Resize Your Pods In-Place with Deterministic eBPF Triggers

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Pablo Chico de Guzman
CTO
Okteto
@pchico83 twitter

Vinay Kulkarni
Principal Architect
Futurewei Technologies
@iSkibum twitter

Resize Your Pods In-Place with Deterministic eBPF Triggers

KubeCon CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

KubeCon CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

October 24-28, 2021

Pablo Chico de Guzman
CTO
Okteto
@pchico83 twitter

Vinay Kulkarni
Principal Architect
Futurewei Technologies
@iSkibum twitter

Agenda

- Cloud-Native Development Environments
  - They should be Cost Effective
- In-Place Pod Resize – the Need and Design
- In-Place Resize with eBPF
- Live Demo
- Takeaways

Use Case

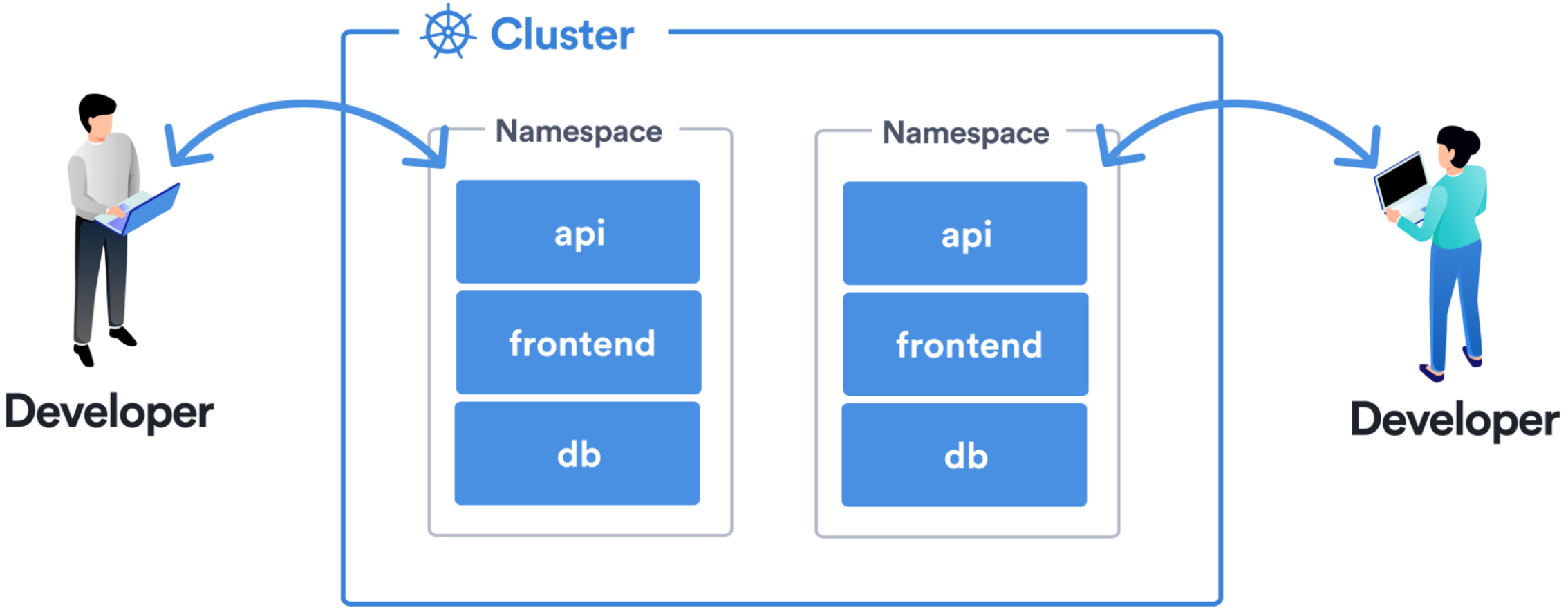
Cloud-Native Development Environments

The End of Local Development

- Bye bye monoliths 👋
- Mimicking production became increasingly tough:
  - Lack of CPU/Memory
  - Deployment of dependencies
  - Local configurations

- Environments disparity:
  - More integration issues
  - Less productivity
  - Less developer happiness

