



Kubernetes

Ingress Explained

(Scenario-Based)

How Ingress simplifies access to multiple applications in a Kubernetes cluster.

by Dinesh kumar
DevOps Engineer

The Scenario: 3 Apps, 1 Cluster

We have 3 applications inside our cluster, and all of them need external access.



app1 – Customer Portal

Main application for customer interaction.



app2 – Admin Portal

Internal tool for administrators.

app3 – Analytics Dashboard

Displays metrics and business intelligence.

The Problem: Scaling Access

How do we expose these 3 apps? Let's look at the "old ways" and their challenges.

Option 1: NodePort Challenges

How It Works

Assigns a port on each Node (30000–32767).

app1 → 30080

app2 → 30081

app3 → 30082

The Problems

- ✗ Too many ports to open/manage.
- ✗ Not user-friendly (must remember ports).
- ✗ Hard to scale (port conflicts).
- ✗ Messy, not ideal for production.

Option 2: LoadBalancer Challenges

How It Works

Each service gets its **own** external IP.

app1 → 34.120.11.10

app2 → 34.120.11.11

app3 → 34.120.11.12

The Problems

- ✗ Expensive (each LB costs money).
- ✗ Many IPs to manage (DNS complexity).
- ✗ Hard to configure central rules.
- ✗ Not scalable for many apps.

Ingress to the Rescue

There must be a smarter, cleaner, and more efficient way. This is where Ingress comes in.

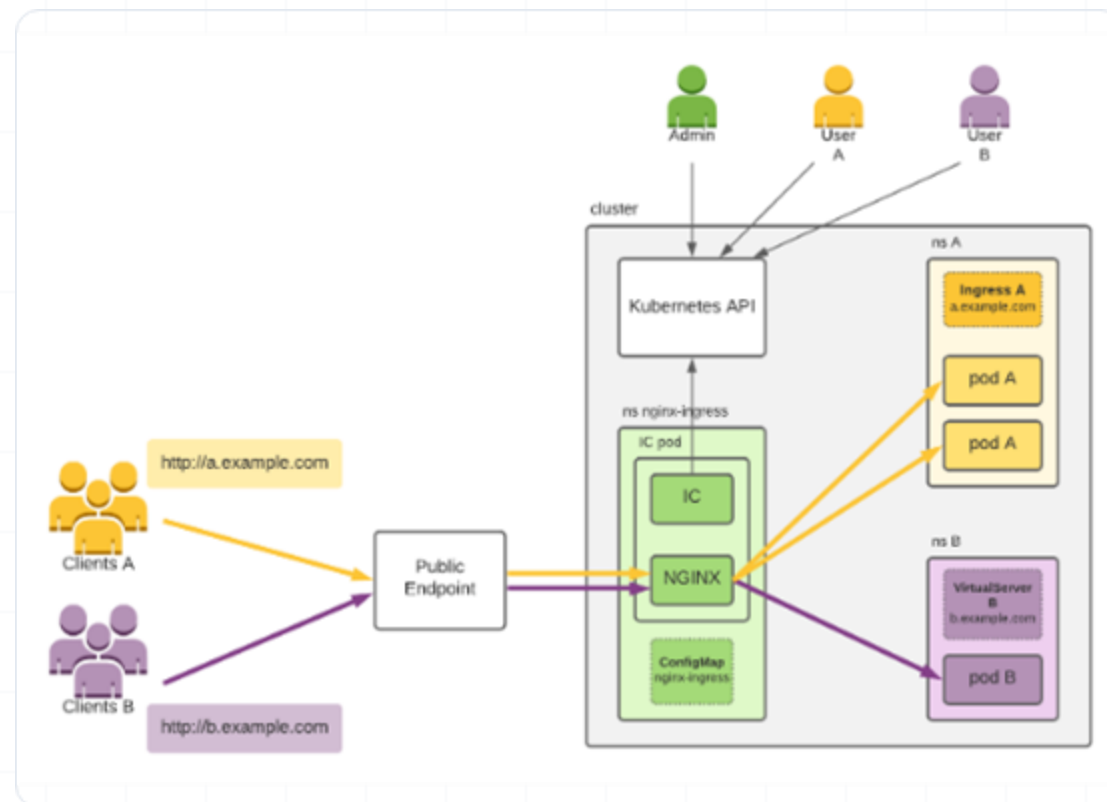
What Ingress Does

Ingress acts like a **smart entry gate** or traffic controller for your cluster.

