, resource pools, connectors and user roles

9.1 Run As accounts and profiles

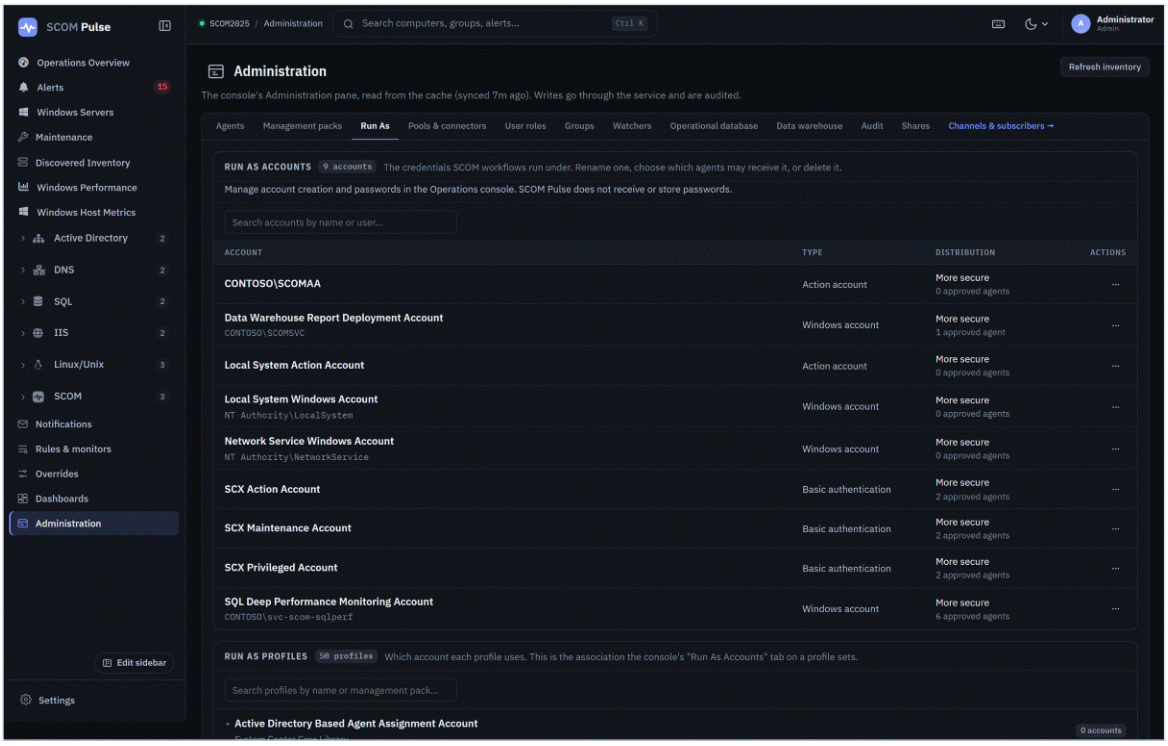


Figure 25 Administration → Run As: accounts with their type and distribution, and the profiles that bind them.

Pulse shows each account's **TYPE** (action account, Windows account, Simple authentication, Basic authentication…), its **DISTRIBUTION** (More secure, with the number of approved agents) and its profile bindings. You can rename an account, choose which agents may receive it, or delete it.

> **Important — Credentials stay in the Operations console**
>
> "*Manage account creation and passwords in the Operations console. SCOM Pulse does not receive or store passwords.*" Creating an account and rotating a password are console operations. What Pulse administers is the **metadata, distribution and profile binding** around them.

Distribution approval targets the **health service id** of each management server ("6 Agents and management servers").

9.2 Resource pools and connectors

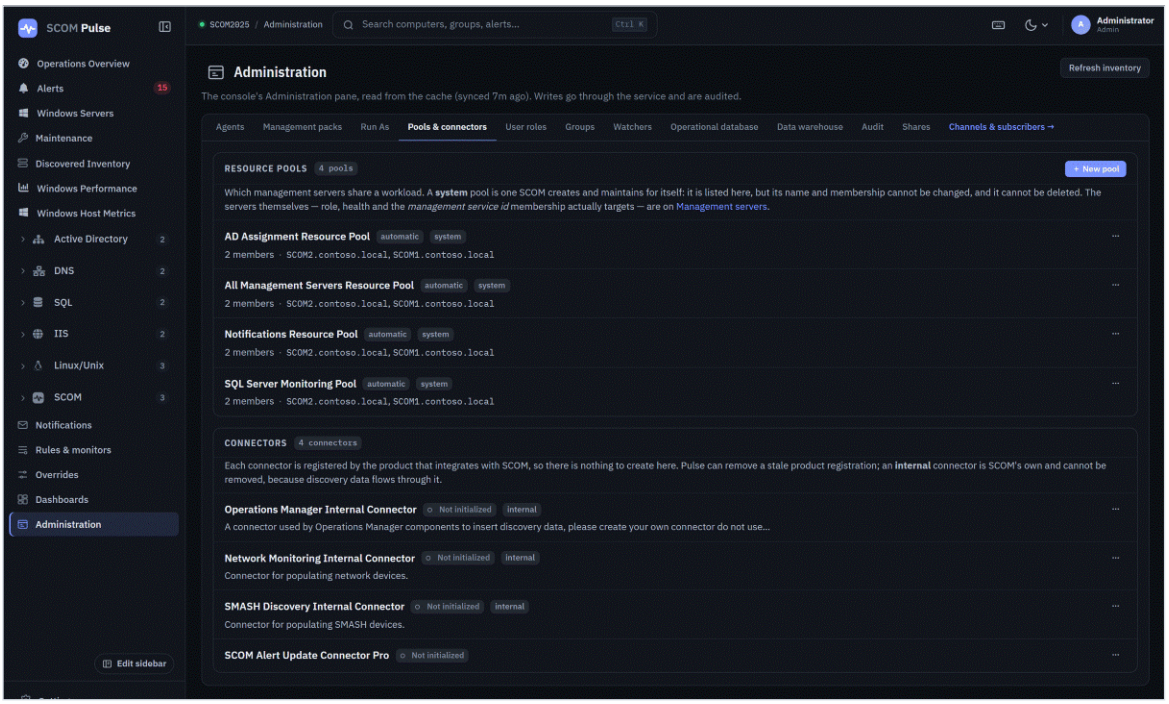


Figure 26 Administration → Pools & connectors.

A **resource pool** is which management servers share a workload. A **system pool** is one SCOM creates and maintains for itself: it is listed, but its name and membership cannot be changed and it cannot be deleted.

