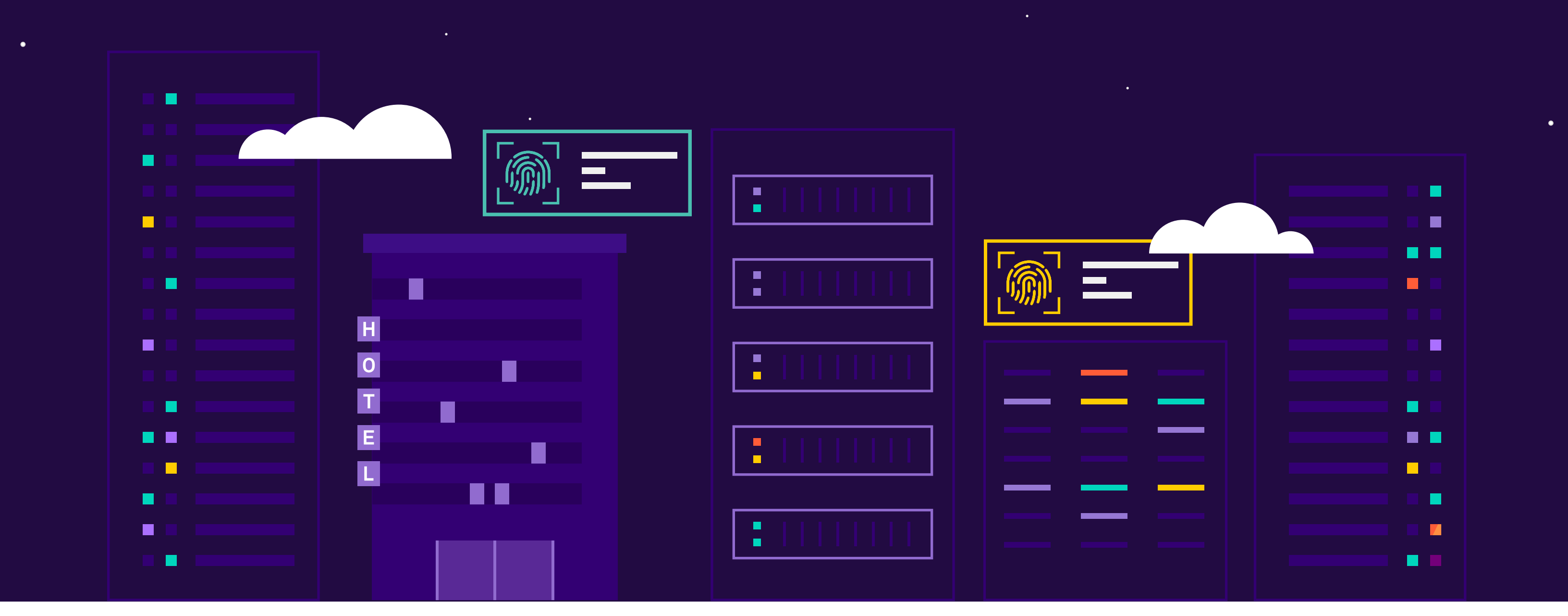
Preventing Noisy and Nosy Neighbors

in Multi-tenant Amazon EKS Environments with D2iQ

Noisy and nosy neighbors aren't exclusive to just your physical living space. In an enterprise environment, a lack of centralized governance and federated management of Amazon EKS clusters can limit application scalability and performance. Learn about these multi-tenancy issues and how to effectively solve them.



Who's Who in an Enterprise Multi-Tenant Environment

Enterprise

A single company that has many tenants using Amazon EKS and would like to share cluster and administrative resources.

Tenant

Business units or development teams within the larger organization who are developing a single workload in a shared Amazon EKS cluster.

