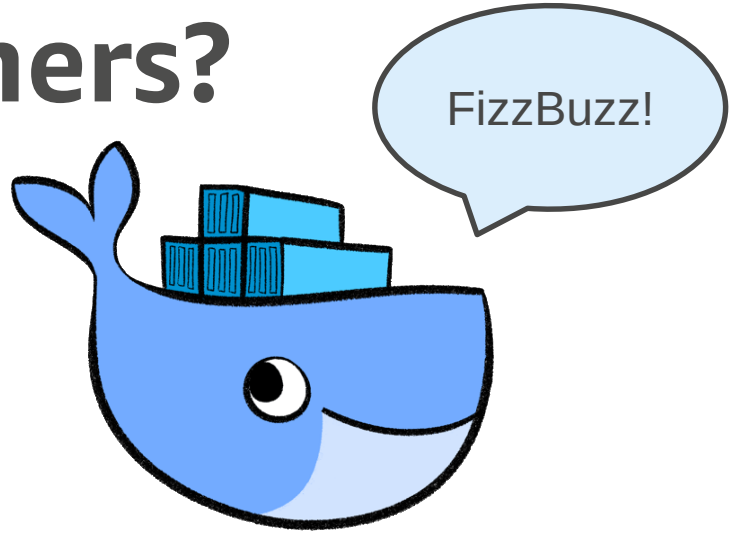
Getting Started with Docker on AWS

Nathan Peck, Developer Advocate for Container Services

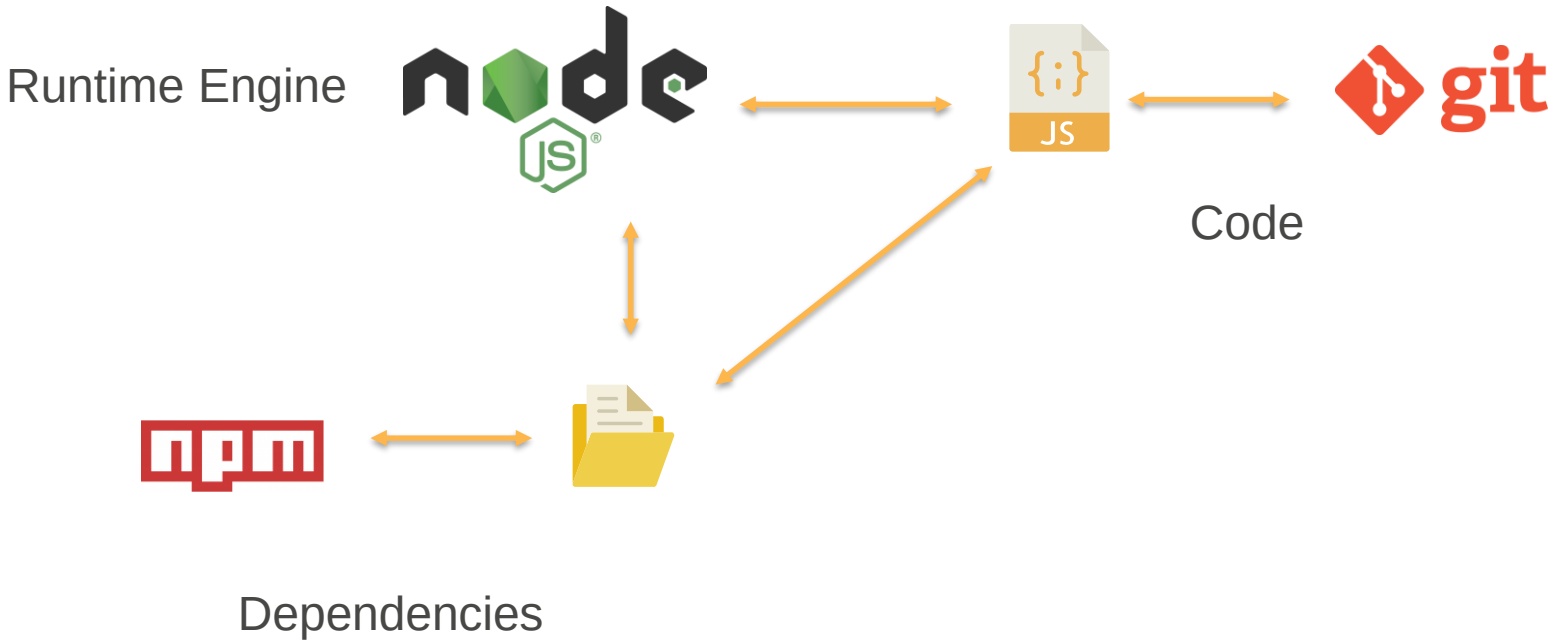
March 21, 2018

aws

Why Docker containers?



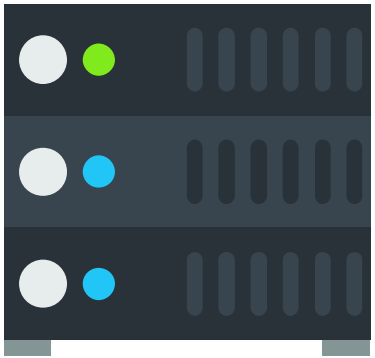
Application environment components



aws

Different environments

Local Laptop



Staging / QA

Production

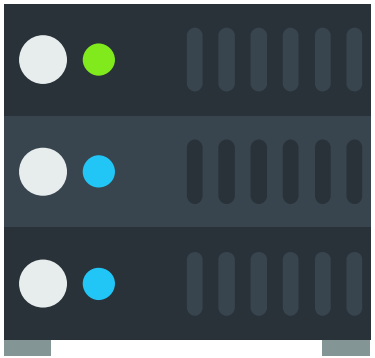
On-Premise

It worked on my machine, why not in production?

v6.0.0

Local Laptop

v7.0.0



Staging / QA

v4.0.0

Production

v7.0.0

On-Prem

aws

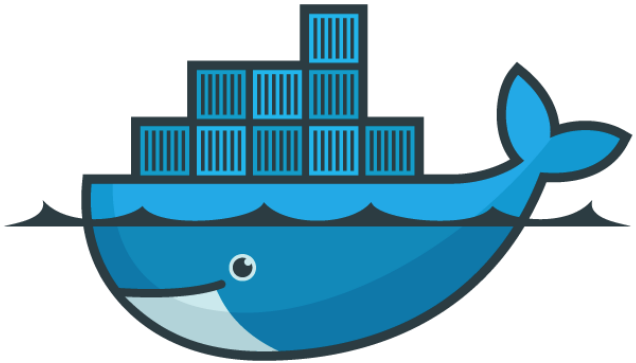
Docker

Lightweight container virtualization platform.

Licensed under the Apache 2.0 license.