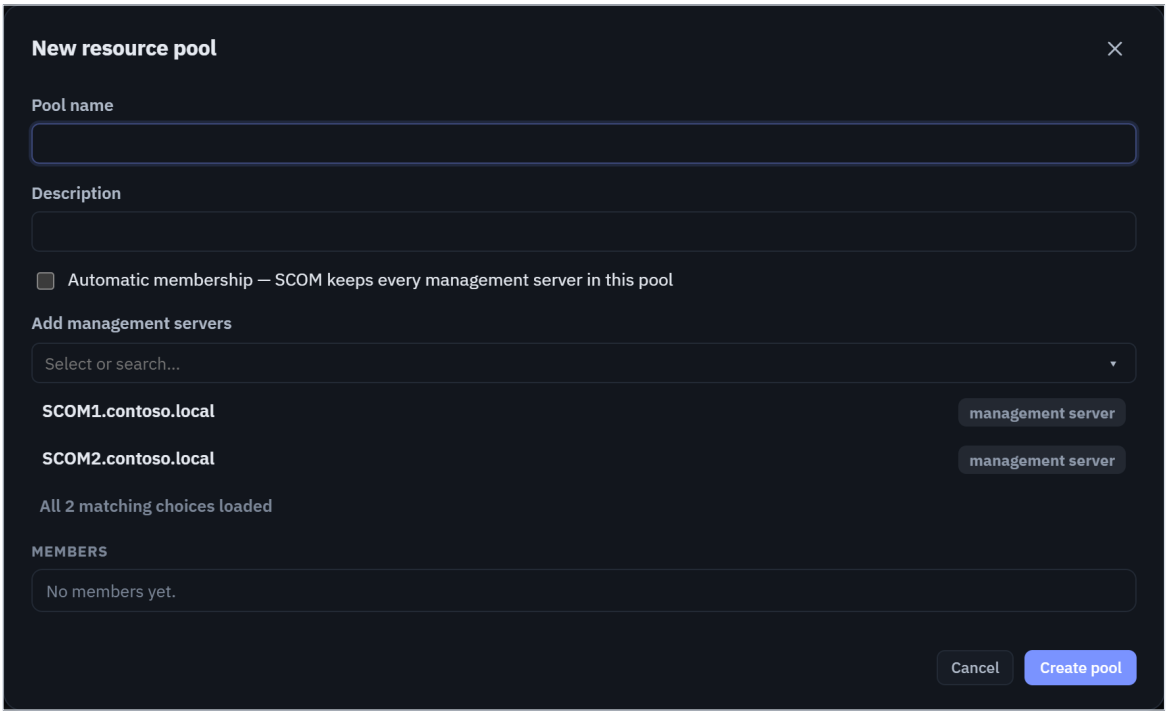


Figure 27 The **New resource pool** dialog, unsaved.

Automatic membership keeps every management server in the pool. Manual membership targets the **management service id** — the reason both ids are shown on the Management servers page.

Connectors are registered by the product that integrates with SCOM, "*so there is nothing to create here.*" Pulse can remove a stale product registration. An **internal** connector is SCOM's own and cannot be removed, because discovery data flows through it.

> **Important — Inventory, not configuration**
>
> This is a connector inventory with a removal action. Configuring what a connector does remains the integrating product's job and, where SCOM is involved, the Operations console's.

9.3 SCOM user roles

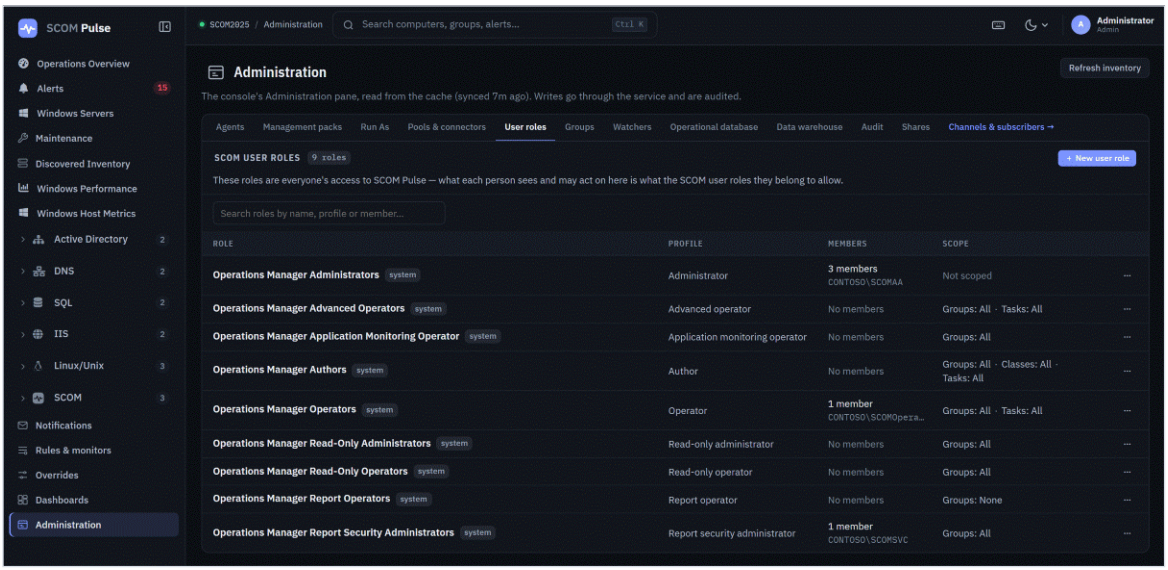


Figure 28 Administration → User roles.

Each row gives the **ROLE** (and whether it is a **system** role), its **PROFILE**, its **MEMBERS** and its **SCOPE** — for example "*Groups: All · Tasks: All*", or "*Not scoped*". A line above the list says what these roles mean here: they are everyone's access to SCOM Pulse.

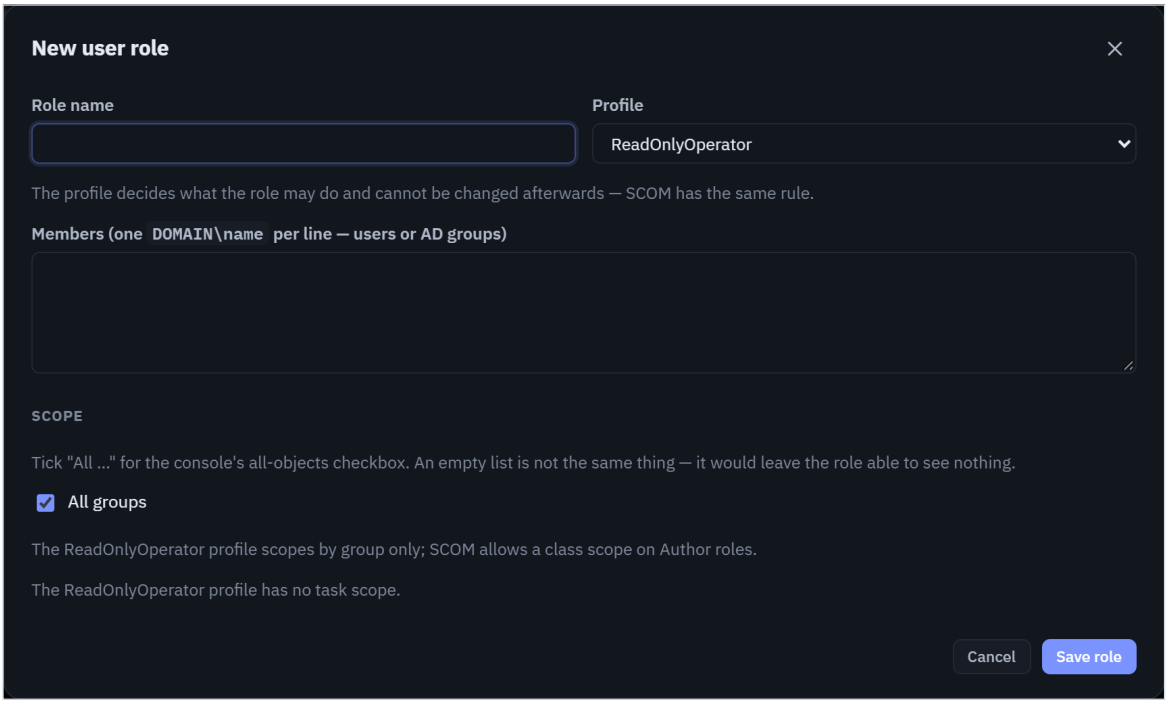


Figure 29 The **New user role** dialog, unsaved, on the ReadOnlyOperator profile.

1. **Role name**.
2. **Profile** — **ReadOnlyOperator**, **Operator**, **AdvancedOperator**, **Author** or **Administrator**. "*The profile decides what the role may do and cannot be changed afterwards — SCOM has the same rule.*"
3. **Members**, one `DOMAIN\name` per line, users or AD groups.
4. **SCOPE**. The available scope depends on the profile: ReadOnlyOperator scopes by group only and has no task scope; a class scope is allowed on Author roles.

