云原生应用中的网络流量管理

Walkley He, Solutions Architect, AWS

Nov 2019

aws

议题

- 概述
- 集群外网络流量管理
- 集群内网络流量管理
- Demo
- Q&A

aws

云原生

云原生应用

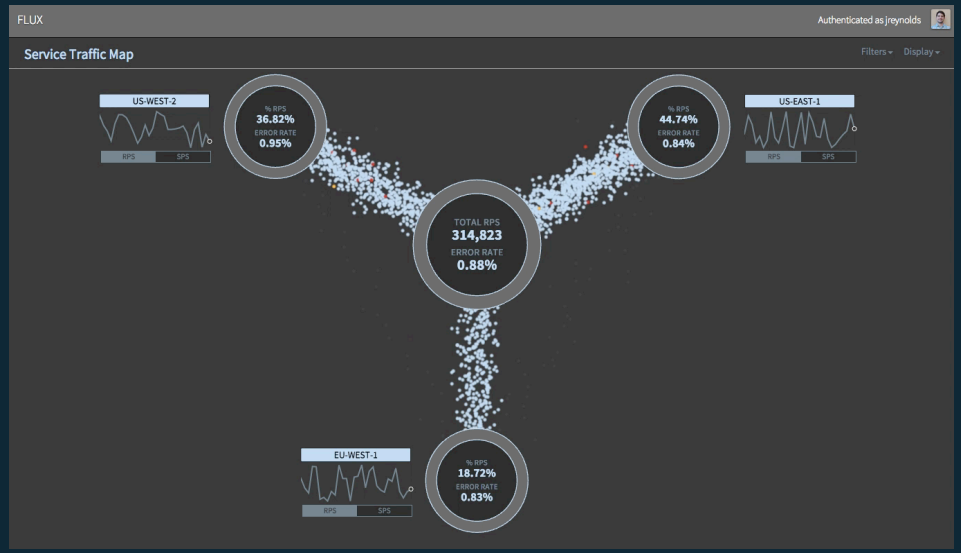
云原生技术有利于各组织在公有云、私有云和混合云等新型动态环境中，构建和运行可弹性扩展的应用。云原生的代表技术包括容器、服务网格、微服务、不可变基础设施和声明式API。

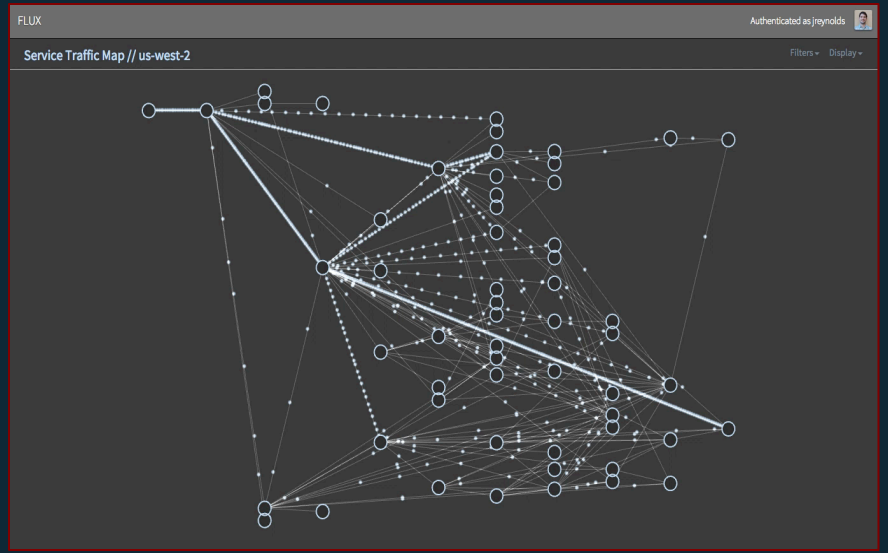
这些技术能够构建容错性好、易于管理和便于观察的松耦合系统。结合可靠的自动化手段，云原生技术使工程师能够轻松地对系统作出频繁和可预测的重大变更。