The Dawn of Cloud-Native Development



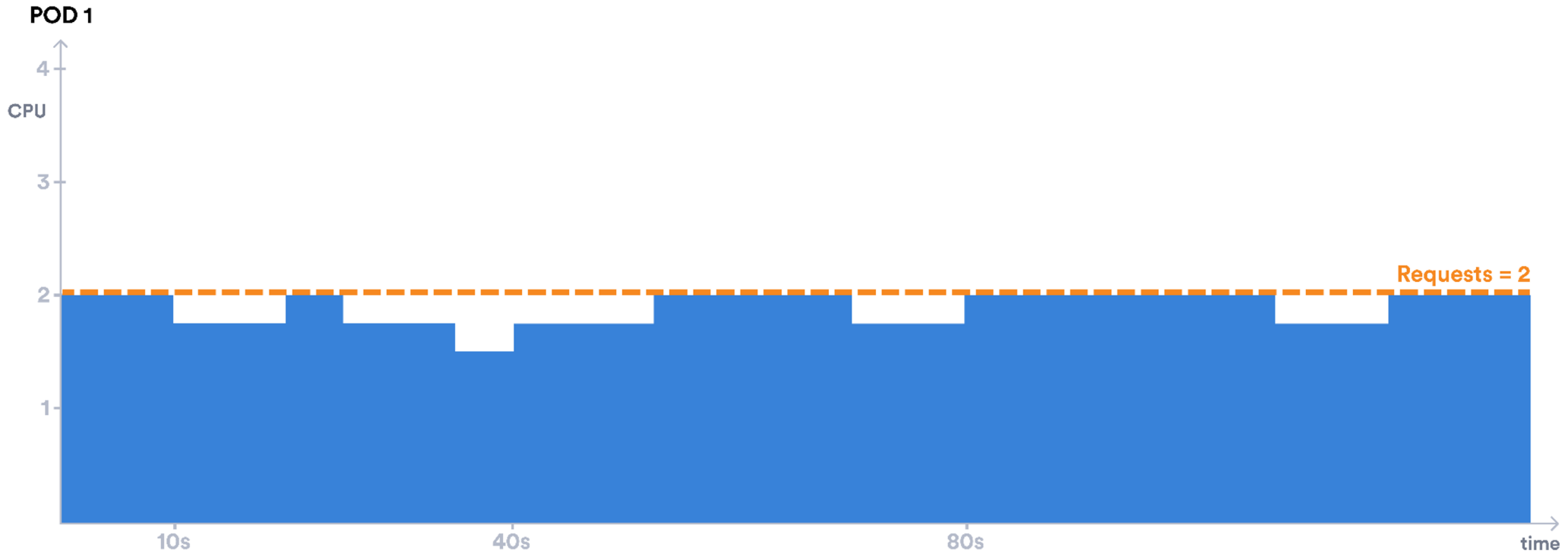
The Problem

Cost Reduction for Cloud-Native Development Environments

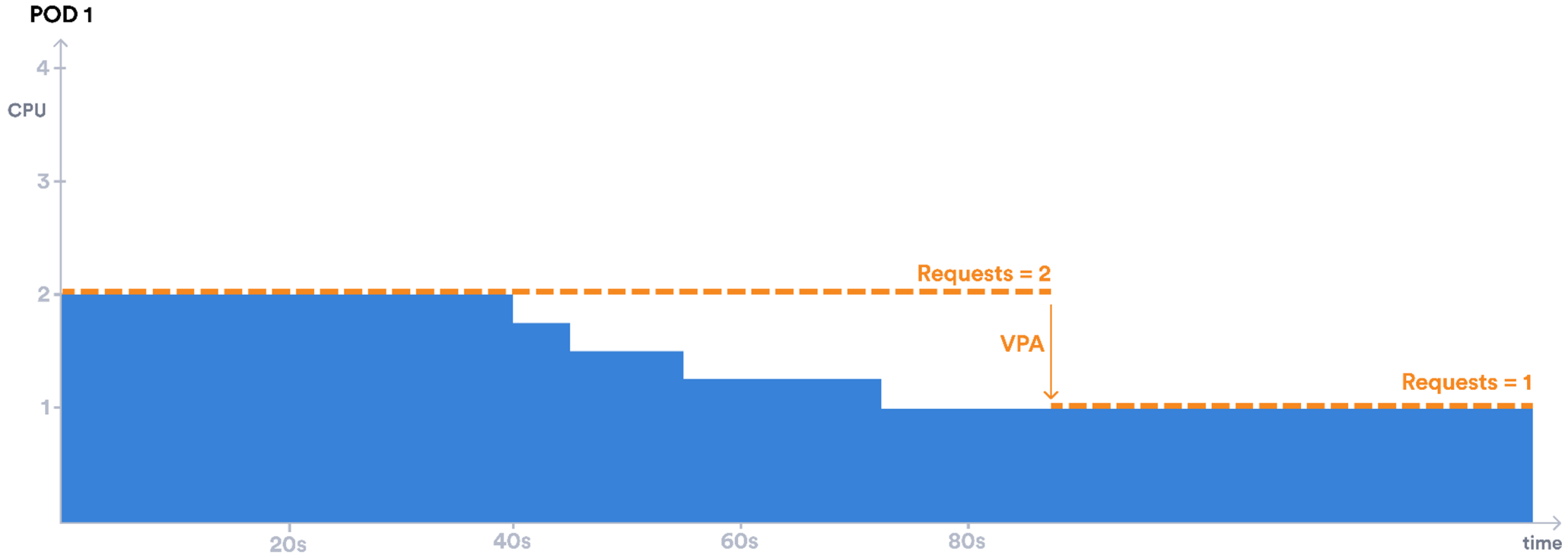
Cost Reduction

- Cloud-Native environments have an impact in your cloud bill 
- Kubernetes is great for resource allocation
  - CPU/Memory shared by several containers
  - Available capacity for spikes using “resource limits”
  - Cluster auto-scales up & down as needed
- But is Kubernetes that good for Development Environments?

