

What I Wish I Knew Before Choosing Spot Instances

Lasse Canth Hels







Hi





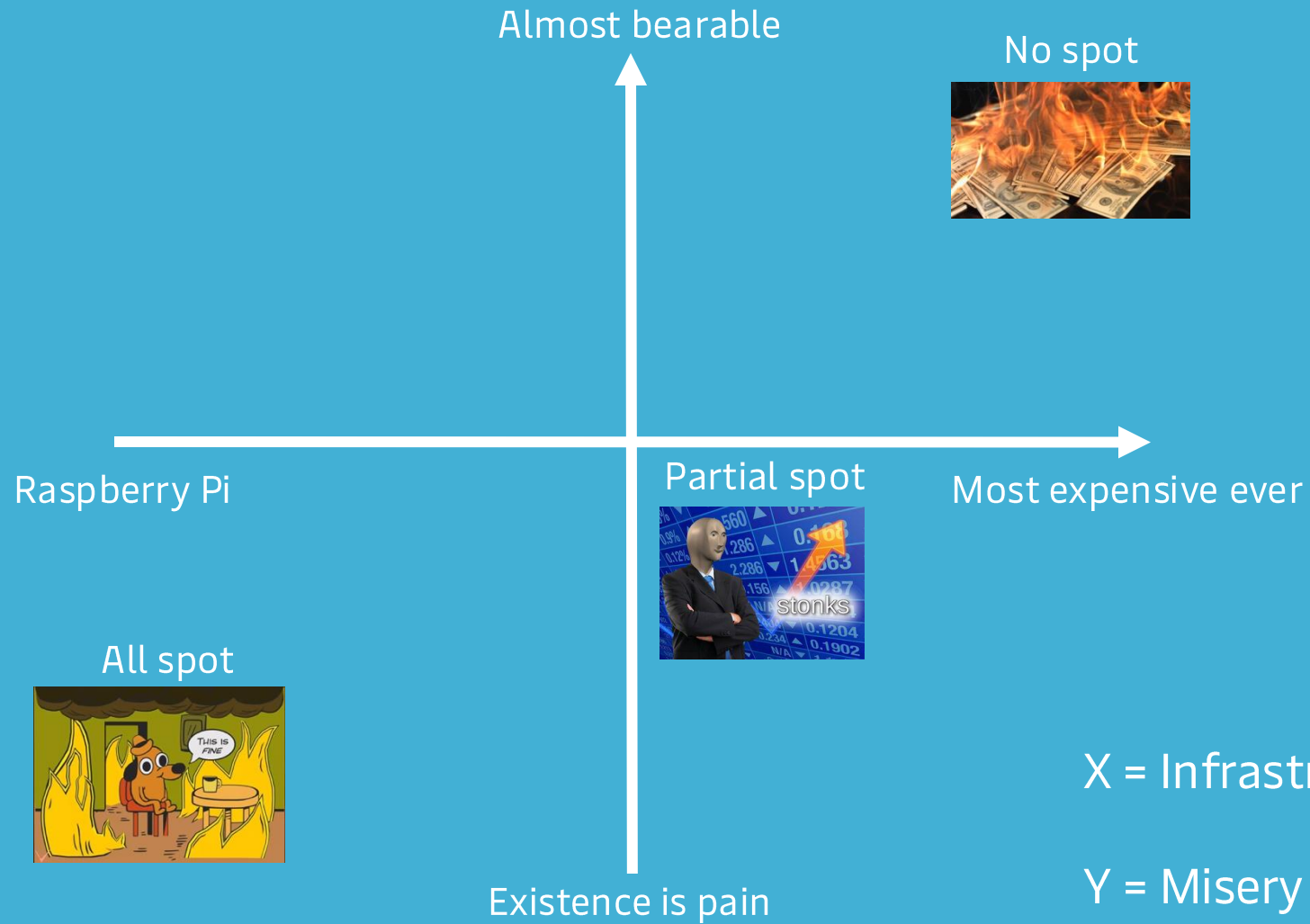
~8000 pods across ~330 nodes across five clusters



Why should we care about any of this?

