# Kubernetes on AWS

## with Amazon EKS

Pradyumna Dash
Solutions Architect, AWS

Jan 2019

Email: pradyd@amazon.co.uk

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Running containers in development is easy…

```
$ vi Dockerfile
$ docker build -t mykillerapp:0.0.1 .
$ docker run -it mykillerapp:0.0.1
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Moving to production is harder…


© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Common questions

- How do I deploy my containers to hosts?
  - How do I do zero downtime or  blue green deployments?
- How do I keep my containers alive?
- How can my containers talk to each other?
  - Linking? Service Discovery?
- How can I configure my containers at runtime?
  - What about secrets?
- How do I best optimise my "pool of compute”?

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# What is Kubernetes?

Open source container management platform

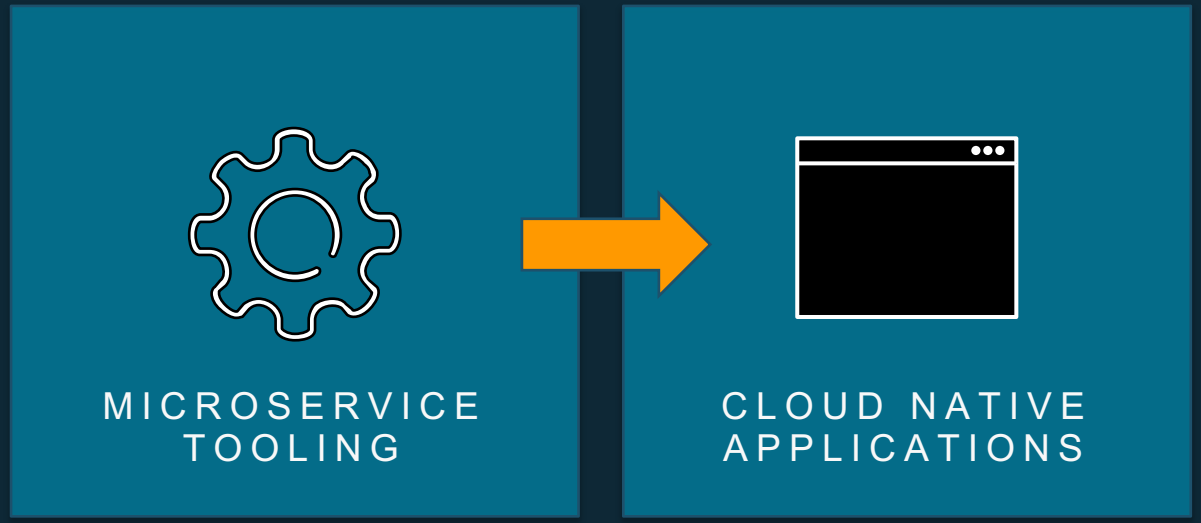
Helps you run containers at scale

Gives you primitives for building modern applications

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Cloud-native applications




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# 51%

of Kubernetes workloads run on AWS today

— Cloud Native Computing Foundation

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# “Run Kubernetes for me.”

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# “Native AWS Integrations.”

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

"An Open Source Kubernetes Experience."

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Amazon EKS

## ELASTIC CONTAINER SERVICE FOR KUBERNETES (EKS)

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws














© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.


Amazon EKS is certified Kubernetes conformant so you can use all existing plugins and tooling from the Kubernetes community.

Any application running on any standard Kubernetes environment is fully compatible.

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Kubernetes Developer Experience

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Deploying some containers

## $ kubectl run nginx --image=nginx --replicas=3 --port=80

```
deployment "nginx" created
```

## $ kubectl get deployments

```
NAME       DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx      3         3         3            3           5s
```

## $ kubectl get pods

```
NAME                     READY     STATUS    RESTARTS   AGE
nginx-4293833666-20vr8   1/1       Running   0          2m
nginx-4293833666-3gzfw   1/1       Running   0          2m
nginx-4293833666-7nBiH   1/1       Running   0          2m
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Exposing pods as services

Three options:

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

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Services: ClusterIP

$ kubectl run nginx --image=nginx --replicas 3 --port=80
$ kubectl expose deployment nginx
$ kubectl get services

```
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
nginx        ClusterIP      100.67.104.10   <none>        80/TCP    17s
```

Now all hosts can connect to 100.67.104.10 (or via DNS as nginx)

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Services: LoadBalancer

```
$ kubectl run nginx --image=nginx --replicas 3 --port=80
$ kubectl expose deployment nginx --type=LoadBalancer
$ kubectl get services -o=wide
```

```
NAME      TYPE           CLUSTER-IP      EXTERNAL-IP                                                                PORT(S)
nginx     LoadBalancer   100.70.217.164  a5cefe533ac1d11e7a38f0a67818e472-1987464052.eu-west-1.elb.amazonaws.com     80:31108/TCP
```

| DNS name | Availability Zones | Type | Port Configuration |
|---|---|---|---|
| a5cefe533ac1d11e7a38f0a67818e472-1987464052.eu-west-1.elb.amazonaws.com | eu-west-1b, eu-west-1c, eu-west-1a | classic | 80 (TCP) forwarding to 31108 (TCP) |

| Instance ID | Name | Availability Zone | Status | Actions |
|---|---|---|---|---|
| i-0478980a1a86faa09 | micro.k8s.demothe.cloud | eu-west-1b | InService | Remove from Load Balancer |
| i-0885393f80f3db7de | micro.k8s.demothe.cloud | eu-west-1a | InService | Remove from Load Balancer |
| i-0d701a00358fb084f | micro.k8s.demothe.cloud | eu-west-1c | InService | Remove from Load Balancer |
| i-0a3b00eeabdf3b0ce | micro.k8s.demothe.cloud | eu-west-1c | InService | Remove from Load Balancer |
| i-08617f4b745d3bb74 | micro.k8s.demothe.cloud | eu-west-1b | InService | Remove from Load Balancer |
| i-077d170e688971c98 | micro.k8s.demothe.cloud | eu-west-1a | InService | Remove from Load Balancer |

# Declarative Configuration as Code

$ kubectl get deployment nginx -oyaml > deployment.yaml

$ kubectl apply –f deployment.yaml

$ kubectl delete –f deployment.yaml

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Diagnose & inspect

## $ kubectl logs <pod>

```
100.103.136.0 - - [06/Oct/2017:11:30:47 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.54.0" "-"
```

## $ kubectl attach <pod>

```
100.103.136.0 - - [06/Oct/2017:11:30:47 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.54.0" "-"
100.103.136.0 - - [06/Oct/2017:11:37:25 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.54.0" "-"
100.103.136.0 - - [06/Oct/2017:11:37:25 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.54.0" "-"
100.103.136.0 - - [06/Oct/2017:11:37:25 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.54.0" "-"
100.103.136.0 - - [06/Oct/2017:11:37:25 +0000] "GET / HTTP/1.1" 200 612 "-" "curl/7.54.0" "-"
<ctrl + c>
```

