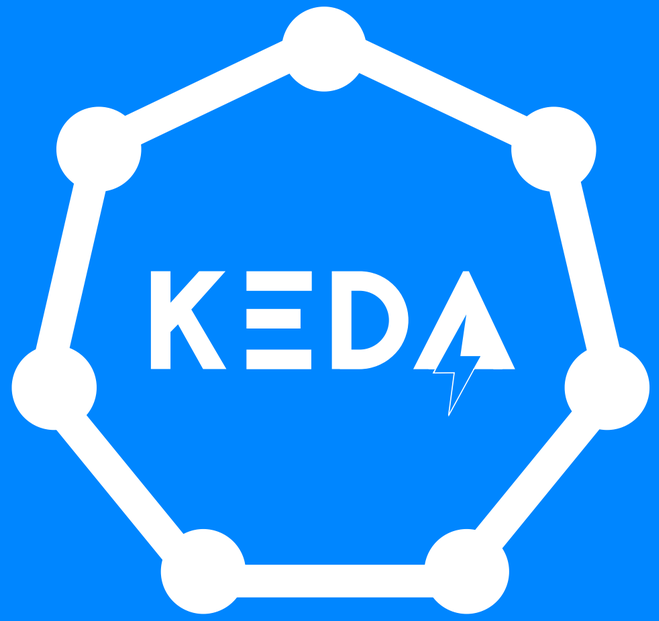
Battle-tested autoscaling paradigms with KEDA



Christian Melendez, AWS
Jan Wozniak, Kedify



10
outstanding questions

18
workers ready

0
pods waiting for a node

0 s
since the first tap

Kubernetes Event-driven Autoscaling

HPA
More or fewer replicas

VPA
CPU / memory sizing

Workload signals
Sessions · outstanding work · queues

→

KEDA
Supplies external metrics

→

Replica count
HPA scales the workload

→

Quiz workers
More CPU inference capacity



```yaml
kind: ScaledObject
spec:
  triggers:
    - type: cpu
      metricType: Utilization
      metadata: {value: "70"}
    - type: kedify-otel
      metadata:
        metricQuery: active_sessions
    - type: prometheus
      metadata:
        query: max_over_time(work_queue[60s])
    - type: cron
      metadata:
        timezone: America/Denver
        start: "0 16 11 11 *"
        end: "40 16 11 11 *"
        desiredReplicas: "3"
```

