Kubernetes Node Autoscaling

Brandon Wagner | AWS

Agenda

- What is Kubernetes?
  - Pods
  - Scheduling
- Node Autoscaling
  - Cluster Autoscaler
  - Karpenter
- DEMO

Agenda

- What is Kubernetes?
  - Pods
  - Scheduling
- Node Autoscaling
  - Cluster Autoscaler
  - Karpenter
- DEMO

Kubernetes (K8s)

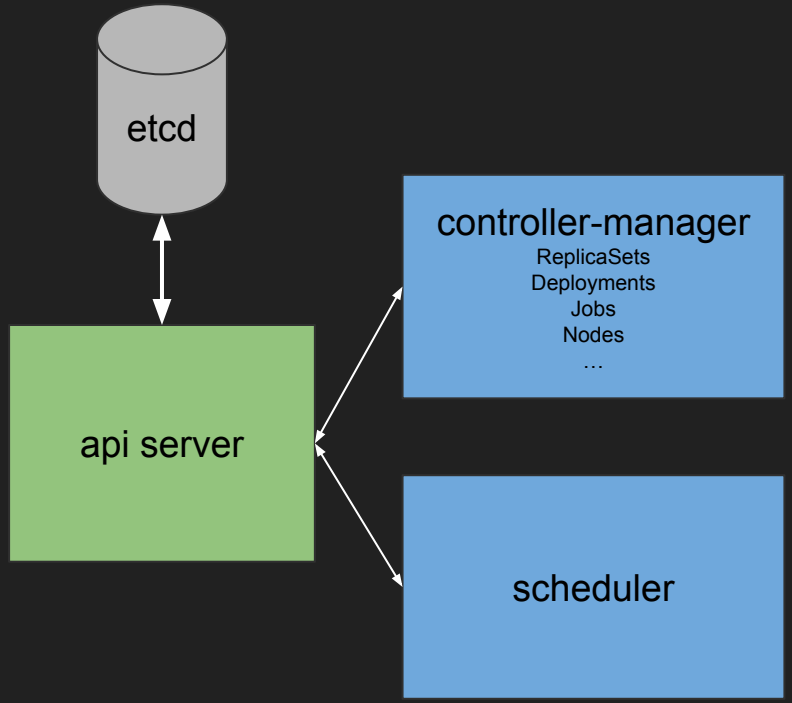
- Open Source Container Orchestration System

- Declarative Resources
  - Controllers Reconcile Resources to Desired State
  - Self Healing
  - Extensible by Adding Controllers

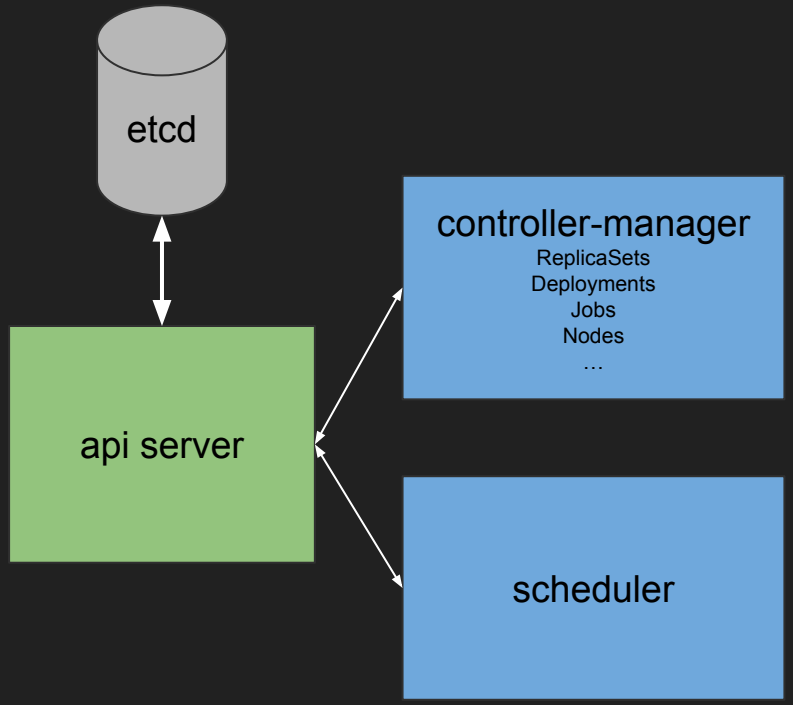
Kubernetes Resources

- Pod
  - 1 or more containers (smallest resource unit)
  - Immutable
- Deployment
  - A group of identical pods
  - If 1 pod dies, the deployment controller creates a new pod
- DaemonSet
  - Pods that are deployed to every node

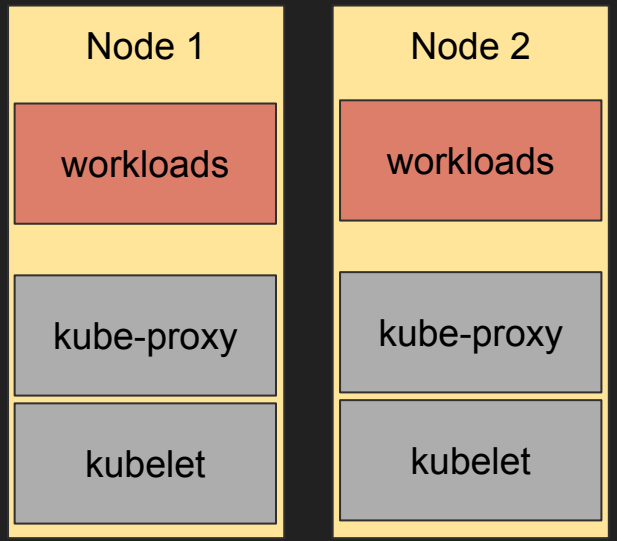
Kubernetes (K8s) - Control Plane



Kubernetes (K8s)