> **Caution — An empty scope list is not "all"**
>
> "*Tick 'All ...' for the console's all-objects checkbox. An empty list is not the same thing — it would leave the role able to see nothing.*" This is the single most common mistake in SCOM role authoring.

Built-in system roles are protected: you may manage membership, not redefine them.

Creating, editing and deleting a user role needs the SCOM Administrator profile in Pulse, even where SCOM would let it be delegated ("4 The access model"). A role created here uses one of the five profiles above. Read-only administrator and Delegated administrator roles are created in the Operations console; Pulse lists them and honours them.

10 Watchers

A watcher is a lightweight check SCOM does not ship: a website, a Windows service, or a TCP port. Pulse authors the classes, discoveries and monitors into an unsealed pack you choose, and **SCOM's agent** runs the check.

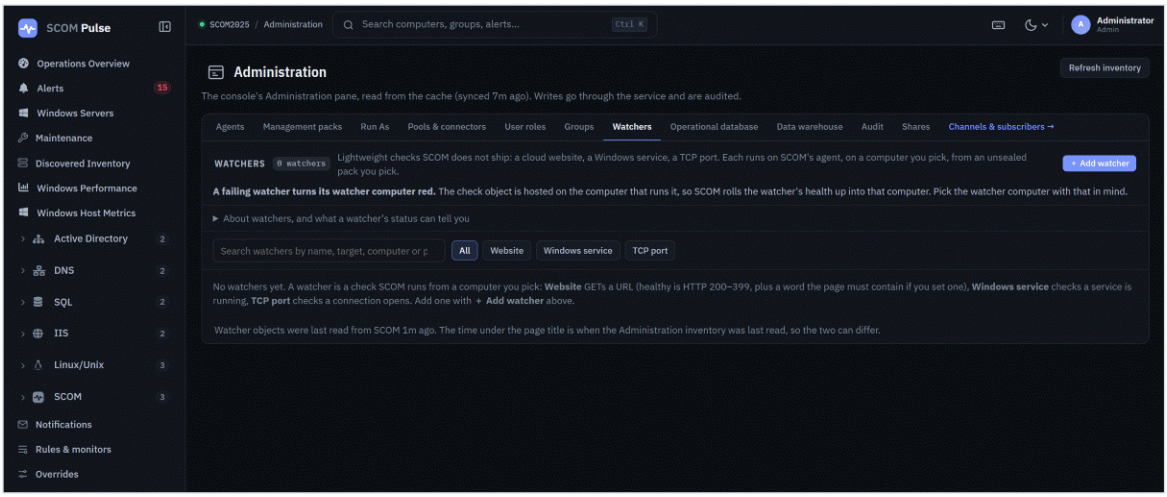


Figure 30 Administration → Watchers, before any watcher has been added.

10.1 The three recipes

Type	Healthy means
Website	A GET returns **HTTP 200–399**, plus a word the page must contain if you set one
Windows service	The named service is running — scoped to where it is installed, to a group, or to specified computers
TCP port	A connection opens

10.2 Choosing the watcher computer

> **Caution — A failing watcher turns its watcher computer red**
>
> The page states it: "*The check object is hosted on the computer that runs it, so SCOM rolls the watcher's health up into that computer. Pick the watcher computer with that in mind.*"
>
> Choosing a **management server** as the watcher host is the case to think hardest about. A website you do not control failing overnight will turn a management server critical, and management-server health drives alerting for everything else. Prefer a utility server whose health you are willing to couple to the thing being watched.

10.3 Creating a watcher

A watcher recipe takes a **watcher computer**, a **destination unsealed pack**, a **target scope**, an **interval** and, for a website, optional **content** to match. The interval is **per recipe per pack**, not per watcher — two watchers of the same type in the same pack share it.

Removal deletes the watcher's discovery. It can leave the shared recipe definitions behind in the pack, which is harmless but worth knowing when you are trying to work out why a pack is not empty.

> **Note — The watcher form was not captured**
>
> **Add watcher** opens a short menu — **Website**, **Windows service**, **TCP port** — and choosing one opens that type's creation dialog. No figure of the dialog appears here: it did not open during the session that captured these screenshots, which later re-testing could not reproduce and which is recorded in `Documentation-Verification.md`. The field list above therefore comes from the product reference and the tab's own help text rather than from a capture.

11 The SCOM operational database

Caution — Three different stores, three different retentions

Keep these apart in your head and in your change requests:

1. **Pulse's SQLite cache** — `pulse.db`. Retention is set in **Settings → Sync & data** and **Service & storage**. Losing it loses no monitoring data; it resyncs from SCOM.
2. **The SCOM operational database** — this chapter. Its retention drives SCOM's nightly grooming. **Deletion is irreversible.**
3. **The SCOM data warehouse** — "12 The SCOM data warehouse". Its own datasets, its own retentions.

A Pulse backup backs up **only the first**. It does not back up either SCOM database.

11.1 Storage usage

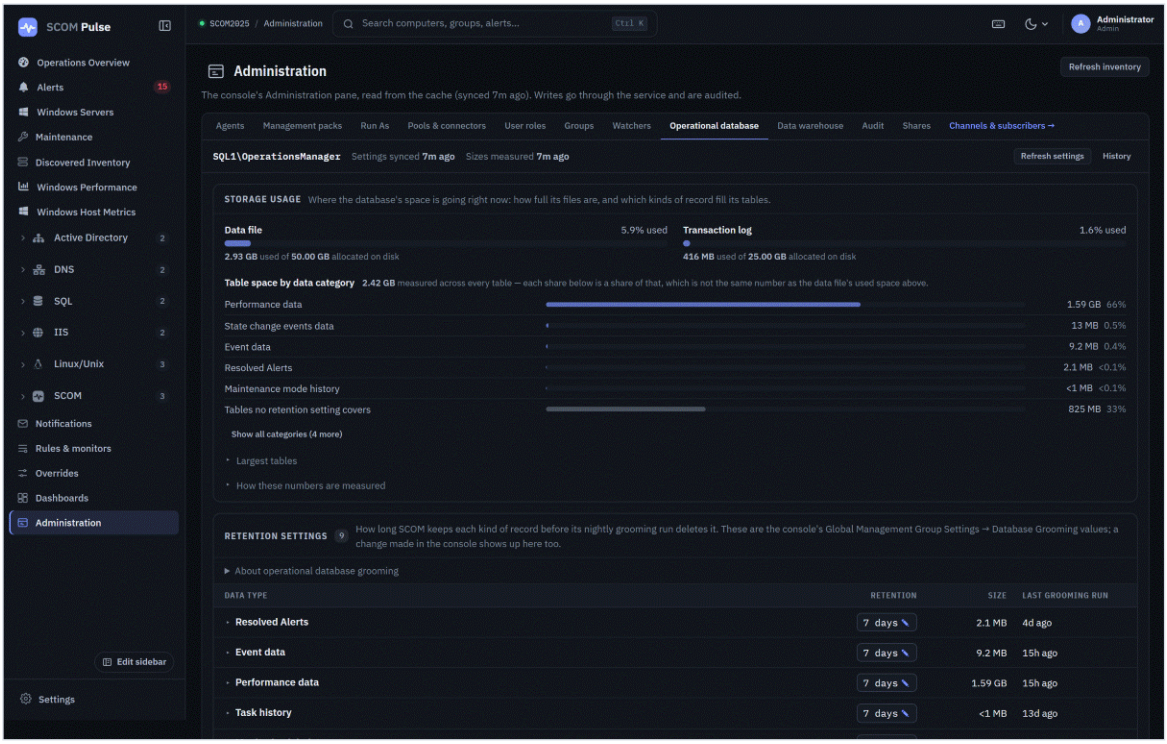


Figure 31 Administration → Operational database: file usage and where the space is going.

