AWS re:Invent

NOV. 27 – DEC. 1, 2023 | LAS VEGAS, NV

CMP328-R

Optimize Amazon EKS workloads: Spot Instances, Karpenter, and Graviton

Aparajita Jas

(she/her)
Principal GTM Specialist – EC2 Spot

Brandon Wagner

(he/him)
Senior Software Engineer – Amazon EKS

Agenda

01 What is Karpenter

02 Karpenter vs. Cluster Autoscaler

04 Flexibility with Karpenter

05 How Karpenter works

06 Demo

07 Q&A

Amazon EKS is the most trusted and secure way to run Kubernetes

Amazon **EKS**

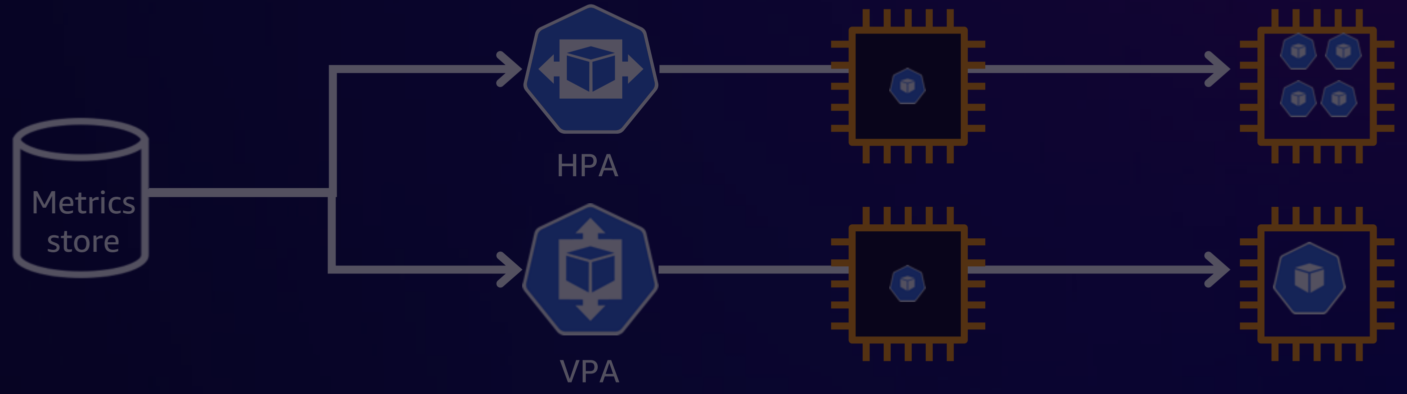
- ✓ Amazon EKS runs vanilla Kubernetes – Amazon EKS is upstream and certified conformant version of Kubernetes (with backported security fixes)
- ✓ Amazon EKS supports 4 versions of Kubernetes, giving customers time to test and roll out upgrades
- ✓ Amazon EKS provides a managed Kubernetes experience for performant, reliable, and secure Kubernetes
- ✓ Amazon EKS makes Kubernetes operations, administration, and management simple and boring

Amazon EKS enables you to build reliable, stable, and secure applications in any environment

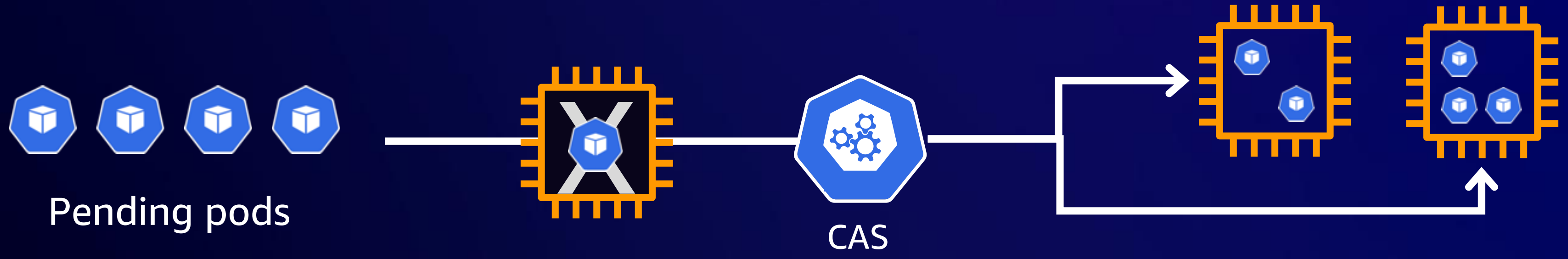
Kubernetes autoscaling overview

1. Horizontal Pod Autoscaler (HPA)

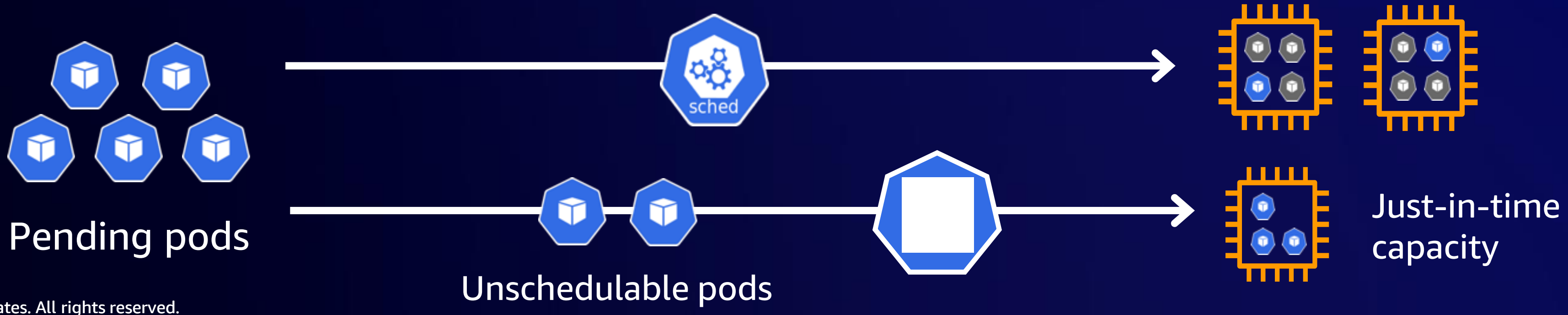
2. Vertical Pod Autoscaler (VPA)



