



AWS Containers

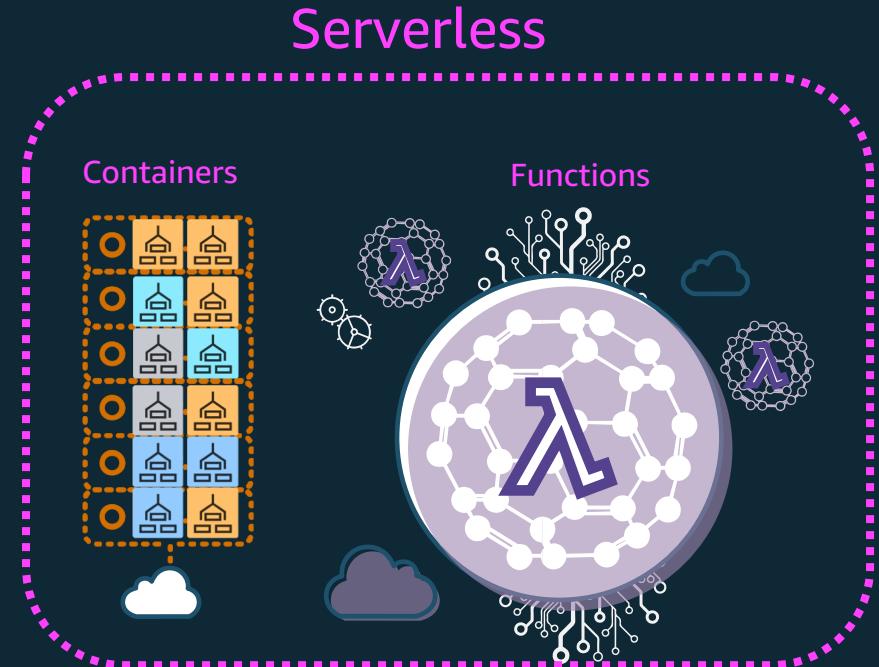
Introduction to ECS, EKS, ECR, Fargate

Claudiu Farcas, AWS Solutions Architect

September 2019



Evolution of computing



What is serverless?

