AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

CON210

Getting to large Amazon EKS clusters

Alex Kestner (he/him)
Senior Product Manager, Amazon EKS
AWS

Shyam Jeedigunta (he/him)
Senior Software Development Engineer, Amazon EKS
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

- What is Amazon EKS?
- Kubernetes scaling considerations
- Scaling the Amazon EKS control plane
- Scaling Kubernetes networking
- Scaling cluster worker nodes
- Cluster autoscaling with Karpenter

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS is the best place to run Kubernetes. 65% of organizations choose AWS to run their containers.



CNCF State of Cloud Native Development

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Why Amazon EKS?



Running and scaling Kubernetes can be difficult and requires significant investment



Securing Kubernetes increases the operational overhead of running applications



Applications need a native way to integrate with other AWS services securely and reliably

All of this extra work was leading to a lot of undifferentiated heavy lifting

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is Amazon EKS?



Amazon EKS

- Amazon EKS runs vanilla Kubernetes; EKS is upstream and a certified conformant version of Kubernetes (with backported security fixes)
- Amazon EKS supports 4 versions of Kubernetes, giving you 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

Amazon EKS helps you build reliable, stable, and secure applications in virtually any environment

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What does “elastic” Kubernetes mean?

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What does “elastic“ Kubernetes mean?

- Portability across environments
- Extensible to use cases

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS is portable and extensible



Amazon EKS in AWS



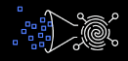
Web applications



Mobile applications



Gaming



Data processing



Amazon EKS and AWS Outposts



Manufacturing



Retail



Gaming



Low latency



Telecom



Amazon EKS and AWS Local Zones



Low latency



Video rendering



Telecom



Gaming



Amazon EKS Anywhere



Data center



Manufacturing



Telecom



Gaming



Amazon EKS Distro



IoT



Self-hosted K8s

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What does “elastic“ Kubernetes mean?

- Portability across environments
- Extensible to use cases
- **Scalable and performant**

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kubernetes scaling considerations

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How is Kubernetes “scalability” defined?

Scalability Envelope

KubeCon | CloudNativeCon
North America 2018

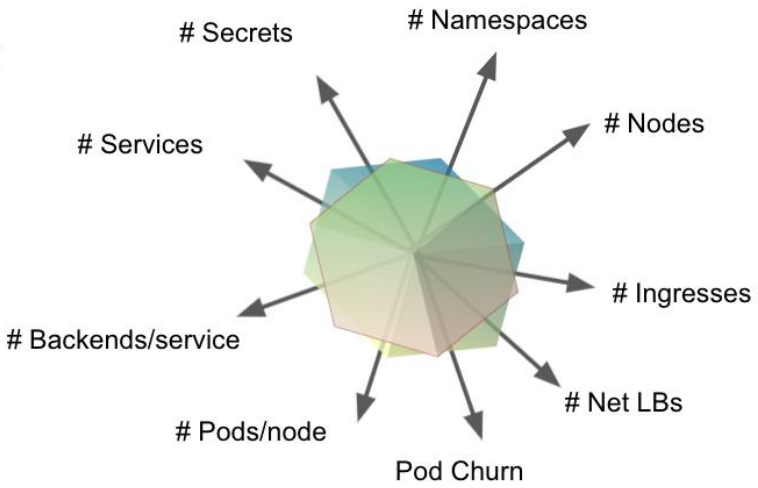
Scalability is a **subspace of configurations**

Think of it as a ~ higher-dimensional cube (not really a cube… see next slide)

If you’re within the envelope, you’re *safe*

By *safe*, we mean:

- Performance SLOs are satisfied
- Your k8s cluster is not badly broken



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

We've been listening . . .

Control plane

Pre-warming for large-scale events

API timeouts/unresponsive

Too many API calls and throttling

etcd write throughput

Job/deployment throughput

Pod scheduling throughput

Add-ons

Fluentd API call volume

DNS request latencies

Cluster auto-scaling latency

Networking

Amazon EC2 network API throttling

IP addresses available per node

Load balancer scaling

Worker nodes

Node startup time

Node launch throughput

Pod startup latency

Pod density

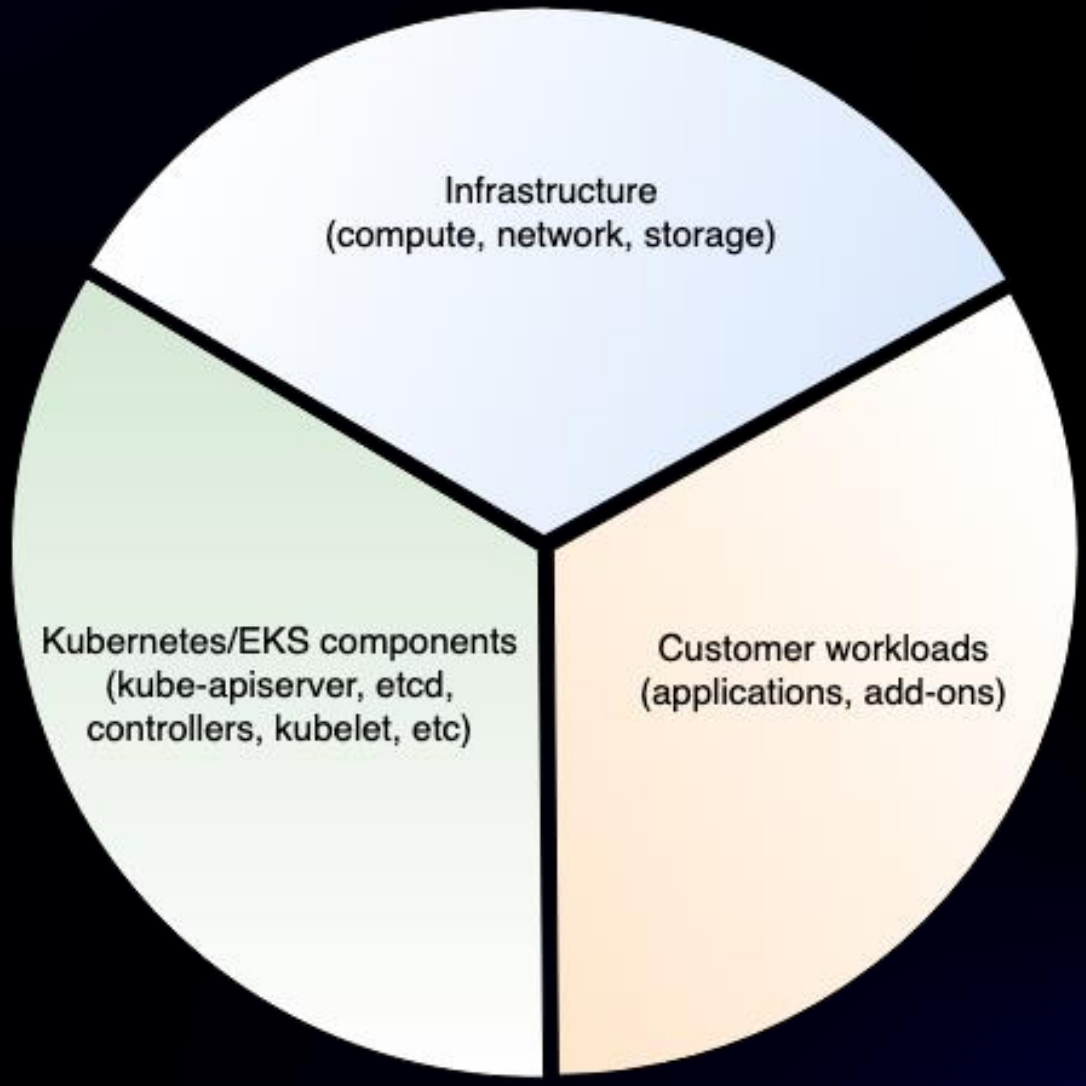
Upgrade time

Insufficient Amazon EC2 capacity

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What goes into a scalable Amazon EKS cluster?