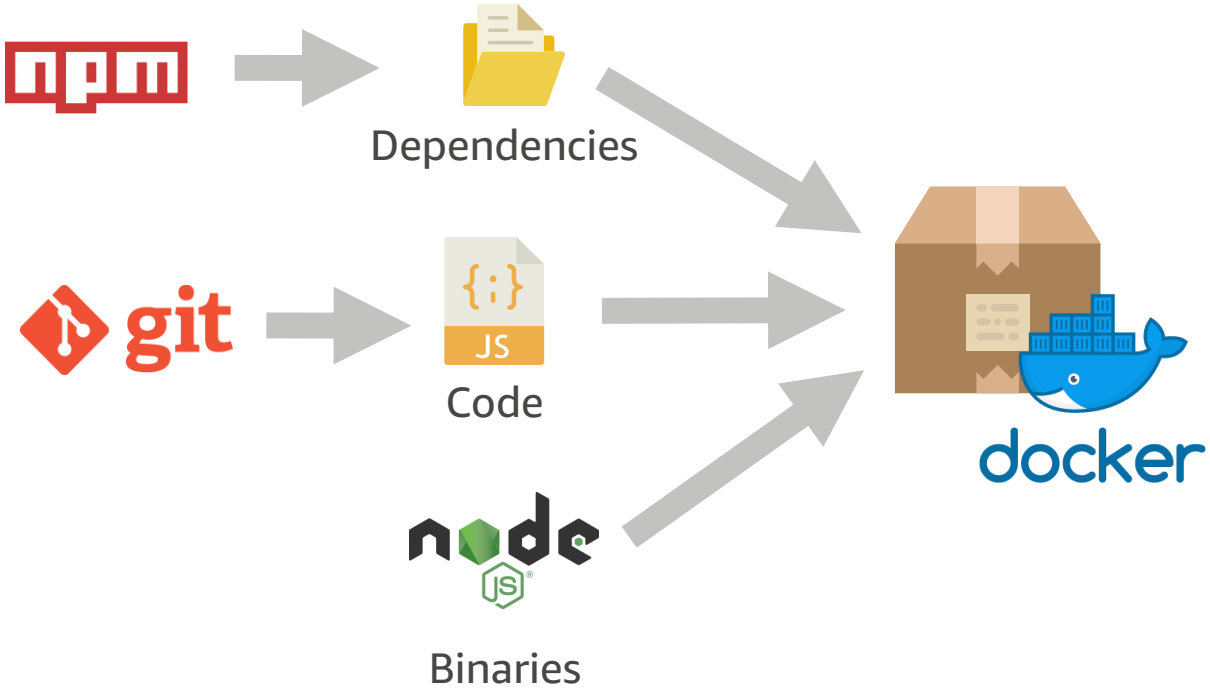
First released March 2013

Built by Docker, Inc.



aws

docker build



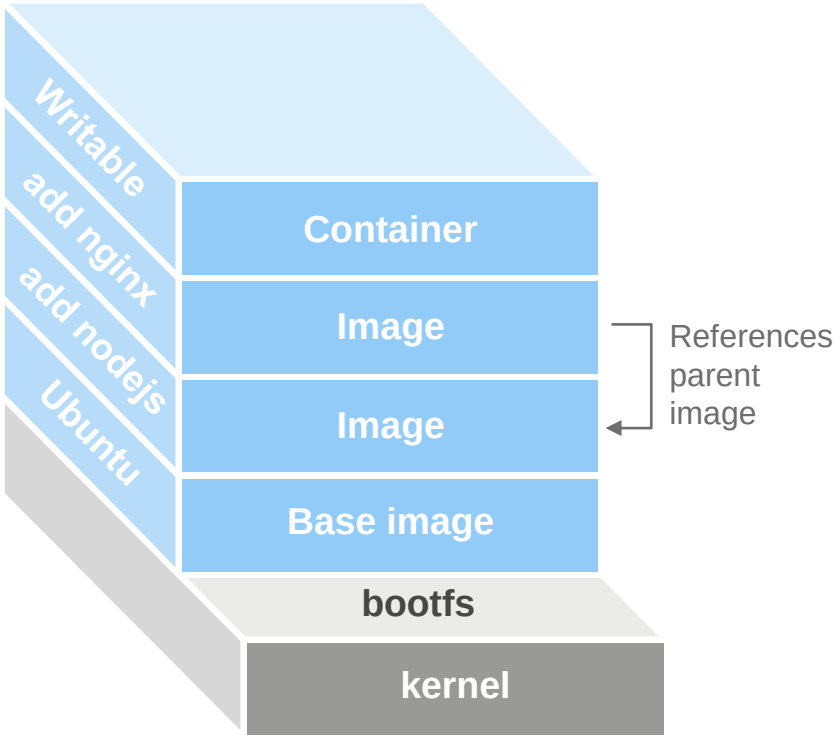
aws

Docker Image

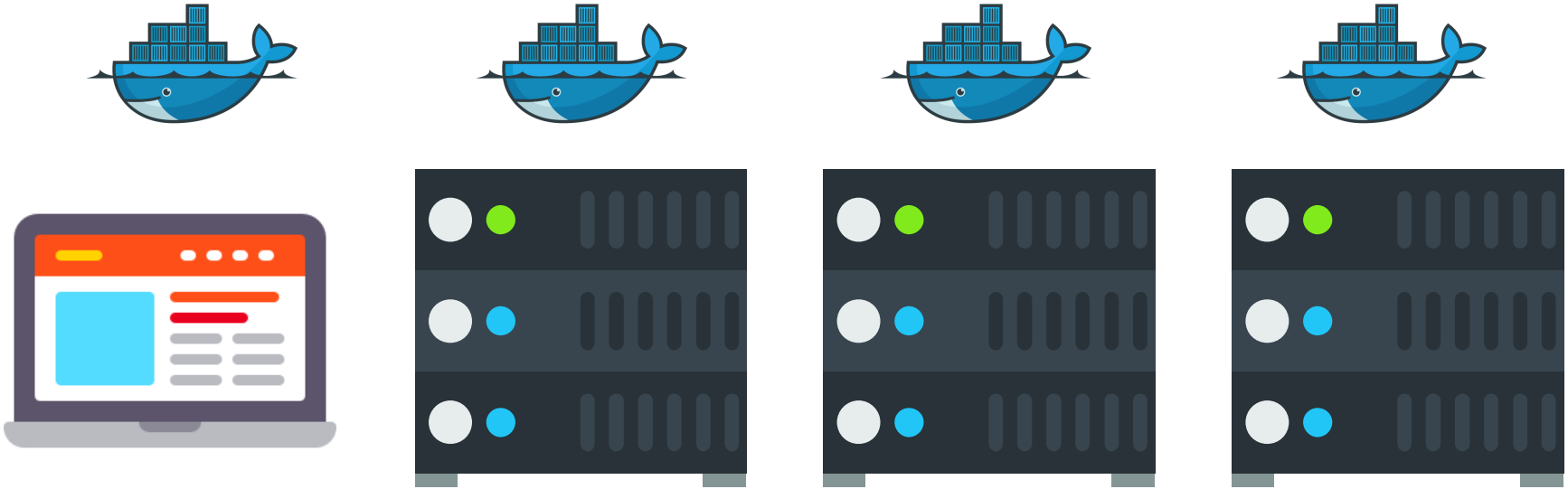
Read only image that is used as a template to launch a container.

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

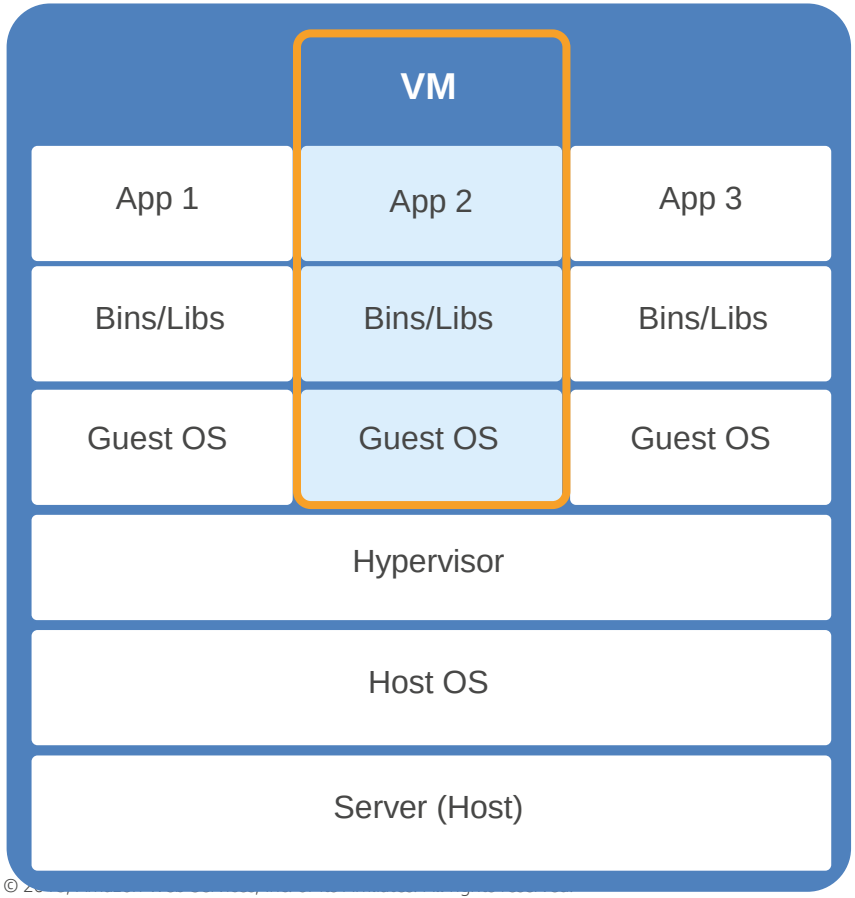
Docker file for easy, reproducible builds.

