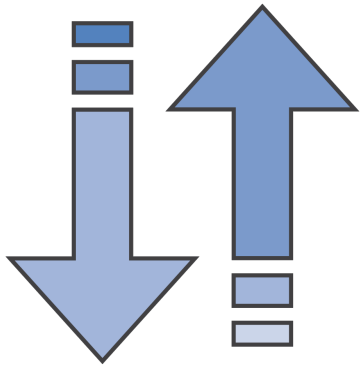
Containers and the Evolution of Computing

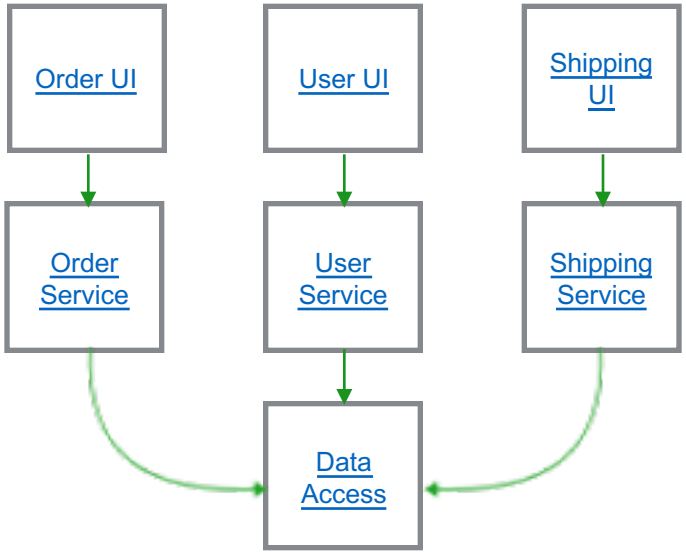
Matt Nowina

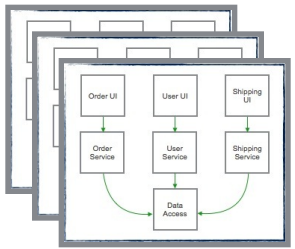
Solutions Architect

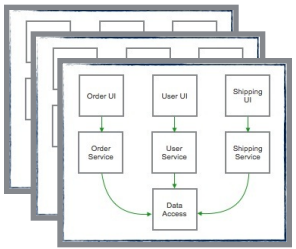
amazon web services

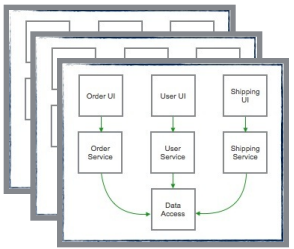
Scaling Applications

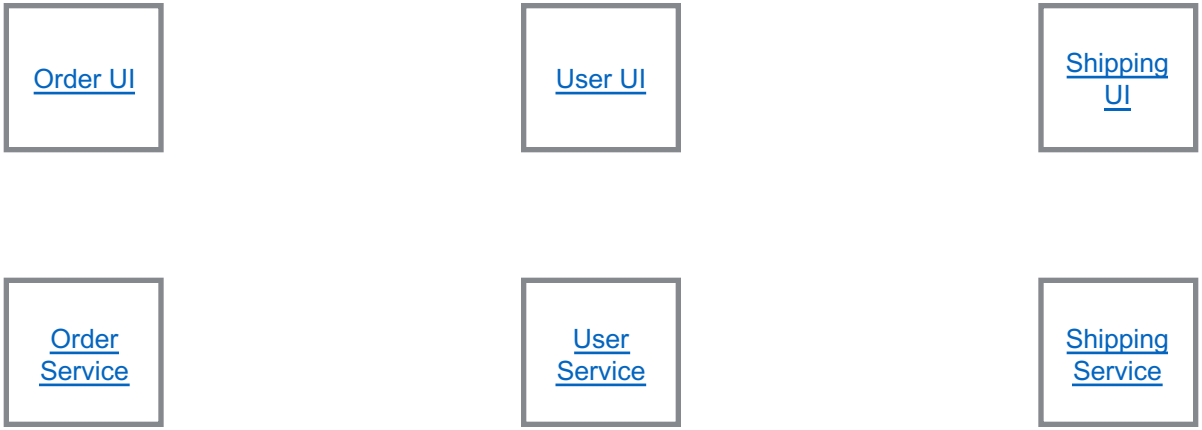


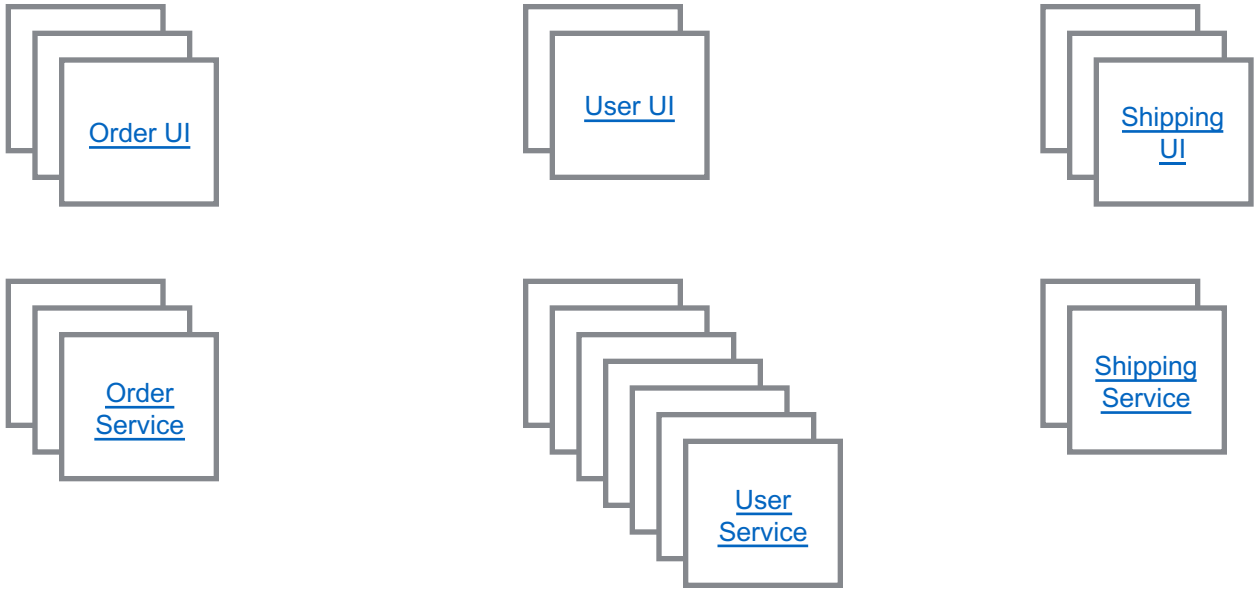

Order UI
User UI
Shipping UI
Order Service
User Service
Shipping Service
Data Access

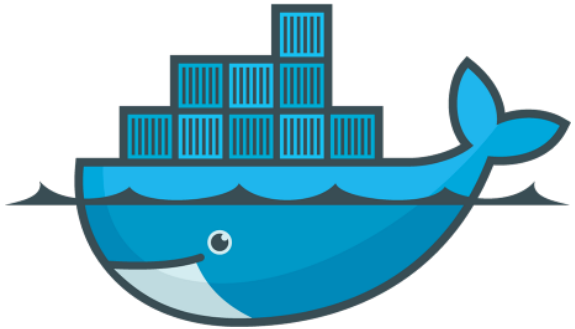

Order UI
User UI
Shipping UI
Order Service
User Service
Shipping Service
Data Access


Order UI
User UI
Shipping UI
Order Service
User Service
Shipping Service
Data Access

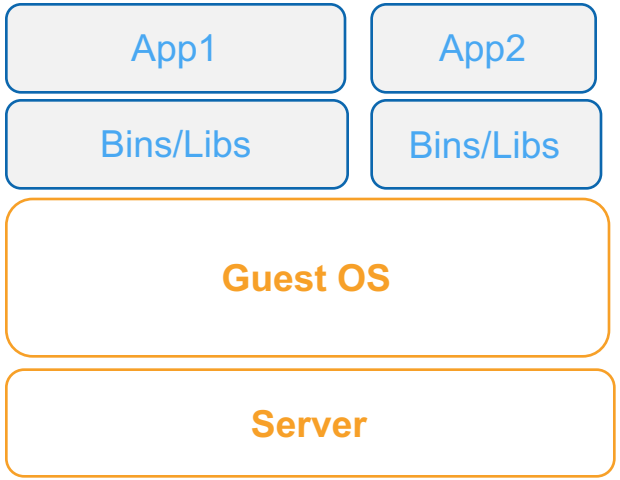

Order UI
User UI
Shipping UI
Order Service
User Service
Shipping Service
Data Access


Order UI
User UI
Shipping UI
Order Service
User Service
Shipping Service


Order UI
User UI
Shipping UI
Order Service
User Service
Shipping Service



What are Containers?



