

What is SAM?

The Beginner's Explainer Guide

A plain-English primer on Software Asset Management — no jargon, no prior experience needed. Understand what SAM is, how it differs from ITAM, the basics of licence compliance and audits, and where certification fits in.

INSIDE THIS GUIDE

- ✓ What software asset management really means, in plain language
- ✓ SAM versus ITAM — simple definitions with everyday examples
- ✓ Licence compliance & audit basics every beginner should know
- ✓ A glossary of key terms — plus where the CSAM certification fits

How to use this guide

This guide is written for complete beginners. You don't need an IT background, and you won't find dense jargon here. Each idea is introduced in plain English, with a simple example to make it stick. Read it cover to cover, or jump to the part you need.

By the end you'll be able to explain what software asset management is, tell SAM and ITAM apart, understand why licence compliance matters, and know roughly what happens during a software audit. That's a solid foundation for going further.

WHAT'S INSIDE

Part 1	What is SAM?	p.3
Part 2	SAM versus ITAM	p.9
Part 3	Licences & compliance	p.15
Part 4	Software audits	p.21
Part 5	Glossary & next steps	p.25

A quick note on terms

Throughout, you'll see a few acronyms: **SAM** (Software Asset Management), **ITAM** (IT Asset Management) and **HAM** (Hardware Asset Management). Don't worry about memorising them yet — each is explained as it appears, and all of them are collected in the glossary at the back.

PART ONE

What is SAM?

Start here. We define software asset management in everyday language and explain why organisations care so much about getting it right.

Software asset management, in plain English

Software Asset Management (SAM) is simply the practice of keeping track of all the software an organisation owns and uses — and making sure it is bought, installed, used and retired in a sensible, legal and cost-effective way.

That's really the whole idea. If a company knows exactly what software it has paid for, where it is installed, who is using it, and whether that matches what it is allowed to use, then it is doing SAM. When any of those pieces are unknown, the company is spending money it can't account for and taking on risk it can't see.

IN ONE SENTENCE

SAM is knowing what software you own, what you actually use, and making sure the two match — so you stay legal and stop wasting money.

SAM is partly a set of **processes** (the steps people follow), partly **data** (accurate records of licences and installations) and partly **governance** (the rules that keep everyone honest). You don't need special tools to start — you need clear information and someone responsible for keeping it up to date.

You're already learning the fundamentals

RELATED

Turn this primer into a recognised credential. The GSDC Certified Software Asset Manager builds directly on everything in this guide.

Enroll Now

A simple way to picture it

Imagine a library. The library owns a certain number of copies of each book. It lends those copies out, keeps a record of who has what, and makes sure it never lends more copies than it actually owns. If it loses track, it ends up with missing books, angry readers, and fines it can't explain.

Software asset management works the same way. The "books" are software licences. The "copies owned" are the licences a company has paid for. The "copies lent out" are the installations sitting on people's laptops and servers. SAM is the librarian who makes sure the numbers always add up.

EVERYDAY EXAMPLE

A company buys 100 licences for a design tool. Over two years, staff install it on 130 machines. Nobody tracked it. The company is now using 30 copies it never paid for — a compliance gap that a vendor audit could turn into a large, unexpected bill. Good SAM catches this long before the auditor does.

Notice that nothing in this example is technical. SAM is mostly about **visibility** and **discipline**, not complicated technology.

The three things SAM keeps track of

At its core, software asset management watches three numbers and keeps them aligned. When they drift apart, money is lost or risk creeps in.

1

Entitlements

What you are *allowed* to use — the licences, subscriptions and rights you have actually purchased, with their terms and limits.

2

Installations

What is *actually deployed* — every copy installed across laptops, desktops, servers and the cloud, whether or not it is being used.

3

Usage

What is *genuinely used* — which installed software people open and rely on, versus what sits idle and could be reclaimed.

When entitlements match installations, you are **compliant**. When installations match usage, you are **efficient**. SAM's job is to close the gaps between all three — staying legal while cutting waste.

Why SAM exists: the problem it solves

Organisations today spend enormous sums on software. A surprising share of that money is wasted — on licences nobody uses, duplicate tools that do the same job, and contracts that were never properly managed. At the same time, software vendors are running more audits than ever, and companies that can't prove they are properly licensed face heavy fines and compliance failures.

