EKS에서 Karpenter 야무지게 활용하기

박동혁, 이호성, 황은빛

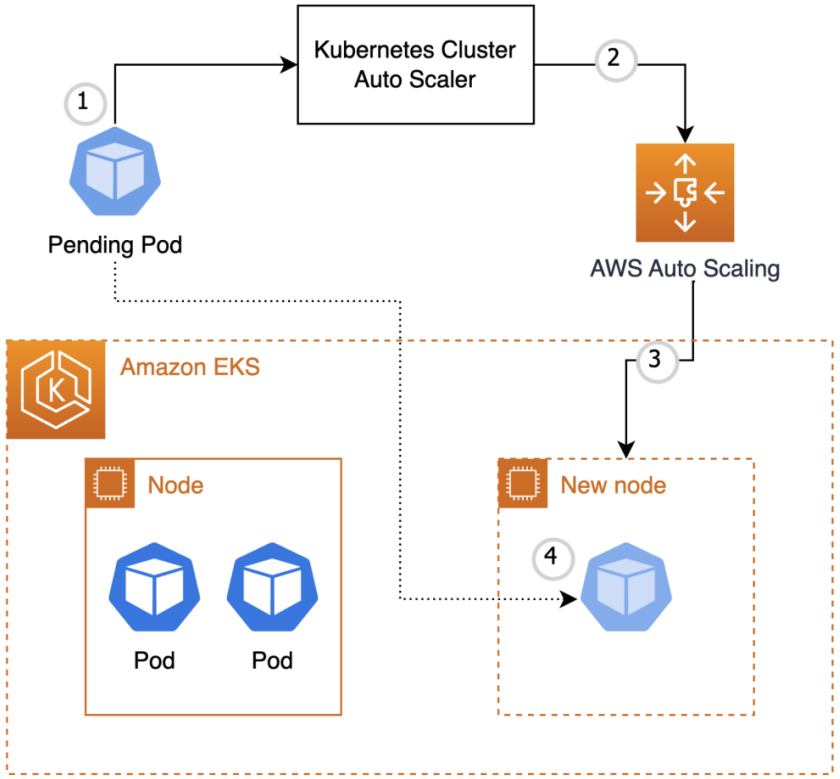
AWS Cloud Support Engineer

Agenda

- Karpenter Overview
- Karpenter CRDs
- Karpenter의 Scheduling
- Karpenter의 Disruption (중단)
- Demo & Troubleshooting

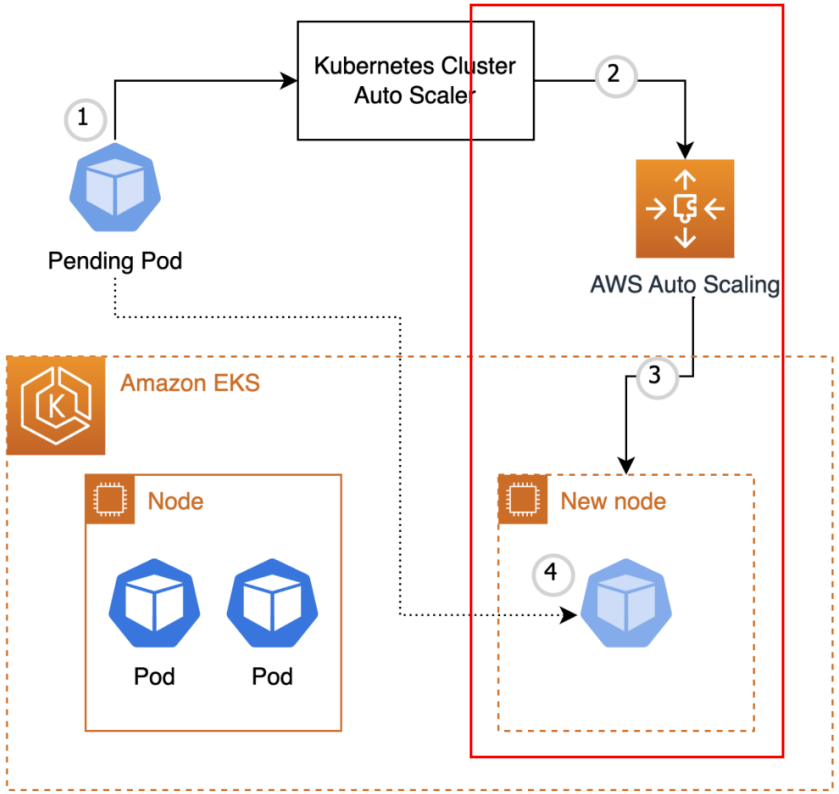
Karpenter - Overview

EKS 환경에서의 Auto Scaling - CAS



1. Unschedulable 상태인 파드가 있는지를 감시합니다.
2. 파드 실행에 필요한 용량만큼 오토 스케일링 그룹의 적정 인스턴스 수(Desired count)를 늘립니다.
3. AWS Auto Scaling을 통해 적정 인스턴스 수에 맞추어 확장합니다.
4. 새 노드가 클러스터에 추가되고 Ready 상태가 되면, Pending 상태였던 파드들이 실행됩니다.

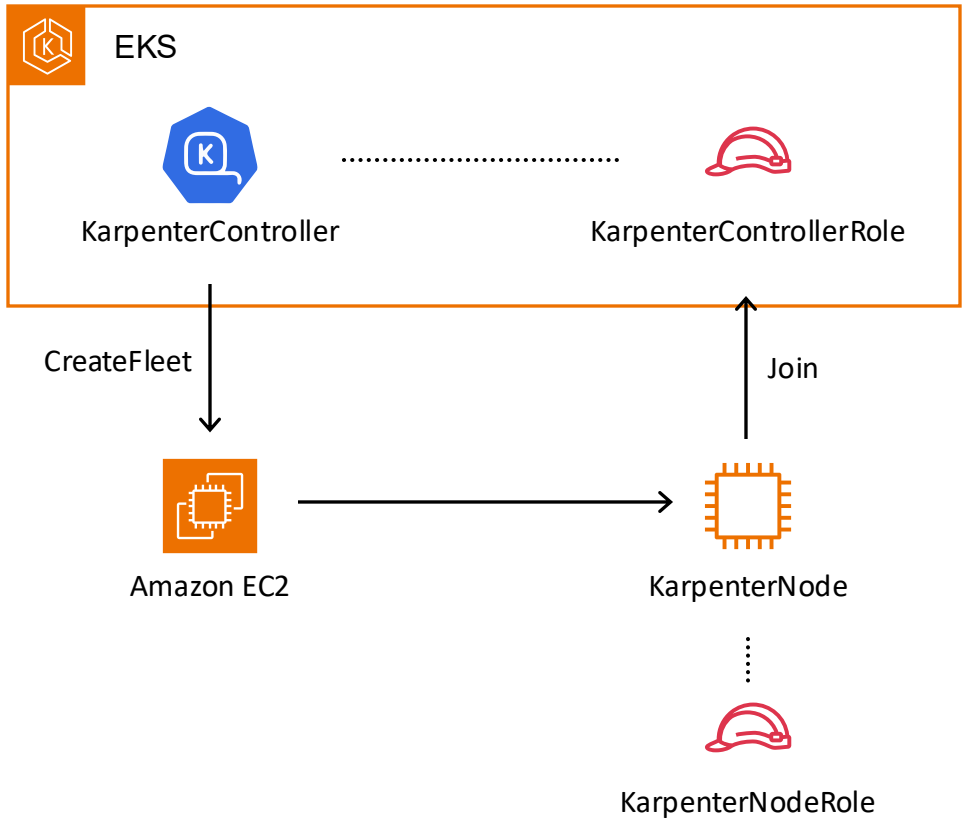
CAS 의 한계



CAS는 구조적으로 AWS의 Auto Scaling에 크게 의존하고 있습니다.
따라서 CA가 ASG를 업데이트한 후, ASG에서 추가 노드를 생성하고 이 노드가 클러스터에 합류하는 과정이 예상보다 시간이 더 소요될 수 있습니다.

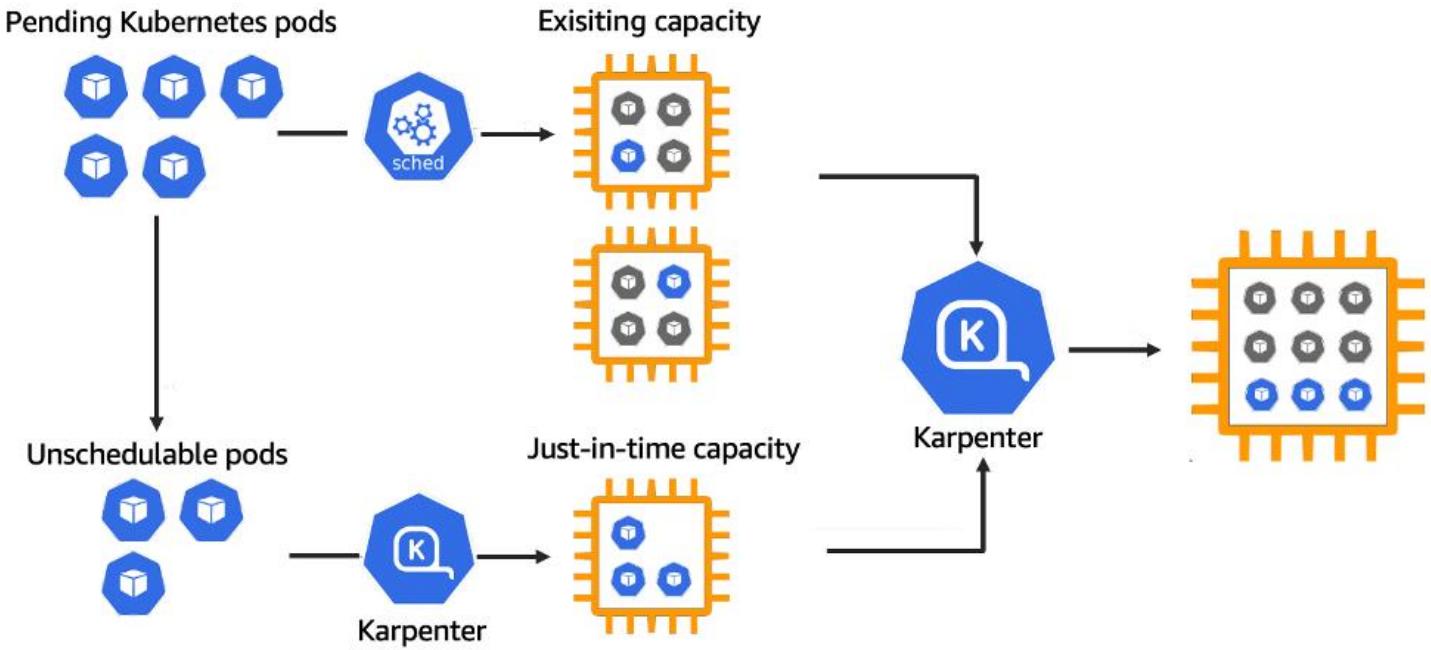
이는 Pod가 확장되어 신속하게 요청을 처리해야 하는 상황에서 충분히 빠른 속도를 제공하지 못할 가능성이 있습니다.

Karpenter 는 어떤 점이 다른가요?



- EC2 Fleet API를 이용해 원하는 인스턴스를 즉시 요청함
- 파드의 리소스 요청에 따라 인스턴스 타입 자동 선택
- 쿠버네티스 스케줄링 제약 조건 기반으로 노드 프로비저닝
- Karpenter에서 정의한 레이블을 사용하여 노드 관리 가능

Karpenter 의 핵심 기능



1. Pending Pod를 발견하면 적절한 Node를 생성
2. NodePool, EC2NodeClass를 반영하여 Node 조정
3. Expiration, Interruption 등을 감지해 Node 삭제
4. 비용 효율적으로 Node 조정

Karpenter CRDs

Karpenter CRDs

NodePool

노드의 논리적 그룹

Node와 Pod의 스케쥴링 제약 조건 정의

- Label
- Annotations
- Taints
- AZ
- Instance Type
- Compute Architecture
- Disruption

EC2NodeClass

NodePool이 생성할 노드의 AWS 관련 구성 정의

- AMI
- Subnet
- Security Group
- IAM Role
- Tags
- Metadata Options
- UserData
- kubelet (일부만 제공)

NodeClaim

Karpenter가 직접 제어하는 Node의 Lifecycle 관리를 위한 객체

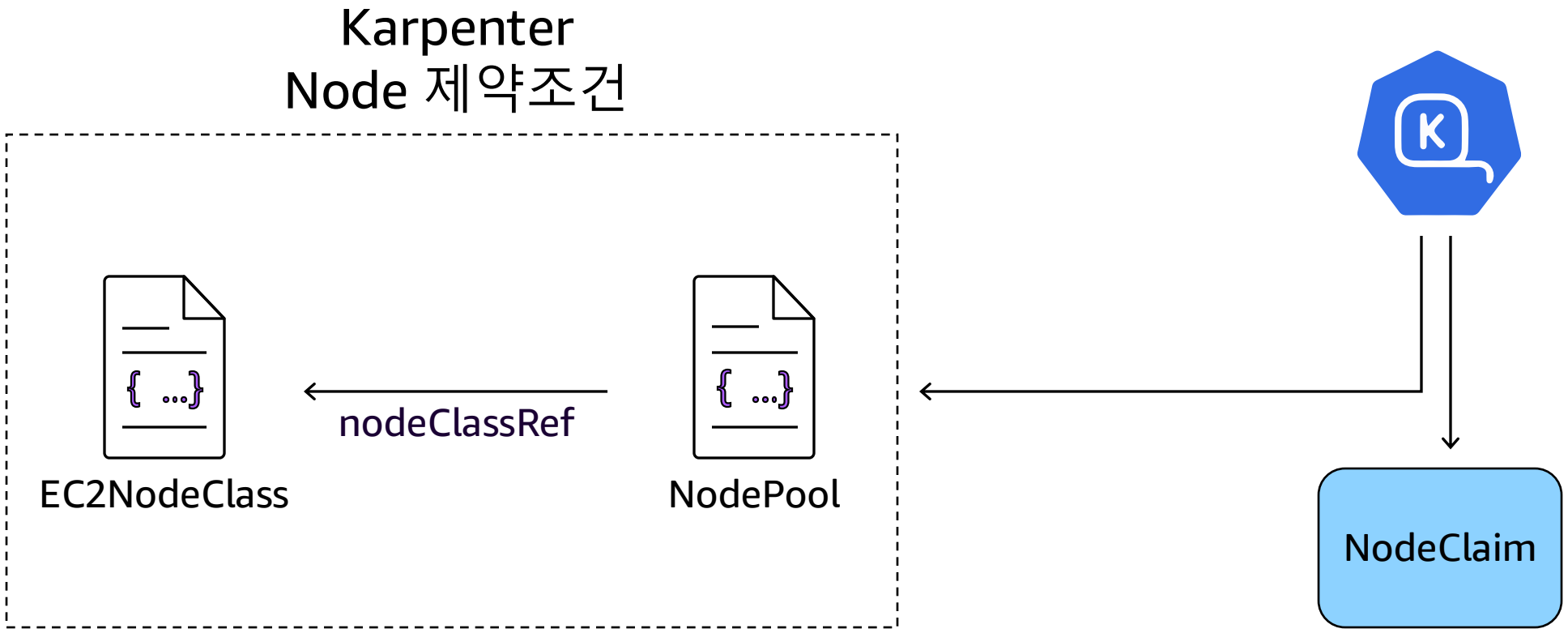
NodeClaim을 모니터링하여 Node의 상태 추적

NodeClaim Spec =

- Pending Pod 요구사항
- NodePool & EC2NodeClass 의 조합

CRD란? : Custom Resource Definition 의 약자로 기본 리소스 외에 사용자가 직접 정의한 새로운 리소스를 관리할 수 있도록 기능을 확장하는 방법

Karpenter CRDs Relationship



NodePool의 주요 구성

labels, annotations, taints, startUpTaints :

- 생성되는 Node에 전파됨
- 이러한 항목의 변경은 노드를 Drifted로 평가하고 노드를 재생성

nodeClassRef :

- NodePool이 사용할 EC2NodeClass 지정

limit :

- NodePool의 최대 vCPU, Memory 크기 설정

```yaml
spec:
  template:
    metadata:
      labels:
        billing-team: my-team
      annotations:
        example.com/owner: "my-team"
    spec:
      taints:
        - key: example.com/special-taint
          effect: NoSchedule
      startupTaints:
        - key: example.com/another-taint
          effect: NoSchedule
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: default
  limits:
    cpu: "1000"
    memory: 1000Gi
```

NodePool의 주요 구성

Requirements :

- 인스턴스 배포 요구사항 정의
- Well-known Kubernetes labels
- Advanced AWS-specific Karpenter labels

Instance Type :

- Instance type, category, family, cpu, memory, generation, hypervisor, size 등의 조건을 설정 가능

Availability Zone :

- 특정 AZ에 노드를 생성하도록 구성 가능
- Pod의 고가용성을 위해서는 topologySpreadConstraints 설정 필요

```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"]
        minValues: 2
      - key: karpenter.k8s.aws/instance-size
        operator: NotIn
        values: ["nano","micro","small"]
      - key: topology.kubernetes.io/zone
        operator: In
        values: ["ap-northeast-2a", "ap-northeast-2b"]
```

NodePool의 주요 구성

CPU architecture :

- amd64
- arm64

Purchase options :

- 설정이 없는 경우, On-Demand 인스턴스 기본 사용
- 여러 type이 지정된 경우, Reserved -> Spot -> On-Demand 순으로 우선순위 지정

```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      requirements:
      - key: kubernetes.io/arch
        operator: In
        values: ["amd64","arm64"]
      - key: karpenter.sh/capacity-type
        operator: In
        values: ["spot","on-demand”,“reserved”]
```

NodePool의 주요 구성

Expiration :

- expireAfter가 지난 노드는 강제종료 됨
- terminationGracePeriod와 함께 사용하여 충분한 draining 시간 확보 필요

Consolidation :

- 비어있거나 사용률이 낮아 더 경제적인 노드로 교체가 가능한 경우 노드를 통합
- consolidateAfter로 안전 버퍼를 확보하거나 통합 자체를 disable 가능

Disruption Budgets :

- Karpenter가 노드를 scale-in 하는 속도를 조절
- Expiration에는 영향을 미치지 않음

```yaml
spec:
  template:
    spec:
      expireAfter: 720h | Never
      terminationGracePeriod: 48h
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized | WhenEmpty
    consolidateAfter: 1m | Never
    budgets:
    - nodes: 10%
    - schedule: "0 9 * * mon-fri"
      duration: 8h
      nodes: "0"
```

EC2NodeClass의 주요 구성

amiFamily :

- 노드의 클러스터 조인을 위한 기본 bootstrap 로직을 결정
- amiSelectorTerms.alias를 지정하지 않은 경우 필수 지정
- amiSelectorTerms.alias를 지정한 경우 amiFamily는 무시되며 alias에 따라 암시적으로 지정됨

```yaml
spec:
  amiFamily: AL2 | AL2023 | Bottlerocket | Custom
  amiSelectorTerms:
  # - alias: al2023@v20240807 | al2@~ | bottlerocket@~
  - tags:
      karpenter.sh/discovery: "${CLUSTER_NAME}"
      environment: test
  - name: "*EKS*"
  - id: ami-123
  - id: ami-456
  - ssmParameter: my-custom-parameter
```

amiSelectorTerms :

- alias(family@version)를 지정하면 EKS Optimized AMI가 선택됨 (mutually exclusive)
- alias version에 latest를 지정하면 최신 버전의 AMI가 선택됨 (Prod 환경 권장하지 않음)
- alias 제외한 동일 term을 중복 지정 가능 (term 간에는 OR / 각 term의 내부 조건은 AND)
- amiSelectorTerms의 값을 바꾸더라도 여전히 NodeClaim과 호환되면 drift가 발생하지 않음

EC2NodeClass의 주요 구성

subnetSelectorTerms :

- 동일 term을 중복 지정 가능 (term 간에는 OR / 각 term의 내부 조건은 AND)
- subnetSelectorTerms의 값을 바꾸더라도 여전히 NodeClaim과 호환되면 drift가 발생하지 않음

securityGroupSelectorTerms :

- 일치하는 모든 SecurityGroup이 적용되기 때문에 장애 방지를 위해 신중하게 지정이 필요

capacityReservationSelectorTerms :

- 일치하는 용량 예약을 우선적으로 사용하고, 부족할 경우에만 On-demand나 Spot 을 사용

```yaml
spec:
  subnetSelectorTerms:
  - tags:
      karpenter.sh/discovery: "${CLUSTER_NAME}"
      environment: test
  - id: subnet-09fa4a0a8f233a921
  securityGroupSelectorTerms:
  - tags:
      karpenter.sh/discovery: "${CLUSTER_NAME}"
      environment: test
  - name: my-security-group
  - id: sg-063d7acfb4b06c82c
  capacityReservationSelectorTerms:
  - tags:
      karpenter.sh/discovery: ${CLUSTER_NAME}
  - id: cr-123
```

EC2NodeClass의 주요 구성

role & instanceProfile:

- KarpenterNodeRole을 명시, 둘 중 하나 지정 필요
- role로 지정한 경우 karpenter는 자동으로 instance profile 생성
- instanceProfile로 지정한 경우 사용자가 직접 생성

metadataOptions :

- 생성된 EC2 인스턴스의 IMDS(Instance Metadata Service)의 노출을 제어
- httpPutResponseHopLimit의 default 값은 1
- 만약 파드에서 인스턴스 메타데이터 사용이 필요하다면 httpPutResponseHopLimit: 2

userData :

- 사용자 지정 UserData는 기본 Bootstrap UserData와 병합됨
- 기본 Bootstrap UserData의 모든 값은 사용자 지정 UserData에 지정된 값보다 우선함

```
spec:
  role: "KarpenterNodeRole-${CLUSTER_NAME}"
  # instanceProfile: "KarpenterNodeInstanceProfile-${CLUSTER_NAME}"
  metadataOptions:
    httpEndpoint: enabled
    httpProtocolIPv6: disabled
    httpPutResponseHopLimit: 1
    httpTokens: required
  userData: |
    echo "Hello world"
  tags:
    team: team-a
    app: team-a-app
```

tags :

- Karpenter는 생성하는 모든 Node 관련 리소스에 기본 태그를 전파
- 사용자 지정 태그 추가 가능

EC2NodeClass의 주요 구성

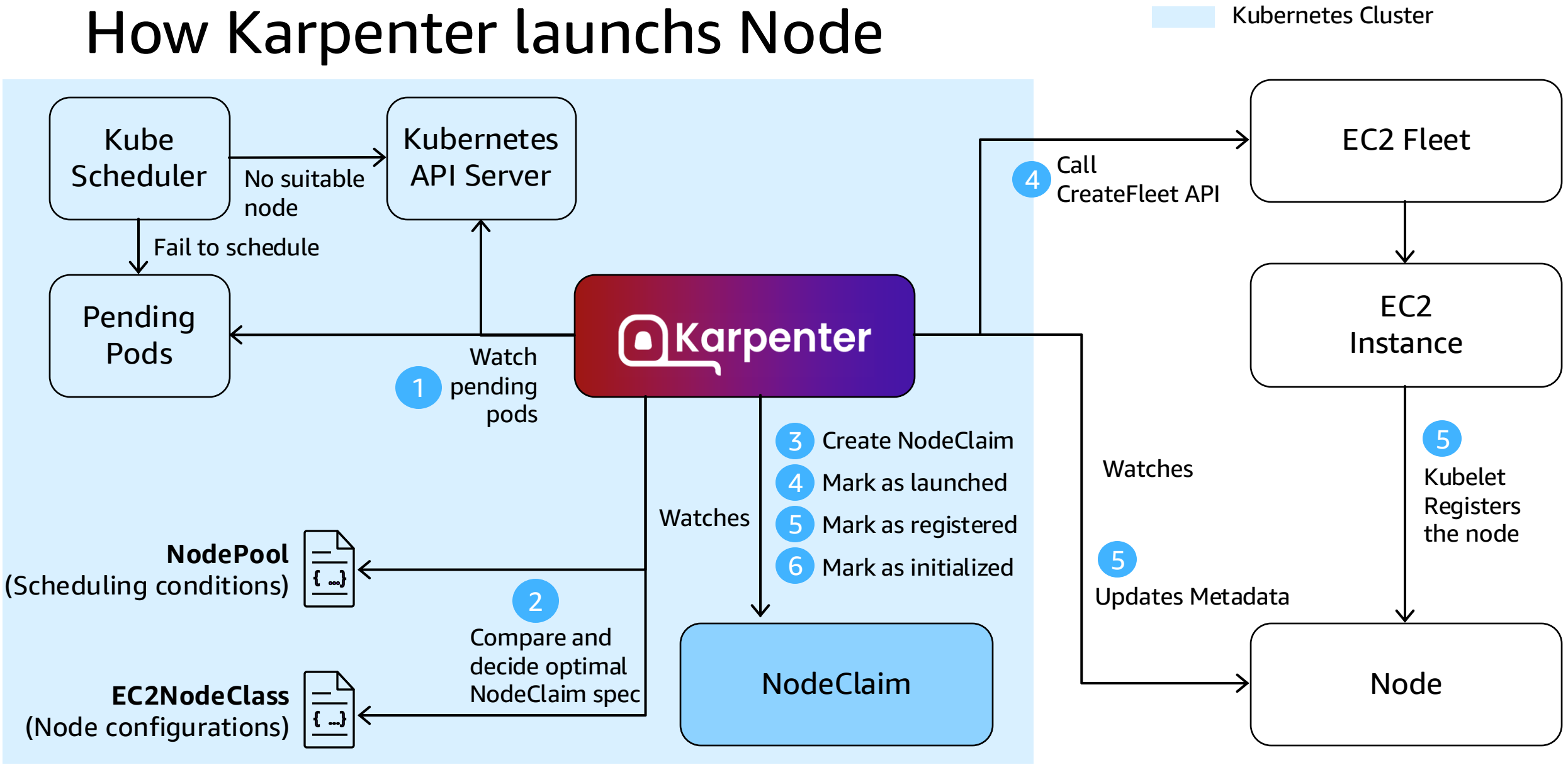
Kubelet :

- 제한된 범위의 kubelet 인수 지정 가능
- spec.kubelet에서 제공되지 않는 인수를 지정해야 하는 경우 UserData를 통해 AMI Family에 적합한 방법으로 지정 필요

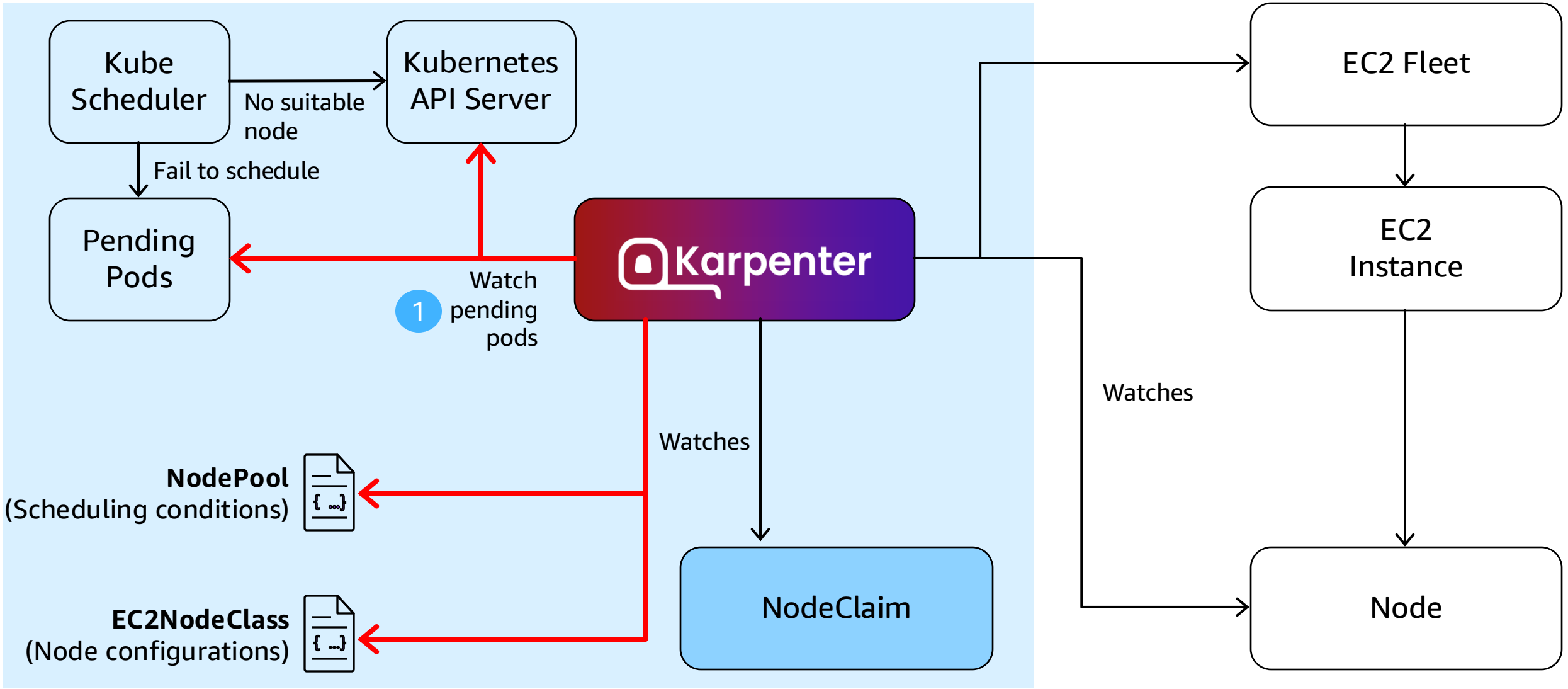
```yaml
spec:
  kubelet:
    podsPerCore: 2
    maxPods: 20
    systemReserved:
      cpu: 100m
      memory: 100Mi
      ephemeral-storage: 1Gi
    kubeReserved:
      cpu: 200m
      memory: 100Mi
      ephemeral-storage: 3Gi
    evictionHard:
      memory.available: 5%
      nodefs.available: 10%
      nodefs.inodesFree: 10%
    evictionSoft:
      memory.available: 500Mi
      nodefs.available: 15%
      nodefs.inodesFree: 15%
    evictionSoftGracePeriod:
      memory.available: 1m
      nodefs.available: 1m30s
      nodefs.inodesFree: 2m
    evictionMaxPodGracePeriod: 60
    imageGCHighThresholdPercent: 85
    imageGCLowThresholdPercent: 80
    cpuCFSQuota: true
    clusterDNS: ["10.0.1.100"]
```

