Effective Containers

How to not screw up and spend a bunch of money when running your Puma app at scale.

Alexander Nicholson

> whoami

Alexander Nicholson

- Living in Tokyo, Japan
- Engineering Manager -
  Data & Site Reliability Engineering
  @ TableCheck
- Building software that venues (and their staff) love to use.
- * Opinions are my own, blah blah blah, legal document goes here.

What will we talk about?

Agenda

- **Why should I care about scaling Puma?**
- The dangers of following YouTube advice.
- What should I do?
- Q&A

Why should I care?

Some reasons to listen

- If you're a SME on AWS, you're probably burning 💰💰, 💰💰💰 to run your app. You could save some money!
- You use EKS or GKE or some other container orchestration for your Rails app.
- Your app crashes under large amounts of traffic, and scaling doesn't seem to be as effective as you expected.
- You would like to improve your application's performance for basically "free".
- This is a gentle introduction. If you want more (and better) information... go ask Nate!
- This is just an introduction, so I will make generalisations.

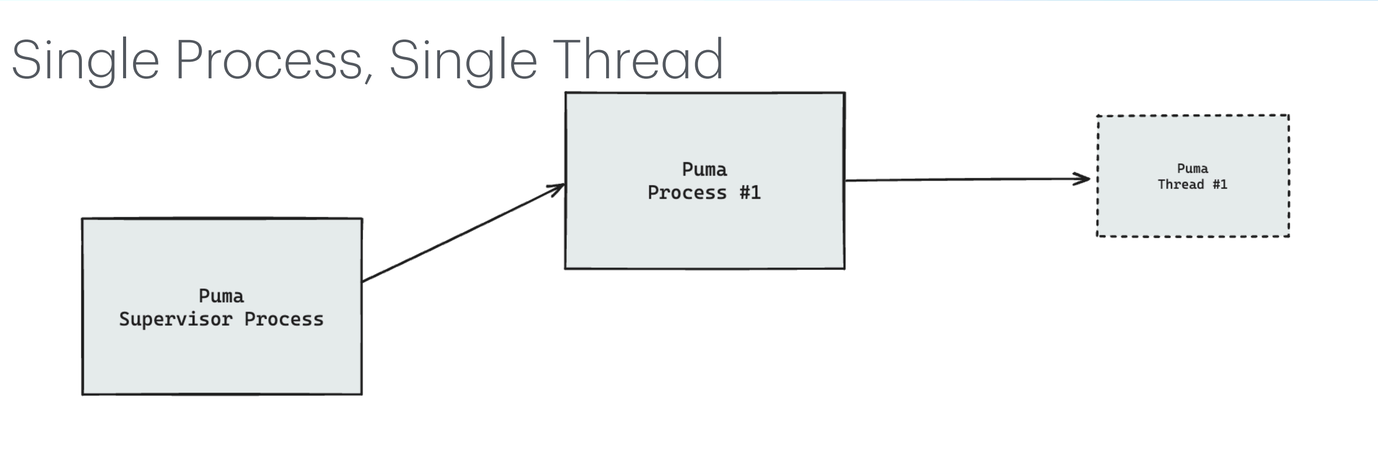
What modes can we configure

Single process, single thread

- Usually the worst option, you add the overhead of a supervisor process which will simply pass down signals.
- Puma actually warns you about this configuration mode!

```
alexandernicholson@Alexanders-MacBook-Air EURUKO Talk % WEB_CONCURRENCY=1 PUMA_MIN_THREADS=1 PUMA_MAX_THREADS=1 ruby app.rb
== Sinatra (v3.1.0) has taken the stage on 4567 for development with backup from Puma
[35771] Puma starting in cluster mode...
[35771] * Puma version: 5.6.5 (ruby 3.2.1-p31) ("Birdie's Version")
[35771] *  Min threads: 1
[35771] *  Max threads: 1
[35771] *  Environment: development
[35771] *   Master PID: 35771
[35771] *      Workers: 1
[35771] *     Restarts: (✔) hot (✔) phased
[35771] * Listening on http://127.0.0.1:4567
[35771] * Listening on http://[::1]:4567
[35771] Use Ctrl-C to stop
[35771] ! WARNING: Detected running cluster mode with 1 worker.
[35771] ! Running Puma in cluster mode with a single worker is often a misconfiguration.
[35771] ! Consider running Puma in single-mode (workers = 0) in order to reduce memory overhead.
[35771] ! Set the `silence_single_worker_warning` option to silence this warning message.
[35771] - Worker 0 (PID: 35772) booted in 0.0s, phase: 0
```

```
    \-+= 31724 alexandernicholson -zsh
      \-+= 35771 alexandernicholson puma 5.6.5 (tcp://localhost:4567) [EURUKO Talk]
        \--- 35772 alexandernicholson puma: cluster worker 0: 35771 [EURUKO Talk]
```




```
[35771] ! WARNING: Detected running cluster mode with 1 worker.
[35771] ! Running Puma in cluster mode with a single worker is often a misconfiguration.
[35771] ! Consider running Puma in single-mode (workers = 0) in order to reduce memory overhead.
[35771] ! Set the `silence_single_worker_warning` option to silence this warning message.
```

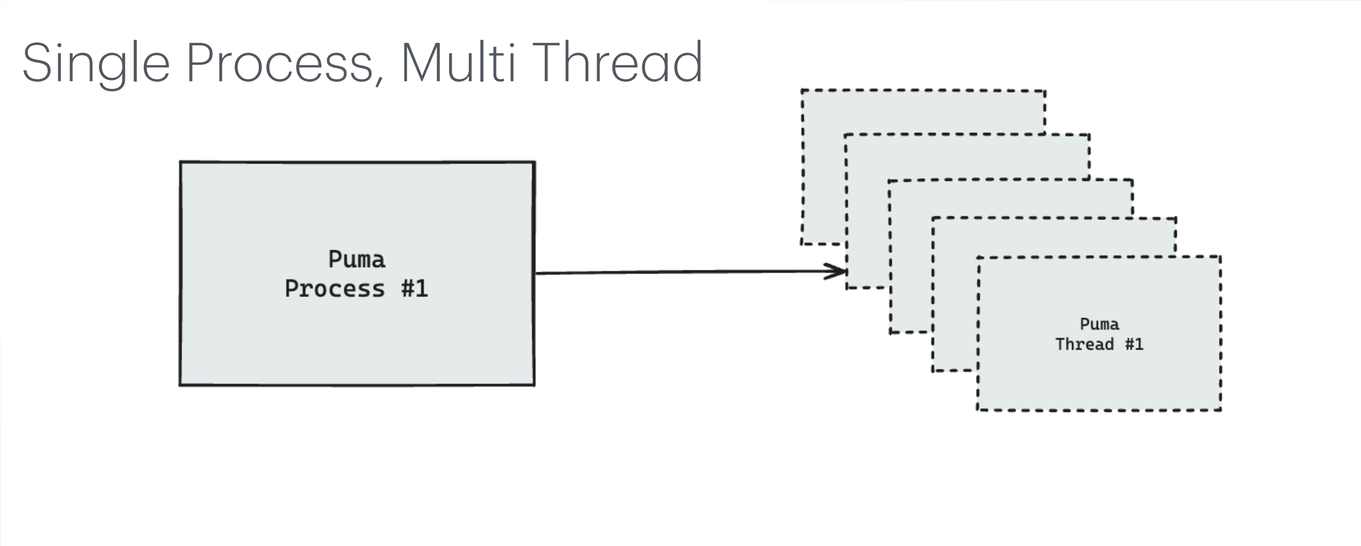
What modes can we configure

Single process, multi thread

```
== Sinatra (v3.1.0) has taken the stage on 4567 for development with backup from Puma
Puma starting in single mode...
* Puma version: 5.6.5 (ruby 3.2.1-p31) ("Birdie's Version")
*  Min threads: 0
*  Max threads: 5
*  Environment: development
*          PID: 32137
* Listening on http://127.0.0.1:4567
* Listening on http://[::1]:4567
```

```
 \-+= 31724 alexandernicholson -zsh
   \--= 32137 alexandernicholson puma 5.6.5 (tcp://localhost:4567) [EURUKO Talk]
```

- The default Puma setup starts in "single mode", which is a single process, multi-thread setup.
- Puma has auto-scaling built in, in the form of scaling the number of threads.
- Reactor (Worker / Puma Process) spawns threads, which have a lower context switching time than processes and can improve performance in IO intensive applications.



Why should I care?

The more threads you have, the more performance you have on IO bound applications!

What is IO?

- External API requests
- Disk
- Database request
- File uploads

What isn't IO?

- Compute-intensive tasks such as calculating Pi to a billion points, traversing matrices, etc.

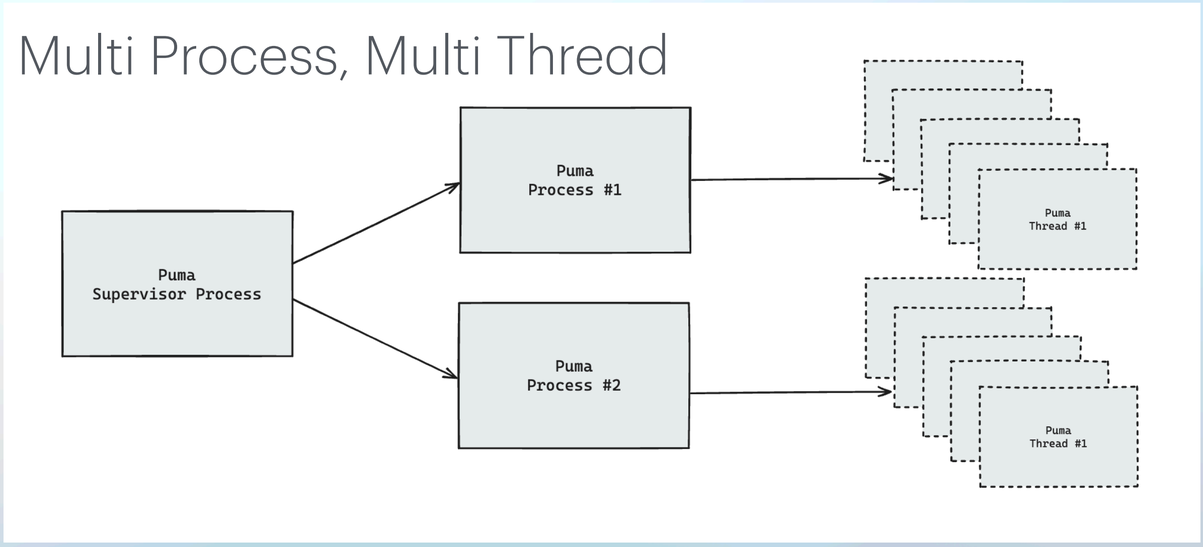
What modes can we configure

Multi process, multi thread

- This is called "Cluster Mode". It's the best way to run Puma from a performance and reliability perspective.
- Benefits of both multi-process and multi-threaded.
- Can better deal with slower application responses since multiple workers are pulling from the queue (incoming requests).
- A supervisor process handles incoming signals, there are lots of modes that are configurable on this supervisor to fine-tune how requests are distributed. More on that in the next slide.

```
alexandernicholson@Alexanders-MacBook-Air EURUKO Talk % WEB_CONCURRENCY=4 PUMA_MIN_THREADS=4 PUMA_MAX_THREADS=4 ruby app.rb
== Sinatra (v3.1.0) has taken the stage on 4567 for development with backup from Puma
[35970] Puma starting in cluster mode...
[35970] * Puma version: 5.6.5 (ruby 3.2.1-p31) ("Birdie's Version")
[35970] *  Min threads: 4
[35970] *  Max threads: 4
[35970] *  Environment: development
[35970] *   Master PID: 35970
[35970] *      Workers: 4
[35970] *     Restarts: (✔) hot (✖) phased
[35970] * Preloading application
[35970] * Listening on http://127.0.0.1:4567
[35970] * Listening on http://[::1]:4567
[35970] Use Ctrl-C to stop
[35970] - Worker 0 (PID: 35971) booted in 0.0s, phase: 0
[35970] - Worker 1 (PID: 35972) booted in 0.0s, phase: 0
[35970] - Worker 2 (PID: 35973) booted in 0.0s, phase: 0
[35970] - Worker 3 (PID: 35974) booted in 0.0s, phase: 0
```

```
    \-+= 31724 alexandernicholson -zsh
      \-+= 35970 alexandernicholson puma 5.6.5 (tcp://localhost:4567) [EURUKO Talk]
        |--- 35971 alexandernicholson puma: cluster worker 0: 35970 [EURUKO Talk]
        |--- 35972 alexandernicholson puma: cluster worker 1: 35970 [EURUKO Talk]
        |--- 35973 alexandernicholson puma: cluster worker 2: 35970 [EURUKO Talk]
        \--- 35974 alexandernicholson puma: cluster worker 3: 35970 [EURUKO Talk]
```