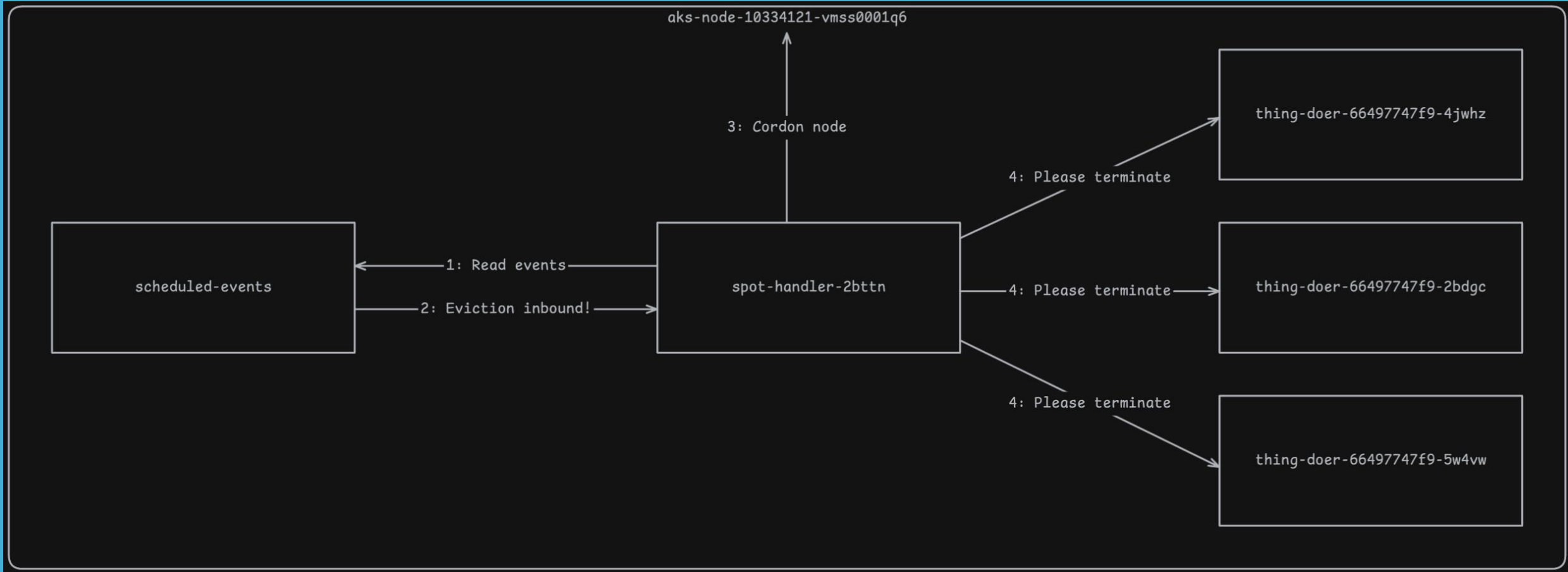


Choose your difficulty



Lesson #1: Workloads *must* terminate gracefully

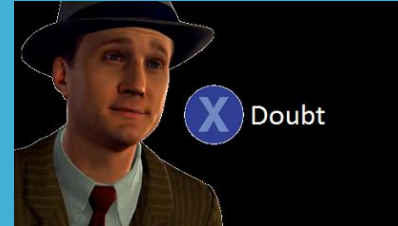
Anatomy of an eviction



Swift evictions

Swift evictions randomly cause fun issues like disk corruption and dropped requests.

infrastructure maintenance is going to affect them. Evictions qualify as infrastructure maintenance. Azure sends out the Preempt signal to all VMs at a minimum of 30 seconds before they're evicted. The Scheduled Events service



"Spot instances provide no guarantee of anything"

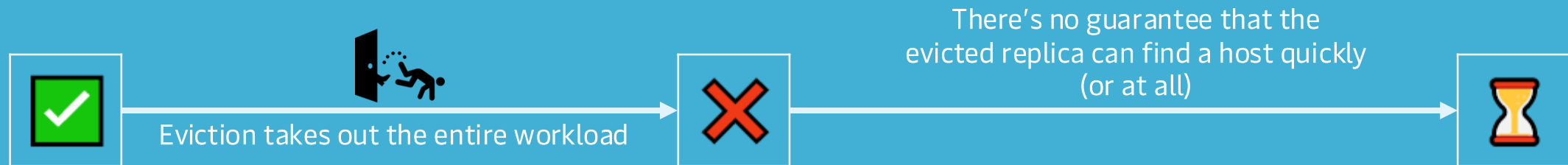
Unironically a perfectly acceptable stance



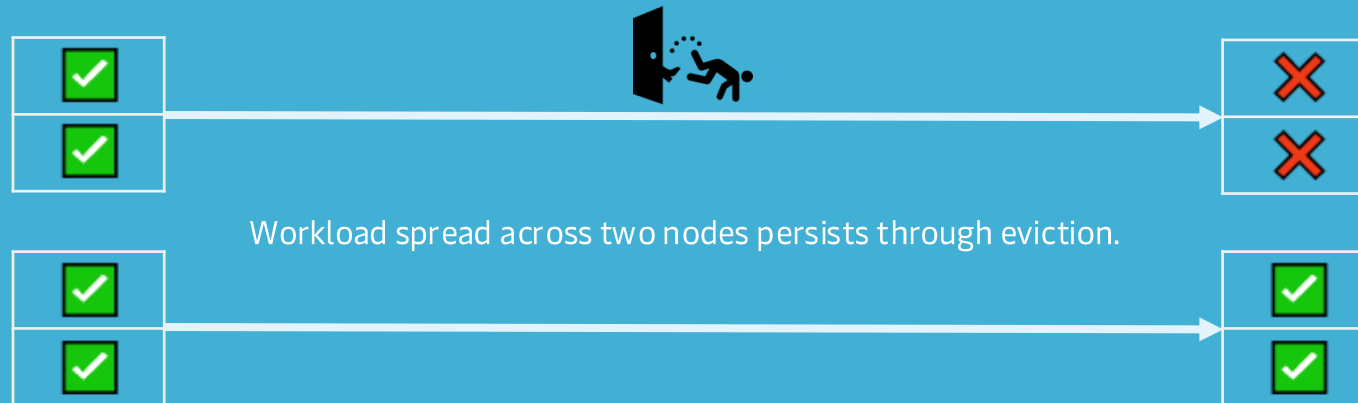
Lesson #2:

Unpredictable eviction patterns *will* expose gaps in our system's resilience

No single-replica workloads!