Metric types

Reactive **Resource usage**

Resources used by work already running

Proactive **Application demand**

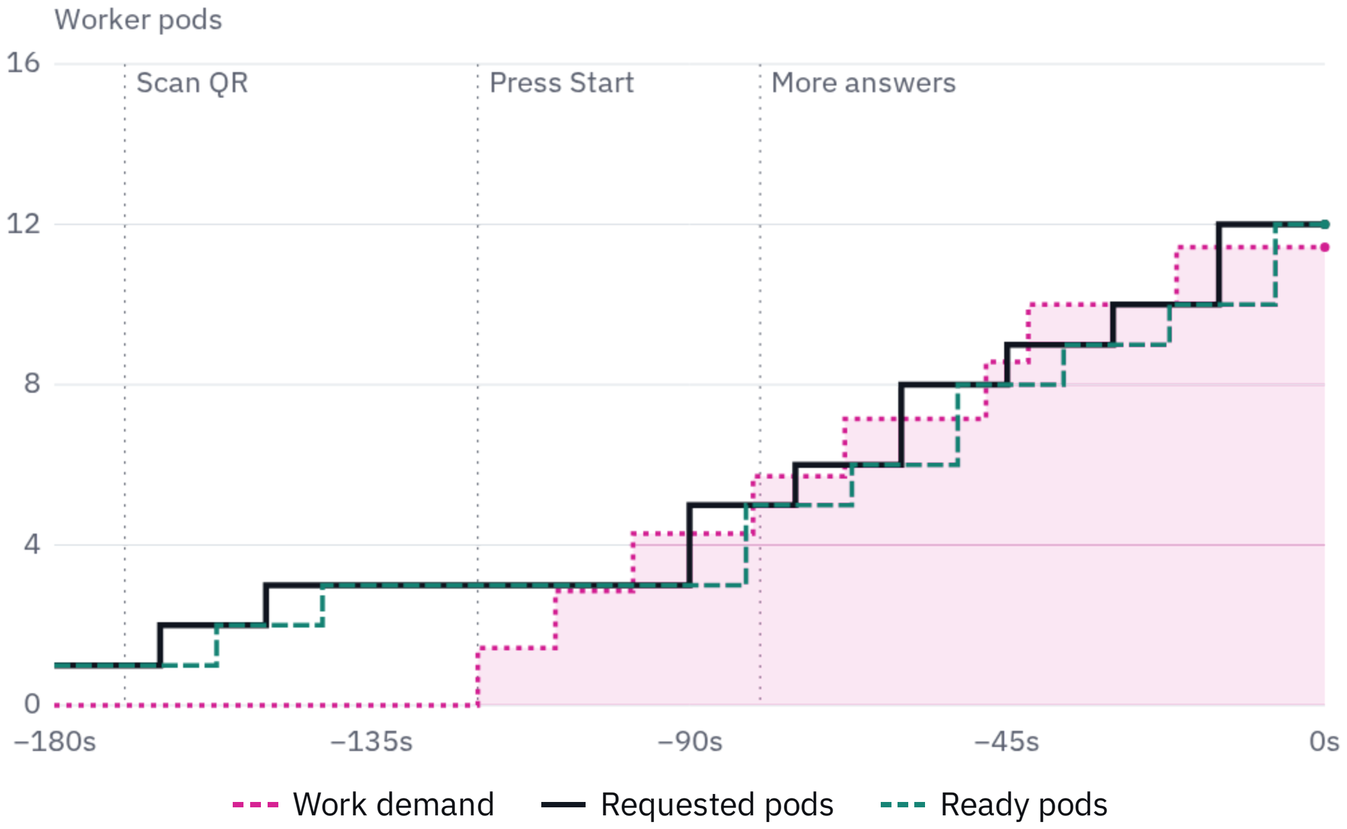
Incoming requests and unfinished work

Predictive **Schedule / forecast**

Prepare for demand you expect

How many workers, and why?