What modes can we configure

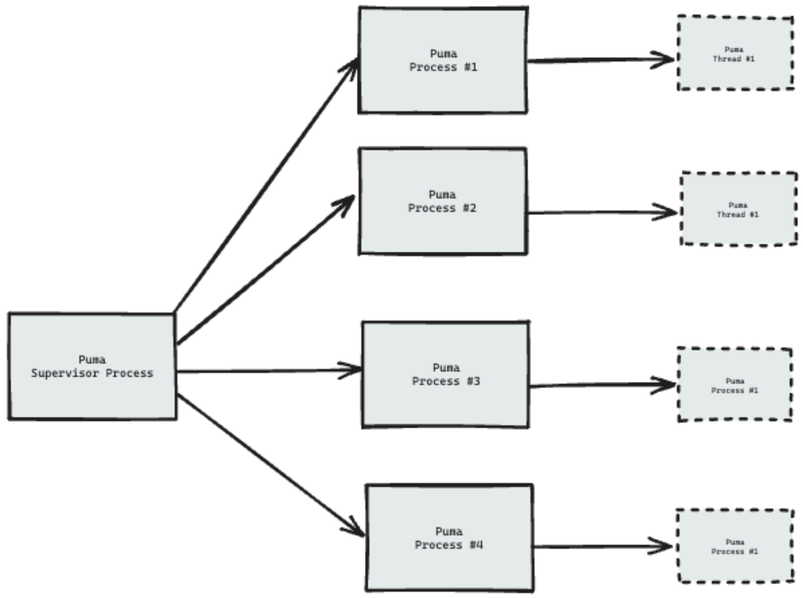
Multi process, single thread

```
alexandernicholson@Alexanders-MacBook-Air EURUKO Talk % WEB_CONCURRENCY=4 PUMA_MIN_THREADS=1 PUMA_MAX_THREADS=1 RACK_ENV=production ruby app.rb
== Sinatra (v3.1.0) has taken the stage on 4567 for production with backup from Puma
[39016] Puma starting in cluster mode...
[39016] * Puma version: 5.6.5 (ruby 3.2.1-p31) ("Birdie's Version")
[39016] *  Min threads: 1
[39016] *  Max threads: 1
[39016] *  Environment: production
[39016] *   Master PID: 39016
[39016] *      Workers: 4
[39016] *     Restarts: (✔) hot (✖) phased
[39016] * Preloading application
[39016] * Listening on http://0.0.0.0:4567
[39016] Use Ctrl-C to stop
[39016] - Worker 0 (PID: 39017) booted in 0.0s, phase: 0
[39016] - Worker 1 (PID: 39018) booted in 0.0s, phase: 0
[39016] - Worker 2 (PID: 39019) booted in 0.0s, phase: 0
[39016] - Worker 3 (PID: 39020) booted in 0.0s, phase: 0
```

- This is also called "Cluster Mode".
- If your application isn't thread safe, but you want the benefits of running multiple processes, you can do so by setting max threads to 1 and setting web_concurrency to as many vCPUs as you have.
- You can also use this mode when your application is not spending much time on IO operations.
- You still get the benefit of sending signals to your Puma workers.
  - TTIN increment the worker count by 1
  - TTOU decrement the worker count by 1
  - TERM send TERM to worker. The worker will attempt to finish then exit.

https://shopify.engineering/ruby-execution-models

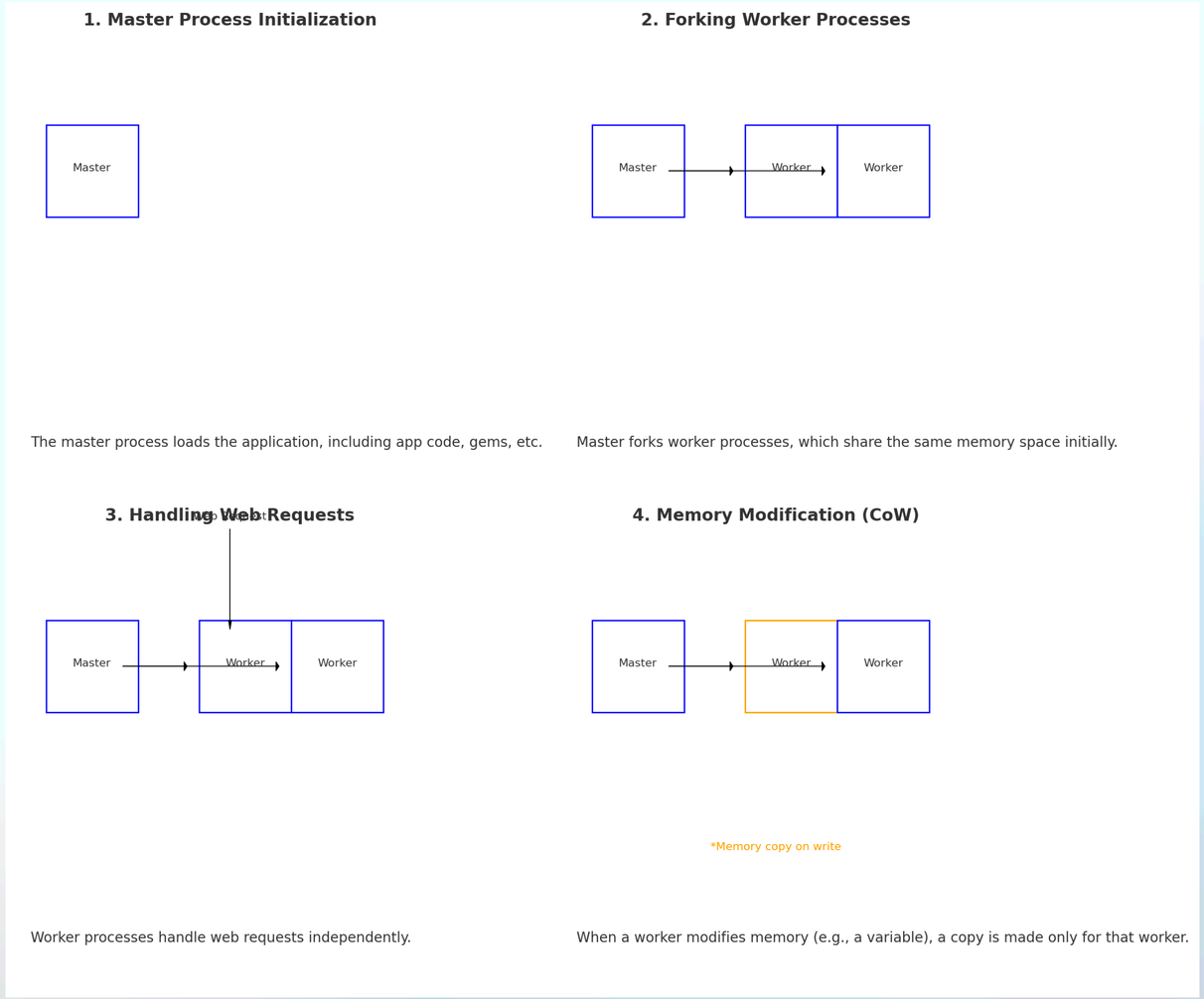
Multi Process, Single Thread



Why should I care?

What is CoW?

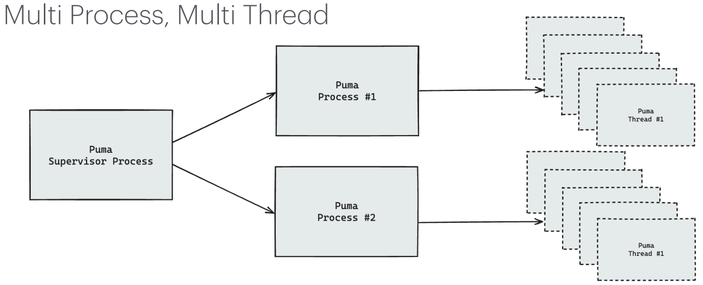
- Puma supports Copy on Write, a technique that allows you to create additional workers for "free" (in terms of memory usage).
- Puma pre-loads your application before forking workers from the supervisory process.
  - preload_app!



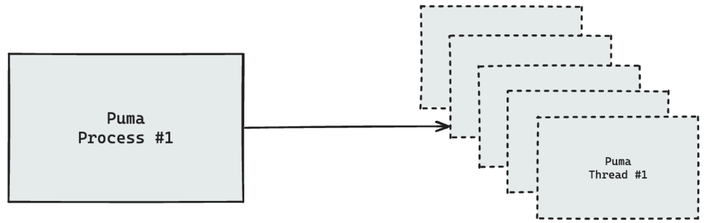
Why should I care?

Comparing each method

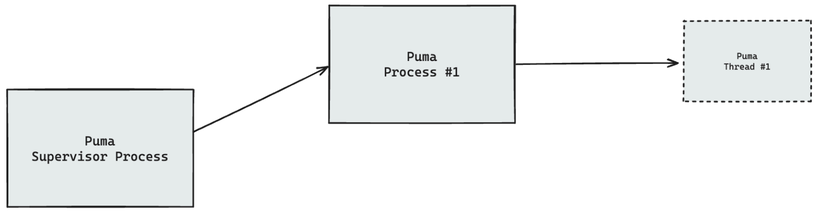
Multi Process, Multi Thread



Single Process, Multi Thread



Single Process, Single Thread



CoW	Worker-level fault-tolerance / redundancy	Accepts SIGnals	Horizontally Scalable	Not vulnerable to slow-clients
✓	✓	✓	✓	✓
✗	✗	— Cannot change process count	— Just threads	✗
✗	✗	— I mean yes.. but no!	✗	✗

What modes can we configure

Multiple Processes, Zero Threads

```
alexandernicholson@Alexanders-MacBook-Air EURUKO Talk % WEB_CONCURRENCY=4 PUMA_MAX_THREADS=0 ruby app.rb
== Sinatra (v3.1.0) has taken the stage on 4567 for development with backup from Puma
[34903] Puma starting in cluster mode...
[34903] * Puma version: 5.6.5 (ruby 3.2.1-p31) ("Birdie's Version")
[34903] *  Min threads: 0
[34903] *  Max threads: 0
[34903] *  Environment: development
[34903] *   Master PID: 34903
[34903] *      Workers: 4
[34903] *     Restarts: (✔) hot (✖) phased
[34903] * Preloading application
[34903] * Listening on http://127.0.0.1:4567
[34903] * Listening on http://[::1]:4567
[34903] Use Ctrl-C to stop
[34903] - Worker 0 (PID: 34904) booted in 0.0s, phase: 0
[34903] - Worker 1 (PID: 34905) booted in 0.0s, phase: 0
[34903] - Worker 2 (PID: 34906) booted in 0.0s, phase: 0
[34903] - Worker 3 (PID: 34907) booted in 0.0s, phase: 0
```

There's nobody home, but the lights are on. All requests fail!

```
^C[34903] - Gracefully shutting down workers...
^C^C^C^C
```

What will we talk about?

Agenda

- Why should I care about scaling Puma?
- **The dangers of following YouTube advice.**
- What should I do?
- Q&A

The dangers of following YouTube advice.

"Containers must have one process."

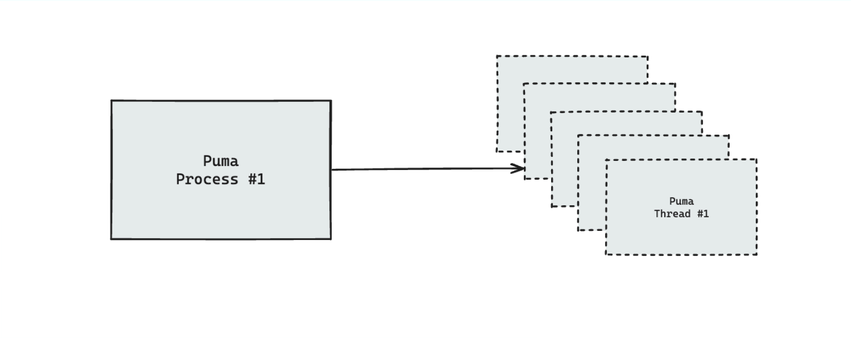
> "Containers must have only one process, this is a rule I saw on a YouTube talk."