Without SAM

- ✗ Paying for licences nobody uses
- ✗ Buying tools the company already owns
- ✗ No idea what's installed or where
- ✗ Surprise bills after a vendor audit
- ✗ Renewals signed blind, at full price

With SAM

- ✓ Spend matched to real usage
- ✓ Duplicate and idle tools removed
- ✓ A clear, current inventory
- ✓ Audit-ready records, no surprises
- ✓ Renewals negotiated from facts

In short: SAM exists to turn software from an untracked cost into a managed, optimised asset.

LIMITED TIME

Make your SAM knowledge official

A vendor-neutral, globally recognised certification — enrolment open for a limited time. Move from reading about SAM to being certified in it.

Enroll Now

What good SAM looks like day to day

SAM isn't a one-off project — it's an ongoing rhythm. Here's what a healthy SAM practice tends to look like in a typical week, in plain terms.

- **Records stay current.** New purchases and new installs are logged as they happen, so the inventory reflects reality rather than last year's guess.

- **Usage is reviewed.** Software that nobody has opened in months is flagged, so those licences can be reclaimed or not renewed.

- **Compliance is checked.** Installations are compared against entitlements, so any gap is found and fixed internally — not by an auditor.

- **Contracts are tracked.** Renewal dates and licence terms are known in advance, so decisions are made deliberately instead of under pressure.

- **People follow the rules.** Staff request software through a known process, so nothing is installed off the books.

None of this requires deep technical skill to understand. It requires good information, clear ownership, and consistency — which is exactly what a structured SAM approach provides.

PART TWO

SAM versus ITAM

Two acronyms that beginners constantly mix up. Here we define both in simple terms and show, with examples, exactly how they relate.

First, what is ITAM?

IT Asset Management (ITAM) is the bigger umbrella. It is the practice of managing *all* of an organisation's IT assets across their whole life — from the moment they are requested and bought, through everyday use, to the day they are retired and disposed of.

"IT assets" means a lot of things: laptops, desktops, servers, phones, network equipment — and the software that runs on all of them. ITAM tries to keep one trustworthy picture of every one of these, so the organisation always knows what it owns, where it is, and what it's worth.

THINK OF IT THIS WAY

ITAM is managing **everything IT** — the physical machines and the software — across the full asset lifecycle. It is the whole toolbox, not a single tool.

Because ITAM covers so much ground, it is usually broken into more focused areas. Two of the most important are **Hardware Asset Management** and **Software Asset Management**.

So where does SAM fit?

Software Asset Management (SAM) is the part of ITAM that focuses purely on *software* — the licences, subscriptions, installations and usage we met in Part 1. It is a specialist discipline inside the broader ITAM family.

Why does software need its own discipline? Because software is uniquely tricky. A laptop is a physical thing you can count once. A software licence is an *agreement* with rules — how many people can use it, on how many machines, in which countries, for how long. Those rules are easy to break by accident, and breaking them costs real money. That complexity is why SAM exists as a field of its own.



So the relationship is simple: **SAM is a specialised branch of ITAM.** All SAM is ITAM, but not all ITAM is SAM.

Half price on the CSAM certification

50% OFF

For a limited window, the Certified Software Asset Manager programme is available at 50% off. Lock in the discount before it closes.

Enroll Now

SAM vs ITAM, side by side

Here is the clearest way to hold the difference in your head.

	ITAM	SAM
Scope	All IT assets — hardware and software	Software only
Main question	What IT do we own and where is it?	Are we correctly & efficiently licensed?
Typical assets	Laptops, servers, phones, networks, software	Licences, subscriptions, cloud software
Biggest risk it manages	Lost, untracked or unsupported equipment	Licence non-compliance & audit fines
Biggest saving it finds	Reusing and retiring hardware well	Cutting unused & duplicate licences
Relationship	The umbrella	A branch under the umbrella

If a question is about a physical device *or* software, it's an ITAM question. If it's specifically about licences, entitlements or software usage, it's a SAM question.

A worked example: one laptop

Let's follow a single laptop to see ITAM, HAM and SAM working together.

HAM

The laptop itself is a hardware asset. Hardware asset management records its model, serial number, who it's assigned to, and when it should be replaced.

SAM

The operating system, the office suite and the design tool installed on it are software assets. SAM records which licences cover them, whether the company is entitled to those installs, and whether they're actually being used.

ITAM

Tying it together — the laptop plus all its software, tracked from purchase to retirement as one connected record — is IT asset management.

When the employee leaves, ITAM ensures the laptop is returned (HAM) and that its software licences are freed up and reassigned rather than wasted (SAM). One device, three perspectives, one joined-up picture.

Why beginners should learn SAM specifically

You might wonder: if ITAM is the bigger umbrella, why focus on SAM? Three reasons make SAM a smart place for a beginner to build expertise.

