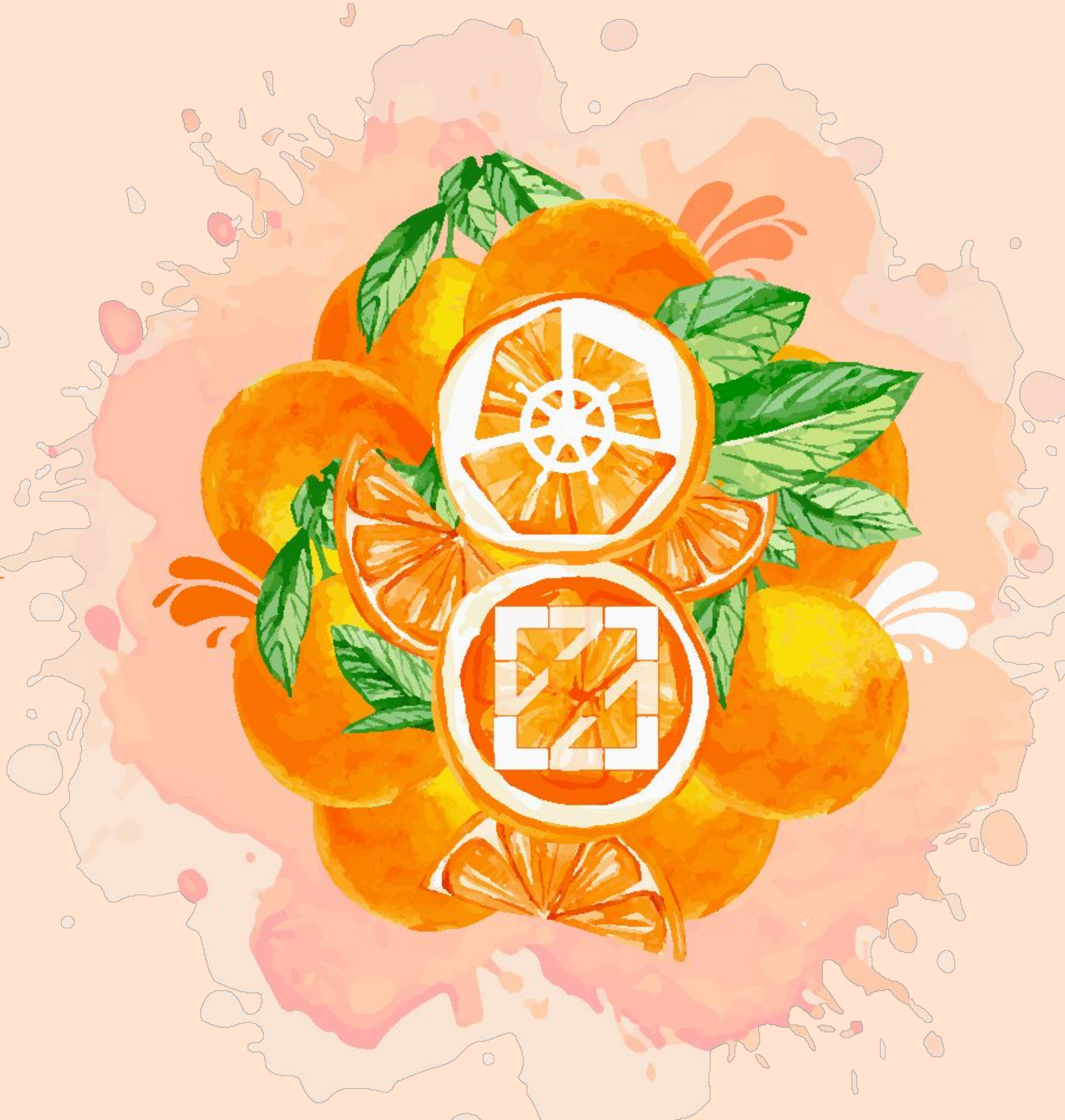
KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA



Scaling K8s Nodes Without Breaking the Bank nor Your Sanity

Brandon Wagner & Nick Tran, AWS

Scaling K8s Nodes Without Breaking the Bank nor Your Sanity

Agenda

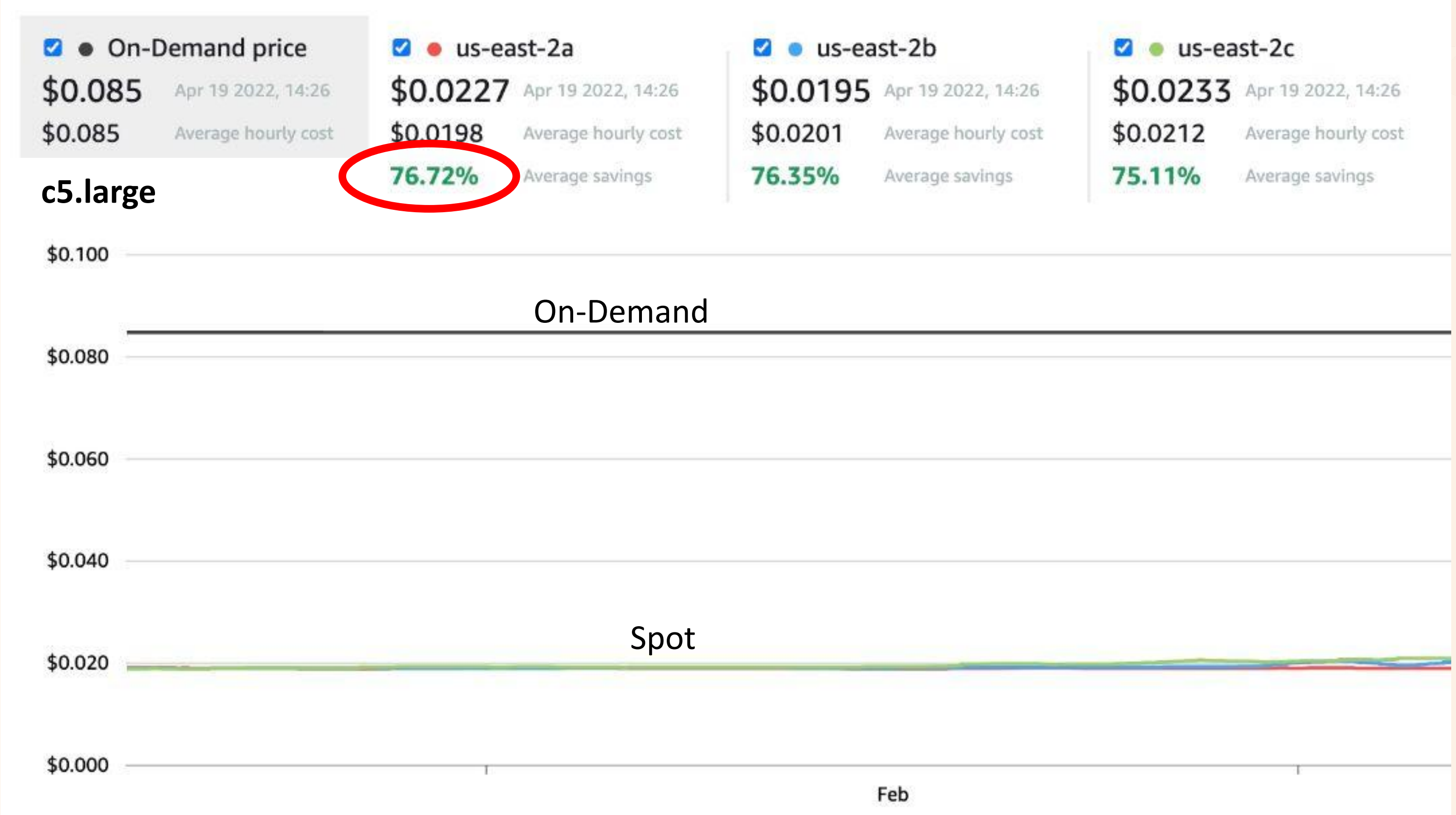
- What is Spot? 🐶
- Best Practices
- K8s & Spot ❤️
- Autoscaling your nodes
  - Cluster Autoscaler
  - Karpenter
- Demo

Brandon Wagner
Software Engineer
AWS

Nick Tran
Software Engineer
AWS

What is EC2 Spot?

- Spare VM Capacity
- Available at a discount
- Interruptible
  - 2-min notice



☑ ● On-Demand price
$0.085 Apr 19 2022, 14:26
$0.085 Average hourly cost

☑ ● us-east-2a
$0.0227 Apr 19 2022, 14:26
$0.0198 Average hourly cost
76.72% Average savings

☑ ● us-east-2b
$0.0195 Apr 19 2022, 14:26
$0.0201 Average hourly cost
76.35% Average savings

☑ ● us-east-2c
$0.0233 Apr 19 2022, 14:26
$0.0212 Average hourly cost
75.11% Average savings

c5.large

- Intro to spot
    - Talk about how spot is cheap why you might want to use vs on demand
- Downfalls of spot and how to best handle it
    - IIN (Instance Interruption Notice)
    - Eviction
- Scale up in Node Autoscaling with K8s and spot
    - HPA/VPA - make pods
    - CAS - need nodes for those specific pods
    - Karpenter - solution for how CAS is not easy (High Level)
- Spot Best Practices and how Karpenter does it
    - Do not use spot max price
    - Flexible instance types
    - Rebalance recommendations
- Demo