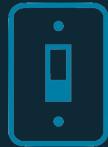


No infrastructure provisioning,
no management



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

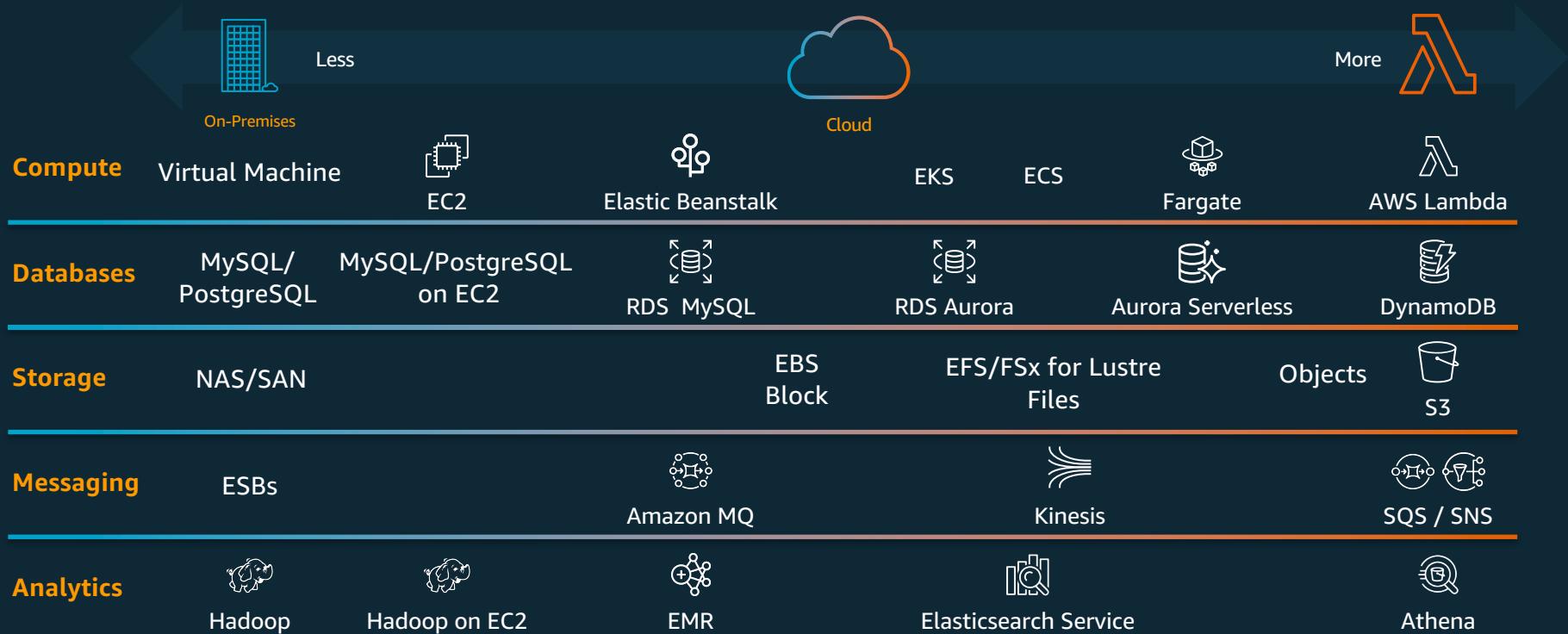


AWS
Step Functions



AWS
AppSync

AWS operational responsibility models



Container-driven Changes

Architectural patterns
Operational model
Software delivery

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



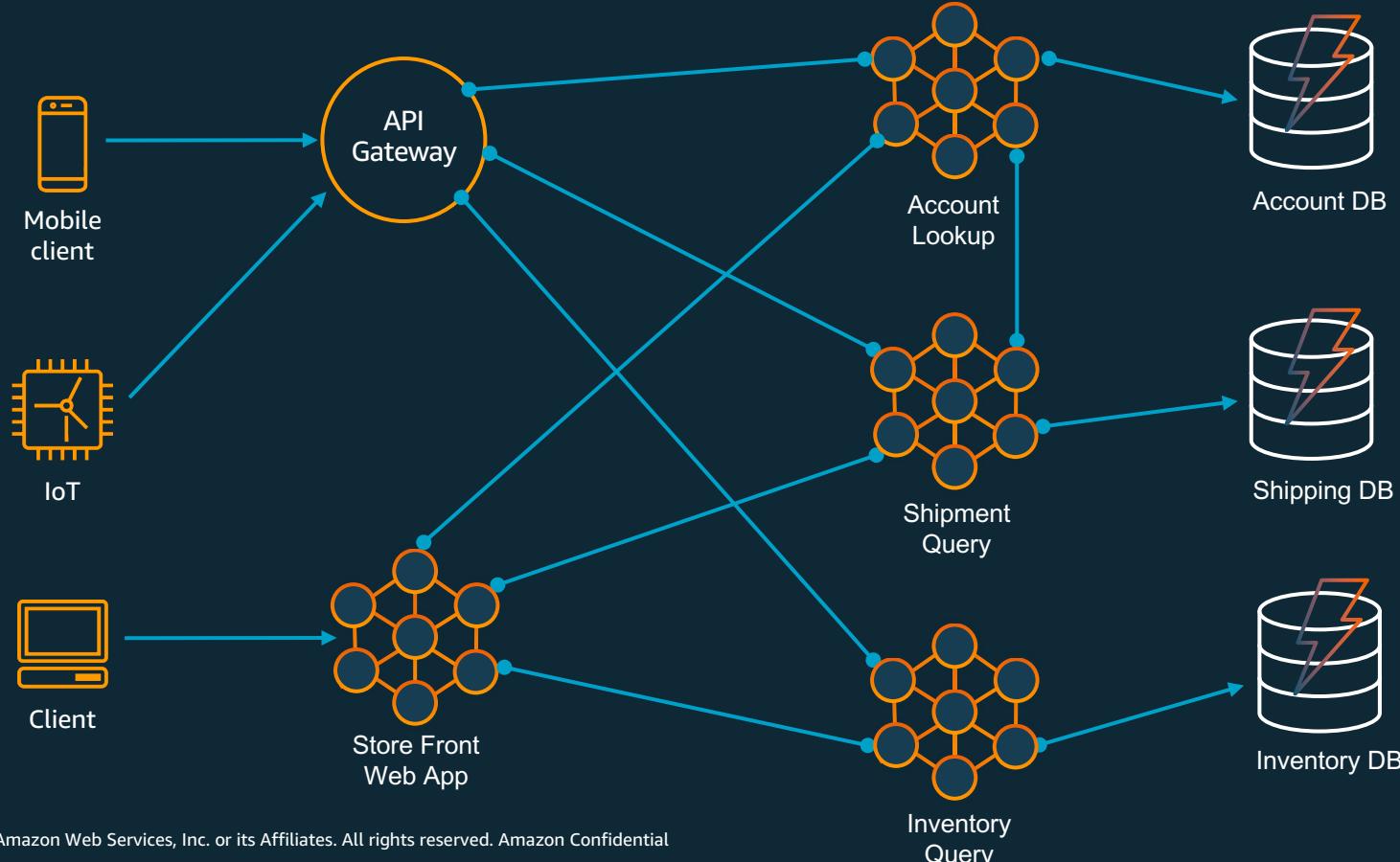
Monolith
Does everything



Microservices
Do one thing

Architectural Pattern: Cloud-native architectures have small pieces, loosely joined

Typical Microservices Architecture



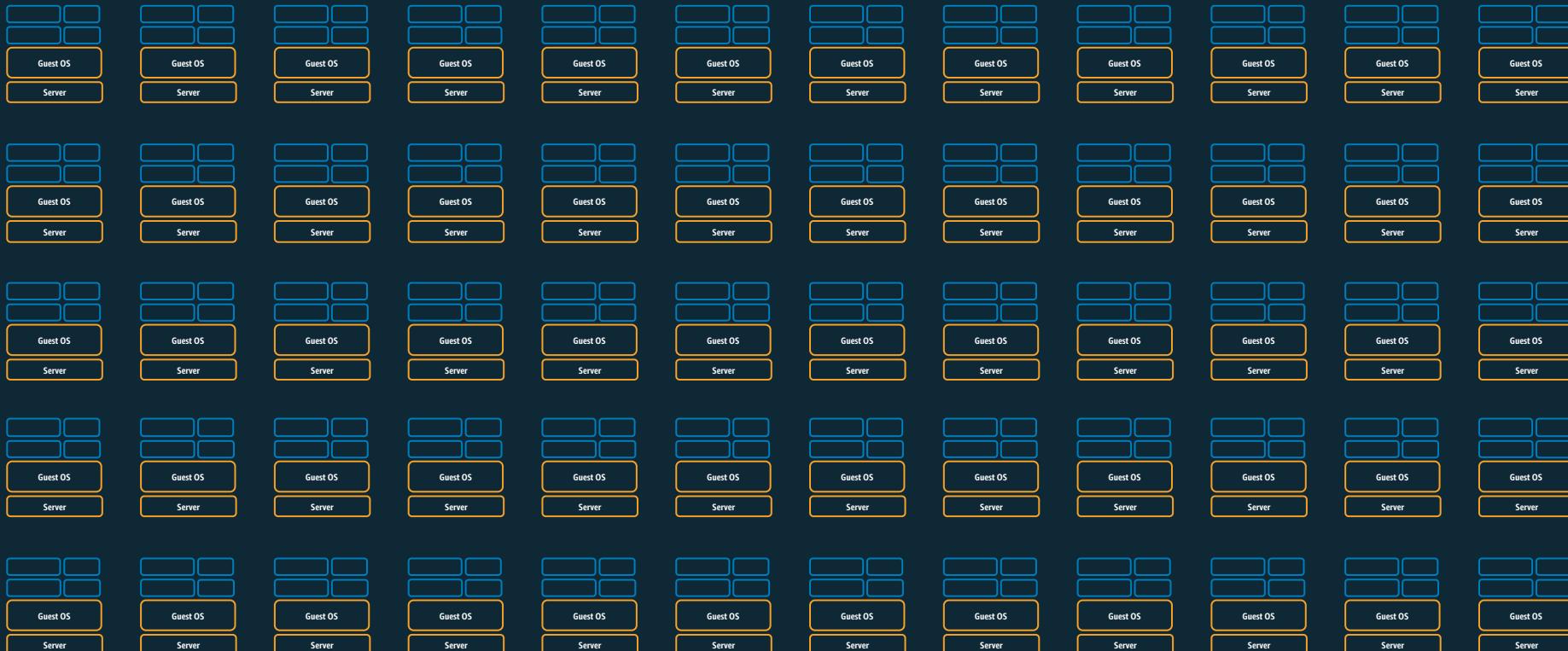
Typical use cases

- Microservices: Java, Node.js, Go, Web Apps, etc.
- Continuous Integration and Continuous Deployment (CI/CD)
- 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