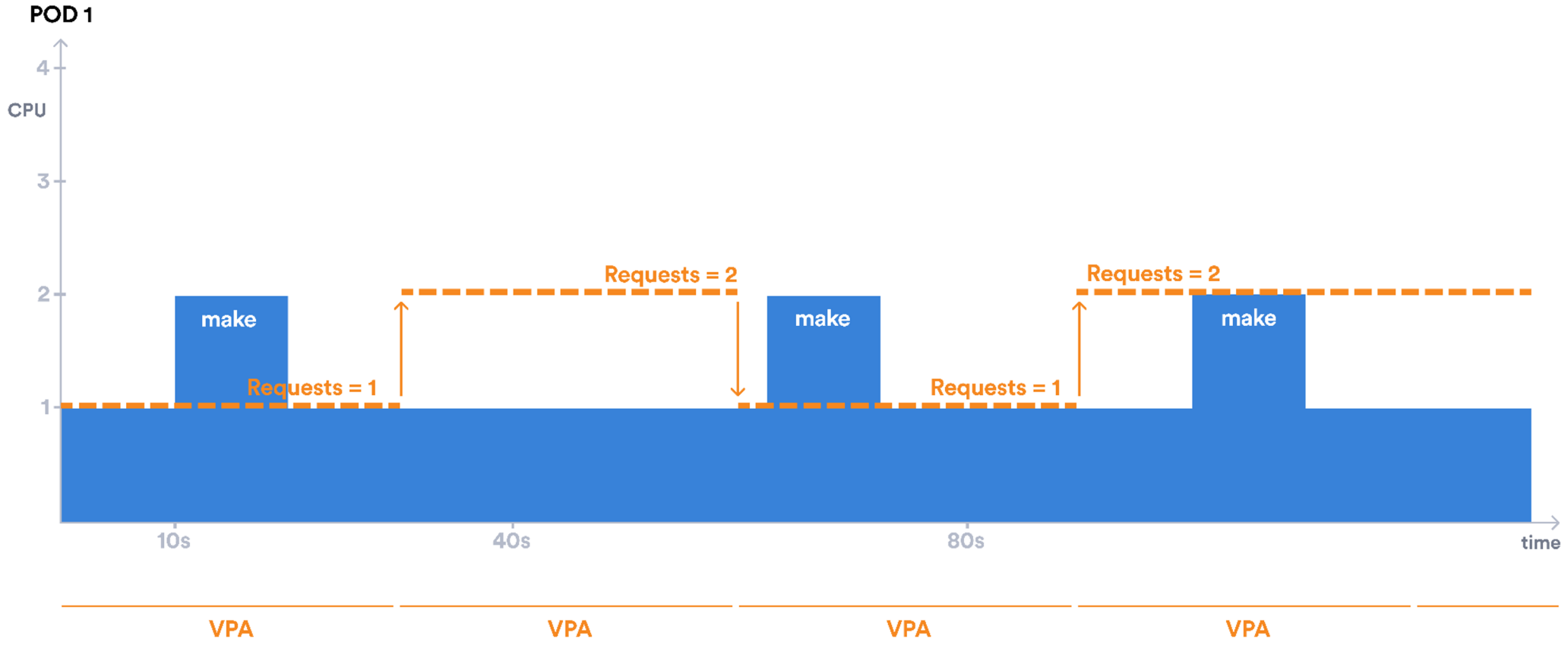
Predictable CPU Usage in Production



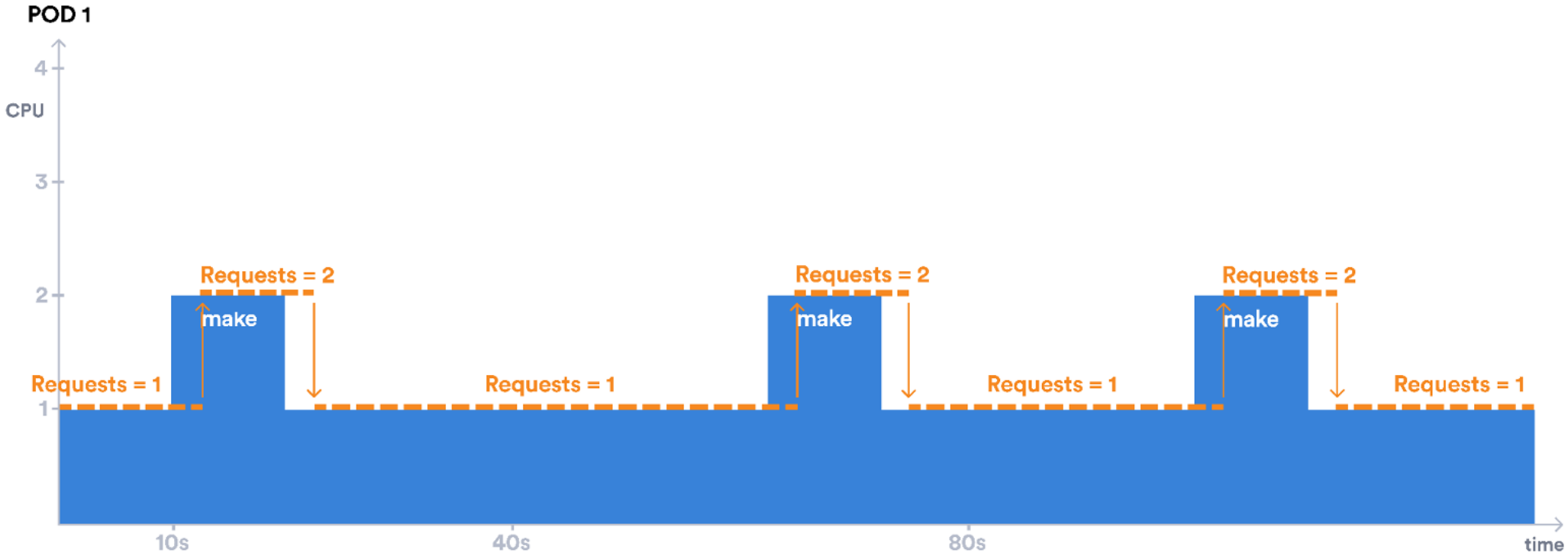
Booting Spikes + VPA + In-Place Pod Resize



“make” Spikes + VPA + In-Place Pod Resize