3. Cluster Autoscaler (CAS)



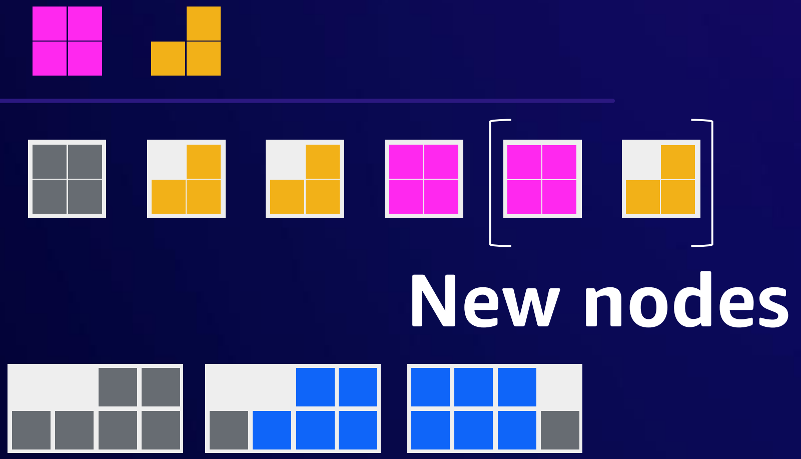
4. Karpenter



Cluster Autoscaler scale-up

HPA >> Pending pods

Amazon EKS Cluster

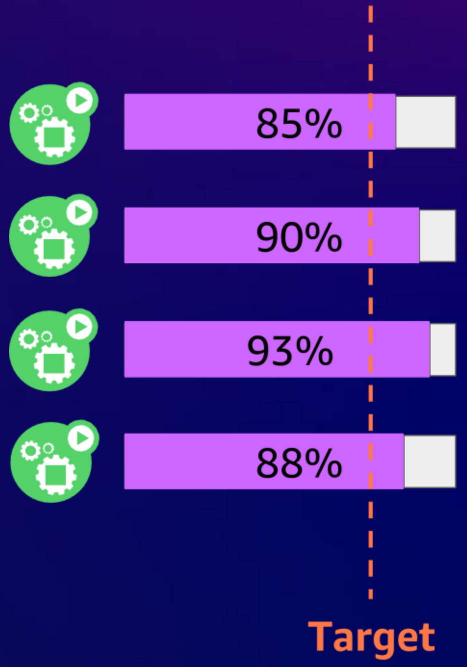


Autoscaling group
4vCPUs 16 GB Amazon
EC2 Spot

10 sec

Expander

Autoscaling group
8vCPUs 32 GB
On-Demand



⬜	Node
🟨	1 vCPU request

Karpenter scale-up

HPA >> Pending pods

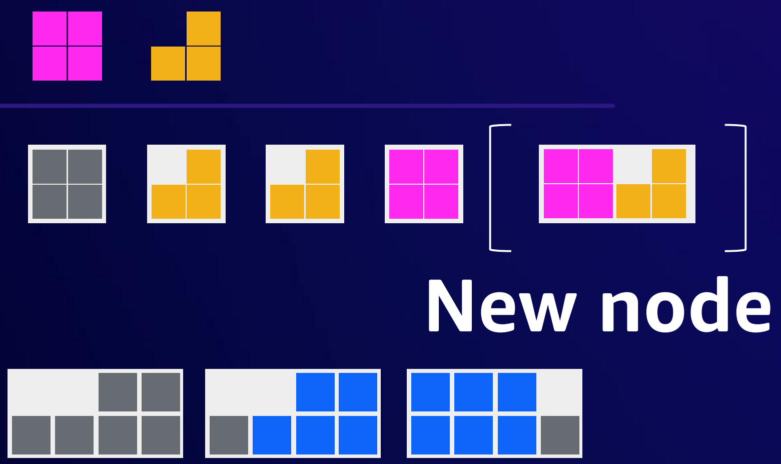
Amazon EKS Cluster

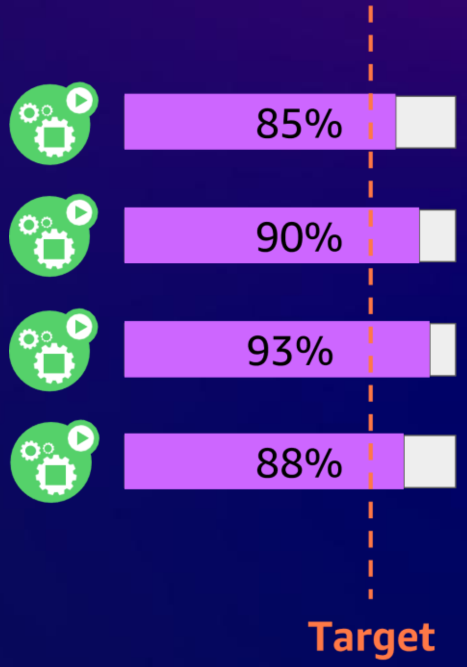
Dynamic

Default: All standard instance types

OR

instanceFamilies: [m5, m5a, m6i, . . .]





