



SQL Data Collector

User Guide & Documentation

ALERTLINK TECHNOLOGIES

Monitoring for the SQL Server estate

1. Introduction

What is SQL Data Collector?

SQL Data Collector is a monitoring tool for teams that operate more than a handful of Microsoft SQL Server instances. Rather than logging into each server to check whether backups ran, whether a drive is filling up, or whether a server is behind on patches, you point SQL Data Collector at the estate once and it keeps a continuously-updated picture for you.

It is built as a single **.NET 8** application that ships as a Windows **MSI**. The MSI installs a **Windows Service** that does two jobs at the same time:

- **The collector** — a background worker that connects to each monitored SQL Server on a schedule and gathers the data it needs (it does *not* require any agent or software to be installed on the monitored servers).
- **The web interface** — a local website (Blazor Server) that presents everything the collector has gathered.

The application keeps its own small database (created automatically) on a SQL Server of your choosing. Everything it collects is stored there.

Who is it for?

- **DBA teams** who need a single pane of glass across many SQL Servers.
- **Infrastructure / operations** staff who care about drive space, patch levels, and uptime.
- **Security and compliance** reviewers who need an auditable record of who has access to what, and a way to certify that access periodically (the Permissions Audit and attestation features).

What it monitors

Area	What you get
Backups	Per-database backup freshness (full / differential / log), timeliness rings, and a fleet roll-up.
Restore points	How far back you could recover each database, and what's standing in the way (RPO analysis).

Area	What you get
Backup trends	How backup sizes and durations change over time.
Database sizes	Per-database size history and growth.
Drive space	True volume capacity and free space (via WMI), fill/free rates, and runway forecasting.
Capacity changes	An audit trail of when a drive's total capacity changed.
C: drive files	Warns when database files sit on the OS drive.
SQL versions	Build/patch level per server, with end-of-life dates.
SQL Agent jobs	Failed and flaky jobs across the fleet.
Failure history	A record of backup job failures over time.
Permissions audit	A point-in-time and historical record of SQL permissions and role membership, with change tracking, review, and sealed attestation.
Security checks	A set of configuration/security posture checks per server.

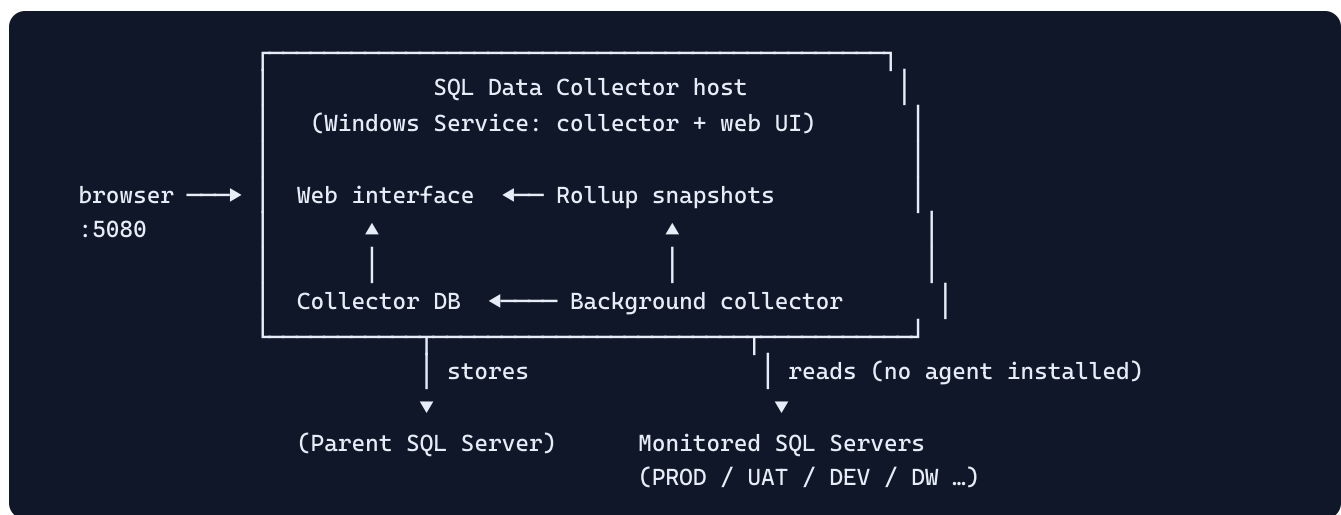
Key concepts

A few terms appear throughout the app and this guide:

- **Parent server / collector database** — the SQL Server that hosts SQL Data Collector's *own* database (where it stores everything it collects). This is separate from the servers you monitor.
- **Monitored server** — a SQL Server you have added for monitoring. Each one can be enabled/disabled and has its own connection settings.
- **Collection pass** — one cycle in which the collector connects to each enabled server and refreshes its data. Passes run on a configurable interval (default every 15 minutes).
- **Rollup snapshot** — to keep the web interface fast, the headline numbers (e.g. the Overview cards) are pre-computed on a short interval and served to every browser from the same snapshot.
- **Dashboard** — a viewer page in the left sidebar. Access to each dashboard can be granted per user.
- **Monitor** — a named collection activity (backups, drive space, SQL versions, ...). Individual monitors can be excluded per server, and their health is tracked on the Collection Health

page.

How collection works (at a glance)



The collector reads from system views, `msdb`, and WMI on each monitored host. Because it pulls data remotely, the only prerequisite on a monitored server is a login with enough rights to read the relevant metadata (and, for true drive capacity, remote WMI access on the host).

Continue to [2. Installation & first-run setup »](#)

2. Installation & first-run setup

This chapter walks through installing SQL Data Collector and completing the one-time setup wizard.

2.1 Prerequisites

- A **Windows** host (Windows Server or Windows 10/11) to run the service. It does not have to be a SQL Server itself.
- Network access from that host to:
 - a **SQL Server** that will host the collector's own database (the *parent* server — this can be one of your existing servers), and
 - the **SQL Servers you want to monitor**.
- A login that the service can use to reach the parent server and the monitored servers. This can be the **service account** (Windows authentication) or a **SQL login** — configurable per server.
- For true drive-capacity figures, the service account needs **remote WMI access** on each monitored host. (Without it, the app still reports the volumes that host SQL data/log files.)

2.2 Install the MSI

1. Run the installer, `SQL Data Collector.msi`.
2. Accept the licence agreement (EULA) and choose an install location.
3. The installer registers and starts a Windows Service:
 - **Service name:** `SQLDataCollector3`
 - **Display name:** `SQLDataCollector 3`
4. By default the web interface listens on `http://localhost:5080`.

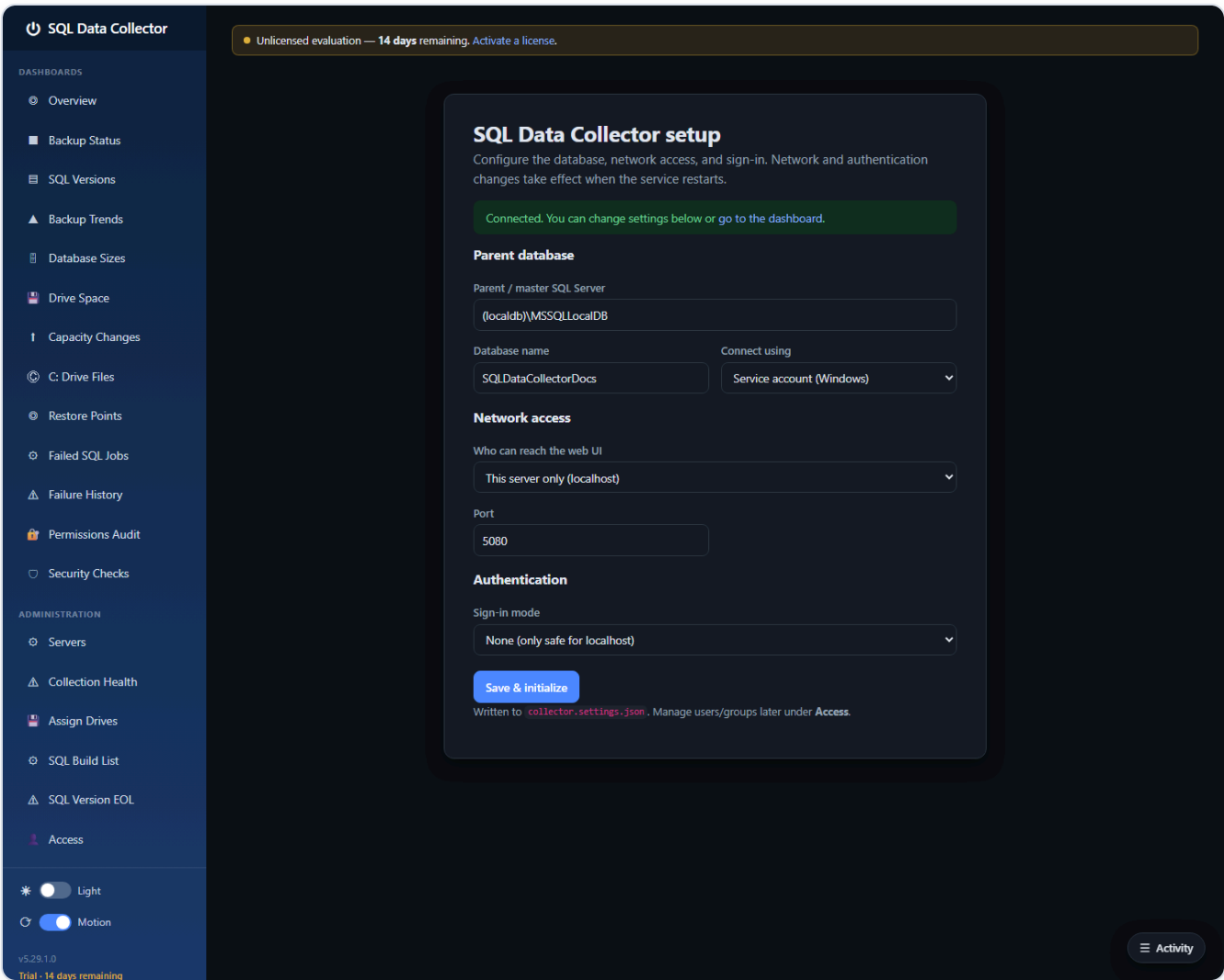
You can confirm the service is running from *Services* (`services.msc`) or with PowerShell:

```
Get-Service SQLDataCollector3
```

Tip — checking the version. The running build is shown in the bottom-left of the navigation sidebar (for example `v5.29.1.0`). After any upgrade, confirm the sidebar shows the new version so you know the new build is the one actually running.

