Build with Containers

Seoul, South Korea

Your presenters

Omar Lari,
Container Service

- Held engineering and leaderships roles at various silicon valley startups, enterprise organization and consulting firms
- 3 years at AWS – Solutions Architect specializing in container adoption and container services business development

aws

Excited to be in South Korea!

- My first time here
- Suggestions for food, sights and soju drinking tips are welcome ☺

aws

Today's Agenda

09:30 **Welcome**
09:35 Building with Kubernetes on AWS
10:20 Amazon EKS Hands-On Lab
12:45 **Lunch**
13:15 Partner Presentation
13:45 Building with Fargate and Amazon ECS
14:30 Fargate Hands-On Lab
16:30 **Closing Discussion**
17:00 End

aws

Important information

- Bathrooms
- Slides will be provided after the event via slideshare
- You will receive a survey after the event in your email. It only takes **3 minutes** to complete! Let us know how we did and what to improve.

aws

Containers on AWS

AWS Container Services Landscape

Application Management

Amazon Elastic Container Registry

Deployment, Scheduling, Scaling & Management of containerized applications

Amazon Elastic Container Service

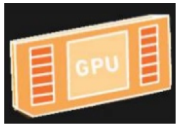
Amazon Elastic Container Service for Kubernetes

Where the containers run

Amazon EC2

AWS Fargate

Where the containers run



aws

Kubernetes on AWS

Open source container management

Omar Lari, Container Services

March 4th, 2019

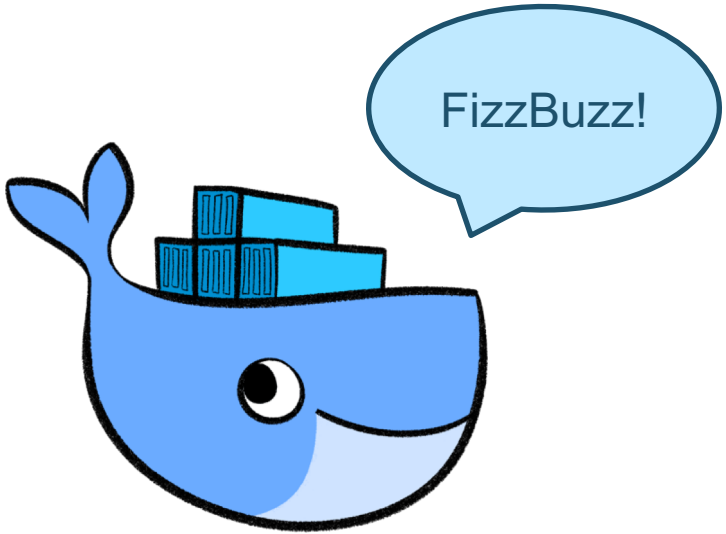
aws

Agenda

- Why containers?
- What is Kubernetes?
- Key concepts
- Running Kubernetes on AWS
- Demonstration

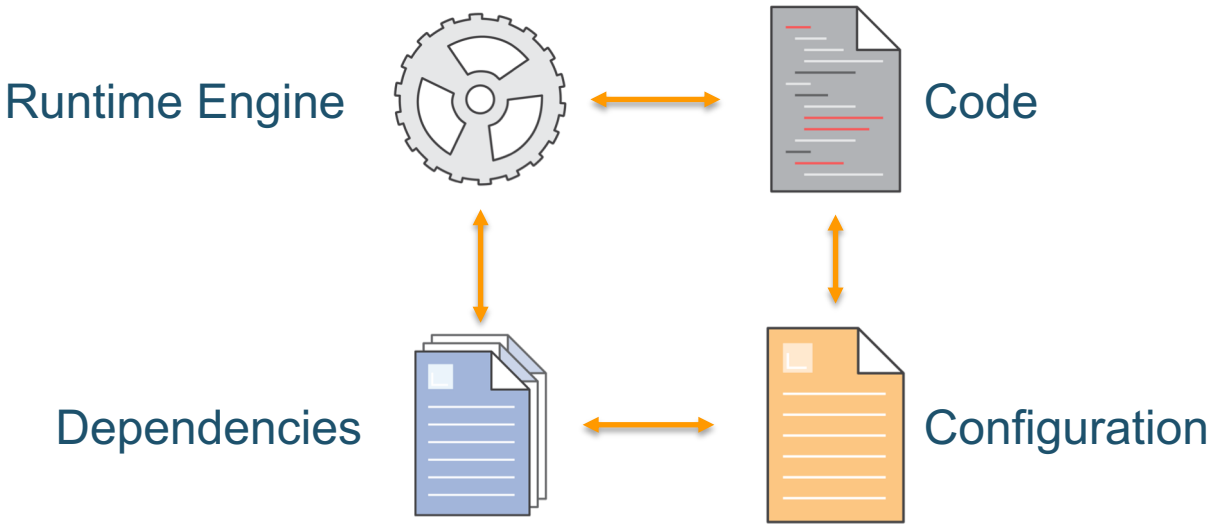
aws

Why containers?



aws

Application environment components



aws

Different environments

Local Laptop

Staging / QA

Production

On-Premises

aws

It worked on my machine, why not in prod?

v6.0.0

Local Laptop

v7.0.0

Staging / QA

v4.0.0

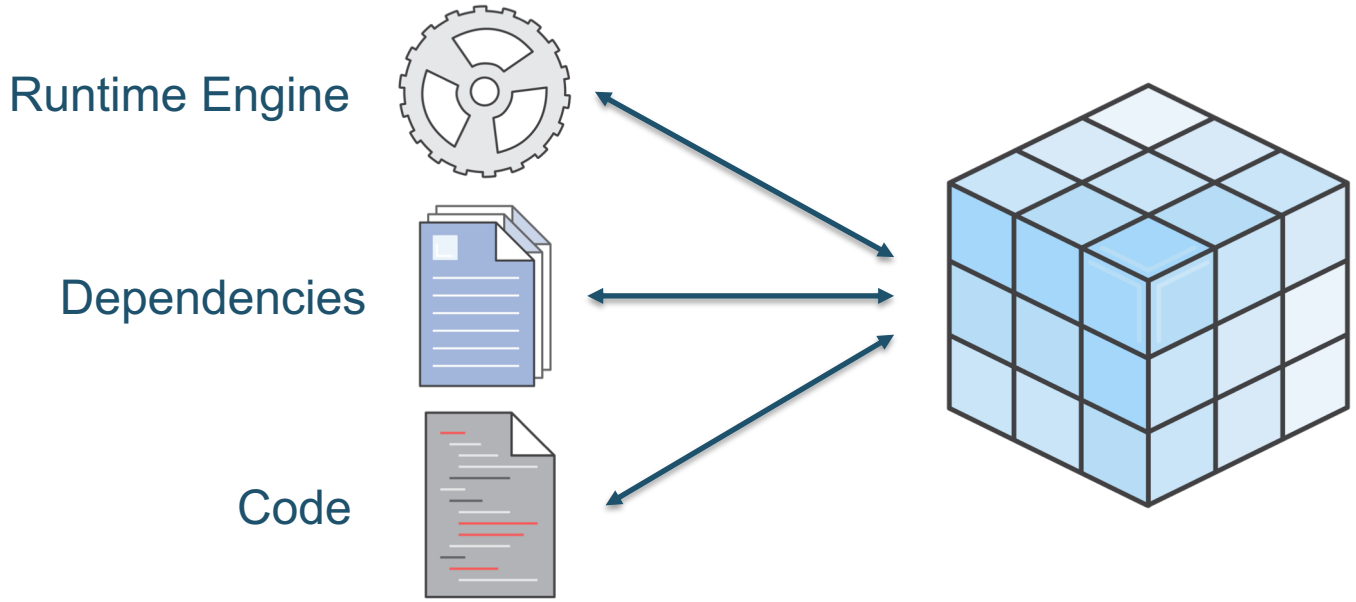
Production

v7.0.0

On-Prem

aws

Containers to the rescue



aws

Docker

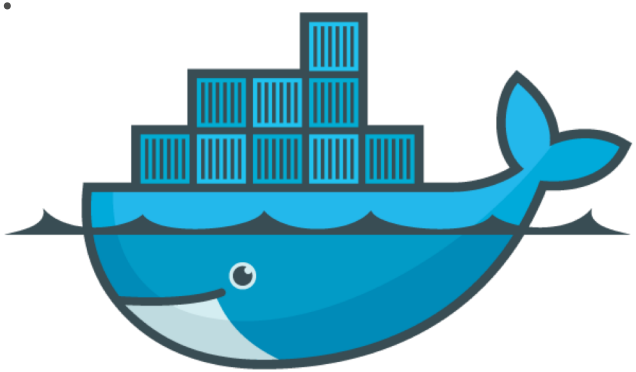
Lightweight container virtualization platform.

Tools to manage and deploy your applications.

Licensed under the Apache 2.0 license.

First released March 2013

Built by Docker, Inc.



