



Containers on AWS

A Journey to Modern Applications

Johannes Langer, Senior Solutions Architect, AWS

June 2019



The new normal: companies are increasingly global and products are increasingly digital

47%

of CEOs said they are being challenged by the board of directors to make progress in digital business

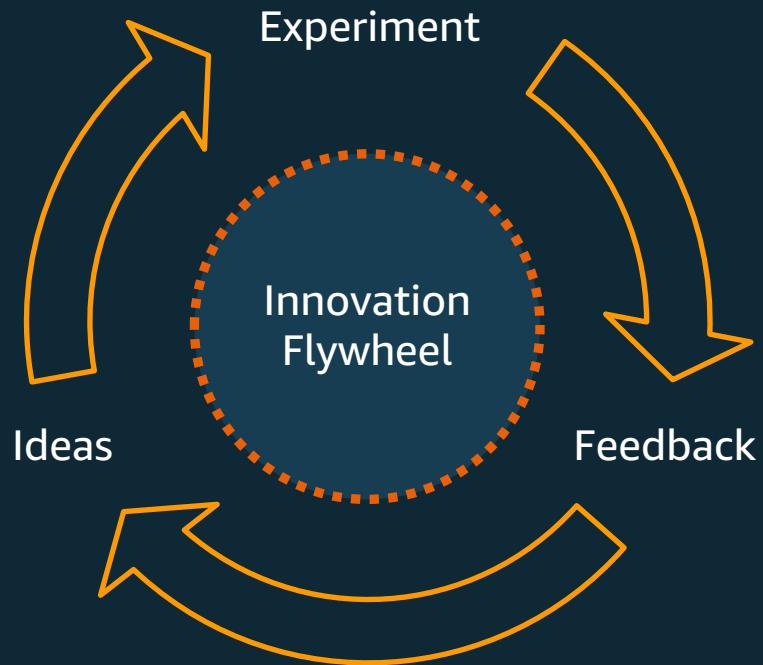
79%

of CIOs believe that digital business is making their IT organizations better prepared to change

67%

of all business leaders believe that they must pick up the pace of digitalization to remain competitive

To maintain competitive advantage, digital businesses must innovate as rapidly as possible



What changes have to be made in this new world?