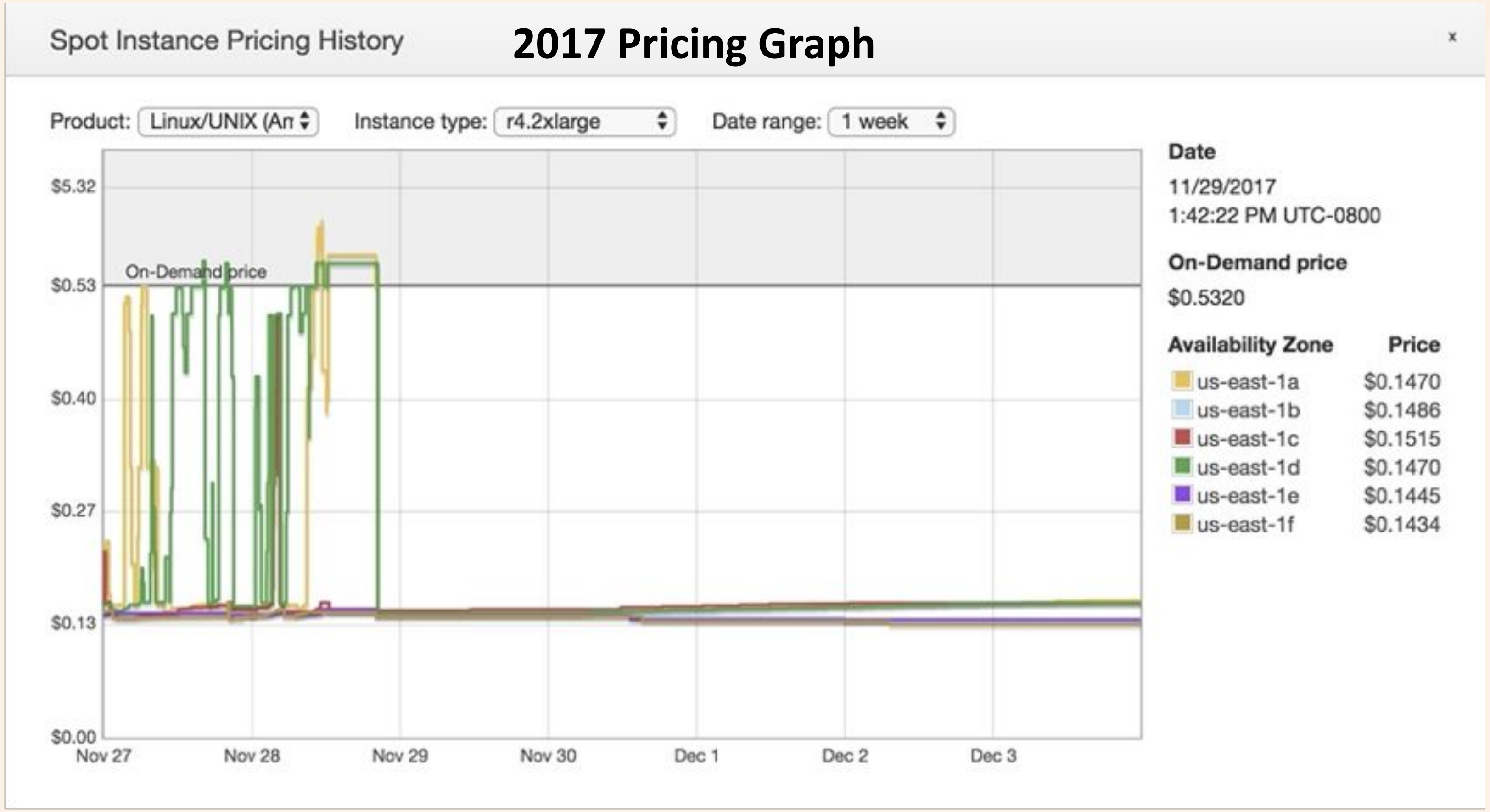
Spot Best Practices

- Don’t set a Spot max price
- Flexible instance type requests
- Rebalance Recommendations

Spot Best Practices - Don’t Set a Max Price

- Spot pricing model overhaul

- Long-term supply and demand



Spot Instance Pricing History

2017 Pricing Graph

Product: Linux/UNIX (An

Instance type: r4.2xlarge

Date range: 1 week

Date

11/29/2017
1:42:22 PM UTC-0800

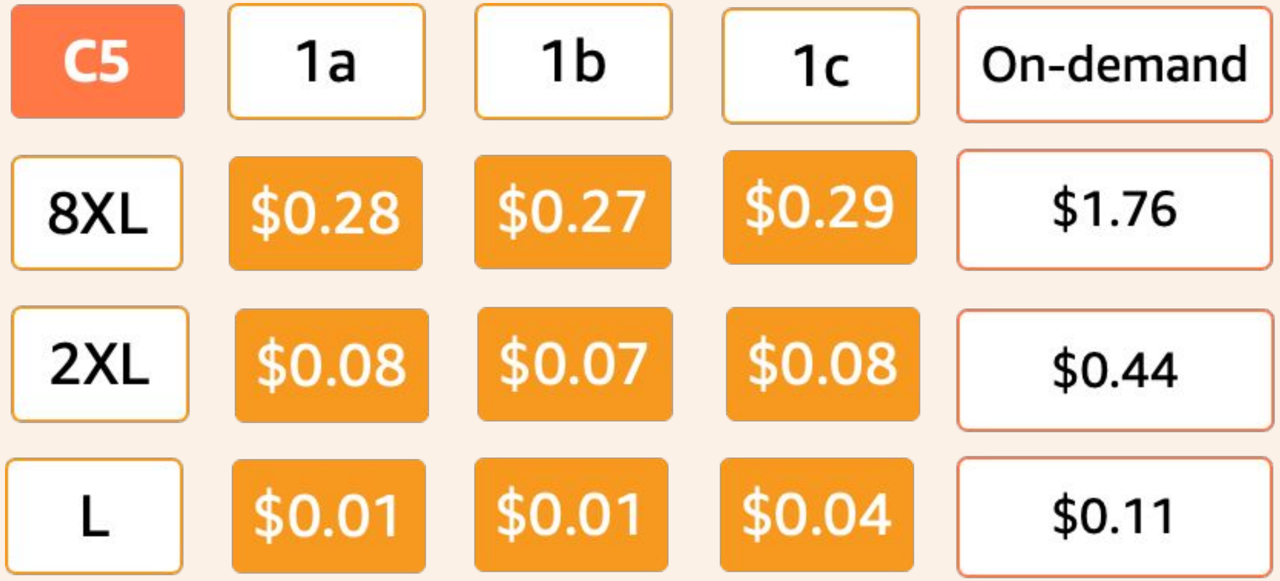
On-Demand price

$0.5320

Availability Zone	**Price**
🟨 us-east-1a	$0.1470
🟦 us-east-1b	$0.1486
🟥 us-east-1c	$0.1515
🟩 us-east-1d	$0.1470
🟪 us-east-1e	$0.1445
🟫 us-east-1f	$0.1434

Spot Best Practices - Flexible Instance Types

- Increases Spot instance availability
- Capacity pools
- Extend instance runtime
  - w/ capacity-optimized

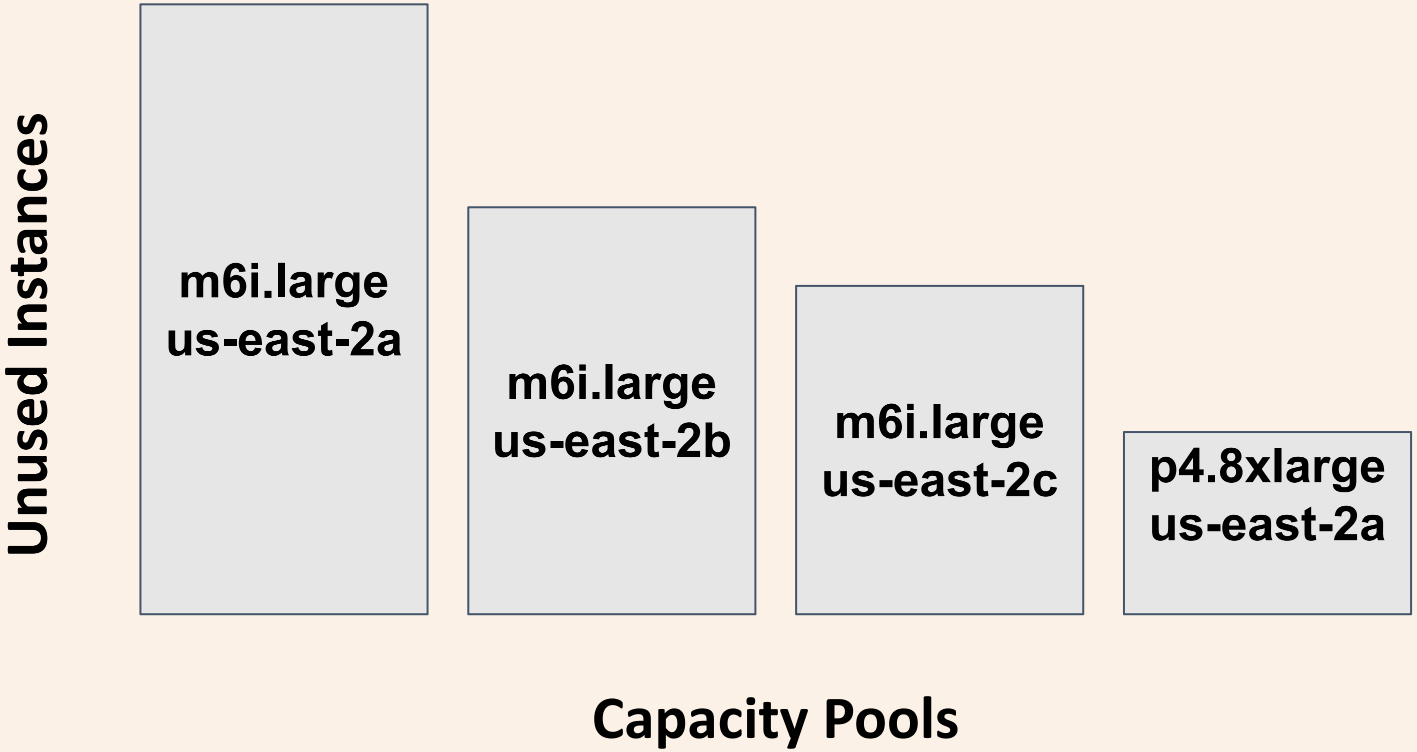


C5	1a	1b	1c	On-demand
8XL	$0.28	$0.27	$0.29	$1.76
2XL	$0.08	$0.07	$0.08	$0.44
L	$0.01	$0.01	$0.04	$0.11

Example Hourly Prices

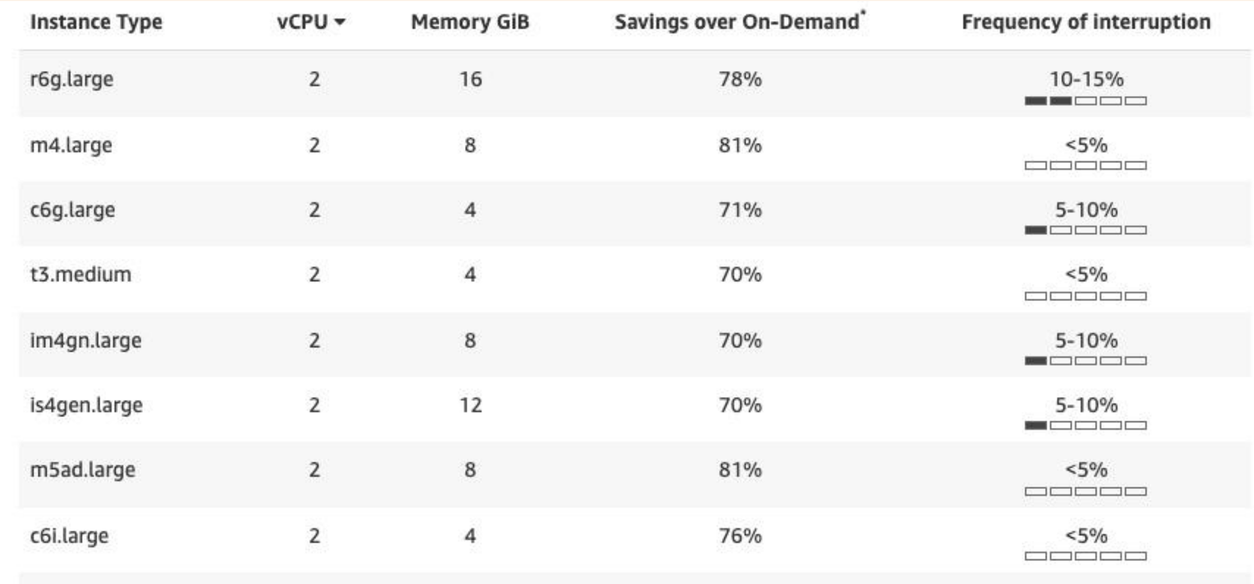
Spot Best Practices - Flexible Instance Types