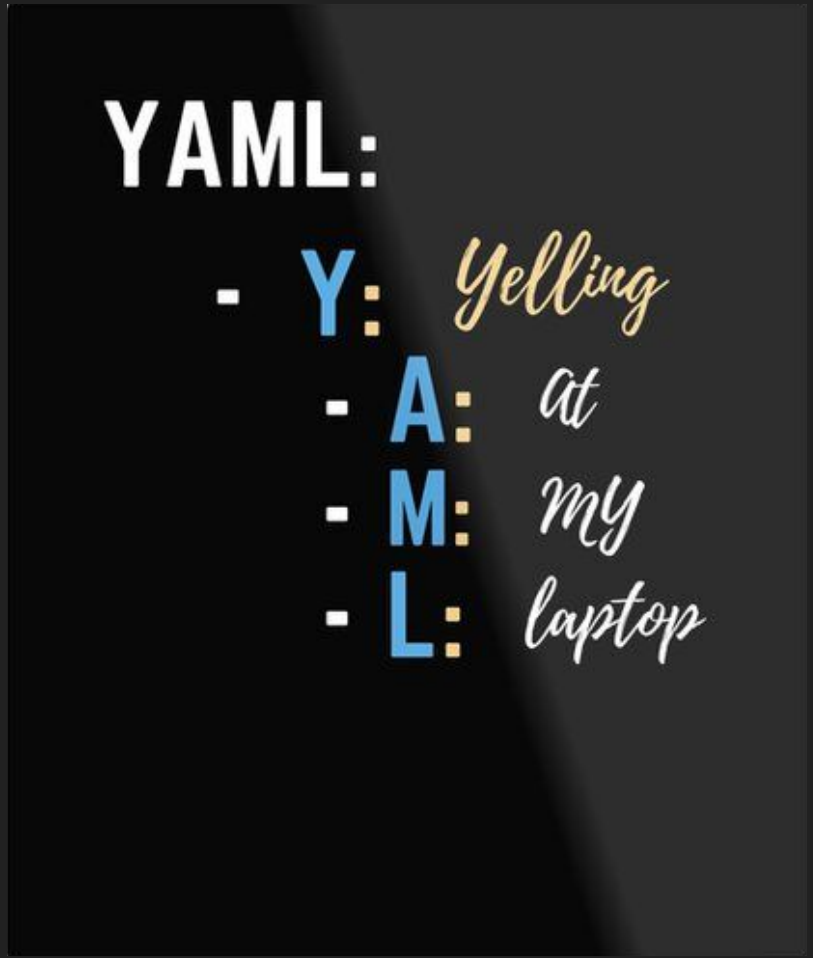


Data Plane



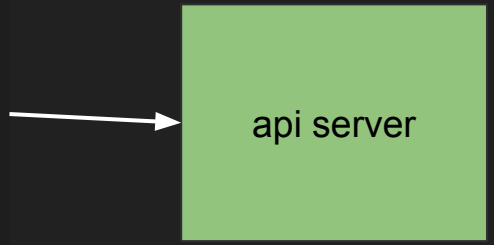
Kubernetes (K8s)

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```



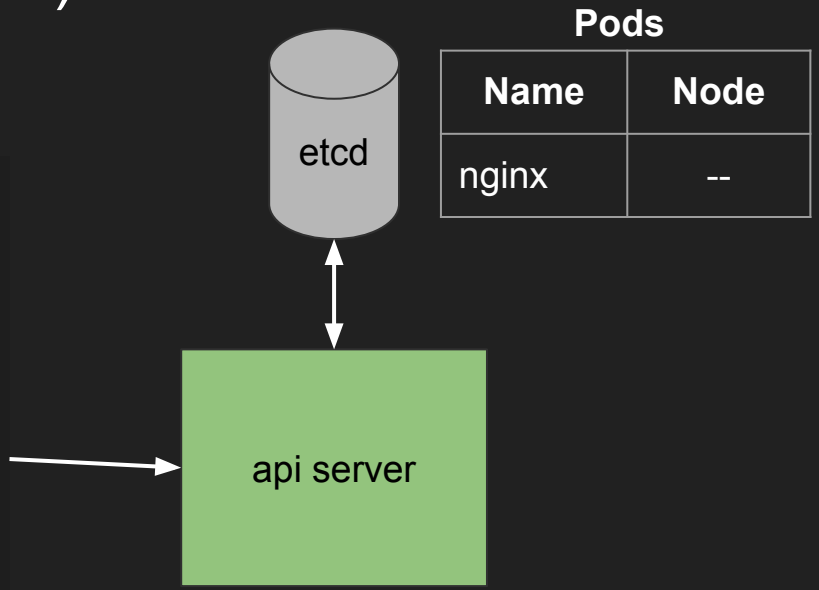
Kubernetes (K8s)

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```



Kubernetes (K8s)

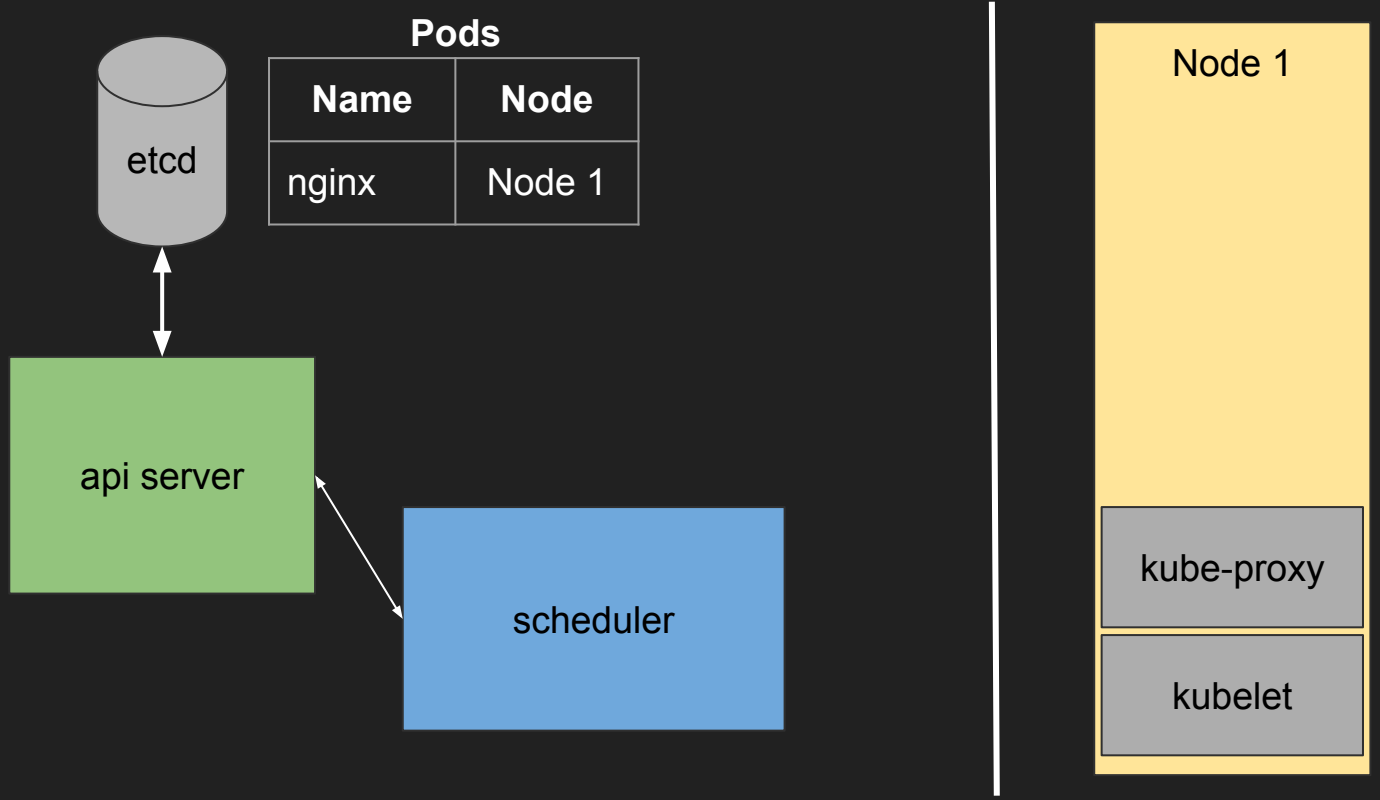
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```