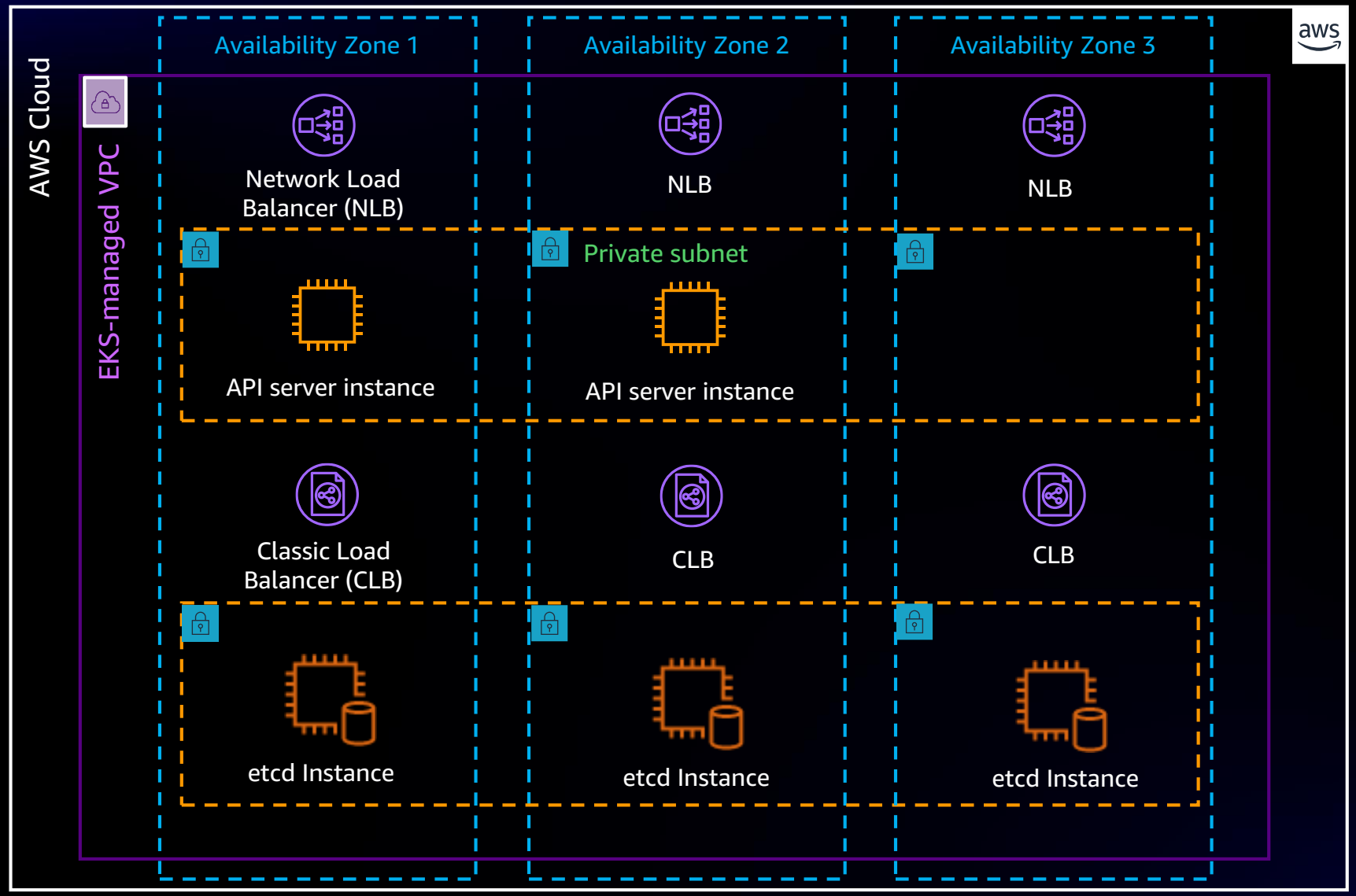


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scaling the control plane

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

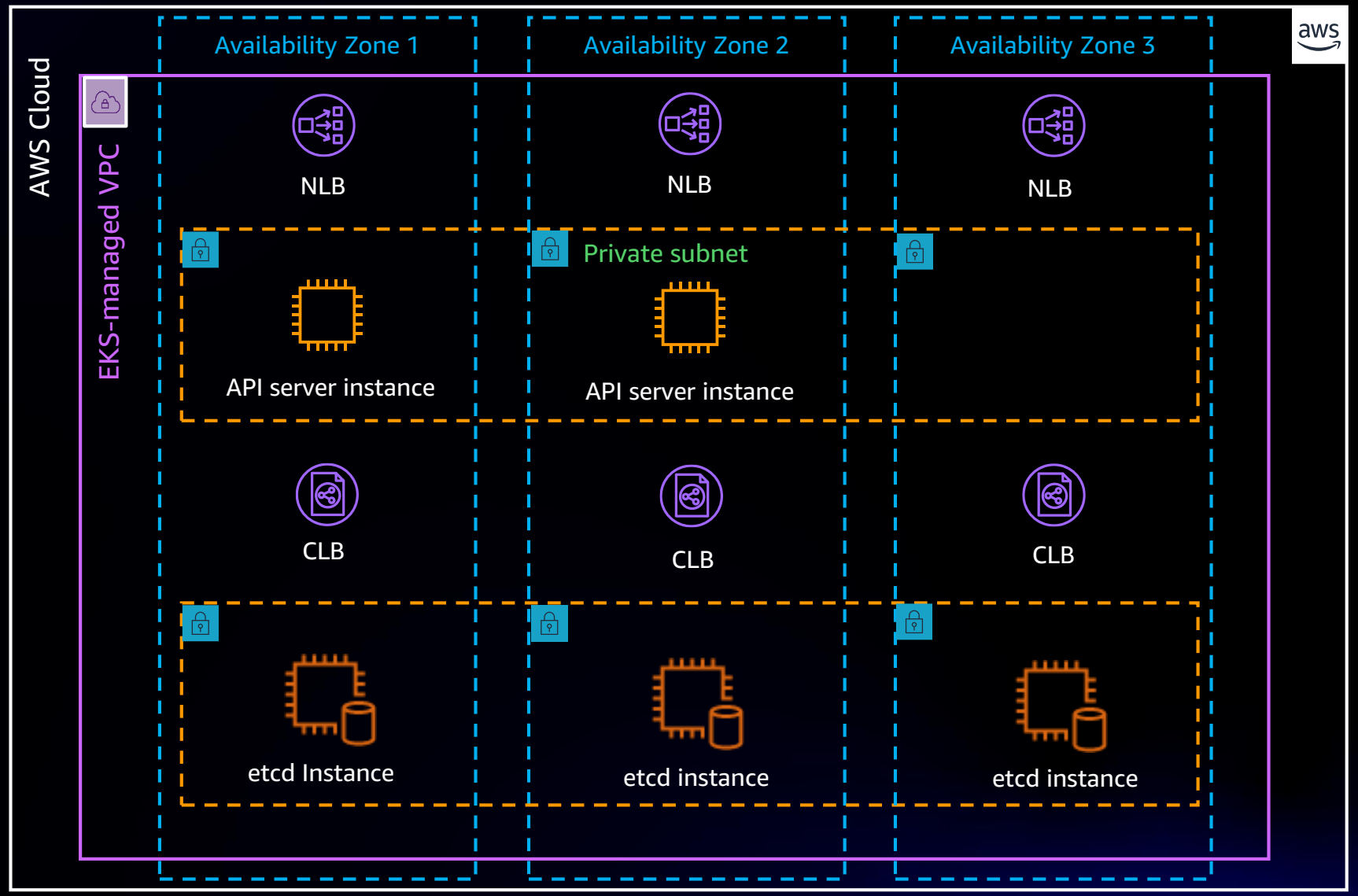
Amazon EKS managed control plane



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How does EKS ensure high availability?

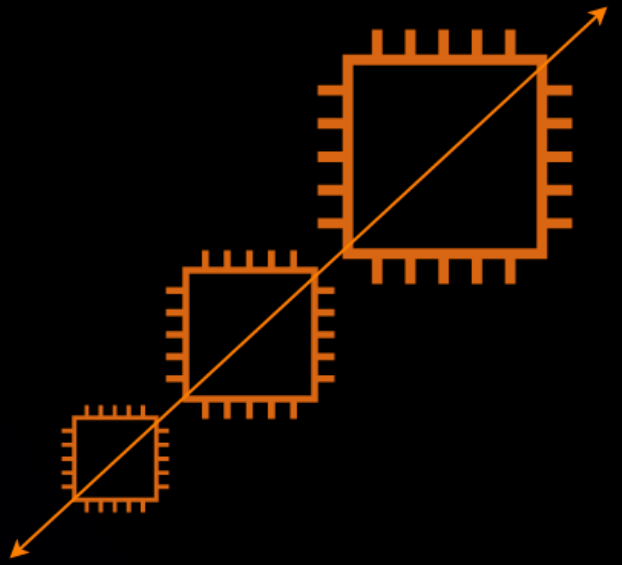
- Survive single-AZ events
- Highly available cluster endpoint
- 99.95% SLA
- 24x7x365 support



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Control plane instance right-sizing





Instances scaled up/down seamlessly

Using signals such as

- CPU usage
- Memory usage
- Cluster size

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Control plane networking



CLB load balances the traffic from the API servers to the etcd nodes

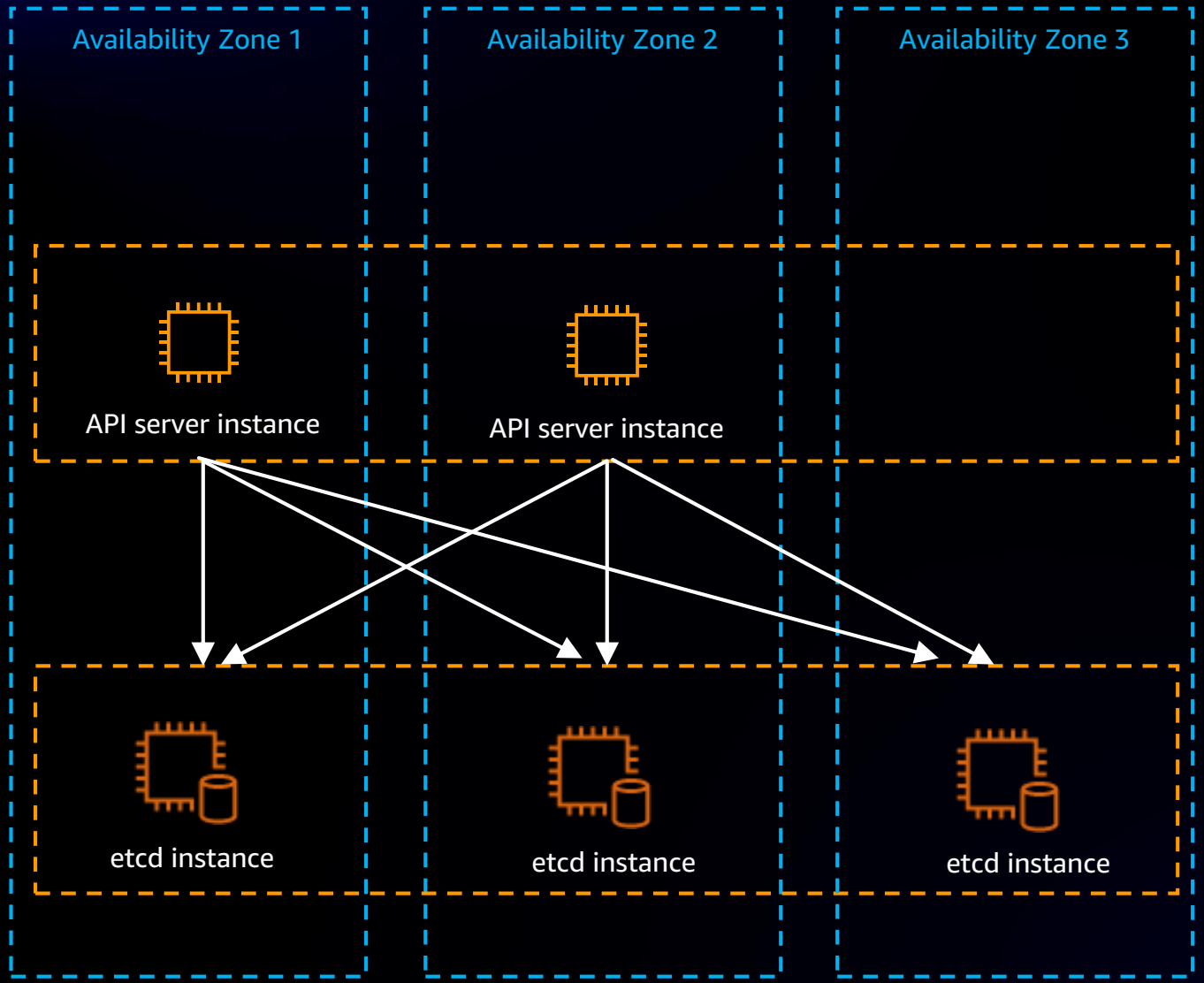
This means

- Extra hop (and latency)
- CLB needs to scale
- Additional dependency

So…

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Control plane networking



We got rid of the CLB

And switched to client-side load balancing

This gives

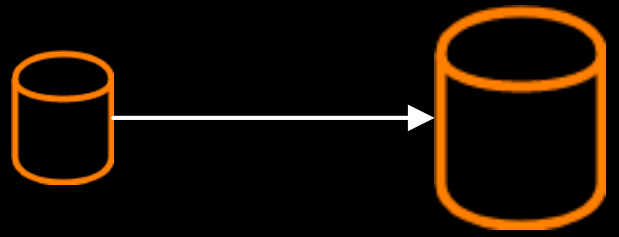
- ENI-to-ENI performance
- Client-level fault tolerance