Workloads may not co-locate



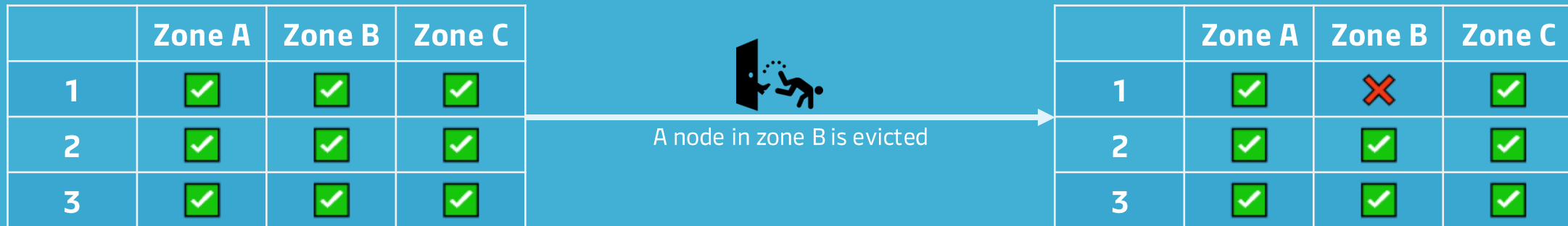
Timing-based evictions



Multi-zone evictions

Pod	State	Last Heartbeat
ingester-zone-a-0	✓	5s ago
ingester-zone-a-1	✓	4s ago
ingester-zone-a-2	✓	5s ago
ingester-zone-b-0	✓	0s ago
ingester-zone-b-1	✓	2s ago
ingester-zone-b-2	✓	3s ago
ingester-zone-c-0	✓	7s ago
ingester-zone-c-1	✗	108s ago
ingester-zone-c-2	✓	3s ago

Multi-zone evictions



Back-to-back evictions

	Zone A	Zone B	Zone C
1	✓	✓	✓
2	✓	✓	✓
3	✓	✓	✓



B-3 and C-1's nodes are evicted.
They leave the ring.

	Zone A	Zone B	Zone C
1	✓	✓	
2	✓	✓	✓
3	✓		✓

B-3 and C-1 rejoin the ring and start
replaying their write-ahead log. This
process takes ~10 minutes.

	Zone A	Zone B	Zone C
1	✓	✓	✗
2	✓	✓	✓
3	✓	✗	✓



B-3 and C-1 are evicted *again* before
being able to replay their write-ahead
log. This time they don't leave the ring
and quorum is lost.

	Zone A	Zone B	Zone C
1	✓	✓	⌚
2	✓	✓	✓
3	✓	⌚	✓

Back-to-back evictions

The perfect storm:

- Multiple nodes in the same zone are evicted within a span of ~10 minutes.
- The evicted nodes must have at least one “overlapping” pod.
- Back-to-back evictions have to happen in two different zones at roughly the same time.
- This sequence of events was rare, *hard* to debug and always caused downtime.

Chaos compounds

Can our platform handle every conceivable combination of odd event?