Karpenter의 Scheduling

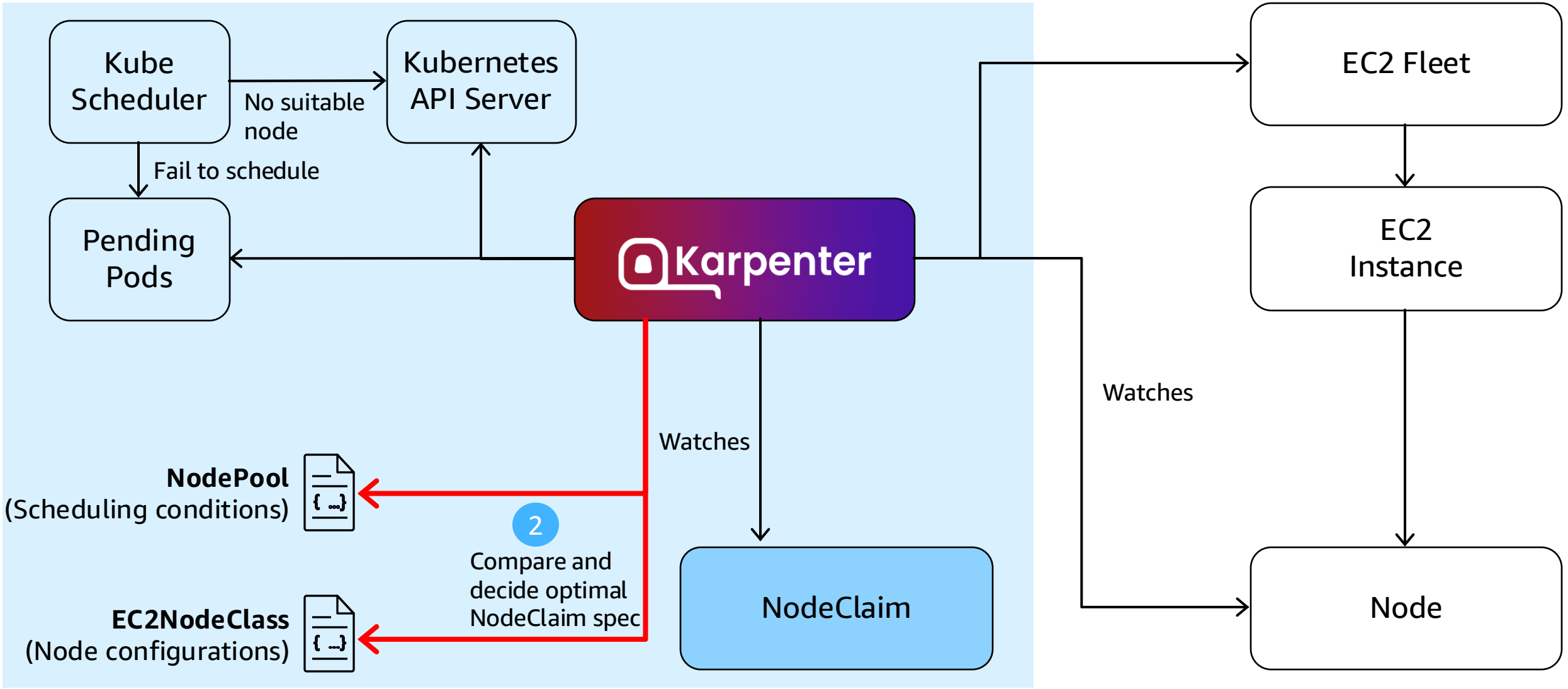
How Karpenter launchs Node



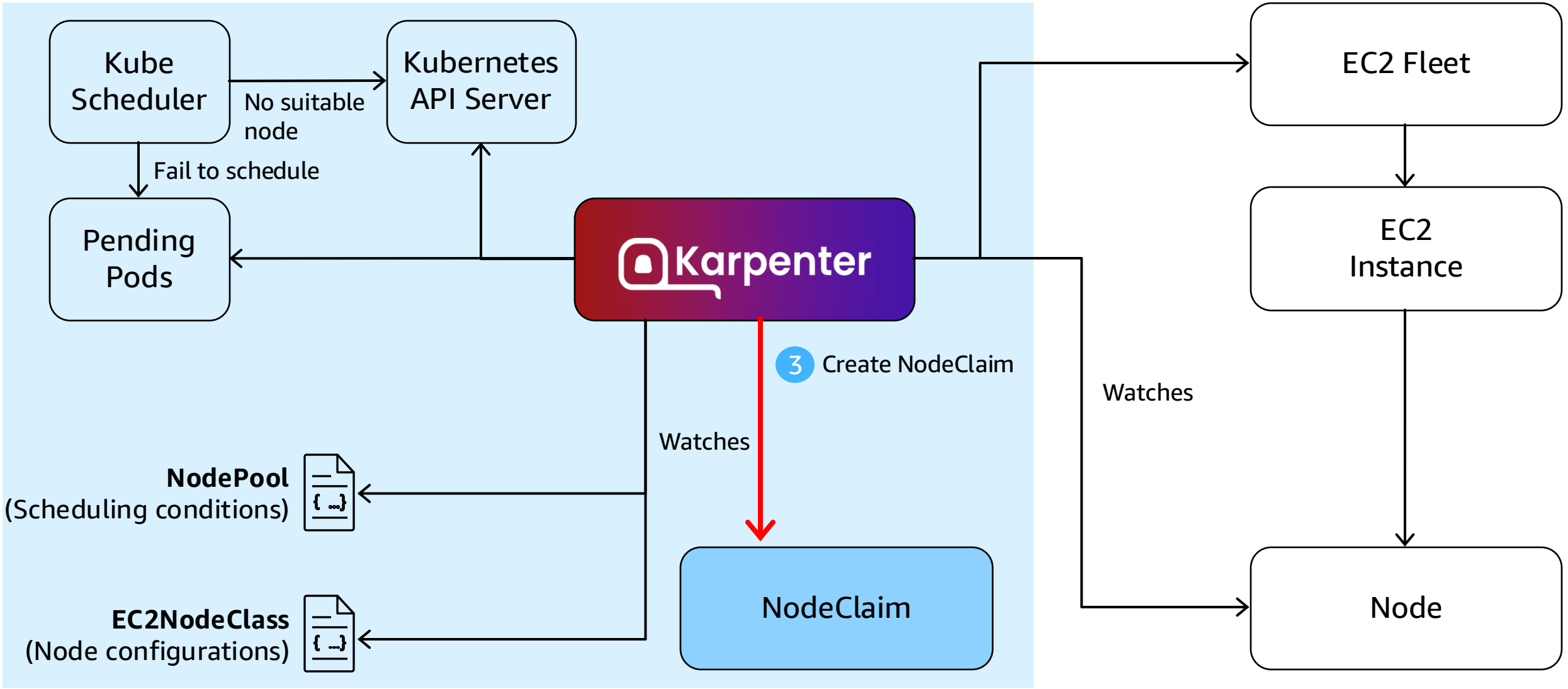
How Karpenter launchs Node



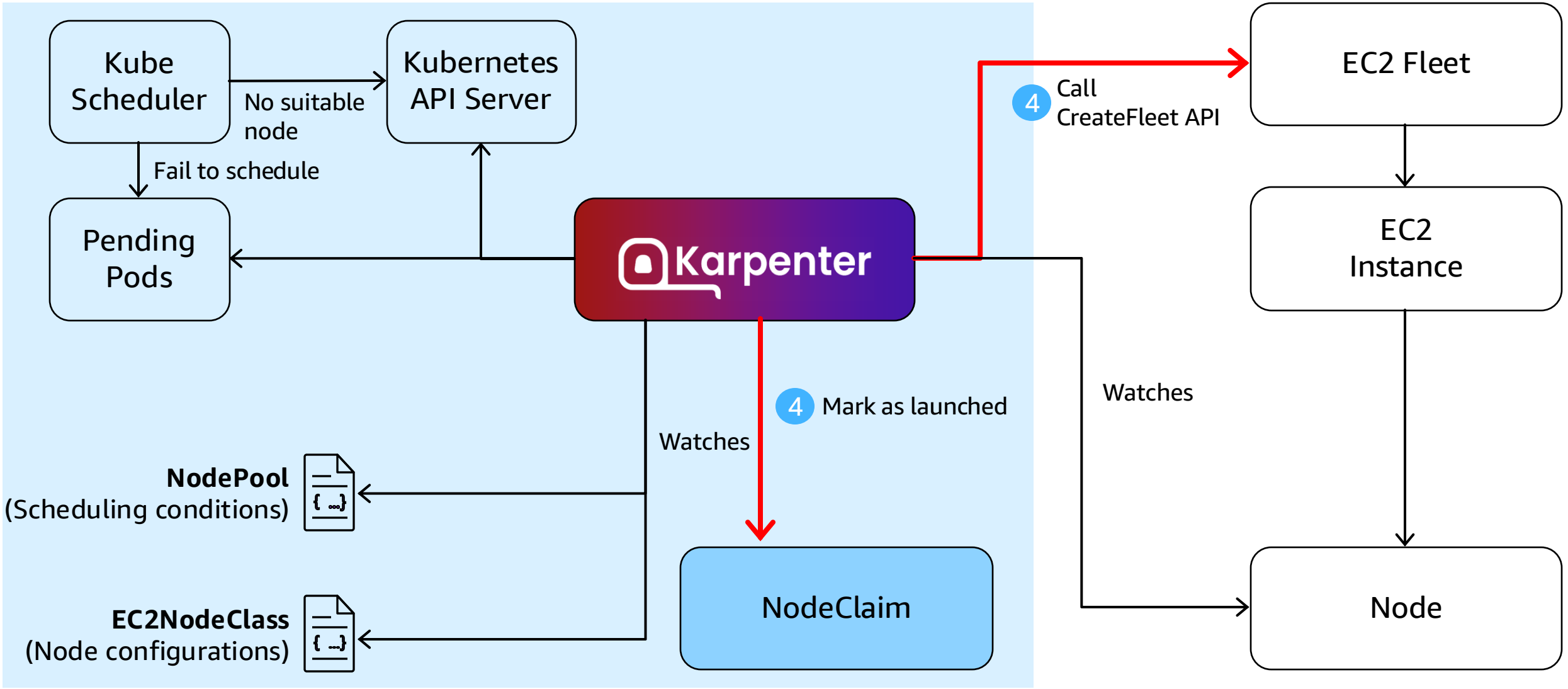
How Karpenter launchs Node



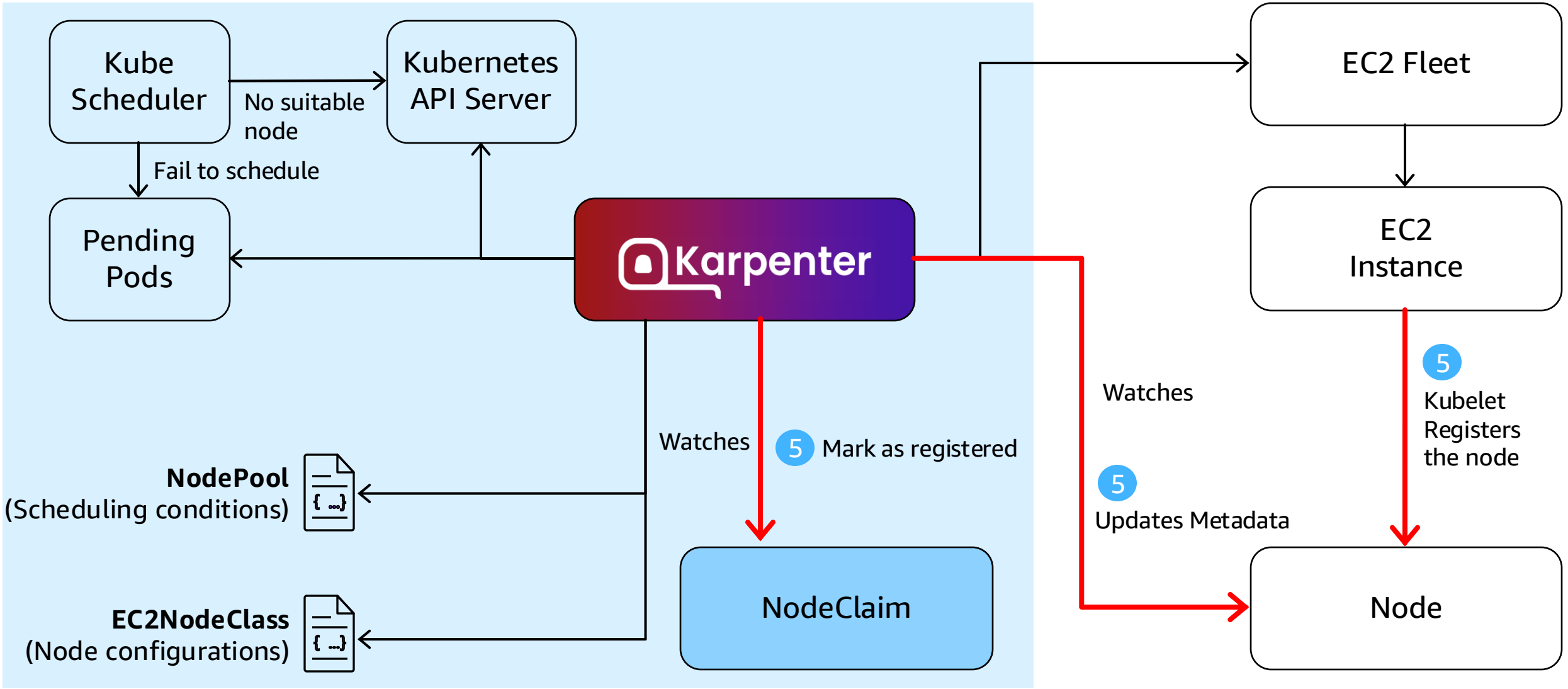
How Karpenter launchs Node



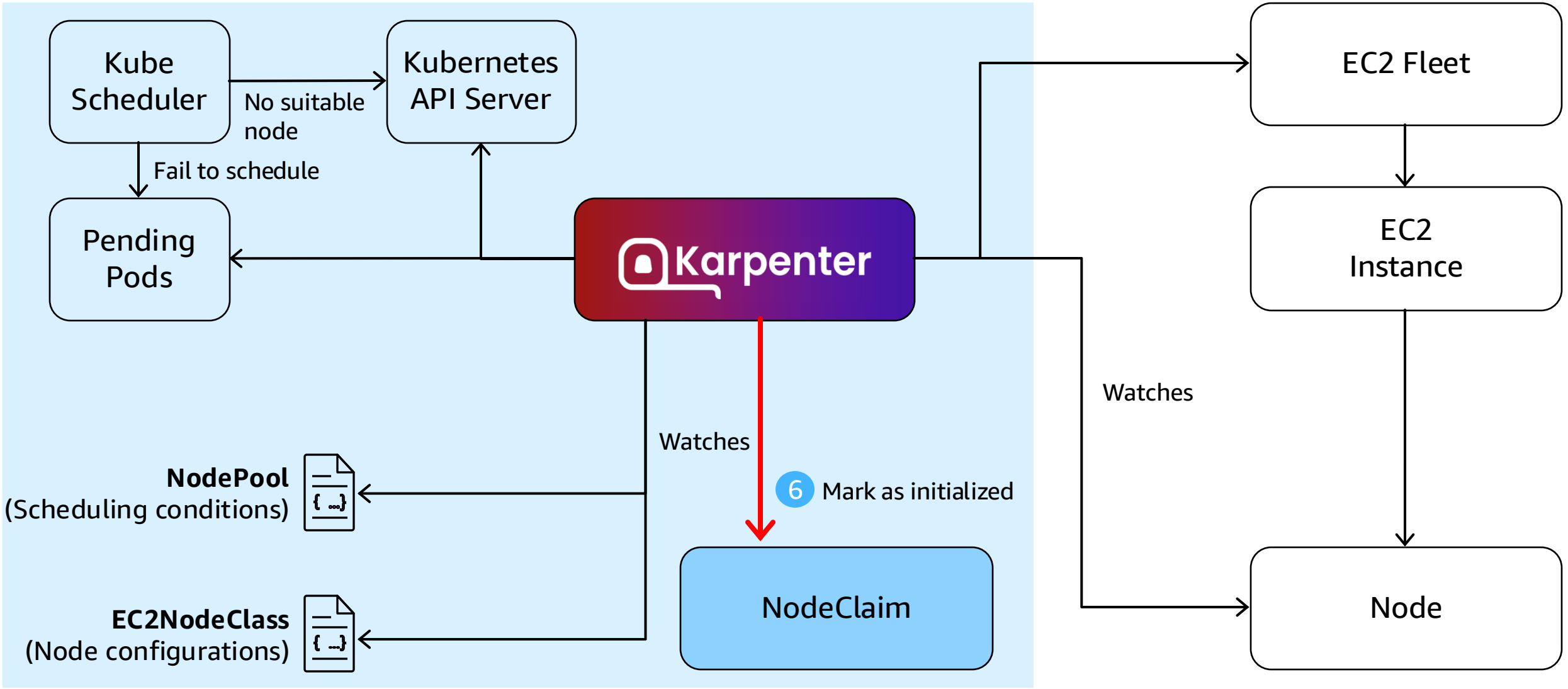
How Karpenter launchs Node



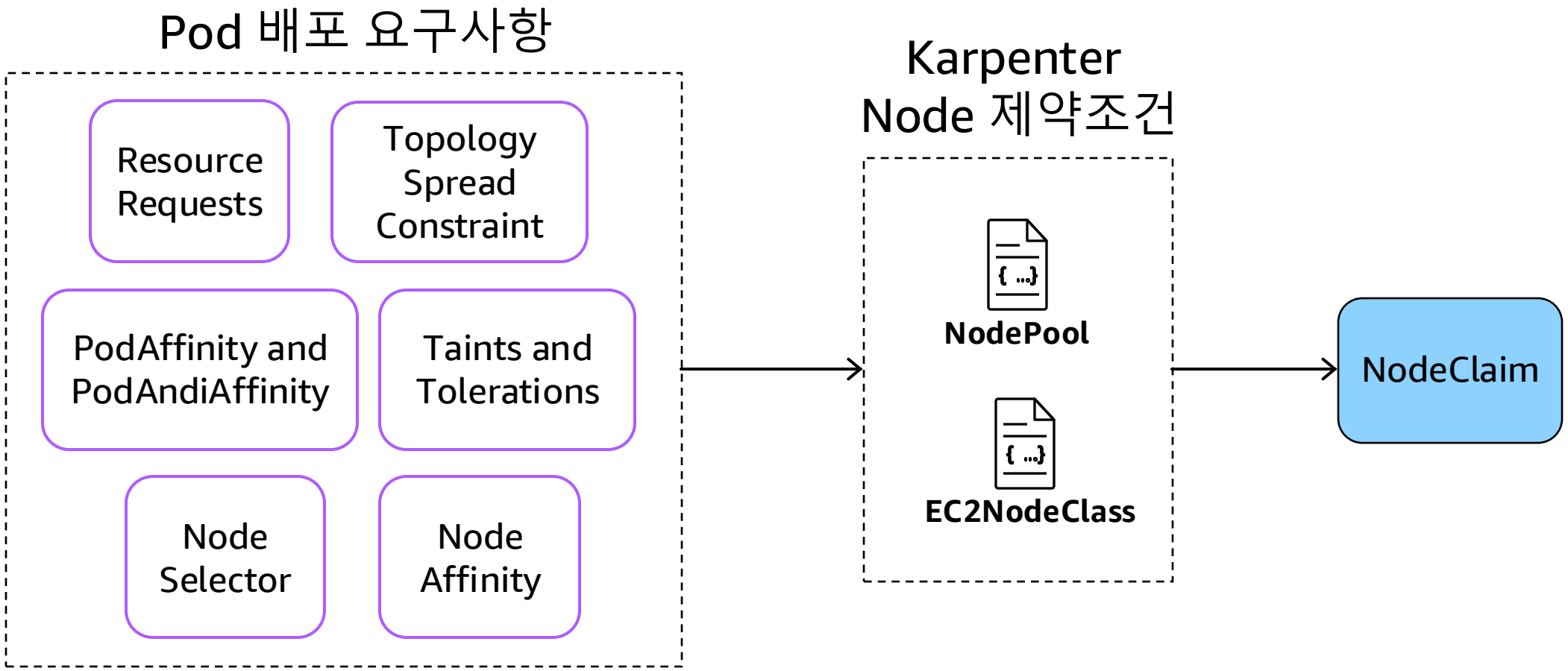
How Karpenter launchs Node



How Karpenter launchs Node



How Karpenter decide NodeClaim Spec



Node launch: scheduling #example

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gpu-test
spec:
  replicas: 1
  template:
    spec:
      tolerations:
        - key: "nvidia.com/gpu"
          operator: "Equal"
          value: "true"
          effect: "NoSchedule"
      containers:
        - name: cuda
          image: nvidia/cuda:11.0.3-base
```

```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default-gpu
spec:
  template:
    spec:
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: default-gpu
      taints:
        - key: nvidia.com/gpu
          value: "true"
          effect: "NoSchedule"
      requirements:
```

- 파드에 GPU 전용 노드에 대한 Tolerations 설정
- 노드에 Tolerations에 매칭되는 Taints를 설정

Node launch: scheduling #example

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: node-affinity-test
spec:
  replicas: 1
  template:
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: kubernetes.io/arch
                    operator: In
                    values:
                      - arm64
```

```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: default
      requirements:
        - key: kubernetes.io/arch
          operator: In
          values: ["amd64", "arm64"]
```

- 파드에 nodeAffinity를 활용해 파드를 arm64 아키텍처 노드에 스케줄링 하도록 설정
- NodePool의 Requirements에 karpenter.io/arch 항목이 arm64를 포함하도록 설정

Best Practice - 워크로드 특성 별 NodePool 구성

워크로드 특성에 따라 NodePool을 분리

Taints / Tolerations :

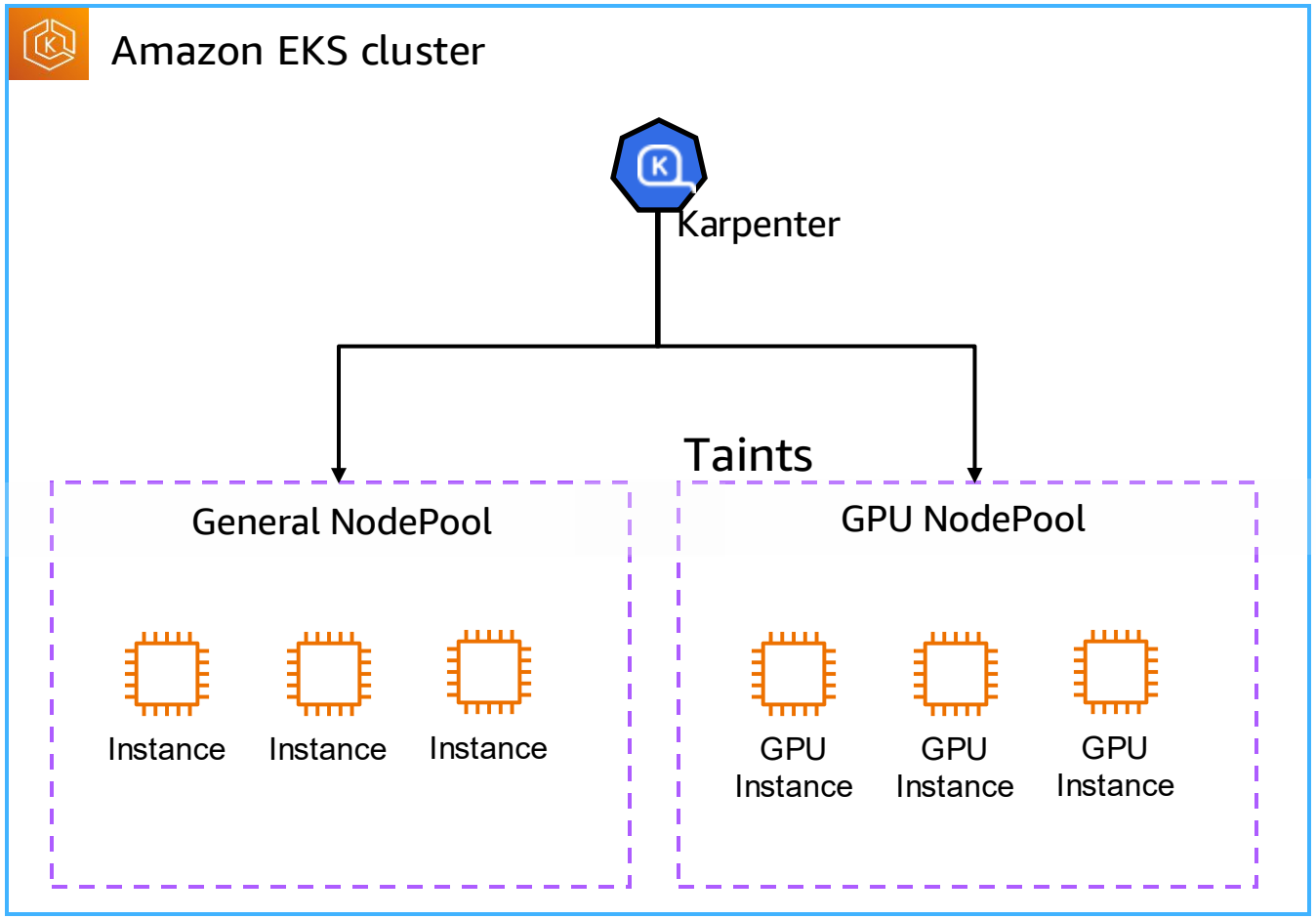
- 특정 파드만 배치되어야 하는 NodePool 지정

TopologySpreadConstraints :

- NodePool의 label을 TopologyKey로 지정
- 여러 NodePool 들의 Node 간 Pod를 균등히 분산

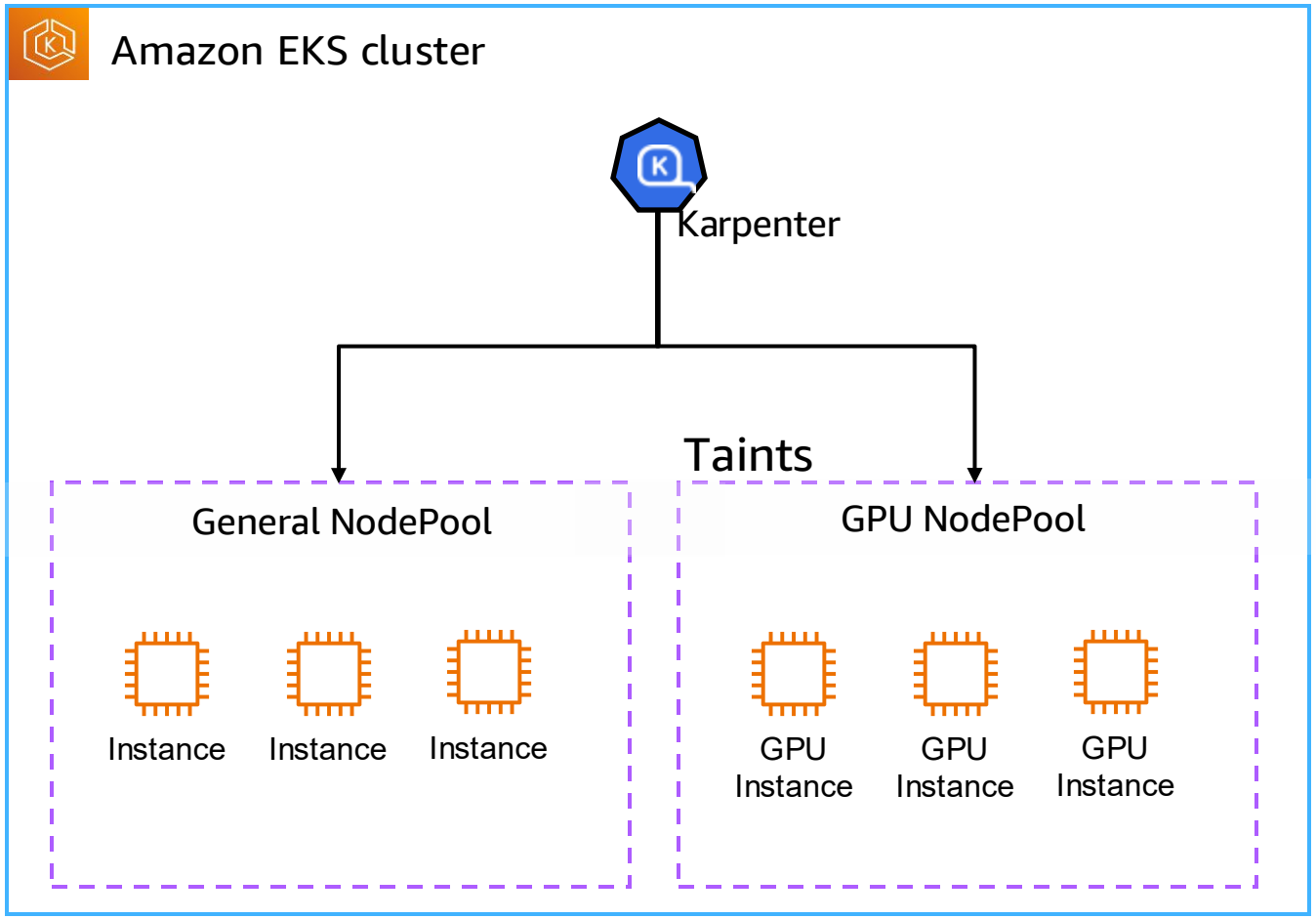
Weight :

- Pod를 스케줄링할 Node 선택시 더 높은 확률을 부여하는 NodePool



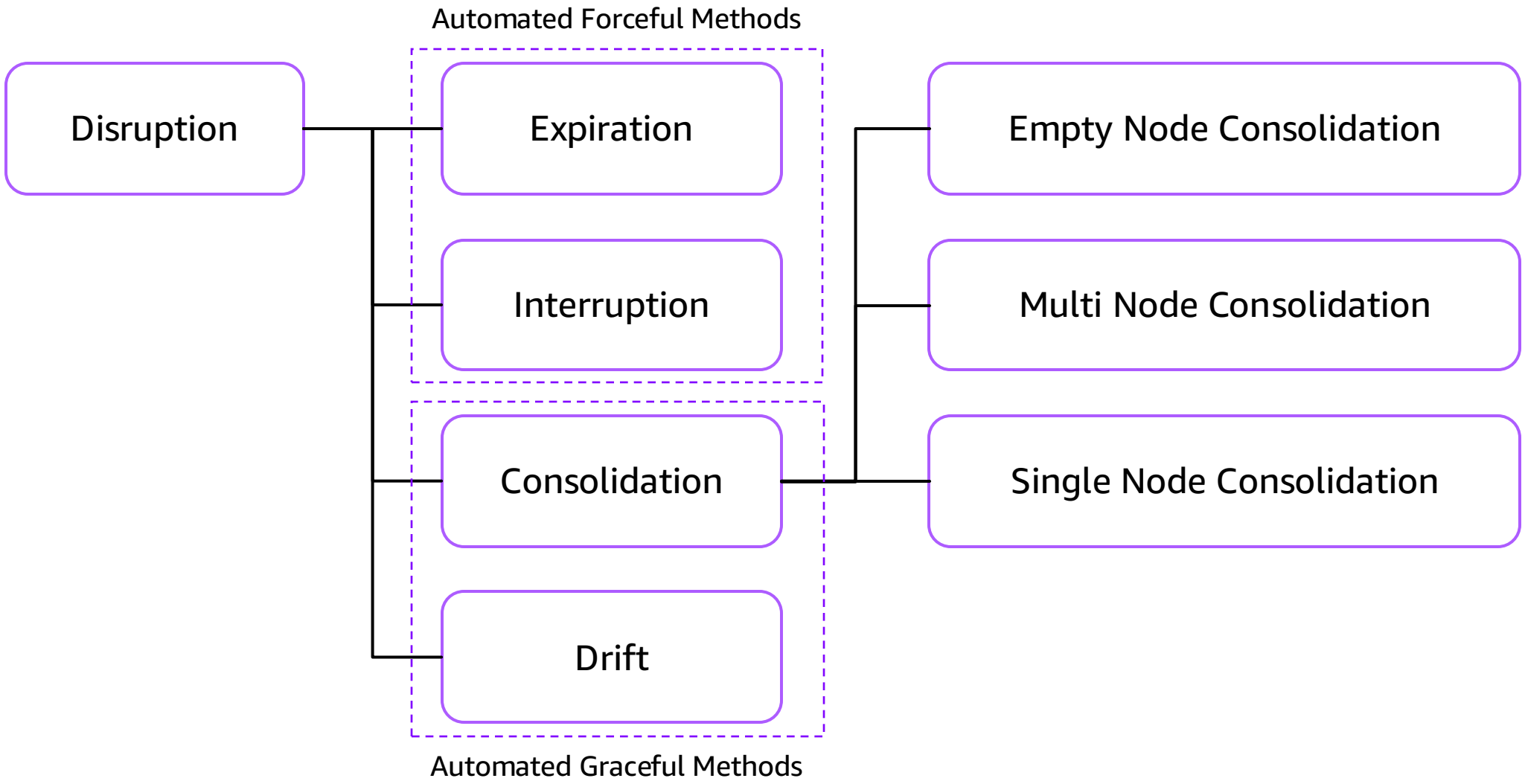
Best Practice - 워크로드 특성 별 NodePool 구성

- 상호 배타적인 (mutually exclusive) NodePool을 생성하는 것을 권장합니다.
- Fargate 또는 노드 그룹(MNG)에 속한 worker node에서 Karpenter 컨트롤러를 실행하는 것을 권장합니다.
- Spot 사용시 중단 처리(Interruption Handling) 활성화하는 것을 권장합니다.



Karpenter의 Disruption (중단)

Karpenter Disruption (중단)의 종류



Forceful, Graceful Methods의 차이

Graceful Methods는 NodePool의 disruption.budgets 을 통해 속도를 제한할 수 있는 자동화된 유예 방법입니다.

Forceful Methods는 조건이 충족되는 즉시 노드 드레이닝을 시작합니다.

- Graceful와 달리, disruption.budgets을 사용하여 속도를 제한할 수 없으며, 파드가 스케줄을 재조정하기 위해 사전 스핀 대체 노드가 정상 상태가 될 때까지 기다리지 않습니다.
- Pod disruption budgets을 애플리케이션 중단 속도를 제한하는 데 사용할 수 있습니다.

```yaml
# NodePool

spec:
  template:
    spec:
      expireAfter: 720h
      terminationGracePeriod: 48h
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 5m
    budgets:
    - nodes: 10%
    - schedule: "0 9 * * mon-fri"
      duration: 8h
      nodes: "0"
```

Expiration (만료)

expireAfter 값에 따라 설정된 시간이 경과한 노드에 대해 Expiration으로 Annotations를 부여하고 노드를 중단합니다.

- expireAfter가 지난 노드는 강제종료 됨
- terminationGracePeriod와 함께 사용하여 충분한 draining 시간 확보 필요

```yaml
spec:
  template:
    spec:
      expireAfter: 720h | Never
      terminationGracePeriod: 48h
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized | WhenEmpty
    consolidateAfter: 1m | Never
    budgets:
    - nodes: 10%
    - schedule: "0 9 * * mon-fri"
      duration: 8h
      nodes: "0"
```

Interruption (중단)

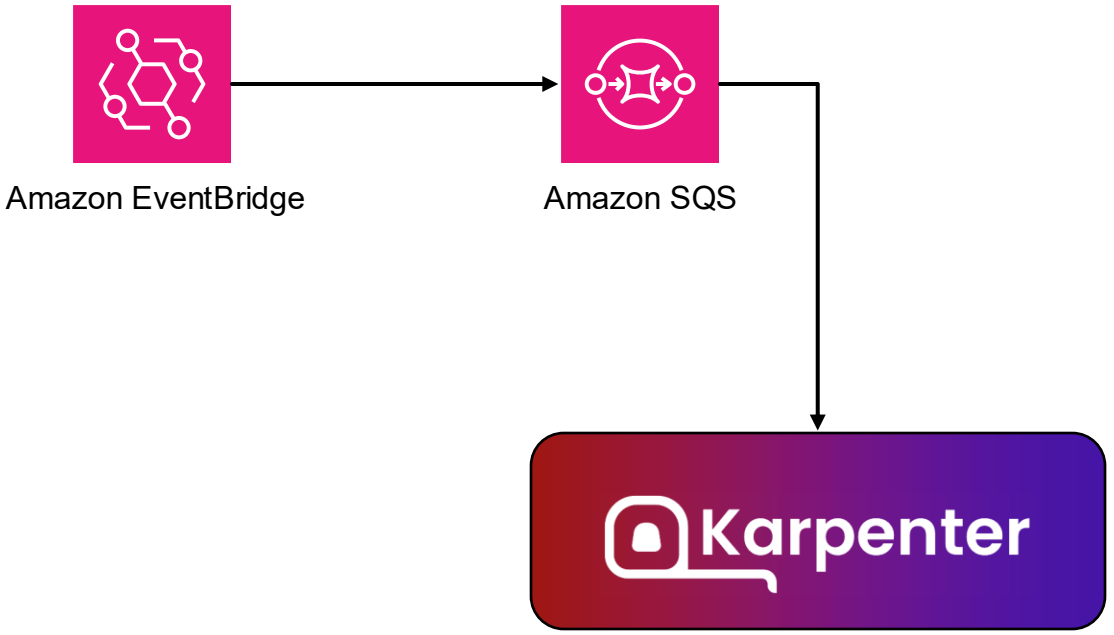
Karpenter가 의도하지 않은 비 자발적 Node 중단 이벤트 감시합니다.

- Spot Interruption Warnings
- Scheduled Change Health Events
- Instance Terminating Events
- Instance Stopping Events

Interruption이 발생할 것을 감지하면 즉시 노드 Cleanup 프로세스 시작

Eventbridge Rule과 SQS Queue를 이용하여 이벤트 메세지를 폴링하여 수집

Spot Interruption Warnings의 경우 2분 전 알림



Drift

Drift는 NodePool/EC2NodeClass에 대한 변경을 처리합니다.