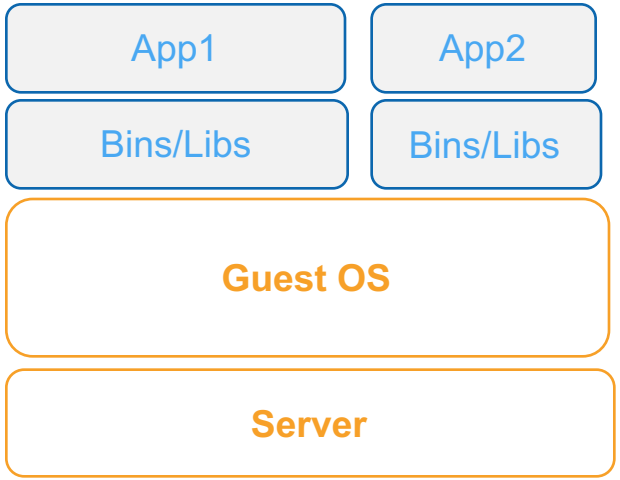
OS virtualization

Process isolation

Images

Automation

Container advantages



Portable

Flexible

Fast

Efficient

Containers are natural for microservices

Simple to model

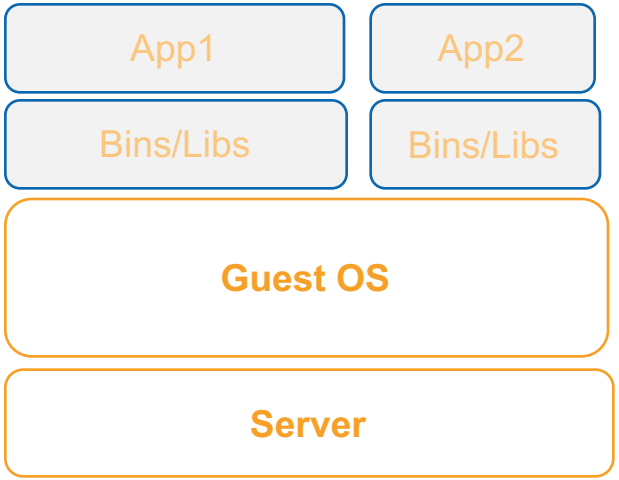
Any app, any language

Image is the version

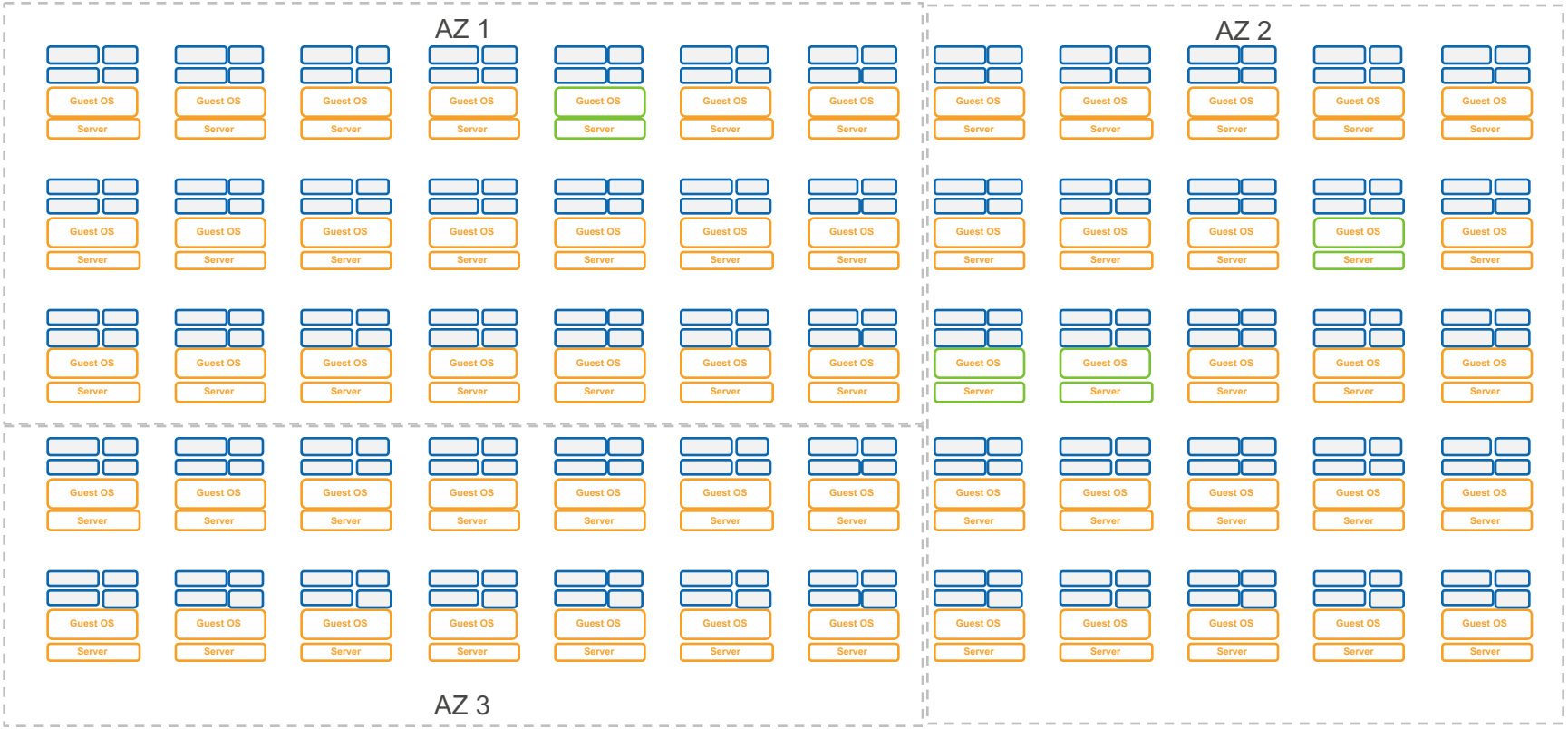
Test & deploy same artifact

Stateless servers decrease change risk

Managing one host is straightforward



Managing a fleet is hard

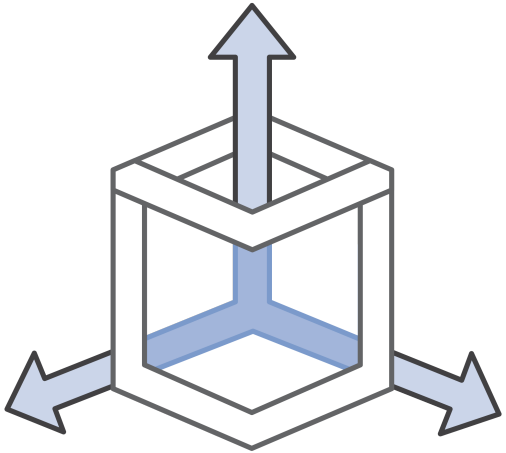


What is Amazon ECS?

Amazon EC2 Container Service (ECS) is a highly scalable, high performance **container management service**. You can use Amazon ECS to **schedule** the placement of containers across your cluster. You can also integrate your own **scheduler** or **third-party scheduler** to meet business or application specific requirements.

Our Goals with Amazon ECS

Container Management at Any Scale



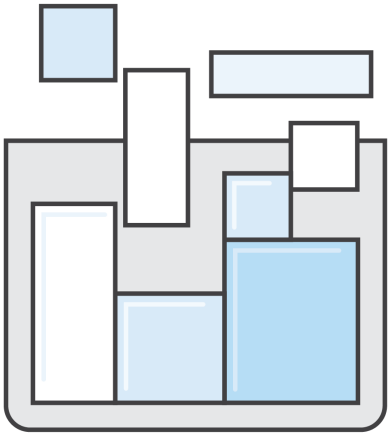
Nothing to run

Complete state

Control and monitoring

Scale

Flexible Container Placement

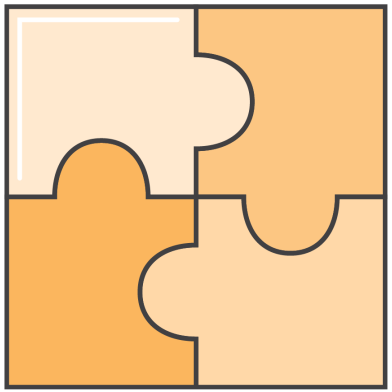


Long running applications

Batch jobs

Multiple schedulers

Integration with the AWS Platform



Elastic Load Balancing

Amazon Elastic Block Store

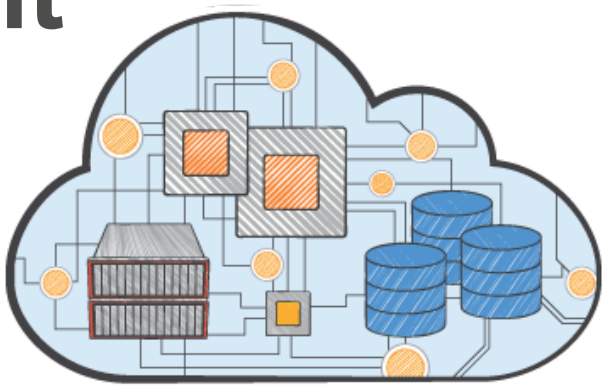
Amazon Virtual Private Cloud

Amazon CloudWatch

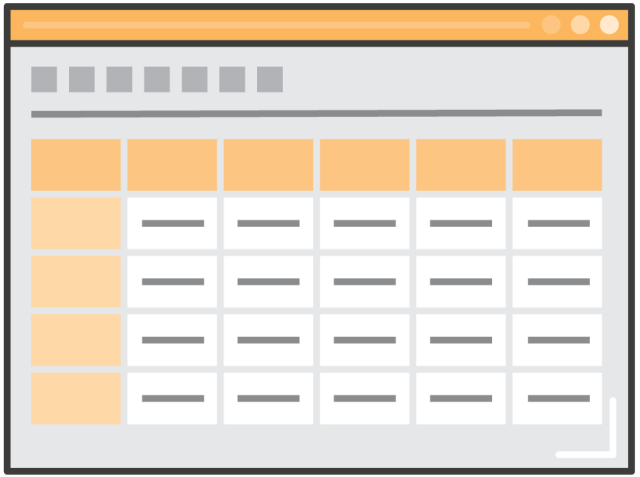
AWS Identity and Access Management

AWS CloudTrail

Container Management



What is a Container Manager?



