M A D 3 0 1

Kubernetes on AWS with Amazon EKS

Nathan Taber
Sr. Product Manager
AWS Container Services

aws SUMMIT

AWS best practices for building modern applications

- Create a culture of innovation by organizing into small DevOps teams
- Continually evaluate your security posture by automating security
- Componentize applications using microservices
- Update applications & infrastructure quickly by automating CI/CD
- Standardize and automate operations by modeling infrastructure as code
- Simplify infrastructure management with serverless technologies
- Improve application performance by increasing observability

We’re making AWS the best place to run containers and Kubernetes

AWS container services landscape

Management

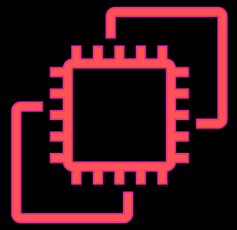
Deployment, Scheduling, Scaling & Management of containerized applications

Amazon Elastic Container Service

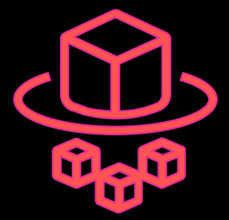
Amazon Elastic Kubernetes Service

Hosting

Where the containers run



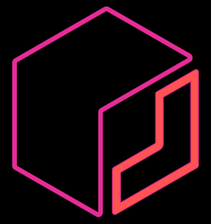
Amazon EC2



AWS Fargate

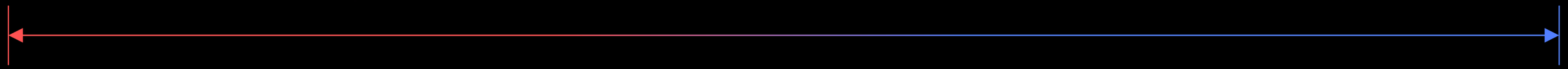
Image Registry

Container Image Repository

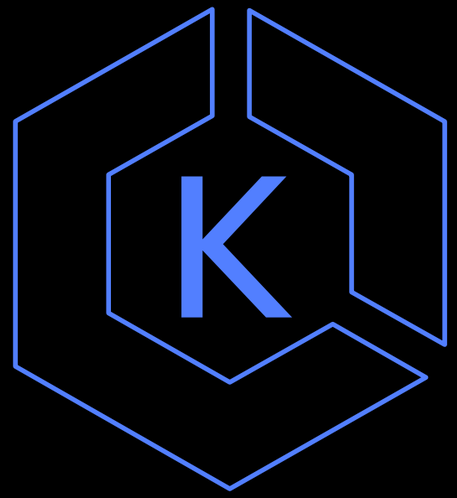


Amazon Elastic Container Registry

Balancing flexibility and simplicity: Workload-by-workload



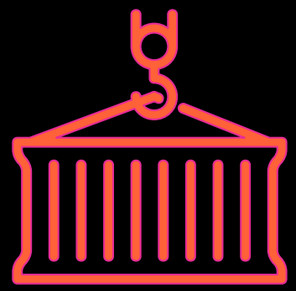
Flexibility focused	Value simplicity
Low level of opinion	High level of opinion
Low level of abstraction	High level of abstraction
Focus on infrastructure and configuration	Focus only on app and primitive
Installing, configuring, and managing my compute environment is critical to achieving my goals	*Having a standardized and on-demand compute environment is critical to achieving my goals*



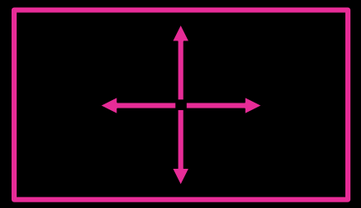
Amazon Elastic Container Service for Kubernetes

aws SUMMIT

What is Kubernetes?



Open source container management platform



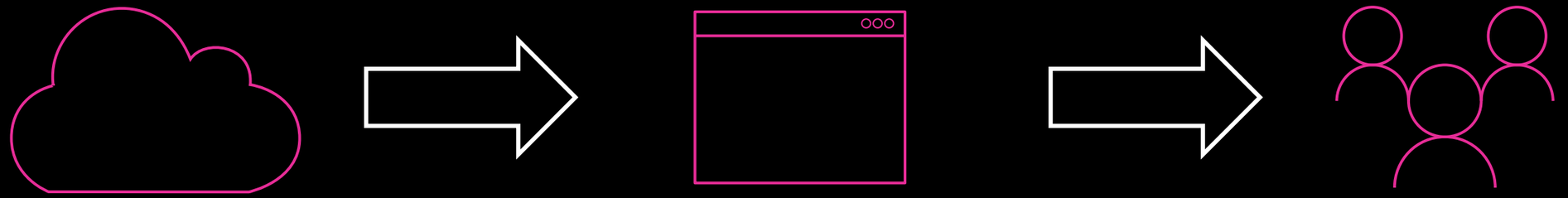
Helps you run containers at scale

Gives you primitives for building modern applications

Community, contribution, choice

kubernetes

But where you run Kubernetes matters



Quality of the cloud platform

Quality of the applications

Your users

+

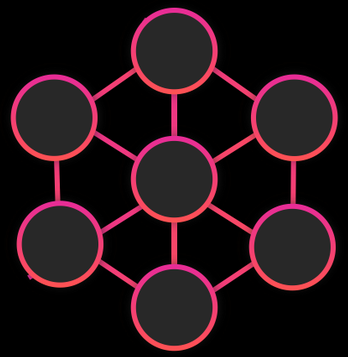
aws

51%

of Kubernetes workloads run on AWS today

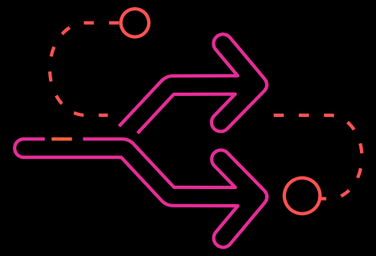
— **CNCF**

How are customer using Amazon EKS?

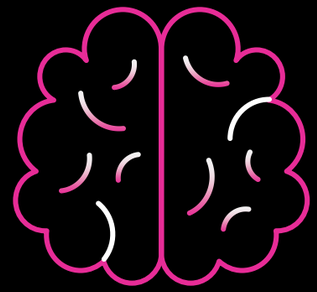


Microservices

Platform as a service



Enterprise App Migration



Machine Learning

Customers adopting Kubernetes on AWS

Customer example: Snap

> “Undifferentiated Heavy Lifting is work that we have to do that doesn’t directly benefit our customers. It’s just work. Amazon EKS frees us up to worry about delivering customer value and allows developers without operational experience to innovate without having to know where their code runs.

More detailed talk: AWS New York Summit 2018 - Run Kubernetes with Amazon EKS (SRV318)

Who is using Amazon EKS?

"We built the next generation of our PaaS using Amazon EKS for large enterprise workloads. We manage thousands of applications and have hundreds of DevOps teams.

Who is using Amazon EKS?

"Kubernetes is fast becoming the preferred solution for container orchestration. Its biggest downside is that it is not simple to set up and operate. Amazon EKS gives us all the benefits of Kubernetes, but takes care of managing the hard stuff.

We can dedicate less resources to deployment and operations as result.

Which customers are using Amazon EKS?

"The performance from Amazon EKS makes it feasible to effectively manage large-scale databases delivering over a million queries per second. Amazon EKS also helps with our cluster management and scalability challenges.

Rich partner ecosystem

Foundation

CANONICAL

RANCHER

docker

Red Hat

DevOps

Spinnaker

pulumi

CloudBees
The Enterprise Jenkins Company

GitLab

ATLASSIAN

Monitoring & logging

circleci

DATADOG

sysdig

New Relic.

Security

aqua

Twistlock.

NeuVector

Networking

linkerd

TiGERA
CLOUD NETWORKS, SECURED

weaveworks

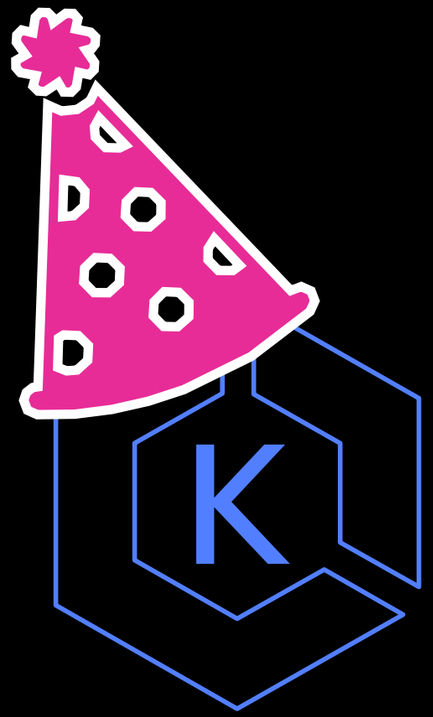
Our tenets

1. Amazon EKS is a platform to run **production-grade workloads**. Security and reliability are our first priority. After that we focus on doing the heavy lifting for you in the control plane, including life cycle-related things like version upgrades.

2. Amazon EKS provides a **native and upstream Kubernetes** experience. Amazon EKS provides vanilla, un-forked Kubernetes. In keeping with our first tenant, we ensure the Kubernetes versions we run have security-related patches, even for older, supported versions as quickly as possible. But there's no special sauce and no lock in.

3. If you want to use additional AWS services, **integrations** are as **seamless** as possible.

4. The Amazon EKS **team** in AWS **actively contributes** to the **upstream Kubernetes** project and the wider CNCF activities, both on the technical level as well as community, from communicating good practices to participation in SIGs and working groups.



Happy Birthday Amazon EKS!

One year, 50+ new features, regions, and improvements

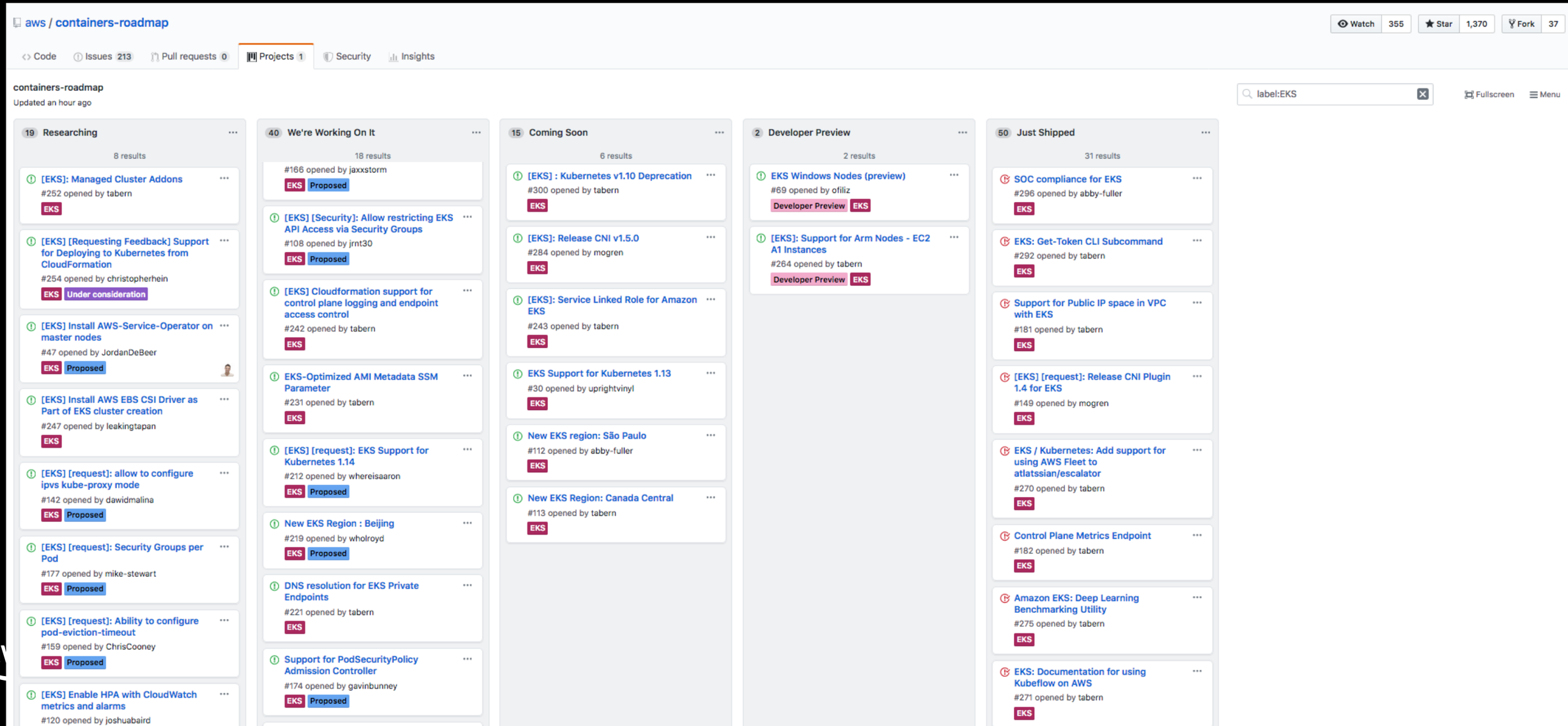
Amazon EKS, a year in review

June – December 2018:
Amazon EKS achieves K8s conformance, HIPAA-eligibility, Generally available
Amazon EKS AMI build scripts and AWS CloudFormation templates available in GitHub.
Support for GPU-enabled EC2 instances, support for HPA with custom metrics.
Amazon EKS launches in Dublin, Ireland
Amazon EKS simplifies cluster setup with `update-kubeconfig` CLI command
Amazon EKS adds support for Dynamic Admission Controllers (Istio), ALB Support with the AWS ALB ingress controller
Amazon EKS launches in Ohio, Frankfurt, Singapore, Sydney, and Tokyo
Amazon EKS adds Managed Cluster Updates and Support for Kubernetes Version 1.11, CSI Driver for Amazon EBS

2019:
Amazon EKS launches in Seoul, Mumbai, London, and Paris
Amazon EKS achieves ISO and PCI compliance, announces 99.9% SLA, cluster creation limit raised to 50
API Server Endpoint Access Control, AWS App Mesh controller
Windows support (preview), Kubernetes version 1.12,
CSI Drivers for Amazon EFS, Amazon FSx for Lustre, Control Plane Logs, A1 (ARM) instance support (preview)
Deep Learning Benchmark Utility, Public IP Address Support,
Simplified cluster authentication, SOC compliance, Kubernetes 1.13, PodSecurityPolicies,
Container Insights, CNI 1.5.0, Amazon ECR, AWS PrivateLink Support

Open-source roadmap

https://github.com/aws/containers-roadmap/



Amazon EKS services roadmap: Highlights

Shipped

- Amazon EKS control plane logs
- Support for public IP space in VPC
- SOC compliance
- Amazon EKS: Deep Learning Benchmarking Utility
- New Amazon EKS Region: Paris, London, Mumbai
- CNI v1.5.0

Coming soon

- Service linked role for Amazon EKS
- EKS Support for K8s version 1.13 + ECR AWS PrivateLink
- EKS-optimized AMI metadata SSM parameter
- IAM for Pods
- New Amazon EKS Regions: Beijing, Ningxia, Hong Kong

Working on it

- Amazon EKS on Fargate
- Managed Nodes
- Managed add-ons
- DNS resolution of Amazon EKS private endpoints
- New Amazon EKS Regions: Sao Paulo, Canada Central
- Next-generation CNI plugin