- NodeClaim은 소유하고 있는 NodePool/EC2NodeClass의 값이 NodeClaim의 값과 일치하지 않으면 Drift된 것으로 감지됩니다.
- 일반적으로 NodeClaim/Instance/NodePool/EC2NodeClass 변경에 의해 트리거됩니다.

latest로 지정시 새롭게 이미지 버전이 런칭될 때마다 Drift가 발생하게 됩니다.

```
spec:
  amiFamily: AL2023
  amiSelectorTerms:
    - alias: al2023@latest
  subnetSelectorTerms:
    - tags:
        karpenter.sh/discovery: "${CLUSTER_NAME}"
        environment: test
  securityGroupSelectorTerms:
    - tags:
        karpenter.sh/discovery: "${CLUSTER_NAME}"
        environment: test
```

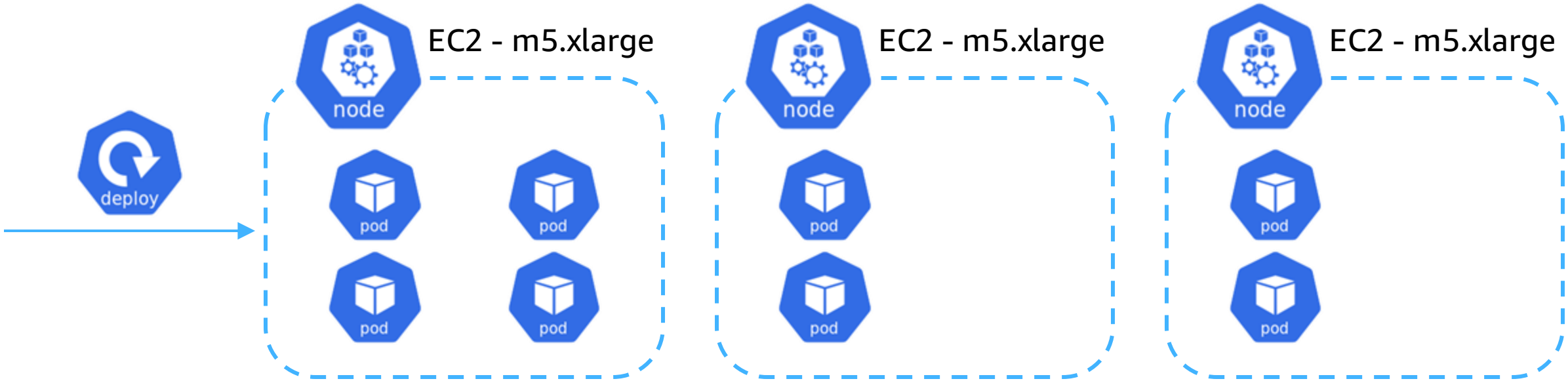
Consolidation (통합)

Karpenter에는 Consolidation을 위한 두 가지 메커니즘이 있습니다

- Deletion : 클러스터 내 다른 노드의 여유 용량으로 모든 파드를 실행할 수 있는 경우 노드를 삭제할 수 있습니다.
- Replace : 클러스터 내 다른 노드의 여유 용량과 더 저렴한 단일 교체 노드의 조합으로 모든 파드를 실행할 수 있는 경우 노드를 교체할 수 있습니다.

```yaml
# NodePool

apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30m
```

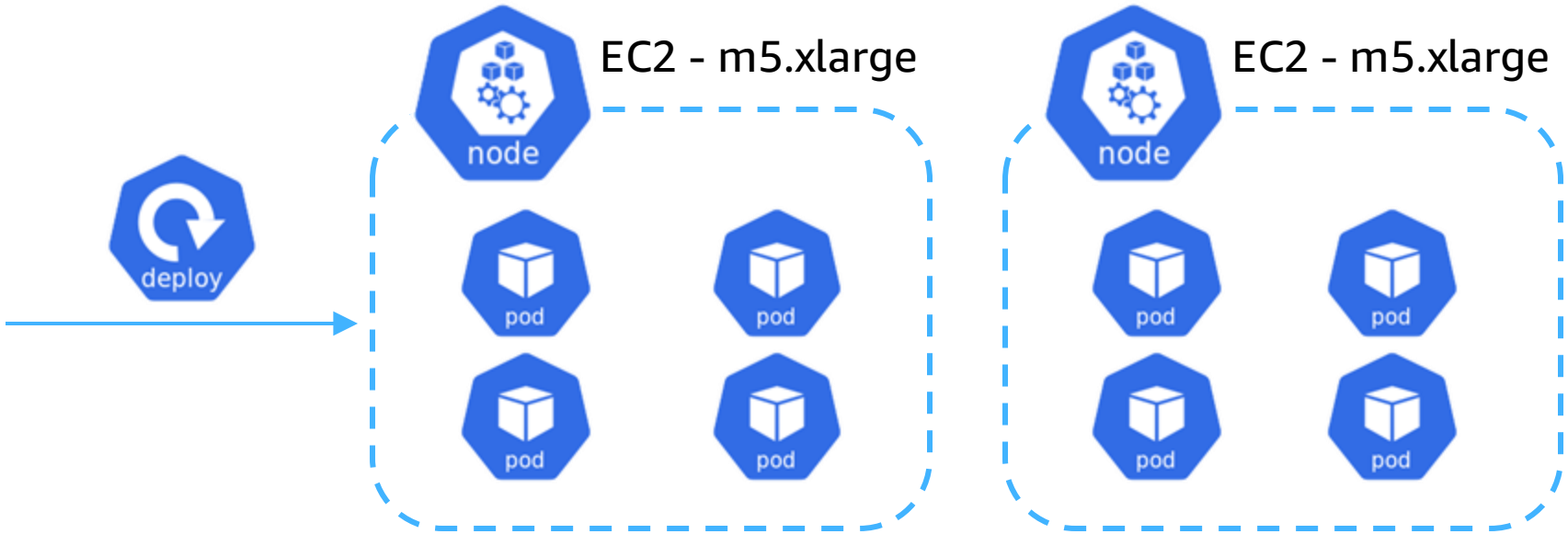
Consolidation (통합) : Example



consolidationPolicy

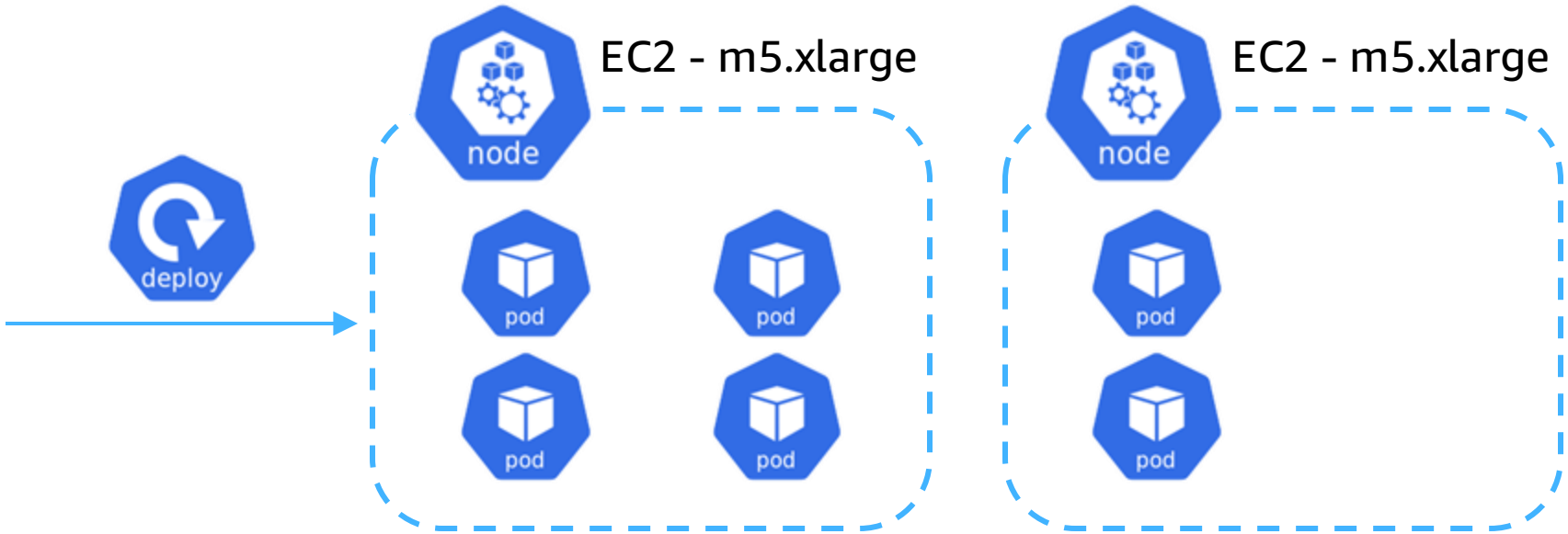
```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
spec:
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
```

Consolidation (통합) : Example

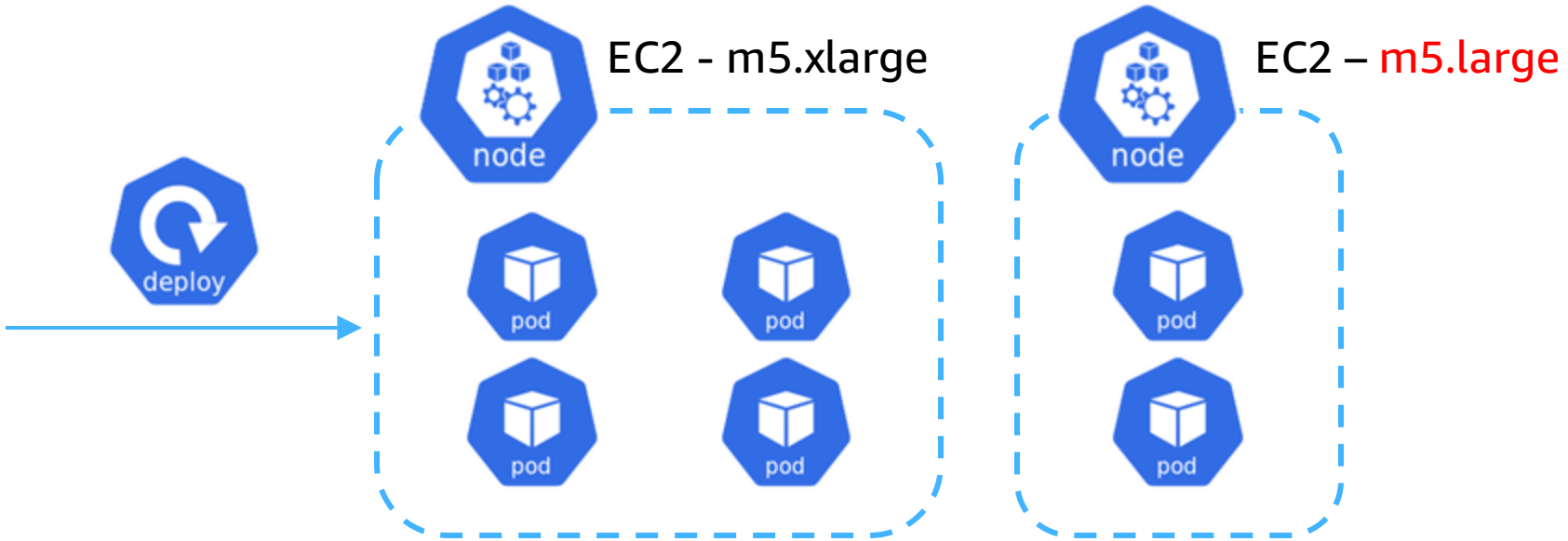


사용량이 적은 노드를 삭제하고 하나의 노드로 통합하여
노드 사용률 개선 및 비용 절감

Consolidation (통합) : Example



Consolidation (통합) : Example



더 적은 사용량 요구사항에 맞도록 노드 사양 최적화하여 비용 절감

Consolidation (통합)

통합에는 통합 작업을 식별하기 위해 수행되는 세 가지 메커니즘이 있습니다

- Empty Node Consolidation - 완전히 비어있는 노드를 병렬로 삭제
- Multi Node Consolidation - 두 개 이상의 노드를 병렬로 삭제하려고 시도하며, 제거되는 모든 노드의 가격보다 낮은 단일 대체 노드를 시작할 수 있음
- Single Node Consolidation - 제거되는 노드의 가격보다 낮은 단일 대체 노드를 시작할 수 있는 단일 노드를 삭제하려고 시도합니다.

```yaml
# NodePool

apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30m
```

Consolidation (통합) : Configuration

whenEmpty :

- 데몬셋 Pod는 고려대상이 아님

whenEmptyOrUnderutilized :

- Pod를 다른 노드에서 실행할 수 있거나 더 낮은 가격의 변경으로 교체

consolidateAfter :

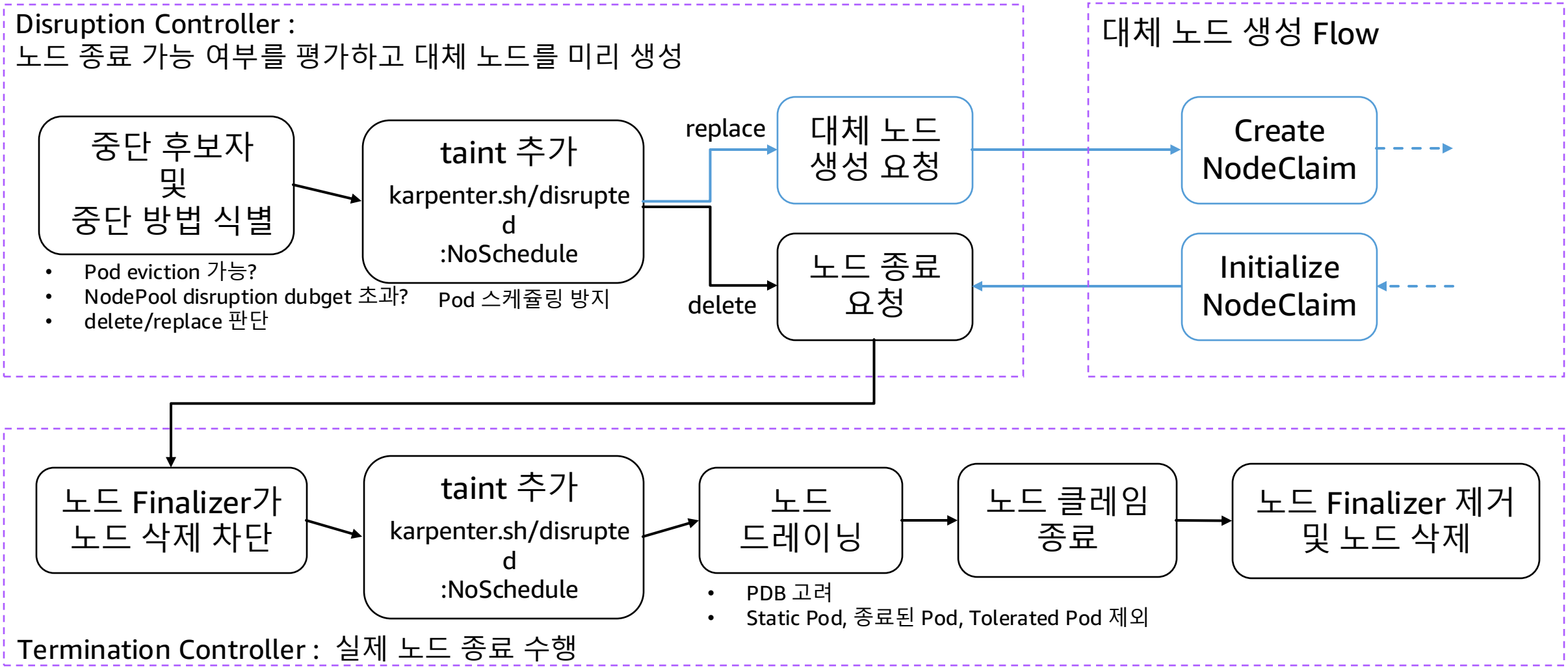
- 노드 상태 변화(Pod 추가/삭제) 이후, 해당 노드를 통합 대상으로 고려하기 까지 기다리는 시간

노드 통합 우선순위 :

- 더 적은 수의 파드를 실행하는 노드
- 곧 expire되는 노드
- Priority가 낮은 파드가 있는 노드

```yaml
# NodePool

apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30m
```

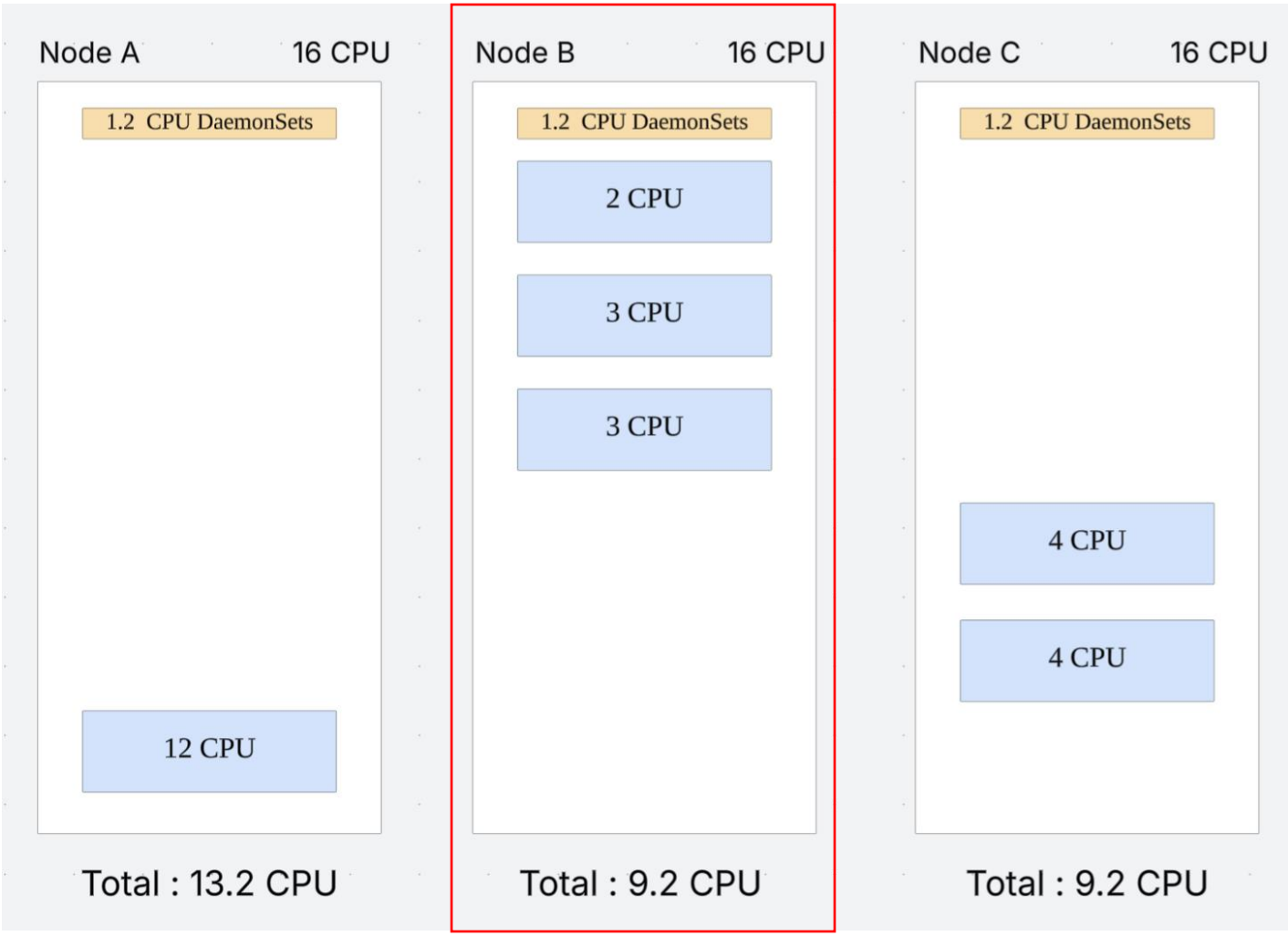
How Karpenter terminates Node



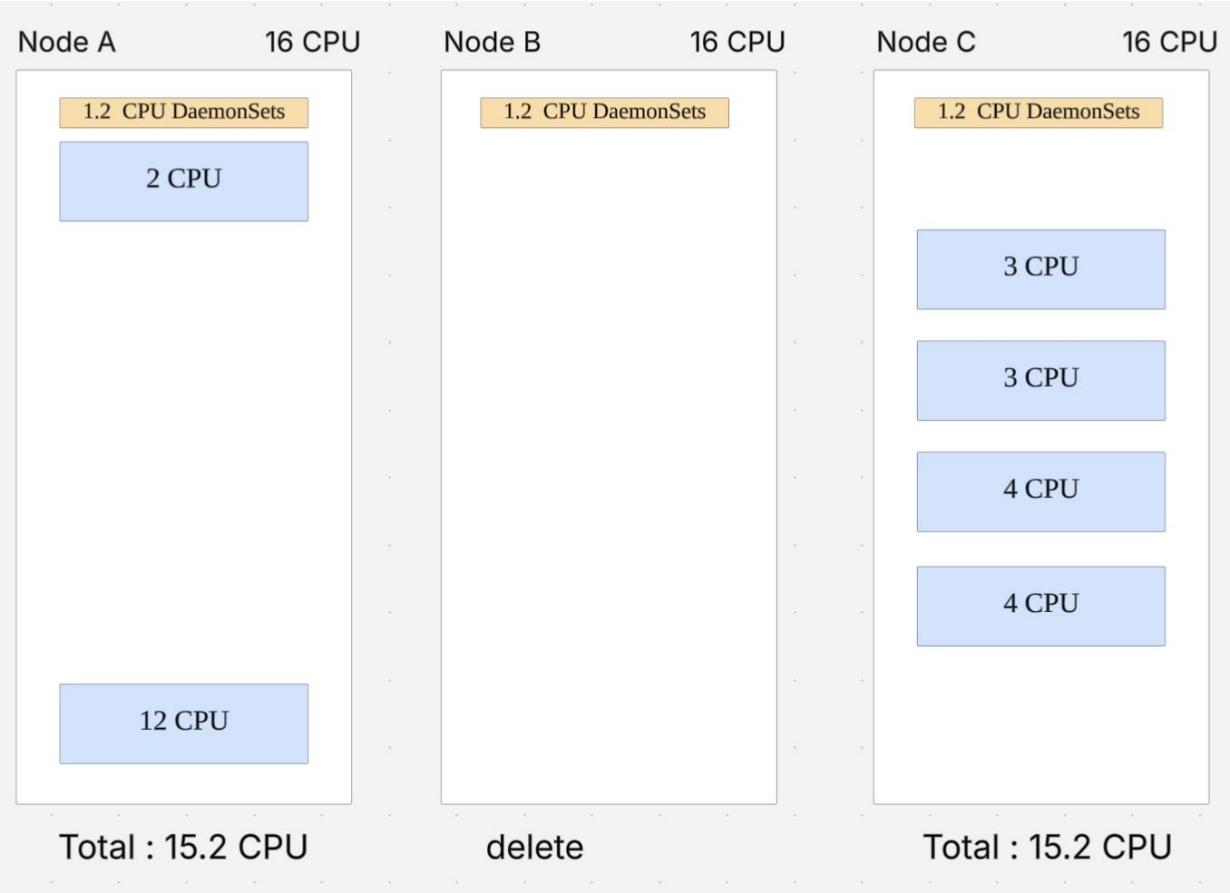
Delete 동작에서 고려해야할 점

Karpenter는 노드가 Consolidation 가능한지 판단하는 과정에서 Pod가 다른 노드로 스케줄링 가능한지 판단하는 Simulation 과정을 거칩니다.

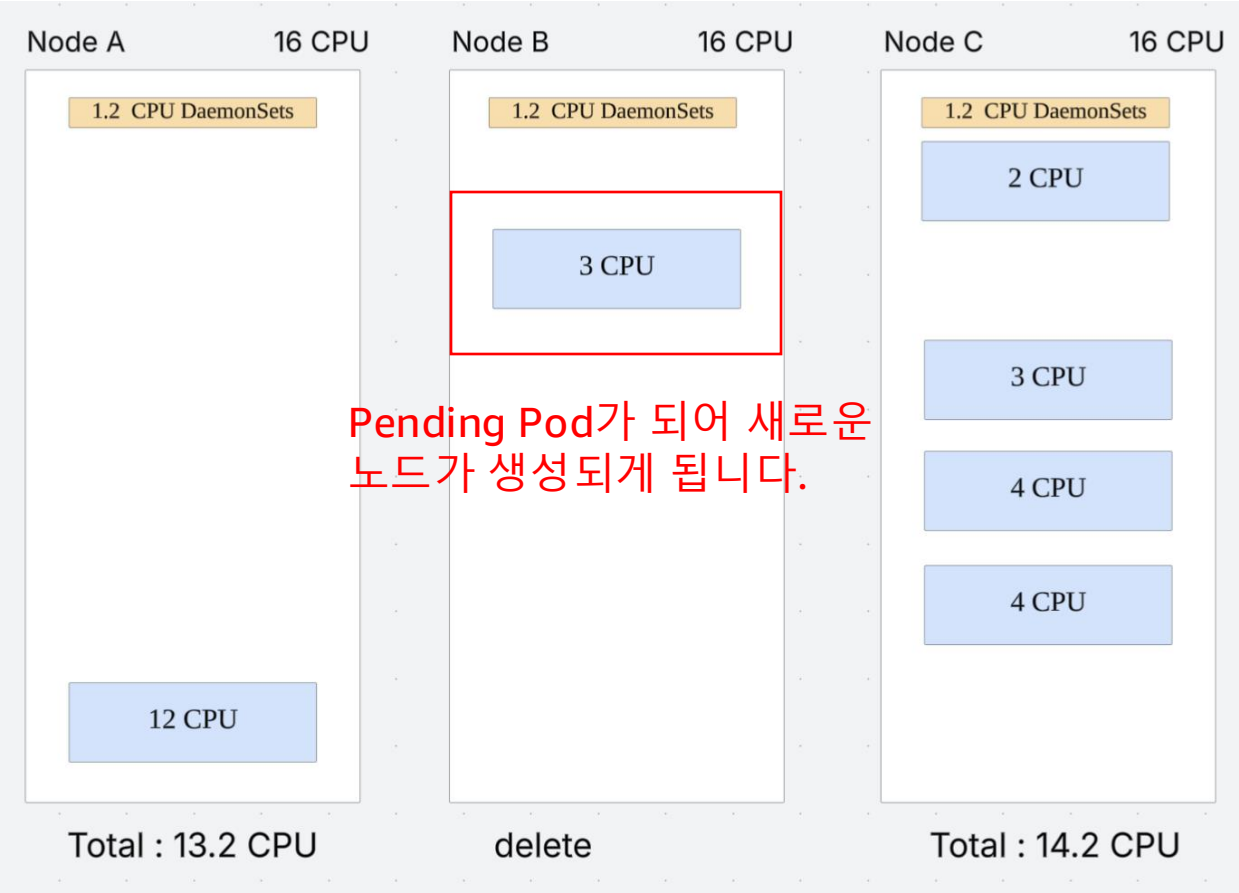
이때 First-fit-decreasing bin packing 알고리즘을 사용하게 되는데 이는 실제 Pod를 스케줄링하는 kube-scheduler의 동작과 다를 수 있습니다.



Karpenter Expectation



Actual Schdeuling by Kube-Scheduler



Node Auto Repair v1.1 Alpah

상태가 좋지 않은 노드를 자동으로 식별하고 교체하여 안정적인 클러스터 상태 유지

- Node Monitoring Agent add-on설치
- NodeRepair=true 기능 플래그 활성화 필요

비정상 상태 Type 별로 허용기간을 초과한 경우 신속한 교체를 위해 즉시 노드를 강제 종료하고 Cleanup 프로세스를 건너 뜀

연쇄적인 장애 방지를 위해 NodePool의 노드 중 20% 이상이 비정상이면 Repair 중단

Type	Status	허용 기간
Kubelet Ready	False	30 m
Kubelet Ready	Unknown	30 m
Accelerate Hardware Ready	False	10 m
Storage Ready	False	30 m
Network Ready	False	30 m
Kernel Ready	False	30 m
Container Runtime Ready	False	30 m

Demo & Troubleshooting

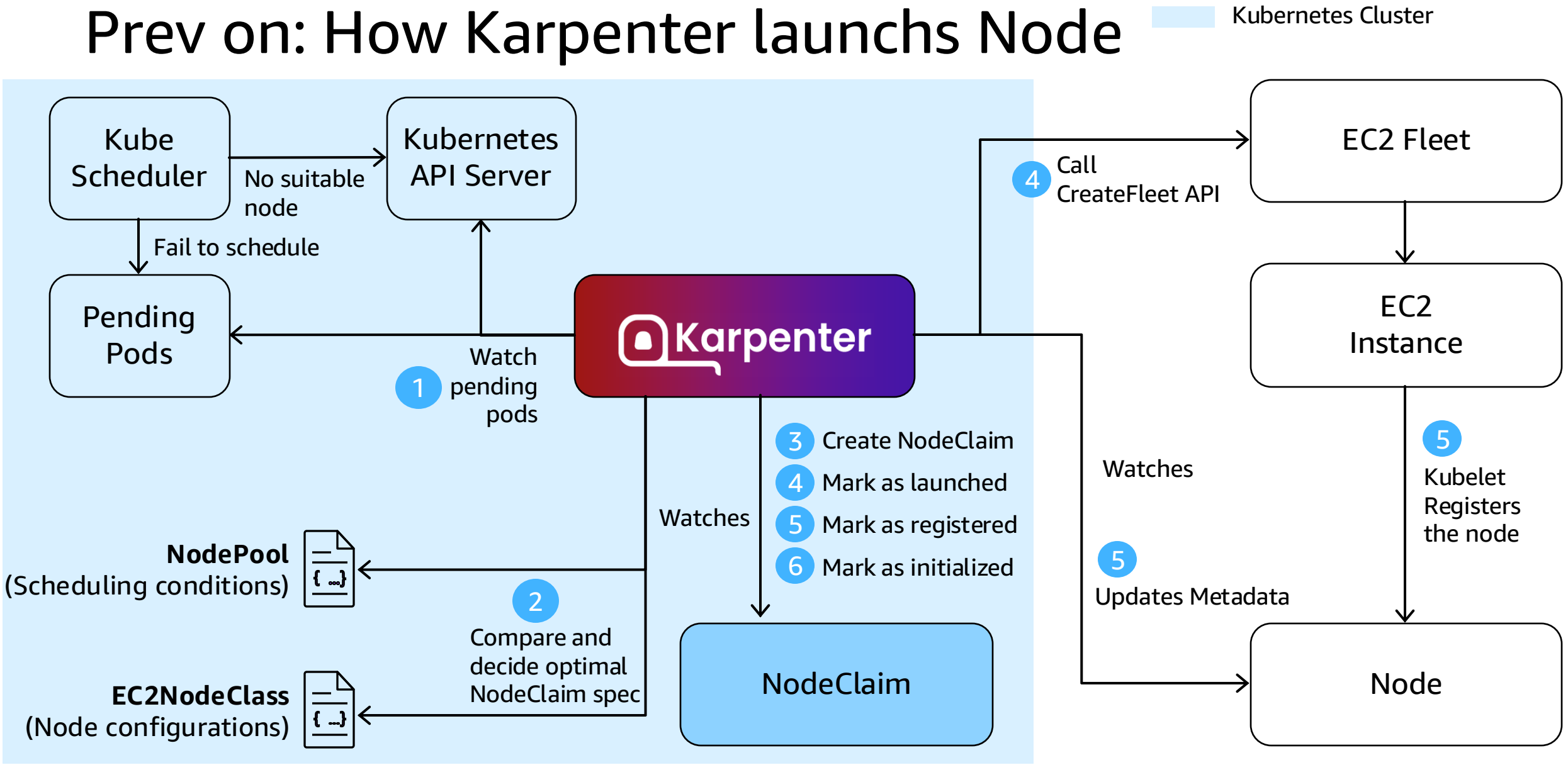
Demo

Karpenter Workshop https://catalog.workshops.aws/karpenter/en-US/

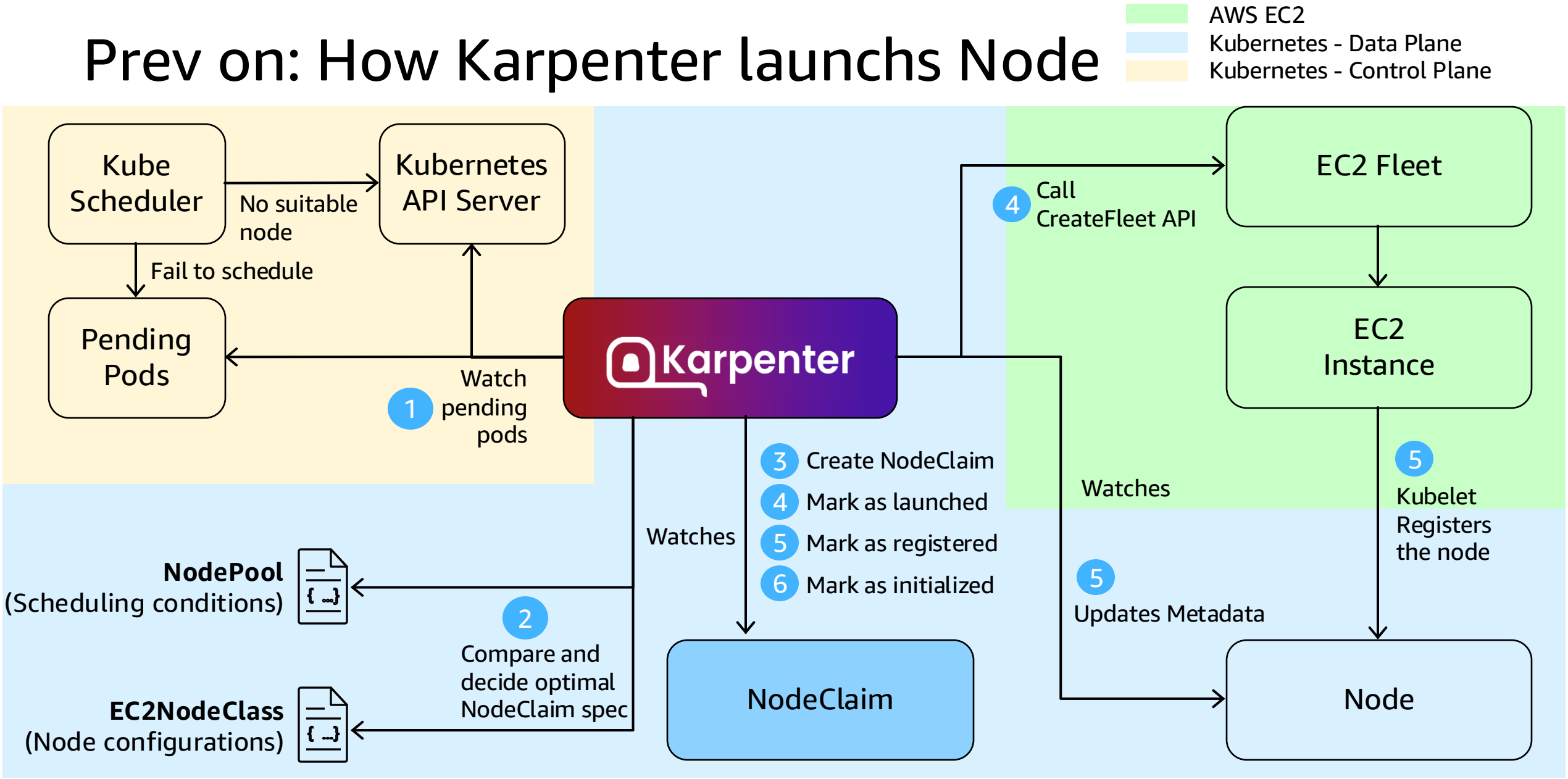
Karpenter Workshop → Basic NodePool

- Karpenter 의 기본 동작 - Scaling and Limit Resources
- Karpenter 의 Disription - Drift and Consolidation