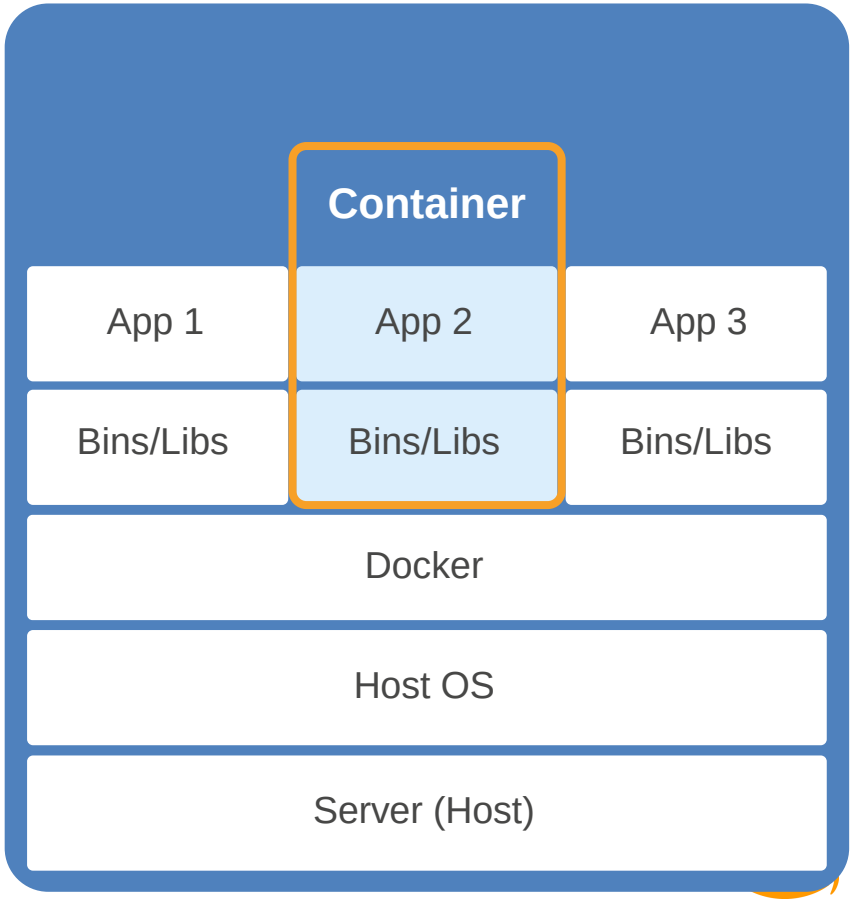


Four environments, same container



VM vs Container





Benefits

Portable runtime application environment

Package application and dependencies in a single artifact

Run different application versions (different dependencies) simultaneously

Faster development & deployment cycles

Better resource utilization

aws

Use Cases

Consistent environment between Development & Production

Service-Oriented Architectures & Microservices

Short lived workflows, batch jobs, cron jobs

Isolated environments for testing

aws

Just four commands to start using Docker

`docker build` (Create an image)

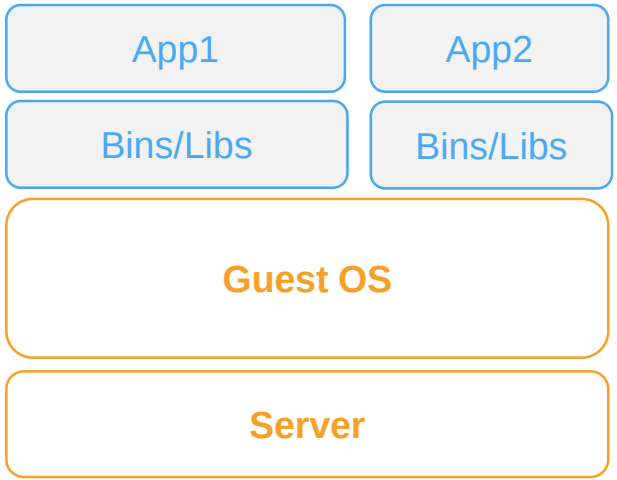
`docker tag` (Set a version for the image)

`docker push` (Store image in a registry to run later)

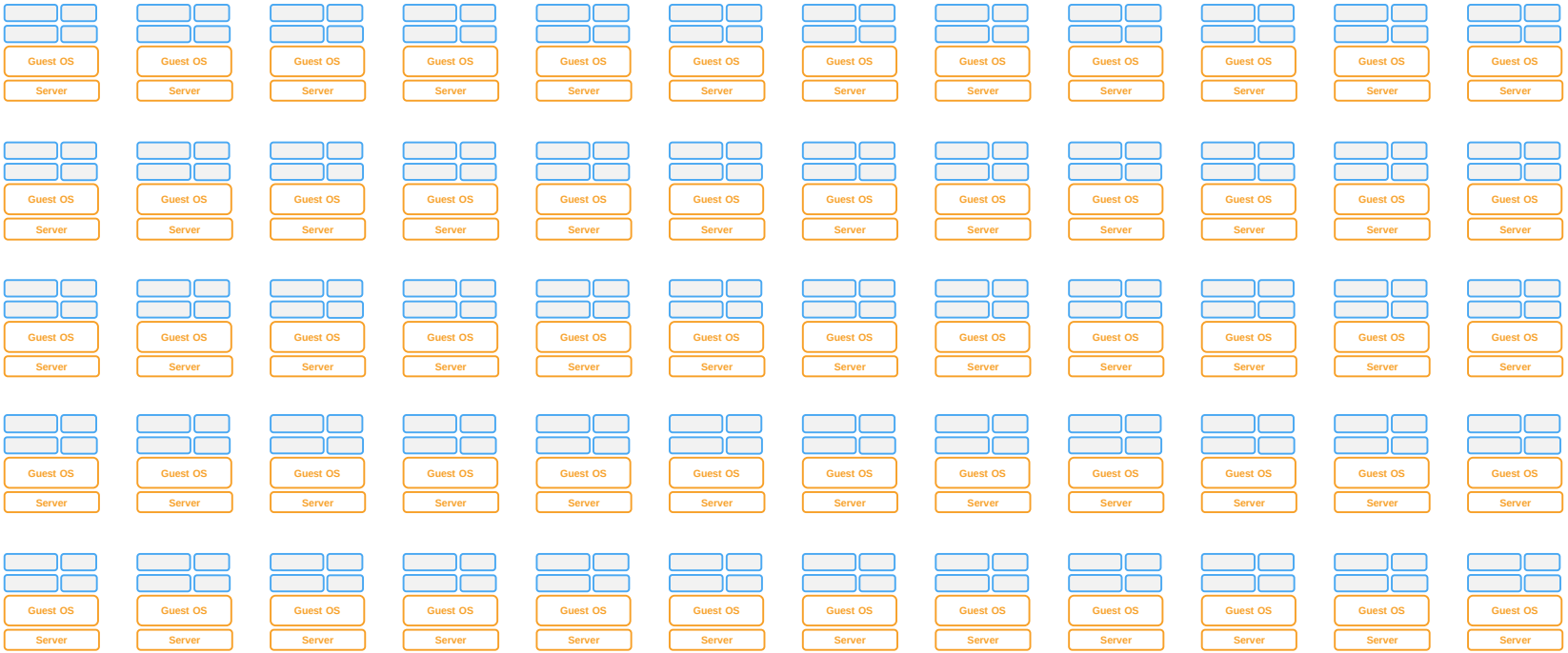
`docker run` (Run the image on a machine)