- Maintains Available Resources
- Tracks Resource Changes
- Accepts Resource Requests
- Guarantees Accuracy and Consistency

Resources

CPU

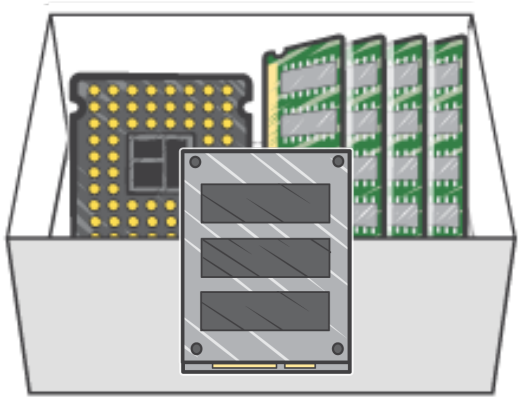
Memory

Ports

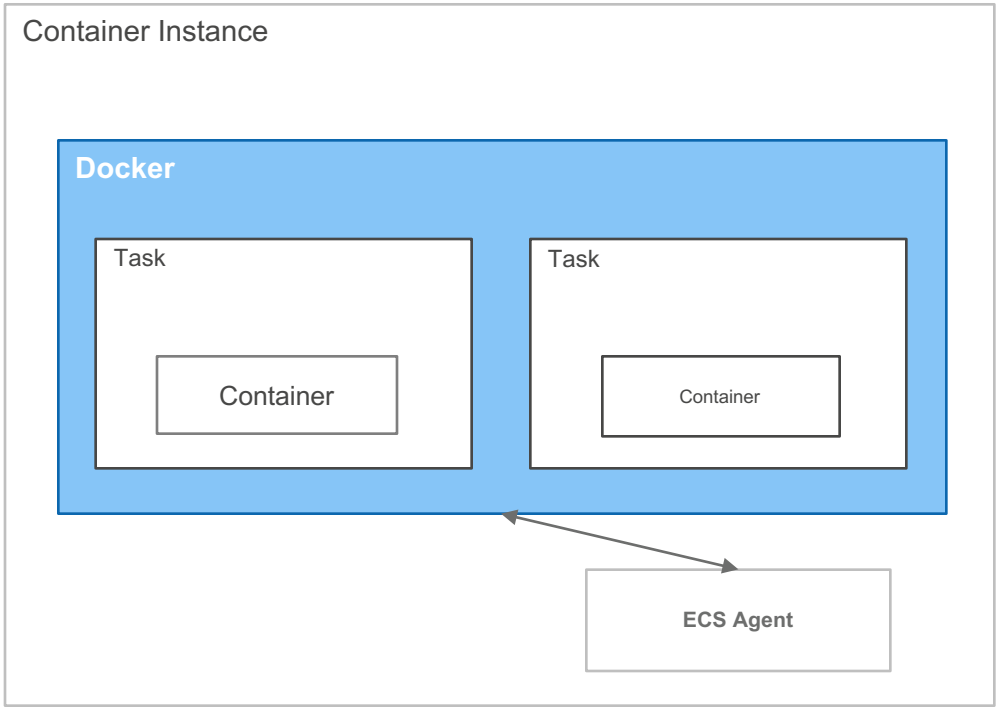
Disk Space

Disk IOPS

Network Bandwidth



ECS Agent



https://github.com/aws/amazon-ecs-agent

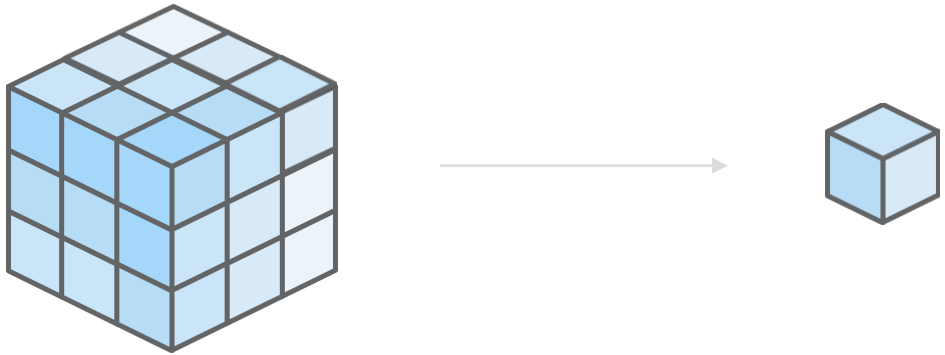
Instance Registration

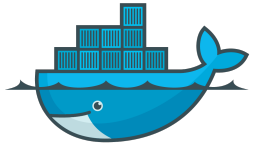
```
register-container-instance --total-resources

[
    {
        “name” : “cpu”,
        “type” : “integerValue”,
        “integerValue” : 2048
     },
     …
]
```

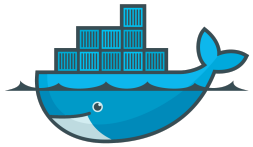
Modifying Exposed Resources

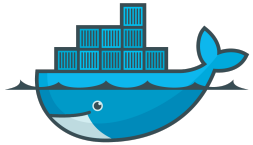
`ECS_RESERVED_MEMORY`	32	Memory, in MB, to reserve for use by things other than containers managed by ECS.	0
`ECS_RESERVED_PORTS`	`[22, 80, 5000, 8080]`	An array of ports that should be marked as unavailable for scheduling on this Container Instance.	`[22, 2375, 2376, 51678]`
`ECS_RESERVED_PORTS_UDP`	`[53, 123]`	An array of UDP ports that should be marked as unavailable for scheduling on this Container Instance.	`[]`

How do you model your applications?



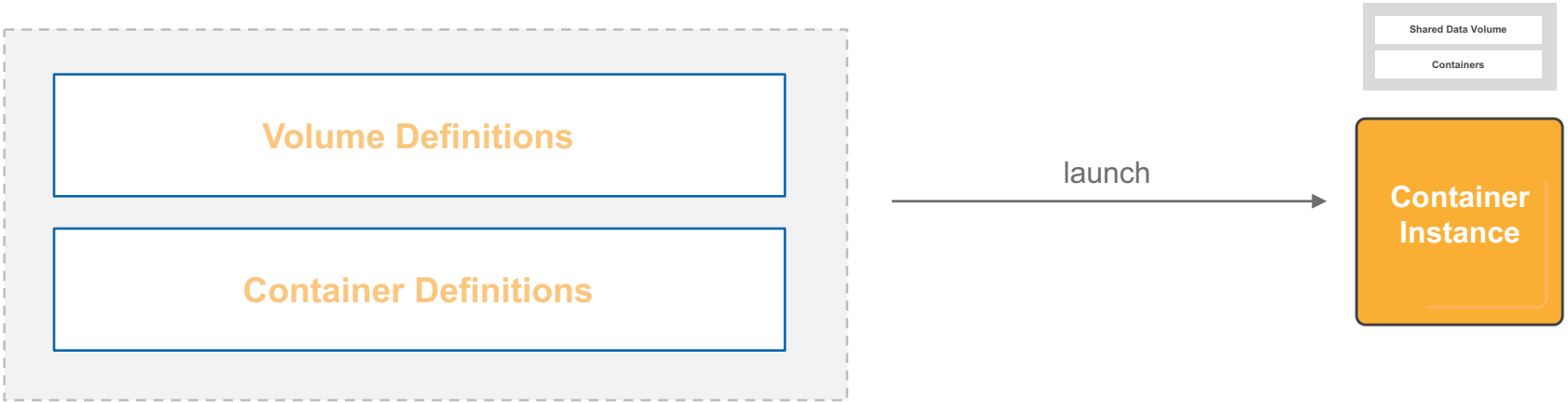




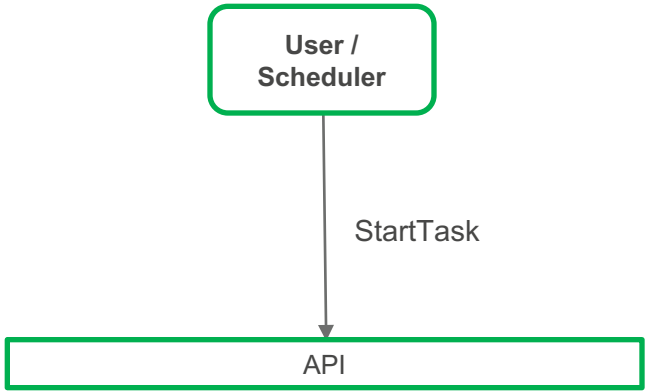


```
{
          "environment": [],
          "name": "simple-demo",
          "image": "my-demo",
          "cpu": 10,
          "memory": 500,
          "portMappings": [
              {
                  "containerPort": 80,
                  "hostPort": 80
              }
          ],
          "mountPoints": [
              {
                  "sourceVolume": "my-vol",
                  "containerPath": "/var/www/my-vol"
              }
          ],
          "entryPoint": [
              "/usr/sbin/apache2",
              "-D",
              "FOREGROUND"
          ],
          "essential": true
      },
```

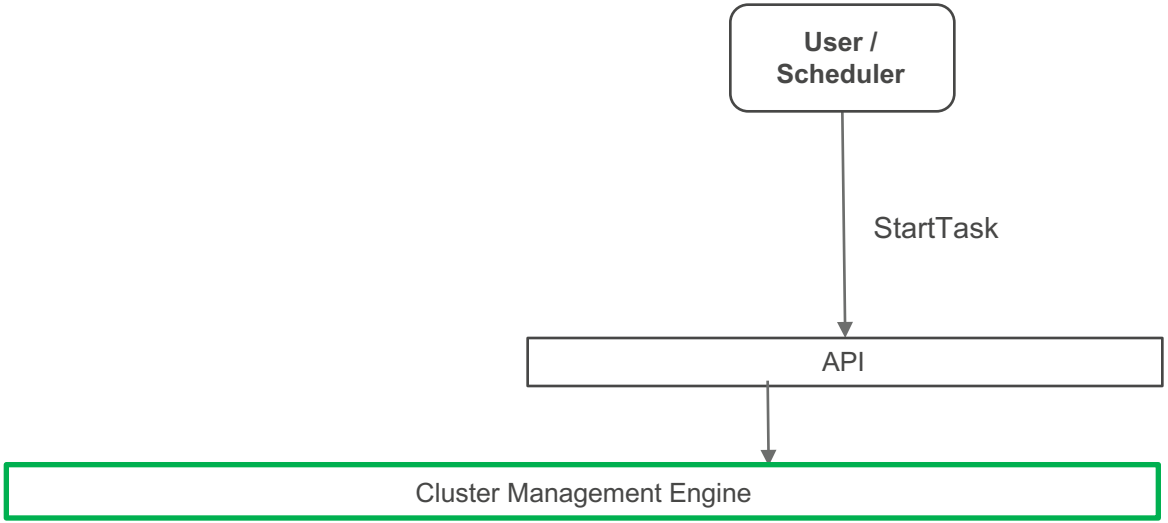
Tasks


Volume Definitions
Container Definitions
launch
Shared Data Volume
Containers
Container Instance