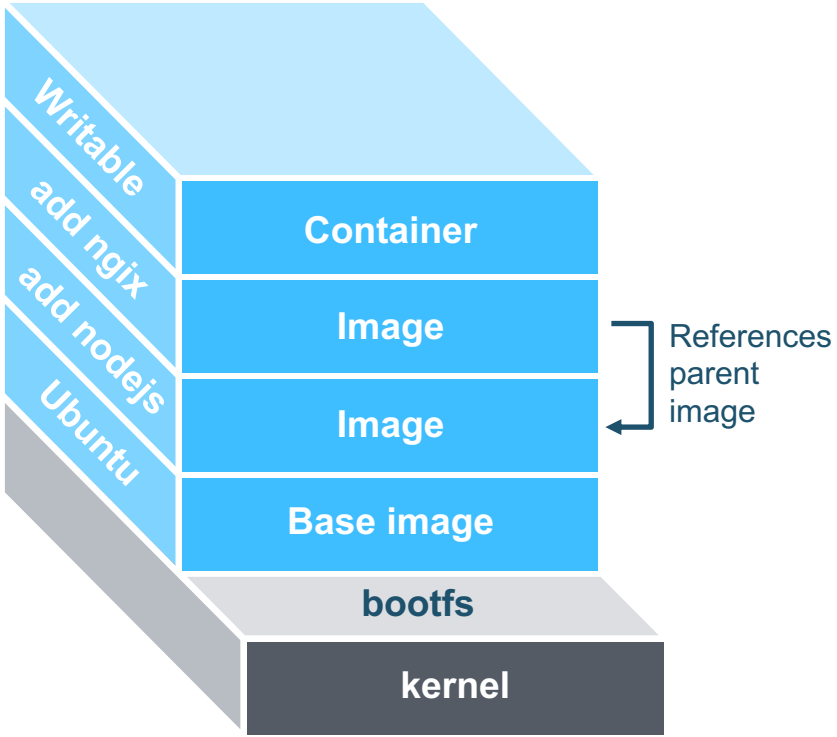
aws

Docker Image

Read only image that is used as a template to launch a container.

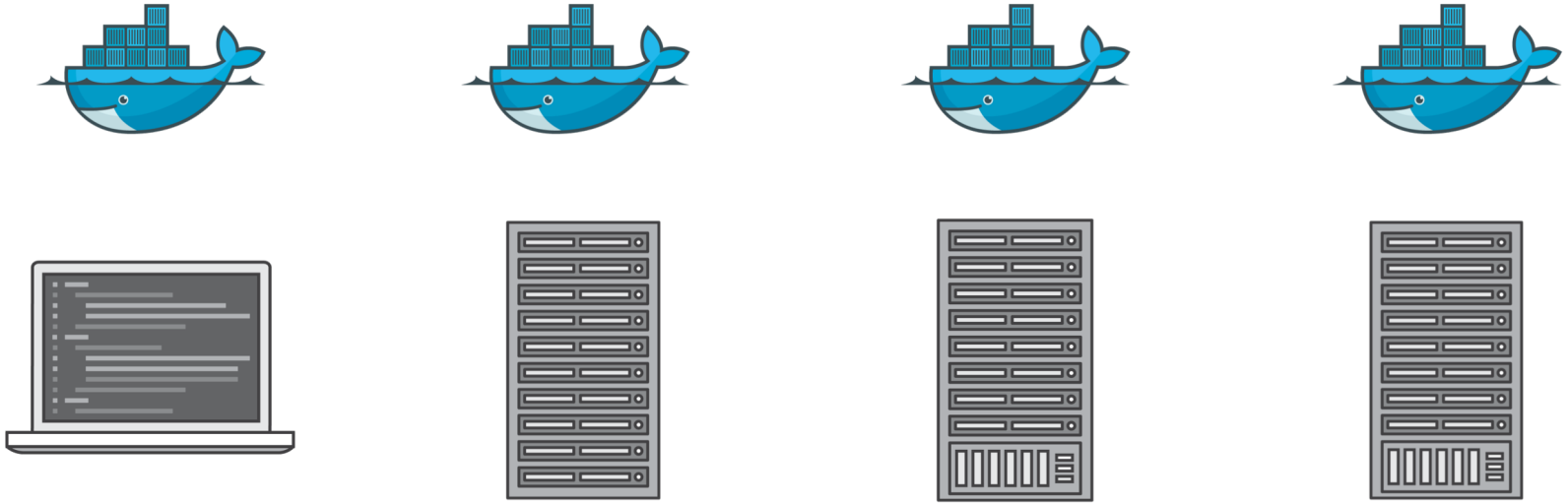
Start from base images that have your dependencies, add your custom code.

Docker file for easy, reproducible builds.



aws

Four environments, same container



Local Laptop

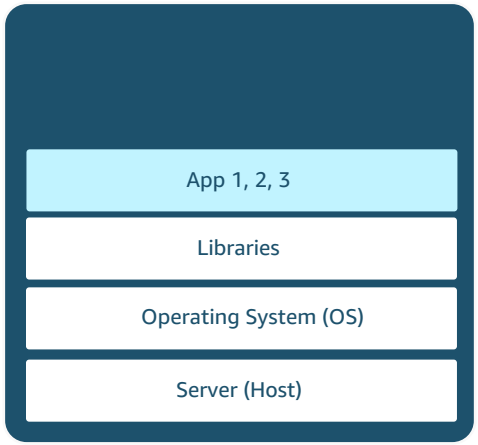
Staging / QA

Production

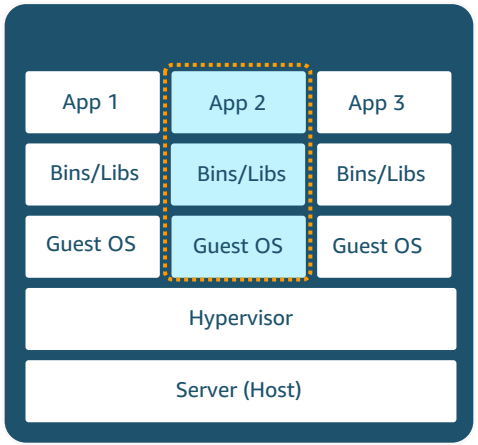
On-Prem

aws

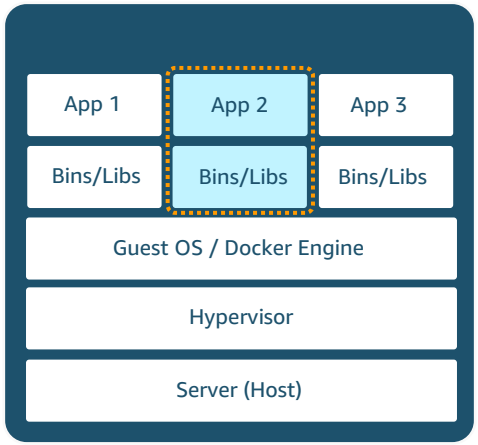
Container Implementation



Bare Metal



Virtual Machine



Containers

aws

But how do we make this work at scale?

aws

We need to

- start, stop, and monitor lots of containers running on lots of hosts
- decide when and where to start or stop containers
- control our hosts and monitor their status
- manage rollouts of new code (containers) to our hosts
- manage how traffic flows to containers and how requests are routed

aws

kubernetes

Amazon ECS

aka ‘k8s’

aws

AWS Container Services Landscape

IMAGE REGISTRY
Container Image Repository

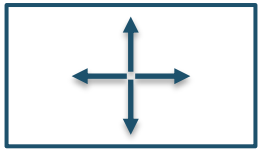
MANAGEMENT
Deployment, Scheduling, Scaling & Management

HOSTING
Where the containers run

aws

What is Kubernetes?

Open source container management platform



Helps you run containers at scale

Gives you primitives for building modern applications

aws

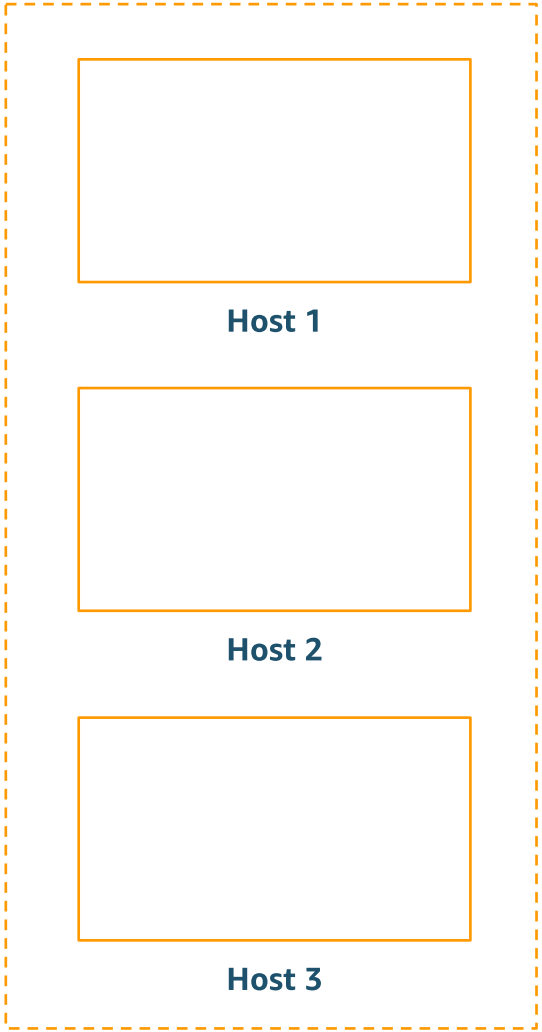
Containers on Hosts

A host is a server – e.g. EC2 virtual machine. We run these hosts together as a cluster.

Our simple example web application is already containerized.

