SCOM Pulse

Installation Guide

Document 1.3 · 16 September 2026

Server requirements

- SCOM Pulse installs on **one Windows server**: the IIS web tier, the `SCOMPulse` Windows service and the SQLite database all live on that machine and share local disk.
- A dedicated Windows server does not need the Operations console: Pulse includes the Operations Manager client libraries. It can also run on a management server.
- **People sign in with the SCOM user roles they already have** — there is no separate list of Pulse users to set up. One thing to check first: if your **Operations Manager Administrators** role still holds only its default member, `BUILTIN\Administrators`, either install Pulse **on a management server** or add an Active Directory group to that role. That default names the management server's own local group, so from any other server nobody can be recognised as a SCOM administrator. Setup, the **Access status** page and **Settings** → **Access** all tell you if this applies.
- **Operations Manager 2016 UR7 or later, 2019, 2022 or 2025.** Pulse includes a SCOM 2019 UR1 client; an existing compatible GAC copy can take precedence. On a server with the Operations console or a management server installed, Pulse uses that installation's SCOM client, which matches the management group. On a server with neither it uses the client it ships with. That pairing has been exercised against a **SCOM 2022** management group from a Windows Server 2025 machine with no SCOM components installed: the shipped client read the user roles, including Read-only administrator and Delegated administrator roles, identically to the SCOM 2022 client. SCOM 2016, 2019 and 2025 management groups are intended to work the same way and were not exercised for this release. A role whose permissions cannot be read grants nothing — `/Pulse/Access` says so. **System Center 2012 R2 is not supported** — the SDK differs there in the areas Pulse uses for subscriptions and Run As distribution.
- **Windows Server 2016 or later.**
- **.NET Framework 4.6.2 or later.**
- **IIS** with ASP.NET 4.x and Windows Authentication. Setup installs the IIS features it needs if they are missing.
- A **domain service account** for the service and the IIS application pool. The **SCOM Data Access Service account** is the safe choice — it is already a SCOM Administrator with database access, and setup offers it by default. Setup grants it *Log on as a service* and the file permissions it needs; you do not have to prepare either.

- **Recommended — direct SQL reads:** that same account should be able to read the **Operations Manager database** (typically `db_datareader`; the SCOM Data Access Service account usually can already). Pulse then reads from the database directly, which is faster and lighter on the management server. Without it, Pulse reads everything through the SDK and works the same. Pulse finds the database through SCOM.
- **For historic performance ranges only:** that same account needs `db_datareader` (or the `OpsMgrReader` role) on the Operations Manager **Data Warehouse** database. Pulse finds the warehouse through SCOM itself, so nobody types a server name. Skip this and everything else still works; only the 30-day, 90-day and 1-year chart ranges stay empty.

> **Caution — Not supported**
>
> Splitting the web tier and the service across two machines, putting the SQLite database on a UNC path, or running a web farm against one database. SQLite's locking assumes a local disk and a single host.

User requirements

- Any current browser — Edge, Chrome, Firefox, or Safari 15 and later. There is nothing to install on a user's machine.
- Users sign in with **Windows Authentication**; they are never asked for a password by Pulse.

Ports and firewall

Setup opens what it needs on the Pulse server itself. The rest is your network's to allow.

Direction	Port	Why
Inbound to the Pulse server	**443**, or **8443** when another site already holds 443	The website, when you install with a certificate
Inbound to the Pulse server	**80**, or **8080** when another site already holds 80	The website, when you install on plain HTTP
Outbound from the Pulse server	**5724/TCP** to the management server	The Operations Manager SDK, which is how Pulse reads and writes SCOM
Outbound from the Pulse server	**1433/TCP** (or your instance's port) to the Operations Manager database's SQL instance	Direct SQL reads. Without it Pulse reads through the SDK
Outbound from the Pulse server	**1433/TCP** (or your instance's port) to the Data Warehouse SQL instance	Historic performance ranges only. Without it live charts still work and the 30-day, 90-day and 1-year ranges stay empty

Setup adds a Windows Firewall rule only for a **non-standard** port — installing the IIS features already opens 80 and 443. The rule is named `SCOM Pulse (TCP <port>)`, and `PulseSetup.exe -remove` deletes it again.

Nothing needs to reach the Pulse server except the browsers that use it. Agents never talk to Pulse; they go on reporting to SCOM exactly as before.

Before you start

Decide these four things. Setup can change all of them later by running **Configure SCOM Pulse** again.

Service account	A domain account in **Operations Manager Administrators**. Setup suggests the SCOM Data Access Service account
Transport	A trusted certificate (recommended) or HTTP. If a suitable certificate is already in **Local Computer** → **Personal**, setup offers it
Web address	Join an existing website — users get `http://server/Pulse` with no port to remember — or give Pulse its own website
Licence	Leave the key blank and the **30-day trial** starts automatically. A key can be pasted later

Part 1 — Install and configure in one wizard

One wizard installs SCOM Pulse *and* configures it. There is no second tool to run afterwards: when it finishes, the service is running, the website is up and Pulse has connected to SCOM.