Managing one container is easy



Managing multiple containers is much harder



AWS container services landscape

Management

Deployment, Scheduling,
Scaling & Management of
containerized applications



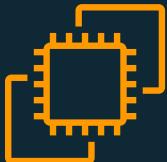
Amazon Elastic
Container Service



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

Amazon ECS



ECS

Highly scalable, high-performance container management system

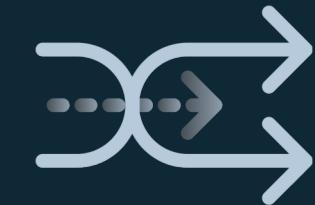
A managed platform



Cluster management



Container orchestration



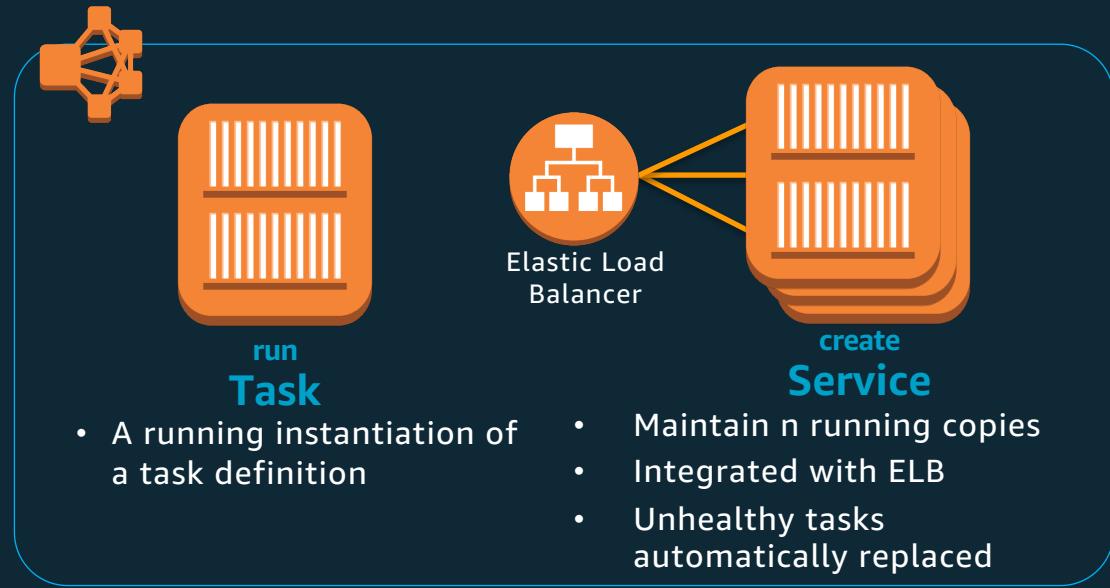
Deep AWS integration

Terminology



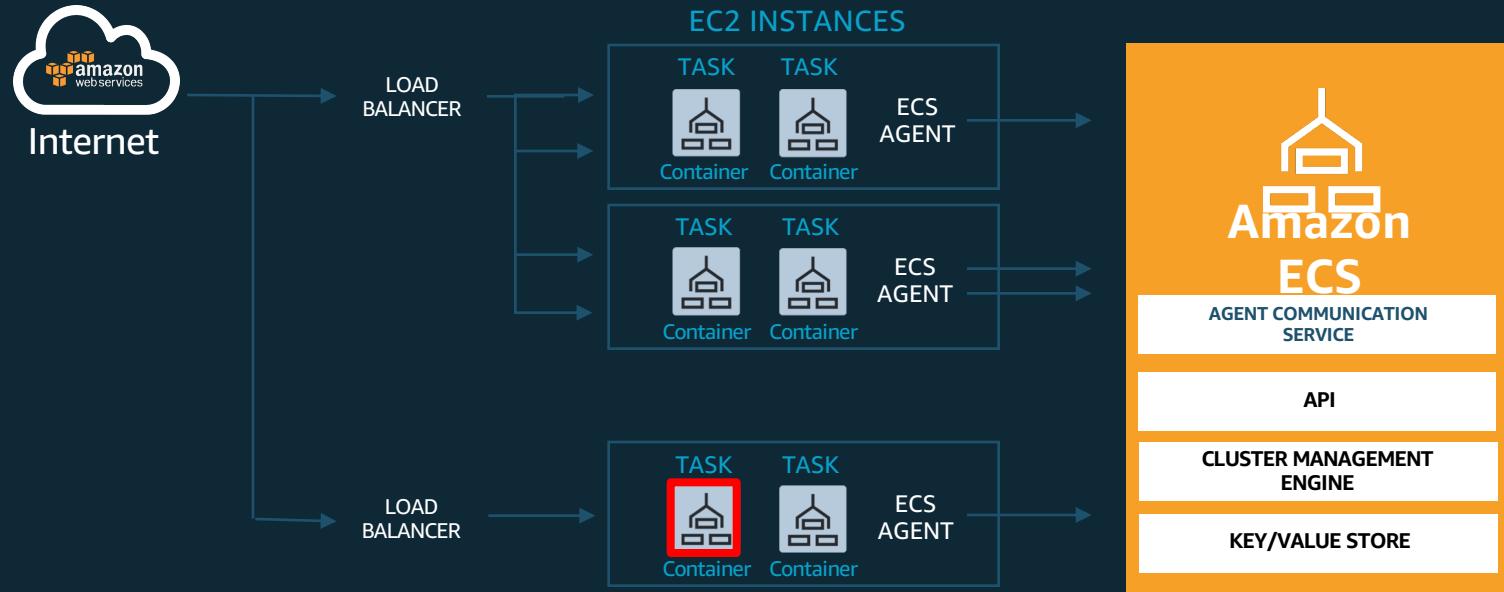
register **Task Definition**

Define application containers:
Image URL, CPU & Memory
requirements, etc.