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

etcd volume performance



We increased the volume throughput 6x



500 IOPS 3,000 IOPS

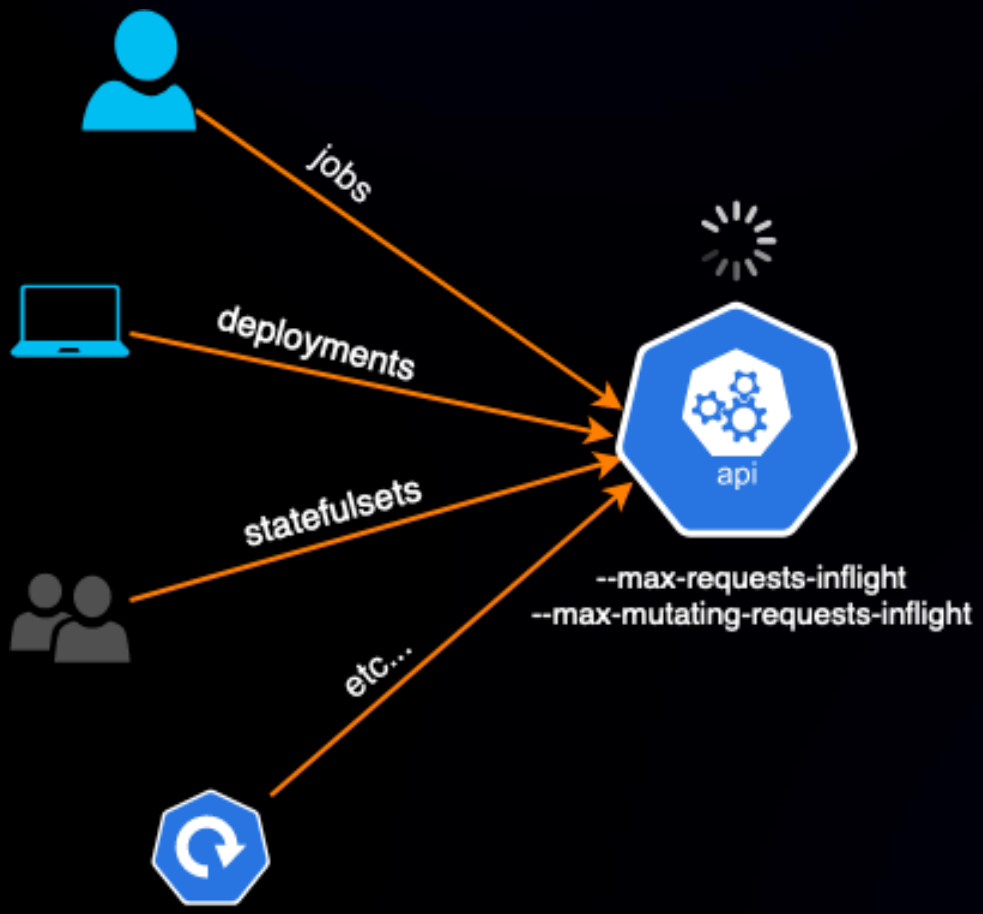
You get
- Higher write throughput for API calls

While continuing to have
- Single-digit ms latencies
- Read API calls served from memory
- Automated etcd snapshots

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Auto-tuning control plane throughput

Coming Soon



We increase the allowed throughput of API calls seamlessly, as your workload grows

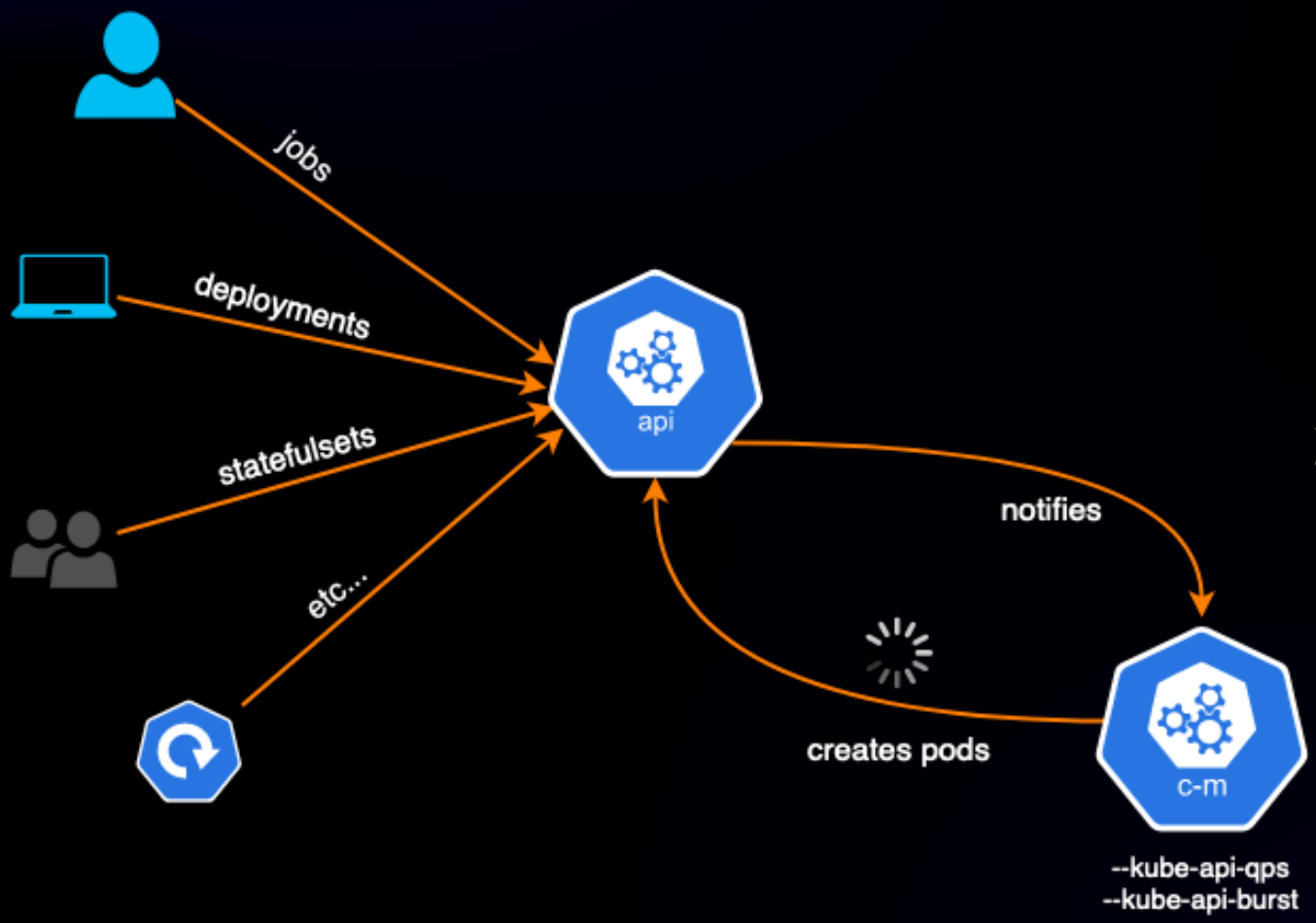
While continuing to meet the API call latency service level objectives (SLO) set by the Kubernetes community

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Auto-tuning control plane throughput

Coming Soon

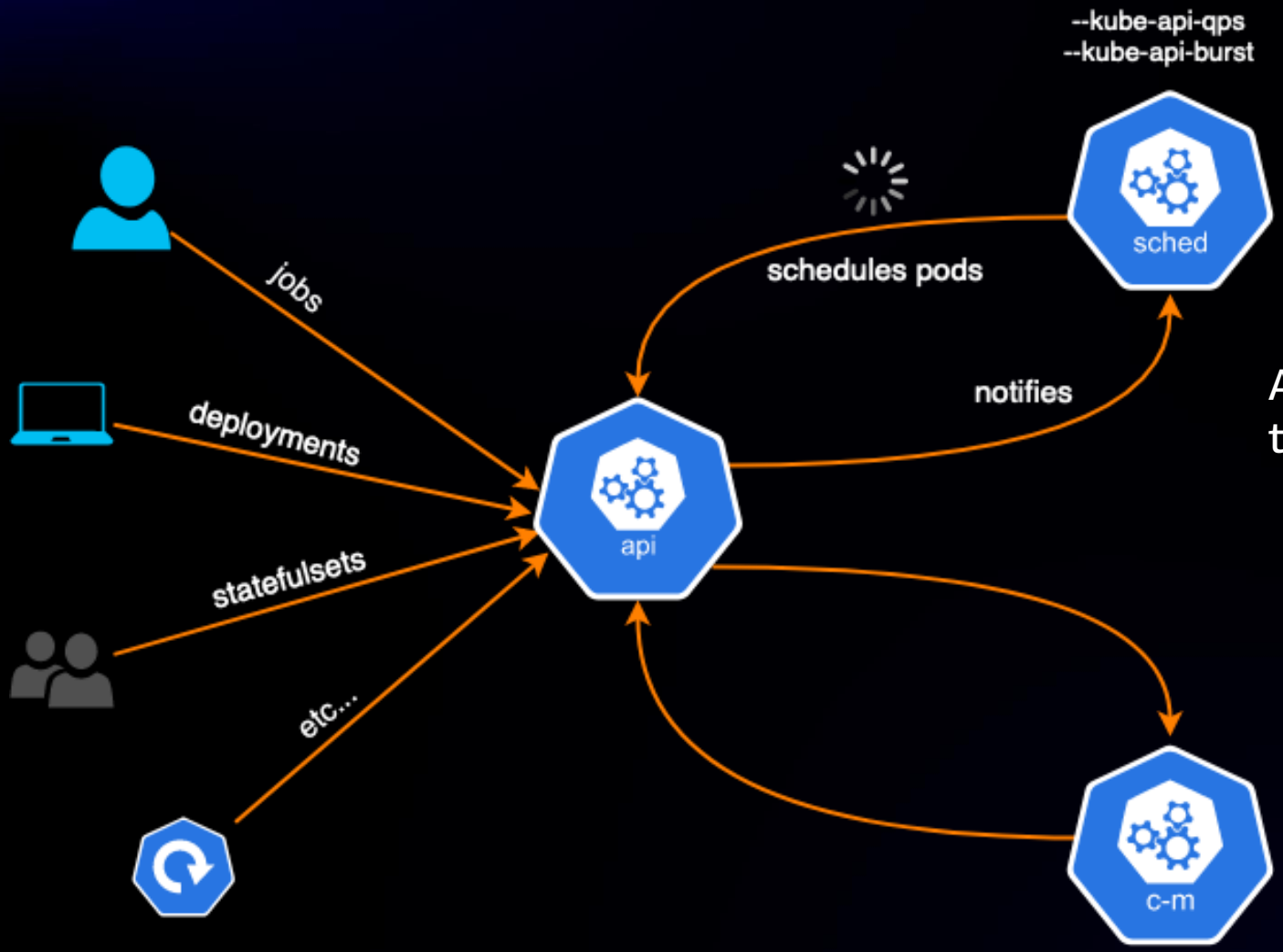


And the throughput of your kube-controllers, to allow for higher pod creation/deletion rates

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Auto-tuning control plane throughput