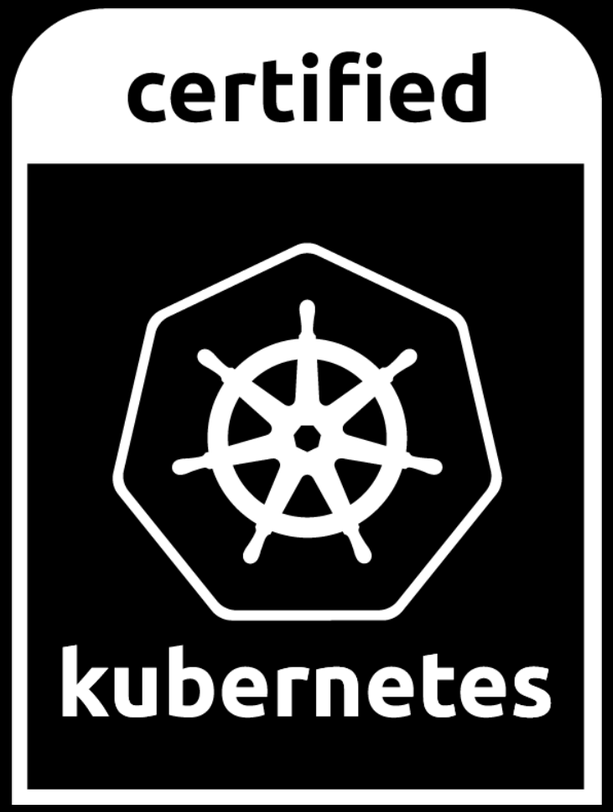
Amazon EKS deep dive

- Configuration & setup
- Availability
- Storage
- Operations
- Security
- Networking
- Logging
- Monitoring
- Application communication

Configuration & setup

aws SUMMIT

Amazon EKS is Kubernetes-certified



Kubernetes conformance

- Guaranteed portability and interoperability
- Timely updates
- Confirmability

Open-source and Amazon EKS

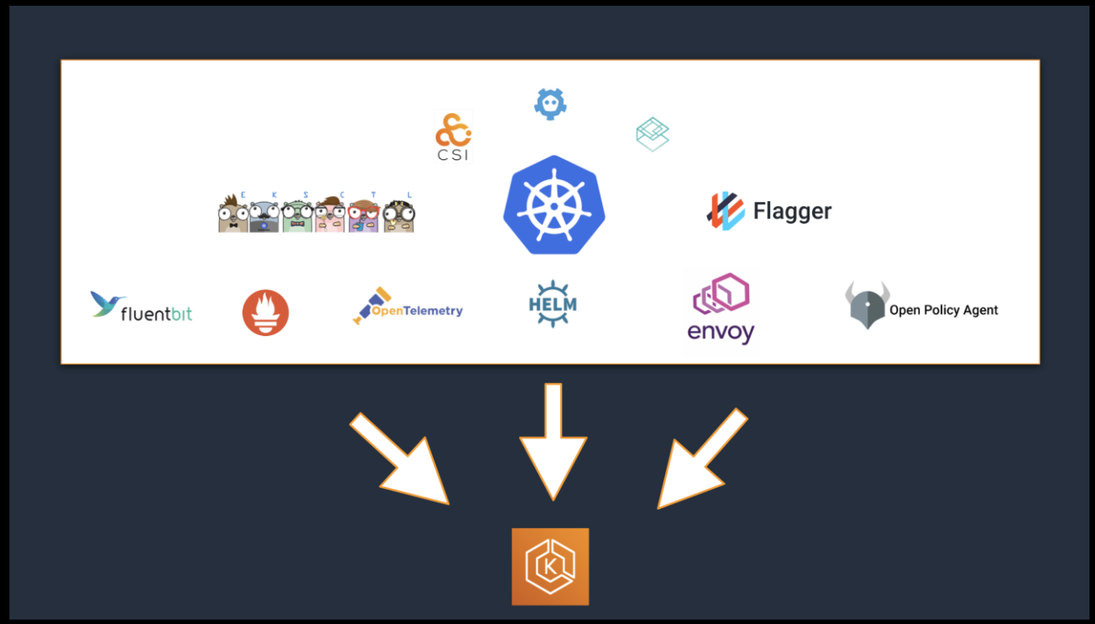
Amazon EKS runs 100% upstream Kubernetes

Key components of Amazon EKS are open source

- Amazon VPC CNI plugin
- AWS IAM authenticator
- Amazon EKS AMI

Team contributes to or manages 20+ OSS projects

- /kubernetes
- /kubernetes/autoscaler
- /aws-labs/aws-service-operator
- /weaveworks/eksctl
- Amazon EBS, Amazon EFS, Amazon FSx CSI drivers



Kubernetes versions

Latest: 1.13
Coming soon: 1.14

Amazon EKS will support up to three versions of Kubernetes at once

Deprecation in line with the community stopping support for older versions

Version 1.10 deprecation on July 22, 2019

eksctl - a CLI for Amazon EKS

- Single command cluster creation

```
eksctl create cluster --nodes=4
```

- Open source and on GitHub
- Built by Weave and AWS
- Official Amazon EKS CLI

Bring your own instances

Instance flexibility

Standard EC2 compute instance types

P2 and P3 accelerated instances

i3 bare metal

Spot Instances

Bring your own OS

Amazon EKS AMI build scripts

https://github.com/awslabs/amazon-eks-ami

Source of truth for Amazon EKS Optimized AMI

Easily build your own Amazon EKS AMI with Packer

Build assets for Amazon EKS AMI for each supported Kubernetes version

Windows containers

Run Windows containers and Windows Server nodes with Amazon EKS

Supports heterogeneous (mixed) clusters.

Kubernetes version 1.11+

Available in all Amazon EKS Regions

Developer preview:
https://github.com/aws/containers-roadmap

Provisioning worker nodes

AWS CloudFormation

eksctl

eksctl

Terraform
Pulumi
Rancher

…more

Partners

Amazon EKS-optimized GPU AMI

Includes NVIDIA packages to support Amazon P2 and P3 instances

Easily run TensorFlow on Amazon EKS

Now supporting P3dn.24xlarge instances

CUDA 10 with NVIDIA v410 coming soon!

Availability

aws SUMMIT

Global availability

Americas

Virginia, Ohio, Oregon

EMEA

Ireland, Frankfurt, London, Paris, Stockholm

Asia Pacific

Singapore, Tokyo, Sydney, Seoul, Mumbai

Service level agreement

99.9%

Service commitment

AWS will use commercially reasonable efforts to make the endpoint for an Amazon EKS Cluster available with a Monthly Uptime Percentage of at least 99.9% during any monthly billing cycle

In the event Amazon EKS does not meet the Monthly Uptime Percentage commitment, you will be eligible to receive a Service Credit.

Storage

aws SUMMIT

Container storage interface (CSI)

A flexible standard for orchestration
and storage provider connections

We support the CSI standard through following drivers:

Amazon Elastic Block Store: Amazon EBS CSI Driver

Amazon Elastic File System: Amazon EFS CSI Driver

Amazon FSx for Lustre: Amazon FSx CSI Driver

Storage volume lifecycle

Provisioning

- Static
- Dynamic*

Binding

- Control loop watches for PVC requests and satisfies if PV is available.
- For Dynamic, PVC will provision PV
- PVC to PV binding is one-to-one mapping

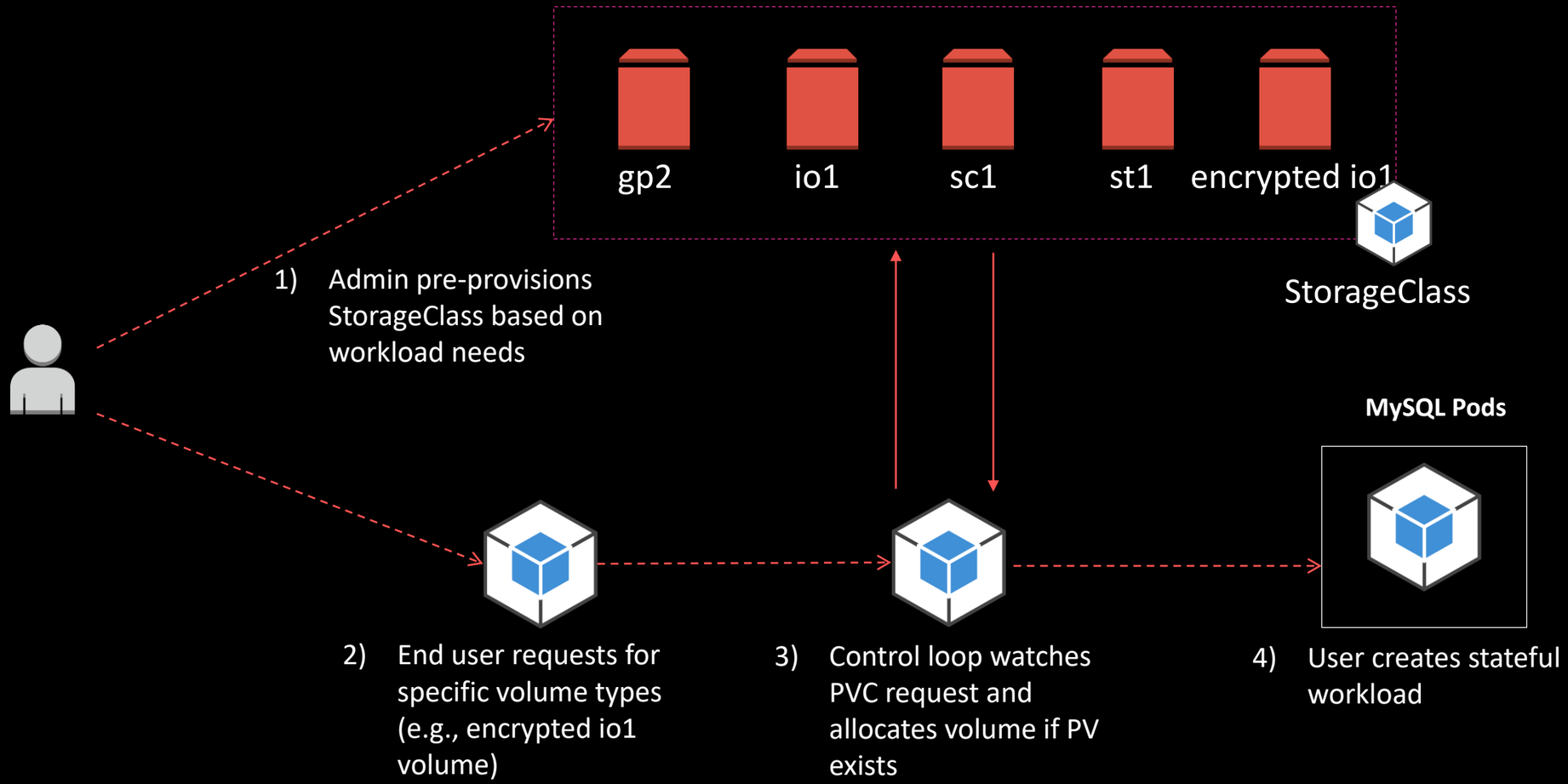
Using

- Cluster mounts volume based on PVC

Reclaiming

- Retain (default)
- Recycle
- Delete

What if I need specific volume type?



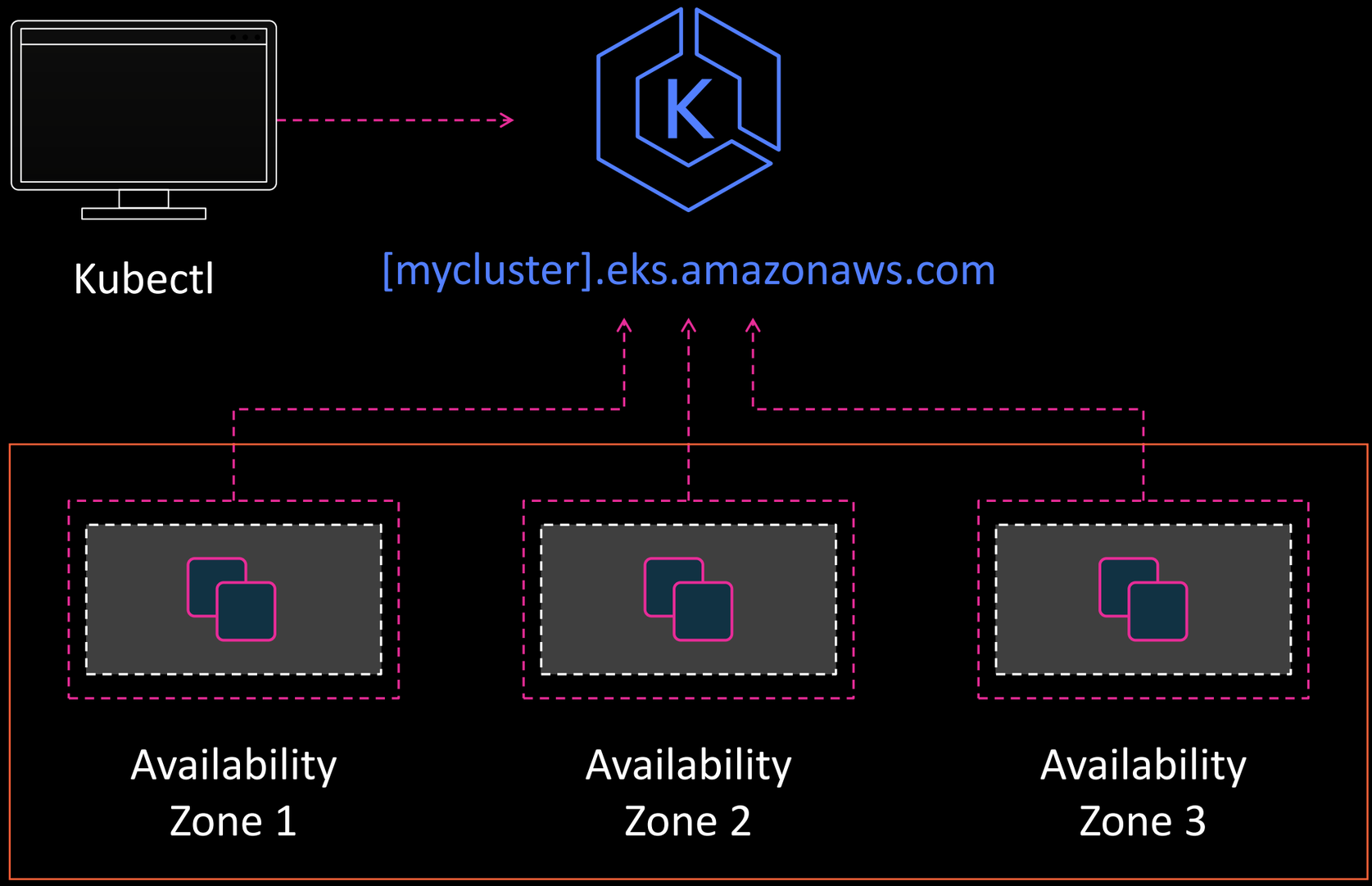
Operations

aws SUMMIT

Amazon EKS operational capability

- Architecture
- CI/CD for applications deployed on Amazon EKS
- Infrastructure elasticity

