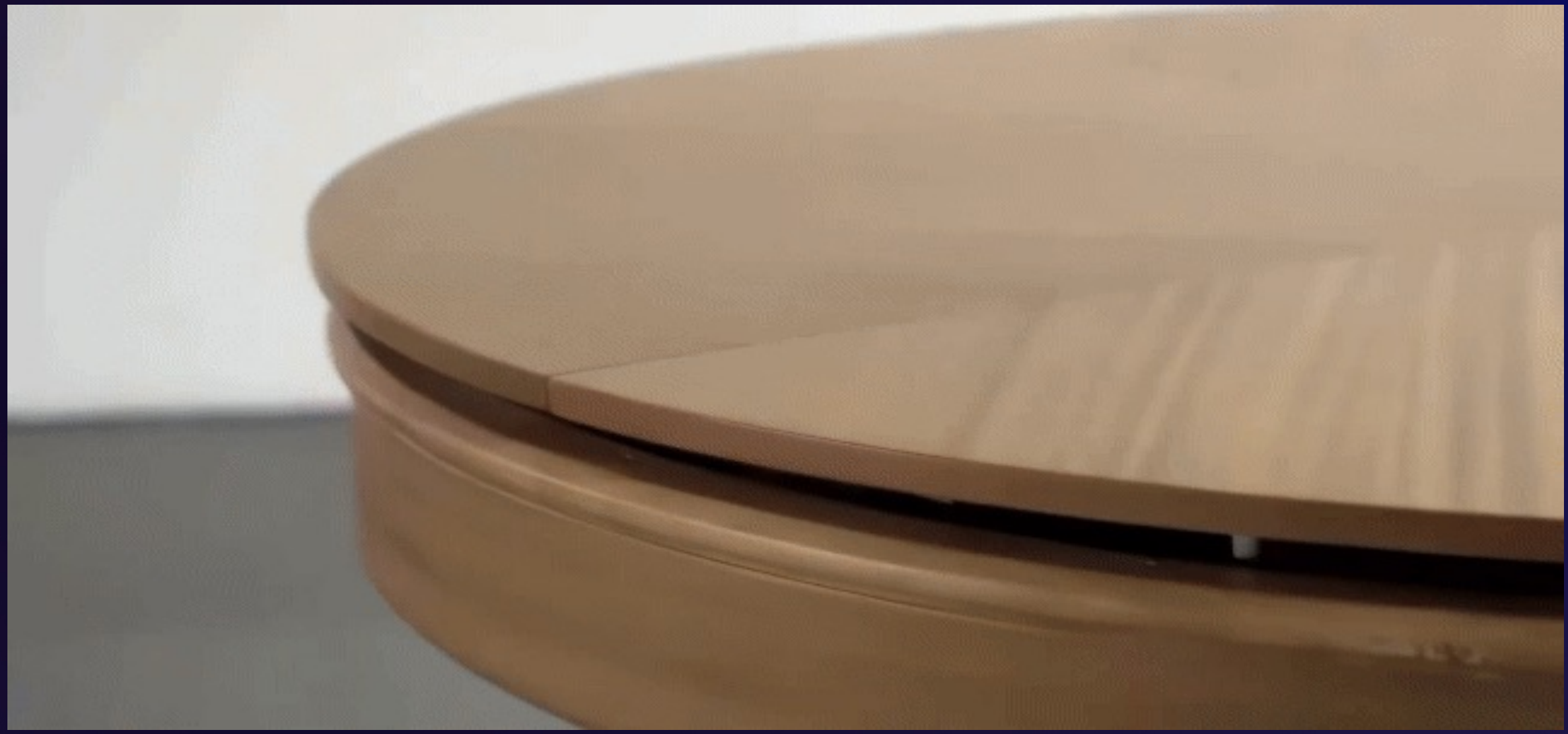
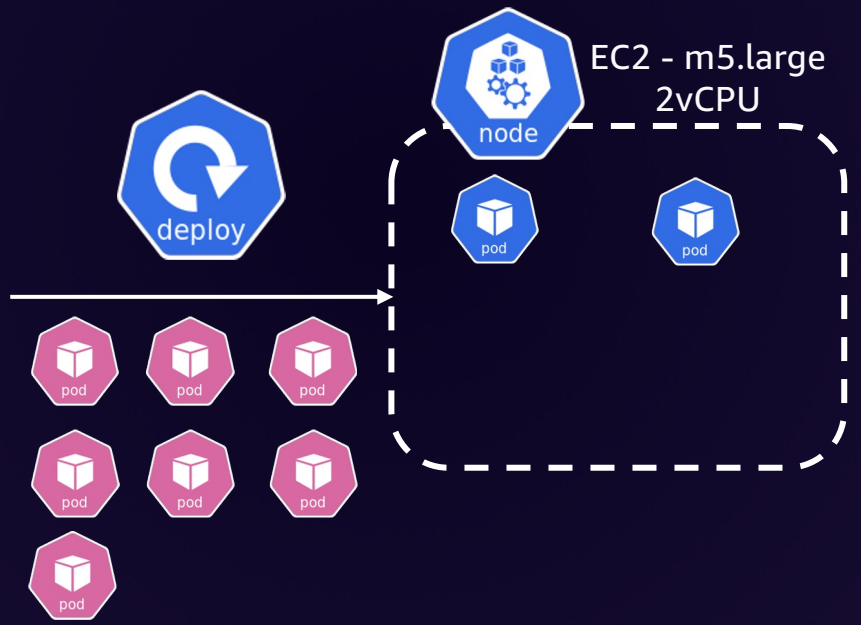
Harness the power of Karpenter to scale, optimize & upgrade Kubernetes

