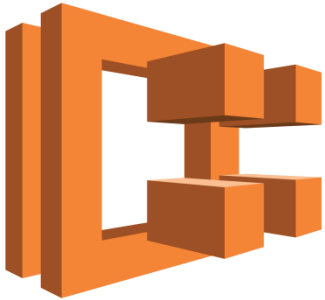


Amazon EC2 Container Service (ECS)

a platform to run production containers

Uttara Sridhar

Software Developer Engineer

amazon web services

Agenda

Why Containers?

Cluster Management

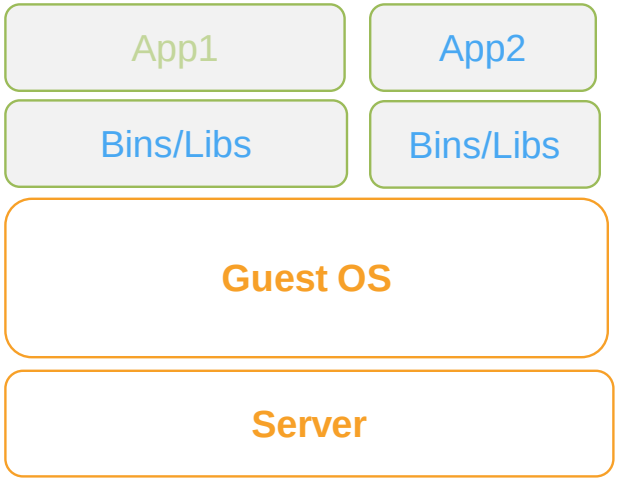
ECS Architecture

Use Cases

Q&A

Why Containers?

What are Containers?



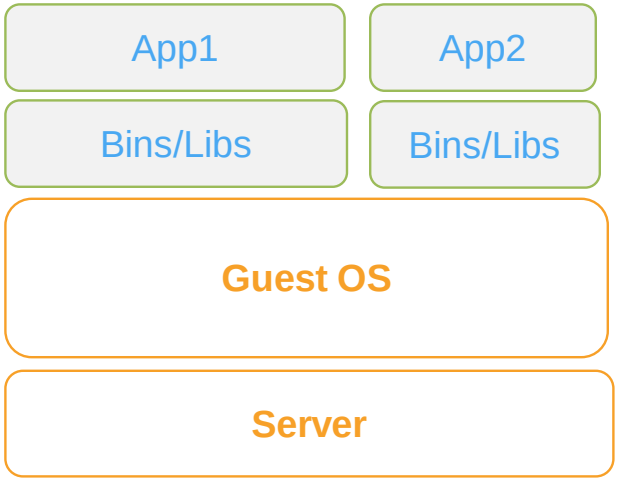
OS virtualization

Process isolation

Images

Automation

Container advantages



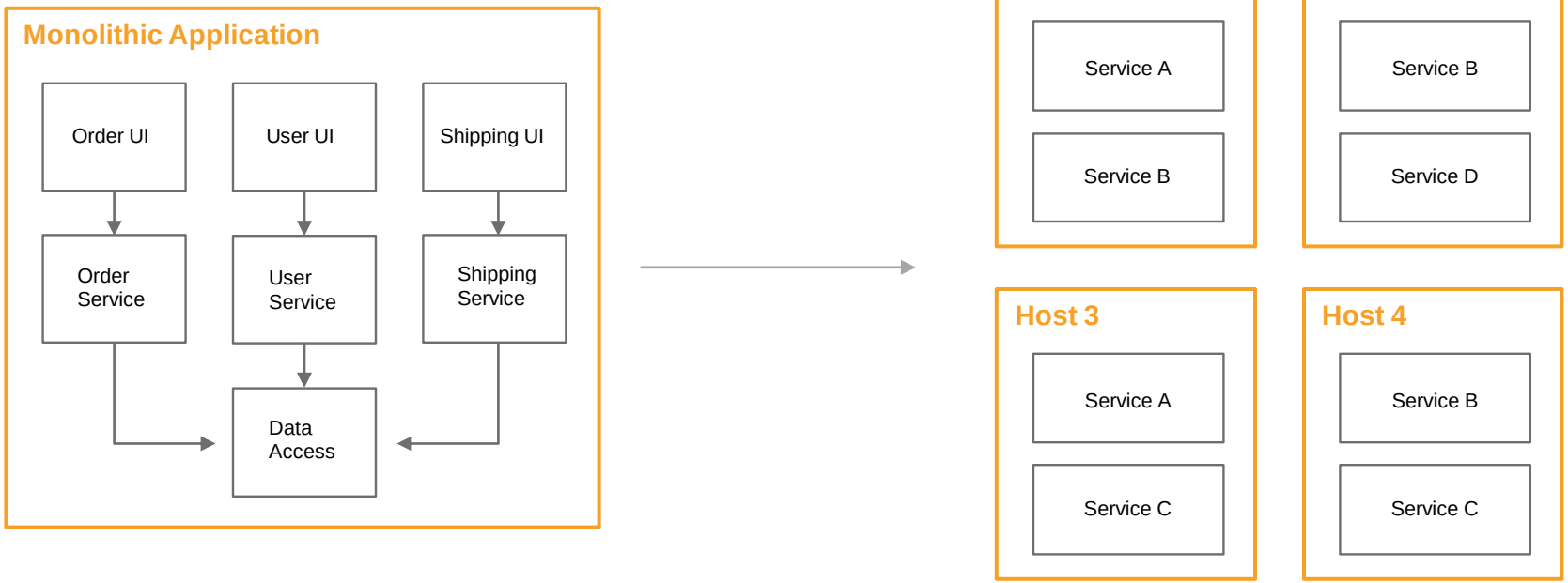
Portable

Flexible

Fast

Efficient

Services evolve to microservices



Containers are natural for microservices

Simple to model

Any app, any language

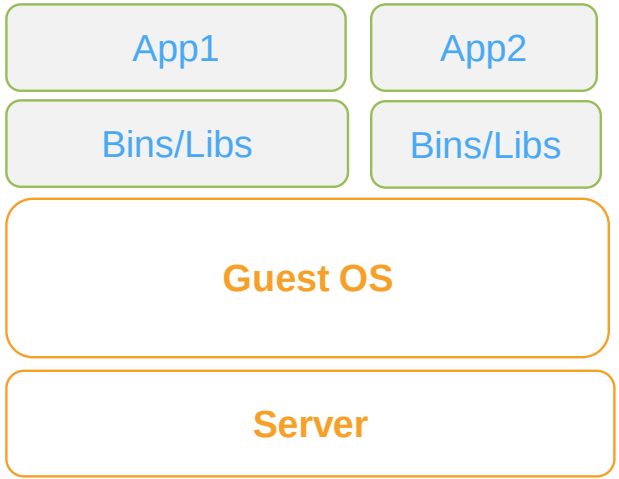
Image is the version

Test & deploy same artifact

Stateless servers decrease change risk

Why build Amazon ECS?

