NeuralMesh Multitenancy Technical Guide

WEKA NeuralMesh Native Multitenancy

NeuralMesh multitenancy provides a scalable and efficient foundation for multi-tenant environments. It enables elastic resource sharing, tenant-scoped networking and security, and unified operations. The result is a platform that supports high-density consolidation while maintaining the control and performance required for demanding workloads.

Overview

NeuralMesh multitenancy introduces a tenant-native model that allows multiple independent environments to operate within a single worker cluster without sacrificing isolation, performance, or control. This model is designed to solve a core customer challenge: how to consolidate infrastructure for efficiency while still maintaining strict separation between users, teams, or customers.

Instead of forcing a tradeoff between isolation and utilization, the platform enables both. Customers can run many tenants in a single cluster while preserving predictable performance, strong security boundaries, and operational simplicity.

NeuralMesh Multitenancy Architecture - The Tenant Model

A tenant is a first-class isolation domain that behaves like an independent storage environment within the cluster. Each tenant encapsulates its own capacity limits, policies, identities, and protocol resources. From a customer perspective, this allows different teams or external customers to operate independently while sharing the same infrastructure.

Each tenant defines its own policy domain, including encryption, authentication methods, and quality of service controls. This allows organizations to support different security postures and compliance requirements within the same platform. The model is extensible, with additional capabilities such as replication and advanced controls being introduced at the tenant level over time.

Resources are not statically reserved per tenant. Compute, memory, and capacity are shared elastically across tenants, while policy boundaries enforce fairness and isolation. This allows

tenants to start small and scale without re-architecting the environment, supporting both long-tail onboarding and growth over time.

Each tenant represents the full unit of operation within the platform, encompassing filesystems, object buckets, protocol access, snapshots, and policy enforcement. This allows tenants to function as self-contained environments for both day-one provisioning and day-two operations, without requiring coordination at the cluster level.

Tenant as a First-Class Entity

Native Multitenancy introduces a new Tenant entity. Multiple tenants coexist within a single worker cluster while maintaining logical isolation. Native multi-tennancy supports 1K+ tenants per cluster and tenants as small as 1TB.

Tenant Capabilities

Each tenant may include:

- Capacity limits and quotas
- Multi-protocol support (POSIX, S3, SMB, NFS)
- Filesystems, object buckets, and protocol exports
- Tenant-level policies for encryption, snapshots, security, and QoS

Virtualized Networking with VRDF

VRDF enables fully virtualized networking within WEKA. Each tenant is assigned one or more network spaces that may include VLANs, custom IP ranges, protocol endpoints, and overlapping IP support. Networking is managed within WEKA rather than relying on host operating systems.

Identity and Authentication

Tenants support independent identity providers, including LDAP and Active Directory, enabling per-tenant authentication and authorization across supported protocols.

Virtualized RDMA Data Fabric (VRDF)

Network spaces are implemented through WEKA's Virtualized RDMA Data Fabric (VRDF), enabling data-plane network virtualization rather than relying on host-level segmentation. Each tenant can define its own VLANs and IP ranges, including overlapping address spaces.

This gives customers the ability to onboard tenants without requiring network redesign or coordination. Each tenant can operate using its own addressing scheme, which aligns with how environments are structured in cloud platforms.

IP addresses are allocated automatically from defined ranges, using sequential allocation and reusing available gaps. Customers define ranges sized for both current needs and future growth.

Network space access enforcement ensures that requests arrive through the correct tenant network. When enabled, the system validates the origin of mount requests and rejects those that originate outside the assigned network space, even if valid credentials are provided. This enforces isolation directly in the data path and reduces the risk of cross-tenant access due to misconfiguration.

Security and Isolation

Isolation is enforced through a combination of networking, authentication, and policy controls. Filesystem authentication ensures that only authorized users can access data. Network enforcement ensures that access originates from the correct tenant context. Access control policies define what operations are permitted.

These controls work together to provide layered protection. Network validation ensures correct origin, while identity and policy controls govern access. This reduces reliance on any single control mechanism and strengthens overall security.

Each tenant can integrate with its own identity providers and key management systems. This enables independent authentication domains and encryption policies within the same cluster, which is critical for multi-organization and regulated environments.

Control Plane and Operational Responsibilities

All tenants operate within a single control plane with a unified API surface and shared observability. This eliminates the need to deploy, upgrade, and manage multiple clusters to achieve isolation.

For customers, this translates directly into lower operational overhead, simpler automation, and fewer failure domains. A single system can be managed consistently, even as the number of tenants grows.