Architecture

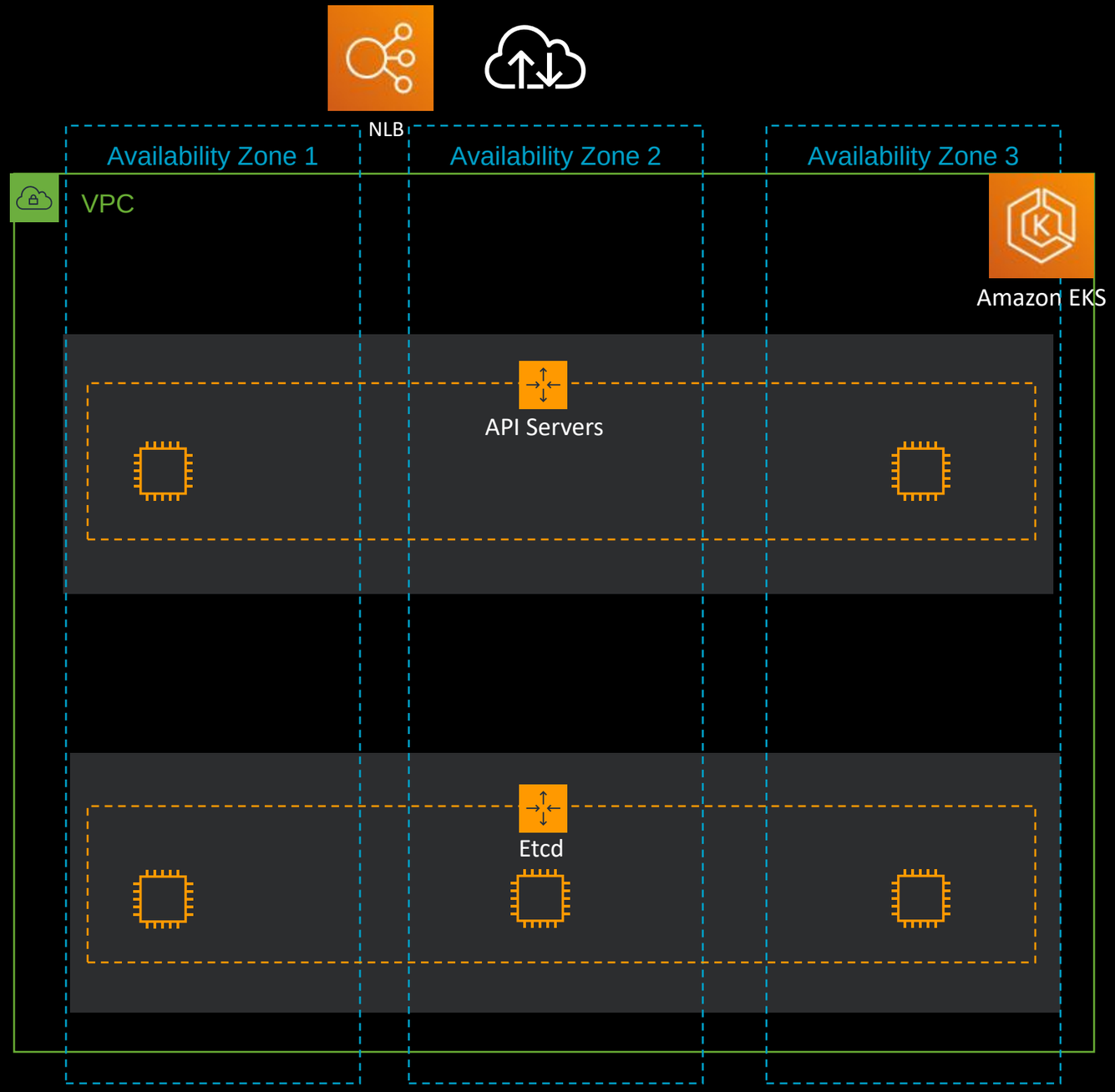


Kubernetes control plane

Highly available and single tenant infrastructure

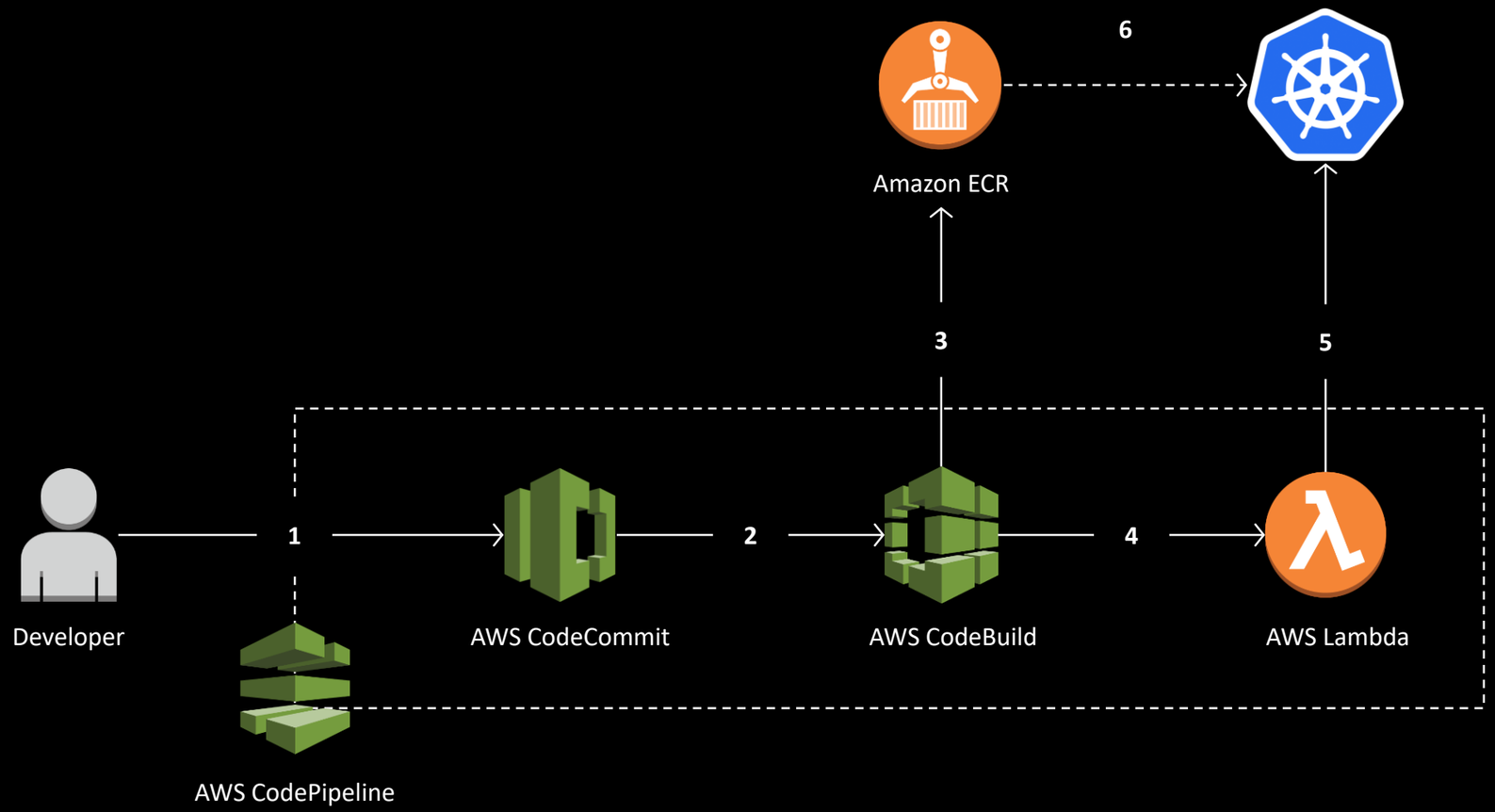
All “native AWS” components

Fronted by a Network Load Balancer



CI/CD

Kubernetes continuous deployment



1. Developers continuously integrate changes into a main branch hosted within a repo
2. Triggers an execution of the pipeline when a new version is found, builds a new image with build id
3. Pushes the newly built image tagged with build id to ECR repo
4. Invokes a Lambda function to trigger application deployment
5. Leverages Kubernetes Python SDK to update a deployment
6. Fetches new container image and performs a rolling update of deployment

Supported CI/CD platforms

- AWS CodeBuild / AWS CodePipeline
- Jenkins
- Spinnikar
- Jfrog
- …any others that work with Kubernetes on AWS!

Infrastructure elasticity

Amazon EKS worker node provisioning with Amazon EC2 Spot

- Recommend using the node labels to identify Amazon EC2 Spot Instances
- Launch Amazon EC2 Spot Instances as part of Auto Scaling group
- Use Amazon EC2 Spot Instances best practice of mixed instance types

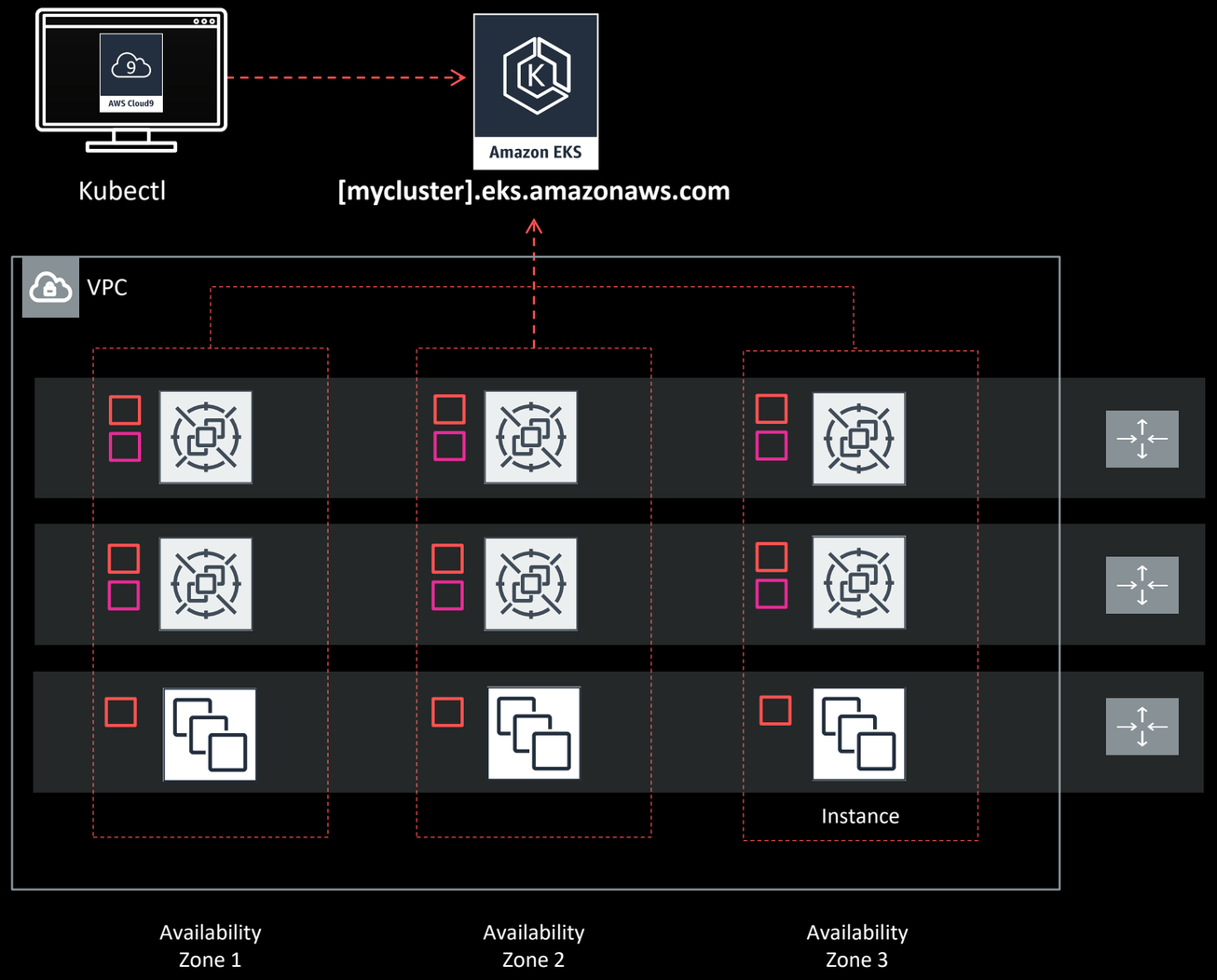
Automatic scaling with Amazon EKS

Two dimensions to scaling

- Amazon EC2 instance through cluster-autoscaler
  → Scale out Amazon EC2 Spot Instances