## $ kubectl exec -it <pod> -- /bin/bash

```
root@nginx-1423793266-xw3km:/#
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Local experience

**Docker for Mac** Easy to get up and running with a local Kubernetes installation in minutes on MacOS. Only available in 'edge' channel today.

**Minikube** is a tool that makes it easy to run Kubernetes locally. Minikube runs a single-node Kubernetes cluster inside a VM on your laptop for users looking to try out Kubernetes or develop with it day-to-day.

https://github.com/kubernetes/minikube

**Telepresence** allows you to transparently mount your local development folder into a remote cluster.

https://www.telepresence.io/

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Demo

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

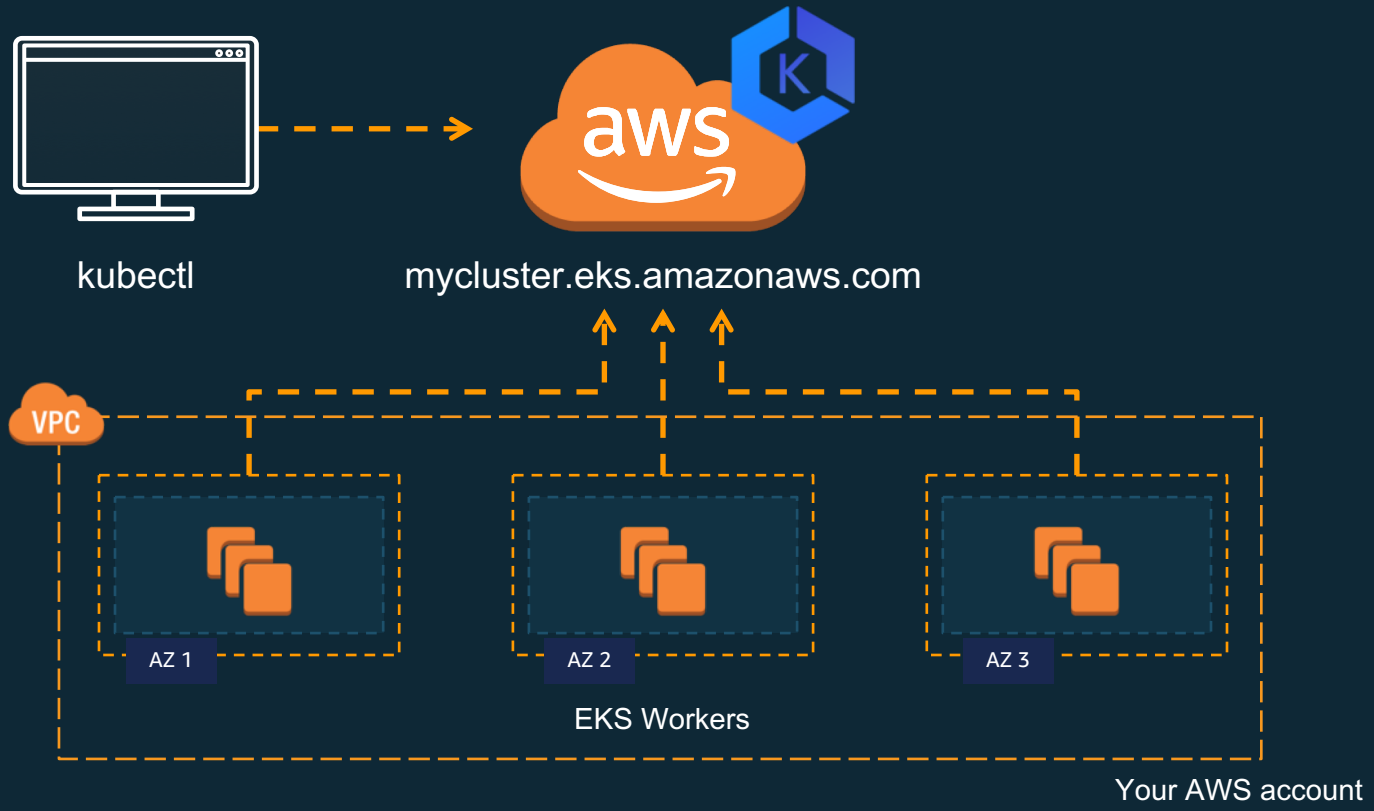
# EKS Architecture

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.


© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

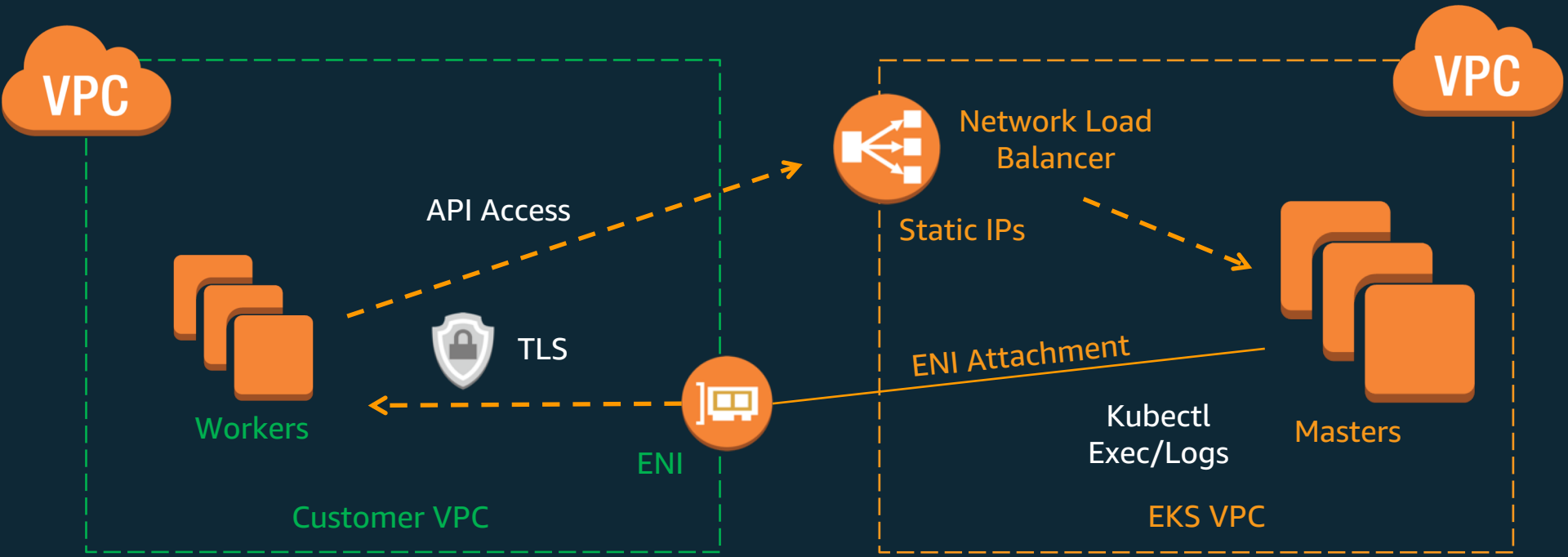

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Amazon EKS




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

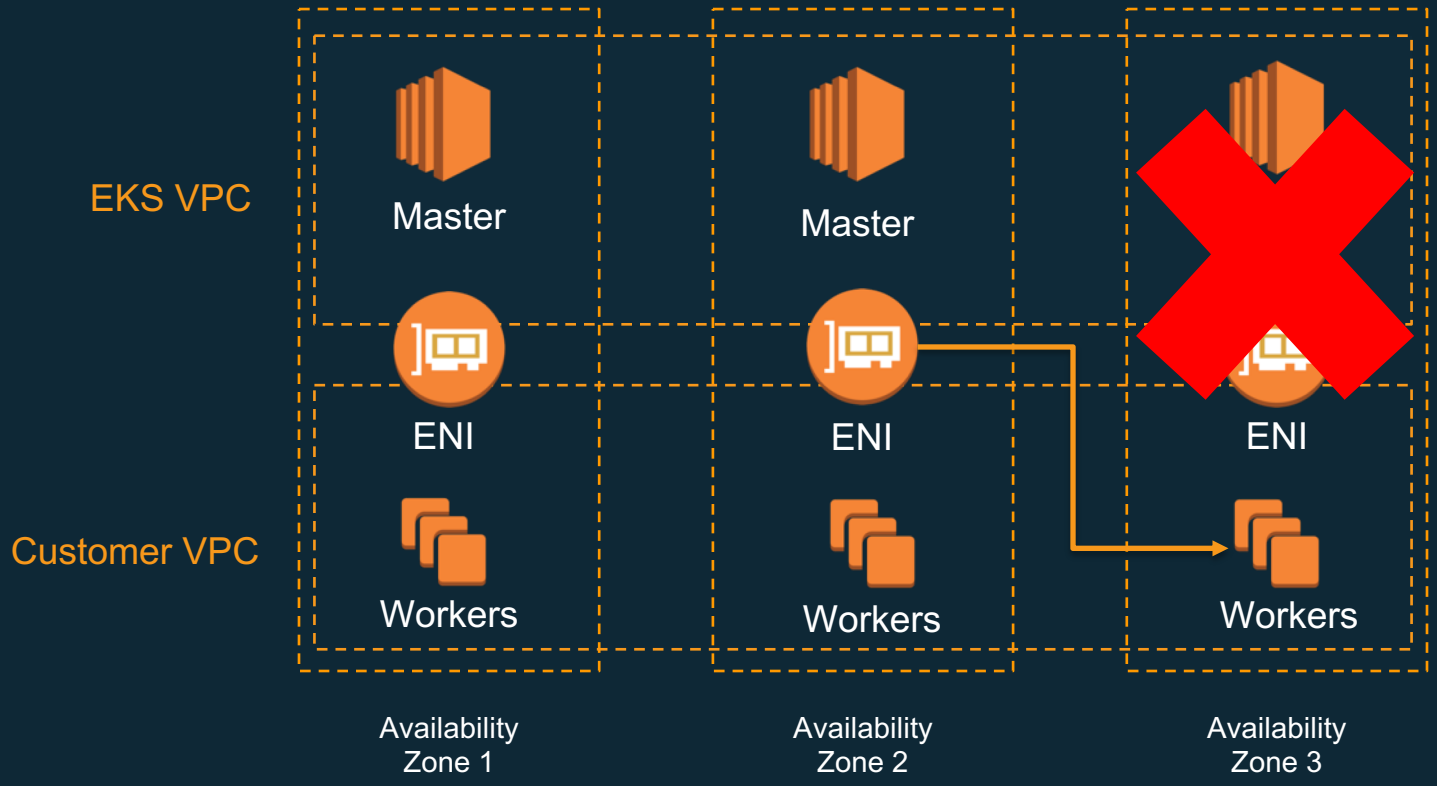
# Cross-account Kubernetes




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

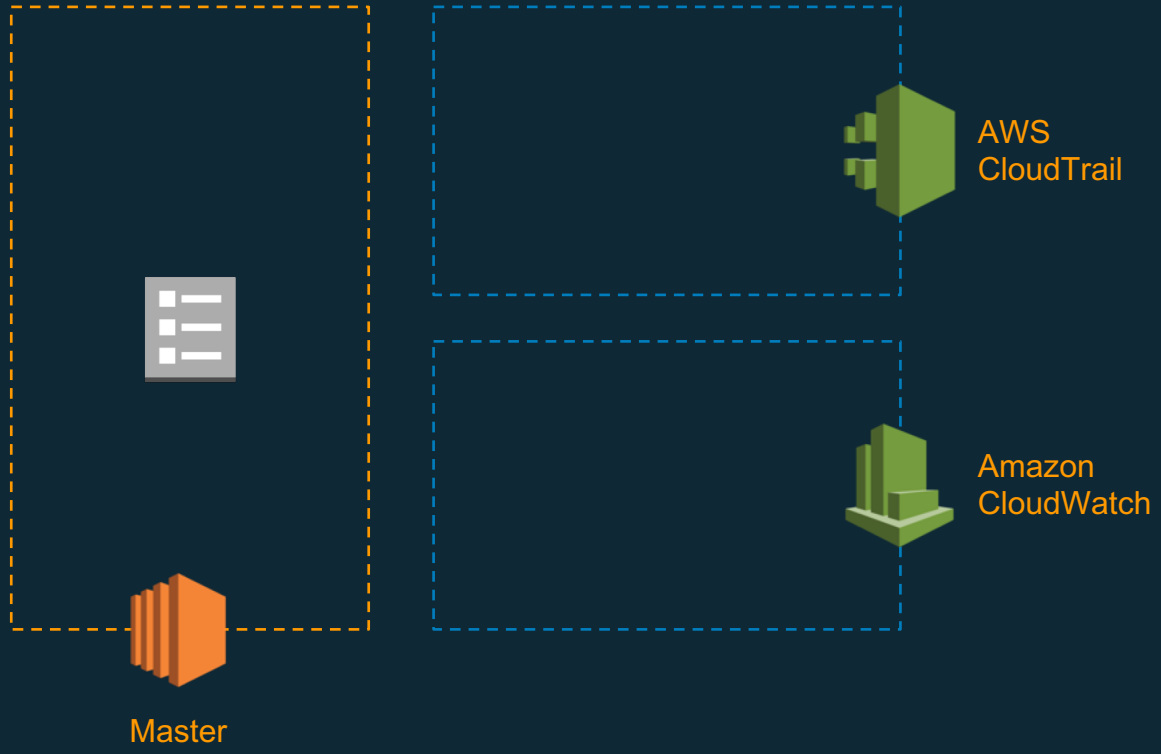
aws

# EKS Cross-Account Networking: Availability Zones




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Master access and visibility




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# What about K8s/etcd upgrades?