Architectural patterns
Operational model
Software delivery

Changes to the architectural patterns



When the impact of change is small, release velocity can increase

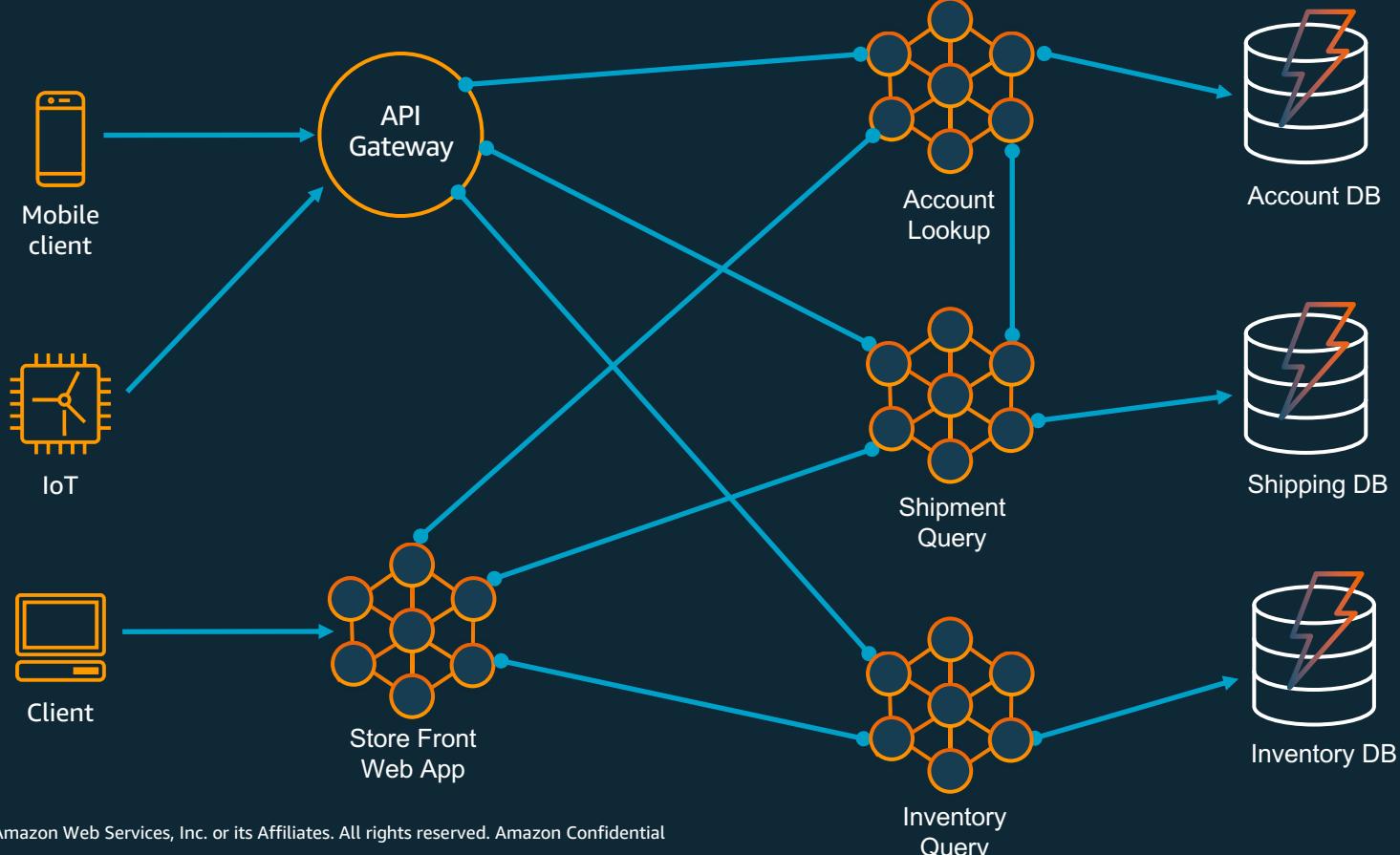


Monolith
Does everything



Microservices
Do one thing

Microservices architectures

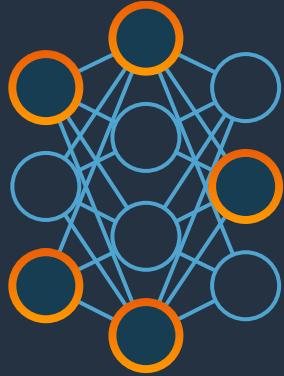




Cloud-native architectures are
small pieces, loosely joined

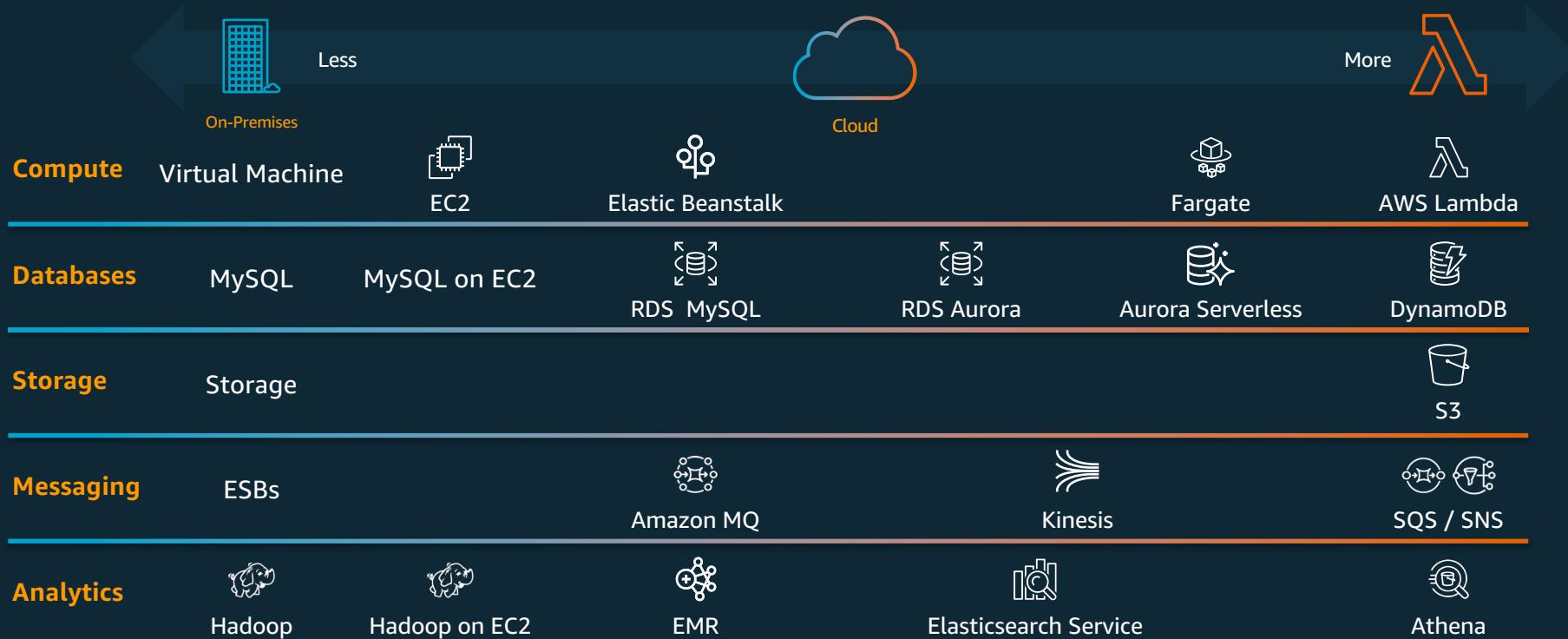
Changes to the operational model





Isn't all of this very hard now that we have lots of pieces to operate?

AWS operational responsibility models



What is serverless?

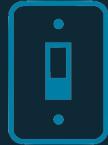


No infrastructure provisioning,
no management



Automatic scaling

Pay for value



Highly available and secure



Serverless is an operational model that spans many different categories of services

COMPUTE



AWS
Lambda



AWS
Fargate

DATA STORES



Amazon
S3



Amazon Aurora
Serverless



Amazon
DynamoDB

INTEGRATION



Amazon
API Gateway



Amazon
SQS



Amazon
SNS



AWS
Step Functions



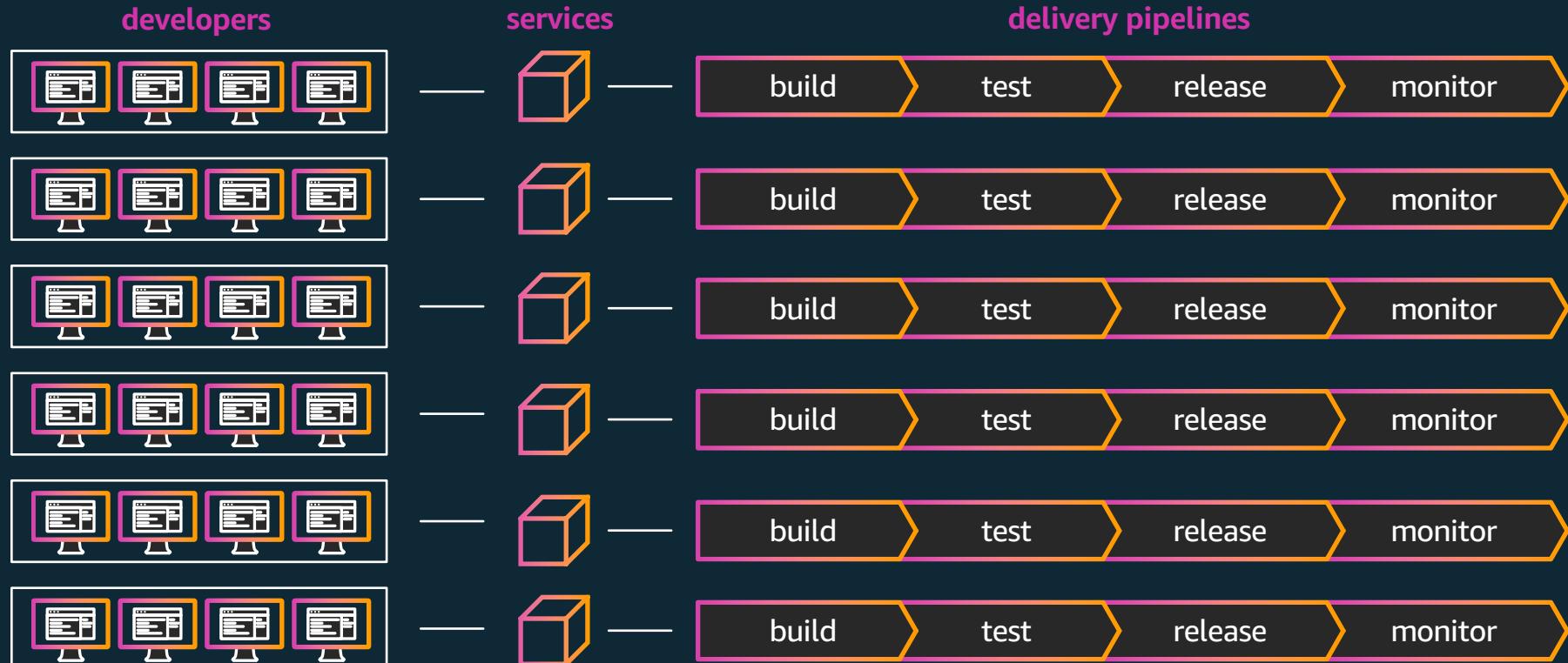
AWS
AppSync

Changes to the delivery of software



How do I develop and deploy code in a serverless microservices architecture?

