
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

CON330

Operating Kubernetes clusters at cloud scale

Vipul Sabhaya (he/him)

Senior Software Development Manager, Amazon EKS
AWS

Shyam Jeedigunta (he/him)

Senior Software Engineer, Amazon EKS
AWS

AWS is a leading place to run Kubernetes. 65% of organizations choose AWS to run their containers.

CNCF State of Cloud Native Development

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

Why Amazon EKS?

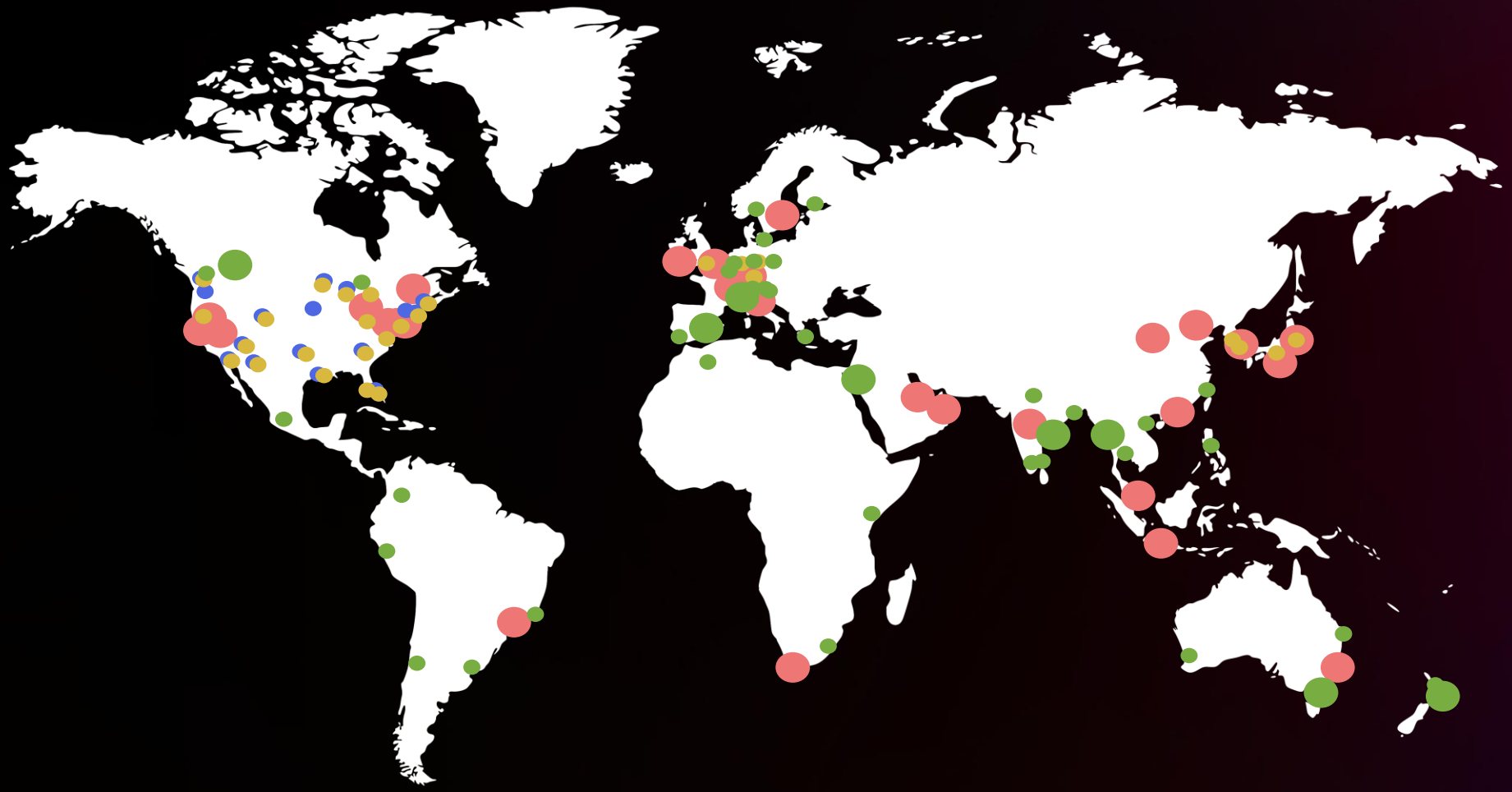
Running and scaling Kubernetes can be difficult and requires significant investment

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

Amazon global reach



30 geographic Regions

96 Availability Zones

21 local zones

29 wavelength zones

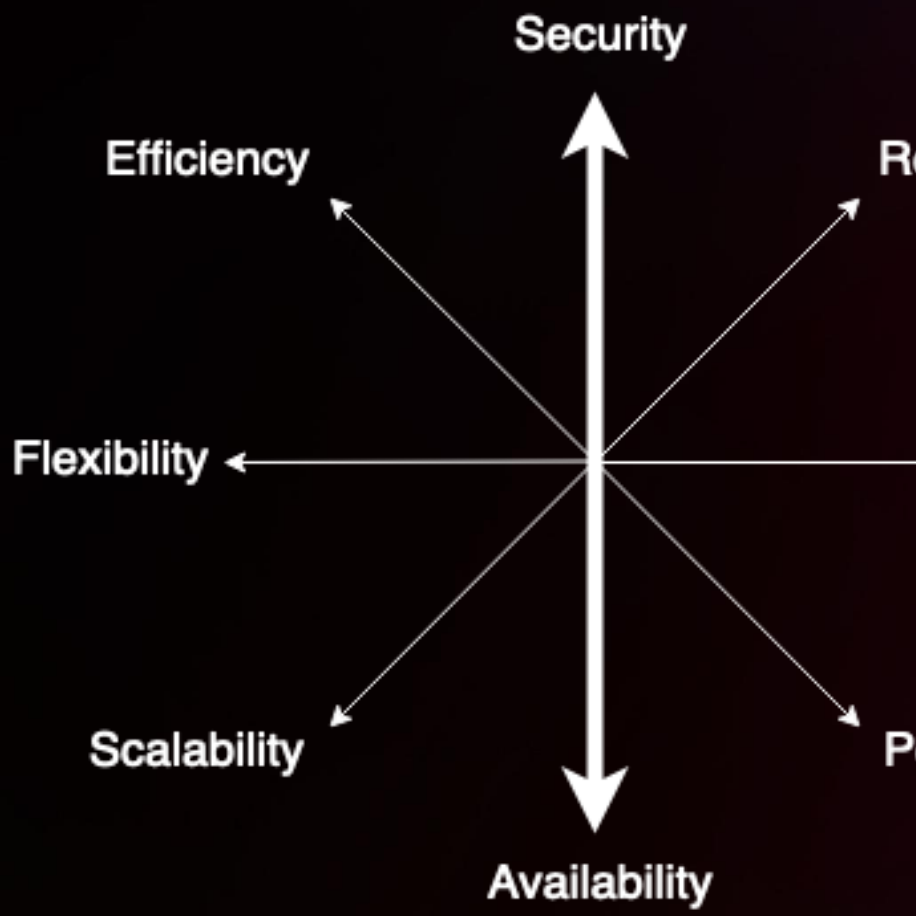
Track record

Thousands of clusters

Millions of upgrades

Billions of cpu hours

Agenda



Security p0

Supply chain | The software and systems you use are safe and the environment is protected from attack

Compliance | Systems meet the standards for your organization, industry sector, and government

Controls | Systems include options that allow you to ensure control over access, information processing, record keeping, and remediation

Availability

99.95%

Service Level Agreement*

24x7 support

* **An agreement**
This is not a goal or an objective
This is not a “best effort”