It's where the money is

Software is one of the largest and fastest-growing IT costs. Skilled SAM saves organisations real, measurable money — which makes the skill valuable.

It's where the risk is

Licence non-compliance leads to audits and fines. People who can prevent that are in demand across every industry.

It's learnable

SAM is built on clear, structured concepts — exactly the kind of thing you can study and certify in without years of prior experience.

That last point matters most for beginners: SAM rewards structured knowledge, and structured knowledge is something you can acquire deliberately.

Your 48-hour enrolment window is open

48 HOURS ONLY

This offer on the GSDC CSAM certification is valid for the next 48 hours. Secure your place today and start studying at your own pace.

Enroll Now

PART THREE

Licences & Compliance

The heart of SAM. We explain what a software licence actually is, the common types, and what 'compliance' really means — all in plain language.

What a software licence actually is

When a company "buys software", it usually isn't buying the software the way you'd buy a chair. It's buying a **licence** — permission to use that software under a specific set of rules set by the vendor. You own the right to use it; you don't own the software itself.

Those rules are the whole game in SAM. A licence might say how many people can use the software, how many devices it can be installed on, whether it can run in the cloud, for how long, and in which regions. Stay inside the rules and you're compliant. Step outside them — even by accident — and you owe the vendor.

THE MENTAL SHIFT

You don't buy software — you buy **permission to use it under conditions**. SAM is the practice of staying inside those conditions while paying for no more than you need.

Common licence types, in plain English

You'll meet many licensing models, but most are variations on a few simple ideas.

Per-user (named)	Tied to a specific person. If five people need it, you buy five — no matter how many devices each uses.
Per-device	Tied to a machine. Anyone using that device is covered, but the software is licensed to the hardware, not the person.
Subscription	You pay regularly (often yearly) to keep using it. Stop paying and the right to use it ends.
Perpetual	A one-time purchase to use that version forever — though updates and support may still cost extra.
Concurrent	A shared pool. You buy a number of simultaneous users, not a user each — handy when usage is occasional.

Knowing which model applies is essential: counting per-device licences as if they were per-user is one of the most common ways organisations accidentally fall out of compliance.

Built for your next career move

50% OFF

CSAM-certified professionals stand out to recruiters. Get the certification — plus job support, resume builder and LinkedIn enhancer — now at 50% off.

Enroll Now

What "compliance" really means

Licence compliance simply means your actual use of software matches what you're licensed for. If you have 100 licences and 100 installs, you're compliant. If you have 100 licences and 130 installs, you're not — and the difference is a liability.

Vendors care intensely about this because under-licensing is lost revenue for them. That's why they reserve the right to audit their customers — to check the numbers and recover what they're owed. Compliance, then, isn't bureaucratic box-ticking. It's the difference between a predictable software bill and a sudden, unbudgeted one.

Two ways to be non-compliant

Most people picture **under-licensing** — using more than you paid for. But **over-licensing** — paying for far more than you use — is just as common and quietly drains budgets. SAM tackles both: avoiding penalties on one side and waste on the other.

Over-licensed vs under-licensed

These two imbalances are the bread and butter of SAM. Both cost money — just in opposite ways. Here's how to tell them apart.

Under-licensed

You're using **more** than you've paid for.

- ✗ The compliance risk
- ✗ Exposed in a vendor audit
- ✗ Leads to back-charges & fines
- ✗ Fixed by buying up or removing installs

Over-licensed

You're paying for **more** than you use.

- The waste risk
- Invisible — never shows in an audit
- Quietly drains the budget every year
- Fixed by reclaiming & not renewing

A mature SAM practice keeps you in the narrow band between the two: enough licences to stay safe, but not a single one more than you need.

The true cost of getting it wrong

The headline cost of non-compliance is the fine — but it's rarely the only one. When an audit uncovers a gap, the bill arrives with company in tow.

- **Back-licensing fees.** You pay for the licences you should have had, often at full list price with no discount.
- **Penalties.** Many agreements allow extra charges on top of the missing licences.
- **Lost negotiating power.** Once you're caught short, you've lost the leverage to negotiate a better deal.
- **Time and disruption.** Audits pull staff away from their jobs for weeks to gather evidence and respond.
- **Reputation.** Repeated compliance failures strain the relationship with vendors you depend on.

Every one of these is avoidable with good SAM — which is precisely why the skill is valued.

LIMITED OFFER

A credential recognised worldwide

GSDC is an independent, vendor-neutral certification body trusted across industries. Add a globally respected SAM credential to your profile.