Coming Soon

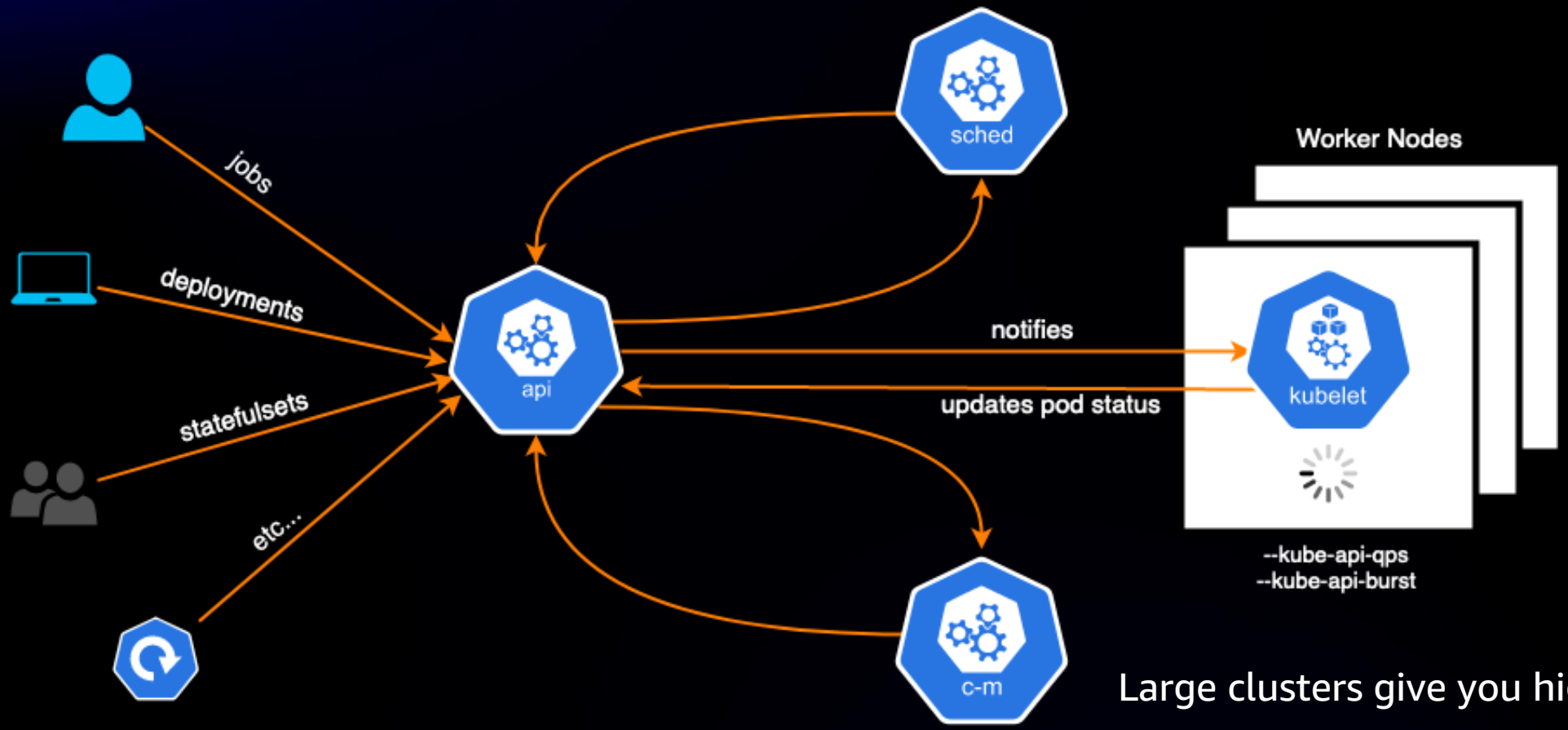


And the throughput of your kube-scheduler, to allow for higher pod scheduling rates

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tuning the kubelet throughput

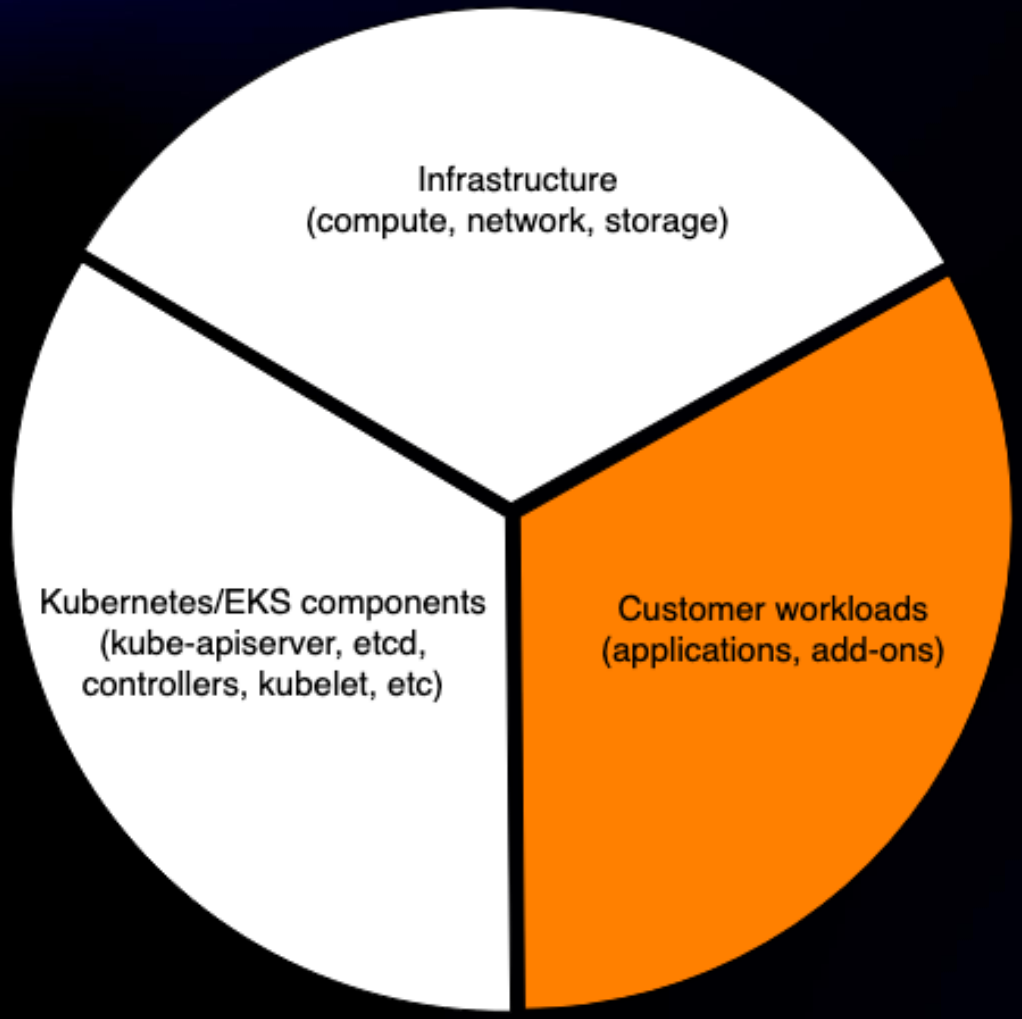


Large clusters give you high pod throughput

To customize per-node throughput, leverage custom launch templates for your nodes

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What goes into a scalable EKS cluster?



Shared responsibility model

EKS shares best practices and guidance

Session	Title
CON316-R2	High performance with Amazon EKS and hundreds of lessons learned

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scaling fluent-bit on large clusters

Top ten API talkers for "LIST" calls

Showing 8 of 13,491 records matched

Hide histogram

537,535 records (742.0 MB) scanned in 4.1s @ 132,430 records/s (182.8 MB/s)



#	userAgent	count
1	rest-client/2.1.0 (linux-gnu x86_64) ruby/2.6.7p197	13283
2	kubelet/v1.19.6 (linux/amd64) kubernetes/49a6c0b	137
3	controller/v0.0.0 (linux/amd64) kubernetes/$Format	42
4	kube-scheduler/v1.19.8 (linux/amd64) kubernetes/96780…	16
5	kube-apiserver/v1.19.8 (linux/amd64) kubernetes/96780…	4
6	cluster-event-watcher/v0.0.0 (linux/amd64) kubernetes…	4
7	kube-controller-manager/v1.19.8 (linux/amd64) kuberne…	4
8	metrics-server/v0.0.0 (linux/amd64) kubernetes/$Format	1

As we dug deeper into these requests, we found that Fluentd (**rest-client/2.1.0**) overwhelms the Kubernetes API server with thousands of GET (~ 36,400) and LIST (~27,800) requests without any production workloads running in the cluster as seen below:

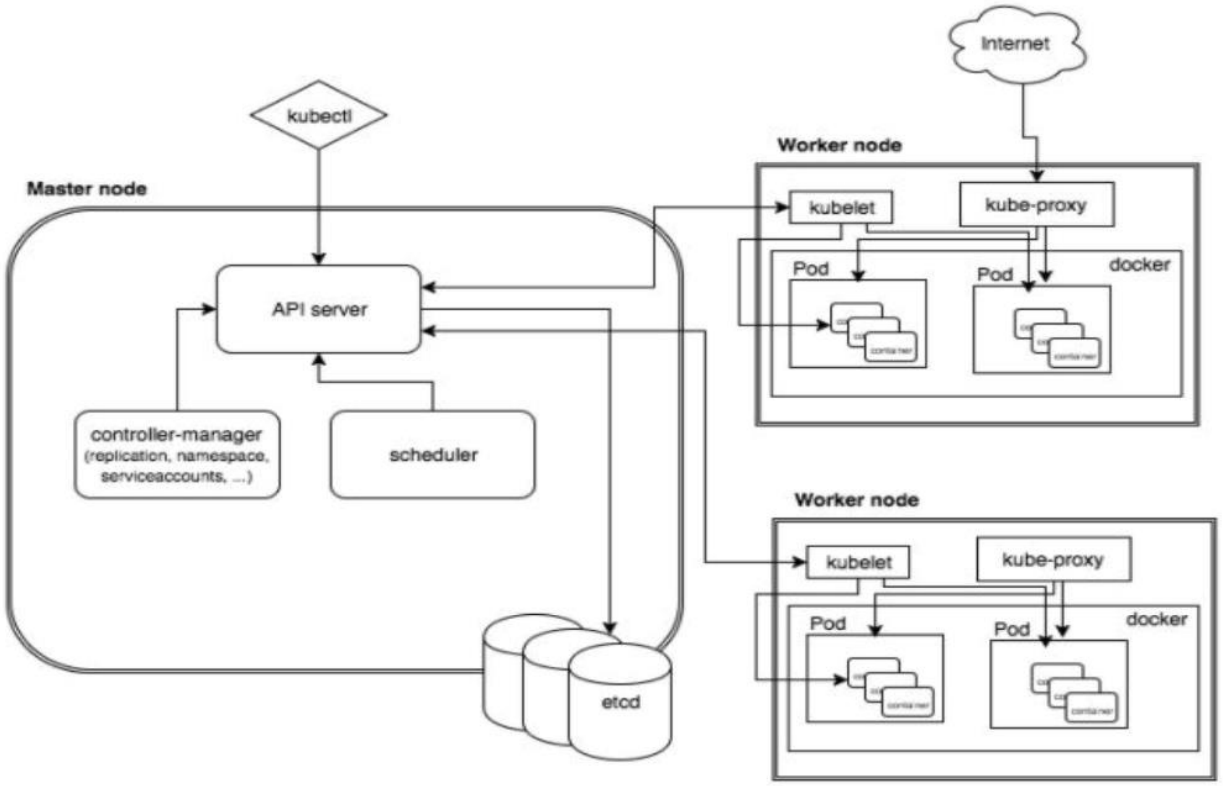
https://aws.amazon.com/blogs/containers/fluentd-considerations-and-actions-required-at-scale-in-amazon-eks/

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scaling fluent-bit on large clusters

Solution

fluentcon
Cloud Native Logging day with Fluent Bit and Fluentd



Fluent bit/Fluentd just need the pods information for logs.

