

# Shopify's Architecture to Handle 80K RPS Celebrity Sales

*Simon Eskildsen – @Sirupsen*  
*Production Engineering Lead, Shopify*



Shopify is  
handling some of  
the **largest sales in**  
the world from  
Kylie Jenner,  
Kanye, Superbowl,  
and others



**“We learned to absorb these shocks and become stronger as a result. [..] The school of hard knocks has taught us well.”**

— Tobi Lütke, CEO in internal essay on why we optimize for flash sales

500K

Merchants powered

\$5.8B

Processed Q2, 2017

80K

Peak RPS

40+

Daily deploys

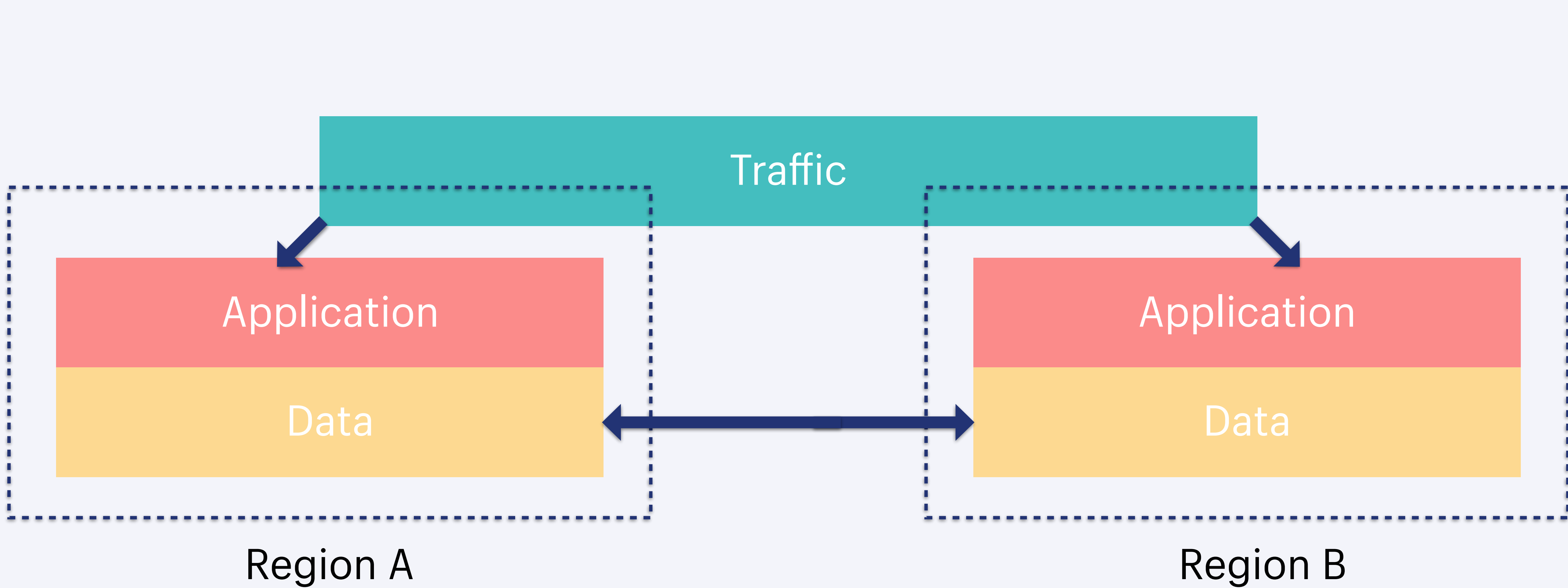
Rails

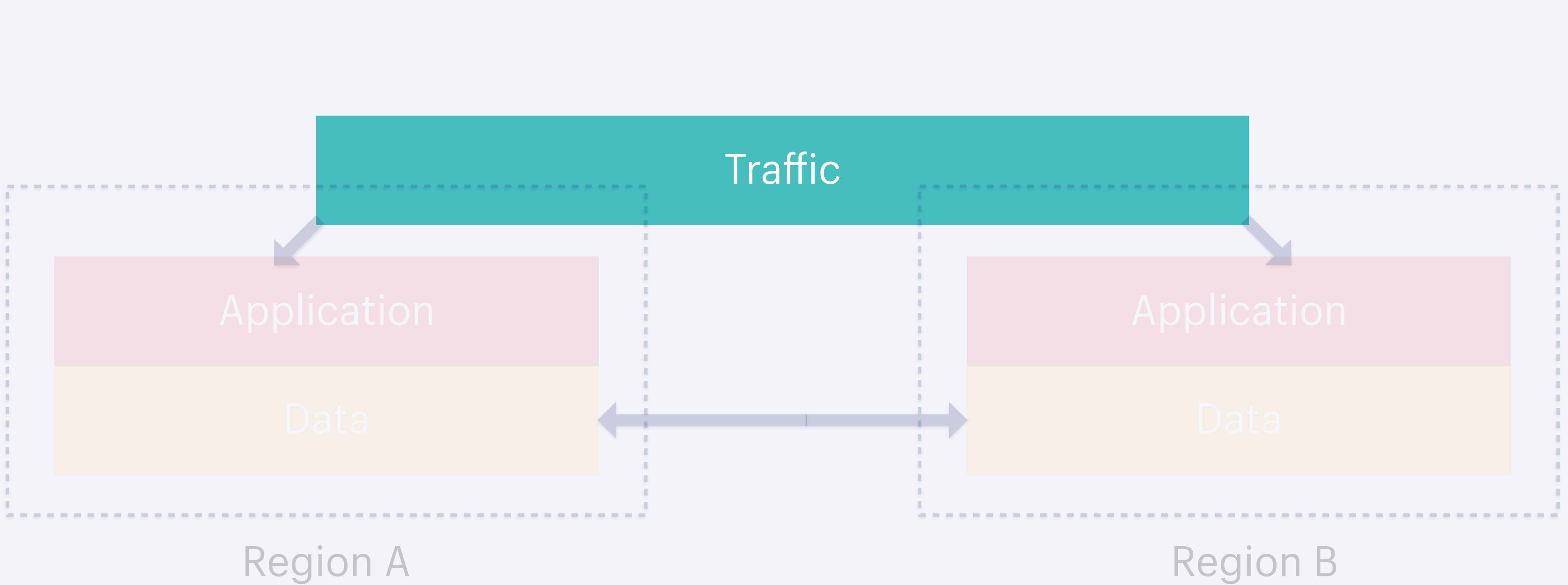
Ruby on Rails since 2006

20000+

Employees







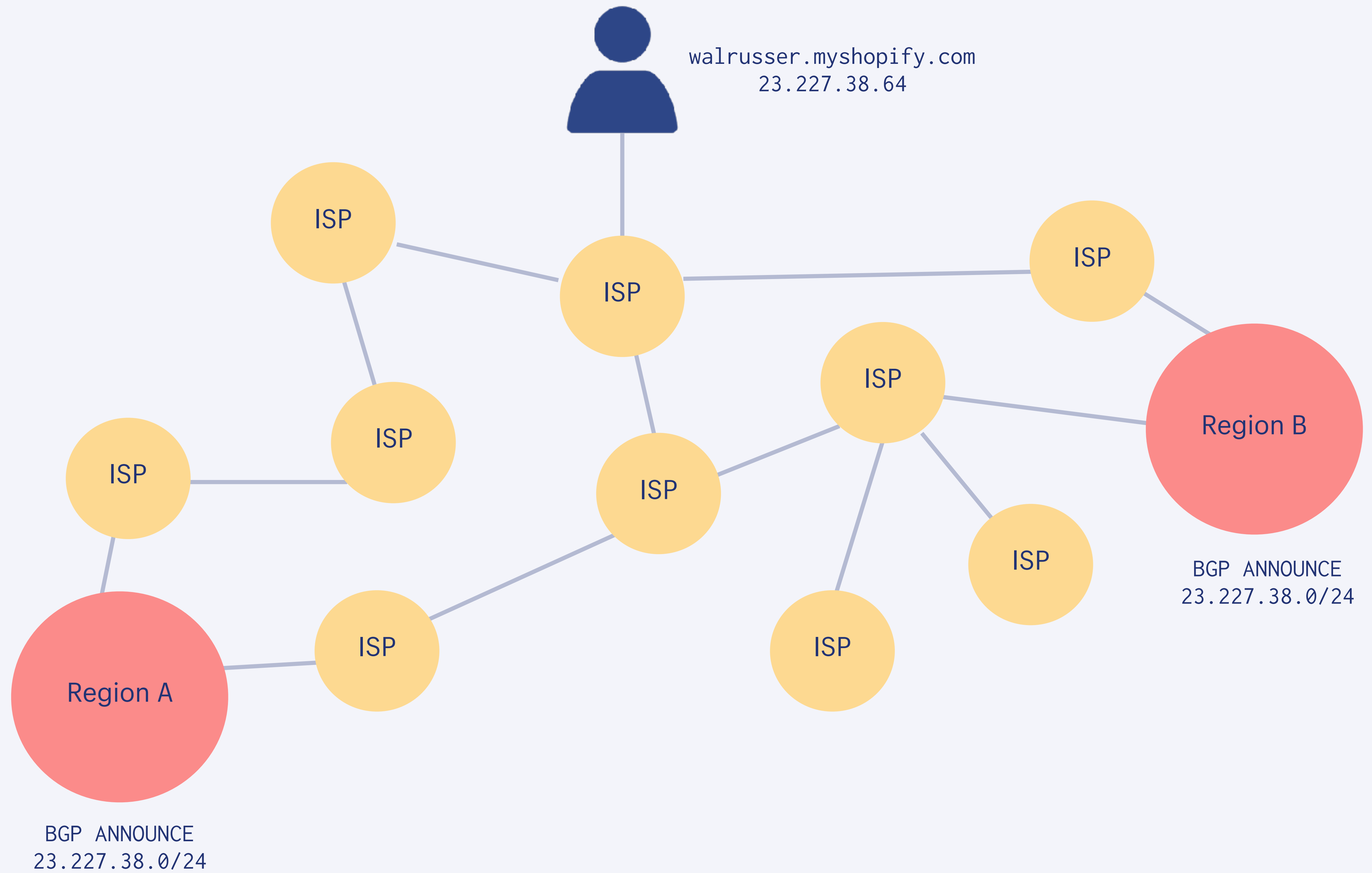




# Traffic

- Global Routing
- Openresty
- Bots
- Cache hits
- Checkout Throttling







OpenResty allows  
Lua scripting of  
your load  
balancers, it's been  
**one of the most  
impactful additions**  
to our stack in  
recent memory



