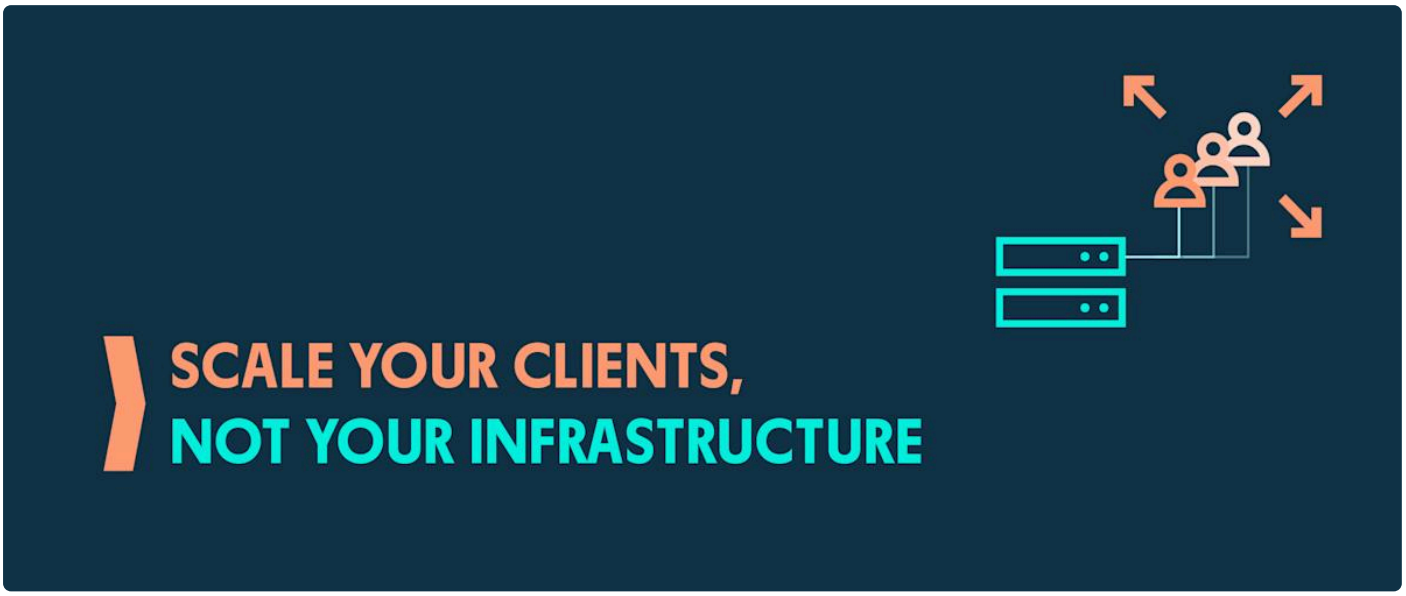
The Multi-Tenant ELK Solution: Scale Your Clients, Not Your Infrastructure

Daniel Gleim · 2026-02-12



The Multi-Tenant ELK Solution: Scale Your Clients, Not Your Infrastructure

For managed service providers and organizations offering Elasticsearch-based services, scalability is not just a question of performance. It is a question of isolation. Each customer expects their data, dashboards and alerts to be strictly separated from everyone else's while you want to avoid running a dedicated ELK stack per client.

Search Guard's multi-tenancy capabilities are designed to solve exactly this challenge. They make it possible to securely operate shared ELK environments without compromising isolation or control.

Why Multi-Tenancy Matters in ELK Environments

By default, Kibana is single-tenant. All users store dashboards, visualizations, and saved searches in a shared index. Without additional controls, this makes it difficult to safely operate a shared Kibana instance for multiple customers.

For service providers, this limitation often leads to one of two costly approaches:

- Running a separate Elasticsearch and Kibana deployment per customer
- Maintaining multiple Kibana instances connected to the same cluster

Both options increase operational complexity, infrastructure cost, and maintenance effort.

Search Guard's approach to multi-tenancy removes this trade-off by enabling logical separation inside a single Elasticsearch cluster and Kibana instance without sacrificing security or control.

How Multi-Tenancy Is Implemented in Kibana

Search Guard extends Kibana with the concept of **tenants**. A tenant is a logically isolated workspace where Kibana saved objects, such as dashboards, visualizations, and searches, are stored separately from other tenants.