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

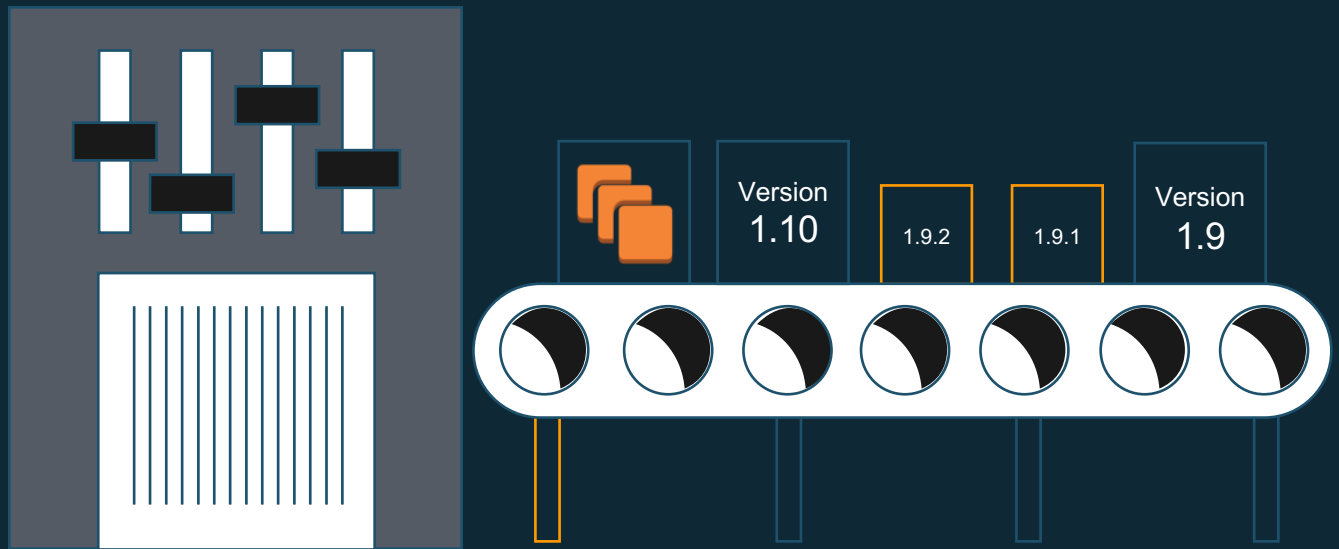
aws

# Kubernetes Upgrades


© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Kubernetes Upgrades




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# How do I provision EKS Worker Nodes?

HashiCorp
Packer

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# How do I get started?

**eksctl CLI**
**https://eksctl.io**

***CloudFormation, APIs, CLI etc available too**

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

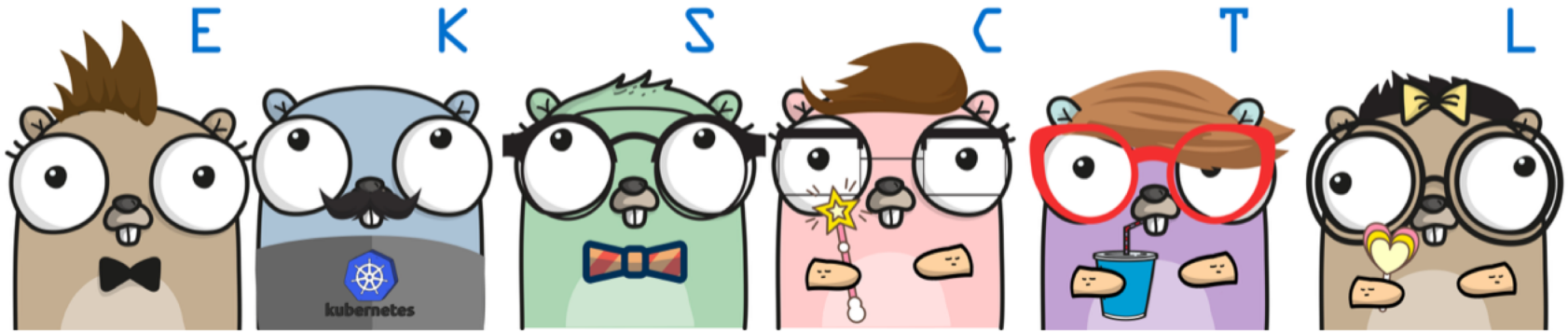
aws

# `eksctl` - a CLI for Amazon EKS

circleci passing | coverage 15% | go report A

`eksctl` is a simple CLI tool for creating clusters on EKS - Amazon's new managed Kubernetes service for EC2. It is written in Go, and based on Amazon's official CloudFormation templates.

You can create a cluster in minutes with just one command – **`eksctl create cluster`**!




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Kubernetes / AWS integrations

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

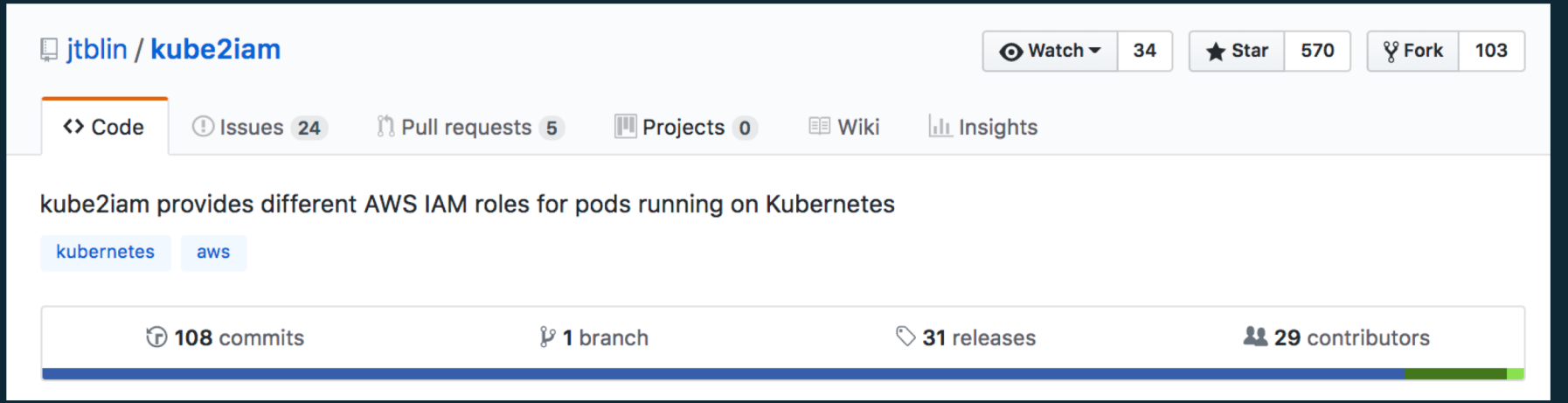
aws

*AWS Integration:*

# *Identity and Access Management (IAM)*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# I want to give a pod permissions to an AWS service




- Runs as a DaemonSet on your workers
- Creates iptables rules to redirect metadata service to kube2iam
- Add annotations to your pods to grant them AWS IAM Roles

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# kube2iam example

```yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  template:
    metadata:
      annotations:
        iam.amazonaws.com/role: arn:aws:iam:123567989012/role/nginx-role
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.9.1
        ports:
        - containerPort: 80
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# I want to use AWS accounts to operate Kubernetes

An open source approach to integrating
AWS IAM authentication with Kubernetes


© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# ~/.kube/config (with IAM)

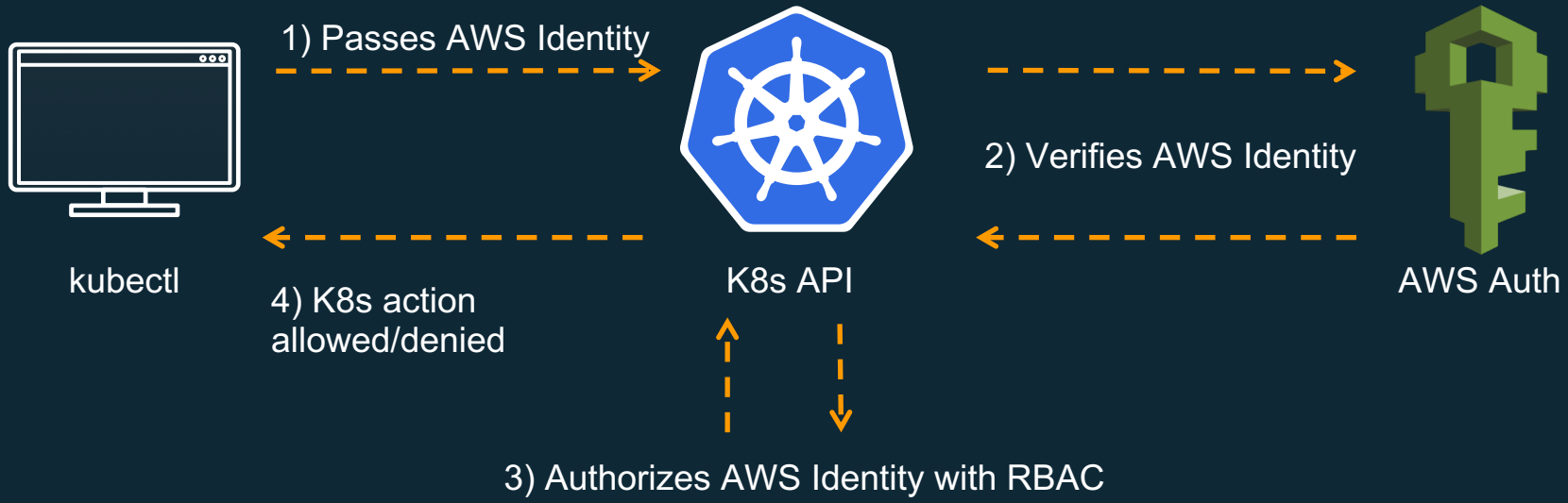
```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: REDACTED
    server: https://F7C341FA2AF732DAC328FB150F48979C.sk1.us-west-2.eks.amazonaws.com
  name: eks
contexts:
- context:
    cluster: eks
    user: eks
  name: eks
current-context: eks
kind: Config
users:
- name: eks
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      command: heptio-authenticator-aws
      args:
      - token
      - -i
      - <cluster name>
```

Config file is no longer user-specific ☺

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# IAM Authentication with kubectl



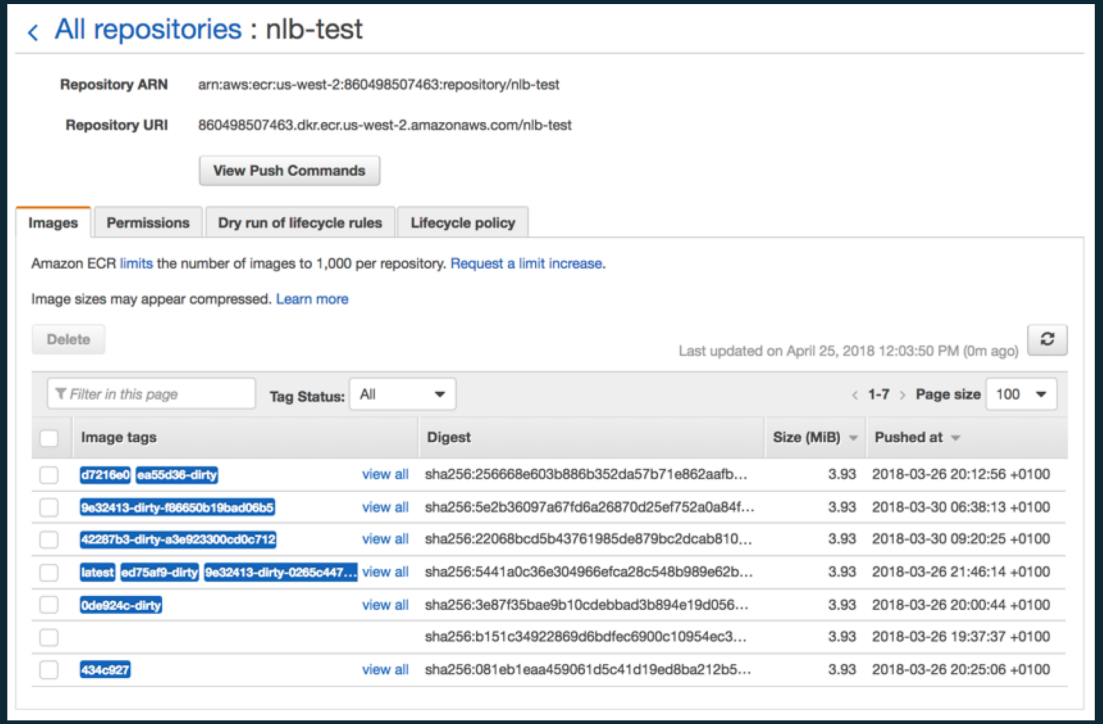

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

*AWS Integration:*

# *Container Registry: Amazon ECR*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Amazon ECR




- Simple to create
- High Availability by default
- IAM permissions
- Lifecycle rules
- Encrypted at rest
- Billed on storage

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

*AWS Integration:*

# *Helm*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Helm

- Package manager that allows you to bundle up deployment resources and publish them


© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Using helm

```
❯ helm search nginx
NAME                    CHART VERSION   APP VERSION     DESCRIPTION
stable/nginx-ingress    0.12.3          0.12.0          An nginx Ingress controller that uses ConfigMap...
stable/nginx-lego       0.3.1                           Chart for nginx-ingress-controller and kube-lego
stable/gcloud-endpoints 0.1.0                           Develop, deploy, protect and monitor your APIs ...

❯ helm list
NAME            REVISION        UPDATED                         STATUS          CHART           NAMESPACE
kube2iam        1               Sat Mar 31 20:02:48 2018        DEPLOYED        kube2iam-0.8.3  kube-system

> helm install stable/nginx-ingress --name nginx-ingress --set rbac.create=true

[displays README + information about deployment]

❯ kubectl get services -owide

NAME                            TYPE          CLUSTER-IP      EXTERNAL-IP                                                                 PORT(S)                      AGE     SELECTOR
kubernetes                      ClusterIP     172.20.0.1      <none>                                                                      443/TCP                      8d      <none>
nginx-ingress-controller        LoadBalancer  172.20.23.100   af403dd953b5c11e8a6d20a4d0bdc52d-2083699508.eu-west-1.elb.amazonaws.com     80:30501/TCP,443:30740/TCP   53s     app=nginx-ingress,component=controller,release=nginx-ingress
nginx-ingress-default-backend   ClusterIP     172.20.20.7     <none>                                                                      80/TCP                       53s     app=nginx-ingress,component=default-backend,release=nginx-ingress
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Hosting Helm repositories

- Anywhere that serves HTTP can host a helm repo
- There's a handy plugin for S3!
- This means IAM Role = auth for your repo ☺

- https://github.com/hypnoglow/helm-s3

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

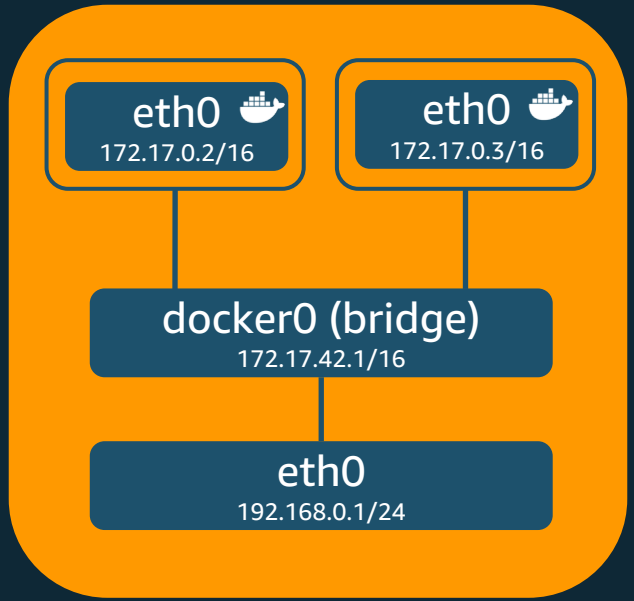
aws

*AWS Integration:*

# *Networking*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Docker: Bridged networking