Provisioning and scheduling decisions

- Works with kube-scheduler to provision the right set of nodes
- Supports all scheduling constraints: Topology Spread, Node/Pod Affinity and Anti-Affinity, etc.

□ Node

🟨 1 vCPU request

Karpenter

Karpenter simplifies Kubernetes infrastructure with just-in-time optimal resource provisioning

Application-first infrastructure

Node provisioning based on Pod requirements

Multiple instance types

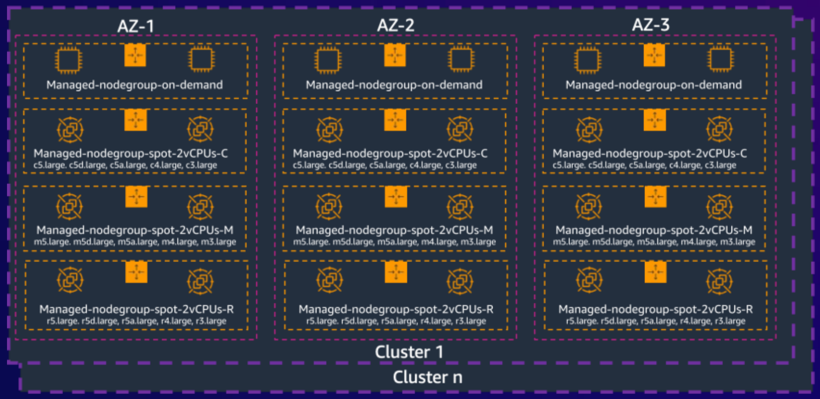
Single configuration with diverse instance types, sizes, architectures, Availability Zones

Diversify with Amazon EC2 Spot and On-Demand

Karpenter helps simplify diversification across purchase options

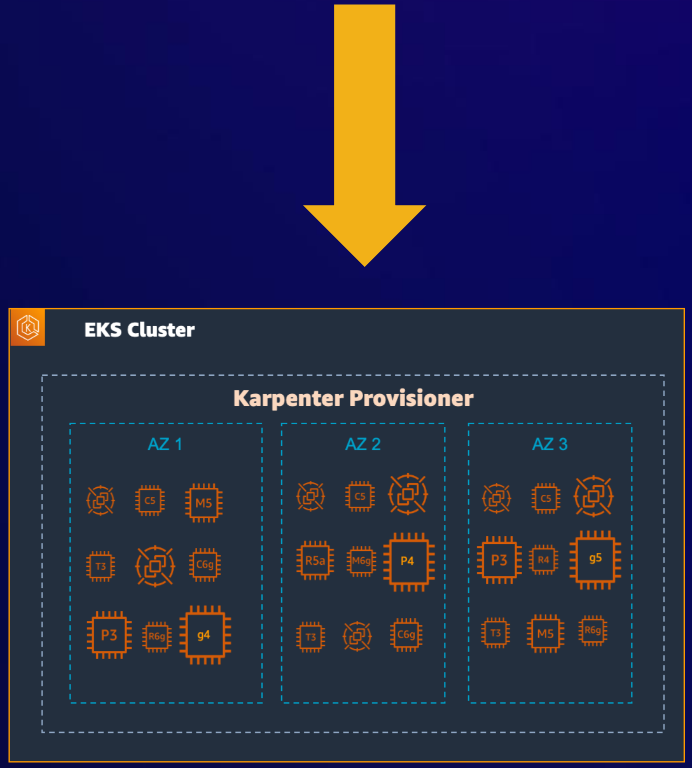
Generally available on AWS

Ready for production workloads since November 2021


AZ-1
AZ-2
AZ-3
Managed-nodegroup-on-demand
Managed-nodegroup-on-demand
Managed-nodegroup-on-demand
Managed-nodegroup-spot-2vCPUs-C
c5.large. c5d.large, c5a.large, c4.large, c3.large
Managed-nodegroup-spot-2vCPUs-C
c5.large. c5d.large, c5a.large, c4.large, c3.large
Managed-nodegroup-spot-2vCPUs-C
c5.large. c5d.large, c5a.large, c4.large, c3.large
Managed-nodegroup-spot-2vCPUs-M
m5.large. m5d.large, m5a.large, m4.large, m3.large
Managed-nodegroup-spot-2vCPUs-M
m5.large. m5d.large, m5a.large, m4.large, m3.large
Managed-nodegroup-spot-2vCPUs-M
m5.large. m5d.large, m5a.large, m4.large, m3.large
Managed-nodegroup-spot-2vCPUs-R
r5.large. r5d.large, r5a.large, r4.large, r3.large
Managed-nodegroup-spot-2vCPUs-R
r5.large. r5d.large, r5a.large, r4.large, r3.large
Managed-nodegroup-spot-2vCPUs-R
r5.large. r5d.large, r5a.large, r4.large, r3.large
Cluster 1
Cluster n

Best practice K8s node groups with CA

Mix of Amazon EC2 Spot & On-Demand



Karpenter right-sized groupless provisioned nodes

Mix of Amazon EC2 Spot & On-Demand

When to use

Cluster Autoscaler

- ✓ Homogeneous pods with well-defined compute needs
- ✓ Workloads need wider multicloud support

Karpenter

- ✓ Spiky heterogeneous workloads with changing compute needs
- ✓ Avoid complexity of managing many node groups
- ✓ Stateless workloads that need to be flexible to run on Amazon EC2 Spot