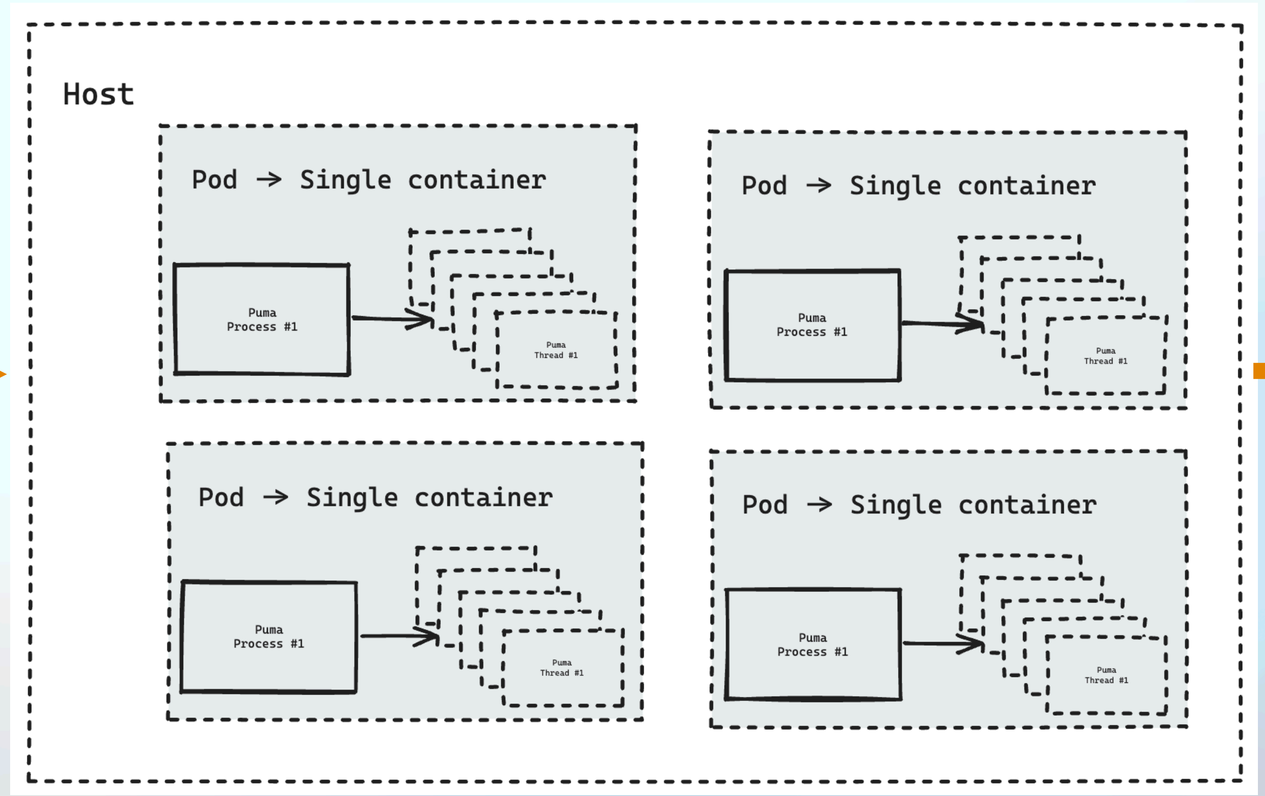
- *Senior Infrastructure Engineer*

The dangers of following YouTube advice.

Ok, let's visualise a single process per container.

"Single mode"

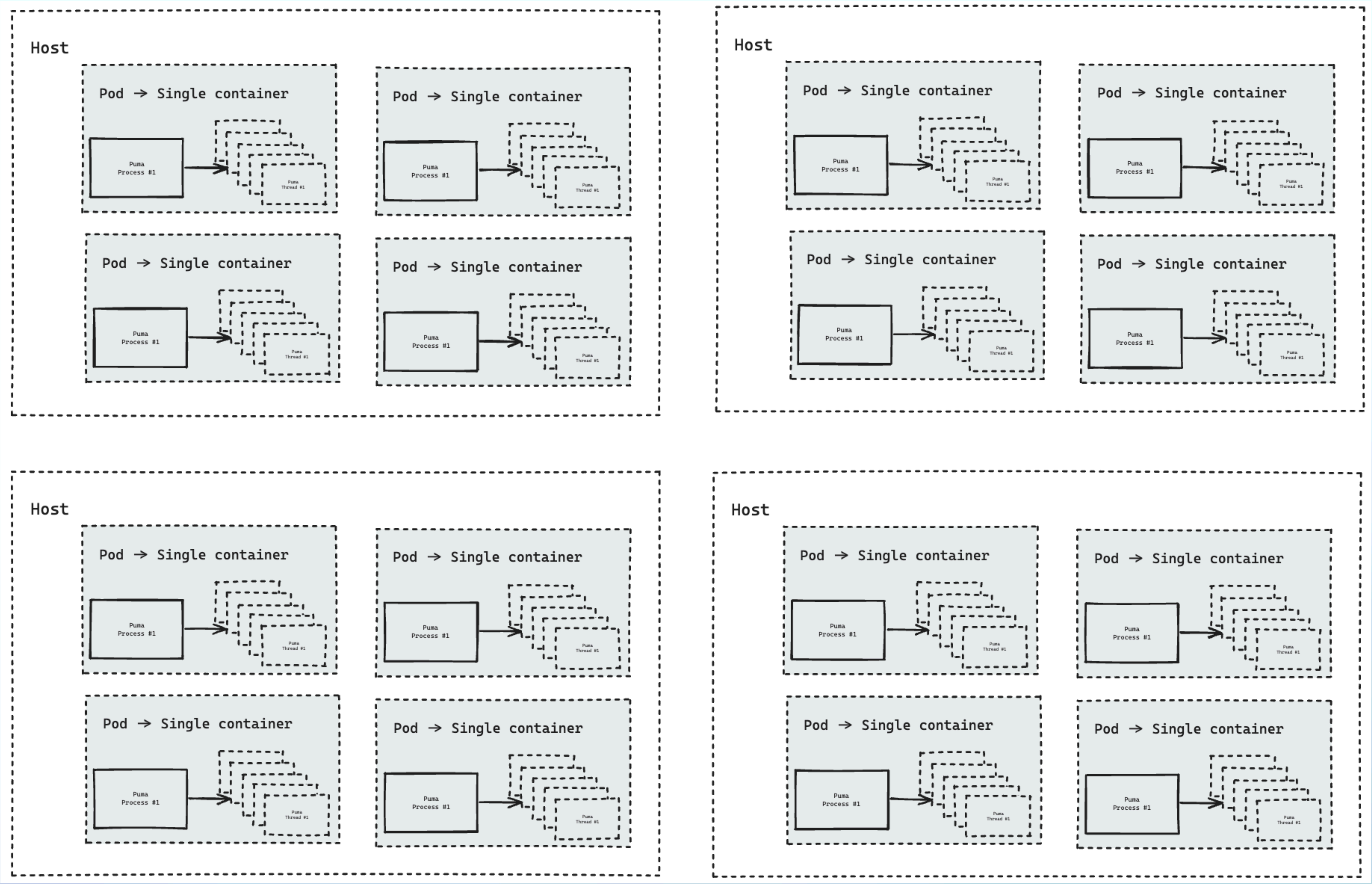




- Each pod has a single puma process (aka "worker).
- We allocate 1 vCPU, 5GB memory for each pod.

The dangers of following YouTube advice.

Let's scale it a bit...



The dangers of following YouTube advice.

Some maths

- 4 hosts
- 16 pods
- 16 Puma processes
- 16 * 5 = 80 Puma threads
- 16 vCPUs
- 20GB memory (and we have less headroom because we did not benefit from CoW for our Puma workers)

(Number of queues / workers pulling from queue)

Queue Time

16/16 = 1!

The dangers of following YouTube advice.

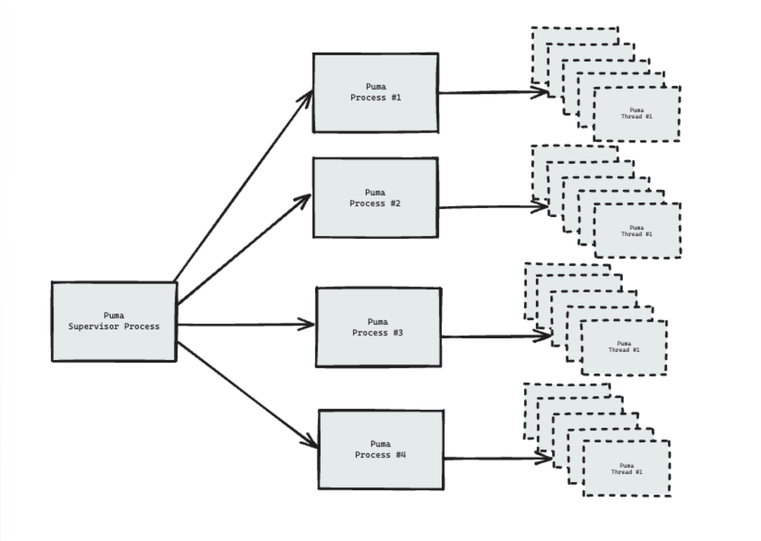
Downsides

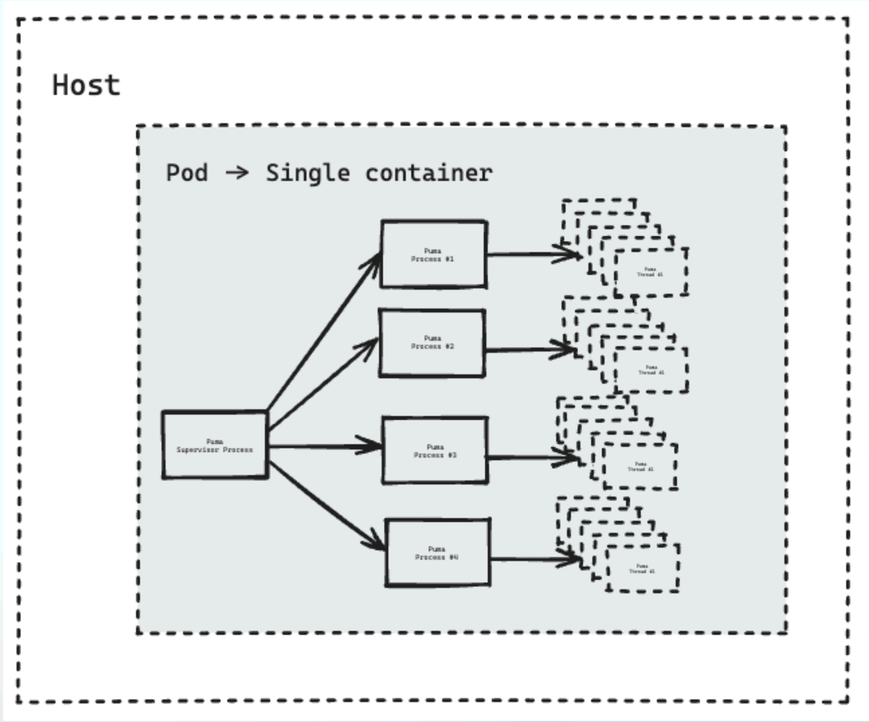
- No built-in load balancer, so you have to add an AWS Application Load Balancer and have it do round-robin to your application instances.
    - Increased latency due to some hosts being busier than others (unless all of your application endpoints take around the same time to generate a reply).
- Potentially lower performance since each pod only has access to a limited amount of resources.
- You are creating [x * pods] request queues, which your load balancer cannot see or adjust for.
- You lose out on CoW! Increased cost since you are wasting resources.
- Slower clients can block entire pods.

The dangers of following YouTube advice.

Let's do this properly.