Microservice development lifecycle



Best practices



Decompose for agility
(microservices, 2 pizza teams)

Automate everything

Standardized tools

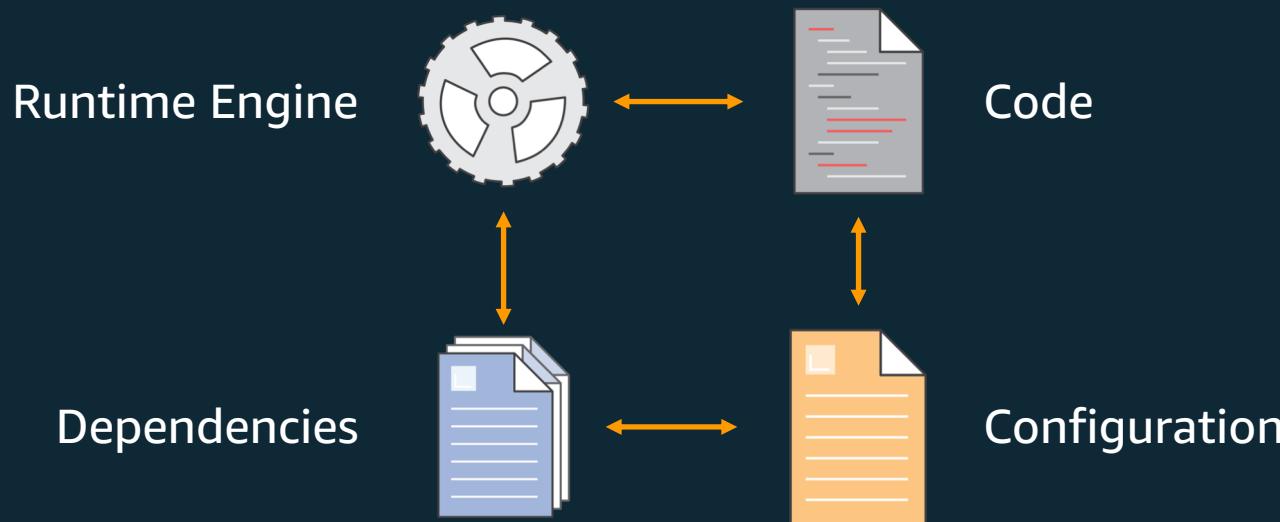
Belts and suspenders
(governance, templates)

Infrastructure as code

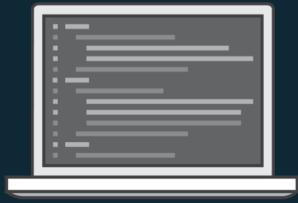


Containers are the best on ramp towards modern applications

Application environment components



Different environments



Local Laptop



Staging / QA

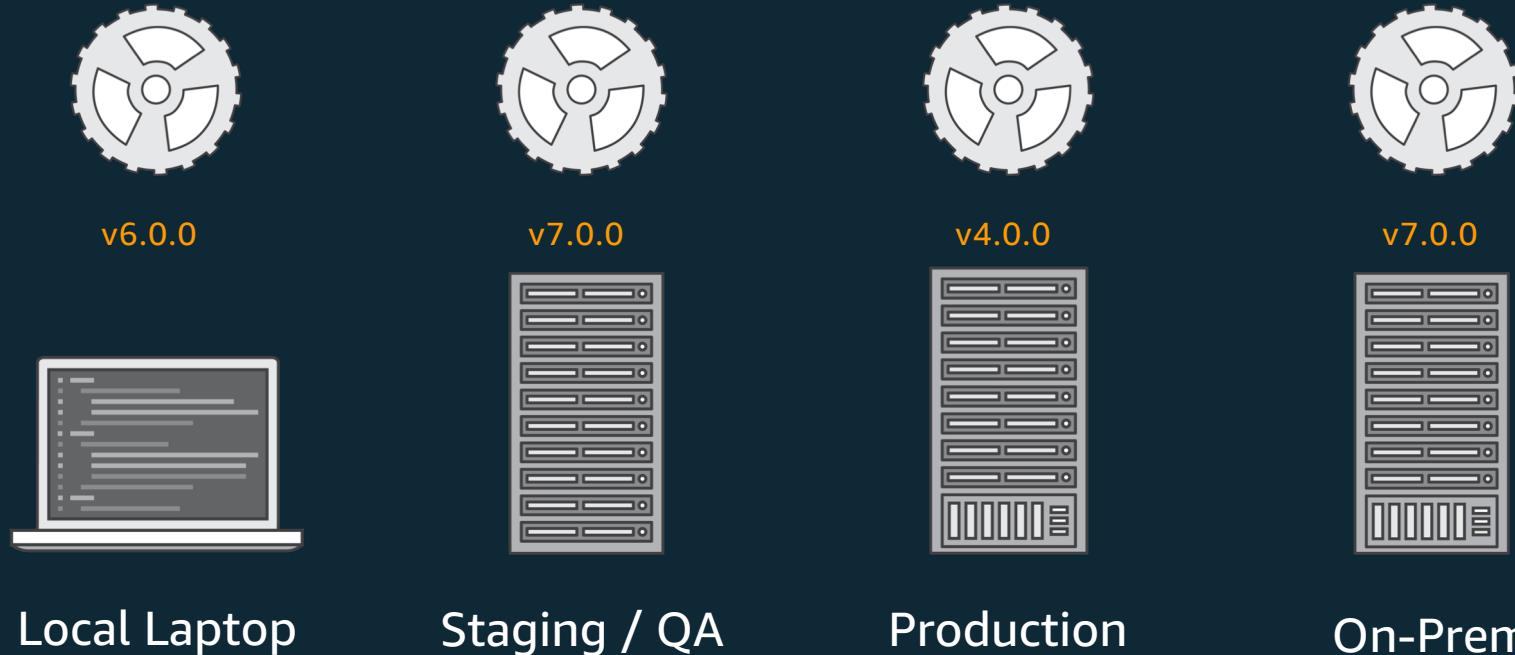


Production



On-Prem

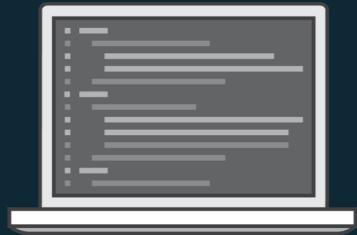
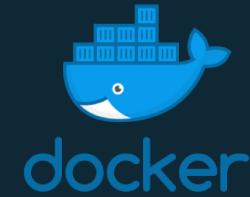
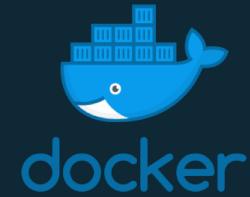
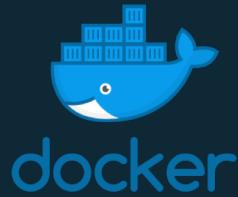
It worked on my machine, why not in prod?



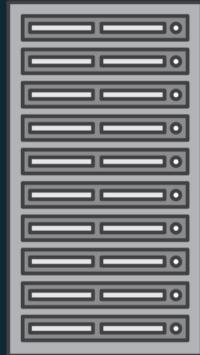
Docker to the rescue



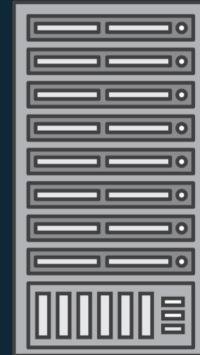
Four environments, same container



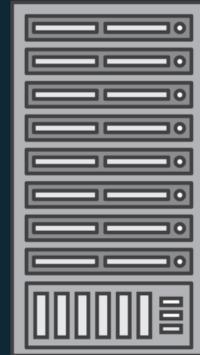
Local Laptop



Staging / QA



Production



On-Prem

The background image shows a busy port terminal with several large blue gantry cranes. One crane has the number '58' on its side. Numerous shipping containers are stacked in the background, with some labeled 'APM TERMINALS'. The sky is clear and blue.