Chakkree Tipsupa
Solutions Architect
Amazon Web Services

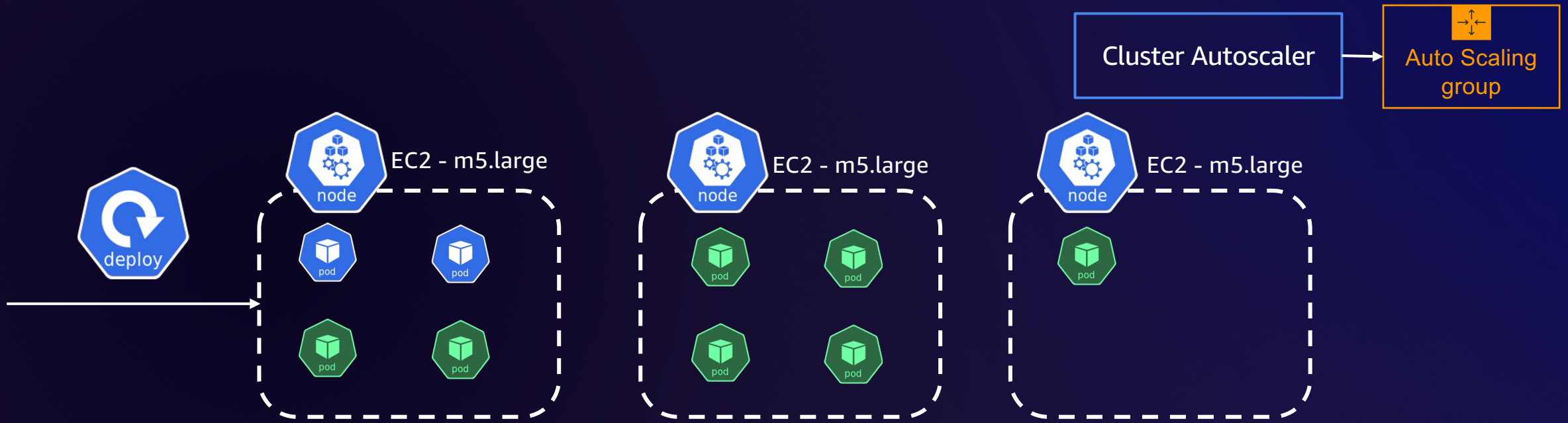


https://www.gadgetify.com/compass-round-expandable-table/

Amazon EKS with Cluster Autoscaler

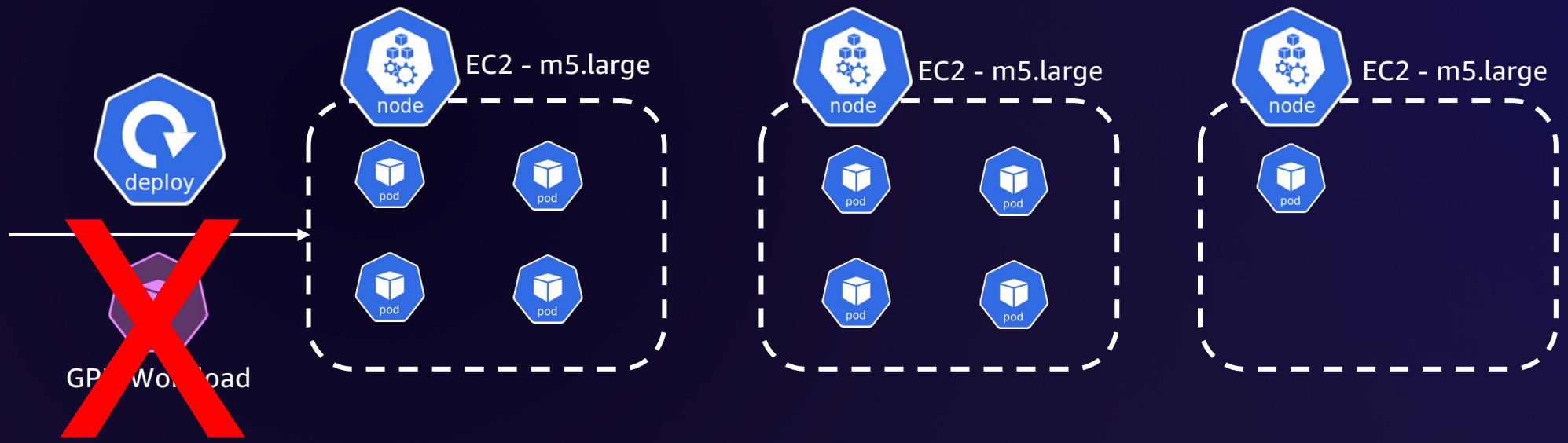


Cluster Autoscaler + ASG Challenges (1)