Nginx with OpenResty

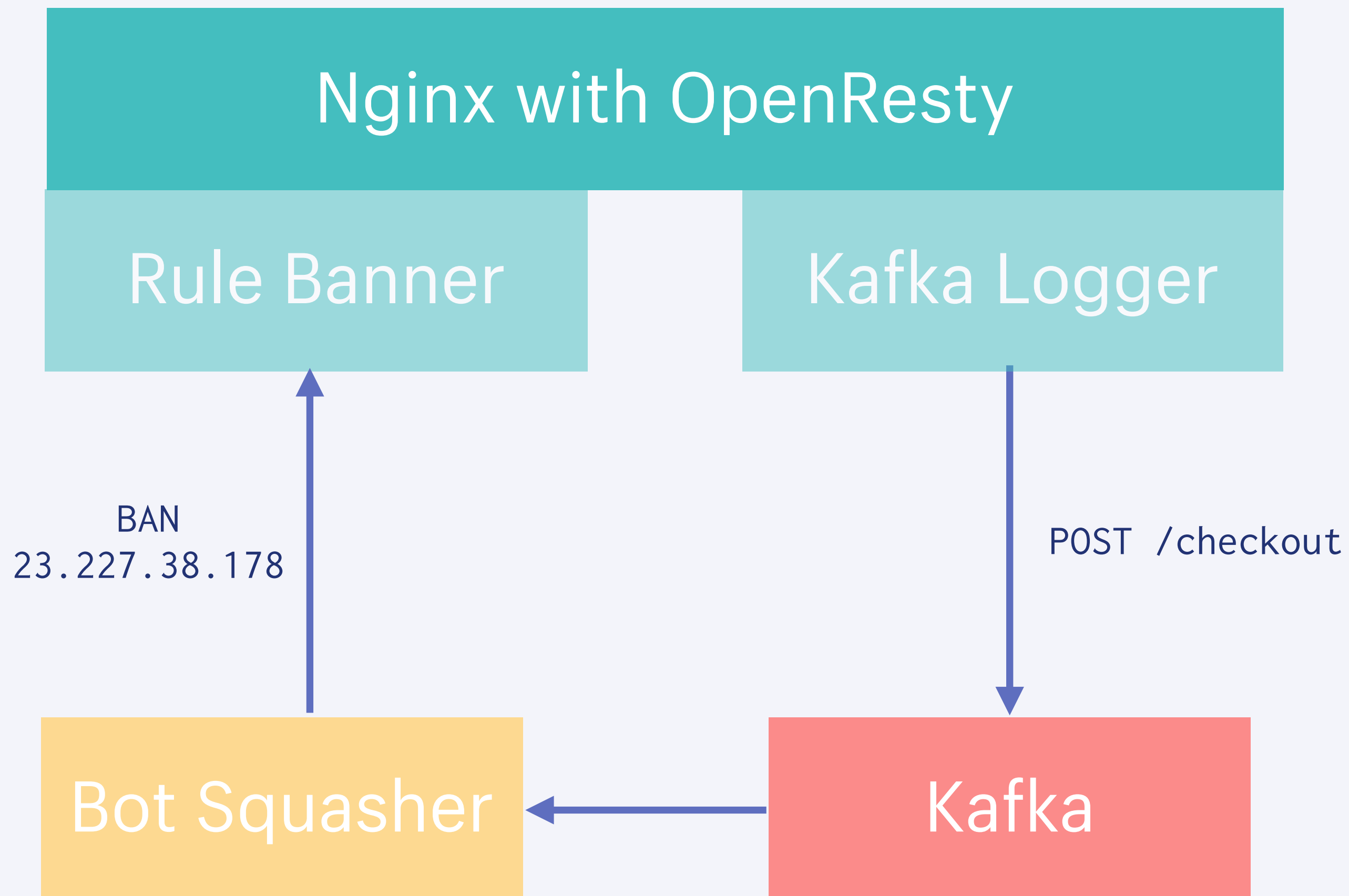
Rule Banner

Kafka Logging

Edgecache

Checkout Throttle

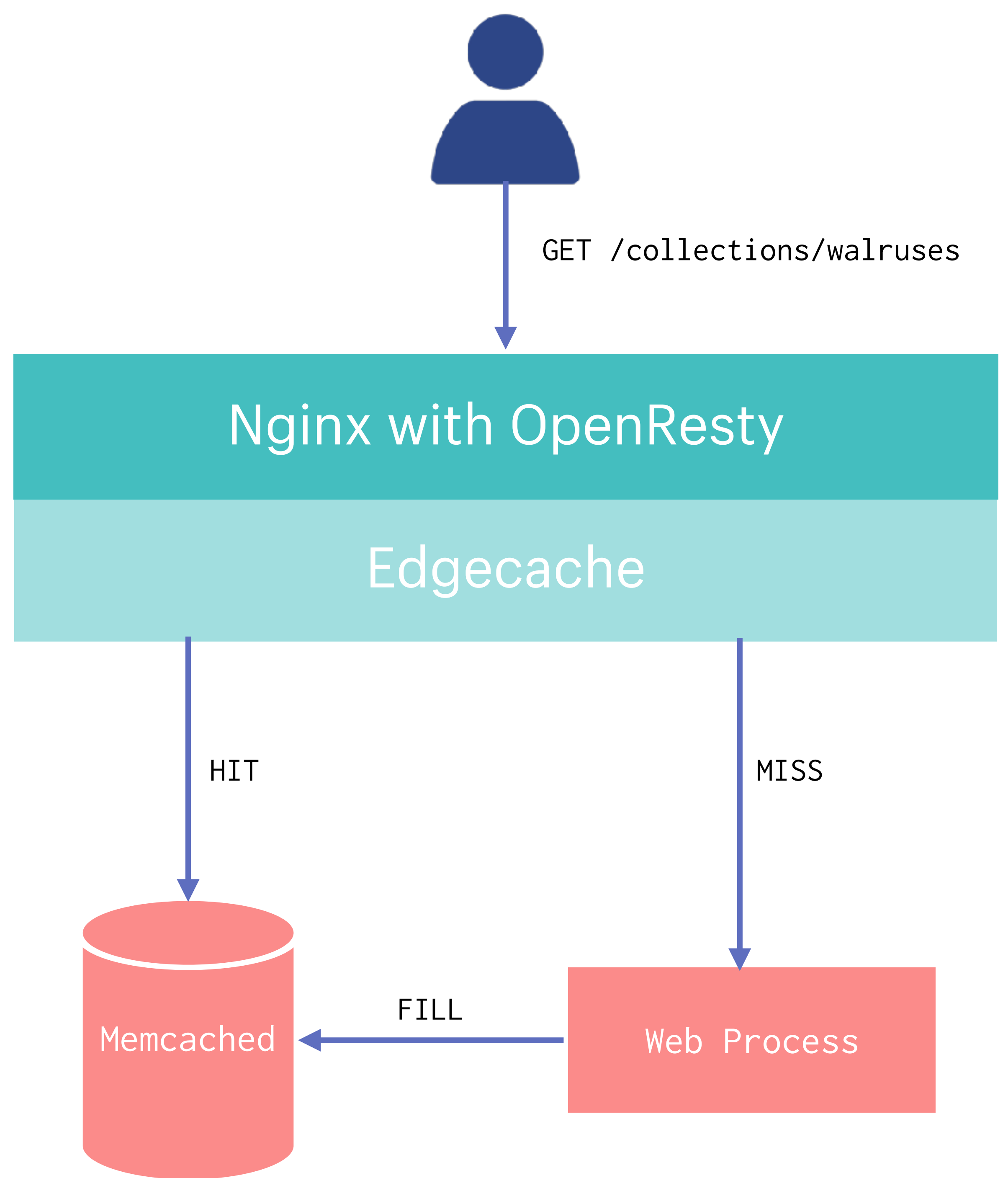
```
worker_processes 1;
error_log logs/error.log;
events {
    worker_connections 1024;
}
http {
    server {
        listen 8080;
        location / {
            default_type text/html;
            content_by_lua '
                ngx.say("<p>hello, world</p>")
            ';
        }
    }
}
```

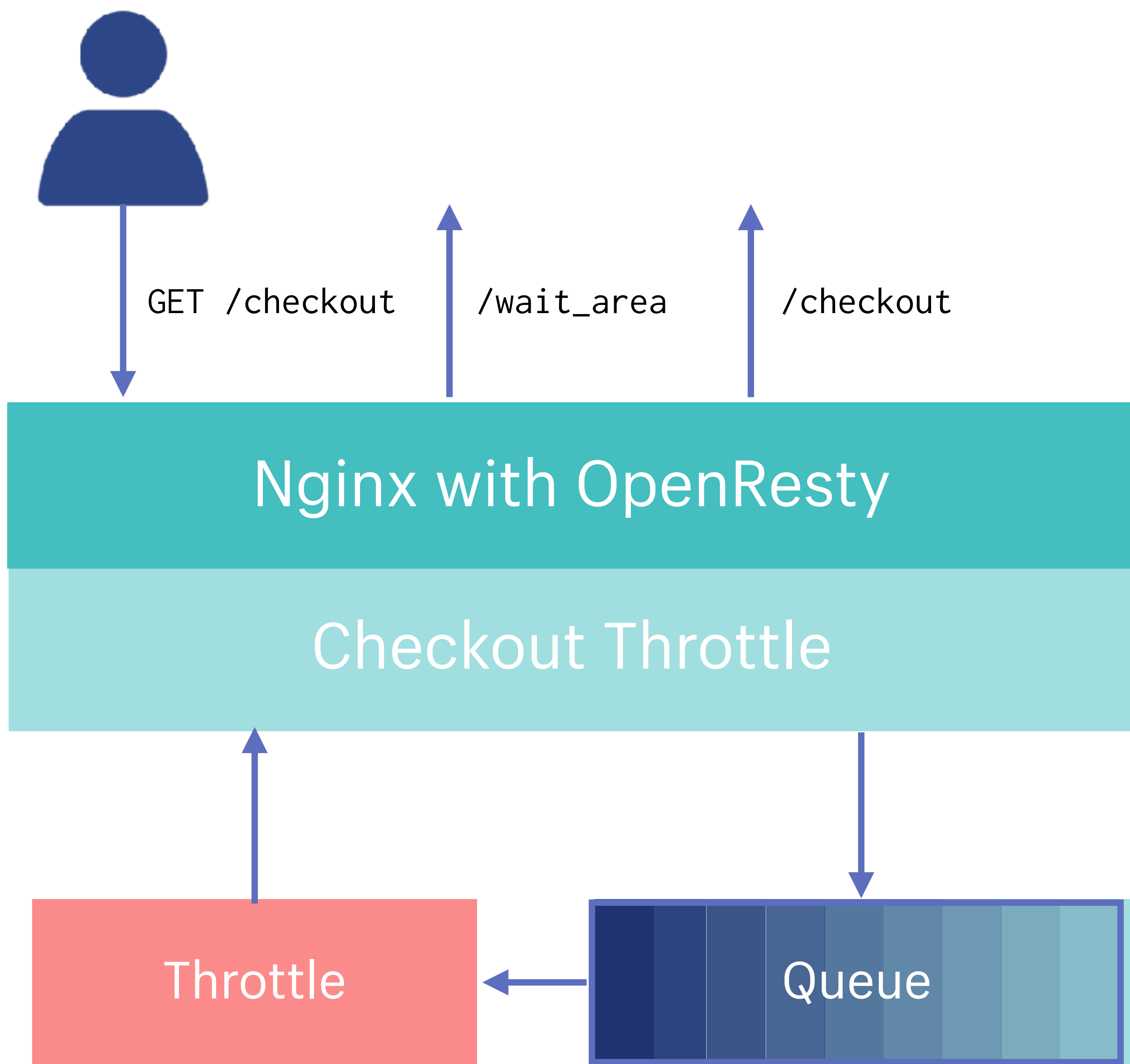


**Bot squasher  
analyzes the **Kafka**  
stream of incoming  
requests to ban  
bots with a rule  
banner module**

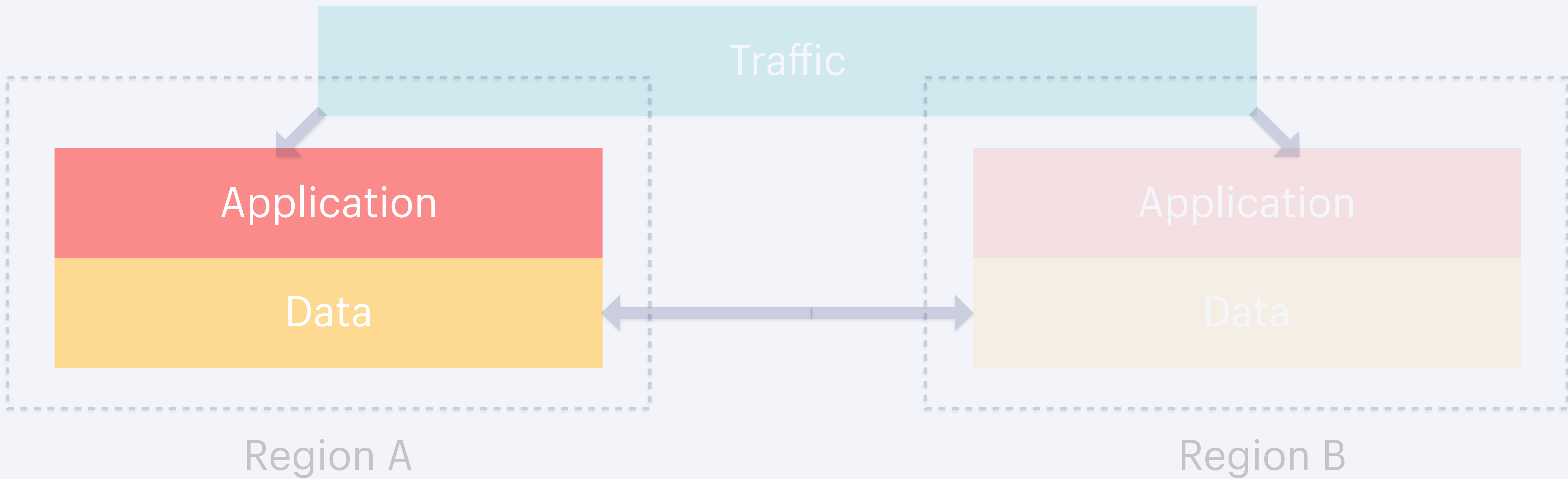


Edgecache can  
serve **full page  
cache hits** out of  
the load-balancers  
in microseconds





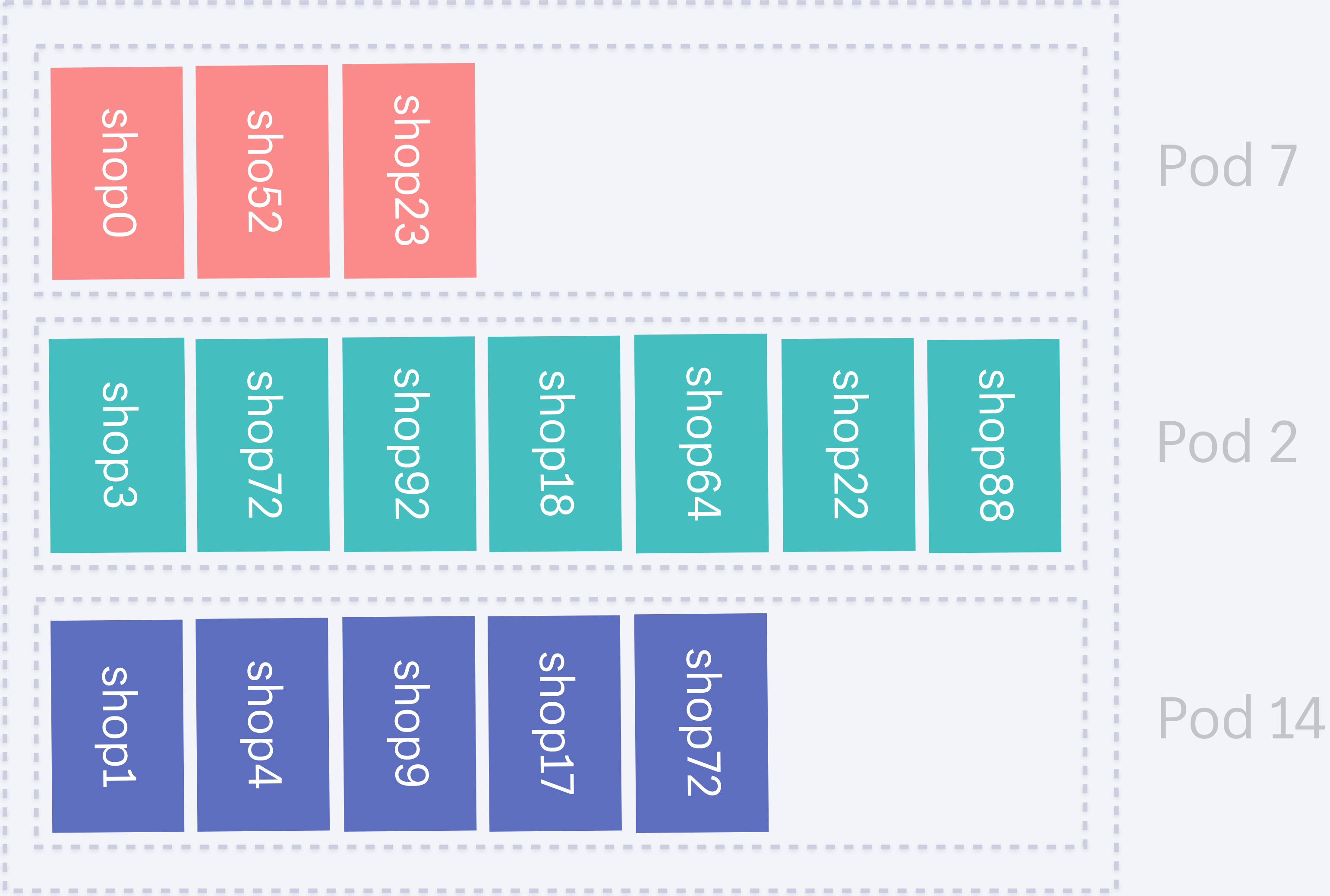
**Checkout Throttle throttles the number of customers in the processing heavy checkout path**



