



Cloud Native Development in AWS with Container and Serverless

使用容器和无服务器在 AWS 中进行云原生开发

Pahud Hsieh

Specialist Solutions Architect, Serverless

Aug 22, 2019



CLOUD NATIVE
COMPUTING FOUNDATION

A little bit about Myself

hunhsieh



Pahud
Hsieh

- Joined AWS in 2016
- Focus on Serverless and Containers
- Public Speaker in AWS Global Summits, KubeCon and communities
- Serverless Specialist SA since 2019 Feb
- Father of a 5-year-old daughter
- Road trip lover

Agenda

1. What's New in Kubernetes on AWS
2. eksctl – A CLI for Amazon EKS
3. AWS CDK with Kubernetes
4. Service Mesh in AWS
5. Kubernetes Ingress and Load Balancing in AWS
6. Building Kubernetes in AWS China Regions
7. Hybrid Serverless and Container in AWS
8. Kubernetes Global Distributions in AWS



kubernetes



51%

of Kubernetes workloads
run on AWS today
—CNCF survey



Amazon EKS

AMAZON ELASTIC KUBERNETES SERVICE
(EKS)

We started in us-east-1 and us-west-2 and soon after...

- Released VPC CNI 1.0
- HIPAA Support
- Released our build scripts on Github
- Released VPC CNI 1.1
- Enabled GPU support
- Support for API Aggregation
- Support for HPA
- Support for eu-west-1
- CLI support for writing the kubeconfig
- Support for Admission Controllers
- Released VPC CNI 1.2
- Allow for additional VPC CIDR ranges
- Support for us-east-2

We started in us-east-1 and us-west-2 and soon after...

- Official support for ALB Ingress
- Container Market Place
- AW Cloud Map integrations
- Support for AWS App Mesh
- Support for eu-central-1, ap-southeast-1, ap-southeast-2, ap-northeast-1
- Support for ap-northeast-2
- ADDED AN SLA... IN LESS THAN A YEAR!

We started in us-east-1 and us-west-2 and soon after...

- Achieved ISO and PCI compliance
- Support for ap-south-1, eu-west-2, eu-west-3
- Released VPC CNI 1.3
- Added a new QuickStart
- Allowed private API Endpoints
- Launched an App Mesh controller at GA
- Public Preview for Windows nodes
- Deep Learning container launch
- Added 1.12 with a new cluster update api

We started in us-east-1 and us-west-2 and soon after...

- Released CSI Drivers for FSx and EFS
- Support for getting control plane logs
- Public Preview of A1 instances
- Released a Machine Learning Benchmark tool
- Support for Public IPs in Cluster VPCs
- Support for the public preview of CloudWatch Container Insights

AWS Containers Public Roadmap

aws / containers-roadmap

Watch 420 Unstar 1,632 Fork 54

Code Issues 292 Pull requests 0 Projects 1 Security Insights

containers-roadmap

Updated yesterday

Filter cards

Fullscreen Menu

26 Researching

① [ecs UI makes it hard to see image name](#)

#432 opened by ORESoftware

Console ECS

① [\[ECS\] Pre-install AWS SSM Agent on the ECS-optimized Amazon Linux 2 AMI](#)

#426 opened by coultn

ECS

① [\[ECS\] Increase maximum number of containers in a task definition from 10 to 30](#)

#368 opened by coultn

ECS

① [EKS - Cost Options on Control Plane \(developer friendly\)](#)

#45 opened by jpoley

EKS Proposed

① [\[EKS\]: EKS Cluster Tagging Propagation](#)

#374 opened by tabern

EKS

① [\[EKS\] \(request\): AWS Secrets Manager / SSM Parameter Store](#)

#168 opened by jaronsummers

38 We're Working On It

① [\[EKS\]: Next Generation AWS VPC CNI Plugin](#)

#398 opened by tabern

EKS Proposed

① [Fargate Log Driver Support - Sumo Logic](#)

#39 opened by nikovirtala

Fargate Proposed

① [\[ECS\] Mixed On-Demand/Spot tasks and services in an ECS cluster, with automatic scaling of On-Demand and Spot instances](#)

#391 opened by coultn

ECS

① [Fargate ephemeral volume encryption \(developer friendly\)](#)

#314 opened by akshayram-wolverine

Fargate

① [Fargate in Paris data center](#)

#309 opened by daya

Fargate Proposed

① [New EKS Region: GovCloud West](#)

#253 opened by joemccall86

EKS Proposed

11 Coming Soon

① [\[ECS\] Support use of swap memory](#)

#322 opened by mhowell-lms

ECS Proposed

① [\[ECS\] CloudFormation support for secrets configuration with logging drivers](#)

#393 opened by coultn

ECS

① [\[ECS, Fargate\] CloudFormation support for scheduled tasks with Fargate Launch Type](#)

#392 opened by coultn

ECS Fargate

① [\[ECS\] \[CloudFormation\]: CloudFormation Support for ipcMode and pidMode \(TaskDefinition\)](#)

#286 opened by joelnyc

ECS Proposed

① [\[ECS\] \[CloudFormation\]: CloudFormation support for SystemControls, interactive, and pseudoTerminal](#)

#96 opened by talawahtech

ECS Proposed

① [New EKS Region : Ningxia](#)

5 Developer Preview

① [Can't see task level cpu/memory utilization in cloudwatch](#)

#106 opened by rutchkiwi

ECS Proposed

① [\[ECS\] \[Metrics\]: Container Insights for ECS](#)

#70 opened by bradenschmidt

ECS Fargate Proposed

① [\[ECS, Fargate\] CloudWatch metrics for number of running and pending tasks per service and cluster](#)

#282 opened by coultn

ECS Fargate

① [EKS Windows Nodes \(preview\)](#)

#69 opened by ofiliz

Developer Preview EKS

① [\[EKS\]: Support for Arm Nodes - EC2 A1 Instances](#)

#264 opened by tabern

Developer Preview EKS

70 Just Shipped

EKS Proposed

① [\[EKS\] Cloudformation support for cluster upgrades](#)

#115 opened by dcherman

EKS Proposed

① [Kubernetes Version 1.11.8](#)

#188 opened by tabern

EKS

① [EKS Support for Kubernetes 1.12](#)

#24 opened by pauncejones

EKS

① [EKS Private Endpoint Support](#)

#22 opened by pauncejones

EKS

① [EKS CNI Plugin 1.3](#)

#25 opened by pauncejones

EKS

① [New EKS Region: Mumbai](#)

#145 opened by mogren

EKS

① [\[EKS\] \[Proposal\]: Container Ordering](#)

#123 opened by oetderek

Amazon EKS Public Roadmap

[aws / containers-roadmap](#)

[Watch](#) 420 [Unstar](#) 1,632 [Fork](#) 54

[Code](#) [Issues 292](#) [Pull requests 0](#) [Projects 1](#) [Security](#) [Insights](#)

containers-roadmap
Updated yesterday

26 Researching2 results

[EKS - Cost Options on Control Plane \(developer friendly\)](#)
#45 opened by jpoley
EKS Proposed

[\[EKS\]: EKS Cluster Tagging Propagation](#)
#374 opened by tabern
EKS

38 We're Working On It13 results

[New EKS Region: GovCloud West](#)
#253 opened by joemccall86
EKS Proposed

[New EKS Region: GovCloud East](#)
#375 opened by tabern
EKS

[\[EKS\] \[Security\]: Allow restricting EKS API Access via Security Groups](#)
#108 opened by jrn30
EKS Proposed

[EKS Tagging](#)
#82 opened by praveenchandran
EKS Proposed

[New EKS Region : Beijing](#)
#219 opened by wholroyd
EKS Proposed

[DNS resolution for EKS Private Endpoints](#)
#221 opened by tabern
EKS

[EKS on Fargate](#)

11 Coming Soon4 results

[New EKS Region : Ningxia](#)
#77 opened by pahud
EKS

[EKS IAM Roles for Pods](#)
#23 opened by pauncejones
EKS

[EKS-Optimized AMI Metadata SSM Parameter](#)
#231 opened by tabern
EKS

[\[EKS\]: Service Linked Role for Amazon EKS](#)
#243 opened by tabern
EKS

5 Developer Preview1 result

[EKS Windows Nodes \(preview\)](#)
#69 opened by ofiliz
Developer Preview EKS

70 Just Shipped25 results

[\[EKS\] New EKS Region: Hong Kong](#)
#267 opened by paulwilljones
EKS Proposed

[EKS Support for Kubernetes 1.13](#)
#30 opened by uprightvinyl
EKS Proposed

[SOC compliance for EKS](#)
#296 opened by abby-fuller
EKS

[EKS: Get-Token CLI Subcommand](#)
#292 opened by tabern
EKS

[Support for Public IP space in VPC with EKS](#)
#181 opened by tabern
EKS

[\[EKS\] \[request\]: Release CNI Plugin 1.4 for EKS](#)
#149 opened by mogren
EKS