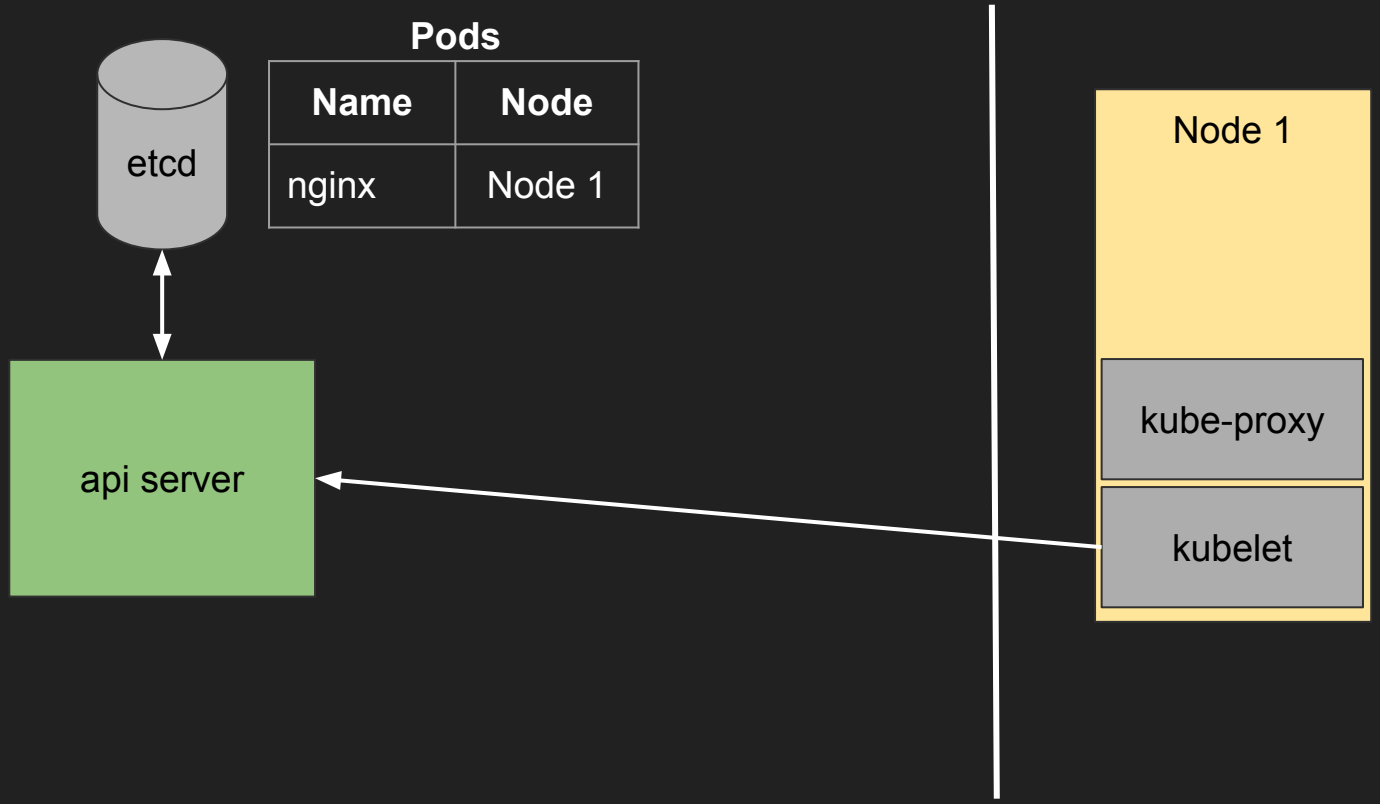
Pods

Name	Node
nginx	--

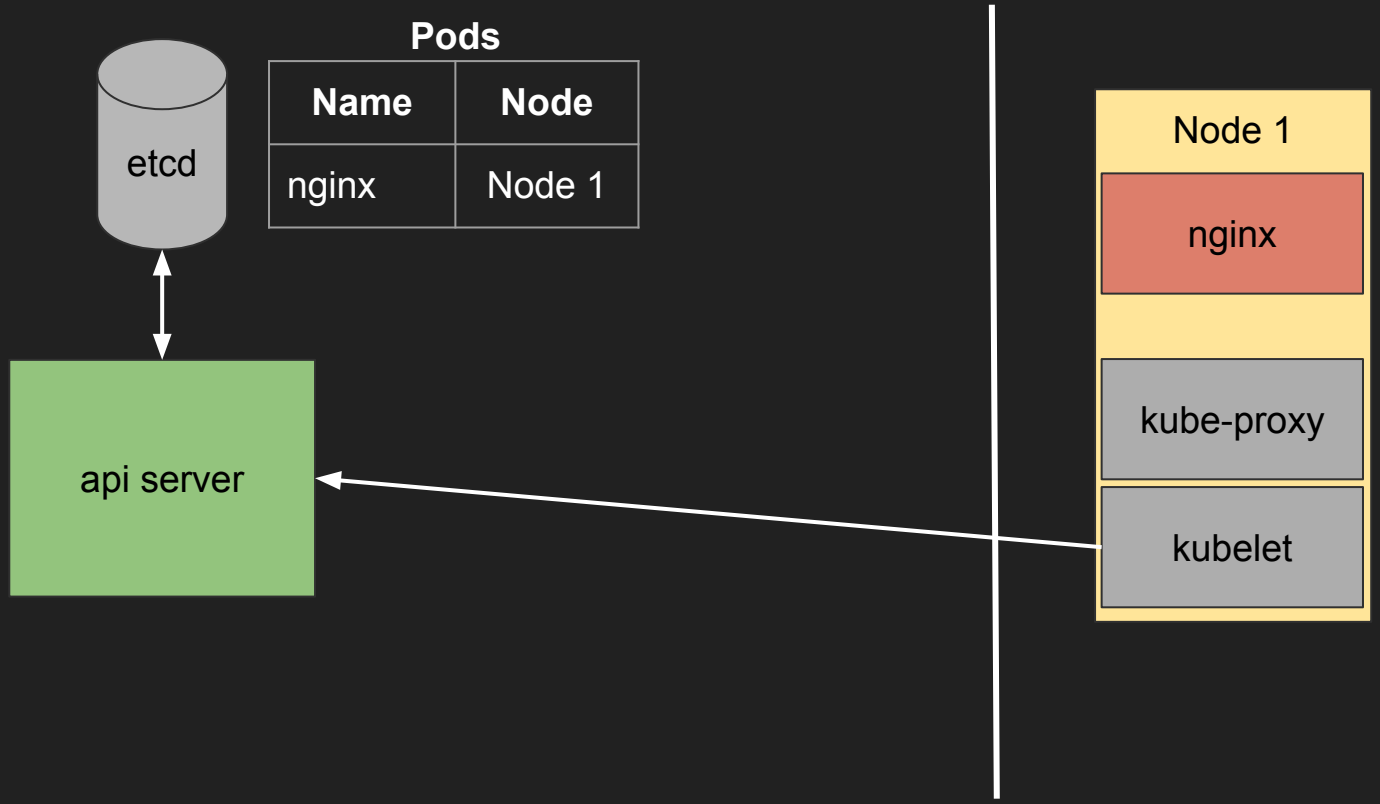
Kubernetes (K8s)



Kubernetes (K8s)