> **Note — Windows may warn that the publisher is not recognised**
>
> Windows Defender SmartScreen can show **"Windows protected your PC"** the first time a newly released installer is run on a machine. It is not a sign that anything is wrong with the file: SmartScreen trusts a download once enough machines worldwide have run that exact file without trouble, and every new release starts that count again.
>
> Choose **More info** and check that the publisher reads **AMTM Tech Inc**, then choose **Run anyway**. If it reads anything else, or is missing, stop and check where the file came from — a genuine SCOM Pulse installer is always signed. Two checks confirm a download independently of SmartScreen: right-click the MSI and look at **Properties** → **Digital Signatures**, which names the publisher and shows the signature as valid with a timestamp; and compare the file's SHA-256 with the checksum published beside the download — in PowerShell, `Get-FileHash .\SCOMPulse-<version>.msi`.
>
> Some organisations configure SmartScreen so that the warning cannot be dismissed. Where that applies, your security team can have the file assessed by submitting it at <https://www.microsoft.com/en-us/wdsi/filesubmission>, or distribute it from a trusted intranet location, which SmartScreen does not check.

1. Run `SCOMPulse-<version>.msi` as an administrator. Accept the licence — the checkbox starts ticked — and either accept `C:\Program Files\SCOM Pulse` as the installation folder or choose another.

2. Fill in Connect to SCOM.

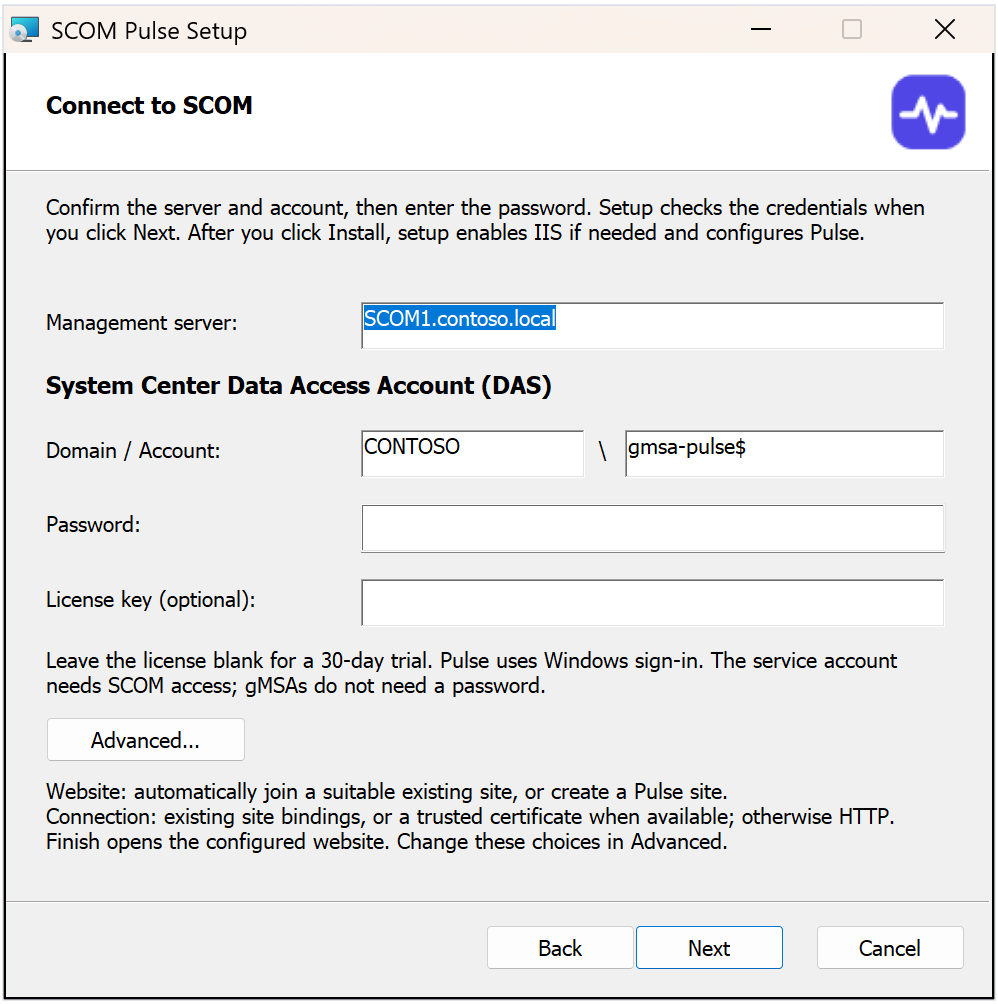


Figure 1 The **Connect to SCOM** page, with the management server and the Data Access account already filled in from this computer. Underneath, the page names the website and connection it will choose unless **Advanced** says otherwise.

- **Management server** and the account under **System Center Data Access Account (DAS)** arrive already filled

in where setup can read them from this computer: the management server it points at, and the account the System Center Data Access Service runs as — the account that already holds the SCOM access Pulse needs. When both are found the wizard opens with the cursor in **Password**; when they are not, it opens on **Management server** and the domain box holds this computer's own domain. Tabbing runs server → domain → account → password.

- **Password** is the service account's. A **gMSA** — an account name ending in $ — needs none: Windows supplies it.
- **Licence key** is optional. Left blank, the **30-day trial** starts by itself, and a key can be pasted in later.
- The lines under **Advanced** say what setup will do: which website it will use, how the browser will reach it, and

that **Finish** opens the site. They change as soon as you change anything in **Advanced**.