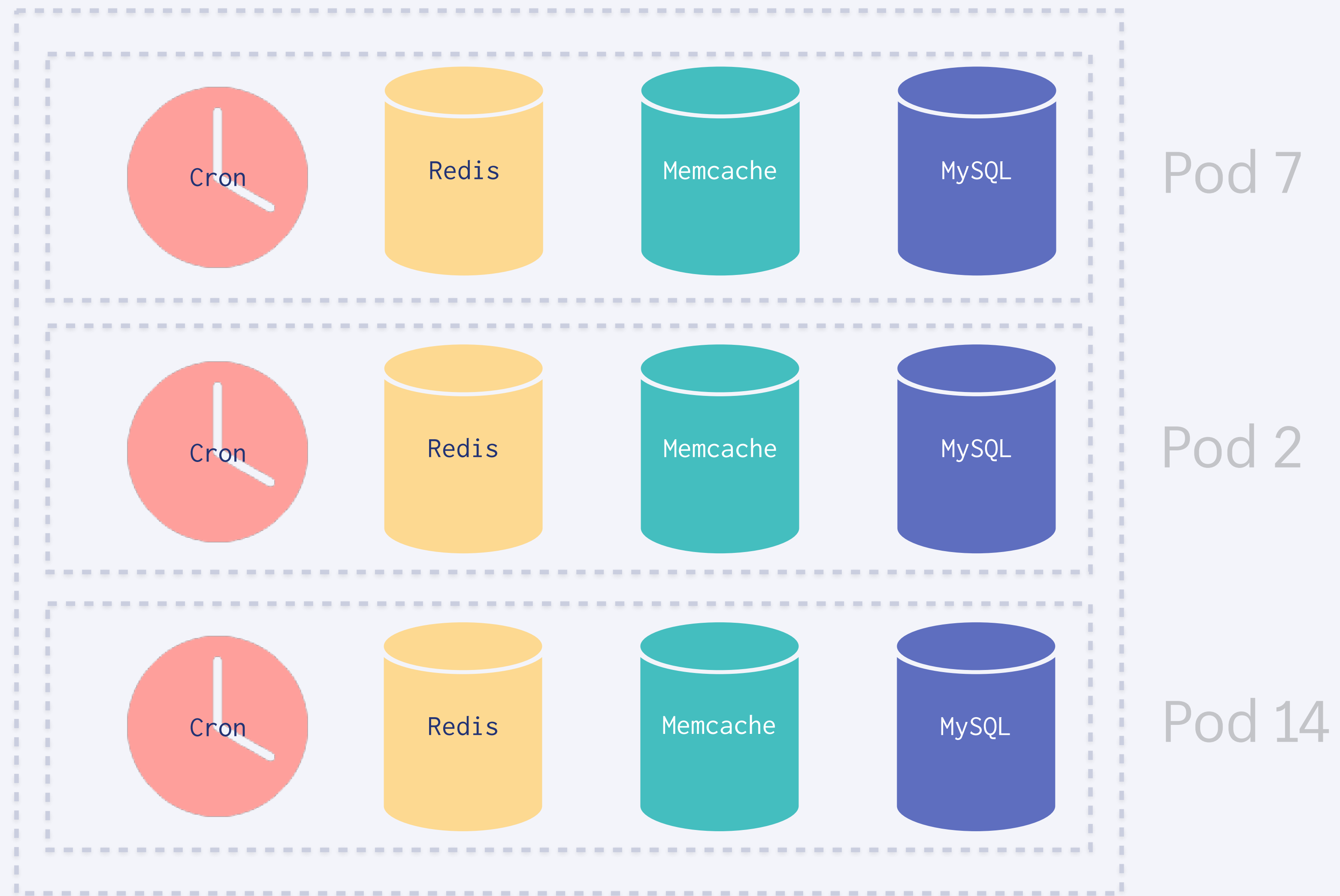


**Pod** is an isolated  
unit of one or more  
shops



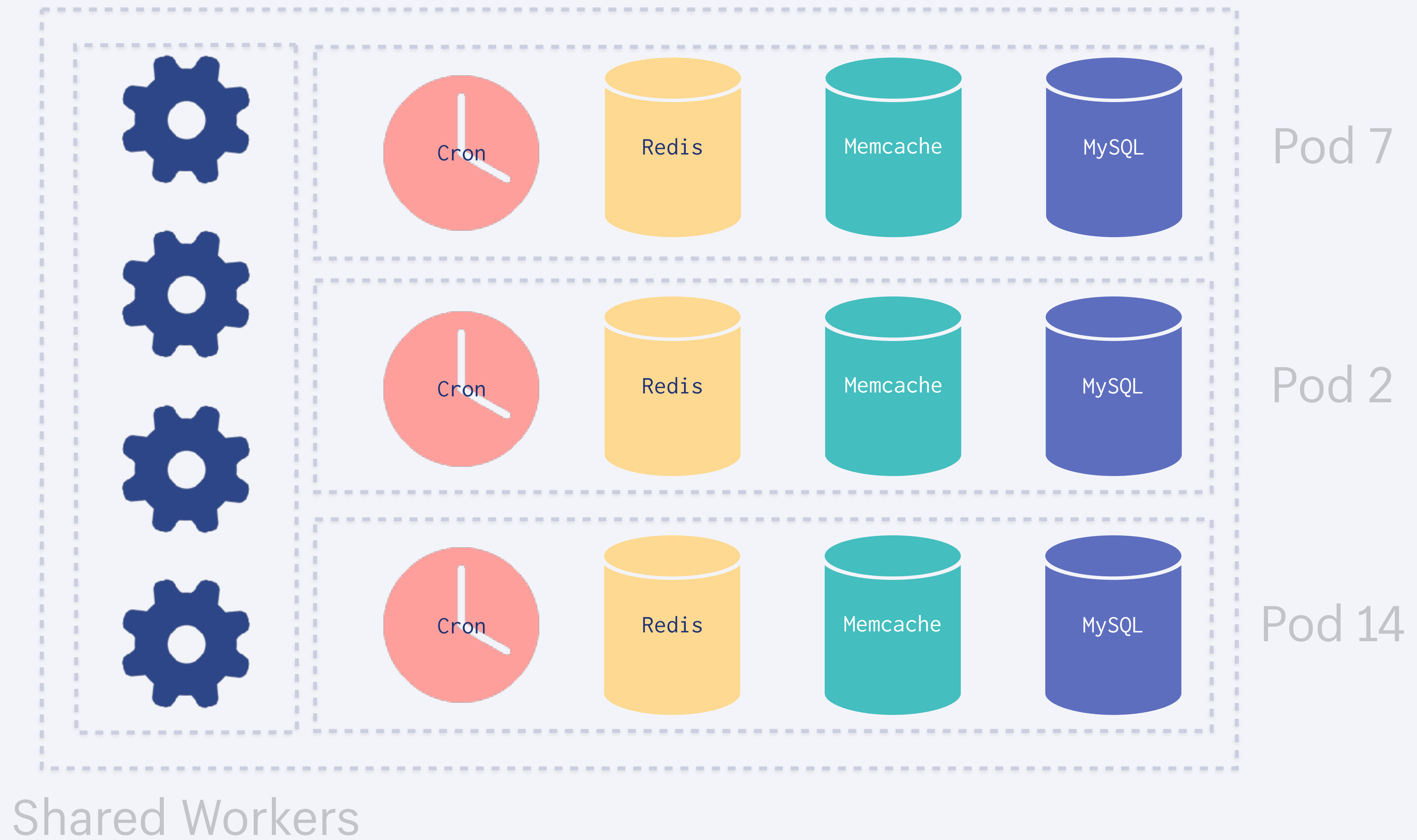


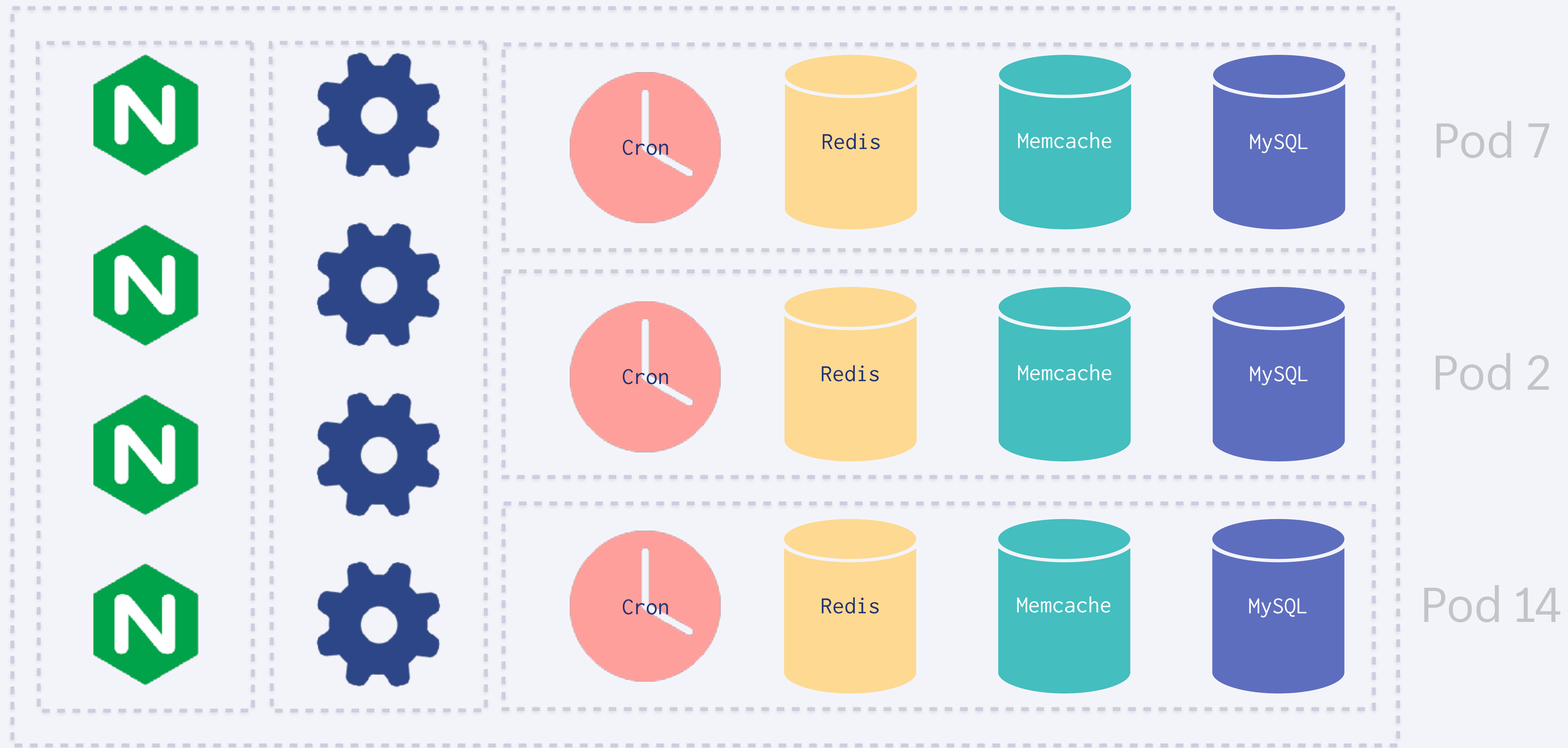
Data in Region A



Each Pod in Region A







Shared Load Balancing

# Genghis is our load-testing tool to test scale

16:16 **bedard** spy genghis enqueue checkout/shard-1.json duration=3m rate=6000

16:16 **spy** APP Enqueued flow checkout/shard-1.json as run id 577

16:16 **Genghis** APP Run 577 was queued and is about to start.

| Run enqueued by @bedard.

Run 577 starting with flow /checkout/flow.lua on master (checkout/shard-1.json) at 6000 executions/min for 3m0s

Find the flow [here](#).

Abort this run with `spy genghis remove 577` or `curl -X DELETE`

`https://genghis.shopifycloud.com/runs/577`

Abort all runs with `spy genghis abort` or `curl -X DELETE`

`https://genghis.shopifycloud.com/runs`

| Run enqueued by @bedard.