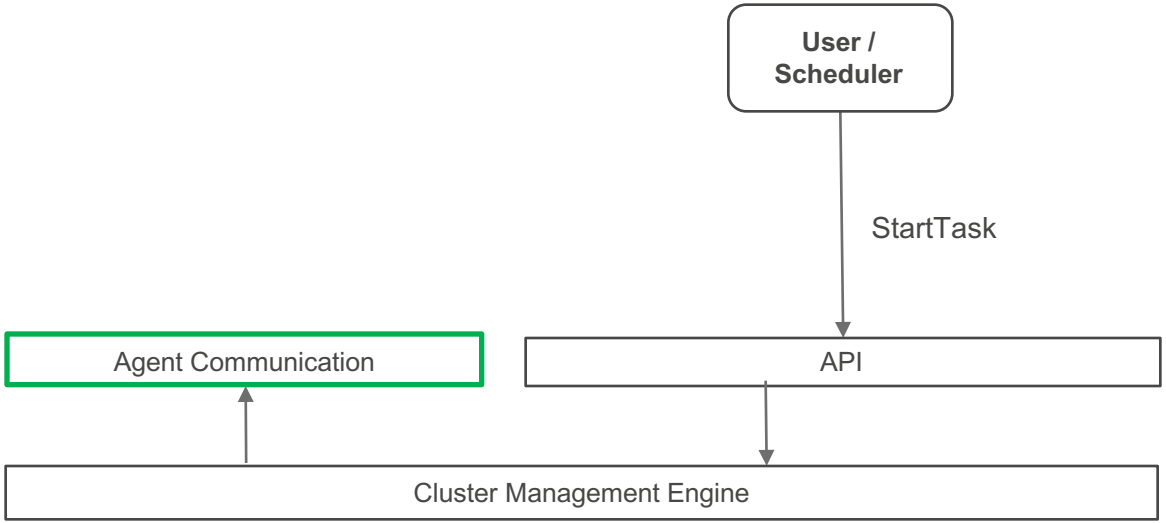
Starting a Task



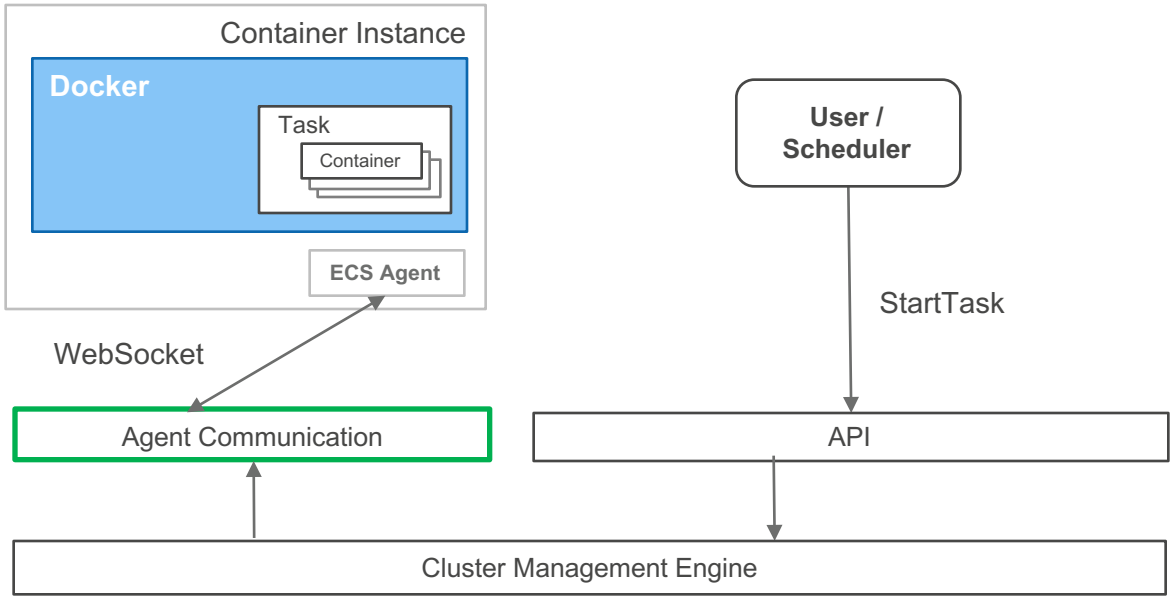
Starting a Task



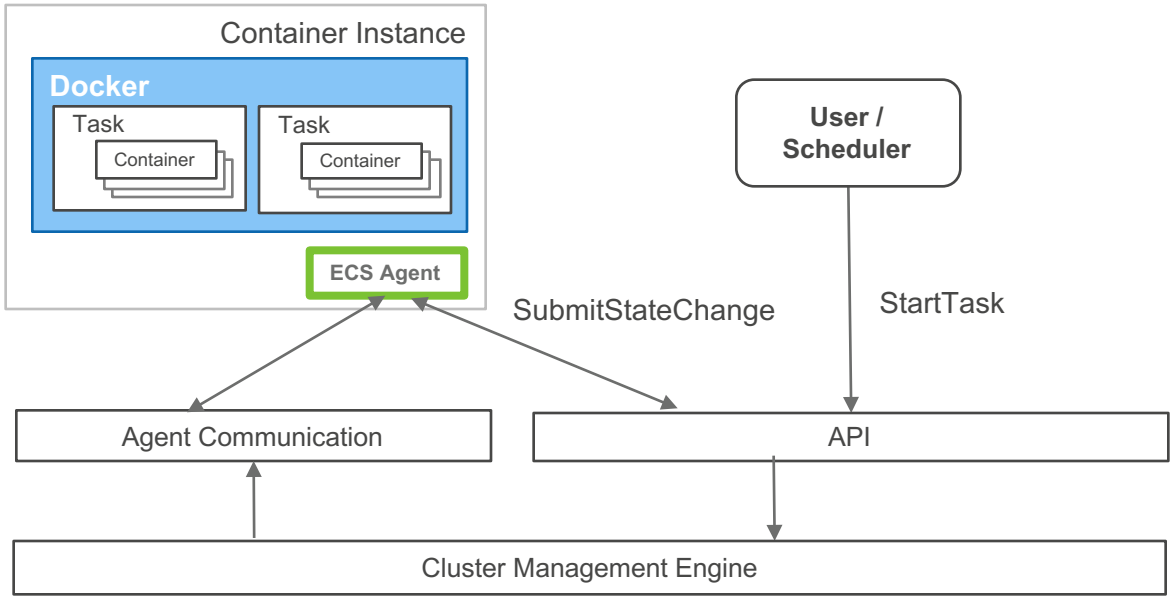
Starting a Task



Starting a Task

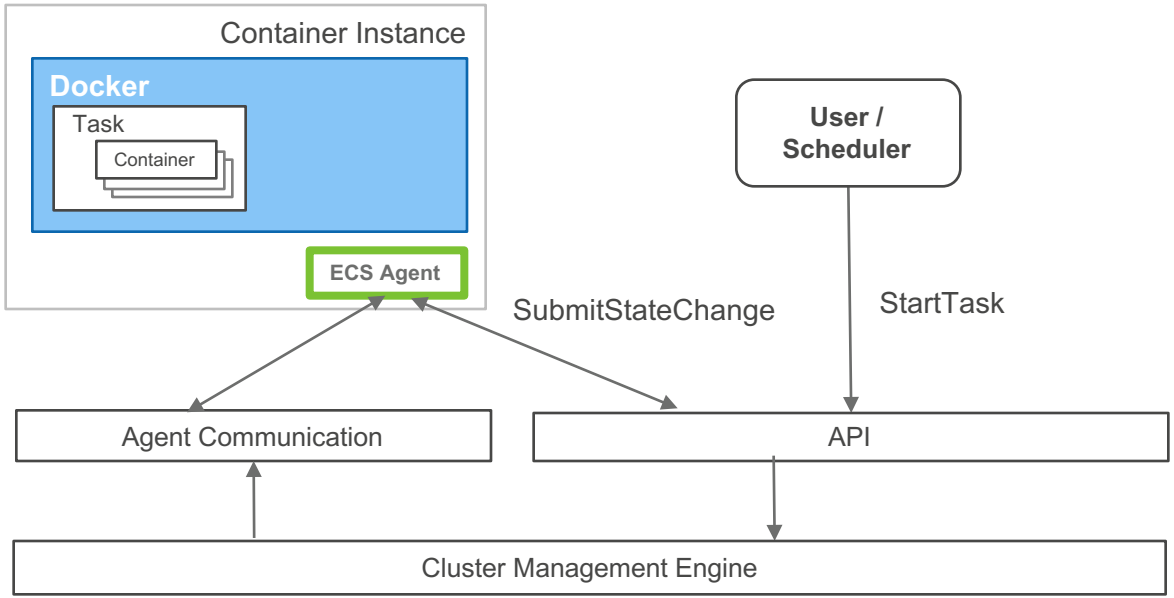


Starting a Task

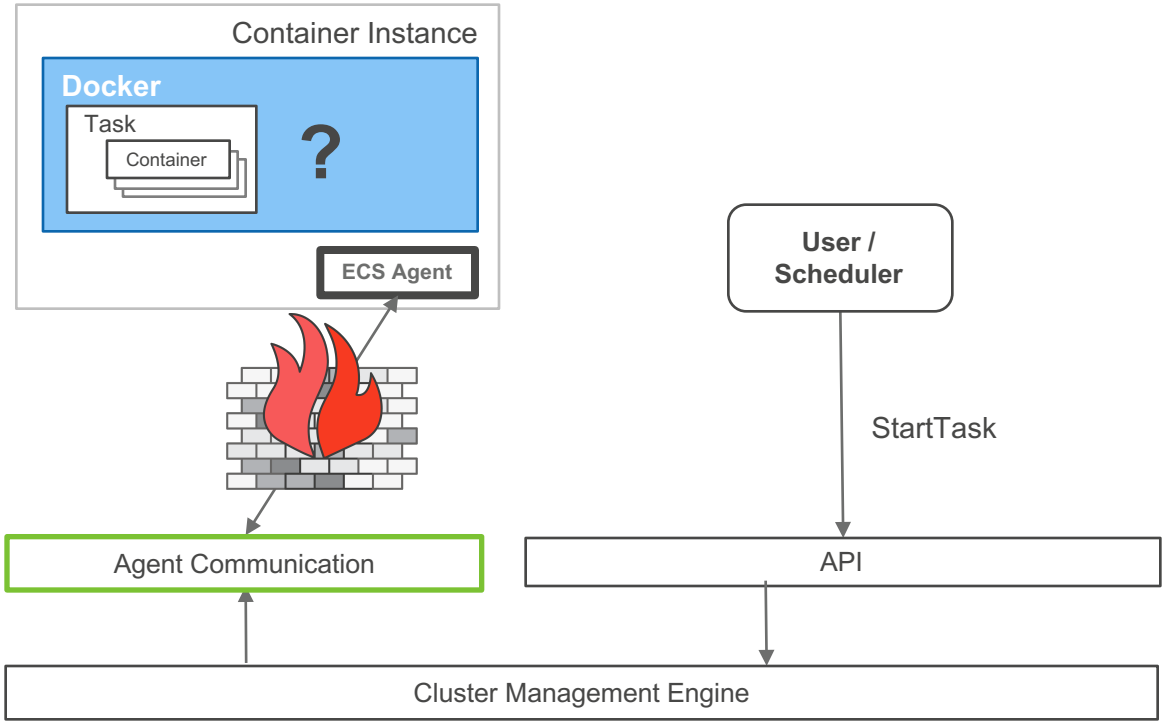


Tracking Resource Changes

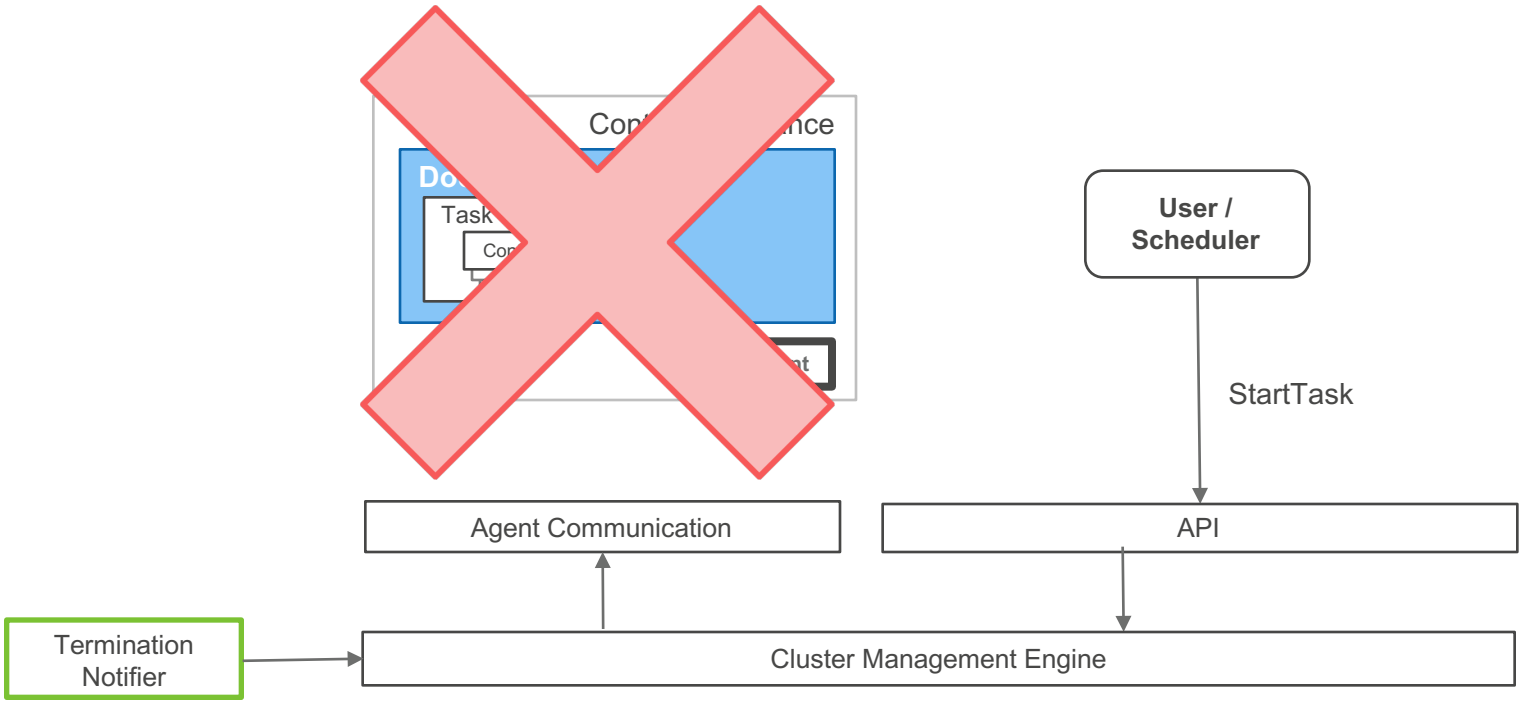
