aws SUMMIT

kubernetes naws

Tiffany Jernigan @tiffanyfayj
Developer Advocate, AWS
Christoph Kassen @christoph_k
Solutions Architect, AWS

Containers

Packaging

Distribution

Immutable infrastructure

kubernetes

What is kubernetes?

Open source container management platform

Helps you run containers at scale

Gives you primitives for building modern applications

A single extensible API

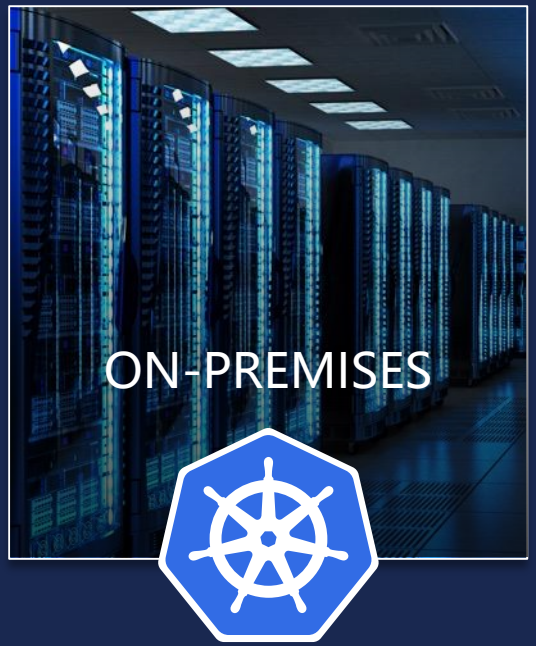
SCALE

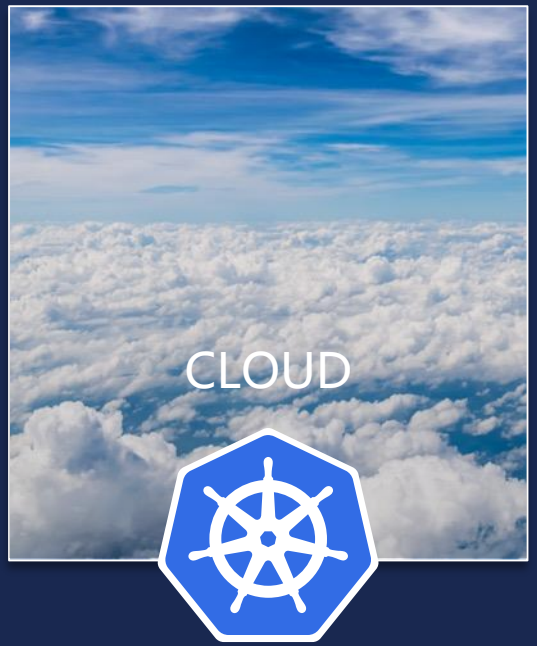
PERFORMANCE

BREADTH

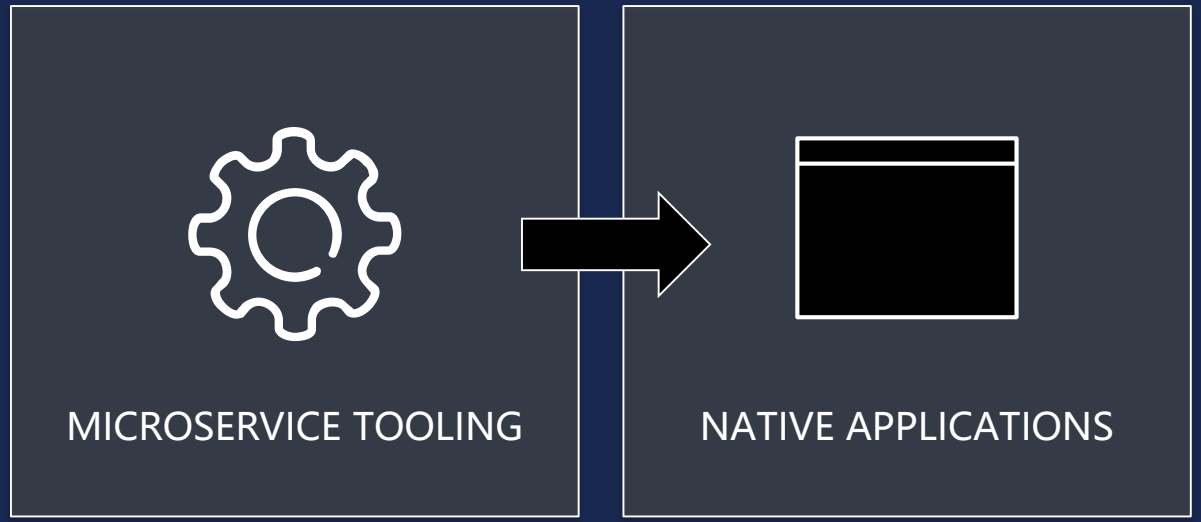
Vibrant and growing community of users and contributors

Kubernetes can be run anywhere!





Cloud-native applications



“Run Kubernetes for me.”

“Native AWS integrations”

“An open source Kubernetes experience.”

Amazon EKS

ELASTIC CONTAINER SERVICE FOR KUBERNETES

GA yesterday 6/5!

Amazon Container Services

Amazon ECS

Amazon EKS

AWS Fargate

Amazon ECR

EKS is Kubernetes Certified

Open Source Kubernetes Community

Kubernetes
https://github.com/kubernetes/kubernetes

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

Heptio AWS Authenticator
https://github.com/heptio/authenticator

Virtual Kubelet
https://github.com/virtual-kubelet/virtual-kubelet/

https://github.com/kubernetes/community/tree/master/sig-aws

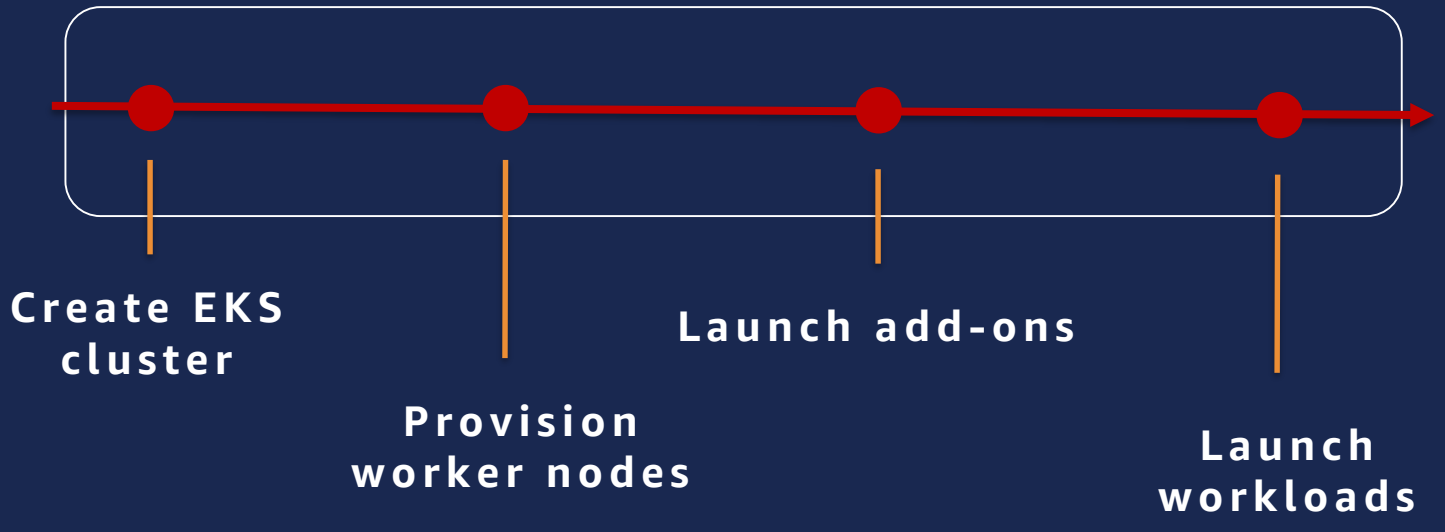
Cloud Provider Working Group
https://github.com/kubernetes/community/tree/master/wg-cloud-provider