Durability

Etcd stores cluster state

Quorum

Backups

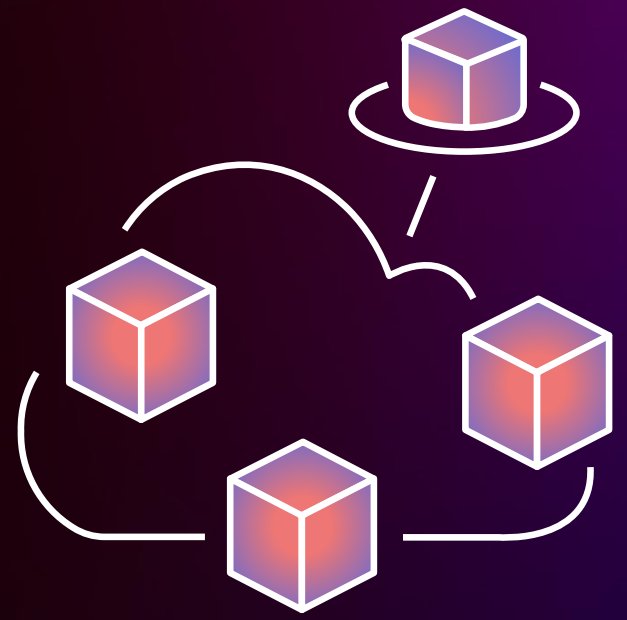
Persistent Volumes

Reliability

- Vetting releases / qualification
- Obsess over end-user outages
- Fixing bugs at scale
- COEs drive corrective actions
- Static stability

Scalability

- Push boundaries
- Work with Kubernetes sig-scalability
- Measure and Improve
- Beyond 5K nodes

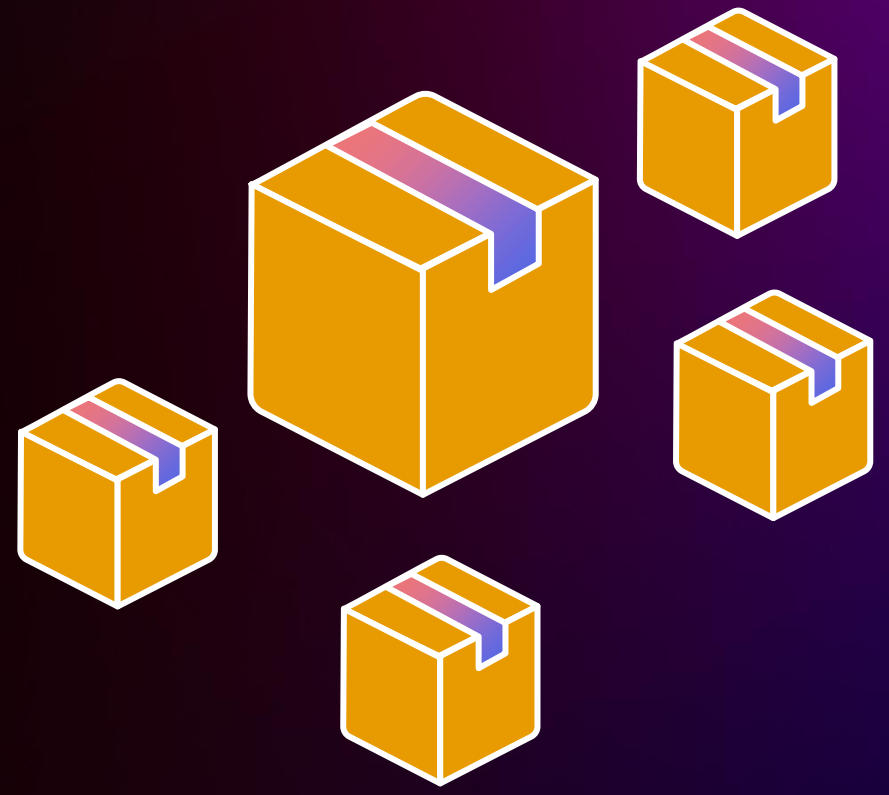


Performance

- Meaningful work is done in with guarantees
- SLOs – measure, improve, and honor
- Applies to EKS + Kubernetes

Efficiency

- Large fleet utilization
- Reduce costs to operate Kubernetes
- Cost savings for customers



