Harness Karpenter : Transforming Kubernetes Clusters with Argo Workflows

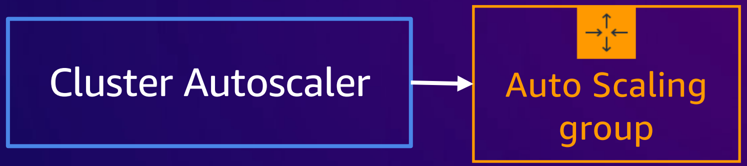
Carlos Santana

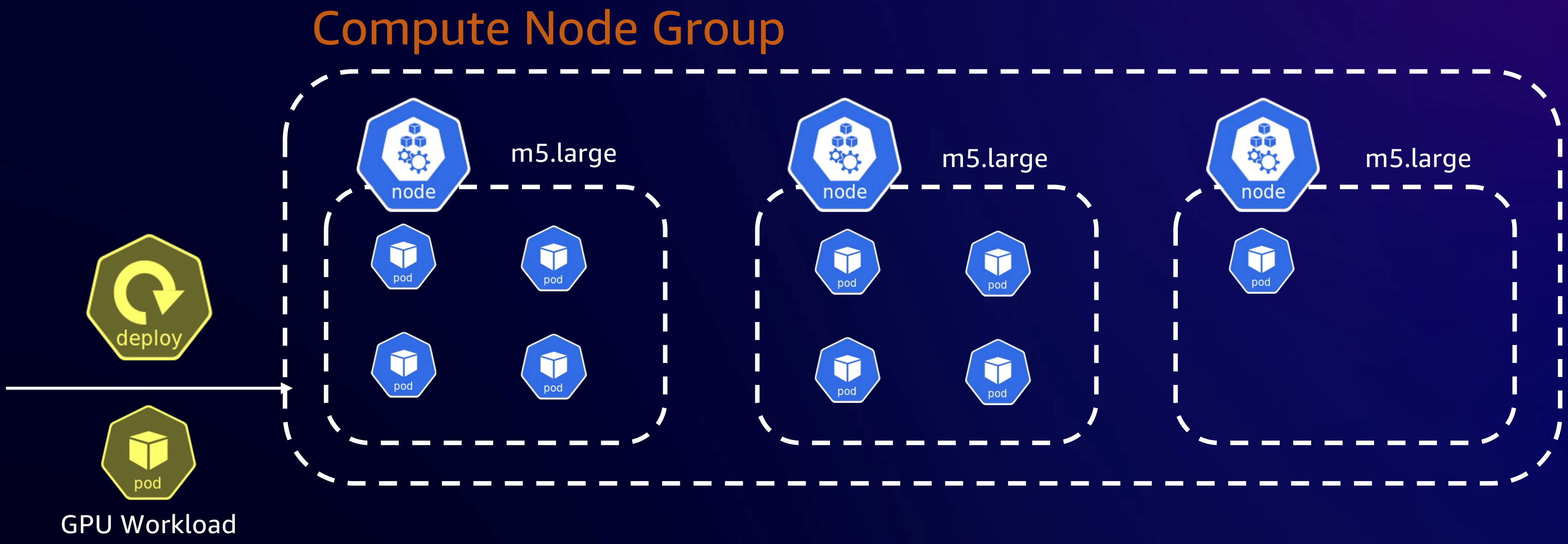
Sr. Kubernetes SA
AWS

Rajdeep Saha

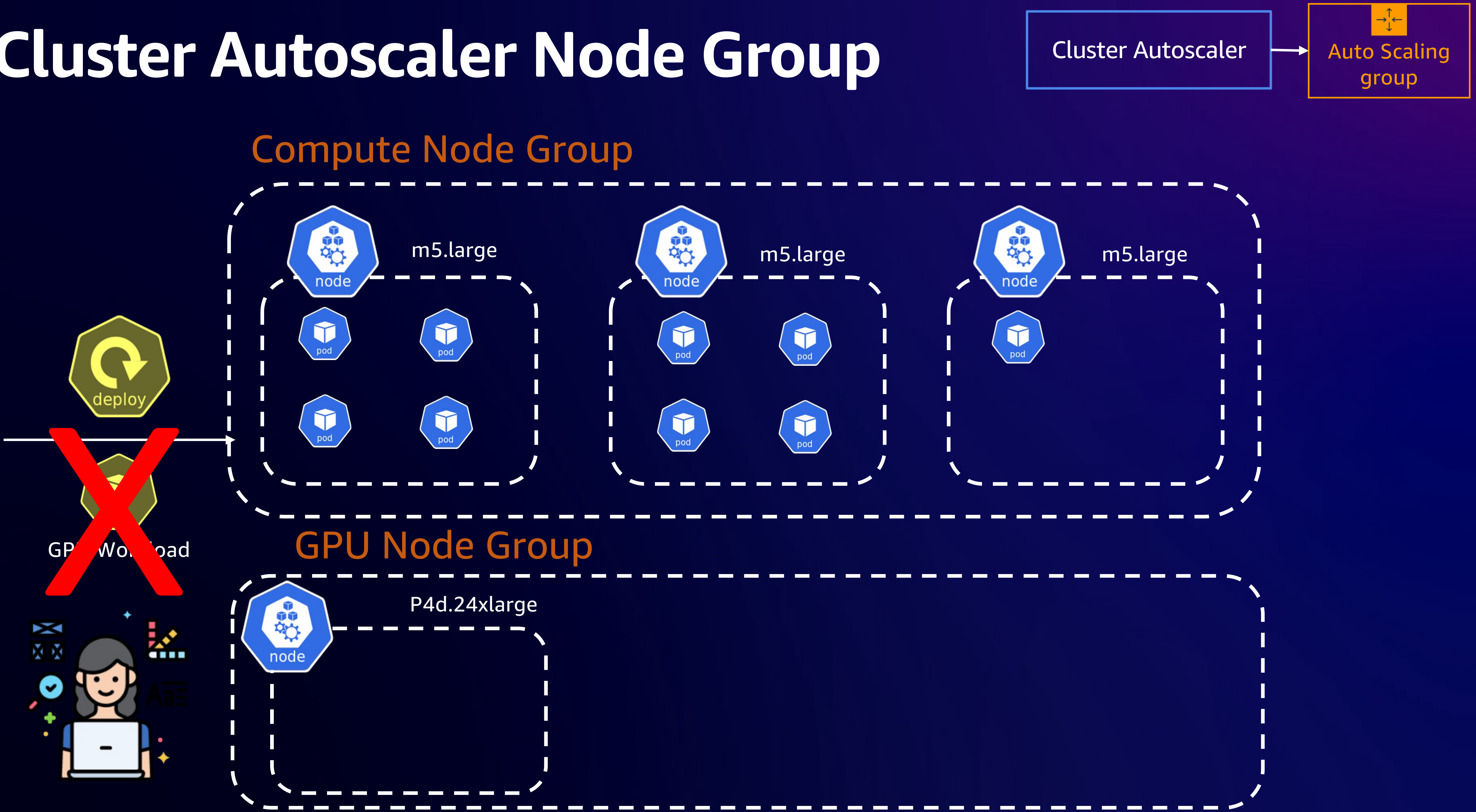
Principal Solutions Architect – Containers/Serverless
AWS