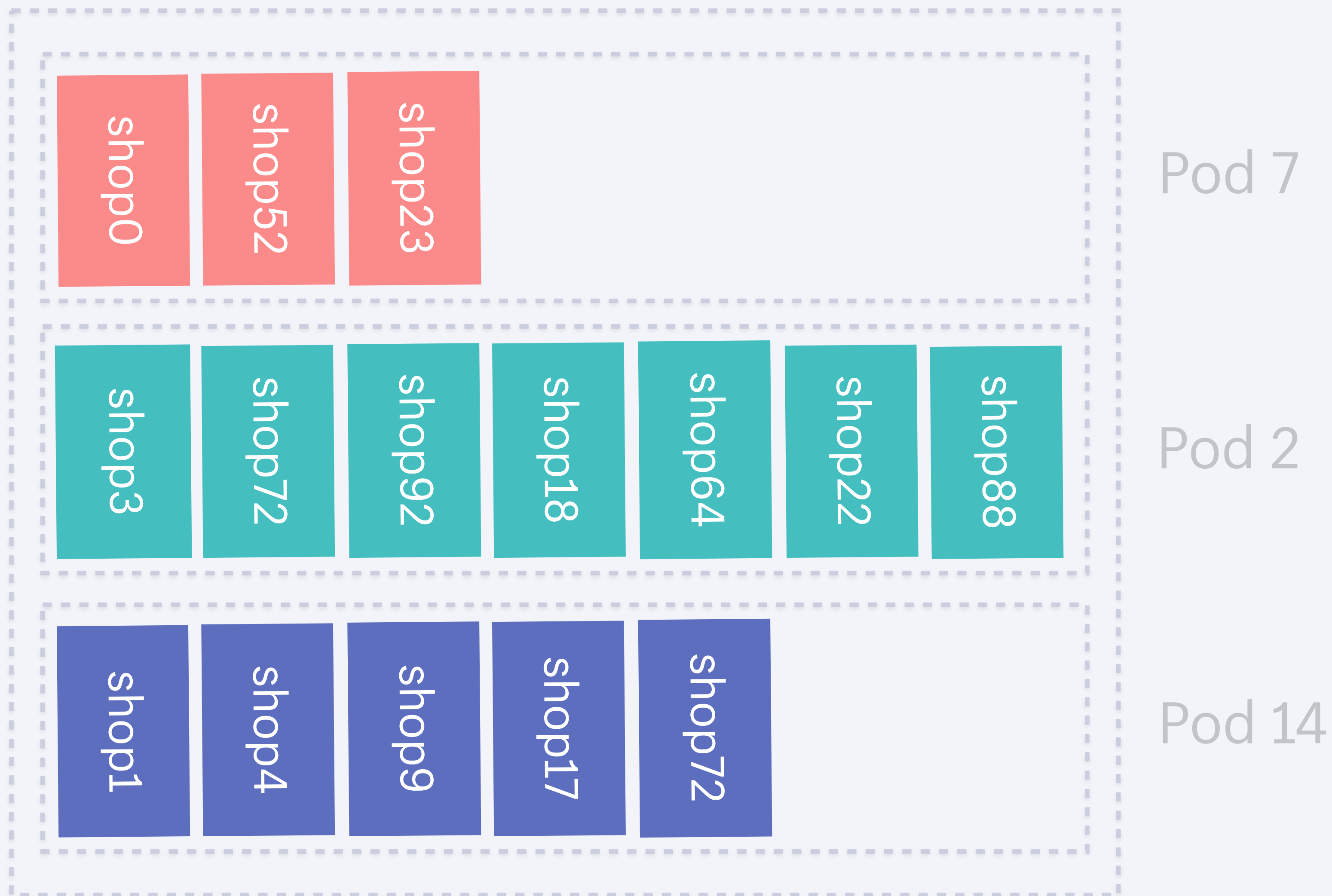
## Pod Balancer

balances shops  
between pods with  
minimal downtime  
to keep load and  
size even