3. Use Advanced only if the defaults do not suit you.

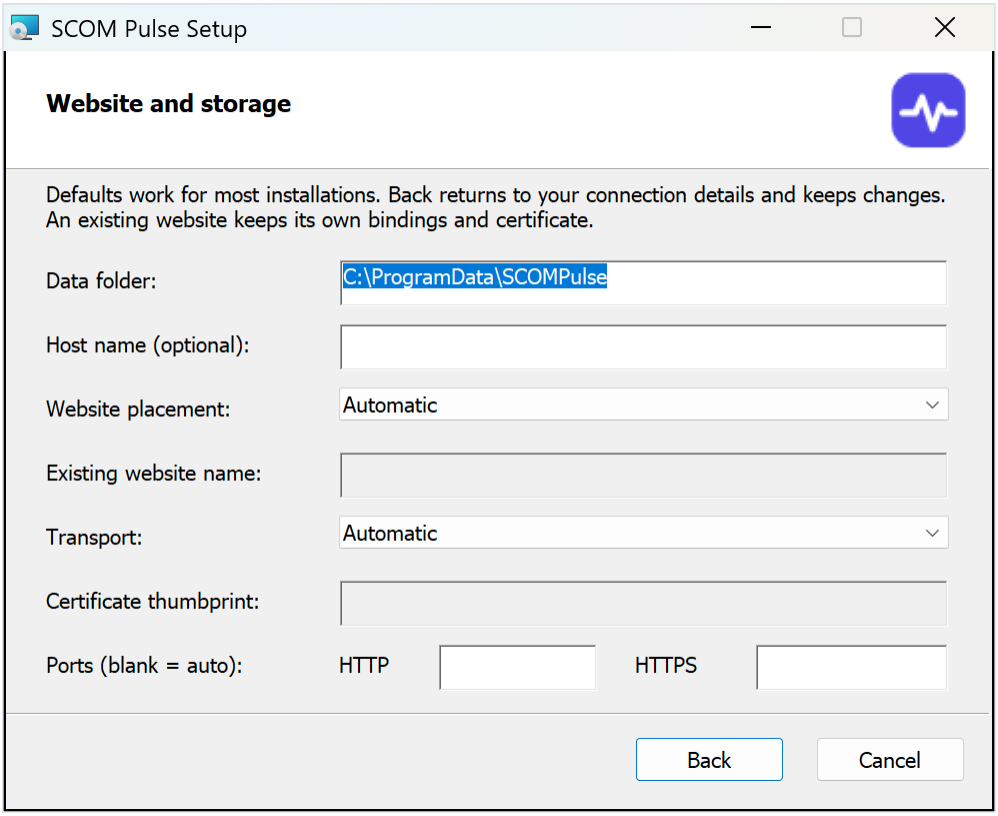


Figure 2 Advanced — storage and the website. The defaults suit most installations; **Back** returns to **Connect to SCOM** and keeps whatever you changed.

- **Data folder** holds the database, the signing key, the logs and the backups. `C:\ProgramData\SCOMPulse` by default.
- **Host name** is the name people will type, when it is not this server's own name.
- **Website placement** — *Automatic* joins a suitable existing website as `/Pulse`, or creates a Pulse site when

there is none. *Separate* always creates its own site. *Join* uses the site you name in **Existing website name**.

- **Transport** — *Automatic* uses an existing site's own bindings, or a trusted certificate for this host if one is

already in **Local Computer** → **Personal**, and otherwise HTTP. *HTTPS* uses the certificate whose thumbprint you give. *HTTP* is plain HTTP. A website you join keeps its own bindings and certificate, so the certificate and placement boxes grey out when they cannot apply.

- **Ports** left blank are chosen for you.

4. Click Next. Setup checks the details before anything is installed. Anything it cannot accept comes back straight away, naming what to fix — the wizard stays where it is, with what you typed still in place.

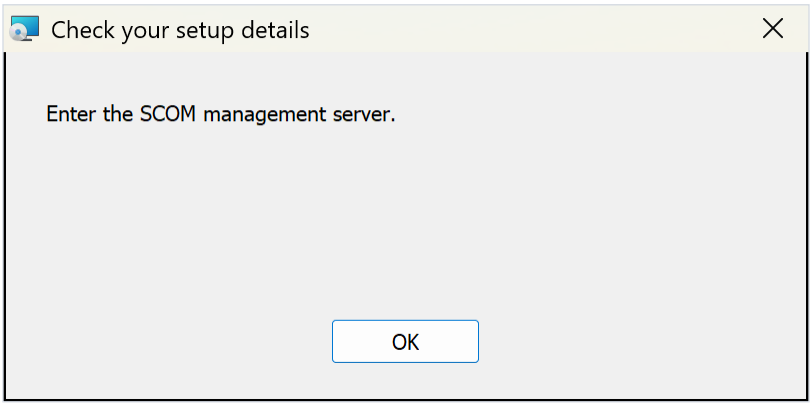


Figure 3 A detail setup cannot use is reported before installation starts, in its own window.

Caution — An empty Default Web Site can lose port 80

Where Pulse creates its own website and port 80 is held only by an **empty Default Web Site** — the one carrying nothing but the IIS welcome page — setup takes the port: it removes that site's `:80` binding, stops the site and turns off its autostart, so Pulse answers on `http://server/` with no port to remember. `setup.log` records it and how to undo it: add the binding back and start the site in IIS Manager. A site that is serving anything is never touched; to keep the default site as it is, choose **Join** in **Advanced** and name a site, or give Pulse its own port under **Ports**.