Why are enterprises adopting containers?

- Accelerate software development
- Build modern applications
- Automate operations at web scale

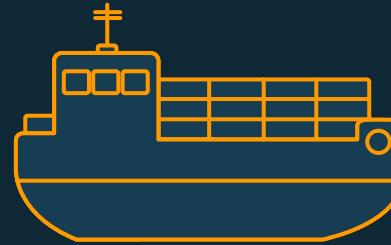
Make AWS the **BEST PLACE** to run **ANY**
containerized applications



Helping customers scale containers



450+%
growth



Hundreds of millions
of containers started each week
of millions
of container instances

Typical use cases

- Microservices: Java, Node.js, Go, Web Apps, etc.
- Continuous Integration and Continuous Deployment (CICD)
- Batch Processing and ETL jobs
- Common PaaS Stack for Application Deployment
- Legacy Application Migration to the Cloud
- Hybrid Workloads
- AI/ML
- Scale Testing
- Backend for IoT use cases

AWS container services landscape

Management

Deployment, Scheduling,
Scaling & Management of
containerized applications



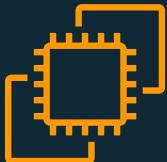
Amazon Elastic
Container Service



Amazon Elastic
Container Service
for Kubernetes

Hosting

Where the containers run



Amazon EC2



AWS Fargate

Image Registry

Container Image Repository



Amazon Elastic
Container Registry



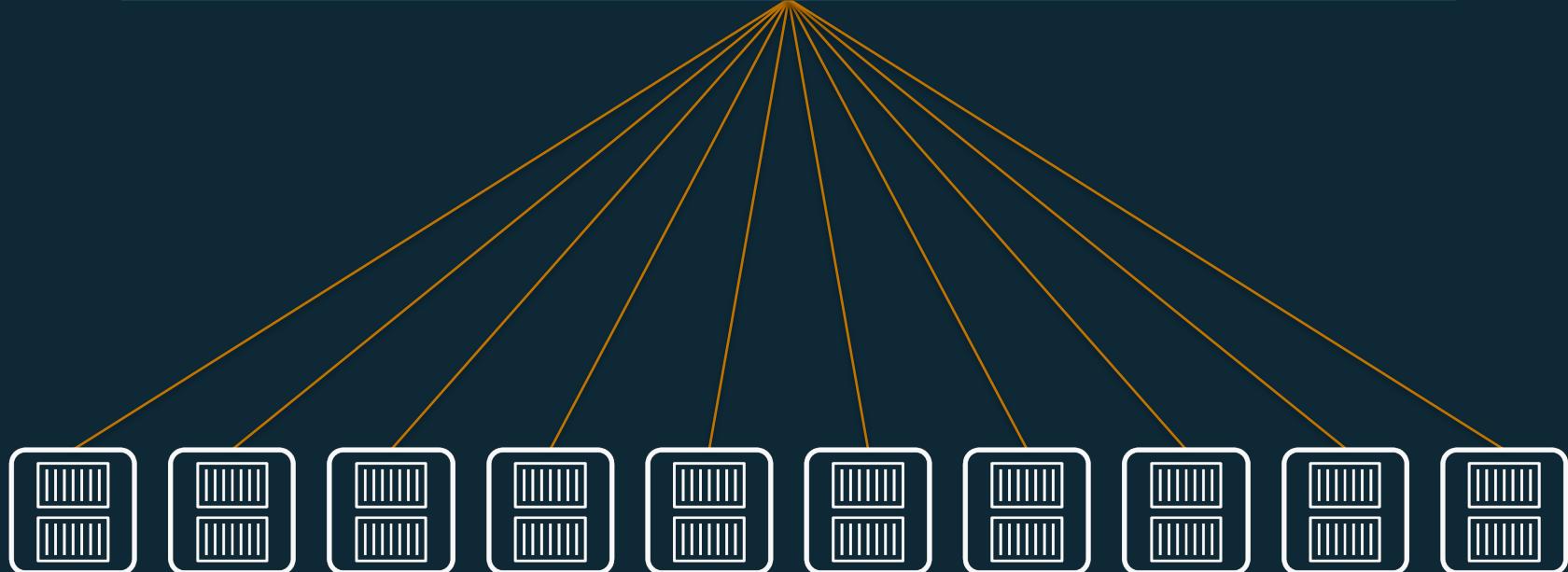
Amazon Elastic Container Service



Scheduling and Orchestration

Cluster Manager

Placement Engine





McDonald's Home Delivery: Why Amazon ECS?