Simulated

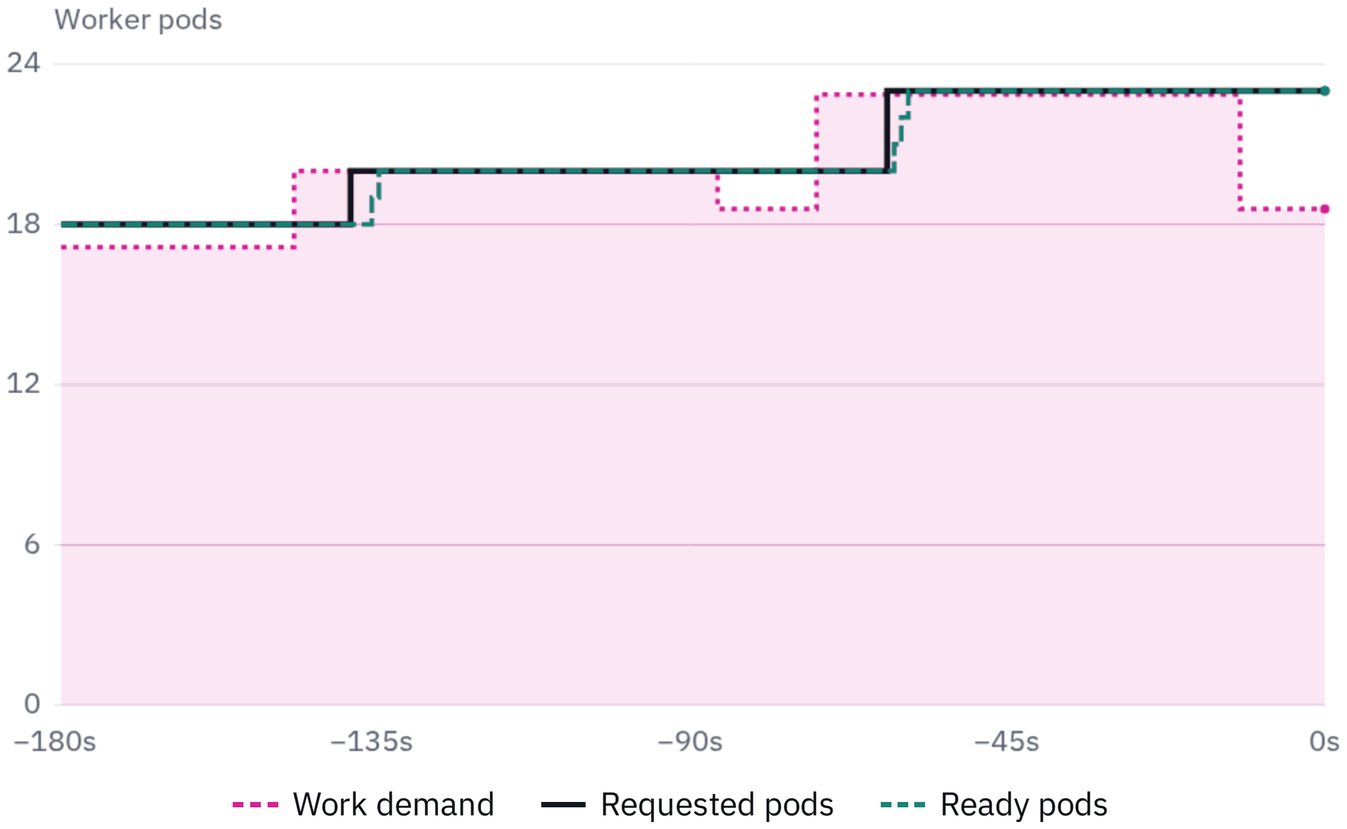


Signal	Measurement	Pods
Sessions	56	3
Work	8 questions	12
CPU	12.10 cores	9
Memory	9400 MiB	8

Largest visible: **12** pods

How many workers, and why?

Live



Signal	Measurement	Pods
Sessions	118	3
Work	13 questions	19
CPU	18.53 cores	13
Memory	8133 MiB	7

Largest visible: **19** pods

Service-level indicators

Simulated

SLO: **95%** successful in under **10s**

Successful under 10s	p95 user wait	Mean user wait
100.0%	**2.9s**	**2.7s**

Service-level indicators

Live

SLO: **95%** successful in under **10s**

Successful under 10s

99.9%

p95 user wait

2.9s

Mean user wait

2.1s

What makes scaling reliable?