Scheduling one resource is straightforward



Scheduling a cluster is hard

Some customer challenges

Cluster management

Availability

Scheduling

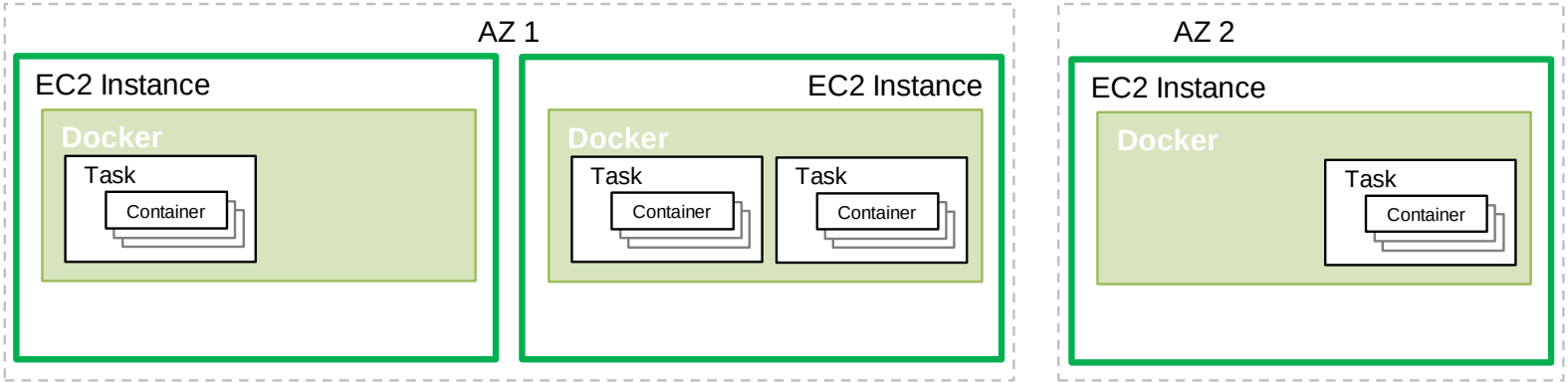
Security

Monitoring

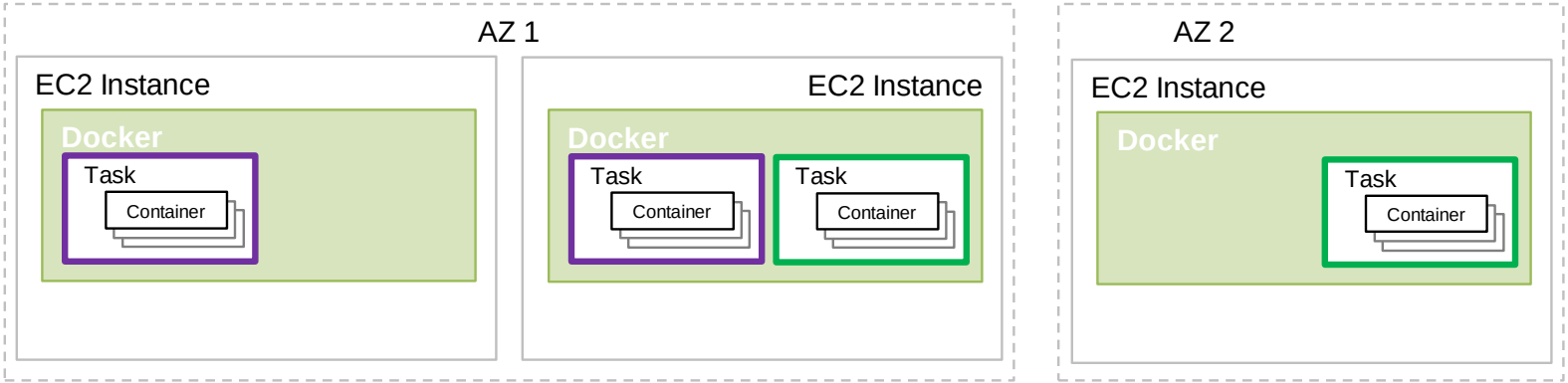
Integration with AWS services

Cluster Management