应用层网络流量管理

- 分发
- 限流
- 熔断
- 重试
- 加密

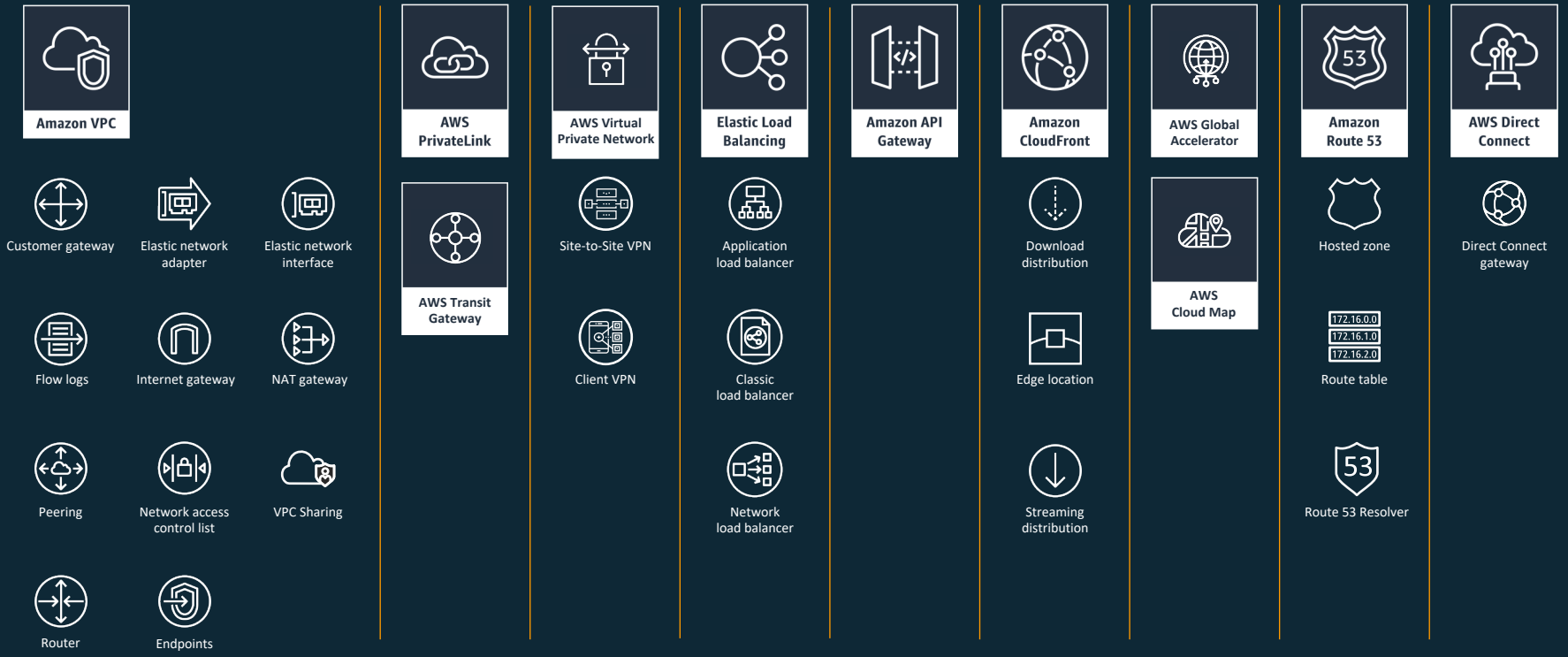
微服务带来的网络管理复杂性



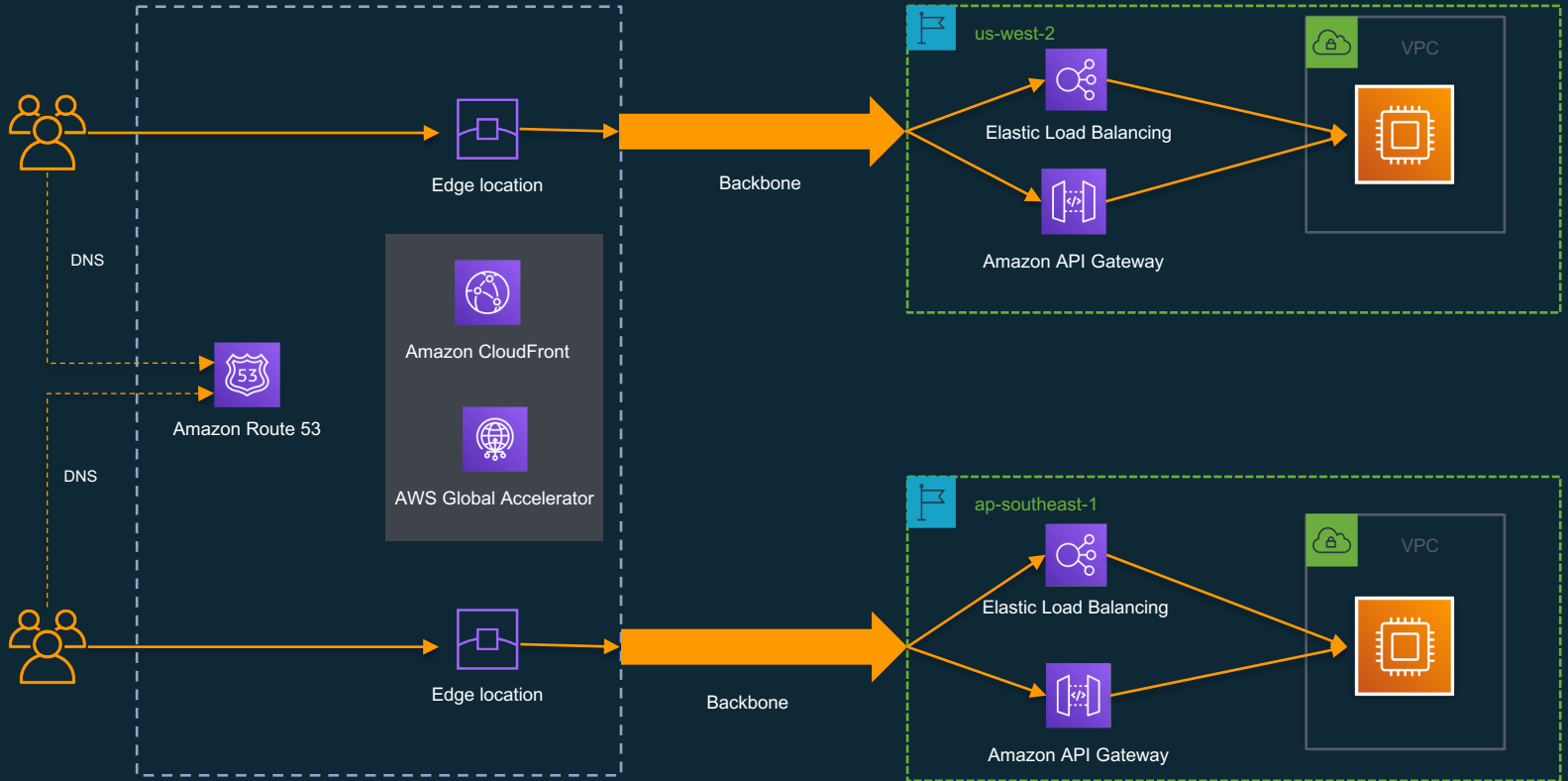


Source: Netflix tech blog

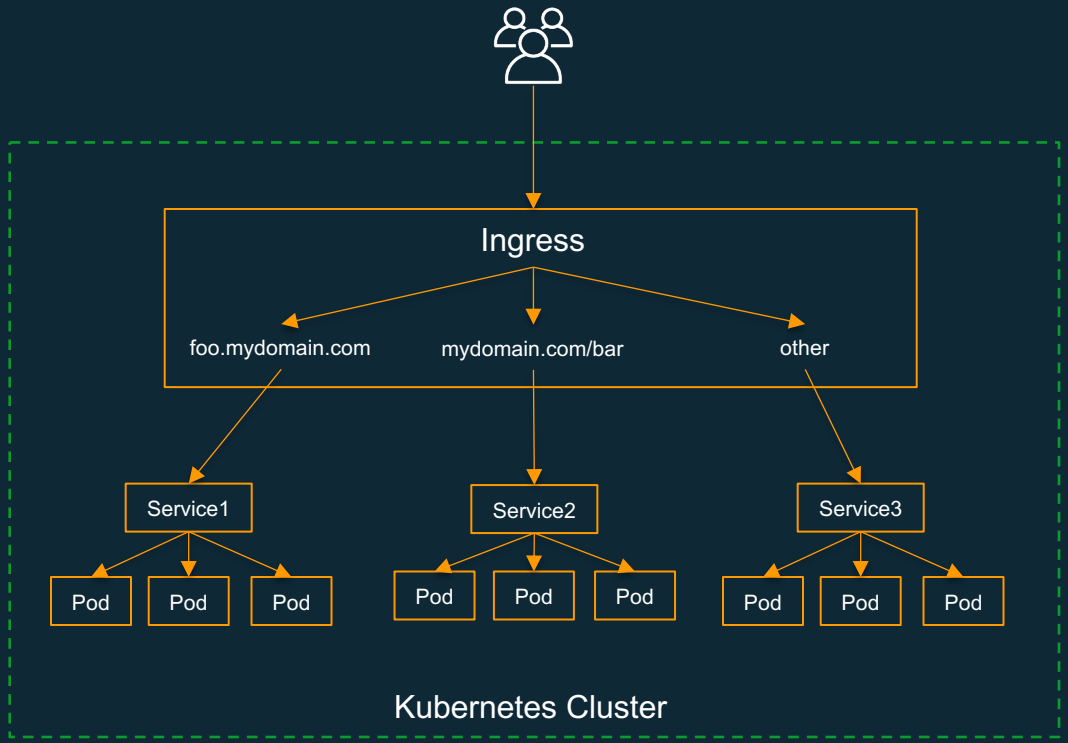
AWS网络相关服务



应用外部网络流量管理



Kubernetes Ingress



用于管理从外部访问集群内服务的的一个API对象，通常是HTTP。

Ingress可以提供负载平衡，SSL终止和基于名称的虚拟主机。

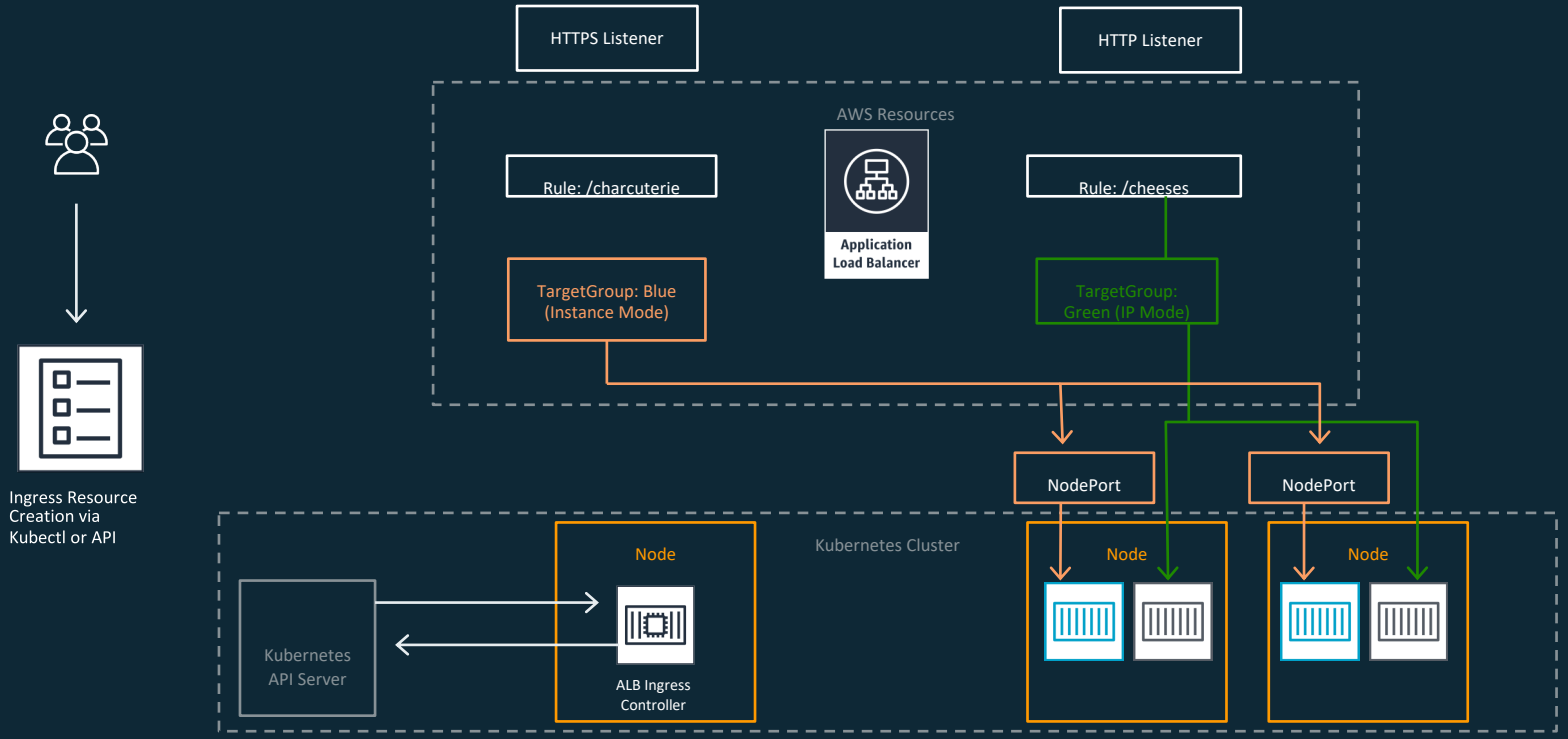
Kubernetes Ingress

	AWS ALB Ingress	ingress-nginx	ambassador	traefik 2.0	kong	istio ingress	contour	haproxy	citrix ingress controller	Gloo Solo	F5 Networks	voyager
backend service discovery	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic
protocol	http, https	http,https,tcp (separate lb),udp,grpc,fastcgi,IPC socket	http,https,grpc,tcp, tcp+ssl/tls	http,https,grpc,tcp + tls	http,https, grpc	tcp,http,https,grpc	http,https,tcp,grpc	http,tcp	http,https,tcp,ssl-tcp,udp	tcp,http,https,grpc	tcp, http, https	http,https,tcp
based on	AWS ALB	nginx	envoy	traefik	kong (nginx)	envoy	envoy	haproxy	Citrix ADC	envoy	F5 ADC	haproxy
ssl termination	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