⌚ Speed to market

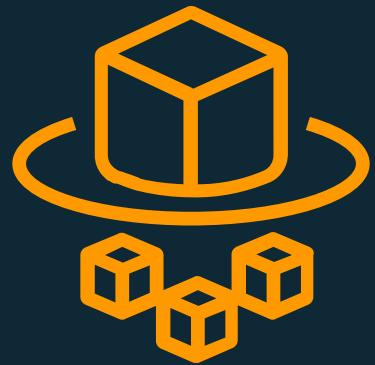
✖ Scalability and reliability

🔒 Security

🔧 DevOps—CI / CD

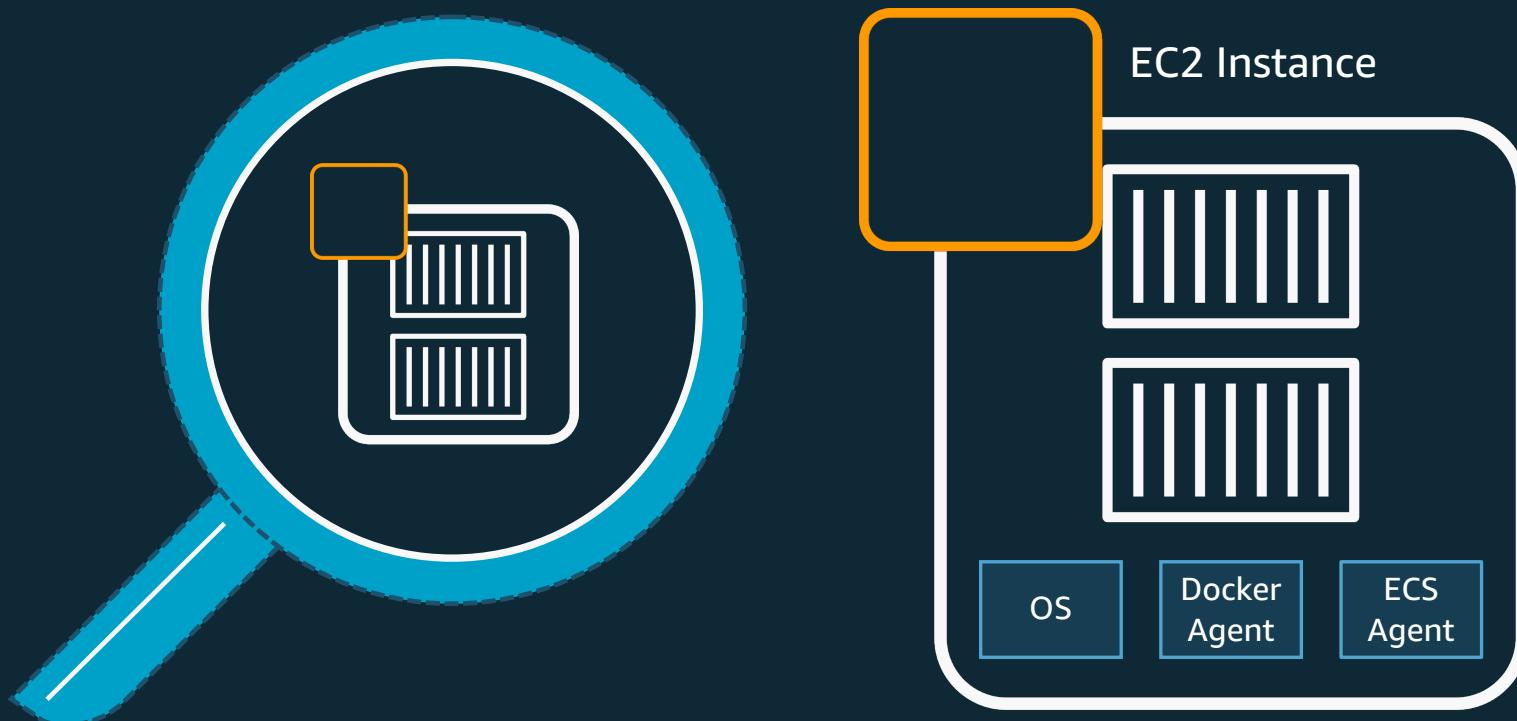
👁 Monitoring



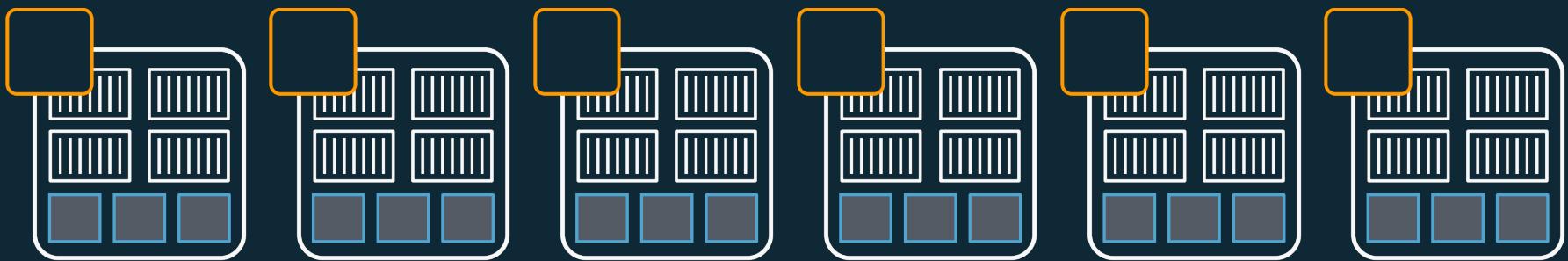


AWS Fargate

Without Fargate, you end up managing more than just containers



- Patching and Upgrading OS, agents, etc.
- Scaling the instance fleet for optimal utilization





Amazon Elastic Container Service





Amazon Elastic Container Service



AWS Fargate



Your
containerized
applications

Managed by AWS

No EC2 Instances to provision, scale or manage

Elastic

Scale up & down seamlessly. Pay only for what you use

Integrated

with the AWS ecosystem: VPC Networking, Elastic Load Balancing, IAM Permissions, CloudWatch and more

Fully managed container environment with AWS ECS + Fargate



Bring existing code

No changes required of existing code, works with existing workflows and microservices built on Amazon ECS



Production ready

ISO, PCI, HIPAA, SOC compliant.
Launch ten or tens of thousands of containers in seconds in 9 global regions (+7 in 2018)



Powerful integrations

Native AWS integrations for networking, security, CICD, monitoring, and tracing

Fargate runs tens of millions of containers for AWS customers every week



migrated ~850 applications running in ~5000 containers
to Fargate to reduce the undifferentiated heavy lifting
that came with managing Kubernetes

AWS Fargate customers

"We moved to **Fargate** because we need the ability to scale quickly up from baseline and get fine-grained network control, without having to manage our own infrastructure"

Product Hunt

"We don't want to babysit any clusters. That has nothing to do with us"

Shimon Tolts
CTO, DATREE



datree.io



• A P T I V •

HARRY'S





Amazon Elastic Container Service for Kubernetes

What is Kubernetes?



Open source container management platform



Helps you run containers at scale



Gives you primitives for building modern applications

Community, contribution, choice



kubernetes

But where you run Kubernetes matters

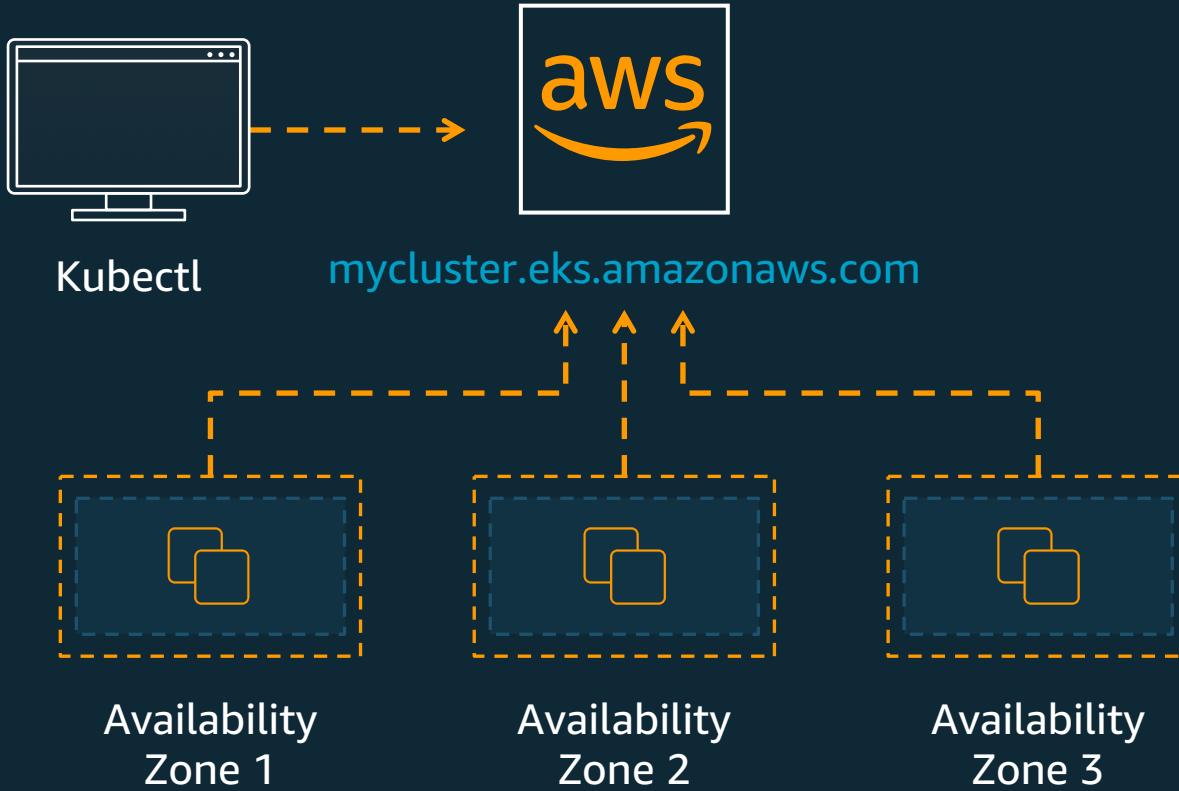




51%

of Kubernetes workloads
run on AWS today

—CNCF survey



EKS is Kubernetes certified



How are customer using Amazon EKS?