Prev on: How Karpenter launchs Node

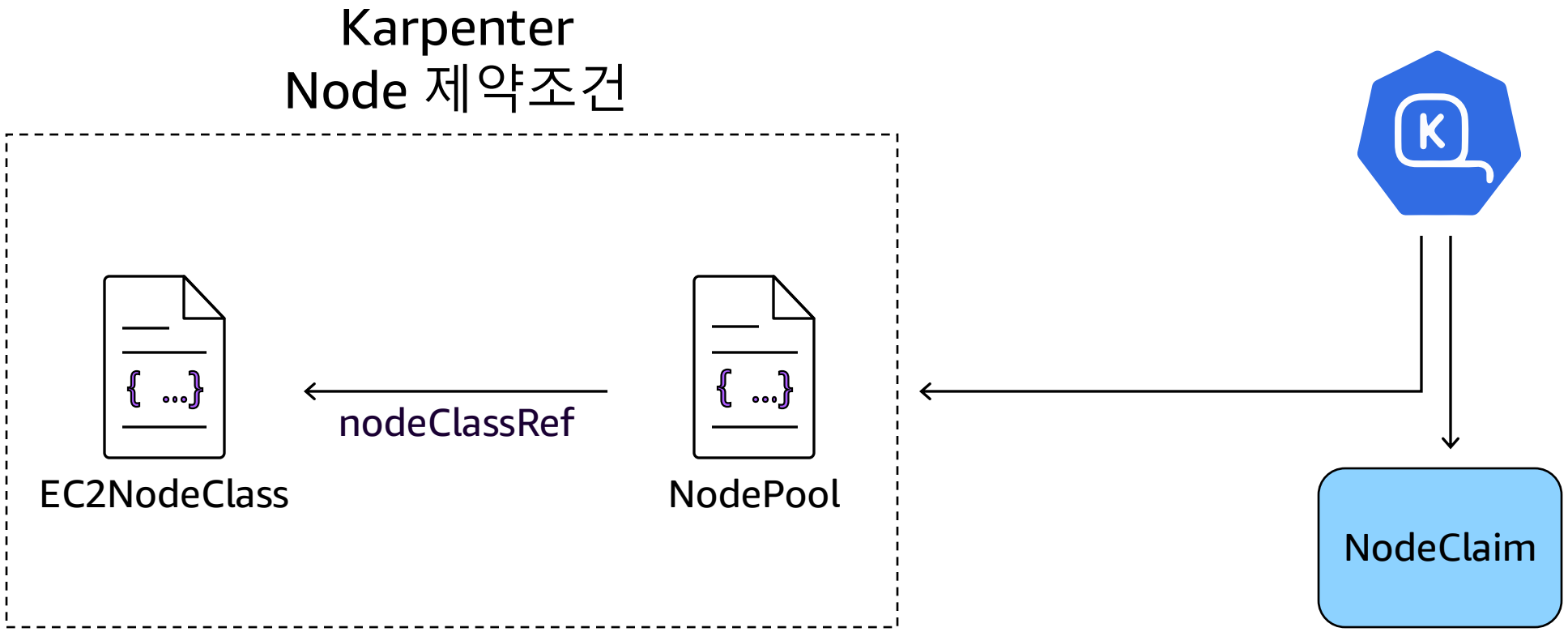


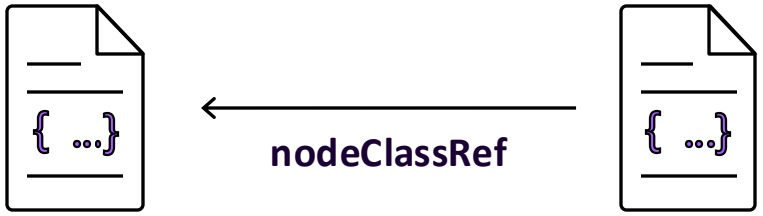
Prev on: How Karpenter launchs Node



Demo:
Scaling and Limit Resources

Prev on: Karpenter CRDs Relationship



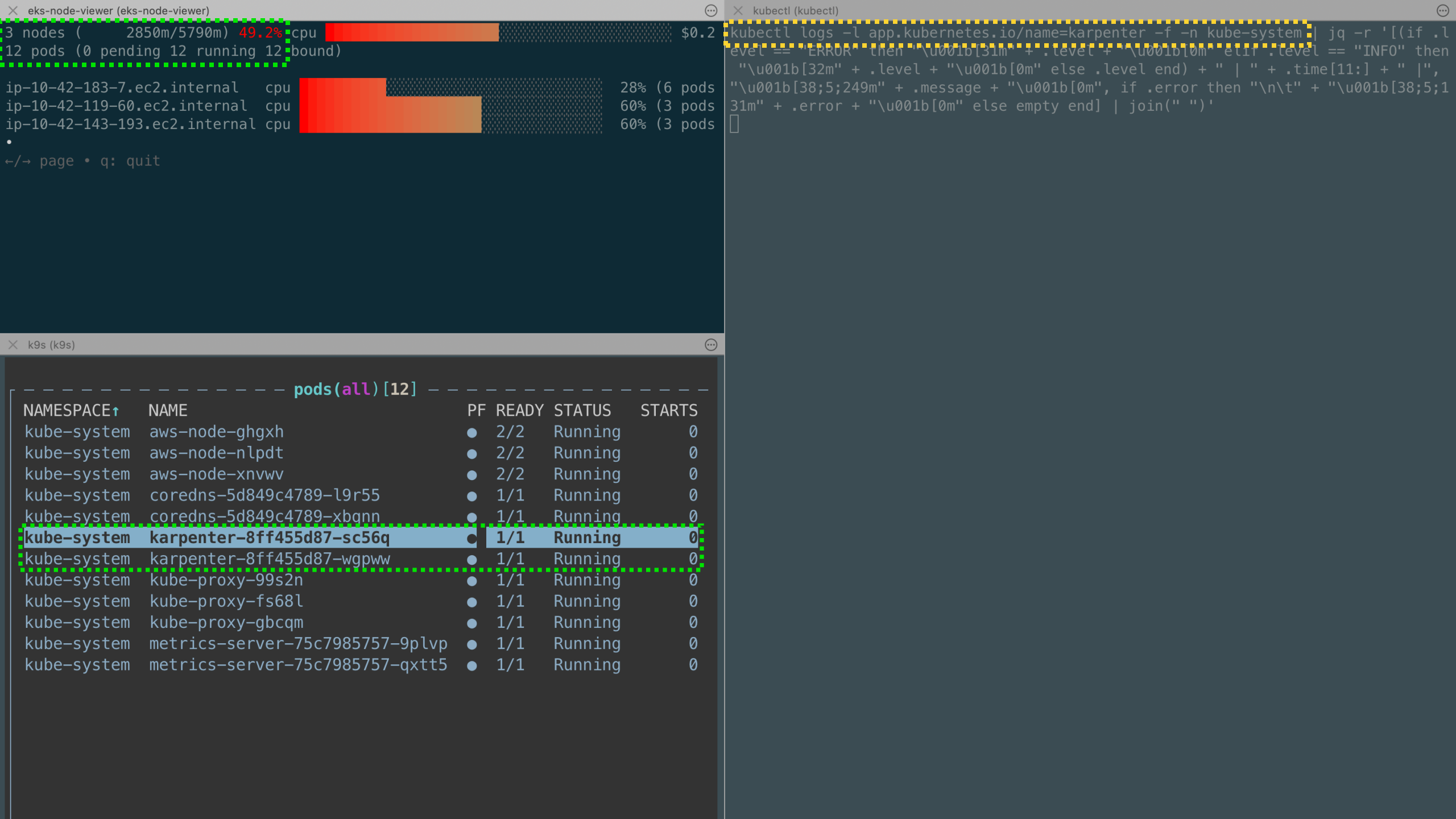


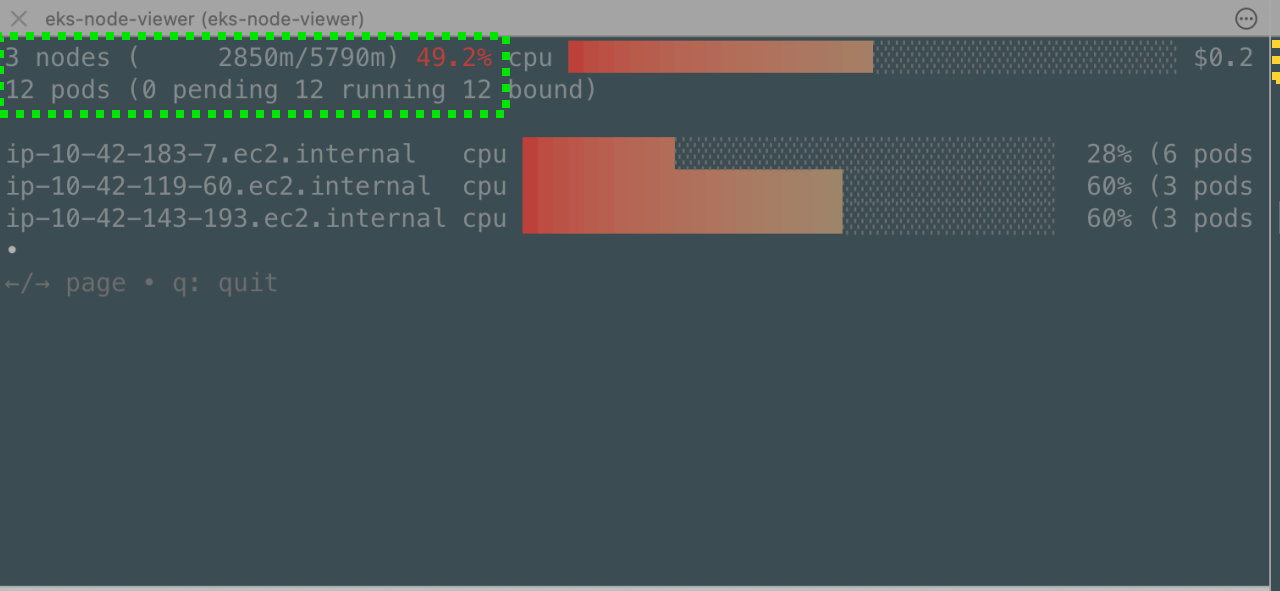
EC2NodeClass
(Node configurations)

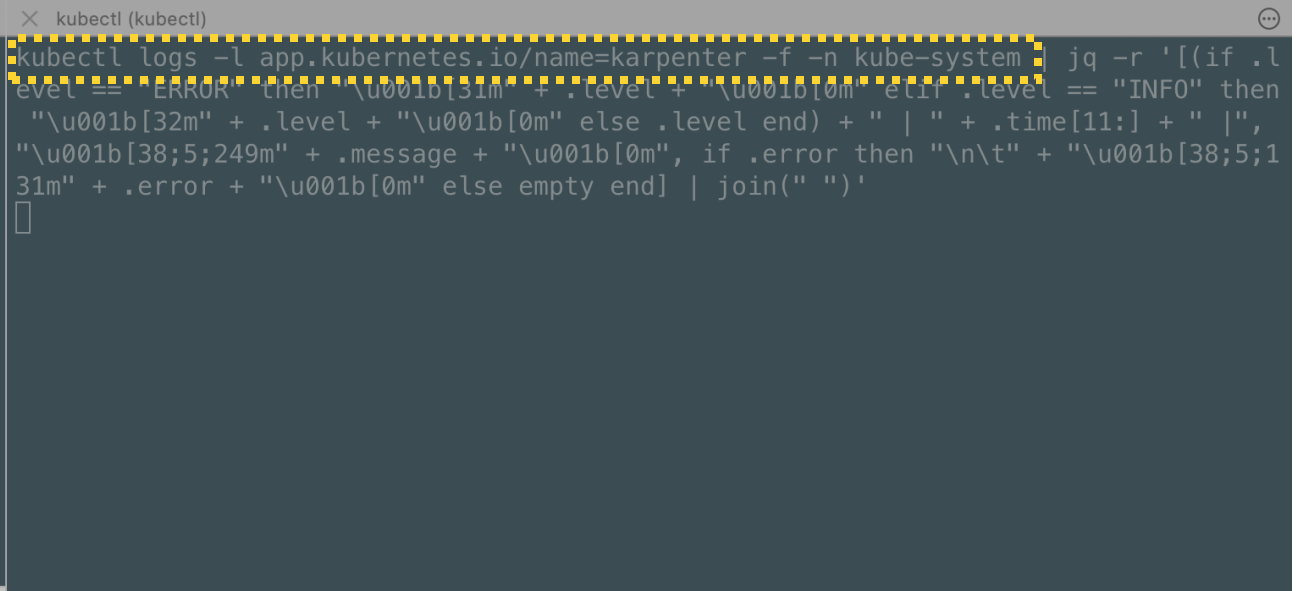
NodePool
(Scheduling conditions)

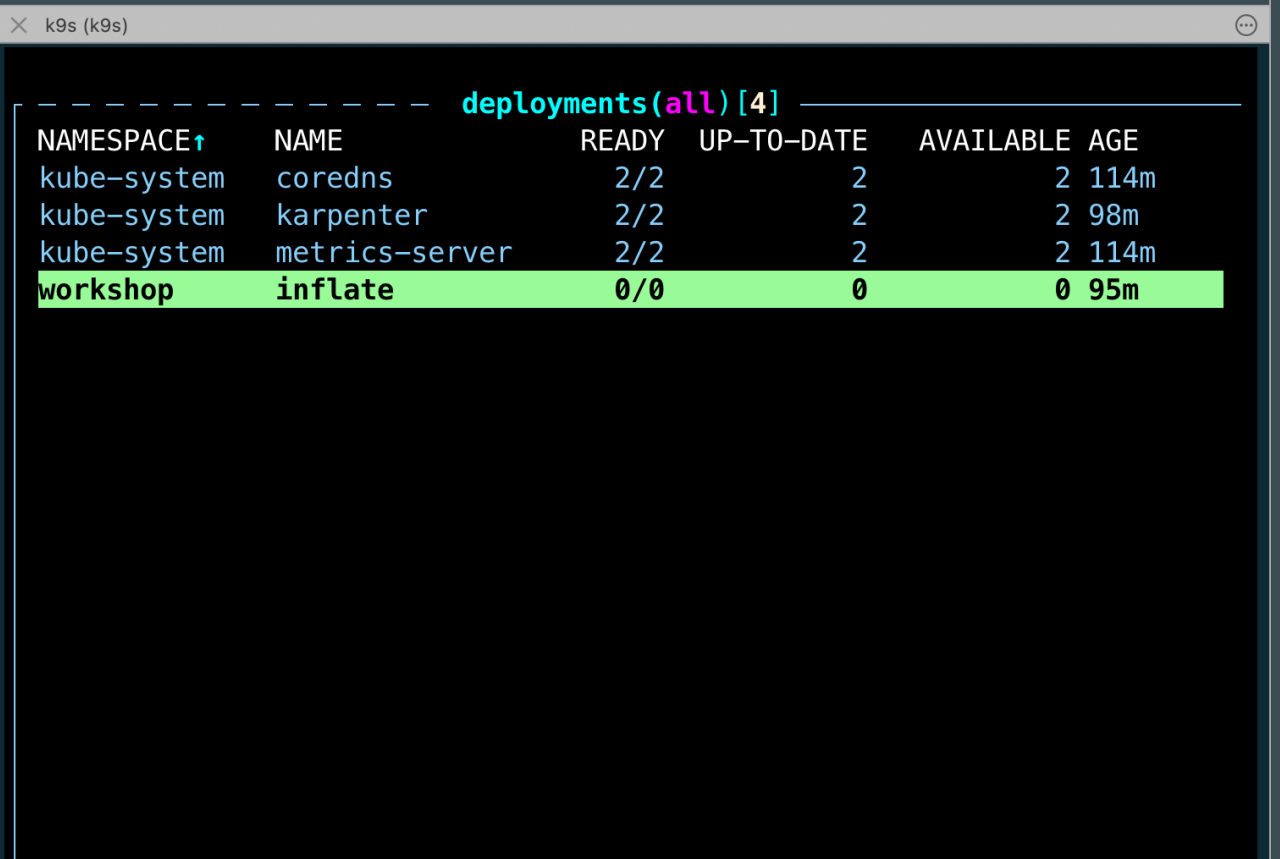
```yaml
apiVersion: karpenter.k8s.aws/v1
kind: EC2NodeClass
metadata:
  name: default
spec:
  amiSelectorTerms:
    - alias: al2023@latest
  role: karpenter-node-role
  securityGroupSelectorTerms:
    - tags: ...
  subnetSelectorTerms:
    - tags: ...
  tags:
    ...
```

```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: default
      expireAfter: Never
      requirements:
        - key: karpenter.k8s.aws/instance-category
          ...
        - key: kubernetes.io/arch
          ...
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30s
  limits:
    cpu: "10"
```


eks-node-viewer (eks-node-viewer)
3 nodes (2850m/5790m) 49.2% cpu $0.2
12 pods (0 pending 12 running 12 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
←/→ page • q: quit
kubectl (kubectl)
kubectl logs -l app.kubernetes.io/name=karpenter -f -n kube-system | jq -r '[(if .level == "ERROR" then "\u001b[31m" + .level + "\u001b[0m" elif .level == "INFO" then "\u001b[32m" + .level + "\u001b[0m" else .level end) + " | " + .time[11:] + " |", "\u001b[38;5;249m" + .message + "\u001b[0m", if .error then "\n\t" + "\u001b[38;5;131m" + .error + "\u001b[0m" else empty end] | join(" ")'
k9s (k9s)
pods(all)[12]
NAMESPACE↑ NAME PF READY STATUS STARTS
kube-system aws-node-ghgxh 2/2 Running 0
kube-system aws-node-nlpdt 2/2 Running 0
kube-system aws-node-xnvwv 2/2 Running 0
kube-system coredns-5d849c4789-l9r55 1/1 Running 0
kube-system coredns-5d849c4789-xbqnn 1/1 Running 0
kube-system karpenter-8ff455d87-sc56q 1/1 Running 0
kube-system karpenter-8ff455d87-wgpww 1/1 Running 0
kube-system kube-proxy-99s2n 1/1 Running 0
kube-system kube-proxy-fs68l 1/1 Running 0
kube-system kube-proxy-gbcqm 1/1 Running 0
kube-system metrics-server-75c7985757-9plvp 1/1 Running 0
kube-system metrics-server-75c7985757-qxtt5 1/1 Running 0


eks-node-viewer (eks-node-viewer)
3 nodes (2850m/5790m) 49.2% cpu $0.2
12 pods (0 pending 12 running 12 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
←/→ page • q: quit


kubectl (kubectl)
kubectl logs -l app.kubernetes.io/name=karpenter -f -n kube-system | jq -r '[(if .level == "ERROR" then "\u001b[31m" + .level + "\u001b[0m" elif .level == "INFO" then "\u001b[32m" + .level + "\u001b[0m" else .level end) + " | " + .time[11:] + " |", "\u001b[38;5;249m" + .message + "\u001b[0m", if .error then "\n\t" + "\u001b[38;5;131m" + .error + "\u001b[0m" else empty end] | join(" ")'


k9s (k9s)
deployments(all)[4]
NAMESPACE↑ NAME READY UP-TO-DATE AVAILABLE AGE
kube-system coredns 2/2 2 2 114m
kube-system karpenter 2/2 2 2 98m
kube-system metrics-server 2/2 2 2 114m
workshop inflate 0/0 0 0 95m

eks-node-viewer (eks-node-viewer)

```
3 nodes (     2850m/5790m) 49.2% cpu                                    $0.2
12 pods (0 pending 12 running 12 bound)

ip-10-42-183-7.ec2.internal   cpu                                  28% (6 pods
ip-10-42-119-60.ec2.internal  cpu                                  60% (3 pods
ip-10-42-143-193.ec2.internal cpu                                  60% (3 pods
•
←/→ page • q: quit
```

kubectl (kubectl)

```
kubectl logs -l app.kubernetes.io/name=karpenter -f -n kube-system | jq -r '[(if .l
evel == "ERROR" then "\u001b[31m" + .level + "\u001b[0m" elif .level == "INFO" then
 "\u001b[32m" + .level + "\u001b[0m" else .level end) + " | " + .time[11:] + " |",
"\u001b[38;5;249m" + .message + "\u001b[0m", if .error then "\n\t" + "\u001b[38;5;1
31m" + .error + "\u001b[0m" else empty end] | join(" ")'
```

k9s (k9s)

```
- - - - - - - - - - - Describe(workshop/inflate) ----------------------
Selector:               app=inflate
Replicas:               0 desired | 0 updated | 0 total | 0 available
| 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=inflate
  Containers:
   inflate:
    Image:      public.ecr.aws/eks-distro/kubernetes/pause:3.7
    Port:       <none>
    Host Port:  <none>
    Requests:
      cpu:         1
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  eks-immersion-team=my-team
  Tolerations:     <none>
```

eks-node-viewer (eks-node-viewer)

```
3 nodes (     2850m/5790m) 49.2% cpu                                  $0.2
12 pods (0 pending 12 running 12 bound)

ip-10-42-183-7.ec2.internal   cpu                                  28% (6 pods
ip-10-42-119-60.ec2.internal  cpu                                  60% (3 pods
ip-10-42-143-193.ec2.internal cpu                                  60% (3 pods
•
←/→ page • q: quit
```

k9s (k9s)

<ec2nodeclass>

Describe(-/def u t)

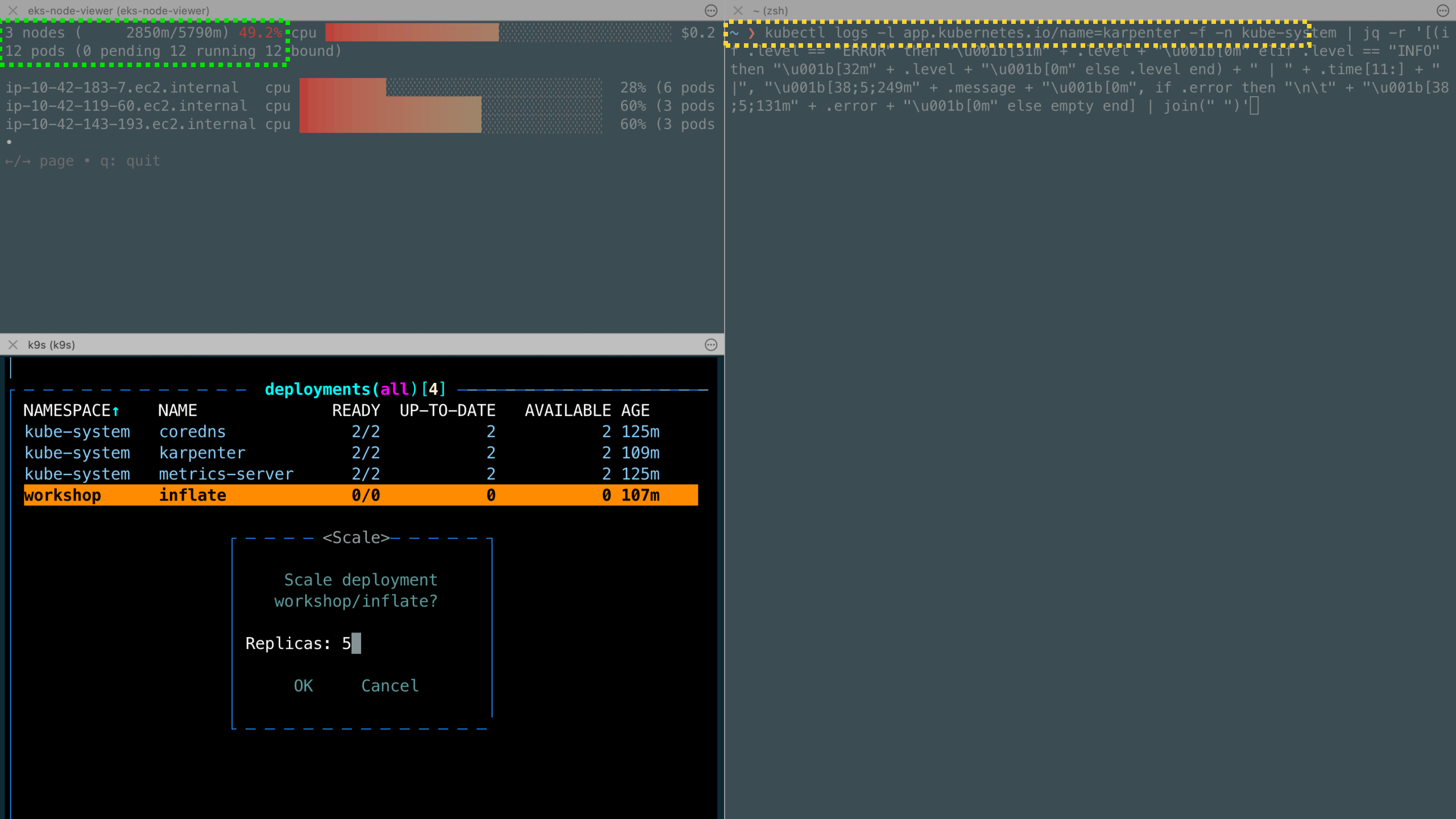
```
Name:         default
Namespace:
Labels:       <none>
Annotations:  karpenter.k8s.aws/ec2nodeclass-hash: 531922910520054505
              karpenter.k8s.aws/ec2nodeclass-hash-version: v4
API Version:  karpenter.k8s.aws/v1
Kind:         EC2NodeClass
Metadata:
  Creation Timestamp:  2025-09-16T03:17:12Z
  Finalizers:
    karpenter.k8s.aws/termination
  Generation:        1
  Resource Version:  27706
  UID:               d3c38afc-a20a-4b01-b7fe-0759f44f2db8
Spec:
  Ami Selector Terms:
    Alias:  al2023@latest
  Metadata Options:
    Http Endpoint:                enabled
    httpProtocolIPv6:             disabled
```

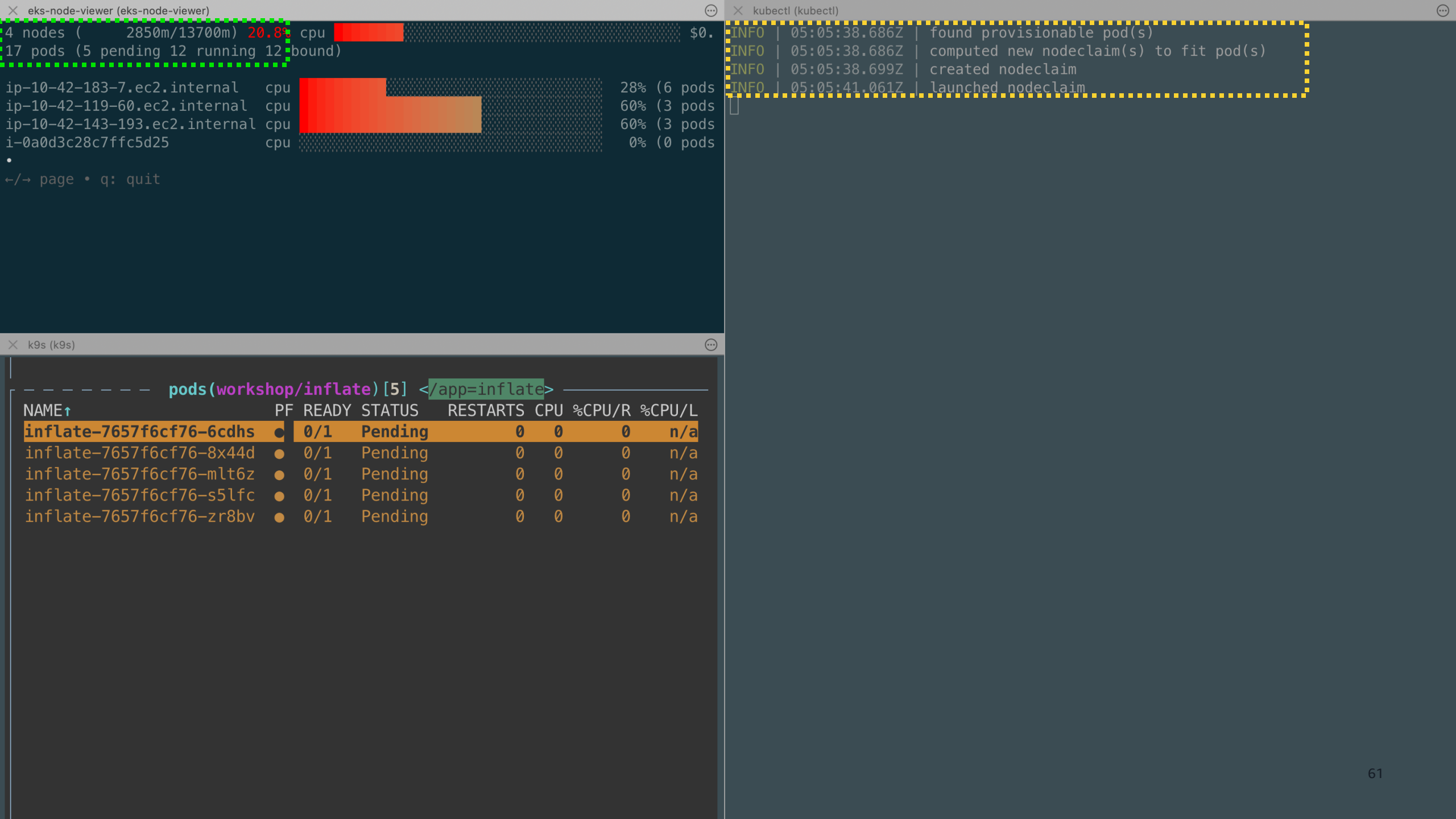
k9s (k9s)

<nodepool>

Describe(-/default)

```
Name:         default
Namespace:
Labels:       <none>
Annotations:  karpenter.sh/nodepool-hash: 16405996431342975497
              karpenter.sh/nodepool-hash-version: v3
API Version:  karpenter.sh/v1
Kind:         NodePool
Metadata:
  Creation Timestamp:  2025-09-16T03:17:11Z
  Generation:          1
  Resource Version:    26328
  UID:                 0b59bcf9-cd82-407d-8514-9f0fecbd4cb9
Spec:
  Disruption:
    Budgets:
      Nodes:                10%
    Consolidate After:      30s
    Consolidation Policy:   WhenEmptyOrUnderutilized
  Limits:
    Cpu:  10
  Template:
    Metadata:
      Labels:
        Eks - Immersion - Team:  my-team
    Spec:
      Expire After:  Never
      Node Class Ref:
        Group:  karpenter.k8s.aws
        Kind:   EC2NodeClass
        Name:   default
      Requirements:
        Key:       karpenter.k8s.aws/instance-generation
        Operator:  Gt
        Values:
          2
        Key:       karpenter.k8s.aws/instance-category
        Operator:  In
        Values:
          c
          m
          r
        Key:       kubernetes.io/arch
```


eks-node-viewer (eks-node-viewer)
3 nodes (2850m/5790m) 49.2% cpu $0.2
12 pods (0 pending 12 running 12 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
←/→ page • q: quit
~ (zsh)
~ ❯ kubectl logs -l app.kubernetes.io/name=karpenter -f -n kube-system | jq -r '[(i
f .level == "ERROR" then "\u001b[31m" + .level + "\u001b[0m" elif .level == "INFO"
then "\u001b[32m" + .level + "\u001b[0m" else .level end) + " | " + .time[11:] + "
|", "\u001b[38;5;249m" + .message + "\u001b[0m", if .error then "\n\t" + "\u001b[38
;5;131m" + .error + "\u001b[0m" else empty end] | join(" ")'
k9s (k9s)
deployments(all)[4]
NAMESPACE↑ NAME READY UP-TO-DATE AVAILABLE AGE
kube-system coredns 2/2 2 2 125m
kube-system karpenter 2/2 2 2 109m
kube-system metrics-server 2/2 2 2 125m
workshop inflate 0/0 0 0 107m
<Scale>
Scale deployment
workshop/inflate?
Replicas: 5
OK
Cancel


eks-node-viewer (eks-node-viewer)
4 nodes (2850m/13700m) 20.8% cpu $0.
17 pods (5 pending 12 running 12 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
i-0a0d3c28c7ffc5d25 cpu 0% (0 pods
←/→ page • q: quit
kubectl (kubectl)
INFO | 05:05:38.686Z | found provisionable pod(s)
INFO | 05:05:38.686Z | computed new nodeclaim(s) to fit pod(s)
INFO | 05:05:38.699Z | created nodeclaim
INFO | 05:05:41.061Z | launched nodeclaim
k9s (k9s)
pods(workshop/inflate)[5] </app=inflate>
NAME↑ PF READY STATUS RESTARTS CPU %CPU/R %CPU/L
inflate-7657f6cf76-6cdhs 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-8x44d 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-mlt6z 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-s5lfc 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-zr8bv 0/1 Pending 0 0 0 n/a

eks-node-viewer (eks-node-viewer)