Optimal “make” Spikes Behavior



Cost Savings

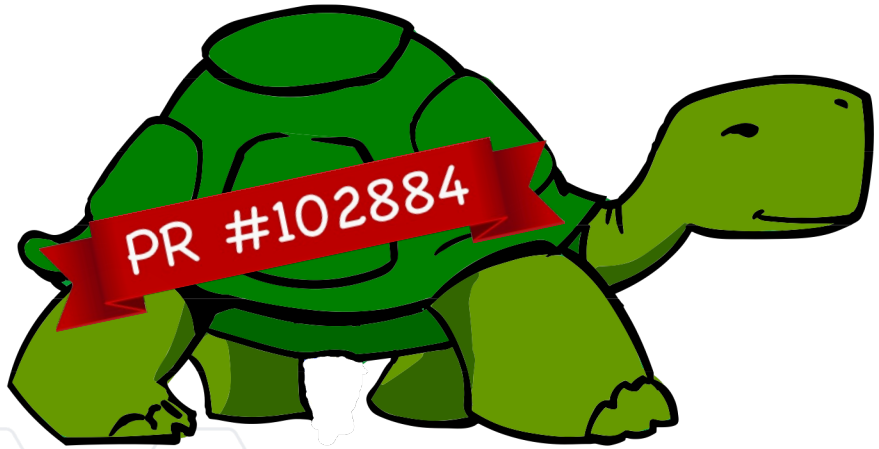
- Standard Kubernetes requests/limits:
  - 8 pods per node
- Ad-Hoc solution based on real cpu/memory load:
  - 80 pods per node