External-DNS
https://github.com/kubernetes-incubator/external-dns

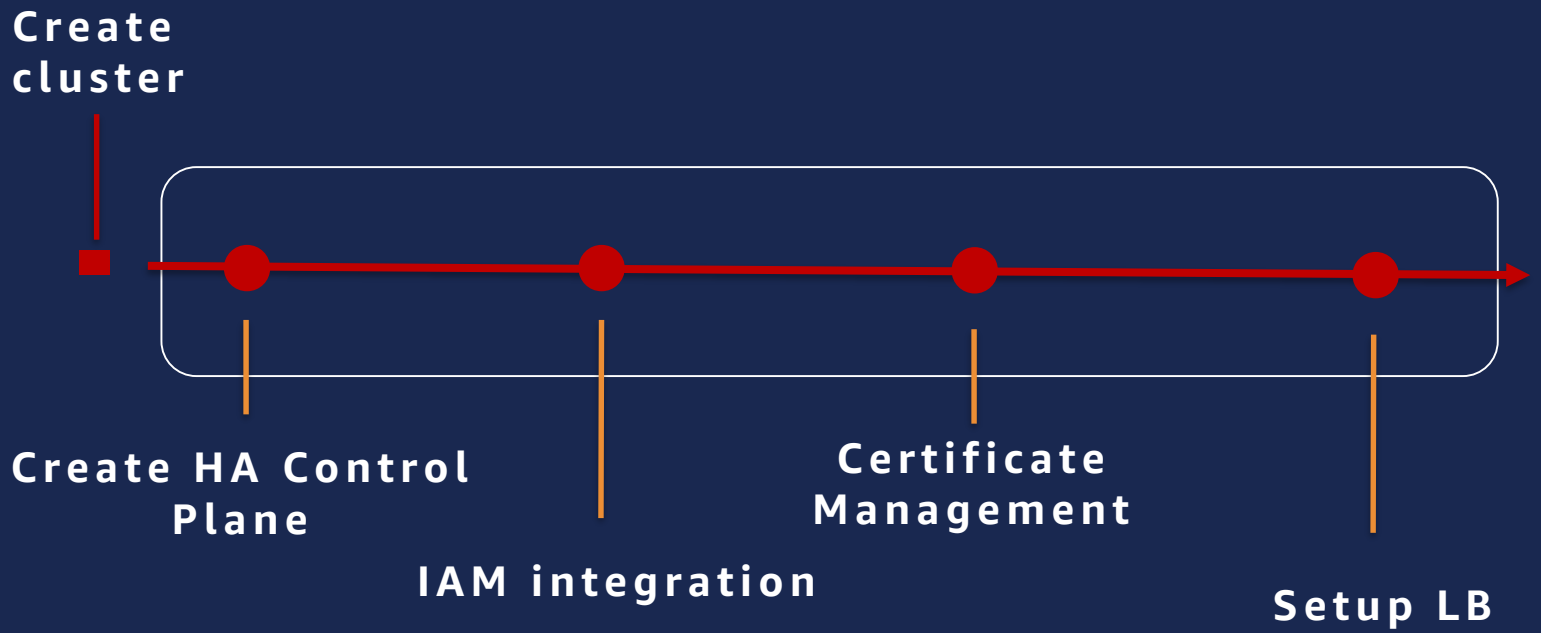
CoreOS ALB Ingress
https://github.com/coreos/alb-ingress-controller