"Cluster mode"

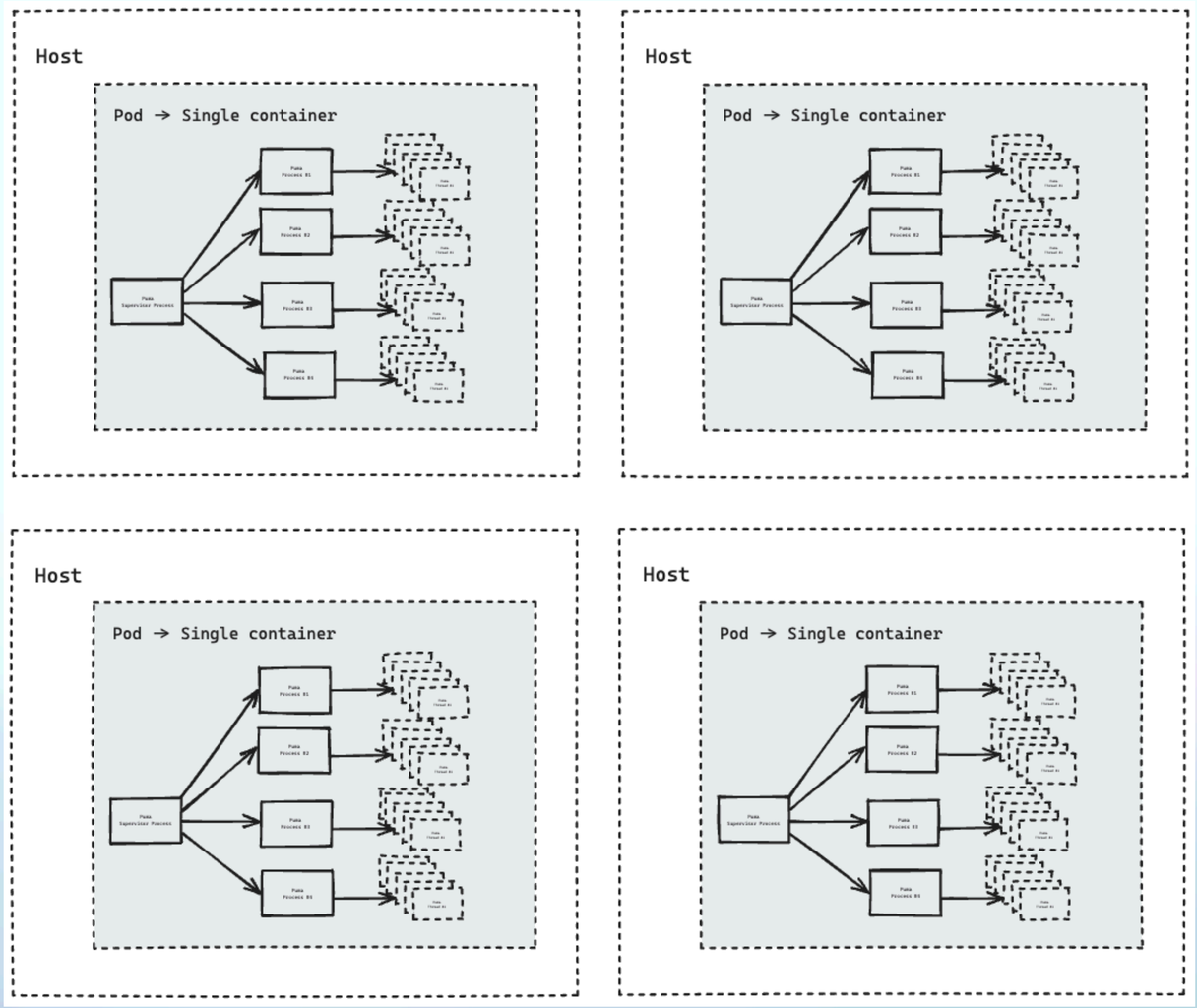




- Each pod has multiple puma processes (aka "worker).
- We allocate 4vCPU, 20GB memory for each pod.

The dangers of following YouTube advice.

Let's scale it a bit...



The dangers of following YouTube advice.

Some maths

- 4 hosts
- 4 pods
- 16 Puma processes
- 16 * 5 = 80 Puma threads
- 16 vCPUs
- 20GB memory (but we have CoW and can potentially fit some more workers)

(Number of queues / workers pulling from queue)

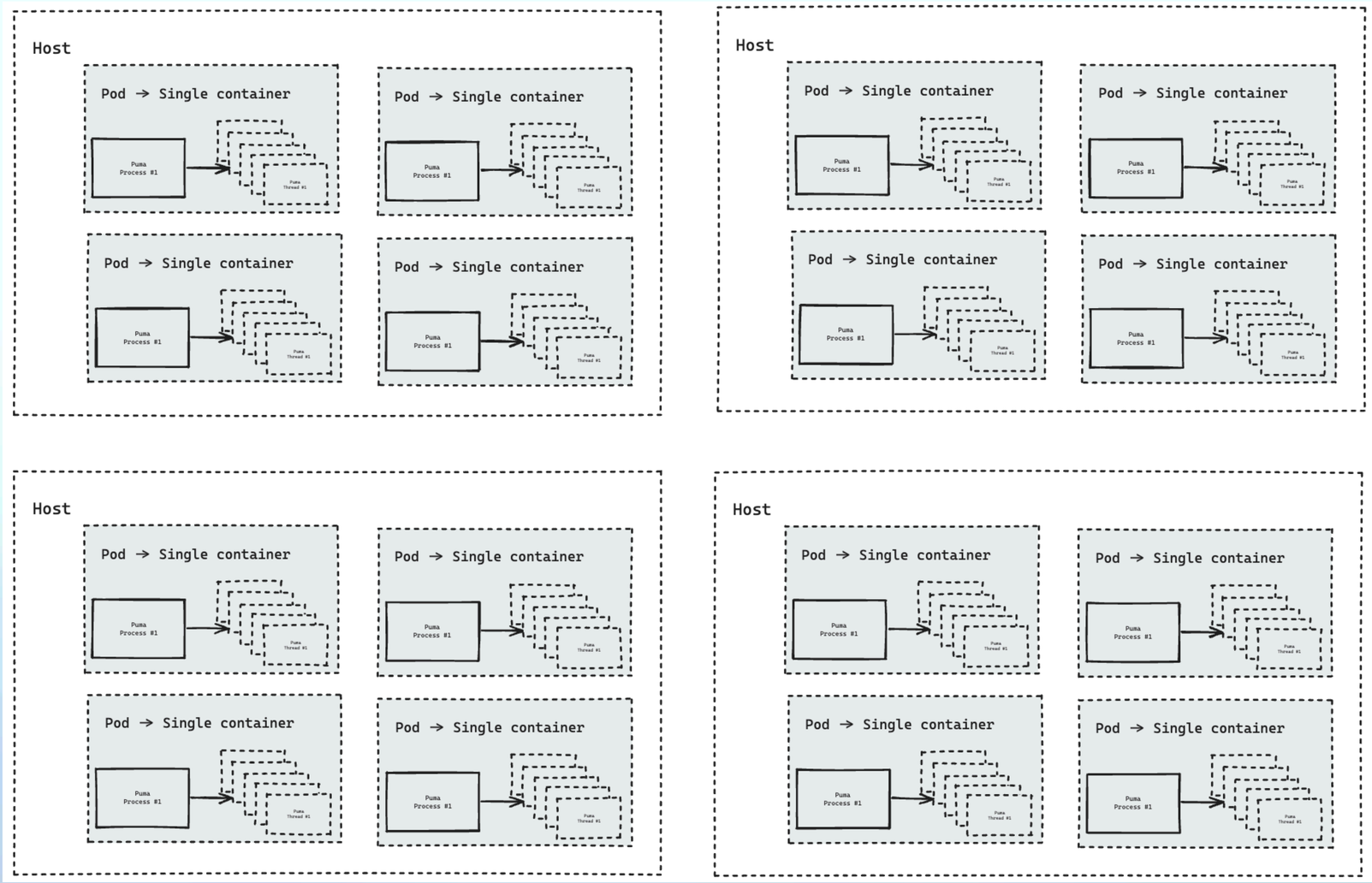
Queue Time

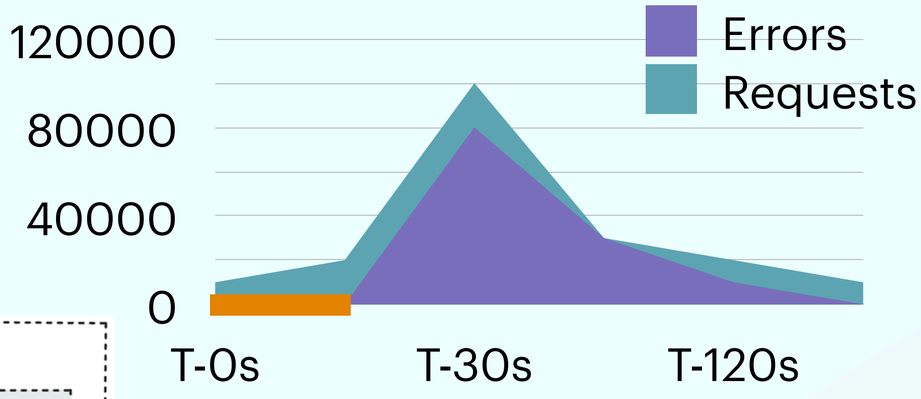
4/16 = 1/4

4x faster!

Simulation of a traffic spike

Worker exhaustion - Single Mode

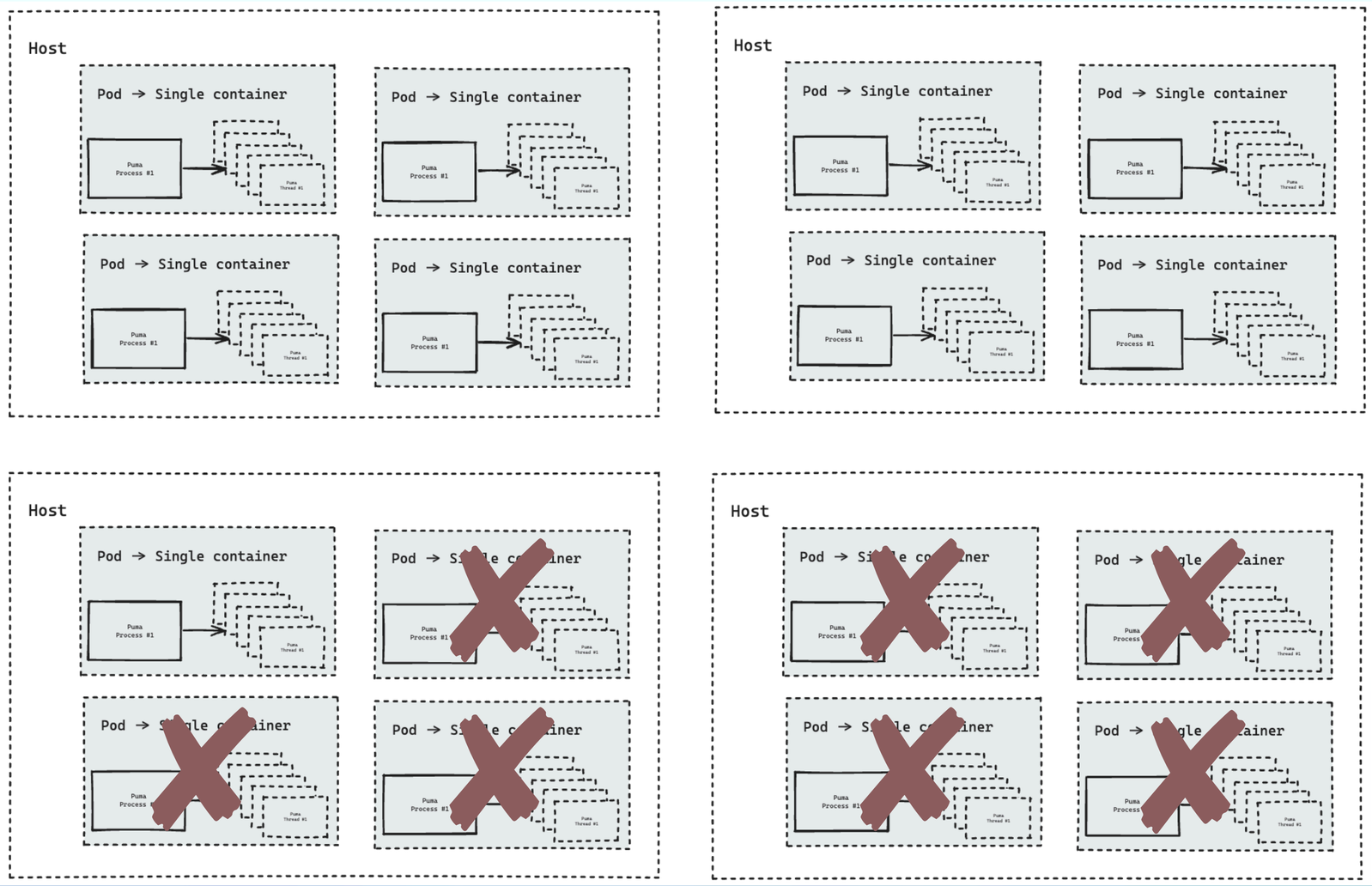


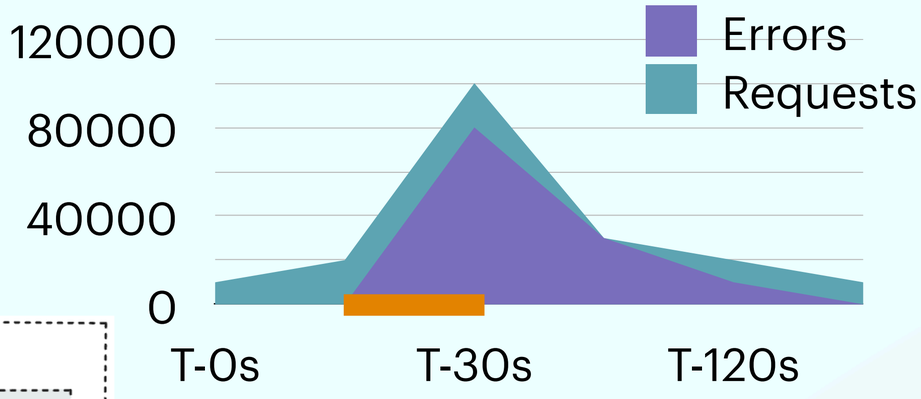


- Everything is fine, our app is working for now.
- But there is a promotion about to start...