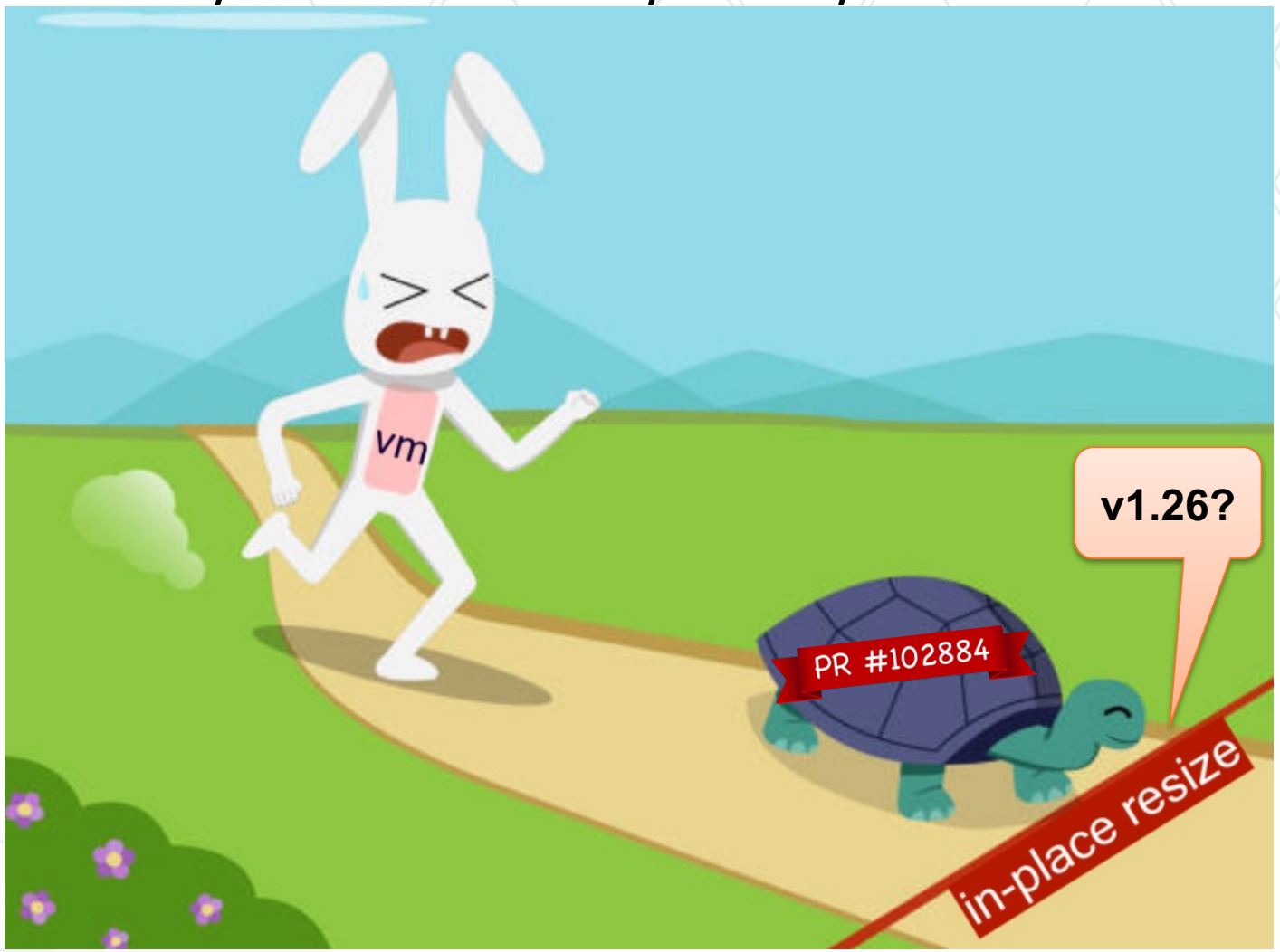
Is there a better way?