- Pods through HPA
  → Scale out pods

Amazon EKS support sophisticated and scalable infrastructure



Cluster Autoscaler Daemonset

Spot Interruption handler Daemonset

Auto Scaling group for m4.large Spot Instances

Auto Scaling group for t2.medium Spot Instances

Auto Scaling group for On-Demand Instances

https://eksworkshop.com/spot/managespot/deployhandler

Amazon EKS is ready for sensitive and regulated workloads

HIPAA-eligible

ISO 9001, 27001, 27017, 27018

PCI DSS

SOC 1,2,3

Security

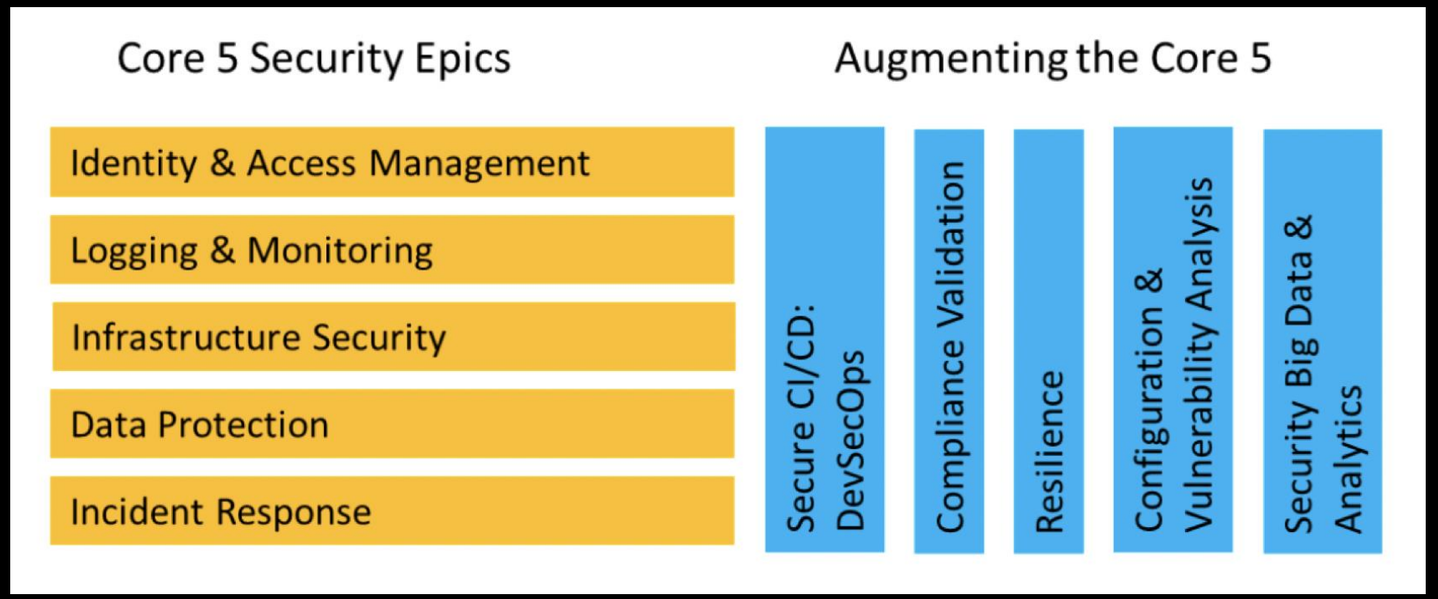
aws SUMMIT

AWS container security: principled and peculiar

Tenets

- All-encompassing
- Shared responsibility
- Cloud native

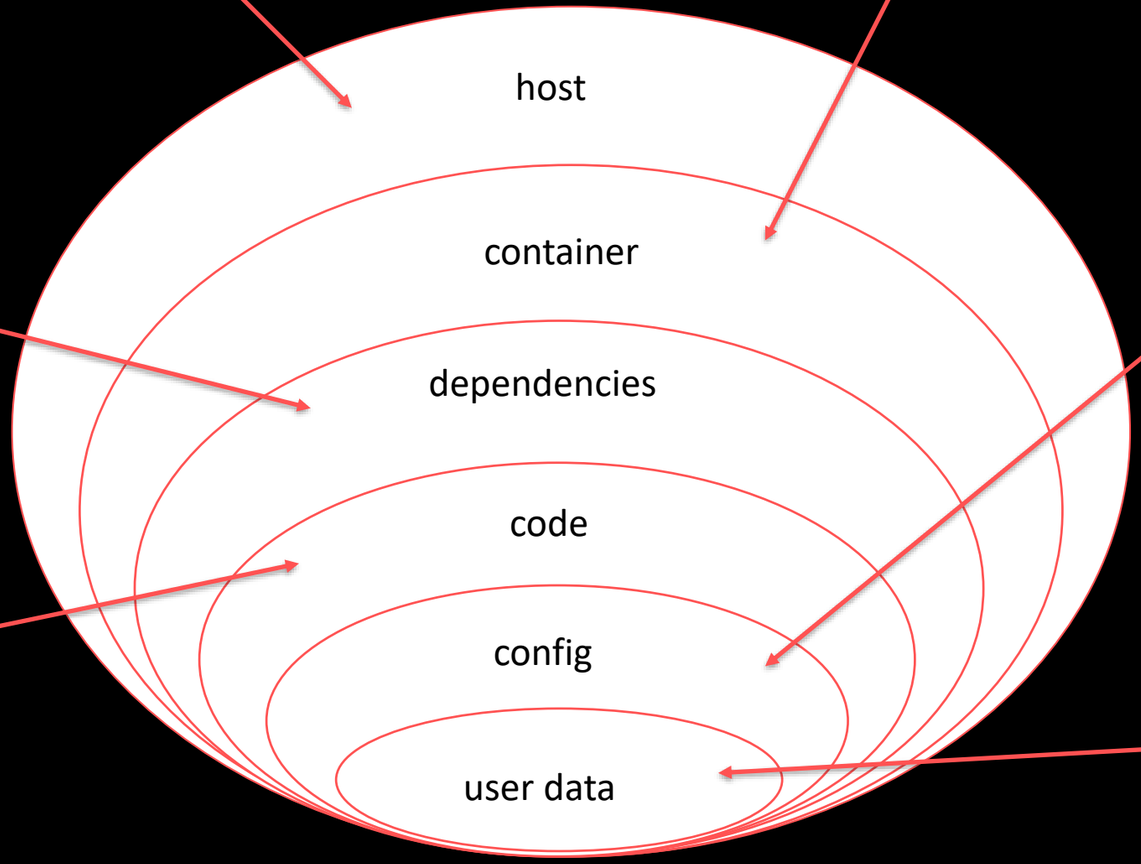
Epics



Container security onion model: Defense in depth

- full blown distro (Ubuntu, AL) vs. minimal environment (container-optimized distribution)
- multi-tenancy requirements
- **gotchas**: Linux packages/CVEs, leaks, GDPR (in Europe)

- runtime/standards (OCI)
- immutability of images
- all containers share a kernel (mitigation: Firecracker)
- **gotchas**: unnecessary privileged users, no scans, trust



- code analysis
- source available?
- **gotchas**: big surface, many languages

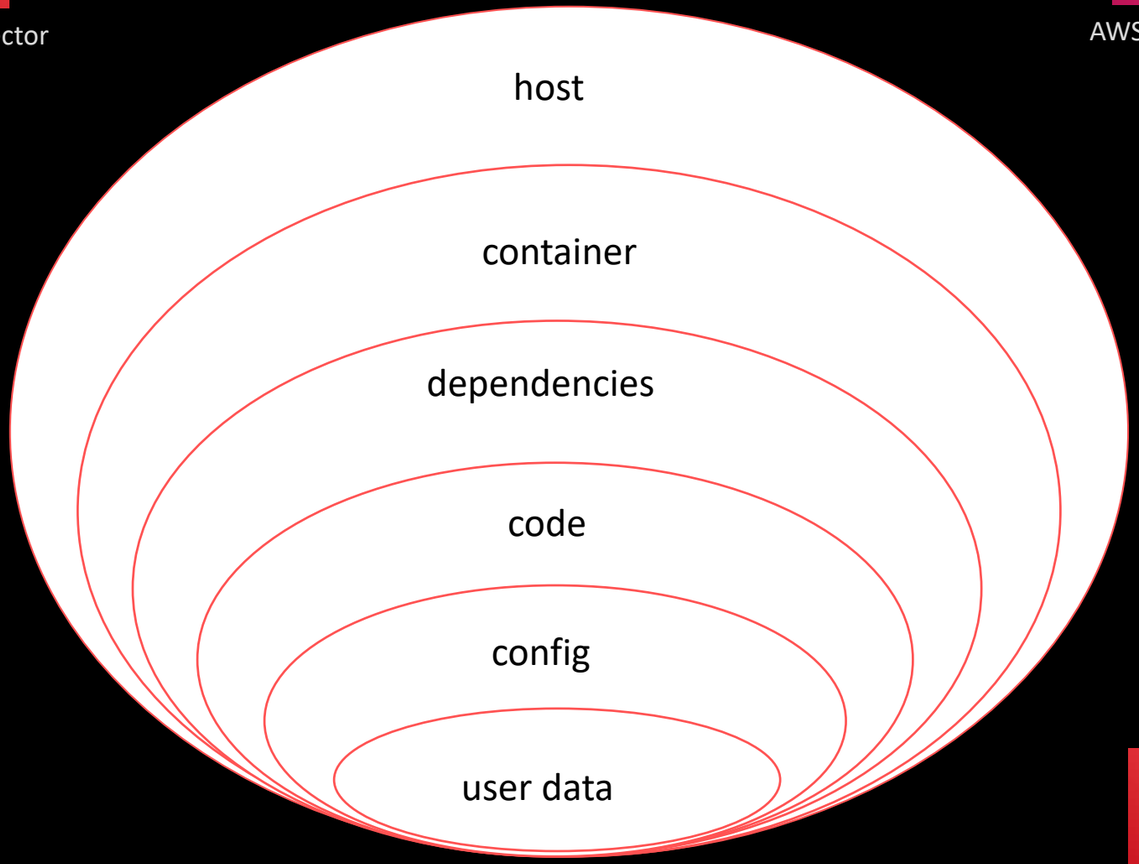
- sensitive config (passwords, API keys, etc.)
- **gotchas**: commits-to-source, non-separated access (dev has cleartext password)

- sanitizing user input
- static code analysis
- **gotchas**: log-leaking

- business core data
- Personal Identifiable Information (PII)
- **gotchas**: leaks, GDPR (in Europe)

aws SUMMIT

Security tooling



AWS Certificate Manager

AWS Security Hub

AWS IAM

Amazon Inspector

AWS CloudTrail

Amazon GuardDuty

AWS WAF

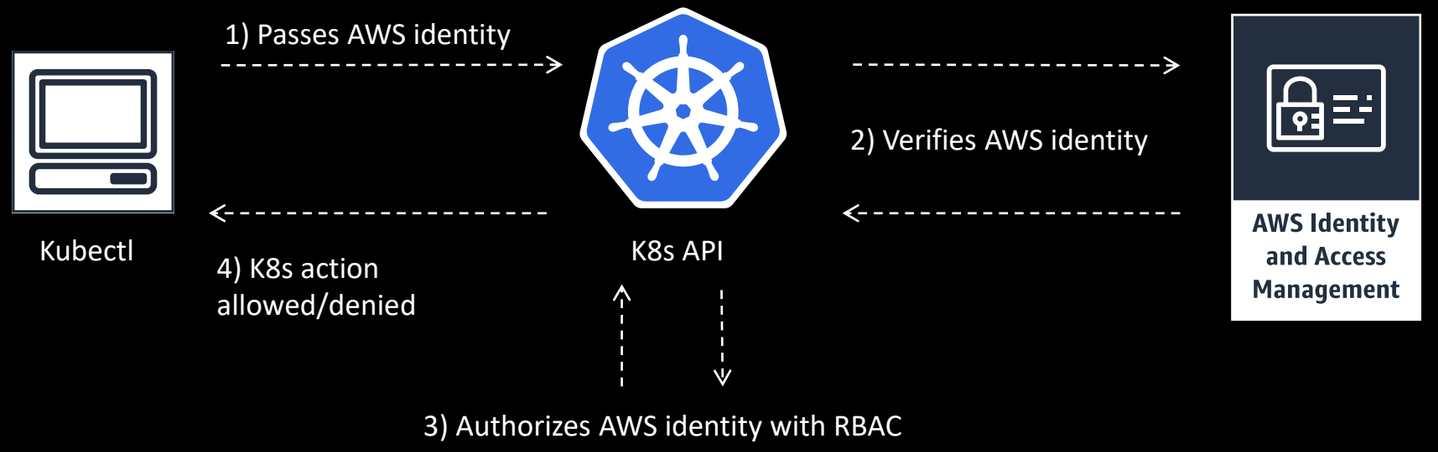
AWS KMS

AWS Secrets Manager

AWS CloudHSM

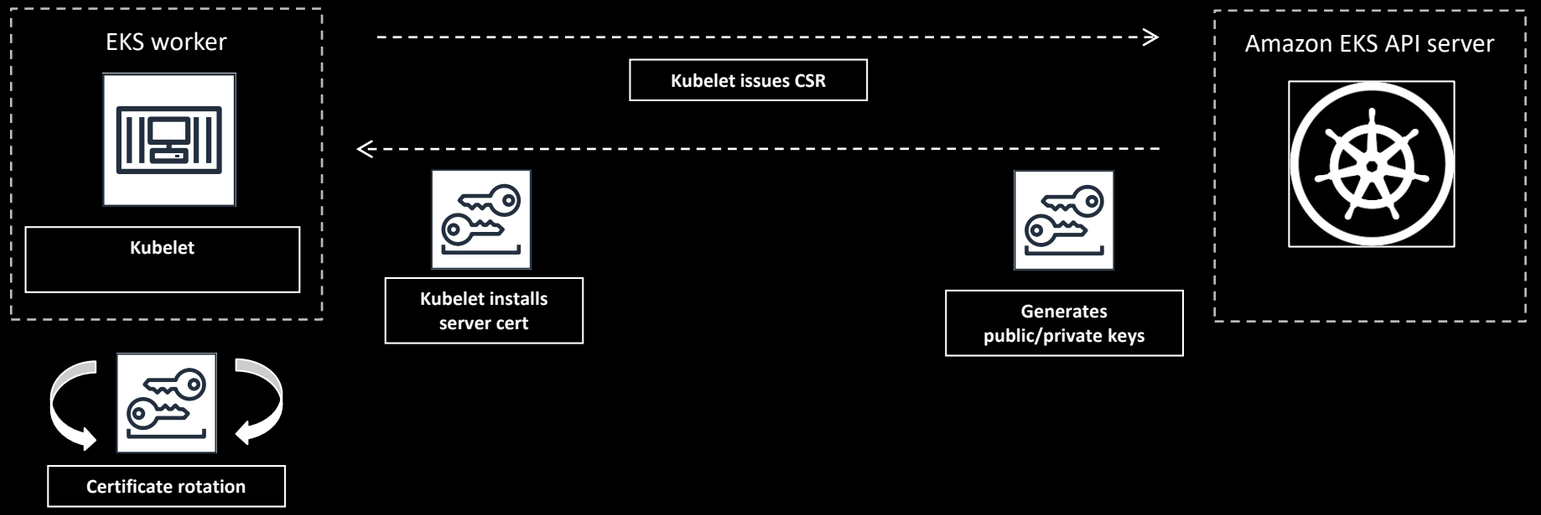
Amazon Macie

AWS Identity and Access Management (IAM)
Authentication



PKI configuration

Each Amazon EKS cluster is a unique CA



IAM for pods
Roadmap

Set IAM access permissions at the pod level

Enables multiple applications with different permission sets to share the same nodes

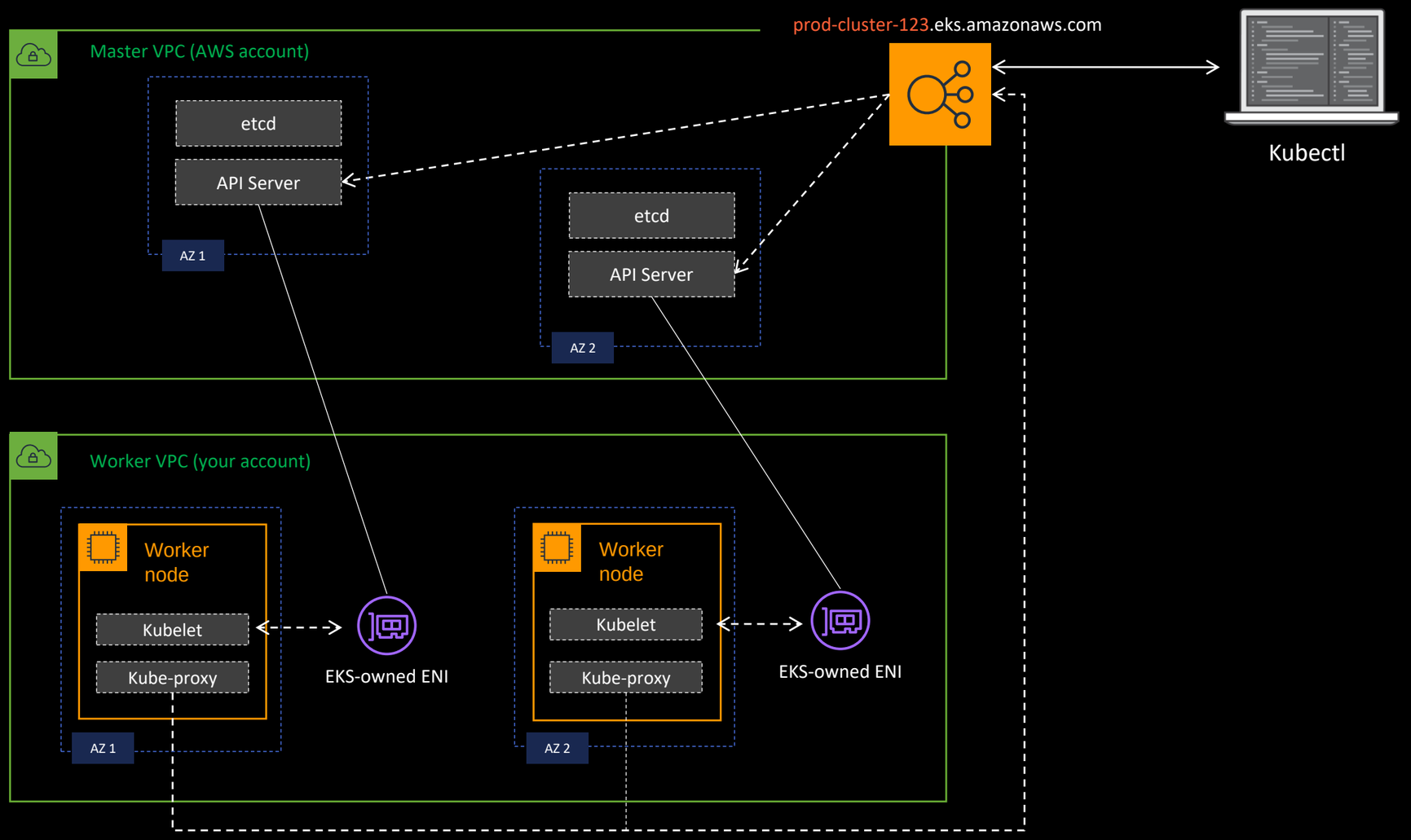
Built using Kubernetes primitives, minimal user configuration