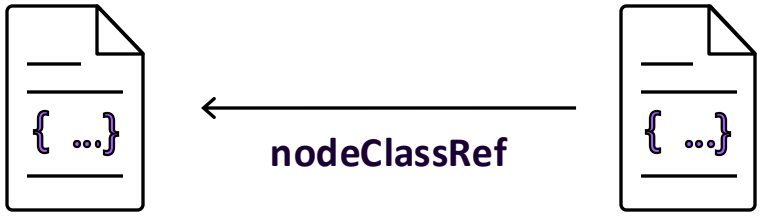
```
4 nodes (         8/13700m) 58.4% cpu                                  $0.
19 pods (0 pending 19 running 19 bound)

ip-10-42-183-7.ec2.internal   cpu                                  28% (6 pods
ip-10-42-119-60.ec2.internal  cpu                                  60% (3 pods
ip-10-42-143-193.ec2.internal cpu                                  60% (3 pods
ip-10-42-56-123.ec2.internal  cpu                                  65% (7 pods
•
←/→ page • q: quit
```

kubectl (kubectl)

```
INFO | 05:05:38.686Z | found provisionable pod(s)
INFO | 05:05:38.686Z | computed new nodeclaim(s) to fit pod(s)
INFO | 05:05:38.699Z | created nodeclaim
INFO | 05:05:41.061Z | launched nodeclaim
INFO | 05:06:00.291Z | registered nodeclaim
INFO | 05:06:13.560Z | initialized nodeclaim
```

k9s (k9s)

```
pods(workshop/inflate)[5] </app=inflate>
NAME↑                          PF READY STATUS    RESTARTS CPU %CPU/R %CPU/L
inflate-7657f6cf76-6cdhs       ● 1/1   Running          0   0     0     n/a
inflate-7657f6cf76-8x44d       ● 1/1   Running          0   0     0     n/a
inflate-7657f6cf76-mlt6z       ● 1/1   Running          0   0     0     n/a
inflate-7657f6cf76-s5lfc       ● 1/1   Running          0   0     0     n/a
inflate-7657f6cf76-zr8bv       ● 1/1   Running          0   0     0     n/a
```

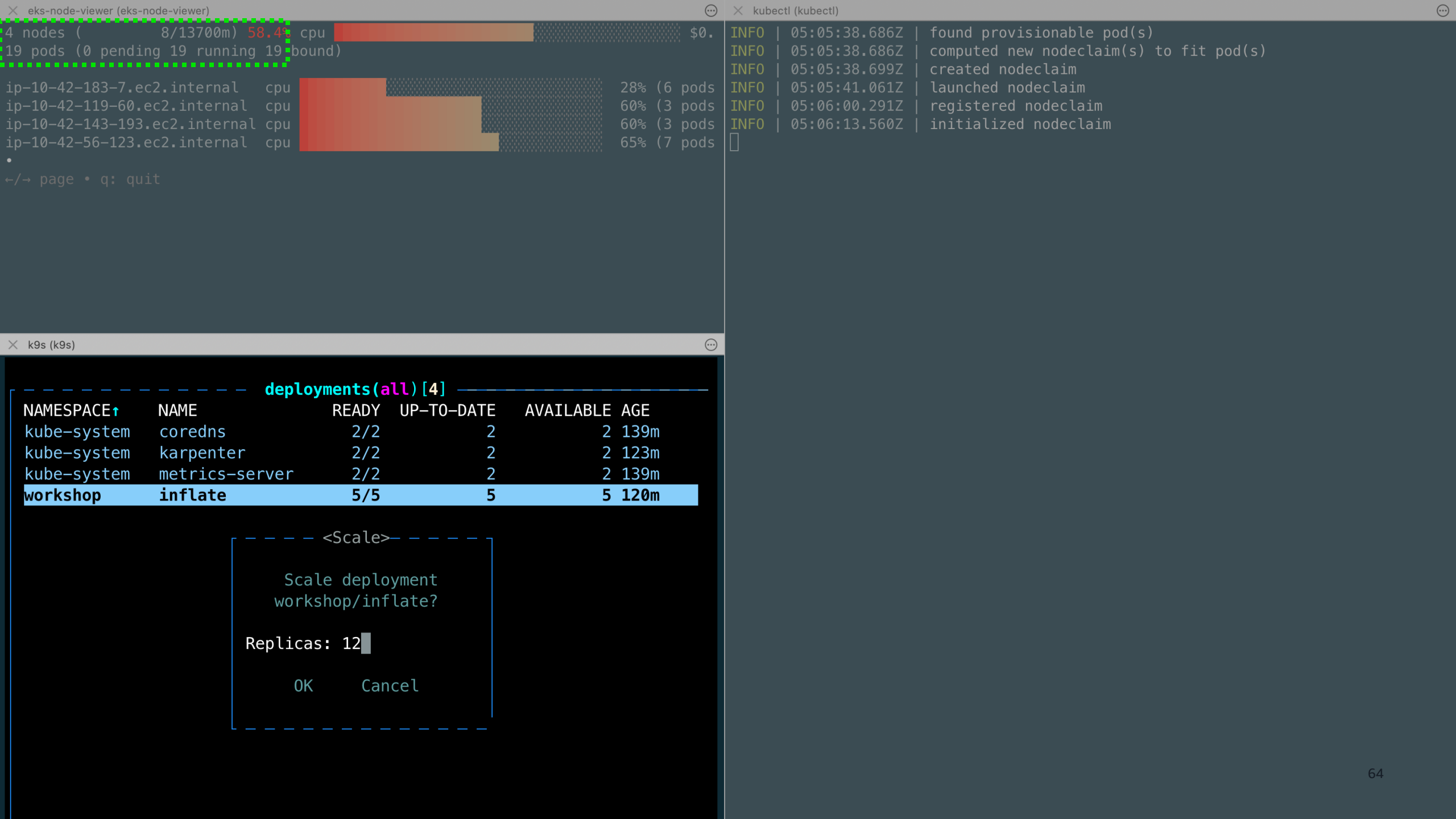


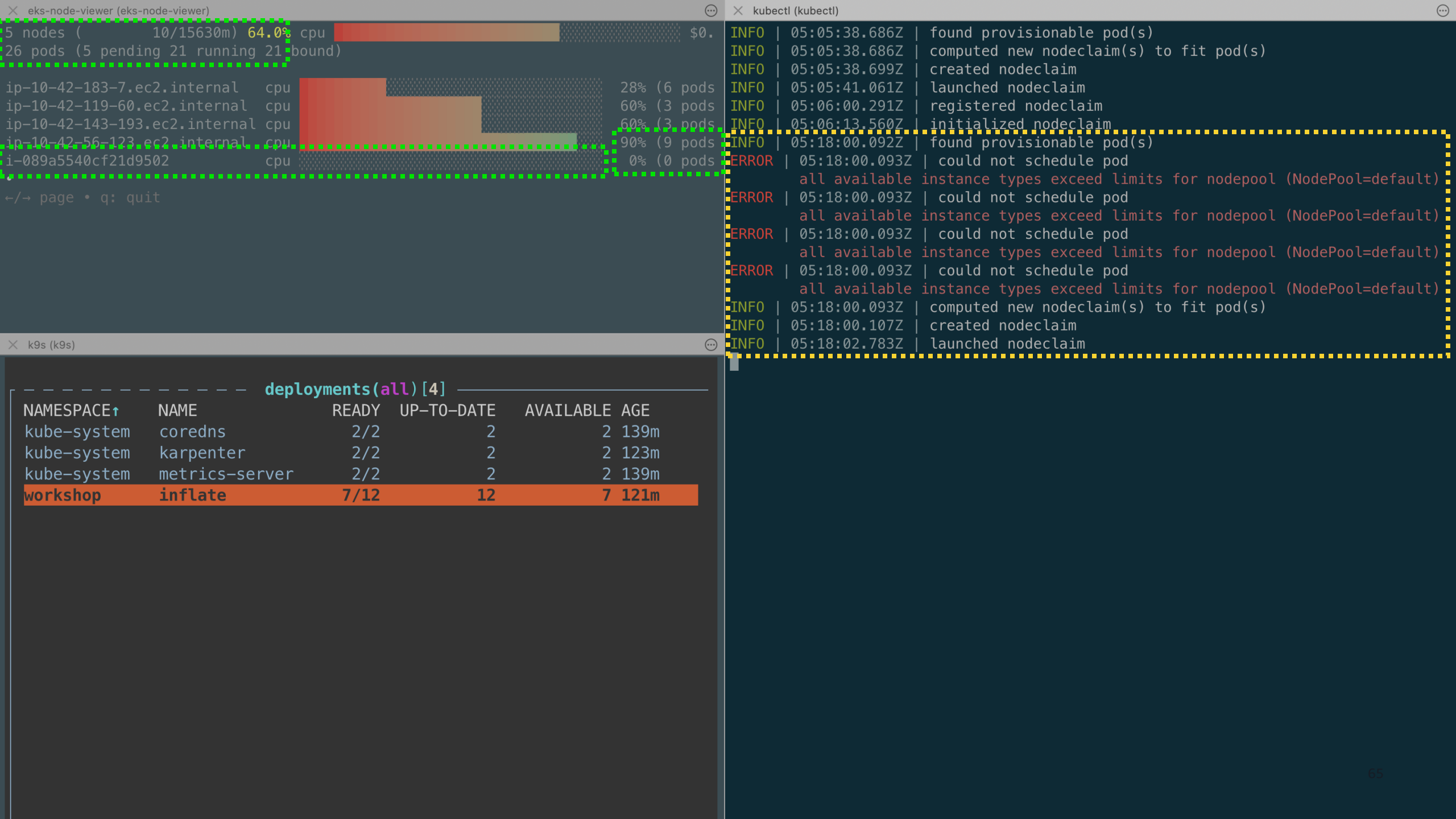
EC2NodeClass
(Node configurations)

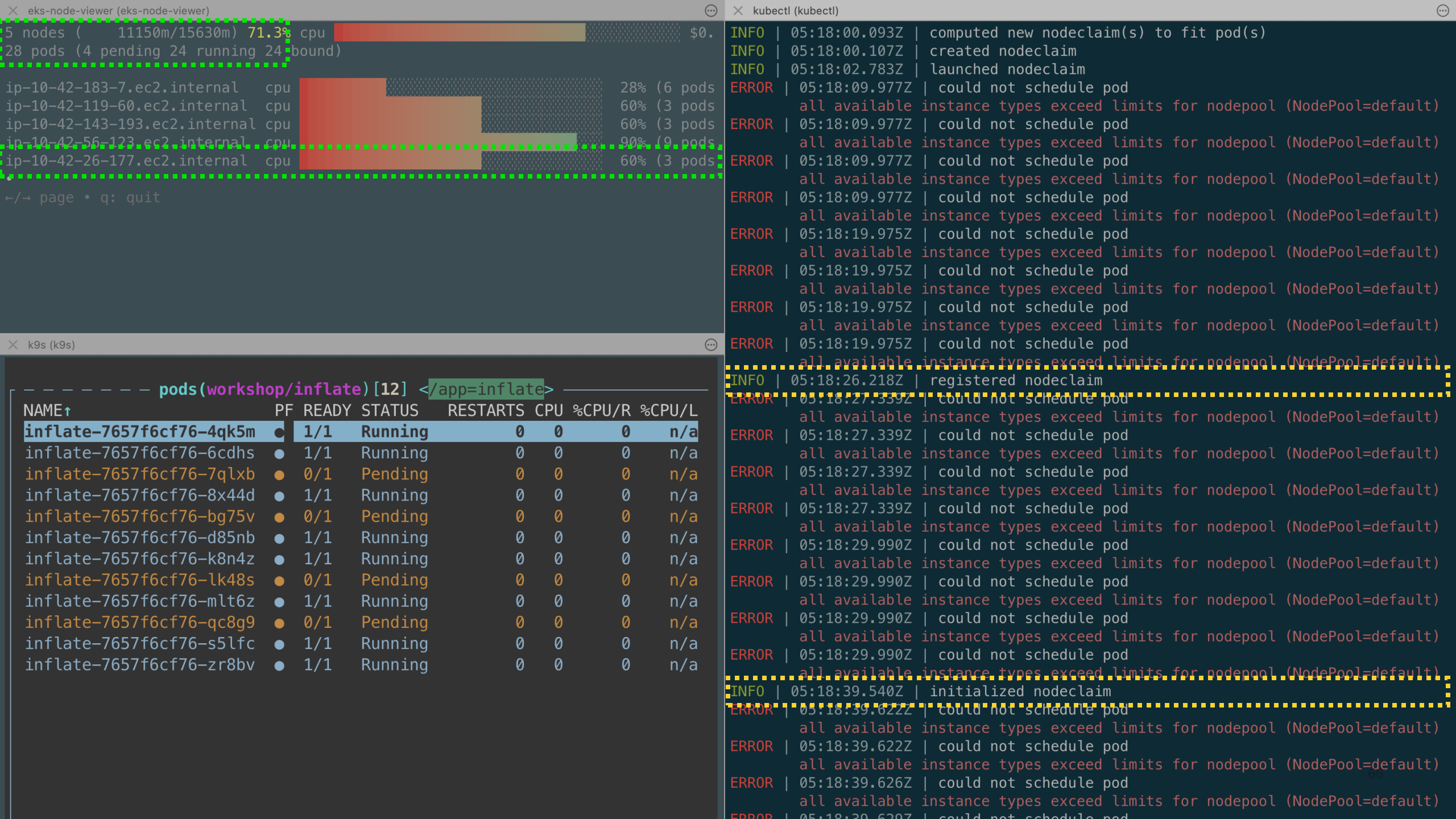
NodePool
(Scheduling conditions)

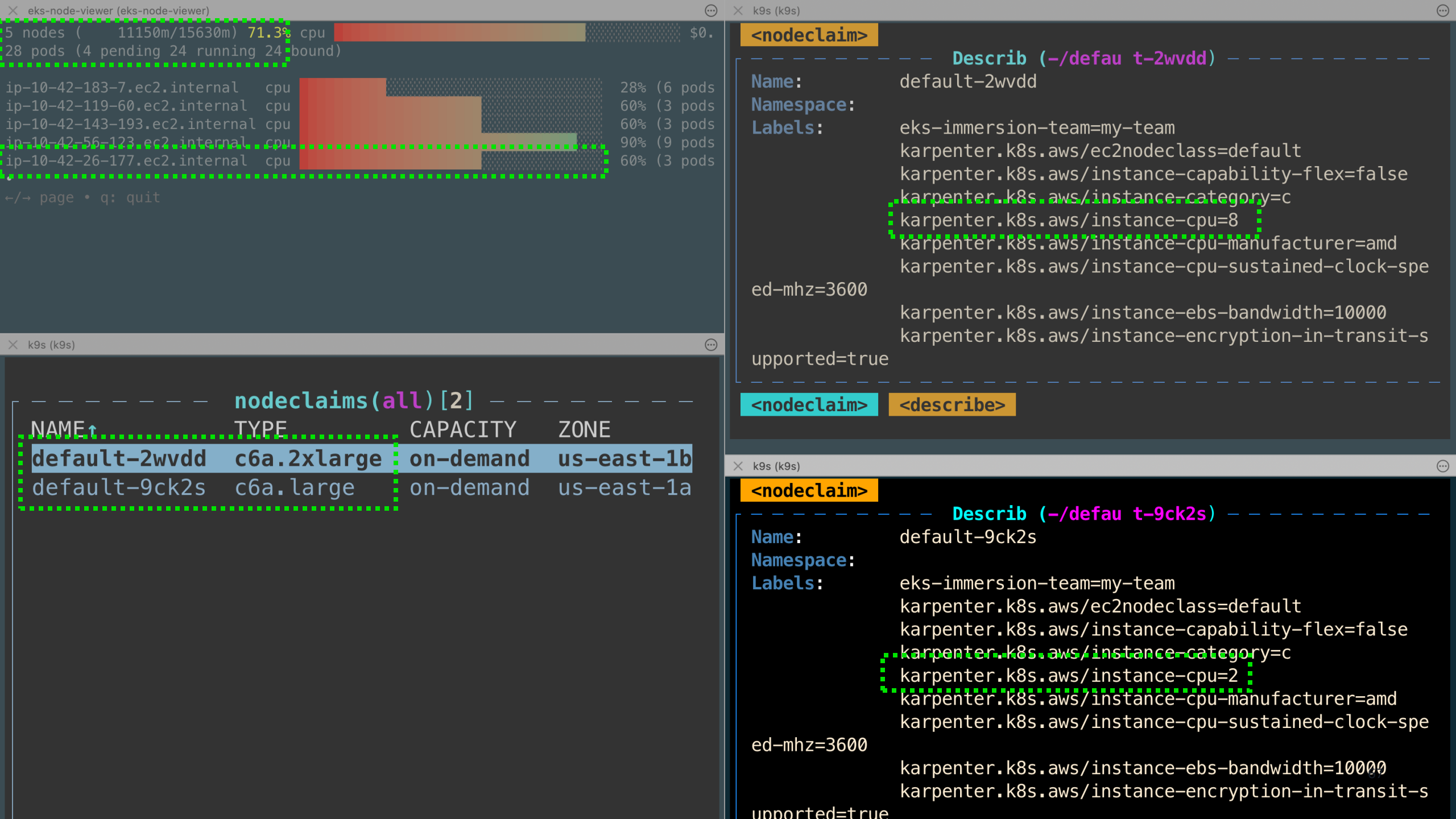
```yaml
apiVersion: karpenter.k8s.aws/v1
kind: EC2NodeClass
metadata:
  name: default
spec:
  amiSelectorTerms:
    - alias: al2023@latest
  role: karpenter-node-role
  securityGroupSelectorTerms:
    - tags: ...
  subnetSelectorTerms:
    - tags: ...
  tags:
    ...
```

```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: default
      expireAfter: Never
      requirements:
        - key: karpenter.k8s.aws/instance-category
          ...
        - key: kubernetes.io/arch
          ...
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30s
  limits:
    cpu: "10"
```

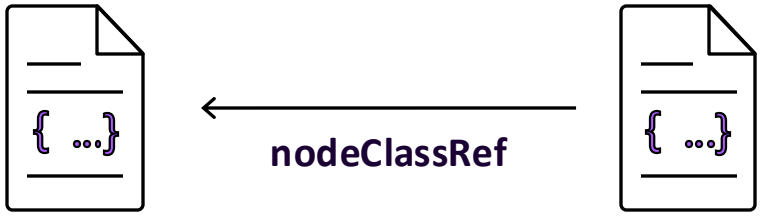

eks-node-viewer (eks-node-viewer)
4 nodes (8/13700m) 58.4% cpu $0.
19 pods (0 pending 19 running 19 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-56-123.ec2.internal cpu 65% (7 pods
←/→ page • q: quit
kubectl (kubectl)
INFO | 05:05:38.686Z | found provisionable pod(s)
INFO | 05:05:38.686Z | computed new nodeclaim(s) to fit pod(s)
INFO | 05:05:38.699Z | created nodeclaim
INFO | 05:05:41.061Z | launched nodeclaim
INFO | 05:06:00.291Z | registered nodeclaim
INFO | 05:06:13.560Z | initialized nodeclaim
k9s (k9s)
deployments(all)[4]
NAMESPACE↑ NAME READY UP-TO-DATE AVAILABLE AGE
kube-system coredns 2/2 2 2 139m
kube-system karpenter 2/2 2 2 123m
kube-system metrics-server 2/2 2 2 139m
workshop inflate 5/5 5 5 120m
<Scale>
Scale deployment
workshop/inflate?
Replicas: 12
OK Cancel


eks-node-viewer (eks-node-viewer)
5 nodes (10/15630m) 64.0% cpu $0.
26 pods (5 pending 21 running 21 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-56-123.ec2.internal cpu 90% (9 pods
i-089a5540cf21d9502 cpu 0% (0 pods
←/→ page • q: quit
kubectl (kubectl)
INFO | 05:05:38.686Z | found provisionable pod(s)
INFO | 05:05:38.686Z | computed new nodeclaim(s) to fit pod(s)
INFO | 05:05:38.699Z | created nodeclaim
INFO | 05:05:41.061Z | launched nodeclaim
INFO | 05:06:00.291Z | registered nodeclaim
INFO | 05:06:13.560Z | initialized nodeclaim
INFO | 05:18:00.092Z | found provisionable pod(s)
ERROR | 05:18:00.093Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:00.093Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:00.093Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:00.093Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
INFO | 05:18:00.093Z | computed new nodeclaim(s) to fit pod(s)
INFO | 05:18:00.107Z | created nodeclaim
INFO | 05:18:02.783Z | launched nodeclaim
k9s (k9s)
deployments(all)[4]
NAMESPACE↑ NAME READY UP-TO-DATE AVAILABLE AGE
kube-system coredns 2/2 2 2 139m
kube-system karpenter 2/2 2 2 123m
kube-system metrics-server 2/2 2 2 139m
workshop inflate 7/12 12 7 121m


eks-node-viewer (eks-node-viewer)
5 nodes (11150m/15630m) 71.3% cpu $0.
28 pods (4 pending 24 running 24 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-56-123.ec2.internal cpu 90% (9 pods
ip-10-42-26-177.ec2.internal cpu 60% (3 pods
←/→ page • q: quit
k9s (k9s)
pods(workshop/inflate)[12] </app=inflate>
NAME↑ PF READY STATUS RESTARTS CPU %CPU/R %CPU/L
inflate-7657f6cf76-4qk5m ● 1/1 Running 0 0 0 n/a
inflate-7657f6cf76-6cdhs ● 1/1 Running 0 0 0 n/a
inflate-7657f6cf76-7qlxb ● 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-8x44d ● 1/1 Running 0 0 0 n/a
inflate-7657f6cf76-bg75v ● 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-d85nb ● 1/1 Running 0 0 0 n/a
inflate-7657f6cf76-k8n4z ● 1/1 Running 0 0 0 n/a
inflate-7657f6cf76-lk48s ● 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-mlt6z ● 1/1 Running 0 0 0 n/a
inflate-7657f6cf76-qc8g9 ● 0/1 Pending 0 0 0 n/a
inflate-7657f6cf76-s5lfc ● 1/1 Running 0 0 0 n/a
inflate-7657f6cf76-zr8bv ● 1/1 Running 0 0 0 n/a
kubectl (kubectl)
INFO | 05:18:00.093Z | computed new nodeclaim(s) to fit pod(s)
INFO | 05:18:00.107Z | created nodeclaim
INFO | 05:18:02.783Z | launched nodeclaim
ERROR | 05:18:09.977Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:09.977Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:09.977Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:09.977Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:19.975Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:19.975Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:19.975Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:19.975Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
INFO | 05:18:26.218Z | registered nodeclaim
ERROR | 05:18:27.339Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:27.339Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:27.339Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:27.339Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:29.990Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:29.990Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:29.990Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:29.990Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
INFO | 05:18:39.540Z | initialized nodeclaim
ERROR | 05:18:39.622Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:39.622Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:18:39.626Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)


eks-node-viewer (eks-node-viewer)
5 nodes (11150m/15630m) 71.3% cpu $0.
28 pods (4 pending 24 running 24 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-56-123.ec2.internal cpu 90% (9 pods
ip-10-42-26-177.ec2.internal cpu 60% (3 pods
←/→ page • q: quit
k9s (k9s)
<nodeclaim>
Describ (-/defau t-2wvdd)
Name: default-2wvdd
Namespace:
Labels: eks-immersion-team=my-team
karpenter.k8s.aws/ec2nodeclass=default
karpenter.k8s.aws/instance-capability-flex=false
karpenter.k8s.aws/instance-category=c
karpenter.k8s.aws/instance-cpu=8
karpenter.k8s.aws/instance-cpu-manufacturer=amd
karpenter.k8s.aws/instance-cpu-sustained-clock-spe
ed-mhz=3600
karpenter.k8s.aws/instance-ebs-bandwidth=10000
karpenter.k8s.aws/instance-encryption-in-transit-s
upported=true
<nodeclaim>
<describe>
k9s (k9s)
nodeclaims(all)[2]
NAME↑ TYPE CAPACITY ZONE
default-2wvdd c6a.2xlarge on-demand us-east-1b
default-9ck2s c6a.large on-demand us-east-1a
k9s (k9s)
<nodeclaim>
Describ (-/defau t-9ck2s)
Name: default-9ck2s
Namespace:
Labels: eks-immersion-team=my-team
karpenter.k8s.aws/ec2nodeclass=default
karpenter.k8s.aws/instance-capability-flex=false
karpenter.k8s.aws/instance-category=c
karpenter.k8s.aws/instance-cpu=2
karpenter.k8s.aws/instance-cpu-manufacturer=amd
karpenter.k8s.aws/instance-cpu-sustained-clock-spe
ed-mhz=3600
karpenter.k8s.aws/instance-ebs-bandwidth=10000
karpenter.k8s.aws/instance-encryption-in-transit-s
upported=true

Demo:
Drift and Consolidation



EC2NodeClass
(Node configurations)

NodePool
(Scheduling conditions)

```yaml
apiVersion: karpenter.k8s.aws/v1
kind: EC2NodeClass
metadata:
  name: default
spec:
  amiSelectorTerms:
    - alias: al2023@latest
  role: karpenter-node-role
  securityGroupSelectorTerms:
    - tags: ...
  subnetSelectorTerms:
    - tags: ...
  tags:
    ...
```

```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: default
      expireAfter: Never
      requirements:
        - key: karpenter.k8s.aws/instance-category
          ...
        - key: kubernetes.io/arch
          ...
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30s
  limits:
    cpu: "10"
```

eks-node-viewer (eks-node-viewer)

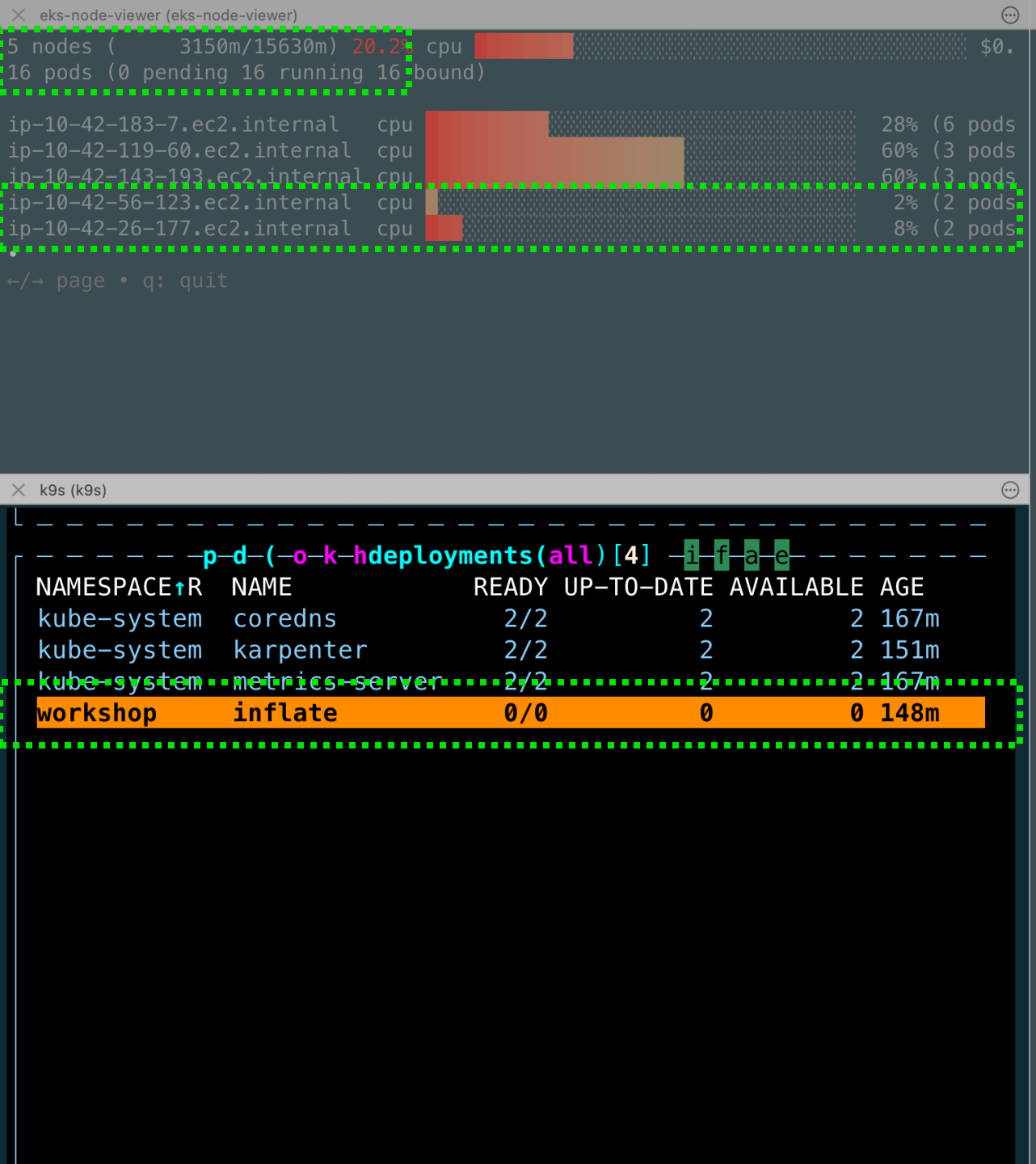
```
5 nodes (    11150m/15630m) 71.3% cpu                                  $0.
28 pods (4 pending 24 running 24 bound)

ip-10-42-183-7.ec2.internal   cpu                                  28% (6 pods
ip-10-42-119-60.ec2.internal  cpu                                  60% (3 pods
ip-10-42-143-193.ec2.internal cpu                                  60% (3 pods
ip-10-42-56-123.ec2.internal  cpu                                  90% (9 pods
ip-10-42-26-177.ec2.internal  cpu                                  60% (3 pods
•
←/→ page • q: quit
```

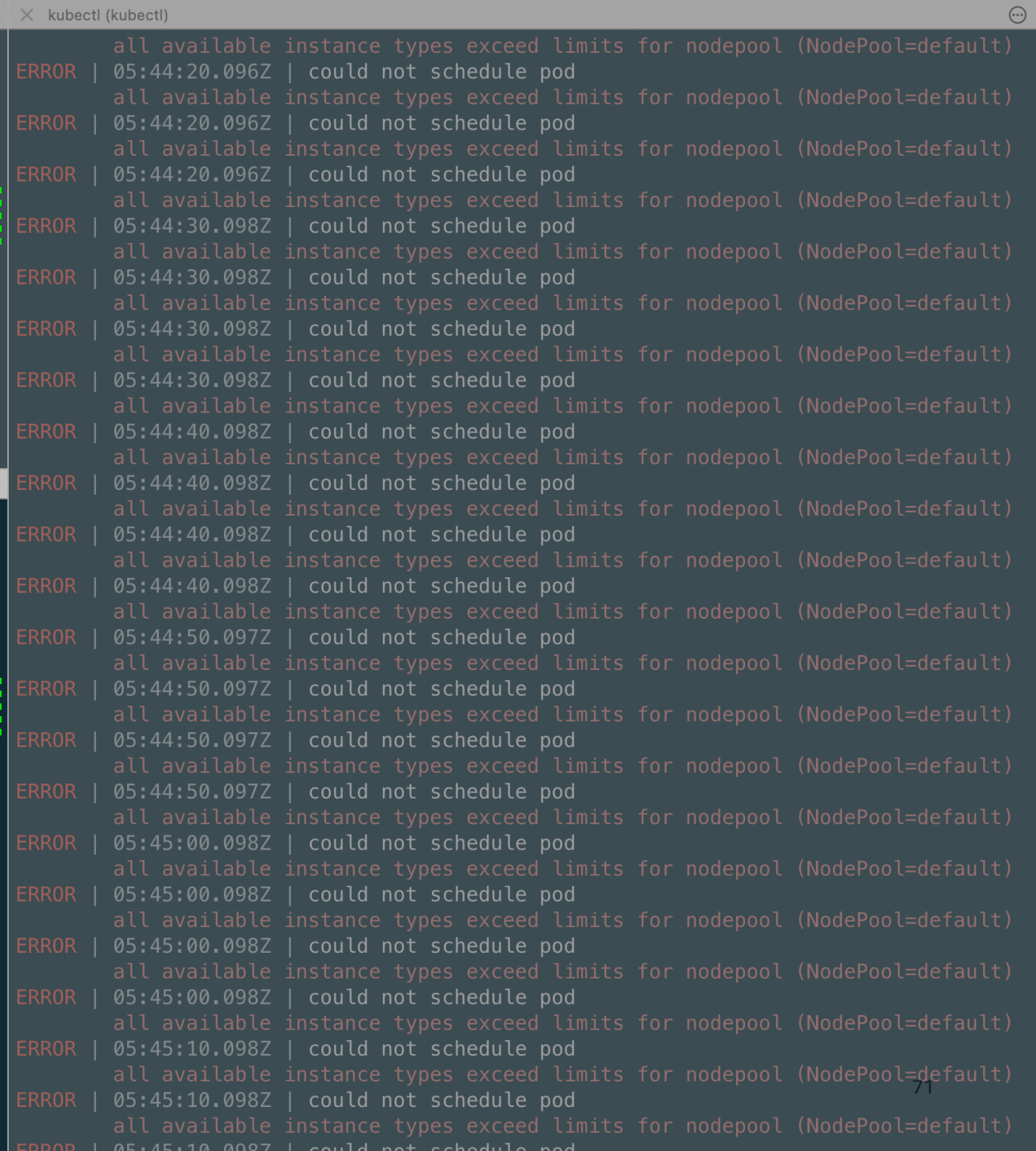
k9s (k9s)

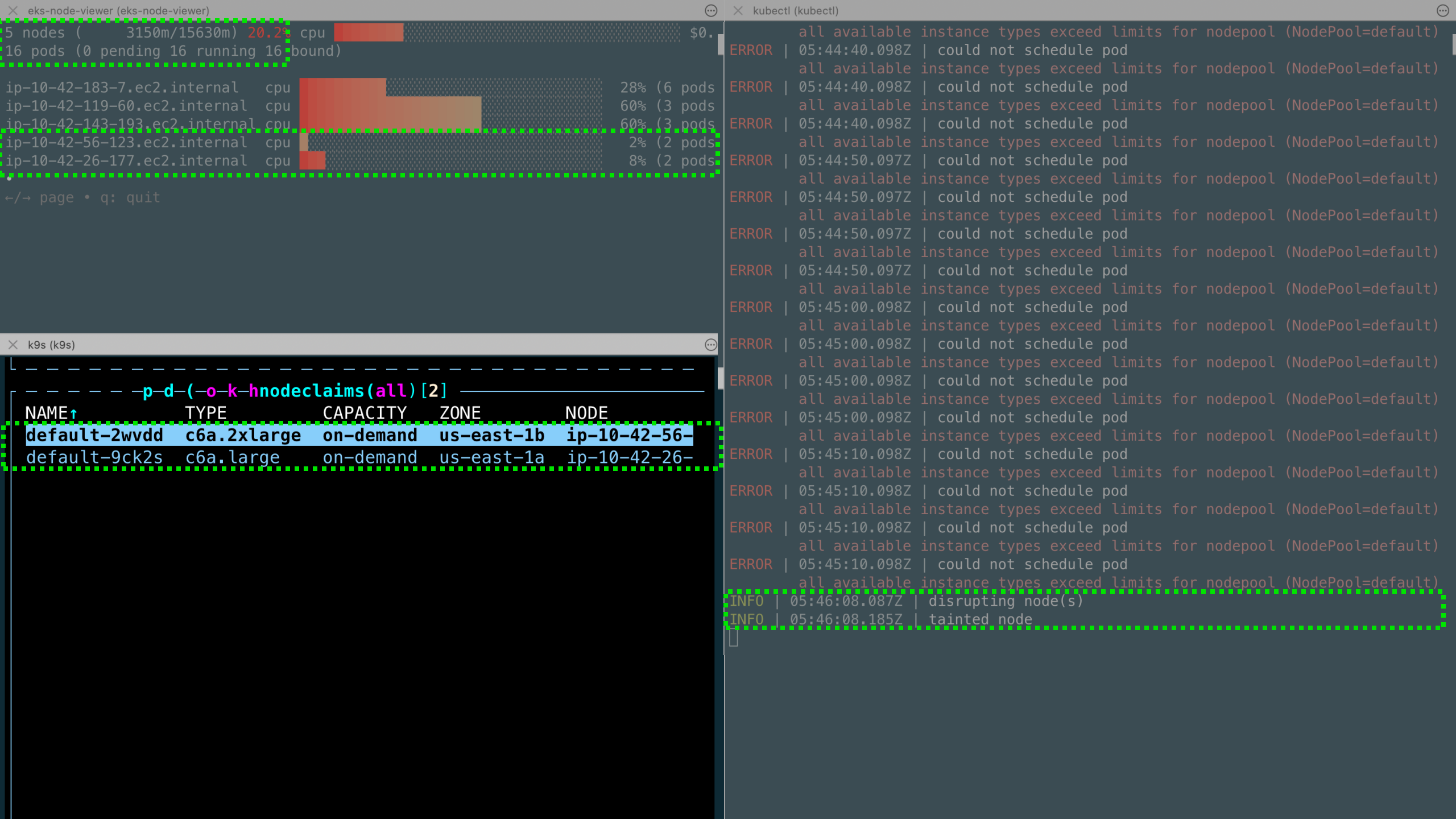
```
deployments(all)[4]
NAMESPACE↑s  NAME            READY UP-TO-DATE AVAILABLE AGE
kube-system  coredns           2/2          2         2 166m
kube-system  karpenter         2/2          2         2 150m
kube-system  metrics-server    2/2          2         2 166m
workshop     inflate          8/12         12         8 147m

          Scale>

     Scale deployment
    workshop/inflate?

  Replicas: 0

      OK     Cancel
```