2.3 First-run setup wizard

The first time you browse to the web interface (or any time the database isn't yet configured), you're taken to the **Setup** page. It has three short sections.



Parent database

Field	What to enter
Parent / master SQL Server	The SQL Server that will host the collector's database, e.g. <code>SQLHOST</code> or <code>SQLHOST\INSTANCE</code> .
Database name	The name for the collector's database (created automatically if it doesn't exist), e.g. <code>SQLDataCollector</code> .
Connect using	<i>Service account (Windows)</i> uses the service's own identity; <i>SQL login</i> lets you supply a username and password.

Network access

Field	What to enter
Who can reach the web UI	<i>This server only (localhost)</i> keeps the interface private to the host; <i>Reachable on the LAN</i> binds all interfaces so colleagues can browse to it.
Port	The TCP port for the web interface (default <code>5080</code>).

Authentication

Mode	Use when
None	Only safe for localhost-only access — no sign-in is required.
Windows / Active Directory	Users sign in with their AD username and password (validated by an LDAP bind, so any browser works). You nominate an initial admin user or group.
Local accounts	Accounts are stored in the collector database. You create an initial admin username and password.

Click **Save & initialize**. The app creates the database if needed, applies its schema, and — if you changed the network or authentication settings — prompts you to **restart the service** so they take effect.

Network and authentication changes always require a service restart; database changes do not.

2.4 After setup

Once the database is ready you land on the **Overview** page. It will be largely empty until you add servers and the first collection pass runs. Head to **Administration** → **Servers** to add your first monitored server (see [3. User guide](#) and [4. Administration](#)).

2.5 Where settings live

Operator-editable settings are stored in `collector.settings.json` next to the installed service executable. Most can also be changed from the in-app **Settings** page (admins only). Changes to the file take effect the next time the service restarts. Notable settings include the collection interval, parallelism, logging destinations, drive-space cadence/retention, and the credential-encryption key that protects stored SQL passwords.

Keep a backup of the credential-encryption key. It is generated on first run and stored in `collector.settings.json`. Without it, stored SQL-login passwords cannot be decrypted.

Continue to [3. User guide — the dashboards »](#)

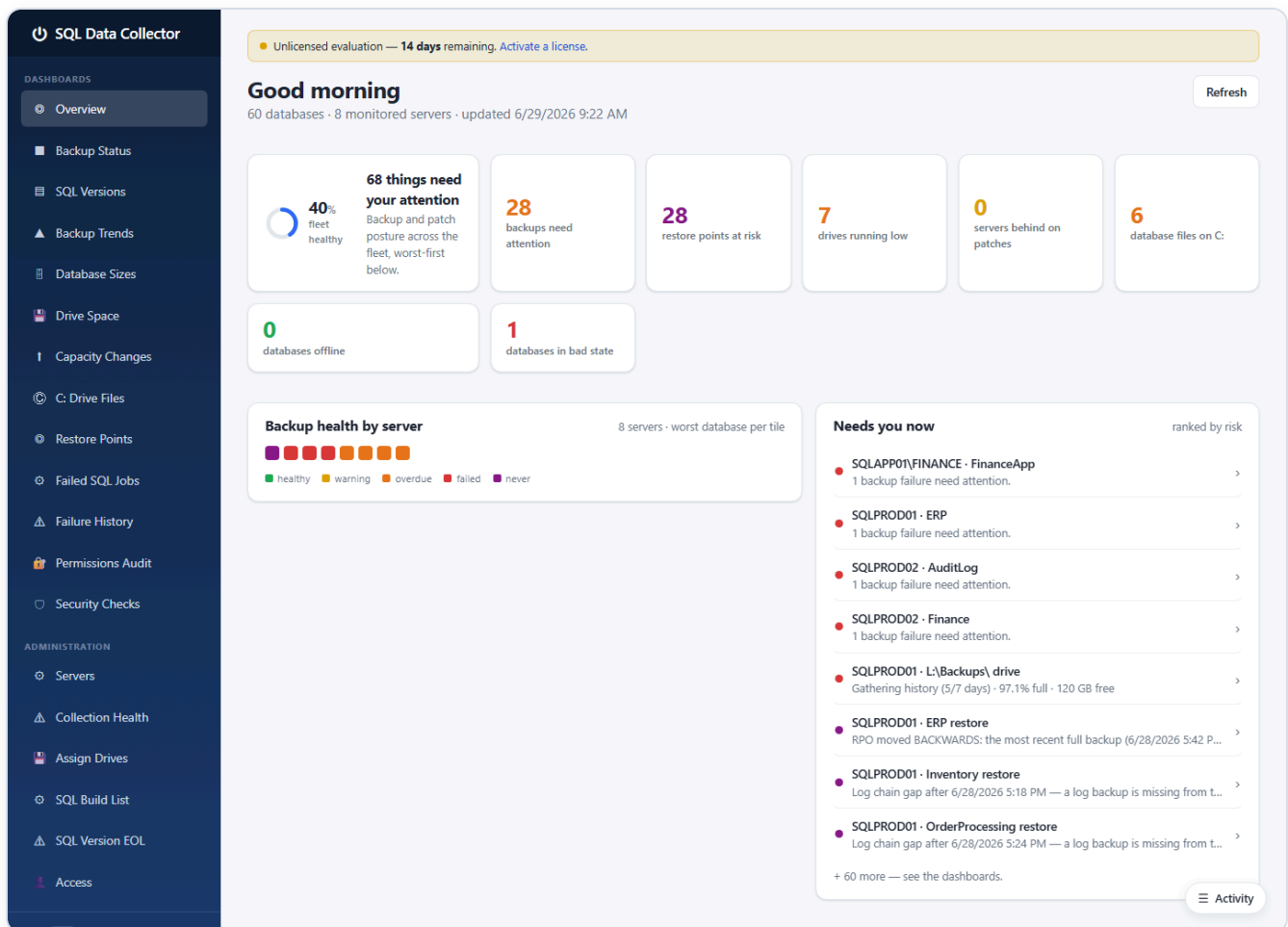
3. User guide — the dashboards

This chapter walks through every viewer dashboard in the left sidebar, in order. Each one answers a specific operational question. Throughout, the status colours are consistent: **green** = healthy, **amber** = warning, **red** = problem, **purple** = broken/missing, **grey** = not applicable.

The screenshots in this chapter come from a demonstration fleet of eight servers (SQLPROD01, SQLPROD02, SQLPROD-CLUSTER\NODE1, SQLUAT01, SQLDEV01, SQLDW01, SQLAPP01\FINANCE, SQLBI01) with representative data.

3.1 Overview

The Overview answers: "Is anything wrong, and what should I look at first?"



The top of the page is a row of **headline cards**:

- A **fleet-health ring** with the percentage of databases that are completely healthy and a count of how many things need attention.
- **Backups need attention, restore points at risk, drives running low, servers behind on patches**, and **database files on C:** — each links straight to the relevant dashboard.
- **Databases offline** and **databases in bad state**.

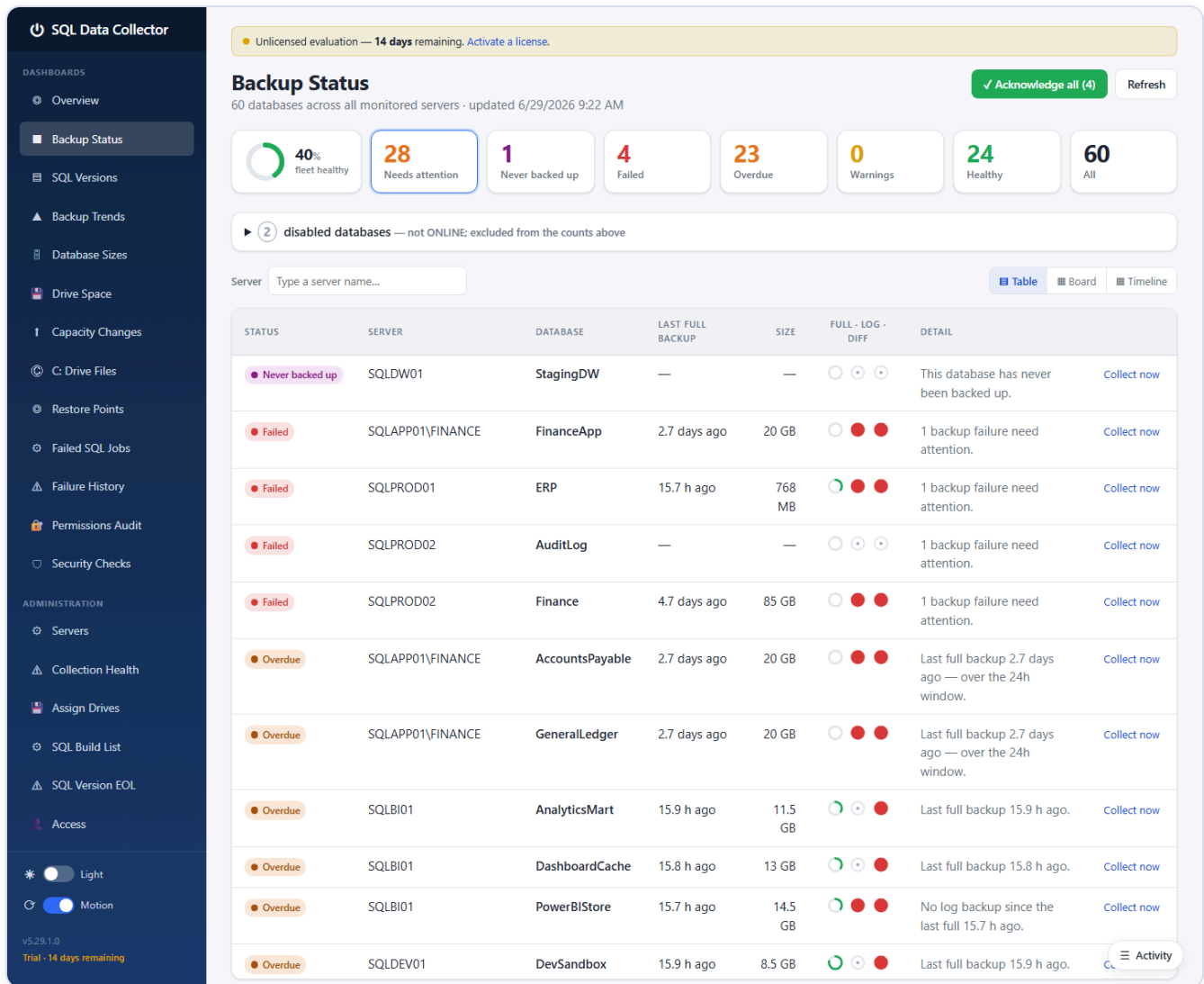
Below the cards:

- **Backup health by server** — one tile per server, coloured by its worst database, so you can spot a problem server instantly.
- **Needs you now** — a single, risk-ranked worklist that mixes every kind of issue (backup failures, broken restore chains, full drives, ...) so you always have a clear "do this first" list.
- **SQL version landscape** — a quick summary of the build/patch spread across the fleet.