Platform Team

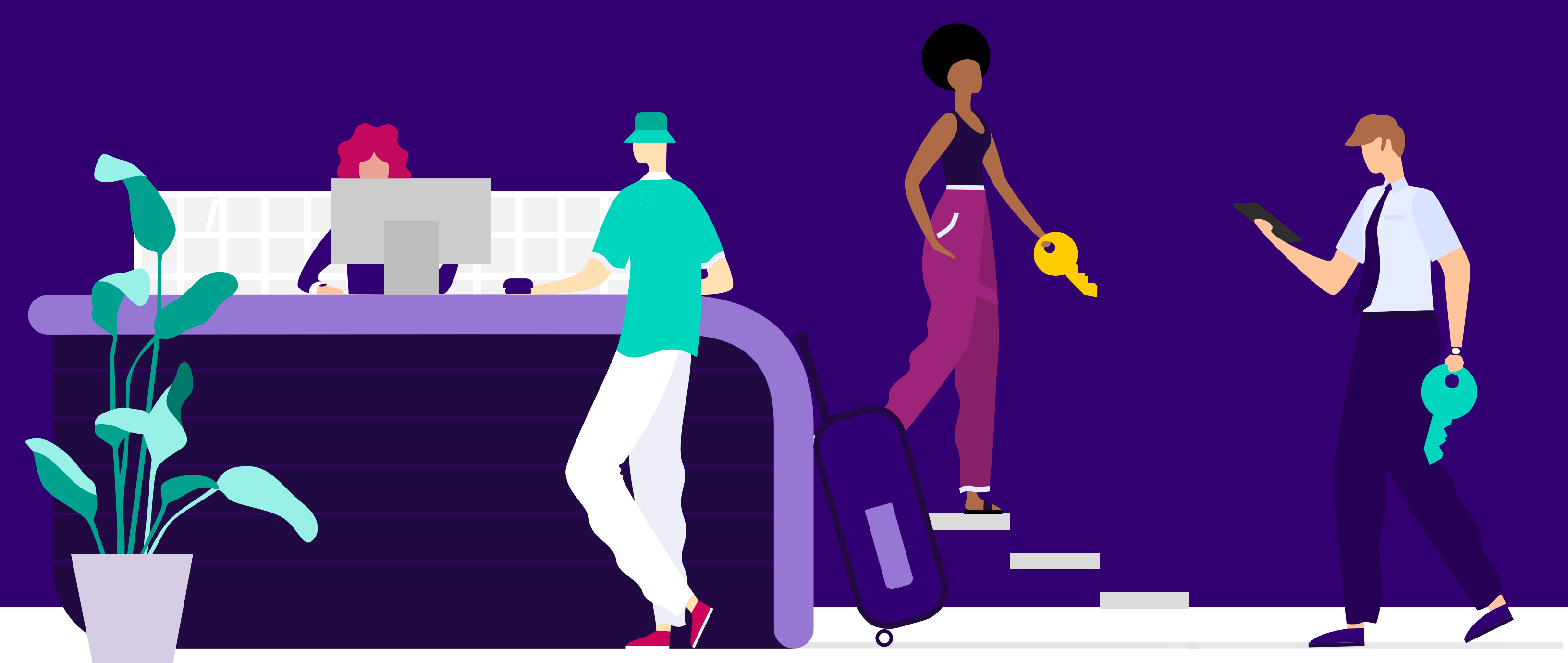
Managers responsible for all Amazon EKS clusters across the entire infrastructure who define the amount of resources each tenant team can use.

The Impact of Noisy and Nosy Neighbors

As an enterprise adopts new cloud clusters, it can introduce all kinds of complexities around visibility, management, and security.

Noisy Neighbor Sharing a cloud cluster with multiple tenants → negatively impacts the bandwidth and performance of workloads.

Nosy Neighbor Lack of centralized governance → unable to identify tenants, configure access to resources, perform compliance checks.



Kubernetes Multi-tenancy Best Practices

Following are best practices for IT admins to safely and efficiently set up multiple multi-tenant clusters on Amazon EKS.