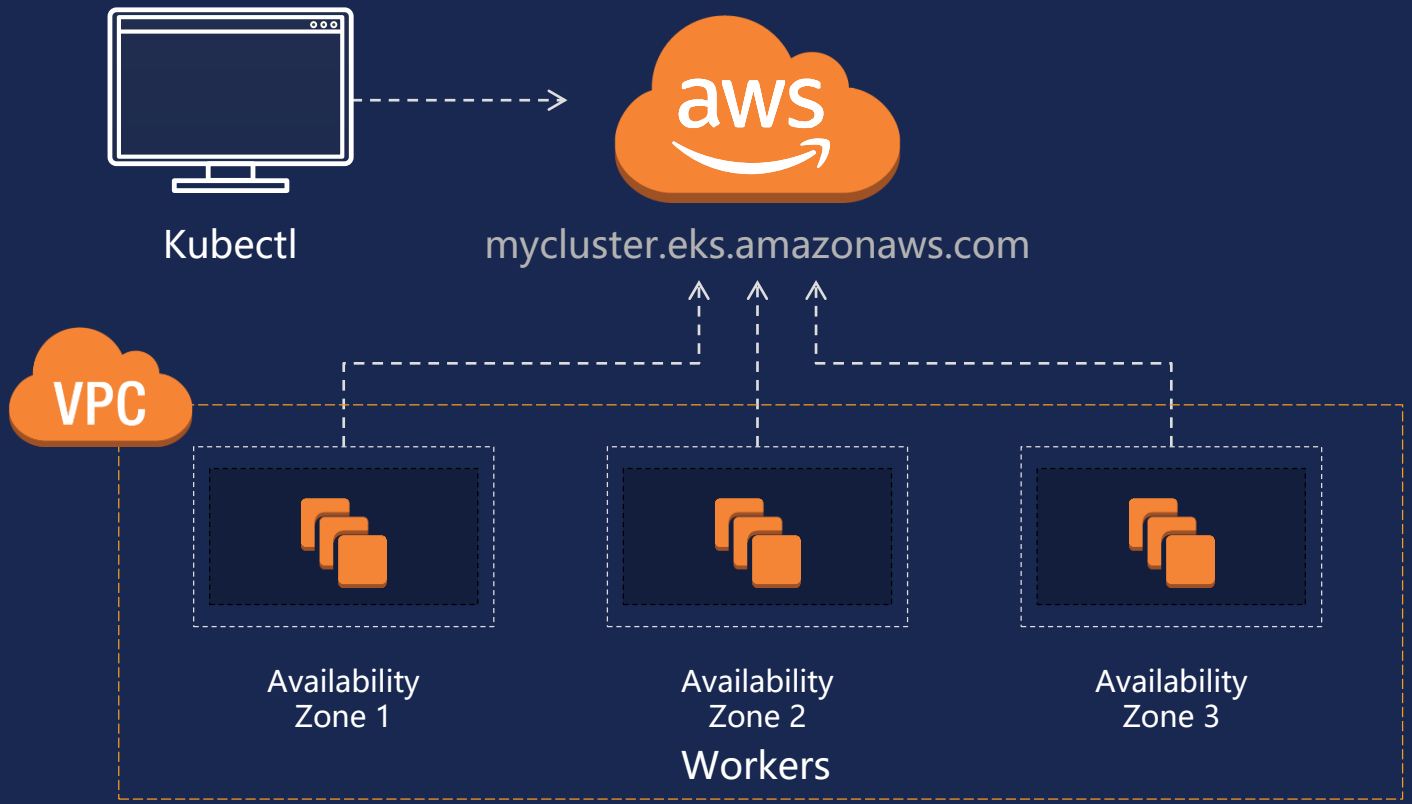
SIG AWS
@CHRISTOPH_K@TIFFANYFAYJ

EKS - Customers



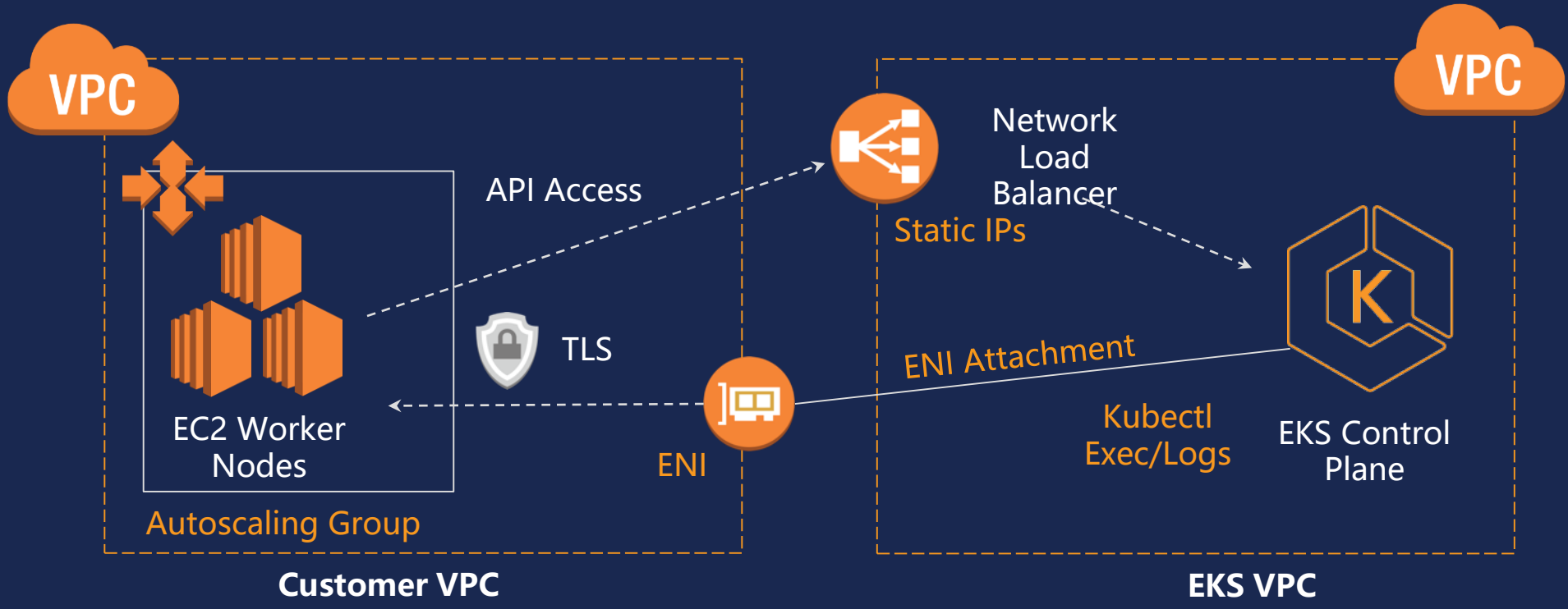
EKS - Kubernetes Control Plane





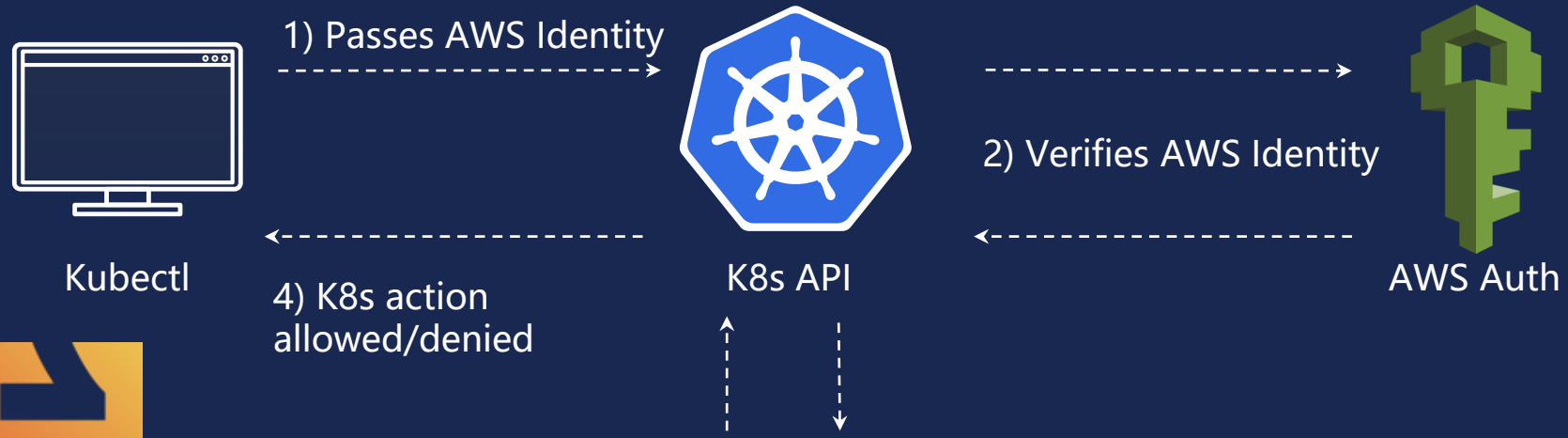
EKS Architecture