Simulation of a traffic spike

Worker exhaustion - Single Mode

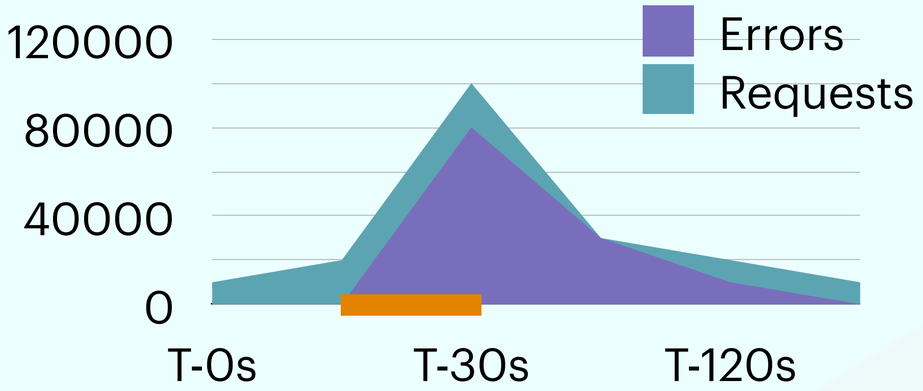


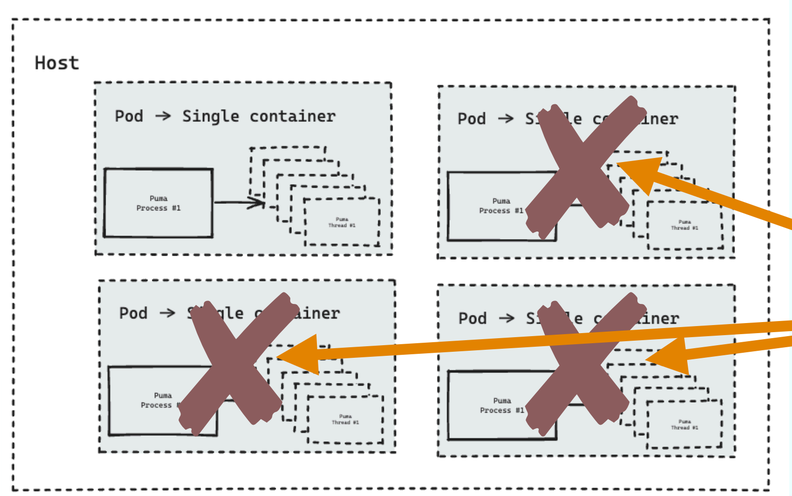


- Suddenly, a large spike of traffic comes in, and around half of all pods are fully saturated (remember they only have limited resources per pod).
- **Some pods may die!**
- The load balancer cannot see that the pod is busy, and keeps sending requests to them.
- Slower requests and clients increase queue processing times.

Simulation of a traffic spike

Worker exhaustion - Single Mode






```
* Listening on http://0.0.0.0:4567
Use Ctrl-C to stop
zsh: killed     WEB_CONCURRENCY=0 PUMA_MIN_THREADS=5 PUMA_MAX_THREADS=5 RACK_ENV=production
```

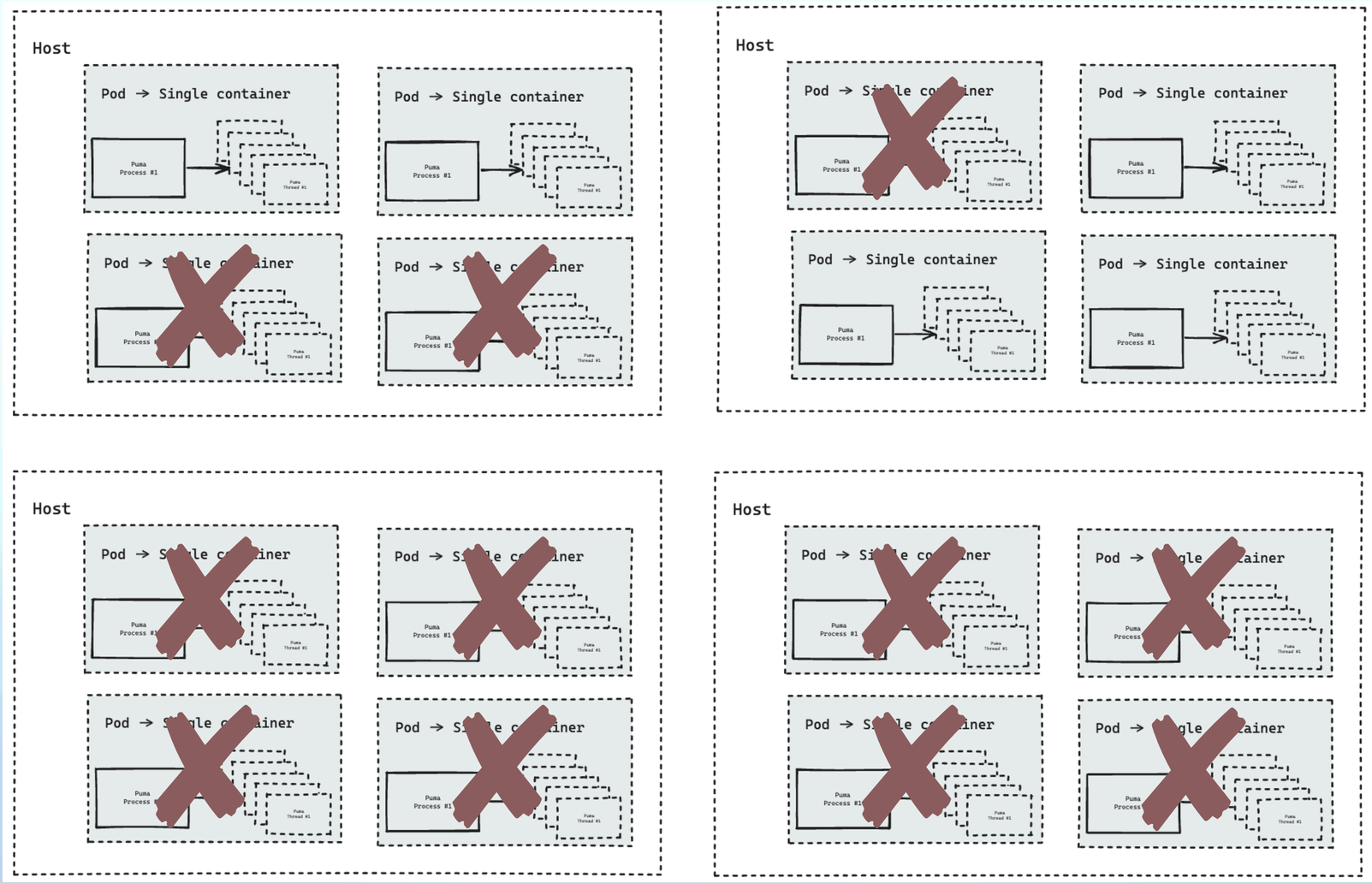
Our app dies, and our single Puma worker exits.

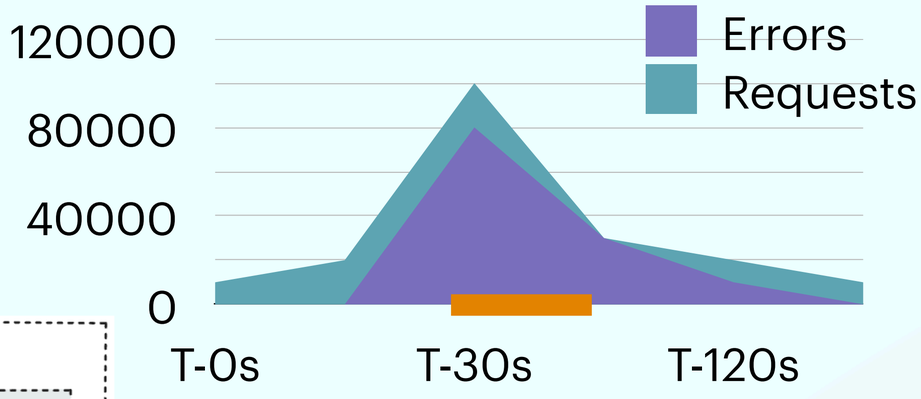
```
alexandernicholson@Alexanders-MacBook-Air EURUKO Talk % WEB_CONCURRENCY=0 PUMA_MIN_THREADS=5 PUMA_MAX_THREADS=5 RACK_ENV=production ruby app.rb
== Sinatra (v3.1.0) has taken the stage on 4567 for production with backup from Puma
Puma starting in single mode...
* Puma version: 5.6.5 (ruby 3.2.1-p31) ("Birdie's Version")
*  Min threads: 5
*  Max threads: 5
*  Environment: production
*          PID: 39146
* Listening on http://0.0.0.0:4567
Use Ctrl-C to stop
zsh: killed     WEB_CONCURRENCY=0 PUMA_MIN_THREADS=5 PUMA_MAX_THREADS=5 RACK_ENV=production
```

The Pod is gone and must be recreated by the scheduler, incurring backoff, image pull times, and scheduling delays - meaning our outage lasts longer!