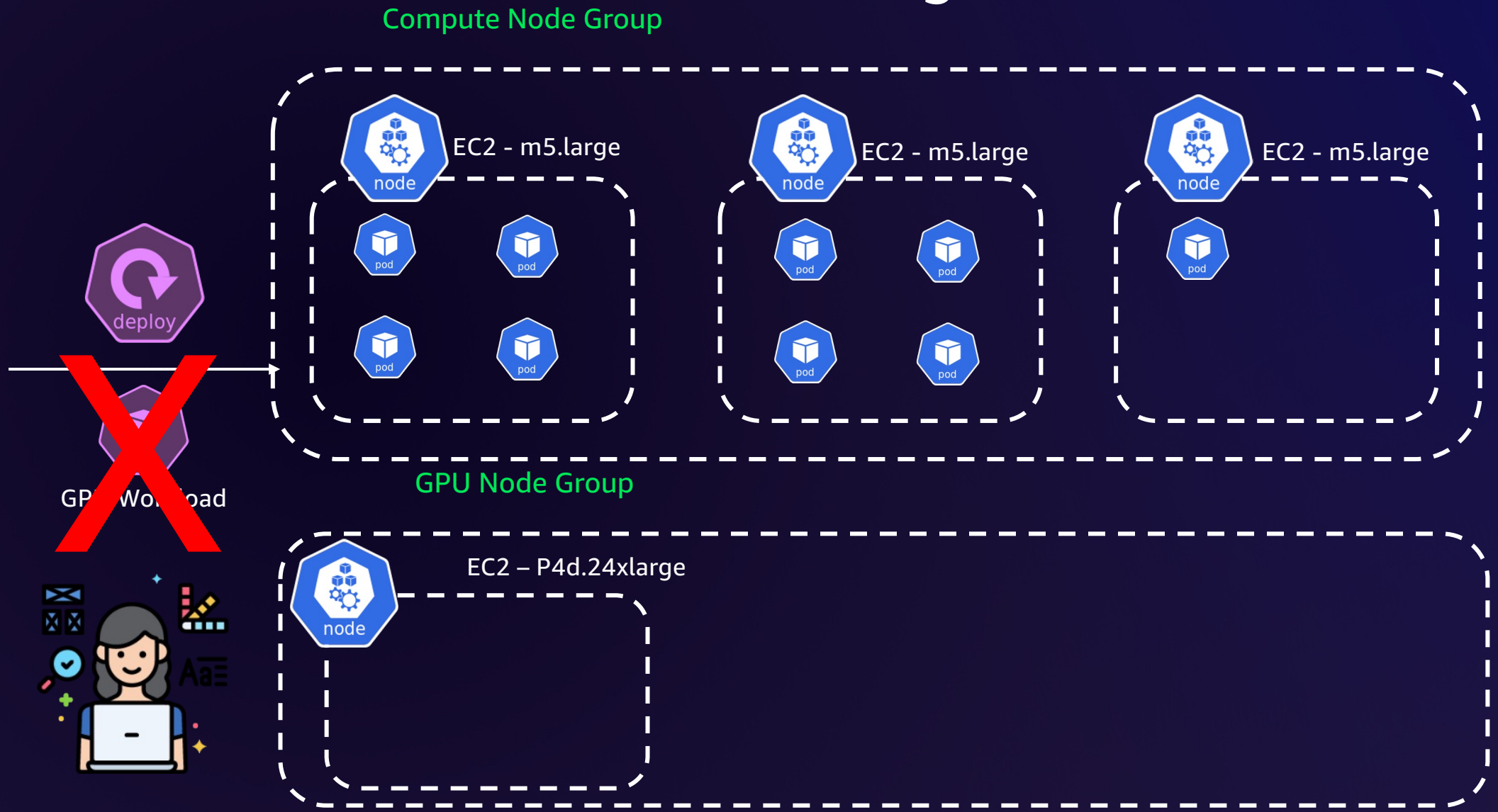


- Takes longer to spin up these multiple instances (CA to ASG to EC2 API)
- Resource underutilization

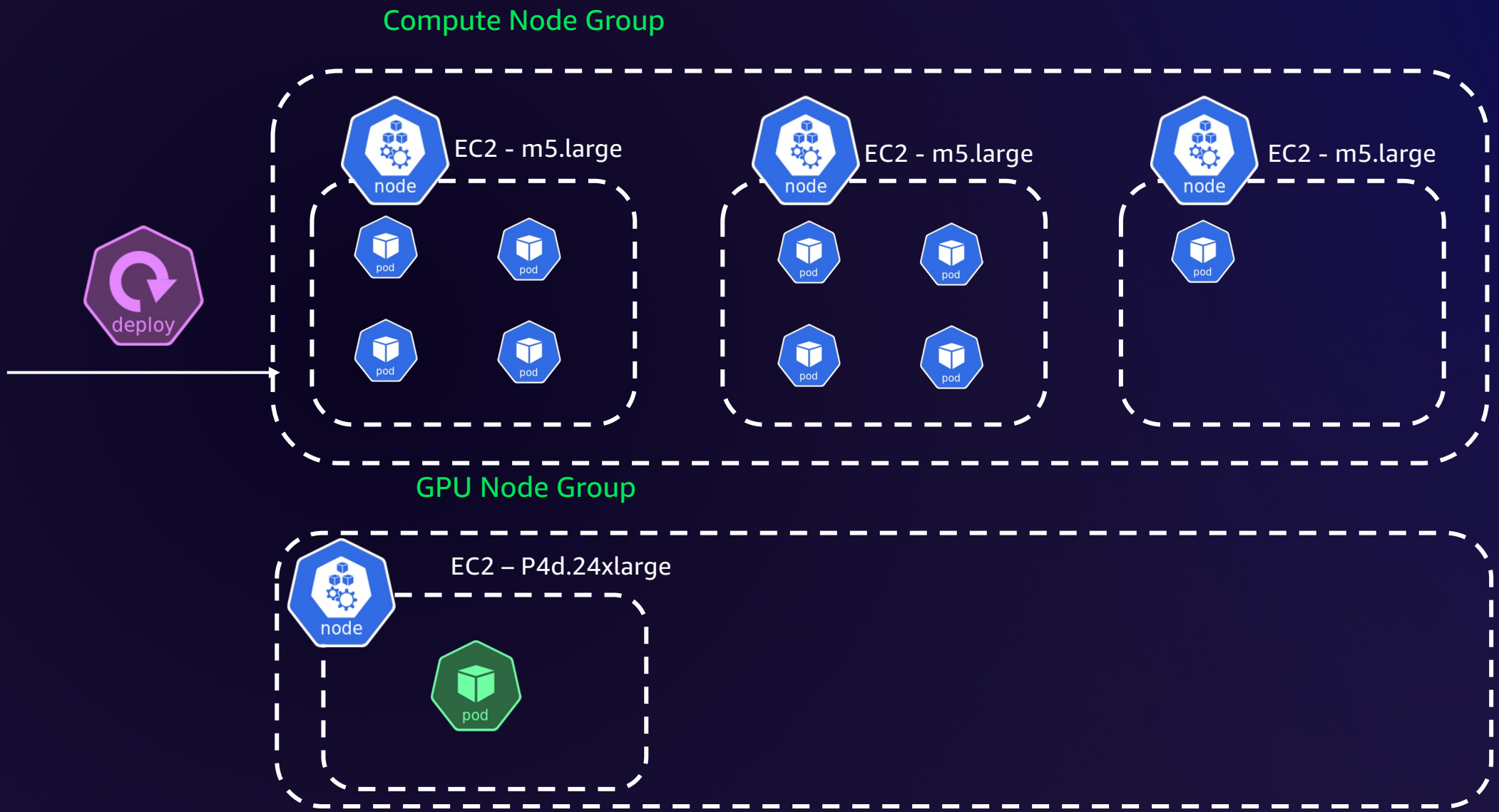
Cluster Autoscaler + ASG Challenges (2)