websocket	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
routing		host,path(with regex)	host,header,path	host,path	host,path (with regex), method, header	host,user	host,path	host,path	host,path	header, query param, http method, path, plugin, function	- Full Ingress support - Openshift Routes - Any L3/L4/L7 info when using AS3 Extension integration	host,path
scope	cross-namespace	cross-namespace	cross-namespace	cross-namespace	cross-namespace	cross-namespace	cross namespace	optional cross-namespace	cross-namespace	cross-namespace	cross-namespace	cross-namespace
resiliency	rate-limit, health-check	rate limit, retries	circuit break, rate limit, retries	https://docs.traefik.io/middlewares/overview/ ; CircuitBreaker, RateLimit, Retry, Buffering, many more.	active and passive health check, circuit break, rate limit, retries	circuit break, retries	retries	-	health check	rate limit, health check	active and passive health check, ramp-up, rate limit, retries	-
lb algorithms		rr,ewma,ip_hash	wrr,ring hash,maglev	HTTP: rr, wrr, mirroring; TCP: RR, WRR;	rr, hash, header, cookie	rr,leastconn,random,passthrough	wrr,wlr,ring hash, maglev, random	rr, srr, leastconn,first,source,uri,url_param,hdr,rdp-	rr,least_conn,wrr,least_response,hash	rr, least request, random	"dynamic-ratio-member", "dynamic-ratio-node", "fastest-app-response", "fastest-node", "least-	rr
auth		basic, digest, external auth	yes	basic, digest and forward auth in alpha	Basic Auth, HMAC, JWT, Key, LDAP, OAuth 2.0, PASETO, plus paid Kong Enterprise options like OpenID Connect	JWT	-	basic	basic	basic, oidc, custom	Wide range of auth options with APM module	basic,oauth
Tracing		yes	yes	yes	yes	yes	-	-	-	yes	-	-
canary/shadow		canary	canary,shadow	canary, mirroring	canary		canary	-	canary	canary	Blue-Green Deployment, A/B Deployment	-
istio integration		-	yes	-	-	yes	-	-	-	yes	-	-
linkrd2		yes	yes	yes	-	-	-	yes	-	yes	-	
state		kubernetes	kubernetes	kubernetes	kubernetes	kubernetes	kubernetes	kubernetes	kubernetes	kubernetes	kubernetes	kubernetes
Paid support		-	yes	yes	yes	-	yes	yes	yes	yes	yes	yes