Simulation of a traffic spike

Worker exhaustion - Single Mode

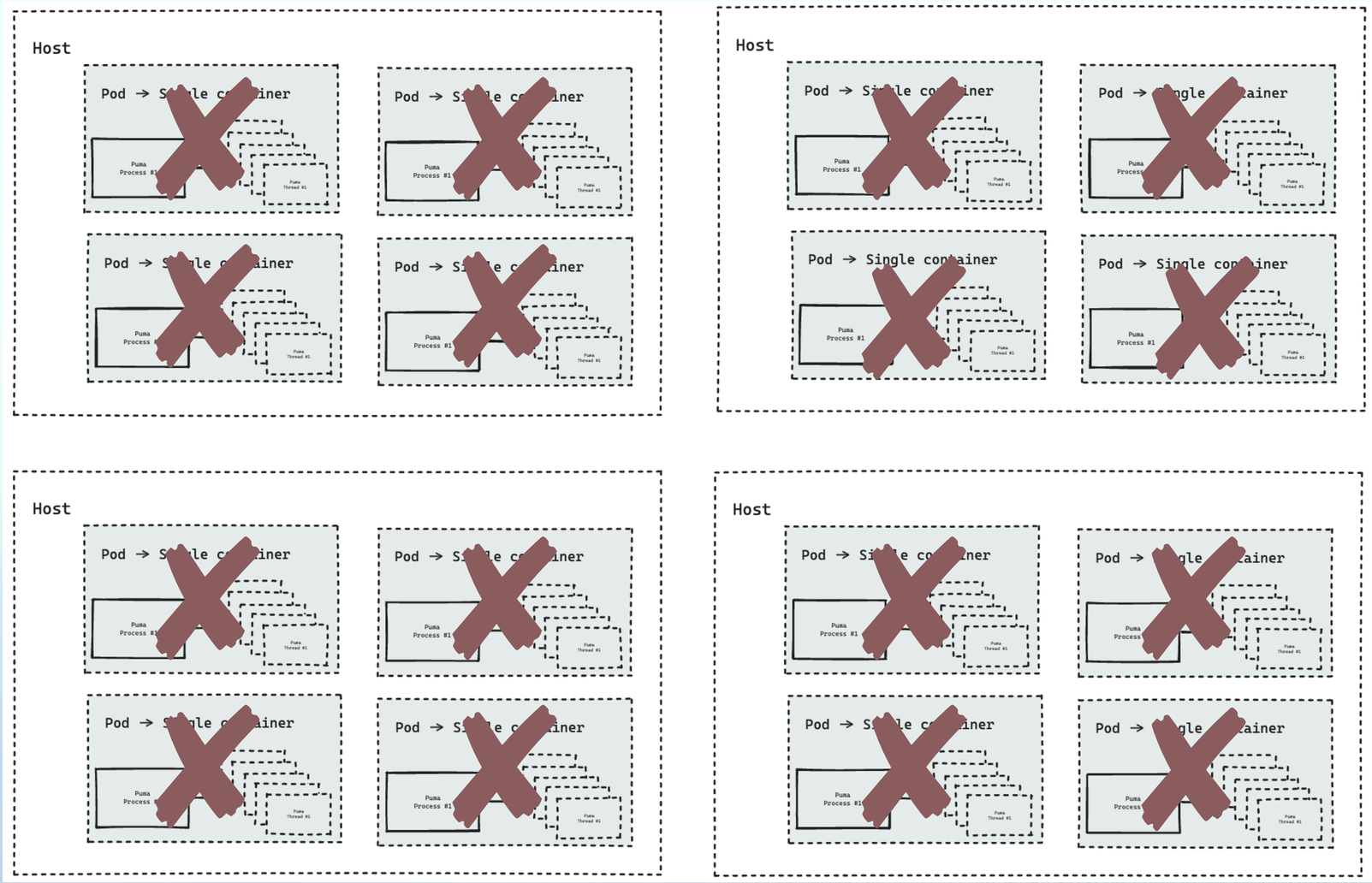


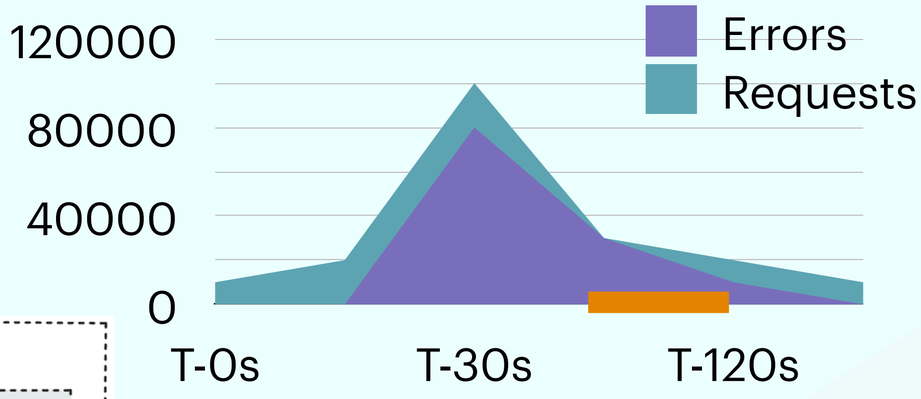


- A large number of requests are getting timeout errors due to queue saturation on our workers. Some workers are dying due to OOM or being killed by Kubernetes.
- Health checks fail on more pods, and more than half the fleet is marked as out of service.
- More traffic will be sent to the remaining pods...

Simulation of a traffic spike

Worker exhaustion - Single Mode

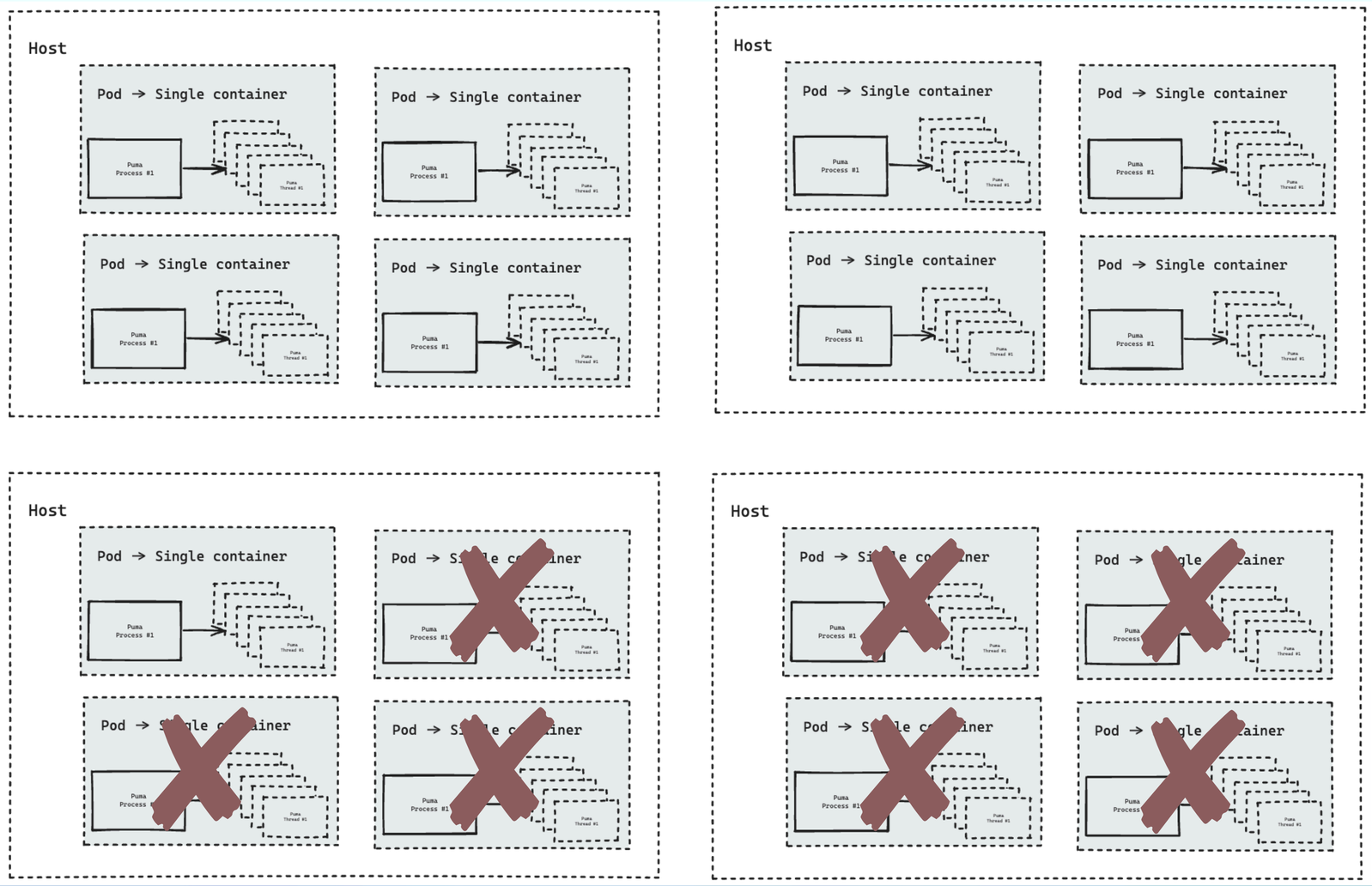


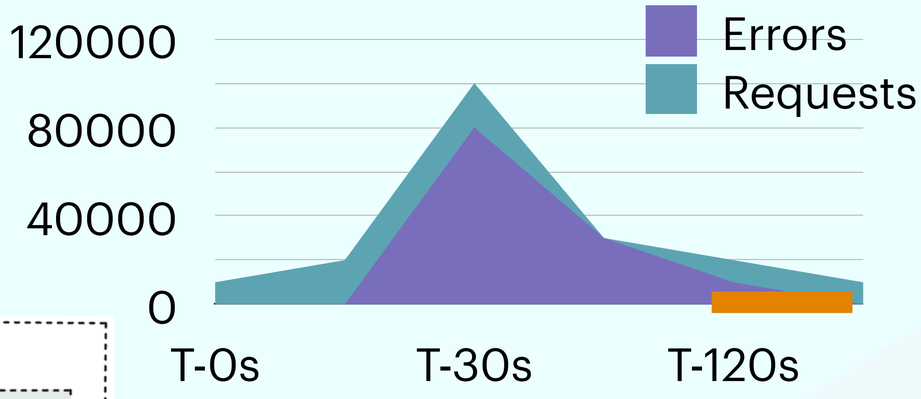


- **Cascading failure!**
- 100% of our pods are marked as out of service since health checks have timed out.
- All requests time out, even though the traffic spike has passed, since all workers are locked by client requests that are already gone.

Simulation of a traffic spike

Worker exhaustion - Single Mode

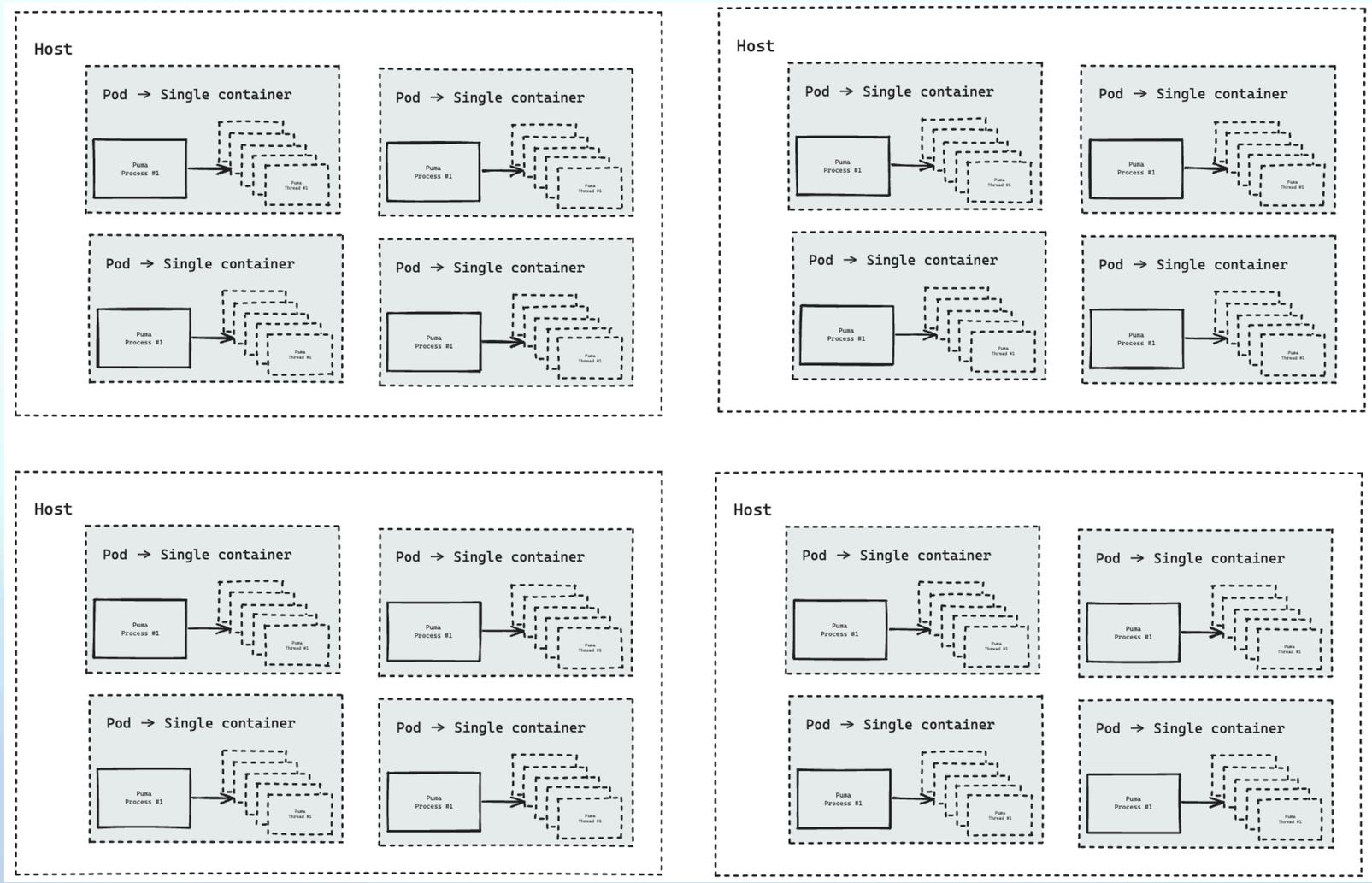


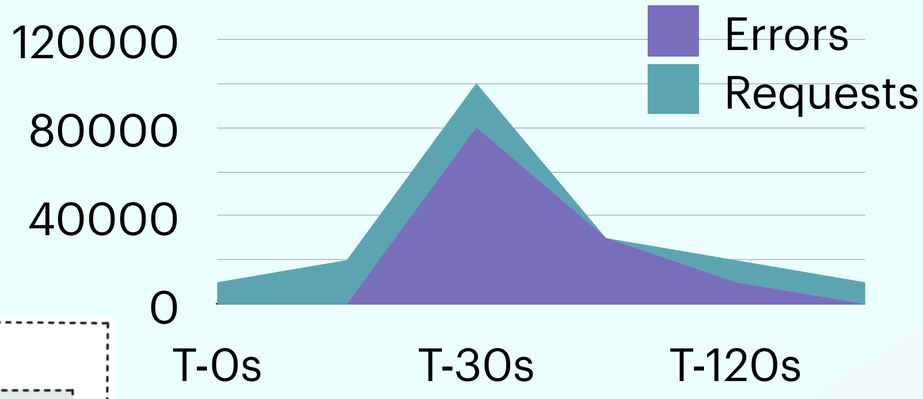


- Pods begin to be put back into service.
- As a result of being taken out of service, most of the requests in their queues have failed, and health checks are now allowed to pass.
- We can see a repeat of the cascading failure in the previous slide as many times as there are spikes in traffic since our capacity is lowered.