- Increases Spot instance availability
- Capacity pools
- Extend instance runtime
  - w/ capacity-optimized



Spot Best Practices - Flexible Instance Types

- Increases Spot instance availability
- Capacity pools
- Extend instance runtime
  - w/ capacity-optimized



Instance Type	vCPU ▾	Memory GiB	Savings over On-Demand*	Frequency of interruption
r6g.large	2	16	78%	10-15% ■■□□□
m4.large	2	8	81%	<5% □□□□□
c6g.large	2	4	71%	5-10% ■□□□□
t3.medium	2	4	70%	<5% □□□□□
im4gn.large	2	8	70%	5-10% ■□□□□
is4gen.large	2	12	70%	5-10% ■□□□□
m5ad.large	2	8	81%	<5% □□□□□
c6i.large	2	4	76%	<5% □□□□□

https://aws.amazon.com/ec2/spot/instance-advisor

Spot Best Practices - Rebalance Recommendations

- Early warning to indicate a possible Spot interruption
- More time to gracefully shutdown workloads

Common Workloads

- Quick Continuous Integration
- Batch processing
- Stateless APIs

TEKTON

envoy

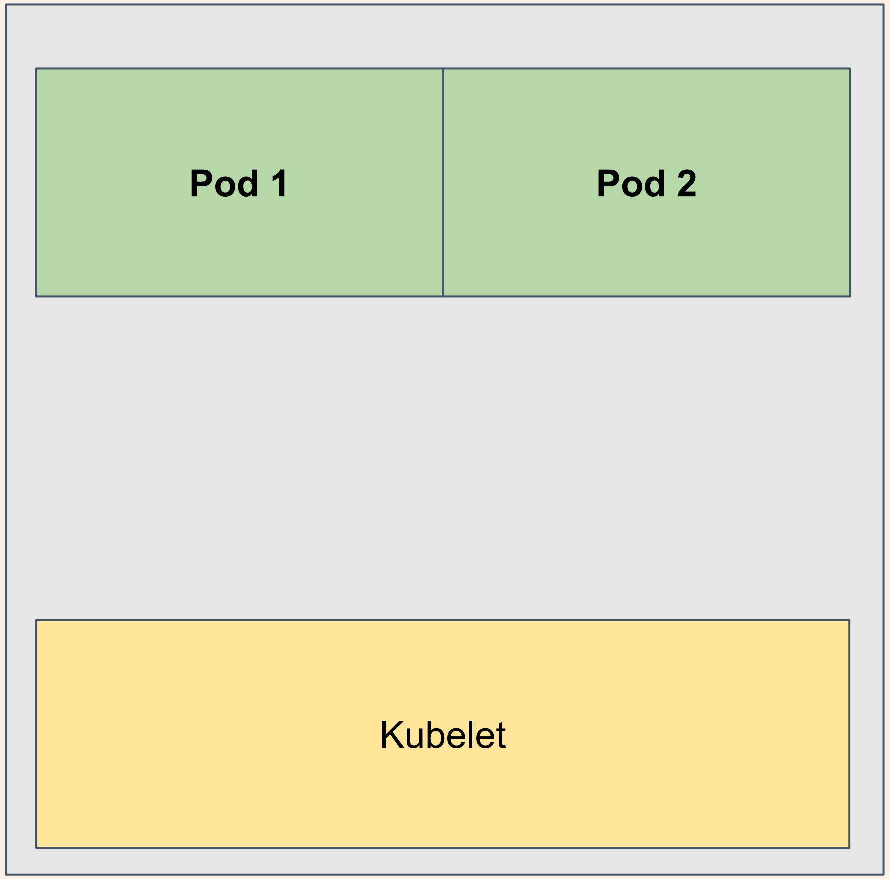
Istio

K8s and Spot

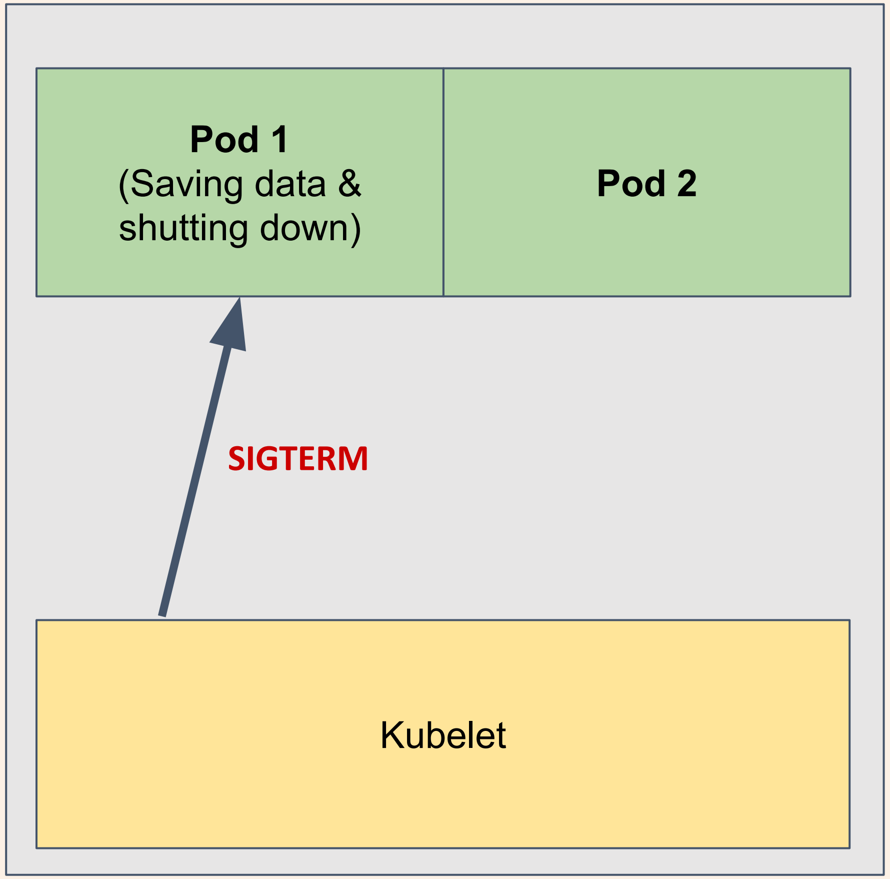
- github.com/aws/aws-node-termination-handler
  - Interruption Termination Notifications
  - Rebalance Recommendations

- Pod Disruption Budgets (PDBs)