The headline numbers are served from a **rollup snapshot** that refreshes on a short interval, so the page stays fast no matter how large the estate.

3.2 Backup Status

Answers: "Did every database get backed up, and on time?"



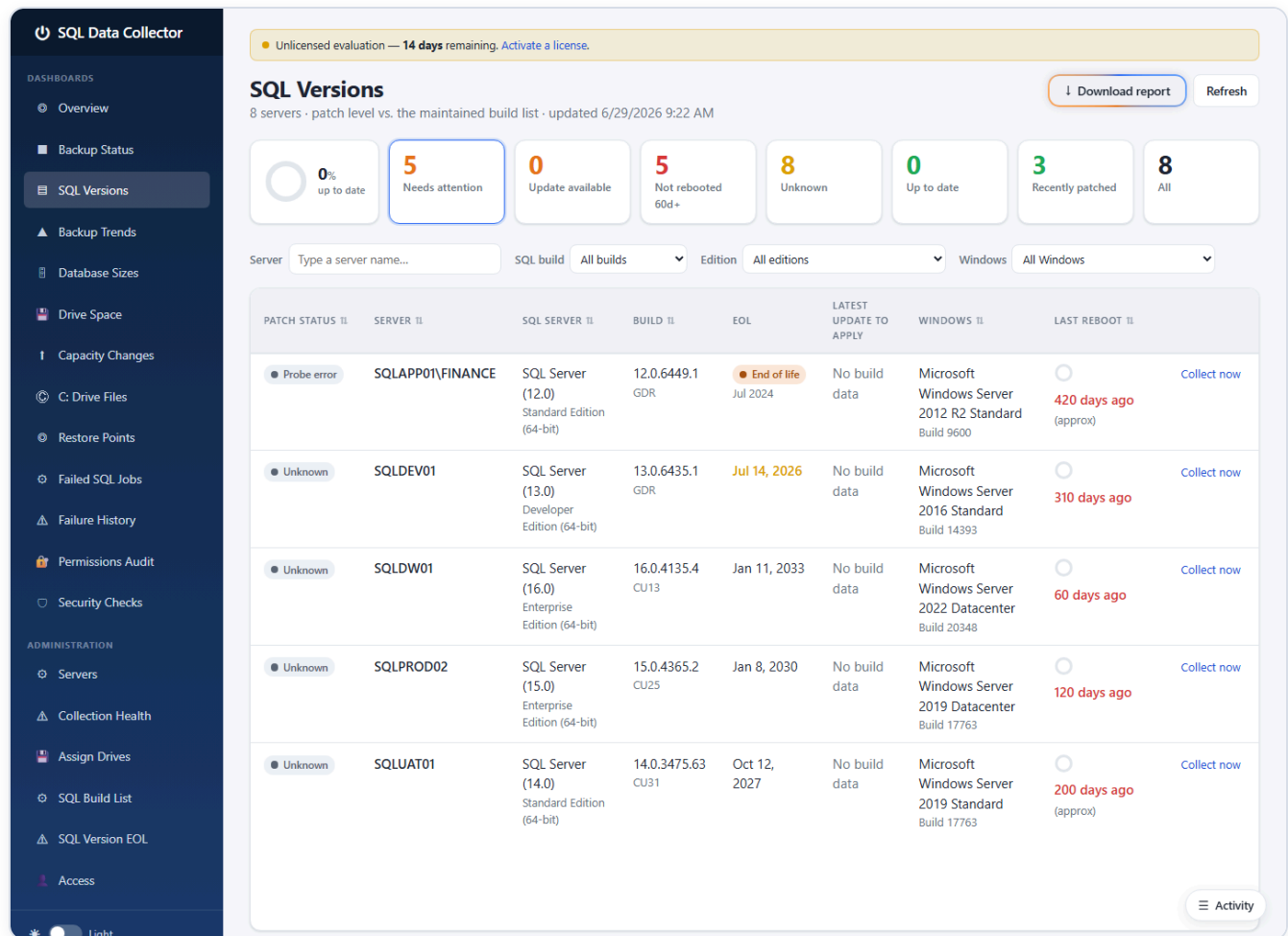
KPI cards across the top summarise the fleet: **healthy**, **needs attention**, **never backed up**, **failed**, **overdue**, and **warnings**. Below, each database is a row showing:

- A **status pill** (Healthy / Warning / Overdue / Failed / Never backed up).
- The server, database, and **last backup age**.
- Three independent **timeliness signals** rendered as small dials:
 - **Full** — a countdown ring that turns red once the database passes its per-server "overdue after" window (default 24 h).
 - **Log** — for FULL/BULK_LOGGED databases, how fresh the last log backup is (green → amber → purple → red). Grey for SIMPLE recovery, where log backups don't apply.
 - **Diff** — how fresh the most recent differential (or full) is.

The worst of the three signals (plus any anomalies) drives the overall status. Use the **Table / Board / Timeline** toggle (top right) to switch between a sortable table, a Kanban-style board grouped by status, and a timeline view. A **server filter** narrows the list.

3.3 SQL Versions

Answers: "What build is each server on, and is any of it out of support?"



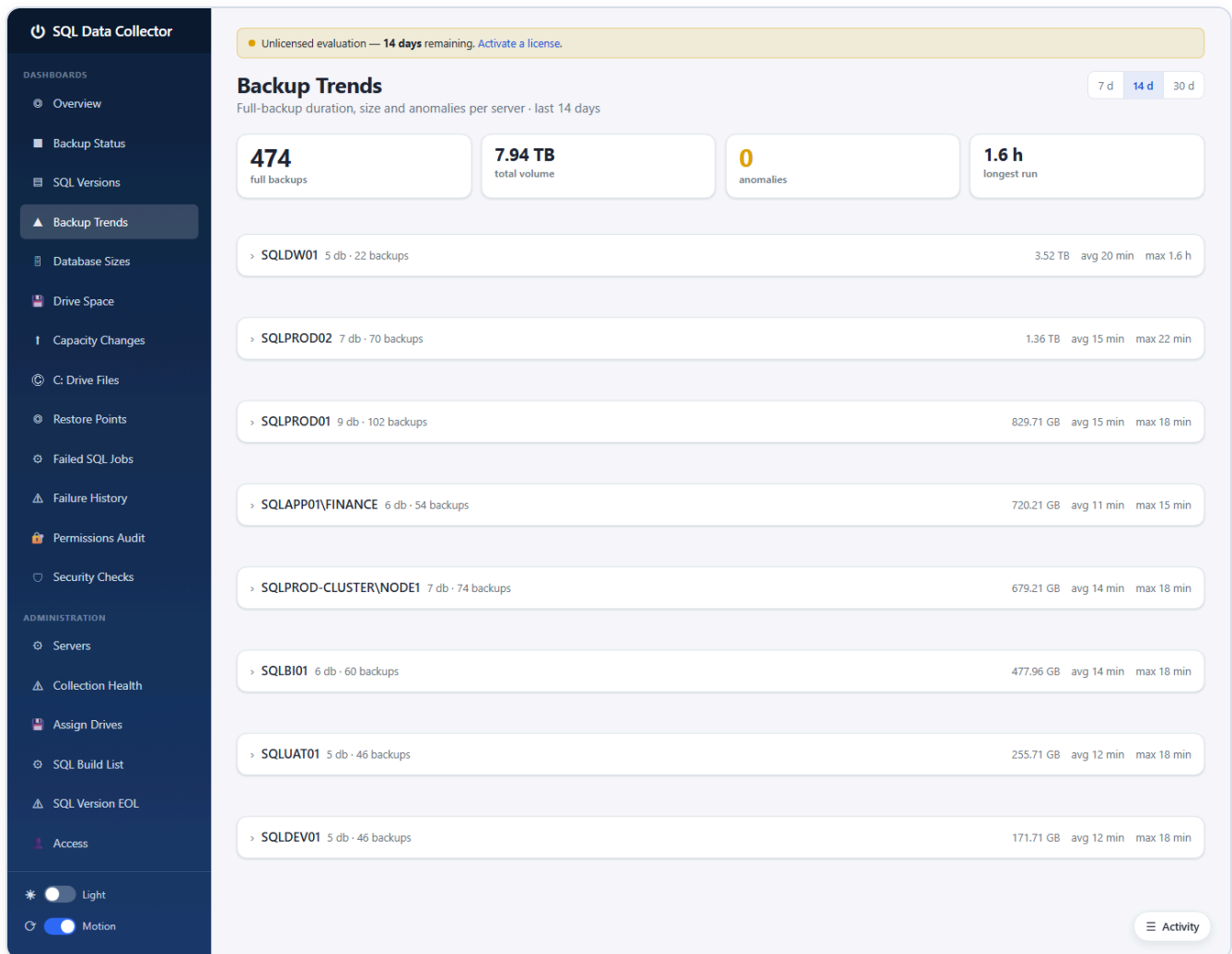
Each server shows its **product version**, **build/patch level**, **edition**, and **Windows OS**. The **end-of-life** column flags lifecycle risk:

- A red **End of life** badge when the major version is past its extended-support date.
- An amber badge when end of life is approaching (within a year).

This is how you catch, for example, a server still on SQL Server 2014 (out of support) or a 2016 instance whose support window is about to close. The build data and the end-of-life dates are both maintained from the Administration pages (see [4. Administration](#)). Use the **Download report** button to export a copy.