[EKS / Kubernetes: Add support for ...](#)

eksctl – The official CLI for Amazon EKS

sponsored by weaveworks

`eksctl` is a simple CLI tool for creating clusters on EKS - Amazon's new managed Kubernetes service for EC2. It is written in Go, uses CloudFormation, was created by [Weaveworks](#) and it welcomes contributions from the community. Create a basic cluster in minutes with just one command :

eksctl create cluster

<https://eksctl.io>

“eksctl is now officially our command line for EKS”

<https://aws.amazon.com/blogs/opensource/eksctl-eks-cli/>

customize your cluster

```
$ eksctl create cluster -f cluster.yaml
```

manage nodegroups

```
$ eksctl create nodegroup --cluster=<clusterName> [--  
name=<nodegroupName>]
```

cluster update

```
$ eksctl update cluster --name=<clusterName>
```

Nodegroup of Mixed Instance Types

```
nodeGroups:
  - name: ng-1
    minSize: 2
    maxSize: 5
    instancesDistribution:
      maxPrice: 0.017
      instanceTypes: ["t3.small", "t3.medium"] # At least two instance typ
      onDemandBaseCapacity: 0
      onDemandPercentageAboveBaseCapacity: 50
      spotInstancePools: 2
```



Nodegroup of GPU Instances

```
nodeGroups:
- name: ng-gpu
  instanceType: mixed
  desiredCapacity: 1
  instancesDistribution:
    instanceTypes:
      - p2.xlarge
      - p2.8xlarge
      - p2.16xlarge
  maxPrice: 0.50
```

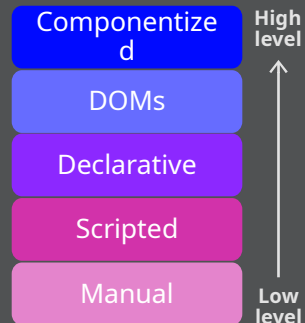
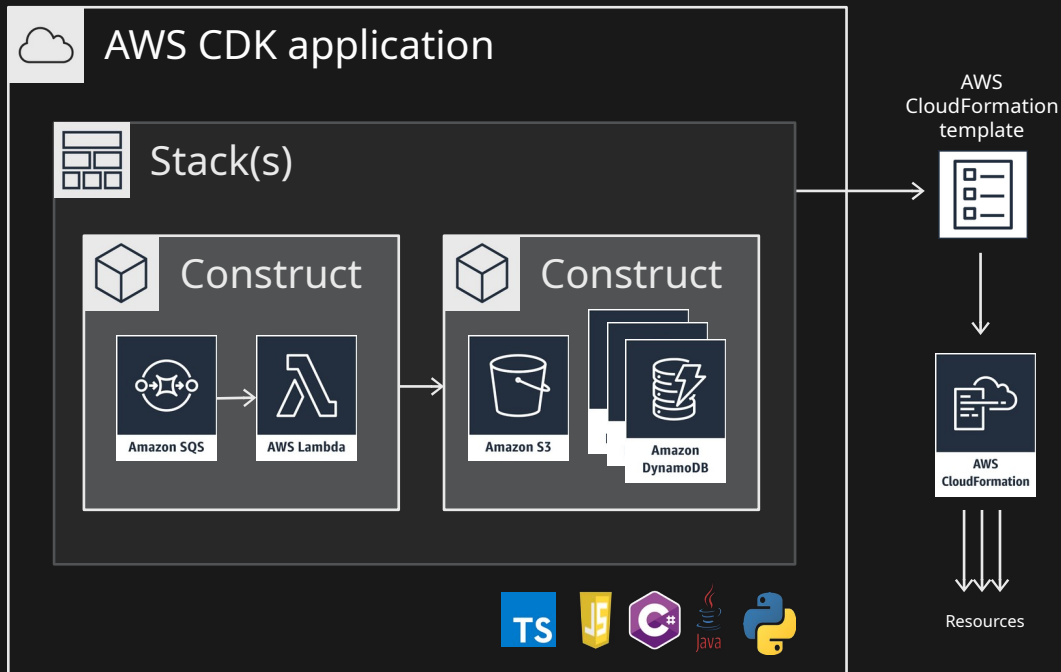


AWS Cloud Development Kit (AWS CDK)




AWS CDK

General Available



AWS CDK

General Available

 AWS Cloud Development Kit

0.17.0

Search docs

Getting Started

Tutorial

Concepts

AWS Construct Library

AWS CloudFormation Library

Examples

Tools


Writing Constructs


Docs » Reference


View page source


AWS Construct Library Reference


- [@aws-cdk/app-delivery](#)
- [@aws-cdk/assets](#)
- [@aws-cdk/aws-amazonmq](#)
- [@aws-cdk/aws-apigateway](#)
- [@aws-cdk/aws-applicationautoscaling](#)
- [@aws-cdk/aws-appstream](#)
- [@aws-cdk/aws-appsync](#)
- [@aws-cdk/aws-athena](#)
- [@aws-cdk/aws-autoscaling](#)
- [@aws-cdk/aws-autoscalingplans](#)
- [@aws-cdk/aws-batch](#)
- [@aws-cdk/aws-budgets](#)
- [@aws-cdk/aws-certificatemanager](#)
- [@aws-cdk/aws-cloud9](#)
- [@aws-cdk/aws-cloudformation](#)
- [@aws-cdk/aws-cloudfront](#)
- [@aws-cdk/aws-cloudtrail](#)
- [@aws-cdk/aws-cloudwatch](#)
- [@aws-cdk/aws-codebuild](#)
- [@aws-cdk/aws-codecommit](#)
- [@aws-cdk/aws-codedeploy](#)



AWS Step Functions



Amazon SQS



Amazon SNS



AWS Lambda


Amazon EC2


Amazon DynamoDB


Amazon ECS


AWS Step Functions


Amazon S3

Componentized

DOMs

Declarative

Scripted

Manual

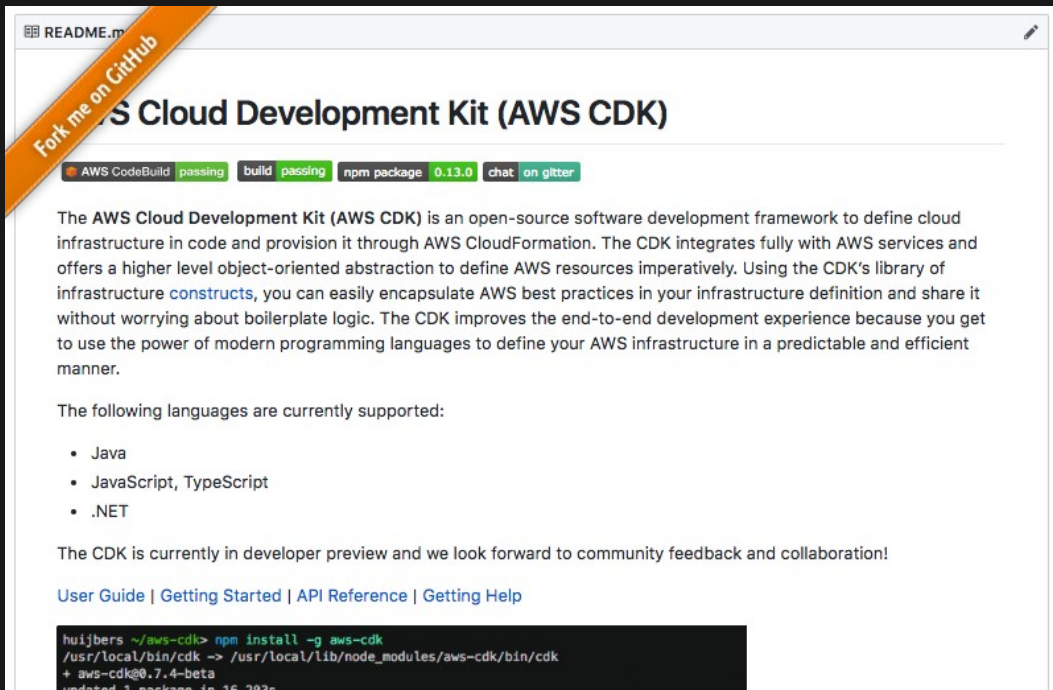
High level

Low level



AWS CDK

General Available



The screenshot shows the GitHub README for the AWS Cloud Development Kit (AWS CDK). It features a 'Fork me on GitHub' banner, a title 'AWS Cloud Development Kit (AWS CDK)', and a status bar with 'AWS CodeBuild passing', 'build passing', 'npm package 0.13.0', and 'chat on gitter'. The main text describes the CDK as an open-source framework for defining AWS infrastructure in code. It lists supported languages: Java, JavaScript, TypeScript, and .NET. A terminal snippet at the bottom shows the installation command: `npm install -g aws-cdk`.