Cluster Autoscaler Challenges

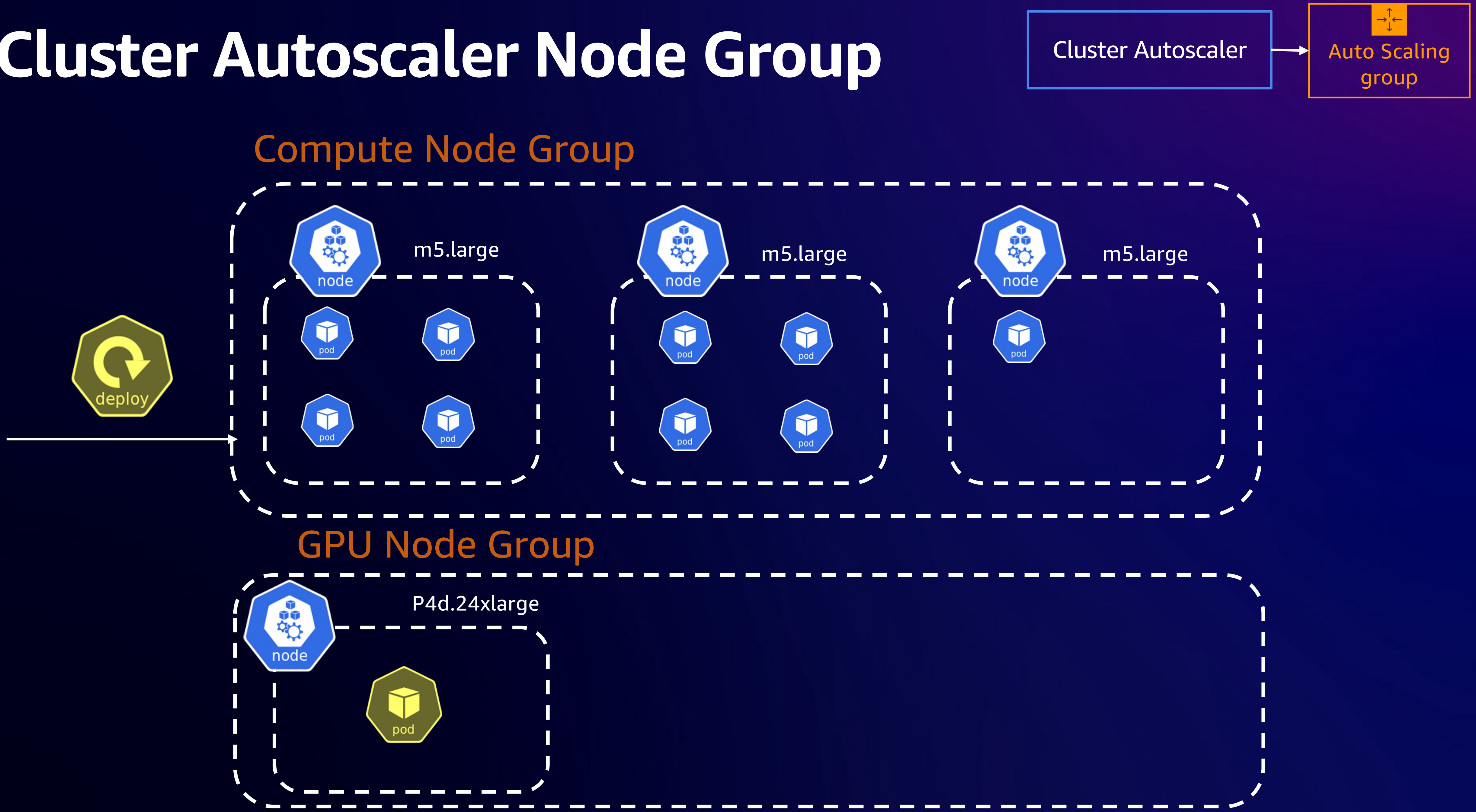




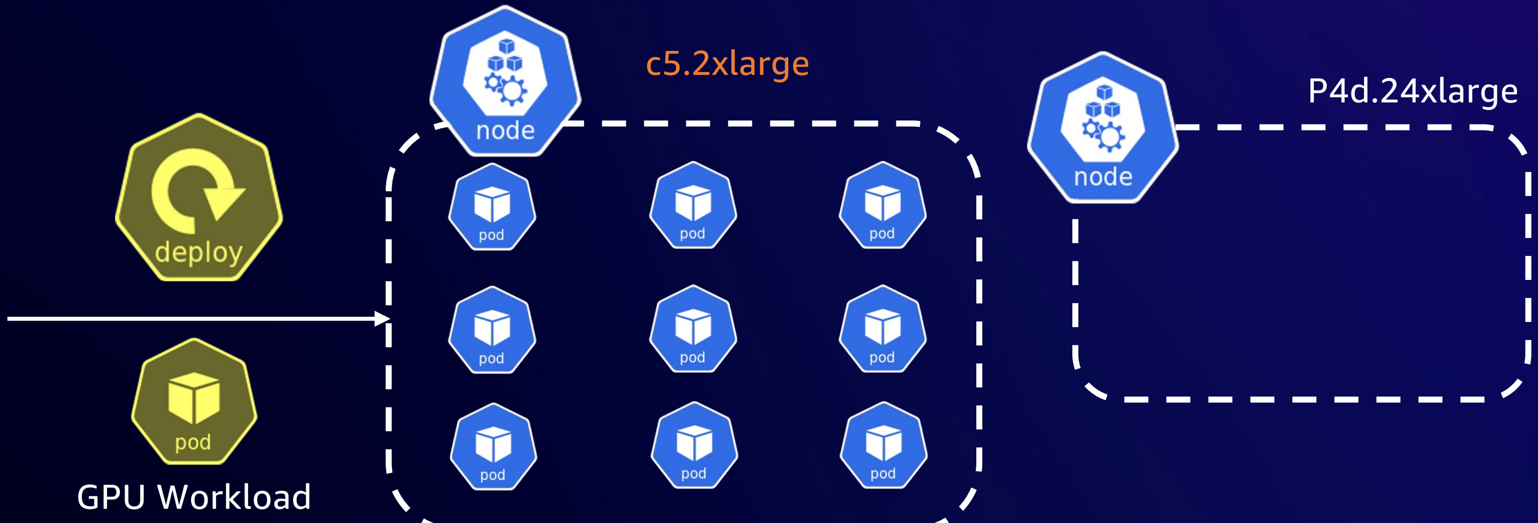
Cluster Autoscaler Node Group



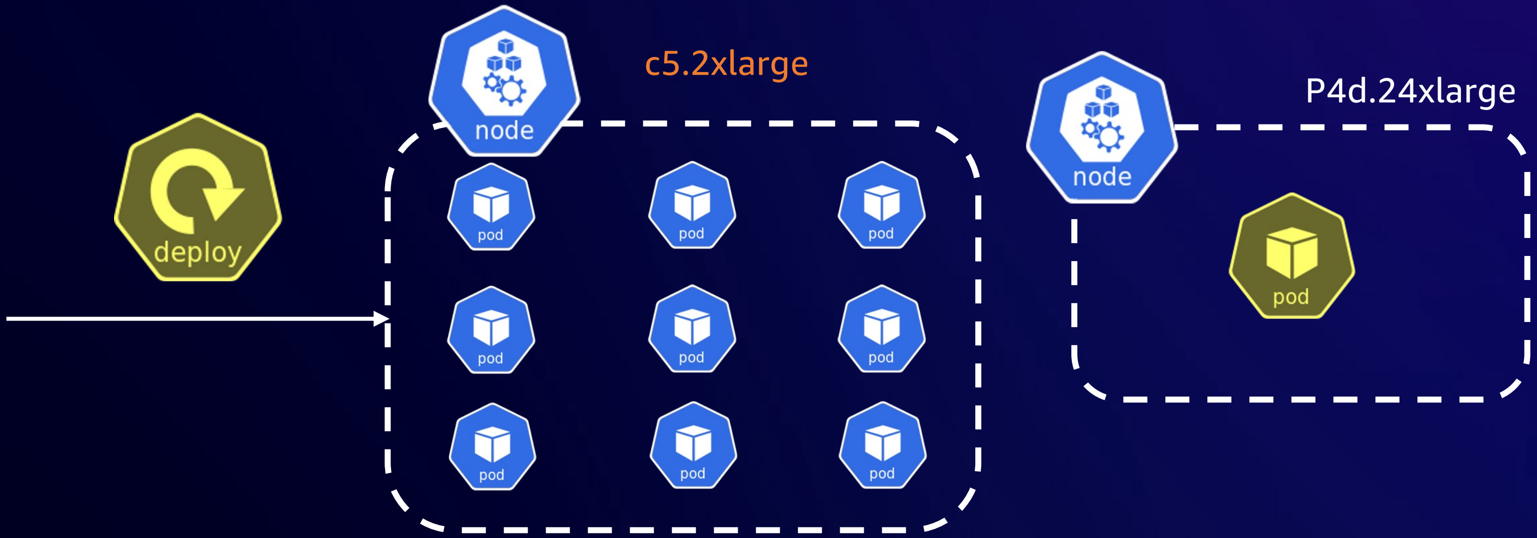
Cluster Autoscaler Node Group



Karpenter – CNCF SIG Autoscaling Project

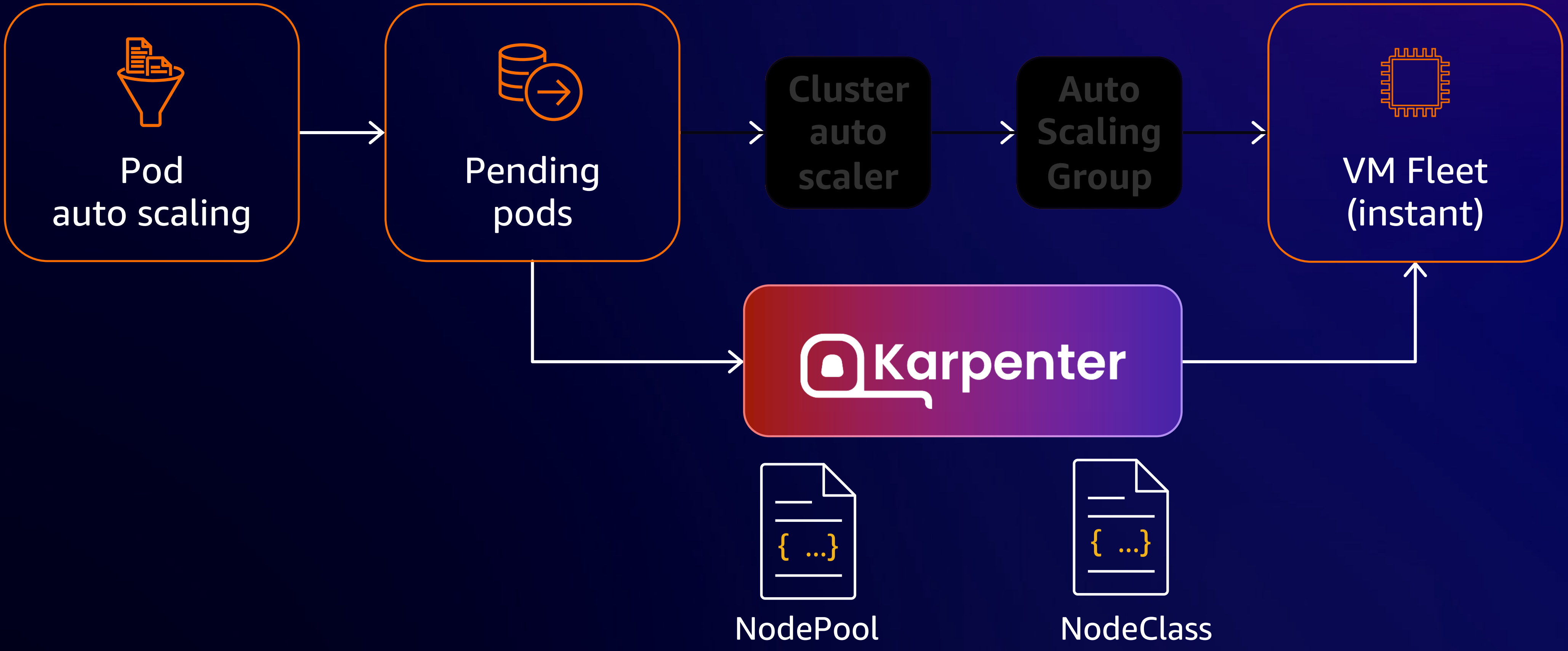


Karpenter – CNCF SIG Autoscaling Project



- Provision appropriate instances based on podspec without separate nodegroups
- Faster than Cluster Autoscaler

How Karpenter works



Compute flexibility

Instance type flexibility

- Attribute-based requirements → sizes, families, generations, CPU architectures
- No list → picks from all instance types
- Limits how many VM instances this NodePool can provision
- Prioritizes cost

AZ flexibility

- Provision in any AZ
- Provision in specified AZs

```yaml
apiVersion: karpenter.sh/v1beta1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      requirements:
        - key: karpenter.k8s.aws/instance-category
          operator: In
          values: ["c","m","r","t"]
        - key: karpenter.k8s.aws/instance-size
          operator: NotIn
          values: ["nano","micro","small”,"medium"]
        - key: karpenter.k8s.aws/instance-hypervisor
          operator: In
          values: ["nitro"]
        - 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
```

Karpenter works with Kubernetes scheduling

Standard K8s pod scheduling mechanisms

- Node selectors
- Node affinity
- Taints and tolerations
- Topology spread

User-defined annotation, labels, taints

```yaml
apiVersion: karpenter.sh/v1beta1
kind: NodePool
spec:
  template:
    metadata:
      annotations:
        application/name: "app-a"
      labels:
        team: team-a
    spec:
      taints:
      - key: example.com/special-taint
        value: "true"
        effect: NoSchedule
```