5. Click Install on Ready to install. This is the first point at which anything on the server changes. Setup then, in order: installs any missing IIS components, creates the data folder, sets up the service account and the website, starts the service and waits until Pulse has reached SCOM. The wizard shows the stage it is on and the elapsed time throughout. Installing IIS is the slow part — Windows can take several minutes over it, and the bar moves on an estimate, labelled as one, rather than pretending to measure it.

6. Leave Open SCOM Pulse ticked and click Finish. The browser opens on the new site, at the address setup chose.

> **Note — If configuration fails**
>
> The installation fails as a whole and takes back what it created: the Pulse service, its application pool and its website. What it did *not* create, it leaves — IIS components it installed, the data folder, the database and `setup.log` all stay, so you can read the log, fix the cause and run the MSI again. Nothing half-configured is left behind, and a retry starts from a clean slate.

> **Note — Installing without the wizard**
>
> See "Unattended installation" for a single `msiexec` command that installs and configures silently.

Part 2 — Change configuration or repair later

The separate **Configure SCOM Pulse** tool remains available in the Start Menu for later changes and repair. The following controls describe that tool, not an additional required first-install step.

Configure SCOM Pulse needs administrative rights; it elevates when you start it.

1. Start **Configure SCOM Pulse** from the Start Menu.
2. **Management server** — the name of a SCOM management server. Setup detects one; change it if you want a different one.
3. **Account** — the service account, as `DOMAIN\user`. Setup pre-fills the SCOM Data Access Service's own logon.
4. **Password** and **Confirm password** — used once, to register the service. **Setup does not store the password** in the configuration.
5. **Transport** — pick a certificate from the list for HTTPS, or **HTTP**. HTTP uses port 80, or 8080 when another site already owns 80.
6. **Web address** — this row appears only when a real website (typically the SCOM Web Console) already owns the standard port. Joining that site puts Pulse at **`/Pulse`** on the port and certificate it already has, and the Web Console keeps working.
7. **License key** — paste a key, or leave it blank for the 30-day trial.
8. Click **Configure**.

> **Note — A group managed service account needs no password**
>
> For a gMSA (`DOMAIN\name$`), leave **Password** blank — Windows supplies it. On a repair, leaving the password blank keeps the account's current password unchanged.

Optional — Advanced

Click **Advanced** □ only if you need one of these:

Field	What it does
Data folder	Where the database, backups, audit export and `action.key` live. Local disk only. Default `C:\ProgramData\SCOMPulse`
Host name	The name used in the URL and on the certificate. Blank = this computer
Lab certificate (self-signed, 3 years)	Labs only. Browsers will show "*Your connection is not private*"
Use this certificate	Any certificate in **Local Computer** → **Personal**, including ones that do not name this host
HTTPS port	443, or 8443 if 443 is already in use
Use the built-in app-pool identity	Off = the pool runs as the service account. Off is recommended
Do not start the service yet	The service stays stopped until you start it in Services

> **Caution — Host name and Windows Authentication**
>
> On HTTP the URL uses this computer's **short name**, which keeps the site in the browser's Local Intranet zone. A dotted HTTP name puts it outside that zone and browsers start prompting for a password. A custom name needs DNS and, for Windows Authentication, an `HTTP/<name>` SPN on this computer's AD object.

What Configure does

Configure works through thirteen steps and shows the result of each. This is also what you see if a step fails, so it is worth knowing the order.

#	Step	What it means when it succeeds
1	Preflight	This computer can run SCOM Pulse

#	Step	What it means when it succeeds
2	Validate service account	Account and password accepted
3	IIS features	IIS, ASP.NET and Windows Authentication installed
4	Log on as a service	Service account allowed to log on as a service
5	Data folder and permissions	Data folder created and permissions set
6	Action signing key (`action.key`)	Action signing key created
7	Write configuration files	Configuration written
8	Windows service	Windows service registered
9	IIS site and application pool	IIS site created, or joined the existing website as `/Pulse`
10	HTTPS binding	HTTPS configured
11	Database and trial	Database created — licence: trial
12	Start service and connect to SCOM	Connected to management group *(your group name)*, followed by the database access results
13	Open the site	The site answered

Everything is written to **`C:\ProgramData\SCOMPulse\setup.log`**. If a step fails, that file is the first place to look.

> **Note — Step 12 also checks database access**
>
> Once the service is connected, it tests, as the service account, whether it can read the **Operations Manager database** and the **data warehouse**, and step 12 reports what it found — for example "*Operations Manager SQL: accessible as …*", naming the account and the database. Pulse turns direct SQL reads on when the Operations Manager database answers and the warehouse on when that answers. Neither result fails the installation: without the database Pulse reads through the SDK, and without the warehouse the long chart ranges stay empty. If the checks have not finished within a minute, step 12 says they are still pending; the results then appear under **Settings** → **Sync & data**.

> **Important — Protect `action.key`**
>
> `action.key` sits beside the database and signs the actions Pulse queues. Keep it with your backups. A different or regenerated key makes already-queued actions fail closed, and they have to be resubmitted.

Part 3 — Verify the installation

1. On the server, open a command prompt in `C:\Program Files\SCOM Pulse\Setup` and run:

```
PulseSetup.exe -verify
```

This walks the same thirteen steps and **changes nothing** — every step reports as already done or skipped. It is the authority on which version is installed on the server; inside Pulse, **Settings** → **Service & storage** shows the website's version and whether the service reports the same one.