Terminated Task



Missing Container Instance

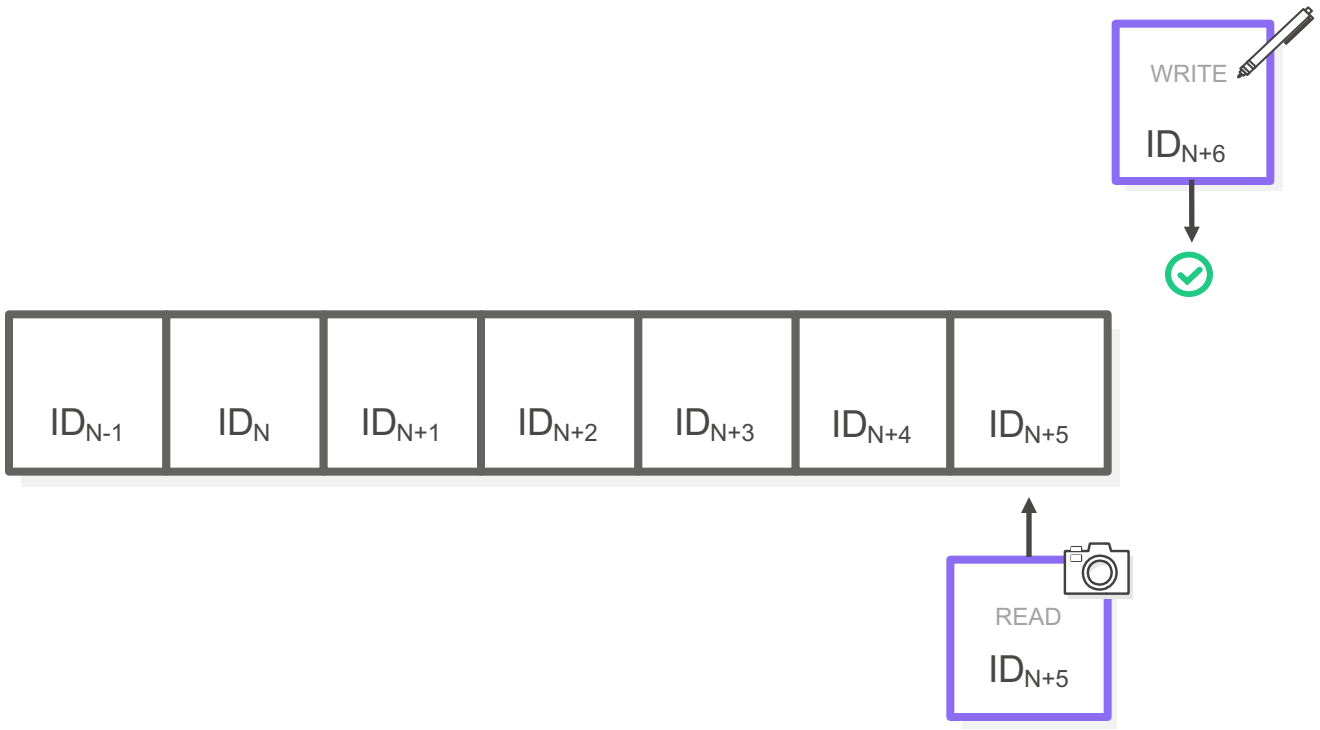


Terminated Container Instance

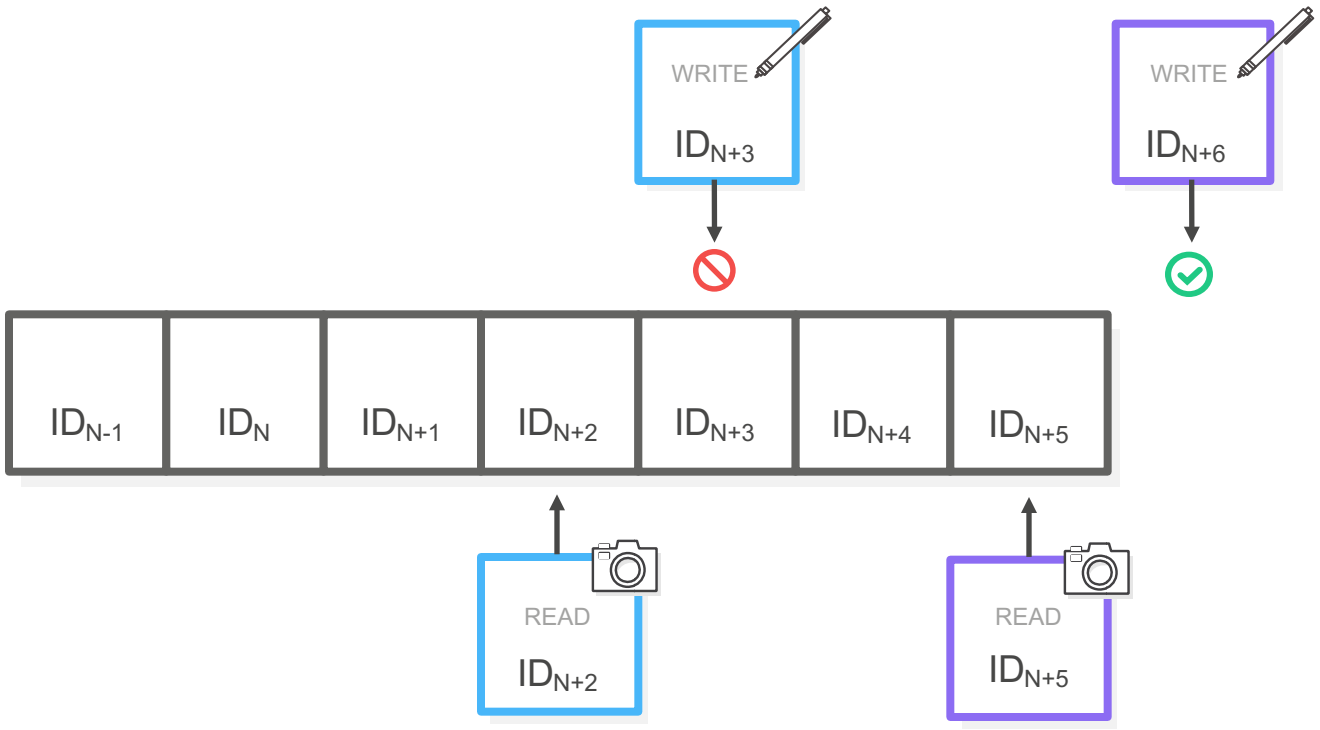


Guaranteeing Accuracy and Consistency

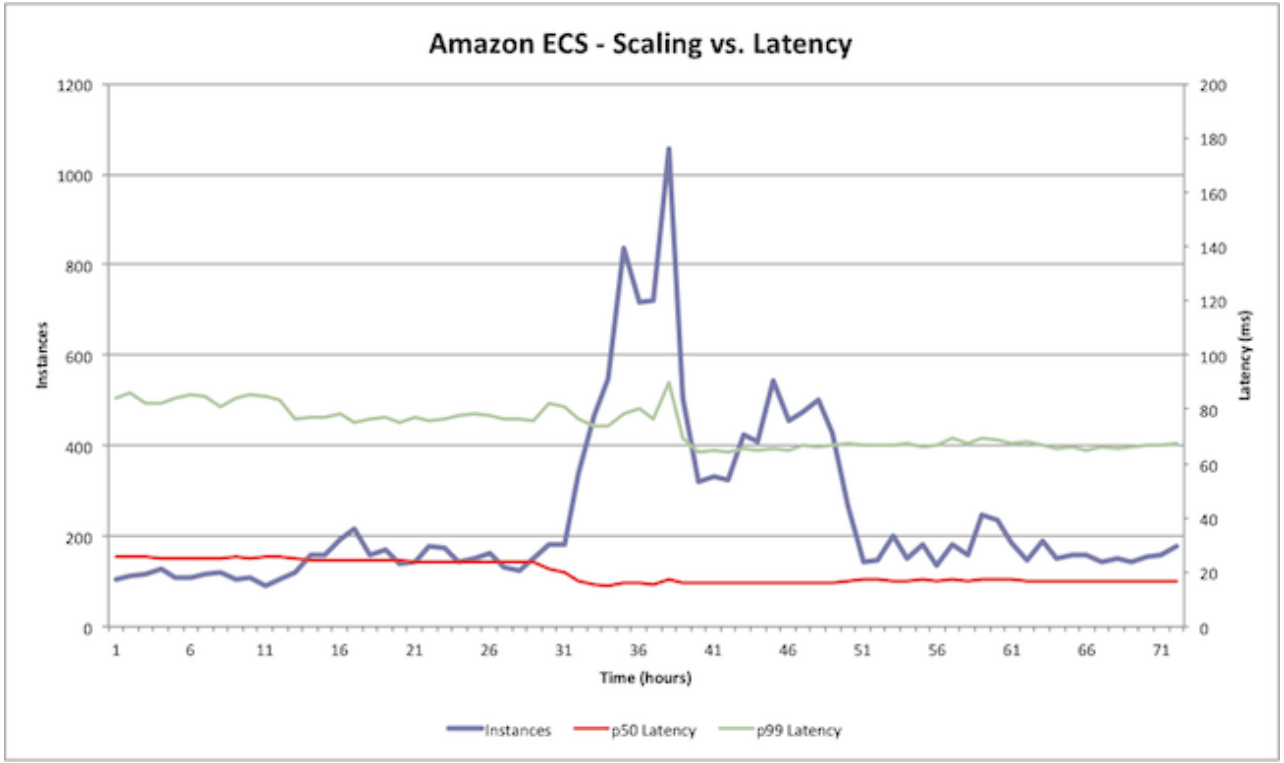
Amazon ECS under the Hood



Amazon ECS under the Hood



Scalable


Amazon ECS - Scaling vs. Latency
Instances