These taints, labels, annotations will be added to all nodes provisioned

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 name: myapp
spec:
   nodeSelector:
    team: team-a
```

Use labels to schedule pods for different apps

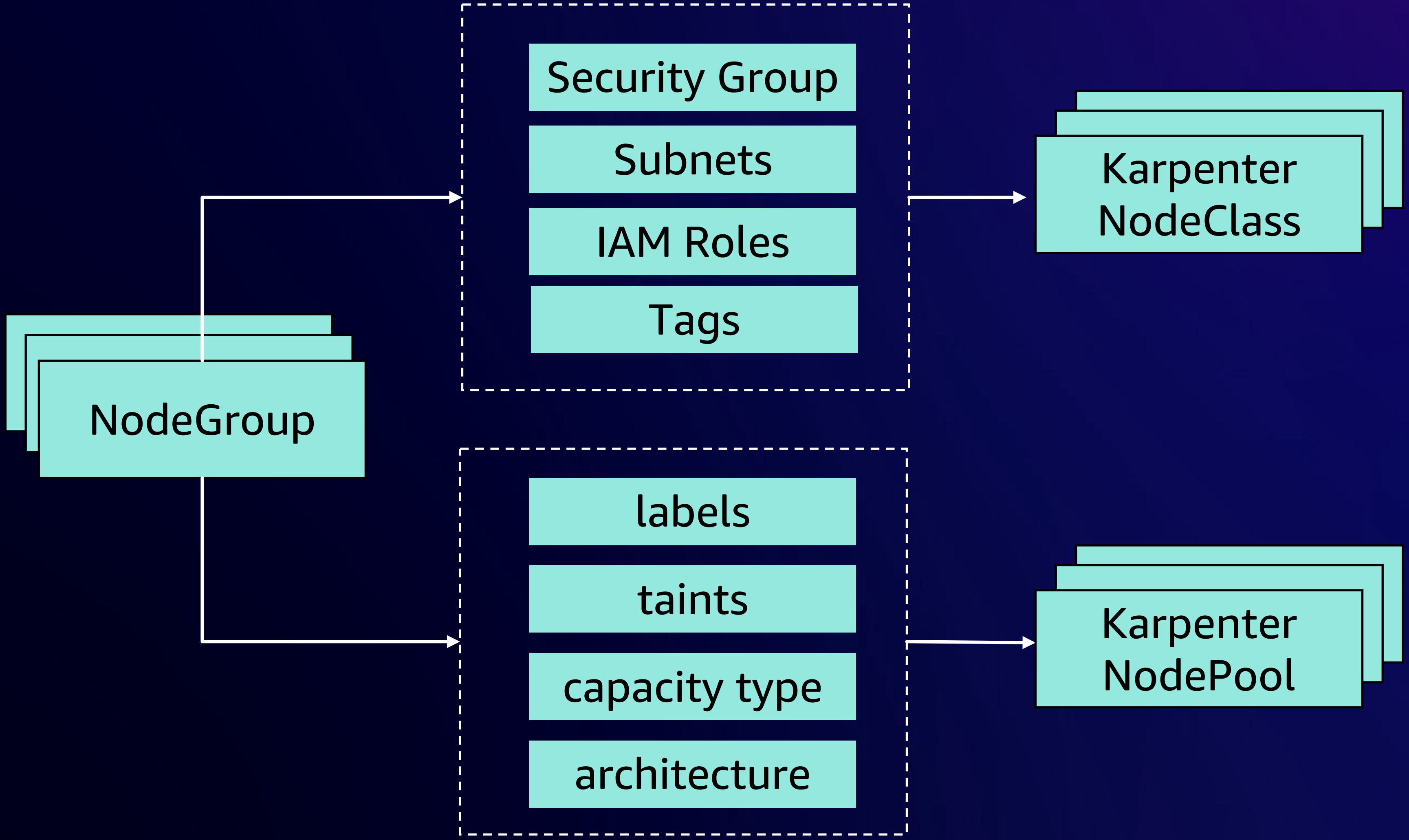
Cluster Autoscaler to Karpenter in Real World

Karpenter

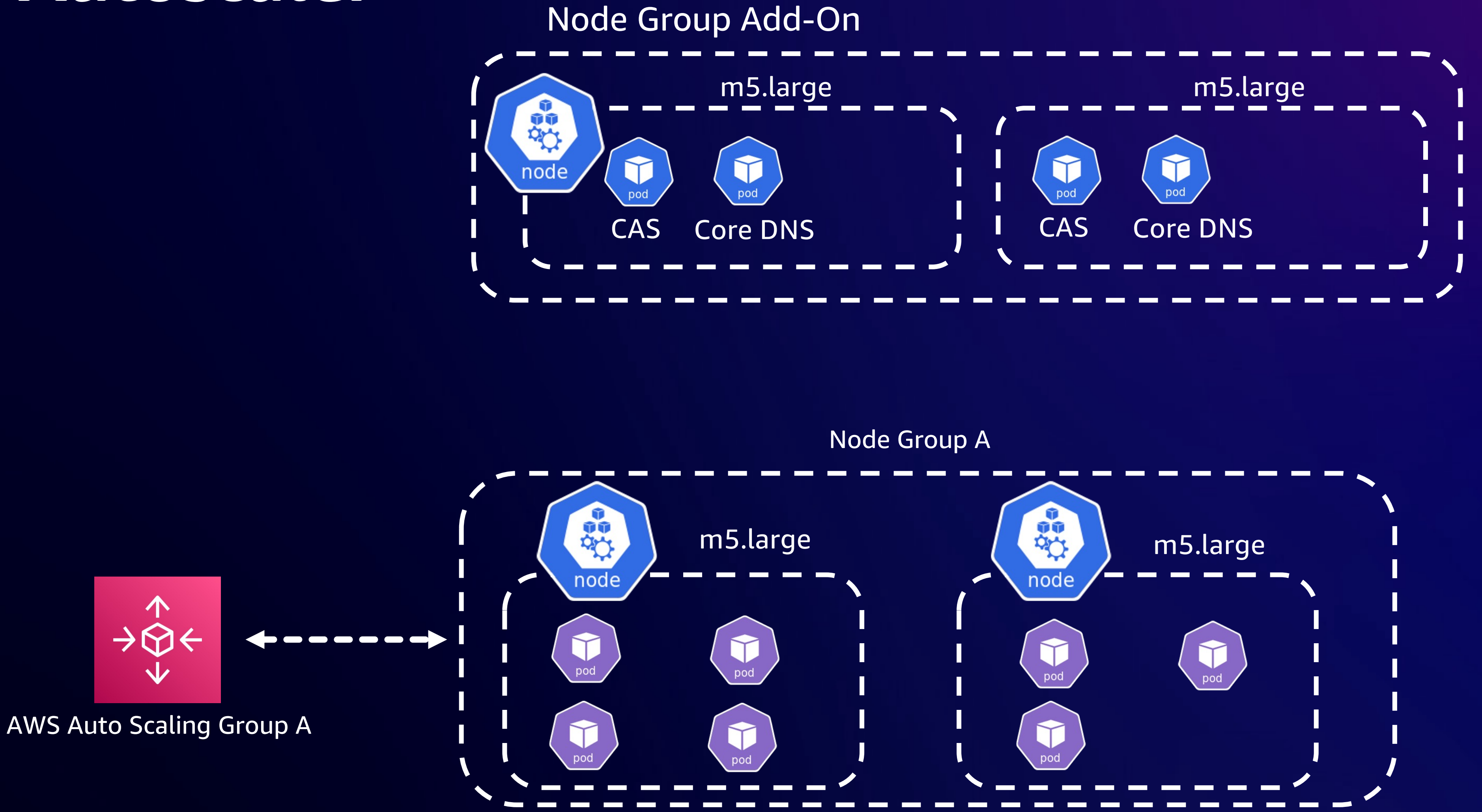