The header names the instance and carries two separate clocks — "*Settings synced 48m ago*" and "*Sizes measured 48m ago*" — with **Refresh settings** and **History**.

Data file and **Transaction log** show used against allocated. **Table space by data category** breaks the measured total down by record type.

> **Important — Two totals that are not the same number**
>
> The page warns of this directly: the table-space total is measured across every table, and each category's share is a share of *that*, which is not the same as the data file's used space. A category that reads 44% is 44% of measured table space, not 44% of the file. There is also a **Tables no retention setting covers** category — real space that no grooming setting will ever reclaim.

11.2 Retention and grooming

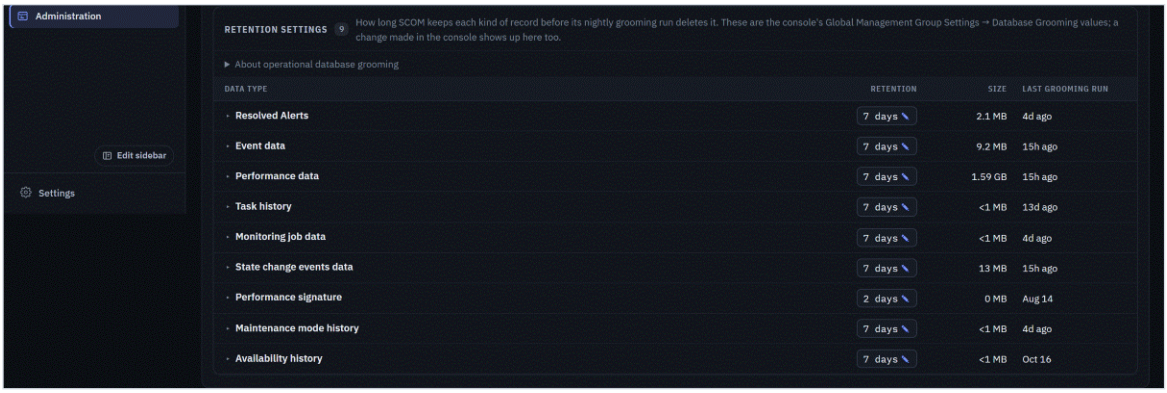


Figure 32 Administration → Operational database: the retention settings, their measured sizes and the last grooming run for each.

These are the console's **Global Management Group Settings → Database Grooming** values. A change made in the Operations console shows up here, and a change made here is a change there.

> **Required access — The SCOM Administrator profile, or a Delegated administrator role with Global Settings — with an explicit confirmation**

1. Click the □ beside a retention value.
2. Enter the new value. The range is **1–99 days**.
3. Confirm.

Expected result — the value updates, the change is audited, and it takes effect at SCOM's **next nightly grooming run**. Nothing is deleted at the moment you save.

> **Caution — Lowering retention deletes data at the next grooming run, permanently**
>
> Grooming is a delete. Once state-change events or performance data are groomed out of the operational database they are gone from it; only what the data warehouse already holds survives. And raising retention afterwards **cannot recover** anything already deleted — it only affects data still present. Agree a retention reduction the way you would agree a deletion, because it is one.

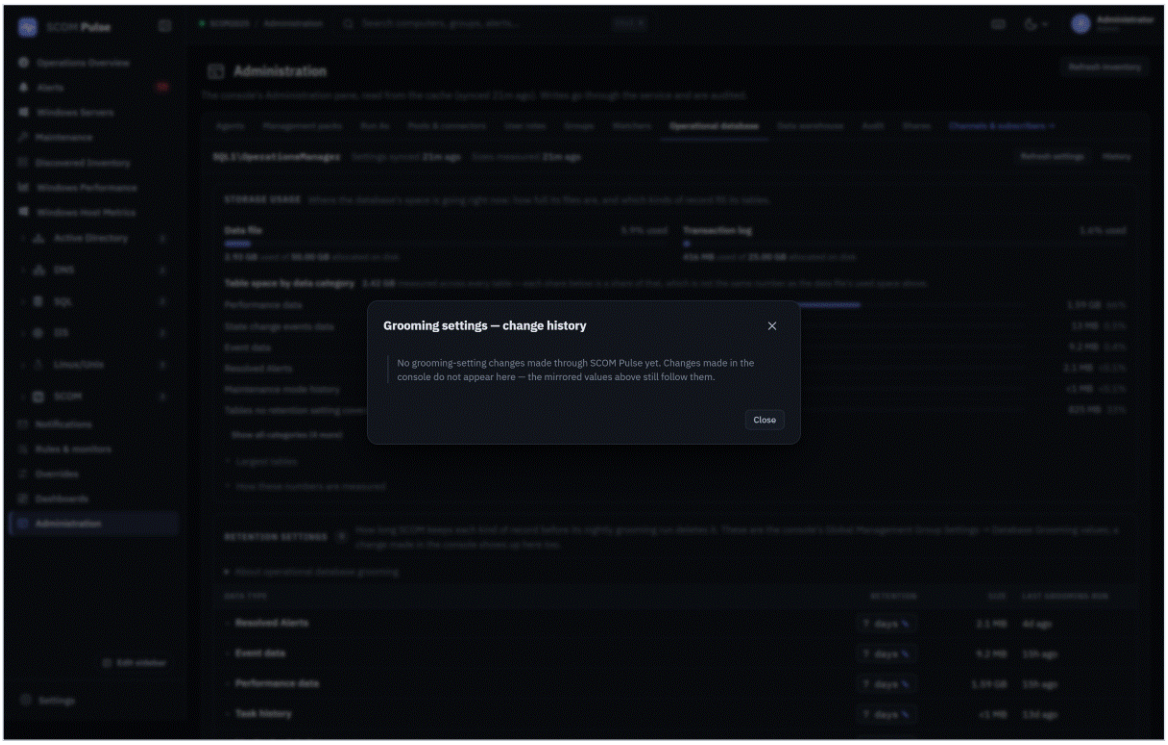


Figure 33 The operational database **History** view.

The **LAST GROOMING RUN** column is a useful health signal in its own right. In the reviewed deployment most types had groomed within a day, while **Performance signature** showed a last run of *Aug 14* — the sort of thing worth understanding before you trust a size.