Fork me on GitHub

AWS Cloud Development Kit (AWS CDK)

AWS CodeBuild passing build passing npm package 0.13.0 chat on gitter

The AWS Cloud Development Kit (AWS CDK) is an open-source software development framework to define cloud infrastructure in code and provision it through AWS CloudFormation. The CDK integrates fully with AWS services and offers a higher level object-oriented abstraction to define AWS resources imperatively. Using the CDK's library of infrastructure **constructs**, you can easily encapsulate AWS best practices in your infrastructure definition and share it without worrying about boilerplate logic. The CDK improves the end-to-end development experience because you get to use the power of modern programming languages to define your AWS infrastructure in a predictable and efficient manner.

The following languages are currently supported:

- Java
- JavaScript, TypeScript
- .NET

The CDK is currently in developer preview and we look forward to community feedback and collaboration!

[User Guide](#) | [Getting Started](#) | [API Reference](#) | [Getting Help](#)

```
huijbers ~/aws-cdk> npm install -g aws-cdk
/usr/local/bin/cdk -> /usr/local/lib/node_modules/aws-cdk/bin/cdk
+ aws-cdk@0.7.4-beta
updated 1 package in 16.203s
```

Componentized

DOMs

Declarative

Scripted

Manual

High level

Low level



Show me the Code!

`new eks.Cluster();`

- Dedicated VPC with default configurations
- Amazon EKS cluster with managed control plane
- Required Security Groups and IAM Role
- 2x m5.large EC2 instances with the latest Amazon EKS-optimized AMI in the nodegroup (just as `eksctl`)

Show me the IDE!

TS cdk-stack.ts ●

lib > TS cdk-stack.ts > ...

```
1  import cdk = require('@aws-cdk/core');
2  import eks = require('@aws-cdk/aws-eks');
3
4  export class CdkStack extends cdk.Stack {
5      constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
6          super(scope, id, props);
7
8          const cluster = new eks.Cluster(this, 'Cluster')
9      }
10 }
11
12
```

● create a cluster
and a nodegroup

lib > TS cdk-stack.ts > CdkStack > constructor > cluster

```
1 import cdk = require('@aws-cdk/core');
2 import eks = require('@aws-cdk/aws-eks');
3
4 export class CdkStack extends cdk.Stack {
5   constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
6     super(scope, id, props);
7
8     const cluster = new eks.Cluster(this, 'Cluster', {
9
10    })
11  }
12 }
13
14
```

clusterName (property) eks.ClusterProps.clusterNam... ⓘ

defaultCapacity

defaultCapacityInstance

kubectlEnabled

mastersRole

role

securityGroup

version

vpc

vpcSubnets

#endregion

#region

Trigger IDE parameter hint

```
export class CdkStack extends cdk.Stack {  
  constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {  
    super(scope, id, props);
```

```
    const vpc = ec2.Vpc.fromLookup(this, 'ExistingVpc', {  
      vpcName: 'DEV'  
    })
```

• Using existing VPC

```
    const cluster = new eks.Cluster(this, 'Cluster', {  
      vpc: vpc,  
      clusterName: 'eks-demo',  
      defaultCapacity: 6,  
      defaultCapacityInstance: new ec2.InstanceType('t3.large'),  
      version: '1.13'  
    })
```

• Create Amazon EKS cluster in this VPC with a customized nodegroup

```
    const spotASg = cluster.addCapacity('spots', {  
      instanceType: new ec2.InstanceType('m5.large'),  
      spotPrice: '0.1',  
      maxCapacity: 10,  
      minCapacity: 2,  
      desiredCapacity: 4  
    })
```

• Add additional spot instances in the nodegroup as well

```
    cluster.defaultCapacity!.scaleOnCpuUtilization('odUp', {  
      targetUtilizationPercent: 60  
    })
```

• Target tracking scaling policy for on-demand instances

```
    spotASg.scaleOnCpuUtilization('spotUp', {  
      targetUtilizationPercent: 50  
    })  
  }  
}
```

• Target tracking scaling policy for spot instances

\$ cdk synth

```
1 Transform: AWS::Serverless-2016-10-31
2 Resources:
3   ClusterDefaultVpcFA9F2722:
4     Type: AWS::EC2::VPC
5     Properties:
6       CidrBlock: 10.0.0.0/16
7       EnableDnsHostnames: true
8       EnableDnsSupport: true
9       InstanceTenancy: default
10      Tags:
11        - Key: Name
12          Value: CdkEksStack/Cluster/DefaultVpc
13      Metadata:
14        aws:cdk:path: CdkEksStack/Cluster/DefaultVpc/Resource
15  ClusterDefaultVpcPublicSubnet1Subnet3BFE180A:
16    Type: AWS::EC2::Subnet
17    Properties:
18      CidrBlock: 10.0.0.0/19
19      VpcId:
20        Ref: ClusterDefaultVpcFA9F2722
21      AvailabilityZone: us-west-2a
22      MapPublicIpOnLaunch: true
23      Tags:
24        - Key: Name
25          Value: CdkEksStack/Cluster/DefaultVpc/PublicSubnet1
26        - Key: aws-cdk:subnet-name
27          Value: Public
28        - Key: aws-cdk:subnet-type
29          Value: Public
30      Metadata:
31        aws:cdk:path: CdkEksStack/Cluster/DefaultVpc/PublicSubnet1/Subnet
32  ClusterDefaultVpcPublicSubnet1RouteTable1DCCDD98:
33    Type: AWS::EC2::RouteTable
34    Properties:
35      VpcId:
36        Ref: ClusterDefaultVpcFA9F2722
37      Tags:
38        - Key: Name
39          Value: CdkEksStack/Cluster/DefaultVpc/PublicSubnet1
40      Metadata:
41        aws:cdk:path: CdkEksStack/Cluster/DefaultVpc/PublicSubnet1/RouteTable
42  ClusterDefaultVpcPublicSubnet1RouteTableAssociationAFBE6789:
43    Type: AWS::EC2::SubnetRouteTableAssociation
44    Properties:
45      RouteTableId:
46        Ref: ClusterDefaultVpcPublicSubnet1RouteTable1DCCDD98
47      SubnetId:
48        Ref: ClusterDefaultVpcPublicSubnet1Subnet3BFE180A
49      Metadata:
50        aws:cdk:path: CdkEksStack/Cluster/DefaultVpc/PublicSubnet1/RouteTableAssociation
51  ClusterDefaultVpcPublicSubnet1DefaultRouteCF22EF6E:
52    Type: AWS::EC2::Route
53    Properties:
54      RouteTableId:
55        Ref: ClusterDefaultVpcPublicSubnet1RouteTable1DCCDD98
56      DestinationCidrBlock: 0.0.0.0/0
57      GatewayId:
58        Ref: ClusterDefaultVpcIGW756BE43E
59    DependsOn:
60      - ClusterDefaultVpcVPCGW756BE43E
61    Metadata:
```