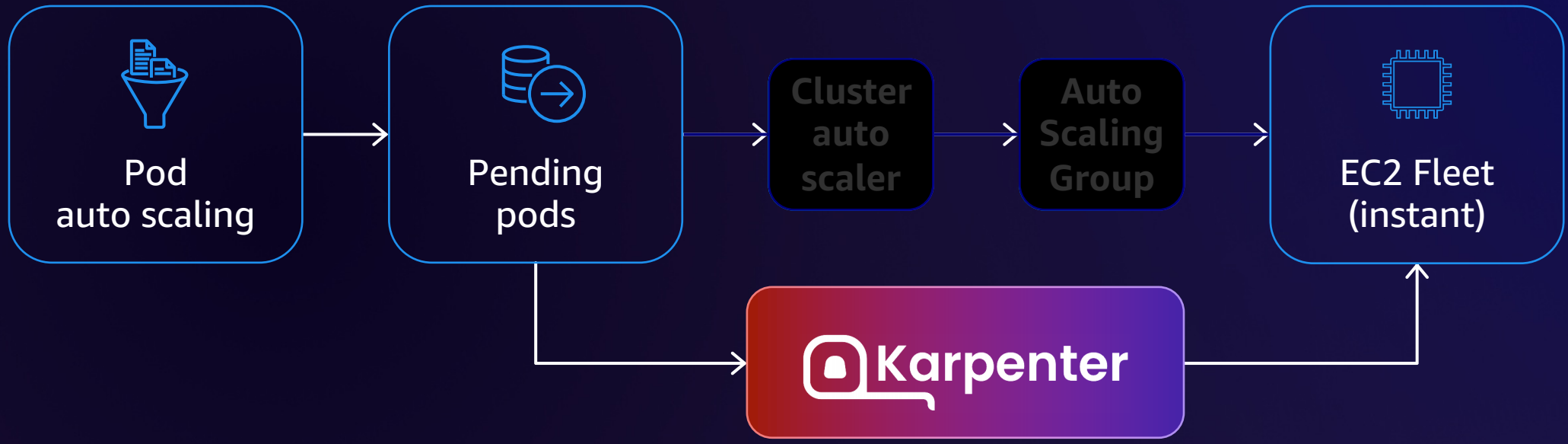
Cluster Autoscaler + ASG Challenges (2)



Cluster Autoscaler + ASG Challenges (2)