1 **Fresh metrics**

A dedicated scaling path

2 **Resilient decisions**

ScalingModifiers and failover

3 **Ready capacity**

Dynamic buffers and cost

4 **Stable scaling**

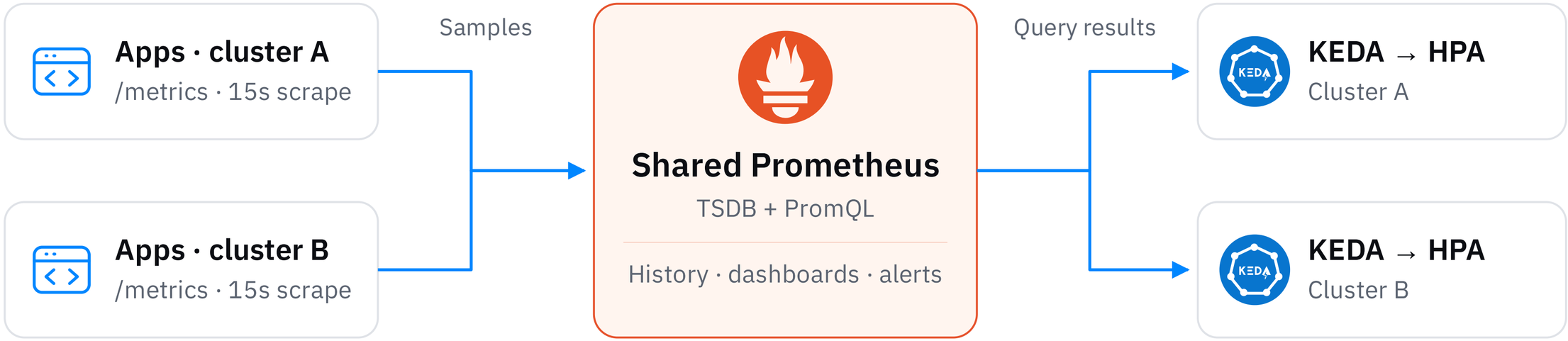
Scale-down windows and bursty traffic

1 / 4

Fresh metrics

A dedicated scaling path

Prometheus: the shared query path



100 ScaledObjects

~7 queries/s

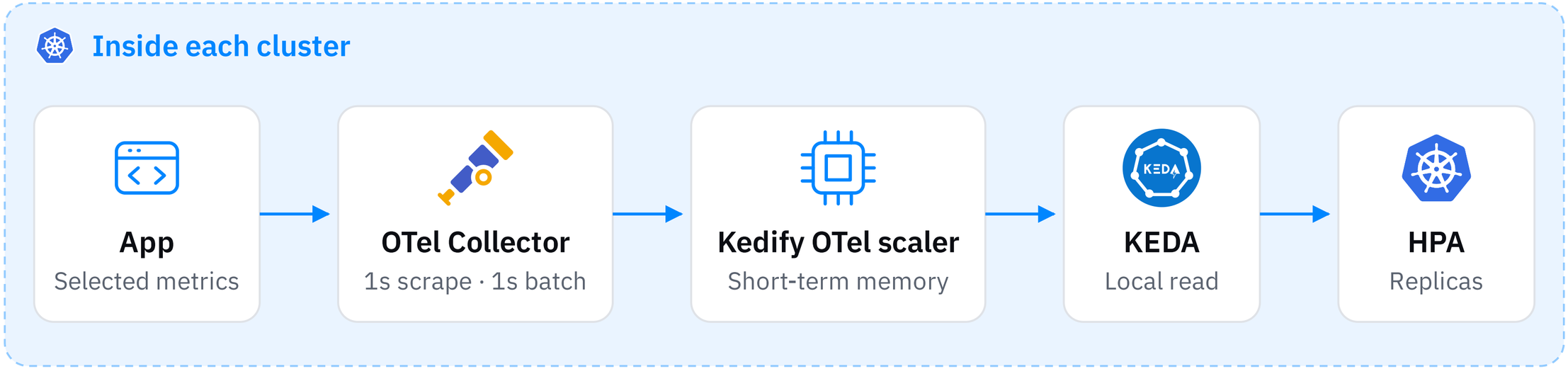
1,000 ScaledObjects

~67 queries/s

5 clusters × 1,000

~333 queries/s

OTel: a dedicated scaling path



Select

Only useful metrics and labels

Tune

Collection · batching · retention

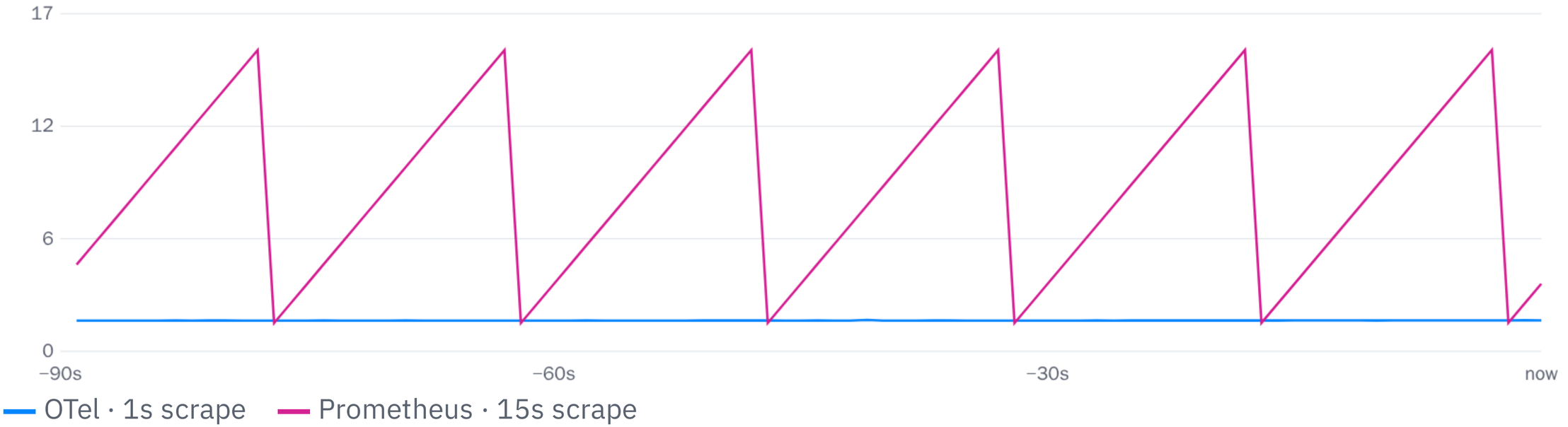
Isolate

Separate from shared observability

How fresh is the signal?