Eviction = 

Swift eviction + WAL replay = 

Network fault + swift eviction = 

Swift back-to-back evictions = 

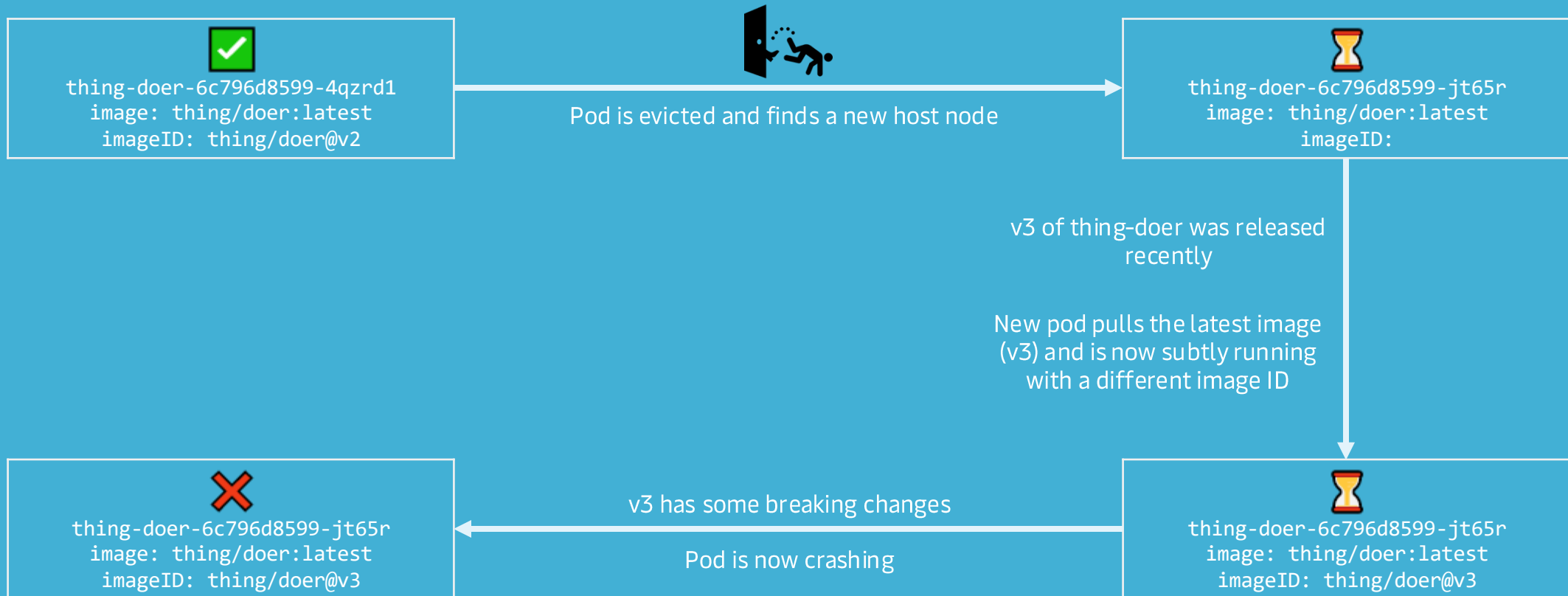
Swift multi-zone evictions = 

Node pool update + eviction = 

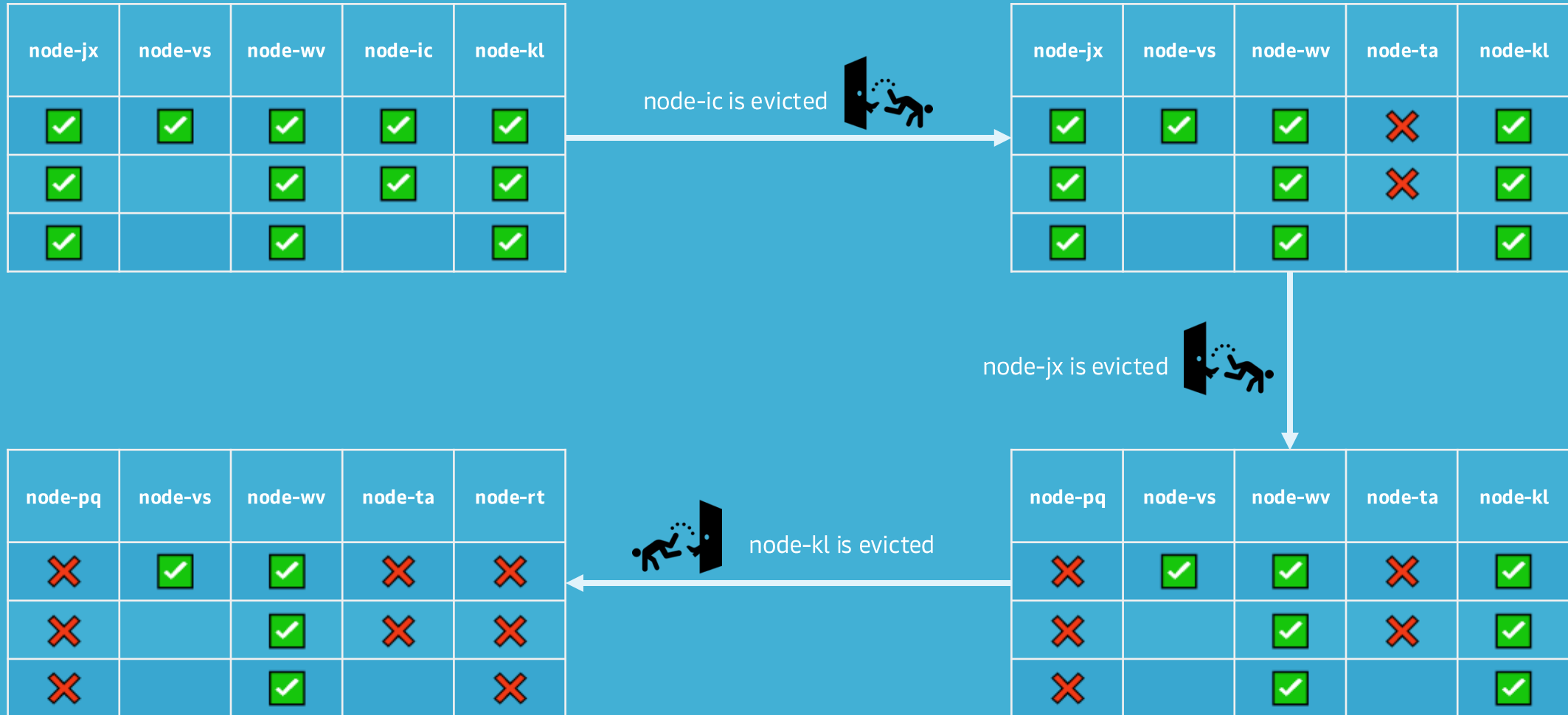
Traffic surge + back-to-back evictions = 