Latency (ms)
Time (hours)
Instances
p50 Latency
p99 Latency

Schedulers


7a
8a
9a
10a
Pick up dry cleaning
11a
12p
Eat lunch
1p
2p
Bike to the store
3p
4p
Relax
5p
6p
7p

What is a Scheduler?

- Determine Desired State
- Check Against Current State
- Perform Action

Amazon ECS Service Scheduler

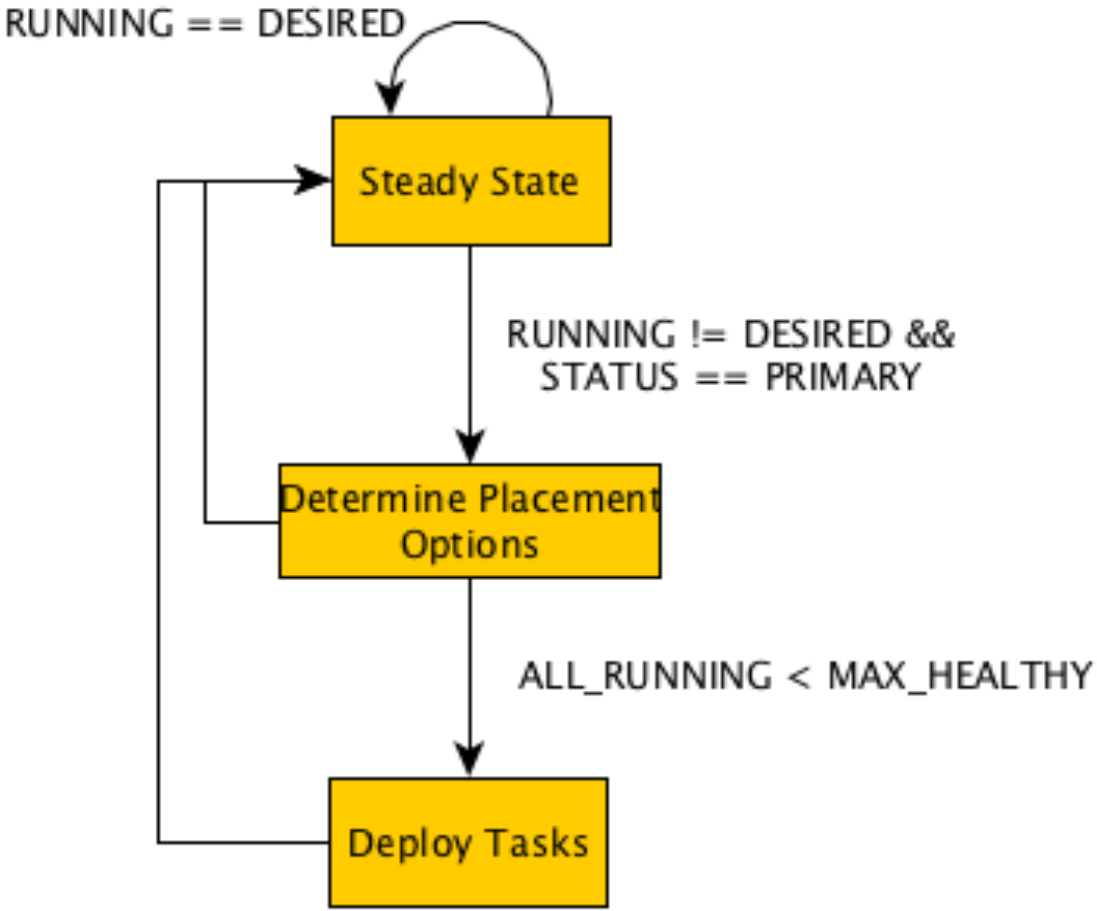
What is a Service?