Diving deeper

EKS cluster control plane

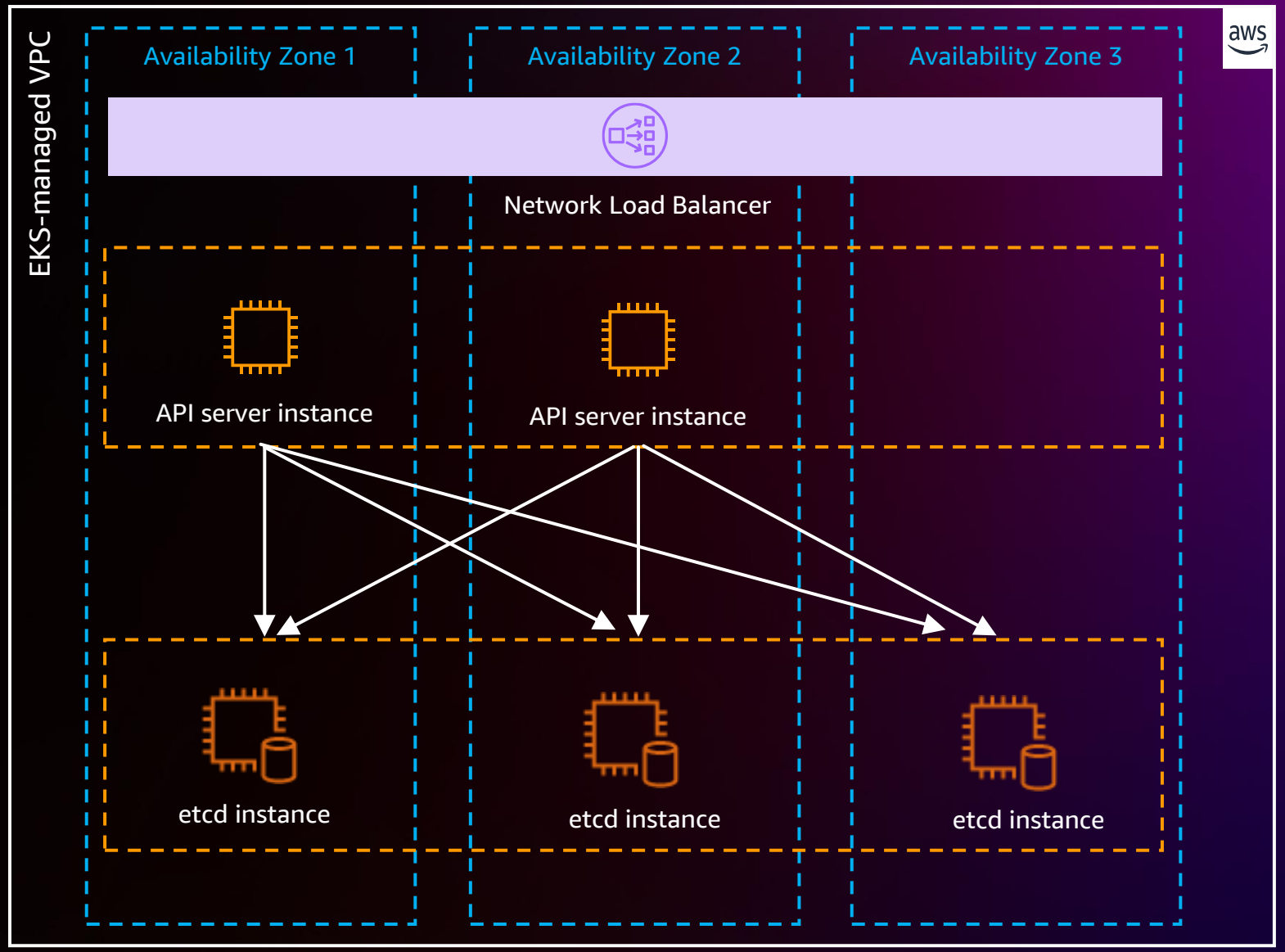
VPC-level isolation

Highly available API

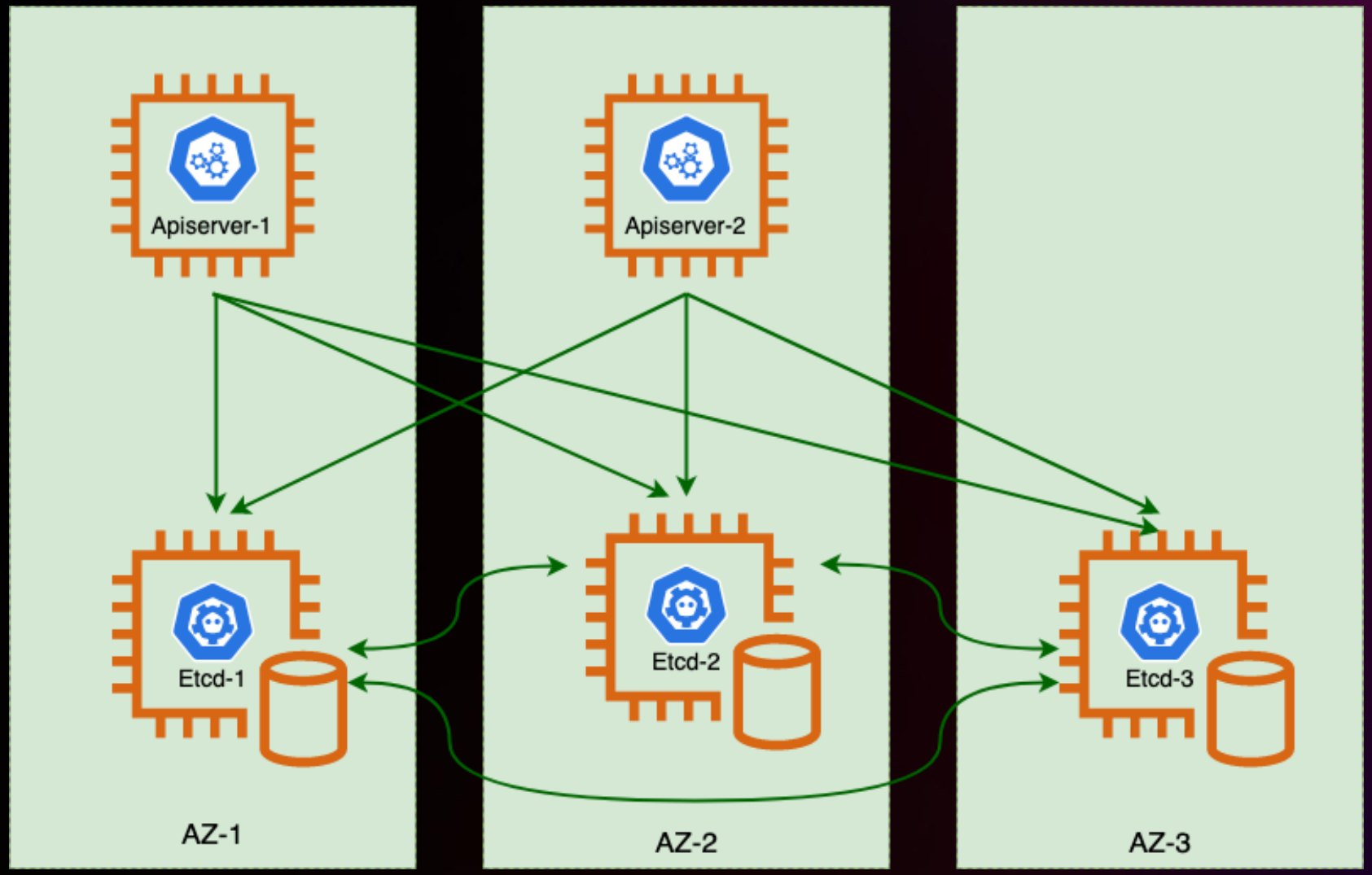
Scalable & performant

Etcd management

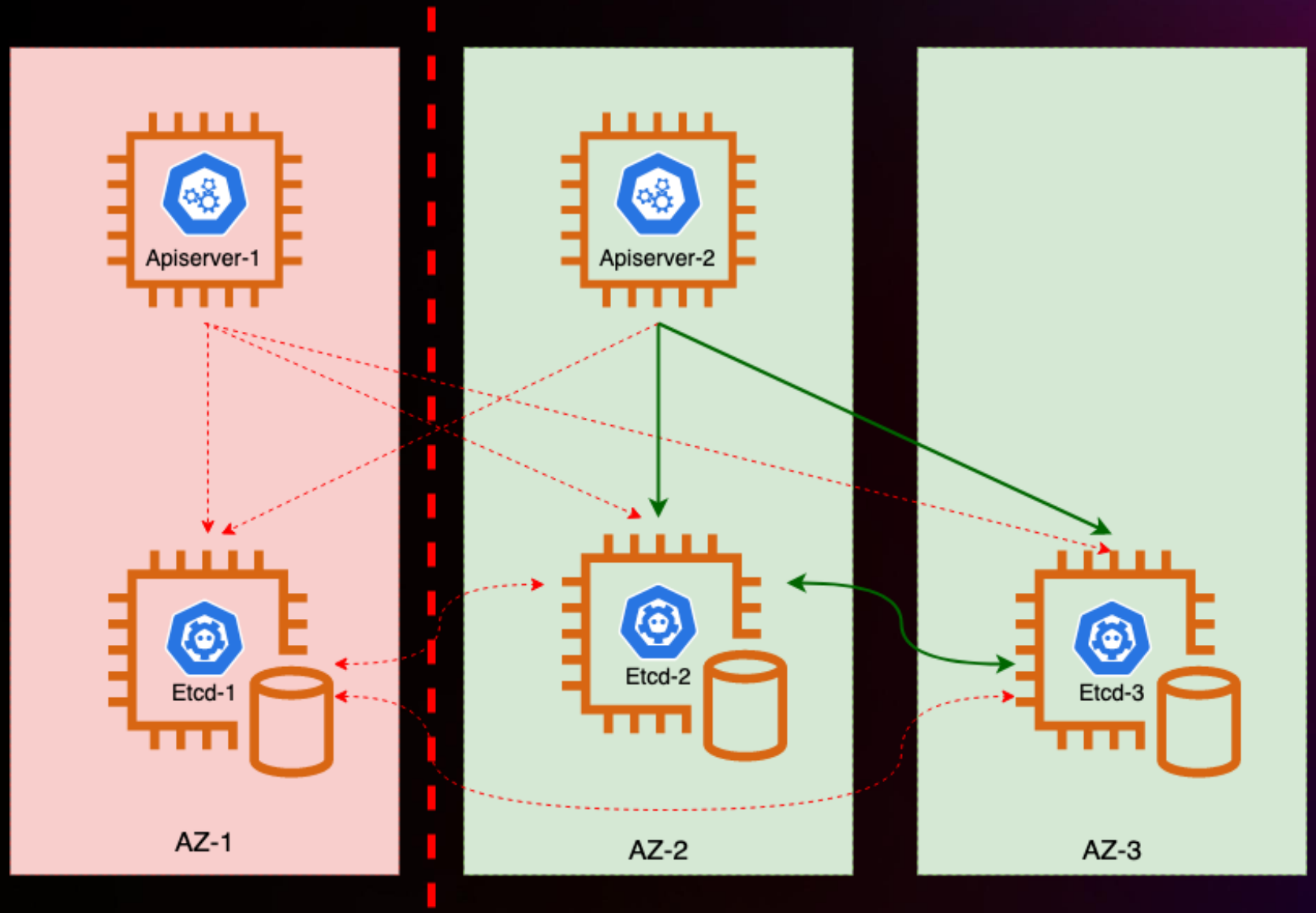
Software always up-to-date



Zonal redundancy



Zonal failure

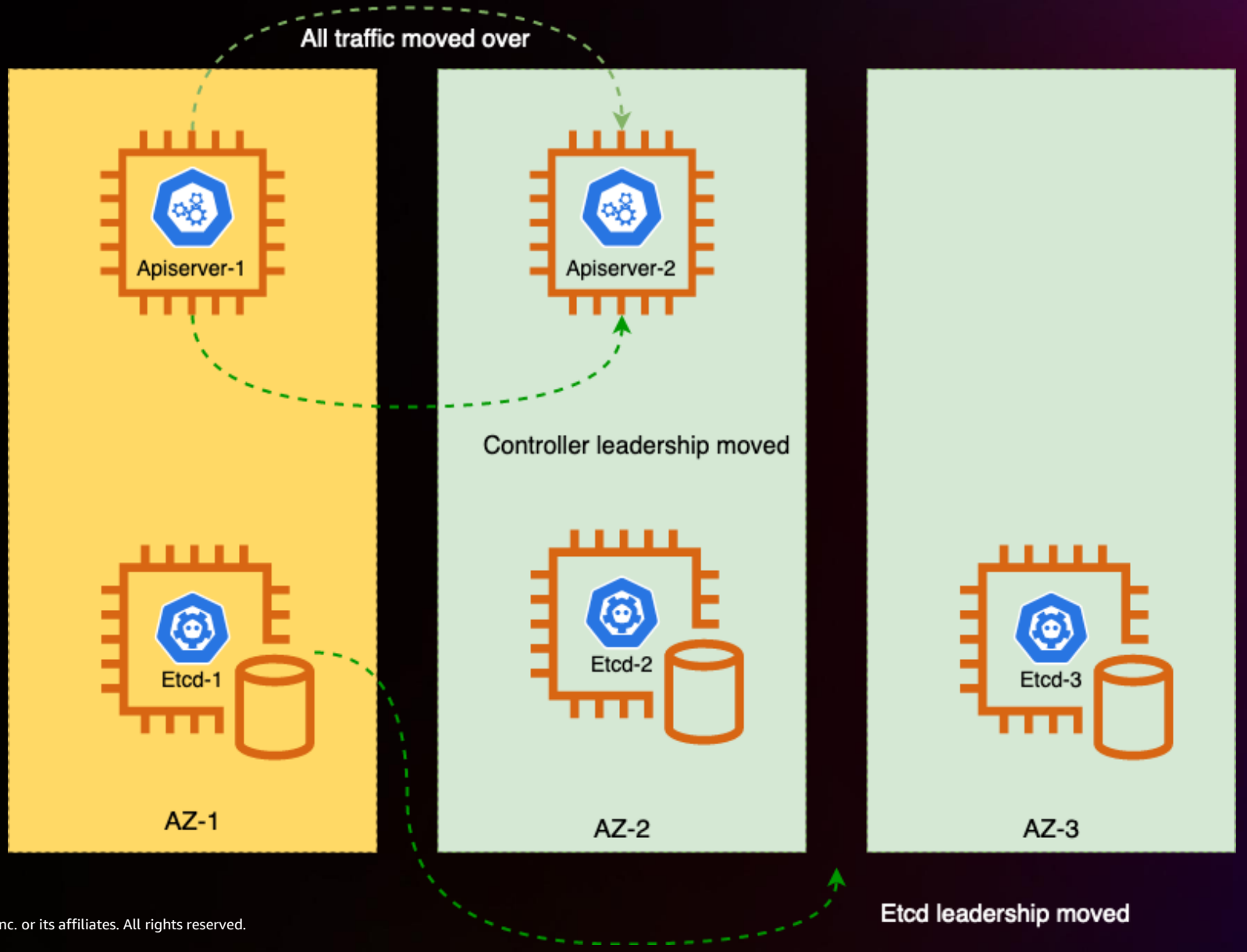


Zonal failure modes can be weird

- Vary in how they affect a zone
- Vary in what capacity they affect the zone
- They’re not 0-1 given grey failure modes
- Impact to software may be non-deterministic

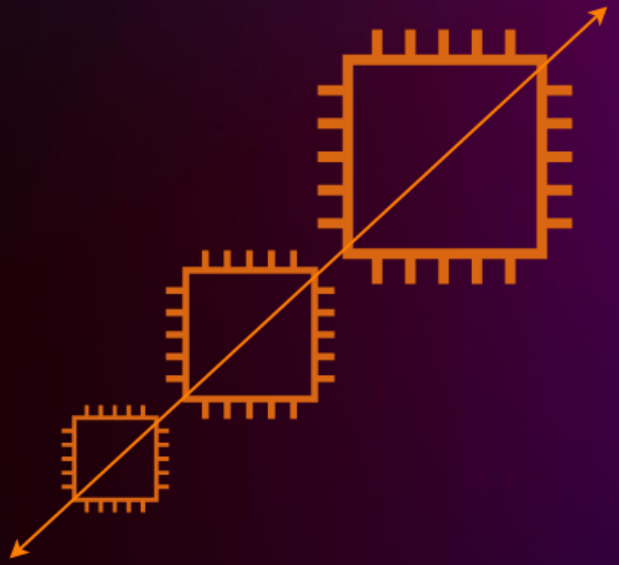
Coming Soon

Deterministic AZ failure resiliency



Scaling and performance