Lesson #3: Instance churn can be dangerous

Avoid :latest at all cost

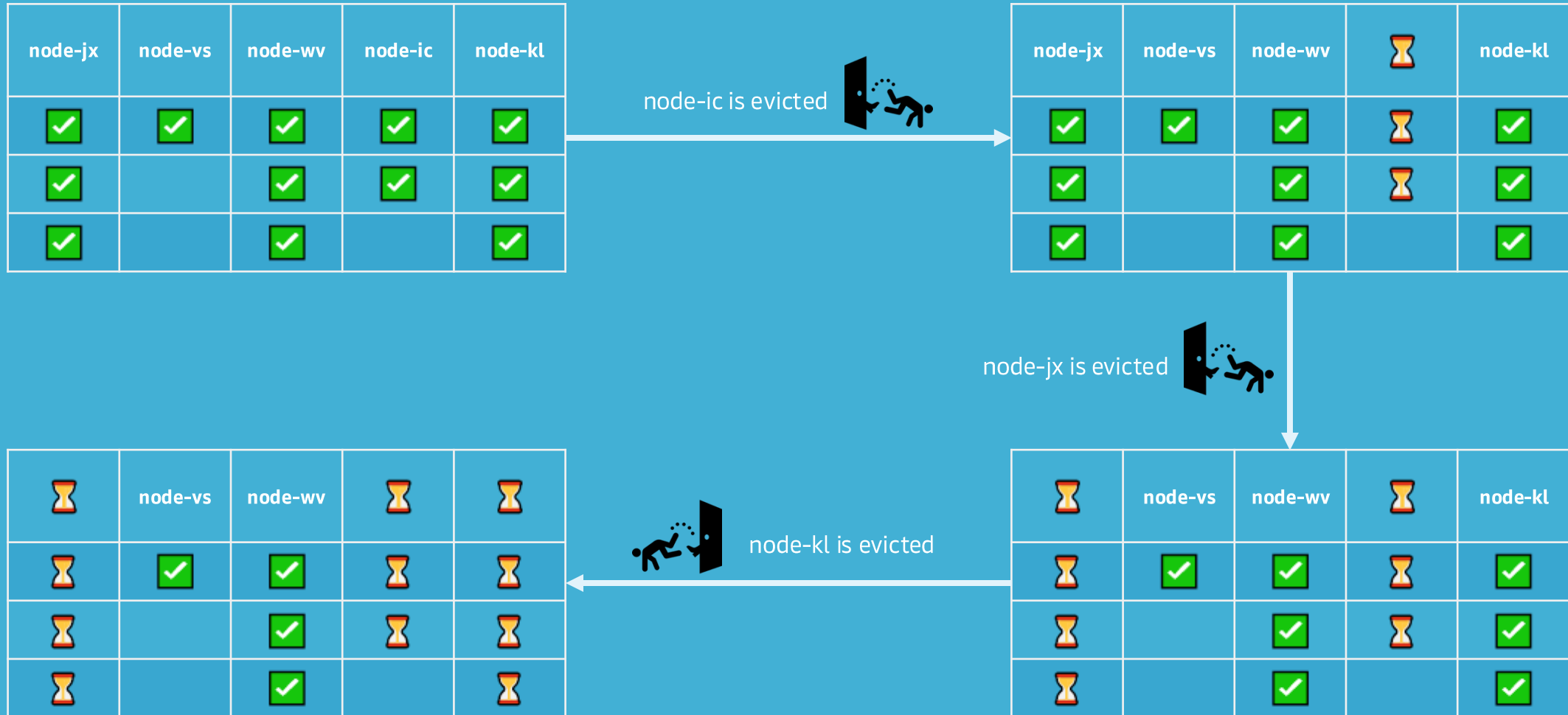


On-restart errors

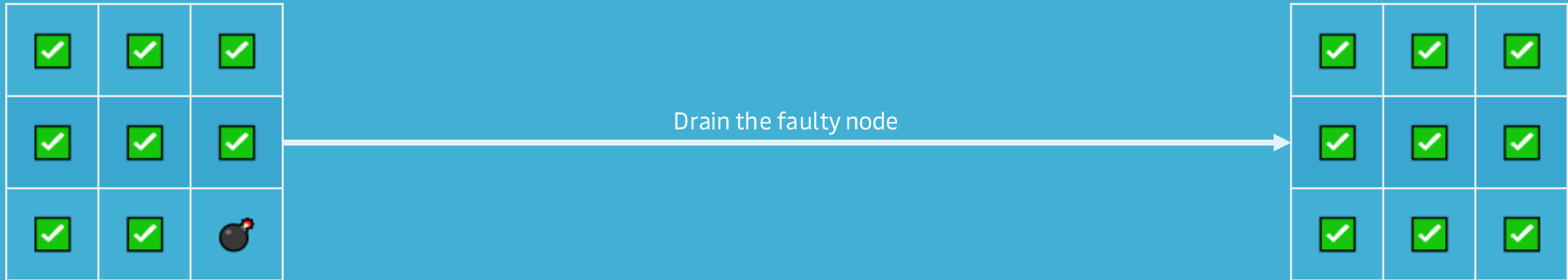


(glossing over some scheduling details)

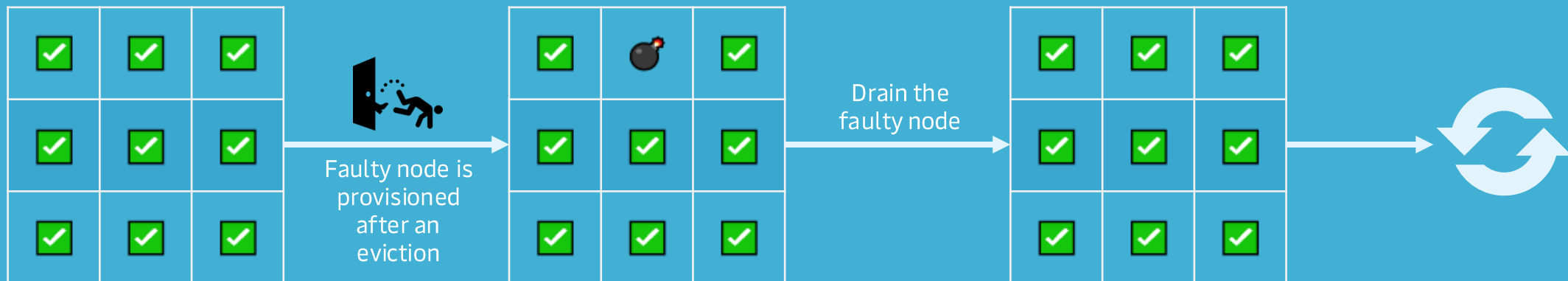
On-restart errors



Node lottery



Nodes fail all the time!
Every provisioning of a new node is a roll of the dice.