Use kubelet /pods endpoint to get the pods information instead of calling API Server

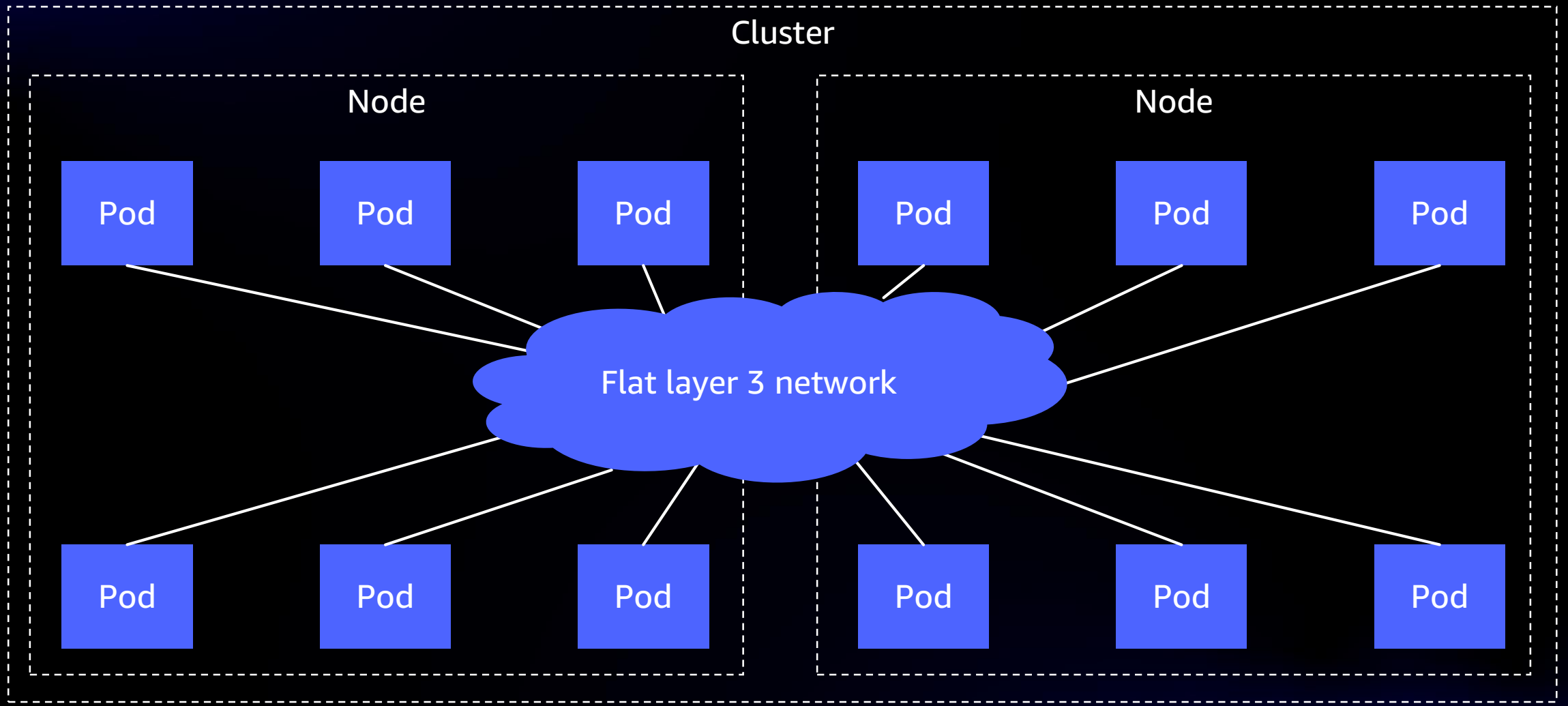
Kubelet is in every worker node, so the requests to API server will reduce to 0 and request increase for each kubelet and node is 1.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scaling cluster networking

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

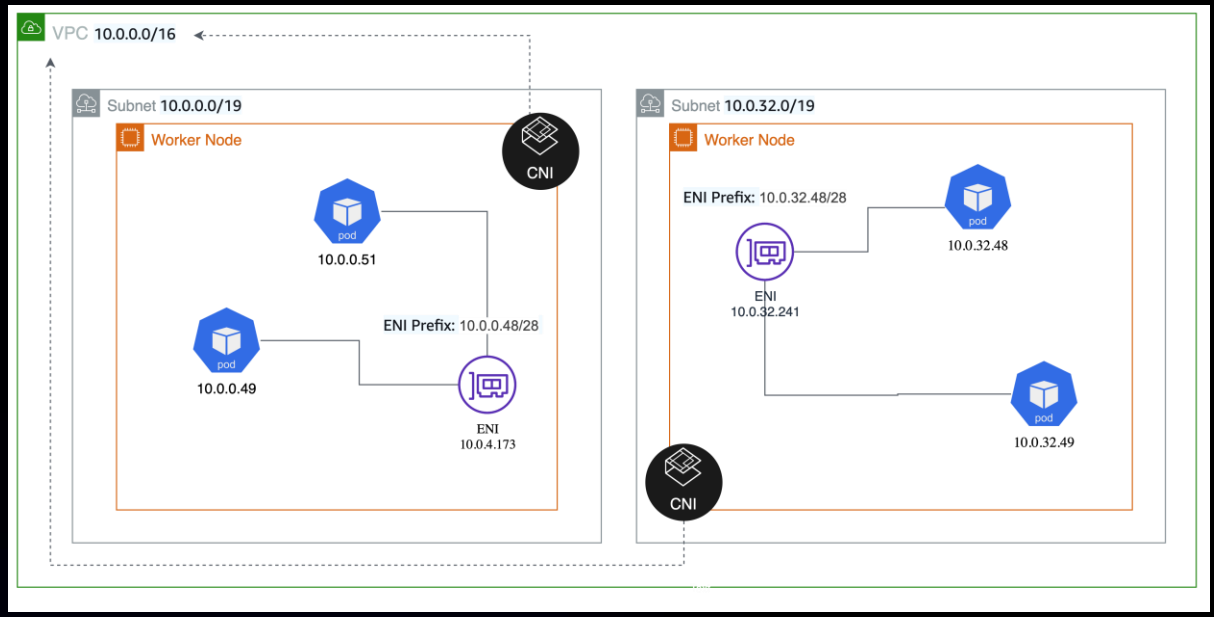
Kubernetes networking model



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

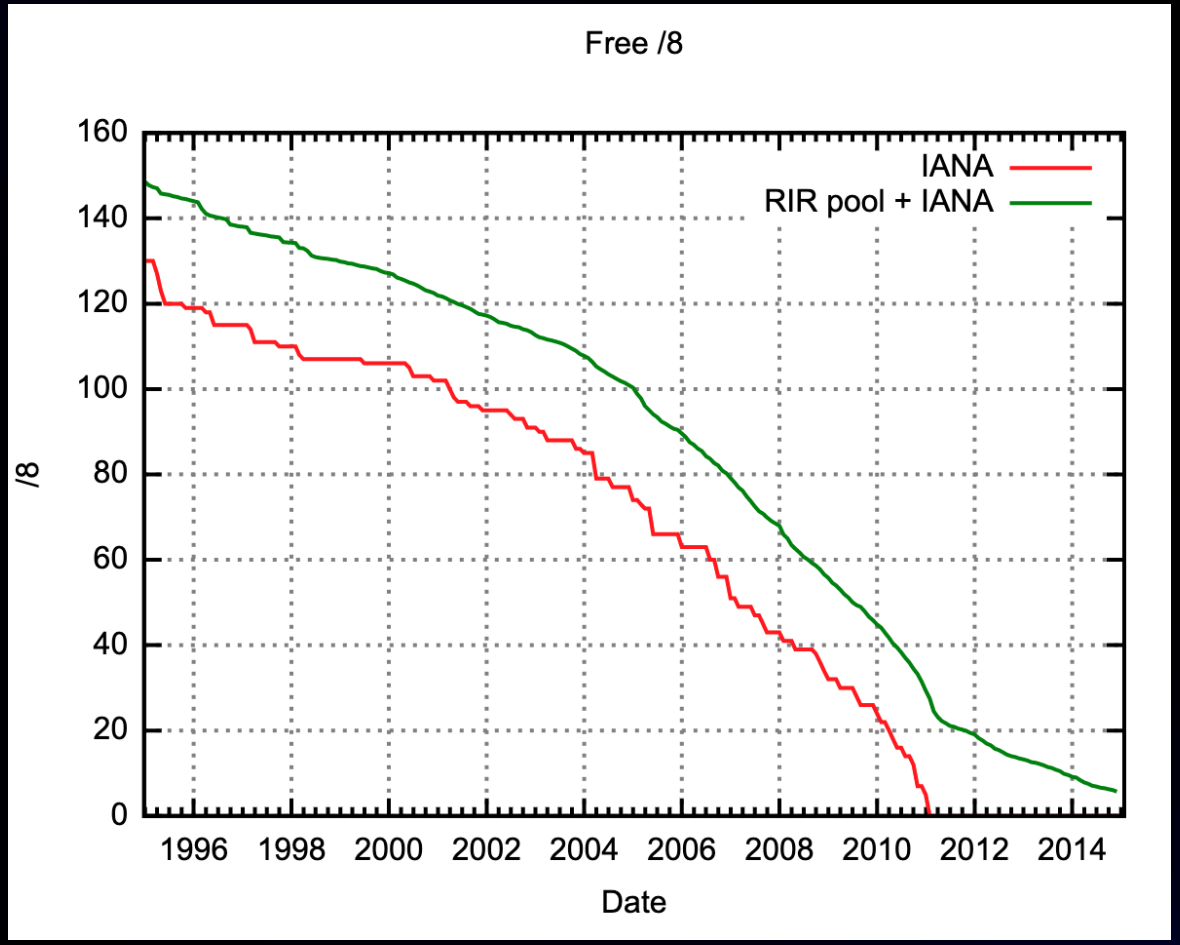
VPC CNI increased pod density

- VPC CNI 1.9 uses *EC2 Prefix Delegation (PD)* feature
- PD enables multiple /28 IP address prefixes to network interfaces on worker nodes
- Enables more pods per node
- Reduces number of calls to Amazon EC2 APIs



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Global shortage of public IPv4 addresses



Source: https://commons.wikimedia.org/wiki/File:Ipv4-exhaust.svg
Mro, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