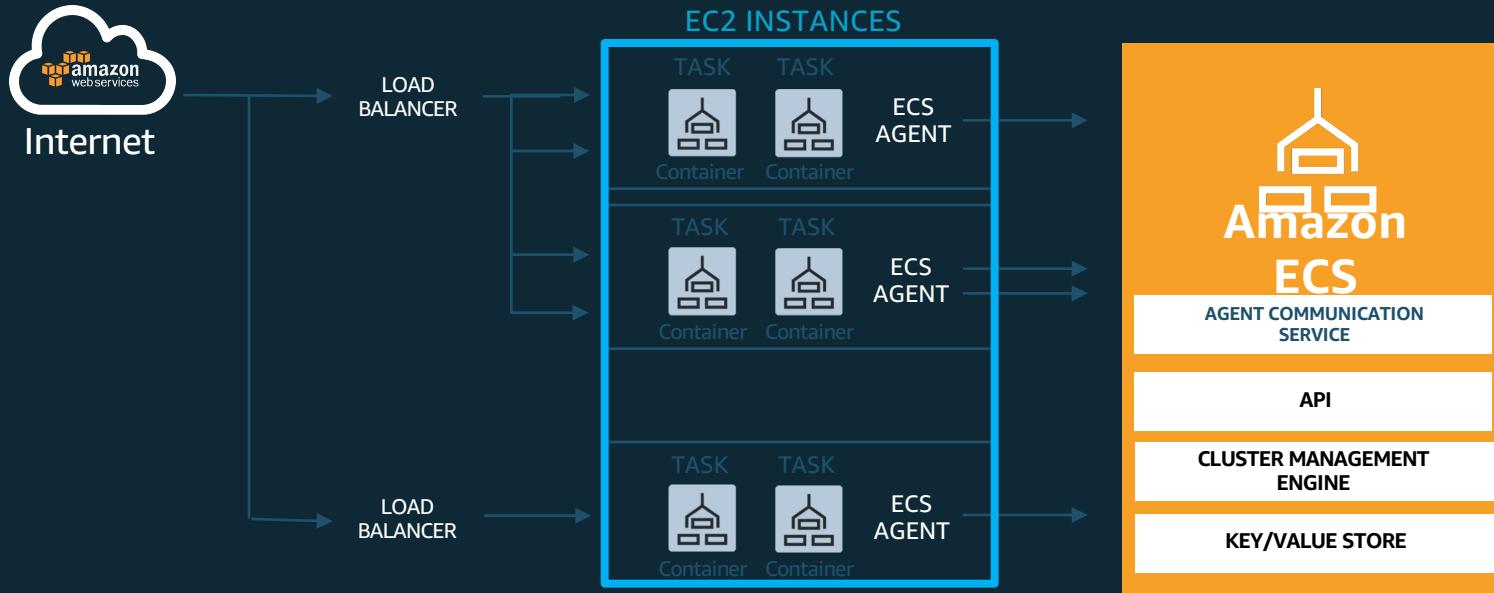


- A running instantiation of a task definition
 - Maintain n running copies
 - Integrated with ELB
 - Unhealthy tasks automatically replaced
-
- Infrastructure Isolation boundary
 - IAM Permissions boundary