Sample age · seconds

1.6s · 3.5s · 30.8s



OTel · work peak
valid · 1.6s old
4

Prometheus · work peak
valid · 3.5s old
3

2 / 4

Resilient decisions

ScalingModifiers and failover

```yaml
kind: ScaledObject
spec:
  triggers:
    - type: kedify-otel
      name: otel_work
      metadata:
        metricQuery: work_worker_estimate
    - type: prometheus
      name: prom_work
      metadata:
        query: max_over_time(work_queue[60s]) / 0.7
  advanced:
    scalingModifiers:
      target: "1"
      metricType: AverageValue
      formula: otel_work ?? prom_work ?? 3
```

ScalingModifiers

Prefer

Outstanding work · OTel

Fall back

Same outstanding work · Prometheus

Last resort

Both unavailable → 3 workers

Scaling resilience

Live · read-only

OTel

CrashLoopBackOff

0/1
Ready replicas

1
Pod restarts

Metric health
Unavailable

Outstanding work
—

Sample age
—

Deployment ↗ Pod details / logs ↗

Prometheus

Running

1/1
Ready replicas

0
Pod restarts

Metric health
Usable

Outstanding work
3

Sample age
5.22s

Deployment ↗ Pod details / logs ↗

Scaling decision

KEDA → HPA → workers

4.29
KEDA composite

→

5
HPA requested

→

5
Workers Ready

HPA details ↗
Replica cap 24

Recovery timeline

Observed in this browser

09:57:24	**OTel**	Process exited
09:56:54	**OTel**	Metric unavailable
09:56:52	**OTel**	Pod not Ready