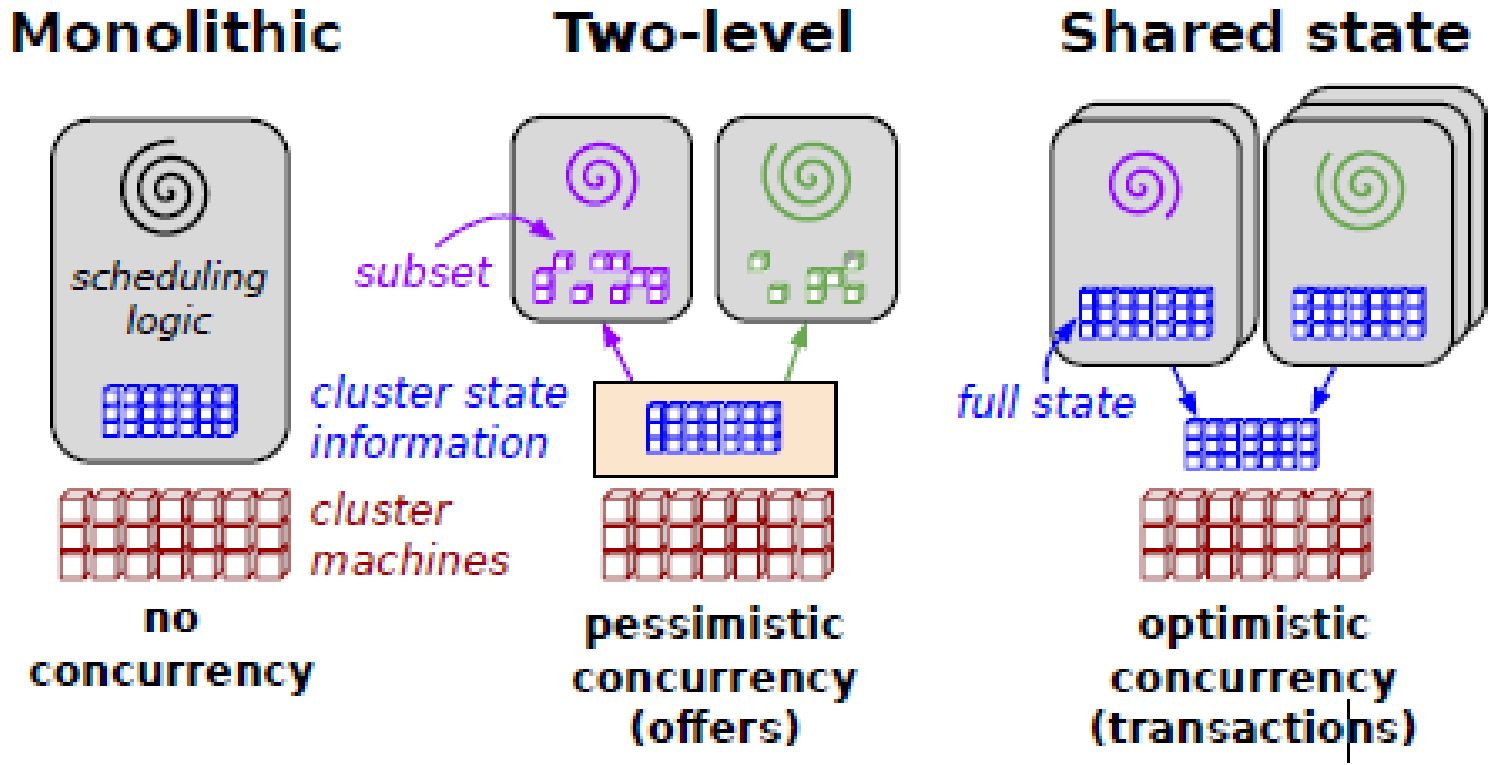
Cluster Management: Resource Management



Cluster Management: Scheduling



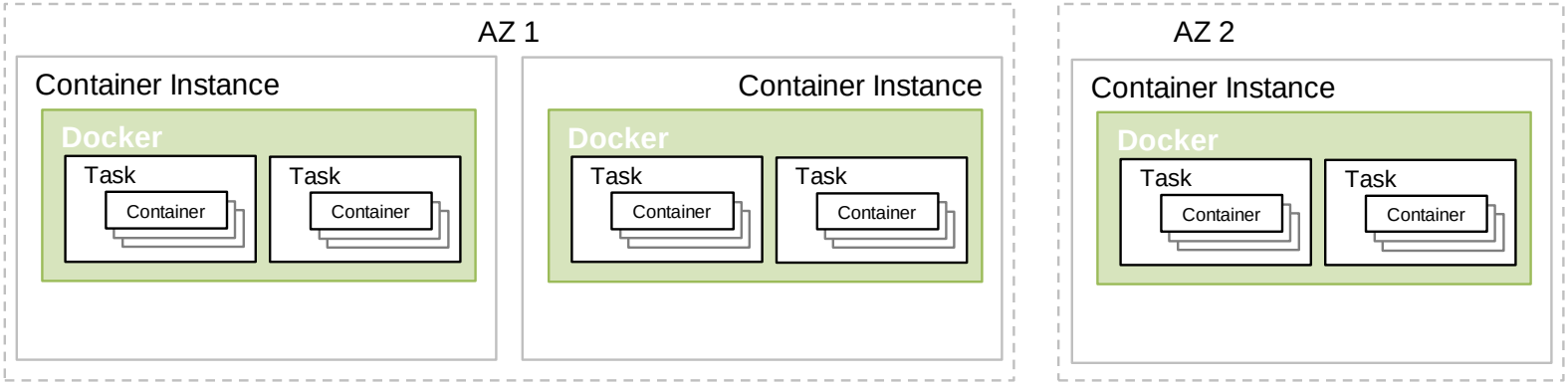
Cluster Management: Scheduling Systems



Ref: http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41684.pdf