- Models a long-running application
- Maintains desired state
- Optionally runs behind an Elastic Load Balancer

Discovering Differences

Deployment	Status	Desired	Pending	Running
ecs-svc/1	PRIMARY	5	0	0

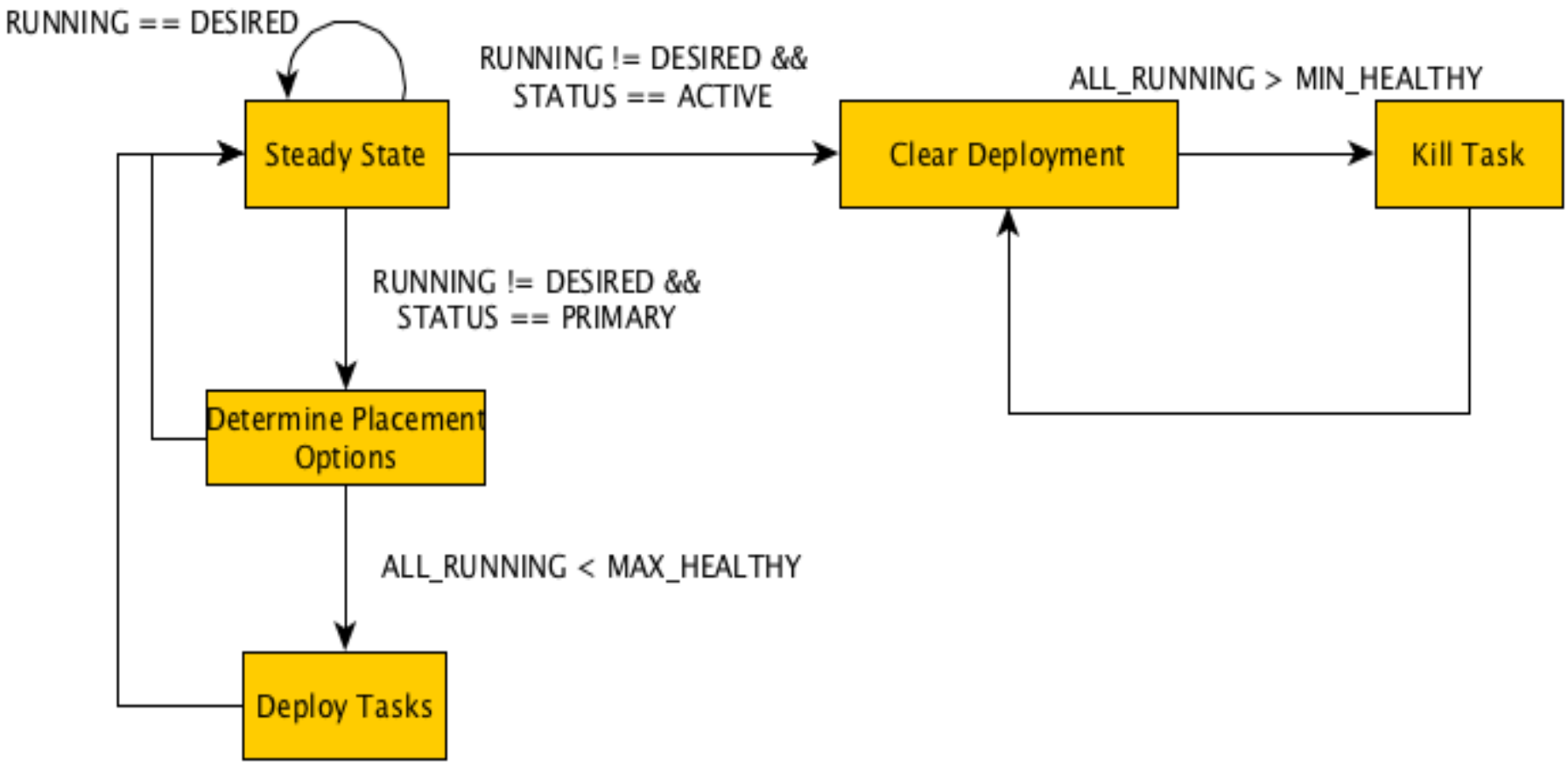
Minimum Healthy	Maximum Healthy
50%	200%


RUNNING == DESIRED
Steady State
RUNNING != DESIRED &&
STATUS == PRIMARY
Determine Placement
Options
ALL_RUNNING < MAX_HEALTHY
Deploy Tasks

Discovering Differences

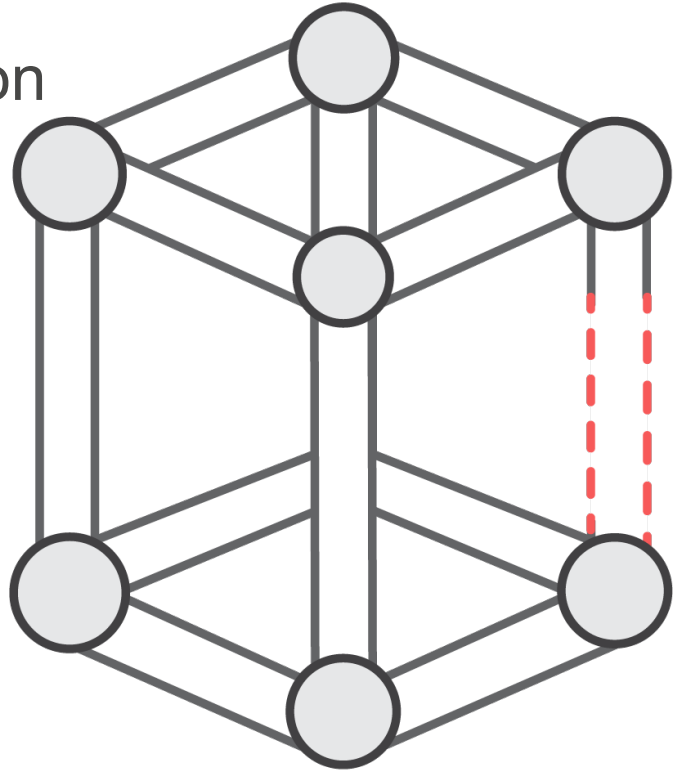
Deployment	Status	Desired	Pending	Running
ecs-svc/2	PRIMARY	10	0	0
ecs-svc/1	ACTIVE	5	0	5

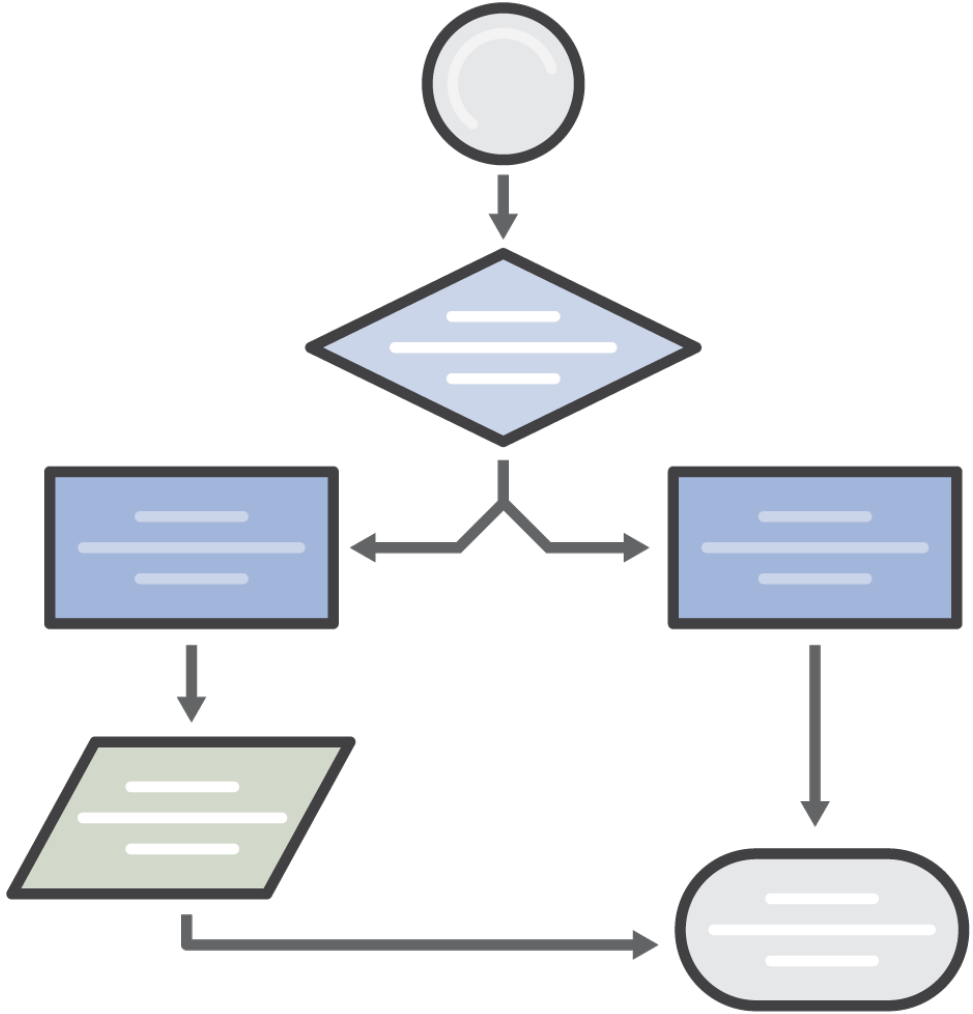
Minimum Healthy	Maximum Healthy
50%	200%


RUNNING == DESIRED
Steady State
RUNNING != DESIRED &&
STATUS == ACTIVE
Clear Deployment
ALL_RUNNING > MIN_HEALTHY
Kill Task
RUNNING != DESIRED &&
STATUS == PRIMARY
Determine Placement
Options
ALL_RUNNING < MAX_HEALTHY
Deploy Tasks