Lose a feed, keep serving

OTel · work peak 3

valid · 1.6s old

OTel · Running · 1/1 ready · 0 restarts

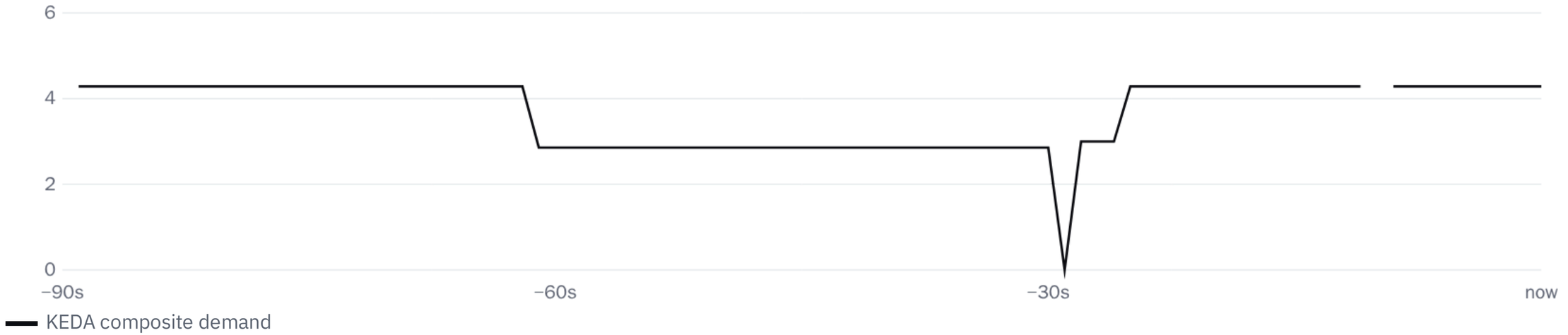
Prometheus · work peak —

invalid · unavailable

Prometheus · CrashLoopBackOff · 0/1 ready · 1 restarts

Composite demand

workers



Ready / requested

5 / 5

Successful under 10s

99.5%

p95 user wait

2.9s

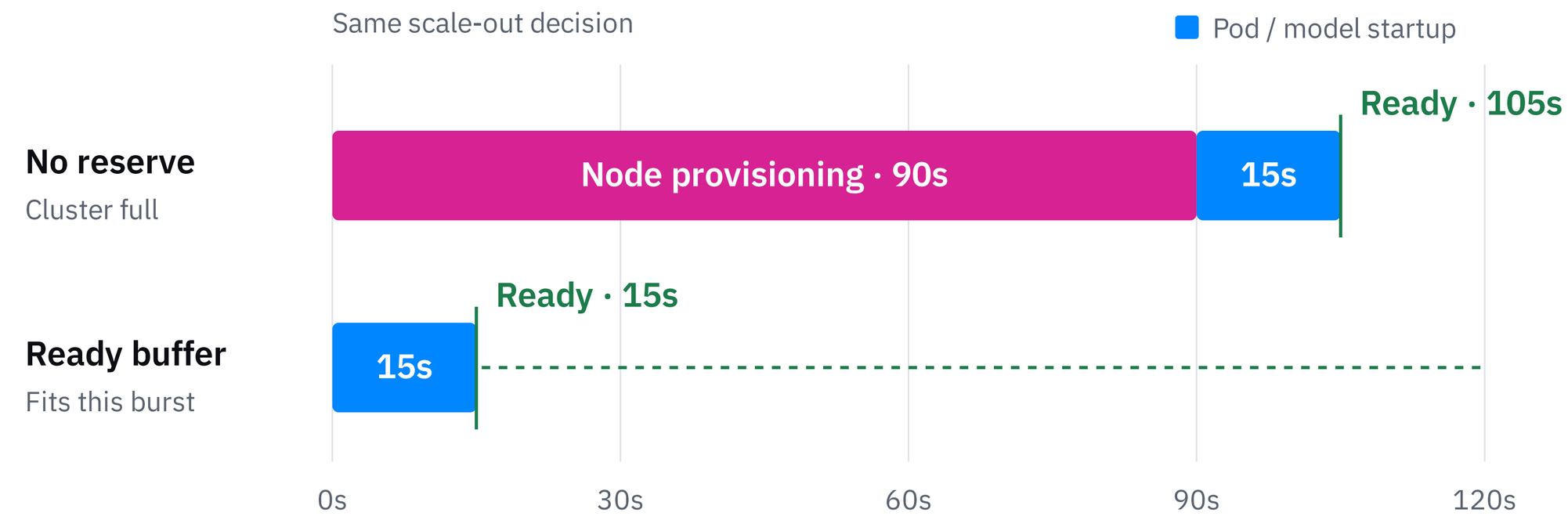
3 / 4

Ready capacity

Dynamic buffers and cost

A fast decision can still wait

Illustration · 90s node provisioning · 15s pod startup



Prepare capacity before demand reaches the scheduler

```yaml
kind: CapacityBuffer
spec:
  podTemplateRef:
    name: quiz-worker-shape
  replicas: 1
---
kind: PodTemplate
template:
  spec:
    containers:
      - name: shape
        resources:
          requests:
            cpu: 2050m
            memory: 1600Mi
```

Capacity is the next delay

Without a buffer

Provision node → start pod

With a ready buffer

Start pod

Size by workload

One replica = room for one worker

```yaml
kind: ScaleAdapter
spec:
  targetRef:
    kind: CapacityBuffer
  desiredReplicasPath: .spec.replicas
  currentReplicasPath: .status.replicas
---
kind: ScaledObject
spec:
  scaleTargetRef:
    kind: ScaleAdapter
  minReplicaCount: 1
  maxReplicaCount: 8
  triggers:
    - type: kedify-otel
      metadata:
        metricQuery: buffer_reserve_slots
```