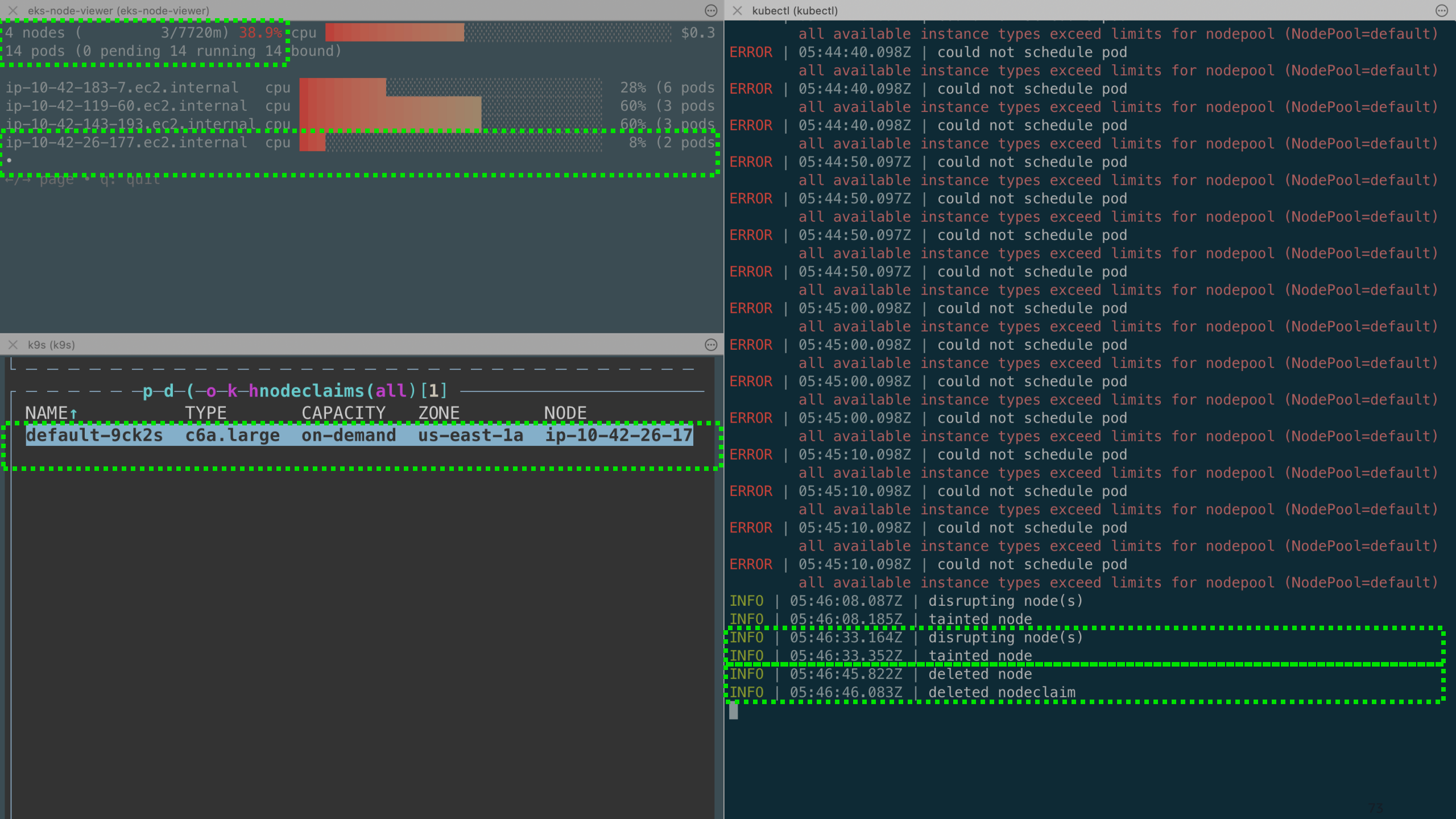
kubectl (kubectl)

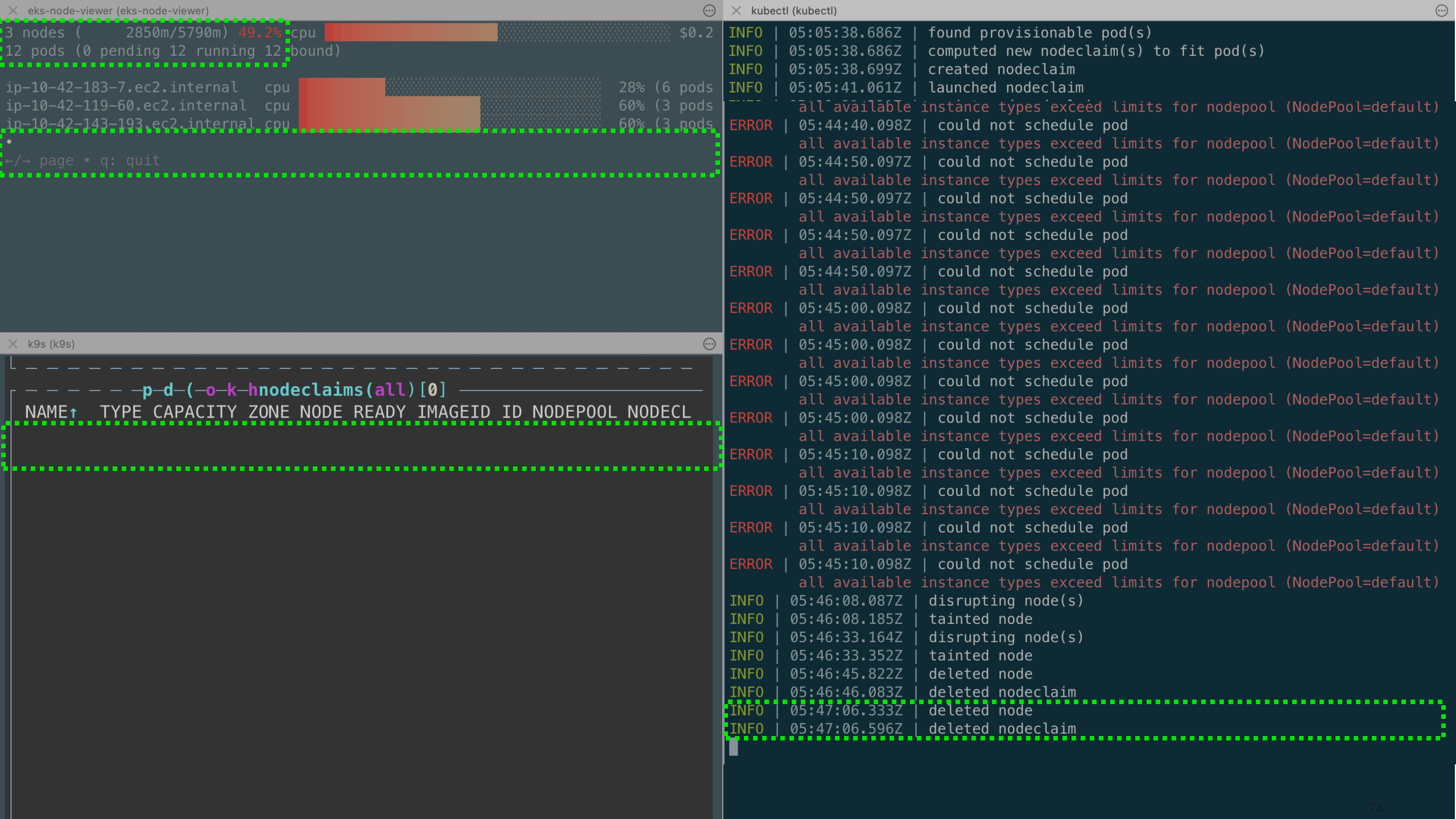
```
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:43:40.093Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:43:40.093Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:43:40.093Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:43:50.094Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:43:50.094Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:43:50.094Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:43:50.094Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:00.101Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:00.102Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:00.103Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:00.104Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:10.095Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:10.095Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:10.095Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:10.096Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:20.096Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:20.096Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:20.096Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:20.096Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:30.098Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:30.098Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:30.098Z | could not schedule pod
        all available instance types exceed limits for nodepool (NodePool=default)
```

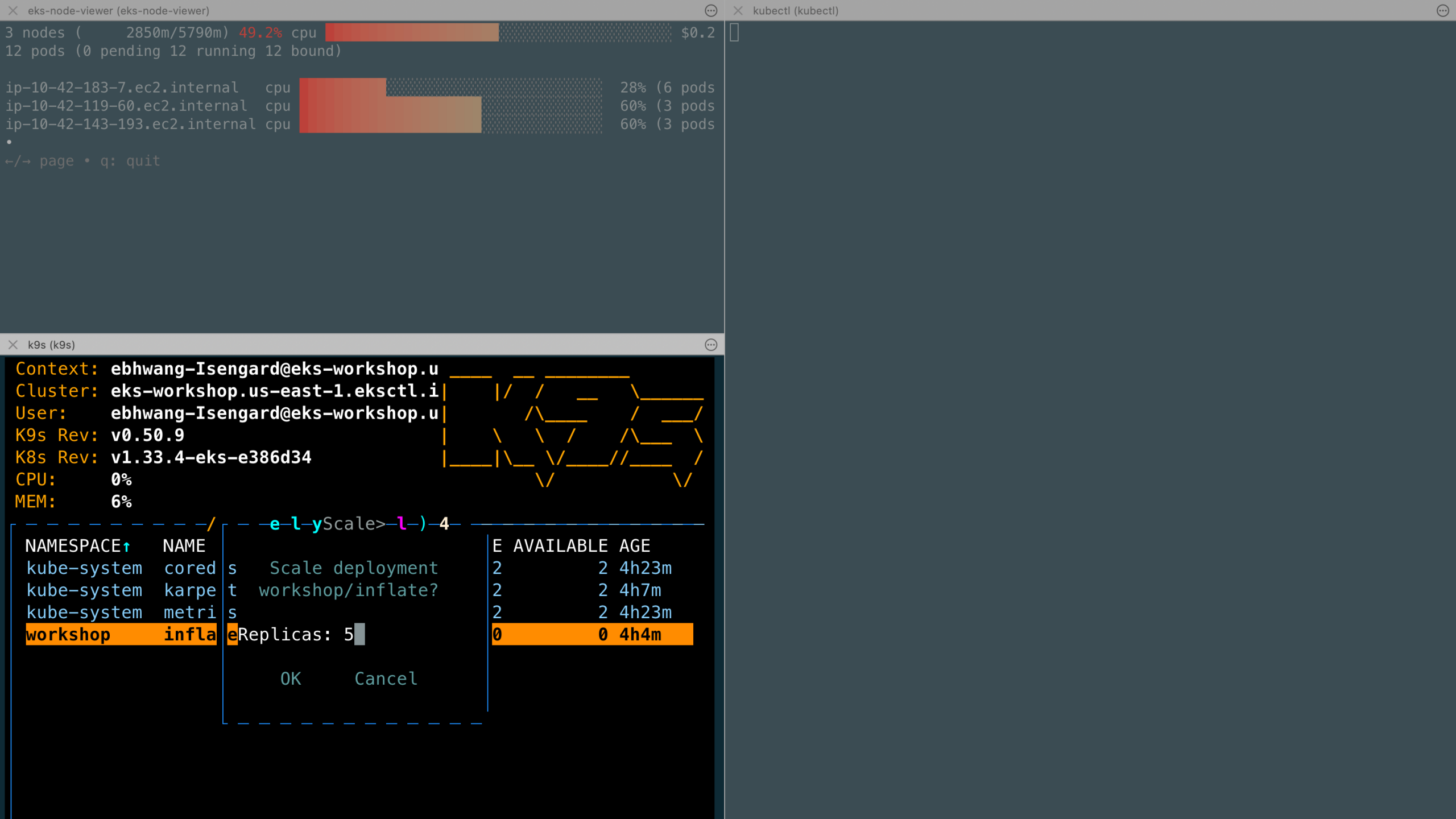

eks-node-viewer (eks-node-viewer)
5 nodes (3150m/15630m) 20.2% cpu $0.
16 pods (0 pending 16 running 16 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-56-123.ec2.internal cpu 2% (2 pods
ip-10-42-26-177.ec2.internal cpu 8% (2 pods
←/→ page • q: quit
k9s (k9s)
p d (o k hdeployments(all)[4] i f a e
NAMESPACE↑R NAME READY UP-TO-DATE AVAILABLE AGE
kube-system coredns 2/2 2 2 167m
kube-system karpenter 2/2 2 2 151m
kube-system metrics-server 2/2 2 2 167m
workshop inflate 0/0 0 0 148m

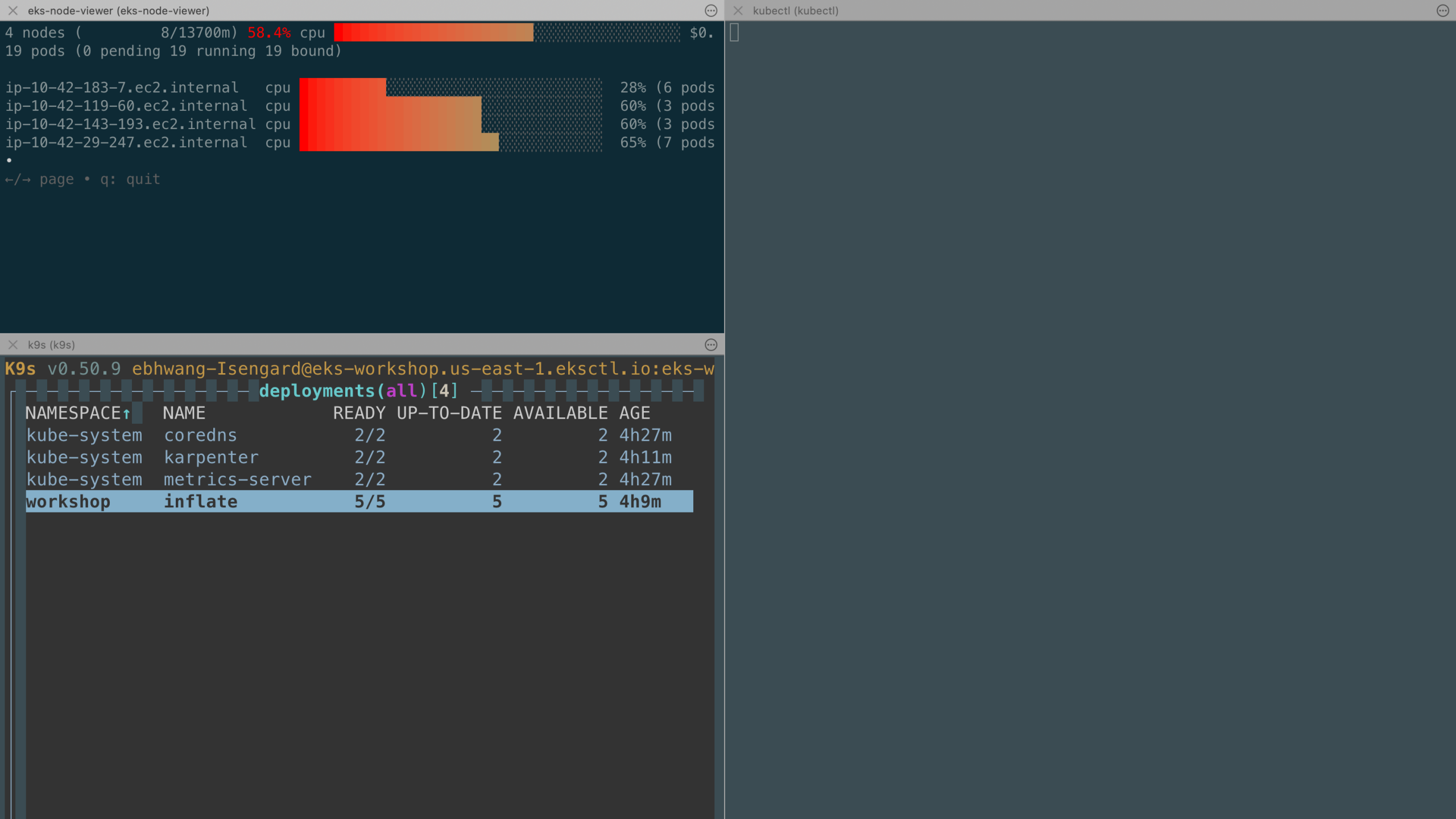

kubectl (kubectl)
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:20.096Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:20.096Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:20.096Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:30.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:30.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:30.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:30.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)

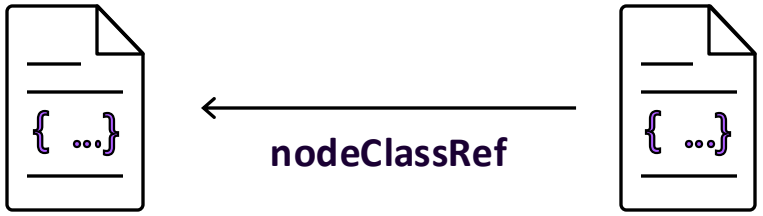

eks-node-viewer (eks-node-viewer)
5 nodes (3150m/15630m) 20.2% cpu $0.
16 pods (0 pending 16 running 16 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-56-123.ec2.internal cpu 2% (2 pods
ip-10-42-26-177.ec2.internal cpu 8% (2 pods
←/→ page • q: quit
k9s (k9s)
p d (o k nodeclaims(all)[2]
NAME↑ TYPE CAPACITY ZONE NODE
default-2wvdd c6a.2xlarge on-demand us-east-1b ip-10-42-56-
default-9ck2s c6a.large on-demand us-east-1a ip-10-42-26-
kubectl (kubectl)
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
INFO | 05:46:08.087Z | disrupting node(s)
INFO | 05:46:08.185Z | tainted node


eks-node-viewer (eks-node-viewer)
4 nodes (3/7720m) 38.9% cpu $0.3
14 pods (0 pending 14 running 14 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-26-177.ec2.internal cpu 8% (2 pods
<→ page ^ q: quit
k9s (k9s)
p d (o k hnodeclaims(all)[1]
NAME↑ TYPE CAPACITY ZONE NODE
default-9ck2s c6a.large on-demand us-east-1a ip-10-42-26-17
kubectl (kubectl)
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
ERROR | 05:44:50.097Z | could not schedule pod
ERROR | 05:45:00.098Z | could not schedule pod
ERROR | 05:45:10.098Z | could not schedule pod
INFO | 05:46:08.087Z | disrupting node(s)
INFO | 05:46:08.185Z | tainted node
INFO | 05:46:33.164Z | disrupting node(s)
INFO | 05:46:33.352Z | tainted node
INFO | 05:46:45.822Z | deleted node
INFO | 05:46:46.083Z | deleted nodeclaim


eks-node-viewer (eks-node-viewer)
3 nodes (2850m/5790m) 49.2% cpu $0.2
12 pods (0 pending 12 running 12 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
←/→ page • q: quit
k9s (k9s)
p-d-(-o-k-hnodeclaims(all)[0]
NAME↑ TYPE CAPACITY ZONE NODE READY IMAGEID ID NODEPOOL NODECL
kubectl (kubectl)
INFO | 05:05:38.686Z | found provisionable pod(s)
INFO | 05:05:38.686Z | computed new nodeclaim(s) to fit pod(s)
INFO | 05:05:38.699Z | created nodeclaim
INFO | 05:05:41.061Z | launched nodeclaim
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:40.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:44:50.097Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:00.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
ERROR | 05:45:10.098Z | could not schedule pod
all available instance types exceed limits for nodepool (NodePool=default)
INFO | 05:46:08.087Z | disrupting node(s)
INFO | 05:46:08.185Z | tainted node
INFO | 05:46:33.164Z | disrupting node(s)
INFO | 05:46:33.352Z | tainted node
INFO | 05:46:45.822Z | deleted node
INFO | 05:46:46.083Z | deleted nodeclaim
INFO | 05:47:06.333Z | deleted node
INFO | 05:47:06.596Z | deleted nodeclaim


eks-node-viewer (eks-node-viewer)
3 nodes (2850m/5790m) 49.2% cpu $0.2
12 pods (0 pending 12 running 12 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
←/→ page • q: quit
kubectl (kubectl)
k9s (k9s)
Context: ebhwang-Isengard@eks-workshop.u
Cluster: eks-workshop.us-east-1.eksctl.i
User: ebhwang-Isengard@eks-workshop.u
K9s Rev: v0.50.9
K8s Rev: v1.33.4-eks-e386d34
CPU: 0%
MEM: 6%
NAMESPACE↑ NAME
kube-system cored
kube-system karpe
kube-system metri
workshop infla
E AVAILABLE AGE
2 2 4h23m
2 2 4h7m
2 2 4h23m
0 0 4h4m
Scale deployment
workshop/inflate?
Replicas: 5
OK Cancel


eks-node-viewer (eks-node-viewer)
4 nodes (8/13700m) 58.4% cpu $0.
19 pods (0 pending 19 running 19 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-29-247.ec2.internal cpu 65% (7 pods
←/→ page • q: quit
kubectl (kubectl)
k9s (k9s)
K9s v0.50.9 ebhwang-Isengard@eks-workshop.us-east-1.eksctl.io:eks-w
deployments(all)[4]
NAMESPACE↑ NAME READY UP-TO-DATE AVAILABLE AGE
kube-system coredns 2/2 2 2 4h27m
kube-system karpenter 2/2 2 2 4h11m
kube-system metrics-server 2/2 2 2 4h27m
workshop inflate 5/5 5 5 4h9m

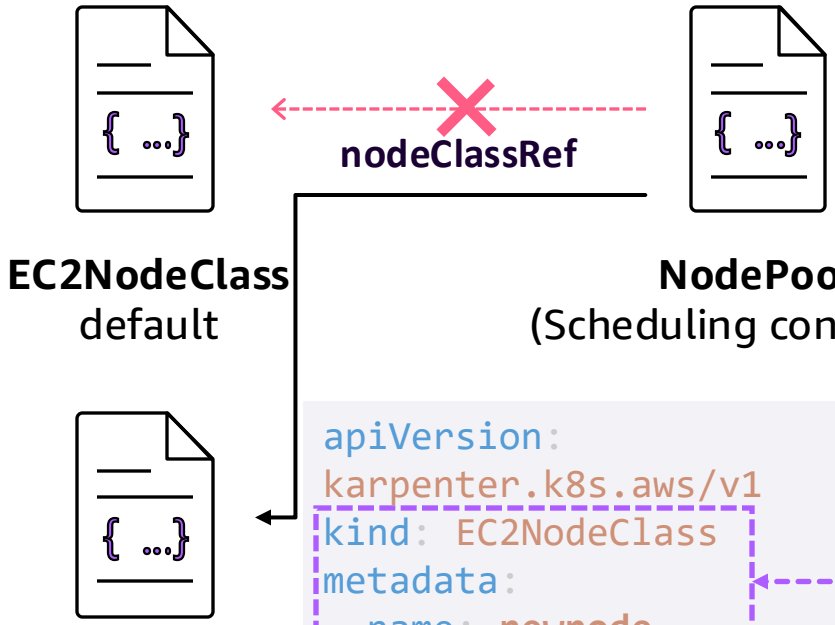


EC2NodeClass
(Node configurations)

NodePool
(Scheduling conditions)

```yaml
apiVersion: karpenter.k8s.aws/v1
kind: EC2NodeClass
metadata:
  name: default
spec:
  amiSelectorTerms:
    - alias: al2023@latest
  role: karpenter-node-role
  securityGroupSelectorTerms:
    - tags: ...
  subnetSelectorTerms:
    - tags: ...
  tags:
    ...
```

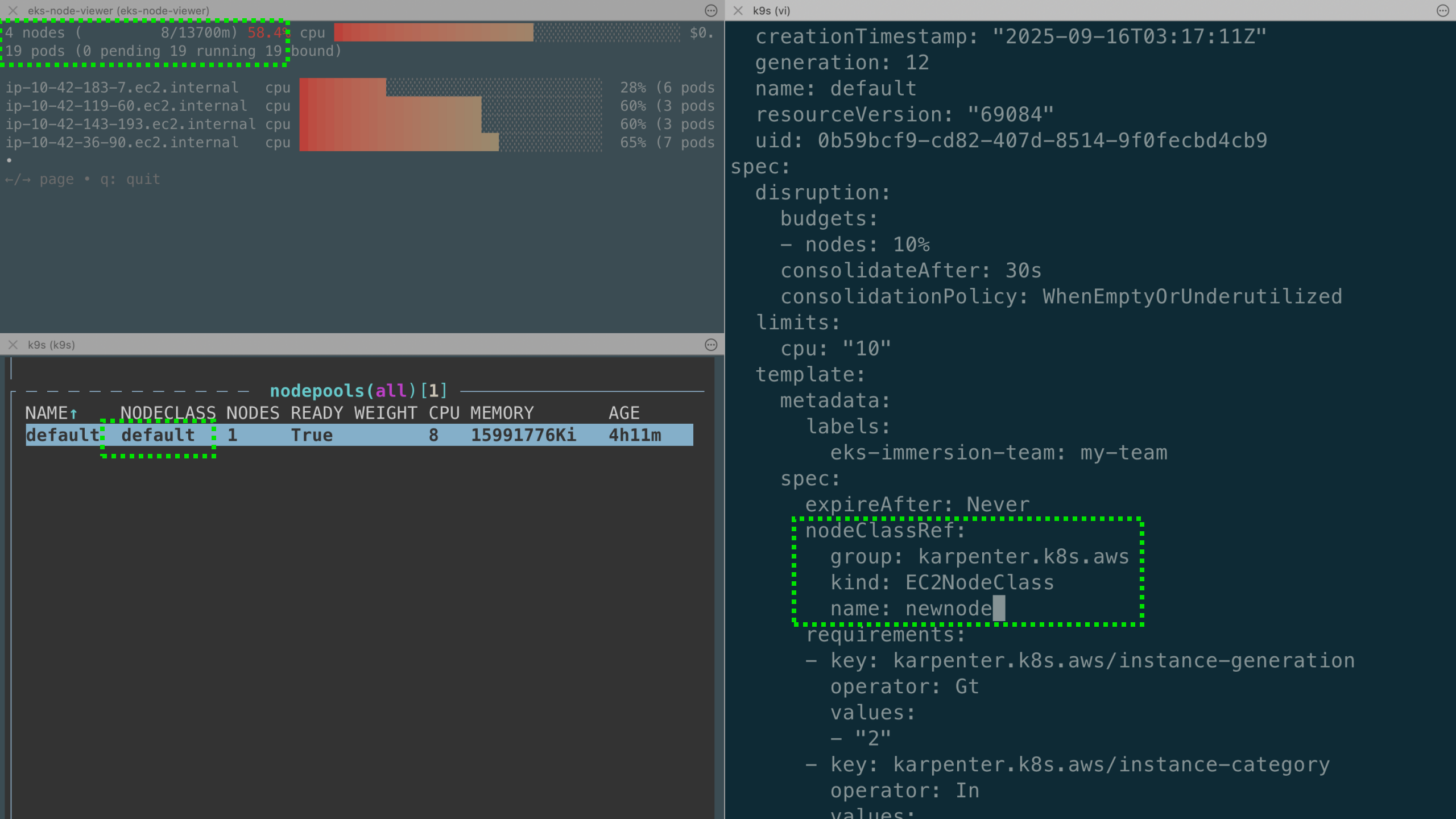
```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: default
      expireAfter: Never
      requirements:
        - key: karpenter.k8s.aws/instance-category
          ...
        - key: kubernetes.io/arch
          ...
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30s
  limits:
    cpu: "10"
```




```yaml
apiVersion: karpenter.k8s.aws/v1
kind: EC2NodeClass
metadata:
  name: newnode
spec:
  amiFamily: AL2023
  amiSelectorTerms:
    - id: ami-02510bd1e8706d4f3
  role: karpenter-node-role
```

```
amiFamily: AL2023
amiSelectorTerms:
- id: ami-02510bd1e8706d4f3
   ...
```

```yaml
apiVersion: karpenter.sh/v1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      nodeClassRef:
        group: karpenter.k8s.aws
        kind: EC2NodeClass
        name: newnode
      expireAfter: Never
      requirements:
        - key: karpenter.k8s.aws/instance-category
          ...
        - key: kubernetes.io/arch
          ...
  disruption:
    consolidationPolicy: WhenEmptyOrUnderutilized
    consolidateAfter: 30s
  limits:
    cpu: "10"
```


eks-node-viewer (eks-node-viewer)
4 nodes (8/13700m) 58.4% cpu $0.
19 pods (0 pending 19 running 19 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-36-90.ec2.internal cpu 65% (7 pods
←/→ page • q: quit
k9s (k9s)
nodepools(all)[1]
NAME↑ NODECLASS NODES READY WEIGHT CPU MEMORY AGE
default default 1 True 8 15991776Ki 4h11m
k9s (vi)
creationTimestamp: "2025-09-16T03:17:11Z"
generation: 12
name: default
resourceVersion: "69084"
uid: 0b59bcf9-cd82-407d-8514-9f0fecbd4cb9
spec:
disruption:
budgets:
- nodes: 10%
consolidateAfter: 30s
consolidationPolicy: WhenEmptyOrUnderutilized
limits:
cpu: "10"
template:
metadata:
labels:
eks-immersion-team: my-team
spec:
expireAfter: Never
nodeClassRef:
group: karpenter.k8s.aws
kind: EC2NodeClass
name: newnode
requirements:
- key: karpenter.k8s.aws/instance-generation
operator: Gt
values:
- "2"
- key: karpenter.k8s.aws/instance-category
operator: In

eks-node-viewer (eks-node-viewer)

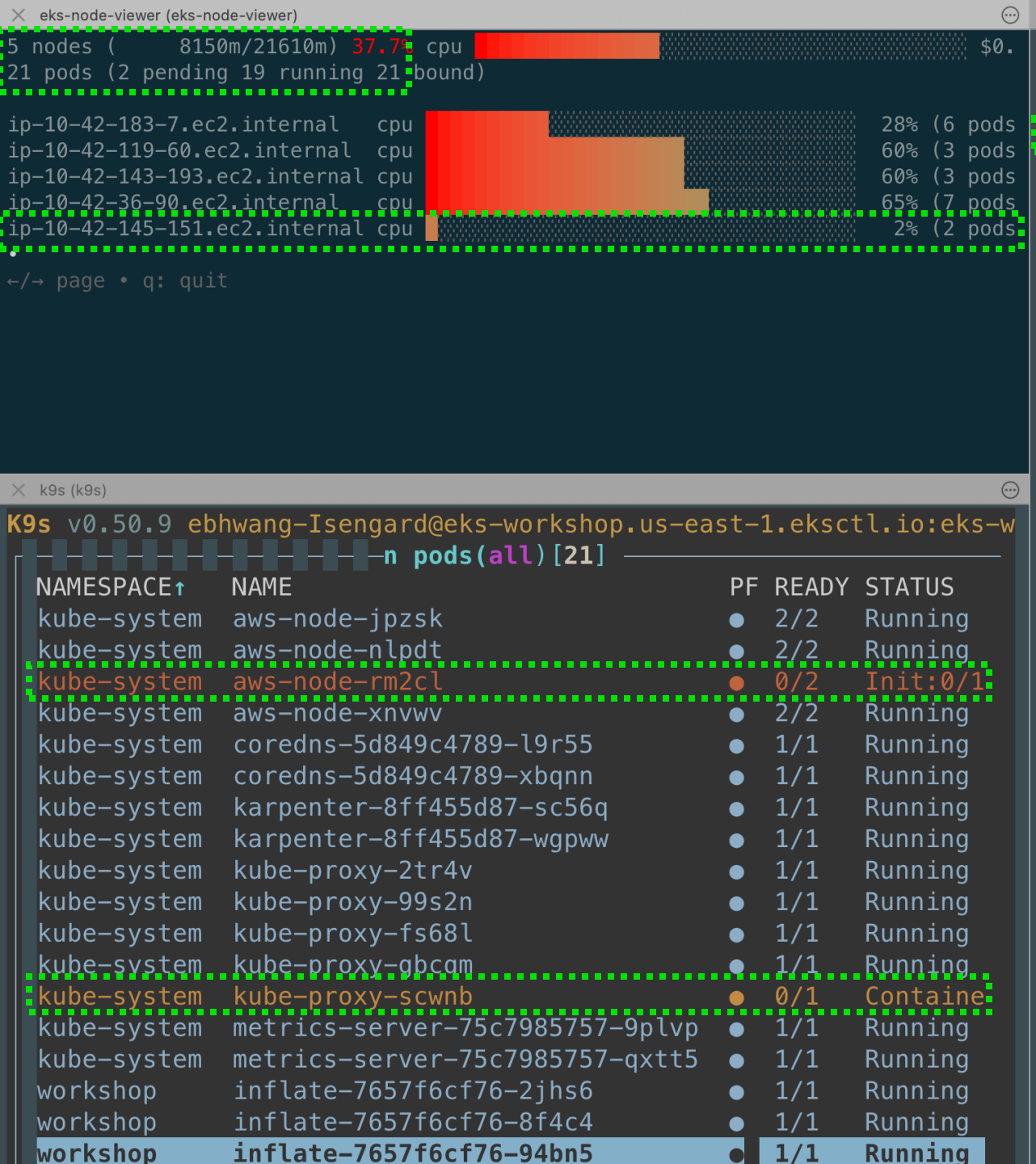
```
5 nodes (         8/21610m) 37.0% cpu ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓ $0.
19 pods (0 pending 19 running 19 bound)

ip-10-42-183-7.ec2.internal   cpu ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓  28% (6 pods
ip-10-42-119-60.ec2.internal  cpu ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓  60% (3 pods
ip-10-42-143-193.ec2.internal cpu ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓  60% (3 pods
ip-10-42-36-90.ec2.internal   cpu ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓  65% (7 pods
i-0032747bfae431bc3           cpu ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓   0% (0 pods

←/→ page • q: quit
```