Lesson #4: Eviction observability is crucial

Eviction overview

Home > Dashboards > Node Evictions > View panel

Search or jump to...

Back to dashboard Export Share

cluster telemetry-logs-prod-we-1 region westeurope node .+ lookback 30d

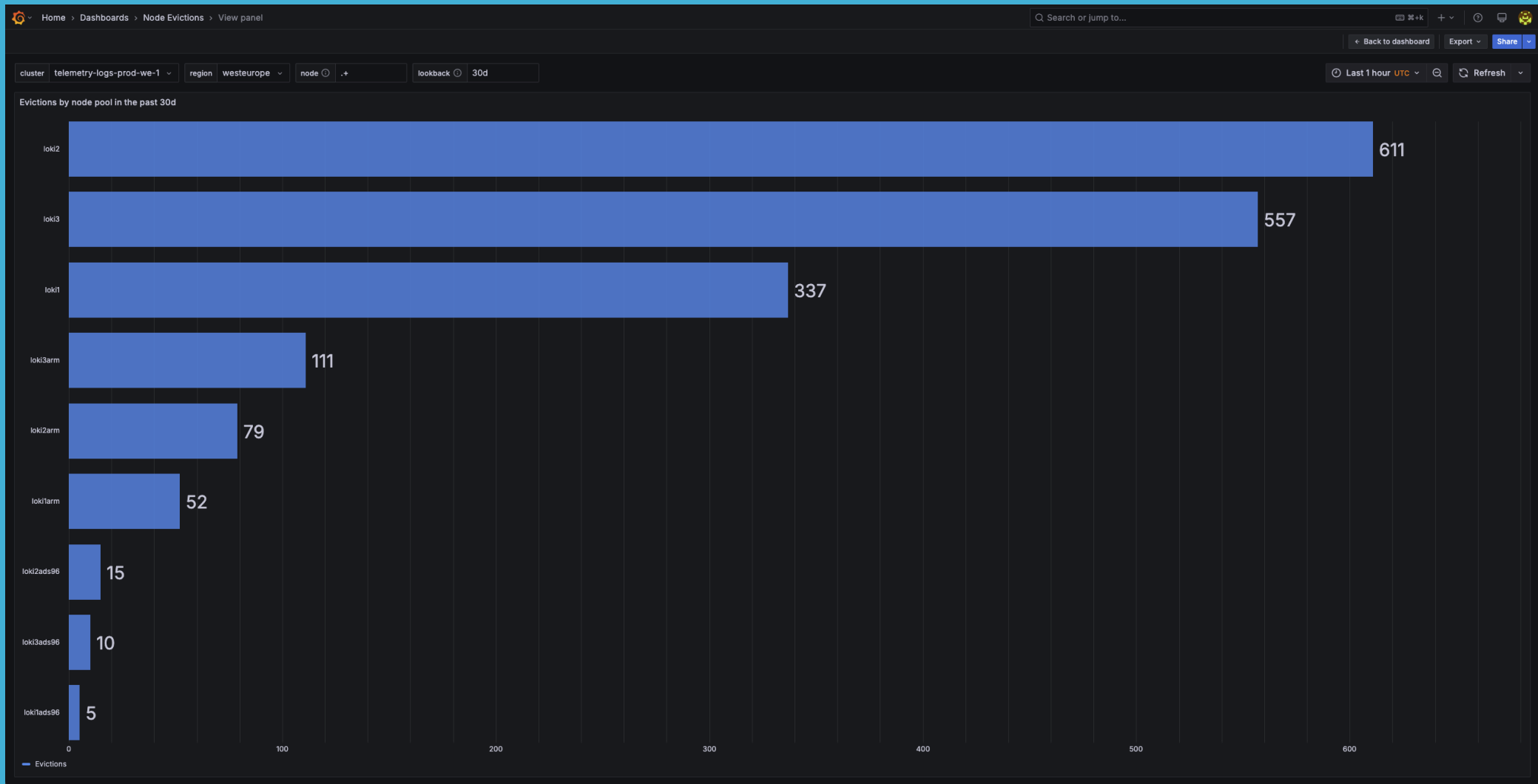
2025-08-19 00:00:00 to 2025-08-19 23:59:59 UTC Refresh