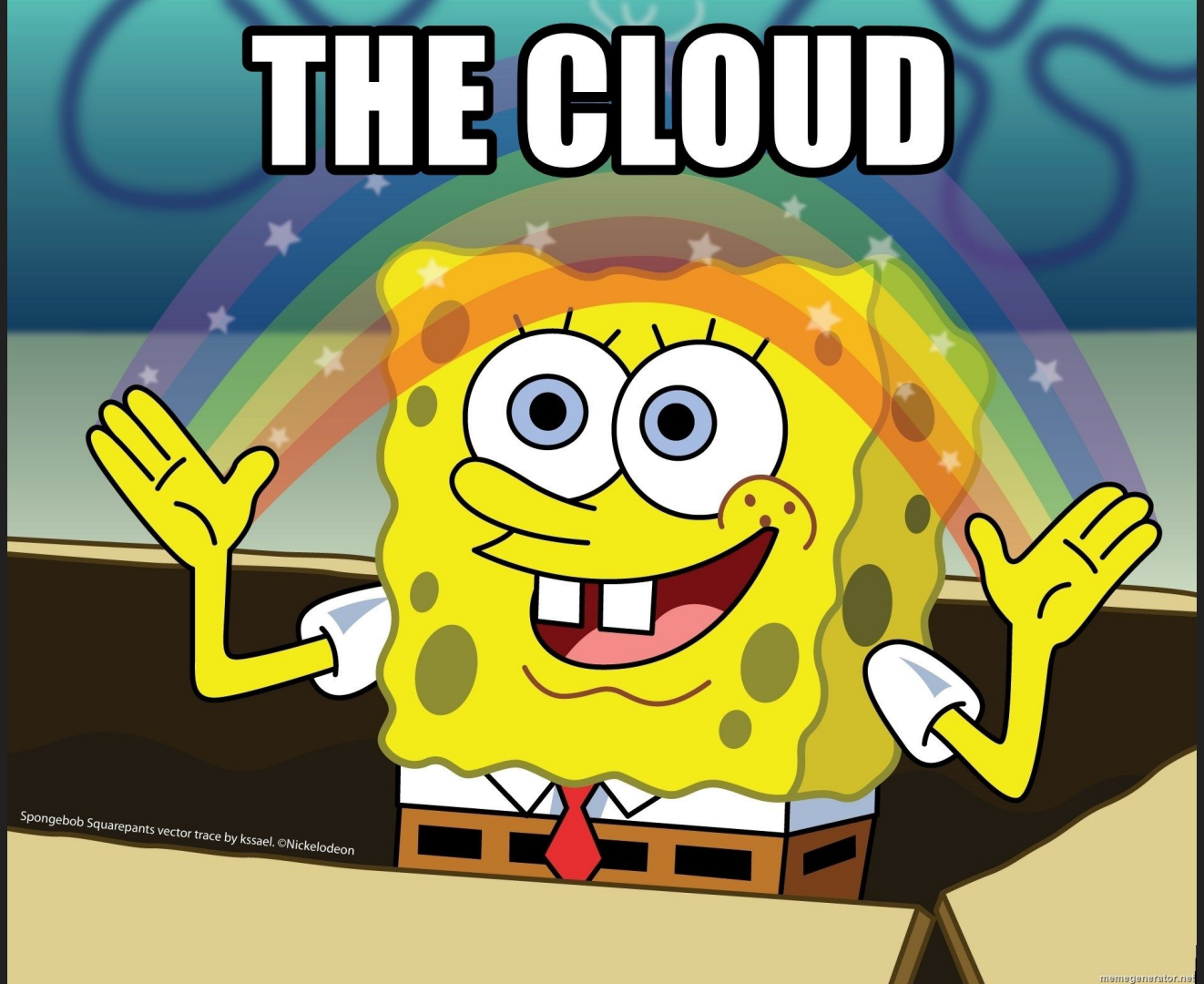


Kubernetes (K8s)

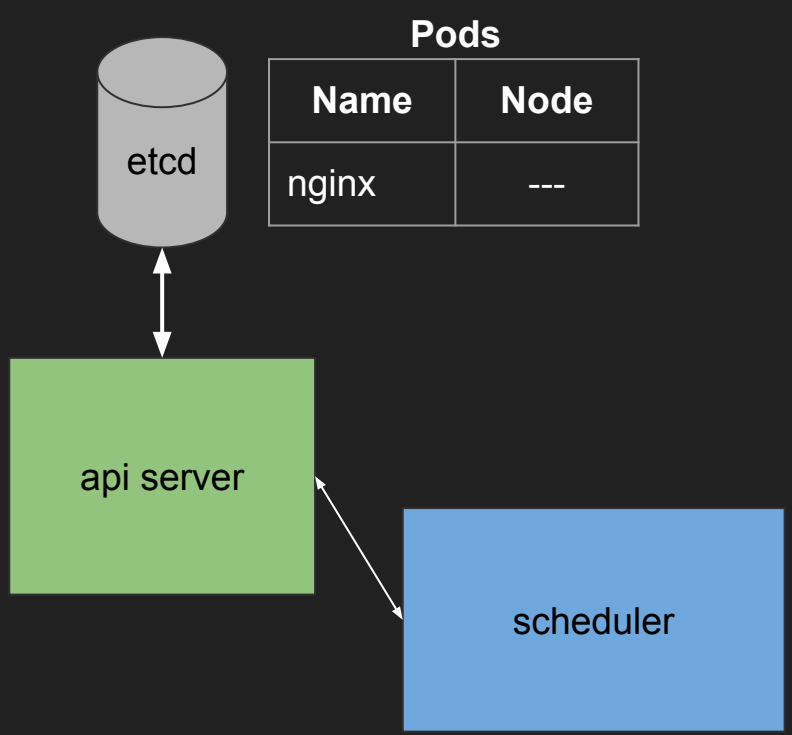



WHAT IF I TOLD YOU
a node isn’t available?
GenerateMemes.com

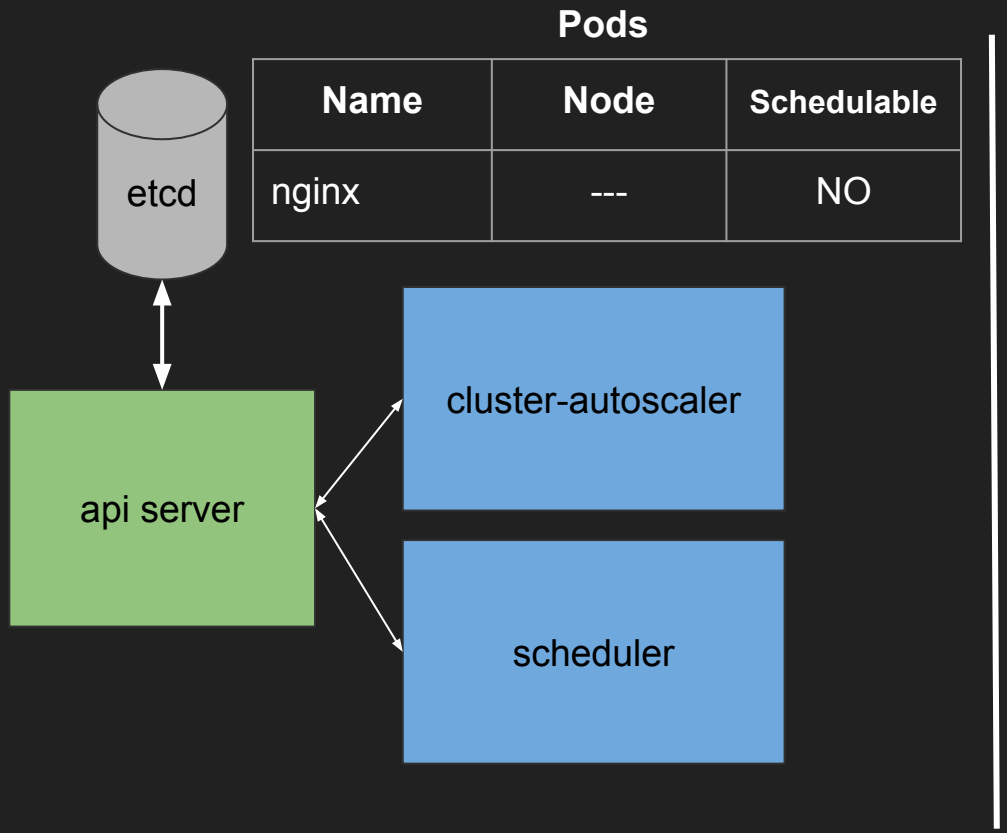

I'M SCARED
makeameme.org


THE CLOUD
Spongebob Squarepants vector trace by kssael. ©Nickelodeon

Kubernetes (K8s)



Kubernetes (K8s)



Kubernetes (K8s)

Pods

Name	Node	Schedulable
nginx	---	NO

etcd

api server

cluster-autoscaler

scheduler

Auto-Scaling Group
(m5.xlarge)

Node 1

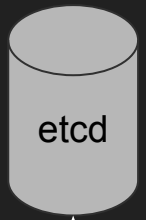
kube-proxy

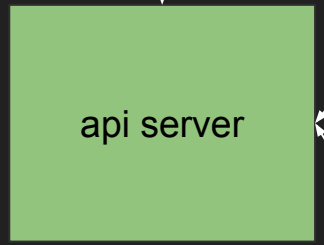
kubelet

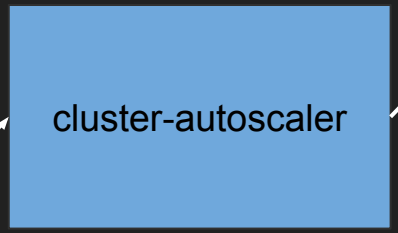
Kubernetes (K8s)