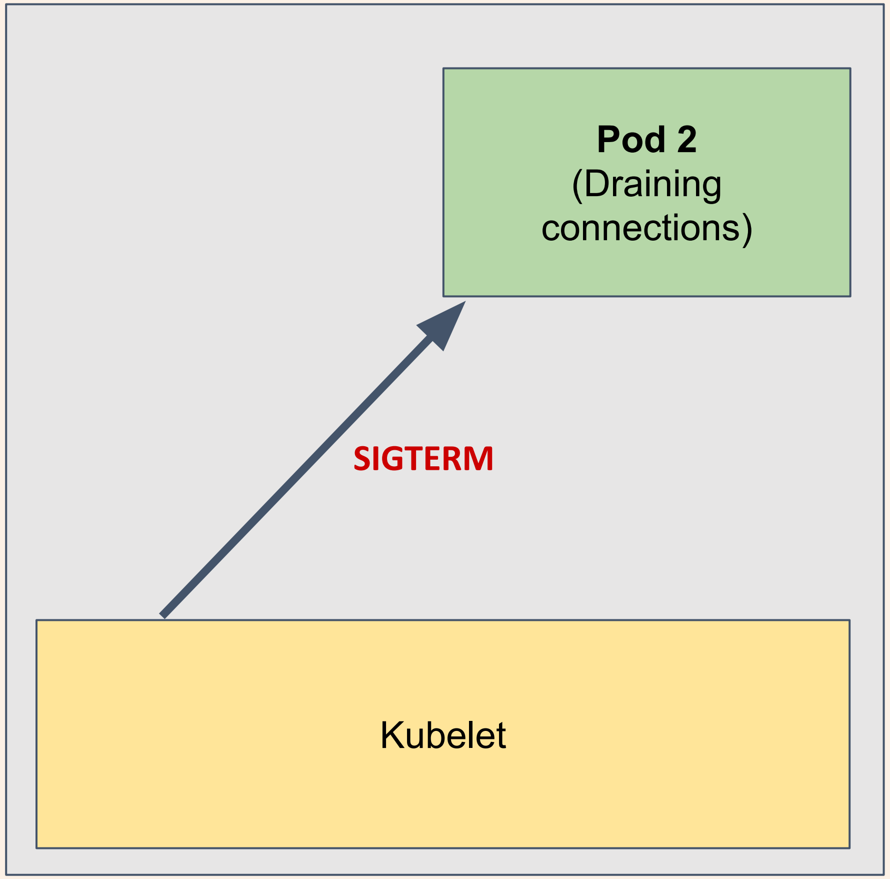
K8s and Spot - K8s Eviction API

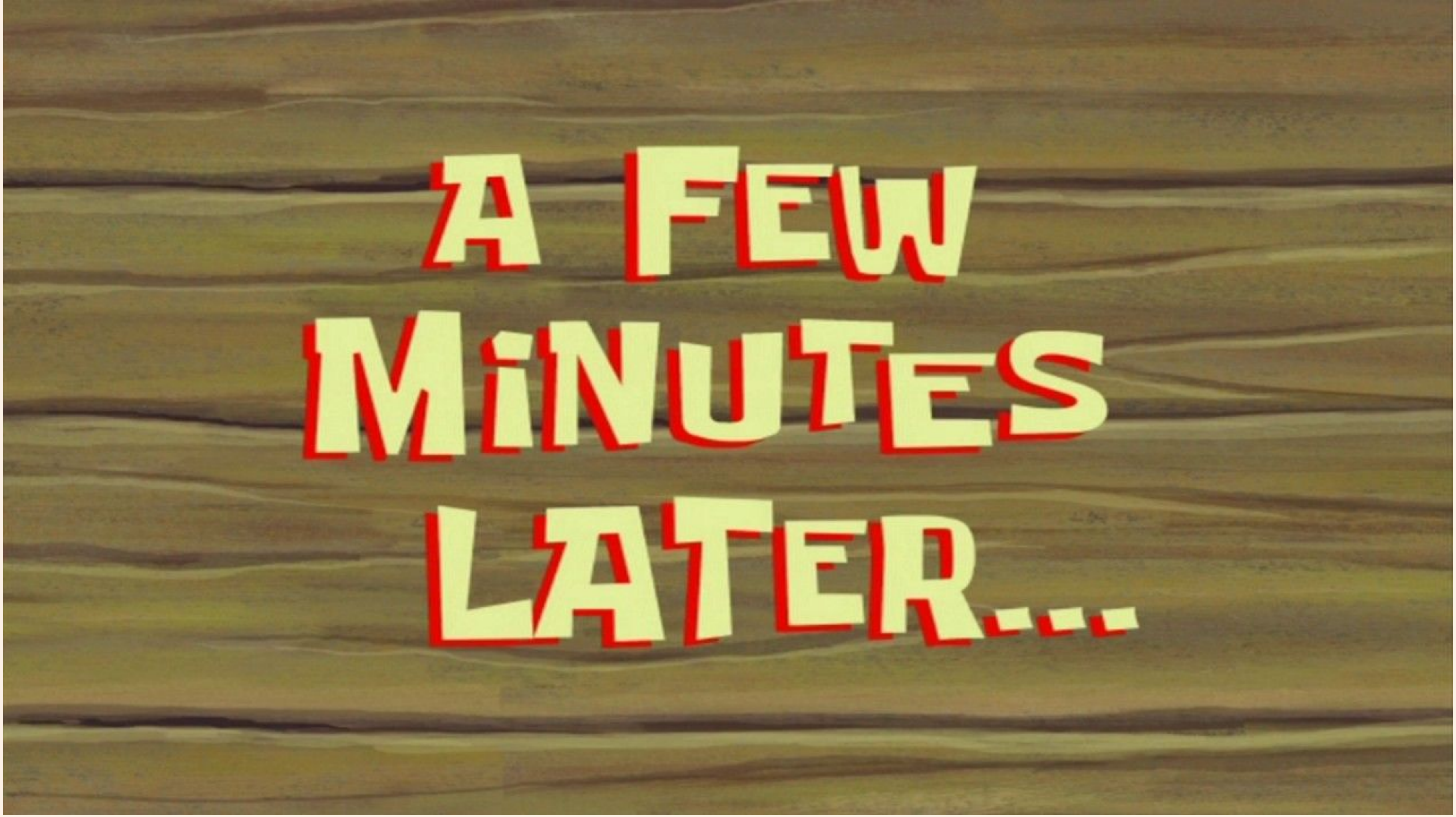


K8s and Spot - K8s Eviction API



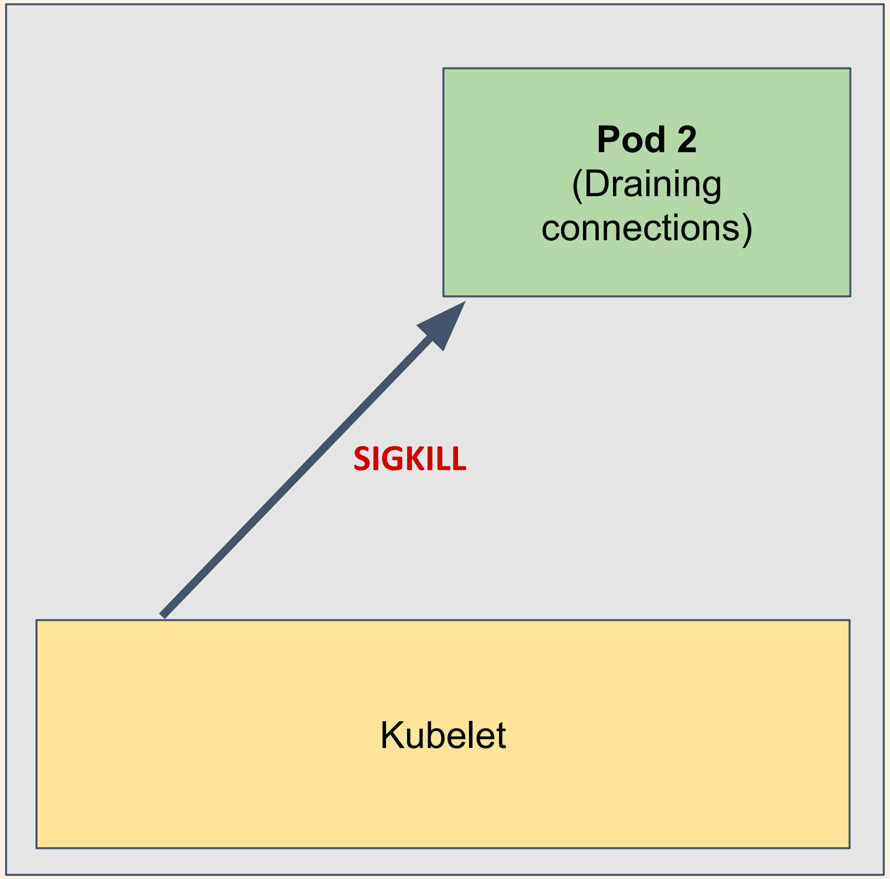
K8s and Spot - K8s Eviction API





A FEW MINUTES LATER...

K8s and Spot - K8s Eviction API

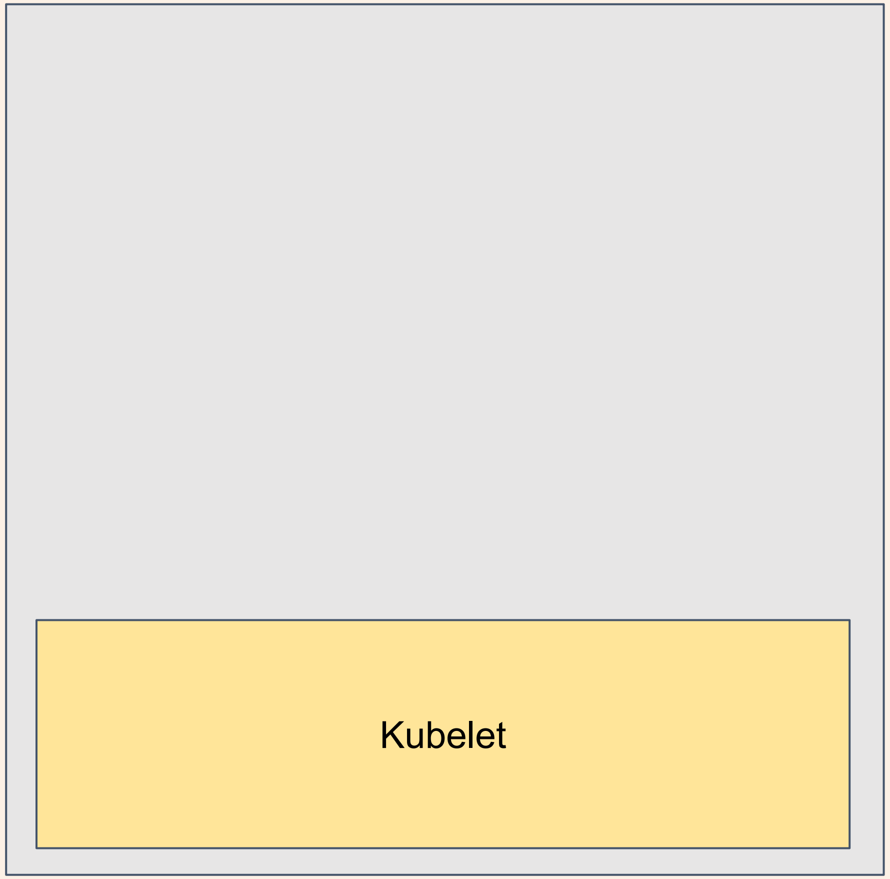


Pod 2
(Draining connections)

SIGKILL

Kubelet

K8s and Spot - K8s Eviction API



Kubelet

Autoscaling your Cluster

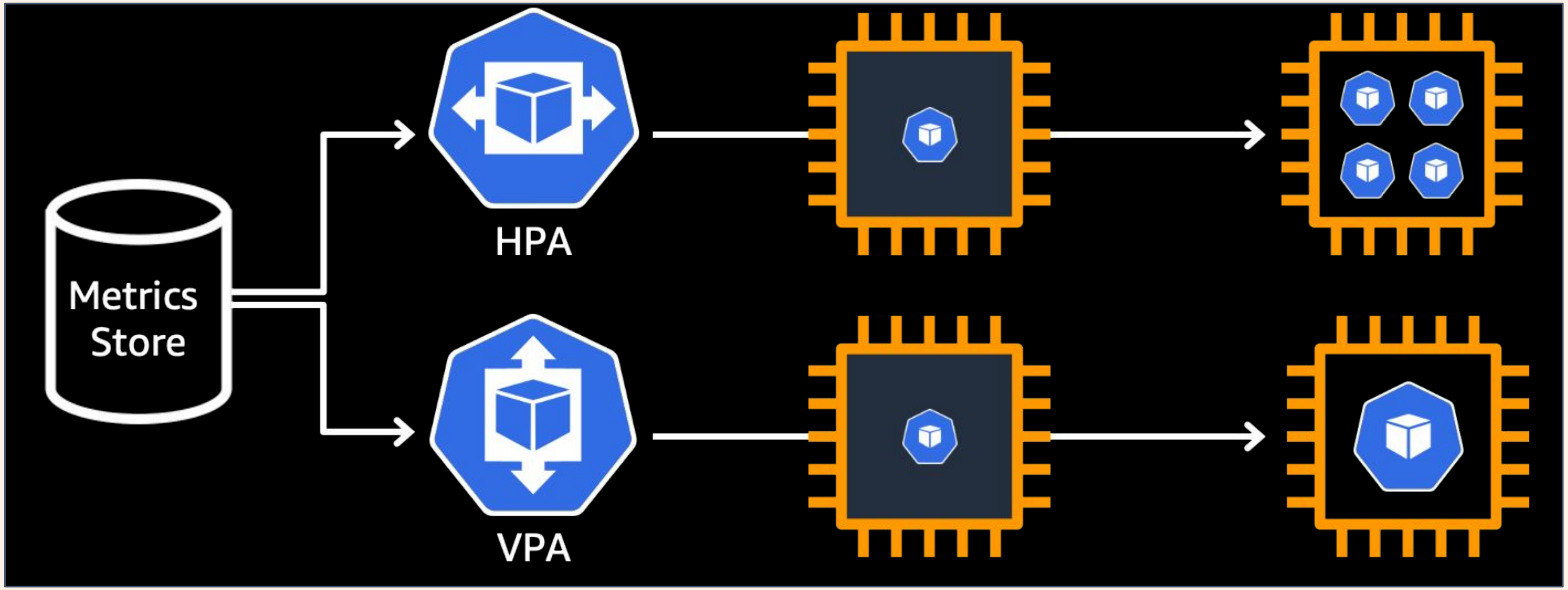
- Pod Autoscaling
  - Horizontal Pod Autoscaler (HPA)
  - Vertical Pod Autoscaler (VPA)

