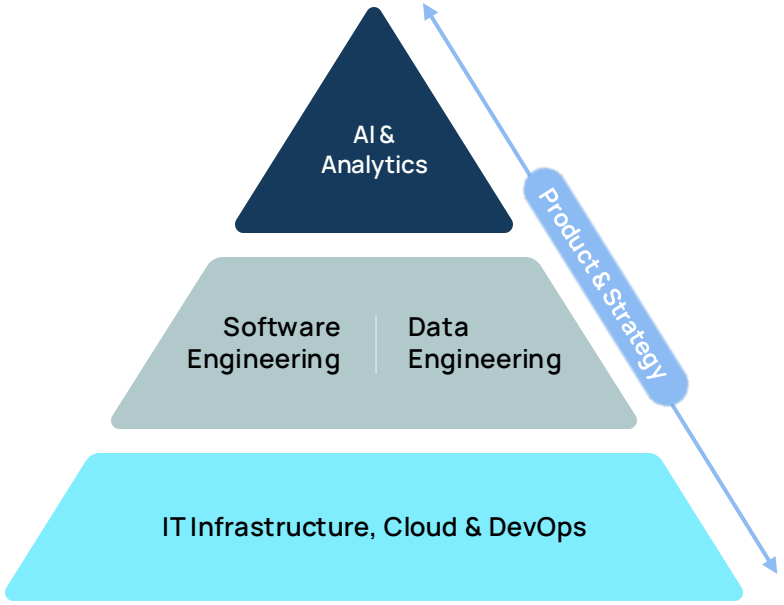
From Blue to Green without seing Red

Mastering deployments on AWS

04/12/2025

Where technology meets business impact

Our standout expertise:



Paris
London
Offices

Supporting our clients across Europe

100 %
Every Lenstra consultant holds at least one valid technical certification.

92 %
Overall retention rate (annualized)