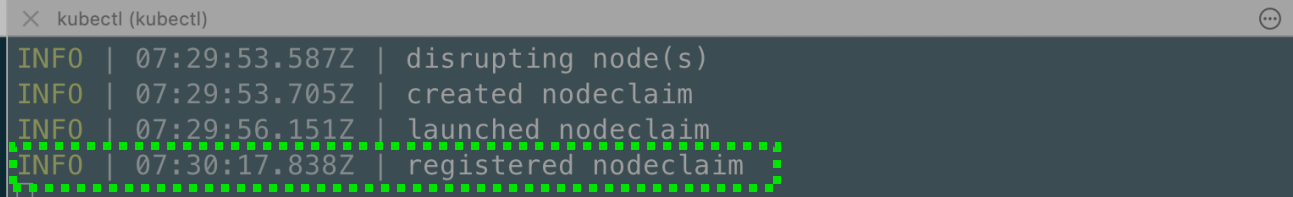
kubectl (kubectl)

```
INFO | 07:29:53.587Z | disrupting node(s)
INFO | 07:29:53.705Z | created nodeclaim
INFO | 07:29:56.151Z | launched nodeclaim
```

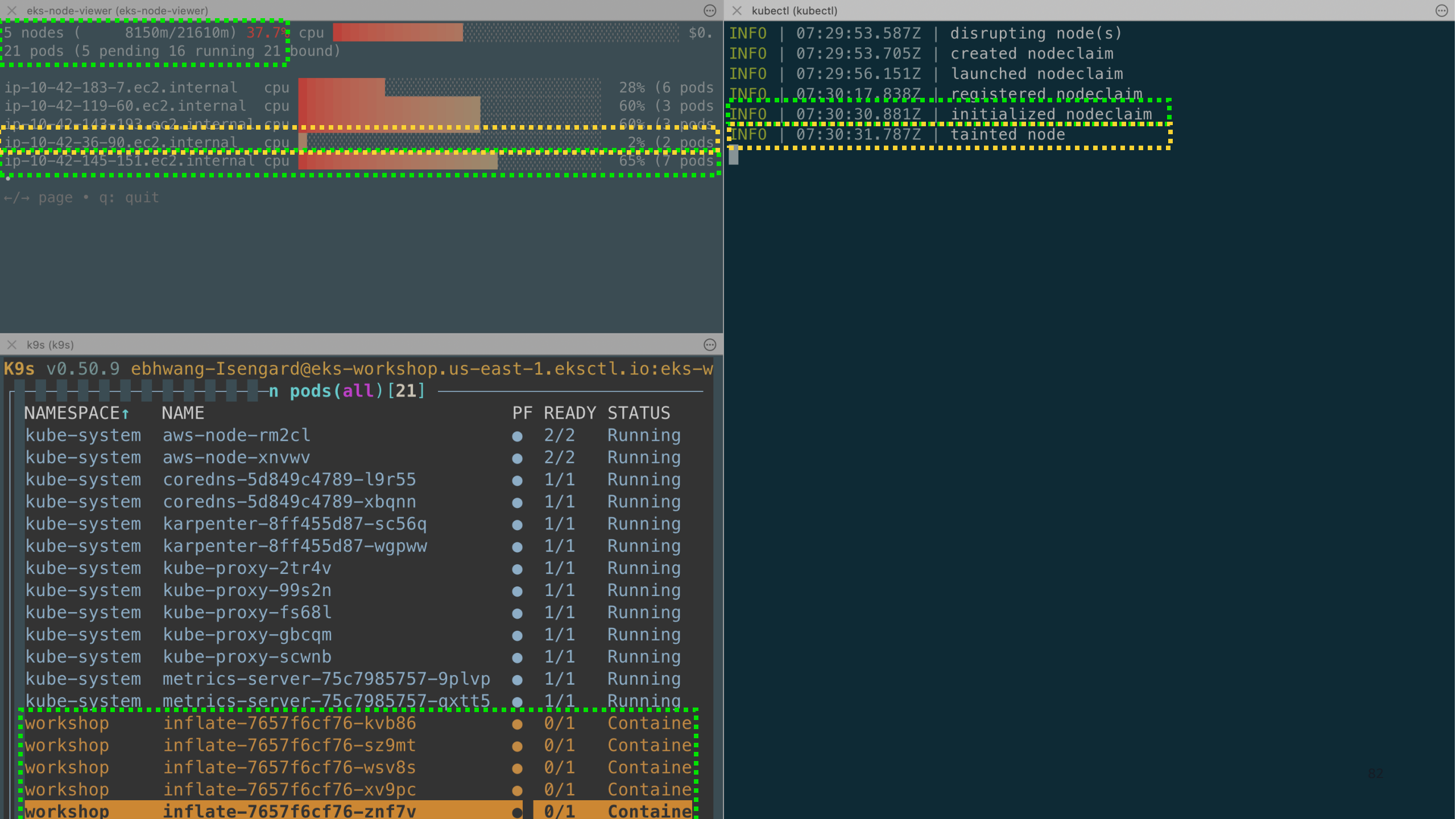
k9s (k9s)

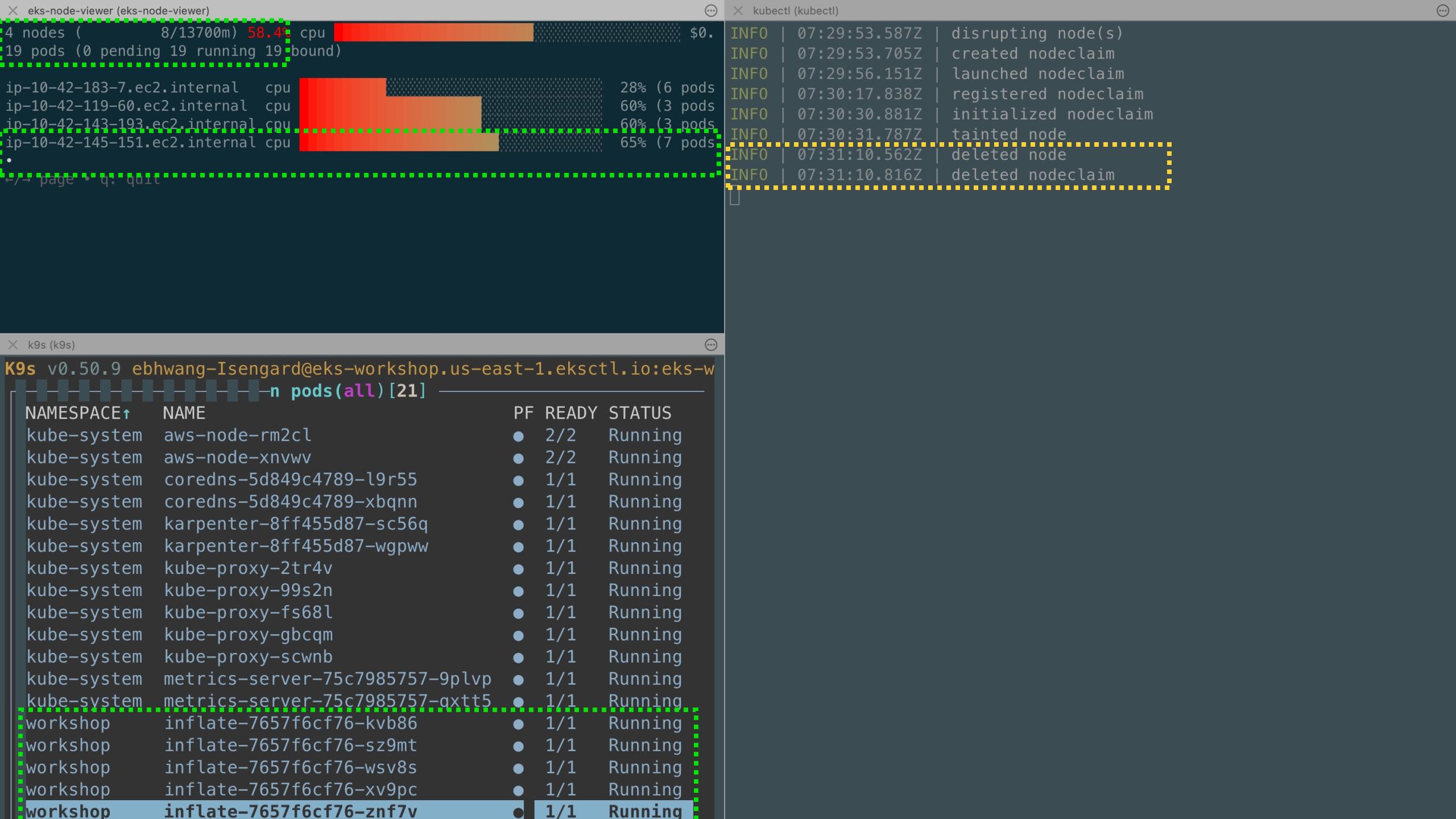
```
K9s v0.50.9 ebhwang-Isengard@eks-workshop.us-east-1.eksctl.io:eks-w
┌──────────────────n pods(all)[19]──────────────────┐
  NAMESPACE↑   NAME                                PF  READY  STATUS
  kube-system  aws-node-jpzsk                      ●   2/2    Running
  kube-system  aws-node-nlpdt                      ●   2/2    Running
  kube-system  aws-node-xnvwv                      ●   2/2    Running
  kube-system  coredns-5d849c4789-l9r55            ●   1/1    Running
  kube-system  coredns-5d849c4789-xbqnn            ●   1/1    Running
  kube-system  karpenter-8ff455d87-sc56q           ●   1/1    Running
  kube-system  karpenter-8ff455d87-wgpww           ●   1/1    Running
  kube-system  kube-proxy-2tr4v                    ●   1/1    Running
  kube-system  kube-proxy-99s2n                    ●   1/1    Running
  kube-system  kube-proxy-fs68l                    ●   1/1    Running
  kube-system  kube-proxy-gbcqm                    ●   1/1    Running
  kube-system  metrics-server-75c7985757-9plvp     ●   1/1    Running
  kube-system  metrics-server-75c7985757-qxtt5     ●   1/1    Running
  workshop     inflate-7657f6cf76-2jhs6            ●   1/1    Running
  workshop     inflate-7657f6cf76-8f4c4            ●   1/1    Running
  workshop     inflate-7657f6cf76-94bn5            ●   1/1    Running
  workshop     inflate-7657f6cf76-jjxtf            ●   1/1    Running
  workshop     inflate-7657f6cf76-wqnrk            ●   1/1    Running
```


eks-node-viewer (eks-node-viewer)
5 nodes (8150m/21610m) 37.7% cpu $0.
21 pods (2 pending 19 running 21 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-36-90.ec2.internal cpu 65% (7 pods
ip-10-42-145-151.ec2.internal cpu 2% (2 pods
←/→ page • q: quit


kubectl (kubectl)
INFO | 07:29:53.587Z | disrupting node(s)
INFO | 07:29:53.705Z | created nodeclaim
INFO | 07:29:56.151Z | launched nodeclaim
INFO | 07:30:17.838Z | registered nodeclaim

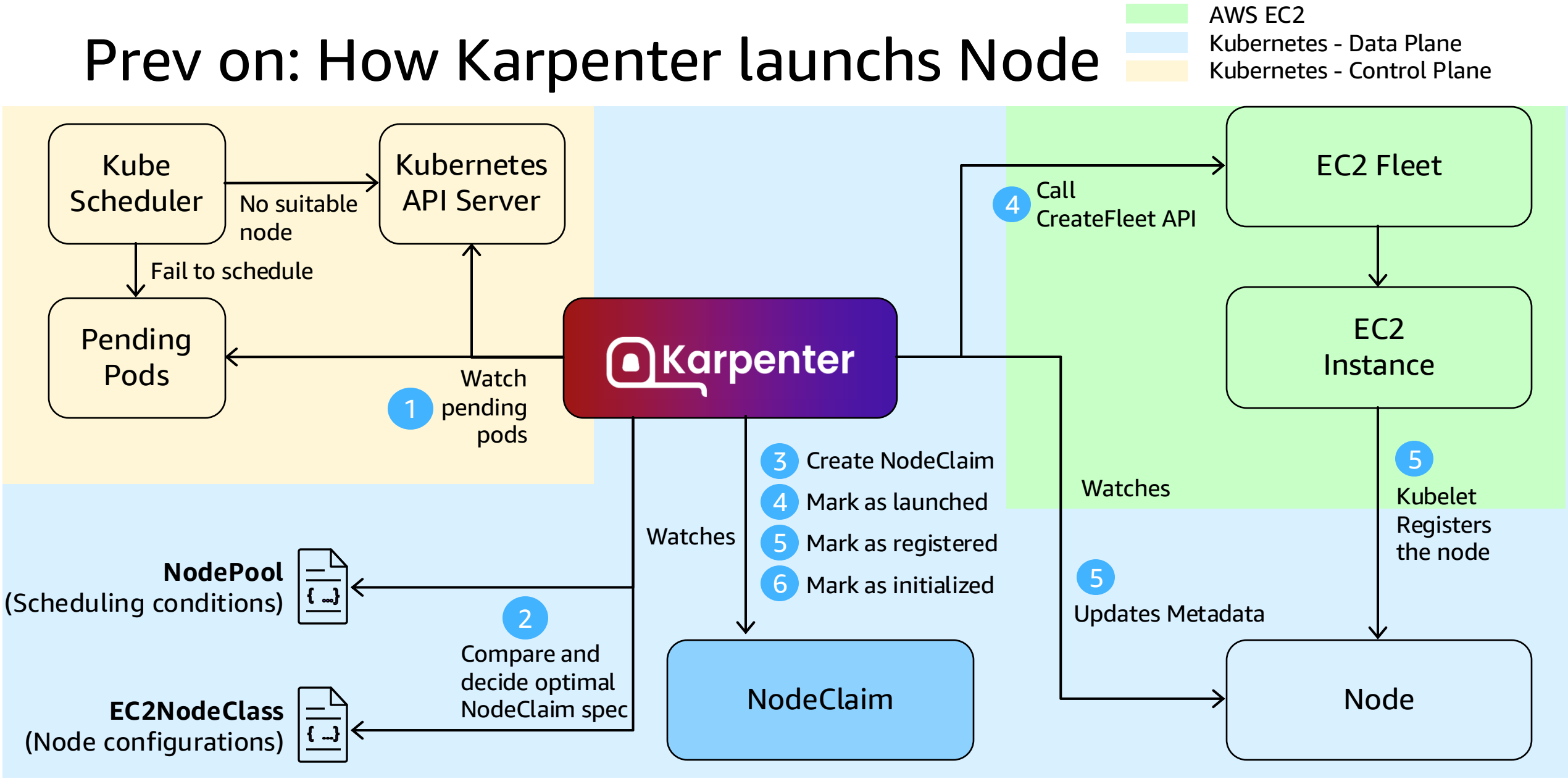
k9s (k9s)
K9s v0.50.9 ebhwang-Isengard@eks-workshop.us-east-1.eksctl.io:eks-w
n pods(all)[21]
NAMESPACE↑ NAME PF READY STATUS
kube-system aws-node-jpzsk 2/2 Running
kube-system aws-node-nlpdt 2/2 Running
kube-system aws-node-rm2cl 0/2 Init:0/1
kube-system aws-node-xnvwv 2/2 Running
kube-system coredns-5d849c4789-l9r55 1/1 Running
kube-system coredns-5d849c4789-xbqnn 1/1 Running
kube-system karpenter-8ff455d87-sc56q 1/1 Running
kube-system karpenter-8ff455d87-wgpww 1/1 Running
kube-system kube-proxy-2tr4v 1/1 Running
kube-system kube-proxy-99s2n 1/1 Running
kube-system kube-proxy-fs68l 1/1 Running
kube-system kube-proxy-gbcqm 1/1 Running
kube-system kube-proxy-scwnb 0/1 Containe
kube-system metrics-server-75c7985757-9plvp 1/1 Running
kube-system metrics-server-75c7985757-qxtt5 1/1 Running
workshop inflate-7657f6cf76-2jhs6 1/1 Running
workshop inflate-7657f6cf76-8f4c4 1/1 Running
workshop inflate-7657f6cf76-94bn5 1/1 Running


eks-node-viewer (eks-node-viewer)
5 nodes (8150m/21610m) 37.7% cpu $0.
21 pods (5 pending 16 running 21 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-36-90.ec2.internal cpu 2% (2 pods
ip-10-42-145-151.ec2.internal cpu 65% (7 pods
←/→ page • q: quit
kubectl (kubectl)
INFO | 07:29:53.587Z | disrupting node(s)
INFO | 07:29:53.705Z | created nodeclaim
INFO | 07:29:56.151Z | launched nodeclaim
INFO | 07:30:17.838Z | registered nodeclaim
INFO | 07:30:30.881Z | initialized nodeclaim
INFO | 07:30:31.787Z | tainted node
k9s (k9s)
K9s v0.50.9 ebhwang-Isengard@eks-workshop.us-east-1.eksctl.io:eks-w
n pods(all)[21]
NAMESPACE↑ NAME PF READY STATUS
kube-system aws-node-rm2cl 2/2 Running
kube-system aws-node-xnvwv 2/2 Running
kube-system coredns-5d849c4789-l9r55 1/1 Running
kube-system coredns-5d849c4789-xbqnn 1/1 Running
kube-system karpenter-8ff455d87-sc56q 1/1 Running
kube-system karpenter-8ff455d87-wgpww 1/1 Running
kube-system kube-proxy-2tr4v 1/1 Running
kube-system kube-proxy-99s2n 1/1 Running
kube-system kube-proxy-fs68l 1/1 Running
kube-system kube-proxy-gbcqm 1/1 Running
kube-system kube-proxy-scwnb 1/1 Running
kube-system metrics-server-75c7985757-9plvp 1/1 Running
kube-system metrics-server-75c7985757-qxtt5 1/1 Running
workshop inflate-7657f6cf76-kvb86 0/1 Containe
workshop inflate-7657f6cf76-sz9mt 0/1 Containe
workshop inflate-7657f6cf76-wsv8s 0/1 Containe
workshop inflate-7657f6cf76-xv9pc 0/1 Containe
workshop inflate-7657f6cf76-znf7v 0/1 Containe

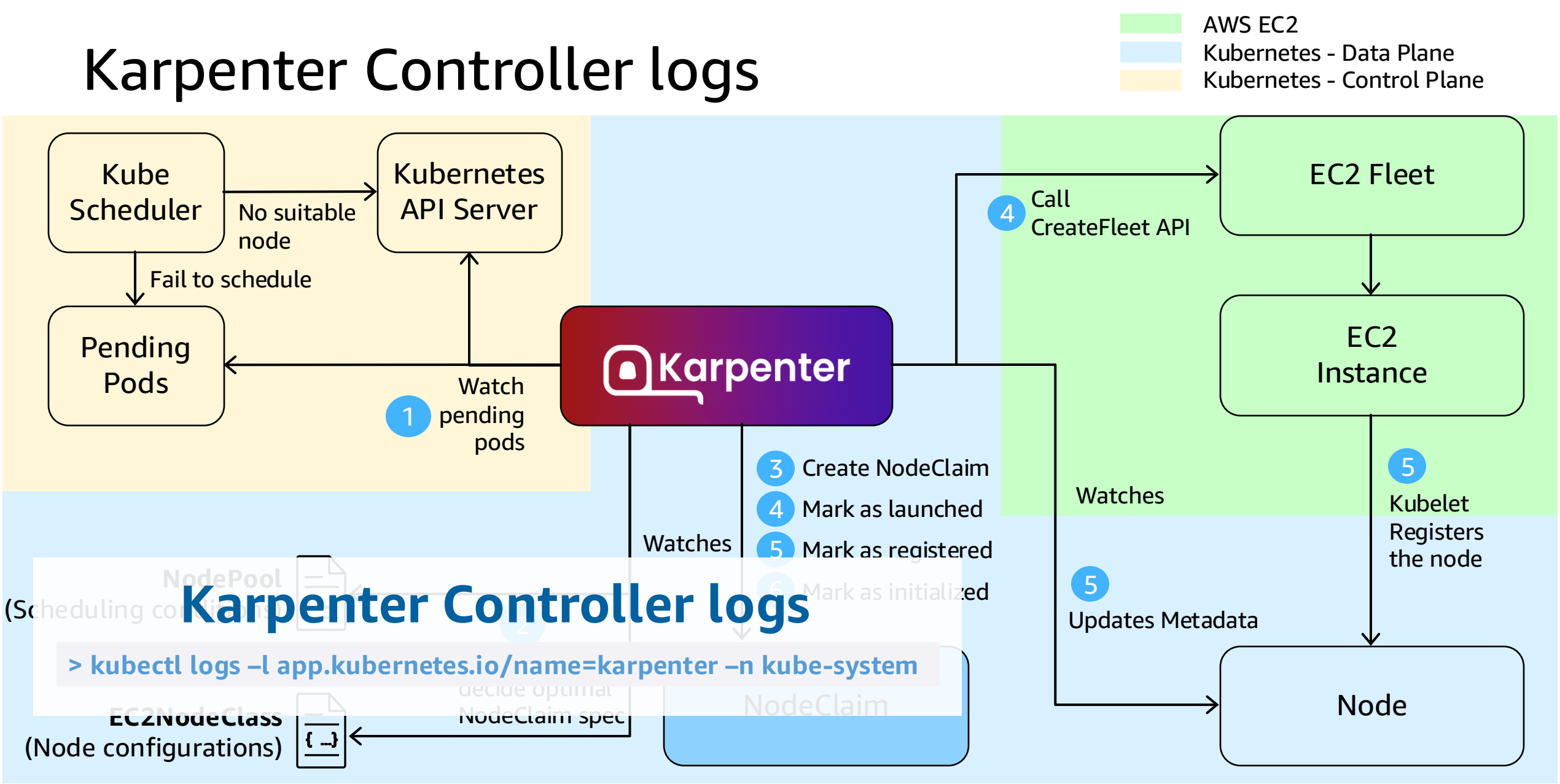

eks-node-viewer (eks-node-viewer)
4 nodes (8/13700m) 58.4% cpu $0.
19 pods (0 pending 19 running 19 bound)
ip-10-42-183-7.ec2.internal cpu 28% (6 pods
ip-10-42-119-60.ec2.internal cpu 60% (3 pods
ip-10-42-143-193.ec2.internal cpu 60% (3 pods
ip-10-42-145-151.ec2.internal cpu 65% (7 pods
kubectl (kubectl)
INFO | 07:29:53.587Z | disrupting node(s)
INFO | 07:29:53.705Z | created nodeclaim
INFO | 07:29:56.151Z | launched nodeclaim
INFO | 07:30:17.838Z | registered nodeclaim
INFO | 07:30:30.881Z | initialized nodeclaim
INFO | 07:30:31.787Z | tainted node
INFO | 07:31:10.562Z | deleted node
INFO | 07:31:10.816Z | deleted nodeclaim
k9s (k9s)
K9s v0.50.9 ebhwang-Isengard@eks-workshop.us-east-1.eksctl.io:eks-w
pods(all)[21]
NAMESPACE NAME PF READY STATUS
kube-system aws-node-rm2cl 2/2 Running
kube-system aws-node-xnvwv 2/2 Running
kube-system coredns-5d849c4789-l9r55 1/1 Running
kube-system coredns-5d849c4789-xbqnn 1/1 Running
kube-system karpenter-8ff455d87-sc56q 1/1 Running
kube-system karpenter-8ff455d87-wgpww 1/1 Running
kube-system kube-proxy-2tr4v 1/1 Running
kube-system kube-proxy-99s2n 1/1 Running
kube-system kube-proxy-fs68l 1/1 Running
kube-system kube-proxy-gbcqm 1/1 Running
kube-system kube-proxy-scwnb 1/1 Running
kube-system metrics-server-75c7985757-9plvp 1/1 Running
kube-system metrics-server-75c7985757-qxtt5 1/1 Running
workshop inflate-7657f6cf76-kvb86 1/1 Running
workshop inflate-7657f6cf76-sz9mt 1/1 Running
workshop inflate-7657f6cf76-wsv8s 1/1 Running
workshop inflate-7657f6cf76-xv9pc 1/1 Running
workshop inflate-7657f6cf76-znf7v 1/1 Running

Troubleshooting

Prev on: How Karpenter launchs Node



Karpenter Controller logs



Karpenter controller logs – Launching new node

```
$ kubectl logs -l app.kubernetes.io/name=karpenter -n $KARPENTER_NAMESPACE

{"level": "INFO", "time": "2025-08-28T05:35:21.771Z", "logger": "controller", "message": "found provisionable pod(s)", "commit": "1c39126", "controller": "provisioner", "namespace": "", "name": "", "reconcileID": "abc123-def456", "Pods": "abcd1234-xyz and 1 other(s)", "duration": "39.358238ms"}
{"level": "INFO", "time": "2025-08-28T05:35:21.771Z", "logger": "controller", "message": "computed new nodeclaim(s) to fit pod(s)", "commit": "1c39126", "controller": "provisioner", "namespace": "", "name": "", "reconcileID": "abc123-def456", "nodeclaims":1, "pods":6}
{"level": "INFO", "time": "2025-08-28T05:35:21.787Z", "logger": "controller", "message": "created nodeclaim", "commit": "1c39126", "controller": "provisioner", "namespace": "", "name": "", "reconcileID": "abc123-def456", "NodePool":{"name": "default"}, "NodeClaim":{"name": "default-5dt5n"}, "requests":{"cpu": "150m", "pods": "9"}, "instance-types": "t3.2xlarge, t3.large, t3.medium, t3.small, t3.xlarge"}
{"level": "INFO", "time": "2025-08-28T05:35:24.530Z", "logger": "controller", "message": "launched nodeclaim", "commit": "1c39126", "controller": "nodeclaim.lifecycle", "controllerGroup": "karpenter.sh", "controllerKind": "NodeClaim", "NodeClaim":{"name": "default-5dt5n"}, "namespace": "", "name": "default-5dt5n", "reconcileID": "abc123-def456", "provider-id": "aws:///xxx.yyy.zzz", "instance-type": "t3.small", "zone": "us-east-1a", "capacity-type": "on-demand", "allocatable":{"cpu": "1930m", "ephemeral-storage": "17Gi", "memory": "1459636Ki", "pods": "11"}}
{"level": "INFO", "time": "2025-08-28T05:36:05.025Z", "logger": "controller", "message": "registered nodeclaim", "commit": "1c39126", "controller": "nodeclaim.lifecycle", "controllerGroup": "karpenter.sh", "controllerKind": "NodeClaim", "NodeClaim":{"name": "default-5dt5n"}, "namespace": "", "name": "default-5dt5n", "reconcileID": "abc123-def456", "provider-id": "aws:///xxx.yyy.zzz", "Node":{"name": "ip-12-345-678-012.ec2.internal"}}
{"level": "INFO", "time": "2025-08-28T05:36:20.021Z", "logger": "controller", "message": "initialized nodeclaim", "commit": "1c39126", "controller": "nodeclaim.lifecycle", "controllerGroup": "karpenter.sh", "controllerKind": "NodeClaim", "NodeClaim":{"name": "default-5dt5n"}, "namespace": "", "name": "default-5dt5n", "reconcileID": "abc123-def456", "provider-id": "aws:///xxx.yyy.zzz", "Node":{"name": "ip-12-345-678-012.ec2.internal"},
```

Karpenter controller log – Disruption

```
$ kubectl logs -l app.kubernetes.io/name=karpenter -n $KARPENTER_NAMESPACE

{"level": "INFO", "time": "2025-08-28T05:45:51.357Z", "logger": "controller", "message": "disrupting
node(s)", "commit": "1c39126", "controller": "disruption", "namespace": "", "name": "",
"reconcileID": "abc123-def456", "command-id": "abcdef123456", "reason": "empty", "decision":
"delete", "disrupted-node-count":1, "replacement-node-count":0, "pod-count":0, "disrupted-
nodes":[{"Node":{"name": "ip-12-345-678-012.ec2.internal"}, "NodeClaim":{"name": "default-5dt5n"},
"capacity-type": "on-demand", "instance-type": "t3.small"}], "replacement-nodes":[]}
{"level": "INFO", "time": "2025-08-28T05:45:51.680Z", "logger": "controller", "message": "tainted
node", "commit": "1c39126", "controller": "node.termination", "controllerGroup": "",
"controllerKind": "Node", "Node":{"name": "ip-12-345-678-012.ec2.internal"}, "namespace": "",
"name": "ip-12-345-678-012.ec2.internal", "reconcileID": "abc123-def456", "NodeClaim":{"name":
"default-5dt5n"}, "taint.Key": "karpenter.sh/disrupted", "taint.Value": "", "taint.Effect":
"NoSchedule"}
{"level": "INFO", "time": "2025-08-28T05:46:43.843Z", "logger": "controller", "message": "deleted
node", "commit": "1c39126", "controller": "node.termination", "controllerGroup": "",
"controllerKind": "Node", "Node":{"name": "ip-12-345-678-012.ec2.internal"}, "namespace": "",
"name": "ip-12-345-678-012.ec2.internal", "reconcileID": "abc123-def456", "NodeClaim":{"name":
"default-5dt5n"}}
{"level": "INFO", "time": "2025-08-28T05:46:44.086Z", "logger": "controller", "message": "deleted
nodeclaim", "commit": "1c39126", "controller": "nodeclaim.lifecycle", "controllerGroup":
"karpenter.sh", "controllerKind": "NodeClaim", "NodeClaim":{"name": "default-5dt5n"}, "namespace":
"", "name": "default-5dt5n", "reconcileID": "abc123-def456", "provider-id": "aws:///xxx.yyy.zzz",
"Node":{"name": "ip-12-345-678-012.ec2.internal"}}
```

사례 1) Karpenter controller logs

- 카펜터가 새 노드를 생성하지 않은 원인


```
{"level":"INFO","time":"2024-10-23T05:48:53.176Z","logger":"controller","message":"computed new nodeclaim(s) to fit pod(s)","commit":"abc123","controller":"provisioner","namespace":"","name":"","reconcileID":"xxx-yyy-zzzz","nodeclaims":1,"pods":1}
{"level":"INFO","time":"2024-10-23T05:48:53.176Z","logger":"controller","message":"computed 4 unready node(s) will fit 16 pod(s)","commit":"abc123","controller":"provisioner","namespace":"","name":"","reconcileID":"xxx-yyy-zzzz"}
```


- 해당시간 카펜터 노드가 삭제/드리프트 된 원인

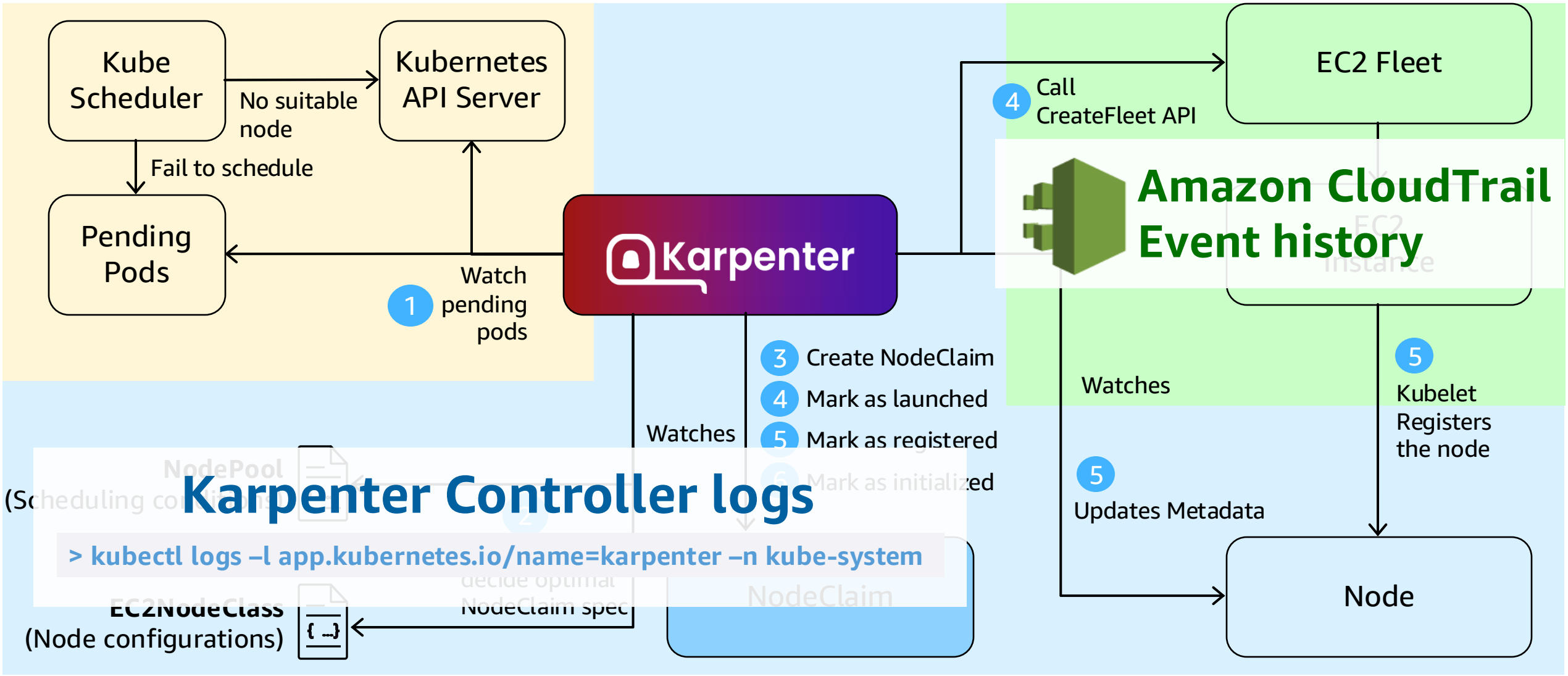
사례 1) Karpenter controller logs


```
{"level":"INFO","time":"2024-10-21T01:04:09.115Z","logger":"controller.provisioner","message":"created nodeclaim",
...}
{"level":"DEBUG","time":"2024-10-21T01:04:09.801Z","logger":"controller.nodeclaim.lifecycle","message":"created
launch , ...}
{"level":"INFO","time":"2024-10-21T01:04:11.395Z","logger":"controller.nodeclaim.lifecycle","message":"launched
nodeclaim", ...}
{"level":"INFO","time":"2024-10-21T01:04:52.814Z","logger":"controller.nodeclaim.lifecycle","message":"registered
nodeclaim", ...}
{"level":"INFO","time":"2024-10-21T01:05:14.595Z","logger":"controller.nodeclaim.lifecycle","message":"initialized
nodeclaim", ...}
{"level":"DEBUG","time":"2024-10-26T05:02:51.521Z","logger":"controller.nodeclaim.disruption","message":"marking
empty","commit":"123abc","nodeclaim":"nodepool-default-a1b2c3"}
{"level":"INFO","time":"2024-10-30T05:27:15.878Z","logger":"controller.interruption","message":"initiating delete
from interruption message","commit":"123abc","queue":"sqs-eks-
workshop","messageKind":"StateChangeKind","nodeclaim":"nodepool-default-
a1b2c3","action":"CordonAndDrain","node":"ip-10-123-456-78.ec2.compute.internal"}
{"level":"INFO","time":"2024-10-30T05:27:15.992Z","logger":"controller.node.termination","message":"tainted
node","commit":"123abc","node":"ip-10-123-456-78.ec2.compute.internal"}
stdout F {"level":"INFO","time":"2024-10-
30T05:28:47.041Z","logger":"controller.node.termination","message":"deleted node","commit":"123abc","node":"ip-10-
123-456-78.ec2.compute.internal"}
{"level":"INFO","time":"2024-10-30T05:28:47.412Z","logger":"controller.nodeclaim.termination","message":"deleted
nodeclaim","commit":"123abc","nodeclaim":"nodepool-default-a1b2c3","node":"ip-10-123-456-
78.ec2.compute.internal","provider-id":"aws:///us-east-1a/i-12345abcde"}
```