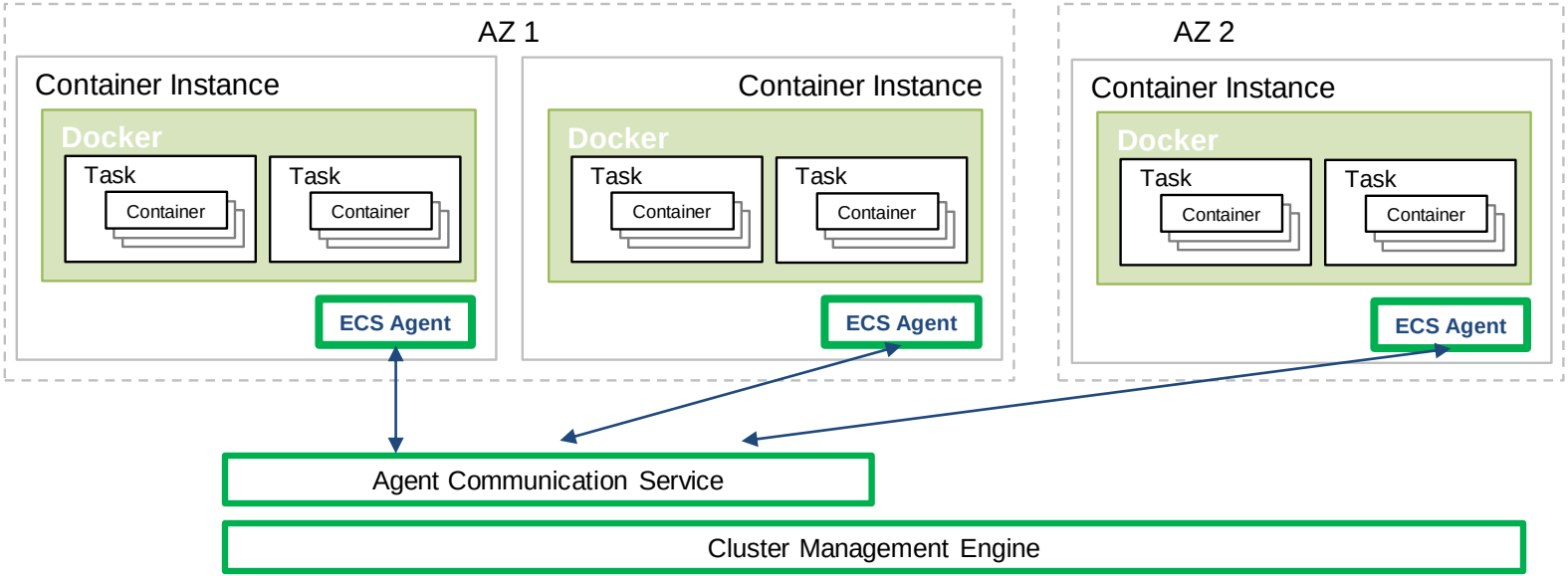
Amazon ECS

Amazon ECS: Resource Management

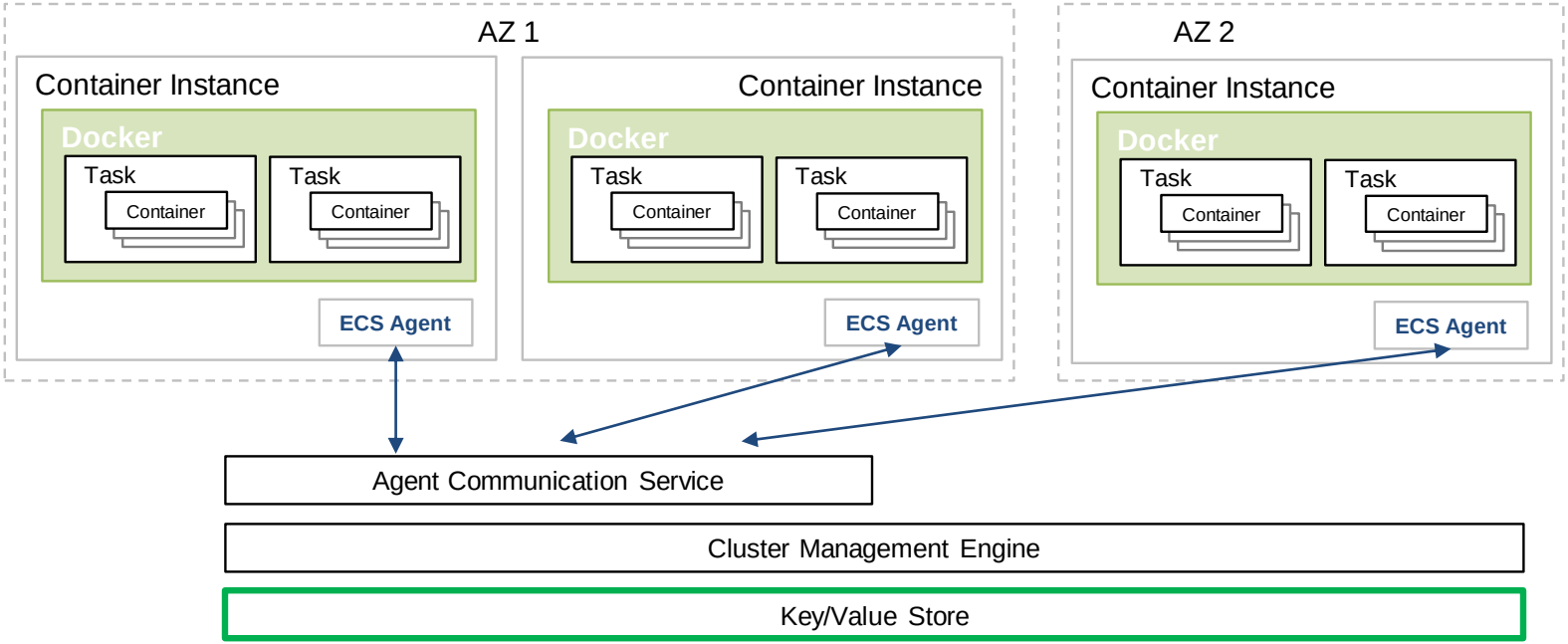


Cluster Management Engine

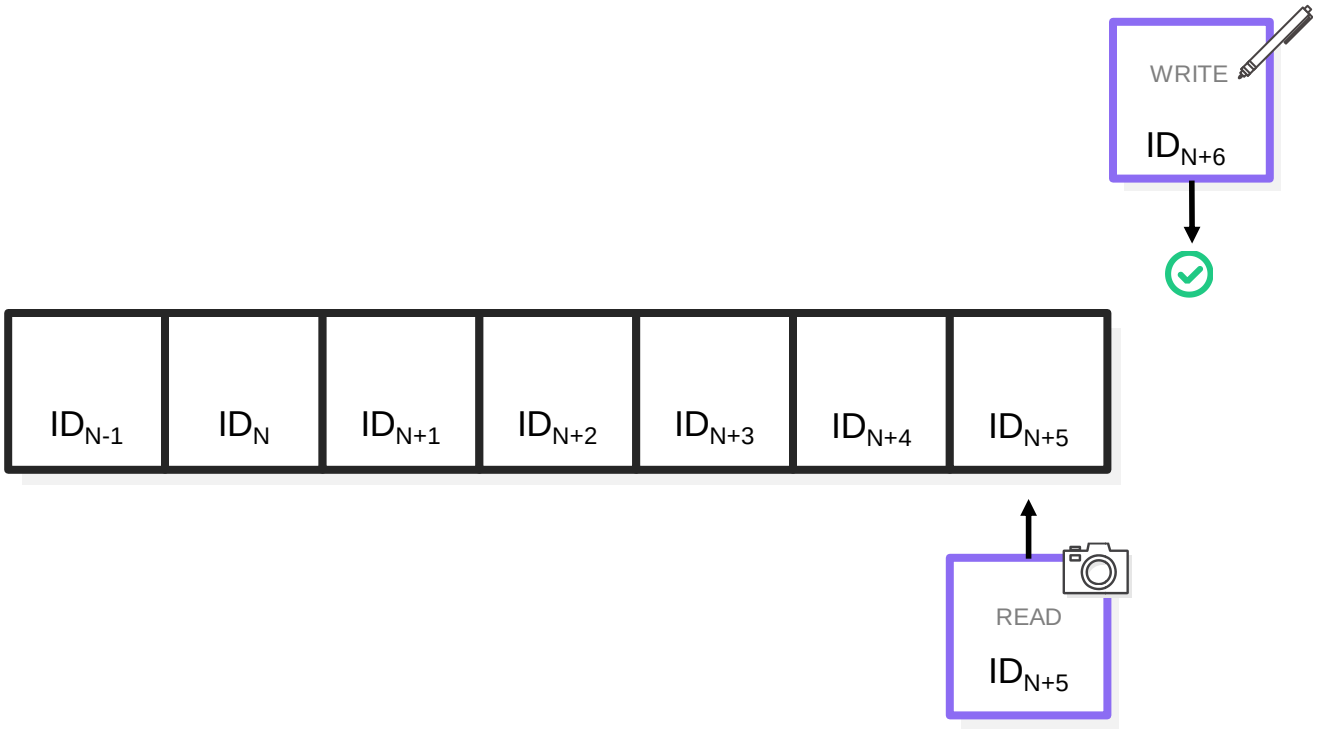
Amazon ECS: Agent Communication



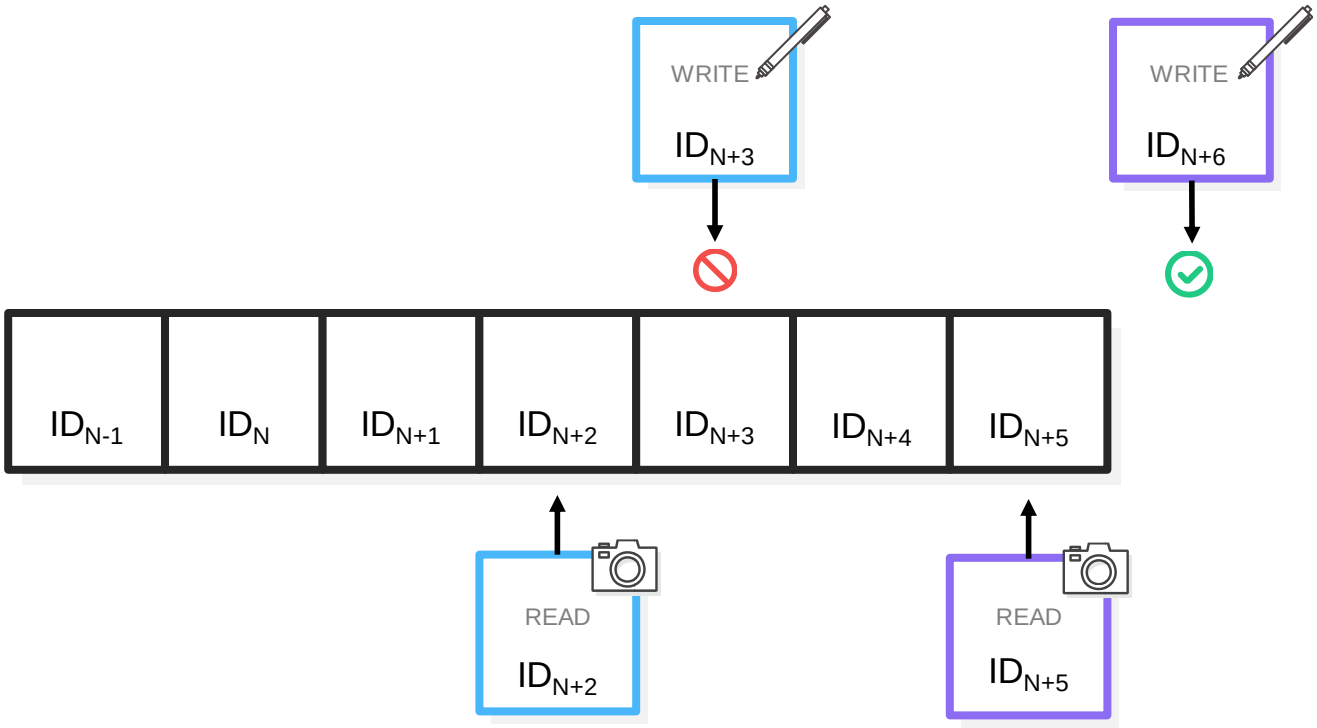
Amazon ECS: Key/Value Store