Web App

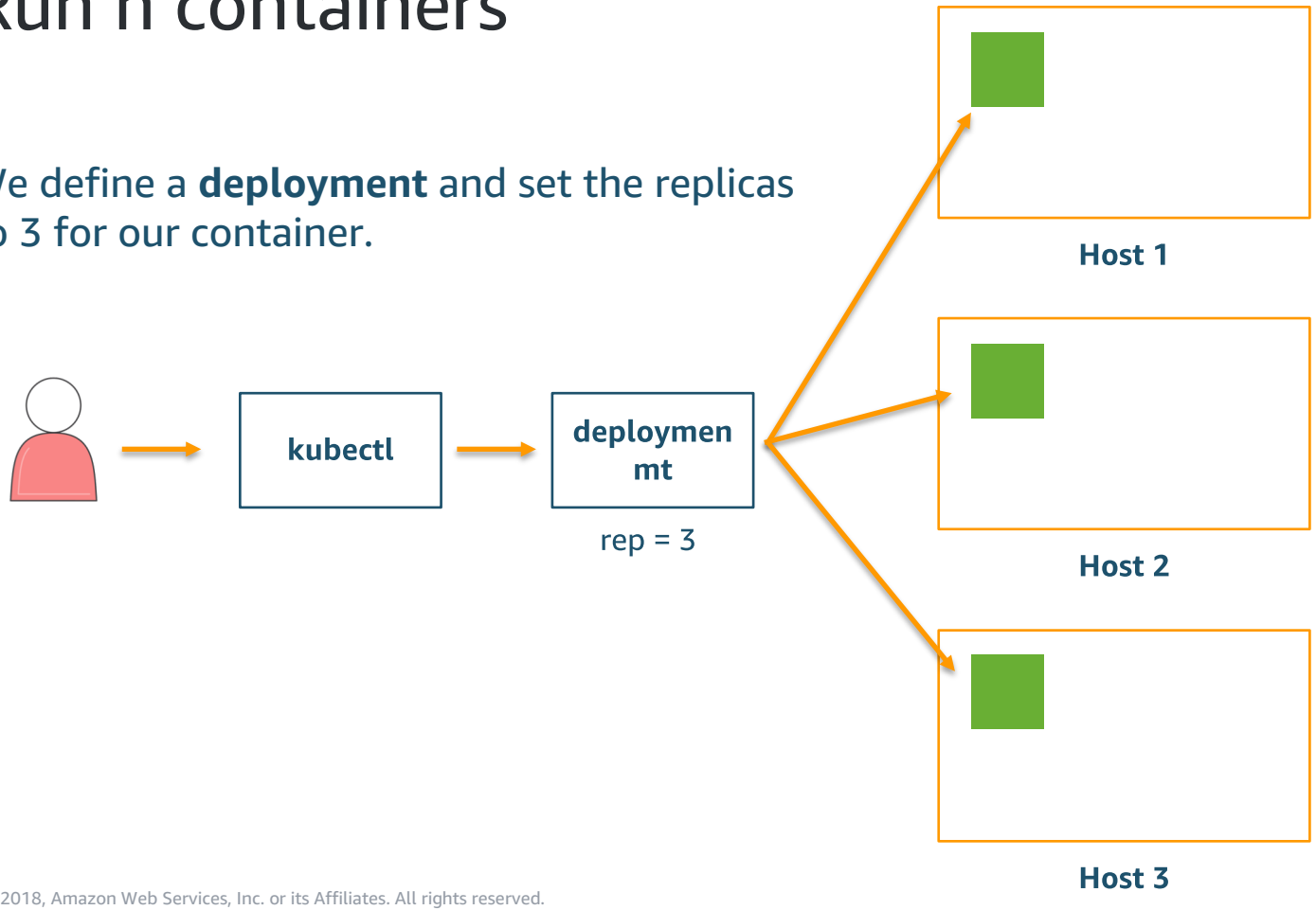
To start let's run a 3 copies of our web app across our cluster of EC2 hosts.



aws

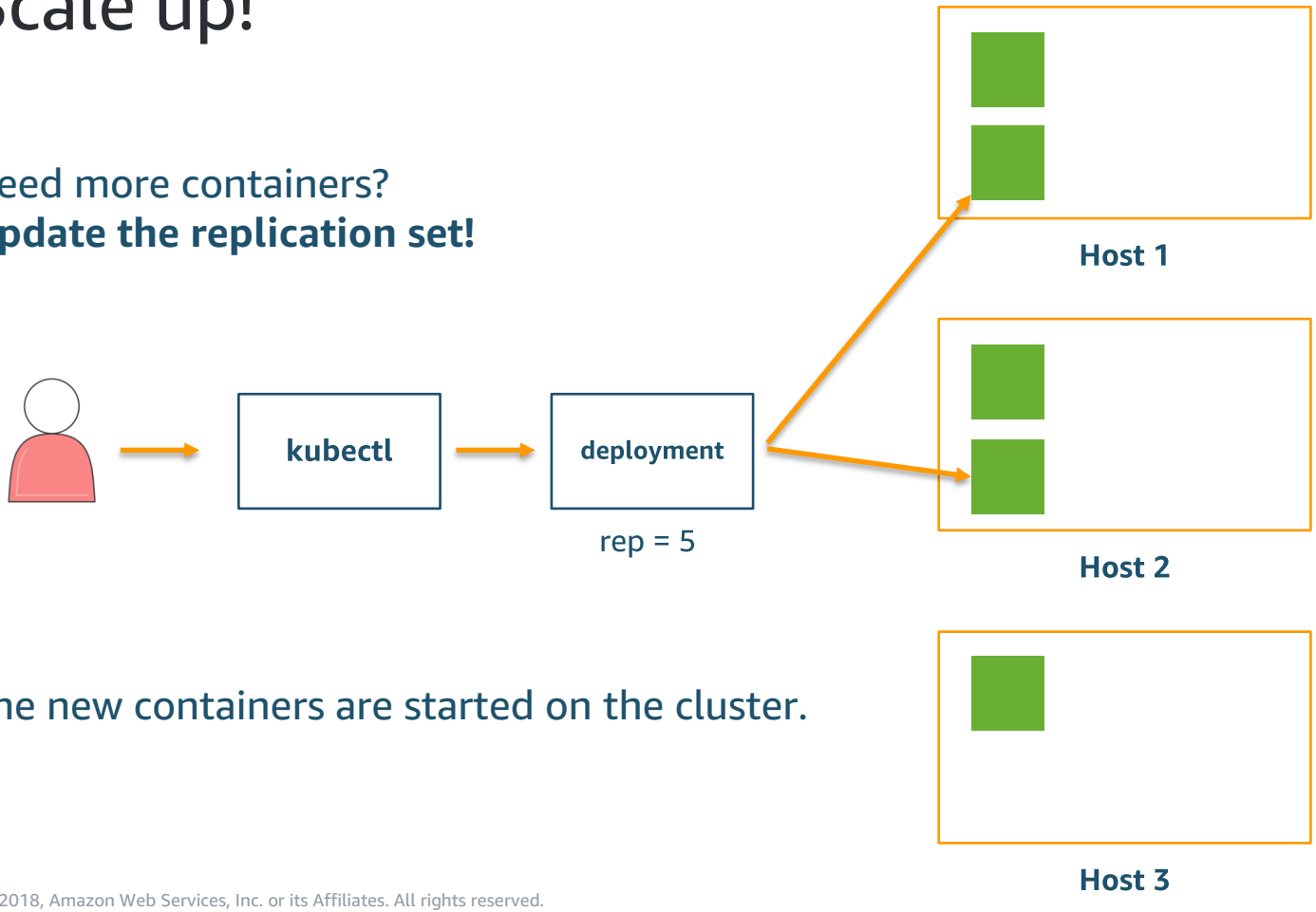
Run n containers

We define a **deployment** and set the replicas to 3 for our container.



Scale up!

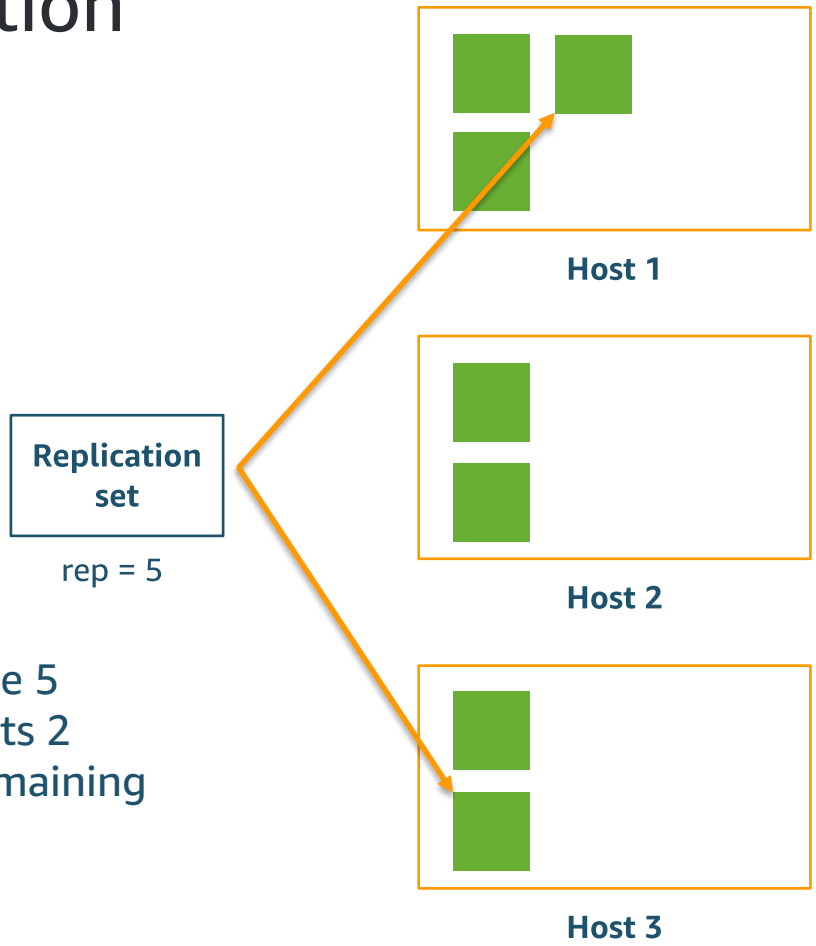
Need more containers?
Update the replication set!