aws

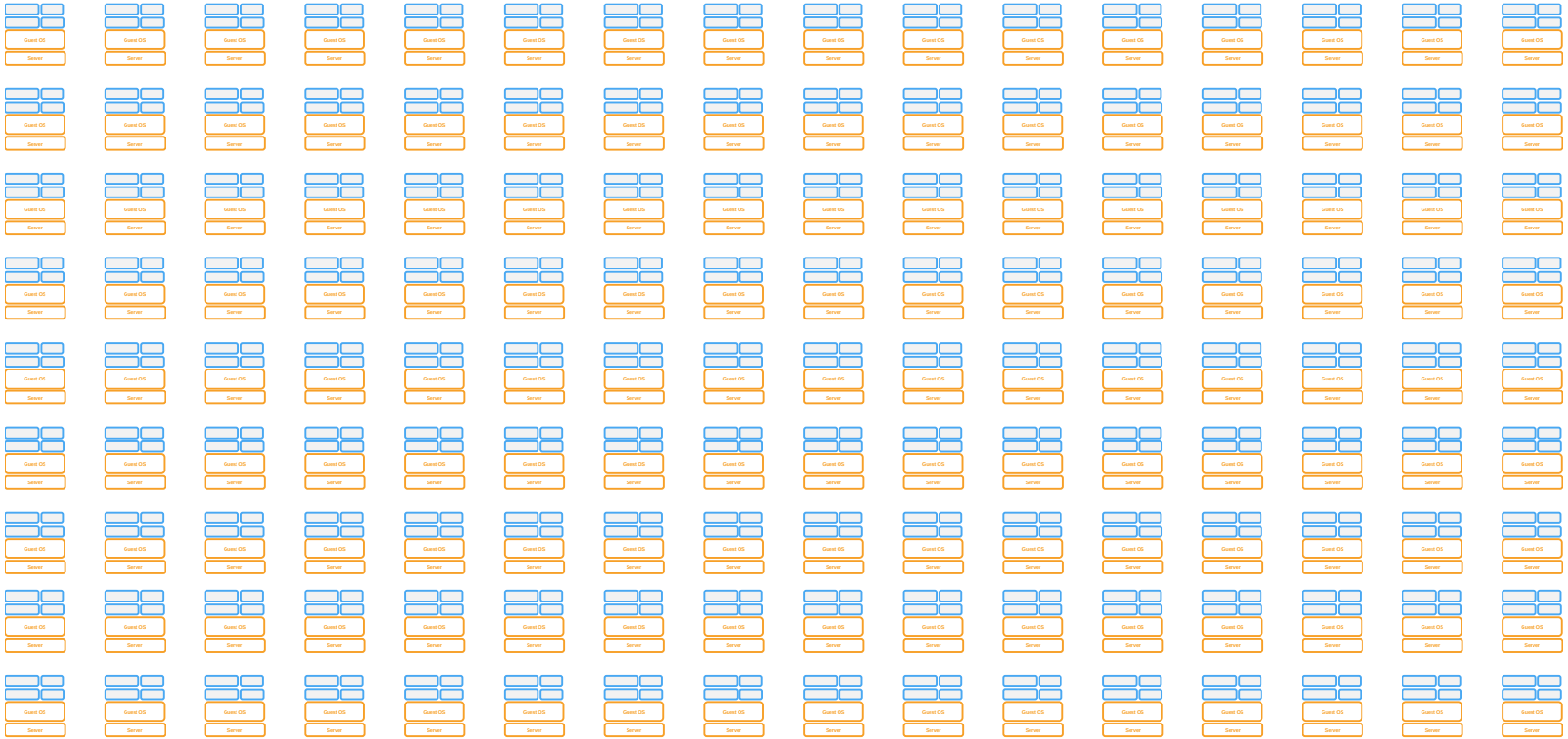
Using Docker is easy!



A few hosts?



Lots of hosts!



How do I deliver container images to all those hosts?

Amazon Elastic Container Registry (Amazon ECR)

- Cloud-based Docker image registry
- Fully managed
- Secure – images encrypted at rest, integrated with IAM
- Scalable and Highly Available
- Integrated with Amazon ECS and the Docker CLI

aws

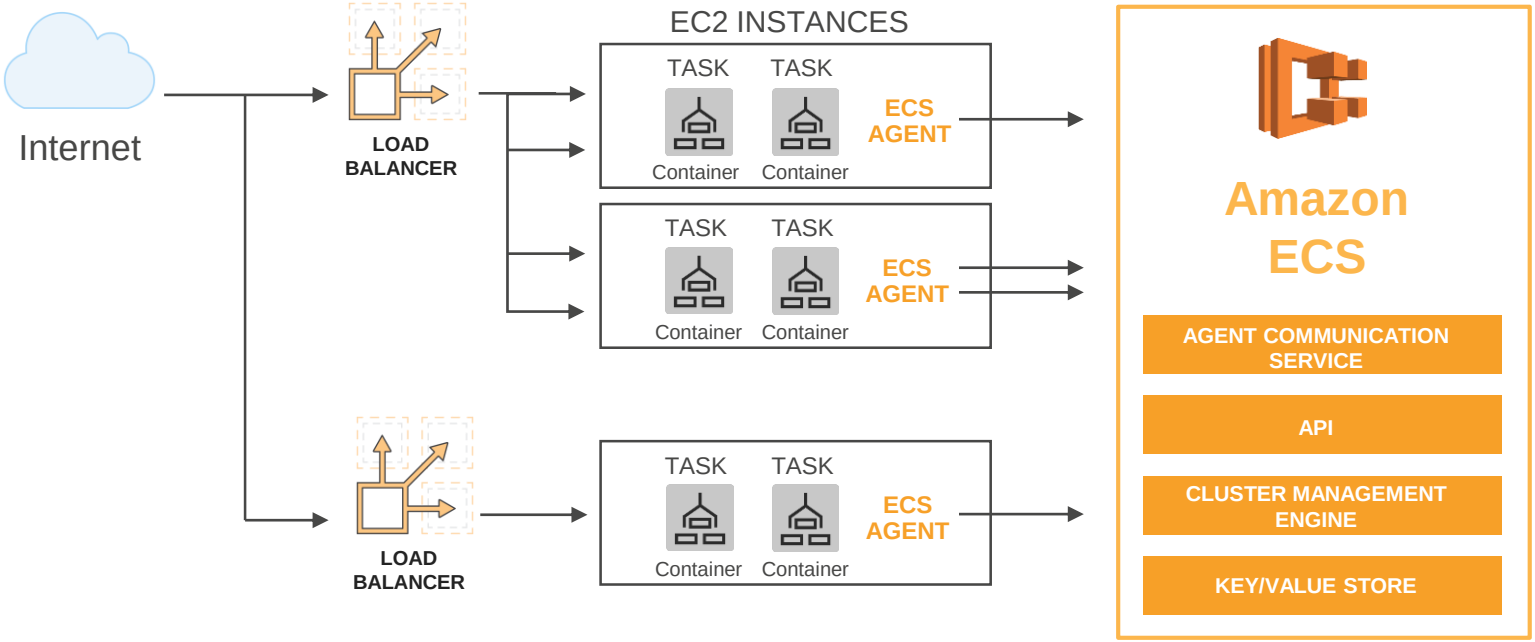
How can I get containers running on my hosts?