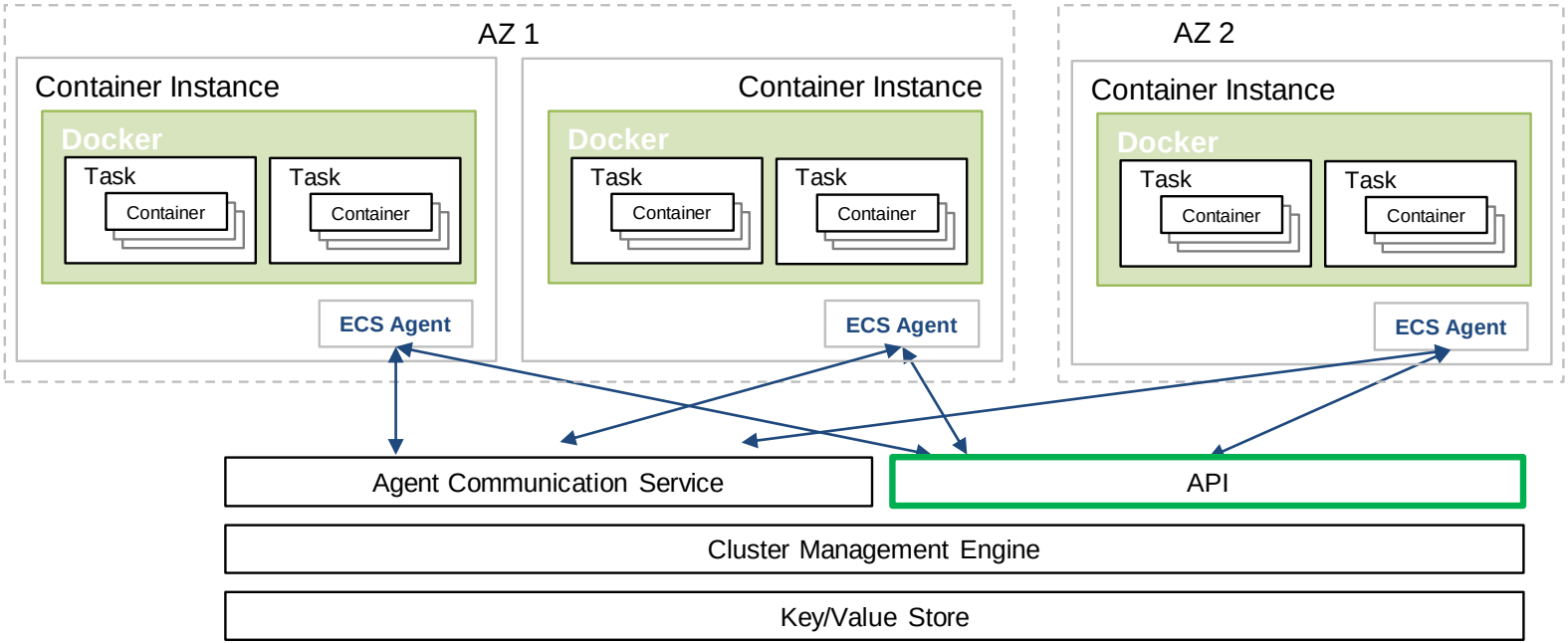
Amazon ECS under the Hood



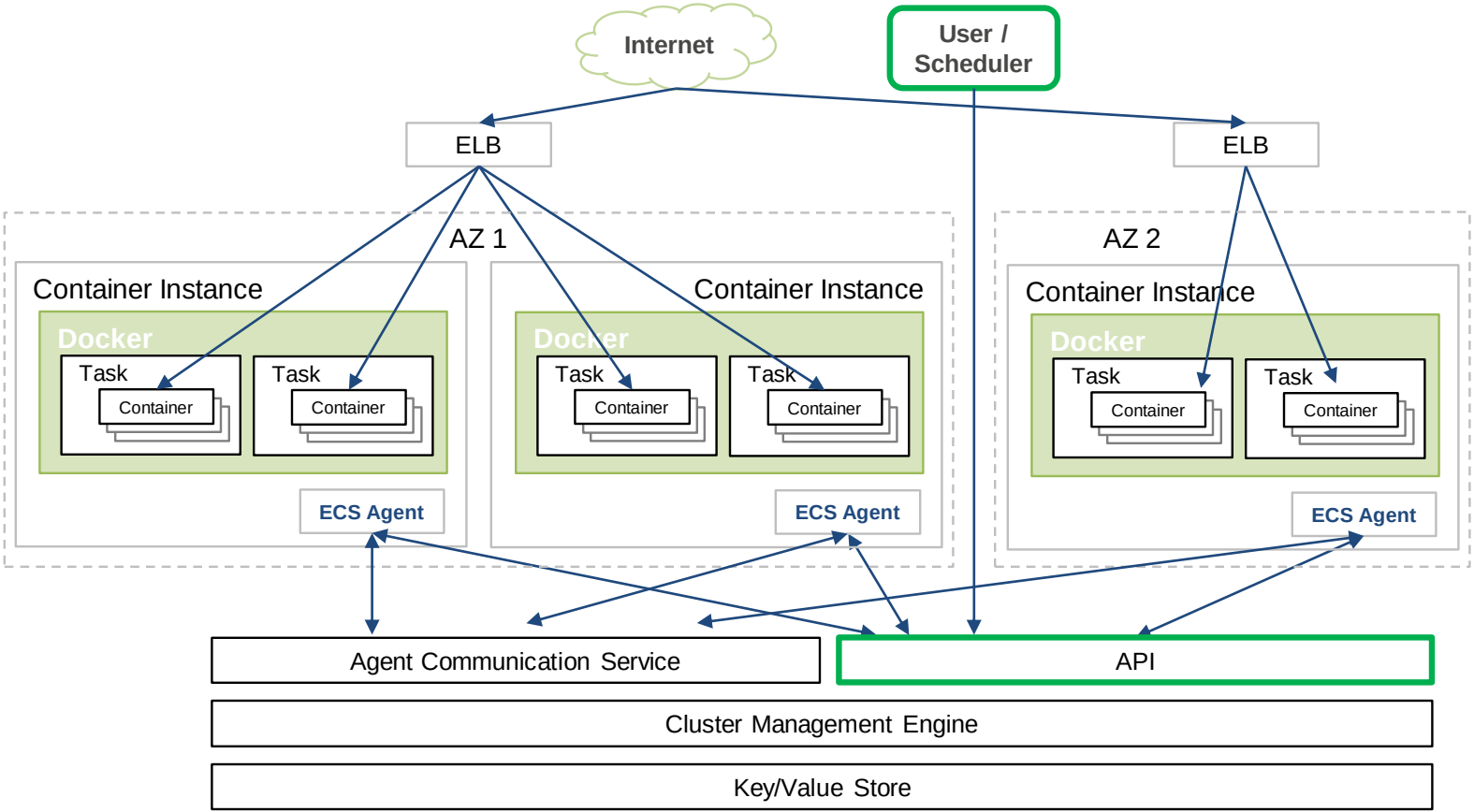
Amazon ECS under the Hood



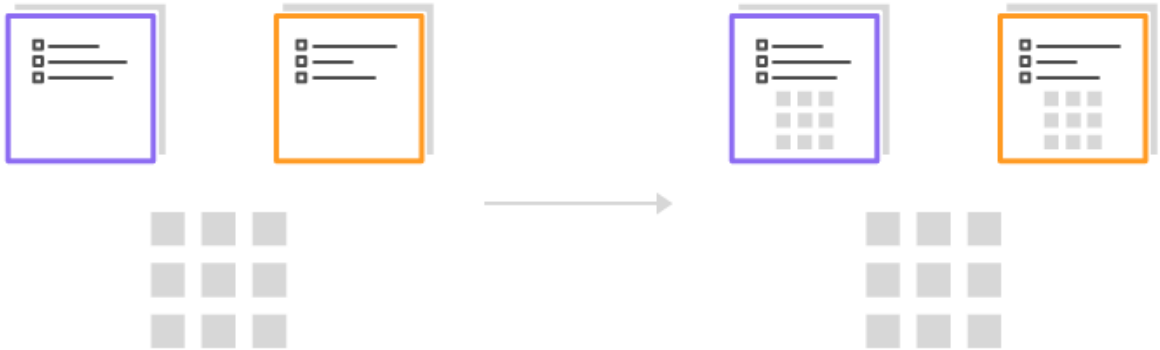
Amazon ECS: APIs



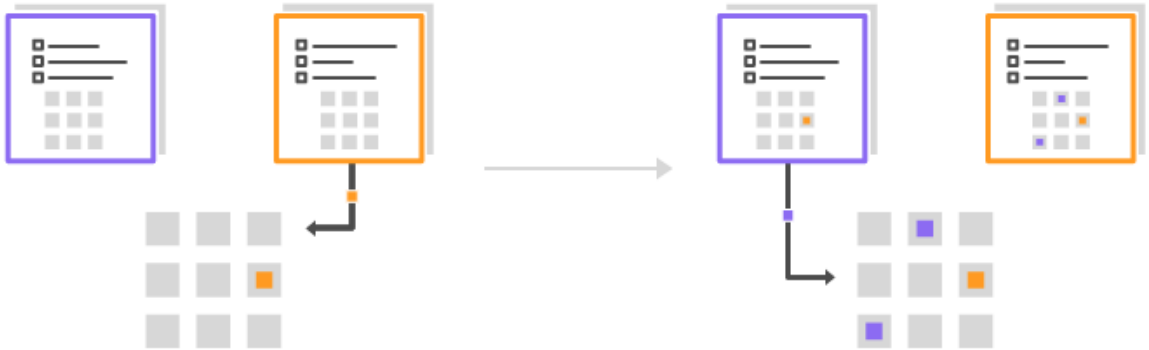
Amazon ECS: Scheduling



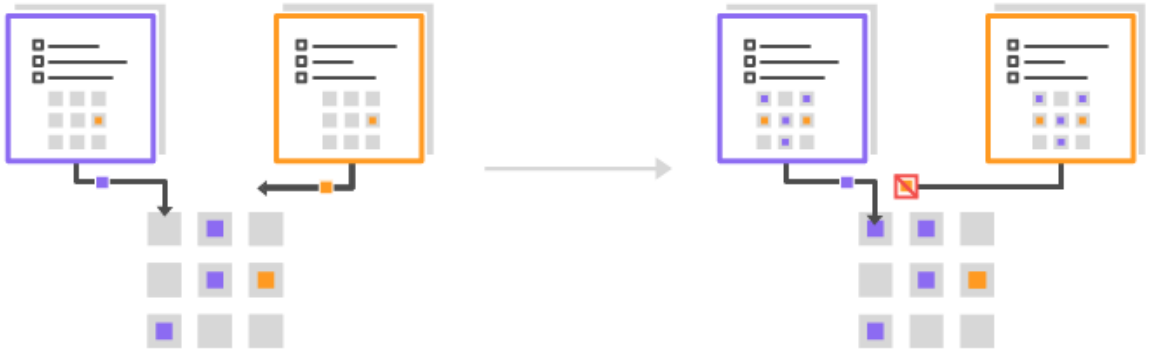