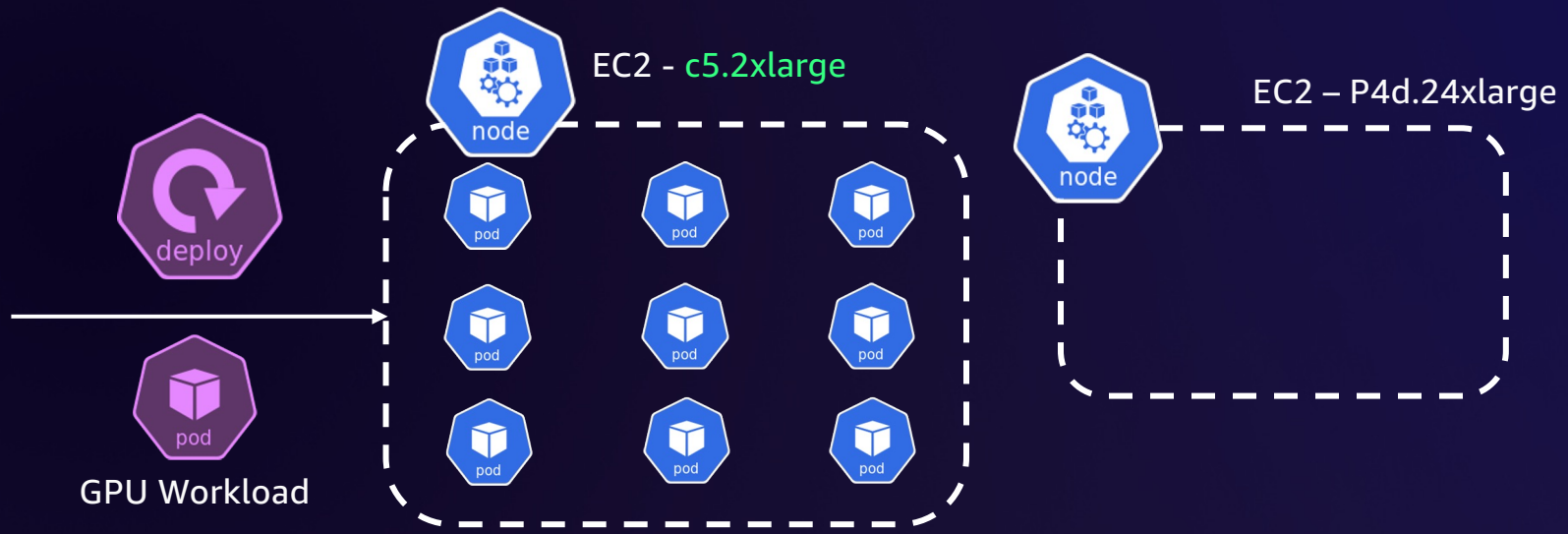
How Karpenter works



Amazon EKS with Karpenter

Amazon EKS

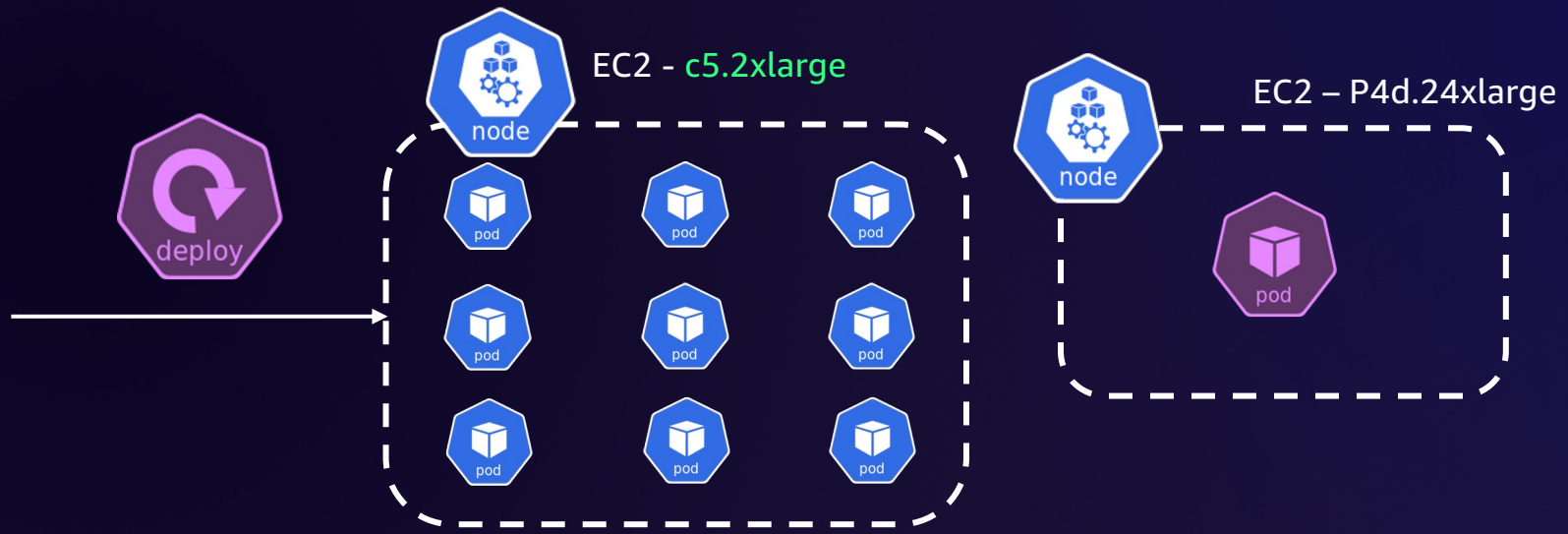
Karpenter



Amazon EKS with Karpenter

Amazon EKS

Karpenter



```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      requirements:
      - key: karpenter.k8s.aws/instance-family
        operator: In
        values: ["c5","m5","r5"]
      - key: karpenter.k8s.aws/instance-size
        operator: NotIn
        values: ["nano","micro","small"]
      - key: topology.kubernetes.io/zone
        operator: In
        values: ["us-west-2a","us-west-2b"]
      - key: kubernetes.io/arch
        operator: In
        values: ["amd64","arm64"]
      - key: karpenter.sh/capacity-type
        operator: In
        values: ["spot","on-demand"]
  limits:
    cpu: 100
```

Strategies for defining NodePools

Single

A single NodePool can manage compute for multiple teams and workloads

Example use cases:

- Single NodePool for a mix of Graviton and x86, while a pending pod has a requirement for a specific processor type

Multiple

Isolating compute for different purposes

Example use cases:

- Expensive hardware
- Security isolation
- Team separation
- Different AMI
- Tenant isolation due to noisy neighbor

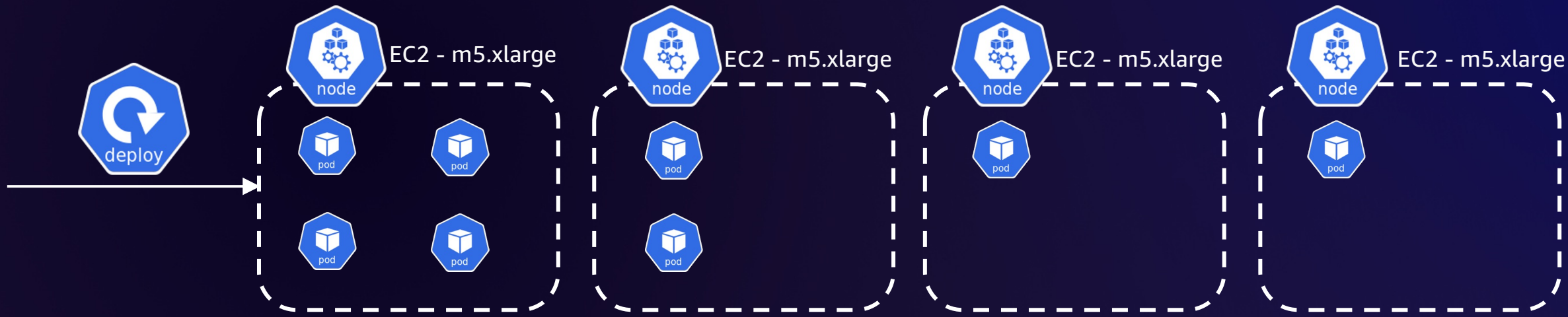
Weighted

Define order across your NodePools so that the node scheduler will attempt to scheudle with one NodePool before another

Example use cases:

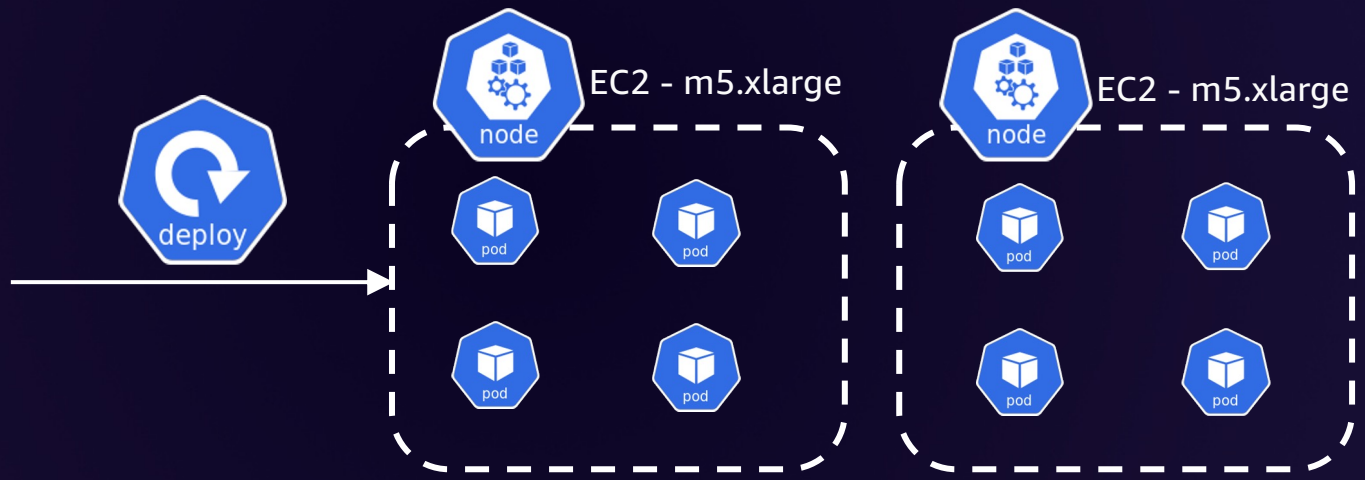
- Prioritize RI and Savings Plan ahead of other instance types
- Default clusterwide configuration
- Ratio split – Spot/OD, x86/Graviton