\$ cdk deploy



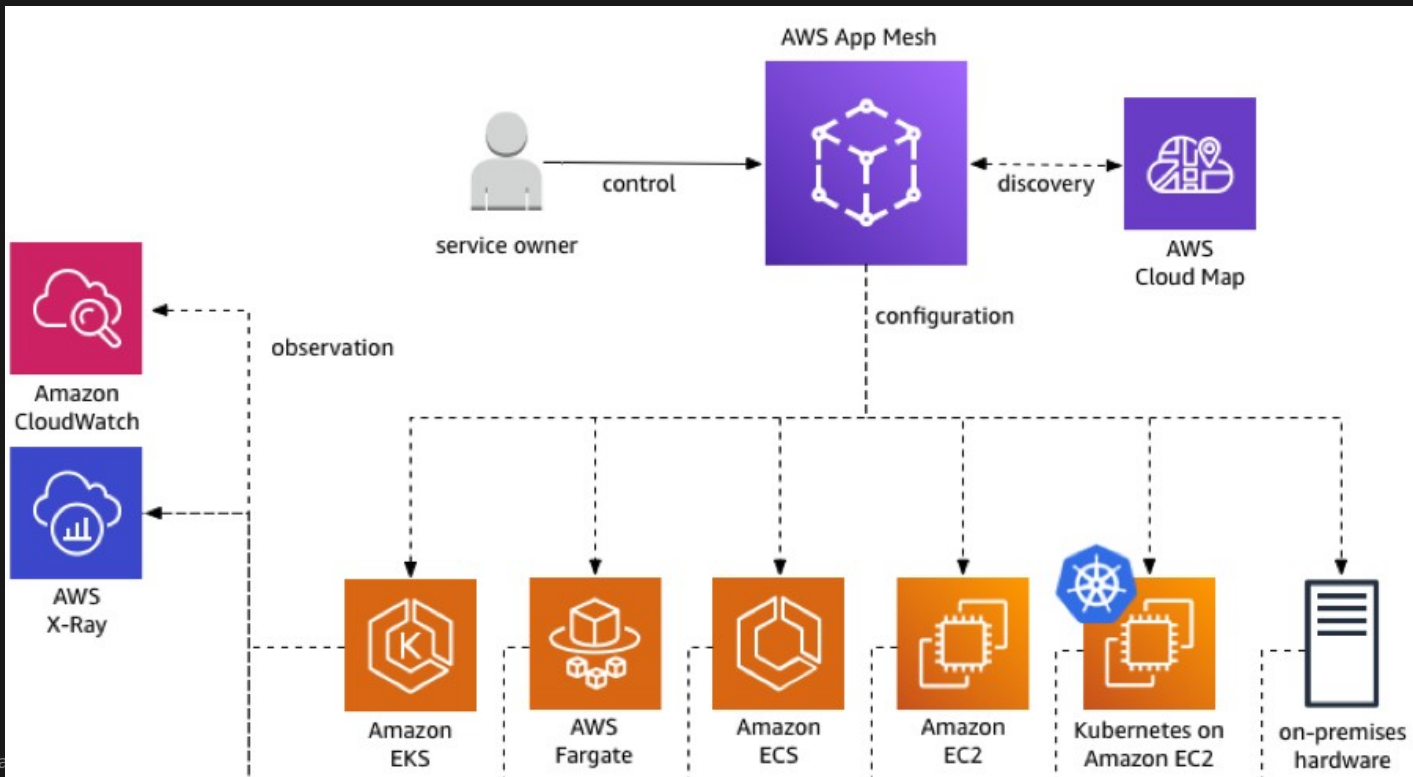
kubernetes

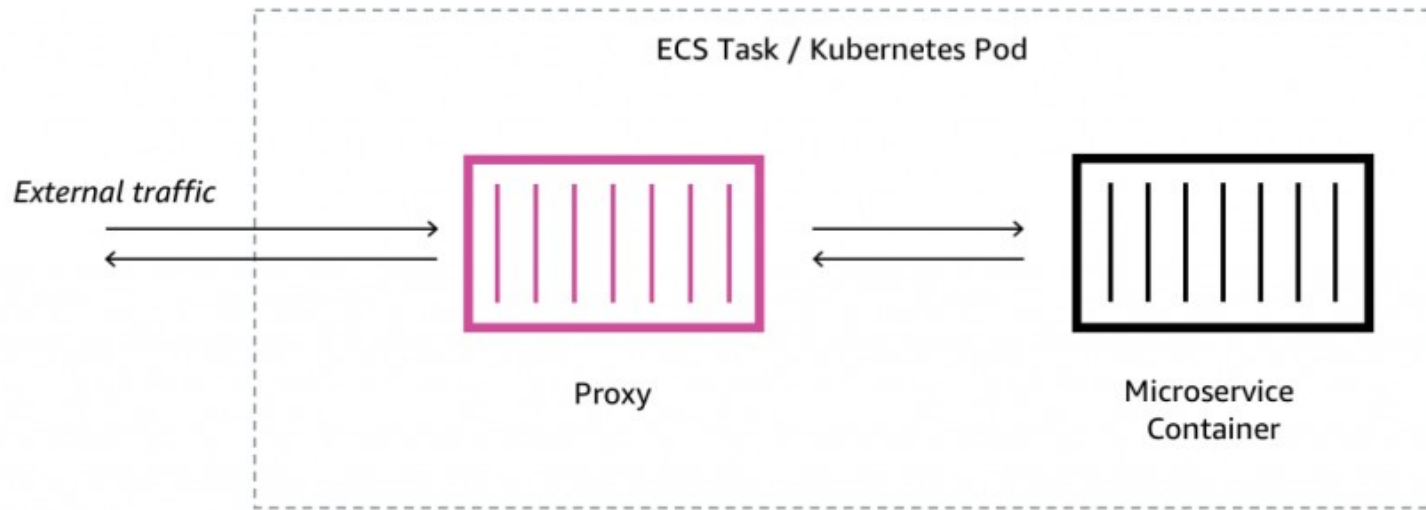
Service Mesh in AWS

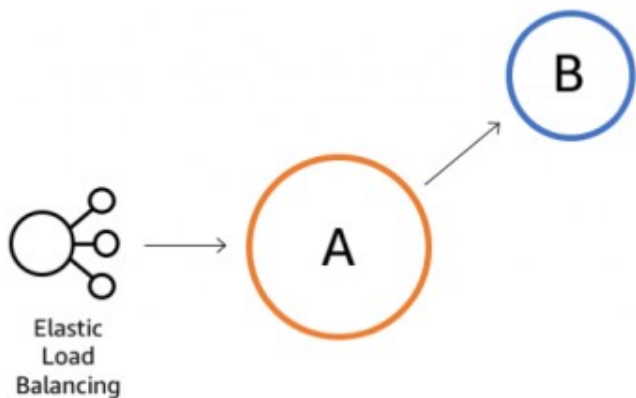


“Our vision for App Mesh is an AWS-native service mesh that integrates equally well with AWS primitives and advanced services.”

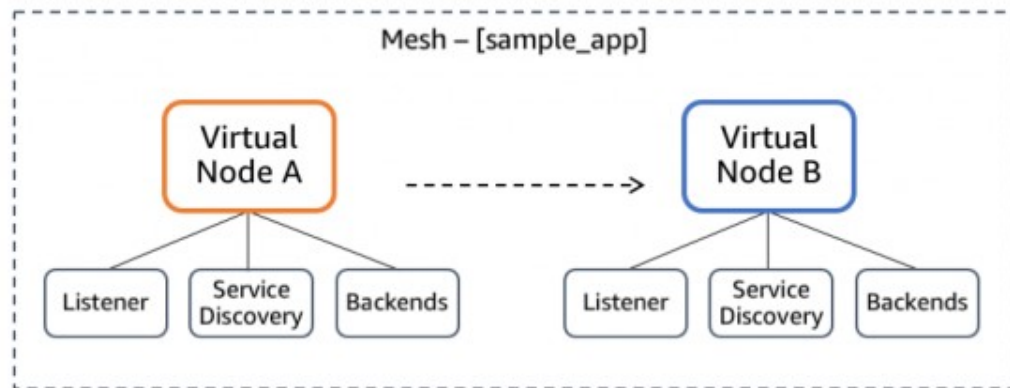
Werner Vogels
CTO - Amazon.com





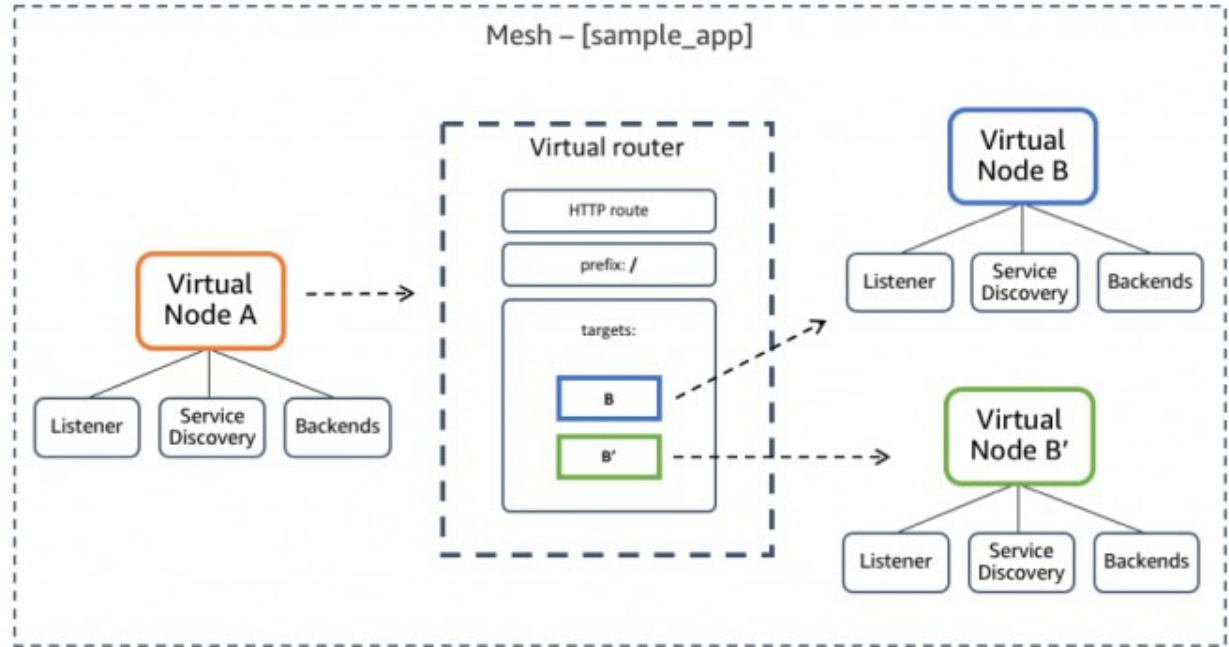
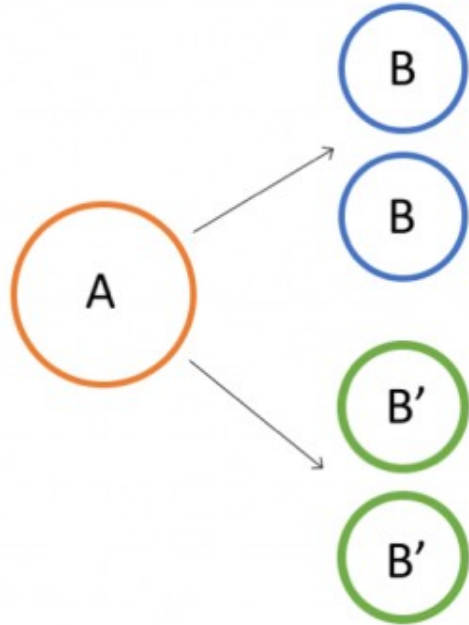