3.4 Backup Trends

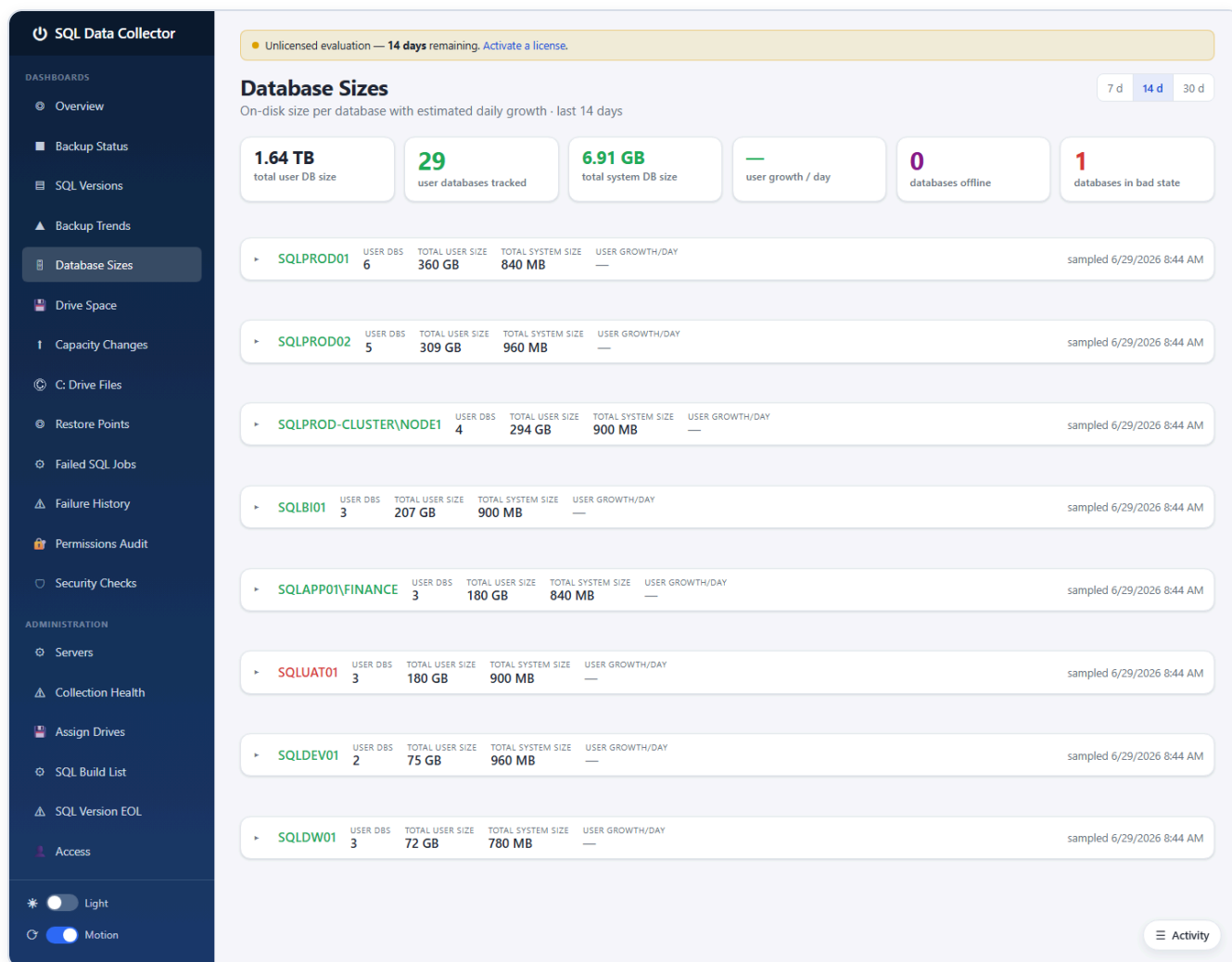
Answers: "How are backup sizes and durations changing over time?"



For a chosen window (7 / 14 / 30 days), the page shows the **number of full backups**, **total volume**, any **anomalies** (a backup that grew far more than expected versus the previous one), and the **longest run**. Each server expands to reveal its databases with average and maximum durations and total size. This is the place to spot a database whose backups are quietly ballooning, or a job whose runtime is creeping up.

3.5 Database Sizes

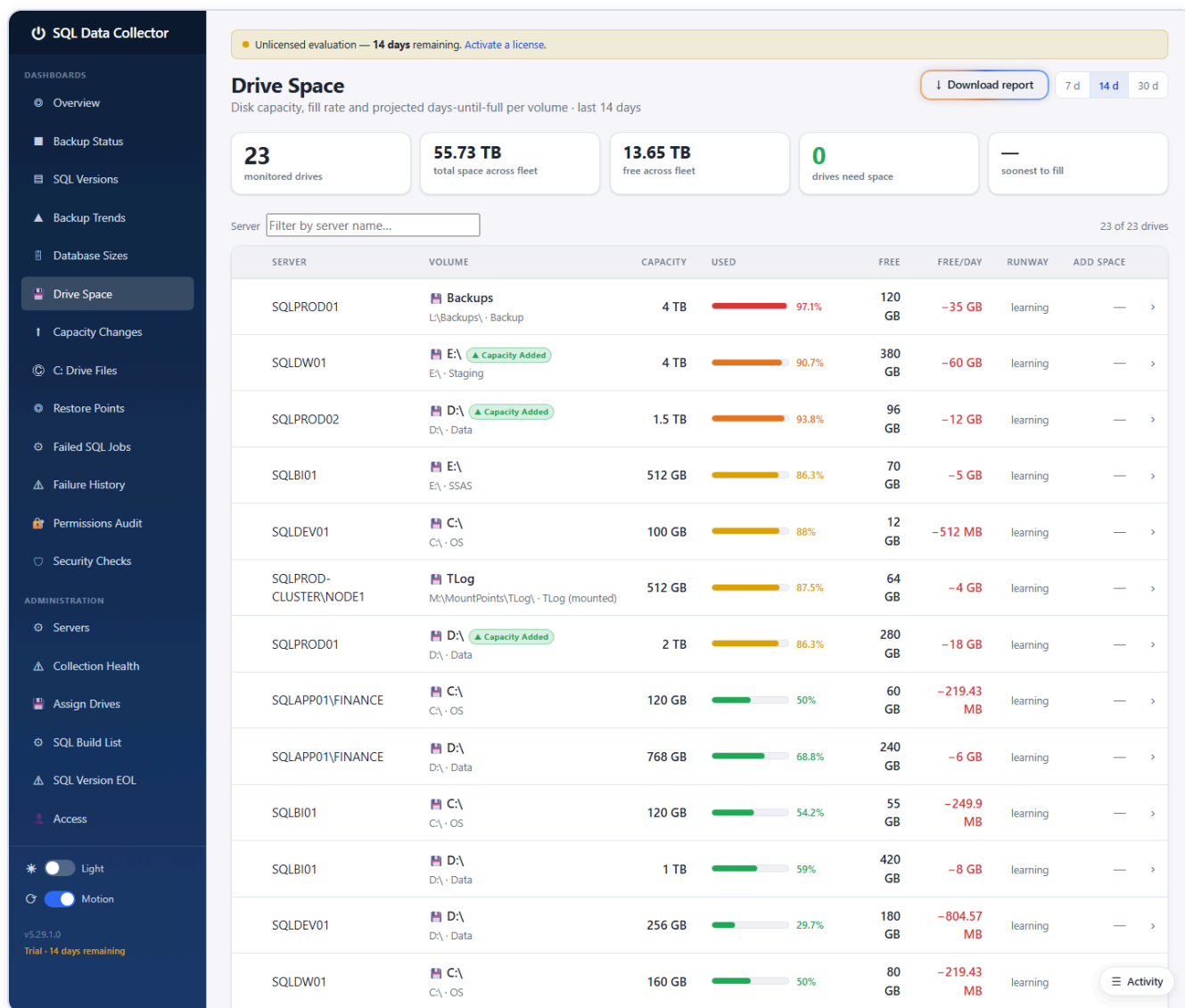
Answers: "How big is everything, and how fast is it growing?"



A fleet total plus an estimated **daily growth** figure sit at the top, with per-server and per-database sizes and small trend sparklines beneath. Because the collector keeps a size history, the growth figures and sparklines reflect real movement over weeks, not a single snapshot — useful for capacity planning and for justifying storage requests.

3.6 Drive Space

Answers: "Which volumes are running out of room, and when?"



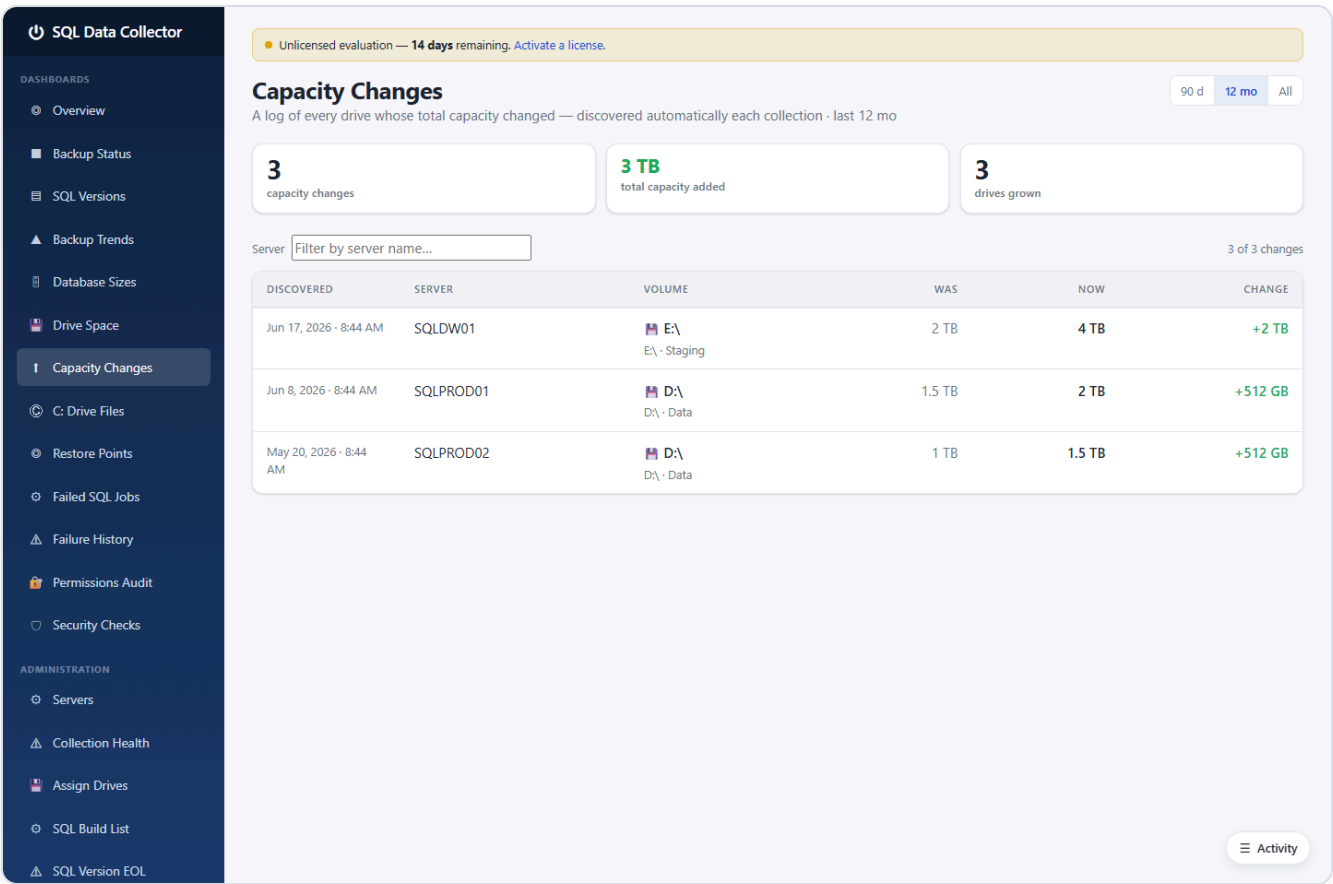
Every volume on every monitored host is listed with its **true capacity** and **free space** (read via WMI, so it includes backup drives and folder-mounted volumes that hold no SQL files). For each volume you get:

- A **used/free capacity bar** and percentage.
- A **Free/day** rate — coloured red when the volume is filling and green when free space is being recovered.
- A **runway** estimate (how long until full at the current rate).