If you are starting with Amazon EKS, use Karpenter

Benefits of leveraging Karpenter

Operational efficiency

Node auto-provisioning

- ✓ Optimal compute infrastructure (Intel, Graviton, GPUs)
- ✓ Flexibility
  No need to configure every node group

Simplified resource management

- ✓ Easy upgrades
- ✓ Efficient management
  For many clusters and node groups

Cost optimization

Node consolidation

- ✓ Increases utilization Consolidates and bin packs pods to fewer node groups
- ✓ Lowers costs
  Consolidation from On-Demand → On-Demand,
  On-Demand → Amazon EC2 Spot

Application availability

Rapid scaling

- ✓ Faster
  Responds quickly and automatically to changes in resource requirements

Compute availability

- ✓ Mixes purchasing options as needed
  Prioritizes Amazon EC2 Spot, spins up On-Demand if Amazon EC2 Spot is not available

How Amazon EC2 Spot Instances work

Stateless, flexible, fault-tolerant workloads

Infrastructure
Amazon EC2 spare capacity

Provisioned from spare capacity, uses same infrastructure as On-Demand (OD)

Stable pricing
Up to 90% off compared to On-Demand

Not bidding: Pricing is based on long-term supply and demand, smooth and predictable

Interruption
Only interrupted if OD needs capacity

AWS can reclaim with 2-minute notice: Issues two types of notifications to help handle interruptions

Diversification
and flexibility are key

Make use of different instance types, sizes, Availability Zones, times, and Regions

How Karpenter provisions Amazon EC2 Spot

AMAZON EC2 SPOT DEPLOYMENT

Allocation strategy

Price capacity optimized

- Reduce the frequency of Amazon EC2 Spot terminations
- Reduce the cost of the instances

Diversify and don’t constrain

AWS Graviton

UP TO 40% BETTER PRICE PERFORMANCE FOR A BROAD RANGE OF WORKLOADS

BEST PRICE PERFORMANCE

Delivers up to 40% better price performance over comparable x86-based instances

EXTENSIVE ECOSYSTEM (Especially containers!)

Supported by popular Linux operating systems along with many popular application and services from AWS and ISVs

ENHANCED SECURITY

Provide key capabilities for application security, including always-on 256-bit DRAM encryption

GRAVITON2 INSTANCES

C6G(D)	M6G(D)	R6G(D)	C6GN	X2GD	T4G	IM4GN	IS4GEN	I4G
COMPUTE OPTIMIZED	GENERAL PURPOSE	MEMORY OPTIMIZED	ENHANCED NETWORKING	ENHANCED MEMORY	BURSTABLE	STORAGE OPTIMIZED	STORAGE OPTIMIZED	STORAGE OPTIMIZED

GRAVITON3 INSTANCES

C7G(D)	M7G(D)	R7G(D)
COMPUTE OPTIMIZED	GENERAL PURPOSE	MEMORY OPTIMIZED

Run these workloads on AWS Graviton

Web and gaming servers

Open source databases

High-performance computing

In-memory caches

Media encoding

Electronic design automation

Analytics

Microservices

> “The simplicity with which instances are assigned to nodes within the cluster, and how easily you can discard nodes to scale down and save costs, are the main advantages of Karpenter”

Sagar Arora
Senior Principal Engineer DevOps
Airtel Digital Limited, 2021–2023

airtel

INDUSTRY
Telecommunications

COUNTRY
India

Bharti Airtel Limited (Airtel) is a leading telecommunications company headquartered in India. Its Wynk Music over-the-top (OTT) streaming platform offers 13.5 billion minutes of content delivered to 90 million active subscribers. Airtel launched Wynk Studio in June 2022 as a digital platform for independent music and podcasts.

Airtel Delivers a Smooth Streaming Experience for Wynk Studio by Adopting Karpenter on Amazon EKS

Business needs

Airtel was experiencing highly variable workloads for its internal applications on Amazon EKS, with traffic multiplying two to four times in seconds. Costs were rapidly increasing and the open source Kubernetes Cluster Autoscaler Airtel adopted was taking approximately 2 minutes to bring up new nodes when workloads spiked. This led to sporadic delays during content playback on the Wynk platform.

Solutions

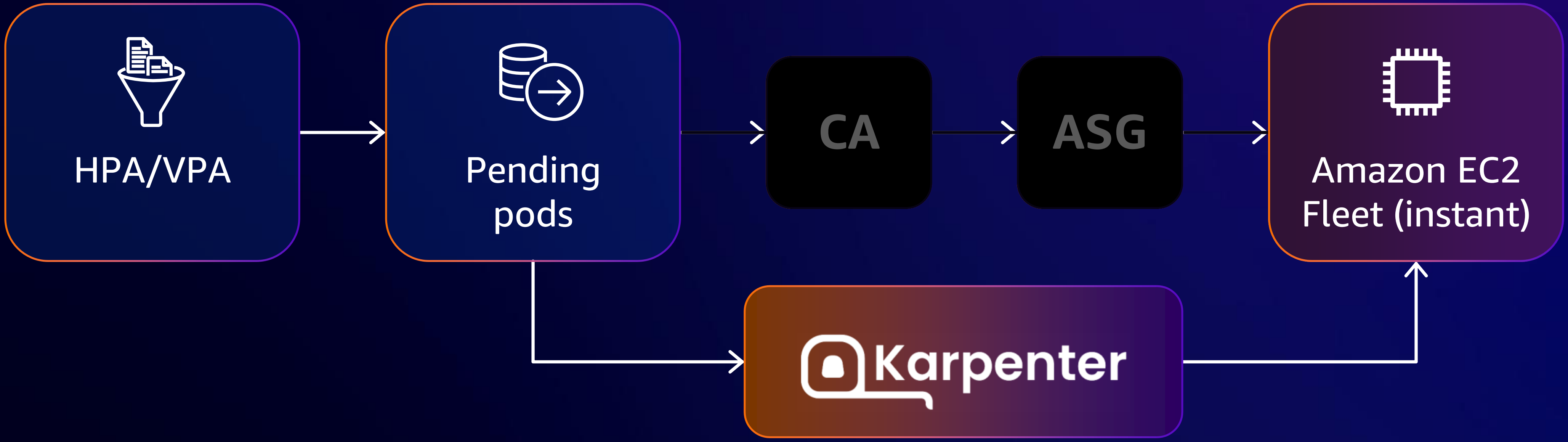
- Amazon **EC2 Spot** Instances
- **Karpenter**
- Amazon Elastic Kubernetes Service