Control plane right-sizing



Instances scaled up/down seamlessly using signals like CPU/memory usage and cluster size.

Fleet-wide insights helped us take this to the next level.

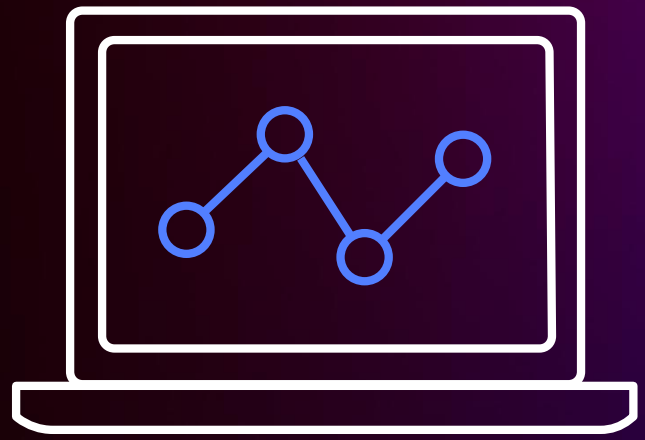
Improvements

Newer signals

- Etcd database size
- Etcd range requests

More dimensions

- Higher kube-controller-manager QPS
- Higher kube-scheduler QPS
- Up to 100k secrets with KMS encryption,



Faster response

- Scaling updates are 4x faster now
- Concurrent scaling of API server and etcd
- Proactively provision excess capacity

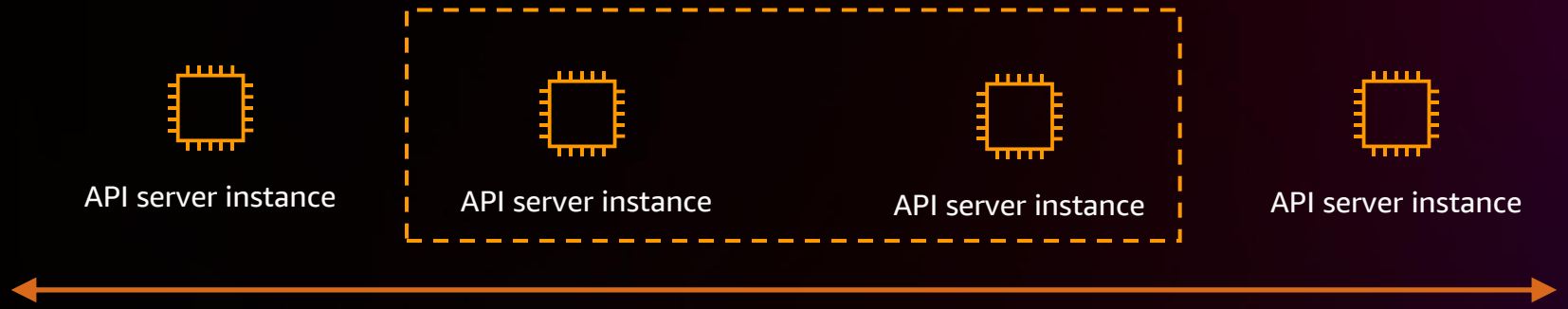
Customers can now scale seamlessly to 5000-node clusters (NO pre-scaling!)