2. Check the service is running: **SCOM Pulse Service** (service name `SCOMPulse`) should be **Running**, **Automatic (Delayed Start)**, under your service account.
3. Open the site. The Start Menu shortcut **SCOM Pulse** opens the address the last successful Configure recorded.
4. **Sign in.** There is no welcome screen and nothing to finish: an authorized person lands on **Operations Overview**, or their own home dashboard. Access is the SCOM user roles the account already has — see **Part 4**. Somebody whose roles give no monitoring access — or who signs in before the service has synchronized the roles — sees an **Access status** page instead, saying which of the two it is, with a **View my access** link to the evidence behind the decision.

> **Important — If the page says nobody can be verified as a SCOM administrator**
>
> Your Operations Manager Administrators role holds only `BUILTIN\Administrators` and this server is not a management server. Add an Active Directory group to the role (Operations console → **Administration** → **Security** → **User Roles** → **Operations Manager Administrators**), sign out of Windows and back in, and refresh. To get in before that change is made, see **Locked out** under Troubleshooting.

5. Open **Settings**. The status strip across the top is the fastest health check in the product.

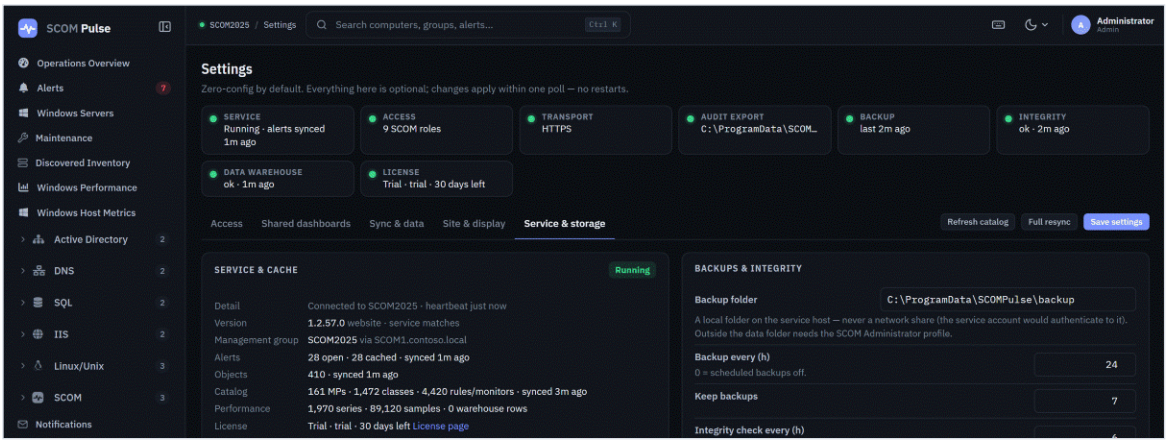


Figure 4 The Settings status strip. **SERVICE Running**, a recent sync, **INTEGRITY ok**, and a recent **BACKUP** is a healthy install.

6. Open **Settings** → **Sync & data**. The **Operations Manager data source** card shows the result of the database access check that Configure asked for, and whether **Direct SQL reads** is on. If it reports the database unavailable, Pulse is working through the SDK; fix the service account's SQL access and click **Test database access**, then set **Direct SQL reads** to **On** and save. The *Administration Guide* describes the card in full.

> **Caution — Check the TRANSPORT tile**
>
> If it reads "*plain HTTP — not encrypted*", Windows Authentication is being negotiated in clear text. Run **Configure SCOM Pulse** again and pick a certificate.

Sign people in without a password box

Pulse signs people in with their Windows logon, so nobody needs to type a password into it. Whether that happens silently is decided **on each PC**, not on the server: Edge and Chrome reliably hand a site the user's Windows credentials only when a policy on the PC names that site; without one they go by whether Windows treats the site as intranet. Without the policy the browser usually shows its grey **Sign in** box — and typing the right password into it often does not help.

This affects almost every HTTPS install. A short name such as `http://scom01/` is usually treated as intranet by Windows on its own; a full name such as `https://scom01.contoso.com/` — which every certificate needs — never is.

The installer's last page points here. **Settings** → **Access** → **Signing in without a password box** shows the exact settings for your address, with **Copy all settings**; so does the last screen of **Configure SCOM Pulse** when you run it again later. Push them by group policy:

Browser	Group policy	Value
Microsoft Edge	Administrative Templates → Microsoft Edge → HTTP authentication → **Configure list of allowed authentication servers**	the site's names, comma-separated — for example `scom01.contoso.com,scom01`
Google Chrome	Administrative Templates → Google Chrome → HTTP authentication → **Authentication server allowlist**	the same names
Windows	Administrative Templates → Windows Components → Internet Explorer → Internet Control Panel → Security Page → **Site to Zone Assignment List**	value name `https://scom01.contoso.com`, value `1` (Local intranet)
Firefox	`network.negotiate-auth.trusted-uris` and `network.automatic-ntlm-auth.trusted-uris` (`about:config` or `policies.json`)	the same names

> **Important — The browser policy is the one that decides**
>
> Once this allowlist policy is set, it is what Edge and Chrome go by. Without it they go by the Windows zone, and a site can sit in the Local intranet zone and still get a password box. Set both — the zone entry costs nothing and older tools read it. If your organisation already pushes an allowlist, **add** these names to it rather than replacing it.