Amazon Elastic Container Service (Amazon ECS)

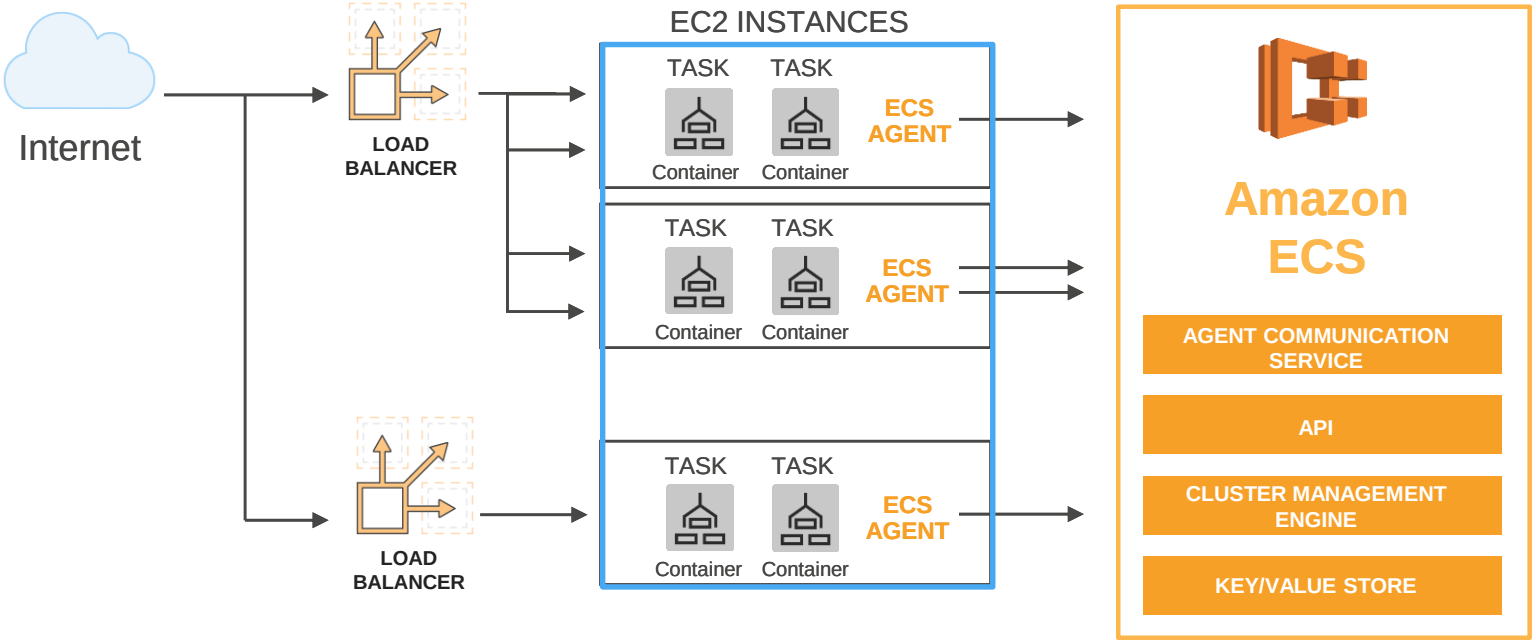
- Container management service
- Fully managed
- Scalable and Highly Available
- Microservices, batch workers, machine learning applications
- Integrated with
  - Amazon ECR
  - AWS networking, storage, management tools
  - AWS Fargate