https://github.com/aws/containers-roadmap/issues/23

API-server endpoint access control

Public == true

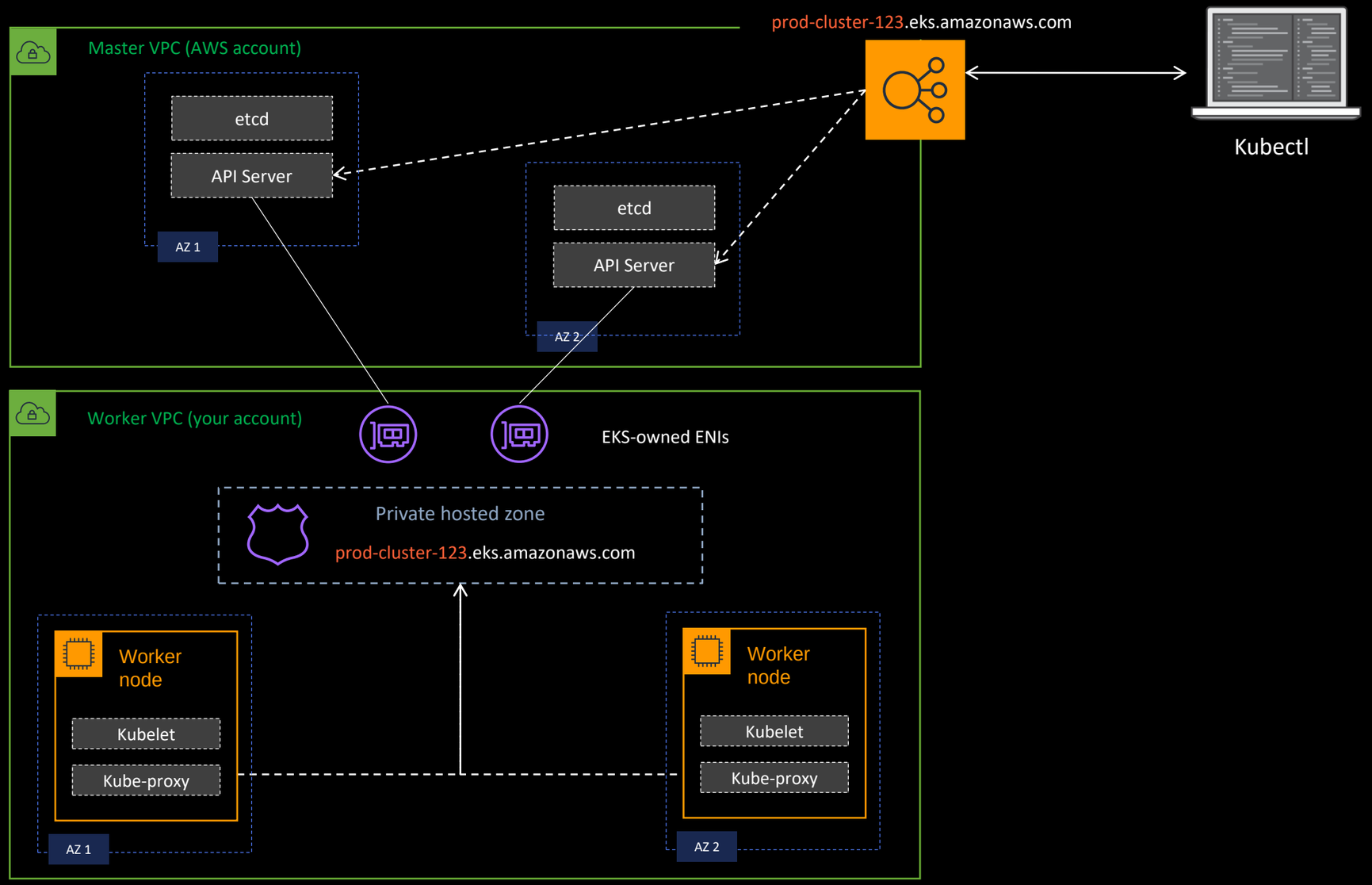
Private == false



API-server endpoint access control

Public == true

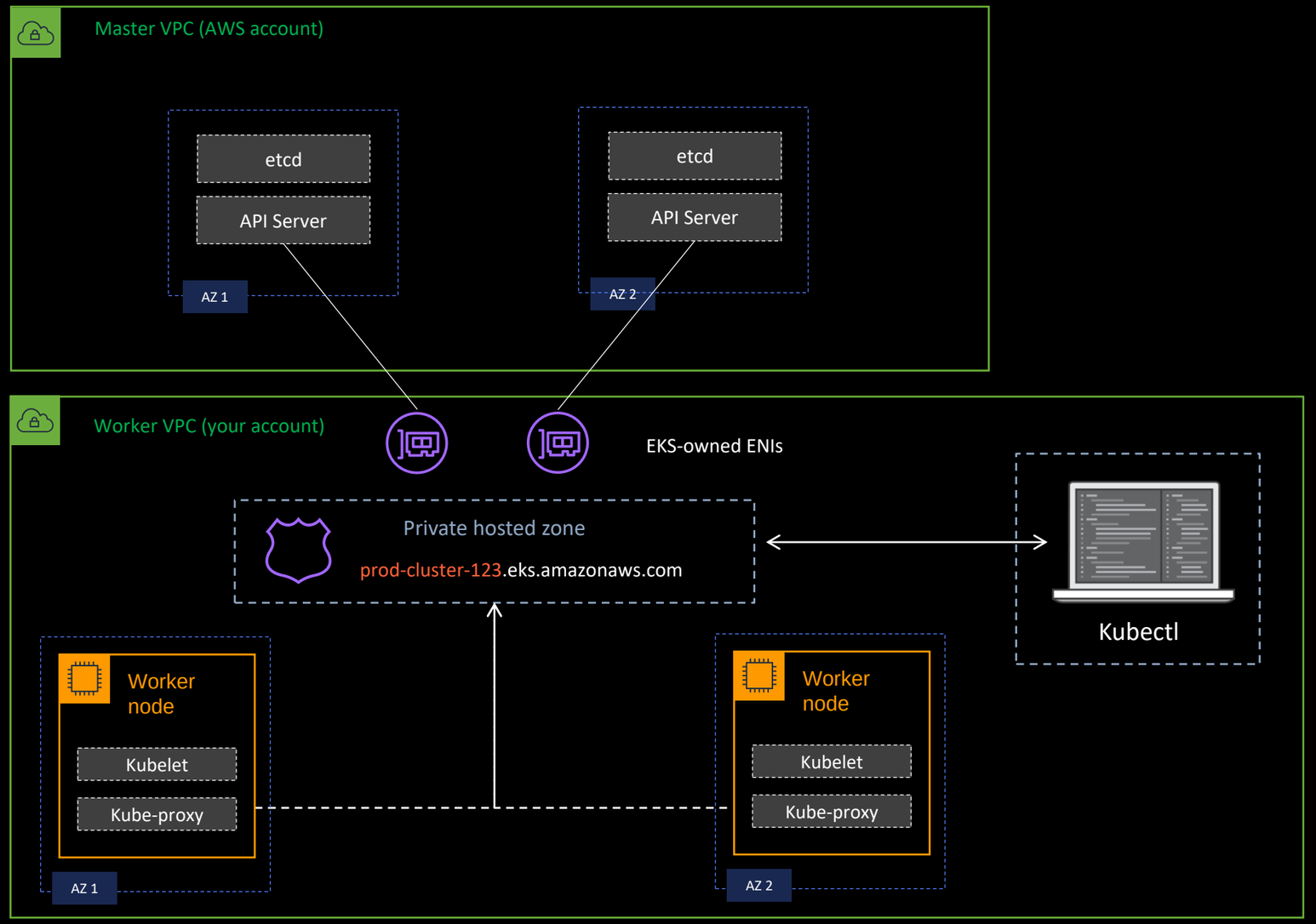
Private == true



API-server endpoint access control

Public == false

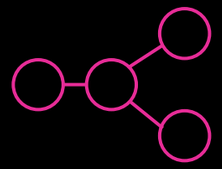
Private == true



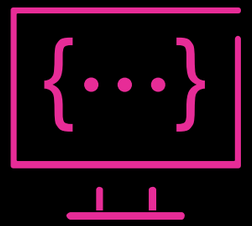
Network support

aws SUMMIT

Amazon VPC CNI plugin

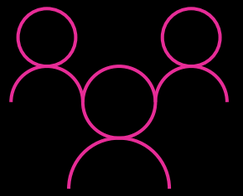


Native VPC networking with CNI plugin



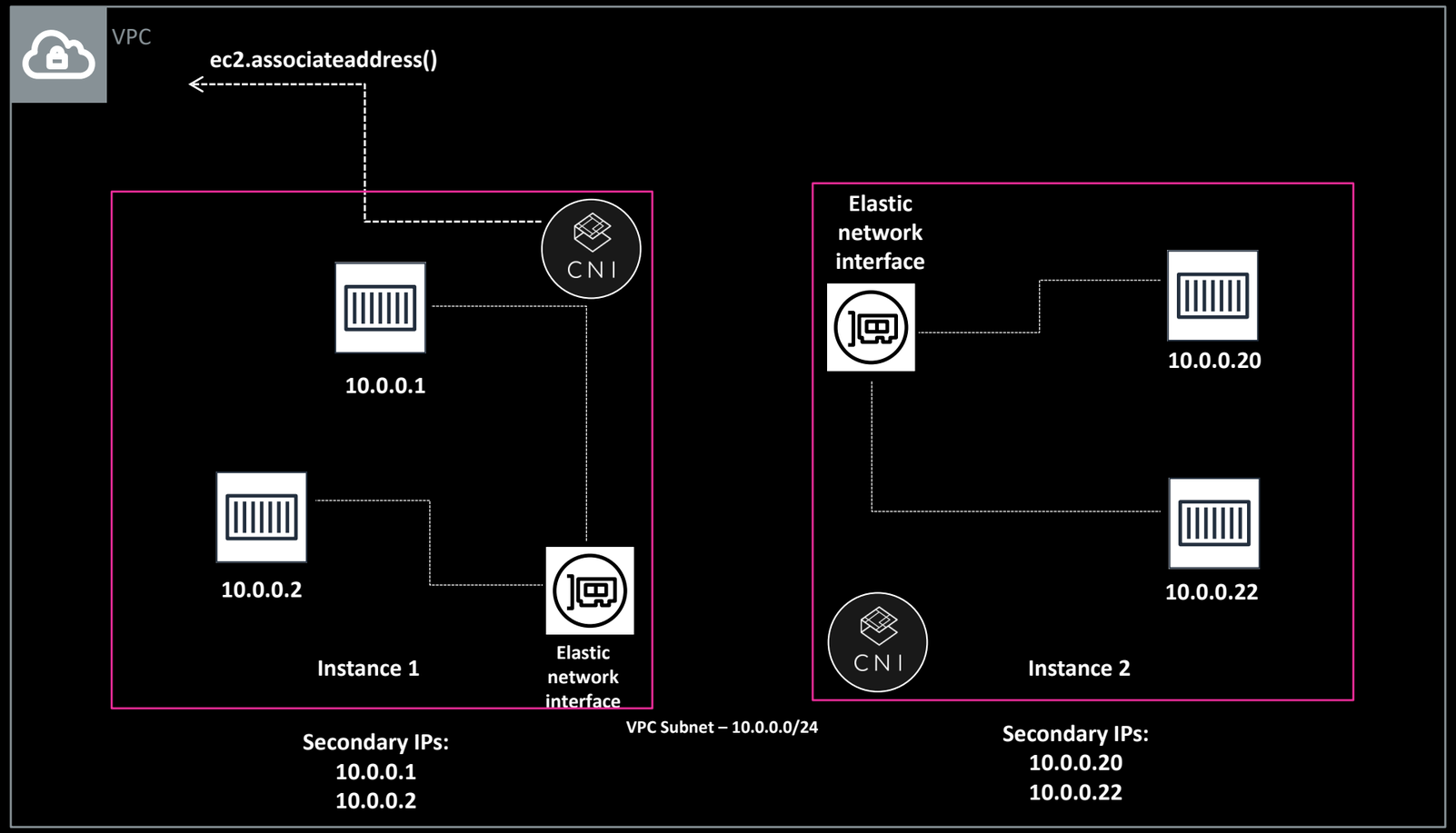
Pods have the same VPC address inside the pod as on the VPC