Administrative responsibilities are clearly separated. Cluster administrators manage infrastructure, tenant lifecycle, quotas, quality of service limits, security policies, and network spaces. Tenant administrators manage resources within their tenant, including filesystems, snapshots, and object storage. Tenant administrators do not have visibility into other tenants or cluster-level configuration.

The model enforces separation of duties by default. Cluster administrators do not access tenant data, which aligns with enterprise governance and service provider requirements. Cluster administrators manage hardware, capacity ceilings, and performance limits. Tenant administrators manage only what exists inside their tenant—filesystems, snapshots, users—without visibility into other tenants or cluster internals.

Cluster Administrator Role

Cluster administrators manage infrastructure-level concerns such as hardware, cluster health, tenant creation, capacity limits, and performance boundaries.

Tenant Administrator Role

Tenant administrators manage resources within their assigned tenant, including filesystems, snapshots, and access controls, constrained by policies set at the cluster level.

Role-Based Access Control (RBAC)

The tenant model is designed to support granular role-based access control beyond the basic separation of cluster and tenant administrators. Roles are scoped to either the cluster or a tenant, with the ability to define permissions around specific operations such as filesystem management or data access. This enables organizations to enforce governance models where infrastructure teams, tenant operators, and data users have clearly defined and non-overlapping responsibilities, with future evolution toward more fine-grained, attribute-based controls.

Protocol Behavior

The tenant abstraction is designed to span all supported protocols, providing a consistent model for file and object services within the same environment. This ensures that as additional protocols are introduced, they inherit the same isolation, policy, and operational boundaries defined at the tenant level.

POSIX access is fully tenant-scoped and operates within tenant network spaces. This ensures that high-performance workloads remain aligned with tenant-defined networking and isolation boundaries, which is critical for AI and HPC use cases.

S3 is implemented as a shared service with tenant isolation enforced through credentials rather than network segmentation. All tenants access S3 through the backend network, and tenant identity is embedded in the credential structure. Bucket names are unique across the cluster.

The S3 credential model separates API credentials from datapath credentials. This allows independent rotation, avoids disruption to running workloads, and enables multiple access keys per user. This aligns with common cloud operational practices while maintaining tenant isolation.

Scaling and Efficiency

The platform supports hundreds of tenants per cluster, with a roadmap to exceed one thousand. Tenants can be very small, starting around one terabyte, and scale as needed without re-architecture.

Cluster sizing for multi-tenant environments follows the same principles as single-tenant deployments, with no meaningful per-tenant CPU or memory overhead introduced by the tenancy model itself. This allows customers to scale tenant count without incurring hidden infrastructure costs, making high-density consolidation both technically and economically viable.

The result is improved economics. Customers can achieve higher utilization and better return on infrastructure investment while maintaining predictable performance through policy-based isolation.

Kubernetes-Native Storage and Composable Infrastructure for Multitenancy

Modern multi-tenant environments require more than logical isolation. They require infrastructure that can dynamically allocate, scale, and isolate resources without introducing inefficiency or operational complexity. In practice, this means storage must behave like the rest of the platform: declarative, elastic, and aware of tenant boundaries.

WEKA extends its Multitenancy model through two complementary capabilities: a Kubernetes-native Operator that aligns storage lifecycle management with cluster operations, and a composable storage architecture that enables fine-grained sharing of physical resources across tenants. Together, these capabilities ensure that storage scales with compute, while maintaining strong isolation and high utilization across diverse workloads.

Kubernetes Operator

The WEKA Kubernetes Operator integrates storage directly into the Kubernetes control plane, allowing it to be deployed and managed using the same declarative workflows as application infrastructure. Instead of operating as an external system, WEKA clusters are exposed as native Kubernetes resources, enabling platform teams to define, provision, and manage storage through familiar constructs such as custom resources, Helm charts, and StorageClasses.

This integration is foundational to Multitenancy. Storage resources can be aligned with Kubernetes namespaces and governed through RBAC, ensuring that tenants consume storage in a controlled and isolated manner while platform teams maintain centralized oversight. Workloads request storage through standard PersistentVolumeClaims, allowing seamless integration without requiring application changes.

Because the platform is built as containerized microservices, lifecycle operations such as upgrades, scaling, and failure recovery are handled in a granular and non-disruptive way. Individual components can be updated or restarted independently, ensuring continuous availability even as

the environment evolves. The Operator manages the full lifecycle of the storage cluster, including deployment, configuration, capacity expansion, node replacement, and rolling upgrades.