Drives over ~90 % used are highlighted so they jump out. The **Download report** button exports the current picture. (To stop monitoring specific volumes on a server — for example shared cluster volumes — use **Assign Drives** under Administration.)

3.7 Capacity Changes

Answers: "When did a drive's total size change?"



Whenever a volume's *total* capacity changes between collection passes (a disk was grown, a LUN resized), SQL Data Collector records it. The page lists **when** it was discovered, the **server and volume**, and the **was** → **now** sizes with the delta. It's a lightweight audit trail that answers "did someone add space to that server, and when?" without digging through tickets.

3.8 C: Drive Files

Answers: "Are any database files sitting on the OS drive?"

SQL Data Collector

DASHBOARDS

Overview

Backup Status

SQL Versions

Backup Trends

Database Sizes

Drive Space

Capacity Changes

C: Drive Files

Restore Points

Failed SQL Jobs

Failure History

Permissions Audit

Security Checks

ADMINISTRATION

Servers

Collection Health

Assign Drives

SQL Build List

SQL Version EOL

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

C: Drive Files

Database files stored on the host C: drive — they compete with the OS for space and should usually live on a dedicated data/log volume.

4
servers affected

5
databases

6
files on C:

32.06 GB
total on C:

Server

[Expand all](#) 6 of 6 files

SQLDEV01

3 files · 2 databases · 14 GB

SQLAPP01\FINANCE

1 file · 1 database · 12 GB

SQLBI01

1 file · 1 database · 6 GB

SQLPROD01

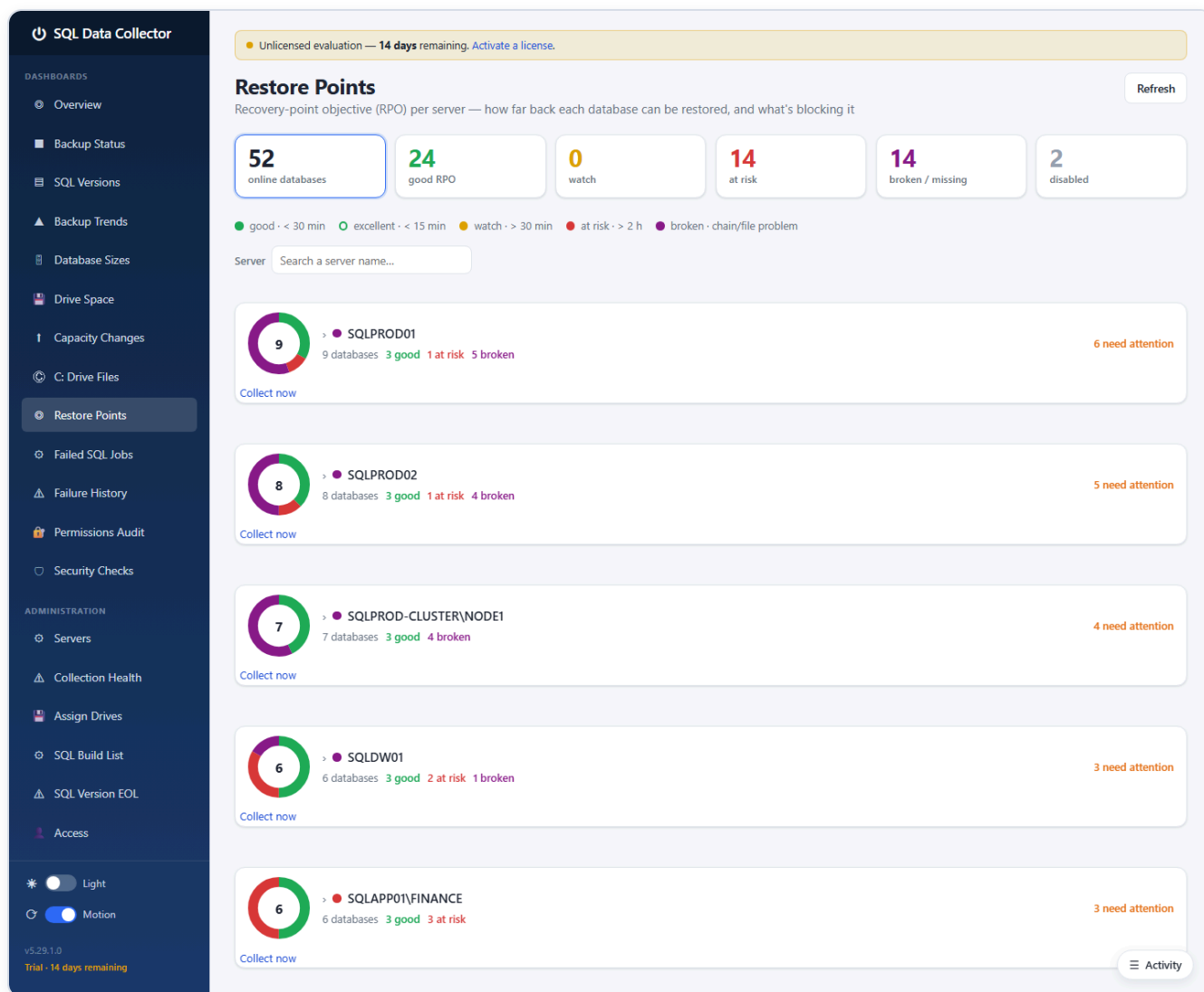
1 file · 1 database · 64 MB

Activity

Database data, log, or filestream files on the host **C:** drive compete with the operating system for space and are a common cause of avoidable outages. This page lists every server, database, and file found on C:, grouped per server, with a positive "all clear" state when there are none.

3.9 Restore Points

Answers: "How far back could I actually restore each database, and what's blocking it?"



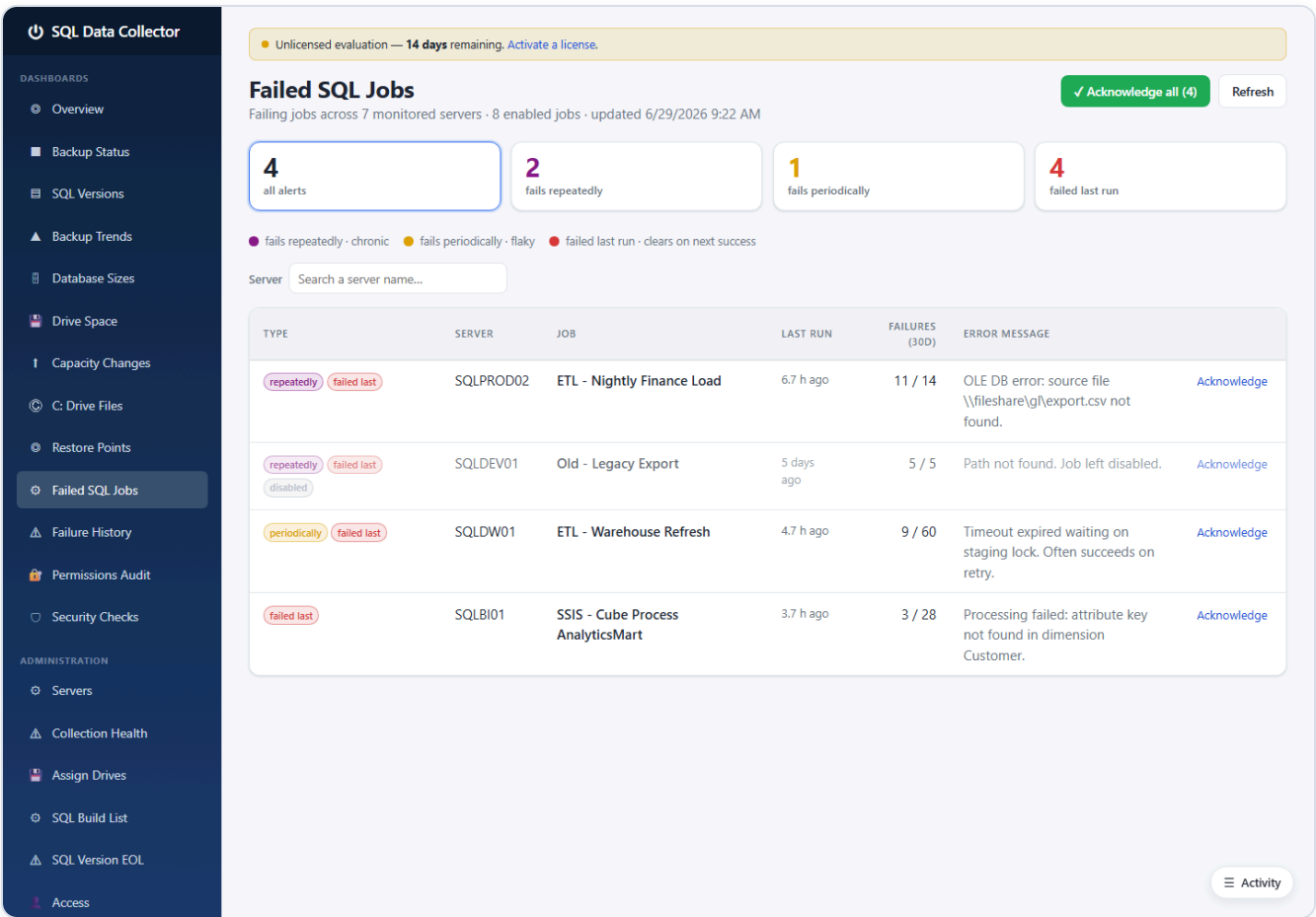
This is recovery-point-objective (RPO) analysis. KPI cards summarise the fleet — **online databases**, **good RPO**, **watch**, **at risk**, **broken/missing**, and **disabled**. Each server is a ring showing how its databases split across those bands:

- **good / excellent** — a recent restore point exists (within minutes).
- **watch / at risk** — the most recent restore point is getting old.
- **broken** — the recovery chain is broken or a backup file is missing, so the database can't be restored to a current point regardless of how recent the last backup looks.