aws

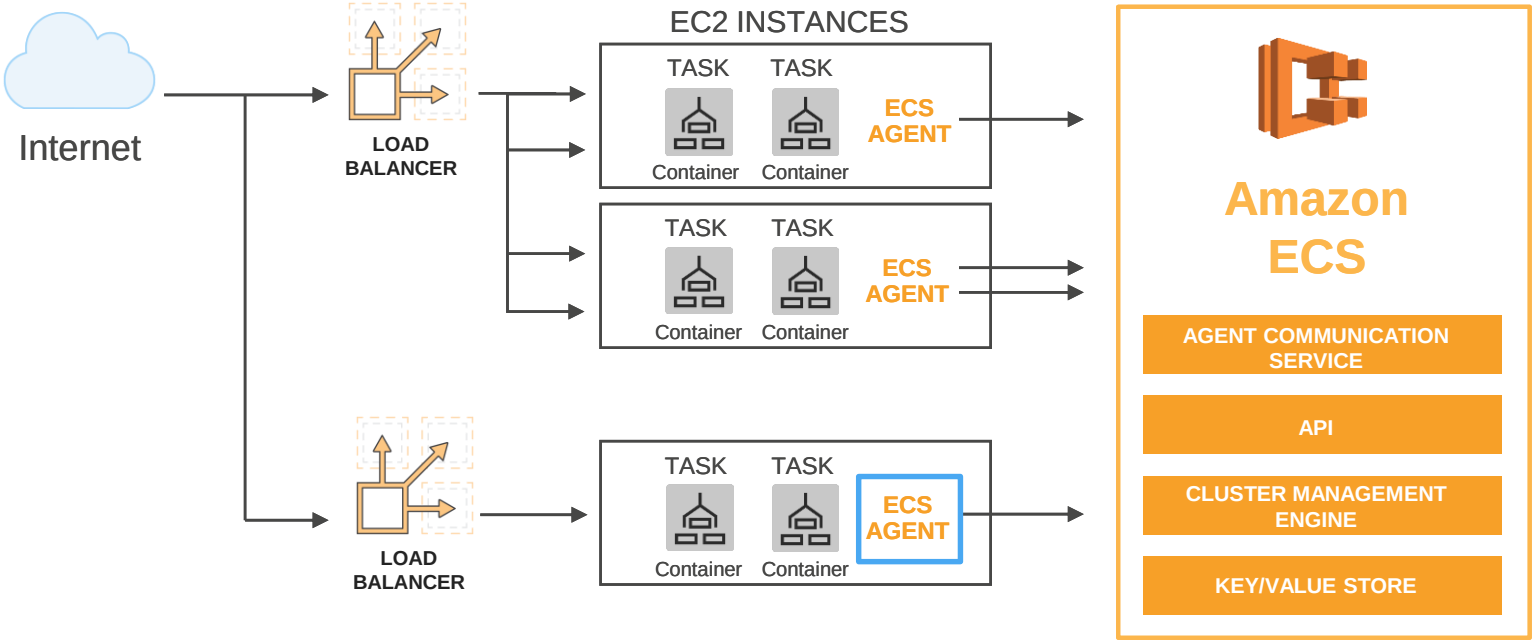
Amazon ECS



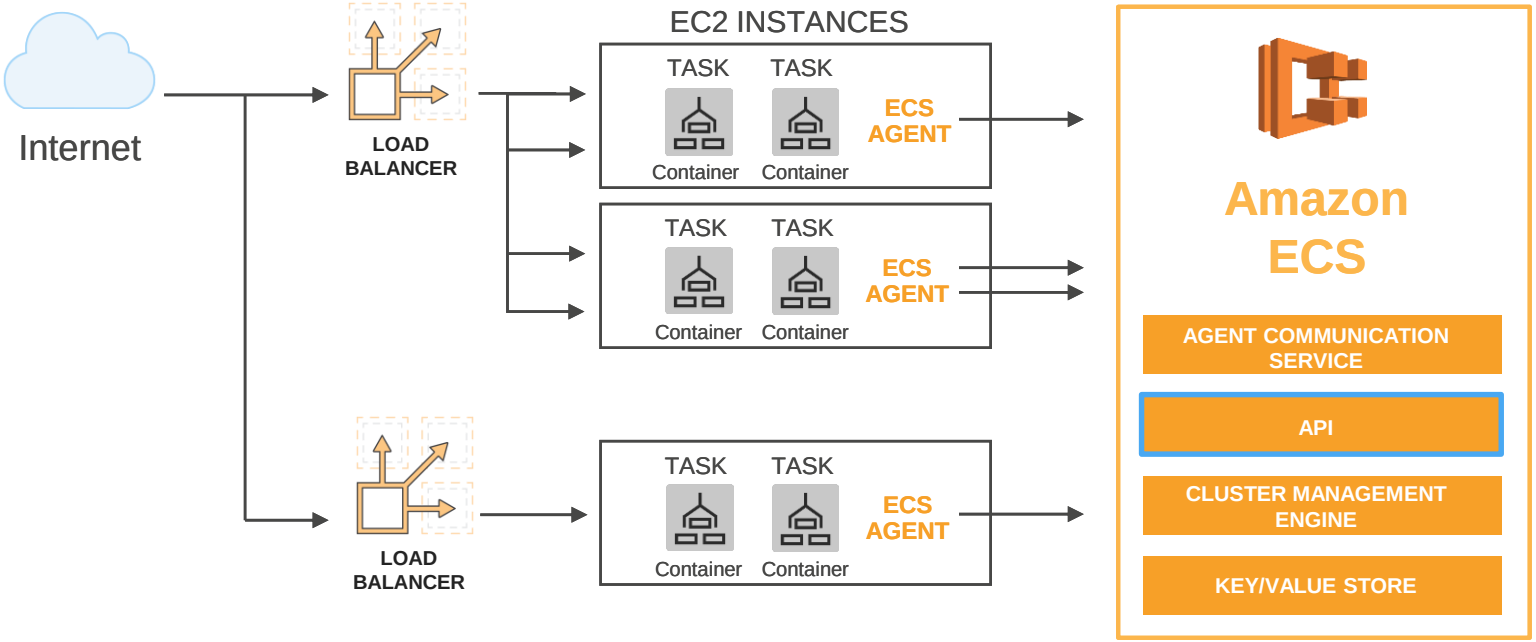
Cluster of hosts



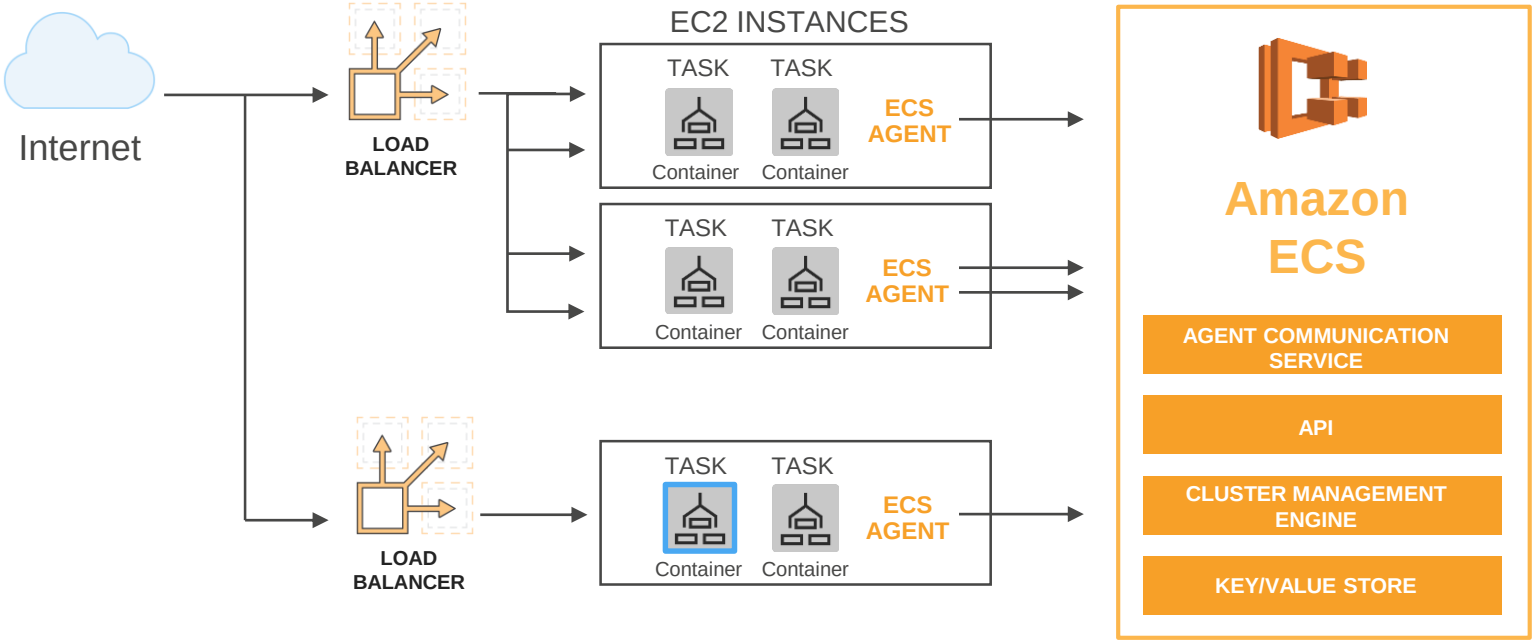
Lightweight agent on each host



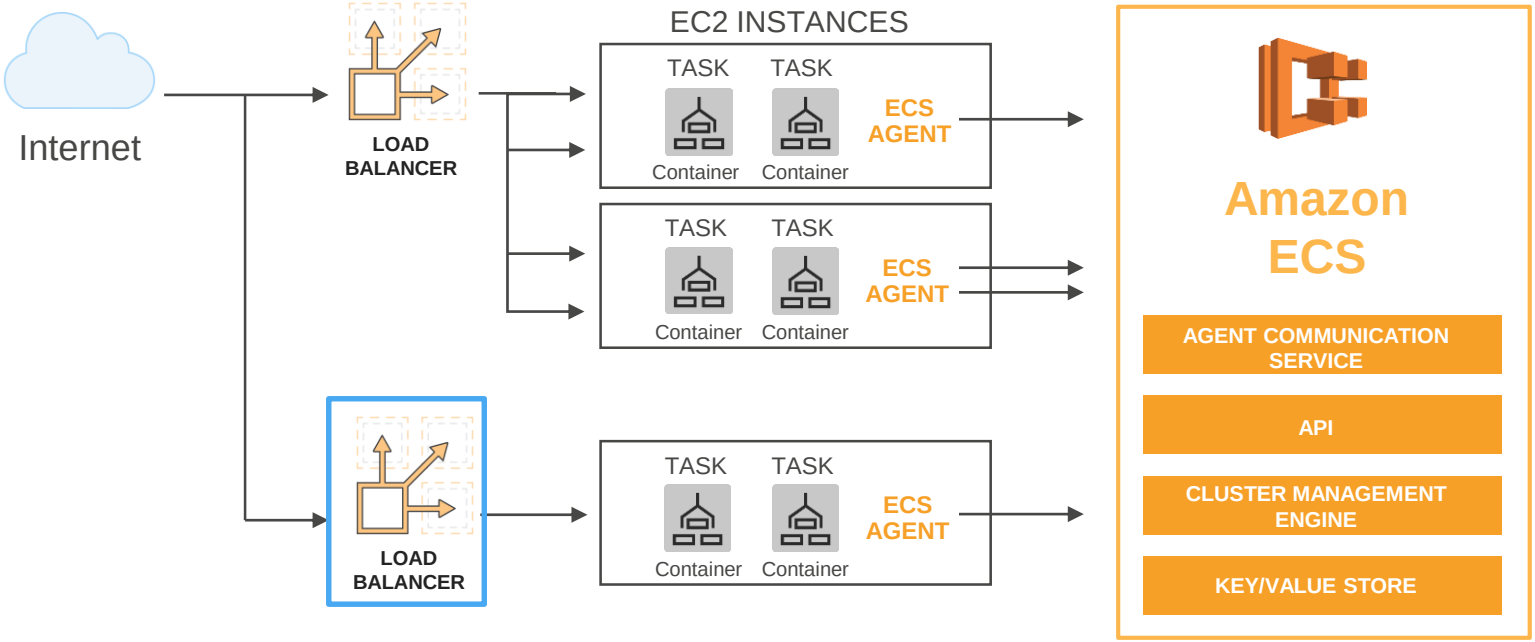
API for launching containers on the cluster