Microservices



Platform-as-a-Service



Enterprise App
Migration



Machine Learning

Customers adopting Kubernetes on AWS



Customer example: Snap



100% on Kubernetes in the cloud

Moving core messaging architecture to AWS and EKS

Currently monolithic, breaking this into SOA and microservices

“Undifferentiated Heavy Lifting is work that we have to do that doesn’t directly benefit our customers. It’s just work. EKS frees us up to worry about delivering customer value and allows developers without operational experience to innovate without having to know where their code runs.”

[More detailed talk: AWS New York Summit 2018 - Run Kubernetes with Amazon EKS \(SRV318\)](#)

Amazon container services



Amazon ECS



Amazon EKS



AWS Fargate



Amazon ECR

THANK YOU! Questions?

<https://aws.amazon.com/containers>

 jlanger@amazon.de

 @johlanger

Rich partner ecosystem

Foundation



DevOps



Monitoring & Logging



Security



Networking



New: AWS Cloud Map



AWS
Cloud
Map

Service discovery for all your cloud resources

Constantly monitor the health of every resource

Dynamically update the location of each microservice

Increase developer productivity

Single registry for all app resources

Define resources with user-friendly names

Integration with Amazon container services

AWS Fargate

Amazon ECS

Amazon EKS

New: AWS App Mesh



Observability & traffic control

Easily export logs, metrics, and traces

Client side traffic policies—circuit breaking, retries

Routes for deployments

Works across clusters and container services

Amazon ECS

Amazon EKS

Kubernetes on EC2

AWS Fargate (coming soon!)

AWS built and run

No control plane to manage

Ease of operations

High scale

Container Customers

EKS Reference Customers:

Fidelity Investments
SNAP Inc
Appcard

EKS Reference Customer: Fidelity Investments



"We built the next generation of our PaaS using EKS for large enterprise workloads. We manage thousands of applications and have hundreds of DevOps teams."

Amr Abdelhalem, Head of Cloud Architecture

EKS Reference Customer: SNAP



"Snapchat serves millions of people around the world every day, and we're thrilled to now leverage Amazon EKS as a core compute service that can meet our needs now, as well as upcoming plans to host several critical workloads in the coming months."

Alex Strand, Senior Director of Engineering, Snap Inc

EKS Reference Customer: Appcard



"Kubernetes is fast becoming the preferred solution for container orchestration. Its biggest downside is that it is not simple to set up and operate. EKS gives us all the benefits of Kubernetes, but takes care of managing the hard stuff. We can dedicate less resources to deployment and operations as result."

Amichay Oren, Co-founder & CTO, AppCard Inc

Fargate Reference Customers:

Turner Broadcasting
99Designs
Harry's Razors

Fargate Reference Customer: Turner Broadcasting



"The Cloud Architecture team began building tooling around Fargate to accelerate the adoption and the move to this new DevOps world. The result ended up with reduced cost and time."

**Joseph Bulger, Principal Architect
Turner Broadcasting System**

Fargate Reference Customer: 99Designs



"We moved to Fargate to reduce operational burden and operational costs. Fargate made running Docker containers easy, removing need to maintain instances."

Robert McNeil, Sr. Engineer, 99designs

Fargate Reference Customer: Harry's Razors

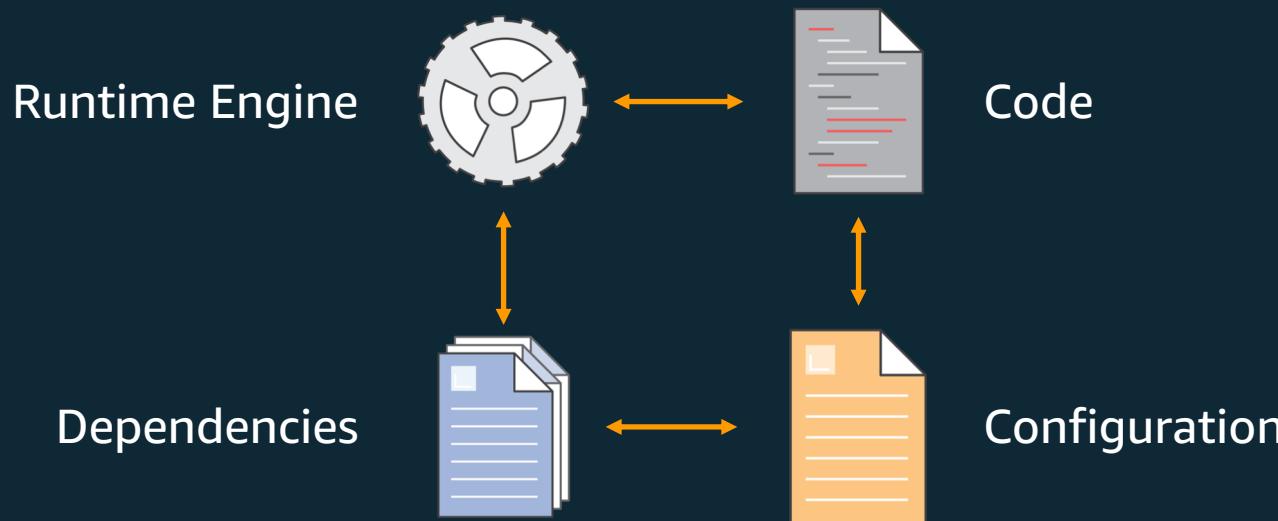
HARRY'S

"With Fargate we eliminated EC2 instances, sizing concerns, instance profiles and policies. Directly leveraging service auto scaling and target tracking policies, migrating without any downtime and simplifying our overall system."