Enroll Now

PART FOUR

Software Audits

What a software audit is, who runs them, what actually happens, and a simple way for beginners to think about staying audit-ready.

What a software audit is

A **software audit** is a formal check — usually requested by a vendor — to confirm that an organisation is using software within the terms of its licences. Think of it as the vendor asking, "Show me you're only using what you paid for."

Audits can be triggered for many reasons: a routine schedule, a merger, unusual download activity, or simply being a large customer. Most large software vendors include the right to audit directly in their contracts, so for big organisations an audit is less a possibility than a certainty over time.

THE KEY INSIGHT

An audit doesn't create a compliance problem — it **reveals** one that was already there. The work of staying compliant happens long before any auditor calls.

The anatomy of a vendor audit

Audits vary, but most follow a recognisable shape. Knowing the steps removes much of the fear.

- 1 Notification.** The vendor formally announces the audit and its scope — which products and which time period.
- 2 Data collection.** Installation and usage data is gathered, sometimes with a tool, sometimes through questionnaires.
- 3 Reconciliation.** What's deployed is compared against what's licensed (your entitlements) to find any gap.
- 4 Findings.** The vendor presents the result — typically a number of unlicensed installs and the cost to settle them.
- 5 Settlement.** The organisation negotiates and pays, or removes the excess software, to return to compliance.

Organisations with strong SAM breeze through reconciliation — because they already know their numbers. Organisations without it discover their exposure for the first time, in the worst possible way.

No heavy prerequisites — start today

50% OFF

Self-paced study materials, practice questions and a final online exam. Beginner-friendly and now half price. There's never been a better time to certify.

Enroll Now

Staying audit-ready: a beginner's checklist

You don't need to be an expert to grasp what "audit-ready" looks like. It comes down to being able to answer a few questions confidently at any time.

- Do we have a complete, current list of the software we own licences for?
- Do we know exactly where each piece of software is installed?
- Can we show that installations don't exceed entitlements?
- Do we keep proof of purchase for every licence (invoices, agreements)?
- Do we review usage so unused licences are reclaimed, not forgotten?
- Is one person or team clearly responsible for keeping this accurate?

If the answer to all six is a confident "yes", an audit holds no terror. Building the knowledge and discipline to make that true is exactly what software asset management — and a structured certification in it — teaches you to do.

PART FIVE

Glossary & Next Steps

A plain-language glossary of the key terms from this guide — and a clear look at where the GSDC CSAM certification fits in.

Glossary of key terms

Audit	A formal vendor check that your software use matches your licences.
Compliance	A state where your actual software use is fully covered by the licences you hold.
Entitlement	What you are permitted to use — the licences and rights you've purchased.
HAM	Hardware Asset Management — the branch of ITAM focused on physical equipment.
Installation	A copy of software actually deployed on a device, used or not.
ITAM	IT Asset Management — managing all IT assets, hardware and software, across their lifecycle.
Licence	Permission to use software under a defined set of rules set by the vendor.
Lifecycle	An asset's full journey, from request and purchase through use to retirement.

Glossary of key terms (continued)

Over-licensed	Paying for more licences than you actually use — a source of wasted spend.
Perpetual licence	A one-time purchase granting the right to use a version indefinitely.
Reconciliation	Comparing what's installed against what's licensed to find any gap.
SAM	Software Asset Management — the branch of ITAM focused on software licences, usage and cost.
Subscription	A licence paid for on a recurring basis; use ends when payment stops.
True-up	Settling a gap by paying for the additional licences a review or audit has revealed.
Under-licensed	Using more software than you've paid for — the core compliance risk.
Usage	The software people genuinely open and rely on, versus what merely sits installed.

Where the CSAM certification fits

Everything in this guide — what SAM is, how it differs from ITAM, licences, compliance and audits — is the foundation. The **GSDC Certified Software Asset Manager (CSAM)** certification is how you turn that foundation into a recognised, career-ready credential.

CSAM is offered by the Global Skill Development Council, an independent, vendor-neutral certification body. Because it isn't tied to any single tool or vendor, the knowledge applies across organisations of every size and type. You study at your own pace with the materials provided, then sit an online exam to earn the credential — and the certification aligns with widely used best-practice frameworks such as ITIL, ISO/IEC 20000 and COBIT.

You've finished the beginner's guide

You now understand the language and the logic of software asset management. The natural next step is to make it official — and to gain the structured depth that turns understanding into expertise.

ENDS SOON

Last chance to lock in your discount

You've finished the guide — take the next step. Enrol in the GSDC Certified Software Asset Manager certification before this offer ends.

Enroll Now