- Node Autoscaling
  - Cluster Autoscaler
  - Karpenter

HPA & VPA

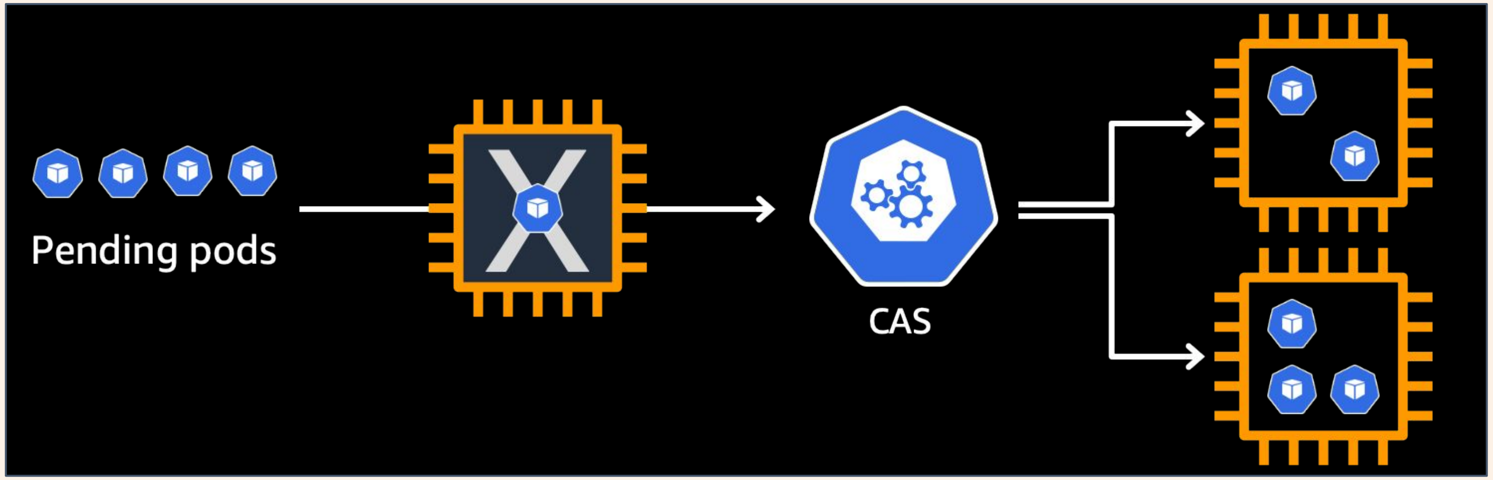
- Horizontally scale: adjust pod replicas

- Vertically scale: adjust resources of pods



Cluster Autoscaler

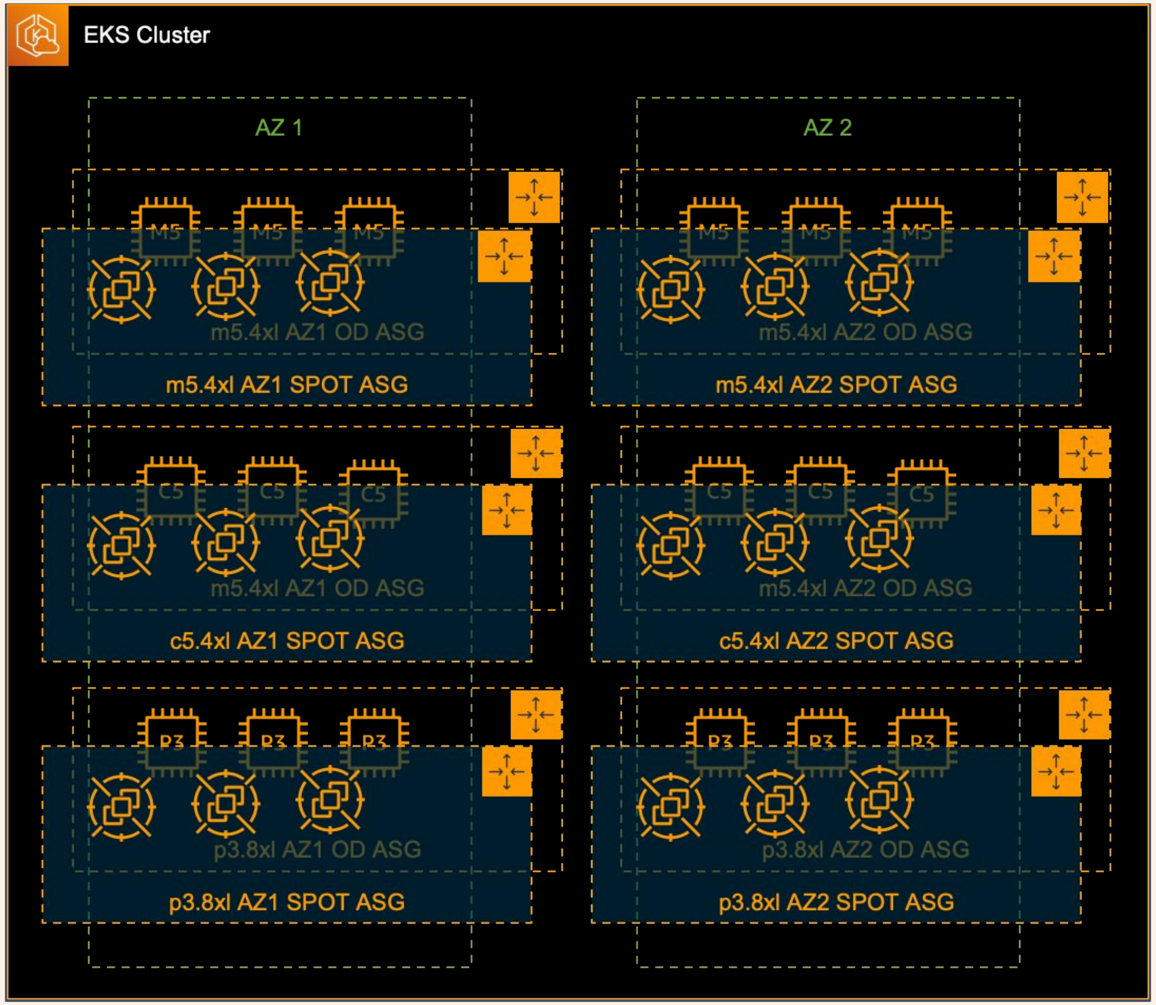
- Simple interface between EC2 AutoScaling Groups (ASGs)
- Increments desired capacity in response to pending pods
- Need to create resource workloads per type of pod resource request



Cluster Autoscaler

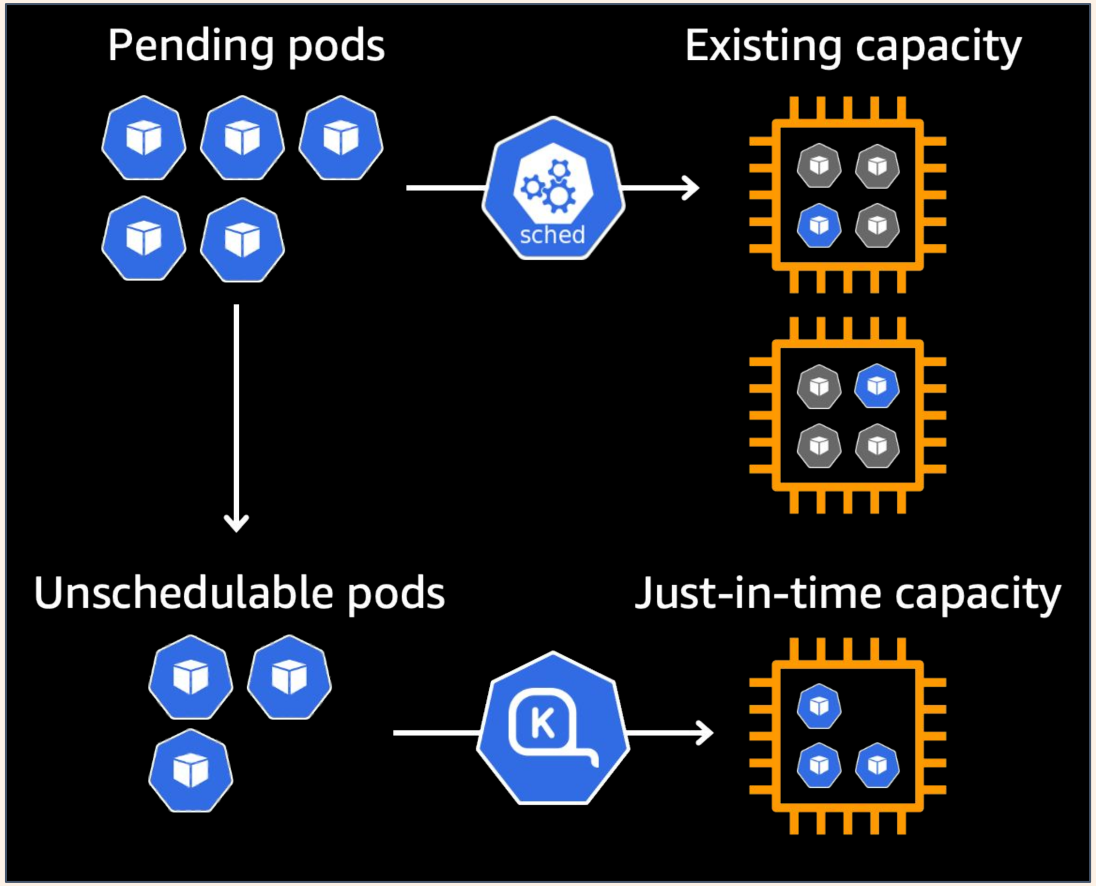
- Externally Managed Infrastructure

- [Spot, OD] x [AZ1, AZ2] x [m5, c5, p3] = 12 ASGs



Karpenter

- Groupless Node Autoscaler
- Just-in-Time Provisioning
  - Pending Pods
- github.com/aws/karpenter
  - Vendor neutral
    cloud provider interface