12 The SCOM data warehouse

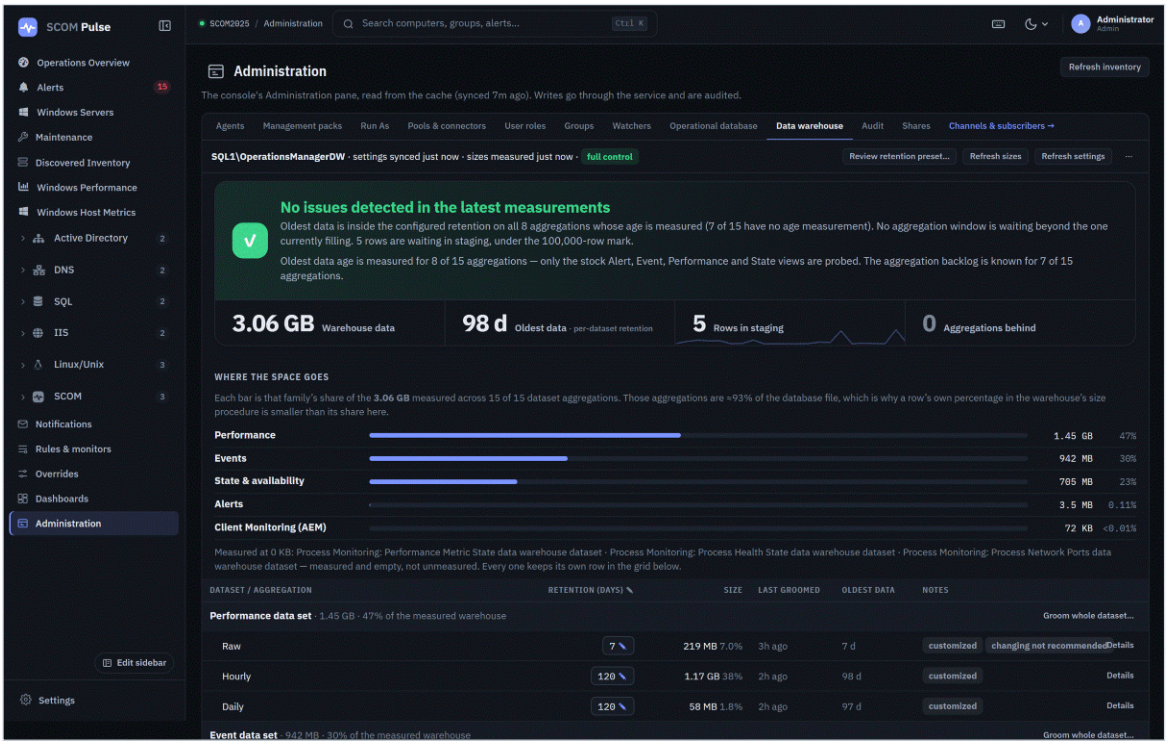


Figure 34 Administration → Data warehouse: the header with capability, the measurement summary, and where the space goes.

12.1 Read the header before the numbers

The header carries the instance, two clocks — *settings synced* and *sizes measured* — and a **capability** indicator ("*full control*" in the reviewed deployment). **Review retention preset...**, **Refresh sizes** and **Refresh settings** are separate actions because they do separate work: sizes are expensive to measure, settings are cheap to read.

> **Important — Measured, unmeasured and zero are three different things**
>
> The reviewed deployment stated: "*Oldest data age is measured for 8 of 15 aggregations … The aggregation backlog is known for 7 of 15 aggregations.*" It also distinguished three datasets "*measured at 0 KB … measured and empty, not unmeasured.*"
>
> Do not report an unmeasured aggregation as healthy, and do not report it as zero. If a number is missing, say it was not measured.

Caution — "Not measured yet" can appear even when other data is available

When the figures for this guide were captured, the warehouse page first reported **Not measured yet** for sizes and oldest data while retention values and other information were already available. Sizes are measured on their own schedule; a fresh install, a recent service restart or a failed measurement all produce it. **Refresh sizes** is the control that answers it.

12.2 Datasets, aggregations and retention



Figure 35 The dataset grid: each dataset's Raw, Hourly and Daily aggregations with their own retention, size, last-groomed time and oldest data.

Each dataset is one group, and **Raw**, **Hourly** and **Daily** keep their own retention, size, last-groomed time and oldest data. **Groom whole dataset** in a group heading runs over all of them at once, because "*SCOM has no way to groom a single aggregation.*" **Details** opens an aggregation's trend, grooming runs and change history.

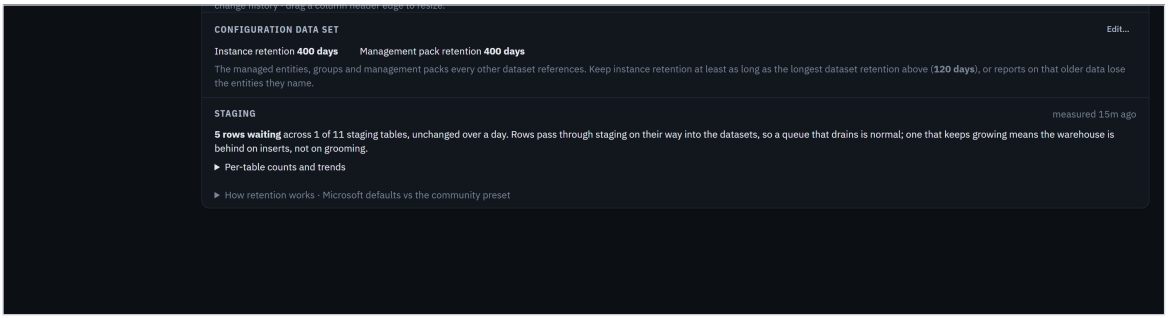


Figure 36 The **Configuration data set** and the staging queue.

> **Important — Keep configuration retention at least as long as the longest dataset retention**
>
> The **Configuration data set** holds the managed entities, groups and management packs every other dataset references. If instance retention is shorter than your longest data retention, reports over that older data lose the entities they name. The page states the rule and the number to match.

Staging is the queue on the way into the datasets. "*A queue that drains is normal; one that keeps growing means the warehouse is behind on inserts, not on grooming.*" The reviewed deployment showed 49 rows across 1 of 11 staging tables, under the 100,000-row mark it flags.

12.3 The retention preset

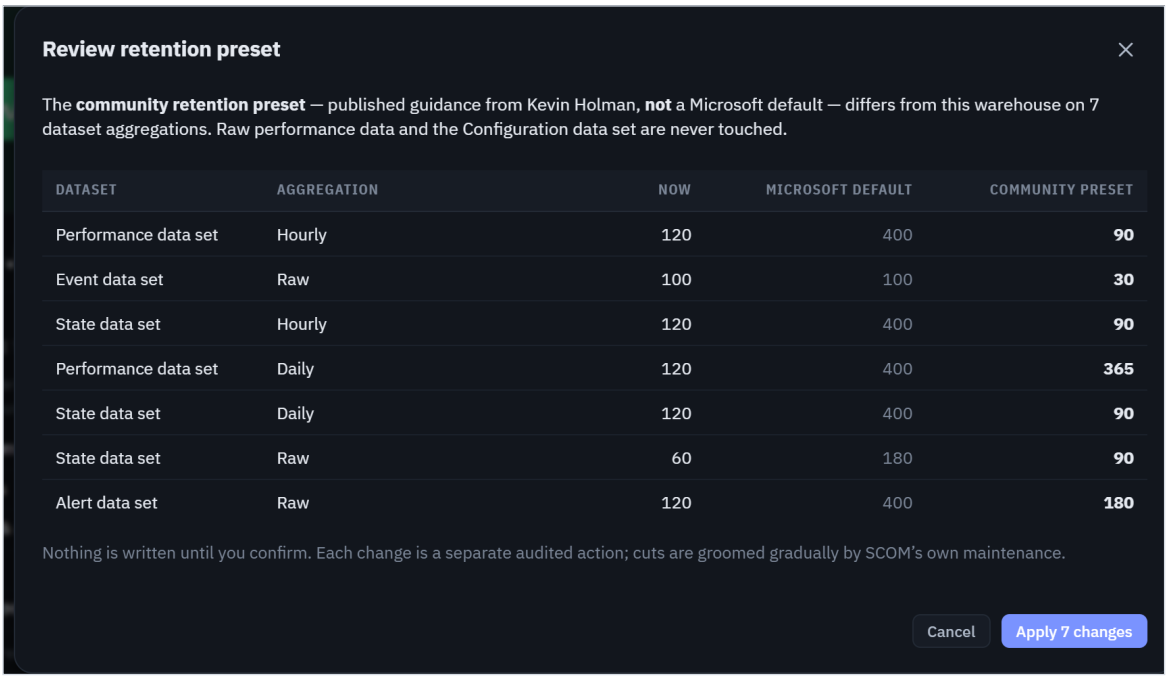


Figure 37 Review retention preset..., unsaved. The dialog compares the current values with the Microsoft default and the community preset, and nothing is written until it is confirmed.

The dialog is explicit about provenance: "*The community retention preset — published guidance from Kevin Holman, not a Microsoft default.*" It compares **NOW**, **MICROSOFT DEFAULT** and **COMMUNITY PRESET**, and states what it will not touch: raw performance data and the Configuration data set.