After the policy reaches a PC, close every Edge or Chrome window once: a browser keeps a failed sign-in until it restarts. `edge://policy` (or `chrome://policy`) shows whether the policy arrived. To try it on one PC before the policy is in place, copy `enable-sso-client.ps1` from `C:\Program Files\SCOM Pulse\Support` on the server to that PC and run `.\enable-sso-client.ps1 -Server <name>` from an elevated prompt there. It sets both, merges into any allowlist the PC already has, and `-Remove` takes it out again.

Macs and PCs that cannot sign in with Windows. An HTTPS install can also offer a sign-in page — a preview, off by default: a user name and password form that password managers can fill, and browsers that stay signed in. Turn it on with the installer's **Advanced** checkbox, `PULSESIGNIN=1` on a silent install, or `PulseSetup.exe -signInPage` later; once it is on, **Settings** → **Access** → **Browser sign-in** switches it off and back on. PCs that have the policy above still sign in silently.

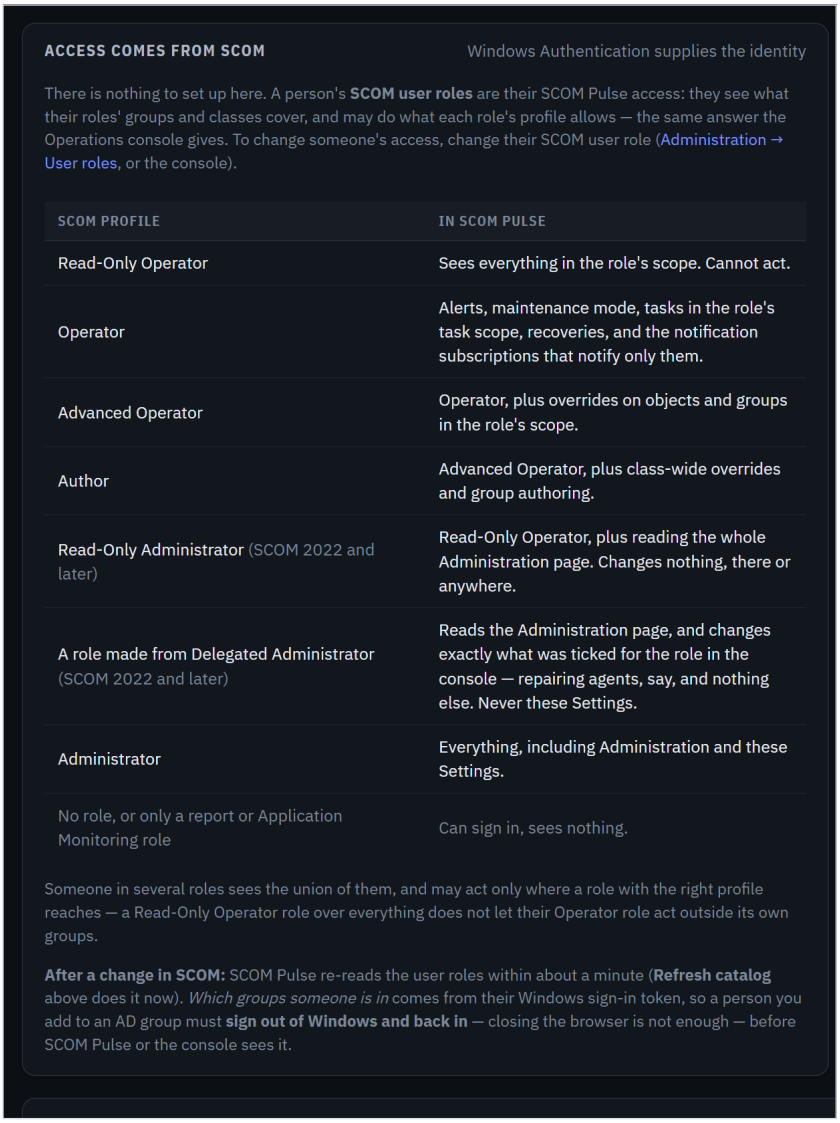
Part 4 — Who can use it

There is nothing to configure. Your existing SCOM user roles determine access to supported SCOM operations in Pulse — a person sees what their roles cover and may do what each role allows, the same

answer the Operations console gives them. It works the same on SCOM 2016, 2019, 2022 and 2025; there is no compatibility mode to select.

SCOM profile	In SCOM Pulse
Read-only operator	Sees everything in the role's scope. Cannot act
Operator	Alerts, maintenance mode, tasks, diagnostics and recoveries, and the notification subscriptions that notify only them
Advanced operator	Operator, plus overrides on objects and groups
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
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. Never **Settings**
Administrator	Everything, including **Administration** and **Settings**
No SCOM user role, or only a report or Application Monitoring role	Can sign in. Sees nothing, and is told why

To give somebody access, add them to a SCOM user role — in the Operations console, or in Pulse under **Administration → User roles**. Permissions are per role and are never added together: a role acts only on what that same role covers. The *Administration Guide*, chapter 4, has the detail, including the few places where Pulse asks for the Administrator profile on purpose.



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 5 Settings → Access says the same thing, and shows the SCOM user roles Pulse has read.