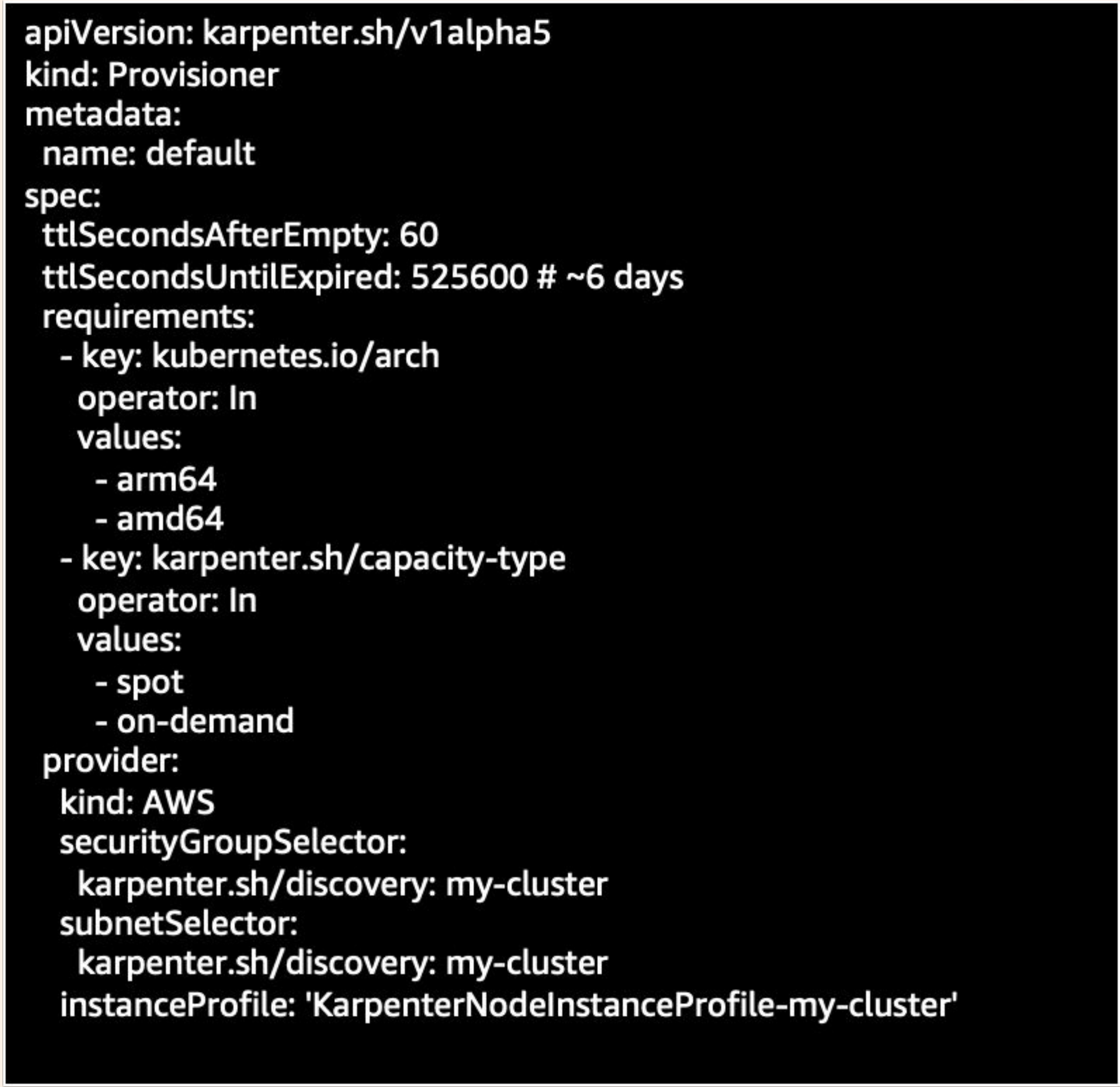


Karpenter

- Provisioner CRD

- Requirements
  - Scheduling Constraints
  - Well Known Labels
  - Capacity Type

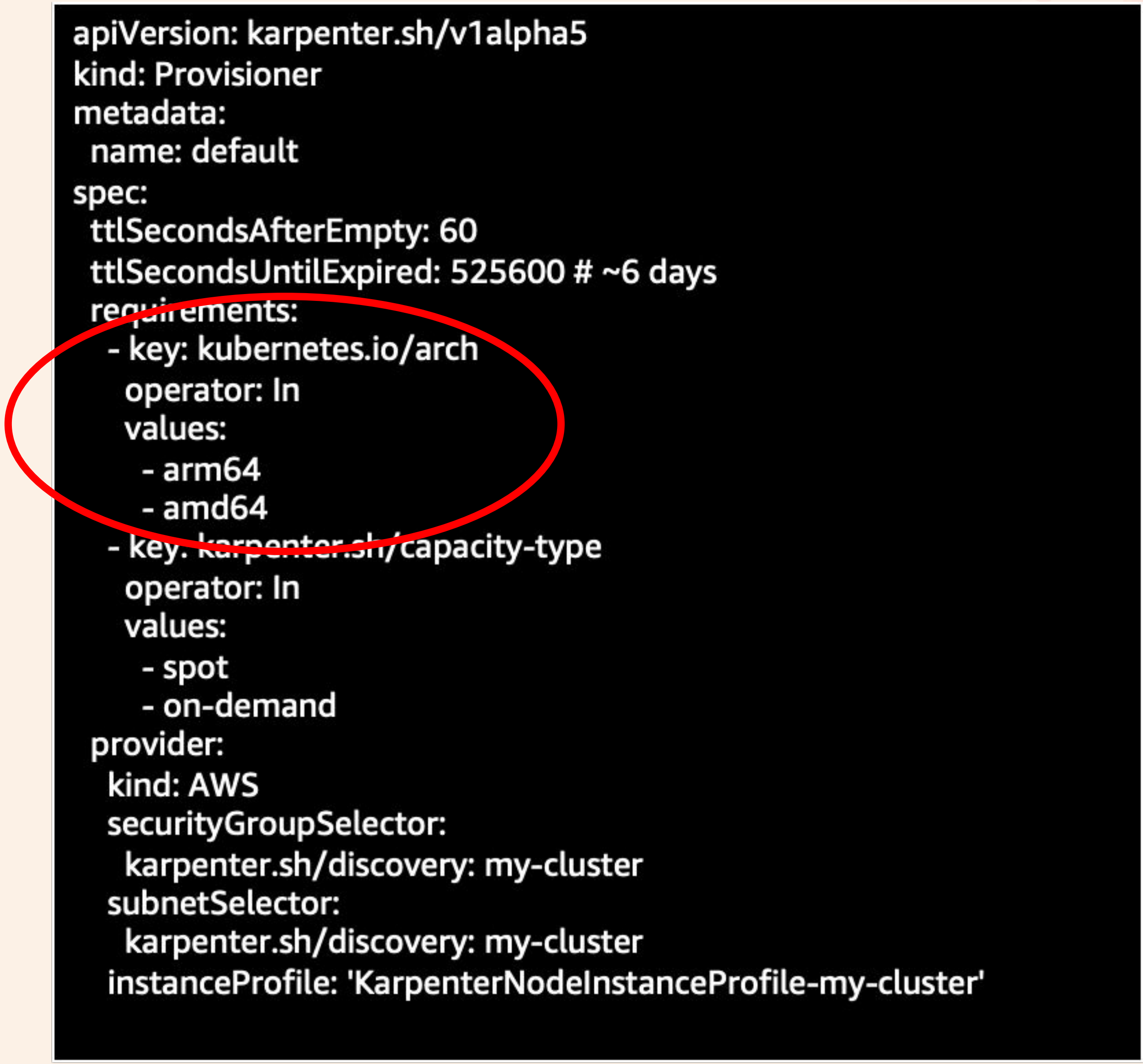
- Cloud Provider



```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  ttlSecondsAfterEmpty: 60
  ttlSecondsUntilExpired: 525600 # ~6 days
  requirements:
    - key: kubernetes.io/arch
      operator: In
      values:
        - arm64
        - amd64
    - key: karpenter.sh/capacity-type
      operator: In
      values:
        - spot
        - on-demand
  provider:
    kind: AWS
    securityGroupSelector:
      karpenter.sh/discovery: my-cluster
    subnetSelector:
      karpenter.sh/discovery: my-cluster
    instanceProfile: 'KarpenterNodeInstanceProfile-my-cluster'
```

Karpenter

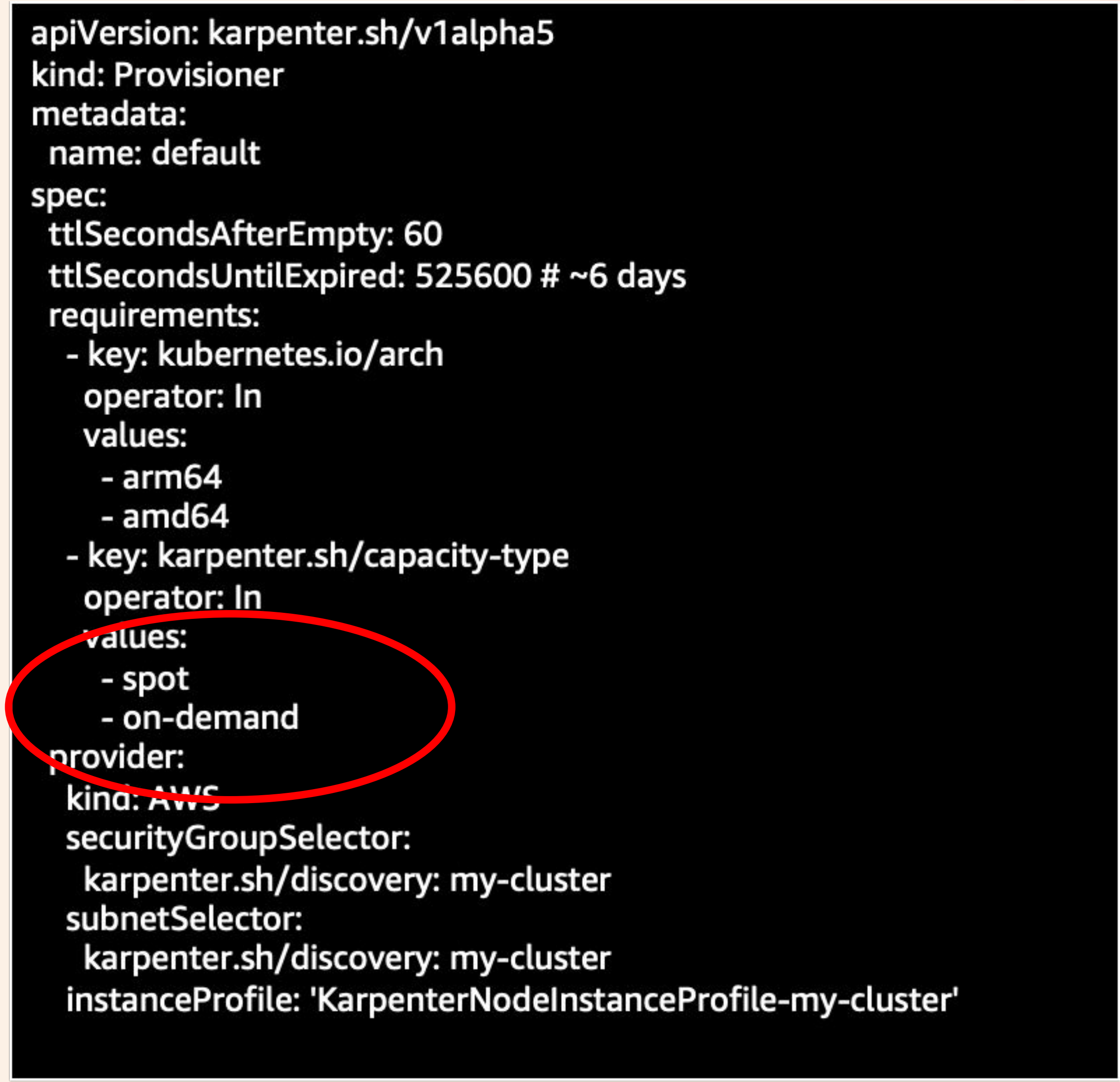
- Flexibility
  - CPU Architecture



```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  ttlSecondsAfterEmpty: 60
  ttlSecondsUntilExpired: 525600 # ~6 days
  requirements:
    - key: kubernetes.io/arch
      operator: In
      values:
        - arm64
        - amd64
    - key: karpenter.sh/capacity-type
      operator: In
      values:
        - spot
        - on-demand
  provider:
    kind: AWS
    securityGroupSelector:
      karpenter.sh/discovery: my-cluster
    subnetSelector:
      karpenter.sh/discovery: my-cluster
    instanceProfile: 'KarpenterNodeInstanceProfile-my-cluster'
```