- Multi tenant cluster
- Apps will gradually move from Cluster Autoscaler to Karpenter
- Both Cluster Autoscaler and Karpenter will co-exist for separate apps
- This process also works for complete move in one go

Migrate to Karpenter

argo
workflows



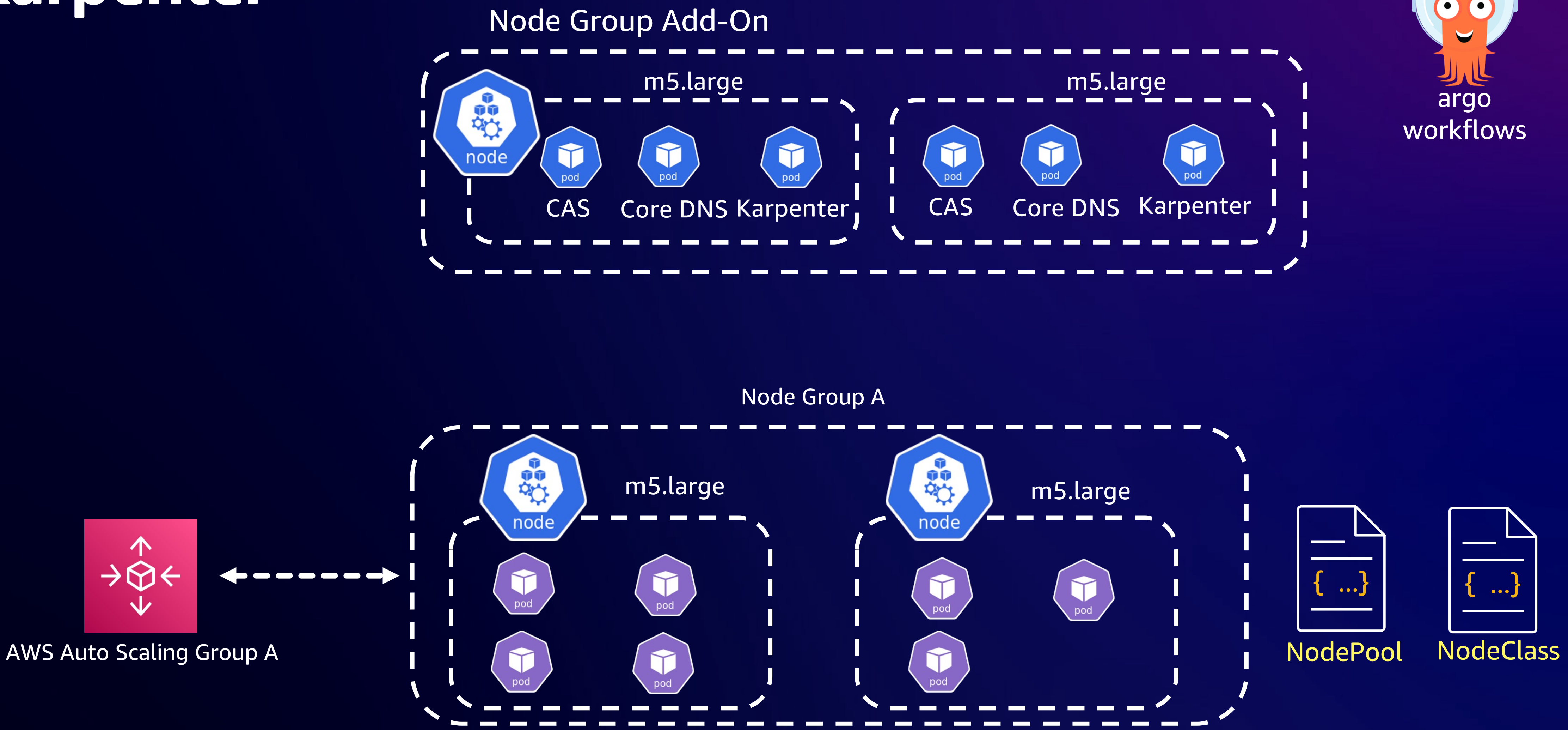
Cluster Autoscaler



argo
workflows

Enter Karpenter