EKS Architecture



IAM Authentication

IAM Authentication + kubectl

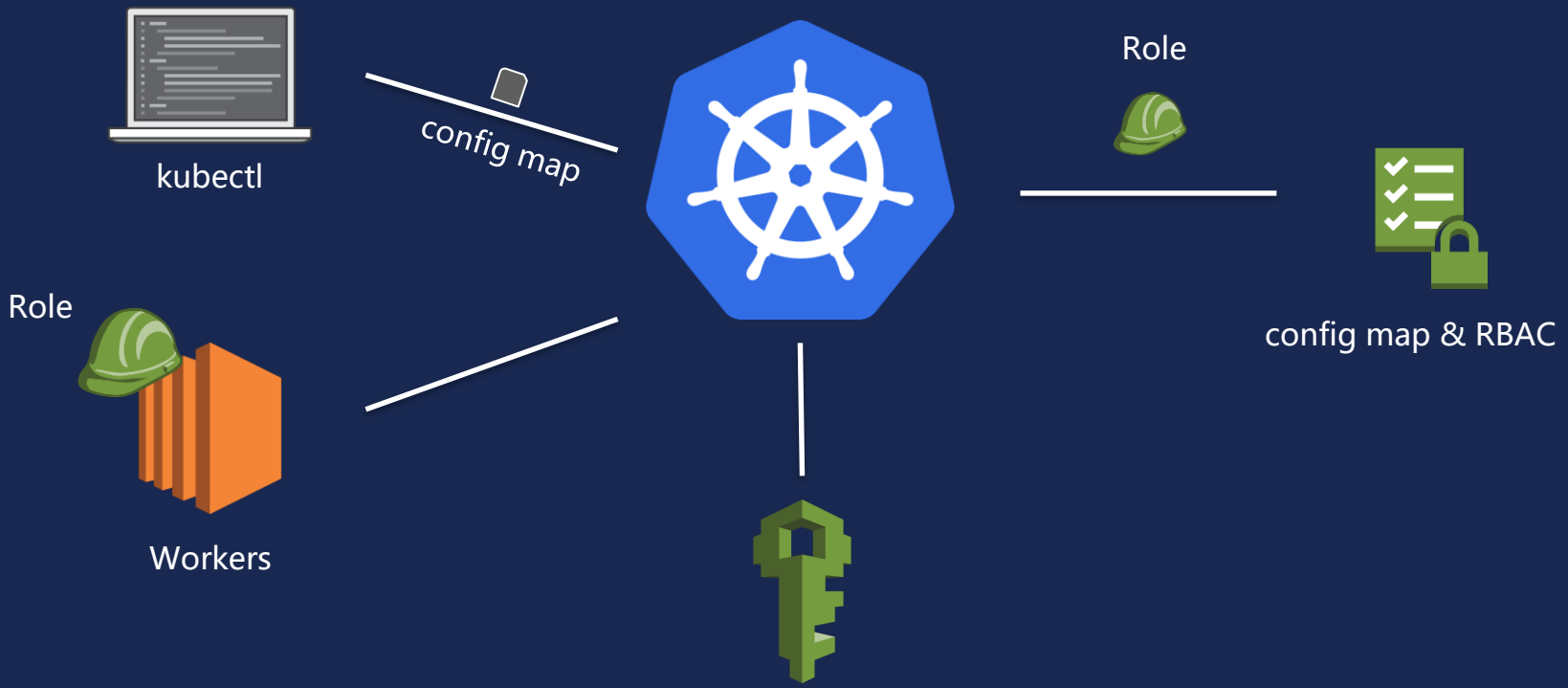


heptio

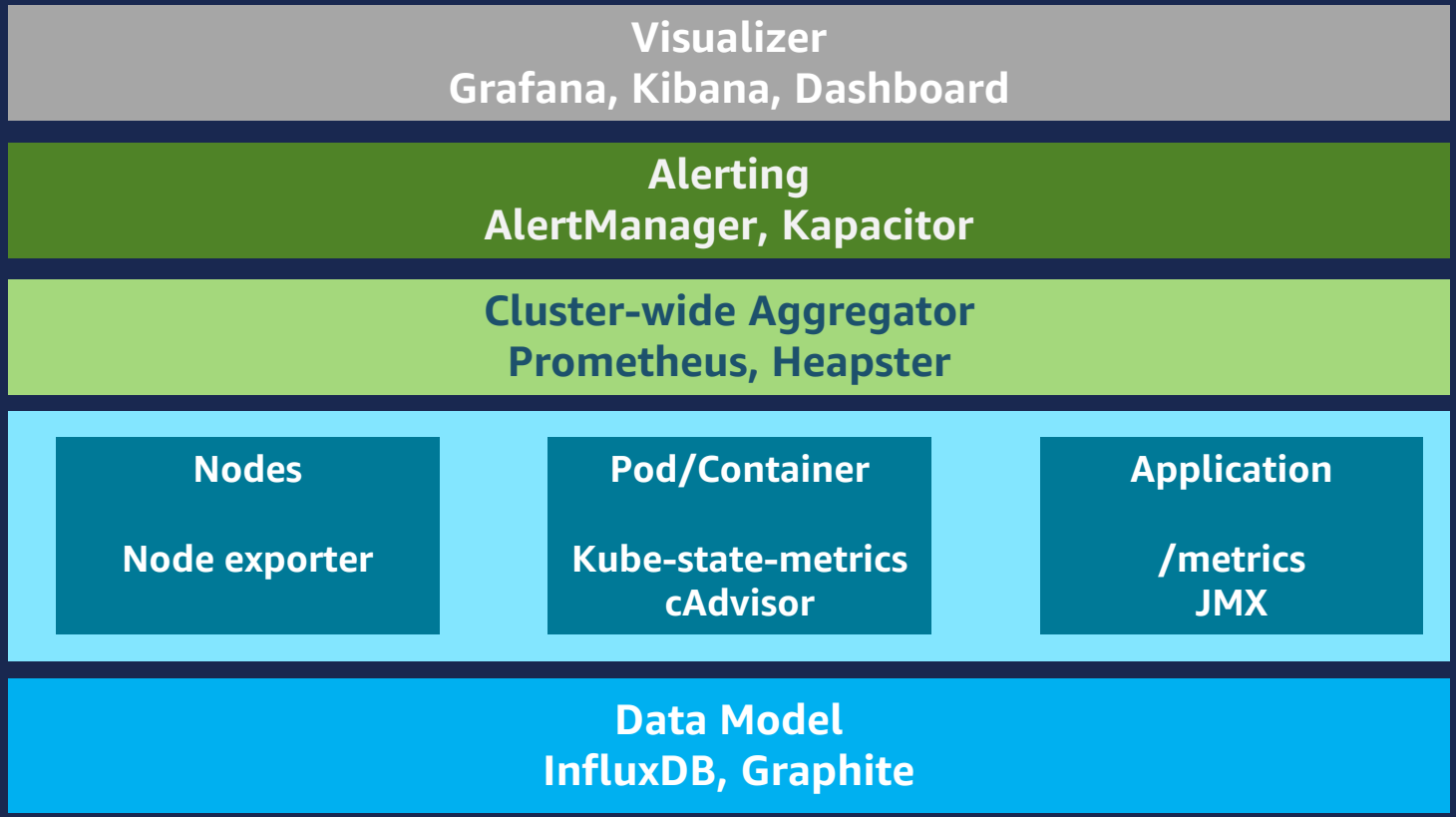
https://github.com/heptiolabs/kubernetes-aws-authenticator