Scale the buffer

Anticipate

Baseline + positive growth × 90s

Bound the reserve

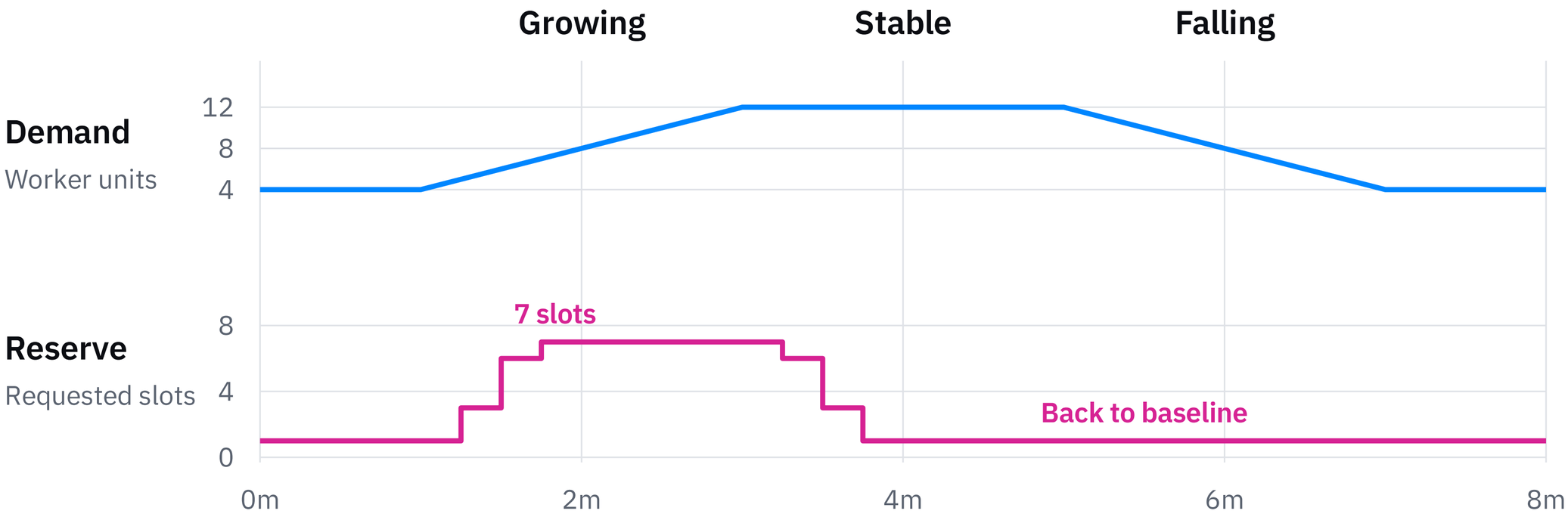
1–8 worker-sized slots

Apply

KEDA → ScaleAdapter → CapacityBuffer

Size the reserve for the next burst

Illustration · 1 spare slot baseline · 90s lead time · cap 8



Reserve = baseline + positive demand growth × node lead time

Pay for the overlap

Illustration · one extra node at $0.80/hour

Reserve policy	Extra node-hours / day	Extra cost / day
Always on	24	$19.20
Business hours	8	$6.40
4 bursts × 15 minutes	1	$0.80

4 / 4

Stable scaling

Scale-down windows and bursty traffic

Keep workers warm

```yaml
kind: ScaledObject
spec:
  advanced:
    horizontalPodAutoscalerConfig:
      behavior:
        scaleUp:
          stabilizationWindowSeconds: 0
        scaleDown:
          stabilizationWindowSeconds: 30
          policies:
            - type: Percent
              value: 50
              periodSeconds: 15
```