Benefits

- Reduces infrastructure costs by 47%
- Lowers time to scale up from 2 minutes to 40 seconds
- Lowers time to scale down from 1 minute to 20 seconds
- Automates node deployment in Kubernetes clusters
- Maintains application availability with automatic failover
- Saves 1 to 2 days for provisioning and upgrading instances

aws

How Karpenter works

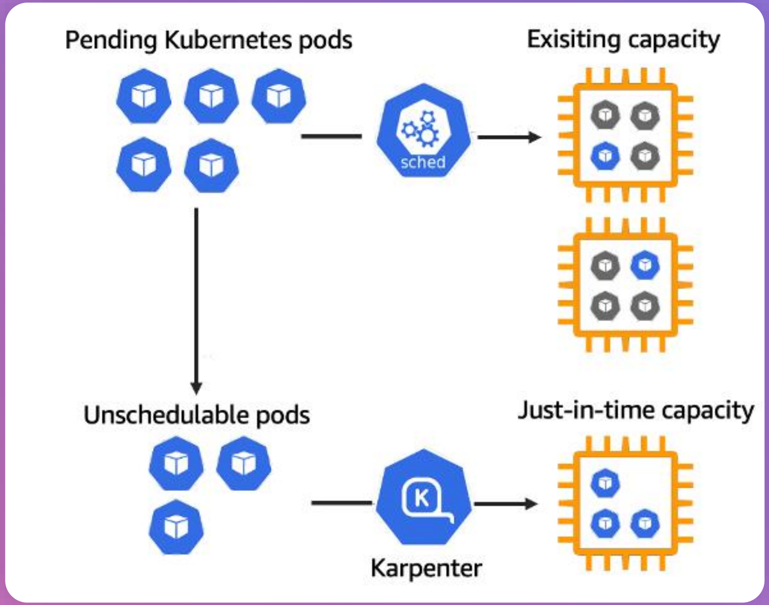


Consolidates instance orchestration responsibilities within a single system

Karpenter

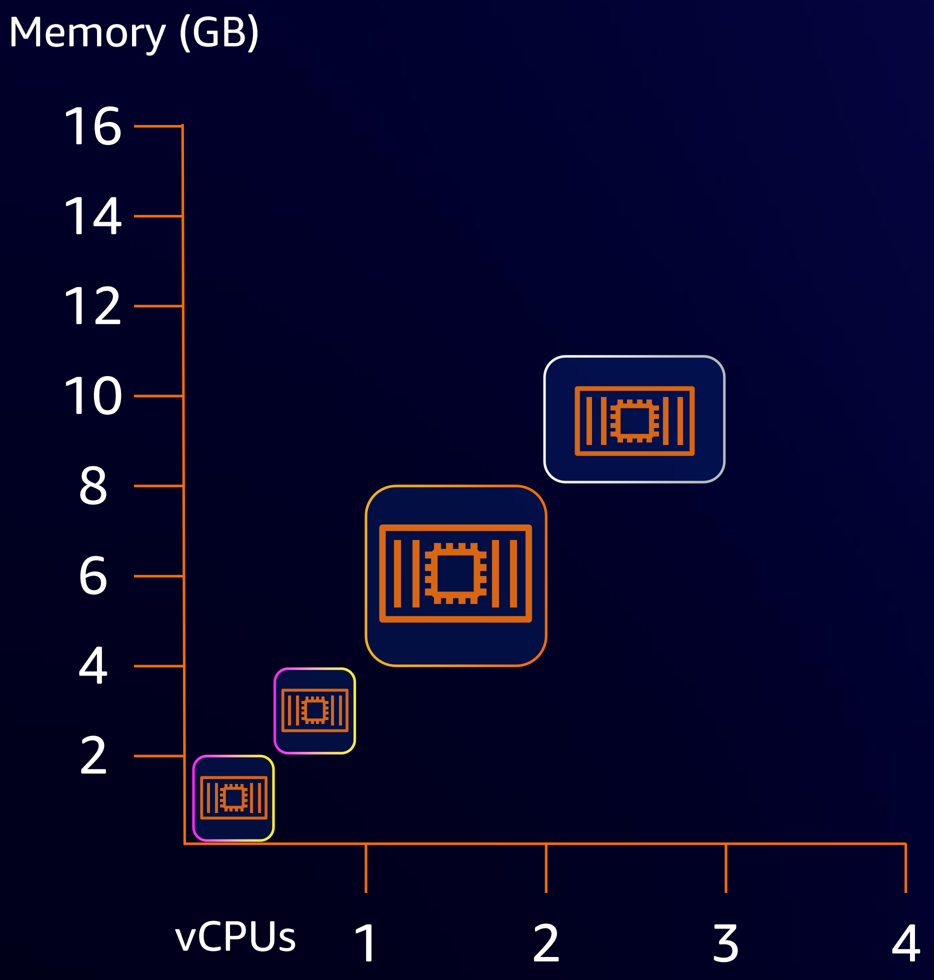
NODE PROVISIONING

- Kube-scheduler gets the first crack at scheduling pending pods – tries to schedule on existing capacity
- Karpenter observes aggregate resource requests of unschedulable pods set by kube-scheduler) to make decisions on what instances to launch