> **Important — Two delays after a change**
>
> Pulse re-reads the SCOM user roles within about a minute. But membership of an AD group comes from the Windows logon token: somebody you add to a group must **sign out of Windows and back in** before Pulse — or the console — sees it; closing the browser is not enough. `/Pulse/Access` shows any signed-in person the SCOM roles their sign-in matched, what each of those roles grants, and every test behind it.

Optional — Import the SCOM Pulse management pack

A management pack ships **in the box**, at `C:\Program Files\SCOM Pulse\ManagementPacks`, beside an `IMPORT.txt` that walks through the import.

Setup does not import it. An import changes your management group, so it stays your decision, and Pulse runs perfectly well without it.

What it adds is SCOM-side monitoring **of Pulse itself**: installed-product discovery, an alert when the service stays down, a bounded restart attempt, and warnings before the licence expires.

1. Open the **Operations console** → **Administration** → **Management Packs**.
2. Right-click and choose **Import Management Packs…**, then **Add** → **Add from disk…**.
3. Browse to `C:\Program Files\SCOM Pulse\ManagementPacks` and select the file you find there.
4. Click **Install**, then **Close**.

> **Note — Exactly one file is laid down**
>
> The sealed `.mp` when the build had one, and the unsealed `.xml` otherwise — so there is never a question of which to import. The difference matters for tuning: SCOM stores an override *inside* a management pack, so overrides against an **unsealed** pack live in ours and the next version overwrites them, while overrides against a **sealed** one have to go in a pack of your own and therefore survive an upgrade.

> **Important — Uninstalling Pulse does not remove the management pack**
>
> Uninstall does not reach into your management group. If you no longer want it, delete it from the Operations console yourself.

Part 5 — Licence

The 30-day trial starts by itself and is the full product. To apply a key:

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

It 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. From a console: `PulseService.exe -license <key>`.

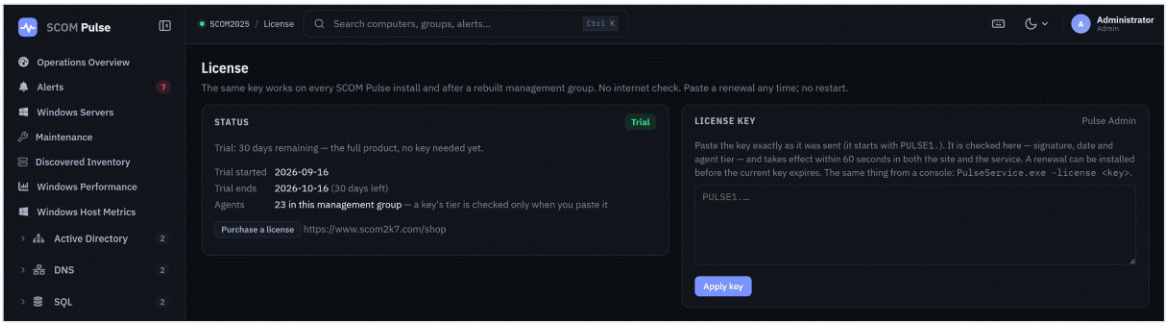


Figure 6 The **License** page during the trial. No key has been applied here.

> **Caution — Never put a licence key in a ticket, document or screenshot**

Unattended installation

For a fresh silent installation, supply MSI properties `PULSESERVER`, `PULSEACCOUNT` and `PULSEPASSWORD` through your deployment system. Use its protected secret mechanism for the password; do not put passwords into shared command files or tickets. A gMSA needs no password:

```
msiexec /i SCOMPulse-<version>.msi /qn PULSESERVER=SCOM1.contoso.local
PULSEACCOUNT="CONTOSO\gmsa-pulse$"
```

> **Caution — A password on the command line is visible while setup runs**
>
> `PULSEPASSWORD` is redacted in the Windows Installer log, but the full `msiexec` command line — the password included — can be read by anyone who can list processes on that server for as long as the installation runs. That is why the example above uses a gMSA. Otherwise, prefer the wizard — it takes the password in a dialog and never puts it on a command line — or run the silent install from a session on a server nobody else is signed in to.

Optional MSI properties: `PULSEDATADIR`, `PULSEHOST`, `PULSELICENSE`, `PULSEPLACEMENT` (`auto`, `own`, `join`), `PULSEJOINSITE`, `PULSETRANSPORT` (`auto`, `http`, `cert`), `PULSETHUMBPRINT`, `PULSEHTTPPORT` and `PULSEHTTPSPORT`. New installs validate required details and configure automatically. Existing MSI upgrades preserve their configuration without resupplying these properties.