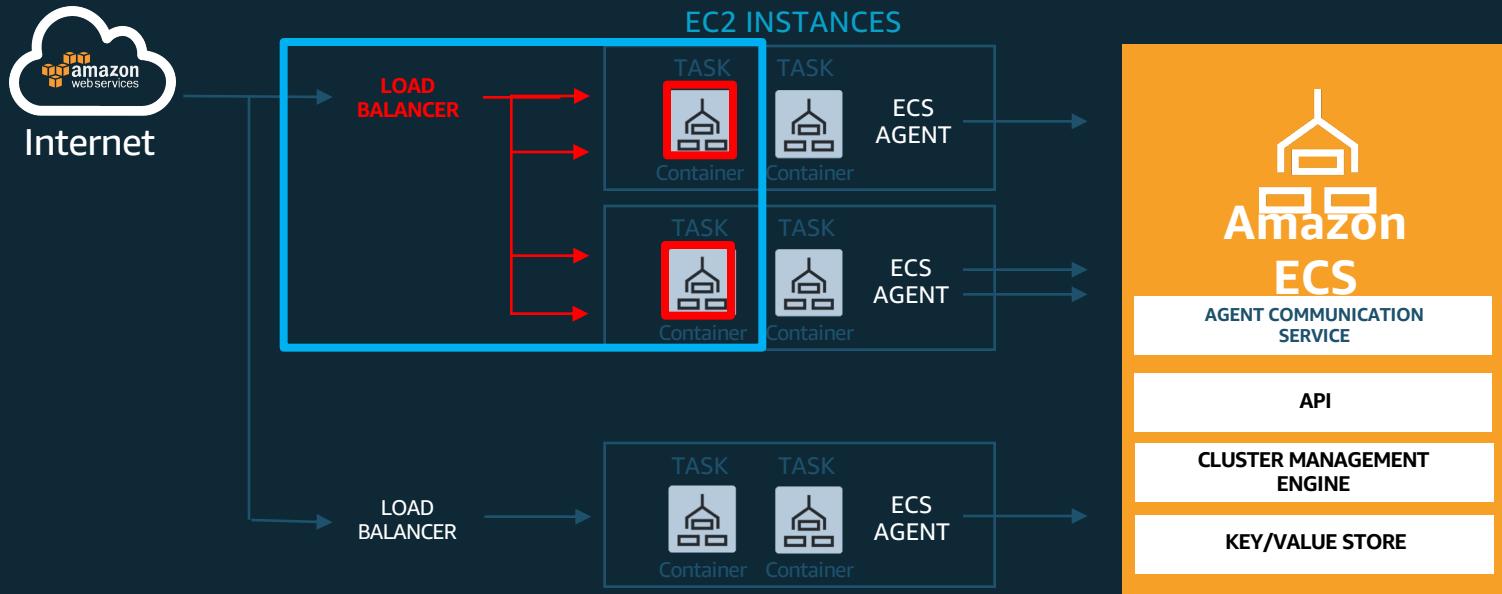
Amazon ECS - Task



Amazon ECS - Cluster



Amazon ECS - Service

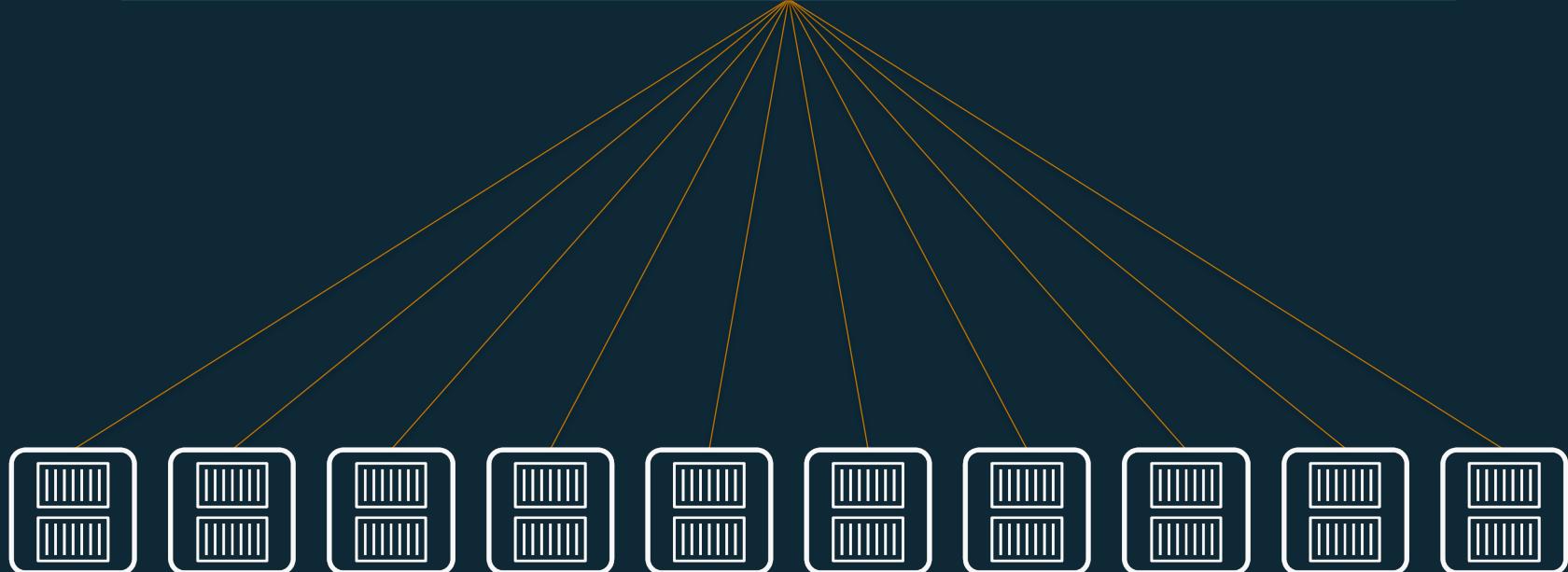




Scheduling and Orchestration

Cluster Manager

Placement Engine



Task definition

Task definition snippet

```
{  
  "family": "mytask",  
  "containerDefinitions": [  
    {  
      "name": "container1",  
      "image": "..."  
    },  
    {  
      "name": "container2",  
      "image": "..."  
    }  
  ]  
}
```

- Immutable, versioned document
- Identified by family:version
- Contains a list of up to 10 container definitions
- All containers will be collocated on the same host
- Each container definition has
 - A name
 - Image URL (Amazon ECR or public images)
 - And more

CPU & memory specification

Units

- CPU: vCPU (string) or CPU units (integer)
(1 vCPU = 256 CPU units)
- Memory: MB (integer) or string (1 GB)

Task-level resources

Total CPU/memory across all containers

Container-level resources

Defines sharing of task resources among containers

Task definition snippet

```
{  
  "family": "mytask",  
  "cpu": "1 vCPU",  
  "memory": "2 GB",  
  "containerDefinitions": [  
    {  
      "name": "container1",  
      "image": "...",  
      "cpu": 256,  
      "memoryReservation": 512  
    },  
    {  
      "name": "container2",  
      "image": "...",  
      "cpu": 768,  
      "memoryReservation": 512  
    }  
  ]  
}
```

Task-level resources

Container-level resources



Amazon Elastic Container Service for Kubernetes

Community, contribution, choice



**CLOUD NATIVE
COMPUTING FOUNDATION**



kubernetes



51%

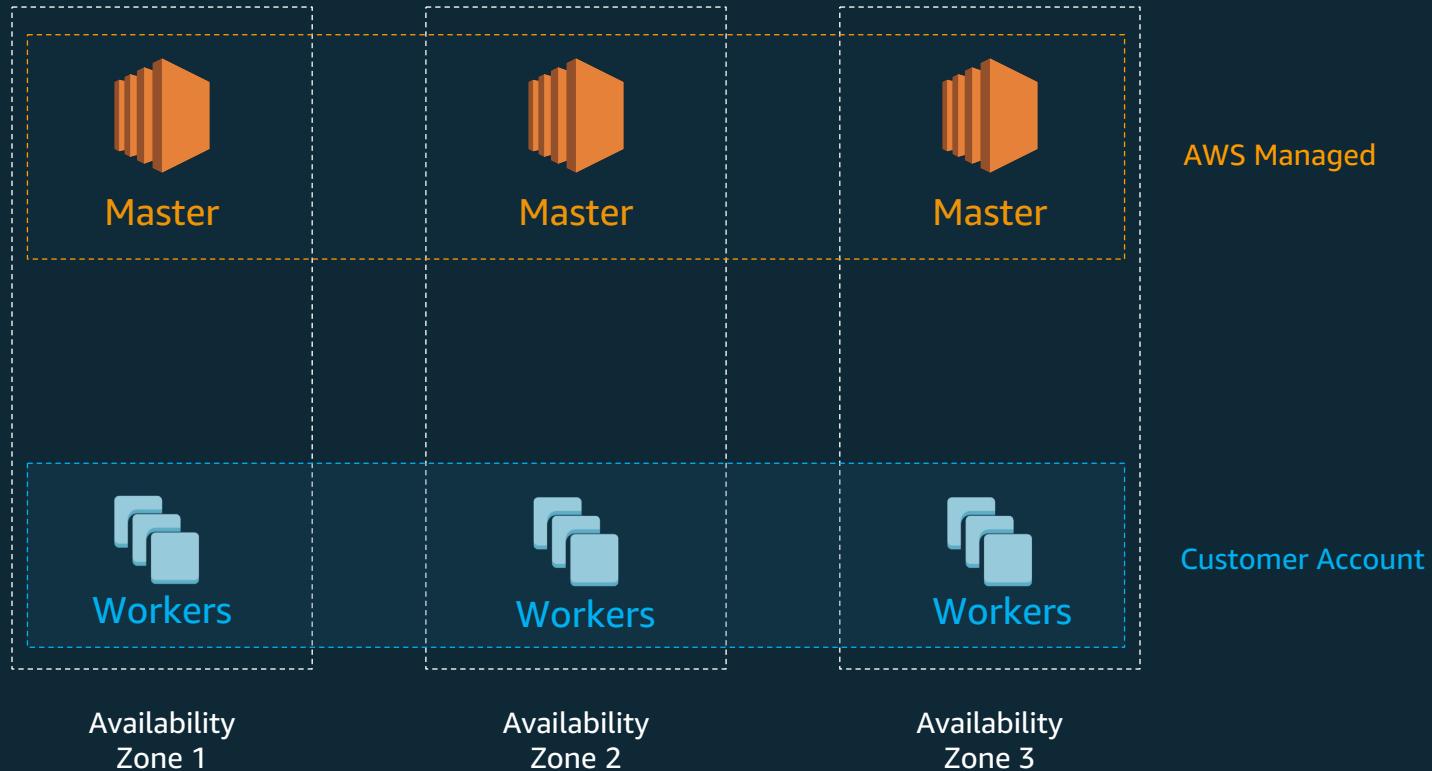
of Kubernetes workloads
run on AWS today