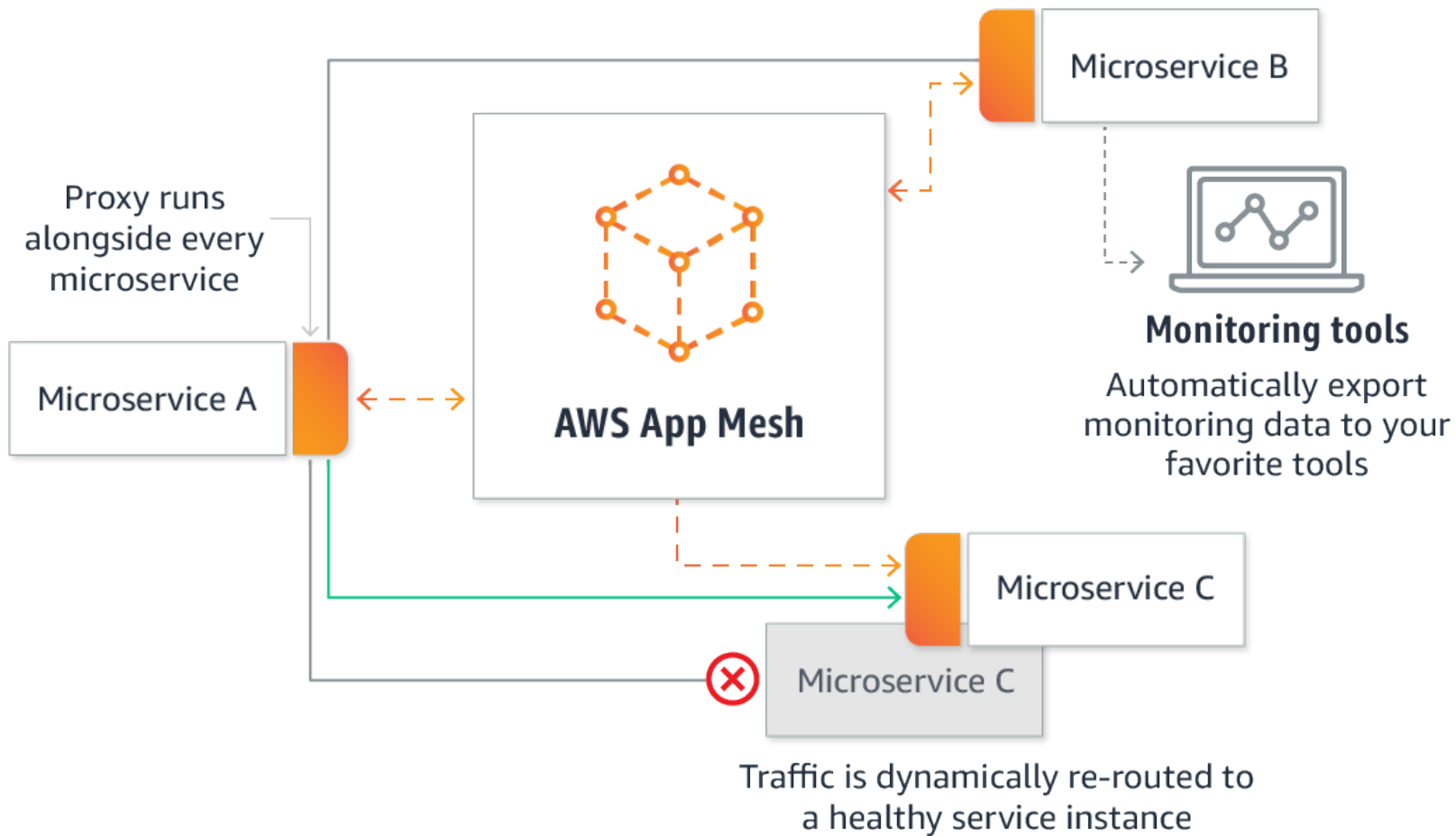
Microservices



App Mesh

Blue/Green Deployment or Traffic Switching





AWS AppMesh Public Roadmap in Github

The screenshot shows the 'aws / aws-app-mesh-roadmap' repository on Github. The repository is updated yesterday and has 53 issues, 0 pull requests, 1 project, 0 security issues, and 0 insights. The roadmap is organized into six columns, each representing a different stage of development. The first column, 'Researching', contains 8 items. The second column, 'We're Working On It', contains 2 items and a progress bar for 'Region expansion' at 13 of 23. The third column, 'Coming Soon', contains 4 items, including 'Proposed' items. The fourth column, 'Available in Preview Channel', contains 4 items. The fifth column, 'Just Shipped', contains 8 items. The sixth column, 'Just Shipped', contains 4 items. The roadmap is organized into six columns, each representing a different stage of development. The first column, 'Researching', contains 8 items. The second column, 'We're Working On It', contains 2 items and a progress bar for 'Region expansion' at 13 of 23. The third column, 'Coming Soon', contains 4 items, including 'Proposed' items. The fourth column, 'Available in Preview Channel', contains 4 items. The fifth column, 'Just Shipped', contains 8 items. The sixth column, 'Just Shipped', contains 4 items.

aws / aws-app-mesh-roadmap

Updated yesterday

Filter cards

Fullscreen Menu

11 Researching

- Provide the Envoy software in each region where App Mesh is available #56 opened by jtoberon
- Open Source the App Mesh Envoy Image build, release, and validation tools #5 opened by dastbe
- HIPAA eligibility #66 opened by shubharao
- Circuit Breaker Policy #6 opened by jamsajones
- GRPC routing #13 opened by jamsajones
- Use App Mesh for ingress routing #37 opened by jamsajones
- Simplify external service egress traffic setup #2 opened by bcelenza
- Enable custom filters for Envoy #41 opened by jamsajones
- Implement Fault Injection #36 opened by jamsajones

2 We're Working On It

- Support App Mesh Across Multiple Accounts #64 opened by dastbe
- Region expansion #1 opened by jamsajones (13 of 23)

4 Coming Soon

- Commit X-Ray tracer plugin to Envoy upstream #21 opened by jamsajones
- Bring Envoy from official release #10 opened by jamsajones
- End to end encryption of traffic with ACM managed certs #39 opened by jamsajones (Proposed)
- End to end encryption of traffic with customer provided certs #38 opened by jamsajones

4 Available in Preview Channel

- Retry Policy #7 opened by jamsajones
- Cookie based routing #14 opened by jamsajones
- HTTP Header based routing #15 opened by jamsajones
- Explicit Route Priorities #77 opened by jamsajones

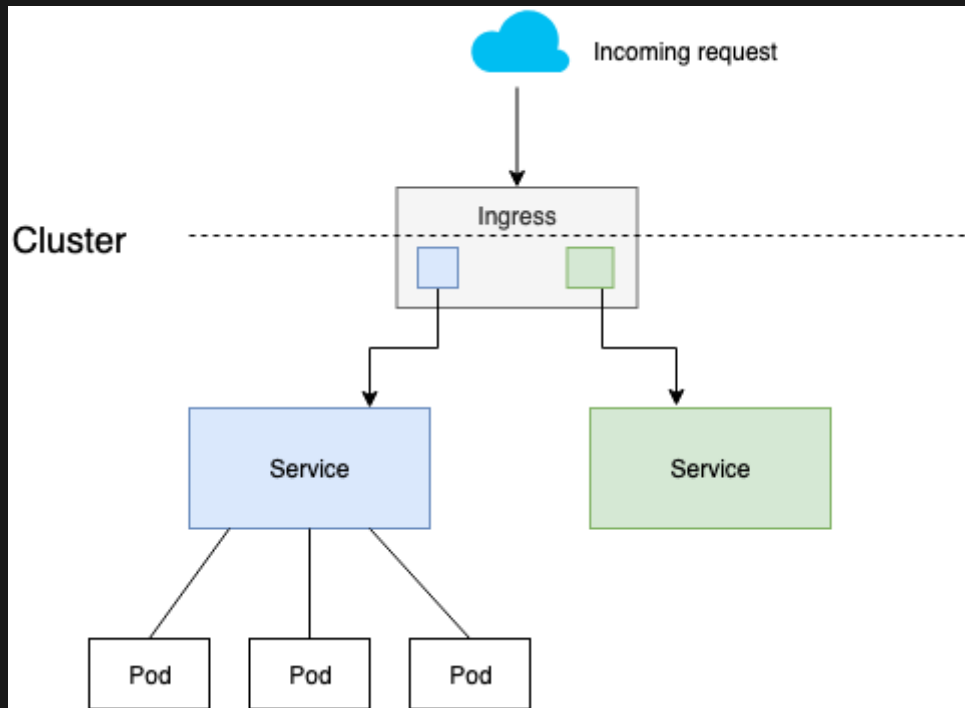
22 Just Shipped

- Support Envoy 1.11 #67 opened by shubharao
- ECS integration with App Mesh in the ECS console #8 opened by jamsajones
- VPC Endpoint/Private Link for App Mesh Envoy xDS API #12 opened by ewbankkit
- Hosted EDS implementation with AWS Cloud Map #11 opened by jamsajones
- AWS Cloud Map selectors #47 opened by coultn
- TCP routing #4 opened by jamsajones
- Resource-based authorization in IAM #20 opened by jamsajones
- App Mesh Console #22 opened by jamsajones
- Integration with EKS #9 opened by jamsajones
- CloudFormation