Coming Soon

Horizontal scaling

Dynamically add/remove API server instances based on cluster size

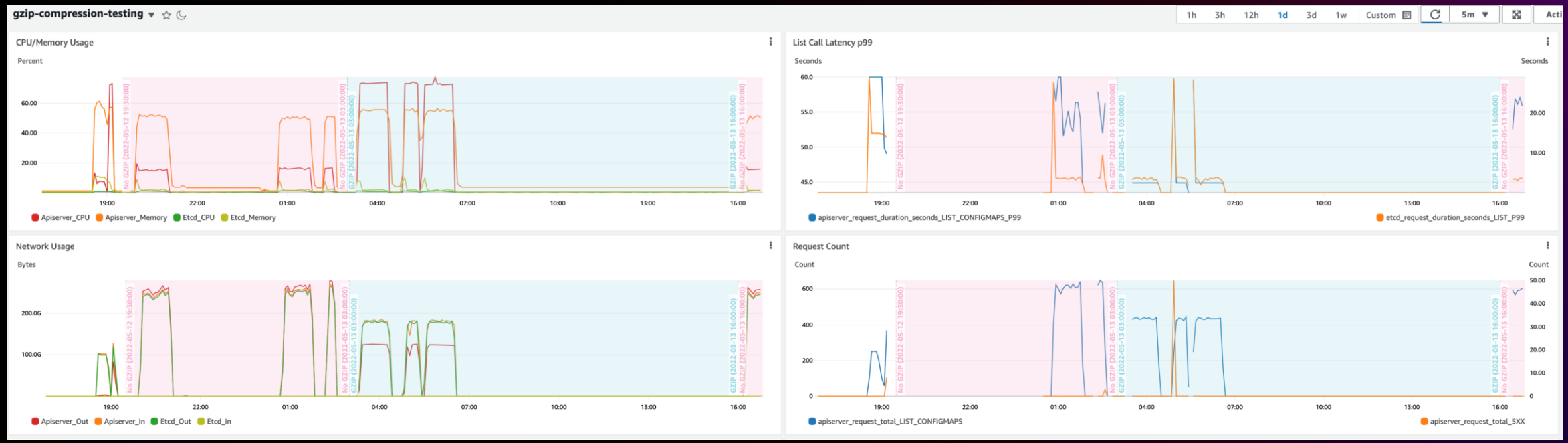
Provides better traffic split and failover at larger scales



Improving utilization and costs

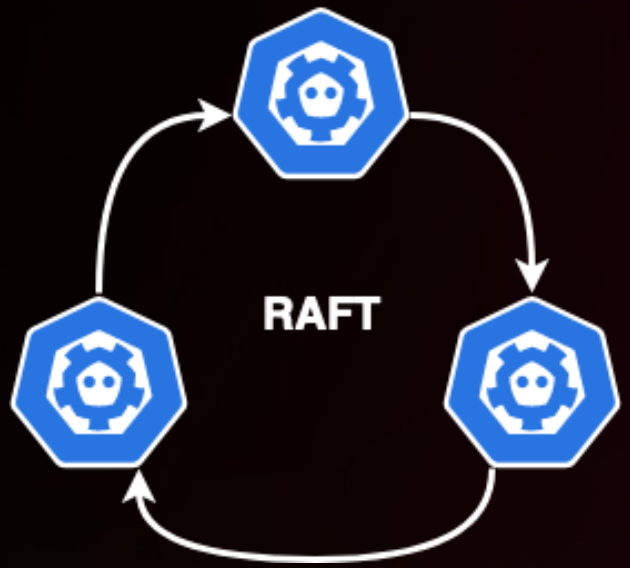
Continuous experimentation and instrumentation

Optimizing memory allocations, GC, gzip compression, etc.

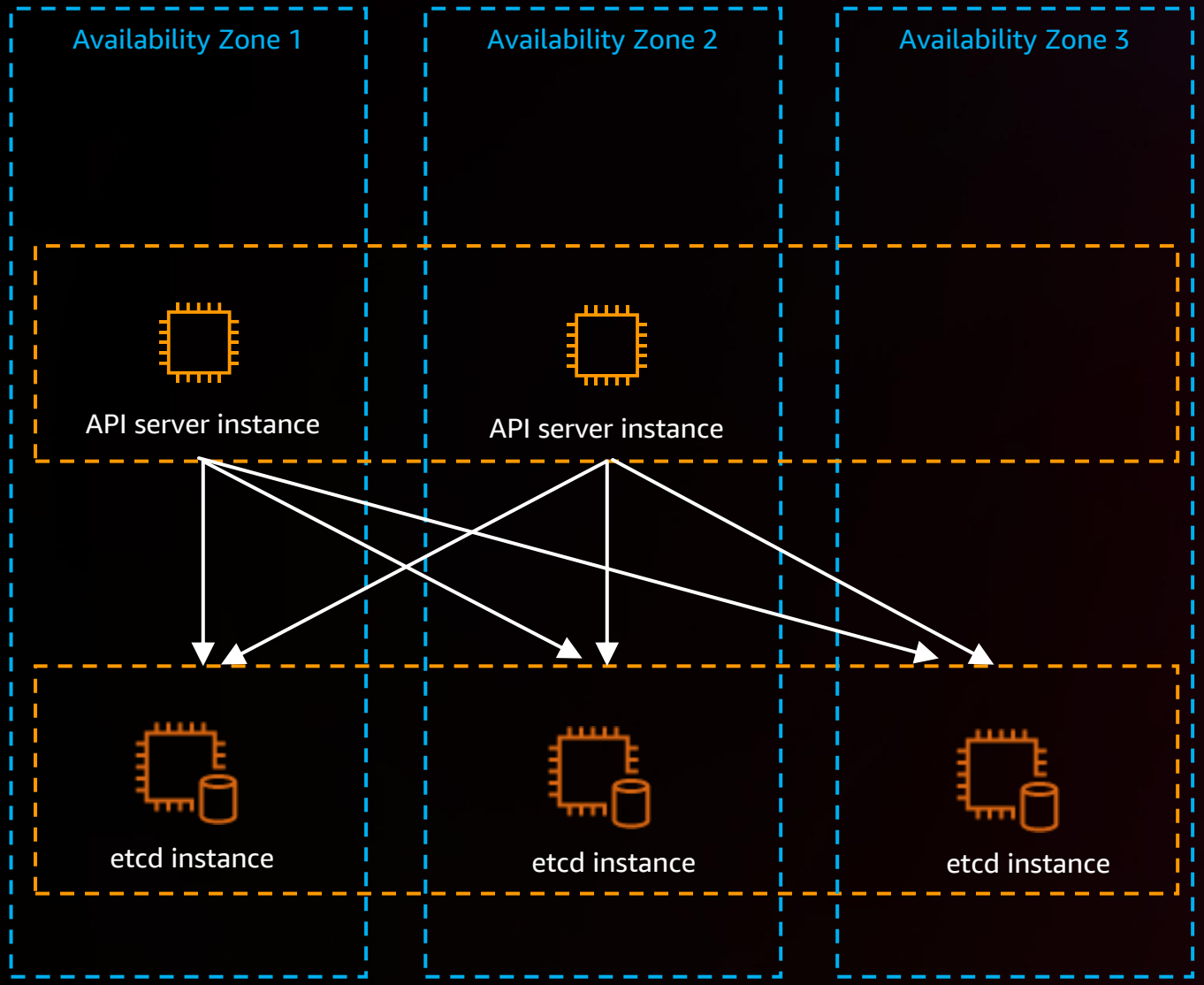


Etcd

- Strongly consistent, distributed key-value store
- Tolerant to single instance failure
- Right-sizing and periodic backups ensure availability and durability