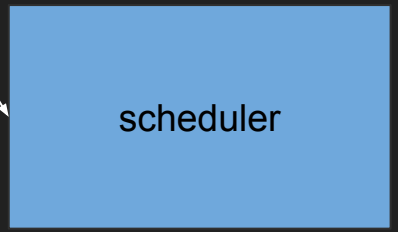
Pods

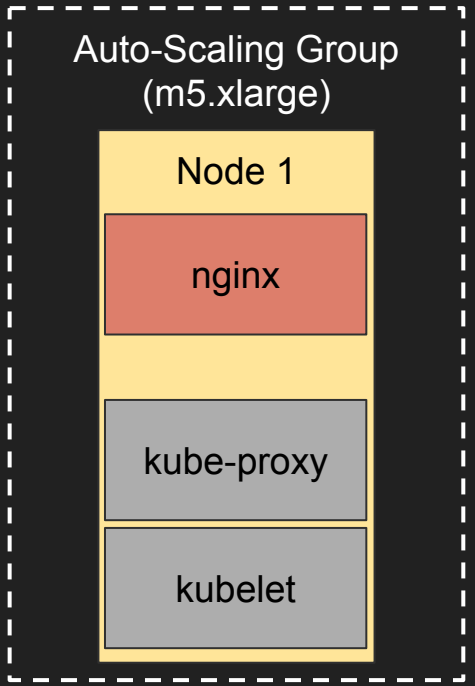
Name	Node	Schedulable
nginx	Node 1	NO











Kubernetes Resources

- Pod
  - 1 or more containers (smallest resource unit)
  - Immutable
- Deployment
  - A group of identical pods
  - If 1 pod dies, the deployment controller creates a new pod
- DaemonSet
  - Pods that are deployed to every node

K8s Scheduling

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.14.2
        name: nginx
        resources:
          requests:
            cpu: "1"
            memory: 256M
```

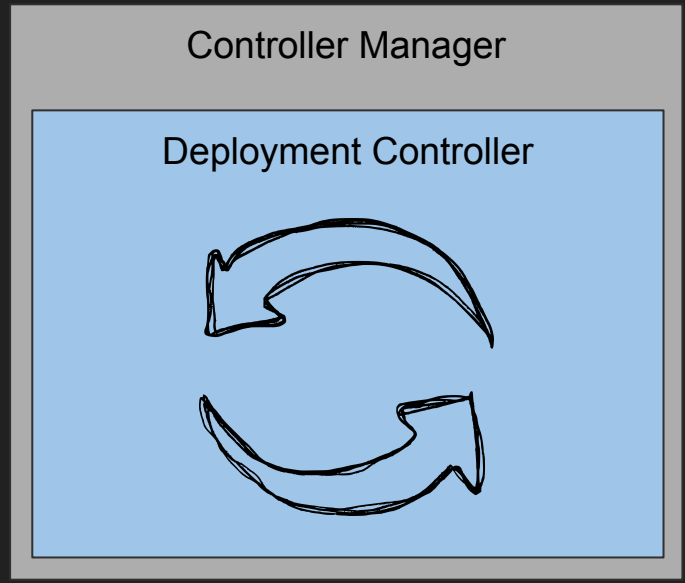
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.14.2
        name: nginx
        resources:
          requests:
            cpu: "1"
            memory: 256M
```

Deployments

Name	Replicas
nginx	3

Pods

Name	Node



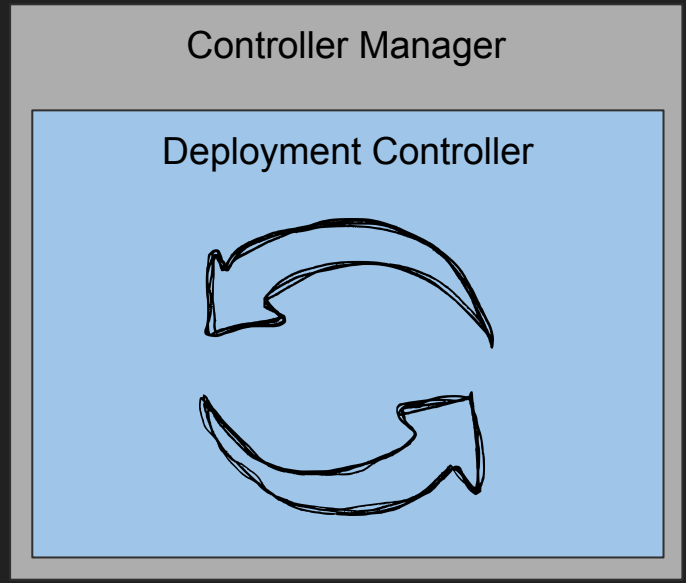
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.14.2
        name: nginx
        resources:
          requests:
            cpu: "1"
            memory: 256M
```

Deployments

Name	Replicas
nginx	3

Pods

Name	Node
nginx-1	---
nginx-2	---
nginx-3	---



K8s Scheduling - Deployment

Deployments

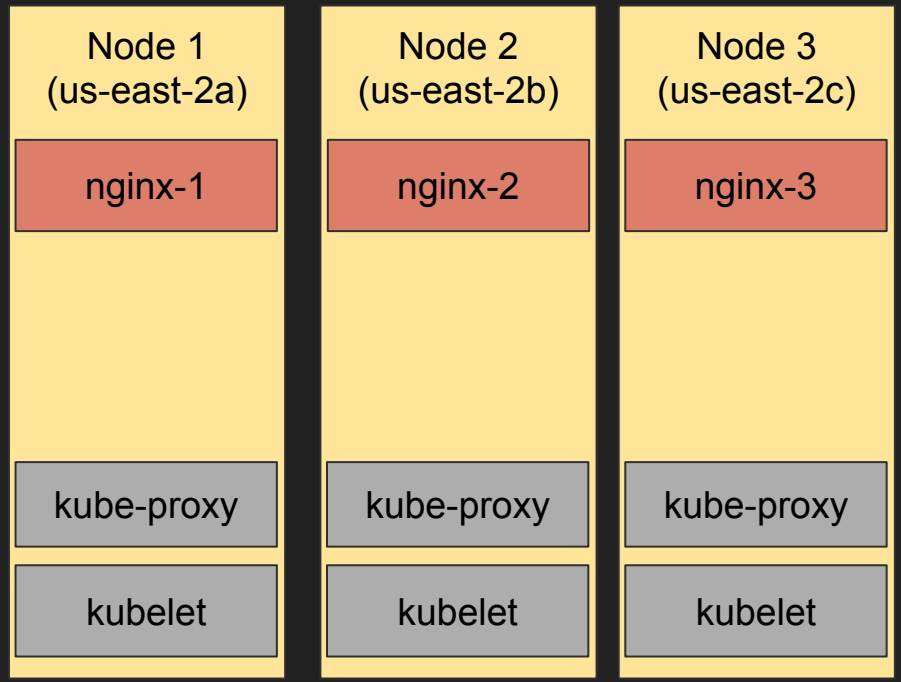
Name	Replicas
nginx	3

Pods

Name	Node
nginx-1	Node 1
nginx-2	Node 1
nginx-3	Node 1

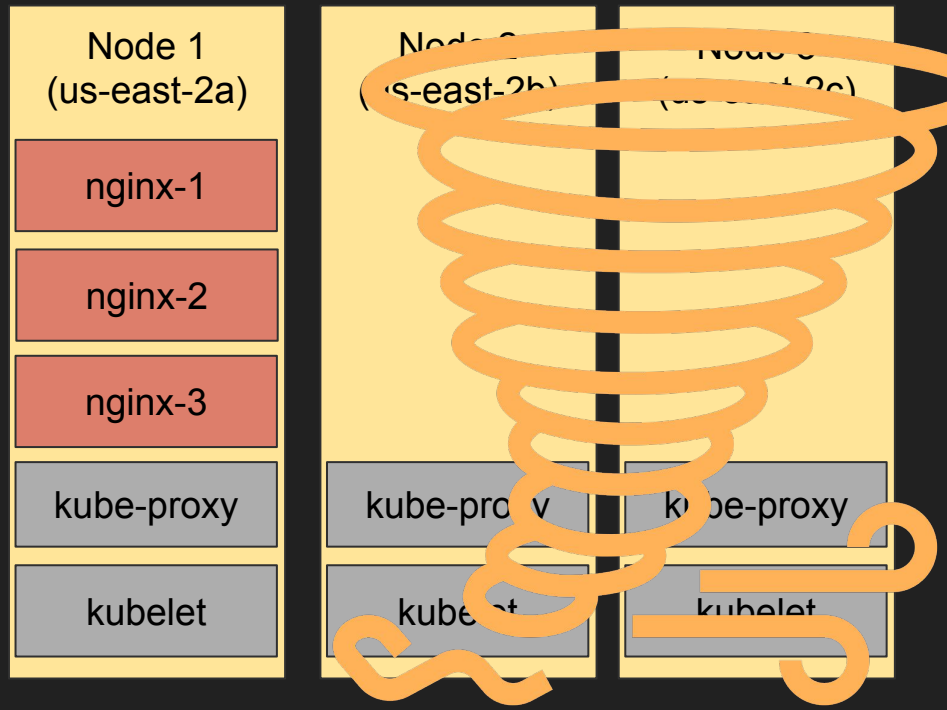
K8s Scheduling - Topology Spread