—CNCF survey

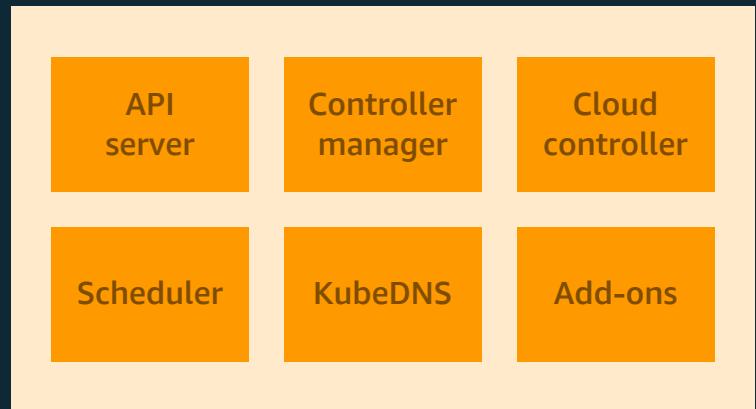
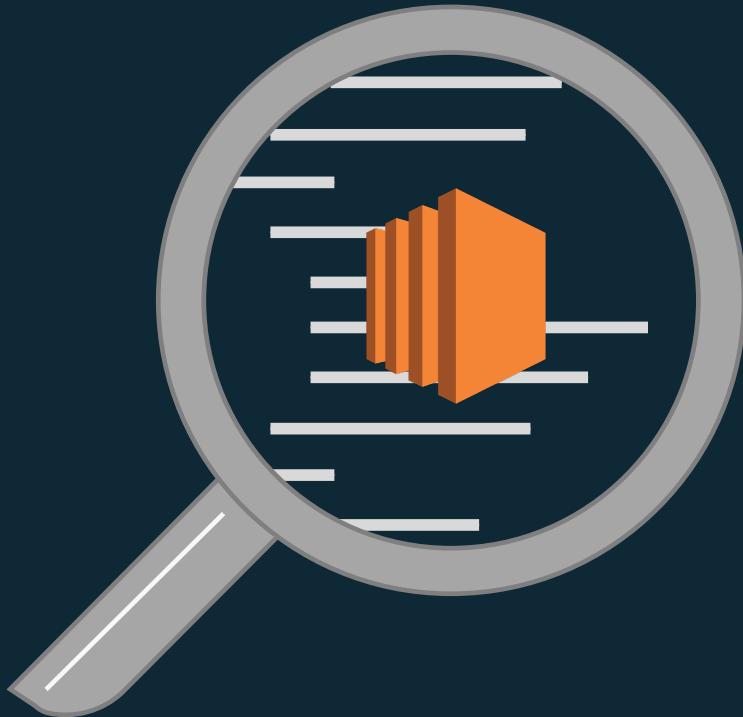
Customers adopting Kubernetes on AWS

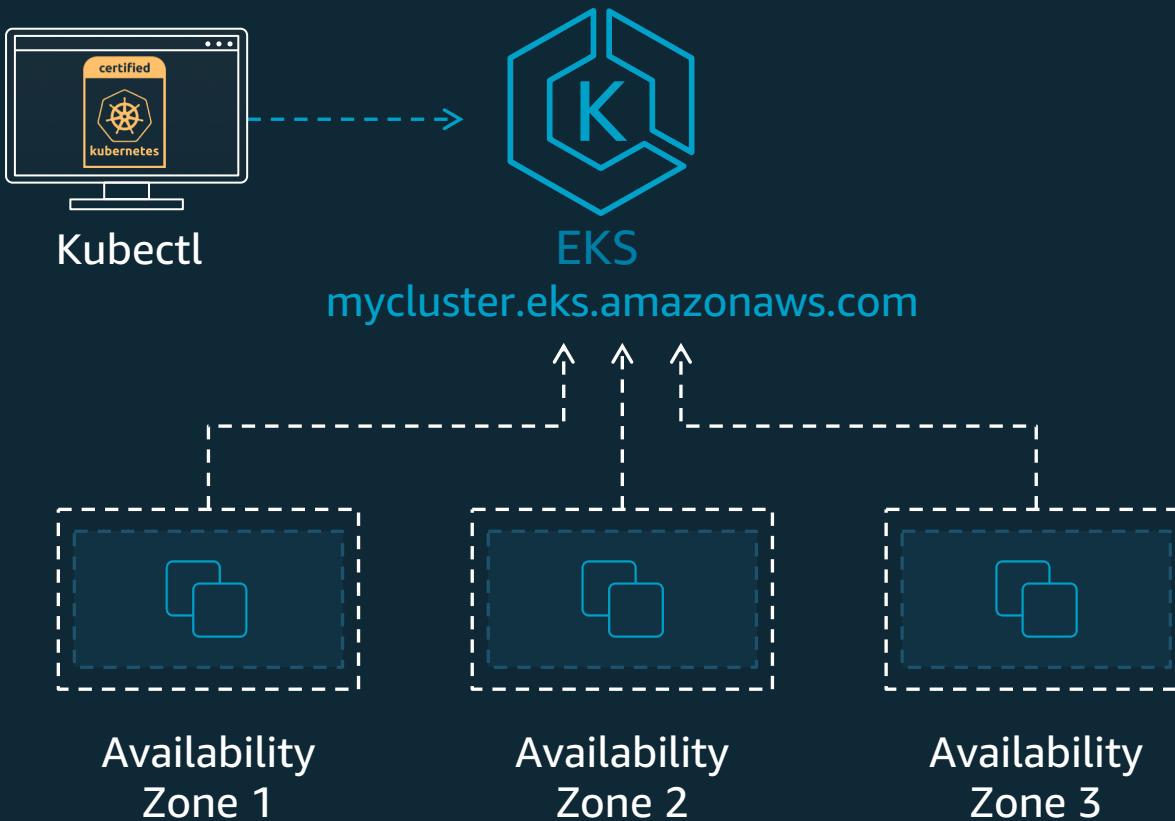


Control vs Data Plane



Kubernetes master 3X





How are customer using Amazon EKS?