STEP 1

Categorize Namespaces

Categorize namespaces to separate workloads, tenants, and applications so that updates and changes are contained within a specific namespace and doesn't impact bandwidth or performance.

STEP 2

Determine Cluster Personas

Create a hierarchy of cluster personas that specifies their responsibilities and account privileges. This is the first step to enabling Role-Based Access Control (RBAC) in Kubernetes.

STEP 3

Enable RBAC

Create cluster personas using Roles and RoleBindings in RBAC to lock out nosy neighbors and restrict unnecessary network access.

STEP 4

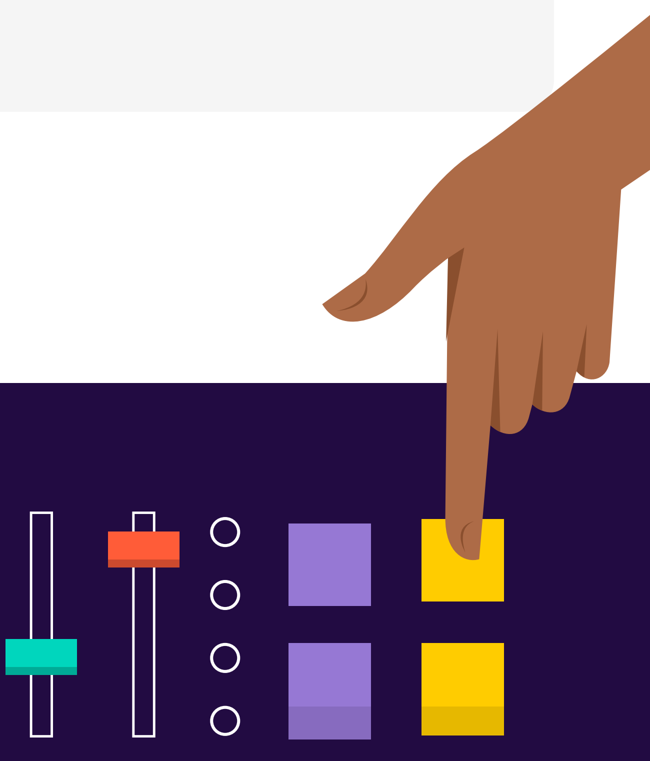
Isolate Namespaces using Network

Apply a default network policy to all tenant namespaces, which blocks access from unwanted neighbors.

STEP 5

Improve Resource Utilization with Resource Quotas

Apply resource quotas to each namespace to control and limit computing resources, such as CPU and memory, so that neighbors have access to the shared resources they agreed upon.



Amazon EKS + D2iQ

A Better Together Alliance for Effective Multi-cluster and Multi-tenant Management

Using the D2iQ Kubernetes Platform (DKP) with Amazon EKS can simplify Kubernetes multi-tenancy and reduce the risk of noisy and nosy neighbors. The unique benefits include:

Unified DKP User Interface

DKP provides the capability to provision EKS clusters using the DKP UI, which radically simplifies EKS provisioning.

Multi-cluster Management

DKP provides out of the box configured observability, monitoring, and alerting for all EKS clusters on any cloud or on-premise environment from a single central point of control.

Centralized Governance and User Access Control

Create federated rules for network policy, quotas, identity management, RBAC secrets, and ConfigMaps to enable consistent multi-cluster, multi-cloud, and multi-tenant management.

Multi-tenant Observability

Within DKP multi-cluster management are three tenancy views:

- **Global** - IT admins manage all clusters across the entire infrastructure, while retaining an overall organization-level view of all Amazon EKS clusters in operation.
- **Workspace** - business units within the larger organization have independent visibility into their set of Amazon EKS clusters that are isolated from other business units.
- **Project** - a development team within the larger organization has visibility into their project namespace and not into other development teams' project namespaces.

To learn how D2iQ can help you enhance your AWS EKS environment, **contact the experts at D2iQ.**

D2IQ