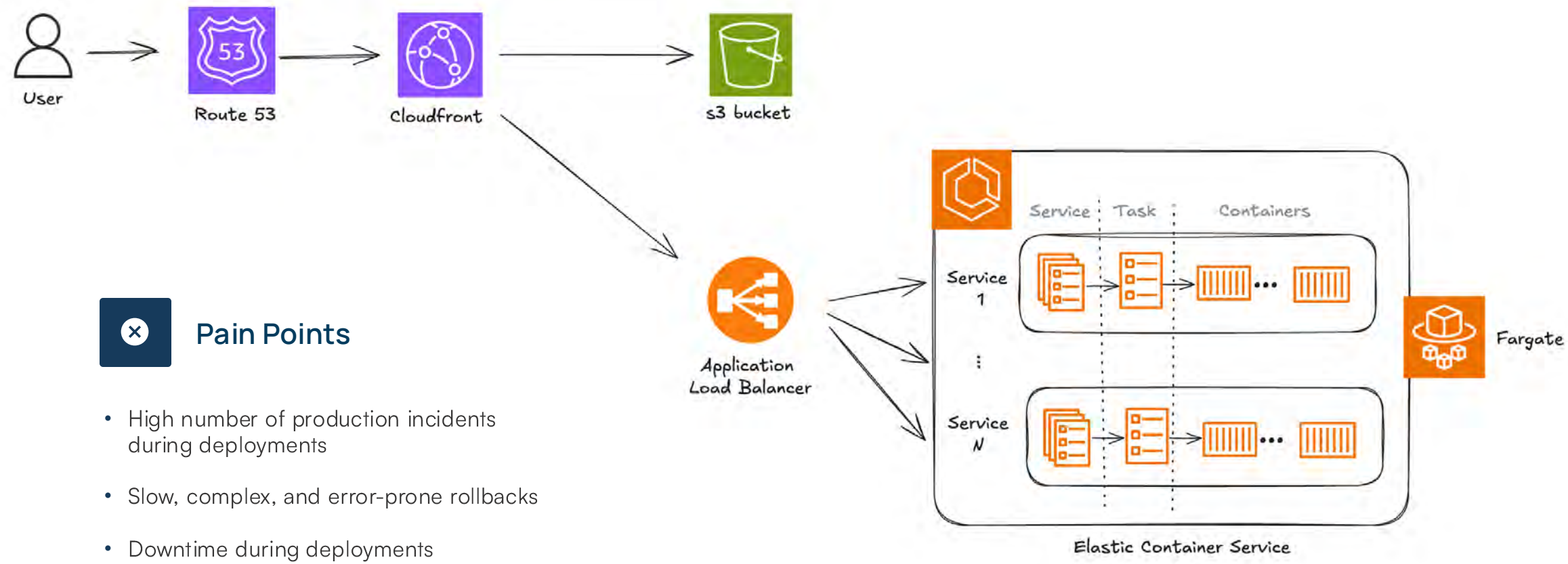
80+
Consultants

27 %
Growth rate over two years

01. Why

lenstra CONF42

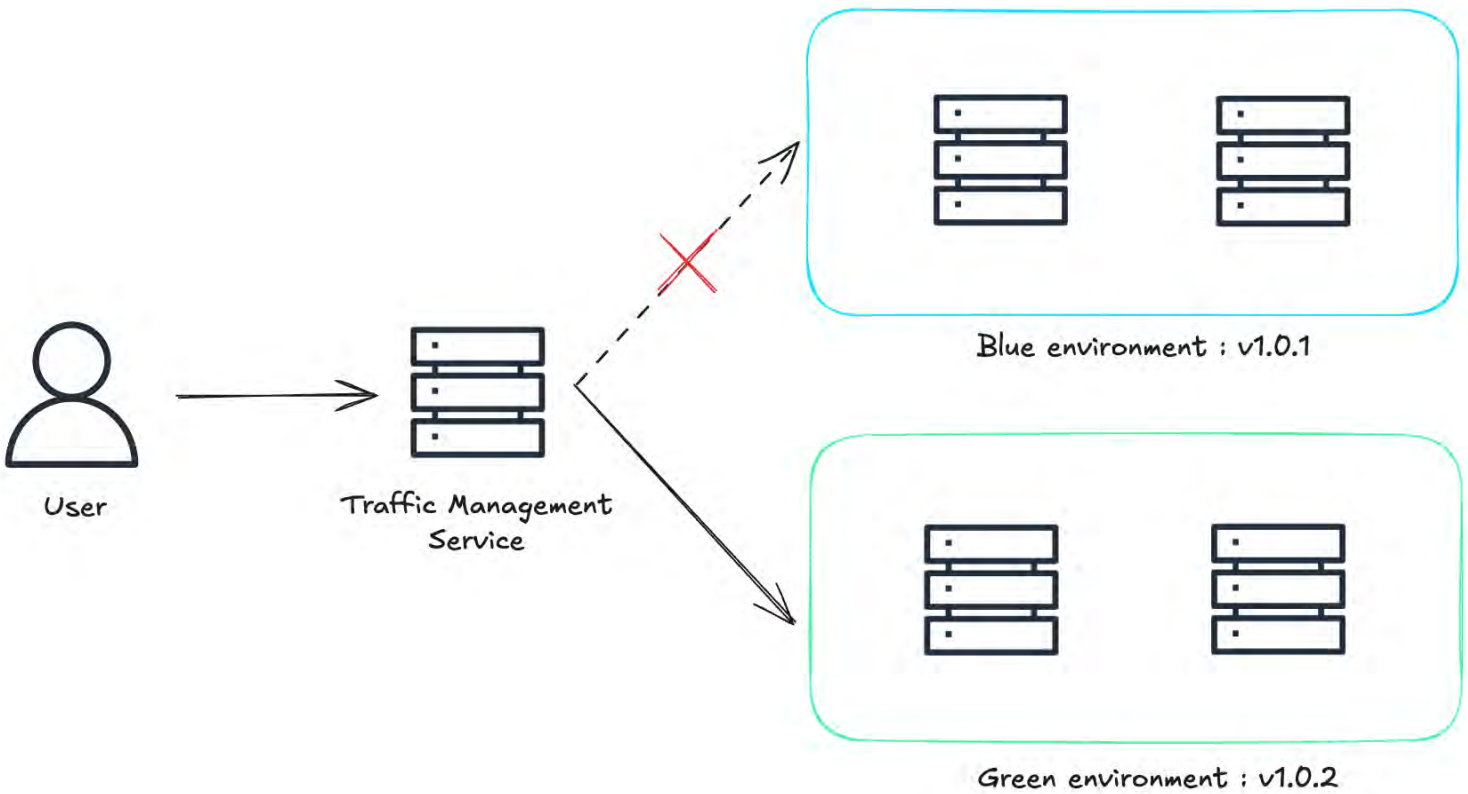
Starting point



Pain Points

- High number of production incidents during deployments
- Slow, complex, and error-prone rollbacks
- Downtime during deployments
- Operational stress for teams