Kubernetes Ingress in AWS

Kubernetes Ingress in AWS

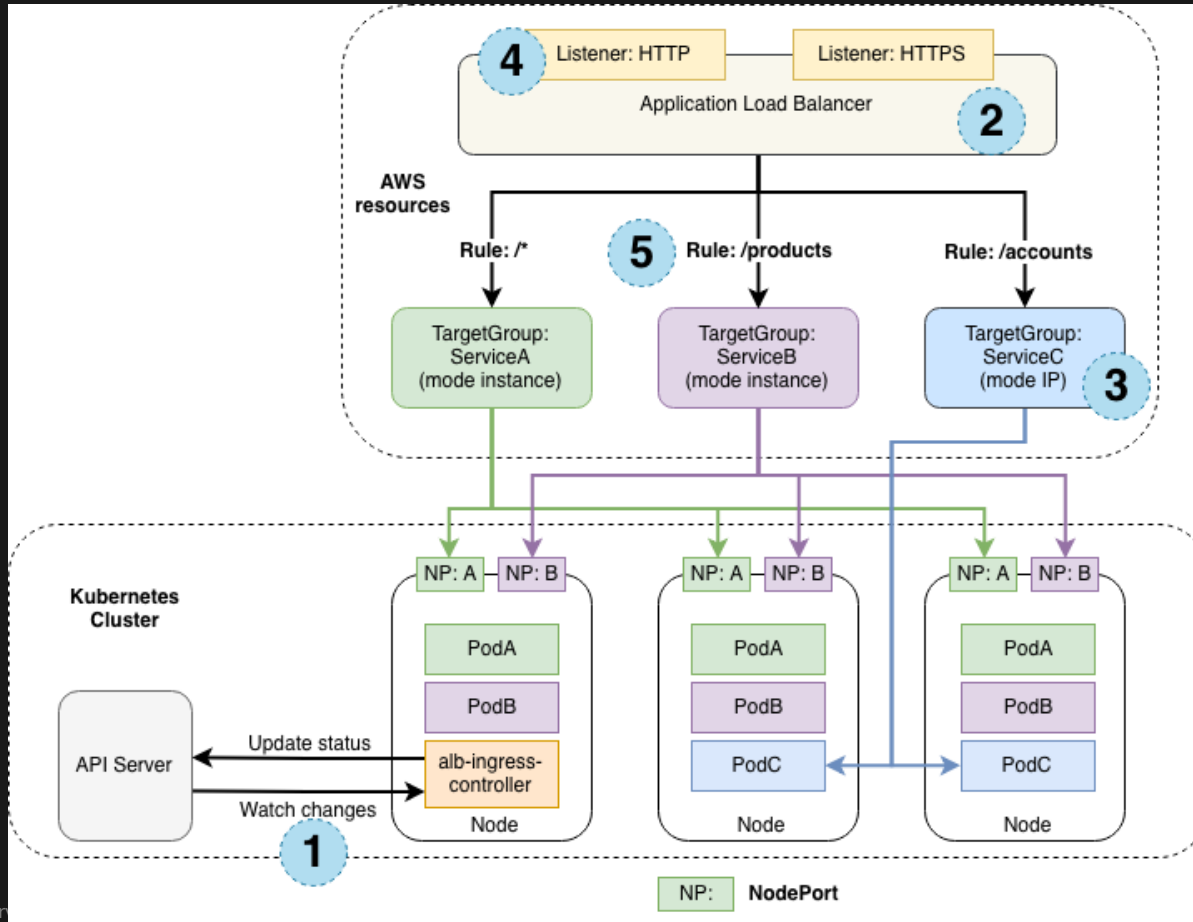
- AWS ALB Ingress Controller
- Nginx Ingress Controller
- HAProxy Ingress Controller
- Traefik Ingress Controller
- Gloo Ingress Controller



AWS ALB Ingress Controller

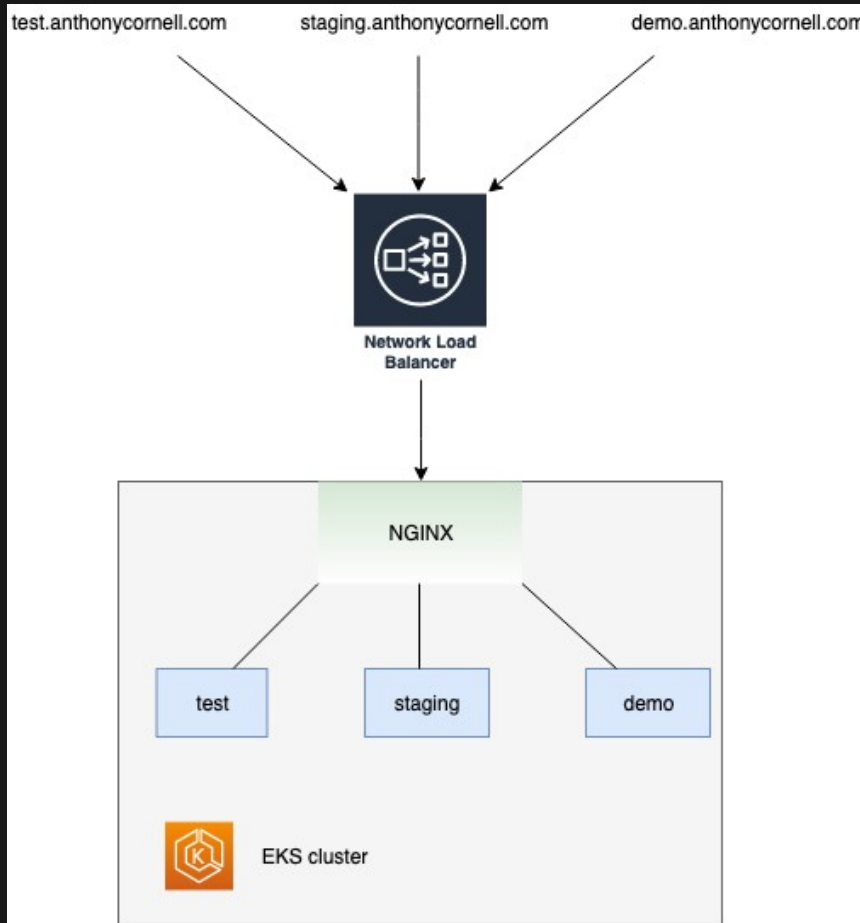
- Originated by Ticketmaster and CoreOS
- Donated to AWS-SIG
- Maintained by SIG-AWS Contributors
- `kubernetes-sigs/aws-alb-ingress-controller`
- Host mode balancing through nodeports
- IP Mode balancing as a container native load balancer
- SSL/TLS termination and AWS ACM/WAF integration

AWS ALB Ingress Controller



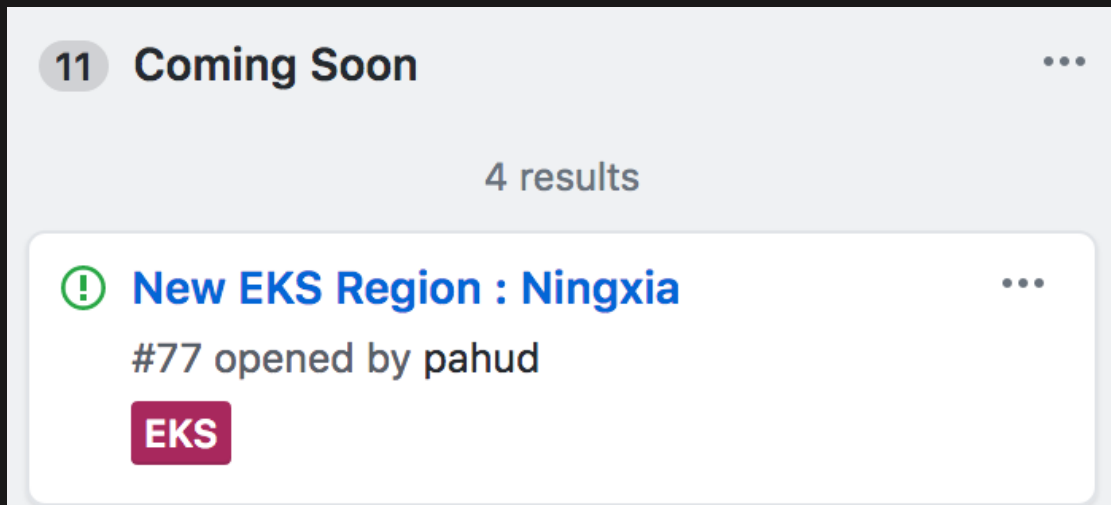
Network Load Balancer with the NGINX Ingress Controller on Amazon EKS