Simulation of a traffic spike

Worker exhaustion - Single Mode

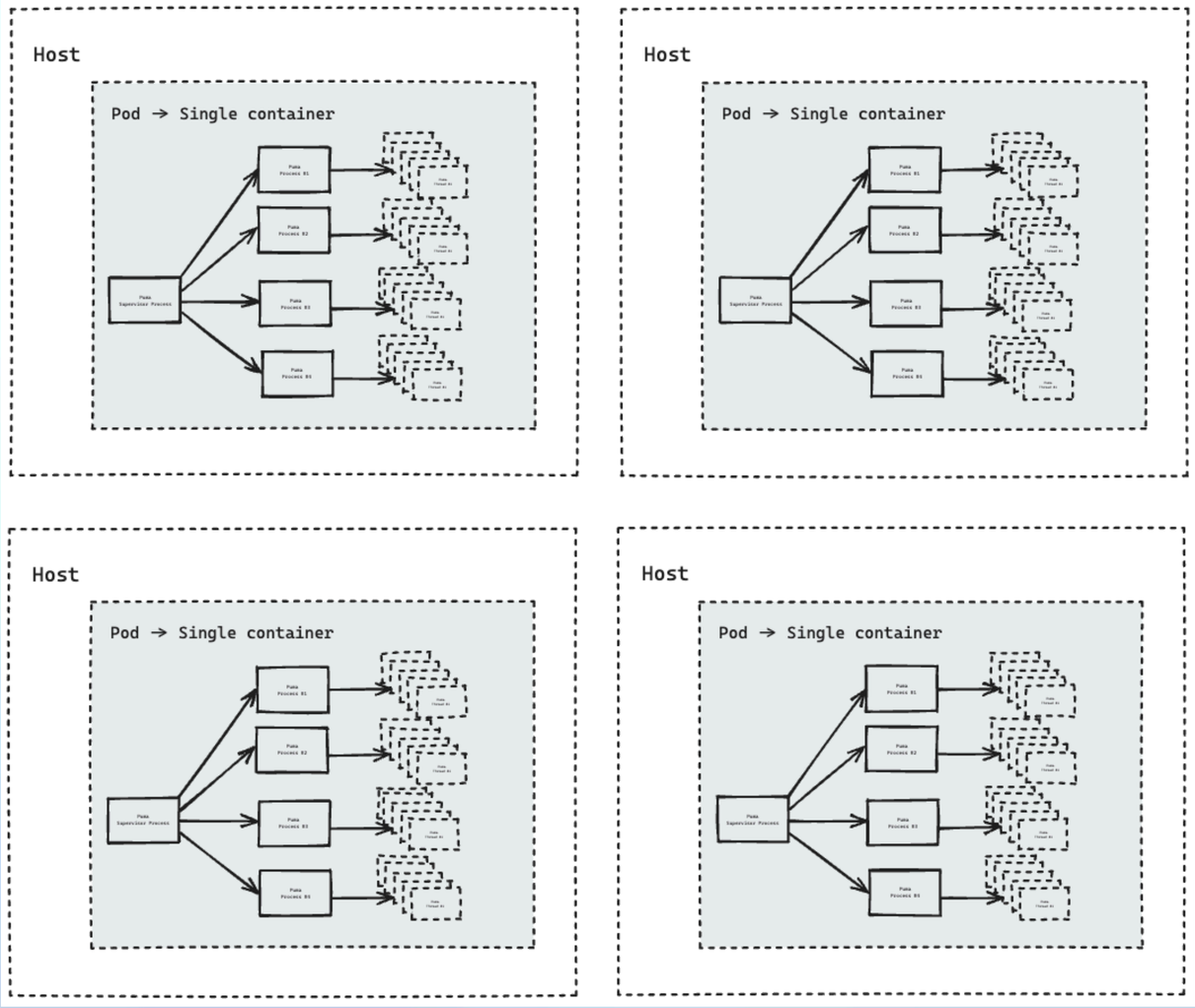


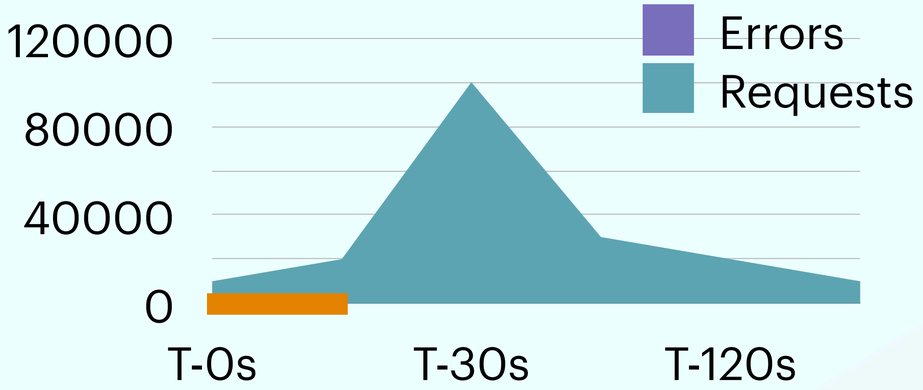


- Back to normal.
- We dropped most of our requests.
- People were not able to book!
- We've lost customer trust.
- **Our promotion was a failure!**

Simulation of a traffic spike

Cluster - Trusting Puma to do the right thing.

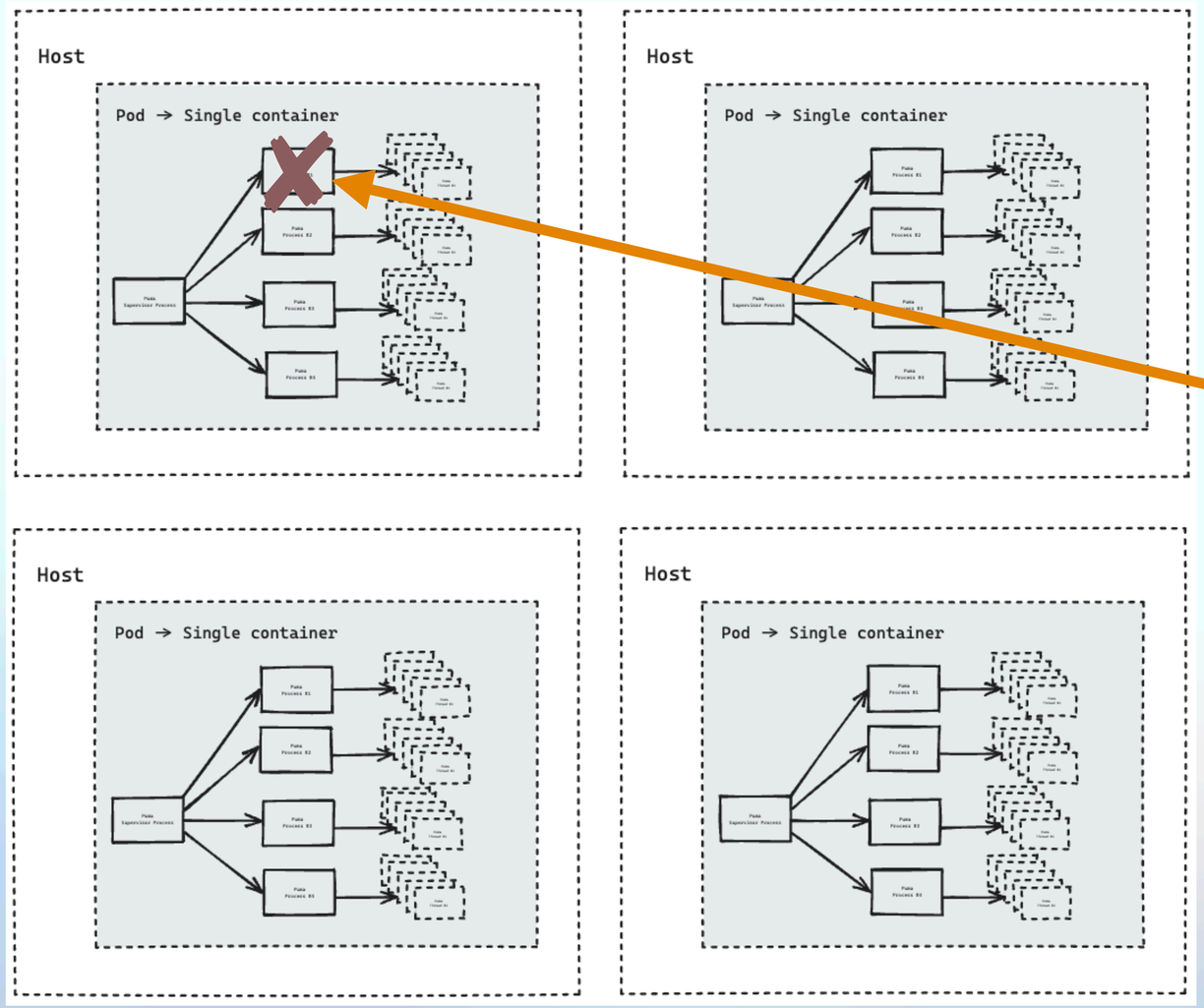


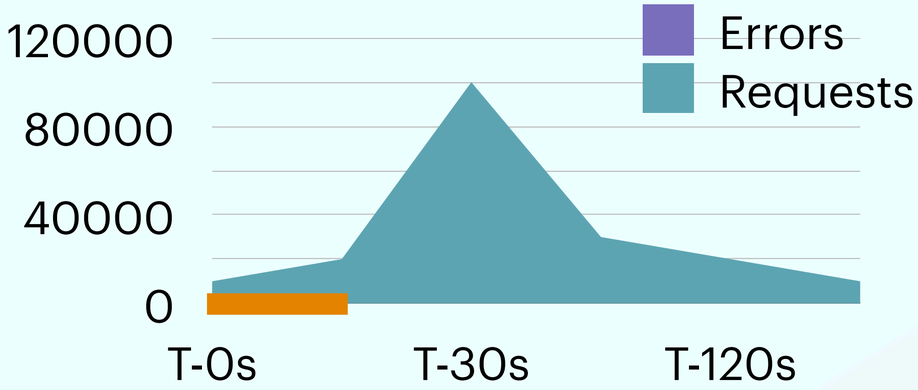


- There are 4 queues, and requests will be round-robin distributed to these queues by our load balancer.
- Puma in cluster mode with wait_for_less_busy_worker enabled (default in Puma 6) reduces latency for high-utilisation servers by picking processes that are less busy.

Simulation of a traffic spike

Cluster - Trusting Puma to do the right thing.





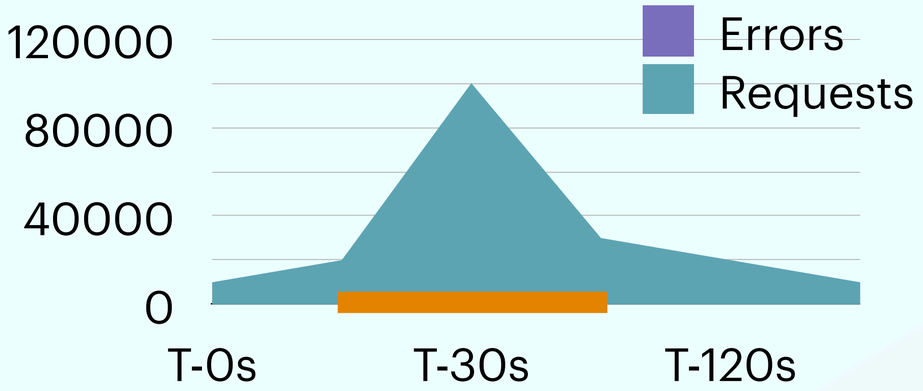
Worker automatically restarted!