Scale up

No stabilization delay

Hold through pauses

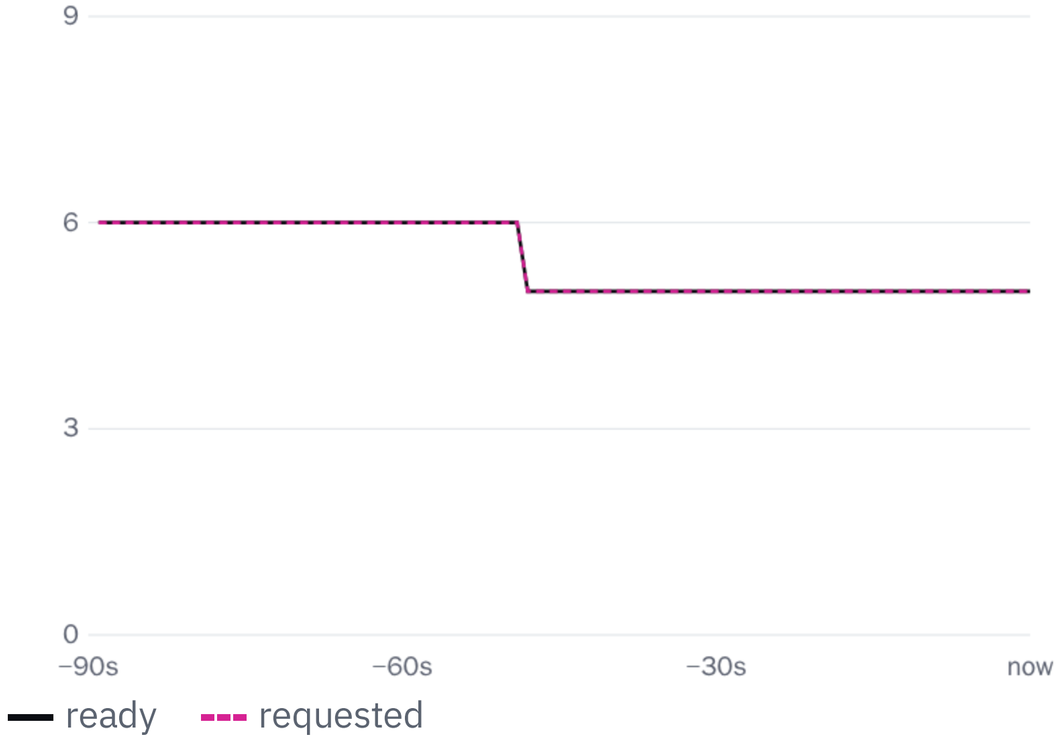
Highest recommendation over 30s

Scale down

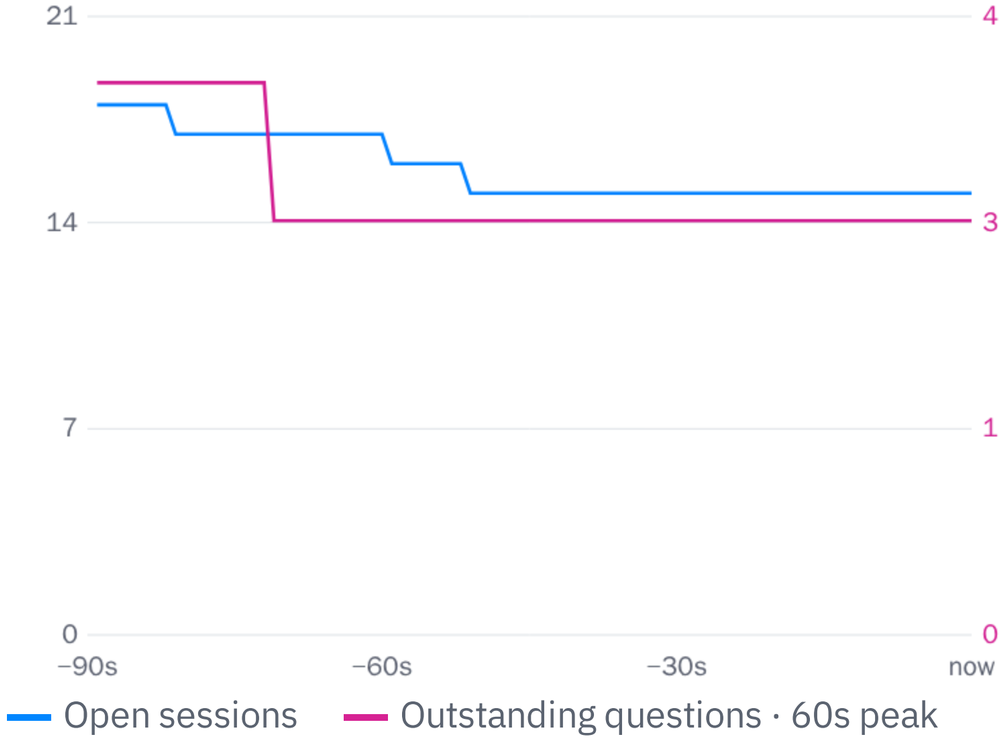
50% / 15s rate policy

Questions pause

Worker pods



Sessions and outstanding work



From a signal to a good experience

1. Deliver a fresh signal
2. Define the degraded decision
3. Prepare compatible capacity
4. Keep capacity through short pauses

For the statistics nerds

3,175
Questions generated

3,019
Answers submitted

77.7%
Correct answers
2346 right / 673 wrong

23
Peak workers

Most prolific player

James Hopper

58 answers · 49 correct

Highest accuracy
10+ answers

Alan Thompson

100.0% · 12/12 correct

Longest correct streak

Edsger Torvalds

37 correct in a row

User wait **1.8s** mean · **2.9s** p95

Thank you · Questions?

Room results ↗