Container task is placed on a host



Traffic is sent to your host

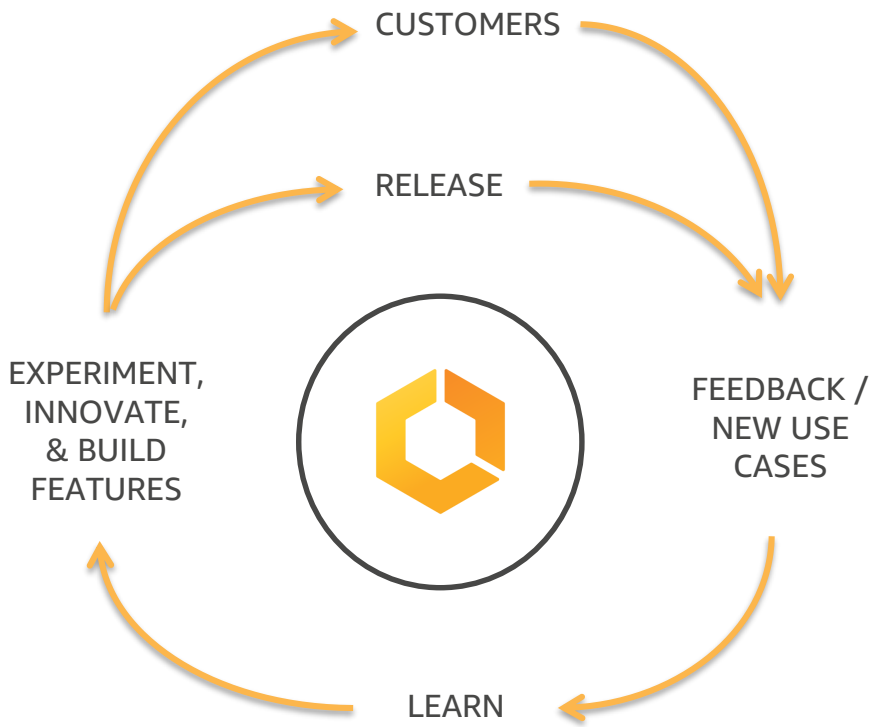


PRODUCTION WORKLOADS ON AWS

- AWS VPC networking mode
- Advanced task placement
- Deep integration with AWS platform
- ECS CLI
- Global footprint
- Powerful scheduling engines
- Auto scaling
- CloudWatch metrics
- Load balancers

Customers Using Containers at Scale

CUSTOMERS ARE OUR KEY!



50+

releases since 2015

What we did with ECS in 2017

Container access to environmental metadata

Task Elastic Network Interface

Application Load Balancer Support

Network Load Balancer support

Console support for SpotFleet

Console UX improvements

Cron and Cloudwatch Event Task scheduling

HIPAA eligibility

CLI V1.0

Override parameters for RunTask and StartTask APIs

Add attributes during boot

Windows containers

Seoul Region

Beijing Region

Container instance draining

Linux capabilities

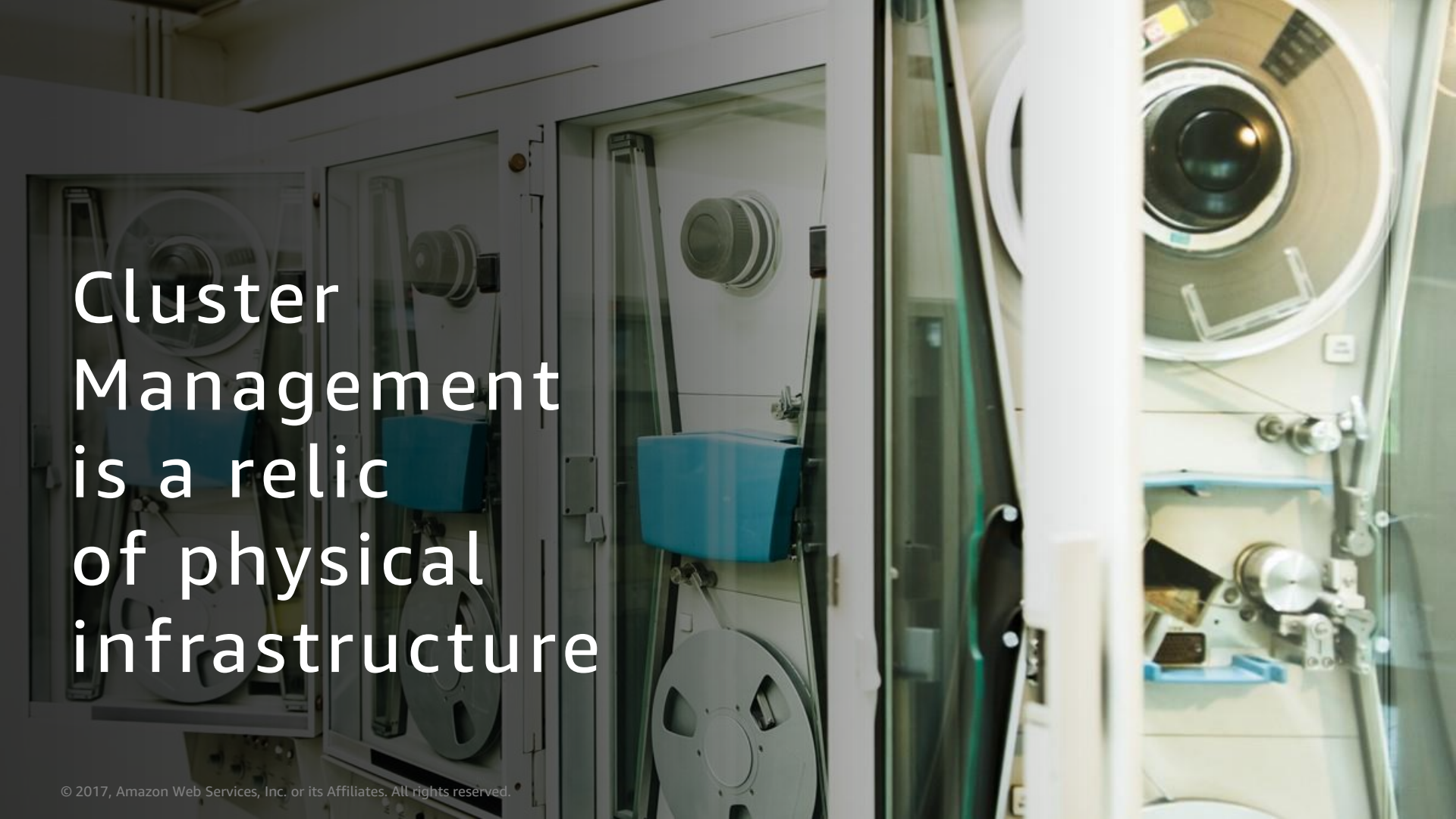
Support for Docker Privileged Mode

Support for Device and Init flags

Lifecycle Policies for container images

aws

I don’t want to deal with hosts at all!