```
$ docker run -p 8080:80 nginx
```

Running a container with ports mapped sets up a NAT with iptables

```
192.168.0.1:80 -> 172.17.0.2:8080
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

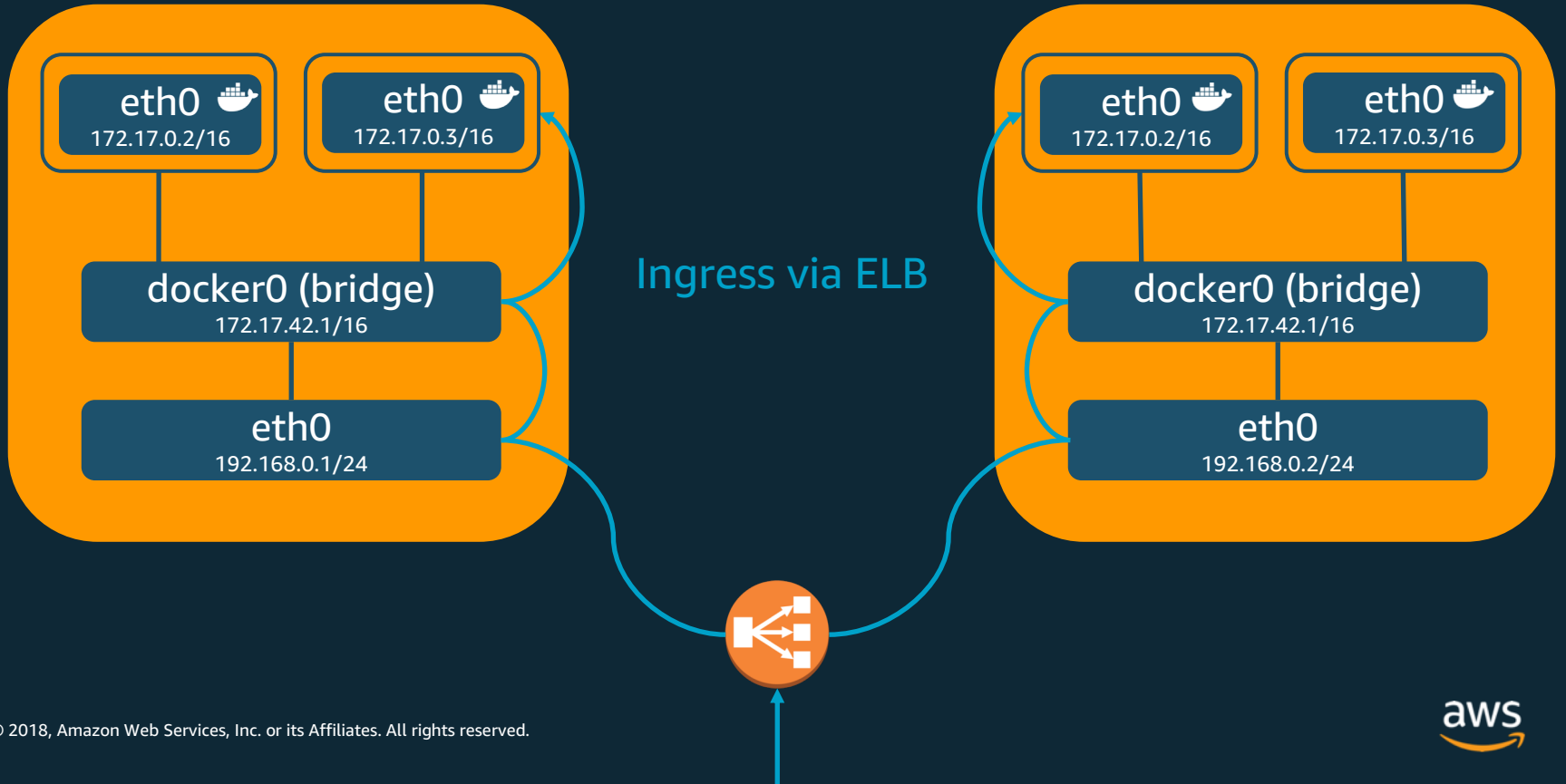
aws

# Docker: Bridged networking


© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

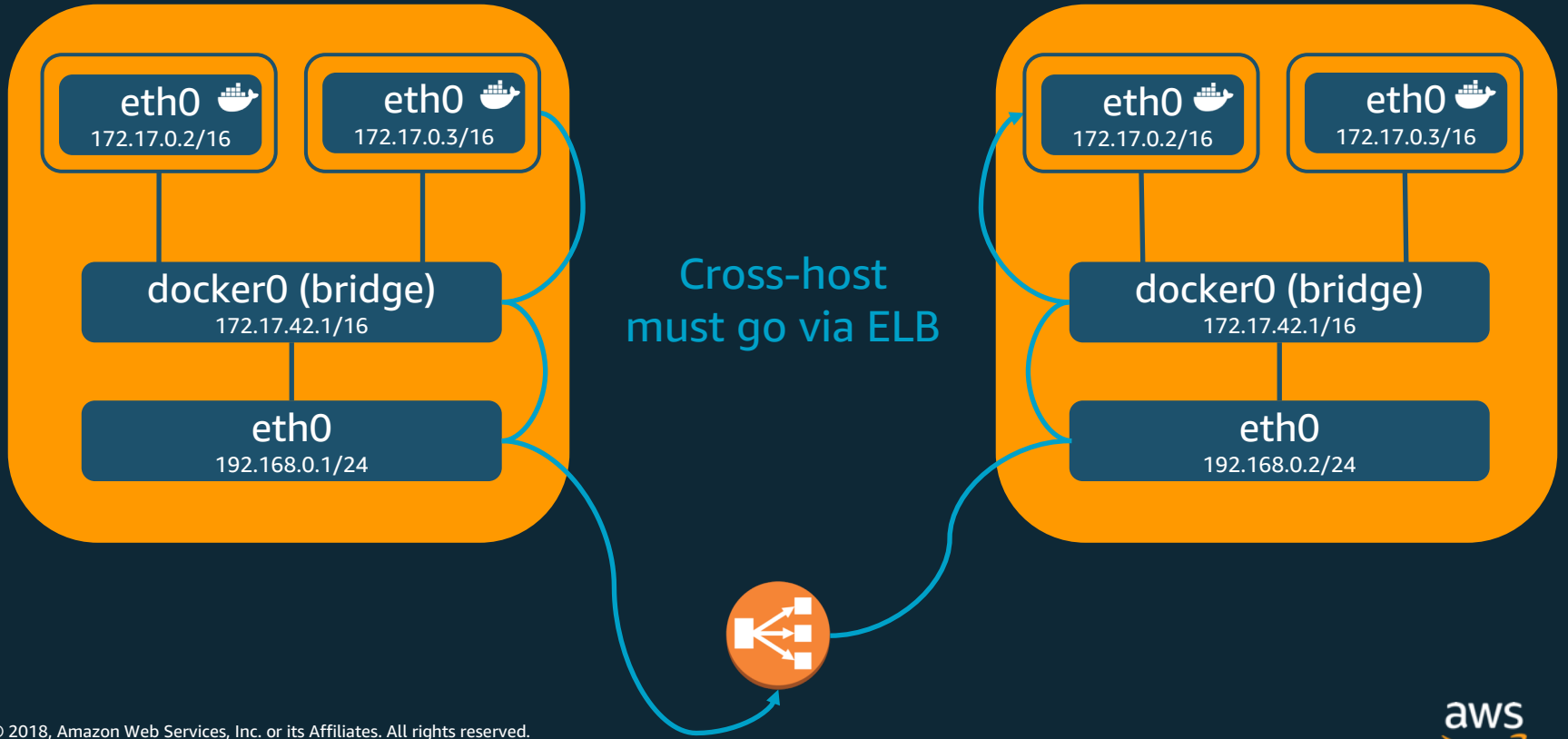
aws

# Docker: Bridged networking



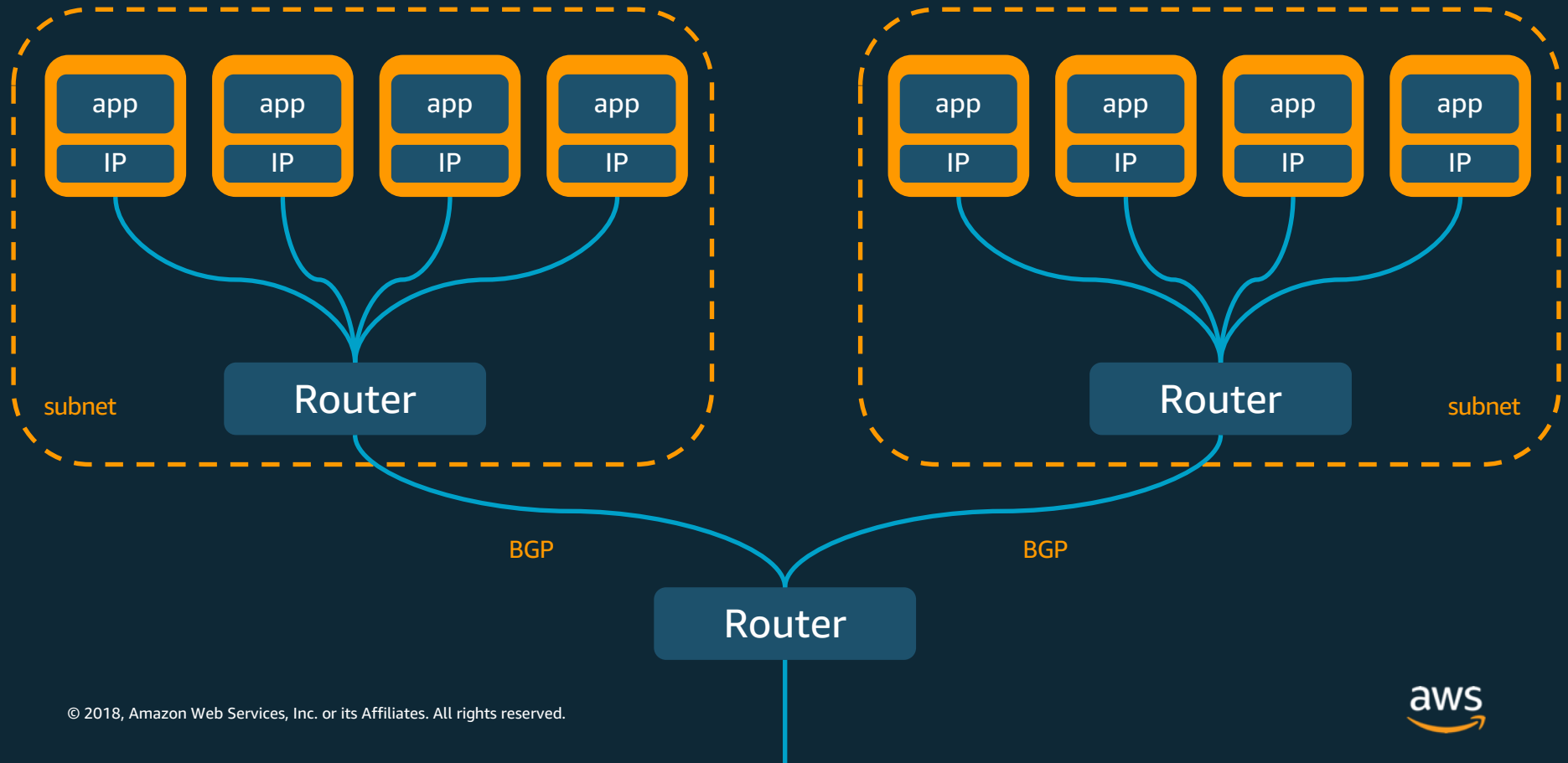

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Docker: Bridged networking



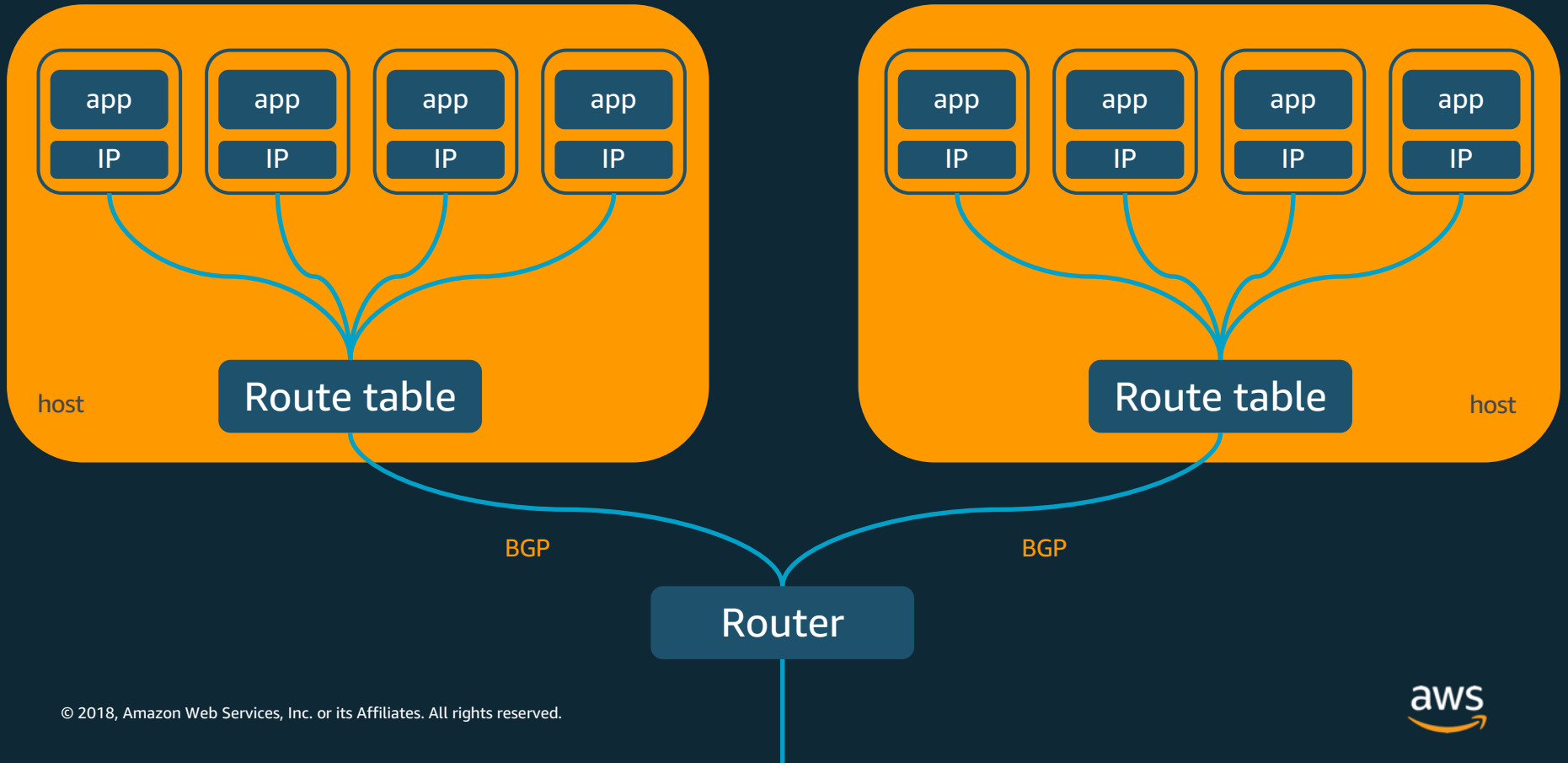

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Back to basics




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Kubernetes networking with Calico CNI plugin




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Kubernetes networking with Calico CNI plugin

- Unlike Docker, there isn't a bridge network*
- All containers can communicate with all other containers without NAT.
- All nodes can communicate with all containers (and vice-versa) without NAT.
- The IP that a container sees itself as is the same IP that others see it as.
- All containers within a pod can access each other on localhost (coordination to avoid conflicts required).

http://blog.mbrt.it/2017-10-01-demystifying-container-networking/
https://kubernetes.io/docs/concepts/cluster-administration/networking/#kubernetes-model

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# A host is connected to the cluster