Uses **one** external IP & LoadBalancer.

Routes traffic to **multiple** services.

Intelligent path-based or host-based routing.



After Ingress: The Solution in Action

Using One Domain (Path-Based)

`https://mycompany.com`

`/customer → app1`

`/admin → app2`

`/analytics → app3`

Or Subdomains (Host-Based)

`customer.mycompany.com → app1`

`admin.mycompany.com → app2`

`analytics.mycompany.com → app3`

All using ONE Load Balancer!

How Ingress Solves Our Problems

- ✓ **One entry point:** Simplifies firewall and DNS.
- ✓ **Clean, user-friendly URLs:** No more port numbers.
- ✓ **Only 1 LoadBalancer = Cost saving:** Huge reduction in cloud costs.
- ✓ **Easy routing with YAML:** Define all rules in one manifest.
- ✓ **Built-in SSL/TLS termination:** Manage certificates in one place.
- ✓ **Works like an API Gateway:** A central hub for all your routing.

Visual Comparison: Before vs. After

Before Ingress

(LoadBalancer Method)

User → LB1 (34...10) → **app1**

User → LB2 (34...11) → **app2**

User → LB3 (34...12) → **app3**

OR

(NodePort Method)

User → Node:30080 / 30081 / 30082

After Ingress

User → **One IP** → Ingress Controller

/customer → **app1**

/admin → **app2**

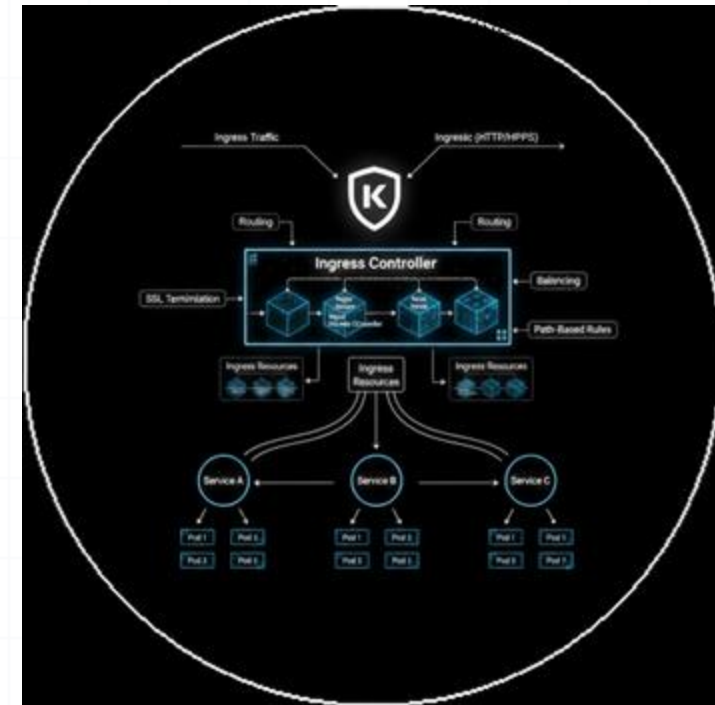
/reports → **app3**

Real-Life Analogy

Ingress = Hotel Main Entrance

Instead of having separate gates for every room, the hotel has one main entrance.

A receptionist (**Ingress Controller**) guides guests to the correct room — just like Ingress routes traffic to the right service.



“Ingress turns complexity into clarity — one gateway, infinite possibilities.”

Thank you for your attention.