Microservices



Platform-as-a-Service



Enterprise App
Migration

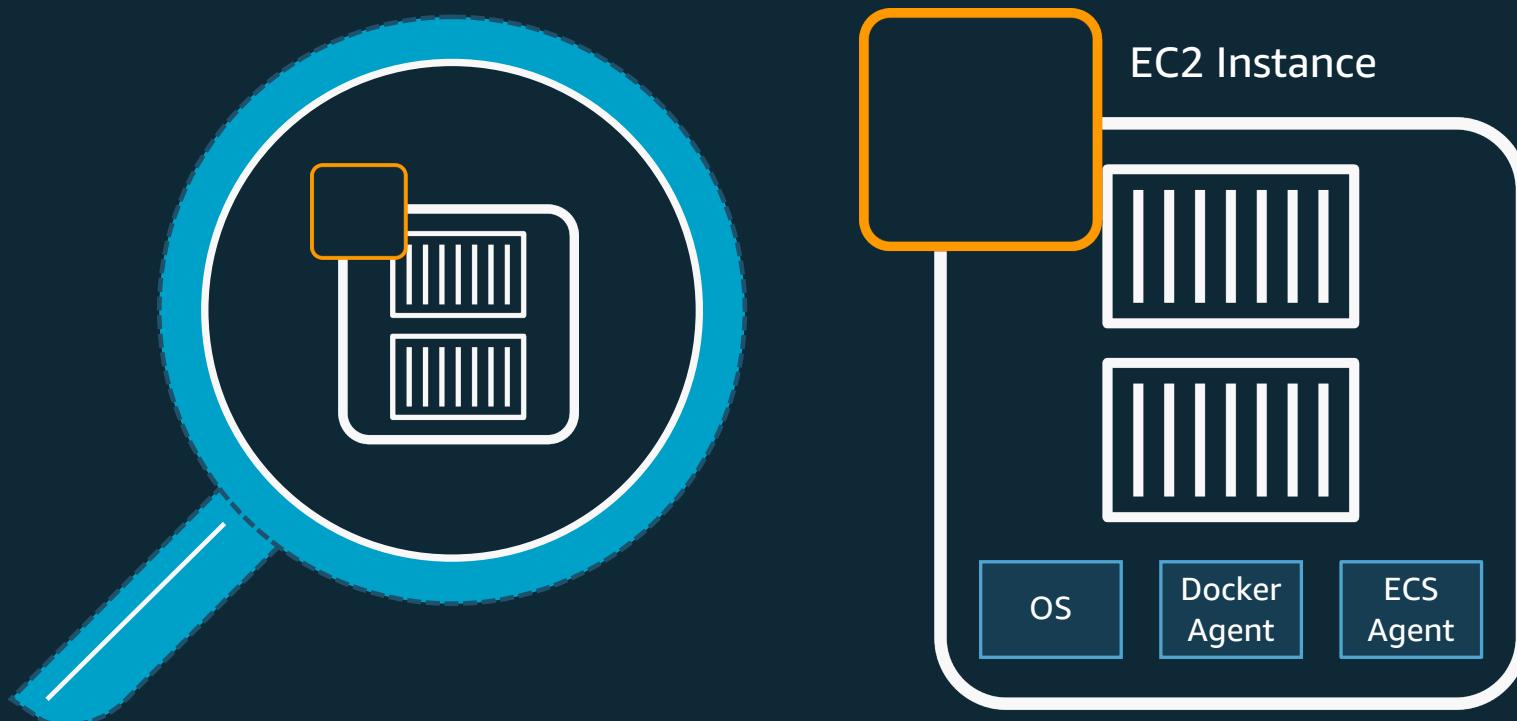


Machine Learning

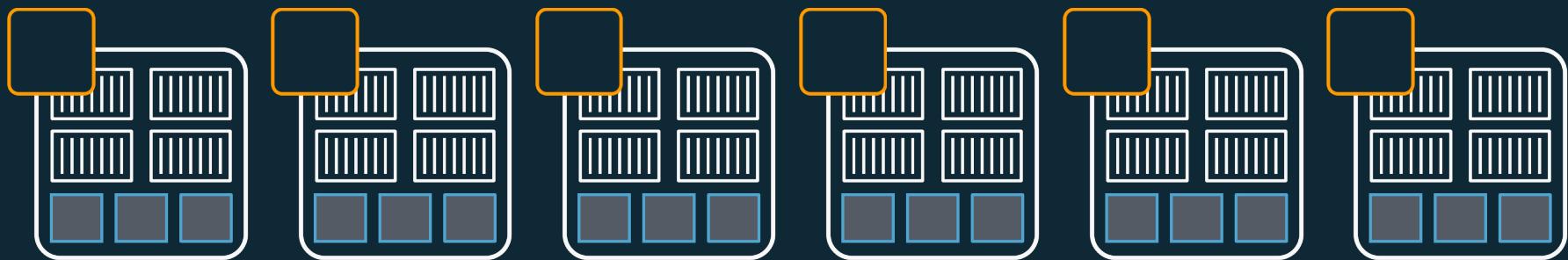


AWS Fargate

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

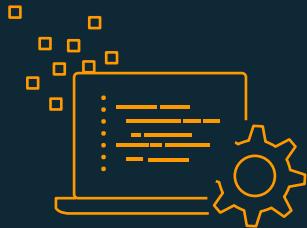
Elastic

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

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



Comparison of operational responsibility

More opinionated



AWS manages

- Data source integrations
- Physical hardware, software, networking, and facilities
- Provisioning



- Container orchestration, provisioning
- Cluster scaling
- Physical hardware, host OS/kernel, networking, and facilities



- Container orchestration control plane
- Physical hardware software, networking, and facilities



- Physical hardware software, networking, and facilities

Customer manages

- Application code
- Data source integrations
- Security config and updates, network config, management tasks
- Application code
- Data source integrations
- Work clusters
- Security config and updates, network config, firewall, management tasks
- Application code
- Data source integrations
- Scaling
- Security config and updates, network config, management tasks
- Provisioning, managing scaling and patching of servers

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



operational model



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!)