https://kubedex.com/ingress/

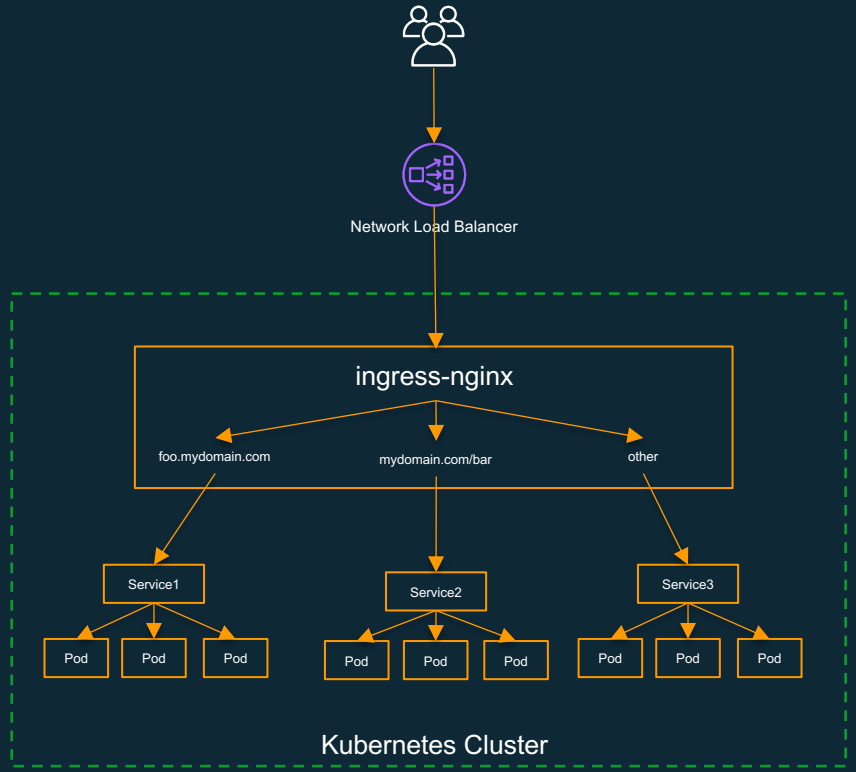
aws

ALB Ingress Controller



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

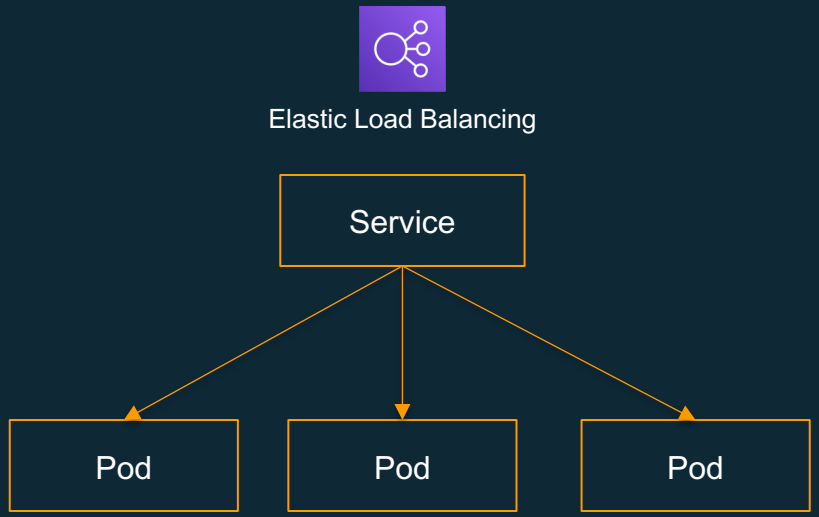
NLB + ingress-nginx



- Static IP/elastic IP addresses
- Scalability
- Zonal isolation
- Source/remote address preservation
- Long-lived TCP connections
- Reduced bandwidth usage
- SSL termination

aws

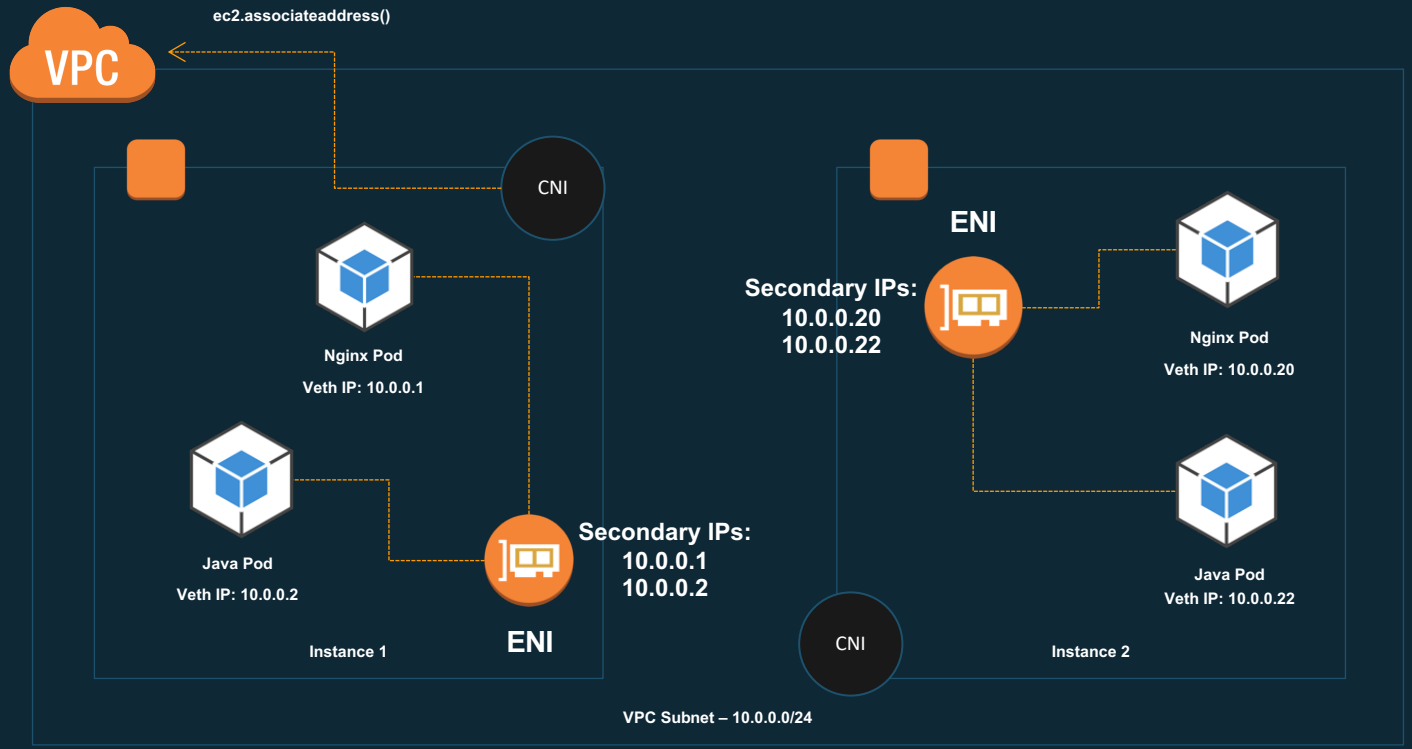
Kubernetes Service



- ClusterIP virtual IP, accessible from all nodes
- LoadBalancer automatically creates a public ELB (using IAM role)
- NodePort bind service to the same port on every host