Amazon ECS: Scheduling



Amazon ECS: Scheduling



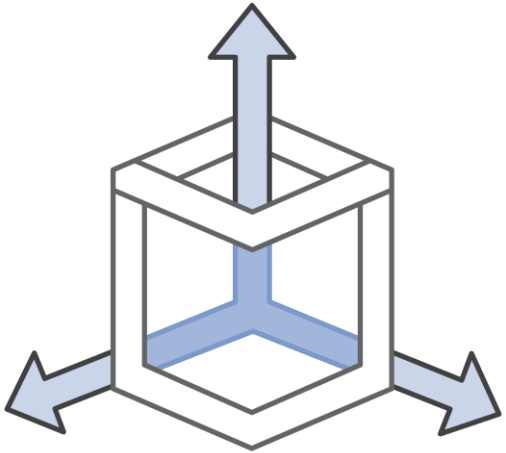
Amazon ECS: Scheduling



Amazon ECS: Scheduling

Use Cases

Easily Manage Clusters for Any Scale



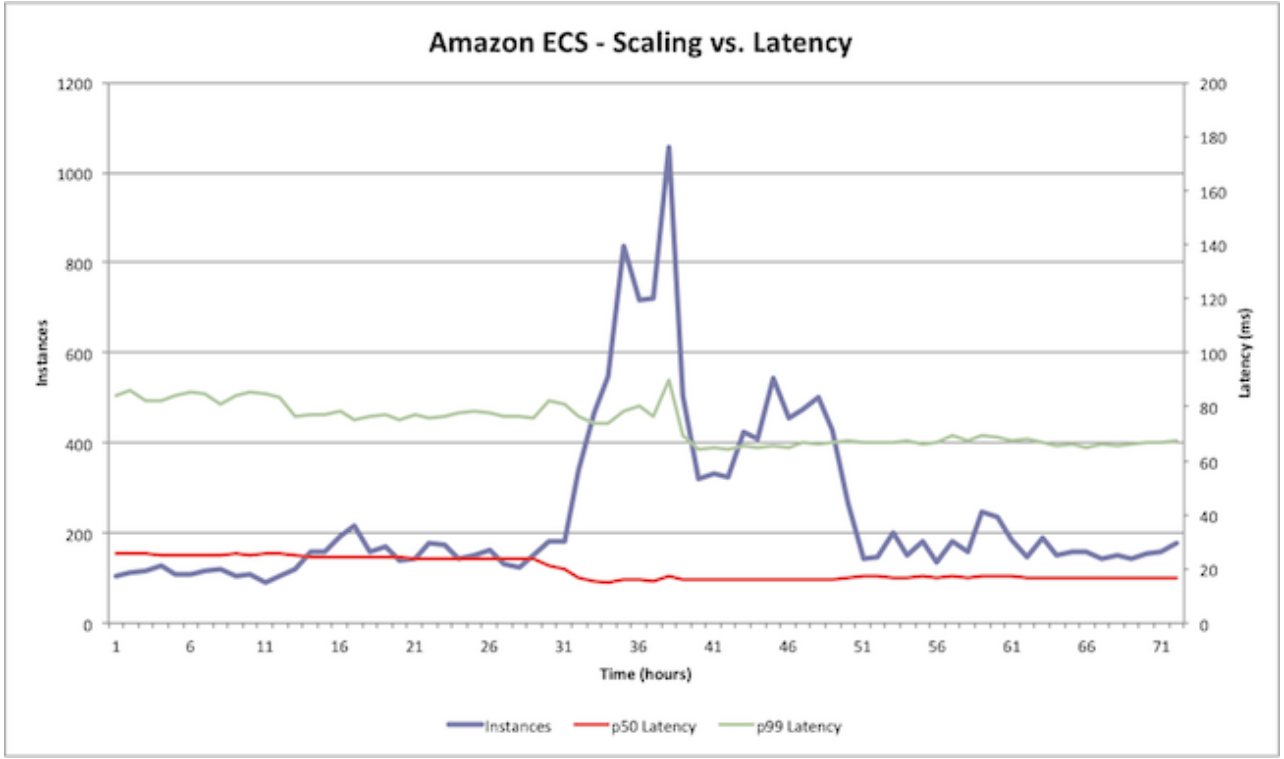
Nothing to run

Complete state

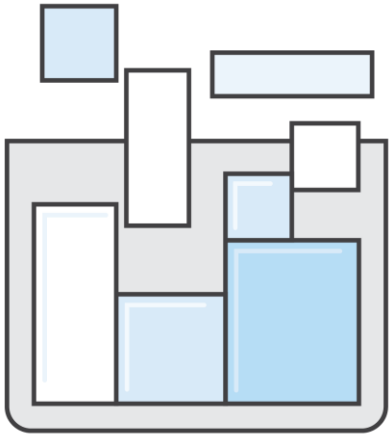
Control and monitoring

Scale

Scalable


Amazon ECS - Scaling vs. Latency
Instances
Latency (ms)
Time (hours)
1200
1000
800
600
400
200
0
200
180
160
140
120
100
80
60
40
20
0
1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
Instances
p50 Latency
p99 Latency