Quote any value that contains a space. A website name usually does — `PULSEJOINSITE="Default Web Site"`. Unquoted, `msiexec` reads the words after the first as separate arguments and can sit indefinitely without writing a log:

```
msiexec /i SCOMPulse-<version>.msi /qn PULSESERVER=SCOM1.contoso.local
PULSEACCOUNT="CONTOSO\gmsa-pulse$" PULSEPLACEMENT=join PULSEJOINSITE="Default Web Site"
```

For later repair or configuration, `PulseSetup.exe -quiet` remains available. Quote a value with a space here too (`-joinSite "Default Web Site"`). Its switches are:

Switch	Meaning
`-account DOMAIN\user`	Service and app-pool account. Required
`-password <pw>` / `-passwordPrompt`	The password, or prompt for it without echoing. Not needed for a gMSA
`-server <name>`	SCOM management server. Default: auto-detected
`-thumbprint <hex>`	Use this certificate from Local Computer → Personal
`-allowHttp`	HTTP on port 80 (8080 when another site owns 80). Not encrypted
`-selfSigned`	Create or reuse a lab certificate. Browsers will warn — labs only
`-joinSite [name]`	Become `/Pulse` under an existing website. No name = Default Web Site
`-ownSite`	Always Pulse's own website, even if that means port 8080
`-hostName <fqdn>`	Name in the URL and on the certificate
`-httpsPort <n>` / `-httpPort <n>`	Override the ports
`-dataDir <path>`	Data folder. Local disk only
`-licenseKey <key>`	Apply a key during setup
`-noStart`	Do not start the service
`-noSaveCredential`	Do not save the service password for unattended upgrades
`-force`	Continue even if the account cannot be validated
`-grantEmergency DOMAIN\name`	Locked out: makes that account or group a Pulse administrator whatever SCOM says. Audited

Exit codes: `0` = success; `10 + step number` identifies the step that failed — so `21` is step 11, *Database and trial*.

> **Note — Reading the exit code from PowerShell**
>
> `PulseSetup.exe` is a windowed program. Use `Start-Process -Wait -PassThru` to get its exit code.

Upgrading

> **Caution — A 3.x preview cannot be upgraded in place**
>
> Uninstall the preview through **Apps & features** first — it leaves `C:\ProgramData\SCOMPulse` in place by design — then install the current 1.3 release. Do not apply the in-place upgrade steps below to that transition.

For a normal upgrade:

1. Take a verified backup and copy it, with `action.key`, off the server:

```
"C:\Program Files\SCOM Pulse\Service\PulseService.exe" -backup-db
```

2. Record the current state: `PulseSetup.exe -verify`, and a screenshot of the Settings status strip.
3. Run the new MSI. It stops the application pool and the service, replaces the files, and carries the service registration over.
4. On the last page, one of three things:
   - If the MSI could carry the service password over, the page offers **Open SCOM Pulse** and says the service and website were restarted with the settings you had. The database migrates on the service's first start.
   - If it could not, which means `-noSaveCredential` or a box first configured by a build that saved no credential, the page runs **Configure** so you enter the password once. Later upgrades will not ask again.
   - Upgrading from **1.3.2 or earlier**, the page says neither: those versions left the wizard nothing to go on. It confirms that your settings, database and website are kept, and tells you to run **Configure SCOM Pulse** once if the website does not answer. Nothing is ticked, so Finish just closes the wizard. This upgrade records the answer, so the next one is back to one of the two above.
5. Run `PulseSetup.exe -verify` and confirm the Settings status strip is healthy.

> **Important — Keep the web and service versions matched**
>
> They share a database and a signing key. A mismatched pair is not a supported configuration.

Repair and uninstall

Repair — run **Configure SCOM Pulse** again. Leaving the password blank keeps the existing service account and its current password, which is what you want when you are repairing a binding rather than changing an identity.

Remove the website only — `PulseSetup.exe -remove` removes the IIS site, application pool and firewall rules, and keeps the database, `action.key`, licence and settings.

Uninstall — through **Apps & features**. `C:\ProgramData\SCOMPulse` is preserved by design.

> **Important — Preserved data is not a backup**
>
> Uninstall leaving the data folder behind is a convenience. It does not protect you against disk loss or somebody deleting the folder. Take a verified backup first.

If setup fails

Message	What to do
Right-click PulseSetup → Run as administrator.	Setup needs elevation
SCOM Pulse requires Windows Server 2016 or later (this is ...)	Wrong operating system
Included Operations Manager client libraries could not be loaded.	Repair or reinstall Pulse to restore its included client libraries
PulseService.exe was not found at ... / The website was not found at ...	The MSI did not complete. Re-run it
Step 2 fails on the account	Check `DOMAIN\user` and the password. `-force` continues past validation, but only use it when you know the account is right
Step 12 fails to connect to SCOM	The service account is not a SCOM Administrator, or the management server name is wrong
The site prompts for a password	The PC's browser has no policy naming the site — almost always the case for an HTTPS (full) host name. See **Sign people in without a password box** in Part 3. If the server itself is in doubt, run `test-auth.ps1` from the install's `Support` folder (`C:\Program Files\SCOM Pulse\Support`)

Message	What to do
Locked out — nobody can administer Pulse	Access comes from SCOM user roles, so the usual fix is in SCOM: put an Active Directory group in the Operations Manager Administrators role. To get in meanwhile, open an elevated command prompt on the server in `C:\Program Files\SCOM Pulse\Setup` and run `PulseSetup.exe -grantEmergency DOMAIN\name`. It makes that one account a Pulse administrator, is audited, and is removed again under **Settings → Access → Advanced**

Attach **`C:\ProgramData\SCOMPulse\setup.log`** to any support request. **Never** include the licence key, `action.key` or any password.

Where to go next

- **SCOM Pulse Quick Start** — the essentials for operators and administrators, on two pages.
- **SCOM Pulse User Guide** — alert triage, objects and health, performance, maintenance, tasks, overrides, dashboards and notifications.
- **SCOM Pulse Administration Guide** — the access model in full, every Settings tab, agents, management packs, the operational database and data warehouse, sharing and audit, backup and recovery.