CASE STUDY · RAZORPAY

How Razorpay Automates RBI Compliance on Kubernetes with Kyverno

Razorpay meets the Reserve Bank of India's Payment Aggregator Master Directions across more than 7,000 Kubernetes nodes by writing its regulatory controls as Kyverno policies. The same policies run in CI, at admission and in periodic scans, so every change is checked against RBI requirements and auditors see live enforcement records instead of spreadsheets.

7,000+	$180 billion	99.99%	40+
KUBERNETES NODES	ANNUALIZED PAYMENTS PROCESSED	AVAILABILITY THROUGH FESTIVAL AND FLASH-SALE PEAKS	PRODUCTS LAUNCHED A YEAR

CUSTOMER	REGULATION	PLATFORM	PRODUCT	IMAGE SIGNING
Razorpay	RBI Payment Aggregator Master Directions	7,000+ Kubernetes nodes	Nirmata Enterprise for Kyverno	Sigstore Cosign

§01 CHALLENGE

What does RBI require of a payment aggregator's infrastructure?

Razorpay processes an annualized $180 billion in payments for millions of businesses and more than 300 million consumers, and powers payments for 80 of India's 100 unicorns. As an authorized payment aggregator, it answers to RBI's PA Master Directions. Three requirements land directly on the platform team:

REQUIREMENT	WHAT IT MEANS ON KUBERNETES
Data localization	Transaction processing systems and their logs stay inside India, within enforced boundaries
Zero-trust access and microsegmentation	Core transaction clusters are isolated, with least-privilege access and segregation of duties
Continuous auditability	Point-in-time screenshots and spreadsheet approvals no longer count; compliance must be provable at any moment

Razorpay used to validate these requirements with architecture reviews, deployment checklists and engineering approvals. That stopped holding at 7,000-plus nodes, dozens of commits a day and autoscaling that adds hundreds of nodes in minutes during festival sales. Reviewers could not read YAML as fast as developers shipped it, and a review of a static manifest says little about ephemeral containers that spin up and down on demand.

Razorpay still launches more than 40 products a year. Slowing that down to satisfy the auditor was not an option, so the controls had to move into the platform itself.

§02 SOLUTION

How Razorpay enforces RBI controls with Kyverno

Razorpay worked with Nirmata to adopt Nirmata Enterprise for Kyverno, the enterprise distribution of the CNCF-graduated policy engine. Kyverno policies are Kubernetes resources written in YAML, so developers read them the way they read their own manifests. The same policies run at three points between a git push and a running container:

LAYER	WHERE IT RUNS	WHAT IT CATCHES
Proactive	IaC and GitOps pipelines	Non-compliant Kubernetes manifests and cloud templates, including S3 and data-residency boundaries, before anything is provisioned
Preventive	Kubernetes API server (admission)	Root containers, wildcard RBAC bindings, missing compliance metadata and unsigned images, at the moment of deployment
Detective	Periodic cluster and cloud scans	Drift, policy bypasses, and existing resources that fall out of line when a policy changes

At the API server, Kyverno **validates** (a pod that runs as root is rejected with an error the developer sees in their own console), **mutates** (secure defaults and compliance labels are added automatically), **generates** (each new namespace gets an isolating NetworkPolicy and Istio authorization settings) and **verifies** (only images built in Razorpay's CI pipelines and signed with Sigstore Cosign are admitted).

Mutation and generation are what make compliance the default path rather than a gate. A developer creating a namespace does nothing extra, and the namespace arrives isolated.

§03 RESULTS

What changed for Razorpay

01. **Autoscaling stays compliant.** Kyverno evaluates policy at admission in milliseconds, so the thousands of containers added during a traffic spike inherit RBI zero-trust rules at creation, with no added latency in payment pipelines.

02. **Audit evidence is generated, not gathered.** Kyverno's policy events feed continuous compliance dashboards. When auditors ask about microsegmentation or data isolation, Razorpay shows enforcement logs instead of checklists.
03. **Developers stopped waiting on committees.** A violation arrives as a readable message about a YAML field, inside the developer's normal workflow.
04. **Only signed images run in production.** Unsigned or unapproved images are rejected at admission, which narrows the window for exploits against unpatched dependencies.

Razorpay's next steps are to contribute reusable Kyverno policy templates for financial regulation back to the community, use AI agents to test policies and gather audit evidence on a schedule, and build a multi-cluster compliance view on OpenReports.

nirmata

Nirmata created Kyverno, the CNCF graduated policy engine for Kubernetes. We help platform and security teams govern clusters, workloads, and AI agents with policy as code.

WATCH THE TALKS
Keynote: Automating RBI Compliance With Unified Policy-as-Code, Jim Bugwadia and Ashwath Kumar
youtube.com/watch?v=MeJCUVnSvJM

WATCH THE TALKS
Cloud Native Live: Compliance at Scale with Kyverno, Razorpay and Nirmata
youtube.com/watch?v=DZ6iC4h32Gc

READ THE FULL STORY
How Razorpay Automates RBI Compliance on Kubernetes with Kyverno
nirmata.com/2026/10/22/how-razorpay-automates-rbi-compliance-on-kubernetes-with-kyverno

SOURCE
CNCF case study: Razorpay
cncf.io/case-studies/razorpay

TALK TO THE KYVERNO MAINTAINERS
nirmata.com/request-a-demo

© 2026 Nirmata, Inc. Kyverno is a CNCF graduated project. Kubernetes is a registered trademark of The Linux Foundation.