Load balancing and failover



We got rid of the L7 load balancer between API servers and etcd

And switched to client-side load balancing

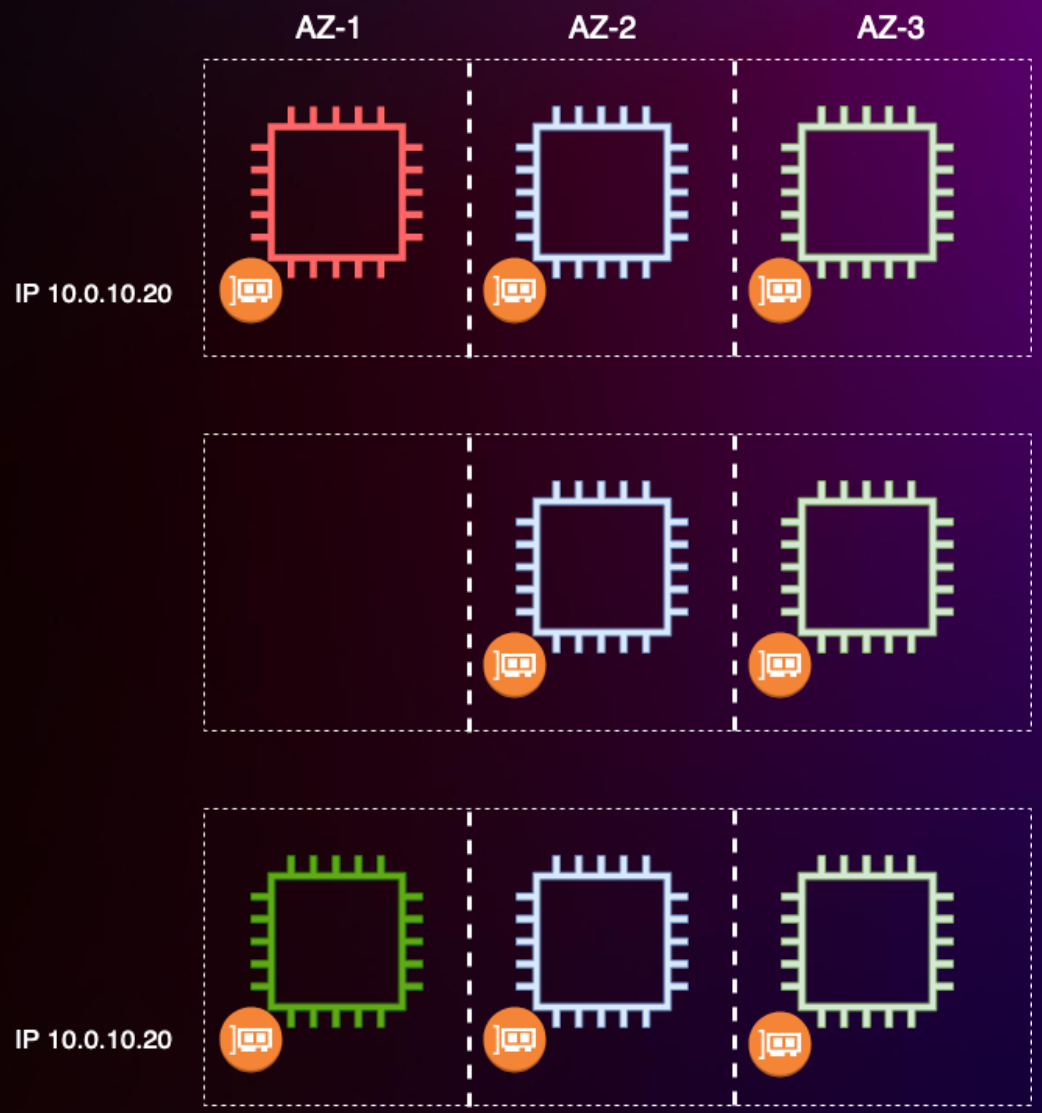
Improved performance, load-balancing and fault tolerance

Static IPs

Etcd members each get a fixed IP per AZ, that are static across instance recycles

Makes updates much more reliable since “membership info” doesn’t change

And API servers have a fixed set of etcd IPs to talk to. No more endpoint discovery!



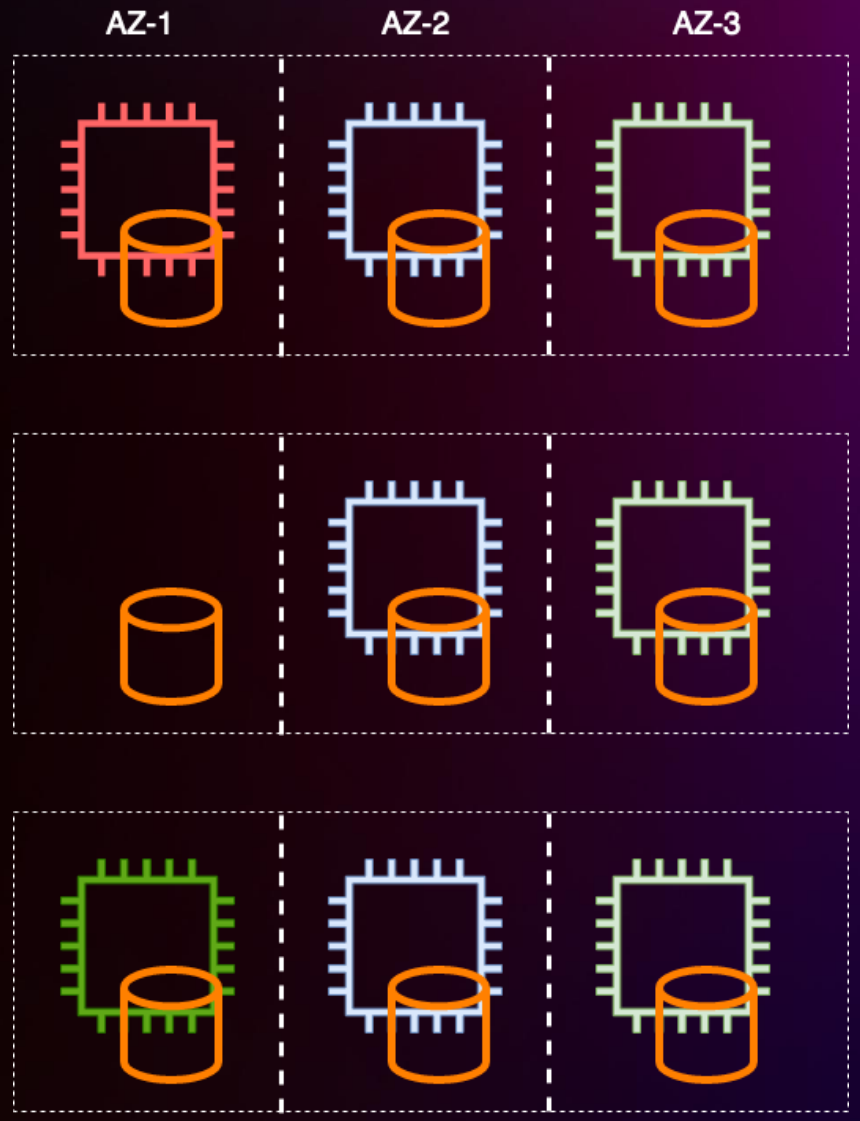
Coming Soon

Static volumes

Persistent volume that lasts longer than the instance's lifetime

Quorum loss no longer needs restore from backup (unless in extreme cases)