The new containers are started on the cluster.

Untimely termination

Oh no! Our host has died!



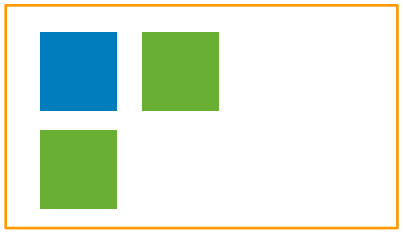
Kubernetes notices only 3 of the 5 containers are running and starts 2 additional containers on the remaining hosts.

aws

Containers IRL

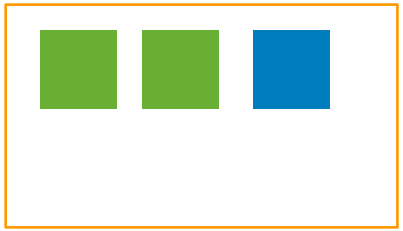
In production, we want to do more complex things like,

- Run a **service** to route traffic to a set of running containers
- Manage the **deployment** of containers to our cluster
- Run multiple containers **together** and specify how they run



Host 1

Host 2



Host 3

aws

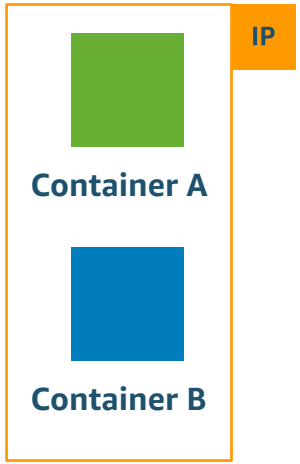
Pods

- Define how your containers should run
- Allow you to run 1 to n containers together

Containers in pods have

- Shared IP space
- Shared volumes
- Shared scaling (you scale pods not individual containers)

When containers are started on our cluster, they are always part of a pod.
(even if it's a pod of 1)

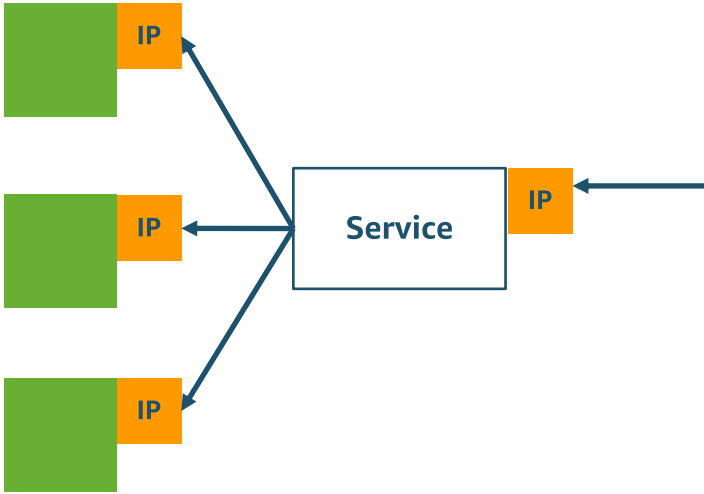


aws

Services

One of the ways traffic gets to your containers.

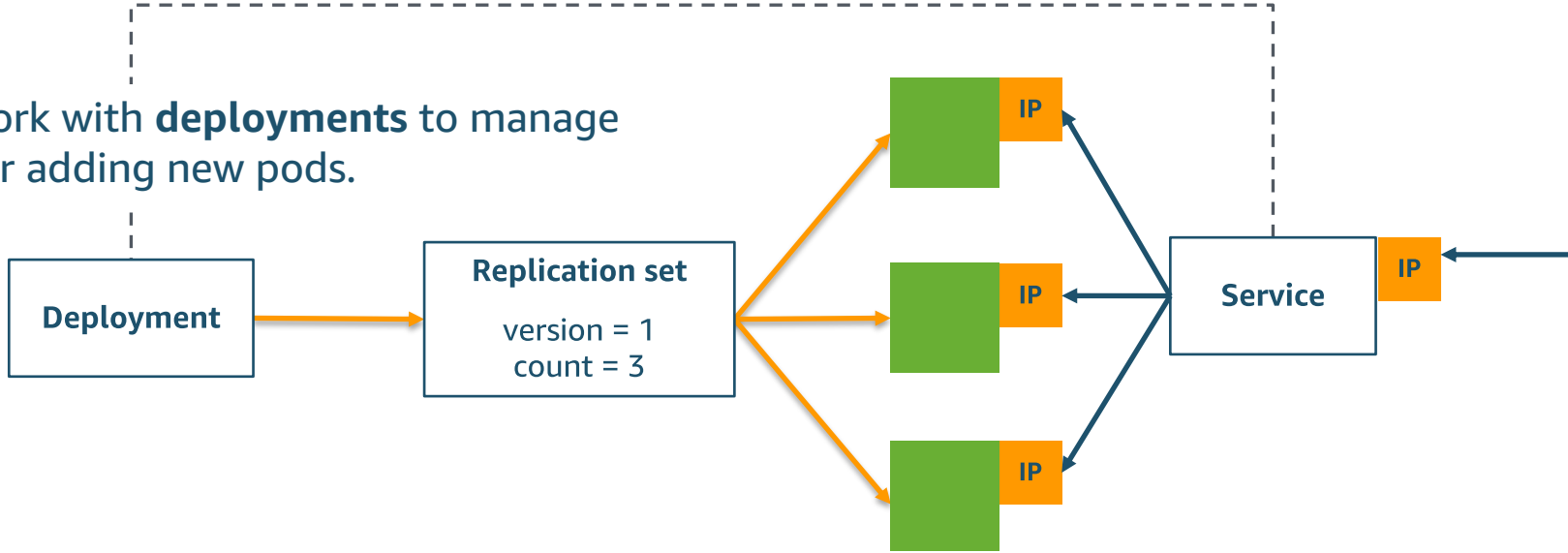
- Internal IP addresses are assigned to each container
- Services are connected to containers and use labels to reference which containers to route requests to



aws

Deployments

Services work with **deployments** to manage updating or adding new pods.