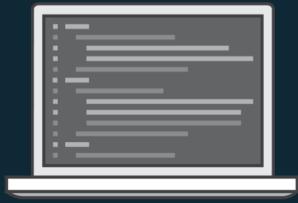
Bryce Lohr, Technical Lead, Core Services, Harry's

Introduction to Containers and Docker

Application environment components



Different environments



Local Laptop



Staging / QA

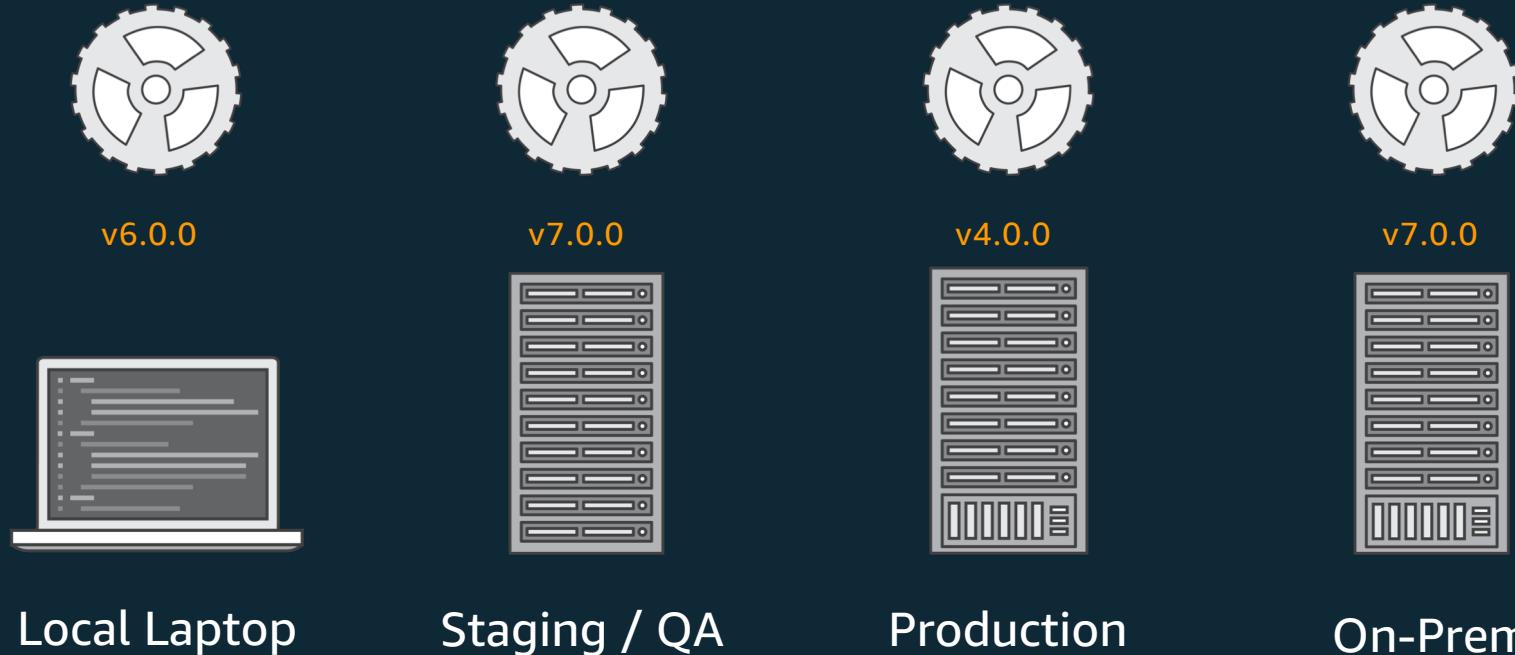


Production

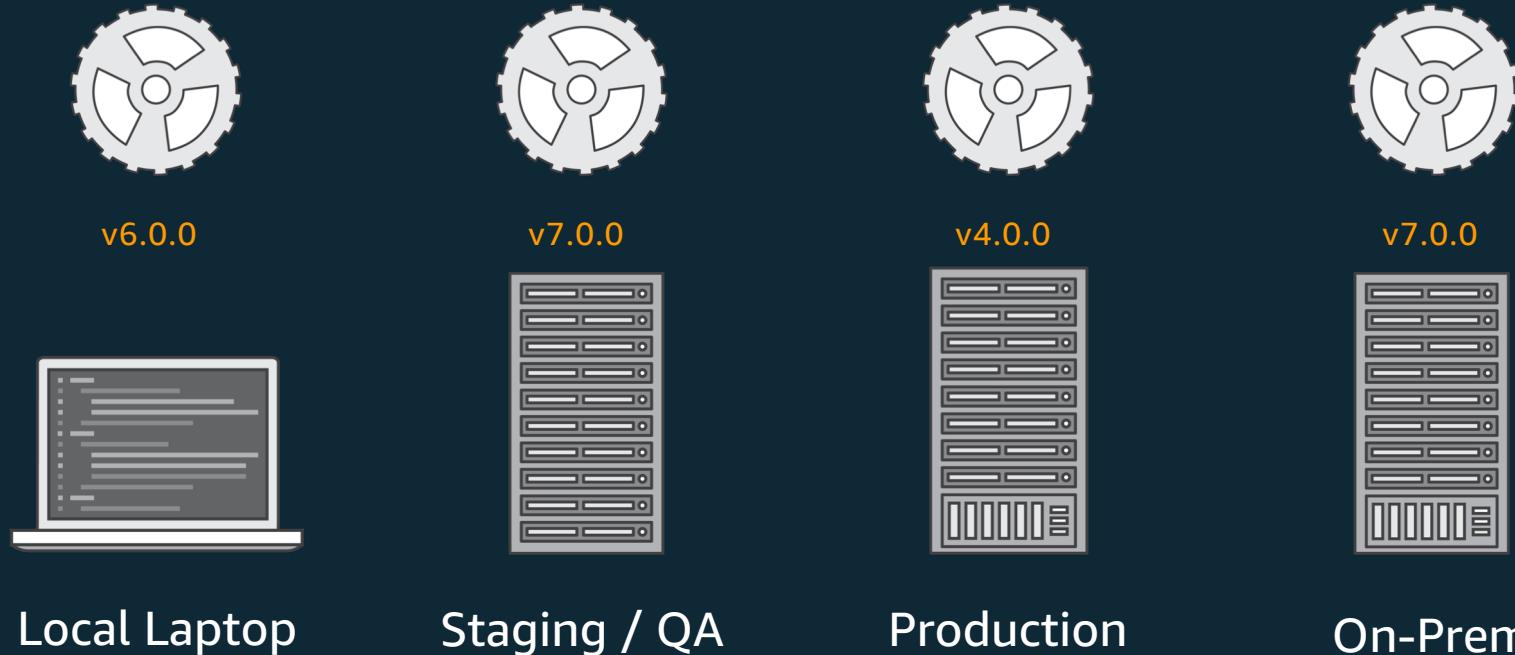


On-Prem

It worked on my machine, why not in prod?



It worked on my machine, why not in prod?