Karpenter optimization (1)

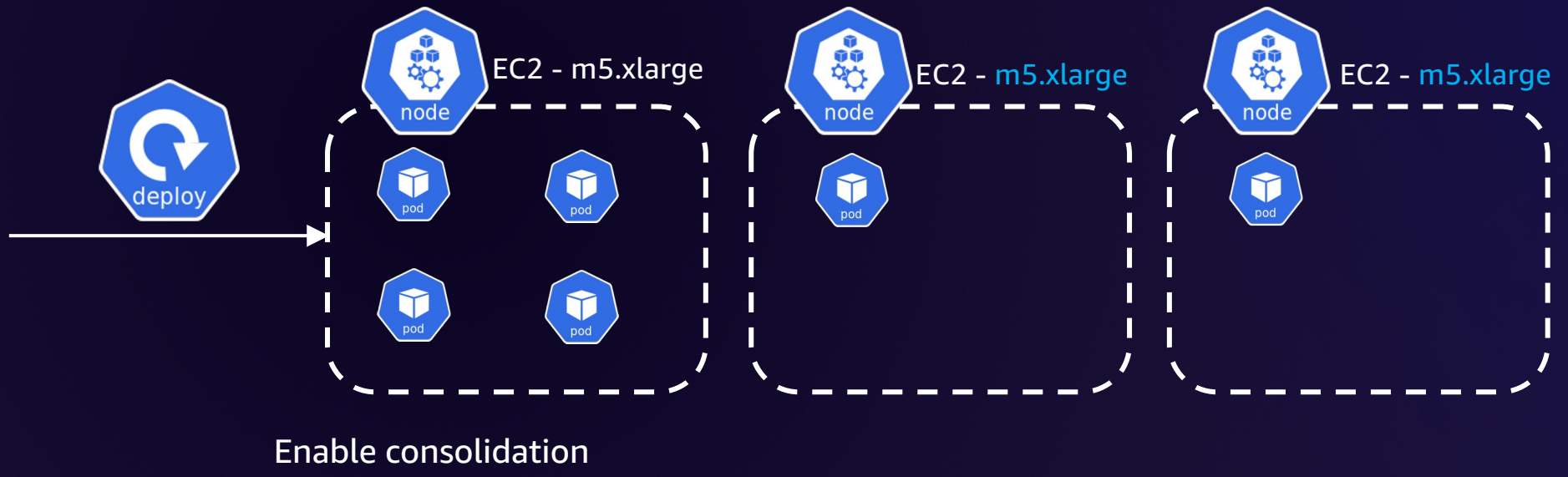



```
apiVersion: karpenter.sh/v1
kind: NodePool
spec:
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 0
```

Karpenter optimization (1)

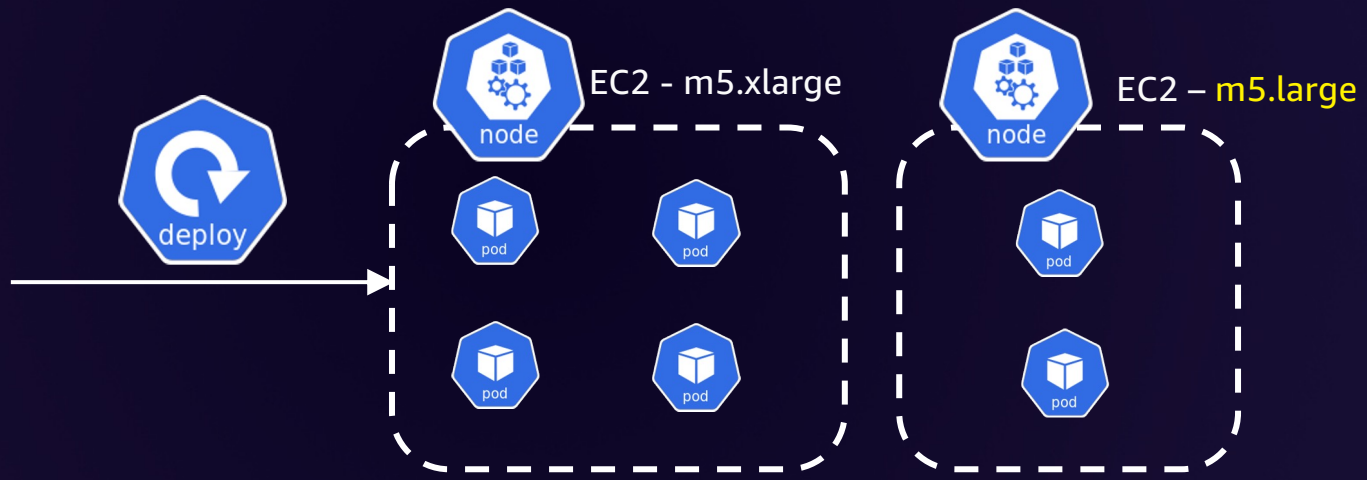


Karpenter optimization (2)