"Nothing is written until you confirm. Each change is a separate audited action; cuts are groomed gradually by SCOM's own maintenance."

> **Caution — Applying the preset is a bulk retention change**
>
> In the reviewed deployment the preset differed on **7** aggregations, and not all in the same direction. Four were cuts — raw event data from 100 days to 30, and hourly performance, hourly state and daily state from 120 to 90 — while three would lengthen retention: daily performance from 120 to 365, raw state from 60 to 90 and raw alerts from 120 to 180. The cuts schedule the permanent deletion of everything outside the new windows, and the increases grow the warehouse. Review each line against what your reporting and your auditors actually need, and treat it as a change, not a tidy-up.

12.4 Permissions and capability

Four capabilities can fail independently: **read**, **measure**, **update** and **groom**. The header's capability indicator is the honest answer for your deployment.

> **Important — Do not prescribe `db_owner`, and do not claim Pulse is read-only against the warehouse**
>
> Read-only historical charts need far less than warehouse administration. Administration additionally exposes retention changes and manual grooming, guarded by admin authorisation, confirmation and the service account's SQL capability. Use the page's own permission diagnostics to find out what your service account can do; the exact grants depend on your SQL configuration and cannot be stated generically.

> **Note — "Not answering in time" is not a permissions problem**
>
> When the page says **The warehouse is not answering in time**, or a warehouse test reports that *a query did not finish in time*, Pulse reached the warehouse and signed in — name resolution, the firewall and the service account's access are working, and granting more rights will not help. The warehouse was busy or blocked: a long-running report, SCOM's own aggregation or grooming, or SCOM adding warehouse storage can hold a query up. Check the warehouse SQL instance for blocking and for CPU, memory or disk pressure. Pulse tries again on its next check, and the figures on the page stay at the last values it read.

> **Caution — Never run grooming to produce documentation or to test a permission**
>
> Manual grooming deletes data. No grooming was run while writing this guide.

13 Sharing and audit

13.1 Published links

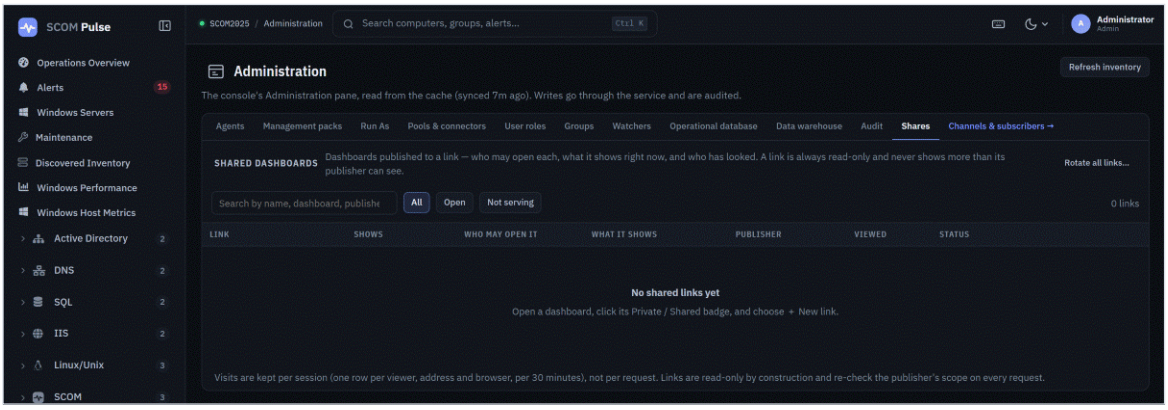


Figure 38 Administration → Shares, with no links published.

This is the all-shares view: "*Dashboards published to a link — who may open each, what it shows right now, and who has looked.*" Owners see their own shares on the dashboard itself; administrators see everyone's here, with **Rotate all links…**.

Two properties are structural rather than configurable:

- **A link is always read-only.**
- **A link never shows more than its publisher can see**, and the publisher's scope is re-checked on every request. Revoking the publisher's access revokes what the link can show.

Publishing requires an owner with the Operator profile or better, or a SCOM Administrator, and the policy in "5 Settings" to be enabled. Exposure of the whole management group requires explicit acknowledgement. Visit records are kept per session — one row per viewer, address and browser, per 30 minutes — not per request.

> **Documentation gap — No link was published or rotated**
>
> Publishing is an outward-facing change, so none was made. The audience, expiry, embedding and revocation controls are documented from the settings page and the product reference; the link-creation dialog's field behaviour is **unverified**.

13.2 The audit trail

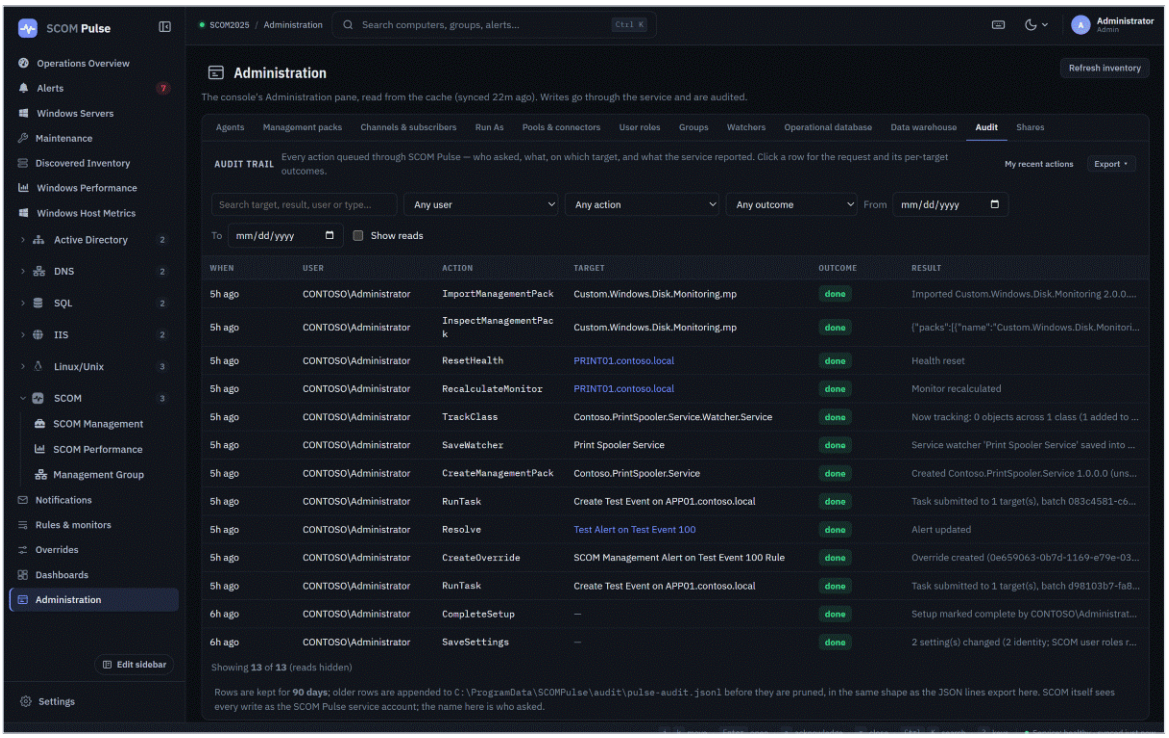


Figure 39 Administration → Audit.

"Every action queued through SCOM Pulse — who asked, what, on which target, and what the service reported." Filter by user, action, outcome (**Failed or partly failed**, **Failed**, **Done**, **Queued**, **Running**) and date range. **Show reads** includes read operations, which are hidden by default. **Export** ⬇ downloads the trail.