Karpenter

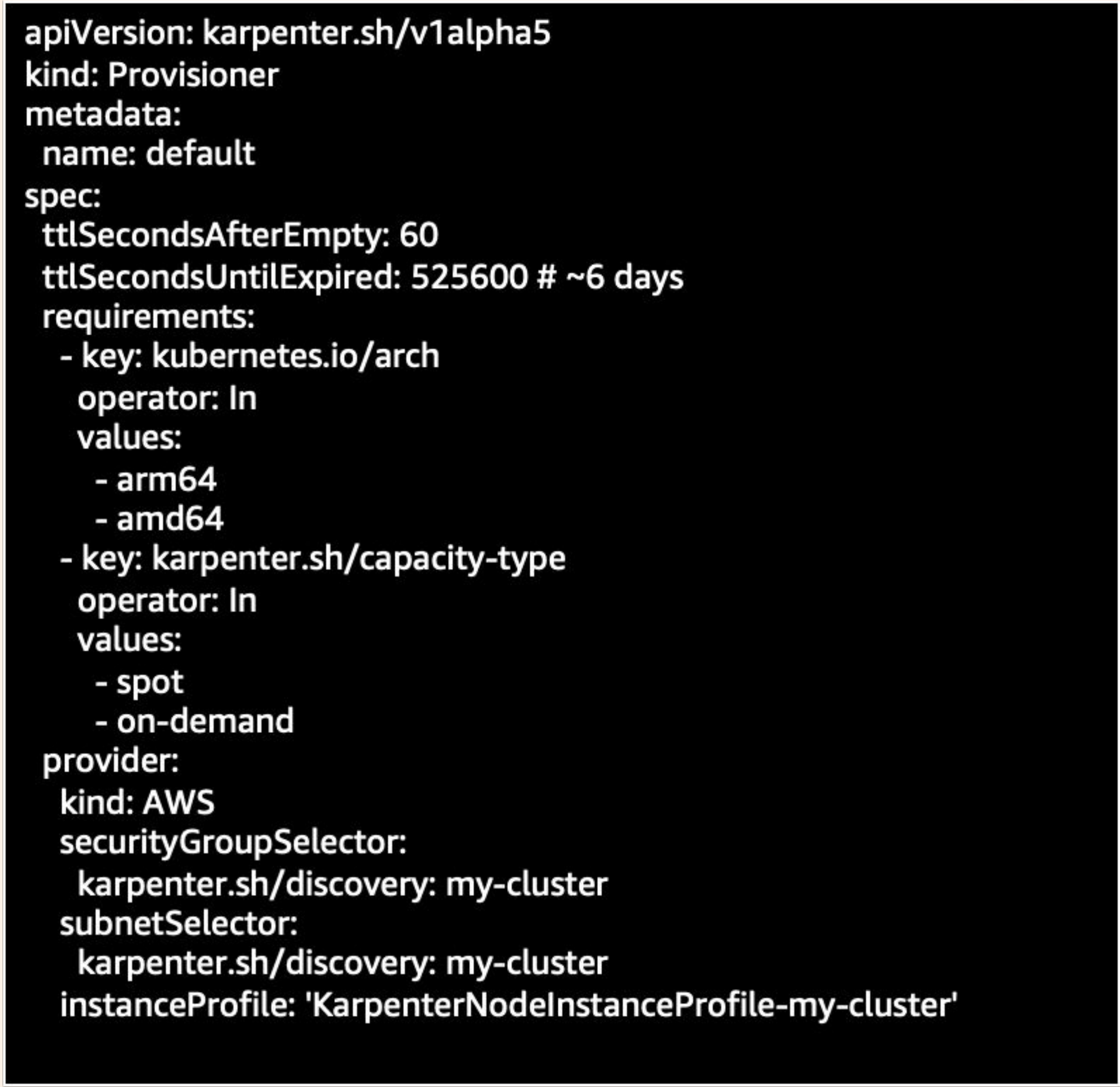
- Flexibility
  - CPU Architecture
  - Capacity Type



```yaml
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  ttlSecondsAfterEmpty: 60
  ttlSecondsUntilExpired: 525600 # ~6 days
  requirements:
    - key: kubernetes.io/arch
      operator: In
      values:
        - arm64
        - amd64
    - key: karpenter.sh/capacity-type
      operator: In
      values:
        - spot
        - on-demand
  provider:
    kind: AWS
    securityGroupSelector:
      karpenter.sh/discovery: my-cluster
    subnetSelector:
      karpenter.sh/discovery: my-cluster
    instanceProfile: 'KarpenterNodeInstanceProfile-my-cluster'
```

Karpenter

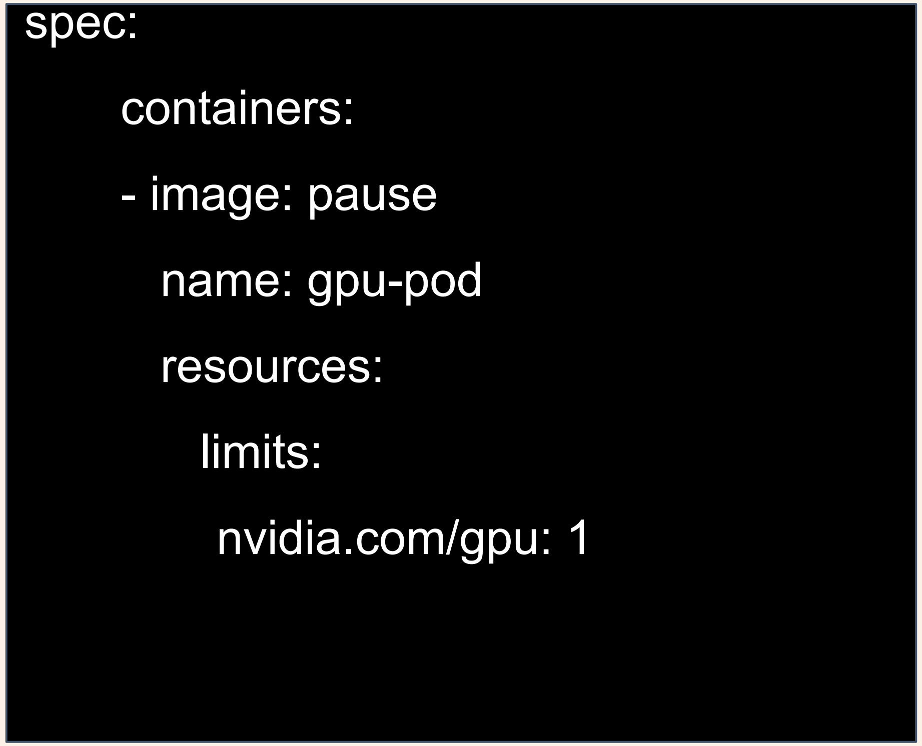
Where are the Instance Types?



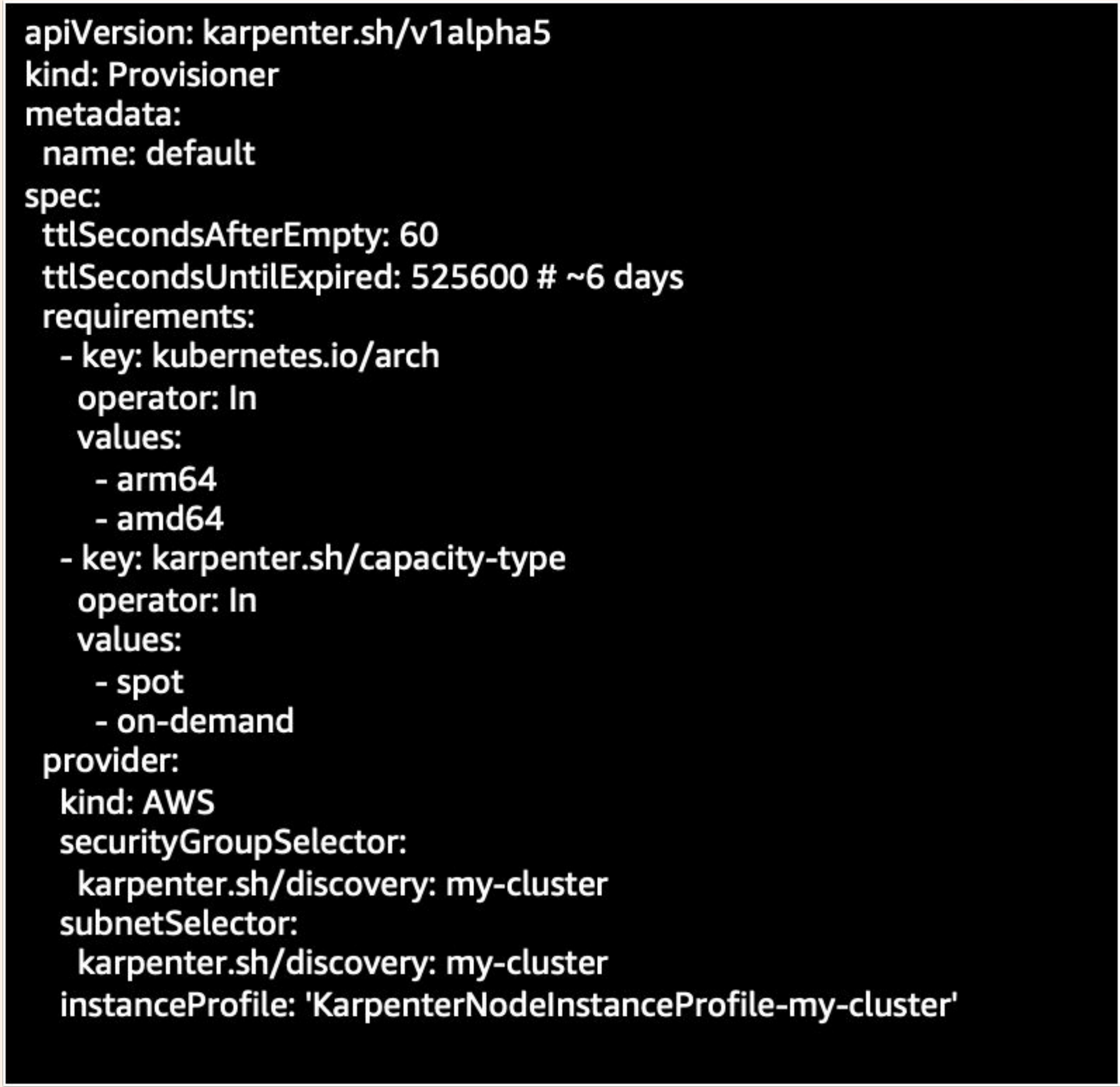
```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  ttlSecondsAfterEmpty: 60
  ttlSecondsUntilExpired: 525600 # ~6 days
  requirements:
    - key: kubernetes.io/arch
      operator: In
      values:
        - arm64
        - amd64
    - key: karpenter.sh/capacity-type
      operator: In
      values:
        - spot
        - on-demand
  provider:
    kind: AWS
    securityGroupSelector:
      karpenter.sh/discovery: my-cluster
    subnetSelector:
      karpenter.sh/discovery: my-cluster
    instanceProfile: 'KarpenterNodeInstanceProfile-my-cluster'
```