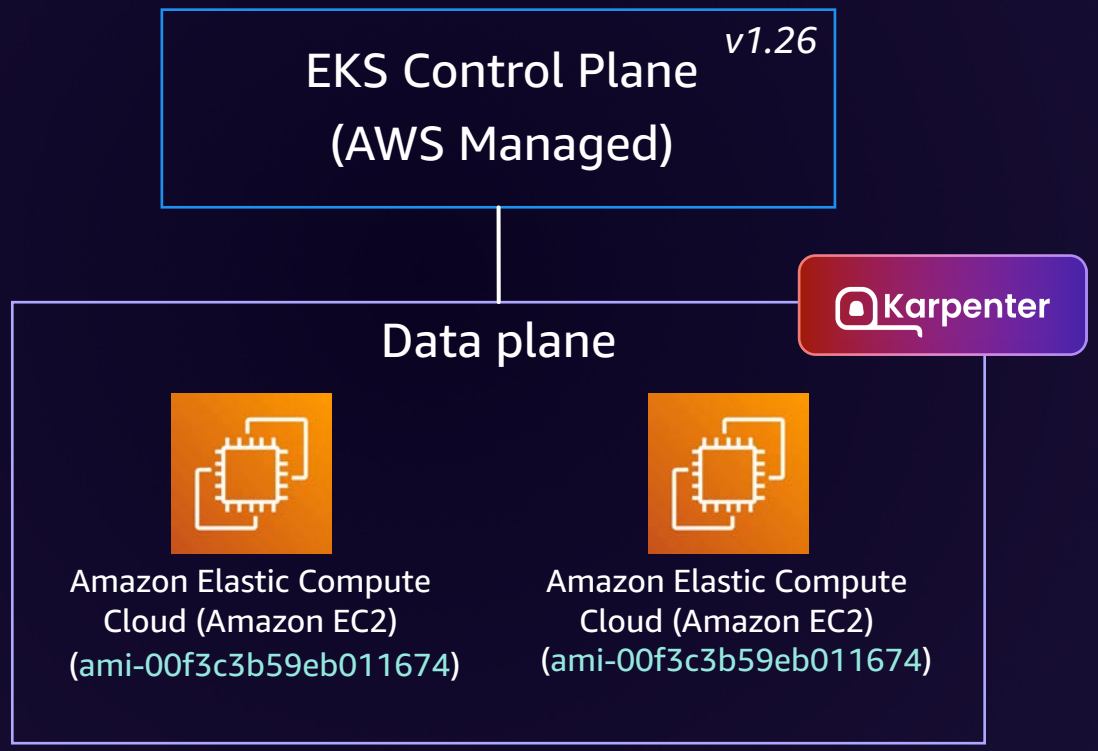
Enable consolidation

Karpenter optimization (2)



Better selection of worker nodes – reduced cost

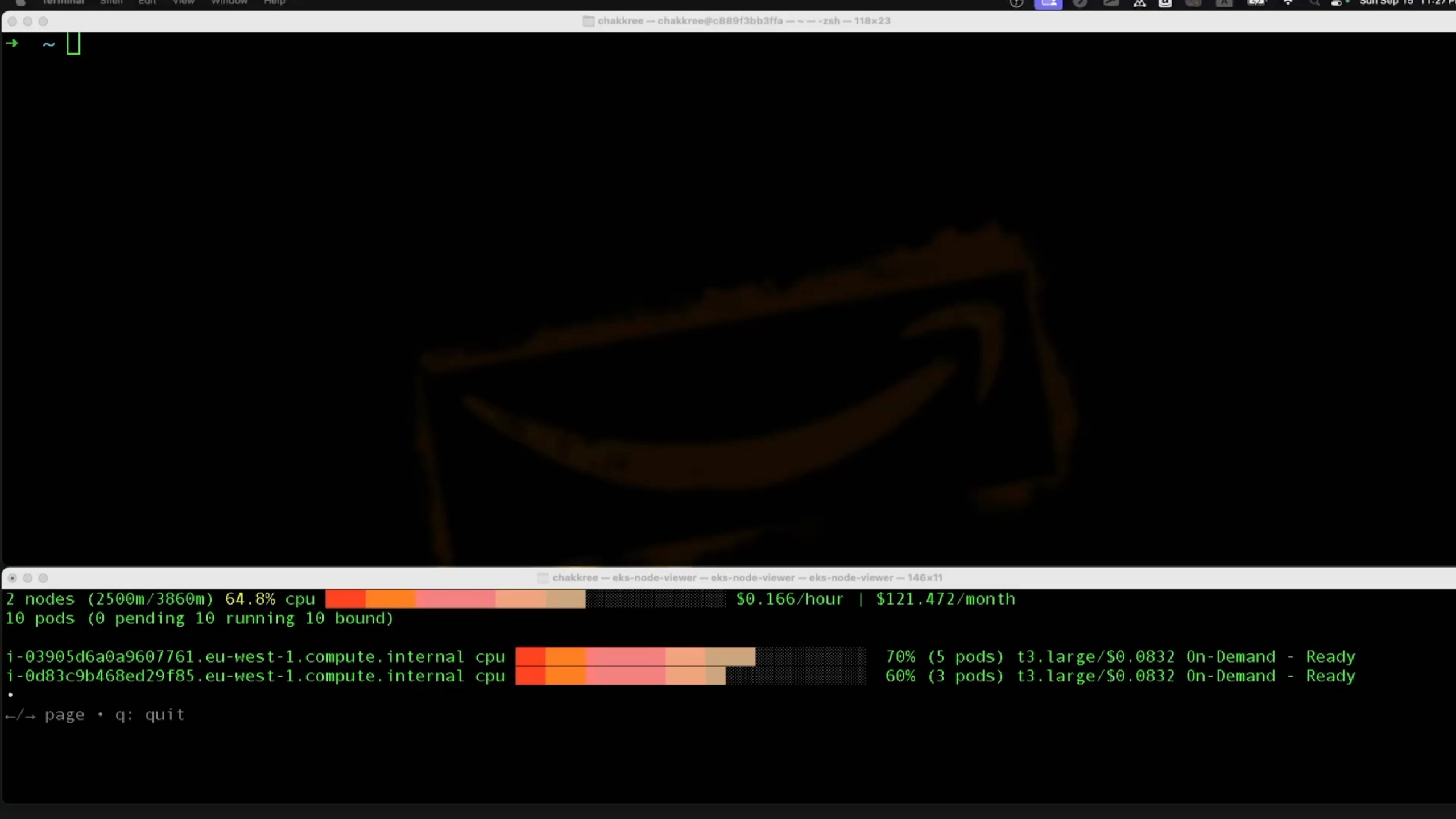
Patching/upgrading with Drift – Default AMI




```
apiVersion: karpenter.k8s.aws/v1
kind: EC2NodeClass
metadata:
  name: default
spec:
  amiSelectorTerms:
              - alias: al2@latest
```