Blue / Green deployment



Benefits

- Real-world testing before switching traffic
- Fast, low-risk rollback
- Safer, more predictable releases
- Zero downtime during deployments

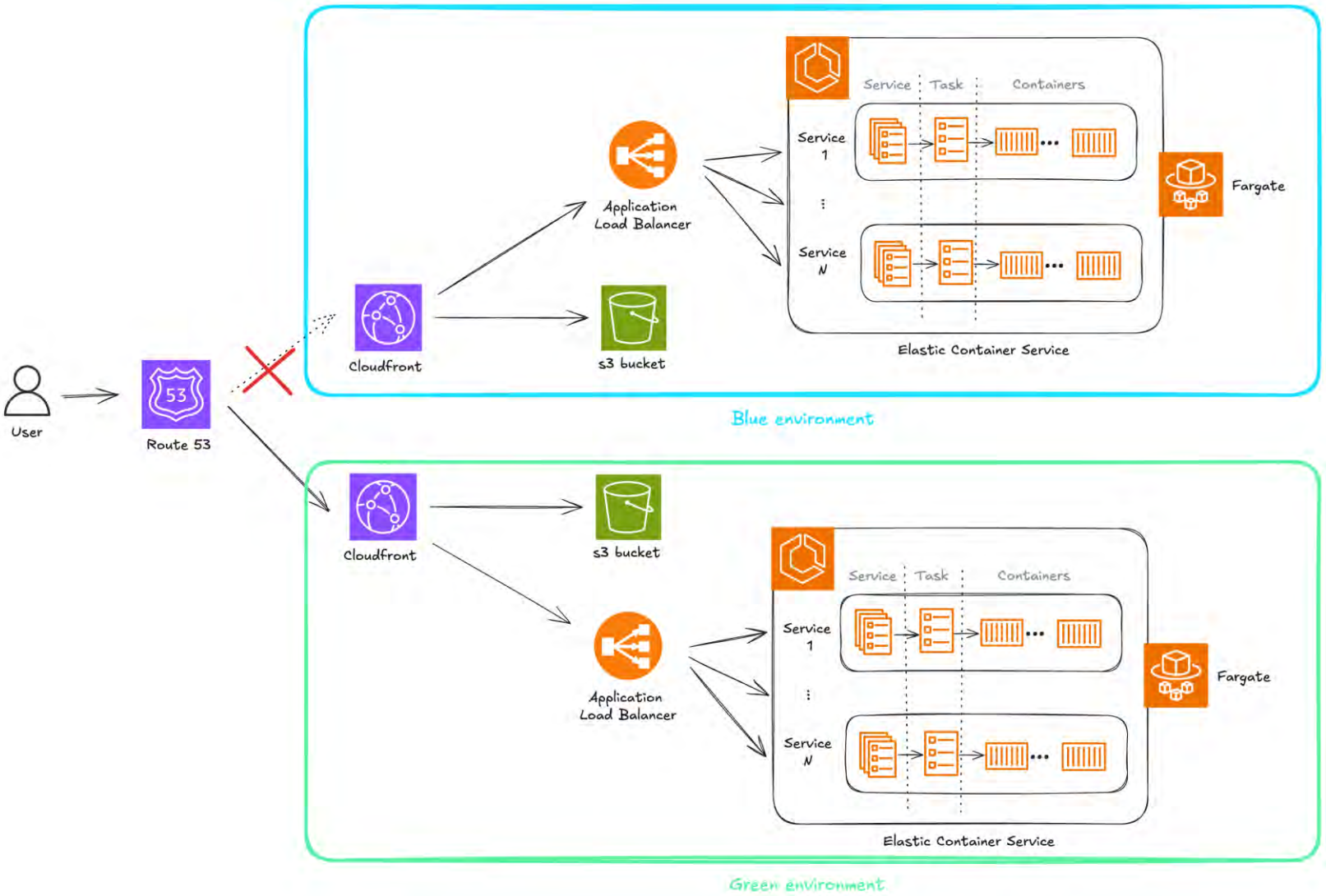
Pitfalls

- Higher costs
- State & database synchronization issues
- Configuration complexity