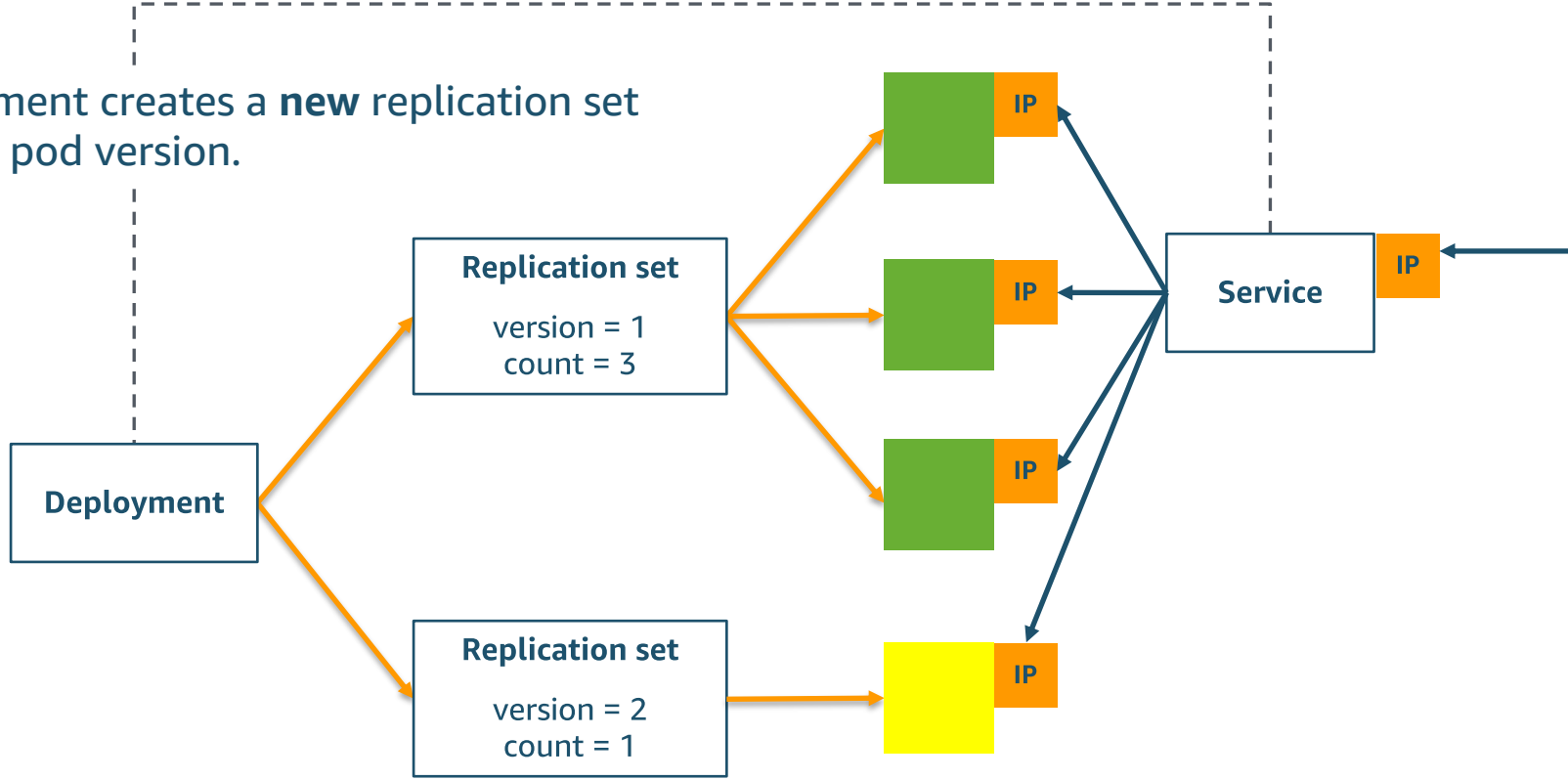


Let’s say we want to deploy a new version of our web app as a ‘canary’ and see how it handles traffic.

aws

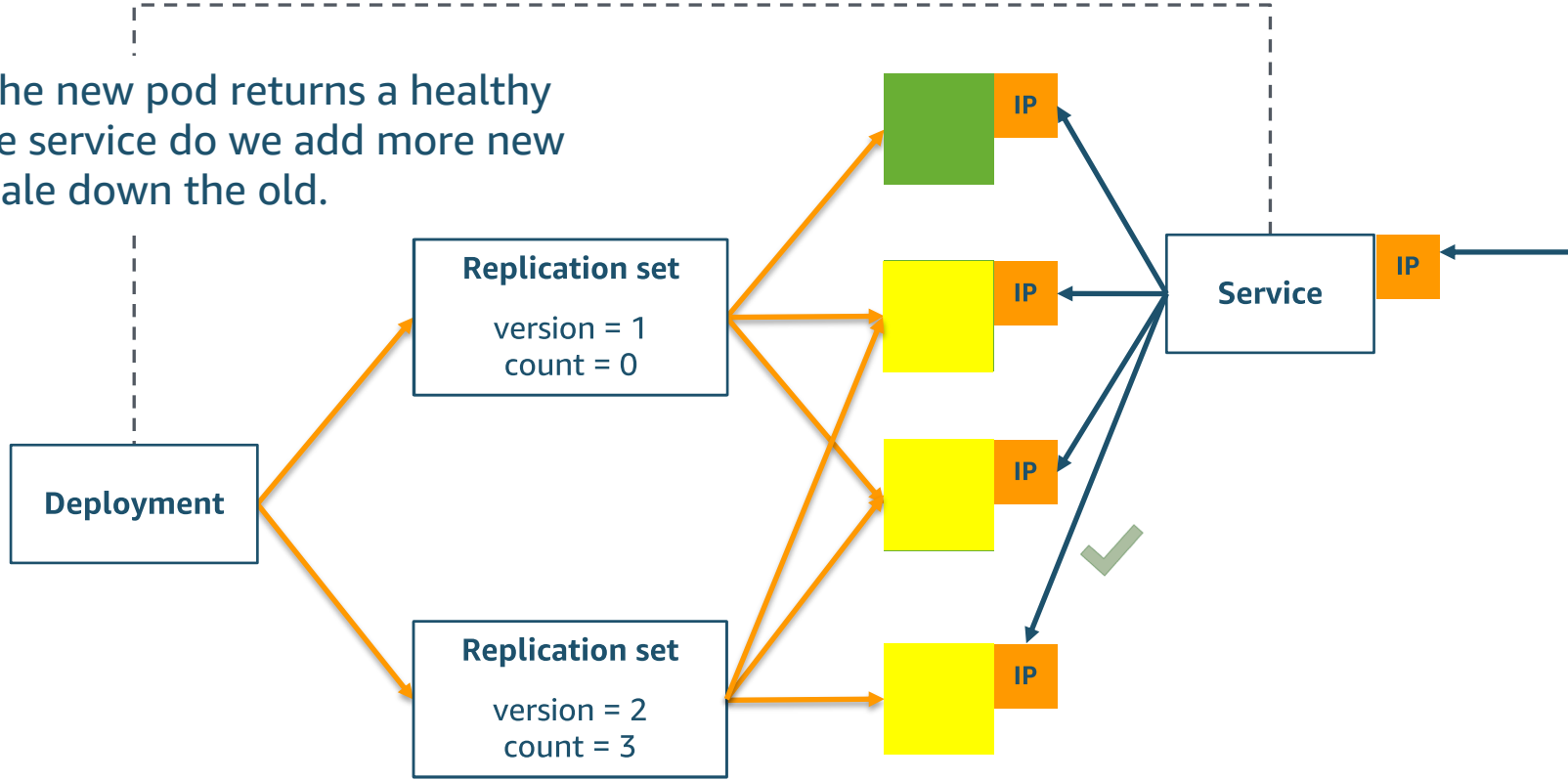
Deployments

The deployment creates a **new** replication set for our new pod version.



Deployments

Only after the new pod returns a healthy status to the service do we add more new pods and scale down the old.



aws

Running Kubernetes on AWS

51%

of Kubernetes workloads run on AWS today

— Cloud Native Computing Foundation

Elastic Container Service for Kubernetes

Managed K8s control plane — highly available API server and etcd nodes

Bring your own EC2 worker nodes

Core tenets

- Platform for enterprises to run production-grade workloads
- Provides a native and upstream Kubernetes experience – Kubernetes certified
- Seamless integration with AWS services
- Actively contributes to the Kubernetes project

APIs

```
aws eks create-cluster \
  --name <> \
  --role-arn <> \
  --resources-vpc-config <> \
  ...
```