- Each host gets a network range for containers

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

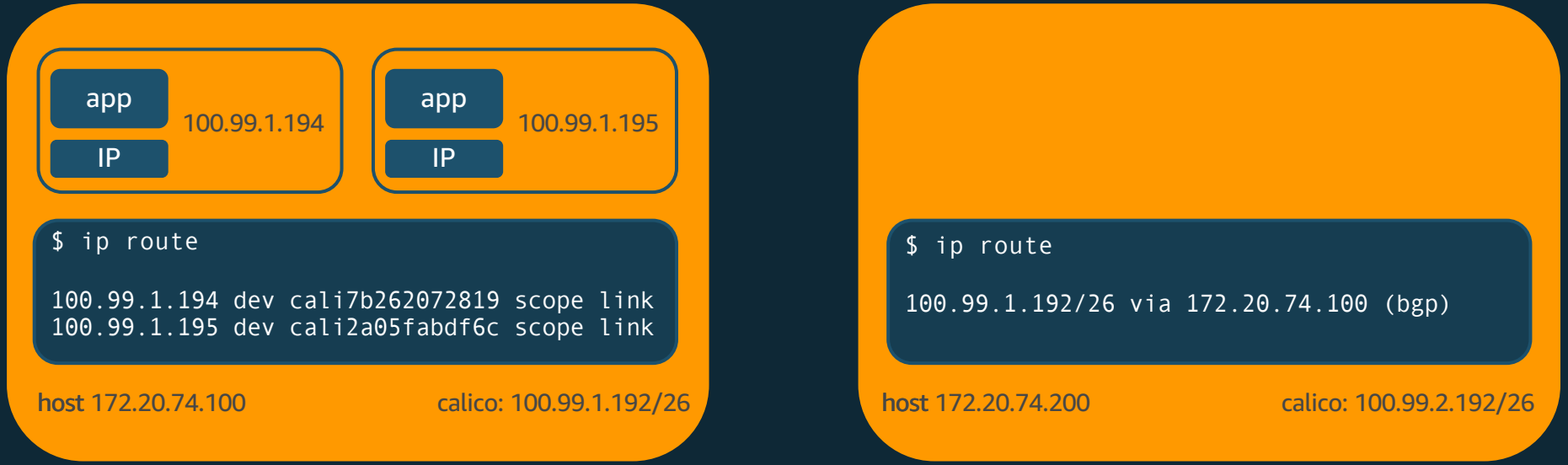
# Some pods are scheduled


- Each host gets a network range for containers
- Each pod gets an IP address in that range
- Now containers on the same host can communicate

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Routes propagated via BGP




Route for the host's /26 propagated to all other hosts in the cluster via BGP

Cross host networking is now possible ☺

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# But there is an easier way…

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Native VPC networking with CNI plugin

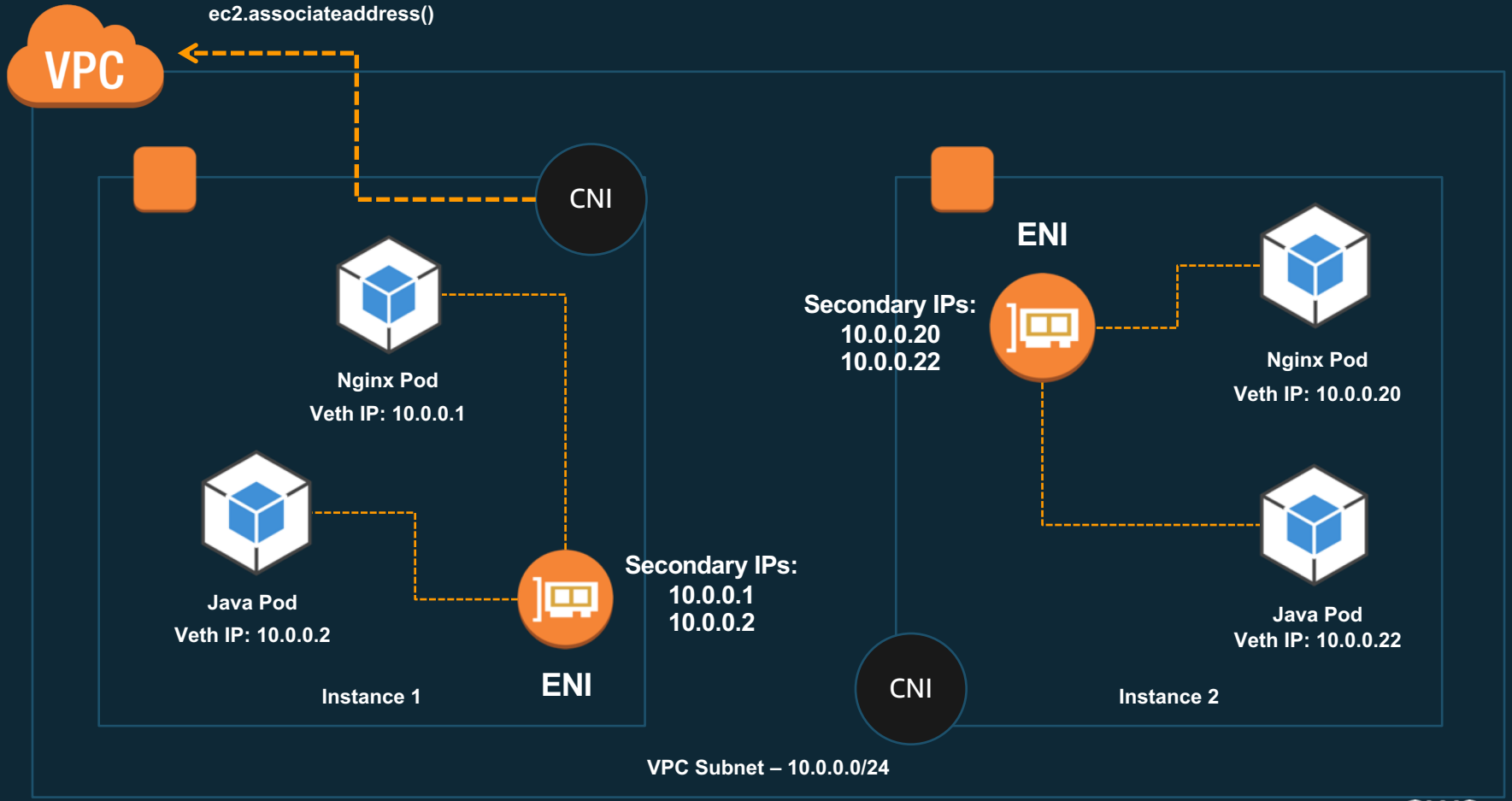
Pods have the same VPC address inside the pod as on the VPC

Simple, secure networking

Open source and on Github

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

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

TIGERA

PROJECT CALICO

Kubernetes Network Policies enforce network security rules

Calico is the leading implementation of the network policy API

Open source, active development (>100 contributors)

Commercial support available from Tigera

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

*AWS Integration:*

# *Load Balancers*

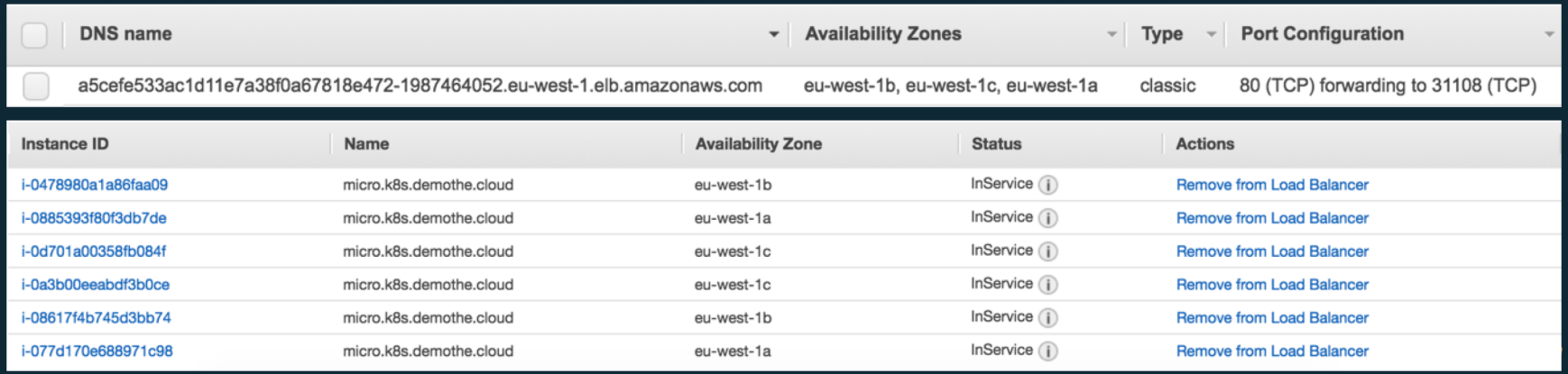
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Services: LoadBalancer

```
$ kubectl run nginx --image=nginx --replicas 3 --port=80
$ kubectl expose deployment nginx --type=LoadBalancer
$ kubectl get services -o=wide
```

```
NAME      TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)
nginx     LoadBalancer   100.70.217.164  a5cefe533ac1d11e7a38f0a67818e472-1987464052.eu-west-1.elb.amazonaws.com   80:31108/TCP
```



| DNS name | Availability Zones | Type | Port Configuration |
|---|---|---|---|
| a5cefe533ac1d11e7a38f0a67818e472-1987464052.eu-west-1.elb.amazonaws.com | eu-west-1b, eu-west-1c, eu-west-1a | classic | 80 (TCP) forwarding to 31108 (TCP) |

| Instance ID | Name | Availability Zone | Status | Actions |
|---|---|---|---|---|
| i-0478980a1a86faa09 | micro.k8s.demothe.cloud | eu-west-1b | InService | Remove from Load Balancer |
| i-0885393f80f3db7de | micro.k8s.demothe.cloud | eu-west-1a | InService | Remove from Load Balancer |
| i-0d701a00358fb084f | micro.k8s.demothe.cloud | eu-west-1c | InService | Remove from Load Balancer |
| i-0a3b00eeabdf3b0ce | micro.k8s.demothe.cloud | eu-west-1c | InService | Remove from Load Balancer |
| i-08617f4b745d3bb74 | micro.k8s.demothe.cloud | eu-west-1b | InService | Remove from Load Balancer |
| i-077d170e688971c98 | micro.k8s.demothe.cloud | eu-west-1a | InService | Remove from Load Balancer |