Flexible Container Placement

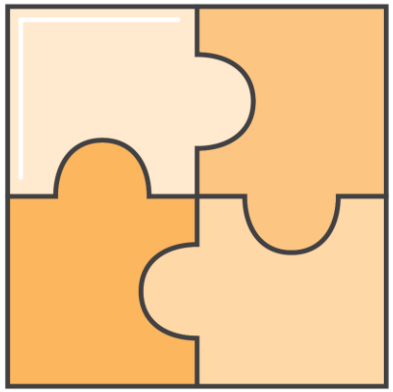


Applications

Batch jobs

Multiple schedulers

Designed for use with other AWS services



Amazon Virtual Private Cloud

AWS Identity and Access Management

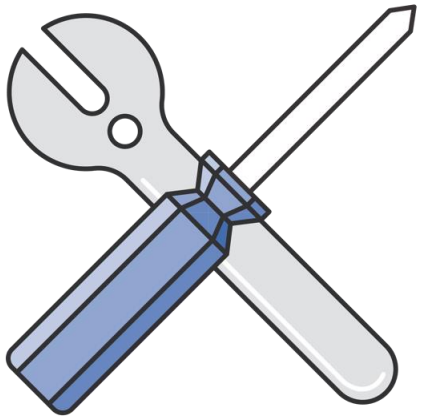
Amazon Elastic Load Balancing

Amazon Elastic Block Store

Amazon CloudWatch

AWS CloudTrail

Extensible



Comprehensive APIs

Custom schedulers

Open source agent and CLI

Common Use Cases

Applications and services

- Configuration and deployment
- Microservices

Batch processing

Case Study: Shippable

> With Amazon ECS, we've practically eliminated the time our developers spent on ops-related tasks. Our senior developers used to spend 80% of their time on back-end infrastructure management features, whereas now they spend 80% of their time on customer features.
>
> **Avi Cavale**
> CEO & Cofounder
>
> **Shippable**

Shippable is a platform providing hosted continuous integration, testing, and deployment from repositories. The Shippable platform consists of two parts: Continuous Integration (CI), and Continuous Delivery (CD) pipelines.

Built a CICD platform with microservices architecture using Docker containers on Amazon but their service discovery solution and monitoring infrastructure was really hard to scale and manage.

Evaluated many open source options, but wanted a solution that was simple and would integrate with the AWS ecosystem. Started using Amazon ECS as a way to offload cluster management and container orchestration to a service.

Amazon ECS service scheduler manages multiple copies of each microservice across the ECS cluster, use Amazon ELB for load-balancing, Amazon Cloudwatch for telemetry and infrastructure logging, Amazon ECR for storing Docker images.

Case Study: Segment

> Switching to Amazon ECS has greatly simplified running a service without needing to worry about provisioning or availability.
>
> **Calvin French-Owen**
> Cofounder and Chief Technology Officer
>
> Segment

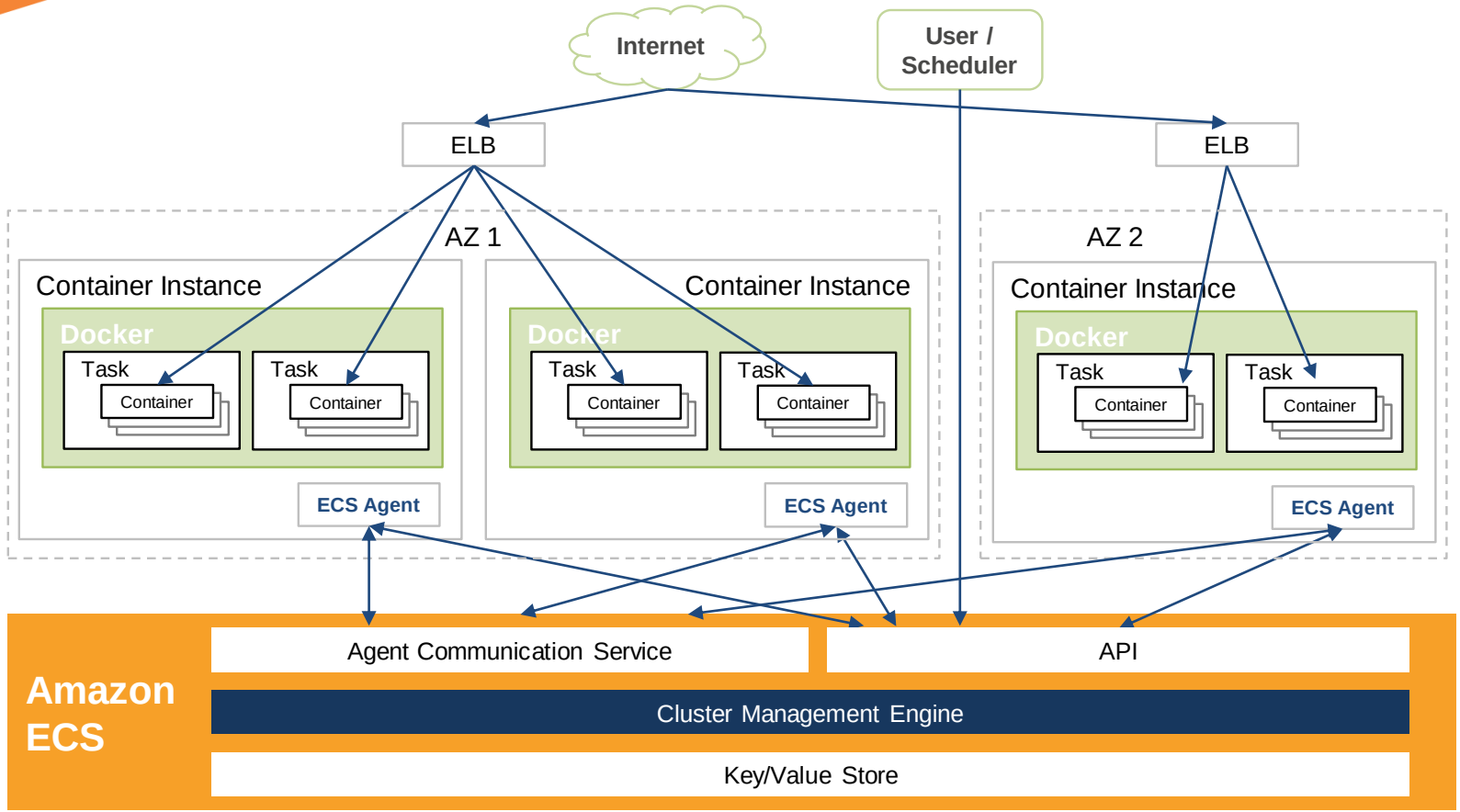
Segment provides a service used by businesses to collect customer data in a single hub for later use in analytics, marketing, and for other purposes.

Moved from Amazon EC2 instances to Docker containers and needed a way to manage and schedule containers at scale for their production environment.

Evaluated many open source options, but wanted a solution that was simple and would integrate with the AWS ecosystem. Started using Amazon EC2 Container Service (Amazon ECS) as a way to offload cluster management and container orchestration to a service.

Amazon ECS manages placement of containers on different Amazon EC2 instances across multiple Availability Zones, providing the Segment service with better availability.

Amazon ECS


Internet
User / Scheduler
ELB
ELB
AZ 1
AZ 2
Container Instance
Container Instance
Container Instance
Docker
Docker
Docker
Task
Container
ECS Agent
Amazon ECS
Agent Communication Service
API
Cluster Management Engine
Key/Value Store

Thank you!

Questions?