aws

eksctl – eks the easy way

CLI to provision and managed your EKS cluster

Quickly get started without worrying about VPC, IAM and Node design

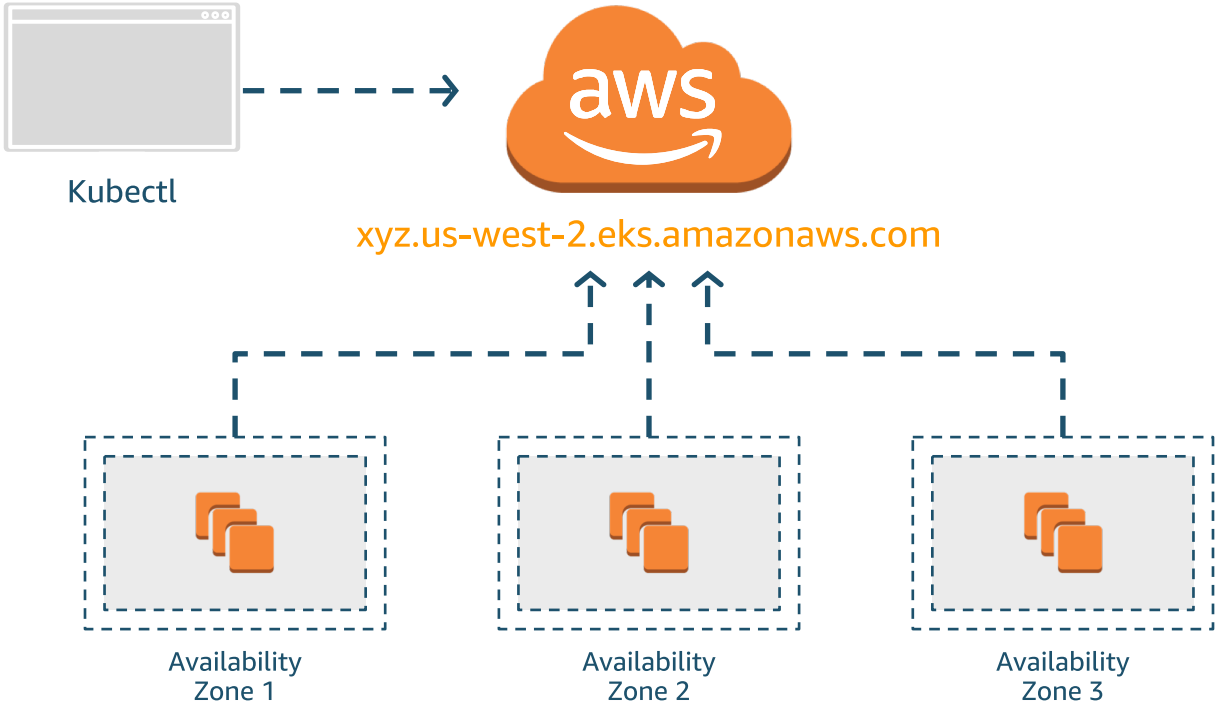
eksctl create cluster

Or define a config file for more detailed configuration control

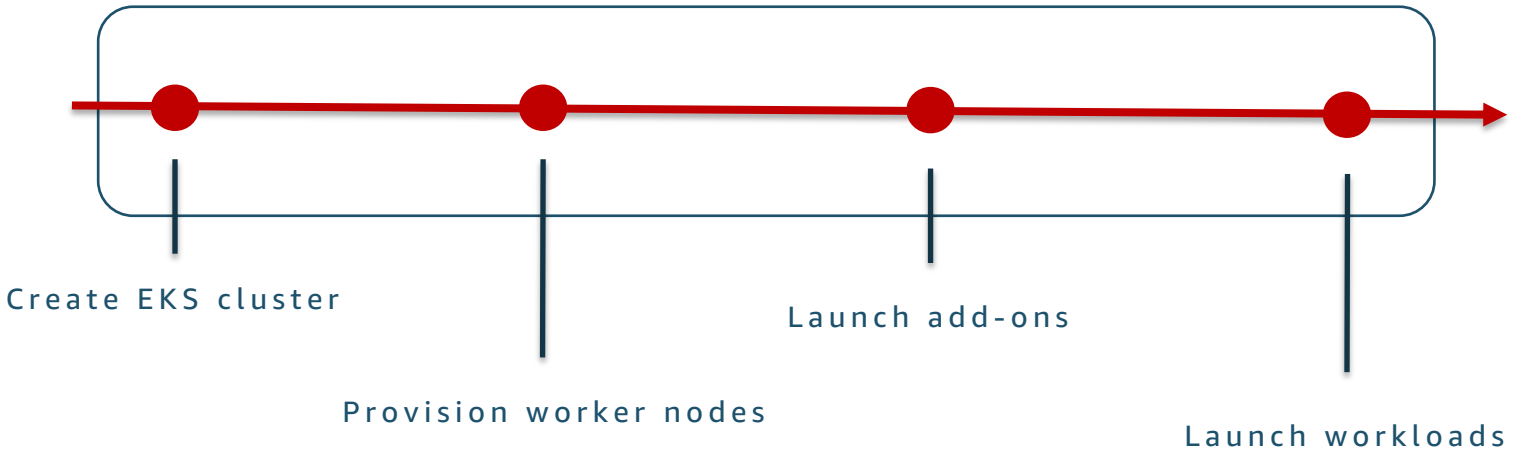
eksctl create cluster –f config.yaml

```
apiVersion: eksctl.io/v1alpha4
kind: ClusterConfig

metadata:
  name: demo
  region: ap-northeast-2
  version: '1.11'
nodeGroups:
...
```

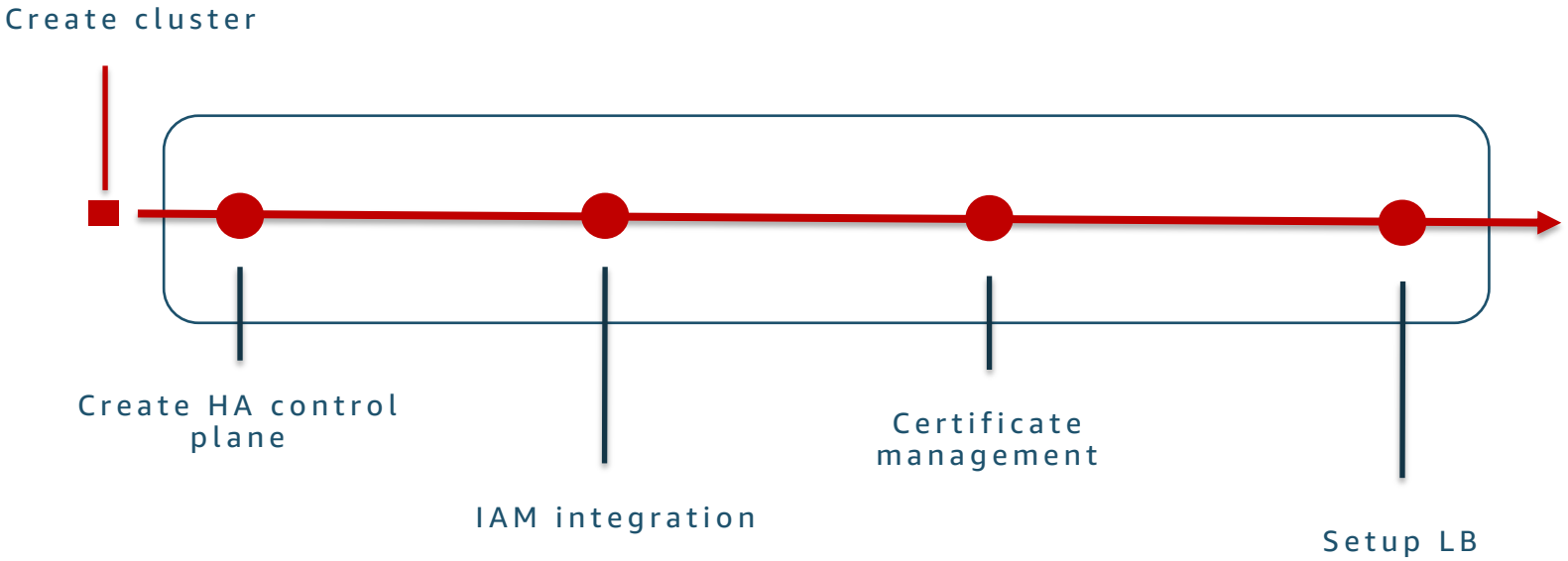

aws



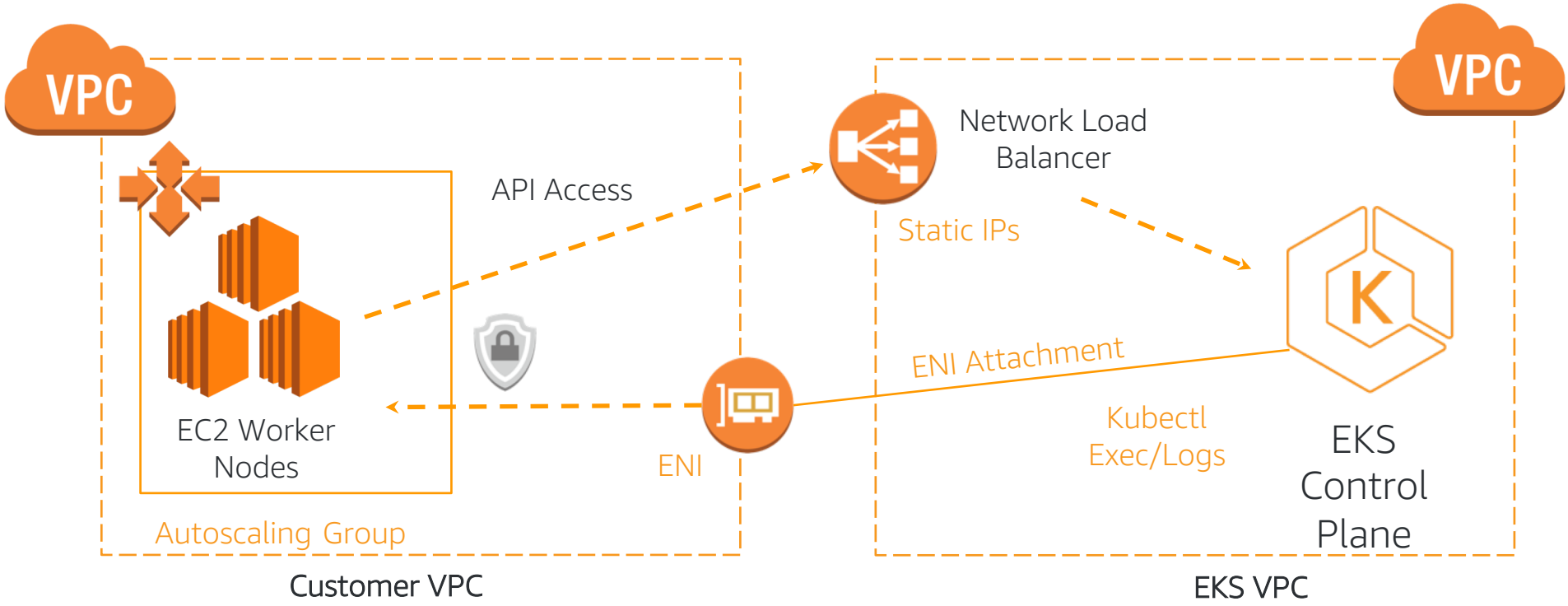
EKS Customers



EKS – Kubernetes control plane



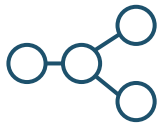
EKS Architecture



aws

Container Networking Interface (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

aws

VPC CNI plugin

- Bridge between the K8s land – AWS VPC
  - AWS Routable IPs
- Thin layer – no performance impact
- Pod IP ⟸ ENI secondary IP

Cluster Identity & Access Management (IAM)

IAM enables secure access to AWS services and resources

Two IAM roles created for K8s cluster

Service	Access level	Resource
Allow (7 of 109 services) Show remaining 102		
Auto Scaling	**Limited**: List, Write	All resources
EC2	Full access	All resources
EC2 Container Registry	**Full**: List **Limited**: Read	All resources
ELB	Full access	All resources
ELB v2	Full access	All resources
Route 53	**Limited**: List	All resources
S3	**Limited**: List, Read, Write, Permissions management	Multiple