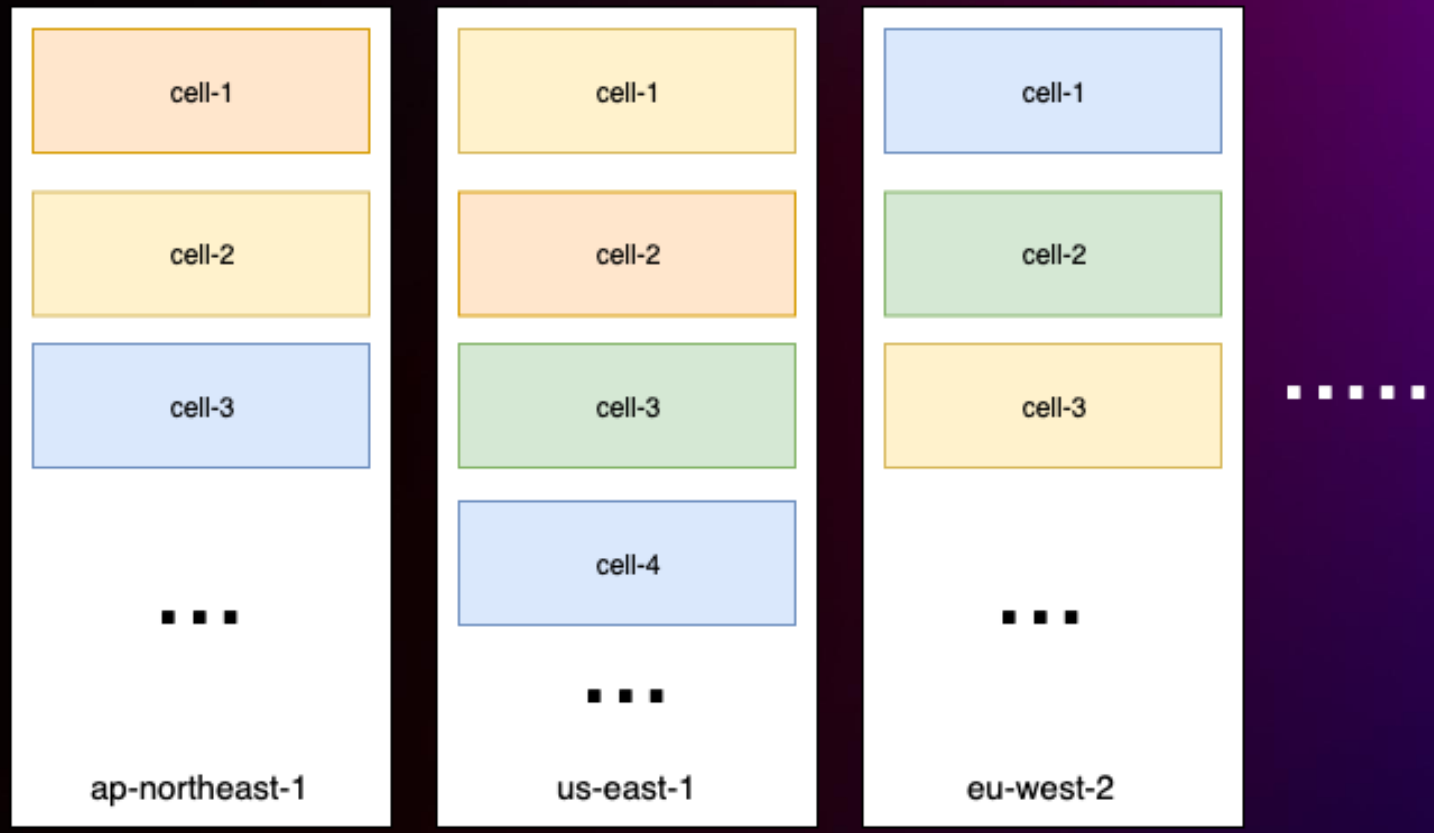
Faster MTTR (mean time to recovery)