EKS Worker Nodes

Worker provisioning



Metrics



Networking

CNI

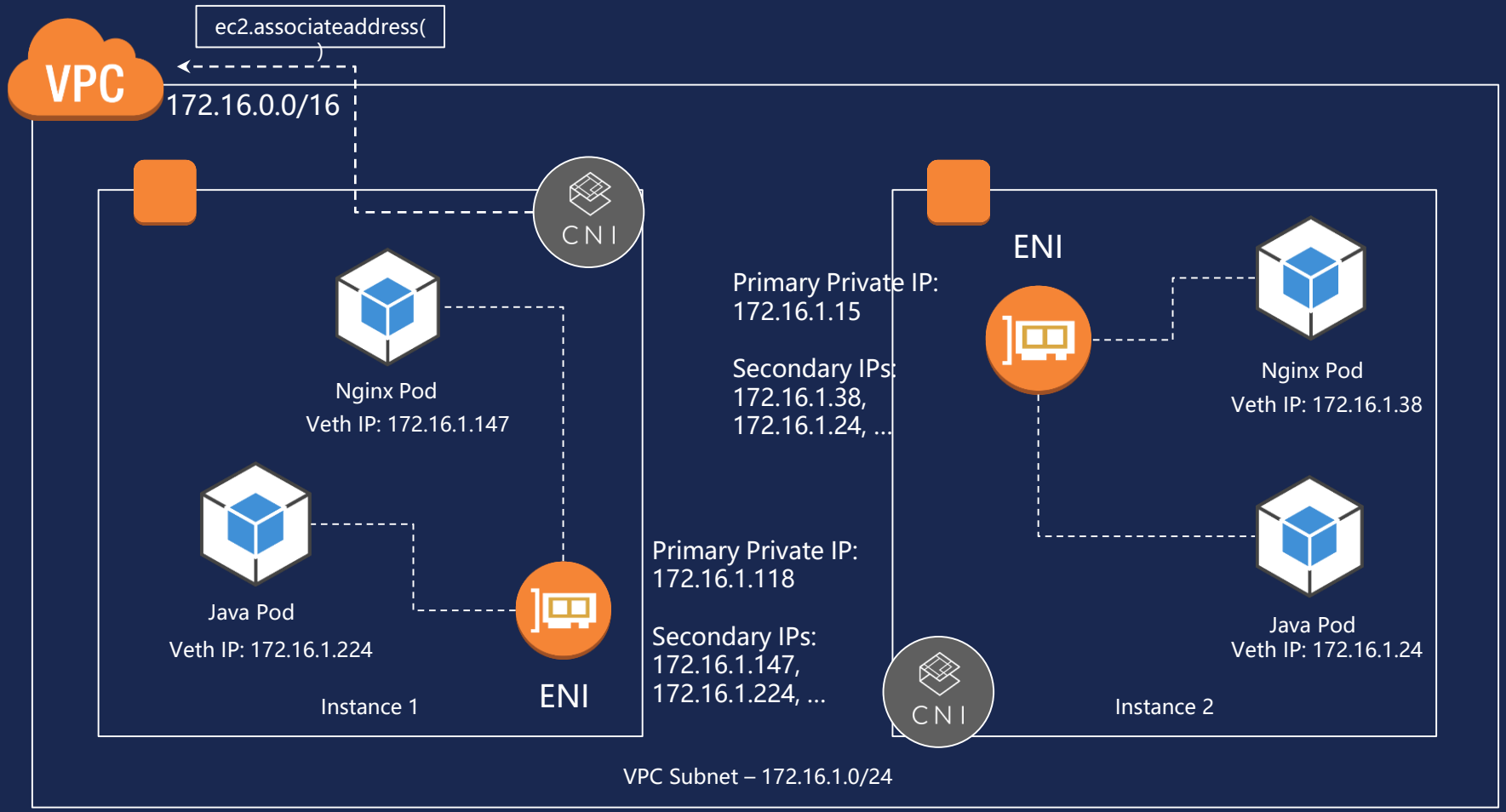
https://github.com/aws/amazon-vpc-cni-k8s

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



How do I configure network security with EKS?

TIGERA

https://www.projectcalico.org/

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

TIGERA

PROJECT CALICO

STAGE SEPARATION

Isolate dev, test, and prod

"TENANT" SEPARATION

Namespaces – without network policy, they are not network isolated

FINE-GRAINED FIREWALLS

Reduce attack surface within microservice-based applications

COMPLIANCE

E.g., PCI, HIPAA

What version of Kubernetes does EKS support?

1.10.3 currently

kubernetes

aws SUMMIT

Kubernetes Autoscaling with Amazon EKS

@CHRISTOPH_K@TIFFANYFAYJ

Auto Scaling

Cluster

Two options
- AWS AutoScaling
- k8s Cluster Auto Scaler

Cluster Autoscaler
- Reactive
- Aware of Pod / Cluster state
- Utilizes AWS AutoScaling

AWS AutoScaling
- Scaling on CloudWatch Metrics

Pods

Horizontal Pod Autoscaler
- Scales pods in response to k8s generated metrics (CPU)

https://github.com/kubernetes/helm

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

› helm search mysql

```
> helm search mysql
NAME                                    CHART VERSION    APP VERSION    DESCRIPTION
stable/mysql                            0.6.0            5.7.14         Fast, reliable, sc
stable/prometheus-mysql-exporter        0.1.0            v0.10.0        A Helm chart for p
stable/percona                          0.3.2            5.7.17         free, fully compat
...
```

› helm install install stable/mysql

[displays README + information about deployment]

› helm list

```
NAME            REVISION        UPDATED                         STATUS          CHART           NAMESPACE
nobby-cow       1               Wed Jun  6 12:54:00 2018        DEPLOYED        mysql-0.6.0     default
```

Hosting Helm repositories

- Anywhere that serves HTTP can host a helm repo
- Host private Helm Repo with Chartmuseum https://github.com/kubernetes-helm/chartmuseum

- There's also a handy plugin for S3!
- This means IAM Role = auth for your repo ☺