Expanding a server lists each database with exactly what it needs (for example "a log backup is missing from the chain" or "no full backup exists"). System databases are evaluated against their own, more relaxed window.

3.10 Failed SQL Jobs

Answers: "Which SQL Agent jobs are failing across the fleet?"



Every monitored server's SQL Agent jobs are gathered and the problem ones surfaced here, classified as:

- **fails repeatedly** (chronic) — purple,
- **fails periodically** (flaky) — amber,
- **failed last run** (clears on the next success) — red.

Each row shows the server, job, last-run time, a failure count over the last 30 days, and the actual error message. A job whose most recent run **succeeded** drops off automatically (its history is still on Failure History). You can **Acknowledge** a job to quiet it, or **Acknowledge all**. Backup jobs are handled separately so they don't double-report with Backup Status.

3.11 Failure History

Answers: "What backup failures have happened, and were they resolved?"

SQL Data Collector

DASHBOARDS

- Overview
- Backup Status
- SQL Versions
- Backup Trends
- Database Sizes
- Drive Space
- Capacity Changes
- C: Drive Files
- Restore Points
- Failed SQL Jobs
- Failure History
- Permissions Audit
- Security Checks

ADMINISTRATION

- Servers
- Collection Health
- Assign Drives
- SQL Build List
- SQL Version EOL

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

Failure History

A persistent log of backup and SQL Agent job failures over the last 90 days — kept for review after they're acknowledged or resolved

17all failures

10failed SQL jobs

7failed backups

✓ Acknowledge all active (4)

30d90d180d

TYPE	STATUS	SERVER	DATABASE	JOB	FAILED	RESOLVED	MESSAGE
SQL job	Logged	SQLBI01	—	SSIS - Cube Process AnalyticsMart	6/29/2026 5:44 AM	—	Attribute key not found in dimension Customer.
SQL job	Logged	SQLDW01	—	ETL - Warehouse Refresh	6/29/2026 4:44 AM	—	Timeout expired waiting on staging lock.
SQL job	Logged	SQLPROD02	—	ETL - Nightly Finance Load	6/29/2026 2:44 AM	—	OLE DB error: source file not found.
backup - log	Active	SQLPROD01	ERP	DBA - Backup LOG - ERP	6/28/2026 1:44 PM	—	BACKUP LOG [ERP] failed because the log chain was broken by a prior minimally-logged opera...
backup - db	Active	SQLPROD02	Finance	DBA - Backup DATABASE - Finance	6/28/2026 4:44 AM	—	BACKUP DATABASE failed: insufficient space on \\backup01\sql\Finance. Operating system err...
SQL job	Logged	SQLPROD02	—	ETL - Nightly Finance Load	6/28/2026 2:44 AM	—	OLE DB error: source file not found.
backup - log	Auto-resolved	SQLPROD-CLUSTER\NODE1	TradingCore	DBA - Backup LOG - TradingCore	6/27/2026 8:44 PM	6/27/2026 9:44 PM	BACKUP LOG [TradingCore] failed: VDI transient error. Retried successfully on next run.
SQL job	Logged	SQLDW01	—	ETL - Warehouse Refresh	6/27/2026 4:44 PM	—	Timeout expired w...

Activity

Where Backup Status shows the *current* state, Failure History is the **log over time** of individual backup-job failures: when each one failed, on which server/database, the job, and the error. Failures that later auto-resolved are marked, and you can **acknowledge** outstanding ones. It's the page you open when someone asks "has this been failing all week or just today?"

3.12 Permissions Audit

Answers: "Who has access to what, what changed, and can we certify it?"

SQL Data Collector

DASHBOARDS

Overview

Backup Status

SQL Versions

Backup Trends

Database Sizes

Drive Space

Capacity Changes

C: Drive Files

Restore Points

Failed SQL Jobs

Failure History

Permissions Audit

Security Checks

ADMINISTRATION

Servers

Collection Health

Assign Drives

SQL Build List

SQL Version EOL

Access

Light

Motion

v5.29.1.0

Trial - 14 days remaining

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

Permissions Audit sox

Deep SQL permission inventory — down to object/column level — with AD-group inheritance resolved. Direct grants and inherited (via-group) access both appear, so you can see every path to a permission.

7 servers audited

18 permission rows

14 direct

4 inherited (via AD)

17 principals

Last audited 6/28/2026 9:52 AM · 13 AD groups expanded · 7 of 8 servers succeeded

Recent audit durations (last 2)

Review

Permissions

Categories

By User

Server Roles

Changes

Change log (who) 2

Remediation

Attestation

Risk-sorted, exception-based review: only the access that's new, changed, or privileged. Certify what's expected to clear it; it returns only if that principal's access changes. Reviewing by **server · login**.

25 needs review

2 changed since sign-off

17 high-risk to review

11 critical access

Suspicious activity — investigate (2)

Flagged from the default trace (Change log), grouped by server. Each must be **investigated** with a written outcome before the attestation can be sealed. A newly-added server's **first capture** back-fills pre-existing activity — that batch can be cleared with **Investigate all**; later refreshes are reviewed individually. Archiving the change log does *not* clear these.

SQLDEV01 1 flagged

WHEN	WHO	WHAT	WHY FLAGGED	
6/28/2026 5:44 AM	CONTOSO\ysmith	ALTER SERVER ROLE [sysadmin] ADD MEMBER [CONTOSO\developers];	HIGH High-privilege server role granted to a broad group outside change window.	Investigated

SQLPROD02 1 flagged

WHEN	WHO	WHAT	WHY FLAGGED	
6/26/2026 10:44 AM	sa	GRANT CONTROL ON DATABASE::Finance TO [CONTOSO\contractor2];	HIGH CONTROL granted to an external contractor on a SOX-scoped database.	Investigated

Needs review

Everything

Sort

Risk (high first)

Server contains...

✓ Certify visible (25)

CRITICAL

CONTOSO\developers

AD GROUP

SQLDEV01

sysadmin · Admin Access · 1 db

Changed since sign-off

via CONTOSO\developers

✓ Certify all

⚠ Revoke all

🗑 Drop user

CRITICAL

CONTOSO\trading_ops

AD GROUP

SQLPROD-CLUSTER\NODE1

serveradmin · Execute · 1 db

New

via CONTOSO\trading_ops

✓ Certify all

⚠ Revoke all

🗑 Drop user

CRITICAL

CONTOSO\bi_admins

AD GROUP

SQLBI01

sysadmin

New

via CONTOSO\bi_admins

✓ Certify all

⚠ Revoke all

☰ Activity

This is the most feature-rich area. The collector captures server-role membership and database permissions across the fleet (expanding Active Directory groups so you see the real people behind a group), and tracks them over time. The page provides:

- **KPI cards** — servers audited, admins, direct grants, over-privileged principals, and more.
- **Tabs** — Review, Permissions, Categories, Server Roles, Change Log, Change Latest, Remediation, and Attestation.
- A **Suspicious activity** panel that flags risky changes (for example a new sysadmin grant) for investigation.
- A **Review** workflow where each principal's access can be approved or flagged, leading to a sealed, certified **attestation** suitable for compliance evidence.
- A **Remediation** worklist that turns "this access should be removed" decisions into ready-to-run SQL scripts and DBA to-do items, which are then auto-verified on the next audit.

Because every capture is timestamped and kept, you can answer "who could read the Finance database on 1 March?" months later.

3.13 Security Checks

Answers: "Is each server configured securely?"

SQL Data Collector

DASHBOARDS

- Overview
- Backup Status
- SQL Versions
- Backup Trends
- Database Sizes
- Drive Space
- Capacity Changes
- C: Drive Files
- Restore Points
- Failed SQL Jobs
- Failure History
- Permissions Audit
- Security Checks

ADMINISTRATION

- Servers
- Collection Health
- Assign Drives
- SQL Build List
- SQL Version EOL
- Access

Light

Motion

v5.29.1.0
Trial - 14 days remaining

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

Security Checks

Common SQL hardening checks across the estate. Read-only — nothing is changed on the servers. Toggle a server's participation under **Servers** → **Dashboards**.

8 servers checked

4 servers with failures

7 high-severity findings

22 checks passing

Server

Check

All checks

☐ Failures only

Re-run now

Last checked 6/29/2026 8:29 AM

SERVER	DEFAULT TRACE RUNNING	SA ACCOUNT HARDENED	XP_CMDSHELL DISABLED	SURFACE AREA MINIMISED	GUEST DISABLED IN USER DATABASES	CHECKED
SQLDEV01	✗	✗	✗	✗	✗	6/29/2026 8:29 AM
HIGH Default trace running: Default trace is NOT running. HIGH sa account hardened: The 'sa' login is ENABLED with default-style name. HIGH xp_cmdshell disabled: xp_cmdshell is ENABLED. HIGH Surface area minimised: Multiple risky surface-area options enabled. MEDIUM Guest disabled in user databases: Guest enabled in all user databases.						
SQLPROD02	✓	✗	✗	✓	!	6/29/2026 8:29 AM
HIGH sa account hardened: The 'sa' login is ENABLED with a weak password policy. HIGH xp_cmdshell disabled: xp_cmdshell is ENABLED. Disable it unless a documented requirement exists.						
SQLUAT01	✓	!	✓	✗	✗	6/29/2026 8:29 AM
HIGH Surface area minimised: OLE Automation Procedures are enabled. MEDIUM Guest disabled in user databases: Guest enabled in 2 user databases.						
SQLAPP01\FINANCE	?	?	?	?	?	6/29/2026 8:29 AM
MEDIUM Default trace running: Could not evaluate: connection to SQLAPP01\FINANCE failed. MEDIUM sa account hardened: Could not evaluate: connection failed. MEDIUM xp_cmdshell disabled: Could not evaluate: connection failed. MEDIUM Surface area minimised: Could not evaluate: connection failed. MEDIUM Guest disabled in user databases: Could not evaluate: connection failed.						
SQLBI01	✓	✓	✓	!	✓	6/29/2026 8:29 AM
SQLDW01	✓	✓	✓	✓	✓	6/29/2026 8:29 AM
SQLPROD-CLUSTER\NODE1	✓	✓	✓	✓	✓	6/29/2026 8:29 AM

Activity

A set of SQL Server hardening checks runs per server (for example: default sa account state, surface-area configuration, guest access in user databases, login policy). Each result is **Pass**, **Warn**, or **Fail** with a severity (Low / Medium / High) and an explanation. KPI cards summarise how many servers are clean versus failing. Toggle a server's participation from the Servers page if a check doesn't apply.