```yaml
spec:
  containers:
  - image: nginx:1.14.2
    name: nginx
    resources:
      requests:
        cpu: "1"
        memory: 256M
  topologySpreadConstraints:
  - labelSelector:
      matchLabels:
        app: nginx
    maxSkew: 1
    topologyKey: topology.kubernetes.io/zone
    whenUnsatisfiable: ScheduleAnyway
```



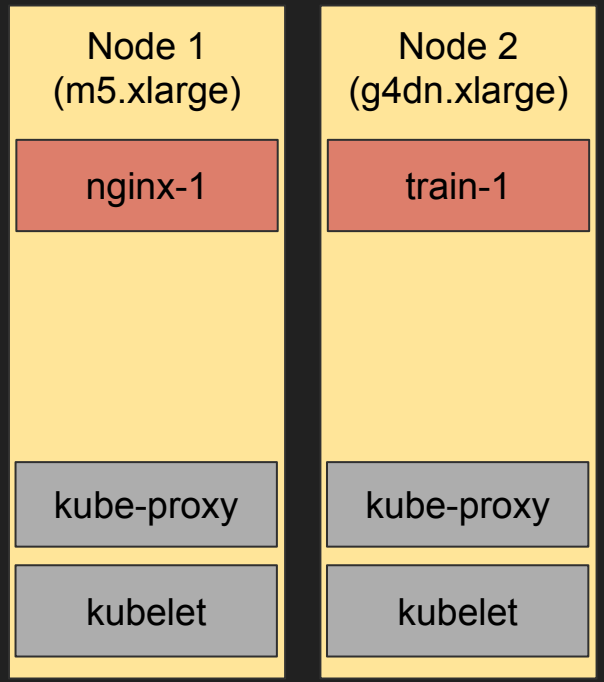
K8s Scheduling - Topology Spread

```yaml
spec:
  containers:
  - image: nginx:1.14.2
    name: nginx
    resources:
      requests:
        cpu: "1"
        memory: 256M
  topologySpreadConstraints:
  - labelSelector:
      matchLabels:
        app: nginx
    maxSkew: 1
    topologyKey: topology.kubernetes.io/zone
    whenUnsatisfiable: ScheduleAnyway
```

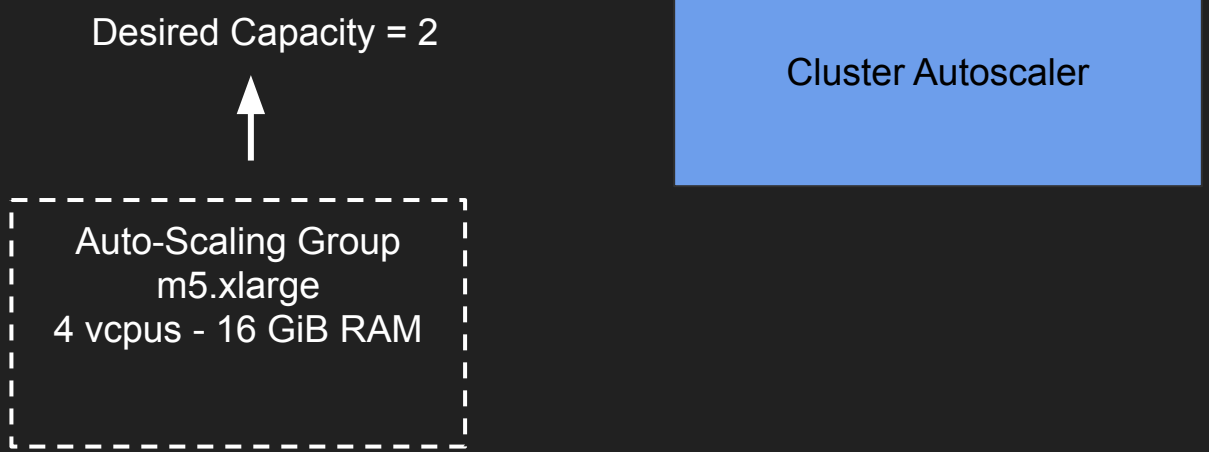


K8s Scheduling - GPU Extended Resource

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: train
spec:
  replicas: 1
  selector:
    matchLabels:
      app: train
  template:
    metadata:
      labels:
        app: train
    spec:
      containers:
      - image: my-trainer-workload:1.0
        name: train
        resources:
          limits:
            nvidia.com/gpu: "1"
          requests:
            cpu: "1"
            memory: 256M
```



K8s Cluster Autoscaler



K8s Cluster Autoscaler

Auto-Scaling Group
1 gpu

Auto-Scaling Group
2 gpus

Auto-Scaling Group
4 gpus

Auto-Scaling Group
8 gpus

Cluster Autoscaler

Auto-Scaling Group
4 vcpus - 16 GiB RAM

Auto-Scaling Group
8 vcpus - 32 GiB RAM

Auto-Scaling Group
16 vcpus - 64 GiB RAM

Auto-Scaling Group
32 vcpus - 128 GiB RAM

K8s Cluster Autoscaler

Auto-Scaling Group
1 gpu

Auto-Scaling Group
2 gpus

Auto-Scaling Group
4 gpus

Auto-Scaling Group
8 gpus

Cluster Autoscaler

Auto-Scaling Group
4 vcpus - 16 GiB RAM
us-east-2a

Auto-Scaling Group
8 vcpus - 32 GiB RAM
us-east-2a

Auto-Scaling Group
16 vcpus - 64 GiB RAM
us-east-2a

Auto-Scaling Group
32 vcpus - 128 GiB RAM
us-east-2a

K8s Cluster Autoscaler

Auto-Scaling Group

1 gpu
4 vcpus - 16 GiB RAM
us-east-2a
us-east-2b
us-east-2c

Auto-Scaling Group

2 gpus
8 vcpus - 32 GiB RAM
us-east-2a
us-east-2b
us-east-2c

Auto-Scaling Group

4 gpus
16 vcpus - 64 GiB RAM
us-east-2a
us-east-2b
us-east-2c

Auto-Scaling Group

8 gpus
32 vcpus - 128 GiB RAM
us-east-2a
us-east-2b
us-east-2c

Cluster Autoscaler

24 Groups

Still missing a lot of instance shapes

What about…

Spot & on-demand
x86 + ARM
New instance types released

Groupless Node Autoscaling with

Overview

- Groupless Node Autoscaling
- Kubernetes Native
- Vendor Neutral
- Open Source

github.com/aws/karpenter

K8s Native

- Groupless Autoscaling
  - No need to pre-configure autoscaling groups
  - Dynamically looks up instance types and specs from the cloud provider
  - Understands special scheduling requirements
    - GPUs (extended resources)
    - Node Affinity and Anti-Affinity
    - Pod Affinity and Anti-Affinity
    - Topology Spreads
    - Persistent Volumes