Service	Access level	Resource
Allow (4 of 109 services) Show remaining 105		
EC2	**Full**: List **Limited**: Read, Write	All resources
EC2 Container Registry	**Full**: List **Limited**: Read	All resources
Route 53	**Limited**: List	All resources
S3	**Limited**: List, Read, Write, Permissions management	Multiple

aws

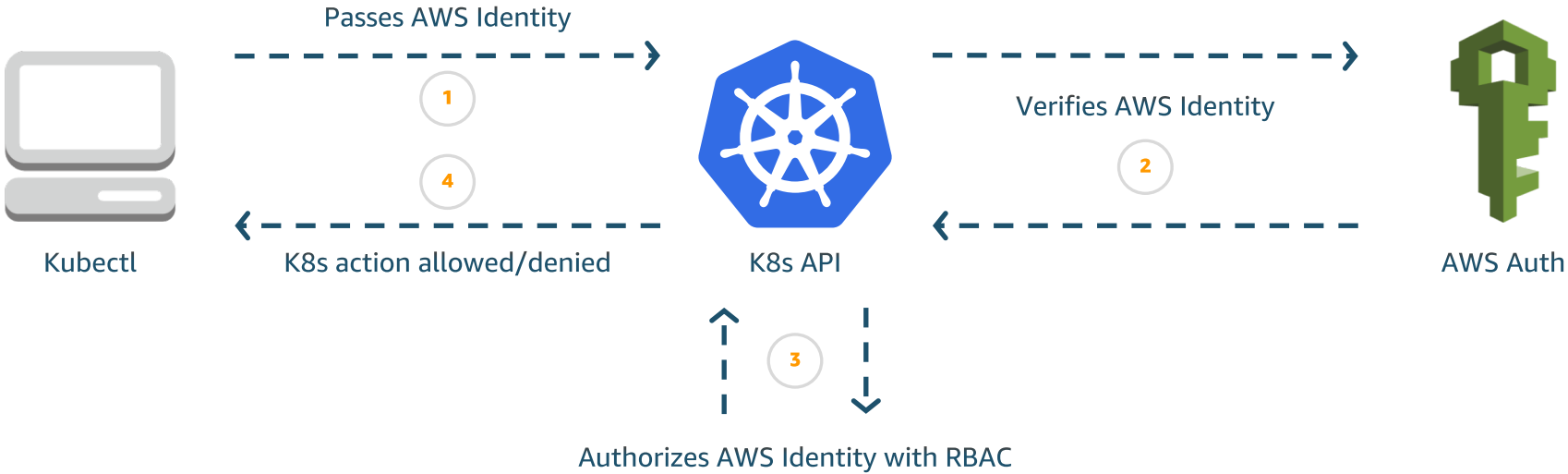
AWS IAM Authenticator for Kubernetes

- AWS native access management
- Built in collaboration with Heptio
- Kubectl and worker nodes
- Works with Kubernetes Role-Based Authentication (RBAC)