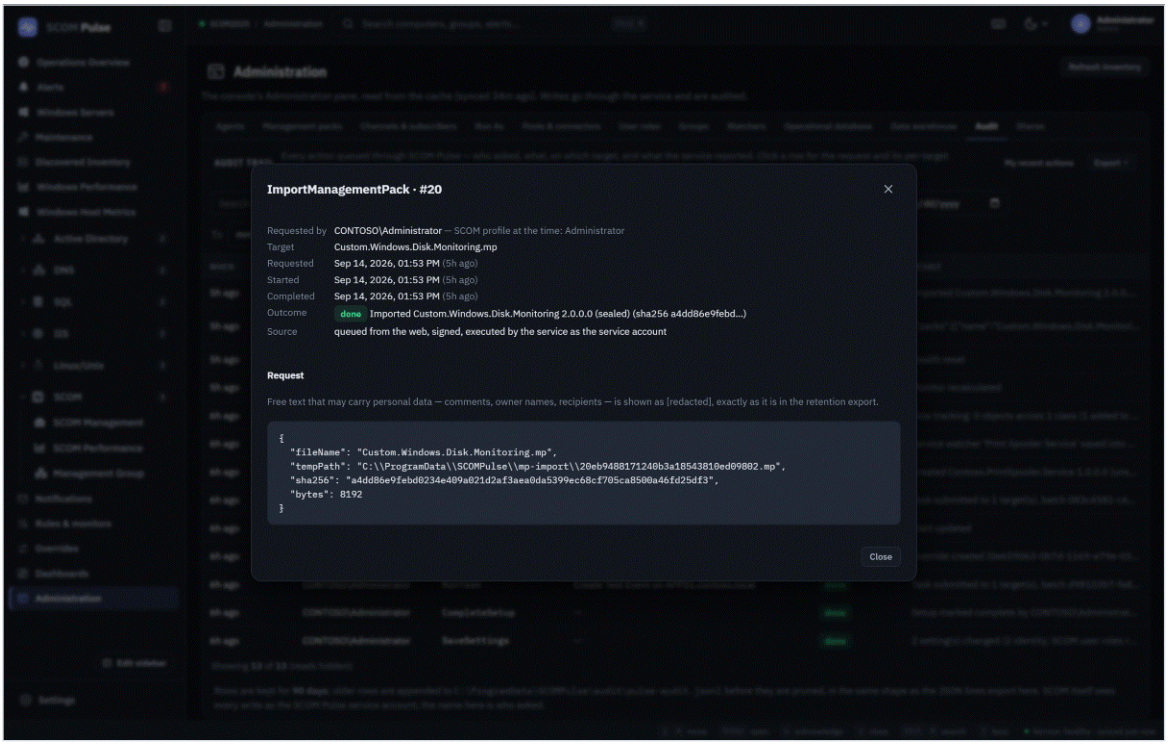


Figure 40 An audit row expanded to its request and per-target outcomes.

> **Important — Optimistic UI is not a confirmed outcome**
>
> The interface may show an action as accepted before the service has finished it. The audit row's **OUTCOME** — `queued`, `running`, `done`, `failed` — is the confirmed answer, and per-target details sit beneath it. When somebody asks whether a change actually happened, this is the place to look, not the toast they remember.

Retention has a floor, and pruned rows are appended to the JSONL export before they are removed, in the same shape as the download. Payloads are redacted. And the footer states the identity model once more: "*SCOM itself sees every write as the SCOM Pulse service account; the name here is who asked.*"

14 Operations, backup and recovery

14.1 Daily health

Check, in this order:

1. **Settings** status strip: **SERVICE Running**, a recent sync, **ACCESS** naming the number of SCOM user roles synced, **INTEGRITY ok**, a recent **BACKUP**, and **LICENSE** as you expect (Figure 7).
2. **Settings** → **Service & storage**: the expected management group, sensible cache counts, database size and integrity time (Figure 1).
3. **Administration** → **Audit**, filtered to **Failed or partly failed**.
4. The Windows event log on the Pulse server, source **SCOMPulse**.

14.2 Backups

Pulse backs up its own SQLite database on the schedule in **Settings** → **Service & storage**: every 24 hours, keeping 7, in a local folder.

> **Caution — Never back up pulse.db by copying the file**
>
> SQLite writes through `-wal` and `-shm` sidecars, and the service writes continuously. An ordinary file copy of a live database can be inconsistent or unusable. Use the service's backup command, which produces a verified backup.

```
PulseService.exe -backup-db [-db <path>] [-to <file.db>]
PulseService.exe -check-db [-db <path>]
PulseService.exe -list-backups
PulseService.exe -restore-db [-db <path>] [-from <backup.db>] [-newest]
```

What a Pulse backup contains and what it does not

- **Contains**: settings, dashboards, sidebar and personal preferences, the action queue and the audit trail, to the moment of the backup. Cached monitoring data too, though that resyncs from SCOM anyway.
- **Does not contain**: the SCOM operational database, the SCOM data warehouse, IIS configuration, or the service account.
- **`action.key` is not automatically part of the SQLite backup.** Protect it separately. A different or regenerated key makes already-queued actions fail closed; they must be resubmitted.

14.3 Integrity

Pulse runs `quick_check` on a schedule (every 6 hours, and at every start). On a detected failure it:

- records an **integrity-fault marker**,
- writes **event 30** under source **SCOMPulse**, and
- answers requests with **503, database needs attention**.

> **Caution — Preserve the damaged file**
>
> Keep the faulted database for support before you replace it. `-check-db` can clear a recorded fault **only if verification passes**. Do not delete the database, bypass the integrity check or disable the protection as a routine fix — those destroy the evidence of what went wrong, and no scheduled check makes any system immune to disk, OS or storage failure.

14.4 Restore

> **Required access — Server console access, and a maintenance window**
>
> Restoring takes Pulse offline. Do not run this to try it out.

> **Important — `-restore-db` is interactive unless you tell it otherwise**
>
> Correcting older documentation: bare `-restore-db` **lists the available backups and asks which to use**. `-from <backup.db>` picks a specific one; `-newest` requests the unattended newest-backup behaviour. Do not assume a bare `-restore-db` silently takes the newest.

1. Stop the **SCOMPulse** service.
2. Stop the affected **IIS site** — that site only. Preserve other applications hosted on the same server.
3. Run `PulseService.exe -restore-db`, choosing the backup, or `-from` / `-newest`.
4. The damaged database is retained under a renamed path. Leave it there until you are satisfied.
5. The restore is re-verified before it completes.
6. Start the IIS site, then the service.
7. Verify: the Settings status strip is healthy, fresh data is arriving, settings and dashboards are as expected, and the audit trail is present.
8. Confirm `action.key` is the one that matches the restored queue. If it is not, queued actions will fail closed and must be resubmitted.

Documentation gap — Recovery was exercised on a copy, not on a running installation

The commands above were run with the shipped `PulseService.exe` against a copy of a database: `-backup-db` wrote a verified backup, a deliberately damaged copy failed `-check-db` with the recovery text, `-restore-db` with no backup named and no console refused rather than guess, `-restore-db -from` kept the damaged file aside as `pulse.db.corrupt-<time>`, restored the backup and re-verified it. What was **not** exercised is the whole procedure on a live server — stopping and starting the service and the IIS site, and the site coming back with its dashboards, ownership and audit history. Rehearse that in a lab before you need it.

14.5 Upgrades

Caution — The 1.3 line cannot upgrade a 3.x preview in place

Uninstall the 3.x preview through Apps & features first, preserving the data directory, then install the current 1.3 release. Do not apply generic in-place upgrade instructions to that transition.

For any upgrade:

1. Take a verified backup (`-backup-db`) and copy it, plus `action.key`, off the server.
2. Note the current state: `PulseSetup.exe -verify`, the Settings status strip, and the configured paths.
3. Upgrade.
4. Run `PulseSetup.exe -verify` again.
5. Confirm the **web and service versions match**. They share a database and a signing key; a mismatched pair is not a supported configuration.
6. Verify as in "3 Installing and configuring SCOM Pulse".