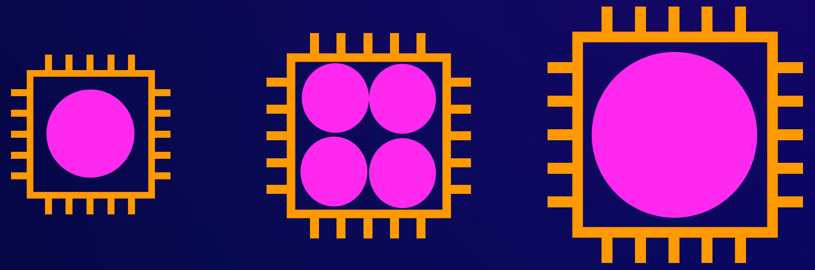


Karpenter

BINPACKING



Online binpacking



Well-known labels

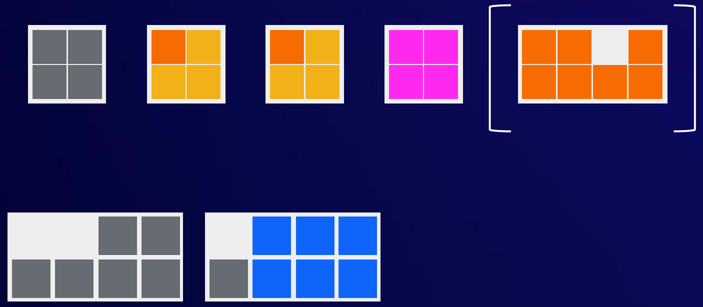
- karpenter.sh/capacity-type=spot
- karpenter.k8s.aws/instance-family=m6i
- kubernetes.io/arch=arm64
- topology.Kubernetes.io/zone=us-west-2a

Karpenter scale-in

HPA << Pending pods

Karpenter

Consolidation actively seeks out opportunities to make the cluster more cost efficient



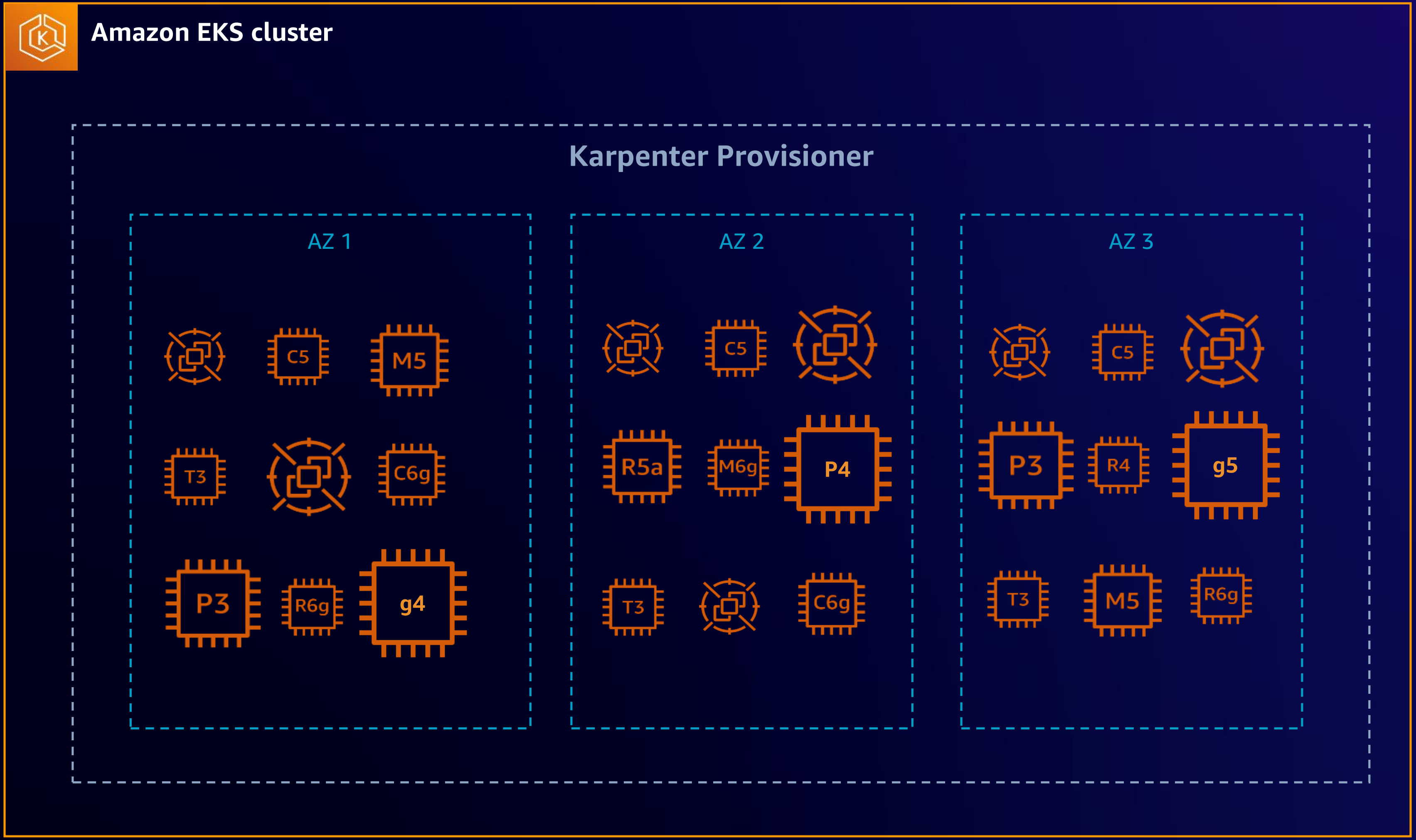
Terminations

- Replace underutilized nodes with more efficient compute
- Node Expiration TTL
- kubectl delete node with graceful draining