Simple, secure networking



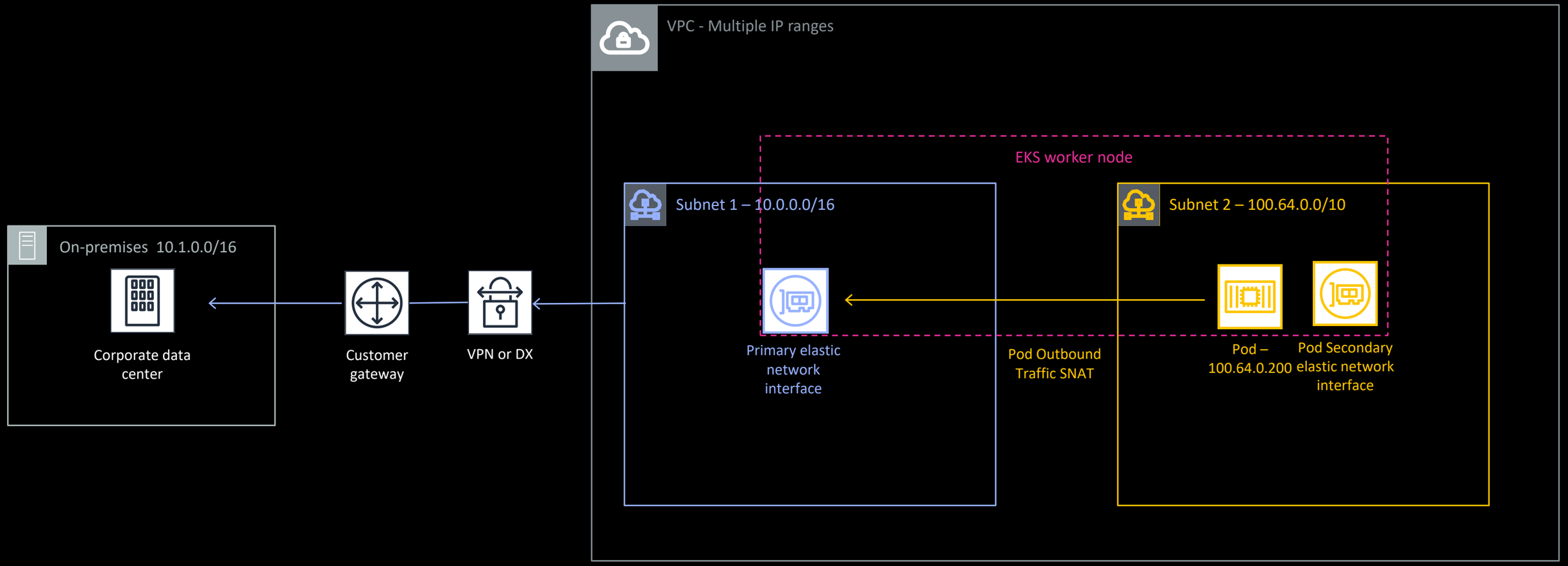
Open source and on GitHub

Amazon VPC CNI plugin



https://github.com/aws/amazon-vpc-cni-k8s

Amazon EKS supports advanced networking architectures



Load balancing

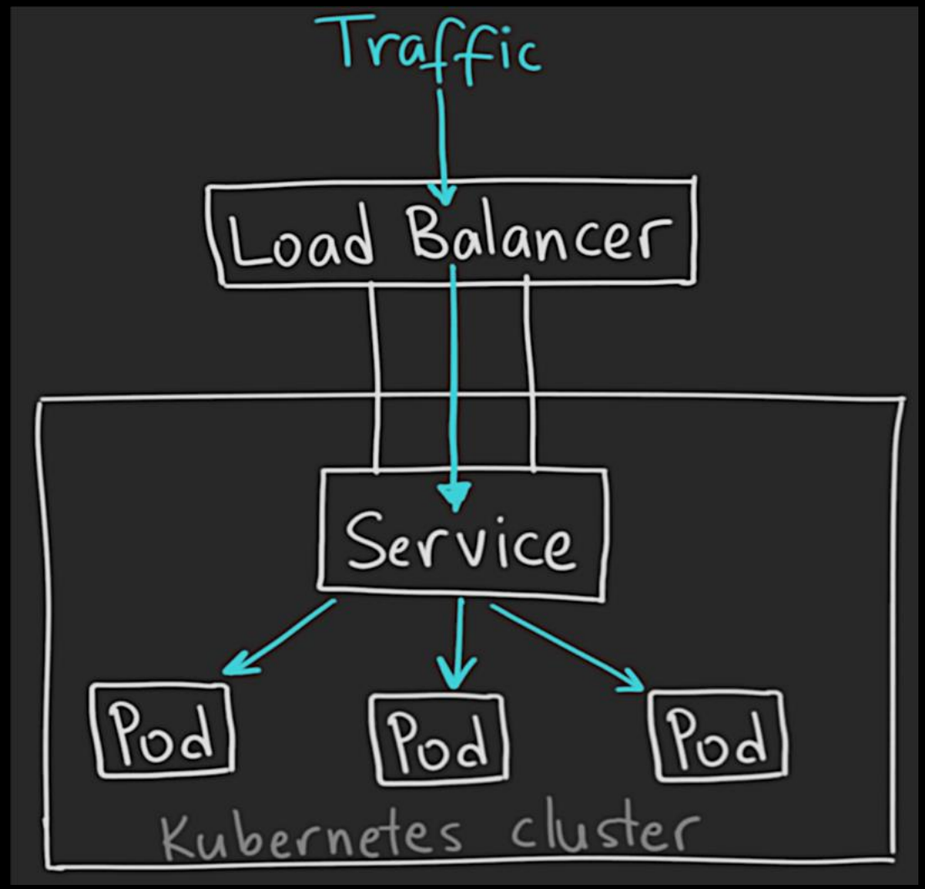
All three Elastic Load Balancing products are supported

NLB and CLB supported by Kubernetes Service
`type=LoadBalancer`

Internal and External Load Balancer support

Kubernetes ServiceType: LoadBalancer

- Exposes the service **externally** using a cloud provider's load balancer
- NodePort and ClusterIP services (to which LB will route) automatically created
- Each service exposed with a LoadBalancer (ELB or NLB) will get its own IP address
- Exposes L4 (TCP) or L7 (HTTP) services



Load balancing

Want to use an Internal Load Balancer? Use annotation:

```
service.beta.kubernetes.io/aws-load-balancer-internal: 0.0.0.0/0
```

Want to use an NLB? Use annotation:

```
service.beta.kubernetes.io/aws-load-balancer-type: nlb
```

Service load balancer: Network Load Balancer

```
apiVersion: v1
kind: Service
metadata:
        name: nginx
        namespace: default
        labels:
                app: nginx
        annotations:
                service.beta.kubernetes.io/aws-load-balancer-type: "nlb"
spec:
        externalTrafficPolicy: Local
        ports:
        - name: http
          port: 80
          protocol: TCP
          targetPort: 80
        selector:
          app: nginx
type: LoadBalancer
```

Network Load Balancer support on AWS [alpha]

> **Warning:** This is an alpha feature and not recommended for production clusters yet.

Service load balancer: Network Load Balancer (NLB)

- NLB supports forwarding the client's IP through to the node

  .spec.externalTrafficPolicy = Local ➔ client ip passed to pod

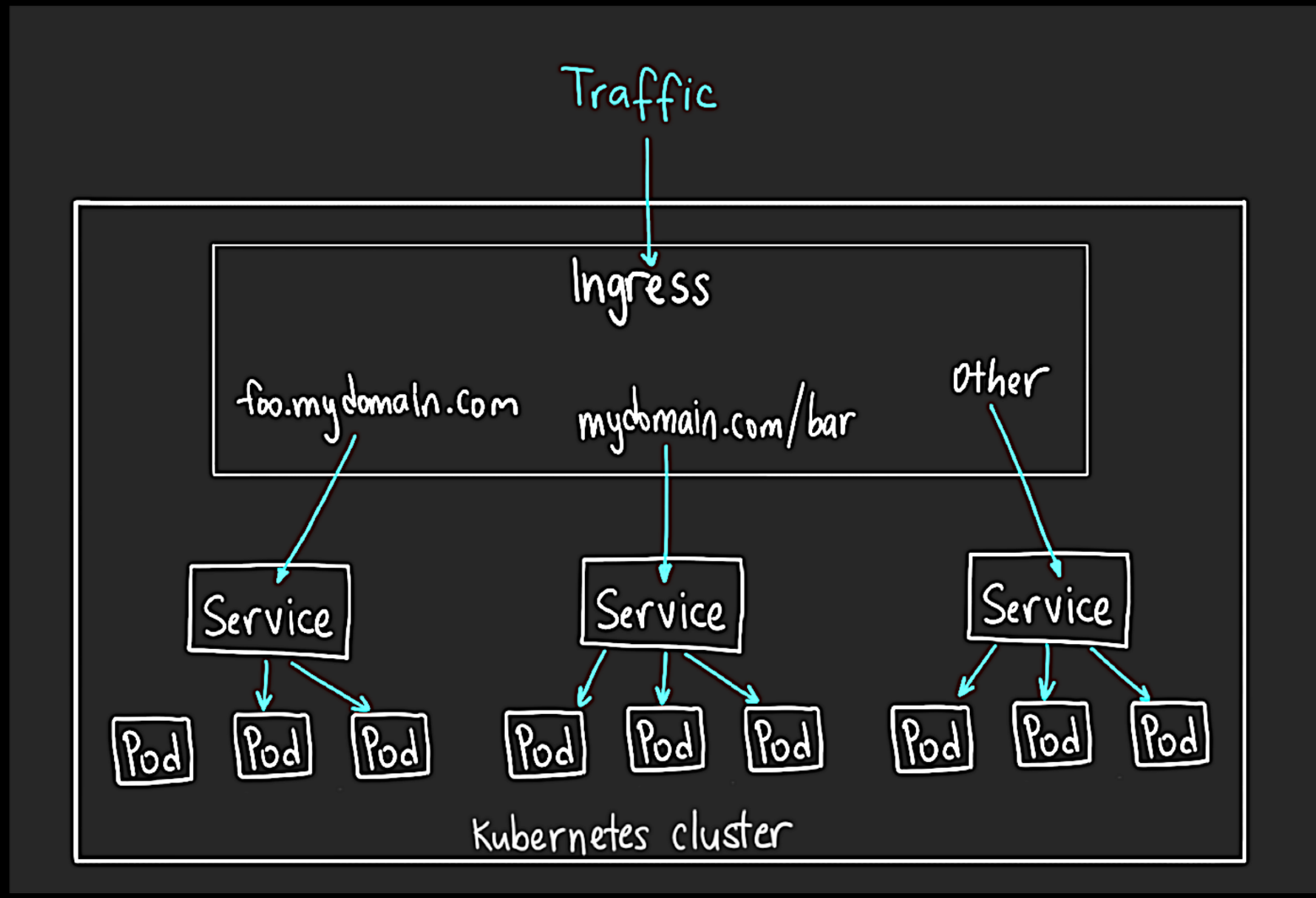
  - Nodes with no matching pods will be removed by specified NLB's health check

    .spec.healthCheckNodePort

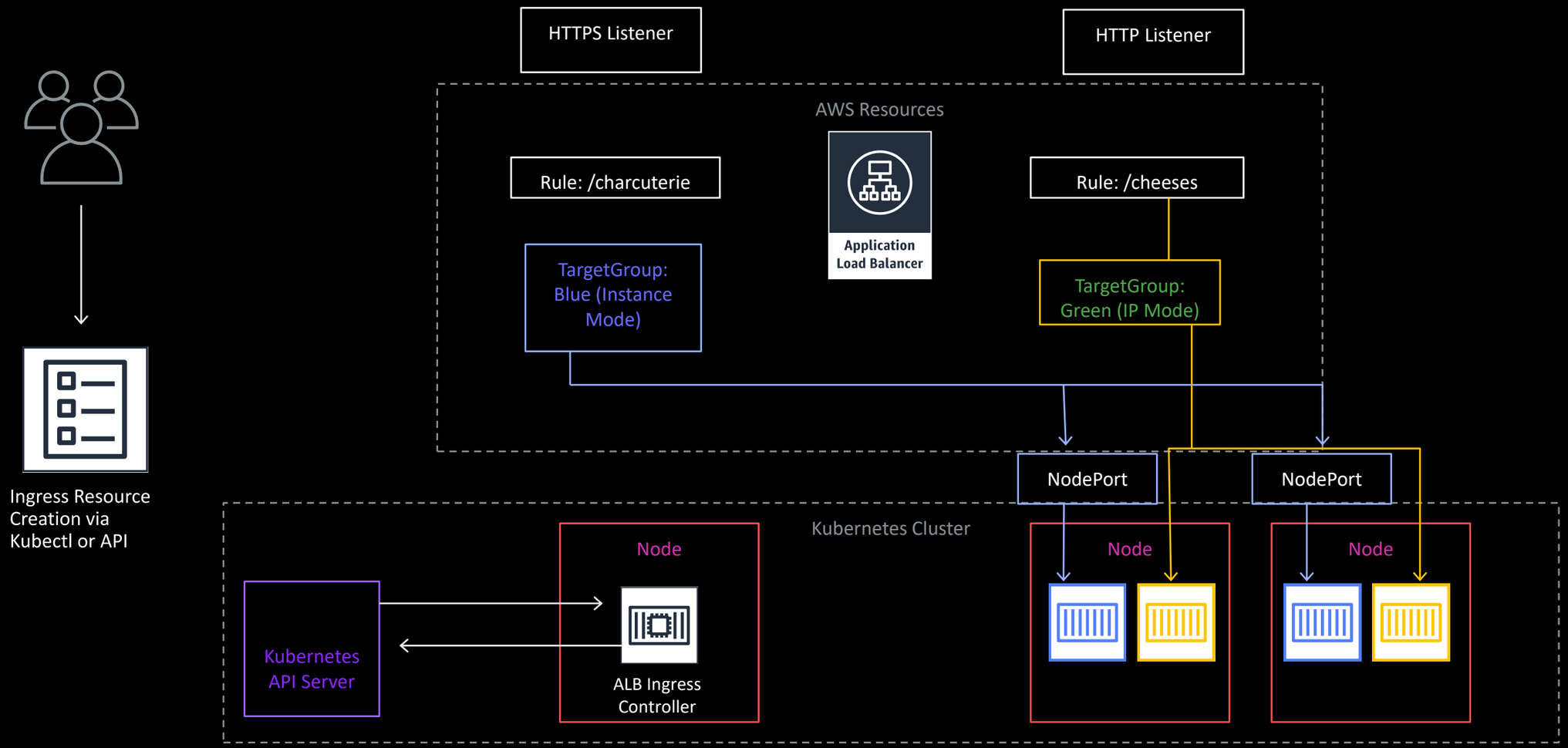
  - Use **DaemonSet** or **pod anti-affinity** to verify even traffic split

Kubernetes Ingress object

- exposes HTTP/HTTPS routes to services within the cluster
- Many implementations: ALB, NGINX, F5, HAProxy etc.
- Default service type: ClusterIP



ALB Ingress controller



ALB Ingress controller

Production-ready 1.0 release

Supported by Amazon EKS team

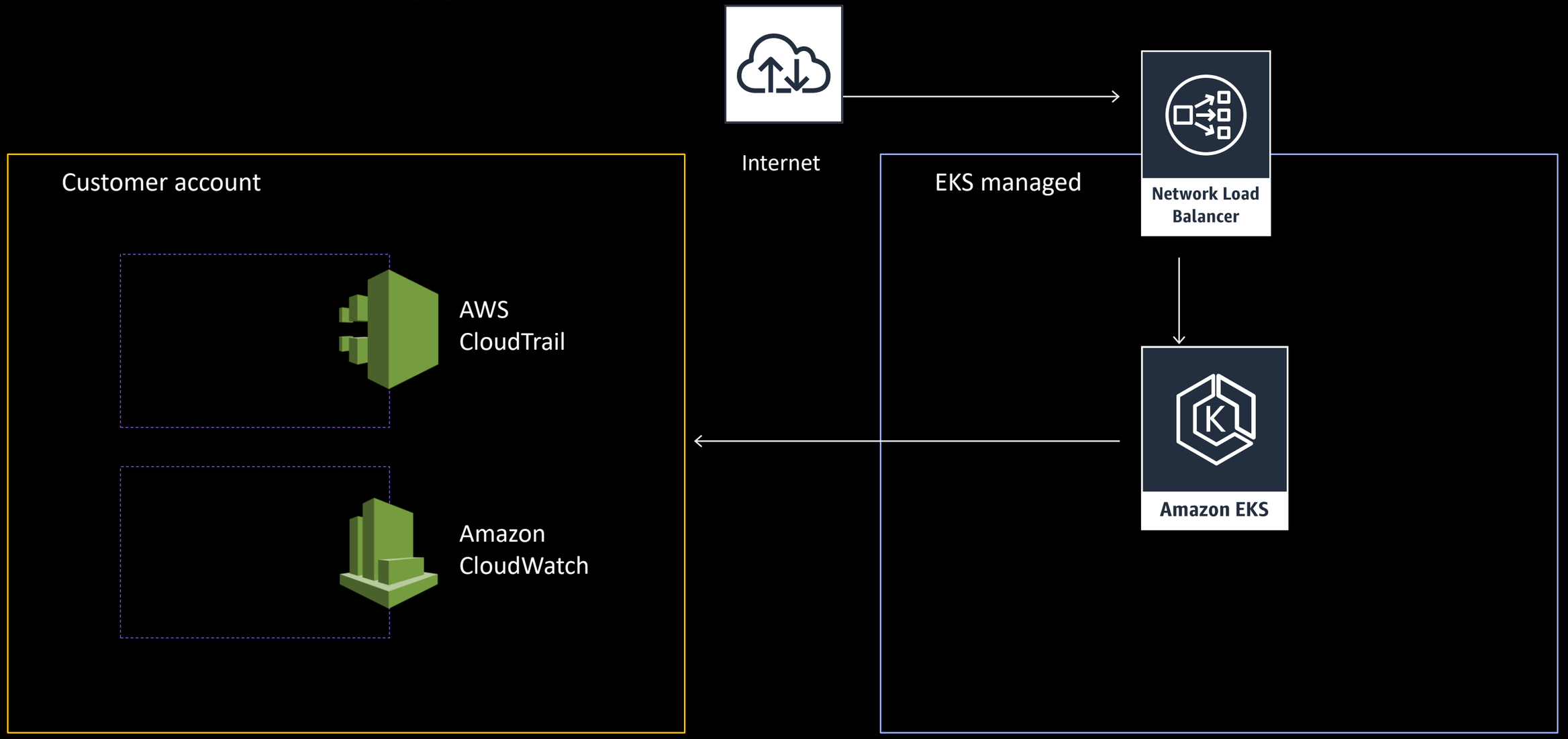
Open source development: https://github.com/kubernetes-sigs/aws-alb-ingress-controller

Customers are using it in production today!