사례 1) Karpenter controller logs


```
{"level":"INFO","time":"2024-10-23T05:49:25.397Z","logger":"controller","message":"registered
nodeclaim","commit":"abc123","controller":"nodeclaim.lifecycle","controllerGroup":"karpenter.sh","controllerKind":"NodeClaim","NodeCl
aim":{"name":"cicd-abc12"},"namespace":"","namname":"cicd-abc12","reconcileID":"xxx-yyy-zzzz","provider-id":"aws:///ap-northeast-
2c/i-dummy123456789","Node":{"name":"ip-10-0-77-149.ap-northeast-2.compute.internal"}}
{"level":"INFO","time":"2024-10-23T05:49:35.413Z","logger":"controller","message":"initialized
nodeclaim","commit":"abc123","controller":"nodeclaim.lifecycle","controllerGroup":"karpenter.sh","controllerKind":"NodeClaim","NodeCl
aim":{"name":"cicd-abc12"},"namespace":"","name":"cicd-abc12","reconcileID":"xxx-yyy-zzzz","provider-id":"aws:///ap-northeast-2c/i-
dummy123456789","Node":{"name":"ip-10-0-77-149.ap-northeast-2.compute.internal"},"allocatable":{"cpu":"63770m","ephemeral-
storage":"965282555124","hugepages-1Gi":"0","hugepages-2Mi":"0","memory":"511196712Ki","pods":"737"}}
{"level":"INFO","time":"2024-10-23T06:05:10.344Z","logger":"controller","message":"discovered ssm
parameter","commit":"abc123","controller":"nodeclass.status","controllerGroup":"karpenter.k8s.aws","controllerKind":"EC2NodeClass","E
C2NodeClass":{"name":"disk-optimized-v3"},"namespace":"","name":"disk-optimized-v3","reconcileID":"xxx-yyy-
zzzz","parameter":"/aws/service/eks/optimized-ami/1.29/amazon-linux-2/recommended/image_id","value":"ami-dummy12345"}
{"level":"INFO","time":"2024-10-23T06:05:10.360Z","logger":"controller","message":"discovered ssm
parameter","commit":"abc123","controller":"nodeclass.status","controllerGroup":"karpenter.k8s.aws","controllerKind":"EC2NodeClass","E
C2NodeClass":{"name":"disk-optimized-v3"},"namespace":"","name":"disk-optimized-v3","reconcileID":"xxx-yyy-
zzzz","parameter":"/aws/service/eks/optimized-ami/1.29/amazon-linux-2-gpu/recommended/image_id","value":"ami-dummy12346"}
{"level":"INFO","time":"2024-10-23T06:05:10.392Z","logger":"controller","message":"discovered ssm
parameter","commit":"abc123","controller":"nodeclass.status","controllerGroup":"karpenter.k8s.aws","controllerKind":"EC2NodeClass","E
C2NodeClass":{"name":"disk-optimized-v3"},"namespace":"","name":"disk-optimized-v3","reconcileID":"xxx-yyy-
zzzz","parameter":"/aws/service/eks/optimized-ami/1.29/amazon-linux-2-gpu/recommended/image_id","value":"ami-dummy12347"}
{"level":"INFO","time":"2024-10-23T06:05:16.910Z","logger":"controller","message":"disrupting nodeclaim(s) via replace, terminating 1
nodes (1 pods) ip-10-0-45-181.ap-northeast-2.compute.internal/c7i.large/on-demand and replacing with on-demand node from types
c7i.large, c7i.xlarge, c7i.2xlarge, c7i.4xlarge, c7i.8xlarge and 6
other(s)","commit":"abc123","controller":"disruption","namespace":"","name":"","reconcileID":"xxx-yyy-zzzz","command-id":"xxx-yyy-
zzzz","reason":"drifted"}
{"level":"INFO","time":"2024-10-23T06:05:16.950Z","logger":"controller","message":"created
nodeclaim","commit":"abc123","controller":"disruption","namespace":"","name":"","reconcileID":"xxx-yyy-
```

Amazon CloudTrail Event history

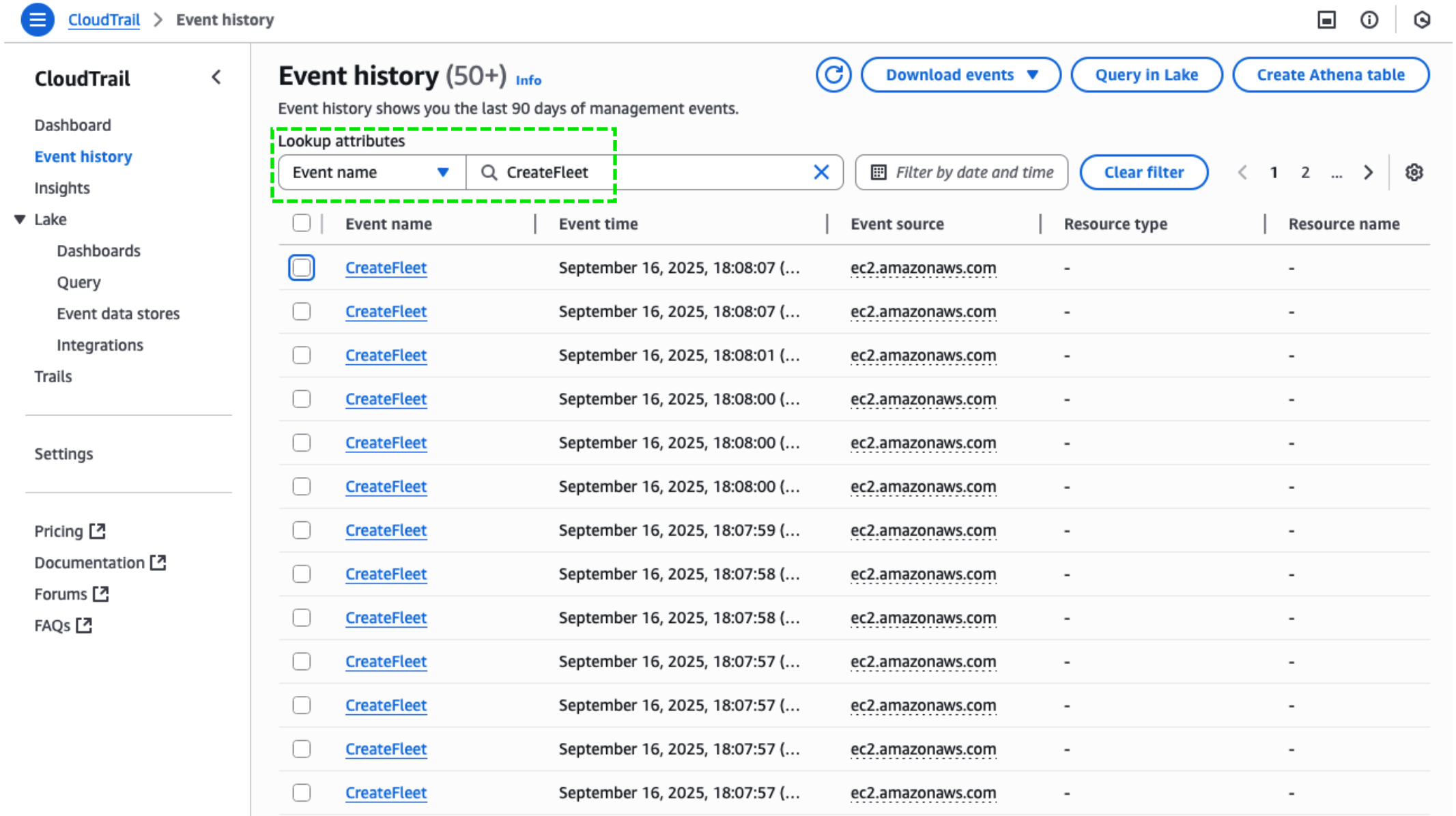


Amazon CloudTrail Event history - KarpenterControllerPolicy

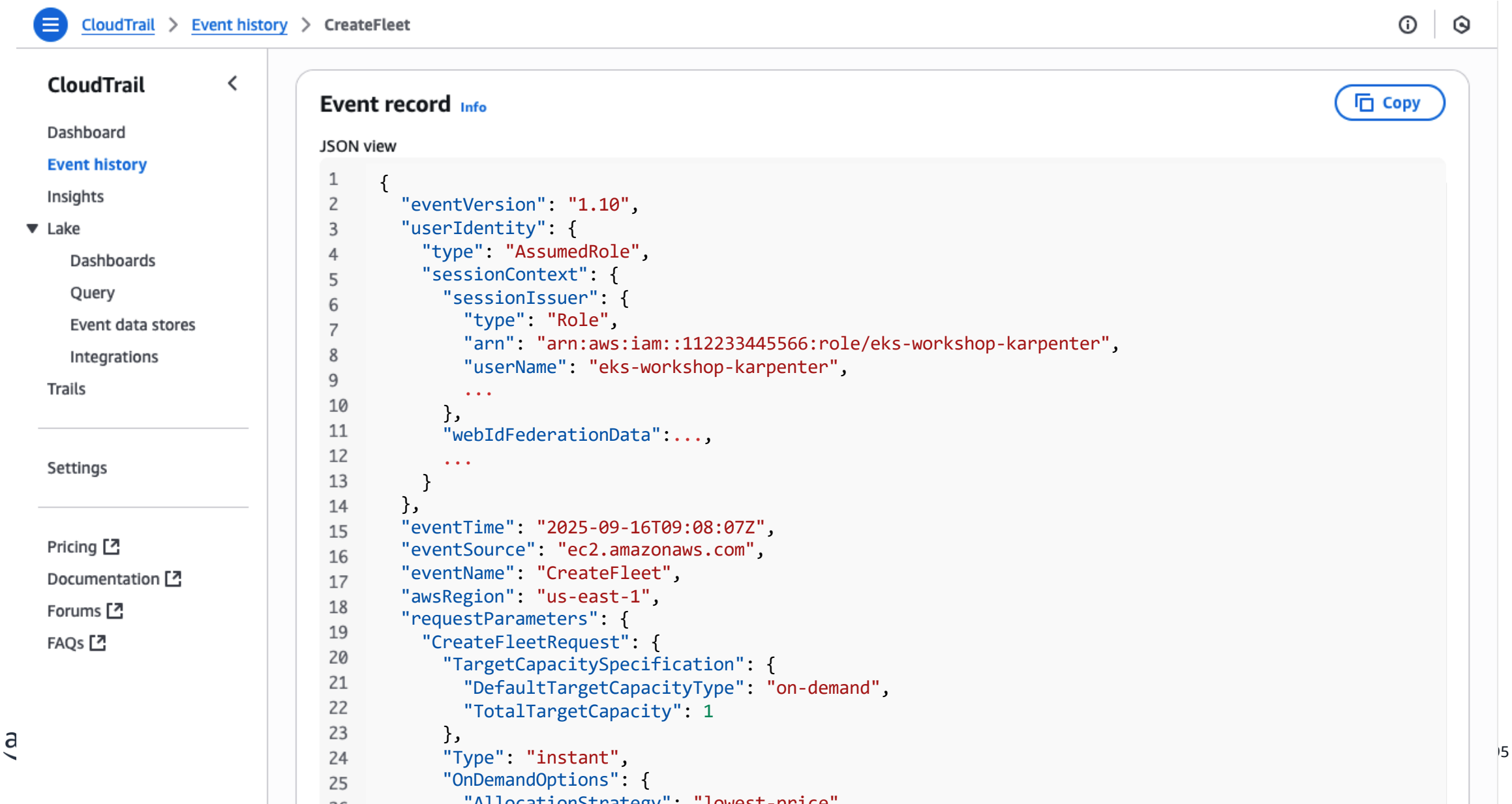
- KarpenterController 가 사용하는 AWS API
  - ec2:CreateFleet
  - ec2:RunInstances
  - ec2:TerminateInstances

```
{
  "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "AllowScopedEC2InstanceAccessActions",
        "Effect": "Allow",
        ...
        "Action": [
          "ec2:RunInstances",
          "ec2:CreateFleet"
        ]
      },
      {
        "Sid": "AllowScopedEC2InstanceActionsWithTags",
        "Effect": "Allow",
        ...
        "Action": [
          "ec2:RunInstances",
          "ec2:CreateFleet",
          "ec2:CreateLaunchTemplate"
        ],
...
}
```

Amazon CloudTrail Event history



Amazon CloudTrail Event history



사례 2) 인스턴스의 시작 실패

- Amazon EC2의 제한
  - InsufficientInstanceCapacity
  - InstanceLimitExceeded
- Capacity Type
  - InsufficientReservedInstanceCapacity
  - MaxSpotInstanceCountExceeded
  - SpotMaxPriceTooLow

RunInstances Info

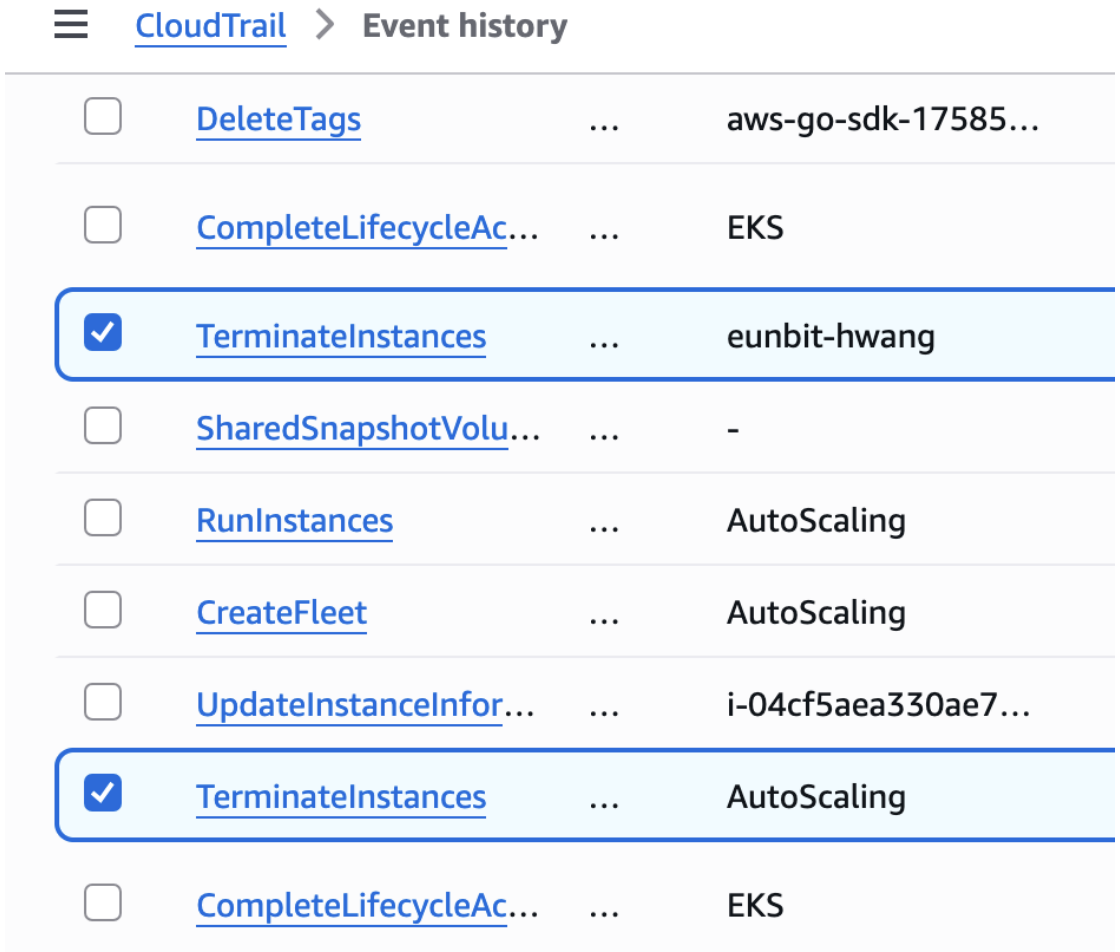
Details Info

Event time September 24, 2025, 12:01:27 (UTC+09:00)	**AWS access key** ABCDEFGHIJKLMNOP	**AWS region** us-east-1
User name 123456789012	**Source IP address** 12.345.67.890	**Error code** Client.RequestLimitExceeded

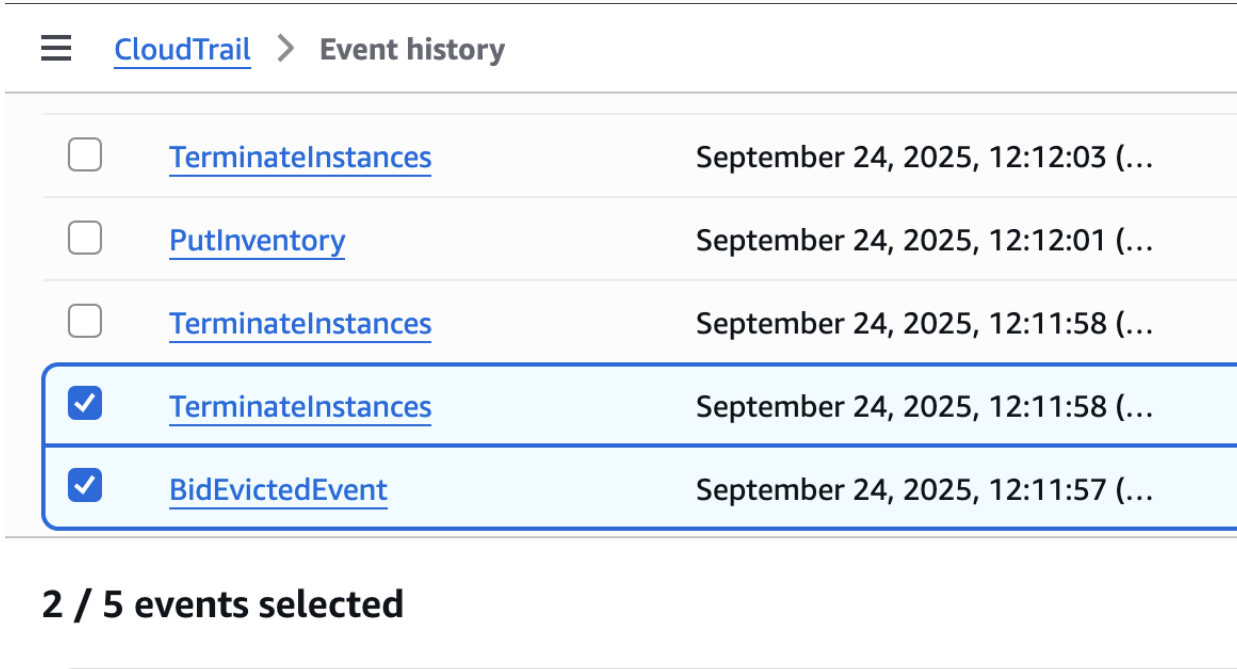
```
"errorCode": "Client.RequestLimitExceeded",
"errorMessage": "Request limit exceeded. Account 123456789012 has been throttled on
                 ec2:RunInstances because it exceeded its request rate limit.",
```

사례 2) 인스턴스 종료의 원인

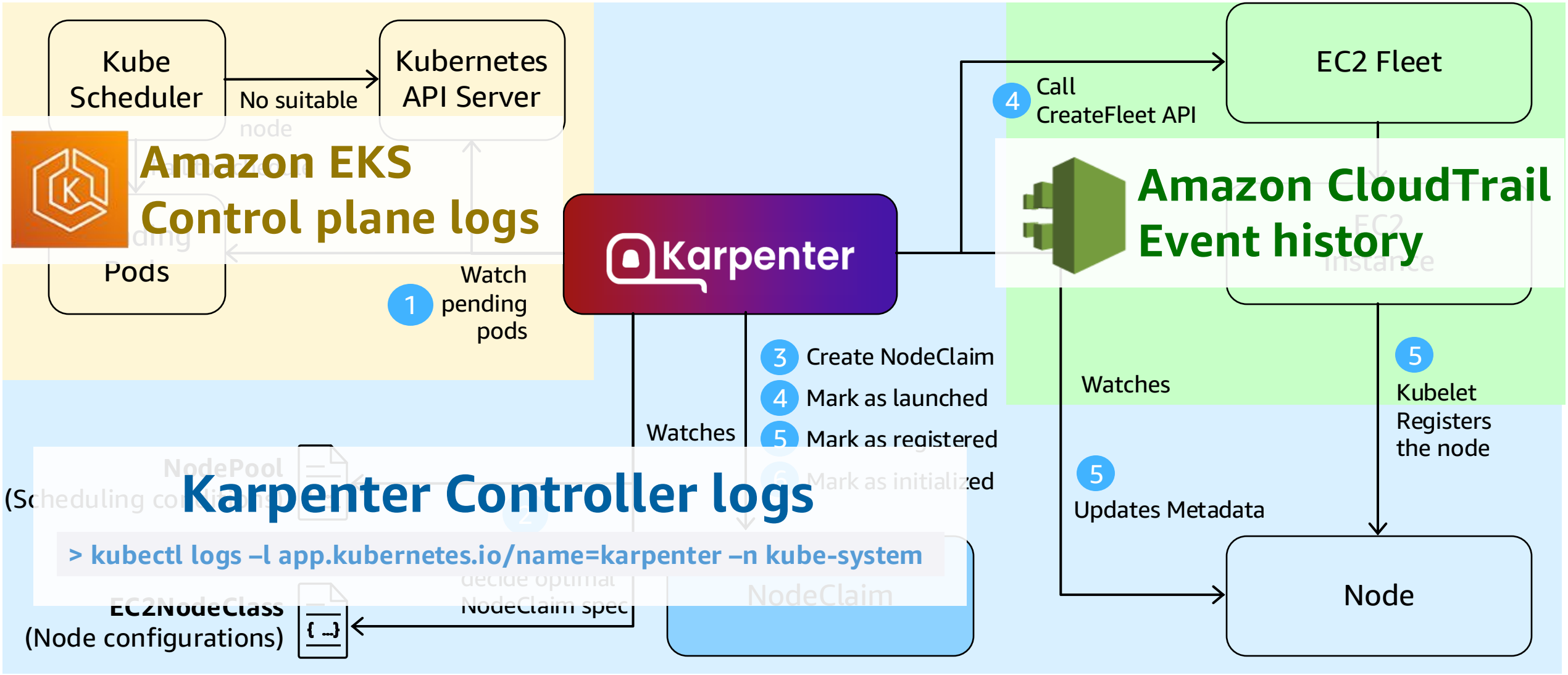
- 종료를 요청한 주체



- AutoScaling
  - BidEvictedEvent
  - AZ Balance
  - StatusCheckFailed_System



Amazon EKS Control plane logs



Amazon EKS Control plane logs

Web Console → EKS → Observability → Control plane logs (default disabled)

Log name	Components
API server	kube-apiserver
Controller manager	kube-controller-manager
Scheduler	kube-scheduler
Authenticator	**EKS Only**, IAM Credentials 을 사용하여 Kubernetes RBAC 인증하는 Components
Audit	Kubernetes auditing, 클러스터의 작업 순서를 문서화하는 보안 관련 시간별 레코드 세트를 제공한다. 클러스터는 사용자, 쿠버네티스 API를 사용하는 애플리케이션 및 컨트롤 플레인 자체에서 생성된 활동을 감사한다.

- CloudWatch Logs로 제공됨
  - CloudWatch의 각 Amazon EKS 클러스터에 대한 그룹에 로그 스트림으로 전송

사례 3) Amazon EKS Control plane logs

“ Karpenter 는 Control plane components 를 갖고있지 않다. “

1) Control plane과 Karpenter controller 간의 요청/응답 확인
2) 확인 할 수 있는 Data plane 로그가 남아있지 않는 경우
   - 노드 및 파드가 제거된 경우
   - Karpenter 로그가 남아있지 않은 경우

⚠️ 각 Log 의 구성요소가 다르기 때문에, 확인이 필요한 정보 및 요구사항에 맞게 적절한 쿼리를 작성해야 한다.

```
filter @logStream like /^kube-apiserver-audit/
| fields @timestamp, userAgent, verb, objectRef.resource, objectRef.name, @message
| filter verb not like /(watch|get)/
| filter @message like /karpenter/
| sort @timestamp desc
| limit 1000
```

Query generator

Saved and sample queries

Query commands

Run query | Cancel | Save | History

Logs Insights QL query can run for maximum of 60 minutes.

Completed. Query executed for 1 log group.

Logs (1k) | Patterns (27) | Visualization

Logs (1k)

Summarize results | Investigate | Export results | Add to dashboard

Data may cross Regions

Showing 1000 of 15,110 records matched

894,684 records (962.5 MB) scanned in 9.4s @ 95,108 records/s (102.3 MB/s)

Show histogram

#	@timestamp	userAgent	verb	objectRef_	objectRef.name	@message
▶ 388	2025-09-16T18:22:39.7…	kube-controller-mana…	update	deployments	karpenter	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 389	2025-09-16T18:22:39.7…	kube-controller-mana…	update	replicasets	karpenter-8ff455d87	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 390	2025-09-16T18:22:39.6…	kubelet/v1.33.0 (lin…	create	events	karpenter-8ff455d87-wgp…	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 391	2025-09-16T18:22:39.6…	Kubernetes Java Clie…	delete	pods	karpenter-8ff455d87-wgp…	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 392	2025-09-16T18:22:39.5…	kube-scheduler/v1.33…	create	events	karpenter-8ff455d87-jf2…	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 393	2025-09-16T18:22:39.5…	kube-scheduler/v1.33…	patch	pods	karpenter-8ff455d87-jf2…	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 394	2025-09-16T18:22:39.5…	kube-controller-mana…	create	events	karpenter-8ff455d87.186…	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 395	2025-09-16T18:22:39.5…	kube-controller-mana…	update	replicasets	karpenter-8ff455d87	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 396	2025-09-16T18:22:39.5…	kube-controller-mana…	create	pods		{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque

CloudWatch Logs → Log Insights

- CloudWatch Logs에서 Amazon EKS 컨트롤 플레인 로그를 검색하려면 어떻게 해야 합니까?
  - https://repost.aws/ko/knowledge-center/eks-get-control-plane-logs
  - → 일반적인 EKS 사용 사례의 샘플 쿼리

kube-system 네임스페이스에서 DaemonSets/Addons 의 변경 사항 찾기

```
filter @logStream like /^kube-apiserver-audit/
| fields @logStream, @timestamp, @message
| filter verb like /(create|update|delete)/ and
strcontains(requestURI,"/apis/apps/v1/namespaces/kube-system/daemonsets")
| sort @timestamp desc
| limit 50
```

Log Insights 쿼리 예시

- karpenter 라는 단어가 포함된 로그

```
filter @logStream like /^kube-apiserver-audit/
| fields @timestamp, userAgent, verb, objectRef.resource, objectRef.name, @message
| filter verb not like /watch/
| filter @message like /karpenter/
| sort @timestamp desc
| limit 1000
```

- nodeclaims 의 delete/create 로그

```
filter @logStream like /^kube-apiserver-audit/
| fields @timestamp, userAgent, verb, objectRef.resource, objectRef.name, @message
| filter verb like /(delete|create)/
| filter objectRef.resource like /nodeclaims/
| sort @timestamp desc
| limit 1000
```

사례 3) Amazon EKS Control plane logs

#	@timestamp	userAgent	verb	objectRef…	objectRef.name	@message
▶ 1	2025-09-16T16:30:31.8…	karpenter/1.7.0	delete	nodeclaims	default-45tm8	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▶ 2	2025-09-16T16:29:53.9…	karpenter/1.7.0	create	nodeclaims		{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque
▼ 3	2025-09-16T16:27:36.7…	karpenter/1.7.0	delete	nodeclaims	default-b2q4b	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Reque

Field	Value
@aws.account	266735814573
@aws.region	us-east-1
@ingestionTime	1758007660564
@log	266735814573:/aws/eks/eks-workshop/cluster
@logStream	kube-apiserver-audit-bc5085610aee8606bf50137a932c6c48
@message	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"RequestResponse","audit ba9b-4d14820dea78","stage":"ResponseComplete","requestURI":"/apis/karpenter.sh/ {"username":"system:serviceaccount:kube-system:karpenter","uid":"59489aa4-b496-

responseObject.status.conditions[]

nTime":"2025-09-16T07:23:58Z","message":"","observedGeneration":1,"reason":"Launched","status":"True","type":"Launched"}

nTime":"2025-09-16T07:24:23Z","message":"","observedGeneration":1,"reason":"Registered","status":"True","type":"Registered"}

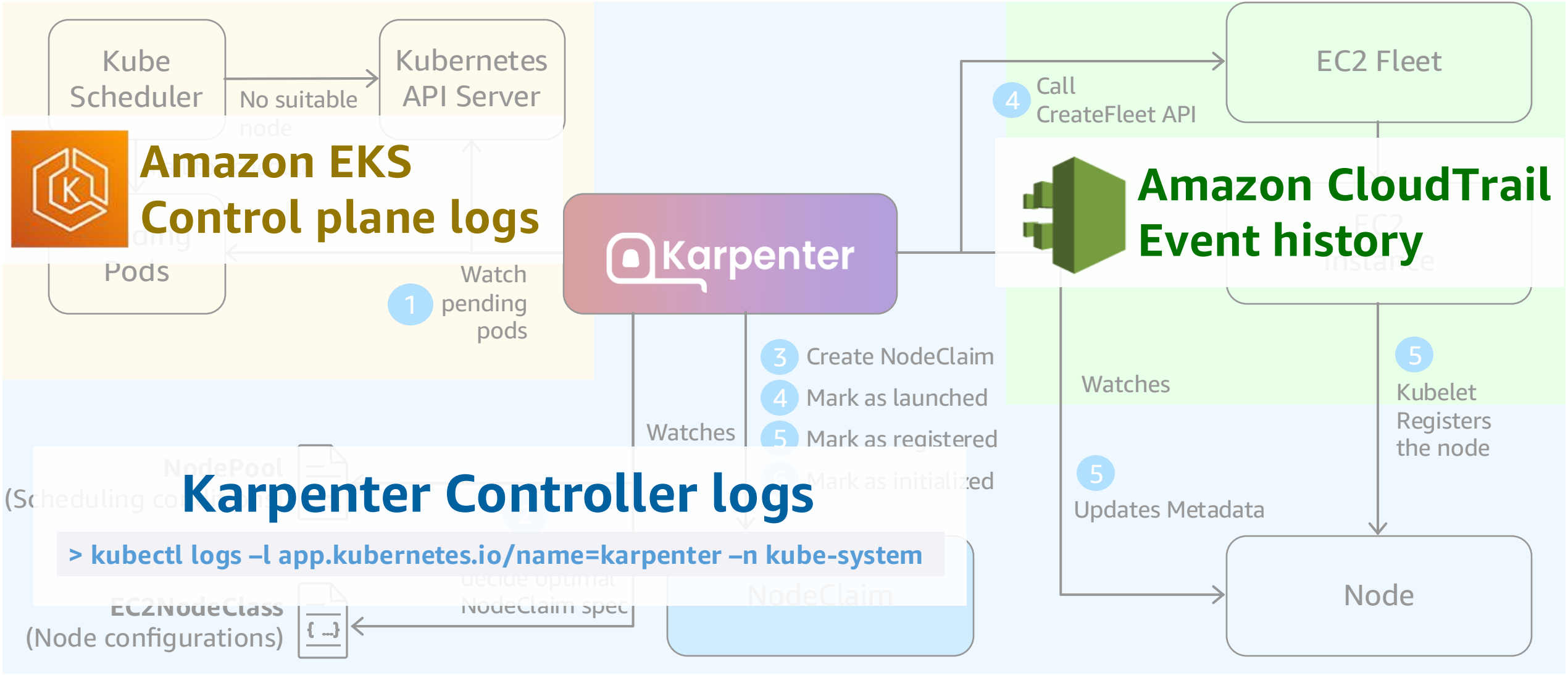
nTime":"2025-09-16T07:24:35Z","message":"","observedGeneration":1,"reason":"Initialized","status":"True","type":"Initialized"}

nTime":"2025-09-16T07:25:06Z","message":"","observedGeneration":1,"reason":"Consolidatable","status":"True","type":"Consolidat

nTime":"2025-09-16T07:26:59Z","message":"NodePoolDrifted","observedGeneration":1,"reason":"NodePoolDrifted","status":"True","t

nTime":"2025-09-16T07:26:59Z","message":"Drifted","observedGeneration":1,"reason":"Drifted","status":"True","type":"Disruption

Troubleshooting



Q&A

Thank you!

박동혁, 이호성, 황은빛

AWS Cloud Support Engineer