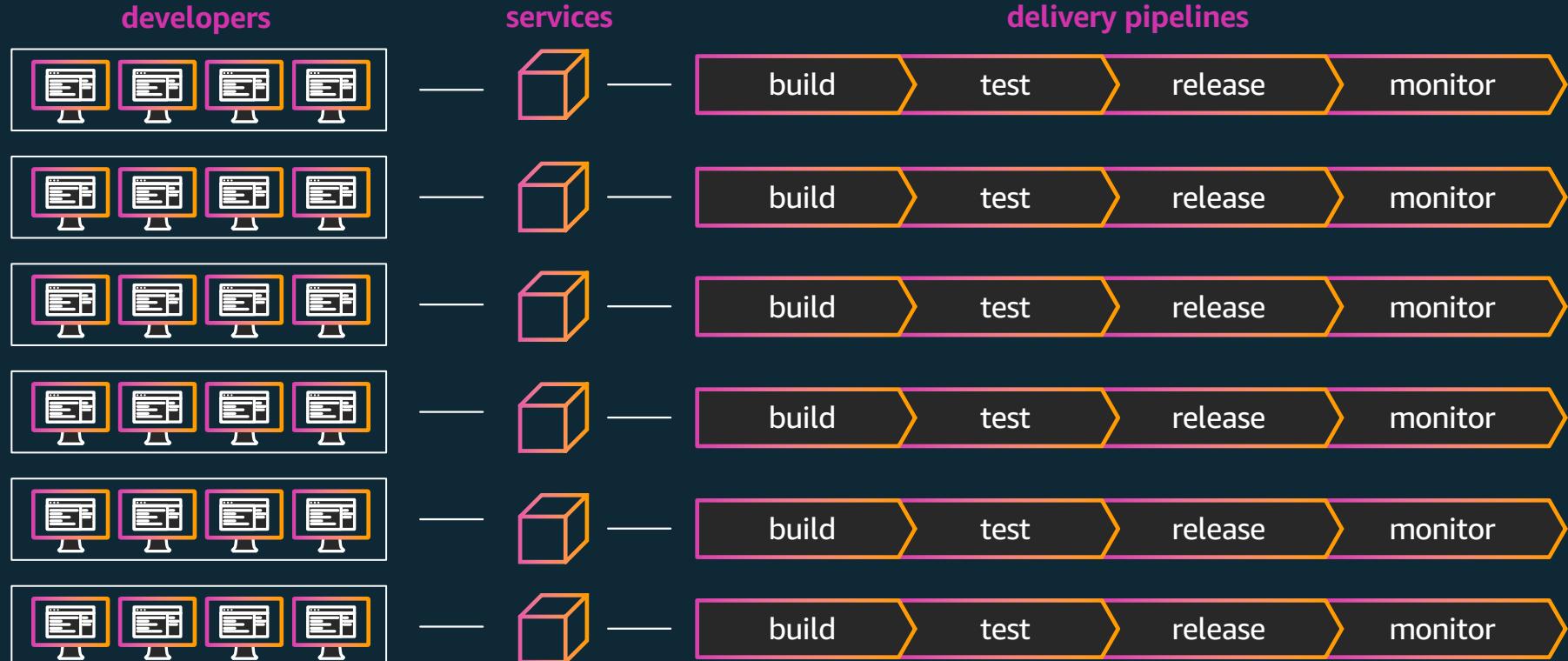
AWS built and run

No control plane to manage

Ease of operations

High scale

Microservice development lifecycle





Amazon Elastic Container Registry

What is Amazon EC2 Container Registry (ECR)?



Fully Managed

Compatible with Docker
Registry v2 API



Secure

Fine grained access control



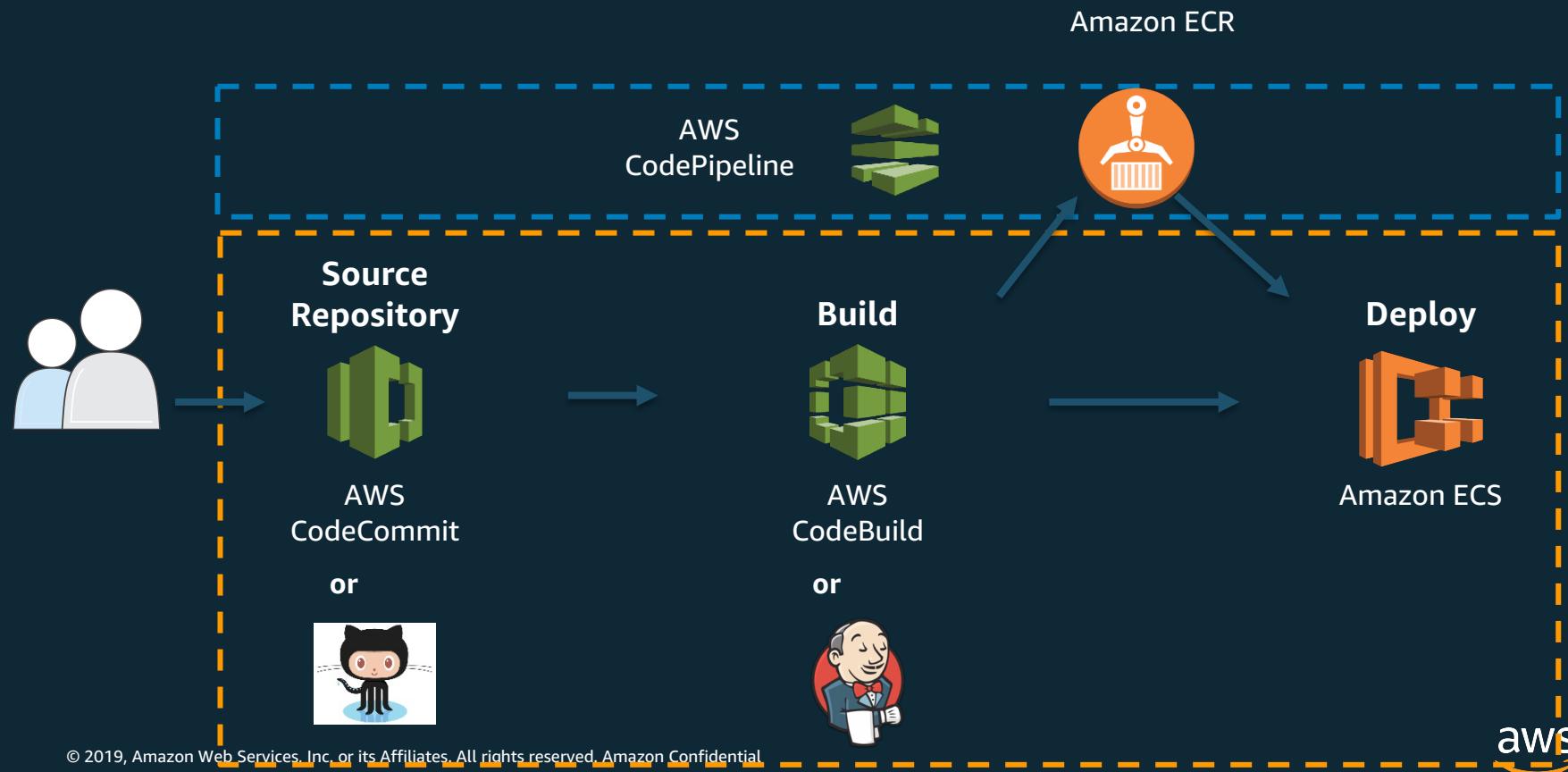
Highly Available



Simplified Workflow

Integrates with Amazon ECS

Using AWS CodePipeline with ECS and ECR



Why customers love AWS container services



Deeply integrated with AWS

Broad selection of compute instances and IAM security, VPC networking, load balancing, and autoscaling



DevOps Workflow

Best place to build and operate a complete DevOps workflow for containers—AWS DevTools and Cloud9



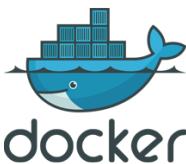
Security and Compliance

ISO, HIPPA, PCI, SOC1, SOC2, SOC3
Infocomm Media Development Auth.

Containers are a first-class citizen of the AWS Cloud

Rich partner ecosystem

Foundation



MESOSPHERE

DevOps



Monitoring & Logging



Security



Networking



TiGERA
CLOUD NETWORKS. SECURED



THANK YOU

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