Within multi-tenant environments, workloads often have different performance and capacity requirements. The Operator enables these requirements to be expressed declaratively, allowing per-workload performance characteristics and quality of service controls to be applied without manual intervention. This ensures that storage behaves as a governed, shared resource rather than a static allocation tied to infrastructure boundaries.

By deploying WEKA through the Operator, customers benefit from a distributed storage architecture that scales horizontally with the Kubernetes platform. As workloads increase in parallelism, storage performance scales alongside them, preventing bottlenecks that would otherwise impact tenant isolation and overall system efficiency.

Composable Clusters

Composable Clusters extend the Multitenancy model by decoupling physical storage resources from logical cluster and tenant boundaries. Instead of dedicating entire drives or nodes to individual tenants, capacity and performance are partitioned and shared dynamically across multiple tenants within the same infrastructure.

This allows tenants to consume only the resources they need, when they need them, rather than requiring fixed allocations sized for peak demand. The result is higher utilization, improved efficiency, and greater flexibility in how infrastructure is consumed.

Within this model, WEKA supports hybrid flash configurations that combine high-performance TLC media with high-capacity QLC media. Data placement is handled intelligently, with latency-sensitive workloads directed to performance tiers and capacity-oriented workloads placed on cost-efficient media. This ensures that different tenant workloads can coexist without competing for the same performance characteristics.

A key enabler of composability is the ability to subdivide physical drives into smaller logical units that can be shared across tenants. Rather than binding entire drives to a single cluster, storage is segmented into granular allocation units that can be distributed across multiple tenants. This allows smaller tenants to be supported efficiently while maintaining high overall utilization of physical resources.

Composable infrastructure is managed through the same Kubernetes-native model used by the Operator, allowing platform teams to define how resources are allocated using declarative configuration. Parameters such as the balance between performance and capacity media and the size of allocation units can be tuned globally or per cluster, enabling precise control over how storage resources are consumed.

Together, these capabilities allow organizations to achieve high-density multitenancy without sacrificing performance or isolation. Storage becomes a flexible pool of resources that can be dynamically allocated, enabling tenants to scale independently while the platform maintains efficiency and control.

Quality of Service

Quality of service is defined at the tenant level, with controls for throughput and IOPS. These controls allow operators to enforce fairness and prevent individual tenants from impacting others in shared environments. The model is evolving toward policy-based QoS and minimum guarantees.

Hardware Requirements

The solution requires ConnectX 7 or newer network interface cards on the backend network. This enables the advanced data path capabilities required for tenant isolation and performance at scale.

Observability

Tenant-level observability includes throughput, IOPS, and latency metrics, integrated into existing WEKA monitoring tools. This allows operators to monitor tenant behavior, troubleshoot issues, and support chargeback or showback models in shared environments.

Tenants are not limited to initial provisioning but support ongoing operational workflows such as snapshot management, lifecycle policies, and performance controls. This ensures that each tenant can be managed independently over time without requiring cluster-level intervention.

Deployment Model

Multitenancy operates consistently on bare metal and in Kubernetes environments. Kubernetes is treated as a deployment option rather than a requirement, allowing customers flexibility in how they deploy the platform.

Multitenancy is best suited for environments that require high tenant density, support for a wide range of tenant sizes, or tenant-specific networking and security requirements. It is particularly effective for service providers onboarding many customers, enterprise platform teams supporting multiple internal groups, and AI or HPC environments where multiple workloads must share infrastructure without interference. The model is optimized for scenarios where operational simplicity, consolidation, and flexibility are more valuable than strict physical separation.

Delivering True Value

Multitenancy removes the traditional tradeoff between isolation and efficiency. Customers can consolidate workloads while maintaining strong isolation, predictable performance, and clear operational boundaries.

The combination of VRDF-based network virtualization, network space enforcement, and identity-based controls provides strong isolation without requiring separate infrastructure. The single control plane reduces operational overhead and simplifies scaling.

For service providers, this improves margins and accelerates tenant onboarding. For enterprise platform teams, it enables internal multitenancy without creating infrastructure silos. For AI and HPC environments, it allows multiple teams to share high-performance infrastructure without interference.

Summary

NeuralMesh multitenancy provides a scalable and efficient foundation for multi-tenant environments. It enables elastic resource sharing, tenant-scoped networking and security, and unified operations. The result is a platform that supports high-density consolidation while maintaining the control and performance required for demanding workloads.

Take the Next Step with NeuralMesh

Dig further into NeuralMesh by checking out our Reference Architecture. If you are interested in discussing how NeuralMesh can help you, click here to contact us.