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

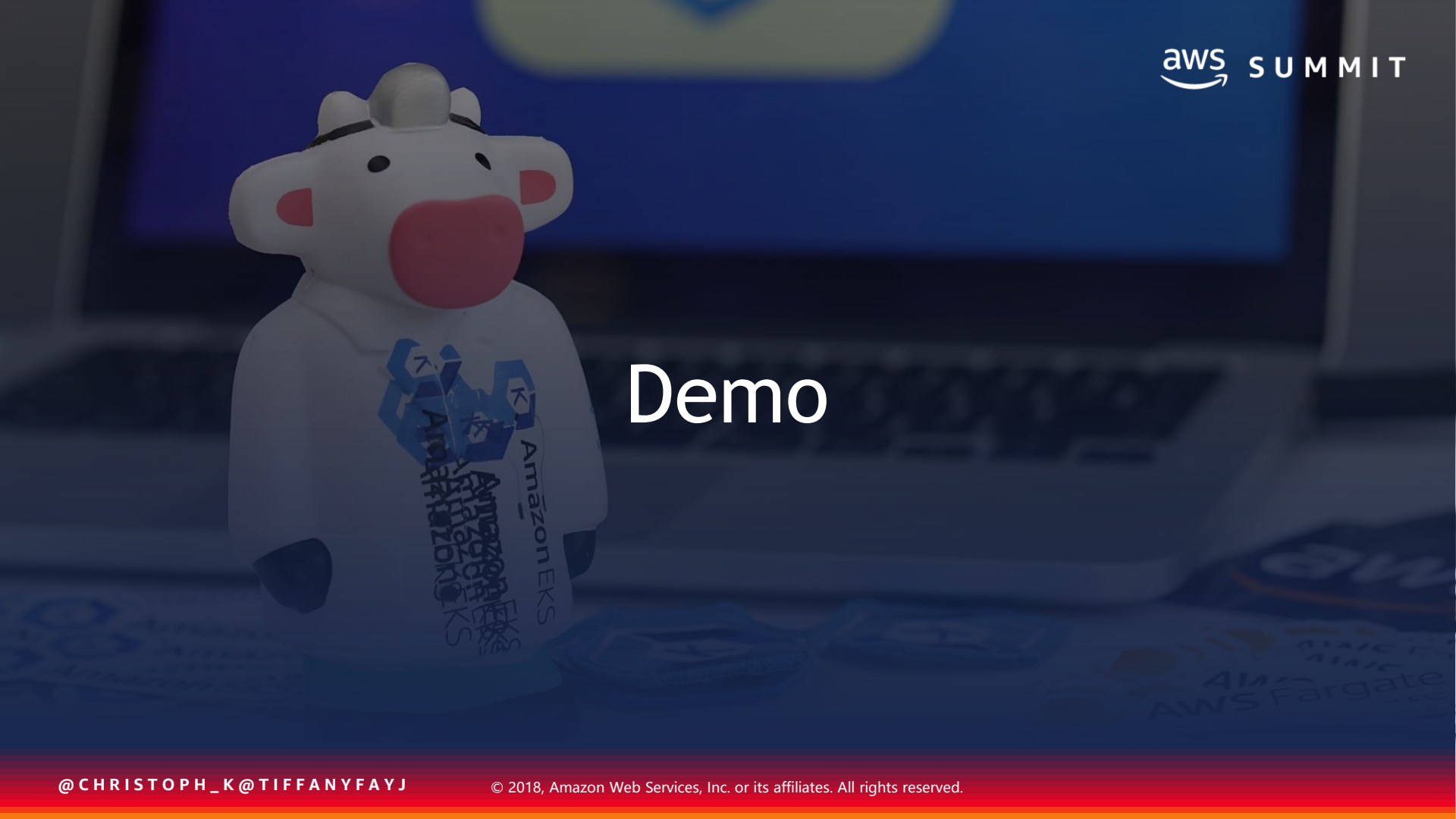
Deploying Helm on EKS

Helm 2.9+ works with EKS
RBAC permissions required

```
kubectl -n kube-system create serviceaccount tiller

kubectl create clusterrolebinding tiller --clusterrole cluster-admin --
serviceaccount=kube-system:tiller

helm init --service-account tiller
```

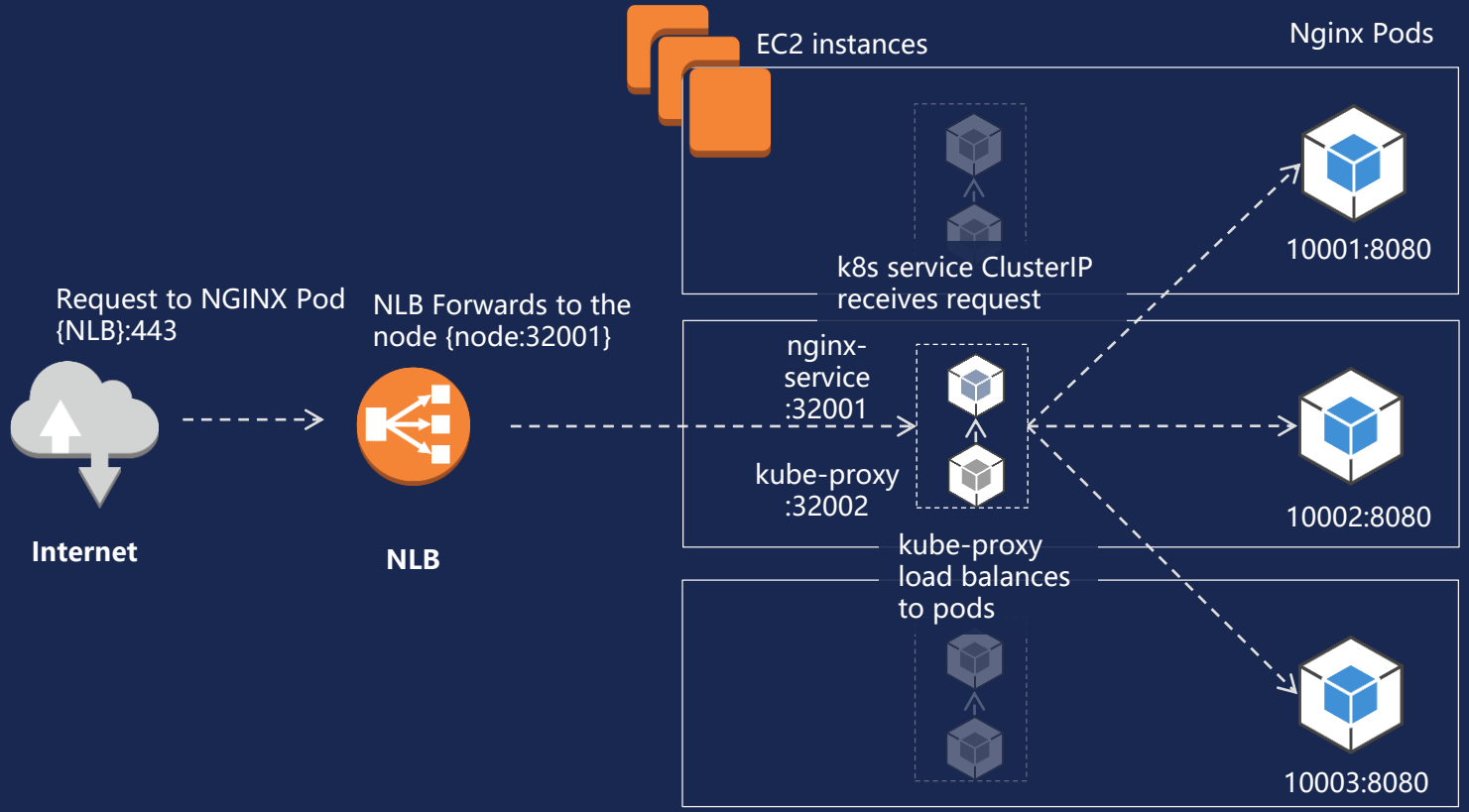


Know-how & Tools

aws SUMMIT

Load Balancing

Service Type - LoadBalancer (NLB)