IPv6 support

Coming Soon



Scale far beyond IPv4 limits with globally unique IPv6 address assigned per pod

Simplified pod-to-internet routing without network address translation

Faster pod launch times

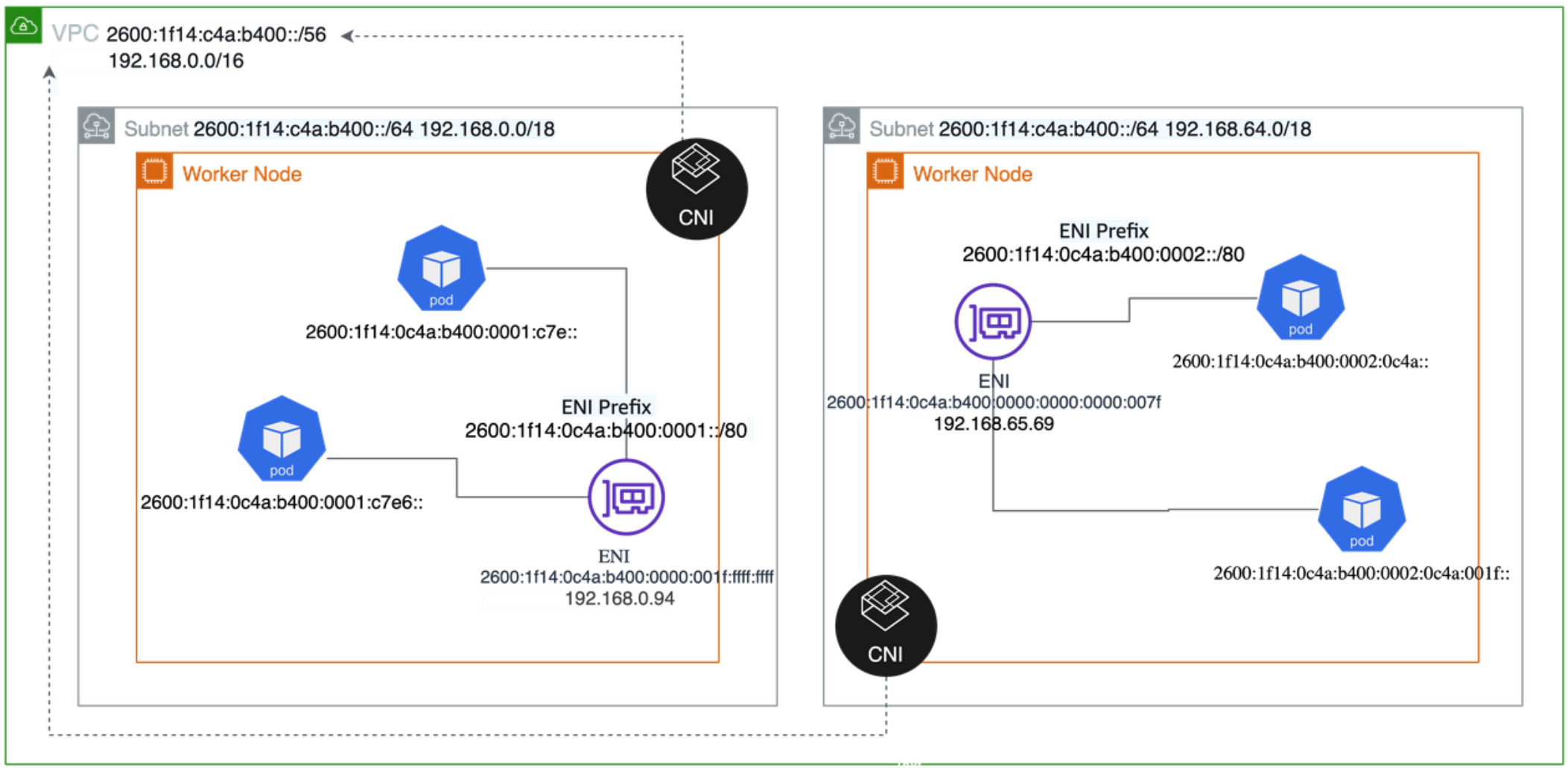
Egress IPv4 traffic support gives you best of both worlds – move to IPv6 on Amazon EKS before the rest of your org supports it

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS IPv6 networking

Coming Soon

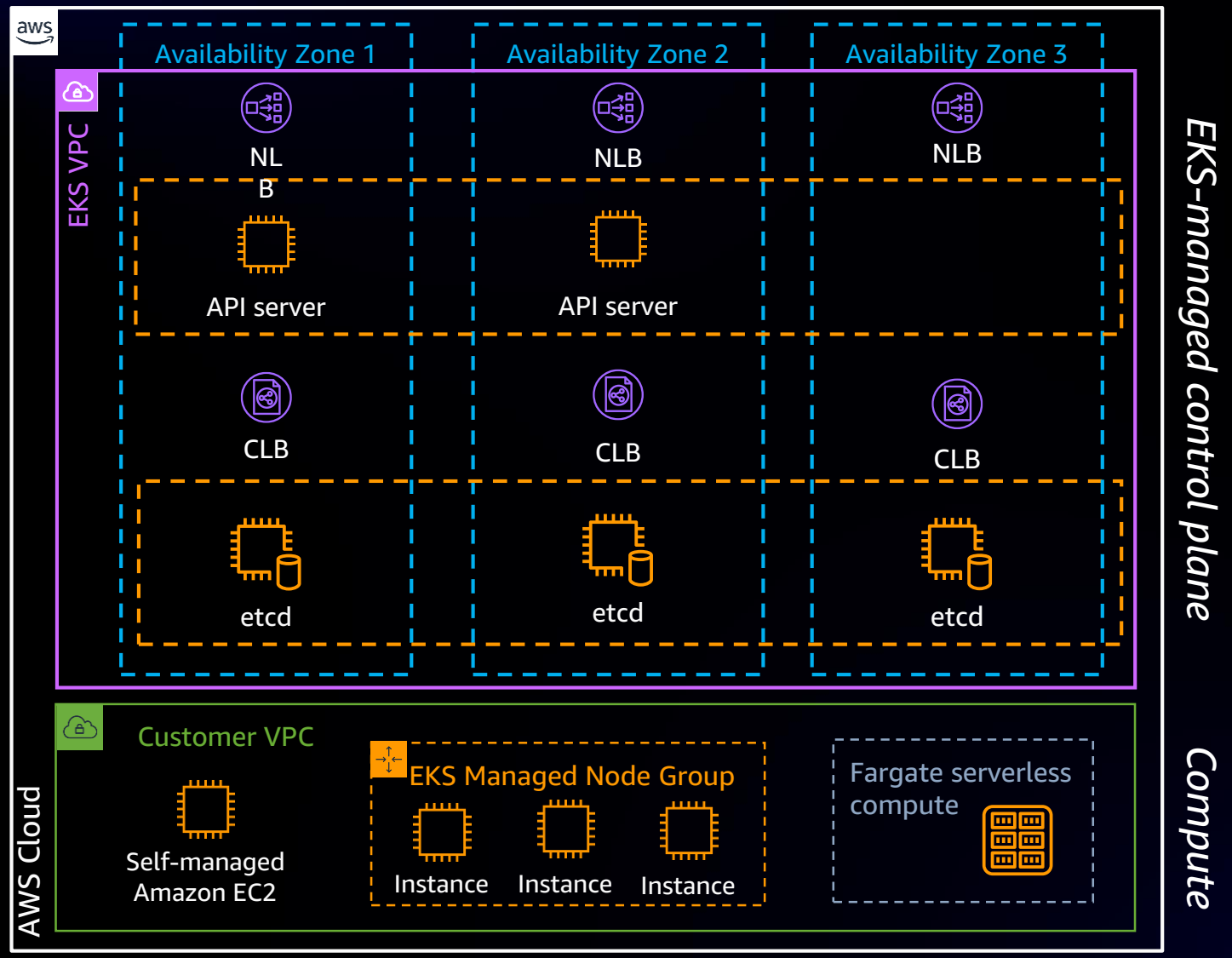


© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Scaling cluster worker nodes

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS worker compute



EKS compute options

Self-managed Amazon EC2 instances

- Run in your account
- Customer managed
- Highly-flexible

Amazon EKS Managed Node Group

- Run in your account
- EKS handles provisioning and instance lifecycle

AWS Fargate

- Serverless, AWS-managed
- Compute is right-sized
- Pod-based billing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EKS Managed Node Groups



Nodes with a single command

Use Amazon EKS console, APIs, eksctl, AWS CloudFormation, or Terraform

Add Kubernetes labels directly to nodes



Stay up to date

Launch with the latest Amazon EKS-optimized AMIs

Get the latest updates with a single command and rolling updates



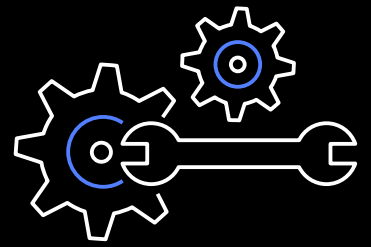
High availability

Run across multiple Availability Zones

Tagged for cluster autoscaler discovery

Integrated configuration health checks

Nodes automatically cordon and drain



Customizable and cost effective

Use Amazon EC2 launch templates to modify worker nodes

Bring your own OS with custom AMI support

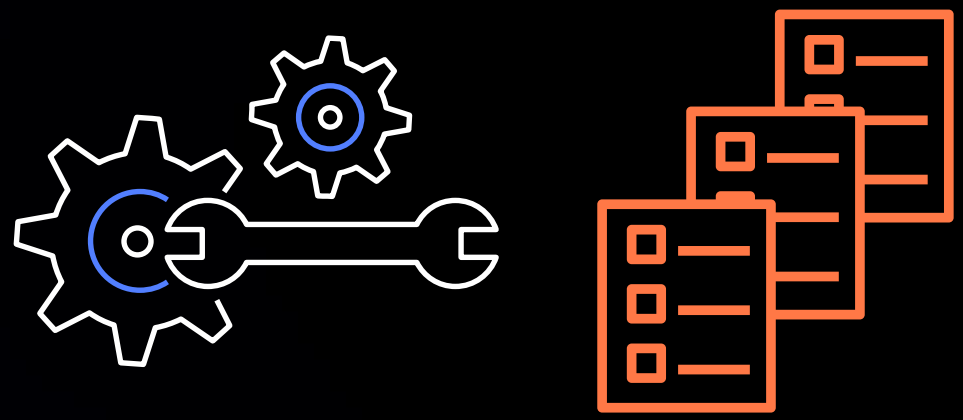
Fully integrated with Amazon EC2 Spot

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Custom launch templates with Amazon EKS

- Use custom launch templates to tune the kubelet running on the node
- Increasing queries per second (QPS) for events and communication with API server can unlock higher performance in large clusters



Every Amazon EKS Managed Node Group is backed by an Amazon EC2 Auto Scaling group (ASG)

These ASGs use launch templates to configure the Amazon EC2 instances launched

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EKS Managed Node Groups parallel upgrades



Amazon EKS Managed Node Groups simplify Amazon EC2 worker node upgrades, but took longer to complete for larger node groups …

Configure the number of nodes to upgrade simultaneously using the *maxUnavailable* parameter

- As a percent
- As number of nodes

Applications that are more fault-tolerant can benefit from reduced time to complete upgrades

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

EKS native support for Bottlerocket OS

Bottlerocket is an open-source operating system purpose-built by AWS for running containers

- Faster node boot time keeps clusters responsive at scale
- Includes only what's needed to run containers, improving resource utilization
- First-class support in Amazon EKS Managed Node Groups