- Scale to millions of requests per second
- Long-lived TCP connections
- Source/remote address preservation
- SSL/TLS Termination



Kubernetes in AWS China Regions

Amazon EKS is Coming to AWS Ningxia Region



<https://github.com/aws/containers-roadmap/issues/77>

AWS中国宁夏区域/北京区域，快速Kops部署K8S集群

当前版本

kops-cn 专案保持跟[上游kops专案](#)版本一致, 上游最新的kops版本可以在[kubernetes/kops/releases/latest](#)查看, 而kops对应的建议k8s版本, 则可以从[stable channels](#)查看。

目前提供 1.12 与 1.13 两个主要版本提供选择, default是 1.12 , 但也可以指定使用 1.13 .

此 README 文件最后一次更新时间所对应的稳定版本是：

主版本	Kops最新版本	K8s搭配版本	AMI
1.12	1.12.3(#99)	1.12.9	kope.io/k8s-1.12-debian-stretch-amd64-hvm-ebs-2019-05-13 (#96)
1.13	1.13.0(#101)	1.13.5	kope.io/k8s-1.12-debian-stretch-amd64-hvm-ebs-2019-05-13 (#96)

Serverless x Containers in AWS

CNCF Survey: Cloud Usage in Asia Has Grown 135% Since March 2018

By Kaitlyn Barnard | November 13, 2018 | December 3rd, 2018 | Blog

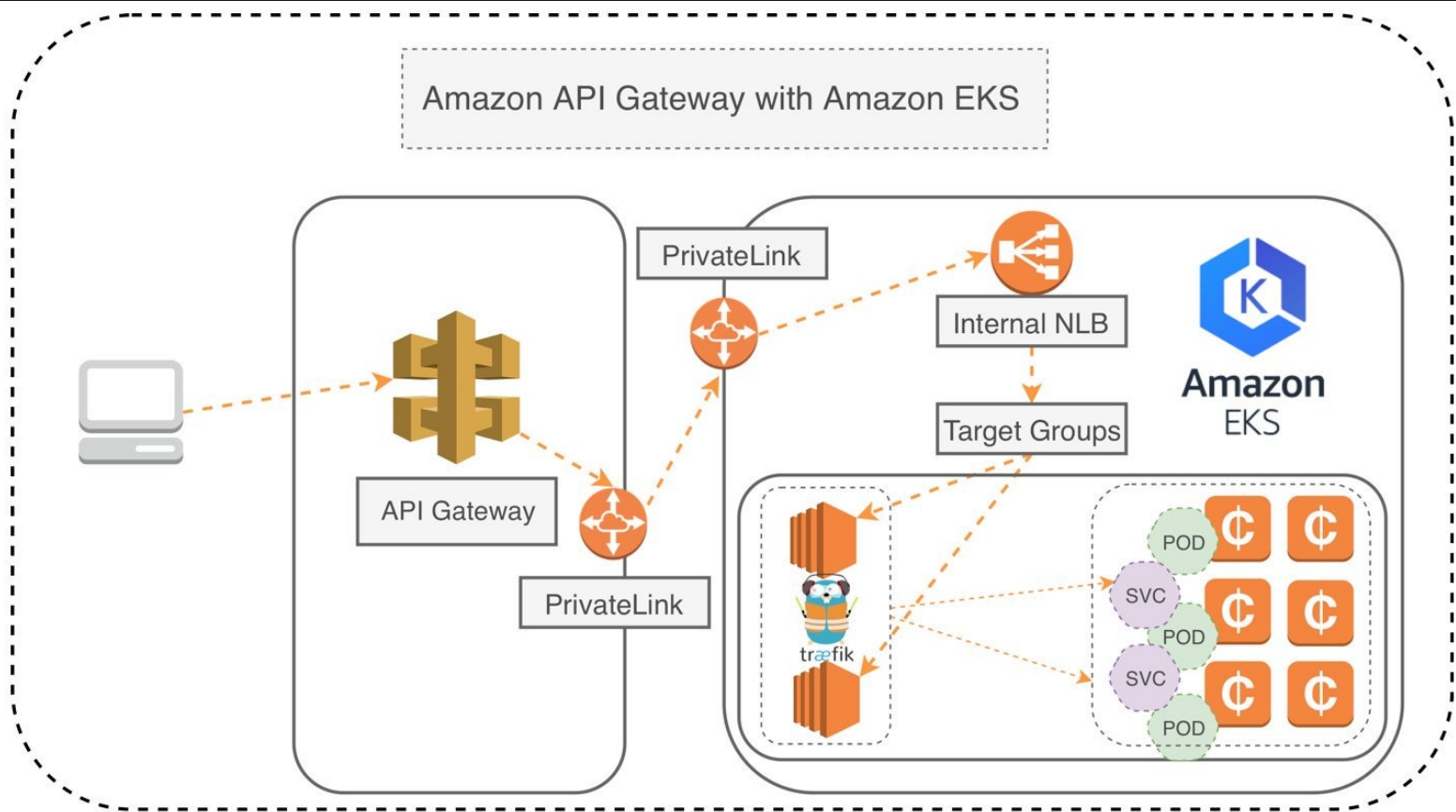
“Use of serverless technology in Asia has spiked 100% with 29% of respondents using installable software and 21% using a hosted platform.”

<https://www.cncf.io/blog/2018/11/13/cncf-survey-china-november-2018/>

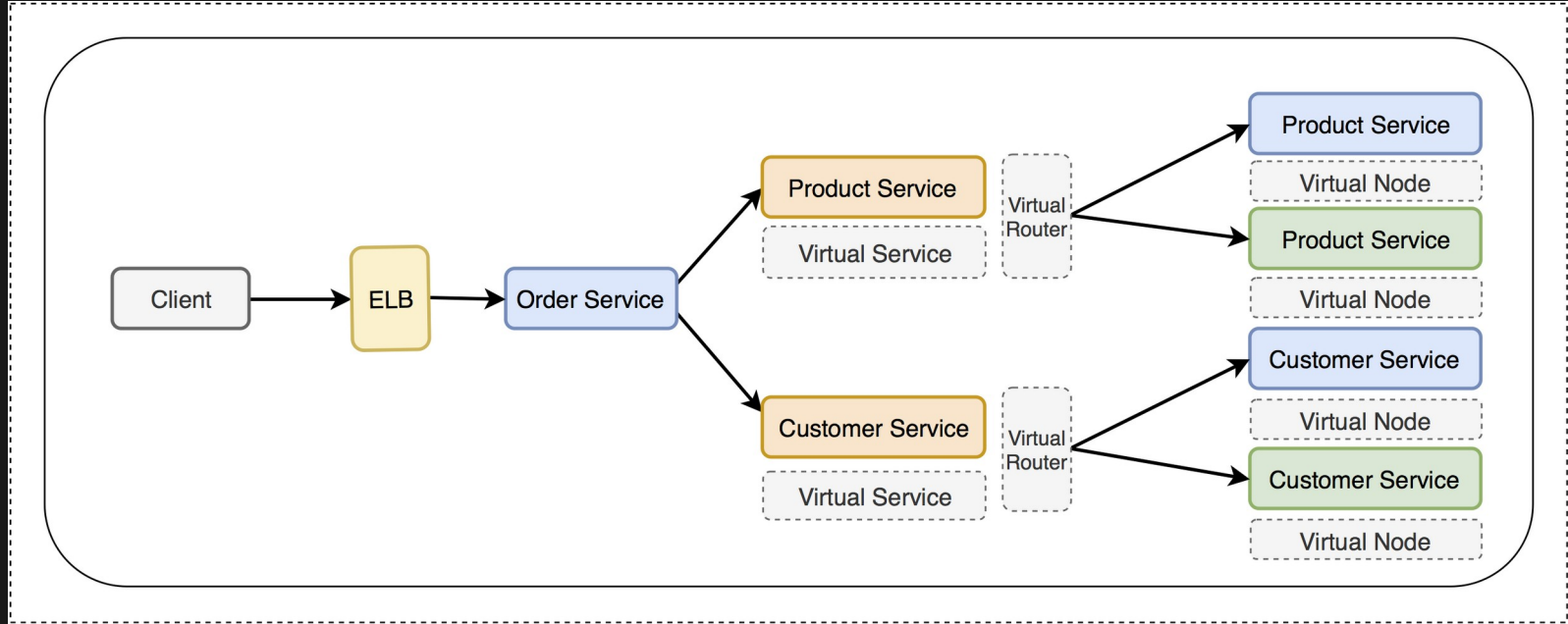
“For hosted serverless platforms, AWS Lambda is the most popular with 11% of respondents citing usage.”