argo
workflows



Node Group Add-On

m5.large

CAS

Core DNS

Karpenter

m5.large

CAS

Core DNS

Karpenter

Node Group A

m5.large

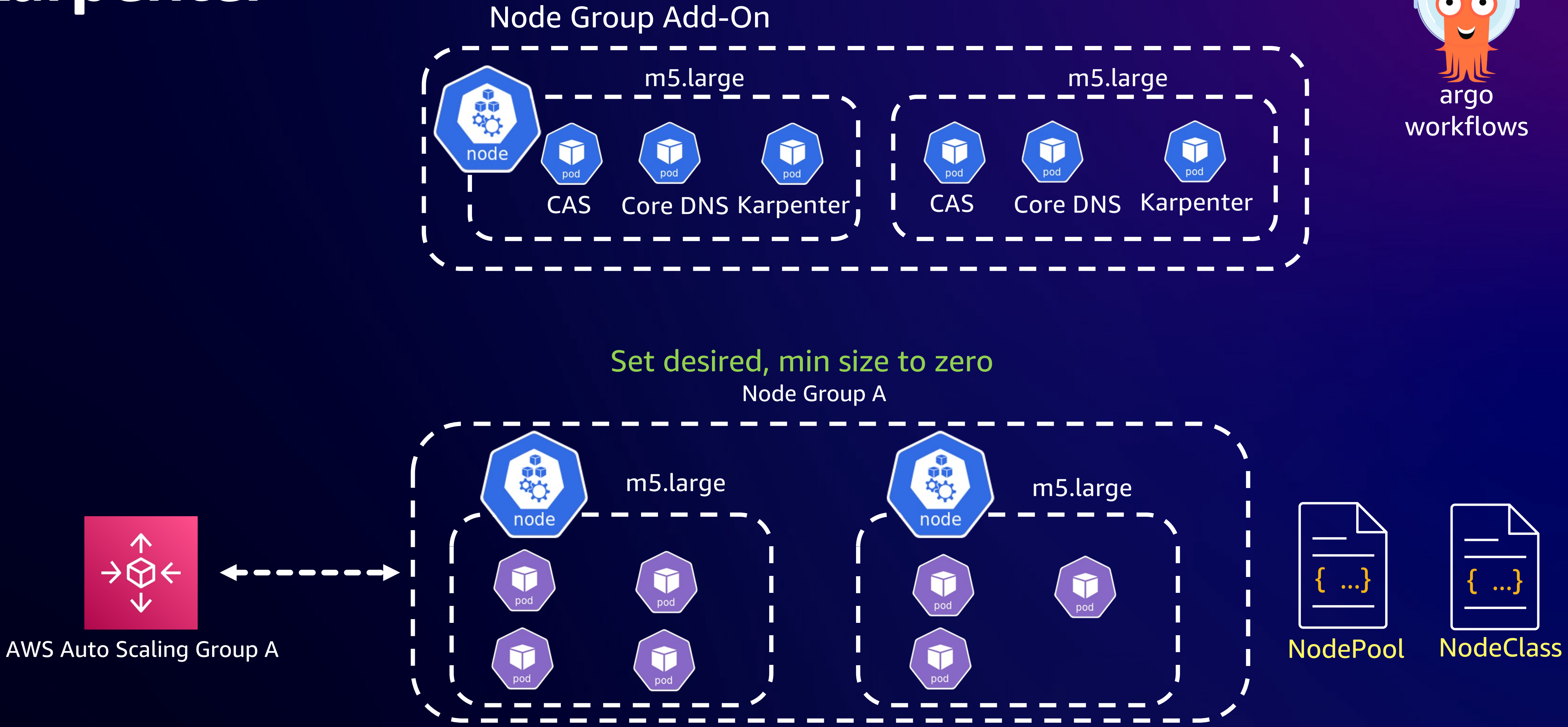
m5.large

AWS Auto Scaling Group A

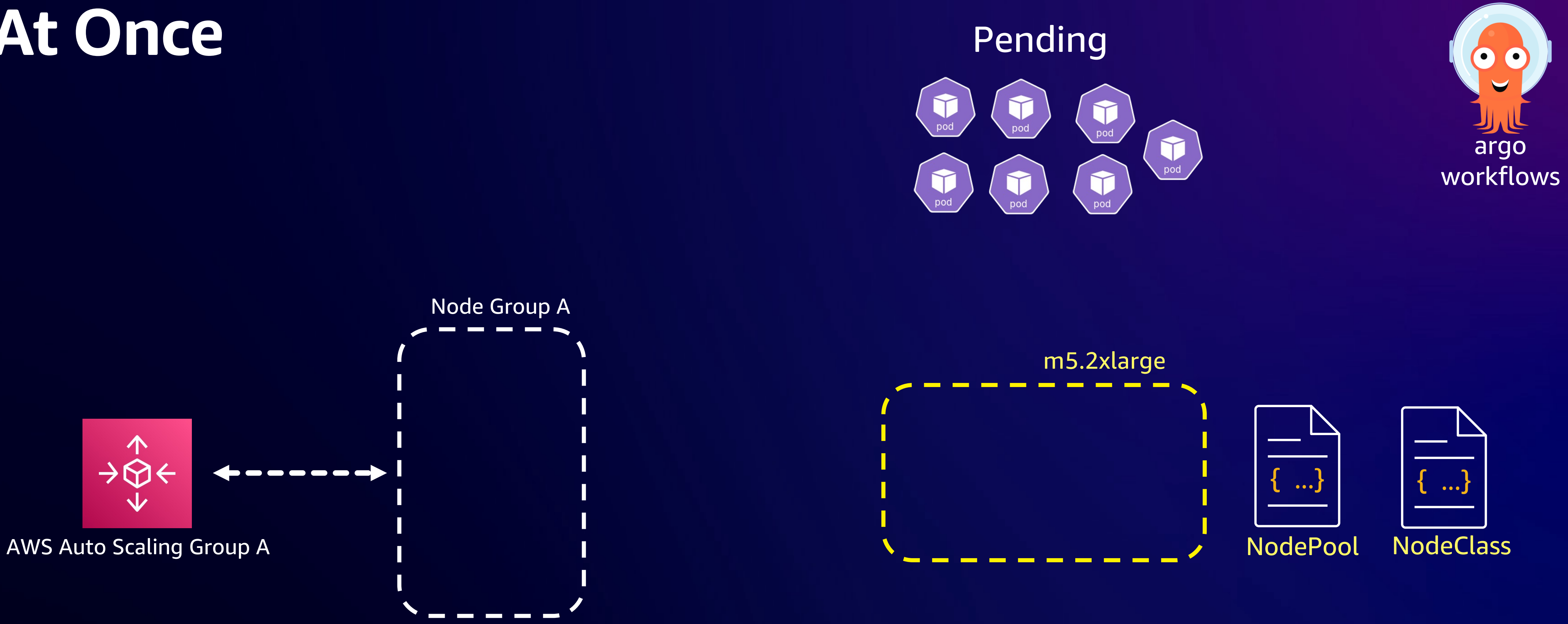
NodePool

NodeClass

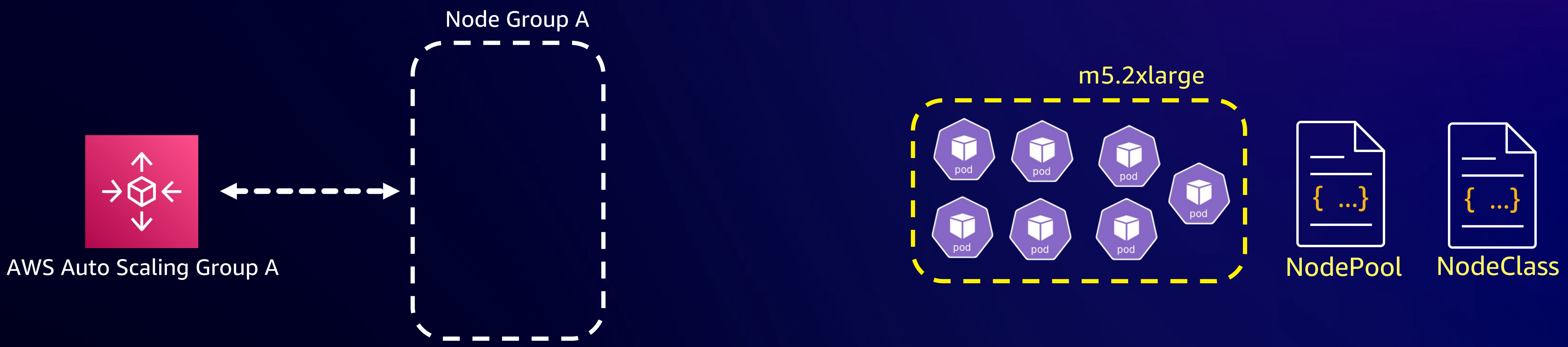
Enter Karpenter



All At Once


Pending
pod