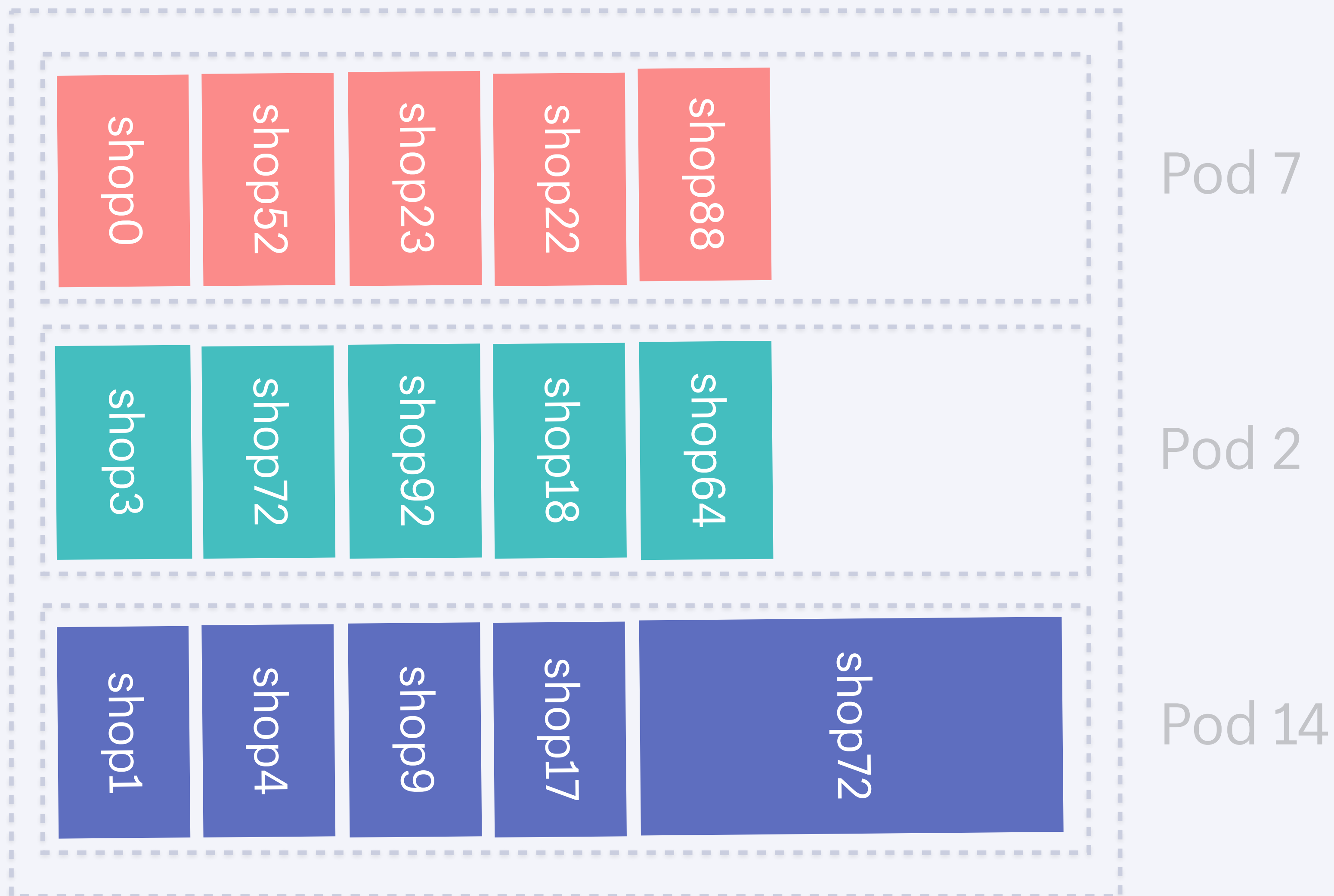


Pod Balancer



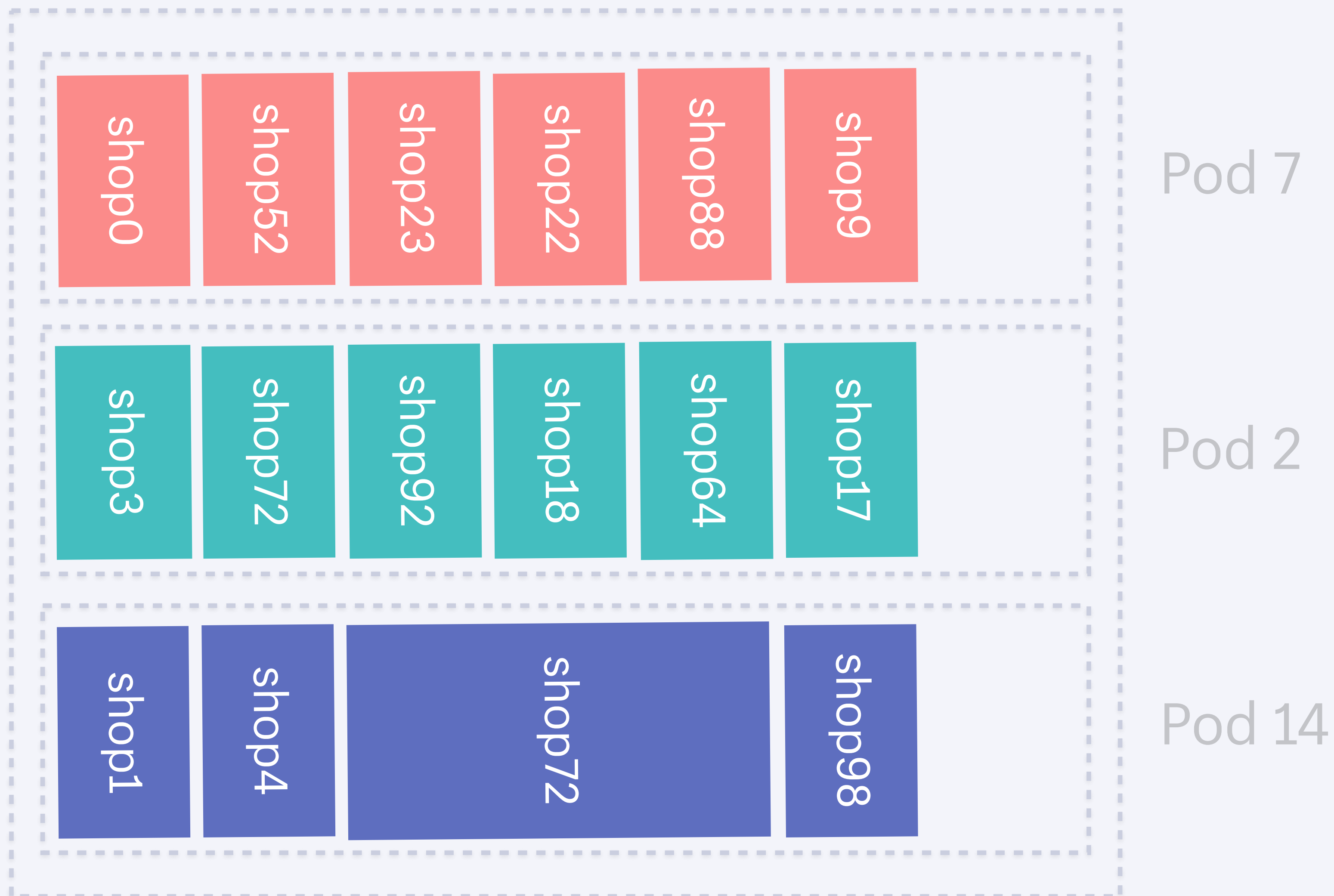


Pod Balancer





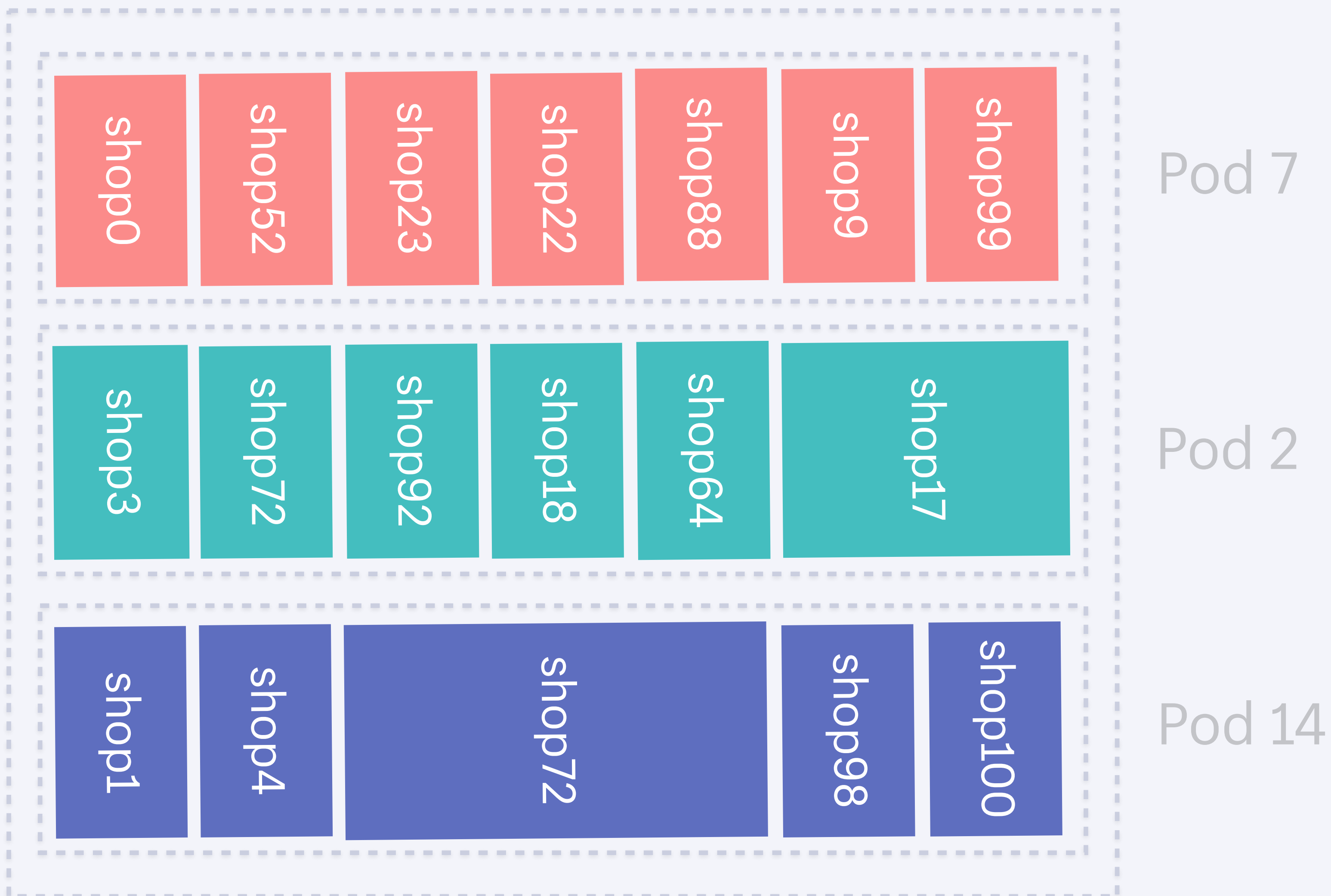
Pod Balancer





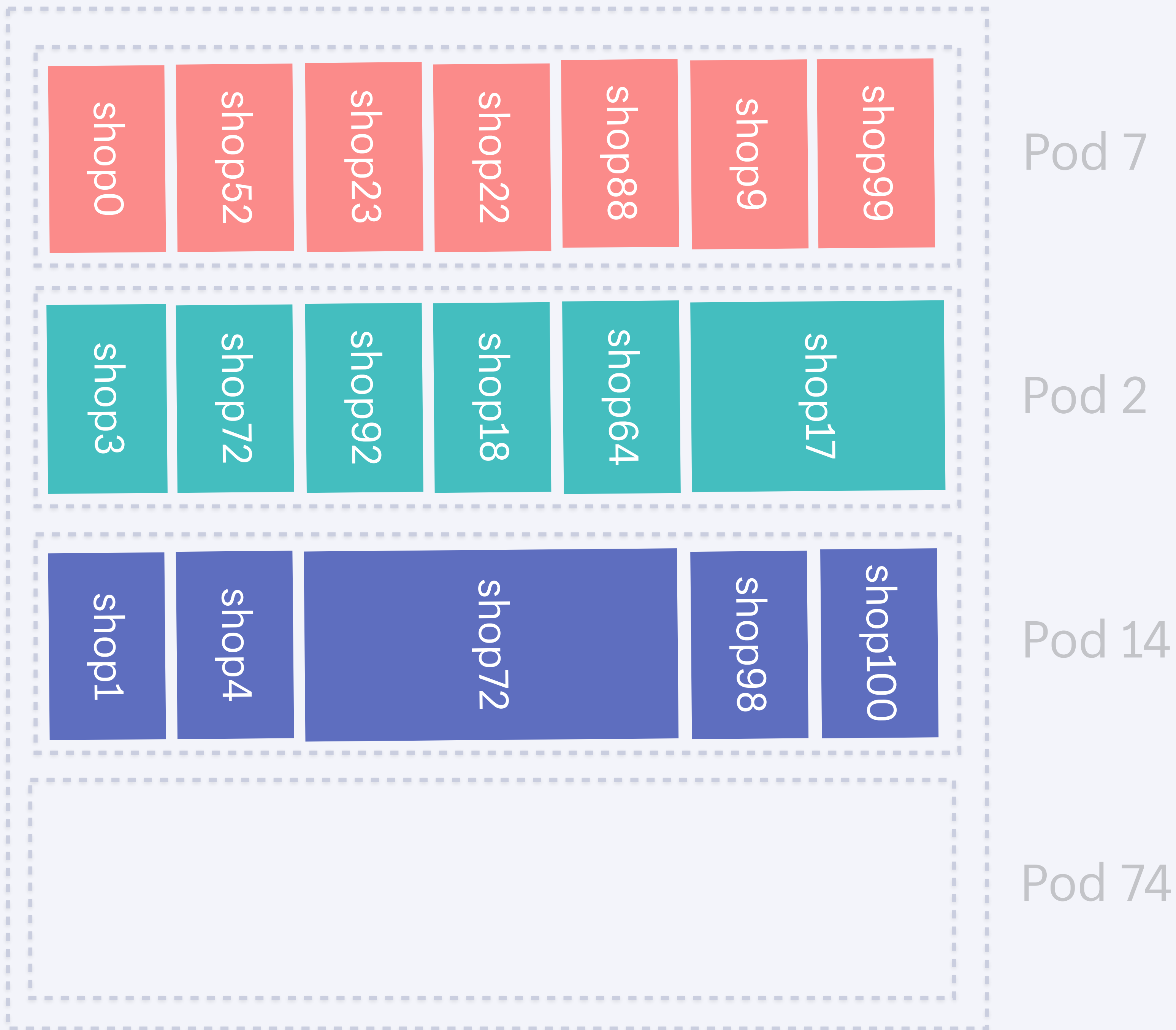


Pod Balancer



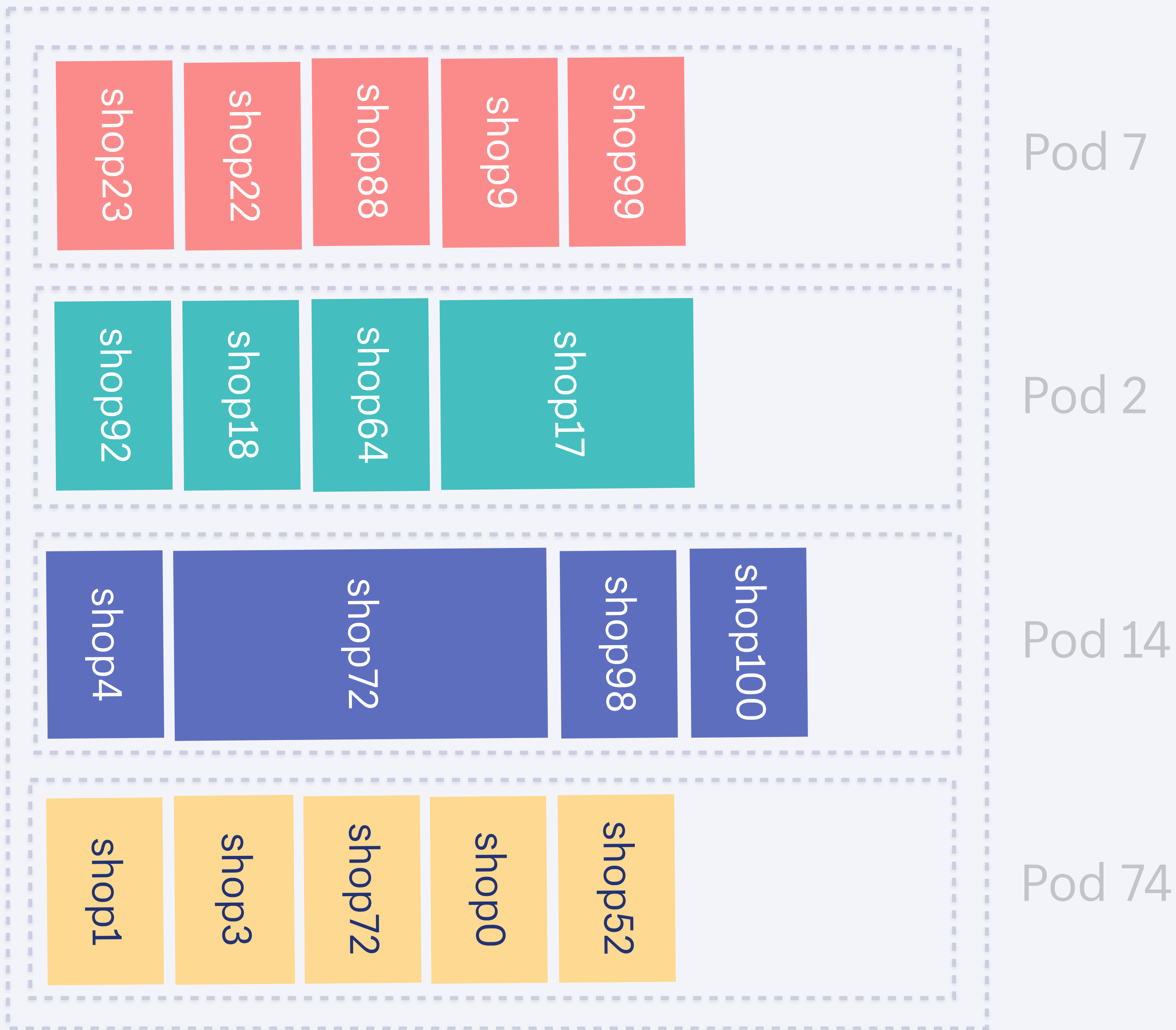


Pod Balancer

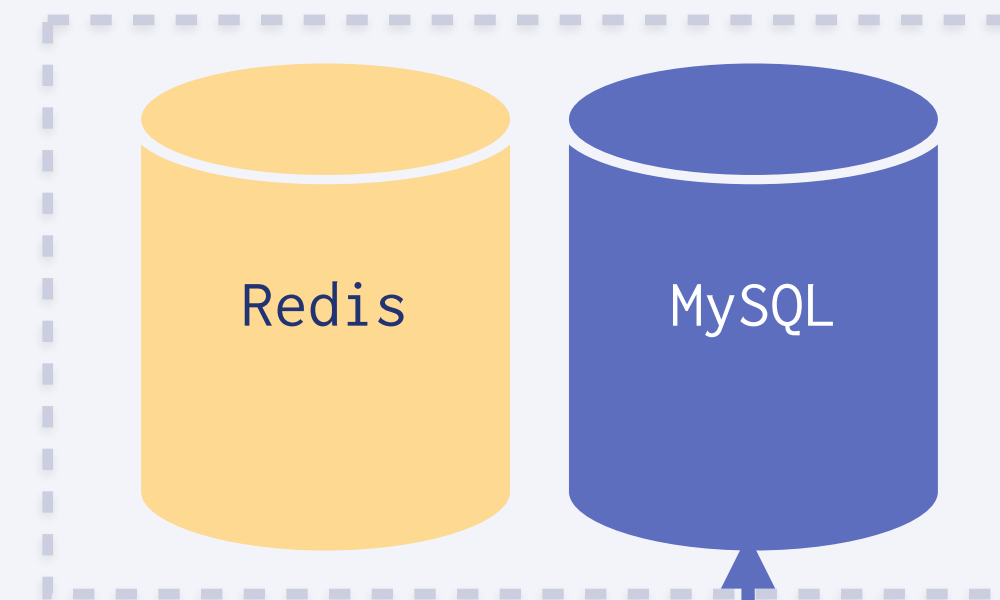
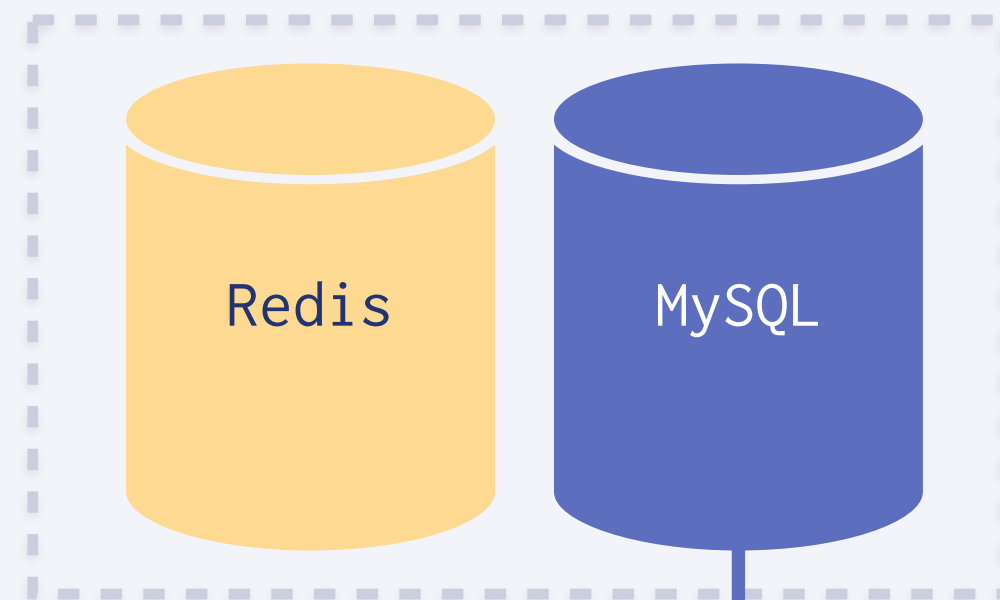




Pod Balancer



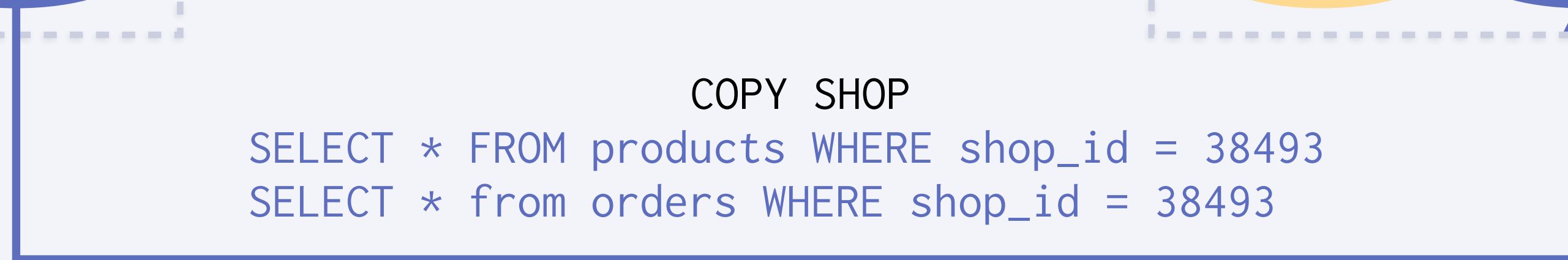
Source  
Pod 9



Target  
Pod 23

COPY SHOP

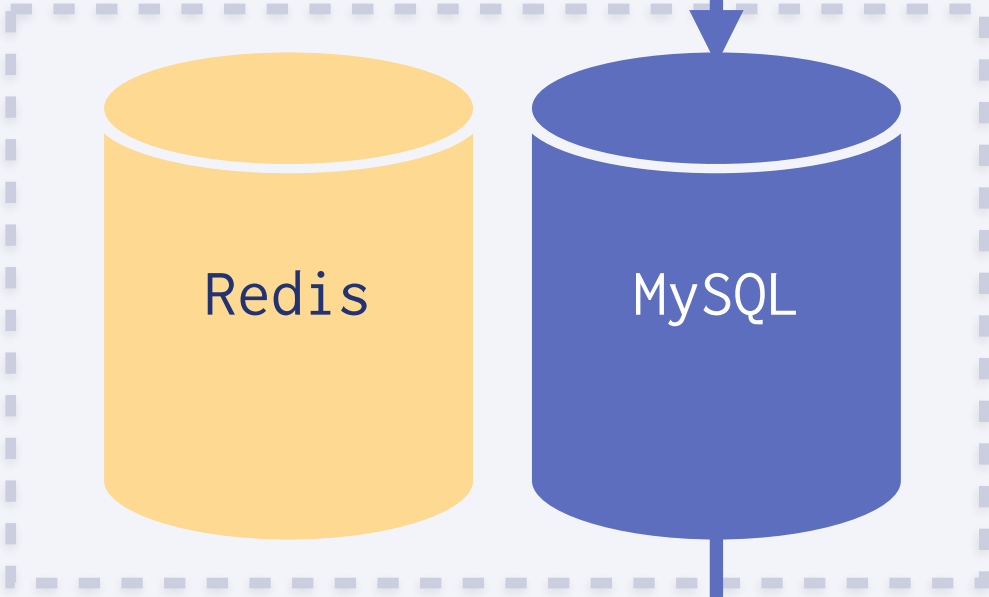
```
SELECT * FROM products WHERE shop_id = 38493  
SELECT * from orders WHERE shop_id = 38493
```



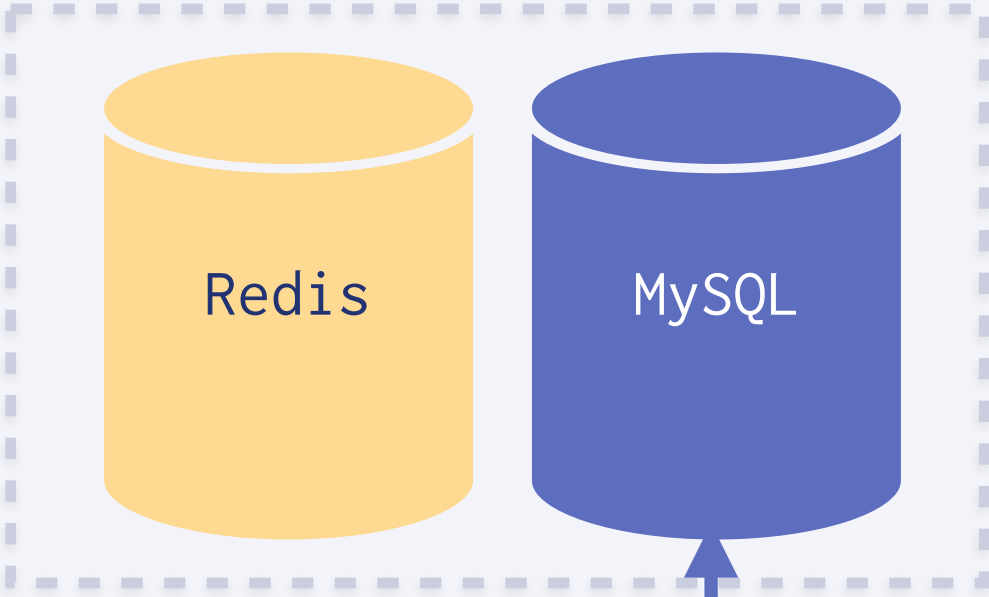


NEW CHECKOUT  
INSERT INTO CHECKOUTS ...