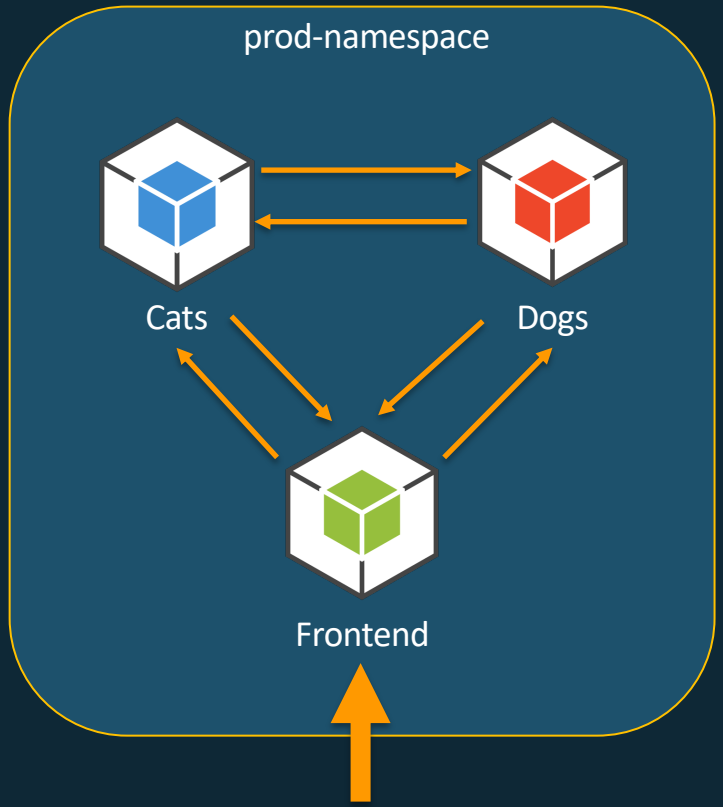
aws

Kubernetes CNI - AWS VPC CNI 插件



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

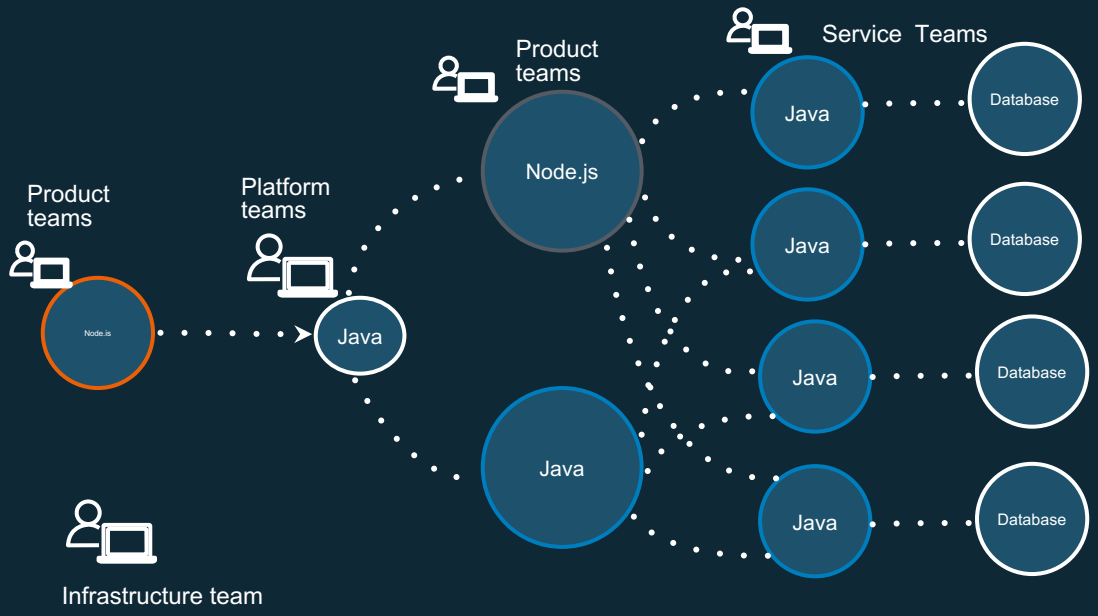
Kubernetes Network Policy




```
kind: NetworkPolicy
apiVersion: extensions/v1beta1
metadata:
 name: default-deny
spec:
 podSelector:
  matchLabels: {}
```


aws

Service Mesh



控制服务与服务之间的通讯

服务与服务之间通讯的可观察性

组织创新的小DevOps团队

自动化的安全合规检测

AWS App Mesh

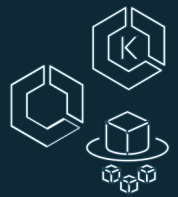
一个全托管的服务网格

利用Sidecar代理机制

App Mesh is a service mesh

不需要开发构建和维护

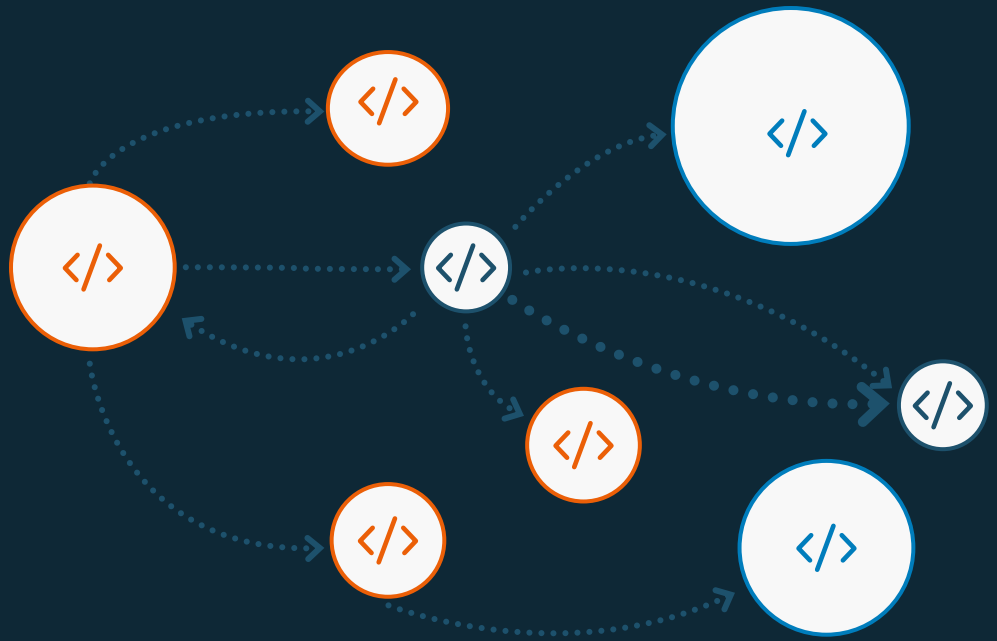
不依赖于应用程序部署平台
(例如：容器编排)