02. Same Goal, Many Roads

lenstra CONF42

Using Route53



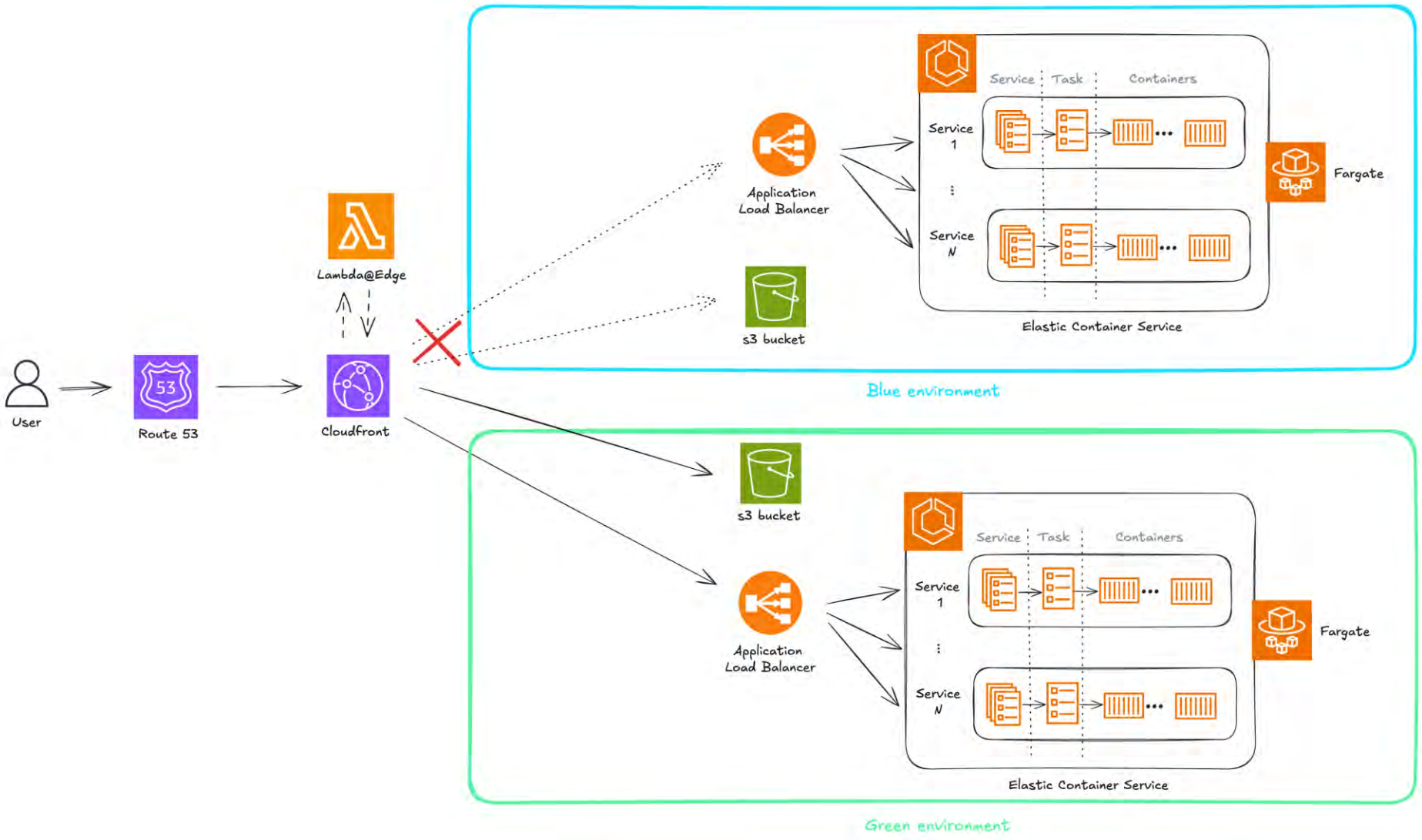
Benefits

- Simple
- Universal and architecture agnostic

Drawbacks

- DNS cache reliant
- Poor control over requests
- Complete infrastructure duplication

Using CloudFront Lambda@Edge



Benefits

- Dynamic and highly flexible routing
- Low latency

Drawbacks

- Complexity
- Deployment delay

Other options

API Gateway

- Flexible routing: can switch versions based on headers, cookies, or stage variables
- Works for APIs/microservices, not S3 static files