Source  
Pod 9



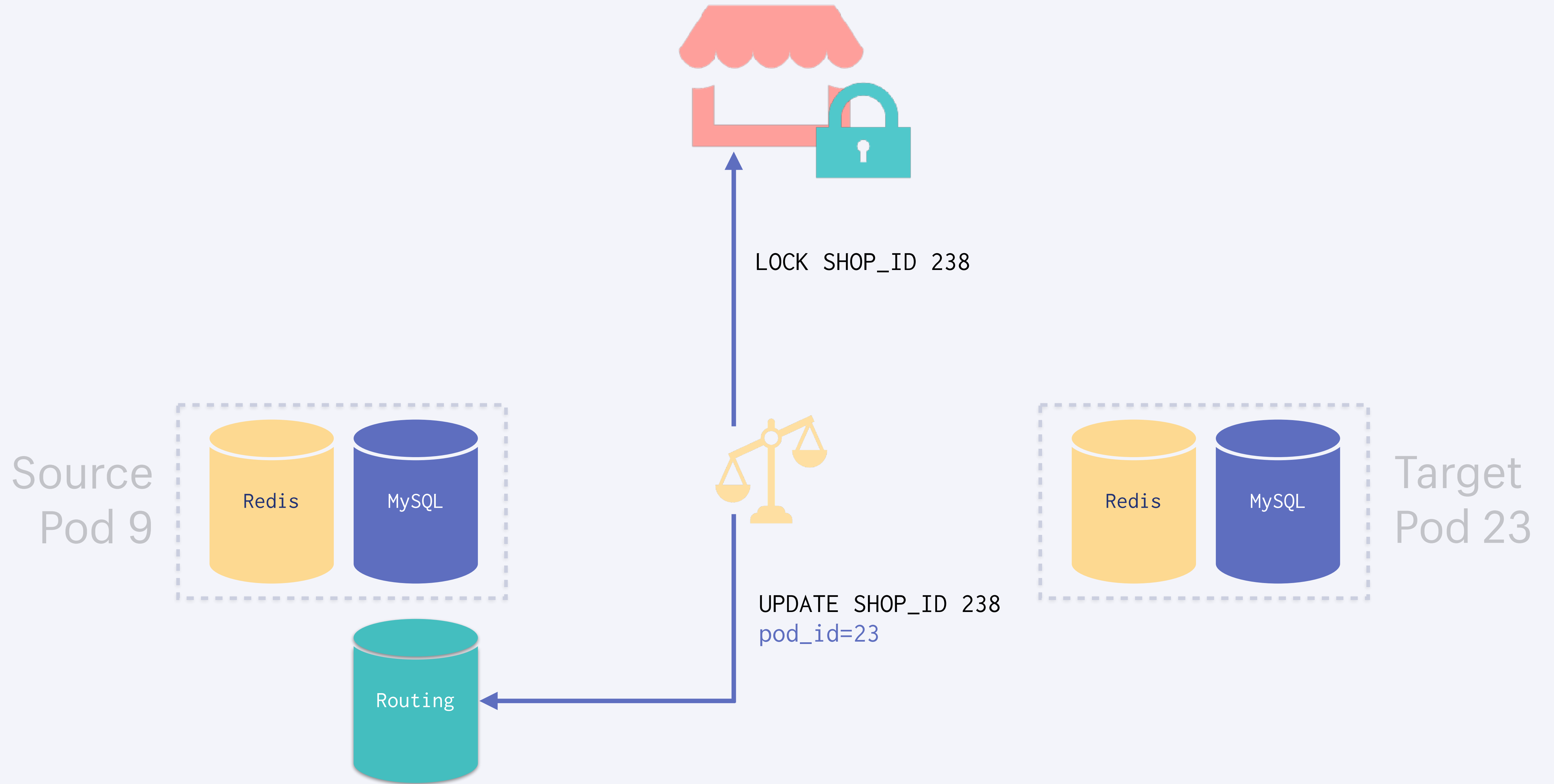
COPY SHOP  
SELECT \* FROM products WHERE shop\_id = 38493  
SELECT \* from orders WHERE shop\_id = 38493