pod
pod
pod
pod
pod
pod
pod
argo
workflows
Node Group A
AWS Auto Scaling Group A
m5.2xlarge
{ ...}
{ ...}
NodePool
NodeClass

All At Once

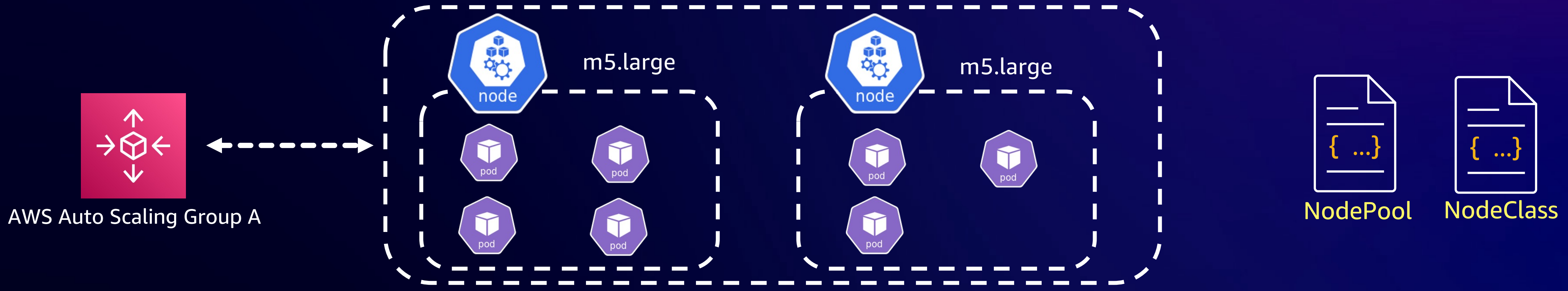


argo
workflows

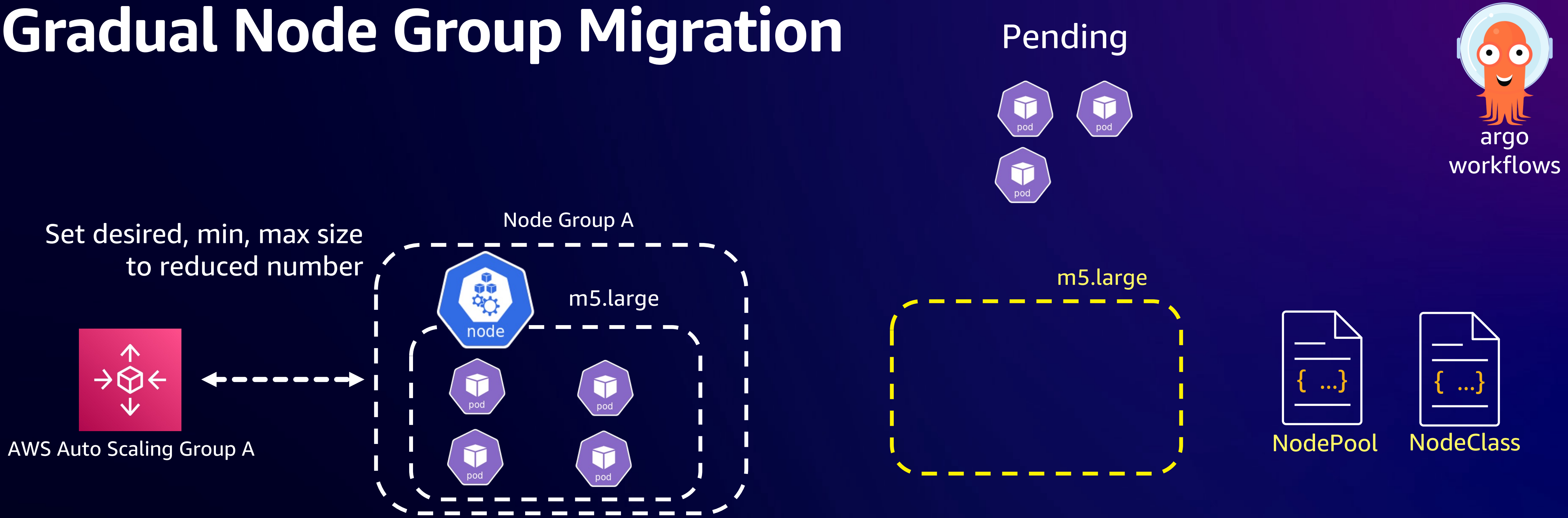
Gradual Node Group Migration

argo workflows

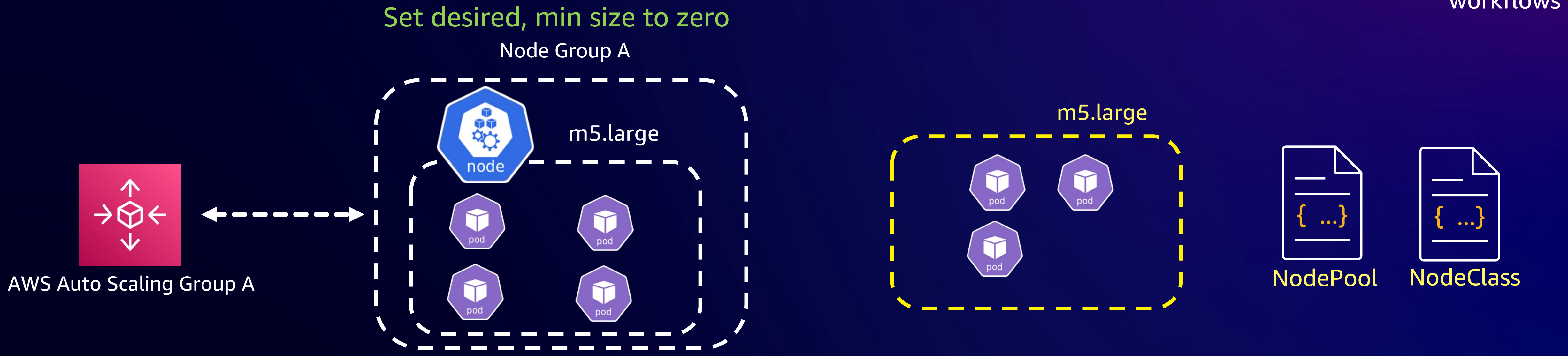
Set desired, min, max size to reduced number