ALB

- Two target groups (Blue/Green) with weight-based traffic
- Switching weights reverts traffic immediately
- Works for APIs/microservices, not S3 static files
- No advanced routing logic

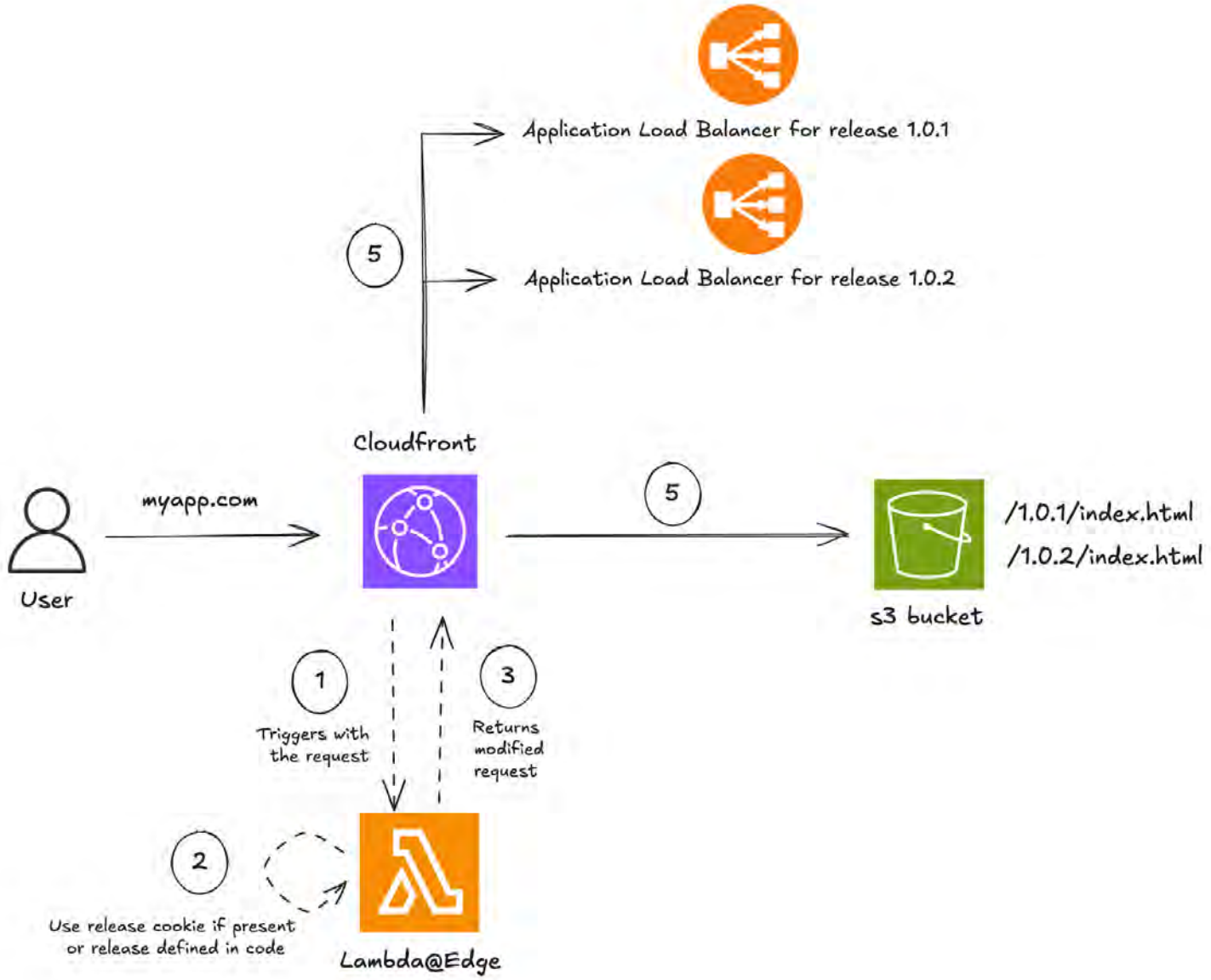
CodeDeploy

- Fully automated workflow including health checks, lifecycle hooks and automatic rollback on failure
- Setup overhead