Continue to [4. Administration »](#)

4. Administration

The **Administration** section of the sidebar is where you manage what gets monitored, who can see it, and how the collector behaves. These pages are visible to administrators.

4.1 Servers

The starting point — add and manage the SQL Servers you monitor.

SQL Data Collector

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

Monitored Servers

SQL Server instances the collector reads from. Use **Dashboards** to choose which monitors each server is pulled for. · 8 of 14 licensed servers used

2 servers with collection failures.

- SQLAPP01\FINANCE — Backup Status, Database Sizes, Drive Space, Security Checks, SQL Agent Jobs, SQL Permissions Audit, SQL Versions failed 6/29/2026 7:14 AM
- SQLDEV01 — SQL Permissions Audit failed 6/28/2026 2:44 AM

[View collection health →](#)

Search

Set dashboards for all servers: ☒ Backup Status ☒ SQL Versions ☒ Drive Space ☒ SQL Agent Jobs ☒ SQL Permissions Audit ☒ Security Checks ☒ Database Sizes [Check all](#)

[Uncheck all](#)

SERVER	AUTH	DBS	OVERDUE AFTER	LAST COLLECTED	DASHBOARDS	STATE
SQLAPP01\FINANCE	Service account (Windows)	7	24 h	6/29/2026 7:14 AM ⚠ Login timeout expired while connecting to SQLAPP01\FINANCE.	☒ Backup Status ☒ SQL Versions ☒ Drive Space ☒ SQL Agent Jobs ☒ SQL Permissions Audit ☒ Security Checks ☒ Database Sizes	Enabled Collect Disable Delete
SQLBI01	Service account (Windows)	7	24 h	6/29/2026 8:39 AM	☒ Backup Status ☒ SQL Versions ☒ Drive Space ☒ SQL Agent Jobs ☒ SQL Permissions Audit ☒ Security Checks ☒ Database Sizes	Enabled Collect Disable Delete
SQLDEV01	Service account (Windows)	7	72 h	6/29/2026 7:54 AM	☒ Backup Status ☒ SQL Versions ☒ Drive Space ☒ SQL Agent Jobs ☒ SQL Permissions Audit ☒ Security Checks ☒ Database Sizes	Enabled Collect Disable Delete
SQLDW01	Service account (Windows)	7	36 h	6/29/2026 8:38 AM	☒ Backup Status ☒ SQL Versions ☒ Drive Space	Enabled Collect Disable Activity

Each monitored server has:

- A **name** (`SERVER` or `SERVER\INSTANCE`) and an **authentication mode** — *Windows* (the service account) or a stored **SQL login** (the password is encrypted at rest).
- An **Enabled** toggle — disable a server to pause collection without deleting its history.

- An **Overdue after (hours)** value — the per-server window that drives the Full backup red threshold on Backup Status.
- A row of **per-monitor checkboxes** (Backup Status, Drive Space, SQL Agent Jobs, SQL Permissions, Security Checks, ...) so you can exclude individual monitors on a given server.

If a server can't be reached, a banner highlights it and the **Collection Health** page explains why. **Collect now** forces an immediate pass.

Licensing note. The number of servers you can actively collect is governed by your licence (or the trial limit). Servers beyond the limit are listed but flagged as not collected.

4.2 Collection Health

Answers: "Is the collector itself working — and if not, why?"

SQL Data Collector

DASHBOARDS

Overview

Backup Status

SQL Versions

Backup Trends

Database Sizes

Drive Space

Capacity Changes

C: Drive Files

Restore Points

Failed SQL Jobs

Failure History

Permissions Audit

Security Checks

ADMINISTRATION

Servers

Collection Health

Assign Drives

SQL Build List

SQL Version EOL

Access

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

Collection Health

A history of collection failures — which monitor failed, when, and why. A record stays here even after a later collection succeeds (it shows as *Recovered*) until you dismiss it.

8 failure records (8 currently failing).

Refresh

Dismiss all

SERVER	COLLECTION	STATUS	LAST FAILED	ERROR	
SQLAPP01\FINANCE	Backup Status	Currently failing	6/29/2026 7:14 AM	Login timeout expired (error 258) connecting to SQLAPP01\FINANCE. The network may be down or the instance offline.	Dismiss
SQLAPP01\FINANCE	SQL Versions	Currently failing	6/29/2026 7:14 AM	Login timeout expired (error 258) connecting to SQLAPP01\FINANCE. The network may be down or the instance offline.	Dismiss
SQLAPP01\FINANCE	Drive Space	Currently failing	6/29/2026 7:14 AM	Login timeout expired (error 258) connecting to SQLAPP01\FINANCE. The network may be down or the instance offline.	Dismiss
SQLAPP01\FINANCE	SQL Agent Jobs	Currently failing	6/29/2026 7:14 AM	Login timeout expired (error 258) connecting to SQLAPP01\FINANCE. The network may be down or the instance offline.	Dismiss
SQLAPP01\FINANCE	SQL Permissions Audit	Currently failing	6/29/2026 7:14 AM	Login timeout expired (error 258) connecting to SQLAPP01\FINANCE. The network may be down or the instance offline.	Dismiss
SQLAPP01\FINANCE	Security Checks	Currently failing	6/29/2026 7:14 AM	Login timeout expired (error 258) connecting to SQLAPP01\FINANCE. The network may be down or the instance offline.	Dismiss
SQLAPP01\FINANCE	Database Sizes	Currently failing	6/29/2026 7:14 AM	Login timeout expired (error 258) connecting to SQLAPP01\FINANCE. The network may be down or the instance offline.	Dismiss
SQLDEV01	SQL Permissions Audit	Currently failing	6/28/2026 2:44 AM	AD group expansion failed: the LDAP server is unavailable (0x51).	Dismiss

Activity

This page tracks each **monitor** on each server and flags any that are failing, with the **last success**, **last failure**, and the actual **error message** (for example a login timeout because a host is unreachable). It's the first place to look when a dashboard seems stale — it separates "the data is genuinely fine" from "we haven't been able to collect it." Failing entries can be **dismissed** once understood.

4.3 Assign Drives

Choose which volumes are monitored on each server.

SQL Data Collector

DASHBOARDS

Overview

Backup Status

SQL Versions

Backup Trends

Database Sizes

Drive Space

Capacity Changes

C: Drive Files

Restore Points

Failed SQL Jobs

Failure History

Permissions Audit

Security Checks

ADMINISTRATION

Servers

Collection Health

Assign Drives

SQL Build List

SQL Version EOL

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

Assign Drives

Choose which volumes each server monitors on the Drive Space dashboard. For clustered instances on Cluster Shared Volumes, every node sees all `C:\ClusterStorage\*` mounts — uncheck the drives that belong to other instances.

Server

8 of 8 servers

SERVER	MONITORED	TOTAL VOLUMES	
SQLAPP01\FINANCE	2	2	Assign drives
SQLBI01	3	3	Assign drives
SQLDEV01	2	2	Assign drives
SQLDW01	4	4	Assign drives
SQLPROD-CLUSTER\NODE1	3	3	Assign drives
SQLPROD01	4	4	Assign drives
SQLPROD02	3	3	Assign drives
SQLUAT01	2	2	Assign drives

Excluded drives are still collected (so you can re-include them here) but hidden from every Drive Space view — the dashboard, the capacity report, the Overview risk feed, and Capacity Changes. Takes effect immediately.

Activity

Clustered SQL Servers see every node's Cluster Shared Volumes, which would otherwise clutter Drive Space with volumes that don't belong to the instance. Assign Drives lets you **exclude** specific volumes per server (excluded volumes are still collected behind the scenes, so you can re-include them at any time). Servers with no exclusions monitor every volume, so non-clustered hosts need no setup.

4.4 SQL Build List

Keep the SQL Server build reference up to date.

SQL Data Collector

DASHBOARDS

Overview

Backup Status

SQL Versions

Backup Trends

Database Sizes

Drive Space

Capacity Changes

C: Drive Files

Restore Points

Failed SQL Jobs

Failure History

Permissions Audit

Security Checks

ADMINISTRATION

Servers

Collection Health

Assign Drives

SQL Build List

SQL Version EOL

Access

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

SQL build list

Upload the maintained SQL Server builds CSV. Rows are upserted into the reference table by build number — new builds added, existing ones updated. Drives the patch-level checks on the SQL Versions dashboard.

Current build list

0 builds on file.

Upload CSV

Expected columns (the public build-list format): `SQLServer, Version, Build, FileVersion, Description, Link, ReleaseDate, SP, CU, HF, RTH, CTP, New, Withdrawn, CVEs`. Column order doesn't matter; `Build` is required and is the upsert key. Export the sheet as CSV and select it here.

SQL builds CSV

Choose File

No file chosen

Add a build

Add or fix a single build row by hand, without importing the whole file. `Build` is required and is the key.

Build (required)	Version line	Product	File version
15.0.4455.2	15.0	2019	2019.150.4455.2
Release date			
mm/dd/yyyy			
Description			
Link			
CVEs (comma-separated)			
Flags			

Activity

The SQL Versions dashboard maps each server's build number to a friendly description and patch level using a build reference table. This page lets you **import** the public build list (export the community-maintained sheet to CSV and upload it — no internet access is required from the collector) or **add/edit a single build** by hand. The columns mirror the standard build list (build number, version, release date, SP/CU/HF flags, description, CVEs).

4.5 SQL Version EOL

Maintain the end-of-life dates that drive the lifecycle badges.

SQL Data Collector

DASHBOARDS

Overview

Backup Status

SQL Versions

Backup Trends

Database Sizes

Drive Space

Capacity Changes

C: Drive Files

Restore Points

Failed SQL Jobs

Failure History

Permissions Audit

Security Checks

ADMINISTRATION

Servers

Collection Health

Assign Drives

SQL Build List

SQL Version EOL

Access

Light

Motion

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

SQL version end-of-life

Support-lifecycle dates per SQL Server major version — drives the EOL column on the SQL Versions dashboard. Edit a row, or add one for a version that isn't listed.