Docker to the rescue

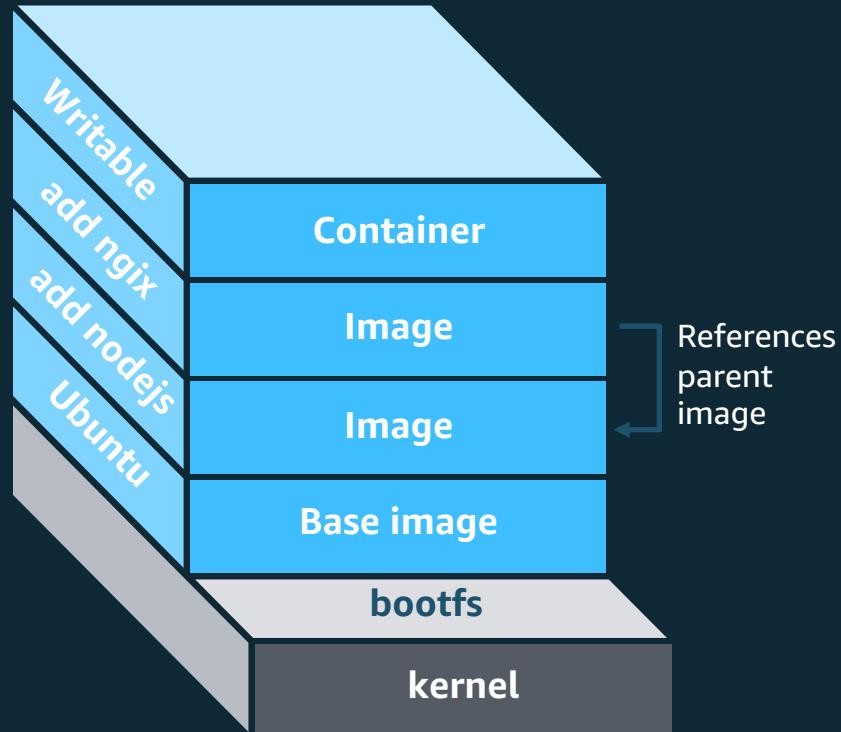


Docker container image

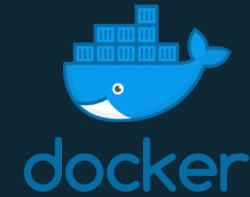
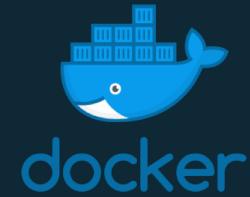
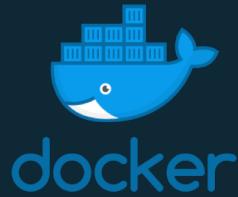
Read only image that is used as a template to launch a container.

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

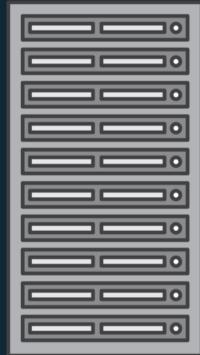
Docker file for easy, reproducible builds.



Four environments, same container



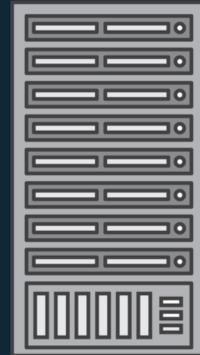
Local Laptop



Staging / QA

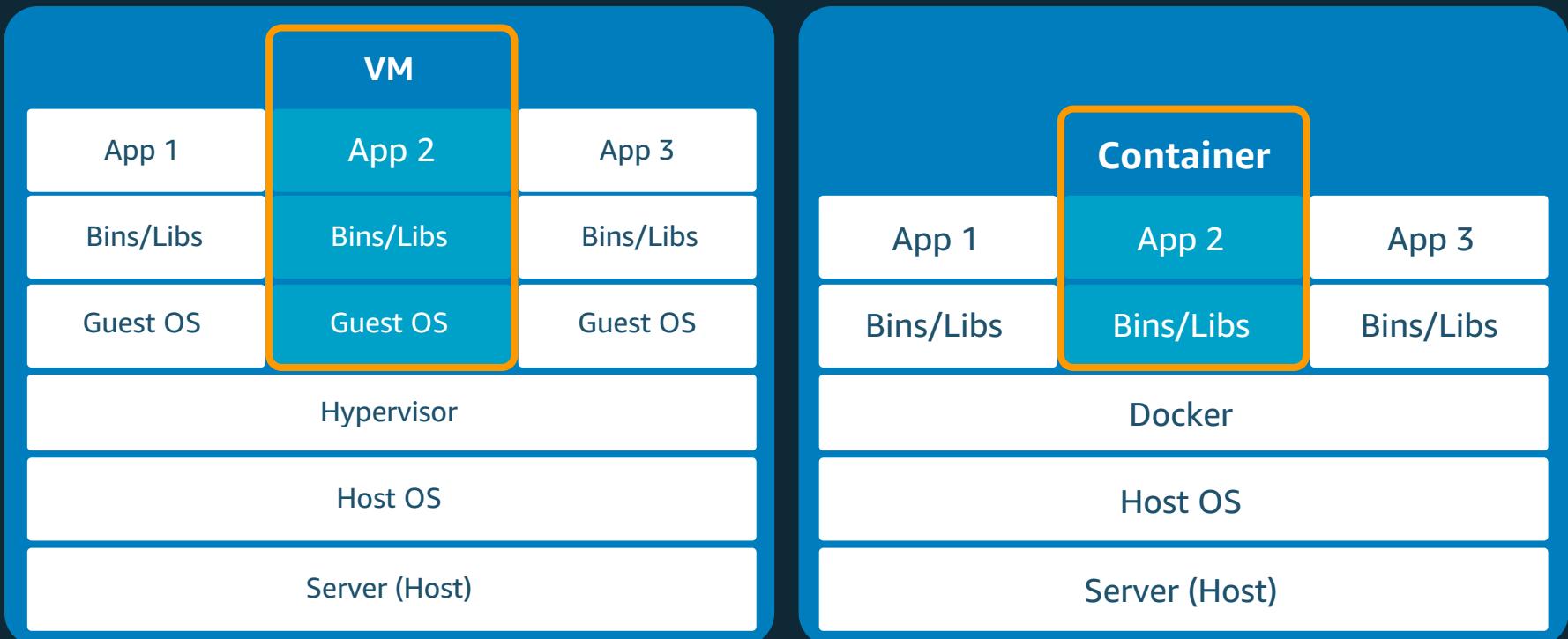


Production



On-Prem

Virtual machine versus Docker



Container & Docker Benefits

Portable application artifact that runs reliably everywhere

Run different applications or application versions with different dependencies simultaneously

Better resource utilization by running multiple lightweight containers per host

Reference Architectures (Informational)

Amazon ECS:

[Rebalancing Amazon ECS Tasks using AWS Lambda](#)

[NGINX Reverse Proxy on Amazon EC2 Container Service](#)

[Java Microservices Deployed on EC2 Container Service](#)

[Amazon ECS Reference Architecture: Batch Processing](#)

[Node.js Microservices Deployed on EC2 Container Service](#)

[Amazon EC2 Container Service - Reference Architecture: Service Discovery to containers using CloudWatch Events, Lambda and Route 53 private hosted zones](#)

[Service Discovery for AWS EC2 Container Service via DNS](#)

[Canary Blue/Green deployments on ECS](#)

[Blue/Green deployments on ECS](#)

[ECS Reference Architecture: Continuous Deployment](#)

[Amazon ECS Scheduler Driver to integrate Apache Mesos with ECS](#)

AWS Fargate

[Blue/Green deployments using Fargate](#)

[How to host an ASP.NET core application in AWS Fargate using Linux containers](#)

Amazon EKS

[CodeSuite - Continuous Deployment Reference Architecture for Kubernetes](#)