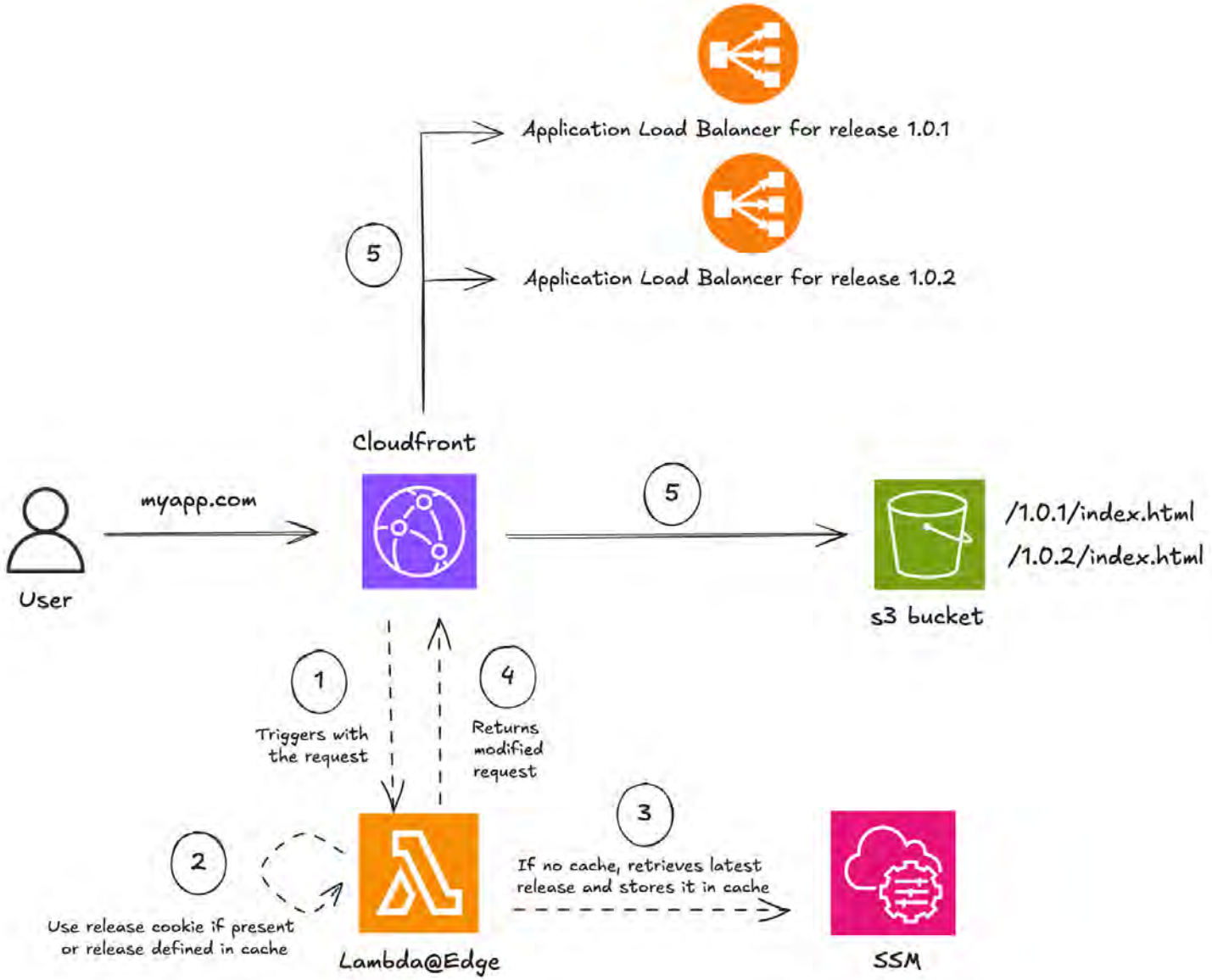
03. Pushing Logic to the Edge

lenstra CONF42

First iteration



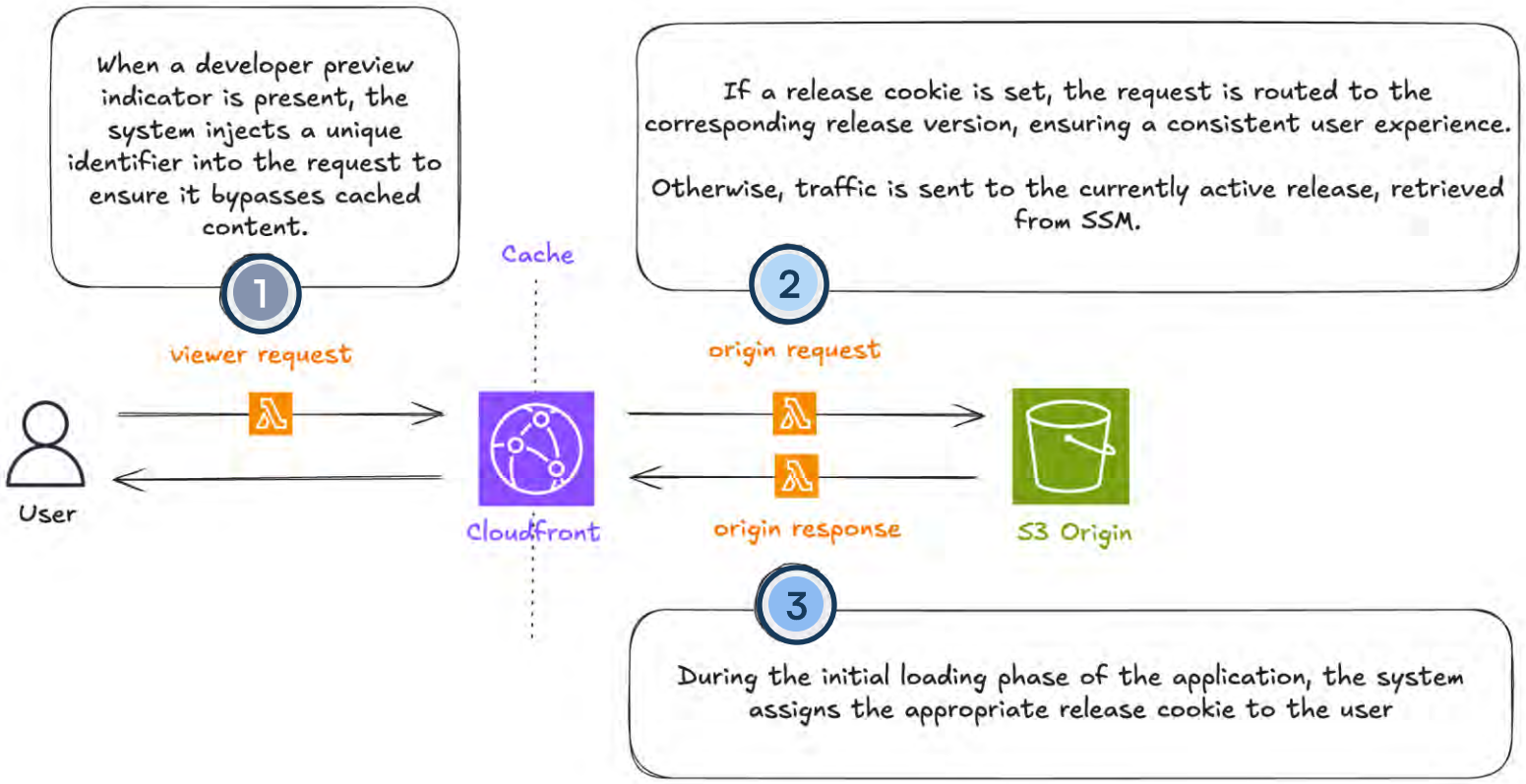
Second iteration



04. Adding Cache in the Balance

lenstra CONF42

Architecture



1. Deploy the new release in the green environment
2. Run tests
3. Update active release in SSM
4. Invalidate Cloudfront cache
5. Destroy the blue environment

Next steps

What we have achieved

- Low-latency, cookie-based routing for Blue/Green traffic switching
- < 1-minute Activation & rollback
- Cache-aware static frontend management (instant, predictable version flips)
- Zero production incidents during releases

Additional considerations

- **Multi-component coordination**
  Sync Blue/Green states across backend services, databases, and microservices
- **Progressive validation**
  Test new versions on a subset of users before switching 100%
- **CI/CD integration**
  Automate Blue/Green promotion, rollback, and validation through the pipeline

Thank you

CONF42 lenstra

Lenstra

34, avenue des Champs-Élysées
75008 Paris
France

www.lenstra.co

contact@lenstra.fr

Rayenn Hamrouni
DevOps Engineer @Lenstra
rayenn.hamrouni@lenstra.fr