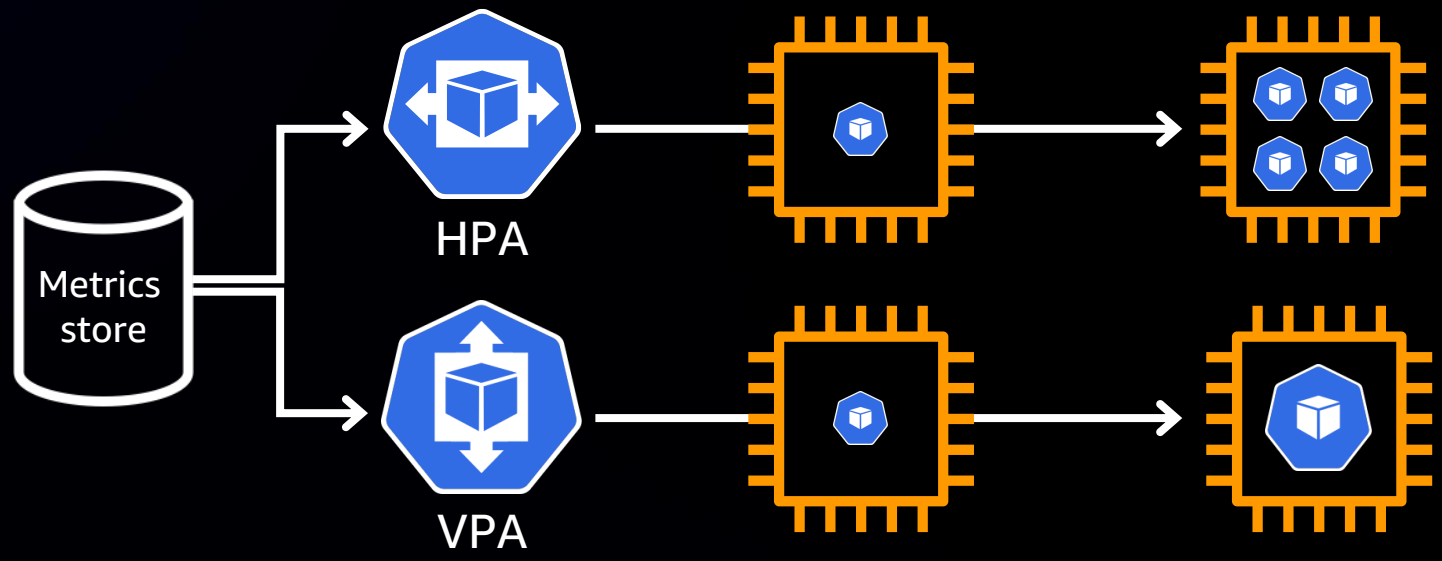
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster autoscaling with Karpenter

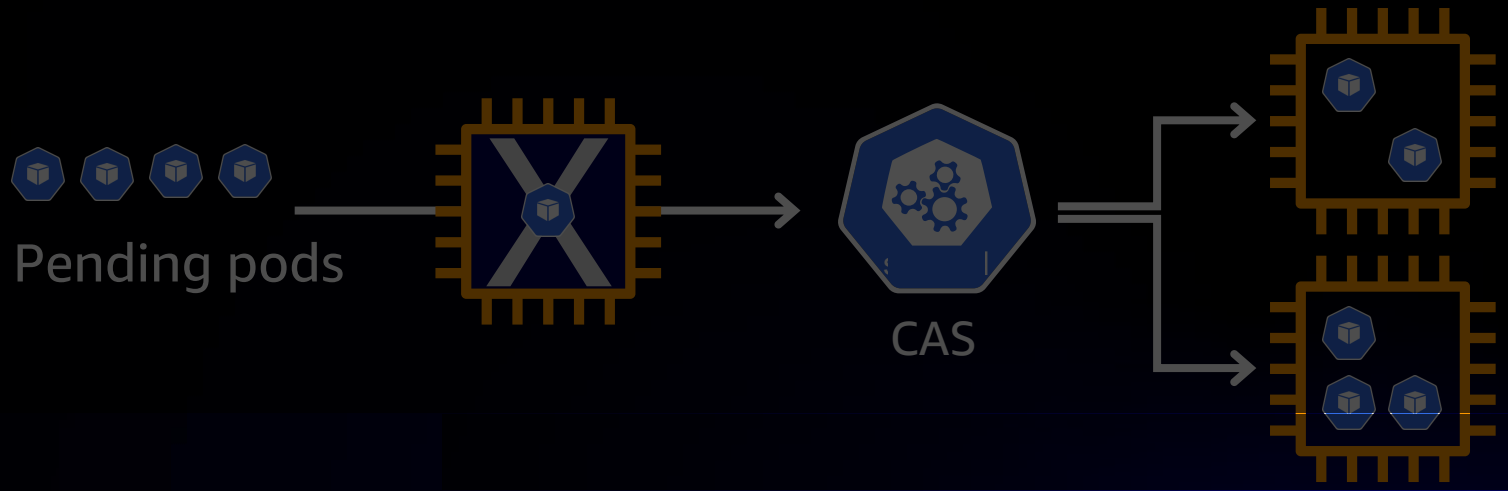
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kubernetes autoscaling overview

1. Horizontal Pod Autoscaling (HPA)
2. Vertical Pod Autoscaling (VPA)



3. Cluster Autoscaler (CAS)

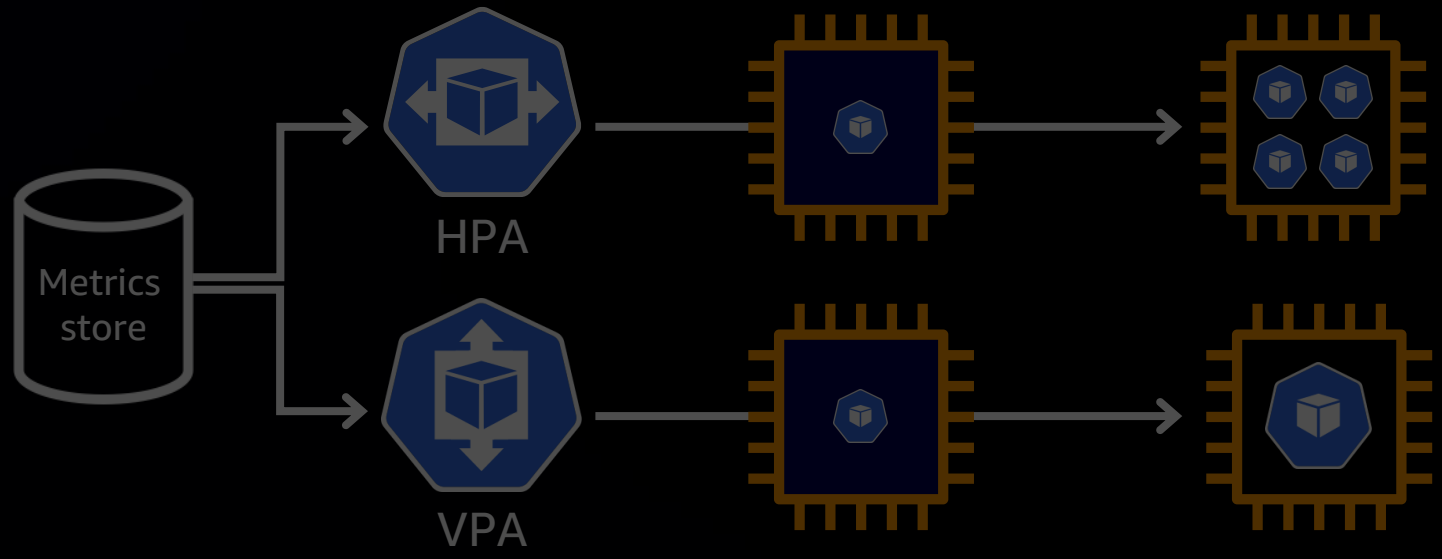


aws

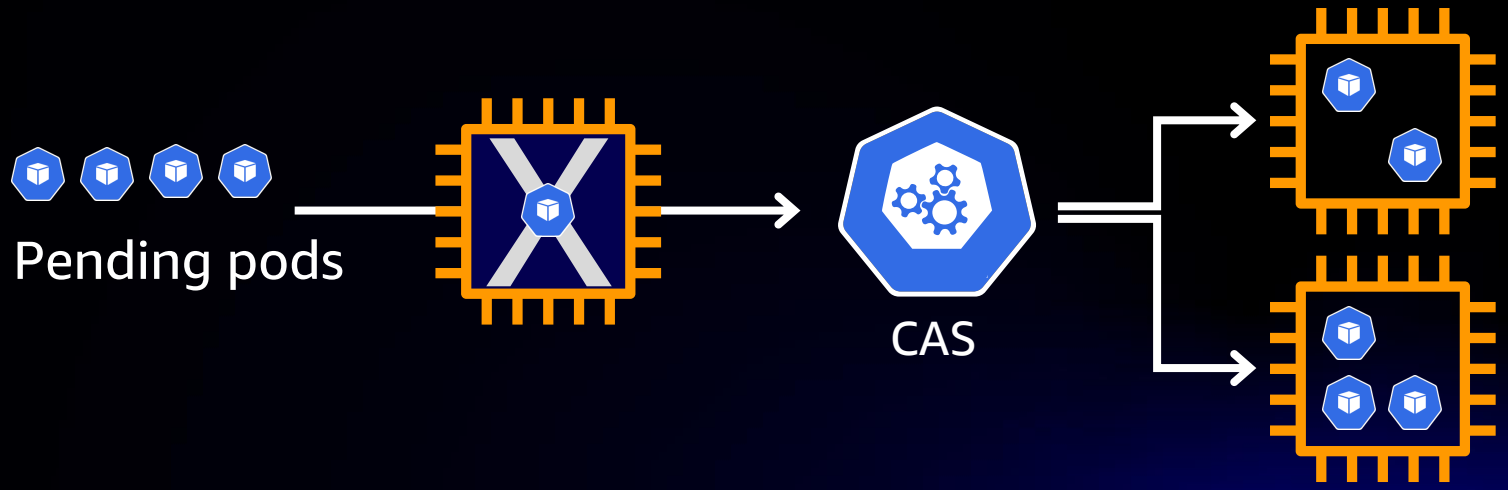
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kubernetes autoscaling overview