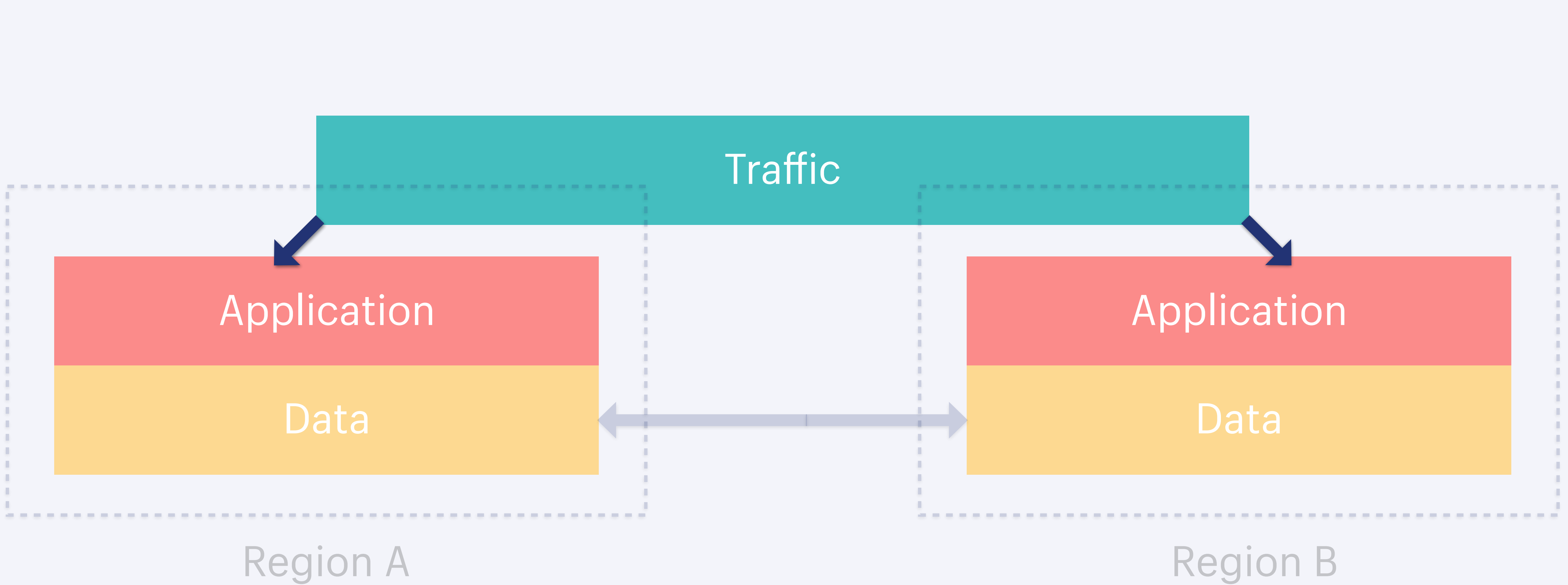


Target  
Pod 23







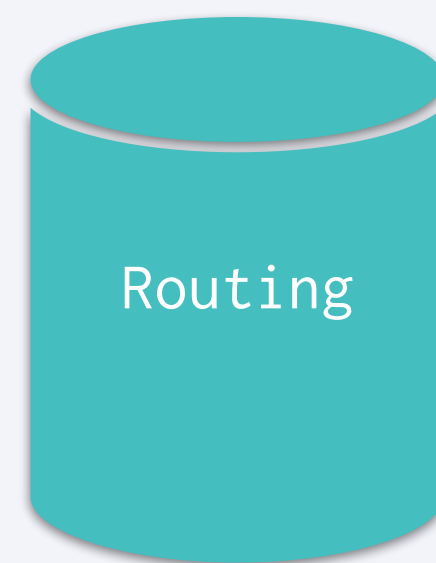


**Sorting Hat** routes  
requests for a shop  
to the region the  
pod is active in





GET /products  
Host: sneakershop.com



Routing

ROUTE  
sneakershop.com

shop238  
pod2:B

Traffic

Sorting Hat

Pod 7 Active

Pod 2 Inactive

Pod 14 Active

Region A

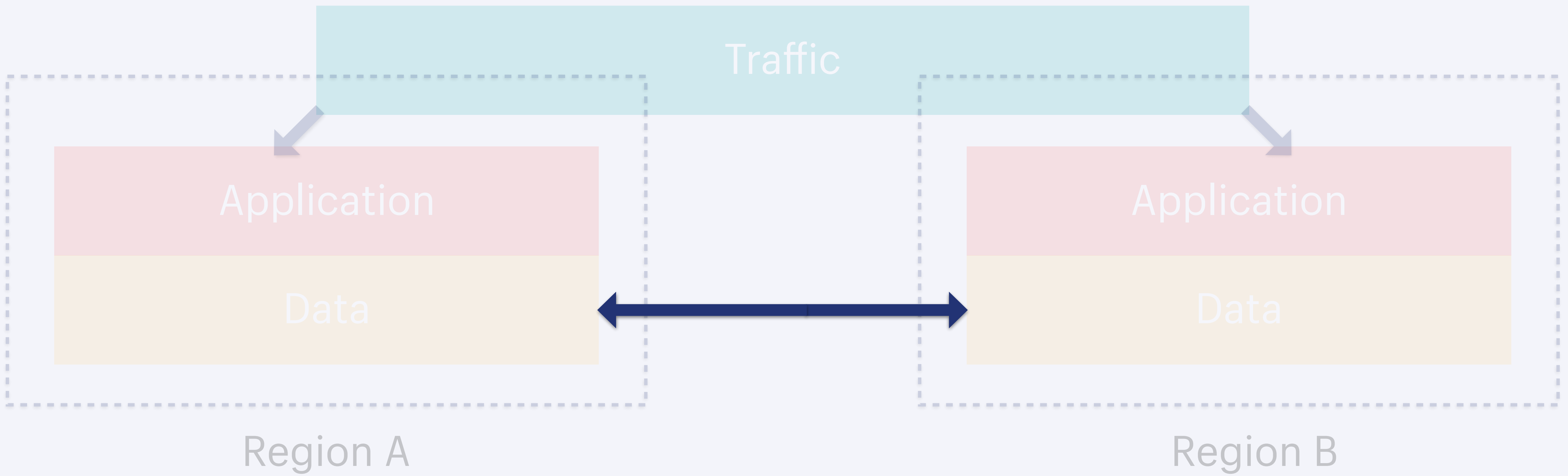
Pod 7 Inactive

Pod 2 Active

Pod 14 Inactive

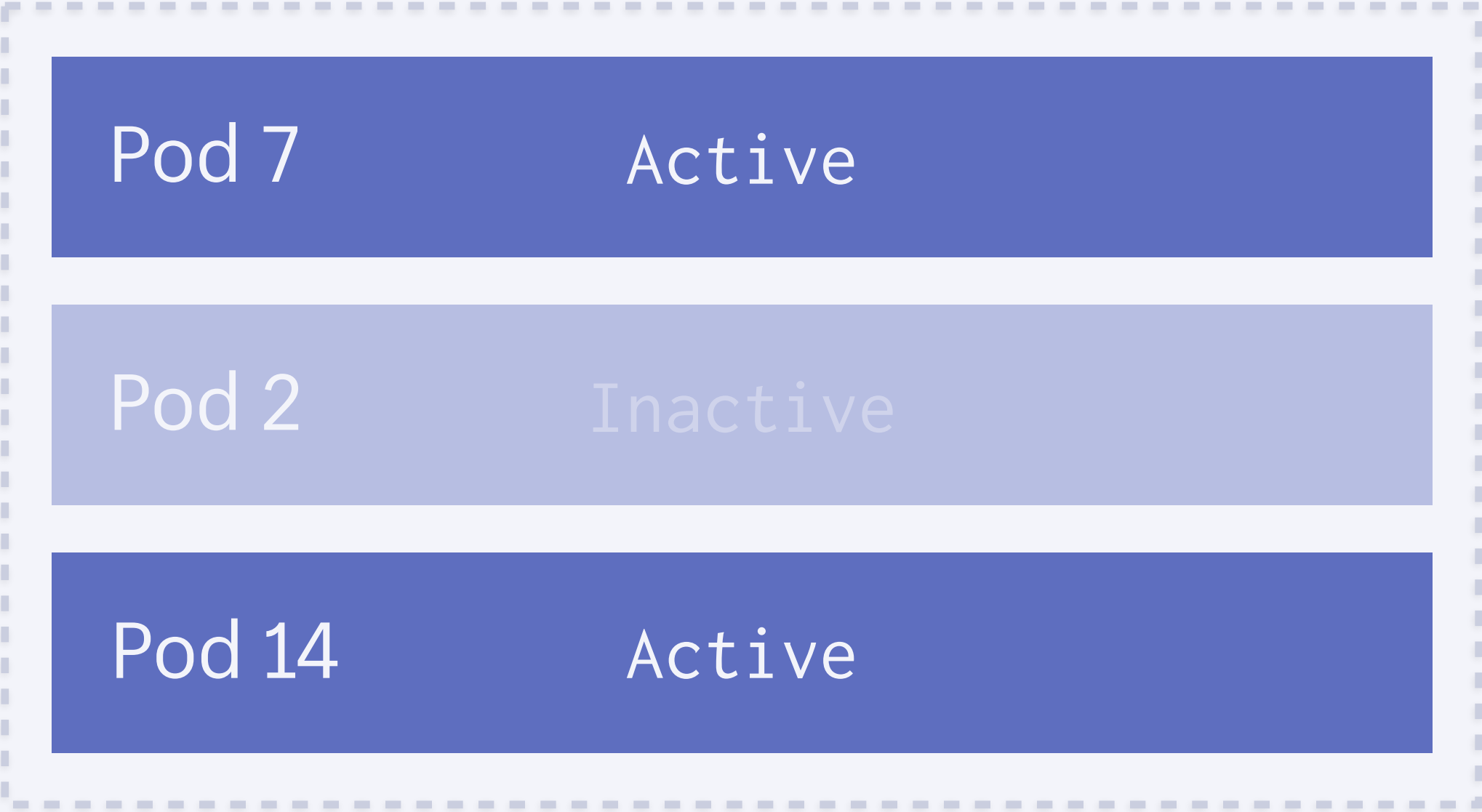
Region B



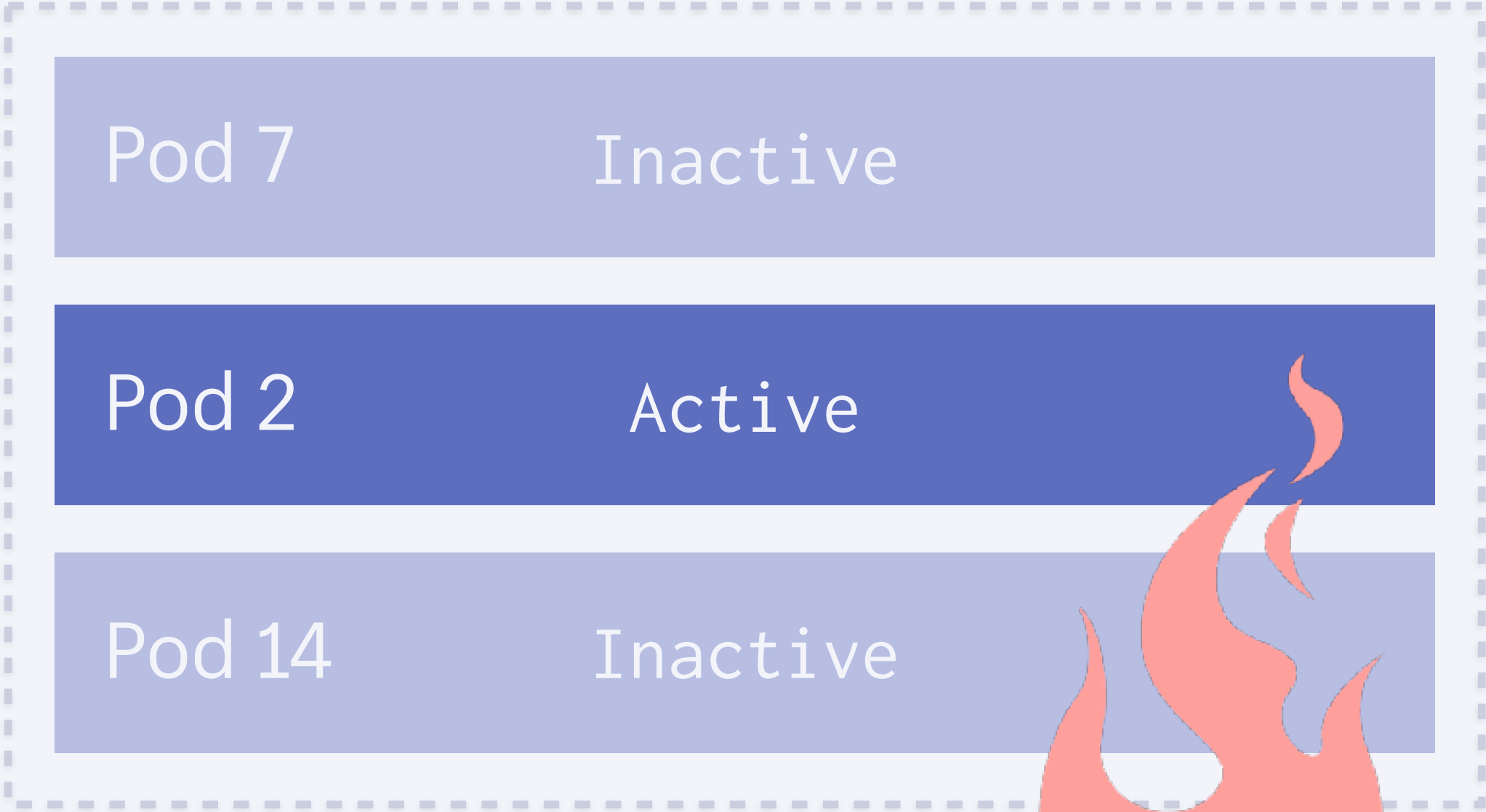


**Pod Mover** moves  
pods between  
regions with  
minimal downtime



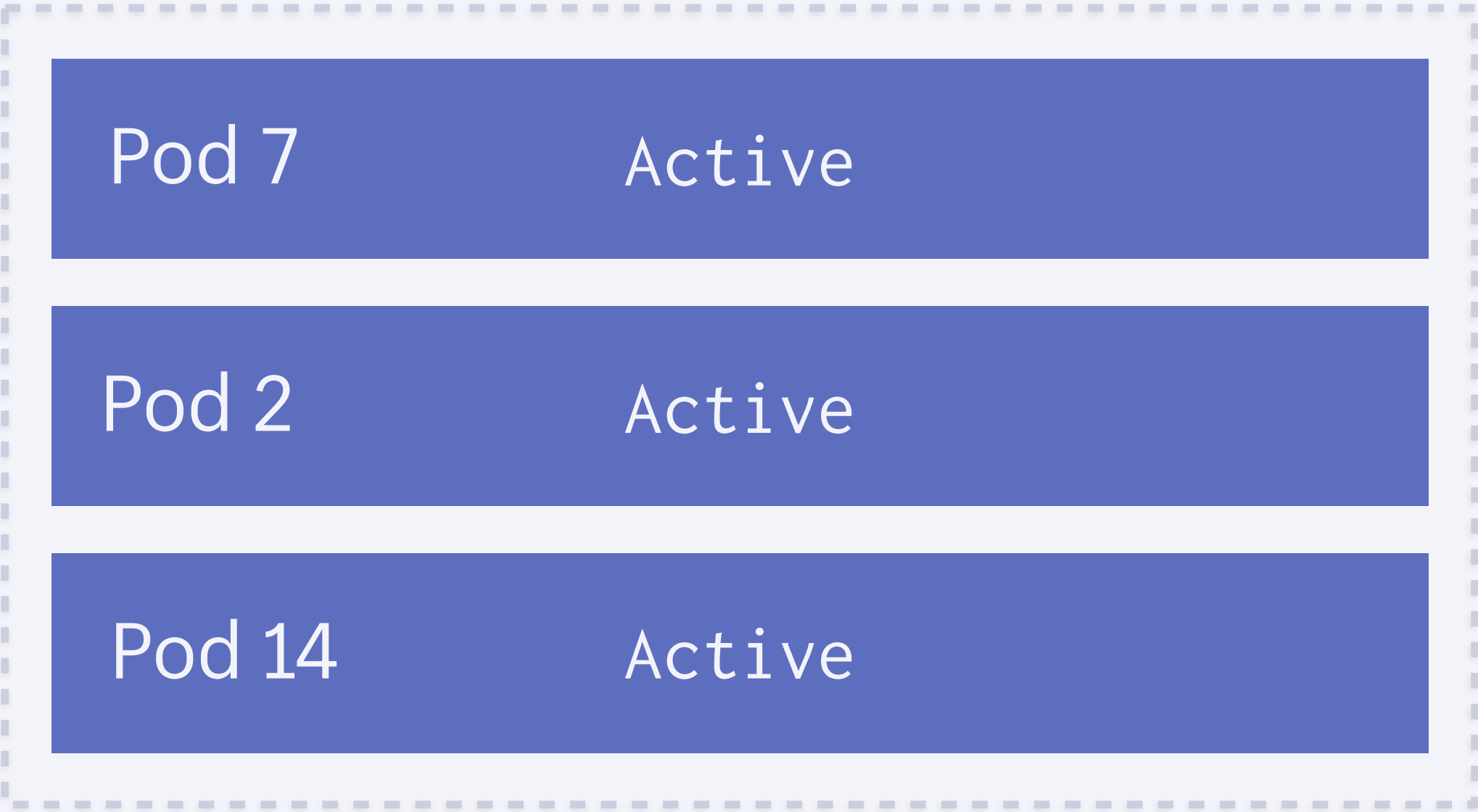


Region A

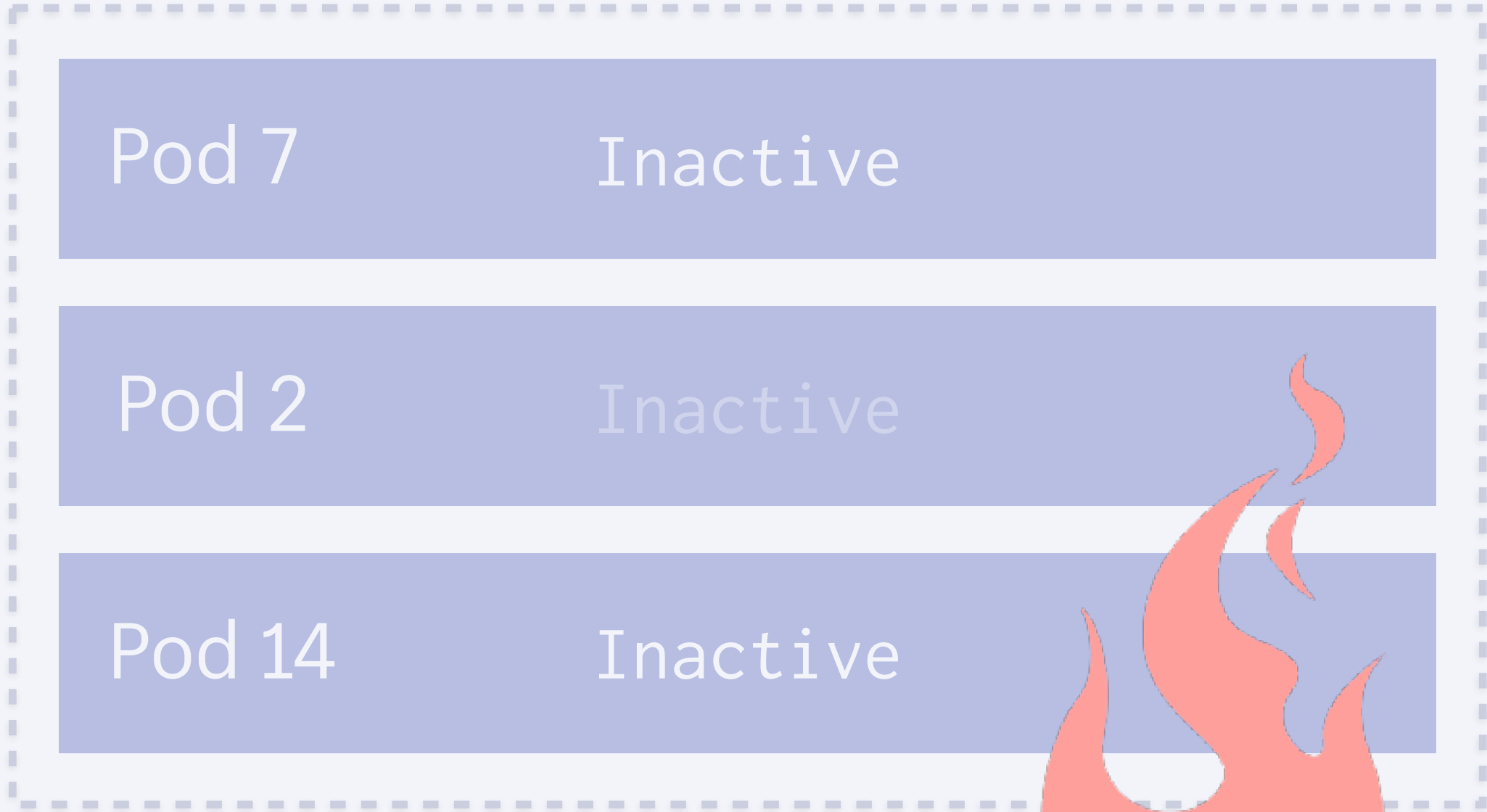


Region B





Region A

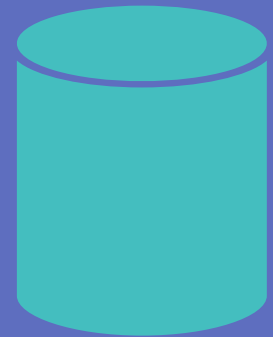


Region B





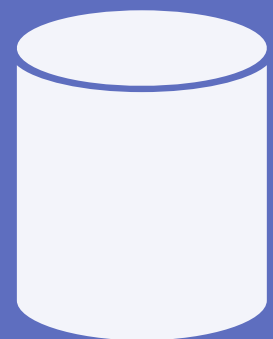
Disable cron in both regions



Update Routing for pod to target region  
pod2:b -> pod2:a



Sorting Hat routes requests to target region



Fail over MySQL to target region



Enable cron in both regions



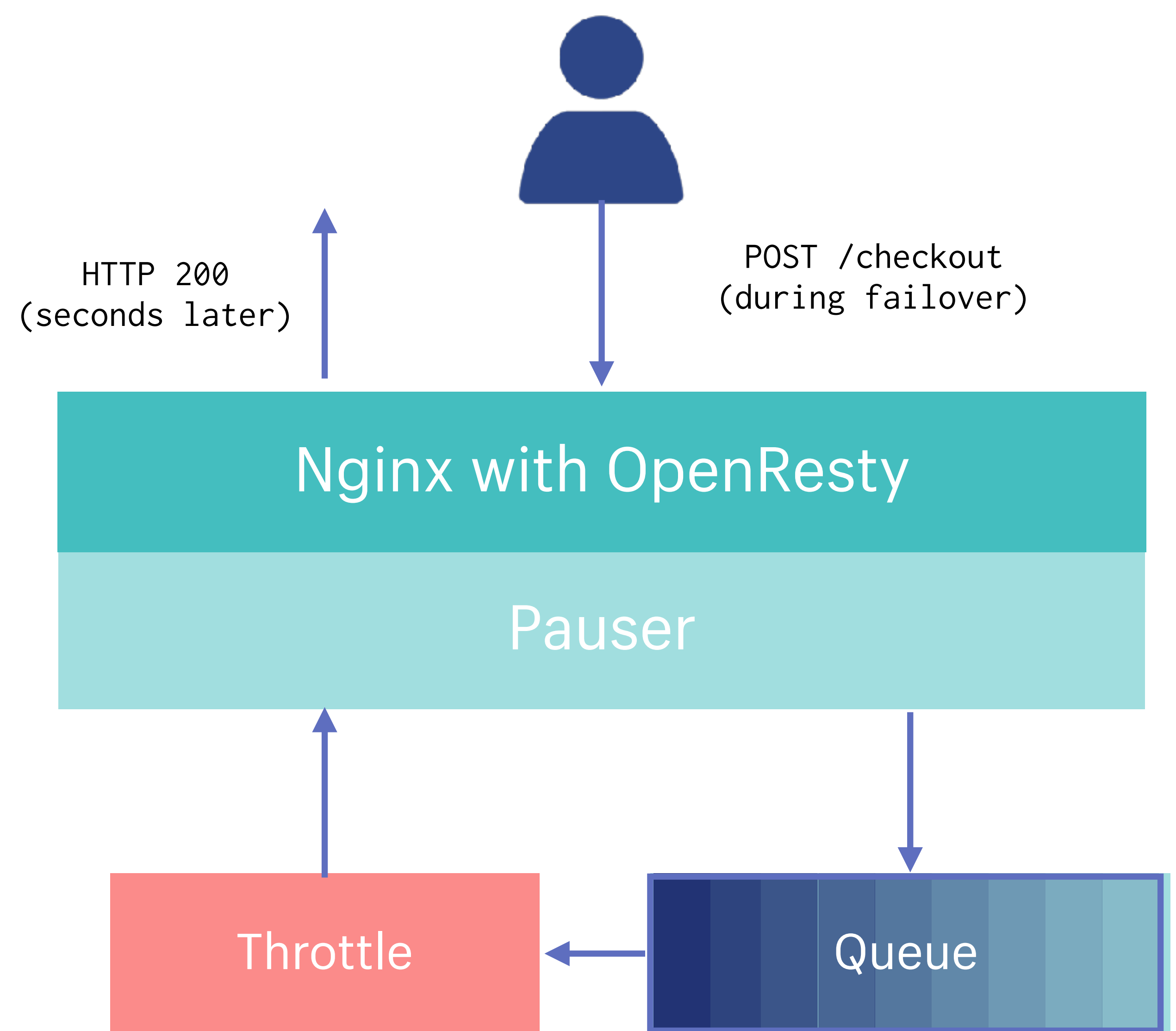
Transfer jobs to target region





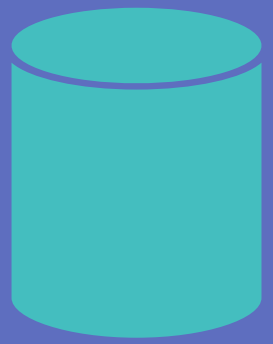
**What about errors while  
the database fails over?**