Network Load Balancer

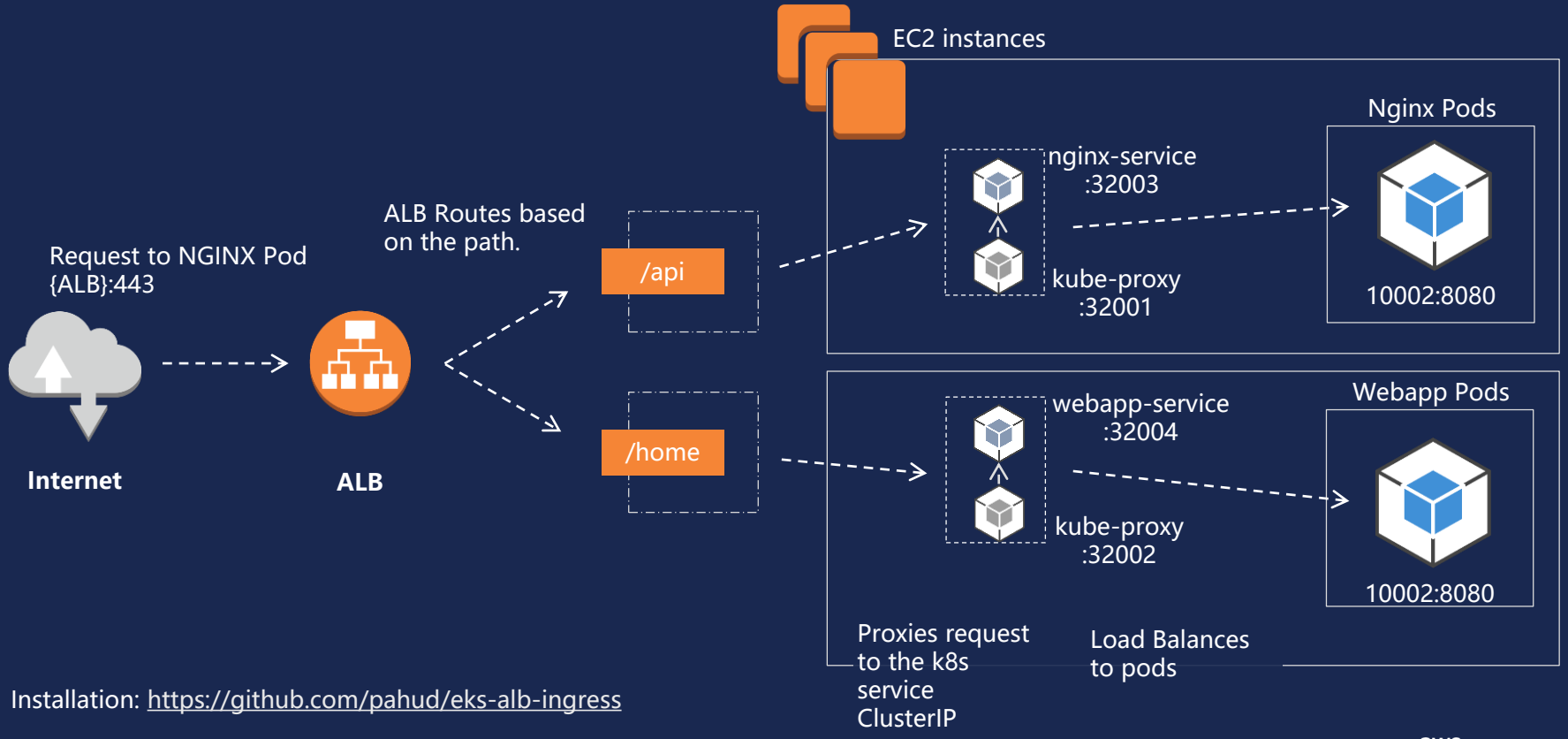
```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  namespace: default
  labels:
    app: nginx
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-type: "nlb"
spec:
  type: LoadBalancer
  externalTrafficPolicy: Local
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: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

More options:

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

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

Ingress Type - CoreOS ALB Ingress



Installation: https://github.com/pahud/eks-alb-ingress

aws SUMMIT

DNS

Automatic Route53 DNS creation for services

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    # Uses https://github.com/kubernetes-incubator/external-dns
    external-dns.alpha.kubernetes.io/hostname: nginx.highlyavailable.systems.
spec:
  type: LoadBalancer
  ports:
  - port: 80
    name: http
    targetPort: 80
  selector:
    app: nginx
```

Automatic Route53 DNS creation for Ingress

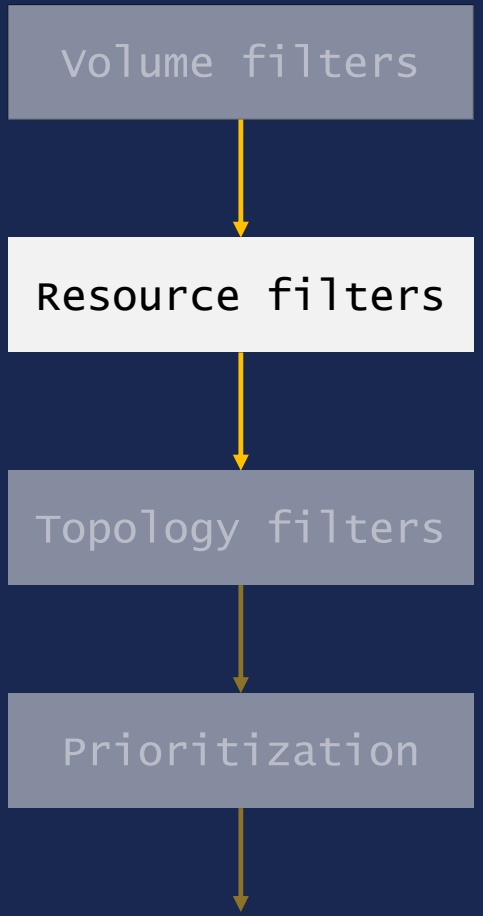
```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx
  annotations:
    kubernetes.io/ingress.class: "nginx"
spec:
  rules:
  - host: nginx.highlyavailable.systems
    http:
      paths:
      - backend:
          serviceName: nginx
          servicePort: 80
```

aws SUMMIT

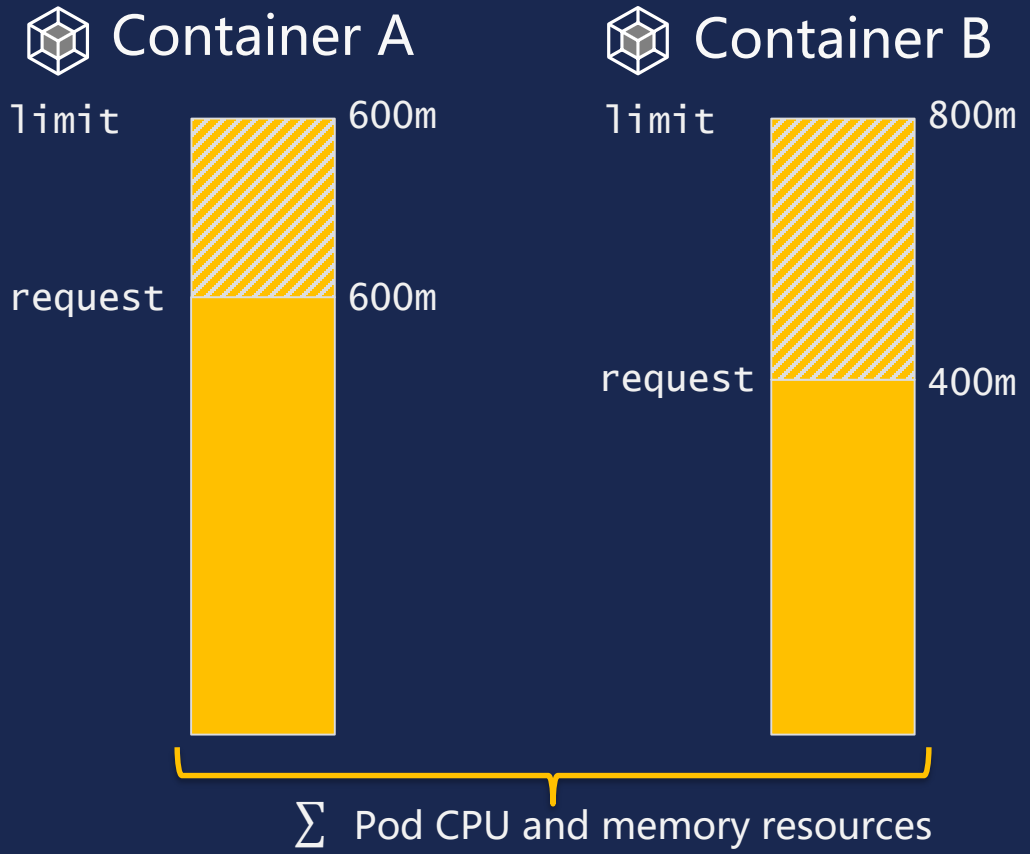
Scheduling

Controlling scheduling

Resource requirements



Limit resource usage



Resource Quotas

Applied per Namespace

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: production
spec:
  hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
```

Pod Resource Request