Add a version

Product (e.g. 2019)

Major version (e.g. 15.0)

Mainstream support ends

Extended support ends (EOL)

2019

15.0

mm/dd/yyyy

mm/dd/yyyy

Notes

Add row

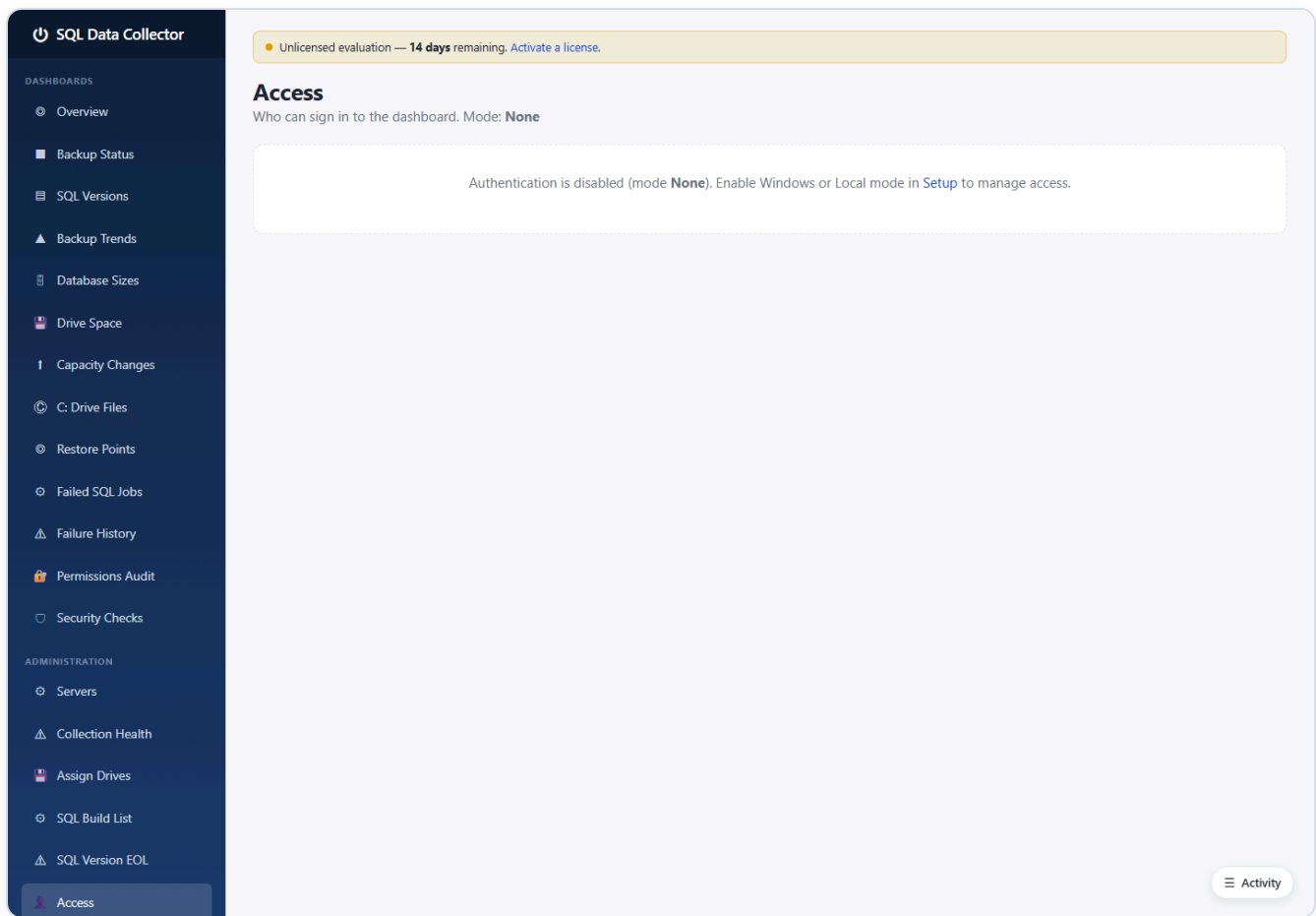
PRODUCT	VERSION	MAINSTREAM END	EOL (EXTENDED END)	NOTES
2000	8.0	2008-04-08	2013-04-09	Edit Delete
2005	9.0	2011-04-12	2016-04-12	Edit Delete
2008	10.0	2014-07-08	2019-07-09	Edit Delete
2008 R2	10.50	2014-07-08	2019-07-09	Edit Delete
2012	11.0	2017-07-11	2022-07-12	Edit Delete
2014	12.0	2019-07-09	2024-07-09	Edit Delete
2016	13.0	2021-07-13	2026-07-14	Edit Delete
2017	14.0	2022-10-11	2027-10-12	Edit Delete
2019	15.0	2025-02-28	2030-01-08	Edit Delete
2022	16.0	2028-01-11	2033-01-11	Edit Delete
2025	17.0	—	—	Lifecycle dates not yet published by Microsoft. Edit Delete

Activity

End-of-life is a lifecycle fact, not part of the build list, so it's maintained separately here: one row per major version with its **mainstream** and **extended** (end-of-life) dates. These dates feed the red/amber EOL badges on SQL Versions. The table ships pre-seeded for SQL Server 2000 through 2025 and is editable one row at a time.

4.6 Access

Control who can sign in and what they can see.



When sign-in is enabled (Windows/AD or Local mode), the Access page is where you:

- Grant access to **AD users or groups** (Windows mode) or create **local accounts** (Local mode).
- Mark a principal as an **administrator** or **auditor**.
- Grant **per-dashboard view access** — a viewer can be limited to, say, only Backup Status and Drive Space.
- Set a **default landing page** per user.

In the demonstration build the sign-in mode is **None** (localhost-only), so the page simply explains that authentication is disabled and points to Setup. Switch to Windows or Local mode in Setup to manage access.

4.7 Login Audit

A record of who signed in (and who failed to).

When sign-in is enabled, every authentication attempt — success or failure — is recorded with the username, method, timestamp, IP address, and user agent. The Login Audit page (admins only) lists them so you can spot repeated failures or unexpected sign-ins. With sign-in set to **None**, there is nothing to audit.

4.8 Settings

Tune how the collector behaves.

SQL Data Collector

DASHBOARDS

Overview

Backup Status

SQL Versions

Backup Trends

Database Sizes

Drive Space

Capacity Changes

C: Drive Files

Restore Points

Failed SQL Jobs

Failure History

Permissions Audit

Security Checks

ADMINISTRATION

Servers

Collection Health

Assign Drives

SQL Build List

SQL Version EOL

Access

Light

Motion

v5.29.1.0

Trial - 14 days remaining

Unlicensed evaluation — 14 days remaining. [Activate a license.](#)

Settings

Collector and dashboard settings (written to collector.settings.json). Most changes apply after a service restart.

License unlicensed evaluation

Status

Servers allowed

Expires

Days remaining

Grace

unlimited

—

14

Challenge code — send to your vendor to (re)license this server

XXDC-AESDQ-NDCHA-3WKOJ-QFUYD-AMJTF-U2DIY-JYFVQ-TEYLB-FUYDO-YJTG-3DIZR-TH8TD-SAAMK-ZKEHQ-2GKJC-UZNBQ-HEYDY-HQ

Response key — paste a new key here to activate or renew

Activate

Collection

Collection interval (minutes)

Startup delay (seconds)

Max parallel servers

Connection timeout (seconds)

15

2

16

15

Command timeout (seconds)

Page auto-refresh fallback (seconds)

Verify backup files on disk

Sample drive space

120

150

Every 1hr

Every 1hr

Keep drive-space history (days)

☐ Background collector enabled (master switch for collecting data)

☒ Log errors to file (logs\collector-*.log)

— warnings & errors only, stays empty when healthy

☐ Log errors to Windows Event Log (off by default)

180

Backup-file verification runs `xc_fileexist` for every full/diff + the current log chain on each server at this cadence (more often = quicker to notice deleted backups, but heavier on large fleets). A restore script always verifies live on open, and a manual "Collect now" forces a re-check immediately. Verbose diagnostics logs each probe result to the service log for troubleshooting.

Drive-space sampling reads each server's volumes (T-SQL) plus true capacity via WMI on this cadence — less often is lighter on large fleets and capacity rarely changes; "Collect now" always samples. History older than the retention window is pruned (0 = keep all); the capacity-change log is kept regardless.

SQL Permissions Audit (SOX)

Run permission audit

Off (on-demand only)

A pass captures a deep permission inventory (down to object/column level) read-only from each server in the SQL Permissions Audit monitor, expands the referenced AD groups, and records who inherits what. It's heavy, so it runs far less often than collection — the scheduled pass fires when a server hasn't been audited within this window. Use "Audit now" to run one immediately.

Audit now

Audit data retention (SOX)

Permission snapshots / deltas

Permission changes

Audit runs

Change log (trace) events — archived, not deleted

1 year

1 year

1 year

90 days

Attestation reviews

Keep everything

The permission store always keeps the **current** state; these bound only the **history** — the closed point-in-time intervals (deltas/snapshots), the change log, the audit-run headers, and the attestation sign-offs. 0 = **keep everything**. Pruned at the end of each audit pass. Point-in-time reports reach back only as far as the snapshot retention.

Activity

The Settings page exposes the operator-editable configuration without hand-editing `collector.settings.json`. Notable groups:

- **Collection** — the collection interval, parallelism, connection/command timeouts, the background-collector master switch, logging destinations (file and/or Windows Event Log), and per-monitor verbose logging.
- **Drive space** — the drive-space collection cadence and how long to retain drive samples.

- **Backup timeliness & Restore points (RPO)** — the thresholds that decide when a backup or restore point is a warning versus a problem, including separate, more relaxed windows for **system databases**.
- **SQL Permissions Audit** — the run mode and audit-data retention.
- **WMI authentication** — whether to use the service account or specific credentials for the WMI calls that read true drive capacity and last-reboot time on remote hosts.
- **Licence** — status, the machine challenge string, and activation (see below).

Most changes apply on the next service restart.

4.9 Licensing

SQL Data Collector uses **node-locked, offline licensing**. There is no phone-home: activation is a challenge/response exchange tied to the machine.

- A fresh install runs on a **time-limited trial** (the banner at the top of every page and the countdown under the version in the sidebar show how long remains).
- To activate, copy the **challenge** string from the Settings → Licence card (or the Activate page) and send it to your vendor, who returns a **response** key that you paste back in.
- The licence governs how many servers can be actively collected.

Because the licence is bound to the machine's identity, moving the install to a different host requires a new activation.

That completes the administration tour. [Back to the top](#).