# Configure your load balancers via annotations

```
service.beta.kubernetes.io/ aws-load-balancer-type
service.beta.kubernetes.io/ aws-load-balancer-internal
service.beta.kubernetes.io/ aws-load-balancer-proxy-protocol
service.beta.kubernetes.io/ aws-load-balancer-access-log-emit-interval
service.beta.kubernetes.io/ aws-load-balancer-access-log-enabled
service.beta.kubernetes.io/ aws-load-balancer-access-log-s3-bucket-name
service.beta.kubernetes.io/ aws-load-balancer-access-log-s3-bucket-prefix
service.beta.kubernetes.io/ aws-load-balancer-connection-draining-enabled
service.beta.kubernetes.io/ aws-load-balancer-connection-draining-timeout
service.beta.kubernetes.io/ aws-load-balancer-connection-idle-timeout
service.beta.kubernetes.io/ aws-load-balancer-cross-zone-load-balancing-enabled
service.beta.kubernetes.io/ aws-load-balancer-extra-security-groups
service.beta.kubernetes.io/ aws-load-balancer-ssl-cert
service.beta.kubernetes.io/ aws-load-balancer-ssl-ports
service.beta.kubernetes.io/ aws-load-balancer-ssl-negotiation-policy
service.beta.kubernetes.io/ aws-load-balancer-backend-protocol
service.beta.kubernetes.io/ aws-load-balancer-additional-resource-tags
service.beta.kubernetes.io/ aws-load-balancer-healthcheck-healthy-threshold
service.beta.kubernetes.io/ aws-load-balancer-healthcheck-unhealthy-threshold
service.beta.kubernetes.io/ aws-load-balancer-healthcheck-timeout
service.beta.kubernetes.io/ aws-load-balancer-healthcheck-interval
```

- Draining
- Logging
- SSL Certs
- Tagging
- Security groups
- Health checks

https://github.com/kubernetes/kubernetes/blob/master/pkg/cloudprovider/providers/aws/aws.go

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Network Load Balancer (layer 4)

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

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Application Load Balancer (layer 7)

CoreOS ALB Ingress Controller: Supported by AWS

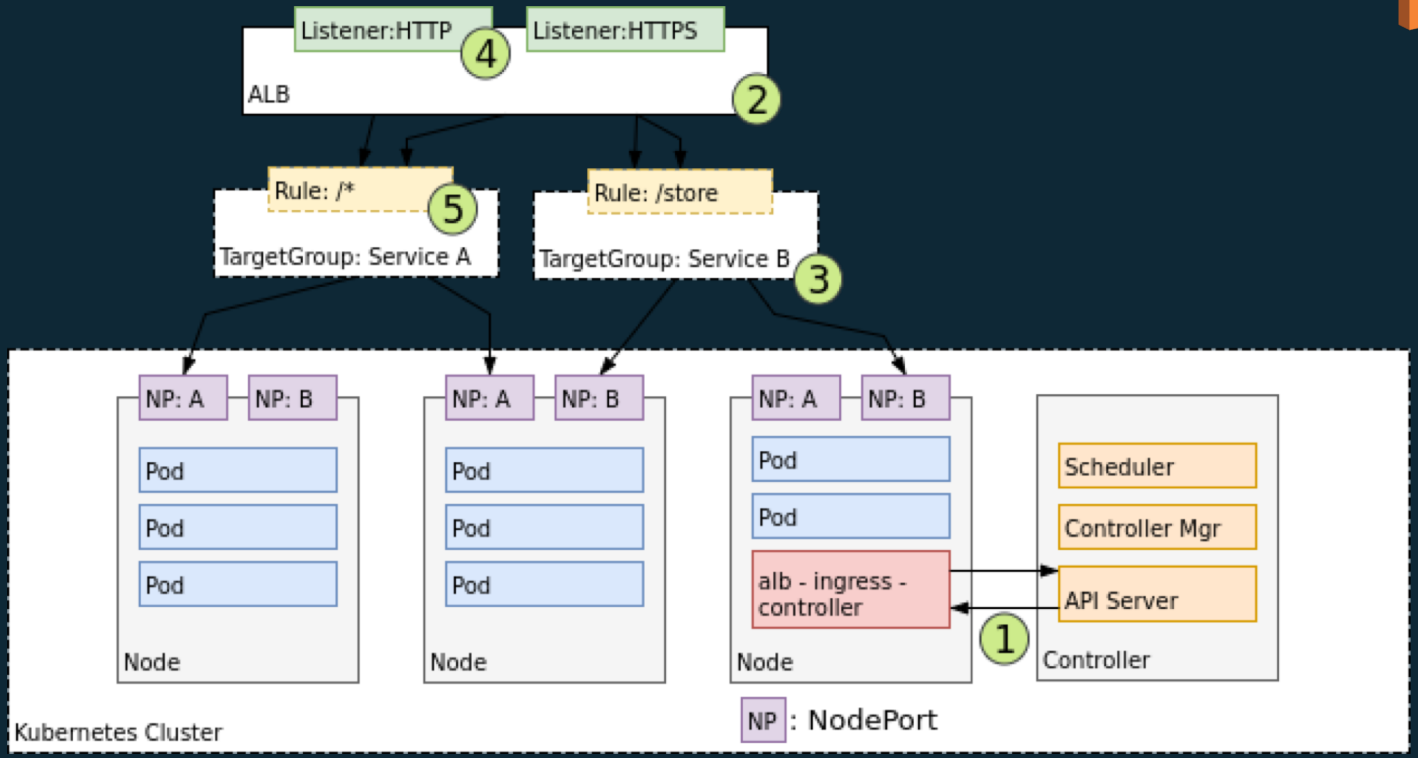
Exposes ALB functionality to Kubernetes via Ingress Resources

Layer 7 load balancing, supports content-based routing by host or path

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Load Balancing




© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

*AWS Integration:*

# *SSL Certificates*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# FREE SSL certificates with AWS Certificate Manager


- Free SSL certificates
- DNS or email verification
- Automatic rotation!
- Wildcard certs, SAN certs
- Can be applied to AWS Load Balancers, CloudFront distributions and Amazon API Gateway
- Configure certificate for load balanced services/ingress via K8s annotation

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

*AWS Integration:*

# *DNS: ExternalDNS*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Automatic Route53 DNS creation for services

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    external-dns.alpha.kubernetes.io/hostname: nginx.demothe.cloud.
spec:
  type: LoadBalancer
  ports:
  - port: 80
    name: http
    targetPort: 80
  selector:
    app: nginx
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# …works with ingress too

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx
  annotations:
    kubernetes.io/ingress.class: "nginx"
spec:
  rules:
  - host: nginx.demothe.cloud
    http:
      paths:
      - backend:
          serviceName: nginx
          servicePort: 80
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

*AWS Integration:*

# *Deployment of other AWS resources*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

# Controllers and Operators

- Controllers in Kubernetes are containers that watch for events, and react to them.
- For example, when a new resource of type xyz is created, speak to the external xyz API and create it
- Operators extend controllers, to add in additional "operations" tasks, such as backups, restores etc

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Etcd Operator

## Overview

The etcd operator manages etcd clusters deployed to Kubernetes and automates tasks related to operating an etcd cluster.

- Create and Destroy
- Resize
- Failover
- Rolling upgrade
- Backup and Restore

There are more spec examples on setting up clusters with different configurations

Read Best Practices for more information on how to better use etcd operator.

Read RBAC docs for how to setup RBAC rules for etcd operator if RBAC is in place.

Read Developer Guide for setting up a development environment if you want to contribute.

See the Resources and Labels doc for an overview of the resources created by the etcd-operator.

https://github.com/coreos/etcd-operator

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# What about deploying other AWS resources

There's an operator for that!

https://github.com/linki/cloudformation-operator

Defines a Custom Resource Definition (CRD) for 'stacks'.

Paste CloudFormation templates into your Kubernetes YAML files, and let the operator deploy/update/manage them for you.

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Deploying AWS resources with K8s

```
apiVersion: cloudformation.linki.space/v1alpha1
kind: Stack
metadata:
  name: my-bucket
spec:
  template: |
    ---
    AWSTemplateFormatVersion: '2010-09-09'

    Resources:
      S3Bucket:
        Type::AWS::S3::Bucket
        Properties:
          BucketName: my-bucket
```

Deploy AWS resources right from your K8s YAML files.

User's don't need AWS permissions, the IAM Role for the host(s) running the operator do.

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# To summarise

- Amazon EKS is Kubernetes Compliant, bakes in best practices and allows you to easily create production-ready clusters in minutes.
- Lots of awesome integrations, but all are open source, and not limited to just Amazon EKS
- Open source Kubernetes == Amazon EKS Kubernetes v1.11

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

# Thank you!

Email: pradyd@amazon.co.uk

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.