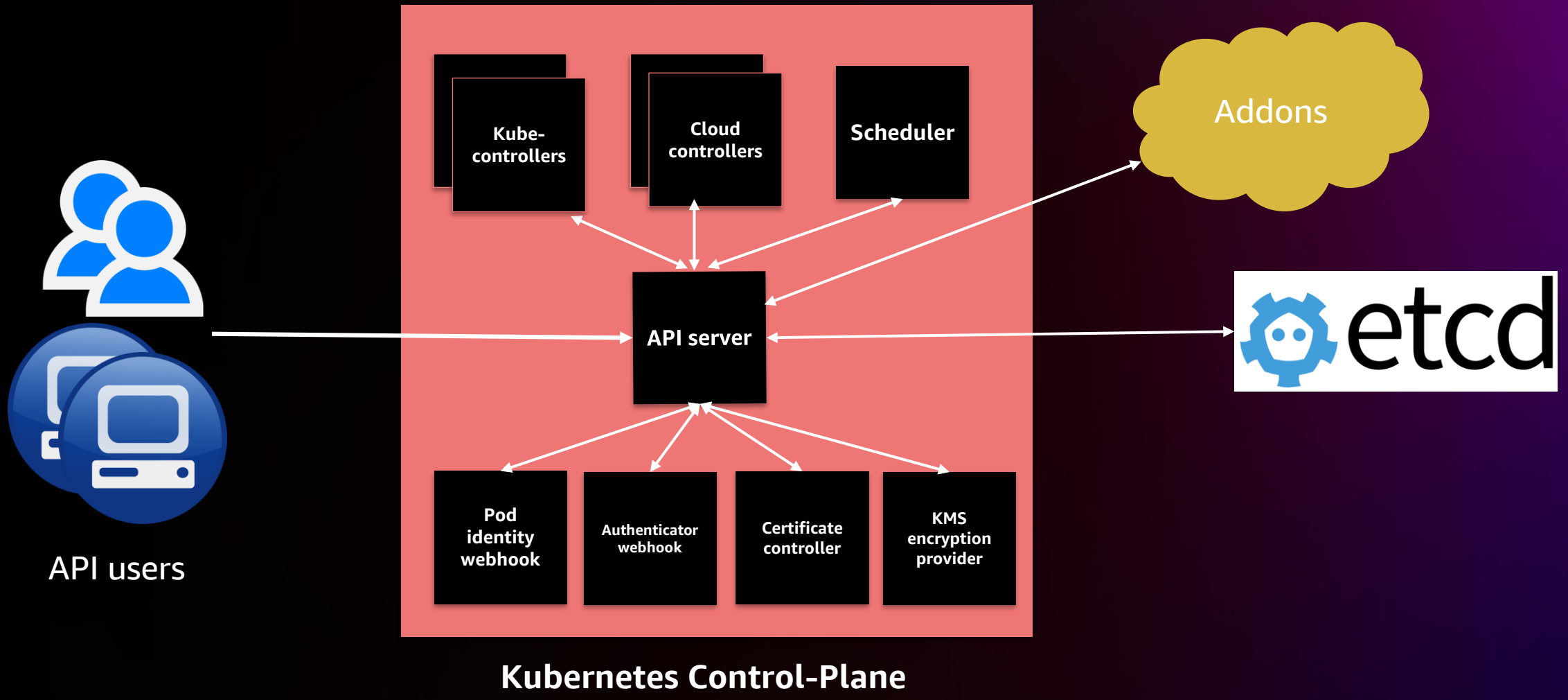
Software delivery

Service architecture

- Regions
- Cells within Regions
- Clusters within cells



Software delivery



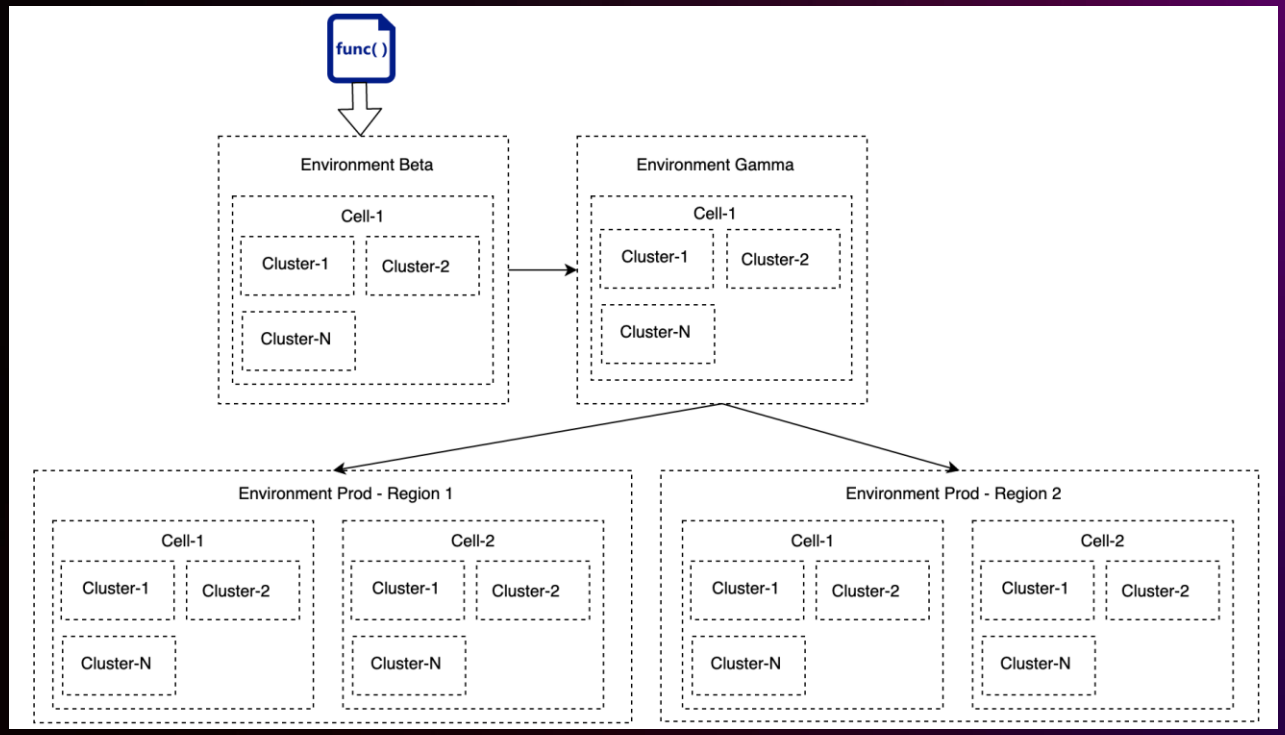
Software delivery

Testing at various levels

Tests of various kinds

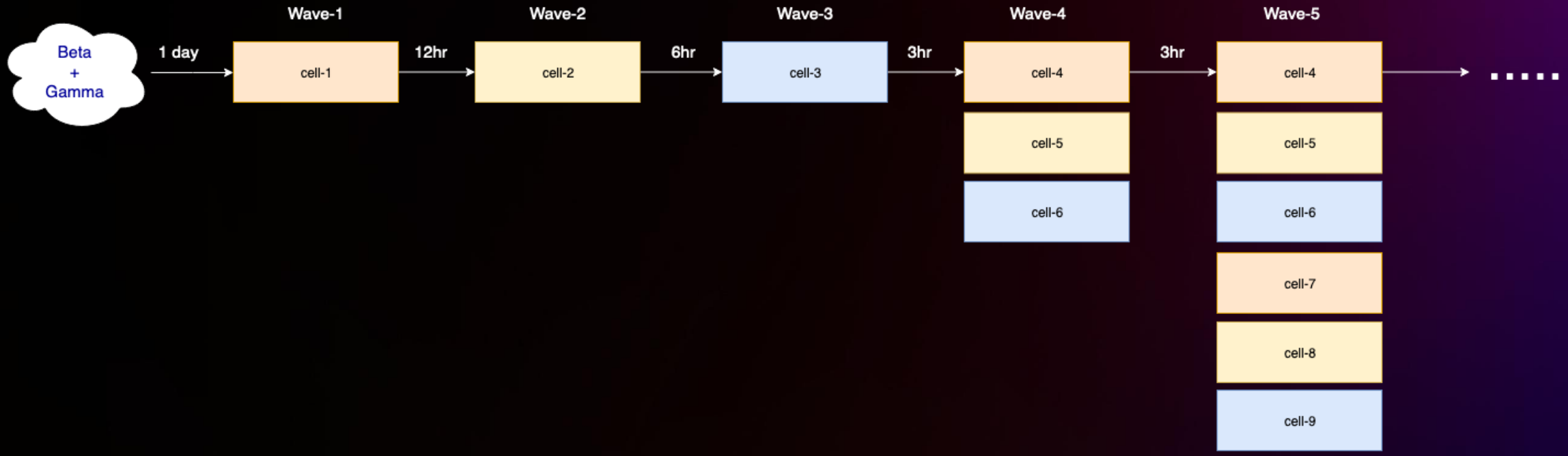
Testing at each deployment stage

Canaries that mimic customers



Platform version updates

- Hundreds of thousands of instances to update
- Need different kinds of software updates (AMIs, containers, agents)
- Balancing safety with velocity is important



Platform version updates

But it was slow…

- Each cluster update took about 40-45 mins
- Regional and cellular rate limits on throughput

Took ~4 weeks for a global rollout

New in 2022

Cluster update time improvements

Average time to update a cluster reduced from 40 minutes to < 10 minutes

Includes various update types, like version upgrades, OIDC provider, and encryption updates

What changed?

- Concurrent API server and etcd updates
- Blue-green style updates for API server
- Faster API server instance bootstrap
- Reduced etcd warmup time

Faster platform version updates

Global PV rollout time reduced from 4 weeks to <10 days

Enabled by the single cluster update time reduction and innovations in AWS services like Network Load Balancer and Amazon Route 53

For critical security fixes and bugs that needs accelerated rollout (<24 hrs), we perform targeted in-place container updates.

Wrap up

Where can I learn more?

Getting started with Amazon EKS

docs.aws.amazon.com/eks/latest/userguide

Amazon EKS best practices guide

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

Follow our public roadmap

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

github.com/aws/containers-roadmap

Related sessions

Session	Title	Time/location
CON401	**Run and manage Amazon EKS clusters on premises**	12/1 @ 11:00am – Mandalay Bay
CON405	**How to monitor and reduce your compute costs**	12/1 @ 11:45am - Wynn
CMP401	**Running efficient Kubernetes clusters on Amazon EC2 with Karpenter**	11/30 @ 1:45pm - Venetian
ANT330	**Run Apache Spark on Kubernetes with Amazon EMR on Amazon EKS**	12/1 @ 2:00pm – Caesars Forum
CON325	**Building containers for AWS**	12/2 @ 10:45am – Caesars Forum

Thank you!

Vipul Sabhaya

in/vipulsabhaya

@vipuls

Shyam Jeedigunta

in/shyamjvs

@jeediguntashyam

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