Node	Drain started ↓
aks-loki2-20973301-vmss0001jj	2025-08-19 10:41:22
aks-loki2-20973301-vmss0001jl	2025-08-19 10:41:22
aks-loki1-20465900-vmss00015c	2025-08-19 09:32:14
aks-loki3-24647283-vmss0001kw	2025-08-19 09:28:01
aks-loki3-24647283-vmss0001l9	2025-08-19 09:04:02
aks-loki1-20465900-vmss00015e	2025-08-19 08:11:37
aks-loki3-24647283-vmss0001kx	2025-08-19 08:06:33
aks-loki1-20465900-vmss000154	2025-08-19 07:53:03
aks-loki3-24647283-vmss0001l1	2025-08-19 07:52:04
aks-loki1-20465900-vmss000158	2025-08-19 07:44:50
aks-loki1-20465900-vmss000159	2025-08-19 07:44:41
aks-loki1-20465900-vmss00015a	2025-08-19 07:44:29
aks-loki1-20465900-vmss00015b	2025-08-19 07:44:19
aks-loki3-24647283-vmss0001l8	2025-08-19 07:43:48
aks-loki3-24647283-vmss0001l5	2025-08-19 07:35:28
aks-loki3-24647283-vmss0001l4	2025-08-19 07:32:23
aks-loki1-20465900-vmss000156	2025-08-19 07:32:02
aks-loki1-20465900-vmss000155	2025-08-19 07:06:04
aks-loki3-24647283-vmss0001l3	2025-08-19 07:02:02
aks-loki3-24647283-vmss0001l2	2025-08-19 06:57:55
aks-loki3-24647283-vmss0001kn	2025-08-19 06:57:53
aks-loki3-24647283-vmss0001ku	2025-08-19 06:53:24
aks-loki3-24647283-vmss0001kz	2025-08-19 06:52:07
aks-loki3-24647283-vmss0001kl	2025-08-19 06:49:28
aks-loki3-24647283-vmss0001kv	2025-08-19 06:48:47
aks-loki2-20973301-vmss0001ji	2025-08-19 06:32:20
aks-loki2-20973301-vmss0001jc	2025-08-19 06:27:17
aks-loki2-20973301-vmss0001ja	2025-08-19 06:26:57
aks-loki1-20465900-vmss000150	2025-08-19 06:26:46

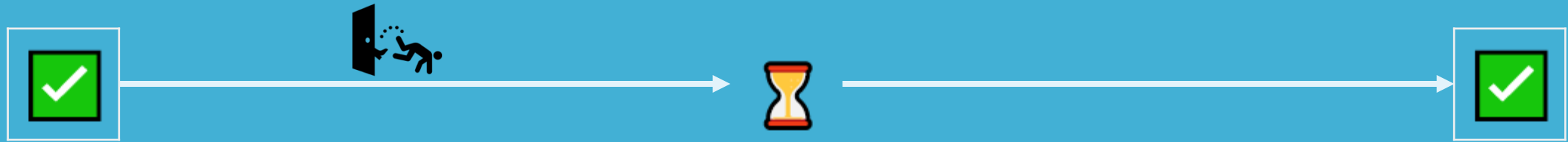
Eviction overview

```
Home > Dashboards > Node Evictions > View panel
Q Search or jump to...
Back to dashboard Export Share
cluster telemetry-logs-prod-we-1 region westeurope node aks-loki3-24647283-vmss000114 lookback 30d
2025-08-19 00:00:00 to 2025-08-19 23:59:59 UTC Refresh
Node Terminator Logs (aks-loki3-24647283-vmss000114)
> 2025-08-19 07:32:38.429 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/events/events.go:184", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/events.sendEvent", "level": "info", "msg": "Message: \u0026{EventId:69499604-EF75-4FF9-AB55-526B8EE7B013 EventType:Preempt ResourceType:VirtualMachine Resou", "time": "2025-08-19T07:32:38Z"}
> 2025-08-19 07:32:38.637 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/events/events.go:62", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/events.ReadEvents", "level": "info", "msg": "Stop reading events", "time": "2025-08-19T07:32:38Z"}
> 2025-08-19 07:32:38.428 10819 07:32:38.427983 1 request.go:697] Waited for 1.898374113s due to client-side throttling, not priority and fairness, request: GET-https://10.0.0.1:443/api/v1/namespaces/outh2-proxy-pensieve/pods/outh2-proxy-ingestion-6b9f759c74-nhv2n
> 2025-08-19 07:32:27.935 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod outh2-proxy-pensieve/authz-5974dccc-tzap2n", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.935 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod loki/ruler-13n", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.935 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod loki/querier-8fcb6855f-1fr5vn", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.934 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod loki/querier-8fcb6855f-gphhkn", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.934 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod loki/querier-8fcb6855f-dwq6vn", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.934 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod loki/querier-8fcb6855f-2tkjcn", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.934 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod loki/ envoy-5856c567-hc8d7n", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.934 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod guard/logs-receiver-guard-795ddcc56-x9nlcn", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.934 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod guard/logs-receiver-guard-795ddcc56-srv2dn", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.933 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod guard/logs-receiver-guard-795ddcc56-lfsn5n", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.933 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod guard/logs-receiver-guard-795ddcc56-gfvldn", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.933 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "evicting pod outh2-proxy-pensieve/outh2-proxy-ingestion-6b9f759c74-nhv2n", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:27.928 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:85", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode.func1", "level": "info", "msg": "WARNING: ignoring DaemonSet-managed Pods: grafana-agent/grafana-agent-logs-meta-wtl", "time": "2025-08-19T07:32:27Z"}
> 2025-08-19 07:32:23.848 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api/api.go:62", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/api.DrainNode", "level": "info", "msg": "Draining node aks-loki3-24647283-vmss000114", "time": "2025-08-19T07:32:23Z"}
> 2025-08-19 07:32:23.756 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/events/events.go:123", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/events.readEndpoint", "level": "info", "msg": "{\n  \"DocumentIncaration\": 1,\n  \"Events\": [\n    {\n      \"EventId\": \"69499604-EF75-4FF9-AB55-526B8EE7B013\",\n      \"EventStatus\": \"Scheduled\",\n      \"Time\": \"2025-08-19T07:32:23Z\"\n    }\n  ]\n}", "time": "2025-08-19T07:32:23Z"}
> 2025-08-19 07:08:34.127 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/web/web.go:42", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/web.Start", "level": "info", "msg": "web address:17923", "time": "2025-08-19T07:08:34Z"}
> 2025-08-19 07:08:33.828 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/client/client.go:45", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/client.Init", "level": "info", "msg": "No kubeconfig file use inCluster", "time": "2025-08-19T07:08:33Z"}
> 2025-08-19 07:08:33.828 {"file": "github.com/maksim-paskal/aks-node-termination-handler/pkg/alert/alert.go:29", "func": "github.com/maksim-paskal/aks-node-termination-handler/pkg/alert.Init", "level": "warning", "msg": "not sending Telegram message, no token", "time": "2025-08-19T07:08:33Z"}
> 2025-08-19 07:08:33.828 {"file": "github.com/maksim-paskal/aks-node-termination-handler/cmd/main.go:55", "func": "main.main", "level": "info", "msg": "Starting 1.0.13-d8d571-1707463489...", "time": "2025-08-19T07:08:33Z"}
```