14.6 Repair and uninstall

Repair runs through **Configure SCOM Pulse** again. A blank password retains the existing, unchanged service account — useful when you are repairing a binding rather than changing an identity.

Uninstall preserves `C:\ProgramData\SCOMPulse` by documented design.

Important — Preserved data is not a backup

Uninstall leaving the data directory behind is a convenience, not a backup strategy. It does not protect against disk loss, a bad restore, or somebody deleting the folder. Take a verified backup before uninstalling.

14.7 Troubleshooting and an escalation bundle

Symptom	Look at
Everything stale	Service state and sync times in **Settings**; the SCOMPulse event log
503, database needs attention	Integrity fault; event 30; "14 Operations, backup and recovery"
Actions queue but never finish	Service state; `action.key`; audit rows stuck at `queued`
A user is read-only unexpectedly	`/Pulse/Access` for that user; then "4 The access model"
Nobody can administer Pulse after installing	The SCOM Administrators role holds only `BUILTIN\Administrators` and Pulse is not on a management server — "3.6 The SCOM Administrators role and its default member"
Setup failed	`C:\ProgramData\SCOMPulse\setup.log`
Warehouse ranges missing	**Settings** → **Sync & data**: enabled, and the check green
The warehouse is unreachable	From the Pulse server: name resolution, the firewall, that the SQL instance is running, and the service account's login and database user — "12 The SCOM data warehouse"
The warehouse is not answering in time	Not a connection or permissions problem: Pulse signed in, and a query ran out of time. Look for blocking and load on the warehouse SQL instance; Pulse retries on its next check — "12 The SCOM data warehouse"
Windows authentication prompting	Browser and enterprise policy, and IIS — never by weakening authentication

For an escalation, gather: `PulseSetup.exe -verify` output; `setup.log`; the SCOMPulse Windows event log; a screenshot of the Settings status strip; the relevant audit rows exported as JSONL; the SCOM version and UR level; and the exact URL and time of the failure. **Never include the licence key, `action.key`, or any password.**

15 Boundaries and checklists

15.1 What still needs the Operations console

- General management pack authoring, and sealing.
- Distributed application design.
- Complex group formulas, sealed and Default-pack group edits, computer-group creation.
- **Editing** company knowledge — Pulse displays it.
- Run As **credential** creation and password rotation.
- Network device and UNIX/Linux discovery, and UNIX/Linux agent lifecycle.
- Agent upgrade and UR patching; gateway and management-server deployment.
- Connector configuration.
- Global management-group tasks beyond the panes Pulse exposes.
- SCOM and SSRS reporting.

15.2 Daily

1. Settings status strip healthy.
2. Audit filtered to **Failed or partly failed** — empty, or explained.
3. No unexpected agents in **Pending management**.
4. Warehouse summary shows no issues, and staging is draining.

15.3 Weekly

1. Review **Administration** → **Shares** — every live link still has a reason and an owner.
2. Review **Administration** → **User roles** — members and scope are everyone's access — and **Settings** → **Access**: emergency exceptions empty.
3. Check operational database and warehouse sizes against their trends.
4. Confirm backups are running and that a recent one verified.
5. Review new overrides and any new unsealed packs.
6. Confirm command channels are still the ones you expect.

15.4 Before an upgrade

1. Verified backup taken, and copied off the server, with `action.key`.
2. `PulseSetup.exe -verify` output captured.
3. Configured paths, binding and certificate recorded.
4. Service account and its SCOM membership recorded.
5. The *Upgrading* section of the *Installation Guide* read — particularly for a 3.x preview, which must be uninstalled first.
6. A rollback decision made and agreed.

15.5 Restore validation

1. Service and site started, in that order, after the restore.
2. Settings status strip: service running, access showing the SCOM user roles synced, integrity ok, licence as expected.
3. Cache counts non-zero and advancing.
4. Settings, dashboards and sidebar as expected.
5. Audit trail present, and continuing.
6. `action.key` matches; queued actions either complete or are resubmitted.
7. The damaged database preserved until the restore is accepted.

16 Glossary and configuration reference

16.1 Glossary

Term	Meaning
Action account	The SCOM account a management server or agent runs work under
`action.key`	The key that signs actions Pulse queues. Sits beside `pulse.db`
Aggregation	A warehouse dataset's Raw, Hourly or Daily rollup, each with its own retention
Configuration data set	The warehouse dataset holding entities, groups and packs the others reference
Emergency admin exception	An account or group that is a Pulse administrator whatever SCOM says — the way back in (“4.6 Emergency admin exceptions”)
Grooming	SCOM's scheduled deletion of data past its retention
Health service id	A box's `Microsoft.SystemCenter.HealthService` instance id
Management service id	A box's `Microsoft.SystemCenter.ManagementService` instance id
Profile (SCOM)	Read-only operator, Operator, Advanced operator, Author, Administrator — and, on SCOM 2022 and later, Read-only administrator and Delegated administrator
Viewer / Operator / Admin	How Pulse summarises a SCOM profile on screen: Read-only operator; Operator, Advanced operator or Author; Administrator. Not something granted separately
Resource pool	The management servers sharing a workload. System pools are SCOM's own
Sealed / unsealed	A sealed pack is read-only; overrides and groups need an unsealed one
Staging	The warehouse queue on the way into the datasets
Watcher	A website, Windows service or TCP port check Pulse authors into SCOM

16.2 Paths

	Path
Install root	`C:\Program Files\SCOM Pulse\` (`Service`, `Setup`, `Web`)
Data root	`C:\ProgramData\SCOMPulse`
Database	`C:\ProgramData\SCOMPulse\pulse.db`
Backups	`C:\ProgramData\SCOMPulse\backup\`
Audit export	`C:\ProgramData\SCOMPulse\audit\pulse-audit.jsonl`
Setup log	`C:\ProgramData\SCOMPulse\setup.log`
Signing key	`action.key`, beside the database

A configured data directory or the `PULSE_DB` environment variable overrides the database location. Confirm on your own server.

16.3 Commands

```
PulseSetup.exe -verify non-mutating install verification
PulseService.exe -backup-db [-db <path>] [-to <file.db>] verified backup
PulseService.exe -check-db [-db <path>] integrity check; clears a fault if it passes
PulseService.exe -list-backups list available backups
PulseService.exe -restore-db [-db <path>] prompts for which backup to use
PulseService.exe -restore-db -from <backup.db> restore a specific backup
PulseService.exe -restore-db -newest unattended newest-backup restore
PulseService.exe -license <key> apply a licence key
```

16.4 Application paths

Page	Path
Home / dashboards	`/Pulse/`, `/Pulse/Dashboards`
Alerts	`/Pulse/Alerts`
Discovered Inventory	`/Pulse/Objects`
Maintenance	`/Pulse/Maintenance`
Instant MM	`/Pulse/Maintenance/InstantMM`
Rules & monitors	`/Pulse/Workflows`
Overrides	`/Pulse/Overrides`
Notifications	`/Pulse/Notifications`

Page	Path
Administration	`/Pulse/Admin` (`?tab=agents`, `mps`, `notify`, `runas`, `pools`, `roles`, `groups`, `watchers`, `opsdb`, `warehouse`, `audit`, `shares`)
Management servers	`/Pulse/ManagementServers`
Settings	`/Pulse/Settings`
Access diagnostics	`/Pulse/Access`
Access status	`/Pulse/GetStarted`
Licence	`/Pulse/License`

Important — `/Pulse` is this deployment's prefix, not a product constant

The prefix is chosen when the application is configured in IIS. Substitute your own throughout, and keep it in every link and every piece of automation.