EC2NodeClass default behavior

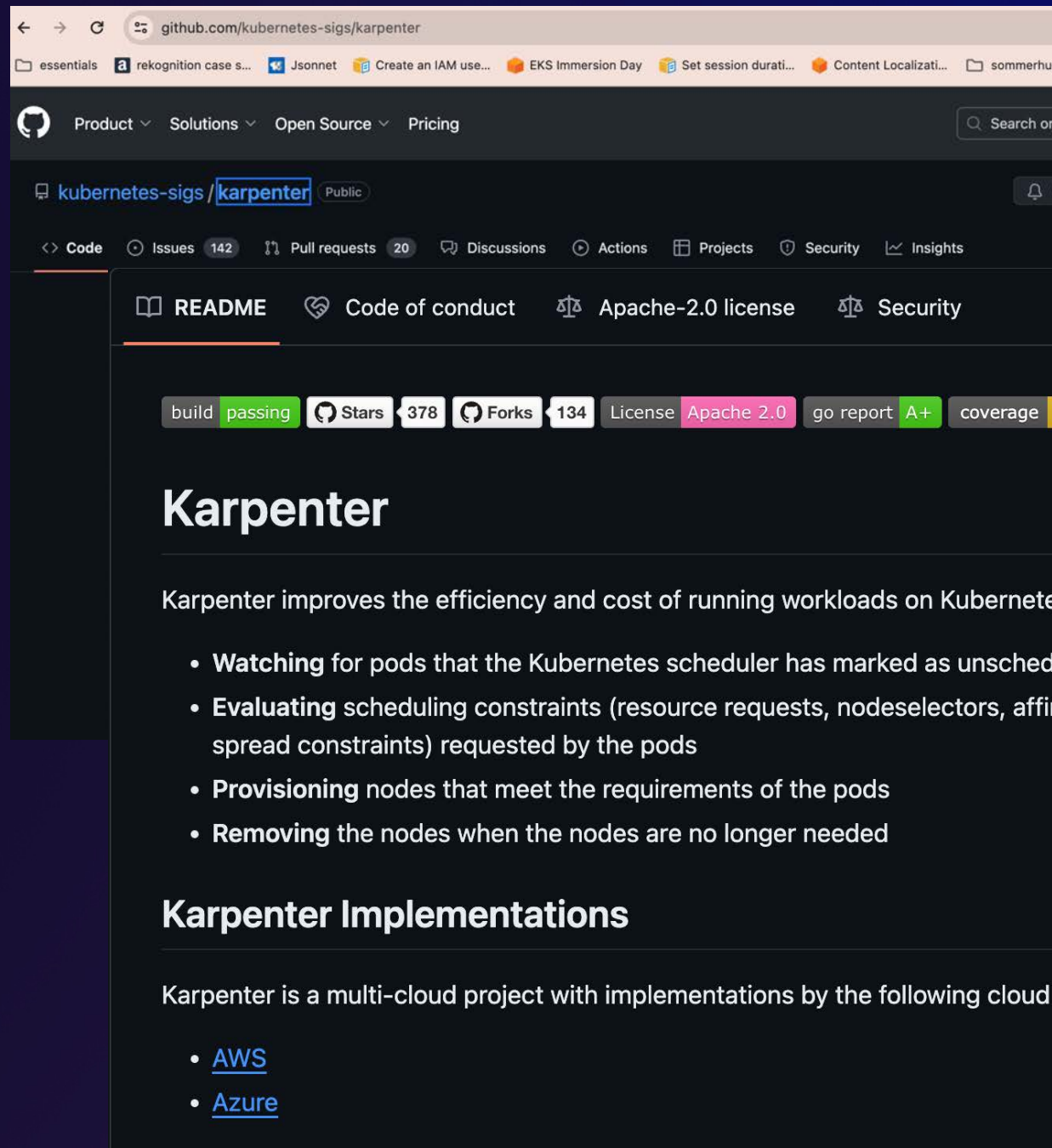
- AWS releases a new AMI for the same EKS version
- AWS updates recommended AMI in AWS Systems Manager
- Karpenter monitors the parameter store
- Karpenter updates the worker nodes AMIs automatically
- Done in rolling deployment fashion


chakkree — chakkree@c889f3bb3ffa — ~ — -zsh — 118×23
~
chakkree — eks-node-viewer — eks-node-viewer — eks-node-viewer — 146×11
2 nodes (2500m/3860m) 64.8% cpu $0.166/hour | $121.472/month
10 pods (0 pending 10 running 10 bound)
i-03905d6a0a9607761.eu-west-1.compute.internal cpu 70% (5 pods) t3.large/$0.0832 On-Demand - Ready
i-0d83c9b468ed29f85.eu-west-1.compute.internal cpu 60% (3 pods) t3.large/$0.0832 On-Demand - Ready
←/→ page • q: quit

Onboarding Karpenter

- Install Karpenter Helm chart from AWS public ECR
- Do not run Karpenter on a node that is managed by Karpenter
- Karpenter controller on Amazon EKS Fargate or on a worker node (2 node nodegroup)
- Migrating from CA
  - Step-by-step guide: Migrating from Cluster Autoscaler
  - Reuse custom AMI pipeline – update AWSNodeTemplate instead of ASG Launch Template

- Karpenter is fast
- Karpenter is simple yet powerful
- Karpenter is cost effective
- Karpenter is secure
- Karpenter is Kubernetes native
- Karpenter is part of SIG Autoscaling (OSS)
- Karpenter is AWSome!



Thank you!

Chakkree Tipsupa