⬜ Node

🟨 1 vCPU request

Compute flexibility with Karpenter



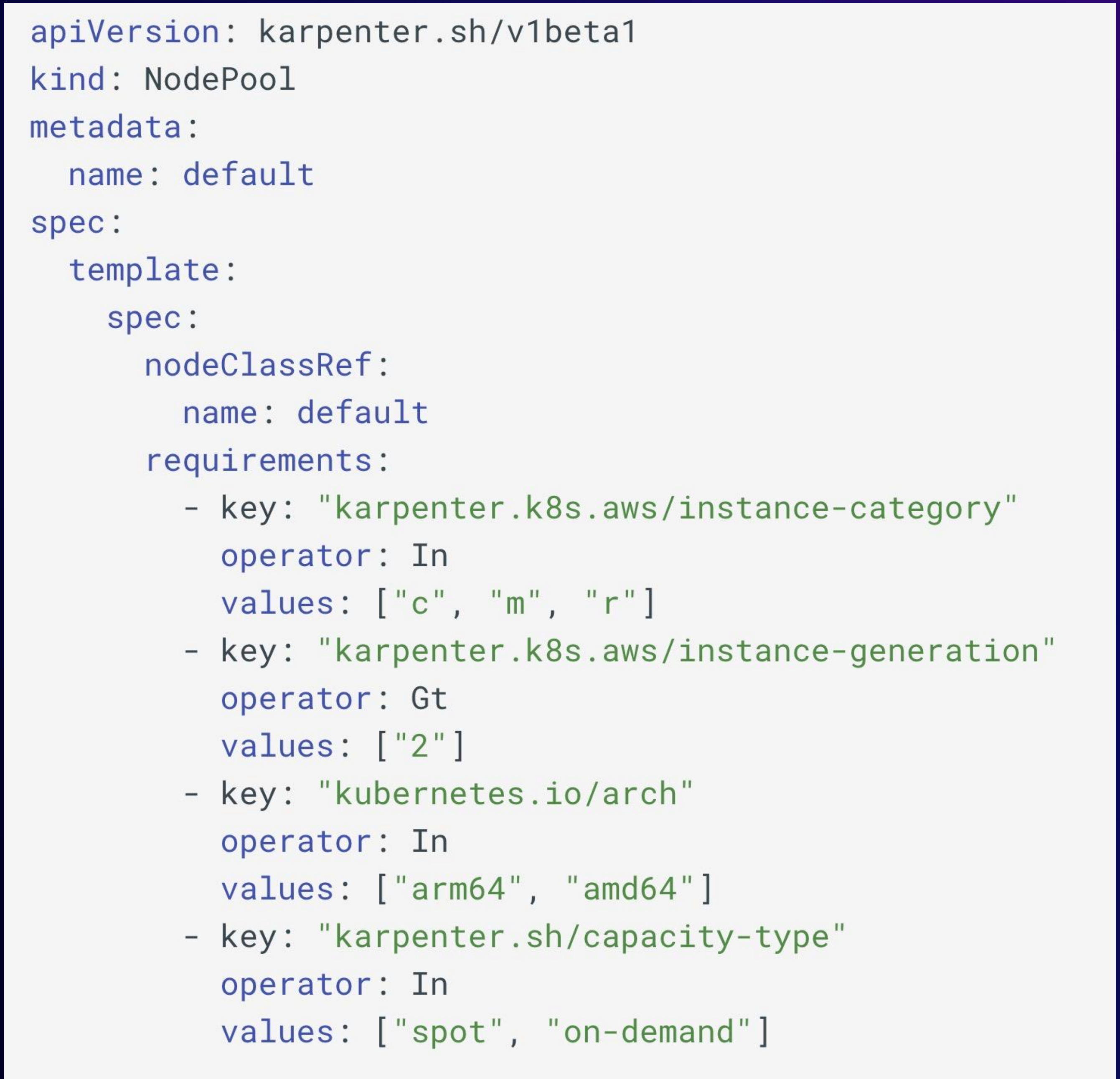
Recent changes on Karpenter

v1beta1 launched in October 2023 will introduce more streamlined APIs

Custom Resources will reference “Node” to better align with upstream concepts:

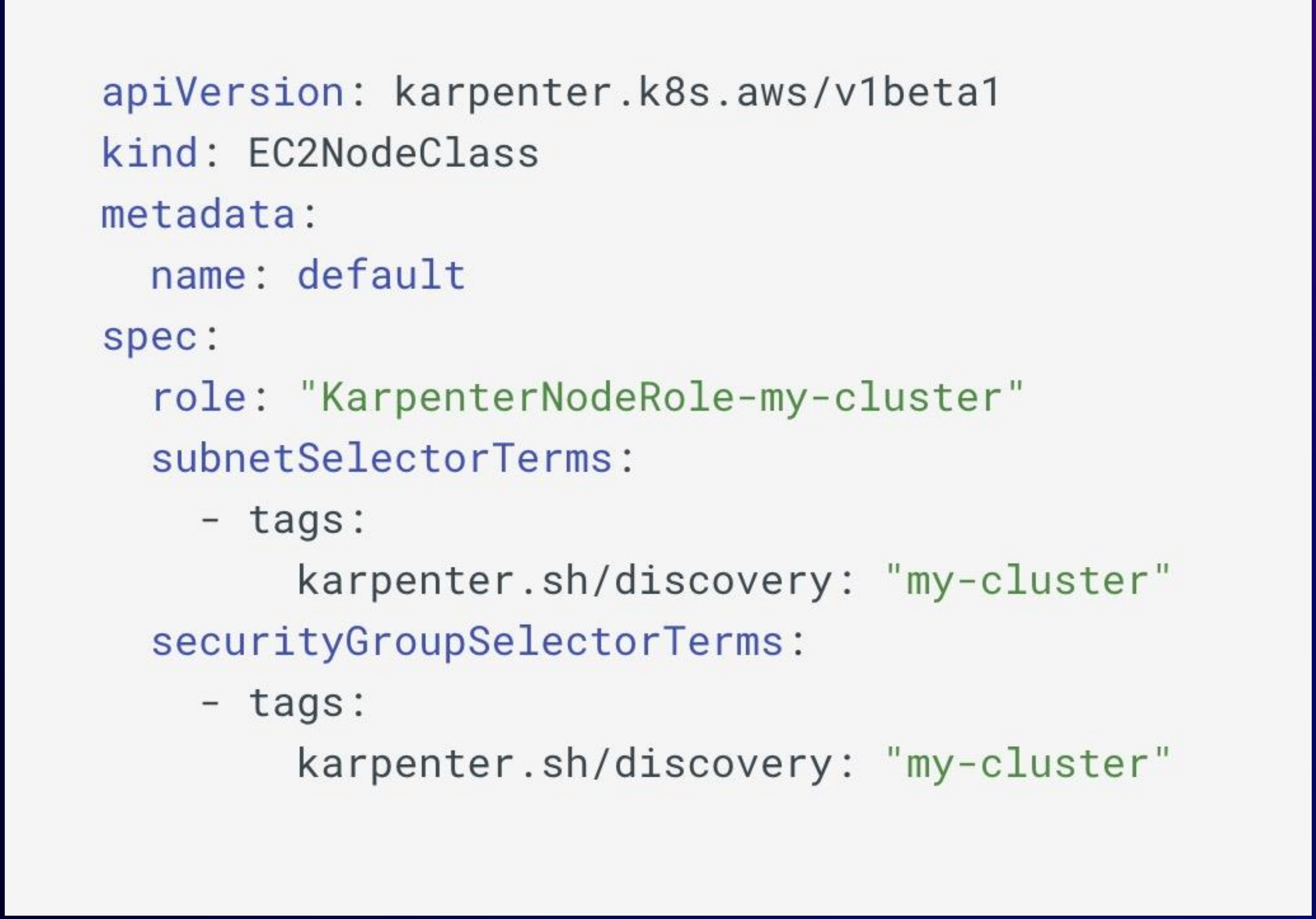
- Provisioner → NodePool
- AWSNodeTemplate → EC2NodeClass
- Machine → NodeClaim

NodePool



```yaml
apiVersion: karpenter.sh/v1beta1
kind: NodePool
metadata:
  name: default
spec:
  template:
    spec:
      nodeClassRef:
        name: default
      requirements:
        - key: "karpenter.k8s.aws/instance-category"
          operator: In
          values: ["c", "m", "r"]
        - key: "karpenter.k8s.aws/instance-generation"
          operator: Gt
          values: ["2"]
        - key: "kubernetes.io/arch"
          operator: In
          values: ["arm64", "amd64"]
        - key: "karpenter.sh/capacity-type"
          operator: In
          values: ["spot", "on-demand"]
```

EC2NodeClass



```yaml
apiVersion: karpenter.k8s.aws/v1beta1
kind: EC2NodeClass
metadata:
  name: default
spec:
  role: "KarpenterNodeRole-my-cluster"
  subnetSelectorTerms:
    - tags:
        karpenter.sh/discovery: "my-cluster"
  securityGroupSelectorTerms:
    - tags:
        karpenter.sh/discovery: "my-cluster"
```

Key takeaways

- Leverage Karpenter for **operational efficiency**, **cost optimization,** and **application availability**
- Scale **large, heterogenous** Amazon EKS environments with many clusters with Karpenter
- Take advantage of Graviton for price/performance benefits for multi-arch workloads with Karpenter
- Seamlessly leverage Amazon EC2 Spot with flexibility that Karpenter provides

Demo

Additional resources

Amazon EKS + Amazon EC2 Spot + Karpenter workshop/blog

https://ec2spotworkshops.com/karpenter.html

https://aws.amazon.com/blogs/containers/using-amazon-ec2-spot-instances-with-karpenter/

Migration from Cluster Autoscaler to Karpenter

https://karpenter.sh/preview/getting-started/migrating-from-cas/

Latest Changes on Karpenter

https://karpenter.sh/docs/upgrading/v1beta1-migration/

Continue your Compute learning

Learn at your own pace

Expand your knowledge with Learning Plans via **AWS Skill Builder**

Increase your knowledge

Use our **Ramp-Up Guides** to build your Compute knowledge

Earn your Compute badge

Demonstrate your knowledge by achieving **digital badges**

Thank you!

☆☆☆☆☆ Please complete the session survey in the mobile app

Aparajita Jas
apajas@amazon.com
linkedin.com/in/aparajitajas

Brandon Wagner
wagnerbm@amazon.com
linkedin.com/in/brandonmarkwagner