Logging

aws SUMMIT

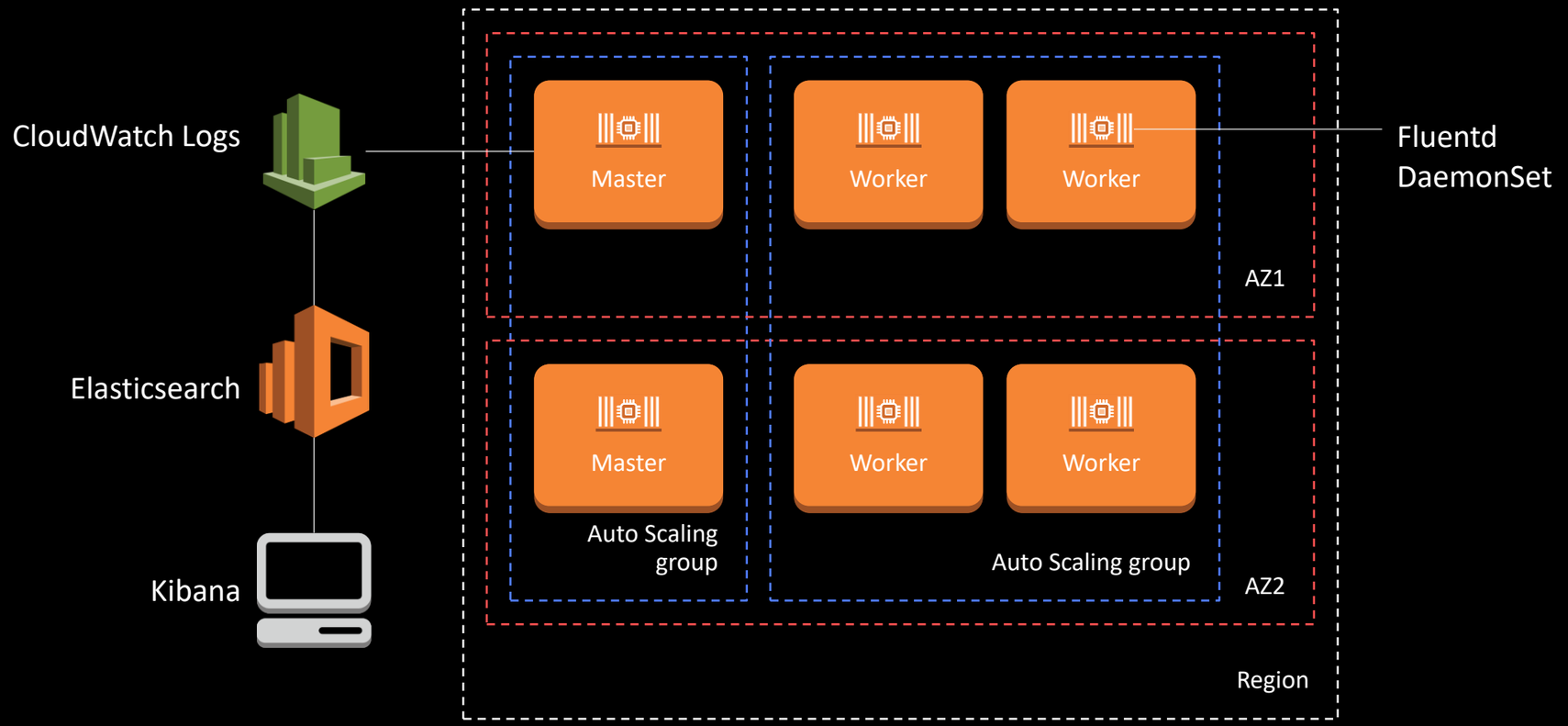
Amazon EKS logging



Amazon EKS logging

Kubectl logs

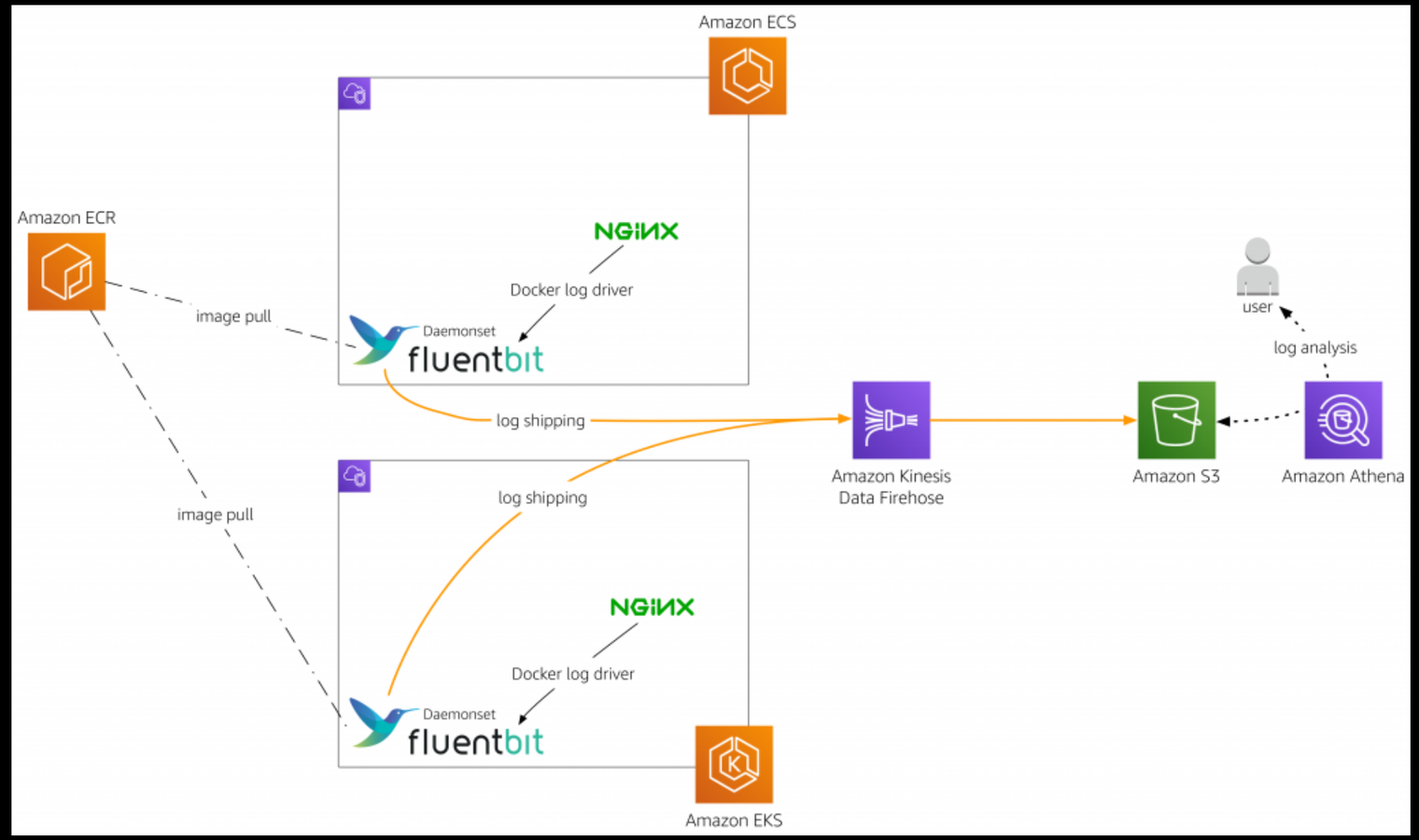
Elasticsearch (index),
Fluentd (store), and
Kibana (visualize)



NEW!

Logging with FluentBit

- New AWS FluentBit container plugin
- Optimize costs. Route logs from Amazon EKS and Amazon ECS clusters directly to Amazon S3 and query with Amazon Athena
- Open source
- More resource-efficient than Fluentd. Tests show Fluentd uses 4x more CPU and 6x more memory



https://aws.amazon.com/blogs/opensource/centralized-container-logging-fluent-bit/

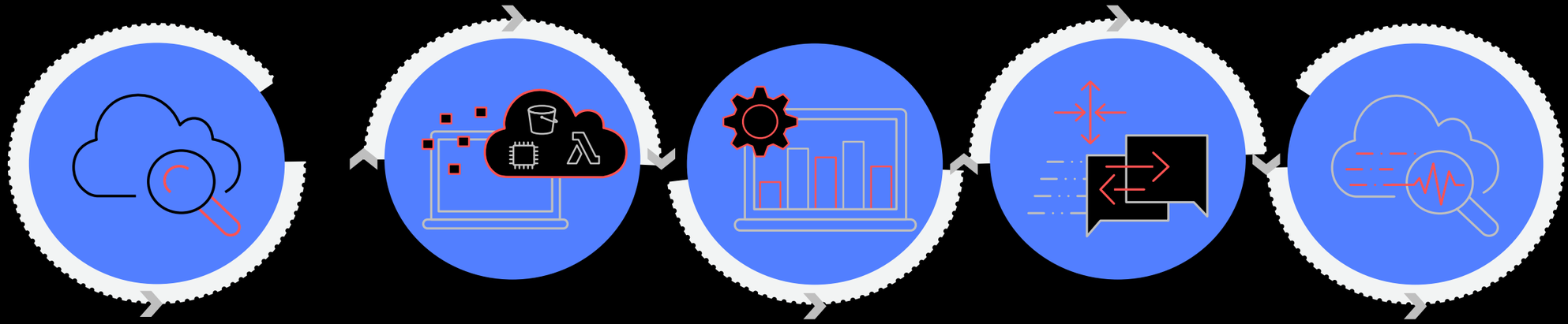
Monitoring

aws SUMMIT

NEW!

Amazon CloudWatch Container Insights

Gives you complete visibility into your cloud resources and applications so you can monitor, troubleshoot, and remediate issues



Amazon CloudWatch

Collect logs & metrics

Monitor, troubleshoot & set alarms

Act

Analyze

NEW!

CloudWatch Container Insights

A fully managed observability service for monitoring, troubleshooting, and alarming on your containerized applications and microservices

- ✓ Collects, aggregates, and summarizes
- ✓ Reliable, secure metrics and logs collection
- ✓ Automated dashboards and analysis
- ✓ Observability experience across metrics, logs, traces
- ✓ Ad hoc analytics

NEW!

Images on DockerHub

Performance Metrics – CloudWatch Agent:
https://hub.docker.com/r/amazon/cloudwatch-agent

- Tag: latest

Logs – FluentBit:

https://hub.docker.com/r/amazon/aws-for-fluent-bit

- Tag: latest

Logs – Fluentd:

https://hub.docker.com/r/fluent/fluentd-kubernetes-daemonset

- Tag: v1.3.3-debian-cloudwatch-1.4

NEW!

Container Insights available now

1. **Fully managed**, AWS native observability service providing automated summary and analysis of compute capacity

2. **Reliable and secure collection** of application logs with built-in analytics capabilities

3. **Prebuilt visualization** to summarize cluster and node errors

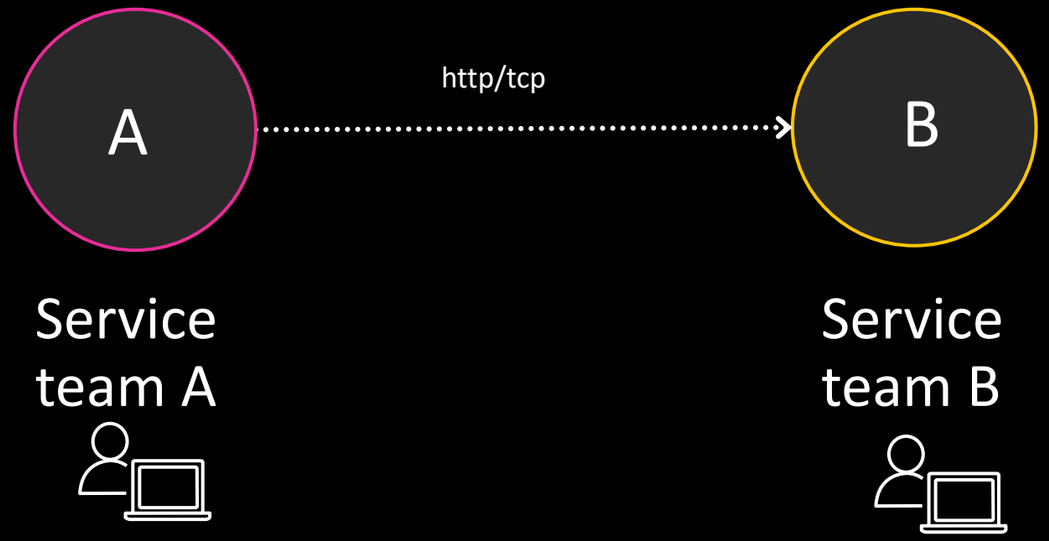
4. **Application & microservice tracing** - Troubleshoot and debug application & microservice

Application communication

aws SUMMIT

Why AWS App Mesh?