1. Horizontal Pod Autoscaling (HPA)
2. Vertical Pod Autoscaling (VPA)



3. Cluster Autoscaler (CAS)



aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

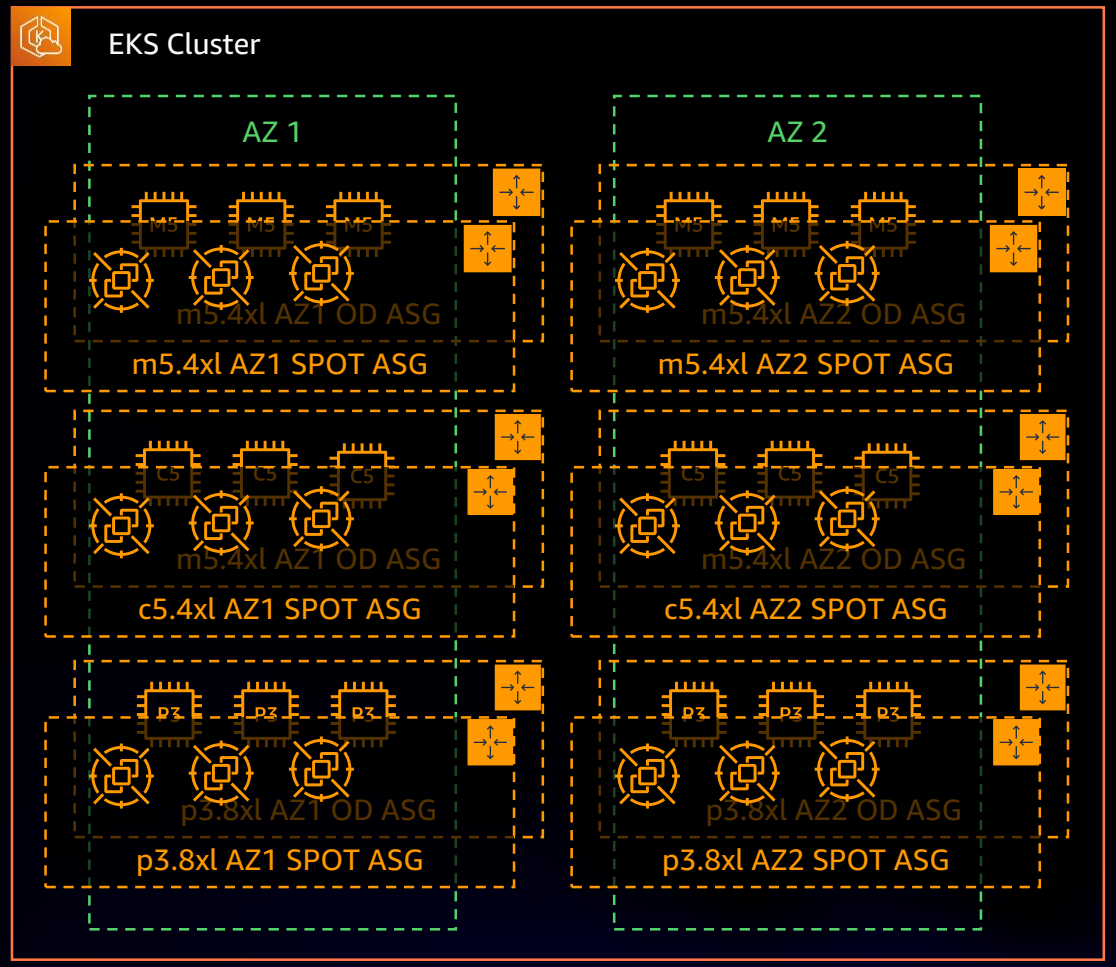
Use cases and challenges

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster autoscaling is hard to configure

Nearly **half** of AWS Kubernetes customers tell us that configuring cluster autoscaling is challenging

- Multi-AZ availability
- Instance type flexibility
- Spot capacity



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

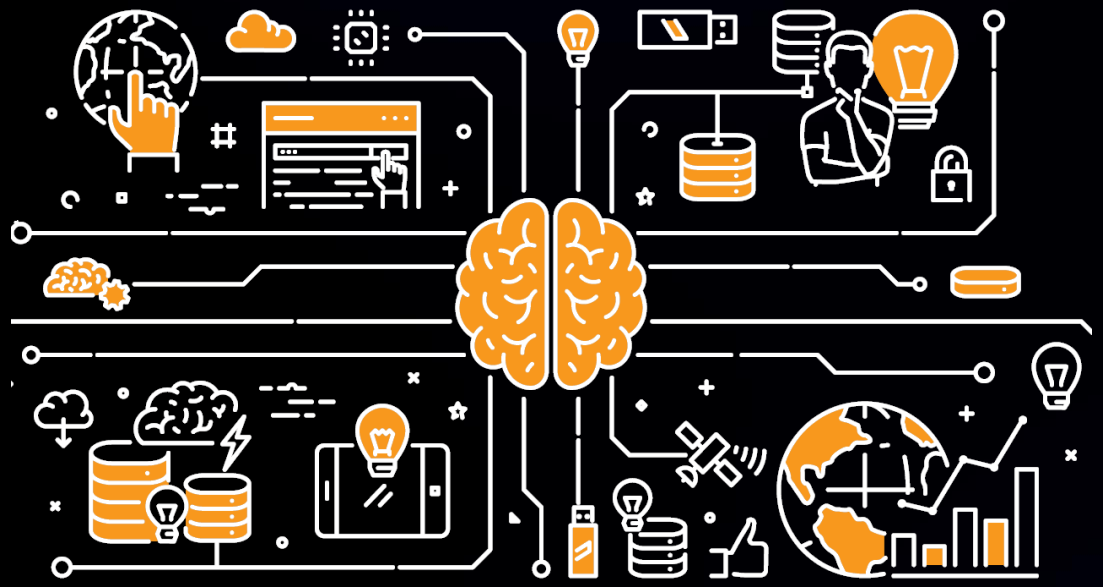
Use case: “Soft” multi-tenant clusters

- Clusters with a variety of “friendly” tenants, e.g., different application teams
- No one-size fits all configuration; different workloads need different compute resources
- Today, hard to get high cluster utilization and efficiency simply



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use case: Machine learning workloads



- Quickly iterate and experiment, each time spinning up many expensive accelerated instances
- Waiting to acquire and deploy capacity slows pace of innovation
- Slow scale-down performance decreases cluster efficiency

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Karpenter v0.5 now generally available

Karpenter is an open-source, flexible, and high-performance Kubernetes cluster autoscaler

- Dynamically chooses best-suited compute resources
- Automatically adds or removes compute resources required
- Scale tested for high-performance
- AWS support in EKS clusters



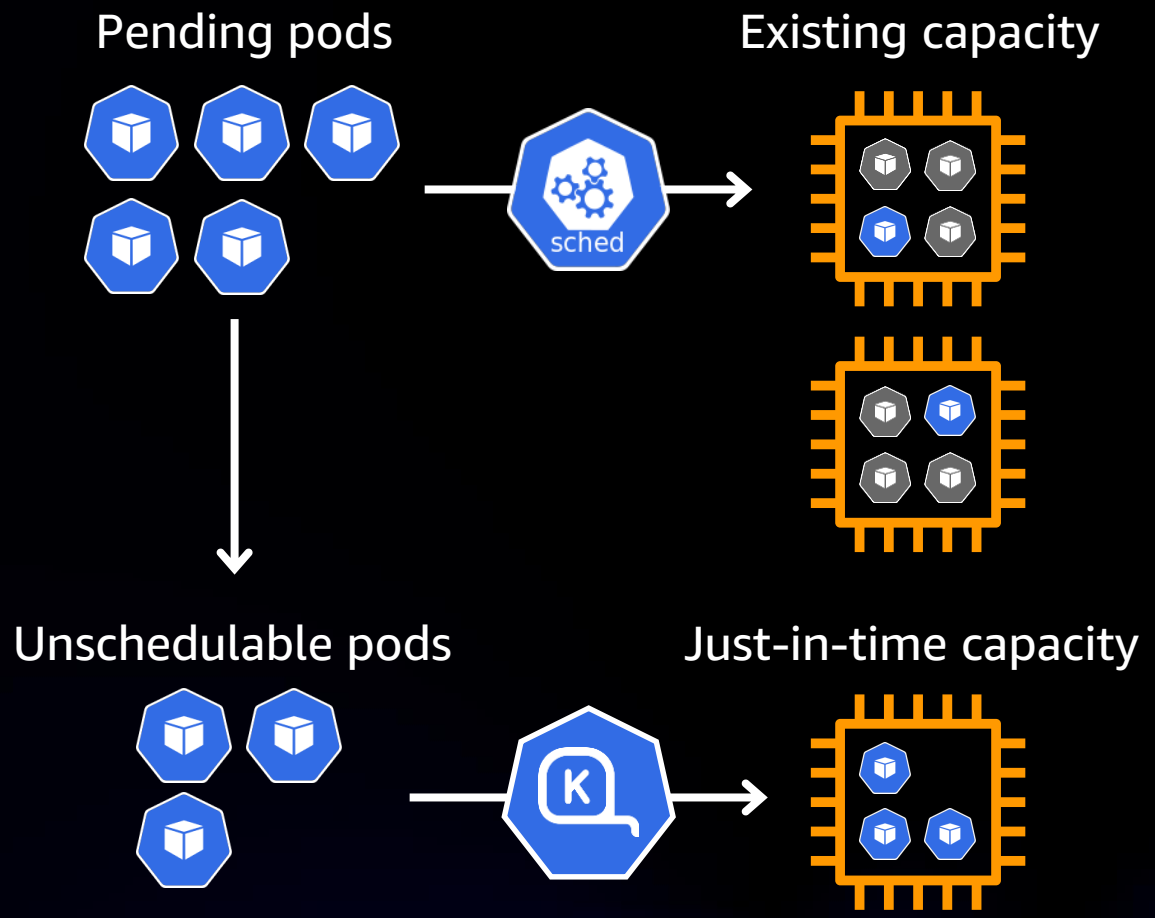
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How Karpenter works

Karpenter works in tandem with the Kubernetes scheduler and a cluster's compute provider (e.g., Amazon EC2)

- Launches right-sized nodes in response to unschedulable pods
- Terminates underutilized nodes
- Works with Kubernetes workload scheduling constraints



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster autoscaling with Karpenter

Karpenter brings **flexibility** and performance

- Compute is best-fit to cluster workloads automatically
- Declarative K8s custom resource definition (CRD) configuration:

```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
spec:
  ttlSecondsUntilExpired: 2592000
  ttlSecondsAfterEmpty: 30
  provider:
    cluster:
      name: my-cluster
```



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cluster autoscaling with Karpenter

Karpenter brings flexibility and **performance**

- Deeply and directly integrating with Amazon EC2 means cluster autoscaling can be **really fast**







0 seconds *15 seconds* *30 seconds*

2022 goal

1 pod 15s, 1,000 pods 30s (p90)

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Karpenter roadmap and community

Karpenter v0.5 is generally available

- Announcing AWS support
- Tested at scale



2022 roadmap

- Cluster capacity optimization
- Volume-aware scheduling
- Amazon EKS Anywhere support
- Spot interruption integration

Learn more at
github.com/aws/karpenter

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Wrap up

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Where can I learn more?

Amazon EKS best practices guide

- Open-sourced best practices guide
- Check back often for new chapters

aws.github.io/aws-eks-best-practices

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Where can I give feedback and submit feature requests?

Containers roadmap

- Stay up to date with what we're working on
- Give us feedback and propose ideas
- Get notified when new features ship

tinyurl.com/3bp4tnzn



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Related sessions

Session	Title
CON206	Kubernetes at AWS: Strategy, road map, and vision
CON316-R1	High performance with Amazon EKS and hundreds of lessons learned
CON307	GitOps as continuous operating model
CON304	Deep dive on Amazon EKS
CON306	Best practices for detecting and mitigating events in Kubernetes clusters

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Alex Kestner
@kestneraws

Shyam Jeedigunta
@jeediguntashyam

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.