K8s Native

- Custom Resource Definitions (CRDs)
  - Provisioner
  - AWSNodeTemplate

K8s Native

- Provisioner

```yaml
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  limits:
    resources:
      cpu: 1k
  providerRef:
    name: default
  requirements:
  - key: karpenter.sh/capacity-type
    operator: In
    values:
    - on-demand
    - spot
  - key: kubernetes.io/arch
    operator: In
    values:
    - amd64
    - arm64
  ttlSecondsAfterEmpty: 30
```

K8s Native

- Provisioner
  - Limits

```yaml
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  limits:
    resources:
      cpu: 1k
  providerRef:
    name: default
  requirements:
  - key: karpenter.sh/capacity-type
    operator: In
    values:
    - on-demand
    - spot
  - key: kubernetes.io/arch
    operator: In
    values:
    - amd64
    - arm64
  ttlSecondsAfterEmpty: 30
```

K8s Native

- Provisioner
  - Limits
  - Provider Reference
  - Requirements

```yaml
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  limits:
    resources:
      cpu: 1k
  providerRef:
    name: default
  requirements:
  - key: karpenter.sh/capacity-type
    operator: In
    values:
    - on-demand
    - spot
  - key: kubernetes.io/arch
    operator: In
    values:
    - amd64
    - arm64
  ttlSecondsAfterEmpty: 30
```

K8s Native

- Provisioner
  - Limits
  - Provider Reference
  - Requirements
  - TTL Seconds After Empty

```yaml
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  limits:
    resources:
      cpu: 1k
  providerRef:
    name: default
  requirements:
  - key: karpenter.sh/capacity-type
    operator: In
    values:
    - on-demand
    - spot
  - key: kubernetes.io/arch
    operator: In
    values:
    - amd64
    - arm64
  ttlSecondsAfterEmpty: 30
```

K8s Native

- AWS Node Template
  - AMI Family

```yaml
apiVersion: karpenter.k8s.aws/v1alpha1
kind: AWSNodeTemplate
metadata:
  name: default
spec:
  amiFamily: Bottlerocket
  subnetSelector:
    karpenter.sh/discovery: "my-cluster"
  securityGroupSelector:
    karpenter.sh/discovery: "my-cluster"
```

K8s Native

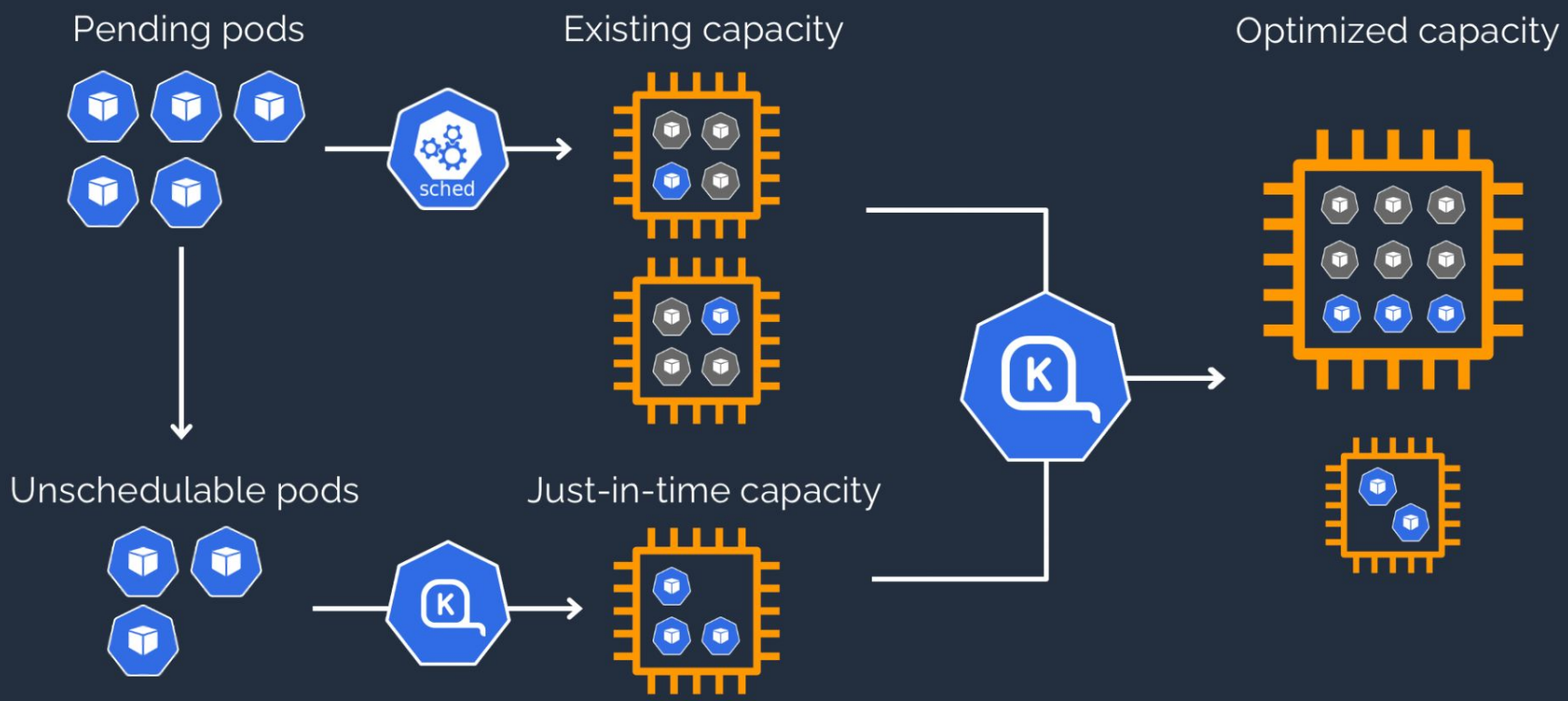
- AWS Node Template
  - AMI Family
  - Subnet Selector

```yaml
apiVersion: karpenter.k8s.aws/v1alpha1
kind: AWSNodeTemplate
metadata:
  name: default
spec:
  amiFamily: Bottlerocket
  subnetSelector:
    karpenter.sh/discovery: "my-cluster"
  securityGroupSelector:
    karpenter.sh/discovery: "my-cluster"
```

K8s Native

- AWS Node Template
  - AMI Family
  - Subnet Selector
  - Security Group Selector

```yaml
apiVersion: karpenter.k8s.aws/v1alpha1
kind: AWSNodeTemplate
metadata:
  name: default
spec:
  amiFamily: Bottlerocket
  subnetSelector:
    karpenter.sh/discovery: "my-cluster"
  securityGroupSelector:
    karpenter.sh/discovery: "my-cluster"
```