Common need: Manage interservice traffic



- How to observe (logs, metrics, traces)
- How to load balance E/W traffic
- How to shift traffic between deployments
- How to decouple service teams
- How to minimize impact to app code

App Mesh uses Envoy proxy

OSS community project

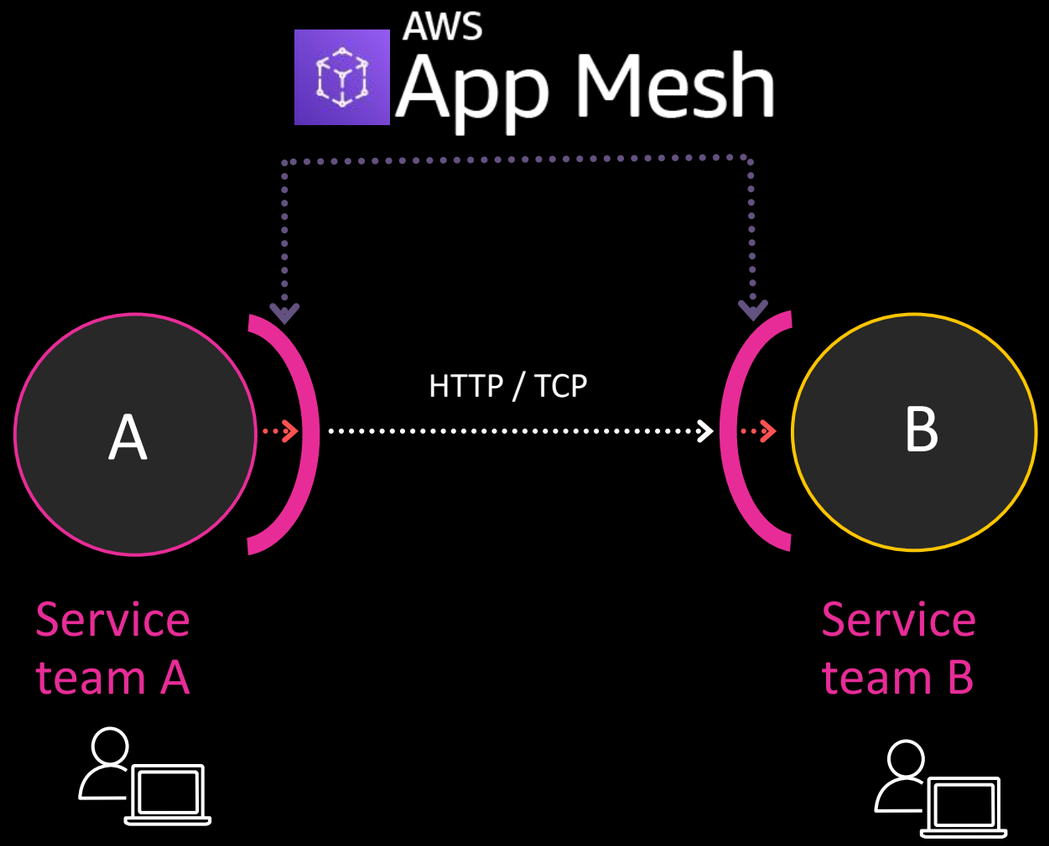
Wide community support, numerous integrations

Stable and production-proven

Graduated Project in Cloud Native Computing Foundation

Started at Lyft in 2016

Why App Mesh?



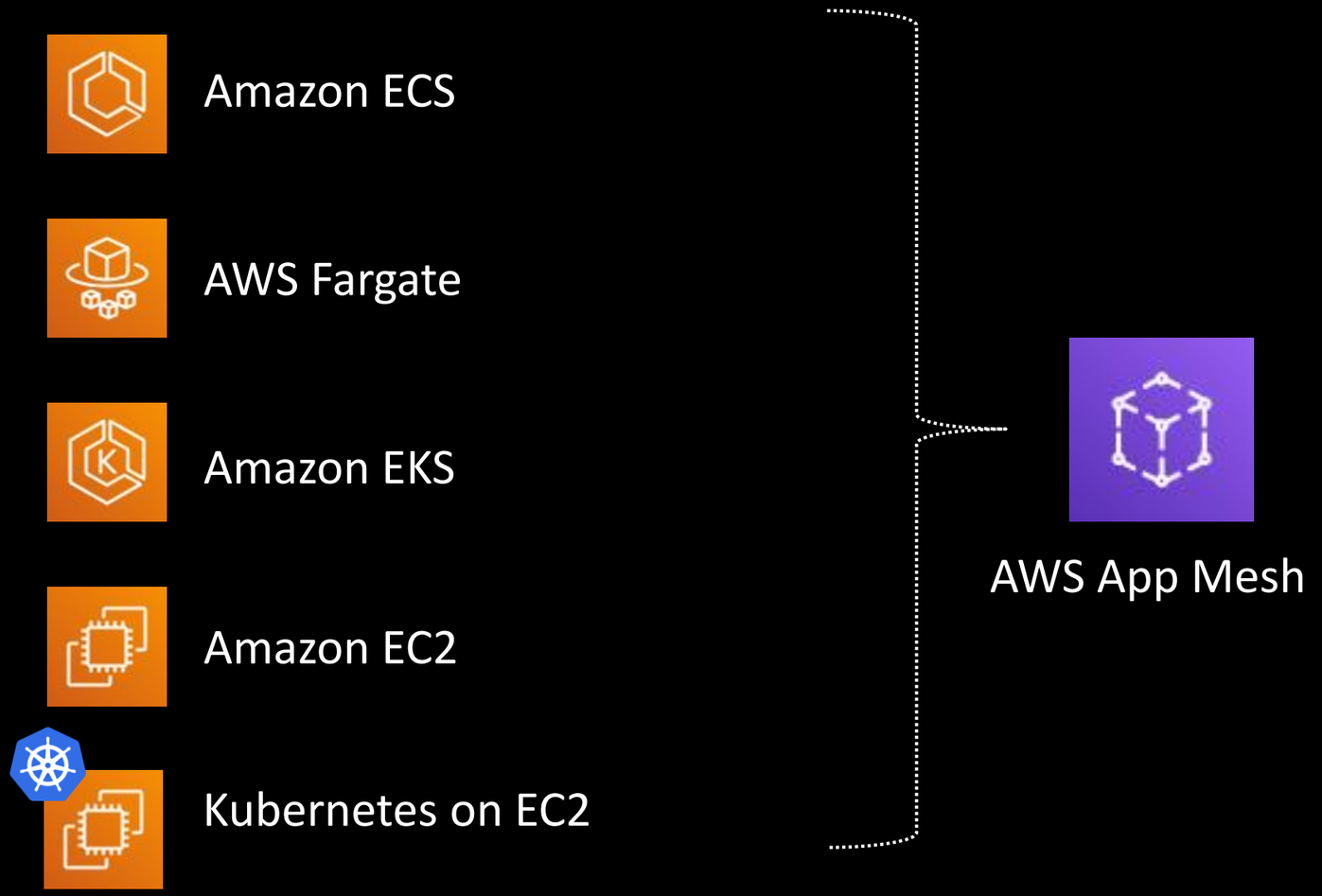
Control plane

Translates logical intent to proxy config
Distributes proxy config

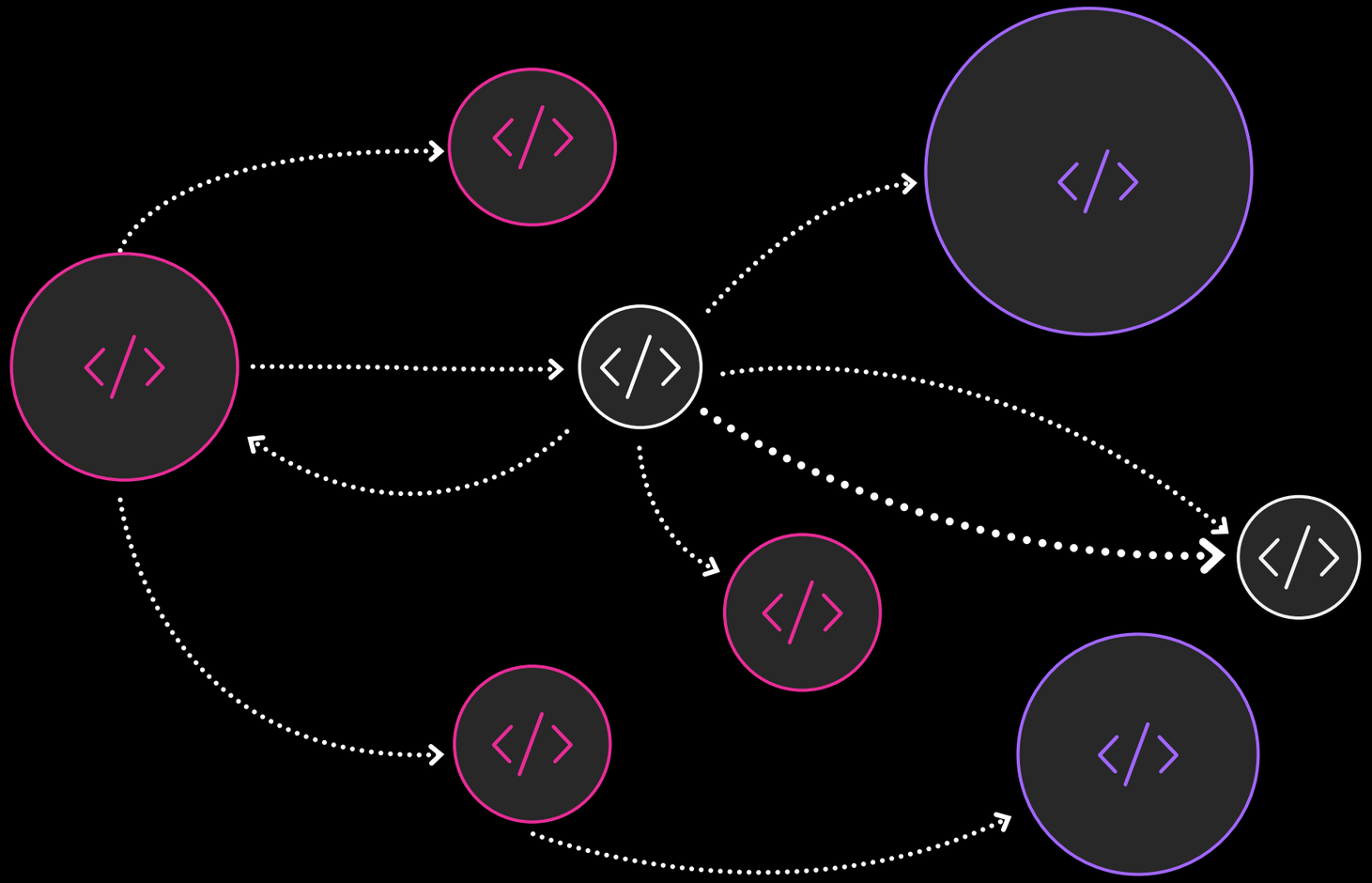
Proxy

Sits between all services
Manages and observes traffic

App Mesh: App-level communication across AWS



App Mesh: Application observability



Logging

HTTP access logging

Amazon CloudWatch Logs

Available as container logs on Amazon ECS, Amazon EKS, AWS Fargate

Metrics

CloudWatch metrics

StatsD (with tags)

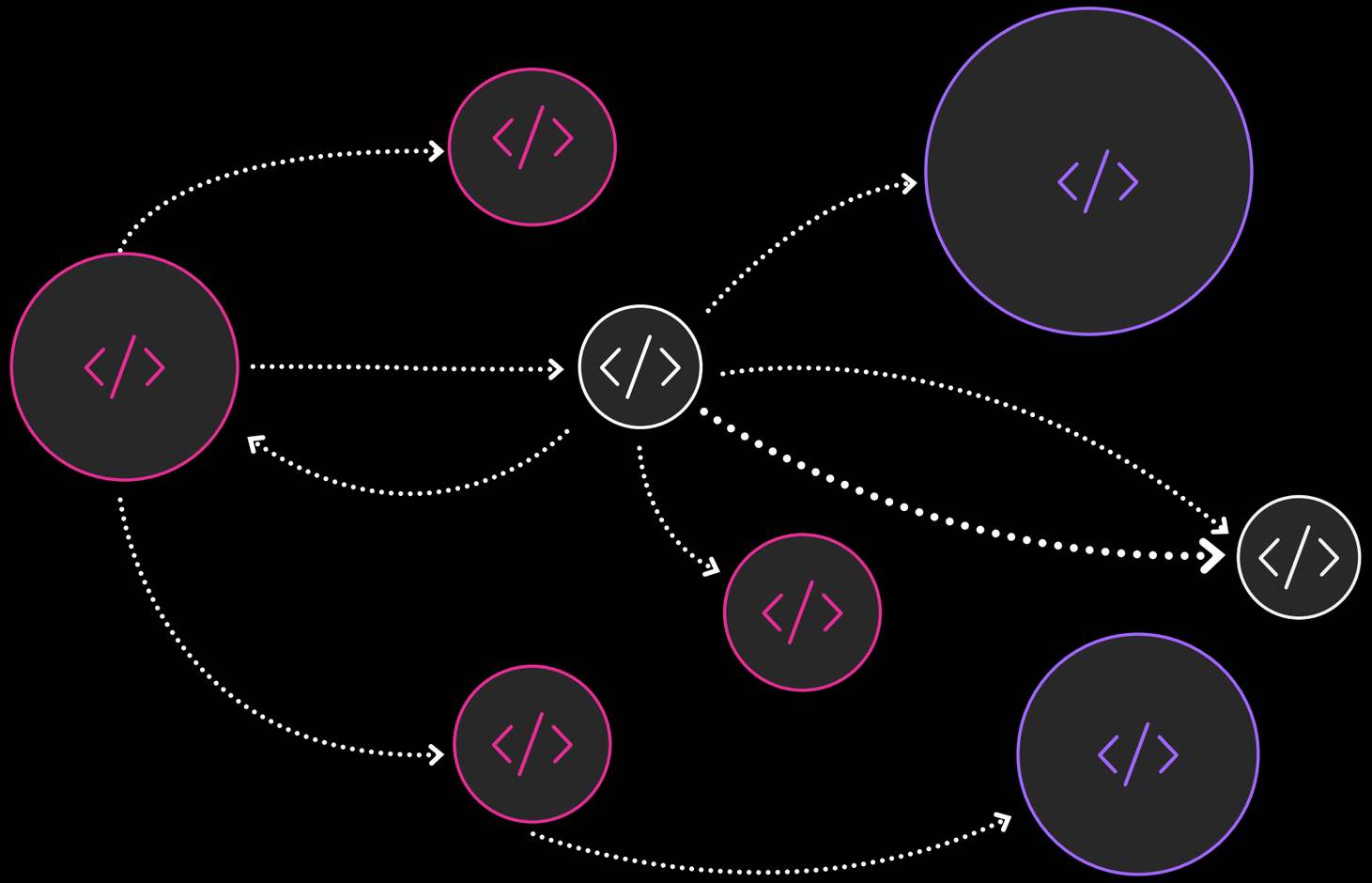
Prometheus

Tracing

AWS X-Ray

Other Envoy tracing drivers

App Mesh: Client-side traffic management



Traffic shaping

Load balancing

Weight targets

Service discovery (DNS + AWS Cloud Map)

Health checks

*Retries**

*Timeouts**

*Circuit breakers**

Routing controls

Protocols support (HTTP, TCP, gRPC*)

Path-based

*Header-based**

*Cookie-based**

*Host-based**

aws SUMMIT

**coming soon*

Questions

aws SUMMIT

Thank you!

aws SUMMIT