In-Place Pod Resize

In-Place Pod Resize

KEP: https://github.com/kubernetes/enhancements/issues/1287





In-Place Pod Resize – What Changed?

- PodSpec.Containers[i].Resources becomes mutable for cpu & memory to allow expressing desired resources
- PodStatus.Resize – new field that shows status of requested pod resize
  - Proposed: API has validated requested resize and updated object store (etcd)
  - InProgress: Resize request has been accepted by kubelet, and is being actuated
  - Deferred: Requested resize is possible, but not right now, kubelet will retry
  - InFeasible: Requested resize is never possible, no retries

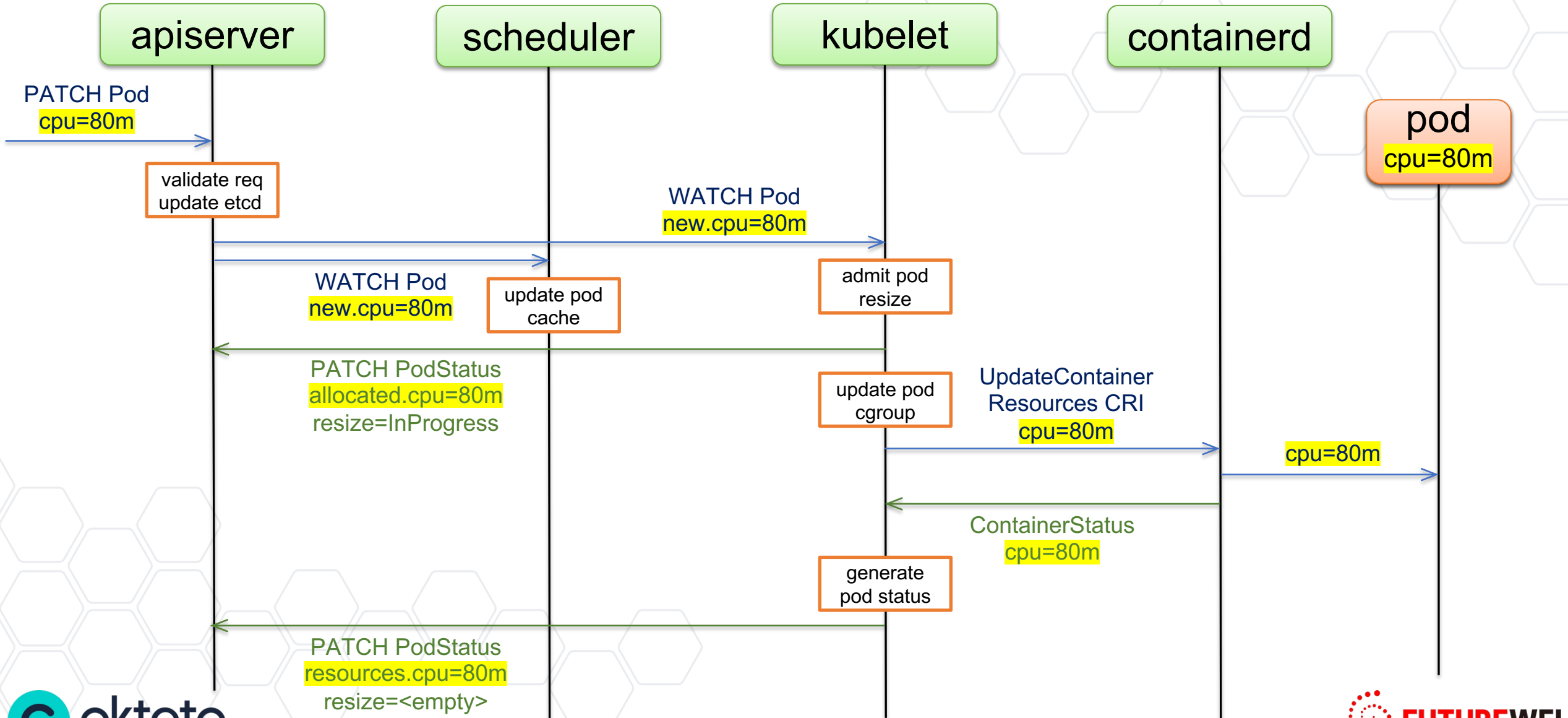
- PodStatus.ContainerStatuses[i].ResourcesAllocated – new field that shows requests allocated by kubelet for the pod
- PodStatus.ContainerStatuses[i].Resources – new field that shows actual resources configured for the pod

- PodSpec.Containers[i].ResizePolicy – new field that controls how to resize
  - RestartNotRequired: Resize container without restarting if possible (default)
  - RestartRequired: Restart the container when resizing it

In-Place Pod Resize – What Does It Involve?