Pending pods
sched
Existing capacity
Optimized capacity
Unschedulable pods
K
Just-in-time capacity
K

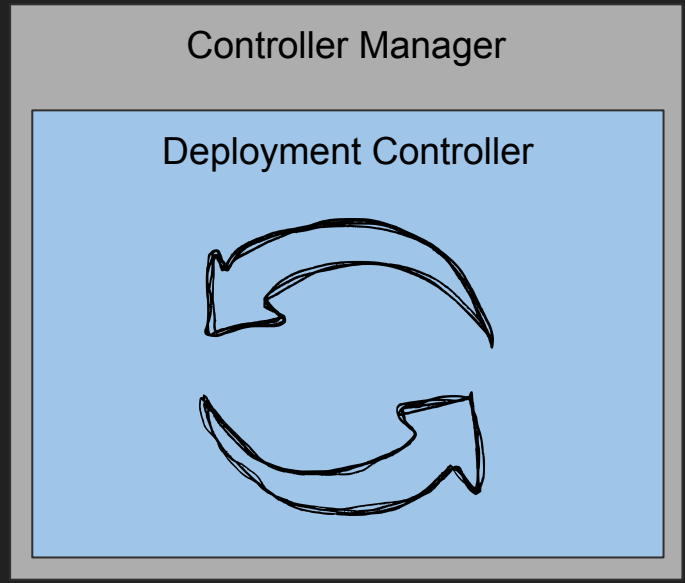
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.14.2
        name: nginx
        resources:
          requests:
            cpu: "1"
            memory: 256M
```

Deployments

Name	Replicas
nginx	3

Pods

Name	Node



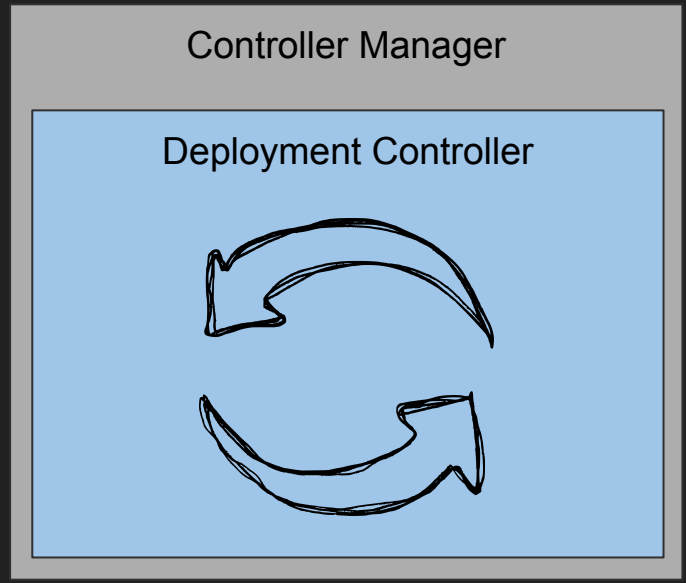
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.14.2
        name: nginx
        resources:
          requests:
            cpu: "1"
            memory: 256M
```

Deployments

Name	Replicas
nginx	3

Pods

Name	Node
nginx-1	---
nginx-2	---
nginx-3	---



K8s Scheduling - Deployment

3 vcpus
768 MiB

c4.xlarge
c5.xlarge
c5a.xlarge
c5ad.xlarge
c5d.xlarge
c5n.xlarge
c6a.xlarge
c6g.xlarge
c6gd.xlarge
c6gn.xlarge
c6i.xlarge
c6id.xlarge
c7g.xlarge
…

K8s Scheduling - Deployment

3 vcpus
768 MiB

c4.xlarge
c5.xlarge
c5a.xlarge
c5ad.xlarge
c5d.xlarge
c5n.xlarge
c6a.xlarge
c6g.xlarge
c6gd.xlarge
c6gn.xlarge
c6i.xlarge
c6id.xlarge
c7g.xlarge
…

K8s Scheduling - Deployment

3 pods
3 vcpus
768 MiB

c4.xlarge
c5.xlarge
c5a.xlarge
c5ad.xlarge
c5d.xlarge
c5n.xlarge
c6a.xlarge
c6g.xlarge
c6gd.xlarge
c6gn.xlarge
c6i.xlarge
c6id.xlarge
c7g.xlarge
m4.xlarge
m5.xlarge
m5a.xlarge
m5ad.xlarge
m5d.xlarge
m5dn.xlarge
m5n.xlarge
m5zn.xlarge
m6a.xlarge
m6g.xlarge
m6gd.xlarge
m6i.xlarge
m6id.xlarge
…

Node 1
m6i.xlarge
spot

nginx-1

nginx-2

nginx-3

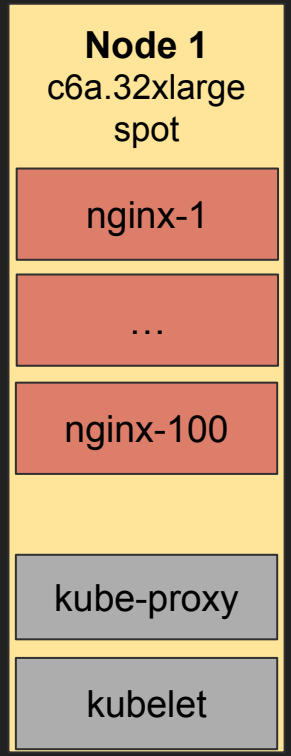
kube-proxy

kubelet

K8s Scheduling - Big Deployments

100 pods
100 vcpus
25 GiB

c6a.32xlarge
c6a.48xlarge
c6i.32xlarge
c6id.32xlarge
i4i.32xlarge
m6a.32xlarge
m6a.48xlarge
m6i.32xlarge
m6id.32xlarge
r6i.32xlarge
r6id.32xlarge
u-6tb1.112xlarge
u-6tb1.56xlarge
x1.32xlarge
x1e.32xlarge
x2idn.32xlarge
x2iedn.32xlarge



What if we can’t get a big instance?

K8s Scheduling - Splitting Pod Batches

100 pods
100 vcpus
25 GiB

c5.18xlarge
c5.24xlarge
c5a.16xlarge
c5a.24xlarge
c5ad.16xlarge
c5ad.24xlarge
c5d.18xlarge
c5d.24xlarge
c5n.18xlarge
c6a.16xlarge
c6a.24xlarge
c6a.32xlarge
c6a.48xlarge
c6g.16xlarge
c6gd.16xlarge
c6gn.16xlarge
c6i.16xlarge
c6i.24xlarge
c6i.32xlarge
c6id.16xlarge
c6id.24xlarge
c6id.32xlarge
c7g.16xlarge

Node 1 c5a.16xlarge spot	**Node 2** c6i.16xlarge spot
nginx-1	nginx-50
…	…
nginx-49	nginx-100
kube-proxy	kube-proxy
kubelet	kubelet

K8s Scheduling - Pod Level Requirements

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:1.14.2
        name: nginx
        resources:
          requests:
            cpu: "1"
            memory: 256M
      nodeSelector:
        karpenter.sh/capacity-type: on-demand
        kubernetes.io/arch: amd64
```

3 pods
3 vcpus
768 MiB

c4.xlarge
c5.xlarge
c5a.xlarge
c5ad.xlarge
c5d.xlarge
c5n.xlarge
c6a.xlarge
c6i.xlarge
c6id.xlarge
m4.xlarge
m5.xlarge
m5a.xlarge
m5ad.xlarge
m5d.xlarge
m5dn.xlarge
m5n.xlarge
m5zn.xlarge
m6a.xlarge
m6i.xlarge
m6id.xlarge
…


BRACE YOURSELVES
A LIVE DEMO IS COMING
memegenerator.net