适用于不同计算平台

可靠地存储和分发配置

App Mesh – 流量与路由管理



流量管理

Load balancing

Weight targets

Service discovery (DNS + AWS Cloud Map)

Health checks

Retries

Timeouts

Circuit breakers

路由控制

Protocols support (HTTP, TCP, gRPC)

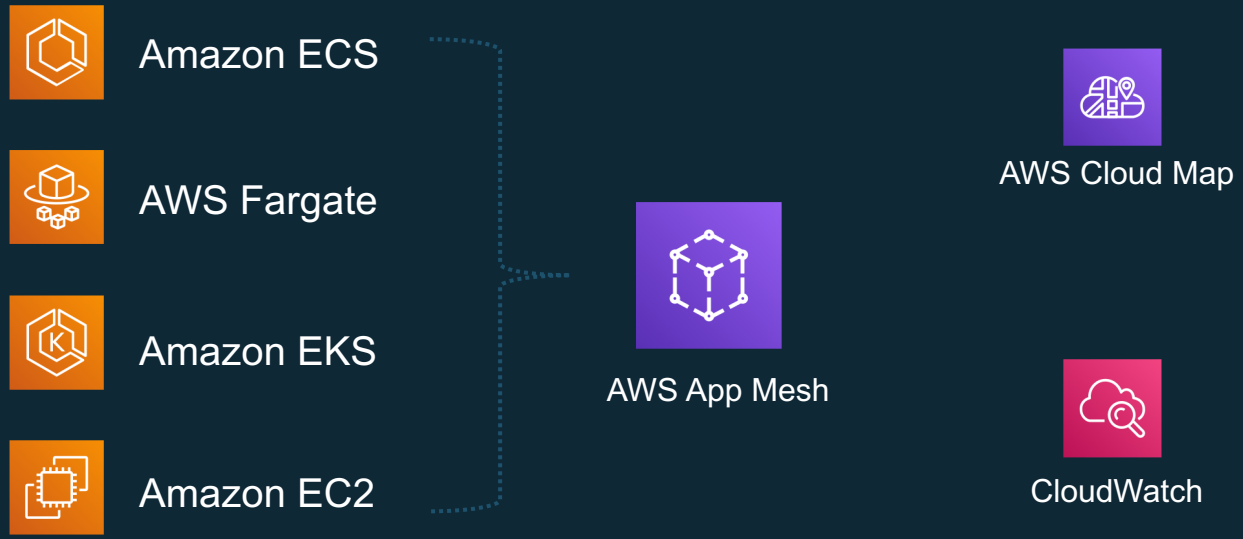
Path-based

Header-based

Cookie-based

https://github.com/aws/aws-app-mesh-roadmap/projects/1

App Mesh: 与AWS服务相集成



App Mesh 构成

Mesh
Virtual node