Eviction statistics



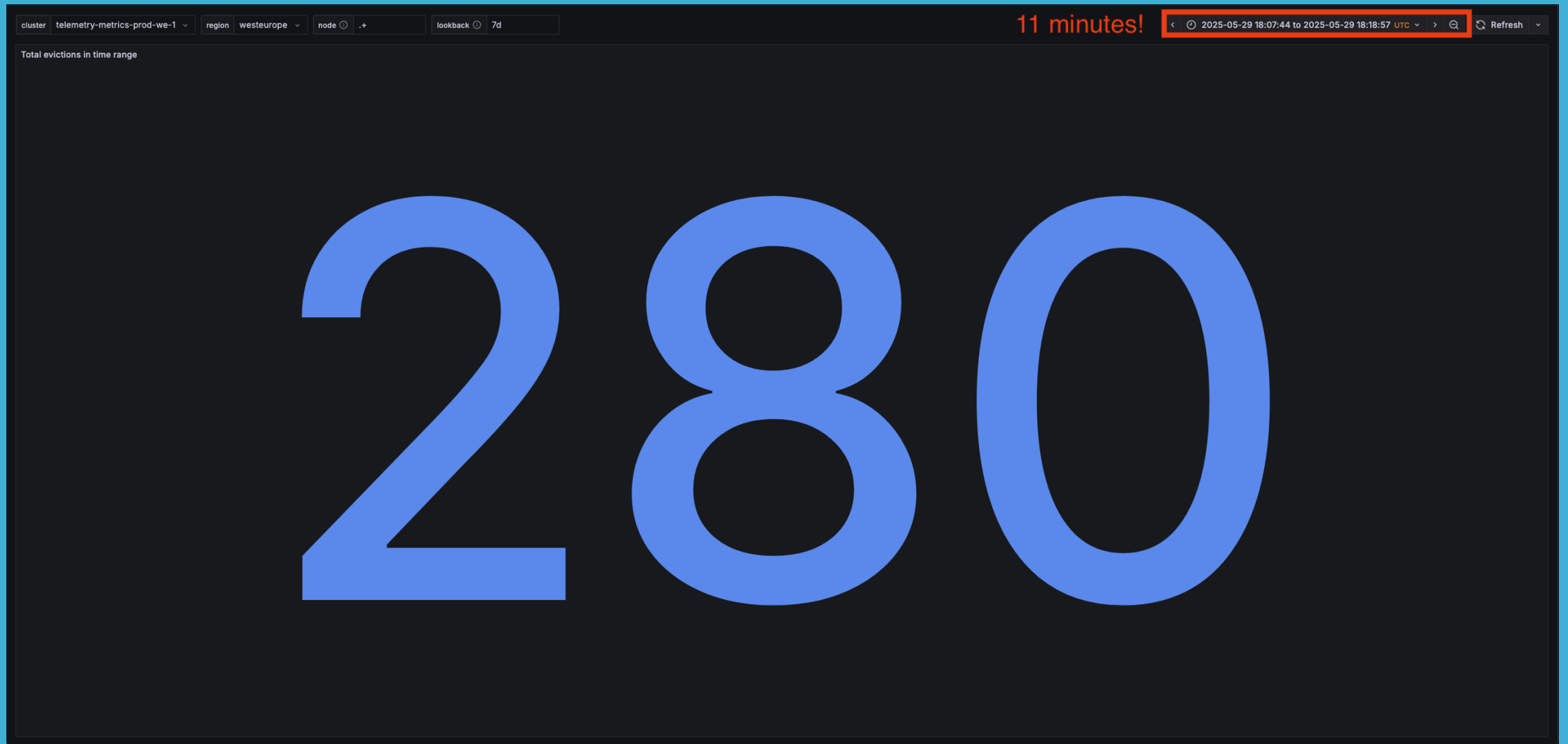
Time-to-provision



Node provisioning can take an arbitrary amount of time and fail for a variety of reasons

The issue that broke the camel's back

Eviction storms



Eviction storms



Conclusion

- Running all workloads on spot instances is not for the faint of heart.
- Spot instances punish bad practices.
- Spot instances will challenge our trust in the system.
- Spot instances increase the risk of engineer burnout.
- Running spot instances has made me a better engineer.

Q&A