Gradual Node Group Migration

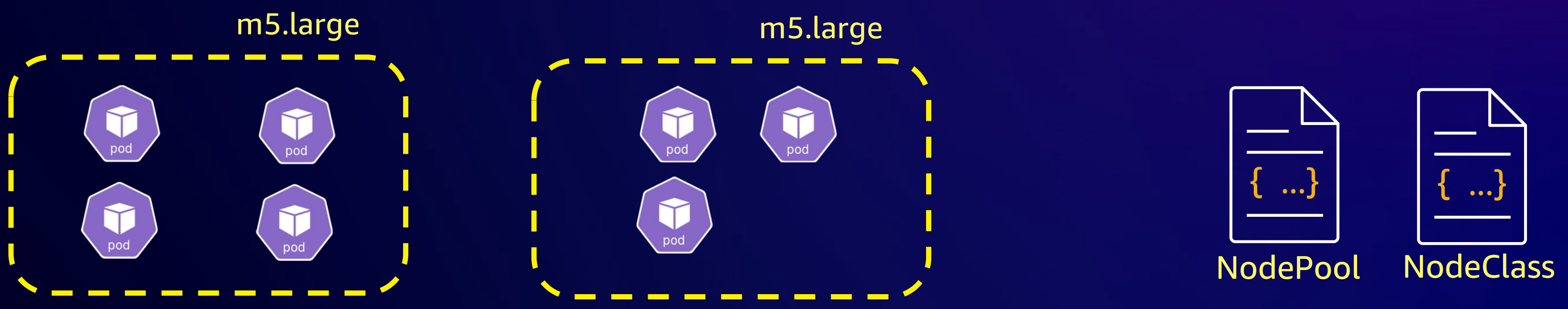


Gradual Node Group Migration



Gradual Node Group Migration

argo
workflows



Demo

argo
workflows

Thank you!

Please complete the session survey in the mobile app

Carlos Santana

csantanapr

csantanapr

csantanapr

Rajdeep Saha

cloudwithraj

cloud with raj

cloudwithraj