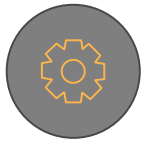


ENABLE FOCUS ON APPLICATIONS

INTRODUCING FARGATE!

AWS Fargate

CHANGING COMPUTE CONSUMPTION MODEL



No instances to manage

Task native API

Resource based pricing

Simple, easy to use, powerful – and new consumption model

PRODUCTION WORKLOADS ON AWS

+

AWS Fargate

- AWS VPC networking mode
- Advanced task placement
- Deep integration with AWS platform
- ECS CLI
- Global footprint
- Powerful scheduling engines
- Auto scaling
- CloudWatch metrics
- Load balancers

I want to use more open source in my environment

kubernetes

aws

CLOUD NATIVE
COMPUTING FOUNDATION

aws

"Run Kubernetes for me."

aws

"Native AWS Integrations."

"An Open Source Kubernetes Experience."

aws

+ aws

=

Amazon EKS

ELASTIC CONTAINER SERVICE FOR KUBERNETES

(EKS)

aws

GENERALLY
AVAILABLE
2018

aws

In Summary

- **Docker** allows you to easily run different code across different machines in a standardized, easily defined environment

- **Amazon Elastic Container Service** allows you to schedule and run Docker containers on AWS

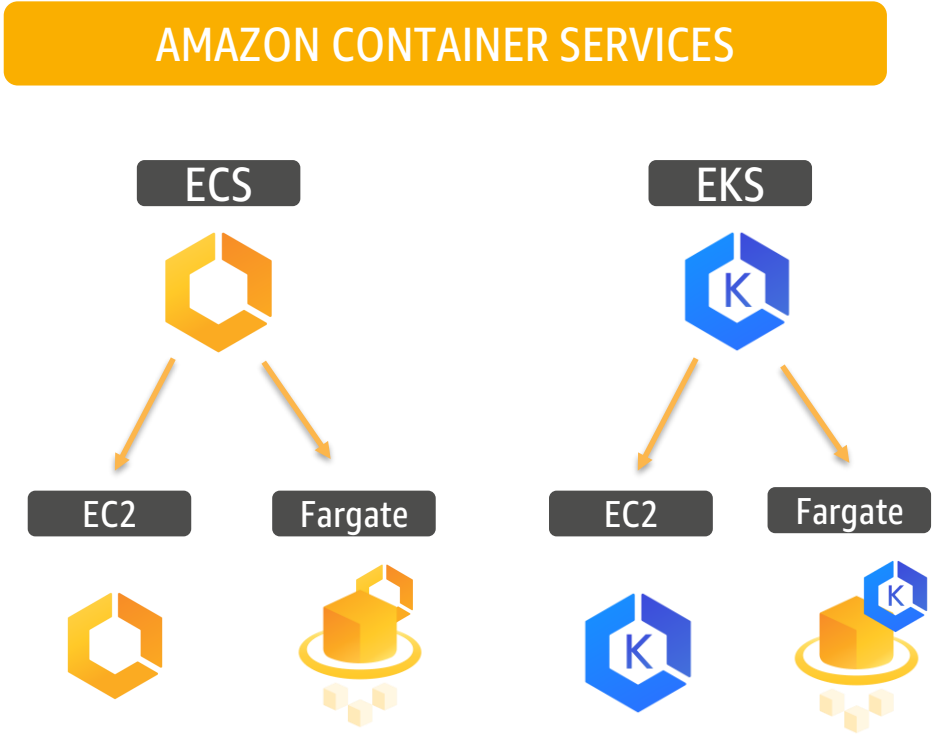
- **Amazon Elastic Container Service for Kubernetes** is a managed service for running Kubernetes on AWS

- **Amazon Elastic Container Registry** is a secure, private registry for Docker container images

aws

We give you the power to choose:

1. Choose your orchestration tool
2. Choose your launch type



The awesome-ecs project:

https://github.com/nathanpeck/awesome-ecs

README.md

A curated list of guides, development tools, and resources for Amazon Elastic Container Service (ECS). This list includes both community created content as well as content created by AWS.

Want to add something? Open a PR! 🙂

Pick your container hosting strategy:

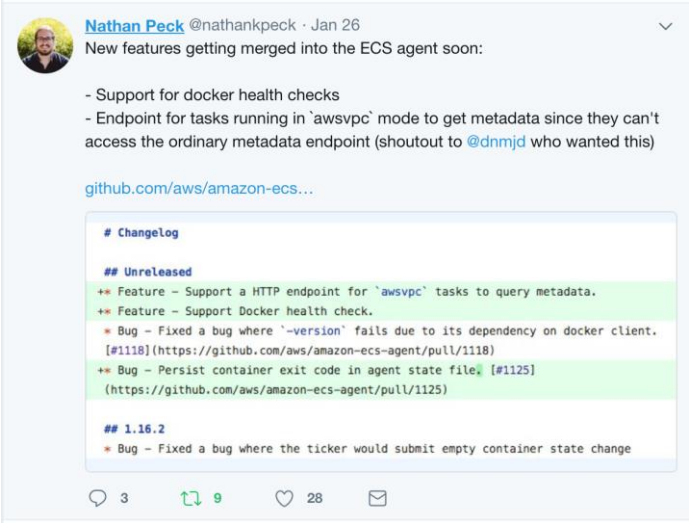
- AWS Fargate - AWS Fargate is a technology for Amazon ECS that allows you to run containers without having to manage servers or clusters.
- Self hosted in EC2 - Running your own cluster of EC2 instances to host your containers gives you the most control over price (ability to run on spot instances or reserved instances) as well as configuration.

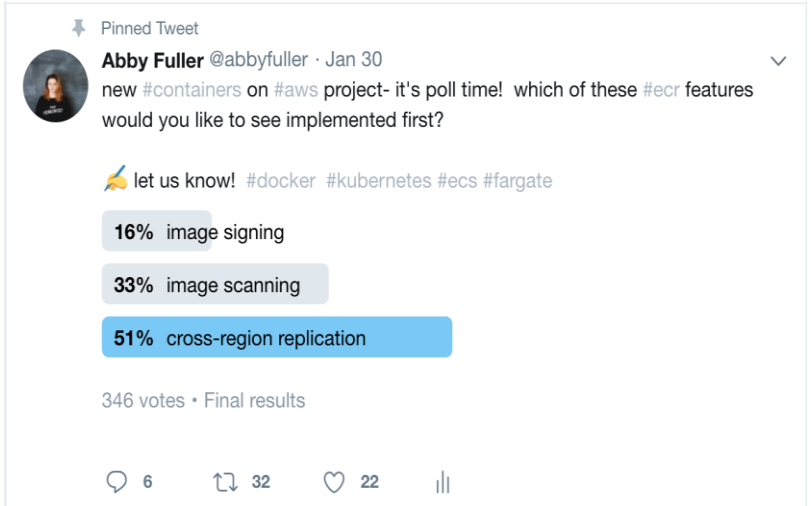
aws

We want to hear from all of you!

More focus on supporting Tasks as compute primitive, more focus on removing undifferentiated heavy lifting.

Our roadmap is driven by feedback:





Thank you!