Virtual router and routes
Virtual service

Create and manage these in App Mesh
API, CLI, SDK, or
AWS Management Console

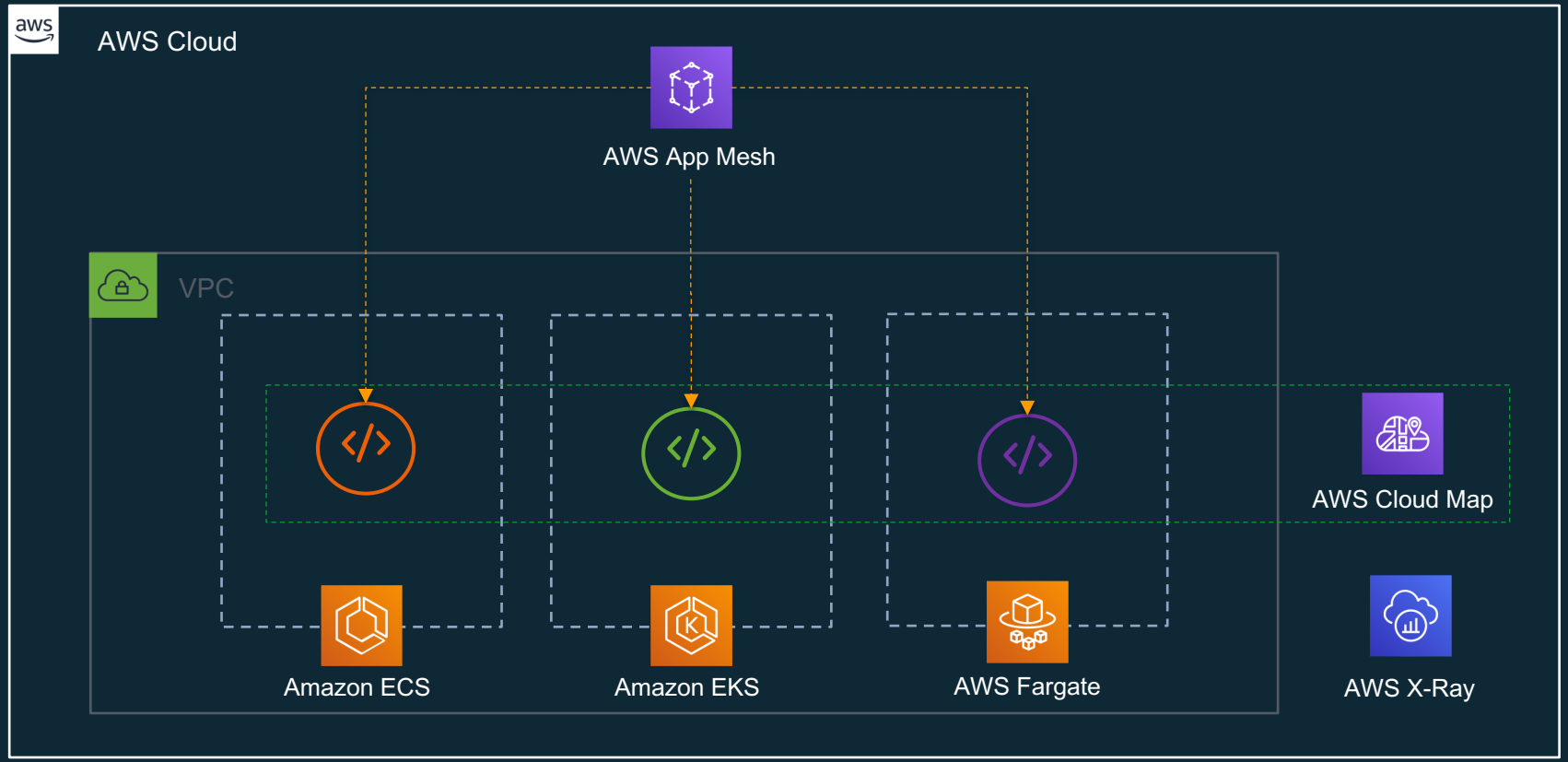
Proxies
Services
Service discovery

Configure and run proxies and services
on Amazon ECS, Fargate, Amazon
EKS, Amazon EC2

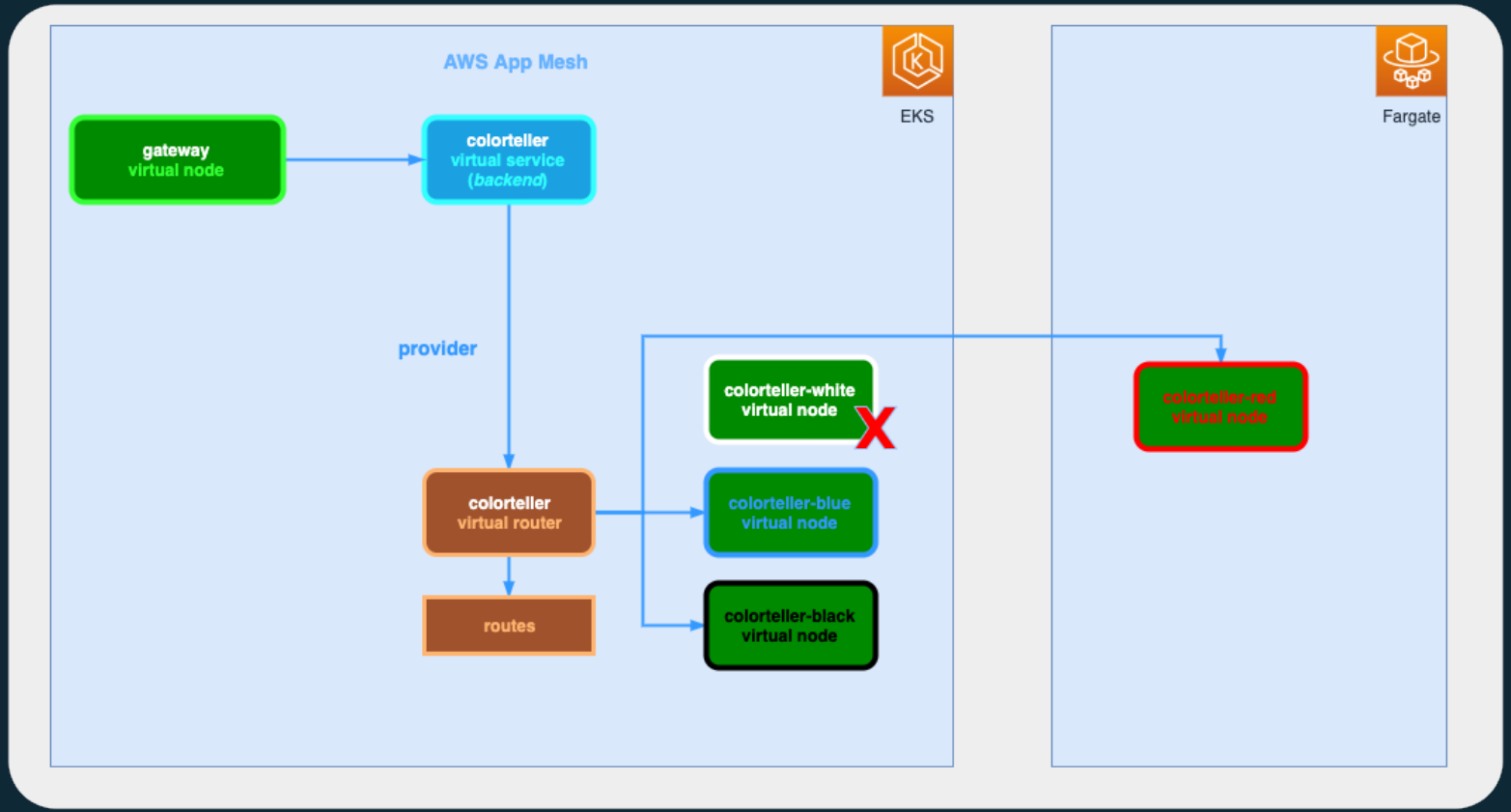
Service discovery with
AWS Cloud Map

aws

App Mesh 跨集群部署管理



Demo



Q&A

Thank you!