<https://www.cncf.io/blog/2018/11/13/cncf-survey-china-november-2018/>

Amazon API Gateway with Amazon EKS

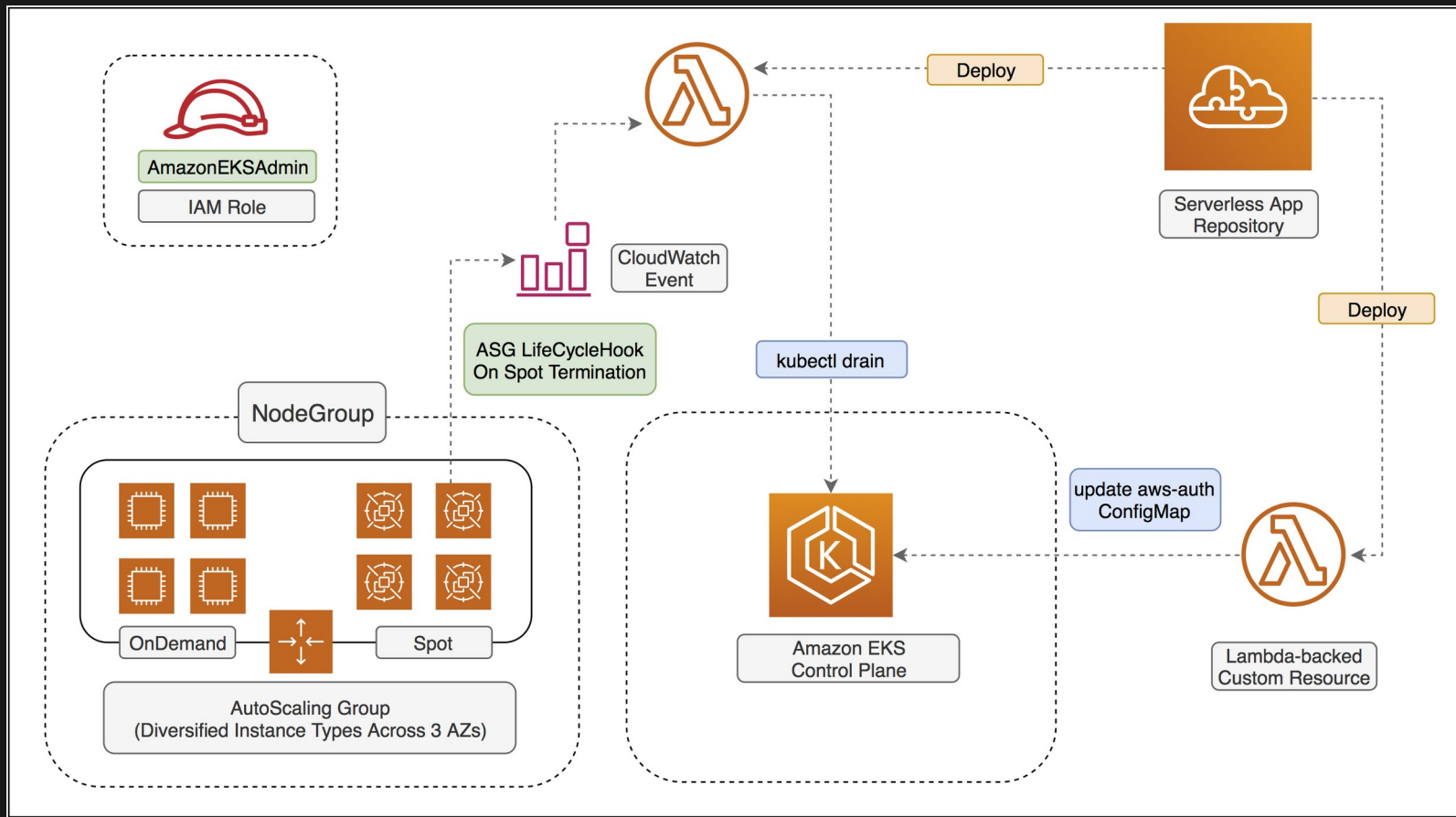


Amazon EKS Canary Deployment with AWS Step Function



<https://github.com/aws-samples/eks-canary-deployment-stepfunction>

Amazon EKS Reference Architecture with AWS Lambda



Distributing to Global

Cross-Regional CodePipeline for Kubernetes Global Distribution

The screenshot displays the AWS CodePipeline console interface. The left sidebar shows the navigation menu with 'CodePipeline' selected. The main content area shows the details of the 'eks-global-PL' pipeline. The 'Source' stage is completed, and the 'Deploy' stage is currently in progress, highlighted by a red box. The 'Deploy' stage is configured with multiple parallel actions across different AWS regions, each using 'AWS CloudFormation' as the provider. The regions listed are Oregon, Tokyo, Seoul, Singapore, Sydney, Mumbai, Virginia, Ohio, Frankfurt, Ireland, London, and Paris. Each action is marked as 'In progress - 1 minute ago'.

Developer Tools **CodePipeline**

Source • CodeCommit
Build • CodeBuild
Deploy • CodeDeploy
▼ Pipeline • CodePipeline
Getting started
Pipelines
Pipeline
History

Feedback
Return to the old experience

Developer Tools > CodePipeline > Pipelines > eks-global-PL

eks-global-PL Edit View history Release change

View current revisions

Source

CodeRepo
GitHub
Succeeded - 1 minute ago
f21234d2
f21234d2 CodeRepo: Merge branch 'master' of github.com:pahud/eks-templates

Disable transition

Deploy View current revisions

Oregon AWS CloudFormation In progress - 1 minute ago Details	Tokyo AWS CloudFormation In progress - 1 minute ago Details	Seoul AWS CloudFormation In progress - 1 minute ago Details	Singapore AWS CloudFormation In progress - 1 minute ago Details	Sydney AWS CloudFormation In progress - 1 minute ago Details
Mumbai AWS CloudFormation In progress - 1 minute ago Details	Virginia AWS CloudFormation In progress - 1 minute ago Details	Ohio AWS CloudFormation In progress - 1 minute ago Details	Frankfurt AWS CloudFormation In progress - 1 minute ago Details	Ireland AWS CloudFormation In progress - 1 minute ago Details
London AWS CloudFormation In progress - 1 minute ago Details	Paris AWS CloudFormation In progress - 1 minute ago Details			

f21234d2 CodeRepo: Merge branch 'master' of github.com:pahud/eks-templates

But, since we have CDK today...

```
import 'source-map-support/register';
import cdk = require('@aws-cdk/core');
import { EksStack } from '../lib/eks';
```

- Import our custom EKS construct as a stack

```
const app = new cdk.App();
```

```
const env = {
  region: app.node.tryGetContext('region') || process.env.CDK_INTEG_REGION || process.env.CDK_DEFAULT_REGION,
  account: app.node.tryGetContext('account') || process.env.CDK_INTEG_ACCOUNT || process.env.CDK_DEFAULT_ACCOUNT
};
```

```
new EksStack(app, 'CdkEksUE1', { env: { region: 'us-east-1' } })
new EksStack(app, 'CdkEksUE2', { env: { region: 'us-east-2' } })
new EksStack(app, 'CdkEksUW2', { env: { region: 'us-west-2' } })
new EksStack(app, 'CdkEksAE1', { env: { region: 'ap-east-1' } })
new EksStack(app, 'CdkEksAS1', { env: { region: 'ap-south-1' } })
new EksStack(app, 'CdkEksAN1', { env: { region: 'ap-northeast-1' } })
new EksStack(app, 'CdkEksAN2', { env: { region: 'ap-northeast-2' } })
new EksStack(app, 'CdkEksEC1', { env: { region: 'eu-central-1' } })
new EksStack(app, 'CdkEksEW1', { env: { region: 'eu-west-1' } })
new EksStack(app, 'CdkEksEW2', { env: { region: 'eu-west-2' } })
new EksStack(app, 'CdkEksEW3', { env: { region: 'eu-west-3' } })
new EksStack(app, 'CdkEksEN1', { env: { region: 'eu-north-1' } })
```

- And distribute it Globally :-)
- With just 25 lines of TypeScript :-)

Thank you!



Pahud Hsieh

Specialist SA, Serverless
Amazon Web Services



@pahudnet



@pahud

hunhsieh@amazon.com