```
apiVersion: v1
kind: Pod
metadata:
  name: production
spec:
  containers:
  - name: nginx-pod
    image: nginx
    resources:
      limits:
        memory: "800Mi"
        cpu: "800m" # 0.8 vCPU
      requests:
        memory: "600Mi"
        cpu: "400m“ # 0.4 vCPU
```

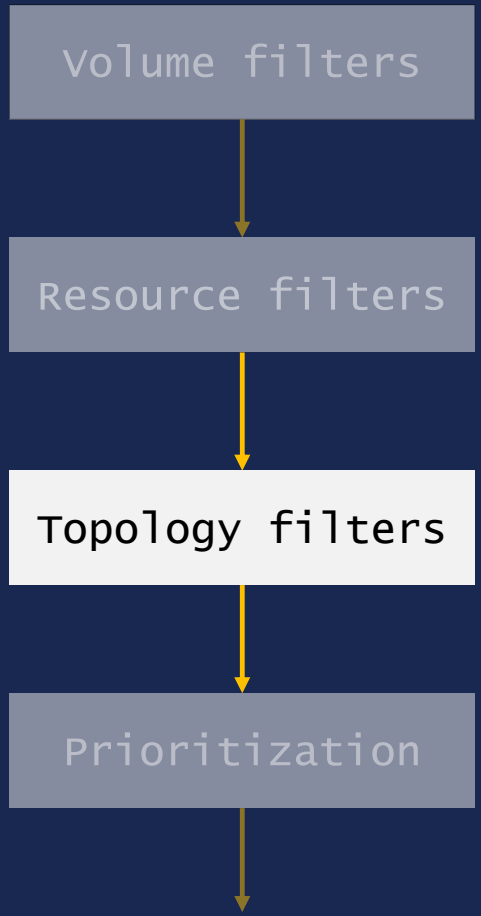
ResourceQuota defined both, so Pod must define both

Controlling scheduling

Resource requirements

Constraints

- Taints — Node-level
- Tolerations — Pod-level



Taints and Tolerations

```
# Taint node
$ kubectl taint nodes ip-10-0-32-12.us-west-2.compute.internal \
    skynet=false:NoSchedule

# Tolerations
kind: Pod
spec:
  tolerations:
  - key: skynet
    operator: Equal
    value: “false”
    effect: NoSchedule

[...]
```

Match taint to schedule onto tainted node

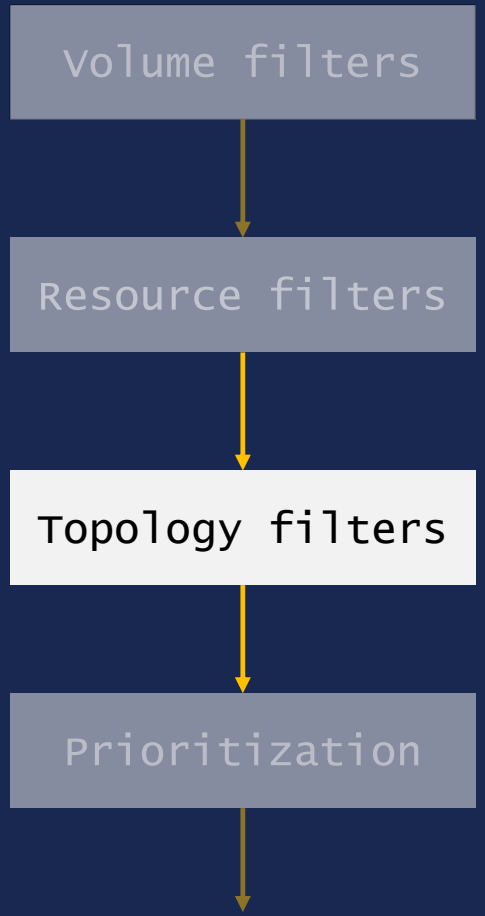
Controlling scheduling

Resource requirements

Constraints

- Taints — Node-level
- Tolerations — Pod-level

Affinity/Anti-Affinity



Affinity / Anti-Affinity

- Control scheduling onto nodes
  - Combine with Taints & Tolerations
- Distribute Pods across cluster

```
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
        - matchExpressions:
          - key: "beta.kubernetes.io/instance-type"
            operator: In
            values: [“r4.large",“r4.xlarge"]
```

aws SUMMIT

Deployment Strategies

Rolling Update

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  replicas: 10
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1           # Numeric or percentage based value
      maxUnavailable: 0
[...]
```

Blue / Green Deployment

Blue

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: my-app-blue
  labels:
    app: my-app
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: my-app
        version: blue
  [...]
```

Green

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: my-app-green
  labels:
    app: my-app
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: my-app
        version: green
  [...]
```

Blue / Green Deployment

```
kubectl patch service my-app -p '{"spec":{"selector":{"version":"green"}}}'
```

Blue

```
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  type: LoadBalancer
  ports:
  - name: http
    port: 80
    targetPort: http
  selector:
    app: my-app
    version: blue
```

Green

```
kind: Service
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  type: NodePort
  ports:
  - name: http
    port: 80
    targetPort: http
  selector:
    app: my-app
    version: green
```

Canary Deployment

Production

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: my-app-prod
  labels:
    app: my-app
spec:
  replicas: 9
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-app
        image: images/container:v1
    [...]
```

Canary

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: my-app-canary
  labels:
    app: my-app
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-app
        image: images/container:v2
    [...]
```

More examples at https://container-solutions.com/kubernetes-deployment-strategies/

aws SUMMIT

Network Policies

Network Policy

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: web-allow-prod
spec:
  podSelector:
    matchLabels:
      app: web
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          purpose: production
```

Select affected Pods

Define traffic that is allowed

aws SUMMIT

Want to learn more?

Tooling and Ecosystem

https://github.com/ramitsurana/awesome-kubernetes

https://discuss.kubernetes.io/

http://slack.k8s.io/

TGIK Playlist:
https://www.youtube.com/playlist?list=PLvmPtYZtoXOENHJiAQc6HmV2jmuexKfrJ

EKS - Getting started

https://aws.amazon.com/eks

https://aws.amazon.com/getting-started/projects/deploy-kubernetes-app-amazon-eks/

https://aws.amazon.com/blogs/aws/amazon-eks-now-generally-available/

https://aws.amazon.com/blogs/compute/
https://aws.amazon.com/blogs/opensource/category/compute/amazon-elastic-container-service-for-kubernetes/

https://medium.com/containers-on-aws

Questions?

Please complete the session survey in the summit mobile app.

aws SUMMIT

Thank You

@christoph_k
@tiffanyfayj

Special thanks to:
Paul Maddox, Abby Fuller, Nishi Davidson, Brandon Chavis, Arun Gupta, Chris Hein, Omar Lari, and many more...

https://aws.amazon.com/containers