- API Server
  - Validates desired resize request
  - Updates object store
- Kubelet
  - Admits desired resize request
  - Enacts pod cgroup resize
  - Invokes CRI for container resize
  - Updates resize status
- Scheduler
  - Supports resize by using max (Desired, Actual) during resize
- Runtime
  - Enacts desired container resources configuration

In-Place Pod Resize - Design



In-Place Pod Resize – Container Resize Order

- If net increase, pod cgroup is resized-up before resizing container(s)
- If net decrease, pod cgroup is resized-down after resizing container(s)
- Container resizes are ordered such that decreases are invoked before increases
- Scenario:
  - 2Gi memory available for pods
  - A single pod with two containers: c1 (1Gi) + c2 (1Gi)
  - Resize: c1(1Gi -> 1.5Gi), c2(1Gi -> 0.5Gi)
  - Resizing order [c1, c2] will fail – exceeds available memory

In-Place Resize – Expectations From Runtime

- Support UpdateContainerResources CRI API
  - Invoked to update container CPU and/or memory resources
- Synchronous and Transactional resource update handling
  - Return Success or Error
  - Do NOT queue a task for deferred/async processing
- Add support for newly added Resources field in ContainerStatus CRI API
  - Runtime response updates the newly added Resources field in PodStatus
- Add / Verify support for cgroup v2

In-Place Pod Resize – Innovating with eBPF

- Remote development environment example: K8s repo in a pod
  - Edit code and build K8s in a pod

```
1   apiVersion: v1
2   kind: Pod
3   metadata:
4     name: kube-build-pod
5   spec:
6     containers:
7     - name: kube-build-ctr
8       image: skiibum/kube-build-arm64:v1.25
9       imagePullPolicy: IfNotPresent
10      command: ["tail", "-f", "/dev/null"]
11      resources:
12        limits:
13          cpu: "5"
14          memory: "50Mi"
15        requests:
16          cpu: "4"
17          memory: "50Mi"
```

In-Place Pod Resize – Innovating with eBPF

KubeCon CloudNativeCon North America 2022

```
# podsnoop.py: Prototype eBPF program that snoops on pod exec activity
#              requires linux-headers, bpfcc-tools, kubectl
#              To run: sudo python3 podsnoop.py
import os
from bcc import BPF

POD_SNOOP_eBPF_CODE = r"""TRACEPOINT_PROBE(syscalls, sys_enter_execve) {
    char task_cmd[32];
    bpf_get_current_comm(&task_cmd, sizeof(task_cmd));
    bpf_trace_printk("Launching program: %s\n", task_cmd);
    return 0;
}"""
bpf = BPF(text = POD_SNOOP_eBPF_CODE)

while True:
    try:
        (task, pid, cpu, flags, ts, msg) = bpf.trace_fields()
        if str.__contains__(msg.decode("utf-8"), "make"):
            pod_name = os.popen("nsenter -t %s -u hostname 2>/dev/null" % pid).read().strip()
            if pod_name == "kube-build-pod":
                patch_str = '{"spec":{"containers":[{
                                "name":"kube-build-ctr",
                               "resources":{"requests":{"memory":"5Gi"},"limits":{"memory":"5Gi"}}}]}}'
                patch_cmd = "kubectl patch pod %s --patch '%s' > /dev/null" % (pod_name, patch_str)
                os.system(patch_cmd)
    except ValueError:
        continue
```

trace exec syscall

resize pod if “make” program launched

```
apiVersion: v1
kind: Pod
metadata:
  name: kube-build-pod
spec:
  containers:
  - name: kube-build-ctr
    image: skiibum/kube-build-arm64:v1.25
    imagePullPolicy: IfNotPresent
    command: ["tail", "-f", "/dev/null"]
    resources:
      limits:
        cpu: "5"
        memory: "50Mi"
      requests:
        cpu: "4"
        memory: "50Mi"
```