Other Considerations

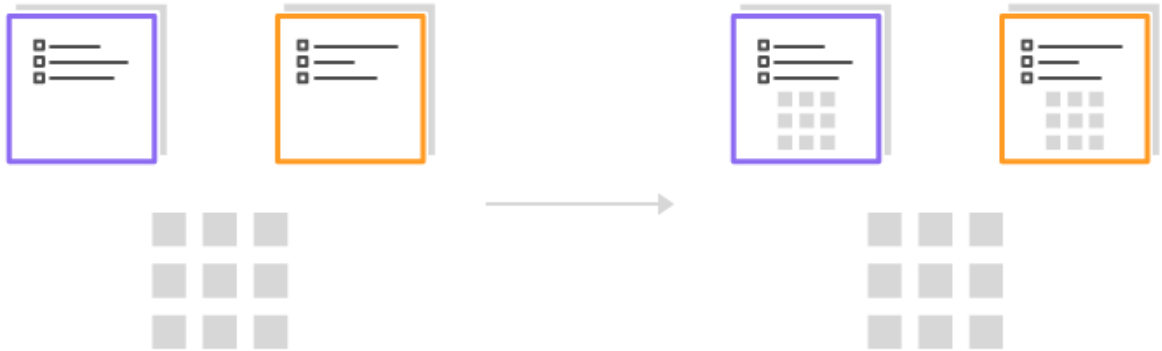
- ELB Registration/Deregistration
- Permissions and Errors
- Task Health
- Scale Down Requests



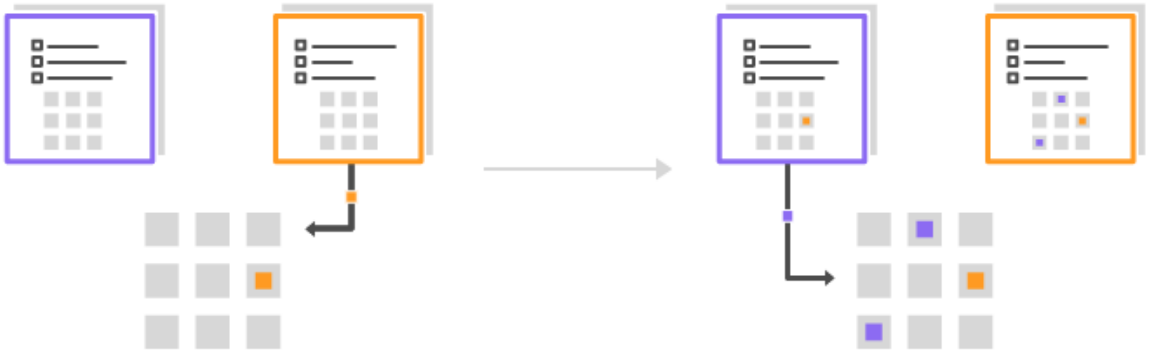


Multiple Schedulers

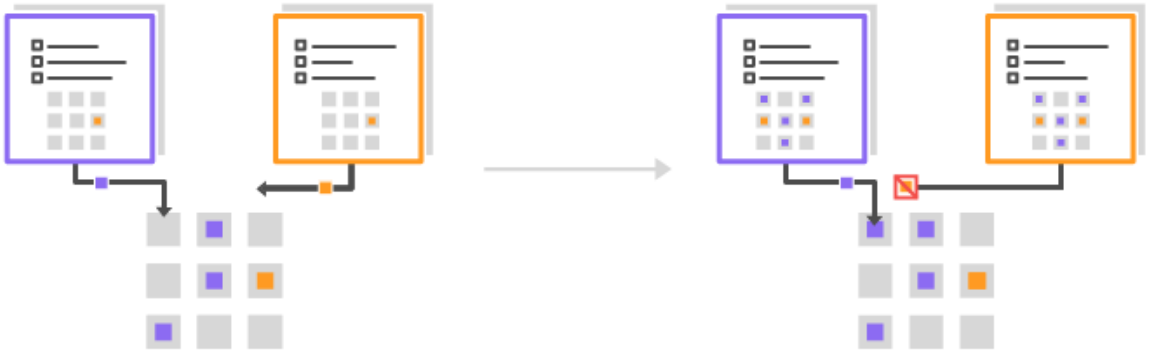
Amazon ECS: Scheduling



Amazon ECS: Scheduling

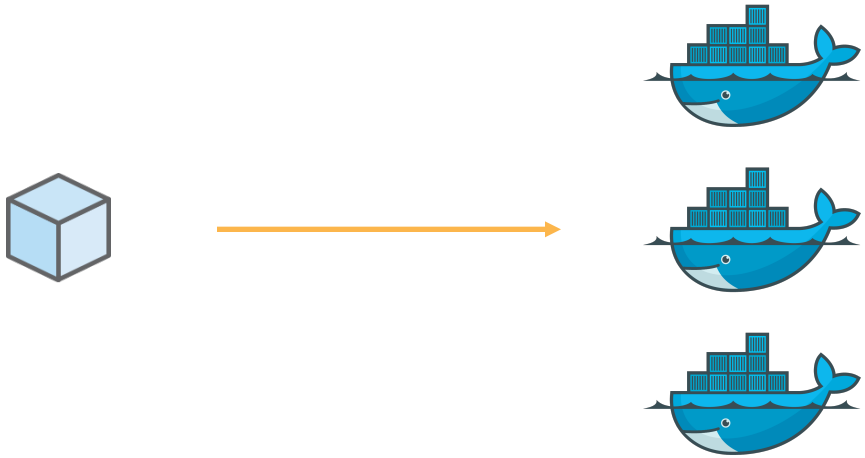


Amazon ECS: Scheduling



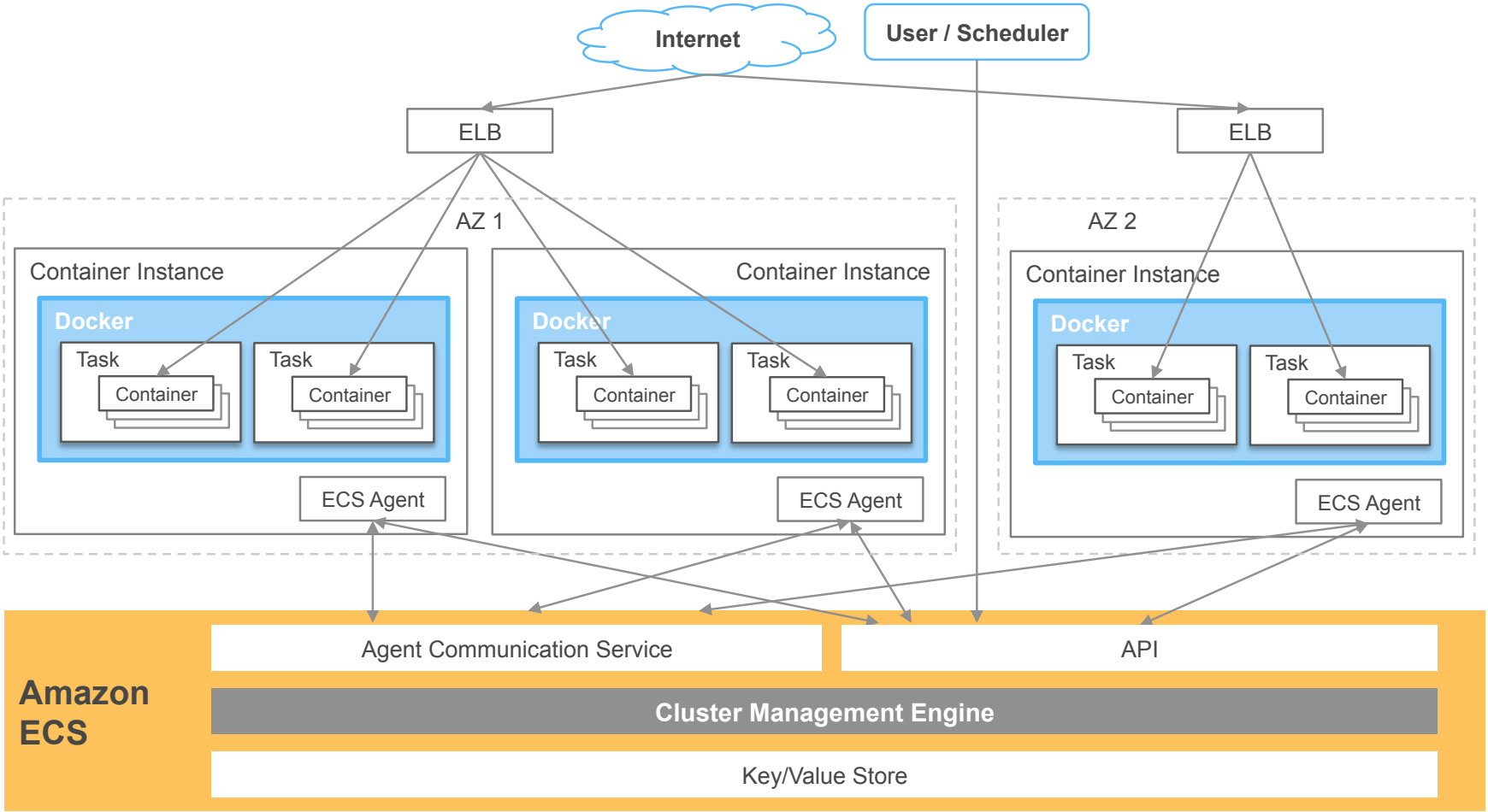
Amazon ECS: Scheduling

Let us recap



“Task Definitions”

```
{
          "environment": [],
          "name": "simple-demo",
          "image": "my-demo",
          "cpu": 10,
          "memory": 500,
          "portMappings": [
              {
                  "containerPort": 80,
                  "hostPort": 80
              }
          ],
          "mountPoints": [
              {
                  "sourceVolume": "my-vol",
                  "containerPath": "/var/www/my-vol"
              }
          ],
          "entryPoint": [
              "/usr/sbin/apache2",
              "-D",
              "FOREGROUND"
          ],
          "essential": true
      },
```


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

Thank you!