**Pauser will pause requests in the middle of failovers to avoid serving errors**





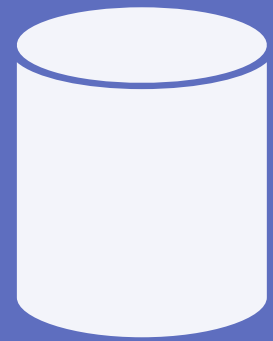
Disable cron in both regions



Update Routing for pod to target region  
pod2:b -> pod2:a



Sorting Hat routes requests to target region  
and pause requests



Fail over MySQL to target region



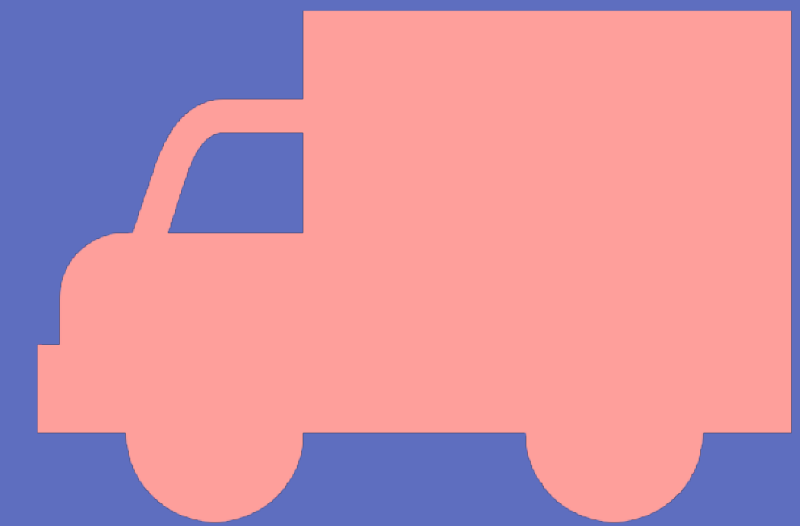
Resume requests




Enable cron in both regions



Transfer jobs to target region




18:17 **insom**  spy failover shopify pod 22 to cloud2


18:17 **spy** APP insom: [https://\[REDACTED\]/auth/aaron.brady](https://[REDACTED]/auth/aaron.brady)

 **@insom** (Aaron Brady) is running **[BETA] Failover a pod** for [activefailover/production](#)

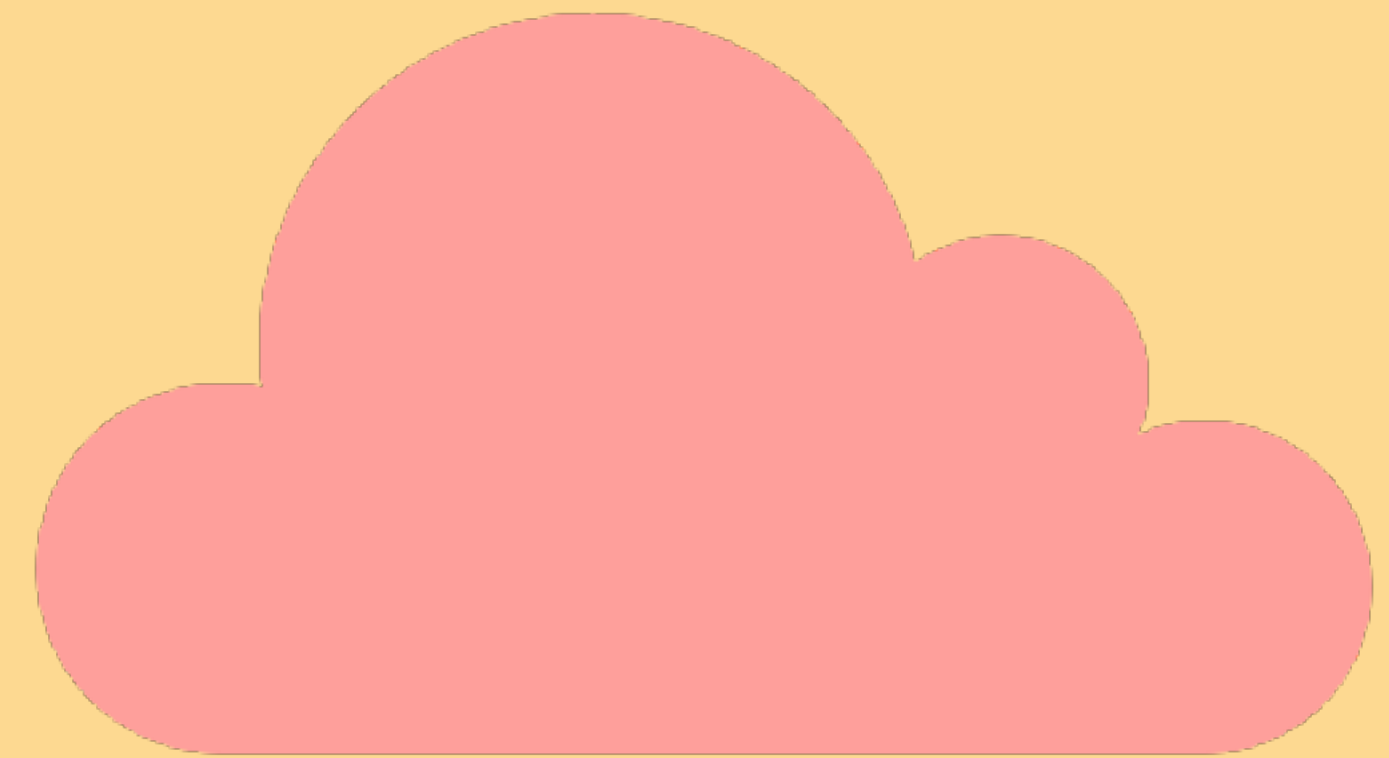
**TARGET=cloud2 POD\_ID=22** ([logs](#))

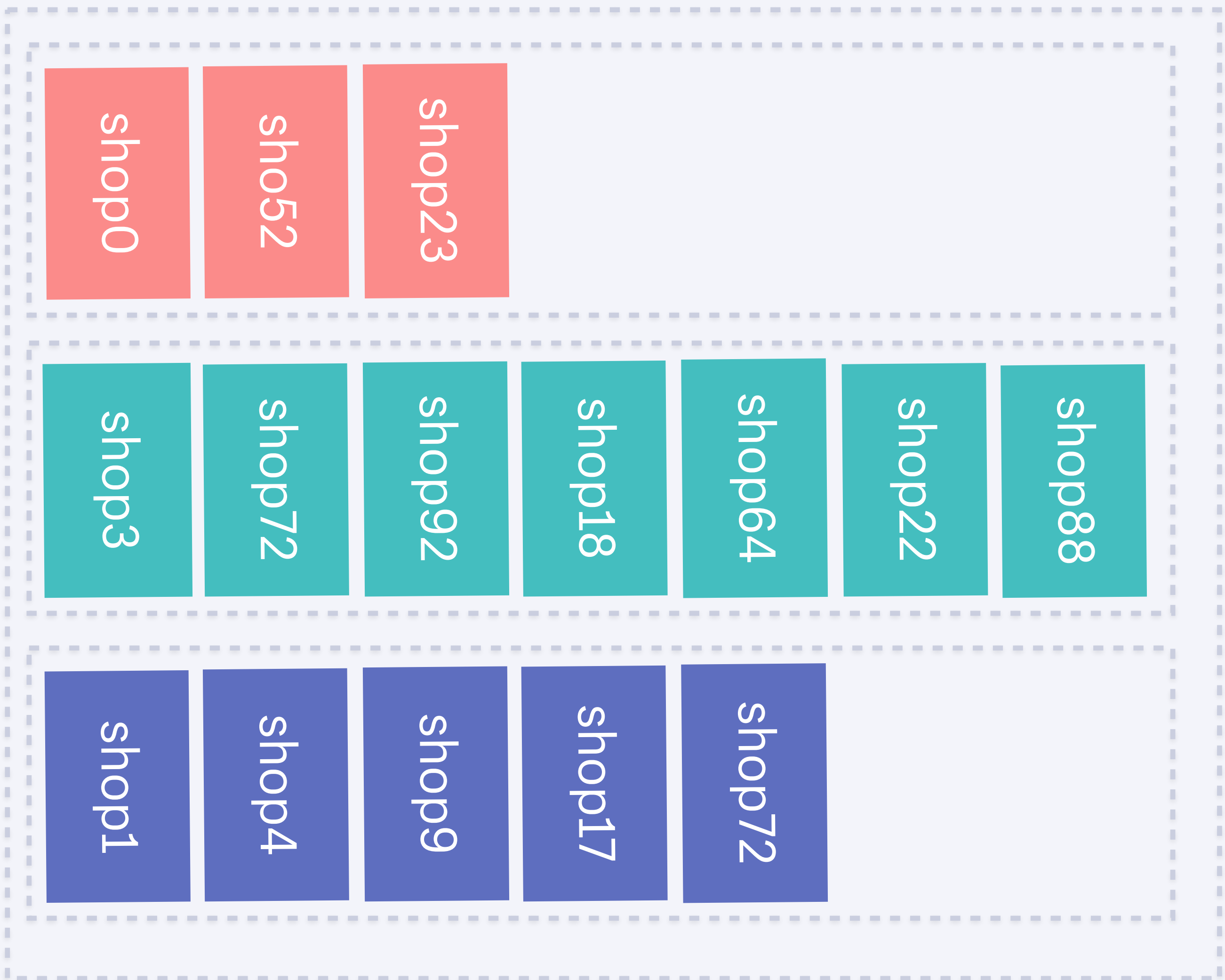
Failing over pod **22** to cloud2. See [#failovers](#)

 (deploy): A failover for pods: **22** to cloud2 has **started**.

 (deploy): A failover for pods: 22 to cloud2 **succeeded**.

# Cloud Migration with the Pods Architecture





Region A



Cloud Region C





Thanks!  
@Sirupsen





**Click 'Rate Session'  
to rate session  
and ask questions.**





*Please*

**Remember to  
rate this session**

*Thank you!*





Did you **remember**  
**to rate** the previous  
session ?

