[**https://github.com/minio/minio/issues/21540**](https://github.com/minio/minio/issues/21540)

**Helm: service — add extraLabels for both the main and console services**

**Case Study 1 – Question 8**

Background

Helm is a package manager for Kubernetes, used to deploy and manage complex apps like MinIO using templates called charts.

The MinIO Helm Chart defines multiple Kubernetes resources — such as:

Main service → handles MinIO API traffic (S3-compatible endpoints).

Console service → handles the MinIO Web UI traffic.

Labels in Kubernetes are key–value pairs used for:

Organizing and selecting resources (app=minio, tier=backend, etc.),

Enabling monitoring, metrics, or security policies.

Sometimes, you need to add custom labels to these services (for example, for Prometheus, Istio, or internal auditing).

By default, older versions of the MinIO Helm chart did not support adding extra custom labels to the service definitions directly.

You could label pods or deployments, but services (main and console) lacked this flexibility.

So, when teams needed to add custom labels (for example team=storage or env=prod), they couldn’t without modifying the chart source.

Recent Helm chart updates (or custom chart modifications) introduced the ability to use the parameter:

service:

extraLabels:

key1: value1

key2: value2

consoleService:

extraLabels:

key1: value1

key2: value2

This allows you to define extra labels for both:

The main MinIO service (service:)

The console service (consoleService:)

These labels are then applied automatically when Helm renders and deploys the chart.

service:

type: ClusterIP

port: 9000

extraLabels:

app.kubernetes.io/part-of: storage-platform

team: devops

consoleService:

type: ClusterIP

port: 9001

extraLabels:

app.kubernetes.io/component: console

managed-by: helm

When you install or upgrade with:

helm upgrade --install minio minio/minio -f values.yaml

Both the main MinIO service and the console service will now have your additional labels.

**Conclusion**

Monitoring systems (Prometheus, Datadog, etc.)

Network policies or service meshes (Istio, Linkerd)

Deployment tracking tools (ArgoCD, Flux)

Avoids manual post-install kubectl label commands.

Keeps all label definitions centralized in the Helm values.yaml.