Karpenter

Where are the Instance Types?



```
spec:
    containers:
    - image: pause
      name: gpu-pod
      resources:
        limits:
          nvidia.com/gpu: 1
```



```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  ttlSecondsAfterEmpty: 60
  ttlSecondsUntilExpired: 525600 # ~6 days
  requirements:
    - key: kubernetes.io/arch
      operator: In
      values:
        - arm64
        - amd64
    - key: karpenter.sh/capacity-type
      operator: In
      values:
        - spot
        - on-demand
  provider:
    kind: AWS
    securityGroupSelector:
      karpenter.sh/discovery: my-cluster
    subnetSelector:
      karpenter.sh/discovery: my-cluster
    instanceProfile: 'KarpenterNodeInstanceProfile-my-cluster'
```

Karpenter - AWS Cloud Provider

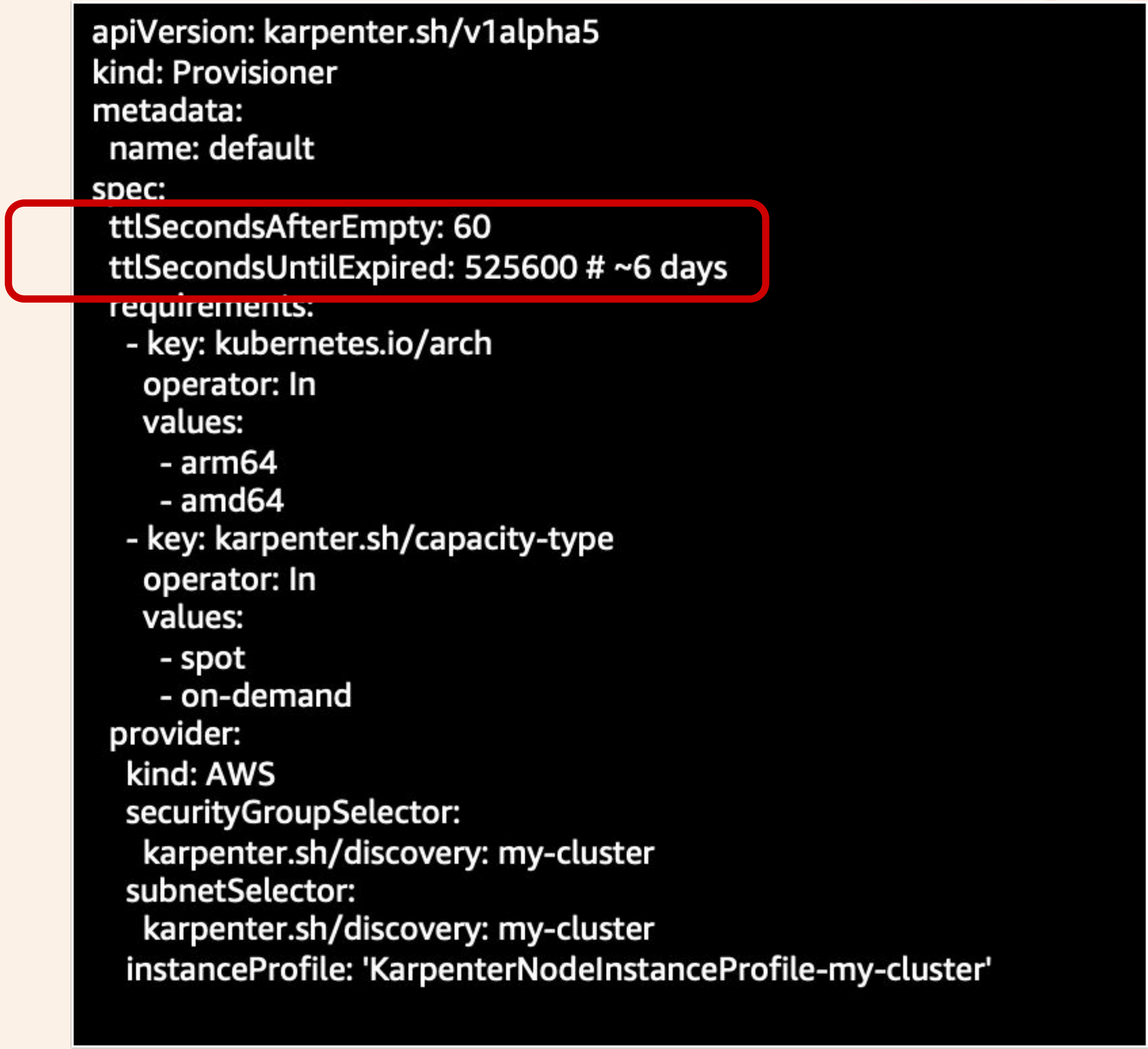
- EC2 Fleet API
  - Flexible to many instance types
  - Chooses optimal AZ and instance type

- Spot to On-Demand Fallback

Autoscaling Nodes - Karpenter

- Scaling down
  - ttlSecondsAfterEmpty
  - ttlSecondsUntilExpired

- Follows Graceful Node Shutdown

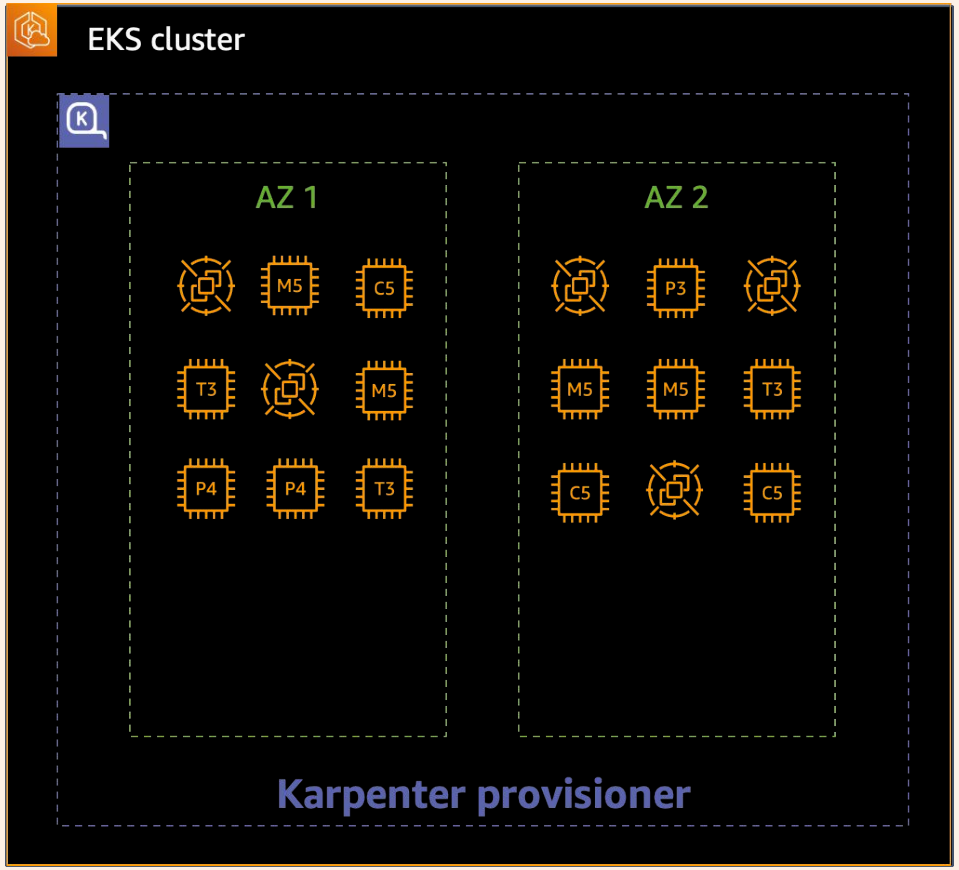


```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  ttlSecondsAfterEmpty: 60
  ttlSecondsUntilExpired: 525600 # ~6 days
  requirements:
    - key: kubernetes.io/arch
      operator: In
      values:
        - arm64
        - amd64
    - key: karpenter.sh/capacity-type
      operator: In
      values:
        - spot
        - on-demand
  provider:
    kind: AWS
    securityGroupSelector:
      karpenter.sh/discovery: my-cluster
    subnetSelector:
      karpenter.sh/discovery: my-cluster
    instanceProfile: 'KarpenterNodeInstanceProfile-my-cluster'
```

Autoscaling Nodes - CAS vs Karpenter

- [Spot, OD] x [AZ1, AZ2] x [m5, c5, p4, t3] = 16 ASGs

- One provisioner!



Wrapping Up

- Spot Best Practices
- K8s and Spot!
- Autoscaling nodes with Cluster Autoscaler and Karpenter

Demo!

- Provisioners
- Stuff
- More stuff

Questions?

Notes

- Switch off less
- Intra-section switches are rough -- overarching story to connect in the beginning helps
    - Pods -> Node Capacity story transition better
- Configuration bloat picture (why is it hard? mixed instance types in CAS, other cloud providers?)
- Kubecon ppl might get mad if we go super hard into AWS rhetoric
- More pictures on that one slide (not just tekton)
- Graceful Node Shutdown in K8s with kubelet vs NTH/Karpenter
- Deeper on fewer subjects better than shallow on more
- Explain instance pools better in combination with the price graph.
    - One .16xl vs 16 .xl?
    - column + row names
- Hourly vs Monthly rates for instances
- How frequent interruptions are
- Re-evaluate common Spot Workloads
- Talk + investigate more about Spot to OD fallback with EC2 folks