okteto

FUTUREWEI Technologies

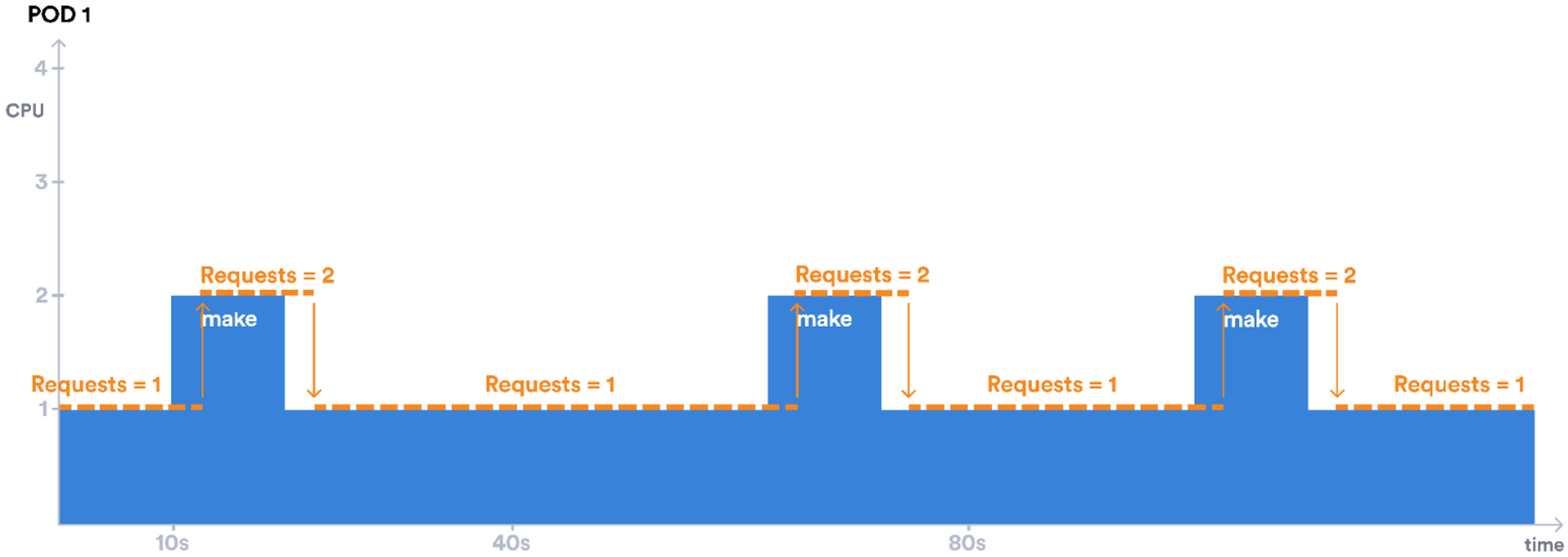
In-Place Pod Resize + eBPF + BPF maps

- eBPF program + BPF map
  - BPF_MAP_TYPE_HASH
  - Key: Container's Cgroup ID
  - Value: Commands to trace
    - E.g: "make"
- Pod Watcher
  - Read 'ebpf-resize' annotation
    - It specifies [container, [command, desired resources]]
  - Create BPF map entries [cgroup_id, [command]] based on annotation
- Initiate resize based on BPF trace
- Deployed as Daemonset
- https://github.com/vinaykul/ebpf-playground/tree/main/ebpf-pod-resize
- https://github.com/vinaykul/ebpf-playground/tree/main/ebpf-pod-resize-libbpf-rs-CO-RE

```
1   apiVersion: v1
2   kind: Pod
3   metadata:
4     name: kube-build-pod
5     annotations:
6       "ebpf-resize": '{
7               "cname": "kube-build-ctr",
8               "commands": ["make"],
9               "resize": "{\"requests\":{\"memory\":\"5Gi\"},\"limits\":{\"memory\":\"5Gi\"}}"
10      }'
11  spec:
12    containers:
13    - name: kube-build-ctr
14      image: skiibum/kube-build-arm64:v1.25
```

In-Place Pod Resize + eBPF – Live Demo

“make” Spikes + In-Place Pod Resize + eBPF



Key Takeaways

- Cloud-Native development environments
  - Ephemeral, Replicable, and Production-like
  - Cost-effective
- Add runtime support for container resource resize
  - Users don’t worry whether their runtime supports in-place resize
- Say “InPlacePodVerticalScaling=true”
  - Help us get to a rock-solid GA
  - Contributions are most welcome!
- Give eBPF a whirl!!

Almost Lunch Time…

KubeCon | CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Please scan the QR Code above to leave feedback on this session