Each tenant is backed by its own Elasticsearch index. When a user selects a tenant, all saved objects they create or access are automatically stored in that tenant's index. Users assigned to different tenants cannot see or access each other's Kibana objects.

From the user's perspective, working in a tenant feels like using a completely separate Kibana instance. From an operator's perspective, it remains a single, centrally managed deployment powered by Search Guard.

Built-In Tenants for Practical Use

To cover common use cases, Search Guard utilizes a flexible tenant model designed for both collaboration and strict isolation:

- **Global tenant:** A shared space for dashboards or visualizations that should be visible to multiple users or teams. This is the default "public square" for your organization.
- **Custom tenants:** Administrators can configure any number of isolated workspaces, for example specific tenants for 'Marketing', 'Development', or individual customers.

Unlike legacy approaches that generate a private index for every single user (often leading to "shard explosion" and instability), Search Guard's modern approach focuses on managed custom tenants. This ensures that the cluster remains performant and governable, even as you scale to thousands of users.

In addition, administrators can create any number of custom tenants. For example one per customer, department, or project. This flexibility makes the tenant model well suited for environments that grow over time.

Role-Based Tenant Access

Tenant isolation is enforced through role-based access control.

Each role can be configured with explicit tenant permissions, defining:

- Which tenants the role can access
- Whether access is read-only or read-write

Users may have multiple roles, and their effective tenant access is the union of all assigned role permissions. Importantly, Kibana only displays tenants that the user is authorized to access. If a tenant is not assigned through a role, it effectively does not exist from the user's perspective.

With Search Guard, onboarding new customers becomes a configuration task rather than an infrastructure task: create a tenant, assign it to a role, and grant that role to the customer's users.

Security Beyond Kibana

Multi-tenancy in Search Guard does not stop at the Kibana layer.

All access control is enforced at the Elasticsearch level. Even if a user attempts to bypass Kibana and interact directly with Elasticsearch APIs, the same authorization rules apply. This prevents accidental or malicious cross-tenant access regardless of how the cluster is accessed.

In practice, this means that tenant isolation is consistent across the entire data path, including:

- Encrypted REST and inter-node communication
- Authentication via common enterprise identity providers
- Fine-grained permissions at index, document, and field level
- Audit logging for security-relevant events

As a result, isolation applies not only to dashboards and visualizations, but also to the underlying data stored in Elasticsearch indices.

Multi-Tenancy for Alerting and Operations

Search Guard's multi-tenancy model also extends to alerting.

Alert definitions are isolated per tenant, meaning users can only view and manage alerts created within the tenants they are authorized to access. Alert execution is isolated as well, ensuring that alerting activity from one tenant does not interfere with others.

For service providers, this is essential. Customers expect full ownership over their alerts without visibility into other tenants' operational signals or configurations.

Scaling Without Adding Servers

The practical impact of Search Guard's multi-tenancy is straightforward: you can serve more clients on the same infrastructure.

Instead of deploying one ELK stack per customer, service providers operate a shared Elasticsearch cluster sized for aggregate demand. New customers are added logically by creating tenants and roles, rather than physically by provisioning new servers.

Infrastructure scales only when resource usage requires it, not when a new customer signs up. This leads to:

- Higher utilization of existing hardware
- Fewer clusters to operate, upgrade, and secure
- Faster customer onboarding
- Lower operational overhead per tenant

Because Search Guard employs an unlimited node licensing model, this approach aligns perfectly with your business growth. Unlike other solutions that penalize you for adding hardware, Search Guard allows you to add as many data nodes as necessary to handle increased load without triggering additional licensing costs. You pay for the capability, not the server count, giving you full control over your margins as you onboard new clients.

A Practical Foundation for Managed ELK Services

Multi-tenancy is not a convenience feature, it is a prerequisite for operating ELK as a service.

By isolating Kibana workspaces through tenants, enforcing access through roles, and applying consistent authorization at the Elasticsearch layer, Search Guard enables shared ELK environments without shared risk.

For service providers looking to scale their offerings without scaling their server count, Search Guard's multi-tenancy solution provides a proven, security-first foundation for multi-tenant Elasticsearch and Kibana deployments. Check out our documentation to learn more!

This article was published on the Search Guard blog: https://search-guard.com/blog/the-multi-tenant-elk-solution-scale-your-clients-not-your-infrastructure/