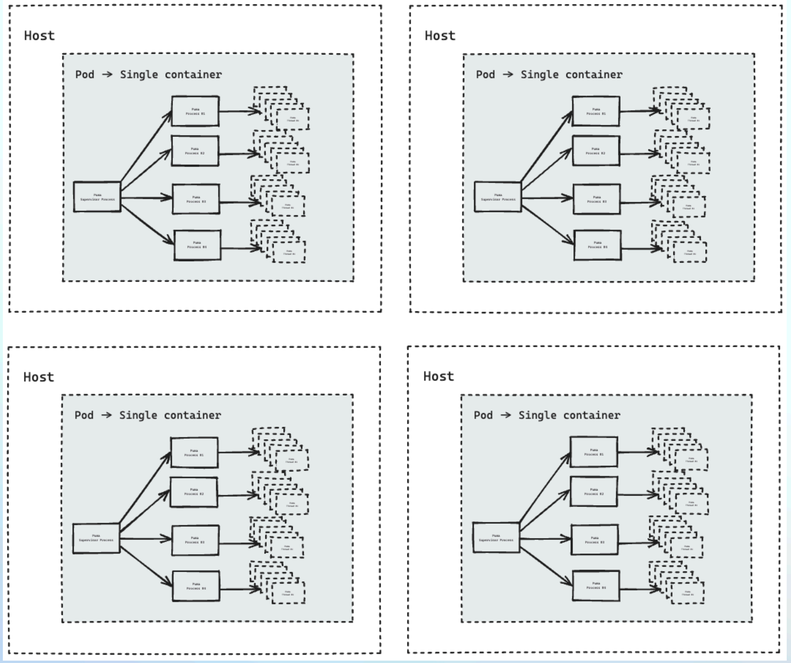
```
[39016] * Listening on http://0.0.0.0:4567
[39016] Use Ctrl-C to stop
[39016] - Worker 0 (PID: 39017) booted in 0.0s, phase: 0
[39016] - Worker 1 (PID: 39018) booted in 0.0s, phase: 0
[39016] - Worker 2 (PID: 39019) booted in 0.0s, phase: 0
[39016] - Worker 3 (PID: 39020) booted in 0.0s, phase: 0
== Sinatra has ended his set (crowd applauds)
[39016] - Worker 2 (PID: 39098) booted in 0.0s, phase: 0
```

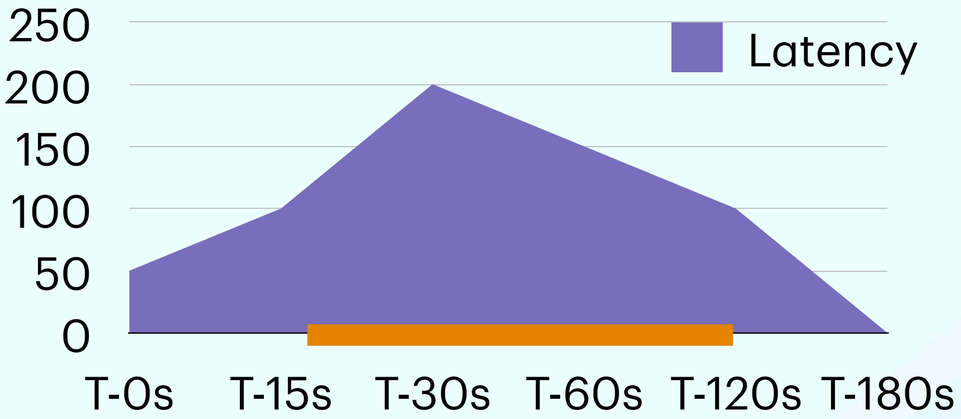
Pod does not die!!!

Simulation of a traffic spike

Cluster - Trusting Puma to do the right thing.



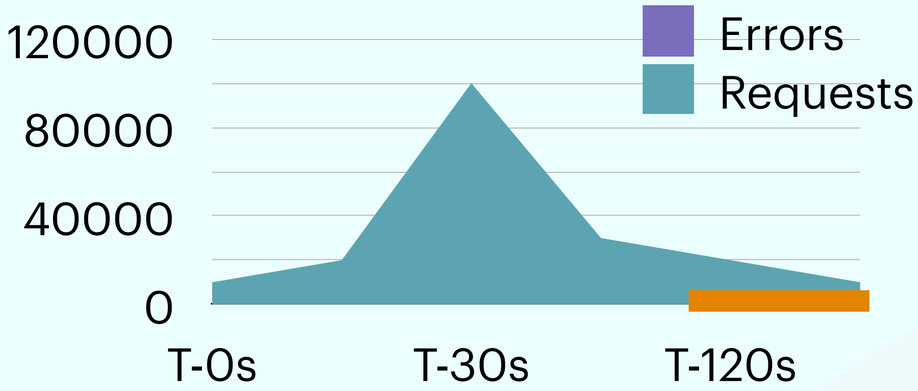




- Unlike our previous promotion, we are able to sustain traffic since we have an actual request queue (Puma's supervisor process).
- The primary metric that suffers is latency, as our process / thread configuration starts to saturate due to incoming requests.

Simulation of a traffic spike

Cluster - Trusting Puma to do the right thing.



- Back to normal.
- All of our requests went through, some were slow, but not in a way that would give customer's browsers timeouts.
- People were able to book!
- **Our promotion was a success!**

What will we talk about?

Agenda

- Why should I care about scaling Puma?
- The dangers of following YouTube advice.
- **What should I do?**
- Q&A

One-slide Puma Scaling

WEB_CONCURRENCY

- Controls the number of processes.
- **Set to the number of cores (or vCPUs).**
- **Do not set to 1.**

preload_app!

- Reduces startup time for each worker process (what you have set WEB_CONCURRENCY to).
- **This is enabled by default** when WEB_CONCURRENCY is set.

PUMA_MIN_THREADS, PUMA_MAX_THREADS

- Controls Puma's thread pool.
- **Set to the same value**.
- **Start with 5.**

Vertical scaling before horizontal scaling.

- Puma's cluster mode is the best request orchestrator.
- Move from an m6i.xlarge to m6i.2xlarge instance (double concurrency + threads) to scale up.

What will we talk about?

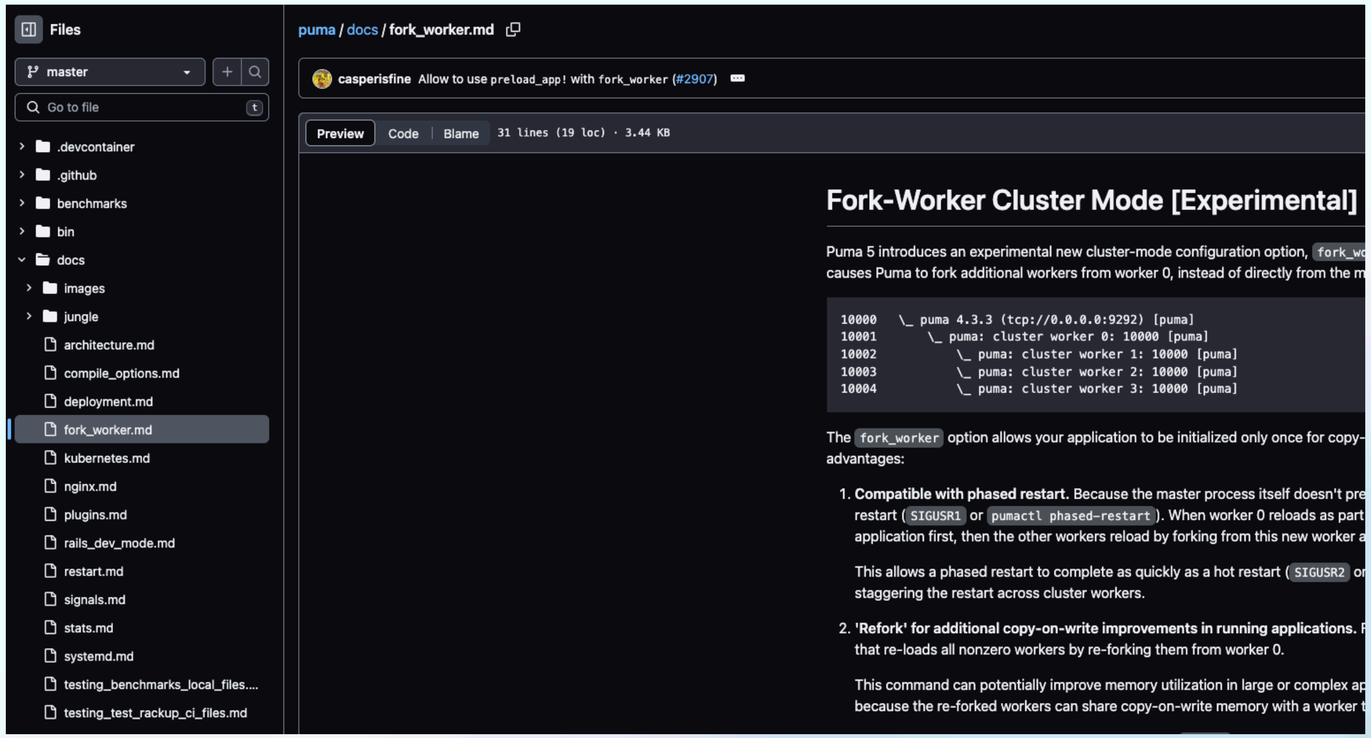
Agenda

- Why should I care about scaling Puma?
- The dangers of following YouTube advice.
- What should I do?
- **Bonus: Further optimisations**
- Q&A

Bonus: Further optimisations

Some extra reading

- Fork-Worker Cluster Mode
  - Allows for phased restarts, which can create cool workflow opportunities!
  - Additional CoW improvements, for lower memory usage. Binpacking is super important at scale! It's basically free money.
  - Has the downside of creating the potential for outages when using Kubernetes since health endpoints might not respond during the restart of worker 0.

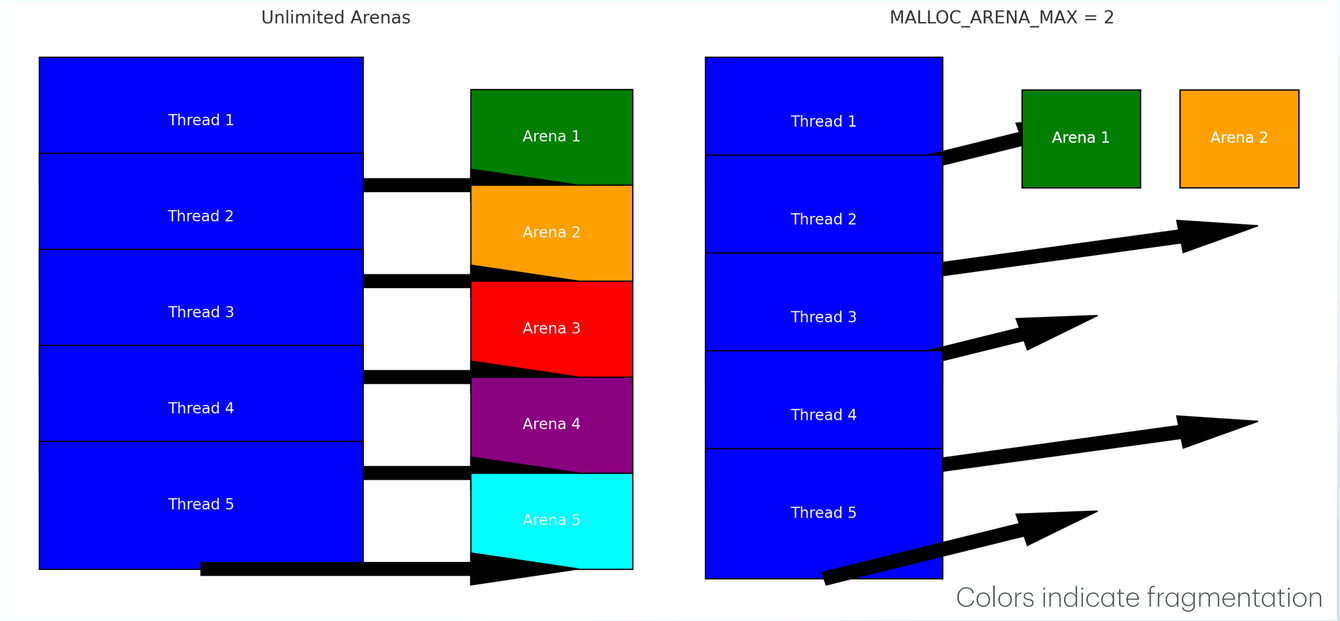


Enable prune_bundler and disable preload_app!

Bonus: Further optimisations

Some extra reading

- Memory Arenas (MALLOC_ARENA_MAX)
  - **Set to 2 initially (if it's not set by default - Heroku does this).**
  - **Perform performance testing (such as monitoring latency) with setting this environment variable to 2, 1, and not set.**
  - Has the potential to reduce the memory usage of your application at the expense of latency.



- https://devcenter.heroku.com/articles/tuning-glibc-memory-behavior
- https://www.speedshop.co/2017/12/04/malloc-doubles-ruby-memory.html
- https://www.mikeperham.com/2018/04/25/taming-rails-memory-bloat/

What will we talk about?

Agenda

- Why should I care about scaling Puma?
- The dangers of following YouTube advice.
- What should I do?
- **Q&A**

Thank you!

TableCheck

- TableCheck is hiring Ruby on Rails, Python and Elixir developers, as well as Data Operations Engineers.
- Also other roles such as project managers, QA engineers, etc.
- Chat with me if you would like to know more!
- Also, thank you to Nate Berkopec, maintainer of Puma!