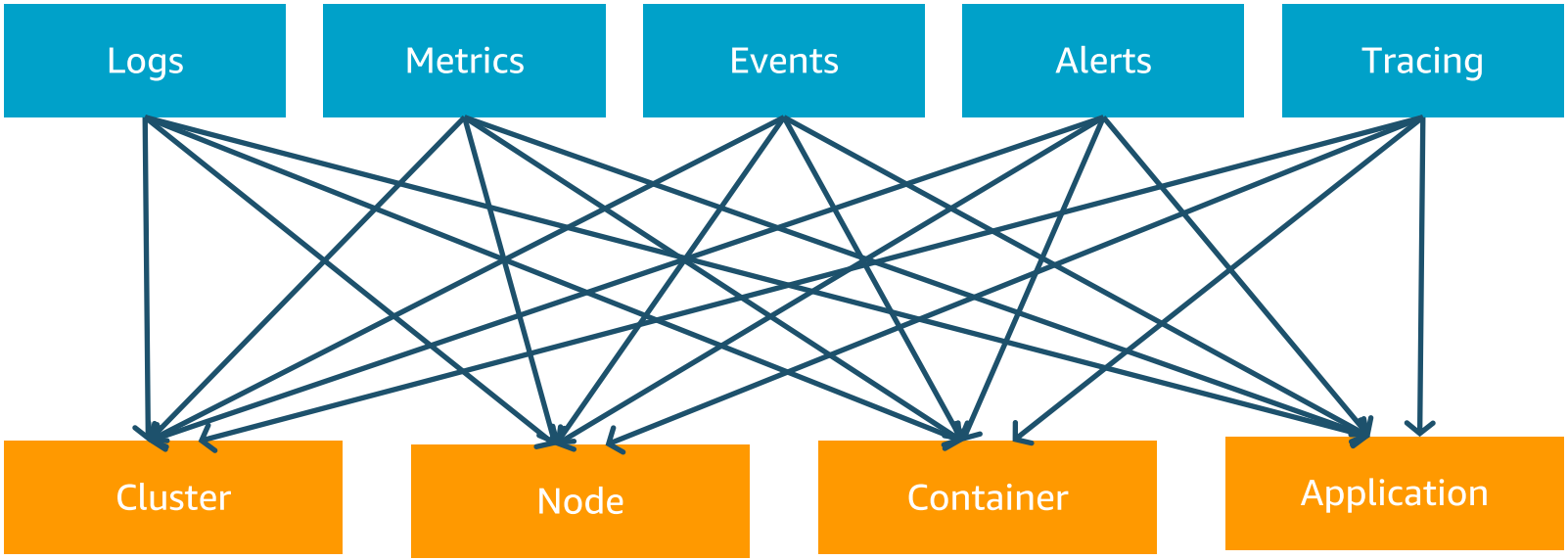
aws

IAM roles for Kubectl

github.com/kubernetes-sigs/aws-iam-authenticator

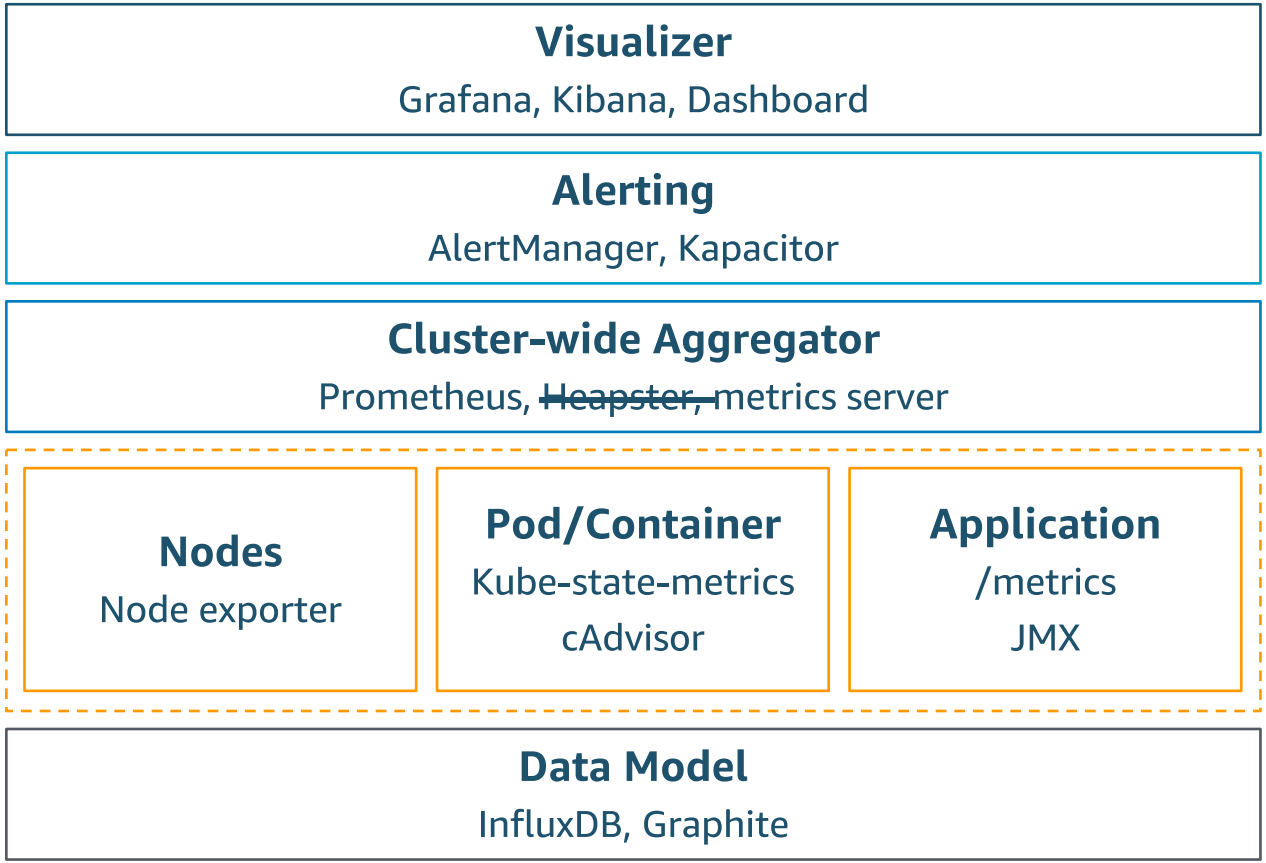


Visibility throughout a Kubernetes cluster



aws

Metrics



aws

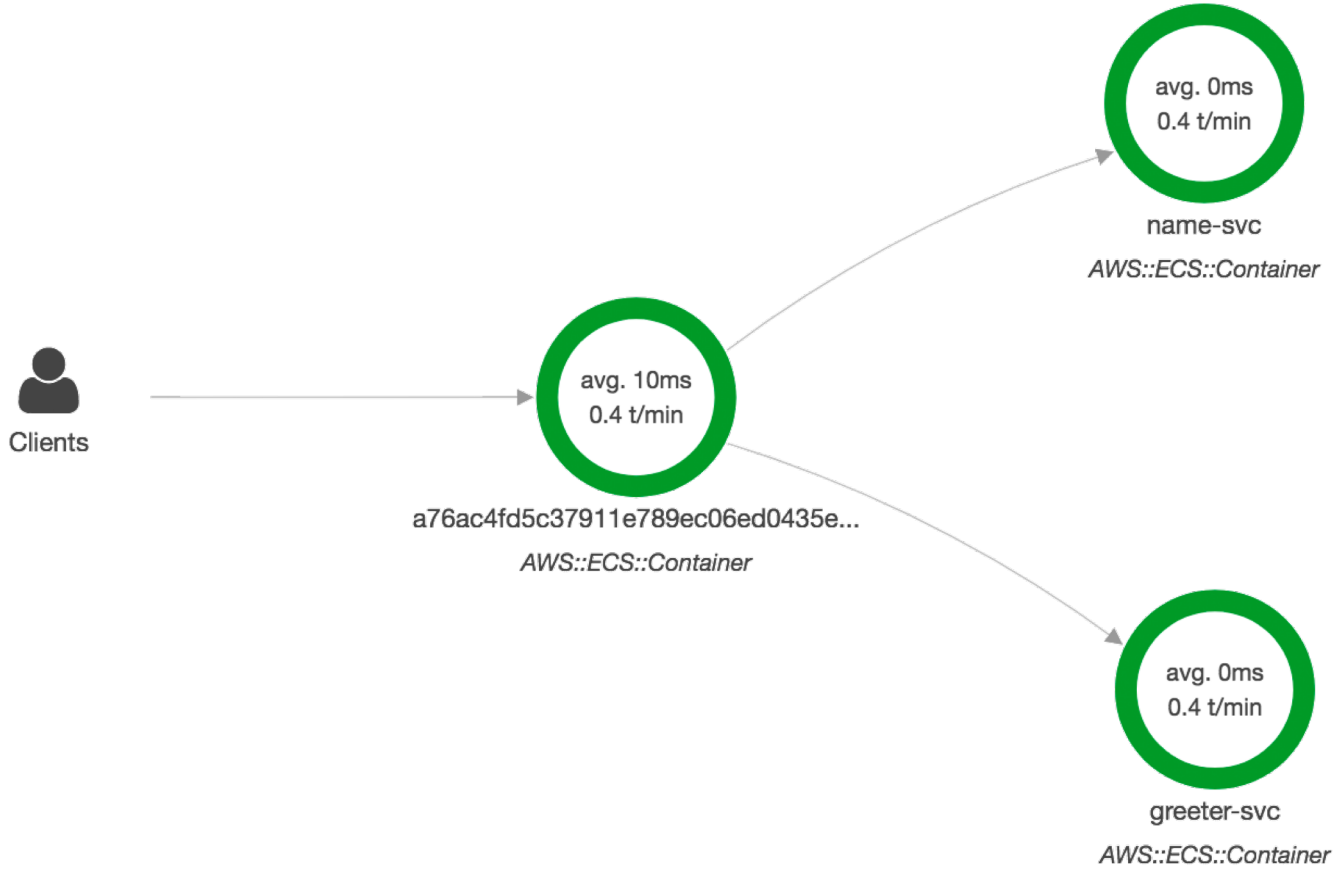
Tracing

Analyze and debug production, distributed applications, such as those built using a microservices architecture

Identify and troubleshoot the root cause of performance issues and errors

End-to-end view of requests as they travel through your application

aws



CICD for Kubernetes Apps

CI/CD of apps on Kubernetes—choices

Jenkins

AWS CodePipeline, AWS CodeCommit, AWS CodeBuild

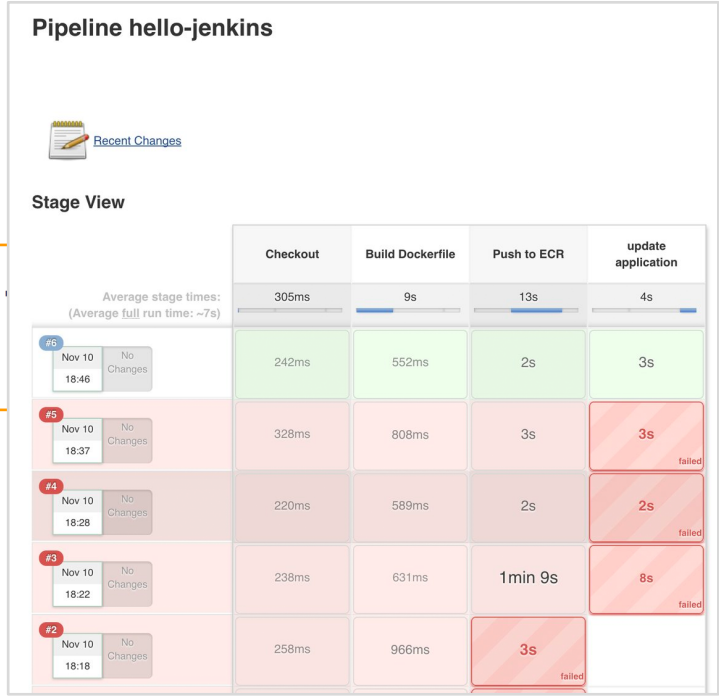
AWS partners

- GitLab
- Shippable
- CircleCI
- Codeship

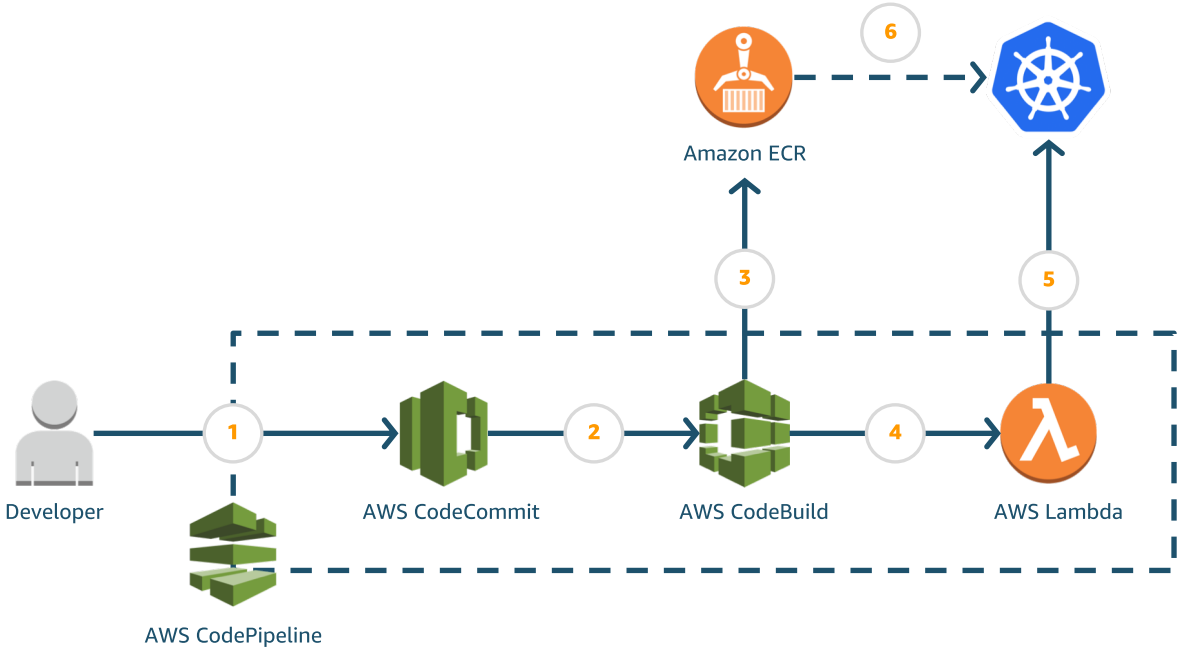
aws

Jenkins

```
node {

    stage 'Checkout'
    git 'https://github.com/omarlari/aws-container-sample-app.git'

    stage 'Build Dockerfile'
    docker.build('hello')

    stage 'Push to ECR'
    sh ("eval \$(docker run awscli aws ecr get-login --region ${REGION} --no-include-email | sed '
    docker.withRegistry('https://${ECR_REPO}') {
        docker.image('hello').push('${BUILD_NUMBER}')
    }

    stage 'update application'

    kubernetes: { node {
    docker.image('kubectl').inside("--volume=/home/ec2-user/.kube:/config/.kube"){
        sh 'kubectl describe deployment ${APP}'
        sh 'kubectl set image deployment/${APP} hello=${ECR_REPO}/hello:${BUILD_NUMBER}'
        sh 'kubectl describe deployment ${APP}'
        }
    }}
}
```



EKS Deployment Pipeline



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 Golang SDK to update a deployment
6. Fetches new container image and performs a rolling update of deployment

aws

Questions?

Workshop

Start at https://eksworkshop.com